

Actuadores inteligentes con protocolo de comunicación I²C y medio físico RS-485

Intelligent actuators with I²C communication protocol and RS-485 physical medium

Erika Pimentel O.

Universidad Distrital Francisco José de Caldas
epimentelo@correo.udistrital.edu.co

Yeferson Yate M.

Universidad Distrital Francisco José de Caldas
yyatem@correo.udistrital.edu.co

Jorge E. Porras B.

Universidad Distrital Francisco José de Caldas
jeporrasb@udistrital.edu.co

Este artículo muestra el desarrollo de un sistema Maestro/Esclavo haciendo uso del protocolo de comunicación I²C por medio físico RS-485 para el control de los actuadores que trabajan en la PPA (Planta de Procesos Análogos) con el fin de reemplazar ciertas funciones del PLC (Controlador Lógico Programable) con el que cuenta actualmente la planta. Este sistema se implementa utilizando PSoC 4200 Prototyping Kit, el sistema es inteligente por tener las características de llevar la cuenta del uso del actuador, el estado del mismo (On/Off) e informar en caso de desconexión por factores externos (desgaste o daño en las líneas de comunicación y/o hardware) o interno (comunicación fallida).

Palabras clave: Actuador inteligente, I²C, PSoC

This paper shows the development of a Master/Slave system using I²C communication protocol for RS-485 physical medium to control actuators working in the APP (Analog Process Plant) in order to replace certain functions of the PLC (Programmable Logic Controller) which currently has the plant. This system is implemented using PSoC 4200 Prototyping Kit, the system is smart to have the features to keep track of the use of the actuator, the condition (On/Off) and report in case of disconnection by external factors (wear or damage in the communication lines and/or hardware) or internal (failed communication).

Keywords: Intelligent actuator, I²C, PSoC

Tipología del artículo: Investigación

Fecha recepción del manuscrito: Octubre 10, 2014

Fecha aceptación del manuscrito: Noviembre 3, 2014

Investigación financiada por: Universidad Distrital Francisco José de Caldas.

Edición digital: <http://revistas.udistrital.edu.co/ojs/index.php/tekhne/issue/view/748>

Cómo citar: Pimentel, E., Yate, Y. y Porras, J. (2015). *Actuadores inteligentes con protocolo de comunicación I²C y medio físico RS-485*. Revista Tekhnê, 12(1), 59-66.

Introducción

En la actualidad se tienen en uso tecnologías de alto costo (PLC) y software privado como la plataforma Rockwell la cual requiere de la adquisición de licencias, para el desarrollo de sistemas de control y automatización necesarios para el crecimiento de la empresa. En este caso se propone hacer uso de tecnologías de bajo costo y software libre para el desarrollo de sistemas de control, implementando el sistema a la PPA ubicada en la Universidad Distrital Francisco José de Caldas, Facultad Tecnológica. Esta se encuentra en proceso de desarrollo y construcción, el cual ha sido afectado por un alto costo de los componentes necesarios para su finalización. Por esta razón se implementa un sistema de comunicación basado en el protocolo I²C haciendo uso del medio físico RS-485, el cual se encarga de la supervisión y control de los actuadores en la PPA.

Partiendo de lo anterior, este proyecto está dividido en cinco bloques fundamentales. El primero de ellos hace énfasis en el protocolo de comunicación I²C entre un PSoC (Maestro) con dos PSoC (Esclavo). Esto permite trabajar diferentes dispositivos con dos hilos de comunicación, los cuales serán los encargados de transmitir las señales correspondientes a las acciones de los actuadores a utilizar. Posterior a esto se implementa una comunicación bidireccional entre el PSoC (Maestro) y los PSoC (Esclavos), permitiendo no solamente transmitir las señales de control, su estado (On/Off) y la respectiva cuenta de activación de los actuadores, además de recibir las señales generadas por los PSoC (Esclavos) supervisando así la correcta conexión.

El segundo bloque se encarga de la comunicación entre la interfaz y el sistema de control por medio del RS-232. El tercer bloque preserva los datos y estado (On/Off) de los esclavos guardando esta información en la EEPROM (ROM programable y borrable eléctricamente). El cuarto bloque busca extender los dos hilos de comunicación I²C dado que este trabaja a cortas distancias se ve limitado en su implementación en sistemas industriales. Se hace uso del I²C dado que la mayoría de microcontroladores no poseen RS-485. Por medio de la conversión de I²C a RS-485 se da un mayor alcance al protocolo ya que este tiene la característica de trabajar a 1.200 m a velocidades de 100 Kbps, llegando a un máximo de 120 esclavos, lo que permite la implementación de diferentes actuadores en el sistema.

En última instancia, el quinto bloque consiste en una interfaz gráfica generada en Python a través de la cual el usuario podrá activar los actuadores mediante botones que permitirán el cambio de estado (On/Off) además de poder visualizar la cuenta de activación y avisar al usuario cuando el actuador realice 100.000 activaciones para realizar mantenimiento preventivo y correctivo al actuador.

Este documento describe la realización del proyecto, el cual busca desarrollar sistemas de control orientados a tareas específicas en plantas de procesos análogos mediante la

interfaz gráfica, haciendo uso de tecnologías de bajo costo esto permite reducir costos y gastos relacionados al control y mantenimiento de dichos sistemas, además de brindar una alternativa a la utilización del PLC en el control de procesos industriales tales como el tratamiento de agua en procesos residuales. Se pretende mostrar la aplicabilidad de software libre y tecnologías asequibles en el mercado, y por ende el impacto que pueden tener en la solución de problemas de la vida real. Por último, se mencionaran los resultados obtenidos y las conclusiones.

Metodología

Propuesta general

La PPA (Fig. 1) es una planta de procesos análogos que cuenta con tres tanques y una bomba de agua que funciona a través de un motor trifásico. El objetivo de la planta es el control de nivel y temperatura en los diferentes tanques de proceso: agua caliente, mezcla y agua fría. Funcionan a través de electroválvulas (Fig. 2) a 110 V, accionadas por medio de relevos a 5 V. El objetivo del proyecto es realizar un sistema de supervisión y control de los actuadores (en este caso las electroválvulas de la PPA), e implementar el protocolo de comunicación I²C utilizando RS-485 como medio físico.



Figura 1. Planta de procesos análogos PPA.

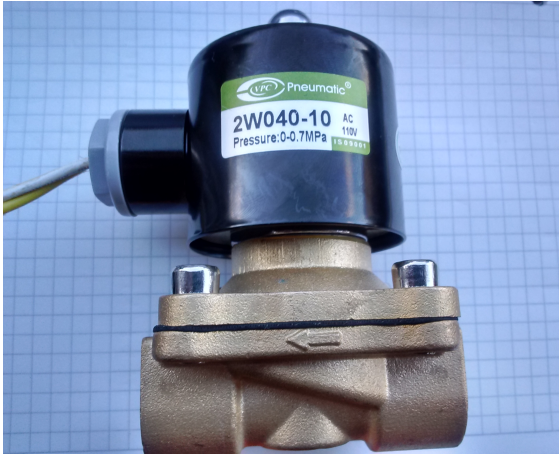


Figura 2. Electrovalvula 1/2 pulgada a 110 V.

Comunicación I²C entre PSoC (Maestro-Eslavos)

En primera instancia se desarrolló el código base en PSoC para la comunicación entre dos dispositivos un PSoC (Maestro) y un PSoC (Esclavo) a través de I²C, permitiendo transmitir señales de control de maestro a esclavo. Para evidenciar la transmisión se hizo uso de un LED (azul) como indicador para verificar visualmente que el proceso de transmisión maestro a esclavo se realizara correctamente.

Posterior a esto se utilizó el código del esclavo para hacer el proceso descrito en el párrafo anterior. Con dos esclavos se evidencia el uso del protocolo I²C al trabajar con diversos dispositivos a través de sus dos hilos de comunicación, la prueba se realizó obteniendo los resultados esperados. Por ende, se modifica el código para implementar dispositivos externos, haciendo uso de un teclado para el maestro y una LCD para el esclavo, de esta manera el teclado selecciona los datos a enviar y la LCD permite visualizar los mismos, con esto se concluye que se realiza una comunicación unidireccional del maestro al esclavo correctamente.

Comunicación I²C bidireccional. Para implementar la comunicación bidireccional se procede a transmitir y recibir datos simultáneamente entre el PSoC (Maestro) y el PSoC (Esclavo). Para esto, se utiliza un proceso similar a la comunicación unidireccional a través de indicadores tanto en el maestro como en el esclavo. Al realizar las pruebas se refleja la transmisión de maestro a esclavo pero la recepción de datos correspondientes al maestro presenta falencias. Se procede a transmitir un dato desde el esclavo y compararlo en el maestro. Al realizar esta comparación correctamente un indicador debe encenderse, después de modificar el código y probarlo se obtiene el resultado esperado (Cypress, 2015b).

Pruebas entre PSoC y RS232

Se realiza la configuración en el PSoC (Maestro) el cual se encarga de comunicar con el PC por medio del RS232.

Haciendo uso del software CoolTerm, el cual simula la comunicación de un puerto serial, con esto se pretende visualizar el dato recibido en el maestro, al realizar las pruebas correspondientes se evidencia que el dato recibido esta en formato hexadecimal, con lo cual se procede a intentar enviar un dato entero. El dato en el esclavo se modifica para ser enviado como un entero sin signo de 8 bytes, lo que nos permite trabajar con datos de envío entre 0-256. Al realizar la prueba se obtienen los datos esperados.

Se procede a enviar datos mediante el puerto serie para que el maestro manipule su actuador o transmita comandos específicos a los esclavos para que estos manipulen sus respectivas actuadores. Se implementa un proceso para llevar la cuenta de activación, pero el dato se ve limitado a una cuenta de 8 bytes, lo cual permite un máximo de 256 cuentas. En vista de esto, es necesario crear una rutina que permita extender la cuenta. Se realizó el siguiente proceso en el PSoC con el fin de llegar a un valor cercano a los 24 Bytes (tabla 1).

El proceso para la cuenta total es el siguiente (ecuación 1):

$$Cuenta = (64516 \times cuenta_2) + (254 \times cuenta_1) + contador \quad (1)$$

Donde *cuenta_1*, *cuenta_2* y *contador* son los datos leídos o generados por el maestro. Durante la realización de este proceso se presenciaron diversos errores en el reconocimiento correspondiente a cada variable, lo que generó datos erróneos en la cuenta. Después se logra sincronizar correctamente la trama de datos recibidos, permitiendo llegar a un valor máximo de 16451834 cuentas, en la Fig. 3 se puede observar los módulos correspondientes a los esclavos en el sistema. Finalmente, se realizan procesos continuos de lectura en el PSoC (Maestro) y verificación de la conexión I²C de los esclavos, en caso de que algún hilo del esclavo este desconectado el indicador (verde) se activa, estableciendo una comunicación bidireccional exitosa.



Figura 3. Módulos PSoC (esclavos).

Tabla 1

Rutina PSoC de conteo.

```

if (contador==255u && cuenta_1 == indicador_cuenta_1 && cuenta_1 != 255u){
    contador=1u;
    indicador_cuenta_1 = indicador_cuenta_1+1u;
    cuenta_1 = cuenta_1+1u;

} else if (cuenta_1 == 255u && cuenta_2 == indicador_cuenta_2 && cuenta_2 != 255u)
{
    contador=1u;
    cuenta_1 = 1u;
    indicador_cuenta_1 = 1u;
    cuenta_2= cuenta_2+1u;
    indicador_cuenta_2 = indicador_cuenta_2+1u;
} else if (cuenta_1 == 255u && cuenta_2 == 255u){
    contador=0;
    cuenta_1=0u;
    cuenta_2=0u;
    indicador_cuenta_1 = 0u;
    indicador_cuenta_2 = 0u;
}
    
```

EEPROM

La característica principal de la EEPROM es que permite guardar y preservar datos hasta 20 años en la memoria del PSoC. Utilizando el código ejemplo brindado por Cypress de la EEPROM en el PSoC, al intentar hacer uso de dicho ejemplo se observa un fallo en el mismo, debido al *bootload* con el cual se carga el programa en el microcontrolador. Al realizar las pruebas correspondientes cortando la alimentación del dispositivo, se genera un cambio de estado en el microcontrolador, obligando a cargar de nuevo el *bootload*. Para solucionar este problema se optó por hacer uso del KitProg, el cual puede programar y depurar el PSoC. Al tener la posibilidad de preservar los datos en la EEPROM como el estado actual del actuador (On/Off). En caso de falla en la fuente de alimentación, la cuenta de activación y el estado del actuador se retornaran al restablecer la alimentación en el dispositivo (Cypress, 2015a).

Medio físico RS-485

Haciendo uso del estándar RS-485 el cual consiste en un bus de transmisión multipunto diferencial, se busca brindar un mayor alcance en las señales del protocolo de comunicación I²C haciendo la correspondiente conversión al estándar RS-485. En la Fig. 4 se muestra un proceso en la adaptación de las señales SDA y SCL, tanto en el maestro como en los esclavos.

El circuito correspondiente al bloque RS-485 se puede observar en la Fig. 5. El módulo de conversión hace uso de los componentes electrónicos detallados en la tabla 2.

Tabla 2

Elementos utilizados en la elaboración del módulo RS-485.

Cantidad	Nombre
2	Max 485
4	Resistencias 2.2K
4	Resistencias 680
2	Diodos 1N4148
1	Trimmer 1K
1	Lm339

Funcionamiento. El Max 485 cuenta con una salida (RO) y una entrada (DI), una habilitación de lectura (RE) y de escritura (DE). Al estar RE y DE en 1 lógico se escribe DI, en caso contrario 0 lógico se lee RO. La alimentación del circuito es de 5 V, los terminales de transmisión A (Pin 6) y B (Pin 7) son los encargados de establecer la comunicación multipunto entre los diferentes dispositivos.

Las líneas SDA y SCL del protocolo I²C son drenador abierto por lo cual manejan una resistencia en *pull-up* estando en un estado lógico alto por defecto, para alternar la escritura y lectura del Max 485 se hace uso de un comparador LM 339 con un voltaje de referencia de 0.6 V, al estar la línea en un nivel superior al de referencia la salida del comparador está en un nivel lógico bajo por lo tanto el Max

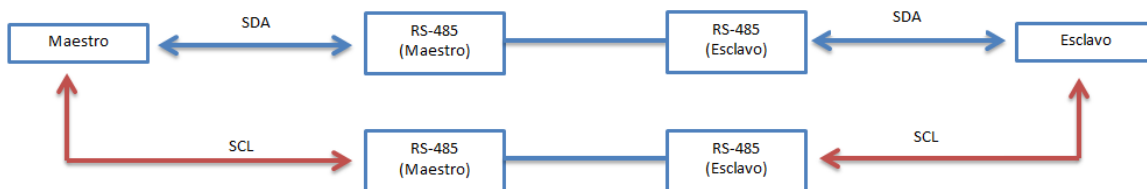


Figura 4. Esquema adaptación de SDA y SCL por medio físico RS-485.

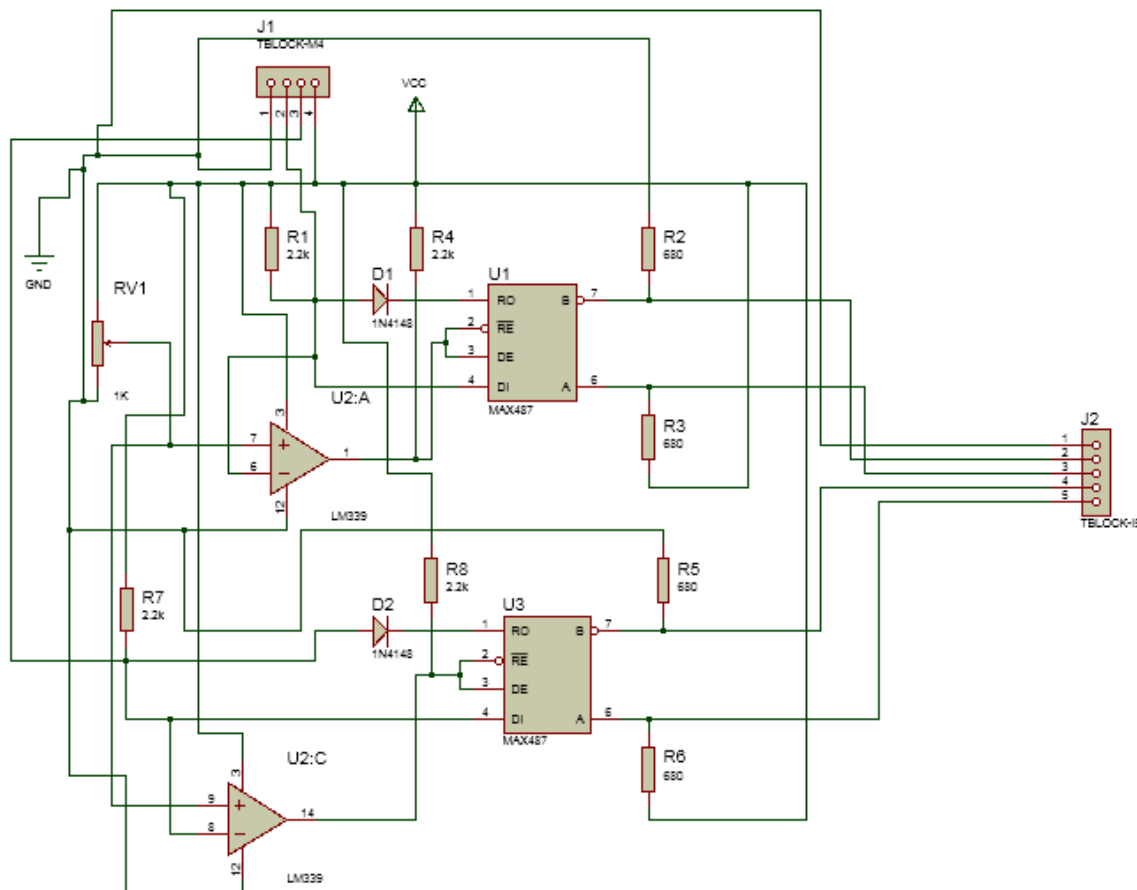


Figura 5. Esquema circuital del módulo RS-485.

485 se encuentra en estado de lectura, en caso contrario el comparador estará en un nivel lógico alto y el Max 485 se encontrará en estado de escritura.

El PSoC trabaja con la familia lógica CMOS esta detecta niveles de tensión bajos en un rango de 0 V-1.5 V y niveles de tensión altos en un rango de 3.5 V-5 V. En vista de esto, se hace uso de un diodo en el terminal RO (Pin 1), lo cual permite que la línea original (Fig. 6) baje a un nivel de 0.7 V en el caso de que reciba un 0 lógico, en caso contrario la línea original estará aislada de la señal recibida, esto permite que el PSoC detecte un cambio en la señal pero sin activar el

comparador y en consecuencia sin transmitir la línea. Cuando la señal original del protocolo aterrice la línea a un nivel menor a 0.6 V se transmitirá la línea por medio del Max 485 permitiendo la comunicación entre los dispositivos (Axelson, 1998).

Interfaz gráfica Python

Python al ser un software libre brinda la posibilidad a los usuarios de trabajar en diferentes plataformas. Dicho lenguaje se caracteriza por su sintaxis limpia en comparación a otros lenguajes de programación, además de ser manejado

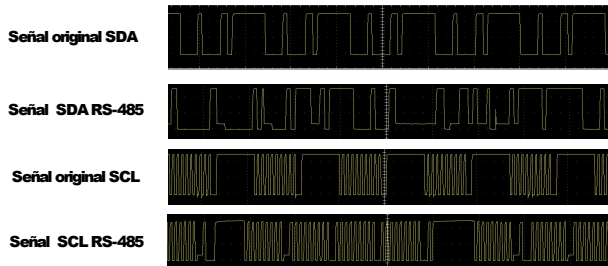


Figura 6. Señales SDA y SCL originales y por medio físico RS-485.

por un intérprete lo que ahorra tiempo y recursos. La versión que se optó a usar es la 3.4 en vista de que está en constante crecimiento. Para el desarrollo de la interfaz gráfica se utiliza Tkinter el cual es un estándar para la interfaz gráfica de usuario (GUI). En él se implementan funciones establecidas para predeterminedar el tamaño de la ventana, botones de usuario, etiquetas de texto, color de texto y fondo, soporta imágenes en formato gif y png además del uso de *messagebox* (Spence, 2014).

El primer paso es la transmisión de datos a través de PySerial la cual es una librería de Python que permite comunicarse serialmente a través de RS-232. Reemplazando el simulador CoolTerm por la interfaz en Python, al realizar las pruebas de transmisión entre el PC y el PSoC (Maestro) se presenta un error al transmitir los datos ya que estos poseen un formato distinto al trabajado en el PSoC (Maestro), por lo cual era incapaz transmitir correctamente el dato. La solución fue codificar el dato enviado por medio de la función `encode('ASCII')` (Liechti, 2015).

Se propone realizar la lectura correspondiente a los datos de cada esclavo como lo son la cuenta de activación, el estado (On/Off) y si el dispositivo está conectado, para esto se realizaron modificaciones del código ya que presentaba errores en la transmisión de datos del PSoC (Maestro) al PC. La solución fue aumentar el tamaño del dato leído en la interfaz. Al recibir los datos los cuales se visualizan en las etiquetas como una cadena de caracteres (formato *string*) y al establecer acciones específicas en función del dato recibido, se hacen notorios varios problemas debido una vez más a la incompatibilidad en el formato de recepción. Después de consultar se utiliza la función `decode()`, la cual permite pasar los datos recibidos tipo byte a cadenas de caracteres, lo anterior es resumido en un diagrama de flujo el cual se puede apreciar en la Fig. 7, dando la posibilidad de usar acciones para cambiar así los colores de las etiquetas, las cuales funcionan como indicadores además del uso de *messagebox* para avisar al usuario los problemas en la conexión del dispositivo (al no estar conectado) y una alerta que informa cuando se debe hacer el mantenimiento al actuador, se incorpora un botón para reiniciar la cuenta de mismo, estas características pueden observarse en la Fig. 8.

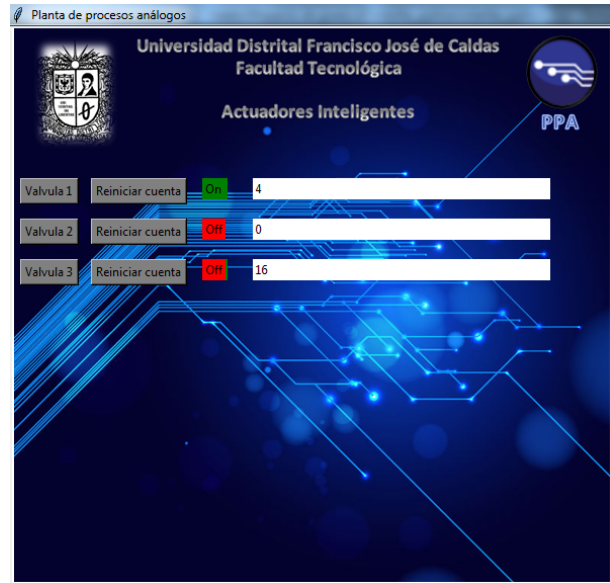


Figura 8. Interfaz gráfica Python.

Diagrama de bloques

El diagrama de la Fig. 9 describe el funcionamiento del sistema de control exponiendo sus cinco bloques fundamentales.

- Comunicación I²C: Protocolo de comunicación encargado de crear una interacción entre el PSoC (Maestro) y los PSoC (Esclavos). El maestro es el encargado de comunicar con el PC, realizando él envío de los comandos y la supervisión correspondiente a los datos recibidos por los esclavos, el sistema se encarga de llevar la cuenta de activación del actuador como su estado (On/Off).
- RS-232: Encargado de la comunicación serial entre el PSoC (Maestro) y el PC.
- EEPROM: Guarda los datos y estado (On/Off) de los actuadores en caso de pérdida en la alimentación del sistema.
- RS-485: Medio físico encargado de comunicar entre los módulos maestro/esclavo.
- Interfaz Python (PC): interfaz gráfica donde el usuario interactúa con el sistema.

Resultados

Al hacer la conversión del protocolo I²C al estándar RS-485, se logra un mayor alcance a las líneas del protocolo, con pruebas realizadas hasta 60 m con un correcto funcionamiento.

RS-485 permite un máximo de 32 dispositivos en sus líneas de comunicación, al hacer la conversión, se puede llegar a un máximo de 120 dispositivos.

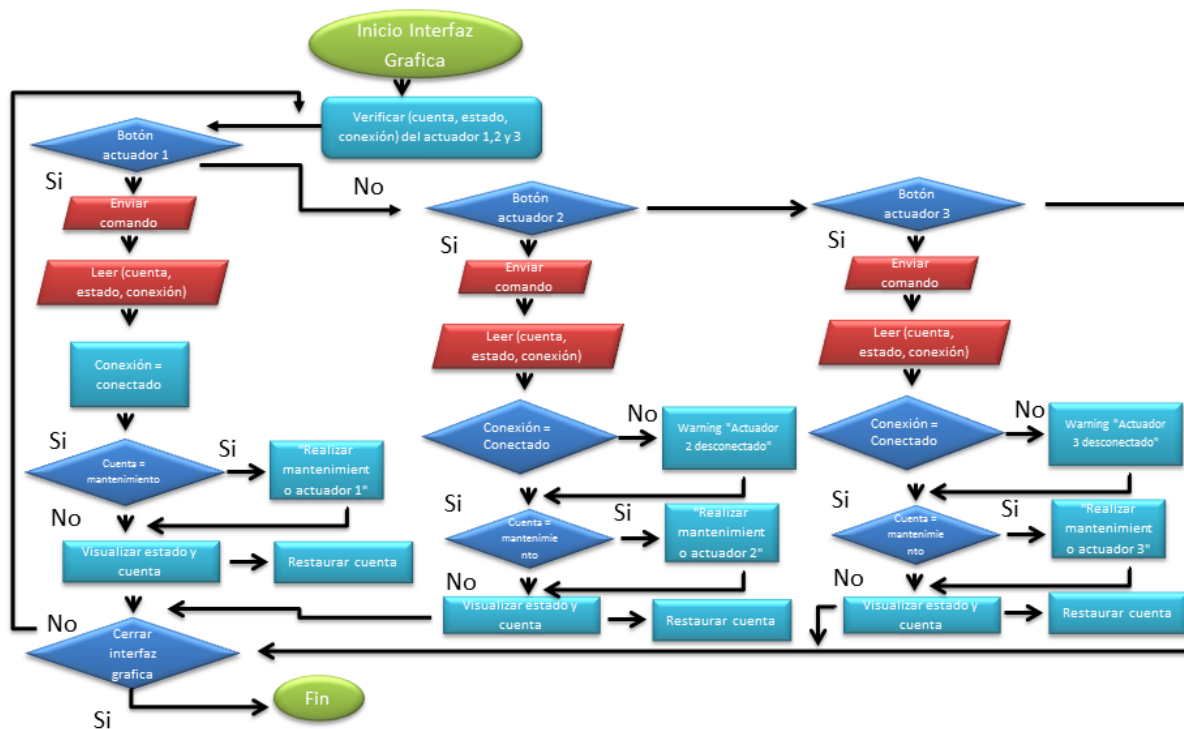


Figura 7. Diagrama de flujo interfaz gráfica Python utilizando el maestro y dos esclavos como actuadores.

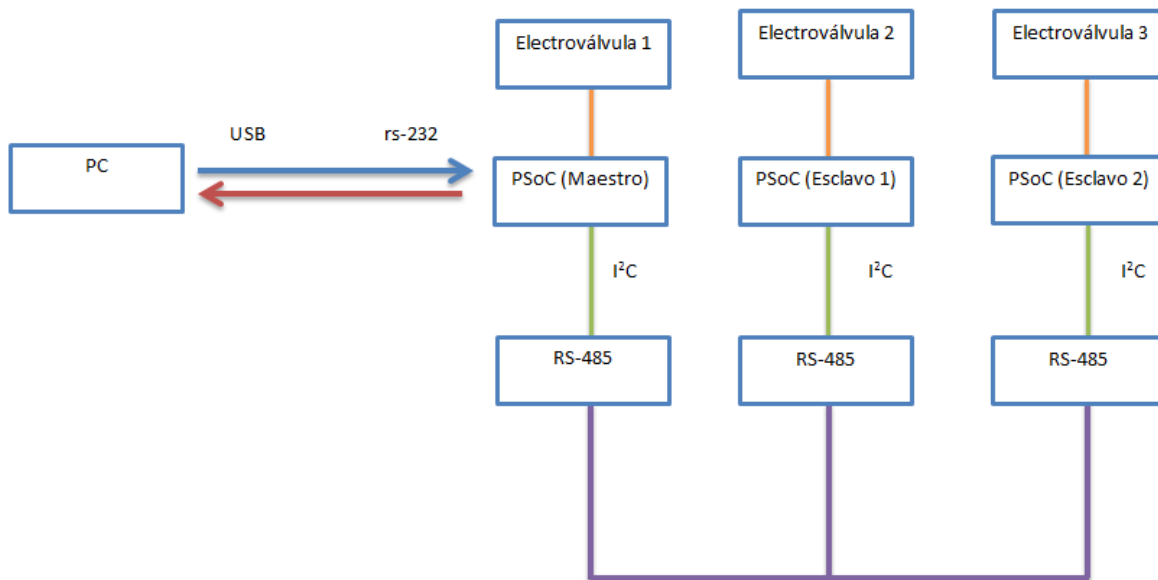


Figura 9. Bloques proyecto actuadores inteligentes con protocolo de comunicación I²C y medio físico RS-485.

Conclusiones

Al transmitir las líneas pertinentes al protocolo (SDA y SCL) se debe realizar un proceso paralelo en ambas líneas de comunicación para asegurar la sincronización debido a

los retardos producidos por la conversión de I²C a RS-485, garantizando el correcto funcionamiento del protocolo.

El uso de un bootloader (carga de arranque) en el PSoC 4 para su programación, interfiere directamente con el uso de la EEPROM, por lo tanto se hizo uso del KitProg para la programación del PSoC 5 por SWD (hilo

serial de depuración) permitiendo trabajar la EEPROM sin inconvenientes.

Con el fin de brindar un mayor alcance al protocolo I²C se optó por manejar una velocidad de 100 Kbps para transmitir por medio de RS-485, ya que a velocidades mayores presenta un menor alcance en la transmisión y recepción de datos.

A través del uso de la interfaz gráfica desarrollada en Python se puede realizar el control del sistema sin hacer uso de un HMI, permitiendo visualizar las cuentas de activación y monitorear el estado de los actuadores.

Referencias

- Axelson, J. (1998). *Serial port complete. programing and circuits for rs-232 and rs-485 links and networks.* Lakeview Research.
- Cypress. (2015a). *Eeprom.* (PSoC Creator Component Datasheet)
- Cypress. (2015b). *Psoc 4 serial communication block (scb).* (PSoC Creator Component Datasheet)
- Liechti, C. (2015). *Python serial port extension.* (pySerial's documentation)
- Spence, E. (2014). *An introduction to gui programming with tkinter.* (SciNet HPC Consortium)

