17-4-2021

ADRIAN OROZCO ANZURES 18300328  
CARLOS DANIEL LOZANO VAZQUEZ 18300249

SiSTEMAS EMBEBIDOS I

CETI COLOMOS

**Menú DipSW**

Actividad 13 – Práctica 7

Logotipo

Descripción generada automáticamente

**Objetivo**

Utilizar los puertos de un Arduino y conectarle dispositivos básicos de entrada y salida para comprobar un programa de menú de opciones.

**Descripción**

Utilizar Arduino MEGA y conectarle un dip switch (interruptores) a uno de sus puertos para introducir datos y conectar 8 LEDs a otro puerto para mostrar resultados (salida).

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

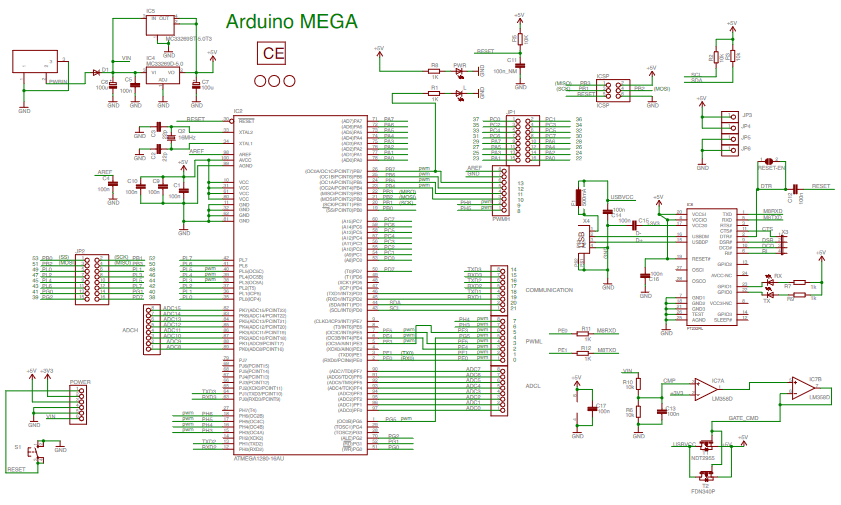
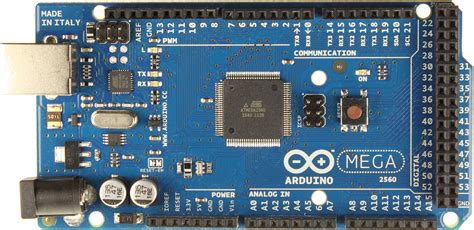
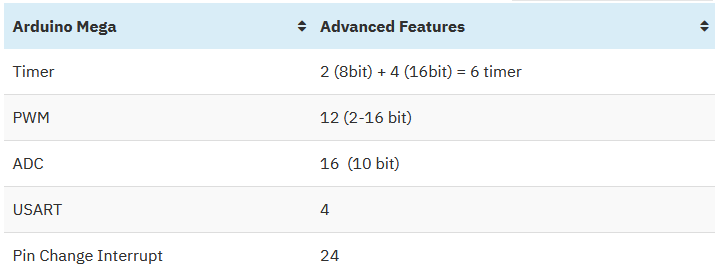
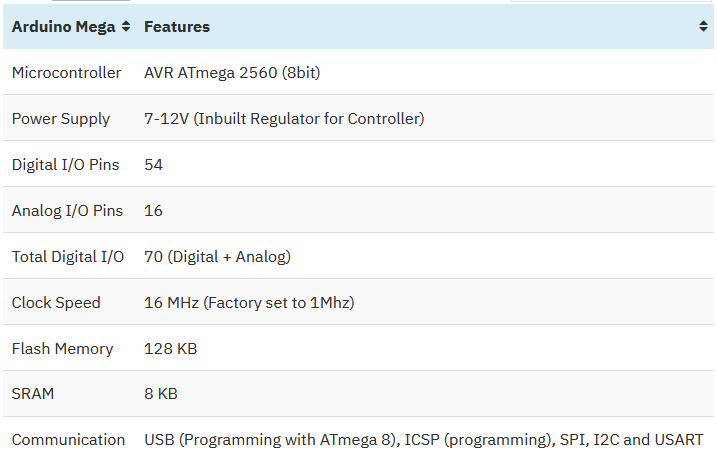
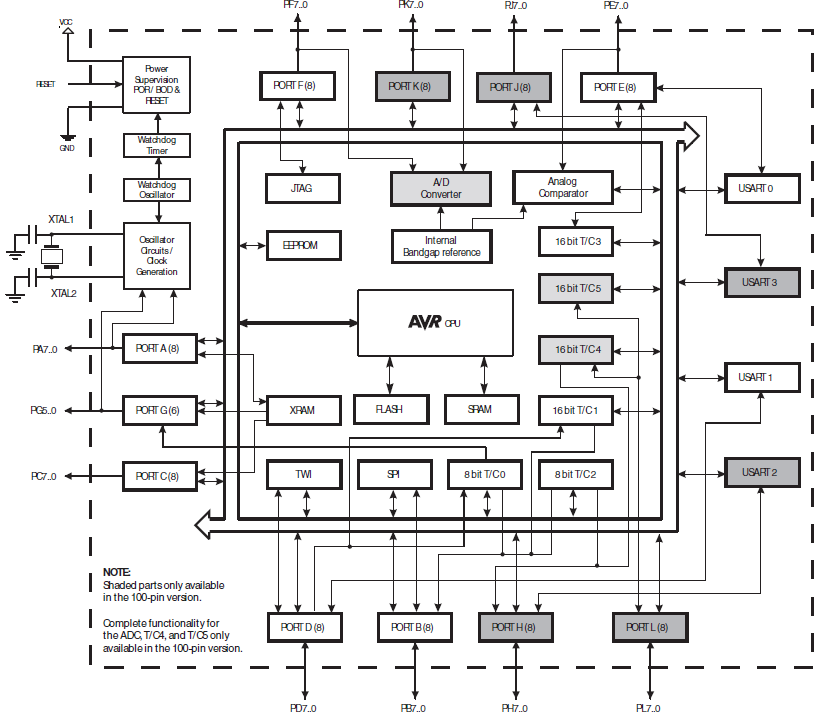
1. Si x=0 mostrar y + z.
2. Si x=1 mostrar y - z.
3. Si x=2 mostrar y \* z.
4. Si x=3 mostrar y / z.
5. Si x=4 mostrar los números de “y” hasta 99 separados por un lapso de 0.5 segundos.
6. Si x=5 mostrar el mayor de y, z.

**Resumen**

Usando una tarjeta de Arduino MEGA, se debe crear un programa que, mediante un dip switch, lea un número (en binario) el cual represente la opción que se va a ejecutar. Para cada caso se deben de ingresa dos datos mediante el dip switch, 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**

**Imagen que contiene circuito, competencia de atletismo

Descripción generada automáticamente**

**Código**

#include <avr/io.h>

int sw[8] = {22, 23, 24, 25, 26, 27, 28, 29};*//PORTA*

int button = 2, sign = 49;*//PORTC y boton*

int enter();

int may(int, int);

void setup()

{

    DDRC = 0xFF;*// se declara como salida PORTC*

    DDRA = 0x00;*//se declara como entrada al portA*

    pinMode(button, INPUT\_PULLUP);*//boton para secuencia*

    pinMode(sign, OUTPUT);*//led de negativo*

}

void loop()

{

    int opc = 0, r = 0;

    int \*matriz;

    digitalWrite(sign, 0);

    PORTC = 0x01;

    while( digitalRead(button) == LOW)*//ciclo de inicio hasta que el boton = 1*

    {

      for(int cont = 0; cont < 7; cont++)

      {

        PORTC = PORTC << 1;*//flujo a la izquierda de 1 bit << "1"*

        delay(200);

      }

      for(int cont = 8; cont > 1; cont--)

      {

        PORTC = PORTC >> 1;*// flujo a la derecha de 1 bit >> "1"*

        delay(200);

      }

      delay (500);*//medio segundo para poder presionar el boton*

    }

    PORTC = 0x00;

    delay(200);

    opc = enter();

    if(opc!=4)*//arreglo de 2*

    {

      matriz = new int[2];

      for (int cont = 0; cont < 2; cont++)*//se ingresan los 2 valores*

           matriz[cont] = enter();

    }

    else*//arreglo de 1*

    {

      matriz = new int [1];*// se ingresa su unico valor*

      matriz[0] = enter();

    }

    switch (opc)

    {

      case 0:

            r = matriz[0] + matriz[1];*//suma de los 2 valores*

            PORTC = r;*//se muestra*

            break;

      case 1:

            if (max(matriz[0], matriz[1]) == matriz[0])*//se analiza cual es ma grande*

            {

              r = matriz[0] - matriz[1];

            }

            else

            {

              r = matriz[1] - matriz[0];*//valor negativo*

              digitalWrite(sign, HIGH);

            }

            PORTC = r;

            break;

      case 2:

            r = matriz[0] \* matriz[1];*//multiplicacion*

            PORTC = r;

            break;

      case 3:

            r = matriz[0] / matriz[1];*//div*

            PORTC = r;

            break;

      case 4:

            for(int cont = matriz[0]; cont <= 99; cont++)*//muestra del valor decimal Y al valor decimal 99*

            {

              PORTC = cont;

              delay(500);

            }

            break;

      case 5:

            PORTC = 0x00;

            delay(500);

            PORTC = max(matriz[0], matriz[1]);*//se muestra el mayor*

            break;

      default:

            PORTC = 0xFF;*// si no es ninguna muestra FF Hex*

            break;

    }

    while(digitalRead(button)==LOW){}*//ciclo para apreciar resultado*

    delay(200);

}

int enter()

{

  int num = 0;

  while(digitalRead(button) == LOW)

   {

      for(int cont = 0; cont < 8; cont++)

      {

        bitWrite(num, cont, digitalRead(sw[cont]));*// se almacena en num lo que hay en PORTA*

        bitWrite(PORTC, cont, bitRead(num, cont));*// se muestra el numero ingresado otra forma PORTC = num;*

      }

   }

  delay(200);

  return num;

}

**Explicación**

El programa comienza en setup(), donde se declara el puerto A como entrada y el puerto C como salida. A su vez, se declara el botón (conectado al pin 2) como entrada y LED (pin 49) como salida.

En la función loop() se encuentra el resto del programa. Al iniciar, el programa espera al pulso del botón antes de comenzar a leer la opción. Para leer cada dato se debe presionar el botón.

El programa cuenta con una función enter() de tipo int, esta función se encarga de leer, mediante el puerto A, lo que se encuentre en el dip switch y, hace su debida conversión de binario a decimal entero.

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 #FF en salida si la opción no existe). Tras hacer la debida acción correspondiente a su caso, el programa espera a que se presione el botón una vez más antes de reiniciar el programa.

**Observaciones**

Realmente fue una práctica muy sencilla y no hubo complicaciones mayores, a excepción de que al inicio queríamos programar y verificar el código en Visual Studio Code, pero los compiladores de VSCode y Arduino IDE son diferentes, así que usamos el de Arduino.

**Conclusión**

Esta práctica es una muy sencilla y simple introducción a el mundo de tarjetas de desarrollo, un simple realizado en un lenguaje de programación muy similar a C/C++. Nos demuestra lo potente que son estas tarjetas incluso para aplicaciones sencillas.