ANEXO III

Código fuente de Arduino

Elías Gil de Avalle Muñoz

Diseño, construcción y ensayo de un equipo de regulación de velocidad para aplicaciones escénicas

CÓDIGO FUENTE DE ARDUINO

A continuación, se presenta el código fuente de la placa base Arduino que trabaja actualmente en la placa Mega 2560. A lo largo del código, se explica la función de cada línea después de las dos barras (// y en gris)

```
#include <LiquidCrystal.h>  // Librería necesaria para LCD Arduino

LiquidCrystal lcd(30,31,32,33,34,35);  // Definir las salidas
digitales a utilizar por la LCD Arduino

int TRIG = 53;  // Envía señal de ultrasonido
int ECO = 52;  // Recibe señal de ultrasonido
int DURACION; // Calcula el tiempo que ha tardado en enviarse la señal
de ultrasonido y la recepción del mismo
int DISTANCIA;  // Convierte tiempo en distancia a través de los
parámetros del fabricante, en este caso 1cm = 58,2
int periodo = 1;  // Delay tiempo ultrasonido
unsigned long TiempoAhora = 0;
int periodo2 = 3000;  // Delay tiempo activado final de carrera de
plataforma giratoria, sentido horario
unsigned long TiempoAhora2 = 0;
int periodo3 = 3000;  // Delay tiempo activado final de carrera de
plataforma giratoria, sentido antihorario
unsigned long TiempoAhora3 = 0;
int periodo4 = 5000;  // Delay tiempo activado final de carrera de
plataforma giratoria, 5 activaciones de Final de carrera
unsigned long TiempoAhora4 = 0;

int MANUAL = 40;  // Interruptor de modo manual
int AUTO = 41;  // Interruptor de modo automático

int M1 = LOW;  // Determina si RELE 1 está activo
int M2 = LOW;  // Determina si RELE 2 está activo
int M3 = LOW;  // Determina si RELE 3 está activo
int M4 = LOW;  // Determina si RELE 4 está activo
int M5 = LOW;  // Determina si RELE 5 está activo

int C = 3;  // Definimos que el CLK del encoder es la entrada digital
6 (Automático)
int D = 5;  // Definimos que el DT del encoder es la entrada digital 5
(Automático)
int PULSADOR2 = 6;  // Definimos que el SW del encoder es la entrada
digital 3 (Automático)
int A = 2;  // Definimos que el CLK del encoder es la entrada digital
6 (Manual)
int B = 4;  // Definimos que el DT del encoder es la entrada digital 5
(Manual)
int PULSADOR = 7;  // Definimos que el SW del encoder es la entrada
digital 3 (Manual)

int RELE1 = 8;  // Definir salida de PARO/MARCHA
int RELE2 = 9;  // Definir salida de AVANCE/RETROCESO
int RELE3 = 10;  // Definir salida de VELOCIDAD 1
int RELE4 = 11;  // Definir salida de VELOCIDAD 2
int RELE5 = 12;  // Definir salida de RAMPA1/RAMPA2

int FINALCARRERAUP = 22;  // Definir entrada del final de carrera
dispuesto abajo del telón
```

```
int FINALCARRERADOWN = 23;  // Definir entrada del final de carrera
dispuesto arriba del telón
int FINALCARRERAPLAT = 51;  // Definir entrada del final de carrera
dispuesto abajo de la plataforma
int FINALCARRERAGIRA = 50;  // Definir entrada del final de carrera
dispuesto en la plataforma giratoria

int ESTADOTELONUP = LOW;  // Nos indica si el telón está subiendo
int ESTADOTELONDOWN = LOW;  // No indica si el telón está bajando
int ESTADOPLATUP = LOW;  // Indica que plataforma está subiendo
int ESTADOPLATDOWN = LOW;  // Indica que plataforma está bajando
int ESTADOGIRA = LOW;  // Indica que plataforma giratoria está girando
en sentido horario
int ESTADOGIRAANTI = LOW;  // Indica que plataforma giratoria está
girando en sentido antihorario
int ESTADOGIRA5 = LOW;  // Indica que plataforma giratoria está
girando hasta que cuente 5
int CONTADORGIRO = 0;  // Cuenta el n° de vueltas que lleva la
plataforma giratoria

int ESTADO1 = LOW;  //Nos permite conocer el estado del Relé que
activa PARO/MARCHA
int ESTADO2 = LOW;  //Nos permite conocer el estado del Relé que
activa AVANCE/RETROCESO
int ESTADO3 = LOW;  //Nos permite conocer el estado del Relé que
activa VELOCIDAD 1
int ESTADO4 = LOW;  //Nos permite conocer el estado del Relé que
activa VELOCIDAD 2
int ESTADO5 = LOW;  //Nos permite conocer el estado del Relé que
activa RAMPA 1 O RAMPA 2

int ANTERIOR = 0;  // Guarda el valor anterior del encoder (Manual)
int ANTERIOR2 = 0; // Guarda el valor anterior del encoder
(Automático)

volatile int POSICION = 0;  // Guarda el valor actual del encoder
(Manual)
volatile int POSICION2 = 0;  // Guarda el valor actual del encoder
(Automático)

void setup() {  //Iniciamos la pantalla y los INPUTS y OUTPUTS
  lcd.begin(16, 2);

  pinMode(A, INPUT);
  pinMode(B, INPUT);
  pinMode(PULSADOR, INPUT_PULLUP);
  pinMode(C, INPUT);
  pinMode(D, INPUT);
  pinMode(TRIG, OUTPUT);
  pinMode(ECO, INPUT);
  pinMode(FINALCARRERAUP, INPUT);
  pinMode(FINALCARRERADOWN, INPUT);
  pinMode(FINALCARRERAPLAT, INPUT);
  pinMode(FINALCARRERAGIRA, INPUT);
  pinMode(PULSADOR2, INPUT_PULLUP);
  pinMode(RELE1, OUTPUT);
  pinMode(RELE2, OUTPUT);
  pinMode(RELE3, OUTPUT);
  pinMode(RELE4, OUTPUT);
  pinMode(RELE5, OUTPUT);
  pinMode(MANUAL, INPUT);
```

```
  pinMode(AUTO, INPUT);

  attachInterrupt(digitalPinToInterrupt(A), encoder, LOW);
  attachInterrupt(digitalPinToInterrupt(C), encoder2, LOW);

  Serial.begin(9600);

}

void loop() {  // Proceso que lee Arduino continuamente sin parar
  if (POSICION != ANTERIOR){  // Determina la posición del encoder
(Manual)
    ANTERIOR = POSICION;
  }

  if (POSICION2 != ANTERIOR2){  // Determina la posición del encoder
(Auto)
    ANTERIOR2 = POSICION2;
  }

    digitalWrite(TRIG, HIGH);  // Determina la distancia del objeto al
emisor de ultrasonidos
  if(millis() >= TiempoAhora + periodo){
    TiempoAhora = millis();
    digitalWrite(TRIG, LOW);
  }
  DURACION = pulseIn (ECO, HIGH);
  DISTANCIA = DURACION / 58.2;

  Serial.println(DISTANCIA);

  if ((digitalRead(MANUAL) == LOW) && (digitalRead(AUTO) == LOW)){ //
Si manual y automatico están desactivados, todos los RELES se
desactivan excepto si del modo automático, tenemos un proceso
corriendo, cuando este finalize, se paran los reles
   lcd.setCursor(0,0);
   lcd.print(" VARIADOR FREQ. ");  // Imprime por pantalla
   lcd.setCursor(0,1);
   lcd.print(" ABB ACS-580-R1 ");
   if (M1 == HIGH  && ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW
&& ESTADOPLATUP == LOW
   && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI ==
LOW && ESTADOGIRA5 == LOW){
      digitalWrite(RELE1, LOW);
      M1 = LOW;
    }
   else if (M2 == HIGH  && ESTADOTELONUP == LOW && ESTADOTELONDOWN ==
LOW && ESTADOPLATUP == LOW
   && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI ==
LOW && ESTADOGIRA5 == LOW){
      digitalWrite(RELE2, LOW);
      M2 = LOW;
    }
   else if (M3 == HIGH  && ESTADOTELONUP == LOW && ESTADOTELONDOWN ==
LOW && ESTADOPLATUP == LOW
   && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI ==
LOW && ESTADOGIRA5 == LOW){
      digitalWrite(RELE3, LOW);
      M3 = LOW;
    }
```

```
   else if (M4 == HIGH  && ESTADOTELONUP == LOW && ESTADOTELONDOWN ==
LOW && ESTADOPLATUP == LOW
   && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI ==
LOW && ESTADOGIRA5 == LOW){
      digitalWrite(RELE4, LOW);
      M4 = LOW;
    }
   else if (M5 == HIGH  && ESTADOTELONUP == LOW && ESTADOTELONDOWN ==
LOW && ESTADOPLATUP == LOW
   && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI ==
LOW && ESTADOGIRA5 == LOW){
      digitalWrite(RELE5, LOW);
      M5 = LOW;
    }
   else if ((digitalRead(FINALCARRERAUP)) == HIGH &&
(digitalRead(FINALCARRERADOWN)) == LOW && ESTADOTELONUP == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE2, LOW);
      digitalWrite(RELE4, LOW);
      ESTADOTELONUP = LOW;
      }
   else if ((digitalRead(FINALCARRERAUP)) == LOW &&
(digitalRead(FINALCARRERADOWN)) == HIGH && ESTADOTELONDOWN == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE4, LOW);
      ESTADOTELONDOWN = LOW;
      }
   else if (DISTANCIA >= 11 && (digitalRead(FINALCARRERAPLAT) == LOW)
&& ESTADOPLATUP == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE2, LOW);
      digitalWrite(RELE3, LOW);
      digitalWrite(RELE5, LOW);
      ESTADOPLATUP = LOW;
      }
   else if (digitalRead(FINALCARRERAPLAT) == HIGH && ESTADOPLATDOWN ==
HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE3, LOW);
      digitalWrite(RELE5, LOW);
      ESTADOPLATDOWN = LOW;
      }
   else if ((millis() >= TiempoAhora2 + periodo2) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRA == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE3, LOW);
      digitalWrite(RELE4, LOW);
      digitalWrite(RELE5, LOW);
      ESTADOGIRA = LOW;
      }
   else if ((millis() >= TiempoAhora3 + periodo3) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRAANTI == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE2, LOW);
      digitalWrite(RELE3, LOW);
      digitalWrite(RELE4, LOW);
      digitalWrite(RELE5, LOW);
      ESTADOGIRAANTI = LOW;
      }
   else if ((millis() >= TiempoAhora4 + periodo4) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRA5 == HIGH){
```

```
      CONTADORGIRO++;
      TiempoAhora4 = millis();
      }
   else if (CONTADORGIRO == 6 && (digitalRead(FINALCARRERAGIRA) ==
HIGH) && ESTADOGIRA5 == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE3, LOW);
      digitalWrite(RELE4, LOW);
      digitalWrite(RELE5, LOW);
      ESTADOGIRA5 = LOW;
      CONTADORGIRO = 0;
      }
  }

  else if ((digitalRead(MANUAL) == HIGH) && (digitalRead(AUTO) == LOW)
&& ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW // Activa el modo
manual, pero no se activará si hay un proceso automático en marcha.
Cuando este finalice, podrá entrar.
  && ESTADOPLATUP == LOW && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW
&& ESTADOGIRAANTI == LOW && ESTADOGIRA5 == LOW){
    pantallaManual();
    paroMarcha();
    avanceRetroceso();
    velocidad1();
    velocidad2();
    rampasMotor();
  }

  else if ((digitalRead(MANUAL) == LOW) && (digitalRead(AUTO) ==
HIGH)){ // Activa el modo automático y desactiva todos los relés si
estan activos (del modo manual)
    if (M1 == HIGH){
      digitalWrite(RELE1, LOW);
      M1 = LOW;
    }
    else if (M2 == HIGH){
      digitalWrite(RELE2, LOW);
      M2 = LOW;
    }
    else if (M3 == HIGH){
      digitalWrite(RELE3, LOW);
      M3 = LOW;
    }
    else if (M4 == HIGH){
      digitalWrite(RELE4, LOW);
      M4 = LOW;
    }
    else if (M5 == HIGH){
      digitalWrite(RELE5, LOW);
      M5 = LOW;
    }
    pantallaAuto();
    subirTelon();
    bajarTelon();
    subirPlataforma();
    bajarPlataforma();
    giraHorario();
    giraAntihorario();
    giraCinco();
  }
```

```
  else if ((digitalRead(MANUAL) == HIGH) && (digitalRead(AUTO) ==
HIGH)){  // Si ambos modos se activan, nos aparece un error y se
desactiva todas las salidas de golpe. Puede servir como reset.
   digitalWrite(RELE1, LOW);
   digitalWrite(RELE2, LOW);
   digitalWrite(RELE3, LOW);
   digitalWrite(RELE4, LOW);
   digitalWrite(RELE5, LOW);
   ESTADOTELONUP = LOW;
   ESTADOTELONDOWN = LOW;
   ESTADOPLATUP = LOW;
   ESTADOPLATDOWN = LOW;
   ESTADOGIRA = LOW;
   ESTADOGIRAANTI = LOW;
   ESTADOGIRA5 = LOW;
   M1 = LOW;
   M2 = LOW;
   M3 = LOW;
   M4 = LOW;
   M5 = LOW;
   lcd.setCursor(0,0);
   lcd.print("ERROR SELECCIONA");
   lcd.setCursor(0,1);
   lcd.print("   UNA OPCION   ");
  }
}

void encoder2(){  // Incrementa o disminuye la posición en la que se
encuentra el encoder (Auto)
  static unsigned long ultimaInterrupcion2 = 0;
  unsigned long tiempoInterrupcion2 = millis();

  if (tiempoInterrupcion2 - ultimaInterrupcion2 > 5){

  if (digitalRead(D) == HIGH){
    POSICION2++;

  }
  else {
    POSICION2--;
  }

  POSICION2 = min(7, max(0, POSICION2));

  ultimaInterrupcion2 = tiempoInterrupcion2;
 }
}

void subirTelon(){  // Función que nos activa y desactiva salidas para
poder subir el telón

  if ((digitalRead(PULSADOR2)) == LOW && POSICION2 == 1 &&
(digitalRead(FINALCARRERADOWN)) == HIGH &&
(digitalRead(FINALCARRERAUP)) == LOW
  && ESTADOTELONDOWN == LOW && ESTADOPLATUP == LOW && ESTADOPLATDOWN
== LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA5 ==
LOW){
      digitalWrite(RELE1, HIGH);
      digitalWrite(RELE2, HIGH);
      digitalWrite(RELE4, HIGH);
      ESTADOTELONUP = HIGH;
```

```
  }
  else if ((digitalRead(FINALCARRERAUP)) == HIGH &&
(digitalRead(FINALCARRERADOWN)) == LOW && ESTADOTELONUP == HIGH){ //
HAY QUE PONER 2 FINALES DE CARRERA!!!!!! O inventarme 2 estados, como
si fueran 2 finales de carrera
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE2, LOW);
      digitalWrite(RELE4, LOW);
      ESTADOTELONUP = LOW;
   }
  else if ((digitalRead(FINALCARRERAUP)) == HIGH &&
(digitalRead(FINALCARRERADOWN)) == HIGH){
      digitalWrite(RELE1, LOW);
  }
}

void bajarTelon(){  // Función que nos activa y desactiva salidas para
poder bajar el telón

  if ((digitalRead(PULSADOR2)) == LOW && POSICION2 == 2 &&
(digitalRead(FINALCARRERADOWN)) == LOW &&
(digitalRead(FINALCARRERAUP)) == HIGH &&
  ESTADOTELONUP == LOW && ESTADOPLATUP == LOW && ESTADOPLATDOWN == LOW
&& ESTADOGIRA == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA5 ==
LOW){  //Copiar primer código del Void de manual y el segundo
cambiarlo por desactivación si se pulsa Final de carrera
      digitalWrite(RELE1, HIGH);
      digitalWrite(RELE4, HIGH);
      ESTADOTELONDOWN = HIGH;
  }
  else if ((digitalRead(FINALCARRERAUP)) == LOW &&
(digitalRead(FINALCARRERADOWN)) == HIGH && ESTADOTELONDOWN == HIGH){
      digitalWrite(RELE1, LOW);
      digitalWrite(RELE4, LOW);
      ESTADOTELONDOWN = LOW;
   }
  else if ((digitalRead(FINALCARRERAUP)) == HIGH &&
(digitalRead(FINALCARRERADOWN)) == HIGH){
      digitalWrite(RELE1, LOW);
  }
}

void subirPlataforma(){  // Función que activa y desactiva las salidas
para elevar la plataforma

  if (POSICION2 == 3 && digitalRead(FINALCARRERAPLAT) == HIGH &&
digitalRead(PULSADOR2) == LOW &&
  ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW && ESTADOPLATDOWN ==
LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA5 ==
LOW){
    digitalWrite(RELE1, HIGH);
    digitalWrite(RELE2, HIGH);
    digitalWrite(RELE3, HIGH);
    digitalWrite(RELE5, HIGH);
    ESTADOPLATUP = HIGH;
  }
  else if (DISTANCIA >= 11 && (digitalRead(FINALCARRERAPLAT) == LOW)
&& ESTADOPLATUP == HIGH){  // Aquí determinamos la altura máxima a la
que queremos que se eleve la plataforma. Max 40cm
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE2, LOW);
```

```
    digitalWrite(RELE3, LOW);
    digitalWrite(RELE5, LOW);
    ESTADOPLATUP = LOW;
  }
}

void bajarPlataforma(){  // Función que activa y desactiva las salidas
para bajar la plataforma

  if (POSICION2 == 4 && (digitalRead(FINALCARRERAPLAT) == LOW) &&
digitalRead(PULSADOR2) == LOW &&
  ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW && ESTADOPLATUP ==
LOW && ESTADOGIRA == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA5 ==
LOW){
    digitalWrite(RELE1, HIGH);
    digitalWrite(RELE3, HIGH);
    digitalWrite(RELE5, HIGH);
    ESTADOPLATDOWN = HIGH;
  }
  else if (digitalRead(FINALCARRERAPLAT) == HIGH && ESTADOPLATDOWN ==
HIGH){
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE3, LOW);
    digitalWrite(RELE5, LOW);
    ESTADOPLATDOWN = LOW;
  }
}

void giraHorario(){  // Activa la plataforma y la hace girar hasta que
detecta el final de carrera. Añadir un pequeño resorte en la
plataforma para que active el final de carrera.

  if (POSICION2 == 5 && digitalRead(FINALCARRERAGIRA) == HIGH &&
digitalRead(PULSADOR2) == LOW &&
  ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW && ESTADOPLATUP ==
LOW && ESTADOPLATDOWN == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA5
== LOW){
    digitalWrite(RELE1, HIGH);
    digitalWrite(RELE3, HIGH);
    digitalWrite(RELE4, HIGH);
    digitalWrite(RELE5, HIGH);
    ESTADOGIRA = HIGH;
    TiempoAhora2 = millis();
  }
  else if ((millis() >= TiempoAhora2 + periodo2) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRA == HIGH){  //
Tiempo de seguridad para que mientras esta arrancando el motor, no se
desactive porque el final de carrera sigue activo, podemos variar este
tiempo.
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE3, LOW);
    digitalWrite(RELE4, LOW);
    digitalWrite(RELE5, LOW);
    ESTADOGIRA = LOW;
  }
}

void giraAntihorario(){  // Activa la plataforma y la hace girar hasta
que detecta el final de carrera. Añadir un pequeño resorte en la
plataforma para que active el final de carrera. Sentido antihorario
```

```
  if (POSICION2 == 6 && digitalRead(FINALCARRERAGIRA) == HIGH &&
digitalRead(PULSADOR2) == LOW &&
  ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW && ESTADOPLATUP ==
LOW && ESTADOPLATDOWN == LOW && ESTADOGIRA == LOW && ESTADOGIRA5 ==
LOW){
    digitalWrite(RELE1, HIGH);
    digitalWrite(RELE2, HIGH);
    digitalWrite(RELE3, HIGH);
    digitalWrite(RELE4, HIGH);
    digitalWrite(RELE5, HIGH);
    ESTADOGIRAANTI = HIGH;
    TiempoAhora3 = millis();

  }
  else if ((millis() >= TiempoAhora3 + periodo3) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRAANTI ==
HIGH){  // Tiempo de seguridad para que mientras esta arrancando el
motor, no se desactive porque el final de carrera sigue activo,
podemos variar este tiempo.
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE2, LOW);
    digitalWrite(RELE3, LOW);
    digitalWrite(RELE4, LOW);
    digitalWrite(RELE5, LOW);
    ESTADOGIRAANTI = LOW;
  }
}

void giraCinco(){ // Activa la plataforma y la hace girar 5 veces en
sentido horario.

  if (POSICION2 == 7 && digitalRead(FINALCARRERAGIRA) == HIGH &&
digitalRead(PULSADOR2) == LOW &&
  ESTADOTELONUP == LOW && ESTADOTELONDOWN == LOW && ESTADOPLATUP ==
LOW && ESTADOPLATDOWN == LOW && ESTADOGIRAANTI == LOW && ESTADOGIRA ==
LOW){
    digitalWrite(RELE1, HIGH);
    digitalWrite(RELE3, HIGH);
    digitalWrite(RELE4, HIGH);
    digitalWrite(RELE5, HIGH);
    ESTADOGIRA5 = HIGH;
  }
  else if ((millis() >= TiempoAhora4 + periodo4) &&
(digitalRead(FINALCARRERAGIRA) == HIGH) && ESTADOGIRA5 == HIGH){  //
Tiempo de seguridad para que mientras esta arrancando el motor, no se
desactive porque el final de carrera sigue activo, podemos variar este
tiempo.
    CONTADORGIRO++;
    TiempoAhora4 = millis();
  }
  else if (CONTADORGIRO == 6 && (digitalRead(FINALCARRERAGIRA) ==
HIGH) && ESTADOGIRA5 == HIGH){
    digitalWrite(RELE1, LOW);
    digitalWrite(RELE3, LOW);
    digitalWrite(RELE4, LOW);
    digitalWrite(RELE5, LOW);
    ESTADOGIRA5 = LOW;
    CONTADORGIRO = 0;
  }
}
```

```
void pantallaAuto(){  // Determina los textos que aparecen pro
pantalla dependiendo de la posición del encoder automático
  if(POSICION2 == 1){
   lcd.setCursor(0,0);
   lcd.print("     SUBIR      ");
   lcd.setCursor(0,1);
   lcd.print("     TELON      ");
  }

  else if(POSICION2 == 2){
   lcd.setCursor(0,0);
   lcd.print("     BAJAR      ");
   lcd.setCursor(0,1);
   lcd.print("     TELON      ");
  }

  else if(POSICION2 == 3){
   lcd.setCursor(0,0);
   lcd.print("     SUBIR      ");
   lcd.setCursor(0,1);
   lcd.print("   PLATAFORMA   ");
  }

   else if(POSICION2 == 4){
   lcd.setCursor(0,0);
   lcd.print("     BAJAR      ");
   lcd.setCursor(0,1);
   lcd.print("   PLATAFORMA   ");
  }

  else if(POSICION2 == 5){
   lcd.setCursor(0,0);
   lcd.print(" BASE GIRATORIA ");
   lcd.setCursor(0,1);
   lcd.print("SENTIDO HORARIO ");
  }

  else if(POSICION2 == 6){
   lcd.setCursor(0,0);
   lcd.print(" BASE GIRATORIA ");
   lcd.setCursor(0,1);
   lcd.print("SENTIDO AHORARIO");
  }

   else if(POSICION2 == 7){
   lcd.setCursor(0,0);
   lcd.print(" BASE GIRATORIA ");
   lcd.setCursor(0,1);
   lcd.print("    5 GIROS     ");
  }

   else if(POSICION2 == 0){
   lcd.setCursor(0,0);
   lcd.print("      MODO      ");
   lcd.setCursor(0,1);
   lcd.print("   AUTOMATICO   ");
  }

  else if(digitalRead(D) == HIGH){
    lcd.clear();
  }
```

```
}

void encoder(){  // Incrementa o disminuye la posición en la que se
encuentra el encoder (Manual)
  static unsigned long ultimaInterrupcion = 0;
  unsigned long tiempoInterrupcion = millis();

  if (tiempoInterrupcion - ultimaInterrupcion > 5){

  if (digitalRead(B) == HIGH){
    POSICION++;

  }
  else {
    POSICION--;
  }

  POSICION = min(6, max(0, POSICION));

  ultimaInterrupcion = tiempoInterrupcion;
}
}

void paroMarcha(){  // Activa o desactiva el PARO  o el MARCHA
  int estadoRele1 = digitalRead(RELE1);

  if (digitalRead(PULSADOR) == LOW && ESTADO1 != HIGH && POSICION ==
1){
    digitalWrite(RELE1, !estadoRele1);
    ESTADO1 = HIGH;
    M1 = HIGH;
  }
  else if (digitalRead(PULSADOR) == HIGH && ESTADO1 != LOW && POSICION
== 1){
    ESTADO1 = LOW;
  }
}

void avanceRetroceso(){  // Activa o desactiva el AVANCE  o el
RETROCESO
  int estadoRele2 = digitalRead(RELE2);

  if (digitalRead(PULSADOR) == LOW && ESTADO2 != HIGH && POSICION ==
2){
    digitalWrite(RELE2, !estadoRele2);
    ESTADO2 = HIGH;
    M2 = HIGH;
  }
  else if (digitalRead(PULSADOR) == HIGH && ESTADO2 != LOW && POSICION
== 2){
    ESTADO2 = LOW;
  }
}

void velocidad1(){  // Activa o desactiva Velocidad 1
  int estadoRele3 = digitalRead(RELE3);

  if (digitalRead(PULSADOR) == LOW && ESTADO3 != HIGH && POSICION ==
3){
    digitalWrite(RELE3, !estadoRele3);
```

```
    ESTADO3 = HIGH;
    M3 = HIGH;
  }
  else if (digitalRead(PULSADOR) == HIGH && ESTADO3 != LOW && POSICION
== 3){
    ESTADO3 = LOW;
  }
}

void velocidad2(){  // Activa o desactiva Velocidad 2
  int estadoRele4 = digitalRead(RELE4);

  if (digitalRead(PULSADOR) == LOW && ESTADO4 != HIGH && POSICION ==
4){
    digitalWrite(RELE4, !estadoRele4);
    ESTADO4 = HIGH;
    M4 = HIGH;
  }
  else if (digitalRead(PULSADOR) == HIGH && ESTADO4 != LOW && POSICION
== 4){
    ESTADO4 = LOW;
  }
}

void rampasMotor(){  // Activa o desactiva Rampa de arranque 1 o 2
  int estadoRele5 = digitalRead(RELE5);

  if (digitalRead(PULSADOR) == LOW && ESTADO5 != HIGH && POSICION ==
5){
    digitalWrite(RELE5, !estadoRele5);
    ESTADO5 = HIGH;
    M5 = HIGH;
  }
  else if (digitalRead(PULSADOR) == HIGH && ESTADO5 != LOW && POSICION
== 5){
    ESTADO5 = LOW;
  }
}

void pantallaManual(){  // Determina los textos que aparecen pro
pantalla dependiendo de la posición del encoder manual
  if(POSICION == 1){
   lcd.setCursor(0,0);
   lcd.print("    FUNCION     ");
   lcd.setCursor(0,1);
   lcd.print("  PARO/MARCHA   ");
  }

  else if(POSICION == 2){
   lcd.setCursor(0,0);
   lcd.print("    FUNCION     ");
   lcd.setCursor(0,1);
   lcd.print("AVANCE/RETROCESO");
  }

  else if(POSICION == 3){
   lcd.setCursor(0,0);
   lcd.print("    MOTOR EN    ");
   lcd.setCursor(0,1);
   lcd.print("   VELOCIDAD 1  ");
  }
```

```
  else if(POSICION == 4){
  lcd.setCursor(0,0);
  lcd.print("    MOTOR EN    ");
  lcd.setCursor(0,1);
  lcd.print("   VELOCIDAD 2  ");
 }

   else if(POSICION == 5){
  lcd.setCursor(0,0);
  lcd.print(" ARRANQUE MOTOR ");
  lcd.setCursor(0,1);
  lcd.print("RAMPA 1/ RAMPA 2");
 }

  else if(POSICION == 0){
  lcd.setCursor(0,0);
  lcd.print("      MODO      ");
  lcd.setCursor(0,1);
  lcd.print("     MANUAL     ");
 }

 else if (POSICION == 6){
  lcd.setCursor(0,0);
  lcd.print("PARA VELOCIDAD 3");
  lcd.setCursor(0,1);
  lcd.print("ACTIVAR V1 Y V2 ");
 }
 else if(digitalRead(B) == HIGH){
   lcd.clear();
 }

}
```