REPORTE: ARDUINO HARDWARE/SOFTWARE

Alumno: Ramiro PerezRamiro Pe

Profesor: Pedro Willging

FACULTAD DE CIENCIAS EXACTAS Y NATURALES
Universidad Nacional de La Pampa

Contenido

¿Porque trabajar y enseñar Robótica?

La enseñanza de la programación les permite a los alumnos formar un pensamiento mucho más analítico, estructurar mejor la toma de decisiones, darse cuenta que una vez que se encuentra con un problema o una necesidad, tiene que dar una serie de pasos ordenados hasta llegar a la solución que efectivamente sea acorde a dicho problema. Por eso mismo lo que se pretende es enseñar a programar desde la secundaria, así los alumnos entrenan desde temprana edad estos conceptos.

Adentrándonos más a su relación con la robótica, se puede decir que los alumnos se verán más motivados a la hora de trabajar, viendo resultados desde el momento de prender un led, siendo ellos los creadores de tecnologías.

¿Porque trabajar con Arduino?

Primero Arduino se trata de un microcontrolador, una placa, un pequeño sistema de procesamiento código abierto (open-source). Al ser de código abierto sus variaciones abundan, de hecho, podemos encontrar tantas configuraciones como desarrolladores dispuestos a hacer cambios en los esquemas puedan existir.

Arduino le ofrece muchas ventajas tanto a profesores como estudiantes en el mundo del aprendizaje, entre ellas podemos destacar:

- **Económico:** las placas Arduino son baratas comparadas con otras plataformas. Su consto está rondando entre los 170$ hasta los 250 o 300$.
- **Multiplataforma:** los softwares para trabajar con Arduino corren en los sistemas operativos más utilizados actualmente, tales como, Windows, Macintosh OSX y GNU/Linux.
- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Con respecto a hacer foco de facilidad para chicos que se encuentran en la secundaria, se han creado otros entornos mucho más amigables como es el caso de MBlock, Visualino, Arduino Create , etc.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

Hardware

En el mundo de Arduino nos podemos encontrar con distintos tipos de placas:

- Arduino UNO
- Arduino Leonardo
- Arduino Due
- Arduino Yún
- Arduino Robot
- Arduino Esplora
- Arduino Mega ADK
- Arduino Ethernet
- Arduino Mega 2560
- Arduino Mini
- Arduino Nano
- Arduino Pro Mini
- Arduino Pro
- Arduino Micro
- Arduino Fio
- LilyPad Arduino USB
- LilyPad Arduino Simple
- LilyPad Arduino SimpleSnap
- LilyPad Arduino

Para más detalle con respecto a las características de cada placa, les dejare un link donde muestra las especificaciones que contienen cada una:

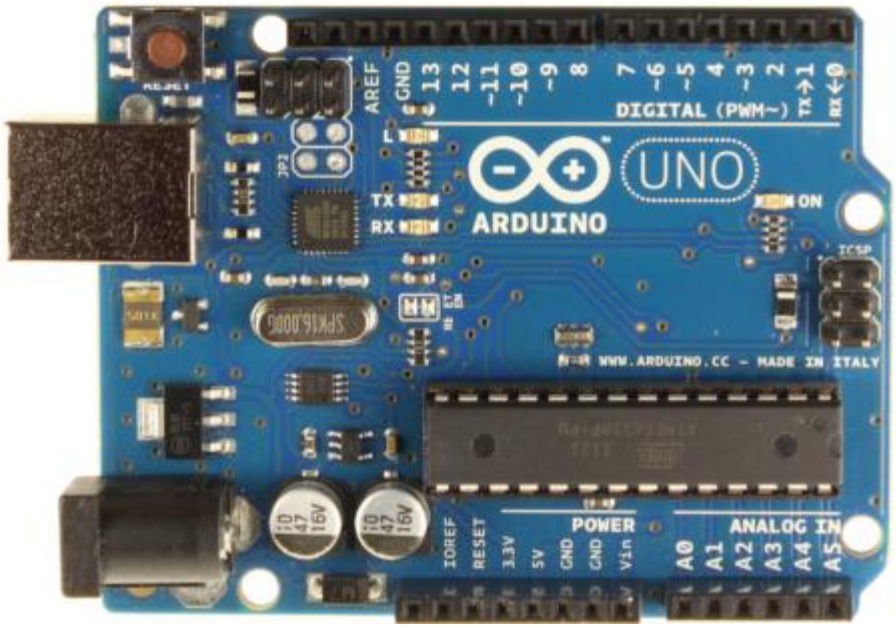
Tipos de placas.

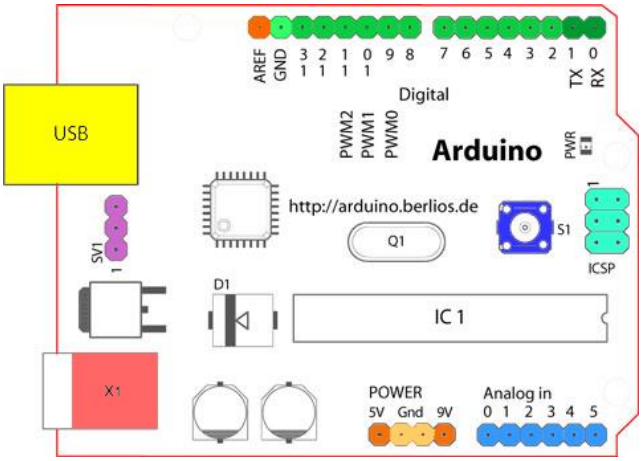
Entre las distintas versiones de arduinos creadas por diferentes fabricantes, escogimos para trabajar la variante Arduino Uno, ya que un docente nos informó de haberlas usado y nos las recomendó. Hablo de que hasta el momento no ha tenido fallos y recalco su bajo costo.

Paso siguiente nos decidimos comprar y testear las placas Arduino Uno R3. Una con chip ATMega 328 y la otra con el chip CH340.

Arduino Uno R3 ATMega 328:

Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteado.





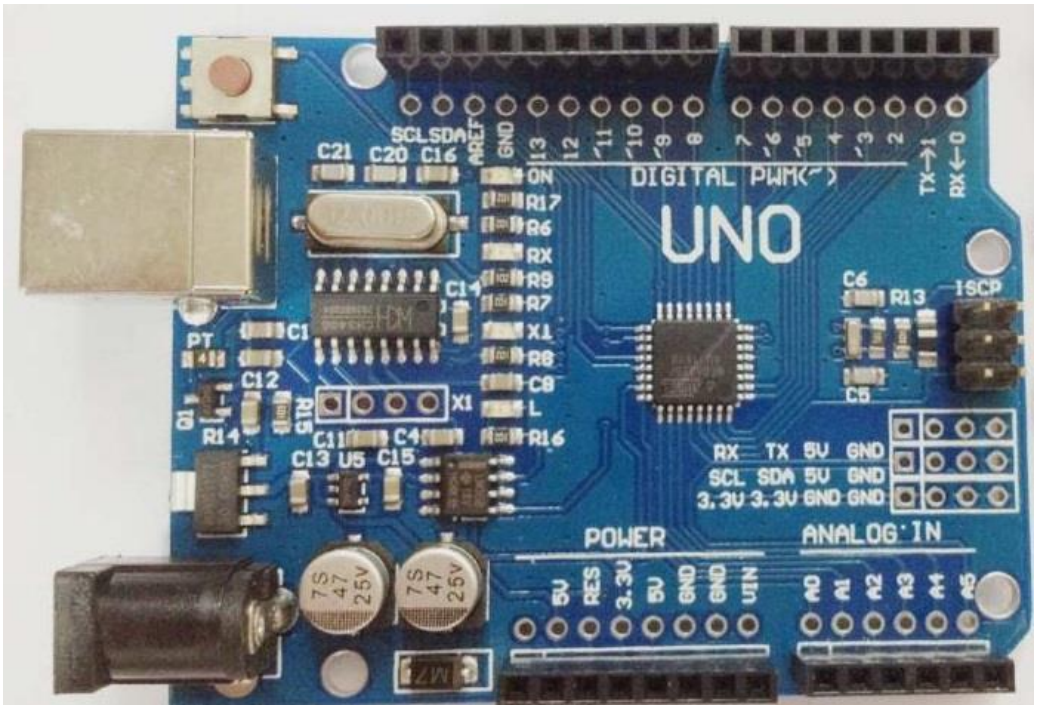
- Terminal de referencia analógica (naranja)
- Tierra digital (verde claro)
- Terminales digitales 2-13 (verde)
- Terminales digitales 0-1/ E/S serie - TX/RX (verde oscuro) - Estos pines no se pueden utilizar como e/s digitales
- Botón de reinicio - S1 (azul oscuro)
- Programador serie en circuito o "ICSP" (azul celeste).
- Terminales de entrada analógica 0-5 (azul claro)
- Terminales de alimentación y tierra (alimentación: naranja, tierras: naranja claro)
- Entrada de alimentación externa (rosa)
- Selector de alimentación externa o por USB (púrpura).
- USB (utilizado para subir programas a la placa y para comunicaciones serie entre la placa y el ordenador; puede utilizarse como alimentación de la placa) (amarillo)

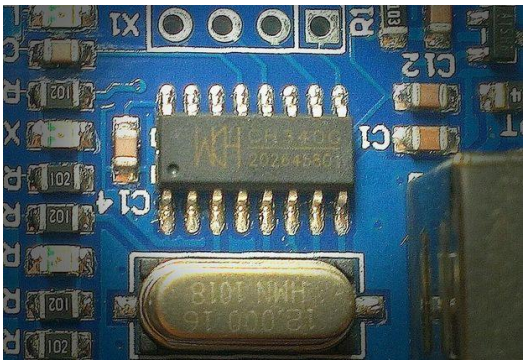
Hicimos barias pruebas con esta placa, entre ellas prender un led, instalar sensores ultrasónicos, de gas y llama. En todos los casos no presentó ninguna falla. Muy recomendable.

Arduino Uno R3 con Chip CH340G:

El CH340G es un nuevo conversor USB / TTL, que sustituye al ATMEGA16U2 y al FT232RL. A toda vista, es mucho más barato porque la aparición de estos modelos ha hecho descender drásticamente los precios de las placas Arduino clónicas.

El resto de especificaciones y características siguen siendo exactamente las mismas. Estas nuevas placas clónicas no solo son compatibles si no que son idénticas a las Arduino “originales”.



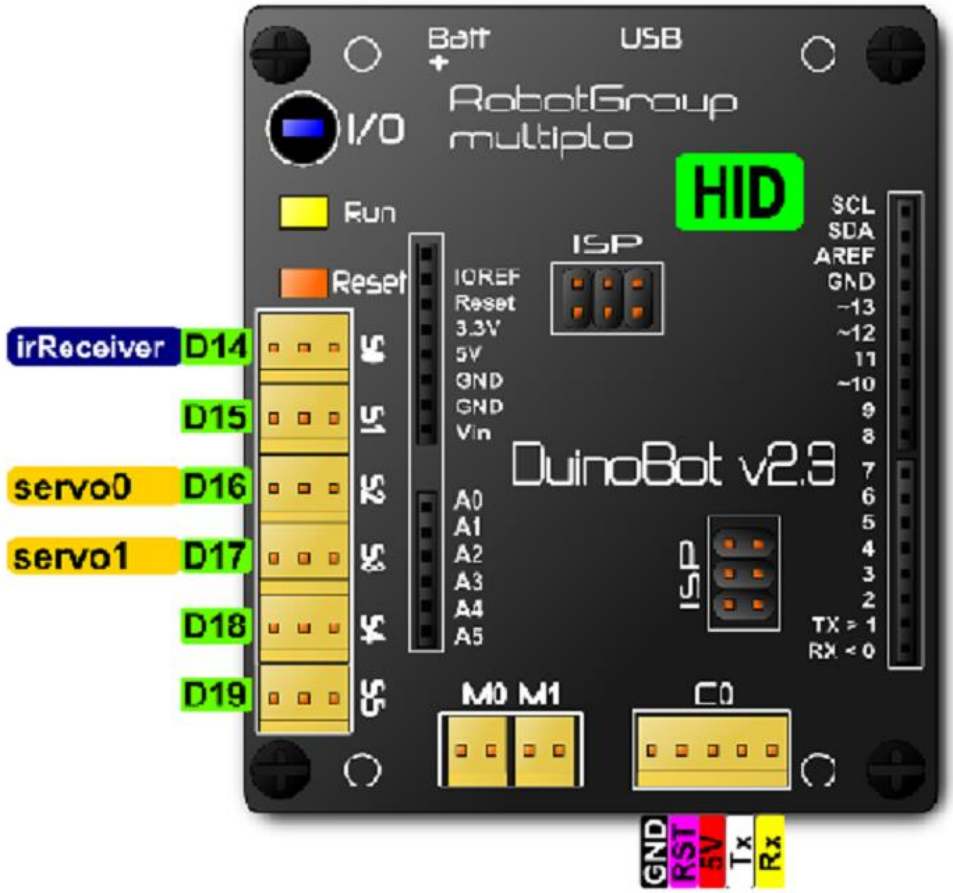


Al igual que la placa vista anteriormente, hicimos un testeo prendiendo un led, instalar sensores ultrasónicos, de gas y llama. Su funcionamiento fue excelente. Se puede decir que es muy recomendable ya que función igual que la original y además es de menor costo económico.

DuinoBot v2.3:

Fue utilizada durante la cursada. La misma venia junto a un pack de robotgroup y accesorios para la creación de distintos robots. Se probó con distintos sensores, entre ellos ultrasónicos y de llama. Con respecto al funcionamiento, optamos por el software Miniblocs que era el recomendando y utilizado en una guía que venía con en el kit.

La placa es la siguiente.



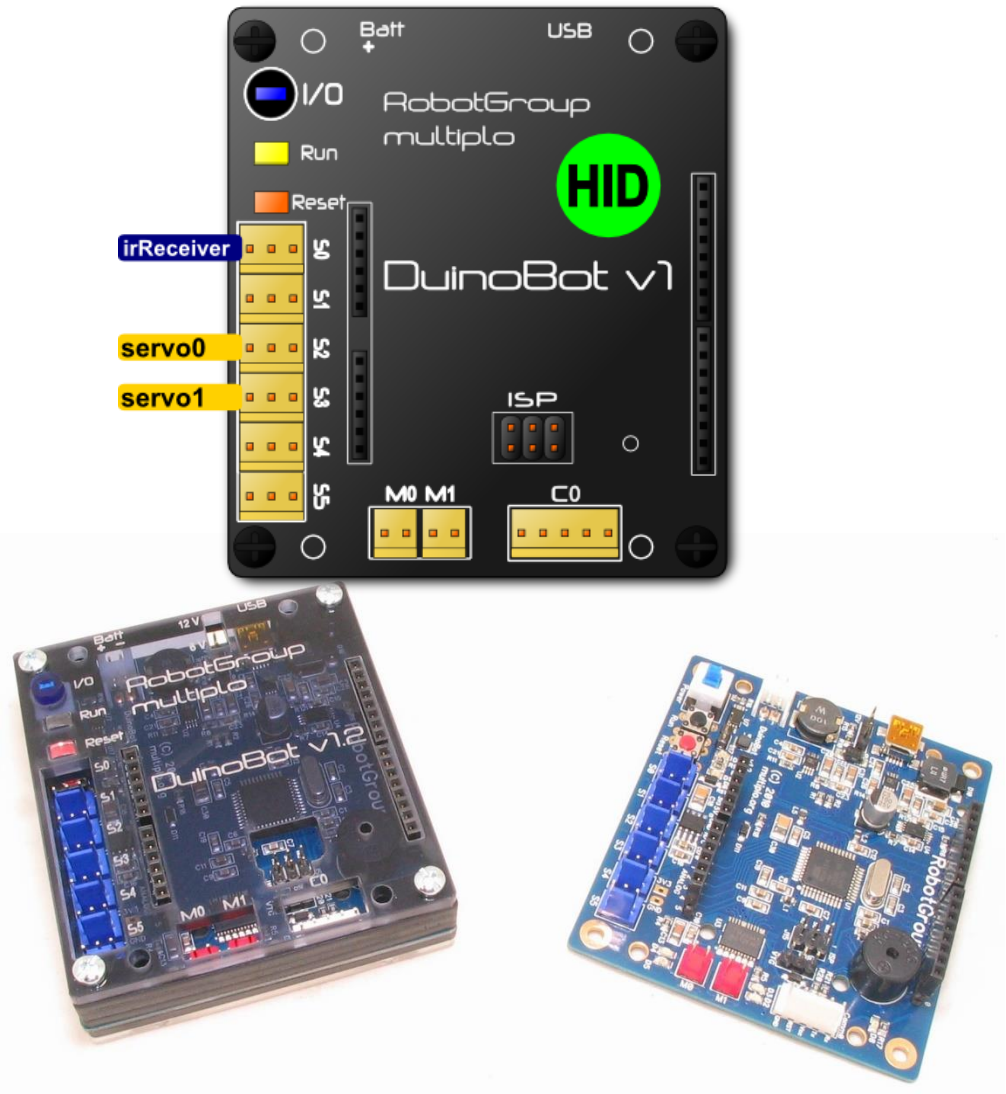
El agregado que contiene comparando con las placas Arduino uno R3 es:

- Botón Run: corre el programa que se encuentra instalado en la placa.
- Botón Reset: resetea (no borra) o reinicia el programa que se encuentra en la placa.
- Los enchufes D14 a D19 que son los servomotores que se pueden instalar directamente a la placa.
- Los enchufes M0 y M1.
- Enchufes ISP.
- Enchufes C0 que tienen un pin GND,5v, Tx, Rx y RST adicionales.

Con respecto al kit (placa + accesorios) se puede decir que es muy completo. El agregado que contiene la placa brinda mucha comodidad. La fabricación de un robot se hace sencilla gracias a la guía que viene con dicho kit. La complejidad puede venir acompañada con el software predefinido ya que no es muy intuitivito. No probamos la placa con otros softwares. ¡Se puede recomendar si quieres iniciar en el mundo de la robótica armando tu primer robot!

DuinoBot v1.2:

Es una placa que viene con otro kit de la empresa Robotgroup. Viene tanto la palca como los accesorios para la creación de distintos robots. No encontramos diferencias a comparación a la otra placa DuinoBot v2.3 desarrollada por el mismo fabricante. El software utilizado para su funcionamiento también es Miniblocs.



El agregado que contiene comparando con las placas Arduino uno R3 es:

- Botón Run: corre el programa que se encuentra instalado en la placa.
- Botón Reset: resetea (no borra) o reinicia el programa que se encuentra en la placa.
- Los enchufes D14 a D19 que son los servomotores que se pueden instalar directamente a la placa.
- Los enchufes M0 y M1.
- Enchufes ISP.
- Enchufes C0 que tienen un pin GND,5v, Tx, Rx y RST adicionales.

Con respecto al kit (placa + accesorios) se puede decir que es muy completo. El agregado que contiene la placa hace a mucha comodidad que las Arduino R3 que vimos con anterioridad. ¡Se puede recomendar si quieres iniciar en el mundo de la robótica armando tu primer robot!

Sensores utilizados:

Modulo Detector De Fuego Sensor De Llama Arduino:

El sensor de la llama, utilizado para detectar fuego. Detecta longitudes de onda de luz de 760 nm a 1100 nm. Se los suele utilizar como los ojos de robots para encontrar la fuente de fuego. El

sensor tiene un ángulo de captación de 60º y es especialmente sensible a la longitud de onda del fuego.

Características

- Receptor IR de alta sensibilidad.
- Sensa longitudes de onda entre 760-1100nm.
- Led indicador de encendido.
- Led indicador de detección.
- Salida digital de nivel superado (DO).
- Salida analógica de nivel sensado (AO).
- Potenciómetro de ajuste de nivel de sensado.
- 60 grados de ángulo de captación.
- Se utilizó en el testeo de las placas y los problemas que hubo fue que dependiendo del fabricante era la respuesta digital que enviaba. (Detectar fuego para un sensor envía el pulso digital 1 y para otros sensores el 0).

Sensor Ultrasónico Hc-sr04 Arduino:

El sensor ultrasónico HC-SR04 ofrece un rango de medición de 2cm a los 500cm, con una precisión de aproximadamente 3mm. Este módulo ultrasónico funciona de manera similar que sus pares, con la única diferencia que lleva por separado el pin de entrada y el de salida.

Características

- Voltaje de alimentación: 5V DC.
- Corriente en reposo: <2mA.
- Angulo de cobertura: <15°.
- Rango de distancia: 2cm – 500 cm.
- Resolución: 0.3 cm.
- Frecuencia ultrasónica: 40k Hz.
- Se utilizó en el testeo de las placas y no se encontraron fallas.

Modulo Detector Sensor Gas Mq2 Humo Monóxido

El sensor de gas MQ2 se puede utilizar en sistemas de protección contra incendios para detectar hidrógeno, isobutano, gas licuado de petróleo, metano, monóxido de carbono, el alcohol, el humo, propano y otros gases nocivos.

Posee una salida analógica que indica la concentración de gas en el ambiente (Cuanto más alto el nivel de salida, mayor la concentración de gas)

Y también posee una salida digital que baja a 0 cuando la concentración de gas supera el nivel prefijado con el preset.

Características

- Alimentación: 5V.
- Consumo: 122mA.

- Salida Digital y Analógica.
- Led de encendido.
- Led de accionamiento salida digital.
- Se utilizó en el testeo de las placas y no se encontraron fallas.

Software

Dado que el Arduino es como una pequeña computadora, que ejecuta una serie de códigos que previamente le hemos introducido, necesitaremos un programa para poder crear e insertar estos códigos a la dicha placa. Entre los tantos entornos de programación para Arduino, hemos optado por utilizar los siguientes:

IDE de Arduino:

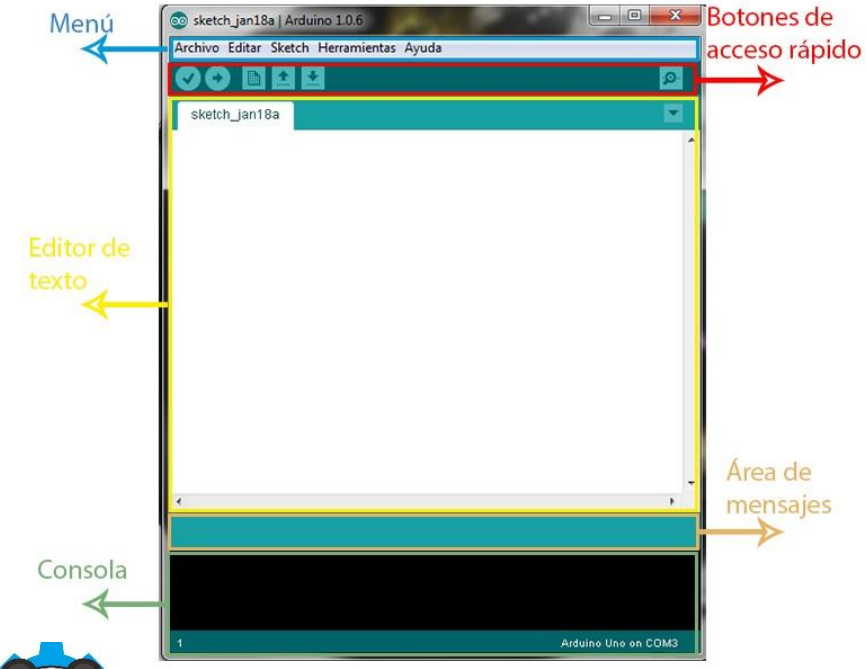
Este programa es llamado IDE debido a que cada sigla significa "Integrated Development Envaronen" ("Entorno de Desarrollo Integrado"). Es un entorno muy sencillo de usar y en él escribiremos el programa que queramos que el Arduino ejecute. Una vez escrito, lo cargaremos a través del USB y Arduino comenzará a trabajar de forma autónoma.

Ventajas.

- Para personas experimentadas en trabajar con distintos lenguajes de programación se les hará sencillo trabajar con el IDE.
- No falla ya que trabaja con el código puro de Arduino.
- Interfaz sencilla y amigable.
- Multiplataforma Mac OS, Windows y Linux.
- Es Open source.

Desventajas.

- Para alguien que inicia en Arduino y además nunca programo en otro lenguaje, se vería con muchas dificultades al inicio.
- No es el más recomendado para enseñar en la secundaria.

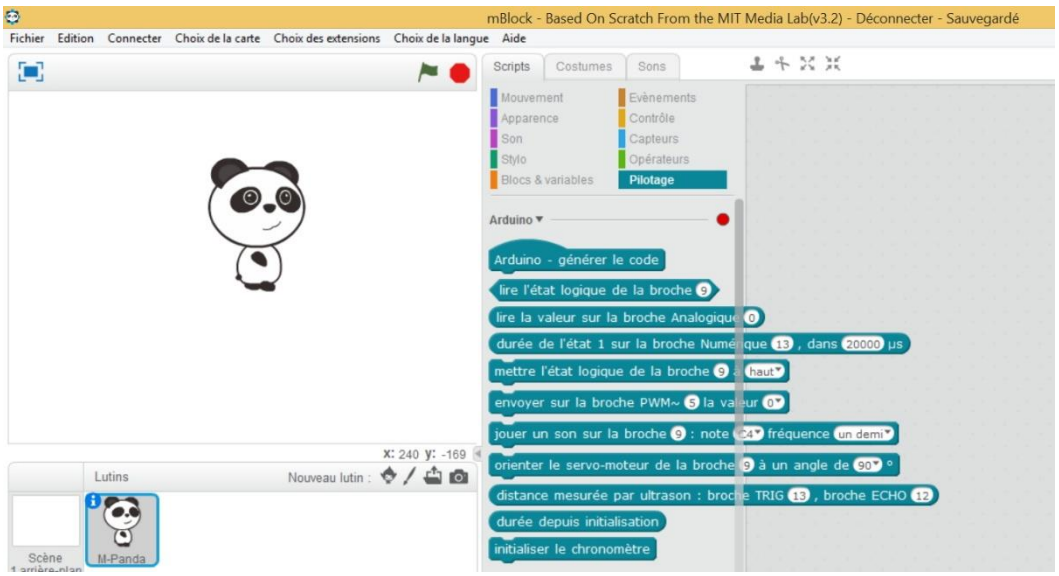


mBlock:

Es un entorno gráfico de programación basado en el editor Scratch, fue creado para la enseñanza de Arduino/Robótica, haciendo foco en el aprendizaje en la escuela Secundaria, haciendo la enseñanza mucho más sencilla y amigable para los alumnos.

En si, la interfaz de mBlock e s muy intuitiva a partir de su programación en bloques, permitiendo darle instrucciones a Arduino de una forma más didáctica.

Además, en el lado derecho del programa, tenemos una ventana, en la cual, nos brinda el código en Arduino de los bloques que llevamos apilados.



Ventajas:

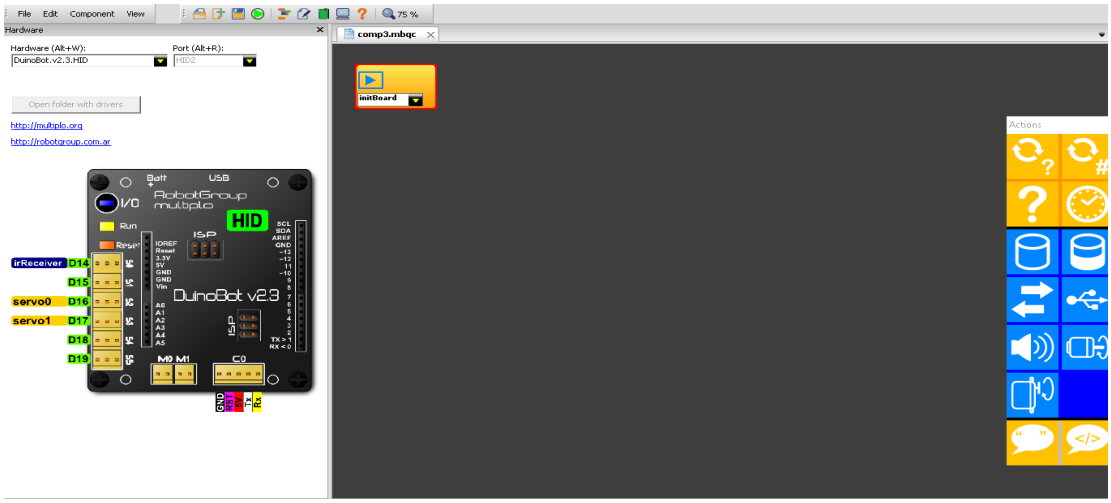
- Muy sencillo de usar, incluso para gente que nunca haya trabajado con ningún lenguaje de programación.
- Intuitivo.
- Interfaz amigable.
- Nos brinda el código puro de Arduino por si queremos pasar al siguiente nivel del aprendizaje o simplemente por curiosidad de ver que estamos haciendo.

Desventajas

- Debes de cuando sabe tener problemas en cargar el código a la placa.

MiniBloq:

Entorno grafico desarrollado en argentina y exclusivo para las placas creadas por el fabricante Robotgroup. A mi punto de vista, es para personas experimentadas con Arduino ya que su interfaz es muy complicada y poco intuitiva.



Ventajas

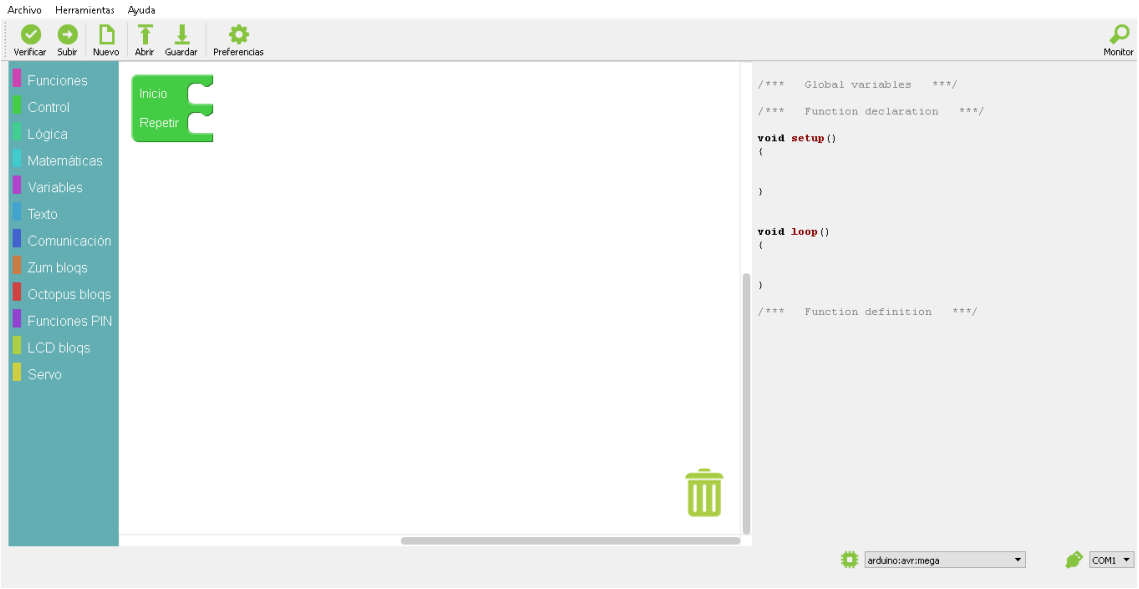
- Muchas funciones.
- Muy preciso
- No hemos encontrado falla durante su uso.

Desventajas

- Interfaz poco amigable.
- Difícil de entenderlo si no has trabajado con otros entornos gráficos.
- Funciona solamente con placas fabricadas por Robotgroup.

Visualino:

Es un entorno similar a Scratch: permite crear programas para Arduino como un puzle. Pero permite programar directamente la placa de Arduino. Además, los bloques generan el código de C/C++ en tiempo real en una ventana. El entorno es similar al del IDE de Arduino, con las mismas opciones principales: Verificar, Subir, Guardar, Cargar y Monitor.



Ventajas

- Muy sencillo de usar, incluso para gente que nunca haya trabajado con ningún lenguaje de programación.
- Intuitivo.
- Interfaz amigable.
- Nos brinda el código puro de Arduino.
- No hemos encontrado fallos en el momento de usarlo.
- De todos los vistos, este es el más recomendable.

Desventajas

- Por el momento no he encontrado ninguna desventaja.

Inclusión de la robótica educativa en la currícula.

Tecnología de los Sistemas Informáticos

EJE: Tecnología aplicada al Hardware.

¿De qué trata el eje?

El desarrollo de este eje permite conocer los distintos dispositivos electrónicos que conforman los sistemas tecnológicos de procesamiento de información y de la comunicación. Este eje se propone analizar cómo se interrelacionan los mismos, para su correcto funcionamiento y comparar estos sistemas informáticos con sistemas tecnológicos que tengan la misma función.

Asimismo, admite la posibilidad de clasificar los sistemas tecnológicos y sus componentes según función y uso.
El tratamiento de los saberes que se proponen en este eje es fundamental para la orientación, proporciona la oportunidad de conocer y analizar la diversidad tecnológica en la que se encuentran inmersos nuestros alumnos.

¿Porque la elegí?

Elegí esta currícula y este eje porque me pareció el más cómodo para implementar la robótica en clase. Como bien dice la descripción del eje, esta trata de que los chicos identifiquen los distintos dispositivos electrónicos que conforman los sistemas informáticos de procesamiento de la información y la comunicación. La idea mi es ir más allá que solo enseñarles que es hardware y software, y darles ejemplos.
Mi objetivo es que ellos se sientan participes que están creando tecnología, tanto sea un programa que prenda un led, como también, instalar un led para que este mismo tenga las condiciones necesarias para prenderse.

¿Pero cómo podemos llegar a esto?

He notado en las clases presenciales que los docentes siempre buscan la forma más didáctica para que los alumnos comprendan como está conformado un sistema informático. Partiendo de que estos se constituyen de una parte física (hardware) y una parte intangible (software). Noto que muchos alumnos siempre llegan a lo mismo, monitor, teclado como hardware y Windows como software. Y muchas veces los docentes ven como "trabajo cumplido" que nombren un par de cada uno y los identifiquen por si están en el exterior o interior, si extraen o dan información, etc. Y acá es donde yo quiero dar una vuelta de tuerca.
Una vez que vea que los chicos comprendan mínimamente los conceptos (hardware y software) o los logren identificar, va a ser el lugar donde voy a iniciar con robótica.
Primero hare que se pongan en grupo de dos personas en una computadora (el número de integrantes aumentaría si son escasas las Computadoras) y busquen en el escritorio el programa mBlock (ya instalado el programa).
¿Por qué el mBlock y no otro? A pesar de las pequeñas fallas a pasar el código a las placas, a mi parecer es el mejor para aprender o introducirse en el mundo de la robótica, tanto sea por su interfaz amigable y lo intuitivo que es.
Una vez abierto el programa les preguntare ¿Qué es lo que abrieron, un hardware o un software? La respuesta es obvia, pero lo que se espera es ver si comprendieron los conceptos anteriores.
Hare que jueguen todo el tiempo que resta de la clase con el panda de mBlock para que se familiaricen con lo que es la programación en bloques. Mi idea en esto es ir dándole mini retos, como mover panda a la derecha, hacer que se mueva 10 pasos, que repita, etc. Obviamente que si veo que el tiempo no alcanza se dará otra clase para que sigan practiquen. Mi idea es que no dure más de dos clases la introducción a mBlock.
Después que vea que los chicos están bastante cancheros, hare lo siguiente:

Mostrándoles que para mi caso lo mejor son las placas Arduino, y preguntar si tienen idea de que es esto. Las respuestas que intuyo que voy a recibir son:

- Una placa.
- Una placa madre en miniatura.
- Una mini computadora.
- Etc.

Mi respuesta será, que Si, es una placa, pero no una computadora en su totalidad, lo que hace es recibir información de una computadora y ejecutar las instrucciones recibidas. Les diré que estas instrucciones pueden ser creadas y enviadas des el software que estuvieron usando hasta el momento. Y ahí es donde les mostrare la sección de robótica que contiene mBlock.

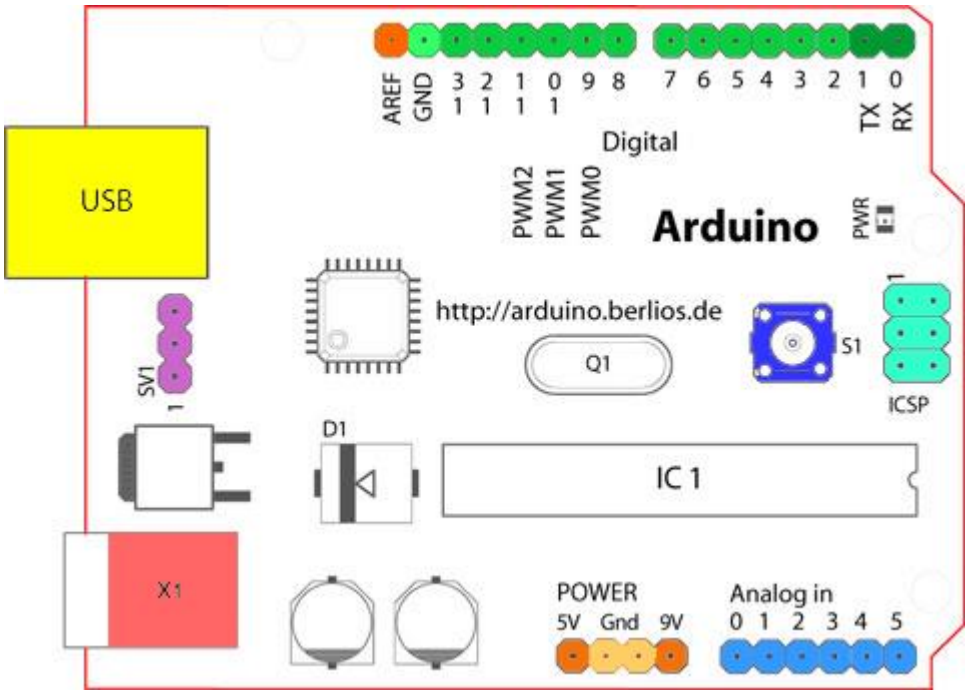
Les explicare el tipo de información con la que trabaja esta placa y mBlock (digital y analógica), es decir, que recibe un Sí o un No que sería la información Digital o Binaria, y que recibe un rango de valores, que sería la información Analógica.
Explicare donde están los diferentes pines y partes de la placa Arduino. Luego, los ayudare a crear su primer "hola mundo" que sería prender un led. Después decirles que prendan dos, pero va a haber un problema, que los pines no son muy cómodos para usar dos leds y por lo tanto introduciré la herramienta protoboard y explicare como se usa.
Mi idea es que lleguen a trabajar con sensores infrarrojos de llama, servomotores, sensores ultrasónicos y tal vez más.
Ya terminando lo que sería mi planificación de clases, haría el cierre del tema. Preguntarles a los chicos porque hicimos esto, ven alguna relación de lo que vimos antes con lo que vimos ahora (antes software y hardware, ahora robótica), les preguntare que, si los programas que crearon son software, si la placa y los sensores seria hardware.
Así sería como implementaría yo la robótica en la secundaria con respecto a este eje. Siento como los chicos se sentirían mucho más motivados en trabajar y viéndose ellos como creadores de software y hardware, comprendiendo como uno lo necesita al otro.

Actividades.

1)Introducción a mBlock:

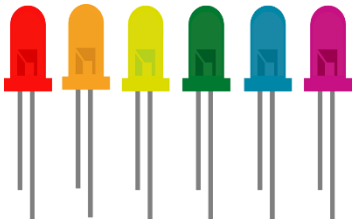
A. Hacer que el panda camine 10 pasos.
B. Hacer que el camine 3 veces 10 pasos.
C. Hacer que el panda gire 90° y luego camine 3 veces 10 pasos.
D. Hacer que el panda dibuje un cuadrado.
E. Hacer que el panda dibuje un rectángulo y diga "¡Gane!" al completarlo.

2)Primer paso. El primer paso en todo lenguaje de programación es crear nuestro primer "hola mundo" o primer programa. Lo que haremos será programar con los bloques, el código para que nuestro led se prenda. Para ello tenemos que tener en cuenta que un led necesita dos cosas: uno es el pulso digital que será enviado por los pines digitales que tienen la placa, y dos seria GND o tierra, para que en caso de que haya un sobre voltaje este sea enviado fuera de la placa y no se dañe.

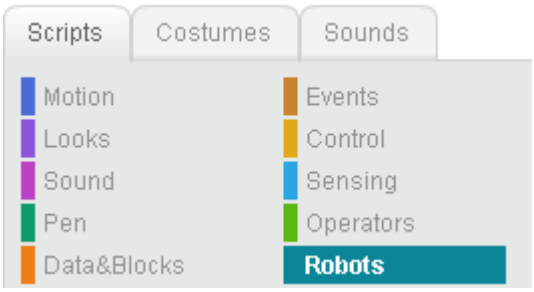


Los pines digitales son los de color Verde.
Los GND o tierra son los de color Verde claro.

El desafío es, conecten un led al pin digital 13 y programen en mBlock para que este se encienda. Tengan en cuenta que el led tiene dos patas diferentes, la más larga va al pin digital y la más corta al GND.

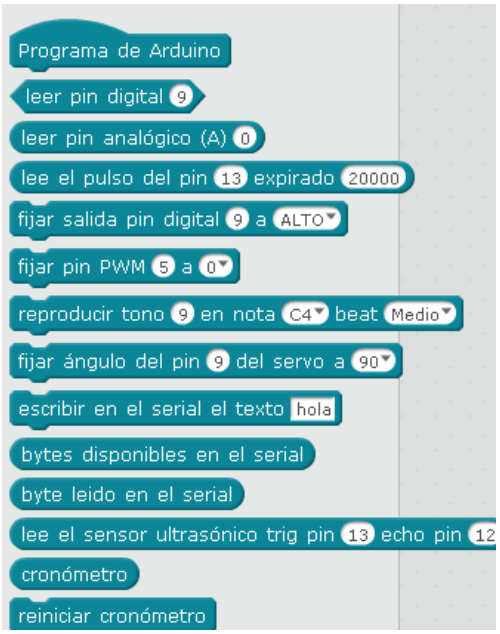


Con respecto a la parte de mBlock. Utilizaremos la solapa Robots



Luego arrastraremos el bloque Programa de Mbot a la parte derecha.

Ahora lo que tenemos que hacer es enviarle al led un pulso digital. ¿Cuál de los siguientes bloques piensas que es?



Si te quedo de esta forma lo hiciste ¡muy bien!

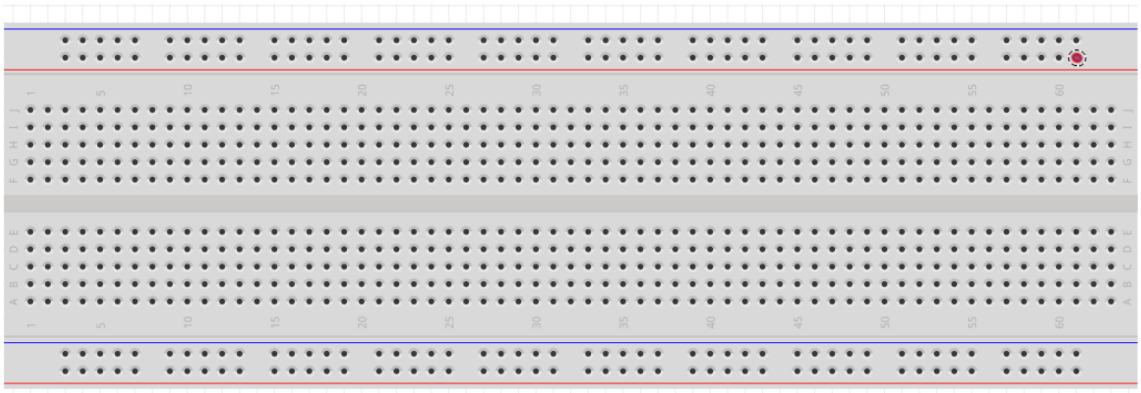
Pero nos faltaría algo, Arduino está constantemente recibiendo y enviando información. Por lo tanto, sería recomendable que utilicemos el bloque Por siempre.

Bien. Ahora queremos que el led se prenda y apague. ¿Qué bloques utilizarías y como los ubicarías para que lo hiciera?

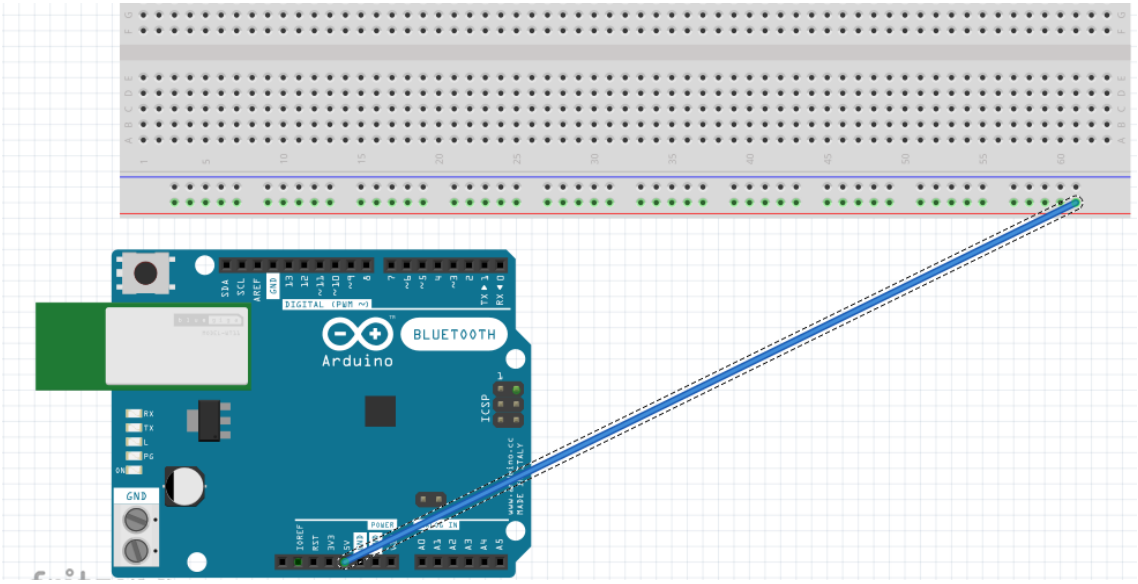
Si hiciste esto, estaría bien. El problema está que la placa recibiría un pulso digital que prende el led y al instante otro que es el que lo apaga. Sería tan rápido el envió de pulsos que no detectaríamos que lo apago, por lo tanto, siempre quedaría apagado o prendido.

La mejor opción es que hagamos que la placa espere un tiempo antes de enviar un pulso digital. Nos quedaría así.

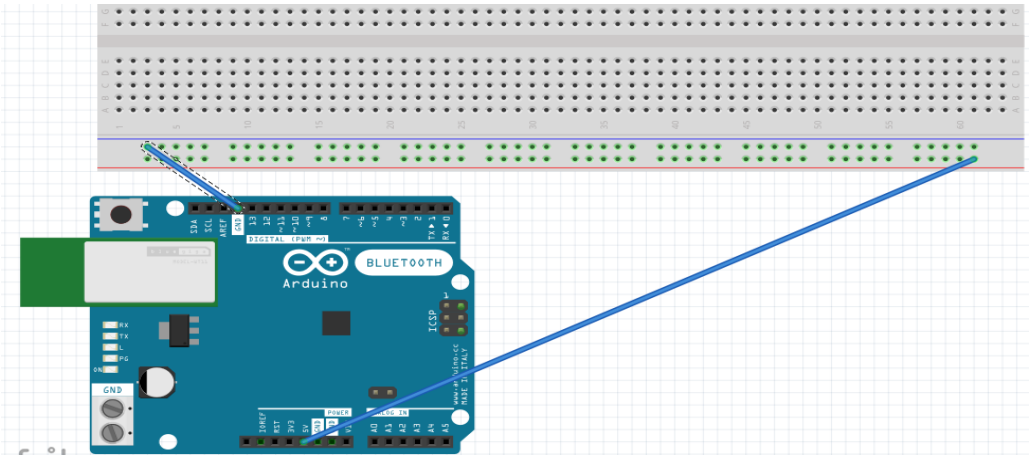
3) Introducir Protoboard. Ahora si queremos conectar más de un led estamos en un problema. La mejor opción es trabajar con una protoboard que nos permitirá conectar muchos dispositivos a nuestro Arduino. Pero ¿Cómo funciona? Bueno, primero tendremos nuestra protoboard:



Lo que haremos primero es conectar corriente, 3v o 5v con un cable desde la placa Arduino a unos de los primero orificios. Lo que hará es que toda la línea horizontal tenga corriente.



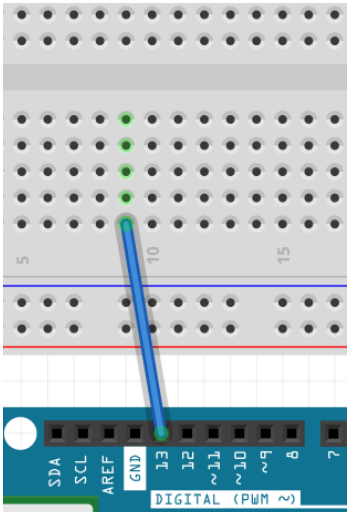
Luego conectamos Tierra o GND a la protoboard desde la placa, al igual que la corriente, la protoboard tendrá tierra en toda la liña horizontal.



Luego, si conectamos con un cable desde un pin a la protoboard, tendremos que hacerlo en lo agujeros interiores, estos trabajan diferent que los inferiores o superiores.

Si conectamos algo en los centrales, estaremos enviando información, pero los agujeros que se encuentran en la parte vertical.

¡De esta forma podemos conectar varios leds u otros dispositivos! Te invito a que intentes conectar 3 leds que se prendan y apaguen de forma simultánea.



4)Sensores. ¡Ahora fabriquemos una alarma!

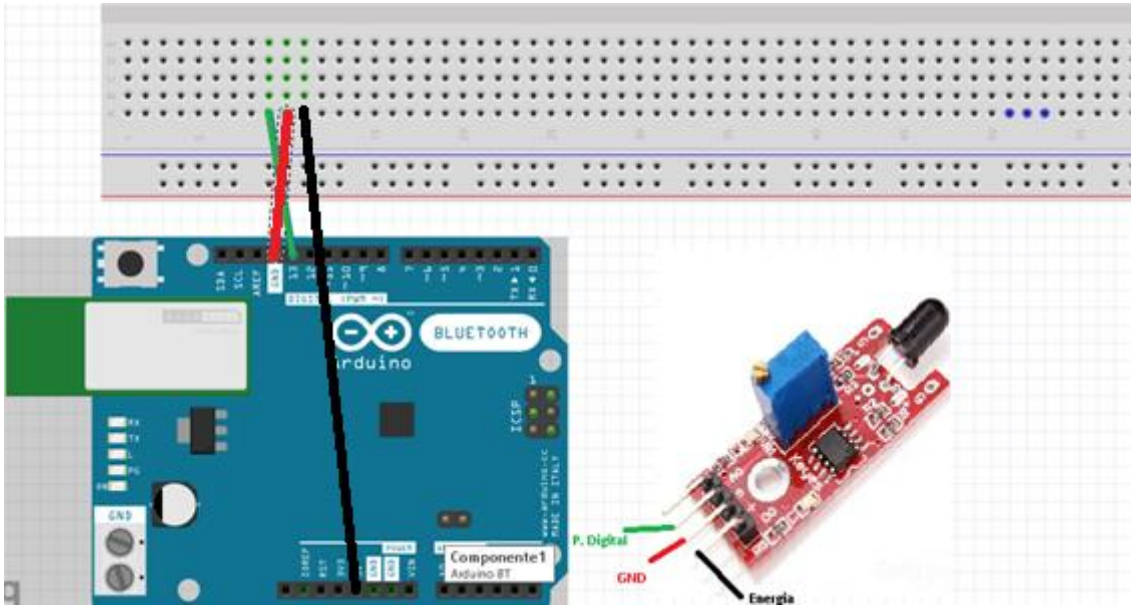
El sensor de llama, para que funcione necesita 3 cosas, energía, GND o tierra y pulso digita. Si observamos la parte de los cables del sensor podemos observar que tiene escrito. GND, VCC (energía), DO (pulso Digital) y AO (pulso Analógico).

Lo que haremos será conectar el sensor a la protoboard. Luego conectarle en el VCC o corriente desde la misma protoboard y el GND, y desde la palca Arduino conectar el pin digitar que deseemos a la parte que dice DO.

El código para que funcione sería el siguiente. Primero necesitaremos de un bloque que se encuentra en la solapa control. Lo que tiene el sensor de particular es que envía pulsos digitales. Por eso usaremos este bloque, en el caso de que el sensor no detecte llama, este enviara el pulso digita 0. Por lo tanto, si queremos que el led que fabricamos antes se apague mientras no detecte llama nuestro sensor, tendremos que codificarlo de la siguiente manera.

En caso de que queramos que se prenda si detecta llama, es decir, cuando el sensor envié a la placa el pulso digital 1, tendremos que codificarlo de la siguiente manera.

Nos tendría que quedar así conectado.



5) ¡Ultimo paso! Ahora que saben conectar un led y un sensor de llama. Los invito a que creen el código para que el led prenda la luz si el sensor de llama se activa y que el led quede apagado si el sensor de llama no detecta nada.

Así terminaría una de las actividades de introducción a Arduino.

Bibliografía:

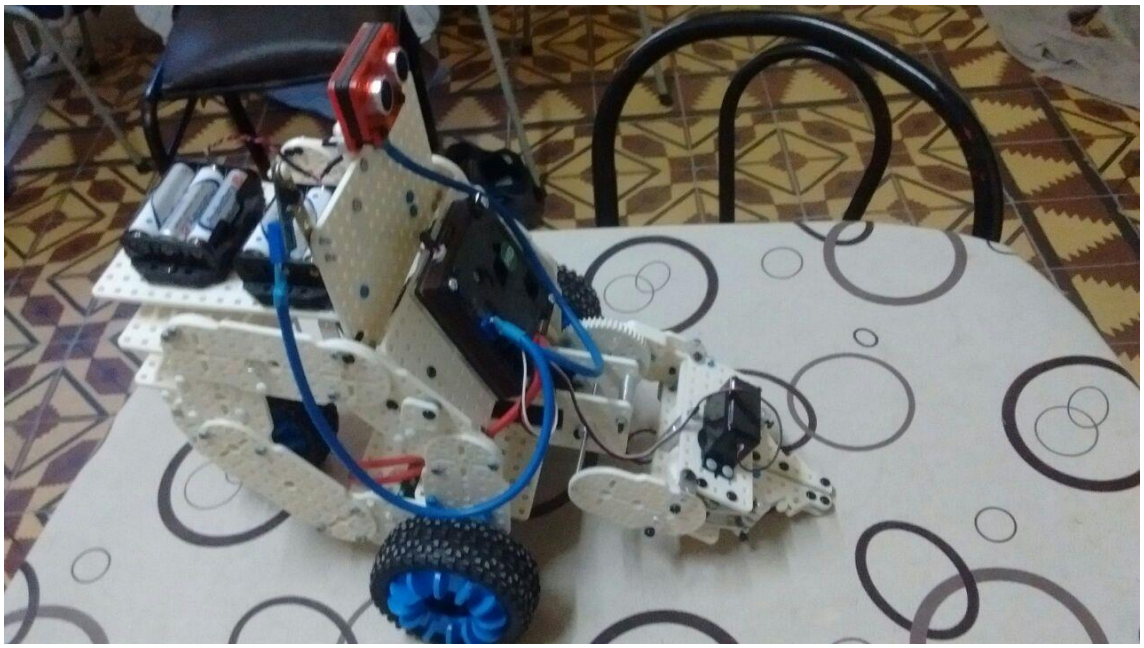
- http://arduino.cl/que-es-arduino/
- http://www.nubbeo.com.ar/
- http://www.prometec.net/ch340g/
- http://www.mblock.cc/
- http://www.visualino.net/index.es.html
- http://robotgroup.com.ar/es/
- https://www.arduino.cc/en/main/software
- http://hacedores.com/cuantos-tipos-diferentes-de-arduino-hay/

Fotos de nuestra experiencia con Arduino y los kits Robotgroup:

Repositorio_de_fotos.

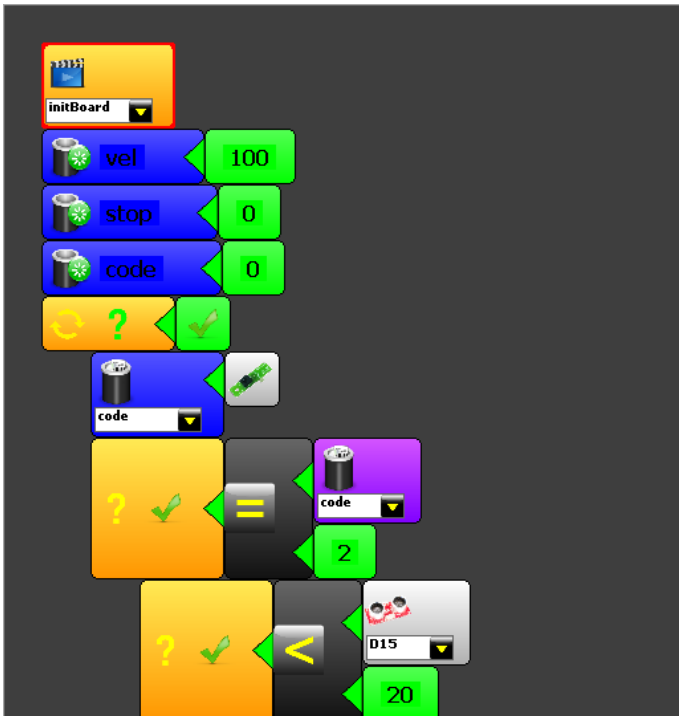
Como se dijo anteriormente, durante la cursada de Informática Educativa 2 estuvimos experimentando con el desarrollo del algoritmo y creación de robot N8 de Kit Avanzado de Tecnología y Robótica de la empresa Robotgroup.

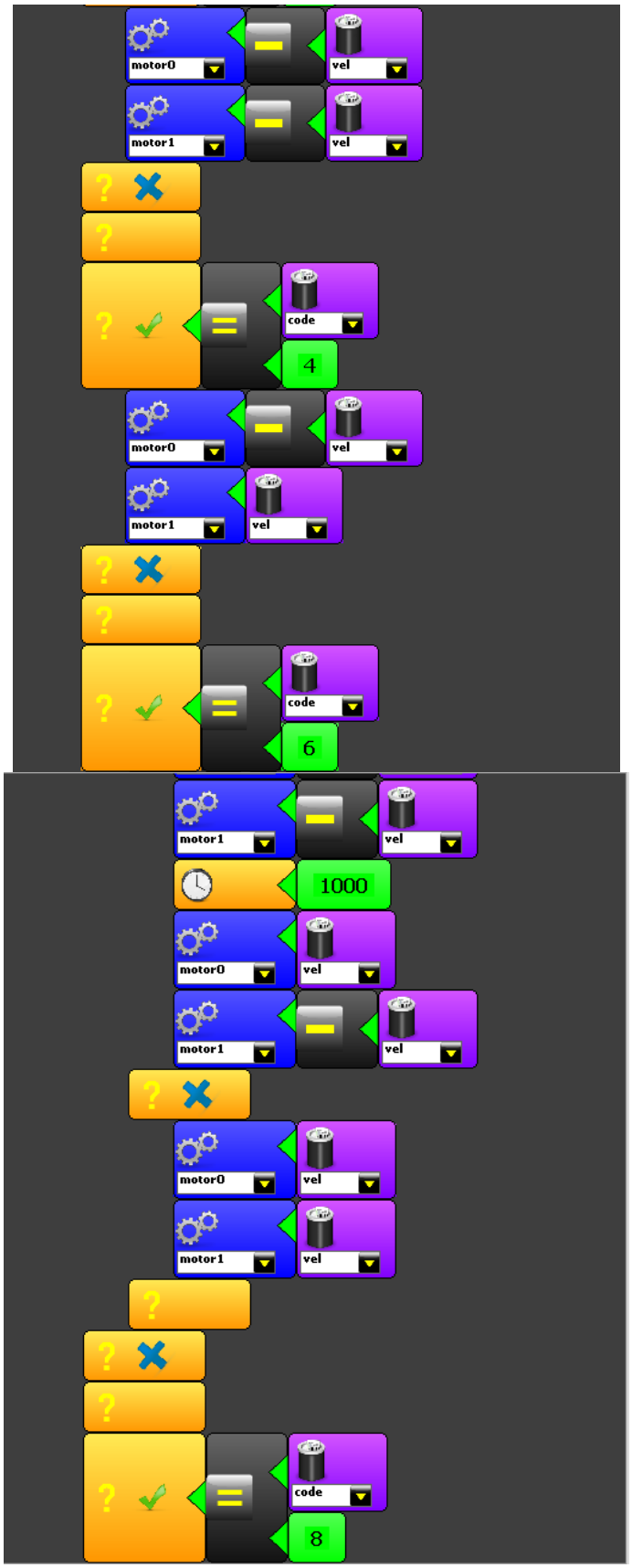
El robot terminado es el siguiente:

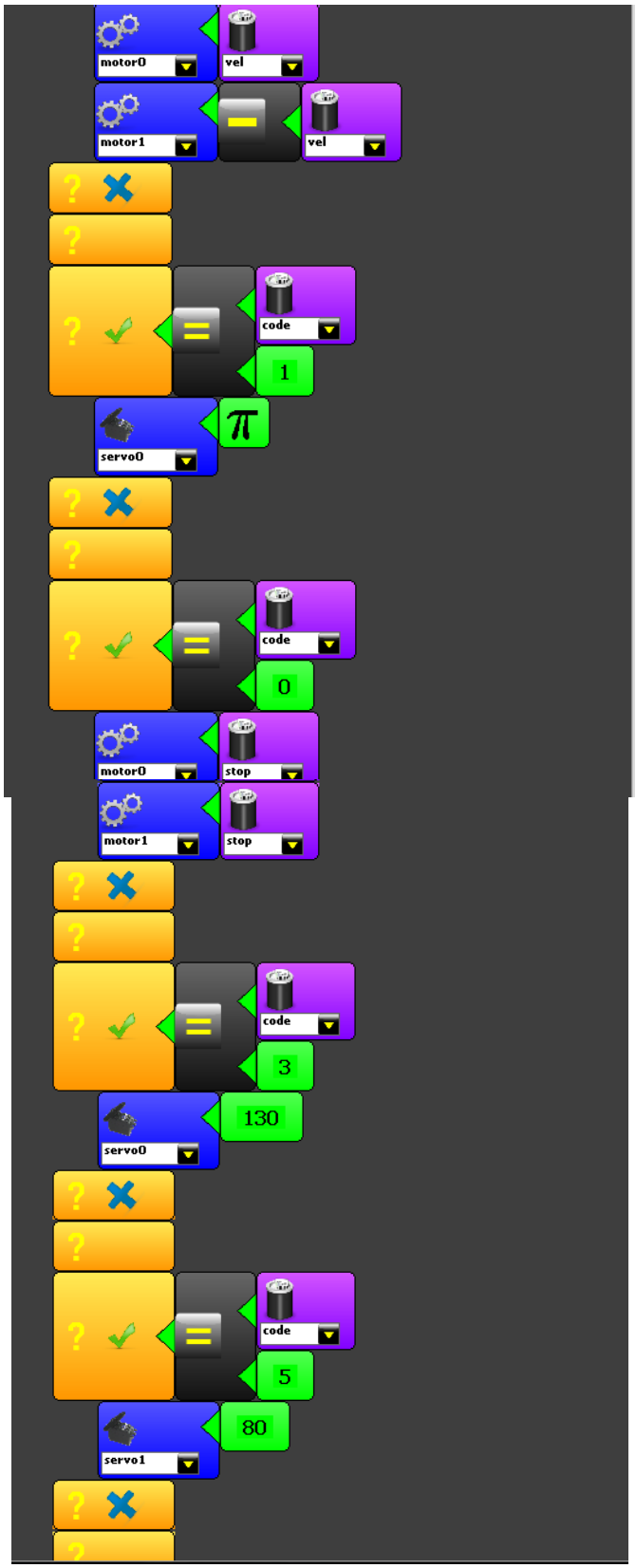


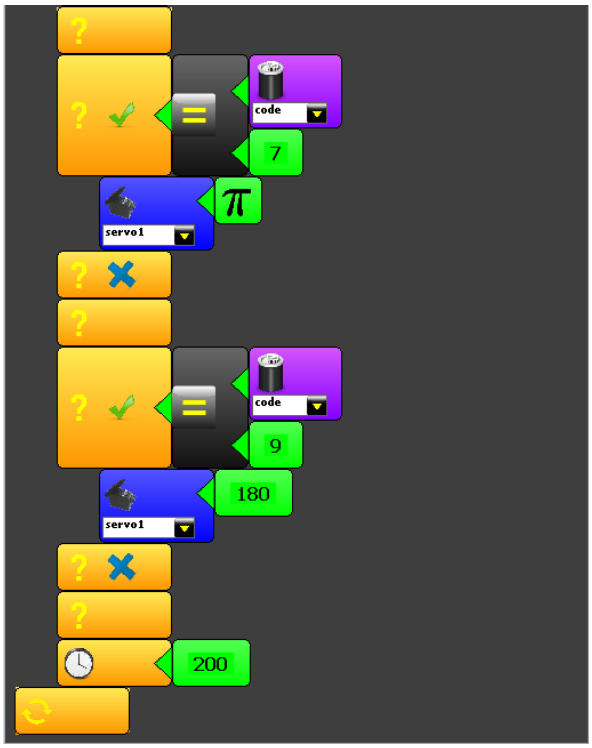
El mismo tiene 2 ruedas que se encuentran conectadas al aplaca DuinoBot v2.3, tiene una pinza con desplazamiento vertical, la pinza tiene programado para que se abra y cierre. Con respecto al movimiento de las ruedas y el de la pinza, estos son manipulados por control remoto y además se encuentra incluido un sensor ultrasónico en la parte superior con el fin de que el robot se detenga si esta por chocarse con algo.

El código en MiniBoq es:




motor0
vel
motor1
vel
code
4
motor0
vel
motor1
vel
code
6
motor1
vel
1000
motor0
vel
motor1
vel
motor0
vel
motor1
vel
code
8


motor0
vel
motor1
vel
code
1
servo0
π
code
0
motor0
stop
motor1
stop
code
3
servo0
130
code
5
servo1
80



Y el código en C o Arduino:

```
#include <mbq.h>
#include <PingIRReceiver.h>

void setup()
{
	initBoard();
	float vel = 100;
	float stop = 0;
	float code = 0;
	while(true)
	{
		code = irReceiver.getIRRemoteCode();
		if(((int)(code)==(int)(2)))
		{
			if((pingMeasureCM(D15)<20))
			{
```

```
                motor0.setPower(-(vel));
                motor1.setPower(-(vel));
                delay(1000);
                motor0.setPower(vel);
                motor1.setPower(-(vel));
        }
        else
        {
                motor0.setPower(vel);
                motor1.setPower(vel);
        }
}
else
{
}
if(((int)(code)==(int)(8)))
{
        motor0.setPower(-(vel));
        motor1.setPower(-(vel));
}
else
{
}
if(((int)(code)==(int)(4)))
{
        motor0.setPower(-(vel));
        motor1.setPower(vel);
}
else
{
}
if(((int)(code)==(int)(6)))
```

```
{
        motor0.setPower(vel);
        motor1.setPower(-(vel));
}
else
{
}
if(((int)(code)==(int)(1)))
{
        servo0.attachAndWrite(M_PI);
}
else
{
}
if(((int)(code)==(int)(0)))
{
        motor0.setPower(stop);
        motor1.setPower(stop);
}
else
{
}
if(((int)(code)==(int)(3)))
{
        servo0.attachAndWrite(130);
}
else
{
}
if(((int)(code)==(int)(5)))
{
        servo1.attachAndWrite(80);
```

```
                }
                else
                {
                }
                if(((int)(code)==(int)(7)))
                {
                        servo1.attachAndWrite(M_PI);
                }
                else
                {
                }
                if(((int)(code)==(int)(9)))
                {
                        servo1.attachAndWrite(180);
                }
                else
                {
                }
                delay(200);
        }
}

void loop()
{
}
```