

Alcoa Youth Talent Competition

**Panel de entrenamiento modular para
la automatización y control de procesos**

Módulo de entrenamiento: Motor de explosión

**Desarrollado por Diego Bercial Argüelles, Marta Caldero Iglesias,
Diego Canteli Suárez, Sergio Martínez Olivar, Adrián Rojo Fernández**

2º Bachillerato

Colegio Corazón de María (Gijón)

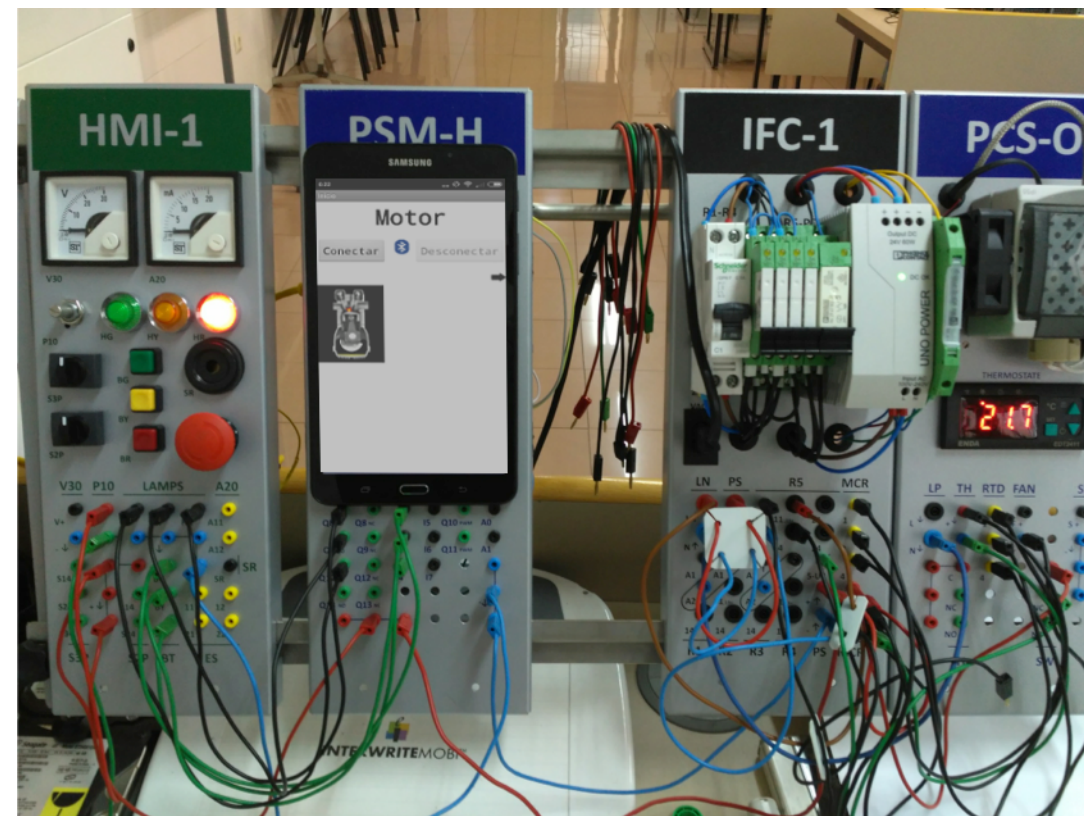


PRÁCTICAS DE AUTOMATIZACIÓN Y CONTROL

MÓDULOS DE PANEL DE ENTRENAMIENTO

Objetivos

- Proyecto de automatización (Motor)
- Control mediante Arduino UNO
- Señales E/S digitales
- Cableado de señales de mando y fuerza
- Programación en IDE Arduino.
- Control mediante placa Arduino.
- Programación de aplicación para Smartphone Android con AppInventor.
- Descarga, depuración y puesta en marcha.



Hardware y Software

Modulo entrenador en automatización:

- Módulo HMI-1: Interface de usuario.
- Módulo IFC-1: Interface de señales (Fuente de alimentación).
- Módulo PSM-H: Incluye Arduino UNO, placa interface de señal y placa con 4 relés para gestión de salidas.
- Smartphone Android

Software IDE Arduino.

AppInventor

Hardware y Software

Modulo HMI-1: Interface de usuario

- Se utiliza el pulsador Verde (acelerador).
- Se utiliza el pulsador Rojo (freno).
- Se utiliza el interruptor de 2 posiciones (arranque/apagado).
- Se utilizan los pilotos luminosos (Verde y rojo) para indicar motor arrancado y frenado del motor.

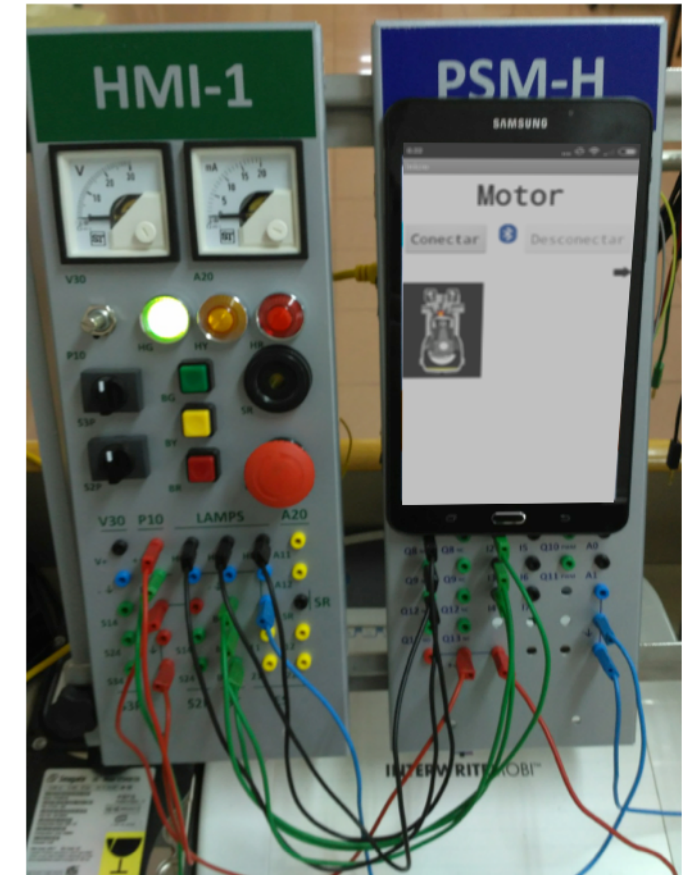
* Desarrollo posterior



Hardware y Software

Modulo PSM-H: Control de proceso

- Se conecta el puerto con el pulsador verde
 - Se conecta el puerto con el pulsador rojo
 - Se conecta el puerto con el interruptor de 2 posiciones
 - Se conectan los puertos con los pilotos luminosos
- * Desarrollo posterior



Hardware y Software

Smartphone Arduino

- Posee el software de simulación del proceso.
- Simula un motor de explosión que aumenta sus revoluciones al ser acelerado.



Desarrollo

Se trata de simular el funcionamiento de un motor de explosión mediante la conexión Bluetooth entre un Arduino y una aplicación realizada con App Inventor. Las funciones simuladas son las siguientes:

- 1)Aceleración (Botón “Verde” en el módulo de Interface). Mantener pulsado el botón acelera el motor. Si se suelta el pulsador el motor se desacelera.
- 2)Freno (Botón “Rojo” en el módulo de Interface). Mantener pulsado el botón frena el motor.
- 3)Arranque del motor (Conmutador de 2 posiciones en el módulo de Interface). Una posición comunica arranque del motor, la otra posición comunica apagado del motor.
- 4)Pilotos luminosos (Rojo y verde). Se enciende el rojo cuando se pulsa el freno, el verde cuando el motor está arrancado.

* Desarrollo posterior

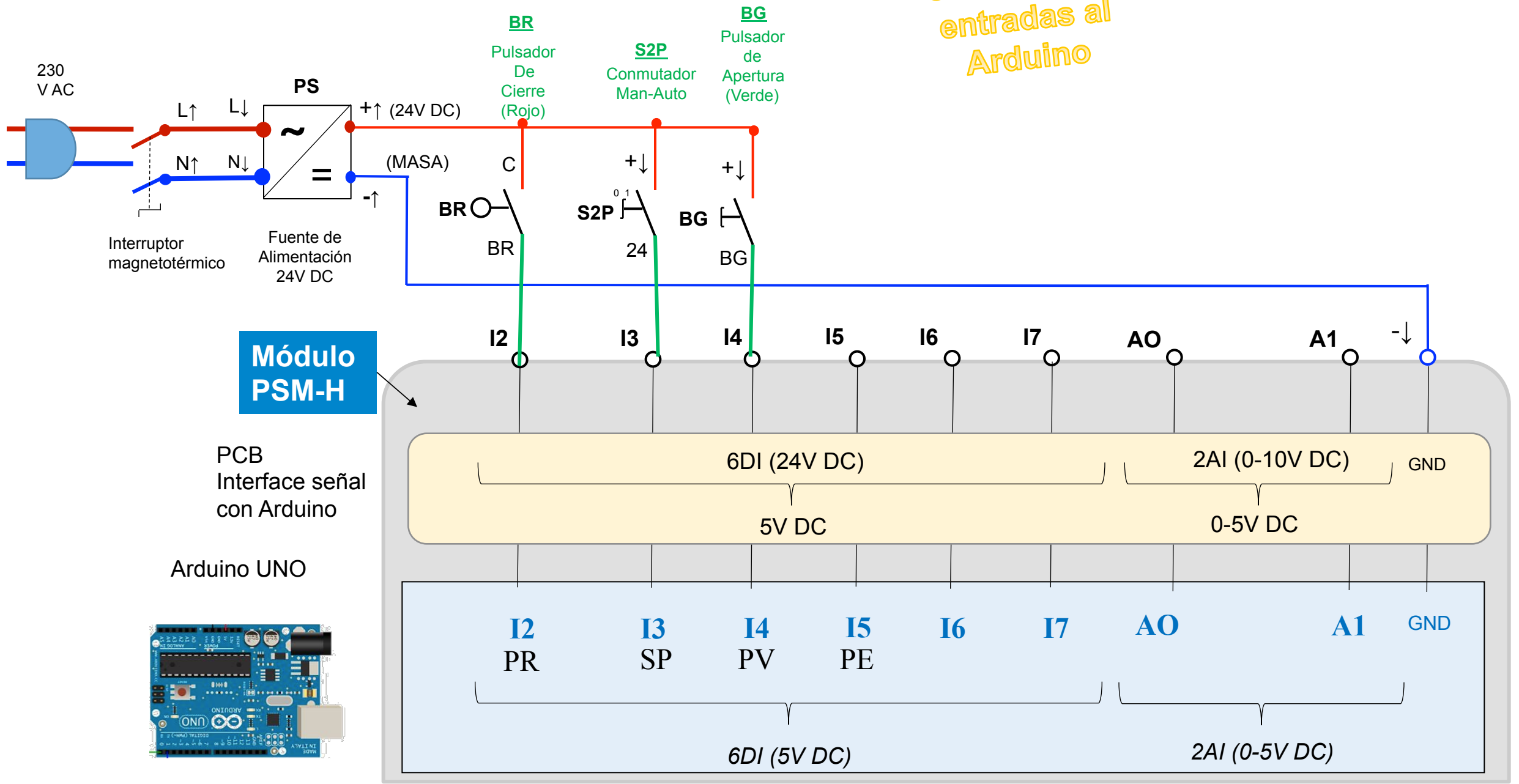
Comunicación entre Arduino y Smartphone

El sistema de control y el sistema de simulación se comunican vía Bluetooth enviando diversos código de estado del sistema:

- 1)Se envía el código 'A1' a la App del Smartphone para indicar que se ha pulsado el botón Verde, se deben subir las revoluciones del motor y por tanto, el motor irá acelerando.
- 2)Se envía el código 'A0' a la App del Smartphone para indicar que se no ha pulsado el botón Verde, se deben bajar las revoluciones del motor y por tanto, el motor irá decelerando.
- 3)Se envía el código 'R' y el valor de las revoluciones que tiene el motor al programa de control de Arduino para que el programa de control sepa si tiene que cambiar de marcha.
- 4)Se envía el código 'M' y el número de marcha a la App del Smartphone para indicar que marcha se debe utilizar de manera que el funcionamiento del motor sea correcto y pueda seguir aumentando su velocidad.
- 5)Se envía el código 'F' a la App del Smartphone para indicar que el pulsador del freno ha sido accionado y se deba bajar su velocidad.
- 6)Se envía el código 'K1' a la App del Smartphone para indicar que se girado el counmutador para indicar el arranque del motor. Si se gira en sentido contrario se envía el código 'K0' a la App del Smartphone para indicar el apagado del motor.

* Desarrollo posterior

Controlador: ARDUINO UNO



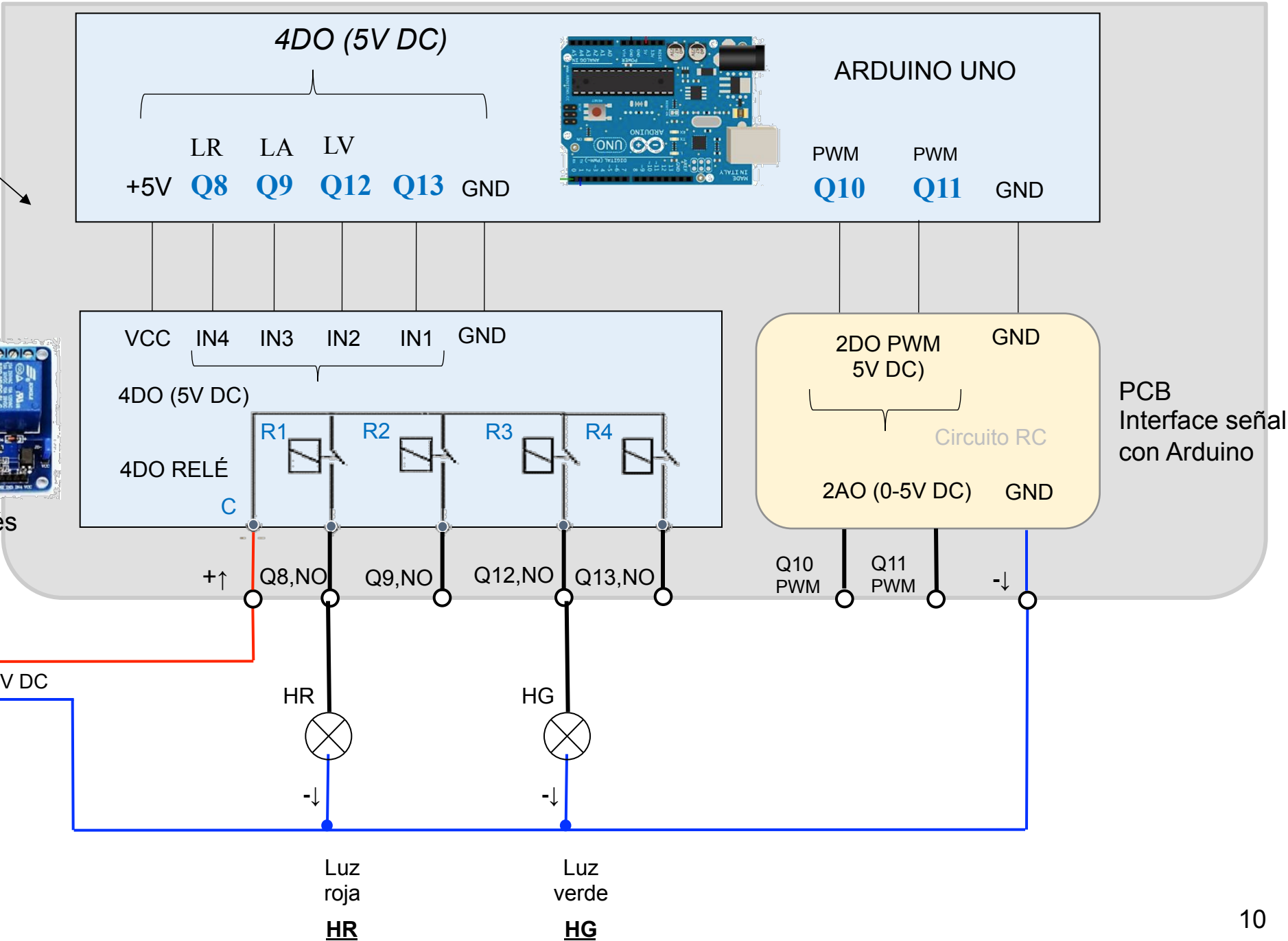
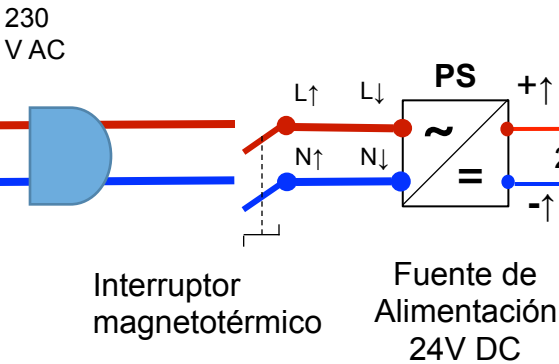
Controlador
ARDUINO UNO

Conexión de
salidas al
Arduino

Módulo
PSM-H



Módulo 4 relés



```

#include "Arduino.h"
#include <SoftwareSerial.h>
char var;
String inString = "";
bool M = 1; //marcha
int rpm = 0;
int rpm_rueda = 0;
int buttonstate;
const int boton = 4;

void setup()
{
  Serial.begin(9600);
  delay(3000);
  pinMode(boton, INPUT);
}

void loop()
{
  buttonstate=digitalRead(boton);
  if(buttonstate == HIGH){
    Serial.println("A1");
  }
  else{
    Serial.println("A0");
  }
}

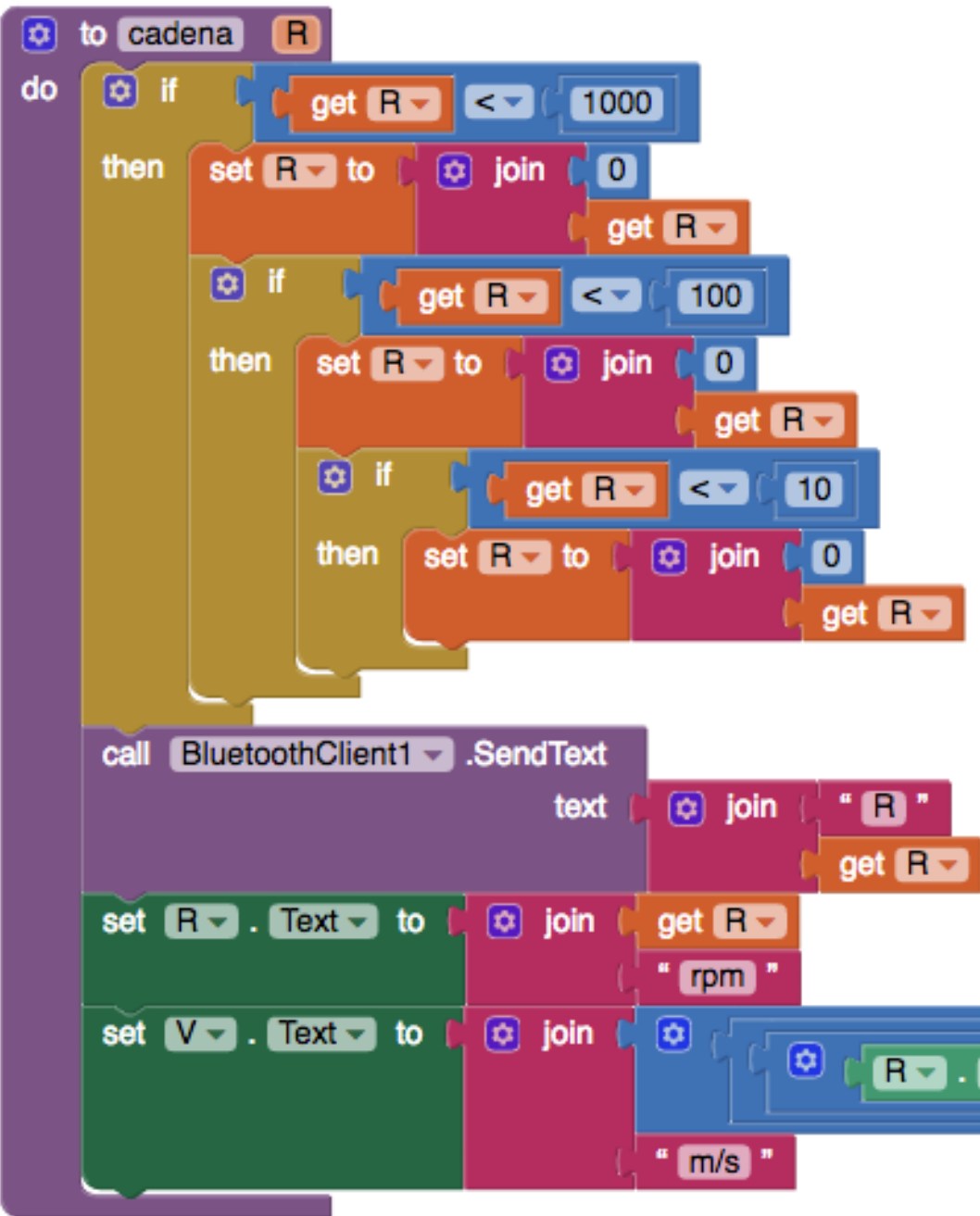
```

```

if (Serial.available()>0 )
{
  var = Serial.read();
  if(var == 'R')
  {
    while (Serial.available() > 0) {
      int var = Serial.read();
      if (isDigit(var)) {
        inString += (char)var;
      }
    }
    rpm = inString.toInt();
  }
  if (rpm > 2000)
  {
    if(M < 4){
      M = M + 1;
    }
    else{
      M = 4;
    }

    Serial.println('M',M);
  }
  if(rpm < 750)
  {
    if(M > 0)
    {
      M = M - 1;
    }
    else{
      M = 0;
    }
    Serial.println('M',M);
  }
}
}
}

```



0:22

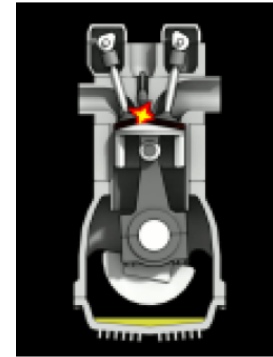
Inicio

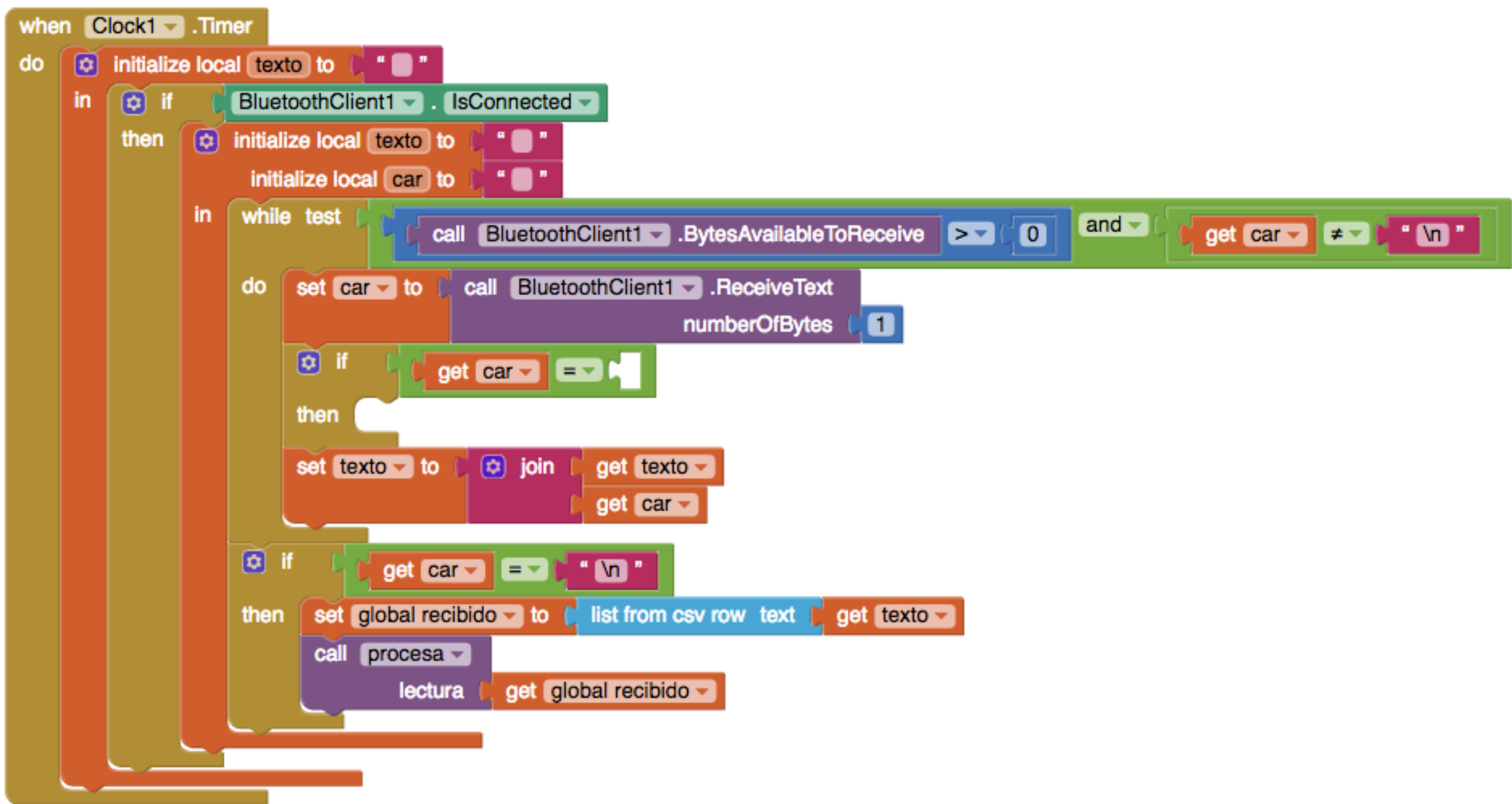
Motor

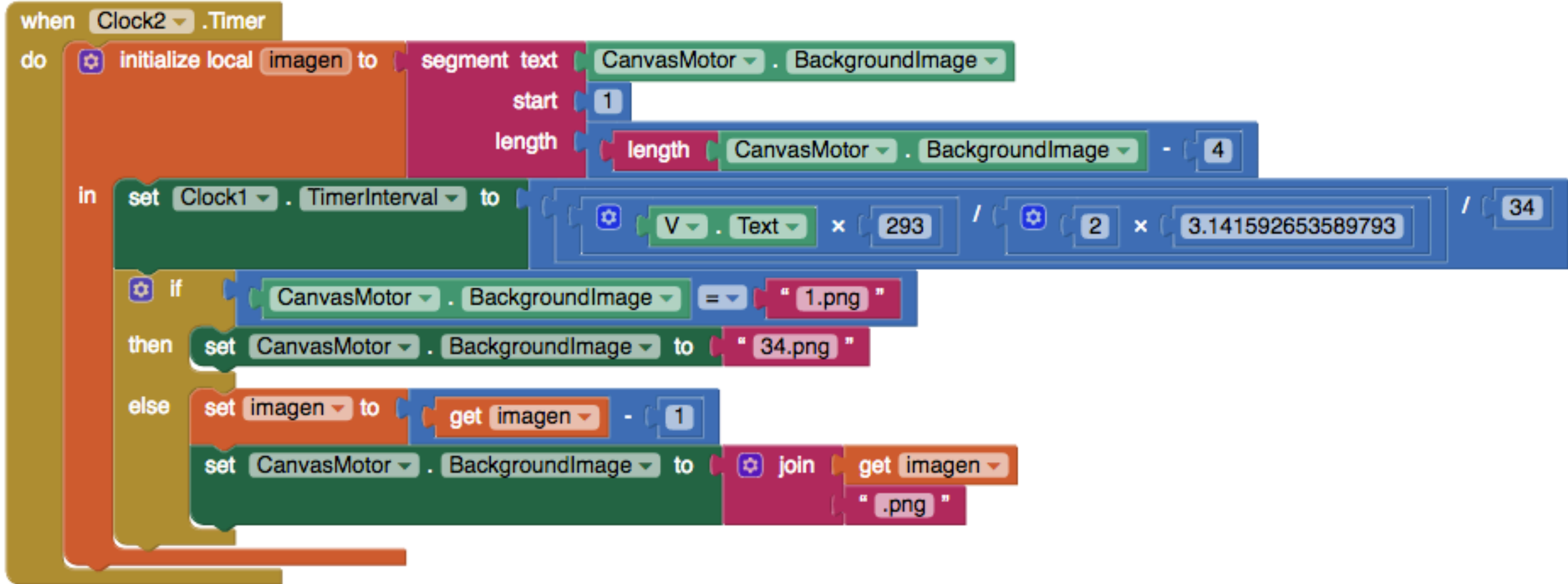
Conectar



Desconectar







initialize global recibido to create empty list

to habilitar_botones
do set Desconectar . Enabled to true

to desahabilitar_botones
do set Desconectar . Enabled to false


```
when Conectar .AfterPicking
do
  if
    call BluetoothClient1 .Connect
      address Conectar . Selection
    then
      call Notifier1 .ShowAlert
        notice " Bluetooth conectado "
      call habilitar_botones
```

```
when Conectar .BeforePicking
do
  set Conectar . Elements to BluetoothClient1 . AddressesAndNames
```

```
when Desconectar .Click
do
  call BluetoothClient1 .Disconnect
  call Notifier1 .ShowAlert
    notice " Bluetooth desconectado "
  call desahabilitar_botones
```

```
to envia car
do
  if
    BluetoothClient1 . IsConnected
    then
      call BluetoothClient1 .SendText
        text get car
    else
      call Notifier1 .ShowAlert
        notice " Error bluetooth: debes activar el bluetooth para..."
```

```
when Screen1.Initialize
do
  call desahibitar_botones
  set Flecha_I.Visible to false
  set HA6.Visible to false
  if BluetoothClient1.IsConnected = false
  then
    call Notifier1.ShowAlert
    notice "Error Bluetooth: Debes conectar el Bluetooth par..."
    set BluetoothClient1.DelimiterByte to 10
```

```
to procesa lectura
do
  if select list item list index 1 get lectura = "M"
  then
    set M.Text to join select list item list index 2 get lectura
    "/4"
  if select list item list index 1 get lectura = "A"
  then
    set M.Text to select list item list index 2 get lectura
```

```
when Flecha_D.Click
do
  set Flecha_D.Visible to false
  set Flecha_I.Visible to true
  set HA1.Visible to false
  set HA2.Visible to false
  set HA3.Visible to false
  set HA4.Visible to false
  set HA5.Visible to false
```