24-4-2021

ADRIAN OROZCO ANZURES 18300328  
CARLOS DANIEL LOZANO VAZQUEZ 18300249

SiSTEMAS EMBEBIDOS I

CETI COLOMOS

**Menú Teclado LCD**

Actividad 14 – Práctica 8

Logotipo

Descripción generada automáticamente

**Objetivo**

Utilizar los puertos de un Arduino y conectarle un teclado matricial para entrada de datos y una LCD para mostrar resultados y comprobar un programa de menú de opciones.

**Descripción**

Utilizar Arduino MEGA y conectarle un teclado matricial de 9, 12 o 16 teclas a uno de sus puertos para introducir datos y conectar una LCD (de 16 x 2) a otro puerto para mostrar resultados.

Realizar un programa para Arduino que funcione como un menú de 6 opciones, el programa debe pedir 3 números x, y, z de 4 o de 8 bits y mostrar la información en la LCD.

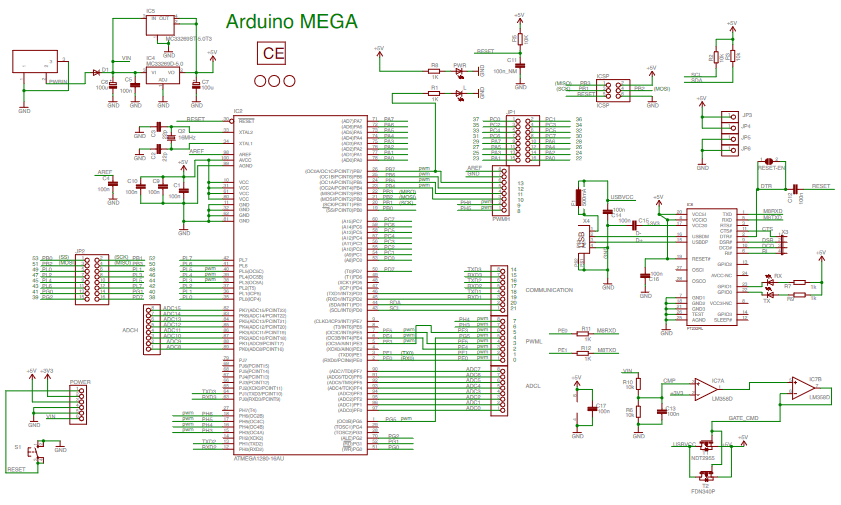
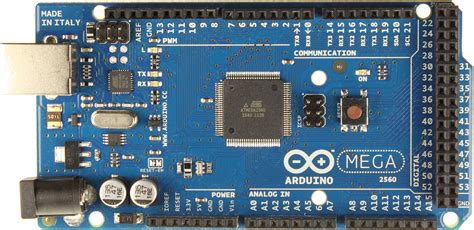
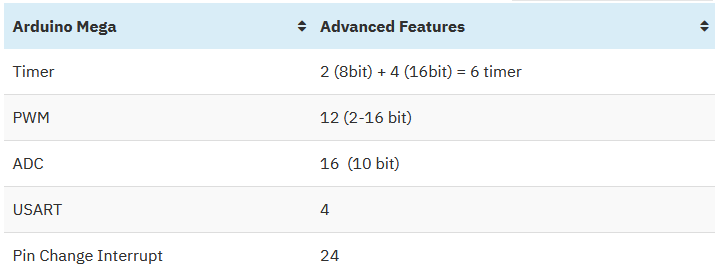
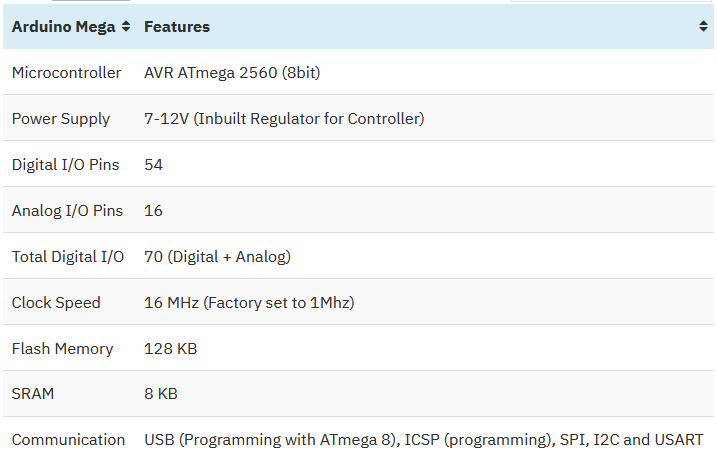
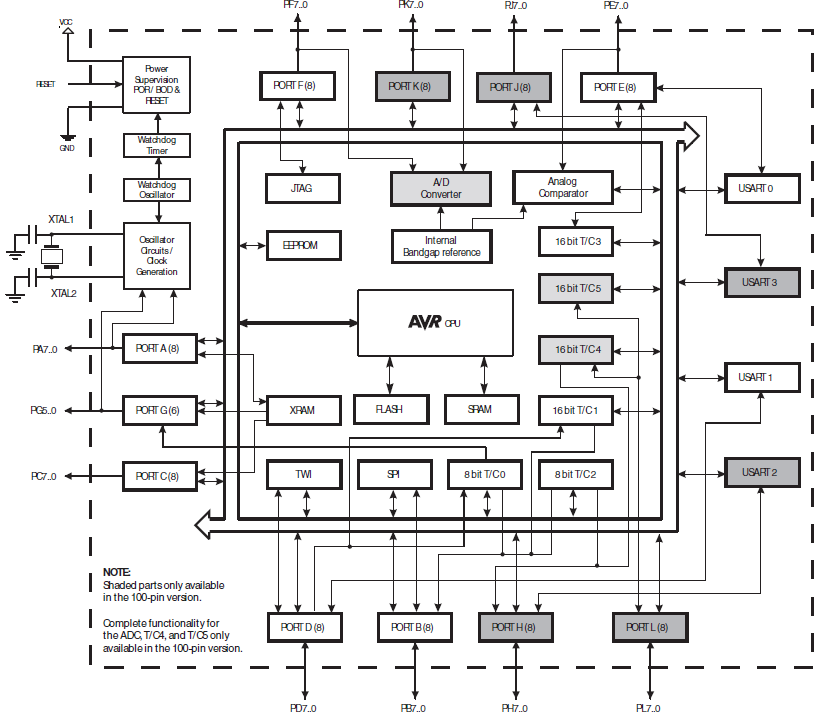
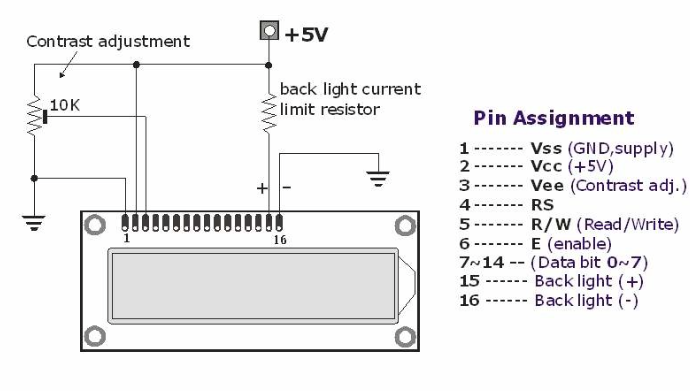
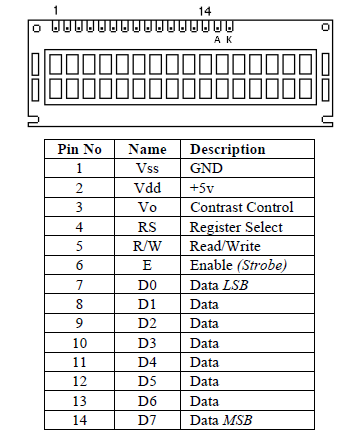
1. Si x=0 mostrar y + z.
2. Si x=1 mostrar y - z.
3. Si x=2 mostrar y \* z.
4. Si x=3 mostrar y / z.
5. Si x=4 mostrar los números de “y” hasta 99 separados por un lapso de 0.5 segundos.
6. Si x=5 mostrar el mayor de y, z.

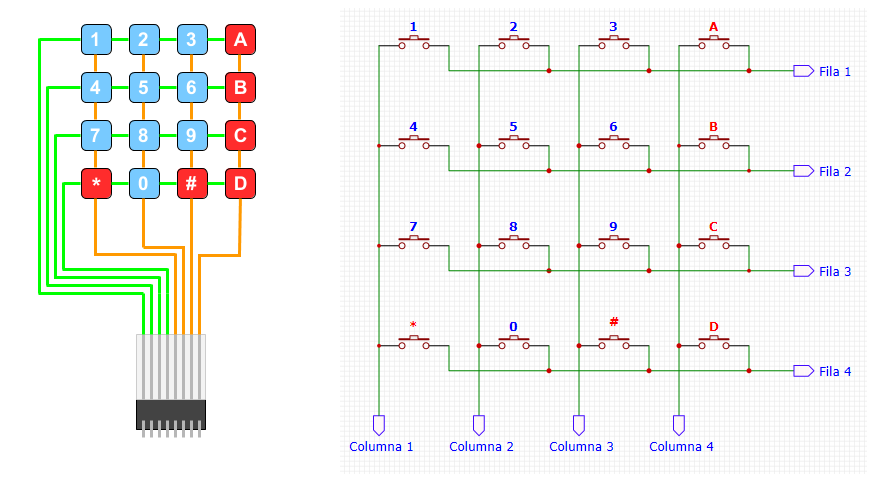
**Resumen**

Usando una tarjeta de Arduino MEGA, un teclado matricial hexadecimal 4x4 y un LCD de 16x2, se debe crear un programa que, mediante el teclado, lea un número el cual represente la opción que se va a ejecutar. Para cada caso se deben de ingresa dos datos mediante el teclado (estos pueden ser de cualquier tamaño), a excepción del 4to caso, el cual solo se ingresa un dato.

El programa es creado en un editor de texto, ya sea Visual Studio Code, Sublime Text, Atom, o hasta el mismo Arduino IDE; pero para utilizar el código este debe ser verificado en el IDE de Arduino.

**Diagrama Eléctrico**

********



**Circuito**

**Diagrama

Descripción generada automáticamente**

**Código**

#include <Keypad.h>

#include <LiquidCrystal.h>

*// Mapa del teclado 4x4*

const char KEYS[4][4] = {

    { '1', '2', '3', 'A' },

    { '4', '5', '6', 'B' },

    { '7', '8', '9', 'C' },

    { '\*', '0', '#', 'D' }

};

const byte ROWS[4] = { 29, 28, 27, 26 }, COLUMNS[4] = { 25, 24, 23, 22 };

Keypad keypad(makeKeymap(KEYS), ROWS, COLUMNS, 4, 4);

LiquidCrystal lcd(53, 52, 51, 50, 49, 48);

*// La función render sirve para imprimir un valor o string*

*// en X y Y, asi como resetear la LCD*

void render (float *val*, int *x* = 0, int *y* = 0, bool *clear* = false);

void render (const char\* *str*, int *x* = 0, int *y* = 0, bool *clear* = false);

float read ();

void setup () {

    lcd.begin(16, 2);

    lcd.setCursor(0, 0);

}

void loop () {

    lcd.clear();

    int opc = 0;

    float\* arr = nullptr;

*// Mostrar el menú*

    render("\* MENU \*", 4, 0, true);

    render("[0] Sum  [1] Dif", 0, 1);

    delay(1500);

    render("[2] Pro  [3] Div", 0, 1);

    delay(1500);

    render("[4] Y99  [5] Max", 0, 1);

    delay(1500);

    delay(200);

    render("OPCION: ", 0, 0, true);

    opc = read();*// Leer la opción*

    render(opc, 8);

*// Condición para saber de que tamaño crear el arreglo*

    if (opc == 4) {

        arr = new float[1];*// Un solo número*

        render("Numero Y: ", 0, 1);

        arr[0] = read();

        delay(1000);

    }

    else if (opc >= 0 && opc <= 5) {

        arr = new float[2];*// Dos números*

        render("Numero 1: ", 0, 1);

        arr[0] = read();

        delay(200);

        render("                ", 0, 1);

        render("Numero 2: ", 0, 1);

        arr[1] = read();

        delay(200);

    }

    render("Resultado: ", 0, 0, true);

    switch (opc) {

        case 0:

            lcd.print(arr[0] + arr[1]);*// Suma*

            break;

        case 1:

            if ((arr[0] - arr[1]) < 0)

                lcd.print("-");*// Signo negativo*

            lcd.print(((arr[0] - arr[1]) >= 0 ? (arr[0] - arr[1]) : (arr[1] - arr[0]) ));*// Resta*

            break;

        case 2:

            lcd.print(arr[0] \* arr[1]);*// Multiplícación*

            break;

        case 3:

            lcd.print(arr[0] / arr[1]);*// División*

            break;

        case 4:

            for (unsigned i = (int)arr[0]; i <= 99; i++) {

                render(i, 11);*// Y a 99*

                delay(500);

            }

            break;

        case 5:

            lcd.print(max(arr[0], arr[1]));*// Mayor de 2 números*

            break;

        default:

            render("\* ERROR \*", 4, 0, true);*// Caso invalido*

            delay(1500);

            return;

    }

    char key = '\0';

*// Repetir bucle hasta que se presione '#'*

*// Usado para apreciar el resultado*

    while (key != '\*')

        key = keypad.getKey();

}

void render (const char\* *str*, int *x*, int *y*, bool *clear*) {

    if (*clear*)

        lcd.clear();

    lcd.setCursor(*x*, *y*);

    lcd.print(*str*);

    delay(200);

    return;

}

void render (float *val*, int *x*, int *y*, bool *clear*) {

    if (*clear*)

        lcd.clear();

    lcd.setCursor(*x*, *y*);

    lcd.print(*val*);

    delay(200);

    return;

}

float read () {

    float val = 0;

    char key = 0;

*// Repetir hasta que se presione '#'*

    while (key != '#') {

        delay(5);

        key = keypad.getKey();

*// Verificar si la tecla presionada es un número*

        if (key >= 48 && key <= 57) {

            lcd.print(key);

            val = (val \* 10) + (key - '0');*// Añadir número al valor*

        }

    }

    delay(200);

    return val;

}

**Explicación**

El programa comienza en setup(), donde se inicializa el LCD con 16 columnas y 2 filas, y se posiciona el cursor en 0,0.

En la función loop() se encuentra el resto del programa. Al iniciar, el programa muestra el menú en el LCD antes de comenzar a leer la opción. Para leer cada dato se debe presionar la tecla de numeral (#) para indicar un “enter” (similar al botón en otras prácticas).

El programa cuenta con una función read() de tipo float, esta función se encarga de leer del teclado lo que se encuentre actualmente presionado y, hace su debida conversión de char a flotante.

El programa también cuenta con una función render(), esta recibe 4 argumentos, el valor a renderizar en el LCD, su posición en X, su posición en Y, y un booleano que indica si se debe limpiar la pantalla; estos últimos 3 argumentos son opcionales. También tiene una sobrecarga para recibir una cadena de caracteres como valor.

Una vez ingresado el dato de opción, se leen los dos números (o uno si la opción es 4) y se ejecuta el caso correspondiente (o se muestra un “\* ERROR \*” si la opción no existe). Tras hacer la debida acción correspondiente a su caso, el programa espera a que se presione la tecla de asterisco (\*) para reiniciar el programa.

**Observaciones**

La parte más tardada fue soldar los pines al LCD y verificar su continuidad, la programación es muy fácil.

**Conclusión**

Está práctica desenvuelve más en la potencia de las tarjetas de desarrollo, pues podemos utilizar componentes externos para ensamblar prácticas más funcionales, como el uso de LCD y el teclado matricial hexadecimal 4x4.