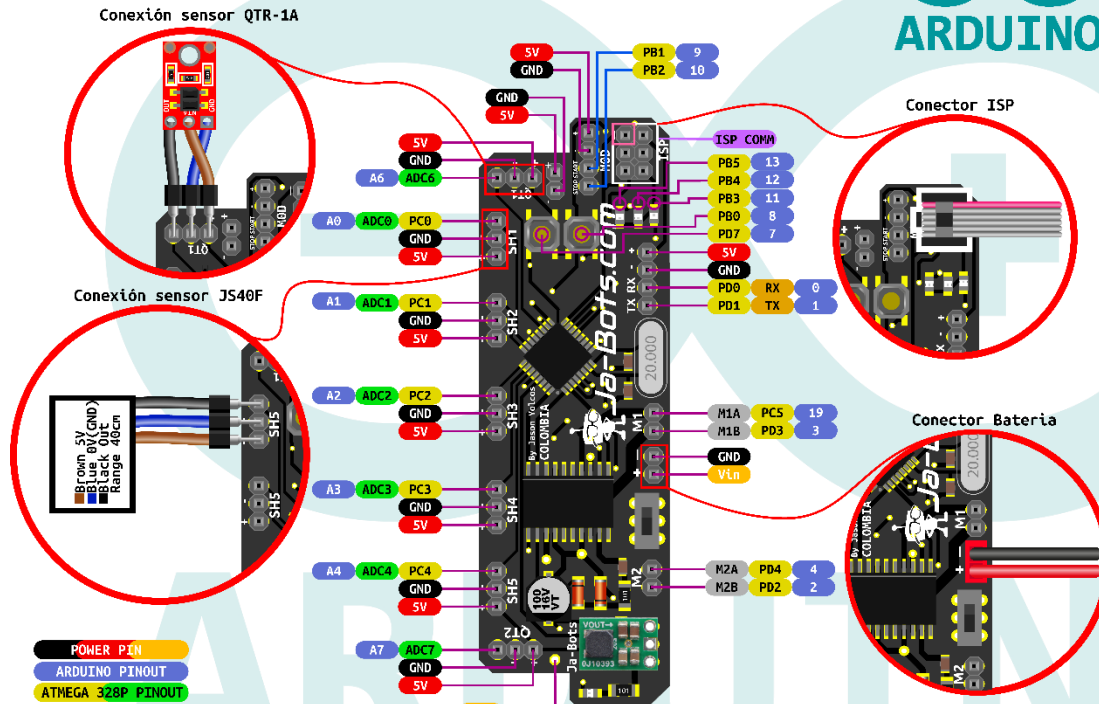


# MINIBLACK- TARJETA PARA MINISUMOS



TM

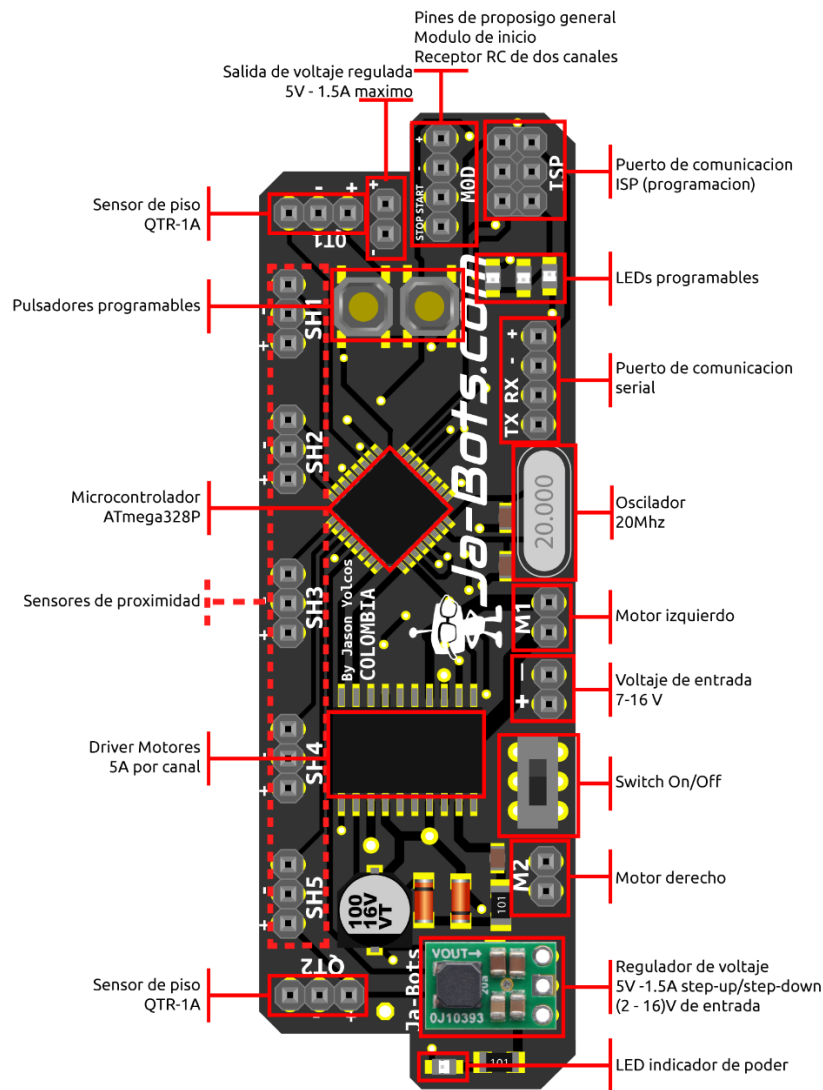


La tarjeta MiniBlack está diseñada principalmente para robots minisumos de competencia, la cual es utilizada por el robot minisumo Asashoryu, ganador de más de 30 torneos nacionales e internacionales. Cuenta con una arquitectura AVR de 8 bits, para procesos de alto rendimiento con un increíble diseño que se adapta a todas las necesidades.

## Descripción

- Microcontrolador: ATMEGA328P.
  - Flash: 32K Bytes.
  - EEPROM: 1K Byte.
  - RAM: 2K Bytes.
  - Rendimiento: alto rendimiento/baja potencia.
- Alimentación de Entrada: (7 – 16) V.
- Drivers: 5A por canal (2 canales) – (2 motores).
- Pulsadores: 2.
- Leds: 3 (Leds de propósito general recomendados para visualizar estado de los sensores).
- Entrada: módulo de inicio.
- Regulador: Regulador de tipo Step Down para inmunidad a caídas de tensión.
- Asociable a Arduino IDE.

Cuenta con protección para cortocircuito y corriente inversa (únicamente para el microcontrolador, el **driver** de los motores **no** cuenta con esta protección, por lo que se debe ser cuidadoso a la hora de conectar la batería, para no hacerlo de manera incorrecta).



## Puertos de comunicación

- Serial RS232 (PD0 y PD1).
- ISP para programación
- Pines PB1 y PB2 son de propósito general, pero en este diseño se usan, bien para conectar el módulo de inicio, necesario en las competencias de minisumo, o para ser usado mediante radio control de dos canales para manipular el robot de forma remota. Esto para las competencias de minisumo RC.

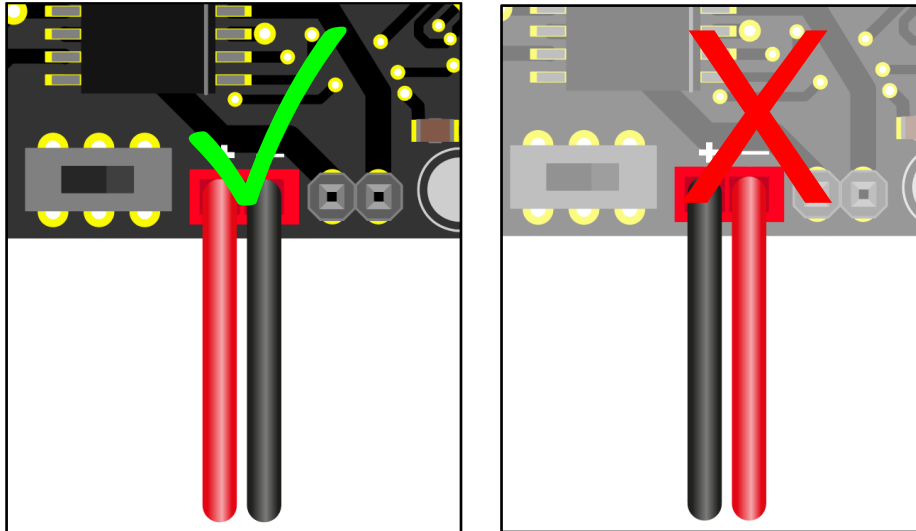
## Pines programables

- 7 pines E/S análogos, 5 los cuales a su vez son digitales
- 4 pines E/S de propósito general.
- 2 pares de pines PWM cada uno para un motor.
- 3 pines asociados a Leds.
- 2 pines asociados a pulsadores.

Basados en las competencias de minisumo se ha ido mejorando el diseño de la tarjeta, es por esto que cada uno de los pines destinados para los sensores cuenta con sus pares de alimentación independientes, esto para hacer más fácil el cambio o la instalación de un nuevo sensor en caso de ser necesario.

## PRECAUCIONES

Si se va a conectar la batería y energizar la tarjeta, hay que hacerlo de forma correcta como se indica en el siguiente esquema, cabe aclarar que si la batería se conecta de manera incorrecta puede generar daños irreversibles en los componentes de la tarjeta y principalmente en el driver de los motores, esto debido a que la protección de voltaje inverso solo esta puesta para el microcontrolador, ya que si se usa el diodo en el driver se disminuye el rendimiento del robot.



## Programación Miniblack +ARDUINO

La tarjeta miniblack desarrollada por la empresa JA-BOTS cuenta con un microcontrolador ATmega328P desafiante en el entorno de desarrollo, además de ser programada desde el ISP mediante el modulo programador de Pololu AVR y con la plataforma Arduino.

A continuación, se detallarán los pasos a seguir para programar la tarjeta miniblack.

### Tabla de contenido.

[Paso 1. Instalar software para programa.](#)

[Paso 2. Instalación del driver para el programador de Pololu AVR V2.](#)

[Paso 3. Conexión para programar:](#)

[Paso 4. Agregar el programador en Arduino:](#)

[Paso 5. Crear nuevo proyecto:](#)

[Paso 6. Código de ejemplo:](#)

[Paso 7. Cargar programa:](#)



## Pines físicos y para programación en Arduino y Atmel Studio (según nuestros códigos de Ejemplo)

PORT	PIN Name	Atmel Name	Arduino Name	Function
B	PB0	SW1	8	Pulsador 1 / Digital
	PB1	MSTA	9	Modulo start / Digital
	PB2	MSTO	10	Modulo stop / Digital
	PB3	Led3	11	MOSI / Led 3 Salida Digital
	PB4	led2	12	MISO / Led 2 Salida Digital
	PB5	Led1	13	SCK / Led 1 Salida Digital

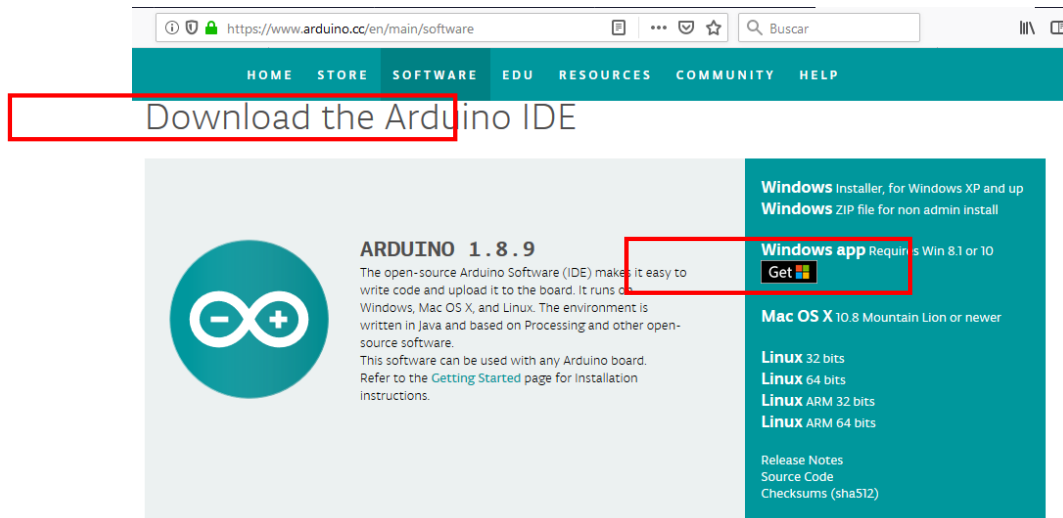
C	PC0	S1	A0	Sharp 1 / digital o analogo
	PC1	S2	A1	Sharp 2 / digital o analogo
	PC2	S3	A2	Sharp 3 / digital o analogo
	PC3	S4	A3	Sharp 4 / digital o analogo
	PC4	S5	A4	Sharp 5 / digital o analogo
	PC5	IN1A	A5	Dirección Motor A / Salida Digital

C	ADC6	q2	A6	Sensor Linea 2/ Analogo
	ADC7	q1	A7	Sensor Linea 1/ Analogo

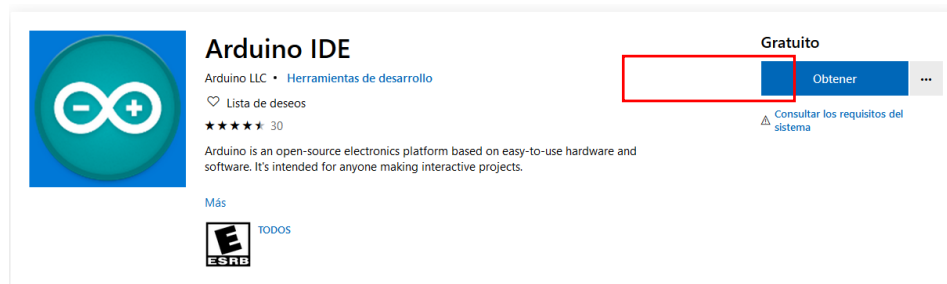
D	PD0	RX	0	Comunicación Serial
	PD1	TX	1	Comunicación Serial
	PD2	IN2B	2	Direccion Motor B / Salida Digital
	PD3	IN1B	3	Direccion Motor B / Salida Digital
	PD4	IN2A	4	Direccion Motor A / Salida Digital
	PD5	PWM1	5	Velocidad del motor A
	PD6	PWM2	6	Velocidad del motor B
	PD7	SW2	7	Pulsador 2 / Digital

### Paso 1. Descarga e instala arduino IDE

Para instalar Arduino IDE basta con ir a la página oficial para descargarlo, o a través del siguiente enlace <https://www.arduino.cc/en/main/software>, donde nos desplazamos a la parte inferior hasta encontrar "Download the Arduino IDE" como se muestra en la imagen, si tienes Windows 8.1 o Windows 10 la descarga se realizara mediante la tienda de Windows, descargamos el archivo que nos convenga y luego procedemos a instalar la aplicación.



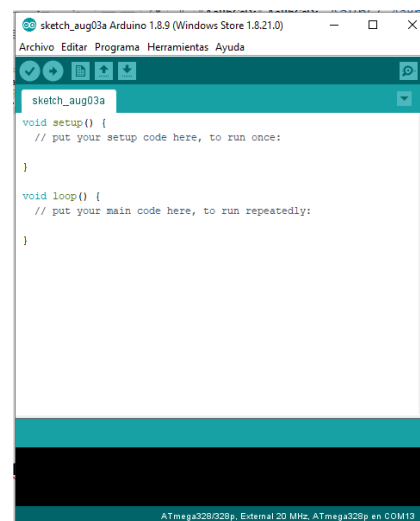
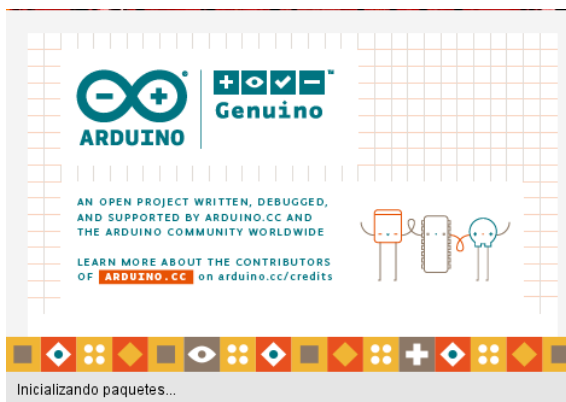
Nosotros descargaremos la versión para Windows 10 a través de la tienda, basta con darle click en obtener y nos dirigirá a la tienda para iniciar la descarga e instalación automática.



Una vez instalado, podemos buscar el icono como se muestra a continuación.



en el inicio para abrir la aplicación, la cual empezara



**Paso 2. Instalación del driver para el programador de Pololu AVR V2.**



Para que nuestro computador pueda reconocer el programador de Pololu como tal, debemos instalar los drivers necesarios, los cuales los podemos encontrar en la página oficial de Pololu, o en el siguiente enlace <https://www.pololu.com/product/3170/resources> el cual nos dirigirá a la sección de recursos.

https://www.pololu.com/product/3170/resources

Description Specs (2) Pictures (11) Resources (20) FAQs (1) On the blog (1)

Documentation and other information

**Pololu USB AVR Programmer v2 User's Guide (Printable PDF)**  
User's manual for the Pololu USB AVR Programmer v2.

**Pololu AVR Programming Quick Start Guide (Printable PDF)**  
This guide explains how to get started programming your Orangutan or 3pi Robot in Windows, Linux or Mac OS X. It covers setting up an AVR development environment (Atmel Studio for Windows users), installing the Pololu AVR C/C++ Library, and setting up the Pololu USB AVR Programmer.

File downloads

**Pololu USB AVR Programmer v2 Software and Drivers for Windows (9MB msi)**  
This installer contains the drivers and software for the Pololu USB AVR Programmer v2 for Microsoft Windows.

**Pololu USB AVR Programmer v2 Software for Linux (x86) (8MB xz)**

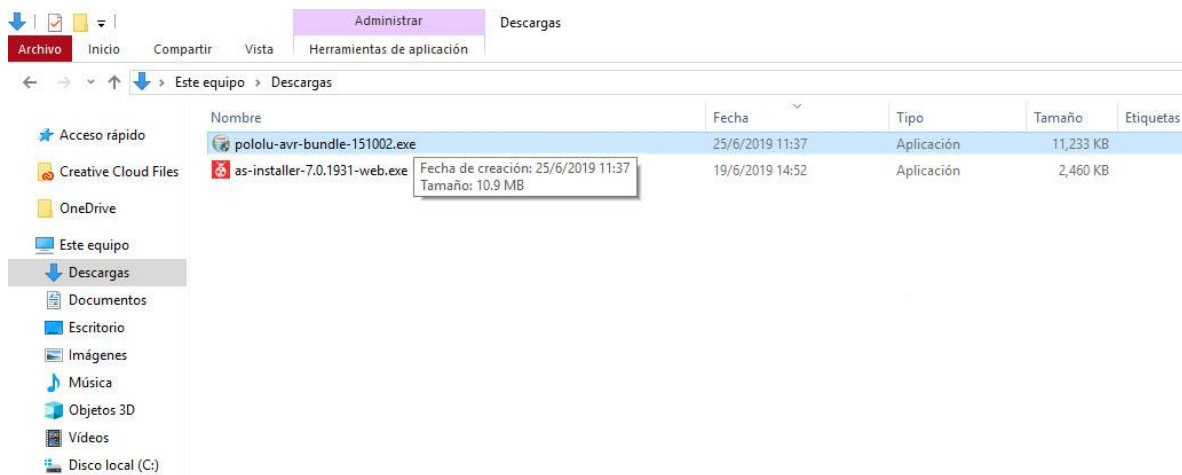
**Pololu USB AVR Programmer v2 Software for Linux (Raspberry Pi) (6MB xz)**

**Pololu USB AVR Programmer v2 Software for Mac OS X (7MB pkg)**  
This installer contains the software for the Pololu USB AVR Programmer v2 for Mac OS X.

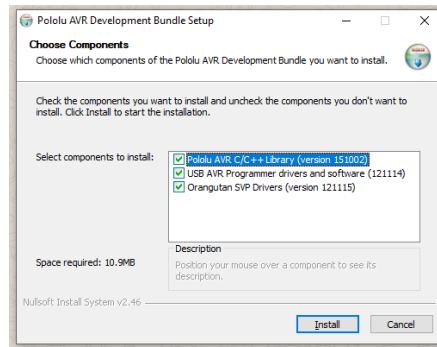
**Pololu AVR Development Bundle for Windows (12MB exe)**  
This bundle contains all the Pololu software you need to get started programming AVR in Windows: the Pololu AVR C/C++ Library, the Pololu USB AVR Programmer drivers and software, and the Pololu Orangutan SVP drivers. We recommend installing Atmel Studio 7.0 before installing this bundle.

Estando ahí, nos desplazamos abajo hasta encontrar la subsección de “File downloads” y descargas el archivo que se ajuste a tus necesidades dependiendo del sistema operativo, pero si estás trabajando con Windows recomendamos descargar el archivo que dice “Pololu AVR Development Bundle for Windows” *si estas utilizando Atmel Studio 7.0 ya que además de los drivers para el reconocimiento del programador, cuenta con librerías de pololu, como lo es una para la baby orangután, o el robot 3pi entre otros.*

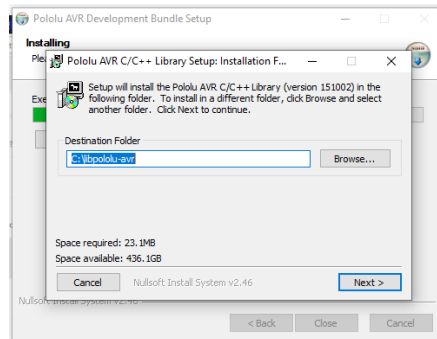
Una vez descargado, daremos doble click sobre el archivo para realizar la instalación.



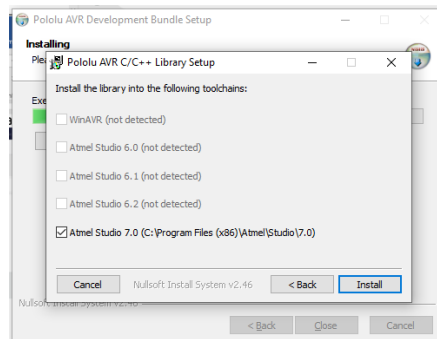
Le daremos permisos para realizar los cambios en el equipo y luego se nos mostrara la siguiente ventana donde nos preguntara que componentes queremos instalar, para esto dejaremos todo seleccionado y le daremos click en “install” para comenzar con la instalacion.



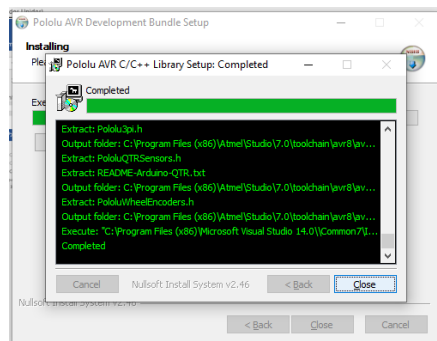
Hecho esto, se nos abrirá una ventana donde nos pedirá una ruta para instalar la librería de pololu, dejamos la ruta por defecto y le damos “next” para continuar.



En caso de estar utilizando Atmel Studio 7.0 el programa de instalación reconocerá la carpeta donde cargará la librería para este programa.

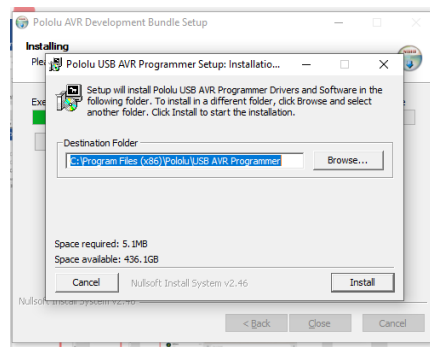


Esperamos a que se realice el proceso y damos click en “close” para continuar

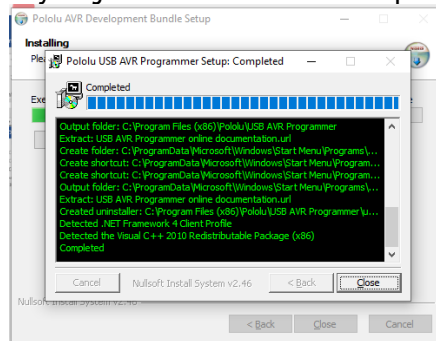


Luego se nos abrirá otra ventana emergente donde instalará el controlador del usb programmer, dejamos la ruta por defecto y le damos click en “install” para continuar

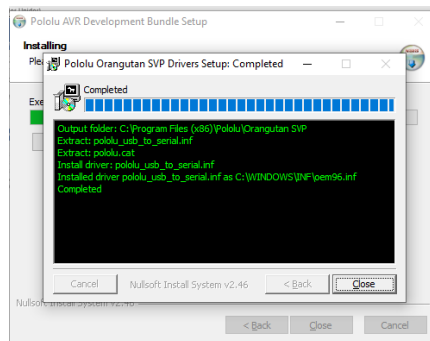




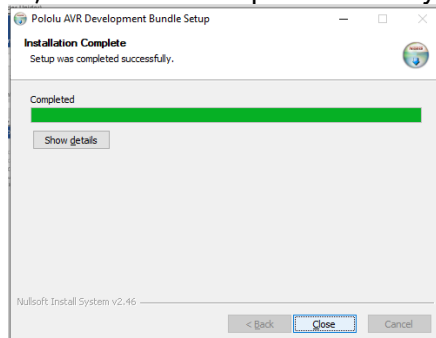
Esperamos que termine el proceso de instalacion y luego le damos click en “close” para continuar



Hecho esto, se nos abra una ultima ventana emergente donde nos solitara de nuevo una ruta de instalacion, simplemente como en los casos anteriores dejaremos la ruta por defecto y daremos click en “install” para continuar, esperamos que termine el proceso y luego le daremos click en “close” para continuar



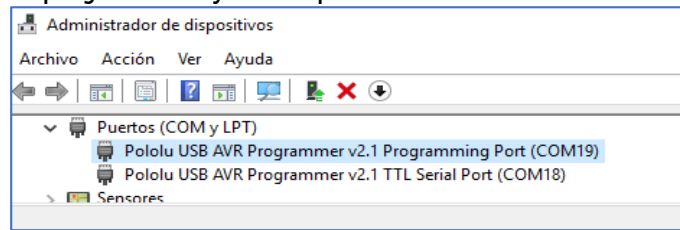
Con eso se finalizaria la instalacion de los drivers, se le da close en la primer ventana y con esto, estarian los drivers instalados.



En caso de tener abierto Arduino IDE cierra y abre de nuevo la aplicación para que los cambios efectuados por el instalador de los drivers tengan efecto.

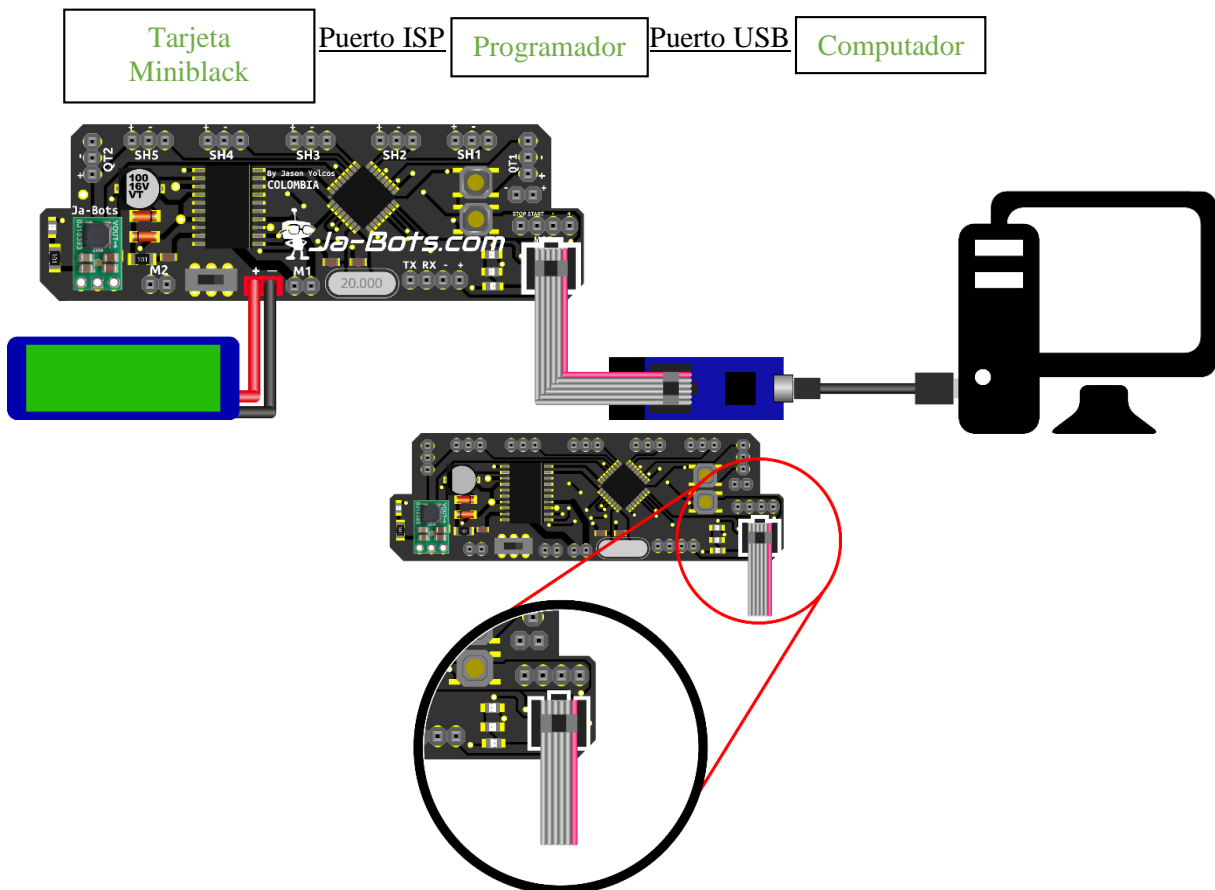
Podemos verificar la correcta instalación de los drivers del programador porque al conectarlo nos lo reconocerá de la siguiente manera si lo buscamos mediante el administrador de dispositivos (*para abrir el administrador de dispositivos,*

*basta con buscarlo en el inicio y darle click), nos dirigimos a la sección de puertos (COM y LPT) donde podemos ver como asigna dos puertos COM, uno para programación y el otro para comunicación serial*



### Paso 3. Conexión para programar:

Luego de tener el software totalmente instalado, conectamos el programador con la tarjeta mediante el puerto de programación ISP y a su vez el programador mediante USB al computador.

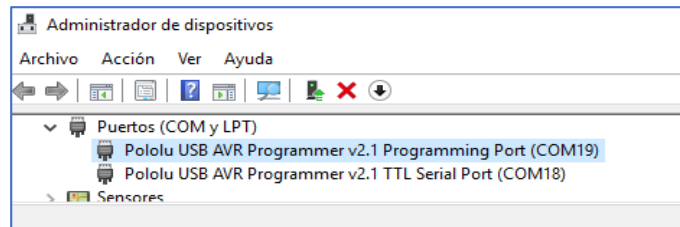


*En la imagen se indica el puerto de ISP de programación, así como su correcta disposición al momento de conectar al programador.*

**Nota:** Si la tarjeta no está energizada y queremos programarla, el programa nos mostrara un error, por lo cual si vas a programarla debes tenerla energizada y encendida.

### Paso 4. Verificar conexión entre el programador y el computador:

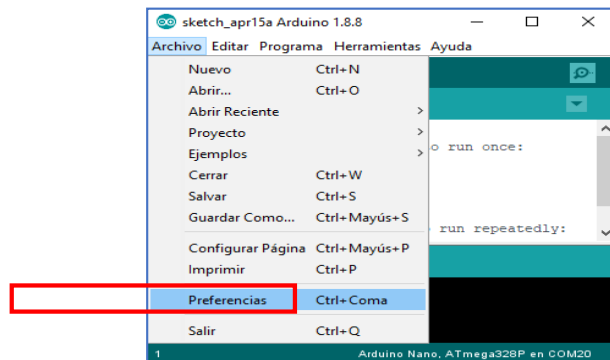
Luego de realizar la anterior conexión, nos dirigimos al *administrador de dispositivos* del computador y hacemos doble click en *puertos (COM Y LPT)* y verificamos que el programador se encuentre conectado como se muestra a continuación.



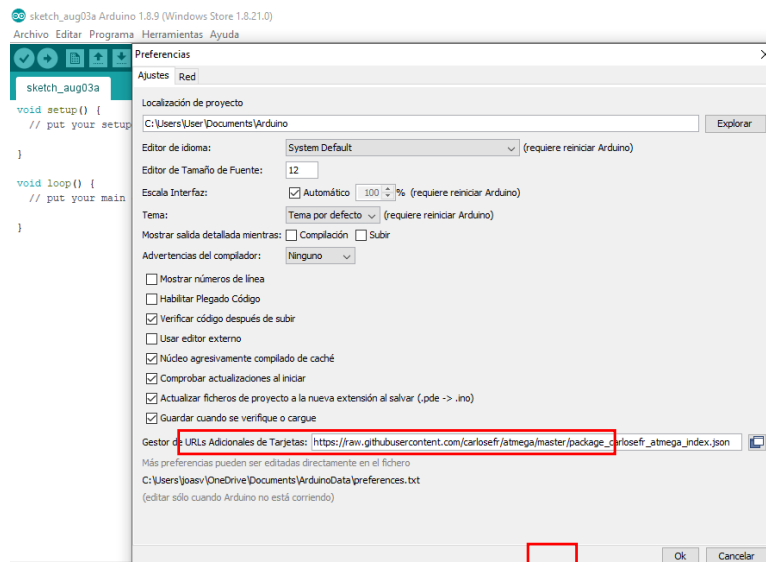
Una vez hecho esto, abrimos Arduino IDE

## Paso 5. Configuración aplicación Arduino IDE:

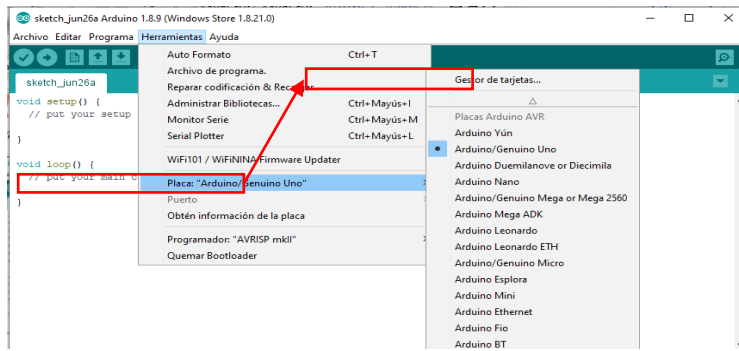
Estando en la aplicación, nos dirigimos a archivo y luego damos click en preferencias



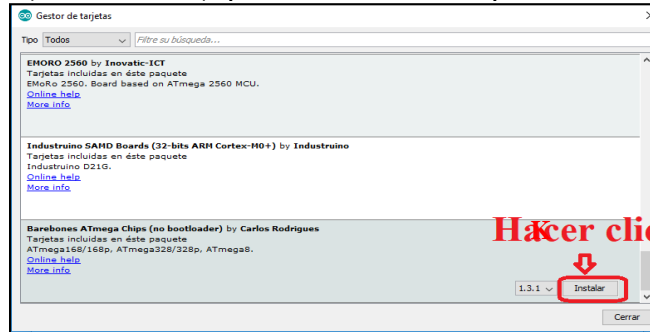
Se nos abrirá una ventana como la que se muestra a continuación en donde se copiará el siguiente enlace [https://raw.githubusercontent.com/carlosefr/atmega/master/package\\_carlosefr\\_atmega\\_index.json](https://raw.githubusercontent.com/carlosefr/atmega/master/package_carlosefr_atmega_index.json) en Gestor de URLs Adicionales de Tarjetas, esto con el fin de que Arduino IDE pueda reconocer y hacer uso del programador de Pololu AVR así como del microcontrolador que estamos usando, el ATmega328P



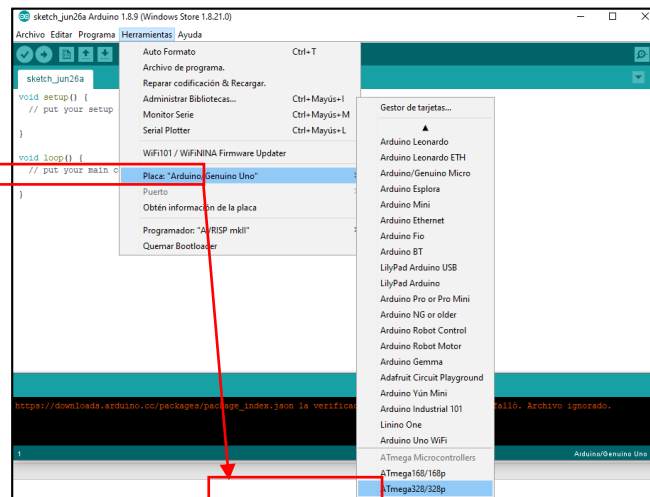
A continuación, nos dirigimos a herramientas y luego a gestor de tarjetas como se indica en la imagen.



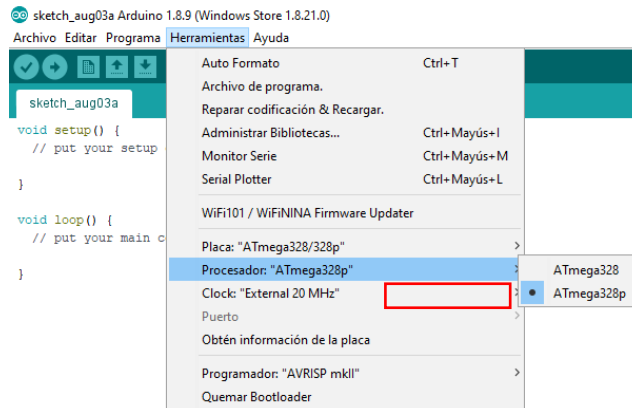
Después de hacer click en *gestor de tarjetas* se muestra una ventana como la que puedes ver a continuación, en donde se instalara *Barebones Atmega Chips (no bootloader)* que se encuentra en la parte inferior de los paquetes.



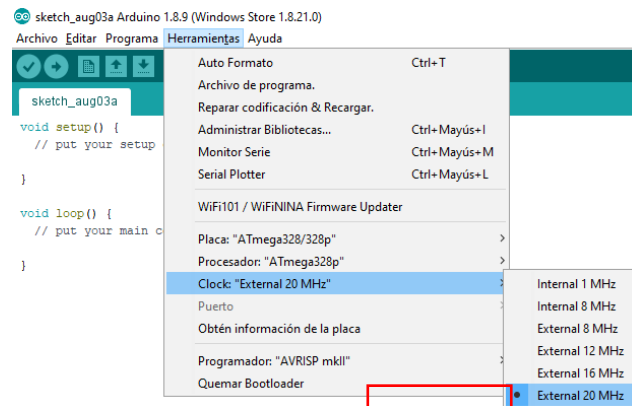
Hecho esto, nos cargara una nueva tarjeta la cual vamos a elegir y es ATmega328/328p, esta se puede encontrar en *Herramientas/Placa:* y hacer click en *ATmega 328/328P para elegirla*.



Ahora debemos seleccionar el procesador, para esto vamos a *Herramientas/procesador* y hacer click en *ATmega328P*



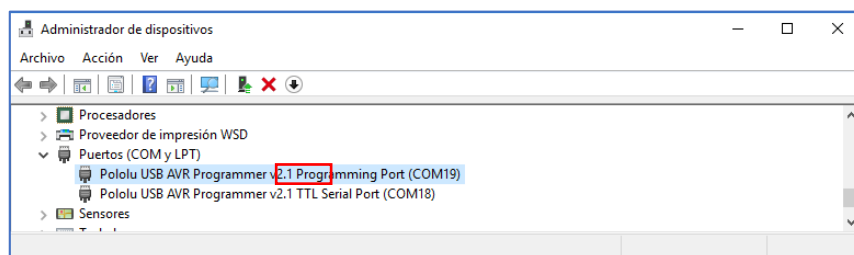
También debemos seleccionar el reloj que usa la tarjeta minibrack, por eso iremos a *Herramientas/Clock* y seleccionamos *External 20 MHz*.



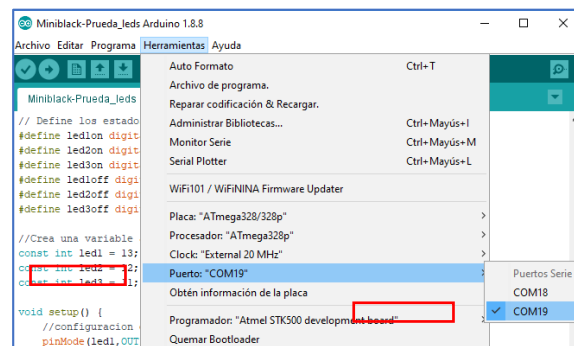
## Paso 6. Configurar programador de pololu avr

Para eso, debemos buscar el puerto COM que vamos a utilizar, este está determinado por la asignación del computador, el cual vimos en el paso 4

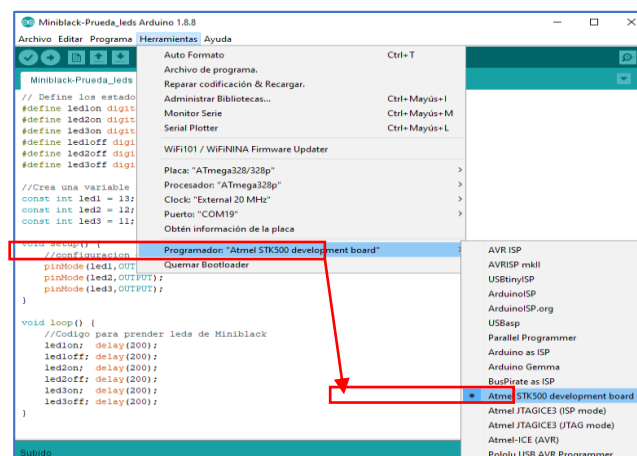
Entonces nos dirigimos a *administrador de dispositivos* de su computador y verificar el *COM* del programador, este debe ser el programming port.



Ahora en la plataforma de Arduino y *Herramientas/Puerto:* y seleccionamos el *COM*, en este ejemplo es el *COM19* que pertenece al *programming port* (en dado caso que no te aparezca como COM19 no importa, tu computador automáticamente le da el nombre al puerto, por lo cual te puede aparecer como COM3, COM5 etc. lo importante es que debe ser el *programming port*).



Procedemos a escoger el programador, para eso nos dirigimos a *Herramientas/programador* y seleccionamos *Atmel STK500 development board*.



Habiendo realizado estos pasos, ya podemos proceder a programar en arduino IDE para lo cual podemos hacer uso del siguiente código de ejemplo

## Paso 7. Programar ejemplo en miniblack:

Copiamos el siguiente ejemplo en el compilador de Arduino IDE que consta de una secuencia de encendido de los tres leds de la tarjeta Miniblack.

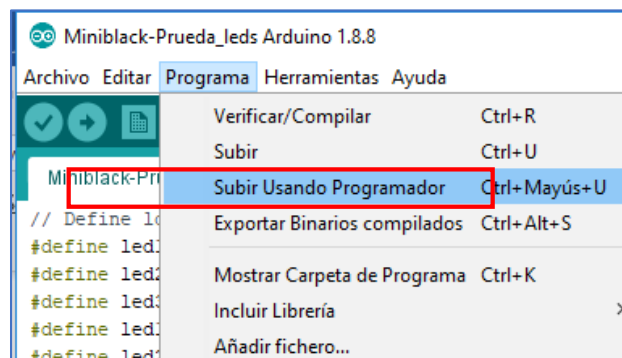
```
// Define los estados HIGH,LOW de cada led
#define led1on digitalWrite(led1, HIGH);
#define led2on digitalWrite(led2, HIGH);
#define led3on digitalWrite(led3, HIGH);
#define led1off digitalWrite(led1, LOW);
#define led2off digitalWrite(led2, LOW);
#define led3off digitalWrite(led3, LOW);

//Crea una variable a cada led
const int led1 = 13;
const int led2 = 12;
const int led3 = 11;

void setup() {
  //configuracion de condiciones iniciales
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
  pinMode(led3,OUTPUT);
}

void loop() {
  //Codigo para prender leds de Miniblack
  led1on; delay(200);
  led1off; delay(200);
  led2on; delay(200);
  led2off; delay(200);
  led3on; delay(200);
  led3off; delay(200);
}
```

Después de copiar el código ir a *Programar* y hacer click en *subir usando programador*. Luego verificamos la secuencia de los leds en la tarjeta Miniblack.

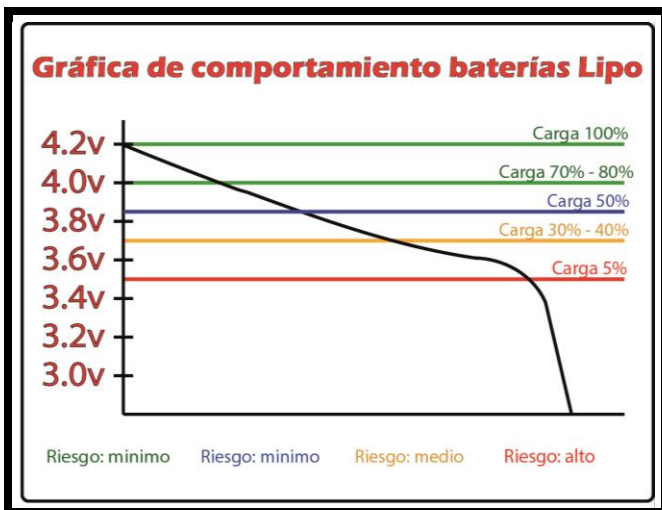


## Secuencia programada de los Leds



Con esto se garantiza que los pasos fueron exitosos y ahora es tu turno de demostrar tus capacidades como programador o Aprendiz para sacar el mayor provecho de esta increíble herramienta que te ofrece la empresa JA-BOTS.COM

## Recomendaciones de uso para la batería



Las baterías lipo se manejan por celdas, cada una cuenta con un voltaje de 3.7v dependiendo del número de celdas este voltaje se duplica o se triplica como es el voltaje de 7.4v para las baterías de dos celdas y 11.4v, Estos voltajes se ven incrementados al cargar totalmente las baterías ya que cada celda totalmente cargada puede alcanzar hasta los 4.3v de modo que hay que mantener los voltajes de la batería a un límite de 3.7v ya que dejar descargar la batería por debajo de este valor podría deteriorar la batería más rápidamente y disminuir el tiempo de su ciclo útil, en casos extremos las celdas de la batería pueden entrar en corto, en este caso puede contactarnos para dar mantenimiento a la batería, sin embargo esto es un coste que deberá asumir el usuario por lo que se recomienda seguir estas indicaciones al pie de la letra para evitar cualquier inconveniente.

En cuanto al uso de la batería lipo se recomienda encarecidamente revisar continuamente los niveles de carga, ya que al ser una batería pequeña los tiempos de descarga podrían ser cortos dependiendo la intensidad de uso dada a fastbot.

El comportamiento de descarga de una batería no es lineal, de modo que antes de que la batería llegue al límite de los 3.7v se descargará de forma lenta pero al superar este límite lo hará de una forma mucho más rápida por lo que nuevamente recomendamos no exceder este valor para el correcto funcionamiento.



Todos los productos se entregan testeados y ensayados para dar mayor seguridad a nuestros clientes, cualquier duda o pregunta fuera de lo ya mencionado en este tutorial puede ser consultado, asegúrese de leer todo el documento antes de hacerlo.

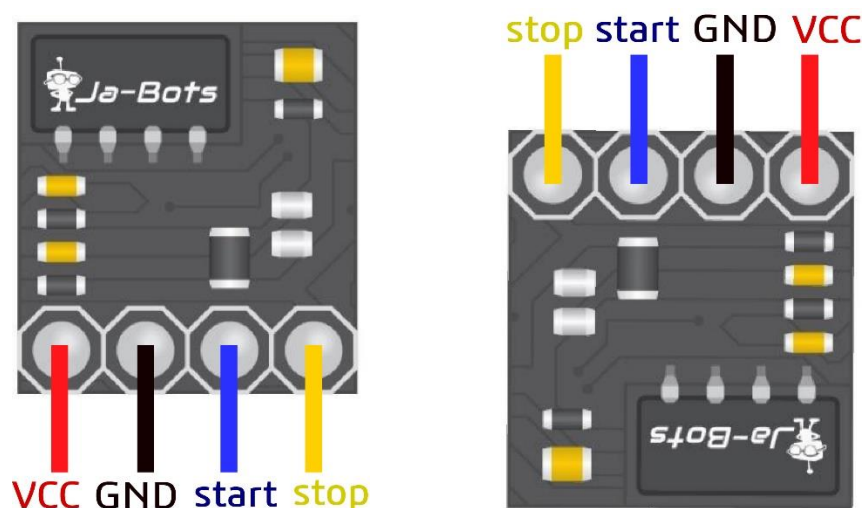
## Configuración control RC

Miniblack puede ser utilizado con un control RC, en nuestra página busca el archivo llamado [Tutorial MiniRC](#) en el encontrarás las conexiones y descripciones generales para el uso del control RC en la competencia.



## Configuración módulo de inicio

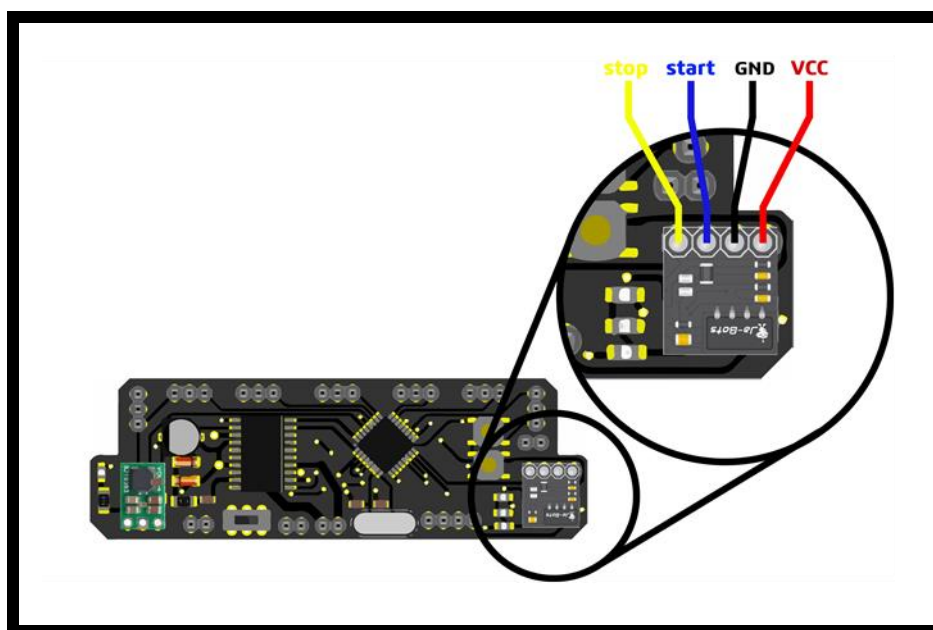
A continuación se describe las características y generalidades sobre el módulo. El módulo cuenta con 4 pines de conexión que son VCC – GND – Start – Stop (este último es opcional) y un tamaño de tan solo 12.5mm x 11.5mm



PIN Name	Atmel Name	Arduino Name	Function
5V	-	-	Alimentación de 5v
GND	-	-	Tierra del circuito
PB2	MSTA	9	Modulo start (pin digital)
PB3	MSTO	10	Modulo stop (pin digital)

Al momento de poner el módulo en miniblack esté seguro de seguir las indicaciones dadas, de otro modo podría generar una mala conexión y dañar el microcontrolador que viene en el módulo de forma irremediable.

El módulo de inicio debe ubicarse de modo que el receptor ubicado en la tarjeta quede hacia la parte trasera de la tarjeta o de otra forma, siguiendo el siguiente esquema de conexiones:



El módulo de inicio está diseñado para proveer funcionalidades y la calidad al momento de competir como de asegurar un ambiente libre de cualquier tipo de trama o falla al momento de poner en marcha los robots, por esta misma razón es necesario explicar a detalle el funcionamiento y método de empleo que el usuario debe seguir al momento de utilizar el control profesional y el módulo de inicio para las pruebas.

El módulo de inicio tiene 3 posibles estados dependiendo del proceso en el que se encuentre o que genera 3 combinaciones diferentes las cuales se presentan a continuación:

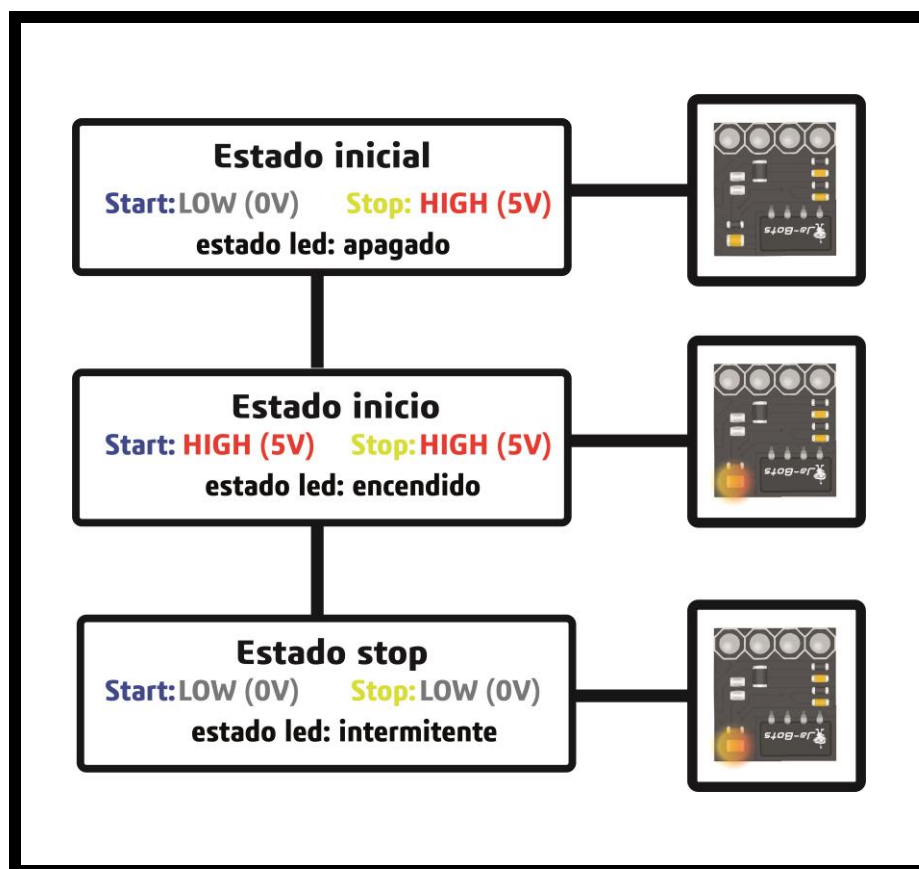
Estado	Start	Stop
Condiciones iniciales	LOW	LOW
Al presionar start	HIGH	LOW
Al presionar stop	LOW	HIGH

Sin embargo se presenta un diagrama de estados donde de forma más gráfica se pueden apreciar las distintas configuraciones posibles y el proceso que normalmente se lleva a cabo hasta el final.

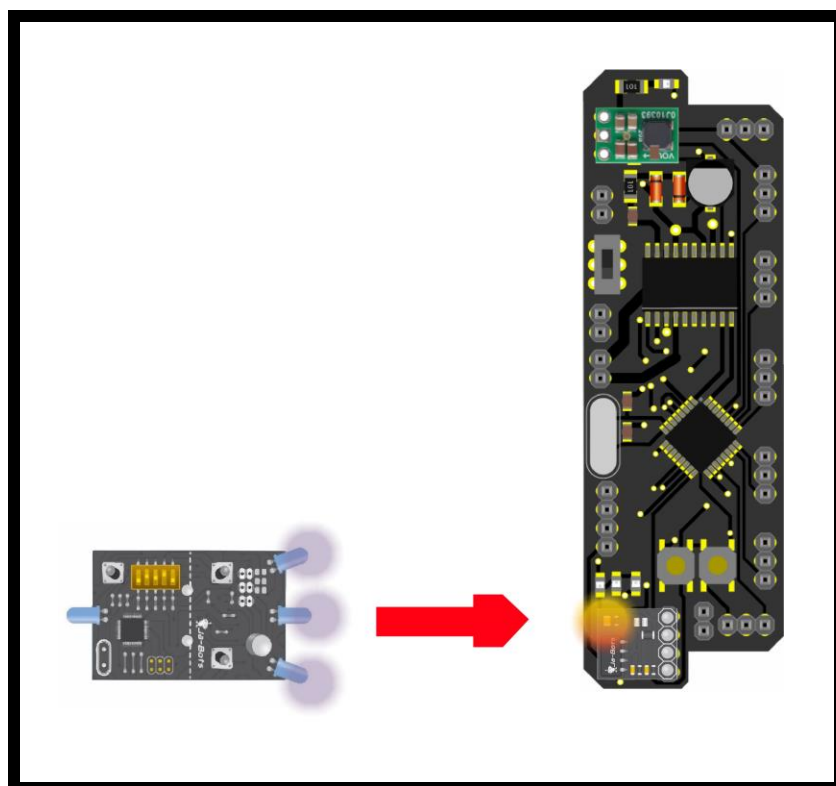
El led indicativo estará apagado en la primer fase que es la del estado inicial marcando solamente el pin de Start en LOW mientras que Stop se encuentra encendido.

El led indicativo estará encendido de forma continua siempre que se le envíe la señal de start al módulo, **este estado no se deshabilitará aún apagando o encendiendo el dispositivo ya que los estados son guardados directamente en la memoria EEPROM, tenga cuidado al momento de utilizar el robot y el estado en que queda establecido.**

Finalmente el estado de Stop generará una señal de luz intermitente con el led, este estado se cambiará al estado inicial al apagar y encender de nuevo el robot.



Estos módulos están diseñados para usarse con un control profesional, en caso de necesitar más información remítase a [Ja-Bots.com](http://Ja-Bots.com) y en la publicación de cada artículo podrá encontrar la información básica y necesaria para cada módulo.



Finalmente se presenta un código de prueba para el testeo del funcionamiento del módulo de inicio implementando arduino usando los estados con los leds:

## Código ejemplo módulo de inicio

```
//pines leds(led1 azul, led2 amarillo, led3 rojo)
int led1 = 12;
int led2 = 13;
int led3 = 11;
```

```
//variables módulo de inicio
int Inicio = 9;
int Stop = 10;
int Estado;
```

```
void setup() {
  pinMode(Inicio, INPUT);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}
```

```
void loop() {
  Estado = digitalRead(Inicio);
```

```
  if(Estado == HIGH){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
  }
  else
  {
    digitalWrite(led2, HIGH);
    digitalWrite(led1, LOW);
  }
}
```