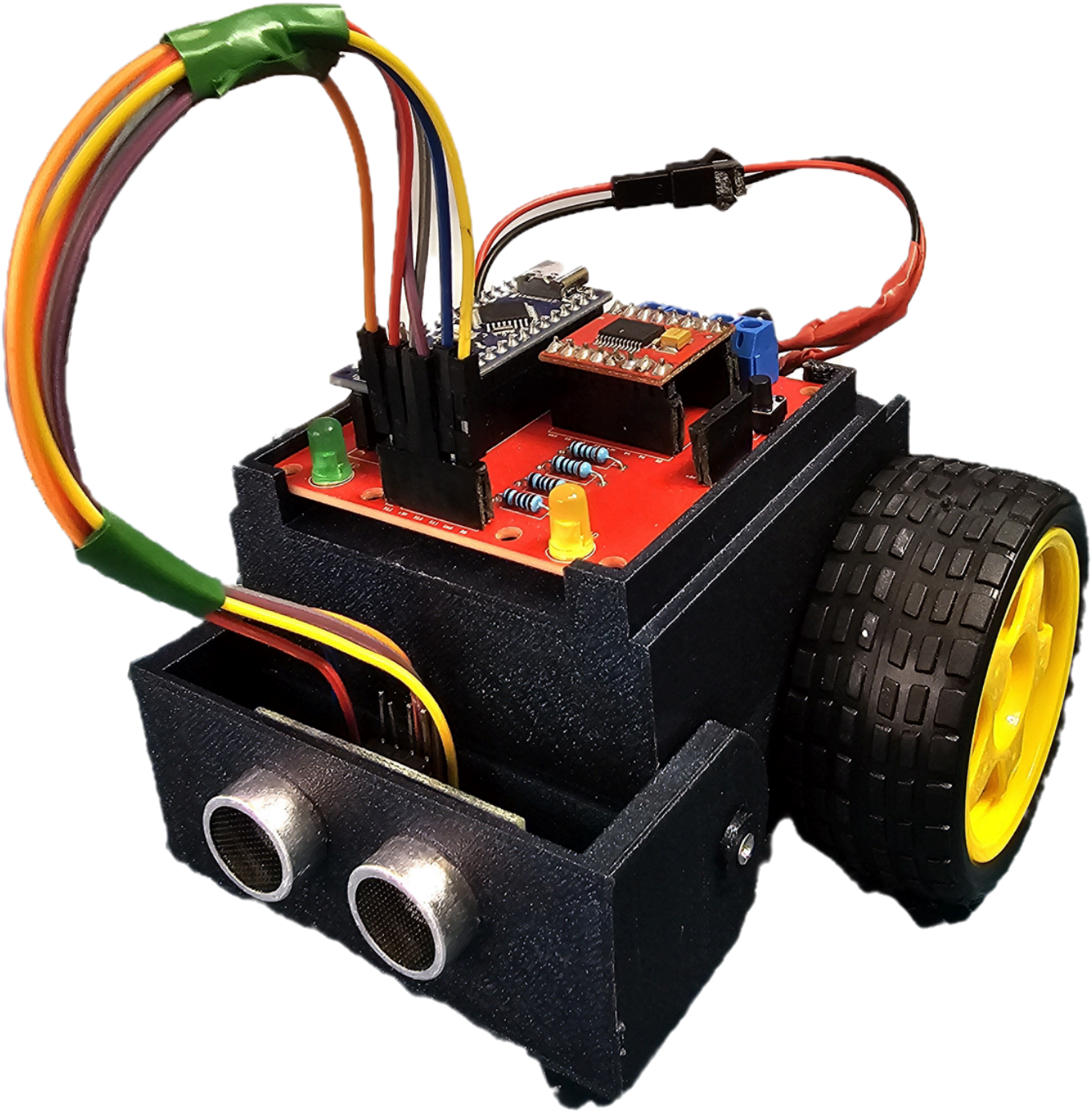
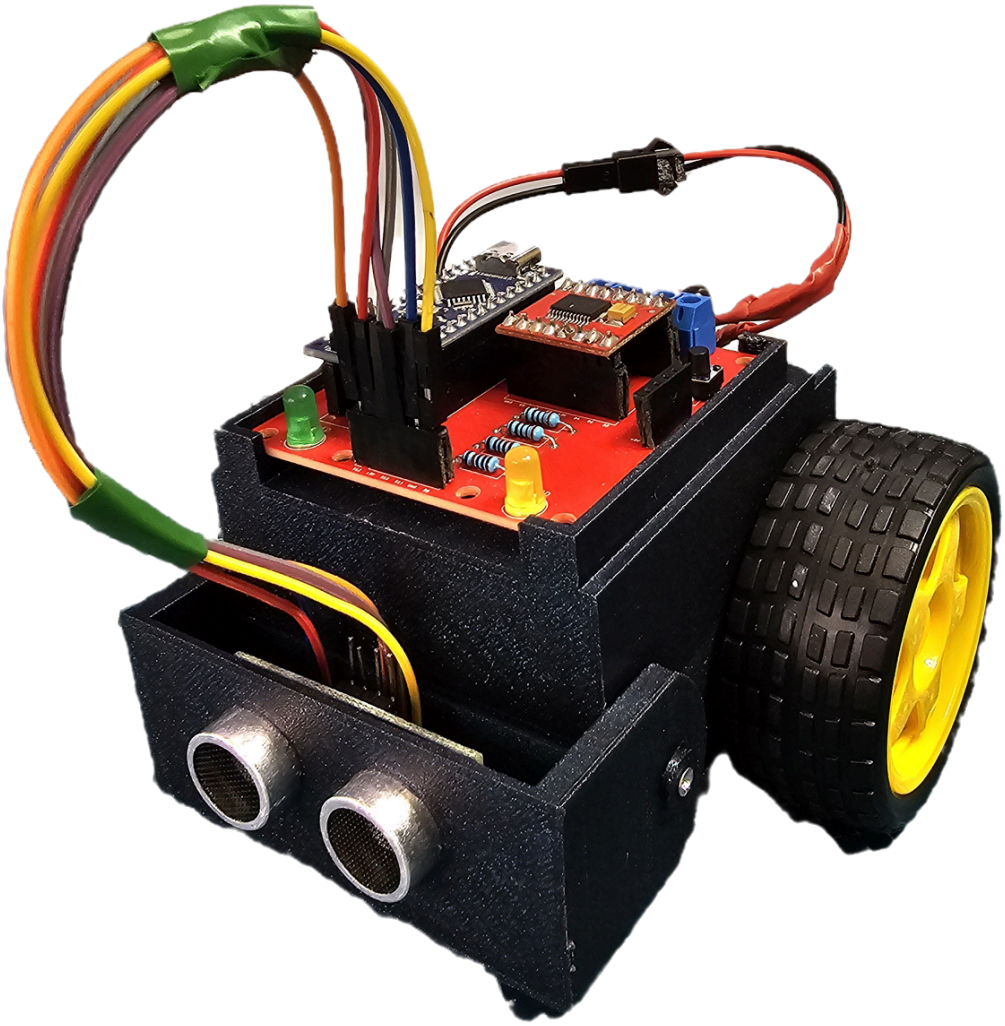
VelociBot

TUTORIAL ROBOT MULTIPROPOSITO



Introducción



Robot Multi Proposito

Es un robot de la categoría Velocitas y sumo que ha sido diseñado por el CITED. Las principales premisas en su diseño fueron las siguientes:

- Sencillez de montaje, utilizando el menor número de tornillos, tuercas, adhesivos,...
- Menor coste de los componentes y material, y facilidad en su adquisición.
- Control realizado con Arduino.
- Chasis impreso en 3D y diseñado con SketchUp.

El objetivo principal de este proyecto es que cualquier persona que quiera construir este robot (profesores, estudiantes, aficionados,...) lo pueda hacer en muy poco tiempo, con las mínimas complicaciones y el mínimo coste.

Material, montaje y conexiones

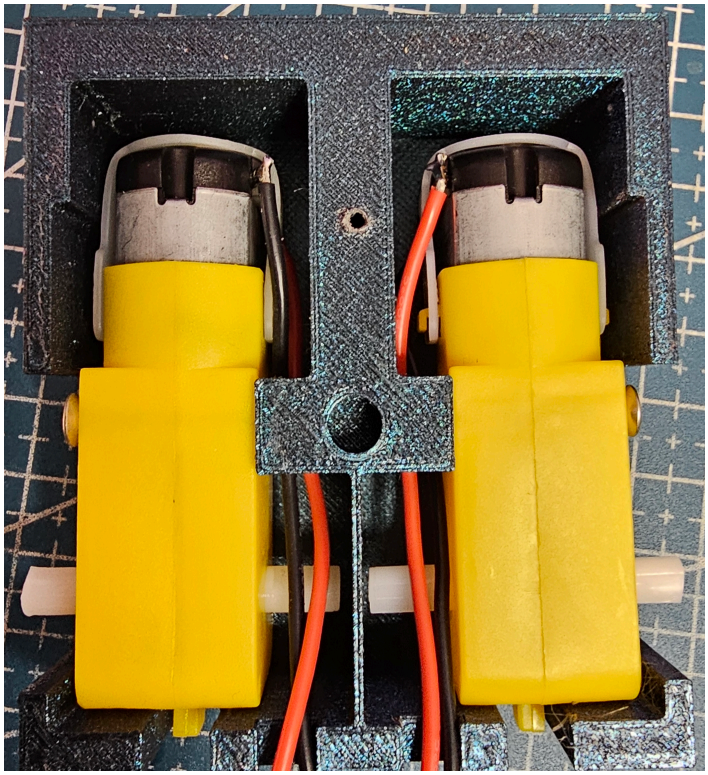
Listado de materiales para el robot

Chasis impreso en 3D

motores con reductora

PCB para control de el robot

Motores



Detalle de colocación del motor

MONTAJE DE LOS MOTORES

Empezaremos montando los dos motorreductores CC en el chasis utilizando dos tornillos de 2,5 x 25 mm por cada motor. Previamente se deben realizar las conexiones de los cables a los terminales del motor.

IMPORTANTE: NO APRETAR DEMASIADO LOS TORNILLOS. Puede llegar a presionar la reductora del motor y hacer que este se quede trabado.

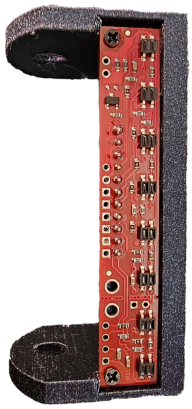
El motorreductor debe ir colocado en la posición de la flecha verde.

Por último, colocaremos las ruedas en los ejes de los motores. Para mejorar el agarre con la pista, se recomienda utilizar materiales como juntas tóricas o cinta tapafugas sobre la superficie de las ruedas.

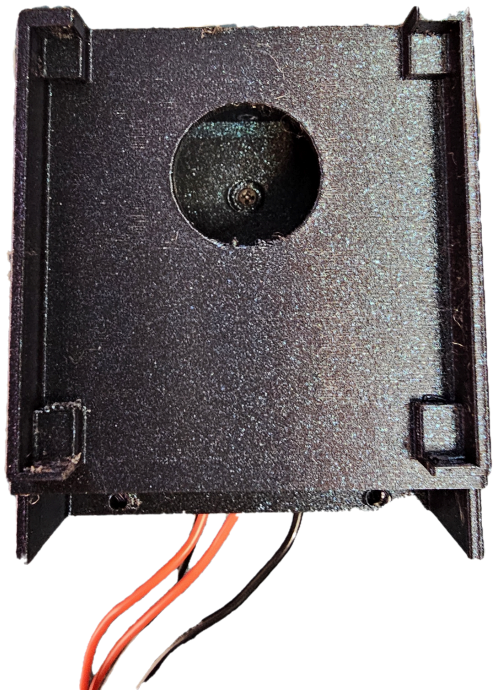
ParteSuperior y PCB

SENSORES INFRARROJOS (IR)

A continuación colocamos el array de sensores de línea negra. En un primer momento su alojamiento está diseñado para colocarlos simplemente por presión, pero se podría sustituir esta sujeción por tornillos de 2x6 mm o punto de termocola.



Array sensores

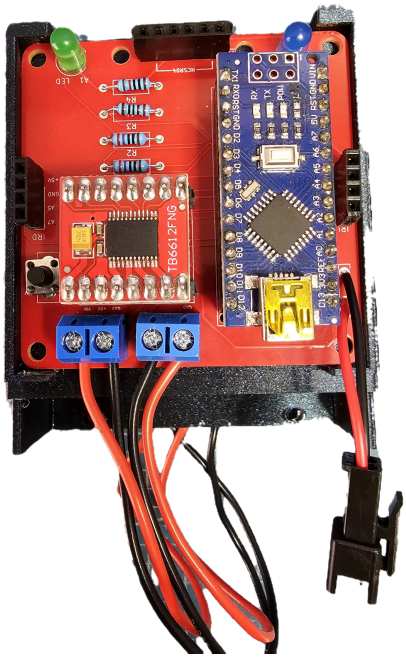


Parte superior de el robot

Parte superior y Placa PCB

Seguidamente, colocaremos parte superior del robot sobre la estructura sujetándola con un tornillo, y sobre ella colocaremos la PCB en la parte superior, tal y como se ve en la foto.

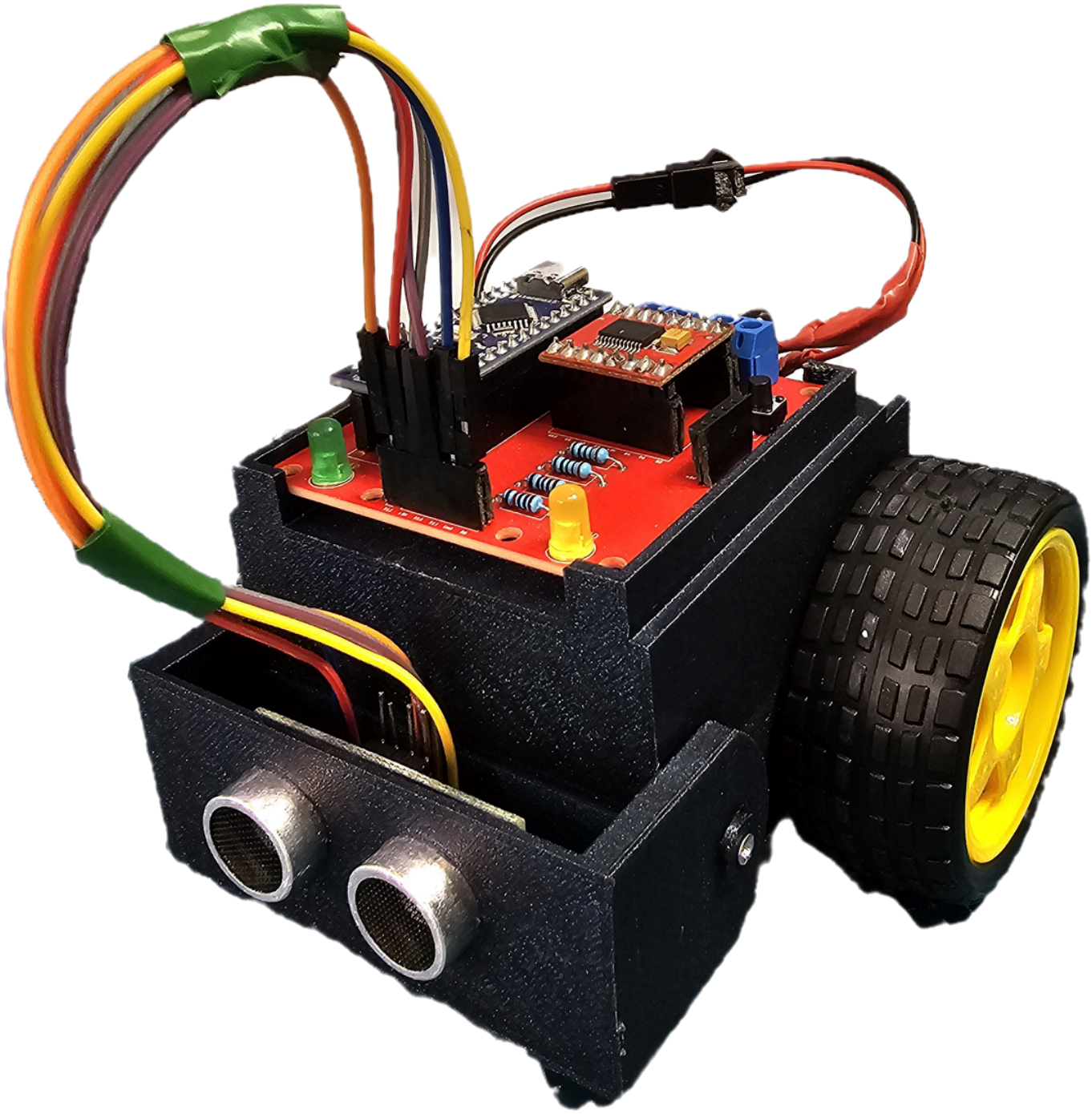
En la parte inferior está el alojamiento para el portapilas.



Detalle PCB

Conexionado

Una vez se realicen las conexiones de los componentes, se obtiene un resultado similar al de la siguiente imagen:



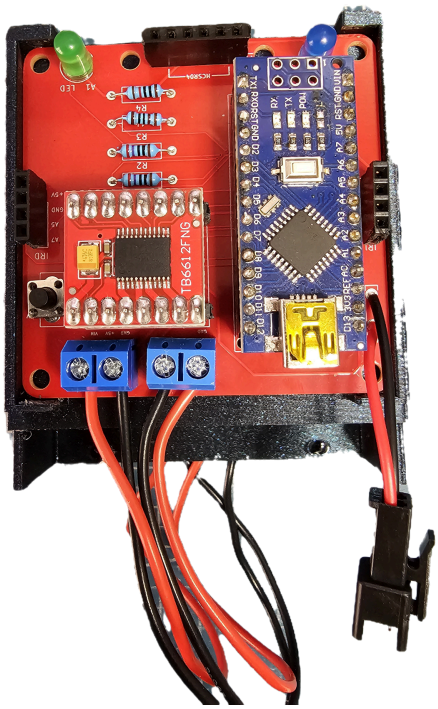
Motores

CONEXIONADO DE MOTORES Y ALIMENTACIÓN

El siguiente paso consiste en conectar los motores a la PCB. Visto el robot desde arriba, el Motor Izquierdo (MI) lo conectaremos a los terminales M1 y el Motor Derecho (MD) a los terminales M2.

IMPORTANTE: Cuando se realice el Test de motores habrá que revisar su polaridad.

El portapilas dispone de interruptor para cortar la alimentación en caso de ser necesario.



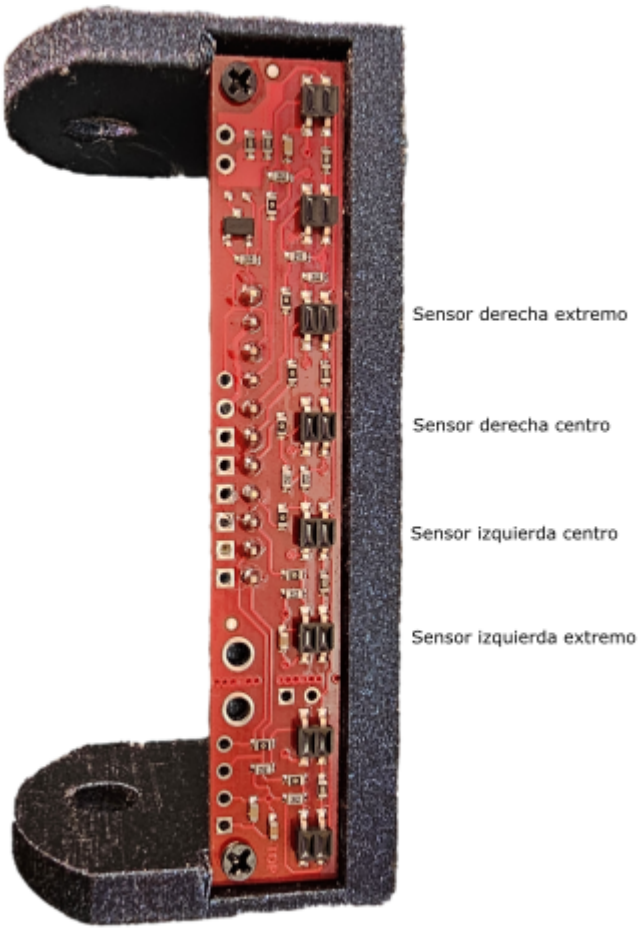
Conexionado motores y alimentación

Sensores IR

CONEXIONADO SENSORES IR

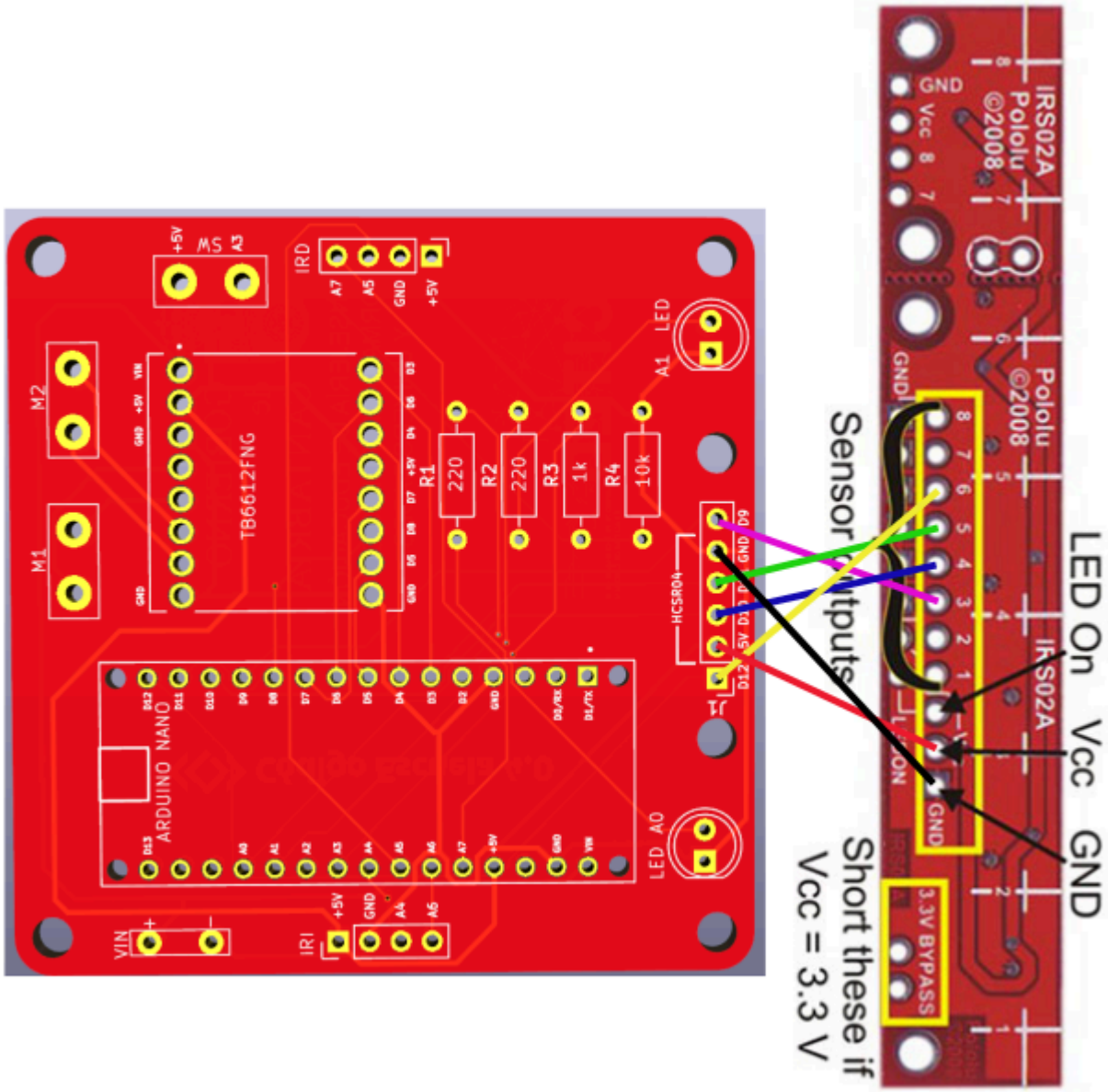
La conexión a el array de sensores es la siguiente:

Sensor Izquierdo Extremo	Pin Digital 12
Sensor izquierdo Central	Pin Digital 11
Sensor derecha Central	Pin Digital 10
Sensor derecha Extremo	Pin Digital 9



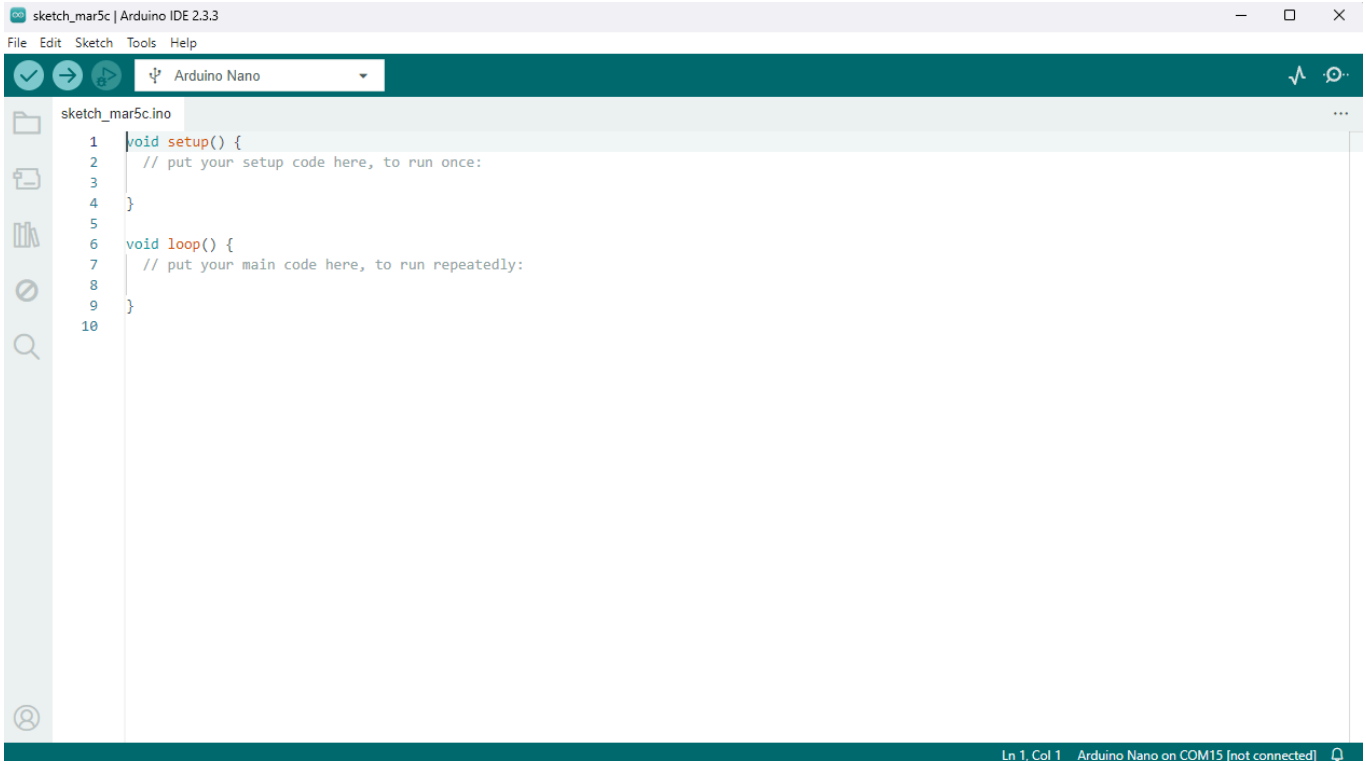
Nomenclatura de los sensores de línea negra

El esquema de conexión para la PCB es el siguiente:


LED On
Vcc
GND
Sensor outputs
Short these if
Vcc = 3.3 V
IRS02A
Pololu
©2008
GND
Vcc
3.3V BYPASS
TB6612FNG
ARDUINO NANO
HCSR04
R1
220
R2
220
R3
1k
R4
10k
LED A0
LED A1
M1
M2
VIN
IRD
IRI
MS

Programación por CÓDIGO

Utilizaremos el entorno Arduino IDE para programar por código.



Test motor derecho (MD)

Con esta práctica se pretende comprobar el correcto conexionado del motor, conocer su sentido de giro y su control. Haremos que gire hacia adelante durante 1 seg, que se pare 2 seg y que gire hacia atrás 1 seg. Puedes descargar el programa pulsando aquí. Una vez compruebes el motor derecho, realiza los cambios necesarios en el programa para comprobar el motor izquierdo.

RECUERDA: Si el motor DERECHO no gira en el sentido que le indicamos, debemos cambiar su polaridad en la Shield.

```
int PwmA=3;
int PwmB=5;
int Ma1=4;
int Ma2=6;
int Mb1=7;
int Mb2=8;
void setup() {
  // put your setup code here, to run once:

}

void loop() {
 analogWrite(PwmA,255);
 digitalWrite(Ma1,HIGH);
 digitalWrite(Ma2,LOW);

}
```

Movimientos básicos

Los movimientos básicos del robot son los siguientes: ADELANTE PARO ATRÁS GIRO DERECHA GIRO IZQUIERDA

Ahora vamos a trabajar con los dos motores a la vez. Para realizar los giros debemos parar el motor que esté en el sentido que queramos girar. Por ejemplo, si queremos girar a la izquierda debemos parar el motor izquierdo y si queremos girar a la derecha, debemos parar el motor derecho.

```
int PwmA=3;
int PwmB=5;
int Ma1=4;
int Ma2=6;
int Mb1=7;
int Mb2=8;
void adelante(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,HIGH);digitalWrite(Mb2,LOW);digitalWrite(Ma1,HIGH);digitalWrite(Ma2,LOW);
  }
void atras(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,HIGH);digitalWrite(Ma1,LOW);digitalWrite(Ma2,HIGH);
  }
void izquierda(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,LOW);digitalWrite(Ma1,HIGH);digitalWrite(Ma2,LOW);
  }
void derecha(){
  analogWrite(PwmA,0);
  analogWrite(PwmB,0);
  digitalWrite(Mb1,HIGH);digitalWrite(Mb2,LOW);digitalWrite(Ma1,LOW);digitalWrite(Ma2,LOW);
  }
void paro(){
  analogWrite(PwmA,0);
  analogWrite(PwmB,0);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,LOW);digitalWrite(Ma1,LOW);digitalWrite(Ma2,LOW);
  }
void setup() {
  // put your setup code here, to run once:

}

void loop() {
adelante();delay(2000);izquierda();delay(2000);derecha();delay(2000);atras();delay(2000);paro();delay(2000);
}
```

Movimientos básicos (subrutinas)

Las SUBRUTINAS (o FUNCIONES) son unos *"miniprogramas"* que se ejecutan al ser llamadas desde el programa principal. Las subrutinas o funciones se ubican después del *SETUP*. Son útiles cuando tenemos que utilizar de forma recurrente un conjunto de instrucciones determinado. Puedes descargar el programa pulsando aquí.

En esta actividad crearemos 5 subrutinas: ADELANTE PARO ATRÁS GIRO DERECHA GIRO IZQUIERDA

```
int PwmA=3;
int PwmB=5;
int Ma1=4;
int Ma2=6;
int Mb1=7;
int Mb2=8;
void adelante(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,HIGH);digitalWrite(Mb2,LOW);digitalWrite(Ma1,HIGH);digitalWrite(Ma2,LOW);
  }
void atras(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,HIGH);digitalWrite(Ma1,LOW);digitalWrite(Ma2,HIGH);
  }
void izquierda(){
  analogWrite(PwmA,255);
  analogWrite(PwmB,255);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,LOW);digitalWrite(Ma1,HIGH);digitalWrite(Ma2,LOW);
  }
void derecha(){
  analogWrite(PwmA,0);
  analogWrite(PwmB,0);
  digitalWrite(Mb1,HIGH);digitalWrite(Mb2,LOW);digitalWrite(Ma1,LOW);digitalWrite(Ma2,LOW);
  }
void paro(){
  analogWrite(PwmA,0);
  analogWrite(PwmB,0);
  digitalWrite(Mb1,LOW);digitalWrite(Mb2,LOW);digitalWrite(Ma1,LOW);digitalWrite(Ma2,LOW);
  }
```

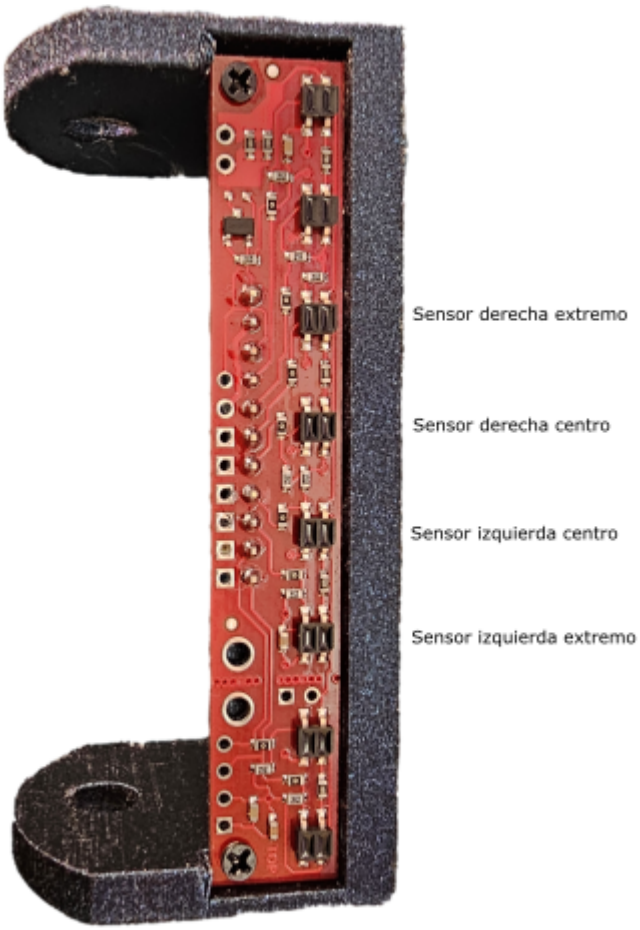
Giros avanzados

Se plantean a continuación 3 tipos de giros: lentos, suaves y rápidos. Los “lentos” consistirán en dejar parados los motores del interior del giro haciendo avanzar los del exterior, para los “suaves” cambiamos el PWM del motor interior a valores más pequeños que el exterior. Y los “rápidos” haciendo que los motores del interior giren hacia atrás mientras que los del exterior van hacia adelante.

Se puede ir variando las velocidades para obtener diferentes resultados en función de lo que se necesite. Como ejemplos de valores tenemos el siguiente cuadro:

Sensores IR

Un sensor infrarrojo es un dispositivo optoelectrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión. El sensor está compuesto por un diodo emisor de luz y un fototransistor. Si el fototransistor recibe luz da una señal de cero, si no recibe luz da un 1.



Array Sensores

A la hora de utilizar este tipo de sensores es importante tener en cuenta lo siguiente:

- La distancia de detección no es muy grande: no debe estar ni pegado a la superficie ni muy separado (aprox. 1 cm).
- Son sensibles a los reflejos que pueda producir la superficie debido a la luz ambiental, por lo que es recomendable que el diseño del chasis haga que siempre trabajen en las mismas condiciones de iluminación para que este factor les afecte lo menos posible.
- Se recomienda comprobar la lectura del sensor antes de empezar a utilizarlo en el programa.
- Son muy sensibles a las polaridades cruzadas: no confundir los cables de alimentación.

En el siguiente programa comprobamos a través del Monitor Serie la lectura de un sensor IR. De esta forma podremos comprobar su correcto funcionamiento antes de comenzar a utilizarlo, además de saber que valor devuelve al leer el color negro o el color blanco ('0' y '1' o viceversa).

Al aplicarlo observamos que cuando el sensor detecta línea negra su valor es 1 y cuando detecta blanco su valor es 0. (En el propio sensor hay un led rojo que se enciende cuando

```
int cnydd=12;
int cnyd=11;
int cnyi=10;
int cnyii=9;
void setup() {
Serial.begin(9600);

}

void loop() {
Serial.print(digitalRead(cnydd));
Serial.print(digitalRead(cnyd));
Serial.print(digitalRead(cnyi));
Serial.println(digitalRead(cnyii));
}
```

marca 0).

El valor de SII es 0

El valor de SIC es 1

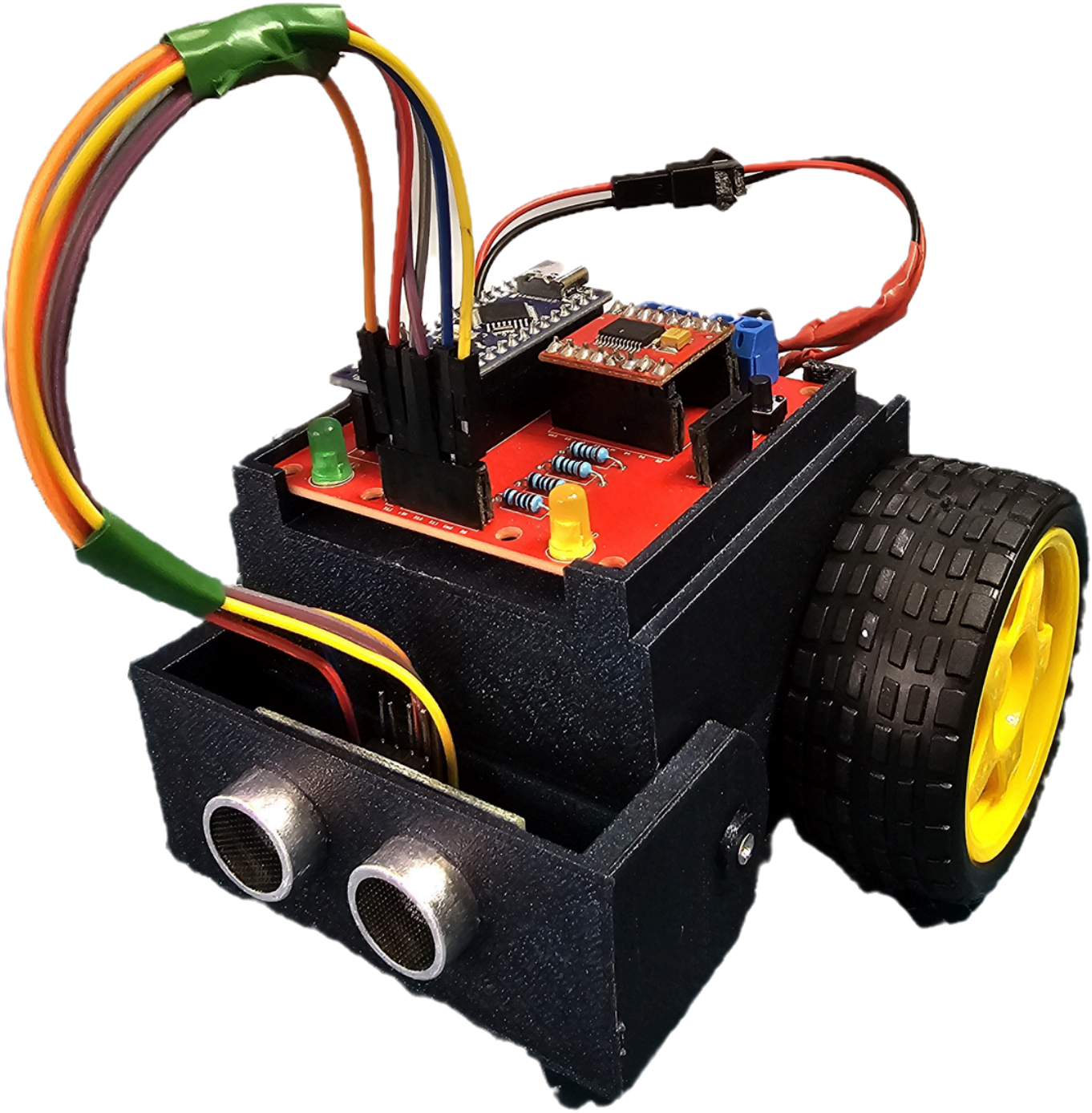
El valor de SDC es 1

El valor de SDD es 0

Seguidor de línea básico

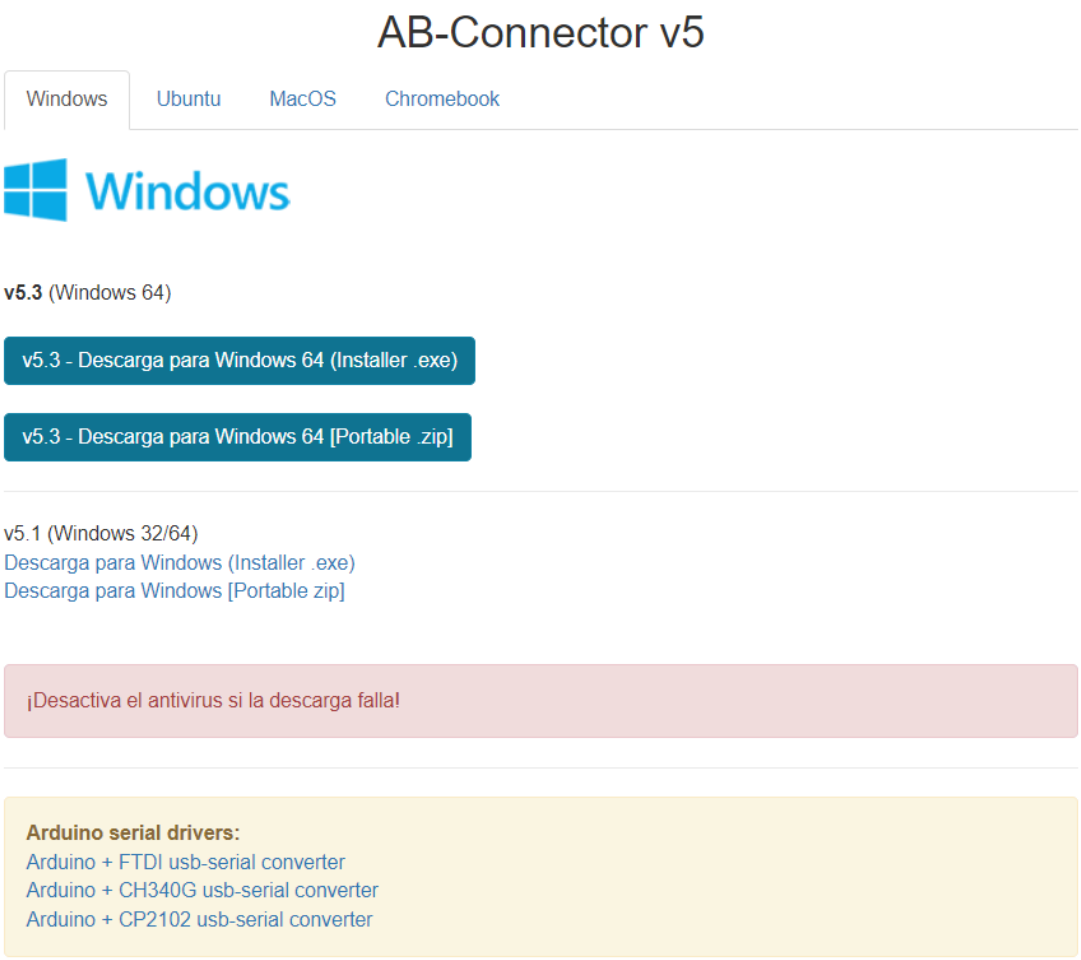
En esta actividad únicamente utilizaremos los dos sensores centrales del robot *(*el *SIC* y el *SDC*) con ello conseguiremos que el robot siga una línea negra de 2,5 cm de grosor sobre un fondo blanco.

A partir de este momento el funcionamiento del robot se puede mejorar utilizando sus 4 sensores y cambiando las velocidades de los motores, pero eso lo dejamos en tus manos...



Programación por BLOQUES

Utilizaremos el entorno de bloques



Página de descarga de AB Connector

ArdunuinoBlocks, creado por Juanjo López. ArduinoBlocks es una plataforma online donde puedes generar y guardar tus proyectos, así como generar ejercicios como profesor y enviárselos a tus alumnos.

Aun siendo online, necesita un pequeño programa instalado en nuestro ordenador para funcionar: ArduinoBlocks Connector, que puedes descargar aquí de su página oficial.

Además, también puedes descargar en esa misma página los drivers necesarios en el caso de que tu ordenador no reconozca tu placa Arduino.

Una vez descargado e instalado ArduinoBlocks Connector, lo ejecutamos y lo dejamos funcionando en segundo plano, mientras que abrimos el navegador y accedemos a la plataforma ArduinoBlocks. Nos registramos en ella y estaremos listos para utilizarla.

Vamos a crear un nuevo proyecto personal, basado en la placa Arduino Nano al que llamaremos Velocista.

Iniciar un nuevo proyecto personal

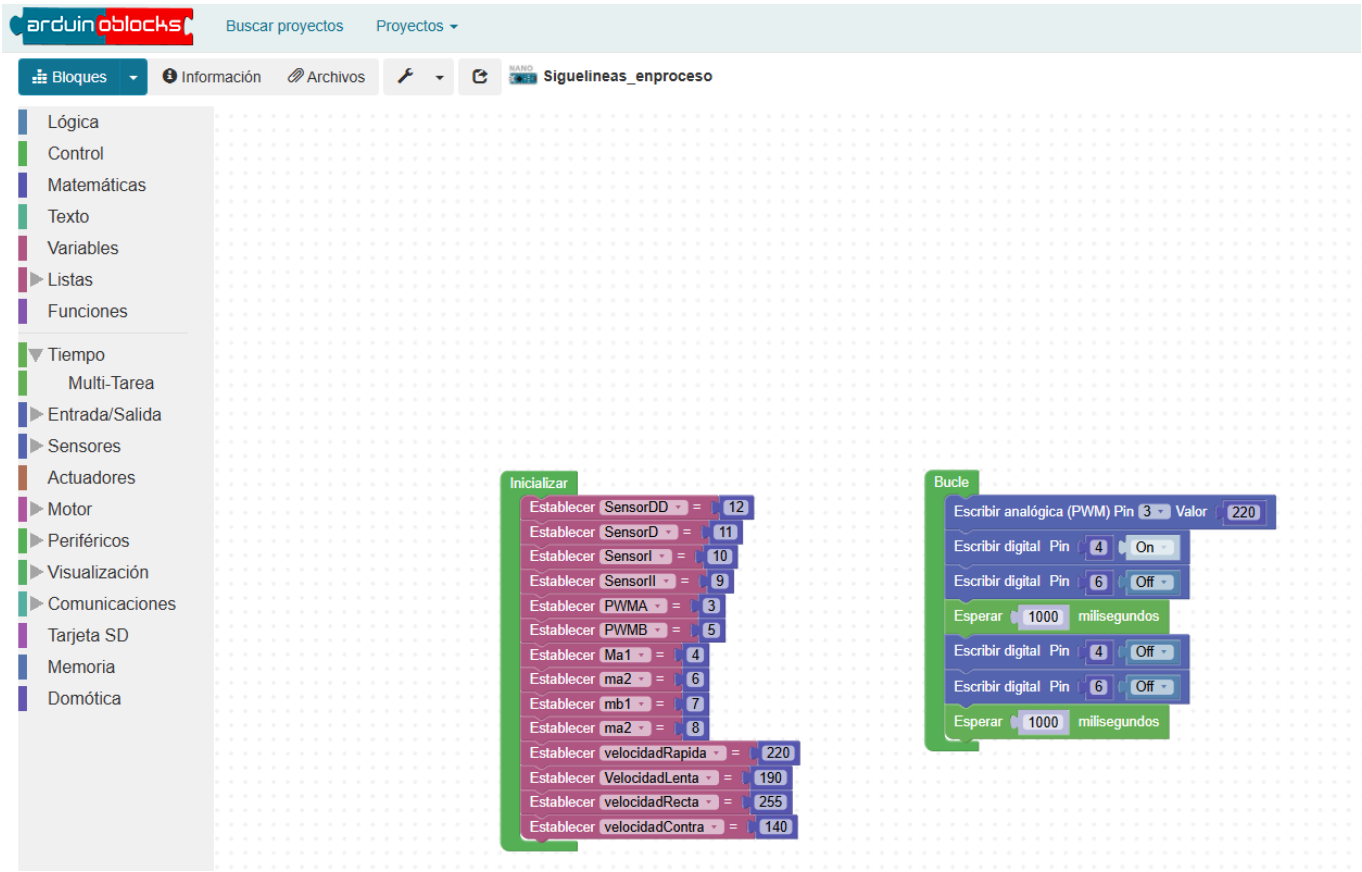
Tipo de proyecto y nombre

Test motor derecho (MD)

Con esta práctica se pretende comprobar el correcto conexionado del motor, conocer su sentido de giro y su control. Haremos que gire hacia adelante durante 1 seg y que se pare 1 seg . Una vez compruebes el motor derecho, realiza los cambios necesarios en el programa para comprobar el motor izquierdo.

RECUERDA: Si el motor DERECHO no gira en el sentido que le indicamos, debemos cambiar su polaridad en la PCB.

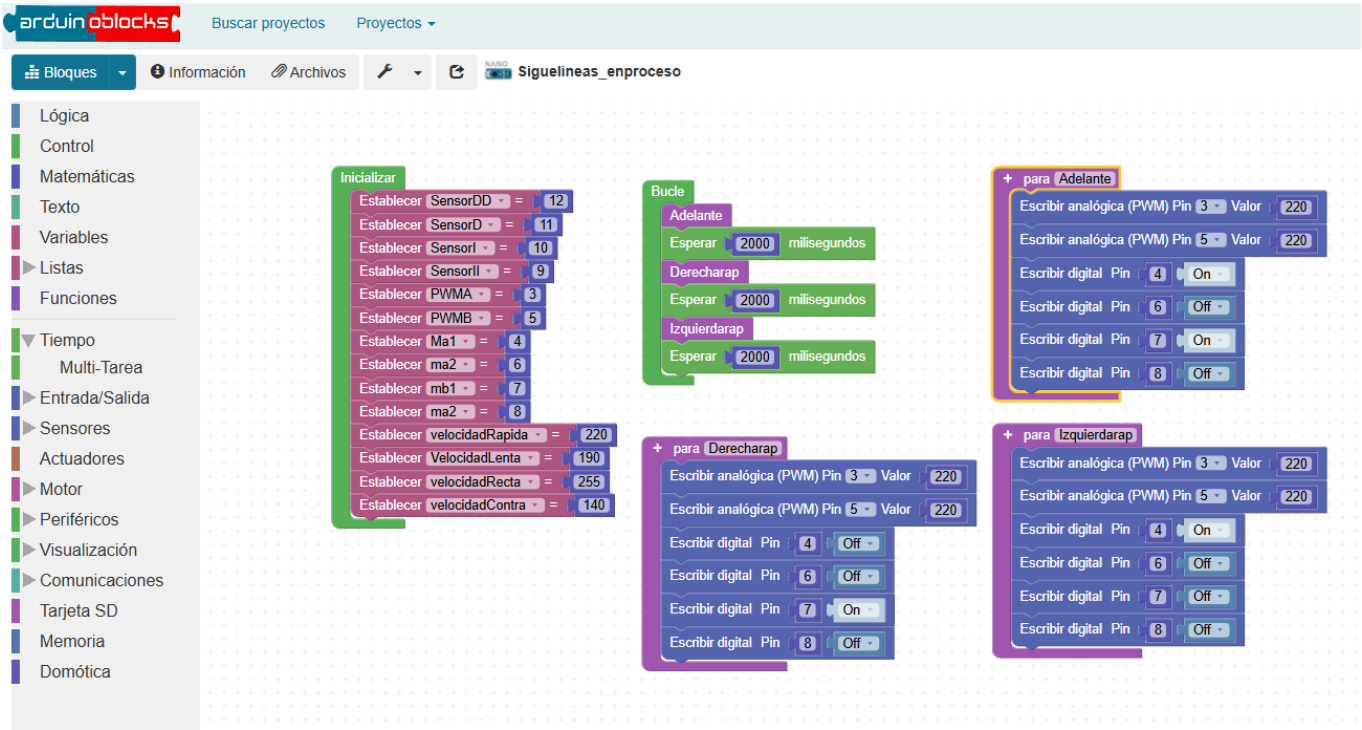
En la siguiente imagen se muestra el código que realiza la prueba del motor derecho, conectando éste en la PCB:



Movimientos básicos

Los movimientos básicos del robot son los siguientes: ADELANTE PARO ATRÁS GIRO DERECHA GIRO IZQUIERDA

Ahora vamos a trabajar con los dos motores a la vez. Para realizar los giros debemos parar el motor que esté en el sentido que queramos girar. Por ejemplo, si queremos girar a la izquierda debemos parar el motor izquierdo y si queremos girar a la derecha, debemos parar el motor derecho.



Movimientos básicos (subrutinas)

Las SUBRUTINAS (o FUNCIONES) son unos "miniprogramas" que se ejecutan al ser llamadas desde el programa principal. Las subrutinas o funciones se ubican después del SETUP. Son útiles cuando tenemos que utilizar de forma recurrente un conjunto de instrucciones determinado. Puedes descargar el programa pulsando aquí.

En esta actividad crearemos 5 subrutinas: ADELANTE PARO ATRÁS GIRO DERECHA GIRO IZQUIERDA