Plataforma para la automatización de ensayos de una red de dispositivos electrónicos con comunicación LIN

Joel Schimansky Roca

Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú. Ingeniero en Electrónica Industrial y Automática

Resumen

El objetivo de este proyecto de final de grado es crear una plataforma que permita la automatización de ensayos dentro de una red LIN (Local Interconnect Network) con diferentes dispositivos conectados. La plataforma de ensayos será controlada por un maestro LIN el cual se comunicará mediante Ethernet-TCP/IP con el ordenador y será capaz de enviar mensajes LIN a los esclavos LIN consiguiendo así poder controlar las placas de control que tienen los relés y pueden realizar lecturas de voltaje, corriente y potencia de la carga conectada. Desde el ordenador se ejecutará un software de QT para tener una interfaz gráfica y así poder seleccionar ensayos y ejecutarlos de forma automática para que al terminar generen un informe con los resultados.
Para la plataforma se ha tenido que desarrollar:
Diseño Hardware LIN y Relés: En el proyecto de la plataforma se han diseñado dos placas de hardware una para la comunicación LIN (Local Interconnect Network) usando el transceptor ATA663211 de Microchip y otra para realizar lecturas de corriente, voltaje y potencia del equipo conectado a la plataforma usando el INA219 de Texas Instrument.
Maestro LIN, comunicación Raspberry Pi Pico – PC: La plataforma se controla desde un ordenador conectado un cable ethernet a la Waveshare RP2040 maestro LIN esta comunicación usa el protocolo TCP/IP. El maestro LIN convierte los mensajes recibidos por TCP/IP a LIN y los envía a los esclavos usando la placa de comunicaciones LIN.
Esclavo LIN: Este dispositivo recibe los mensajes LIN de la red usando también la placa de comunicaciones LIN y realiza acciones sobre la placa de control.
Software UI (User Interface): Desde el ordenador se ejecuta un software UI (User Interface) que usa el protocolo TCP/IP y envía comandos para controlar la plataforma. A parte el software contiene un selector de ensayos (tests) los cuales se ejecutan sobre los equipos conectados
Informe de resultado: Al terminar los ensayos ejecutados en la plataforma se crean unos informes con los resultados del ensayo (Report) usando una plantilla html y Jinja2, el informe creado tiene formado PDF.
Juntando todos los datos explicados anteriormente se consigue una plataforma de ensayos.

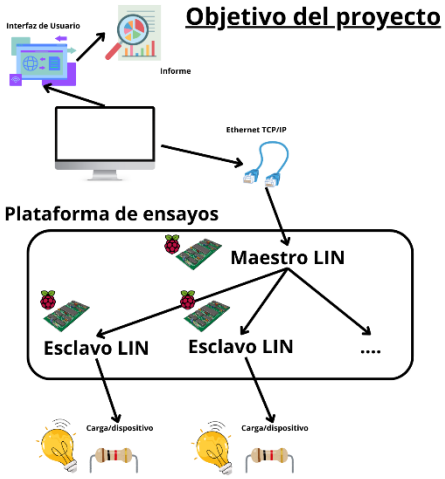
Palabras clave: Automatización, Python, Raspberry Pi Pico, hardware, LIN, Test, Pyside6(QT), Report.

Introducción

En el mundo actual se tiende a automatizar todos los procesos para ahorrar tiempo, se intentan crear maneras de poder realizar tareas sin tener una persona pendiente del proceso, para ahorrar en costes y mejorar en tiempos. La Plataforma para la automatización de ensayos de una red de dispositivos electrónicos con comunicación LIN pretende cubrir esta necesidad, pudiendo realizar ensayos de forma automática y obteniendo un informe al terminar el proceso.

Objetivo

El objetivo de este proyecto de final de grado es crear una plataforma que permita la automatización de ensayos dentro de una red LIN con diferentes dispositivos conectados. En la imagen se puede observar la estructura y el formato que tendrá el diseño del proyecto, donde desde un ordenador se puede controlar la plataforma de ensayos y esta se comunica internamente por protocolo LIN.



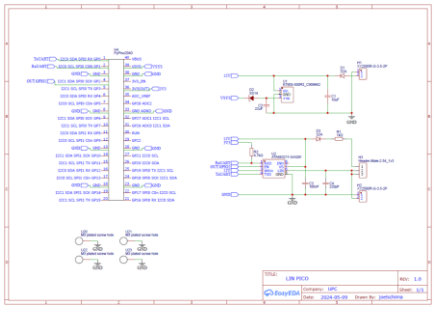
La plataforma de ensayos está dentro de una red de dispositivos LIN la cual está controlada desde un ordenador. Los microcontroladores utilizados para la plataforma son dispositivos Raspberry Pi Pico y Waveshare RP2040-Eth el cual usa el mismo diseño que las Raspberry Pi Pico y nos proporciona la posibilidad de usar cable Ethernet y comunicaciones TCP/IP. Los microcontroladores estarán en una red LIN con un Maestro LIN el cual controlará varios Esclavos LIN, en nuestro caso serán dos Esclavos LIN, pero el número de dispositivos puede variar según las necesidades del equipo a probar, esto nos proporciona una plataforma con escalabilidad.

Desarrollo Comunicación LIN

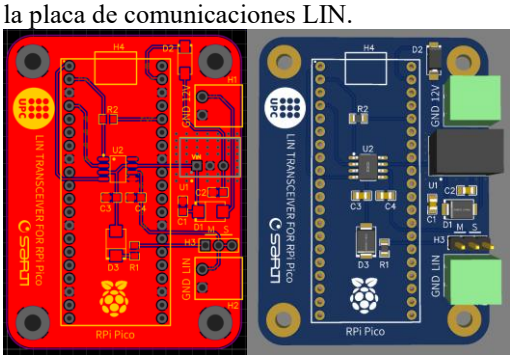
Para las comunicaciones internas, la plataforma utiliza el protocolo de comunicación LIN. En este proyecto se usa un maestro LIN y dos esclavos LIN. El maestro LIN gobierna sobre los dos esclavos y es el que gestiona la comunicación y manda comandos y peticiones a los esclavos. El protocolo LIN es barato de implementar dado que solo necesita dos cables y los transceptores LIN son muy asequibles.

El microcontrolador elegido para la plataforma es la Raspberry Pi Pico [1] la cual tiene como sistema de comunicación la UART, I2C, SPI. Se ha desarrollado una placa de comunicaciones LIN que es compatible con Raspberry Pi Pico para que la plataforma pueda comunicarse mediante LIN.

La placa de comunicaciones LIN desarrollada usa el transceptor LIN ATA663211 de Microchip Tech [4], el cual nos permite transformar la señal del a UART a LIN y poder comunicarnos con otras placas de comunicación LIN. En este sistema todos los dispositivos internos tendrán la placa de comunicaciones LIN pudiendo así comunicarse usando este protocolo.
Dado que la placa de comunicaciones LIN se usa en todos los dispositivos también será la placa que gestione la alimentación de la Raspberry Pi Pico convirtiendo la señal de la fuente de alimentación de 12V a 5V usando un DC-DC. Este es el esquemático del circuito de la placa de comunicaciones LIN:

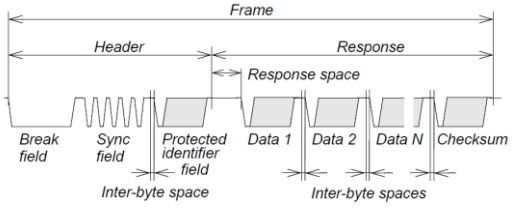


En la imagen se puede observar dos circuitos el de alimentación con el DC-DC 12V-5V K7805-500R3, este circuito está protegido con un diodo D3 S2A dado que la Raspberry Pi Pico también puede ser alimentada desde el USB y el circuito LIN el cual con la señal de la UART crea la señal LIN que trabaja a una tensión de 12V.
Este sería el diseño de la PCB y el diseño 3D de la placa de comunicaciones LIN.



En el diseño de la PCB se ha intentado conseguir un tamaño reducido, la mayor parte del espacio lo ocupa la Raspberry Pi Pico. El diseño tiene dos conectores uno para la alimentación y otro para las comunicaciones LIN. La línea LIN tiene que ser alimentada a 12V por el maestro LIN, los esclavos LIN simplemente se conectan a la línea, pero no añaden 12V. Para poder aprovechar la misma PCB y tener las dos opciones está el conector donde se puede poner un jumper, para hacer que la placa sea maestro LIN o esclavo LIN. El conector tiene una serigrafía donde pone M (maestro) y S (esclavo). El jumper en M hace que se conecten los 12V y el jumper en S no conecta los 12V.

El firmware de la Raspberry Pi Pico también se ha desarrollado para que sea compatible con la placa de comunicaciones LIN y con el protocolo de comunicaciones LIN. En este caso los pines GPIO usados por la Raspberry Pi Pico son GP0 y GP1 para la uart tx y uart rx y el GP2 para la activación del ATA633211. Ahora la Raspberry Pi Pico tiene que cumplir con las especificaciones del protocolo LIN2.1 [5], esto implica que el dispositivo tiene que ser capaz de cumplir con un formato al enviar los mensajes.



Los mensajes LIN siguen el siguiente formato, un break con un mínimo de 13 bits, seguido de un byte de sincronización 55h y el identificador que tiene que estar protegido, los bytes de datos

y el checksum. El byte del identificador protegido está formado por un numero entre 0 y 59, esto deja libre dos bits que se usan para calcular la paridad y añadirla, así se protege el identificador. La ecuación para calcular la paridad es esta:
P0 = ID0 ⊕ ID1 ⊕ ID2 ⊕ ID4
P1 = ¬ (ID1 ⊕ ID3 ⊕ ID4 ⊕ ID5)
El identificador protegido queda como en la imagen:

En el protocolo LIN hay dos tipos de mensajes uno donde el maestro no espera respuesta del esclavo y otro donde el maestro espera una respuesta del esclavo.
Para este segundo ya tendríamos completado el mensaje break + byte sync + id protegido, se mandaría este mensaje y el esclavo respondería con bytes de datos y el checksum.

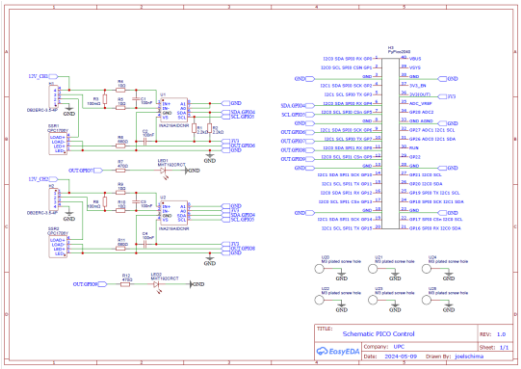
Para el primer caso donde el maestro no necesita respuesta del esclavo el formato sería: break + byte sync + id protegido + byte datos + checksum.

El cálculo del checksum [6][7] consiste en la suma invertida con el acarreo de todos los bytes de datos y el byte identificador.

El maestro LIN esta programado para que genere estos mensajes y el esclavo LIN esta programado para que pueda leer estos mensajes comprobar que son correctos y ejecutar las acciones consecuentes.

Desarrollo Placa Control

El objetivo de la plataforma de ensayos es poder realizar pruebas sobre equipos, se ha desarrollado una placa electrónica con la cual puedes conectar una carga y realizar lecturas de voltaje, corriente y potencia. Esta placa contiene dos relés [8] y dos sensores INA219 [12], esto permite conectar la carga y controlar la activación con los relés pudiendo hacer lecturas con el INA219.
Se ha querido que en el diseño de la placa siga siendo compatible con la Raspberry Pi Pico y con la placa de comunicaciones LIN. El esquemático del circuito es este:



En este esquemático tenemos dos circuitos casi simétricos que tienen el mismo objetivo. Son dos conectores de cuatro entradas donde puede conectar la alimentación 12V y GND y la carga, esta conexión para por el relé de estado sólido para así poder conectar o desconectar la alimentación y la carga esta conectada al INA219 para así poder realizar lecturas de voltaje, corriente y potencia.

El control de los relés se realiza con los pines de la Raspberry Pi Pico y también se ha añadido unos leds para poder ver visualmente cuando los relés están activados o desactivados.

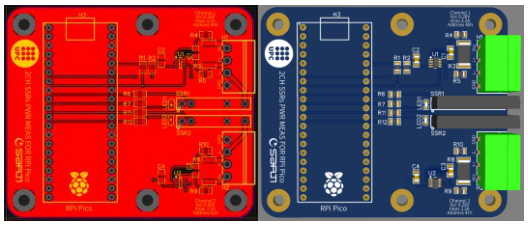
Para la configuración del INA219 se ha elegido una resistencia shunt de 0,1Ω y 2W. Esta resistencia se a elegido en base a los cálculos de la potencia disipada.

$$P_{disipada} = I^2 * R_{shunt}$$

$$I = \sqrt{\frac{P_{disipada}}{R_{shunt}}} = \sqrt{\frac{1\ W}{0.1\ \Omega}} = 3{,}1622\ A\ \cong 3{,}2A$$

El límite de corriente que puede aguantar el relé son 4A y con el calculo de potencia disipada por la resistencia con 1W es de 3,2A se ha elegido una resistencia que aguante más potencia en este caso 2W y siendo 3,2A inferior al límite del relé sea decidido que esté es el límite de amperaje que puede aguantar la placa.
El INA219 permite la comunicación I2C este es el bus de comunicaciones que se usa para poder controlarlo desde la Raspberry Pi Pico, dado que usamos dos INA219 cada sensor se le ha asignado una dirección diferente usando los pines A0 y A1. Siendo el sensor 1 dirección 40h y el sensor 2 dirección 41h.

El diseño del circuito y la PCB se pueden observar en la imagen:



El diseño de la placa tiene compatibilidad con la Raspberry Pi Pico y con la Placa de comunicaciones LIN. Los limites de la carga son de 0-26V (límite del INA219) y 3,2A límite calculado.

El firmware de las Raspberry Pi Pico se ha programado para que pueda controlar la Placa de Control y que se pueda controlar mediante el protocolo LIN.
Se usan dos pines GPIO para el bus I2C y poder comunicarse con el INA219 y 4 pines output para controlar los relés y los leds.
El firmware también contiene los identificadores de comunicación LIN que realizarán acciones sobre esta Placa de Control. Como hay dos Esclavos los identificadores con rango 30-39 son para el Esclavo 1 y los identificadores con rango 40-49 son para el Esclavo 2.

Activación/desactivación de relés:
Identificador: 30 o 40
Datos: [rele1, rele2] donde rele1 y rele2 puede ser 1 o 0 para activar o desactivar el relé.

Lectura estado de los relés:
Identificador: 31 o 41
La respuesta del Esclavo es [rele1, rele2] donde rele1 y rele2 pueden ser 1 o 0, 1 activado y 0 desactivado.

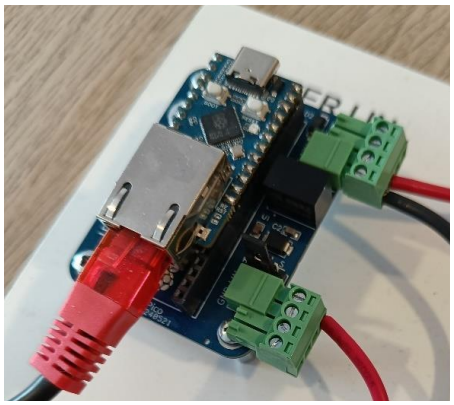
Lectura del INA219:
Identificadores: 35, 36, 45 o 46.
La respuesta del Esclavo es [voltaje, 'V', corriente, 'A', potencia, 'W'] devuelve el valor del voltaje en V (voltios), el valor de corriente en A (amperios) y el valor de potencia en W (Watios). La petición es por cada sensor por separado, 35 o 45 para el sensor 1 y 36 o 46 para el sensor 2.

Desarrollo de la comunicación TCP/IP Waveshare 2040-Eth al ordenador

Una de las características de la plataforma es que las comunicaciones internas son LIN, pero las comunicaciones de la plataforma con el ordenador son Ethernet TCP/IP. Para ello se ha usado el Waveshare RP2040-Eth [3], este dispositivo es un diseño de microcontrolador que usa el mismo que la Raspberry Pi Pico el RP2040, pero tiene un conector Ethernet y permite las comunicaciones ethernet usando el chip ch9120, en el diseño de este equipo mantiene el mismo formato que las Raspberry Pi Pico, lo que nos permite que sea compatible con nuestras placas. Waveshare tiene una librería demo donde permite crear un servidor TCP/IP en el RP2040. Para este proyecto se ha usado esa librería y se configurado con nuestras necesidades para crear este servidor.

El maestro LIN es el único dispositivo que tiene el Waveshare RP2040-Eth, el resto de Esclavos usan la Raspberry Pi Pico. Es el maestro LIN el que se podrá comunicar con el ordenador y recibir mensajes que envía a la red LIN para realizar acciones sobre los esclavos LIN.

El Waveshare RP2040-Eth se puede observar en la imagen:

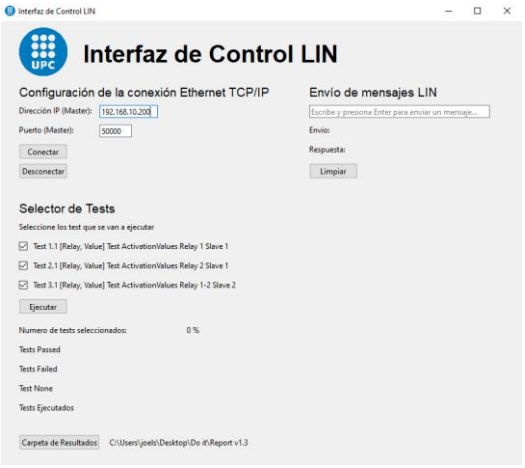


La IP seleccionada para el servidor es la 192.168.10.200 y el puerto es el 50000.
El servidor responde a los siguientes mensajes:

- help: 'The commands use follow a CSV structure, example: open,115200 tx,30,20,1 rx,31,2 close', da unos ejemplos de los comandos válidos.
- open,baudrate: Este comando activa las comunicaciones LIN, baudrate tiene que ser un valor baudrate como 115600.
- close: Este comando cierra las comunicaciones LIN.
- tx,id,data: Comando donde no se espera respuesta por parte del esclavo LIN, tx es tx, id es el identificador que queremos mandar y data son los bytes de datos.
- rx,id,readbytes: Comando donde se espera respuesta por parte del esclavo LIN, rx es rx, id es el identificador y readbytes es el número de bytes que esperamos de la lectura.

Desarrollo Software Pyside6 y Reporte Jinja2

Para facilitar al usuario la utilización de la plataforma se ha creado una interfaz de usuario con Pyside6 (Qt), el resultado se puede ver en la siguiente figura:



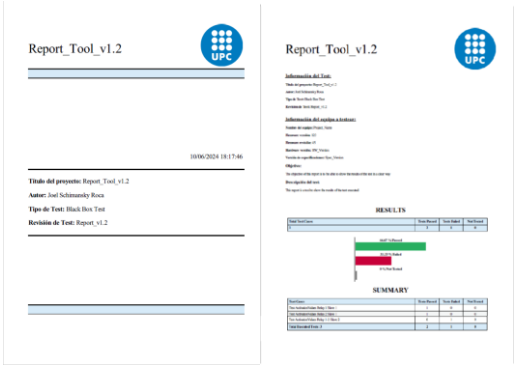
La interfaz se puede dividir en tres zonas:

- La conexión TCP/IP "Configuración de la conexión Ethernet TCP/IP", donde permite seleccionar la IP y el puerto al que te vas a conectar
- El "Envío de mensajes LIN" una vez conectado desde este recuadro se pueden mandar mensajes al Master LIN con el formato descrito anteriormente (open, close, tx, rx, help).

Estas dos zonas permiten realizar pruebas sobre la plataforma, para comprobar valores o activar o desactivar relés.

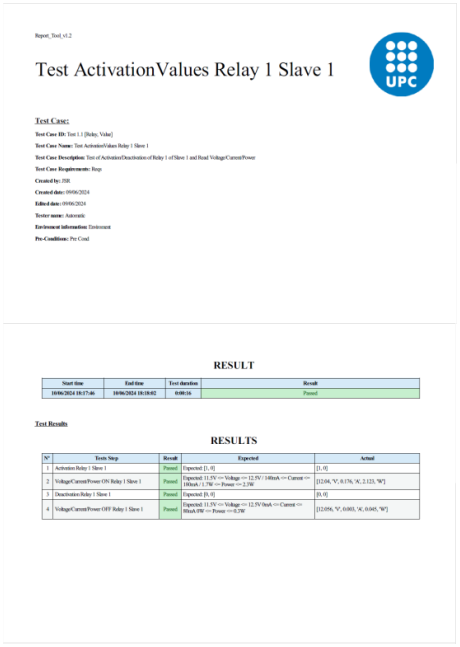
La tercera zona contiene el "Selector de Tests" donde se pueden ejecutar de forma automática Test diseñados previamente, en este caso se han diseño 3 test diferentes, el software permite seleccionar cualquier número de test y ejecutarlos. La aplicación es capaz de ejecutar los test usando multiprocessing, de esta manera la aplicación sigue pudiéndose usar. Mientras los test van terminando la aplicación va actualizando los resultados en Test Passed, Test Failed. Test None y Test ejecutados.
Una vez los test han terminado el sistema es capaz de crear un reporte con los resultados.

Este reporte está diseñado usando la librería Jinja2 que permite crear plantillas que luego son rellenadas con los resultados del test. En este caso las plantillas son creadas en html y luego convertidas en formato PDF usando la librería pdfkit. Dentro de cada test se crea un objeto report que luego se carga en la plantilla. El resultado final del reporte queda así:



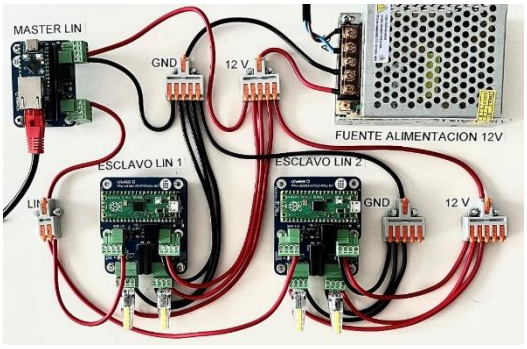
Este es el reporte principal con una portada datos sobre el equipo y proyecto del cual se realizan ensayos y contiene una lista de todos los test ejecutados y sus resultados.

También se genera un test individual por cada test creado:



En este reporte individual se puede ver los detalles del ensayo, información más detallada en que consistía la prueba y el resultado del test con los pasos. En este ejemplo todos los pasos fueron correctos y el ensayo fue Passed.

Conclusiones



El objetivo del proyecto era hacer una Plataforma para la automatización de ensayos de una red de dispositivos electrónicos con comunicación LIN y se ha conseguido. Se ha creado una plataforma capaz de ejecutar test automáticos con una comunicación interna que usa el protocolo de comunicaciones LIN y tiene tres equipos: un maestro y dos esclavos. Además, la plataforma tiene comunicación TCP/IP con el ordenador, con lo que permite mandar comandos desde el PC para controlar la plataforma. Se ha creado un software con Pyside6 para facilitar al usuario el control de la plataforma y la ejecución de ensayos y al terminar los ensayos se crea un reporte en formato PDF con los resultados.

Referencias

[1] Datasheet Raspberry Pi Pico Pinout [Online] [Consulta 26 abril 2024]. Disponible en: <https://datasheets.raspberrypi.com/pico/Pico-R3-A4-Pinout.pdf?_gl=1*rmqxmm*_ga*MTE5MzQ4OTkwOS4xNzE4Nzk2NTQ0*_ga_22FD70LWDS*MTcxODc5NjU0NC4xLjAuMTcxODc5NjU0NC4wLjAuMA>

[2] Waveshare, waveshare rp2040-Ethernet Pinout [Online] [Consulta 4 abril 2024]. Disponible en: <https://www.waveshare.com/product/raspberry-pi/boards-kits/raspberry-pi-pico-cat/rp2040-eth.htm>

[3] Waveshare, wiki RP2040-ETH [Online] [Consulta 8 mayo 2024]. Disponible en: <https://www.waveshare.com/wiki/RP2040-ETH>

[4] Datasheet ATA663211, transceptor LIN [Online] Consulta 20 mayo 2024]. Disponible en: <https://www.lcsc.com/datasheet/lcsc_datasheet_2208171700_Microchip-Tech-ATA663211-GAQW_C2934711.pdf>

[5] Especificaciones LIN 2.1, [Online] [Consulta 28 de junio 2024]. Disponible en: <https://www.lin-cia.org/fileadmin/microsites/lin-cia.org/resources/documents/LIN-Spec_Pac2_1.pdf>

[6] Calculadora de checksum y bit paridad LIN, [Online] [Consulta 30 marzo 2024]. Disponible en: <https://elearning.vector.com/mod/page/view.php?id=329>

[7] Calculadora de checksum LIN, [Online] [Consulta 30 de marzo 2024]. Disponible en: <https://linchecksumcalculator.machsystems.cz/>

[8] Datasheet Relé estado sólido, [Online] [Consulta el 10 de mayo 2024]. Disponible en: <https://www.lcsc.com/datasheet/lcsc_datasheet_2105230110_Littelfuse-IXYS-CPC1706Y_C1558848.pdf>

[9] Micropython, librería clase machine, [Online] [Consulta 16 mayo 2024]. Disponible en: <https://docs.micropython.org/en/latest/library/machine.UART.html>

[10] Jinja2 API y ejemplos, [Online] [Consulta 3 septiembre 2023]. Disponible en: <https://jinja.palletsprojects.com/en/3.1.x/>

[11] Datasheet DC-DC 12V-5V, [Online] [Consulta 21 abril 2024]. Disponible en: <https://www.lcsc.com/datasheet/lcsc_datasheet_2011180207_DEXU-Electronics-K7805-500R3_C909662.pdf>

[12] Datasheet INA219 sensor, [Online] [Consulta 24 mayo 2024]. Disponible en: <https://www.lcsc.com/datasheet/lcsc_datasheet_1811032121_Texas-Instruments-INA219AIDCNR_C87469.pdf>