

Proyecto Sistema SCADA.

Franco, Chiabo; Lautaro, Kuhn; Mateo, Palomeque

*UNRaf, Universidad Nacional de Rafaela, Santa Fe, Argentina.
Ingeniería en Computación 1 (IC)*

Autor Corresp.: fchiabo2017@gmail.com, 1221kuhn@gmail.com, mateopalo3@gmail.com

Abstract:

The project presented corresponds to the design and implementation of a SCADA (Supervision, Control and Data Acquisition) system applied to an automated juice production plant. This system combines hardware and software to optimize industrial processes, using components such as Arduino Mega 2560, ultrasonic sensors, flowmeters, pumps, and Bluetooth modules for wireless communication with a mobile application.

The system automates critical functions, including filling tanks from a tanker truck, monitoring liquid levels, mixing ingredients at adjustable ratios, and emptying the final product. The mobile application allows you to control the system, establishing parameters such as quantities to mix and operating times, in addition to receiving real-time feedback on the status of the process.

The development was complemented with robust programming in Arduino IDE and the creation of a mobile application in MIT App Inventor, facilitating remote supervision. The modularity and adaptability of the design ensure possibilities for improvement, such as the integration of new sensors and functionalities.

Keywords: SCADA, Automation, Arduino MEGA 2560, Process control, Bluetooth communication, Remote supervision.

Resumen:

El proyecto presentado corresponde al diseño e implementación de un sistema SCADA (Supervisión, Control y Adquisición de Datos) aplicado a una planta automatizada de producción de jugos. Este sistema combina hardware y software para optimizar procesos industriales, utilizando componentes como Arduino Mega 2560, sensores ultrasónicos, caudalímetros, bombas, y módulos Bluetooth para comunicación inalámbrica con una aplicación móvil.

El sistema automatiza funciones críticas, incluyendo el llenado de depósitos desde un camión cisterna, la supervisión de niveles de líquido, la mezcla de ingredientes en proporciones ajustables y el vaciado del producto final. La aplicación móvil permite controlar el sistema, estableciendo parámetros como cantidades a mezclar y tiempos de operación, además de recibir retroalimentación en tiempo real sobre el estado del proceso.

El desarrollo se complementó con una programación robusta en Arduino IDE y la creación de una aplicación móvil en MIT App Inventor, facilitando la supervisión remota. La modularidad y

adaptabilidad del diseño aseguran posibilidades de mejora, como la integración de nuevos sensores y funcionalidades.

Palabras claves: SCADA, Automatización, Arduino MEGA 2560, Control de procesos, Comunicación Bluetooth, Supervisión remota.

1. Introducción

En la materia Ingeniería en Computación 1, se planteó la realización de un proyecto integrador que aplicara los conocimientos adquiridos durante el cursado. Este proyecto debía incorporar un dispositivo digital conectado a componentes electrónicos, comunicándose mediante un programa desarrollado en Arduino. Además, se requería que el sistema recibiera órdenes y valores desde un dispositivo móvil, mediante conexión Bluetooth o WiFi. La finalidad del trabajo era poner en práctica la teoría aprendida y los conceptos de asignaturas como Física, Química y Análisis Matemático.

Nuestro grupo decidió trabajar con fluidos, implementando un sistema SCADA (Supervisión, Control y Adquisición de Datos), una tecnología fundamental en la industria 4.0, que optimiza la supervisión y el control de procesos a distancia. Nos ubicamos en un contexto hipotético en el que nuestra "empresa" recibe el encargo de diseñar y automatizar una planta de producción de jugos.

La maqueta diseñada para representar la planta incluye dos depósitos de almacenamiento para

diferentes líquidos, un bombo de mezcla, donde convergen los fluidos provenientes de los depósitos, en proporciones ajustables según la receta, un depósito de entrada, que simula un camión transportador y un depósito de salida, que representa el contenedor final, listo para la línea de ensamblaje o despacho.

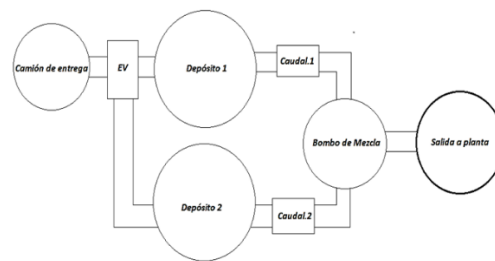


Figura 1: Boceto inicial del sistema.

Para automatizar la planta, se incorporaron diferentes sensores y actuadores, tales como caudalímetros, para medir el flujo de los líquidos, sensores ultrasónicos, para registrar el nivel de llenado de los depósitos y bombas de agua, un motor de corriente continua, relés y electroválvulas, que permiten realizar las acciones requeridas.

El procesamiento de datos y la comunicación a distancia se logran mediante un Arduino Mega 2560 y un módulo Bluetooth, integrados con un programa en Arduino y una aplicación móvil diseñada para maximizar las funcionalidades del sistema.

Todos estos componentes conforman el hardware del sistema, complementado por el desarrollo del software necesario. Para facilitar el montaje y la experimentación, se utilizaron

experimentadores y cables unipolares y multipolares, evitando la fabricación de placas electrónicas. Esto permitió construir una maqueta funcional para visualizar y analizar los procesos implementados.

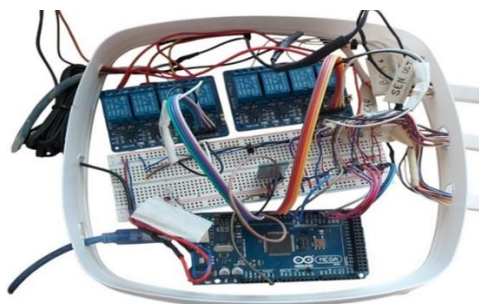


Figura 2: Centro de control internamente.



Figura 3: Sistema SCADA completo.

2. Estado del Arte

El desarrollo de sistemas SCADA (Supervisión, Control y Adquisición de Datos) ha sido ampliamente investigado e implementado en diversas áreas de la industria, debido a su capacidad para optimizar procesos, monitorear variables en tiempo real y controlar remotamente dispositivos. Ejemplos destacados incluyen la automatización en laboratorios de biotecnología, como el caso presentado en [1], donde se diseñó

un SCADA para un laboratorio de seguridad biológica, permitiendo un monitoreo riguroso de las condiciones de operación y asegurando la seguridad del entorno. Otro trabajo relevante es el análisis de los sistemas SCADA en la automatización industrial [2], que describe cómo estos sistemas han transformado la gestión de infraestructuras críticas mediante la integración de tecnologías robustas y confiables.

A diferencia de los sistemas mencionados, nuestro proyecto aborda la automatización en una planta de producción de jugos, utilizando hardware de bajo costo y accesible, como el Arduino Mega 2560, en combinación con sensores ultrasónicos, caudalímetros y comunicación Bluetooth. Este enfoque se diferencia por priorizar la accesibilidad económica y la modularidad, lo que lo hace ideal para entornos educativos o pequeñas industrias que buscan adoptar tecnologías de la Industria 4.0. Además, la integración con una aplicación móvil diseñada en MIT App Inventor permite una supervisión remota práctica, algo menos común en proyectos similares.

En comparación con implementaciones industriales más avanzadas, nuestro sistema se centra en ofrecer una solución funcional y didáctica que pueda adaptarse a diversas necesidades, manteniendo un diseño flexible y de fácil escalabilidad.

3. Marco Teórico

Para empezar, nos gustaría explicar o detallar algunos conceptos y características de los componentes, dispositivos e incluso metodologías que se utilizaron.

Sistema SCADA:

SCADA (Supervisory Control and Data Acquisition) es un sistema de control y supervisión utilizado en la industria para monitorear, recopilar y procesar datos en tiempo real, así como para controlar remotamente equipos y dispositivos de manera automática. Estos sistemas son fundamentales en entornos donde se requiere un alto grado de automatización y control eficiente de procesos. Permite a los operadores tener una visión completa y detallada de lo que sucede en la planta o sistema que están supervisando, además de la capacidad de intervenir cuando sea necesario para ajustar parámetros o responder a alarmas. Se utilizan principalmente para mejorar la eficiencia, seguridad y control de procesos industriales y de infraestructuras críticas.

Este tipo de sistemas utiliza protocolos de comunicación, formatos de transmisión y dispositivos especializados, diseñados para ser inmunes al ruido del ambiente industrial, robustos mecánicamente y simples en cuanto a software, con el fin de simplificar y garantizar la comunicación. Estas características especializadas llevan a que los dispositivos utilizados tengan un alto coste, pero a cambio garantizan fiabilidad y desempeño en entornos adversos.

Arduino MEGA 2560:

El microcontrolador utilizado es un Arduino MEGA 2560, este dispositivo es

una placa de desarrollo basada en el microcontrolador ATmega2560, diseñada para aplicaciones más complejas que requieren más pines de entrada/salida (I/O), memoria y capacidad de procesamiento que otras placas de la familia Arduino, como el Arduino Uno que fue la principal sugerencia para trabajar al inicio de este curso. Sus principales características son su microcontrolador ATmega2560, con una arquitectura de 8 bits, un voltaje de operación de 5V, aunque algunos pines soportan 3.3V para compatibilidad con sensores y tiene un voltaje de entrada recomendado de 7-12V, a través de un conector de alimentación externa o mediante un puerto USB. En el apartado de memoria, cuenta con una memoria Flash de 256 KB para almacenar el programa, de los cuales 8 KB son utilizados por el bootloader, una SRAM de 8 KB para el almacenamiento de variables temporales durante la ejecución del programa y una EEPROM de 4 KB que permite guardar datos de manera persistente. Esta placa tiene diversos pines de Entrada/Salida (I/O), 54 pines digitales, de los cuales 15 pueden ser utilizados como salidas PWM (Modulación por Ancho de Pulso) y 16 pines analógicos, para lectura de señales analógicas mediante un convertidor A/D de 10 bits de resolución. Posee 4 puertos UART que nos permiten la comunicación serial (RX/TX) con otros dispositivos. Tiene una frecuencia de reloj de 16 MHz, lo que permite un buen equilibrio entre velocidad y consumo de energía, incluye un puerto USB tipo B para programación y comunicación con una computadora, un conector ICSP (In-Circuit Serial Programming): Para la programación directa del microcontrolador y

cargar el bootloader si es necesario y su tamaño de 101.52 mm x 53.3 mm.



Figura 4: Placa Arduino MEGA 2560.

Como se explica y da a entender mediante sus características, esta placa fue seleccionada por encima de las demás debido a su capacidad de procesamiento, el tamaño de su memoria ya que requeríamos de realizar un código de una dimensión mayor a lo habitual, y también por la cantidad de pines que nos permite trabajar con muchos sensores, actuadores y módulos de comunicación.

Componentes:

Para la comunicación inalámbrica con el dispositivo móvil hemos decidido utilizar un módulo bluetooth llamada Serial HC-05 que nos facilita la comunicación a distancia. Este componente se puede utilizar en modo maestro o esclavo. El HC-05 tiene 6 pines, status, RXD, TXD, GND, VCC y EN. Los pines RXD y TDX se conectan a los pines digitales para realizar la comunicación con el Arduino, el pin VCC a 5 voltios y el pin GND a la tierra. Es muy usado en el para dar conectividad inalámbrica a través de una interfaz serial

TTL entre Microcontroladores y otros dispositivos como PC, laptops o celulares Smartphone. Este módulo cuenta con el protocolo Bluetooth V2.0+EDR que nos permite la comunicación punto a punto de forma inalámbrica, también posee un protocolo de comunicación UART, el cual se explicará más adelante. Se alimenta con 5V y tiene una tensión de comunicación de 3.3V. La frecuencia con la que se maneja es de 2.4Ghz Banda ISM. Su velocidad de transmisión en baudios es ajustable entre 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, pero por defecto su configuración es de 9600 baud rate y la contraseña para su conexión es 1234. Por último, este módulo es compacto ya que solo mide 4.3cm x 1.6cm x 0.7cm y tiene una cobertura de aproximadamente 10 metros.

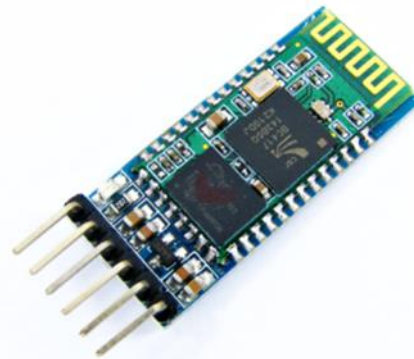


Figura 5: Módulo Bluetooth HC-05.

Como se aclara en las características de HC-05, este componente utiliza un protocolo de comunicación UART. Un UART (Universal Asynchronous ReceiverTransmitter) es un tipo de circuito integrado que se usa para enviar y recibir datos a través de un puerto serie en un equipo o dispositivo periférico. Los UART son ampliamente utilizados y conocidos por su sencillez. Sin embargo, a diferencia de SPI e I2C, los UART no admiten múltiples dispositivos subordinados.

La comunicación UART no requiere una señal de reloj compartida. En su lugar, utiliza bits de inicio y parada para definir los límites de cada paquete de datos. Los datos se transmiten bit a bit. Un paquete típico incluye un bit de inicio que indica el comienzo de un paquete, los bits de datos que generalmente son entre 5 y 9 bits por paquete, un bit de paridad (opcional) para detección de errores y por último un bit de parada que pueden ser uno o dos bits que indican el final del paquete. La velocidad de transmisión, medida en baudios (bits por segundo), debe ser la misma para el transmisor y el receptor. Los UARTs suelen permitir comunicación Full Duplex, lo que significa que pueden enviar y recibir datos simultáneamente a través de dos líneas diferentes.

Utilizamos dos módulos relé de 4 canales, esto son componentes electrónicos diseñados para controlar dispositivos de alta potencia mediante una señal de baja potencia, esto permite encender o apagar dispositivos más grandes desde un microcontrolador. En este caso, hablamos de un módulo con 4 relés que operan a 5V y pueden manejar cargas de hasta 10A cada uno. Funciona con una alimentación de 5V en el lado de la bobina, lo que lo hace compatible con la mayoría de los microcontroladores. Cada uno de los 4 relés puede manejar cargas de hasta 10A a 250V AC o 30V DC. Esto significa que pueden controlar dispositivos de alta potencia. Tiene 4

canales, lo que permite controlar 4 dispositivos de manera independiente desde un solo módulo. Cada relé puede ser controlado individualmente por un microcontrolador, lo que ofrece flexibilidad en la gestión de los dispositivos conectados. En cuanto a su interfaz de control, las entradas de control del módulo suelen estar etiquetadas como IN1, IN2, IN3, IN4, cada una de las cuales activa el relé correspondiente. Las entradas de control son de bajo voltaje (5V), por lo que se pueden conectar directamente a las salidas digitales de un microcontrolador. Algunas versiones tienen un pin de señal de activación con lógica activa baja, lo que significa que el relé se activa cuando la señal de control es baja (0V) y se desactiva cuando es alta (5V). El módulo de relé incluye optoacopladores, que son dispositivos que utilizan luz para transferir la señal eléctrica entre dos circuitos aislados. Esto proporciona un aislamiento seguro entre el lado de control (microcontrolador) y el lado de alta potencia, protegiendo el microcontrolador de picos de voltaje o interferencias que puedan provenir de los dispositivos conectados. Cada canal tiene un LED indicador que se enciende cuando el relé está activado, lo que facilita la identificación del estado de cada relé. Cuenta con un bloque de terminales de 3 pines: COM (común), NO (normalmente abierto), y NC (normalmente cerrado), COM, es el punto común del relé, NO (Normally Open), el circuito está abierto cuando el relé está desactivado y se cierra cuando el relé se activa y NC (Normally Closed), el circuito está cerrado cuando el relé está desactivado y se abre cuando el relé se activa.



Figura 6: Módulo Relé de 4 canales 5V.

Una mini bomba de agua sumergible DC 3-5V es un pequeño dispositivo diseñado para mover agua de un lugar a otro de manera eficiente. Funciona mediante un motor de corriente continua (DC) que impulsa un rotor o impulsor, el cual genera el flujo de agua. La bomba se conecta a una fuente de alimentación de 3 a 5V DC lo que hace que el motor interno de la bomba comienza a girar, por lo que el impulsor (una pequeña hélice) dentro de la bomba también gire. El giro crea una presión diferencial que succiona agua a través de la entrada de la bomba y la expulsa por la salida. Esto permite que el agua se mueva a través de la bomba a una velocidad determinada. Funciona con un rango de 3 a 5V DC. Esto la hace compatible con una amplia variedad de fuentes de energía de bajo voltaje y normalmente consume entre 100 a 200 mA de corriente, lo que la hace eficiente en cuanto al consumo energético. La bomba tiene un caudal de 70 a 120 litros por hora (L/h), lo que significa que puede mover esta cantidad de agua dependiendo del voltaje y la altura de elevación. A mayor voltaje, el caudal aumenta, mientras que a menor

voltaje disminuye. Puede elevar el agua a una altura de entre 0.4 a 1.1 metros, dependiendo de la potencia aplicada. Generalmente, estas bombas son compactas, con dimensiones aproximadas de 45 mm x 24 mm x 30 mm, lo que facilita su integración en espacios reducidos y suele pesar alrededor de 30 a 50 gramos, lo que la hace fácil de transportar e instalar. Están construidas con materiales plásticos resistentes al agua (como ABS) para evitar la corrosión y asegurar la durabilidad cuando se sumergen. El diseño está pensado para permitir la circulación del agua de forma eficiente sin fugas.

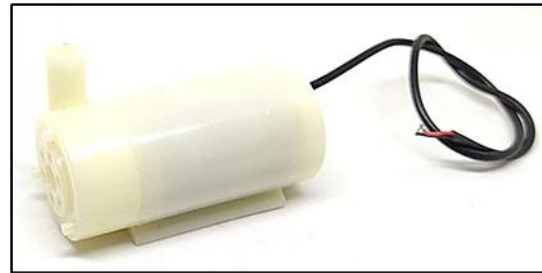


Figura 7: Mini Bomba de Agua 3-5V.

Para el bombo inicial usamos una bomba de agua sobo Wp 4000 - 2000 L/h. Esta es similar a las mini bombas de agua, pero es una versión mejorada de las mismas, esta tiene un mayor tiraje de agua y una mayor presión que las anteriores. A cambio de obtener en la cantidad y presión de agua que puede manejar esta utiliza 220V de AC, a diferencia de las anteriores que usan 5V de DC. Este tipo de bomba se usa en peceras y en fuentes decorativas de patio, nosotros la usamos para que realice el vaciado del “camión” y envía el líquido a los bombos de almacenamiento. Era necesario utilizar una bomba más potente para esta etapa del sistema ya que entre los bombos hay una electroválvula que hace necesario tener más fuerza de empuje del agua para que pase por la que está abierta,

con las primeras bombas el agua no podía pasar por la electroválvula ya que no tienen fuerza suficiente.



Figura 8: Bomba de Agua 4000 – 2000 L/h.

En la distribución de líquidos se empleó una electroválvula doble, esta es un dispositivo electromecánico utilizado para controlar el flujo de líquidos en sistemas de tuberías mediante la acción de una señal eléctrica. El contenido entra por un único conducto, y la válvula decide a cuál de las dos salidas dirigir el flujo. Este tipo de válvula es útil cuando se quiere dividir el flujo o redirigirlo a dos lugares distintos de manera alternada o selectiva dependiendo de cuál de los solenoides esté activado, el fluido se redirige a una de las dos salidas. Si se activa el solenoide 1, el fluido fluye hacia la salida 1 y si se activa el solenoide 2, el fluido fluye hacia la salida 2. Es posible, en algunos diseños, que ambas salidas estén cerradas simultáneamente si ambos solenoides están desactivados. Las electroválvulas suelen operar con voltajes de 12V, 24V, 110V, o 220V en corriente continua (DC) o corriente alterna (AC). La elección del voltaje depende del sistema en el que se va a integrar la válvula. La mayoría de las

electroválvulas dobles están diseñadas para controlar el flujo de agua, pero algunas pueden manejar gases como el aire comprimido, dependiendo de su construcción y materiales de sellado. El dispositivo puede estar fabricado en latón, acero inoxidable, o plástico de alta resistencia. El material se selecciona en función del tipo de fluido y la presión de trabajo. Las juntas de sellado usualmente están hechas de materiales como nitrilo, silicona, o EPDM, también dependiendo del tipo de fluido que manejen y la temperatura de operación. Las conexiones estándar pueden variar, siendo comunes las de 1/2 pulgada, 3/4 pulgada, y 1 pulgada en roscas BSP o NPT. El tamaño de las conexiones determina el caudal máximo que puede manejar la válvula. Las electroválvulas dobles están diseñadas para operar dentro de un rango de presiones, por ejemplo, de 0.1 a 10 bares, dependiendo del modelo. Esto es importante para garantizar un funcionamiento adecuado en sistemas de alta o baja presión. Cada una de las dos válvulas pueden ser controlada de forma independientes mediante una señal eléctrica, lo que permite controlar dos líneas de entrada para mezclar líquidos, desviar el flujo de una sola entrada hacia dos salidas diferentes y controlar el paso de fluido en dos circuitos separados, lo que aumenta la versatilidad en la gestión del flujo.



Figura 9: Electroválvula Doble.

En cuanto a la medición de líquido que pasa al último depósito se utiliza un sensor de caudal o caudalímetro que es un dispositivo diseñado para medir la cantidad de líquido que fluye a través de una tubería en un periodo de tiempo específico. En este caso, el sensor tiene un rango de medición de 0.3 a 6 litros por minuto (L/min) y una conexión de 1/4 de pulgada, lo cual se refiere al diámetro del conector de la tubería. Un caudalímetro de tipo hall-effect (efecto Hall), que es el más común para estos rangos de caudal, funciona mediante el líquido que fluye a través del cuerpo del sensor, lo cual hace girar un pequeño rotor o turbina interna. Esta turbina tiene aletas que giran proporcionalmente a la velocidad del flujo del líquido. Dentro del sensor, hay un imán incrustado en la turbina, y un sensor de efecto Hall que detecta cada vez que el imán pasa frente a él. Cada giro del imán genera un pulso eléctrico que es capturado por el sensor. Estos pulsos son enviados a un microcontrolador, que los cuenta y los convierte en una lectura de caudal. La frecuencia de los pulsos generados es directamente proporcional al caudal de líquido que pasa a través del sensor. Cuanto mayor es el flujo, más rápida es la rotación de la turbina y, por lo tanto, más alta es la frecuencia de los pulsos. Con una calibración previa, el microcontrolador puede convertir la cantidad de pulsos en una lectura de litros por minuto (L/min). Mide flujos de 0.3 a 6 litros por minuto (L/min). Esto lo

hace adecuado para aplicaciones donde se manejan flujos de baja a moderada cantidad de agua. La rosca de 1/4 de pulgada es estándar para tuberías pequeñas. Esto facilita la conexión del sensor a mangueras o tuberías de este diámetro. Normalmente, este tipo de sensores opera a un voltaje de 5V DC, lo que los hace compatibles con microcontroladores y otros sistemas de control electrónico. Proporciona una salida de pulsos digitales que representan la velocidad de flujo del líquido. Cada pulso corresponde a una cierta cantidad de agua que pasa a través del sensor. La precisión del sensor depende del modelo, pero generalmente puede tener un margen de error de entre 1% y 5% en la medición del flujo, lo que es adecuado para la mayoría de las aplicaciones no industriales de alta precisión. El cuerpo del sensor suele estar fabricado en plástico de alta resistencia o nylon para garantizar la durabilidad y resistencia a la corrosión. El diseño interno asegura que el flujo de agua no se vea significativamente afectado por la presencia del sensor, manteniendo una pérdida mínima de presión.



Figura 10: Caudalímetro 1/4".

Para la medición del nivel de líquido en los bombos se utilizan sensores HC-SR04, estos son sensores ultrasónicos ampliamente utilizados para medir distancias de manera precisa y sin

contacto. Funciona mediante la emisión y recepción de ondas ultrasónicas, lo que permite detectar la distancia a la que se encuentran objetos en su entorno. El funcionamiento del HC-SR04 se basa en el principio del eco de las ondas ultrasónicas. El sensor cuenta con dos componentes principales: el transmisor (ultrasonic transmitter) y el receptor (ultrasonic receiver). Cuando el sensor es activado, el transmisor emite una onda ultrasónica de 40 kHz (fuera del rango audible para los humanos). Esta onda viaja a través del aire hasta que choca con un objeto y se refleja de vuelta hacia el sensor. El receptor del HC-SR04 capta la onda reflejada y mide el tiempo que tarda en regresar desde el objeto hasta el sensor. Con la fórmula $Distancia = (Tiempo \times Velocidad\ del\ sonido) / 2$, se calcula la distancia al objeto. La velocidad del sonido en el aire es de aproximadamente 343 metros por segundo a 20°C. La división por 2 se debe a que el tiempo medido corresponde al recorrido de ida y vuelta de la onda. La distancia calculada se da en centímetros (cm) o milímetros (mm) según el procesamiento de la señal por el microcontrolador. Mide distancias desde 2 cm hasta 400 cm (4 metros). Esto lo hace adecuado para detectar objetos en un rango relativamente amplio en aplicaciones de corto a medio alcance. La precisión del HC-SR04 es de aproximadamente 3 mm, lo que permite obtener mediciones bastante exactas

para la mayoría de las aplicaciones de detección de proximidad. Tiene un ángulo de detección de aproximadamente 15 grados, lo que significa que es más efectivo cuando se apunta directamente hacia el objeto a medir. Esto lo hace ideal para aplicaciones donde se requiere un haz de detección enfocado. Funciona con una tensión de 5V DC, lo que lo hace compatible con microcontroladores y otros sistemas de control de 5V. Tiene cuatro pines, **VCC es la conexión de 5V** para alimentar el sensor, **GND es la conexión a tierra**, **Trigger (TRIG) es el pin de entrada** que recibe la señal para iniciar la emisión del pulso ultrasónico y **Echo (ECHO) es el pin de salida** que emite una señal de alta (HIGH) durante el tiempo que la onda tarda en regresar. La duración de esta señal se utiliza para calcular la distancia.

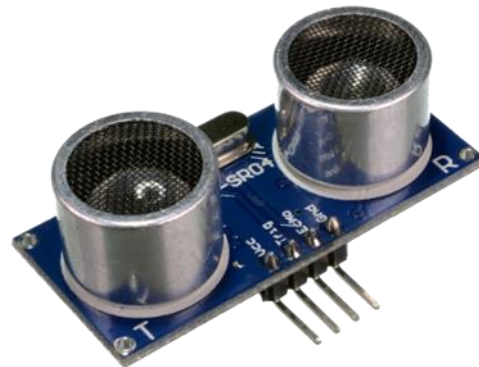


Figura 11: Sensor Ultrasónico HC-SR04.

En el mezclador se utiliza un mini motor DC (motor de corriente continua) es un motor eléctrico que convierte la energía eléctrica en energía mecánica en forma de rotación, utilizando una fuente de alimentación de corriente continua (DC). Son motores compactos y ligeros. Su tamaño reducido y facilidad de control los hacen ideales para proyectos que requieren un movimiento preciso y de bajo consumo. Los componentes internos básicos son:

- **Rotor (Armature):** Es la parte móvil del motor que rota cuando se aplica corriente. Contiene bobinados de cobre.
- **Estator:** Parte fija que genera un campo magnético. En mini motores, el estator suele estar compuesto por imanes permanentes.
- **Colector (Commutator):** Conjunto de segmentos de cobre que ayuda a cambiar la dirección de la corriente en el rotor, permitiendo la rotación continua.
- **Escobillas (Brushes):** Son piezas de contacto que conectan la fuente de alimentación al rotor a través del colector, asegurando la transmisión de corriente.

Su diseño pequeño y liviano los hace ideales para proyectos donde el espacio es limitado y se requiere un motor de bajo peso. Los mini motores DC suelen funcionar a voltajes bajos, típicamente entre 1.5V y 12V. Esto los hace compatibles con baterías y fuentes de alimentación de baja potencia como las que se utilizan en proyectos de electrónica. Un voltaje más alto generalmente proporciona una velocidad de rotación mayor, mientras que un voltaje más bajo reduce la velocidad. El consumo de corriente varía según la carga y el tamaño del motor, pero en general, los mini motores tienen un consumo bajo (desde unos pocos miliamperios hasta 1A o 2A en

condiciones de carga máxima). Su velocidad de rotación se mide en revoluciones por minuto (RPM), y puede variar ampliamente dependiendo del modelo y el voltaje aplicado. Algunos mini motores pueden alcanzar entre 3000 RPM y 15000 RPM. La velocidad se puede controlar mediante la modificación del voltaje de entrada o utilizando un controlador de velocidad (controlador PWM). Aunque los mini motores DC son rápidos, su par motor (la fuerza de giro que pueden generar) es limitado debido a su tamaño compacto. En aplicaciones que requieren un par mayor, es común acoplar estos motores a cajas de engranajes para reducir la velocidad y aumentar el torque. Las escobillas y el colector son piezas que están en constante contacto, lo que puede provocar un desgaste con el tiempo. Los mini motores DC requieren mantenimiento si se usan de forma prolongada. Existen versiones de mini motores sin escobillas (brushless) que son más duraderas, aunque son más complejas y requieren controladores especializados.



Figura 12: Mini Motor DC.

Por último, para alimentar al sistema se utiliza una fuente de alimentación 5V 3A es un dispositivo que convierte la corriente alterna (AC) de la red eléctrica en corriente continua (DC) con el voltaje adecuado. Esta fuente está diseñada para proveer una alimentación estable y segura a los dispositivos, asegurando su funcionamiento

continuo y sin interrupciones. La referencia a 3A indica que la fuente de alimentación puede proporcionar un máximo de 3 amperios de corriente mientras que el 5V hace referencia a la tensión que entrega a la salida, a su vez el diseño en forma de cargador de celular ayuda a disminuir el tamaño del sistema en comparación de si usáramos una fuente de alimentación en base a transformadores y cuadro de diodos. La fuente de alimentación se conecta a la corriente alterna (AC) de la red eléctrica, generalmente de 220V. Mediante un transformador switching, la fuente convierte el voltaje de entrada de la red a un voltaje más bajo adecuado, según lo indicado 5V de DC. El voltaje reducido pasa por un circuito que mediante la conmutación de un circuito convierte la corriente alterna en corriente continua (DC). Luego, se realiza un proceso de filtrado mediante capacitores para eliminar cualquier tipo de ripple o ruido en la señal de salida, proporcionando una corriente continua estable y adecuada para equipos electrónicos sensibles. La fuente de alimentación tiene una capacidad de 3 amperios. Esto significa que puede alimentar múltiples dispositivos simultáneamente, siempre y cuando la suma de los consumos de los dispositivos no exceda los 3A.



Figura 13: Fuente de Alimentación de 5V.

4. Desarrollo

El funcionamiento de nuestro proyecto consiste que una vez que llega el camión cisterna a descargar la materia prima, el operario a través de la aplicación instalada en un dispositivo móvil y mediante la característica bluetooth podrá comunicarse con el sistema para decidir a qué depósito desea enviar la carga. Una vez seleccionado comenzará a funcionar una bomba que, en conjunto con una electroválvula, se encargará de que la materia prima llegue al destino seleccionado. En caso de que el depósito este lleno o se llene rápidamente aun habiendo materia prima dentro de camión, el proceso se detendrá y se le alertará al operario de la situación.

Por otro lado, gracias a los sensores ultrasónicos además de poder frenar el proceso cuando ocurre la situación mencionada anteriormente, también se puede avisar al operario cuando el nivel de cualquier depósito disminuye de un cierto valor. Permitiendo que se haga un pedido de más materia prima antes de que se termine la reserva y se frene la producción.

En cuanto a la parte de elaboración del producto, con la aplicación en conjunto con los sensores y actuadores tenemos diferentes opciones en cuanto a cómo configurar el proceso de elaboración.

El operario podrá elegir la cantidad de que quiere que venga desde cada deposito, esto es posible gracias a los caudalímetros que nos indican la cantidad de fluido que paso por los tubos de alimentación.

A su vez, mediante la medición obtenida con el sensor ultrasónico colocado en el mezclador, podremos indicarle al operario si vació o no el mezclador una vez terminado el proceso o si la cantidad introducida de líquido supera la capacidad de este.

También con un motor que cuenta con una paleta en la punta podremos realizar una acción de mezclado para revolver el fluido y que este se mezcle y quede homogéneo. El operario a través de la aplicación podrá establecer cuanto debe durar este proceso o detenerlo por si hay alguna emergencia.

Finalmente, cuando el operario lo disponga, podrá a través de la app, activar el mecanismo de vaciado, que consistirá en que se ponga en marcha un motor que enviara la producción a otro deposito, a una planta de envasado o a otro camión cisterna que se encargara de transportarlo a otro destino.

Explicación del código Arduino Mega 2560:

A lo largo del escrito se mencionó la idea del proyecto, como se iba a desempeñar, los componentes utilizados y cuál era su finalidad, pero para poder culminar con dicho trabajo se requiere de una programación para que el dispositivo

digital, el Arduino Mega 2560, sea capaz de realizar las acciones, mediciones y recolección de datos de forma automática. Estos códigos estarán a su vez relacionados con la aplicación móvil diseñada para la comunicación de datos y ejecución de las actividades. Cabe aclarar que se hará mención de esta aplicación, pero será explicada, tanto el diseño como su programación, más detalladamente en la posterioridad del informe.

La programación se realiza en el entorno Arduino IDE que es una plataforma de desarrollo de código abierto utilizada para programar microcontroladores Arduino. Este entorno facilita la escritura, compilación y carga de código en la placa Arduino, permitiendo crear una gran variedad de proyectos electrónicos y de automatización. El siguiente programa ha sido desarrollado en el lenguaje de programación C/C++ propio del Arduino IDE y está diseñado para automatizar el control de niveles de líquido en depósitos, la activación de bombas y la mezcla de diferentes líquidos. Utiliza sensores de nivel (ultrasónicos) para medir los niveles de los depósitos y caudalímetros para medir la cantidad de líquido que pasa a través de las tuberías. La comunicación Bluetooth facilita la interacción con una aplicación móvil para supervisar y controlar el sistema.

Para empezar, se incluyen dos librerías, la primera llamada *TimerFive* se utiliza para configurar el Timer 5 del microcontrolador, permitiendo ejecutar funciones a intervalos de tiempo específicos, en nuestro caso es para poder leer datos provenientes de la comunicación Bluetooth de forma periódica. La segunda, *SoftwareSerial*, permite crear una comunicación

serial en pines digitales diferentes del puerto serial por defecto, para mantener disponible la conexión serial principal para la comunicación con la PC. Esta librería nos permite definir la comunicación Bluetooth mediante los pines 11 y 12, que luego dentro de la función *SoftwareSerial BT()*, definimos al pin 11 para transmitir (Tx) y al pin 12 para recibir (Rx) datos del módulo Bluetooth.

En *void Setup()* se realiza la configuración inicial del programa, en esta instancia se preparan los pines, sensores e interrupciones que se necesitan para el funcionamiento del sistema.

Primero configuramos el puerto serie para mandar señales de control al monitor de la PC. Mediante la sentencia *Serial.begin(9600)* se inicia la comunicación con una velocidad de 9600 baudios para enviar y recibir datos. Continuamos con la configuración de los pines de los sensores ultrasónicos. Se define como INPUT a los pines que reciben las señales de rebote (eco) y como OUTPUT a los que emiten el pulso ultrasónico (trig). Definimos los pines como salidas para controlar las bombas (DC y AC), el motor de mezcla DC y las electroválvulas que regulan el flujo de líquido. Luego establecemos los estados iniciales de los dispositivos, mediante HIGH las bombas están apagadas y las válvulas cerradas. Configuramos interrupciones externas en los pines 2 y 3 para detectar los pulsos generados por

los caudalímetros. En la función se incluye *pulse1* y *pulse2* que manejan los eventos para actualizar los datos de flujo en las variables *waterFlow1* y *waterFlow2*. Con RISING determinamos que disparará la interrupción cuando haya un flanco de subida en la señal, es decir, al inicio de un pulso. Con la sentencia *BT.begin(9600)* iniciamos la comunicación Bluetooth con una velocidad de 9600 baudios. Esto nos permite enviar y recibir datos desde el dispositivo móvil donde se encuentra la aplicación. Utilizamos un bucle *for* en el que su función es encargarse de poner todas las posiciones de las cadenas que se usaran para hacer los cálculos estadísticos. Esto asegura que los cálculos sean correctos desde el inicio. Y por último configuramos el timer 5 para que cada 250 ms lea los datos que vienen del celular mediante la función lectura, la configuración se debe de escribir en micro segundos. La función *attachInterrupt* se encarga de recibir variables tipo texto desde el módulo, a su vez también transforma esos datos en variables numéricas aptas para su trabajo.

En el bucle principal, *void loop()*, se encuentra el código que permitirá la funcionalidad de programa y que se ejecuta de forma continua, en él están las funciones que realizan las acciones de, envió y recepción de datos, mediciones y cálculos mediante los sensores y dispositivos utilizados.

Empezamos con una función que nos permitirá monitorear y controlar el proceso desde la PC. Recibiremos datos cada 2 segundos al monitor serial y se llama a la función *monitoreo()* que imprimirá los valores actuales de las variables *liquido1* y *liquido2*, que representan la cantidad de líquido en los depósitos correspondientes,

indicará la duración total del proceso configurada en horas, en minutos y en milisegundos, también se presenciara el tiempo restante y el estado de la variable empezar, que mostrará "1" si el proceso ha iniciado o "0" si está detenido. Se mostrarán los valores del tiempo que el proceso estuvo en pausa, el transcurrido desde el inicio y el faltante para finalizar. Al final se visualizará el estado en el que se encuentra la mezcla, si está en curso será "1" o si no "0". Su finalidad será facilitar la identificación de problemas mostrando todos los valores clave en tiempo real.

En el control de los depósitos se hace un llamado a las funciones *nivel()* y *filtrado()*. Con la primera función nos encargamos de obtener la distancia del contenido con los sensores ultrasónicos. La función recopila datos de cada sensor secuencialmente y los almacena en sus variables específicas. Al inicio se emplea un bucle *while* con la condición de que tiene que ser menor que 3 para asegurarse de iterar esa cantidad de veces, una por cada sensor. Se establece que cada 5 microsegundos el sensor se activa enviando un pulso por el pin *trig*. Para medir el tiempo de rebote se emplea la función *pulseIn* que mide el tiempo, en microsegundos, durante el cual el pin *eco* permanece en alto. Este tiempo representa el viaje de ida y vuelta de la señal ultrasónica. Una vez obtenida la duración, se calcula la distancia mediante una operación de división con

la constante 58.2, derivada de las especificaciones del sensor, basada en la velocidad del sonido en aire a 20 °C. Luego, dependiendo el índice en el que se encuentre el bucle, se almacena la distancia en una variable específica de cada sensor, y para esto se incrementa en 2 los valores de *eco* y *trig* para apuntar al siguiente par de pines asignados a los otros sensores, también se le suma 1 al índice para pasar a otra iteración. Por último, se restablecen los valores de los pines y vuelven a sus iniciales, los pines del primer sensor, y el índice se reinicia para prepararse para la siguiente ejecución de la función.

La función *filtrado()* se utiliza con el fin de reducir el error de las mediciones de los sensores, como usamos sensores ultrasónicos para medir el nivel de llenado de los bombos el ruido al usar ultrasonido puede generar errores en la medición, puede ser que una misma onda llegue dos veces y por eso esta función toma 10 mediciones de cada sensor y genera una salida promedio con el fin de suavizar los errores y el ruido.

Al inicio se restará la lectura más antigua de la suma total eliminando ese valor antes de agregar el nuevo. Después se almacenan las nuevas distancias en las variables y se actualiza la suma total añadiendo las nuevas lecturas (NUM_READINGS).

Se incrementa el índice para apuntar al siguiente elemento en el arreglo y si el índice supera el tamaño del arreglo (NUM_READINGS), se reinicia a 0, haciendo que el arreglo actúe como un buffer circular.

Se calcula el promedio dividiendo la suma total entre el número de lecturas (NUM_READINGS). Este promedio representa

una medida filtrada, eliminando el ruido y las mediciones espurias. Siguiendo, con la función *map* se deben ajustar las características físicas del sistema, convirtiendo el rango de distancias medidas, de 27 cm a 6 cm, en un rango de 0% a 100%. 27 representa el nivel vacío, y 6 el nivel lleno de un recipiente. Utilizando *constrain* limitamos los valores calculados para asegurarse de que no sean menores a 0 ni mayores a 100. Esto evita errores en las mediciones que puedan dar porcentajes fuera del rango esperado.

Para realizar la comunicación Bluetooth y sus acciones relacionadas con la aplicación móvil. Se establece un condicional que al transcurrir un segundo desde la última tarea ejecutada se accede a la función *enviarValores()*. Esta se encarga de enviar datos mediante el módulo Bluetooth a la aplicación del dispositivo externo. Luego de completar la emisión de la información se actualiza la variable *tiempoEnvio* con el tiempo actual. Esto reinicia el contador para que el intervalo de un segundo vuelva a comenzar.

La función *enviarValores()* al cumplir con su condicional envía los valores leídos a la aplicación a través de Bluetooth. Se ejecuta cada vez que se recibe el carácter que indica una nueva conexión, en este caso definimos 'E'. La transmisión utiliza el comando *BT.print()* donde se añade la variable con el dato que queremos imprimir a través de los

puertos Rx y Tx de conexión establecidos anteriormente. Luego para se repite la sentencia, pero esta vez utilizará "I" para delimitar cada valor y asignarle a cada uno una posición en una lista que podrá ser interpretada como número, de realiza de esta forma ya que la aplicación Android lo recibirá todo como texto y eso nos ayudará para mostrar o visualizar todos los datos que queramos. Estas interacciones con la aplicación y su comportamiento se detallarán cuando se explique su conformación y programación en la posterioridad.

Continuamos con el llamado a la función *llamadaRepo()* que se encarga de la comunicación para controlar el llenado de los bombos y la mezcla dependiendo de la combinación obtenida.

Con el uso de dos condiciones definimos a cuáles de los bombos queremos reponer y la cantidad de líquido a depositar. Si la variable combinación está entre 1000 y 1100, el bombo 1 es seleccionado y la cantidad a reponer es la diferencia con 1000. En caso de que sea entre 2000 y 2200, se selecciona el bombo 2 y la cantidad se obtiene restandole 2000.

Dentro de un condicional se empleará la variable *flagParadaR* que, al estar en 0, el sistema estará activo y podrá realizar la reposición. En el caso de que se presione parar por algún motivo u ocurra un error, se pondrá en 1 el valor de la variable perteneciente al condicional y la reposición se detiene, se desactivarán las siguientes ejecuciones y se deberán cargar los valores nuevamente si se quiere reanudar. Cuando se accede al bombo 1 para poder depositar el líquido en él, se presenta un condicional con la función de verificar errores, en

caso de que el nivel actual del contenido ya supera o iguala la cantidad deseada y la bomba no está activa, se genera un error con el código 722. Este código es enviado a la aplicación y al detectar este valor, dará una alerta indicando lo sucedido. Posteriormente se establece otra condición para iniciar la reposición. Si el nivel es menor que el objetivo y el bombo no está lleno, se activa la bomba y la electroválvula para llevar a cabo el proceso de llenado. Y para finalizar con la reposición al alcanzar el nivel deseado, la bomba y la electroválvula se apagan, y el sistema se prepara para una nueva operación. Al acceder al bombo 2 el procedimiento es el mismo pero las acciones afectan a sus variables específicas y usan los pines correspondientes, electroválvula en pin 8 y bomba en pin 9.

A continuación en el bucle principal, se contempla la función *llamadaProduccion()*, esta es responsable de gestionar la producción de una mezcla en función de los ingredientes especificados, la duración del proceso y las acciones requeridas, como pausar, continuar o desechar el proceso. En primera instancia, convierte las horas y minutos especificados a milisegundos para controlar el tiempo de mezcla. Luego calcula la cantidad de líquido que hay en ambos bombos.

Dada la orden de desechar el contenido en el bombo de mezcla hace que este se vacíe y reinicia todas las

variables relacionadas con la producción ya que se descarta el proceso que se estaba realizando. En el caso de que el nivel del depósito de mezcla es menor al 10%, se detiene el vaciado automáticamente.

Después se especifica el llenado del depósito de mezcla. Inicialmente se activa la bomba de llenado para el primer líquido si el nivel actual es menor al requerido, y este se detiene cuando se alcanza el nivel deseado. Luego se realiza el llenado con el segundo líquido una vez que el primero ha terminado. Cuando ambas bombas han terminado de llenar los líquidos, se inicia la mezcla, registrando el tiempo de inicio. Este proceso se puede alterar de forma manual. El usuario de la aplicación puede indicar que la producción se detenga mediante una pausa y reanudarla cuando él disponga. Si se cumple con la condición de pausa, el sistema detiene las bombas y el mezclador, registrando el tiempo pausado. Y se reanuda el proceso donde había quedado una vez que se recibe la acción de continuar. El proceso de mezclado cumple con un ciclo de tiempo en el que el motor se mantiene encendido por 5 segundos y luego se apaga por los próximos 2 segundos. Esta acción se repite hasta que el tiempo total de mezclado se cumpla. Una vez terminado el proceso de mezclado se activa una señal de vaciado. Culminando así con la producción de unificar los dos líquidos provenientes de los bombos 1 y 2.

Como última función presente en el bucle principal, *caudal()*, se encargan de medir el flujo de agua mediante los caudalímetros y calcular el nivel de líquido agregado al sistema.

Anteriormente se definen dos funciones que solo se ejecutan por interrupciones externas,

definidas previamente, de forma independiente generadas por los caudalímetros cuando detectan un flujo de agua. Cada vez que ocurre una interrupción, se actualiza el flujo acumulado para el líquido y se almacena su valor en variables de carácter global que luego se usarán para controlar los procesos del sistema. La primera función mide la cantidad de flujo del caudalímetro 1 y cuenta con la condición de contabilizar el flujo en caso de que el llenado del primer líquido este en progreso. Si se cumple se ejecuta la sentencia `waterFlow1 += 1.0 / 450` que corresponde a la cantidad de flujo medida en litros por cada pulso generado por el caudalímetro de 1/2 pulgada. Nosotros contamos con 1/4 pulgada, lo que le correspondería un valor de 5880 pulsos por litro, este cálculo se debería ajustar si fuera necesario.

La segunda función mide el flujo del segundo caudalímetro y es similar a la anterior pero solo se ejecuta cuando el primer líquido ya se ha llenado y el sistema está trabajando en el llenado del segundo líquido.

Para finalizar, nos quedaron por explicar funciones importantes que son implementadas para la comunicación Bluetooth con la aplicación móvil. La función `lectura()` se utiliza para leer que es lo que llega desde la aplicación del celular, decodifica los datos entrantes desde el módulo Bluetooth y según el carácter recibido, realiza acciones

específicas o almacena valores para su uso posterior. Primero comprueba que haya datos disponibles en el puerto serie Bluetooth mediante el comando `BT.available()`, en caso de que la condición se cumpla, lee el valor recibido.

Dependiendo del carácter que obtuvimos, se realizará una acción en específico. Para el control de reposición y mezclado se utilizan:

- 'F': Detiene la reposición.
- 'R': Indica cantidad máxima para reposición.
- 'G': Selecciona el bombo de reposición.
- 'S': Señala que se han configurado valores para la reposición.
- 'C' / 'c': Indica el valor de ingredientes para el mezclado y comienza el proceso, 'c' también establece la variable empezar en 1.
- 'D': Detiene el proceso de mezclado.
- 'A': Reanuda el proceso de mezclado.
- 'V': Activa el vaciado del bombo de mezcla.
- 'H': Indica las horas de duración del mezclado.
- 'h': Indica los minutos de duración del mezclado.
- 'X': Indica la orden de desechar el contenido del bombo de mezcla.

En la ejecución de la acción en base al carácter 'F', se llama a la función `frenadoReposicion()` que tiene como objetivo detener el proceso de reposición en el sistema, desactivando las bombas y las electroválvulas relacionadas con los bombos de reposición. Es una función sencilla pero crítica para garantizar un control seguro y

ordenado del sistema en situaciones donde se requiere detener el proceso de llenado.

Por último, la función a la cual la mayoría de los caracteres llaman en la función anterior, es *obtencionEntero()*. Se encarga de recibir y convertir los datos enviados desde Bluetooth, posteriores al carácter principal, en valores enteros. Estos valores se almacenan en las variables correspondientes dependiendo del contexto "g". Captura los datos de entrada mediante un bucle *while* y los guarda en una variable tipo *char* para crear un *string*.

Si la cadena tiene datos, se convierte a entero con *toInt()* y se almacena en la variable correspondiente dependiendo del valor de "g". Una vez finalizado vuelve la variable "g" a 0 y limpia la variable para poder leer posteriormente nuevos datos.

Explicación aplicación para dispositivo móvil:

Según los requisitos de del proyecto el sistema debe de tener una funcionalidad inalámbrica, dentro de las opciones que barajamos estaban los módulos bluetooth o los módulos wifi. Nosotros elegimos el módulo bluetooth para establecer comunicación con dispositivos móviles donde estos podrán controlar y monitorear el funcionamiento del sistema.

Creamos una aplicación usando "MIT APP INVENTOR", que es un

entorno de programación web de acceso gratuito y que permite crear aplicaciones totalmente funcionales para teléfonos Android, iPhones y tabletas Android/iOS basándose en la programación en bloques lo que lo hace mucho más simple que métodos tradicionales de programación.

Nuestra aplicación tiene diferentes pantallas con el fin de controlar varios aspectos del sistema de control y monitorear los valores de los sensores.



Figura 14: Pantalla de inicio.

El usuario inicia la aplicación móvil y energiza el circuito del sistema. Una luz roja en la caja de señalización confirma que el sistema está alimentado y listo para operar. A continuación, el usuario vincula el dispositivo móvil al módulo Bluetooth del sistema mediante el apartado de configuración de la aplicación.



Figura 15: Pantalla de Configuración.

La comunicación entre la aplicación y el Arduino se realiza mediante el envío de comandos formados por caracteres y valores numéricos:

- Los caracteres definen las acciones a ejecutar.
- Los valores numéricos asociados establecen los parámetros requeridos para dichas acciones.
- Algunas funciones solo requieren caracteres sin valores adicionales.

Estos comandos se transmiten a través de Bluetooth hacia el módulo receptor, que los envía al Arduino mediante comunicación serial. Al recibir los datos, el Arduino los interpreta y ejecuta las acciones necesarias sobre los actuadores del sistema.

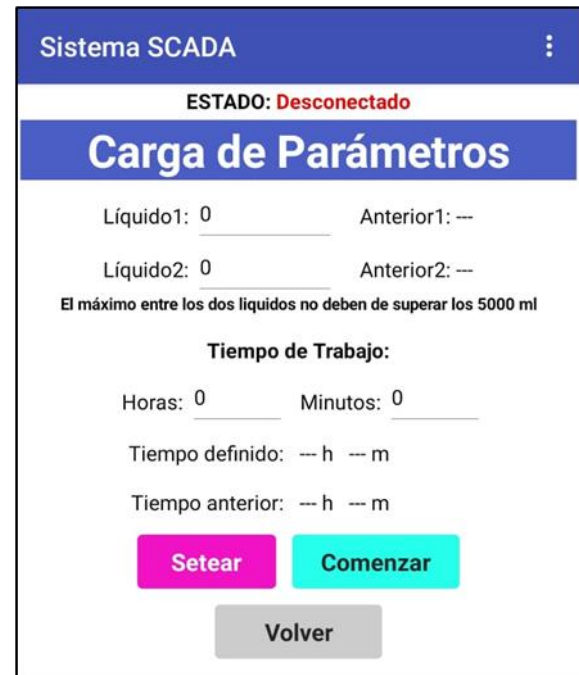


Figura 16: Pantalla de Carga de Parámetros.

Simultáneamente, el Arduino supervisa constantemente las variables de control del sistema, como niveles y flujos. Estos datos se organizan en una trama de información que el Arduino envía de regreso a la aplicación móvil, siguiendo el flujo:

- El Arduino transmite los datos al módulo Bluetooth por comunicación serial.
- El módulo retransmite la información al dispositivo móvil.

La aplicación recibe la trama, procesa los datos mediante bloques de código, actualiza las variables globales y muestra los valores en tiempo real en las pantallas correspondientes.

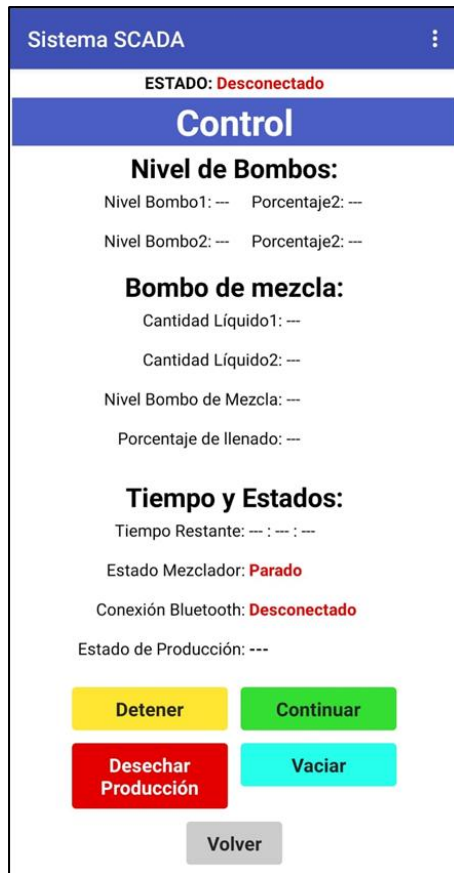


Figura 17: Pantalla de Control.

La aplicación incorpora una verificación periódica de conexión. En intervalos definidos, envía un carácter de prueba al módulo Bluetooth. Si no se recibe respuesta, el estado de la aplicación cambia a "desconectado", se desvincula del módulo, y el usuario debe restablecer la conexión desde el menú de configuración.

El sistema desarrollado simula un entorno SCADA funcional, replicando los procesos básicos de supervisión y control sin incluir los dispositivos y protocolos específicos que comúnmente se emplean en aplicaciones industriales reales. Esto permite enfocarse en la lógica operativa y la interacción entre hardware y software, ofreciendo una

aproximación simplificada pero efectiva para comprender el funcionamiento de sistemas de automatización industrial.

5. Conclusiones

El desarrollo de este sistema SCADA aplicado a una planta de producción que manipula líquidos nos permitió integrar conocimientos teóricos y prácticos en un proyecto de ingeniería. Mediante el uso de tecnologías como Arduino Mega 2560, sensores ultrasónicos, caudalímetros y módulos de comunicación Bluetooth, logramos automatizar y supervisar los procesos críticos de almacenamiento y mezcla de fluidos.

La implementación de este sistema refleja la importancia de la supervisión remota y la adquisición de datos en tiempo real, características esenciales en la Industria 4.0. Adicionalmente, el diseño modular y adaptable de nuestro sistema facilita futuras mejoras, como el control de temperatura o la integración de herramientas de registro y planificación.

Los desafíos enfrentados, desde la limitación de las aplicaciones móviles hasta problemas mecánicos en el ensamblaje, resaltaron la importancia de la planificación, la flexibilidad y el trabajo en equipo. Gracias a ello, no solo resolvimos los problemas, sino que también identificamos áreas clave para innovaciones futuras, como el registro automatizado de producción y el monitoreo de variables adicionales.

Este proyecto no solo cumplió con los objetivos iniciales, sino que también demostró cómo las herramientas tecnológicas modernas pueden transformar procesos industriales en sistemas más inteligentes y sostenibles.

6. Bibliografía

- [1] L. A. García Gutiérrez y E. Villarreal López. "Implementación de un sistema SCADA para la automatización de un laboratorio de biotecnología de nivel de seguridad biológica 3", núm. 14, pp. 119-129, junio 2009.
- [2] E. Pérez López. "Los sistemas SCADA en la automatización industrial", Vol. 28, Nº 4, pp. 3-14, octubre-diciembre 2015.
- [3] Naylamp Mechatronics. (2017, 09 de abril). Tutorial de Arduino y sensor ultrasónico HC-SR04. Disponible: <https://naylampmechatronics.com/blog/10-tutorial-de-arduino-y-sensor-ultrasonico-hc-sr04.html>
- [4] DFRobot. (2016, 13 de marzo). Water Flow Sensor - 1/8" (SKU: SEN0216). Disponible: https://wiki.dfrobot.com/Water_Flow_Sensor_-_1_8_SKU_SEN0216
- [5] MIT App Inventor. (2015, 01 de julio). Documentation. Disponible: <https://appinventor.mit.edu/explore/librariy>
- [6] Random Arduino Projects. (2018, 16 junio). Sensor de caudal con Arduino (Water Flow Sensor) [Video]. YouTube. Disponible: <https://www.youtube.com/watch?v=Jjha1pyXrZc>
- [7] Random Arduino Projects. (2022, 18 de mayo). Arduino Projects: Sensor Applications [Lista de reproducción]. YouTube. Disponible: https://www.youtube.com/playlist?list=PLCTD_CpMeEKS15LJHtAlocRCnHSLFuktA
- [8] Kio4. (2020, 01 de mayo). Proteus con Arduino. Disponible: http://kio4.com/arduino/68_Proteus.htm
- [9] EquipoPI. IC1. GitHub. <https://github.com/equipoPI/IC1>