|  |  |  |  |
| --- | --- | --- | --- |
| Nombre y Apellido: | | Curso: 1M10 |  |
| Practico Nº: 8 | Fecha: | Nota: |
| TECNICATURA UNIVERSITARIA EN MECATRÓNICA **Herramientas Informáticas** | | |
| Objetivos:   * Aplicar y profundizar lo visto sobre programación en C/C++ a un entorno de sistemas embebidos. * Utilizar la plataforma de desarrollo Arduino, manejar su IDE y las herramientas de depuración que brinda. * Poner en practica conocimientos de matemática y física. * Poner en practica conocimientos de electrónica, tanto del área analógica como del área digital.   Aclaraciones:   * Las soluciones de las actividades propuestas deberán quedar acentuada en su correspondiente carpeta de trabajos prácticos que deberá ser presentada al finalizar el cursado de la materia.  1. Realizar un programa para Arduino que me permita simular el funcionamiento de las operaciones lógicas. El circuito presentará:    1. 3 entradas que permitirán seleccionar entre las distintas operaciones (NOT, AND, OR, XOR, NAND, NOR, XNOR).    2. 2 entradas que funcionaran como variables independientes de la operación lógica. 3) Una salida que me mostrará el resultado de dicha operación. 2. Escribir un programa para Arduino que permita destellar 3 led’s con frecuencias de 500ms, 200ms y 100ms respectivamente. 3. Escribir una programa para Arduino que encienda en forma de barrido de derecha a izquierda, 8 led’s conectados a sus puertos. El tiempo total en el que se debe barrer los 8 led’s es de 500ms. 4. Modificar el programa anterior para que mediante un pulsador se pueda cambiar la secuencia de barrido (de izquierda a derecha, de derecha a izquierda, del centro a los costados, de los costados al centro) y con otro pulsador se podrá modificar el tiempo del barrido (100ms, 200ms, 500ms, 1s). 5. Controlar un display de 7 segmentos mediante Arduino para que el mismo muestre una cuenta descendente del 9 al 0 (que corresponderán a segundos) luego de presionado un pulsador. El valor mostrado en el display también deberá ser enviado a la PC mediante la comunicación serie. 6. Controlar 4 displays de 7 segmentos para que muestren un valor numérico desde el 0000 al 9999. El número mostrado por los displays deberá ser enviado al Arduino mediante la comunicación serie. 7. Realizar un controlador para un motor de corriente continua (CC) mediante Arduino. Se deberá poder indicar mediante la comunicación serie el sentido de giro que tendrá el motor y ademas regular su velocidad entre un 0%, 25%, 50%, 75% y 100%. Recuerde que debe utilizar un puente H para llevar a cabo esta tarea. Podrá utilizar el driver L293, L298 u otro dispositivo similar para suministrar la corriente necesaria al motor. 8. Escribir un programa para Arduino que permita realizar una celda fotosensible. El controlador deberá leer un sensor del tipo LDR (fotorresistencia) y un potenciómetro que se utilizará como referencia. Cuando el valor de iluminación este por debajo del valor indicado mediante el potenciómetro, se encenderá una salida para activar una lampara led. 9. Realizar un programa para Arduino que permita conocer la distancia a la que el dispositivo se encuentre de algún objeto utilizando un sensor ultrasonido (similar al HC-SR04). El valor de la distancia se deberá enviar a la PC mediante la comunicación serie. 10. Controlar un LCD alfanumérico de 2 lineas por 16 caracteres con Arduino. El texto que se mostrará en pantalla será enviado desde la PC al Arduino por la comunicación serie. | | | |

/\*A) Realizar un programa para Arduino que me permita simular el funcionamiento de las operaciones

lógicas. El circuito presentará:

1) 3 entradas que permitirán seleccionar entre las distintas operaciones (NOT, AND, OR, XOR,

NAND, NOR, XNOR).

2) 2 entradas que funcionaran como variables independientes de la operación lógica.

3) Una salida que me mostrará el resultado de dicha operación.\*/

byte logica;

void setup() {

  for(byte i=2; i<=6;){

    pinMode(i,INPUT);

    }

    pinMode(13,OUTPUT);

}

void loop() {

  logica= digitalRead(4)+digitalRead(5)<<1+digitalRead(6)<<2;

  switch(logica){

    case 0;

     digitalWrite(13,(digitalRead(2)&digitalRead(3));

     break;

    case 1;

     digitalWrite(13,(digitalRead(2)|digitalRead(3));

     break;

    case 2;

     digitalWrite(13,!(digitalRead(2)^digitalRead(3));

     break;

     case 3;

     digitalWrite(13,(digitalRead(2)^digitalRead(3));;

     break;

    case 4;

      digitalWrite(13,!(digitalRead(2)&digitalRead(3));

     break;

     case 5;

      digitalWrite(13,!(digitalRead(2)|digitalRead(3));

     break;

    case 6;

      digitalWrite(13,!digitalRead(2));

     break;

    }

}

/\*B) Escribir un programa para Arduino que permita destellar 3 led’s con frecuencias de 500ms, 200ms y

100ms respectivamente.

\*/

void setup() {

  long int tiempo=0, tiempo2=0,tiempo3=0;

pinMode(12,OUTPUT);

pinMode(11,OUTPUT);

pinMode(10,OUTPUT);

digitalWrite(12,HIGH);

digitalWrite(11,HIGH);

digitalWrite(10,HIGH);

}

void loop(){

  if((millis()-tiempo1)>500){

    tiempo1=millis();

    digitalWrite(12,digitalRead(12));

    }

  if((millis()-tiempo2)>200){

    tiempo1=millis();

    digitalWrite(11,digitalRead(11));

    }

    if((millis()-tiempo3)>100){

    tiempo1=millis();

    digitalWrite(10,digitalRead(10));

    }

}

/\*C) Escribir una programa para Arduino que encienda en forma de barrido de derecha a izquierda, 8

led’s conectados a sus puertos. El tiempo total en el que se debe barrer los 8 led’s es de 500ms.\*/

int pinArray [] = {1,2,3,4,5,6,7,8 };    // Vector donde se van a declarar los LEDs

int waitStart= 500;  // Tiempo entre encender un LED y otro

int tailLength = 0;     // Numero de LEDs activos

int lineSize = 8;       // Numero total de LEDs

void setup()

  {

    int i;

    for (i=0; i&lt; lineSize; i++)

    {

      pinMode(pinArray[i], OUTPUT);

    }

  }

void loop()

   {

     int i;

     int tailCounter = tailLength;

     for (i=0; i&lt;lineSize; i++)

     {

      digitalWrite(pinArray[i],HIGH);

      delay(waitStart); //

      if (tailCounter == 0)

      {

        digitalWrite(pinArray[i-tailLength],LOW);

      }

      else

         if (tailCounter &gt; 0)

            tailCounter--;

      }

      for (i=(lineSize-tailLength); i&lt;lineSize; i++)

      {

       digitalWrite(pinArray[i],LOW);

       delay(waitStart);

      }

 }

/\*C) Escribir una programa para Arduino que encienda en forma de barrido de derecha a izquierda, 8

led’s conectados a sus puertos. El tiempo total en el que se debe barrer los 8 led’s es de 500ms.\*/

int A = 2;

int B = 3;

int C = 4;

int D = 5;

int E = 6;

int F = 7;

int G = 8;

int Dp = 9;

void setup() {

  pinMode (A, OUTPUT);

  pinMode (B, OUTPUT);

  pinMode (C, OUTPUT);

  pinMode (D, OUTPUT);

  pinMode (E, OUTPUT);

  pinMode (G, OUTPUT);

  pinMode (F, OUTPUT);

  pinMode (Dp, OUTPUT);

}

void loop() {

 digitalWrite(A, HIGH);

 delay(300);

 digitalWrite(A, LOW);

 delay(300);

 digitalWrite(B, HIGH);

 delay(300);

 digitalWrite(B, LOW);

 delay(300);

 digitalWrite(C, HIGH);

 delay(300);

 digitalWrite(C, LOW);

 delay(300);

 digitalWrite(D, HIGH);

 delay(300);

 digitalWrite(D, LOW);

 delay(300);

 digitalWrite(E, HIGH);

 delay(300);

 digitalWrite(E, LOW);

 delay(300);

 digitalWrite(F, HIGH);

 delay(300);

 digitalWrite(F, LOW);

 delay(300);

 digitalWrite(G, HIGH);

 delay(300);

 digitalWrite(G, LOW);

 delay(300);

 digitalWrite(Dp, HIGH);

 delay(300);

 digitalWrite(Dp, LOW);

 delay(600);

 digitalWrite(A, HIGH);

 digitalWrite(B, HIGH);

 digitalWrite(C, HIGH);

 digitalWrite(D, HIGH);

 digitalWrite(E, HIGH);

 digitalWrite(F, HIGH);

 digitalWrite(G, HIGH);

 delay(600);

 digitalWrite(A, LOW);

 digitalWrite(B, LOW);

 digitalWrite(C, LOW);

 digitalWrite(D, LOW);

 digitalWrite(E, LOW);

 digitalWrite(F, LOW);

 digitalWrite(G, LOW);

 delay(600);

}

/\*E) Controlar un display de 7 segmentos mediante Arduino para que el mismo muestre una cuenta

descendente del 9 al 0 (que corresponderán a segundos) luego de presionado un pulsador. El valor

mostrado en el display también deberá ser enviado a la PC mediante la comunicación serie.\*/

int contador = 0;

int presionado1;

int presionado2;

int pulsador\_disminuir = 9;

int pulsador\_aumentar = 10;

void encender(int a, int b, int c, int d, int e, int f, int g)

{

  //Funcion que permite mostrar los numeros del 0 al 9

  //Asigna un segmento del display a cada pin

  digitalWrite(2, a);  //Asigna el segmento 'a' al pin 2

  digitalWrite(3, b);  //Asigna el segmento 'b' al pin 3

  digitalWrite(4, c);  //Asigna el segmento 'c' al pin 4

  digitalWrite(5, d);  //Asigna el segmento 'd' al pin 5

  digitalWrite(6, e);  //Asigna el segmento 'e' al pin 6

  digitalWrite(7, f);  //Asigna el segmento 'f' al pin 7

  digitalWrite(8, g);  //Asigna el segmento 'g' al pin 8

}

void setup() {

  //Inicializa los pines como entradas/salidas digitales

  pinMode(2, OUTPUT);

  pinMode(3, OUTPUT);

  pinMode(4, OUTPUT);

  pinMode(5, OUTPUT);

  pinMode(6, OUTPUT);

  pinMode(7, OUTPUT);

  pinMode(8, OUTPUT);

  pinMode(9, INPUT);  //Pulsador aumentar como entrada digital

  pinMode(10, INPUT);  //Pulsador disminuir como entrada digital

}

void loop() {

  //Condicionales para antirrebote y uso del pulsador aumentar

  if (digitalRead(pulsador\_aumentar) == LOW)

  {

    presionado1 = 1;   //Variable del antirrebote que cambia cuando se presiona el pulsador

  }

  if (digitalRead(pulsador\_aumentar) == HIGH && presionado1 == 1)

  {

    presionado1 = 0;  //Se reinicia la variable antirrebote

    contador++;       //Aumenta el contador

    if (contador > 9)

    {

      contador = 9; //Si el contador esta en 9 y se aumenta, sigue mostrando el 9

    }

  }

  //Condicionales para antirrebote y uso del pulsador disminuir

  if (digitalRead(pulsador\_disminuir) == LOW)

  {

    presionado2 = 1;    //Variable del antirrebote que cambia cuando se presiona el pulsador

  }

  if (digitalRead(pulsador\_disminuir) == HIGH && presionado2 == 1)

  {

    presionado2 = 0;   //Se reinicia la variable antirrebote

    contador--;        //Disminuye el contador

    if (contador < 0) //Si el contador esta en 0 y se disminuye, sigue mostrando el 0

    {

      contador = 0;

    }

  }

  switch (contador)

  {

    //Dependiendo del contador, se muestra el numero indicado

    //Envia estados (1 y 0) al display dependiendo del numero a mostrar

    case 0:

      encender (1, 1, 1, 1, 1, 1, 0); //Muestra el numero 0

      break;

    case 1:

      encender (0, 1, 1, 0, 0, 0, 0); //Muestra el numero 1

      break;

    case 2:

      encender (1, 1, 0, 1, 1, 0, 1); //Muestra el numero 2

      break;

    case 3:

      encender (1, 1, 1, 1, 0, 0, 1); //Muestra el numero 3

      break;

    case 4:

      encender (0, 1, 1, 0, 0, 1, 1); //Muestra el numero 4

      break;

    case 5:

      encender (1, 0, 1, 1, 0, 1, 1); //Muestra el numero 5

      break;

    case 6:

      encender (1, 0, 1, 1, 1, 1, 1); //Muestra el numero 6

      break;

    case 7:

      encender (1, 1, 1, 0, 0, 0, 0); //Muestra el numero 7

      break;

    case 8:

      encender (1, 1, 1, 1, 1, 1, 1); //Muestra el numero 8

      break;

    case 9:

      encender (1, 1, 1, 0, 0, 1, 1); //Muestra el numero 9

      break;

  }

}