



PROYECTO FINAL

I. Objetivos.

- Desarrollar un sistema de intercambio de información entre un maestro y diferentes esclavos.
- Identificar las causas de colisión y efectos de estas en un sistema de comunicación.
- Desarrollar un protocolo de comunicación que impida o reduzca la colisión de datos del bus de comunicaciones.

II. INTRODUCCIÓN

Los buses de comunicación permiten intercambiar información entre los diferentes componentes de un circuito distribuyendo las tareas entre los diferentes dispositivos, facilitando el desarrollo y la detección de errores.

El uso de módulos de comunicación que están diseñados para ser usados bajo protocolos estándares de comunicación digital permite la rápida integración de nuevos dispositivos aun circuito, sin ser necesario modificar las partes ya existentes del circuito.

Con la capa física de la red comunicación se establecen las conexiones físicas y los niveles de voltaje usados para representar los datos. Para la gestión del tiempo, origen y destino de los datos se debe crear un protocolo de comunicación, que determine en que momento cada uno de los terminales de la red tiene acceso al bus de comunicación.

El control o ejecución de cada uno de los procesos del protocolo se puede realizar usando software o hardware. El desarrollo por software permite que el algoritmo sea modificado si el protocolo se cambia, pero se limita a las prestaciones del

periférico o al módulo de conexión usado para conectarse al bus.

El protocolo por hardware garantiza mayores velocidades de comunicación, pero se limita a una versión específica del protocolo dado que el controlador no se puede actualizar.

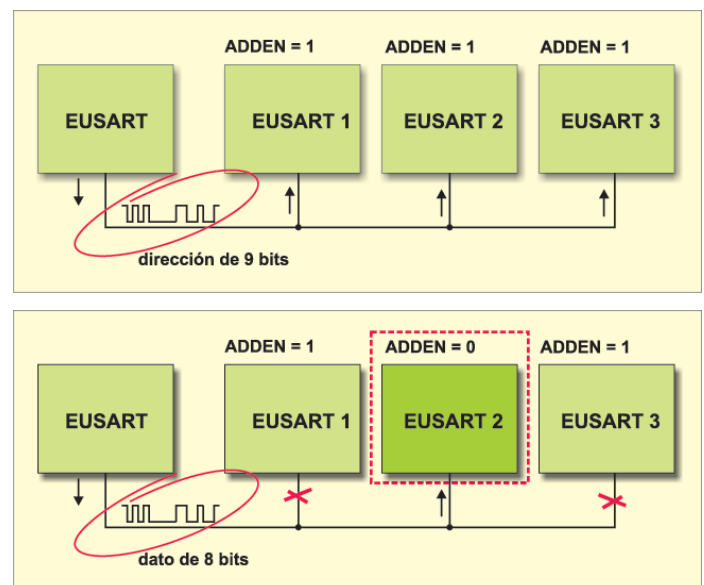


Fig 1: Red de comunicación usando bus RS-232 y protocolo por hardware.

En la Fig 1 se detalla la conexión de diferentes módulos a una red de comunicación serial usando el bus RS-232, detallando el proceso del protocolo para determinar cuál de los módulos es el que tiene acceso al bus para transmitir los datos.

Primero en la red se asigna una dirección única a cada módulo y esta se direcciona usando una longitud de 9 bits para diferenciarla de los datos a transmitir. El valor de la dirección llega a todo los módulos de la red pero solo al que le corresponde la dirección se activa y usa el bus. Este modelo no permite intercambiar información entre módulos esta primero debe llegar al controlador de la red.

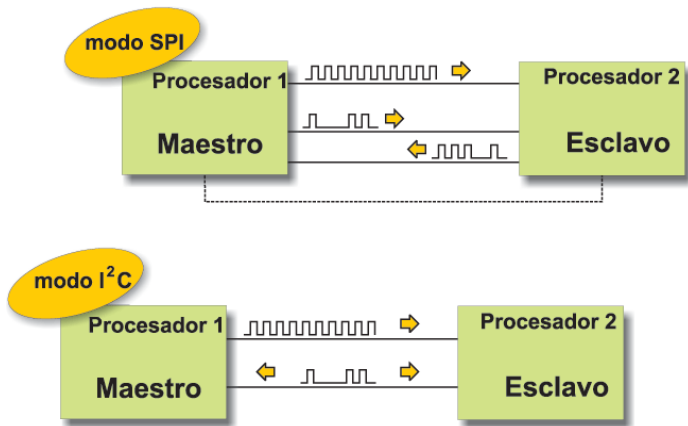


Fig 2: Comunicación SPI y I2C.

Otras interfaces de comunicación son el SPI (serial peripheral interface bus) y el I2C (inter-integrated circuit), ambos son buses de comunicación serial y tiene un protocolo para determinar cuál de los dispositivos usara el bus de comunicación.

El bus SPI usa un bit de selección para seleccionar el módulo que usara el bus. La transmisión es full-duplex, por lo cual se requiere dos hilos para transmitir los datos, otro para una señal de reloj de sincronización y uno de selección por cada dispositivo esclavo.

Para el bus I2C la comunicación es duplex, solo requiere dos hilos y la selección del módulo se realiza transmitiendo primero la dirección y después el dato. Se requiere de resistencias de pull-up para mantener en alto las líneas del bus lo que indica un estado de disponible.

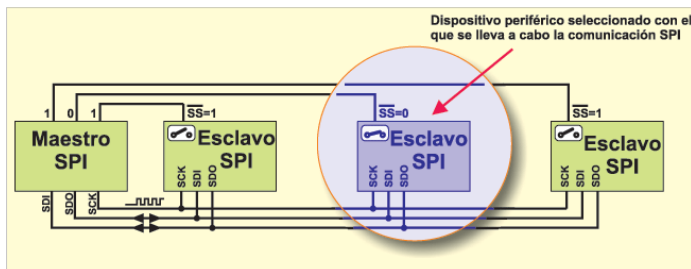


Fig 3: Red usando bus SPI.

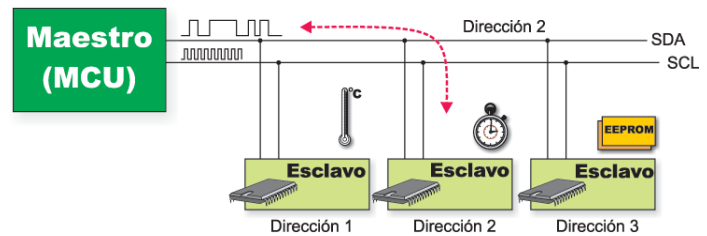


Fig 4: Red usando bus I2C.

III. DESCRIPCIÓN DE LA PRÁCTICA

En las Fig 1, Fig 3 y Fig 4, se describe la conexión diferentes dispositivos usando un protocolo de comunicación serial. Cada uno de los grupos debe implementar un sistema de comunicación con un mínimo de tres dispositivos esclavos y un maestro.

Los grupos deben desarrollar el protocolo de comunicación que usaran para evitar la colisión de los datos transmitidos por cada uno de los dispositivos.

Cada grupo tendrá un montaje diferente al igual que un sistema de visualización de los datos intercambiados entre los periféricos. La visualización tendrá un método de respaldo en una computadora usando una terminal.

IV. EVALUACIÓN.

Cada grupo debe realizar una presentación en la que explique el proceso de diseño del algoritmo de cada esclavo y el del protocolo diseñado para controlar el flujo de datos por la red.

Se considerara el funcionamiento como el 60% de la nota, el porcentaje restante corresponderá a la sustentación y presentación.



UNIVERSIDAD PEDAGOGICA Y TECNOLOGICA DE COLOMBIA
FACULTAD SEDE SOGAMOSO
ESCUELA DE INGENIERIA ELECTRONICA
LABORATORIO DE MICROCONTROLADORES



GRUPO	BUS	VISUALIZACIÓN	Esclavo 1	Esclavo 2	Esclavo 3
1	SPI	GUI	ADC, AMPLITUD, FRECUENCIA, PLOT	TERMOMETRO	VELOCIDAD Y SENTIDO GIRO DE MOTOR
2	SPI	GUI	ADC, AMPLITUD, FRECUENCIA, PLOT	ALTIMETRO	DISTANCIA POR ULTRASONIDO
3	I2C	GUI	ADC, FRECUENCIA, AMPLITUD, PLOT	DISTANCIA POR ULTRASONIDO	VISION ESTEROSCOPICA CON CONTROL DE BRILLO 7 SEGMENTOS.
4	I2C	GUI	DAC, SENO, TRIANGULAR, FRECUENCIA. SIN TABLAS Y VARIACION DE FRECUENCIA.	VELOCIDAD Y SENTIDO DE GIRO DE MOTOR	MEMORIA EEPROM, CALENDARIO UN MES.
5	RS-232	GUI	DAC, SENO, TRIANGULAR, AMPLITUD. SIN TABLAS Y VARIACION DE FRECUENCIA.	VISION ESTEROSCOPICA CON CONTROL DE BRILLO 7 SEGMENTOS	LCD, TECLADO, CLAVE DE INICIO DEL SISTEMA
6	RS-232	GUI	DAC, PRODUCTO DE SENO Y TRIANGULAR. SIN TABLAS Y VARIACION DE FRECUENCIA.	MENSAJE EN SISTEMA MATRICIAL DE 32 LINEAS	FRECUENCIOMETRO
7	SPI	GUI	DAC, PRODUCTO TRIANGULAR x TRIANGULAR CON DESFAZ DE 90. SIN TABLAS Y VARIACION DE FRECUENCIA.	CONTADOR DE CUADRATURA, SENTIDO DE GIRO Y RPM.	VELOCIDAD Y SENTIDO DE GIRO DE MOTOR.
8	I2C	GUI	ADC, PLOT, AMPLITUD, FRECUENCIA	ALTIMETRO	MEMORIA EEPROM, LISTA DE CUMPLEAÑOS (10)
9	RS-232	GUI	DAC, RESTA DE SENO MENOS TRIANGULAR, SIN TABLAS Y VARIACION DE AMPLITUD.	LCD, TECLADO, CLAVE DE INICIO DEL SISTEMA	FRECUENCIOMETRO
10	SPI	GUI	ADC, PLOT, AMPLITUD, FRECUENCIA	DISTANCIA POR ULTRASONIDO	VELOCIDAD Y SENTIDO DE GIRO DE MOTOR
11	I2C	GUI	DAC, SUMA DE SENO, COSENO Y TRIANGULAR, SIN TABLAS Y VARIACION DE AMPLITUD.	TERMOMETRO	MENSAJE EN SISTEMA MATRICIAL DE 32 LINEAS
12	RS-232	GUI	ADC, AMPLITUD, FRECUENCIA, PLOT.	FRECUENCIOMETRO	DAC, RECONSTRUCTOR DE SEÑAL DE ESCLAVO 1.