

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Industria 4.0

BlueTank 2.0: Sistema de filtrado IoT.

Trabajo Fin de Máster

presentado por: Alonso Rollán, Sergio

Director/a: Quesada Tabares, Roylan

Agradecimientos

A mi hermana Laura, por darme la idea que desencadenó toda esta locura.

A mi padre, por poner toda su experiencia, que no es poca, a mi disposición.

A mi madre y a Sandra, por no dejar que me rinda nunca.

A mi primo Avelino, por toda la ilusión y las ganas de aprender que me contagia cada día.

A Jorge y Angel, por su paciencia y sus acertados consejos.

A todos mis compañeros de trabajo, porque de ellos aprendo cada día

Resumen

El presente documento pretende ser una guía que permita al lector el entendimiento del proyecto BlueTank 2.0: Sistema de filtrado IoT. En este desarrollo entrarán se utilizarán técnicas punteras en la industria 4.0, como el IoT, la monitorización, el modelado 3D, la fabricación inteligente y el Big Data con la finalidad de desarrollar un dispositivo de filtrado IoT para acuarios domésticos. Con este proyecto se pretende dar solución a la problemática de encontrar un dispositivo IoT adaptado a la acuariofilia que permita la monitorización y control de los parámetros fundamentales para la vida en un acuario.

Este documento describirá cada fase del desarrollo del proyecto de forma detallada y con especial énfasis en aquellas decisiones de diseño que pudieran ser de especial interés para el completo entendimiento del proceso de desarrollo.

Palabras clave: Internet of Things, IoT, Monitorización, Acuario, Sensores.

Abstract

This document is a guide to enable the reader to understand the BlueTank 2.0: IoT Filtering System project. This development will use cutting-edge industry 4.0 techniques such as IoT, monitoring, 3D modelling, smart manufacturing and Big Data in order to develop an IoT filtering device for domestic aquariums. This project aims to provide a solution to the problem of finding an IoT device adapted to aquariums that allows the monitoring and control of the fundamental parameters for life in an aquarium.

This document will describe each phase of the development of the project in detail and with special emphasis on those design decisions that could be of special interest for the complete understanding of the development process.

Keywords: : Internet of Things, IoT, Monitoring, Aquarium, Sensors.

Índice de contenidos

Contenido

1. Introducción	15
1.1. Motivación	16
1.1.1. Motivación general.....	16
1.1.2. Motivación personal.....	16
1.2. Planteamiento del trabajo	16
1.3. Estructura de capítulos	17
2. Contexto y estado del arte	18
2.1. Descripción general del contexto del proyecto.....	18
2.2. Proyectos relacionados con el tema del TFM	18
2.2.1. BlueTank: Sistema de monitorización de acuarios. (Alonso Rollán, S, 2019) .	18
2.2.2. Comparison with HTTP and MQTT on Required Network Resources for IoT (Yokotani & Sasaki, 2016)	19
2.2.3. Controlador integral para acuario marino (de Rey García, 2017)	20
2.2.4. Tecnología Sustentable IoT-Mobile: Sistema de Monitoreo y Diagnóstico de Acuarios. (Rovitto, F., & Nisim, A. O. 2018)	20
2.2.5. Control y supervisión mediante un sistema microcontrolador de los parámetros de calidad de agua de un estanque. (Pérez Laguarda, 2017)	21
2.2.6. Research on 1-Wire Bus Temperature Monitoring System (Jingzhuo & Chenglong, 2007)	21
2.2.7. Soluciones comerciales	21
2.3. Tecnologías relacionadas con el tema del TFM.....	22
2.3.1. Habilitadores de la Industria 4.0.....	22
2.3.2. Dashboard.....	24
2.3.3. Software	26
2.3.4. Almacenamiento de datos	26
2.3.5. Comunicaciones	27
2.3.6. Hardware.....	29

2.3.1.	Protocolos y conceptos hardware	31
2.4.	Conclusiones sobre el estado del arte	35
3.	Descripción general de la contribución del TFM.....	36
3.1.	Objetivos	36
3.2.	Metodología del trabajo	37
3.2.1.	Distribución de roles	38
3.2.2.	Implementación de la metodología	38
3.3.	Descripción general de las partes o componentes de la propuesta	43
3.3.1.	Alcance y limitaciones	43
3.3.2.	Tecnologías implicadas	43
3.3.3.	Arquitectura, componentes e integración de tecnologías.	44
3.3.4.	Resultados esperados	46
3.3.5.	Presupuesto y retorno esperado de la inversión	46
3.3.6.	Planificación general.....	50
4.	Desarrollo específico de la contribución.....	51
4.1.	Objetivo específico: Desarrollar el hardware.....	52
4.1.1.	Sprint 1: Diseño de la electrónica	52
4.1.2.	Sprint 4: Montaje y soldadura electrónica	58
4.2.	Objetivo específico: Desarrollar y fabricar un modelo 3D.....	59
4.2.1.	Sprint 2: Diseño del modelo 3D del filtro	60
4.2.2.	Sprint 3: Fabricación del filtro	66
4.3.	Desarrollo del Firmware.....	70
4.3.1.	Lenguaje de implementación	72
4.3.2.	Bibliotecas utilizadas	72
4.3.3.	Calibración de sensores	72
4.4.	Objetivo específico: Implementar un sistema de configuración del dispositivo	75
4.4.1.	Sprint 6	76
4.5.	Objetivo específico: Mejorar la escalabilidad	78

4.5.1.	Sprint 7	79
4.5.2.	Sprint 8	81
4.6.	Objetivo específico: Mejorar la experiencia de usuario	87
4.6.1.	Sprint 9	87
4.6.2.	Sprint 10	89
5.	Conclusiones y trabajos futuros	91
5.1.1.	Líneas de trabajo futura	92
	Referencias bibliográficas	93
Anexo A.	Descripción del sistema inicial	95
A.1	Electrónica:	95
A.2	Creación de un dispositivo IoT	96
A.3	Aspecto visual:	98
A.4	Escalabilidad:	101
A.5	Plataforma IoT:	101
A.6	Sistema de alarmas:	101
Anexo B.	Manual de usuario web	102
B.1	Vista principal	103
B.2	Vista de Información	104
B.3	Vista de configuración	104
B.4	Vista a tiempo real	105
B.5	Vista de históricos	106
B.6	Alarmas	106
B.7	Vista de histórico de alarmas	108
B.8	Vista “acerca de”:	108
Anexo C.	Sistema de configuración de dispositivo	109
C.1	Conexión a la red BlueTank Wifi Manager	109
C.2	Acceso a la interfaz web	110
Anexo D.	Software desarrollado	111

D.1 Firmware.....	111
D.2 Cloud	117
D.2.1 Servidor NTP.....	118
D.2.2 Gestor de colas	118
D.2.3 Desarrollo web.	120
Anexo E. Diseño electrónica	121
Anexo F. Datasheets y certificados.....	127

Índice de figuras

Ilustración 1. Sonda DS18B20	29
Ilustración 2. Caudalímetro YF-S201	30
Ilustración 3. Sensor TSL2561.....	30
Ilustración 4. Microcontrolador ESP32	31
Ilustración 5. Conexión protocolo 1Wire.....	32
Ilustración 6. Bus protocolo 1Wire.....	32
Ilustración 7. Componentes en una comunicación I2C. Imagen tomada de la publicación de (Mankar, J., Darode, C., Trivedi, K., Kanoje, M., & Shahare, P. 2014)	33
Ilustración 8. Esquema resistencia de pull-up	34
Ilustración 9. Esquema de conexión divisor de tensión	34
Ilustración 10. Fórmula cálculo divisor de tensión	34
Ilustración 11. Vista general Trello	40
Ilustración 12. Ejemplo de Product Backlog	40
Ilustración 13. Ejemplo de tarjeta.....	41
Ilustración 14. vista general Trello con columnas.....	42
Ilustración 15. Etiquetas Trello.....	42
Ilustración 16. Arquitectura del sistema	45
Ilustración 17. Planificación temporal.....	51
Ilustración 18. Documento Sprint Backlog del Sprint 1.....	52
Ilustración 19. Esquema conceptual PCB electrónica	53
Ilustración 20. Conexión sensor temperatura DS18B20.....	54
Ilustración 21. Conexión caudalímetro YF-S201	54
Ilustración 22. Conexión sensor lux TSL2561	55
Ilustración 23. Cálculo de ancho de pistas.....	56
Ilustración 24. Placa diseñada con Altium.....	56
Ilustración 25. Esquemático del diseño de la PCB	57
Ilustración 26. Vista delantera PCB.....	58

Ilustración 27. Vista trasera PCB	58
Ilustración 28. Documento Sprint Backlog del Sprint 4.....	58
Ilustración 29. Placa soldada y montada en la caja con conectores prensaestopas.....	59
Ilustración 30.Documento Sprint Backlog del Sprint 2.....	60
Ilustración 31. Ficha técnica del filtro	64
Ilustración 32. Ficha técnica de la cesta del filtro	65
Ilustración 33. Documento Sprint Backlog del Sprint 3.....	66
Ilustración 34. Simulación por elementos finitos y cálculo de deformaciones.....	67
Ilustración 35. Pieza fabricada con tecnología FDM	68
Ilustración 36. Prototipo de sistema de filtrado IoT completo	68
Ilustración 37. Instalación del filtro general	69
Ilustración 38. Instalación del filtro detalle.....	69
Ilustración 39. Documento Sprint Backlog del Sprint 5.....	70
Ilustración 40. Diagrama de flujo general de ejecución del firmware	71
Ilustración 41. Diagrama de flujo detalle de manejadora efecto Hall	71
Ilustración 42. Recta de calibración luxómetro	73
Ilustración 43. Recta de calibración del caudalímetro	75
Ilustración 44. Documento Sprint Backlog del Sprint 6.....	76
Ilustración 45. Añadido de sistema de configuración wifi en el firmware	76
Ilustración 46. Diagrama de flujo de la configuración wifi en el firmware	77
Ilustración 47. Interfaz de la configuración wifi en el firmware	78
Ilustración 48. Documento Sprint Backlog del Sprint 7.....	79
Ilustración 49. Documento Sprint Backlog del Sprint 8.....	81
Ilustración 50. Añadido de sistema de configuración MQTT en el firmware	82
Ilustración 51. Diagrama de flujo del bucle principal del firmware modificado para envío MQTT	83
Ilustración 52. Diagrama de flujo de recepción de parámetros de configuración MQTT	84
Ilustración 53. Diagrama firmware completo	86

Ilustración 54. Documento Sprint Backlog del Sprint 9.....	87
Ilustración 55. Documento Sprint Backlog del Sprint 10.....	89
Ilustración 56. Visión en tiempo real	90
Ilustración 57. Visión datos históricos	91
Ilustración 58. Visualización de alerta.....	91
Ilustración 59. Prototipo Bluetank inicial.....	95
Ilustración 60. Electrónica Bluetank final.....	96
Ilustración 61. Dispositivo Bluetank final 1	Ilustración 62. Dispositivo Bluetank final 2
.....	97
Ilustración 63. Dispositivo Bluetank montaje final detalle	97
Ilustración 64. Dispositivo Bluetank montaje final en acuario	98
Ilustración 65. Interfaz Bluetank antigua 1	98
Ilustración 66. Interfaz Bluetank antigua 2	99
Ilustración 67. Interfaz Bluetank final. Vista históricos.....	99
Ilustración 68. Interfaz Bluetank final. Vista tiempo real	99
Ilustración 69. Interfaz Bluetank móvil. Vista tiempo real	100
Ilustración 70. Interfaz Bluetank tablet. Vista tiempo real	100
Ilustración 71. Esquemático del diseño de la PCB	121
Ilustración 72. Diseño de la PCB. Vista delantera	122
Ilustración 73. Diseño de la PCB. Vista trasera.....	123
Ilustración 74. Render de la PCB. Vista delantera	124
Ilustración 75. Render de la PCB. Vista trasera	125
Ilustración 76. Render de la PCB. Vista 3D.....	126

Índice de tablas

Contenido	6
Tabla 1. Presupuesto de componentes.....	46
Tabla 2. Presupuesto costes ingeniería	48
Tabla 3. Presupuesto molde de fabricación de pieza	48
Tabla 4. Presupuesto molde de fabricación de pieza por inyección	48
Tabla 5. Costes totales de fabricación	49
Tabla 6. Costes fijos mensuales	49
Tabla 7. Parámetros principales de impresión	68
Tabla 8. Comparativa entre sensor y luxómetro calibrado	73
Tabla 9. Tabla de transformación ofrecida por el fabricante.....	74

1. Introducción

El concepto de Internet of Things (Radouan Ait Mouha, 2021) o, de forma abreviada, IoT ha revolucionado el mundo de internet y la interconexión global, llegando a ser una de las tecnologías más dominantes en la industria actual. La aplicación de técnicas IoT permite conocer en tiempo real y de forma remota el estado de aquellos procesos o elementos que queremos monitorizar, además de posibilitar la actuación remota y/o automática en caso de necesidad. Estas técnicas, unidas a los habilitadores de la industria 4.0, como el Big Data, permiten el control, optimización de tiempo y recursos y la maximización de resultados de aquellos procesos industriales o domésticos sobre los que se aplican.

La monitorización y recopilación de datos, unida a las técnicas de análisis de estos son una de las combinaciones tecnológicas más potentes y que más están modelando el panorama industrial actual.

Este proyecto, pretender ser culminación del proyecto *BlueTank: Sistema de monitorización de acuario* (Alonso Rollán, S, 2019), desarrollado por mí mismo, como trabajo de fin de grado (TFG) en el grado de Ingeniería informática en la Universidad de Salamanca.

La idea surgió al unir dos de mis grandes pasiones, la acuariofilia y la tecnología. El proyecto pretendía dar solución a la dificultad de conocer los parámetros del agua en tiempo real y mantener un histórico de los mismos, permitiendo responder a tiempo ante cualquier situación adversa, como una rotura del calentador de agua. El resultado fue un prototipo básico, casi más cercano a una maqueta funcional, que consistía en un microordenador al que se conectaban los sensores de temperatura, luminosidad y un caudalímetro además de proporcionar la capacidad de controlar el encendido y apagado de luces. Las lecturas de los sensores se enviaban a una base de datos remota centralizada en un servidor, a partir de la cual la plataforma de visualización tomaba los datos para mostrárselos al usuario, permitiendo personalizar alertas y configurando el rango de encendido de luces.

La propuesta aquí planteada pretende, adaptando y mejorando el prototipo anterior, evolucionar los sistemas de filtrado tradicionales para obtener un primer prototipo de sistema de filtrado IoT

La aplicación de técnicas propias de la industria 4.0, como el IoT, la innovación digital, el Big data o el modelado 3D sobre el prototipo existente permitirá evolucionar este con el fin de ofrecer una solución de filtrado IoT más completa y escalable. Proporcionando al usuario una solución de monitorización completa y de instalación no más compleja que un sistema de filtrado tradicional.

1.1. Motivación

El apartado abordará la motivación del proyecto desde diferentes puntos de vista:

1.1.1. Motivación general

El desarrollo de este proyecto pretende dar solución a la problemática de encontrar un sistema IoT que permita la monitorización y el control de los parámetros fundamentales para la vida en un acuario doméstico. La acuariofilia puede llegar a ser una afición compleja y de costes elevados. Haciendo necesaria la aparición de herramientas tecnológicas que faciliten la tarea y permitan al usuario conocer el estado del agua para detectar posibles desajustes incompatibles con la vida.

1.1.2. Motivación personal.

En el apartado personal, me gustaría culminar el proyecto que comencé como trabajo de fin de grado en la carrera. Por no mencionar, la satisfacción personal que obtendría demostrando, a mí mismo y ante un tribunal, los conocimientos adquiridos durante el máster de Industria 4.0.

Pretendo, además, crear una herramienta con utilidad real, que me ayude en mi día a día como aficionado a la acuariofilia.

1.2. Planteamiento del trabajo

Este proyecto abordará el desarrollo una herramienta capaz de dar solución a la problemática de encontrar un sistema IoT que permita la monitorización y el control de los parámetros fundamentales para la vida en un acuario, siendo estos: Temperatura, cantidad de luz (en luxes) y suciedad del agua (medido mediante el porcentaje de obstrucción del filtro).

Para algunos parámetros, como la temperatura, encontramos herramientas simples y de bajo coste, como un termómetro. No obstante, este no permitirá realizar un seguimiento de la evolución de la temperatura a lo largo de las estaciones. En el caso de parámetros como la cantidad de luz o el grado de suciedad, no es posible encontrar soluciones comerciales asequibles ni estarán optimizadas para su aplicación en la acuariofilia.

El hecho de que alguno de estos parámetros se dispare fuera de rangos óptimos causará desajustes en el delicado equilibrio del acuario que puedan ser incompatibles con la vida.

Existe la necesidad de la aparición de herramientas tecnológicas que faciliten la tarea de monitorización, permitiendo al usuario conocer, de forma sencilla y rápida, el estado del acuario para permitir la actuación del usuario en caso de necesidad.

1.3. Estructura de capítulos

Este documento se subdivide en diferentes capítulos.

- **Capítulo 1. Introducción:** El capítulo de introducción pretende introducir al lector en la lectura del documento de este proyecto, ayudando a establecer una idea general de lo que se encontrará en él. Definiendo el marco tecnológico y el ámbito de trabajo sobre el que se sustenta el proyecto y permitiendo, de forma rápida, situar al lector en la problemática abordada por esta solución.
- **Capítulo 2. Contexto y estado del arte:** El capítulo 2 versará sobre el análisis del panorama actual respecto a la aplicación de este tipo de tecnologías al ámbito de la acuariofilia. Realizará una síntesis sobre diversas publicaciones científicas que traten tecnologías similares a las aquí planteadas. Se realizará un estudio de las tendencias más punteras de mercado en cuanto a las tecnologías subyacentes al desarrollo. El objetivo de este capítulo es recoger diversos estudios, con validez científica, sobre los que apoyar las decisiones de diseño.
Para finalizar, el capítulo termina describiendo dispositivos con una finalidad similar al planteado en este proyecto.
- **Capítulo 3. Descripción general de la contribución del TFM:** Este capítulo tratará de esclarecer los objetivos que se pretenden cumplir con la consecución de este proyecto de fin de estudios, describir la metodología de trabajo que marcará el desarrollo y describir las diferentes partes o componentes en los que se subdivide el sistema propuesto, describiendo además la arquitectura planteada para el mismo.
- **Capítulo 4. Desarrollo específico de la contribución:** El capítulo 4 describe el proceso de desarrollo del sistema bajo la metodología seleccionada. Donde se describirán en detalle los pasos implementados y las decisiones tomadas a lo largo del desarrollo que permitirá la consecución de los diferentes objetivos en los que se subdivide el presente proyecto.
- **Capítulo 5. Conclusiones y trabajo futuros:** El capítulo 5 tratará de sintetizar las conclusiones obtenidas tras el desarrollo del sistema, si ha sido o no posible el desarrollo y por qué. Resaltando los aspectos que podrían ser mejorables y estableciendo cuál sería la evolución natural de este proyecto.

2. Contexto y estado del arte

2.1. Descripción general del contexto del proyecto

Este punto tiene la finalidad de ofrecer al lector la perspectiva sobre las tecnologías y estudios que marcarán y caracterizarán este proyecto.

El trabajo se desarrollará en el marco de las tecnologías relacionadas con el concepto de Internet of Things (IoT) acompañadas de habilitadores de la industria 4.0 como el Big data, la simulación 3D o la fabricación inteligente.

El ámbito principal del proyecto es el de Internet of Things (IoT), debido a que la finalidad principal del mismo consiste en monitorizar y actuar sobre un elemento concreto, tratándose en este caso un acuario doméstico.

La captura masiva de mediciones alcanza su máximo punto de interés en el momento en el que estos datos son analizados y estudiados con el fin de encontrar patrones, realizar predicciones y actuar sobre el mundo real maximizando una medida de rendimiento, siendo este el propósito principal del Big data.

Las técnicas de modelado y simulación 3D se convierten en grandes aliados a la hora de diseñar un producto, permitiendo visualizar el resultado final antes de fabricarlo, con el fin de detectar fallos en el planteamiento o en el diseño y que pudieran llegar imposibilitar la consecución final de los objetivos.

2.2. Proyectos relacionados con el tema del TFM

Se realiza una investigación con el fin de encontrar casos de estudio similares al propuesto o publicaciones científicas basadas en tecnologías susceptibles de ser utilizadas en este proyecto de fin de estudios.

Existen multitud proyectos basados en el uso de la tecnología IoT, no obstante, en este estudio nos centraremos en el uso de este tipo de sistemas aplicados al ámbito de acuarios domésticos, o en publicaciones y estudios que aborden tecnologías similares a las planteadas para la consecución de este proyecto.

2.2.1. BlueTank: Sistema de monitorización de acuarios. (Alonso Rollán, S, 2019)

El proyecto desarrollado por Sergio Alonso Rollán como Trabajo de Fin de Grado del grado en Ingeniería informática de la Universidad de Salamanca, sentará las bases sobre las

que partirá este trabajo de fin de estudios. BlueTank consiste en un sistema de muestreo y monitorización de un acuario. Dicho sistema permitirá al usuario conocer los valores de los parámetros fundamentales para garantizar el correcto equilibrio del acuario, en cualquier momento del día y desde cualquier lugar, de forma rápida y visual. Además, el usuario tendrá a su disposición los datos recogidos con anterioridad, permitiendo consultar datos históricos e incluso pudiendo realizar comparativas entre datos recogidos en diferentes fechas, con el fin de ver la evolución de dichos parámetros a lo largo del tiempo.

Los parámetros que se medirán serán la temperatura (sonda DS18B20), la iluminancia (sensor TSL2561) y el caudal del filtro (sensor YF-S201). Se han escogido estas variables por ser las más representativas a la hora de mantener bajo control la salud de nuestro acuario. Contaremos con diferentes sensores especializados en dichos parámetros y un microprocesador capaz de gestionar las lecturas de los sensores.

Esta herramienta busca ampliar el conocimiento que el usuario puede tener de su acuario. Permitiendo que el usuario mantenga bajo control las variables observadas, sin depender de realizar lecturas manuales, más imprecisas y tediosas. De esta manera el usuario será capaz de responder ante valores inadecuados que podrían desencadenar problemas mayores en el acuario, tales como aparición de algas.

Este proyecto cuenta con dos partes, la lectura y validación de parámetros, también llamada backend, que se ha realizado en lenguaje Python, cuya misión es realizar mediciones correctas y almacenarlas en la base de datos para su posterior consulta. Por otro lado, tenemos el frontend, que consiste en una interfaz web capaz de representar los datos obtenidos de la base de datos de forma gráfica, con el fin de que sea fácilmente visualizable por el usuario.

2.2.2. Comparison with HTTP and MQTT on Required Network Resources for IoT (Yokotani & Sasaki, 2016)

Este artículo aborda la problemática que supone utilizar HTTP cuando se trata transmitir por la red grandes cantidades de pequeños bloques de datos. Este problema, es causado en gran medida, debido a la sobrecarga de peticiones.

Un ámbito en el que esta problemática sucede es el de Internet of Things, en el que tendremos una gran cantidad de dispositivos emitiendo pequeños bloques de datos en la red.

El artículo en cuestión compara los rendimientos de arquitecturas IoT verticales frente a arquitecturas IoT horizontales en las que el procesamiento de la información y la comunicación

se realizan de forma compartida. Explicando la importancia de reducir al máximo el número de peticiones TCP/IP para maximizar el rendimiento.

Como conclusión, obtenemos que MQTT, por su construcción asimétrica, reduce la carga de peticiones TCP/IP, por lo que es un protocolo que soporta mejor la alta carga de pequeños bloques de información. Este hecho queda demostrado mediante una gráfica comparativa del ancho de banda necesario para transmitir las peticiones de 1000 dispositivos utilizando el protocolo HTTP frente al protocolo MQTT, obteniendo como resultado que MQTT necesita un ancho de banda hasta tres veces menor.

2.2.3. Controlador integral para acuario marino (de Rey García, 2017)

Se trata de un proyecto de fin de grado para el grado de ingeniería informática. Propone una solución de monitorización IoT para un acuario marino. Pretende ofrecer una solución de monitorización y control sobre el acuario, con la que el usuario interactuará mediante una plataforma web. Permitiendo monitorizar la temperatura y el nivel del agua, y controlando la iluminación y las bombas de circulación de agua con el fin de mantener la temperatura y la salinidad del entorno.

Utiliza sensor de temperatura DS18B20, un sensor de nivel de agua basado en una boya todo o nada que se activará cuando el nivel esté por debajo de la boya, un reloj para el control horario y un sistema de interruptores que permiten controlar el encendido y apagado de los elementos electrónicos.

2.2.4. Tecnología Sustentable IoT-Mobile: Sistema de Monitoreo y Diagnóstico de Acuarios. (Rovitto, F., & Nisim, A. O. 2018)

El proyecto presenta una propuesta de IoT para la monitorización de parámetros relevantes en un acuario. Propone una arquitectura IoT característica mediante la cual los datos recogidos se vuelcan a un servidor, donde con consultados por una aplicación IOS/Android que permite su visualización.

El proyecto en cuestión utiliza los sensores DS18B20 para temperatura, SEN0161 para el pH y DFR0300 como sensor de conductividad. A partir de estos sensores pretende determinar la temperatura, la calidad del agua y el nivel de oxígeno disuelto.

2.2.5. Control y supervisión mediante un sistema microcontrolador de los parámetros de calidad de agua de un estanque. (Pérez Laguarda, 2017)

El proyecto en cuestión hace referencia a la problemática de encontrar un producto comercial que permita el control y la monitorización de los parámetros fundamentales para asegurar la calidad del agua en un acuario. El proyecto en cuestión pretende implementar un sistema de control y monitorización de coste reducido que trabajará sobre los diferentes parámetros del agua, entre ellos, la temperatura o el PH.

2.2.6. Research on 1-Wire Bus Temperature Monitoring System (Jingzhuo & Chenglong, 2007)

Se trata de un estudio basado en una sonda de temperatura DS18B20, que funciona bajo el protocolo de comunicación 1Wire y un chip de transformación de protocolo DS2480. El estudio realizado da lugar a un circuito de interfaz que actúa como intérprete entre el bus 1Wire de la sonda DS18B20 y el puerto COM del chip DS2480.

Aplicado a nuestra problemática actual, será interesante atender al diseño de hardware desarrollado para la conexión de múltiples sondas de temperatura DS18B20, que, más allá de su versión de encapsulado, mantienen el mismo mecanismo de conexión.

2.2.7. Soluciones comerciales

Existen diferentes soluciones comerciales que pretenden suplir la misma problemática que este proyecto. Sin embargo, ninguna solución comercial proporciona una solución tan completa como la que aquí se plantea.

Por línea general, estas soluciones, tan solo proporcionan un control de apagado/encendido de luces, en función del horario, y un monitor de temperatura con un control todo o nada. No proporcionan la capacidad de mantener históricos ni un control basado en la luz natural recibida desde el exterior ni la posibilidad de realizar una consulta remota de los parámetros medidos.

Las soluciones comerciales tampoco ofrecen la posibilidad de monitorizar el nivel de obstrucción del sistema de filtrado, lo que es vital para conocer el nivel de suciedad del acuario.

Soluciones comerciales:

- Controlador de temperatura Elitech STC-1000PRO (*Elitech STC-1000PRO Controlador de temperatura multiusos de calefacción y refrigeración con termostato digital*, s. f.)
- Controlador de temperatura KETOTEK (*KETOTEK Digital Termostato Enchufe Calentamiento Enfriamiento con Sonda 220V, Controlador de Temperatura Enchufe Momento Cuenta Regresiva para Incubadora Invernadero Acuario Reptil: Amazon.es: Bricolaje y herramientas*, s. f.)

2.3. Tecnologías relacionadas con el tema del TFM

2.3.1. Habilitadores de la Industria 4.0

2.3.1.1. IoT

El concepto IoT o Internet of Things hace referencia a la conexión de objetos tradicionales a la red de internet. Este concepto tuvo su origen en 1999, cuando fue propuesto por Kevin Ashton de Massachusetts Institute of Technology (MIT). “Internet of Things” ha revolucionado el mundo tal y como lo conocemos. Dotar a los objetos cotidianos de la posibilidad de conectarse a internet, comunicarse con su entorno e interactuar con él abre la puerta a un elevado número de nuevas posibilidades. Existen multitud de estudios y autores que han tratado el tema de IoT, algunos como (Radouan Ait Mouha, 2021) o (Madakam et al., 2015) son de particular interés por los estudios realizados sobre el impacto y la transformación que ha supuesto el IoT sobre los objetos cotidianos, la forma de interactuar con ellos y en definitiva, el mundo tal y como lo conocemos.

Ligado al concepto de IoT, surgió la sensorización, dotando a estos objetos de la capacidad de realizar mediciones sobre su estado o el de su entorno en tiempo cuasi real. La capacidad que ofrecen este tipo de dispositivos para monitorizar su funcionamiento, responder ante estímulos externos y generar grandes cantidades de datos recogidos en un entorno real, a supuesto un antes y un después en el panorama industrial de la actualidad.

Generar una gran cantidad de información diaria implica disponer de sistemas de almacenamiento masivos, rápidos y accesibles, lo que ha fomentado el desarrollo de nuevos motores y paradigmas de bases de datos, dando lugar a sistemas de almacenamiento descentralizados. Por otra parte, esta acumulación masiva de datos ha permitido el desarrollo de otros campos de la ingeniería como son el Big data.

La combinación de IoT y Big data permite maximizar el rendimiento y minimizar el consumo de recursos por parte de este tipo de objetos evolucionados.

2.3.1.2. Big Data

El concepto de Big data hace referencia a la ciencia de la computación que pretende extraer conocimiento útil a partir del estudio sistemático de grandes cantidades de datos. Atendiendo a la argumentación de (Puyol Moreno, 2014), podemos afirmar que el propósito final del Big data servir como soporte para la toma de decisiones que, en base a los resultados obtenidos del estudio de los datos, permitan maximizar una medida de rendimiento.

La aplicación de diferentes técnicas de Big data permitirán comprender, resumir o representar mejor los datos a los que se apliquen, permitiendo identificar patrones, predecir tendencias y clasificar elementos.

La aparición y desarrollo del Big data ha permitido a las grandes compañías, mejorar la toma de decisiones, reducir costes y aumentar las ventas, a través del estudio de patrones de comportamiento de los clientes. Pero estas técnicas no solo son aplicables al estudio de los datos humano, sino que pueden aplicarse al comportamiento de máquinas, permitiendo que estas tomen decisiones de forma autónoma y acumulen conocimiento, pudiendo responder ante imprevistos y siendo capaces de predecir colapsos del sistema o situaciones desfavorables para responder antes de que esta situación ocurra.

2.3.1.3. Modelado y simulación 3D

El modelado 3D es una técnica que permite realizar un diseño digital del producto a fabricar. Este tipo de modelador permiten visualizar el resultado del prototipo final antes de que este sea construido.

Esta técnica de diseño permite detectar posibles fallos de diseño, de ensamblaje o irregularidades en la pieza. Existe la posibilidad de realizar simulaciones a partir de la pieza diseñada. Pudiendo realizar test de fatiga de materiales, test químicos, test aerodinámicos etc.

El modelado 3D y la simulación 3D han tomado una relevancia tal, que han transformado el proceso de diseño y testeo, convirtiéndose en un paso obligatorio en todo proceso de ingeniería. Para apoyar esta afirmación solo es necesario observar la gran cantidad de estudios realizados en base al modelado y simulación 3D y sus aplicaciones tan heterogéneas. Estos estudios abarcan desde el análisis de defectos en soldadura (Santiago, D., Pereyra, S., Lombera, G., & Urquiza, S. 2006), o aplicadas a la medicina, concretamente a la fabricación de férulas (Herrera Gil, 2019).

Se trata de una tecnología que encaja muy bien con las técnicas de fabricación aditiva tan populares en la actualidad. Haciendo que el proceso de obtener un prototipo de pruebas sea mucho más ágil y simple.

En la actualidad se han desarrollado multitud de softwares que permiten la posibilidad de diseñar piezas con formas complejas de forma sencilla, y realizar las pertinentes pruebas, simulaciones y modificaciones para lograr un resultado óptimo.

2.3.1.4. Fabricación inteligente

Se trata de un proceso de fabricación avanzada, íntegramente ligado al modelado 3D, que se basa en el uso de tecnologías de información y comunicación (TIC), junto con tecnologías de procesamiento de datos, como el Big Data, mediante las cuales se pretende obtener un mecanismo de fabricación óptimo, eficiente, inteligente y autónomo.

Entre estas técnicas de fabricación, destaca la técnica “Fused Deposition Modeling” o “Modelado por Deposición Fundida”, más conocida como FDM. Se trata de una técnica mediante la cual se funde el material y es depositado en forma de capas apiladas dando lugar al objeto que se desea obtener. Esta es una técnica asequible muy útil para prototipar, como será el caso del prototipo desarrollado en este proyecto.

2.3.1.5. Sensores y protocolos de comunicación

Se trata de un habilitador de la industria 4.0, directamente relacionado con Internet of Things y la sensorización de procesos. Este habilitador nos aporta conocimientos sobre los diferentes tipos de sensores y sus cualidades y los protocolos de bajo nivel necesarios para la comunicación con estos. Un aspecto importante respecto a la sensorización será la calibración del sensor.

Es posible encontrar sensores para todo tipo de magnitudes tales como visión artificial, proximidad, movimiento, velocidad, temperatura, humedad, vibración, presencia de diferentes químicos, fuerza, carga, presión, etc. entre otros.

Algunos de los protocolos más comunes en el hardware son: I2C, SPI, UART, 1Wire o CAN Bus.

2.3.2. Dashboard

El término Dashboard IoT se refiere a un panel, habitualmente en un entorno web o de escritorio, que permite a un usuario remoto visualizar de forma gráfica y simple la información

recogida por los sensores. Permite, además, la interacción de usuario y sistema, de forma que el primero pueda comandar al segundo o programar respuestas controladas que se efectuarán cuando se produzcan una cierta situación irregular.

2.3.2.1. Thingsboard (Deep Intelligence, s. f.)

Plataforma lot de código abierto que proporciona herramientas para la recogida de datos, permite la personalización de dashboard de visualización, el procesamiento de datos mediante cadenas de reglas y la respuesta frente a estos. Admite conectividad IoT mediante protocolos MQTT, CoAP y HTTP para la recogida y el envío de datos y cuenta con un sistema de almacenamiento escalable y basado en la disponibilidad.

2.3.2.2. Deep Intelillence (Deep Intelligence, s. f.)

Deep intelillence o Deepint es una plataforma web orientada al Big data. El objetivo de la plataforma es guiar al usuario, permitiendo la aplicación de técnicas de inteligencia artificial, machine learning o algoritmos complejos de forma visual y sin necesidad de un amplio conocimiento previo de en la materia. La plataforma admite múltiples fuentes de datos de formato variado, incluyendo bases de datos relacionales y no relacionales, y proporciona al usuario la posibilidad de crear un dashboard de visualización exportable, de tal forma que pueda ser embebidos en cualquier otra plataforma web o móvil.

La usabilidad y viabilidad de esta herramienta ha sido demostrada en numerosos estudios realizados como por ejemplo “Deepint.net: una plataforma de implementación rápida para territorios inteligentes” (Corchado et al., 2021), donde se demuestra la viabilidad de la herramienta para para crear una plataforma ciber física eficiente para la gestión de territorios inteligentes. Permitiendo el uso de fuentes de datos variadas, desde mediciones de dispositivos IoT hasta bases de datos, relacionales y no relacionales. Proporcionando la capacidad de procesamiento de datos mediante técnicas de inteligencia artificial para el análisis de datos, incluyendo técnicas de clasificación de datos, clustering, pronóstico, optimización, visualización, entre otras.

2.3.2.3. Grafana (Grafana: The Open Observability Platform, s. f.-c)

Proyecto de código abierto que surgió para desarrollar una plataforma IoT especializada en la creación de dashboards de visualización de datos, ofrece una amplia variedad de estilos y gráficos, logrando un alto nivel de personalización. Permite la posibilidad de programar alertas

sobre las métricas establecidas y cuenta con soporte para conectar diferentes fuentes de datos como MySQL o PostgreSQL

2.3.3. Software

2.3.3.1. Altium Designer

Software de diseño electrónico para placas de circuito impreso o PCBs desarrollado por la empresa Altium Limited.

El software permite una gran variedad de funcionalidades relacionadas con el diseño de componentes hardware, entre las que destacan, el diseño de esquemáticos de conexión, diseño de placas por capas, ruteado de pistas y el diseño de PCB en 3D. Altium Designer permite los archivos gerber de cada una de las capas de la placa, necesarios para la fabricación automatizada del circuito impreso.

2.3.3.2. Autodesk Fusion360

Software de modelado de objetos 3D desarrollado por la empresa Autodesk.

Utilizado en ingeniería para el prototipado, producción y detección de errores, este software proporciona la capacidad de diseñar piezas por planos simples de dos dimensiones (2D) que posteriormente dan lugar a piezas en tres dimensiones (3D). Permite realizar el ensamblaje de piezas.

El uso de este tipo de software permite la detección temprana de errores en el diseño de la pieza. Una vez obtenido el modelo definitivo, el software aporta el archivo necesario para que la pieza pueda ser procesada y fabricada mediante técnicas de fabricación inteligente.

2.3.4. Almacenamiento de datos

Las bases de datos permitirán memorizar datos masivos obtenidos para cada dispositivo BlueTank. Estos datos podrán ser analizados y graficados posteriormente. Para este trabajo se valorarán los siguientes motores de almacenamiento, cada uno como representante de los dos tipos de tecnologías de almacenamiento más dominantes en la actualidad.

2.3.4.1. PostgreSQL

Se trata de un sistema de bases de datos relacionales SQL. Este motor de código abierto está pensado principalmente para garantizar la escalabilidad del sistema, de forma distribuida y

con mecanismos de redundancia que garantizan la robustez frente a fallos y la disponibilidad de los datos. Cuenta con diferentes herramientas gráficas que facilitan el trabajo con la herramienta.

Como todo sistema de bases de datos relacional, presenta las ventajas de: atomicidad de las operaciones, e integridad de tipos y compatibilidad de los datos, aunque cuenta con algunas características propias que no existen en todos los sistemas de bases de datos relacionales.

2.3.4.2. MongoDB

Sistema de bases de datos no relacional, encajaría en la categoría de almacén de documentos. Permite almacenar estructuras de datos complejas. Algunas de sus principales características son la indexación de cualquier campo dentro de un documento o las consultas ad hoc.

MongoDB es un sistema pensado para ser replicado, permitiendo de forma nativa la gestión de las réplicas secundarias. Otra característica interesante es la escalabilidad, se trata de un sistema escalable horizontalmente que permite realizar el balanceo de carga entre diferentes servidores.

Cuenta con las ventajas propias de un sistema de bases de datos no relacional como son la escalabilidad y descentralización, los cambios en caliente y el rendimiento en las consultas.

2.3.5. Comunicaciones

Las comunicaciones serán vitales en este proyecto, se estudiará la viabilidad de cada tipo de comunicación aquí presentada con el fin de tomar la mejor decisión tomando como objetivo aumentar la escalabilidad del sistema y soportar el mayor número de comunicaciones simultáneas posibles.

2.3.5.1. API REST

REST son las siglas de Representational State Transfer, y podría traducirse como transferencia de representación de estado. Este enfoque de desarrollo fue desarrollado por Roy Fielding en el año 2000.

Un servicio REST no es estrictamente un tipo de arquitectura software sino un conjunto de restricciones sobre las que crear aplicaciones. Las características de un servicio REST son: estará formado por una arquitectura cliente-servidor, el servicio será accesible a través de una URI y no se mantiene ningún estado, es decir, las peticiones al servidor son independientes.

Este servicio estará construido sobre el estándar HTTP, que puede ser accedido a través de URI para el intercambio de recursos.

Para resumir, diremos que un servicio es REST si cumple que:

- Es accesible a través de una URI.
- Los datos están representados en XML o JSON
- Solo admite operaciones: HTTP: GET, PUT, DELETE, POST.
- La API es accesible a través de hipertexto.

El significado de las operaciones admitidas es:

- Post: Crea un nuevo elemento en el servidor.
- Get: Obtiene un elemento del servidor.
- Put: Actualiza un elemento del servidor.
- Delete: Elimina un elemento.

Una API REST se puede definir como un conjunto de servicios web que cumplen las restricciones definidas por REST.

2.3.5.2. MQTT

Message Queuing Telemetry Transport, es un protocolo ligero de transporte de mensajes Machine to Machine (M2M). El protocolo está montado usualmente sobre la red TCP/IP, actuando sobre las capas 5-7 de los niveles OSI. El protocolo se basa en la suscripción y publicación de mensajes mediante topic. El protocolo se basa en un servidor o bróker central que tiene como función registrar las suscripciones de los clientes a los diferentes topics, recibir los mensajes publicados para estos topics y redirigirlos hacia los clientes suscritos.

Un topic conecta a un nodo emisor con un suscriptor, permitiendo que cualquier nodo pueda suscribirse o publicar en cualquier topic, lo que implica una comunicación bidireccional. Los topics se representan con una cadena de texto que tiene una estructura jerárquica marcada por el carácter “/”. Un ejemplo de topic sería: casa1/habitación1, lo que nos indicaría que “habitación1” se encuentra dentro de “casa1”.

La ligereza y sencillez de este protocolo lo convierte en uno de los más recomendables para sistemas IoT. Al estar montado sobre TCP/IP, garantiza la entrega de los mensajes al destinatario. La posibilidad de definir topics en función de las necesidades es otra de las grandes ventajas, que permite, además, identificar el dispositivo IoT en función del topic.

Una aproximación más completa sobre el funcionamiento de este protocolo la podemos encontrar en la publicación de (Hunkeler et al., 2008). A pesar de ser una publicación con cierta antigüedad, hablando en términos informáticos, las bases y conceptos principales de este protocolo se han mantenido hasta nuestros días y permitiendo considerar este informe como correcto y adecuado al caso de uso presentado en estas líneas.

2.3.6. Hardware

2.3.6.1. Sensor de temperatura

Las mediciones de temperatura serán efectuadas por el sensor seleccionado. Se trata de una sonda digital de temperatura. Para este proyecto se ha seleccionado el sensor DS18B20. Se trata de un sensor muy utilizado en proyectos de IoT que funciona bajo el protocolo 1Wire. El mismo sensor cuenta con dos versiones de encapsulado. Una versión estándar, y una versión en forma de sonda sumergible, que será utilizada en esta implementación.



Ilustración 1. Sonda DS18B20

El sensor cuenta con tres terminales, los dos primeros para la alimentación a 3.3V y un tercero para la implementación del protocolo de comunicación.

Según las especificaciones del fabricante, el rango de actuación del sensor será de entre -55°C hasta 125°C, con un margen de error en la medición de 0.5°C.

Como ejemplo para este sensor, se ha tomado como referencia el estudio sobre monitorización de temperatura realizado por (Jingzhuo & Chenglong, 2007) en el que se muestra un claro ejemplo de uso del protocolo 1Wire con una sonda de temperatura de la misma familia que la sonda DS18B20, concretamente la sonda DS2480.

2.3.6.2. Sensor caudal

El caudal del filtro será medido con un sensor YF-S201. Se trata de un sensor analógico de efecto Hall. El sensor contiene un eje en su interior sobre el que giran unas aspas. El caudal de agua, al circular a través del sensor, hará girar las aspas. Con cada vuelta completa de las aspas, un imán situado en una de las aspas producirá un pulso electromagnético que será detectado por el microcontrolador. El datasheet del sensor proporciona una recta de conversión que permite transformar el número de pulsos por minuto en litros/hora.

El sensor cuenta con 2 cables de alimentación y un cable para la transmisión de pulsos. Los pulsos tendrán forma de onda cuadrada de 5V.

El caudalímetro es capaz de actuar en temperaturas de entre -25°C hasta 85°C con un error de hasta el 10%.



Ilustración 2. Caudalímetro YF-S201

2.3.6.3. Sensor de luminosidad (luxes)

La cantidad de luz recibida de forma natural en el acuario será monitorizada mediante el sensor de luminosidad TSL2561. Se trata de un dispositivo I2C capaz de trabajar tanto en el espectro de luz visible como en el rango de infrarrojos.

El sensor cuenta con dos pines de alimentación a 3.3V, y dos más para la implementación del protocolo I2C. El luxómetro puede operar entre -30°C y 80°C ofreciendo un nivel de precisión de 0.1 lux.

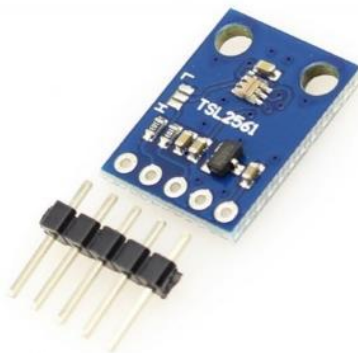


Ilustración 3. Sensor TSL2561

2.3.6.4. Microcontrolador ESP32

El término ESP32 hace referencia a una familia de microcontroladores económicos tipo SoC (System on a chip) desarrollados por Espressif Systems.

El uso de este tipo de microcontroladores está cada vez más extendido en arquitecturas para proyectos de IoT. Entre sus principales características destaca el bajo consumo energético, permitiendo funcionar durante meses conectado únicamente a batería. Este hecho, unido a la versatilidad que ofrece contar con conectividad Wifi 802.11 y Bluetooth 4, incluyendo la variante BLE (Bluetooth Low Energy) de bajo consumo, lo convierten a

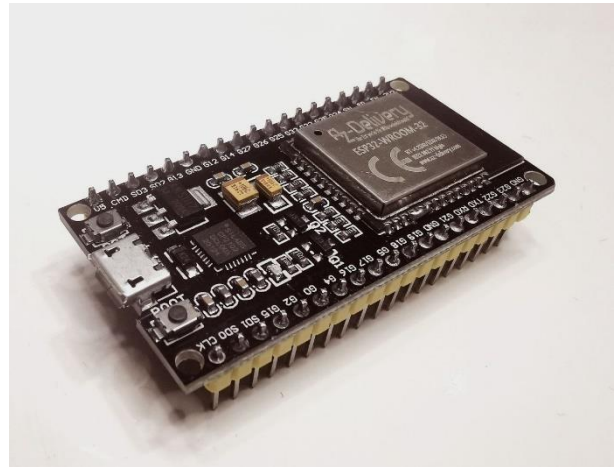


Ilustración 4. Microcontrolador ESP32

este microcontrolador en un candidato ideal para soluciones IoT. Al margen de estas características principales ya mencionadas, los microcontroladores ESP32 cumplen de forma satisfactoria con la mayoría de las características deseables en un controlador de pequeño formato. Estas son:

- Procesador Xtensa de 32 bits con doble núcleo hasta 240 MHz.
- Coprocesador de baja energía.
- 38 o 30 pines (dependiendo de su variante de encapsulado), entre los que encontramos: soporte para los protocolos de comunicación hardware más comunes, como SPI, I2C, I2S, UART o CAN Bus, además de multitud de pines GPIO de propósito general.
- Soporta las exigencias de seguridad estándar recogidas por IEEE, como arranque seguro, cifrado flash, procesador criptográfico compatible con AES, SHA-2 y RSA entre otros.
- Conversores analógico-digital (ADC) de 12 bits de resolución y digital-analógico (DAC) de 8 bits.
- Programable en múltiples lenguajes, destacando: C, micropython y LUA.
- Compatible con las de bibliotecas de Arduino.
- Gran comunidad de desarrollo.

2.3.1. Protocolos y conceptos hardware

Este apartado describirá los protocolos utilizados por los sensores usados en el sistema.

Este apartado describirá los protocolos utilizados por los sensores usados en el sistema.

2.3.1.1. 1Wire

El protocolo de comunicación desarrollado por Dallas Semiconductor basado en la transmisión de datos a través de una sola línea de comunicación, lo que da nombre al protocolo.

La implementación de este protocolo, por línea general, se apoya en el uso de una resistencia de pull-up encargada de mantener el voltaje alto. El protocolo permite trabajar desde 3.3V hasta 5V y cuenta con un funcionamiento maestro esclavo. El esclavo utilizará los flancos de bajada para sincronizar su reloj interno con el del maestro.

Este protocolo tiene un funcionamiento de maestro-esclavo. El maestro enviará datos a los esclavos mediante pulsos de voltaje durante un intervalo de tiempo determinado, dando lugar al envío de 0 o de 1. Todos los dispositivos capaces de trabajar con 1Wire deben contar reloj interno, este se sincronizará con el del maestro mediante flanco de bajada.

El voltaje del bus se mantendrá en alto, y solo se utilizarán pulsos de estado bajo para la transmisión de datos. La transmisión en el bus será iniciada por el maestro, que bajará el voltaje a 0 durante 480 μ s, los esclavos responderán con otro pulso bajo, esta vez de 60 μ s. A continuación, el maestro comenzará la transmisión, con un pulso bajo de 15 μ s para representar un 1 y un pulso bajo de 60 μ s para representar un 0. 1Wire cuenta con un sistema CRC para detección de errores.

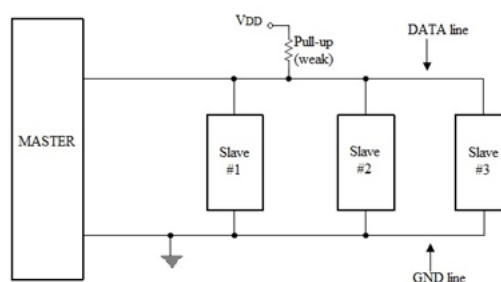


Ilustración 5. Conexión protocolo 1Wire

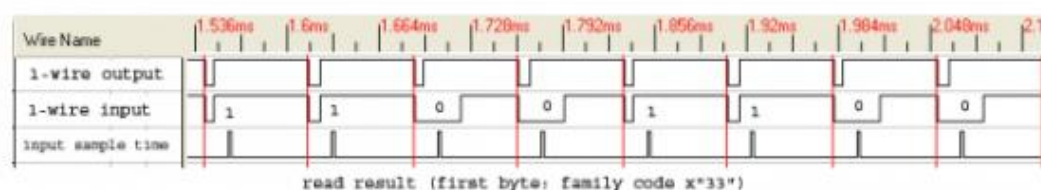


Ilustración 6. Bus protocolo 1Wire

2.3.1.2. Protocolo I2C

Protocolo de comunicación hardware muy común en periféricos de baja velocidad desarrollado por Philips Semiconductor. Tal y como indican los autores (Mankar, J., Darode, C., Trivedi, K., Kanoje, M., & Shahare, P. 2014) se trata de un protocolo síncrono basado en un funcionamiento multi-maestro – multi-esclavo que permite un número de dispositivos por bus que vendrá determinado por el espacio de direcciones de 7 bits.

El protocolo utiliza dos líneas para alimentación, permitiendo diferentes variantes entre 3.3v y 5V y otras dos líneas para la transmisión bidireccional de datos. La tasa de transmisión de I2C variará entre 10Kbits/s y 400Kbits/s dependiendo de la implementación de este.

Cualquier nodo del bus podrá adoptar el rol de maestro o de esclavo, recayendo sobre el maestro la responsabilidad de transmitir la señal de reloj utilizada en la sincronización. El maestro, al iniciar la transmisión, enviará en primer lugar la dirección del esclavo con el que quiere comunicar, una vez recibido el mensaje ACK del esclavo, comenzará la transmisión, manteniendo en alto el voltaje de la línea SDA para la transmisión de un 1 y bajo para la transmisión de un 0.

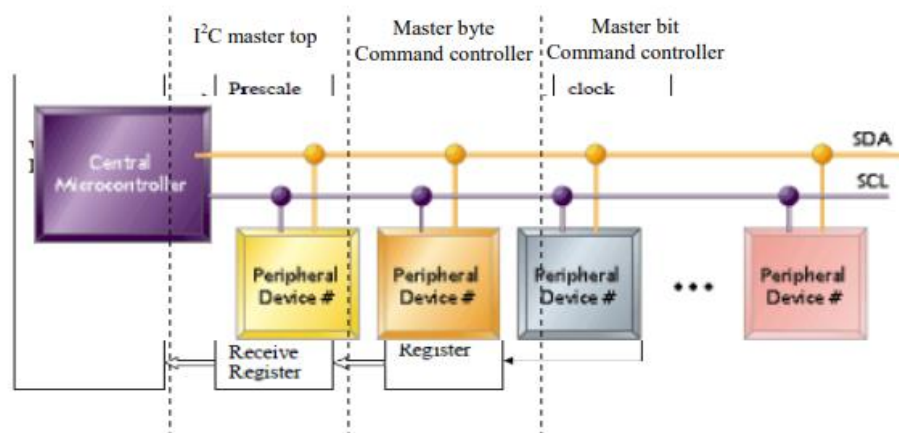


Ilustración 7. Componentes en una comunicación I2C. Imagen tomada de la publicación de (Mankar, J., Darode, C., Trivedi, K., Kanoje, M., & Shahare, P. 2014)

2.3.1.3. Resistencia de pull-up

Resistencia utilizada para mantener el voltaje en alto en una línea de un circuito, con esta técnica se consigue mantener un voltaje fijo en una línea de transmisión de datos cuando la transmisión no está activa. La configuración más habitual es utilizar una resistencia de 10KΩ.

El gráfico muestra un esquema de una resistencia de pull-up:

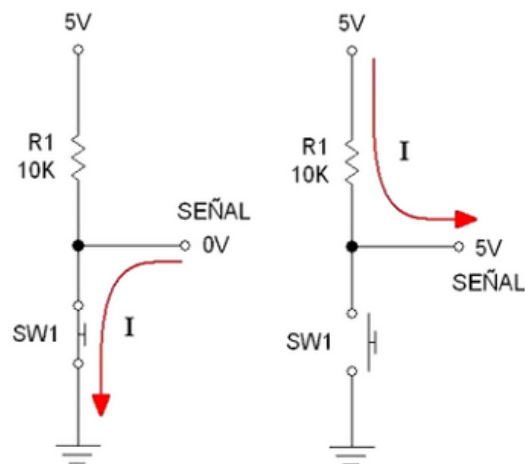


Ilustración 8. Esquema resistencia de pull-up

2.3.1.4. Divisor de voltaje

Un divisor de voltaje es una técnica, que, aplicada sobre un circuito eléctrico, permite repartir el voltaje entre diferentes puntos, dando como resultado un voltaje menor, en el punto deseado.

Este circuito, contará con una entrada V_{in} , a la que queremos disminuir su voltaje en la salida V_{out} . Esto lo seguimos con la siguiente configuración. Podemos calcular el voltaje que obtendremos a la salida con la fórmula que se escribe a continuación.

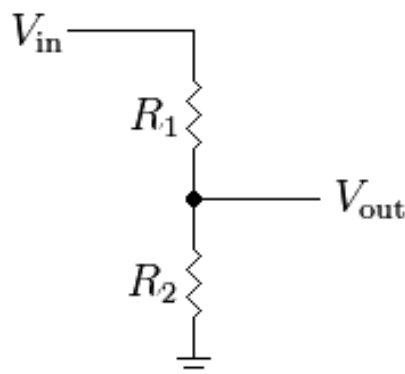


Ilustración 9. Esquema de conexión divisor de tensión

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

Ilustración 10. Fórmula cálculo divisor de tensión

2.3.1.5. Grado de resistencia IP

El término IPXX hace referencia al grado de protección frente a polvo y líquidos. Se trata de un estándar internacional establecido en la norma UNE-EN 60529:2018/A2:2018/AC:2019-02 (Asociación Española de Normalización. 2019) que utiliza la nomenclatura “IP”, seguida de dos dígitos, que hacen referencia al grado de resistencia frente al polvo y a líquidos. Siento el primer dígito el que representa la resistencia al polvo y el segundo el que representa el grado de impermeabilidad.

Los dispositivos dotados de la marca IP67 ofrecen garantía de protección contra polvo y frente a la inmersión en líquidos a profundidad menor a 1 metro durante un tiempo inferior a 30 minutos.

Los dispositivos con marca IP68 ofrecen garantías de protección contra polvo y frente a la inmersión continuada en líquidos.

2.3.1.6. Límite elástico

El concepto de límite elástico hace referencia a la tensión máxima que puede soportar un material sin sufrir deformaciones irreparables. Encontramos una descripción mucho más detallada proporcionada por (Bueche & Hecht, 2005). No obstante, no será necesario conocer más detalles sobre el concepto para la comprensión de este documento.

2.3.1.7. Tensión de Von Mises (Ruiz, M. C. 2009)

Concepto que debe su nombre a su autor, Richard Edler von Mises. Hace referencia a la magnitud física que proporciona la medida de energía de distorsión. Permite determinar que un material sufrirá fallo elástico cuando la energía de distorsión aplicada sobre un material supera el límite elástico de este. Este proceso permite determinar si un material sufrirá rotura o deformación o se mantendrá intacto

2.4. Conclusiones sobre el estado del arte

Existen multitud de proyectos que pretenden dar solución a la monitorización y control de los parámetros fundamentales de un acuario. Muchos de estos proyectos proponen soluciones similares, en las que los datos son tomados por una Raspberry Pi o un Arduino y son llevados hasta un servidor, donde son leídos para ser mostrados por una plataforma web o móvil.

Estos proyectos, en su mayoría, son trabajos de fin de grado. Al igual que en mi caso, que como trabajo de fin de grado en el grado de Ingeniería informática propuse un sistema similar. Los proyectos consultados, como (de Rey García, 2017; Pérez Laguarda, 2017; Alonso Rollán, S, 2019) reafirman varias de las hipótesis aquí planteadas. En primer lugar, coinciden, en su gran mayoría, en cuales son los parámetros de calidad del agua más relevantes. Por otra parte, reafirman la idea de que es complicado encontrar soluciones comerciales satisfactorias. Es cierto que existen algunas soluciones comerciales, sin embargo, estas no son más que meros sistemas de medición colocados sobre el acuario, con una pantalla para la visualización de los datos actuales y no permiten el almacenamiento ni análisis de datos históricos. Mientras que otras soluciones más completas requieren de una instalación compleja y no son asequibles.

Este proyecto, a diferencia de todo lo anterior, propone implementar un sistema de filtrado que incorpore el sistema IoT necesario para la monitorización y el control propuestos, dando lugar a un sistema de filtrado IoT. Este hecho representa la innovación de este proyecto.

La consecución de este proyecto daría lugar a un prototipo de filtro IoT, que, en líneas de trabajo futuras, podría ser llevado a un producto comercializable, mediante el cual el usuario tendría acceso a un dispositivo IoT que admite sensorización, sistema de alertas y recomendación dentro de un elemento, el filtro, un elemento ya existente en todo acuario doméstico.

3. Descripción general de la contribución del TFM

3.1. Objetivos

Objetivo general

Desarrollar un prototipo funcional de un sistema de filtrado IoT para acuarios domésticos, evolucionando un dispositivo de filtración tradicional para convertirlo en un prototipo de dispositivo de filtración propio de la industria 4.0.

Objetivos específicos

Para lograr el objetivo principal de este proyecto, será necesario alcanzar los objetivos específicos en los que se descompone el mismo. Estos objetivos serán los que marquen el desarrollo del sistema, acometiéndolos y gestionándolos en el tiempo de forma individual.

- Desarrollar el hardware: Desarrollar el hardware existente para dar lugar a un circuito impreso que reduzca el tamaño del prototipo, evite cableados exteriores y facilite la conexión con los sensores.
- Desarrollar y fabricar un modelo 3D: Desarrollar un modelo 3D del dispositivo de filtrado externo que permita la fabricación de este y la integración del hardware.
- Mejorar la experiencia de usuario: mejorar el apartado visual de la plataforma de usuario e implementar un sistema de alertas sobre los parámetros monitorizados incluyendo la necesidad próxima de limpieza del filtro.
- Implementar un sistema de configuración del dispositivo: Implementar un prototipo de dispositivo de fácil instalación y puesta en marcha para una persona sin conocimientos avanzados de informática.
- Mejorar la escalabilidad: Implementar mejoras en la arquitectura que permitan mejorar la escalabilidad del sistema IoT.

3.2. Metodología del trabajo

La metodología escogida para el desarrollo de este proyecto es la metodología SCRUM. El motivo principal de seleccionar esta metodología es que se trata de una metodología ágil. Será necesario un desarrollo ágil para cumplir con los plazos de entrega previstos en este trabajo de fin de estudios. El hecho de seleccionar SCRUM frente a otras metodologías ágiles se debe a la familiaridad del estudiante con esta metodología.

Se ha planteado un TFM abierto en algunos aspectos, con la posibilidad de seleccionar entre diferentes tecnologías. La metodología SCRUM permite la rápida adaptabilidad frente a los cambios que pudieran surgir durante el desarrollo.

Las principales características de la metodología SCRUM son:

- Planificación por bloques temporales: La planificación se realiza mediante bloques de tiempo, llamados “sprints”, estos tendrán una duración corta, de no más de dos semanas.
- Desarrollo incremental: El trabajo se divide en pequeñas iteraciones o sprints. La consecución de cada sprint dará lugar a una nueva funcionalidad del sistema. Originando una versión del sistema más completa y potencialmente entregable.
- Compatible con equipos de trabajo reducidos: Característica ideal, teniendo en cuenta que el desarrollo lo realizará una sola persona.
- Adaptación a cambios: La subdivisión de tiempo en sprints de desarrollo admite una rápida respuesta frente a cambios en los planteamientos o requisitos del sistema.

3.2.1. Distribución de roles

Scrum establece los siguientes roles, que serán repartidos entre el alumno y el tutor como se especifica a continuación:

- **Product Owner:** Su misión es conocer los requisitos y necesidades del proyecto y asegurar que el desarrollo se adecue a las perspectivas de negocio. El rol de product owner recaerá sobre el alumno, ya que el proyecto ha sido propuesto por mi mismo, y conozco las características y funcionalidad que requiere la implementación.
- **Scrum Master:** Asume la responsabilidad de asegurar el desarrollo, validando el cumplimiento de los objetivos y eliminando impedimentos con sus conocimientos sobre el dominio del problema. El rol en cuestión será asignado al tutor, encargado de validar la ejecución de los sprints, indicando posibles mejoras. Asimismo, será la figura sobre la que apoyarse en caso de encontrar alguna dificultad.
- **Equipo Scrum:** Equipo reducido encargado de la implementación y desarrollo del proyecto. El trabajo de fin de estudios es un caso excepcional en el que el equipo de desarrollo estará formado únicamente por el alumno.

3.2.2. Implementación de la metodología

Este apartado describirá la forma en la que se aplicarán las especificaciones de la metodología seleccionada para el desarrollo del trabajo de fin de estudios.

3.2.2.1. Documentos:

Scrum establece la creación de los siguientes documentos.

- **Product Backlog:** Documento de texto que describe a alto nivel la funcionalidad que debe cubrir el proyecto. Este documento viene dado por la definición de objetivos realizada en el formulario del trabajo de fin de master, así como en la descripción de requisitos de este mismo documento. No obstante, estos objetivos estarán incluidos, a alto nivel, en la herramienta de gestión del proyecto.
- **Sprint Backlog:** Describe detalladamente los objetivos concretos que deben implementarse en el próximo sprint. La herramienta de gestión del proyecto facilitará enormemente esta tarea.
- **Burn down:** Gráfico que representa el estado del sprint. El eje X marca los días restantes hasta la conclusión del sprint, mientras que el eje Y marca el número de tareas restantes, contabilizadas en forma de horas de trabajo. Del mismo modo que ocurría con los documentos anteriores, la herramienta de gestión del proyecto ejecutará esta tarea.

3.2.2.2. Reuniones

La propia naturaleza de este trabajo implicará algunas variantes respecto a las establecidas por la propia metodología, que vendrán marcadas principalmente por el hecho de contar con un equipo de desarrollo de un solo miembro. No obstante, se definen las reuniones propuestas por Scrum.

- Sprint planning meeting: Reunión que da comienzo un nuevo sprint cuya finalidad es definir qué objetivos recogidos en el Product Backlog serán incluidos en el sprint actual. Se realiza un desglose de tareas y subtareas que serán necesarias para la implementación de los objetivos y se estima, en número de horas, la dificultad de cada una de estas. Este tipo de reuniones serán llevadas a cabo por el alumno a lo largo de los sucesivos sprints con la finalidad de definir el alcance de este.
- Daily Scrum: Reunión diaria de corta duración que permite mantener un seguimiento del proceso de desarrollo por parte del Scrum Master. Estas reuniones, por motivos obvios referentes a la casuística del proyecto, no aplicarán.
- Sprint Review: Revisión de resultados llevada a cabo al final del plazo de desarrollo de cada Sprint. Serán realizadas por el alumno para dar finalización al sprint y tratar de realizar un balance de objetivos cumplidos, replanificando aquellos que no hayan podido implementarse.
- Sprint Retrospective: Mantenida a continuación de la Sprint Review, esta reunión será efectuada entre el alumno y el tutor con la finalidad detectar fallos en la implementación llevada a cabo en el sprint y definir posibles mejoras.

3.2.2.3. Herramienta de gestión del proyecto.

La herramienta seleccionada para la gestión de este proyecto no es otra que “Trello”. Este apartado describirá el modo en el que se utilizará la herramienta para la aplicación de Scrum.

Trello es una popular herramienta de gestión de proyectos basada en la idea de tablero kanban. Este tipo de tableros utilizan la existencia de una serie de columnas personalizadas que contendrán tarjetas de tareas. Dichas tarjetas ocuparán una columna u otra dependiendo del estado de la implementación de las tareas referenciadas en ellas.

El tablero de desarrollo tendrá un aspecto similar a este:

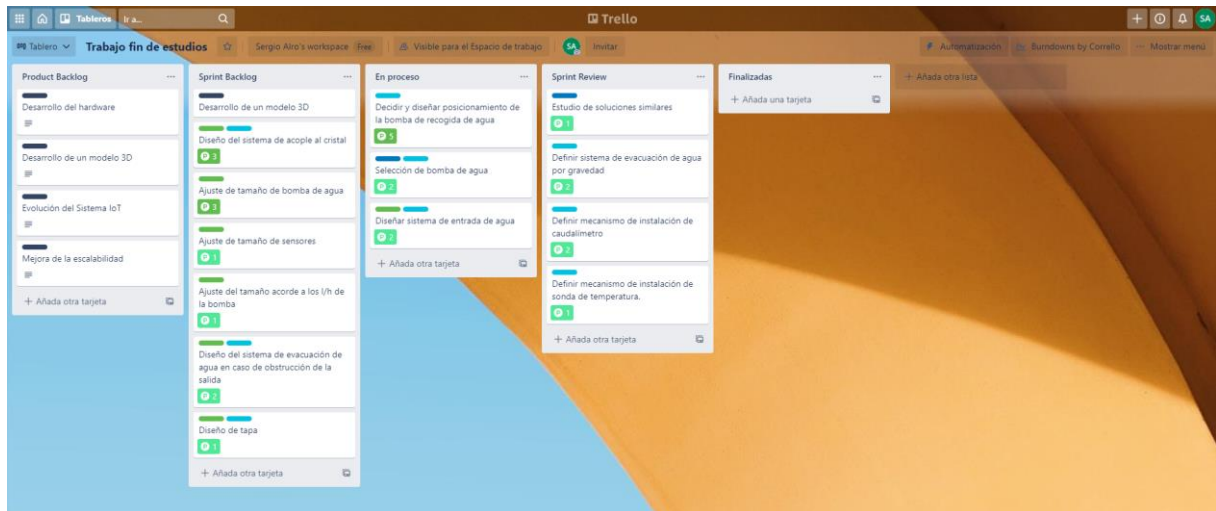


Ilustración 11. Vista general Trello

Para la implementación de la metodología Scrum se han fijado las siguientes columnas:

- Product Backlog: Se corresponde con el documento del mismo nombre que especifica a alto nivel los objetivos a cumplir.

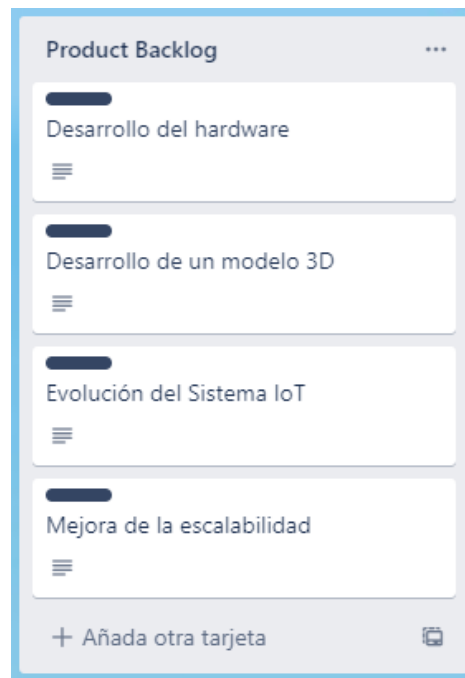


Ilustración 12. Ejemplo de Product Backlog

Cada una de las tarjetas contendrá una descripción más detallada del objetivo a cumplir:

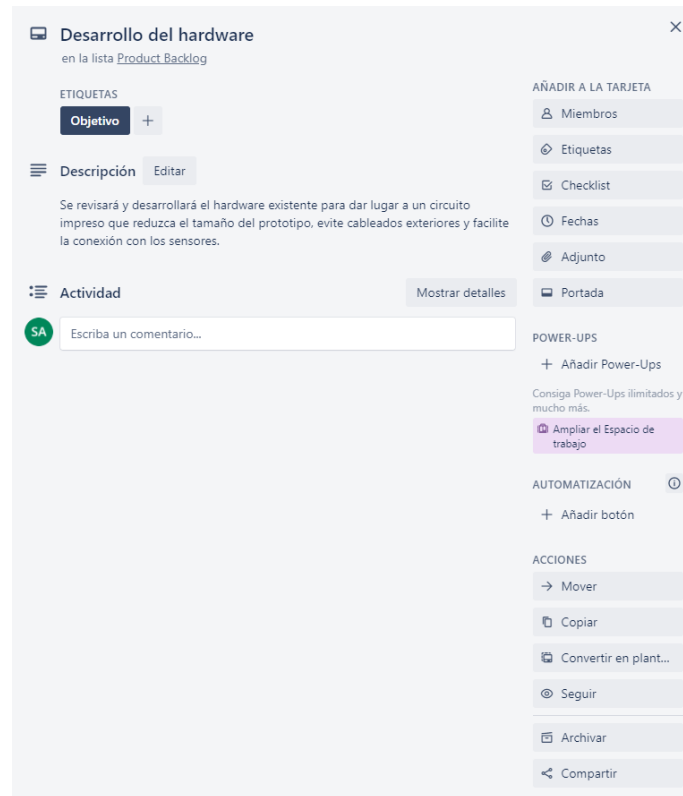


Ilustración 13. Ejemplo de tarjeta

- Sprint Backlog: Esta columna del tablero kanban se corresponde con el documento homónimo definido por la metodología. Esta columna estará vacía inicialmente, tomará forma en la reunión Sprint planning meeting en la que se definirán las diferentes tareas, representadas por tarjetas, que será necesario desarrollar para la consecución del objetivo tratado en el presente Sprint. Estas tareas recibirán una descripción y podrán contener una lista de subtareas. Durante el Sprint backlog, se realizará una estimación de esfuerzo, en forma de número de horas, necesario para la consecución de cada tarea.

Las tareas atendidas pasarán a la columna “en proceso” y, una vez completado su desarrollo, pasarán a la columna “Sprint Review” donde esperarán a la revisión llevada a cabo al final del sprint.

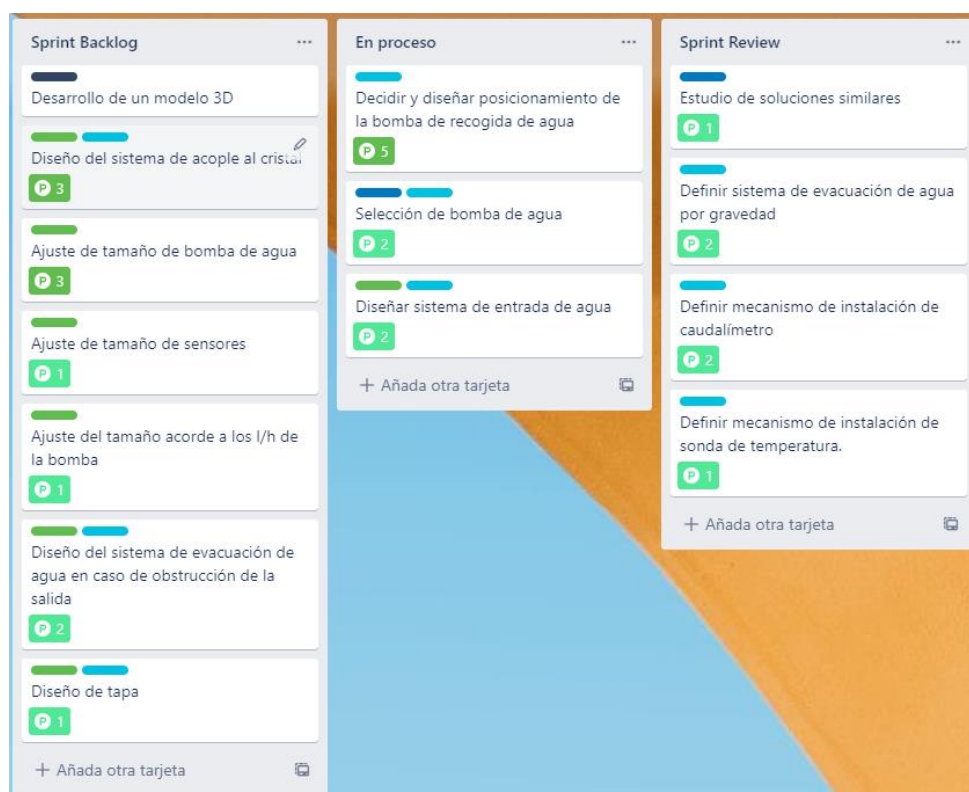


Ilustración 14. vista general Trello con columnas

Las tareas que pudieran no haber sido atendidas al final del plazo asignado para el sprint quedarán reprogramadas en el Sprint Backlog para el siguiente sprint.

Se crean además una serie de etiquetas que permitirán describir el tipo de tarea a la que hacen referencia. Las etiquetas quedan definidas como se muestra en la imagen.

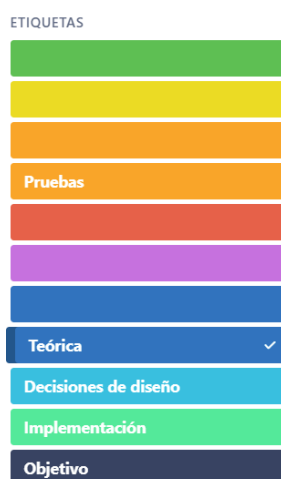


Ilustración 15. Etiquetas Trello

El gráfico Burndown se automatizará utilizando uno de los plugins ofrecidos por Trello, llamados power-ups. El power-up en cuestión recibe el nombre de “Burndown Charts by Corrello”

3.3. Descripción general de las partes o componentes de la propuesta

El proyecto existente, desarrollado como TFG, propone una solución IoT para la monitorización y control de los parámetros fundamentales para la vida en un acuario. Los parámetros monitorizados son: la temperatura, el caudal del filtro, para detectar el nivel de obstrucción de este y poder determinar el nivel de suciedad y la cantidad de luz natural recibida por las plantas del acuario. El prototipo desarrollado proporciona un sistema de control mediante el cual se adita o no luz artificial, en función de la luz natural recibida en el acuario. Los datos recogidos son enviados por el sistema IoT a una base de datos contenida en un servidor a través de una API, donde pueden ser visualizados mediante un panel dashboard. La plataforma desarrollada permite configurar alertas cuando alguno de los parámetros oscila fuera del rango establecido por el usuario como óptimo.

Para este proyecto se plantea transformar el sistema ya existente, mejorando apartados como el hardware, el sistema de alertas o la plataforma de visualización con el fin de convertirlo en un prototipo de sistema de filtrado IoT. El prototipo inicial o maqueta se transformará para dar lugar a un sistema de filtrado IoT capaz de realizar las funciones de monitorización y actuación sobre el acuario, además de establecer alertas y poder realizar recomendaciones personalizadas en base a las lecturas del acuario.

3.3.1. Alcance y limitaciones

El presente proyecto pretende mejorar el sistema de monitorización y control desarrollado en el trabajo de fin de grado con el fin de transformarla e integrarla en un sistema de filtrado, dando lugar a un sistema de filtrado IoT con su respectiva plataforma de visualización y su sistema de alertas personalizadas.

El prototipo no permitirá monitorizar pH, siendo este otro de los parámetros fundamentales del agua. Tampoco permitirá actuar sobre la temperatura o el caudal del filtro, siendo la luz el único punto de actuación.

3.3.2. Tecnologías implicadas

En cuanto a las tecnologías habilitadoras de la industria 4.0 aplicadas en el desarrollo de este proyecto, se propone:

- IoT para el desarrollo de un objeto avanzado, propio de la industria 4.0. Un objeto capaz de comunicarte e interactuar con su entorno, así como de proporcionar las mediciones pertinentes a través de dashboard de visualización.
- Modelado 3D: Para realizar un renderizado de la placa PCB así como para el diseño del propio sistema de filtrado. Valorando todos los posibles aspectos del diseño y permitiendo visualizar el objeto antes de fabricarlo, con el fin de detectar posibles fallos en etapas tempranas del proyecto.
- Fabricación inteligente, será el proceso que permitirá la creación del objeto definido a través del modelado 3D.
- Se aplicarán conocimientos de “innovación tecnológica y transformación digital” como marco de trabajo para realizar la transformación de la maqueta en un prototipo funcional.
- Big data, implementado a través de las herramientas ofrecidas por la aplicación IoT, con el fin permitir la realización de series temporales que permitan predecir un valor a través del tiempo.

3.3.3. Arquitectura, componentes e integración de tecnologías.

El sistema contará con el prototipo desarrollado, que será un sistema de filtrado IoT. Este se encargará de realizar las mediciones necesarias, los datos serán enviados a un servidor central mediante la red wifi doméstica. El mecanismo o protocolo mediante el cual se enviarán al servidor dependerá de la fuente de datos seleccionada, se estudiarán diferentes alternativas como la REST API ya existente o el uso de un gestor de colas.

Los datos serán almacenados en el servidor, en una base de datos. Serán mostrados mediante una plataforma de visualización y se les aplicarán técnicas de estudio de datos con el fin realizar recomendaciones de limpieza de filtro personalizadas, y decidir cuando es necesario encender o apagar la luz.

El esquema muestra los módulos principales en los que se compone la aplicación:

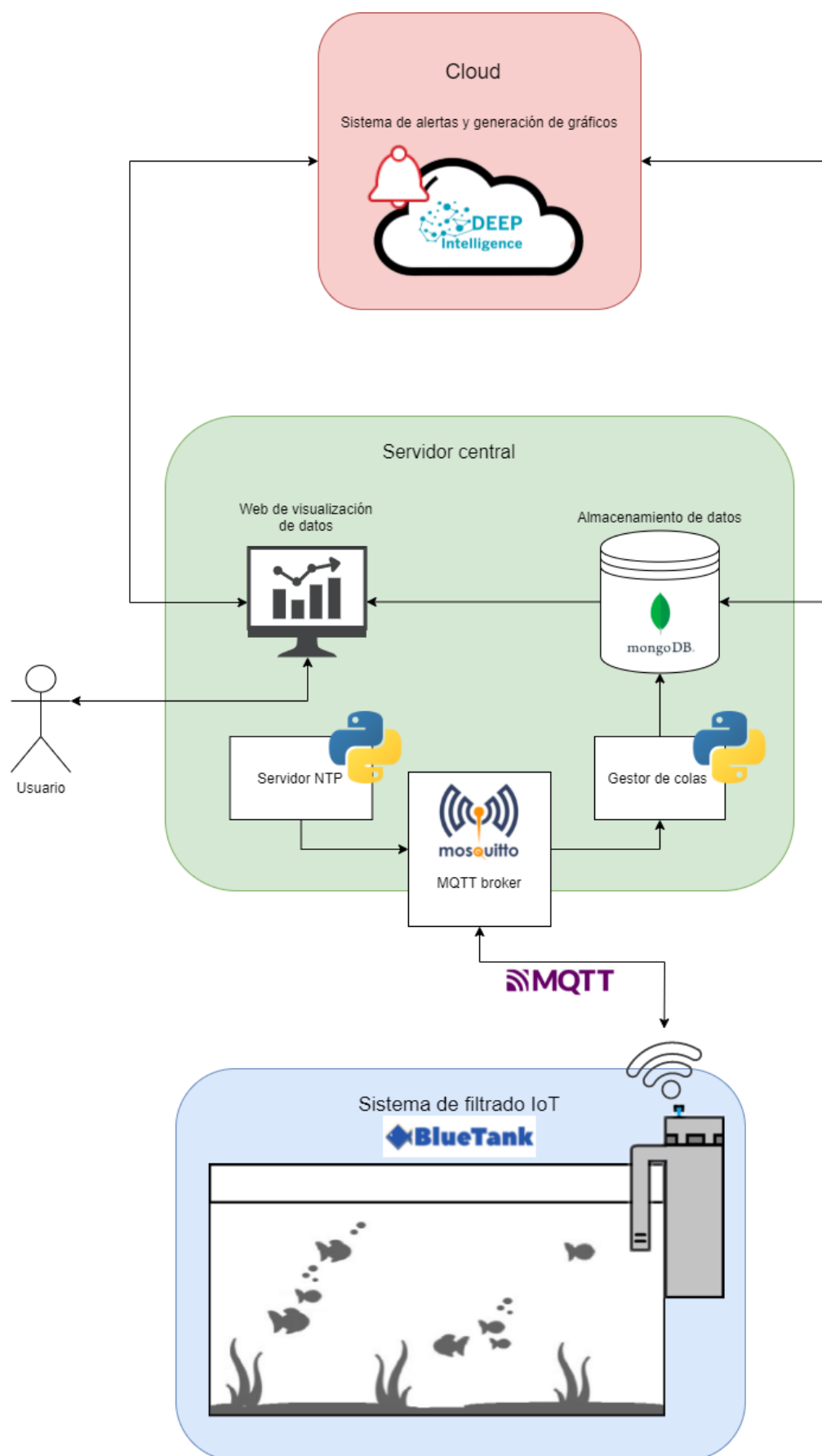


Ilustración 16. Arquitectura del sistema

3.3.4. Resultados esperados

Con la consecución de este proyecto espero obtener, a nivel de negocio, un prototipo funcional de un sistema de filtrado IoT.

A nivel técnico, este desarrollo pretende demostrar las habilidades y conocimientos obtenidos en el máster de industria 4.0, principalmente en los ámbitos de sensorización, IoT, Big data y fabricación inteligente. En cuanto al nivel organizativo, se pretende demostrar, por parte del alumno, la capacidad de gestión de un proyecto real.

3.3.5. Presupuesto y retorno esperado de la inversión

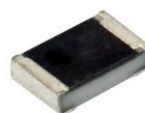
Se presenta el siguiente presupuesto, que se subdivide en tres apartados diferenciados. El presupuesto de coste de fabricación de la electrónica, el presupuesto de horas de ingeniería y montaje y el presupuesto de fabricación de la pieza.

El presupuesto se ha calculado para la fabricación de 1000 unidades del producto.

3.3.5.1. Presupuesto de fabricación de electrónica

Tabla 1. Presupuesto de componentes

Imagen	Componente	Coste Ud	Uds	Total dispositivo	Referencia
	ESP32	5,99€	1	5,99€	URL
	módulo Step-Down	0,889€	1	0,889€	URL
	módulo relé	1,06€	1	1,06€	URL
	sensor TSL2561 (luxes)	3,03€	1	3,03€	URL
	sensor DS18B20 1m (temperatura)	1,08€	1	1,08€	URL

	sensor YF-S201 (caudal)	2,30€	1	2,30€	URL
	Resistencia 4,7 kΩ	0,0167€	2	0,0334€	URL
	Resistencia 10 kΩ	0,0168€	1	0,0168€	URL
	Conectores PHOENIX 3 pines	0,446€	4	1,784€	URL
	Placas diseñadas	0,129€	1	0,129€	URL
	Caja IP68	10,16€	1	10,16€	URL
	Conectores IP68 PG9	0,164€	5	0,82€	URL
	Bomba agua	2,98€	1	2,98€	URL
	Fuente alimentación 12V (EU Plug)	3,91€	1	3,91€	URL
TOTAL por unidad fabricada				33,53 € / unidad	
TOTAL 1000 uds				33 526,2 €	

3.3.5.2. Presupuesto ingeniería y montaje

Tabla 2. Presupuesto costes ingeniería

Descripción	Coste por unidad	Cantidad total	Coste total
Horas ingeniería	38€/hora	960	36480€
Alquiler oficina 50m2	500€/mes	6 meses	3000€
Conexión internet	45€/mes	6 meses	270€
Equipos informáticos	500 unidad	2	1000€
TOTAL por unidad		40,750 €/unidad	
TOTAL 1000 uds		40 750 €	

3.3.5.3. Presupuesto de fabricación de pieza

Para la comprensión de este apartado se hace necesario matizar que la fabricación del prototipo se ha realizado con tecnología FDM, más accesible para el alumno. No obstante, no es la tecnología de fabricación más adecuada en el supuesto de llevar este proyecto a un producto comercial. Este presupuesto se ha realizado para una fabricación de pieza por inyección.

Tabla 3. Presupuesto molde de fabricación de pieza

Descripción	Coste Total	Coste por pieza (1000 uds)
Molde de inyección de pieza	8000€	8€

Tabla 4. Presupuesto molde de fabricación de pieza por inyección

Descripción	Coste Ud	Cantidad por dispositivo	Total por dispositivo
Fabricado pieza	0,78 €	1	0,78€
Total por unidad		0,78€	

Total 1000 uds	780€
----------------	-------------

3.3.5.4. Costes totales

Tabla 5. Costes totales de fabricación

Costes totales del proyecto	83 056,2 €
Coste por unidad (1000 uds)	83,06 €

3.3.5.5. Costes fijos mensuales

Estos costes se emitirán mensualmente a lo largo del ciclo de vida del proyecto, invariablemente del éxito de ventas.

Tabla 6. Costes fijos mensuales

Descripción	Coste por unidad (1000 Uds)	Coste total
Coste mensual de alquiler del servidor	0,075€	75€
Coste de la plataforma Deep Intelligence	0,6€	600€

3.3.5.6. Retorno de la inversión

Se plantea la posibilidad de vender el filtro a 65€, compitiendo en precios con los filtros de cascada convencionales que rondan los 55€ o 60€. A pesar de estar vendiendo por debajo de los costes de fabricación, será obligatorio para el usuario pagar la suscripción anual a la plataforma, por la que se cobrará 4,99€ / mes siendo la suscripción mínima de 6 meses.

Una unidad vendida a 65€ más su correspondiente suscripción 29,94€ (suscripción mínima) hacen un total de 94,94 € quedando amortizado el producto.

Si se venden todas las unidades obteniendo 94,94€ por unidad, estimando que se venderán 250 unidades/año, tenemos el siguiente retorno de inversión:

$$\text{Beneficio anual} = (94.94 * 250) - \text{Gastos anuales} = (94.94 * 250) - 8100 = 15635€$$

$$RI = \frac{83056}{15635} = 5,3 \text{ años de amortización}$$

3.3.6. Planificación general

El desarrollo del proyecto contará con diversas fases, repartidas a lo largo del tiempo.

- Fase de búsqueda de información: Esta fase pretende recopilar toda la información necesaria respecto a bibliografía, estado del arte, tecnologías y plataformas con el fin de facilitar la tarea de la toma de decisiones en cuanto a la selección de tecnologías, protocolos, hardware y plataformas en la implementación posterior. Esta fase tendrá una duración total de veinte días.
- Fase de planificación: Esta fase pretenderá planificar establecer los tiempos necesarios para la consecución de las fases posteriores, dando lugar a diversos gráficos o diagramas que describirán la evolución ideal del desarrollo a lo largo del tiempo. Se dedicarán cinco días completa para la consecución de esta fase.
- Fase de definición: Se describirán los objetivos del proyecto y este se definirá de forma detallada. Realizando el pertinente análisis de requisitos, el diseño de la arquitectura del sistema y la toma de decisiones en cuanto a selección de tecnologías y plataformas, determinando que elementos son aprovechables del trabajo anterior y que elementos deben ser mejorados o sustituidos. Esta fase será la base sobre la que asentar la fase de desarrollo. La realización de esta fase dará lugar a la documentación necesaria para describir el sistema a implementar. La fase de definición sentará los pilares sobre los que apoyar el proyecto, errores en esta fase podrían llegar a ser críticos, por ese motivo se dedica un total de entre quince y veinte días para su realización.
- Fase de desarrollo: Realizada bajo la metodología SCRUM, pretende dar forma al prototipo, partiendo del diseño realizado y llegando a la construcción de este. El desarrollo cuenta con cuatro bloques muy marcados, que se implementarán de forma secuencial, estos son: desarrollo del hardware, desarrollo del modelo 3D, mejora de la experiencia de usuario, desarrollo de un sistema de instalación y configuración del sistema. Cada bloque se corresponde con cada uno de los objetivos específicos, a excepción de la mejora en la escalabilidad, que viene dado por la arquitectura y protocolos seleccionados. Al término de esta fase se obtendrá un prototipo funcional. Se estima una duración de unos sesenta días para lograr estos objetivos.
- Fase de validación: Se realizarán las pruebas pertinentes con el fin de validar el correcto funcionamiento del sistema. La fase de validación se intercalará al final del desarrollo de cada uno de los bloques a desarrollar, con el fin de identificar posibles errores de forma temprana, teniendo así margen de mejora. Se dedicarán

aproximadamente veinticinco días, repartidos al final de cada fase. Esta fase incluirá pruebas de todo tipo, desde probar y validar el funcionamiento de diferentes plataformas IoT, protocolos de comunicación, gestores de bases de datos, etc.. con el fin de seleccionar el mejor candidato en cada caso, hasta realizar test de funcionamiento del desarrollo implementado.

Al final del desarrollo, una vez obtenido el prototipo, se validará el correcto funcionamiento durante varias semanas más, solapándose estas con la fase final de documentación.

- Fase de documentación: Se documentará el desarrollo realizado, incluyendo la documentación pertinente respecto al hardware, software y componentes utilizados. La consecución de esta fase dará lugar a la memoria terminada. La fase de documentación se realizará de forma distribuida, dedicando tiempo al final de cada una de las fases anteriores para documentar la misma. Se estima una duración de entre veinte y veinticinco días para la realización de la documentación, que serán distribuidos al final de cada fase de forma proporcional a la dificultad de esta.

El siguiente diagrama refleja la planificación temporal realizada de las fases anteriormente descritas:



Ilustración 17. Planificación temporal

4. Desarrollo específico de la contribución

Este apartado describirá con alto nivel de detalle el proceso que ha permitido la realización de este proyecto. Partiendo de los objetivos marcados a alto nivel hasta conseguir la implementación de estos.

El apartado está subdividido en diferentes bloques que se corresponderán con los sprints de desarrollo marcados por la metodología seleccionada. Cada sprint trata de acometer una funcionalidad concreta del sistema, siendo esta todo o parte de un objetivo específico descrito para este proyecto.

La duración de los sprints vendrá determinada por la dificultad y número de las tareas programadas para cada uno de ellos.

4.1. Objetivo específico: Desarrollar el hardware.

La consecución de este objetivo vendrá dada a partir de la implementación de dos sprints. Estos sprints no se realizan de forma consecutiva debido a los tiempos de fabricación del hardware diseñado.

4.1.1. Sprint 1: Diseño de la electrónica

El primer sprint implementado tratará de recoger los requisitos funcionales del hardware y de realizar un diseño de una placa PCB que cumpla con las funcionalidades establecidas.

El éxito de este sprint permitirá establecer una base sólida sobre la que apoyar el resto de los componentes del sistema. Definirá las funcionalidades admitidas por el dispositivo y establecerá las limitaciones del mismo. El éxito de este sprint será crucial en el desarrollo de todo el proyecto.

Al comienzo del sprint se describen, en el Sprint backlog, las diferentes tareas que será necesario enfrentar para la consecución del objetivo. El desglose de tareas permitirá la estimación y el cumplimiento de los plazos establecidos. El proceso da comienzo con el estudio de alternativas hardware, y las decisiones de selección de componentes.

El documento Sprint backlog quedará definido de la siguiente forma:



Ilustración 18. Documento Sprint Backlog del Sprint 1

Para este sprint se ha establecido una duración máxima de quince días.

El sprint dará comienzo realizando un estudio sobre las diferentes alternativas hardware disponibles en el mercado y realizando una búsqueda de los datasheets y la información relevante para su uso, proceso que culminará con la selección del hardware más adecuado que permita el cumplimiento de los objetivos establecidos.

La decisión de diseño más relevante viene determinada por la selección del tipo procesador a emplear, valorando entre el uso de una Raspberry Pi 4, como opción de ordenador de placa reducida, que cuenta con un microprocesador y ofrece un sistema operativo completo, y un ESP32, como opción de microcontrolador. Finalmente, habiendo verificado que todos los objetivos del proyecto pueden ser satisfechos con un microcontrolador, se selecciona el ESP32 como candidato, principalmente debido al precio, ya que permitirá abaratar el prototipo, sin olvidar su menor consumo energético y su menor tamaño.

Se propone el siguiente esquema para el desarrollo de la electrónica.

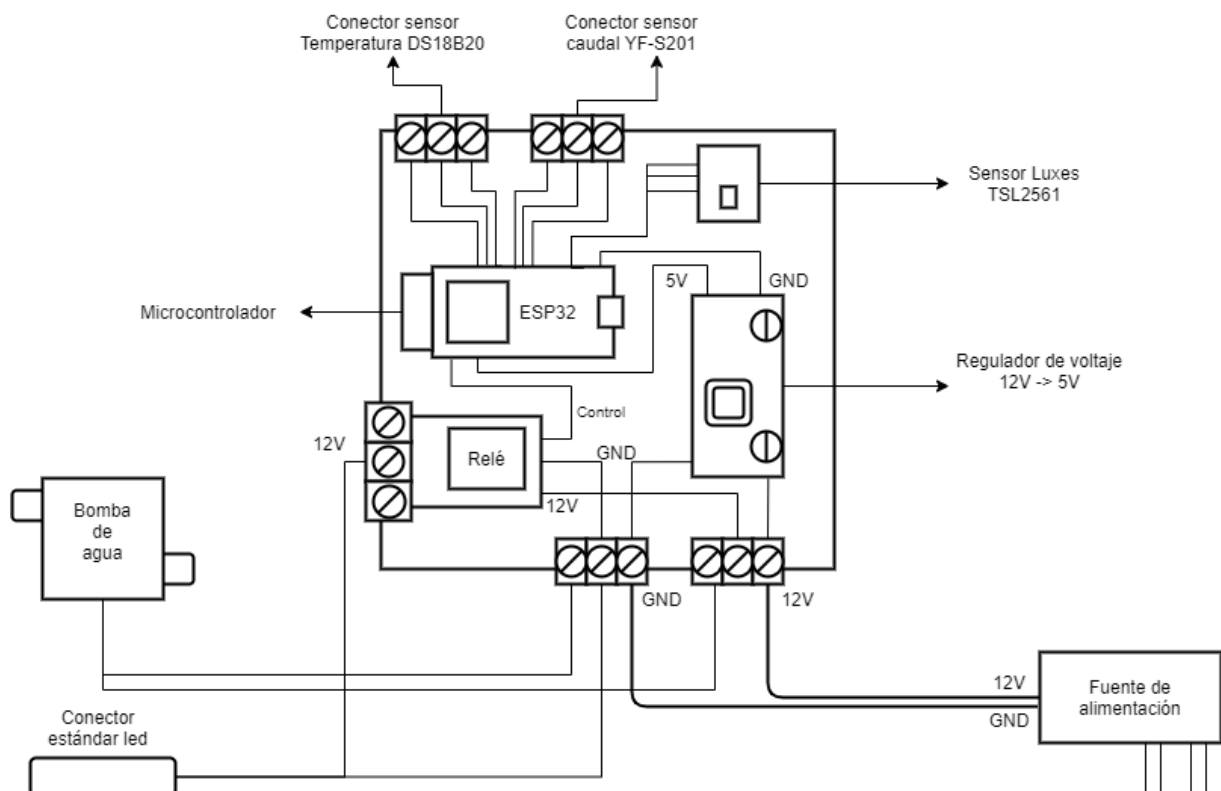


Ilustración 19. Esquema conceptual PCB electrónica

El esquema da comienzo con una entrada de alimentación a 12 voltios, a partir de la cual se deriva la alimentación de la bomba de agua y la alimentación de los leds que ya contiene el propio acuario. Pasando esta última a través de un relé que permitirá el encendido y apagados de luces.

A continuación, se presenta un regulador de voltaje, en el que este se reduce mediante el uso de un step-down. La salida del step-down, calibrada a 5 voltios, será la que alimenta al microcontrolador ESP32 a través de sus pines de alimentación, indicados en el datasheet adjunto a esta memoria.

Los sensores, serán alimentados a través de la alimentación interna, a 3.3 voltios, del propio ESP32, que cuenta con un regulador de voltaje interno.

Los sensores seleccionados los determinará la propia experiencia del alumno en cuanto a protocolos de comunicación, escogiendo la sonda DS18B20 con el protocolo 1Wire para la monitorización de temperatura, el sensor de luxes TSL2561 que implementa el protocolo I2C y el sensor YF-S201, de efecto hall, para las mediciones de caudal. La PCB diseñada contará con conectores de bloque de 3 terminales, que permitirán la conexión del sensor de temperatura y del sensor de caudal. El sensor de luxes quedará integrado en la propia placa.

Determinado el hardware necesario, se representa un boceto de conexiones de los sensores.

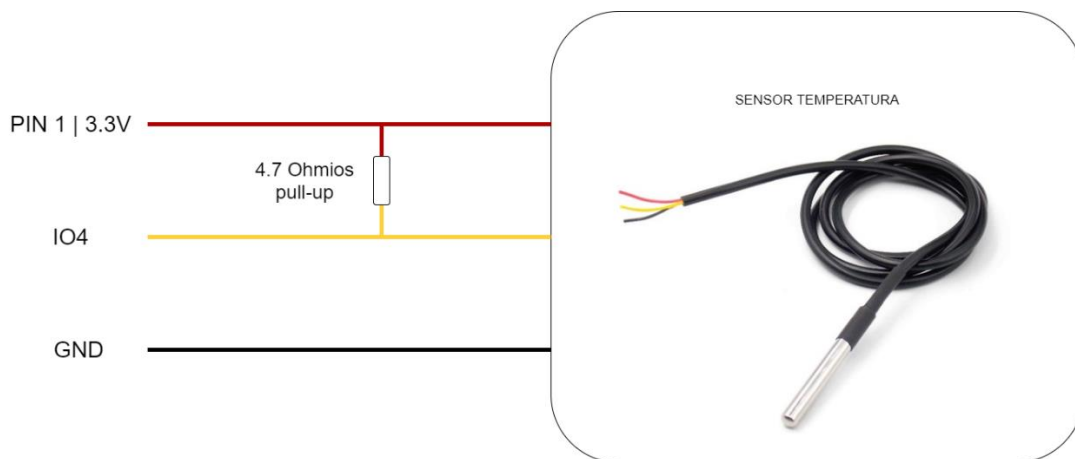


Ilustración 20. Conexión sensor temperatura DS18B20

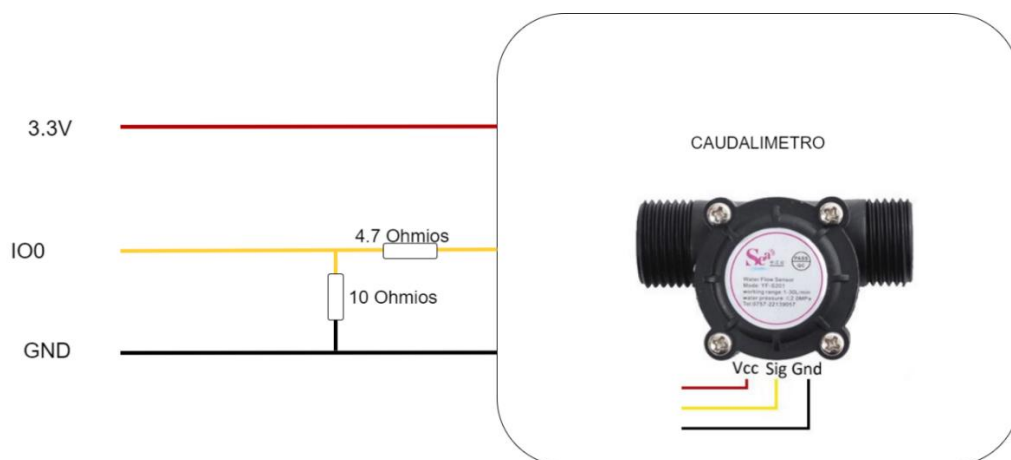


Ilustración 21. Conexión caudalímetro YF-S201

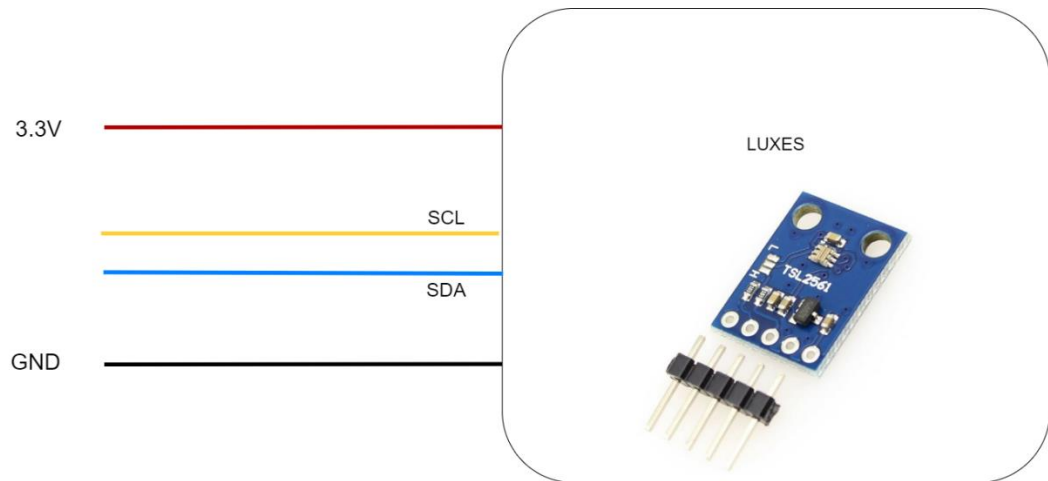


Ilustración 22. Conexión sensor lux TSL2561

Esta placa, está diseñada para estar colocada en una caja con protección IP67, certificación que encontraremos adjunta a esta memoria. La electrónica diseñada quedará en el interior de la placa a excepción de la bomba de agua, la sonda de temperatura y el caudalímetro, pensadas para trabajar de forma sumergida, los cables de alimentación, que finalizarán en un transformador para un enchufe doméstico y los cables de alimentación de los leds, que terminarán en un conector cilíndrico estándar compatible con una gran variedad de leds del mercado. Los elementos externos a la caja IP67 quedarán conectados a esta a través de un conector tipo prensaestopa IP68 PG9 del cual, una vez más, encontraremos la información anexada a este proyecto.

Definido y validado el hardware, la consecución de este sprint pasará por la implementación de la PCB en un programa de diseño electrónico, como es el caso de Altium, obteniendo el siguiente esquemático.

Implementado el esquemático, el siguiente paso consistirá en definir las rutas o pistas de la placa, siendo particularmente necesaria la mención del procedimiento que ha permitido definir el ancho de las pistas.

La placa contará con un plano de masa en la parte trasera al que conectarán todas las líneas “gnd”. Para calcular el ancho de las pistas restantes se utiliza la herramienta proporcionada por “Digikey” en su web oficial. Estableciendo 0.2 A, siendo este el consumo máximo de la placa de evaluación del ESP32 e indicando que no es deseable que la temperatura de la pista suba más de 5°C para una temperatura ambiente de 25°. El último parámetro, 1 oz/ft², vendrá dado por el propio fabricante de la placa. Obteniendo, que el ancho de pista de la placa será de 5,08mm

Corriente 0.2 A Espesor 1 oz/ft² Aumento de temperatura 5 °C Temperatura ambiente 25 °C Longitud de trazo Ingresar longitud... in	FÓRMULA La calculadora de ancho de traza de PCB utiliza fórmulas de IPC-2221. Ancho mínimo de traza 5.089299623 mil Capas internas Ancho de traza requerido = 5.089299623 mil a la corrosión = 0.000000000 Ω
---	--

Ilustración 23. Cálculo de ancho de pistas

Los resultados de este sprint son los presentados a continuación, dando como resultado los archivos gerber de cada capa y el archivo de troquelado para la CNC, que será enviado al fabricante para su producción.

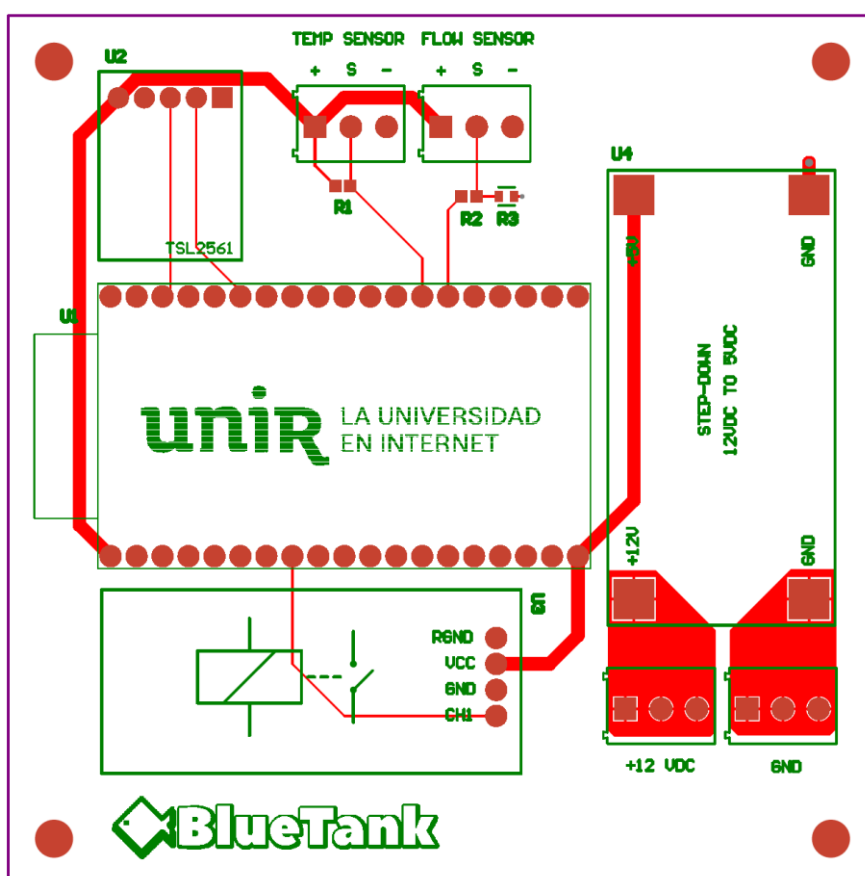


Ilustración 24. Placa diseñada con Altium

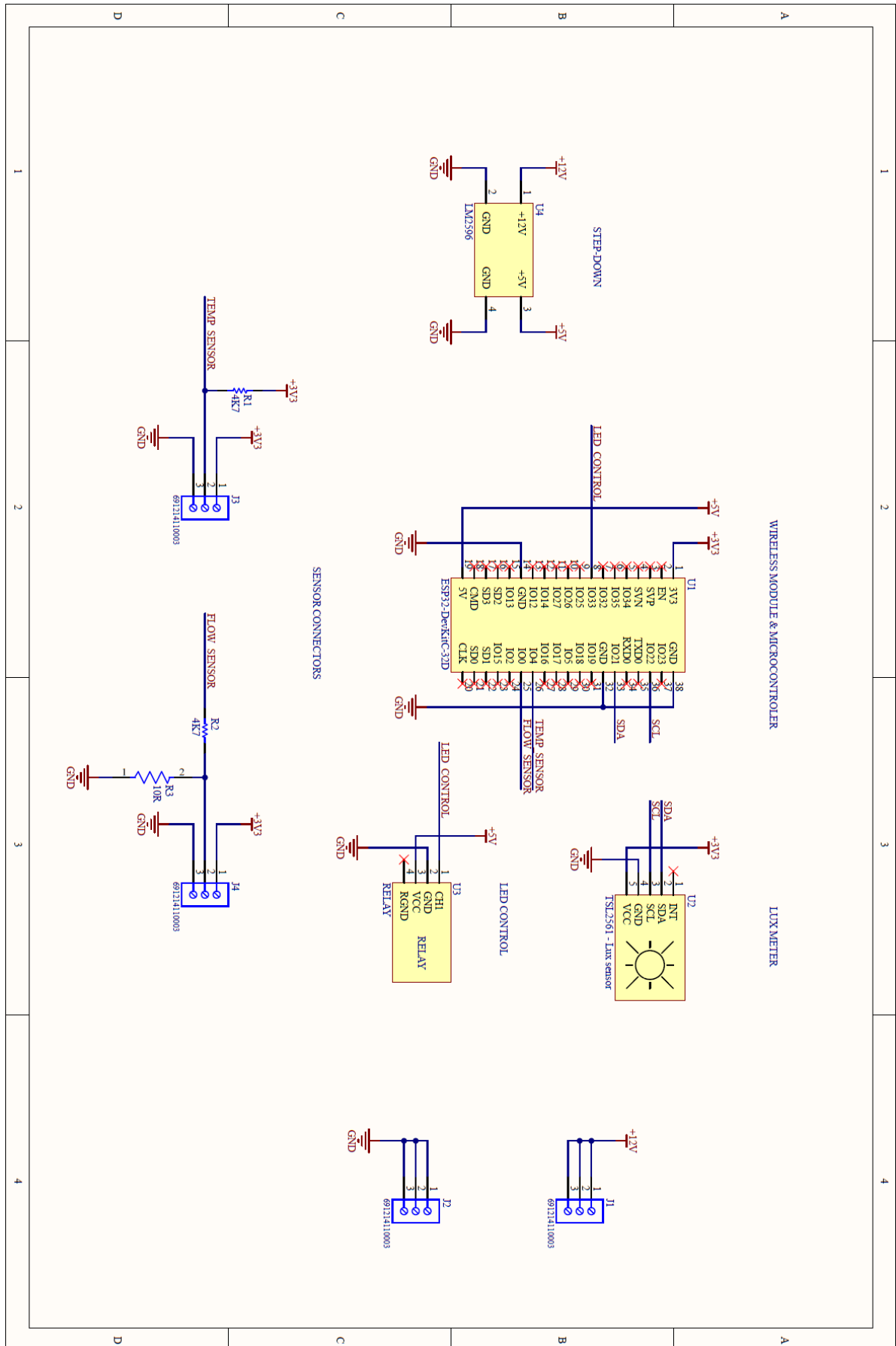


Ilustración 25. Esquemático del diseño de la PCB

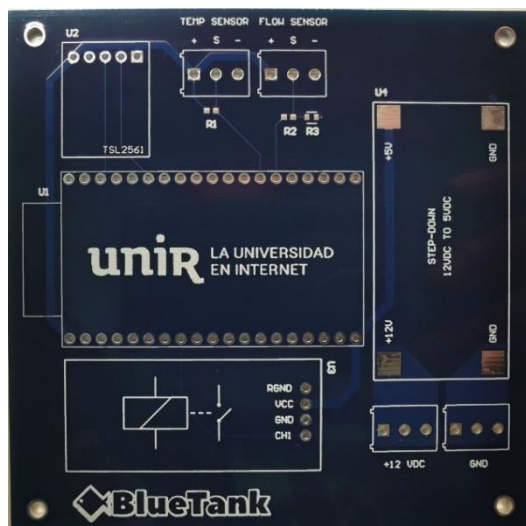


Ilustración 26. Vista delantera PCB

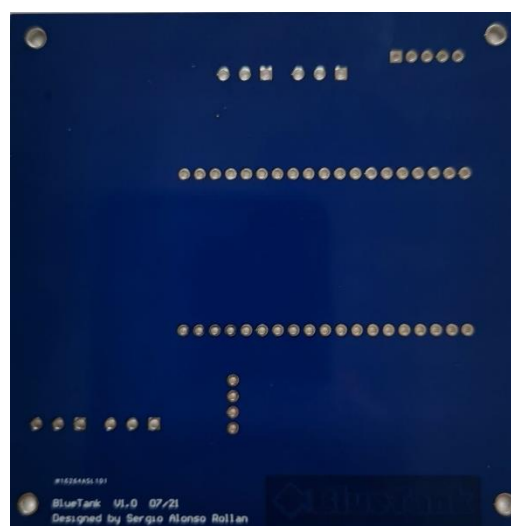


Ilustración 27. Vista trasera PCB

4.1.2. Sprint 4: Montaje y soldadura electrónica

Se trata de un sprint sencillo que tiene como objetivo el montaje de la placa con sus correspondientes soldaduras. El tiempo establecido para este sprint será de un día completo.

El sprint backlog queda definido como se indica en la imagen.

El sprint da comienzo con la placa recibida, en primer lugar, se soldarán las resistencias SMD con pasta de soldadura mediante un proceso de fundición por calor.

Una vez comprobado el éxito de la soldadura con el polímetro se procede a soldar con soldador de estaño el resto de los componentes sobre la placa. Merece la pena hacer especial mención al hecho de que los conectores se han soldado por debajo de la placa con el fin de ocultar el cableado y se han soldado unos pines en la base del ESP32 con el fin de que este pueda ser extraído con facilidad. Este sprint también comprende el proceso de integrar la electrónica en la caja IP67 adquirida, realizando la instalación de los conectores prensaestopas IP68. La caja tiene una tapa transparente para permitir el paso de la luz al sensor de luxes. La placa diseñada cuenta con cuatro



Ilustración 28. Documento Sprint Backlog del Sprint 4

bornes de conexión, estos se conectan, uno para el sensor de caudal, otro para el sensor de temperatura, uno para voltaje positivo y otro para voltaje negativo. La fuente de alimentación de 12V se conecta a los bornes positivo y negativo, además se conecta a esos mismos bornes la salida de alimentación de la bomba de agua y la salida de alimentación de los leds. Para los leds se utiliza un conector estándar que permitirá adaptarse a cualquier tira led

A continuación, se presentan algunas fotos del proceso realizado.

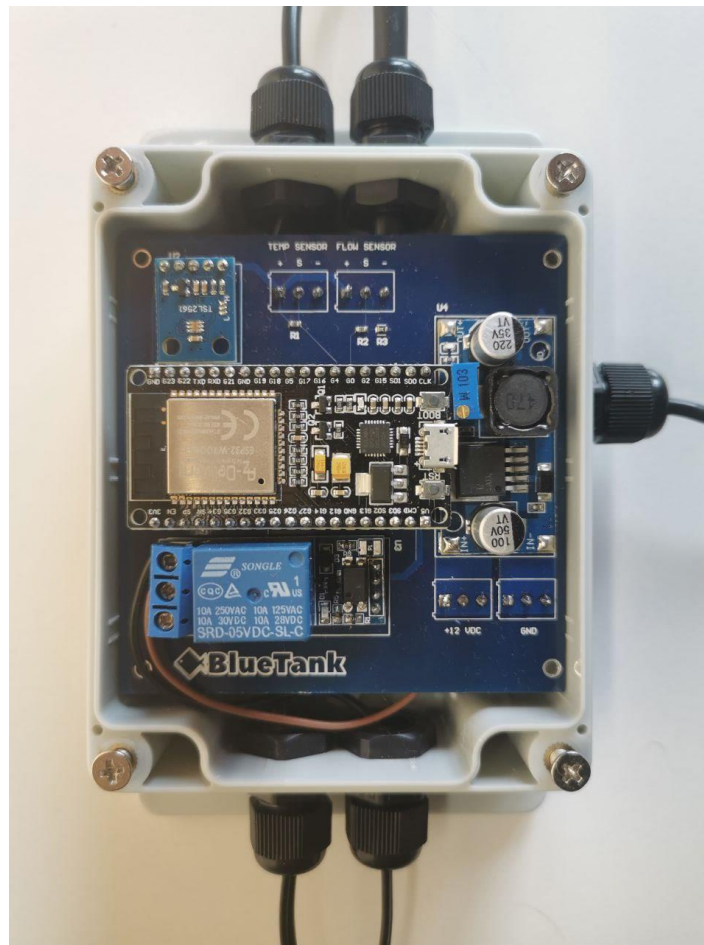


Ilustración 29. Placa soldada y montada en la caja con conectores prensaestopas

4.2. Objetivo específico: Desarrollar y fabricar un modelo 3D.

El presente apartado describe la implementación del objetivo específico referente al desarrollo y la fabricación del modelo 3D de la pieza que formará el propio filtro e integrará la caja que contiene la electrónica. El objetivo se implementa a través de dos sprints consecutivos. Para lograr el éxito los resultados aquí recogidos ha sido necesario realizar tres iteraciones en los sprints descritos a continuación. Cada iteración ha propuesto mejoras y se han pulido algunos

aspectos de diseño hasta obtener el resultado deseado. El tiempo total dedicado a este objetivo ha sido de treinta días.

4.2.1. Sprint 2: Diseño del modelo 3D del filtro

El sprint encargado del desarrollo del modelo 3D filtro queda definido por el siguiente documento sprint backlog. La duración establecida para este sprint es de cinco días, incluyendo el estudio de soluciones similares, la implementación del modelo 3D.

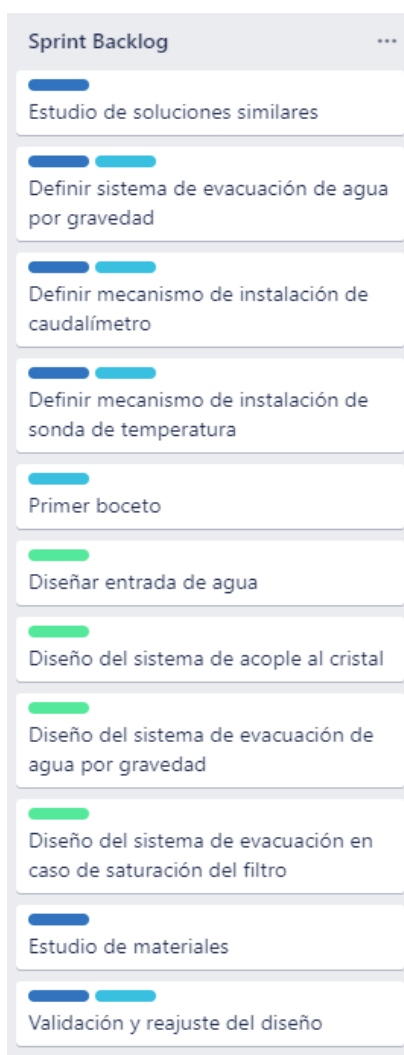


Ilustración 30. Documento Sprint Backlog del Sprint 2

El desarrollo del modelo 3D es el sprint más importante del desarrollo del presente proyecto. Se trata de un punto crítico, cuyo éxito o fracaso marcará el devenir del resto de la implementación.

El prototipo del filtro, una vez fabricado, será el que permita integrar el resto de los componentes del sistema a su alrededor. Siendo además el objetivo específico que más nos acerca a la resolución del objetivo final.

Este sprint se ha implementado mediante el software de diseño CAD *Autodesk Fusion 360*, mencionado anteriormente y estudiado en el presente máster.

El modelado inicial del filtro vendrá inspirado en la comprensión de soluciones comerciales similares. Se establecen ciertas restricciones que debe cumplir el filtro, estas limitaciones determinarán el diseño final del objeto.

- El filtro debe ser externo, de forma que quede colocado en un lateral del acuario, lo que facilitará la inclusión de la electrónica en el sistema.
- El sistema de retorno de agua debe realizarse por gravedad. El retorno del agua por gravedad imposibilita que el nivel del agua dentro del objeto pueda alcanzar la parte superior del filtro y desbordar fuera del filtro, donde se colocará la electrónica. El diseño permitirá que el agua comience a caer dentro del acuario antes de salir hacia afuera.
- Existirán rebosaderos adicionales que permitirán evacuar el agua hacia el acuario en caso de que el rebosadero principal pudiera atascarse.
- Se diseñará una cesta que contendrá el material filtrante de forma que pueda ser extraído y cambiado con facilidad.

Se describirán de forma detallada algunos de los aspectos más relevantes de la implementación de este sprint.

4.2.1.1. Diseño del sistema de entrada de agua

La entrada de agua al filtro se realizará a través de una oquedad, colocada en la parte superior del filtro, por encima del nivel de los rebosaderos, lo que impedirá que el agua pudiera salir por el agujero. Este sistema permitirá introducir una manguera de agua que en el extremo contrario estará conectada a la bomba que permitirá subir el agua desde el acuario hasta el filtro.

Este diseño presenta dos ventajas respecto a los sistemas de succión incluidos en los filtros convencionales. En primer lugar, el hecho de tratarse de una manguera y no de un tubo rígido, permitirá la sustitución de esta en caso de deterioro o atasco. En segundo lugar, un problema que se achaca siempre a este tipo de sistemas de filtrado es el hecho de que la toma de agua está muy cercana al lugar donde el filtro evacúa el agua. El sistema aquí propuesto permite alejar la bomba de agua, incluso colocarla en el extremo contrario del acuario, provocando así un flujo de agua constante que abarcará toda la superficie del acuario.

4.2.1.2. Sistema de evacuación de agua

El sistema de evacuación de agua, tal y como especifican las limitaciones establecidas para este objetivo, se realizará por gravedad. Será necesario sobredimensionar el hueco que permitirá el paso del agua de vuelta al acuario, de tal forma que se evite el desbordamiento del filtro. La capacidad de desagüe ofrecida por el diseño será la que limite la capacidad de la bomba, no obstante, la oquedad diseñada para tal cometido está sobredimensionada de tal forma que la integridad del filtro no se verá comprometida por una bomba de acuario de pequeño formato (50L/h – 150L/h). Esta parte del filtro es probablemente la parte más relevante, y será la que delimite el éxito o fracaso del diseño. Se incluyen además dos rebosaderos secundarios que permitirán evacuar el agua cuando el filtro quede obstruido en su parte central (parte en la que se coloca el material filtrante), de tal forma que el agua pueda salir por estos rebosaderos laterales.

4.2.1.3. Colocación del caudalímetro

Para obtener mediciones fiables, el caudalímetro o sensor de caudal debe estar colocado en un punto en el que todo el flujo del agua lo atraviese. Esto plantea la problemática de valorar el comportamiento del filtro ante un posible atasco en el sensor. Con el fin de minimizar los posibles atascos, se decide colocar el caudalímetro en la salida del filtro, una vez que el agua contiene menos suciedad. No obstante, y por seguridad, se ha decidido mantener una pendiente en la desembocadura del filtro provocando que el agua desemboque por gravedad en caso de atasco, además de añadir dos rebosaderos laterales. El caudalímetro quedará colocado en el orificio diseñado en la desembocadura del filtro mediante una rosca isométrica de paso 2 mm.

4.2.1.4. Colocación de la sonda de temperatura

La sonda de temperatura partirá de la caja situada en la parte superior del filtro que contendrá la electrónica, saliendo de esta con un conector con protección antihumedad (no competente a esta parte del desarrollo) que evitará las posibles salpicaduras en el interior de la caja. El sensor de temperatura quedará colocado en el interior del compartimento principal del filtro, atravesando la tapa superior por un agujero.

4.2.1.5. Paredes del filtro

La salida del agua por gravedad evita que las paredes estén sometidas a presión más allá de la realizada por el propio peso del agua. El filtro debe tener un grosor suficiente para soportar

esta presión. Los filtros comerciales tienen un grosor de 2mm (aproximadamente), sin embargo, en este caso se ha decidido sobredimensionar hasta los 5 mm para asegurar la integridad de la pieza.

En el caso de llevar este prototipo de sistema de filtrado IoT a un producto comercial sería necesario realizar un estudio de resistencia de materiales. Este estudio permitiría asegurar la integridad de la pieza del filtro reduciendo al mínimo el material utilizado y así abaratar los costes de fabricación.

4.2.1.6. Sujeción del filtro

El diseño de la sujeción del filtro al cristal del acuario es, junto con la salida del agua, el punto crucial de este diseño. Este tipo de filtros se sujetan al cristal mediante hendidura, en la forma “U” invertida, que actuará a modo de enganche en la pared del acuario. El diseño de esta hendidura se realizará de tal modo que el grosor de la pared sea suficiente, asegurando que esta podrá cargar el peso del acuario lleno de agua. No obstante, en la fase de fabricación, se realizará un estudio de materiales y técnicas de fabricado que permitirán definir el grosor final de la pieza.

4.2.1.7. Modelo de pieza obtenida

A continuación, se presentan los planos resultantes del diseño de la pieza. Estos planos han sido generados con Autodesk Fusion 360, el mismo software con el que la pieza ha sido diseñada. Los planos muestran cuatro vistas: Alzado, planta, perfil y modelo 3D. En estos esquemas se muestra la pieza del filtro y la pieza de la cesta que será colocada en su interior para contener el material filtrante.

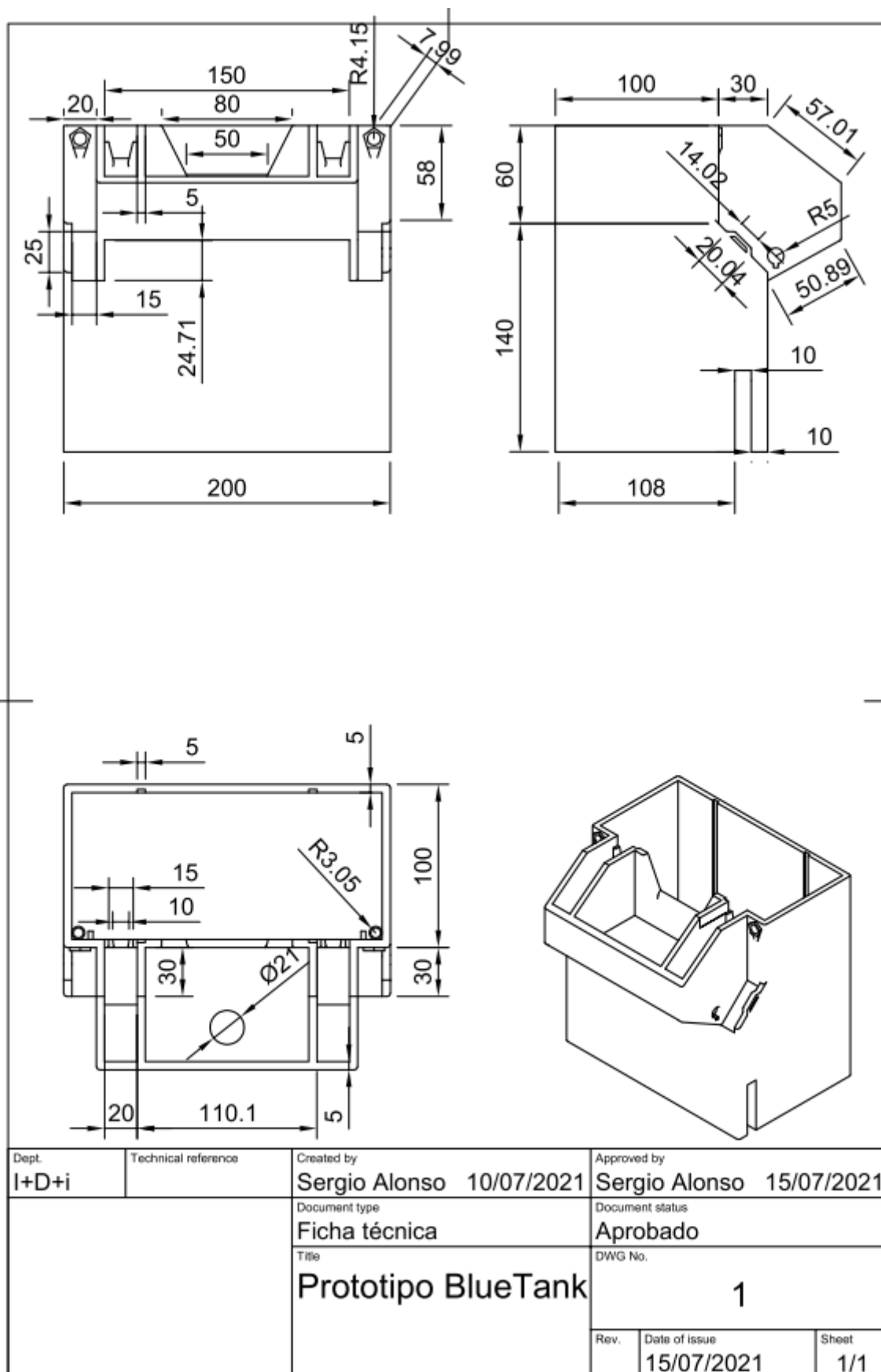


Ilustración 31. Ficha técnica del filtro

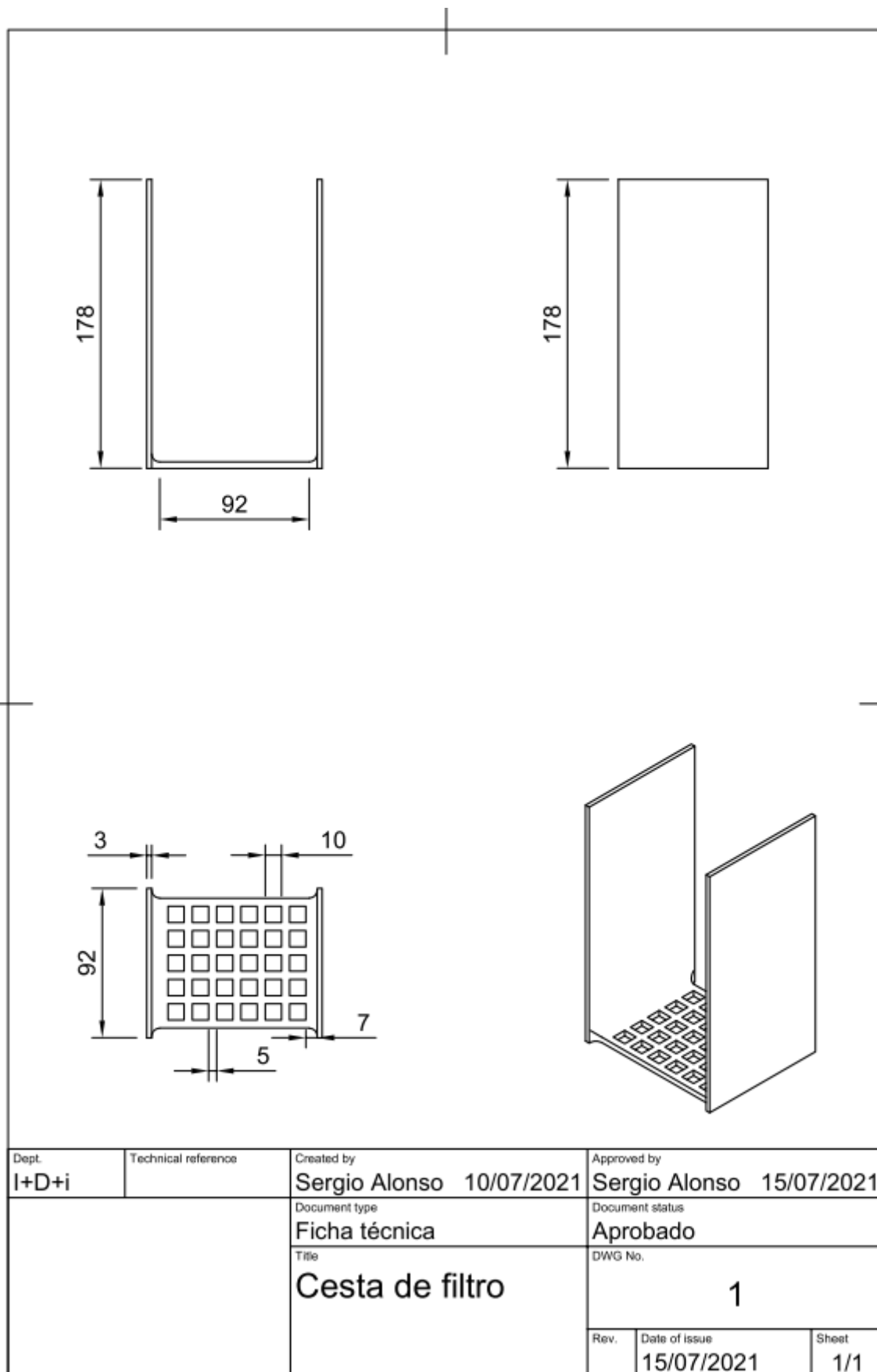


Ilustración 32. Ficha técnica de la cesta del filtro

4.2.2. Sprint 3: Fabricación del filtro

El presente sprint pretende culminar con la fabricación del filtro. El sprint backlog propuesto para este sprint queda definido de como se aprecia en la imagen. La duración establecida para este sprint es de cinco días.

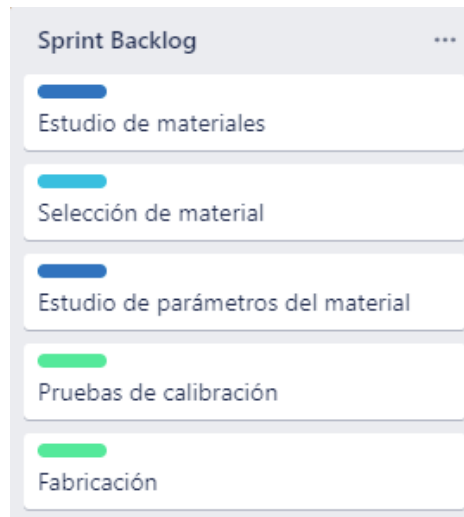


Ilustración 33. Documento Sprint Backlog del Sprint 3

Se trata de un sprint sencillo. Se implementará mediante tecnología FDM, ya que es el tipo de tecnología de fabricación que se encuentra al alcance del alumno. Este tipo de tecnología no es la más adecuada para la fabricación de recipientes estancos. Una técnica de fabricación mucho más viable para estos casos sería la fabricación por inyección de plástico, pero los elevados costes del molde imposibilitan el uso de esta técnica para este proyecto. No obstante, el presupuesto proporcionado en la memoria hace referencia a los costes de fabricación con una técnica de inyección.

Cabe destacar la importancia en la selección del material, se trata del material PETG. Un material apto para fabricar materiales de consumo, ya que no deja residuo y que cuenta con certificación de no toxicidad. Se trata del material más común en la fabricación de botellas de agua para consumo humano.

Además de sus características inocuas, el PETG cuenta con otras características que lo hacen idóneo para esta tarea. Tiene una resistencia alta y no se degrada, a corto y medio plazo, en contacto con el agua.

Según las especificaciones del fabricante adjuntas a esta memoria, el material tiene un límite elástico de 68 MPa.

Se realiza una simulación por elementos finitos con la herramienta con Autodesk Inventor. Este proceso permitirá determinar las deformaciones que sufrirá la pieza sometida a las cargas a las que está expuesta.

El primer paso es conocer el peso que habrá sobre nuestra pieza, en este caso unos 5Kg aproximadamente entre el peso del agua y el de la pieza (fabricada con una densidad de relleno del 100%). Realizando el cálculo en Newtons obtenemos:

$$P = 5Kg * 9.8 = 49 N$$

Siendo 9.8 la fuerza de la gravedad terrestre.

Esto nos da una carga de 286.5 Pascales:

$$W = \frac{49}{0,171} = 286.5 Pa$$

Siendo 0,171 m³ el área de la base del filtro diseñado.

286.5 Pa será la fuerza que utilizaremos en el Autodesk inventor para realizar la simulación, estableciendo las restricciones en el acople de la pieza al cristal, ya que será el punto donde la pieza soportará el peso. Autodesk Inventor, de forma automática, calculará las tensiones de Von Mises, obteniendo que la tensión máxima sufrida por esta pieza aplicando estas fuerzas será de 0.02882 Mpa, y que el límite elástico, según el fabricante, soportado por el material es de 68 Mpa, deducimos que la pieza, con el diseño realizado, no se rompe y no sufrirá deformación.

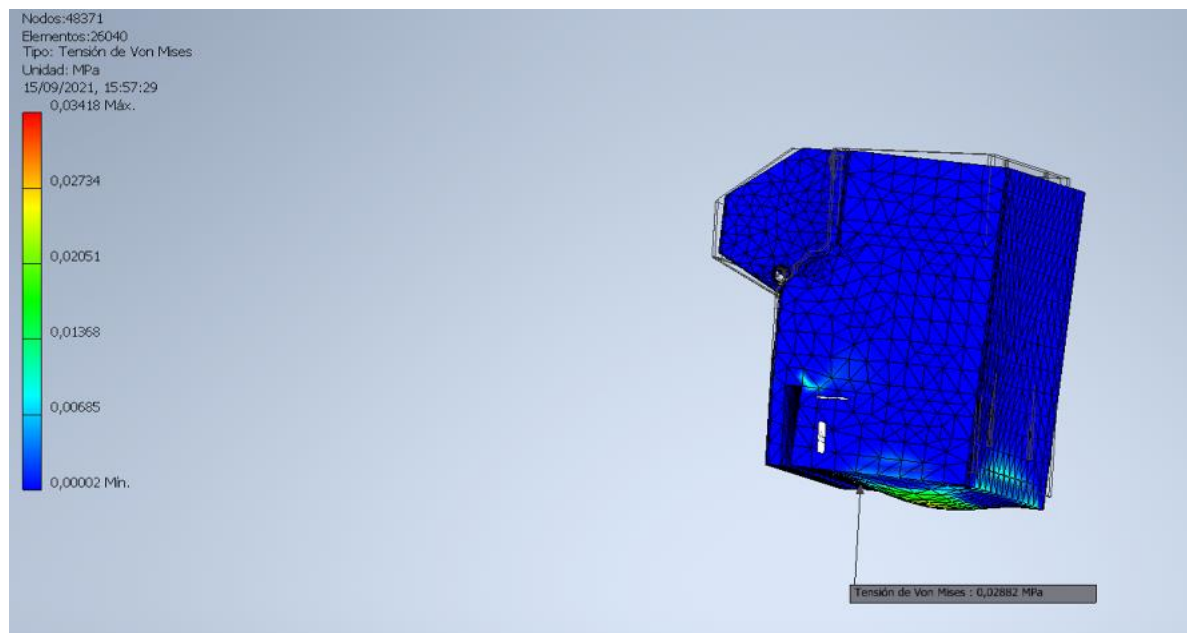


Ilustración 34. Simulación por elementos finitos y cálculo de deformaciones.

Tras los cálculos, se realizan las pruebas pertinentes para la fabricación, se recoge una tabla con los parámetros más relevantes utilizados en la fabricación

Tabla 7. Parámetros principales de impresión

Parámetro	Valor
Temperatura de fusión	235 °C
Densidad de relleno	100%
Patrón de diseño	“Triángulos”
Altura de capa	0.25 mm
Temperatura de la cama	85 °C
Retracciones	Activadas
Soportes	Activados

La densidad de relleno se ha establecido al 100% con la finalidad de evitar las fugas de agua, el resultado de esta operación ha sido exitoso.

4.2.2.1. Resultado obtenido



Ilustración 35. Pieza fabricada con tecnología FDM



Ilustración 36. Prototipo de sistema de filtrado IoT completo



Ilustración 37. Instalación del filtro general

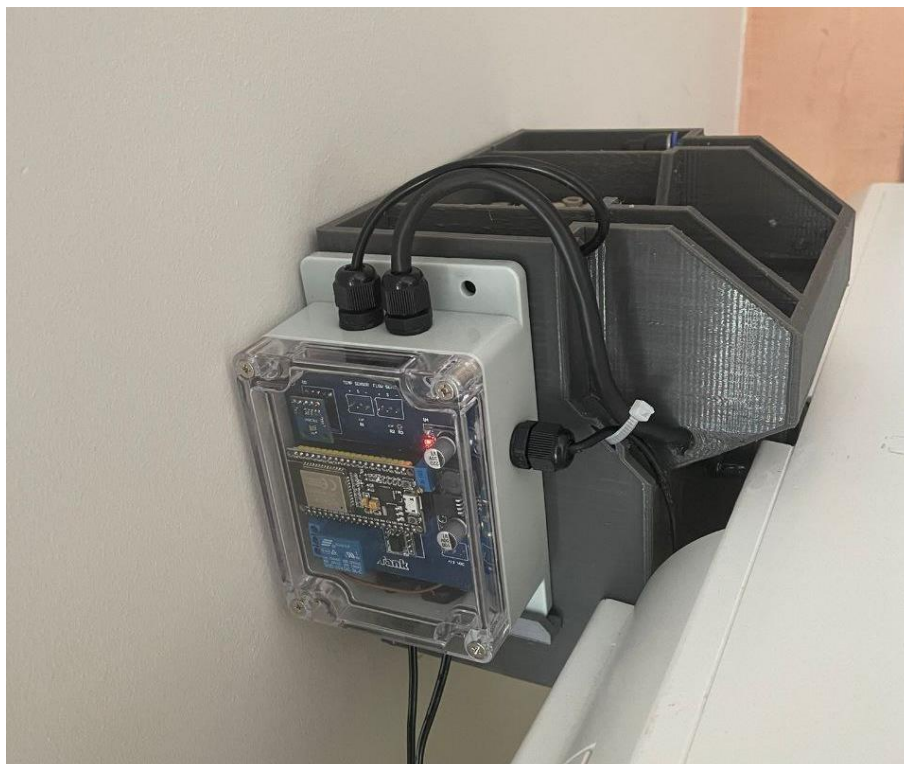


Ilustración 38. Instalación del filtro detalle

4.3. Desarrollo del Firmware

El desarrollo del firmware no comprende la consecución de un objetivo específico, no obstante, es un paso que resulta imprescindible para la resolución del objetivo principal de este proyecto, que no es otro que desarrollar un sistema de filtrado IoT. El firmware se ejecutará en el microcontrolador ESP32 incluido en el desarrollo de la electrónica.

Para la implementación del firmware se define el siguiente sprint backlog, que será llevado a cabo en un sprint de quince días.

El firmware implementado queda definido por el flujo de ejecución definido en el diagrama presentado a continuación. No obstante, este firmware será completado con funcionalidades añadidas propias de los objetivos específicos que se implementan a continuación en esta misma memoria.



Ilustración 39. Documento Sprint Backlog del Sprint 5

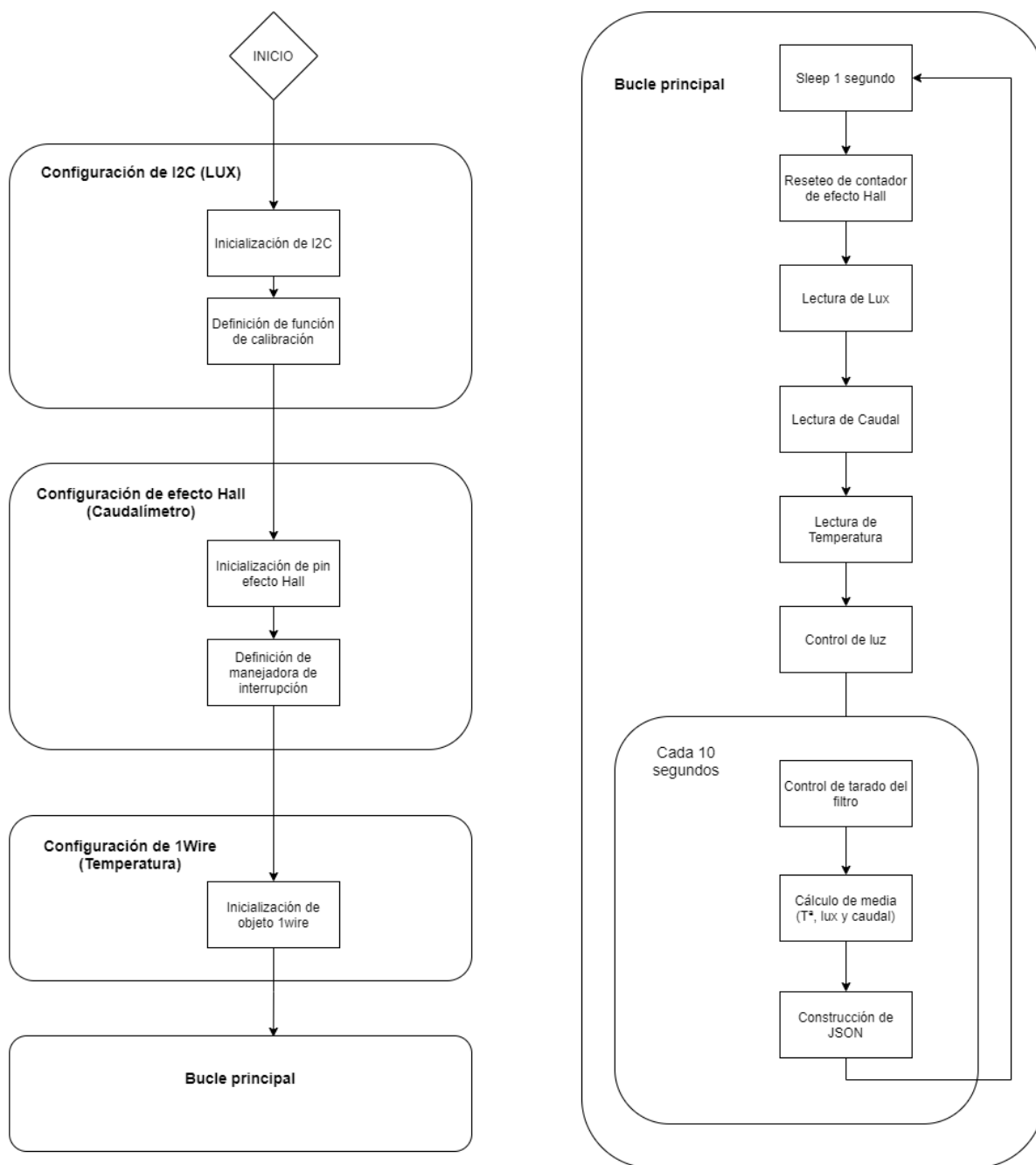


Ilustración 40. Diagrama de flujo general de ejecución del firmware



Ilustración 41. Diagrama de flujo detalle de manejadora efecto Hall

Será preciso dedicar una explicación a determinadas áreas del firmware que deben ser especificadas en mayor detalle.

4.3.1. Lenguaje de implementación

La totalidad del firmware desarrollado ha sido implementada en el lenguaje Micropython. Una variante del conocido lenguaje de programación Python desarrollada específicamente para la implementación de firmware. Este lenguaje mantiene una sintaxis idéntica a Python y permite la compatibilidad con un gran número de bibliotecas de este lenguaje, lo que convierte a Micropython en un lenguaje muy potente y versátil. La utilización de este lenguaje para el desarrollo del firmware vendrá dada principalmente por la familiaridad del alumno con el lenguaje de programación Python 3.

4.3.2. Bibliotecas utilizadas

- `tsl2561`: Biblioteca diseñada por Adafruit para el sensor de luxes homónimo. Esta biblioteca gestiona el protocolo de comunicación I2C, permitiendo abstraer la lógica de control de este y manejar la lectura/escritura a través de un objeto de programación.
- `onewire`: Biblioteca propia del lenguaje Micropython dedicada al manejo del protocolo 1wire. Proporciona un objeto que permite trabajar con el protocolo de forma abstracta.
- `ds18x20`: Biblioteca implementada en Micropython que permite manejar los sensores de la familia ds18x20 entre los que se encuentra el sensor de temperatura utilizado. La biblioteca proporciona las herramientas necesarias para gestionar la lectura de estos sensores haciendo uso de un objeto *onewire* obtenido de la biblioteca anterior
- `machine`: Biblioteca propia del lenguaje Micropython que permite manejar los pines y las interfaces físicas del microcontrolador.
- `time`: Biblioteca propia de Python adaptada al lenguaje Micropython. Permite, entre otras funcionalidades, realizar esperas no ocupadas en el dispositivo.
- `esp`: Biblioteca específica del esp32 que proporciona la capacidad de trabajar con elementos propios del microcontrolador.

4.3.3. Calibración de sensores

4.3.3.1. Sensor de temperatura

El sensor de temperatura seleccionado, la sonda DS18B20, cuenta con una biblioteca propia desarrollada por el fabricante. Esta biblioteca implementa la recta de calibración necesaria para transformar las lecturas del sensor a °C. No obstante, los resultados obtenidos han sido comparados con las mediciones de un termómetro digital de origen comercial con resultados satisfactorios.

4.3.3.2. Sensor de luxes

Como sensor de iluminancia, medida en luxes, se ha seleccionado el sensor TSL2561. La calibración de este sensor se ha realizado de forma manual.

La recta de calibración necesaria para ajustar las mediciones del sensor se ha obtenido a partir del uso de un luxómetro profesional, modelo T10A, cuya ficha se adjunta a esta memoria. El proceso de muestreo recoge los resultados presentados a continuación, siendo estos obtenidos por el sensor y el luxómetro en las mismas condiciones de luminosidad. A partir de estos resultados es posible obtener una recta de calibración que permitirá ajustar las mediciones del sensor.

Tabla 8. Comparativa entre sensor y luxómetro calibrado

Luxómetro profesional (lux)	Sensor TSL2561 (lux)
416	34
185	15
110	8
19	1

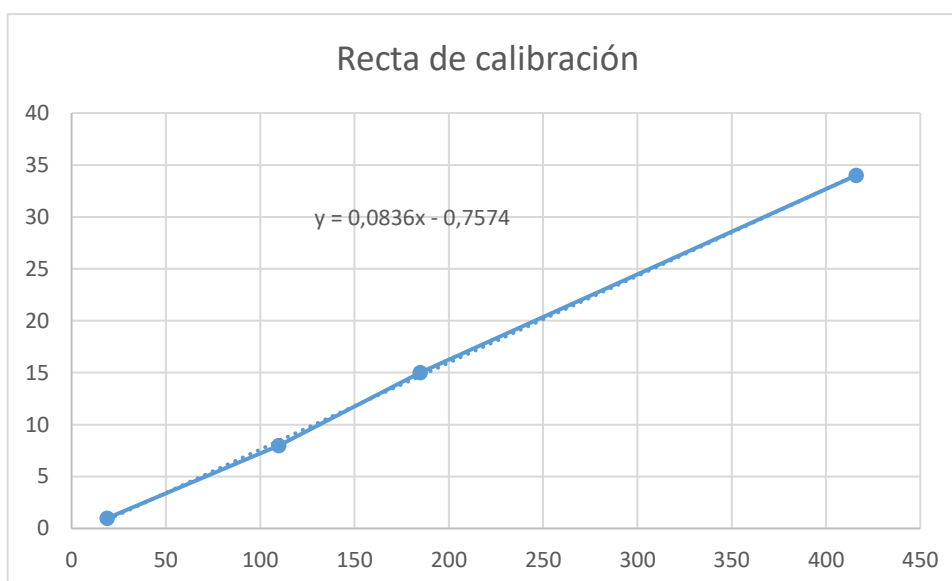


Ilustración 42. Recta de calibración luxómetro

El resultado obtenido muestra la siguiente recta de calibración, habiendo sido esta implementada en el firmware.

$$Y = 0.0836X - 0.7574$$

4.3.3.3. Caudalímetro

Se trata de un sensor de efecto Hall. La lectura de este sensor consiste en leer los pulsos emitidos por el sensor durante un segundo. El fabricante del sensor proporciona una recta de conversión que permite transformar los pulsos a litros hora. No obstante, merece la pena señalar que la variable caudal del filtro se ha tratado como un porcentaje. El proyecto implementa la posibilidad de que el usuario establezca el 100% del caudal. Para esto el usuario realiza un tarado en el momento de la limpieza del filtro, cuando su caudal es máximo. El valor recogido en el proceso de tarado representa el punto de referencia a partir del cual se determinarán las variaciones en el porcentaje de caudal.

Tabla 9. Tabla de transformación ofrecida por el fabricante

L/H	Frecuencia (Hz)
120	16
240	32.5
360	49.3
480	65
600	82
720	90.2

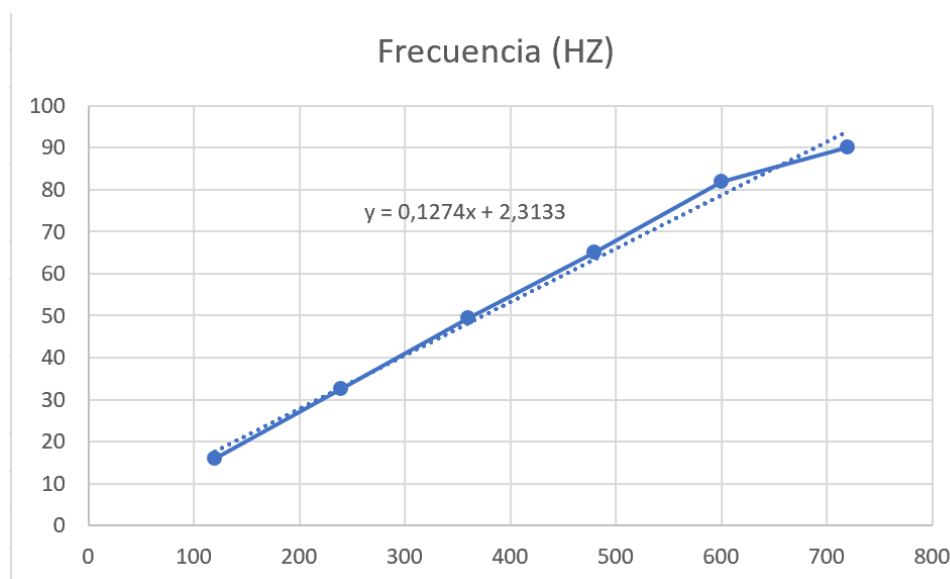


Ilustración 43. Recta de calibración del caudalímetro

La recta de calibración resultante es:

$$Y = 0.1274X + 2.3133$$

4.4. Objetivo específico: Implementar un sistema de configuración del dispositivo

En este apartado se especifica la implementación del objetivo específico consistente en implementar un sistema de configuración del dispositivo, esto es, se plantea que el dispositivo debe poder ser configurable por un usuario con conocimientos básicos de informática, entendiendo por básicos el manejo de un navegador web y la familiaridad con un entorno Android, iOS o Windows. Además, el filtro debe tener un sistema de instalación no más complejo que un filtro comercial. Entendiendo por instalación la colocación y encendido del sistema de filtrado en el propio acuario.

La instalación sencilla del filtro está implícita en su propia construcción, esto es, el hecho de mantener una construcción similar a los filtros de carácter comercial implica que la instalación del filtro aquí desarrollado no será de superior dificultad a una instalación convencional.

4.4.1. Sprint 6

Para acometer la característica de configuración sencilla, se realizará el sprint definido en el sprint backlog representado en la imagen. Este sprint ampliará el firmware diseñado dotando a este de la capacidad de realizar una configuración sencilla de la conexión wifi doméstica. Esta capacidad se considera un aspecto fundamental para el éxito de este producto por dos motivos. Se trata del único punto que requiere una configuración específica, obviando la configuración de alertas, que siempre será opcional. Por otra parte, se hace necesario implementar un sistema de configuración sencilla, ya que, de otro modo, el usuario podría rechazar el producto.

Para la realización de este sprint se establece un plazo de cinco días.

Este sprint implica modificaciones en el firmware para implementar la nueva funcionalidad requerida. Esta funcionalidad se implementará al comienzo del bucle principal de ejecución, añadiendo la siguiente modificación al diagrama de flujo del firmware.



Ilustración 44. Documento Sprint Backlog del Sprint 6

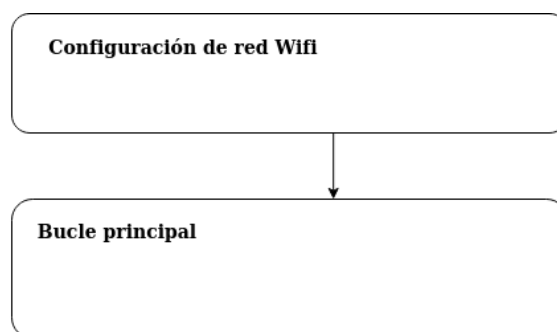


Ilustración 45. Añadido de sistema de configuración wifi en el firmware

La configuración de la red wifi se detalla de acuerdo al siguiente diagrama. En este caso el flujo de control de este procedimiento viene impuesto por la funcionalidad de la propia biblioteca utilizada.

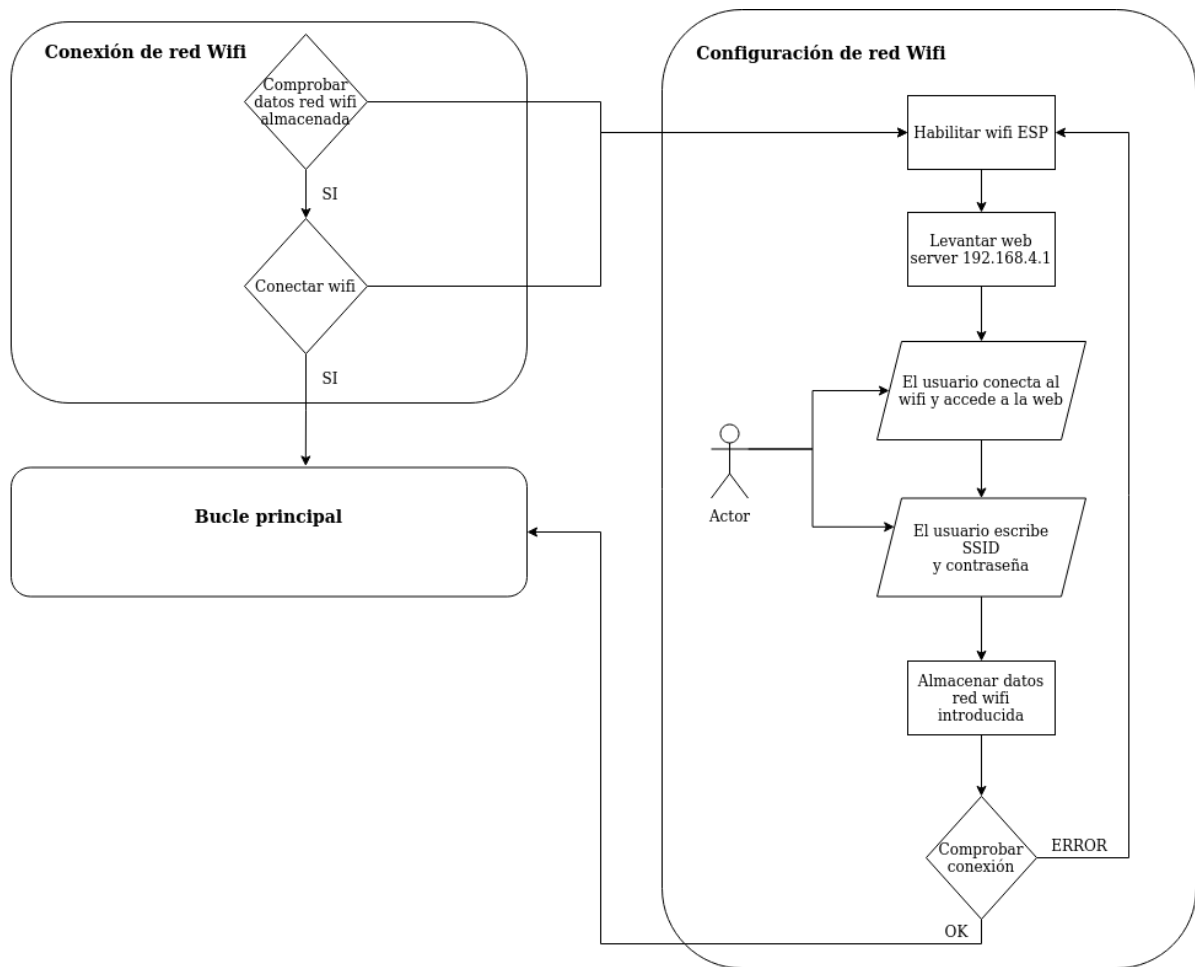


Ilustración 46. Diagrama de flujo de la configuración wifi en el firmware

4.4.1.1. Biblioteca utilizada

- wifimgr: *wifimgr* o “Wifi Manager” consiste en una implementación no oficial para el lenguaje micropython de la famosa biblioteca de arduino *WifiManager*. Se trata de una biblioteca de código abierto descargable en la página de Github “tayfunulu/WifiManager”. La biblioteca ha sido adaptado para encajar con el aspecto estético de este proyecto. Esta biblioteca proporciona la funcionalidad descrita en el diagrama de flujo anterior, permitiendo almacenar en memoria no volátil los datos de acceso a la red Wifi introducidos por el usuario. En el supuesto de tener en memoria los datos de acceso a la red o estos no ser válidos, la propia biblioteca levanta un servidor en el que muestra una interfaz web que permite configurar la red wifi a la que se conecta el dispositivo.

4.4.1.2. Interfaz web

La biblioteca proporciona una interfaz web para la configuración de los parámetros de acceso a la red wifi. Esta interfaz ha sido modificada en el apartado estético con la finalidad de mantener una coherencia visual en lo referente a colores y tipografía seleccionados para este desarrollo.

Ilustración 47. Interfaz de la configuración wifi en el firmware

4.5. Objetivo específico: Mejorar la escalabilidad

El objetivo de mejorar la escalabilidad del sistema está apoyado en el estudio de (Yokotani & Sasaki, 2016). Un ecosistema IoT en el que se realice un intercambio de información constante y que requiera características de tiempo real debe ser capaz de soportar la conexión simultánea de múltiples dispositivos. Los autores (Yokotani & Sasaki, 2016) concluyen que el protocolo MQTT, por tratarse de un protocolo de implementación ligera, tiene un comportamiento muy superior a HTTP en una red de nodos IoT, donde, por regla general, el número de mensajes transmitidos es elevado pero el contenido de estos es de pocos bytes.

Al margen de este estudio, otro aspecto estable del protocolo MQTT será la capacidad de los brokers MQTT para trabajar de forma colaborativa, permitiendo que ciertos brokers actúen como clientes de otros de tal forma que la carga pueda ser balanceada en tiempo real incrementando la escalabilidad del sistema.

Además de la importancia de la escalabilidad, se toma como aspecto crucial la importancia de la seguridad en el intercambio de información. El protocolo MQTT soporta cifrados mediante un sistema de clave pública-privada y asegura la entrega del mensaje. A pesar de la importancia de la privacidad e integridad de la información, este no se considera un objetivo específico de este trabajo de fin de estudios, por ser ya un proyecto demasiado denso. Sin embargo, la arquitectura y tecnologías seleccionadas en el sistema propuesto no deben estar reñidas con la capacidad de crear un sistema seguro, de tal forma que esta seguridad pueda ser integrada en un futuro como parte del proyecto, sin tener que realizar un rediseño estructural.

Estas características hacen que se decida implementar el sistema de paso de mensajes mediante el protocolo MQTT, frente a la API REST desarrollada en la versión 1.0 del sistema Bluetank.

Este objetivo se implementará mediante la consecución de dos sprints, dedicados a la instalación del bróker MQTT y la base de datos en el lado del servidor y a la implementación del envío de mensajes MQTT en el lado del dispositivo.

4.5.1. Sprint 7

Este sprint pretende implementar la configuración de un servidor “Ubuntu server”, encargado de alojar el sistema cloud del proyecto Bluetank 2.0, compuesto por una base de datos, el bróker MQTT y una interfaz web (implementada más adelante). Para este sprint se define el sprint backlog presentado en la siguiente imagen y se establece un plazo máximo de cinco días.

4.5.1.1. Selección de base de datos

La selección del motor de base de datos seleccionado para esta implementación determinará en su totalidad la manera en la que los datos serán almacenados. Se plantean dos alternativas, MongoDB como base de datos no relacional y PostgreSQL como base de datos relacional. Una vez más, atendiendo a razones de escalabilidad, se

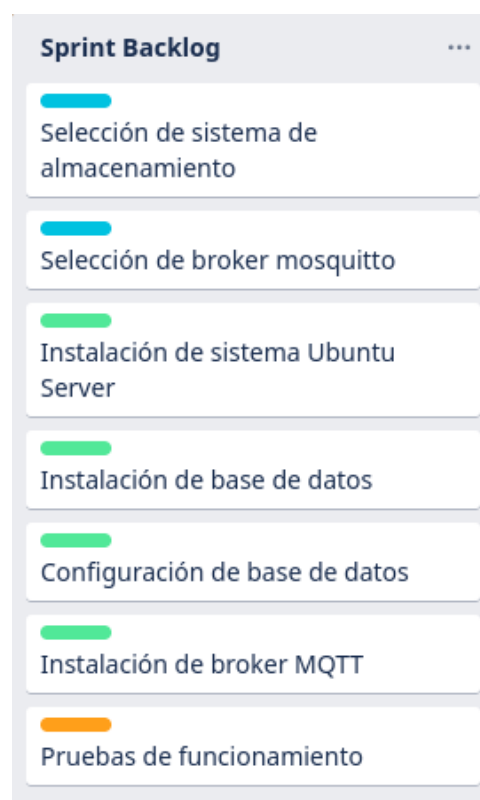


Ilustración 48. Documento Sprint Backlog del Sprint 7

opta por seleccionar el motor de bases de datos no relacional, mucho más óptimo en materia de rendimiento cuando se trata de crear almacenes de datos procedentes de las mediciones tomadas en múltiples dispositivos, sin que estas tengan ninguna (o muy poca) relación entre sí, más allá del identificador de dispositivo del que han sido obtenidas.

La base de datos seleccionada, MongoDB, se instalará en el servidor habilitado para el alojamiento de los servicios cloud incluidos en este proyecto.

4.5.1.2. Selección del broker MQTT

Existen múltiples implementaciones open source de brokers que soportan el protocolo MQTT. En este proyecto se opta por el broker Eclipse Mosquitto de la compañía Oracle. El broker será instalado y configurado en el servidor habilitado para esta tarea. La selección de este broker viene dada por la familiaridad del alumno con este sistema.

4.5.1.3. Implementación de gestor de mensajes

El gestor de mensajes será un elemento que permitirá la recogida de mensajes en el bróker, procedentes de cada uno de los dispositivos de la red IoT y se encargará de redirigir ese tráfico hasta la base de datos implementada.

En este caso se ha implementado un pequeño script en Python ejecutado de forma continua en el servidor que se encargará de suscribirse al tópico de recepción de mensajes del bróker y redirigirá los mensajes entrantes hacia la base de datos MongoDB, identificando al dispositivo emisor del mensaje para registrar estos en la *collection* correspondiente al dispositivo en cuestión.

Una de las ventajas de una arquitectura modular es que uno de sus componentes puede ser sustituido sin afectar al resto. Este script, en una implementación futura, podría ser sustituido por un gestor de colas como Apache Kafka sin afectar al resto de módulos del sistema.

4.5.2. Sprint 8

Para lograr la comunicación con el sistema cloud será necesario implementar un segundo sprint con una duración total de cinco días. Este sprint pretende implementar el protocolo MQTT en el lado del dispositivo, dotando a este de la capacidad de comunicarse con el bróker instalado en el servidor de Bluetank.



Ilustración 49. Documento Sprint Backlog del Sprint 8

4.5.2.1. Implementación del firmware

Será necesario ampliar el firmware dotándolo de la capacidad de transmitir y recibir mensajes a través del protocolo MQTT. A continuación, se presenta un diagrama descriptivo del flujo de ejecución del protocolo MQTT en el firmware previamente desarrollado.

La implementación del protocolo MQTT en el firmware implica cambios en diferentes partes del firmware. En primer lugar, será necesario realizar la configuración del cliente MQTT una vez se haya realizado con éxito la conexión al wifi.

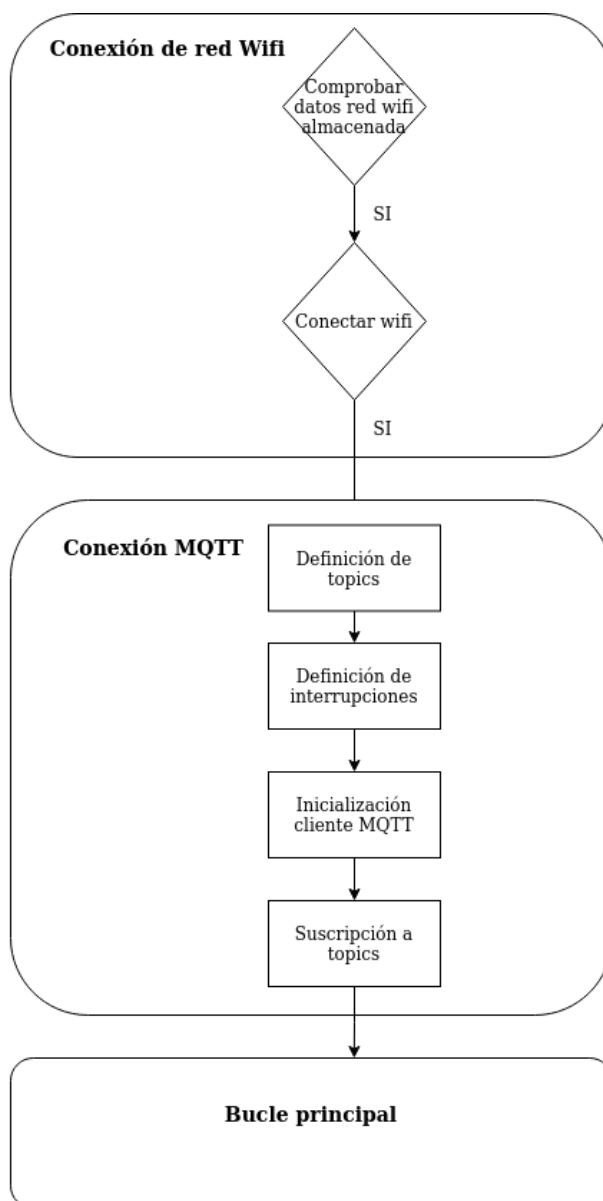


Ilustración 50. Añadido de sistema de configuración MQTT en el firmware

Por otra parte, será necesario incluir el envío de datos dentro del bucle principal, una vez obtenido el JSON con las mediciones resultantes, quedando dicho bucle implementado como se indica en la imagen.

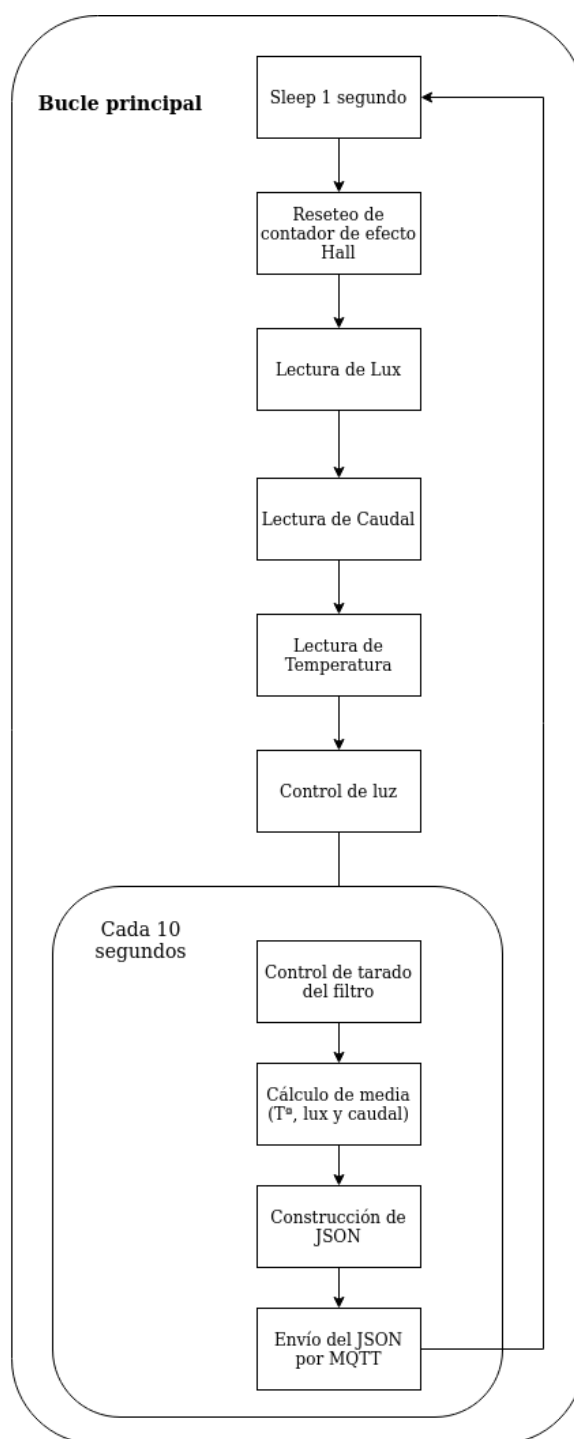


Ilustración 51. Diagrama de flujo del bucle principal del firmware modificado para envío MQTT

Se hace necesario para la comprensión total del firmware describir las interrupciones declaradas para manejar los mensajes entrantes del protocolo MQTT, de tal forma que exista la posibilidad de cerrar el lazo, obteniendo una comunicación bidireccional entre el dispositivo y el entorno cloud, permitiendo así la posibilidad de comandar y configurar el dispositivo desde una posición remota.

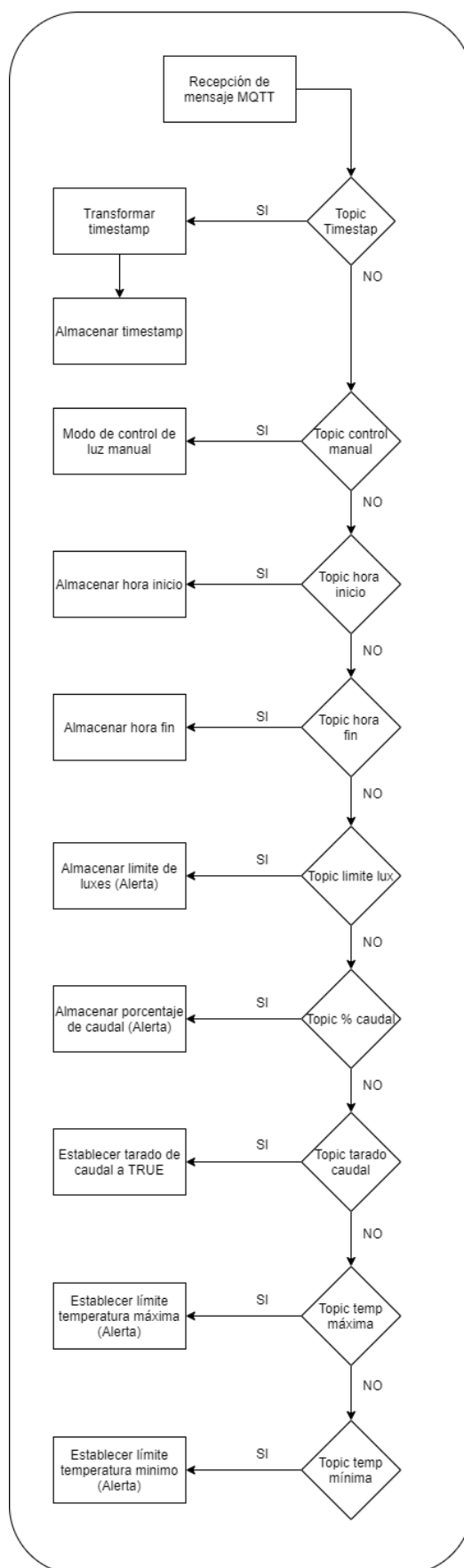


Ilustración 52. Diagrama de flujo de recepción de parámetros de configuración MQTT

4.5.2.2. Bibliotecas utilizadas

- umqttsimple: Se trata de una biblioteca que implementa el protocolo MQTT en el lenguaje Micropython. El manejo del protocolo se realiza a través de un objeto cliente que permite abstraer la lógica de control del propio protocolo.
- ubinascii: Biblioteca utilizada para la gestión de cadenas de texto transmitidas en formato binario a través del objeto cliente proporcionado por la biblioteca umqttsimple mencionada anteriormente.

4.5.2.3. Servidor NTP

Se implementa un servidor NTP con la finalidad de proporcionar el timestamp o marca temporal a los dispositivos conectados al bróker. El servidor NTP vía MQTT estará montado en el cloud de Bluetank. Se ha implementado en Python 3, el código se adjunta a esta entrega.

Este es el último sprint que implica al firmware, obteniendo al final del proceso un diagrama de flujo de fimware como el que se muestra en la ilustración 50.

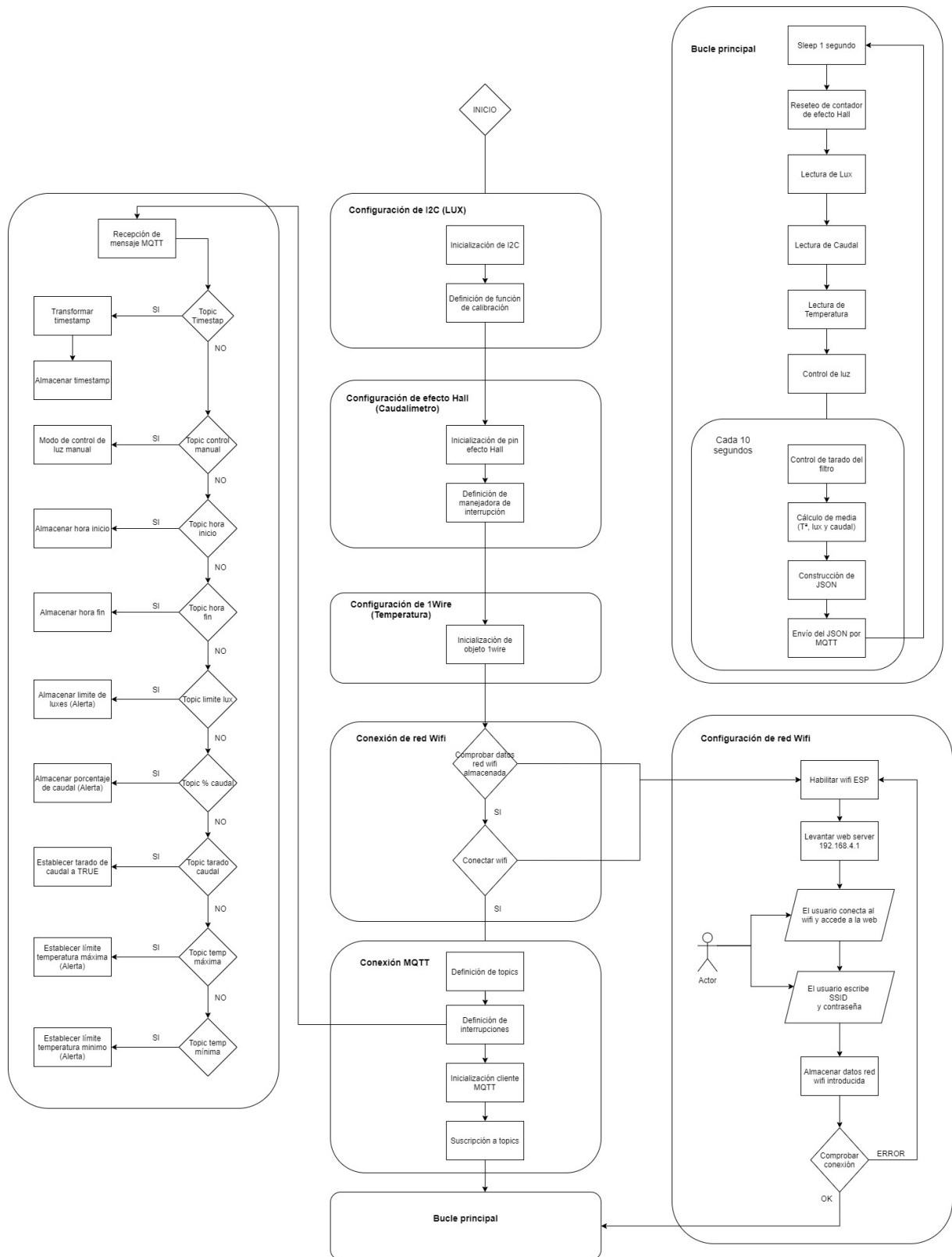


Ilustración 53. Diagrama firmware completo

4.6. Objetivo específico: Mejorar la experiencia de usuario

El objetivo de mejorar la experiencia de usuario pretende ofrecer al usuario una experiencia de IoT completa. Se implementará una interfaz sencilla que permita al usuario visualizar los datos procedentes de su dispositivo, visualizando gráficas en tiempo real, graficas de datos históricos y alertas por parámetros, además de otorgarle la capacidad de configurar los valores límite de las alarmas en función de las necesidades específicas.

Con la finalidad de aumentar la calidad del sistema, la web desarrollada se integrará con una plataforma IoT que permitirá el análisis y la visualización de datos.

Este objetivo se subdivide en dos sprints dedicados al desarrollo de la página web y a la integración de la plataforma IoT.

4.6.1. Sprint 9

Este sprint se dedica a la implementación de la web. Esta web cumple la función de integrar todos los elementos desarrollados hasta el momento en una sola plataforma, a través de la cual el usuario podrá interactuar con el sistema.

Se pretende desarrollar una interfaz amigable y sencilla de forma que sea aceptada por todos los tipos de usuarios.

El sprint queda definido en el documento sprint backlog y para su realización de este sprint se planifican diez días.



Ilustración 54. Documento Sprint Backlog del Sprint 9

El sprint dará comienzo realizando una recopilación de ideas y ejemplos sobre la posible interfaz y seleccionando la plantilla que mejor cumple con los requerimientos de la web.

4.6.1.1. Plantilla seleccionada

Para la implementación se ha utilizado esta plantilla gratuita de la plataforma WrapPixel (Homepage, s. f.). La plantilla proporciona un menú limpio, funcional y modificable que otorga el acceso a las diferentes páginas sobre las que, no obstante, se realizará una modificación completa del contenido.

4.6.1.2. Implementación del sistema de login

Con el fin de salvaguardar la información del usuario, es requisito indispensable que el acceso a la web se realice mediante un sistema de login. Se trata de un sistema de login simple que solo almacena un usuario, no obstante, en un sistema en producción sería necesario implementar una API de gestión de usuarios con su base de datos correspondiente que permita vincular los dispositivos a los usuarios. Al tratarse de un sistema modular, esta parte podría ser sustituida sin afectar al resto de elementos.

4.6.1.3. Envío MQTT

Se hace imprescindible el echo de que la web pueda comunicarse con el dispositivo a través del protocolo MQTT ya implementado, de forma que el lazo de control quede cerrado. Para esta web se ha implementado un apartado configuración que permitirá al usuario establecer la configuración de aletas deseada, además de ejecutar el control de luz. Los datos recogidos en el formulario de configuración serán enviados por MQTT con diferentes topics. El envío por MQTT se ha implementado en PHP haciendo uso de las herramientas Linux “mosquitto_pub” y “mosquitto_sub” propias del bróker Mosquitto.

4.6.1.4. Integración de la plataforma IoT

La plataforma IoT será la encargada de la representación gráfica de los datos en tiempo real, datos históricos e generación de alertas. La integración entre el contenido generado por esta plataforma y la aplicación web desarrollada se realizará a través de iframes contenidos en las vistas de la web. Los iframes permitirán mostrar dentro de la web el contenido generado por la plataforma.

4.6.2. Sprint 10

El presente sprint pretende dar término al desarrollo del proyecto implementando la funcionalidad requerida de la plataforma IoT. Esta es, generación de visualizaciones, a tiempo real e históricas y generación de alertas. El sistema IoT, además, debe permitir el análisis de datos.

La finalidad principal de la recogida de datos masivos es el estudio de estos. A pesar de no ser un objetivo específico de este proyecto, se hace de especial interés que la plataforma IoT seleccionada admita el análisis de datos masivos.

Se realiza un sprint de cuatro días cuyo sprint backlog queda definido de la siguiente forma.



Ilustración 55. Documento Sprint Backlog del Sprint 10

4.6.2.1. Selección de plataforma IoT

En este proyecto se valorarán las características de tres plataformas IoT diferentes, estas son: Grafana, Thingsboard y Deep Intelligence. La selección de la plataforma que mejor se adapte a las necesidades es un aspecto importante que marcará la funcionalidad soportada.

Tras un estudio y una prueba de concepto se ha seleccionado la plataforma de IoT Deep Intelligence por diferentes motivos.

- No se trata solo de una plataforma IoT sino que además se trata de una plataforma de Big Data, lo que proporciona la funcionalidad de realizar diferentes análisis de datos con el fin de obtener información y generar conocimiento a través de la recogida de los mismos. Entre las posibilidades ofrecidas se encuentran: Análisis de series temporales, Modelos de aprendizaje supervisado como clasificadores y regresores, modelos de aprendizaje no supervisado, como outliers o reglas de asociación y procesamiento de lenguaje natural. A pesar de que la plataforma Thingsboard admite el análisis de datos, no es una herramienta de análisis tan completa como en el caso de Deep Intelligence.

- Manejo de alertas: Deep Intelligence permite el manejo de sistemas de alertas estableciendo condiciones sobre los parámetros recibidos, que, en caso de ser afirmativas, generarán una

alerta visual en la plataforma y se enviará un correo electrónico al usuario informando de la alerta producida.

- Ingesta de datos: La plataforma Deep Intelligence permite realizar ingesta de datos de múltiples fuentes heterogéneas. En este caso, es de particular interés, la capacidad de realizar una ingesta de datos de una base de datos MongoDB, como la que se utiliza en el servidor cloud de Bluetank.

Estos motivos hacen de Deep Intelligence la plataforma más adecuada para la situación actual.

Creado el usuario en la plataforma el primer paso será realizar la ingesta de datos, para ello se habilita el conector correspondiente a MongoDB ofrecido por Deep Intelligence. Una vez volcados los datos la creación de paneles de visualización con monitores en tiempo real o monitores de históricos es una tarea trivial. Por último, se configuran las alertas que se generarán de forma visual en cualquiera de los paneles y en forma de correo electrónico. La plataforma permite, además mantener un histórico de alertas.

Los paneles generados serán compartidos a través de una url, funcionalidad ofrecida por la plataforma, que permitirá la integración de estos paneles en la web de visualización a través de los iframes, de tal modo, que un usuario del sistema solo necesita su usuario de la aplicación web, y para la plataforma IoT es algo transparente que visualiza desde la misma web.

Finalizando el apartado referente al objetivo de mejorar la experiencia de usuario, se procede a presentar el resultado de la web desarrollada, en combinación con la plataforma IoT integrada en ella.

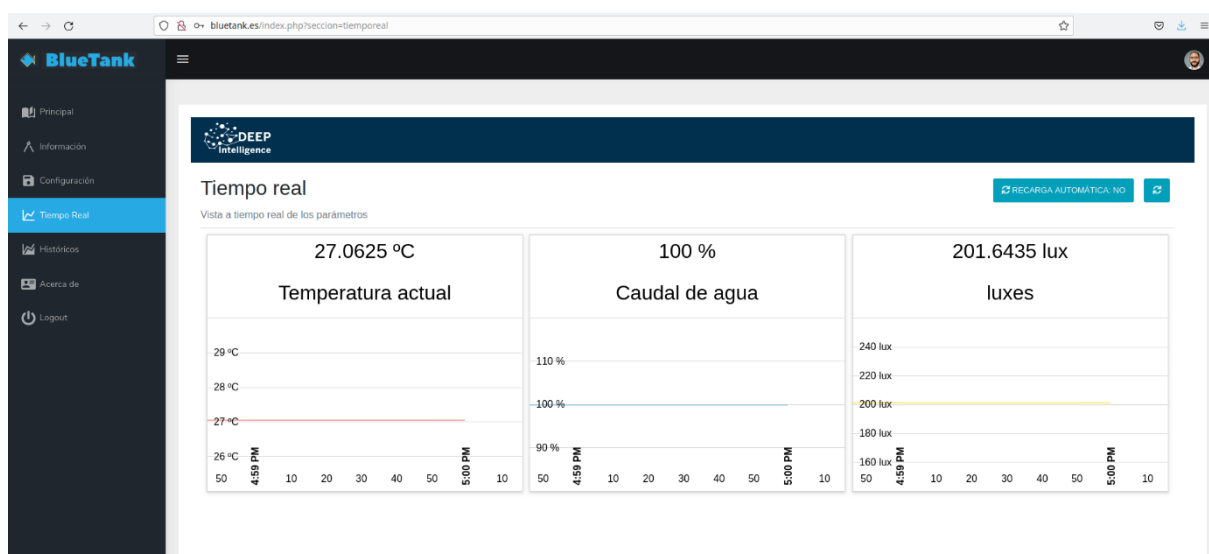


Ilustración 56. Visión en tiempo real

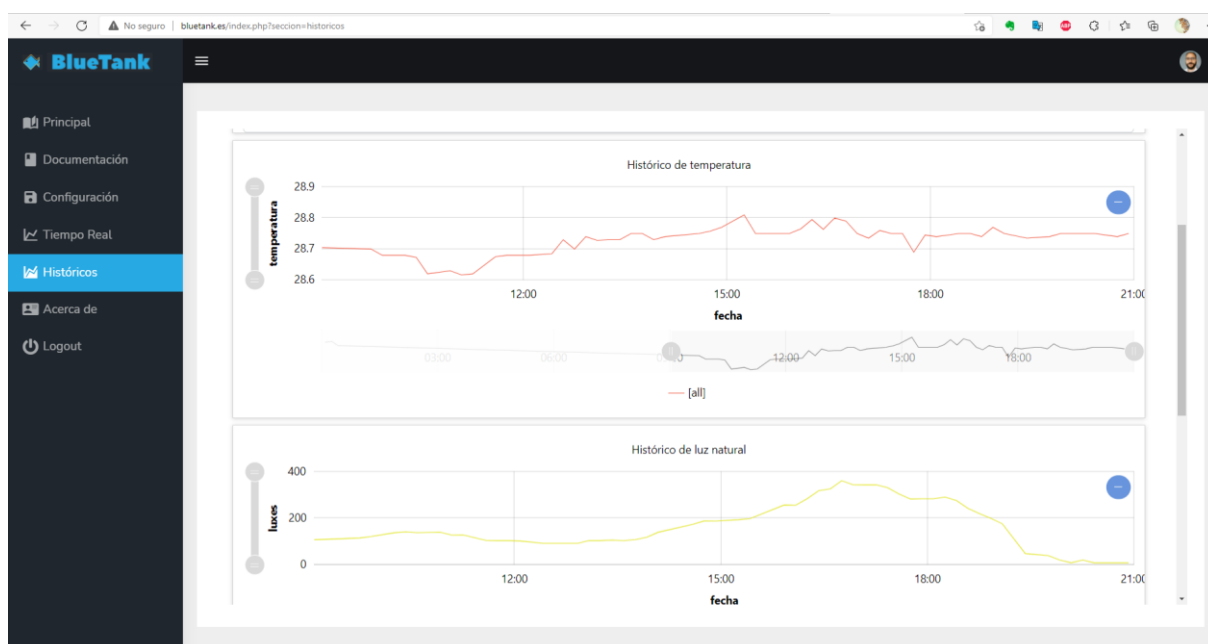


Ilustración 57. Visión datos históricos



Ilustración 58. Visualización de alerta

5. Conclusiones y trabajos futuros

La presente memoria permite concluir que la problemática inicial, encontrar un sistema de filtrado IoT para acuarios domésticos, ha quedado satisfecha. Como se indica en diferentes puntos de esta memoria, no es posible encontrar un sistema comercial que proporcione esta funcionalidad, o no a un precio asequible. En el proceso se ha obtenido un prototipo funcional de un sistema de filtrado IoT basado en el concepto ya existente de un filtro de cascada, que no solo permite el filtrado del agua si no que proporciona al usuario la capacidad de mantener un control fino sobre los parámetros fundamentales para la vida en el acuario. El prototipo obtenido cuenta con todos los elementos de una arquitectura IoT típica, desde la captura de datos, hasta su almacenamiento, presentación y posible análisis. Atendiendo más

directamente a cada uno de los objetivos específicos que permiten alcanzar el objetivo principal, determinamos que se ha diseñado y fabricado con éxito la PCB que contiene electrónica y el sistema de filtrado. La escalabilidad del sistema se ha incrementado con éxito a través del uso del protocolo MQTT y el sistema de base de datos no relacional. Por último, el sistema cuenta con una interfaz web, accesible desde cualquier dispositivo, que permitirá mejorar la experiencia de usuario, y se ha desarrollado un sistema de configuración sencilla que permite poner en marcha el dispositivo sin una complejidad elevada.

Todos estos aspectos validan el cumplimiento del objetivo inicial, determinando así el éxito del proyecto.

5.1.1. Líneas de trabajo futuras

El trabajo aquí presentado sienta unas bases sólidas sobre las que continuar trabajando, Así, en líneas de trabajo futuras, se debería partir de aquí para ampliar y mejorar el prototipo en diferentes aspectos, en primer lugar, se hace necesario contar con una remodelación de la electrónica que permita integrar todos los circuitos impresos en una sola PCB. En segundo lugar, se hace necesario implementar una API que permita la gestión de usuarios, utilizar un gestor de colas profesional para la transmisión de mensajes hasta la base de datos, por ejemplo, Apache Kafka, tratándose de dos elementos fácilmente integrables debido a la arquitectura modular planteada. Para una puesta en producción real, en un entorno distribuido, es necesario utilizar un sistema de encriptación de clave pública-privada que permita salvaguardar la integridad de los datos. Por último, como aficionado a la acuariofilia, sería interesante encontrar un producto que incluya un mayor número de parámetros monitorizados, principalmente el PH del agua.

Referencias bibliográficas

- Alonso Rollán, Sergio. (2019). BlueTank: Sistema de monitorización de acuario (Grado en Ingeniería Informática). Universidad de Salamanca, Salamanca.
- Rovitto, F., & Nisim, A. O. (2018). Tecnología sustentable IoT-Mobile: sistema de monitoreo y diagnóstico de acuarios. In XXI Concurso de Trabajos Estudiantiles (EST)-JAIIO 47 (CABA, 2018).
- Santiago, D., Pereyra, S., Lombera, G., & Urquiza, S. (2006). Analisis de defectos en soldadura por fricción-agitación mediante un modelado 3D. Mecánica Computacional, 2217-2226.
- Mankar, J., Darode, C., Trivedi, K., Kanoje, M., & Shahare, P. (2014). Review of I2C protocol. International Journal of Research in Advent Technology, 2(1).
- Elitech STC-1000PRO Controlador de temperatura multiusos de calefacción y refrigeración con termostato digital. (s. f.). Elitech UK. Recuperado 14 de septiembre de 2021, de <https://www.elitecheu.com/es/products/elitech-stc-1000-pro-digital-temperature-controller-thermostat?variant=40187921432729>
- KETOTEK Digital Termostato Enchufe Calentamiento Enfriamiento con Sonda 220V, Controlador de Temperatura Enchufe Momento Cuenta Regresiva para Incubadora Invernadero Acuario Reptil : Amazon.es: Bricolaje y herramientas. (s. f.). Amazon. Recuperado 14 de septiembre de 2021, de https://www.amazon.es/KETOTEK-Controlador-Temporizador-Calentamiento-Enfriamiento/dp/B085Q5BFJ8/ref=sr_1_20?mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=controlador+acuuario&qid=1621711402&sr=8-20
- Deep Intelligence. (s. f.). Deep Intelligence. Recuperado 14 de septiembre de 2021, de <https://deepint.net/>
- Grafana: The open observability platform. (s. f.-c). Grafana Labs. Recuperado 14 de septiembre de 2021, de <https://grafana.com/>
- Asociación Española de Normalización. (2019). Grados de protección proporcionados por las envolventes (Código IP) (UNE-EN 60529:2018/A2:2018/AC2019-02)
- Homepage. (s. f.). WrapPixel. Recuperado 14 de septiembre de 2021, de <https://www.wrappixel.com/>
- Bueche, F., & Hecht, E. (2005). Schaum's Outline of College Physics, 10th edition. McGraw-Hill Education.

- Ruiz, M. C. (2009). *Mecánica de Estructuras Libro I Resistencia de Materiales (Vol. 111)*. Univ. Politèc. de Catalunya.
- Corchado, J. M., Chamoso, P., Hernández, G., Gutierrez, A. S. R., Camacho, A. R., González-Briones, A., Pinto-Santos, F., Goyenechea, E., García-Retuerta, D., Alonso-Miguel, M., Hernandez, B. B., Villaverde, D. V., Sanchez-Verdejo, M., Plaza-Martínez, P., López-Pérez, M., Manzano-García, S., Alonso, R. S., Casado-Vara, R., Tejedor, J. P., ... Omatu, S. (2021). Deepint.net: A Rapid Deployment Platform for Smart Territories. *Sensors*, 21(1). <https://doi.org/10.3390/s21010236>
- de Rey García, D. (2017). *Controlador integral para acuario marino*. <https://e-archivo.uc3m.es/handle/10016/26837>
- Herrera Gil, L. (2019). *Diseño y desarrollo de una férula de miembro superior escaneada e impresa en 3D*. <http://uvadoc.uva.es/handle/10324/37756>
- Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January). MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*. <https://doi.org/10.1109/COMSWA.2008.4554519>
- Jingzhuo, W., & Chenglong, G. (2007, August). Research on 1-Wire Bus Temperature Monitoring System. *2007 8th International Conference on Electronic Measurement and Instruments*. <https://doi.org/10.1109/ICEMI.2007.4351019>
- Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. *Journal of Computer and Communications*, 03(05). <https://doi.org/10.4236/jcc.2015.35021>
- Pérez Laguarda, A. (2017). *Control y supervisión mediante un sistema microcontrolador de los parámetros de calidad de agua de un estanque*. <https://idus.us.es/handle/11441/61095#.YSqKxBol8wk.mendeley>
- Puyol Moreno, J. (2014). Una aproximación a Big Data. *Revista de Derecho de La UNED (RDUNED)*, 14. <https://doi.org/10.5944/rduned.14.2014.13303>
- Radouan Ait Mouha, R. A. (2021). Internet of Things (IoT). *Journal of Data Analysis and Information Processing*, 09(02). <https://doi.org/10.4236/jdaip.2021.92006>
- Yokotani, T., & Sasaki, Y. (2016, September). Comparison with HTTP and MQTT on required network resources for IoT. *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*. <https://doi.org/10.1109/ICCEREC.2016.7814989>

Anexo A. Descripción del sistema inicial

Este anexo pretende ilustrar el punto de partida de este trabajo de fin de estudios y sus diferencias con el resultado final. A pesar de que este punto inicial contaba con funcionalidad, es preciso recalcar, que el único aspecto reutilizado entre la queta inicial y la implementación final han sido los esquemas de conexión de los sensores. Los objetivos de este proyecto han implicado hacer un desarrollo totalmente novedoso en todas las áreas del trabajo.

A.1 ELECTRÓNICA:

El primer aspecto diferencial es que en el trabajo de fin de grado nunca se intentó crear un dispositivo físico de filtrado IoT, si no que se limitaba a una maqueta con funcionalidad montada en la parte superior del acuario.

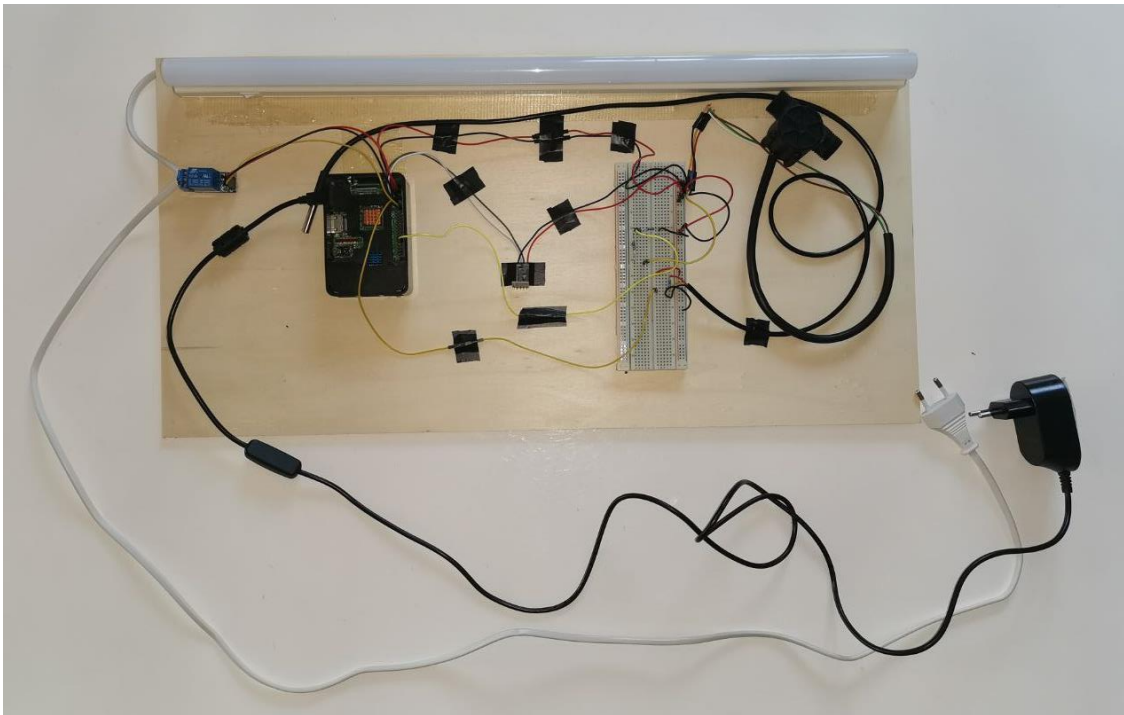


Ilustración 59. Prototipo Bluetank inicial

Este proyecto ha permitido desarrollar un prototipo funcional de sistema de filtrado IoT que incluya toda la electrónica integrada en una misma PCB.

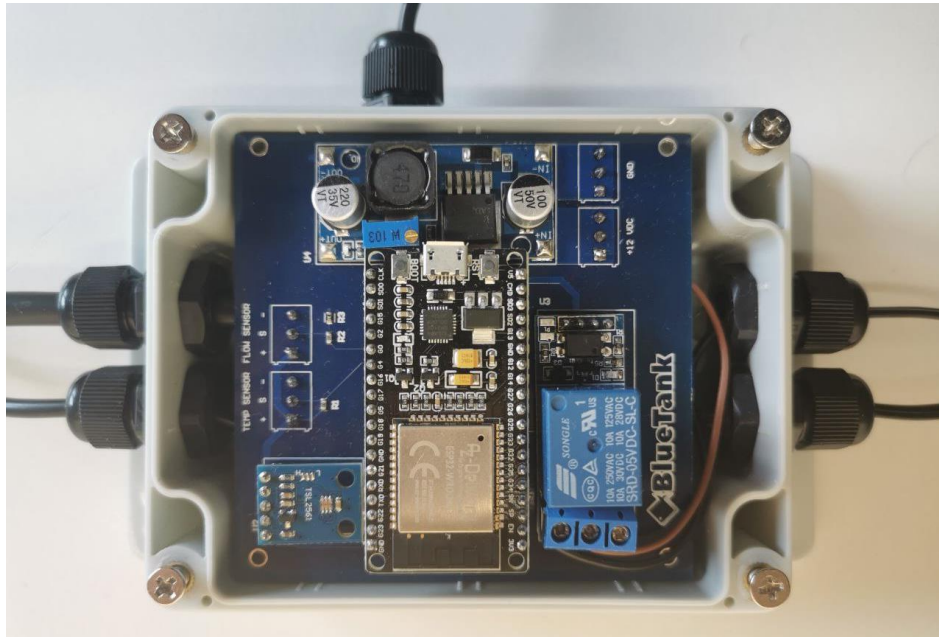


Ilustración 60. Electrónica Bluetank final

A.2 CREACIÓN DE UN DISPOSITIVO IOT

Otro aspecto diferencial es que la maqueta no hace ningún acercamiento al intento de introducir toda la electrónica en un único dispositivo como tampoco trata de presentar un sistema con resistencia IP. El rediseño de la electrónica y el cambio en la selección del microcontrolador ha implicado un cambio de lenguaje en el firmware y se ha realizado un desarrollo completamente nuevo en este apartado.



Ilustración 61. Dispositivo Bluetank final 1



Ilustración 62. Dispositivo Bluetank final 2

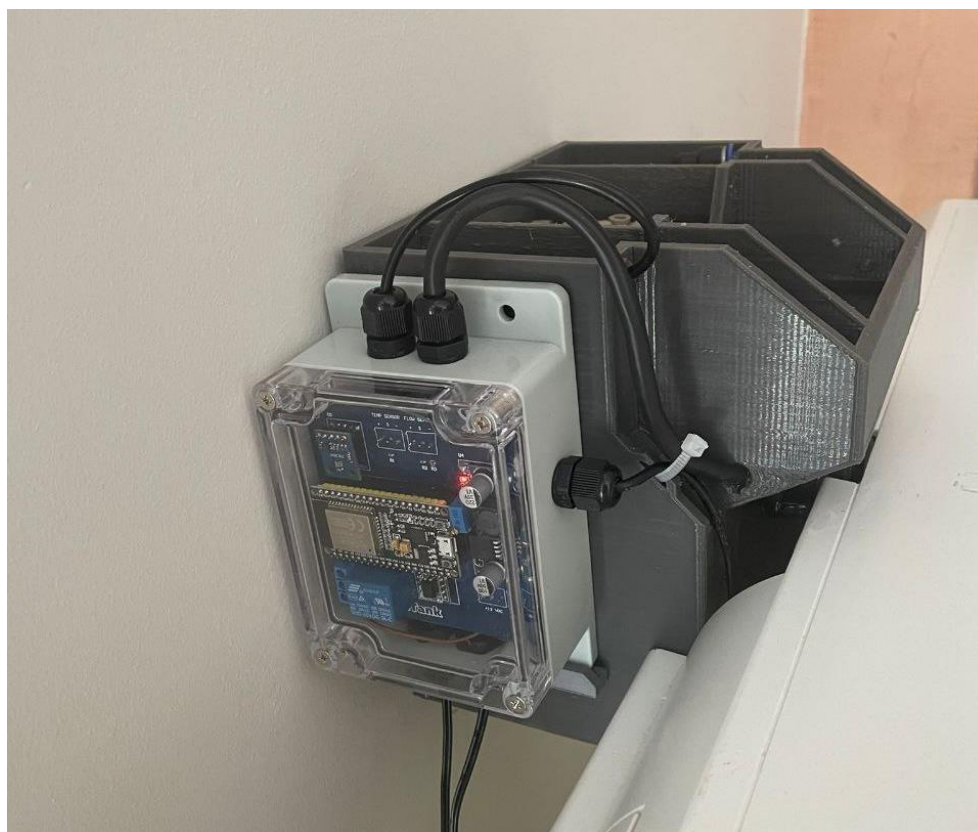


Ilustración 63. Dispositivo Bluetank montaje final detalle



Ilustración 64. Dispositivo Bluetank montaje final en acuario

A.3 ASPECTO VISUAL:

Respecto al apartado visual, la necesidad de mejorar la experiencia de usuario ha conllevado al uso de una plataforma IoT y al desarrollo de una página web completamente nueva. El sistema contaba con una web muy básica como se aprecia en las fotografías. Esta web ha sido rediseñada partiendo de cero y se ha integrado la plataforma IoT con el fin de proporcionar una experiencia de usuario agradable. La web antigua no contaba con un sistema de login salvaguardando la información privada del usuario, ni tampoco estaba disponible en un servidor, si no que estaba montada en un ordenador local.

El aspecto de la web inicial era este:

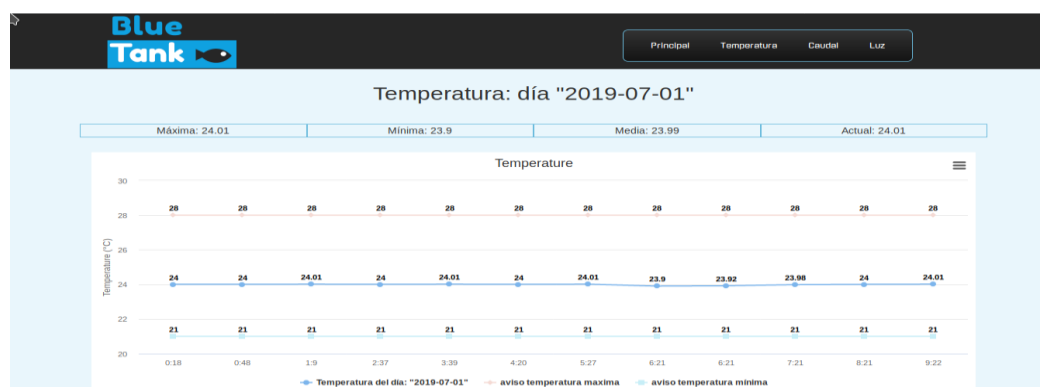


Ilustración 65. Interfaz Bluetank antigua 1

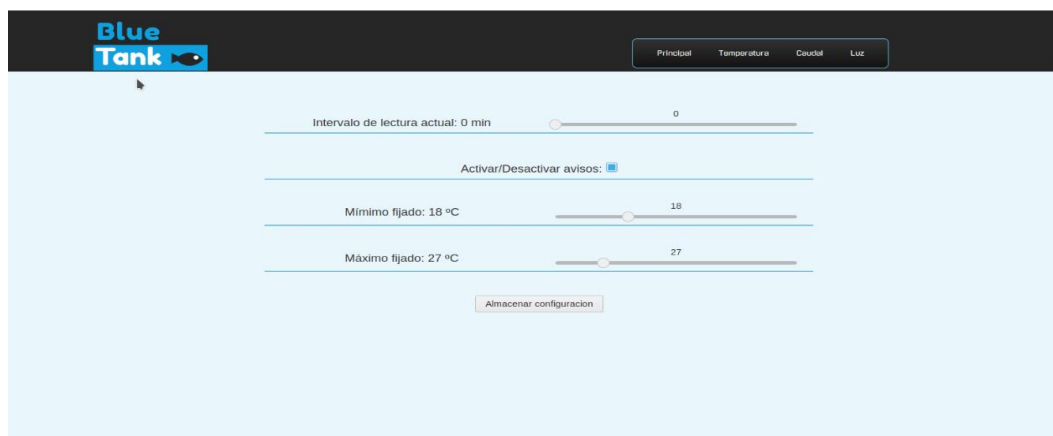


Ilustración 66. Interfaz Bluetank antigua 2

El aspecto visual una vez desarrollado el proyecto es el siguiente:

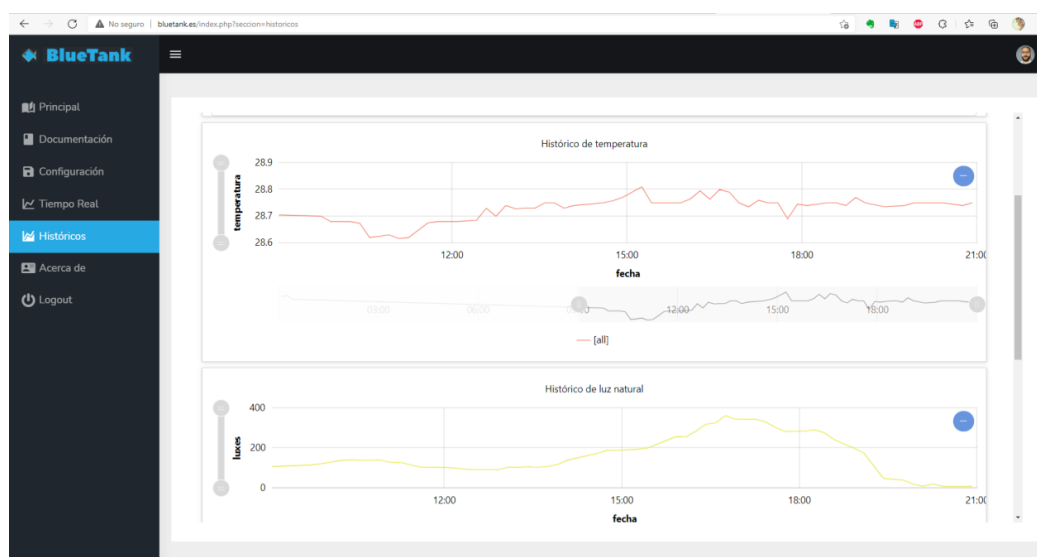


Ilustración 67. Interfaz Bluetank final. Vista históricos



Ilustración 68. Interfaz Bluetank final. Vista tiempo real

La interfaz es más limpia y cuidada, el logo se ha profesionalizado y se ha añadido un menú más estético y funcional que permite saber que pestaña está seleccionada. Los gráficos son interactivos y muestran el valor al pasar el puntero por encima. Además, se ha integrado la plataforma Deep Intelligence.

La nueva web, además permite un escalado para ser accesible desde móvil, en la que el menú se adaptará para permitir que la información de la web sea correctamente visualizada.

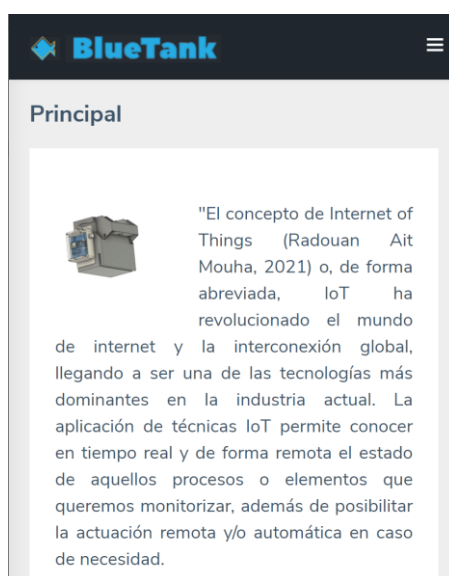


Ilustración 69. Interfaz Bluetank móvil. Vista tiempo real

También permite un escalado para pantallas de tamaño intermedio.

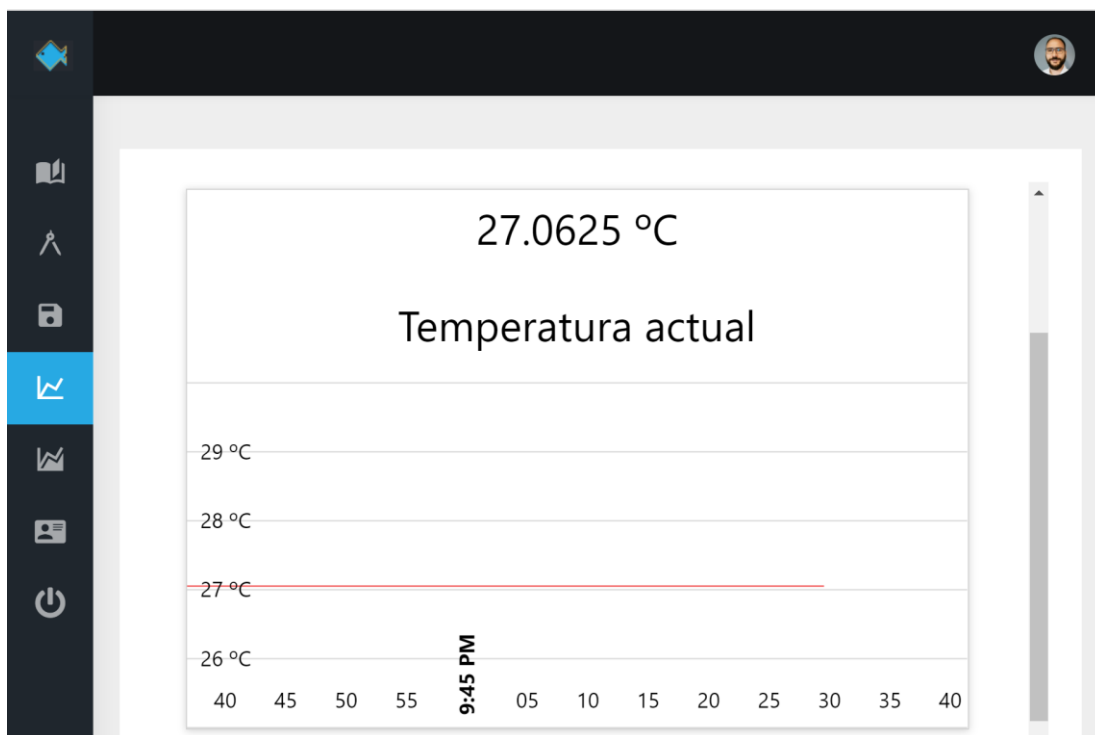


Ilustración 70. Interfaz Bluetank tablet. Vista tiempo real

A.4 ESCALABILIDAD:

La necesidad de escalabilidad ha implicado realizar un cambio en el protocolo de mensajes, evolucionando hacia el protocolo MQTT, mucho más recomendable en entornos puramente IoT. Este hecho ha obligado a la implementación de este protocolo en todos los elementos del sistema. Lo que ha implicado cambios en el servidor, que antes recibía las mediciones a través de una API REST. Esta API ha sido desechada y sustituida por un broker MQTT y un gestor de colas escrito en Python. Además, atendiendo a este mismo aspecto, se ha utilizado una base de datos no relacional, frente a la base de datos MySQL instalada anteriormente.

A.5 PLATAFORMA IOT:

Un elemento dispar entre el punto de partida y el punto final es el uso de una plataforma IoT que proporciona potentes herramientas de análisis de datos, siendo este un añadido importante. Un sistema de monitorización de datos carece de utilidad si estos datos no son utilizados para generar conocimiento y actuar en consecuencia.

La arquitectura del sistema ha sido modularizada de tal forma que sus componentes mantienen un bajo acoplamiento, permitiendo la evolución y sustitución de estos sin afectar al resto.

A.6 SISTEMA DE ALARMAS:

El sistema mostrará alertas visuales cuando alguno de los parámetros configurados desde la web salga de los rangos establecidos como óptimos por el usuario. Las alarmas serán visualizadas tanto en la vista de tiempo real como en la vista de históricos, además, se dedica una vista que recoge el histórico de alarmas del sistema, por último, el usuario recibirá un correo electrónico indicativo del tipo de alarma activada.

Anexo B. Manual de usuario web

Se trata de un anexo breve que pretende mostrar el mecanismo de acceso a la web de forma que el tribunal pueda comprobar el funcionamiento del proyecto y así evaluarlo correctamente.

La web cuenta con un sistema de DNS dinámico, siendo accesible a través de la url: www.bluetank.es

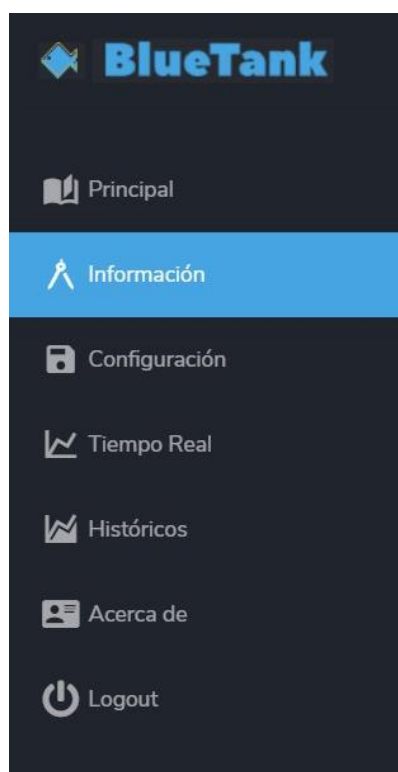
Accediendo a la url proporcionada el usuario encontrará la siguiente pantalla principal o pantalla de login en la que se muestra una animación sobre el funcionamiento del sistema de filtrado Bluetank. La propia web solicitará un nombre de usuario y una contraseña. Estos se han modificado para ser más simples y así facilitar la prueba del sistema.

- **Usuario:** sergioALRO

- **Contraseña:** 1234567



Una vez que se ha realizado el acceso a la plataforma, se proporciona un menú lateral que permite la navegabilidad a través de la página donde encontramos diferentes apartados:



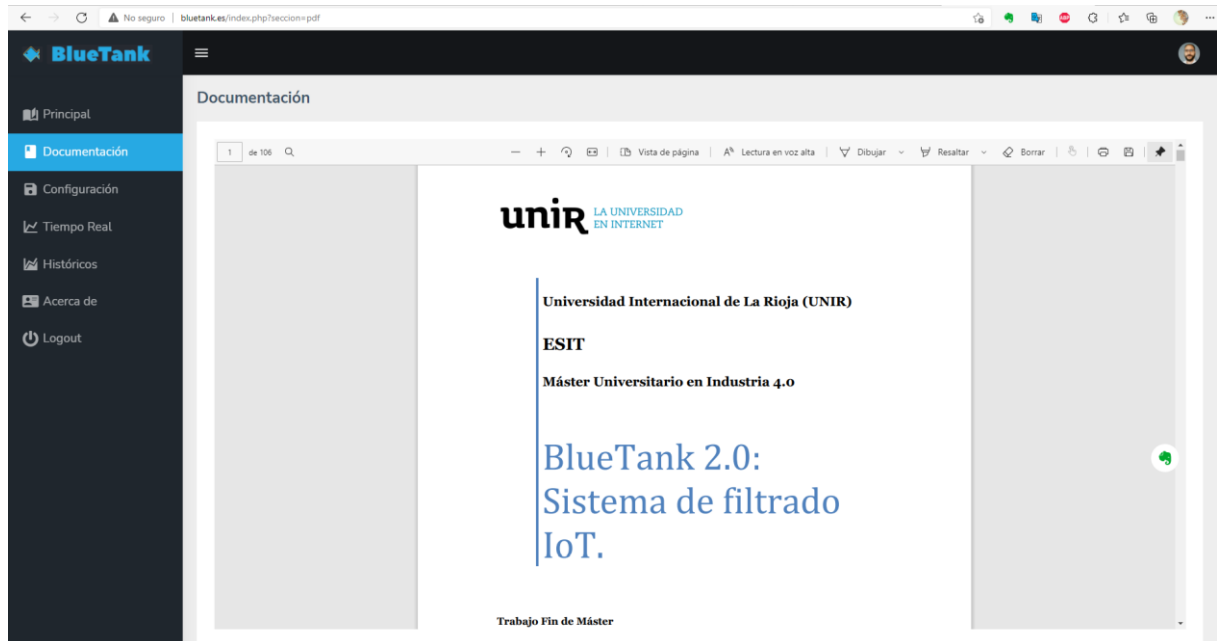
B.1 VISTA PRINCIPAL

El usuario encontrará una vista principal, en la que se realiza una introducción al ámbito del proyecto, poniendo al lector en situación e indicándole que es lo que puede encontrar. Además, muestra animaciones del dispositivo fabricado.



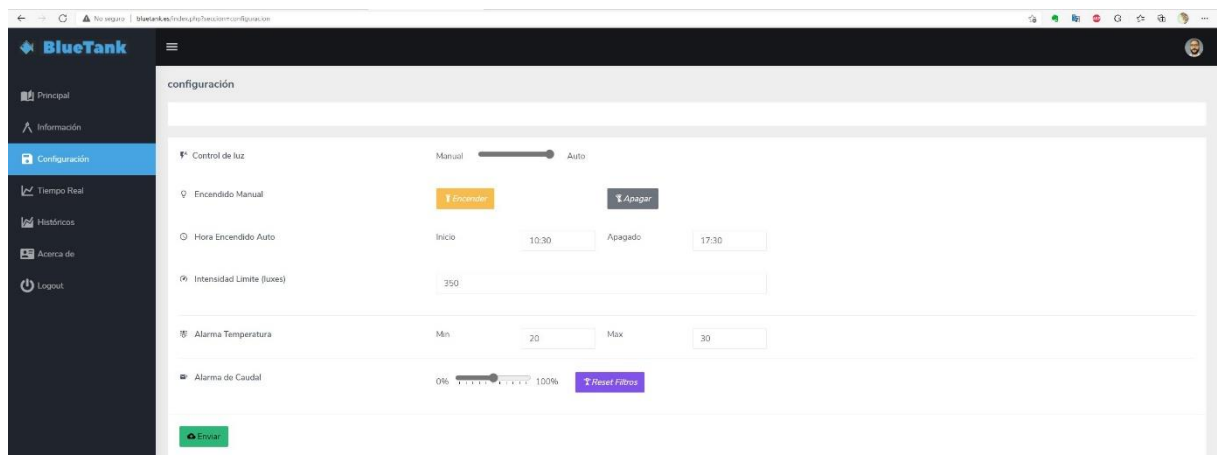
B.2 VISTA DE INFORMACIÓN

Muestra la información sobre la memoria que recoge los aspectos clave en el desarrollo del proyecto como son el diseño del filtro y el diseño de la PCB.



B.3 VISTA DE CONFIGURACIÓN

Este apartado de la web permite al usuario establecer la configuración de parámetros deseados y así, modificar el comportamiento del dispositivo y las alertas de la plataforma.



Para esta vista se detallán los aspectos configurables del sistema a través de esta vista de configuración.

- Control de luz: hace referencia a los modos de control de luz, manual, en la que el usuario envía las ordenes de encendido o apagado y automático en la que el

dispositivo decide cuando encender o apagar las luces en función de los horarios establecidos y el límite de luxes a partir del cual las luces se apagaran ya que se considera una cantidad de luz natural adecuada.

- Encendido manual: proporciona los botones para encendido y apagado de luces cuando el control de luz está en modo manual
- Hora encendido Auto: indica el horario dentro del cual el dispositivo encenderá las luces en modo automático, fuera de este horario las luces permanecerán apagadas.
- Intensidad límite (luxes): Cantidad de luxes límite. Por encima del límite, el dispositivo apaga las luces ya que se considera una cantidad de luz recibida adecuada, por debajo del límite el sistema encenderá la luz artificial para suplir la poca luz natural recibida.
- Alarma temperatura: Valores límite inferior y superior a partir de los cuales se hará saltar la alerta de temperatura.
- Alerta caudal: Valor límite a partir del cual se envía la alerta de necesidad de limpieza del filtro.
- Botón de reset: Establece la referencia de caudal que marcará el 100% de flujo de agua. Esta referencia se establece después de cada limpieza del filtro cuando el caudal es máximo.

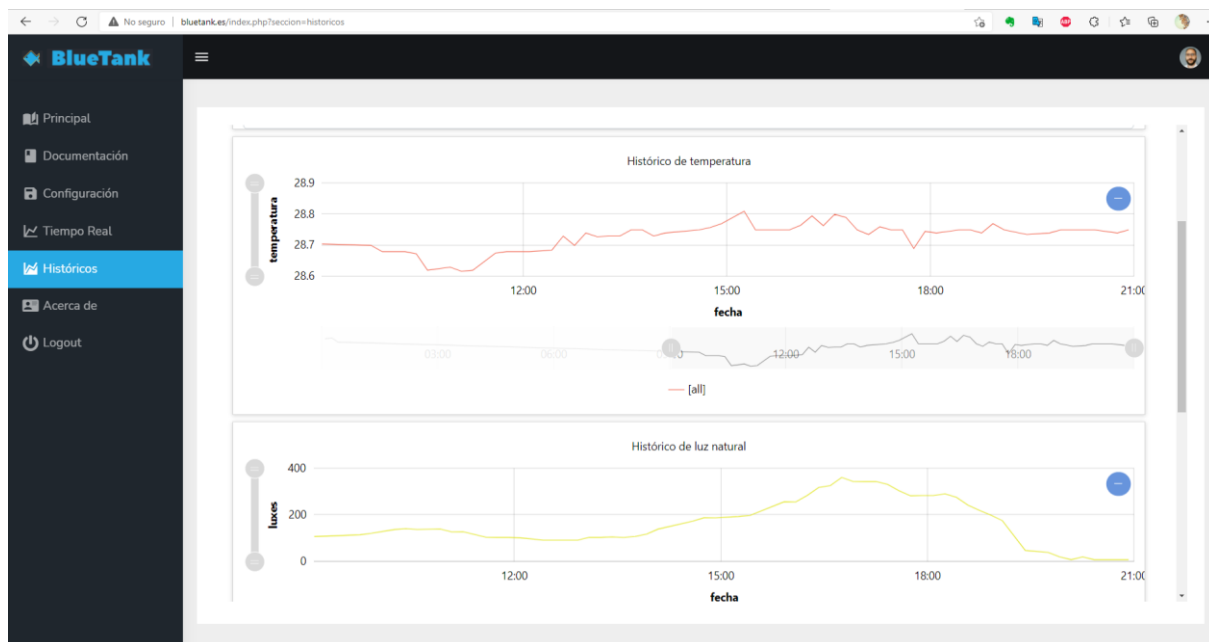
B.4 VISTA A TIEMPO REAL

Contiene la visualización en tiempo real de los datos recogidos por el dispositivo.



B.5 VISTA DE HISTÓRICOS

Contiene las visualizaciones de datos históricos recogidos por el dispositivo, permitiendo además seleccionar el rango de fechas que se desea visualizar.



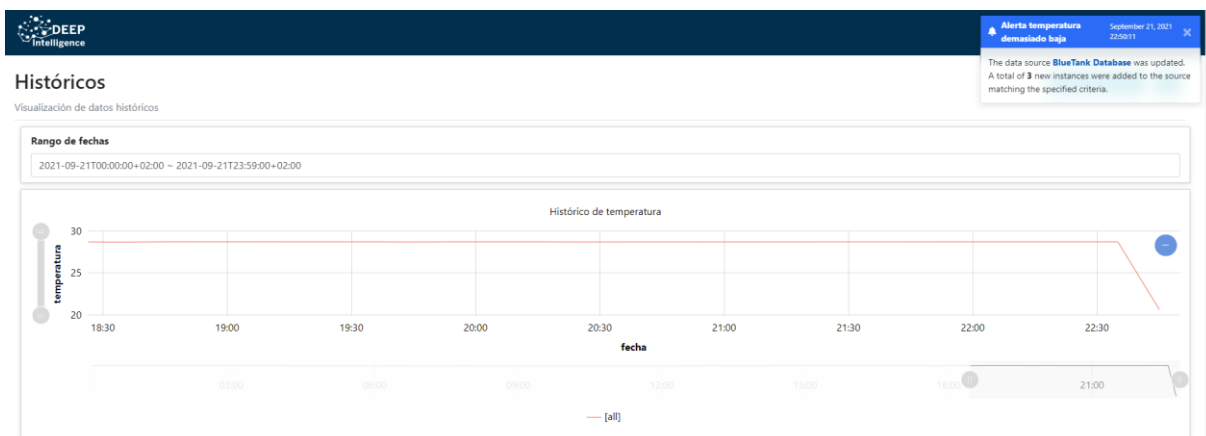
B.6 ALARMAS

Las alarmas son visualizables tanto en la vista de tiempo real como en la vista de históricos.

El sistema mostrará una alarma cuando la temperatura es demasiado elevada



El sistema mostrará una alarma cuando la temperatura es demasiado baja



El sistema mostrará una alarma cuando el caudal del filtro está próximo al límite establecido.



El sistema mostrará una alarma cuando el caudal del filtro alcance el límite establecido.



* Para generar las mediciones de caudal, por simplicidad, se ha establecido el límite al 100% de caudal.

Además de la alerta visual, el sistema envía una alerta al correo electrónico del usuario

Deep Intelligence - Alert report

 Deep Intelligence <no-reply@deepint.net>
22:57



Para: sergio.alrolln94@gmail.com

Deep Intelligence - Alert report

Hi, you are receiving this automated email because an alert you are subscribed to was triggered.

Alert triggered: [Alerta de caudal próximo a limpieza](#)

The data source [BlueTank Database](#) has been updated and we found 1 new instances that matched with the specified criteria. We show some of them below.

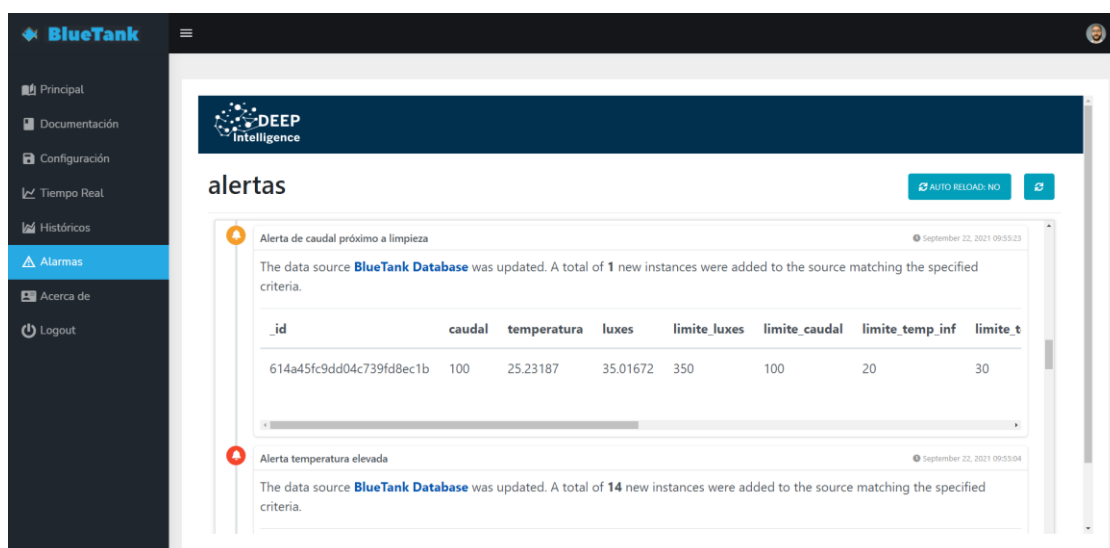
_id	caudal	temperatura	luxes	limite_luxes	limite_caudal	limite_temp_inf	limite_temp_sup	identificador	timestamp	alerta_caudal	alerta_caudal_prox	alerta_temp_max	alerta_temp_n
614a45fc9dd04c739fd8ec1b	100	25.23187	35.01672	350	100	20	30	B2:0C:23:20:C9:EE2	2021-09-21T20:52:05.036Z	true	true	false	false

If you want to unsubscribe, please follow this link: <https://app.deepint.net/workspace?ws=0000017b845452d2-cfef7d7f-e932c6e8-f466301b&s=alert&i=0000017ba102689a-6c21c5ed-4d474ba3-c6b7fecc>.

[Deep Intelligence](#) | [Terms of Service](#) | [Privacy Policy](#)

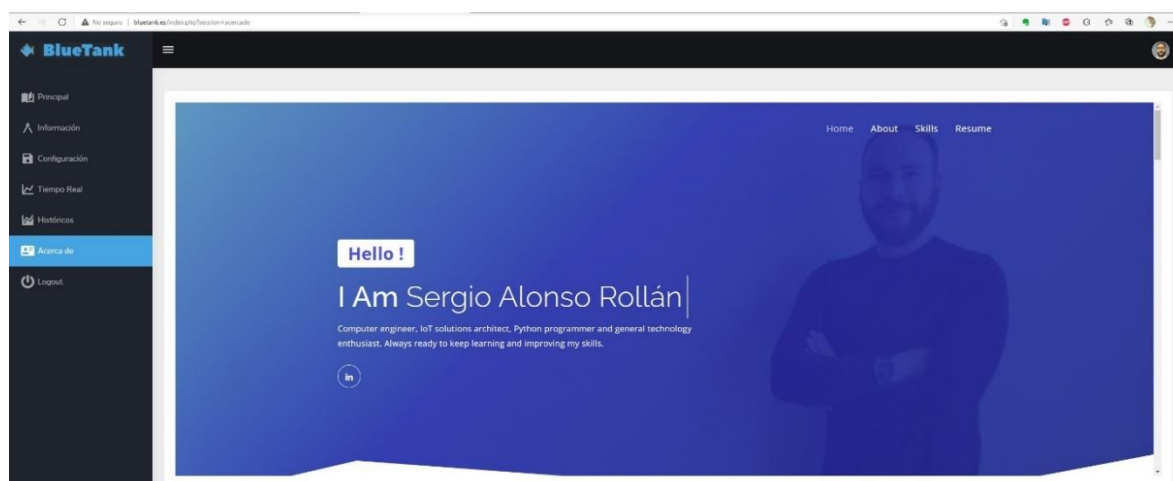
B.7 VISTA DE HISTÓRICO DE ALARMAS

Además de las alarmas que son reflejadas en las vistas de la web a tiempo real, la propia web cuenta con una vista que recoge el histórico de alarmas activadas en el sistema.



B.8 VISTA “ACERCA DE”:

Se trata de una vista que muestra la información profesional acerca del autor del trabajo.



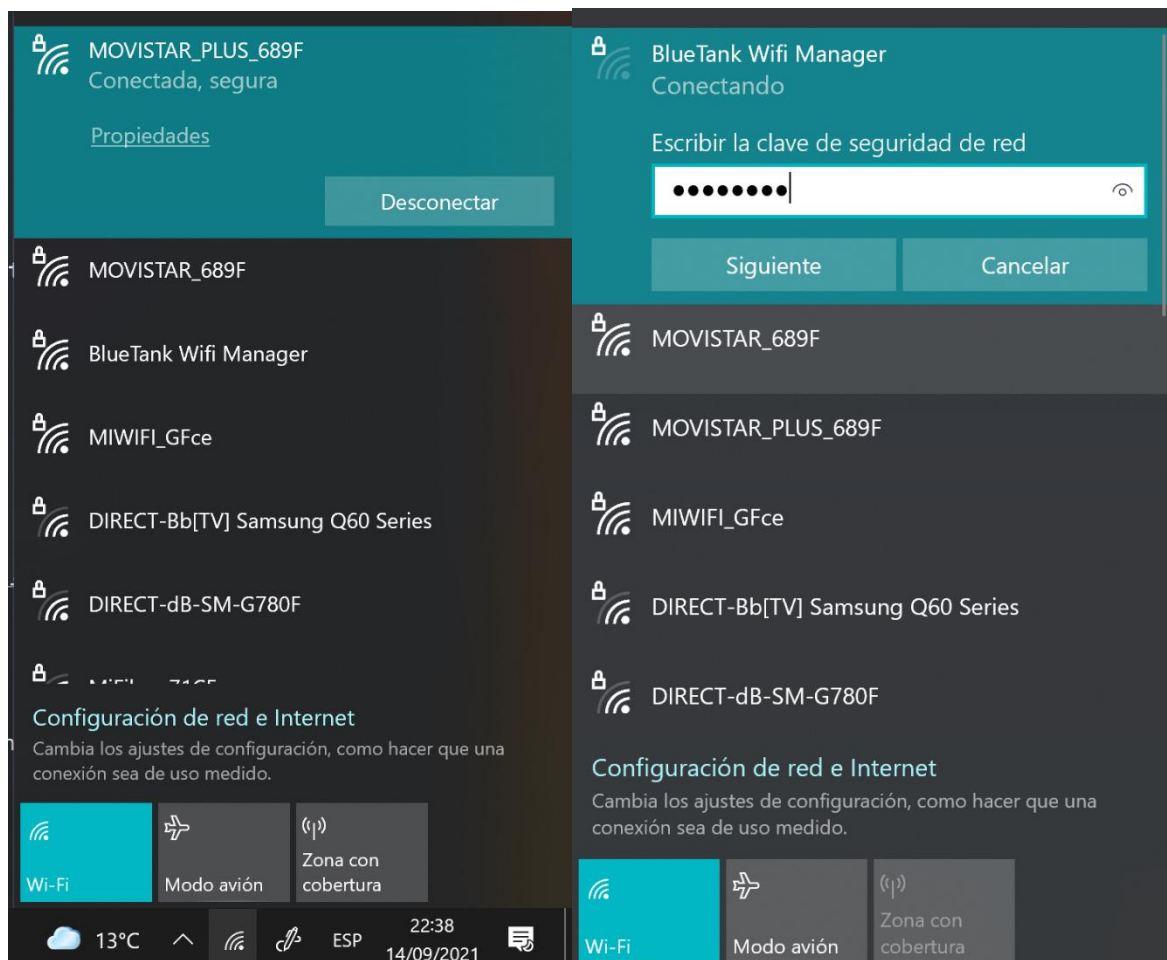
Anexo C. Sistema de configuración de dispositivo

El prototipo implementa un sistema de configuración sencilla y amigable para el usuario. El objetivo es que un usuario sin conocimiento informáticos avanzados sea capaz de realizar la configuración inicial del dispositivo.

El dispositivo, por su propia condición de microcontrolador, es capaz de arrancar de forma automática e iniciar la ejecución del código sin necesidad de intervención humana. No obstante, es necesario configurar la red wifi doméstica para que el dispositivo tenga acceso a internet.

C.1 CONEXIÓN A LA RED BLUETANK WIFI MANAGER

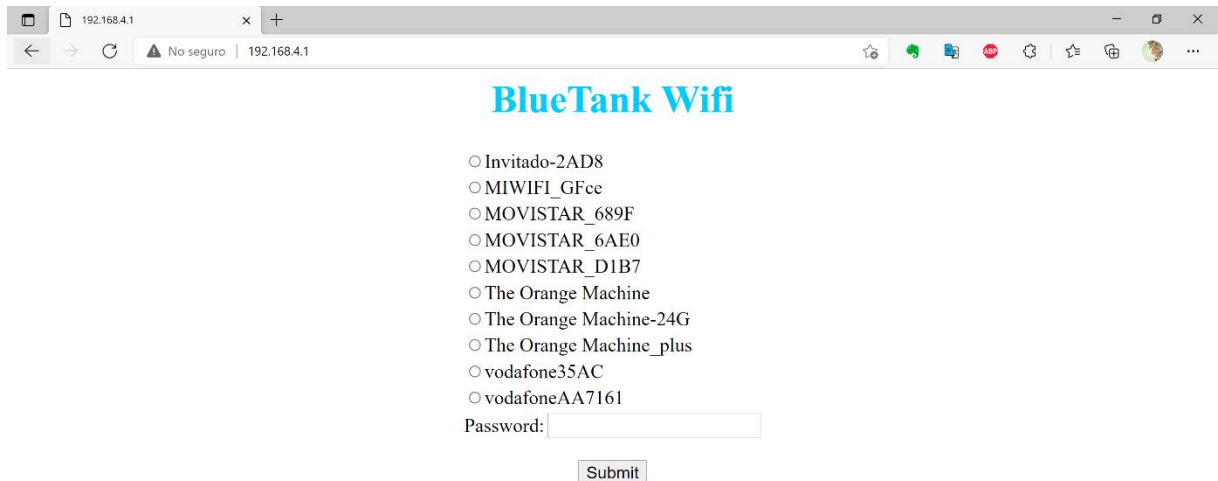
El dispositivo, al conectarse a la luz realizará una comprobación de la última red wifi almacenada, en caso de no existir datos, o esta red ser inalcanzable, levantará una red wifi local llamada “BlueTank Wifi Manager”, el usuario se conectará a esta red para realizar la configuración del dispositivo.



La contraseña predefinida es “12345678”, se trata de un aspecto que no afecta a la seguridad ya que esta red wifi solo se existirá durante el proceso de la configuración del dispositivo.

C.2 ACCESO A LA INTERFAZ WEB

El dispositivo BlueTank ofrece, a través de esta red, una interfaz web accesible desde cualquier navegador escribiendo en la barra de url la IP 192.168.4.1 tal y como se muestra en la imagen. El usuario deberá seleccionar su red wifi e introducir la contraseña correspondiente.



The screenshot shows a web browser window with the address bar set to 192.168.4.1. The page title is "BlueTank Wifi". It features a list of Wi-Fi networks with radio buttons for selection: Invitado-2AD8, MIWIFI_GFce, MOVISTAR_689F, MOVISTAR_6AE0, MOVISTAR_D1B7, The Orange Machine, The Orange Machine-24G, The Orange Machine_plus, vodafone35AC, and vodafoneAA7161. Below the list is a "Password:" label followed by a text input field and a "Submit" button.

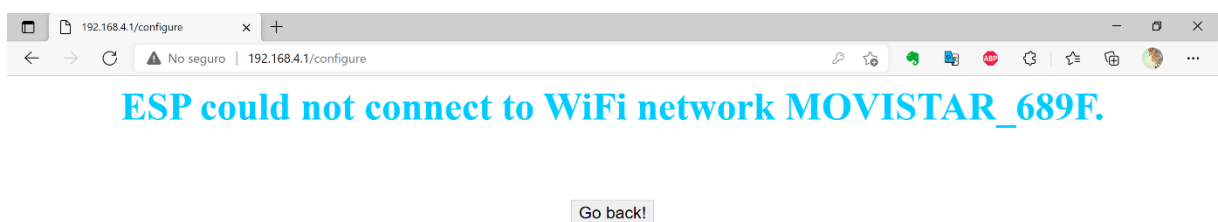
Your ssid and password information will be saved into the "wifi.dat" file in your ESP module for future usage. Be careful about security!

Some useful infos:

- BlueTank bluetank.es.
- This code available at tayfunulu/WiFiManager.

Acceso Correcto

Una vez introducida la contraseña y pulsado el botón "Submit", el dispositivo realizará un intento de conexión a la red. En caso de ser exitoso mostrará la siguiente pantalla.



The screenshot shows a web browser window with the address bar set to 192.168.4.1/configure. The page displays a large red error message: "ESP could not connect to WiFi network MOVISTAR_689F." Below the message is a "Go back!" button.

En caso contrario el dispositivo volverá a tener accesible la vista de configuración de red wifi.

Anexo D. Software desarrollado

El software desarrollado está repartido en tres ámbitos distintos, utilizando en cada caso el lenguaje de programación más adecuado.

D.1 FIRMWARE

En primer lugar, se encuentra el firmware, siendo este el elemento software más relevante y siendo el que permite dar vida al dispositivo físico Bluetank 2.0 Sistema de filtrado IoT. Como tal, ha sido ampliamente documentado en la memoria principal a través de diagramas de flujo, no obstante, merece la pena incluir el código en este anexo.

El firmware consta de diferentes archivos, entre los que se encuentran las bibliotecas utilizadas, que deben ser descargadas e incluidas en el microcontrolador. Una vez más, estas han sido descritas en la memoria en el punto 4. Desarrollo específico de la contribución.

No se mostrará el código de las bibliotecas utilizadas, tan solo merece la pena destacar, que la biblioteca “wifimmgr” ha sido modificada para incluir los colores y el nombre propios del proyecto Bluetank.

El firmware implementado está contenido en el archivo “boot.py” que será ejecutado al arranque del microcontrolador. Ha sido desarrollado en Micropython.

El firmware da comienzo con la inclusión de las bibliotecas necesarias

```
import wifimgr
import machine
import usocket as socket
import time
# Para I2C:
import tsl2561
from machine import Pin, SoftI2C
# Para 1wire:
import onewire, ds18x20
# Para TSL2561 (Luxes)
import tsl2561
# Para MQTT
from umqttsimple import MQTTClient
import ubinascii
import micropython
import esp
esp.osdebug(None)
import gc
gc.collect()
```

Después realiza la declaración de variables utilizadas, estas variables serán controlables por el usuario a través de la interfaz web. Una vez inicializadas con valores por defecto, estas tomarán el valor configurado por el usuario.

```
timestamp = 0
lux_init_hour = 9.00
lux_fin_hour = 16.00
hour = 0.0
lux_manual_control = 'OFF'
limit_lux = 200
num_seconds = 10 # Cada cuantos segundos se realiza el envio
caudal_porcentaje_limite = 0
tarado_filtro = True
limite_temp_max = 30
limite_temp_min = 20
alarma_temp = 0
```

Definidas las variables, el firmware procede a definir e inicia los objetos necesarios para las comunicaciones necesarias con los tres sensores y el relé.

```
# Control de relé
relay = Pin(33, Pin.OUT)
```

I2C, donde se define además la función de calibración del sensor.

```
# Inicialización de comunicación I2C
#-----
i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=100000)
lux_sensor = ts12561.TSL2561(i2c)

def calibracion_lux(lux): # Función de calibración
    return (lux + 0.7574) / 0.0836
```

Efecto Hall, donde se define la función de interrupción encargada de incrementar el contador de pulsos emitidos por el sensor, que posteriormente se transformarán a litros/hora utilizando la función de calibración proporcionada por el fabricante.

```
# Caudalímetro efecto Hall
#-----
counter_hall = 1
lectura = 0

def handle_interrupt(pin): # función de interrupción
    global counter_hall
    counter_hall += 1

hall_pin = Pin(0, Pin.IN, Pin.PULL_UP)
hall_pin.irq(trigger=Pin.IRQ_RISING, handler=handle_interrupt) # activo por flanco de subida
```

conexión 1Wire, donde se inicializan los objetos necesarios para la comunicación con el sensor ds18b20.

```
# Sensor de temperatura 1wire
#-----
onewire_pin = machine.Pin(4)
ds18b20 = ds18x20.DS18X20(onewire.OneWire(onewire_pin)) # crea el objeto 1wire
temp_sensor = ds18b20.scan()[0]
```

Una vez inicializadas las comunicaciones hardware el firmware inicializa la conexión con la red wifi. Para ello se utiliza la biblioteca "wifimgr", siendo esta una versión, no oficial, de la famosa biblioteca "WiFiManager" de Arduino, implementada en el repositorio de github "tayfunulu/ WiFiManager". El flujo de ejecución de esta biblioteca queda definido en los diagramas de flujo de la memoria principal, no obstante, se adjunta el código correspondiente.

```
# Configuración red Wifi
#-----
# Si la biblioteca puede conectar al wif esta parte se ignora
wlan = wifimgr.get_connection()
if wlan is None:
    machine.reset()
    print("Could not initialize the network connection.")

try: # crea el socket para escuchar en la red 192.168.4.1
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind(('', 80))
    s.listen(5)
except OSError as e:
    machine.reset()
```

Realizada la conexión a la red wifi el siguiente paso es inicializar la comunicación con el broker MQTT, se da comienzo declarando el host, usuario y contraseña del broker y los topics utilizados para la comunicación.

```
# +-----+
# |  MQTT CONFIGURACION  |
# +-----+
mqtt_server = 'bluetank.es'
user_mqtt = 'BlueTank_user_B20C2320C9EE'
pass_mqtt = 'EE9C0232C02B'

client_id = ubinascii.hexlify(machine.unique_id())

topic_time = b'Bluetank/Timestamp' # Para obtener el timestamp
topic_lux_control = b'Bluetank/Lux' # para recibir el control de encendido/apagado de luces. Formato: 'ON' o 'OFF'
topic_init_hour = b'Bluetank/Init_H' # Para recibir la hora de inicio de control de luces. Formato: 'HH:mm'
topic_fin_hour = b'Bluetank/Fin_H' # Para recibir la hora de fin de control de luces. Formato: 'HH:mm'
topic_limit_lux = b'Bluetank/Lux_limit' # Para recibir el valor de luxes a partir del cual apagamos la luz.
Formato: '135'
#(numero en
formato string)
topic_percentage_caudal = b'Bluetank/Caudal' # Para recibir el % de caudal a partir del cual se enviara una
alerta.
topic_hour = b'Bluetank/hour' # para recibir la hora en formato HH:mm
topic_tarado_filtro = b'Bluetank/Tarado_filtro' # ejecuta el tarado del filtro.
topic_auto_manual = b'Bluetank/AUTO-MANUAL' # Detección de modo automático o manual.
topic_temperatura_maxima = b'Bluetank/Temp_Max' # Valor límite superior de alarma de temperatura.
topic_temperatura_minima = b'Bluetank/Temp_Min' # Valor límite inferior de alarma de temperatura.
topic_pub = b'Bluetank/Meditations' # Para enviar las mediciones obtenidas al servidor.
```

A continuación, se define la función de interrupción ejecutada cada vez que se reciba un mensaje MQTT, esta función es la encargada de almacenar las configuraciones

correspondientes provenientes de la web. El funcionamiento de esta función queda definido en el diagrama de flujo adjunto a esta memoria.

```
def sub_cb(topic, msg):
    global timestamp
    global lux_manual_control
    global lux_init_hour
    global lux_fin_hour
    global limit_lux
    global hour
    global caudal_porcentaje_limite
    global limite_temp_min
    global limite_temp_max
    global lux_manual_control

    # timestamp
    if topic == topic_time:
        timestamp = msg.decode('utf8')
        # get hour from timestamp
        try:
            hour = timestamp.split(" ")[1]
            hour = float(hour.split(":")[0] + "." + hour.split(":")[1])
            print(hour)
        except:
            print("error parse hour")
```

```
# Establecer control manual/automático
if topic == topic_lux_control:
    if msg.decode('utf8') == "MANUAL":
        lux_manual_control = 'ON'
    if msg.decode('utf8') == "AUTO":
        lux_manual_control = 'OFF'

# control manual de luces
if topic == topic_lux_control and lux_manual_control == 'ON':
    if msg.decode('utf8') == "ON":
        relay.value(1)
    if msg.decode('utf8') == "OFF":
        relay.value(0)

# lux init hour
if topic == topic_init_hour:
    #try:
        lux_init_hour = msg.decode('utf8')
        lux_init_hour = float(lux_init_hour.split(":")[0] + "." + lux_init_hour.split(":")[1])
        print(lux_init_hour)
    #except:
        #print("error parse init hour")

# lux fin hour
if topic == topic_fin_hour:
    try:
        lux_fin_hour = msg.decode('utf8')
        lux_fin_hour = float(lux_fin_hour.split(":")[0] + "." + lux_fin_hour.split(":")[1])
        print(lux_fin_hour)
    except:
        print("error parse hour")
```

```
# lux fin hour
if topic == topic_fin_hour:
    try:
        lux_fin_hour = msg.decode('utf8')
        lux_fin_hour = float(lux_fin_hour.split(":")[0] + "." + lux_fin_hour.split(":")[1])
        print(lux_fin_hour)
    except:
        print("error parse hour")

# lux limit
if topic == topic_limit_lux:
    limit_lux = int(msg.decode('utf8'))
    print(limit_lux)

# caudal %
if topic == topic_percentage_caudal:
    caudal_porcentaje_limite = int(msg.decode('utf8'))
    print(caudal_porcentaje_limite)

# Tarado caudal
if topic == topic_tarado_filtro:
    tarado_filtro = True
    print("Tarado filtro: ", tarado_filtro)

# Límite temperatura máxima
if topic == topic_temperatura_maxima:
    limite_temp_max = int(msg.decode('utf8'))

# Límite temperatura mínimo
if topic == topic_temperatura_minima:
    limite_temp_min = int(msg.decode('utf8'))
```

El último paso para terminar de realizar la comunicación MQTT es realizar la suscripción a los topics definidos.

```
def connect_and_subscribe():
    global client_id, mqtt_server, topic_sub
    client = MQTTClient(client_id, mqtt_server, user=user_mqtt, password=pass_mqtt, port=8883)
    client.set_callback(sub_cb)
    client.connect()
    client.subscribe(topic_time, qos = 0)
    client.subscribe(topic_lux_control, qos = 0)
    client.subscribe(topic_init_hour, qos = 0)
    client.subscribe(topic_fin_hour, qos = 0)
    client.subscribe(topic_limit_lux, qos = 0)
    client.subscribe(topic_percentage_caudal, qos = 0)
    client.subscribe(topic_tarado_filtro, qos = 0)

    try:
        client = connect_and_subscribe()
    except OSError as e:
        restart_and_reconnect()
```

El último elemento del firmware es el bucle infinito de ejecución, en el que entrará el firmware para realizar la lectura de sensores, control de luces y el envío de datos. Una vez más, el funcionamiento de este bucle queda descrito en el diagrama de flujo incluido en la memoria principal. A pesar de eso, en este anexo se detallarán algunos elementos importantes.

El bucle da comienzo con la comprobación de mensajes vía MQTT y la espera no ocupada de un segundo. Esta espera es necesaria para la lectura del caudalímetro, que necesita acumular pulsos durante un segundo para obtener la frecuencia que después transformar a litros/hora.

```
while True:
    client.check_msg()
    time.sleep(1) # sleep 1 second
```

A continuación, se realiza la lectura de los tres sensores y el valor obtenido se acumula para realizar una media.

```
# lectura del sensore de efecto hall (durante 1 segundo)
litros_hora = counter_hall / 16 * 120
litros_hora_media += litros_hora

# lectura del sensor de temperatura
temp = 0
ds18b20.convert_temp()
temp = ds18b20.read_temp(temp_sensor)
temperatura_media += temp

# lectura de iluminancia
lux = calibracion_lux(int(lux_sensor.read()))
luxes_media += lux
```

El siguiente paso es, en función de la configuración obtenida a través de MQTT, procesada en la función "sub_cb()" ya descrita, se realiza el control de luces del acuario.

- Si estamos fuera del horario de encendido de luces y el dispositivo está en modo automático, las luces se apagarán.
- Si estamos dentro del horario de encendido de luces, el dispositivo está en modo automático y la luz natural recibida es superior al límite establecido por el usuario, las luces se apagarán ya que el acuario recibe luz suficiente de forma natural.
- Si estamos dentro del horario de encendido de luces, el dispositivo está en modo automático y la luz natural recibida es inferior al límite establecido por el usuario, las luces se encenderán ya que el acuario no recibe luz suficiente de forma natural.

```
# Control de luces
if ((hour < lux_init_hour or hour > lux_fin_hour) and lux_manual_control == "OFF"):
    # Si estamos fuera del horario, y el control de luces NO esta en manual
    relay.value(0) # Apagar luces
elif ((hour > lux_init_hour and hour < lux_fin_hour) and lux_manual_control == "OFF" and lux > limit_lux):
    # Si estamos en el horario de luces, NO estamos en control manual y la luz natural supera el limite:
    relay.value(0) # Apagar luces
elif ((hour > lux_init_hour and hour < lux_fin_hour) and lux_manual_control == "OFF" and lux < limit_lux):
    # Si estamos en el horario de luces, NO estamos en control manual y la luz natural NO supera el limite:
    relay.value(1) # Encender luces
```

Por último, cada 10 segundos, el firmware realiza el tarado, realiza las medias de las mediciones obtenidas, genera el JSON de envío y lo envía.

```
# every 10 seconds
if (times >= num_seconds):

    litros_hora = counter_hall/num_seconds / 16 * 120
    litros_hora_media += litros_hora

    # Calcular media caudal
    caudal = litros_hora_media / num_seconds
    print("caudal: ", caudal)
    # Tarado del filtro:
    if (tarado_filtro == True):
        tarado_filtro = False
        caudal_100 = caudal
        print("caudal_tarado", caudal_100)

    # calcular % caudal
    caudal_porcentaje = caudal * 100 / caudal_100
    litros_hora_media = 0
    counter_hall = 1 # effect hall counter

    # calcular media temp
    temperatura = temperatura_media / num_seconds
    temperatura_media = 0

    # calcular media lux
    lux = luxes_media / num_seconds
    luxes_media = 0

times = 0
try:
    msg = '{\'caudal\' : ' + str(caudal_porcentaje) + ', ' + '#% litros_hora % temp_sensor'
    msg += '\\'temperatura\' : ' + str(temperatura) + ', '
    msg += '\\'luxes\' : ' + str(lux) + ', '
    msg += '\\'limite_luxes\' : ' + str(limit_lux) + ', '
    msg += '\\'limite_caudal\' : ' + str(caudal_porcentaje_limite) + ', '
    msg += '\\'alarma_temp\' : ' + str(alarma_temp) + ', '
    msg += '\\'identificador\' : ' + '\B2:0C:23:20:C9:EE\' + ', '
    msg += '\\'timestamp\' : \'' + str(timestamp) + '\''
    msg += '}'
    msg = bytes(msg, 'utf-8')
    print(msg)
    client.publish(topic_pub, msg)
except OSError as e:
    restart_and_reconnect()

times += 1
```

D.2 CLOUD

Otro elemento de este sistema es el cloud que contiene el almacenamiento, este cloud proporcionará a los dispositivos un servidor NTP vía MQTT mediante el que podrán conocer la fecha y hora. Además, contiene un gestor de colas encargado de recoger las mediciones

provenientes de los dispositivos a través del bróker Mosquitto y encaminarlos hacia la base de datos no relacional MongoDB. Ambos servicios están implementados en Python 3.

D.2.1 Servidor NTP

El servidor NTP está desarrollado en Python, proporciona la funcionalidad de un servidor ntp al uso, pero en este caso lo hace a través del protocolo mqtt, encargado de proporcionar la fecha y hora actualizadas a los dispositivos Bluetank, de tal forma que estos puedan obtener una marca temporal que asignar a cada una de las mediciones emitidas.

Se utilizan las bibliotecas "paho.mqtt.client" para la comunicación MQTT, "datetime" para trabajar con fechas y "time" para realizar las esperas no ocupadas.

El script entra en un bucle infinito donde cada segundo actualiza la hora.

```
import paho.mqtt.client as mqtt
import datetime
import time

topic = "Bluetank/Timestamp"

client = mqtt.Client()
client.username_pw_set(username="BlueTank_user_B20C2320C9EE",password="EE9C0232C02B")
client.connect("192.168.1.152", 1883, 60)

while True:
    ct = datetime.datetime.now()
    client.publish(topic, str(ct), qos=0)
    time.sleep(10)
```

D.2.2 Gestor de colas

El gestor de colas está contenido en el script queue_manager.py, que está corriendo como daemon en el cloud de Bluetank. Es el encargado de recoger las mediciones provenientes de los dispositivos para encaminarlos a la base de datos. Al momento de escribir los datos en la base de datos, estos se almacenarán en la "collection" correspondiente al identificador del dispositivo, en este caso la MAC.

Este script utiliza la biblioteca paho.mqtt.client para la comunicación con el broker Mosquitto. En y la biblioteca "pymongo" para la comunicación con la base de datos.

En primer lugar, crea el objeto correspondiente y realiza la conexión con el servidor.

```
import paho.mqtt.client as mqtt
import json
import datetime
import time
import pymongo

topic = "Bluetank/Meditations"
mongodb_client = pymongo.MongoClient('localhost', 27017)

client = mqtt.Client()
client.username_pw_set(username="BlueTank_user_B20C2320C9EE",password="EE9C0232C02B")
client.connect("192.168.1.152", 1883, 60)
```

A continuación, define la función de interrupción que se ejecutará cada vez que se reciba un mensaje. Esta función es la encargada de encaminar los datos a la base de datos.

```
client.on_message = on_message
```

La función queda definida de la siguiente manera.

```
def on_message(client, userdata, message):
    global topic
    if (message.topic == topic):
        data = message.payload.decode('utf8').replace("'", '')
        payload = json.loads(data)
        print(payload)
        db = mongodb_client['BlueTank']
        collection = db[payload['identificador']]
        id = collection.insert_one(payload).inserted_id
```

Para finalizar, el script se suscribe al topic definido y arranca el bucle infinito de escucha por mqtt.

```
client.subscribe(topic, qos=0)
client.loop_forever()
```

Según la arquitectura modular planteada, este script podrá ser sustituido por un gestor de colas profesional como Apache Kafka u otros. De esta manera se logrará una escalabilidad total y se ahorran costes de mantenimiento.

D.2.3 Desarrollo web.

La plataforma web está incluida en el cloud de Bluetank, sin embargo, por importancia, se le asigna un apartado específico. La web ha sido descrita en la memoria. A pesar de eso se dedica una breve explicación más detallada. No se incluirá el código de la web al tratarse de numerosos archivos, carpetas, subcarpetas, etc. A pesar de esto, la web es accesible como se indica en el Anexo B. Manual de usuario.

Para la implementación de esta web se ha utilizado esta plantilla gratuita de la plataforma WrapPixel (Homepage, s. f.). La plantilla proporciona un menú limpio y funcional que incluye la posibilidad de modificarse. Este elemento será el encargado de proporcionar la navegabilidad de la web, permitiendo acceder a las diferentes páginas que lo componen.

Esta plantilla ha sido modificada con el logotipo y los colores propios de Bluetank de tal forma que quede integrada completamente con el resto de los elementos del proyecto como son la PCB o el firmware. El contenido de la página web ha sido completamente modificado para contener los elementos propios de este proyecto, como son la introducción, la documentación y la configuración.

Se ha implementado un sistema de login con el fin de salvaguardar la información del usuario. Se trata de un sistema de login simple que solo almacena un usuario.

Por otra parte, para dar funcionalidad a la vista de configuración, se ha implementado, en el lenguaje PHP, un sistema de comunicación MQTT que permite comandar de forma remota el dispositivo.

Por último, la web integra la plataforma IoT y Big Data llamada Deep Intelligence. La integración de esta plataforma se ha realizado a través de iframes, esta integración se utiliza para la vista en tiempo real, la vista de históricos y la vista de histórico de alarmas. Además, se incluye una cuarta vista basada en iframe que incluye mi portfolio personal.

Anexo E. Diseño electrónica

Este anexo incluye los resultados del diseño de la electrónica. Estos han sido incluidos en la memoria, sin embargo, aquí se encuentran más detallados.

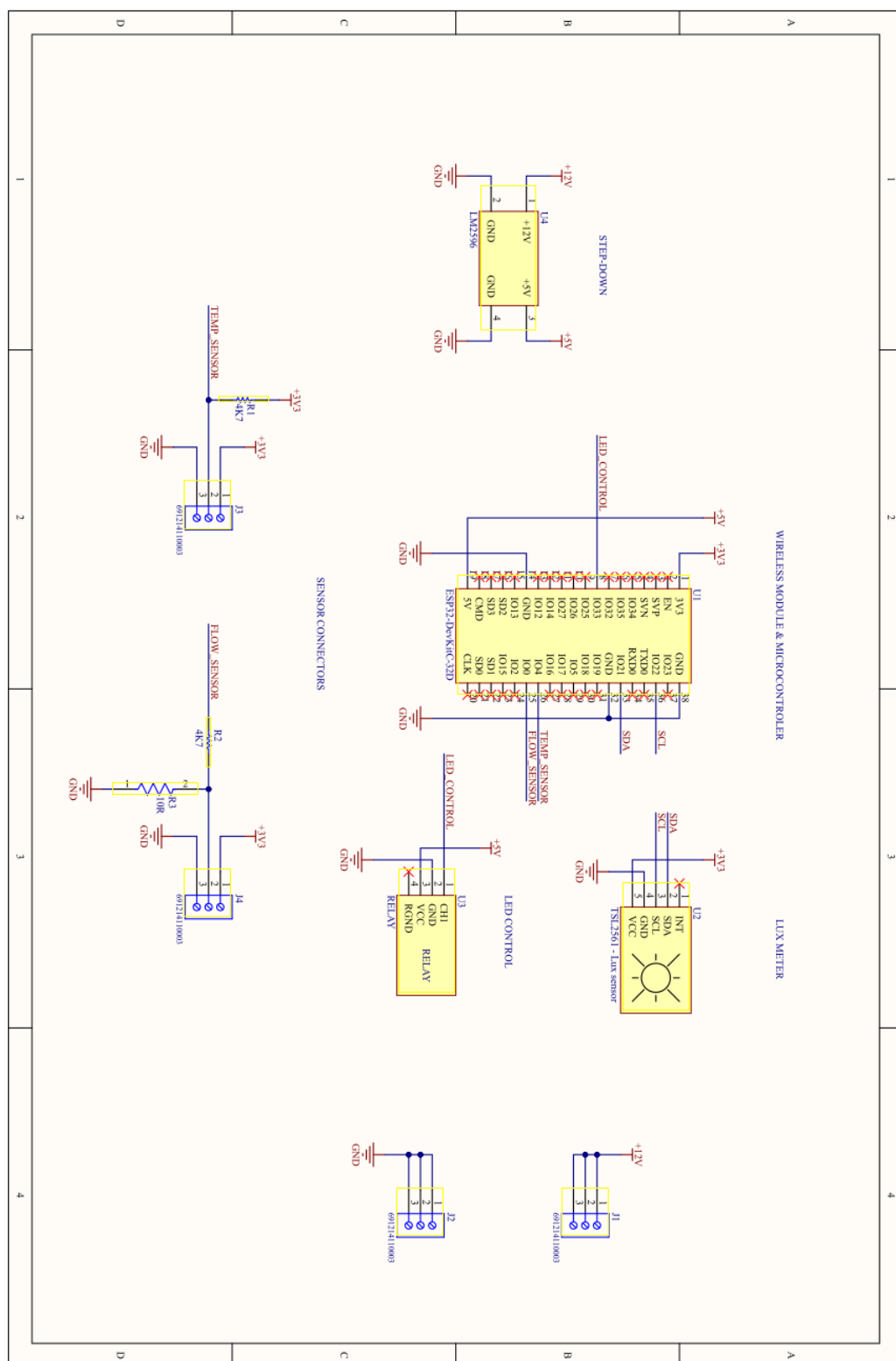


Ilustración 71. Esquemático del diseño de la PCB

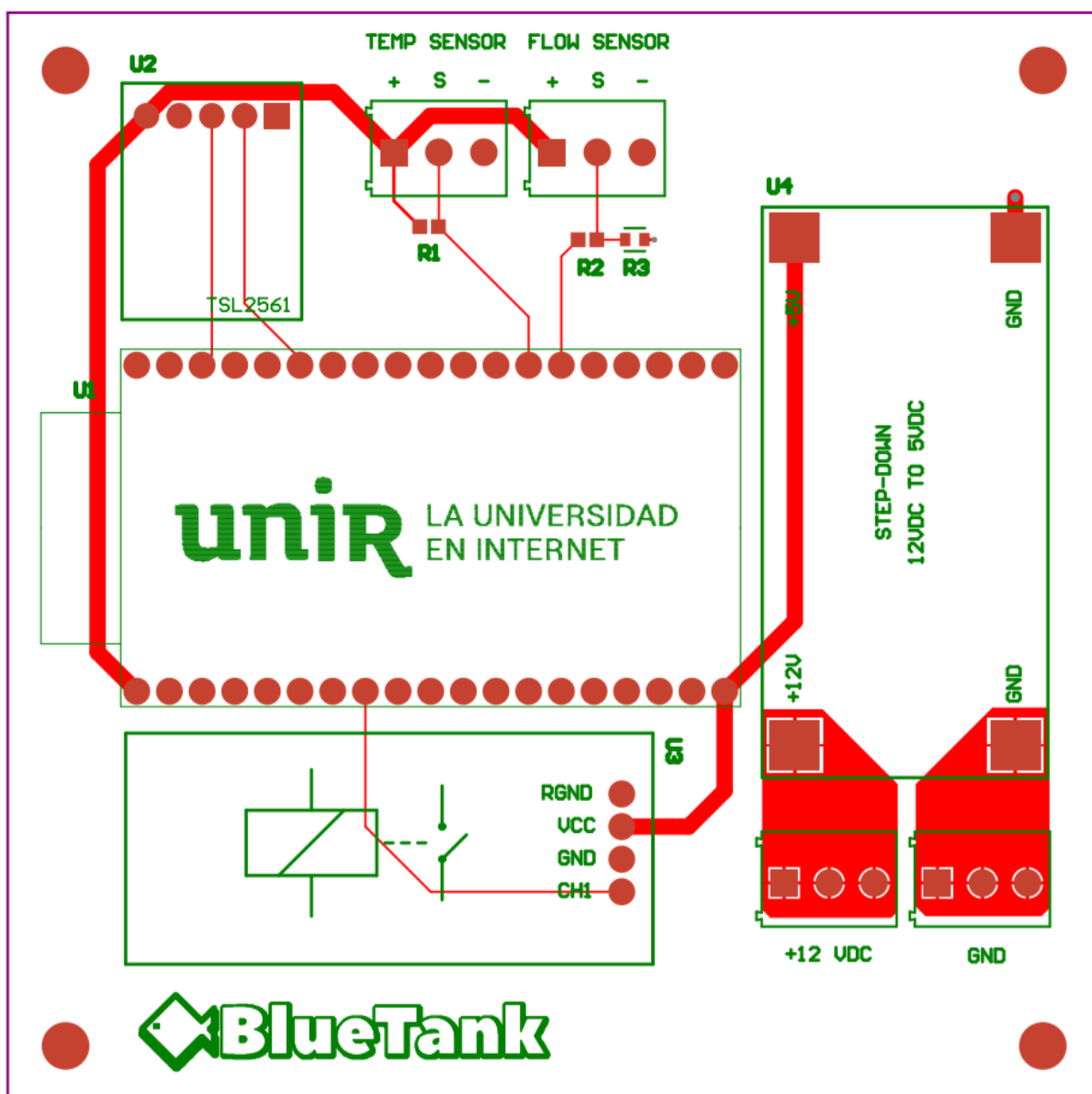


Ilustración 72. Diseño de la PCB. Vista delantera

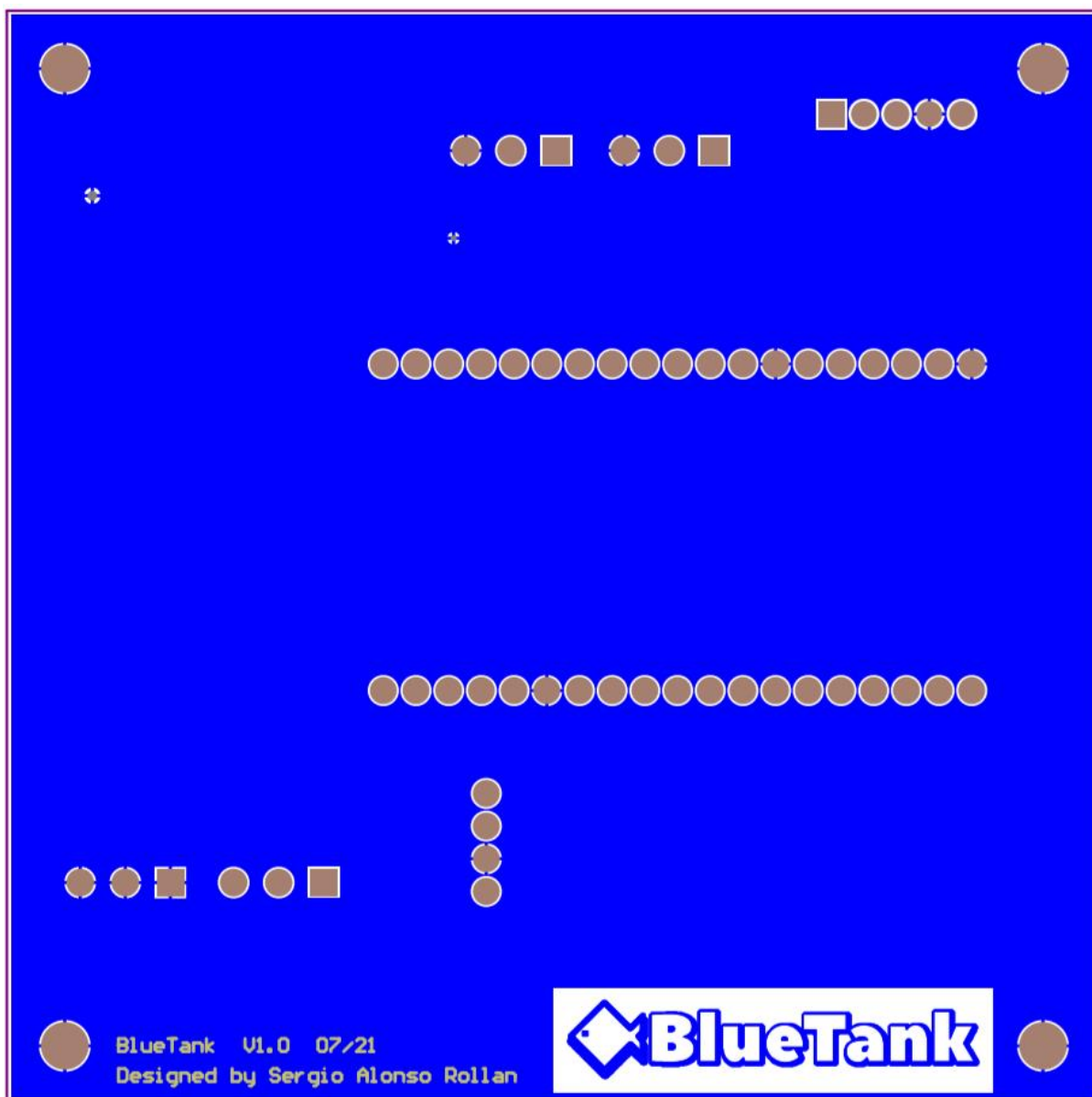


Ilustración 73. Diseño de la PCB. Vista trasera

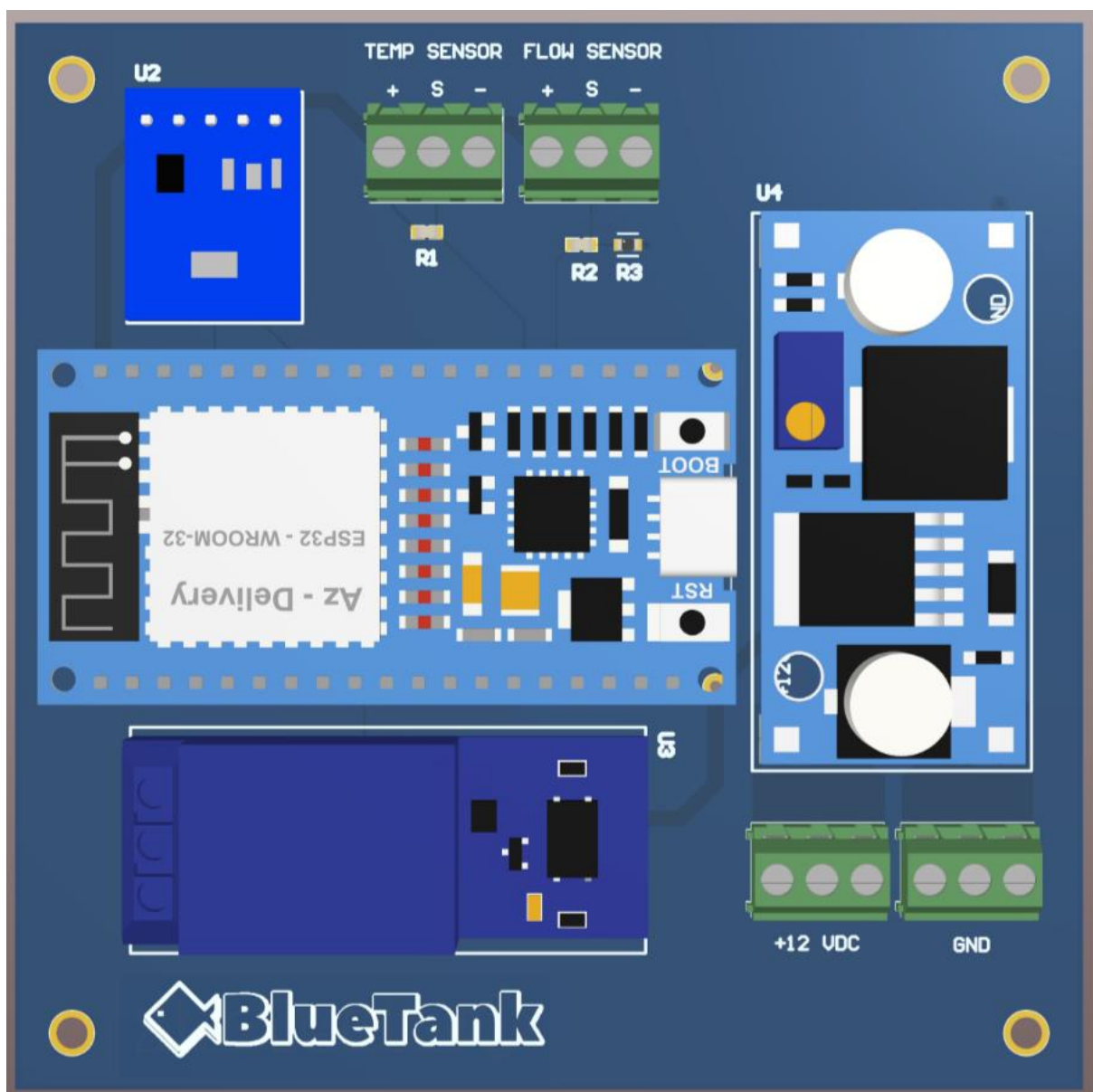


Ilustración 74. Render de la PCB. Vista delantera

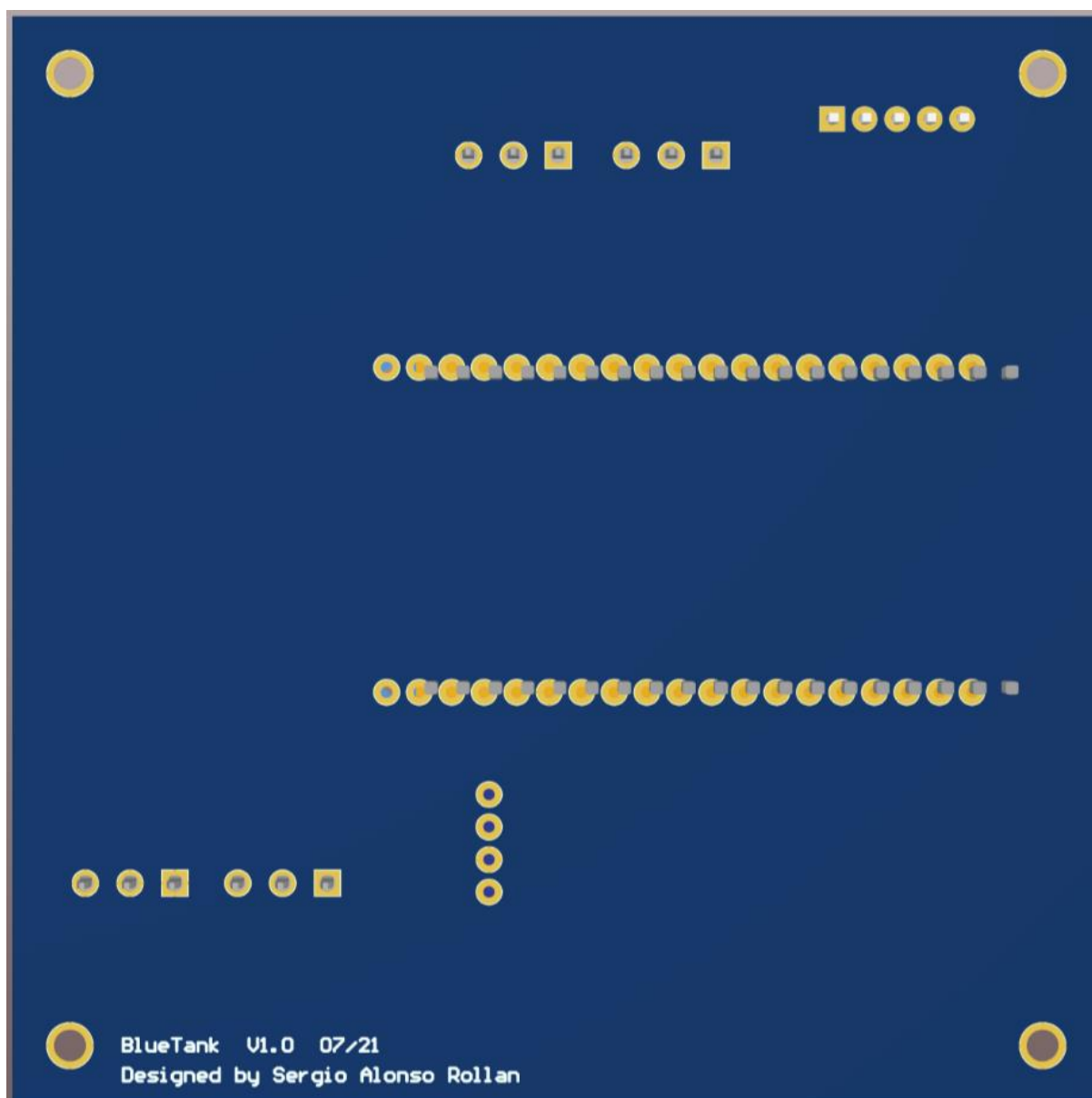


Ilustración 75. Render de la PCB. Vista trasera

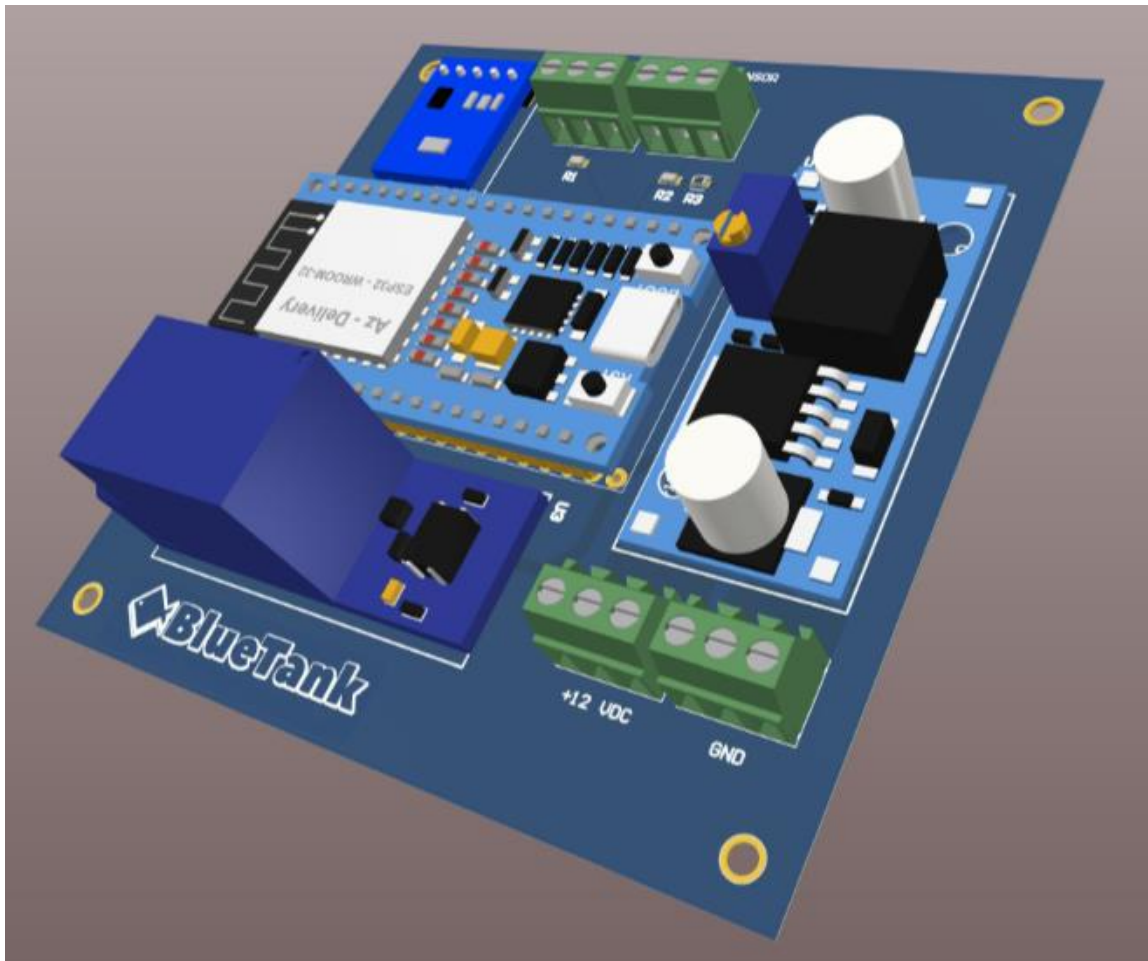


Ilustración 76. Render de la PCB. Vista 3D

Anexo F. Datasheets y certificados

Este anexo contiene los datasheets correspondientes a los componentes hardware utilizados y los certificados sobre los que se basa este trabajo.

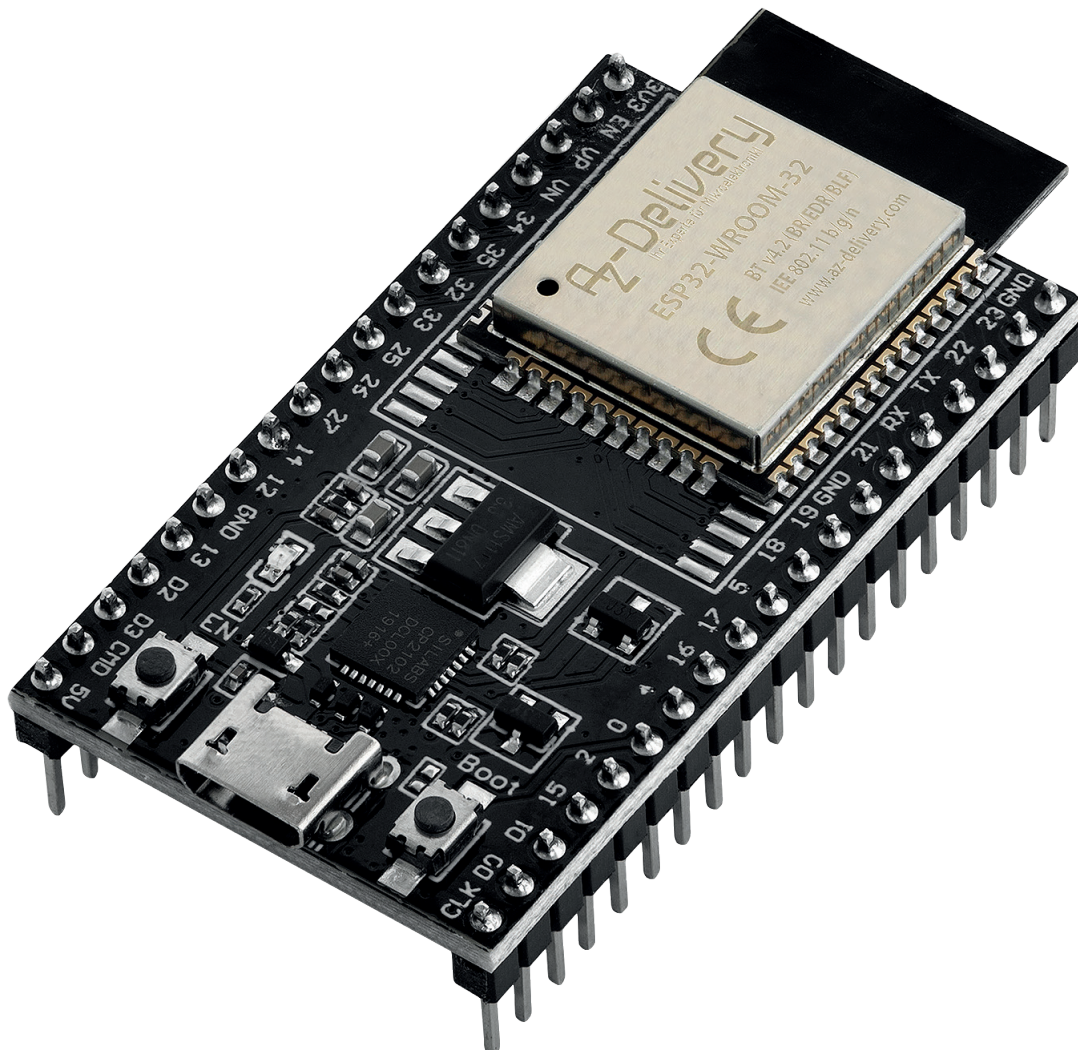
En cuanto a los datasheets encontramos.

- Microcontrolador ESP32.
- Sensor DS18B20 (Temperatura).
- Sensor TSL2561 (Iluminancia).
- Sensor YF-S201 (Caudal).
- Iluminancímetro o luxómetro profesional.
- Material de impresión PETG.

En cuanto a certificados encontramos.

- Certificación IP67 de la caja.
- Certificación IP68 de los conectores prensaestopas.
- Certificación del material PETG.

ESP-32 NodeMCU Development Board DevKit C V4 Datenblatt



Contents:

- 1. Features**
- 2. Pinout**
- 3. Specifications**

1. Features

NodeMCU is an open source IoT platform. ESP32 is a series of low cost, low power system-on-chip (SoC) microcontrollers with integrated Wi-Fi & dual-mode Bluetooth. The ESP32 series employs a Tensilica Xtensa LX6 microprocessor in both dual-core and single-core variations, with a clock rate of up to 240 MHz. ESP32 is highly integrated with built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules.

Features:

- Able to achieve ultra-low power consumption.
- Built-in ESP-WROOM-32 chip.
- Breadboard Friendly module.
- Light Weight and small size.
- On-chip Hall and temperature sensor
- Uses wireless protocol 802.11b/g/n.
- Built-in wireless connectivity capabilities.
- Built-in PCB antenna on the ESP32-WROOM-32
- Capable of PWM, I2C, SPI, UART, 1-wire, 1 analog pin.
- Uses CP2102 USB Serial Communication interface module.
- Programmable with ESP-IDF Toolchain, LuaNode SDK supports Eclipse project (C language).

3. Specifications

Wireless Standard	FCC/CE/IC/TELEC/KCC/SRRC/NCC
Wireless Protocol	802.11 b/g/n/d/e/l/k/r
Frequency Range	2.4 - 2.5 GHz
Bluetooth Protocol	Bluetooth v4.2 BR/EDR and BLE specification
Bluetooth Specifications	NZIF Receiver with -98dBm sensitivity Class-1, Class-2 and Class-3 transmitter AFH, CVSD and SBC
Memory	4 MB Flash, 520KB SRAM
Wireless Form	On-board PCB Antenna
IO Capability	UART, I2C, SPI, I2S, PWM, SDIO, GPIO, ADC, DAC
Electrical Characteristic	3.3 V Operated 15 mA output current per GPIO pin 80 mA average working current
Operating Temperature	-40 to +125 °C
Wireless Network Type	Station / SoftAP / SoftAP + Station / P2P
Security Type	WPA / WPA2 / WPA2-Enterprise / WPS
Encryption Type	AES / RSA / ECC / SHA
Firmware Upgrade	UART Download / OTA / Host
Network Protocol	IPv4, IPv6, SSL, TCP / UDP / FTP / HTTP / MQTT
User Configuration	AT + Order Set, Web Android / iOS, Cloud Server

DS18B20

Programmable Resolution 1-Wire Digital Thermometer

General Description

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

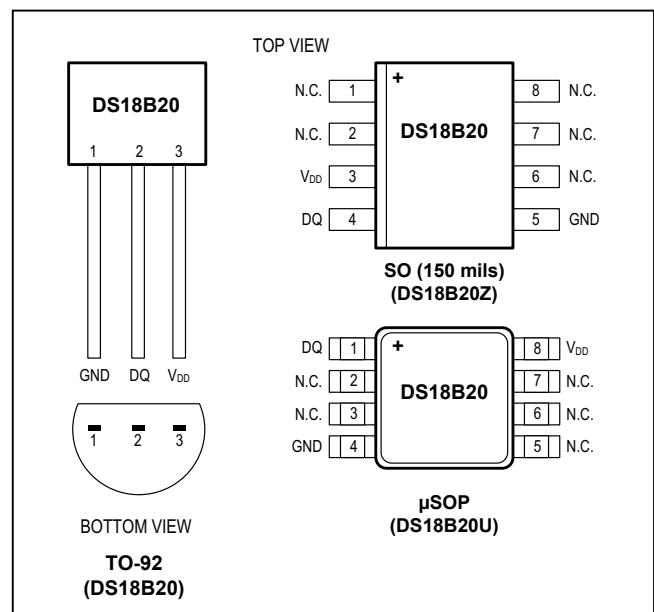
Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

Benefits and Features

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Reduce Component Count with Integrated Temperature Sensor and EEPROM
 - Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
 - ±0.5°C Accuracy from -10°C to +85°C
 - Programmable Resolution from 9 Bits to 12 Bits
 - No External Components Required
- Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
 - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

Pin Configurations



Ordering Information appears at end of data sheet.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground -0.5V to +6.0V
Operating Temperature Range..... -55°C to +125°C

Storage Temperature Range -55°C to +125°C
Solder Temperature Refer to the IPC/JEDEC
J-STD-020 Specification.

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC Electrical Characteristics

(-55°C to +125°C; V_{DD} = 3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{DD}	Local power (Note 1)	+3.0		+5.5	V
Pullup Supply Voltage	V_{PU}	Parasite power	+3.0		+5.5	V
		Local power	+3.0		V_{DD}	
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C
		-30°C to +100°C			±1	
		-55°C to +125°C			±2	
Input Logic-Low	V_{IL}	(Notes 1, 4, 5)	-0.3		+0.8	V
Input Logic-High	V_{IH}	Local power	+2.2	The lower of 5.5 or $V_{DD} + 0.3$		V
		Parasite power	+3.0			
Sink Current	I_L	$V_{I/O} = 0.4V$	4.0			mA
Standby Current	I_{DDs}	(Notes 7, 8)		750	1000	nA
Active Current	I_{DD}	$V_{DD} = 5V$ (Note 9)		1	1.5	mA
DQ Input Current	I_{DQ}	(Note 10)		5		μA
Drift		(Note 11)		±0.2		°C

Note 1: All voltages are referenced to ground.

Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU} . In order to meet the V_{IH} spec of the DS18B20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.

Note 3: See typical performance curve in [Figure 1](#). Thermometer Error limits are 3-sigma values.

Note 4: Logic-low voltages are specified at a sink current of 4mA.

Note 5: To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.

Note 6: Logic-high voltages are specified at a source current of 1mA.

Note 7: Standby current specified up to +70°C. Standby current typically is 3μA at +125°C.

Note 8: To minimize I_{DDs} , DQ should be within the following ranges: $GND \leq DQ \leq GND + 0.3V$ or $V_{DD} - 0.3V \leq DQ \leq V_{DD}$.

Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.

Note 10: DQ line is high ("high-Z" state).

Note 11: Drift data is based on a 1000-hour stress test at +125°C with $V_{DD} = 5.5V$.

AC Electrical Characteristics–NV Memory(-55°C to +125°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{WR}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC Electrical Characteristics(-55°C to +125°C; $V_{DD} = 3.0V$ to 5.5V)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Temperature Conversion Time	t_{CONV}	9-bit resolution	(Note 12)		93.75	ms
		10-bit resolution			187.5	
		11-bit resolution			375	
		12-bit resolution			750	
Time to Strong Pullup On	t_{SPON}	Start convert T command issued			10	μs
Time Slot	t_{SLOT}	(Note 12)	60		120	μs
Recovery Time	t_{REC}	(Note 12)	1			μs
Write 0 Low Time	t_{LOW0}	(Note 12)	60		120	μs
Write 1 Low Time	t_{LOW1}	(Note 12)	1		15	μs
Read Data Valid	t_{RDV}	(Note 12)			15	μs
Reset Time High	t_{RSTH}	(Note 12)	480			μs
Reset Time Low	t_{RSTL}	(Notes 12, 13)	480			μs
Presence-Detect High	t_{PDHIGH}	(Note 12)	15		60	μs
Presence-Detect Low	t_{PDLOW}	(Note 12)	60		240	μs
Capacitance	$C_{IN/OUT}$				25	pF

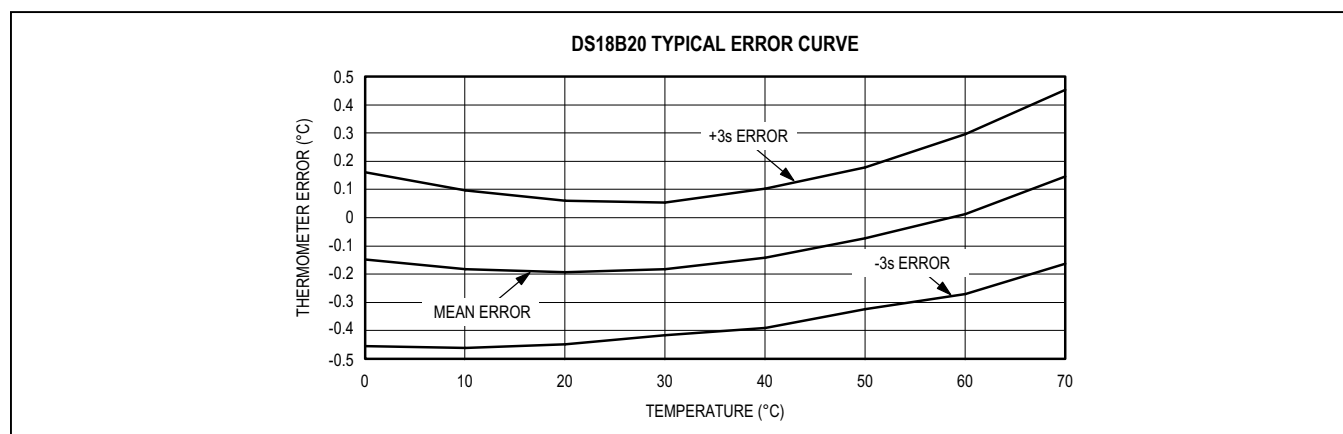
Note 12: See the timing diagrams in [Figure 2](#).**Note 13:** Under parasite power, if $t_{RSTL} > 960\mu s$, a power-on reset can occur.

Figure 1. Typical Performance Curve

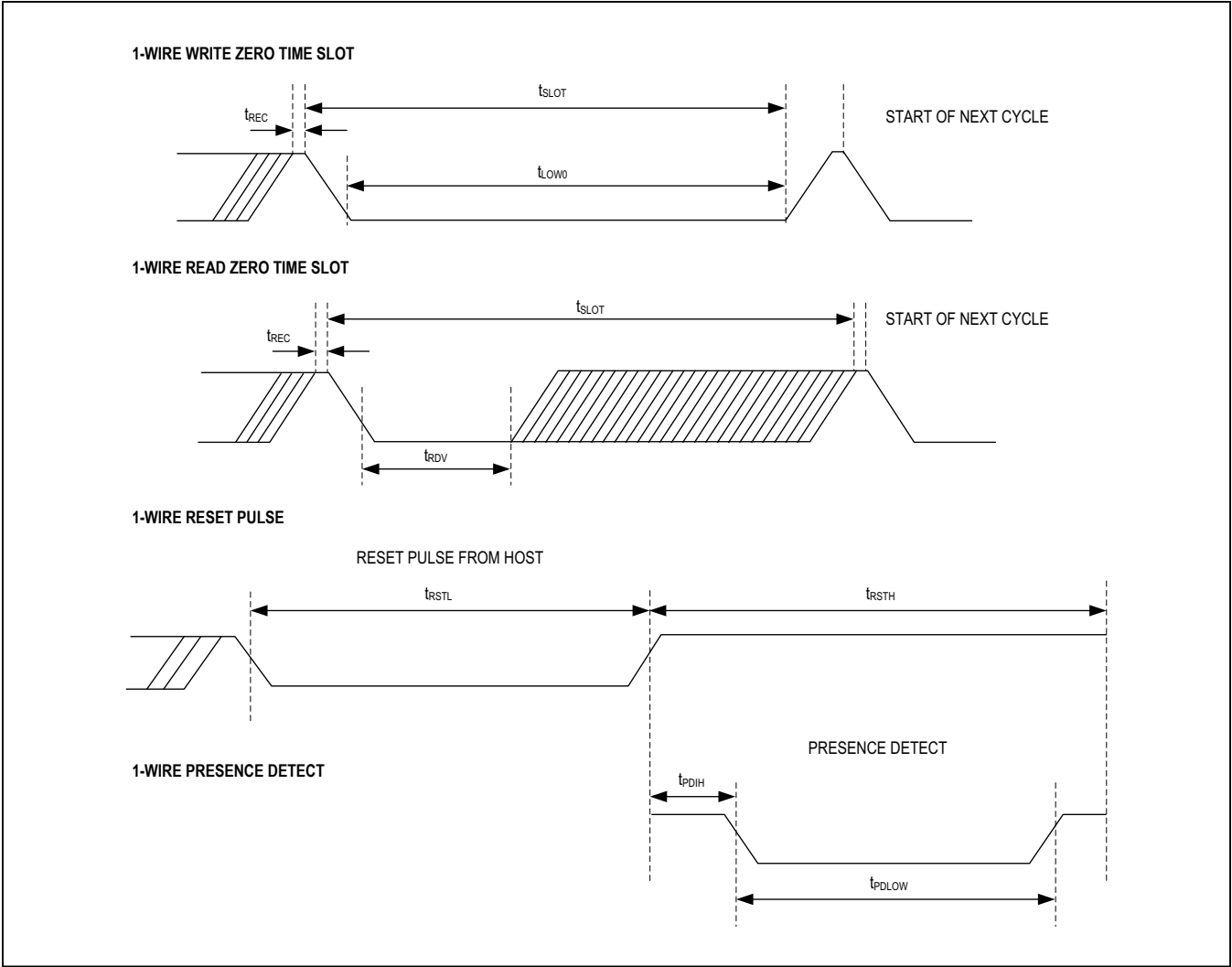


Figure 2. Timing Diagrams

Pin Description

PIN			NAME	FUNCTION
SO	μSOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V _{DD}	Optional V _{DD} . V _{DD} must be grounded for operation in parasite power mode.
4	1	2	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the <i>Powering the DS18B20</i> section.)
5	4	1	GND	Ground

Overview

Figure 3 shows a block diagram of the DS18B20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T_H , T_L , and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the [1-Wire Bus System](#) section.

Another feature of the DS18B20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor through the

DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18B20 may also be powered by an external supply on V_{DD} .

Operation—Measuring Temperature

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C , 0.25°C , 0.125°C , and 0.0625°C , respectively. The default resolution at power-up is 12-bit. The DS18B20 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its idle state. If the DS18B20 is powered by an external supply, the master can issue "read time slots" (see the [1-Wire Bus System](#) section) after the Convert T command and the DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the [Powering the DS18B20](#) section.

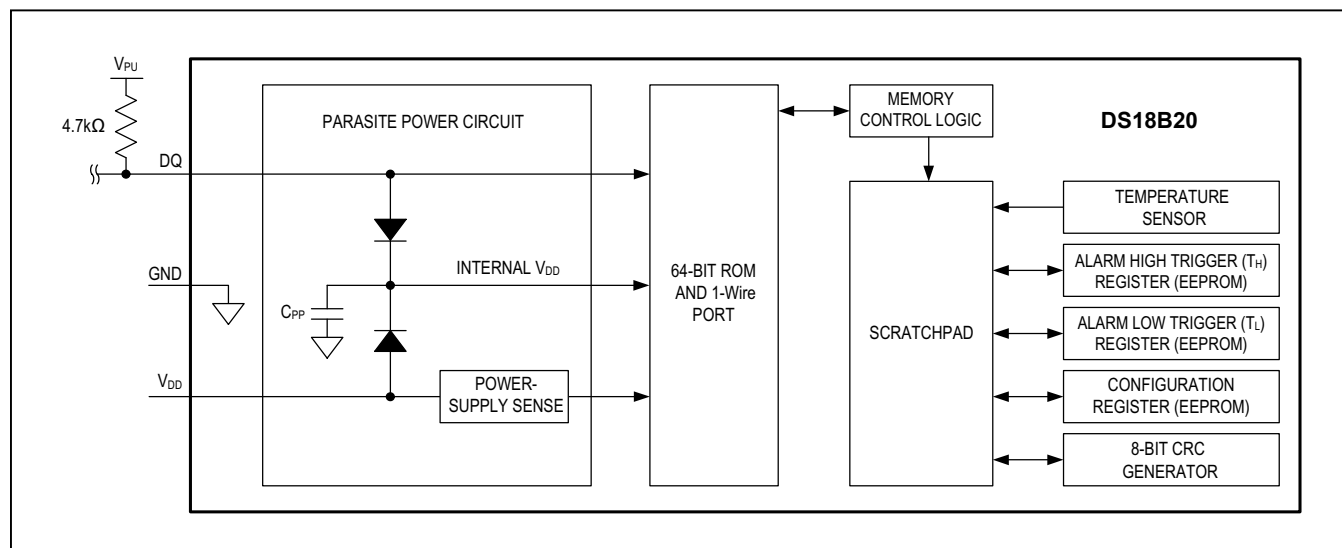


Figure 3. DS18B20 Block Diagram

The DS18B20 output temperature data is calibrated in degrees Celsius; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see [Figure 4](#)). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution bits 2, 1, and 0 are undefined. [Table 1](#) gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

Operation—Alarm Signaling

After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see [Figure 5](#)). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the [Memory](#) section.

Only bits 11 through 4 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the measured temperature is lower than

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2 ⁶	2 ⁵	2 ⁴

S = SIGN

Figure 4. Temperature Register Format

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

*The power-on reset value of the temperature register is +85°C.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

Figure 5. T_H and T_L Register Format

or equal to T_L or higher than or equal to T_H , an alarm condition exists and an alarm flag is set inside the DS18B20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS18B20s on the bus by issuing an Alarm Search [ECh] command. Any DS18B20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18B20s have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

Powering the DS18B20

The DS18B20 can be powered by an external supply on the V_{DD} pin, or it can operate in “parasite power” mode, which allows the DS18B20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or that are very space constrained. Figure 3 shows the DS18B20's parasite-power control circuitry, which “steals” power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18B20 while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low. When the DS18B20 is used in parasite power mode, the V_{DD} pin must be connected to ground.

In parasite power mode, the 1-Wire bus and CPP can provide sufficient current to the DS18B20 for most operations as long as the specified timing and voltage requirements are met (see the [DC Electrical Characteristics](#) and [AC Electrical Characteristics](#)). However, when the DS18B20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied

by C_{PP} . To assure that the DS18B20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 6. The 1-Wire bus must be switched to the strong pullup within 10 μ s (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{CONV}) or data transfer ($t_{WR} = 10$ ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18B20 can also be powered by the conventional method of connecting an external power supply to the V_{DD} pin, as shown in Figure 7. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

The use of parasite power is not recommended for temperatures above +100°C since the DS18B20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18B20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18B20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read time slot”. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

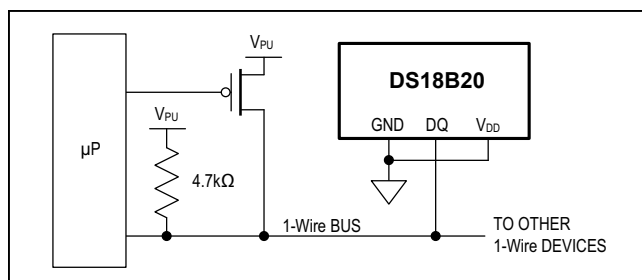


Figure 6. Supplying the Parasite-Powered DS18B20 During Temperature Conversions

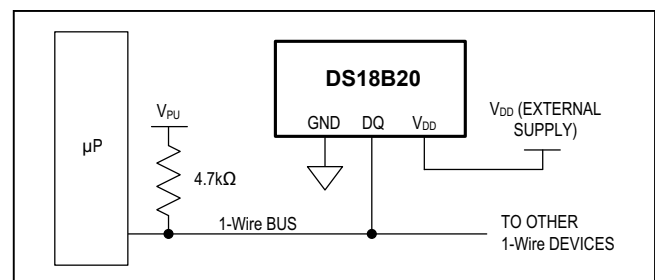


Figure 7. Powering the DS18B20 with an External Supply

64-BIT Lasered ROM code

Each DS18B20 contains a unique 64-bit code (see [Figure 8](#)) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the [CRC Generation](#) section. The 64-bit ROM code and associated ROM function control logic allow the DS18B20 to operate as a 1-Wire device using the protocol detailed in the [1-Wire Bus System](#) section.

Memory

The DS18B20's memory is organized as shown in [Figure 9](#). The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L) and configuration register. Note that if the DS18B20 alarm function is not used, the TH and TL registers can serve as general-purpose memory. All memory commands are described in detail in the [DS18B20 Function Commands](#) section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to TH and TL registers. Byte 4 contains the configuration regis-

ter data, which is explained in detail in the [Configuration Register](#) section. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in the [CRC Generation](#) section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H, T_L and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E² [B8h] command. The master can issue read time slots following the Recall E² command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

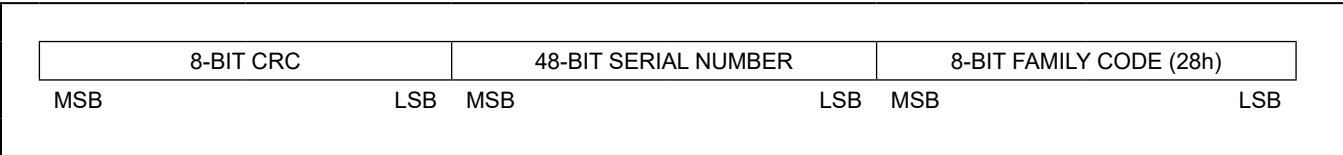


Figure 8. 64-Bit Lasered ROM Code

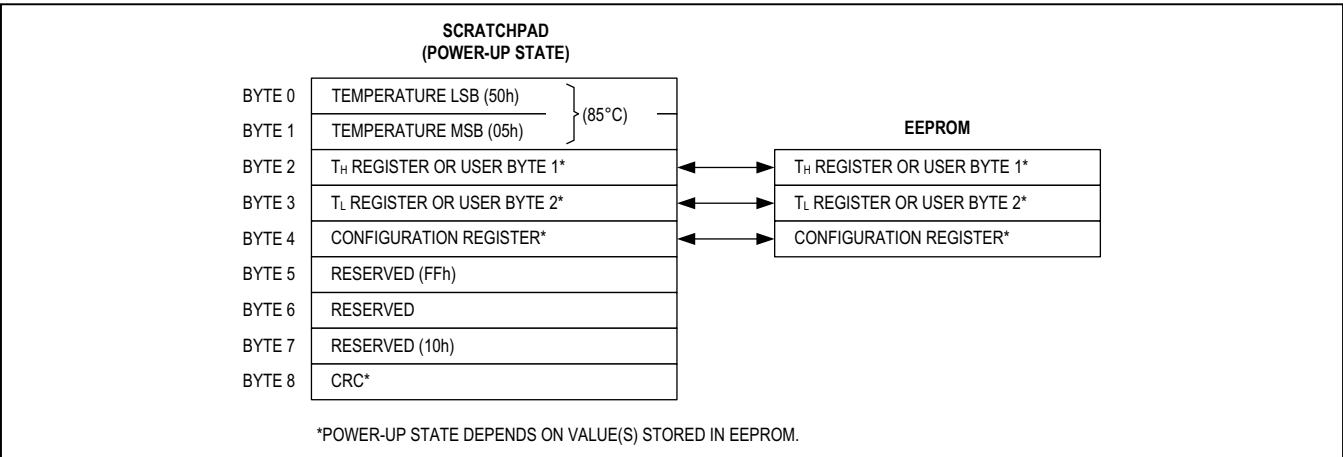


Figure 9. DS18B20 Memory Map

Configuration Register

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in [Figure 10](#). The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in [Table 2](#). The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten.

CRC Generation

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18B20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been

received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$CRC = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in [Figure 11](#). This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the recalculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18B20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Maxim 1-Wire cyclic redundancy check is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products*.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

Figure 10. Configuration Register

Table 2. Thermometer Resolution Configuration

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	(t _{CONV} /8)
0	1	10	187.5ms	(t _{CONV} /4)
1	0	11	375ms	(t _{CONV} /2)
1	1	12	750ms	(t _{CONV})

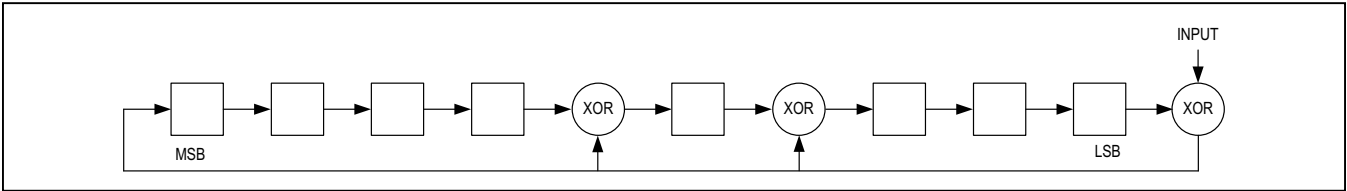


Figure 11. CRC Generator

1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 is always a slave. When there is only one slave on the bus, the system is referred to as a “single-drop” system; the system is “multidrop” if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

Hardware Configuration

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open-drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in [Figure 12](#).

The 1-Wire bus requires an external pullup resistor of approximately 5k Ω ; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480 μ s, all components on the bus will be reset.

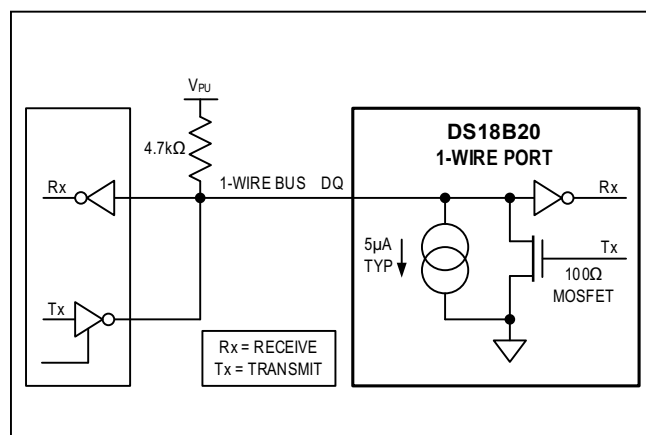


Figure 12. Hardware Configuration

Transaction Sequence

The transaction sequence for accessing the DS18B20 is as follows:

- Step 1. Initialization
- Step 2. ROM Command (followed by any required data exchange)
- Step 3. DS18B20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18B20 is accessed, as the DS18B20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18B20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the [1-Wire Signaling](#) section.

ROM Commands

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18B20 function command. A flowchart for operation of the ROM commands is shown in [Figure 13](#).

Search Rom [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices.

If there is only one slave on the bus, the simpler Read ROM [33h] command can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to *Application Note 937: Book of iButton® Standards*. After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

Read Rom [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

Match Rom [55H]

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

Skip Rom [CCh]

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18B20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

Alarm Search [ECh]

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18B20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus

master must return to Step 1 (Initialization) in the transaction sequence. See the [Operation—Alarm Signaling](#) section for an explanation of alarm flag operation.

DS18B20 Function Commands

After the bus master has used a ROM command to address the DS18B20 with which it wishes to communicate, the master can issue one of the DS18B20 function commands. These commands allow the master to write to and read from the DS18B20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18B20 function commands, which are described below, are summarized in [Table 3](#) and illustrated by the flowchart in [Figure 14](#).

Convert T [44h]

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18B20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion (t_{CONV}) as described in the [Powering the DS18B20](#) section. If the DS18B20 is powered by an external supply, the master can issue read time slots after the Convert T command and the DS18B20 will respond by transmitting a 0 while the temperature conversion is in progress and a 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

Write Scratchpad [4Eh]

This command allows the master to write 3 bytes of data to the DS18B20's scratchpad. The first data byte is written into the T_H register (byte 2 of the scratchpad), the second byte is written into the T_L register (byte 3), and the third byte is written into the configuration register (byte 4). Data must be transmitted least significant bit first. All three bytes MUST be written before the master issues a reset, or the data may be corrupted.

Read Scratchpad [BEh]

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

iButton is a registered trademark of Maxim Integrated Products, Inc.

Copy Scratchpad [48h]

This command copies the contents of the scratchpad T_H , T_L and configuration registers (bytes 2, 3 and 4) to EEPROM. If the device is being used in parasite power mode, within 10 μ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the [Powering the DS18B20](#) section.

Recall E² [B8h]

This command recalls the alarm trigger values (T_H and T_L) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots

following the Recall E² command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

Read Power Supply [B4h]

The master device issues this command followed by a read time slot to determine if any DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the [Powering the DS18B20](#) section for usage information for this command.

Table 3. DS18B20 Function Command Set

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 (T_H , T_L , and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies T_H , T_L , and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E ²	Recalls T_H , T_L , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

Note 1: For parasite-powered DS18B20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.

Note 2: The master can interrupt the transmission of data at any time by issuing a reset.

Note 3: All three bytes must be written before a reset is issued.

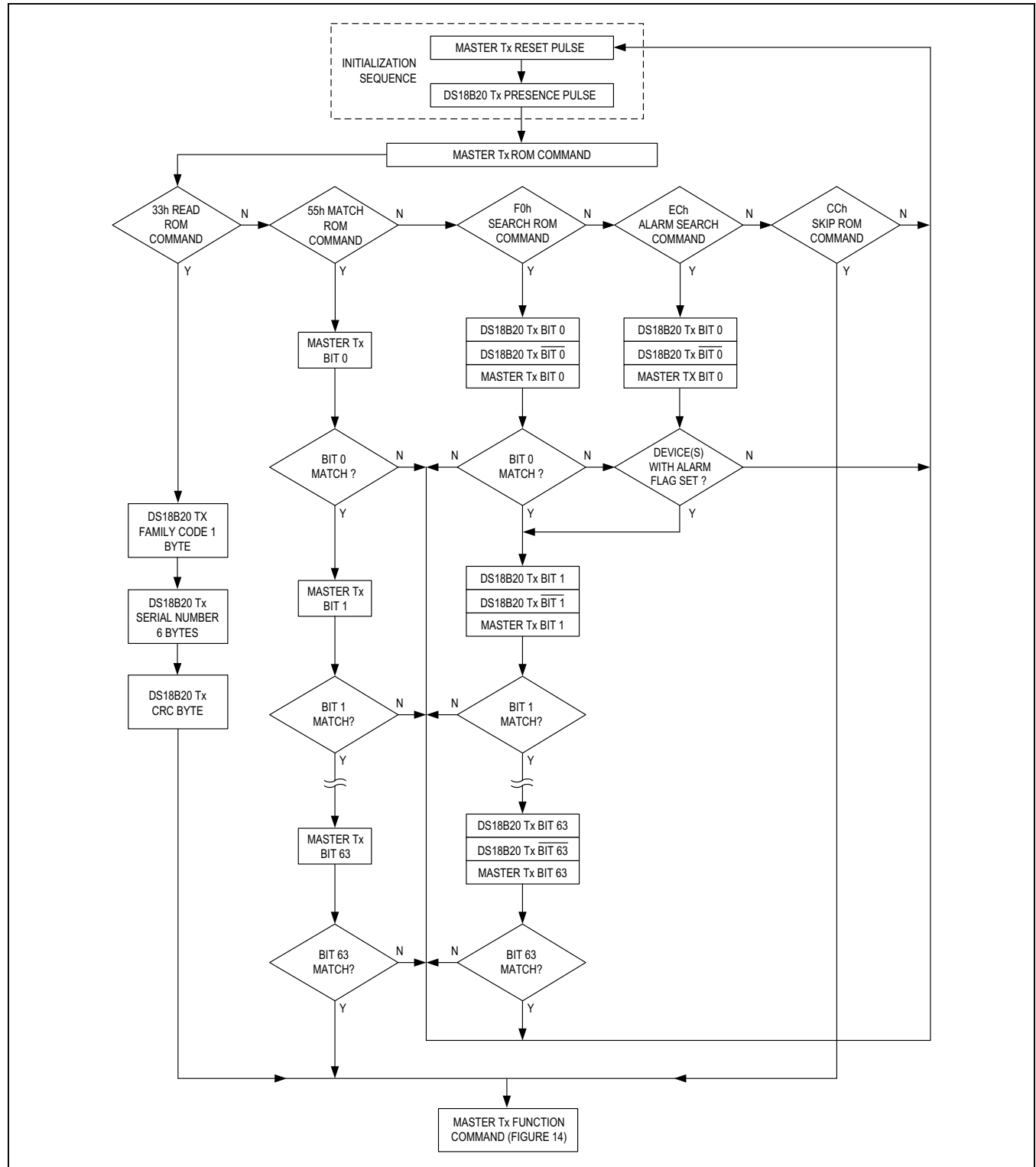


Figure 13. ROM Commands Flowchart

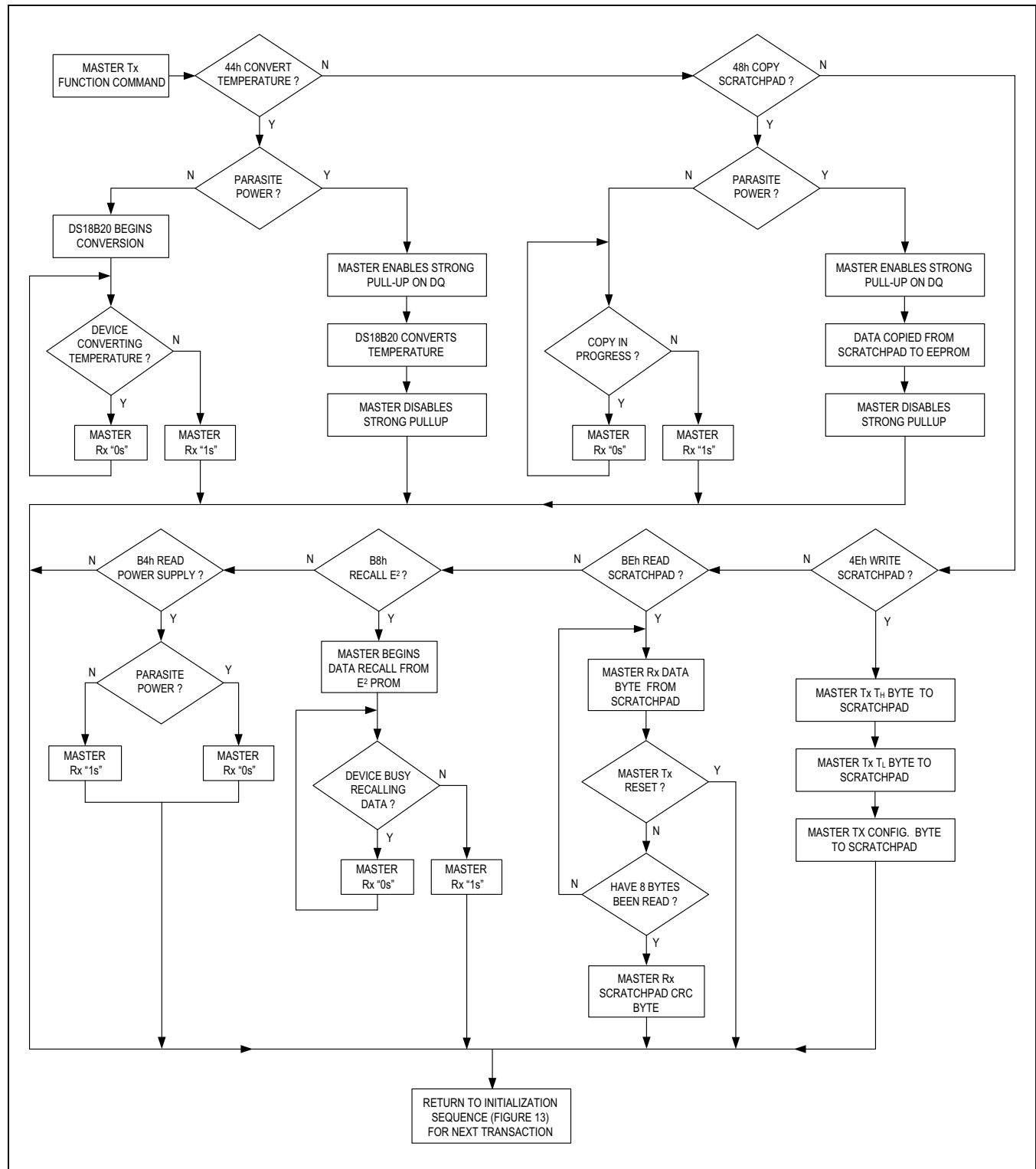


Figure 14. DS18B20 Function Commands Flowchart

1-Wire Signaling

The DS18B20 uses a strict 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. The bus master initiates all these signals, with the exception of the presence pulse.

Initialization Procedure—Reset And Presence Pulses

All communication with the DS18B20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18B20. This is illustrated in [Figure 15](#). When the DS18B20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (T_X) the reset pulse by pulling the 1-Wire bus low for a minimum of 480 μ s. The bus master then releases the bus and goes into receive mode (R_X). When the bus is released, the 5k Ω pullup resistor pulls the 1-Wire bus high. When the DS18B20 detects this rising edge, it waits 15 μ s to 60 μ s and then transmits a presence pulse by pulling the 1-Wire bus low for 60 μ s to 240 μ s.

Read/Write Time Slots

The bus master writes data to the DS18B20 during write time slots and reads data from the DS18B20 during read time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

Write Time Slots

There are two types of write time slots: “Write 1” time slots and “Write 0” time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18B20 and a Write 0 time slot to write a logic 0 to the DS18B20. All write time slots must be a minimum of 60 μ s in duration with a minimum of a 1 μ s recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see [Figure 14](#)).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within 15 μ s. When the bus is released, the 5k Ω pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60 μ s).

The DS18B20 samples the 1-Wire bus during a window that lasts from 15 μ s to 60 μ s after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18B20. If the line is low, a 0 is written to the DS18B20.

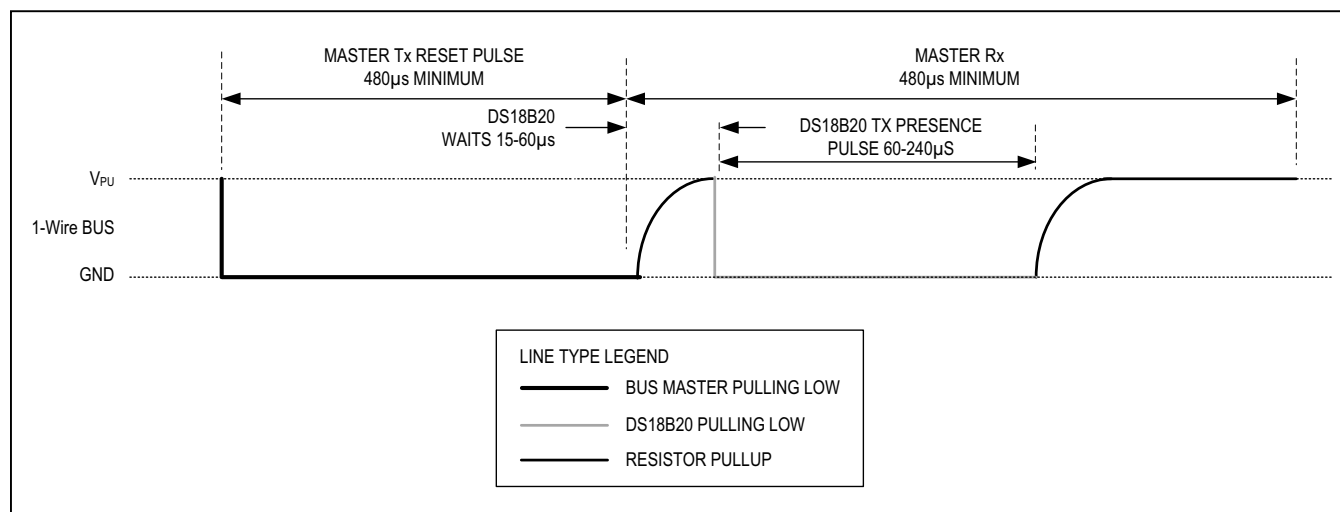


Figure 15. Initialization Timing

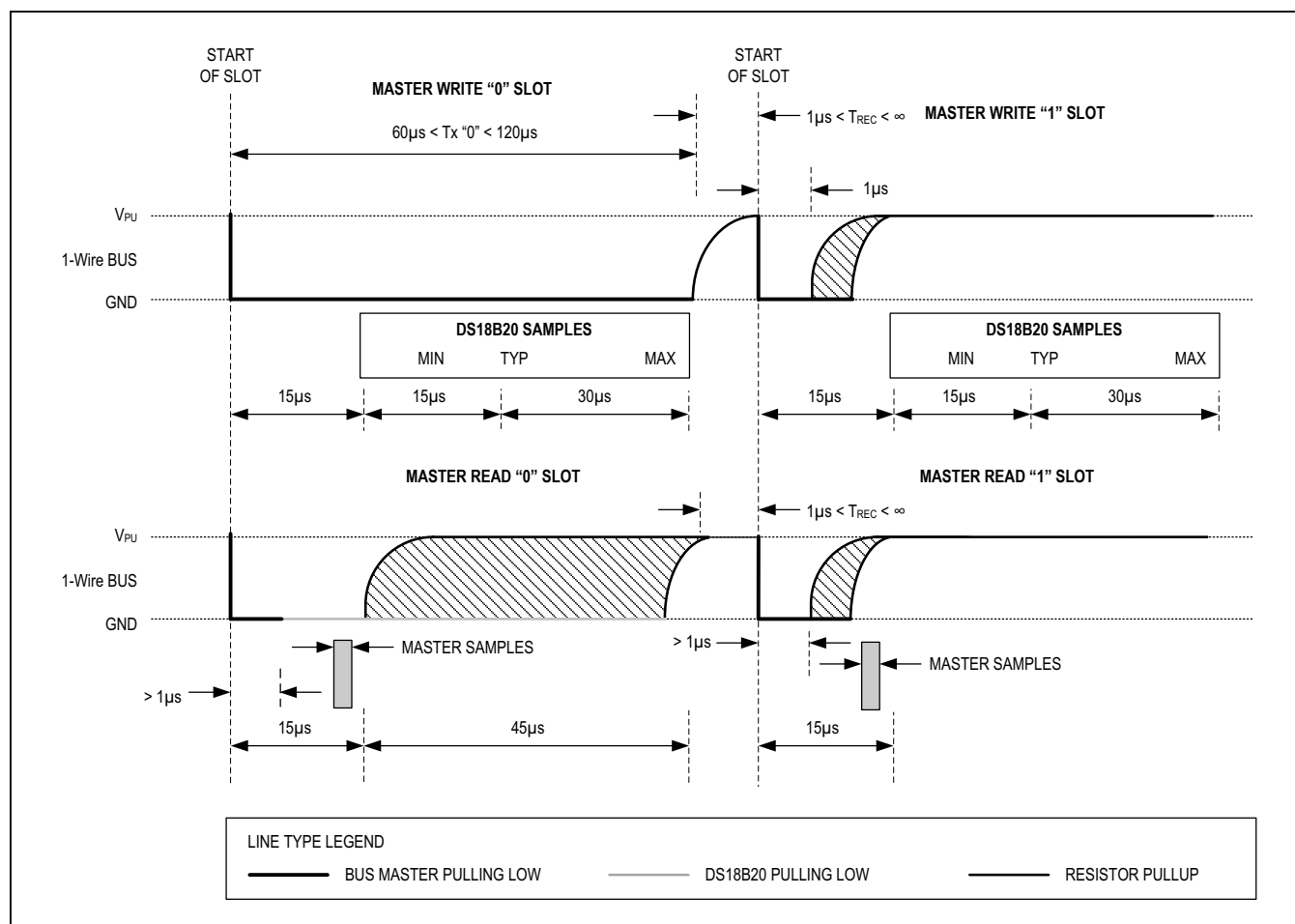


Figure 16. Read/Write Time Slot Timing Diagram

Read Time Slots

The DS18B20 can only transmit data to the master when the master issues read time slots. Therefore, the master must generate read time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18B20 can provide the requested data. In addition, the master can generate read time slots after issuing Convert T [44h] or Recall E2 [B8h] commands to find out the status of the operation as explained in the [DS18B20 Function Commands](#) section.

All read time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see [Figure 16](#)). After the master initiates the

read time slot, the DS18B20 will begin transmitting a 1 or 0 on bus. The DS18B20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18B20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output data from the DS18B20 is valid for 15µs after the falling edge that initiated the read time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

[Figure 17](#) illustrates that the sum of T_{INIT} , T_{RC} , and T_{SAMPLE} must be less than 15µs for a read time slot. [Figure 18](#) shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as short as possible and by locating the master sample time during read time slots towards the end of the 15µs period.

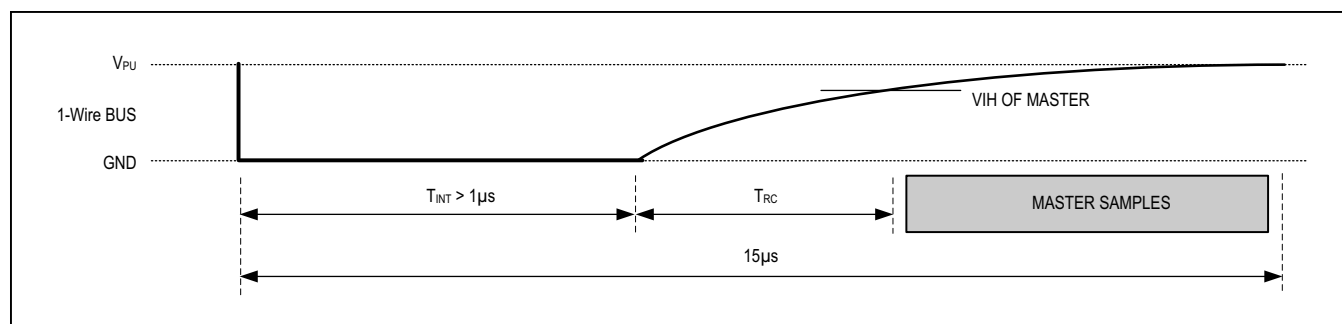


Figure 17. Detailed Master Read 1 Timing

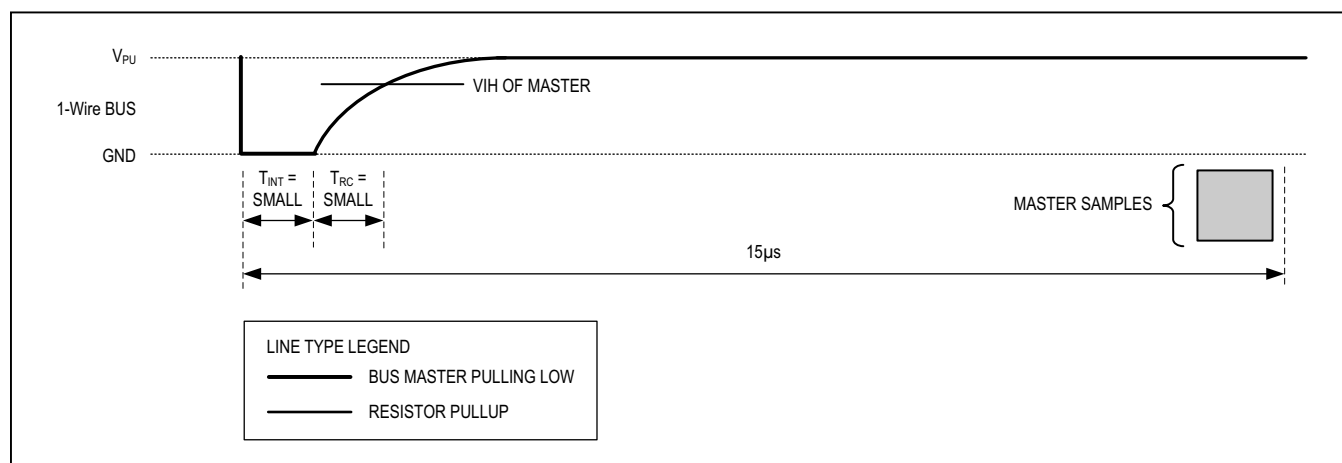


Figure 18. Recommended Master Read 1 Timing

Related Application Notes

The following application notes can be applied to the DS18B20 and are available at www.maximintegrated.com.

Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products

Application Note 122: Using Dallas' 1-Wire ICs in 1-Cell Li-Ion Battery Packs with Low-Side N-Channel Safety FETs Master

Application Note 126: 1-Wire Communication Through Software

Application Note 162: Interfacing the DS18x20/DS1822 1-Wire Temperature Sensor in a Microcontroller Environment

Application Note 208: Curve Fitting the Error of a Bandgap-Based Digital Temperature Sensor

Application Note 2420: 1-Wire Communication with a Microchip PICmicro Microcontroller

Application Note 3754: Single-Wire Serial Bus Carries Isolated Power and Data

Sample 1-Wire subroutines that can be used in conjunction with *Application Note 74: Reading and Writing iButtons via Serial Interfaces* can be downloaded from the Maxim website.

DS18B20 Operation Example 1

In this example there are multiple DS18B20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18B20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion (t_{CONV}).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18B20 ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

DS18B20 Operation Example 2

In this example there is only one DS18B20 on the bus and it is using parasite power. The master writes to the TH, TL, and configuration registers in the DS18B20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	3 data bytes	Master sends three data bytes to scratchpad (T_H , T_L , and config).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18B20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 FSOP	18B20
DS18B20U+	-55°C to +125°C	8 FSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 FSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

+Denotes a lead-free package. A "+" will appear on the top mark of lead-free packages.

T&R = Tape and reel.

*TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.

Revision History

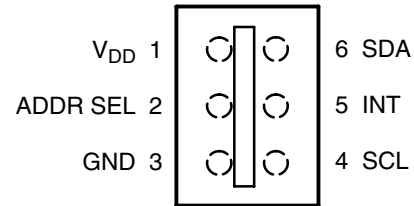
REVISION DATE	DESCRIPTION	PAGES CHANGED
3/1/07	In the Absolute Maximum Ratings section, removed the reflow oven temperature value of +220°C. Reference to JEDEC specification for reflow remains.	19
10/12/07	In the <i>Operation—Alarm Signaling</i> section, added “or equal to” in the description for a TH alarm condition	5
	In the <i>Memory</i> section, removed incorrect text describing memory.	7
	In the <i>Configuration Register</i> section, removed incorrect text describing configuration register.	8
4/22/08	In the <i>Ordering Information</i> table, added TO-92 straight-lead packages and included a note that the TO-92 package in tape and reel can be ordered with either formed or straight leads.	2
1/15	Updated <i>Benefits and Features</i> section	1
09/18	Updated <i>DC Electrical Characteristics</i> table	2
7/19	Updated Figure 12	10

For pricing, delivery, and ordering information, please visit Maxim Integrated's online storefront at <https://www.maximintegrated.com/en/storefront/storefront.html>.

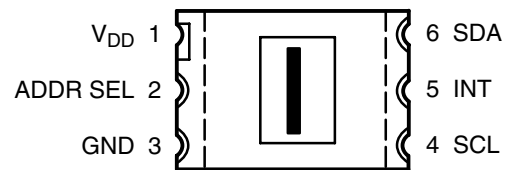
Maxim Integrated cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim Integrated product. No circuit patent licenses are implied. Maxim Integrated reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.

- Approximates Human Eye Response
- Programmable Interrupt Function with User-Defined Upper and Lower Threshold Settings
- 16-Bit Digital Output with SMBus (TSL2560) at 100 kHz or I²C (TSL2561) Fast-Mode at 400 kHz
- Programmable Analog Gain and Integration Time Supporting 1,000,000-to-1 Dynamic Range
- Automatically Rejects 50/60-Hz Lighting Ripple
- Low Active Power (0.75 mW Typical) with Power Down Mode
- RoHS Compliant

**PACKAGE CS
6-LEAD CHIPSCALE
(TOP VIEW)**



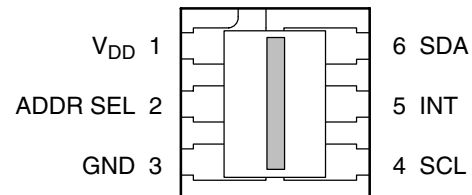
**PACKAGE T
6-LEAD TMB
(TOP VIEW)**



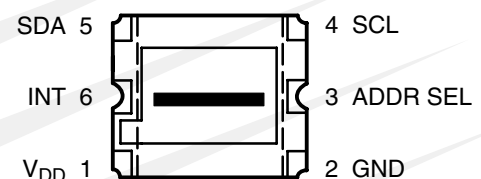
Description

The TSL2560 and TSL2561 are light-to-digital converters that transform light intensity to a digital signal output capable of direct I²C (TSL2561) or SMBus (TSL2560) interface. Each device combines one broadband photodiode (visible plus infrared) and one infrared-responding photodiode on a single CMOS integrated circuit capable of providing a near-photopic response over an effective 20-bit dynamic range (16-bit resolution). Two integrating ADCs convert the photodiode currents to a digital output that represents the irradiance measured on each channel. This digital output can be input to a microprocessor where illuminance (ambient light level) in lux is derived using an empirical formula to approximate the human eye response. The TSL2560 device permits an SMB-Alert style interrupt, and the TSL2561 device supports a traditional level style interrupt that remains asserted until the firmware clears it.

**PACKAGE FN
DUAL FLAT NO-LEAD
(TOP VIEW)**



**PACKAGE CL
6-LEAD ChipLED
(TOP VIEW)**



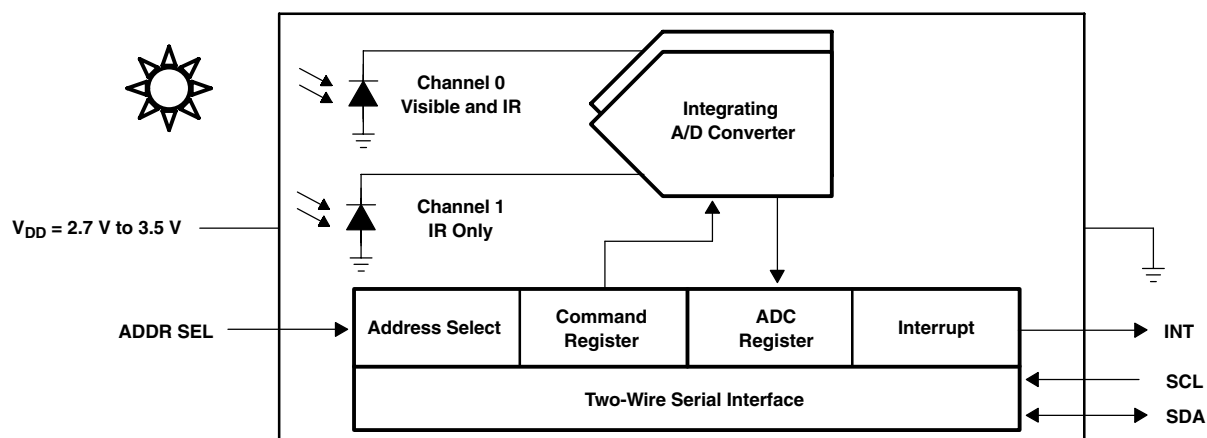
Package Drawings are Not to Scale

While useful for general purpose light sensing applications, the TSL2560/61 devices are designed particularly for display panels (LCD, OLED, etc.) with the purpose of extending battery life and providing optimum viewing in diverse lighting conditions. Display panel backlighting, which can account for up to 30 to 40 percent of total platform power, can be automatically managed. Both devices are also ideal for controlling keyboard illumination based upon ambient lighting conditions. Illuminance information can further be used to manage exposure control in digital cameras. The TSL2560/61 devices are ideal in notebook/tablet PCs, LCD monitors, flat-panel televisions, cell phones, and digital cameras. In addition, other applications include street light control, security lighting, sunlight harvesting, machine vision, and automotive instrumentation clusters.

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

Functional Block Diagram



Detailed Description

The TSL2560 and TSL2561 are second-generation ambient light sensor devices. Each contains two integrating analog-to-digital converters (ADC) that integrate currents from two photodiodes. Integration of both channels occurs simultaneously. Upon completion of the conversion cycle, the conversion result is transferred to the Channel 0 and Channel 1 data registers, respectively. The transfers are double-buffered to ensure that the integrity of the data is maintained. After the transfer, the device automatically begins the next integration cycle.

Communication to the device is accomplished through a standard, two-wire SMBus or I²C serial bus. Consequently, the TSL256x device can be easily connected to a microcontroller or embedded controller. No external circuitry is required for signal conditioning, thereby saving PCB real estate as well. Since the output of the TSL256x device is digital, the output is effectively immune to noise when compared to an analog signal.

The TSL256x devices also support an interrupt feature that simplifies and improves system efficiency by eliminating the need to poll a sensor for a light intensity value. The primary purpose of the interrupt function is to detect a meaningful change in light intensity. The concept of a *meaningful change* can be defined by the user both in terms of light intensity and time, or persistence, of that change in intensity. The TSL256x devices have the ability to define a threshold above and below the current light level. An interrupt is generated when the value of a conversion exceeds either of these limits.

Available Options

DEVICE	INTERFACE	PACKAGE – LEADS	PACKAGE DESIGNATOR	ORDERING NUMBER
TSL2560	SMBus	Chipscale	CS	TSL2560CS
TSL2560	SMBus	TMB-6	T	TSL2560T
TSL2560	SMBus	Dual Flat No-Lead – 6	FN	TSL2560FN
TSL2560	SMBus	ChipLED-6	CL	TSL2560CL
TSL2561	I ² C	Chipscale	CS	TSL2561CS
TSL2561	I ² C	TMB-6	T	TSL2561T
TSL2561	I ² C	Dual Flat No-Lead – 6	FN	TSL2561FN
TSL2561	I ² C	ChipLED-6	CL	TSL2561CL

Terminal Functions

TERMINAL NAME	CS, T, FN PKG NO.	CL PKG NO.	TYPE	DESCRIPTION
ADDR SEL	2	3	I	SMBus device select — three-state
GND	3	2		Power supply ground. All voltages are referenced to GND.
INT	5	6	O	Level or SMB Alert interrupt — open drain.
SCL	4	4	I	SMBus serial clock input terminal — clock signal for SMBus serial data.
SDA	6	5	I/O	SMBus serial data I/O terminal — serial data I/O for SMBus.
V _{DD}	1	1		Supply voltage.

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)[†]

Supply voltage, V _{DD} (see Note 1)	3.8 V
Digital output voltage range, V _O	–0.5 V to 3.8 V
Digital output current, I _O	–1 mA to 20 mA
Storage temperature range, T _{stg}	–40°C to 85°C
ESD tolerance, human body model	2000 V

[†] Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltages are with respect to GND.

Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V _{DD}	2.7	3	3.6	V
Operating free-air temperature, T _A	–30		70	°C
SCL, SDA input low voltage, V _{IL}	–0.5		0.8	V
SCL, SDA input high voltage, V _{IH}	2.1		3.6	V

Electrical Characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
I _{DD} Supply current	Active		0.24	0.6	mA
	Power down		3.2	15	μA
V _{OL} INT, SDA output low voltage	3 mA sink current	0		0.4	V
	6 mA sink current	0		0.6	V
I _{LEAK} Leakage current		–5		5	μA

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

Operating Characteristics, High Gain (16×), $V_{DD} = 3\text{ V}$, $T_A = 25^\circ\text{C}$, (unless otherwise noted) (see Notes 2, 3, 4, 5)

PARAMETER	TEST CONDITIONS	CHANNEL	TSL2560T, FN, & CL TSL2561T, FN & CL			TSL2560CS, TSL2561CS			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
f_{osc} Oscillator frequency			690	735	780	690	735	780	kHz
Dark ADC count value	$E_e = 0$, $T_{int} = 402\text{ ms}$	Ch0	0		4	0		4	counts
		Ch1	0		4	0		4	
Full scale ADC count value (Note 6)	$T_{int} > 178\text{ ms}$	Ch0			65535			65535	counts
		Ch1			65535			65535	
	$T_{int} = 101\text{ ms}$	Ch0			37177			37177	
		Ch1			37177			37177	
	$T_{int} = 13.7\text{ ms}$	Ch0			5047			5047	
		Ch1			5047			5047	
ADC count value	$\lambda_p = 640\text{ nm}$, $T_{int} = 101\text{ ms}$ $E_e = 36.3\text{ }\mu\text{W}/\text{cm}^2$	Ch0	750	1000	1250				counts
		Ch1		200					
	$\lambda_p = 940\text{ nm}$, $T_{int} = 101\text{ ms}$ $E_e = 119\text{ }\mu\text{W}/\text{cm}^2$	Ch0	700	1000	1300				counts
		Ch1		820					
	$\lambda_p = 640\text{ nm}$, $T_{int} = 101\text{ ms}$ $E_e = 41\text{ }\mu\text{W}/\text{cm}^2$	Ch0				750	1000	1250	counts
		Ch1					190		
	$\lambda_p = 940\text{ nm}$, $T_{int} = 101\text{ ms}$ $E_e = 135\text{ }\mu\text{W}/\text{cm}^2$	Ch0				700	1000	1300	counts
		Ch1					850		
ADC count value ratio: Ch1/Ch0	$\lambda_p = 640\text{ nm}$, $T_{int} = 101\text{ ms}$		0.15	0.20	0.25	0.14	0.19	0.24	
	$\lambda_p = 940\text{ nm}$, $T_{int} = 101\text{ ms}$		0.69	0.82	0.95	0.70	0.85	1	
R_e Irradiance responsivity	$\lambda_p = 640\text{ nm}$, $T_{int} = 101\text{ ms}$	Ch0		27.5			24.4		counts/ ($\mu\text{W}/\text{cm}^2$)
		Ch1		5.5			4.6		
	$\lambda_p = 940\text{ nm}$, $T_{int} = 101\text{ ms}$	Ch0		8.4			7.4		
		Ch1		6.9			6.3		
R_v Illuminance responsivity	Fluorescent light source: $T_{int} = 402\text{ ms}$	Ch0		36			35		counts/ lux
		Ch1		4			3.8		
	Incandescent light source: $T_{int} = 402\text{ ms}$	Ch0		144			129		
		Ch1		72			67		
ADC count value ratio: Ch1/Ch0	Fluorescent light source: $T_{int} = 402\text{ ms}$			0.11			0.11		
	Incandescent light source: $T_{int} = 402\text{ ms}$			0.5			0.52		
R_v Illuminance responsivity, low gain mode (Note 7)	Fluorescent light source: $T_{int} = 402\text{ ms}$	Ch0		2.3			2.2		counts/ lux
		Ch1		0.25			0.24		
	Incandescent light source: $T_{int} = 402\text{ ms}$	Ch0		9			8.1		
		Ch1		4.5			4.2		
(Sensor Lux) / (actual Lux), high gain mode (Note 8)	Fluorescent light source: $T_{int} = 402\text{ ms}$		0.65	1	1.35	0.65	1	1.35	
	Incandescent light source: $T_{int} = 402\text{ ms}$		0.60	1	1.40	0.60	1	1.40	



- NOTES:
2. Optical measurements are made using small-angle incident radiation from light-emitting diode optical sources. Visible 640 nm LEDs and infrared 940 nm LEDs are used for final product testing for compatibility with high-volume production.
 3. The 640 nm irradiance E_e is supplied by an AlInGaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 640$ nm and spectral halfwidth $\Delta\lambda_{1/2} = 17$ nm.
 4. The 940 nm irradiance E_e is supplied by a GaAs light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 940$ nm and spectral halfwidth $\Delta\lambda_{1/2} = 40$ nm.
 5. Integration time T_{int} , is dependent on internal oscillator frequency (f_{osc}) and on the integration field value in the timing register as described in the *Register Set* section. For nominal $f_{osc} = 735$ kHz, nominal $T_{int} = (\text{number of clock cycles})/f_{osc}$.
 Field value 00: $T_{int} = (11 \times 918)/f_{osc} = 13.7$ ms
 Field value 01: $T_{int} = (81 \times 918)/f_{osc} = 101$ ms
 Field value 10: $T_{int} = (322 \times 918)/f_{osc} = 402$ ms
 Scaling between integration times vary proportionally as follows: $11/322 = 0.034$ (field value 00), $81/322 = 0.252$ (field value 01), and $322/322 = 1$ (field value 10).
 6. Full scale ADC count value is limited by the fact that there is a maximum of one count per two oscillator frequency periods and also by a 2-count offset.
 Full scale ADC count value = $((\text{number of clock cycles})/2 - 2)$
 Field value 00: Full scale ADC count value = $((11 \times 918)/2 - 2) = 5047$
 Field value 01: Full scale ADC count value = $((81 \times 918)/2 - 2) = 37177$
 Field value 10: Full scale ADC count value = 65535, which is limited by 16 bit register. This full scale ADC count value is reached for 131074 clock cycles, which occurs for $T_{int} = 178$ ms for nominal $f_{osc} = 735$ kHz.
 7. Low gain mode has 16× lower gain than high gain mode: $(1/16 = 0.0625)$.
 8. The sensor Lux is calculated using the empirical formula shown on p. 22 of this data sheet based on measured Ch0 and Ch1 ADC count values for the light source specified. Actual Lux is obtained with a commercial luxmeter. The range of the (sensor Lux) / (actual Lux) ratio is estimated based on the variation of the 640 nm and 940 nm optical parameters. Devices are not 100% tested with fluorescent or incandescent light sources.

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

AC Electrical Characteristics, $V_{DD} = 3\text{ V}$, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER†		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$t_{(CONV)}$	Conversion time		12	100	400	ms
$f_{(SCL)}$	Clock frequency (I ² C only)		0		400	kHz
	Clock frequency (SMBus only)		10		100	kHz
$t_{(BUF)}$	Bus free time between start and stop condition		1.3			μs
$t_{(HDSTA)}$	Hold time after (repeated) start condition. After this period, the first clock is generated.		0.6			μs
$t_{(SUSTA)}$	Repeated start condition setup time		0.6			μs
$t_{(SUSTO)}$	Stop condition setup time		0.6			μs
$t_{(HDDAT)}$	Data hold time		0		0.9	μs
$t_{(SUDAT)}$	Data setup time		100			ns
$t_{(LOW)}$	SCL clock low period		1.3			μs
$t_{(HIGH)}$	SCL clock high period		0.6			μs
$t_{(TIMEOUT)}$	Detect clock/data low timeout (SMBus only)		25		35	ms
t_F	Clock/data fall time				300	ns
t_R	Clock/data rise time				300	ns
C_i	Input pin capacitance				10	pF

† Specified by design and characterization; not production tested.

PARAMETER MEASUREMENT INFORMATION

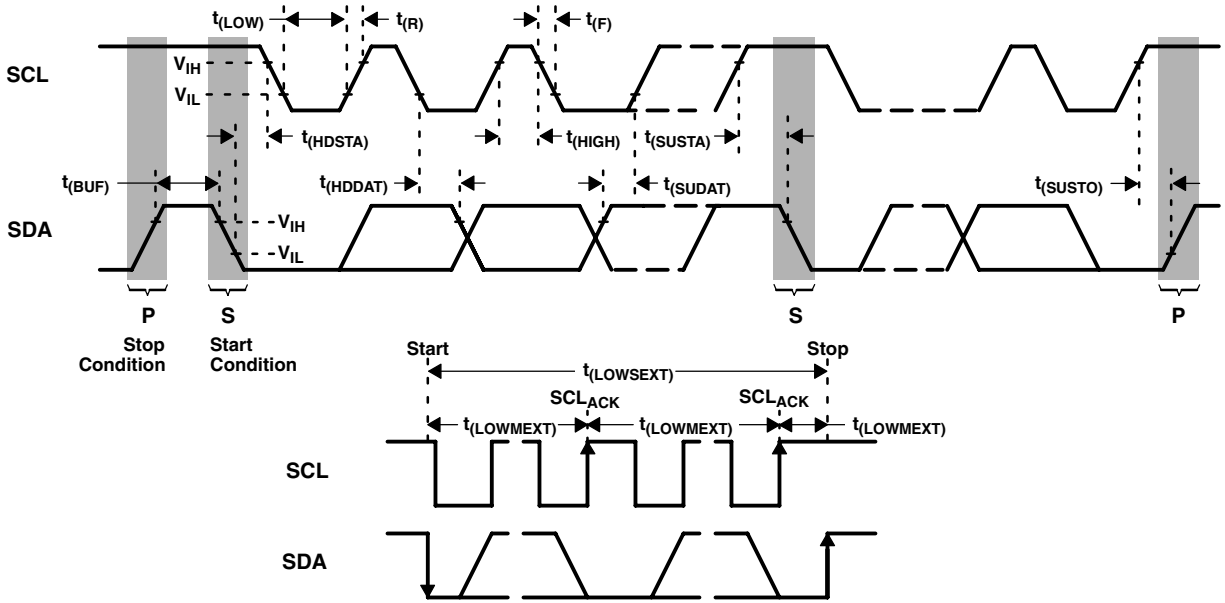


Figure 1. Timing Diagrams

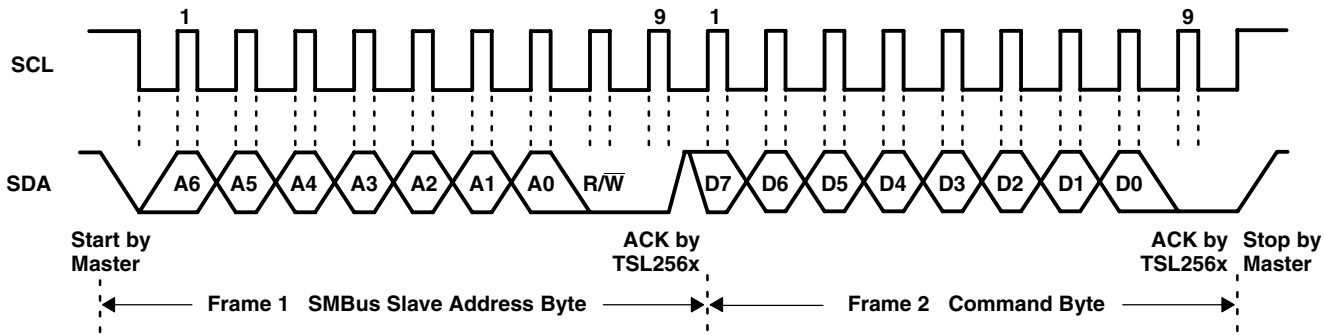


Figure 2. Example Timing Diagram for SMBus Send Byte Format

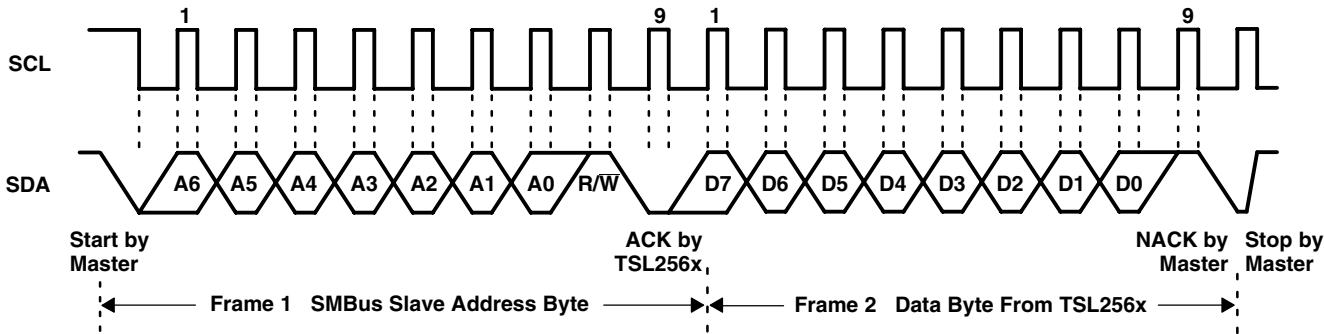


Figure 3. Example Timing Diagram for SMBus Receive Byte Format

TYPICAL CHARACTERISTICS

SPECTRAL RESPONSIVITY

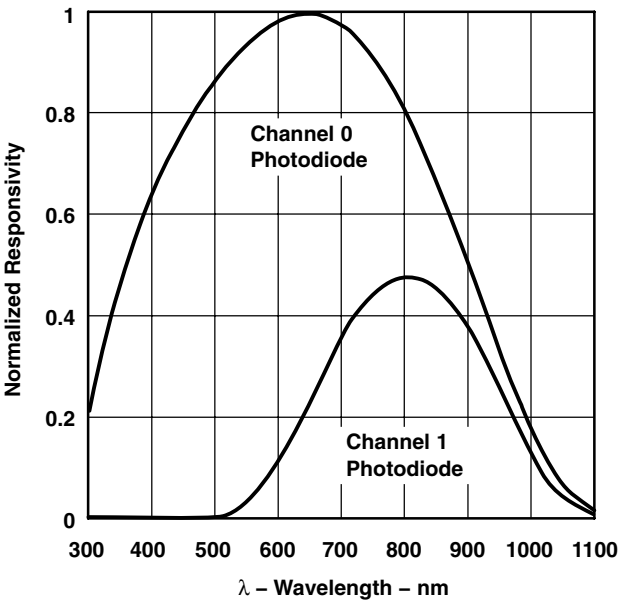


Figure 4

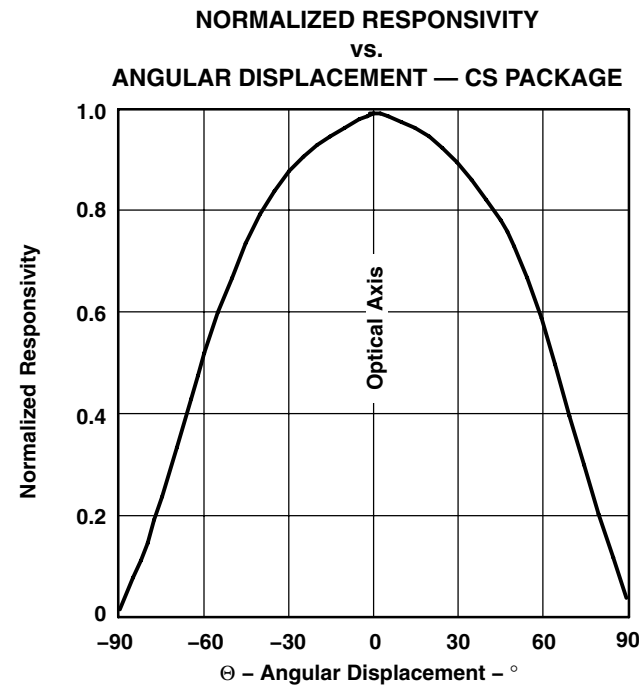


Figure 5

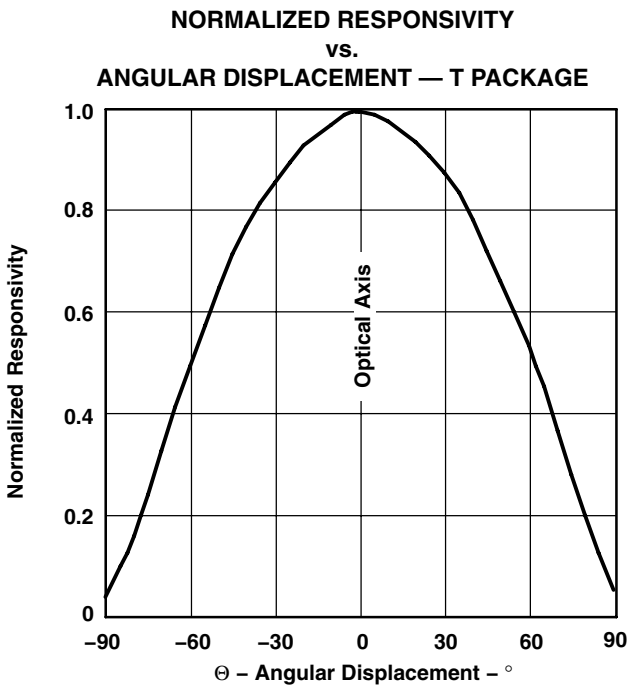


Figure 6

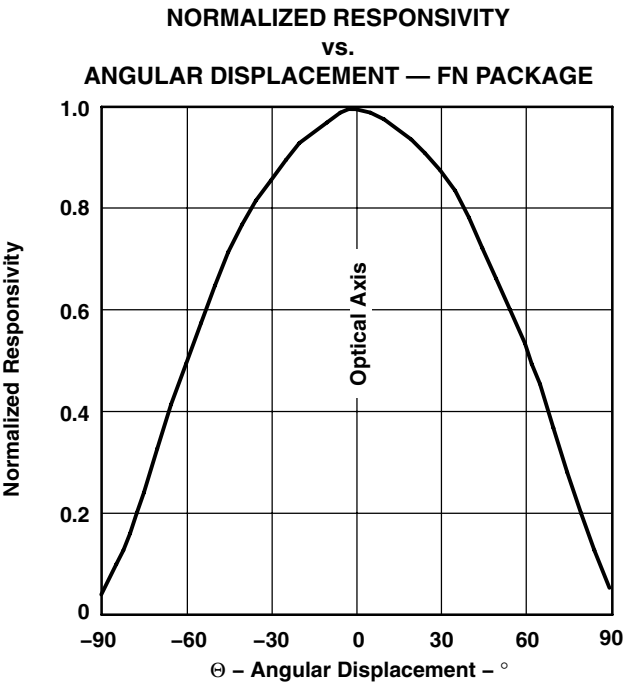


Figure 7

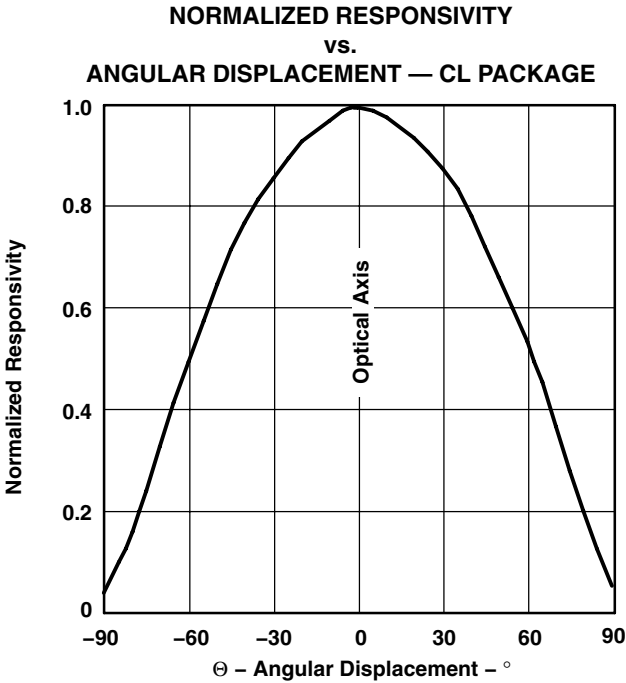


Figure 8

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

PRINCIPLES OF OPERATION

Analog-to-Digital Converter

The TSL256x contains two integrating analog-to-digital converters (ADC) that integrate the currents from the channel 0 and channel 1 photodiodes. Integration of both channels occurs simultaneously, and upon completion of the conversion cycle the conversion result is transferred to the channel 0 and channel 1 data registers, respectively. The transfers are double buffered to ensure that invalid data is not read during the transfer. After the transfer, the device automatically begins the next integration cycle.

Digital Interface

Interface and control of the TSL256x is accomplished through a two-wire serial interface to a set of registers that provide access to device control functions and output data. The serial interface is compatible with System Management Bus (SMBus) versions 1.1 and 2.0, and I²C bus Fast-Mode. The TSL256x offers three slave addresses that are selectable via an external pin (ADDR SEL). The slave address options are shown in Table 1.

Table 1. Slave Address Selection

ADDR SEL TERMINAL LEVEL	SLAVE ADDRESS	SMB ALERT ADDRESS
GND	0101001	0001100
Float	0111001	0001100
VDD	1001001	0001100

NOTE: The Slave and SMB Alert Addresses are 7 bits. Please note the SMBus and I²C protocols on pages 9 through 12. A read/write bit should be appended to the slave address by the master device to properly communicate with the TSL256X device.

SMBus and I²C Protocols

Each *Send* and *Write* protocol is, essentially, a series of bytes. A byte sent to the TSL256x with the most significant bit (MSB) equal to 1 will be interpreted as a COMMAND byte. The lower four bits of the COMMAND byte form the register select address (see Table 2), which is used to select the destination for the subsequent byte(s) received. The TSL256x responds to any Receive Byte requests with the contents of the register specified by the stored register select address.

The TSL256X implements the following protocols of the SMB 2.0 specification:

- Send Byte Protocol
- Receive Byte Protocol
- Write Byte Protocol
- Write Word Protocol
- Read Word Protocol
- Block Write Protocol
- Block Read Protocol

The TSL256X implements the following protocols of the Philips Semiconductor I²C specification:

- I²C Write Protocol
- I²C Read (Combined Format) Protocol

When an SMBus Block Write or Block Read is initiated (see description of COMMAND Register), the byte following the COMMAND byte is ignored but is a requirement of the SMBus specification. This field contains the byte count (i.e. the number of bytes to be transferred). The TSL2560 (SMBus) device ignores this field and extracts this information by counting the actual number of bytes transferred before the Stop condition is detected.

When an I²C Write or I²C Read (Combined Format) is initiated, the byte count is also ignored but follows the SMBus protocol specification. Data bytes continue to be transferred from the TSL2561 (I²C) device to Master until a NACK is sent by the Master.

The data formats supported by the TSL2560 and TSL2561 devices are:

- Master transmitter transmits to slave receiver (SMBus and I²C):
 - The transfer direction in this case is not changed.
- Master reads slave immediately after the first byte (SMBus only):
 - At the moment of the first acknowledgment (provided by the slave receiver) the master transmitter becomes a master receiver and the slave receiver becomes a slave transmitter.
- Combined format (SMBus and I²C):
 - During a change of direction within a transfer, the master repeats both a START condition and the slave address but with the R/W bit reversed. In this case, the master receiver terminates the transfer by generating a NACK on the last byte of the transfer and a STOP condition.

For a complete description of SMBus protocols, please review the SMBus Specification at <http://www.smbus.org/specs>. For a complete description of I²C protocols, please review the I²C Specification at <http://www.semiconductors.philips.com>.

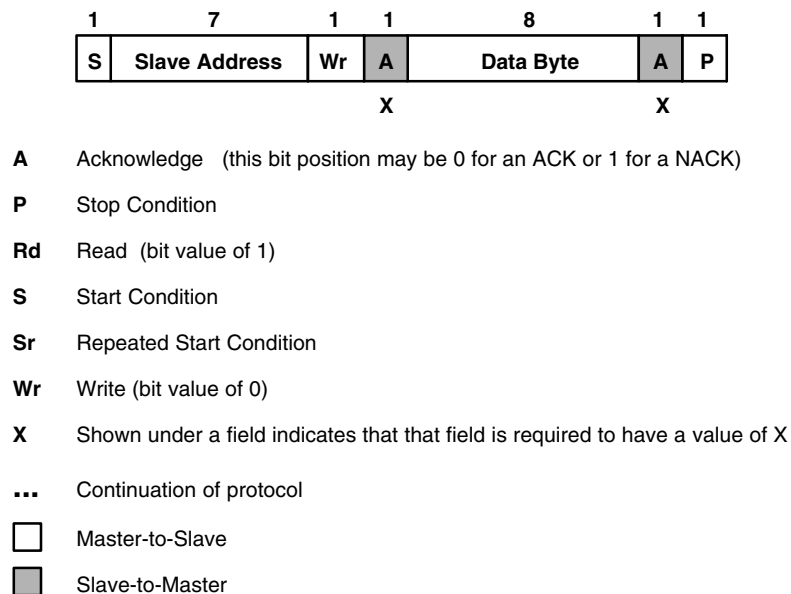


Figure 9. SMBus and I²C Packet Protocol Element Key

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

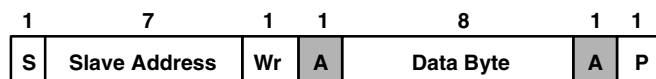


Figure 10. SMBus Send Byte Protocol

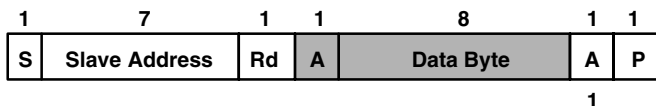


Figure 11. SMBus Receive Byte Protocol

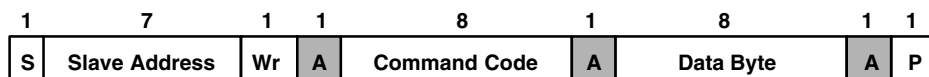


Figure 12. SMBus Write Byte Protocol

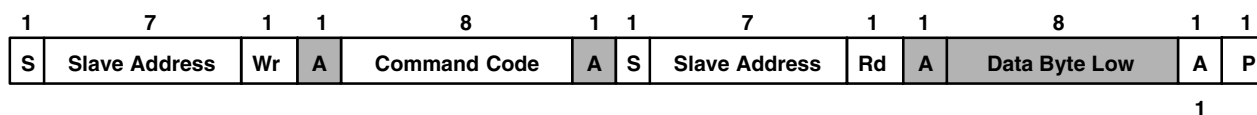


Figure 13. SMBus Read Byte Protocol

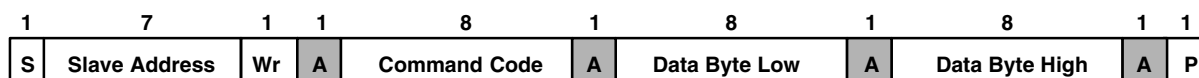


Figure 14. SMBus Write Word Protocol

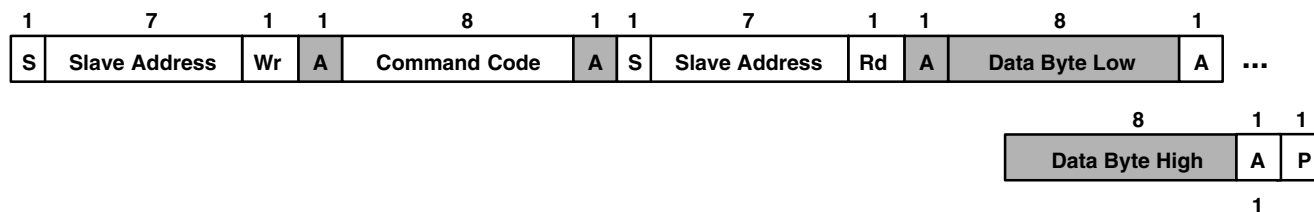


Figure 15. SMBus Read Word Protocol



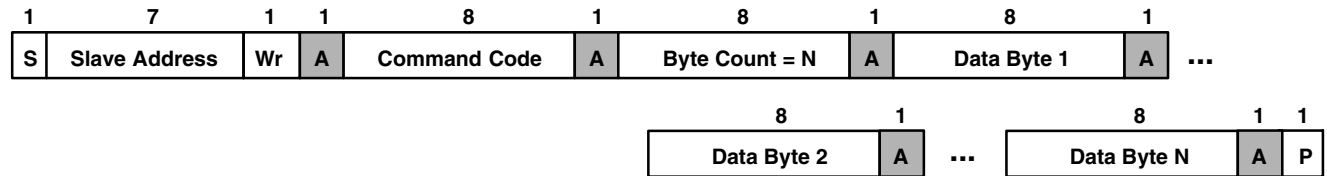


Figure 16. SMBus Block Write or I²C Write Protocols

NOTE: The I²C write protocol does not use the Byte Count packet, and the Master will continue sending Data Bytes until the Master initiates a Stop condition. See the Command Register on page 13 for additional information regarding the Block Read/Write protocol.

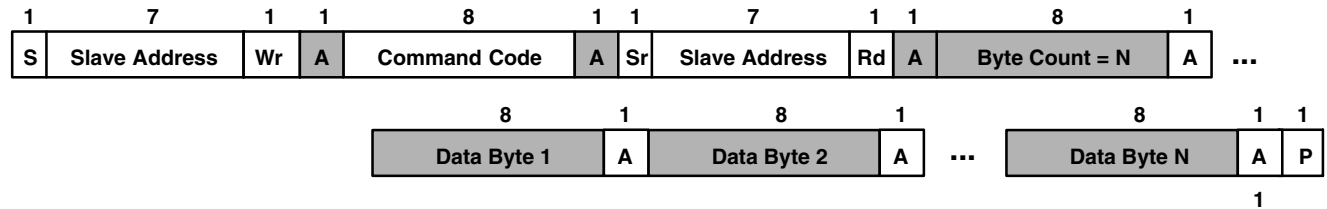


Figure 17. SMBus Block Read or I²C Read (Combined Format) Protocols

NOTE: The I²C read protocol does not use the Byte Count packet, and the Master will continue receiving Data Bytes until the Master initiates a Stop Condition. See the Command Register on page 13 for additional information regarding the Block Read/Write protocol.

Register Set

The TSL256x is controlled and monitored by sixteen registers (three are reserved) and a command register accessed through the serial interface. These registers provide for a variety of control functions and can be read to determine results of the ADC conversions. The register set is summarized in Table 2.

Table 2. Register Address

ADDRESS	REGISTER NAME	REGISTER FUNCTION
--	COMMAND	Specifies register address
0h	CONTROL	Control of basic functions
1h	TIMING	Integration time/gain control
2h	THRESHLOWLOW	Low byte of low interrupt threshold
3h	THRESHLOWHIGH	High byte of low interrupt threshold
4h	THRESHHIGHLOW	Low byte of high interrupt threshold
5h	THRESHHIGHHIGH	High byte of high interrupt threshold
6h	INTERRUPT	Interrupt control
7h	--	Reserved
8h	CRC	Factory test — not a user register
9h	--	Reserved
Ah	ID	Part number/ Rev ID
Bh	--	Reserved
Ch	DATA0LOW	Low byte of ADC channel 0
Dh	DATA0HIGH	High byte of ADC channel 0
Eh	DATA1LOW	Low byte of ADC channel 1
Fh	DATA1HIGH	High byte of ADC channel 1

The mechanics of accessing a specific register depends on the specific SMB protocol used. Refer to the section on SMBus protocols. In general, the COMMAND register is written first to specify the specific control/status register for following read/write operations.

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

Command Register

The command register specifies the address of the target register for subsequent read and write operations. The Send Byte protocol is used to configure the COMMAND register. The command register contains eight bits as described in Table 3. The command register defaults to 00h at power on.

Table 3. Command Register

	7	6	5	4	3	2	1	0	
	CMD	CLEAR	WORD	BLOCK	ADDRESS				COMMAND
Reset Value:	0	0	0	0	0	0	0	0	
FIELD	BIT	DESCRIPTION							
CMD	7	Select command register. Must write as 1.							
CLEAR	6	Interrupt clear. Clears any pending interrupt. This bit is a write-one-to-clear bit. It is self clearing.							
WORD	5	SMB Write/Read Word Protocol. 1 indicates that this SMB transaction is using either the SMB Write Word or Read Word protocol.							
BLOCK	4	Block Write/Read Protocol. 1 indicates that this transaction is using either the Block Write or the Block Read protocol. See Note below.							
ADDRESS	3:0	Register Address. This field selects the specific control or status register for following write and read commands according to Table 2.							

NOTE: An I²C block transaction will continue until the Master sends a stop condition. See Figure 16 and Figure 17. Unlike the I²C protocol, the SMBus read/write protocol requires a Byte Count. All four ADC Channel Data Registers (Ch through Fh) can be read simultaneously in a single SMBus transaction. This is the only 32-bit data block supported by the TSL2560 SMBus protocol. The BLOCK bit must be set to 1, and a read condition should be initiated with a COMMAND CODE of 9Bh. By using a COMMAND CODE of 9Bh during an SMBus Block Read Protocol, the TSL2560 device will automatically insert the appropriate Byte Count (Byte Count = 4) as illustrated in Figure 17. A write condition should not be used in conjunction with the Bh register.

Control Register (0h)

The CONTROL register contains two bits and is primarily used to power the TSL256x device up and down as shown in Table 4.

Table 4. Control Register

	7	6	5	4	3	2	1	0	
0h	Resv	Resv	Resv	Resv	Resv	Resv	POWER		CONTROL
Reset Value:	0	0	0	0	0	0	0	0	
FIELD	BIT	DESCRIPTION							
Resv	7:2	Reserved. Write as 0.							
POWER	1:0	Power up/power down. By writing a 03h to this register, the device is powered up. By writing a 00h to this register, the device is powered down. NOTE: If a value of 03h is written, the value returned during a read cycle will be 03h. This feature can be used to verify that the device is communicating properly.							

Timing Register (1h)

The TIMING register controls both the integration time and the gain of the ADC channels. A common set of control bits is provided that controls both ADC channels. The TIMING register defaults to 02h at power on.

Table 5. Timing Register

	7	6	5	4	3	2	1	0	
1h	Resv	Resv	Resv	GAIN	Manual	Resv	INTEG		TIMING
Reset Value:	0	0	0	0	0	0	1	0	
FIELD	BIT	DESCRIPTION							
Resv	7–5	Reserved. Write as 0.							
GAIN	4	Switches gain between low gain and high gain modes. Writing a 0 selects low gain (1×); writing a 1 selects high gain (16×).							
Manual	3	Manual timing control. Writing a 1 begins an integration cycle. Writing a 0 stops an integration cycle. NOTE: This field only has meaning when INTEG = 11. It is ignored at all other times.							
Resv	2	Reserved. Write as 0.							
INTEG	1:0	Integrate time. This field selects the integration time for each conversion.							

Integration time is dependent on the INTEG FIELD VALUE and the internal clock frequency. Nominal integration times and respective scaling between integration times scale proportionally as shown in Table 6. See Note 5 and Note 6 on page 5 for detailed information regarding how the scale values were obtained; see page 22 for further information on how to calculate lux.

Table 6. Integration Time

INTEG FIELD VALUE	SCALE	NOMINAL INTEGRATION TIME
00	0.034	13.7 ms
01	0.252	101 ms
10	1	402 ms
11	--	N/A

The manual timing control feature is used to manually start and stop the integration time period. If a particular integration time period is required that is not listed in Table 6, then this feature can be used. For example, the manual timing control can be used to synchronize the TSL256x device with an external light source (e.g. LED). A start command to begin integration can be initiated by writing a 1 to this bit field. Correspondingly, the integration can be stopped by simply writing a 0 to the same bit field.

Interrupt Threshold Register (2h – 5h)

The interrupt threshold registers store the values to be used as the high and low trigger points for the comparison function for interrupt generation. If the value generated by channel 0 crosses below or is equal to the low threshold specified, an interrupt is asserted on the interrupt pin. If the value generated by channel 0 crosses above the high threshold specified, an interrupt is asserted on the interrupt pin. Registers THRESHLOWLOW and THRESHLOWHIGH provide the low byte and high byte, respectively, of the lower interrupt threshold. Registers THRESHHIGHLOW and THRESHHIGHHIGH provide the low and high bytes, respectively, of the upper interrupt threshold. The high and low bytes from each set of registers are combined to form a 16-bit threshold value. The interrupt threshold registers default to 00h on power up.

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

Table 7. Interrupt Threshold Register

REGISTER	ADDRESS	BITS	DESCRIPTION
THRESHLOWLOW	2h	7:0	ADC channel 0 lower byte of the low threshold
THRESHLOWHIGH	3h	7:0	ADC channel 0 upper byte of the low threshold
THRESHHIGHLOW	4h	7:0	ADC channel 0 lower byte of the high threshold
THRESHHIGHHIGH	5h	7:0	ADC channel 0 upper byte of the high threshold

NOTE: Since two 8-bit values are combined for a single 16-bit value for each of the high and low interrupt thresholds, the Send Byte protocol should not be used to write to these registers. Any values transferred by the Send Byte protocol with the MSB set would be interpreted as the COMMAND field and stored as an address for subsequent read/write operations and not as the interrupt threshold information as desired. The Write Word protocol should be used to write byte-paired registers. For example, the THRESHLOWLOW and THRESHLOWHIGH registers (as well as the THRESHHIGHLOW and THRESHHIGHHIGH registers) can be written together to set the 16-bit ADC value in a single transaction.

Interrupt Control Register (6h)

The INTERRUPT register controls the extensive interrupt capabilities of the TSL256x. The TSL256x permits both SMB-Alert style interrupts as well as traditional level-style interrupts. The interrupt persist bit field (PERSIST) provides control over when interrupts occur. A value of 0 causes an interrupt to occur after every integration cycle regardless of the threshold settings. A value of 1 results in an interrupt after one integration time period outside the threshold window. A value of N (where N is 2 through 15) results in an interrupt only if the value remains outside the threshold window for N consecutive integration cycles. For example, if N is equal to 10 and the integration time is 402 ms, then the total time is approximately 4 seconds.

When a level Interrupt is selected, an interrupt is generated whenever the last conversion results in a value outside of the programmed threshold window. The interrupt is active-low and remains asserted until cleared by writing the COMMAND register with the CLEAR bit set.

In SMBAlert mode, the interrupt is similar to the traditional level style and the interrupt line is asserted low. To clear the interrupt, the host responds to the SMBAlert by performing a modified Receive Byte operation, in which the Alert Response Address (ARA) is placed in the slave address field, and the TSL256x that generated the interrupt responds by returning its own address in the seven most significant bits of the receive data byte. If more than one device connected on the bus has pulled the SMBAlert line low, the highest priority (lowest address) device will win communication rights via standard arbitration during the slave address transfer. If the device loses this arbitration, the interrupt will not be cleared. The Alert Response Address is 0Ch.

When INTR = 11, the interrupt is generated immediately following the SMBus write operation. Operation then behaves in an SMBAlert mode, and the *software set* interrupt may be cleared by an SMBAlert cycle.

NOTE: Interrupts are based on the value of Channel 0 only.

Table 8. Interrupt Control Register

	7	6	5	4	3	2	1	0	
6h	Resv	Resv	INTR		PERSIST				INTERRUPT
Reset Value:	0	0	0	0	0	0	0	0	
FIELD	BITS		DESCRIPTION						
Resv	7:6		Reserved. Write as 0.						
INTR	5:4		INTR Control Select. This field determines mode of interrupt logic according to Table 9, below.						
PERSIST	3:0		Interrupt persistence. Controls rate of interrupts to the host processor as shown in Table 10, below.						

Table 9. Interrupt Control Select

INTR FIELD VALUE	READ VALUE
00	Interrupt output disabled
01	Level Interrupt
10	SMBAlert compliant
11	Test Mode: Sets interrupt and functions as mode 10

NOTE: Field value of 11 may be used to test interrupt connectivity in a system or to assist in debugging interrupt service routine software.

Table 10. Interrupt Persistence Select

PERSIST FIELD VALUE	INTERRUPT PERSIST FUNCTION
0000	Every ADC cycle generates interrupt
0001	Any value outside of threshold range
0010	2 integration time periods out of range
0011	3 integration time periods out of range
0100	4 integration time periods out of range
0101	5 integration time periods out of range
0110	6 integration time periods out of range
0111	7 integration time periods out of range
1000	8 integration time periods out of range
1001	9 integration time periods out of range
1010	10 integration time periods out of range
1011	11 integration time periods out of range
1100	12 integration time periods out of range
1101	13 integration time periods out of range
1110	14 integration time periods out of range
1111	15 integration time periods out of range

ID Register (Ah)

The ID register provides the value for both the part number and silicon revision number for that part number. It is a read-only register, whose value never changes.

Table 11. ID Register

	7	6	5	4	3	2	1	0	
Ah	PARTNO				REVNO				ID
Reset Value:	–	–	–	–	–	–	–	–	
FIELD	BITS	DESCRIPTION							
PARTNO	7:4	Part Number Identification: field value 0000 = TSL2560, field value 0001 = TSL2561							
REVNO	3:0	Revision number identification							

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

ADC Channel Data Registers (Ch – Fh)

The ADC channel data are expressed as 16-bit values spread across two registers. The ADC channel 0 data registers, DATA0LOW and DATA0HIGH provide the lower and upper bytes, respectively, of the ADC value of channel 0. Registers DATA1LOW and DATA1HIGH provide the lower and upper bytes, respectively, of the ADC value of channel 1. All channel data registers are read-only and default to 00h on power up.

Table 12. ADC Channel Data Registers

REGISTER	ADDRESS	BITS	DESCRIPTION
DATA0LOW	Ch	7:0	ADC channel 0 lower byte
DATA0HIGH	Dh	7:0	ADC channel 0 upper byte
DATA1LOW	Eh	7:0	ADC channel 1 lower byte
DATA1HIGH	Fh	7:0	ADC channel 1 upper byte

The upper byte data registers can only be read following a read to the corresponding lower byte register. When the lower byte register is read, the upper eight bits are strobed into a shadow register, which is read by a subsequent read to the upper byte. The upper register will read the correct value even if additional ADC integration cycles end between the reading of the lower and upper registers.

NOTE: The Read Word protocol can be used to read byte-paired registers. For example, the DATA0LOW and DATA0HIGH registers (as well as the DATA1LOW and DATA1HIGH registers) may be read together to obtain the 16-bit ADC value in a single transaction

APPLICATION INFORMATION: SOFTWARE

Basic Operation

After applying V_{DD} , the device will initially be in the power-down state. To operate the device, issue a command to access the CONTROL register followed by the data value 03h to power up the device. At this point, both ADC channels will begin a conversion at the default integration time of 400 ms. After 400 ms, the conversion results will be available in the DATA0 and DATA1 registers. Use the following pseudo code to read the data registers:

```
// Read ADC Channels Using Read Word Protocol - RECOMMENDED
    Address = 0x39                                //Slave addr - also 0x29 or 0x49

    //Address the Ch0 lower data register and configure for Read Word
    Command = 0xAC                                //Set Command bit and Word bit

    //Reads two bytes from sequential registers 0x0C and 0x0D
    //Results are returned in DataLow and DataHigh variables
    ReadWord (Address, Command, DataLow, DataHigh)
    Channel0 = 256 * DataHigh + DataLow

    //Address the Ch1 lower data register and configure for Read Word
    Command = 0xAE                                //Set bit fields 7 and 5

    //Reads two bytes from sequential registers 0x0E and 0x0F
    //Results are returned in DataLow and DataHigh variables
    ReadWord (Address, Command, DataLow, DataHigh)
    Channel1 = 256 * DataHigh + DataLow            //Shift DataHigh to upper byte

// Read ADC Channels Using Read Byte Protocol
    Address = 0x39                                //Slave addr - also 0x29 or 0x49
    Command = 0x8C                                //Address the Ch0 lower data register
    ReadByte (Address, Command, DataLow)           //Result returned in DataLow
    Command = 0x8D                                //Address the Ch0 upper data register
    ReadByte (Address, Command, DataHigh)          //Result returned in DataHigh
    Channel0 = 256 * DataHigh + DataLow            //Shift DataHigh to upper byte

    Command = 0x8E                                //Address the Ch1 lower data register
    ReadByte (Address, Command, DataLow)           //Result returned in DataLow
    Command = 0x8F                                //Address the Ch1 upper data register
    ReadByte (Address, Command, DataHigh)          //Result returned in DataHigh
    Channel1 = 256 * DataHigh + DataLow            //Shift DataHigh to upper byte
```

APPLICATION INFORMATION: SOFTWARE

Configuring the Timing Register

The command, timing, and control registers are initialized to default values on power up. Setting these registers to the desired values would be part of a normal initialization or setup procedure. In addition, to maximize the performance of the device under various conditions, the integration time and gain may be changed often during operation. The following pseudo code illustrates a procedure for setting up the timing register for various options:

```
// Set up Timing Register
//Low Gain (1x), integration time of 402ms (default value)
Address = 0x39
Command = 0x81
Data = 0x02
WriteByte(Address, Command, Data)

//Low Gain (1x), integration time of 101ms
Data = 0x01
WriteByte(Address, Command, Data)

//Low Gain (1x), integration time of 13.7ms
Data = 0x00
WriteByte(Address, Command, Data)

//High Gain (16x), integration time of 101ms
Data = 0x11
WriteByte(Address, Command, Data)

//Read data registers (see Basic Operation example)

//Perform Manual Integration
//Set up for manual integration with Gain of 1x
Data = 0x03
//Set manual integration mode - device stops converting
WriteByte(Address, Command, Data)

//Begin integration period
Data = 0x0B
WriteByte(Address, Command, Data)

//Integrate for 50ms
Sleep (50)                                //Wait for 50ms

//Stop integrating
Data = 0x03
WriteByte(Address, Command, Data)

//Read data registers (see Basic Operation example)
```



APPLICATION INFORMATION: SOFTWARE

Interrupts

The interrupt feature of the TSL256x device simplifies and improves system efficiency by eliminating the need to poll the sensor for a light intensity value. Interrupt styles are determined by the INTR field in the Interrupt Register. The interrupt feature may be disabled by writing a field value of 00h to the Interrupt Control Register so that polling can be performed.

The versatility of the interrupt feature provides many options for interrupt configuration and usage. The primary purpose of the interrupt function is to provide a meaningful change in light intensity. However, it also be used as an end-of-conversion signal. The concept of a *meaningful change* can be defined by the user both in terms of light intensity and time, or persistence, of that change in intensity. The TSL256x device implements two 16-bit-wide interrupt threshold registers that allow the user to define a threshold above and below the current light level. An interrupt will then be generated when the value of a conversion exceeds either of these limits. For simplicity of programming, the threshold comparison is accomplished only with Channel 0. This simplifies calculation of thresholds that are based, for example, on a percent of the current light level. It is adequate to use only one channel when calculating light intensity differences since, for a given light source, the channel 0 and channel 1 values are linearly proportional to each other and thus both values scale linearly with light intensity.

To further control when an interrupt occurs, the TSL256x device provides an interrupt persistence feature. This feature allows the user to specify a number of conversion cycles for which a light intensity exceeding either interrupt threshold must persist before actually generating an interrupt. This can be used to prevent transient changes in light intensity from generating an unwanted interrupt. With a value of 1, an interrupt occurs immediately whenever either threshold is exceeded. With values of N , where N can range from 2 to 15, N consecutive conversions must result in values outside the interrupt window for an interrupt to be generated. For example, if N is equal to 10 and the integration time is 402 ms, then an interrupt will not be generated unless the light level persists for more than 4 seconds outside the threshold.

Two different interrupt styles are available: Level and SMBus Alert. The difference between these two interrupt styles is how they are cleared. Both result in the interrupt line going active low and remaining low until the interrupt is cleared. A level style interrupt is cleared by setting the CLEAR bit (bit 6) in the COMMAND register. The SMBus Alert style interrupt is cleared by an Alert Response as described in the Interrupt Control Register section and SMBus specification.

To configure the interrupt as an end-of-conversion signal, the interrupt PERSIST field is set to 0. Either Level or SMBus Alert style can be used. An interrupt will be generated upon completion of each conversion. The interrupt threshold registers are ignored. The following example illustrates the configuration of a level interrupt:

```
// Set up end-of-conversion interrupt, Level style
Address = 0x39                      //Slave addr also 0x29 or 0x49
Command = 0x86                      //Address Interrupt Register
Data = 0x10                         //Level style, every ADC cycle
WriteByte(Address, Command, Data)
```

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

APPLICATION INFORMATION: SOFTWARE

The following example pseudo code illustrates the configuration of an SMB Alert style interrupt when the light intensity changes 20% from the current value, and persists for 3 conversion cycles:

```
// Read current light level
Address = 0x39                      //Slave addr also 0x29 or 0x49
Command = 0xAC                      //Set Command bit and Word bit
ReadWord (Address, Command, DataLow, DataHigh)
Channel0 = (256 * DataHigh) + DataLow

//Calculate upper and lower thresholds
T_Upper = Channel0 + (0.2 * Channel0)
T_Lower = Channel0 - (0.2 * Channel0)

//Write the lower threshold register
Command = 0xA2                      //Addr lower threshold reg, set Word Bit
WriteWord (Address, Command, T_Lower.LoByte, T_Lower.HiByte)

//Write the upper threshold register
Command = 0xA4                      //Addr upper threshold reg, set Word bit
WriteWord (Address, Command, T_Upper.LoByte, T_Upper.HiByte)

//Enable interrupt
Command = 0x86                      //Address interrupt register
Data = 0x23                        //SMBAlert style, PERSIST = 3
WriteByte(Address, Command, Data)
```

In order to generate an interrupt on demand during system test or debug, a test mode (INTR = 11) can be used. The following example illustrates how to generate an interrupt on demand:

```
// Generate an interrupt
Address = 0x39                      //Slave addr also 0x29 or 0x49
Command = 0x86                      //Address Interrupt register
Data = 0x30                        //Test interrupt
WriteByte(Address, Command, Data)

//Interrupt line should now be low
```



APPLICATION INFORMATION: SOFTWARE

Calculating Lux

The TSL256x is intended for use in ambient light detection applications such as display backlight control, where adjustments are made to display brightness or contrast based on the brightness of the ambient light, as perceived by the human eye. Conventional silicon detectors respond strongly to infrared light, which the human eye does not see. This can lead to significant error when the infrared content of the ambient light is high, such as with incandescent lighting, due to the difference between the silicon detector response and the brightness perceived by the human eye.

This problem is overcome in the TSL256x through the use of two photodiodes. One of the photodiodes (channel 0) is sensitive to both visible and infrared light, while the second photodiode (channel 1) is sensitive primarily to infrared light. An integrating ADC converts the photodiode currents to digital outputs. Channel 1 digital output is used to compensate for the effect of the infrared component of light on the channel 0 digital output. The ADC digital outputs from the two channels are used in a formula to obtain a value that approximates the human eye response in the commonly used Illuminance unit of Lux:

CS Package

For $0 < CH1/CH0 \leq 0.52$	$Lux = 0.0315 \times CH0 - 0.0593 \times CH0 \times ((CH1/CH0)^{1.4})$
For $0.52 < CH1/CH0 \leq 0.65$	$Lux = 0.0229 \times CH0 - 0.0291 \times CH1$
For $0.65 < CH1/CH0 \leq 0.80$	$Lux = 0.0157 \times CH0 - 0.0180 \times CH1$
For $0.80 < CH1/CH0 \leq 1.30$	$Lux = 0.00338 \times CH0 - 0.00260 \times CH1$
For $CH1/CH0 > 1.30$	$Lux = 0$

T, FN, and CL Package

For $0 < CH1/CH0 \leq 0.50$	$Lux = 0.0304 \times CH0 - 0.062 \times CH0 \times ((CH1/CH0)^{1.4})$
For $0.50 < CH1/CH0 \leq 0.61$	$Lux = 0.0224 \times CH0 - 0.031 \times CH1$
For $0.61 < CH1/CH0 \leq 0.80$	$Lux = 0.0128 \times CH0 - 0.0153 \times CH1$
For $0.80 < CH1/CH0 \leq 1.30$	$Lux = 0.00146 \times CH0 - 0.00112 \times CH1$
For $CH1/CH0 > 1.30$	$Lux = 0$

The formulas shown above were obtained by optical testing with fluorescent and incandescent light sources, and apply only to open-air applications. Optical apertures (e.g. light pipes) will affect the incident light on the device.

Simplified Lux Calculation

Below is the argument and return value including source code (shown on following page) for calculating lux. The source code is intended for embedded and/or microcontroller applications. Two individual code sets are provided, one for the T, FN, and CL packages, and one for the CS package. All floating point arithmetic operations have been eliminated since embedded controllers and microcontrollers generally do not support these types of operations. Since floating point has been removed, scaling must be performed prior to calculating illuminance if the integration time is not 402 ms and/or if the gain is not 16× as denoted in the source code on the following pages. This sequence scales first to mitigate rounding errors induced by decimal math.

```
extern unsigned int CalculateLux(unsigned int iGain, unsigned int tInt, unsigned int
    ch0, unsigned int ch1, int iType)
```

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

```
//*****  
//  
// Copyright © 2004-2005 TAOS, Inc.  
//  
// THIS CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY  
// KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE  
// IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR  
// PURPOSE.  
//  
// Module Name:  
// lux.cpp  
//  
//*****  
  
#define LUX_SCALE 14 // scale by 2^14  
#define RATIO_SCALE 9 // scale ratio by 2^9  
  
//-----  
// Integration time scaling factors  
//-----  
  
#define CH_SCALE 10 // scale channel values by 2^10  
#define CHSCALE_TINT0 0x7517 // 322/11 * 2^CH_SCALE  
#define CHSCALE_TINT1 0x0fe7 // 322/81 * 2^CH_SCALE  
  
//-----  
// T, FN, and CL Package coefficients  
//-----  
// For Ch1/Ch0=0.00 to 0.50  
// Lux/Ch0=0.0304-0.062*((Ch1/Ch0)^1.4)  
// piecewise approximation  
// For Ch1/Ch0=0.00 to 0.125:  
// Lux/Ch0=0.0304-0.0272*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.125 to 0.250:  
// Lux/Ch0=0.0325-0.0440*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.250 to 0.375:  
// Lux/Ch0=0.0351-0.0544*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.375 to 0.50:  
// Lux/Ch0=0.0381-0.0624*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.50 to 0.61:  
// Lux/Ch0=0.0224-0.031*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.61 to 0.80:  
// Lux/Ch0=0.0128-0.0153*(Ch1/Ch0)  
//  
// For Ch1/Ch0=0.80 to 1.30:  
// Lux/Ch0=0.00146-0.00112*(Ch1/Ch0)  
//  
// For Ch1/Ch0>1.3:  
// Lux/Ch0=0  
//-----  
#define K1T 0x0040 // 0.125 * 2^RATIO_SCALE  
#define B1T 0x01f2 // 0.0304 * 2^LUX_SCALE  
#define M1T 0x01be // 0.0272 * 2^LUX_SCALE  
  
#define K2T 0x0080 // 0.250 * 2^RATIO_SCALE
```



```
#define B2T 0x0214 // 0.0325 * 2^LUX_SCALE
#define M2T 0x02d1 // 0.0440 * 2^LUX_SCALE

#define K3T 0x00c0 // 0.375 * 2^RATIO_SCALE
#define B3T 0x023f // 0.0351 * 2^LUX_SCALE
#define M3T 0x037b // 0.0544 * 2^LUX_SCALE

#define K4T 0x0100 // 0.50 * 2^RATIO_SCALE
#define B4T 0x0270 // 0.0381 * 2^LUX_SCALE
#define M4T 0x03fe // 0.0624 * 2^LUX_SCALE
#define K5T 0x0138 // 0.61 * 2^RATIO_SCALE
#define B5T 0x016f // 0.0224 * 2^LUX_SCALE
#define M5T 0x01fc // 0.0310 * 2^LUX_SCALE

#define K6T 0x019a // 0.80 * 2^RATIO_SCALE
#define B6T 0x00d2 // 0.0128 * 2^LUX_SCALE
#define M6T 0x00fb // 0.0153 * 2^LUX_SCALE

#define K7T 0x029a // 1.3 * 2^RATIO_SCALE
#define B7T 0x0018 // 0.00146 * 2^LUX_SCALE
#define M7T 0x0012 // 0.00112 * 2^LUX_SCALE

#define K8T 0x029a // 1.3 * 2^RATIO_SCALE
#define B8T 0x0000 // 0.000 * 2^LUX_SCALE
#define M8T 0x0000 // 0.000 * 2^LUX_SCALE

//-----
// CS package coefficients
//-----
// For 0 <= Ch1/Ch0 <= 0.52
//     Lux/Ch0 = 0.0315-0.0593*((Ch1/Ch0)^1.4)
//     piecewise approximation
//         For 0 <= Ch1/Ch0 <= 0.13
//             Lux/Ch0 = 0.0315-0.0262*(Ch1/Ch0)
//         For 0.13 <= Ch1/Ch0 <= 0.26
//             Lux/Ch0 = 0.0337-0.0430*(Ch1/Ch0)
//         For 0.26 <= Ch1/Ch0 <= 0.39
//             Lux/Ch0 = 0.0363-0.0529*(Ch1/Ch0)
//         For 0.39 <= Ch1/Ch0 <= 0.52
//             Lux/Ch0 = 0.0392-0.0605*(Ch1/Ch0)
// For 0.52 < Ch1/Ch0 <= 0.65
//     Lux/Ch0 = 0.0229-0.0291*(Ch1/Ch0)
// For 0.65 < Ch1/Ch0 <= 0.80
//     Lux/Ch0 = 0.00157-0.00180*(Ch1/Ch0)
// For 0.80 < Ch1/Ch0 <= 1.30
//     Lux/Ch0 = 0.00338-0.00260*(Ch1/Ch0)
// For Ch1/Ch0 > 1.30
//     Lux = 0
//-----
#define K1C 0x0043 // 0.130 * 2^RATIO_SCALE
#define B1C 0x0204 // 0.0315 * 2^LUX_SCALE
#define M1C 0x01ad // 0.0262 * 2^LUX_SCALE

#define K2C 0x0085 // 0.260 * 2^RATIO_SCALE
#define B2C 0x0228 // 0.0337 * 2^LUX_SCALE
#define M2C 0x02c1 // 0.0430 * 2^LUX_SCALE

#define K3C 0x00c8 // 0.390 * 2^RATIO_SCALE
#define B3C 0x0253 // 0.0363 * 2^LUX_SCALE
#define M3C 0x0363 // 0.0529 * 2^LUX_SCALE
```

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

```
#define K4C    0x010a // 0.520 * 2^RATIO_SCALE
#define B4C    0x0282 // 0.0392 * 2^LUX_SCALE
#define M4C    0x03df // 0.0605 * 2^LUX_SCALE

#define K5C    0x014d // 0.65 * 2^RATIO_SCALE
#define B5C    0x0177 // 0.0229 * 2^LUX_SCALE
#define M5C    0x01dd // 0.0291 * 2^LUX_SCALE

#define K6C    0x019a // 0.80 * 2^RATIO_SCALE
#define B6C    0x0101 // 0.0157 * 2^LUX_SCALE
#define M6C    0x0127 // 0.0180 * 2^LUX_SCALE

#define K7C    0x029a // 1.3 * 2^RATIO_SCALE
#define B7C    0x0037 // 0.00338 * 2^LUX_SCALE
#define M7C    0x002b // 0.00260 * 2^LUX_SCALE

#define K8C    0x029a // 1.3 * 2^RATIO_SCALE
#define B8C    0x0000 // 0.000 * 2^LUX_SCALE
#define M8C    0x0000 // 0.000 * 2^LUX_SCALE

// lux equation approximation without floating point calculations
//
// Routine:      unsigned int CalculateLux(unsigned int ch0, unsigned int ch1, int iType)
//
// Description:  Calculate the approximate illuminance (lux) given the raw
//               channel values of the TSL2560. The equation is implemented
//               as a piece-wise linear approximation.
//
// Arguments:    unsigned int iGain - gain, where 0:1X, 1:16X
//               unsigned int tInt - integration time, where 0:13.7mS, 1:100mS, 2:402mS,
//               3:Manual
//               unsigned int ch0 - raw channel value from channel 0 of TSL2560
//               unsigned int ch1 - raw channel value from channel 1 of TSL2560
//               unsigned int iType - package type (T or CS)
//
// Return:       unsigned int - the approximate illuminance (lux)
//
//
// unsigned int CalculateLux(unsigned int iGain, unsigned int tInt, unsigned int ch0,
//               unsigned int ch1, int iType)
//
// {
//
//     //-----
//     // first, scale the channel values depending on the gain and integration time
//     // 16X, 402mS is nominal.
//     // scale if integration time is NOT 402 msec
//     unsigned long chScale;
//     unsigned long channel1;
//     unsigned long channel0;
//     switch (tInt)
//     {
//         case 0:    // 13.7 msec
//             chScale = CHSCALE_TINT0;
//             break;
//         case 1:    // 101 msec
//             chScale = CHSCALE_TINT1;
//             break;
//         default:    // assume no scaling
//             chScale = (1 << CH_SCALE);
//     }
// }
```

```

        break;
    }

    // scale if gain is NOT 16X
    if (!iGain) chScale = chScale << 4;    // scale 1X to 16X

    // scale the channel values
    channel0 = (ch0 * chScale) >> CH_SCALE;
    channel1 = (ch1 * chScale) >> CH_SCALE;
    //-----

    // find the ratio of the channel values (Channel1/Channel0)
    // protect against divide by zero
    unsigned long ratio1 = 0;
    if (channel0 != 0) ratio1 = (channel1 << (RATIO_SCALE+1)) / channel0;

    // round the ratio value
    unsigned long ratio = (ratio1 + 1) >> 1;

    // is ratio <= eachBreak ?
    unsigned int b, m;
    switch (iType)
    {

    case 0: // T, FN and CL package
        if ((ratio >= 0) && (ratio <= K1T))
            {b=B1T; m=M1T;}
        else if (ratio <= K2T)
            {b=B2T; m=M2T;}
        else if (ratio <= K3T)
            {b=B3T; m=M3T;}
        else if (ratio <= K4T)
            {b=B4T; m=M4T;}
        else if (ratio <= K5T)
            {b=B5T; m=M5T;}
        else if (ratio <= K6T)
            {b=B6T; m=M6T;}
        else if (ratio <= K7T)
            {b=B7T; m=M7T;}
        else if (ratio > K8T)
            {b=B8T; m=M8T;}
        break;

    case 1:// CS package
        if ((ratio >= 0) && (ratio <= K1C))
            {b=B1C; m=M1C;}
        else if (ratio <= K2C)
            {b=B2C; m=M2C;}
        else if (ratio <= K3C)
            {b=B3C; m=M3C;}
        else if (ratio <= K4C)
            {b=B4C; m=M4C;}
        else if (ratio <= K5C)
            {b=B5C; m=M5C;}
        else if (ratio <= K6C)
            {b=B6C; m=M6C;}
        else if (ratio <= K7C)
            {b=B7C; m=M7C;}
    }
}

```

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

```
        else if (ratio > K8C)
            {b=B8C; m=M8C;}
        break;
    }

    unsigned long temp;
    temp = ((channel0 * b) - (channel1 * m));

    // do not allow negative lux value
    if (temp < 0) temp = 0;

    // round lsb (2^(LUX_SCALE-1))
    temp += (1 << (LUX_SCALE-1));

    // strip off fractional portion
    unsigned long lux = temp >> LUX_SCALE;

    return(lux);
}
```

APPLICATION INFORMATION: HARDWARE

Power Supply Decoupling and Application Hardware Circuit

The power supply lines must be decoupled with a 0.1 μF capacitor placed as close to the device package as possible (Figure 18). The bypass capacitor should have low effective series resistance (ESR) and low effective series inductance (ESI), such as the common ceramic types, which provide a low impedance path to ground at high frequencies to handle transient currents caused by internal logic switching.

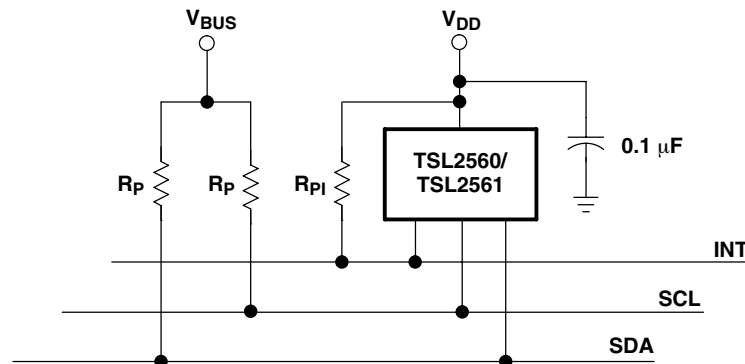


Figure 18. Bus Pull-Up Resistors

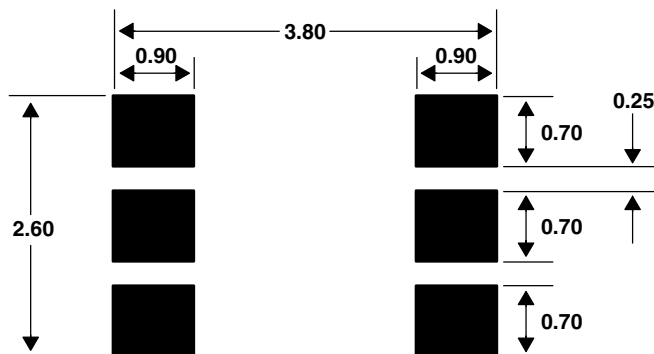
Pull-up resistors (R_p) maintain the SDAH and SCLH lines at a *high* level when the bus is free and ensure the signals are pulled up from a low to a high level within the required rise time. For a complete description of the SMBus maximum and minimum R_p values, please review the SMBus Specification at <http://www.smbus.org/specs>. For a complete description of I²C maximum and minimum R_p values, please review the I²C Specification at <http://www.semiconductors.philips.com>.

A pull-up resistor (R_{PI}) is also required for the interrupt (INT), which functions as a wired-AND signal in a similar fashion to the SCL and SDA lines. A typical impedance value between 10 k Ω and 100 k Ω can be used. Please note that while Figure 18 shows INT being pulled up to V_{DD} , the interrupt can optionally be pulled up to V_{BUS} .

APPLICATION INFORMATION: HARDWARE

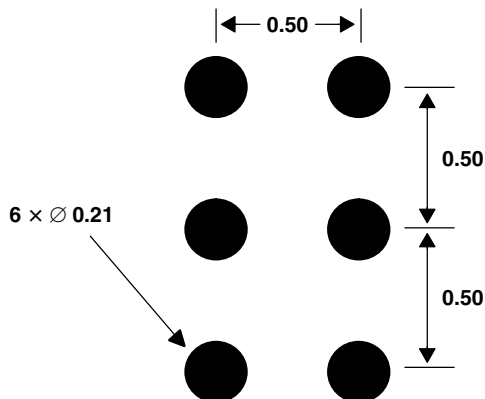
PCB Pad Layout

Suggested PCB pad layout guidelines for the TMB-6 (T) surface mount package, chipscale (CS) package, Dual Flat No-Lead (FN) surface mount package, and ChipLED-6 (CL) surface mount package are shown in Figure 19, Figure 20, Figure 21, and Figure 22.



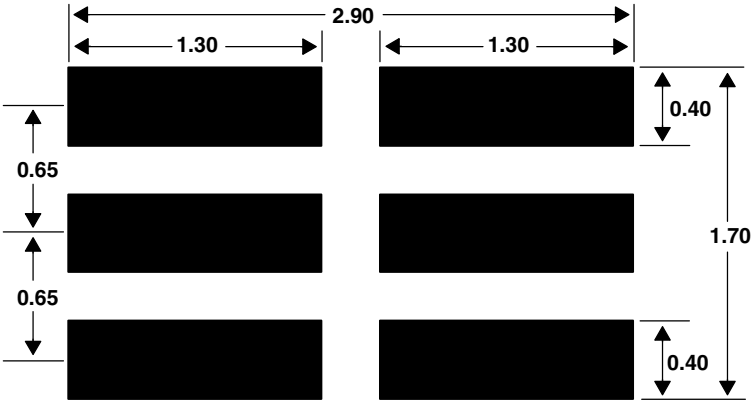
- NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Figure 19. Suggested T Package PCB Layout



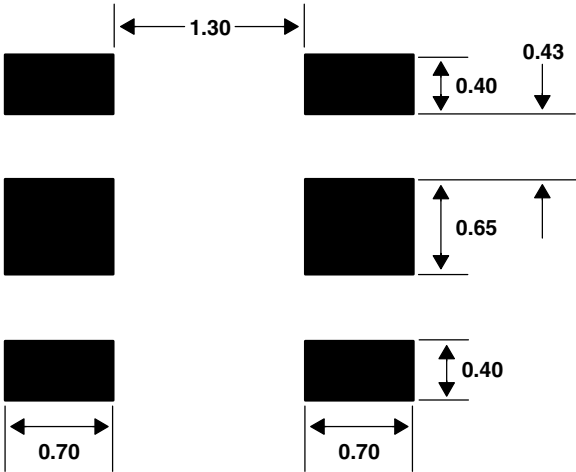
- NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Figure 20. Suggested CS Package PCB Layout



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Figure 21. Suggested FN Package PCB Layout



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

Figure 22. Suggested CL Package PCB Layout

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

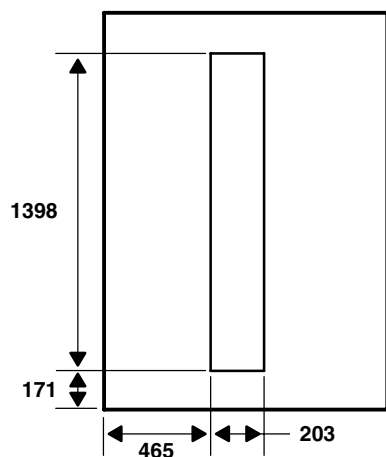
TAOS059N – MARCH 2009

MECHANICAL DATA

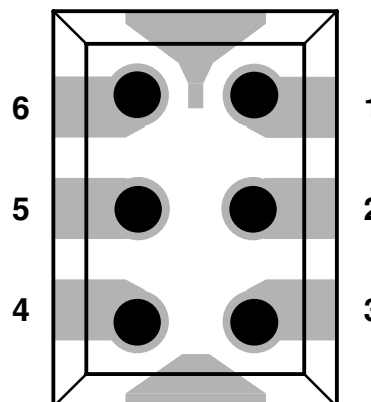
PACKAGE CS

Six-Lead Chipscale Device

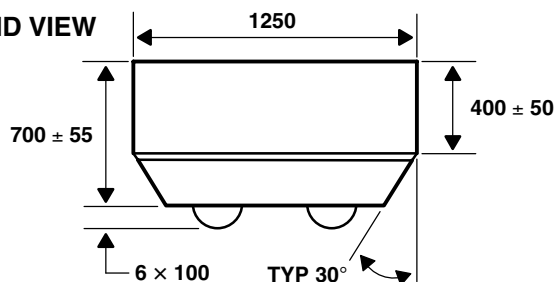
TOP VIEW



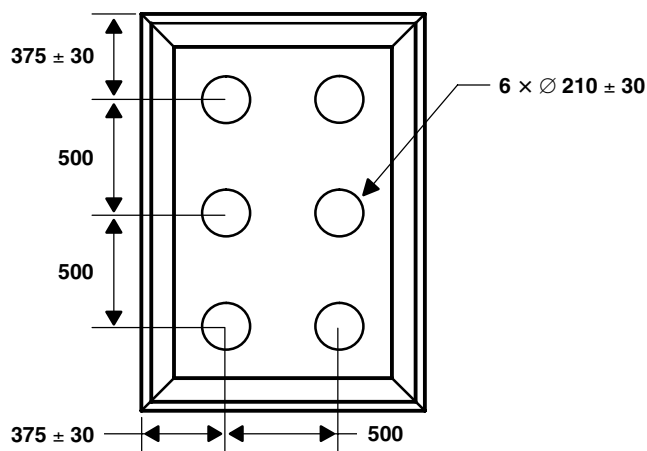
PIN OUT BOTTOM VIEW



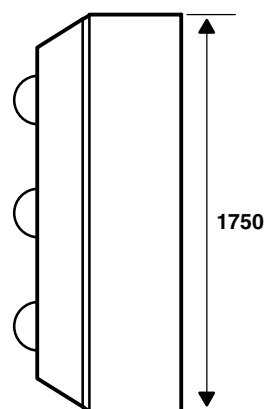
END VIEW



BOTTOM VIEW



SIDE VIEW



Lead Free

- NOTES: A. All linear dimensions are in micrometers. Dimension tolerance is $\pm 25 \mu\text{m}$ unless otherwise noted.
 B. Solder bumps are formed of Sn (96.5%), Ag (3%), and Cu (0.5%).
 C. The top of the photodiode active area is $410 \mu\text{m}$ below the top surface of the package.
 D. The layer above the photodiode is glass and epoxy with an index of refraction of 1.53.
 E. This drawing is subject to change without notice.

Figure 23. Package CS — Six-Lead Chipscale Packaging Configuration

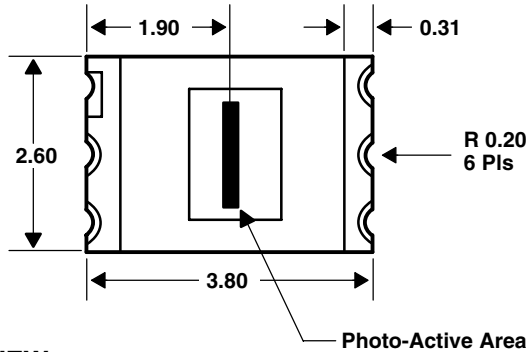


MECHANICAL DATA

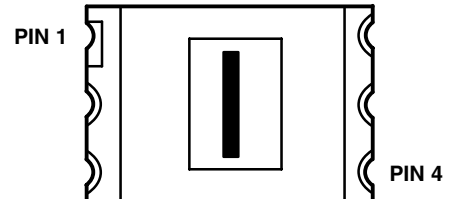
PACKAGE TMB-6

Six-Lead Surface Mount Device

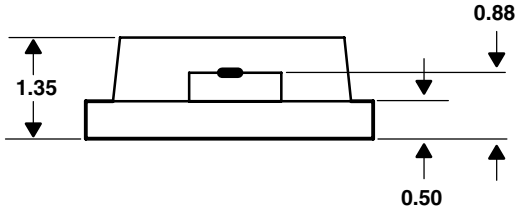
TOP VIEW



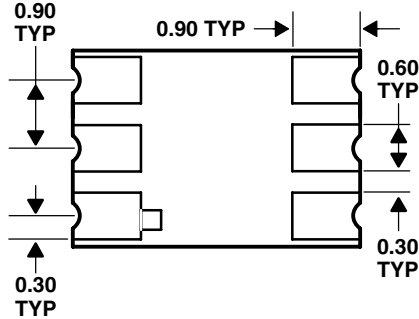
TOP VIEW



END VIEW



BOTTOM VIEW



Lead Free

- NOTES:
- All linear dimensions are in millimeters. Dimension tolerance is ± 0.20 mm unless otherwise noted.
 - The photo-active area is $1398 \mu\text{m}$ by $203 \mu\text{m}$.
 - Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
 - Contact finish is $0.5 \mu\text{m}$ minimum of soft gold plated over a $18 \mu\text{m}$ thick copper foil pattern with a $5 \mu\text{m}$ to $9 \mu\text{m}$ nickel barrier.
 - The underside of the package includes copper traces used to connect the pads during package substrate fabrication. Accordingly, exposed traces and vias should not be placed under the footprint of the TMB package in a PCB layout.
 - This package contains no lead (Pb).
 - This drawing is subject to change without notice.

Figure 24. Package T — Six-Lead TMB Plastic Surface Mount Packaging Configuration

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

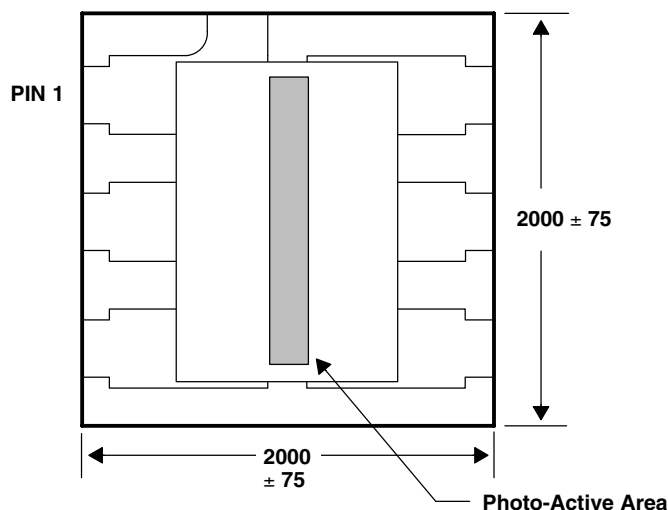
TAOS059N – MARCH 2009

MECHANICAL DATA

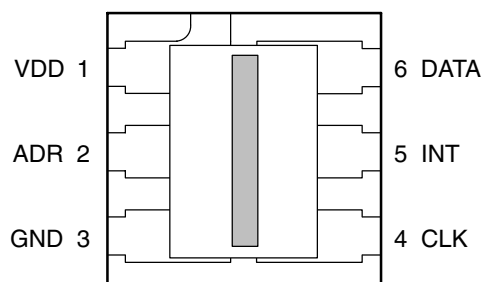
PACKAGE FN

Dual Flat No-Lead

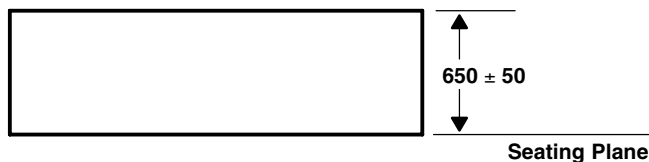
TOP VIEW



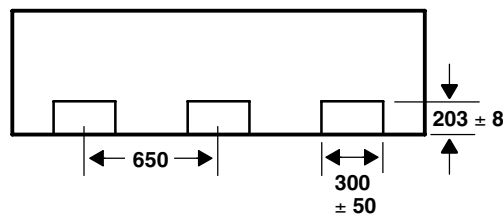
PIN OUT TOP VIEW



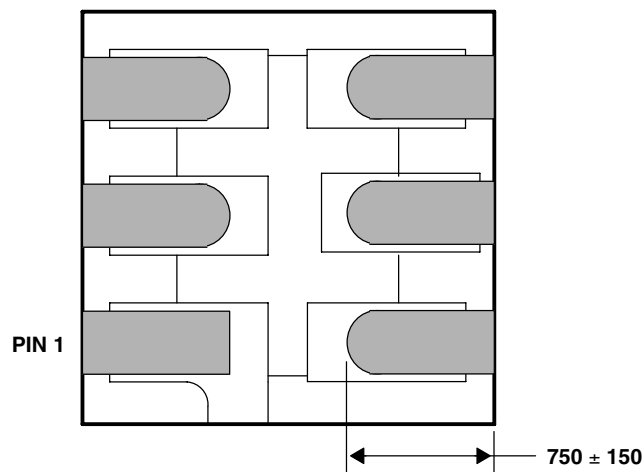
END VIEW



SIDE VIEW



BOTTOM VIEW



Lead Free

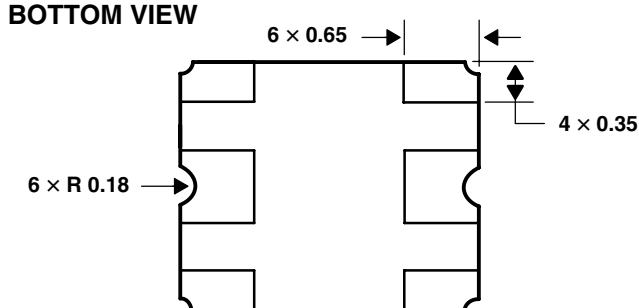
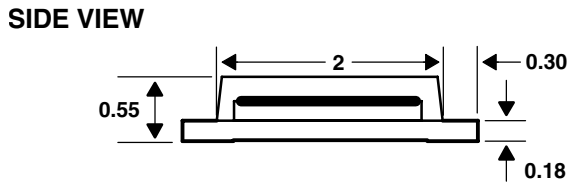
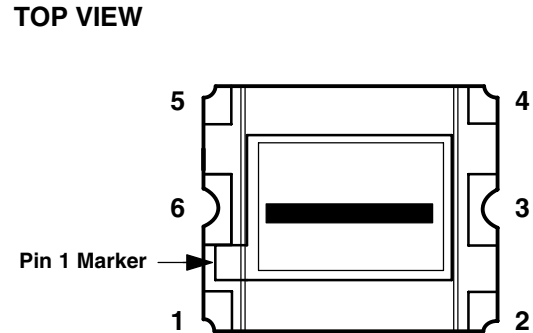
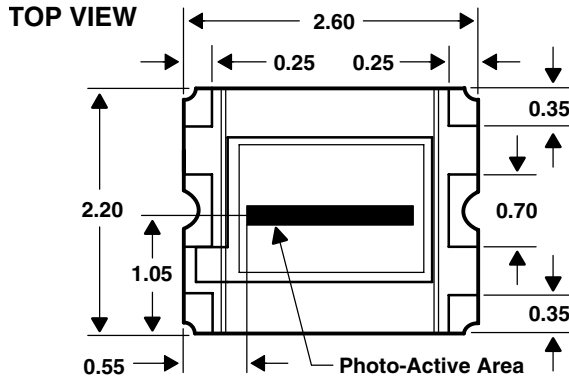
- NOTES: A. All linear dimensions are in micrometers. Dimension tolerance is $\pm 20 \mu\text{m}$ unless otherwise noted.
 B. The photo-active area is $1398 \mu\text{m}$ by $203 \mu\text{m}$.
 C. Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
 D. Contact finish is copper alloy A194 with pre-plated NiPdAu lead finish.
 E. This package contains no lead (Pb).
 F. This drawing is subject to change without notice.

Figure 25. Package FN — Dual Flat No-Lead Packaging Configuration

MECHANICAL DATA

PACKAGE CL-6

Six-Lead Surface Mount Device



Lead Free

- NOTES: A. All linear dimensions are in millimeters. Dimension tolerance is ± 0.20 mm unless otherwise noted.
B. The photo-active area is 1398 μm by 203 μm .
C. Package top surface is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
D. Contact finish is 0.1 μm (minimum) to 1.0 μm (maximum) of soft gold plated over a 15 μm (minimum) to 30 μm (maximum) thick copper foil pattern with a 3 μm (minimum) to 15 μm (maximum) nickel barrier.
E. This package contains no lead (Pb).
F. This drawing is subject to change without notice.

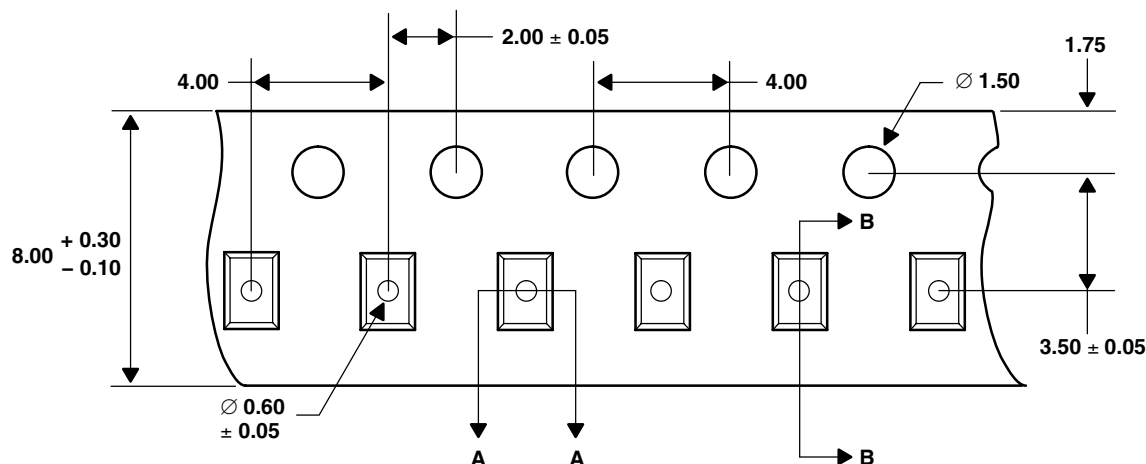
Figure 26. Package CL — Six-Lead ChipLED Plastic Surface Mount Packaging Configuration

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

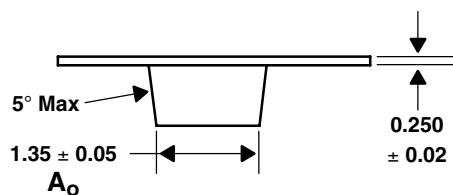
TAOS059N – MARCH 2009

MECHANICAL DATA

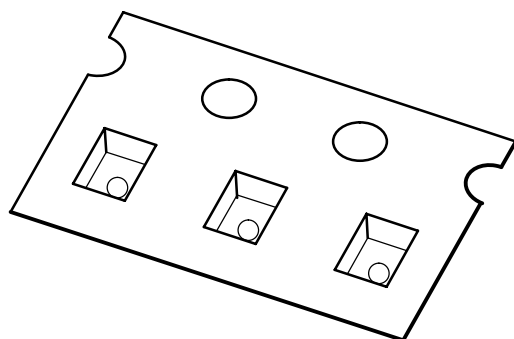
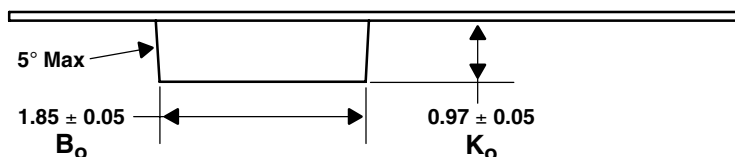
TOP VIEW



DETAIL A



DETAIL B

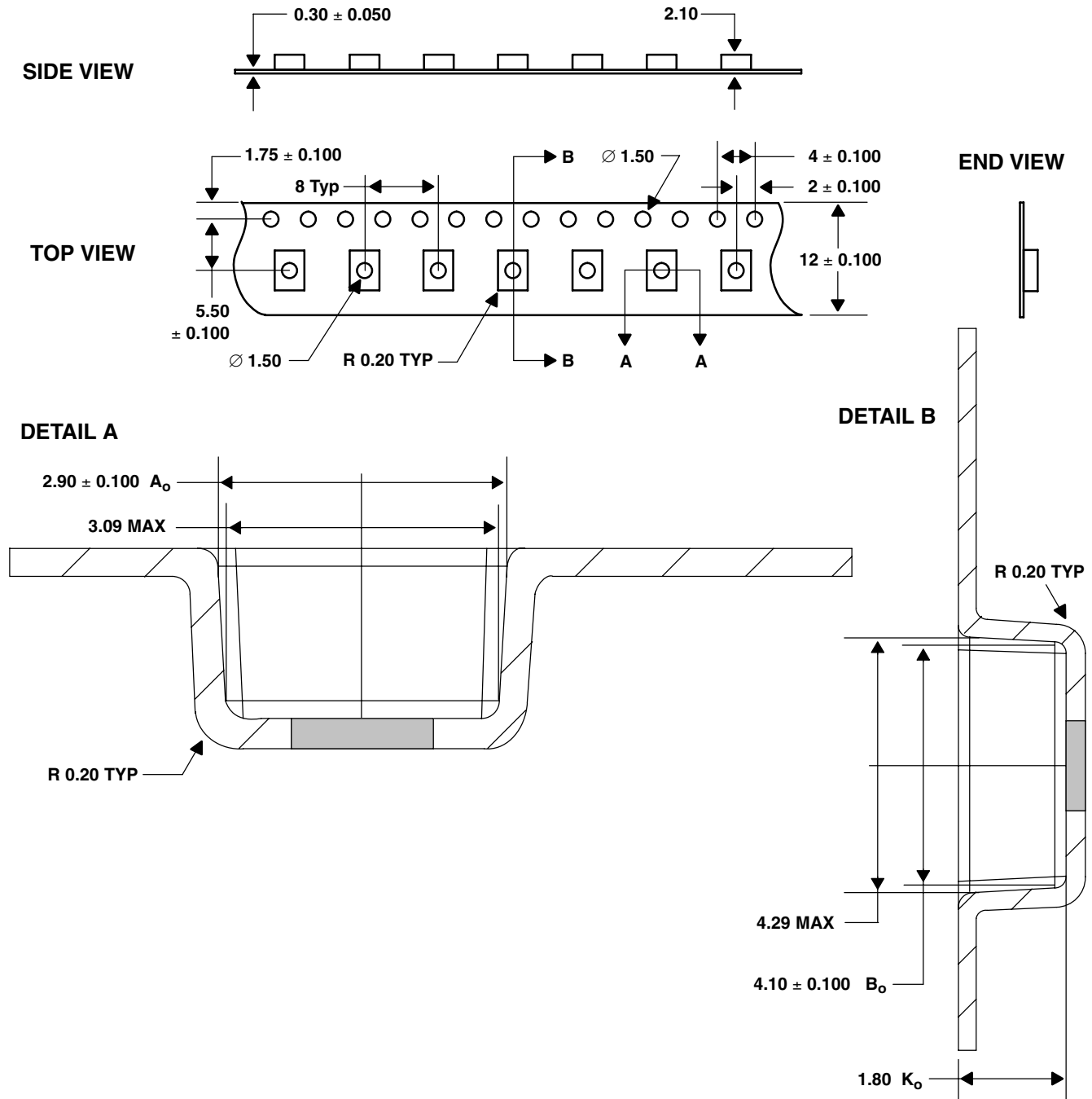


- NOTES:
- A. All linear dimensions are in millimeters. Dimension tolerance is ± 0.10 mm unless otherwise noted.
 - B. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
 - C. Symbols on drawing A_o , B_o , and K_o are defined in ANSI EIA Standard 481-B 2001.
 - D. Each reel is 178 millimeters in diameter and contains 3500 parts.
 - E. TAOS packaging tape and reel conform to the requirements of EIA Standard 481-B.
 - F. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
 - G. This drawing is subject to change without notice.

Figure 27. TSL2560/TSL2561 Chipscale Carrier Tape



MECHANICAL DATA



- NOTES: A. All linear dimensions are in millimeters.
 B. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
 C. Symbols on drawing A_o , B_o , and K_o are defined in ANSI EIA Standard 481-B 2001.
 D. Each reel is 178 millimeters in diameter and contains 1000 parts.
 E. TAOS packaging tape and reel conform to the requirements of EIA Standard 481-B.
 F. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
 G. This drawing is subject to change without notice.

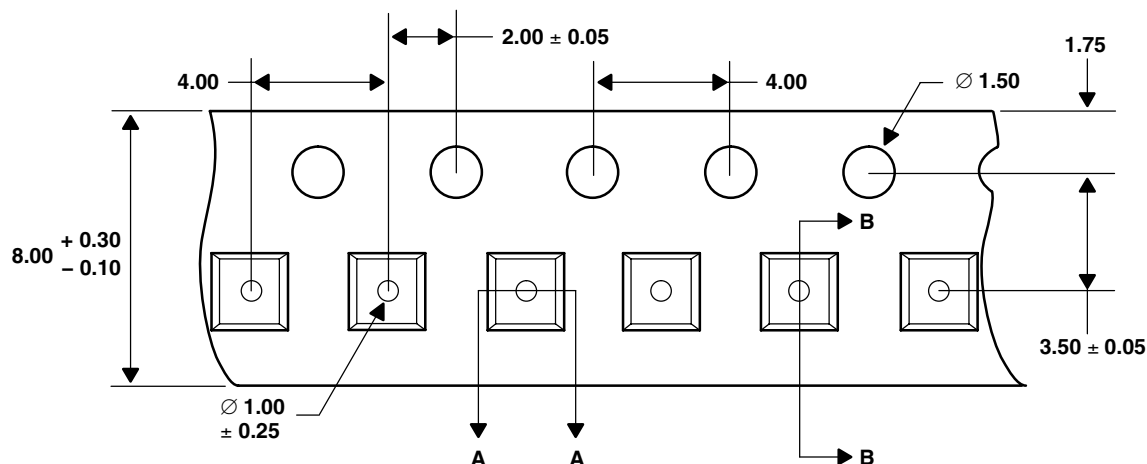
Figure 28. TSL2560/TSL2561 TMB Carrier Tape

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

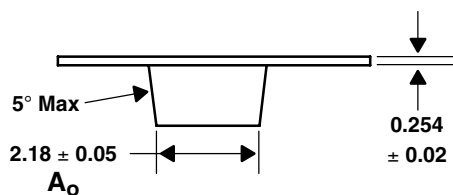
TAOS059N – MARCH 2009

MECHANICAL DATA

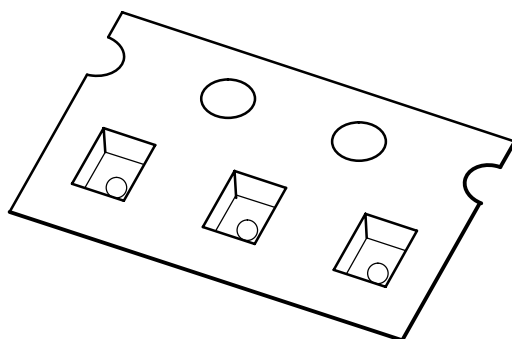
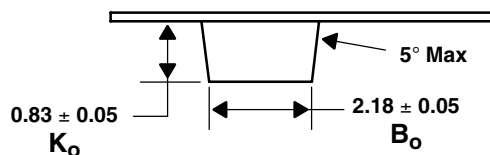
TOP VIEW



DETAIL A



DETAIL B



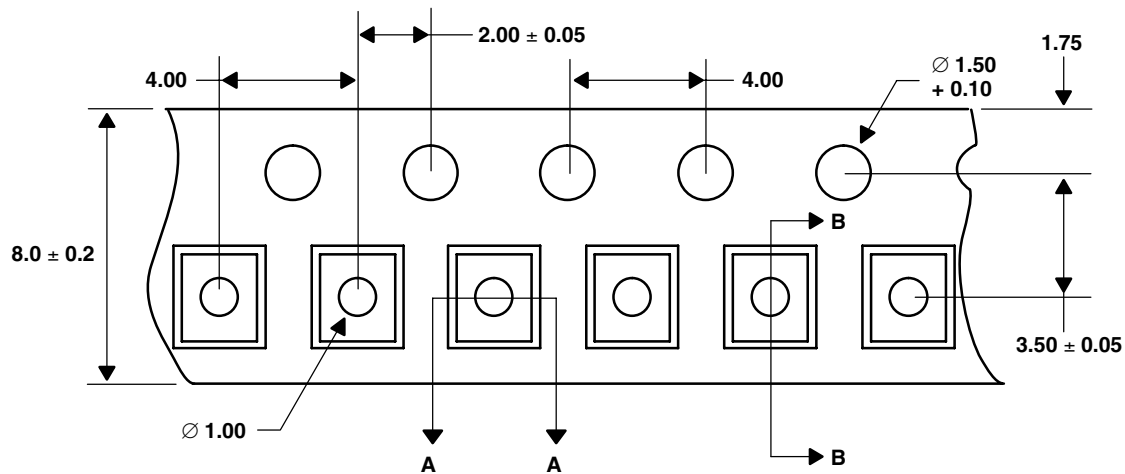
- NOTES:
- H. All linear dimensions are in millimeters. Dimension tolerance is ± 0.10 mm unless otherwise noted.
 - I. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
 - J. Symbols on drawing A_0 , B_0 , and K_0 are defined in ANSI EIA Standard 481-B 2001.
 - K. Each reel is 178 millimeters in diameter and contains 3500 parts.
 - L. TAOS packaging tape and reel conform to the requirements of EIA Standard 481-B.
 - M. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
 - N. This drawing is subject to change without notice.

Figure 29. TSL2560/TSL2561 FN Carrier Tape

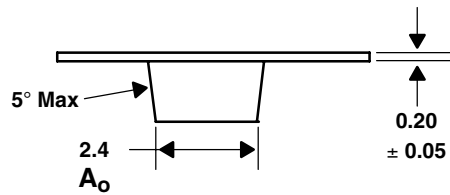


MECHANICAL DATA

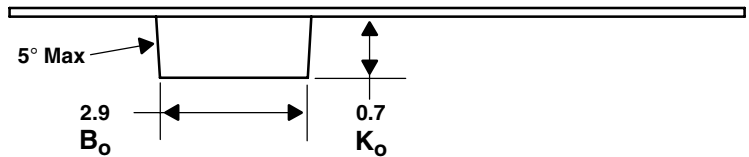
TOP VIEW



DETAIL A



DETAIL B



- NOTES: A. All linear dimensions are in millimeters. Dimension tolerance is ± 0.10 mm unless otherwise noted.
 B. The dimensions on this drawing are for illustrative purposes only. Dimensions of an actual carrier may vary slightly.
 C. Symbols on drawing A_o , B_o , and K_o are defined in ANSI EIA Standard 481-B 2001.
 D. Each reel is 178 millimeters in diameter and contains 2500 parts.
 E. TAOS packaging tape and reel conform to the requirements of EIA Standard 481-B.
 F. In accordance with EIA standard, device pin 1 is located next to the sprocket holes in the tape.
 G. This drawing is subject to change without notice.

Figure 30. TSL2560/TSL2561 CL Carrier Tape

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

MANUFACTURING INFORMATION

The CS, T, FN, and CL packages have been tested and have demonstrated an ability to be reflow soldered to a PCB substrate. The process, equipment, and materials used in these test are detailed below.

The solder reflow profile describes the expected maximum heat exposure of components during the solder reflow process of product on a PCB. Temperature is measured on top of component. The components should be limited to a maximum of three passes through this solder reflow profile.

Table 13. TSL2560/61 Solder Reflow Profile

PARAMETER	REFERENCE	TSL2560/61
Average temperature gradient in preheating		2.5°C/sec
Soak time	t_{soak}	2 to 3 minutes
Time above 217°C	t_1	Max 60 sec
Time above 230°C	t_2	Max 50 sec
Time above $T_{\text{peak}} - 10^\circ\text{C}$	t_3	Max 10 sec
Peak temperature in reflow	T_{peak}	260° C (–0°C/+5°C)
Temperature gradient in cooling		Max –5°C/sec

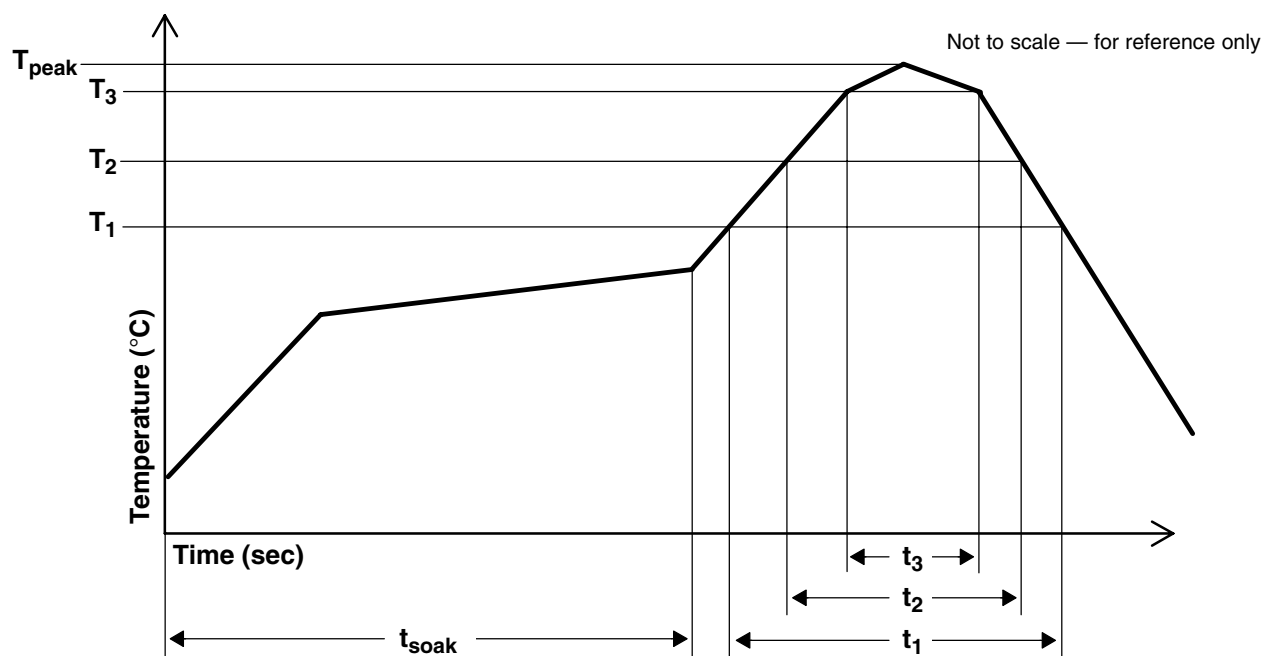


Figure 31. TSL2560/TSL2561 Solder Reflow Profile Graph



MANUFACTURING INFORMATION

Moisture Sensitivity

Optical characteristics of the device can be adversely affected during the soldering process by the release and vaporization of moisture that has been previously absorbed into the package molding compound. To ensure the package molding compound contains the smallest amount of absorbed moisture possible, each device is dry-baked prior to being packed for shipping. Devices are packed in a sealed aluminized envelope with silica gel to protect them from ambient moisture during shipping, handling, and storage before use.

The CS package has been assigned a moisture sensitivity level of MSL 2 and the devices should be stored under the following conditions:

Temperature Range	5°C to 50°C
Relative Humidity	60% maximum
Floor Life	1 year out of bag at ambient < 30°C / 60% RH

Rebaking will be required if the aluminized envelope has been open for more than 1 year. If rebaking is required, it should be done at 90°C for 3 hours.

The T, FN, and CL packages have been assigned a moisture sensitivity level of MSL 3 and the devices should be stored under the following conditions:

Temperature Range	5°C to 50°C
Relative Humidity	60% maximum
Total Time	6 months from the date code on the aluminized envelope — if unopened
Opened Time	168 hours or fewer

Rebaking will be required if the devices have been stored unopened for more than 6 months or if the aluminized envelope has been open for more than 168 hours. If rebaking is required, it should be done at 90°C for 4 hours.

TSL2560, TSL2561 LIGHT-TO-DIGITAL CONVERTER

TAOS059N – MARCH 2009

PRODUCTION DATA — information in this document is current at publication date. Products conform to specifications in accordance with the terms of Texas Advanced Optoelectronic Solutions, Inc. standard warranty. Production processing does not necessarily include testing of all parameters.

LEAD-FREE (Pb-FREE) and GREEN STATEMENT

Pb-Free (RoHS) TAOS' terms *Lead-Free* or *Pb-Free* mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TAOS Pb-Free products are suitable for use in specified lead-free processes.

Green (RoHS & no Sb/Br) TAOS defines *Green* to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material).

Important Information and Disclaimer The information provided in this statement represents TAOS' knowledge and belief as of the date that it is provided. TAOS bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TAOS has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TAOS and TAOS suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

NOTICE

Texas Advanced Optoelectronic Solutions, Inc. (TAOS) reserves the right to make changes to the products contained in this document to improve performance or for any other purpose, or to discontinue them without notice. Customers are advised to contact TAOS to obtain the latest product information before placing orders or designing TAOS products into systems.

TAOS assumes no responsibility for the use of any products or circuits described in this document or customer product design, conveys no license, either expressed or implied, under any patent or other right, and makes no representation that the circuits are free of patent infringement. TAOS further makes no claim as to the suitability of its products for any particular purpose, nor does TAOS assume any liability arising out of the use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

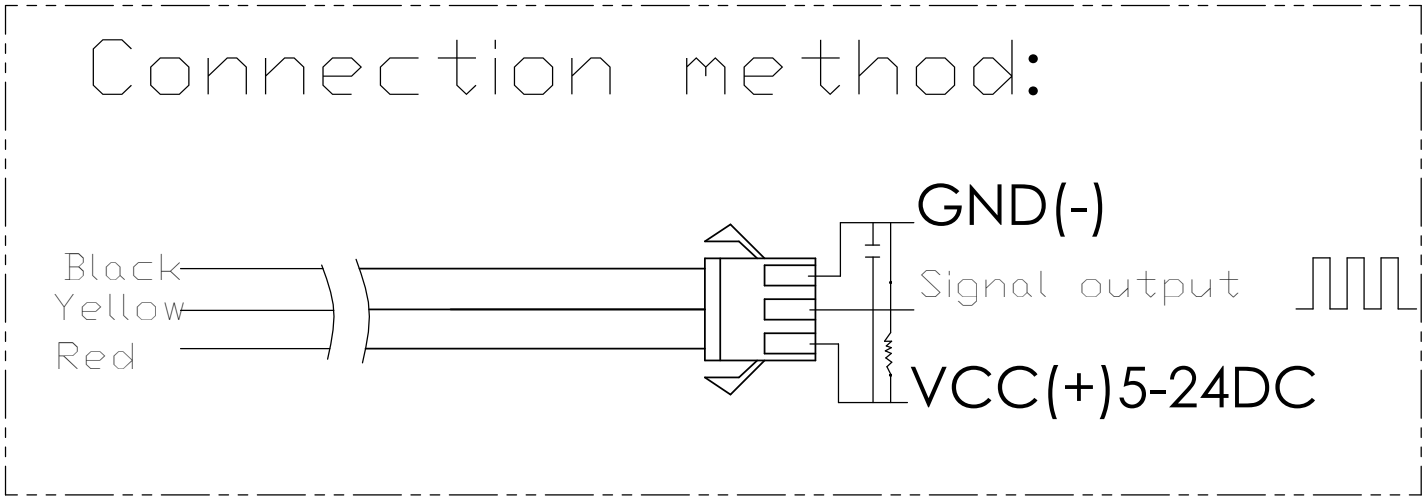
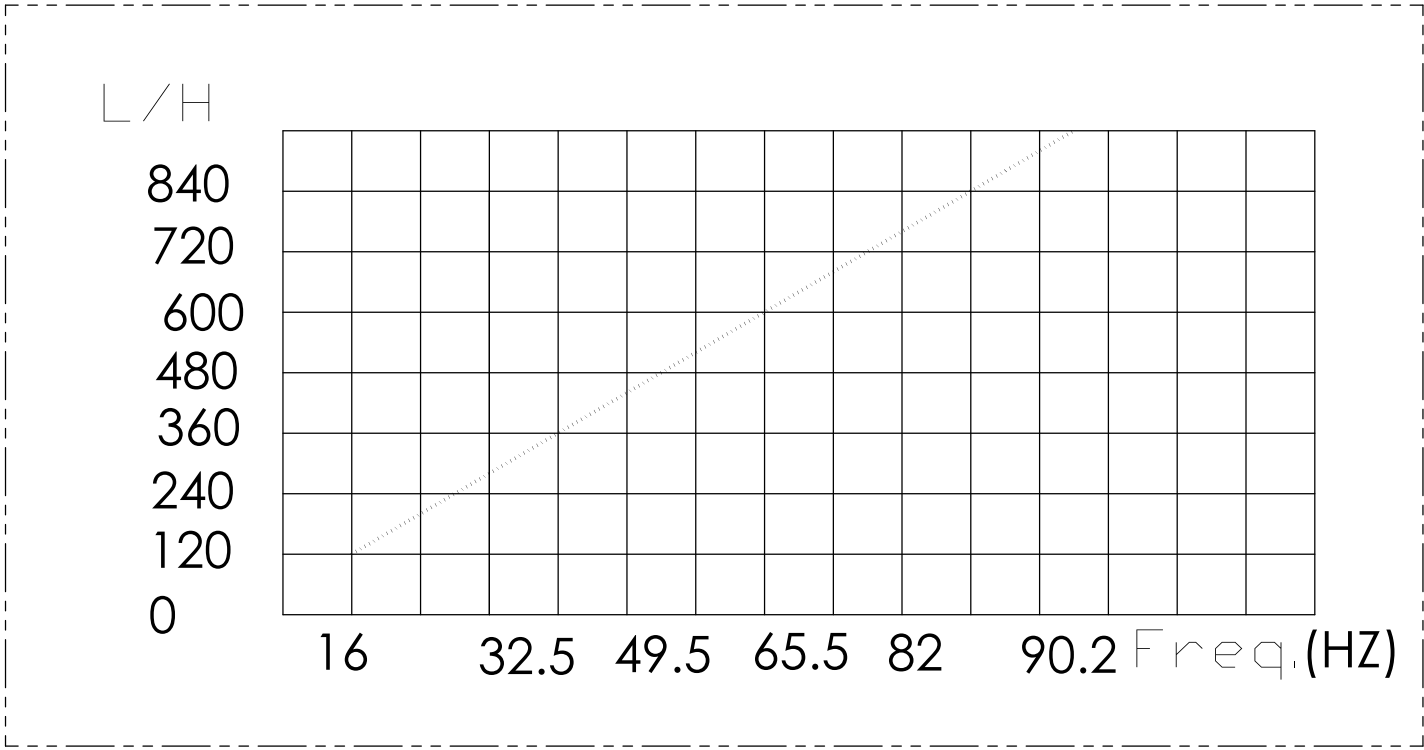
TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS, INC. PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS IN WHICH THE FAILURE OR MALFUNCTION OF THE TAOS PRODUCT MAY RESULT IN PERSONAL INJURY OR DEATH. USE OF TAOS PRODUCTS IN LIFE SUPPORT SYSTEMS IS EXPRESSLY UNAUTHORIZED AND ANY SUCH USE BY A CUSTOMER IS COMPLETELY AT THE CUSTOMER'S RISK.

LUMENOLOGY, TAOS, the TAOS logo, and Texas Advanced Optoelectronic Solutions are registered trademarks of Texas Advanced Optoelectronic Solutions Incorporated.

Flow-Pulse	Flow-Curve	Connection method
------------	------------	-------------------

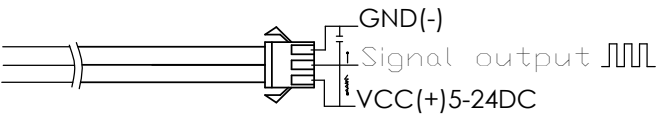
一.Flow Range:100L/H-/1800H-L/H

Flow (L/H)	Frezq.(HZ)	Erro range
120L/H	16	±10
240L/H	32.5	
360L/H	49.3	
480L/H	65.5	
600L/H	82	
720L/H	90.2	



YIFA the plastics Ltd
Prodcut Introduction

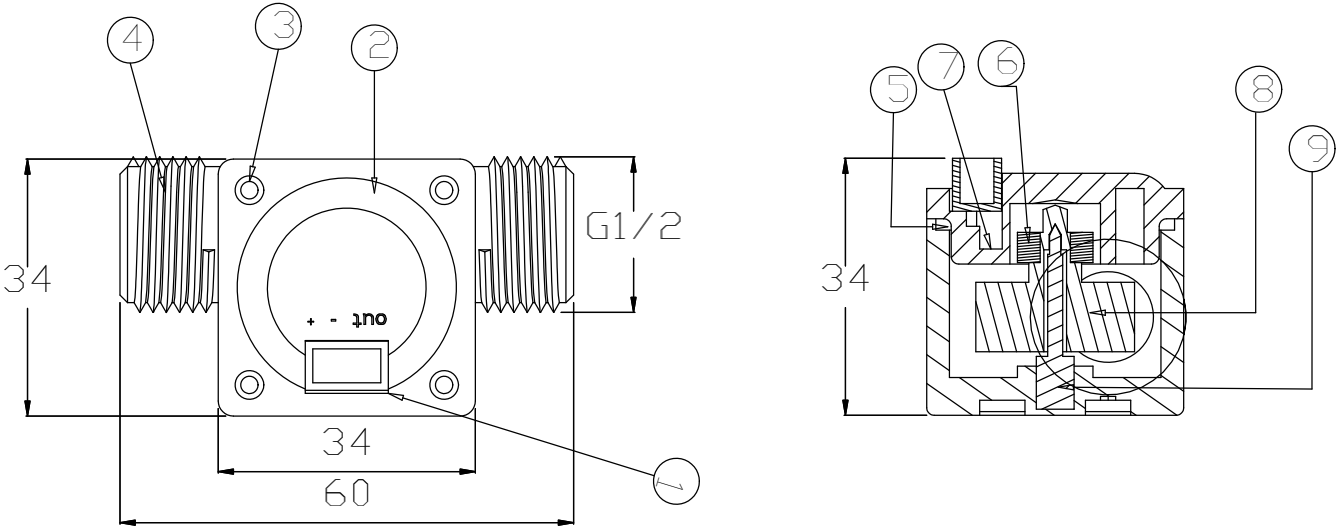
- 1.Modle:YF-21
- 2.Product Name:Hall sensor
- 3.Flow Range: 1-30L/MIN
- 4.(1)Connection Method



- (2)Voltage Range 3.5-24VDC, Pluse Characteristic:F=7Q(L/MIN).
- (3)Extent of error:±5%.
- (4)Flow-Pulse
2L/MIN=16HZ 4L/MIN=32.5HZ 6L/MIN=49.3HZ
8L/MIN=65.5HZ 10L/MIN=82HZ

5.Bom

No.	Item	Material	Qty.
1	Connection wire		1
2	Bonnet	PA	1
3	Screw		4
4	Valve body	PA	1
5	Leak press valve		1
6	Magnet		1
7	Hall		1
8	Impeller	POM	1
9	Rustless steel axis	SUS304	1
10			
11			





KONICA MINOLTA

Compatibles con fuentes de luz de pulso controlado PWM

Nuevo

Iluminancímetro T-10A series

*Satisfacen requisitos JIS Clase AA y DIN Clase B.
Compatibles con la nueva generación de fuentes de luz de pulso controlado PWM.*



Es posible un sencillo y asequible sistema de medición multipunto. El minireceptor disponible permite la medición de iluminancia en espacios pequeños.

Giving Shape to Ideas

Para medición rápida pero precisa de iluminancia. Permiten el desarrollo de sistemas de medición multipunto de iluminancia, de manera sencilla.

Iluminancímetros precisos sin ser complicados, dentro de requisitos JIS Clase AA y DIN Clase B.

Los iluminancímetros T-10A y T-10MA satisfacen los requisitos Clase AA de JIS C 1609-1:2006 “Medidores de iluminancia Parte 1: instrumentos de medición general” y Clase B de DIN 5032 Parte 7 “Fotometría: clasificación de medidores de iluminancia y de luminancia”, proporcionando mediciones de alta precisión y fiabilidad sin complicaciones.

Medición multipunto (2 a 30 puntos) sencilla y asequible.

Se puede medir sencillamente la distribución en iluminancia de un proyector, etc. con un único instrumento y varios receptores.

Compatibles con la nueva generación de fuentes de luz de pulso controlado PWM. Permiten la medición de nuevos dispositivos de iluminación.

En general, los iluminancímetros convencionales no pueden medir con precisión las fuentes de luz de pulso controlado PWM, pero la serie de iluminancímetros T-10A permite medir con precisión incluso dicho tipo de iluminación.

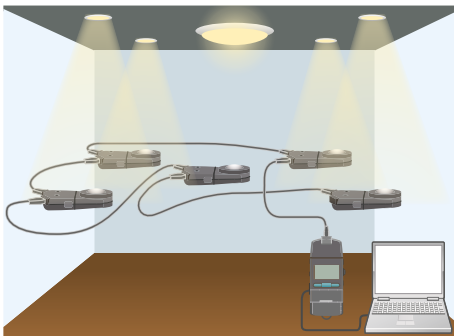
Receptores separables.

Se puede interconectar un instrumento con varios receptores en serie mediante cable LAN, siendo muy sencillo instalar dicho sistema de inspección.

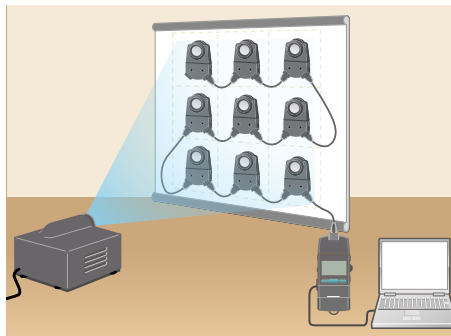


Sistema multipunto de medición de iluminancia

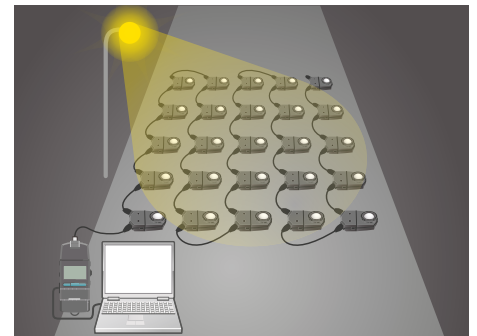
Ejemplo de 5 puntos: arquitectura, etc.



Ejemplo de 9 puntos: proyectores, etc.



Ejemplo de 25 puntos: iluminación variada, etc.



Principales aplicaciones



- Organismos gubernamentales de inspección
- Investigación e inspección por fabricantes de equipos de iluminación
- Mantenimiento en fábricas, oficinas, hospitales, etc.
- Control de iluminación de seguridad, iluminación variada, etc.

- Inspección de iluminación en arquitectura
- Control de iluminación en fabricación de LEDs
- Sensor para sistemas de medición de flujo luminoso total o de caracterización de distribución luminosa, etc.



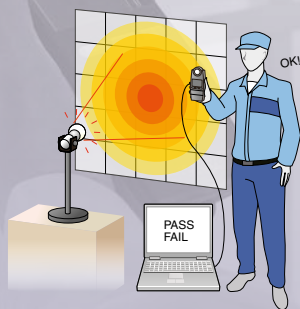
Programa de gestión de datos T-S10w (accesorio opcional)

**Práctico y sencillo programa
complemento en Excel®**

Captura los datos de medición de la serie de iluminancímetros T-10A directamente en Excel®. El tratamiento adicional de datos se puede efectuar libremente mediante las múltiples funciones de Excel®.

Botón de transferencia de datos desde el instrumento

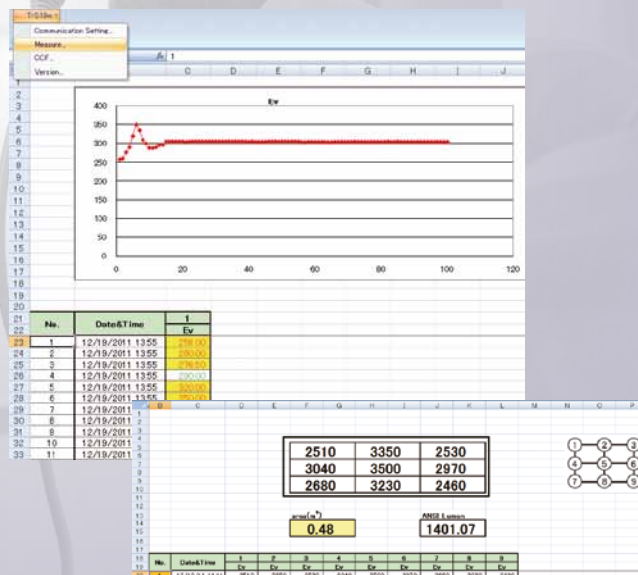
El programa T-S10w controla el instrumento por completo desde el ordenador, pero también es posible utilizar el botón de transferencia en el T-10A para enviar a Excel® una medición concreta.



Posibilidad de medición multipunto y de calibración CCF

Es posible gestionar hasta 30 receptores de medición.

Se dispone de la función CCF (Factor de Corrección de Color) para ajustar cada sensor con una misma referencia.



Principales especificaciones del programa de gestión de datos T-S10w

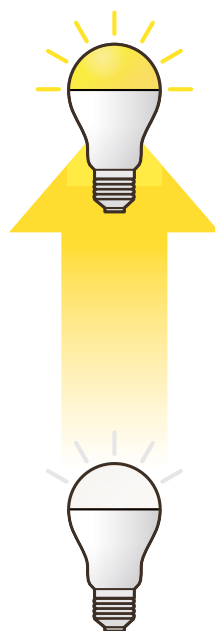
Tipo	Complemento para Excel® (se requiere Excel® en el computador)
Sistema Operativo	Uno de los siguientes entornos con Excel® instalado: *Lenguajes en paréntesis indican el lenguaje del SO. Windows® XP + Excel® 2003 (inglés, japonés o chino simplificado) Windows® 7 + Excel® 2010 (inglés, japonés o chino simplificado) Para requisitos de sistema para las versiones de Windows® y Excel® consultar sus respectivas especificaciones
Instrumentos compatibles	T-10A, T-10MA, T-10WSA, T-10WA, T-10, T-10M, T-10WS, T-10W

Acerca de la iluminación de pulso controlado PWM

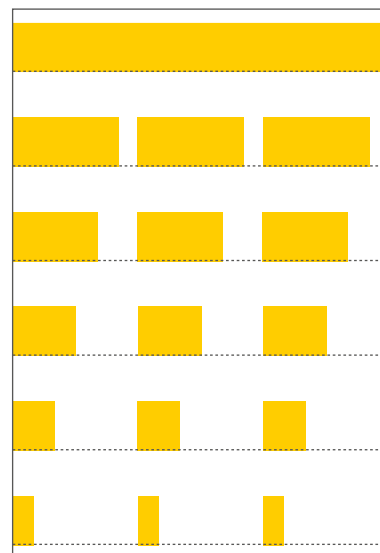
PWM es la abreviatura de Modulación por Ancho de Pulsos, y se refiere al método de controlar la intensidad de una señal modulando la proporción entre el período de apagado y encendido de dicha señal pulsada.

Una señal pulsada es una señal que oscila alternativamente entre apagada y encendida, y el porcentaje de período de encendido durante un ciclo se denomina “ciclo de trabajo”.

Un dispositivo de pulso controlado PWM permite controlar la intensidad de iluminación mediante el control del ciclo de trabajo (tiempo de encendido) de los pulsos de emisión de una lámpara. A mayor duración de cada pulso de luz la iluminación es más intensa, y al contrario, a menor duración de los pulsos de luz la iluminación es menos intensa.



Período de emisión de luz



<Receptor estándar>

T-10A



Ventana
difusora
receptor:
Ø 25 mm

<Mini receptor>

T-10MA/T-10WsA/T-10WL



Ventana
difusora
receptor:
Ø 14 mm

T-10A

T-10MA (cable de 1 m)

T-10WsA (cable de 5 m)

T-10WL (cable de 10 m)

Satisface JIS Clase AA y
DIN class B

Satisface JIS Clase AA y
DIN class B

Satisfacen requisitos JIS para
medidores especiales de iluminancia

Permite la medición de
iluminancia en áreas pequeñas.

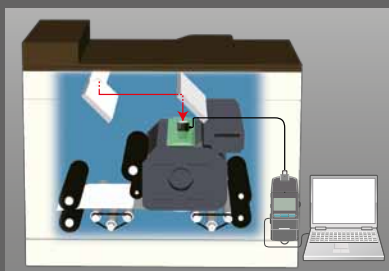
Sumergible

Pedido Especial

Se puede utilizar para mediciones
generales de iluminancia.

Se puede utilizar para mediciones de
iluminancia en pequeños espacios
donde no cabe el receptor
estándar. Se puede instalar fácilmente
en posibles tipos de sistemas que
requieran controlar el nivel de
iluminación existente.

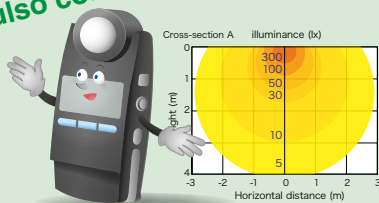
El mini receptor y su cable son
sumergibles, pudiendo utilizarse para
mediciones dentro del agua. Se puede
utilizar para control de iluminación en
actividades piscícolas (piscifactorías,
etc.) o para medición exterior de
iluminancia en días con lluvia.



Trío para medición de iluminancia de Konica Minolta Sensing

La línea de instrumentos Konica Minolta Sensing para medición de iluminancia incluye no sólo el iluminancímetro T-10A que permite medir fuentes de luz de pulso controlado (PWM), sino también el colorímetro CL-200A que permite medir temperatura de color y el espectroiluminancímetro CL-500A que permite medir el índice de rendimiento de color.

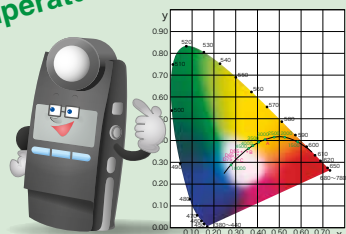
**Iluminancímetro
que puede medir iluminación
de pulso controlado PWM**



Iluminancímetro T-10A

Satisface las normativas DIN Clase B y JIS Clase AA. Iluminancímetro capaz de medir con precisión las nuevas generaciones de lámparas, incluso de iluminación mediante pulso controlado PWM. Posibilidad de múltiples receptores para medición multipunto sencilla y asequible. Disponibilidad de modelo minireceptor para medición de iluminancia en espacios reducidos.

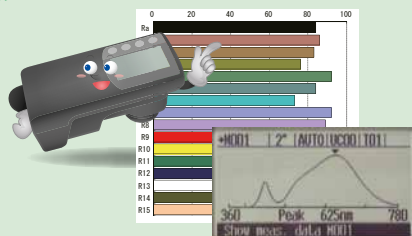
**Medición
de temperatura de color**



Colorímetro CL-200A

Referencia "de facto" industrial para la medición de temperatura de color. Proporciona valores de iluminancia (JIS AA Class) y cromaticidad. Compacto y ligero, con receptor separable y cable de extensión múltiple. Incluye el práctico y sencillo programa para PC como accesorio estándar.

**Medición del índice
de rendimiento de color**



Espectroiluminancímetro CL-500A

El primer espectroiluminancímetro que satisface las normativas JIS Clase AA y DIN Clase B. Compacto y ligero, se puede instalar sencillamente en sistemas de inspección y es ideal para la evaluación de las propiedades de rendimiento de color. Incluye el práctico y sencillo programa para PC como accesorio estándar.

Espectroradiómetro CS-2000A adaptado para iluminancia

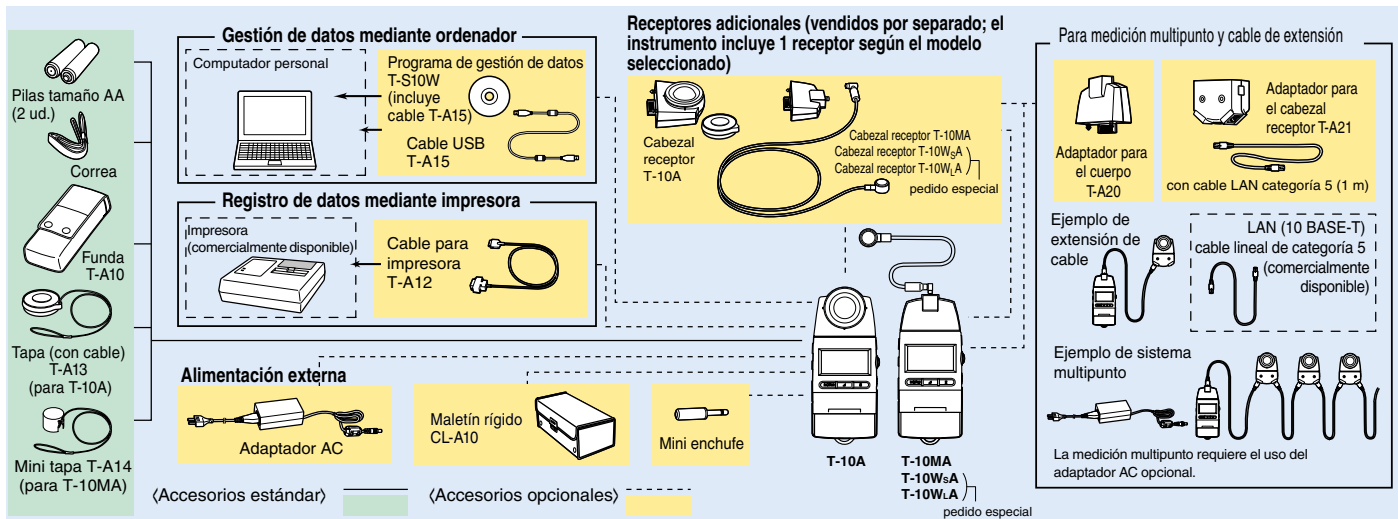
La medición de la irradiancia espectral es posible mediante el empleo del adaptador de iluminancia. Conveniente para la evaluación de iluminancia de proyectores y dispositivos LED o EL. Un único instrumento para ambas mediciones de radiancia espectral e irradiancia espectral.

Nuestro tope de gama CS-2000 es conocido para la medición de todo tipo de pantallas de alta definición, y en 2008 ha merecido el Gran Premio de la 13ª edición del "Advanced Display of the Year" en la categoría de equipos de evaluación de pantallas.

Ancho de banda espectral: 5 nm o menos (medio ancho de banda) Rango de medición de iluminancia:
- Apertura de medición 1° : 0,01 a 75.000 lx
- Apertura de medición 0.1° : 1,00 a 7.500.000 lx



Diagrama de sistema



Principales especificaciones del T-10A

Modelo		Iluminancímetro T-10A (cabezal receptor estándar)	Iluminancímetro T-10MA (cabezal receptor mini)	Iluminancímetro T-10WsA (cabezal receptor sumergible mini)	Iluminancímetro T-10WLA (cabezal receptor sumergible mini)
Tipo		Iluminancímetro digital multifunción con cabezal receptor separable (posible medición multipunto de 2 a 30 puntos)			
Clase de iluminancímetro		Satisface requisitos de Clase AA de JIS C 1609-1:2006 "Medidores de iluminancia Parte 1: instrumentos de medición general." Satisface requisitos de Clase B de DIN 5032 Parte 7		Satisface requisitos de JIS C 1609-1:2006 para iluminancímetros especiales **	
Receptor		Fotocélulas de silicio			
Respuesta espectral relativa		Dentro del 6% (f1') de la eficiencia luminosa espectral de CIE V (λ)			
Respuesta de coseno (f2)		Dentro del 3%		Dentro del 10%	
Rango de medición		Rango automático (5 rangos manuales en caso de salida analógica)			
Funciones de medición		Iluminancia (lx), diferencia de iluminancia (lx), cociente de iluminancia (%), iluminancia integrada (lx·h), tiempo de integración (h), iluminancia promedio (lx)			
Rango medida	Iluminancia	0,01 a 299.900 lx; 0,001 a 29.990 fcd		1,00 a 299.900 lx; 0,1 a 29.990 fcd **2	
	Iluminancia integrada	0.01 to 999,900 x 10 ³ lx·h 0.001 to 99,990 x 10 ³ fcd·h / 0.001 to 9999 h			
Función de calibración de usuario		Función de ajuste CCF (factor de corrección de color); valor de medición multiplicado entre 0.5 y 2.0			
Linealidad		±2% ±1 dígito del valor mostrado			
Dispersión por temperatura/ humedad		Dentro del ±3%			
Salida digital		USB			
Salida analógica		1 mV/dígito, 3 V valor máximo; impedancia salida: 10 KΩ; tiempo de respuesta al 90%: 28 ms			
Alimentación		LCD con 3 ó 4 dígitos significativos e iluminación de fondo (automática)			
Power source		2 pilas tamaño AA / adaptador AC-A308 (opcional; de 1 a 10 receptores) adaptador AC-A311 (opcional; de 1 a 30 receptores)			
Duración pilas		72 horas o más (usando pilas alcalinas) en medición continua			
Rango de temperatura/ humedad de manejo		- 10° a 40° C, humedad relativa 85% o menor (a 35° C) sin condensación		5° a 40° C, humedad relativa 85% o menor (a 35° C) sin condensación	
Rango de temperatura/ humedad de almacenamiento		- 20° a 55° C, humedad relativa 85% o menor (a 35° C) sin condensación		0° a 55° C, humedad relativa 85% o menor (a 35° C) sin condensación	
Dimensiones		69 x 174 x 35 mm	Cuerpo: 69 x 161.5 x 30 mm Receptor: Ø16.5 x 13.8 mm		
Longitud Cable		—	1 m	5 m	10 m
Peso (sin pilas)		200 g (7.0 oz.)	205 g	260 g (sólo receptor: 120 g)	340 g (sólo receptor: 200 g)

*1 Satisface requisitos de Clase AA de JIS C 1609-1:2006 en todos los parámetros excepto la respuesta de coseno (f2').

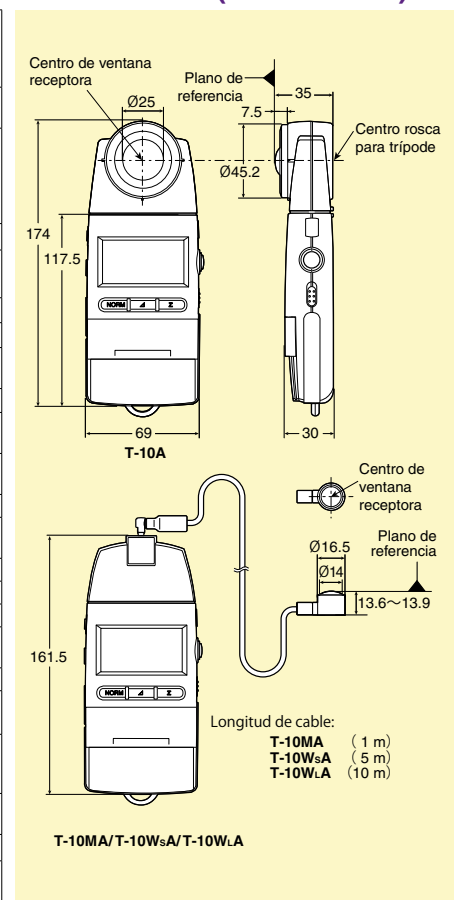
*2 Es posible la medición por debajo de 1.00 lx, pero puede ser poco seguro debido a efectos de ruido electrónico.

*Notas respecto receptores mini y mini sumergible>

*No tocar el cable durante la medición. Ello puede proporcionar resultados de medición poco fiables.

*Garantizar la estabilidad del cable durante la medición. En caso contrario son posibles resultados de medición poco fiables.

Dimensiones (Unidad:mm)



PRECAUCIONES DE SEGURIDAD

Para un uso correcto y para su seguridad, asegúrese de leer el manual de instrucciones antes de utilizar el instrumento.

- Conecte siempre el instrumento al suministro de energía del voltaje especificado. Una conexión inadecuada puede producir fuego o una descarga eléctrica.
- Utilice siempre las pilas especificadas. Unas pilas inadecuadas pueden producir fuego o una descarga eléctrica.

- KONICA MINOLTA, su logo, su símbolo de marca Konica Minolta y "Giving Shape to Ideas" son marcas registradas de KONICA MINOLTA HOLDINGS, INC.
- Windows® y Excel® son marcas de Microsoft Corporation en USA y otros países.
- Las especificaciones e ilustraciones mostradas están sujetas a cambio sin previo aviso
- Los ejemplos de pantalla sólo tienen propósito ilustrativo.



Certificate No. WA 0937 154
Registration Date:
March 3, 1995



Certificate No. JQA-E-80027
Registration Date:
March 12, 1997

AQ - Distribuidor España y Portugal

Konica Minolta Sensing Europe B.V.

Spain Office

European Headquarter/BENELUX
German Office
French Office
UK Office
Italian Office
Belgian Office
Swiss Office
Nordic Office
Polish Office

Valencia, Spain

Nieuwegein, Netherland
München, Germany
Roissy, France
Warrington, United Kingdom
Milan, Italy
Zaventem, Belgium
Dietikon, Switzerland
Västra Frölunda, Sweden
Wrocław, Poland

Phone: +34 96 330 20 03

Phone: +31 (0)30 248-1193
Phone: +49 (0)89 4357 156 0
Phone: +33 (0)1 80-111070
Phone: +44 (0) 1925-467300
Phone: +39 02849488.00
Phone: +32 (0)2 7170 933
Phone: +41 (0)43 322-9800
Phone: +46 (0)31 7099464
Phone: +48 (0)71 33050-01

aquekcnica@aquekcnica.com

info.sensing@seu.konicaminolta.eu
info.germany@seu.konicaminolta.eu
info.france@seu.konicaminolta.eu
info.uk@seu.konicaminolta.eu
info.italy@seu.konicaminolta.eu
info.benelux@seu.konicaminolta.eu
info.switzerland@seu.konicaminolta.eu
info.nordic@seu.konicaminolta.eu
info.poland@seu.konicaminolta.eu





TECHNICAL DATA SHEET

PETG

DESCRIPTION

Polyethylene Terephthalate Glycol (PETG) is one of the world's most common thermoplastic polymers. Extruder PETG was developed for a wide range of applications where the main requirement is a good balance between mechanical and optical material properties. The raw material is approved according to the FDA-, REACH- and RoHS-Standards. PETG is flame retardant according to UL 94 at a wall thickness of 3.2 mm.

FEATURES

- Good mechanical properties
- High chemical resistance
- Low warping tendency
- Low shrinking
- Recycleable

PROPERTIES ¹

TEST	METHOD	UNIT	VALUE
Flexural modulus (E-Modulus)	ISO 178	MPa	2100
Flexural strength	ISO 178	MPa	68
Tensile modulus (E-Modulus)	ISO 527	MPa	3100
Yield stress	ISO 527	MPa	51
Elongation at yield	ISO 527-2	%	4
Strength	ISO 527	MPa	61
Elongation at strength	ISO 527	%	4
Nominal elongation at break	ISO 527-2	%	28
Notched impact strength	ISO 180	kJ/m ²	4,7
Unnotched impact strength	ISO 180	kJ/m ²	no break
VICAT A (VST)	ISO 306	°C	78*
Melting temperature	ISO 3146-C	°C	180-200
MFR	ISO 1133	g/10min	6
Shrinking	ISO 294-4	%	0,5
Density	ASTM D792	g/cm ³	1,29
Flammability	UL 94	V-2	-

*Temperature resistance tested at a minimum wall thickness of 4 mm.

CERTIFICATIONS & ADDITIONAL INFORMATION ²



FREE OF
SILICONE



RECYCLABLE

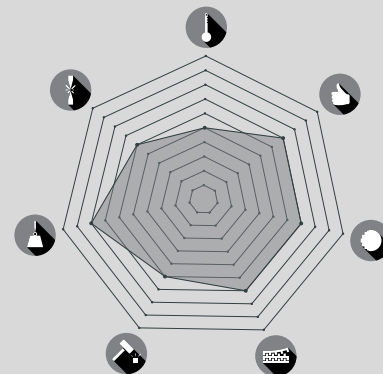


FLAMERETARDANT
UL 94 V-2

STORAGE AND SHELF LIFE

Store in a dry room at room temperature (18-27°C / 65-80°F). Keep out of direct heat and sunlight. When stored correctly, this material has a shelf life of 2 years.

1. Additional info in our regulatory, additional information and chemical resistance data sheets.
2. Certifications depend on colors in final product. More info in the additional information sheet.



TEMPERATURE
RESISTANCE 5

EASE OF
PRINTING 7

VISUAL
QUALITY 7

LAYER
ADHESION 7

IMPACT
RESISTANCE 6

MAXIMUM
STRESS 8

ELONGATION
AT BREAK 6

PRINT SETTINGS

Nozzle	210-230°C
Heatbed	60-70°C
Adhesive	not required
Speed	40-60mm/s
Cooling	20-50%

Recommended settings for printers with a 0.4mm Nozzle. Max. 50% layerheight. Optimal print settings may vary between different printers and also depend on environmental factors.

NEED HELP?

If you have any question about the product and/or you are experiencing an issue, please contact us via support@extruder.com



**BST****A RELIABLE TESTING FOR TRUST**

GLOBAL TESTING AND CERTIFICATION PRECISION SERVICE CLOUD FACTORY

Certificate of Compliance

Certificate NO. : BSTXD190412500501SC**Applicant : WENZHOU TIANPIN ELECTRIC TECHNOLOGY CO., LTD**
: Houjie Industrial Zone, Liushi Town, Yueqing City, Zhejiang Province**Manufacturer : WENZHOU TIANPIN ELECTRIC TECHNOLOGY CO., LTD**
: Houjie Industrial Zone, Liushi Town, Yueqing City, Zhejiang Province**Product Name : ELECTRIC SEALING BOX****Main Test Model : TP-AG****Additional Model : TP-AT, TP-OS, TP-AD, TP-AH, TP-PG, TP-F, HK, HA, HT****Test Standard : EN 60529:1991+A1:2000+A2:2013****As shown in the Test Report No. : BSTXD190412500501SR**

The EUT described above has been tested by us with the listed standards.
The certificate applies to the tested sample above mentioned only and shall not imply an assessment of the whole production.



IP67

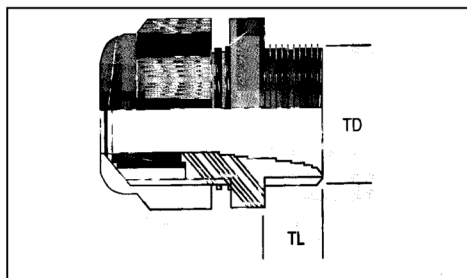
Christina Deng
Manager
Apr. 26, 2019

BST Technology(Shenzhen)Co., Ltd

Add: No. 7, New Era Industrial Zone, Guantian, Bao'an District, Shenzhen, Guangdong, China

Certificate Search: <http://www.bst-lab.com>, Tel: 400-882-9628, 8009990305, E-mail: christina@bst-lab.com

GAESTOPAS	FICHA DE CARACTERISTICAS DE PRODUCTO	ES-CO-01	ED.2ª(07/04)
		JULIO 2006	HOJA. 1 de 1



MODELO: PRENSAS."GADI" (363. DE POLIAMIDA)



MATERIAL	PA6 (Poliamida 6) Libre de Halógenos
JUNTA	Neopreno
NORMAS DE FABRICACION	SEGÚN DIN EN 50 262 (VDE 0619)
CERTIFICACIONES	VDE (Nº 40004337)

CARACTERISTICAS FISICO – QUIMICAS	
GRADO DE PROTECCION	IP 68, 5 bar
TEMPERATURA DE TRABAJO	-30° C hasta +80° C permanentemente
TEMPERATURA MÁXIMA	hasta+120° C intermitentemente
COLOR	Gris (RAL 7035)(RAL 7001) Negro (RAL 9005)

DIMENSIONALES			
Métrica	Para Cables ϕ mm	Ancho de la Llave mm	Longitud de la Rosca mm
M12x1,5	3-6,5	15	8
M16x1,5	5-10	22	10
M20x1,5	10-14	27	10
M25x1,5	13-18	33	10
M32x1,5	18-25	42	18
M40x1,5	22-32	53	18
M50x1,5	30-38	60	18
M63x1,5	34-44	65	18
Las pruebas han sido efectuadas y documentadas de acuerdo a la Norma EN 50262			

Nota: GAESTOPAS certifica que los datos, aquí expuestos, son una fiel reproducción de los datos facilitados por el fabricante.



MATERIAL SAFETY DATA SHEET

PETG

1. PRODUCT IDENTIFICATION

TRADE NAME Extrudr PETG

MANUFACTURER FD3D GmbH
Klosterstrasse 13
6923 Lauterach
AUSTRIA

info@extrudr.com

USE OF PRODUCT PETG Copolyester, suitable for 3D printing filament

2. HAZARD IDENTIFICATION

CLASSIFICATION No need for classification according to GHS criteria for this product (according to Regulation (EC) No. 1272/2008 [CPL]).

LABEL ELEMENTS According to Regulation (EC) No. 1272/2008 (CLP) this product does not require a hazard warning label in accordance with GHS criteria.

SPECIAL ADVICE ON HAZARDS Danger of burns in contact with hot polymer. Hazardous vapours in case of burning.

3. COMPOSITION

CHEMICAL CHARACTERISTICS The polymer contains minor additives such as stabilizers and catalysts. These additives are immobilized by the polymer and are not released with normal use.

4. FIRST-AID MEASURES

ON SKIN CONTACT Cool skin rapidly with cold water after contact with molten polymer. Do not peel polymer from the skin. Obtain medical attention.

AFTER INHALATION Move exposed person to fresh air in case of accidental inhalation of dust or fumes from overheating or combustion. Consult a physician after significant exposure.

ON INGESTION Do not induce vomiting unless directed by medical personnel. Never give anything by mouth to an unconscious person. Get medical attention if symptoms occur and show the TDS.

ON EYES CONTACT Immediately flush eyes with plenty of water for at least 20 minutes. Get medical attention if symptoms occur.

5. FIRE-FIGHT MEASURES

FIRE EXTINGUISHING MEDIA	Water, dry chemical extinguisher, foam, carbon dioxide.
SPECIAL EXPOSURE HAZARDS	Carbon monoxide, carbon dioxide, acetaldehyde.
SPECIAL PROTECTIVE EQUIPMENT	Wear self-contained breathing apparatus, protective clothing and headgear to prevent contact with skin and eyes.

6. ACCIDENTIAL RELEASE MEASURES

PERSONAL PRECAUTIONS	Use suitable protective clothing. Avoid eye contact and inhalation of dusts. Keep ignition sources away.
METHODS FOR CLEANING	Sweep up material and place in a container, risk of slipping. Avoid ingress of material into drainage systems.

7. HANDLING AND STORAGE

HANDLING	Avoid contact with molten polymer. Avoid generation of dust and electrostatic charge.
STORAGE	Protect against moisture and UV radiation. Store cool and keep packaging closed when not in use. Avoid sources of ignition.

8. PERSONAL PROTECTION

TECHNICAL SAFETY MEASURES	With suitable ventilation the threshold limits assumably will not be reached. Avoid electrostatic charge by use of grounding cables.
PERSONAL SAFETY EQUIPMENT	Use adequate safety equipment, e.g. protective clothing, eye protection glasses, heat protection gloves. In case of dust formation wear mask with particle filter.
WORK HYGIENE	No eating or drinking during working. Avoid contact of hot material with the skin. Avoid breathing dust and vapours.

9. PHYSICAL AND CHEMICAL PROPERTIES

FORM	Filament
COLOR	colored
MELTING RANGE	180-210°C
OXIDISING PROPERTIES	not self igniting / flammable
DENSITY	1.30g/cm ³

10. STABILITY AND REACTIVITY

STABILITY	The product is stable at recommended storage conditions.
CONDITIONS TO BE AVOIDED	Avoid exposure to extreme heat and all sources of ignition. Thermal decomposition > 250 °C.
SUBSTANCES TO BE AVOIDED	Acetic Anhydride, acetone, aniline, benzene, chloroform, chromic acid, cyclohexanone, dimethylformamide, dioxane, ethyl acetate, phenol, tetrahydrofuran. Reactive with strong oxidizing agents, as well as strong acids and caustic will decompose polyester.
HAZARDOUS DECOMPOSITION PRODUCTS	Carbon monoxide, carbon dioxide, acetaldehyde.

11. TOXICOLOGICAL INFORMATION

LOCAL IRRITATION	No known significant effects or critical hazards.
OTHER REMARKS	No known significant effects or critical hazards.

12. ECOLOGICAL INFORMATION

ECOTOXICAL EFFECTS	not available
BIOLOGICAL DEGRADATION	not available
BIOACCUMULATION	No known significant effects or critical hazards.

13. DISPOSAL CONSIDERATIONS

PRODUCT	Generation of waste should be minimised, check possibility for recycling. Waste product can be incinerated or dumped together with domestic waste in compliance with local authority requirements.
SPECIAL PRECAUTIONS	Do not dump into any sewers, on the ground, or into any body of water. All disposal methods must be in compliance with all Federal, State/Provincial and local laws and regulations.

14. TRANSPORT INFORMATION

SPECIAL PRECAUTIONS FOR USER RELATED TRANSPORT OR TRANSPORTATION MEASURES	Local transport follows in accordance with Dangerous goods Safety Management Law. Package and transport follow in accordance with Department of Transportation (DOT) and other regulatory agency requirements. EmS FIRE SCHEDULE : Not available EmS SPILLAGE SCHEDULE : Not available
---	--

15. REGULATORY INFORMATION

EU REGULATIONS	Safety, health and environmental regulations/legislation specific for the substance or mixture EU Regulation (EC) No. 1907/2006 (REACH)
----------------	---

16. OTHER INFORMATION

This data is based on the current state of our information and experience. This safety data sheet describes our product in terms of safety requirements. Preceding data is not applicable as a warranty of product properties. It is the responsibility of the recipient to observe the existing legal regulations for the use of this product.