



ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

LABORATORIO DE MICROPROCESADORES Y ENSAMBLADOR

GUÍA NO. 2

TABLA DE CONTENIDOS

1. INTRODUCCIÓN.....	1
2. OBJETIVOS	1
3. EL LENGUAJE DE PROGRAMACIÓN C.....	1
4. TARJETA DE DESARROLLO STK600	2
5. MANEJO DE PUERTOS,ENTRADAS-SALIDAS	2
6. PANTALLA LCD	3
7. TECLADO MATRICIAL	4
8. MATERIALES PARA LA PRÁCTICA	5
9. PROCEDIMIENTO	5
10. INFORME.....	8
11. REFERENCIAS	8

1. INTRODUCCIÓN

La práctica 2 del Laboratorio de Microprocesadores y Ensamblador, introduce los conceptos básicos de programación en lenguaje C para microcontroladores, haciendo uso de la tarjeta de desarrollo STK600, que está basada en un microcontrolador Atmega2560. En esta práctica se abordarán conceptos que permitirán al estudiante comprender el manejo básico de puertos y periféricos comúnmente utilizados como la pantalla LCD y el teclado matricial.

2. OBJETIVOS

- Familiarizarse con la programación de microcontroladores en el lenguaje C.
- Desarrollar programas simples para microcontroladores AVR, haciendo uso de la tarjeta de desarrollo STK600 y el software de simulación Proteus.
- Manejar los puertos de entrada y salida del microcontrolador por medio de la programación en C.
- Familiarizarse con el uso de periféricos como las pantallas LCD y los teclados matriciales.

3. EL LENGUAJE DE PROGRAMACIÓN C

C es un lenguaje de programación de alto nivel, que a diferencia de otros lenguajes, permite al programador tener acceso a los bytes del procesador. En cierto modo, C puede considerarse como una clase de lenguaje ensamblador universal.

En los dispositivos microcontrolados, el lenguaje C permite una interacción con el dispositivo mucho más amigable que el lenguaje ensamblador, haciendo que los programas sean fácilmente sostenibles, legibles y expandibles. Estas características hacen de C el lenguaje más usado para el desarrollo de aplicaciones en sistemas microcontrolados.

Los compiladores existentes permiten una interacción con el dispositivo a través de librerías que facilitan el proceso de desarrollo, sin embargo, al igual que en el lenguaje ensamblador es necesario tener un conocimiento del microcontrolador y sus registros para lograr un adecuado funcionamiento.

4. TARJETA DE DESARROLLO STK600

La tarjeta STK600 es un kit de desarrollo básico que brinda al programador la posibilidad de descargar programas sobre diferentes referencias de microcontroladores. En esencia la STK600 es una tarjeta con ruteado genérico, por lo cual puede adaptarse a un amplio número de dispositivos microcontrolados de la familia AVR. Además, cuenta con hardware específico, como periféricos de comunicación serial, USB, pulsadores, leds, entre otros.

La tarjeta de desarrollo STK600 es soportada por AVR Studio, desde donde es posible programar y configurar diferentes parámetros de operación del dispositivo microcontrolado.

Las tarjetas utilizadas para el laboratorio de microprocesadores y ensamblador cuentan con un procesador ATmega2560, el cual tiene las siguientes características:

- Arquitectura RISC.
- 84 pines de I/O, 11 Puertos.
- 32 registros (8 bits) de propósito general, conectados directamente a la Unidad Lógica Aritmética (ALU).
- 256K bytes de memoria flash, 4K bytes EEPROM, 2 Timers/Contadores de 8 bits, 4 Timers/Contadores de 16 bits, 16 canales de 10 bits de ADC, 4 USART, 8 Canales de PWM, interrupciones internas y externas, Interfaz para manejo de memorias externas, Interface Serial, SPI, 2- wire, entre otros.

5. MANEJO DE PUERTOS, ENTRADAS-SALIDAS

El ATmega2560 dispone de 84 pines de I/O, distribuidos en 11 Puertos. El control y configuración de los pines de entrada/salida es realizado mediante los registros DDRx, PORTx y PINx,

El registro DDRx, permite la configuración de la dirección con la que va a operar cada pin. Un cero lógico configura el pin como salida y un uno lógico configura el pin como entrada. Cada puerto del microcontrolador tiene asociado un registro único para la configuración de dirección, por lo cual cada pin del puerto puede ser configurado de forma independiente.

Los datos a escribir sobre un puerto son enviados hacia PORTx y los datos a leer de un puerto son capturados mediante PINx. Si PORTx es Igual a 1 lógico, cuando el pin ha sido configurado como entrada la resistencia de Pull-Up asociada es activada. En la figura 1 se muestra el diagrama esquemático que representa el funcionamiento de un pin de entrada/salida del microcontrolador.

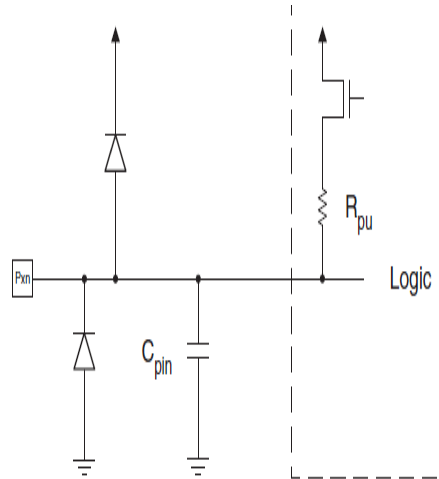


Figura 1. Esquema Equivalente a un pin I/O

6. PANTALLA LCD

La pantalla de cristal líquido o LCD (Liquid Crystal Display) es un dispositivo microcontrolado de visualización gráfico para la presentación de caracteres, símbolos o incluso dibujos (en algunos modelos).

En esta práctica se utiliza un LCD de 2 filas de 16 caracteres conectado al puerto C microcontrolador a través de un bus de cuatro bits y tres señales de control (enable, read, write). La conexión utilizada se presenta en la figura 2.

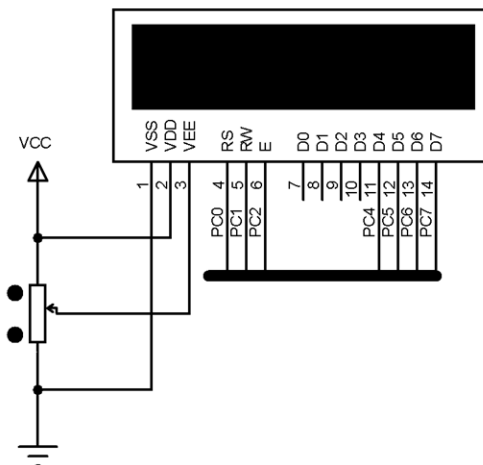


Figura 2. Conexión de una Pantalla LCD al Puerto C del Microcontrolador

La comunicación con la pantalla LCD, se realiza por medio de la librería "*lcd.h*" compuesta principalmente por las siguientes funciones:

void lcd_init(LCD_DISP_ON): Inicializa la pantalla LCD y prepara la conexión con el puerto C.

void lcd_clrscr(void): Limpia la pantalla LCD.

void lcd_home(void): Envía el cursor a la posición (0,0).

void lcd_gotoxy(uint8_t x, uint8_t y): Envía el cursor a la posición x,y deseada.

void lcd_putc(char c): Imprime en la posición actual del cursor un carácter tipo char.

*void lcd_puts(const char *s):* Imprime a partir de la posición actual del cursor una cadena de caracteres.

7. TECLADO MATRICIAL

Un teclado matricial es un arreglo de botones dispuestos de manera organizada en filas y columnas. Comercialmente se consiguen de 3x4 y 4x4 (Columnas x Filas).

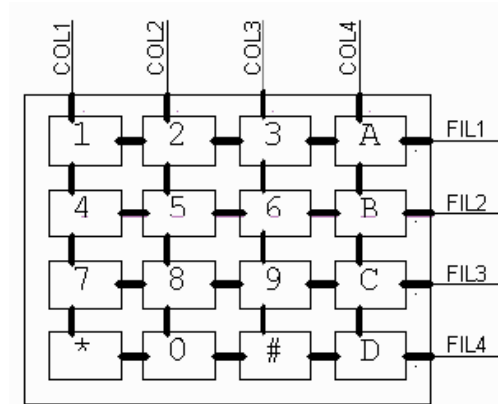


Figura 3. Filas vs Columnas de un Teclado Matricial

Debido a su facilidad de uso, son ampliamente utilizados para ingreso de datos. Aunque existen circuitos decodificadores para este tipo de teclados, NO es indispensable utilizarlos debido a la fácil implementación de rutinas tanto de *polling* como por *interrupción* para leer sus valores.

En esta práctica el teclado matricial será utilizado mediante una rutina de *polling* de la siguiente manera: Por cada fila y cada columna se tiene un pin a disposición. Cada botón cierra el circuito entre una fila y una columna. Por ejemplo, si se presiona el botón 1, la fila 1 y la columna 1 quedan en corto. Si se presiona el botón 7, la columna 1 y la fila 3 quedan en corto. Por lo tanto, se utilizan los pines de columna generalmente como **salidas**, mientras que los de fila se utilizan como **entradas** (activando previamente las resistencias de *pull-up*). De esta manera para leer un valor del teclado matricial es necesario realizar un bucla constante de envío de datos a los puertos de salida, con lo cual se busca activar con un cero lógico cada una de las columnas del teclado. En el momento en que se activa cada columna, se realiza la lectura de los cuatro pines de entrada (filas); si uno de estos está en corto con la columna activada (si la lectura da un cero lógico), significa que el botón en la intercepción entre la fila y columna en corto ha sido presionado.

La rutina descrita anteriormente es implementada en la librería "*mat_kbrd.h*", compuesta por dos funciones: "*void kbrd_init(void)*" que inicializa el puerto L para la conexión del teclado y "*char kbrd_read(void)*" que realiza la lectura de del dígito presionado.

En la figura 4 se muestra un teclado matricial 4x4. Los pines de conexión ubicados en la parte bajo del diagrama están asociados a las columnas (pines 0-3) y a las filas (pines 4-7) del teclado matricial.



Figura 4. Teclado matricial 4x4

8. **MATERIALES PARA LA PRÁCTICA**

Para el desarrollo de la práctica es necesario contar con los siguientes elementos:

- Tarjeta de Desarrollo STK600
- Teclado Matricial 4X4
- LCD 2X16
- Protoboard, cable, resistencias, pinzas, etc.

9. **PROCEDIMIENTO**

- Configuración y programación de la AVR STK600

Descargue el código del Ejemplo 1 sobre la tarjeta de desarrollo AVR-STK600. Conecte los pines PC4, PC5, PD0 y PD1, a leds y verifique el funcionamiento. Cambie la frecuencia de encendido de los leds.

```
#define F_CPU 16000000UL           //freq 16 MHz
#include <avr/io.h>
#include <util/delay.h>
#include "FreeRTOS.h"
#include "task.h"

#include "queue.h"

#include "mytasks.h"
#include "lcd.h"
#include "ADC_routines.h"
#include "UART_routines.h"

//TAREA QUE COLOCA A PARPADEAR EL PUERTO C
void vLEDFlashTask( void *pvParameters )
{
    DDRC=0xff;
    for( ;; )
    {
        PORTC^=0xF0;
        vTaskDelay(50);
    }
}
```



```
        PORTC^=0xF0;
        vTaskDelay(50);
    }
}

void vLEDFlashTask2( void *pvParameters )
{
    DDRD=0xFF;
    for( ;; )
    {
        int a =50;
        PORTD^=0x0F;
        vTaskDelay(a);
        PORTD^=0x0F;
        vTaskDelay(a);
    }
}

//TAREA QUE ENVIA UNA CADENA DE CARACTERES POR LA USART
void vUSARTdriverTask( void *pvParameters )
{
    //Inicialización USART
    usart_init();
    portTickType xLastWakeTime;
    const portTickType xFrequency = 50;
    xLastWakeTime=xTaskGetTickCount();
    for( ;; )
    {
        //Envío de una Cadena por la USART
        transmitString("FreeRTOS Atmega2560 - 2018",26);
        vTaskDelayUntil(&xLastWakeTime,xFrequency);
    }
}

void myADC( void *pvParameters )
{
    char BUF[16]; //Variable para almacenar cadena de caracteres
    ADC_init();
    int i=0;
    float volt= readVoltage(0x00);
    lcd_init(LCD_DISP_ON);
    //limpiar la pantalla (clear screen)
    lcd_clrscr();
    //enviar el puntero a la posicion inicial
    lcd_home();
    //Ingresar buffer
    sprintf(BUF, "%2.2f", volt); //Almacena la cadena de caracteres en un arreglo Char
    lcd_puts("  LAB. MICRO  ");
    lcd_gotoxy(0,1);
    lcd_puts(BUF);
}
```

Modifique el código anterior de tal forma que, al presionar dos pulsadores, se enciendan los leds conectados a los pines PD0 y PD1 respectivamente.



- Calculadora básica

Descargue el código del Ejemplo 2 sobre la tarjeta AVR-STK600. Realice el cableado necesario para la conexión del teclado matricial y la pantalla LCD con la tarjeta de desarrollo.

```
#define F_CPU 16000000UL           //freq 16 MHz
#include <avr/io.h>
#include <util/delay.h>
#include "FreeRTOS.h"
#include "task.h"

#include "queue.h"

#include "mytasks.h"
#include "lcd.h"
#include "ADC_routines.h"
#include "UART_routines.h"

//TAREA QUE COLOCA A PARPADEAR EL PUERTO C
void vLEDFlashTask( void *pvParameters )
{
    DDRC=0xff;
    for( ;; )
    {
        PORTC^=0xF0;
        vTaskDelay(50);
        PORTC^=0xF0;
        vTaskDelay(50);
    }
}

void vTeclado( void *pvParameters )
{
    //VARIABLE PARA ALMACENAR EL CARACTER PRESIONADO
    char caracter;
    lcd_init(LCD_DISP_ON);
    //TECLADO EN EL PUERTO C
    kbrd_init();
    //LIMPIAR LA PANTALLA (CLEAR SCREEN)
    //ENVIAR EL CURSOR A LA POSICION INICIAL
    lcd_home();
    lcd_puts("  LAB. MICRO  ");
    char BUF[16]; //VARIABLE PARA ALMACENAR CADENA DE CARACTERES
    int a =10;
    caracter=0;

    for( ;; )
    {
        caracter = kbrd_read();

        if(caracter != 0){

            //LIMPIAR LA PANTALLA (CLEAR SCREEN)
            lcd_clrscr();
```



```
//ENVIAR EL CURSOR A LA POSICIÓN INICIAL
lcd_home();

//sprintf(BUF,"    NUM:%c",caracter); //ALMACENA LA CADENA DE
CARACTERES EN UN ARREGLO CHAR
//lcd_puts(BUF);//IMPRIME LA CADENA DE CARACTERES EN LA PANTALLA

lcd_gotoxy(0,1),//MUEVE EL CURSOR A LA POSICIÓN X=0,Y=1
lcd_putc(caracter);//IMPRIME EL CARACTER
}

vTaskDelay(a);
}

}
```

Tomando como base el código anterior, realice una aplicación que permita realizar las operaciones básicas (suma, resta, multiplicación y división) sobre dos números decimales de hasta tres cifras con dos dígitos de precisión. El programa deberá tener las siguientes características: La entrada de los números, los signos de operación, el punto y el igual, será tomada de un teclado matricial 4X4. Cada dígito ingresado será mostrado en forma de eco en una pantalla LCD. El resultado de la operación deberá ser impreso en la pantalla, junto con la cadena de caracteres operada.

Ejemplo: Por medio del teclado se introduce los números 123 y 5.67. Estos dos dígitos son operados mediante una suma que genera como resultado el número 128.67. En la pantalla LCD se deberá mostrar de la siguiente manera:

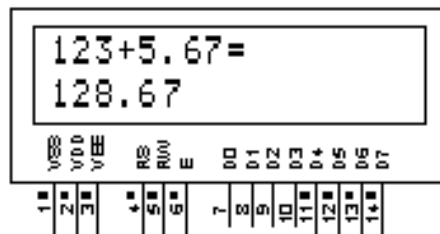


Figura 2. Salida en Pantalla LCD

10. INFORME

Se deberá presentar un informe escrito en formato IEEE de **no más de dos páginas**, en las cuales se presenten los siguientes ítems: Introducción, Desarrollo de cada uno de los problemas propuestos (descripción de puertos, registros y/o estrategias utilizadas en el desarrollo), Resultados Obtenidos y Conclusiones. No es necesario incluir códigos, sin embargo estos, junto con el informe, deberán ser presentados y explicados al momento de la sustentación, al igual que los montajes y/o simulaciones funcionando.

11. REFERENCIAS

STK600 User Guide:

<http://support.atmel.no/knowledgebase/avr32studiohelp/com.atmel.avr32.tool.stk600/html/stk600.html>

Hoja de Datos ATMEGA2560:

<http://www.atmel.com/Images/doc2549.pdf>

Notas De Clase, Laboratorio de Microprocesadores y Ensamblador. Universidad del Valle, 2013.