

Manual SamiBot v2

Gracias por adquirir uno de nuestros fantásticos productos, a continuación detallaremos como preparar el entorno Arduino para usar este robot y también te proporcionaremos un código de ejemplo con el que podrás iniciar.

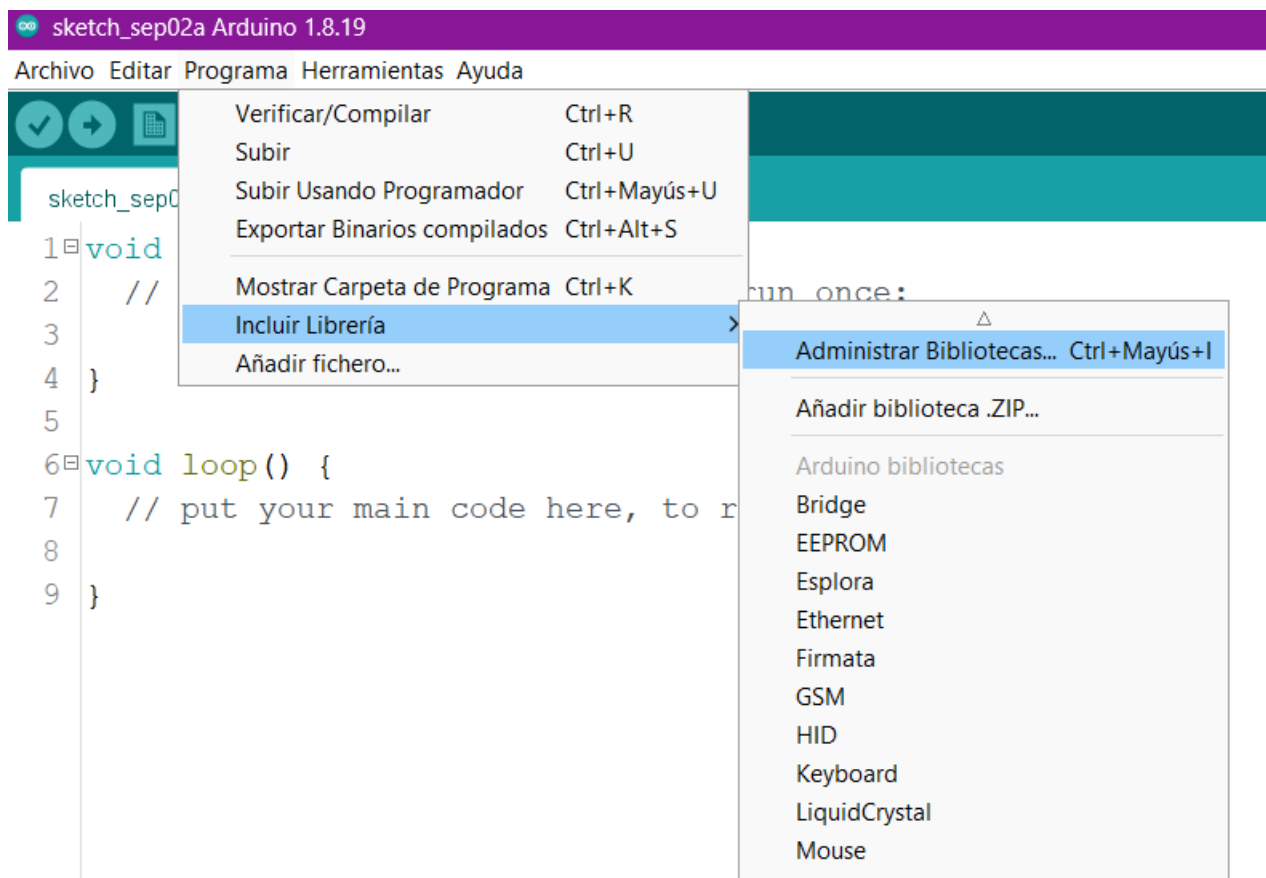
SamiBot es un robot que esta equipado con 2 de nuestros Smart Motor Drivers, los cuales te permitirán manipular los motores con un sistema de control de lazo cerrado para velocidad y distancia. Lo mejor de todo es que no se requiere programación avanzada para utilizarlos. Con nuestro código de ejemplo y las instrucciones de este manual podrás empezar de inmediato sin mayor complicación.

Preparando el entorno Arduino IDE

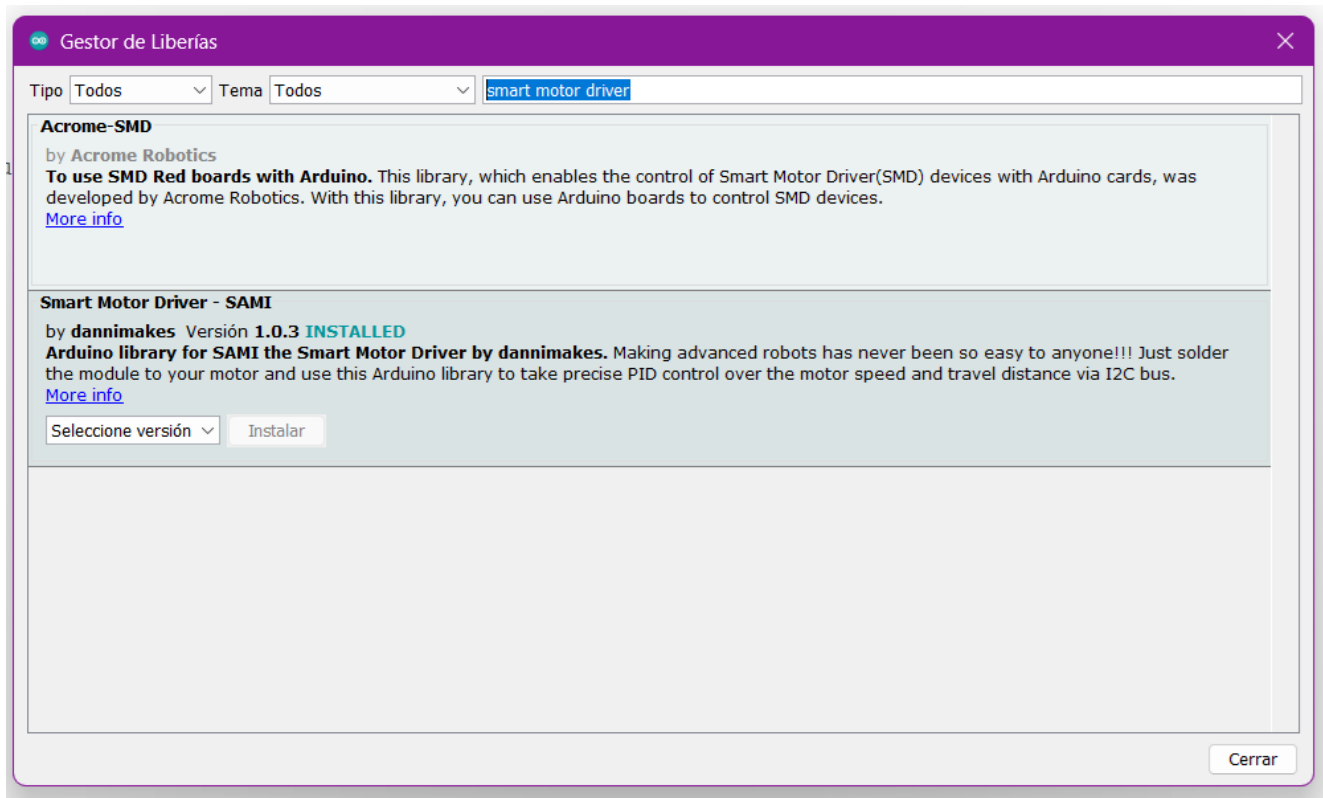
Lo primero que tienes que hacer es descargar el IDE de Arduino, para este caso en especifico necesitas la versión Legacy. Obtenla en este enlace: <https://downloads.arduino.cc/arduino-1.8.19-windows.exe>

Una vez descargado, procede a instalarlo. Tras terminar la instalación abre y programa y sigue estos pasos:

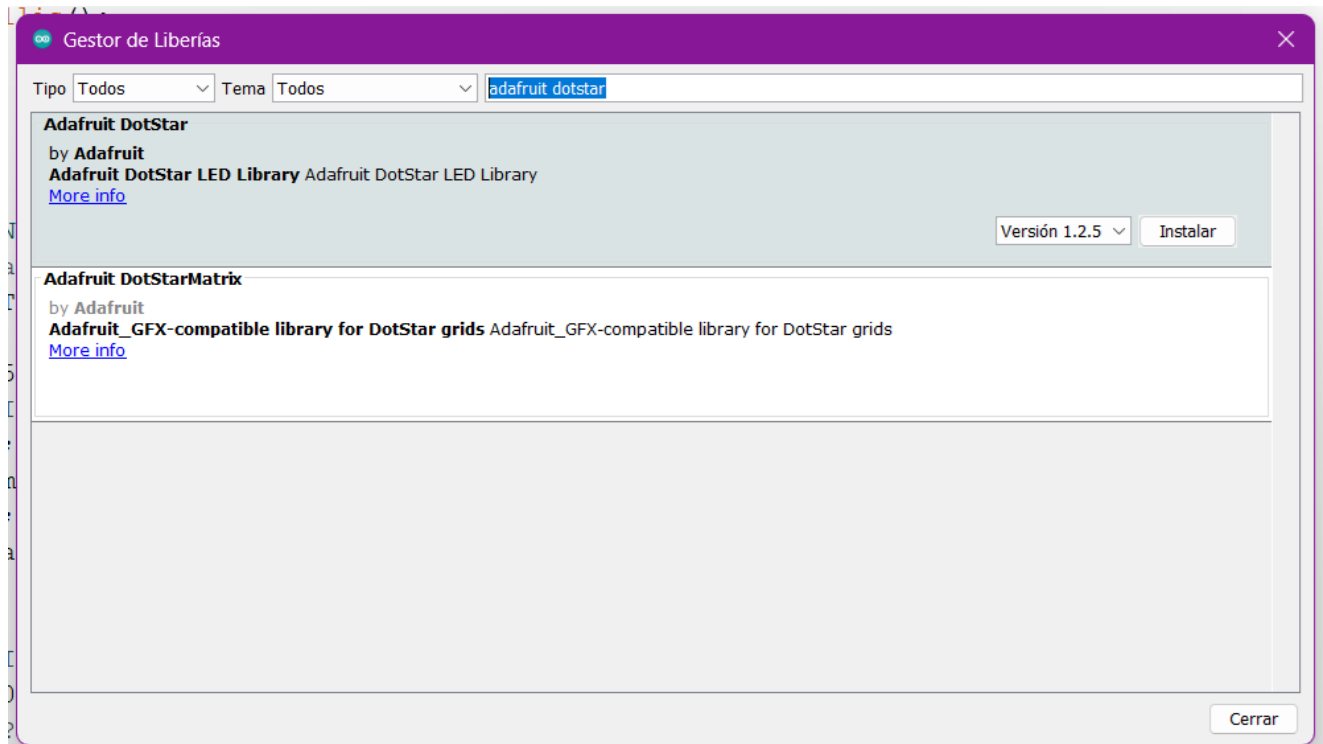
Lo primero será instalar nuestra librería **"Smart Motor Driver"** para Arduino. Dentro del IDE haz clic en el menú **"Programa"**, luego en **"Incluir Librería"** y finalmente en **"Administrar Bibliotecas"**.



Una vez abierto el gestor de librerías da clic en el cuadro de búsqueda e introduce “**smart motor driver**”. Te saldrá una llamada “**Smart Motor Driver – SAMI**” del usuario dannimakes. El resto las puedes ignorar. Haz clic en el botón instalar y espera a terminar el proceso.

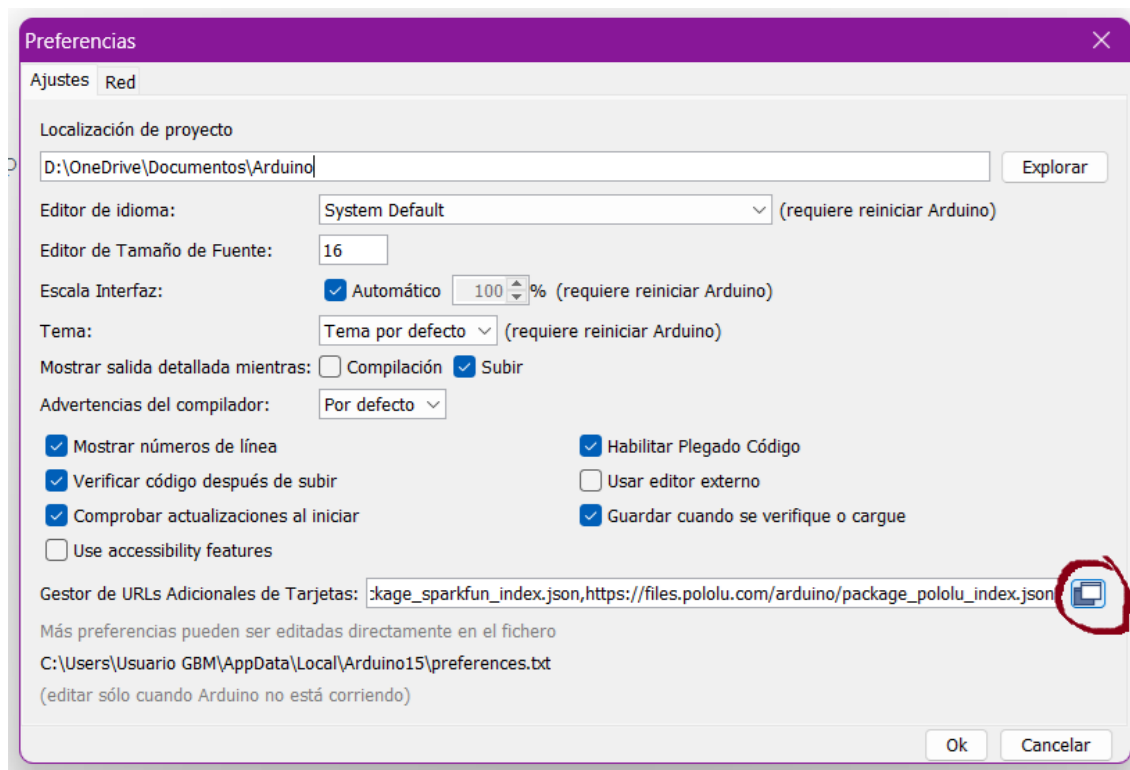


Repita este mismo procedimiento, pero para la librería “**Adafruit Dotstar**”. Te será más sencillo esta vez.



Ya tenemos las librerías necesarias instaladas, ahora tenemos que instalar la tarjeta de control que usa este robot. Al no ser estándar se necesita realizar una serie de pasos adicionales. Pero tranquilo, es muy fácil.

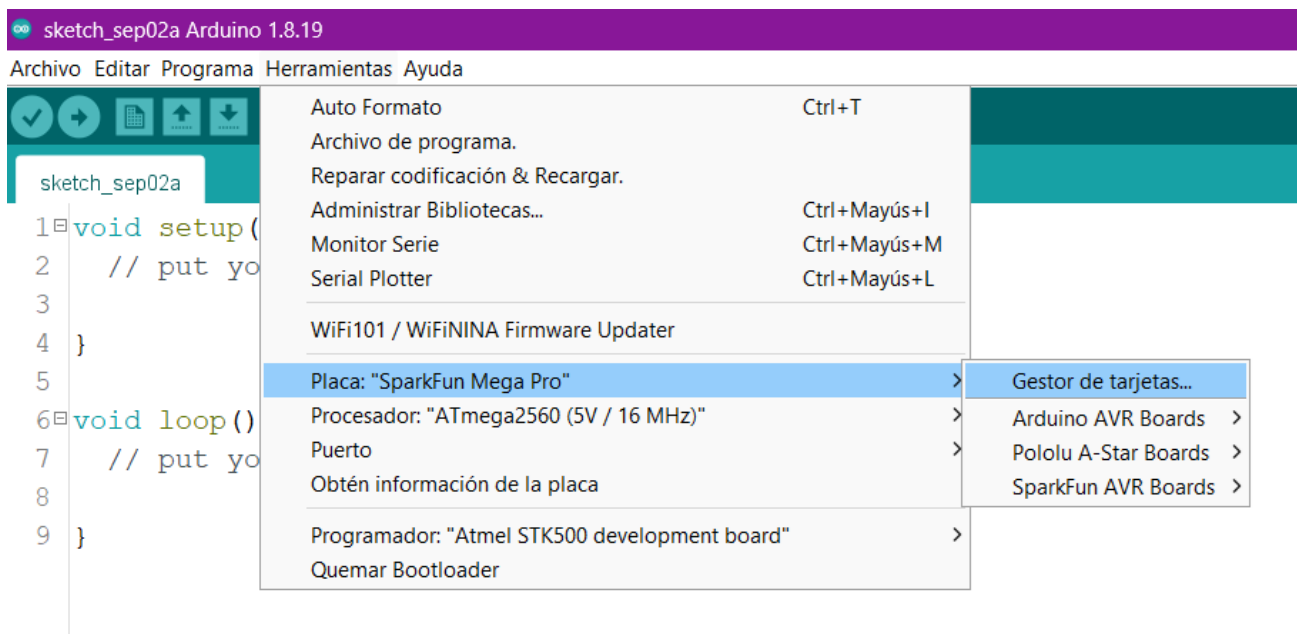
Da clic en el menú **"Archivo"** y después en **"Preferencias"**, te abrirá la siguiente ventana:



Haz clic en el botón que hemos encerrado en el círculo, te abrirá un menú. Copia y pega lo siguiente https://adafruit.github.io/arduino-board-index/package_adafruit_index.json

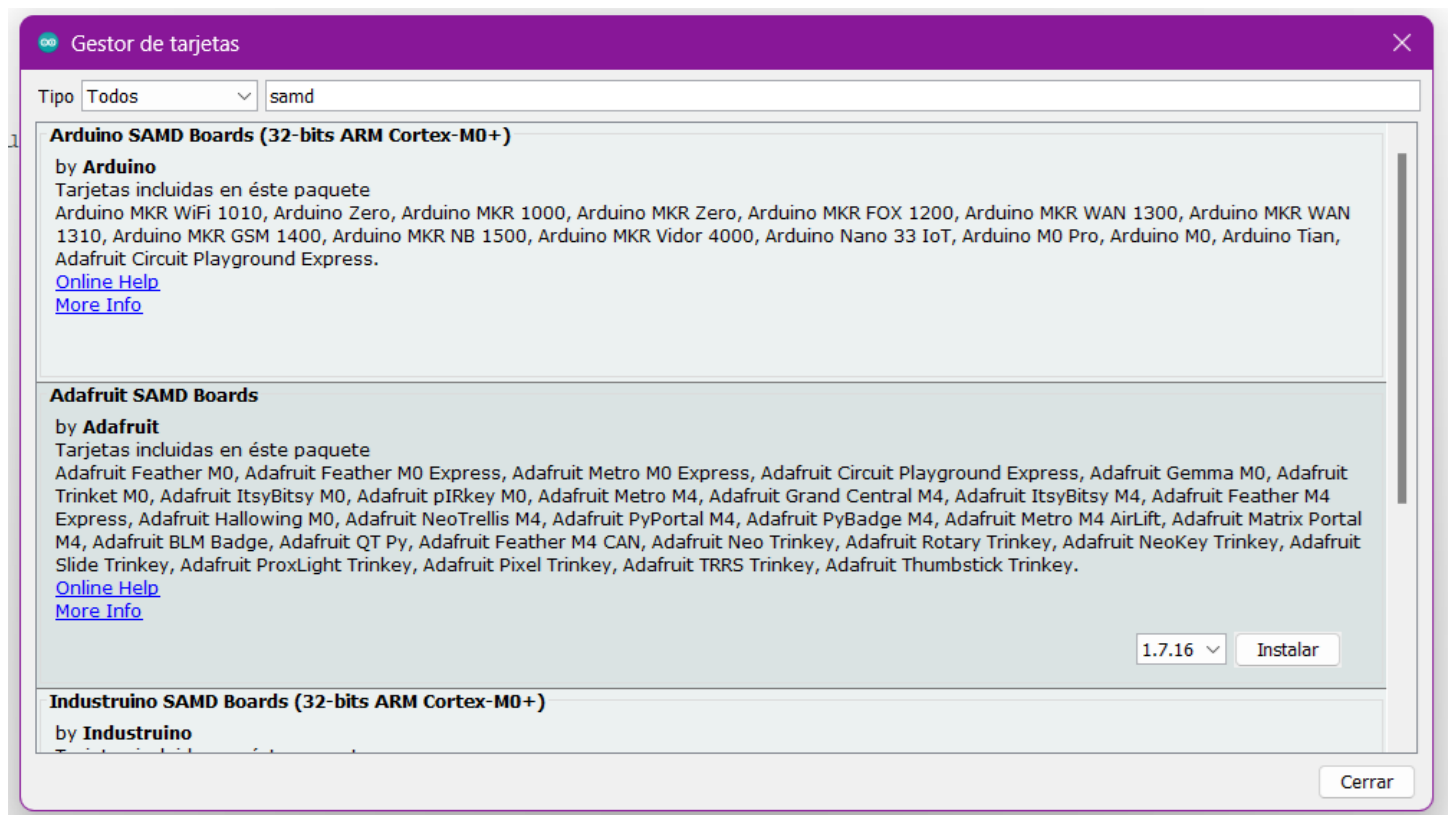
A continuación, da clic en el botón "ok" de ambos cuadros para salir de ahí. Ahora ya podremos proceder a instalar la tarjeta para que lo reconozca el IDE de Arduino.

Ahora dirígete a **"Herramientas"** -> **"Placa"** -> **"Gestor de tarjetas"** y se te abrirá una nueva ventana.

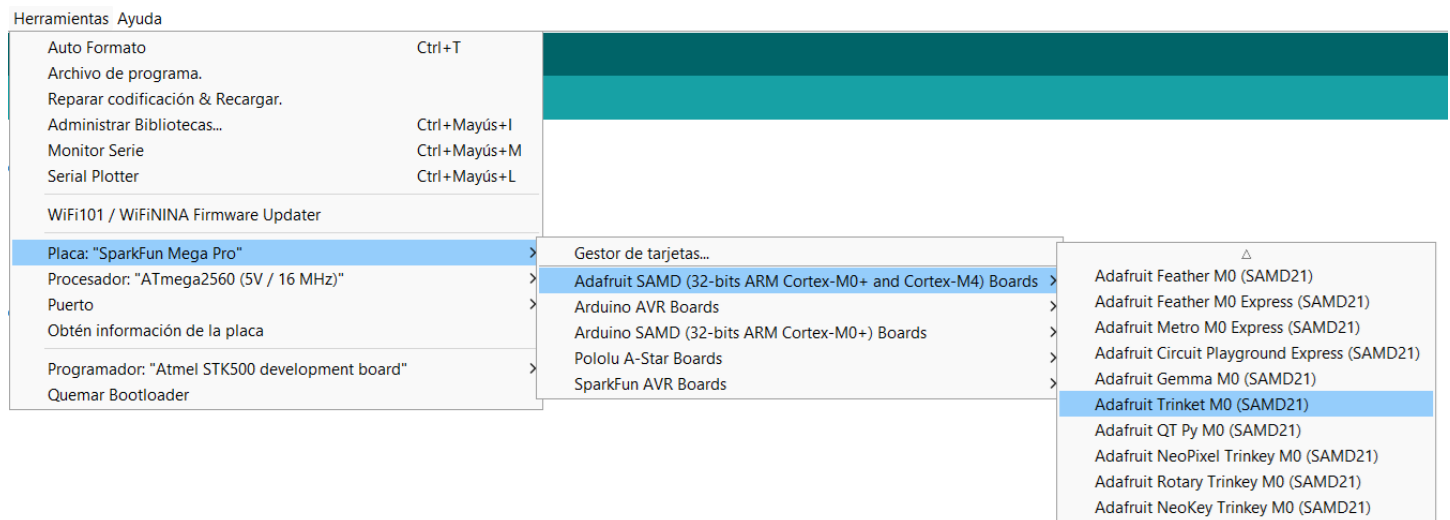


Una vez allí tendrás que instalar varios paquetes, lo realizas del mismo modo que como hicimos con las librerías. Debes instalar los paquetes: “Arduino SAMD” y “Adafruit SAMD”

Te pedirá que instales varias dependencias como parte del proceso, dale “Instalar” a todo lo que te aparezca.



Ahora en el menú abre el menú “Herramientas”-> “Placas” – “Adafruit SAMD” y selecciona “Adafruit Trinket M0”



Con esto ya hemos terminado los preparativos para usar nuestro robot, ahora vamos a cargar nuestro primer programa.

Ya puedes conectar el robot usando el cable micro USB a tu computadora, quizás se instalen un par de drivers. Espera a que el proceso concluya con éxito y una vez terminado podrás seguir con esta guía.

Ejemplo y primeros pasos

El primer paso es crear un sketch nuevo, asegúrate de seleccionar tu tarjeta desde el menú "Herramientas" -> "Puerto". Debería aparecer el puerto COM de tu robot y en paréntesis te saldrá el modelo de la tarjeta. "Adafruit Trinket M0". Selecciona esta opción y a continuación borra todo lo que tenga escrito el editor de Arduino para después copiar y pegar el siguiente código de ejemplo:

```
#include "SAMI_2BRobots.h"  
#include <Adafruit_DotStar.h>
```

```
#define circunferencia 256 //256mm = 360° circunferencia del robot, se calcula con la distancia entre centro de  
llantas siendo este el radio  
//La velocidad maxima de este motor son 430RPM. Relacion de motores 75:1 con ruedas de 42mm, velocidad  
maxima 56 metros/minuto
```

```
SAMI_2BRobots Mizq = SAMI_2BRobots();  
SAMI_2BRobots Mder = SAMI_2BRobots();  
Adafruit_DotStar strip = Adafruit_DotStar(1, INTERNAL_DS_DATA, INTERNAL_DS_CLK, DOTSTAR_BGR);
```

```
void setup()  
{  
  strip.begin();  
  strip.setPixelColor(0, 64, 0, 0); strip.show(); //led red  
  delay(100);
```

```
  Mizq.begin(0x25);  
  Mder.begin(0x26);  
  Mizq.RESET();  
  Mder.RESET();  
  delay(50);
```

```
  Mizq.setMode(3);  
  Mizq.setGear(75);  
  Mizq.setDiameter(42);  
  Mizq.setRPM_PID_KP(1.81);  
  Mizq.setRPM_PID_KD(11.73);  
  Mizq.setRPM_PID_KI(0.18);  
  Mizq.setATS_PID_KP(25.65);  
  Mizq.setATS_PID_KD(0.87);  
  Mizq.setATS_PID_KI(0.00);  
  Mizq.inverted(false);  
  Mder.setMode(3);  
  Mder.setGear(75);  
  Mder.setDiameter(42);  
  Mder.setRPM_PID_KP(1.81);
```

```

Mder.setRPM_PID_KD(11.73);
Mder.setRPM_PID_KI(0.18);
Mder.setATS_PID_KP(5.65);
Mder.setATS_PID_KD(0.87);
Mder.setATS_PID_KI(0.00);
Mder.inverted(true);

strip.setPixelColor(0, 0, 64, 0); strip.show(); //led green
delay(500);
}

void loop()
{
  strip.setPixelColor(0, 0, 0, 64); strip.show(); //led blue
  avanzar(300, 120); //el primer dato es la distancia en mm (puede ser negativo para reversa) y el segundo las
  RPM
  vuelta(90, 80); //el primer datos son los grados que pueden ser positivos o negativos y el segundo dato son las
  RPM
  vuelta(-180, 80);
  vuelta(90, 80);
  avanzar(-300, 120);
  strip.setPixelColor(0, 64, 0, 0); strip.show(); //led red
  while (1); //aqui detenemos el programa indefinidamente
}

void avanzar(int distancia, int rpm) //pasamos la distancia que queremos avanzar en mm y las rpm
{
  rpm = abs(rpm);
  if (distancia < 0)
  {
    Mizq.setSpeed(rpm * -1);
    Mder.setSpeed(rpm * -1);
  }
  else
  {
    Mizq.setSpeed(rpm);
    Mder.setSpeed(rpm);
  }

  Mizq.setDistance(distancia);
  Mder.setDistance(distancia);

  Mizq.Run(1);
  Mder.Run(1);
}

```

```

while (Mizq.isRunning() | Mder.isRunning());
delay(200);
}

void vuelta(int angulo, int rpm) //pasamos los grados que queremos girar y las rpm
{
  rpm = abs(rpm);
  if (angulo > 0) //giro a la derecha
  {
    int d = (int)(((float)circunferencia / (float)360) * (float)angulo); //calculamos los grados proporcionalmente con
    la circunferencia
    Mizq.setSpeed(rpm);
    Mder.setSpeed(rpm * -1);

    Mizq.setDistance(d);
    Mder.setDistance(d * -1);

    Mizq.Run(1);
    Mder.Run(1);
  }
  else //giro a la izquierda
  {
    angulo = abs(angulo);
    int d = (int)(((float)circunferencia / (float)360) * (float)angulo);
    Mizq.setSpeed(rpm * -1);
    Mder.setSpeed(rpm);

    Mizq.setDistance(d * -1);
    Mder.setDistance(d);

    Mizq.Run(1);
    Mder.Run(1);
  }
  while (Mizq.isRunning() | Mder.isRunning());
  delay(200);
}

```

Como pudiste notar este código ya hace una pequeña rutina con tu robot. Ahora cárgalo al robot y podrás comenzar tu aventura. Las secuencias de movimiento se programan dentro de la función “**loop**”.

Cualquier duda o problema no dudes en escribirnos, con gusto te ayudaremos.

hi@2brobots.com