

// Pin 13 has an LED connected on most Arduino boards.

// give it a name:

int ledRed = 13;

int ledOrange = 12;

int ledBlue = 11;

int ledGreen = 10;

int btnSecuence = 4;

int btnBinaryUp = 2;

int btnBinaryDown = 3;

int btnReset = 7;

//Sumador para aumentar y decrementar los numeros binarios

int binaryInt = 0;

//Indica si algun led se encuentra prendido

bool anyLedOne = false;

//Mantiene el numero del ultimo led prendido va de 0 a 3 siendo 0 el primero y 3 el ultimo

//ademas la variable anyLedOne debe estar en true

int ledOn = 0;

//Estado de boton 1 secuencia

int btnSecuenceState = 0;

int btnResetState = 0;

//Boton de secuencia

int btnBinaryUpState = 0;

int btnBinaryDownState = 0;

//Indica que comienzo a utilizar los botones de binario

bool useBinarioBtn = false;

void setup()

{

Serial.begin(9600); // open the serial port at 9600 bps:

pinMode(ledRed, OUTPUT);

pinMode(ledOrange, OUTPUT);

pinMode(ledBlue, OUTPUT);

pinMode(ledGreen, OUTPUT);

//Boton para aumentar secuencia

pinMode(btnSecuence, INPUT\_PULLUP);

//BOtones para aumentar y decrementar binarios

pinMode(btnBinaryUp, INPUT\_PULLUP);

pinMode(btnBinaryDown, INPUT\_PULLUP);

pinMode(btnReset, INPUT\_PULLUP);

}

// the loop routine runs over and over again forever:

void loop()

{

btnResetState = digitalRead(btnReset);

if (btnResetState == HIGH)

{

turnOffAll();

binaryInt = 0;

}

btnBinaryUpState = digitalRead(btnBinaryUp);

btnBinaryDownState = digitalRead(btnBinaryDown);

if (btnBinaryUpState == HIGH)

{

//cancelo la funcionalidad de secuencia

btnSecuenceState = false;

if (useBinarioBtn == false)

{

//Primera vez que apreto el boton de secuencia UP

useBinarioBtn = true;

}

if (binaryInt < 15)

{

++binaryInt;

String binary = getBinary(binaryInt);

Serial.println(binary);

turnWithBinary(binary);

}

}

if (btnBinaryDownState == HIGH)

{

//cancelo la funcionalidad de secuencia

btnSecuenceState = false;

if (useBinarioBtn == false)

{

//Primera vez que apreto el boton de secuencia UP

useBinarioBtn = true;

}

if (binaryInt > 0)

{

--binaryInt;

String binary = getBinary(binaryInt);

turnWithBinary(binary);

}

}

//Leo el valor del boton 1 de secuencia

btnSecuenceState = digitalRead(btnSecuence);

if (btnSecuenceState == LOW)

{

}

else

{

useBinarioBtn = false;

anyLedOne = true;

if (ledOn == 0)

{

//Apago todo los leds

turnOffAll();

digitalWrite(ledRed, HIGH);

++ledOn;

}

else if (ledOn == 1)

{

turnOffAll();

digitalWrite(ledOrange, HIGH);

++ledOn;

}

else if (ledOn == 2)

{

turnOffAll();

digitalWrite(ledBlue, HIGH);

++ledOn;

}

else if (ledOn == 3)

{

turnOffAll();

digitalWrite(ledGreen, HIGH);

ledOn = 0;

}

}

//digitalWrite(ledRed, HIGH);

//digitalWrite(ledOrange, HIGH);

//digitalWrite(ledBlue, HIGH);

//digitalWrite(ledGreen, HIGH);

delay(200);

}

void turnOffAll()

{

digitalWrite(ledRed, LOW);

digitalWrite(ledOrange, LOW);

digitalWrite(ledBlue, LOW);

digitalWrite(ledGreen, LOW);

}

void turnWithBinary(String binary)

{

//Apago todos los led's y segun el caracter binario que le corresponde prendo uno o el otro

turnOffAll();

if (binary[0] == '1')

digitalWrite(ledRed, HIGH);

if (binary[1] == '1')

digitalWrite(ledOrange, HIGH);

if (binary[2] == '1')

digitalWrite(ledBlue, HIGH);

if (binary[3] == '1')

digitalWrite(ledGreen, HIGH);

}

//Le envio un numero y me lo convierte a binario

String getBinary(int number)

{

int zeros = String(number, BIN).length(); //This will check for the length of myNum in binary.

String myStr = "";

for (int i = 0; i < 4 - zeros; i++)

{ //This will add zero to string as need

myStr = myStr + "0";

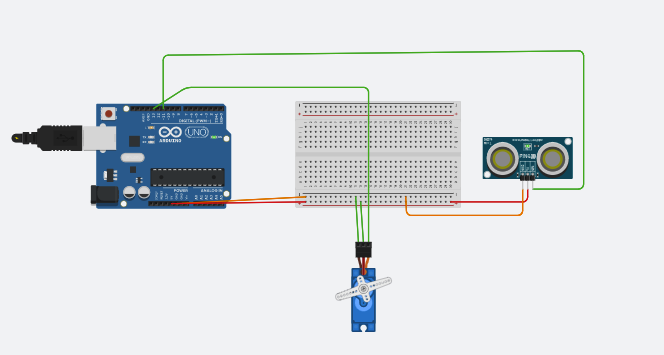
}

myStr = myStr + String(number, BIN); //Convierto a binario

return myStr;

}

Parte 2



#include <Servo.h>

const int trigPin = 11;

const int echoPin = 11;

Servo servoMotor;

void setup()

{

// initialize serial communication:

Serial.begin(9600);

servoMotor.attach(13);

}

void loop()

{

long duration, inches, cm;

pinMode(trigPin, OUTPUT);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

pinMode(echoPin, INPUT);

duration = pulseIn(echoPin, HIGH);

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

servoMotor.write(inches);

delay(100);

}

long microsecondsToInches(long microseconds)

{

return microseconds / 74 / 2;

}

long microsecondsToCentimeters(long microseconds)

{

return microseconds / 29 / 2;

}