

**Solicitud n.º 2024-2-HU01-KA210-VET-000271388,
Aplicación de tecnologías de la Industria 4.0 en la
producción y operación de hogares inteligentes.**



Contenido

1.	Introducción al proyecto.....	4
1.1.	Introducción – El proyecto y el entorno educativo.....	4
1.2.	Propósito y función del manual.....	4
1.3.	Breve introducción y objetivos del proyecto	5
1.4.	La idea básica del proyecto “casa inteligente / granja inteligente”.....	6
1.5.	Arquitectura educativa integrada de la Industria 4.0 y entorno de desarrollo internacional.....	6
2.	Estructura del proyecto y lógica de implementación	7
2.1.	Fases del proyecto y su interdependencia.....	7
2.2.	Participantes y funciones.....	8
2.3.	Resultados de aprendizaje y competencias a desarrollar	9
3.	Presentación y procesamiento detallados del kit de granja inteligente.....	9
3.1.	El papel del Kit de Granja Inteligente en el proceso de aprendizaje.....	9
3.2.	Componentes principales del kit de granja inteligente.....	9
3.2.1.	Controlador y módulos básicos.....	10
3.2.2.	Elementos de entrada – Sensores.....	11
3.2.3.	Elementos de salida: actuadores y pantallas.....	14
3.2.4.	Accesorios y herramientas de montaje.....	17
3.2.5.	El papel de los componentes en la educación	17
3.3.	El papel de las instrucciones de montaje de fábrica	17
3.3.1.	Ensamblaje mecánico y electrónico y diseño modular	18
3.3.2.	Consideraciones de ingeniería basadas en la experiencia de montaje.....	19
3.4.	Programación del kit de granja inteligente	20
3.4.1.	Entorno de programación basado en bloques.....	20
3.4.2.	Conexión a entornos de programación basados en texto.....	21
3.4.3.	Vinculación con etapas posteriores del proyecto.....	22
3.4.3.1.	Proyecto 1 – Control de Iluminación (Sistema de Iluminación).....	22
3.4.3.2.	Proyecto 2 – Sistema de control de iluminación	26
3.4.3.3.	Proyecto 3 – Detección de distancia e intervención automática (Sistema de alimentación inteligente).....	29
3.4.3.4.	Proyecto 4 – Sistema de control de temperatura	33
3.4.3.5.	Proyecto 5 – Sistema de riego automático.....	38
3.4.3.6.	Proyecto 6 – Sistema de granja inteligente controlado por WiFi.....	42
3.4.3.7.	Resumen de la interpretación pedagógica.....	45
3.5.	Implementación práctica del capítulo.....	47
4.	Diseña digitalmente tu propio modelo de hogar inteligente/granja inteligente (Fusion 360).....	47
4.1.	Punto de partida: análisis del diseño físico del Kit de Granja Inteligente.....	48
4.2.	Limitaciones de la impresión 3D e implicaciones de diseño.....	49
4.3.	Concepto físico como descripción del sistema.....	50
4.4.	Entorno de diseño Fusion 360: conceptos básicos y enfoque.....	50
4.4.1.	Lienzo – aprendizaje del movimiento espacial	52
4.4.2.	Conceptos básicos de geometría y diseño – de forma tangible.....	54

4.5.	Preparación para la impresión 3D: enfoque práctico final	56
4.6.	Resultados del proceso de diseño digital: presentación e interpretación de los modelos de los estudiantes	57
4.7.	Resumen de los resultados de aprendizaje basados en los modelos completados.....	60
4.8.	Implementación práctica del capítulo.....	60
5.	Experiencias en impresión y fabricación 3D.....	61
5.1.	Presentamos la impresora 3D y Bambu Studio.....	61
5.1.1.	Breve descripción del procedimiento aplicado.....	61
5.1.2.	Estructura y funcionamiento del hardware.....	61
5.1.3.	El sistema de alimentación de filamentos AMS.....	62
5.1.4.	Principios de segmentación	63
5.1.5.	Materiales (PLA, PETG, ABS)	64
5.2.	Pasos del proceso de impresión.....	64
5.2.1.	Preparación y verificación del modelo.....	64
5.2.2.	Impresión y postproducción	65
5.2.3.	Control de calidad y corrección de errores	66
5.3.	Problemas de talla y ajuste.....	67
5.3.1.	Manejo de imprecisiones de fabricación	67
5.3.2.	Iteraciones de diseño.....	67
5.3.3.	La importancia educativa de los procesos, las experiencias de los estudiantes y las lecciones aprendidas.....	67
6.	Sistema de control integrado – fundamentos conceptuales.....	69
6.1.	Propósito del diseño del sistema.....	69
6.2.	Ventajas técnicas de la arquitectura distribuida	71
6.2.1.	Modelo y enfoque de comunicación	71
6.2.2.	Nodo de control de campo – unidad de muestreo independiente	72
6.3.	Estructura del programa dividida en unidades funcionales	73
6.3.1.	Nodo de control de campo con expansión de red – WiFi + Modbus TCP.....	76
6.3.2.	Nivel de supervisión – HMI basado en Python / Ciclo de supervisión	82
6.4.	Preparación e implementación del sistema de monitorización basado en Raspberry Pi.....	83
6.5.	El rol del cliente Python	86
6.6.	Registro de datos de sensores en la base de datos SQL.....	89
6.7.	Capa de gestión web: su función en el sistema.....	91
6.8.	Desarrollo de la interfaz HMI web por parte de los estudiantes en un trabajo en equipo	95
6.9.	Resumen: integración del sistema y resultados del aprendizaje	97
7.	Resumen	99

1. Presentación del proyecto

1.1. Introducción – El proyecto y el entorno educativo

Este manual constituye la documentación profesional y metodológica de un proyecto educativo internacional colaborativo basado en la Industria 4.0 y las tecnologías IoT. El proyecto se centra en un modelo de casa/granja inteligente, cuyo desarrollo involucrará a estudiantes de diversas disciplinas, desde diseño digital y fabricación aditiva hasta integración electrónica y desarrollo de software, culminando en la creación de un complejo sistema de gestión de edificios basado en la web.

El proyecto comenzó con un kit de capacitación comercial (Smart Farm Kit), que proporcionó una buena base para aprender sobre sensores, actuadores y control basado en IoT. Sin embargo, el proceso de desarrollo no se limitó a la aplicación de soluciones industriales: los participantes realizaron análisis críticos, identificaron las limitaciones del sistema y, posteriormente, respondieron a ellas desarrollando su propia arquitectura de sistema escalable, siguiendo un modelo industrial.

El objetivo del manual no es solo documentar la implementación de un proyecto específico, sino también presentar un modelo educativo abierto y adaptable que otras instituciones y grupos de aprendizaje puedan seguir y desarrollar aún más, incluso sin el uso de elementos fabricados en serie.

1.2. Propósito y función del manual

El objetivo de este manual es brindar una guía profesional y metodológica integral para un proyecto educativo que fomente el desarrollo de las competencias digitales, técnicas y de ingeniería de los estudiantes mediante la aplicación práctica de las tecnologías de la Industria 4.0 y el IoT. El documento no solo ofrece soluciones predefinidas, sino que también presenta un proceso de aprendizaje y desarrollo en el que la resolución de problemas, el análisis y la iteración desempeñan un papel fundamental.

El manual parte de la premisa de que la educación no se basa en la transmisión pasiva de conocimientos, sino en la actividad creativa activa. Durante el proyecto, los estudiantes no son meros usuarios, sino diseñadores, implementadores y desarrolladores de un sistema técnico complejo. El documento respalda este enfoque en todo momento, desde el diseño, la producción y la programación hasta la integración del sistema.

La función principal del manual es servir como material didáctico. Su contenido es adecuado para:

- servir de base para el trabajo en proyectos de clase y extracurriculares,
- encajar en programas de formación profesional, gestión del talento o formación basada en proyectos,
- Proporcionar apoyo metodológico a los docentes en la implementación de proyectos técnicos y de TI complejos.

El enfoque metodológico se centra en el aprendizaje basado en proyectos y en el aprendizaje basado en el descubrimiento. Los estudiantes progresan gradualmente desde subareas más simples hasta la comprensión de sistemas complejos en pasos secuenciales. En consecuencia, el manual no es una única "correcta"

No solo registra una solución, sino que también presenta alternativas, puntos de decisión y oportunidades de desarrollo.

El documento es relevante para varios grupos objetivo:

- Para los estudiantes: el manual presenta los procesos de desarrollo con una lógica accesible para ellos, brindando la oportunidad de experimentar de forma independiente, cometer errores y rediseñar. Durante el proyecto, los estudiantes experimentan los fundamentos del pensamiento de ingeniería: definen un problema, diseñan una solución, la prueban, la evalúan y la perfeccionan.
- Para los educadores: el manual facilita la organización de las lecciones, la asignación diferenciada de tareas y la gestión de grupos heterogéneos. Ofrece orientación sobre cómo involucrar a estudiantes con diferentes perfiles en un proyecto común y cómo abordar el contenido técnico de forma pedagógicamente estructurada pero flexible.
- Para las instituciones: el proyecto presenta un modelo adaptable que puede integrarse en los planes de estudio locales, las semanas de proyectos o las colaboraciones internacionales. Su carácter abierto permite a las instituciones adaptar la implementación a sus propios equipos, infraestructura y perfil de formación.

1.3. Breve introducción y objetivos del proyecto

Conexión con la formación técnica y secundaria en informática: el proyecto se inició en un entorno técnico, donde los estudiantes de informática industrial, mecatrónica y otras disciplinas afines ya poseían conocimientos tecnológicos básicos. Sin embargo, durante la cooperación internacional, se incorporó al programa un instituto de enseñanza secundaria teórica, cuyos alumnos carecían de experiencia técnica previa. Esta situación representó tanto un reto como una oportunidad pedagógica.

Ventajas:

- La colaboración de estudiantes con diferentes procedencias fortaleció el pensamiento explicativo y sistemático.
- Los alumnos de la escuela técnica asumieron el papel de mentores, lo que profundizó sus propios conocimientos.
- Para los estudiantes del liceo teórico, el proyecto supuso un puente entre el conocimiento abstracto de la informática y los sistemas físicos.

Posibles dificultades:

- La falta de conocimientos técnicos básicos puede haber ralentizado el progreso inicial.
- Era necesario hacer mayor hincapié en la introducción gradual de conceptos básicos (electrónica, sensores, control).
- El papel del instructor se ha vuelto más facilitador y de apoyo.

Basándonos en la experiencia del proyecto, se puede afirmar que el modelo presentado no solo es funcional en un entorno de escuela técnica. Con una estructura metodológica y una diferenciación adecuadas, también puede aplicarse con éxito en instituciones de enseñanza secundaria no técnicas.

1.4. La idea básica del proyecto “casa inteligente / granja inteligente”

La idea fundamental del proyecto es crear un modelo físico y digital que ilustre de forma tangible el funcionamiento de los sistemas inteligentes, presentando la aplicación de las tecnologías modernas a través de problemas comprensibles y reales para los estudiantes. El modelo de casa inteligente/granja inteligente ofrece la oportunidad de integrar aspectos técnicos, informáticos y ambientales como un sistema unificado en la educación.

El modelo es adecuado para presentar simultáneamente:

- el papel de los sensores y actuadores en el monitoreo y control de las condiciones ambientales,
- el proceso de recopilación de datos, procesamiento de datos y retroalimentación,
- la lógica del control y la supervisión en entornos locales y remotos,
- Opciones de acceso remoto basadas en la web como herramienta para un funcionamiento eficiente en el uso de energía y recursos.

En este proyecto, la casa inteligente y la granja inteligente no solo se presentan como una demostración tecnológica, sino también como un vehículo para un enfoque respetuoso con el medio ambiente. Mediante la medición con sensores y la intervención automatizada, los estudiantes aprenden cómo el control inteligente puede contribuir a la optimización del consumo energético, el uso responsable de los recursos ambientales y la operación sostenible (por ejemplo, iluminación, ventilación, control de riego).

Este modelo no es un producto final, sino una herramienta de aprendizaje abierta y en constante evolución que fomenta el desarrollo del pensamiento sistémico mediante ciclos de diseño, experimentación, evaluación y rediseño. Este enfoque ayuda a los estudiantes no solo a dominar las soluciones técnicas, sino también a comprender sus contextos ambientales, económicos y sociales.

1.5. Arquitectura educativa integrada de la Industria 4.0 y entorno de desarrollo internacional

El proyecto sigue los principios del enfoque de la Industria 4.0:

- arquitectura de sistemas distribuidos,
- operación basada en datos,
- uso de dispositivos en red,
- utilizando protocolos de comunicación estándar.

El uso de las tecnologías IoT permite a los estudiantes no trabajar con dispositivos aislados, sino interpretar los componentes de hardware y software como parte de un sistema coherente.

El kit Smart Farm se diseñó deliberadamente como punto de partida. Uno de sus elementos clave fue analizar el sistema de la fábrica, identificar sus limitaciones y, posteriormente, abordarlas desarrollando una arquitectura de sistema escalable propia que sigue un patrón industrial. Este proceso puso de manifiesto que el pensamiento crítico y el desarrollo iterativo son los pilares del progreso tecnológico.

El enfoque se basa completamente en proyectos. Los estudiantes no siguen pasos predefinidos, sino que resuelven problemas, toman sus propias decisiones y analizan sus consecuencias. Durante el proceso de aprendizaje, los errores no se consideran fracasos, sino una parte natural del desarrollo.

Los modelos físicos, la retroalimentación inmediata y los resultados visibles aumentan de forma natural la motivación de los estudiantes. Elementos de gamificación como desafíos, hitos y objetivos compartidos aparecen a lo largo del proyecto para estructurar y mantener el interés.

Desarrolla competencias digitales de forma integral: programación, gestión de bases de datos, conocimientos de redes, diseño digital y documentación. Además, fortalece el pensamiento ingenieril, el pensamiento sistémico y el reconocimiento de las relaciones entre unidades funcionales.

El proyecto se implementó mediante cooperación internacional, lo que aportó un valor añadido significativo en términos de desarrollo de competencias y contenido pedagógico, además del contenido profesional. Las instituciones participantes operan en diferentes países, sistemas educativos y perfiles institucionales, por lo que el trabajo conjunto creó un entorno de colaboración real que refleja fielmente las futuras situaciones del mercado laboral.

Las diferencias lingüísticas fueron evidentes durante las actividades conjuntas. En muchos casos, la comunicación profesional se desarrolló en inglés, lo que mejoró las competencias lingüísticas de los estudiantes y les hizo conscientes de la importancia de una comunicación técnica precisa y clara.

La colaboración multicultural se convirtió en parte natural del proceso de trabajo. El trabajo conjunto de estudiantes con diferentes antecedentes educativos y estrategias de resolución de problemas fortaleció la flexibilidad, la adaptabilidad y el diálogo profesional.

Su funcionamiento reflejaba las expectativas de las empresas multinacionales en varios aspectos: comunicación en idiomas extranjeros, gestión conjunta de la documentación, trabajo en equipo y responsabilidad en sistemas complejos. Las experiencias adquiridas contribuyeron a que los estudiantes pudieran interpretar sus conocimientos técnicos en un contexto internacional más amplio.

2. Estructura del proyecto y lógica de implementación

2.1. Fases del proyecto y su interdependencia

El proyecto se implementó por etapas que se construyeron de forma consciente unas sobre otras. Esta estructura no solo respondía a una necesidad técnica, sino también a una decisión pedagógica: el objetivo era que los estudiantes, partiendo de sus propias experiencias, alcanzaran gradualmente tareas de desarrollo independientes. La división en fases permitió la participación de estudiantes con diferentes perfiles, así como la reflexión y la evaluación de cada etapa.

En la primera fase, los estudiantes se encontraron con un sistema funcional ya preparado. El objetivo era aprender los conceptos básicos, las relaciones operativas y la lógica del sistema. El énfasis no estaba en «copiar» la solución, sino en comprender: qué hace el sistema, por qué lo hace, de qué elementos se compone y cómo se conectan entre sí.

Esta etapa resultó ser particularmente importante para los estudiantes con menos conocimientos técnicos, ya que creó un punto de partida común y redujo las barreras de entrada para tareas posteriores más complejas.

La segunda fase, la de análisis consciente, siguió a la fase de familiarización. Los estudiantes no solo utilizaron el sistema, sino que también examinaron sus límites operativos, fortalezas y debilidades. Este paso fue fundamental para comprender que una solución técnica siempre es el resultado de concesiones y se optimiza para fines específicos.

Mediante la evaluación crítica, aprendieron a formular sus propias necesidades de desarrollo y a reconocer cuándo es apropiado seguir desarrollando o replantear un sistema existente. Esta forma de pensar está directamente relacionada con los principios de la ingeniería y la resolución de problemas informáticos.

En la tercera fase del proyecto, los estudiantes iniciaron su propio trabajo de desarrollo, partiendo de sus experiencias analíticas. En esta etapa, el proceso ya no se guiaba por pasos predeterminados, sino por decisiones conjuntas, planificación iterativa y pruebas continuas.

En esta etapa, la ventaja del enfoque por proyectos se hizo realmente evidente: los estudiantes experimentaron que un sistema complejo no se "completa" de una sola vez, sino que va tomando forma gradualmente, y que cada decisión tiene un impacto en toda la operación.

2.2. Participantes y roles

Una característica distintiva del proyecto fue la división consciente de roles entre los participantes. Estudiantes, profesores e instituciones colaboraron desempeñando funciones diferentes pero que se reforzaban mutuamente, lo que contribuyó a la flexibilidad y sostenibilidad del proyecto.

Durante su participación, los estudiantes no fueron receptores pasivos, sino desarrolladores activos. Colaboraron en la planificación, la interpretación de problemas, las pruebas y la evaluación de soluciones. Al distribuir las subtareas, experimentaron la importancia de asumir responsabilidades y cómo su trabajo se integra en un sistema más complejo. Este rol fortaleció su independencia, cooperación y visión a largo plazo.

El papel de los instructores fue principalmente de apoyo y orientación. El énfasis se puso en guiar, cuestionar y estructurar el proceso de aprendizaje, en lugar de proporcionar soluciones prefabricadas.

Este enfoque permitió a los estudiantes encontrar soluciones a través de sus propias experiencias, al tiempo que les proporcionaba una sólida base profesional y pedagógica.

La cooperación internacional dio lugar a grupos de aprendizaje mixtos, en los que estudiantes con diferentes antecedentes y experiencias educativas trabajaron juntos. Esta situación fortaleció las habilidades comunicativas, la adaptabilidad y la resolución conjunta de problemas.

El trabajo en equipo multidisciplinario acompañó todo el proyecto y creó situaciones de colaboración reales que trascendieron el marco escolar tradicional.

2.3. Resultados de aprendizaje y competencias a desarrollar

Durante la fase de diseño, se buscó conscientemente no limitar los resultados del aprendizaje a una sola área de competencia. El proceso de desarrollo fomentó simultáneamente el desarrollo de habilidades técnicas, cognitivas y sociales.

El proyecto contribuyó al uso consciente de herramientas digitales, al trabajo estructurado y a la creación en un entorno digital. Los estudiantes experimentaron cómo un sistema complejo se vuelve comprensible y manejable.

Se enfrentaron a problemas complejos que requerían múltiples pasos, planificación y retroalimentación para su solución. Esto fortaleció el pensamiento sistémico, el reconocimiento de las relaciones de causa y efecto y la capacidad de planificar a largo plazo.

Documentar y presentar el trabajo realizado fue una parte fundamental del proyecto. Los estudiantes aprendieron a registrar de forma transparente el proceso de desarrollo y a presentarlo de manera comprensible. Los portafolios resultantes también permiten aprovechar a largo plazo los resultados del aprendizaje.

3. Presentación y procesamiento detallados del Kit de Granja Inteligente El

3.1. papel del Kit de Granja Inteligente en el proceso de aprendizaje

El kit Smart Farm es un kit modular de IoT diseñado con fines educativos, que demuestra los principios básicos de los sistemas inteligentes mediante un modelo tangible. El objetivo del kit es familiarizar a los estudiantes con un sistema real y funcional antes de que comiencen a desarrollar sus propias soluciones.

En este proyecto, el kit no se presenta como un producto final, sino como una herramienta de aprendizaje. Durante su desarrollo, se hace hincapié en comprender el funcionamiento de los elementos, mapear la lógica del sistema y sentar las bases para su posterior desarrollo.

3.2. Componentes principales del kit de granja inteligente

Los componentes electrónicos del Kit de Granja Inteligente están diseñados para demostrar un sistema IoT complejo, que incluye múltiples sensores de entrada, módulos de control y actuadores de salida. La lista se basa en la información del fabricante y en descripciones de productos fiables, y abarca los dispositivos reales utilizados durante el programa de aprendizaje del kit.

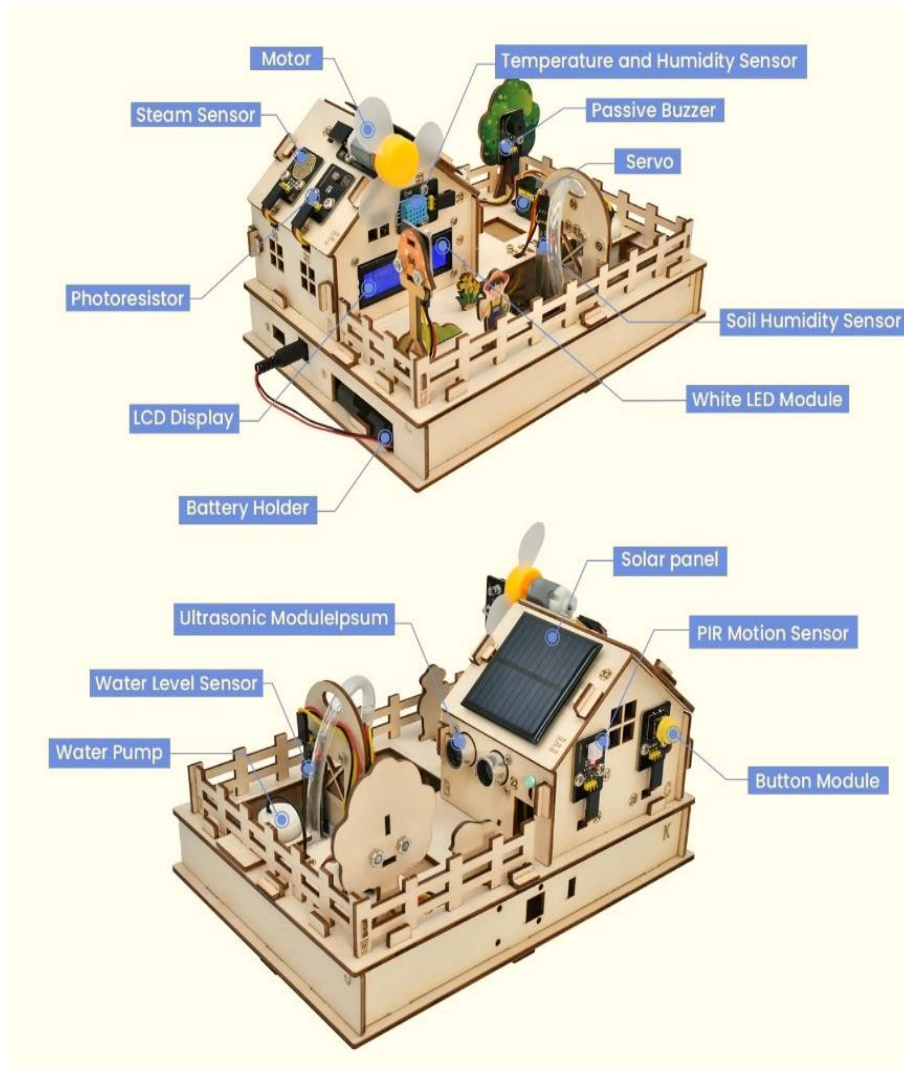


Figura 1: Partes principales del Smart Farm Kif (Fuente: <https://www.keyestudio.com>)

3.3. Módulos de control y base

- Microcontrolador ESP32:
 - él Estructura: Microcontrolador integrado con módulo Wi-Fi y Bluetooth.
 - él Función: El "cerebro" del kit, encargado de gestionar los datos de los sensores, la lógica y la comunicación.
 - él Importancia educativa: Estudio de la comunicación en red, el manejo de eventos paralelos y la lógica de entrada/salida.

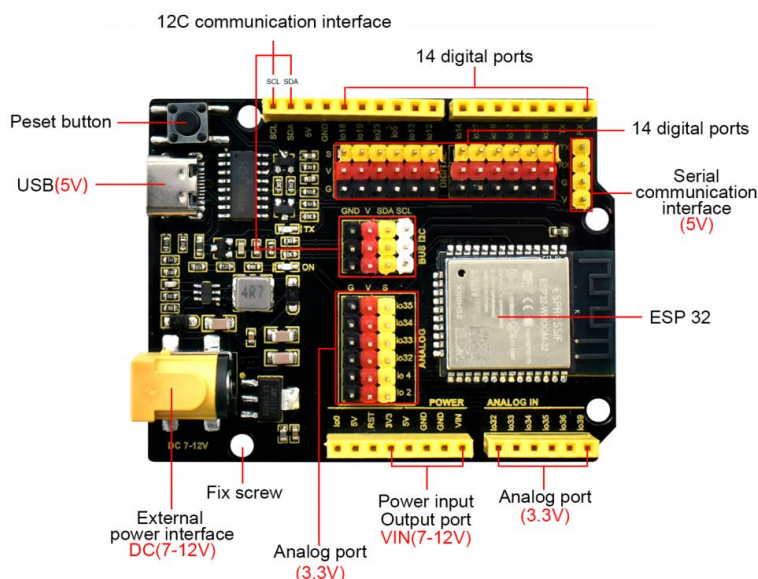


Figura 2: Microcontrolador ESP32 (Fuente: <https://docs.keyestudio.com>)

- Módulo de relé:
 - él Construcción: Relé optoaislado con control de 5 V
 - él Función: Control de dispositivos de alto consumo (p. ej., bombas).
 - él Objetivo educativo: Aislamiento y control seguro de actuadores.

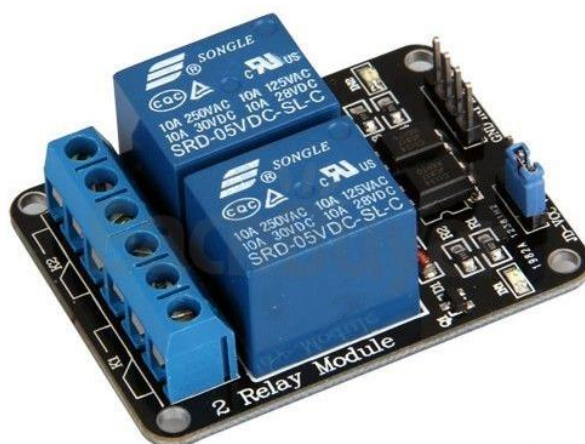


Figura 3: Módulo de relé (Fuente: <https://docs.keyestudio.com>)

3.4. Elementos de entrada: sensores

- Fotorresistencia: él
 - Construcción: resistencia semiconductor fotosensible
 - él Funcionamiento: la resistencia varía en función de la intensidad de la luz (señal analógica). Objetivo
 - él educativo: medición de condiciones de iluminación, manejo de entradas analógicas.



Figura 4: Fotorresistencia (Fuente: <https://docs.keyestudio.com>)

- Sensor de humedad del suelo: él
 - Él Estructura: dos electrodos y transductor
 - él Funcionamiento: el agua en el suelo reduce la resistencia → mayor valor de la señal. Objetivo
 - él educativo: técnicas de medición analógica e interpretación de datos ambientales.

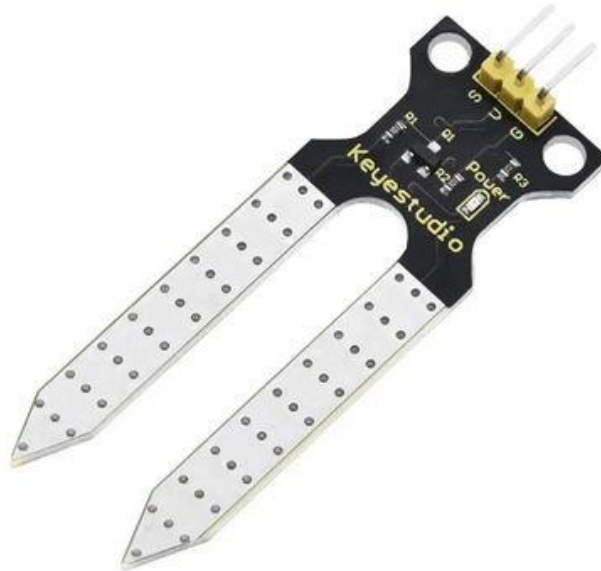


Figura 5: Sensor de humedad del suelo (Fuente: <https://docs.keyestudio.com>)

- Sensor de nivel de agua:
 - él Estructura: contacto multielectrodo
 - él Funcionamiento: señal de apertura/cierre basada en el nivel del
 - él líquido. Finalidad educativa: medición digital basada en eventos.

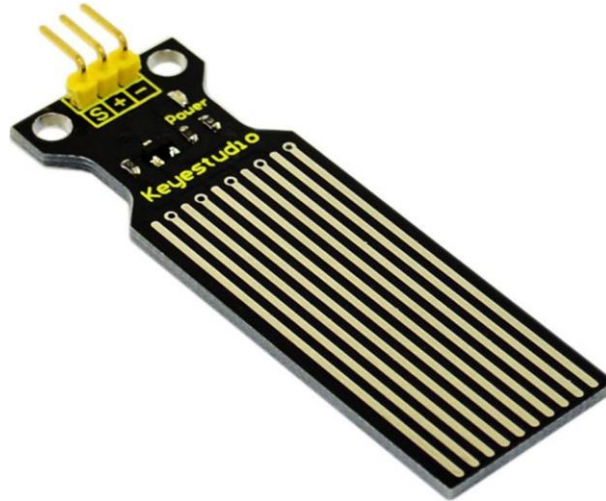


Figura 6: Sensor de nivel de agua (Fuente: <https://docs.keyestudio.com>)

- Sensor de temperatura y humedad DHT11: él
Estructura: sensor digital integrado en la unidad.
él Función: calcula la temperatura y la humedad mediante un algoritmo integrado.
él Finalidad educativa: comunicación digital de datos y medición del entorno físico.

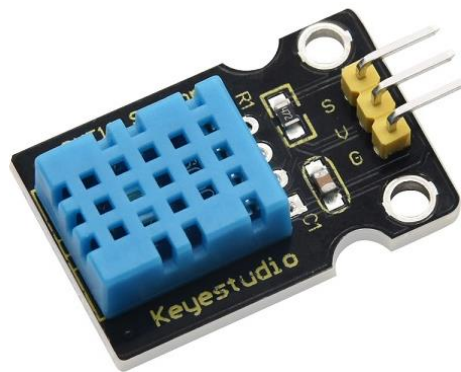


Figura 7: Sensor de temperatura y humedad DHT11 (Fuente: <https://docs.keyestudio.com>)

- Sensor de distancia ultrasónico SR01 V3: él
Estructura: elementos piezoeléctricos transmisor y receptor.
él Funcionamiento: pulso ultrasónico → reflexión → cálculo de distancia a partir de la medición del tiempo.
él Objetivo educativo: comprender las ondas físicas y la medición basada en el tiempo.



Figura 8: Sensor de distancia ultrasónico SR01 V3 (Fuente: <https://docs.keyestudio.com>)

- Sensor de movimiento PIR:

- él Construcción: sensor infrarrojo pasivo
- él Función: Detecta cambios bruscos de temperatura en el campo de visión. Objetivo
- él educativo: Lógica basada en eventos y detección en tiempo real.



Figura 9: Sensor de movimiento PIR (Fuente: <https://docs.keyestudio.com>)

- Zumbador pasivo: él
 - Construcción: altavoz piezoeléctrico
 - él Funcionamiento: responde a pulsos de voltaje emitiendo sonido. Objetivo educativo:
 - él conexión entre módulos de salida y sistema de señalización práctico.

3.5. Elementos de salida: actuadores y pantallas

- Módulo LED blanco:
 - él Construcción: LED premontado
 - él Funcionamiento: la salida digital controla la iluminación. Finalidad
 - él educativa: retroalimentación visual para lógica simple.



Figura 10: Módulo LED blanco (Fuente: <https://docs.keyestudio.com>)

- Servomotor SG90 9G: él
 - Estructura: micro servomotor con retroalimentación de posición.
 - él Funcionamiento: control de posición basado en señal PWM.

élObjetivo educativo: control preciso de actuadores y efectos mecánicos.



Figura 11: Servomotor SG90 9G (Fuente: <https://docs.keyestudio.com>)

- Motor/ventilador de CC:
 - él Construcción: Motor de CC con ventilador pequeño
 - él Función: dispositivo de salida que sirve para mover el aire ambiente. Finalidad
 - él educativa: control de motores y monitorización del rendimiento.

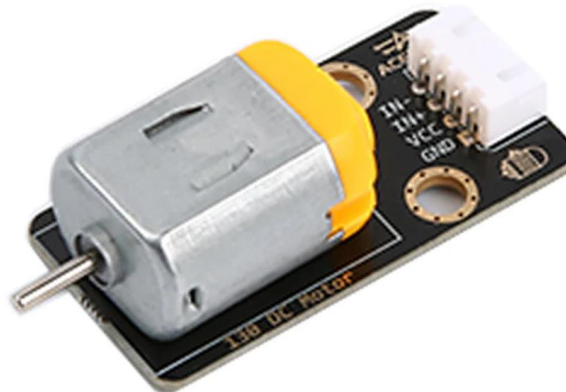


Figura 12: Motor/ventilador de CC (Fuente: <https://docs.keyestudio.com>)

- Pantalla LCD I2C 1602:
 - él Estructura: 2 líneas × 16 caracteres, comunicación I2C
 - él Función: mostrar datos de caracteres a partir de mensajes del microcontrolador. Objetivo
 - él educativo: aprender la visualización de datos y los protocolos de comunicación.

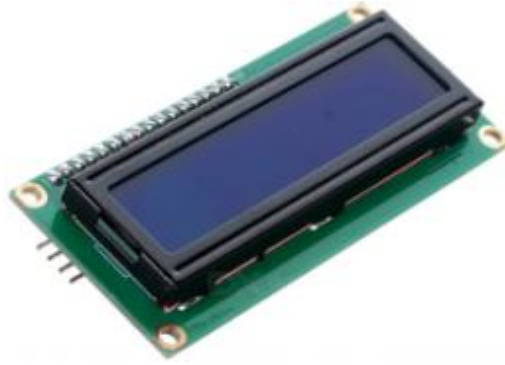


Figura 13: Pantalla LCD I2C 1602 (Fuente: <https://docs.keyestudio.com>)

- Panel solar:

- él Estructura: pequeño panel fotovoltaico. Funcionamiento: efecto fotovoltaico → electricidad. Objetivo educativo: fuentes de energía renovables en sistemas IoT.



Figura 14: Panel solar (Fuente: <https://docs.keyestudio.com>)

- Bomba de agua:

- él Construcción: pequeña bomba de CC
- él Funcionamiento: transferencia de líquido con control por relé
- él Objetivo educativo: comprender el control de los actuadores y el trabajo mecánico.



Figura 15: Bomba de agua (Fuente: <https://docs.keyestudio.com>)

3.6. Accesorios y herramientas de montaje

- elementos acrílicos transparentes y revestimiento de virutas de madera
- cables de alimentación, cables DuPont, tornillos, sujetadores
- soporte o portapilas
- Cable USB para comunicación y alimentación

3.7. El papel de los componentes en la educación

Los 17 componentes electrónicos incluidos en el kit cubren adecuadamente el programa de introducción al IoT y la automatización:

- Los sensores proporcionan transmisión de señales desde el entorno físico,
- La unidad de control interpreta y decide,
- Los actores se manifiestan o intervienen en el entorno.

Esta estructura proporciona una visión general sistemática de cómo los elementos funcionan juntos en un sistema IoT real y cómo se pueden conectar mediante la programación, la lógica de medición y el control.

El montaje del kit de granja inteligente es el primer paso del proyecto, donde los estudiantes trabajan con un sistema físico complejo y multicomponente. La documentación del fabricante guía al usuario paso a paso en el montaje de los elementos mecánicos y electrónicos, lo que también puede interpretarse como un proceso de aprendizaje consciente.

El montaje no es simplemente una tarea técnica, sino una experiencia que sienta las bases para el posterior trabajo independiente de diseño y fabricación.

3.8. El papel de las instrucciones de montaje de fábrica

El manual de fábrica del kit Smart Farm sirve como referencia principal durante el montaje. El manual:

- determina la secuencia de cada paso de ensamblaje,
- muestra la posición y fijación correctas de las piezas,
- ayuda a conectar de forma segura los componentes electrónicos,
- Facilita la comprensión mediante ilustraciones visuales.

Al utilizar esta guía, los estudiantes comprobarán que el montaje de un sistema de ingeniería no es una cuestión de improvisación, sino un proceso estructurado y basado en la documentación.

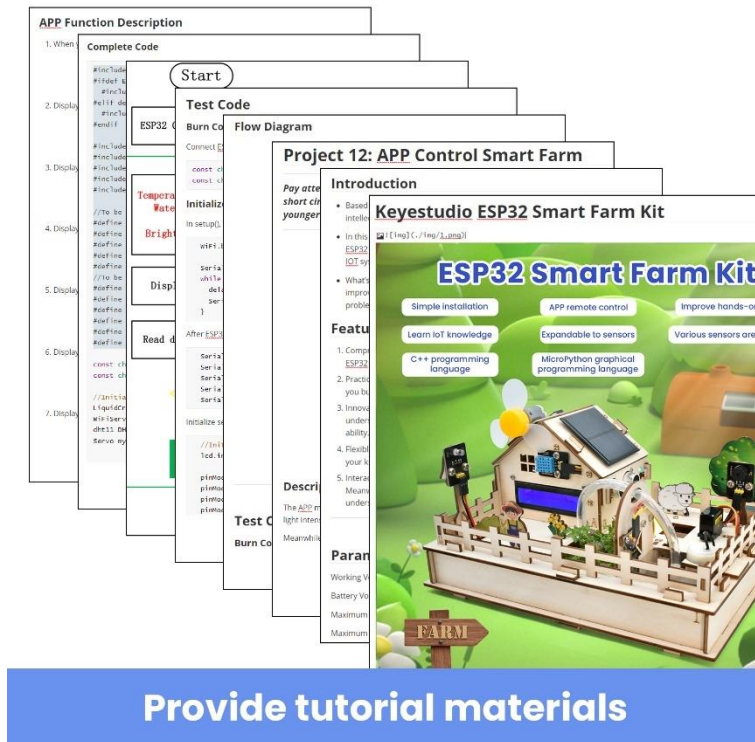


Figura 16: Instrucciones de montaje de fábrica (Fuente: <https://www.keyestudio.com>)

3.9. Ensamblaje mecánico y electrónico y diseño modular

Durante el montaje, los alumnos conocerán el diseño mecánico modular del kit. Cada elemento cuenta con puntos de montaje prediseñados basados en distancias de conexión estándar, lo que permite la sustitución y reubicación de los componentes.

Este enfoque modular transmite un principio de ingeniería importante: la capacidad de expansión y el mantenimiento del sistema se deciden a nivel de diseño físico.

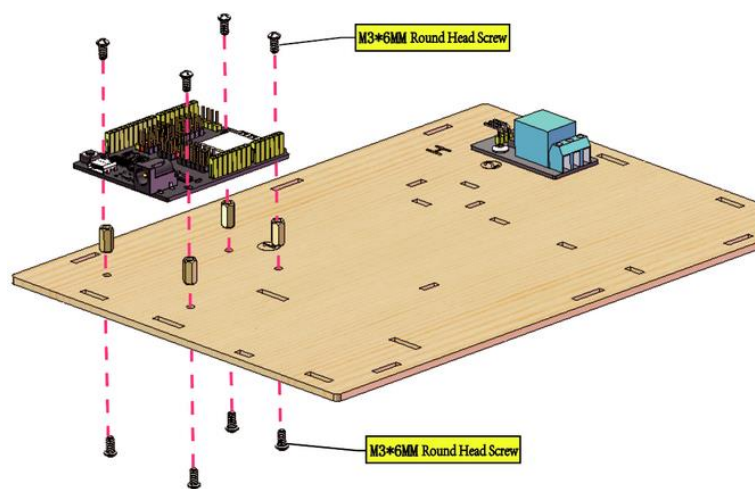


Figura 17: Placa base para hogar inteligente (Fuente: <https://docs.keyestudio.com>)

Durante el montaje de componentes electrónicos, los estudiantes identifican los puntos de conexión de los sensores y actuadores, aprenden sobre la función de los cables de alimentación y de señal, y siguen los diagramas de cableado de fábrica.

Durante las conexiones, se presta especial atención al manejo correcto de la polaridad, la distinción entre entradas digitales y analógicas, y la protección mecánica y la claridad de los cables.

Este paso sienta las bases para la posterior resolución de problemas y el diagnóstico del sistema.

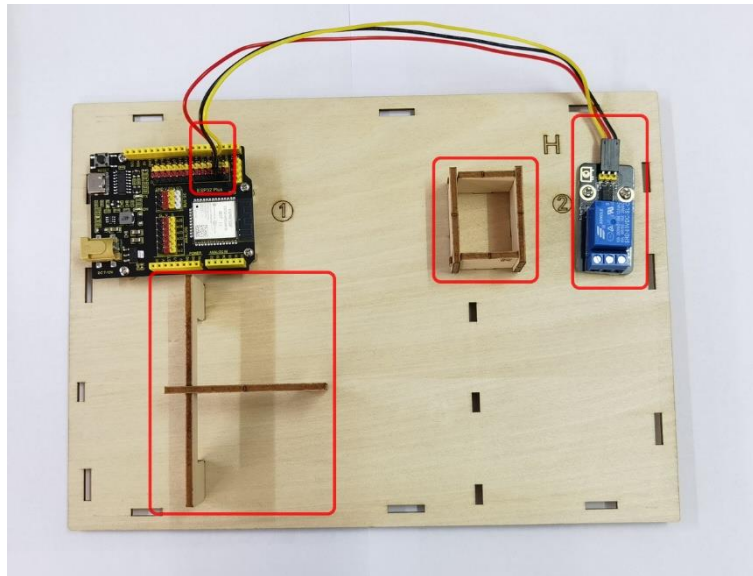


Figura 18: Puntos de conexión de la placa base para hogares inteligentes (Fuente: <https://docs.keyestudio.com>)

El ensamblaje no es un proceso único y cerrado, sino una actividad iterativa. Después de cada etapa del ensamblaje, los estudiantes:

- comprobar la estabilidad mecánica,
- ellos prueban las conexiones eléctricas,
- El funcionamiento de cada elemento se valida mediante breves programas de prueba.

Este enfoque previene errores complejos posteriores y resalta la importancia de la verificación gradual.

3.10. Consideraciones de ingeniería basadas en la experiencia de montaje.

Durante el proceso de ensamblaje, se hacen visibles varios aspectos de ingeniería que pueden utilizarse directamente en la posterior creación de elementos de diseño propio:

- El ajuste geométrico, la precisión de las dimensiones y los puntos de montaje de los componentes influyen fundamentalmente en el ensamblaje.
- El diseño de las rutas de cableado y la disposición de los cables influyen en la transparencia, la facilidad de mantenimiento y la estética.
- La accesibilidad y la facilidad de mantenimiento, así como la ubicación de los sensores y los controles, influyen en la posibilidad de realizar modificaciones y ampliaciones posteriores.

- En términos de modularidad e intercambiabilidad, la solución de fábrica pone de manifiesto cómo se puede construir un sistema de tal manera que los elementos individuales puedan funcionar de forma independiente.

Estas experiencias hacen que los estudiantes tomen conciencia de que el diseño físico y el funcionamiento funcional forman una unidad inseparable.

Al concluir la fase de ensamblaje, los estudiantes ya no son solo usuarios, sino también analistas del sistema. Las experiencias adquiridas aquí los preparan directamente para:

- las mediciones geométricas subsiguientes,
- tus propias tareas de diseño 3D,
- el diseño de fijaciones individuales y elementos de alojamiento,
- Desarrollo o sustitución de las soluciones de fábrica.

Por lo tanto, el montaje del kit de granja inteligente no es un punto final, sino un punto de partida para un proceso creativo de ingeniería independiente y de nivel superior.

3.11. Programación del entorno de programación basado

3.12. en bloques del kit de granja inteligente

La programación por bloques desempeña un papel fundamental en la comprensión de los sistemas y en la transición a otros modelos de programación. El entorno de programación gráfica del Smart Farm Kit (KidsBlock) sigue la lógica de Scratch. La programación se realiza mediante el ensamblaje de bloques visuales, respaldados por instrucciones de control reales y ejecutables. Esto permite a los estudiantes evitar tener que lidiar con los detalles sintácticos de los lenguajes de programación mientras controlan el funcionamiento real del sistema.

Desde una perspectiva educativa, este entorno cumple varias funciones que se refuerzan mutuamente:

- Separa el pensamiento lógico de las dificultades lingüísticas, de modo que los estudiantes puedan centrarse en los procesos de control, las relaciones de causa y efecto y los cambios de estado.
- Proporciona retroalimentación inmediata, ya que el funcionamiento del sistema físico se puede observar directamente después de modificar el programa.
- Permite la experimentación iterativa, el ciclo natural de aprendizaje basado en la mejora mediante ensayo y error.
- Te prepara para la programación textual posterior, ya que las estructuras básicas como el manejo de eventos, las pruebas de condiciones, los bucles y la gestión de estados aparecen detrás de los bloques.

La programación basada en bloques en este proyecto no es un punto final, sino un nivel de entrada elegido conscientemente en una ruta de aprendizaje de programación de varios pasos.

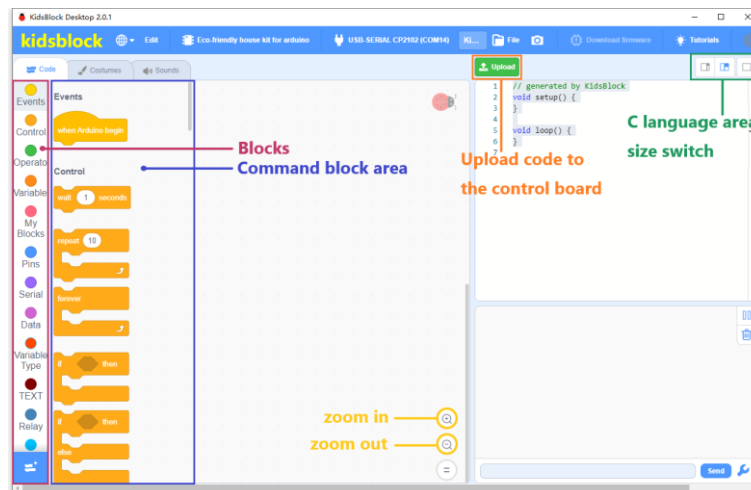


Figura 19: Interfaz de programación basada en bloques, descripción general de las principales categorías de bloques (Fuente: <https://docs.keyestudio.com>)

3.13. Conexión a eventos de programación basados en texto

La unidad de control principal del Smart Farm Kit, el microcontrolador ESP32, no solo es programable en un entorno gráfico. El proyecto también ofrece la oportunidad de implementar las mismas funciones en un lenguaje de programación basado en texto, como un entorno basado en Arduino.

El entorno Arduino:

- Utiliza sintaxis basada en C/C++,
- permite la gestión de hardware de bajo nivel,
- Te proporciona información directa sobre la ejecución del programa, las variables y el uso de la memoria.

En términos pedagógicos, este cambio:

- te ayuda a comprender que hay un código específico ejecutándose detrás de los elementos basados en bloques,
- desarrolla la capacidad de abstraer (correspondencia bloque ↔ línea de código),
- Te prepara para el uso de entornos de desarrollo más profesionales.

Durante la implementación, la programación basada en bloques y la programación basada en texto se presentan como herramientas complementarias en lugar de mutuamente excluyentes.

El valor educativo del proyecto Smart Farm se ve reforzado por el hecho de que la lógica de control que aprenden los estudiantes está estrechamente relacionada con los principios de programación utilizados en la automatización industrial.

Las oportunidades profesionales se amplían aún más gracias a la plataforma de software de código abierto OpenPLC, que permite la programación según la norma IEC 61131-3 no solo en PLC clásicos, sino también en controladores basados en Arduino y ESP32. El entorno OpenPLC ofrece compatibilidad total con el lenguaje IEC 61131-3, lo que posibilita la implementación de lógica de control industrial en hardware de bajo coste, y admite protocolos de comunicación industrial, en particular los estándares Modbus y Modbus TCP.

Este enfoque cierra la brecha entre el desarrollo educativo de microcontroladores y los sistemas de automatización industrial del mundo real. Los estudiantes pueden aprender sobre el

con programación industrial, que es tecnológicamente más sencilla pero totalmente compatible con los sistemas profesionales en su enfoque.

3.14. Enlace a fases posteriores del proyecto

En las fases posteriores del proyecto Granja Inteligente, la comunicación industrial desempeñará un papel fundamental. Por consiguiente:

- El uso del protocolo Modbus / Modbus TCP constituye la base de la integración del sistema,
- La programación en lenguaje C, basada en Arduino, todavía se utiliza en el control de microcontroladores.
- El enfoque basado en bloques y la norma IEC 61131-3 sirven principalmente como marco de pensamiento y diseño.

Esta decisión garantiza que los estudiantes aprendan sobre los estándares y las expectativas de la industria mientras trabajan en un entorno de desarrollo flexible y favorable a la educación, y que se preparen para trabajar posteriormente con sistemas basados en PLC o sistemas de IoT industrial.

El proyecto Smart Farm utiliza la programación por bloques no de forma aislada, sino integrada en un contexto industrial más amplio. La introducción del estándar IEC 61131-3, la plataforma/proyecto OpenPLC y la comunicación basada en Modbus permiten a los estudiantes adquirir un enfoque sistémico de nivel intermedio que se ajusta directamente a las expectativas del entorno industrial multinacional moderno, manteniendo al mismo tiempo la seguridad, el carácter gradual y motivador de la enseñanza.

3.15. Proyecto 1 – Sistema de iluminación

El control de la iluminación es uno de los proyectos de ejemplo más sencillos, pero a la vez didácticos, del Kit de Granja Inteligente. En esta tarea, la cadena de control de entrada digital → lógica de decisión → salida digital aparece por primera vez como una unidad completa y cerrada. El objetivo del proyecto no es comprender la iluminación como una función, sino entender la lógica básica de la automatización.

El botón pulsador como entrada digital

Un pulsador es el dispositivo de entrada digital más simple que puede transmitir dos estados distintos al controlador:

- estado no pulsado → nivel lógico BAJO (0),
- estado pulsado → nivel lógico ALTO (1).

Para el controlador (ESP32), el botón pulsador no es un "objeto físico", sino una señal digital que aparece en un pin de entrada específico. El estado de la entrada se estabiliza normalmente mediante una resistencia pull-up o pull-down interna o externa, de modo que el controlador detecta un valor ALTO o BAJO claro en todo momento.

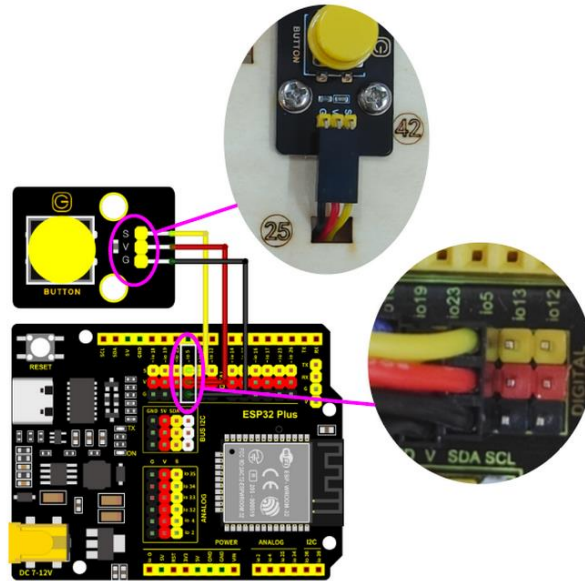


Figura 20: Diagrama de cableado del pulsador como entrada digital (GPIO del ESP32) (Fuente: <https://docs.keyestudio.com>)

El pulsador es excelente para introducir el concepto de señal digital, ya que no hay un valor "intermedio", el estado siempre se puede interpretar con claridad y la información no proviene de la magnitud del valor, sino del cambio de estado.

Este enfoque contrasta marcadamente con el de los sensores analógicos, donde la señal es continua y requiere interpretación numérica. Comprender la diferencia entre ambos es fundamental para el aprendizaje de los sistemas de automatización.

El botón pulsador es solo una de las muchas entradas digitales. Durante el proyecto, o como desarrollo posterior del mismo, los siguientes dispositivos también funcionan con un principio similar:

- Relé de láminas (contacto magnético): cierra o abre un circuito bajo la influencia de un campo magnético; su aplicación típica es la detección de puertas y ventanas.
- Interruptor de límite: se utiliza para detectar la posición límite del movimiento mecánico.
- Salida digital del sensor de movimiento PIR: emite una señal ALTA cuando detecta movimiento; de lo contrario, permanece en estado BAJO.
- Retroalimentación de contacto de relé digital: una forma común de retroalimentación en sistemas industriales.

La característica común de estos dispositivos es que proporcionan información sobre el estado, no una cantidad medida.

La importancia educativa del uso de entradas digitales, especialmente de botones, radica en que desarrolla varios patrones de pensamiento fundamentales:

- pensamiento orientado a eventos (algo sucede → se produce una reacción),
- el concepto básico de la interfaz hombre-máquina,
- la abstracción, durante la cual la acción física se transforma en información digital,
- comprensión de las lógicas de control y de las máquinas de estados posteriores.

En este punto, los estudiantes comprenden en primer lugar que un sistema automatizado no "monitorea constantemente", sino que responde a los eventos, lo cual es un principio fundamental de los controles industriales.

Salida digital: LED como actuador controlado

El LED (diodo emisor de luz) es uno de los dispositivos de salida digital más simples del kit Smart Farm, que proporciona una representación visual de las decisiones del controlador. Funciona en dos estados:

- Salida BAJA → LED apagado,
- Salida ALTA → LED encendido.

El pin de salida digital del controlador (ESP32) controla el LED, generalmente a través de una resistencia limitadora de corriente, que protege el componente contra sobrecorrientes. Aquí, los estudiantes se enfrentan por primera vez al hecho práctico de que el controlador no puede controlar directamente ninguna carga, sino que debe diseñarse teniendo en cuenta el entorno electrónico.

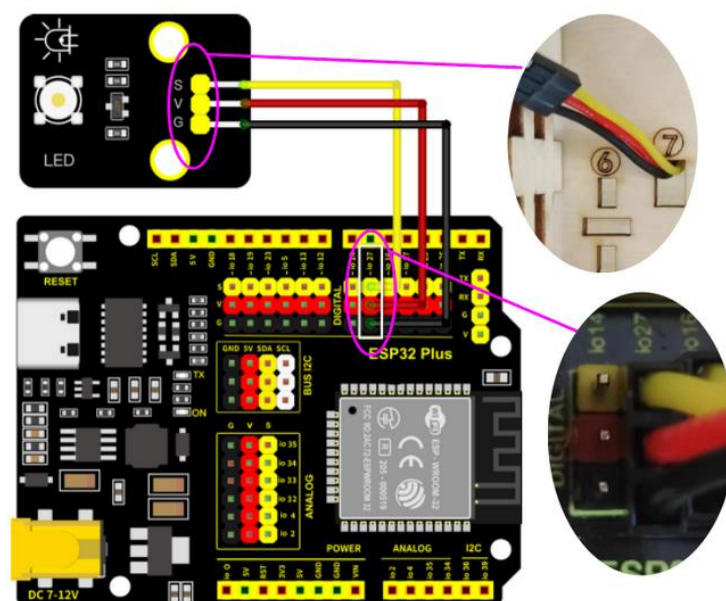


Figura 21: Diagrama de cableado del LED para salida digital (GPIO del ESP32 + resistencia limitadora de corriente) (Fuente: <https://docs.keyestudio.com>)

El concepto de salida digital es fundamentalmente diferente al de entrada. Mientras que las entradas proporcionan información sobre el sistema, las salidas representan una intervención activa, ya que el sistema no solo detecta sino que también afecta a su entorno.

En esta etapa, el LED no se utiliza para iluminar, sino como indicador de estado. Los estudiantes comprenden que el estado del LED representa la lógica interna del programa, ya que la salida siempre es el resultado de un proceso de decisión. La retroalimentación física desempeña un papel fundamental en la comprensión y la depuración.

Al controlar el LED, los estudiantes experimentan por primera vez toda la cadena de control:

1. evento de entrada (por ejemplo, presionar un botón),
2. procesamiento / decisión (verificación de condición, verificación de estado),
3. Intervención de salida (encender o apagar el LED).

Esta cadena sirve como modelo abstracto para sistemas posteriores más complejos (control de motores, relés, automatización).

Desde una perspectiva educativa, el LED como salida digital prepara la comprensión de los siguientes dispositivos:

- módulos de relé (conmutación de dispositivos de mayor potencia),
- zumbadores/soneras,
- semáforos (rojo-amarillo-verde),
- Líneas de control para pantallas digitales.

Los estudiantes comprenderán así que el LED es solo un modelo simplificado, detrás del cual subyace el mismo principio de control que en el caso de los actuadores industriales. Su aplicación se desarrolla a continuación:

- pensamiento de causa y efecto,
- el enfoque de sistemas (entrada → procesamiento → salida),
- interpretación de la retroalimentación visual,
- la capacidad de abstracción, que posteriormente nos permite prescindir de los dispositivos físicos y comprender los sistemas simulados.

En este punto, los estudiantes comprenden que un sistema automatizado "no piensa", sino que reacciona según reglas predefinidas, y que el resultado en cada caso es consecuencia del estado lógico del programa.

Control de iluminación: conexión de entrada y salida

El siguiente paso del proyecto consiste en conectar el pulsador y el LED mediante una lógica de control sencilla pero completa. Esta tarea es la primera en presentar el modelo de entrada-procesamiento-salida como un sistema unificado.

El modelo lógico de control de iluminación se compone de los siguientes elementos:

- evento: cambio de estado del botón pulsador,
- Decisión: comprobar el estado actual del LED,
- Intervención: encender o apagar el LED.

Esta estructura describe ya un ciclo de automatización completo, aunque la implementación sea técnicamente sencilla.

Los estudiantes deben tener en cuenta que la tarea en sí no requiere un microcontrolador. Un pulsador y un LED pueden conectarse como un circuito cableado simple, donde el pulsador cierra directamente el circuito y el LED responde inmediatamente al interruptor físico. Sin embargo, esto no es automatización, sino operación directa del hardware.

El objetivo de utilizar el microcontrolador no es "resolver" la tarea, sino modelar toda la cadena de efectos: digitalizar un evento físico, tomar decisiones lógicas mediante software y ejecutar intervenciones programadas. Esta distinción es fundamental para que los estudiantes comprendan el sistema.

Elementos de programación (en un entorno basado en bloques)

Durante la programación basada en bloques, aparecen los siguientes elementos:

- Bloques de activación de eventos (pulsación de botón o cambio de estado),
- bloques de control de salida digital (LED encendido/apagado),
- Variable de estado opcional que almacena el estado actual del LED.

La introducción de la variable no es una limitación técnica, sino una decisión de diseño consciente que separa el evento (lo que sucedió) del estado (cuál es el estado actual del sistema).

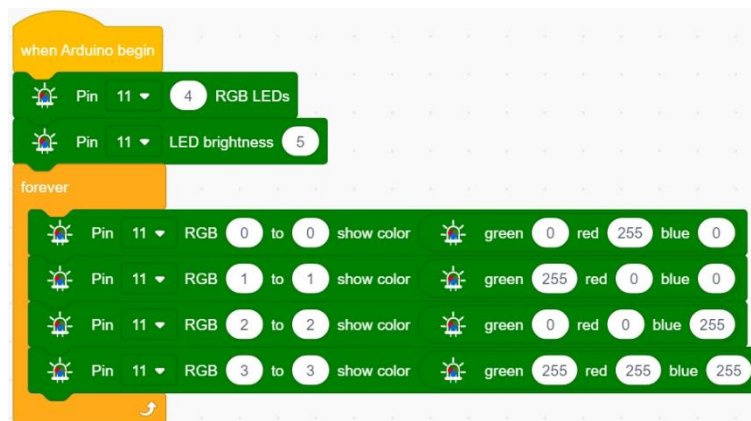


Figura 22: Estructura completa de un programa de bloque de control de iluminación, con uso de variables (Fuente: <https://docs.keyestudio.com>)

Durante la tarea, los estudiantes se dan cuenta rápidamente de que el controlador no "recuerda" automáticamente los estados anteriores, sino que preservar el estado es una responsabilidad de programación, lo que significa que el evento y el estado son conceptos que deben tratarse por separado.

Esta experiencia constituye una preparación directa para la automatización multiestado, los controles temporizados y los sistemas asíncronos en red.

Reflexión educativa

A pesar de su simplicidad, el proyecto de control de iluminación tiene un valor educativo excepcional. Los estudiantes experimentan que los sistemas digitales no piensan, sino que ejecutan, lo que significa que el funcionamiento del sistema depende enteramente de la lógica diseñada; por lo tanto, incluso una tarea aparentemente trivial se basa en decisiones de ingeniería conscientes.

Comparar una solución cableada con una implementación basada en un controlador ayuda a los estudiantes a comprender que la automatización no se trata de las herramientas, sino de la mentalidad.

Esta parte del proyecto crea una base sólida para comprender los sistemas de control posteriores basados en sensores, en condiciones y en red, y proporciona una transición natural hacia soluciones de IoT más complejas.

3.16. Proyecto 2 – Sistema de control de iluminación

Este proyecto introduce por primera vez la medición y el procesamiento de magnitudes analógicas, lo que supone una diferencia fundamental con respecto a las tareas anteriores basadas únicamente en lógica digital. Aquí, los estudiantes ya no trabajan con señales de dos estados, sino que interpretan y transforman magnitudes físicas en constante cambio en decisiones de control.

Funcionamiento del sensor de luz (LDR) – entrada analógica

Un LDR (resistor dependiente de la luz) es un resistor basado en semiconductores cuyo valor de resistencia varía en función de la intensidad de la luz que incide sobre él:

- Iluminación intensa → menor resistencia,
- Poca luz → mayor resistencia.

Este cambio es continuo, por lo que es de naturaleza analógica. El ESP32 no "ve" el brillo directamente, sino que mide el voltaje que aparece en el circuito divisor de voltaje formado con la fotorresistencia en su entrada analógica.

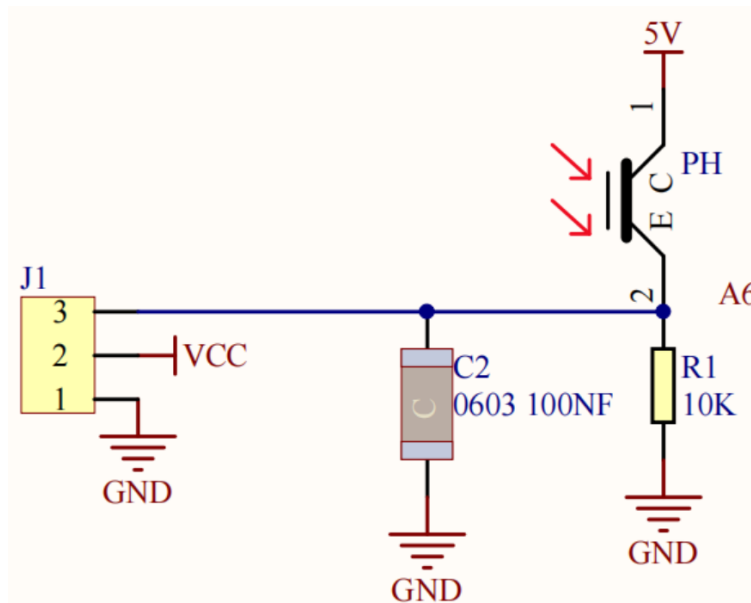


Figura 23: Característica LDR y diagrama de conexión del divisor de voltaje (entrada analógica ESP32) (Fuente: <https://docs.keyestudio.com>)

La entrada analógica:

- no da un valor ALTO o BAJO,
- pero un resultado de medición dentro de un rango numérico (por ejemplo, 0-4095),
- que es proporcional a la intensidad de la luz.

Los estudiantes que participan en el proyecto se enfrentan al hecho de que el valor de la medición no es absoluto, sino que depende del contexto, la precisión del sensor y de la electrónica es limitada, y la interpretación de los datos medidos siempre requiere contexto.

Proceso de decisión e intervención: procesamiento analógico La

estructura lógica más simple de un sistema controlado por luz:

1. Medición de la intensidad de la luz (entrada analógica),
2. comparación con un valor límite,
3. Realizar una intervención (encender o apagar la iluminación).

Este modelo sigue generando una decisión digital basada en una entrada analógica.



Figura 24: Estructura del bloque de control condicional para entrada analógica (Fuente: <https://docs.keystudio.com>)

La importancia educativa de la transición analógica y digital

Este paso hace hincapié en la digitalización de la señal analógica, el problema de elegir el valor límite y el manejo de datos de medición ruidosos y fluctuantes.

Los estudiantes comprobarán que un límite demasiado bajo o demasiado alto puede provocar un funcionamiento inestable, por lo que puede ser necesario el uso de histéresis o filtrado, demostrando así que las decisiones de ingeniería afectan al comportamiento del sistema.

Salida analógica: atenuación basada en PWM

En la siguiente fase del proyecto, la intervención ya no es binaria, sino continua. El brillo del LED no solo se enciende y se apaga, sino que se regula proporcionalmente.

La modulación por ancho de pulso (PWM) es una técnica que nos permite lograr un efecto analógico con una salida digital:

- La salida se enciende y se apaga rápidamente.
- La proporción de tiempo de funcionamiento (ciclo de trabajo) determina el rendimiento promedio,
- El brillo del LED es proporcional al factor de llenado.

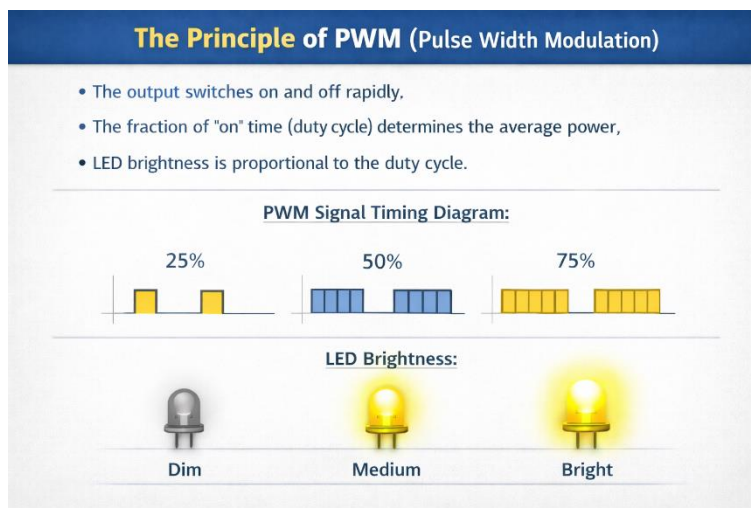


Figura 25: Diagrama de temporización de la señal PWM y cambio de brillo del LED (Fuente: Generado por IA)

En este proyecto, los estudiantes crearán una relación proporcional:

- Entrada: intensidad de luz medida (valor analógico),
- procesamiento: escalado, transformación,
- Salida: El brillo del LED se controla mediante una señal PWM.

Este enfoque ya refleja una perspectiva de control de procesos y es un precursor directo de las regulaciones industriales.

Reflexión educativa

El proyecto del sistema de control por luz desempeña un papel fundamental en el desarrollo del pensamiento de los estudiantes:

- distinguir entre cantidades digitales y analógicas,
- comprender que el mundo real es continuo y los controladores son discretos,
- reconocer el principio de regulación proporcional,
- Consideran que un funcionamiento "mejor" requiere más planificación.

Esta pieza constituye un puente natural hacia los sistemas posteriores de control de temperatura, control de motores y optimización de energía, y representa un paso significativo para profundizar en el pensamiento de ingeniería.

3.17. Proyecto 3 - Detección de distancia e intervención automática (Sistema de alimentación inteligente)

El Sistema de Alimentación Inteligente es un proyecto integrado en el que los estudiantes combinan un sensor de tiempo y un actuador mecánico para crear un sistema de automatización funcional. Simula una situación sencilla pero realista: al detectar la proximidad de un objeto, se ejecuta una acción mecánica.

Esto abarca por completo la cadena operativa básica de los sistemas automatizados:



La finalidad funcional del Sistema de Alimentación Inteligente es que el sistema implementa el funcionamiento de un alimentador automático que detecta la aproximación de un objeto (por ejemplo, una mano), toma una decisión en función de la distancia medida y realiza un movimiento mecánico (por ejemplo, abrir/cerrar una tapa) mediante un servomotor.

Su funcionamiento es sencillo de comprender, pero al mismo tiempo requiere un pensamiento técnico y de ingeniería complejo.

Sensor de distancia ultrasónico

Su función es determinar la distancia a un objeto mediante la medición del tiempo. El funcionamiento del sensor se basa en el control digital (disparo/eco) y se calcula el valor de la distancia.

Importancia educativa:

- aprender sobre un nuevo principio de medición,
- separación de los datos de medición brutos y la información interpretada.

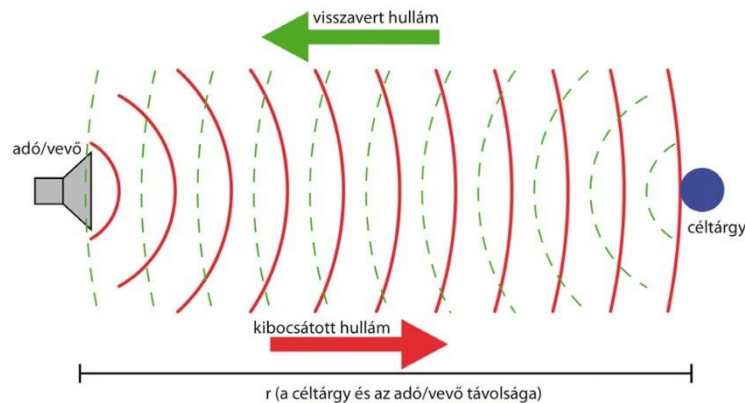


Figura 26: Diagrama esquemático de la medición de distancia por ultrasonidos (Fuente: <https://docs.keyestudio.com>)

Servomotor como actuador mecánico

El servomotor es la decisión de control físico para la implementación respuesta.

Se puede controlar mediante una señal PWM, ajustar a posiciones angulares específicas y proporciona un movimiento preciso y repetible.

Importancia educativa:

- conectando la electrónica y la mecánica,
- experimentar el efecto físico directo del control.

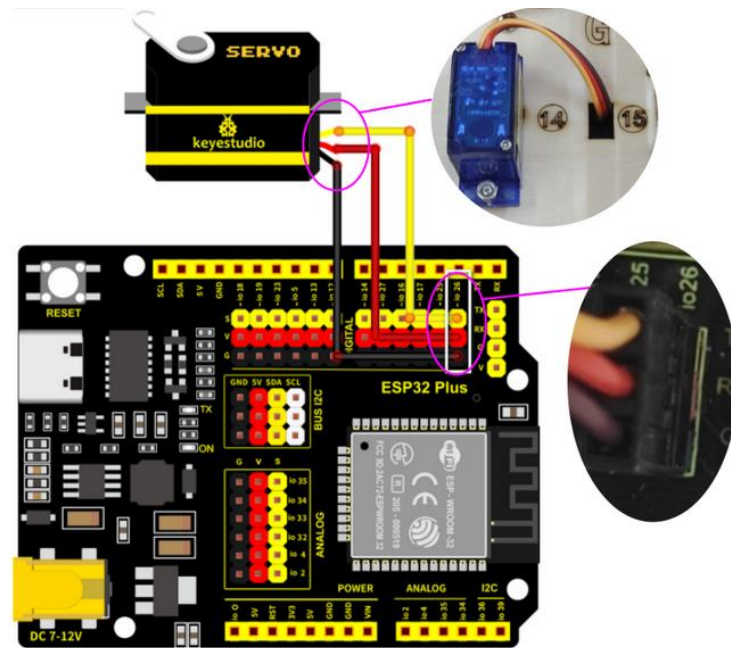


Figura 27: Estructura y principio de control del servomotor (Fuente: <https://docs.keyestudio.com>)

Principio de medición de distancia por

ultrasonidos El proceso de medición ultrasónica:

1. El sensor emite un pulso sonoro
2. El sonido se refleja en el objeto
3. El controlador mide el tiempo transcurrido
4. La distancia se puede determinar mediante cálculo.

No se trata de una medición de voltaje analógica, sino de una medición basada en el tiempo a partir de señales digitales, lo que requiere un paso computacional.

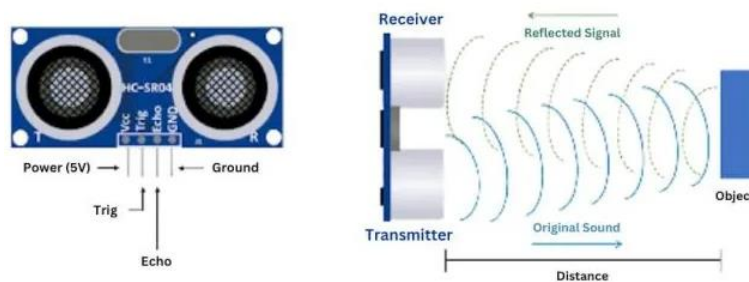


Figura 28: Proceso de medición ultrasónica (Fuente: <https://docs.keyestudio.com>)

Para facilitar el procesamiento de los datos de los sensores y la lógica de decisión, la distancia medida constituye la base para la decisión de control. El proceso lógico es el siguiente:

1. medir distancia
2. comparación con el valor límite
3. tomar una decisión
4. iniciar una intervención

Esta estructura ya describe un proceso completamente automatizado.



Figura 29: Procesamiento de datos de distancia por bloques (Fuente: <https://docs.keyestudio.com>)

Control de servomotor

El servomotor se controla mediante una señal PWM, donde el ancho del pulso determina la posición angular y el control no regula la potencia, sino la posición.

Esto supone una diferencia significativa en comparación con la atenuación PWM de los LED.



Figura 30: Relación entre la señal PWM y la posición del servomotor (Fuente: <https://docs.keyestudio.com>)

Sistema de alimentación inteligente: funcionamiento integrado

El funcionamiento del Sistema de Alimentación Inteligente constituye un todo lógico unificado: si la distancia medida se encuentra dentro de un valor umbral, el sistema toma una decisión, el servomotor se mueve y se produce una operación mecánica.



Figura 31: Programa de bloques completo del sistema de alimentación inteligente (Fuente: <https://docs.keystudio.com>)

Reflexión educativa

El proyecto del Sistema de Alimentación Inteligente es especialmente adecuado para que los estudiantes:

- ver el funcionamiento del control a nivel de sistema,
- comprender las particularidades de la medición basada en el tiempo,
- reconocer la relación entre la lógica de decisión y el efecto mecánico,
- Desarrollar un pensamiento de ingeniería consciente.

El proyecto demuestra claramente que el funcionamiento de los sistemas automatizados no reside en los elementos individuales, sino en su interconexión y en la lógica diseñada.

3.18. Proyecto 4 – Sistema de control de temperatura

El sistema de control de temperatura modela una tarea de control clásica, fundamental en la automatización industrial y los sistemas de gestión de edificios. El proyecto se centra en la medición continua de una magnitud ambiental, su visualización y el control automático de un dispositivo de intervención.

Toda la cadena operativa del proyecto:



Esta estructura es fácil de entender para los estudiantes, a la vez que refleja la lógica regulatoria industrial real.

Sensor de temperatura y humedad (sensor DHT)

El sensor DHT es un sensor ambiental digital que mide la temperatura y la humedad relativa y transmite los datos al controlador en forma de un paquete de datos digitales.

Características electrónicas:

- se comunica a través de una única línea de datos,
- El resultado de la medición no es un voltaje analógico, sino un valor digital procesado.
- El sensor utiliza un mecanismo interno de temporización y control.

Importancia educativa:

- separa la medición analógica de la comunicación de datos digitales,
- señala que algunos de los sensores ya proporcionan datos "procesados",
- Introduce el concepto de muestreo y tiempo de actualización.

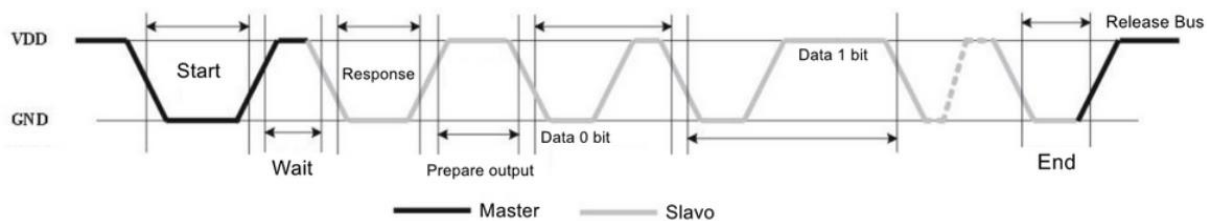


Figura 32: Principio de funcionamiento del sensor DHT y proceso de transmisión de datos (Fuente: <https://docs.keyestudio.com>)

Pantalla LCD 1602: visualización de datos de medición.

Un elemento importante del sistema de control de temperatura es la información que proporciona al usuario. Esta función la cumple la pantalla LCD 1602, que muestra los datos medidos en formato numérico y permite la monitorización continua del estado actual del sistema.

Función:

- Visualización de temperatura y humedad,
- Información sobre el estado de funcionamiento (por ejemplo, ventilador activo/inactivo),
- mostrar información de diagnóstico.

Importancia educativa:

- enfatiza la separación entre medición y visualización,
- demuestra que el funcionamiento del sistema no es "invisible",
- Permite la depuración y la comprensión del sistema.

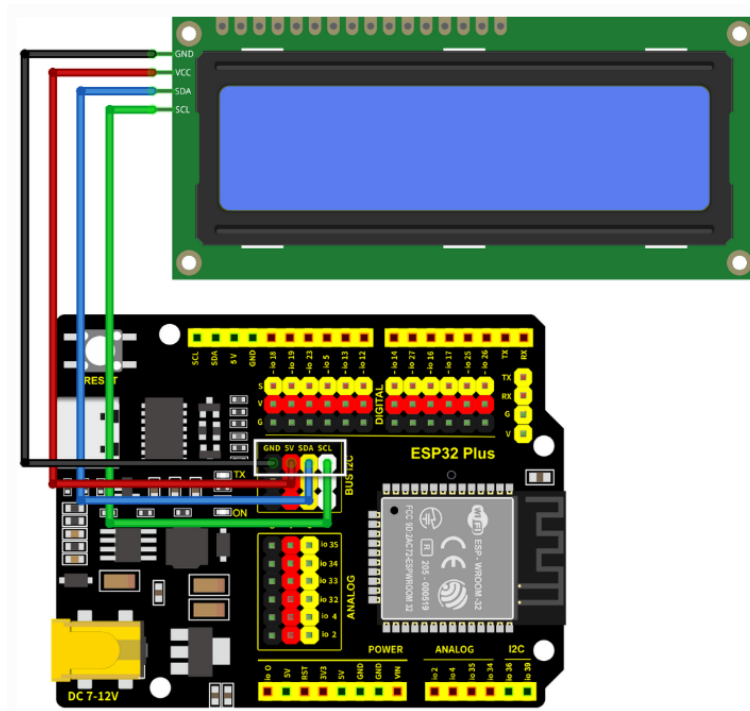


Figura 33: Diagrama de cableado de la pantalla LCD 1602 (Fuente: <https://docs.keyestudio.com>)

El ventilador como elemento de intervención

El ventilador es un actuador activo del sistema, que se utiliza para modificar las condiciones ambientales. El controlador activa el ventilador directamente o mediante un controlador a través de una salida digital.

Lógica de control:

- Si la temperatura medida supera un valor límite establecido, el ventilador se enciende.
- Si la temperatura desciende por debajo del valor límite, el ventilador se apaga.

Importancia educativa:

- demuestra un control simple de dos estados,
- comprender las ventajas y limitaciones del control basado en límites,
- Prepara el terreno para la posterior comprensión de la histéresis y el control PID.

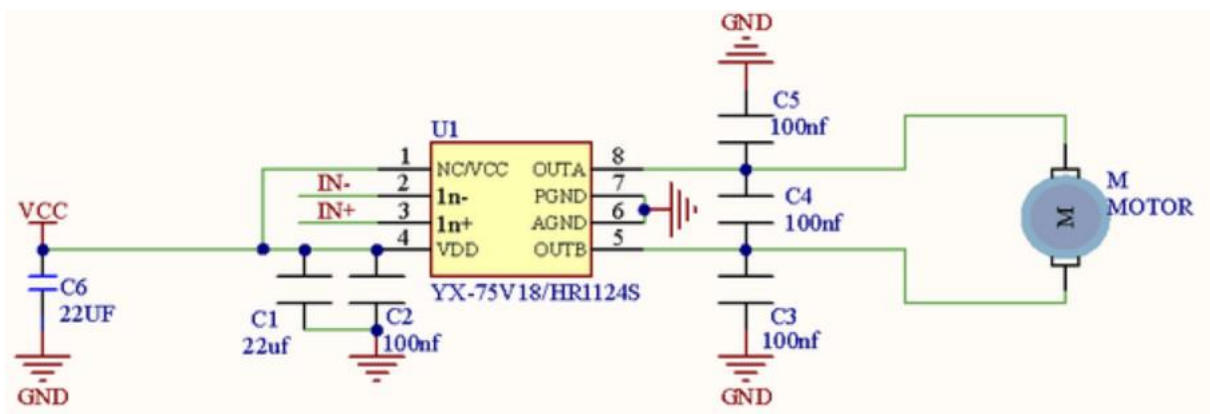


Figura 34: Diagrama de cableado del control del ventilador (Fuente: <https://docs.keyestudio.com>)



Figura 35: Diagrama lógico del bloque de control del ventilador (Fuente: <https://docs.keyestudio.com>)

Funcionamiento integrado del sistema de control de temperatura.

Todo el proyecto del Sistema de Control de Temperatura funciona según la siguiente lógica:

1. El sensor DHT realiza una medición
2. Los datos se procesan
3. El valor medido se muestra en la pantalla LCD.
4. El controlador compara el valor con el valor límite.
5. El estado del fan cambia en consecuencia

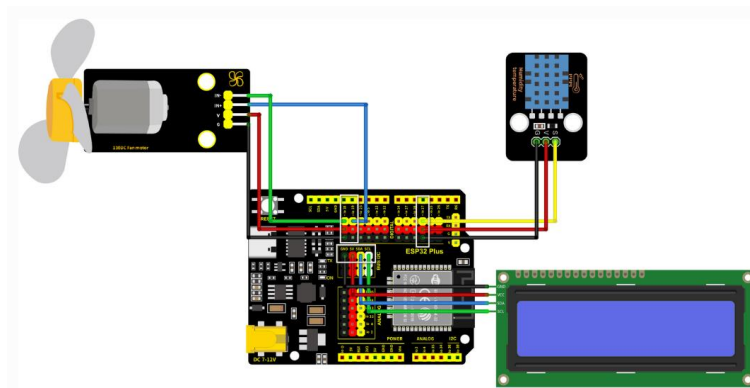


Figura 36: Diagrama de cableado del sistema de control de temperatura (Fuente: <https://docs.keyestudio.com>)

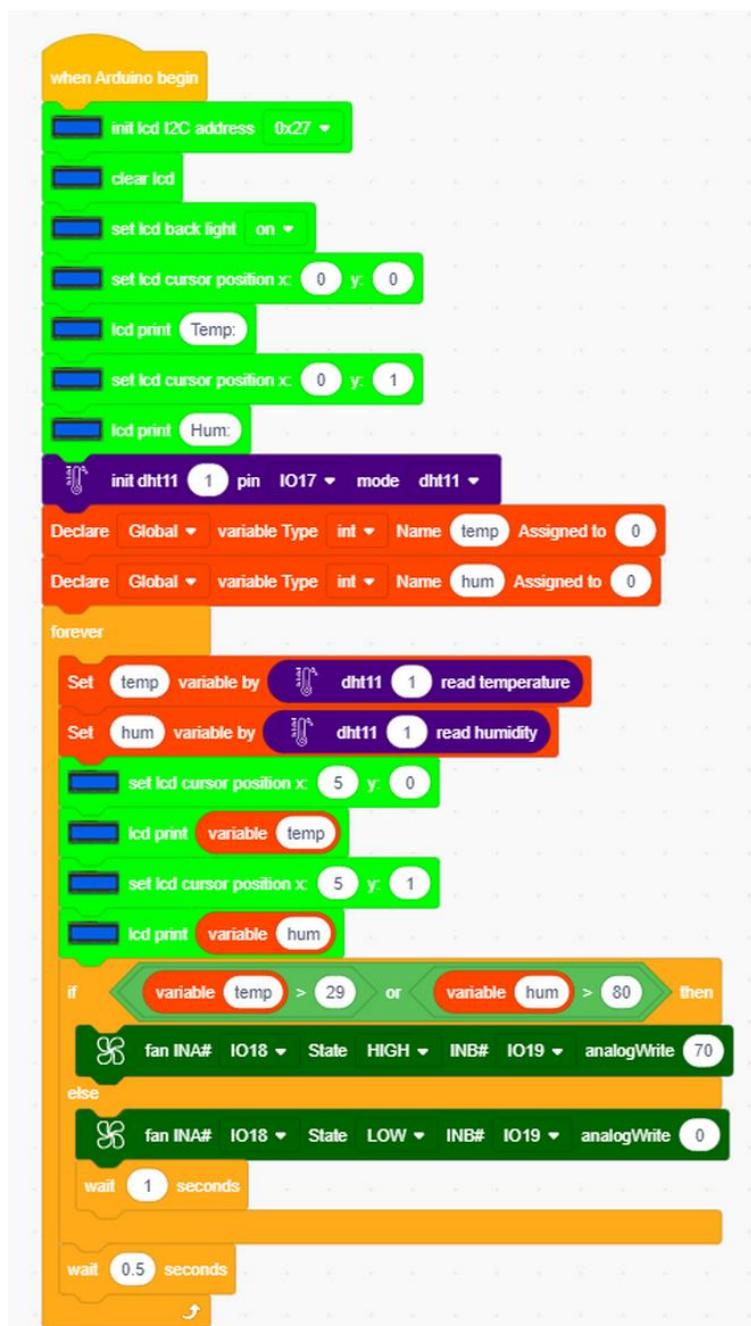


Figura 37: Programa de bloques completo de un sistema de control de temperatura (Fuente: <https://docs.keyestudio.com>)

Esta estructura ya constituye un modelo simplificado de un sistema regulatorio real.

Reflexión educativa

El proyecto del Sistema de Control de Temperatura ofrece a los estudiantes la oportunidad de:

- comprender el papel de los datos ambientales en la automatización,
- distinguir entre medición, visualización e intervención,
- reconocer el funcionamiento y las limitaciones de la regulación basada en límites,
- Piensa sistemáticamente en una tarea aparentemente sencilla.

Esto encaja perfectamente con el concepto de hogar inteligente y granja inteligente, y crea una base sólida para comprender posteriormente soluciones de control y de red más complejas.

3.19. Proyecto 5 – Sistema de riego automático

El proyecto del Sistema de Autorriego pone de manifiesto, en primer lugar, el problema de ingeniería que supone la insuficiencia de las capacidades eléctricas de la unidad de control para accionar directamente el actuador. El objetivo es crear un modelo de sistema de riego automatizado que establezca una interfaz segura, de estilo industrial, entre la toma de decisiones y la ejecución física.

Cadena operativa del proyecto:



Esta estructura es fundamental para todos los sistemas de automatización industrial reales.

Limitaciones de las salidas del microcontrolador

El controlador del kit Smart Farm es un sistema basado en un microcontrolador cuyas salidas digitales funcionan a niveles de voltaje bajos (normalmente 3,3 V o 5 V CC) y solo pueden suministrar corrientes pequeñas (del orden de unas pocas decenas de mA).

Esto es suficiente para controlar LEDs, circuitos electrónicos más pequeños o señales lógicas.

Sin embargo, no es suficiente para accionar directamente motores, bombas, válvulas o dispositivos conectados a la red eléctrica (por ejemplo, 230 V CA).

Desde una perspectiva educativa, este es un punto clave: los estudiantes reconocen que control y rendimiento no son el mismo concepto.

Niveles de voltaje y diferencias de potencia

El riego automático es un buen ejemplo de sistemas donde coexisten múltiples niveles de voltaje:

- lado de control: bajo voltaje (5 V CC),
- Lado del actuador: mayor voltaje o potencia (por ejemplo, 12 V CC, 24 V CC, hasta 230 V CA).

Durante el proyecto, se hace hincapié en por qué un motor no puede conectarse directamente a la salida del microcontrolador, por qué es necesario un elemento de adaptación de potencia y cómo separar de forma segura ambos mundos.

Módulo de relé como elemento de adaptación de potencia

El elemento central del sistema de riego automático es el módulo de relés, que permite que una señal de baja potencia procedente del microcontrolador controle un circuito de mayor potencia.

Basándose en el principio de funcionamiento del relé, el controlador conmuta una bobina, el campo magnético de la bobina cierra o abre mecánicamente los contactos, y estos operan en un circuito separado.

Esto implementa el aislamiento galvánico, que protege la electrónica de control, aumenta la fiabilidad operativa del sistema y cumple con los principios de la automatización industrial.

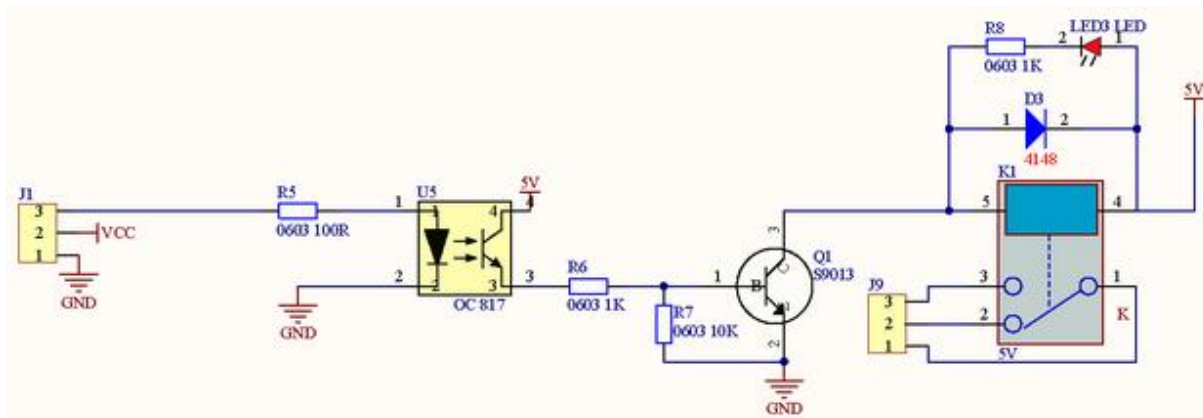


Figura 38: Contactos del módulo de relé y lógica de funcionamiento (Fuente: <https://docs.keyestudio.com>)

Paralelo industrial – Soluciones de salida PLC

El uso de relés no es una "solución de aficionado", sino un principio industrial. Las salidas de los controladores lógicos programables (PLC) también suelen ser:

- salidas de relé,
- salidas de transistor (PNP/NPN),
- Con menos frecuencia se utilizan salidas de triac para corriente alterna.

Ventajas de las salidas de relé del PLC:

- amplio rango de voltaje,
- Conmutación de cargas de CA y CC,
- aislamiento eléctrico.

En este punto, el proyecto Smart Farm refleja una mentalidad industrial, incluso si la herramienta utilizada tiene fines educativos.

Lógica de control del sistema de riego automático

El funcionamiento del sistema es sencillo, pero logra una automatización completa:

1. El controlador evalúa las condiciones de entrada
2. los compara con un valor límite,
3. decide sobre la necesidad de riego
4. Activa el relé mediante salida digital
5. El relé activa la unidad de riego

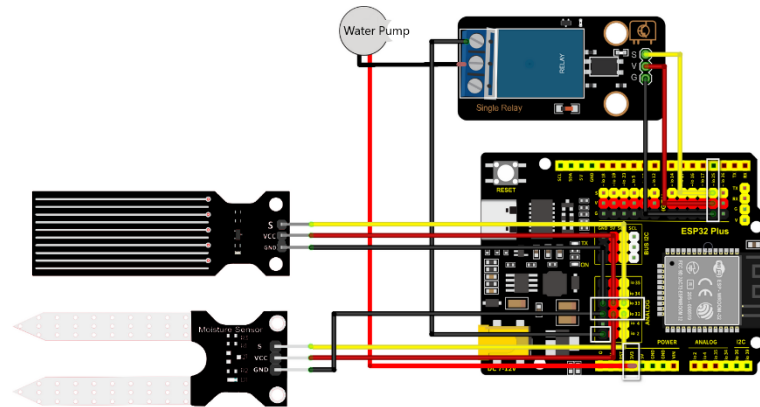


Figura 39: Conexión del sistema de riego automático (Fuente: <https://docs.keyestudio.com>)

The code is written in a Scratch-style block-based language for an Arduino. It starts with a 'when Arduino begin' block. The initialization sequence includes: 'init lcd I2C address' (0x27), 'clear lcd', and 'set lcd back light' (on). Two global variables are declared: 'waterlevel' and 'soilhum', both of type 'int' and assigned the value '0'. A 'forever' loop contains the main logic: 'Set waterlevel' (variable) by 'read the value of water level' (IO33) and 'Set soilhum' (variable) by 'read the value of soil moisture' (IO32). The LCD is cleared, and the cursor is moved to (0,0) to print 'WaterLevel:' and (0,1) to print 'SoilHum:'. The cursor is then moved to (11,0) to print the 'waterlevel' variable and (11,1) to print the 'soilhum' variable. Two conditional blocks follow: an 'if' block where '700 > waterlevel' triggers a sequence of three tones on IO16 (NOTE_E5, NOTE_A5, NOTE_B5) with a duration of 131, followed by 'NoTone' on IO16; another 'if' block where '1200 > soilhum' triggers a sequence of three tones on IO16 (NOTE_D3, NOTE_F3, NOTE_A3) with a duration of 131, followed by 'NoTone' on IO16. A combined 'if' block where 'soilhum < 1200 and waterlevel > 700' triggers a relay pin IO25 to output HIGH, followed by a 0.4-second wait, then output LOW, followed by a 0.7-second wait. The loop concludes with a 0.5-second wait before repeating.

Figura 40: Control basado en bloques del sistema de riego automático (Fuente: <https://docs.keystudio.com>)

Esta estructura puede relacionarse directamente con los ciclos de control de procesos industriales.

Lecciones educativas y de ingeniería

Durante el proyecto de riego automático, los estudiantes:

- comprender las limitaciones físicas de los microcontroladores,
- reconocer la necesidad de alineación del desempeño,
- encontrar el concepto de aislamiento galvánico,
- Utilizan soluciones análogas a los principios de la automatización industrial.

El proyecto destaca especialmente por su capacidad para concienciar sobre las consecuencias físicas de las decisiones de control, preparar al usuario para el pensamiento basado en PLC y sentar las bases para la comprensión de sistemas posteriores que están conectados en red y utilizan la comunicación industrial.

Este capítulo constituye una transición natural entre el mundo de la automatización IoT sencilla y los sistemas de control industrial profesionales.

3.20. Proyecto 6 - Sistema de granja inteligente controlado por WiFi

El proyecto de Granja Inteligente controlada por Wi-Fi representa la transición de la automatización de operación local a sistemas conectados en red y monitoreados remotamente. En esta etapa, los estudiantes ya no solo controlan sensores y actuadores, sino que implementan un modelo de monitoreo completo basado en IoT.

La idea central del proyecto es que el sistema:

- se conecta a la red,
- proporciona datos a un cliente remoto,
- Se puede controlar mediante una interfaz web HMI.

Esta funcionalidad va más allá de los sistemas integrados simples y te acerca a soluciones reales para dispositivos industriales e inteligentes.

El papel de la conexión WiFi en el sistema IoT

El controlador ESP32 del kit Smart Farm incorpora un módulo WiFi que permite la conexión directa a una red inalámbrica. En este proyecto, la conexión WiFi no es solo una mejora técnica, sino un cambio de perspectiva fundamental: el control y la monitorización abandonan el entorno cableado local y se integran en un contexto de red.

Al conectarse a la red, el sistema no funciona como un controlador aislado, sino que se puede acceder a los datos de los sensores de forma remota a través de un navegador, ya que las intervenciones no están relacionadas con la entrada física directa, sino con las solicitudes de red.

Desde una perspectiva educativa, aquí es donde el concepto de Internet de las Cosas (IoT) se vuelve tangible: el dispositivo no existe en sí mismo, sino que es un nodo activo en un sistema en red.

Al establecer una conexión WiFi, los estudiantes se familiarizarán con conceptos básicos de redes, como el de punto de acceso. El ESP32 se conecta a una red WiFi existente (por ejemplo, el router de una escuela o de una casa) que actúa como punto de acceso, de modo que comprenden que el controlador se conecta como cliente y la red proporciona el medio de comunicación.

Un ESP32 conectado a la red recibe una dirección IP única que identifica el dispositivo en la red, permite el acceso desde un navegador y constituye la base para la comunicación posterior con el sistema de gestión de clientes.

Esta es la primera vez que los estudiantes experimentan que un dispositivo físico tiene una "dirección" en la red, similar a la de otros ordenadores o sistemas de vigilancia.

Durante el proyecto, queda claro que el ESP32, el ordenador o dispositivo móvil del usuario se encuentran en la misma red, lo que significa que la comunicación solo es posible si estos dispositivos se "detectan" entre sí a nivel de red. El control y la recuperación de datos se realizan mediante mensajes de red.

Importancia educativa

Durante la introducción de la conexión WiFi, los estudiantes:

- **Dominan el pensamiento básico de redes,**
- comprender el papel del acceso basado en IP en los sistemas modernos,
- **reconocer la diferencia entre control local y control de red,**
- experimentar una de las condiciones básicas de funcionamiento de los sistemas IoT.

Este conocimiento te prepara directamente para:

- comprender la interfaz de administración web,
- el procesamiento de arquitecturas de sistemas de supervisión de clientes subsiguientes,
- y la adopción de un enfoque de red para los sistemas industriales y distribuidos.

Funcionamiento del sistema de monitorización web integrado

En la solución Smart Farm de fábrica, la interfaz de administración web se implementa mediante un sistema de administración web que se ejecuta directamente en el ESP32. Esto significa que el propio controlador proporciona el sitio web, por lo que no hay ningún sistema de administración externo ni ordenador en el sistema, sino que el cliente (navegador) se conecta directamente al ESP32.

Esta solución es extremadamente intuitiva desde una perspectiva educativa porque aparece integrada en una sola herramienta:

- recopilación de datos,
- el procesamiento,
- **la pantalla,**
- el control.

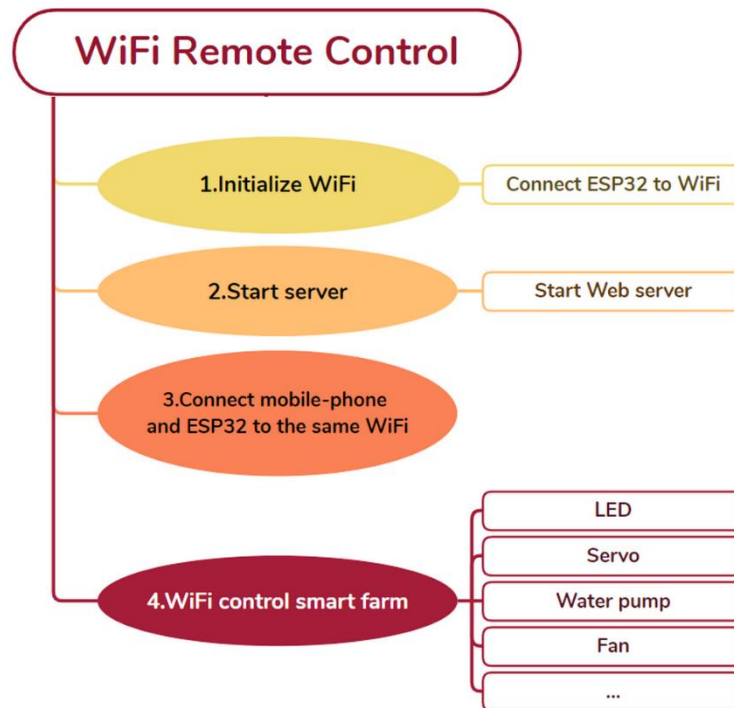


Figura 41: Modelo de comunicación de un sistema de monitorización web que se ejecuta en ESP32 (Fuente: <https://docs.keystudio.com>)

La página de administración web es la "ventana" del sistema al usuario. Según el tutorial de fábrica, la interfaz generalmente implementa las siguientes funciones:

- mostrar datos actuales del sensor,
- retroalimentación del estado de salida,
- Proporcionando controles sencillos (botones, interruptores).

La interfaz no destaca por su complejidad gráfica, sino por su capacidad para reflejar el estado del sistema físico en tiempo real.

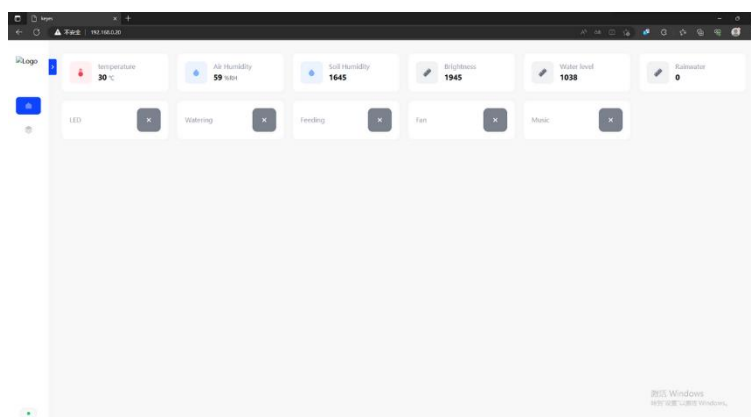


Figura 42: Aspecto de la página de administración web de Smart Farm (Fuente: <https://docs.keystudio.com>)

Un elemento educativo importante es reconocer que un sistema de monitorización web no funciona con recursos ilimitados. En el caso de ESP32, los estudiantes se enfrentan a los siguientes factores:

- memoria limitada,
- espacio de almacenamiento limitado,
- gestionar un número limitado de conexiones simultáneas,

- utilizando estructuras HTML simples.

Esto da como resultado una funcionalidad del sitio web deliberadamente simple, por lo que la cantidad de datos mostrados es limitada, lo que significa que el ESP32 no es adecuado para ejecutar aplicaciones web complejas.

Esta constatación es una lección clave de ingeniería: todos los sistemas se construyen sobre la base de compromisos.

Los elementos de control que se muestran en la interfaz web HMI permiten al usuario enviar un comando al sistema, el controlador lo interpreta como una instrucción digital y los actuadores responden a la señal proveniente de la red.

Esta operación demuestra el principio básico de los sistemas IoT modernos, donde el usuario no está físicamente presente, pero el control es inmediato y se basa en la retroalimentación.

Durante el proyecto de Granja Inteligente controlada por WiFi, los estudiantes:

- comprender la esencia del control basado en redes,
- reconocer las limitaciones de recursos de los sistemas embebidos,
- experimentar los fundamentos del modelo de sistema de supervisión de clientes,
- **Separan las tareas de control y visualización.**

La fortaleza del proyecto radica en que:

- proporciona una experiencia IoT real con requisitos mínimos de dispositivos,
- señala claramente los límites de las soluciones de fábrica,
- Proporciona una preparación natural para comprender posteriormente arquitecturas industriales, externas y sistemáticas.

Este capítulo concluye el análisis de las posibilidades del sistema Smart Farm de la fábrica y sienta las bases para el desarrollo de su propia arquitectura de sistema, separada por recursos y siguiendo un patrón industrial.

3.21. Resumen de la interpretación pedagógica

La parte del proyecto de Granja Inteligente, basada en programación por bloques, no se presenta como un objetivo de aprendizaje independiente, sino como una transición pedagógica planificada conscientemente a lo largo del proyecto. En esta fase, los estudiantes adquieren conceptos básicos, relaciones y principios de funcionamiento que posteriormente serán esenciales para comprender y desarrollar sistemas más complejos de tipo industrial.

Para fomentar el pensamiento sistémico, los estudiantes trabajan de forma consistente con el mismo modelo básico a lo largo de todo el proyecto:



Esta cadena de influencia aparece en cada parte del proyecto:

- entradas digitales (por ejemplo, pulsador, interruptor),
- entradas analógicas (por ejemplo, LDR, potenciómetro),
- mediciones basadas en el tiempo (sensor ultrasónico),
- Salidas digitales y basadas en PWM (LED, ventilador, servomotor),
- conmutación de potencia por relés (sistema de riego),
- Comunicación de red y control web (WiFi).

Los estudiantes se dan cuenta gradualmente de que los elementos individuales no pueden interpretarse de forma aislada, sino que son partes funcionales de un sistema más amplio.

La conexión entre el mundo analógico y el digital

Un resultado pedagógico destacado es que los estudiantes pueden hacer una distinción clara entre:

- señales digitales (biestado, controladas por eventos),
- cantidades analógicas (valores continuos),
- el papel de la conversión analógica a digital,
- entre valores límite, control proporcional y modulación PWM.

El concepto de ruido de medición, la importancia de elegir el umbral de decisión y el enfoque de "medición ≠ información" aparecen durante el control de la luz, el control de la temperatura y el control de los ventiladores.

Debido a la comprensión de la intervención y la gestión de energía, al utilizar el módulo de relés y los actuadores motorizados, los estudiantes se enfrentan al hecho de que las salidas de los microcontroladores son capaces de suministrar corriente y potencia limitadas; por lo tanto, el control y la transmisión de potencia están separados, de modo que las señales lógicas de bajo voltaje (3,3–5 V CC) pueden controlar circuitos de mayor potencia (por ejemplo, 12 V CC, 230 V CA).

Esto crea un paralelismo directo con el funcionamiento de los controles industriales, donde los PLC controlan actuadores externos a través de salidas de relé, transistor o semiconductor.

Pensamiento en red y perspectiva del IoT

En este proyecto basado en WiFi, los estudiantes experimentarán:

- los fundamentos del modelo de sistema de supervisión de clientes,
- el funcionamiento del sistema de monitorización web integrado,
- el impacto de las limitaciones de recursos físicos (memoria, almacenamiento) en el diseño del sistema,
- las posibilidades y limitaciones de la monitorización y el control remotos.

Esta experiencia nos prepara para reconocer que, en un entorno industrial, las tareas suelen estar separadas (controlador de nodo de campo – monitorización central – servicios de back-end).

La programación basada en bloques ayuda a prepararse para un cambio en el enfoque de programación, que en este contexto:

- desacopla la lógica de control de las dificultades sintácticas,
- visualizar la gestión del estado, los eventos y las condiciones,
- Se puede adaptar fácilmente a paradigmas de programación industrial (por ejemplo, IEC 61131-3, bloques de funciones, máquinas de estados).

Los estudiantes no están "aprendiendo Scratch", sino aprendiendo a pensar de forma lógica y controlada, lo que posteriormente podrán aplicar en la programación de microcontroladores basada en C, PLC industriales, comunicación basada en Modbus y arquitecturas de sistemas distribuidos.

Resumen

Esta fase del proyecto crea una base sólida para el desarrollo de elementos de hardware de diseño propio, la transición a la programación basada en texto, la comprensión de arquitecturas distribuidas orientadas a la industria y el posterior desarrollo de sistemas con una estructura de cliente-nodo de campo y backend-frontend para la monitorización.

Por lo tanto, el entorno basado en bloques no es una estación terminal, sino una escalera de aprendizaje construida conscientemente que conduce gradualmente a los estudiantes hacia la resolución de problemas reales de ingeniería e industria.

3.22. Implementación práctica del capítulo

El montaje y la puesta en marcha del kit de granja inteligente fue la primera experiencia verdaderamente práctica para los estudiantes. Observar el funcionamiento del kit de fábrica les dio una rápida sensación de éxito, pero pronto quedó claro que detrás de las automatizaciones aparentemente sencillas se escondían complejas conexiones lógicas y electrónicas.

El procesamiento de la documentación de fábrica, especialmente la interpretación de las descripciones en idiomas extranjeros, supuso un gran desafío. Durante la traducción parcial al húngaro y la adaptación didáctica, los estudiantes participaron activamente en la interpretación del contenido, lo que profundizó significativamente su comprensión. La programación basada en bloques parecía sencilla al principio, pero la gestión de estados y el control basado en condiciones requerían un análisis más profundo.

4. Diseño digital de su propio modelo de hogar inteligente/granja inteligente (Fusion 360)

Este capítulo constituye un hito fundamental del proyecto: aquí, la lógica de sensores, control e IoT previamente aprendida se materializa en una estructura física concreta y tangible. En esta etapa, los estudiantes transitan del mundo de la electrónica y el software al campo del diseño de ingeniería, donde cada decisión tiene consecuencias geométricas, de ensamblaje y tecnológicas de fabricación.

El objetivo de este capítulo no es simplemente crear un modelo 3D, sino comprender que un modelo físico destinado a la impresión 3D es una estructura de soporte: debe proporcionar espacio para el controlador, los sensores, los actuadores y el cableado, a la vez que permite el montaje, las mediciones y las modificaciones posteriores. En este sentido, el modelo es una herramienta de aprendizaje, no un producto final.

La ruta de aprendizaje está estructurada deliberadamente en varios pasos. El diseño digital está precedido por una fase de concepto físico, que analiza el diseño de fábrica del Smart Farm Kit,

Se basa en la identificación de elementos funcionales y el reconocimiento de las limitaciones de la fabricación aditiva. A continuación, se aprende sobre el entorno de diseño Fusion 360 y se desarrolla un modelo digital paramétrico basado en componentes, hasta alcanzar un estado listo para la producción.

Durante el proceso, los estudiantes experimentan:

- Un buen diseño no empieza con el software,
- La forma es siempre consecuencia de la función,
- El pensamiento paramétrico permite la iteración y el rediseño,
- El modelo digital actúa como un “prototipo virtual” para ayudar a prevenir problemas de fabricación y montaje.

Esta parte, por lo tanto, constituye un puente entre la comprensión del sistema electrónico-informático y su implementación física, y proporciona una base sólida para las etapas posteriores del proyecto relacionadas con la impresión 3D y la integración del sistema.

4.1. Punto de partida: análisis del diseño físico del kit de granja inteligente.

El primer paso en el proceso de diseño digital no consiste en utilizar software CAD, sino en analizar detenidamente las soluciones físicas existentes. El Smart Farm Kit sirve como sistema de referencia en este proyecto: un punto de partida indispensable para crear un modelo propio, funcional y optimizado para fines educativos.

El diseño físico del Kit de Granja Inteligente se presenta no como un modelo para copiar, sino como un ejemplo para analizar. El objetivo de los estudiantes es explorar:

- qué elementos cumplen una función real,
- cómo los componentes electrónicos están conectados al espacio físico,
- Qué soluciones tienen fines educativos y cuáles se desarrollaron por razones tecnológicas de producción o estéticas.

Este enfoque establece desde una etapa temprana el principio fundamental del diseño orientado a la función: la forma es consecuencia de la función, no su punto de partida. Los estudiantes aprenden a analizar críticamente las soluciones prefabricadas y a distinguir los elementos necesarios de los detalles contextuales u opcionales.

La base para la conceptualización es conocer las dimensiones reales. Los estudiantes examinan los elementos de la fábrica con herramientas sencillas y accesibles para todos. Utilizando un calibrador, determinan las dimensiones circundantes, luego realizan bocetos proporcionales en papel cuadriculado y registran la ubicación de las aberturas, los puntos de montaje y la posición de los sensores.

Desde un punto de vista pedagógico, es importante que en esta fase no se elaboren dibujos técnicos estándar, ya que el objetivo principal no es la precisión formal, sino la comprensión de las relaciones espaciales y las proporciones.



Figura 43: Medición de los elementos físicos de una granja inteligente (Fuente: Generada por IA)

Este paso prepara directamente para el pensamiento paramétrico posterior, al tiempo que refuerza la conexión entre la realidad física y el diseño abstracto.

4.2. Limitaciones de la impresión 3D e implicaciones para el diseño

Al analizar componentes de fábrica, los estudiantes se enfrentarán a las limitaciones prácticas de la fabricación aditiva. Reconocerán que algunos detalles geométricos no se pueden reproducir con la impresión 3D de escritorio, ya que las paredes delgadas, las cavidades cerradas y los socavados representan un problema, por lo que se requiere una simplificación consciente en lugar de una fidelidad geométrica completa.

Esta constatación introduce el enfoque de diseño para la fabricación y subraya que las decisiones de diseño siempre están relacionadas con la tecnología de fabricación elegida.

El siguiente paso en el análisis físico es la toma de decisiones conscientes: qué conservar y qué omitir al desarrollar tu propio modelo.

Los alumnos determinan conjuntamente qué elementos son necesarios para colocar los sensores, actuadores y controladores, dónde proporcionar los puntos de fijación para ellos y qué partes se pueden omitir sin comprometer los objetivos de aprendizaje.

Las decisiones siempre se basan en la función y la capacidad de entrenamiento, no en la similitud formal con la solución de fábrica. El objetivo de la conceptualización es crear una estructura física que:

- no contiene espacios en blanco innecesarios,
- ahorra material y se imprime rápidamente,
- fácil de instalar y modificar,
- Proporciona un entorno educativo transparente.

Este paso tiene un impacto directo en el tiempo de impresión, el consumo de material y la velocidad de los ciclos de iteración.

4.3. Concepto físico como descripción del sistema

En esta etapa, los estudiantes ya no interpretan el modelo como una "casa" o una "caja", sino como un sistema de puntos para colocar sensores, una estructura que proporciona un rango de movimiento para los actuadores y un soporte para el control y el cableado.

Este enfoque ayuda a garantizar que el diseño no se vea ahogado en detalles formales demasiado pronto, sino que siga siendo interpretable a nivel sistémico.

Preparación conceptual para el diseño paramétrico

Si bien este capítulo aún no introduce el uso de Fusion 360, sí presenta la idea básica del diseño paramétrico. Las dimensiones no son números aislados, sino parámetros interrelacionados, donde una modificación puede afectar a múltiples elementos y donde el objetivo del diseño es la adaptabilidad y la capacidad de rediseño.

Esta mentalidad constituye una transición directa a la siguiente parte, donde el concepto físico se transforma en un modelo digital paramétrico.

Resumen pedagógico

Esta fase preparatoria y de conceptualización garantiza que el diseño posterior basado en Fusion 360 no sea un mero ejercicio de uso de software, sino una representación digital de decisiones de ingeniería conscientes.

Los estudiantes comprobarán que un buen diseño comienza antes de encender el ordenador, que la simplificación y la abstracción son virtudes de la ingeniería y que los mundos físico y digital forman una unidad estrecha.

4.4. Entorno de diseño Fusion 360: conceptos básicos y enfoque.

En esta etapa, los estudiantes ya no hablan de diseño digital a nivel teórico, sino que experimentan a través de operaciones concretas cómo un concepto físico se convierte en un modelo digital modificable.

Los estudiantes aprenderán los conceptos básicos de Fusion 360 y su uso mediante la publicación oficial de Autodesk, "Autodesk Fusion 360 Training: The Future of Making Things Attendee Guide".

Basándose en el material educativo procesado, el proceso de aprendizaje no se basa en el aprendizaje de listas de funciones, sino en el desarrollo de rutinas de diseño básicas, por ejemplo:

- ¿Cómo lanzamos un nuevo modelo?
- cómo navegamos en el espacio 3D,
- Cómo crear geometría simple pero estable.

El objetivo pedagógico es que los estudiantes reconozcan que el diseño CAD es una actividad, no la creación de una imagen visual.

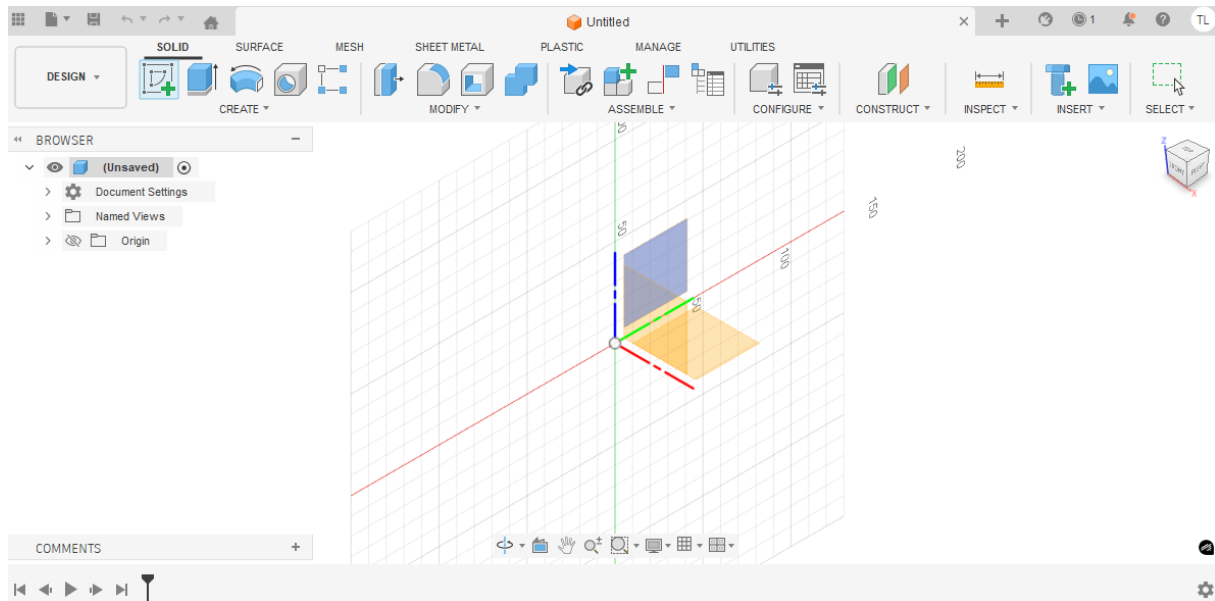


Figura 44: Entorno de diseño Fusion 360: conceptos básicos y enfoque (Fuente: Edición propia)

Fusion 360 se presenta en la formación como una «superficie de pensamiento». El documento destaca que, incluso a nivel principiante, se puede completar un ciclo de diseño completo: crear el modelo, modificarlo, revertir a estados anteriores y observar el impacto de los cambios. Este enfoque introduce a los usuarios al mundo del modelado paramétrico y el diseño narrativo, donde el modelo no es un objeto estático, sino el resultado de un proceso.

Actividades específicas para los estudiantes:

- Crear un nuevo diseño
- Guardar en el proyecto (comprensión del funcionamiento en la nube)
- Regresar a un estado anterior en la línea de tiempo.

Términos clave del documento:

- Modelado paramétrico
- Diseño basado en la historia
- Edición no destructiva

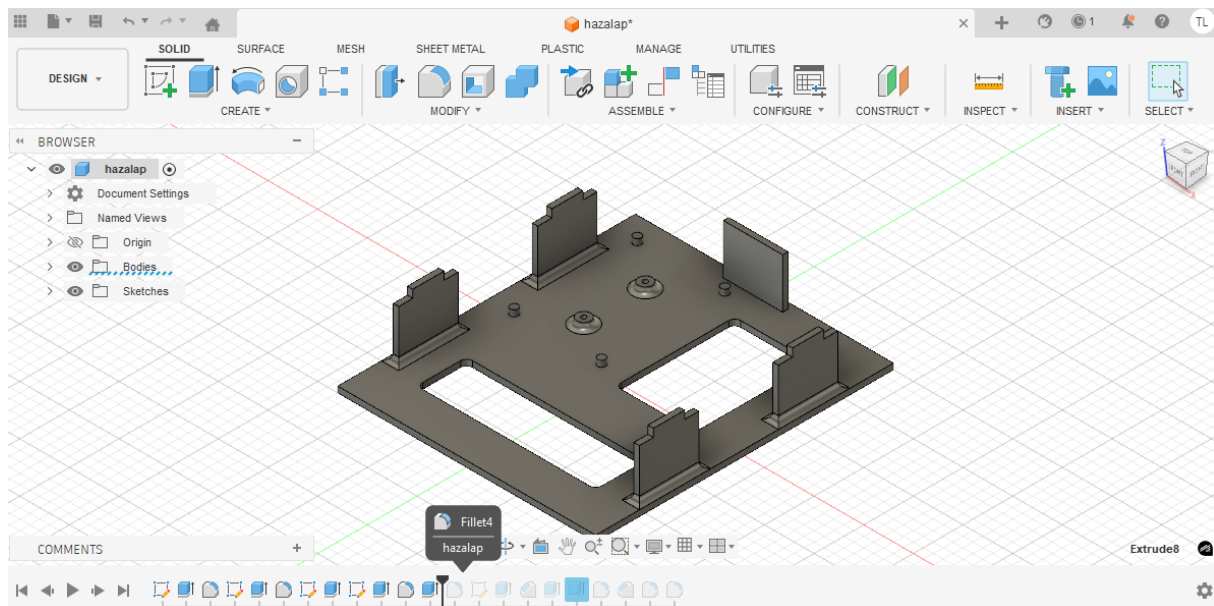


Figura 45: Creación de un nuevo diseño: base para hogar inteligente (Fuente: Edición propia)

Enfoque pedagógico: los estudiantes experimentan que la historia del modelo es tan importante como su estado actual.

El material de formación de Autodesk Fusion 360 presenta la interfaz específicamente para principiantes, basándose en la lógica de uso en lugar de en las funciones.

4.5. Lienzo: dominando el movimiento espacial

Se otorga especial importancia al desarrollo del pensamiento espacial. Los estudiantes utilizan activamente operaciones de manipulación de vistas (rotación, zoom, desplazamiento), así como vistas básicas, para desenvolverse con seguridad en el espacio. Esta habilidad es un requisito previo para la posterior creación de modelos precisos y bien elaborados.

Ejercicios para estudiantes:

- rotación del modelo (órbita),
- Acercar/alejar,
- Cacerola,
- utilizando las vistas básicas (superior, frontal, derecha).

Objetivo pedagógico: desarrollar el pensamiento espacial incluso antes de dibujar.

Navegador: el "mapa interno" del modelo.

El "mapa interno" del sistema, el Navegador, desempeña un papel importante en la comprensión de la estructura del modelo. Los estudiantes aprenden a gestionar bocetos, cuerpos y componentes, y se dan cuenta de que un modelo bien estructurado no solo es más transparente, sino también más fácil de modificar posteriormente.

Actividades estudiantiles:

- Ocultar/mostrar boceto,
- Separación de cuerpos,

- elementos de nomenclatura.

Términos clave:

- Carpeta de bocetos
- Carpeta de cuerpos
- Componentes

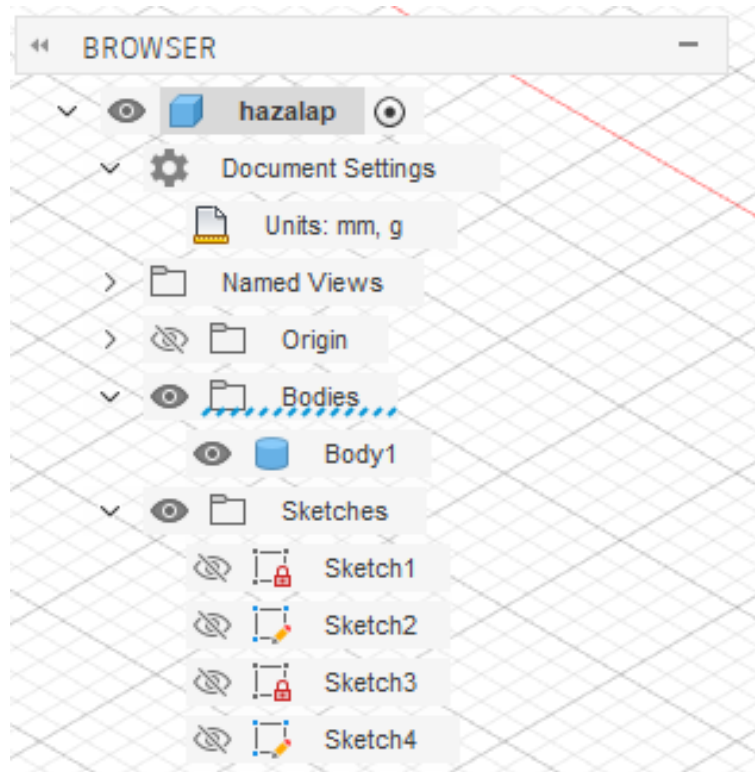


Figura 46: "Mapa interno" del modelo (Fuente: Edición propia)

Paralelamente, el uso de la línea de tiempo hace visible el sistema causal del proceso de diseño: modificar un paso anterior afecta a todo el modelo, lo que ayuda a desarrollar el pensamiento sistémico.

El documento hace un uso extensivo de la línea de tiempo como herramienta de aprendizaje.

Práctica del estudiante:

- modificar un paso de extrusión,
- observando las consecuencias en todo el modelo.

Palabra clave: Historia del diseño



Figura 47: Uso de la línea de tiempo (Fuente: Edición propia)

4.6. Conceptos geométricos y de diseño básicos, de forma tangible.

Boceto: no un dibujo, sino un conjunto de reglas.

Es importante comprender el concepto de boceto, que no es un dibujo, sino un conjunto de reglas. Los estudiantes experimentan que un boceto no es un simple dibujo, sino un sistema definido matemáticamente. El uso de dimensiones y restricciones geométricas garantiza que el modelo sea estable y claro. Los problemas de los bocetos incompletos se hacen evidentes rápidamente, por lo que los estudiantes comprenden la importancia de un diseño preciso a partir de su propia experiencia.

Según la formación de Autodesk Fusion 360, el error más común que cometen los diseñadores principiantes es un boceto indefinido.

Práctica del estudiante:

- dibujar un rectángulo,
- **agregando tamaño,**
- aplicación de restricciones.

Términos clave:

- Boceto totalmente restringido
- Grados de libertad
- Restricciones geométricas

Reflexión pedagógica: lo que no se define causará problemas más adelante.

Cuerpo y componente: ¿cuándo usar cada uno?

La distinción entre sólidos y componentes también se revela como una decisión importante en ingeniería. Los estudiantes comprenden que, si bien un sólido es adecuado para un diseño rápido, el uso de componentes requiere un pensamiento estructural, especialmente para modelos más complejos. Esta distinción también los prepara para tareas de ensamblaje posteriores.

El documento muestra a través de ejemplos prácticos:

- Cuerpo = moldeado rápido,
- Componente = pensamiento estructural.

Situación de decisión del estudiante:

- cuando un cuerpo es suficiente,
- cuando se justifica un componente separado.

Concepto clave: Crear componente a partir del cuerpo.

Los parámetros como herramientas de diseño

Durante la introducción al diseño paramétrico, los estudiantes experimentan que modificar una sola dimensión puede afectar a todo el modelo. Esto representa no solo un cambio técnico, sino también un cambio de perspectiva: el modelo deja de ser una forma fija para convertirse en un sistema modificable que se adapta a las necesidades del diseño.

Basándose en la formación de Autodesk Fusion 360, los alumnos intentarán lo siguiente:

- cambiar de tamaño,
- la transformación de todo el modelo.

Términos clave:

- Parámetros
- Cuadro de diálogo para cambiar parámetros

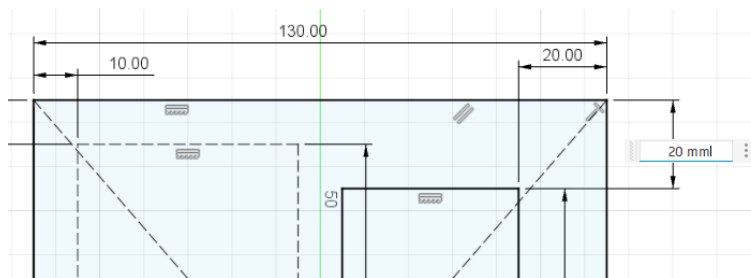


Figura 48: Uso de parámetros (Fuente: Edición propia)

Modelado corporal: operaciones básicas con mano firme.

Las operaciones básicas de modelado sólido, como la extrusión, el redondeo o la repetición, no se presentan de forma aislada, sino que siempre se vinculan a situaciones de diseño específicas. De esta manera, los estudiantes no aprenden comandos, sino que comprenden su función y sus consecuencias para el modelo en su conjunto.

Operaciones específicas y enseñables basadas en la formación de Autodesk Fusion 360:

- Extruir (Unir/Cortar),
- Filete y chaflán,
- Patrón (repetición lineal).

Términos clave:

- Unir vs. Cortar
- Filete de borde
- Patrón rectangular

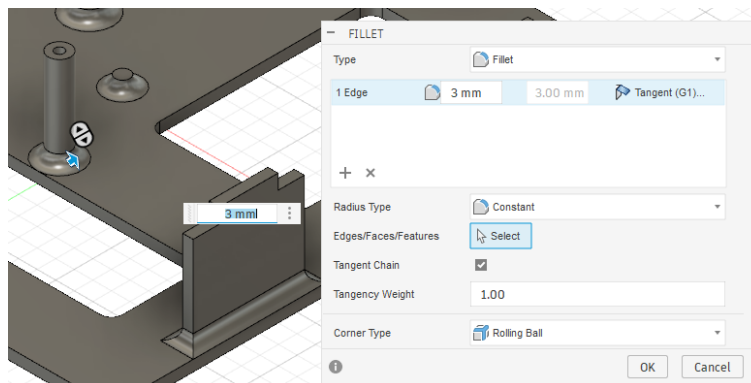


Figura 49: Modelado del cuerpo (Fuente: Edición propia)

4.7. Preparación para la impresión 3D: enfoque práctico final.

Las páginas 62 a 65 del manual de formación de Autodesk Fusion 360 se centran específicamente en el estado de "modelo terminado".

En la etapa final del proceso de diseño, la atención se centra en la viabilidad física del modelo digital. Los estudiantes aprenden que un modelo bien diseñado no necesariamente es apto para la fabricación automatizada, por lo que se debe prestar especial atención a los requisitos de la impresión 3D.

Al revisar el modelo, la consideración principal es asegurar una geometría cerrada, ya que la impresión solo es posible con volúmenes claramente definidos. Además, se deben examinar el grosor de las paredes, el tamaño de los detalles y la estabilidad geométrica. Los estudiantes experimentarán cómo las decisiones de diseño impactan directamente en la calidad y el éxito de la fabricación.

La exportación del modelo introduce los formatos necesarios para la fabricación (como STL o 3MF) y el proceso de creación de un archivo de datos imprimible a partir del modelo CAD. Este paso cierra el ciclo de diseño y, además, proporciona información: los errores y deficiencias de diseño se hacen visibles de inmediato durante la creación del modelo físico.

Lista de verificación del estudiante:

- cuerpo cerrado (estanco),
- espesor de pared adecuado,
- Exportar en formato STL/3MF.

Términos clave:

- Vista previa de la malla
- Exportación para la fabricación

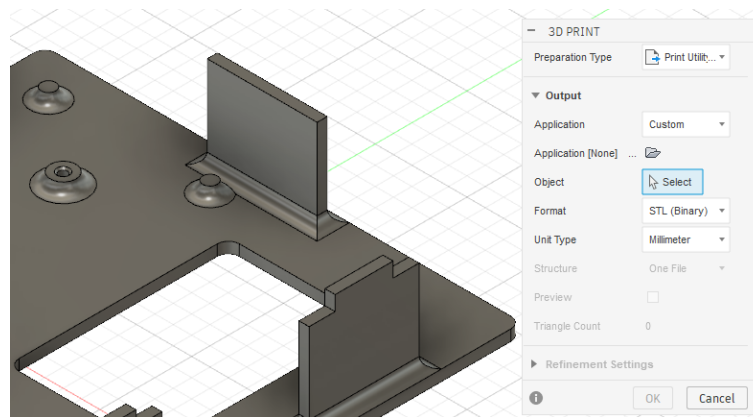


Figura 50: Preparación para la impresión 3D (Fuente: Edición propia)

Resumen didáctico

Esta etapa es particularmente importante desde un punto de vista didáctico, ya que deja claro a los estudiantes que el diseño digital no es una actividad en sí misma, sino la preparación de sistemas que se implementarán en el mundo físico real.

Al finalizar el proceso, los estudiantes no solo crearán un modelo, sino que también comprenderán el ciclo completo de diseño: desde la idea inicial hasta el modelo paramétrico y la forma fabricada. Esta experiencia sienta las bases del enfoque de ingeniería en el que el diseño y la implementación constituyen una unidad inseparable.

En esta parte de la implementación del proyecto, los estudiantes no aprenden a usar un programa de software, sino que comienzan a diseñar, a construir su primer modelo digital paso a paso y a comprender que el CAD es la representación externa del pensamiento.

Esto crea una base sólida para la siguiente fase, donde el modelo digital se convierte en un objeto físico real mediante la impresión 3D.

4.8. Resultados del proceso de diseño digital: presentación e interpretación de los modelos de los estudiantes.

Esta subsección presenta los modelos digitales creados durante el proyecto. Estos elementos no son muestras prefabricadas, sino modelos 3D que los estudiantes crearon basándose en principios de diseño aprendidos previamente y con la guía del profesor.

Durante la creación de los modelos, los estudiantes aplicaron conscientemente el principio del diseño orientado a la función, el pensamiento paramétrico, los aspectos de modularidad y ensamblaje, y las limitaciones geométricas de la fabricación aditiva (impresión 3D).

Desde una perspectiva pedagógica, estos modelos son huellas tangibles del proceso de aprendizaje: muestran cómo los principios de diseño abstractos se manifiestan en formas físicas concretas.

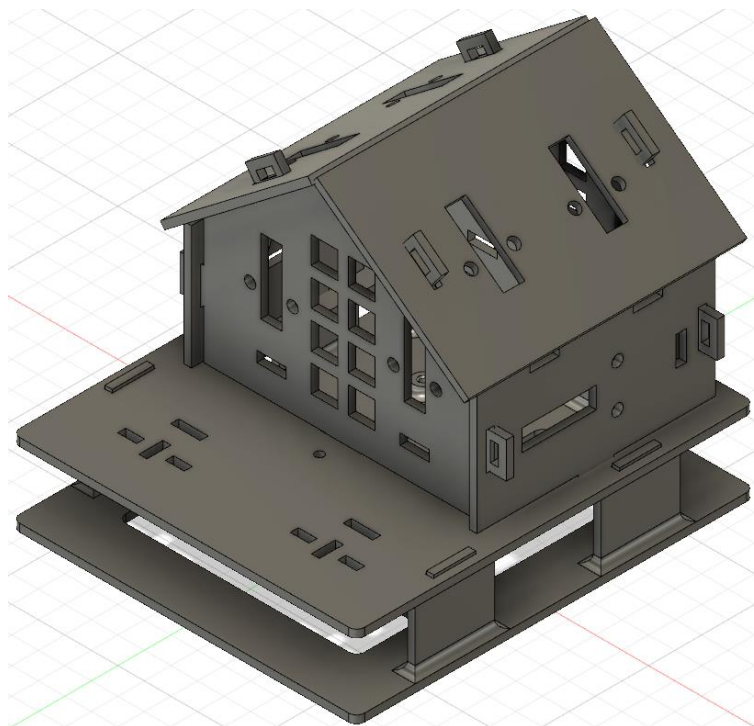


Figura 51: Modelo de ensamblaje de una casa inteligente (Fuente: Edición propia)

En conjunto, los archivos STL creados por los estudiantes forman la base física de un modelo modular de hogar inteligente/granja inteligente. El objetivo del diseño no era crear un objeto estéticamente final, sino una estructura de soporte que:

- adecuado para alojar componentes electrónicos,
- admite la grabación de sensores y actuadores,
- permite el ensamblaje, la medición y la modificación,
- Cumple con los requisitos de la impresión 3D de escritorio.

De este modo, los estudiantes comprendieron que el modelo digital no es un objetivo independiente, sino un marco físico para proyectos de electrónica y control.

Placa base y elementos estructurales inferiores: decisiones de diseño de los estudiantes

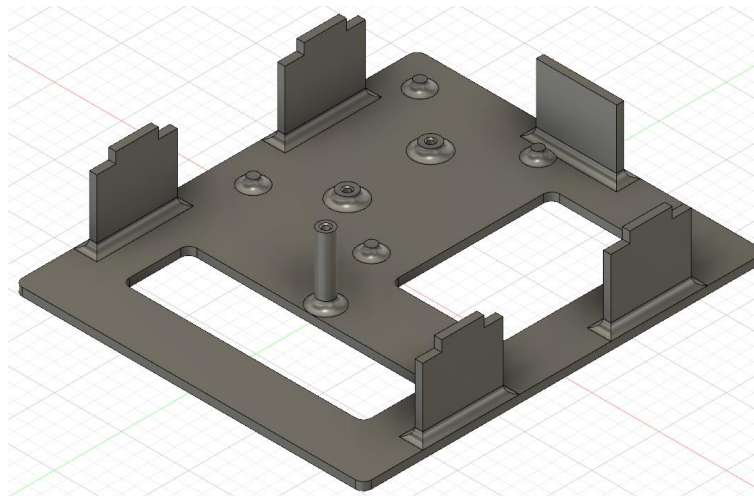


Figura 52: Placa base y elementos estructurales inferiores (Fuente: Edición propia)

Al diseñar la placa base y la tapa inferior, los estudiantes aplicaron los aspectos de ingeniería que habían aprendido previamente.

Decisiones tomadas en el proceso de diseño:

- creando una superficie de apoyo estable y plana,
- preparación para el montaje de módulos electrónicos,
- garantizar un tendido de cables ordenado.

Importancia didáctica: los estudiantes experimentaron que la calidad de la estructura básica determina la usabilidad de todo el sistema, y que incluso los elementos aparentemente "simples" tienen un diseño de ingeniería consciente detrás.

Paneles laterales: función y accesibilidad

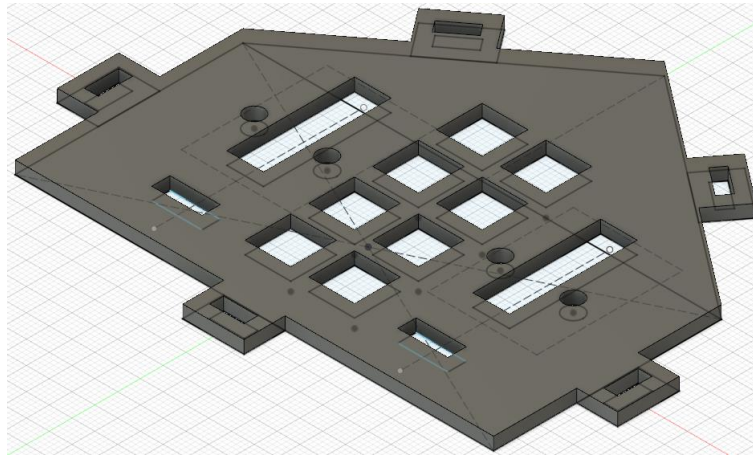


Figura 53: Paneles laterales (Fuente: Edición propia)

Al diseñar los paneles laterales, los estudiantes se propusieron conscientemente garantizar que la maqueta no fuera un recinto cerrado, sino una estructura "legible" que apoyara los objetivos educativos.

Entre las consideraciones de diseño se incluían garantizar la ubicación de los sensores y las pantallas, el acceso para el montaje y la medición, y la transparencia visual del interior.

Lección pedagógica: los alumnos reconocieron que la portada del modelo educativo no oculta, sino que hace que el funcionamiento sea comprensible.

Elementos de cubierta: equilibrio entre protección y desmontaje.

Al diseñar los elementos del techo, los estudiantes utilizaron soluciones que protegen los componentes electrónicos sin dificultar el desmontaje y permitiendo modificaciones posteriores.

Esta tarea de diseño puso de manifiesto que "cierre" no significa finalidad, sino más bien un compromiso consciente entre protección y accesibilidad.

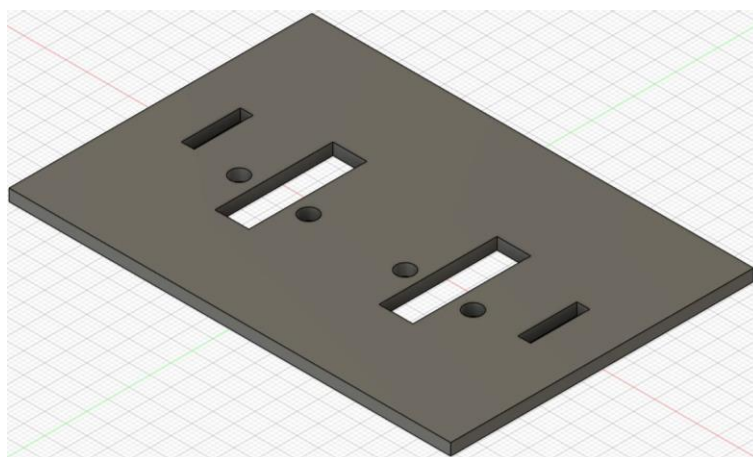


Figura 54: Elemento de techo (Fuente: Edición propia)

4.9. Resumen de los resultados del aprendizaje basados en los modelos completados

Los modelos creados durante el proyecto brindan a los estudiantes la oportunidad de reflexionar sobre sus propias decisiones de diseño y reconocer sus consecuencias.

Conceptos clave plasmados en los modelos:

- diseño paramétrico,
- modularidad,
- instalación,
- diseño para la fabricación,
- desarrollo iterativo.

Esta sección deja claro que los estudiantes no solo han aprendido a usar el software, sino también un enfoque de ingeniería, que se completará en los siguientes capítulos con la impresión 3D y el ensamblaje físico.

4.10. Implementación práctica del capítulo

Fue en esta etapa cuando quedó muy claro lo difícil que les resultaba a los estudiantes aceptar que el diseño digital se basa en una comprensión precisa de la realidad física. Al principio, el análisis de medidas, bocetos proporcionales y relaciones espaciales les parecía lento y "menos espectacular", pero más tarde se dieron cuenta de que las imprecisiones en el modelo y en la impresión se reflejaban inmediatamente.

Al introducir Autodesk Fusion, a menudo era necesario ralentizar el proceso deliberadamente. Los estudiantes querían dibujar mientras aprendían el pensamiento paramétrico. El punto de inflexión se produjo cuando comprendieron, por experiencia propia, que un boceto mal estructurado causa problemas graves posteriormente, mientras que una estructura bien pensada da como resultado un modelo estable y fácilmente modificable.

Al diseñar nuestro propio modelo, se tomaron decisiones de ingeniería reales: qué simplificar, dónde facilitar el acceso, qué se consideraba una función real. El hacinamiento era frecuente y debía ser controlado. Desde la perspectiva del profesor, uno de los resultados más importantes fue el cambio en la forma de pensar de los alumnos: comenzaron a ver el modelo no como un objeto, sino como un sistema funcional.

5. Experiencia en impresión y fabricación 3D

Un elemento clave del proyecto fue la implementación física de los procesos de diseño digital. El uso de la impresión 3D brindó a los estudiantes la oportunidad de experimentar directamente la relación entre el modelado por computadora y la fabricación en el mundo real, y de comprender la importancia de los principios del diseño para la producción. La impresión 3D, popular hoy en día y con un abanico de aplicaciones en constante crecimiento, resultó ser, en muchos sentidos, la opción ideal para la implementación física de esta fase del proyecto.

Durante la fase de producción, quedó claro que las soluciones que parecían correctas en el ámbito digital a menudo requerían modificaciones durante la implementación física. Esta experiencia fue particularmente valiosa desde una perspectiva educativa, ya que los estudiantes abordaron los problemas, su análisis y solución no como tareas aisladas, sino como parte de un proceso de desarrollo coherente.

5.1. Introducción a la impresora 3D y Bambu Studio

5.2. Breve introducción al proceso utilizado

El proceso de fabricación aditiva utilizado en el proyecto fue la tecnología FDM (Modelado por Deposición Fundida), uno de los métodos de impresión 3D más extendidos y accesibles en la actualidad. El proceso consiste en que la impresora funde un filamento de plástico continuo y lo deposita capa a capa siguiendo trayectorias predeterminadas.

La tecnología FDM resulta especialmente idónea para entornos educativos, ya que permite seguir fácilmente las etapas del proceso de fabricación y observar de inmediato los efectos de los cambios en los parámetros de impresión. Esto brinda a los estudiantes la oportunidad no solo de observar el resultado final, sino también de comprender e interpretar todo el proceso de fabricación.

5.3. Estructura y funcionamiento del hardware

Los trabajos de impresión se envían a un **Bambu Lab X1 Carbon**. Implementamos el proyecto con una impresora 3D de escritorio cerrada. El dispositivo cuenta con un sistema de sensores avanzado, nivelación automática de la mesa y un sistema de movimiento de alta precisión, que en conjunto garantizan un funcionamiento estable y fiable. Elegimos esta impresora durante la implementación del proyecto porque buscábamos, entre las alternativas disponibles en el mercado, un dispositivo que no solo produjera objetos de alta calidad, con numerosas opciones de calibración y personalización, sino que también lo hiciera de manera eficiente en cuanto al tiempo. El uso de la impresora requirió muchos menos procesos de calibración en segundo plano que otros dispositivos, lo que nos permitió dedicar más tiempo a áreas más importantes del proyecto. También seleccionamos una impresora Prusa i3 MK3S+ como alternativa, pero finalmente nos decidimos por la X1 Carbon debido a su manejo más sencillo y su diseño profesional.



Figura 55: Impresora 3D Bambu Lab X1 Carbon (Fuente: <https://www.3djake.hu/bambu-lab/x1c>)

El espacio de impresión cerrado desempeñó un papel fundamental en el proyecto, ya que redujo la aparición de errores de impresión al minimizar las influencias ambientales (temperatura, circulación del aire). Esto permitió a los estudiantes centrarse en el contexto tecnológico y evitar que los problemas derivados de la inestabilidad ambiental dominaran su experiencia.

5.4. El sistema de alimentación de filamentos AMS



Figura 56: Dispensador de filamento AMS (Fuente: <https://www.3djake.hu/bambu-lab/x1c>)

El sistema de alimentación automática de filamento AMS (Sistema Automático de Material) conectado a la impresora es capaz de manejar varios rollos de material simultáneamente. Si bien el proyecto se centraba principalmente en la producción de piezas de un solo color y tipo de material, el uso del sistema AMS aportó valiosos conocimientos tecnológicos. Además, facilitó el proceso de producción, ya que no era necesario extraer e insertar el filamento en la impresora al reemplazar rollos vacíos o cambiar de color.

Los estudiantes aprendieron sobre la importancia del manejo de materiales y el estado de las materias primas, con especial atención al efecto de la humedad y la importancia del almacenamiento, destacando los sistemas de manejo de materiales pasivos y activos.

dispositivos de secado de filamentos. Además, el funcionamiento de AMS proporcionó un buen punto de partida para la posterior introducción de capacidades de fabricación multimaterial y multicolor.

5.5. Principios de corte

La conversión de modelos digitales en formas fabricables es **Estudio de bambú** Lo hicimos utilizando un software de segmentación. Bambu Studio es el software de código abierto del fabricante de la impresora (Bambu Lab), basado en el programa de segmentación PrusaSlicer y disponible para los principales sistemas operativos de escritorio.

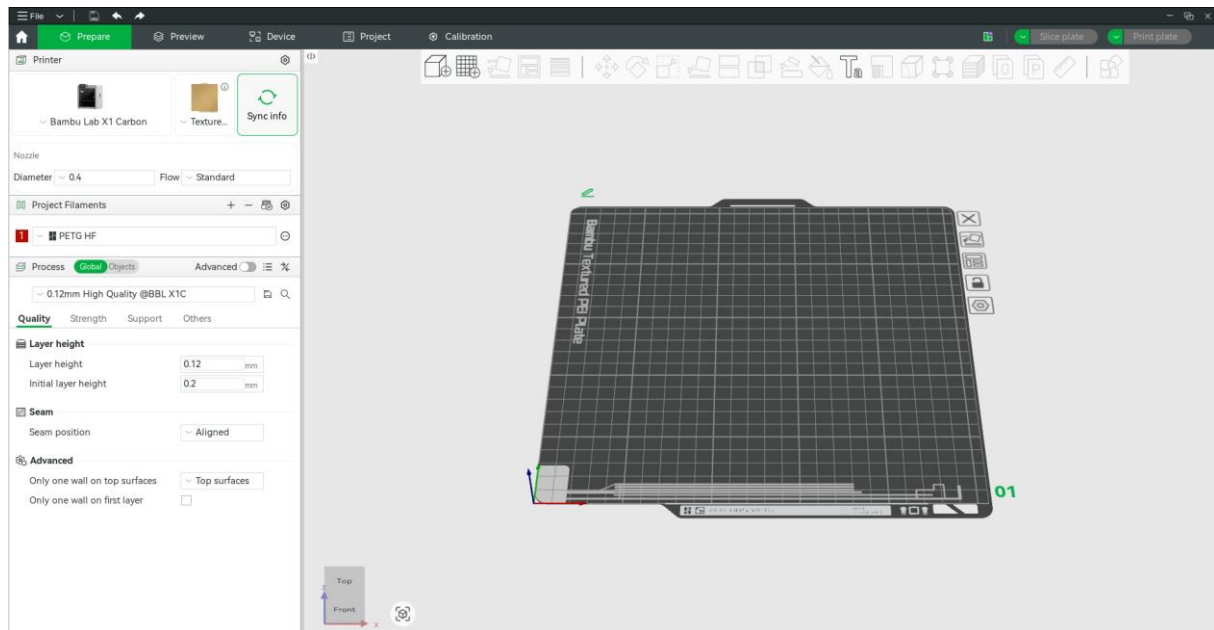


Figura 57: Software de segmentación Bambu Studio (Fuente: Edición propia)

Durante el proceso de segmentación, los estudiantes se enfrentaron al hecho de que la geometría por sí sola no es suficiente para una impresión exitosa; una parametrización adecuada es al menos igual de importante.

Al modificar los ajustes de segmentación, se hizo evidente la relación entre el tiempo de impresión, el consumo de material y la estabilidad estructural. Esta experiencia ayudó a los estudiantes a comprender el proceso de fabricación como un sistema complejo, en lugar de una serie de pasos aislados. Durante los procesos de segmentación, los perfiles de impresión predeterminados integrados en el software proporcionaron un buen punto de partida, que no solo abordaba la calidad de impresión, sino también la configuración óptima resultante del material utilizado. Con su ayuda, la diferencia entre las distintas configuraciones se hizo aún más clara.

Además de especificar los parámetros de impresión, la orientación de los modelos impresos, la configuración de color y la preparación de la bandeja eran igualmente importantes. Se debía prestar especial atención a la adhesión óptima de los modelos y a la necesidad de soportes.

5.6. Materiales (PLA, PETG, ABS)

Durante el proyecto, examinamos las propiedades de varios materiales de impresión 3D de uso común; sin embargo, para la producción real...**PETG** Elegimos el material. La decisión se basó en consideraciones tanto profesionales como pedagógicas.

El PETG ofrece un buen equilibrio entre facilidad de impresión y resistencia mecánica, además de ser menos propenso a la deformación que el ABS. Esto resultó especialmente beneficioso para piezas funcionales con tiempos de impresión prolongados. De esta forma, los estudiantes pudieron experimentar la importancia de la selección de materiales en una situación real de toma de decisiones de ingeniería.

Cabe destacar que, durante la ejecución del proyecto, también disponíamos de filamento PLA, considerado el material estándar para la mayoría de los proyectos de impresión 3D. Sin embargo, debido a su poca durabilidad y resistencia a los impactos físicos, lo descartamos y lo utilizamos únicamente para la creación de prototipos y pruebas. Además de los tres tipos de materiales mencionados, no investigamos el uso de otros tipos de filamento (por ejemplo, TPU).

5.7. Pasos en el proceso de impresión

La implementación práctica de la impresión 3D se llevó a cabo mediante pasos estructurados e interconectados. La estructura consciente del proceso permitió a los estudiantes interpretar las fases individuales no como operaciones aisladas, sino como parte de un sistema de producción coherente. Durante las etapas de impresión, se hizo especial hincapié en la preparación, el seguimiento del proceso y la evaluación de las piezas terminadas.

5.8. Preparación y verificación del modelo

En todos los casos, la impresión fue precedida por una preparación detallada del modelo, cuyo objetivo era prevenir errores de fabricación y utilizar los recursos de manera eficiente. Los estudiantes verificaron la corrección geométrica de los modelos digitales, prestando especial atención a los cuerpos cerrados, los espesores de las paredes y las superficies de contacto críticas.

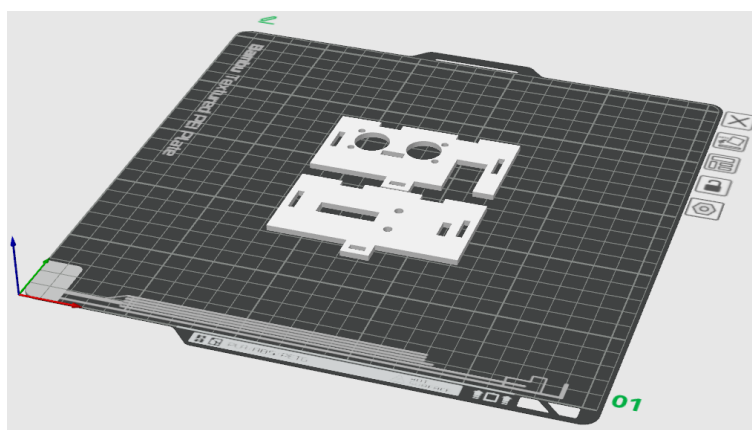


Figura 58: Preparación y verificación del modelo
(Fuente: Edición propia)

Durante la preparación, la elección de la orientación de impresión también jugó un papel destacado. Los estudiantes experimentaron que la colocación de la pieza en el espacio de impresión tiene un impacto directo en

Se requiere calidad superficial, resistencia mecánica y la cantidad adecuada de material de soporte. Esta situación de toma de decisiones facilitó el desarrollo de la percepción espacial y la aplicación consciente de aspectos de la tecnología de producción. Huelga decir que, con la disposición óptima de los elementos, los estudiantes tuvieron la oportunidad de iniciar procesos de impresión más optimizados, ya que podían colocar varios modelos en una misma bandeja.

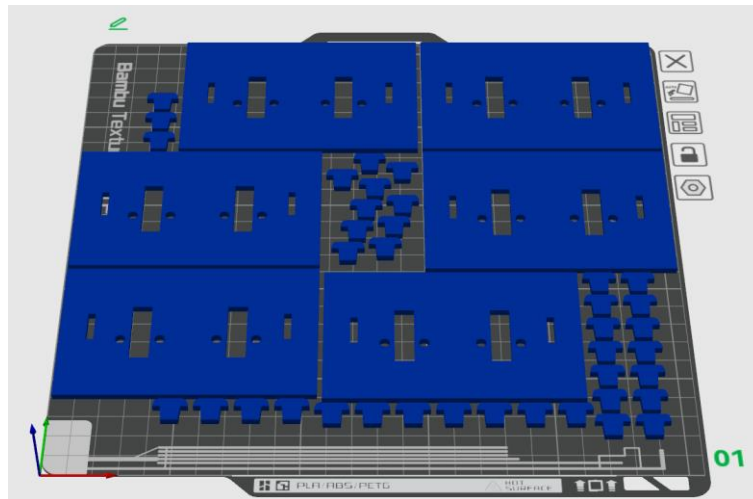


Figura 59: Optimización de la impresión
(Fuente: Edición propia)

Al finalizar el proceso de verificación, los modelos se validaron antes del seccionamiento, lo que confirmó la idea de que la detección temprana de errores es significativamente más efectiva que su corrección posterior.

5.9. Impresión y postproducción

Al inicio del proceso de impresión, los estudiantes prestaron especial atención a la correcta adhesión de la primera capa, uno de los factores más críticos para el éxito de toda la producción. La impresora X1 Carbon cuenta con un asistente de IA integrado que detiene la impresión brevemente tras la colocación de la primera capa y la inspecciona mediante la cámara incorporada. Si detecta algún problema, se lo indica al usuario a través del programa de laminado (o la aplicación móvil Bambu Handy, que se puede instalar por separado). Tras iniciar la impresión, consideramos más útil que los estudiantes participantes analizaran visualmente la calidad de la impresión. La observación de la primera capa les permitió recibir retroalimentación inmediata sobre la correcta preparación y configuración.



Figura 60. La impresora en funcionamiento.

La monitorización continua del proceso durante las impresiones de mayor duración también constituyó un elemento de aprendizaje importante. Los estudiantes comprobaron que la impresión no es una operación totalmente autónoma, sino un proceso de fabricación que puede requerir intervención o rediseño. Esto ocurre, por ejemplo, si un objeto se desplaza o si el filamento se agota en el dispensador AMS. Los estudiantes también tuvieron la oportunidad de observar cómo se gestionan estas situaciones.

Tras la impresión, las piezas terminadas se sometieron a un proceso de postproducción. Este proceso incluyó la corrección de irregularidades superficiales y el refinamiento de los puntos de unión. Durante el postprocesamiento, se hizo evidente que la impresión 3D a veces da como resultado un producto semiacabado que requiere trabajo manual adicional para su uso funcional. Este trabajo se incrementa con la complejidad de los modelos y la cantidad de soportes.



Figura 61: Resultados de una impresión incorrecta (Fuente: Edición propia)

5.10. Control de calidad y corrección de errores

En todos los casos, las piezas terminadas fueron sometidas a un control de calidad exhaustivo. La inspección no solo abarcó aspectos estéticos, sino que evaluó principalmente la idoneidad funcional y el montaje.

Los estudiantes compararon las dimensiones de las piezas impresas con los datos del modelo digital y analizaron las causas de las discrepancias. Al identificar los errores, también tuvieron en cuenta los efectos de la configuración de laminado, las características del material y el entorno de impresión. Por lo tanto, la corrección de errores no fue una intervención aislada, sino un ciclo de retroalimentación a lo largo de todo el proceso de fabricación.

Este enfoque fomentó el desarrollo del pensamiento analítico y sentó las bases para la ejecución profesionalmente justificada de las iteraciones de diseño posteriores.

5.11. Problemas de talla y ajuste

Una de las lecciones más importantes del proyecto fue la importancia práctica de las cuestiones de tamaño y ajuste. Los estudiantes comprobaron de primera mano que las dimensiones nominales utilizadas en el diseño digital no siempre garantizan un ajuste adecuado durante la fabricación; por lo tanto, es fundamental gestionar cuidadosamente las tolerancias.

5.12. Manejo de imprecisiones de fabricación

Las imprecisiones durante la impresión 3D fueron consecuencia de una combinación de factores, como la contracción del material, las características de la estructura de las capas y la orientación de impresión. Estos fenómenos permitieron a los estudiantes comprender las limitaciones de la tecnología de fabricación y requirieron ajustes mínimos en el tamaño de los modelos.

Para abordar las variaciones de fabricación, los estudiantes aprendieron sobre el uso de holguras de ajuste y que las piezas móviles o entrelazadas requieren un enfoque de diseño diferente al de las piezas sólidas. Este enfoque contribuyó a profundizar su pensamiento técnico.

5.13. Iteraciones de diseño

La solución de problemas de dimensionamiento a menudo requería múltiples ciclos de diseño y fabricación. También era necesario abordar casos en los que las piezas no encajaban debido a errores de diseño, y no al proceso de impresión. Basándose en su experiencia, los estudiantes modificaron los modelos y reimprimieron las piezas afectadas, lo que permitió obtener retroalimentación rápida sobre la corrección de sus decisiones.

La ventaja del diseño paramétrico se hizo especialmente evidente durante el proceso iterativo, ya que la capacidad de modificar y refabricar dimensiones rápidamente permitió una experimentación eficiente. Este método ayudó a los estudiantes a pensar en un proceso de desarrollo flexible en lugar de en soluciones estáticas.

5.14. La importancia educativa de los procesos, las experiencias de los estudiantes y las lecciones aprendidas.

El manejo de los problemas de dimensionamiento y ajuste tuvo un valor educativo excepcional, ya que los estudiantes se enfrentaron a situaciones reales de ingeniería. Su pensamiento independiente y su capacidad para tomar decisiones responsables se desarrollaron mediante el análisis de problemas, la identificación de errores y la elaboración de soluciones.

Los estudiantes comprendieron que el diseño y la fabricación están estrechamente relacionados, y que ninguno puede entenderse sin el otro. Estas experiencias contribuyeron al proyecto no solo transmitiendo conocimientos tecnológicos, sino también desarrollando un enfoque que puede utilizarse a largo plazo.

Además, la experiencia adquirida en la impresión 3D ofrece oportunidades adicionales de transferencia de conocimientos, que pueden utilizarse al aprender otras materias (matemáticas, física, materias técnicas), así como incluso en el ámbito doméstico, al manejar su propia impresora 3D.



Figura 62. Carcasa impresa y ensamblada

6. Sistema de control integrado: fundamentos conceptuales. El

6.1. propósito del diseño del sistema.

En la fase de electrónica y software del proyecto se tomó una decisión de diseño consciente: el sistema no es un dispositivo monolítico basado en un único microcontrolador, sino que está implementado en forma de una arquitectura de control distribuida.

Es importante destacar que esto no se debe a la supuesta "debilidad" del ESP32. El ESP32 es un excelente controlador de campo y unidad de adquisición de datos: proporciona una gestión de E/S digitales estable, cuenta con múltiples interfaces de comunicación (I²C, SPI, UART, WiFi) y también es adecuado para ejecutar lógica de control independiente. Si el número de GPIO no es suficiente, de acuerdo con la práctica habitual en la industria, se pueden utilizar expansores de E/S (por ejemplo, expansores de puertos I²C) o nodos adicionales; esto, por sí solo, apunta hacia sistemas distribuidos.

La limitación real durante el desarrollo no estuvo en el área de E/S o comunicación, sino en la implementación de las funciones de monitoreo y visualización (HMI). El ESP32:

- tiene memoria RAM y espacio de almacenamiento flash limitados,
- optimizado para ejecutar interfaces de usuario no web, administración de bases de datos y registro persistente,
- Al mostrar gráficos complejos y atender a varios clientes, rápidamente alcanza los límites de recursos.

En la práctica, esto significaba que, si bien el ESP32 era capaz de realizar tareas de campo de forma estable (sensores, actuadores, control local), la capa de monitorización del sistema justificaba la inclusión de un dispositivo con mayor capacidad de procesamiento.

La arquitectura resultante divide el sistema en capas claramente diferenciadas:

Nivel	Función	Dispositivo
Nivel de campo	Recopilación de datos de sensores, control de actuadores, lógica local	ESP32
Control / nivel de gestión de datos	Procesamiento cíclico, decisiones lógicas, rol de maestro Modbus	Raspberry Pi / PC (Modbus)
Nivel de supervisión (HMI)	Visualización web, control de usuario, registro de datos	Raspberry Pi / PC (servidor basado en Flask)

En este modelo, el ESP 32 actúa como un nodo de campo y, siguiendo un patrón industrial, proporciona datos al sistema de nivel superior mediante un protocolo estándar (Modbus TCP). Es importante destacar que en un sistema monolítico de un solo dispositivo, este estándar de comunicación no sería necesario; sin embargo, la arquitectura distribuida lo justifica.

Por lo tanto, el objetivo del proyecto no era simplemente crear un modelo de granja inteligente funcional, sino demostrar que:

- cómo un microcontrolador se convierte en un nodo de campo,
- cómo se organiza por encima de él una capa lógica del lado del sistema de supervisión y una capa HMI,
- y cómo la arquitectura en capas y en red de la automatización industrial puede modelarse en un entorno educativo.

Este enfoque acerca a los estudiantes al funcionamiento de sistemas industriales reales, al tiempo que el sistema puede ampliarse fácilmente con nuevos sensores, nodos y servicios.

El concepto de sistema distribuido

El modelo de control desarrollado en el proyecto sigue conscientemente una arquitectura de dos niveles y en capas. Esta estructura no fue solo una decisión técnica, sino también un paso adelante en cuanto al enfoque, pasando del diseño monolítico de "todo en un solo microcontrolador" a la estructura típica de los sistemas industriales.

El nivel inferior es el nivel de campo, donde se gestionan directamente los sensores y actuadores. El ESP32 desempeña esta función en el sistema. Sus tareas incluyen el procesamiento de señales digitales y analógicas, la recopilación de datos de los sensores, el preprocesamiento básico y la operación de salidas físicas (por ejemplo, LED, relé, servo). En este nivel se establece la conexión directa con el proceso físico real.

La capa superior es el nivel de supervisión, que en este proyecto se implementa mediante un sistema basado en Raspberry Pi (o PC). Una diferencia importante es que, en este nivel, la lógica de control central no se ejecuta, sino que se muestra una imagen digital del estado del sistema. La Raspberry Pi recopila datos de los dispositivos de campo (ESP32), los registra y los muestra en una interfaz web HMI, lo que permite la intervención remota.

El control real se realiza a nivel de campo, en el ESP32. La gestión de E/S, el procesamiento de datos de los sensores y la lógica operativa básica se ejecutan localmente, por lo que el sistema permanece operativo incluso en modo sin conexión. Un fallo en la capa de monitorización no detiene el proceso físico; solo se ven afectados la visualización y el acceso remoto.

Por lo tanto, esta estructura puede compararse con mayor precisión con el siguiente modelo industrial:



Los equivalentes en el proyecto son:

- ESP32 como una unidad de campo inteligente que
 - él Realiza el manejo de E/S
 - él recopila datos de sensores
 - él ejecuta lógica de control local
 - él Se comunica como un sistema de supervisión Modbus.
- Raspberry Pi + Python + Flask como capa de gestión, que
 - él Recopila y muestra datos que
 - él reflejan el estado del sistema.

- él permite la intervención del operador, pero no es
- él fundamental para el funcionamiento del proceso.

En el futuro, se podrán implementar funciones adicionales a nivel de Raspberry Pi, tales como:

- registro de datos y análisis de tendencias a largo plazo
- **visualización gráfica**
- gestión de autorización de usuarios
- notificaciones, registros de alarmas
- Interfaz de configuración remota

Estas funciones deben ser únicamente de supervisión e información. Es un principio de diseño que las adiciones que se ejecutan en la Raspberry Pi no deben afectar la funcionalidad básica ni la seguridad del sistema. La lógica de control crítica y la gestión básica del estado permanecen en el nivel de campo.

6.2. Ventajas técnicas de la arquitectura distribuida

Una de las claves del sistema es la separación funcional, que aumenta directamente la fiabilidad y la escalabilidad. El ESP32 se mantiene a nivel de campo: los sensores, las entradas, las salidas y la lógica de control básica se ejecutan localmente, por lo que el funcionamiento del proceso físico no depende de una conexión de red continua. Esto se traduce en un comportamiento de E/S más determinista y un funcionamiento más estable.

La capa superior se encarga de la monitorización, la gestión de datos y la visualización, no del control crítico. Esta arquitectura por capas sigue un patrón industrial, donde el dispositivo de campo, la capa de control y el sistema de monitorización operan con funciones independientes.

Una ventaja significativa de esta arquitectura es su escalabilidad: se pueden añadir nuevos nodos basados en ESP a la red, integrar sensores adicionales y ampliar funcionalmente el sistema sin sobrecargar los recursos de un solo dispositivo. Esta expansión horizontal es una característica fundamental de los sistemas distribuidos.

Desde la perspectiva de la fiabilidad, el principio es que el nivel de campo permanezca autónomo: un fallo del sistema de supervisión o de la red no interrumpe las operaciones locales, solo se ven afectadas la monitorización y la intervención remota. Esto se corresponde directamente con la práctica de la automatización industrial.

6.3. Modelo y enfoque de comunicación

La conexión entre las capas se basa en Modbus TCP. Este protocolo implementa un intercambio de datos basado en estados con una estructura de registro estándar. Los datos de los sensores y las señales de control no se transmiten mediante solicitudes y respuestas específicas, sino como estados accesibles en direcciones bien definidas. Esto difiere del enfoque transaccional típico basado en HTTP y se asemeja más a los modelos de comunicación industrial, donde el controlador lee cíclicamente los estados de los campos y luego escribe en los registros de salida correspondientes.

Marco del sistema: estructura lógica

A grandes rasgos, la ruta de datos del sistema se puede describir de la siguiente manera:

Los sensores (botones, DHT, etc.) y actuadores del mundo físico se conectan al ESP32. El ESP32 actúa como un sistema supervisor Modbus y proporciona el estado de las entradas y salidas a través de registros. Mediante la red TCP basada en Wi-Fi, el sistema que opera como supervisor (Raspberry Pi o PC) lee y escribe estos datos cíclicamente como cliente Modbus. Los estados procesados y los valores medidos se muestran en la interfaz web HMI, donde también se implementan la visualización y el control remoto.

En este modelo, el ESP no es un controlador central, sino una unidad inteligente de E/S de campo, mientras que la Raspberry Pi actúa como centro de lógica y monitorización.

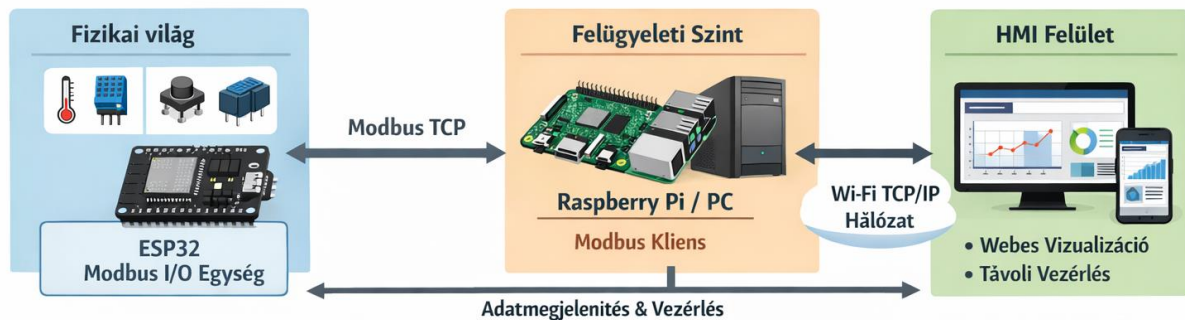


Figura 63: Estructura del sistema (Fuente: Generado por IA)

Importancia educativa

La arquitectura aplicada no es solo una solución técnica, sino también una herramienta para moldear actitudes. Los estudiantes comprenden la razón de la separación entre los niveles de trabajo y supervisión, el papel de la asignación de recursos y cómo se construye la estructura por capas de un sistema de automatización industrial. Esto proporciona una base directa para comprender la programación de PLC, el diseño de HMI y los sistemas SCADA, ya que el modelo utilizado en el proyecto es una representación simplificada pero realista de estos.

En la siguiente sección, presentamos la implementación específica del software de la arquitectura: el programa de campo ESP, el procesamiento cíclico en el lado de la monitorización y el funcionamiento de la interfaz web.

6.4. Nodo de control de campo: unidad de muestreo independiente

El nivel inferior del sistema electrónico del proyecto está representado por un nodo de campo autónomo, basado en un microcontrolador ESP32. Esta unidad está conectada directamente al mundo físico: lee sensores, controla actuadores y proporciona retroalimentación local. La configuración presentada es un ejemplo didáctico deliberadamente sencillo que ilustra la lógica básica del sistema y no representa la versión final.

El nodo gestiona los siguientes elementos:

- un botón pulsador que funciona como entrada digital,
- un LED de salida digital,
- un sensor de temperatura y humedad DHT11,
- una pantalla LCD de 16x2 caracteres conectada a un bus I²C.

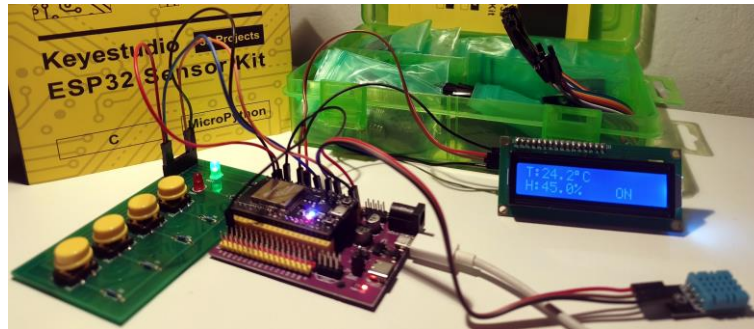


Figura 64: Entorno de prueba (Fuente: Edición propia)

La esencia de su funcionamiento radica en que el dispositivo realiza sus tareas básicas de forma totalmente independiente, sin conexión a la red. El usuario puede intervenir en el estado del sistema (encendido/apagado del LED) mediante el botón físico, mientras que los datos ambientales se miden y visualizan continuamente en la pantalla. Este enfoque reproduce el principio industrial de que el equipo de campo debe poder operar de forma autónoma, independientemente del sistema de monitorización.

El programa presentado demuestra el comportamiento de una unidad de E/S inteligente. Su estructura es modular: se pueden añadir fácilmente nuevos sensores, dispositivos de salida o pantallas mediante GPIO o buses de comunicación adicionales. Posteriormente, este nodo también puede integrarse con sistemas de nivel superior (PLC, HMI, SCADA), pero en este capítulo nos centramos exclusivamente en la gestión de recursos locales.

6.5. Estructura del programa desglosada en unidades funcionales.

Bibliotecas y definiciones de hardware.

```
1 // ===== KÖNYVTÁRAK BETÖLTÉSE =====
2 // Ebben a szakaszban történik a felhasznált perifériákhoz szükséges könyvtárak betöltése.
3 // DHT könyvtár a szenzor kommunikációját valósítja meg.
4 // A Wire és hd44780_I2Cexp az I²C LCD vezérléséhez szükséges.
5 #include <DHT.h>
6 #include <Wire.h>
7 #include <hd44780.h>
8 #include <hd44780ioClass/hd44780_I2Cexp.h>
9
10
11 // ===== HARDVER KIOSZTÁS =====
12 //A #define direktívák rögzítik a fizikai bekötés és a program közötti kapcsolatot, így a
13 //hardverkonfiguráció egyértelműen dokumentált és könnyen módosítható.
14 #define BTN 17 // Fizikai gomb bemenet
15 #define LED 2 // LED kimenet
16 #define DHTPIN 4 // DHT adatvonal
17 #define DHTTYPE DHT11 //DHT típus megadása
```

En esta etapa, se cargan las bibliotecas necesarias para los periféricos utilizados.

- La biblioteca DHT implementa la comunicación entre sensores.
- Se requieren los módulos Wire y hd44780_I2Cexp para controlar la pantalla LCD I²C.

Las directivas #define registran la relación entre la conexión física y el programa, de modo que la configuración del hardware queda claramente documentada y se puede modificar fácilmente.

Objetos y variables globales

```
20 //Itt jönnek létre a szenzor- és kijelzőobjektumok. Ezek a program teljes futása alatt elérhetők, így
    bármelyik ciklusban használhatók.
21 // Szenzor objektum
22 DHT dht(DHTPIN, DHTTYPE);
23 // I2C LCD objektum
24 hd44780_I2Cexp lcd;
```

Aquí es donde se crean los objetos del sensor y la pantalla. Están disponibles durante toda la ejecución del programa, por lo que pueden utilizarse en cualquier ciclo.

Manejo del rebote de botones

```
27 // ===== GOMB PRELLEGÉS SZŰRÉS =====
28 // A mechanikus gombok érintkezési zárásakor rövid ideig „rezegnek” (prellegnek), ami több hamis jelváltást
    okozhat.
29 // Ez a logika időalapú szűréssel biztosítja, hogy csak a stabil állapotváltás legyen érvényes.
30
31 bool lastReading = false; // Legutóbbi nyers bemenet
32 bool stableState = false; // Szűrt, stabil állapot
33 unsigned long lastChangeTime = 0;
34 const unsigned long debounceTime = 50; // 50 ms stabilitási küszöb
```

Los contactos de los botones mecánicos vibran (rebotan) brevemente al cerrarse, lo que puede provocar múltiples transiciones de señal erróneas. Esta lógica utiliza un filtrado basado en el tiempo para garantizar que solo sean válidas las transiciones de estado estables.

Variables de estado y medición del LED

```
37 // A LED állapota külön változóban tárolódik. A DHT mérések időzítéséhez és az aktuális értékek
    tárolásához is globális változók szükségesek.
38 bool ledState = false;
39 unsigned long lastDHTread = 0; // Utolsó olvasás időpontja
40 float temp = 0; // Hőmérséklet cache
41 float hum = 0; // Páratartalom cache
```

El estado del LED se almacena en una variable independiente. También se requieren variables globales para la medición de la temporización del DHT y el almacenamiento de los valores actuales.

Inicialización - configuración()

```
50 void setup() {
51
52     // Hardver inicializálás
53     pinMode(BTN, INPUT);
54     pinMode(LED, OUTPUT);
55
56     // DHT inicializálás
57     dht.begin();
58
59     // I2C inicializálás ESP32 alapértelmezett lábakkal
60     Wire.begin(21,22);
61
62     // LCD inicializálás
63     lcd.begin(16,2);
64     lcd.backlight();
65
66     lcd.setCursor(0,0);
67     lcd.print("Smart Farm Node");
68     delay(2000);
69     lcd.clear();
    }
```

La función setup() se ejecuta una sola vez al iniciar el sistema. Esto es lo que sucede:

- configurar las direcciones GPIO,
- inicialización del sensor y del bus I²C,
- Encendiendo la pantalla LCD.

Control por botones y conmutación LED

Este bloque implementa el cambio de color del LED al presionar el borde. El LED solo cambia cuando se presiona el botón de forma continua.

```
74 //Ez a blokk valósítja meg a lenyomási élre történő LED-váltást.
75 //A LED csak akkor változik, amikor a gomb stabilan lenyomott állapotba kerül.
76 // Gomb olvasása
77 bool reading = digitalRead(BTN);
78
79 // Prellégés detektálása
80 if (reading != lastReading) {
81     lastChangeTime = millis();
82 }
83
84 // Stabil állapot vizsgálata
85 if ((millis() - lastChangeTime) > debounceTime) {
86
87     // Állapotváltozás kezelése
88     if (reading != stableState) {
89         stableState = reading;
90
91         // Lenyomás detektálás
92         if (stableState == true) {
93             ledState = !ledState; // LED kapcsolás
94         }
95     }
96 }
97
98 lastReading = reading;
```

Medición periódica del DHT11

```
100 // DHT szenzor olvasás időzített
101 //A szenzor csak 2 másodpercenként kerül kiolvasásra, ami megfelel a DHT11 időzítési követelményeinek.
102 if(millis() - lastDHTread > 2000){
103     lastDHTread = millis();
104
105     float h = dht.readHumidity();
106     float t = dht.readTemperature();
107
108     // Hibás olvasás kizárása
109     if(!isnan(h) && !isnan(t)){
110         temp = t;
111         hum = h;
112     }
113 }
```

El sensor se lee solo cada 2 segundos, lo que cumple con los requisitos de temporización del DHT11.

Pantalla LCD

```
115 // LCD kijelzés
116 //A kijelző a mért adatokat és a LED állapotát jeleníti meg, így a rendszer állapota hálózat nélkül is ellenőrizhető.
117 lcd.setCursor(0,0);
118 lcd.print("T:");
119 lcd.print(temp,1);
120 lcd.print((char)223);
121 lcd.print("C ");
122
123 lcd.setCursor(0,1);
124 lcd.print("H:");
125 lcd.print(hum,1);
126 lcd.print("% ");
127
128 lcd.setCursor(11,1);
129 lcd.print(ledState ? "ON " : "OFF");
```

La pantalla muestra los datos medidos y el estado de los LED, por lo que se puede comprobar el estado del sistema incluso sin conexión a la red.

Resumen

Este programa demuestra el funcionamiento de una unidad de control de campo autónoma:

- procesamiento de entrada física
- control del actuador
- recopilación de datos de sensores
- visualización local

La estructura del código separa bien las funciones individuales, lo que facilita su posterior expansión y conexión con sistemas de nivel superior.

6.6. Nodo de control de campo con expansión de red – WiFi + Modbus TCP

En esta versión, el controlador de campo independiente presentado anteriormente se complementa con capacidades de red.

El funcionamiento local (botón → LED, medición DHT, pantalla LCD) permanece sin cambios, pero el dispositivo ahora es capaz de:

- Conéctate a una red WiFi
- Funciona como un sistema de monitorización Modbus TCP.
- Hacer que el estado de las entradas y salidas físicas esté disponible a través de registros.
- para lectura e intervención remota mediante un sistema externo (PLC/software de monitorización)

Es importante que la lógica no salga del dispositivo de campo: el control local permanece autónomo, la red solo se utiliza para compartir el estado y para la intervención remota.

Estructura del programa según unidades funcionales

Bibliotecas y definiciones de hardware

```
1 #include <WiFi.h> // ESP32 WiFi stack
2 #include <ModbusTCP.h> // Modbus TCP szerver kezelés
3 #include <DHT.h> // DHT szenzor magas szintű kezelése (protokollt elrejt)
4 #include <Wire.h> // I2C kommunikáció (LCD miatt)
5 #include <hd44780.h> // LCD vezérlő
6 #include <hd44780ioClass/hd44780_I2Cexp.h> // I2C LCD expander kezelés
7
8 ModbusTCP mb; // Az ESP Modbus TCP szerverének létrehozása.
9
10
11 // ===== HARDVER KIOSZTÁS =====
12 // GPIO-k definiálása hardver absztrakcióként
13
14 #define BTN 17 // Fizikai gomb bemenet
15 #define LED 2 // Fő LED kimenet (vezérelt fogyasztó)
16 #define WIFI_LED 5 // WiFi státusz visszajelző LED (terepi hibajelzés)
17 #define DHTPIN 4 // DHT adatvonal
18 #define DHTTYPE DHT11 // Szenzor típusa
19
20 // Szenzor objektum létrehozása
21 DHT dht(DHTPIN, DHTTYPE);
22
23 // I2C LCD objektum létrehozása (automatikus I2C címfelismerés)
24 hd44780_I2Cexp lcd;
```

Parametrización de red

```
27 // ===== WIFI HÁLÓZATI PARAMÉTEREK =====
28 // Ezek konfigurálják a hálózati kapcsolatot, valamint biztosítják,
29 // hogy az ESP mindig ugyanazon IP címen legyen elérhető.
30
31 const char* ssid = "plc";
32 const char* pass = "12345678";
33
34 IPAddress local_IP(192, 168, 137, 200);
35 IPAddress gateway(192, 168, 137, 1);
36 IPAddress subnet(255, 255, 255, 0);
37 IPAddress primaryDNS(8,8,8,8);
38 IPAddress secondaryDNS(8,8,4,4);
39
40
41 // ===== WIFI ÚJRACSATLAKOZÁSI LOGIKA =====
42 // A rendszer nem blokkoló reconnect stratégiát használ.
43 // Ha a kapcsolat megszakad, a program nem áll meg,
44 // hanem meghatározott időközönként újrapróbál csatlakozni.
45
46 unsigned long lastReconnectAttempt = 0; // Utolsó reconnect próbálkozás időpontja
47 const unsigned long reconnectInterval = 300000; // 5 perc újrapróbálkozási ciklus
48 bool wifiConnected = false; // Hálózati állapot logikai jelző
49
50 // WiFi csatlakozási rutin
51 // Ez a függvény bontja az előző kapcsolatot,
52 // majd új kapcsolatot kezdeményez a megadott SSID-re.
53 void connectToWiFi() {
54     WiFi.disconnect(true);
55     delay(100); // Rövid stabilizációs várakozás
56     WiFi.begin(ssid, pass);
57 }
58
59
60 // ===== WIFI HIBA LED IDŐZÍTÉS =====
61 // Amennyiben nincs WiFi kapcsolat,
62 // a WIFI_LED másodperces periódussal villog.
63 // A megvalósítás millis() alapú, tehát nem blokkolja a fő ciklust.
64
65 unsigned long lastBlinkTime = 0; // Utolsó LED állapotváltás
66 const unsigned long blinkInterval = 1000; // 1 másodperces villogási periódus
67 bool wifiLedState = false; // WiFi LED aktuális állapota
```

Gestión de eliminación de rebotes de botones, estado del LED y variables de medición

```
70 // ===== BELSŐ ÁLLAPOT RÉTEG =====
71 // A rendszer egyetlen "igazságforrást" használ a LED állapotára.
72 // A fizikai gomb, a Modbus és a fizikai LED is ezt az állapotot követi.
73
74 bool ledState = false; // A LED központi logikai állapota
75
76
77 // ===== GOMB PRELL SZŰRÉS =====
78 // Mechanikus gomb érintkezése záráskor pattog (prell).
79 // A stabil állapot detektálás millis alapú időszűréssel történik.
80
81 bool lastReading = false; // Utolsó nyers bemeneti érték
82 bool stableState = false; // Szűrt, stabil állapot
83 unsigned long lastChangeTime = 0;
84 const unsigned long debounceTime = 50; // 50 ms stabilitási küszöb
85
86
87 // ===== DHT MINTAVÉTELI IDŐZÍTÉS =====
88 // A DHT szenzor olvasása nem történhet túl gyakran,
89 // ezért 2 másodperces mintavételezési ciklust alkalmazunk.
90
91 unsigned long lastDHTread = 0;
92 float temp = 0;
93 float hum = 0;
```

Inicialización

```
97 // ===== SETUP =====
98 // =====
99 void setup() {
100
101     Serial.begin(115200);
102
103     // Hardver inicializálás
104     pinMode(BTN, INPUT);
105     pinMode(LED, OUTPUT);
106     pinMode(WIFI_LED, OUTPUT);
107
108     // ===== Stabil statikus IP konfiguráció =====
109     // Az ESP32 WiFi stack sajátossága miatt:
110     // - Station mód explicit beállítása szükséges
111     // - DHCP cache törlése
112     // - Statikus IP konfigurálása csatlakozás előtt
113
114     WiFi.mode(WIFI_STA);
115     WiFi.persistent(false);
116     WiFi.disconnect(true);
117     delay(100);
118
119     WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS);
120     WiFi.begin(ssid, pass);
121
122     // Induláskori maximum 10 másodperces várakozás.
123     // Ha ezalatt nem jön létre kapcsolat,
124     // a program offline módban folytatódik.
125     unsigned long startAttemptTime = millis();
126     while (WiFi.status() != WL_CONNECTED &&
127           millis() - startAttemptTime < 10000) {
128         delay(500);
129     }
130
131     wifiConnected = (WiFi.status() == WL_CONNECTED);
132
133     // Szenzor és perifériák inicializálása
134     dht.begin();
135     Wire.begin(21,22);
136
137     lcd.begin(16,2);
138     lcd.backlight();
139     lcd.setCursor(0,0);
140     lcd.print("Smart Farm Node");
141     delay(2000);
142     lcd.clear();
```

Asignación de registros Modbus

```
144 // ===== MODBUS REGISZTERKIOSZTÁS =====
145 // Coil 0 → LED állapot
146 // Ists 0 → Gomb állapot
147 // Hreg 0 → Hőmérséklet (x10 skálázva)
148 // Hreg 1 → Páratartalom (x10 skálázva)
149
150 mb.server();
151 mb.addCoil(0);
152 mb.addIsts(0);
153 mb.addHreg(0);
154 mb.addHreg(1);
155
156 // Coil inicializálása a belső logikai állapotból
157 mb.Coil(0, ledState);
158 }
```

Monitorización WIFI, servicio Modbus con conexión activa

```
164 void loop() {
165
166     // ===== WIFI FELÜGYELET =====
167     // Nem blokkoló hálózati állapotellenőrzés.
168     // Ha nincs kapcsolat, 5 percenként újrapróbál.
169
170     if (WiFi.status() != WL_CONNECTED) {
171
172         wifiConnected = false;
173
174         if (millis() - lastReconnectAttempt >= reconnectInterval) {
175             lastReconnectAttempt = millis();
176             connectToWiFi();
177         }
178     }
179     else {
180         wifiConnected = true;
181     }
182
183     // ===== WIFI HIBA LED KEZELÉS =====
184     // Offline állapotban 1 Hz villogás.
185     // Online állapotban LED kikapcsolva.
186
187     if (!wifiConnected) {
188
189         if (millis() - lastBlinkTime >= blinkInterval) {
190             lastBlinkTime = millis();
191             wifiLedState = !wifiLedState;
192             digitalWrite(WIFI_LED, wifiLedState);
193         }
194     }
195     else {
196         digitalWrite(WIFI_LED, LOW);
197         wifiLedState = false;
198     }
199
200     // ===== MODBUS KISZOLGÁLÁS =====
201     // Csak aktiv hálózat esetén kezel TCP kéréseket.
202     if (wifiConnected) {
203         mb.task();
204     }
205
206     // =====
207     // ===== MODBUS - BELSŐ ÁLLAPOT SZINKRON =====
208     // =====
209     // Ha külső rendszer írja a Coil 0-t,
210     // a belső állapot frissül.
211
212     bool coilState = mb.Coil(0);
213
214     if (coilState != ledState) {
215         ledState = coilState;
216     }

```

Lectura de botones, control del estado del LED y monitorización remota.

```
218 // =====
219 // ===== GOMB - BELSŐ ÁLLAPOT =====
220 // =====
221 // Fizikai gombnyomás esetén a belső állapot toggolódik,
222 // majd a Coil regiszter tükrözi azt.
223
224 bool reading = digitalRead(BTN);
225
226 if (reading != lastReading) {
227     lastChangeTime = millis();
228 }
229
230 if ((millis() - lastChangeTime) > debounceTime) {
231
232     if (reading != stableState) {
233         stableState = reading;
234
235         if (stableState == true) {
236             ledState = !ledState;
237             mb.Coil(0, ledState);
238         }
239     }
240 }
241
242 lastReading = reading;
243
244 // =====
245 // ===== FIZIKAI LED VEZÉRLÉS =====
246 // =====
247 // A fizikai kimenet mindig a belső állapotot követi.
248
249 digitalWrite(LED, ledState);
```

Medición periódica, monitorización remota del DHT11, informe del estado de los botones mediante Modbus.

```
251 // =====
252 // ===== DHT IDŐZÍTETT OLVASÁS =====
253 // =====
254
255 if (millis() - lastDHTread > 2000) {
256
257     lastDHTread = millis();
258
259     float h = dht.readHumidity();
260     float t = dht.readTemperature();
261
262     if (!isnan(h) && !isnan(t)) {
263         mb.Hreg(0, (int)(t * 10));
264         mb.Hreg(1, (int)(h * 10));
265         temp = t;
266         hum = h;
267     }
268 }
269
270 // Gomb státusz jelentése Modbuson
271 mb.Ists(0, stableState);
```

Pantalla LCD

```
273 // =====  
274 // ===== LCD KIJELZÉS =====  
275 // =====  
276 // Első sor: Hőmérséklet + Hálózati állapot  
277 // Második sor: Páratartalom + LED állapot  
278  
279 lcd.setCursor(0,0);  
280 lcd.print("T:");  
281 lcd.print(temp,1);  
282 lcd.print((char)223);  
283 lcd.print("C ");  
284 lcd.print(wifiConnected ? "N-ON " : "N-OFF");  
285  
286 lcd.setCursor(0,1);  
287 lcd.print("H:");  
288 lcd.print(hum,1);  
289 lcd.print("% ");  
290 lcd.print(ledState ? "L-ON " : "L-OFF");  
291  
292 delay(10); // CPU terhelés csökkentése  
293 }
```

Diagrama de bloques del sistema

El diagrama de bloques ilustra la arquitectura en capas, similar a la de un PLC, del firmware basado en ESP32. En el nivel superior se encuentran la interfaz WiFi y la comunicación ModbusTCP, responsables exclusivamente del intercambio de datos de red y la gestión del espacio de registros. El núcleo del sistema es la capa de estado, que almacena todas las variables de estado internas (como el estado de los LED, la temperatura medida, la humedad y el estado de la conexión WiFi) como una única fuente de información fiable.

La capa de entrada gestiona las entradas físicas, mientras que la capa de salida gestiona las salidas, y ambas dependen directamente de la capa de estado interno. El control reside completamente en el ESP32, es decir, a nivel de campo, por lo que el sistema opera de forma autónoma, independientemente de la red. Las capas Modbus y de supervisión solo reflejan el estado interno como una representación digital, pero no contienen la lógica de control. Esta estructura garantiza un funcionamiento determinista y sin conexión, así como la arquitectura robusta típica de los sistemas industriales.

Firmware Architecture

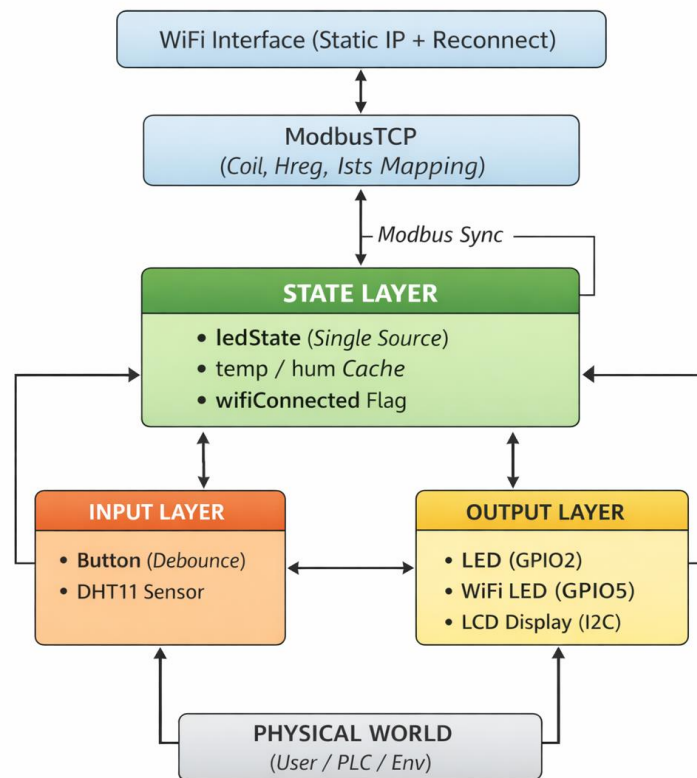


Figura 65: Diagrama de bloques del sistema (Fuente: Generado por IA)

6.7. Nivel de supervisión: HMI basada en Python / ciclo de supervisión

El nodo ESP32 de campo permite acceder al estado de los sensores y actuadores mediante registros Modbus. El nivel superior no se encarga del control físico, sino del mapeo y la monitorización digital del sistema.

Esta capa del proyecto es una aplicación cíclica basada en Python, que funciona como un sistema de monitorización de tipo HMI/SCADA.

Tareas:

- consulta cíclica de datos de campo
- estados de almacenamiento y procesamiento
- proporcionar opciones de intervención remota
- servicio de visualización web

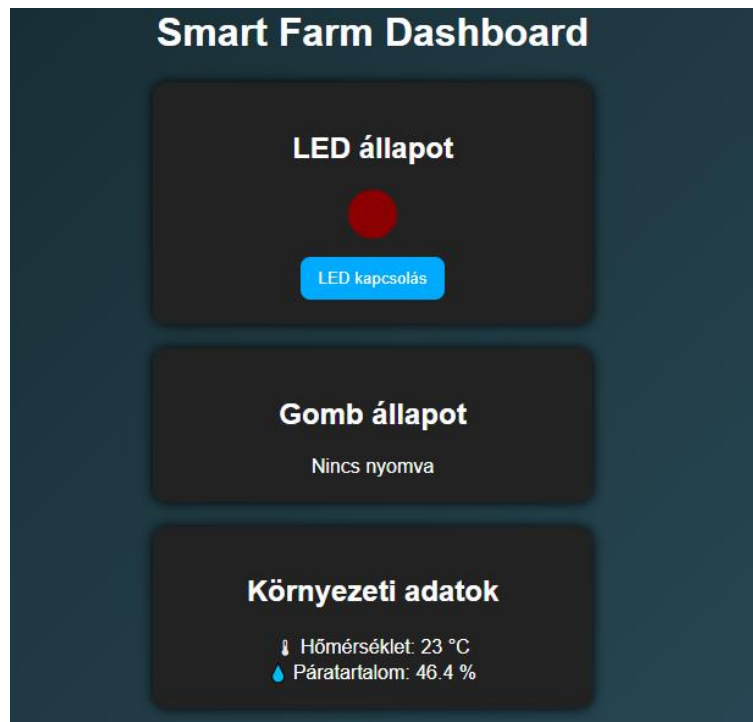


Figura 66: Interfaz hombre-máquina (HMI) basada en Python (Fuente: Edición propia)

6.8. Preparación e implementación de un sistema de monitorización basado en Raspberry Pi.

El siguiente paso fue ejecutar la lógica de monitorización del software en un hardware independiente de bajo consumo que también puede utilizarse en entornos industriales. Para ello **Raspberry Pi** Utilizamos un ordenador basado en Raspberry Pi, que proporciona una plataforma ideal para aplicaciones basadas en Python que incluyen comunicación en red e interfaz web. La Raspberry Pi es un ordenador de alto rendimiento con una buena relación calidad-precio, que se adapta con flexibilidad a diversas aplicaciones. Por defecto, ejecuta Raspberry OS, una distribución basada en Debian, que cumplía plenamente con los requisitos del proyecto.



Figura 67: Raspberry Pi (Fuente: <https://malnapp.hu/raspberry-pi-5-4gb>)

La Raspberry Pi en esta arquitectura **como una capa de monitorización y visualización** Funciona como un controlador que se comunica con el subsistema basado en ESP32 a través del protocolo Modbus TCP, al tiempo que proporciona una interfaz HMI basada en web para los usuarios.

Instalación del sistema operativo y del sistema base

La preparación del dispositivo comenzó con la instalación de un sistema operativo Linux estable y con soporte a largo plazo. Si bien es posible instalar sistemas operativos personalizados, Raspberry Pi OS proporcionó el soporte de hardware adecuado, la estabilidad del sistema y un fácil acceso al entorno de desarrollo de Python. El sistema se instaló utilizando el programa oficial Raspberry Pi Imager, que, tras especificar los ajustes de configuración básicos, proporcionó a los estudiantes un entorno de escritorio listo para trabajar.



Figura 68: Instalación de Raspberry Pi OS (Fuente: Edición propia)

Durante la instalación del sistema, ocurrió lo siguiente:

- configurar la cuenta de usuario predeterminada,
- actualización del sistema,
- y comprobando la conexión de red.

El funcionamiento fiable de la red era de suma importancia, ya que toda la lógica de monitorización se basa en la comunicación IP.

Preparación del entorno de ejecución de Python y sus dependencias.

Dado que la aplicación de monitorización está escrita en Python, se creó un entorno Python unificado y mantenible en la Raspberry Pi. El sistema incluye Python 3.

```
tanulo@raspberrypi: ~  
Fájl Szerkesztés Lapok Súgó  
tanulo@raspberrypi:~$ python3  
Python 3.9.2 (default, Feb 28 2021, 17:03:44)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Versión 1.

Durante la preparación del entorno, se hizo hincapié en:

- utilizando el gestor de paquetes de Python (PIP),
- instalación de las bibliotecas externas necesarias (servidor web, cliente Modbus),
- y evitando conflictos de versiones.

```
tanulo@raspberrypi:~$ pip install flask
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: flask in /usr/lib/python3/dist-packages (1.1.2)
tanulo@raspberrypi:~$ pip install pymodbus
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pymodbus
  Downloading https://www.piwheels.org/simple/pymodbus/pymodbus-3.8.6-py3-none-any.whl (164 kB)
    |#####| 164 kB 938 kB/s
Installing collected packages: pymodbus
  WARNING: The script pymodbus.simulator is installed in '/home/tanulo/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pymodbus-3.8.6
tanulo@raspberrypi:~$
```

Este paso llamó la atención de los estudiantes sobre la reproducibilidad y la mantenibilidad de los sistemas de software.

Configuración de red y conexión Modbus TCP

El requisito fundamental para el funcionamiento del sistema es una conexión Modbus TCP estable entre la Raspberry Pi y el ESP32. Para lograrlo, la Raspberry Pi operó con una dirección IP fija en un entorno de red predecible, de modo que la aplicación de monitorización siempre pudiera acceder al subsistema.

Los estudiantes comprendieron que la fiabilidad de la comunicación en red es crucial para un sistema de vigilancia.

En este contexto, la conexión Modbus no se utilizó para el control de bajo nivel, sino para la consulta de estado y la transmisión de comandos, lo que encaja bien con el concepto del ciclo del PLC de software.

Gestión del lanzamiento de aplicaciones y del funcionamiento en paralelo

Cuando se inicia el programa de monitorización, la lógica para realizar actualizaciones de estado cíclicas se ejecuta en un hilo independiente, mientras que el servidor web basado en Flask gestiona las solicitudes de los usuarios. Esta solución garantiza que la interfaz web no bloquee las funciones de monitorización del sistema.

Los recursos de la Raspberry Pi resultaron ser suficientes:

- para comunicación Modbus continua,
- para el procesamiento de estados cíclicos,
- y para dar servicio a la interfaz hombre-máquina web.

Este modelo operativo demostró claramente a los estudiantes la diferencia entre el procesamiento paralelo y el funcionamiento determinista, aunque no en tiempo real.

Experiencia en educación e integración de sistemas

El desarrollo del entorno de ejecución basado en Raspberry Pi fue fundamental para las etapas previas del proyecto. Los estudiantes pudieron observar un sistema completo en funcionamiento, donde los sensores y actuadores se presentan a nivel físico, la comunicación se realiza mediante protocolos industriales y la monitorización y el control se implementan utilizando herramientas de software.

Además, el sistema operativo basado en Debian brindó a los estudiantes la oportunidad de profundizar sus conocimientos de Linux. Dado que la configuración de estos dispositivos en un entorno industrial no se realiza necesariamente a través de una interfaz gráfica, nuestro objetivo era que los estudiantes trabajaran lo máximo posible mediante una interfaz de línea de comandos.

Sin embargo, durante la implementación práctica, los estudiantes y yo alcanzamos las limitaciones de hardware de la Raspberry Pi. Cuando todo el equipo de estudiantes se conectó a los servidores web simultáneamente, se produjo una ralentización en el procesamiento de datos. Además, la carga adicional habría requerido la instalación de refrigeración activa, ya que el microcontrolador se reinició varias veces debido al sobrecalentamiento. Si bien cumplió con los requisitos del proyecto, sirvió como una lección importante durante la implementación: en un sistema de nivel industrial, planificar con anticipación los recursos del servidor y evaluar la carga prevista debe ser una prioridad.

6.9. El papel del cliente Python

El sistema de monitorización ESP Modbus, el programa en Python, tiene como objetivo implementar la lógica de monitorización y la "sombra" digital.

Conectando con ESP

```
1 # ===== FLASK + MODBUS TCP PLC RÉTEG =====
2 # Ez a program egy "szoftveres PLC ciklust" valósít meg.
3 # Feladata:
4 #   • ESP32 Modbus eszköz lekérdezése
5 #   • LED vezérlési logika kezelése
6 #   • Szenzoradatok továbbítása a webes HMI felé
7
8
9 # ===== MODULOK IMPORTÁLÁSA =====
10 from flask import Flask, render_template, jsonify
11 from pymodbus.client import ModbusTcpClient
12 import threading, time
13
14
15 # ===== WEB SZERVER OBJEKTUM =====
16 # Flask kezeli a HTTP kéréseket (HMI felület)
17 app = Flask(__name__)
18
19
20 # ===== MODBUS KLIENS (ESP32 FELÉ) =====
21 # Ez a sor létrehozza a TCP kapcsolatot az ESP Modbus felügyeleti rendszerével.
22 esp = ModbusTcpClient("192.168.0.200", port=502)
23
24
25 # ===== ÁLLAPOTVÁLTOZÓK LÉTREHOZÁSA =====
26 # Ezek a változók a rendszer állapotát tükrözik a szerver oldalon
27
28 led_state = False      # LED logikai állapot (kimenet)
29 btn_state = False      # Fizikai gomb állapota
30 prev_btn_state = False # Előző ciklus gomb állapota (élelérzékeléshez)
31
32 temperature = 0        # Hőmérséklet cache
33 humidity = 0           # Páratartalom cache
```

Estructura del ciclo de supervisión

```
37 # ===== PLC CIKLUS (IPARI LOGIKA MINTAJÁRA) =====
38 # Ez a függvény a felügyeleti alkalmazás ciklikusan futó központi része.
39 # Bár a neve „felügyeleti ciklus”, funkcionálisan nem valós idejű vezérlési ciklust, hanem egy
    állapotfrissítő, felügyeleti lekérdezési ciklust valósít meg.
40 # Ez egy végtelen ciklus, amely fix periódusban (200 ms) végzi az I/O beolvasást és a vezérlési logikát.
41
42 def plc_cycle():
43     global led_state, btn_state, prev_btn_state, temperature, humidity
44
45     while True:
46
47         # ===== DIGITÁLIS BEMENET OLVASÁSA (FIZIKAI GOMB) =====
48         # Modbus Input Status regiszter 0
49         # Az ESP az aktuális gombállapotot a Discrete Input 0 címre írja.
50         rr_btn = esp.read_discrete_inputs(address=0, count=1)
51
52         if not rr_btn.isError():
53             btn_state = rr_btn.bits[0]
54
55
56         # ===== ÉLÉRZÉKELÉS (EDGE TRIGGER) =====
57         # Ez a PLC logika:
58         #   # ha a gomb éppen most lett lenyomva – LED állapot vált,
59         #   # a parancs visszakerül az ESP-re
60         # Ipari PLC logikában ez "RISING EDGE DETECT".
61         # Csak akkor reagálunk, amikor a gomb 0-1 állapotba vált
62
63         if btn_state and not prev_btn_state:
64             led_state = not led_state # LED állapot váltás
65             esp.write_coil(0, led_state) # Fizikai LED frissítés Modbuson
66
67         prev_btn_state = btn_state
68
69
70         # ===== ANALÓG ADATOK OLVASÁSA =====
71         # Holding Register 0 – hőmérséklet
72         # Holding Register 1 – páratartalom
73         # Az ESP tizedes pontossággal küldi az adatot.
74
75         rr_temp = esp.read_holding_registers(address=0, count=2)
76
77         if not rr_temp.isError():
78             temperature = rr_temp.registers[0] / 10.0
79             humidity = rr_temp.registers[1] / 10.0
80
81
82         # ===== PLC CIKLUS IDŐZÍTÉS =====
83         # 200 ms ciklusidő (ipari PLC mintára)
84         time.sleep(0.2)
```

Pantalla web (HMI)

```
88 # ===== WEB OLDAL =====
89 # A HMI felület betöltése
90
91 @app.route("/")
92 def index():
93     return render_template("index.html")
94
95
96 # ===== ÁLLAPOT API =====
97 # A webes felület innen kapja az aktuális rendszerállapotot
98
99 @app.route("/status")
100 def status():
101     return jsonify({
102         "led": led_state,      # LED állapot
103         "btn": btn_state,     # Gomb állapot
104         "temp": temperature,  # Hőmérséklet
105         "hum": humidity       # Páratartalom
106     })
107
108
109 # ===== VIRTUÁLIS GOMB (WEB HMI) =====
110 # Ez a fizikai gomb funkcióját utánozza a weboldalon keresztül
111
112 @app.route("/toggle_led", methods=["POST"])
113 def toggle_led():
114     global led_state
115
116     led_state = not led_state      # LED állapot váltás
117     esp.write_coil(0, led_state)   # Parancs küldése az ESP-nek
118
119     return "OK"
120
121
122
123 # ===== PROGRAM INDÍTÁS =====
124 if __name__ == "__main__":
125
126     # PLC ciklus külön szálon fut, hogy a web szerver ne blokkolódjon
127     threading.Thread(target=plc_cycle, daemon=True).start()
128
129     # Flask webservert indítása
130     app.run(host="0.0.0.0", port=5000)
131
```

6.10. Registro de datos de sensores en una base de datos SQL

Como siguiente paso en el desarrollo del sistema de monitorización, nos propusimos no solo mostrar los datos de los sensores en tiempo real en la interfaz HMI web, sino también registrarlos en un formato estructurado y con capacidad de búsqueda. Para lograrlo, se creó una integración de base de datos SQL que complementa la aplicación Python existente.

Este avance añadió una nueva dimensión al proyecto, ya que el sistema ahora no solo realiza funciones de monitorización, sino también de recopilación de datos, lo que permite su posterior análisis.

El papel del registro de datos en el sistema de vigilancia

Almacenar los datos de los sensores en una base de datos permite realizar un seguimiento de los cambios a lo largo del tiempo, un análisis a largo plazo del funcionamiento y el posterior procesamiento de los datos. Esto demostró claramente a los estudiantes que, en sistemas reales, mostrar únicamente el estado actual no es suficiente; el almacenamiento histórico de datos es un requisito fundamental.

Durante el desarrollo, se hizo hincapié en comprender que la recopilación de datos no tiene fines de control continuo, sino que crea una base para el análisis y la toma de decisiones.

Selección de bases de datos y estructura lógica

Se creó una estructura de base de datos sencilla, basada en SQL, adaptada al entorno educativo. Esta estructura es fácil de usar y, a la vez, ilustra claramente los principios de la gestión de datos relacionales. La base de datos registra los datos medidos en una única tabla bien definida. Para ello, utilizamos el motor de base de datos MariaDB (basado en MySQL).

Los datos almacenados son típicamente (**datos del sensor** mesa):

- identificador de grabación (**ID de datos**, int, autoincremento, clave primaria)
- marca de tiempo (**tiempo de medición**, fecha y hora, valor predeterminado current_timestamp)
- valores de temperatura (**paso**, doble),
- datos de humedad (**tararear**, doble),

Esta estructura permite la recuperación cronológica de datos y un procesamiento estadístico sencillo.

Complementa tu aplicación Python con la gestión de bases de datos.

Durante la expansión de la aplicación existente basada en Flask, la gestión de la base de datos se integró como una unidad lógica independiente. Los datos de los sensores se siguen leyendo cíclicamente mediante comunicación Modbus TCP, pero los valores actualizados también se introducen en la base de datos.

Si bien en la implementación actual los datos se envían continuamente a la base de datos, les hicimos notar a los estudiantes que, en un sistema complejo, las distintas funciones operan con diferentes niveles de criticidad temporal. En base a esto, el siguiente paso en el desarrollo de la aplicación podría ser reducir la frecuencia de registro de datos a tan solo cada pocos segundos.

Para ampliar el programa Python, primero fue necesario importar el módulo mysql.connector.

```
9 # ===== MODULOK IMPORTÁLÁSA =====
10 from flask import Flask, render_template, jsonify
11 from pymodbus.client import ModbusTcpClient
12 import threading, time
13 import mysql.connector #Az SQL kapcsolatot kezelő modul importálása.
```

En el siguiente paso, creamos los objetos necesarios para gestionar la conexión a la base de datos y enviar los datos.

```
26 # ===== ADATBÁZIS KAPCSOLAT KONFIGURÁLÁSA =====
27 # Az adatbázis kapcsolatot kezelő objektum létrehozása.
28 db = mysql.connector.connect(
29     host="localhost",
30     port="3306",
31     user="root",
32     password="",
33     database="sensordatabase"
34 )
35
36 #Az adstbázist kezelő cursor objektum létrehozása.
37 dbcursor = db.cursor()
```

Finalmente, al consultar los datos de los sensores analógicos, ampliamos el código enviando los datos al servidor de la base de datos:

```
85 # ===== ANALÓG ADATOK OLVASÁSA =====
86 # Holding Register 0 - hőmérséklet
87 # Holding Register 1 - páratartalom
88 # Az ESP tizedes pontossággal küldi az adatot.
89
90 rr_temp = esp.read_holding_registers(address=0, count=2)
91
92 if not rr_temp.isError():
93     temperature = rr_temp.registers[0] / 10.0
94     humidity = rr_temp.registers[1] / 10.0
95     dbcursor.execute("INSERT INTO sensordata (temp,hum) VALUES (" + temperature + ","+humidity+")")
96
97
98 # ===== PLC CIKLUS IDŐZÍTÉS =====
99 # 200 ms ciklusidő (ipari PLC mintára)
100 time.sleep(0.2)
```

Importancia educativa y experiencias estudiantiles

La integración de la base de datos SQL amplió significativamente el valor didáctico del proyecto. Los estudiantes no solo aprendieron sobre la gestión de datos en tiempo real, sino también sobre el almacenamiento de datos estructurados y su posterior uso.

Durante el desarrollo, se produjo lo siguiente:

- conocimientos básicos de gestión de datos y modelado de datos,
- pensamiento sistémico,
- y sensibilidad en materia de seguridad y coherencia de los datos.

El registro de los datos de los sensores en una base de datos se integró de forma orgánica en el proyecto en su conjunto y reforzó aún más la idea de que la base de los sistemas informáticos e industriales modernos es la gestión consciente y estructurada de los datos.

6.11. Capa de gestión web: su función en el sistema

La aplicación Python proporciona acceso a los datos a través de puntos finales HTTP, mientras que la interfaz HTML + JavaScript que se ejecuta en el navegador se encarga de la visualización y la interacción con el usuario.

Esta capa:

- no controla directamente la E/S física
- muestra una sombra digital del estado del ESP
- reenvía los comandos del usuario a la aplicación Python.

La ruta de los datos:



Bloque de estado LED

Este botón no controla directamente el ESP, sino el sistema de gestión de Python.

```
11 <!-- ===== LED KÁRTYA ===== -->
12 <!-- Ez a blokk a LED állapotát mutatja és innen lehet vezérelni -->
13 <div class="card">
14 <h2>LED állapot</h2>
15
16 <!-- A LED vizuális visszajelzője
17 A színét a JavaScript módosítja:
18 zöld = világít
19 piros = nem világít -->
20 <div id="led" class="led"></div>
21
22 <!-- Virtuális kapcsoló gomb
23 onclick esemény - toggleLED() JS függvény fut le
24 Ez HTTP POST kérést küld a Flask szervernek (/toggle_led)
25 Ez a gomb nem közvetlenül az ESP-t vezérli, hanem a Python felügyeleti rendszert.-->
26 <button onclick="toggleLED()">LED kapcsolás</button>
27 </div>
```

Bloque de estado del botón

```
30 <!-- ===== FIZIKAI GOMB ÁLLAPOT ===== -->
31 <!-- Ez CSAK kijelzés, nem vezérlő szerv -->
32 <div class="card">
33 <h2>Gomb állapot</h2>
34
35 <!-- A fizikai gomb állapotának szöveges kijelzése.
36 A Flask szerver PLC ciklusa frissíti.
37 "Nyomva" / "Nincs nyomva" szöveg jelenik meg -->
38 <div id="btn">Nincs nyomva</div>
39 </div>
40
```

Visualización en texto del estado del botón físico.

Bloque de sensores

```
42 <!-- ===== KÖRNYEZETI ADATOK ===== -->
43 <!-- DHT11 szenzor által mért adatok -->
44 <div class="card">
45 <h2>Környezeti adatok</h2>
46
47 <!-- Hőmérséklet kijelzés
48 Ezek a DHT szenzor értékei számára fenntartott mezők.
49 A program ide írja be a /status API-ből érkező értéket. -->
50 <div>🌡 Hőmérséklet: <span id="temp">--</span> °C</div>
51
52 <!-- Páratartalom kijelzés -->
53 <div>💧 Páratartalom: <span id="hum">--</span> %</div>
54 </div>
```

Estos campos están reservados para los valores del sensor DHT.

JavaScript - actualización en vivo

```
1 // ===== ÁLLAPOTFRISSÍTŐ FÜGGVÉNY =====
2 // Ez a függvény 200 ms-onként lefut.
3 // Feladata: lekérdezni a Flask szervertől az aktuális rendszerállapotot és frissíteni a weboldalon
  látható adatokat.
4
5 function update(){
6
7 // HTTP GET kérés a szerver /status végpontjára
8 // A szerver JSON formátumban küldi vissza az adatokat:
9 // { led: bool, btn: bool, temp: float, hum: float }
10 fetch("/status")
11
12 // A válasz átalakítása JSON objektummá
13 .then(r => r.json())
14
15 // A kapott adatok feldolgozása
16 .then(data => {
17
18 // ===== LED VIZUÁLIS ÁLLAPOT =====
19 // A LED kör színének beállítása:
20 // zöld ha világít (true), sötétpiros ha nem (false)
21 document.getElementById("led").style.background =
22 data.led ? "limegreen" : "darkred";
23
24
25 // ===== FIZIKAI GOMB ÁLLAPOT KIÍRÁS =====
26 // A Flask PLC ciklus által olvasott bemenet jelenik meg
27 document.getElementById("btn").innerText =
28 data.btn ? "Nyomva" : "Nincs nyomva";
29
30
31 // ===== HŐMÉRSÉKLET KIÍRÁS =====
32 // DHT11 szenzor adata az ESP - Modbus - Flask útvonalon
33 document.getElementById("temp").innerText = data.temp;
34
35
36 // ===== PÁRATARTALOM KIÍRÁS =====
37 document.getElementById("hum").innerText = data.hum;
38 });
39 }
```

JavaScript recupera y actualiza estas variables.

Funcionamiento del botón del interruptor LED

```
43 // ===== VIRTUÁLIS LED KAPCSOLÓ =====
44 // Ez a függvény akkor fut le, amikor a felhasználó megnyomja a weboldalon a "LED kapcsolás" gombot.
45
46 function toggleLED(){
47
48 // HTTP POST kérés küldése a szervernek
49 // A Flask oldalon ez a /toggle_led útvonalat hívja meg, ami megfordítja a LED állapotát és kiírja
  Modbus-on az ESP-re.
50 fetch("/toggle_led", { method: "POST" });
51
52 // Fontos: itt nincs válaszfeldolgozás, a következő update() ciklus már lekéri az új állapotot.
53 }
```

Proceso:



Ciclo de actualización

```
57 // ===== IDŐZÍTETT FRISSÍTÉS =====
58 // A böngésző 200 ms-onként automatikusan meghívja az update() függvényt.
59 // Ez kvázi PLC ciklus a weboldalon (HMI polling).
60 // Ez megfelel a Python programban lévő ciklus ütemének.
61
62 setInterval(update, 200);
```

Esto coincide con la frecuencia de ciclo de Python, por lo que el sistema se mantiene sincronizado.

El papel del estilo en el sistema

CSS garantiza que la interfaz web HMI:

- ser legible incluso en condiciones de poca luz.
- Proporcionar información visual clara sobre los estados.
- bloques de información separados (lógica del panel HMI)

Esta estructura sigue el enfoque minimalista de SCADA / HMI industrial (Interfaz Hombre-Máquina).

CSS completo - comentado en detalle

```
1  /* ===== OLDAL ALAPSTÍLUS ===== */
2  body {
3      font-family: Arial;
4      background: linear-gradient(135deg,#0f2027,#203a43,#2c5364);
5      color: white;
6      text-align: center;
7      margin: 0;
8  }
9
10 /* ===== KÁRTYA (PANEL) STÍLUS =====
11 Minden információs blokk ezt használja:
12 LED, Gomb, Szenzor adatok*/
13 .card {
14     background: #222;
15     margin: 20px auto;
16     padding: 20px;
17     width: 320px;
18     border-radius: 12px;
19     box-shadow: 0 0 10px #000;
20 }
21
22 /* ===== LED VISSZAJELZŐ KÖR =====
23 Ez a vizuális LED indikátor*/
24 .led {
25     width: 40px;
26     height: 40px;
27     margin: 15px auto;
28     border-radius: 50%;
29     background: darkred; /* Alapértelmezett állapot: LED kikapcsolva. A program dinamikusan felülírja:
30 limegreen = bekapcsolva */
31 }
32
33 /* ===== GOMB STÍLUS =====
34 Virtuális LED kapcsoló*/
35 button {
36     padding: 10px 15px;
37     border-radius: 8px;
38     border: none;
39     background: #00aaff;
40     color: white;
41     cursor: pointer;
42 }
```

Esto se corresponde con los principios de visualización de las interfaces hombre-máquina (HMI) de PLC reales, los terminales SCADA y los paneles de control industrial.

¿Por qué es esto importante en la educación?

Los estudiantes no solo visualizan datos, sino que también aprenden cómo una HMI (Interfaz Hombre-Máquina) puede:

- sistema visual funcional
- donde el color, el tamaño y la disposición tienen significado

La capa web ya está completa:



6.12. Desarrollo por parte de los estudiantes de la interfaz hombre-máquina web en un entorno de trabajo en equipo.

Tras implementar las funciones básicas del sistema de monitorización, el siguiente paso del proyecto fue el desarrollo de la interfaz hombre-máquina (HMI) web. En esta fase, los estudiantes dejaron de implementar soluciones predefinidas y, en cambio, desarrollaron la interfaz existente basada en HTML y CSS a partir de sus propias ideas y en trabajo en equipo.

El objetivo de la tarea era que los estudiantes experimentaran cómo una interfaz funcional pero de aspecto básico puede hacerse más fácil de usar, más clara y más informativa visualmente, manteniendo al mismo tiempo la estabilidad funcional del sistema.

Trabajo en equipo y reparto de tareas

Los estudiantes trabajaron en pequeños grupos, donde la división de tareas fue deliberada. Algunos se centraron principalmente en transformar la estructura (organización de elementos HTML, agrupación lógica), mientras que otros se centraron en perfeccionar la apariencia (colores, diseño, retroalimentación visual).

Esta forma de organización del trabajo permitió a los estudiantes experimentar los beneficios y los desafíos del desarrollo conjunto, así como practicar la coordinación y el desarrollo basados en el trabajo de los demás.

Mejora funcional y visual de la interfaz.

La interfaz web, que sirvió como punto de partida, ya incluía la visualización de los estados básicos del sistema y las opciones de control básicas. Los estudiantes desarrollaron aún más esta interfaz:

- La presentación de la información ha sido reorganizada,
- Ampliaron el rango de datos de los sensores mostrados, preparándose para un desarrollo posterior.
- Se desarrolló una retroalimentación visual más clara.
- y el estilo visual se modificó según las necesidades de su equipo.

Durante las modificaciones, se prestó especial atención a la claridad y la usabilidad, sobre todo para garantizar que la interfaz sea fácil de interpretar en un entorno de monitorización real.

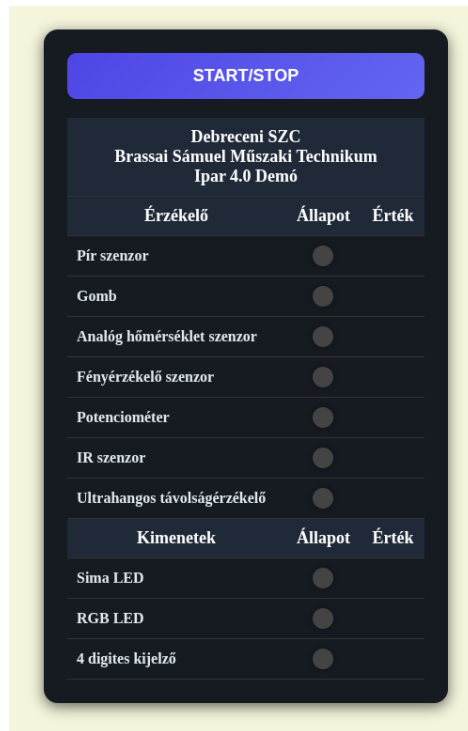


Figura 69: Trabajo mejorado de los estudiantes 1. (Fuente: Edición propia)

Surgimiento de necesidades individuales y soluciones creativas

Durante el proceso de desarrollo, los estudiantes pudieron dar forma a la apariencia y la funcionalidad de la interfaz según sus propias ideas. Como resultado, se crearon diversas soluciones, por ejemplo:

- Codificación de colores diferente para los indicadores de estado,
- diseños alternativos para mostrar datos de sensores,
- retroalimentación más pronunciada durante las intervenciones.

Esta libertad brindó la oportunidad para que floreciera la creatividad, al tiempo que los estudiantes aprendían que las ideas individuales debían equilibrarse con las limitaciones técnicas y funcionales del sistema.

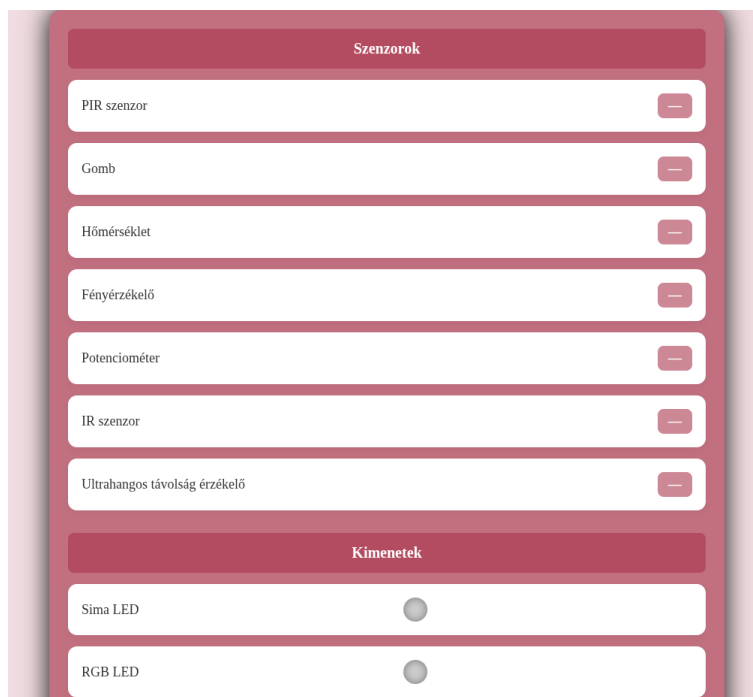


Figura 70: Trabajo mejorado de los estudiantes 2. (Fuente: Edición propia)

Importancia educativa y experiencias de desarrollo

Durante el desarrollo posterior de la interfaz HMI web, los estudiantes aplicaron sus conocimientos previamente adquiridos de manera compleja. La tarea se desarrolló simultáneamente:

- competencias digitales y de TI,
- habilidades de cooperación y comunicación,
- y una mentalidad centrada en el usuario.

El proceso estuvo bien alineado con los objetivos del proyecto, ya que los estudiantes no solo crearon soluciones técnicas, sino que también pensaron en la interfaz como un sistema en funcionamiento, considerando su papel en la arquitectura general de vigilancia.

6.13. Resumen: integración del sistema y resultados del aprendizaje

Las actividades presentadas en este capítulo constituyeron una de las fases más complejas y de mayor valor pedagógico del proyecto. En esta fase, los estudiantes implementaron un sistema de vigilancia completo y funcional en el que los elementos de hardware, la comunicación en red, la lógica de control del software y la visualización web funcionaban como un sistema unificado.

La preparación del entorno de ejecución basado en Raspberry Pi y el despliegue de la aplicación basada en Flask brindaron a los estudiantes la oportunidad de aplicar sus habilidades de programación y operación de sistemas en entornos industriales y de TI reales. El uso de la comunicación Modbus TCP fue una experiencia particularmente importante, ya que los estudiantes pudieron

Pudieron trabajar con un protocolo estándar de la industria y comprender su función en una arquitectura de vigilancia.

Durante el desarrollo de la interfaz HMI web, los estudiantes no solo crearon soluciones técnicas, sino que también dominaron un enfoque centrado en el usuario. En el trabajo en equipo, se fomentaron la creatividad individual, la toma de decisiones independiente y la consulta profesional conjunta. La personalización de las interfaces contribuyó a que los estudiantes sintieran que el sistema era suyo y asumieran la responsabilidad de su funcionamiento.

Desde una perspectiva pedagógica, cabe destacar que los estudiantes recibieron retroalimentación continua sobre las consecuencias de sus decisiones a lo largo de todo el proceso de desarrollo. Probar el funcionamiento del sistema, identificar y corregir errores, y evaluar conjuntamente las soluciones fortaleció la autorreflexión y el pensamiento crítico. Abordar los errores no se consideró un fracaso, sino una parte natural del proceso de aprendizaje.

Las actividades implementadas en el Capítulo 6 contribuyeron al desarrollo de varias competencias clave, en particular:

- competencia digital,
- habilidades de colaboración y comunicación,
- pensamiento algorítmico y sistémico,
- y en los ámbitos del aprendizaje independiente y la responsabilidad.

En resumen, este capítulo es un buen ejemplo de cómo un enfoque práctico y basado en proyectos puede conectar el conocimiento teórico con un entorno de aplicación real. Los estudiantes no solo crearon un sistema técnico funcional, sino que también adquirieron experiencias que sentarán las bases para su desarrollo profesional y su orientación laboral a largo plazo.

7. Resumen

El documento presenta un proyecto educativo complejo, basado en la Industria 4.0 y el IoT, centrado en el desarrollo de un modelo de hogar/granja inteligente. Según la filosofía del proyecto, el aprendizaje no es una recepción pasiva, sino un proceso creativo activo, donde los estudiantes encuentran soluciones a través de problemas reales. El énfasis no está en las respuestas prefabricadas, sino en el proceso de pensamiento, las decisiones y la iteración. Los estudiantes no solo utilizan la tecnología, sino que la diseñan, analizan y desarrollan, experimentando que el funcionamiento de un sistema siempre es el resultado de compromisos y decisiones conscientes.

Los objetivos educativos, por consiguiente, van más allá del conocimiento técnico clásico. El proyecto desarrolla simultáneamente conocimientos de programación y electrónica, pensamiento sistémico y habilidades de colaboración. Los estudiantes aprenden a interpretar los fenómenos del mundo físico en sistemas digitales, a reconocer relaciones de causa y efecto y a resolver problemas complejos en varias etapas. Es especialmente importante que el proyecto brinde la oportunidad de involucrar a estudiantes con diferentes perfiles, lo que fortalece el pensamiento explicativo y la resolución conjunta de problemas.

Desde una perspectiva técnica, el proyecto busca crear un sistema basado en IoT donde sensores, controladores y actuadores trabajen conjuntamente como un sistema lógico unificado. El punto de partida es un kit de granja inteligente fabricado en serie, que, sin embargo, solo sirve como base: la tarea de los estudiantes no consiste en reproducirlo, sino en comprender el sistema y luego desarrollarlo hasta crear su propia arquitectura orientada a la industria. Este proceso refleja fielmente el trabajo de ingeniería en el mundo real, donde se crean sistemas nuevos y optimizados tras analizar las soluciones existentes.

La implementación se lleva a cabo mediante un proceso claramente estructurado pero flexible, con hitos clave claramente identificables. El proceso de aprendizaje no es lineal, sino que generalmente sigue los siguientes pasos:

- aprender y ensamblar el sistema de fábrica (Kit de Granja Inteligente),
- análisis del funcionamiento de los componentes individuales (sensores, actuadores, control),
- revelando las limitaciones y deficiencias del sistema,
- definir objetivos y funciones de desarrollo propios,
- diseño de elementos mecánicos individuales y piezas de carcasa (por ejemplo, modelado 3D),
- producción de elementos físicos (por ejemplo, impresión 3D),
- rediseño y ampliación del sistema electrónico,
- programación (basada en bloques → basada en texto),
- integración de subsistemas en un único sistema,
- Pruebas, depuración y desarrollo iterativo.

Este proceso no solo da como resultado un avance técnico, sino que también enseña a los estudiantes que un sistema complejo nunca está "terminado", sino que está en constante evolución y perfeccionamiento.



Figura 71: Hogar inteligente (Fuente: Imagen propia, fondo generado por IA)

Las dificultades surgidas durante el proyecto —la falta de conocimientos básicos, la dificultad para comprender las diferencias entre señales analógicas y digitales, y la resolución de problemas— se transformaron en valiosas experiencias de aprendizaje al finalizar el proceso. Los problemas técnicos, como los datos de medición con ruido o el funcionamiento inestable debido a valores límite inadecuados, ayudaron a los estudiantes a comprender que el comportamiento del sistema es consecuencia directa de las decisiones de diseño.

Al finalizar el proyecto, los estudiantes eran capaces de pensar en sistemas, interpretar la lógica de entrada-salida y comprender la relación entre el mundo físico y el digital. Las soluciones creadas eran sistemas funcionales, contruidos de forma consciente.

La lección más importante es que el proyecto no solo proporcionó conocimientos técnicos, sino que también desarrolló un enfoque que puede aplicarse directamente en el entorno industrial y tecnológico moderno.

Financiado por la Unión Europea. La información y las declaraciones aquí contenidas representan las opiniones de los autores y no reflejan necesariamente la opinión oficial de la Unión Europea ni de la Fundación Pública Tempus. Ni la Unión Europea ni la entidad financiadora se hacen responsables de su contenido.