



INSTITUTO TECNOLÓGICO DE OAXACA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Sistemas Programables

Prácticas:

- ✓ Estacionamiento con Twitter y Ruby
- ✓ Carrito automático

Docente: Hermes Ojeda Ruíz

Alumnos:

Cruz Aquino Jessica Margarita
Terrazas Salinas Froylán
Ortiz Venegas Jonathan Daniel

29/07/14

Estacionamiento con Twitter y Ruby

Arduino

```
#include <Servo.h>
```

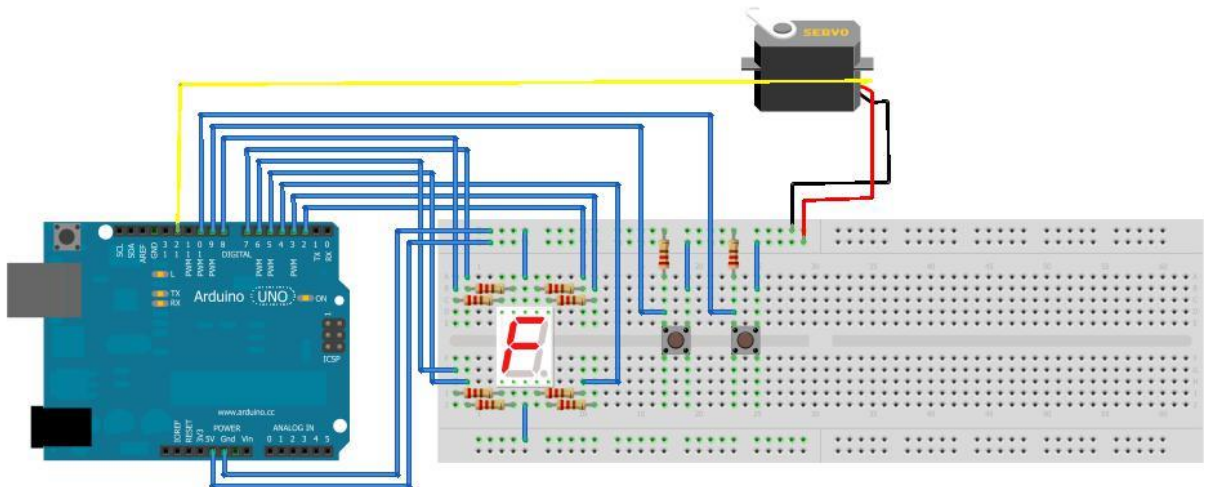
```
const byte contadorPin[] = {2,3,4,5,6,7,8};
const byte valor[10]= {B00111111, B00000110,
B01011011, B01001111,
B01100110, B01101101, B01111101,
B00000111, B01111111, B01101111};
byte mascara = 1;
int switchEstado1=0;
int switchEstado2=0;
int ocupado=8;
Servo myServo;
```

```
void setup()
{
  Serial.begin(9600);
  int i = 0;
  for ( i = 0; i < 7; i++)
    pinMode(contadorPin[i], OUTPUT);
  pinMode(9,INPUT);
  pinMode(10,INPUT);
  myServo.attach(12);
  contador(8);
}

void loop()
{
  switchEstado1=digitalRead(9);
  switchEstado2=digitalRead(10);
  if (switchEstado1==HIGH && ocupado!=0)
```

```
{ myServo.write(177);
  delay(3000);
  myServo.write(80);
  ocupado --;
  contador(ocupado);
  Serial.print(ocupado);
}
else
if (switchEstado2==HIGH && ocupado<8)
{ myServo.write(177);
  delay(3000);
  myServo.write(80);
  ocupado++;
  contador(ocupado);
  Serial.print(ocupado);
}
}

void contador(int numero)
{ byte j=0;
  for (mascara = 0000001; mascara>0; mascara
<= 1)
  { if (valor[numero] & mascara)
    { digitalWrite(contadorPin[j],HIGH);
    }
    else { digitalWrite(contadorPin[j],LOW);
    }
    j++;
  }
}
```



Ruby

```
require 'twitter'
require 'serialport'
def send_tweet(txt)
  client = Twitter::REST::Client.new do |config|
    config.consumer_key = 'CttHz8NgS24n3ZJhulxtOSb17'
    config.consumer_secret = 'zkp90JHC5qWfKawd6BbDzEghOH1IDu1KwXR4ODFSjIEbFPjmJc'
    config.oauth_token = '252924295-5iykxILUtxpV1ZPQyty8LhGeoNOgfh6RWPCAz35c'
    config.oauth_token_secret = 'idyBCkadfbLvx44eLOZTjVIHcGT0MzwQqiWLa381TZFY7'
  end
  puts "se publico"
  client.update(txt)
end

anterior=0
reciente=0

port_str = "/dev/ttyACM0" #may be different for you
baud_rate = 9600
data_bits = 8
stop_bits = 1
parity = SerialPort::NONE
sp = SerialPort.new(port_str, baud_rate, data_bits, stop_bits, parity)

while true do
  while (i = sp.gets) do
    puts i
    send_tweet("espacio disponibles"+i)
  end
end

sp.close
```

Carrito Automático

Arduino

```
const int controlPin1 =2;
const int controlPin2 =3;
const int enablePin =9;

const int trigpin = 10;
const int echopin = 11;
long duracion, distancia;

int motorEnabled =1;
int motorSpeed = 0;
int motorDirection =1;

const int controlRotPin1 =4;
const int controlRotPin2 =5;
const int enableRotPin=8;

int motorEnabled2=0;
int motorDirection2=1;

long intervalo2 = 2000; // es el tiempo
de nuestro delay
long tiempo2 = 0;
long tiempoAnterior2 = 0;

long intervalo = 300; // es el tiempo de
nuestro delay
long tiempo = 0;
long tiempoAnterior = 0;

void setup()
{pinMode(trigpin,OUTPUT);
pinMode(echopin,INPUT);
pinMode(controlPin1, OUTPUT);
pinMode(controlPin2, OUTPUT);
pinMode(enablePin, OUTPUT);

digitalWrite(enablePin, LOW);
Serial.begin(9600);
}

void loop()
{ Serial.println(motorSpeed);
  calcularDistancia();
  avanzar();
  if(distancia >0 && distancia <=25)
  { analogWrite(enablePin, 0);
    rotarDerecha();
    retroceder();
    delay(1000);
    rotarIzquierda();
    avanzar();
  }
  else
  { avanzaNormal();
    avanzar();
  }
}
```

```
void avanzar(){
  digitalWrite(controlPin1,LOW);
  digitalWrite(controlPin2, HIGH);
  analogWrite(enablePin, 200);
}

void retroceder(){
  digitalWrite(controlPin1, HIGH);
  digitalWrite(controlPin2, LOW);
  analogWrite(enablePin, 209);
}

void avanzaNormal(){
  digitalWrite(controlRotPin1, HIGH);
  digitalWrite(controlRotPin2, HIGH);
  analogWrite(enableRotPin, 69);
}

void rotarDerecha(){

  digitalWrite(controlRotPin1, HIGH);
  digitalWrite(controlRotPin2, LOW);
  analogWrite(enableRotPin, 300);
}

void rotarIzquierda(){
  digitalWrite(controlRotPin1, LOW);
  digitalWrite(controlRotPin2, HIGH);
  analogWrite(enableRotPin, 300);
}

int calcularDistancia(){
  digitalWrite(10,LOW);
  delayMicroseconds(5);
  digitalWrite(10,HIGH);
  delayMicroseconds(10);
  duracion = pulseIn(11, HIGH);
  distancia=int(0.017*duracion);

  Serial.print("duracion:");
  Serial.println(duracion);
  Serial.print("distancia:");
  Serial.print(distancia);
  Serial.println();

  // delay(10);
  return distancia;
}

boolean calculaTiempo(){
  if(tiempo - tiempoAnterior > intervalo)
  {
    return true;
  }
  else
  return false;
}
```

