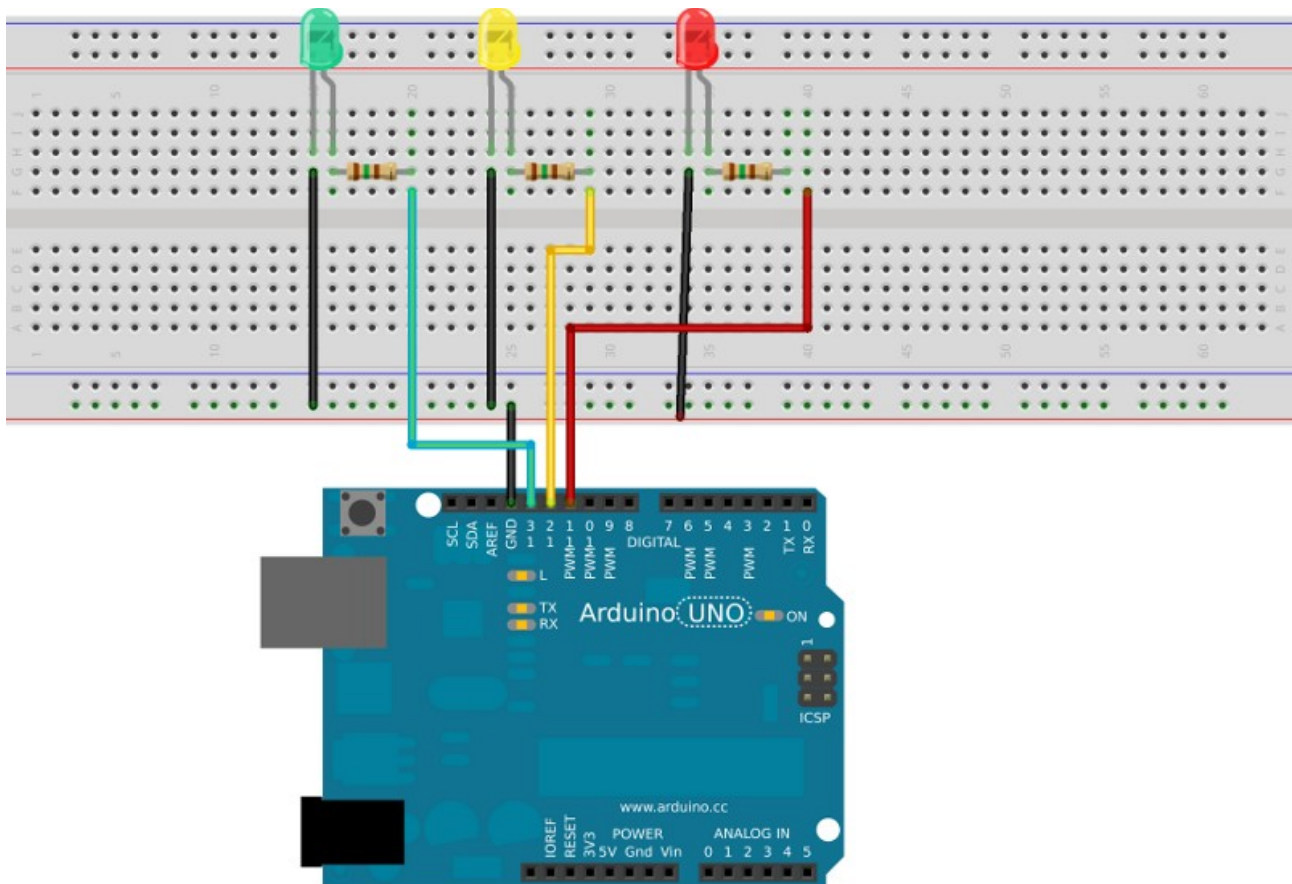


# Dossier de solucions

## 2- Blink

a) **Podeu provar a fer un semàfor afegint dos LEDs més, un de taronja i un de vermell. Heu de fer que s'encenguin i s'apaguin un rere l'altre.**

Com ja sabem, els LEDs han d'anar connectats a un pin digital amb una resistència de 150 ohms. Per exemple, podem fer servir els pins digitals 12 i 11, i quedaria quelcom així:



Made with  Fritzing.org

El codi crea una constant per cada pin, les estableix com a sortides (OUTPUT) i les encén en diferents intervals de temps:

```
#define VERD 13
#define GROC 12
#define VERMELL 11

void setup()
{
  pinMode(VERD, OUTPUT);
  pinMode(GROC, OUTPUT);
  pinMode(VERMELL, OUTPUT);
}

void loop()
{
  digitalWrite(VERD, HIGH);
  delay(2000);
  digitalWrite(VERD, LOW);
```

```
    digitalWrite(GROC, HIGH);  
    delay(1000);  
    digitalWrite(GROC, LOW);  
    digitalWrite(VERMELL, HIGH);  
    delay(2000);  
    digitalWrite(VERMELL, LOW);  
}
```

**b) Guardant el temps dins una variable, podeu fer que a cada volta del loop el temps d'encesa del LED sigui el doble que l'anterior.**

Farem servir el circuit original del Blink amb un sol LED, i treballarem a partir del codi d'exemple del dossier.

```
#define LED 13 //Constant per guardar el pin del led  
  
int temps = 50; //Creem una variable per guardar el temps  
void setup()  
{  
    pinMode(LED, OUTPUT); //Establim el pin 13 com a sortida  
}  
void loop()  
{  
    digitalWrite(LED, HIGH); //Engeguem  
    delay(temps); //Esperem 0'5 segons (1000 = 1s)  
    digitalWrite(LED, LOW); //Apaguem  
    delay(temps);  
    temps = temps*2; //Multiplica el valor de temps per dos  
}
```

### 3- Blink amb pulsador

a) Afegiu un else, de tal manera que si no es prem el pulsador tots els LEDs estiguin encesos.

Com l'enunciat indica, només ens cal modificar el codi.

```
#define BOT0 7
#define VERD 13
#define GROC 11
#define VERMELL 8

int lectura; //Variable on guardarem el valor del boto (HIGH o LOW)

void setup()
{
    //Configurem els pins
    pinMode(BOT0, INPUT); //Establím el pin del boto com a entrada
    pinMode(VERD, OUTPUT); //Els pins dels LEDs seran sortides
    pinMode(GROC, OUTPUT);
    pinMode(VERMELL, OUTPUT);
}

void loop()
{
    lectura = digitalRead(BOT0); //Llegim el pin BOT0 i guardem el seu valor
    if(lectura == HIGH) //Si el valor del pin BOT0 es HIGH...
    {
        //... escala de color!!! :D
        digitalWrite(VERD, HIGH);
        delay(100);
        digitalWrite(VERD, LOW);
        digitalWrite(GROC, HIGH);
        delay(100);
        digitalWrite(GROC, LOW);
        digitalWrite(VERMELL, HIGH);
        delay(100);
        digitalWrite(VERMELL, LOW);
    }

    //=====PART NOVA=====
    else
    {
        digitalWrite(VERD, HIGH);
        digitalWrite(GROC, HIGH);
        digitalWrite(VERMELL, HIGH);
    }
}
```

b) Canvieu el codi original de manera que:

- Si es prem una vegada el pulsador s'encengui el LED verd
- Si es prem dues vegades s'encengui el LED groc
- Si es prem tres vegades s'encengui el LED vermell
- Si es prem quatre vegades s'apaguin tots
- Si es prem cinc vegades torni a començar (podeu fer servir else if(condició))

En aquest cas tampoc necessitem fer cap tipus de canvi en el circuit original, només modificarem el codi.

En primer lloc crearem una variable per guardar el nombre de vegades que es prem el pulsador. Aquesta variable començarà estant a 0, i cada vegada que premem el botó li sumarem 1. Quan arribi a 5, la posarem a 0.

```
#define BOTO 7
#define VERD 13
#define GROC 11
#define VERMELL 8

int lectura; //Variable on guardarem el valor del boto (HIGH o LOW)

int contador = 0;

void setup()
{
    //Configurem els pins
    pinMode(BOTO, INPUT); //Establim el pin del boto com a entrada
    pinMode(VERD, OUTPUT); //Els pins dels LEDs seran sortides
    pinMode(GROC, OUTPUT);
    pinMode(VERMELL, OUTPUT);
}

void loop()
{
    lectura = digitalRead(BOTO); //Llegim el pin BOTO i guardem el seu valor
    if(lectura == HIGH) //Si el valor del pin BOTO es HIGH...
    {
        contador++; //Sumem 1 al comptador
        delay(500);
    }

    else
    {}

    //Creem una condicio per a cada valor valid del comptador (1,2,3,4)
    if(contador == 1)
    {
        digitalWrite(VERD, HIGH);
        digitalWrite(GROC, LOW);
        digitalWrite(VERMELL, LOW);
    }
    else if(contador == 2)
    {
        digitalWrite(VERD, LOW);
        digitalWrite(GROC, HIGH);
        digitalWrite(VERMELL, LOW);
    }
    else if(contador == 3)
    {
        digitalWrite(VERD, LOW);
        digitalWrite(GROC, LOW);
        digitalWrite(VERMELL, HIGH);
    }
    else if(contador == 4)
```

```

    {
        digitalWrite(VERD, LOW);
        digitalWrite(GROC, LOW);
        digitalWrite(VERMELL, LOW);
    }
    else{}

    //Si el contador == 5, aleshores el posem a 0
    if(contador == 5)
    {
        contador = 1;
    }
}

```

També podem fer una versió avançada amb un bucli while, de tal manera que al mantenir premut el botó, no vagi avançant el comptador.

```

#define BOT0 7
#define VERD 13
#define GROC 11
#define VERMELL 8

int lectura; //Variable on guardarem el valor del boto (HIGH o LOW)

int contador = 0;

bool premut = false;

void setup()
{
    //Configurem els pins
    pinMode(BOT0, INPUT); //Establim el pin del boto com a entrada
    pinMode(VERD, OUTPUT); //Els pins dels LEDs seran sortides
    pinMode(GROC, OUTPUT);
    pinMode(VERMELL, OUTPUT);
}

void loop()
{
    premut = false;
    lectura = digitalRead(BOT0); //Llegim el pin BOT0 i guardem el seu valor
    while(lectura == HIGH) //Si el valor del pin BOT0 es HIGH...
    {
        //Mentre no es deixi anar el boto, es repetira aquest bucli
        premut = true;
        lectura = digitalRead(BOT0);
    }

    if(premut)
    {
        contador++;
    }
    else{}

    //Creem una condicio per a cada valor valid del comptador (1,2,3,4)

```

```

if(contador == 1)
{
  digitalWrite(VERD, HIGH);
  digitalWrite(GROC, LOW);
  digitalWrite(VERMELL, LOW);
}
else if(contador == 2)
{
  digitalWrite(VERD, LOW);
  digitalWrite(GROC, HIGH);
  digitalWrite(VERMELL, LOW);
}
else if(contador == 3)
{
  digitalWrite(VERD, LOW);
  digitalWrite(GROC, LOW);
  digitalWrite(VERMELL, HIGH);
}
else if(contador == 4)
{
  digitalWrite(VERD, LOW);
  digitalWrite(GROC, LOW);
  digitalWrite(VERMELL, LOW);
}
else{}

//Si el contador == 5, aleshores el posem a 0
if(contador == 5)
{
  contador = 1;
}
}

```

**EXTRA! Modifiqueu el codi original, de manera que faci exactament el mateix, però que si es deixa anar el pulsador quan està fent l'escala de color, es parin tots els LEDs.**

Hi ha moltes formes de solucionar aquest problema, algunes molt simples, altres més complicades. Potser la primera solució que ens ve al cap és crear un bucle i, cada vegada que s'encén un LED, es comprovi si el pulsador està premut. Si no, es surt del bucle.

```

#define BOTO 7
#define VERD 13
#define GROC 11
#define VERMELL 8
int lectura; //Variable on guardarem el valor del boto (HIGH o LOW)
void setup()
{
  //Configurem els pins
  pinMode(BOTO, INPUT); //Establim el pin del boto com a entrada
  pinMode(VERD, OUTPUT); //Els pins dels LEDs seran sortides
  pinMode(GROC, OUTPUT);
  pinMode(VERMELL, OUTPUT);
}

void loop()
{
  lectura = digitalRead(BOTO);

  while(lectura == HIGH)
  {
    digitalWrite(VERD, HIGH);

```

```

//Hem ences el primer LED, comprovem que el boto encara esta a
//HIGH i si no apaguem el LED verd i sortim del bucle
lectura = digitalRead(BOTO);
if(lectura != HIGH)
{
  digitalWrite(VERD, LOW);
  break;
}
delay(100);

digitalWrite(VERD, LOW);
digitalWrite(GROC, HIGH);
lectura = digitalRead(BOTO);
if(lectura != HIGH)
{
  digitalWrite(GROC, LOW);
  break;
}
delay(100);

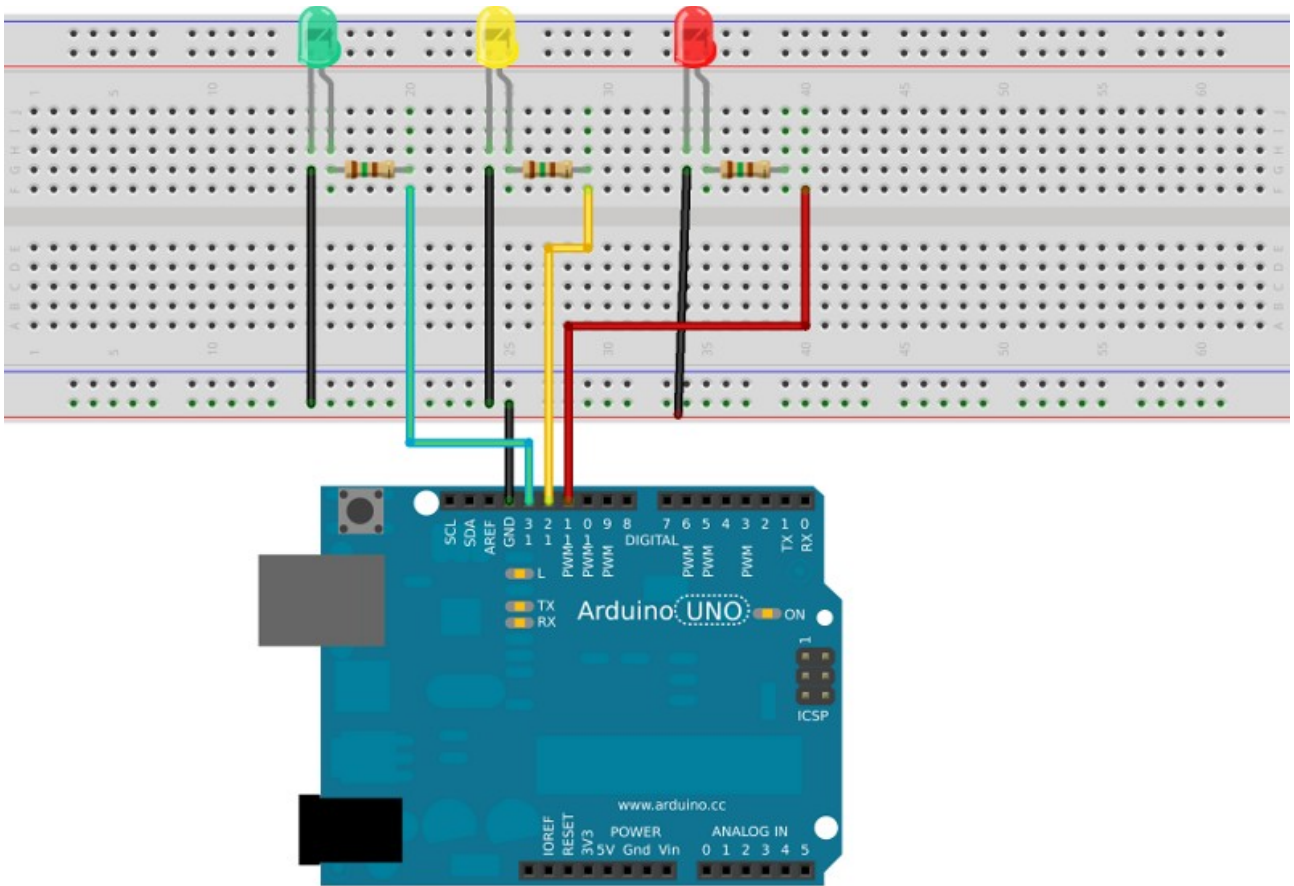
digitalWrite(GROC, LOW);
digitalWrite(VERMELL, HIGH);
lectura = digitalRead(BOTO);
if(lectura != HIGH)
{
  digitalWrite(VERMELL, LOW);
  break;
}
delay(100);
digitalWrite(VERMELL, LOW);
}
}

```

#### 4- BLINK PER SERIAL

a) **Poseu tres LEDs i, en funcio de les entrades per Serial, n'enceneu un o altre.**

Construïm un circuit amb 3 LEDs connectats als pins digitals. Per exemple:



Made with  Fritzing.org

En quant al codi, enviarem 4 caràcters per Serial:

- 'R' encendrà el LED vermell (Red)
- 'G' encendrà el LED verd (Green)
- 'Y' encendrà el LED groc (Yellow)
- 'S' apagarà tots els LEDs (Stop)

Si el caràcter no es cap dels anteriors, el programa no farà res.

```
#define VERD 13
#define GROC 12
#define VERMELL 11
```

```
byte entrada; //variable per guardar el caracter entrat
```

```
void setup()
{
  Serial.begin(9600); //iniciem el serial

  pinMode(13, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(8, OUTPUT);
}
```

```
void loop()
{
  if(Serial.available() >= 1) //Si s'envia quelcom per serial
  {
    entrada = Serial.read();

    if(entrada == 'G')
```



```

    {
      digitalWrite(VERD, HIGH);
      digitalWrite(GROC, LOW);
      digitalWrite(VERMELL, LOW);
    }
    else if(entrada == 'R')
    {
      digitalWrite(VERD, LOW);
      digitalWrite(GROC, LOW);
      digitalWrite(VERMELL, HIGH);
    }
    else if(entrada == 'Y')
    {
      digitalWrite(VERD, LOW);
      digitalWrite(GROC, HIGH);
      digitalWrite(VERMELL, LOW);
    }
    else if(entrada == 'S')
    {
      digitalWrite(VERD, LOW);
      digitalWrite(GROC, LOW);
      digitalWrite(VERMELL, LOW);
    }
    else{}
  }
}

```

b) **Modifiqueu el codi de manera que si entres un número, t'escrigui per Serial el mateix número que has entrat.**

En aquest exercici no ens cal cap circuit, només crear un codi que escrigui per Serial les entrades. Per això crearem una variable char que guardarà el valor de l'entrada i l'escriurà.

No fem servir el comande Serial.println() ja que ens escriuria un salt de línia després de cada caràcter i no ens interessa. Per exemple, el 55 l'agafaria com a dos caràcters (5 i 5), amb un salt de línia després de cada un. Per tant, fem servir el comande Serial.print() que farà el salt de línia únicament quan premem enter.

```

char entrada;

void setup()
{
  Serial.begin(9600); //iniciem el serial
}

void loop()
{
  if(Serial.available() >= 1)
  {
    entrada = Serial.read();
    Serial.print(entrada);
  }
}

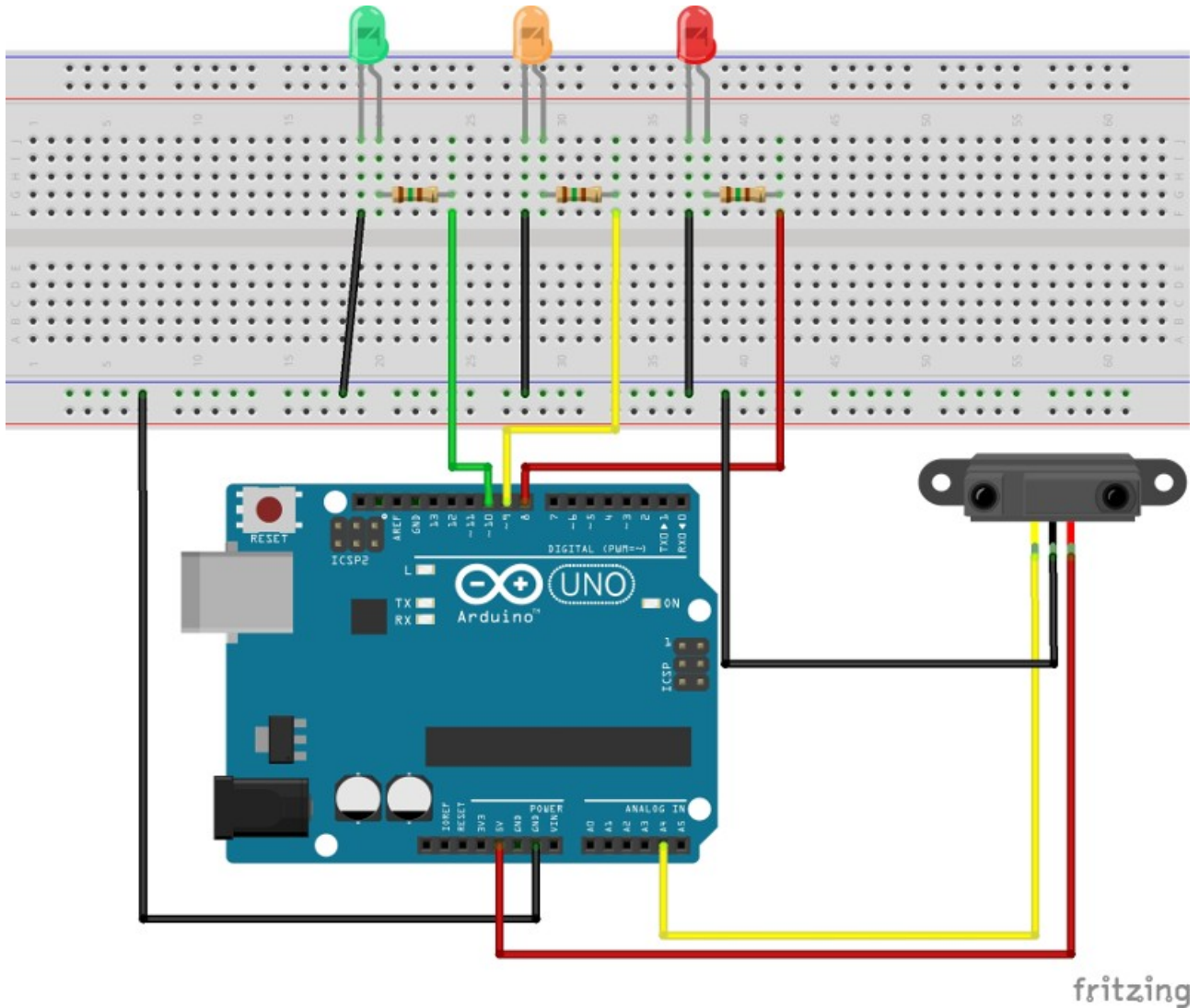
```

## 5- Sensor de distància SHARP

a) Segons la distància del Sensor, s'ha de fer el següent:

- (300-400): s'encén un LED verd i s'envia per serial un missatge a l'estil "OBJECTE ALLUNYAT"
- (400-500): s'encén un LED groc i s'escriu per serial "OBJECTE PROPER"
- (distància > 500): s'encén un LED vermell i s'escriu per serial "OBJECTE MOLT PROPER"

El circuit serà el següent. Recordem que si canviem de posició els pins digitals i/o analògics, haurem de canviar també el codi.



Farem un codi que guardi la distància del sensor en una variable. Després, hi haurà un condicional amb quatre possibilitats, les tres primeres encendran els LEDs Verd, Groc i Vermell i la última apagarà tots els llums si no detecta res.

```
#define RED 8 //LED vermell
#define YLL 9 //LED groc
#define GRN 10 //LED verd

int dist; //variable per guardar distancia

void setup()
{
  //Configurem els pins
  pinMode(RED, OUTPUT);
```

```

pinMode(YLL, OUTPUT);
pinMode(GRN, OUTPUT);

//Inicialitzem Serial
Serial.begin(9600);
}

void loop()
{
  dist = analogRead(4); //Llegim el pin analògic 4

  //Conditionals
  if(dist <= 400 and dist > 300)
  {
    Serial.println("OBJECTE ALLUNYAT!");
    digitalWrite(GRN, HIGH);
    digitalWrite(YLL, LOW);
    digitalWrite(RED, LOW);
  }
  else if(dist <= 500 and dist > 400)
  {
    Serial.println("OBJECTE PROPER!");
    digitalWrite(YLL, HIGH);
    digitalWrite(GRN, LOW);
    digitalWrite(RED, LOW);
  }
  else if(dist > 500)
  {
    Serial.println("OBJECTE MOLT PROPER!");
    digitalWrite(RED, HIGH);
    digitalWrite(GRN, LOW);
    digitalWrite(YLL, LOW);
  }
  else
  {
    //Si no es dona cap cas, no apaga tots els llums
    digitalWrite(RED, LOW);
    digitalWrite(GRN, LOW);
    digitalWrite(YLL, LOW);
  }
  delay(100);
}

```

**b) A l'hora de fer el robot de sumo, ens serà molt útil tenir una funció per detectar objectes. Escriviu una funció booleana que rebí com a argument un número enter i retorni cert si el sensor analògic del número entrat com a argument detecta un objecte.**

Ja sabem què és una funció: és un fragment de codi que es pot cridar diverses vegades durant el codi, així no l'hem d'estar escrivint varies vegades.

Necessitarem una funció que retorni una variable booleana, per tant la nostra funció serà de tipus bool i es dirà...

```
bool llegirDistancia()
```

Aquesta funció rebrà un argument que serà el pin que volem llegir. Això és necessari perquè l'Arduino no té superpoderes bovins i no pot llegir-nos la ment per saber quin pin necessitem. Per tant, el codi complet serà:

```
#define LED 13
```

```
//La variable per la distancia la declararem dins la funcio
```

```

void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}

//(La funcio esta al final de tot)
void loop()
{
  //Si el resultat de la funcio es cert...
  if(llegirDist(4))
  {
    digitalWrite(LED, HIGH);
  }
  else
  {
    digitalWrite(LED, LOW);
  }

  delay(100);
}

bool llegirDist(int analog)
{
  //Rep com a argument un enter que sera el pin que volem llegir

  int dist = analogRead(analog);

  return (dist > 350);
  /*Si el resultat de la comparacio dist > 350
  es vertader, retornara TRUE, si no retornara FALSE*/
}

```

Potser ja haureu notat que void loop() i void setup() també són funcions. Arduino necessita aquestes funcions sempre per funcionar.

8- "Yo... VIVO"! construcció d'un avioder

**a) Afegiu el segon sensor al pin A2. Col·loqueu un dels sensors que apunti davant del robot, inclinat cap a la dreta i l'altre cap a l'esquerra. Modifiqueu el codi de manera que si el sensor de la dreta detecta un obstacle, giri a l'esquerra, i si es el de l'esquerra que detecta un obstacle, que giri a la dreta.**

Haurem de modificar el codi original afegint un else if que llegeixi el sensor connectat a A2.

```

#include <ZumoMotors.h>
#include <Pushbutton.h>
ZumoMotors motors;
Pushbutton button(ZUMO_BUTTON); //A l'igual que amb els motors, creem un objecte
button
int dreta = 0; //Aqui guardarem la lectura del sensor
int esquerra = 0; //Nova variable per la lectura del nou sensor
int vel = 200; //Variable per establir la velocitat. 200 es una velocitat
estandar.
void setup()
{
  while(not button.isPressed()); //Mentre no es premi el boto, no es continuara el
  programa
  delay(3000); //Esperem tres segons
}
void loop()
{

```

```

dreta = analogRead(3); //Llegim el pin analògic 3
esquerra = analogRead(2); //Llegim el pin analògic 2
if(dreta > 450)
{
  motors.setSpeeds(-vel, -vel);
  delay(500);
  motors.setSpeeds(-vel, vel);
  delay(200);
}
else if(esquerra > 450)
{
  motors.setSpeeds(-vel, -vel);
  delay(500);
  motors.setSpeeds(vel, -vel);
  delay(200);
}
else
{
  motors.setSpeeds(vel, vel);
}
}

```

b) **Modifiqueu el codi per tal que el robot vagi donant voltes sobre si mateix i, si detecta un obstacle, avanci cap a ell.**

Conectarem el sensor al pin analògic 3

```

#include <ZumoMotors.h>
#include <Pushbutton.h>

ZumoMotors motors;
Pushbutton button(ZUMO_BUTTON);
int sensor = 0; //Aqui guardarem la lectura del sensor
int vel = 200;

void setup()
{
  while(not button.isPressed());
  delay(3000); //Esperem tres segons
}

void loop()
{
  dreta = analogRead(3); //Llegim el pin analògic 3
  if(sensor > 450)
  {
    motors.setSpeeds(vel, vel);
    delay(100);
  }
  else
  {
    motors.setSpeeds(-vel, vel); //Donara voltes sobre si mateix
  }
}

```

Si ara afegíssim un sistema per detectar la línia i no passar-nos, ja tindríem el robot de sumo més senzill possible.

## 9 – CODI DEL ROBOT DE SUMO SENZILL

Finalment, aquí tenim el codi per fer un robot de sumo simple. A partir d'aquest és possible afegir més funcions i moviments més complexes.

Aquest robot anirà fent voltes sobre si mateix fins que no detecti un contrincant o una línia. Si detecta un contrincant, anirà cap a ell a velocitat màxima. Si detecta la línia, tirarà endarrere.

És una estratègia defensiva, el robot espera a que s'acosti el contrari per envestir-lo.

```
#include <ZumoMotors.h>
#include <Pushbutton.h>
#include <QTRSensors.h>
#include <ZumoReflectanceSensorArray.h>

byte pins[] = {4, A3, A2, 5};
ZumoReflectanceSensorArray sensors(QTR_NO_EMITTER_PIN);
unsigned int sensor_values[4];

ZumoMotors motors;

int vel = 200;

void setup()
{
  sensors.init(pins, 4);
  Serial.begin(9600);
}

void loop()
{
  if(detectaObjecte(0))
  {
    //Si detecta un objecte, avança a tota maquina!!!!
    motors.setSpeeds(400, 400);
    delay(100);
  }
  else if(detectaLinia())
  {
    //Si detectes una línia i no hi ha objectes, gira
    motors.setSpeeds(-vel, -vel);
    delay(300);
    motors.setSpeeds(vel, -vel);
    delay(400);
  }
  else
  {
    //Si no, ves fent voltes
    motors.setSpeeds(vel, -vel);
    delay(100);
  }
}

bool detectaObjecte(int pin)
{
  int dist = analogRead(pin);
  return (dist > 350);
}

bool detectaLinia()
{
```

```
sensors.read(sensor_values);  
bool hiHaLinia = false;  
int i = 0;  
while(i < 4)  
{  
    if(sensor_values[i] > 500)  
    {  
        hiHaLinia = true;  
        break;  
    }  
    i++;  
}  
return hiHaLinia;  
}
```