

Laboratorio 2

Neo Barrios
UTEC
Fray Bentos, Uruguay
neo.barrios@estudiantes.utec.edu.uy

RESUMEN

En este informe se encuentra el desarrollo e implementación de un sistema basado en la comunicación SPI, compuesta por un maestro que recibirá datos de sensores y le enviará estos datos al esclavo, el cual controlará una serie de actuadores en base a los datos recibidos por parte del maestro. También se encuentra la construcción de un auto a control remoto haciendo uso del módulo bluetooth HC-05.

También se presenta un uso práctico de diferentes conocimientos en el área de microcontroladores, como el uso de del protocolo SPI por medio de la configuración de los registros especiales del Atmega328p encargados de administrar esta labor, el uso de PWM para controlar la velocidad de giro de las ruedas del auto y el uso de comunicación bluetooth haciendo uso de una comunicación UART entre el microcontrolador y el módulo bluetooth de esta forma también configurando el módulo haciendo uso de los comandos AT.

Los resultados para la parte de comunicación SPI entre el maestro y el esclavo fueron aceptables, no obstante no se añadió el sensor DHT11 como uno de los sensores que leía el maestro y tampoco se añadió una pantalla LCD para mostrar los datos del sensor DHT11.

El auto a control remoto funcionó de forma adecuada permitiendo 8 cantidades posibles de movimientos.

INTRODUCCIÓN

En este informe se presenta el desarrollo de un sistema basado en la comunicación SPI, utilizando el microcontrolador ATmega328P programado en C. El sistema está compuesto por un maestro encargado de recibir datos de sensores y enviarlos a un esclavo, el cual controla una serie de actuadores en función de esta información.

Se diseñó un auto a control remoto utilizando un módulo Bluetooth HC-05, lo que permite una interacción inalámbrica con el sistema haciendo uso de bluetooth 2.0 + EDR.

Consigna

Implementación A: SPI: La idea consiste en implementar un sistema maestro-esclavo utilizando dos microcontroladores ATmega328P conectados mediante comunicación SPI. El sistema implementa una arquitectura distribuida con microcontroladores, donde un maestro se encarga de recopilar datos de al menos tres sensores y de enviar comandos al esclavo. Este último ejecuta acciones en al menos tres actuadores. El diseño

separa claramente las tareas de monitorización y control, optimizando la eficiencia y coordinación entre dispositivos. En este esquema:

- **Microcontrolador maestro:** Uno de los ATmega328P estará conectado a un DHT11 y a una serie de sensores a elección del grupo (LDR, PT100, botones, potenciómetros, entre otros, al menos 2). Además, contará con una pantalla LCD que mostrará en tiempo real las acciones que se envían al microcontrolador esclavo.
- **Microcontrolador esclavo:** El segundo ATmega328P estará encargado de controlar los actuadores, como motores, LEDs, buzzer, etc., respondiendo a las órdenes que recibe del microcontrolador maestro.

Implementación C: Auto: Implementar el auto modelo de los disponibles, que cumpla los siguientes parámetros:

- **Dimensiones:** El auto debe ser uno de los modelos presentados en el KIT Fitchertecnik para un auto.
- **Control remoto:** Los autos deben ser controlados a distancia usando cualquier tipo de sistema de control remoto basado en radiofrecuencia (RF), infrarrojo (IR) o Bluetooth.
- **Batería:** Los autos deben ser alimentados por baterías recargables. No se permite el uso de combustibles líquidos, sólidos o gaseosos.

OBJETIVO GENERAL

Centrarse en diferentes aplicaciones con microcontroladores en las que se hagan uso de diferentes componentes y dispositivos electrónicos, y agrupar diferentes conceptos dentro del campo de electrónica y los microcontroladores para ver como se relacionan al igual que programación básica en C orientada a micros.

OBJETIVOS ESPECÍFICOS

- Intercomunicar 2 Atmega328p haciendo uso de la comunicación SPI.
- Programar, simular y construir un sistema el cual permita monitorear datos de sensores por parte del microcontrolador maestro y manejar actuadores con el microcontrolador esclavo.
- Configurar el módulo HC-05 junto al atmega328p para recibir comandos por parte de una app mobile y controlar los motores del auto en base a los comandos aplicando diferentes PWM a cada motor.

METODOLOGÍA

Marco conceptual

Para comprender mejor los conceptos involucrados en este laboratorio, a continuación se presentan algunas definiciones clave:

1) **SPI**: El SPI (Serial Peripheral Interface) es un protocolo de comunicación serial síncrono utilizado principalmente para la transferencia rápida de datos entre un microcontrolador y periféricos, como sensores, memorias, o pantallas. Funciona en un esquema maestro-esclavo, donde el maestro controla el reloj SCLK y coordina las transferencias. Los datos se transmiten mediante dos líneas: MOSI (Master Out Slave In) para enviar y MISO (Master In Slave Out) para recibir. Además, utiliza un pin de selección de esclavo (SS) para habilitar el dispositivo con el que desea comunicarse. Al ser un protocolo rápido para enviar datos de forma serial es preferentemente usado si la cantidad de periféricos o esclavos es muy baja, como por ejemplo en la comunicación entre un maestro y un solo esclavo para avarcar 2 áreas, las cuales serían la recolección de datos y el control de mecanismos así dividiendo las tareas y mejorando el rendimiento del sistema al poseer mayor cantidad de recurso.

2) **Bluetooth y modulo HC-05**: Bluetooth es un estándar de comunicación inalámbrica para la transmisión de datos a corta distancia, que utiliza la banda de 2.4 GHz y permite la conexión entre dispositivos como teléfonos, computadoras, y microcontroladores en un rango de hasta 100 metros dependiendo de su clase siendo la clase 1 la de 100 metros (y mayor consumo de energía) y la clase 4 la de 0.5 metros (y menor consumo de energía). Por otro lado están las versiones, las cuales definen la velocidad de transferencia de datos, siendo la más rápida y actual la versión 5.0 aunque muchos dispositivos siguen usando la 4.0 hoy en día. El módulo HC-05 es de clase 2 teniendo un rango de 10 metros (aunque en entornos ideales puede alcanzar mayores distancias) y utiliza la versión 2.0 + EDR (Enhanced Data Rate), lo que le permite alcanzar velocidades de hasta 3 Mbps en entornos ideales. Este estándar es eficiente para aplicaciones de comunicación en serie, como el intercambio de datos entre microcontroladores y otros dispositivos periféricos.

Materiales

- 2 Atmega328p
- HC-05
- Potenciómetro
- 3 LED
- 3 Resistencia 0.5k
- L293D
- 2 Pulsador
- Kit Fischertechnik Robotics TXT 4.0
- Software Arduino RC Car/Tank
- Software Arduino IDE
- Software ATTEL STUDIO 7

Procedimiento

A continuación una breve explicación de los pasos llevados a cabo para afrontar cada problemática y como se procedió para cada caso.

Implementación A: -Se programó una comunicación SPI básica y se cargaron los códigos a placas arduino uno r3 en proteus 8 para ver si se lograba una comunicación correcta entre los dos micros. -Se programó la lectura de los pines y se asignó una variable de 8 bits sin signo a cada valor de lectura de los sensores, para el caso del potenciómetro se dividió el valor ADC por 4 para ajustar el tamaño de 10 bits a 8 bits de resolución del valor ADC. Posteriormente estableció que se enviaran los 3 bytes de forma secuencial hacia el esclavo. -Se programó en el Atmega esclavo un contador y un arreglo de bytes, el propósito es que si se detecta una recepción, se salta a la rutina de interrupción SPI para leer el valor de buffer y almacenarlo en una posición del array, luego utilizando el contador se asegurará el programa de que se almacenen solo tres bytes en el array. -Se estableció una lógica para aplicar un PWM a un LED o motor en base al valor ADC del potenciómetro y se asignaron lógicas para encender o apagar los 2 LED EN base a como se opriman los pulsadores.

Implementación C: -Se construyó el auto basándose en uno de los ejemplos de autos que trae el Kit. -Se configuró el módulo HC-05 haciendo usos de comandos AT enviados por el Atmega328p. -Se programó el código basándose en una estructura switch case donde existe 8 casos posibles dependiendo de la letra que se reciba desde la app móvil. En cada caso se modifican los registros referentes a los dutycycles de los PWM de los motores, de esta forma permitiendo al auto a girar en 4 direcciones extras gracias al uso de PWM. -Se incluyó la posibilidad de setear 2 modos posibles, un modo lento y un modo rápido, de esta forma si desde la app se envía un carácter u otro se puede establecer un valor para una variable de control de velocidad que permita incrementar o decrementar los valores de los PWM en todos los casos posibles.

RESULTADOS

Implementación A

Circuito simulado en proteus:

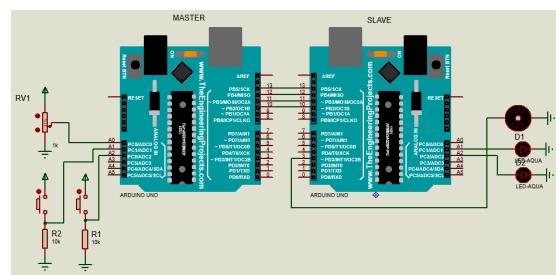


Fig. 1: Circuito en proteus-Implementación A

Circuito probado en protoboard:

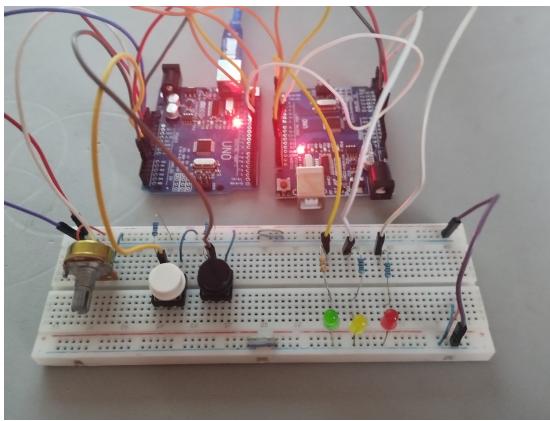


Fig. 2: Circuito en fisico-Implementación A

El video demostrativo del circuito se encuentra en anexos,

Implementación C

Montaje final del coche:

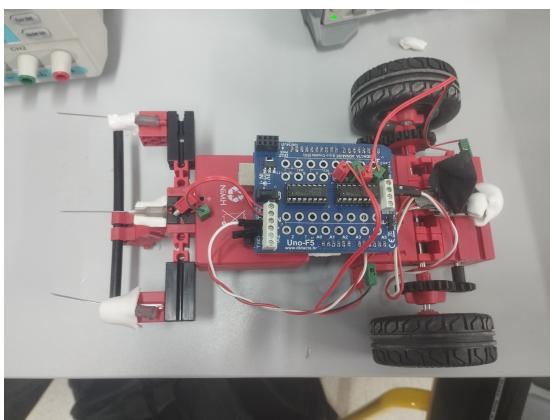


Fig. 3: Auto ya armado

CONCLUSIONES

Implementación A

El algoritmo usando para enviar una trama de 3 bytes es funcional pero no practico ya que al no haber un sistema de autenticación o de sincronia entre los bytes enviados, si existe un error en el sistema y un byte no es recibido por parte del esclavo toda la trama se ve corrida un byte lo que generaria errores a la hora de decodificar el valor del byte para la acción en la que se requiere. Para solucionar esto se podria haber usado un sistema parecido al que usa el sensor DHT11 el cual envia por el protocolo one wire una trama de datos y al final un byte llamado check sum el cual devuelve la suma de los primeros 4 bits de las tramas de humedad y de temperatura, de esta forma se podria verificar si los bytes recibidos son validos para asignar a los bytes del array.

Implementación C

El auto cuenta con 8 direcciones posibles lo cual fue posible gracias a que la app mobile utilizada contaba con las codificaciones para la suma de las 4 direcciones posibles (adelante,atras,izquierda,derecha) esto permitio establecer casos donde el PWM de una rueda era menor a de la otra probocando que el auto avanzara y a su vez rotara, aunque esto a costa de velocidad debido a que esta diminuye siempre que se gire, cosa que no pasaría si existiera un mecanismo de giro en la rueda delantera(cosa que seria imposible debido a que el kit no trae 3 ruedas de goma).

La forma de cambiar la velocidad sería bastante util si se tratara de un auto simple el cual requiera un modo de ahorro de bateria. Aunque en este caso no tiene mucha funcionalidad ya que el auto tiene una baja velocidad de por si.

BIBLIOGRAFÍA

- Electronic Wings. (s/f). Bluetooth Module HC-05 Pinout, AT Commands Arduino Programming. Recuperado el 20 de noviembre de 2024, de <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05->

ANEXOS

Repositorio de github

Evidencia implementación A