

Tecnología en Microprocesamientos: Taller 4

Andrés Nicolás Gonzalez Lopez

Ingeniería en Mecatrónica

Universidad Tecnológica (UTEC)

Fray Bentos, Uruguay

andres.gonzalez.1@estudiantes.utec.edu.uy

Jerónimo Fernández González

Ingeniería en Mecatrónica

Universidad Tecnológica (UTEC)

Fray Bentos, Uruguay

jeronimo.fernandez@estudiantes.utec.edu.uy

Priscila Rossi Ferrari

Ingeniería en Mecatrónica

Universidad Tecnológica (UTEC)

Fray Bentos, Uruguay

priscila.rossi@estudiantes.utec.edu.uy

RESUMEN

Este informe se detalla el desarrollo e implementación de sistemas electrónicos distribuidos utilizando microcontroladores ATmega328P, enfocados en tres aplicaciones principales: un sistema maestro-esclavo, un sistema de monitoreo y control industrial, y un vehículo controlado a distancia. En la primera implementación, se estableció una comunicación SPI entre dos microcontroladores, integrando sensores, actuadores y una pantalla LCD mediante I2C. La segunda implementación aborda el control de motores y pistones en aplicaciones industriales, utilizando módulos Bluetooth HC-05 para la comunicación inalámbrica. Finalmente, la tercera implementación presenta un vehículo diseñado para una competencia, con control remoto y mecanismos funcionales. Se realizaron simulaciones previas en Proteus y pruebas en hardware para validar cada sistema, garantizando un desempeño confiable y eficiente.

INTRODUCCIÓN

En el desarrollo de sistemas electrónicos distribuidos, la comunicación entre dispositivos es importante para garantizar el intercambio eficiente de datos y la sincronización de tareas. Dos tecnologías clave en este ámbito son el protocolo Serial Peripheral Interface (SPI) y los módulos Bluetooth.

El protocolo SPI es utilizado para la comunicación de alta velocidad entre microcontroladores y periféricos en configuraciones maestro-esclavo. En este caso, SPI fue implementado en un sistema maestro-esclavo utilizando microcontroladores ATmega328P, permitiendo la integración de sensores y actuadores gracias a su capacidad de transmisión full dúplex y su flexibilidad en la conexión.

Por otro lado, los módulos Bluetooth, como el HC-05, ofrecen una alternativa inalámbrica para la comunicación entre dispositivos. Lo que permite conexiones a distancias medias y es ideal para aplicaciones que requieren control remoto o supervisión sin la necesidad de cables. Bluetooth fue utilizado en un sistema de monitoreo y control industrial para establecer comunicación inalámbrica entre dos microcontroladores, facilitando la sincronización de tareas como

la inversión de giro de motores y el control secuencial de pistones. También fue utilizado en el control del auto en el concurso de autos.

Objetivo General

Implementar sistemas electrónicos distribuidos utilizando microcontroladores ATmega328P, integrando sensores, actuadores, y módulos de comunicación para resolver problemáticas específicas en aplicaciones de control.

OBJETIVOS ESPECÍFICOS

Implementación A: Sistema Maestro-Esclavo

- Diseñar un sistema maestro-esclavo utilizando microcontroladores ATmega328P conectados mediante comunicación SPI.
- Integrar sensores como DHT11, potenciómetros y botones en el microcontrolador maestro para la adquisición de datos ambientales y de usuario.
- Implementar el microcontrolador esclavo para gestionar actuadores como motores, LEDs y buzzer, siguiendo las órdenes recibidas del maestro.

Implementación B: Sistema de Monitoreo y Control Industrial

Sistema de Motores:

- Diseñar un sistema de control para la inversión de giro de motores AC de 380/220V utilizando relés y contactores.
- Integrar botones para la activación manual como la automatización del motor, incluyendo una funcionalidad de parada de emergencia.
- Desarrollar un sistema que evite la activación simultánea de los relés de giro.

Sistema de Pistones:

- Implementar un sistema de control para la activación secuencial de pistones y el sistema de vacío mediante relés.
- Configurar un modo automático que permita la ejecución completa del sistema durante un tiempo definido.

Implementación C: Vehículo Controlado a Distancia

- Diseñar y construir un vehículo eléctrico controlado remotamente utilizando sistemas de control Bluetooth.
- Incorporar un mecanismo de ataque seguro y efectivo para romper globos de los vehículos contrincantes durante la competencia.

I. MARCO TEÓRICO

ATmega328P

El ATmega328P es un microcontrolador de 8 bits de la familia AVR de Microchip, ampliamente utilizado en proyectos de electrónica y sistemas embebidos debido a su bajo costo y facilidad de uso. Dispone de una memoria EEPROM interna para almacenamiento persistente, puertos digitales y analógicos, y módulos de comunicación como UART, SPI e I2C, que lo hacen ideal para aplicaciones de control y automatización.

Protocolo SPI

El protocolo *Serial Peripheral Interface* (SPI) es una interfaz serial síncrona diseñada para la comunicación a corta distancia en un modelo maestro-esclavo. Este protocolo permite la conexión de varios dispositivos secundarios (esclavos) a un maestro, soportando velocidades de transferencia de datos de hasta 50 MHz, lo que lo hace significativamente más rápido que otros protocolos como I2C. Una de sus características distintivas es su capacidad de comunicación full dúplex, que permite transmitir y recibir datos simultáneamente. A diferencia de I2C, SPI no sigue un estándar estricto, lo que lo hace más flexible, aunque menos uniforme para algunas aplicaciones.

Líneas de Comunicación:

- SCK (Serial Clock): Línea de reloj generada por el maestro para sincronizar la transmisión de datos.
- CS (Chip Select): Línea utilizada por el maestro para seleccionar al esclavo con el que desea comunicarse.
- MOSI (Master Out Slave In): Línea de datos del maestro hacia el esclavo.
- MISO (Master In Slave Out): Línea de datos del esclavo hacia el maestro.

Procedimiento de Comunicación:

- El maestro selecciona un esclavo específico bajando su línea CS correspondiente.
- Genera pulsos de reloj en la línea SCK para sincronizar la transmisión de datos.
- Los datos se envían a través de MOSI desde el maestro al esclavo, mientras que el esclavo puede enviar datos simultáneamente al maestro a través de MISO.
- Una vez completada la transferencia, el maestro libera al esclavo subiendo la línea CS, permitiendo la comunicación con otros dispositivos.

Características del protocolo I2C	
Modo de Comunicación	Serial síncrono
Modelo de Comunicación	Maestro-Múltiples Esclavos (bus compartido)
Líneas de Comunicación	2 líneas (SDA y SCL)
Velocidad Máxima	Hasta 5 MHz (modo ultra rápido)
Control de Esclavo	Dirección de dispositivo
Número de Dispositivos	Múltiples esclavos en el mismo bus
Protocolo	Más complejo (requiere direccionamiento y ACK)
Consumo de Pines	Menor (2 pines compartidos)
Idoneidad	Aplicaciones de baja velocidad y múltiples dispositivos en un bus

TABLE I
CARACTERÍSTICAS DEL PROTOCOLO I2C

Características del protocolo SPI	
Modo de Comunicación	Serial síncrono
Modelo de Comunicación	Maestro-Múltiples Esclavos (selección individual)
Líneas de Comunicación	4 líneas (SCK, CS, MOSI, MISO)
Velocidad Máxima	Hasta 50 MHz
Control de Esclavo	Línea de selección de chip (CS)
Número de Dispositivos	Múltiples esclavos con líneas de CS
Protocolo	Más simple (no requiere ACK)
Consumo de Pines	Mayor (mínimo 4 pines)
Idoneidad	Aplicaciones de alta velocidad y transferencia rápida de datos

TABLE II
CARACTERÍSTICAS DEL PROTOCOLO SPI

Bluetooth HC-05

El módulo Bluetooth HC-05 es un dispositivo ampliamente utilizado en sistemas embebidos para establecer comunicación inalámbrica mediante el estándar Bluetooth. Opera en la banda de 2.4 GHz y soporta modos maestro y esclavo, lo que lo hace adecuado para aplicaciones donde un dispositivo central coordina la interacción con varios periféricos. Entre sus características más destacadas se incluyen un alcance de hasta 100 metros, baja latencia en la transmisión de datos (hasta 2.1 Mbps) y bajo consumo energético, lo que lo hace ideal para sistemas portátiles. La configuración del HC-05 se realiza a través de comandos AT enviados por una interfaz UART, permitiendo ajustes como el cambio de nombre, velocidad de transmisión y rol (maestro/esclavo). Para garantizar una comunicación estable, se recomienda trabajar con una velocidad de 38400 baudios en modo AT. En aplicaciones prácticas, este módulo se integra con microcontroladores como el Arduino, permitiendo la transmisión y recepción de datos en tiempo real. Además, soporta múltiples conexiones en modo maestro, lo que lo diferencia de módulos similares como el HC-06. En este proyecto, el HC-05 se utiliza para establecer una comunicación inalámbrica entre los microcontroladores maestro y esclavo, permitiendo la sincronización de las órdenes en los sistemas de control, como la activación de pistones y relés.

Bluetooth Esclavo	
AT Command	AT
Nombre	AT+NAME=BTEsclavoTostatronic
Contraseña	AT+PSWD=0000
UART	AT+UART=9600,0,0
Rol	AT+ROLE=0 (0 = Esclavo, 1 = Maestro)
Modo	AT+CMODE=1 (1 = Conexión a cualquier MAC)
Dirección MAC	AT+ADDR (Extraer dirección MAC del dispositivo)
MAC	MAC:2016,5,310042

Bluetooth Maestro	
AT Command	AT
Nombre	AT+NAME=BTMaestroTostatronic
Contraseña	AT+PSWD=0000
UART	AT+UART=9600,0,0
Rol	AT+ROLE=1 (0 = Esclavo, 1 = Maestro)
Modo	AT+CMODE=0 (0 = Conexión a una MAC específica)
Vinculación	AT+BIND=???? (Ingresar MAC del Esclavo con : reemplazados por ;)

TABLE III
CONFIGURACIÓN DE MÓDULOS BLUETOOTH ESCLAVO Y MAESTRO

Comunicación serial USART-UART

La comunicación serial USART-UART (Universal Asynchronous Receiver-Transmitter) permite la transmisión de datos bit a bit entre dispositivos mediante una línea de recepción (RX), una línea de transmisión (TX) y una conexión a tierra común (GND). Opcionalmente, se puede utilizar una línea de reloj (CLK) para sincronización en comunicaciones síncronas. Este protocolo es asíncrono por defecto, lo que elimina la necesidad de una línea de reloj externa, y es ampliamente utilizado en sistemas embebidos debido a su simplicidad y eficiencia.

La UART se inicializa configurando el baud rate mediante los registros UBRRH y UBRL, habilitando la transmisión y recepción con los bits TXEN0 y RXEN0, y definiendo el formato de datos (generalmente 8 bits por trama). Para enviar datos, se utilizan funciones como UART_sendChar, que envía un carácter tras verificar la disponibilidad del registro UDR0, o UART_sendString, que transmite cadenas de texto carácter por carácter. De forma similar, la función UART_receiveChar permite recibir datos al leer el registro UDR0 una vez que los datos están disponibles.

Protocolo I2C

El protocolo I2C (Inter-Integrated Circuit) es un estándar de comunicación serial síncrona que utiliza dos líneas, SDA (Serial Data Line) para el transporte de datos y SCL (Serial Clock Line) para la sincronización, permitiendo conectar hasta 112 dispositivos en un mismo bus. Un dispositivo maestro controla el reloj y gestiona la comunicación con los esclavos, identificados mediante direcciones únicas de 7 o 10 bits. La comunicación inicia cuando el maestro genera una condición de inicio (SDA pasa de 1 a 0 mientras SCL está en 1), seguida por el envío de la dirección del esclavo y un bit de lectura o escritura. El esclavo responde con un bit de reconocimiento (ACK) y comienza la transferencia de datos en paquetes de 8 bits, cada uno confirmado con un ACK. Al finalizar, el maestro genera una condición de parada (SDA

pasa de 0 a 1 mientras SCL está en 1), liberando el bus. Este protocolo es ideal para sistemas embebidos gracias a su simplicidad, bajo consumo de pines y capacidad de soportar múltiples dispositivos, aunque puede requerir resistencias pull-up y es más lento que otros protocolos como SPI.

Sensor DHT11

El sensor DHT11 es un dispositivo ampliamente utilizado en la medición de variables ambientales como temperatura y humedad debido a su simplicidad, bajo costo y capacidad para integrarse en proyectos electrónicos. Este sensor opera con un protocolo de comunicación de un solo hilo (1-Wire), que permite la transmisión de datos de manera eficiente entre el sensor y el microcontrolador, utilizando un solo cable para comunicación bidireccional. Este protocolo se caracteriza por su diseño maestro-esclavo, donde el microcontrolador actúa como el maestro que inicia las comunicaciones y el sensor como esclavo que responde con los datos requeridos.

El DHT11 utiliza un termistor de coeficiente de temperatura negativo (NTC) para medir la temperatura, aprovechando el principio de que la resistencia del material semiconductor disminuye con el aumento de la temperatura. Por otro lado, para medir la humedad, incorpora un sensor capacitivo compuesto por dos electrodos y un material dieléctrico en medio, cuya capacitancia varía según el nivel de humedad presente en el ambiente.

El proceso de comunicación del sensor incluye la transmisión de datos que combinan información sobre temperatura, humedad y un mecanismo de verificación (checksum) para garantizar la integridad de los datos recibidos.

Puente H

El puente H es un circuito electrónico que permite controlar la dirección de giro de un motor de corriente continua (DC) utilizando un conjunto de interruptores o transistores. Este diseño, ampliamente empleado en sistemas de robótica y control, es esencial para cambiar la polaridad de la corriente suministrada al motor, lo que determina si gira en sentido horario o antihorario. Un puente H típico consta de cuatro interruptores dispuestos en forma de "H". Cuando se activan dos interruptores en esquinas opuestas, la corriente fluye en una dirección, haciendo que el motor gire hacia un lado. Al cambiar el par de interruptores activados, la corriente fluye en la dirección opuesta, invirtiendo el giro del motor. Este sistema puede ser controlado mediante señales digitales provenientes de microcontroladores como el Arduino UNO, permitiendo además modular la velocidad del motor mediante técnicas como PWM (modulación por ancho de pulso). El utilizado para este caso es el de la 10.

METODOLOGÍA

Parte A (Comunicación SPI)

El proceso comenzó con la simulación en PROTEUS para establecer la comunicación SPI entre dos microcontroladores ATmega328P, configurados como maestro y esclavo, respectivamente. Se realizaron las conexiones necesarias entre los pines MISO, MOSI, SCK y CS. Además, se conectó un

botón al microcontrolador maestro y un LED al esclavo. Posteriormente, se desarrolló un código base que incluyó la comunicación UART, lo cual permitió comprobar la transmisión de datos de manera eficiente.

Una vez verificada la comunicación SPI, se integraron los demás componentes del sistema. Se añadieron en el esclavo un LED, un buzzer y otro LED que simula un motor, cada uno configurado para ser operado mediante un botón independiente. También en el maestro se incorporó un sensor de temperatura y humedad DHT11 y una pantalla LCD con módulo I2C, lo que permitió optimizar el uso de los pines del microcontrolador maestro, ya que estos eran limitados debido a la cantidad de componentes conectados.

En la pantalla LCD, inicialmente se mostró un mensaje de bienvenida con un retardo programado. Posteriormente, se visualizaron en pantalla los datos de temperatura y humedad enviados por el sensor DHT11. A través del protocolo SPI, al accionar un botón, se comandaba la activación de un LED, mostrando un mensaje en la pantalla cuando este estaba encendido. Al presionar nuevamente, el LED se apagaba, acompañado de un mensaje en pantalla que indicaba su estado apagado. El mismo comportamiento se implementó para el buzzer y para el LED que simula el encendido y apagado de un motor.

Parte B (Inversión de giro)

Inicialmente, se desarrolló una simulación que utiliza LEDs para representar los relés del circuito. Esto se realizó como una medida de seguridad con el objetivo de prevenir accidentes en el laboratorio debido a posibles errores en el código. En la Fig. 7, que se encuentra en los anexos, se presenta el circuito simulado diseñado para este propósito. A partir de la simulación, se elaboró el código maestro del sistema, el cual es responsable de enviar las órdenes al microcontrolador esclavo. Este último controla directamente los relés encargados de gestionar el motor. Una vez que la simulación validó el correcto funcionamiento del código maestro, se procedió con la configuración de los módulos Bluetooth HC-05 mediante comandos AT. Durante esta etapa, se asignaron los roles de maestro y esclavo a los módulos y se estableció el vínculo entre ellos para garantizar la transmisión de mensajes. Para el microcontrolador esclavo, se desarrolló un código específico. Este código fue una adaptación del código maestro, ya que la simulación previamente diseñada incluía la interacción simulada entre ambos dispositivos mediante los LEDs. Una vez completada la configuración de los módulos Bluetooth y comprobado el vínculo, el código esclavo fue ajustado para controlar directamente los relés. Finalmente, se verificó la interacción correcta entre ambos módulos Bluetooth. Para esto se comprobó que los LEDs asociados a los relés, al presionar los botones asignados, se enciendan. Con estos pasos completados, se procedió a alimentar el sistema con corriente eléctrica y se implementó el sistema de comunicación inalámbrica entre los dos microcontroladores utilizando los módulos Bluetooth.

Parte B (Control de Pistones)

Para llevar a cabo esta parte del práctico, primero se observó físicamente el circuito implementado para identificar la lógica que presentaba y determinar la función de cada relé, con el fin de simular correctamente el control del sistema en PROTEUS y evitar errores en la secuencia requerida. Una vez que se identificaron los relés y sus actuadores correspondientes, se procedió a realizar la simulación del sistema, utilizando LEDs para representar cada actuador. Inicialmente, se empleó un Arduino para verificar que el funcionamiento del maestro y el esclavo fuera eficiente tanto en modo manual como en automático, asegurando además la sincronización de los accionamientos y evitando activaciones simultáneas. Una vez finalizada la simulación y validada la lógica del sistema, se procedió a implementar el módulo Bluetooth HC-05. Para su funcionamiento, se configuraron ambos módulos mediante comandos AT, definiendo uno como maestro y el otro como esclavo. Una vez sincronizados y con la comunicación establecida, se implementó UART en el desarrollo del código, lo que permitió visualizar los caracteres enviados al presionar un pulsador, identificarlos y desarrollar la lógica necesaria para comandar desde el módulo Bluetooth maestro al esclavo.

Parte C (Auto)

El desarrollo del sistema para el automóvil eléctrico comenzó con la selección de uno de los modelos proporcionados en el kit del laboratorio. Una vez que se armó el automóvil, se procedió a diseñar e integrar un mecanismo de ataque que consistió en una estructura hecha por alfileres y palos de brochetas para poder explotar el globo de los oponentes sin dañar sus automóviles. Dicho mecanismo fue fijado en el parachoques del automóvil para tener una mayor estabilidad durante la competencia y no interfiriera con su movilidad.

Para el control remoto, se utilizó un único módulo Bluetooth HC-05 conectado a un microcontrolador ATmega328P como esclavo. Este módulo fue configurado para recibir comandos enviados desde la aplicación móvil "Arduino Car", que permitió controlar de forma inalámbrica los movimientos del vehículo. Además, se utilizó la placa "UNO-F5 board" para manejar el giro de los motores.

El código fue desarrollado para interpretar los comandos recibidos vía Bluetooth y traducirlos en acciones precisas: movimientos hacia adelante, atrás, giros a la izquierda o derecha, y la activación del mecanismo de ataque. Para esto se activó un pin a la vez para comprender el sentido que tomaría cada motor. También se llevaron a cabo pruebas en hardware, evaluando el rango de control remoto y la precisión de los movimientos. Durante estas pruebas, se verificó la capacidad del sistema para responder correctamente a los comandos enviados desde la aplicación, garantizando una sincronización adecuada entre los motores y el mecanismo de ataque.

II. RESULTADOS

Implementación A

El sistema maestro-esclavo propuesto fue implementado con éxito, cumpliendo con la comunicación efectiva entre los microcontroladores ATmega328P mediante el protocolo SPI. Durante el desarrollo, se integraron correctamente los sensores, la pantalla LCD y los actuadores, logrando una interacción coordinada entre ambos microcontroladores. Aunque inicialmente hubo problemas en la visualización de caracteres en la pantalla LCD debido a la falta de espaciado adecuado en las líneas de texto, este inconveniente fue solucionado añadiendo espacios en ambas líneas. La comunicación SPI demostró ser estable, permitiendo el control fluido de los actuadores, incluyendo el encendido y apagado de LEDs y el funcionamiento correcto del buzzer. La transmisión de datos se llevó a cabo sin interrupciones, y la funcionalidad de los botones y de los datos obtenidos por el sensor DHT11. En la figura 3 se aprecia la implementación del sistema de comunicación SPI. En la figura 1 se observa la simulación del sistema. Ambas imágenes se encuentran en anexos.

Implementación B

Control de Inversión de giro

Los resultados obtenidos durante el desarrollo e implementación del sistema de inversión de giro confirman el correcto funcionamiento de cada etapa del proceso. En primer lugar, la simulación inicial permitió validar tanto el diseño del circuito como el comportamiento del código maestro, representando de forma segura la activación de los relés mediante LEDs. Esto garantizó que los errores potenciales fueran detectados y corregidos antes de alimentar el sistema con corriente eléctrica.

Durante la configuración de los módulos Bluetooth HC-05, se estableció una comunicación inalámbrica estable entre el microcontrolador maestro y el esclavo. La asignación de roles mediante comandos AT y el posterior enlace entre los módulos se verificaron a través de pruebas que demostraron la capacidad del sistema para transmitir y recibir mensajes sin interrupciones.

El código adaptado para el microcontrolador esclavo funcionó correctamente, respondiendo a las órdenes enviadas por el maestro para activar y desactivar los relés. La interacción entre ambos microcontroladores y los módulos Bluetooth operó sin problemas significativos, lo que asegura una respuesta inmediata del sistema.

Finalmente, al alimentar el sistema con corriente eléctrica, se observó que los relés activaban las bobinas correspondientes de forma controlada, permitiendo la inversión de giro del motor sin errores ni fallas. El sistema de seguridad integrado previno con éxito la activación simultánea de los relés y cumplió con los requisitos funcionales y de protección establecidos.

Control de Pistones

En la práctica, se logró implementar y validar el sistema de control de pistones con éxito, tanto en el modo manual como en el automático. Durante el desarrollo, se presentaron algunos inconvenientes iniciales, principalmente relacionados con la configuración de los módulos Bluetooth HC-05 y los comandos AT. Sin embargo, estos problemas fueron solucionados, permitiendo establecer una comunicación estable entre el maestro y el esclavo.

Otro desafío surgió al enviar caracteres mediante UART, ya que al presionar un botón, se transmitían valores incorrectos. Ajustando los delays, este inconveniente fue resuelto, logrando una transmisión más precisa. También se observó que al presionar simultáneamente los cuatro botones, UART enviaba correctamente los caracteres 1, 2, 3, 4, pero en la comunicación por Bluetooth no se recibían estos valores correctamente. Para solucionar esto, se modificaron los caracteres enviados por cada botón a 5, 3, 9 y A. Esta modificación permitió lograr una comunicación efectiva cumpliendo con los requisitos del sistema.

Finalmente, el sistema de control demostró un funcionamiento correcto, tanto de forma manual como automática, controlando los tres actuadores y habilitando un botón adicional para el modo automático. Esto incluyó una sincronización adecuada entre los accionamientos y el cumplimiento de la secuencia inversa al finalizar el proceso.

En las figura que se encuentra en anexos como Fig. 4 y Fig 2 se puede observar la implementación en físico del sistema y la simulación hecha respectivamente.

Implementación C

El sistema de control remoto del automóvil eléctrico fue implementado correctamente utilizando el módulo Bluetooth HC-05, conectado al microcontrolador ATmega328P, y controlado a través de la aplicación móvil "Arduino Car". Durante las pruebas de funcionalidad, se comprobó que el vehículo respondía de manera eficiente a los comandos enviados, permitiendo controlar el movimiento del automóvil de forma inalámbrica en todas las direcciones: hacia adelante, hacia atrás, y giros a la izquierda y derecha. El mecanismo de ataque, diseñado con alfileres y palos de brocheta, demostró ser efectivo para romper el globo sin dañar el automóvil del oponente. Al estar fijado en el parachoques, el mecanismo se mantuvo estable durante la competencia y no interfirió con la movilidad del vehículo. En cuanto al manejo de los motores, se verificó que los motores respondían correctamente a las señales enviadas desde el microcontrolador, lo que permitió movimientos precisos del vehículo durante las pruebas en pista. No se observaron problemas significativos de sincronización entre los motores y el mecanismo de ataque, y la estabilidad general del vehículo fue adecuada durante la ejecución de las maniobras. Se puede ver en la figura 6.

III. CONCLUSIÓN

Comunicación SPI

La implementación del sistema maestro-esclavo basado en comunicación SPI fue satisfactoria, lo que valido tanto

la propuesta como la metodología de trabajo. Los ajustes realizados en la visualización de la pantalla LCD y la optimización del código aseguraron un desempeño óptimo del sistema. La coordinación entre los sensores y los actuadores, gestionada por el microcontrolador maestro, junto con la ejecución precisa de las órdenes por parte del esclavo, demostraron que esta arquitectura es eficiente.

Control de Inversión

Se logró implementar un sistema seguro y funcional que cumple con los objetivos planteados. La simulación inicial con LEDs permitió validar el diseño del circuito y el código maestro de forma segura, evitando posibles riesgos durante las pruebas físicas. La configuración y vinculación de los módulos Bluetooth HC-05 mediante comandos AT y se estableció una comunicación inalámbrica entre el microcontrolador maestro y el esclavo. Las pruebas finales confirmaron la correcta interacción entre los módulos Bluetooth y la funcionalidad esperada del sistema. El sistema desarrollado valida la efectividad del sistema y también demuestra la viabilidad de la comunicación inalámbrica en aplicaciones industriales.

Control de Pistones

La práctica permitió implementar un sistema de control confiable para los pistones, validando tanto su funcionalidad manual como automática. Los ajustes realizados durante el desarrollo, como la configuración de Bluetooth y la optimización de UART, fueron fundamentales para garantizar un rendimiento óptimo. Este ejercicio destacó la importancia de una correcta configuración de hardware y software, así como la resolución de problemas en tiempo real, para alcanzar los objetivos propuestos con éxito.

Control del Auto

El desarrollo y la implementación del automóvil eléctrico controlado por Bluetooth fue exitoso, cumpliendo con los requisitos establecidos para el concurso. El vehículo demostró un rendimiento fiable tanto en el control remoto como en el mecanismo de ataque, permitiendo una ejecución precisa de las maniobras dentro del área de competencia. El rango de control remoto y la capacidad para romper los globos de los oponentes confirmaron la efectividad del diseño. El uso de la placa "UNO-F5 board" y el microcontrolador ATmega328P fue adecuado para manejar el sistema de control, proporcionando una comunicación fluida entre los componentes del automóvil. Las pruebas realizadas evidenciaron que el automóvil es capaz de operar de manera autónoma dentro de los parámetros establecidos, y se obtuvo un vehículo funcional que puede participar con éxito en el concurso sin comprometer su rendimiento o seguridad. La integración de todos los sistemas fue realizada con eficacia, y las evaluaciones de pruebas confirmaron la precisión y estabilidad del diseño en condiciones de competencia.

REFERENCES

- [1] https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel7810AutomotiveMicrocontrollersATmega328P_Datasheet.pdf
Datasheet puente H.

ANEXOS

Repositorio de Lab4 en github

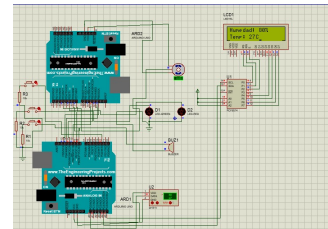


Fig. 1. Simulación Parte A

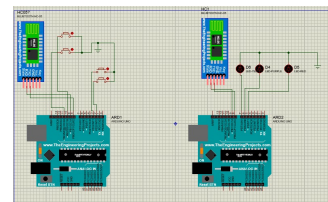


Fig. 2. Simulación Parte B

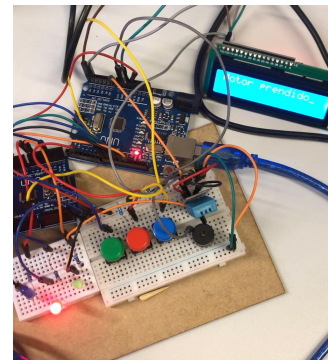


Fig. 3. Implementación A

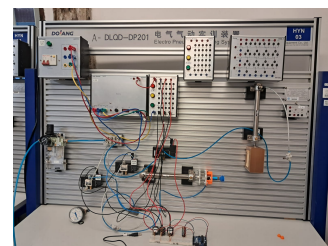


Fig. 4. Implementación B

