Universidad Zaragoza
1542

Trabajo Fin de Grado

Grado en Ingeniería Electrónica y Automática

Diseño de control de calidad para SmartGate e implementación mediante software con generación de reportes para su industrialización.

Quality control design for SmartGate and software implementation with report generation for industrialisation.

Autor/es

Raúl Muro Fadrique

Director

David Notivoli Sánchez

Ponente

José Luis Villarroel Salcedo

Escuela de Ingeniería y Arquitectura

2024

RESUMEN

Este trabajo se ha realizado en el contexto de las prácticas en Kintech Instruments y gestionadas por UNIVERSA, con el objetivo principal de mejorar el proceso de verificación del SmartGate. Este dispositivo es una pasarela industrial creada para conectar sensores digitales con salida RS-485/RS-232 en sistemas de monitoreo usando el protocolo Modbus, en sus variantes RTU y TCP/IP. Su adaptabilidad y confiabilidad lo hacen un componente clave para redes industriales que requieren una comunicación eficiente.

El proyecto, realizado como base del Trabajo de Fin de Grado (TFG), se enfoca en poner en práctica un sistema de control de calidad adaptado a procesos de industrialización, uniendo soluciones de *hardware* y *software* para asegurar el correcto funcionamiento del dispositivo.

En una primera fase, un análisis exhaustivo del producto, abarcando su diseño de *hardware*, los protocolos de comunicación implementados, y la configuración de su PCB. Este estudio establece las bases para identificar los aspectos críticos a evaluar durante el control de calidad.

En la segunda fase, se desarrolla el sistema de control de calidad, compuesto por un *hardware* externo para la verificación automatizada de las pruebas de comunicación en los puertos RS-232, RS-485 y *Ethernet,* así como los *tests* de resistencias *Pull-Up*, *Pull-Down* y de terminación en los puertos RS-485. Estas pruebas se realizan mediante un programa Linux ejecutado en la Raspberry Pi del SmartGate, que automatiza la configuración y validación del dispositivo.

Después de comprobar el correcto funcionamiento del sistema, la tercera fase se concentra en la generación automática de reportes.

Estos incluyen:

1) Un archivo de texto que registra los datos enviados, recibidos y medidos, indicando los valores esperados y obtenidos, junto con sus respectivos estados. Este reporte funciona como registro interno para la empresa.
2) Un documento en formato PDF que, dependiendo de los resultados de las pruebas, se generará un Certificado de Calidad o un reporte detallado de errores encontrados durante la verificación.

Finalmente, se crea un software de escritorio robusto e intuitivo que integra todas las funcionalidades mencionadas. Este software automatiza el proceso de verificación, reduciendo notablemente los errores humanos y asegurando un sistema de trabajo confiable y efectivo.

SUMMARY

This work has been carried out in the context of the internship at Kintech Instruments and managed by UNIVERSA, with the main objective of improving the verification process of the SmartGate. This device is an industrial gateway created to connect digital sensors with RS-485/RS-232 output in monitoring systems using the Modbus protocol, in its RTU and TCP/IP variants. Its adaptability and reliability make it a key component for industrial networks that require efficient communication.

The project, carried out as the basis of the Final Degree Project (TFG), focuses on implementing a quality control system adapted to industrialisation processes, combining hardware and software solutions to ensure the correct functioning of the device.

In a first phase, an exhaustive analysis of the product, covering its hardware design, the communication protocols implemented, and the configuration of its PCB. This study establishes the basis for identifying the critical aspects to be evaluated during quality control.

In the second phase, the quality control system is developed, consisting of external hardware for the automated verification of the communication tests on the RS-232, RS-485 and Ethernet ports, as well as the Pull-Up, Pull-Down and termination tests on the RS-485 ports. These tests are performed by a Linux program running on the SmartGate's Raspberry Pi, which automates the configuration and validation of the device.

After verifying the correct functioning of the system, the third phase concentrates on the automatic generation of reports.

These include:

1) A text file that records the data sent, received and measured, indicating the expected and obtained values, together with their respective statuses. This report serves as an internal record for the company.

2) A PDF document that, depending on the results of the tests, will generate a Quality Certificate or a detailed report of errors found during the verification.

Finally, a robust and intuitive desktop software is created that integrates all the above mentioned functionalities. This software automates the verification process, significantly reducing human errors and ensuring a reliable and effective working system.

Índice

Índice de Figuras

Índice de Tablas

LISTA DE ABRTEVIATURAS

TFG. Trabajo de fin de grado.

SCADA. Supervisory Control and Data Acquisition.

PCB. Printed Circuit Board.

GPIO. General-Purpose Input/Ouput.

SoC. System on Chip.

ADC. Analog-to-Digital Converter.

UART: Universal Asynchronous Receiver-Transmitter.

CRC. Comprobación de Redundancia Cíclica.

LRC. Longitudinal Redundancy Check.

LRC. Comprobación Longitudinal Redundante.

LSB. Least Significant Bit.

ICMP. Internet Control Message Protocol.

IP. Internet Protocol.

MAC. Media Access Control.

1. Introducción

A continuación, se presenta la memoria del Trabajo de Fin de Grado (TFG) titulada "Diseño e implementación de un sistema de control de calidad para el dispositivo SmartGate con generación automatizada de reportes orientados a su industrialización".

Este proyecto tiene como objetivo principal el desarrollo de un sistema de *hardware* y *software* robusto, eficiente e intuitivo, diseñado para llevar a cabo el control de calidad del SmartGate, dispositivo desarrollado por Kintech Instruments.

1.1 SmartGate

El SmartGate es un dispositivo pasarela conversor de datos y medios físicos para canales digitales de comunicación de alta fiabilidad y robustez para su uso en redes de datos industriales, permitiendo integrar una amplia gama de sensores con salida digital, destinados a la medición de parámetros ambientales, facilitando la compatibilidad con sensores que cuentan con salida digital RS-485 a su infraestructura de monitoreo, se ilustra en la figura 1. [1]

Figura 1. Referencia visual del SmartGate

En la Tabla 1 se pueden ver las especificaciones del producto que ofrece la empresa.

SPECIFICATIONS

Operating voltage	6 - 30 VDC (12VDC recommended)
Average current consumption	250 mA (at 12VDC)
Maximum operating temperature	70 ºC
Minimum operating temperature	-30 ºC
Dimensions	108 x 89 x 62 mm
Weight	195 gr
Mounting system	DIN-rail mounting / Panel mounting

Tabla 1. Especificaciones el SmartGate

Características técnicas del SmartGate:

El SmartGate tiene cinco puertos serie que pueden funcionar con protocolos físicos RS-485 (half-duplex) o RS-232 (full-duplex), y también cuenta con dos puertos Ethernet. Es compatible con dispositivos que usen el protocolo industrial Modbus en sus versiones RTU/ASCII y TCP/IP, permitiendo comunicación a través de puertos serie y Ethernet. Este protocolo de maestro-esclavo permite que el SmartGate funcione como interfaz para conectar dispositivos externos, configurándose como maestro o esclavo según se requiera.

El dispositivo también proporciona una configuración flexible, que permite asignar y convertir datos de entrada a salida según los requisitos específicos. En aplicaciones sin conversión de datos, el SmartGate funciona como un simple convertidor de medios físicos, facilitando la comunicación en redes industriales.

Aplicaciones del SmartGate:

- **Integración de sensores:** Permite conectar sensores que funcionan como esclavos Modbus RTU sobre RS-485 o RS-232, al DataLogger* Orbit 360 de la empresa, que actúa como maestro, y al mismo tiempo a un sistema SCADA que opera como maestro Modbus TCP/IP. Así se soluciona el problema de tener dos maestros Modbus conectados a un mismo esclavo que quieren obtener datos en tiempo real.

- **Ampliación de canales:** En caso de que uno de los canales del DataLogger presente fallos, se presenta como solución para evitar el reemplazo del DataLogger, pudiendo ser una reparación costosa, en su lugar se agrega solo el SmartGate que es más asequible.

- **Mapeo y transformación de datos:** Facilita la conversión aplicando en los datos recolectados por ModBus un *Slope** y/o un *Offset**, en caso de ser necesarios. El mapeo permite unir áreas de direcciones Modbus distantes de modo que se pueden leer en una única consulta Modbus, lo que es útil en redes Modbus muy saturadas y con limitaciones de tiempo para obtener datos.

1.2 Objetivos y alcance del proyecto

El principal objetivo de este TFG es el desarrollo de un sistema de control de calidad automático para el dispositivo SmartGate, alineado con los procesos de industrialización requeridos en la fabricación de dispositivos electrónicos. Este sistema busca reducir costes, tiempos y errores humanos en la verificación del producto, implementando pruebas funcionales automatizadas que garanticen la fiabilidad del dispositivo.

Necesidades del proyecto:

1. **Diseño de *hardware* y *software* de verificación:**
   - Diseñar un *hardware* de control de calidad externo que permita la verificación automatizada de:
     - Tres puertos RS-485.
     - Dos puertos RS-485/RS-232 configurables.
     - Dos puertos *Ethernet*, asegurando su correcto funcionamiento en comunicación TCP/IP.
   - Medir y validar las tensiones en los bornes de conexión del dispositivo, incluyendo:
     - Tensión de alimentación principal.
     - Tensión de las *Outputs* (VCCs y GNDs)
     - Tensiones en las líneas A y B de los canales RS-485 para verificar resistencias de *pull-up*, *pull-down* y terminación.

2. Desarrollo de un sistema de generación de reportes

- Crear un sistema que automatice la generación de los siguientes reportes:
  - Reporte técnico en formato .txt: Registro detallado de las pruebas realizadas, los valores esperados y obtenidos, indicando claramente si el dispositivo pasa (OK) o no (Revisar) cada prueba.
  - Certificado de calidad en formato PDF: Documento destinado al cliente final que incluya:
    - Fecha de la prueba.
    - Número de serie del dispositivo.
    - Resultados de las pruebas realizadas en los puertos y tensiones.
    - Sello oficial de la empresa como validación.
  - Relleno Excel de control de *stock* con todos los datos necesarios.

3. Desarrollo de un *software* de escritorio intuitivo y robusto

Diseñar una interfaz gráfica que permita al personal técnico interactuar con el sistema de control de calidad de manera sencilla, incluyendo:

- Indicadores visuales (verde/rojo) para cada prueba, que permitan identificar rápidamente el estado del dispositivo.
- Un panel de configuración donde se puedan ajustar parámetros específicos de las pruebas, las IP y rutas de acceso a generación de reportes y Excel.
- Un terminal de resultados para visualizar en tiempo real el estado de ejecución del testeo.
- Funciones de exportación automática de los reportes generados a una carpeta con el número de serie del producto (PDF, TXT, Excel).

Objetivos que cumplir:

1. Optimización del proceso de fabricación mediante automatización

Implementar un sistema que permita realizar las pruebas de calidad de manera automática, eliminando la necesidad de intervención humana directa durante las pruebas.

Acortar los tiempos de producción.

2. Pruebas de validación y documentación del sistema

Realizar pruebas exhaustivas para validar el funcionamiento correcto del producto en diferentes situaciones operativas.

Registrar todo el proceso, desde el diseño inicial hasta la implementación final, asegurando el seguimiento del proyecto y facilitando futuras mejoras o cambios futuros del sistema.

1.3 Ámbito y motivación

Este documento recoge un proceso de investigación planteado por la empresa Kintech Instruments y delegado en mí, alumno de prácticas y TFG, gestionado a través de UNIVERSA, Servicio de Orientación y Empleo de la Universidad de Zaragoza.

La motivación principal de este proyecto se basa en la oportunidad de trabajar junto a una empresa de referencia en el sector como es Kintech Instruments, fomentada por la continua innovación de sus productos e investigaciones, la calidad de trabajo y vocación de todos sus trabajadores, la perspectiva de la empresa por su crecimiento internacional, así como, por supuesto, es poner en práctica conocimientos adquiridos durante toda la carrera en el grado de Ing. Electrónica y Automática en la Universidad de Zaragoza.

1.4 Cronograma y planificación

El diagrama de Gantt mostrado en la tabla 2 presenta la distribución temporal de las fases por las que ha transitado el proyecto:

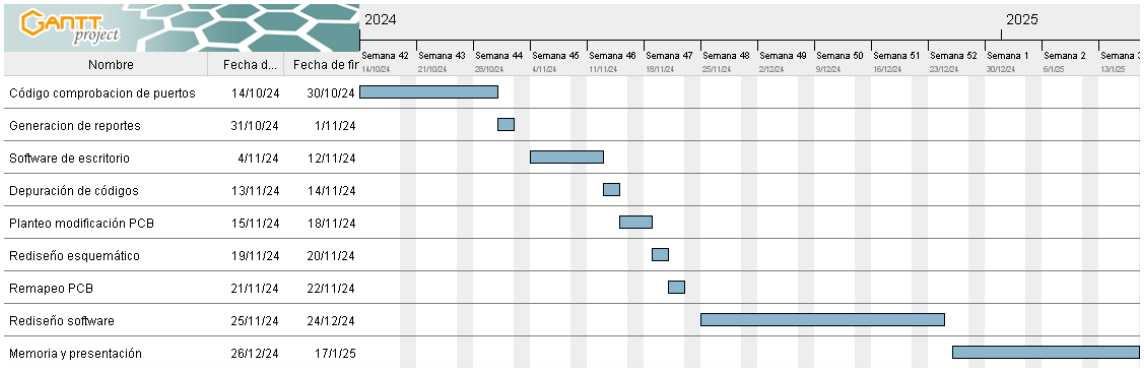


Tabla 2. Cronograma del proyecto

2. Descripción del SmartGate.

2.1 Hardware

Este apartado tiene como objetivo describir en detalle los componentes *hardware* que conforman el SmartGate, ya que estos son los elementos principales para evaluar en el sistema de control de calidad desarrollado en este proyecto.

2.1.1 Núcleo o Core (Raspberry Pi)

La Raspberry Pi es un ordenador en miniatura compacto de placa reducida, ilustrado en la figura 2, que implementa tecnología SoC. Su gran potencia y su tamaño reducido lo hacen muy eficiente para aplicaciones de bajo coste con capacidades de procesamiento superiores a las de los microcontroladores. Este dispositivo funciona con un sistema operativo Linux, lo que le permite ser altamente versátil y personalizable. [2]

Las utilidades que se contemplan con este módulo son: [3]

- **Procesamiento de datos:** actúa como un servidor en miniatura para recopilar, procesar y almacenar datos.
- **Gestión de protocolos:** convierte datos entre diferentes protocolos de comunicación.
- **Conexión remota:** facilita la conexión con sistemas remotos.
- **Supervisión local:** permite que operadores locales accedan a la interfaz del SmartGate para monitorear parámetros.
- **Automatización:** permite ejecutar *scripts* de Python u otras herramientas, que utilizamos para comprobar las funciones del control de calidad.

Las principales razones para usar la Raspberry son:

- Bajo costo.
- Versatilidad.
- Capacidad de procesamiento.

Además, la Raspberry Pi es compatible con una amplia gama de periféricos y dispositivos adicionales, lo que la hace en una opción versátil para múltiples tareas dentro del sistema de control de calidad. Gracias a sus GPIO* y módulos expandidos, se puede personalizar para interactuar con diferentes sistemas y dispositivos según las necesidades específicas del proyecto.

Por último, la capacidad de la Raspberry Pi para trabajar con diferentes protocolos de comunicación, como *Ethernet*, RS-485 y RS-232, la hace una herramienta clave para integrar múltiples dispositivos y consolidar todos los recursos en un sistema único y eficiente.

Figura 2. Referencia visual de la Raspberry Pi utilizada (Compute module 4)

2.1.2 Puertos de comunicación sobre medios físicos

Como se mencionó previamente, el dispositivo SmartGate cuenta con un total de cinco puertos de comunicación serie, de los cuales dos puertos son configurables entre RS-232 y RS-485, mientras que los tres puertos restantes están exclusivamente configurados para RS-485, a demás de los dos puertos *Ethernet*.

1. SELECCIÓN ENTRE RS232 Y RS485

La PCB del SmartGate implementa un conjunto de etapas configurables ilustradas en la figura 3, mediante 4 multiplexores 74LVC1G53DC se configura la comunicación de las señales en los puertos RX1, RX2, TX1, TX2. Esta configuración permite determinar si la señal se transmite a través de RS-232 o RS-485, aplicables únicamente a los puertos 1 y 2.

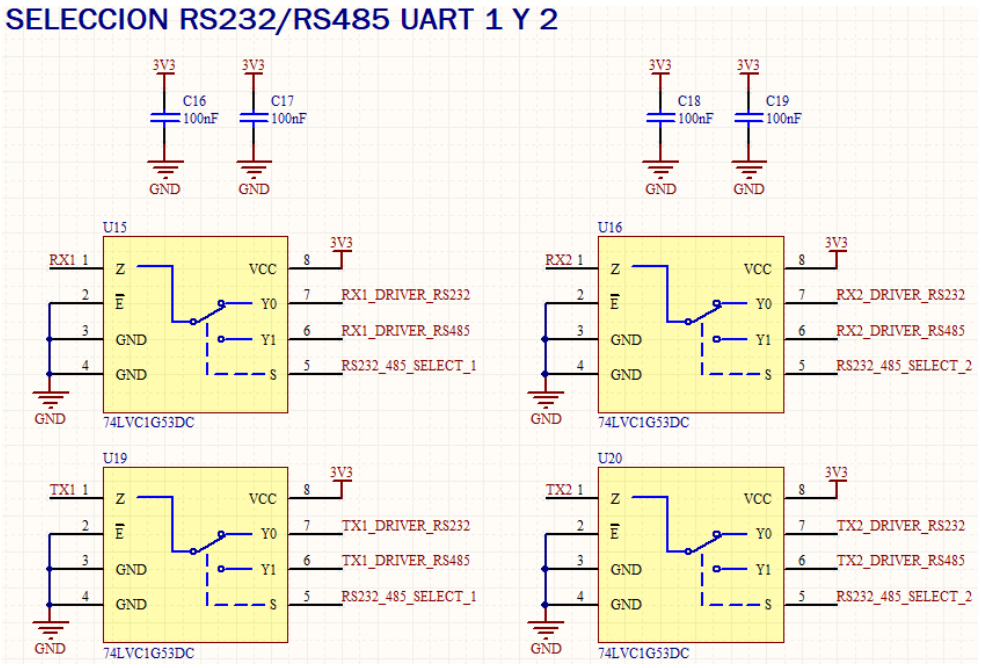


Figura 3. Etapas de selección de RS-232 o RS-385 para puertos 1 y 2

Los multiplexores se configuran de modo que las señales se redirijan de la siguiente manera: RX1 se conecta a RX1_DRIVER_RS232 o RX1_DRIVER_RS485, TX1 a TX1_DRIVER_RS232 o TX1_DRIVER_RS485, RX2 a RX2_DRIVER_RS232 o RX2_DRIVER_RS485 y TX2 a TX2_DRIVER_RS232 o TX2_DRIVER_RS485. De esta forma, la señal se redirigirá al driver* consecuente a través del que se realiza la comunicación. Esta configuración se selecciona mediante la GPIO7 de la Raspberry Pi, la cual se establece en estado 'low' para RS-232 y en "high" para RS-485, tal como se detalla en el Mapa GPIO de la Raspberry que aparece en el **Anexo D**.

2. PUERTOS RS-232

Una vez seleccionada la configuración de comunicación RS-232 en los multiplexores, la señal es gestionada mediante un *driver* MAX3224EEAP, ilustrado en la figura 4.

MAX3224EEAP es un transceptor RS-232 que convierte señales TTL (0V a 3.3V) a los niveles de voltaje estándar RS-232, aproximadamente ±12V. Este dispositivo permite la comunicación bidireccional entre el SmartGate y los dispositivos conectados por RS-232. Se caracteriza por su bajo consumo y está diseñado para soportar sobrecargas y cortocircuitos, lo que lo hace una opción fiable para aplicaciones industriales.

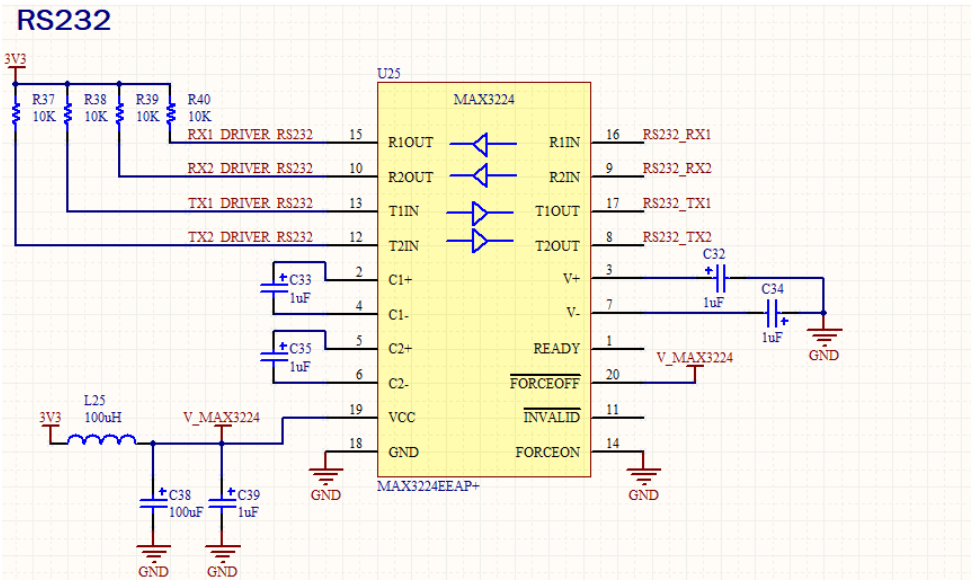


Figura 4. Driver para comunicación RS-232

3. PUERTOS RS-485

En cuanto a los puertos RS-485, se utilizan dos tipos de *drivers* dependiendo del puerto. Para los puertos 1, 2, 3 y 4, se emplean *drivers* MAX3471EUA+ como le ilustra en la figura 5, los cuales están acompañados de una etapa de control de flujo y un multiplexor con las resistencias de *bias* y terminación.

Una vez configurado el sistema de comunicación RS-485, la señal es gestionada por el transceptor MAX3471EUA+, que permite la transmisión de datos a través de distancias largas, garantizando alta inmunidad al ruido. Convierte las señales de bajo voltaje del microcontrolador (3V3 o 5V) a los niveles de voltaje requeridos para RS-485. En este protocolo, la lógica "1" se representa con un voltaje diferencial positivo (línea A mayor que línea B), mientras que la lógica "0" corresponde a un voltaje diferencial negativo (línea B mayor que línea A), garantizando una comunicación robusta y confiable incluso en entornos con interferencias electromagnéticas. Además, se caracteriza por su bajo consumo de energía, protección contra cortocircuitos y conmutación automática de dirección.

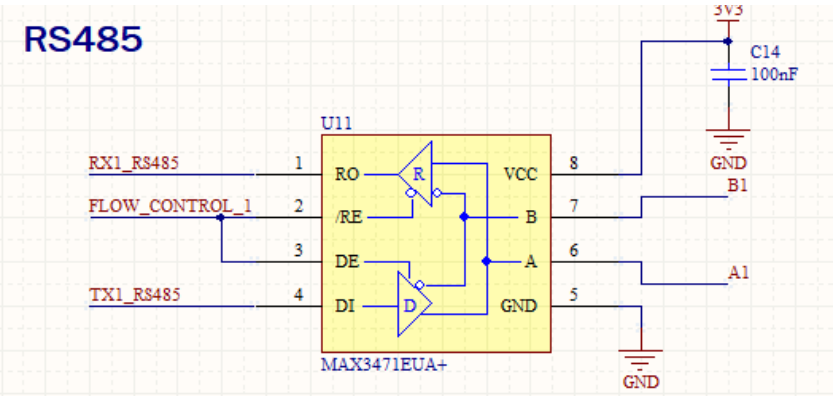


Figura 5. Driver para comunicación RS-485

En la figura 6 se muestra la etapa de control de flujo encargada de garantizar que las señales de transmisión y recepción estén correctamente habilitadas y adaptadas.

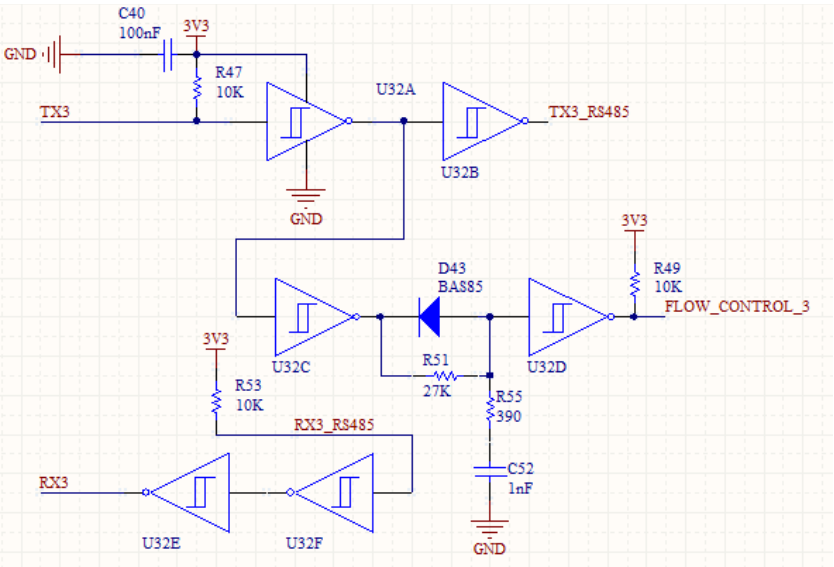


Figura 6. Etapa de control de flujo RS-485

En la Figura 8 se presenta la etapa de conmutación observando diferentes configuraciones para las señales RS-485, en la cual se encuentran las resistencias de *bias* y terminación. Las resistencias de *bias* (*Pull-Up* y *Pull-Down*) son imprescindibles para mantener estabilidad en la transmisión de señales diferenciales en buses RS-485.

En la comunicación RS-485, se transmiten dos señales diferenciales, A y B, y la interpretación de la señal depende de la diferencia de tensión entre ellas:

- Si la diferencia de tensión es mayor a 200mV, se interpreta como un 1 lógico.
- Si la diferencia de tensión es menor a -200mV, se interpreta como un 0 lógico.
- Si la diferencia de tensión se encuentra entre -200mV y 200mV, se considera un estado de incertidumbre.

Los estados de incertidumbre pueden generar fallos en la comunicación y se generan en los siguientes casos:

- **Bus en estado inactivo:** Todos los transceptores están en espera de recibir datos, pero no se genera ninguna señal diferencial, lo que resulta en un voltaje cercano a 0V.
- **Desconexión de un transceptor:** Cuando un transceptor se desconecta, las líneas A y B quedan flotantes sin una diferencia de voltaje suficiente.
- **Incapacidad del controlador:** El controlador RS-485 no puede generar la diferencia de voltaje necesaria para representar un 0 o 1 lógico.

Para evitar estos estados de incertidumbre, se colocan las resistencias de *Pull-Up* y *Pull-Down*, en la figura 7 se muestra su disposición, las cuales aseguran que, incluso en un estado inactivo o de desconexión, se mantenga una diferencia de potencial adecuada, eliminando estos posibles estados de incertidumbre. Estas resistencias deben garantizar que la diferencia de voltaje sea mayor a 200mV o menor a -200mV, sin ser demasiado pequeñas para evitar un consumo excesivo de energía, y deben ajustarse a las características del circuito.

En redes RS-485 de larga distancia y a altas velocidades, es recomendable utilizar resistencias de terminación de 120Ω, lo que minimiza las reflexiones de línea. Aunque la terminación aumenta el consumo de corriente, mejora la integridad de la señal. En aplicaciones con distancias de cable más cortas, el uso de estas resistencias es menos importante.

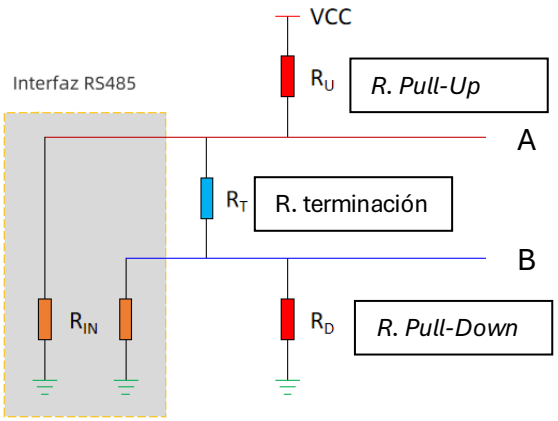


Figura 7. Referencia disposición de resistencias de bias y terminación

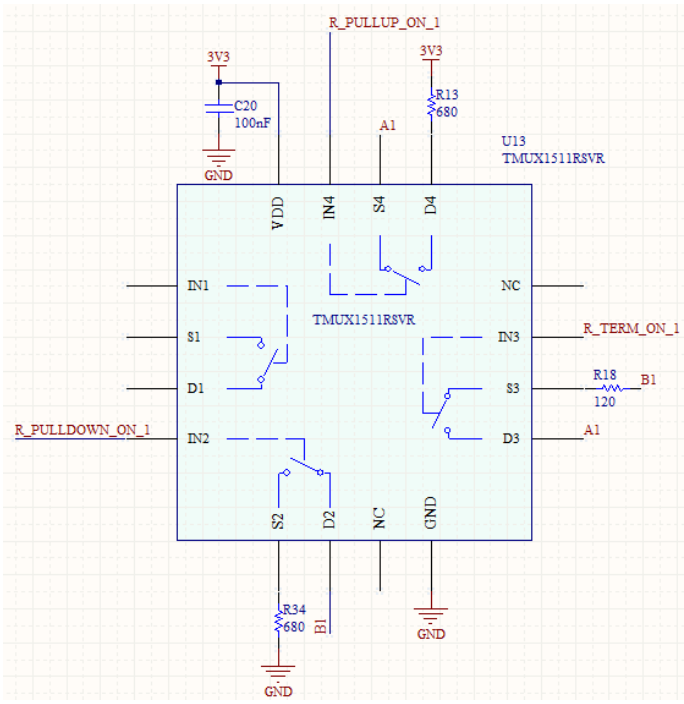


Figura 8. Etapa de configuración de línea RS-485

En el caso del puerto 5, se utiliza un *driver* RS-485 diferente, que incluye aislamiento en la alimentación, acompañado de una etapa de control de flujo igual a la de los demás puertos RS-485, tal y como se muestra en la figura 6, así como una etapa de aislamiento para las señales de control del bus RS-485.

El dispositivo que se ilustra en la figura 9 convierte las señales de bajo voltaje del microcontrolador a los niveles de voltaje requeridos para la comunicación en bus, con una diferencia de voltaje entre las líneas A y B de +1.5V a +5V para lógica "1" y -1.5V a -5V para lógica "0". Esto garantiza una transmisión fiable y robusta de datos en entornos industriales.

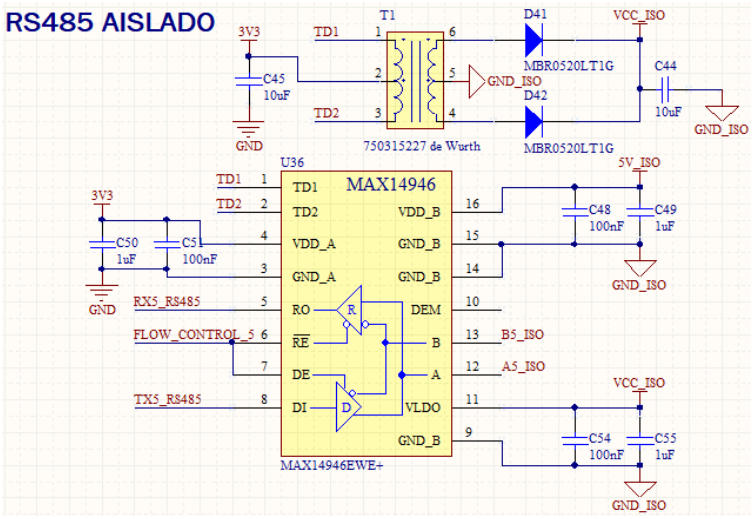


Figura 9. Driver para comunicación RS-485

Al igual que con la alimentación se aíslan también las señales de control provenientes de las resistencias de *bias* y terminación para el bus RS-485, protegiendo contra interferencias o diferencias de potencial, tal y como se indica en la figura 10.

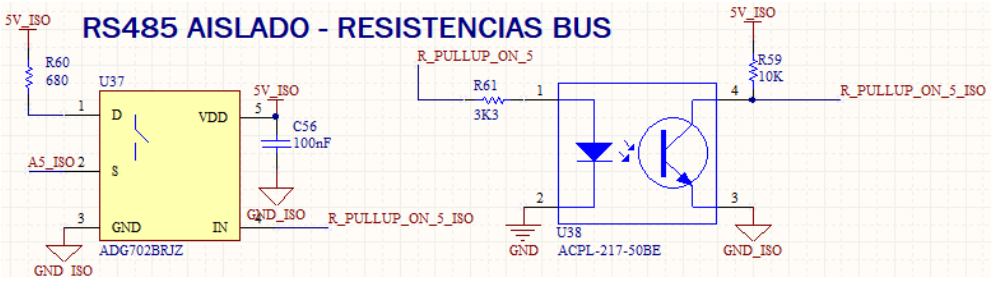


Figura 10. Etapa de aislamiento en señales RS-485

2.1.3 Ethernet

En el SmartGate se implementa una etapa con un circuito integrado diseñado a para la protección de líneas de datos de alta velocidad contra descargas electrostáticas. En conjunto también se diseña una etapa con un conector modular RJ45 con transformadores integrados diseñados específicamente para redes *Ethernet*.

Para el puerto *Ethernet*2 se implementa también un controlador de interfaz de red USB 2.0 a *Ethernet* diseñado para proporcionar conectividad de red rápida y fiable en dispositivos que no cuentan con un puerto *Ethernet* integrado, en conjunto con un con un circuito integrado de protección contra descargas electrostáticas y un conector modular RJ45 al igual que el puerto *Ethernet*1, tal y como se muestra en la figura 11.

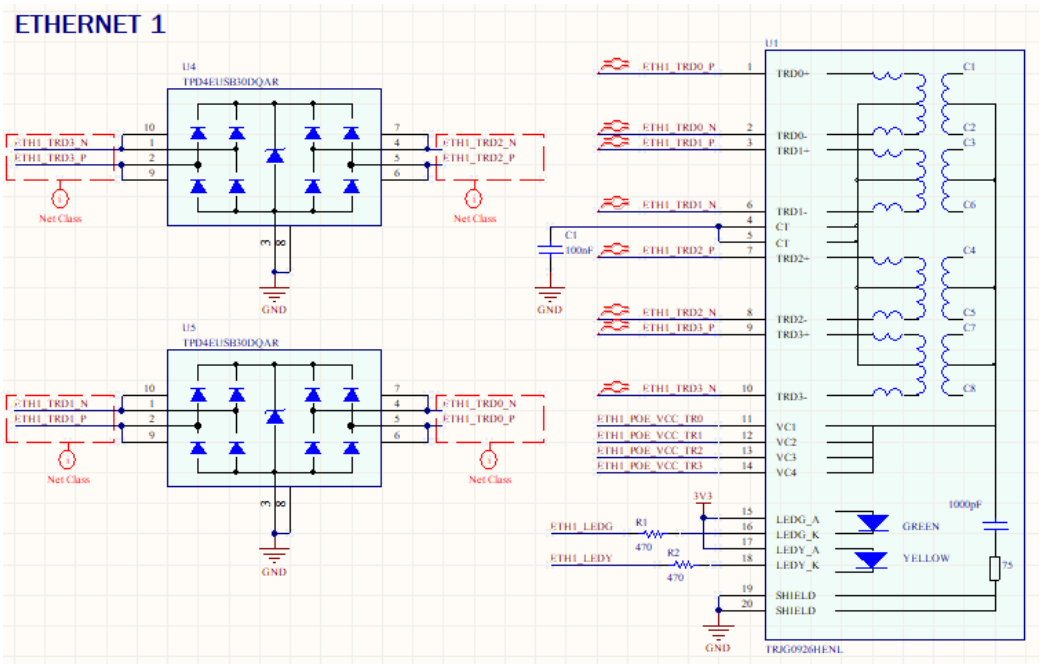


Figura 11. Etapa de protección y conexión para puertos Ethernet

2.1.4 Bornas de alimentación.

En los laterales de la PCB del SmartGate se encuentran ubicados los borneros, tal y como se muestra en la figura 12, los cuales están diseñados para albergar las siguientes conexiones:

- Alimentación del SmartGate: Incluye la alimentación Vin y su correspondiente GND.
- Cuatro salidas de alimentación para sensores (VCC): Estas están unidas directamente a Vin a través de descargadores de gas para protección contra rayos. Cada una cuenta con su respectivo terminal GND, conectado directamente a masa.
- Dos alimentaciones a 5V: Estas provienen del regulador interno del SmartGate.
- Señales RX y TX para RS-232: Incluyen RX1, TX1, RX2 y TX2, que vienen del *driver* RS-232 ilustrado en la Figura 4. Estas señales son adaptadas mediante fusibles, diodos de protección y descargadores de gas para garantizar la seguridad del sistema frente a sobrecargas y descargas de rayos.
- Señales A y B de los puertos RS-485: Estas corresponden a los cinco puertos del SmartGate.
    - Para los puertos 1-4, las señales A y B provienen de los *drivers* RS-485 estándar, que se ilustra en la figura 5.
    - Para el puerto 5, las señales A y B provienen del *driver* RS-485 aislado, que se ilustra en la figura 9.

Al igual que las señales de RS-232, estas señales se adaptan mediante fusibles, diodos de protección y descargadores de gas para asegurarlo y protegerlo de posibles daños eléctricos.

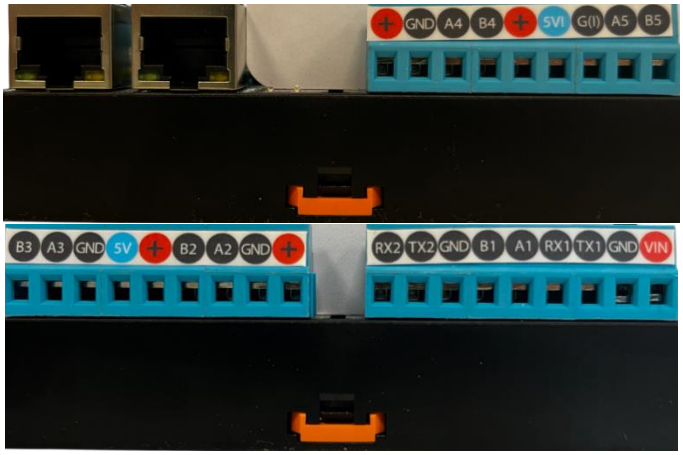


Figura 12. Conexionado borneros

2.2 Software

Este apartado tiene como objetivo describir en detalle los componentes *software* que forman parte del SmartGate, ya que estos constituyen el núcleo funcional del sistema de control de calidad desarrollado en este proyecto.

2.2.1 Protocolos de comunicación

Protocolo MODBUS

Modbus es un protocolo de comunicación muy utilizado en entornos industriales, basado en un modelo de comunicación maestro-esclavo. En esta disposición, el maestro controla la comunicación, solicitando datos a otros dispositivos de la red, mientras que el esclavo responde solo a las solicitudes del maestro. Un esclavo no puede iniciar una comunicación por sí mismo, sino que espera a ser solicitado. En una red Modbus, siempre hay un único maestro y múltiples esclavos. [4]

1. **MODBUS TCP/IP (PC a SmartGate)**

Modbus TCP es una versión del protocolo Modbus diseñada para funcionar sobre redes *Ethernet*. En este caso, se utiliza para establecer la comunicación entre la Raspberry Pi y el PC. Al inicializar cada dispositivo SmartGate, se instala una imagen de sistema que se encarga de la configuración. Las direcciones IP son fijas y permiten a la empresa acceder fácilmente a los SmartGate para realizar las comprobaciones de la comunicación y llevar a cabo las pruebas necesarias en el sistema de control de calidad. Esta configuración optimiza el proceso de pruebas al asegurar que la conexión y la comunicación con cada dispositivo sean accesibles.

2. MODBUS RTU (SmartGate a PCB o sensores a SmartGate)

Modbus RTU es una de las versiones más utilizadas del protocolo Modbus en entornos industriales debido a su alto rendimiento y capacidad de transmitir datos de manera compacta. La transmisión de datos en Modbus RTU se realiza utilizando un formato binario, lo que permite una mayor densidad de información, optimizando el uso del espacio en la transmisión. Esto se logra mediante la codificación de cada byte de 8 bits, que contiene dos dígitos hexadecimales de 4 bits, proporcionando mayor eficiencia que el formato ASCII, especialmente a velocidades altas de comunicación.

Cada mensaje Modbus RTU está compuesto por un encabezado que identifica al dispositivo destinatario, un cuerpo de datos que contiene la información a transmitir y un código de verificación, que generalmente se implementa mediante el uso del CRC. El CRC, que es una suma de control de redundancia cíclica, tiene como función asegurar la integridad de los datos transmitidos. Este valor se calcula por el dispositivo transmisor y se verifica por el receptor; si los valores no coinciden, se detecta un error en la transmisión.

El principal beneficio de utilizar Modbus RTU es su eficiencia en la transmisión de datos, ya que permite el envío de mayor cantidad de información en menor tiempo y con menor consumo de ancho de banda. Sin embargo, esta mayor eficiencia viene con el costo de una mayor complejidad en la lectura y depuración de los mensajes.

3. MODBUS ASCII (SmartGate a PCB o sensores a SmartGate)

El protocolo Modbus ASCII es una variante que utiliza texto para codificar la información transmitida. Aunque su formato es más comprensible para los usuarios, es menos eficiente que Modbus RTU debido a que los mensajes ocupan más espacio y requieren un mayor ancho de banda para la misma cantidad de información.

Una de las principales ventajas de Modbus ASCII es la facilidad para leer y depurar los mensajes. Esto lo hace adecuado para aplicaciones en las que la simplicidad y la interpretación de los datos son más importantes que la eficiencia en la transmisión.

Los mensajes en Modbus ASCII comienzan con el carácter ":" y finalizan con los caracteres de retorno de carro (CR) y salto de línea (LF), los cuales marcan el fin del mensaje. Si el tiempo entre caracteres excede un segundo, se interpreta como un error en la transmisión.

Al igual que Modbus RTU, Modbus ASCII incluye una verificación de errores mediante un método de comprobación llamado LRC. El LRC es un byte que se calcula sumando los valores de los bytes del mensaje, excluyendo los caracteres ":" y CRLF, descartando los acarreos y realizando el complemento a dos del resultado. Este valor se agrega al mensaje por el emisor y se verifica por el receptor durante la transmisión. Si los valores calculados no coinciden, se detecta un error en la comunicación.

Comunicación física [5]

1. Comunicación RS-232

La interfaz RS-232 es un protocolo de comunicación fiable utilizado para la transferencia de datos en configuración punto a punto, diseñado para la comunicación serie entre dispositivos en distancias cortas. En este tipo de comunicación, un único transmisor se comunica con un único receptor, lo que lo hace adecuado para aplicaciones en las que se requiere enviar información desde sensores o terminales transmisores a un receptor o equipo de comunicación. En el contexto del SmartGate, esta interfaz se emplea para la comunicación de datos entre el dispositivo de control y los sistemas de medición de los puertos RS-232, permitiendo la transmisión de datos esenciales para la evaluación del rendimiento del dispositivo.

Este protocolo emplea líneas no balanceadas, donde cada señal está referenciada a masa, utilizando voltajes positivos y negativos para definir los niveles lógicos. En este caso, un nivel lógico '0' se representa con una tensión igual o mayor a 5 V, y un nivel lógico '1' con una tensión igual o inferior a -1 V. Los receptores son capaces de manejar señales con ruido, aceptando niveles en el rango de 3V y -3V. Aunque RS-232 es resistente al ruido, se implementa aislamiento galvánico mediante acoplamiento óptico o magnético para proteger los circuitos de interferencias o sobretensiones, minimizando el riesgo de perturbaciones en cables largos.

El aislamiento en este tipo de comunicación se logra mediante la división del circuito en secciones sin conexión galvánica, donde el acoplamiento óptico o magnético permite la transferencia de datos y energía entre las secciones, filtrando el ruido. También es posible aislar las conexiones a tierra, lo que aumenta la inmunidad del circuito ante sobretensiones y ruido provenientes de conexiones a tierra compartidas con otros circuitos cercanos.

Este aislamiento es crucial en entornos industriales donde las interferencias y las sobretensiones pueden afectar el rendimiento y la fiabilidad de los dispositivos.

En la comunicación serie RS-232 se distinguen dos modos: *half-duplex y full-duplex*. En el modo *half-duplex*, se utiliza una sola línea para la transmisión y recepción de datos, pero no puede realizar ambos procesos simultáneamente. Al contrario, en el modo *full-duplex* se emplean dos líneas, permitiendo la transmisión y recepción simultánea. En este proyecto, se utiliza el modo *half-duplex*, ya que, al actuar como maestro en la comunicación, el dispositivo regula el proceso solicitando información y luego recibiendo datos de los sensores, sin necesidad de una comunicación simultánea. Esta configuración es ideal para el sistema de control de calidad, ya que optimiza los tiempos de comunicación y reduce la complejidad del proceso de transmisión de datos entre los diferentes dispositivos involucrados.

Además, el uso de RS-232 para la conexión con dispositivos de medición permite una implementación sencilla y fiable en distancias cortas, lo que resulta adecuado para el control de calidad en entornos industriales, donde la rapidez y precisión en la transmisión de información son esenciales para el diagnóstico y monitoreo de las condiciones del sistema.

2. Comunicación RS-485

La interfaz RS-485 es la más utilizada en el ámbito industrial. Esta utiliza la conocida tipología multipunto, la cual puede conectar múltiples receptores y transmisores como se ve en la figura 13, con una transmisión de datos mediante señales diferenciales. Es un estándar de comunicaciones de transmisión serial y asíncrona, es decir, los bits se transmiten de uno en uno sin una señal que sincronice transmisor y receptor. [6] [7] [8]

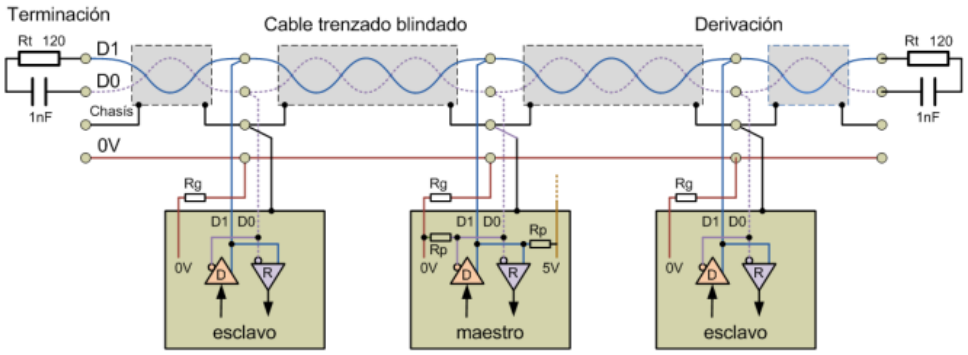


Figura 13. Diagrama red RS485 con múltiples transceptores

A diferencia del RS-232 es menos costosa ya que únicamente necesita una fuente de 5V o menos para generar la diferencia mínima de 1.5V, permite un conexionado mayor de dispositivos, tiene mayor inmunidad al ruido debido a que se cancela igual en ambas líneas y son más tolerables a diferencias de potencial.

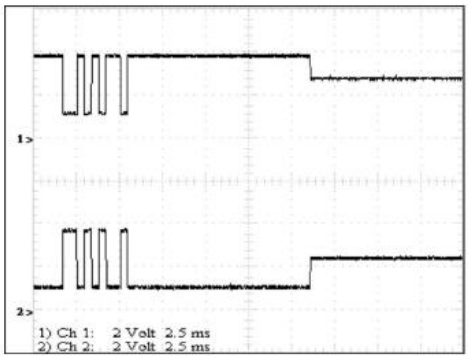


Figura 14. Referencia visual de las salidas de un controlador RS-485

El principio de funcionamiento del RS-485 es mediante diferencia de voltajes entre las líneas A y B, es decir, cuando VA sea mayor que VB se encuentra en nivel lógico alto (1) y cuando al contrario VB es mayor que VA se encuentra en nivel lógico bajo (0). En la figura 14 se muestra un ejemplo de la comunicación vista con un osciloscopio.

- **Requisitos de la señal:** Para el controlador, la diferencia de voltaje entre A y B debe ser de al menos 1,5 V. En el receptor, basta con una diferencia mínima de 0,2 V para determinar el nivel lógico. Este margen de ruido de 1,3 V proporciona robustez adicional frente a interferencias.

- **Velocidad vs distancia:** Una linea RS-485 puede tener una velocidad de bits de hasta 10Mbps o una longitud de hasta 1200 m, pero ambas al mismo tiempo serian inviable. Los cabes más largos requieren velocidades más lentas. Por ejemplo, a 10 Mbps, la longitud máxima es de aproximadamente 15 metros, mientras que a 90 kbps puede alcanzar hasta 1200 metros.

  A continuación, en la figura 15 se muestra la relacion de ambas variables.

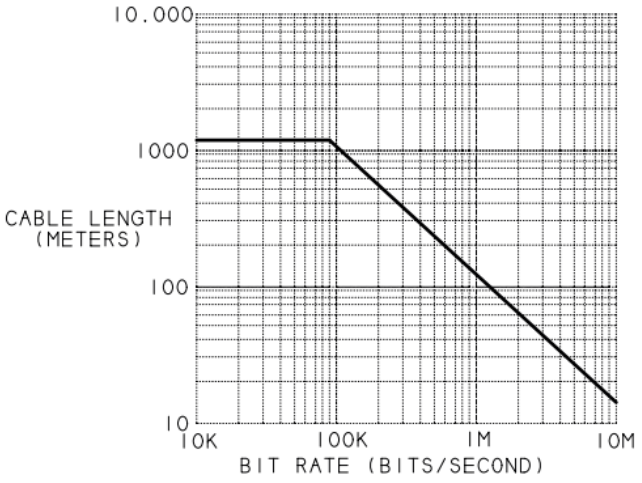


Figura 15. Gráfica que relaciona la distancia con la velocidad en una linea RS-485

Modos de operación y diseño de redes:

Al igual que para la interfaz RS-232, se diferencian los dos modos *half-duplex* y *full-duplex*.

- *Half-Duplex:* Es el modo más común en redes RS-485. En la implementación half-duplex se utiliza una sola línea para la comunicación la cual sirve para transmitir o recibir datos, no será capaz de realizar ambos procesos simultáneamente. Permite que un único nodo transmita datos en un momento dado, mientras que los demás nodos están recibiendo, se reduce el número de cables y se implementa configurando pines del microcontrolador como entradas o salidas, se muestra una representación de la disposición de elementos en la figura 16.

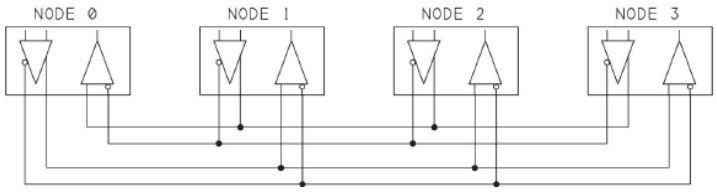


Figura 16. Representación de red half-duplex

- *Full-Duplex:* A diferencia del *full-duplex* este utiliza dos pares de líneas diferenciales y por tanto si será capaz de realizar transmisiones simultaneas. En el caso de este proyecto se trabaja en modo *half-duplex* también.

Diseño y consideraciones de red:

En una red RS-485 solo un controlador debe de estar habilitado a la vez. Si varios controladores intentan transmitir simultáneamente se pueden generar conflictos de voltajes e incluso fallos en la comunicación.

- **Polarización:** Se utilizan resistencias de *Pull-Up* y *Pull-Down* con el fin de garantizar niveles lógicos definidos cuando no hay transmisión activa en la línea.

- **Terminación:** En redes largas y a altas velocidades es adecuado usar resistencias de terminacion de 120 Ω minimizando asi reflexiones de linea. La terminacion aumenta el consumo de corriente, pero mejora la integridad de la señal. En aplicaciones con distancias de cable más cortas es menos importante, como se puede apreciar en este proyecto seguira transmitiendo informacion cuando se comprueba con esa combinacion de resistencias, y así se reduce el consumo de energia.

- **Líneas largas:** Un cable RS-485 puede comportarse como una línea eléctricamente larga o corta. Para determinar estos dos tipos no se hace referencia a la longitud física del cable, sino al tiempo que requiere una señal para propagarse hasta el receptor, la cantidad de tiempo varia con la longitud, la frecuencia de la señal y la velocidad a la que viaja la señal en el cable.

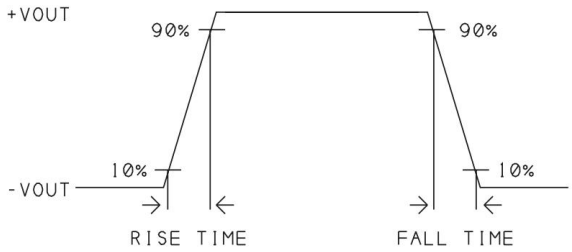


Figura 17. Tiempo de subida de una señal

Como se puede apreciar en la figura 17 el tiempo de subida es el que tarda la salida en cambiar del 10 al 90% del rango completo. Tanto el de subida como el de bajada suelen ser prácticamente idénticos. Un tiempo de subida más corto indica una velocidad de respuesta más rápida y a velocidades de bits posibles más rápidas.

- **Captación de ruido en un cable:** En primer lugar, hay que señalar que el ruido es cualquier señal que no se desea en un circuito. El ruido puede interferir en un cable de formas distintas como acoplamiento inductivo, impedancia común, magnético, capacitivo o electromagnético.
  El acoplamiento conductivo requiere contacto directo entre cables de señal y el que transporta el ruido.
  El acoplamiento de impedancia común ocurre cuando dos circuitos comparten un cable con retorno a tierra común.
  Los otros tipos surgen de interacciones entre campos eléctricos y magnéticos que surgen de los propios cables y componentes.

- **Aislamiento de señales: L**a interfaz de cada nodo tiene una fuente de alimentación aislada y una línea de datos opto aislada. El cable de tierra de las líneas de datos no tiene conexión a la tierra física de ningún nodo, de forma que con esta disposición se puede proteger de señales de tomas de tierra ruidosas y variaciones de voltaje en tierra de alguno de los nodos.
  Cabe destacar que también es posible el aislamiento parcial pudiendo ser más barato o conveniente que el aislamiento total siendo en algunos casos suficiente.

En la tabla 3 se muestra un breve resumen de características, mostrando las principales diferencias entre RS-232 y RS-485. [9]

PARÁMETROS	**RS-232**	**RS-485**
Configuración de línea	*Single ended*	*Differential*
Longitud máxima de cable	15 m	1200 m
Veloc. de transmisión máxima	20 kbits/s	10 Mbits/s
Niveles lógicos	±5 V - ±15 V	±1.5 V - ±6 V
Impedancia mínima de entrada	3Ω - 7Ω	12 Ω
Sensibilidad del receptor	±3V	±200mV

Tabla 3: Diferencias entre comunicación serial RS-232 y RS-485

1.Configuracion en línea:

RS-232: *single-ended* quiere decir que transmite datos con referencia a un único punto de masa. Por ello será más susceptible a que haya ruidos e interferencias.
RS-485: diferencial quiere decir que utiliza dos líneas diferenciales A y B para transmitir datos en diferencias de voltaje.

2.Longitud máxima de cable:

RS-485 ofrece más distancia de cable debido a que trabaja con niveles de tensión más estables debido a su configuración diferencial, a diferencia de RS-232 que es más sensible a la distancia con la aparición de ruidos e interferencias a distancias largas.

3.Veloc. de transmisión:

Estos datos son referentes a longitudes máximas de cable, por tanto es entendible que para RS-232 a máxima longitud tenga velocidades de transmisión mayores, en cambio RS-485 para distancias menores ofrece velocidades más altas.

4.Niveles lógicos:

RS-232 trabaja con niveles lógicos altos lo que indica que habrá un mayor consumo de energía, a diferencia de RS-485 que debido a su tensión diferencial trabaja con niveles menores, lo que lo hará más eficiente energéticamente.

5.Impedancia mínima de entrada al receptor:

RS-232 se limita bastante el número de dispositivos conectados en red, a diferencia de RS-485 permite una conexión de hasta 32 dispositivos en el mismo bus (originalmente), aunque en implementaciones modernas pueden conectarse hasta 256, como es el caso del transceptor utilizado MAX3471EUA.

$$Nº\ max.de\ dispositivos = \frac{32}{Carga\ unitaria\ del\ receptor} = \frac{32}{1/8} = 256$$

6.Sensibilidades del receptor:

RS-232 necesita una diferencia de tensión notable para interpretar señales por lo que no es muy buena elección para aplicación en las que puede aparecer ruido notable, a diferencia de RS-485 que al tener una diferencia de tensión tan baja lo hará más robusto en entornos ruidosos.

3. Antecedente del control de calidad.

Debido a que la producción del SmartGate es un proceso de fabricación que incluye varias fases, es muy importante establecer un sistema de control de calidad que asegure su buen funcionamiento y el cumplimiento de las tareas establecidas.

Con el crecimiento de la empresa y el aumento de las ventas, se ha vuelto necesario avanzar en la automatización de los procesos industriales, especialmente en el control de calidad de los productos. Aunque el control de calidad manual es posible, este enfoque tiene varias desventajas, como el tiempo implementado, y la intervención humana posibilitando la aparición de errores. Además, actividades como el control de *stock*, la elaboración de informes de ventas o la inspección de puertos en caso de fallos requieren un esfuerzo extra, lo que aumenta el tiempo necesario para cada revisión.

El control manual también presenta un mayor riesgo de errores, como olvidar realizar pruebas en algunos puertos o hacer mediciones incorrectas. Por esta razón, la introducción de un sistema automatizado de control de calidad no solo disminuye significativamente la posibilidad de errores, sino que también mejora el tiempo, aumenta la calidad del proceso y asegura resultados más confiables, ayudando a la eficacia del sistema de producción.

En el sistema anterior, el control se realizaba a través de los siguientes pasos:

El SmartGate se configuraba de manera predeterminada con ciertas direcciones IP. Si era necesario, estas IP se podían modificar en **TCP Configuration**, como se ilustra en la Figura 18. Para hacer la verificación, el ordenador se configuraba con direcciones de subred que eran compatibles con las IP asignadas y se conectaba a los puertos *Ethernet* del SmartGate. A través de estas conexiones, se comprobaba que los puertos *Ethernet* funcionaran correctamente, accediendo a la interfaz web de configuración del dispositivo o ejecutando un comando ping a las direcciones IP configuradas.

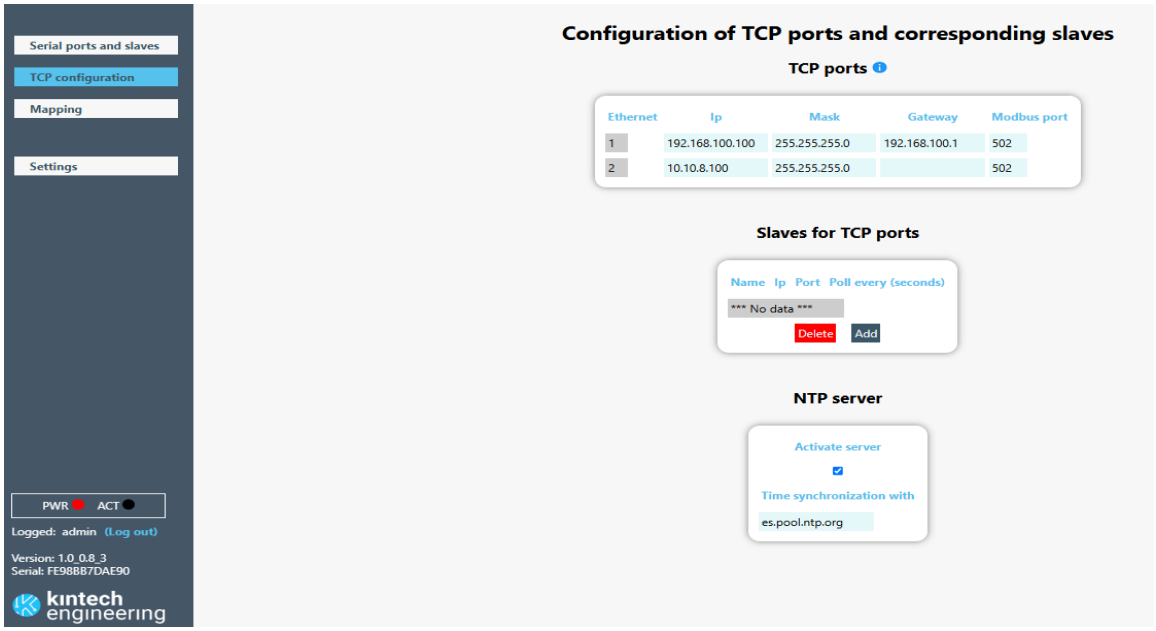


Figura 18. Referencia visual TCP Configuration

Una vez comprobados los puertos *Ethernet* se configuraba el Mapa Modbus accediendo a **ModBus Port Mapping** como se indica en la figura 19. En la parte izquierda se configura la información que se va a comunicar, generalmente se comprobaba con minuto y segundo, y en la parte derecha se muestra el mapeo, es decir, donde se encuentra la información que se va a pedir para comprobar la comunicación y el formato en el que se va a representar.

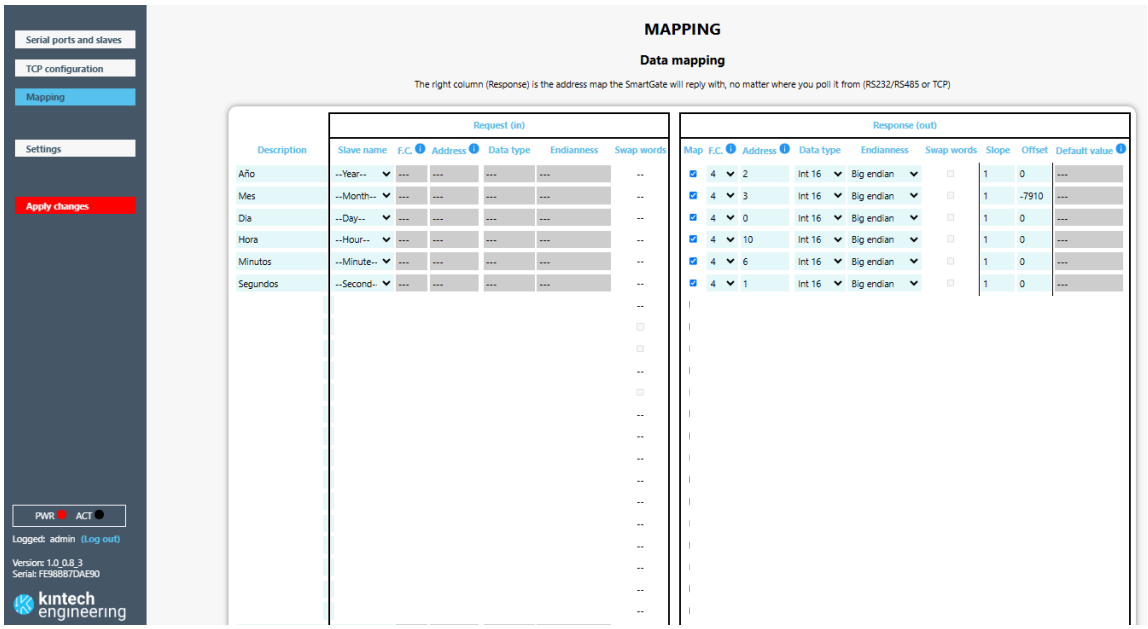


Figura 19. Referencia visual Mapa Modbus

Seguidamente, en la sección **Serial Ports and Slaves** (figura 20), se configuraba el tipo de puerto serial que se utilizará para la comunicación (RS-485 o RS-232), así como ajustar diversos parámetros imprescindibles en la comunicación, como la velocidad de comunicación, paridad y bits de parada, así como activar o desactivar la actuación de las resistencias de polarización (*bias*) y terminación.

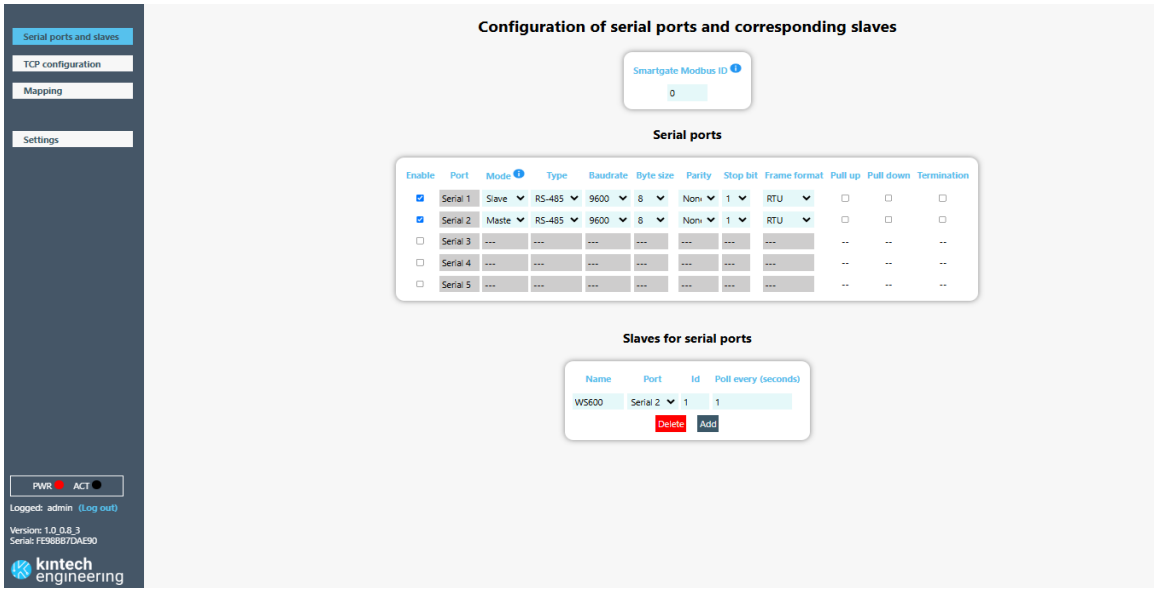


Figura 20. Referencia visual Serial ports and slaves

Finalmente, Conexionando el PC al SmartGate con un adaptador y mediante un programa se procedía a leer continuamente los minutos y segundos, comprobando así, como se actualizaban los datos asegurando el correcto funcionamiento de la comunicación RS-232 o RS-485 y del reloj interno del SmartGate. Este proceso es idéntico a ambas comunicaciones, a diferencia de la configuración que se indicaba en la capa física escogida (RS-232 o RS-485).

Para la comprobación de los valores de los Outputs era necesario el uso de instrumentos adicionales como un *tester* calibrado, lo cual no era lo más eficiente, captando medida a medida de cada bornero y comprobando que tuviera la tensión correcta.

Además, el estado de las resistencias de *bias* y terminación no se comprobaba.

4. Descripción del control de calidad

El sistema de control de calidad creado consta de tres partes principales:

- **PCB externo:** responsable de gestionar las conexiones entre los puertos físicos y medir las señales en los borneros, asegurando la correcta captura de datos.
- **Programa Linux:** Ejecutado en la Raspberry Pi, este programa se encarga de la configuración necesaria para verificar los puertos de comunicación y generar los reportes.
- **Software de escritorio:** Es encargado de conexionar todas las partes del sistema, lo que facilita su uso y garantiza que todo funcione de manera integrada.

4.1 Planteamiento general del control de calidad.

La idea principal de control de calidad es diseñar un sistema capaz de realizar la verificación del SmartGate, como se muestra en la figura 21. El sistema debe:

- Comprobar que las resistencias de *bias* y terminación de los 5 puertos del SmartGate están físicamente implantadas en el diseño y bien soldadas para su correcto funcionamiento.
- Configurar los puertos 1 y 2 del SmartGate para comunicar por RS-232 y comprobar que la comunicación es correcta.
- Configurar los puertos 1 y 2 del SmartGate para comunicar por RS-485 y comprobar que la comunicación es correcta.
- Comprobar que la comunicación es correcta en los tres puertos RS-485 restantes.
- Comprobar que se reciben en los borneros las tensiones correctas (*Outputs*).
- Comprobar el correcto funcionamiento de los dos puertos *Ethernet*.
- Generar reportes para su verificación por el técnico de montaje.

La PCB externa se diseña del mismo tamaño que la del SmartGate, con la misma disposición de los borneros para un mejor manejo.

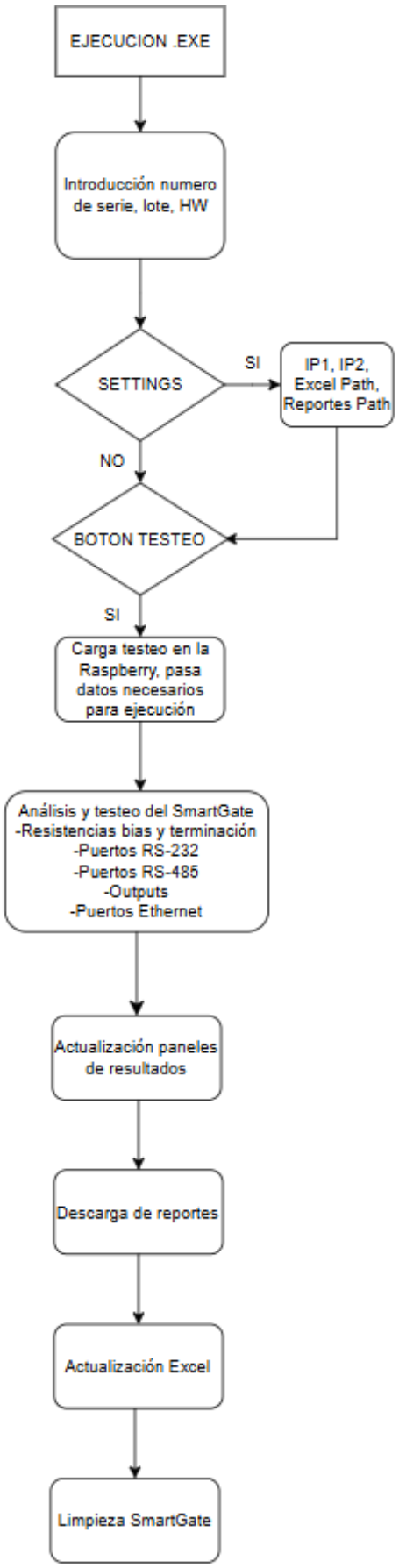


Figura 21. Diagrama de flujo proceso completo

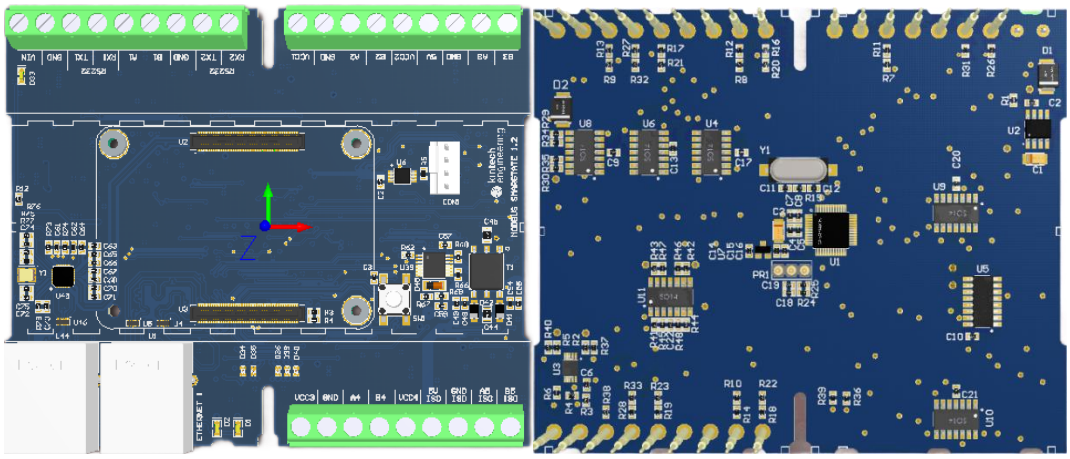


Figura 22. Referencia visual PCB SmartGate y PCB externa respectivamente.
(Cara Top del SmartGate y cara Bottom de la PCB externa)

Figura 23. Referencia visual Conexionado PCB SmartGate y PCB externa

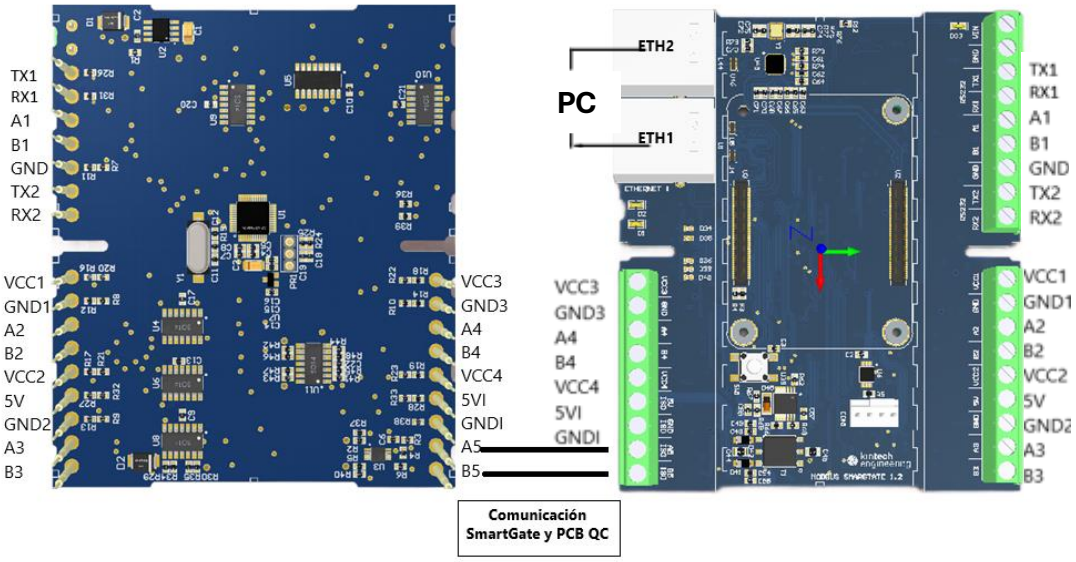


Figura 24. Referencia visual conexionado SmartGate y PCB externa para comunicación

Como se puede apreciar en las figuras 22 y 23, la PCB externa irá situada sobre la PCB del SmartGate, con las sondas en contacto con la parte superior de los borneros. El contacto eléctrico se asegura por presión una vez se coloca la PCB externa de control de calidad sobre el SmartGate a testear. En la figura 24 se muestra un esquema general de comunicación entre PC y SmartGate y SmartGate y la PCB externa.

El programa Linux que se ejecuta en la Raspberry Pi automatiza el proceso de control de calidad. Este *software* incluye rutinas específicas para verificar los puertos *Ethernet*, RS-485 y RS-232, comprobar el estado de las resistencias *pull-up*, *pull-down* y de terminación, validar las conexiones eléctricas, y asegurar la correcta transmisión de datos por el dispositivo.

Se diseña un software de escritorio sencillo pero completo, con indicaciones visuales y botones para interactuar con el SmartGate. Una vez introducidos los datos necesarios para el análisis, como el número de serie, IP del SmartGate, versión de *hardware* y lote, se pulsa un botón para iniciar el análisis del producto.

El *software* de escritorio se comunica con la Raspberry Pi, cargando en ella el programa Linux necesario y dando inicio al proceso de control de calidad. Los detalles de las pruebas realizadas durante este procedimiento se explican en los capítulos siguientes.

4.1.1 Comprobación resistencias *Pull-Up*, *Pull-Down* y Terminación

Para comprobar la existencia y correcto funcionamiento de estas resistencias se procede a analizar las medidas que se obtienen en los borneros referentes a las señales A y B de los 5 puertos RS-485, dependiendo del valor de estas tensiones, es capaz de determinar su estado. En la tabla 4 se representan los registros referentes a las tensiones de los canales A y B de los puertos RS-485.

Registro	Campo	Lectura/Escritura	Formato de datos
12	A1_uC	RO	uint16
13	B1_uC	RO	uint16
14	A2_uC	RO	uint16
15	B2_uC	RO	uint16
16	A3_uC	RO	uint16
17	B3_uC	RO	uint16
18	A4_uC	RO	uint16
19	B4_uC	RO	uint16
20	A5_uC	RO	uint16
21	B5_uC	RO	uint16

Tabla 4. Tabla de registros del cliente Modbus canales A B

4.1.2 Puertos RS-232

Este *test* debe realizarse en modo pasarela, es decir, los *bytes* que se envíen desde un puerto deben llegar al otro. El control de calidad conecta los puertos 1 y 2 RS-232 (RX del puerto 1 con TX del puerto 2 y viceversa). La secuencia de la prueba de comunicación es la siguiente:

- Configurar el puerto 1 y 2 en modo RS-232.
- Enviar *bytes* por el puerto 1, deben llegar al puerto 2.
- Enviar *bytes* por el puerto 2, deben llegar al puerto 1.

4.1.3 Puertos RS-485 (1-4)

Este *test* debe realizarse en modo pasarela, es decir, los *bytes* que se envíen desde un puerto deben llegar al otro. La secuencia de pasos es la siguiente:

- Configurar el control de calidad para que puentee el puerto 1 con el puerto 2, 3 o 4, a partir de ahora lo denominaremos (X).
- Configurar el puerto 1 y (X) en modo RS-485.
- Enviar *bytes* por el puerto 1, deben llegar al puerto (X).
- Enviar *bytes* por el puerto (X)., deben llegar al puerto 1.

Se repiten todas estas pruebas en todas las configuraciones posibles de resistencia de *bias* y terminación. Las combinaciones posibles son las siguientes:

- Sin resistencias.
- Solo resistencia de terminación en puerto 1. Puerto (X) sin resistencias.
- Solo resistencias de polarización en puerto 1. Puerto (X) sin resistencias.
- Resistencia de terminación y de polarización en puerto 1. Puerto (X) sin resistencias.
- Solo resistencia de terminación en puerto (X). Puerto 1 sin resistencias.
- Solo resistencias de polarización en puerto (X). Puerto 1 sin resistencias.
- Resistencia de terminación y de polarización en puerto (X). Puerto 1 sin resistencias.

(El valor X ira variando entre 2,3,4 dependiendo del puerto de comprobación)

Se debe realizar comprobaciones de comunicación en las distintas configuraciones de resistencias comentadas al inicio del subapartado y con todos los puertos del 1 al 4, que se configuraran mediante las GPIO correspondientes a las resistencias de *Pull-Up*, *Pull-Down* y terminación que se muestran en el **Anexo D**, Mapa GPIO Raspberry.

4.1.4 Puertos RS-485 (5)

En el caso del puerto RS-485 5 se realiza una comprobación distinta a las anteriores. En un principio comprobando que el resto de las pruebas anteriores son correctas ya se podría decir que la comunicación de este puerto es correcta, debido a que este se utiliza para comunicar la Raspberry con el microcontrolador de la PCB externa, por lo que si todo ha funcionado correctamente la comunicación para su ejecución también ha sido correcta. Aun así, se deben realizar estos procesos:

- Configurar el puerto 5 como maestro Modbus, *baudrate* a 9600 8N1 sin resistencia de terminación ni de polarización y leer los registros 1 al 12 del control de calidad.
- Configurar el puerto 5 con resistencia de terminación y sin polarización y leer los registros 1 al 12.
- Configurar el puerto 5 con resistencia de terminación y polarización y leer los registros 1 al 12.
- Configurar el puerto 5 sin resistencia de terminación y con polarización y leer los registros 1 al 12.

4.1.5 Outputs

Por último, se deben medir las *Outputs* de en los borneros para saber si el SmartGate está funcionando correctamente, que se ilustran en la tabla 5.

Registro	Campo	Medida	
1	*Expected VCC* **(no es medida del bornero)**	V Alimentación	Depende de la fuente que alimente (generalmente 12V)
2	VCC1	*Expected VCC*	
3	VCC2		
4	VCC3		
5	VCC4		
6	5V ISO	5V	
7	5V		
8	GND1	2.5V	
9	GND2		
10	GND3		
11	GND4		

Tabla 5. Tabla de Outputs

Una vez haya finalizado este proceso, en los laterales de la ventana se establecerán en rojo o verde los paneles referentes a las pruebas que se han realizado, cada uno con su etiqueta correspondiente, indicando si el resultado de la prueba ha sido correcto o no.

A continuación, se descargan los reportes generados en una carpeta con el número de serie del producto que se está testeando y se actualiza el Excel.

Finalmente se eliminan de la Raspberry todos los archivos cargados y generados dejándolo listo para su venta.

4.2 *Hardware* externo.

Una vez comprendido cual es la función que debe cumplir esta placa se lleva a cabo el estudio y diseño de las etapas y componentes necesarios para su correcto funcionamiento.

En el **Anexo B** se muestra el *hardware* diseñado al completo y como queda el resultado.

Estas son las etapas diseñadas:
La comunicación entre SmartGate y PCB externa se implementa mediante la etapa que se muestra en la figura 25, con la función de realizar una comunicación entre la Raspberry y el microcontrolador de la PCB externa, en la cual se presenta la entrada digital por DI, RO como salida receptora en la que se entrega la señal convertida a lógica digital y las líneas diferenciales A y B para comunicación RS-485 estabilizadas con resistencias de *Pull-Up*(R6), *Pull-Down*(R3) y terminación (R4).

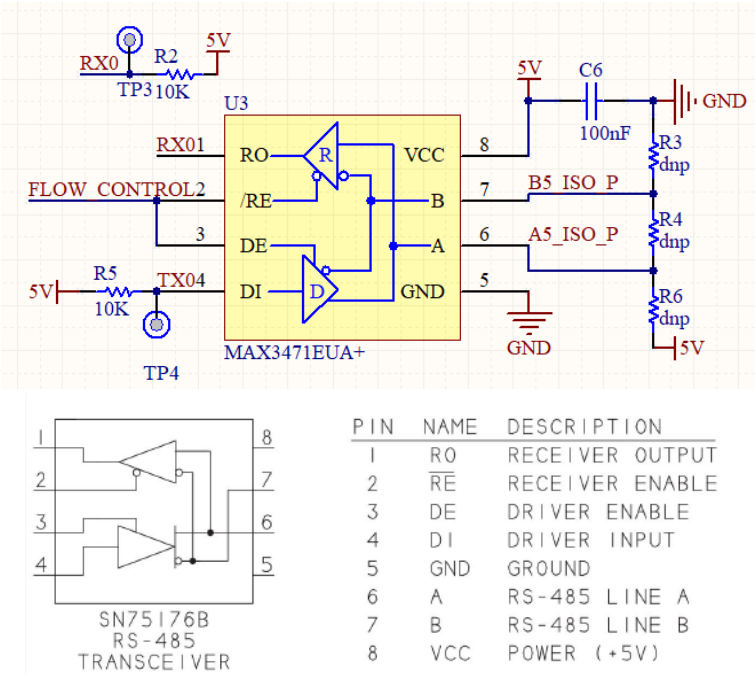


Figura 25. Etapa de comunicación entre el SmartGate y la PCB externa

La placa está diseñada con un conexionado físico de los puertos RS-232, de forma que se conecta TX1 a RX2 y TX2 a RX1 tal y como se muestra en la figura 26, para poder comprobar la comunicación de estos dos puertos.

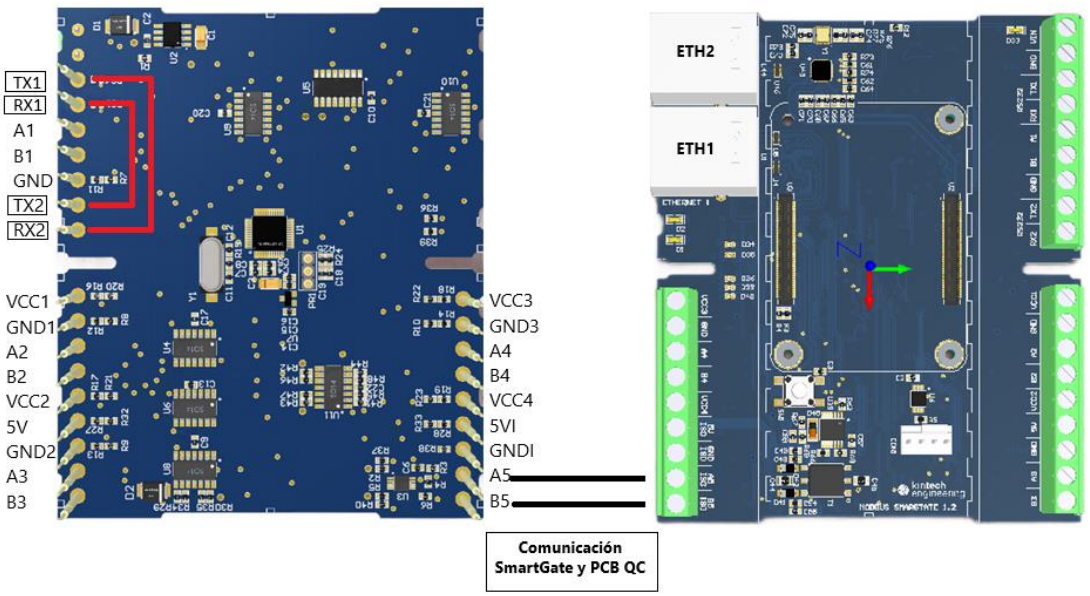
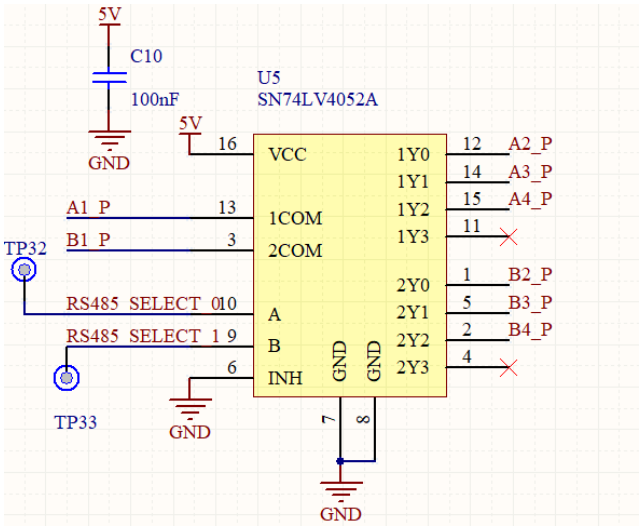


Figura 26. Conexionado físico puertos RS-232

En cuanto a los Puertos RS-485 se diseña otra etapa en la PCB externa, la cual mediante un multiplexor se conexiona el puerto 1 con los puertos 2, 3 y 4 como se muestra en la figura 27, el conexionado se determina mediante la escritura del registro 0 "Config RS485 Bridge", como se indica en la tabla siguiente, que comunica al microcontrolador del SmartGate por el puerto 5.



Config RS485 Bridge	
Valor	Enlace
0	Sin enlace
1	P1 con P2
2	P1 con P3
3	P1 con P4
Otros valores	Sin enlace

(Esta tabla indica los valores que configuran las posibles combinaciones del multiplexor)

Figura 27. Etapa de selección de conexionado de puertos RS-485

Una vez configurado el multiplexor se realiza un conexionado físico tal y como se puede apreciar en la figura 28, en la cual el color rojo muestra el conexionado de los puertos 1 y 2, el azul muestra el conexionado de los puertos 1 y 3 y el verde muestra el conexionado de los puertos 1 y 4.

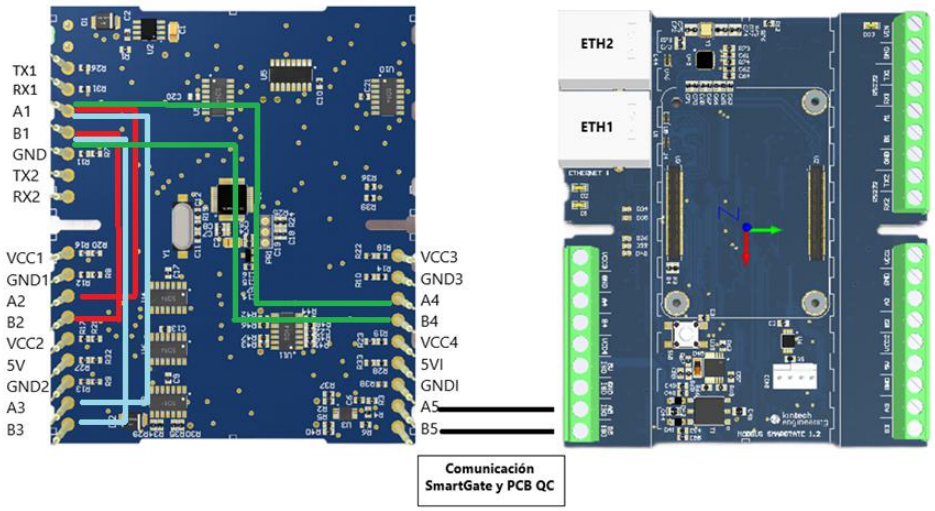


Figura 28. Conexionado físico de puertos RS-485

Lectura de los Outputs del SmartGate, los conectores del SmartGate ofrecen una serie de pines para el conexionado de sensores en los que encontramos 4 VCC´s a la tensión que se alimente el SmartGate, que irán atenuadas mediante un divisor de tensión con escala 0.1 y unido a un seguidor para medir su tensión y adaptarla a la entrada del ADC del microcontrolador, representado en la figura 30, 4 GND'S conectadas de 5V a un divisor con escala 0.5 para atenuar la señal unido a un seguidor de tensión también, representado en la figura 29, por ultimo dos señales de 5V adaptadas con un divisor de con escala 0.5 unida a un seguidor con la misma función, representado en la figura 31.

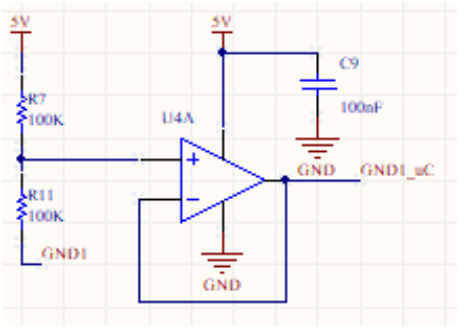


Figura 29: Divisor con escala 0.5, con seguidor para medir GNDs

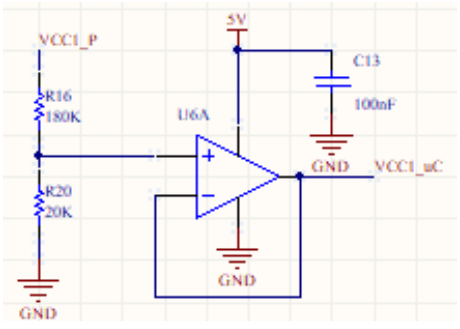


Figura 30: Divisor con escala 0.1, con seguidor para medir VCCs

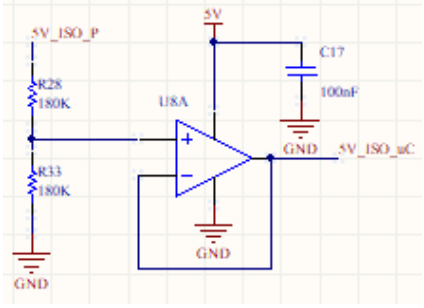


Figura 31: Divisor con escala 0.5, con seguidor para medir 5V y 5VI

Lectura de tensiones en los canales A y B de los 5 puertos RS-485, para ello se añaden 10 etapas similares a las de medida de los outputs. Se procede mediante divisores a la adaptación de las señales, posteriormente acompañadas de un seguidor y conectadas al microcontrolador para proceder a su gestión. Para los puertos 1-4 se dimensiona teniendo en cuenta que la tensión máxima podrá ser 3,3V, ilustrada en la figura 32, y para el puerto 5 5V, ilustrada en a figura 33.

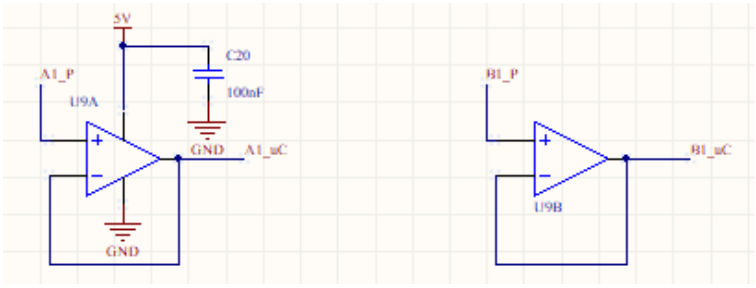


Figura 32: Seguidor para medir canales A y B puerto 1-4

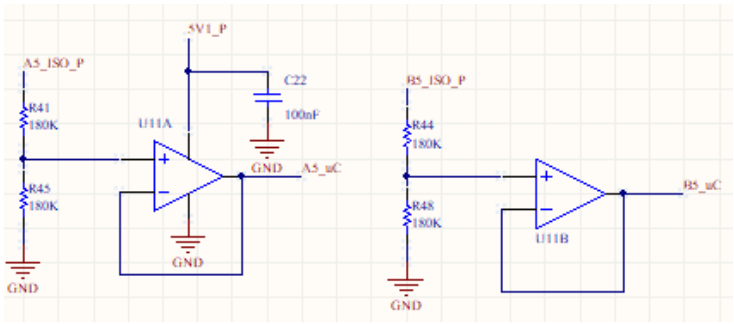


Figura 33: Divisor con escala 0.5, con seguidor para medir canales A y B puerto 5

Lectura VCC Expected, para conocer que tensión debería de haber en las VCC lo que se consigue es medir la tensión que le entra de la fuente, adaptada con un divisor con escala 0.1, protegida contra polaridad inversa con un Schottky SS23 y seguido un AO para acceder a su medida mediante el microcontrolador, como se ilustra en la figura 34.

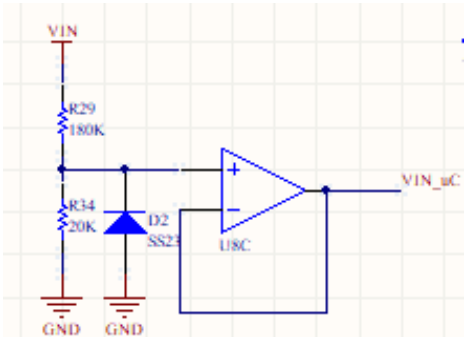


Figura 34: Divisor con escala 0.1, con seguidor para medir V Alimentación

4.3 Programación sobre Linux.

Este *software* comienza con la creación de un cliente Modbus RTU y escribiendo un 0 en el registro 0 del ADC para indicar que los puertos están desconectados unos de otros, se activan de las GPIO de todas las resistencias de *bias* y terminación de los 5 puertos RS-485 para proceder a una lectura de las tensiones referentes a cada puerto, una vez medidas, se decide que resistencias están activadas o no dependiendo de estas tensiones, tal y como se indican en las figuras 35 y 36.

```
179 cliente = ModbusSerialClient(port='/dev/ttyAMA4', framer=ModbusRtuFramer, baudrate=9600,
    bytesize=8, parity='N', stopbits=1)
180 cliente.write_registers(0, 0)
```

```
350 configurar_resistencias5(pullup5= True, pulldown5= True, terminacion5= True)
351 time.sleep(0.5)
352 configurar_resistencias1(pullup1= True, pulldown1= True, terminacion1= True)
353 time.sleep(0.5)
354 configurar_resistencias2(pullup2= True, pulldown2= True, terminacion2= True)
355 time.sleep(0.5)
356 configurar_resistencias3(pullup3= True, pulldown3= True, terminacion3= True)
357 time.sleep(0.5)
358 configurar_resistencias4(pullup4= True, pulldown4= True, terminacion4= True)
359 time.sleep(1)
```

Figura 35. Fragmento de código creación del cliente y configuración de resistencias bias y terminación

```
370 mensaje="PUERTO 1"
371 print("")
372 print(mensaje)
373 archivo.write(f"{mensaje}\n")
374 comprobar_activacion14(tensiones_canales[12],tensiones_canales[13], mensaje)    #Lee y
    compara que este dentro de los rangos necesarios para que se de una combinacion u otra.
375 mensaje="PUERTO 2"
376 print("")
377 print(mensaje)
378 archivo.write(f"{mensaje}\n")
379 comprobar_activacion14(tensiones_canales[14],tensiones_canales[15], mensaje)    #Lee y
    compara que este dentro de los rangos necesarios para que se de una combinacion u otra.
380 mensaje="PUERTO 3"
381 print("")
382 print(mensaje)
383 archivo.write(f"{mensaje}\n")
384 comprobar_activacion14(tensiones_canales[16],tensiones_canales[17], mensaje)    #Lee y
    compara que este dentro de los rangos necesarios para que se de una combinacion u otra.
385 mensaje="PUERTO 4"
386 print("")
387 print(mensaje)
388 archivo.write(f"{mensaje}\n")
389 comprobar_activacion14(tensiones_canales[18],tensiones_canales[19], mensaje)    #Lee y
    compara que este dentro de los rangos necesarios para que se de una combinacion u otra.
390 mensaje="PUERTO 5"
391 print("")
392 print(mensaje)
393 archivo.write(f"{mensaje}\n")
394 comprobar_activacion5(tensiones_canales[20],tensiones_canales[21], mensaje)     #Lee y
    compara que este dentro de los rangos necesarios para que se de una combinacion u otra.
```

Figura 36. Fragmento código comprobación resistencias de bias y terminación

Una vez comprobadas las resistencias se procede a evaluar la comunicación de los puertos RS-232, se limpian los *buffers* para que no haya valores residuales en la comunicación, se comunica de un puerto a otro y viceversa y se comparan que el mensaje enviado sea igual que el recibido, como se puede apreciar en la figura 38.

Una vez comprobado RS-232 se procede a probar RS-485, de forma idéntica a RS-232, limpiando *buffers* y comunicando ambos puertos, como diferencia en estos puertos, hay que probar todas las combinaciones posibles de resistencias de *bias* y terminación activándolas anteriormente. Como se indican en las figuras 37 y 39.

```
# Caso 1: Sin resistencias
conexion="--SIN RESISTENCIAS"
print(f"Ejecutando TEST RS485 P1 {conexion}\n")
configurar_resistencias1(pullup1= False, pulldown1= False, terminacion1= False)
configurar_resistencias2(pullup2= False, pulldown2= False, terminacion2= False)
prueba_comunicacion(uart1, uart2, conexion, puerto)

# Caso 2: Solo resistencia de terminacion en puerto 1. Puertop 2 sin resistencias
conexion="--SOLO TERMINACION"
print(f"Ejecutando TEST RS485 P1 {conexion}\n")
configurar_resistencias1(pullup1= False, pulldown1= False, terminacion1= True)
configurar_resistencias2(pullup2= False, pulldown2= False, terminacion2= False)
prueba_comunicacion(uart1, uart2, conexion, puerto)
```

Figura 37. Fragmento de código comprobación RS-485 distintas combinaciones

```
def test_rs232():
    try:
        global puerto5_ok
        # Aseguraro de limpiar los buffers antes de empezar
        uart1.reset_input_buffer()
        uart1.reset_output_buffer()
        uart2.reset_input_buffer()
        uart2.reset_output_buffer()

        mensaje = b"Hola desde puerto 1 "
        uart1.write(mensaje)
        time.sleep(1)

        data= uart2.read_all().decode('UTF-8', errors='replace').strip()
        archivo.write("\n")
        archivo.write("COMUNICACION RS-232 P1 Y P2\n")

        mensaje2 = b"Hola desde puerto 2 "
        uart2.write(mensaje2)
        time.sleep(1)

        data2= uart1.read_all().decode('UTF-8', errors='replace').strip()

        # Comprobacion si ambos mensajes coinciden (REPORTES)
        if (data == mensaje.decode('UTF-8').strip()) and (data2 == mensaje2.decode('UTF-
8').strip()):
            resultados.append('RS232 P1, OK')

        else:
            resultados.append('RS232 P1, Revisar')
```

Figura 38. Fragmento de código comprobación comunicación RS-232

```
484 def prueba_comunicacion(uart1, uart2, conexion, puerto):
485     try:
486         global conteo_ok
487         # Aseguraro de limpiar los buffers antes de empezar
488         uart1.reset_input_buffer()
489         uart1.reset_output_buffer()
490         uart2.reset_input_buffer()
491         uart2.reset_output_buffer()
492 
493         mensaje = b"Hola desde puerto P1 "
494         uart1.write(mensaje)
495         time.sleep(1)
496 
497         data= uart2.read_all().decode('UTF-8', errors='replace').strip()
498 
499         mensaje2 = f"Hola desde puerto {puerto}".encode('utf-8')
500         uart2.write(mensaje2)
501         time.sleep(1)
502 
503         data2= uart1.read_all().decode('UTF-8', errors='replace').strip()
504 
505         # Comprobar si ambos mensajes coinciden (REPORTES)
506         if (data == mensaje.decode('UTF-8').strip()) and (data2 == mensaje2.decode('UTF-
    8').strip()):
507             resultados.append(f"RS-485 P1 {conexion} , OK")
508 
509         else:
510             resultados.append(f"RS-485 P1 {conexion}, Revisar")
```

Figura 39. Fragmento de código comprobación comunicación RS-485

El puerto 5 es una excepción debido a que es el que comunica con la PCB externa, para el testeo si se realiza como en la figura 37 comprobando las distintas combinaciones de resistencias, pero a diferencia de los demás en vez de enviar y recibir mensajes en los puertos se comprueba la medida de las *Outputs, como se indica en la figura 40*.

Se leen los registros del 1 al 11 que son los referentes a las *Outputs*, y debido a que no todas miden igual se debe dividir por rangos, al igual que en todas pruebas los datos se registran en un archivo.

```
539 def leer_registros(conexion):
540     try:
541         # Leer registros de 1 a 12 (dirección 0 a 11)
542         lectura = cliente.read_holding_registers(0, 12)
543         nombres_registros = ["Config RS485 Bridge", "Expected VCC", "VCC1",
544             "VCC2", "VCC3", "VCC4", "5V ISO", "5V", "GND1", "GND2", "GND3",
545             "GND4" ]
546         resultados.append(f"RS-485 P5 {conexion}")
547         archivo.write(f'FIELD\t   ESPERADO\t     RECIBIDO\t     ESTADO\n')
548         for i in range(2, 6):
549             if (lectura.registers[1]-100) < lectura.registers[i] <
(lectura.registers[1]+100):
550                 resultados.append(f"Registro {i + 1}: OK")
551                 archivo.write(f'{nombres_registros [i]}\t   {(lectura.registers[1]+50)/100}
V\t     {lectura.registers[i]/100} V\t        OK\n')
552                 datos_output.append((nombres_registros [i] , lectura.registers[i]/100,
"OK"))
553                 conteo_ok+=1
554             else:
555                 resultados.append(f"Registro {i + 1}: Fuera de rango")
556                 archivo.write(f'{nombres_registros [i]}\t   {(lectura.registers[1]+50)/100}
V\t     {lectura.registers[i]/100} V\t   Revisar\n')
557                 datos_output.append((nombres_registros [i] , lectura.registers[i]/100 ,
"Revisar"))
558
559         for i in range(6, 12):
560             if 2400 < lectura.registers[i] < 2600:
561                 resultados.append(f"Registro {i + 1}: OK")
562                 if i<8:
563                     archivo.write(f'{nombres_registros [i]}\t   5 V\t
{lectura.registers[i]/1000*2} V\t        OK\n')
564                     datos_output.append((nombres_registros [i] , lectura.registers[i]/1000*2
, "OK"))
565                     conteo_ok+=1
566                 else:
567                     archivo.write(f'{nombres_registros [i]}\t   2.5 V\t
{lectura.registers[i]/1000} V\t        OK\n')
568                     datos_output.append((nombres_registros [i] , lectura.registers[i]/1000 ,
"OK"))
569                     conteo_ok+=1
570             else:
571                 resultados.append(f"Registro {i + 1}: Fuera de rango")
572                 if i<8:
573                     archivo.write(f'{nombres_registros [i]}\t   5 V\t
{lectura.registers[i]/1000*2} V\t  Revisar\n')
574                     datos_output.append((nombres_registros [i] , lectura.registers[i]/1000*2
, "Revisar"))
575                 else:
576                     archivo.write(f'{nombres_registros [i]}\t   2.5 V\t
{lectura.registers[i]/1000} V\t   Revisar\n')
577                     datos_output.append((nombres_registros [i] , lectura.registers[i]/1000 ,
"Revisar"))
```

Figura 40. Fragmento de código comprobación de Outputs

Para comprobar los puertos *Ethernet* se realiza un pineado a ambas IP mandando 4 paquetes de datos ICMP comprobando que se envían correctamente, como se indica en la figura 41.

```
894 def ping(host):
895     command = ['ping', '-c', '4', host]  # Para Linux o macOS
896     # command = ['ping', '-n', '4', host]  # Para Windows
897     result = subprocess.run(command, stdout=subprocess.PIPE)
898     return result.returncode == 0
```

Figura 41. Fragmento de código comprobación de comunicación Ethernet

De cada una de las pruebas se guardan los datos referentes a los resultados de la prueba para posteriormente poder utilizarlos en la generación de proyectos o en la actualización del Excel de control de stock.

4.4 *Software* de escritorio.

Su misión principal es cargar el *software* de testeo .py en la Raspberry Pi y gestionar todos los documentos que genera, así como actualizar el Excel de inventario. Se muestra el diagrama del proceso en la figura 42.

En cuanto a contenido de código, se generan 3 archivos con distintas funciones. Un primer archivo App Config que será el encargado de inicializar datos que generalmente van a mantener el mismo valor, dando la opción al usuario de no tener que ingresarlos cada vez que se ejecute y dar la opción de mantener el valor cambiado si alguna vez es necesario variarlo.

Un segundo archivo será el general y encargado de realizar todas las opciones de ejecución en el que se encuentra el main. Comienza con una lectura e inicialización de datos de entrada necesarios para la ejecución correcta del programa y testeo, guardando estas configuraciones.

A través de botones se accede a las distintas opciones del *software*, un primer botón será el que ejecuta el control del producto y el botón *Settings*, el cual da acceso a otra ventana en la que se encuentran las IP de los puertos *Ethernet* y las rutas que tendrán cada una un botón *Browse* que abre una ventana de directorios donde es posible seleccionar la ruta necesaria para el proceso.

Una vez finalizado el testeo del SmartGate se procede limpiando todos los archivos que se han subido a la Raspberry.

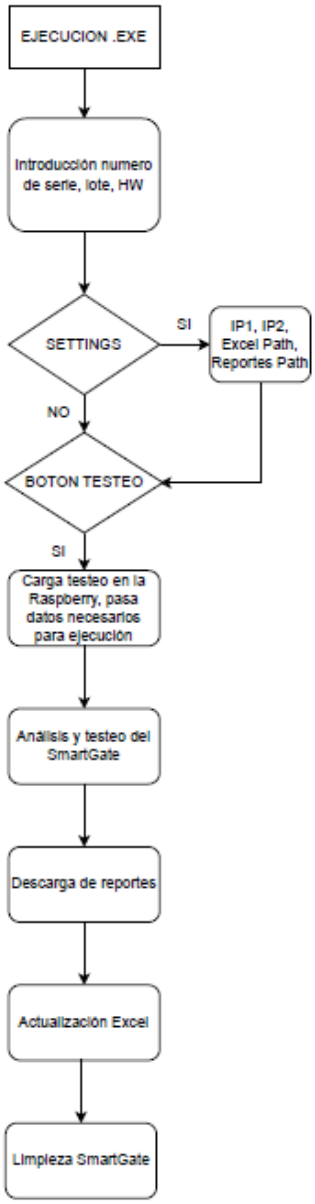


Figura 42. Diagramade flujo software de escritorio

4.5 Reportes y generación de Outputs

4.5.1 Reporte de calidad

La forma de comprobación para la generación de este reporte se realiza mediante comparación de ficheros. Es decir, conforme se va ejecutando el programa de testeo, en un archivo .csv se escriben los resultados para cada elemento a comprobar, correspondiéndoles “OK” o “Revisar”, de esta forma cuando el programa ha finalizado y el archivo está completamente escrito lo compara con un .csv previamente generado con todas las comprobaciones correctas, siendo este un archivo único que nunca se modificará, manteniendo el mismo formato que tiene el archivo que se genera cada vez que se realiza un testeo.

En caso de que esta comparación sea veraz, es decir todos los procesos de testeo han sido correctos, mediante código generara el PDF de Control de Calidad en el cual aparecerá:

- Número de serie
- Modelo del producto: "SmartGate"
- "*PASSED*", es decir que el producto ha pasado el testeo correctamente
- Fecha de realización del QC.
- Sello de la empresa
- QR con acceso a la página principal de la empresa
- Fecha y hora
- Tablas con los elementos que se han comprobado en el proceso

Se adjunta un ejemplo de Certificado de Calidad correcto en el **Anexo C**.

4.5.2 Reporte de fallos

La forma de comprobación para la generación de este reporte se realiza mediante comparación de ficheros al igual que el de calidad. Es decir, con el archivo. csv que genera conforme va ejecutando el programa, lo compara con un .csv previamente generado con todas las comprobaciones correctas, siendo este un archivo único que nunca se modificará, manteniendo el mismo formato que tiene el archivo que se genera cada vez que se realiza un testeo.

En caso de que esta comparación sea falsa, es decir alguno de los procesos de testeo han sido incorrectos, mediante código generará el PDF de Fallos en el cual aparecerá:

- Número de serie
- Puerto donde se ha encontrado el fallo
- Fallo que se asocia al puerto

Se adjunta un ejemplo de Certificado de fallos en el **Anexo C**.

4.5.3 Reporte .txt de la empresa

Con el fin de tener un registro de resultados, más allá de tener "OK" o "Revisar" se genera un fichero de texto el cual representa los valores obtenidos para cada una de las pruebas de este control.

Una vez realizadas las comprobaciones de los puertos, en este fichero se muestran los resultados para cada comunicación, cada puerto y cada combinación de resistencias en el caso de los puertos RS-485, el mensaje que envía y el que recibe.

En el caso de los *Outputs* en el puerto RS-485 5 para cada combinación de resistencias genera una tabla en la cual se muestra *FIELD* (campo a medir), valor de medida ESPERADO, valor de medida RECIBIDO y ESTADO.

Para la detección de las resistencias de *bias* y terminación se ofrece una solución similar, una tabla en la cual se encuentran las 3 resistencias cada uno de los 5 puertos RS-485 en las cuales se mostrarán, para las resistencias de *Pull-Up* el valor esperado, recibido y el estado, que es referente a la medida del canal A del puerto, para las resistencias de *Pull-Down* el valor esperado, recibido y el estado referente a la medida del canal B, y la resistencia de terminación no se representa con medidas sino únicamente con el estado.

Por último, para los dos puertos *Ethernet* muestra las IPs y si cada una de ellas está accesible o no.

Se adjunta un ejemplo de reporte de empresa correcto en el **Anexo C**.

4.5.4 Reporte en Excel para control de stocks

Para la empresa es muy importante tener un control de *stock* en el cual aparezcan todos los productos fabricados con sus características y resultados de calidad. En este Excel aparece:

- Número de serie
- Lote: Se introducirá manualmente por el operario
- MAC CM4*: Se genera automáticamente en el testeo, la función intenta abrir el archivo /sys/class/net/{interface}/address que es donde se encuentra la MAC en la interfaz de Linux y la muestra.
- HW: Se introducirá manualmente por el operario
- FW: Se genera automáticamente en el testeo, la función intenta abrir la carpeta '/remapper/process/version', escribe el valor, luego abre '/remapper/backend/version' y escribe su valor junto al anterior y por último, abre '/remapper/config/version' y une ese valor con los otros dos formando el *firmware* y lo muestra.
- Modelo: Se introducirá manualmente por el operario (K725, sin protecciones y K720 con protecciones)
- Contraseña: Se genera automáticamente la contraseña mediante un código particular de la empresa.
- Fecha: Se genera automáticamente.
- Comprobaciones (ETH1, ETH2, RS-485 P1, RS-232 P1, RS-485 P2, RS-232 P2, RS-485 P3, RS-485 P4, RS-485 P5, VOUT+GND, R. PUERTO 1, R. PUERTO 2, R. PUERTO 3, R. PUERTO 4, R. PUERTO 5)

Se adjunta un ejemplo de reporte Excel de control de stocks del proceso correcto en el **Anexo C**.

4.5.5 Reporte del QC

Para ofrecer una mayor seguridad se realiza automáticamente una captura de la ventana de comunicación del *software* al final de las ejecuciones que muestra perfectamente el resultado del testeo y queda constancia.

Todos estos reportes menos el Excel se guardan en una carpeta dentro del servidor de la empresa en la cual se ordenan en carpetas denominadas con su número de serie, y a su vez los archivos se ordenan por numerodeserie_ nombre_fecha(año/mes/día) _hora(h/min/s), manteniendo un orden lógico en caso de hacer varios testeos a un mismo producto.

A diferencia del Excel como bien digo que está separado, ya que se usa un único Excel para todo el inventario como es lógico.

Se adjunta un ejemplo de reporte de captura correcto en el **Anexo C**.

5. Verificación experimental

La verificación se ha realizado con un banco de pruebas que consiste en un SmartGate salido de fabrica al cual se le acopla el hardware de control de calidad externo tal y como se muestra en las figuras 43 y 44:

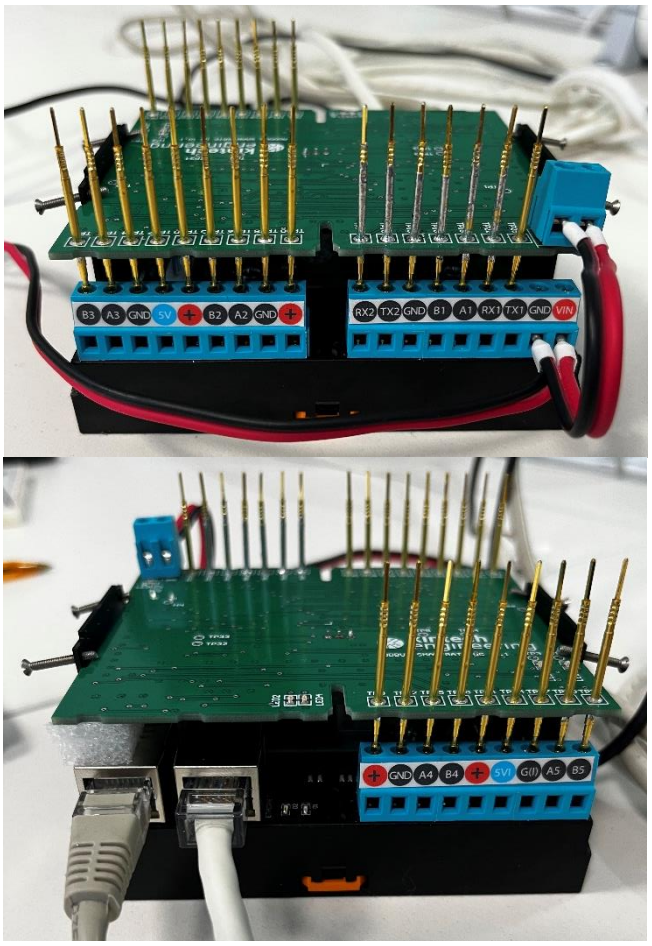


Figura 43. Referencia visual lateral del sistema de control de calidad

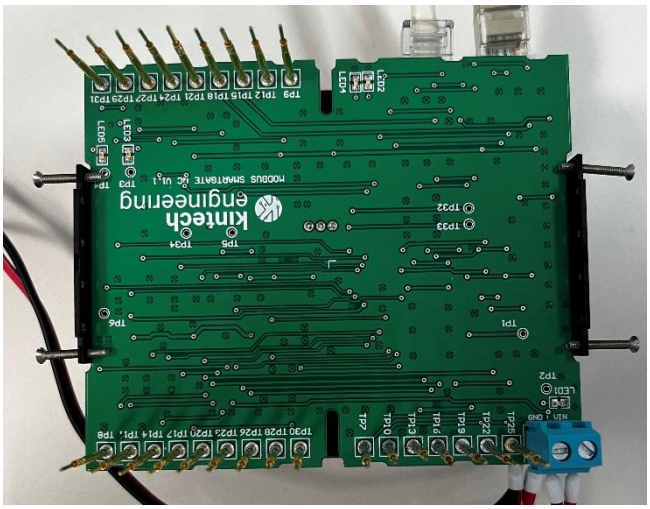


Figura 44. Referencia visual vertical del sistema de control de calidad

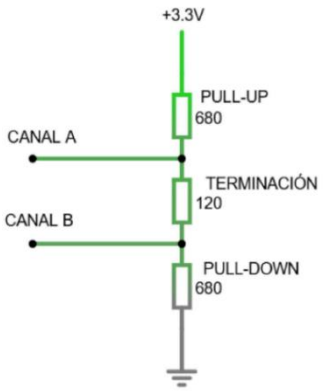
5.1 Resistencias de *bias* y terminación.

La forma de determinar el estado de las resistencias es mediante un análisis de las tensiones en los canales A y B de cada puerto RS-485. Para poder determinar estas tensiones se necesita un estudio analítico de las 8 posibles combinaciones de resistencias para comprobar las tensiones ideales que se presentan en cada canal, un estudio con un SmartGate de pruebas con acceso a poner y quitar las posibles resistencias necesarias para comprobar las situaciones y por último una comparación de estos dos con una tabla de verdad para obtener una solución.

A continuación, muestro un esquema eléctrico de la disposición que tendrían estas resistencias de valor fijo y las posibles combinaciones de activación, se ilustran detalladamente en las figuras 45, 46, 47 y 48.

Cosas a tener en cuenta:

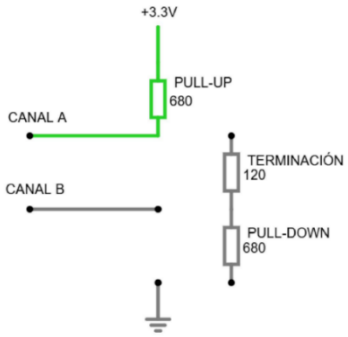
- Estos esquemas tienen por alimentación 3,3V que si es correcto para el análisis de los puertos 1 al 4, el análisis será igual para el puerto 5, a diferencia que en vez de una alimentación de 3,3V será de 5V.
- Estos valores son ideales, pero en primer lugar estimo los valores que debo leer y, en segundo lugar, para el puerto 5, se atenúan para leerlos en el microcontrolador mejor, alimentado a 5V.
- No muestro todas las combinaciones posibles en los siguientes cálculos, aunque si está realizado ese estudio que se muestra en la Tabla 6. Cuando la resistencia de *PULL-UP* no está presente o no funciona correctamente las medidas son tensiones cercanas a 0 y variables debido a la no alimentación (señales flotantes), por tanto, para una correcta determinación del estado es correcto observar cuando esta resistencia está presente y el circuito está unido a una alimentación estable.
- El estudio se realiza con valores ideales, se debe tener en cuenta caídas de tensión y LSB entre otros.



- **Todas las resistencias:**

I = 3.3V/(680ohm*2+120ohm) = 0.00338 A VA = 3.3V – 680ohm*0.00338 A = 1.784 V VB = 0V + 680ohm*0.00338 A = 1.516 V
I = 5/(680ohm*2+120ohm) = 0.00223 A VA = 5V – 680ohm*0.00223 A = 2.7016 V VB = 0V + 680ohm*0.00223 A = 2.2984 V

Figura 45: R. bias y terminación, presentes y funcionan correctamente

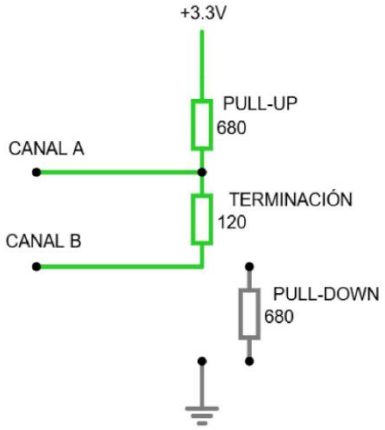


- **Solo *PULL-UP***

VA = 3.3V
VB= 0V (flotante)

VA = 5V
VB= 0V (flotante)

Figura 46: R. PULL-UP solo

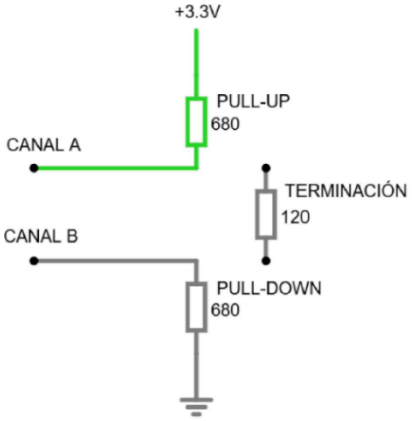


-Sin *PULL-DOWN*

VA = 3.3V
VB = 3.3V

VA = 5V
VB = 5V

Figura 47: R. PULL-UP y PULL-DOWN, presentes y funcionan correctamente



- **Sin Terminación**

VA = 3.3V
VB = 0V

VA = 5V
VB = 0V

Figura 48: R. PULL-UP y TERM, presentes y funcionan correctamente

Este estudio son cálculos ideales, por supuesto no son los valores finales que se utilizan para determinar los rangos, se debe hacer otro estudio práctico con el SmartGate una vez comprobadas las comunicaciones e implementados los ajustes de *hardware* necesario para recibir las medidas de los borneros mediante el micro.

Una vez realizada la investigación de como poder conocer si cada una de estas resistencias están presentes o no, es necesario realizar una comprobación para conocer si la tesis es correcta, sabido que de esa forma se puede conocer hay que ponerla en práctica. Para ello mediante una placa externa de pruebas y un código .py en el cual se conectan o desconectan las resistencias del SmartGate de pruebas midiendo con un *tester calibrado*, analicé las tensiones que se observan en los borneros con las distintas combinaciones pudiendo rellenar la Tabla 6.

Para los puertos 1-4:

- En caso de que en el canal A se midan 0V±tolerancia se conoce que la resistencia de *Pull-Up* no funciona correctamente y no se puede conocer el estado del resto de resistencias.

- En caso de que en el canal A se midan 1.784V±tolerancia se conoce que *Pull-Up* funciona correctamente pero se debe observar que tensión se miden en el canal B:
    - Si cae 1.516V±tolerancia se conoce que *Pull-Down* funciona correctamente y Terminación también.
    - Si no cae esa tensión entramos en una indeterminación y se debería hacer un nuevo testeo por posibles fallos en el contacto de las sondas.

- En caso de que en el canal A se midan 3,3V±tolerancia se conoce que *Pull-Up* funciona correctamente, pero se debe observar también que tensión se mide en el canal B:
    - Si cae 3.3V±tolerancia la resistencia de terminación funciona correctamente y *Pull-Down* no.
    - Si cae 0V±tolerancia se observa otra indeterminación puede darse que Terminación únicamente no funcione correctamente o que Terminación y *Pull-Down* no funcionen correctamente. Para ello se realiza un nuevo estudio, que se muestra en el **Anexo G**.

En ambas posibilidades se da el mismo caso, VA = 3.3V y VB = 0V, según el estudio realizado quitando estas resistencias en un SmartGate y analizando sus tensiones en cuestión del tiempo que se le da para su estabilización se demuestra que cuando solo falla terminación le cuesta menos estabilizarse en 3.3V al canal A (Al no haber terminación, la única carga en el bus es la capacitancia parasitaria de las líneas, lo que permite que las señales alcancen rápidamente sus valores finales) que cuando le faltan tanto polarización como *Pull-Down* (La señal B queda flotante, lo que genera un comportamiento inestable y más ruido. Además, la falta de *Pull-Down* en B introduce más desacoplamiento entre A y B, lo que ralentiza el proceso de estabilización de ambas señales).

En la tabla 6, analizando los resultados se comprueba diferentes rangos de tensión para la detección entre los puertos del 1 al 4 y el puerto 5. Lo que se aprecia es que el puerto 5 en caso de tener alguna de las líneas flotantes en vez de aparecer un nivel de tensión de 0V hay un nivel de continua de 1.5V, al ser distintos *drivers*, en este caso un *driver* de RS-485 aislado, estando en reposo debido a la alta impedancia varían los valores de detección para el correcto funcionamiento, pudiendo delimitar aun así los rangos correctos para las comprobaciones debido a los distintos estudios realizados.

Para el puerto 5:

- En caso de que en el canal A no se midan 5V±tolerancia o 2.7V±tolerancia se conoce que la resistencia *de Pull-Up* no funciona correctamente y no se puede conocer el estado del resto de resistencias.

- En caso de que en el canal A se midan 2.7V±tolerancia se conoce que *Pull-Up* funciona correctamente pero se debe observar que tensión se miden en el canal B:
    - Si cae 2.3V±tolerancia se conoce que *Pull-Down* funciona correctamente y Terminación también.
    - Si no cae esa tensión entramos en una indeterminación y se debería hacer un nuevo testeo por posibles fallos en el contacto de las sondas.

- En caso de que en el canal A se midan 5V±tolerancia se conoce que *Pull-Up* funciona correctamente, pero se debe observar también que tensión se mide en el canal B:
    - Si cae 5V±tolerancia la resistencia de terminación funciona correctamente y *Pull-Down* no.
    - Si cae 1.5V±tolerancia se observa que no funciona correctamente ni *Pull-Down* ni terminación.
    - Si cae 0V±tolerancia la resistencia de *Pull-Down* funciona correctamente y terminación no.

En la figura 49 se ilustra un ejemplo de comprobación de estas visto en terminal.

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                ESPERADO        RECIBIDO      ESTADO
PUERTO 1
PULL-UP PUERTO 1           1.77V/3.3V      1.771 V        OK
PULL-DOWN PUERTO 1           1.516 V       1.507 V        OK
TERMINACION PUERTO 1                                      OK
PUERTO 2
PULL-UP PUERTO 2           1.77V/3.3V      1.773 V        OK
PULL-DOWN PUERTO 2           1.516 V       1.505 V        OK
TERMINACION PUERTO 2                                      OK
PUERTO 3
PULL-UP PUERTO 3           1.77V/3.3V      1.772 V        OK
PULL-DOWN PUERTO 3           1.516 V       1.51 V         OK
TERMINACION PUERTO 3                                      OK
PUERTO 4
PULL-UP PUERTO 4           1.77V/3.3V      1.773 V        OK
PULL-DOWN PUERTO 4           1.516 V       1.506 V        OK
TERMINACION PUERTO 4                                      OK
PUERTO 5
PULL-UP PUERTO 5           2.702V/5V       2.626 V        OK
PULL-DOWN PUERTO 5           2.298 V       2.23 V         OK
TERMINACION PUERTO 5                                      OK
```

Figura 49: Comprobación resistencias de bias y terminación.

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	False	False	False	False	False	False
A(V)	0			1,447		
B(V)	0			1,443		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	False	False	True	False	False	True
A(V)	0			1,449		
B(V)	0			1,449		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	False	True	False	False	True	False
A(V)	0			1,446		
B(V)	0			0		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	False	True	True	False	True	True
A(V)	0			0,03		
B(V)	0			0,03		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	True	False	False	True	False	False
A(V)	3,285			4,83		
B(V)	0			1,443		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	True	True	False	True	True	False
A(V)	3,28			4,83		
B(V)	0			0		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	True	False	True	True	False	True
A(V)	3,27			4,8		
B(V)	3,26			4,79		

Resistor	up1/4	down1/4	term1/4	up5	down5	term5
Estado	True	True	True	True	True	True
A(V)	1,773			2,637		
B(V)	1,509			2,239		

Tabla 6: Medidas obtenidas del módulo externo.
(las medidas en los puertos 1 al 4 son prácticamente iguales, para que se vea correctamente la tabla elimino las columnas de 2,3,4, se muestra completamente en el **anexo F**)

5.2 Puertos RS-232

La verificación experimental de los puertos RS-232 se realiza mediante la conexión física de ambos puertos y tiene como objetivo garantizar la transmisión y recepción de datos. Para ello, se comprueba mediante una pantalla de terminal y un osciloscopio, como se puede apreciar en las figuras 50 y 51.

```
COMUNICACION RS-232 P1 Y P2
Recibido en P2: Hola desde puerto 1
Recibido en P1: Hola desde puerto 2
```

Figura 50: Comprobación comunicación puertos RS-232 terminal

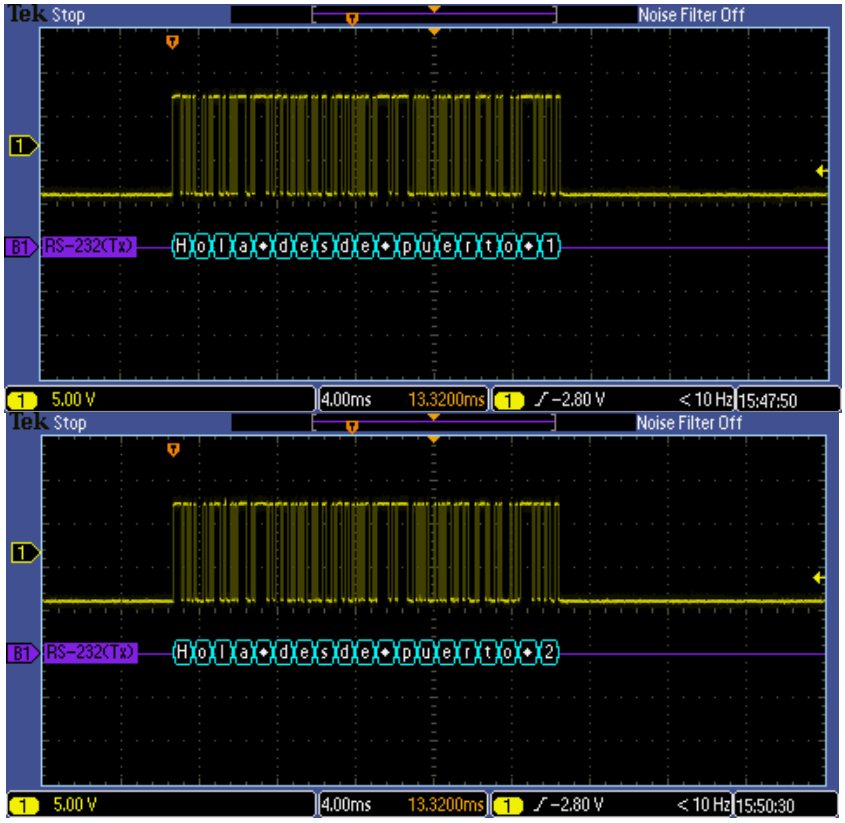


Figura 51: Comprobación comunicación puertos RS-232 osciloscopio.

5.3 Puertos RS-485

La verificación experimental de los puertos RS-485 se realiza mediante la conexión física de ambos puertos y tiene como objetivo garantizar la transmisión y recepción de datos. Para ello, se comprueba mediante una pantalla de terminal y un osciloscopio, como se ilustra en las figuras 52 y 53.

Se debe comprobar con todas las combinaciones de resistencias posibles.

```
COMUNICACION RS-485 P2
--SIN RESISTENCIAS
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO POLARIZACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--POLARIZACION Y TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
```

Figura 52: Comprobación comunicación puertos RS-485 terminal

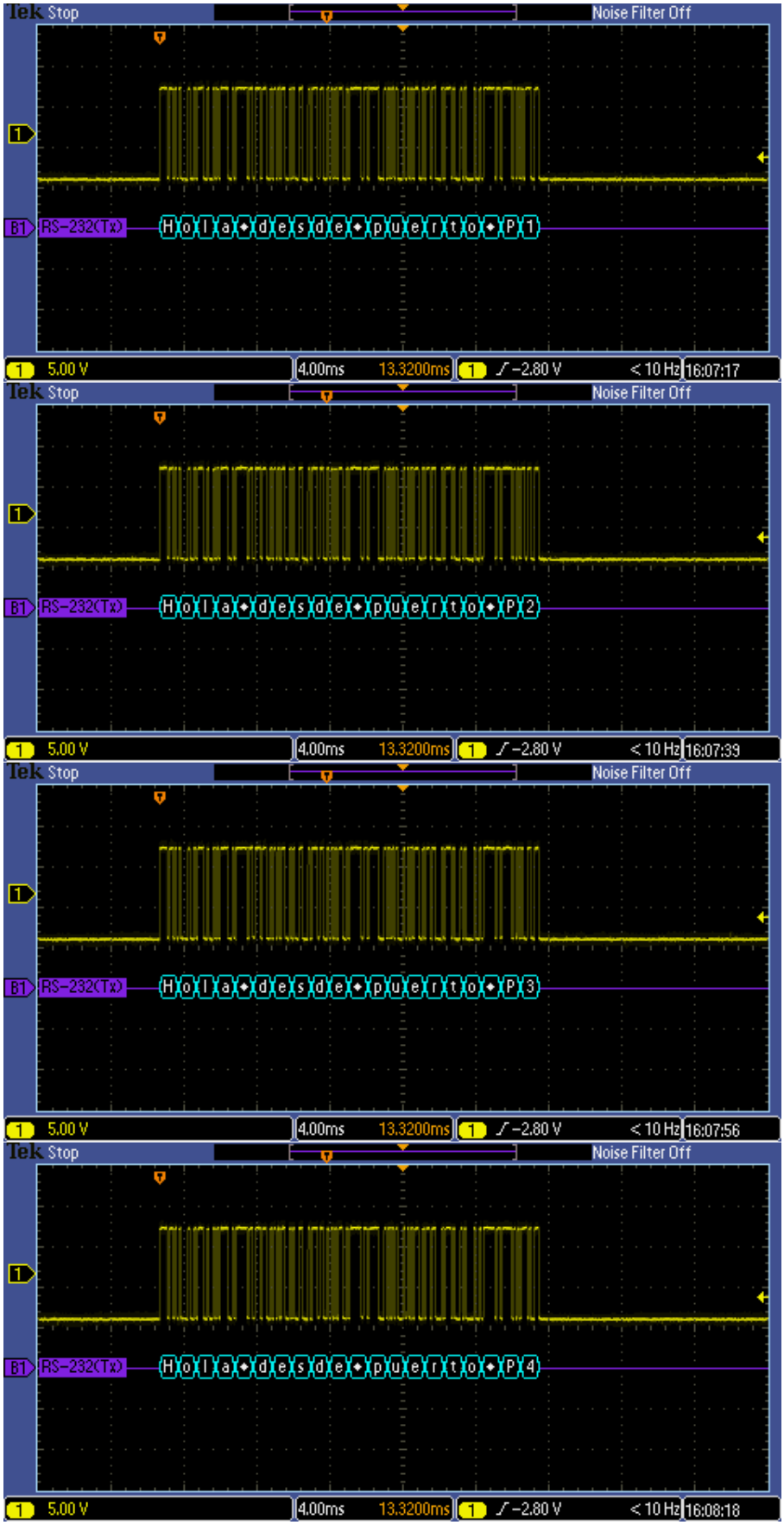


Figura 53: Comprobación comunicación puertos RS-485 osciloscopio

5.4 Outputs

La verificación de las salidas digitales del SmartGate tiene como objetivo comprobar su correcto funcionamiento en términos de niveles de voltaje. Se debe comprobar con todas las combinaciones de resistencias posibles. Se muestra un ejemplo en la figura 54 con un ejemplo de comprobación visto en terminal.

```
COMUNICACION RS485 P5
--SIN RESISTENCIAS
FIELD      ESPERADO          RECIBIDO        ESTADO
VCC1       12.19 V           12.15 V           OK
VCC2       12.19 V           12.14 V           OK
VCC3       12.19 V           12.15 V           OK
VCC4       12.19 V           12.15 V           OK
5V ISO     5 V                4.9 V            OK
5V         5 V                4.932 V          OK
GND1       2.5 V             2.507 V           OK
GND2       2.5 V             2.506 V           OK
GND3       2.5 V             2.505 V           OK
GND4       2.5 V             2.507 V           OK
--SOLO TERMINACION
FIELD      ESPERADO          RECIBIDO        ESTADO
VCC1       12.19 V           12.15 V           OK
VCC2       12.19 V           12.14 V           OK
VCC3       12.19 V           12.15 V           OK
VCC4       12.19 V           12.15 V           OK
5V ISO     5 V                4.9 V            OK
5V         5 V                4.932 V          OK
GND1       2.5 V             2.507 V           OK
GND2       2.5 V             2.506 V           OK
GND3       2.5 V             2.505 V           OK
GND4       2.5 V             2.507 V           OK
--SOLO POLARIZACION
FIELD      ESPERADO          RECIBIDO        ESTADO
VCC1       12.19 V           12.15 V           OK
VCC2       12.19 V           12.14 V           OK
VCC3       12.19 V           12.15 V           OK
VCC4       12.19 V           12.15 V           OK
5V ISO     5 V                4.9 V            OK
5V         5 V                4.932 V          OK
GND1       2.5 V             2.507 V           OK
GND2       2.5 V             2.506 V           OK
GND3       2.5 V             2.505 V           OK
GND4       2.5 V             2.507 V           OK
--TERMINACION Y POLARIZACION
FIELD      ESPERADO          RECIBIDO        ESTADO
VCC1       12.18 V           12.15 V           OK
VCC2       12.18 V           12.14 V           OK
VCC3       12.18 V           12.15 V           OK
VCC4       12.18 V           12.15 V           OK
5V ISO     5 V                4.9 V            OK
5V         5 V                4.932 V          OK
GND1       2.5 V             2.507 V           OK
GND2       2.5 V             2.506 V           OK
GND3       2.5 V             2.505 V           OK
GND4       2.5 V             2.507 V           OK
```

Figura 54: Comprobación Outputs terminal

5.5 *Ethernet*

La verificación experimental de los puertos *Ethernet* del SmartGate tiene como objetivo comprobar su correcto funcionamiento en términos de transmisión y recepción de datos. Las pruebas se llevan a cabo haciendo un ping con las IPs, tal y como se indica en la figura 55. Si el ping es correcto y se reciben todos los paquetes se marca el *test* como "OK".

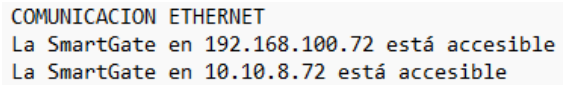


Figura 55: Comprobación puertos Ethernet terminal

6. Conclusiones y líneas futuras.

6.1 Conclusión.

El desarrollo e implementación del Sistema de Control de Calidad para SmartGate ha permitido alcanzar los objetivos planteados, estableciendo un sistema de control de calidad automático que reduce tiempos y errores humanos, mejorando así considerablemente el proceso de industrialización de este dispositivo.

Las principales conclusiones obtenidas a lo largo del proyecto se resumen a continuación:

1. Eficiencia y fiabilidad del sistema automatizado: Se ha desarrollado un sistema automatizado que ha reducido significativamente el tiempo empleado y los posibles errores debidos al proceso manual. En concreto, la integración de *hardware* para la comprobación de los puertos RS-232, RS-485 y *Ethernet* más la validación de tensiones y resistencias ha resultado ser una solución muy sólida y fiable para encontrar fallos en los dispositivos fabricados.

2. Creación de *software* intuitivo: El diseño del *software* de escritorio con una interfaz gráfica sencilla ha ayudado a los técnicos de la empresa a trabajar con el sistema. Elementos como la visualización inmediata del estado de las pruebas mediante señales visuales (verde/rojo) y la generación automática de informes han ayudado a mejorar la comprobación por parte del usuario y a agilizar el flujo de trabajo.

3. Creación de informes automatizados: Los reportes de calidad automatizados en archivos PDF, de texto y Excel garantizan que se obtendrán los resultados de cada prueba como registros completos y ordenados. Esto ayuda a la trazabilidad, al control interno y también a la comunicación externa con el cliente final mediante la posible entrega de certificados de calidad detallados.

4. Adaptabilidad del sistema: El sistema diseñado es flexible a posibles adaptaciones en futuras versiones de SmartGate. Esto incluye la posibilidad de añadir nuevos *tests* o aumentar las funcionalidades del *software* en función de las necesidades de producción.

5. Contribución a la industrialización: Agiliza el camino para una mayor industrialización del proceso de fabricación de SmartGate. La automatización de los *tests* y la estandarización de los procedimientos de verificación establecerán un control de calidad coherente y repetible, acorde con los requisitos de producción a gran escala.

6.2 Líneas Futuras.

Ha surgido una mejora a última hora que no permite su implementación debido a la falta de tiempo, pero es una posible versión mejorada del control de calidad.

Anteriormente se comenta que en caso de detección de resistencias de *bias* y terminación en el caso de detección en los puertos 1,2,3,4 se producen dos indeterminaciones como son:

- Cuando la resistencia de *Pull-Up* falla en un canal no se puede conocer el estado de las demás debido al estado flotante de las líneas. Figura 56.
- Cuando la resistencia de terminación falla o fallan a la vez terminación y *Pull-Down* se muestran las mismas tensiones en los canales y hay que hacer más pruebas para comprobar, caso el cual se ofrece una solución que funciona pero tiene una posible mejora.

La mejora que se comenta está basada en la implementación del *software* de otro conjunto de resistencias de *Pull-Up, Pull-Down* y terminación con los mismos valores que los que se implementan ya, conectadas a un switch que permita cada vez que se detecte un fallo en las resistencias del SmartGate conectar las nuevas resistencias a la línea, que si se conoce que están implementadas y actúan correctamente, y poder realizar otro análisis.

Estas resistencias adicionales podrían activarse con un switch activado mediante código, añadiendo al *software* de escritorio un botón para cada resistencia que pudiera manejar esta variación.

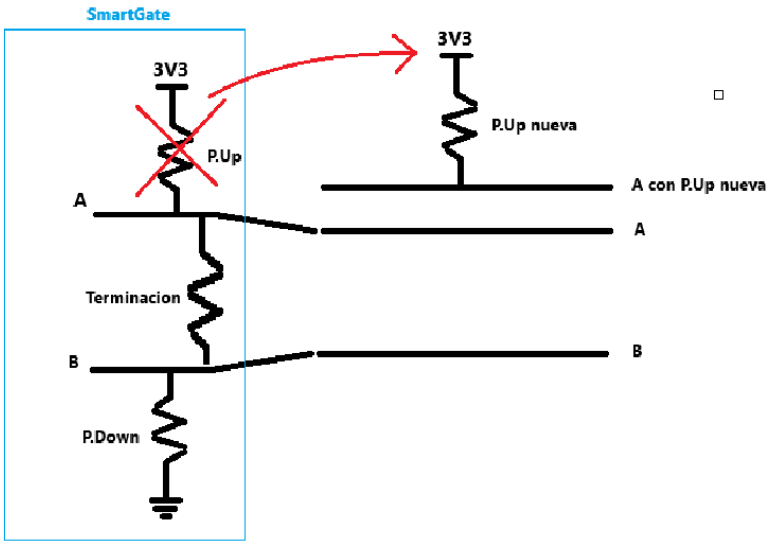


Figura 56. Variación en caso de que P.Up no funcione correctamente

En caso de la segunda indeterminación podría ser suficiente únicamente con comprobar con una nueva terminación, si así funciona correctamente es la que se debe cambiar como se muestra en la figura 57, en caso de que persista el error se demostraría que estaban fallando las dos y se verán valores de tensión de 3V3 en ambos canales como cuando solo falla *Pull-Down*. En caso de comprobar añadiendo terminación y *Pull-Down* conjuntamente podría ser que solo fallara 1 de ellas y se realizaría un cambio innecesario de las 2.

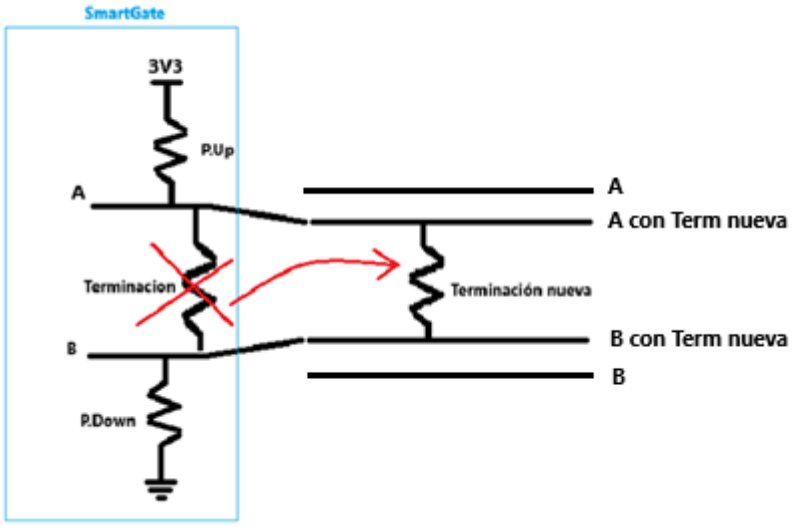


Figura 57. Variación en caso de que no funcione terminación o terminación y P.Down

6.3 Conclusión personal.

La realización de este TFG me ha permitido experimentar la dificultad de crear un nuevo "producto" considerando el cumplimiento de plazos, desarrollo de *hardware* y *software* y automatización de procesos industriales. En consecuencia, he podido comprobar, consolidar y ampliar los conocimientos adquiridos durante la etapa estudiantil en la Universidad de Zaragoza. Este proyecto no solo ha contribuido en mi formación técnica, sino también en la adquisición de visión para aplicar mis conocimientos en futuros proyectos.

Así mismo, he podido formarme y manejar nuevos conceptos teniendo la posibilidad de trabajar con un conjunto de ingenieros y técnicos, con una trayectoria consolidada y que me han brindado un apoyo excepcional.

7. Bibliografía

[1] *Datasheet SmartGate Kintech Instruments. (s/f). Kintech-engineering.com. Recuperado el 26 de diciembre de 2024, de https://www.kintech-engineering.com/pdf_docs/EN_Smartgate_K720.pdf*

[2] *Raspberry Pi -- Compute Module 4*. (s/f). Raspberry Pi. Recuperado el 26 de diciembre de 2024, de https://www.raspberrypi.com/products/compute-module-4/?variant=raspberry-pi-cm4001000

[3] Zeng, X. (2022, junio 27). *What is the Raspberry Pi Compute Module 4 (CM4)?* PiCockpit; PiCockpit.com. https://picockpit.com/raspberry-pi/what-is-the-raspberry-pi-compute-module-4-cm4/

[4] *ESCUELA POLITÉCNICA NACIONAL*. (s/f). *Diseño y construcción de un sistema didáctico de comunicación industrial bajo el protocolo Modbus. Edu.Ec. Retrieved December 30, 2024, from https://bibdigital.epn.edu.ec/bitstream/15000/826/1/CD-1201.pdf*

[5] Axelson, Janet L. (2007) *Serial Port Complete: COM Ports, USB Virtual COM Portsand Ports for Embedded Systems (2ª ed.)*. Lakeview Research LLC

[6] *Comunicación con RS-485 y Modbus*. (s/f). Rua.Ua.Es. Retrieved December 30, 2024, from https://rua.ua.es/dspace/bitstream/10045/18990/1/AA-p3.pdf

[7] Chengdu Ebyte Electronic Technology Co. ,Ltd. (s/f). *Explicación detallada del conocimiento de RS-485 y el principio de conversión de señales RS-485 a Ethernet*. Es-ebyte.com. Retrieved December 31, 2024, from https://www.es-ebyte.com/news/521

[8] Chengdu Ebyte Electronic Technology Co. ,Ltd. (s/f). *Why does RS485 need to add a pull-down resistor*. Es-ebyte.com. Retrieved December 31, 2024, from https://www.cdebyte.com/news/547

[9] Weis, O. (2024, October 3). *Diferencia entre RS-232 y RS-485*. Virtual Serial Port Driver; Electronic Team. https://www.virtual-serial-port.org/es/article/what-is-serial-port/rs232-vs-rs485.html

8. Anexos

A. Glosario

DataLogger: Dispositivo electrónico diseñado para registrar y almacenar datos automáticamente a lo largo del tiempo, generalmente provenientes de instrumentos.

Slope: Factor de escala que relaciona el cambio en la salida del sensor.

Offset: Constante que corrige la salida del sensor cuando la magnitud medida es cero.

GPIO: Pin configurable en un microcontrolador o Raspberry que puede actuar como entrada o salida digital, dependiendo de la configuración del *software*.

Driver: Es un integrado que controla dispositivos, convirtiendo señales de bajo voltaje en las adecuadas para su funcionamiento.

MAC: Es un código único que identifica a cada dispositivo capaz de conectarse a una red. Permite que los dispositivos se reconozcan y se comuniquen dentro de una red local.

B. Diagramas eléctricos del sistema y PCB

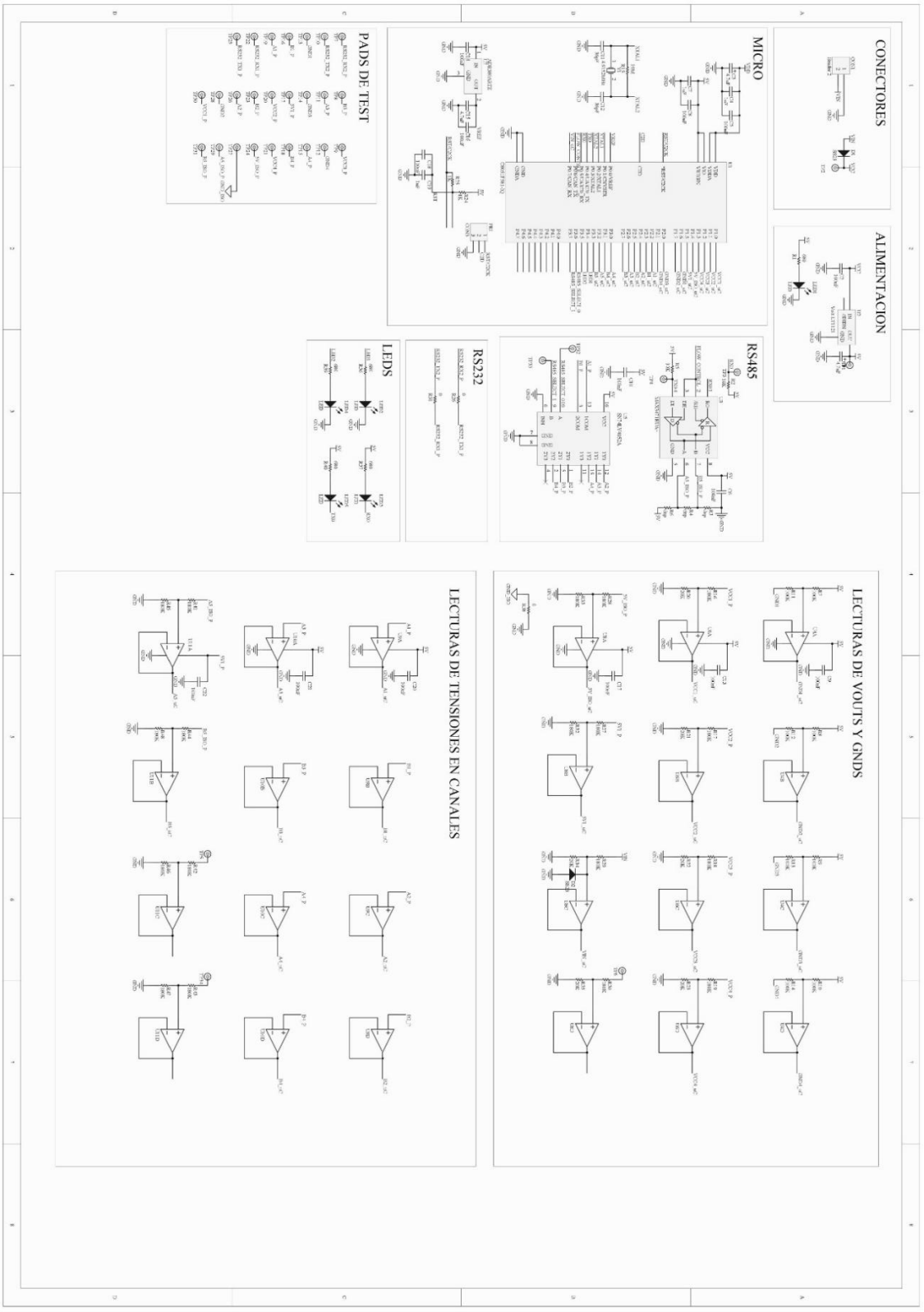


Figura 58: Esquemático PCB externa

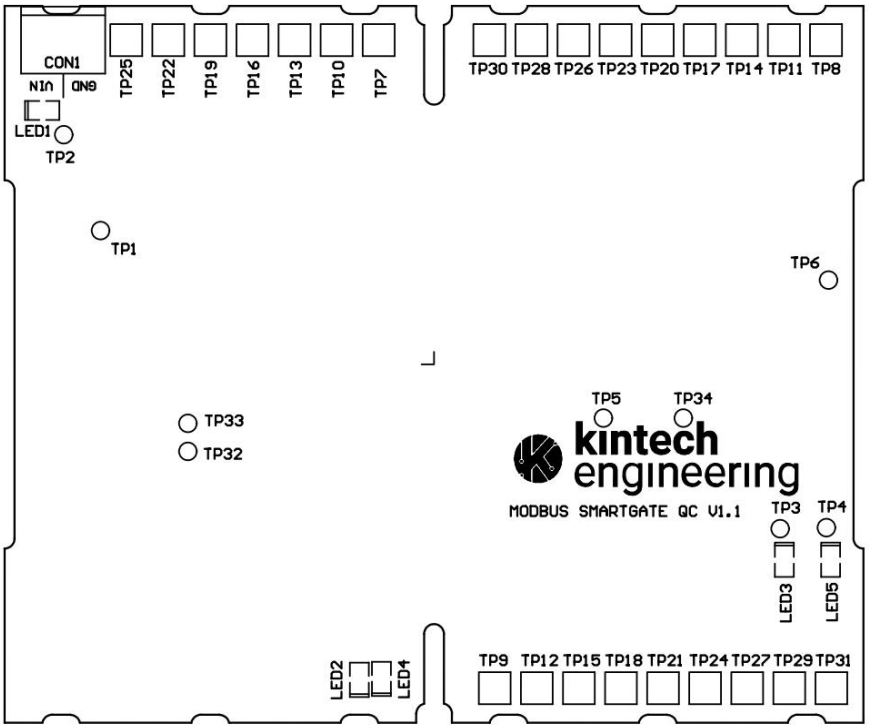


Figura 59: PCB externa 2D (Top Overlay)

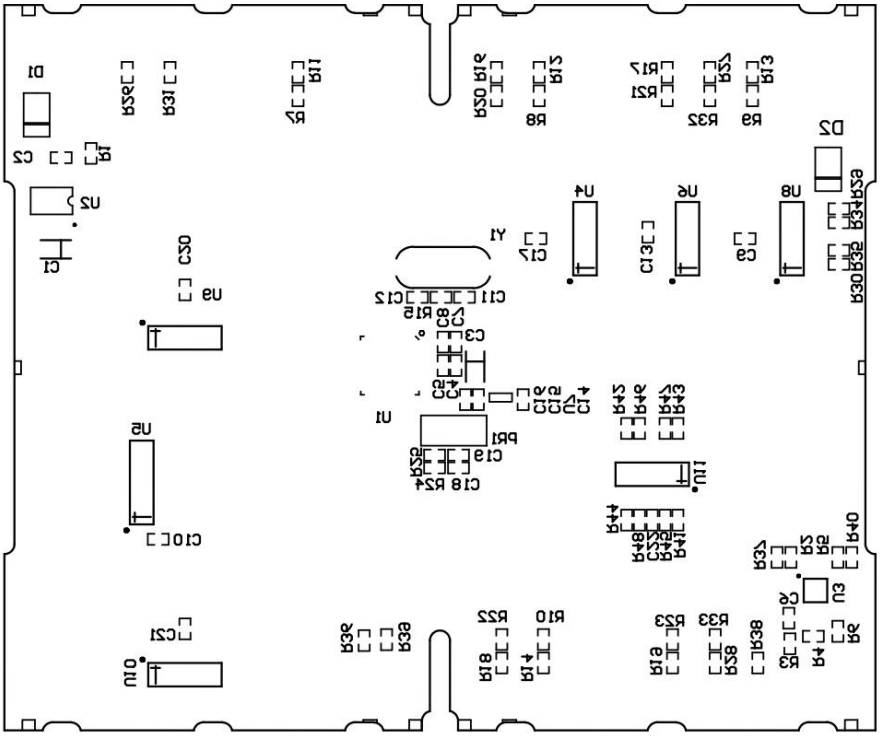


Figura 60: PCB externa 2D (Bottom Overlay)

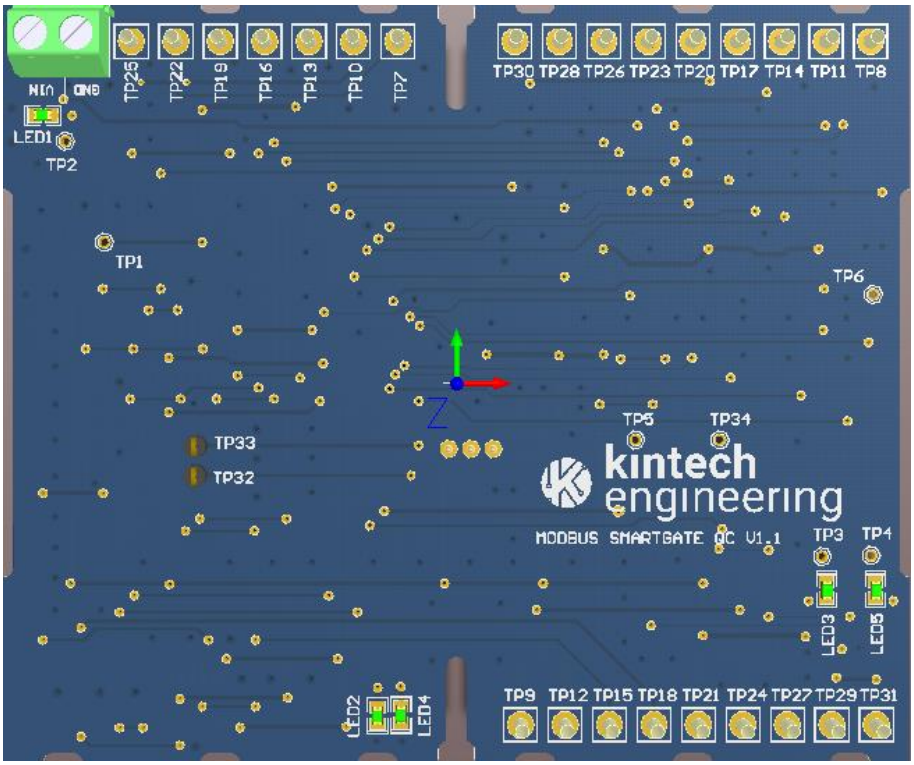


Figura 61: PCB externa 3D (Cara Top)

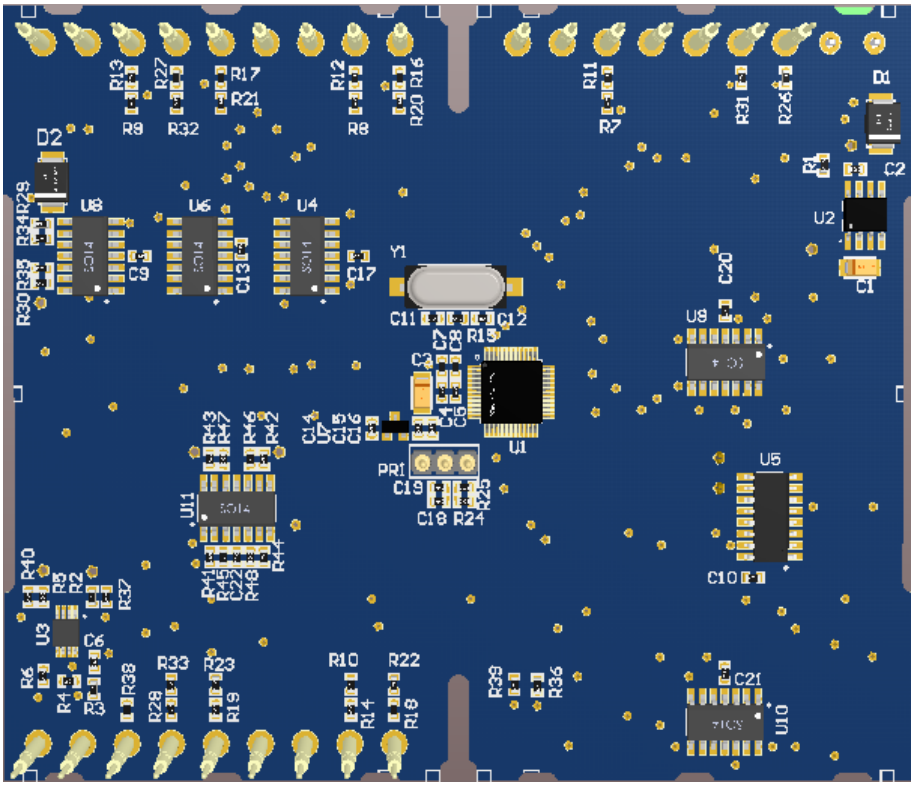


Figura 62: PCB externa 3D (Cara Bottom)

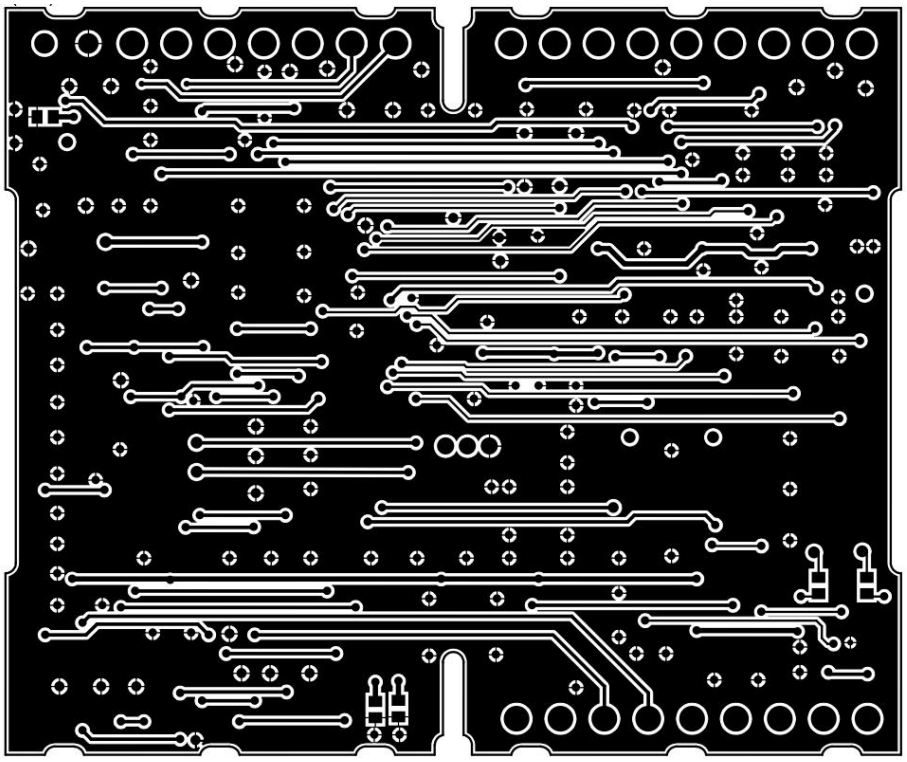


Figura 63: PCB externa 2D (Top Layer)

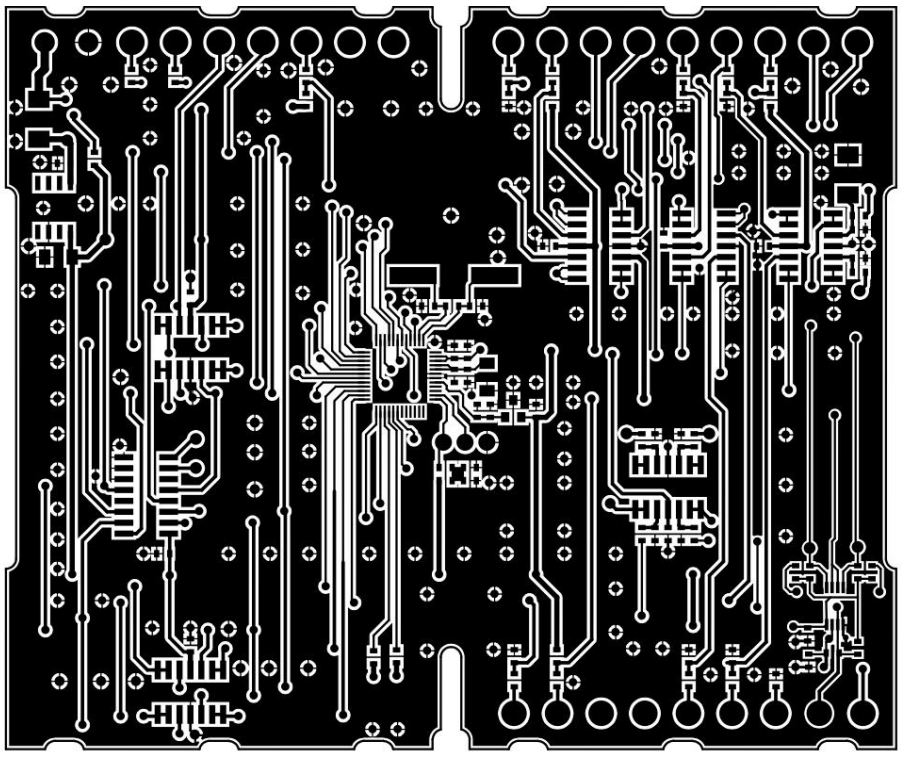


Figura 64: PCB externa 2D (Bottom Layer)

C. Ejemplos de reportes generados

REPORTE CONTROL DE CALIDAD

SMARTGATE SERIAL NUMBER: 7705003042

Quality Certificate

SMARTGATE

Manufactured by Kintech Engineering

Overall Test Result:

PASSED

The SmartGate was tested on the following date: 03/12/2024

This quality control process is to ensure that all SmartGates fully comply with our technical specification and high quality standards.

Access to
our website:
kintech-engineering.com

kintech
engineering

Date: 03/12/2024 Hour: 11:20:27

Serial Number: 7705003042

RS-485 COMMUNICATION

	PORT 1	PORT 2	PORT 3	PORT 4	PORT 5
WITHOUT RESISTORS	OK	OK	OK	OK	OK
ONLY TERMINATION	OK	OK	OK	OK	OK
ONLY BIAS	OK	OK	OK	OK	OK
ALL RESISTOR	OK	OK	OK	OK	OK

RS-232 COMMUNICATION

RS-232 1	RS-232 2
OK	OK

ETHERNET COMMUNICATION

ETHERNET 1	ETHERNET 2
OK	OK

OUTPUT TEST

	Config	Expected VCC	VCC1	VCC2	VCC3	VCC4
WITHOUT RESISTORS	OK	OK	OK	OK	OK	OK
ONLY TERMINATION	OK	OK	OK	OK	OK	OK
ONLY BIAS	OK	OK	OK	OK	OK	OK
ALL RESISTOR	OK	OK	OK	OK	OK	OK

	5V ISO	5V	GND1	GND2	GND3	GND4
WITHOUT RESISTORS	OK	OK	OK	OK	OK	OK
ONLY TERMINATION	OK	OK	OK	OK	OK	OK
ONLY BIAS	OK	OK	OK	OK	OK	OK
ALL RESISTOR	OK	OK	OK	OK	OK	OK

Date: 03/12/2024 Hour: 11:20:27

Serial Number: 7705003042

BIAS RESISTORS ACTIVATION

	PORT RS485-1	PORT RS485-2	PORT RS485-3	PORT RS485-4	PORT RS485-5
PULL-UP	OK	OK	OK	OK	OK
PULL-DOWN	OK	OK	OK	OK	OK
TERMINATION	OK	OK	OK	OK	OK

REPORTE FALLOS

Fallos en el control de calidad SmartGate -- Serial Number: 2558003043

Puerto	Estado
PULL-UP PUERTO 1	**Revisar**
PULL-DOWN PUERTO 1	**Sin PULL-UP no se puede saber**
TERMINACION PUERTO 1	**Sin PULL-UP no se puede saber**
TERMINACION PUERTO 2	**Revisar**
PULL-DOWN PUERTO 3	**Revisar**
PULL-DOWN PUERTO 4	**Revisar**
TERMINACION PUERTO 4	**Revisar**

(Ejemplo cualquiera de fallos provocados)

REPORTE DE LA EMPRESA .TXT

```
COMUNICACION RS-232 P1 Y P2
Recibido en P2: Hola desde puerto 1 -> OK
Recibido en P1: Hola desde puerto 2 -> OK

COMUNICACION RS-485 P1
--SIN RESISTENCIAS
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO POLARIZACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--POLARIZACION Y TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK

COMUNICACION RS-485 P2
--SIN RESISTENCIAS
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--SOLO POLARIZACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK
--POLARIZACION Y TERMINACION
Recibido en P2: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P2 -> OK

COMUNICACION RS485 P1 Y P3
--SIN RESISTENCIAS
Recibido en P3: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P3 -> OK
--SOLO TERMINACION
Recibido en P3: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P3 -> OK
--SOLO POLARIZACION
Recibido en P3: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P3 -> OK
--POLARIZACION Y TERMINACION
Recibido en P3: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P3 -> OK

COMUNICACION RS485 P1 Y P4
--SIN RESISTENCIAS
Recibido en P4: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P4 -> OK
--SOLO TERMINACION
Recibido en P4: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P4 -> OK
--SOLO POLARIZACION
Recibido en P4: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P4 -> OK
--POLARIZACION Y TERMINACION
Recibido en P4: Hola desde puerto P1 -> OK
Recibido en P1: Hola desde puerto P4 -> OK
```

COMUNICACION RS485 P5
--SIN RESISTENCIAS

FIELD	ESPERADO	RECIBIDO	ESTADO
VCC1	12 V	12.17 V	OK
VCC2	12 V	12.17 V	OK
VCC3	12 V	12.17 V	OK
VCC4	12 V	12.15 V	OK
5V ISO	5 V	4.874 V	OK
5V	5 V	4.992 V	OK
GND1	2.5 V	2.502 V	OK
GND2	2.5 V	2.505 V	OK
GND3	2.5 V	2.505 V	OK
GND4	2.5 V	2.507 V	OK

--SOLO TERMINACION

FIELD	ESPERADO	RECIBIDO	ESTADO
VCC1	12 V	12.15 V	OK
VCC2	12 V	12.15 V	OK
VCC3	12 V	12.17 V	OK
VCC4	12 V	12.15 V	OK
5V ISO	5 V	4.874 V	OK
5V	5 V	4.992 V	OK
GND1	2.5 V	2.502 V	OK
GND2	2.5 V	2.505 V	OK
GND3	2.5 V	2.505 V	OK
GND4	2.5 V	2.507 V	OK

--SOLO POLARIZACION

FIELD	ESPERADO	RECIBIDO	ESTADO
VCC1	12 V	12.15 V	OK
VCC2	12 V	12.15 V	OK
VCC3	12 V	12.15 V	OK
VCC4	12 V	12.14 V	OK
5V ISO	5 V	4.878 V	OK
5V	5 V	4.992 V	OK
GND1	2.5 V	2.502 V	OK
GND2	2.5 V	2.505 V	OK
GND3	2.5 V	2.505 V	OK
GND4	2.5 V	2.507 V	OK

--TERMINACION Y POLARIZACION

FIELD	ESPERADO	RECIBIDO	ESTADO
VCC1	12 V	12.15 V	OK
VCC2	12 V	12.15 V	OK
VCC3	12 V	12.15 V	OK
VCC4	12 V	12.14 V	OK
5V ISO	5 V	4.878 V	OK
5V	5 V	4.992 V	OK
GND1	2.5 V	2.502 V	OK
GND2	2.5 V	2.5 V	OK
GND3	2.5 V	2.5 V	OK
GND4	2.5 V	2.505 V	OK

COMPROBACION RESISTENCIAS BIAS Y TERMINACION

RESISTENCIA	ESPERADO	RECIBIDO	ESTADO
PULL-UP PUERTO 1	1.784 V	1.77 V	OK
PULL-DOWN PUERTO 1	1.516 V	1.52 V	OK
TERMINACION PUERTO 1			OK
PULL-UP PUERTO 2	1.784 V	1.77 V	OK
PULL-DOWN PUERTO 2	1.516 V	1.52 V	OK
TERMINACION PUERTO 2			OK

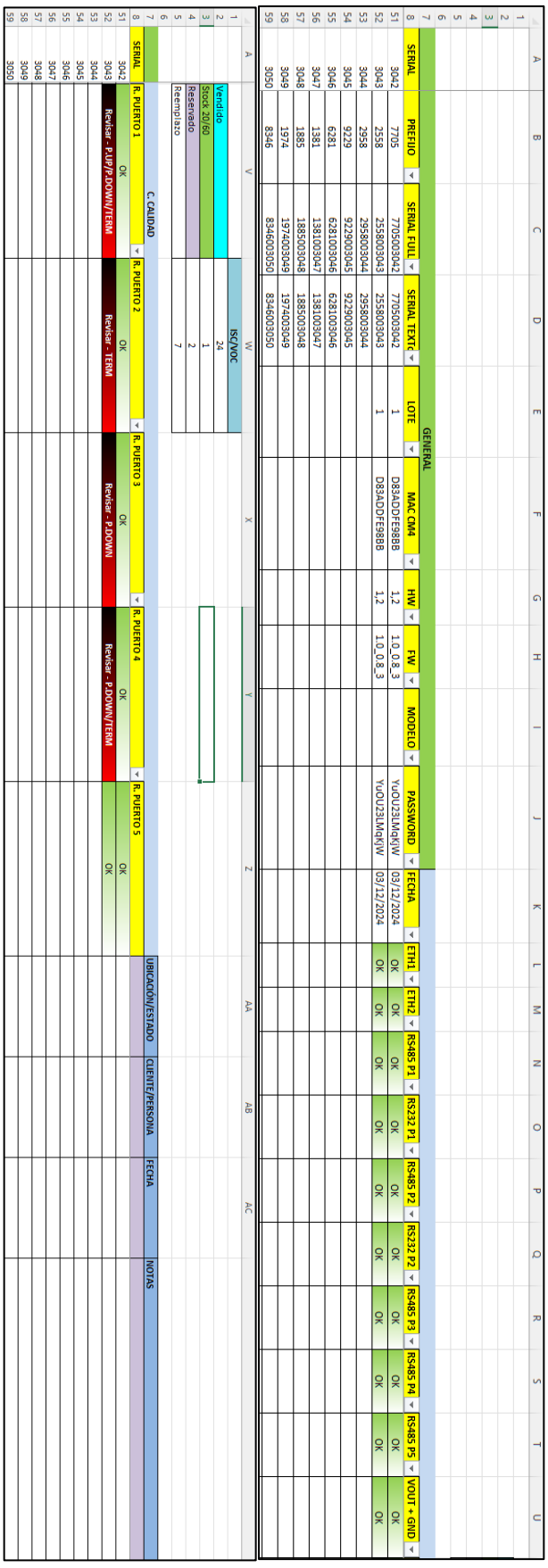
```
PULL-UP PUERTO 3          1.784 V             1.77 V     OK
PULL-DOWN PUERTO 3        1.516 V             1.52 V     OK
TERMINACION PUERTO 3                                     OK

PULL-UP PUERTO 4          1.784 V             1.77 V     OK
PULL-DOWN PUERTO 4        1.516 V             1.52 V     OK
TERMINACION PUERTO 4                                     OK

PULL-UP PUERTO 5          2.702 V             2.7 V      OK
PULL-DOWN PUERTO 5        2.298 V             2.3 V      OK
TERMINACION PUERTO 5                                     OK

COMUNICACION ETHERNET
La SmartGate en 192.168.100.72 está accesible
La SmartGate en 192.168.100.72 está accesible
```

REPORTE EXCEL INVENTARIO



SERIAL	PREFIJO	SERIAL FULL	SERIAL TEXT	LOTE	MAC CM4	HW	FW	MODELO	PASSWORD	FECHA	ETH1	ETH2	RS485 P1	RS232 P1	RS485 P2	RS232 P2	RS485 P3	RS485 P4	RS485 P5	VOUT + GND
3042	7705	7705003042	7705003042	1	D83ADDFE88B	1.2	1.0_0.8_3		YuOU23LMqKjW	03/12/2024	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
3043	2558	2558003043	2558003043	1	D83ADDFE88B	1.2	1.0_0.8_3		YuOU23LMqKjW	03/12/2024	OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
3044	2958	2958003044	2958003044																	
3045	9229	9229003045	9229003045																	
3046	6281	6281003046	6281003046																	
3047	1381	1381003047	1381003047																	
3048	1885	1885003048	1885003048																	
3049	1974	1974003049	1974003049																	
3050	8346	8346003050	8346003050																	

ISC/VOC
24
1
2
7

Vendido
Stock 20/60
Reservado
Reemplazo

SERIAL	R. PUERTO 1	R. PUERTO 2	R. PUERTO 3	R. PUERTO 4	R. PUERTO 5	C. CALIDAD	UBICACIÓN/ESTADO	CLIENTE/PERSONA	FECHA	NOTAS
3042	OK	OK	OK	OK	OK					
3043	Revisar - P.UP/P.DOWN/TERM	Revisar - TERM	Revisar - P.DOWN	Revisar - P.DOWN/TERM	OK					
3044					OK					
3045										
3046										
3047										
3048										
3049										
3050										

REPORTE EXCEL INVENTARIO

	A	B	C	D	E	F	G	H	I	J	K
7		GENERAL									
8	SERIAL	PREFIJO	SERIAL FULL	SERIAL TEXTO	LOTE	MAC CM4	HW	FW	MODELO	PASSWORD	FECHA
51	3042	7705	7705003042	7705003042	1	D83ADDFE98BB	1,2	1.0_0.8_3		YuOU23LMqKjW	03/12/2024
52	3043	2558	2558003043	2558003043	1	D83ADDFE98BB	1,2	1.0_0.8_3		YuOU23LMqKjW	03/12/2024
53	3044	2958	2958003044	2958003044							
54	3045	9229	9229003045	9229003045							
55	3046	6281	6281003046	6281003046							
56	3047	1381	1381003047	1381003047							
57	3048	1885	1885003048	1885003048							
58	3049	1974	1974003049	1974003049							
59	3050	8346	8346003050	8346003050							
60	3051	4028	4028003051	4028003051							
61	3052	9991	9991003052	9991003052							
62	3053	5825	5825003053	5825003053							
63	3054	1317	1317003054	1317003054							
64	3055	3328	3328003055	3328003055							
65	3056	2553	2553003056	2553003056							
66	3057	3836	3836003057	3836003057							
67	3058	3450	3450003058	3450003058							
68	3059	2976	2976003059	2976003059							
69	3060	9238	9238003060	9238003060							

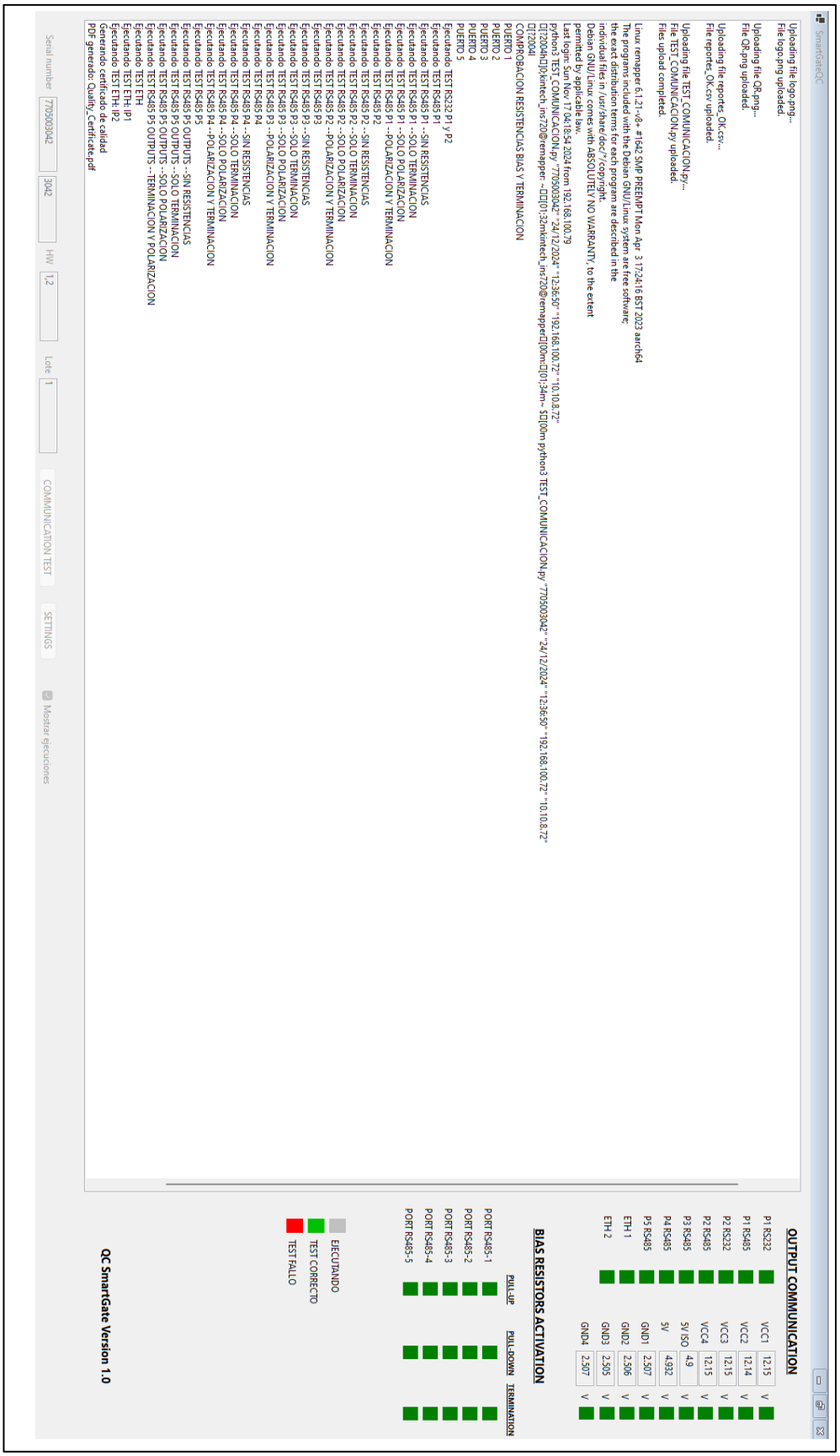
L	M	N	O	P	Q	R	S	T	U
ETH1	ETH2	RS485 P1	RS232 P1	RS485 P2	RS232 P2	RS485 P3	RS485 P4	RS485 P5	VOUT + GND
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK
OK	OK	OK	OK	OK	OK	OK	OK	OK	OK

	V	W
1		ISC/VOC
2	Vendido	24
3	Stock 20/60	1
4	Reservado	2
5	Reemplazo	7

A	V	W	X	Y	Z
7	C. CALIDAD				
8 SERIAL	R. PUERTO 1	R. PUERTO 2	R. PUERTO 3	R. PUERTO 4	R. PUERTO 5
51 3042	OK	OK	OK	OK	OK
52 3043	Revisar - P.UP/P.DOWN/TERM	Revisar - TERM	Revisar - P.DOWN	Revisar - P.DOWN/TERM	OK
53 3044					
54 3045					
55 3046					
56 3047					
57 3048					
58 3049					
59 3050					
60 3051					
61 3052					
62 3053					
63 3054					
64 3055					
65 3056					
66 3057					
67 3058					
68 3059					
69 3060					

Aquí se puede apreciar más detalladamente el contenido del Excel.

REPORTE CAPTURA DE PANTALLA DEL SOFTWARE DE ESCRITORIO



D. Mapa GPIO Raspberry

PIN CM4	GPIO	Pull	Function on Remapper	Descripcion	VALOR = 0	VALOR = 1
36	GPIO0	High	TX2	UART 2 TX	-	-
35	GPIO1	High	RX2	UART 2 RX	-	-
58	GPIO2	High	INPUT DEFAULT SETTINGS	RESETEAR LA IP A SUS VALORES POR DEFECTO		
56	GPIO3	High	R_PULLDOWN_ON_4	ACTIVACION RESISTENCIA PULL DOWN RS485 PORT 4	R PULLDOWN OFF	R PULLDOWN ON
54	GPIO4	High	TX3	UART 3 TX	-	-
34	GPIO5	High	RX3	UART 3 RX	-	-
30	GPIO6	High	R_PULLDOWN_ON_5	ACTIVACION RESISTENCIA PULL DOWN RS485 PORT 5	R PULLDOWN OFF	R PULLDOWN ON
37	GPIO7	High	RS232_485_SELECT_1	SELECCIÓN RS232/R485 UART 1	RS232 SELECCIONADO	RS485 SELECCIONADO
39	GPIO8	High	TX4	UART 4 TX	-	-
40	GPIO9	Low	RX4	UART 4 RX	-	-
44	GPIO10	Low	R_TERM_ON_3	ACTIVACION RESISTENCIA TERMINACION 120OHM RS485 PORT 3	R120 OFF	R120 ON
38	GPIO11	Low	R_PULLUP_ON_3	ACTIVACION RESISTENCIA PULL UP RS485 PORT 3	R PULLUP OFF	R PULLUP ON
31	GPIO12	Low	TX5	UART 5 TX	-	-
28	GPIO13	Low	RX5	UART 5 RX	-	-
55	GPIO14	Low	TX1	UART 1 TX	-	-
51	GPIO15	Low	RX1	UART 1 RX	-	-
29	GPIO16	Low	R_PULLDOWN_ON_1	ACTIVACION RESISTENCIA PULL DOWN RS485 PORT 1	R PULLDOWN OFF	R PULLDOWN ON
50	GPIO17	Low	R_TERM_ON_4	ACTIVACION RESISTENCIA TERMINACION 120OHM RS485 PORT 4	R120 OFF	R120 ON
49	GPIO18	Low	R_PULLDOWN_ON_2	ACTIVACION RESISTENCIA PULL DOWN RS485 PORT 2	R PULLDOWN OFF	R PULLDOWN ON
26	GPIO19	Low	R_TERM_ON_5	ACTIVACION RESISTENCIA TERMINACION 120OHM RS485 PORT 5	R120 OFF	R120 ON
27	GPIO20	Low	R_TERM_ON_1	ACTIVACION RESISTENCIA TERMINACION 120OHM RS485 PORT 1	R120 OFF	R120 ON

E. Software de escritorio

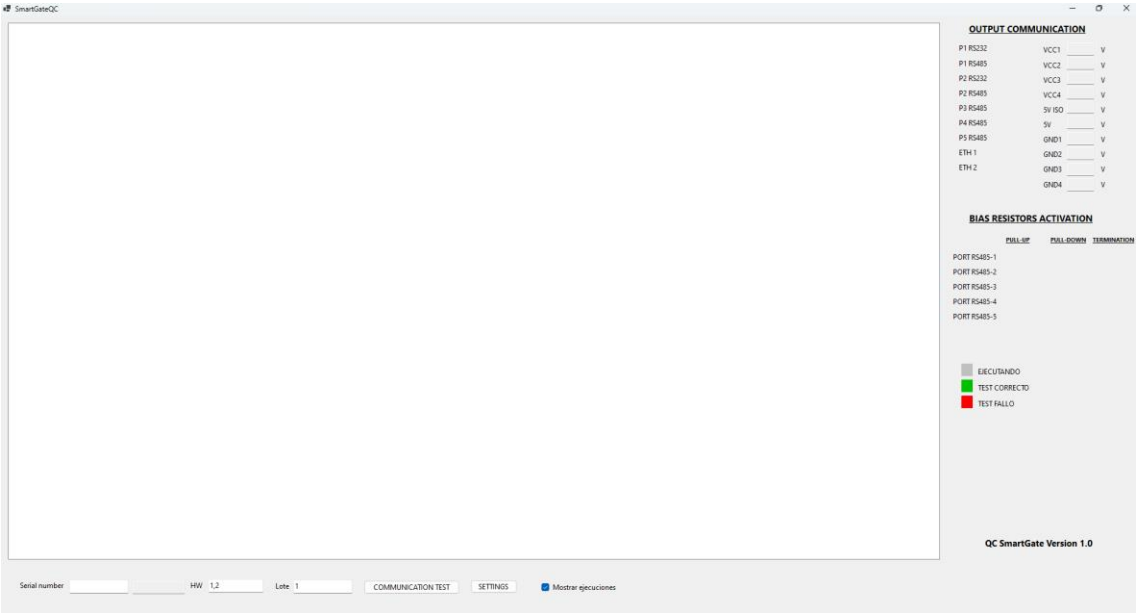


Figura 65: Pantalla principal del software de escritorio

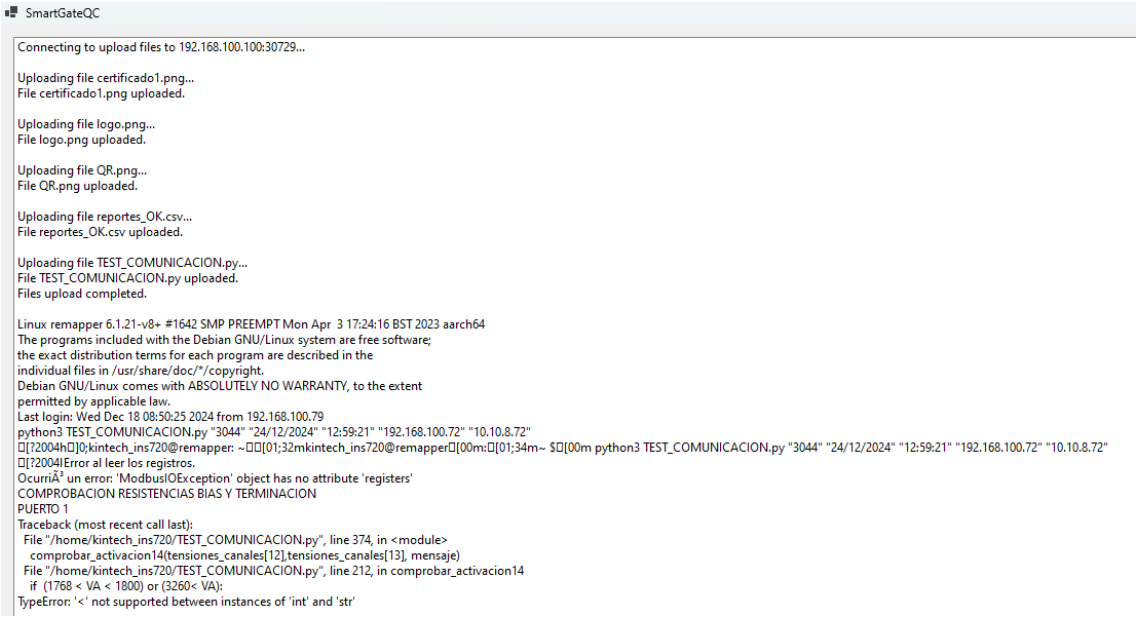


Figura 66: Terminal en ejecución

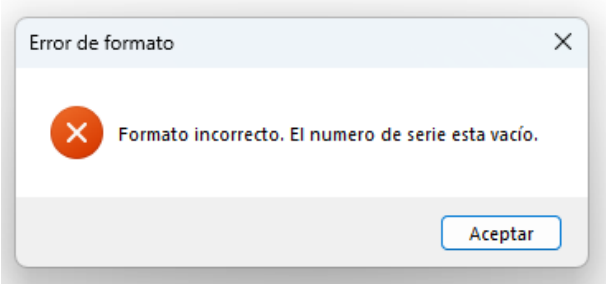


Figura 67: Aviso formato incorrecto del número de serie

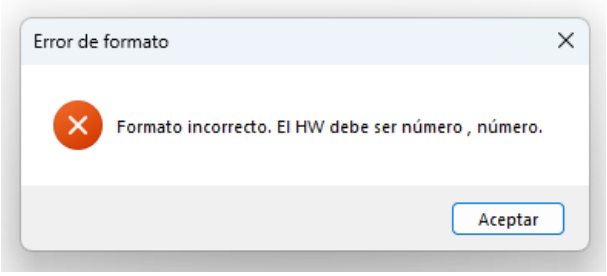


Figura 68: Aviso formato incorrecto del HW

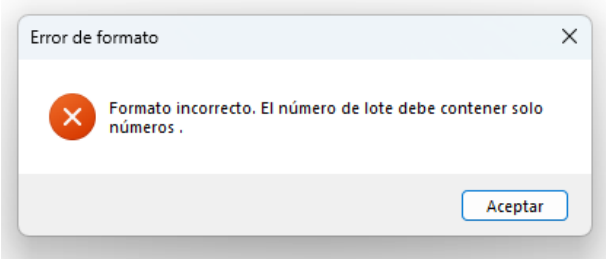


Figura 69: Aviso formato incorrecto del lote

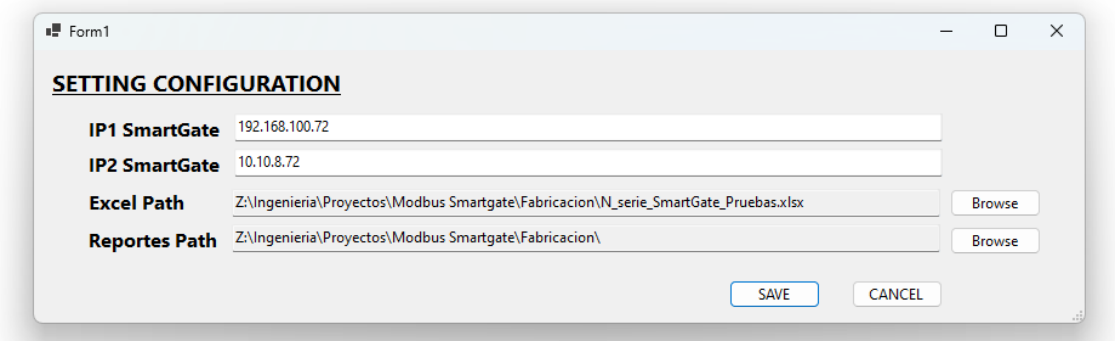


Figura 70. Ventana de Settings

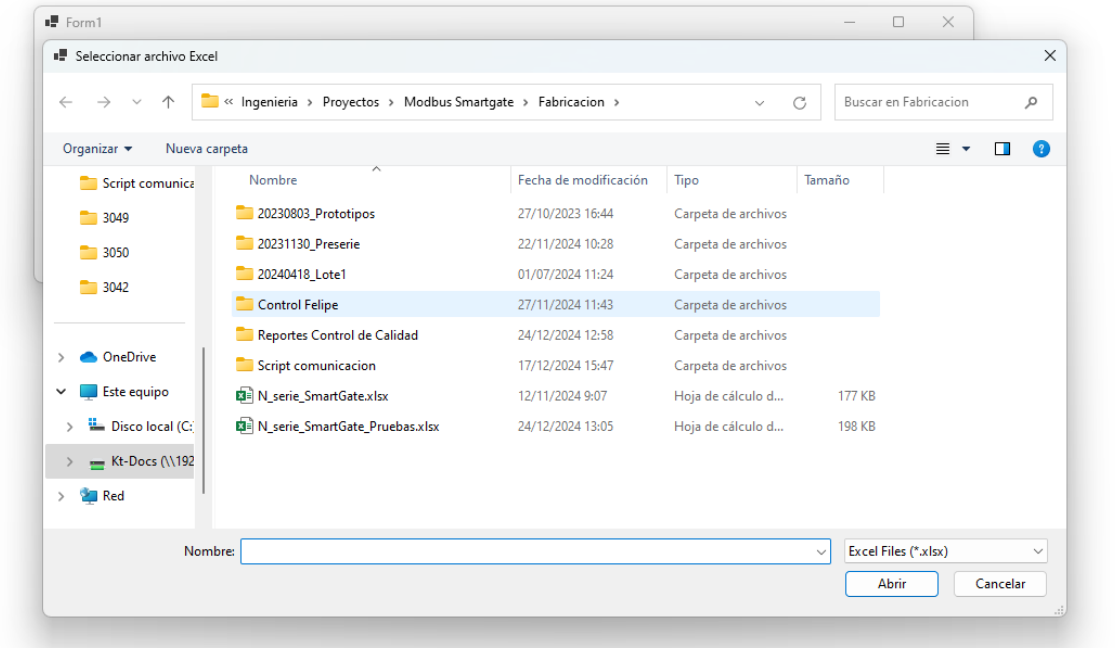


Figura 71: Ventana de directorios botón Browse

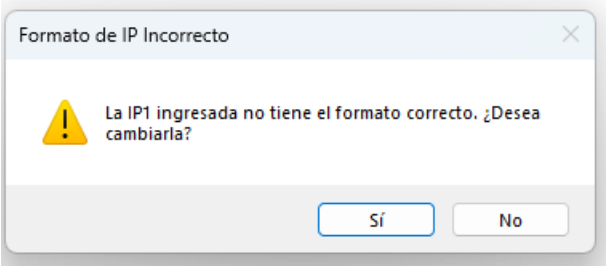


Figura 72: Aviso formato incorrecto de las IP

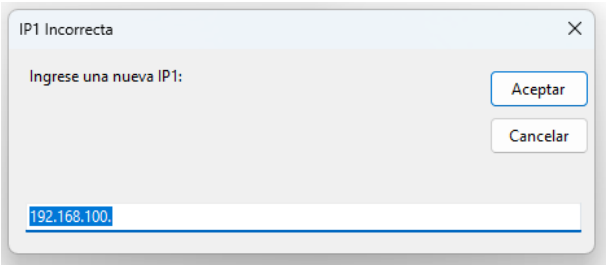


Figura 73: Ventana modificación de IP en caso de fallo

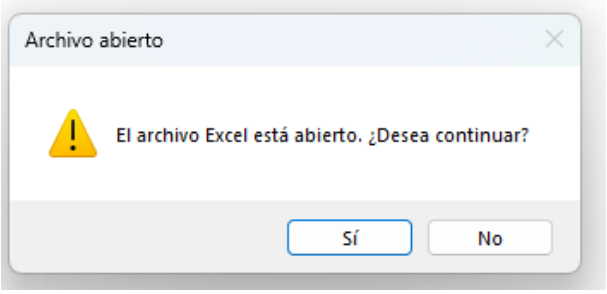


Figura 74: Aviso Excel inventario abierto

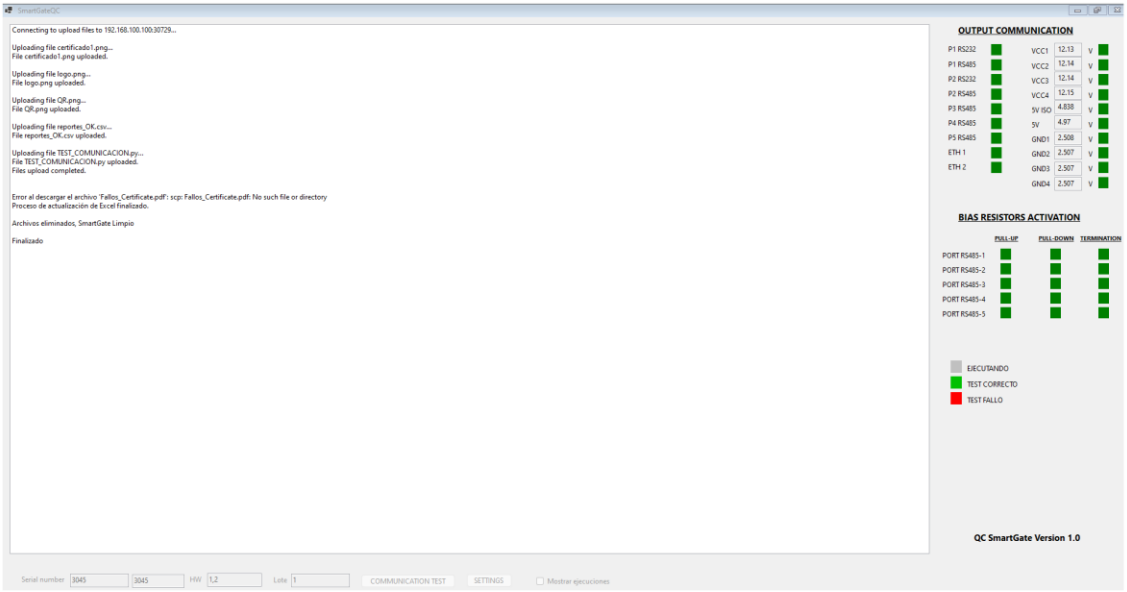


Figura 75: Pantalla principal del software ejecutado

F. Estudio de tensiones en los canales A y B de los puertos RS-485 según la combinación de resistencias.

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	False	False	False	False	False	False	False	False	False	False	False	False	False	False	False
A(V)	0	0		0	0		0	0		0	0		1,447	1,447	
B(V)	0	0		0	0		0	0		0	0		1,443	1,443	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	False	False	True	False	False	True	False	False	True	False	False	True	False	False	True
A(V)	0	0		0	0		0	0		0	0		1,449	1,449	
B(V)	0	0		0	0		0	0		0	0		1,449	1,449	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	False	True	False	False	True	False	False	True	False	False	True	False	False	True	False
A(V)	0	0		0	0		0	0		0	0		1,446	1,446	
B(V)	0	0		0	0		0	0		0	0		0	0	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	False	True	True	False	True	True	False	True	True	False	True	True	False	True	True
A(V)	0	0		0	0		0	0		0	0		0,03	0,03	
B(V)	0	0		0	0		0	0		0	0		0,03	0,03	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	True	False	False	True	False	False	True	False	False	True	False	False	True	False	False
A(V)	3,285	0		3,285	0		3,285	0		3,285	0		4,83	4,83	
B(V)	0	0		0	0		0	0		0	0		1,443	1,443	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	True	True	False	True	True	False	True	True	False	True	True	False	True	True	False
A(V)	3,28	0		3,28	0		3,28	0		3,28	0		4,83	4,83	
B(V)	0	0		0	0		0	0		0	0		0	0	

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	True	False	True	True	False	True	True	False	True	True	False	True	True	False	True
A(V)	3,27			3,27			3,27			3,27			4,8		
B(V)	3,26			3,26			3,26			3,26			4,79		

Resistor	up1	down1	term1	up2	down2	term2	up3	down3	term3	up4	down4	term4	up5	down5	term5
Estado	True	True	True	True	True	True	True	True	True	True	True	True	True	True	True
A(V)	1,773			1,771			1,771			1,776			2,637		
B(V)	1,509			1,507			1,507			1,51			2,239		

G. Estudio para el caso de fallo de P. Down y terminación

SIN TERMINACION NI DOWN

SLEEP 0s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 TTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
```

SLEEP 1s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT       1.77V/3.3V       3.277 V        OK
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
TERMINACION PUERTO 2 TTT                                          Revisar
```

SLEEP 0.3s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 TTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
```

SOLO TERMINACION

SLEEP 0s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 TTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
```

SLEEP 1s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT       1.77V/3.3V       3.274 V        OK
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
TERMINACION PUERTO 2 TTT                                          Revisar
```

SLEEP 0.3s

```
COMPROBACION RESISTENCIAS BIAS Y TERMINACION
RESISTENCIA                 ESPERADO        RECIBIDO      ESTADO
PUERTO 2 FFT
PULL-UP PUERTO 2 FFT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FFT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FFT       1.516 V        0.0 V          Revisar
PUERTO 2 FTF
PULL-UP PUERTO 2 FTF -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTF       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTF       1.516 V        0.0 V          Revisar
PUERTO 2 FTT
PULL-UP PUERTO 2 FTT -> NO OK, no se puede saber el estado de las demás
PULL-UP PUERTO 2 FTT       1.77V/3.3V       0.0 V          Revisar
PULL-DOWN PUERTO 2 FTT       1.516 V        0.0 V          Revisar
PUERTO 2 TTT
PULL-UP PUERTO 2 TTT       1.77V/3.3V       3.274 V        OK
PULL-DOWN PUERTO 2 TTT       1.516 V        0.0 V          Revisar
TERMINACION PUERTO 2 TTT                                          Revisar
```