



Laboratorio de Microprocesadores - 86.07

Registrador de firmas digital

Profesor:			Ing. Guillermo Campiglio							
Cuatrimestre/Año:			1er/2017							
Turno de las clases prácticas										
Jefe de trabajos prácticos:										
Docente guía:										
Autores			Seguimiento del proyecto							
Nombre	Apellido	Padrón								
Maxim	Dorogov	95631								
Franco	Accifonte	93799								

Observaciones:

.....

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

Coloquio	
Nota final	
Firma profesor	

1. Introducción

En el siguiente informe se pretende explicar el proceso de diseño de un programa para un registrador de firma digital a través del microcontrolador Atmega 328P en lenguaje Assembly. Para ello se utilizó un display TFT de 2.4 pulgadas provisto del controlador ILI9325D con una pantalla táctil resistiva. Los datos registrados son transferidos a través de una interfaz serie/USB y procesados por Matlab. Dada la complejidad en el desarrollo del circuito impreso para el display, se optó por adquirir un Shield compatible con Arduino con ambos controladores, ILI9325D y ADC. Se utilizó la placa de desarrollo Arduino UNO provista de un microcontrolador Atmega 328P.

2. Desarrollo

Para poder interactuar en cualquier manera con el display, es necesario inicializarlo. La rutina de inicialización consta de 51 instrucciones que se envían al controlador del display mediante un bus de 8 bits. Estas instrucciones son palabras de 16 bits, por lo que deben ser enviadas en dos partes.

Una vez inicializado el display, es posible enviarle imágenes para mostrarlas en la pantalla. En esta aplicación enviamos la información pixel a pixel, estos datos se graban en la memoria ram del controlador ILI9325D para mantener en pantalla todos los pixeles a medida que se traza la firma en tiempo real hasta que el dispositivo sea reiniciado para leer una nueva firma.

2.1. Comunicación con controlador ILI9325D

El dispositivo usado para controlar la parte gráfica es el ILI9325D. Es un integrado desarrollado para pantallas graficas de hasta 232 mil colores y una resolución de 240x320 pixeles, en cuanto a la comunicación cuenta con 2 modos : SPI y bus paralelo configurable en 8, 9, 16 o 18 bits. Posee además cuatro entradas de control:

- RS : Register select
- WR: Write enable
- CS: Chip select
- RST: Reset

Para nuestra aplicación se descartó el uso de SPI debido a la velocidad de actualización requerida para registrar una firma en tiempo real, por lo cual se optó por un bus de comunicación paralelo de 8bits de datos. Si bien los comandos que el display recibe son palabras de 16bits se programó el sistema de forma tal que sea posible mandar el comando empaquetado en dos bytes. En la figura 1 se muestra un diagrama de bloques simplificado del controlador.

Para comenzar la comunicación con este periférico es necesario inicializarlo y ponerlo en un modo de espera de comandos, para ello se implementó la rutina `INIT_LCD` en la cual se inicializa el modulo y se colocan todas las entradas a valores seguros. La rutina consiste en un envío sucesivo de valores al BUS de 8 bits del display los cuales fueron grabados en la memoria de programa del microprocesador y accedidos a través de un puntero

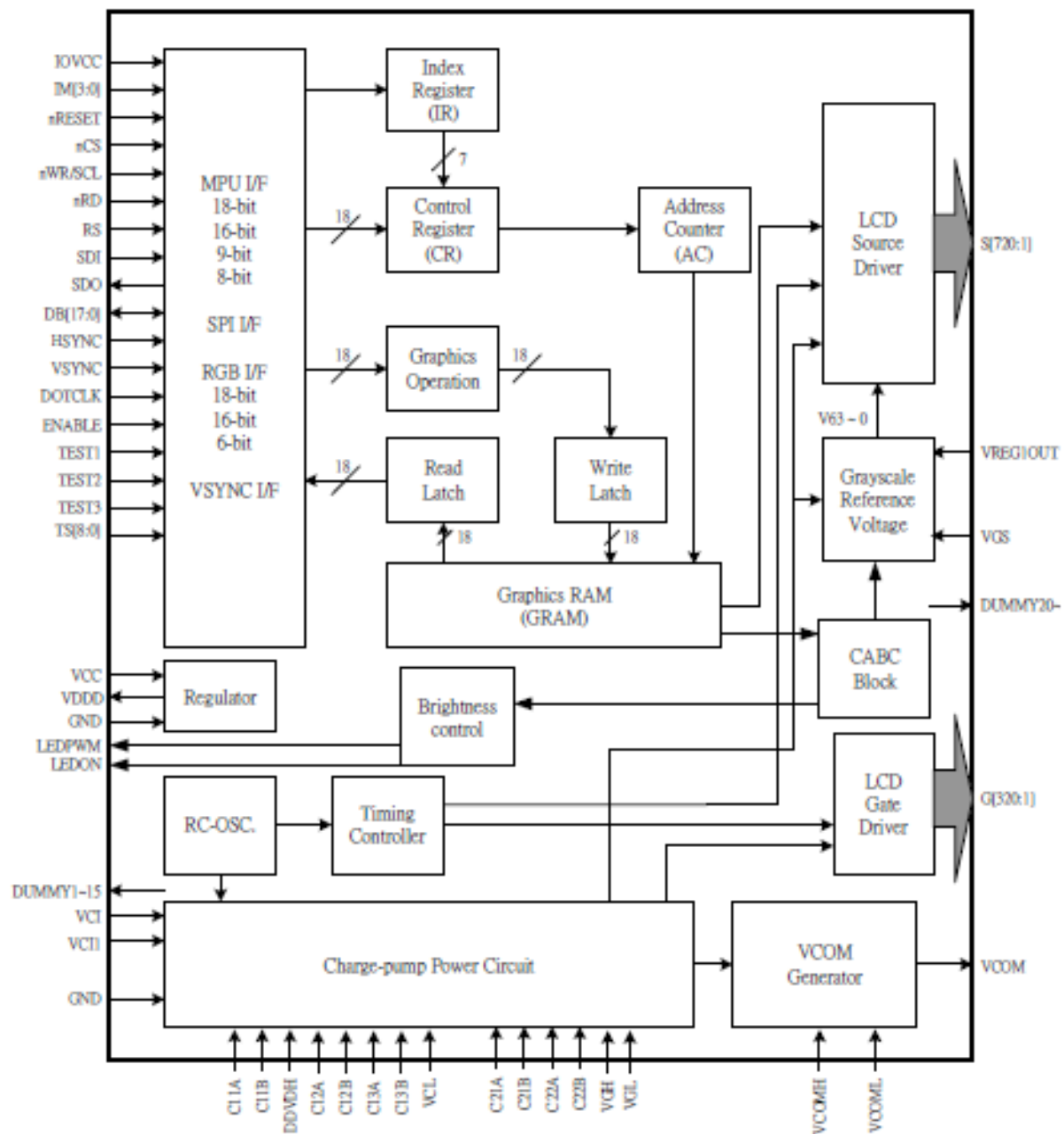


Figura 1: Diagrama en bloques simplificado del ILI9325D

Una vez inicializado el modulo es necesario configurar el área de pantalla sobre la cual se va a trabajar, para ello se implemento la macro `SET_XY_AREA`. Esta macro es llamada por todas las demás macros que utiliza el microprocesador para controlar la pantalla. Luego se llena la pantalla del color deseado con `FILL_SCREEN` y se procede a enviar píxel por píxel las coordenadas X e Y recibidas del sensor táctil a la macro `DRAW_PIXEL`. Esta ultima se encarga de grabar en la memoria ram del controlador las coordenadas a visualizar y envía los comandos necesarios para mostrar las coordenadas anteriores junto con la ultima en pantalla. Se prefirió el uso de macros en lugar de rutinas debido a la alta velocidad de respuesta requerida para la aplicación y además para no atentar contra la portabilidad del sistema. Todas las macros implementadas relacionadas con el manejo de la pantalla se encuentran en el archivo `tft_macros.mac`.

2.2. Comunicación con ADC

Para censar las coordenadas del píxel presionado, es necesario comunicarse con otra parte del display táctil: el conversor analógico-digital `ADS7843`. Este sistema tiene su propio protocolo de comunicación, con otro juego de ordenes. Las entradas de control disponibles son las siguientes:

- TCLK : Clock interno del ADC
- TCS: Chip select del ADC
- TDIN: Bit de entrada de datos al ADC
- TDOUT: Bit de salida de datos del ADC

El sensor táctil consiste básicamente de un divisor resistivo doble, como se ve en la figura 2. Cuando se le pide al controlador censar en la coordenada X, este conecta el terminal X+ a una tensión positiva, el terminal X- a una tensión negativa, y el terminal Y+ a la entrada del conversor.

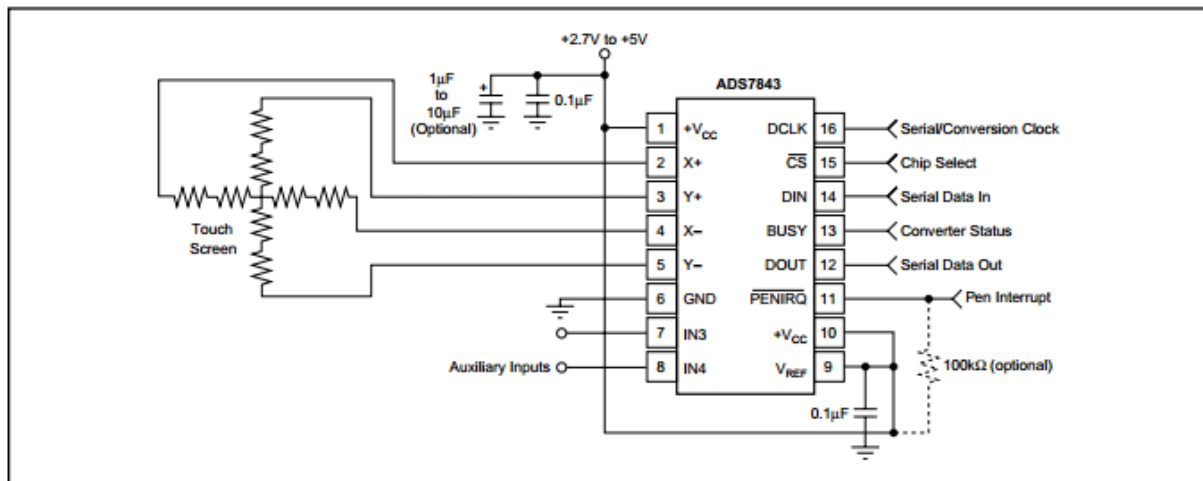


Figura 2: Operación básica del ADS7843

Para especificar cual coordenada se leerá, se envía por comunicación serie la palabra 0xD0 para leer la coordenada X, o la palabra 0x90 para leer la coordenada Y.

Una vez finalizado el censado el controlador devuelve, nuevamente por comunicación serie, una palabra de 12 bits variando entre 0x000 y 0xFFF, la cual es separada en parte alta y parte baja y almacenada en dos registros. Este registro es luego convertido en la coordenada correspondiente del pixel seleccionado.

Se muestra en la imagen 3 el tren de pulsos que se envía para iniciar la adquisición, y el tren de pulsos que se recibe tras el censado.

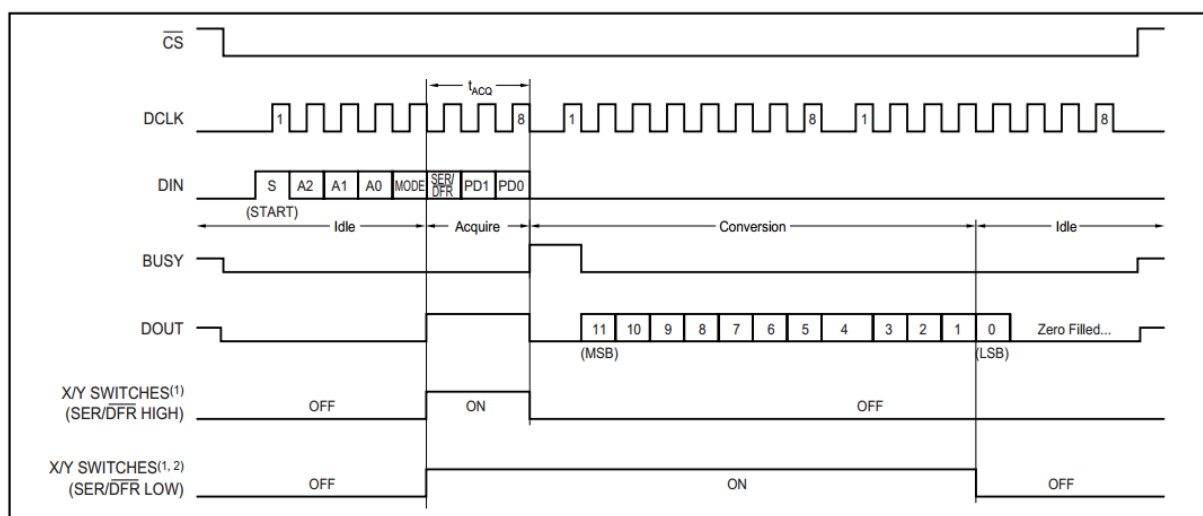


Figura 3: Comunicación con ADC

2.3. Conversión de datos

A causa de imperfecciones en el film táctil, cuando se toca con el lápiz cerca de los extremos el conversor no devuelve 0x000 o 0xFFFF, sino que devuelve valores cercanos. Es necesario tener en cuenta estos valores al hacer la conversión, porque de lo contrario se obtendrá un error importante cuando se grafique el pixel en la pantalla.

Para realizar la conversión, se implementaron las siguientes ecuaciones:

$$Y_{pix} = \frac{Y_{size}}{Y_{top} - Y_{bottom}}(Y_{ADC} - Y_{top})$$
$$X_{pix} = \frac{X_{size}}{X_{right} - X_{left}}(X_{ADC} - X_{left})$$

Donde X_{right} , X_{left} , Y_{top} e Y_{bottom} son constantes de calibración, y Y_{ADC} y X_{ADC} es la lectura entregada por el ADC.

Para implementar esto, se halló el valor de las constantes en base 16 y se programó un esquema de producto por punto fijo.

2.4. Envío de datos por puerto serie y recepción por Matlab

Mientras la firma es graficada en la pantalla táctil los datos son transferidos a la computadora para ser procesados y almacenados en una base datos, el programa de adquisición se desarrolló en Matlab. Su función es la recibir, validar y almacenar los datos provenientes del microcontrolador, graficarlos y a su vez darle la orden al microcontrolador para comenzar o finalizar la adquisición de la firma. La adquisición se hace a través del protocolo serie, vía USB.

Se configuró la USART del microcontrolador para transmitir datos de 8 bits de largo, con un bit de stop y un bit de inicio a una velocidad de 38400 baudios. Para ello se implementó la rutina la inicialización `INIT_USART`, y `DISABLE_USART` para interrumpir la comunicación. Para transmitir el dato se implementó la macro `DATA_TX` disponible en el archivo `IO.mac`.

Dado que el modulo utiliza la totalidad de pines del microcontrolador se tuvo que implementar un algoritmo de transmisión que no genere incompatibilidades entre los comandos enviados por el puerto hacia el software de adquisición y lo que puede interpretar el display. Para ello se activa la USART únicamente al momento de transmitir y luego se desactiva antes de pasar al estado de sensado. Dado que la memoria ram de un atmega328P es insuficiente para almacenar los píxeles de una firma se optó por realizar la transmisión pixel a pixel en tiempo real en paralelo al sensado. A continuación se ilustra una foto con un dibujo en la pantalla táctil y la imagen generada por el software de adquisición.



Figura 4: Foto de un dibujo realizado sobre la pantalla táctil



Figura 5: Imagen recibida por el software de adquisición

2.5. Hardware utilizado

El modulo de display a utilizar acepta niveles lógicos de hasta 3.3 V, lo cual es un problema al momento de poner en practica el diseño. Para ello se hizo uso de un modulo que adapta las tensiones de salida de los puertos de la placa de desarrollo Arduino al nivel soportado por el display. El dispositivo adapta las tensiones de salida de 5 V provenientes de los puertos de Atmega 328p a logica de 3.3 V y realiza lo mismo con los datos provenientes del display hacia el microcontrolador mediante cuatro integrados 74HC541PW.

Se decidió usar una plataforma de desarrollo arduino dada la compatibilidad de diversos módulos o "Shields" disponibles en el mercado para esta plataforma. Para el proyecto se usó un Arduino UNO. A continuación se muestran las imágenes correspondientes al mapa de puertos del Arduino UNO, la conexión realizada entre este y el display y un diagrama esquemático del modulo adaptador de tensiones.

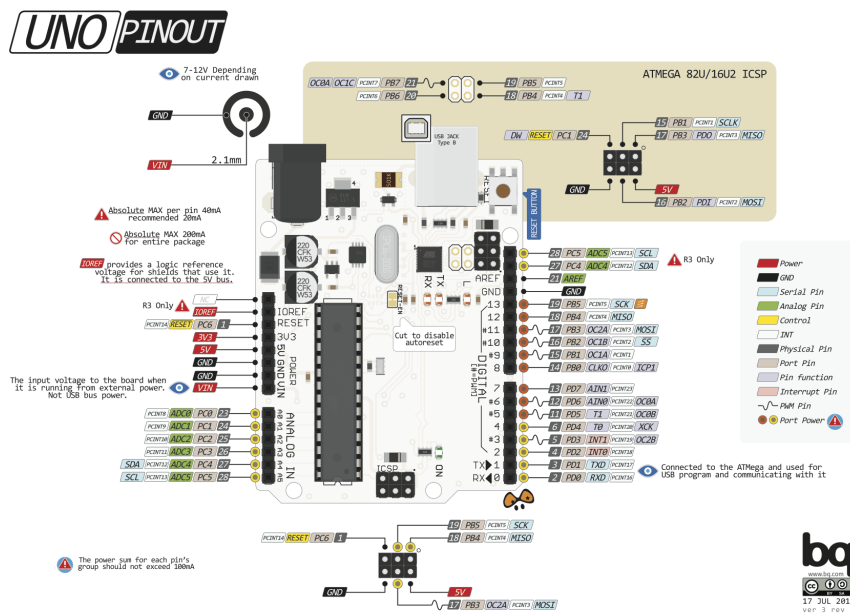


Figura 6: Mapa de puertos de un Arduino UNO.

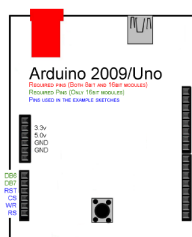


Figura 7: Conexión realizada entre el Arduino y el display

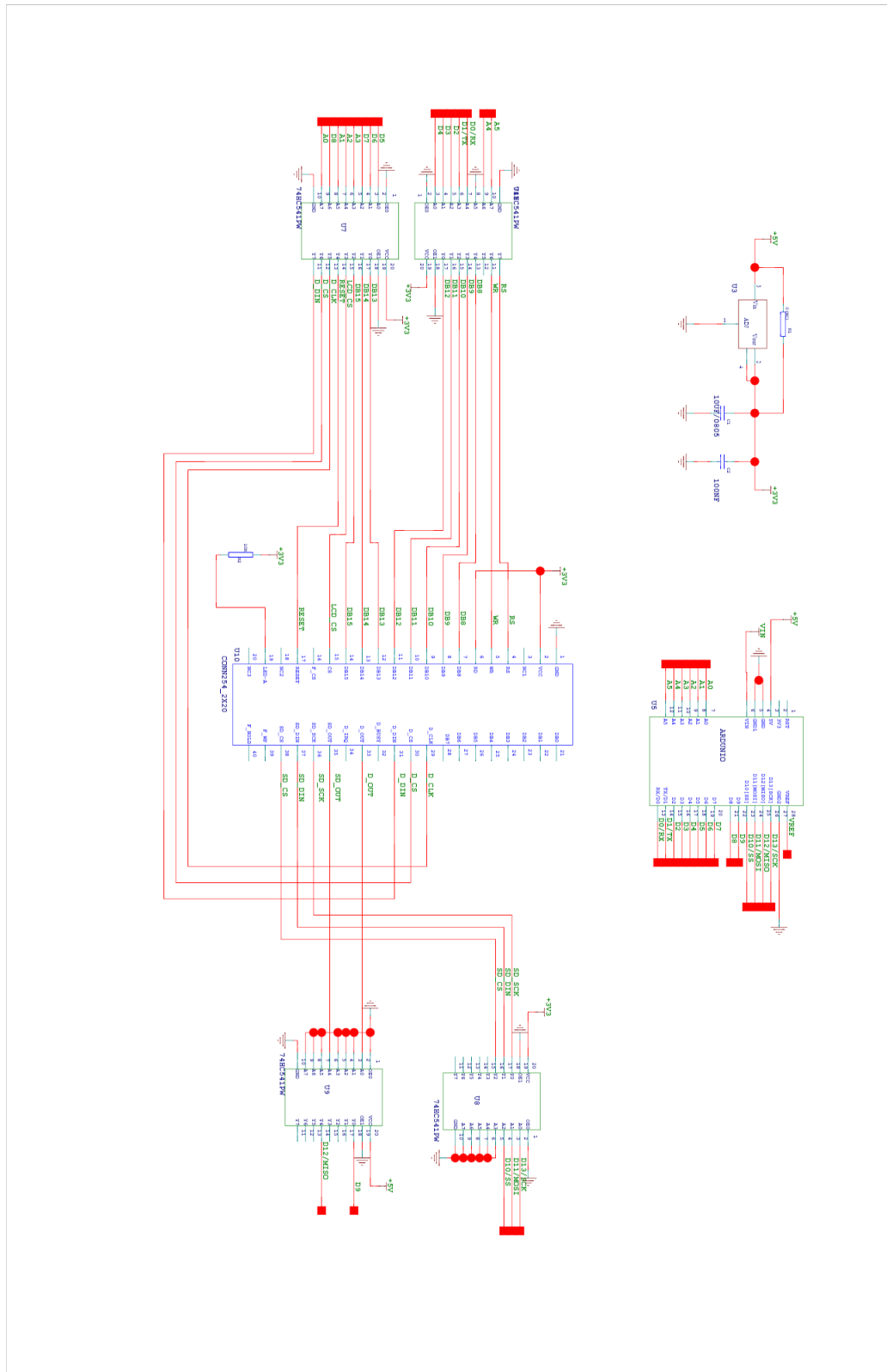


Figura 8: Diagrama esquemático del adaptador de niveles lógicos

3. Conclusiones

Se logro implementar con éxito el modulo de registro y el software de adquisición para procesamiento de las firmas. Algunas de las mejoras para implementar a futuro: optimizar la transmisión serie para

reducir el delay generado entre que se realiza la firma y su visualización en pantalla, transmitir los datos en forma inalámbrica por medio de bluetooth para mayor comodidad y portabilidad del dispositivo, migrar a una plataforma de 32 bits para manejar pantallas mas grandes con mayor resolución con el fin de desarrollar una tableta gráfica con prestaciones comerciales.