//Asignamos variables,constantes

#define boton\_pulsado = HIGH

const int led\_1 = 9; //MOTOR

const int led\_2 = 10; //PORTON

const int boton = 8; //BOTON

const int FCC = 11; //Final de carrera Cerrado

const int FCA = 12; //Final de carrera Abierto

int EstadoBoton;

byte porton

// int estadoLed;

int ult\_est\_bot = HIGH;

// Variables para almacenar el estado del LED y el tiempo de retraso.

// bool estadoled = LOW;

unsigned long anteriorMillis = 0;

// Define la duración del rebote y el tiempo de retraso.

const int debounceDuration = 50;

const int delayTime = 2000;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void setup() {

  pinMode(led\_1, OUTPUT);

  pinMode(led\_2, OUTPUT);

  pinMode(boton, INPUT\_PULLUP);

  pinMode(FCC, INPUT\_PULLUP);

  pinMode(FCA, INPUT\_PULLUP);

**Serial**.begin(9600); //INICIAMOS EL MONITOR SERIAL

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void loop() {

  EstadoBoton = digitalRead(boton);

  anteriorMillis = millis();

  switch (porton) {

      if (ult\_est\_bot != EstadoBoton) {

        digitalWrite(9, HIGH);

        digitalWrite(10, HIGH);

**Serial**.println("se está abriendo el porton");

        EstadoBoton = ult\_est\_bot;

        delay(2000);

        while (FCC == LOW) {

**Serial**.println("el porton se abrió");

        }

      }

      else if (ult\_est\_bot == EstadoBoton) {

        digitalWrite(9, LOW);

        digitalWrite(10, LOW);

**Serial**.println("el portón está cerrado");

        EstadoBoton != ult\_est\_bot;

        //delay(2000);

        //Serial.println("el portón está cerrado");

      }

      else if (ult\_est\_bot == EstadoBoton) {

**Serial**.println("se esta cerrando el porton");

      }

      else {

        digitalWrite(led\_1, LOW);

        digitalWrite(led\_2, LOW);

      }

      // esperamos el tiempo de retraso

      anteriorMillis = millis();

      //         while (millis() - anteriorMillis < delayTime) {}

  }

  //Si el led esta apagado, muestra el mensaje

}