|  |
| --- |
| Título |
| Monitoreo y control de nivel de un tanque |
| Resumen |
| Implementación de protocolos de comunicación MQTT, UART e I2C con el fin de controlar el nivel de un tanque mediante la variación de velocidad de una bomba. Se hace uso los sistemas embebidos Wemos D1 R1, Arduino UNO y BeagleBone Black; así como también un computador que desplegará la interfaz gráfica donde se coordina todo el sistema. |

|  |
| --- |
| Autores |
| Juan Diego Peña Narvaez |
| José Jorge Rodríguez Salgado |
| 11/01/2022 |

|  |  |  |  |
| --- | --- | --- | --- |
| Rev. | Descripción | Elaborado | Aprobado |
| 01 | Primera edición |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Índice

[1. Introducción 3](#_Toc90816473)

[1.1. Objeto 3](#_Toc90816474)

[1.2. Alcance 3](#_Toc90816475)

[1.3. Normas y referencias 3](#_Toc90816476)

[1.4. Definiciones y abreviaturas 3](#_Toc90816477)

[2. Descripción general (concepto) 5](#_Toc90816478)

[3. Requisitos Error! Bookmark not defined.](#_Toc90816479)

[3.1. Requisitos de funcionamiento normal **Error! Bookmark not defined.**](#_Toc90816480)

[3.2. Requisitos de parada normal **Error! Bookmark not defined.**](#_Toc90816481)

[3.3. Requisitos de parada de emergencia **Error! Bookmark not defined.**](#_Toc90816482)

[3.4. Requisitos de control (mando) **Error! Bookmark not defined.**](#_Toc90816483)

[4. Arquitectura 5](#_Toc90816484)

[5. Diseño e implementación 6](#_Toc90816485)

[5.1. Cinta de entrada **Error! Bookmark not defined.**](#_Toc90816486)

[5.2. Báscula (pesado) **Error! Bookmark not defined.**](#_Toc90816487)

[5.3. Clasificador **Error! Bookmark not defined.**](#_Toc90816488)

[5.4. Cintas de salida **Error! Bookmark not defined.**](#_Toc90816489)

[5.5. Mando (control de la planta) **Error! Bookmark not defined.**](#_Toc90816490)

[6. Resultados 14](#_Toc90816491)

[7. Conclusiones Error! Bookmark not defined.](#_Toc90816492)

[7.1. Trabajo futuro 14](#_Toc90816493)

# Introducción

## Objeto

El objetivo del proyecto es la implementación del control y monitoreo de nivel de un tanque a través de un control PID. Las variables más relevantes serán visualizadas mediante una interfaz gráfica y se usaran sistemas embebidos, que por medio de los protocolos de comunicación UART, I2C y MQTT se encargaran de obtener y publicar información de la planta en la interfaz, así como implementar el algoritmo de control. El correcto funcionamiento se evaluará en una planta física.

## Alcance

El nivel del tanque deberá funcionar acorde a los parámetros establecidos del control PID; en cuanto a las comunicaciones, la información deberá actualizarse de manera eficaz y no se deberá perder información. La interfaz deberá ejecutar correctamente todas sus funcionalidades como apagar o encender la planta, actualizar los parámetros del PID y monitorear los datos. Para este proyecto no se contempla ninguna capa de seguridad o encriptamiento de la información, así como tampoco su almacenamiento para un posterior análisis.

## Normas y referencias

|  |  |
| --- | --- |
| [1] | «Arduino,» [En línea]. Available: https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf. |
| [2] | R. Keim, «All about circuits,» [En línea]. Available: https://www.allaboutcircuits.com/technical-articles/low-pass-filter-a-pwm-signal-into-an-analog-voltage/. [Último acceso: 065 01 2022]. |
| [3] | B. Beauregard, «https://github.com/br3ttb/Arduino-PID-Library,» [En línea]. Available: https://github.com/br3ttb/Arduino-PID-Library. [Último acceso: 07 01 2022]. |

## Definiciones y abreviaturas

**UART** Universal asynchronous receiver-transmitter

**MQTT** MQ Telemetry Transport

**I2C**  Inter-Integrated Circuit

**PCB** Printed Circuit Board

**PID** Controlador proporcional, integral y derivativo

**SPI** Serial Peripherical Interface

**PC** Computador personal

**AU** Arduino UNO

**WD** Wemos D1 R1

**BB** BeagleBone Black

**PV** Variable de proceso

**CP** Variable de control

**KP** Constante proporcional

**TI** Constante integral

**TD** Constante derivativa

**F** Frecuencia [Hz]

**R** Resistividad [Ω]

**C** Capacitancia [f]

# Descripción general (concepto)

La planta cuenta con un tanque de abastecimiento además del tanque utilizado para controlar el nivel. Este último tanque, cuenta con un sensor de ultrasonidos que genera una señal analógica de 0 a 10 V según sea el nivel del tanque. La planta esta interconectada mediante tuberías y una bomba centrifuga, encargada de mantener el nivel deseado variando su velocidad. Esta velocidad cambia según una señal analógica, también en el rango de 0 a 10 V.

La planta cuenta también con una fuente de poder que servirá como alimentación del sistema embebido encargado de su control.

A picture containing wall, indoor

Description automatically generated

Figura . Planta laboratorio.

# Arquitectura

En la figura 2, se muestra la descripción de la arquitectura implementada. Se van a controlar dos plantas, utilizando una placa AU conectada a una placa WD. Las placas AU se encargarán de ejecutar el control PID y a su vez enviar las señales PV y CP a la placa WD. Finalmente, la placa WD se comunica con un bróker local instalado en la placa BB que sirve como servidor y puente entre la planta y la GUI.

A picture containing diagram

Description automatically generated

**I2C**

**UART**

**MQTT**

Figura . Arquitectura implementada.

# Diseño e implementación

En esta sección se explicará de manera detallada los factores tenidos en cuenta durante la etapa de diseño e implementación, también se mencionan asunciones y cálculos realizados.

## PCB

El uso de sistemas embebidos como el AU, WD y BB hace necesario que al interactuar con plantas industriales se dispongan circuitos de acondicionamiento, esto con el fin de regular el voltaje, filtrar y convertir de señales.

Aunque dentro del laboratorio donde se realizan las pruebas, existen ya estos tipos de circuitos, se realiza el diseño de uno que permita la comunicación entre un AU y la planta.

Un posible circuito entre la planta y una placa WD o BB llevarían los mismos componentes, con un cambio en los valores de resistencias y capacitores, así como una variación en la geometría. Sin embargo, el diseño aquí planteado se puede extrapolar para cualquier sistema embebido.

El diseño del circuito se llevó a cabo mediante el uso de EAGLE con una licencia estudiantil.

A circuit board with wires

Description automatically generated with low confidence

Figura 3. PCBs disponibles en el laboratorio.

### Fuente de poder

Según el datasheet de la placa AU, esta puede ser alimentada con voltajes de 6 a 12 V con una corriente máxima de 410 mA [1]. Ya que el tablero de la planta cuenta con una salida de voltaje de 24V se deberá acondicionar el voltaje a 12 V. Para esto, se hizo uso de un bloque predefinido en EAGLE que disminuye el voltaje de 24 a 12 V y cuenta también con diferentes protecciones.

Diagram, schematic

Description automatically generated

Figura 4. Fuente de poder Arduino uno

### Acondicionamiento de entradas

La planta a controlar dispone de un sensor de ultrasonido que funciona en los rangos de voltaje de 0 a 10 V. Por otro lado, la entrada analógica de la placa AU maneja niveles de 0 a 5 V. Además de regular el voltaje también se propone un filtro que atenúe la señal del ruido. Ya que este viene mayoritariamente de la red eléctrica y en España la frecuencia del sistema eléctrico es 60 Hz, se estableció este valor como frecuencia de corte y mediante un amplificador operacional se filtra y atenúa la señal del sensor.

Diagram, schematic

Description automatically generated

Figura 5. Acondicionamiento señal de sensor.

Los valores de R3 y C4 se obtuvieron mediante la formula de frecuencia de corte en un circuito RC, como se muestra continuación.

Asumiendo un valor de 1000 Ω y un valor de frecuencia de 60 Hz establecido anteriormente, el valor de capacitancia seria entonces de 2.65 µf. Sin embargo, el valor comercial mas cercano es 2.2 por lo que la frecuencia de corte seria 72 Hz.

En cuanto a los valores de R5 y R4 se siguió la formula de un amplificador operacional no inversor en la configuración sallen-key, en donde la ganancia se puede calcular de la siguiente forma:

### Acondicionamiento de salidas

La velocidad de la bomba será controlada por una señal digital PWM de amplitud 0 a 5V generada por la placa AU. Esta bomba funciona con niveles de tensión de 0 a 10V. por lo que se deberá convertir la señal digital a una analógica y posteriormente aumentar la tensión a los niveles que usa la bomba.

Para la conversión a analógica se hizo uso también de un filtro paso bajo, según recomendaciones de [2], la frecuencia de corte del circuito será alrededor de 200 veces menor de la frecuencia PWM que se quiere convertir, ya que Arduino genera señales a 490 Hz y usando la fórmula de frecuencia de corte anterior, se obtuvieron los siguientes valores.

Diagram, schematic

Description automatically generated

Figura 6. Circuito bomba.

Finalmente, el circuito completo se muestra a continuación:

A picture containing text, electronics, circuit

Description automatically generated

Figura 7. PCB final

## Controlador

Las dos placas AU, se encargan de mantener el nivel del agua según el *setpoint* fijado. Para esto se implementa un control PID, basándose en la librería *PID v1* [3]. Esta librería, usando la señal analógica del sensor, determina el ciclo de trabajo de la señal PWM que se enviara a la bomba. Los parámetros del controlador están preestablecidos. Sin embargo, se cuenta con la opción de modificarlos desde la GUI. Por lo que dentro del algoritmo no se contempla ningún proceso de optimización que genere dichos parámetros automáticamente.

## Programación del sistema

La coordinación y el manejo de la información, junto con la implementación del control se realizó mediante la programación de algoritmos en las dos placas utilizadas. En términos generales, se evalúa siempre el estado de la planta (encendida o apagada). Cuando esta en funcionamiento, hay un flujo constante de información, la placa AU esta ejecutando el control y enviando las variables CD y PV a la placa WD, la que a su vez le envia al AU que parámetros debe llevar el control PID y envía la información recolectada por el AU al bróker.

Ya que la placa WD siempre está enviando los parámetros al AU, estos se actualizan automáticamente cada vez que se indica en la GUI.

En las figuras 8 y 9 se explica de manera mas detallada el funcionamiento de los algoritmos implementados. Además, se anexa el código dentro de la entrega.

Figura 8. Codigo arduino.

No

Encendido*?*

Parámetros PID

Comando ON/OFF

Ejecuta PID

Lee y asigna parámetros

Escribe datos comunicación serial

Si

{CP = 0}

No

Establece conexión WIFI

Conectado*?*

Información GUI:

On/off, sp, kp… etc

Si

No

Conexión con el bróker

Lee información de la GUI

Conectado*?*

Lee y envia datos comunicación serial

Publica información

Parámetros nuevos?

Actualiza parámetros

No

Si

No

Encendido*?*

Si

Si

Figura . Algoritmo placa WD.

Figura . Algoritmo placa AU.

### UART

En la primera planta, la comunicación entre las placas AU y WD se realiza mediante el protocolo de comunicación serial UART. Dentro del entorno de programación de Arduino existen funciones que permiten el envío y lectura de datos a través de este protocolo.

Para enviar los mensajes se utilizó la función “*Serial.println”* ypara la recepción se utilizó la función “*Serial.parseFloat”* y “*Serial.parseInt”*. Todas las variables se enviaron una a una y se recibieron de la misma manera.

Además, para garantizar que la comunicación sea correcta, la placa que este recibiendo la información, envía el carácter “k” al finalizar la lectura, lo que indica el inicio de la escritura para la otra placa.

### I2C

En cuanto a la segunda planta, su configuración hardware es idéntica a la primera, con excepción del protocolo que fue remplazado por I2C. En esta configuración, la placa AU actúa como esclavo y la placa WD como maestro. En lugar de enviar las variables una a una como en el protocolo UART, se decidió por empaquetar la información en una cadena de texto, separando las variables con “;”. Por lo que cuando se establecía la comunicación, las placas recibían una cadena de texto que luego se desglosaba y asignaba a cada variable correspondiente.

### MQTT

El protocolo MQTT funciona con mediante agentes “publicadores” y “suscriptores” de información que es alojada en un bróker bajo una dirección denominada “topic”. La comunicación entre el bróker y la placa WD se basa en el flujograma de la figura 9. Cuando se ordena la puesta en marcha de la planta, la placa WD actúa como constante publicador de la información a graficar. Cuando se solicita un cambio en los parámetros, la placa WD los actualiza, publicándolos solo una vez.

## Interfaz grafica

La planta podrá ser manipulada por un usuario mediante una interfaz gráfica. Esta interfaz fue previamente desarrollada específicamente para la planta por lo que en este proyecto solo se comunica con ella mediante el formato explicado en el apartado MQTT.

Chart, line chart, scatter chart

Description automatically generated

Figura . GUI utilizada

# Resultados y Conclusiones

Luego de numerosas pruebas en la planta real, se logro implementar con éxito el control de nivel y a su vez coordinar el flujo de información mediante los protocolos de comunicación MQTT, I2C y UART.

Al implementar el control PID, aunque cumple su propósito de mantener el nivel en un punto determinado, la velocidad de la bomba puede detenerse por completo en ciertos momentos para luego ponerse en marcha a su mayor velocidad. Este comportamiento, aunque efectivo, no es optimo para el control de la planta.

Finalmente, se puede concluir que el uso de sistemas embebidos es una herramienta apropiada para el prototipado rápido de proyectos debido a su fácil implementación, bajo costo y gran extensión de librerías en diversas áreas de conocimiento.

## Trabajo futuro

Ya que el objetivo del presente proyecto era explorar las diferentes opciones disponibles en cuanto a sistemas embebidos, y cómo lograr una comunicación efectiva entre dichos sistemas. Es interesante implementar diferentes configuraciones para el control de más plantas, aumentando su complejidad con diferentes protocolos y sistemas embebidos.

El flujo de información entre las placas se encuentra vulnerable a hackeos, sobre todo el protocolo MQTT, la implementación de una capa de seguridad sobre este protocolo beneficiaría al proyecto.

Las comunicaciones seriales planteadas en este proyecto se realizan entre placas que manejan diferentes niveles de voltaje, 3.3 V para la placa WD y 5V para la placa AU. Aunque durante las pruebas no se presentaron inconvenientes, es prudente realizar un análisis más exhaustivo en este ámbito.