

2017

Informática Educativa 2



[ACTIVIDAD INTEGRADORA]

Robótica Educativa - Informe

Alumno: Lobos Martín Mariano.

Profesor: Pedro Willging.

Asignatura: Informática Educativa 2



FACULTAD DE CIENCIAS
EXACTAS Y NATURALES

Universidad Nacional de La Pampa

INDÍCE

Introducción	1
Hardware.....	1
Placas de Robótica	1
 <i>Arduino Uno Mega328 R3</i>	1
Descripción General	1
Características	2
Experiencias y usos.....	2
Conclusión	3
 <i>Arduino Uno con CH340</i>	3
Descripción General	3
Experiencias, usos y Características	4
Conclusión	5
 <i>DuinoBot</i>	5
Descripción General	5
Características	5
Experiencias y Usos	6
Conclusiones.....	7
Sensores y otros módulos que utilizamos	7
 <i>Sensor de Llama KY-026</i>	7
Descripción General	7
Características	7
Experiencias y uso	8
Conclusiones.....	9
 <i>Sensor de Humo y Gases MQ2</i>	10
Descripción General	10
Características	10
Experiencias y uso	10
 <i>Speaker (Parlante)</i>	12
Descripción general	12
Experiencia y Usos.....	12
Conclusiones.....	12

LED de 5mm	13
Descripción General	13
Experiencia y Usos	13
Software	13
Entornos de Desarrollo	14
Mblock	14
Descripción General	14
Experiencias y Usos	15
Conclusiones.....	15
IDE Arduino	16
Descripción General	16
Experiencias y Usos	18
Conclusiones.....	18
Minibloq	18
Descripción General	18
Experiencias, Usos y Conclusión	18
Inclusión en la Curricula	19
Ejes de los Materiales Curriculares	19
Asignatura: Tecnología de los Sistemas Informáticos	20
EJE: Tecnología aplicada al Hardware.	20
Asignatura: Tecnología de la Información y la Comunicación	21
EJE: Tecnología que medían la gestión de la información y la presentación de los resultados.	21
Proyectos Realizados (En Mblock)	22
Alarma de Fuego	22
Bloques.....	22
Conexión.....	26
Alarma de Humo	27
Bloques.....	27
Conexión.....	31
Robot Group: Robot N8	31
Descripción	31
Código.....	33

Conclusión..... 39

ANEXO 1..... 39

Bibliografía 39

Introducción

A lo largo de la Cursada de la Materia Informática Educativa 2, de la carrera Profesorado Universitario en Computación, se abordan contenidos pertenecientes a la Robótica Educativa. En este espacio, experimentamos con diferentes IDE's, sensores, placas de Robótica y algunos Robots ya armados.

Por lo tanto, el objetivo de este trabajo, es realizar un informe dividido en 3 ejes (**Hardware, Software, Inclusión en la Curricula**), que resuma nuestra actividad en torno a la Robótica Educativa, y a su vez, dar algunas recomendaciones en base a la experiencia y dificultades que nos encontramos, para facilitar el inicio a cualquier persona, que quiera iniciarse en el Mundo de la Robótica Educativa.

Hardware

Comencemos con las Placas de Robótica con las cuales trabajamos:

Placas de Robótica

Arduino Uno Mega328 R3

Descripción General

“El Uno es una placa basada en el ATmega328. Tiene 20 pines digitales de entrada/salida (de las cuales 6 se puede utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector



Ilustración 1: Arduino UNO Mega 328 R3

de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para empezar a usar el micro controlador. Basta con conectarlo a una computadora con

un cable USB o una fuente de alimentación para empezar.”

Características

Las características, serán a continuación enlistadas en base a la siguiente ilustración de referencia

- 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"

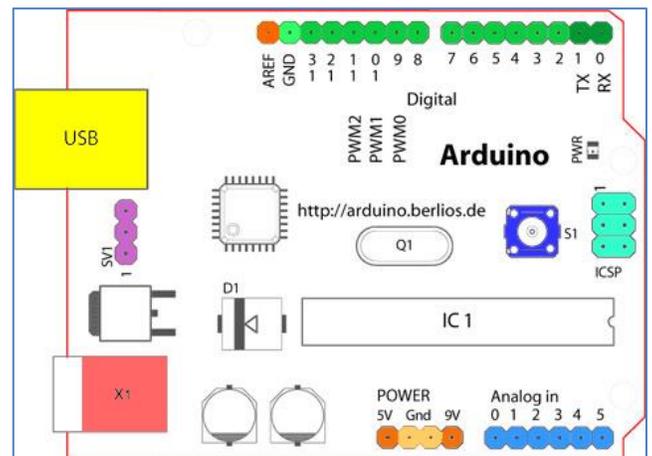


Ilustración 2: Esquema Arduino UNO R3

(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)

Experiencias y usos

Esta placa, básicamente es la réplica más fiel que se puede encontrar en el mercado de la placa Arduino Uno Original, respetando todas sus características originales tanto Hardware y Software. A simple vista incluso es muy difícil de diferenciarla.

Por lo tanto, ningún IDE o Lenguaje de Programación para las placas Arduino presenta problemas de reconocer estas placas.

Se le dio uso para:

- Hacer alarmas detectoras de Fuego.
- Hacer alarmas detectoras de Humo.
- Uso intenso con LED (prendido y apagado).

Y en ninguna de estas actividades presento problemas.

Conclusión

Con un precio al día de la Fecha (Mayo del año 2017) que varía de unos 230 ARS\$ a unos 260 ARS\$ (Placa + Cable), esta placa supero las expectativas, gracias a su gran parecido con una placa original. Sumamente recomendables.

Arduino Uno con CH340

Descripción General

“Esta tarjeta de desarrollo no es una versión diferente del Arduino UNO R3 mas bien es la evolución de ella. El bajo costo de esta tarjeta se debe a que contiene el chip adaptador USB – SERIAL CH340 en vez del 16U2 más costoso y por si fuera poco tiene un controlador certificado de Windows lo que hace que no requiera controladores adicionales. Otra característica que lo hace más barato es que utiliza el ATmega328 en su versión SMD o de montaje superficial que al fabricarse en cantidades enormes lo hace más económico.

La tarjeta de desarrollo Arduino UNO R3 CH340G SMD es una placa electrónica basada en el micro controlador ATmega328 cuenta con 14 pines

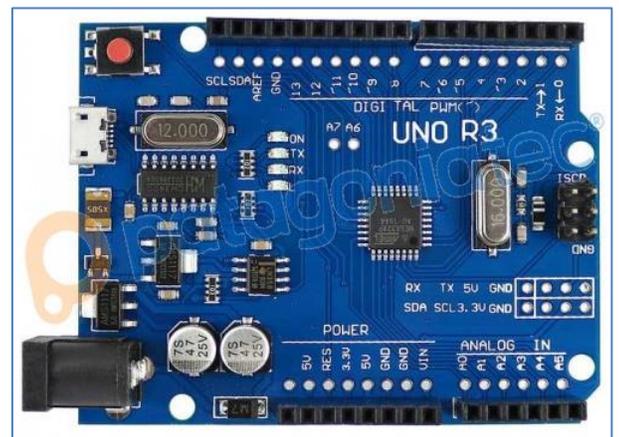


Ilustración 4: Arduino UNO con CH340

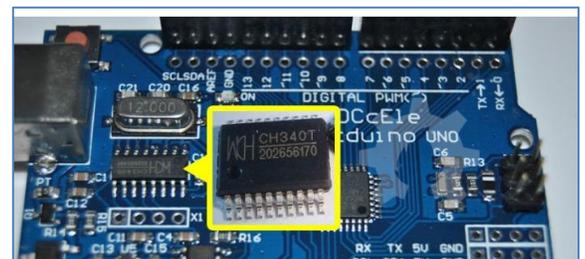


Ilustración 3: Chip CH340

in/out digitales (de los cuales 6 pueden ser utilizados como salidas PWM) y 6 entradas analógicas, un conector USB, uno de alimentación y un botón de reset.”

Experiencias, usos y Características

Esta placa, es una copia china de Arduino Uno R3, de las modificadas y editadas que existen, ya que Arduino es Hardware libre, y en esta versión, no se contuvieron a la hora de crearla.

Entre la diferencias que podemos encontrar en comparación con la anterior mencionada (muy fiel a la Arduino Uno original), es que en lugar de usar un Cable Serial para el puerto de datos, nos encontramos con un cable Mini USB a USB, lo cual puede ser considerado una gran ventaja ya que es un cable muy utilizado en nuestra vida cotidiana. Esta característica es típica de las Arduino Leonardo Originales.

También tiene 2 salidas de 5V, a diferencia de las Originales que traen uno.

Básicamente, es una mezcla entre Arduino Uno y Arduino Leonardo, pero es totalmente funciona como una UNO.

Se le dio uso para:

- Hacer alarmas detectoras de Fuego.
- Hacer alarmas detectoras de Humo.
- Uso intenso con LED (prendido y apagado).

Y en ninguna de estas actividades presento problemas, pero al ser una versión muy editada, si la conectamos simplemente no será reconocida por ningún IDE o Lenguaje de Programación para Arduino, antes debemos descargar el driver para que funcione.

Es muy simple de Instalar y sumamente fácil de encontrar, con simplemente googlear “Arduino Uno ch340 driver” será más que suficiente.



Conclusión

Por un precio que varía de 150 ARS\$ has unos 200 ARS\$(generalmente viene la placa sin cable) al día de la Fecha (Mayo del año 2017), esta placa supero totalmente nuestras expectativas. Por su baja precio y su procedencia extremadamente china, fue una compra que la hicimos con mucha desconfianza. Teníamos más esperanza que se auto destruya al momento de conectarla en lugar de que funcione, ¿y saben qué? ¡¡Resultado ser una placa de altísima calidad a bajo Precio!!

No dio ningún problema, no se quemó ante ningún error durante las conexiones y el uso de Cable Mini USB en el puerto de datos, facilito muchísimo su uso.

¡Sumamente Recomendable!

DuinoBot

Descripción General

“Duinobot es el microcontrolador, el cerebro detrás los robots desarrollados en Robotgroup. Puede ser utilizado tanto con Arduino, así como con los lenguajes pensados para nuestros robots. Para aquellos que ya tienen alguna experiencia programado ofrecemos la posibilidad de usar la potencia de DuinoPack, y para los que recién arrancan en el maravilloso mundo de la robótica y la programación,

también puede usar Duinobot con programación gráfica, haciendo uso de minibloq”

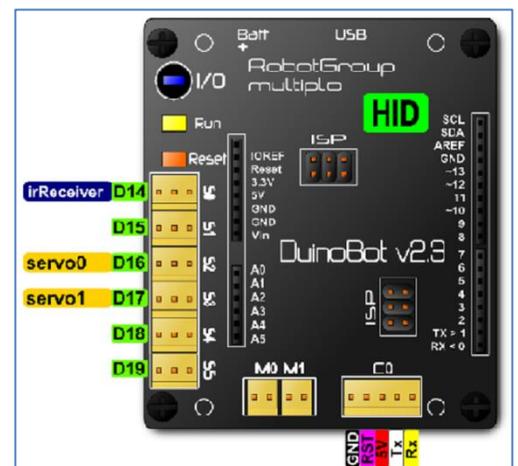


Ilustración 5: Placa DuinoBot

Características

Al ser una Arduino modificada tiene muchas características de la Arduino Uno, pero tenemos los siguientes agregados:

- 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.

Experiencias y Usos

Utilizamos esta placa gracias a los materiales brindados por el Profesor de Catedra Pedro Wingling, los cuales son Kits de la Empresa Robot Group una empresa argentina dedicada al diseño, fabricación, investigación y capacitación en robótica cuya misión es la de insertar la esta disciplina como sistema interdisciplinario de aprendizaje en todas las escuelas, colegios y universidades del país.



Ilustración 6: Placa DuinoBot

Por lo tanto, no solo se habla de la placa sino de todo el kit que viene con ella, el cual es realmente asombroso, por la gran variedad de artículos que trae: Sensores, Tornillos, Grampas, materiales de construcción de los Robots, cables, etc.; son incluidos con la placa lo que la hace un potente kit de inicio de Arduino.

Cabe destacar, que sus puertos D14 a D19, son excelentes físicamente por su tamaño (molex de 3 pines de conexiones de 1 pin como tienen las Arduino normales) para niños o personas que no le guste trabajar con cables chicos.

Además, estos puertos simplifican toda la tarea, ya que Electricidad, Datos y Tierras van todo junto por un solo puerto, simplificando y facilitando la tarea a la hora de conectar un sensor.

Podríamos considerar como único punto bajo, el IDE que trae por defecto o es recomendado por Robot Group: Minibloq (el cual detallare más adelante).

Su interfaz es realmente poco intuitiva y amigable, haciendo un gran problema querer replicar algoritmos que se hayan hecho en otros IDE.

Conclusiones

Es una Excelentísima opción, si la intención es iniciarse en el mundo de Arduino con un Gran Kit que te permite llevar a cabo muchísimos proyectos. Sus puertos de Conexiones y su simpleza hacen que conectar un sensor se reduzca a conectar 1 solo cable (en lugar de 3 o 4 que necesitan las placas de Arduino UNO).

El punto bajo es su IDE Minibloq, que como mencione antes, es poco intuitivo y nada recomendable en mi opinión.

Sensores y otros módulos que utilizamos

Sensor de Llama KY-026

Descripción General

“El Sensor KY 026 es capaz de detectar llamas y posee una estructura LED detectora de fuego, cuando el circuito capta las ondas emitidas por la llama este enciende una advertencia de tal modo que enciende un LED que ya está integrado en el sensor.



Ilustración 7: Sensor de Llama KY-026

La longitudes de ondas de llama pueden ser detectadas entre 760 nm y los 1100nm, cuando utiliza métodos infrarrojos es más sensible 60 grados.”

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.

Experiencias y uso

Como podemos observar en la siguiente imagen, a la hora de realizar la compra de los sensores, si bien son el mismo modelo (KY 026), tal parece que son de fabricantes diferentes, y esto en su uso marco una pequeña diferencia.

Los sensores azules (comprados en www.nubbeo.com.ar), resultaron tener muchísimo mejor rango y detección de las llamas. Ambos fueron testeados con llamas producidas por fósforos y por encendedores y a pesar de la diferencia de rango o detección (la cual debe ser una cuestión de calidad de fabricación), su funcionamiento y conexión no varía en uno del otro.

Datos a considerar en el uso del sensor, es su conexión y su potenciómetro.

El Potenciómetro, según como lo ajustemos, nos aumenta el rango de detección, y si lo llevamos al máximo puede incluso confundir el fuego con la luz de un Foco, así que es recomendable dejarlo en la posición de fábrica.

Respecto a las conexiones, nos encontramos con los siguientes pines, y que deberemos tener en cuenta más adelante para cuando veamos como armar una alarma de fuego:

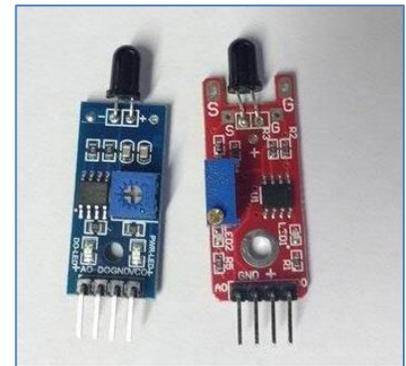


Ilustración 8: Sensores de Llama KY-026 - Azul y Rojo

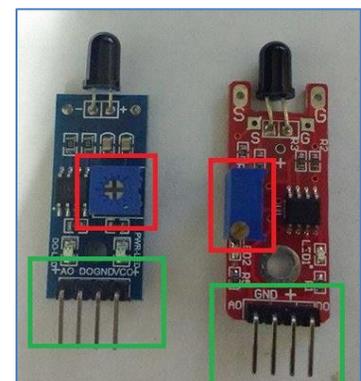


Ilustración 9 En Rojo: Potenciómetro | En Verde: Pines de Conexión

- **GND:** Ground o Tierra, es indispensable conectar esto a la GND de una protoboard o de la Placa Arduino para que no se queme el sensor. Todo sensor, modulo o extensión de Arduino cuenta con una conexión GND.
- **VCC o +:** La alimentación, este pin nos dará electricidad y por lo tanto encenderá nuestro sensor, al igual que el pin de GND, todo sensor, modulo o extensión de Arduino cuenta con una conexión GND.
- **DO (Digital Output):** A través de este pin, el sensor nos envía información **Digital**, la cual se caracteriza por utilizar 2 valores: 1 (HIGH) y 0 (LOW). Por lo tanto, si el sensor detecta fuego, nos enviara un 1 o HIGH, en cambio si no detecta fuego nos enviara constantemente un 0 o LOW. En el caso de los sensores de llama, este será el pin que utilizaremos para conectar nuestra placa.
- **AO (Analog Input) :):** A través de este pin, el sensor nos envía información **Analógica**, la cual se caracteriza por utilizar un Rango de valores. Por lo tanto, si el sensor detecta fuego, nos enviara valores más altos en la medida que más cerca detecte el fuego.

Conclusiones

Con un precio de 70 ARS\$ al día de la fecha (Mayo del año 2017), aunque tuvimos el mismo sensor de 2 fabricantes diferentes, no presentaron ningún problema en general, y su detección de llamas es inmediata. Lo que lo hace sumamente recomendable si necesitas encarar un proyecto donde necesites detección de llamas.

Lo único a tener en cuenta, es que algunos de estos sensores pueden tener mejor fabricación que otros, y por ende mejor detección al fuego (a mayor distancia). Otro factor a tener en cuenta, es que hay que tener cuidado con el potenciómetro de no darle demasiada potencia, ya que afectaría a la correcta detección de llamas.



Sensor de Humo y Gases MQ2

Descripción General

“La serie MQ de sensores usan un precalentador en su interior y un sensor electroquímico. Son sensibles a una gran cantidad de gases. El MQ2 es particularmente sensible al gas natural de hornallas lo que lo hace ideal para domótica. También detecta LPG butano, propano, metano, alcohol, hidrógeno y "humo". (GNC metano hidrogeno LPG monóxido CO propano)”

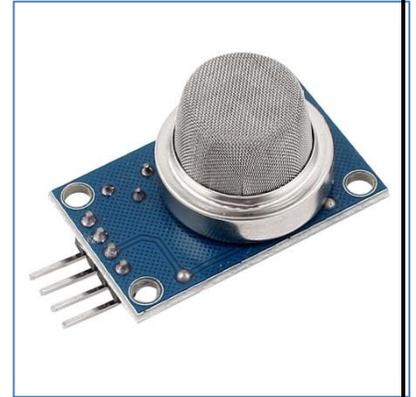


Ilustración 10: Sensor de Humo y Gases MQ2

Características

- Alimentación: 5V.
- Consumo: 122mA.
- Salida Digital y Analógica.
- Led de encendido.
- Led de accionamiento salida digital.

Experiencias y uso

El sensor MQ2, fue utilizado para hacer alarmas detectoras de gases y de humo.

Básicamente lo que hace, es detectar e gas en el ambiente o el nivel de humo en el ambiente. Generalmente, es utilizado con la señal analógica, donde nos envía la información del nivel de gases o humo que encuentra en el ambiente.

Nos encontramos que a la hora de utilizar el IDE Mblock (del cual hablaremos más adelante) el sensor presenta fallas.

El al igual que el sensor de fuego, el sensor cuenta con un potenciómetro y los pines de conexión **VCC, AO, DO Y GND**.



Ilustración 11: En verde: Potenciómetro | En Rojo: Pines de Conexión.

Respecto al Potenciómetro, al igual que el sensor de Llamas, aumentar el rango de este puede ocasionar un mal funcionar del sensor, y sobre todo de este, ya que al aumentar su potencia, cualquier gas que encuentre naturalmente en el ambiente será detectado. Por lo tanto, nuevamente recomendamos dejarlo en su estado de fábrica.

Respecto a las conexiones, nos encontramos con los siguientes pines, y que deberemos tener en cuenta más adelante para cuando veamos como armar una alarma de fuego:

- **GND:** Ground o Tierra, es indispensable conectar esto a la GND de una protoboard o de la Placa Arduino para que no se quemé el sensor. Todo sensor, modulo o extensión de Arduino cuenta con una conexión GND.
- **VCC o +:** La alimentación, este pin nos dará electricidad y por lo tanto encenderá nuestro sensor, al igual que el pin de GND, todo sensor, modulo o extensión de Arduino cuenta con una conexión GND.
- **DO (Digital Output):** A través de este pin, el sensor nos envía información **Digital**, la cual se caracteriza por utilizar 2 valores: 1 (HIGH) y 0 (LOW). Por lo tanto, si el sensor detecta humo o gases, nos enviara un 1 o HIGH, en cambio si no detecta humo o gases y nos enviara constantemente un 0 o LOW. En el caso de los sensores de Humo, este pin no lo utilizamos, ya que no nos da como resultado una proporción del nivel de humo en el ambiente.
- **AO (Analog Input):** A través de este pin, el sensor nos envía información **Analógica**, la cual se caracteriza por utilizar un Rango de valores. Por lo tanto, si el sensor detecta Humo o Gas, nos enviara valores más altos en la medida que más cerca detecte el fuego. Este pin será el que utilizaremos en la alarma de humo y gas, ya que nos da un rango de valores que representa el nivel de humo o gas en el ambiente.

Otra cosa que debo recalcar, que el sensor detecta más rápido el gas que con el Humo (los valores aumentan más rápido con gas que con humo en su señal analógica), tal vez los



sensores de mayor gama (MQ3 o MQ7) sea más sensible al humo que al gas.

Speaker (Parlante)

Descripción general

Los Speakers o Parlantes, no son nada más ni nada menos que los parlantes interiores de equipos de audio pequeños, parlantes antiguos o generalmente los que vienen en los más comunes y de bajo precio.



Ilustración 12: Parlante/Speaker

Experiencia y Usos

Simplemente, desarmamos un par de parlantes antiguos y quitamos el parlante de su interior. Hacerlo funcionar es muy simple, generalmente como vemos en la Ilustración 8 tienen 2 cables:

- Cable Negro: Este cable es la tierra o GND, debemos conectarlos para que no se quemé y funcione correctamente nuestro parlante.
- Cable de color: El color del cable de color suele variar (Rojo, Amarillo, Verde, Azul, etc.), pero suele representar el cable de entrada digital o DI (Digital Input). Conectando dicho cable a un puerto digital de Arduino y mandarle una señal digital, será todo lo que necesitemos para hacerlo sonar.

Conclusiones

¡Que más decir!, ¡se consiguen fácilmente y son muy útiles!, si tienes un equipo de audio viejo, ¡no lo tires!

LED de 5mm

Descripción General

Leds de tamaño de 5mm, elegidos de este tamaño ya que los de 3mm son demasiados pequeños a mi parecer.



Ilustración 13: LEDs de 5mm

Experiencia y Usos

Los LED de 5mm, los utilizamos para mostrar el estado de detección de las alarmas. Encendíamos un LED Rojo (peligro) cuando algún sensor detecta la precedencia de gas/humo/llama y en caso que no encuentre nada, un LED Verde (Todo ok).

Por si nunca utilizaron un LED, la forma de conectarlo es la siguiente:

Como vemos en la Ilustración 10, el led tiene dos pines o "Patas", las cuales no son así por que sí.



Ilustración 14: "Patas" o Pines de un LED

SIEMPRE deberemos conectar el pin más largo (o pata más larga) a una entrada digital y la pata más corta a tierra, caso contrario, quemaremos el led.

Software

En este apartado voy a detallar los Software que utilizamos para llevar adelante los 2 proyectos que realizamos: Alarma de Fuego y Alarma de Humo y Gases.

Específicamente, los Software que utilizamos son IDE'S (Entornos de Desarrollo) para Arduino.

Los IDE que revisamos y vimos fueron los siguientes.

- Visualino
- Mblock
- Minibloq

- Arduino (IDE por defecto de fábrica)
- Scratch For Arduino

Sin embargo, en este informe profundizaremos solo en Mblock, Minibloq y Arduino, ya que estos son los únicos con los cuales llevamos a cabo nuestros proyectos. El motivo de su elección fue por 2 razones:

1. Su interfaz gráfica y elementos que permitan la fácil enseñanza en la escuela secundaria
2. Su Compatibilidad respecto a los sensores que compramos y las placas que utilizamos.

Entornos de Desarrollo

Mblock

Descripción General

Mblock, tiene la particularidad de ser un IDE o Lenguaje de Programación Blocky o Icónico (programación a través de bloques), el cual está desarrollado a partir de Scratch (de aquí su interfaz idéntica, que a su vez, a la hora de programar, nos muestra la equivalencia del algoritmo que hayamos construido, con su equivalente en el código de



Ilustración 15: Logo de Mblock

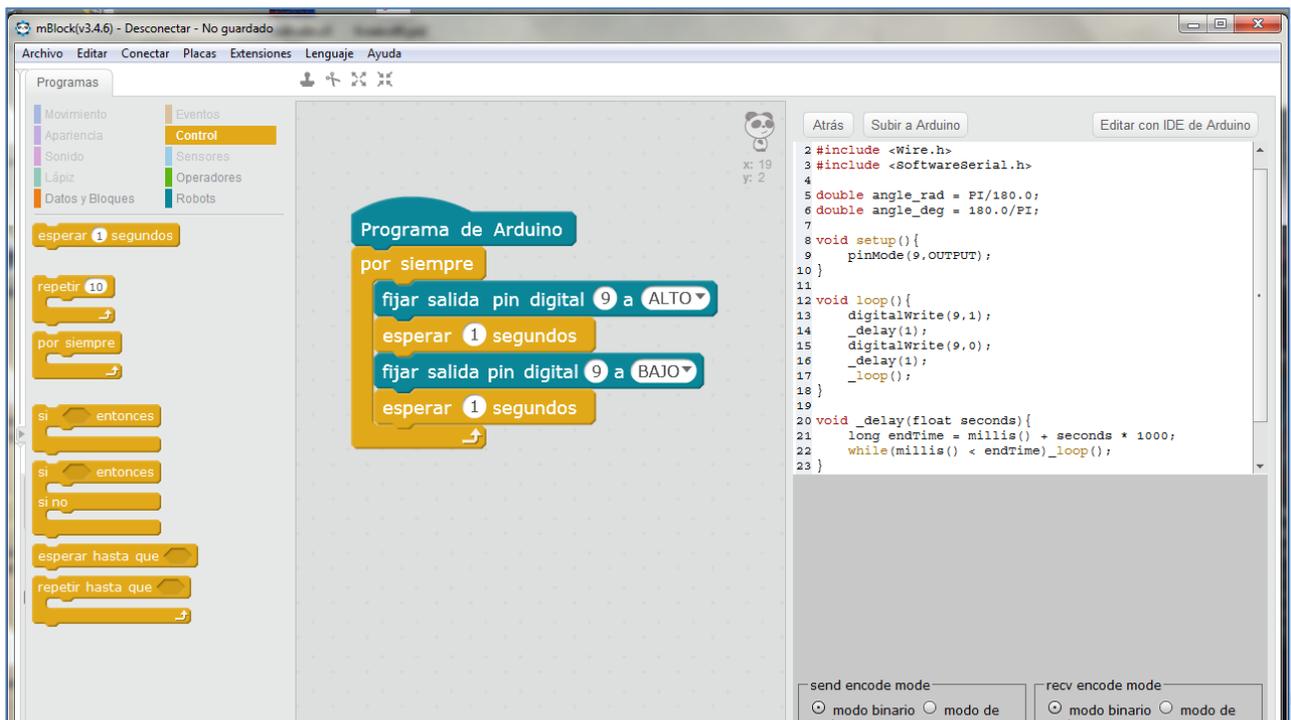


Ilustración 16: Equivalencia -> Programa en Bloques / Programa en Código Arduino

Arduino (C++).

Experiencias y Usos

Aprender a usar Mblock es muy sencillo, sobre todo si tenes conocimientos sobre Scratch, porque es prácticamente idéntico. Su equivalencia de código entre Bloques y Código Arduino lo hace una excelente herramienta a la hora de educar con él.

Pero no todo es magia, tiene un defecto bastante particular:

Muchas veces, a la hora de cargar un programa, si seguir ningún patrón ni tener motivos, los programas cargados no funcionan como se deben, por ejemplo: El sensor de Fuego no Detecta Fuego en el estado HIGH sino en el LOW, el sensor de Humo ni siquiera funciona, etc.

Lo cual hace que muchas veces pensemos que la programación que hayamos realizado es incorrecta, pero todo lo contrario, el problema fue Mblock. No digo que este problema sea recurrente, pero en un 80% de los casos funciona bien el IDE, pero a veces tiene estas fallas.

Conclusiones

Es una excelente forma de comenzar la programación en Arduino, ya que a diferencia del código C++ que por naturaleza usa el IDE de Arduino, tal vez no sea lo más apropiado para iniciarnos en el mundo de Arduino si no tenemos conocimientos sólidos en Programación.

Al ser desarrollado en base a Scratch, su interfaz intuitiva y amigable hace que sea uno de nuestros favoritos y el que utilizamos para todos los proyectos que llevamos a cabo, ya que esto permite llevar el mundo de Arduino a las personas que no poseen conocimientos de Programación.

Pero tiene este gran defecto, que cada tanta falla, y más de una vez te dará un dolor de cabeza.



IDE Arduino

Descripción General

Se llama IDE ya que en inglés significa "Integrated Development Environment" que significaría "Entorno de Desarrollo Integrado". Este es el Software de Fábrica o por defecto, que los creadores de Arduino nos proporcionan para desarrollar nuestros programas en Arduino. Su código es una versión del Lenguaje de Programación C++, modificado para la placa Arduino.

Esto hace que tenga un gran potencial a la hora de desarrollar diferentes proyectos de electrónica, robótica o domótica.



Ilustración 17: Logo del IDE Arduino

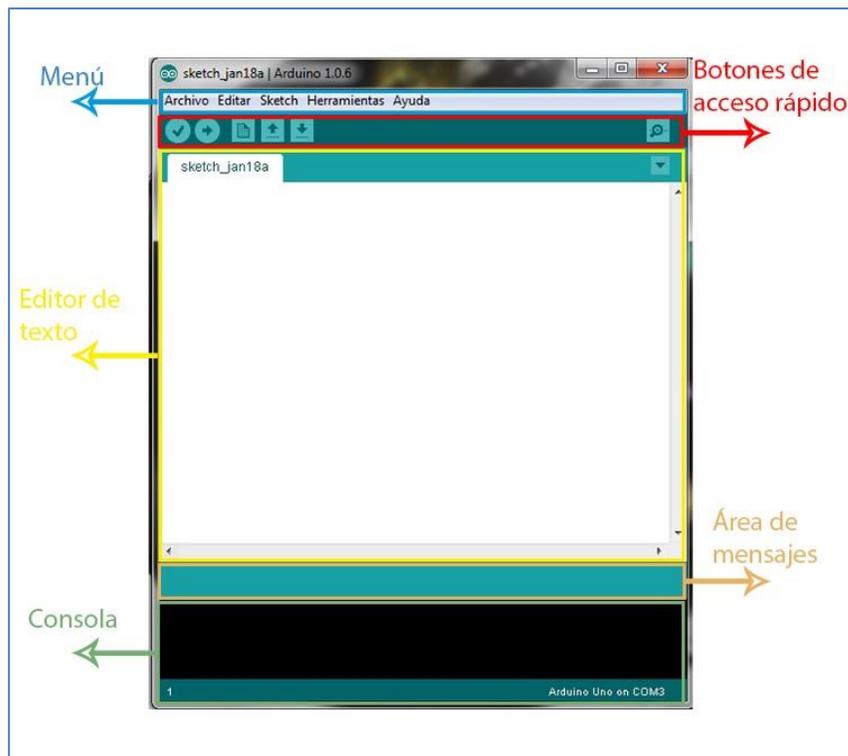
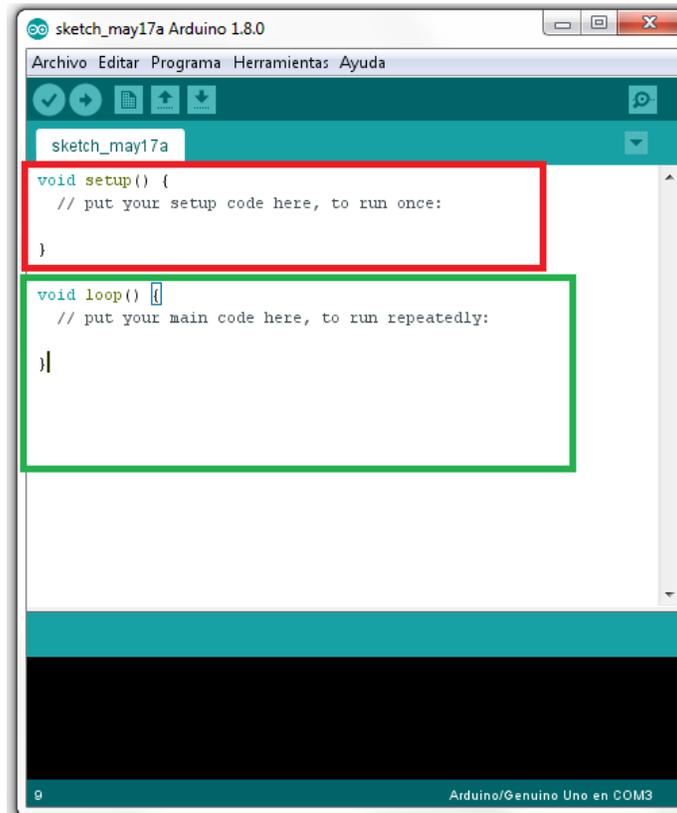


Ilustración 18: IDE Arduino

Tiene 2 grandes estructuras a partir de las cuales trabajar su código:



```
sketch_may17a Arduino 1.8.0
Archivo Editar Programa Herramientas Ayuda
sketch_may17a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Ilustración 19: Set Up - Loop en el IDE de Arduino

Como vemos en la Ilustración 15 en Rojo, tenemos la sentencia **SetUp**, es aquí donde definiremos nuestras variables, sensores, etc.

Luego tenemos en verde la sentencia **Loop**, aquí es donde ingresaremos el algoritmo de nuestro proyecto de Arduino, y este se repetirá siempre.

Esta es una característica especial de la sentencia repetitiva Loop (su nombre viene de la palabra "Loop" -> en español ciclo), ya que funciona igual que muchos partes de los sistemas operativos que están recibiendo continuamente información.

El IDE aplica esto a los sensores, módulos, accesorios, etc.; ya que constantemente busca información de estos.

Experiencias y Usos

Decir que usamos o Programamos en el IDE de Arduino básicamente seria mentir, ya que nos enfocamos más en Mblock que otro IDE.

Pero, sin embargo, cuando fallaba Mblock, testeábamos dichos códigos en él, y aquí fue cuando nos dimos cuenta de la falla que presenta Mblock de vez en cuando.

Conclusiones

Si tenes conocimientos en programación, es excelente (ya que no presenta ninguna falla) y la opción por defecto para programar en Arduino. Caso contrario, existen otros IDE'S con interfaces graficas más atractivas las cuales pueden iniciarte en el mundo de Arduino mas fácil.

Minibloq

Descripción General

Entorno Grafico desarrollado en Argentina, el cual permite de forma nativa la programación en las placas DuinoBot.

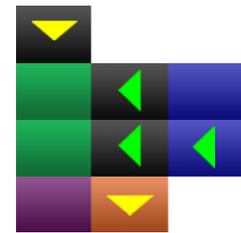


Ilustración 20: Icono de Minibloq

Experiencias, Usos y Conclusión

A pesar de la facilidad que nos brindan las placas DuinoBot de los Kit de RobotGroup, su IDE es todo lo contrario. Pésima Interfaz Gráfica, poco intuitivo y extremadamente incomodo a la hora de programar. Si se puede evitar utilizarlo por otras alternativas, ¡Hágalo!

Inclusión en la Curricula

Este apartado lo dividiré en 2 secciones.

Primero, los **ejes desde los materiales curriculares de la provincia de la Pampa** a través de los cuales podemos enseñar Robótica Educativa.

Y Segundo **guía de como realizamos los 2 proyectos**: Alarma de Humo y Alarma de Fuego

En MiniBlock.

Ejes de los Materiales Curriculares

El porqué de la utilización de las placas de Arduino, y la Robótica Educativa en las escuelas secundarias, tiene una simple explicación tomando conceptos de la Asignatura “Didáctica” de la autora **Vilma Pruzzo**.

En su libro “**EVALUACIÓN CURRICULAR: EVALUACIÓN PARA EL APRENDIZAJE** “, hace mucho hincapié en la enseñanza en base Epistemológica.

¿Y qué es esto? Significa, enseñar de tal forma, que recreemos el proceso de construcción de conocimiento de una ciencia específica.

Esto en otras palabras, significa, recrear en el aula como aprenden e investigan los investigadores de nuestra ciencia, la computación.

Ya que no existe mejor manera de aprender investigando, la cual es la manera a través de la cual se crea el conocimiento de las ciencias.

En este proceso, se busca despertar en los alumnos, procesos mentales propios de nuestra disciplina.

Por lo tanto, si queremos motivar y realmente despertar el interés de nuestros alumnos, llevarlos a la práctica y al contacto directo con la Robótica, sería una excelente manera de llevar a nuestros alumnos a una Enseñanza en Base Epistemológica.



A la hora de plantear la creación de diferentes proyectos de Arduino y de Robotica, estamos llevando a los alumnos a explorar, investigar y crear, en base a los conceptos y saberes que pretenden que los alumnos adquieran en las diferentes materias informáticas del secundario.

Ya no estaríamos dando un texto de Hardware, ya no estaríamos dando un Texto de Software, estaríamos dando un gran paso, ya que estaríamos creando Hardware y Software. Como lo hacen los científicos de nuestra ciencia.

Asignatura: Tecnología de los Sistemas Informáticos

EJE: Tecnología aplicada al Hardware.

Me enfocare en los siguientes saberes y fundamentare, porque me parece importante

- ***Analizar la evolución tecnológica de la electrónica, que permitieron la***

Evolución de la computadora.

A través de pequeños proyectos de Arduino, podemos representar y recrear, pequeñas partes del Hardware de una computadora.

Sin contar, el estudio de la parte electrónica de las placas Arduino para el entendimiento de los circuitos del Hardware (Resistencia, voltaje, Bus, etc.)

Displays, Sensores, Leds o Coolers son algunos de los proyectos que podemos encarar para entender el funcionamiento del Hardware de una computadora.

- ***Conocer y comprar los distintos hardwares con respecto a los de la***

Computadora personal.

Arduino tiene la peculiaridad de ser Hardware Libre, y una placa de Robótica que es un Micro Controlador, con el cual podemos explorar todas sus versiones y

diversificaciones respecto a su hardware.

También, la realización de cualquier proyecto de Arduino (Aparatos pequeños de electrónica, Robots Simples, etc.).

- ***Entender el funcionamiento interno de distintos sistemas tecnológicos que usan sistemas informáticos para realizar el proceso de comunicación.***

Nuevamente, encarando desde el punto de vista de la variedad de proyectos que nos puede brindar Arduino, podemos explorar o dividir en pequeñas partes distintos sistemas tecnológicos que utilizan las computadoras para comunicarse.

Receptores de Wifi, Bluetooth y Controles de Infra Rojos son algunos pocos ejemplos de los proyectos que se pueden encarar para entender en pequeña escala, el funcionamiento de pequeños dispositivos que dentro de una computadora, permiten la comunicación.

Asignatura: Tecnología de la Información y la Comunicación

EJE: Tecnología que median la gestión de la información y la presentación de los resultados.

Me enfocare en los siguientes saberes y fundamentare, porque me parece importante:

- ***La Pre-programación en forma sencilla en cualquier software que permita resolver situaciones problemáticas desde la lógica o con la utilización de algoritmos sencillos.***
- ***Conocer y utilizar software que resuelva problemas planteados a través de Algoritmos o funciones lógicas.***
- ***Desarrollar proyectos que permitan aplicar software sencillo a problemáticas Reales.***

Estos 3 saberes que propone el material curricular de la asignatura TIC, son los más orientados a la programación, pero sobre todo la búsqueda **“aplicar software sencillo a problemáticas reales”**, es un gran punto fuerte que desde la robótica se puede abordar.

Nuevamente, desde una enseñanza en base epistemológica, investigar y crear diferentes proyectos que solucionen problemas de la vida cotidiana, y de estos ejemplos hay muchos: Alarmas, Generadores de Energía Sencillos (Eólica, Solar, etc.) son algunos de los ejemplos que se pueden dar.

Proyectos Realizados (En Mblock)

Alarma de Fuego

El objetivo de este tutorial, es crear una alarma la cual, a partir de la detección de la llama, sonara un parlante y se encenderá un led rojo.

Caso contrario (no detectar una llama), el parlante no emitirá sonido alguno, y un led verde estar prendido, indicando que no estamos en peligro.

Primero, veremos los bloques necesarios en Mblock para llevar a cabo la alarma:

Bloques

1) “Programa Arduino y “por siempre””: Estos bloques son la base de un programa en Mblock, ya que básicamente nos indican, que al hacer doble click en programa Arduino se ejecutara continuamente (“Por siempre”) lo indicado.



Ilustración 21: Bloques -> Programa Arduino y por siempre

2) Bloque “Si <> entonces -> Sino”:

Este bloque, es de los llamados bloques “Selectivos”, sirven para evaluar situaciones. En nuestro caso, si el sensor nos indica que encontró fuego, **prenderemos un led rojo y haremos sonar el parlante**. Dentro del <> ira la condición.



Ilustración 22: Si <> Entonces -> Si no

3) Bloque “leer pin digital ()”:

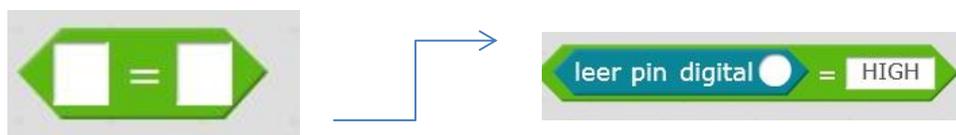
este bloque me permite, leer información de un Pin en específico ingresado en el espacio en blanco. Como hemos mencionado anteriormente, utilizásemos en la salida digital (Digital Output) del sensor de llama, por lo tanto en el espacio en blanco colocaremos el pin digital al que fue conectado el pin DO del sensor.



Ilustración 23: Leer Pin Digital

4) Operador “[] = [] “:

Nos servirá para evaluar la información que nos manda el sensor al pin digital, por lo tanto si encastro el bloque anterior a este quedaría...



Ahora, si encastro con el “Si <> Entonces -> Sino” quedaría

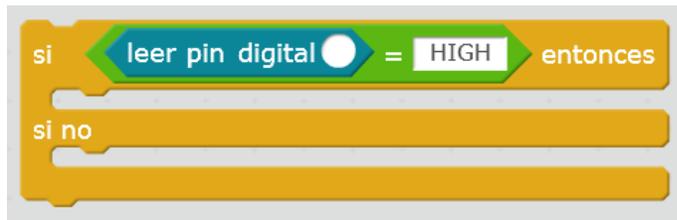


Ilustración 24: Si + Leer Pin digital

Este conjunto de Bloques se interpretaría: “Si el pin digital detecta llama entonces -> Si no”. El HIGH me indica que ha sido detectada la llama por el sensor, como anteriormente mencione, Cuando el sensor detecta llama, envía por su pin DO HIGH o 1.

Si fuera LOW, estaría fijándome el caso que no se detecte llama.

5) “Fijar salida Digital () (Alto/bajo)”: Fijar salida digital, nos permite enviar un pulso digital a cualquier dispositivo que conectemos al Pin que designemos en el primer espacio en blanco. En nuestro caso, prender y apagar los leds que necesitamos



Ilustración 25: Fijar Salida Digital ALTO – BAJO

En este caso, nos servirá para encender un LED, si tenemos el led conectado por ejemplo en el Pin 10 y queremos encenderlo:

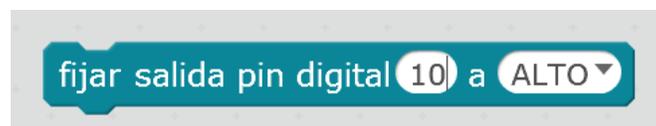


Ilustración 26L: LED 10 Encendido

En caso que deseemos apagarlo, seleccionaremos LOW en lugar de HIGH.

6) “Reproducir tono () en nota () beat ()”: Este bloque nos permitirá enviar un pulso digital al parlante (espacio en blanco indicaremos el pin digital donde hallamos conectado el parlante) y esto hará que suene.



Ilustración 27: Reproducir Tono

Como resultado final de combinar todos estos bloques, nos debería quedar el siguiente Programa:

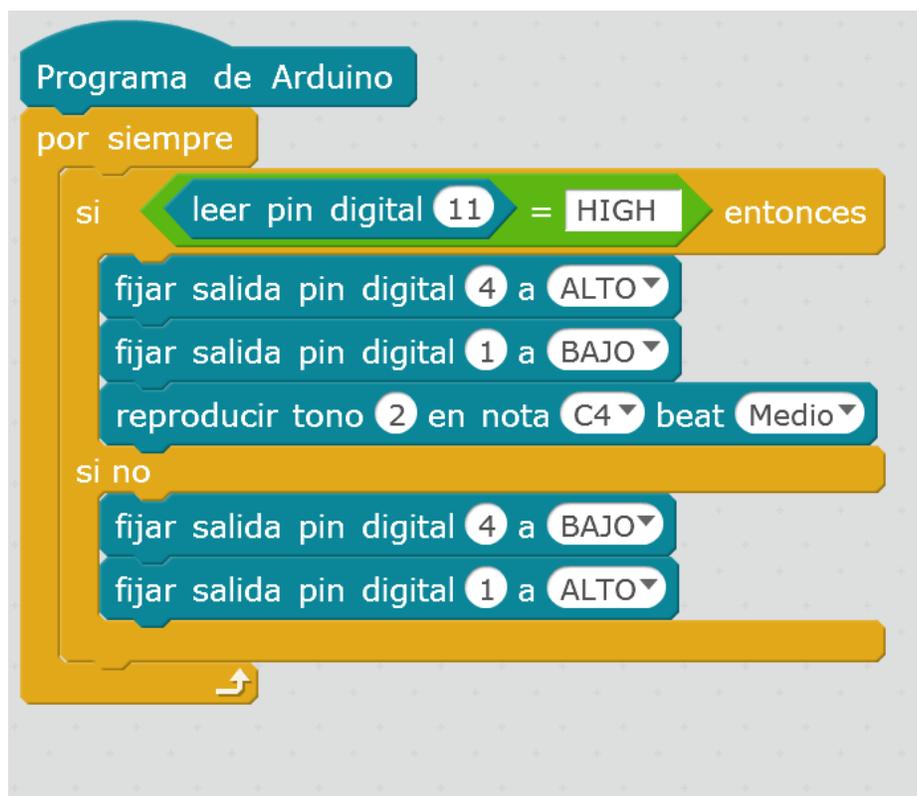
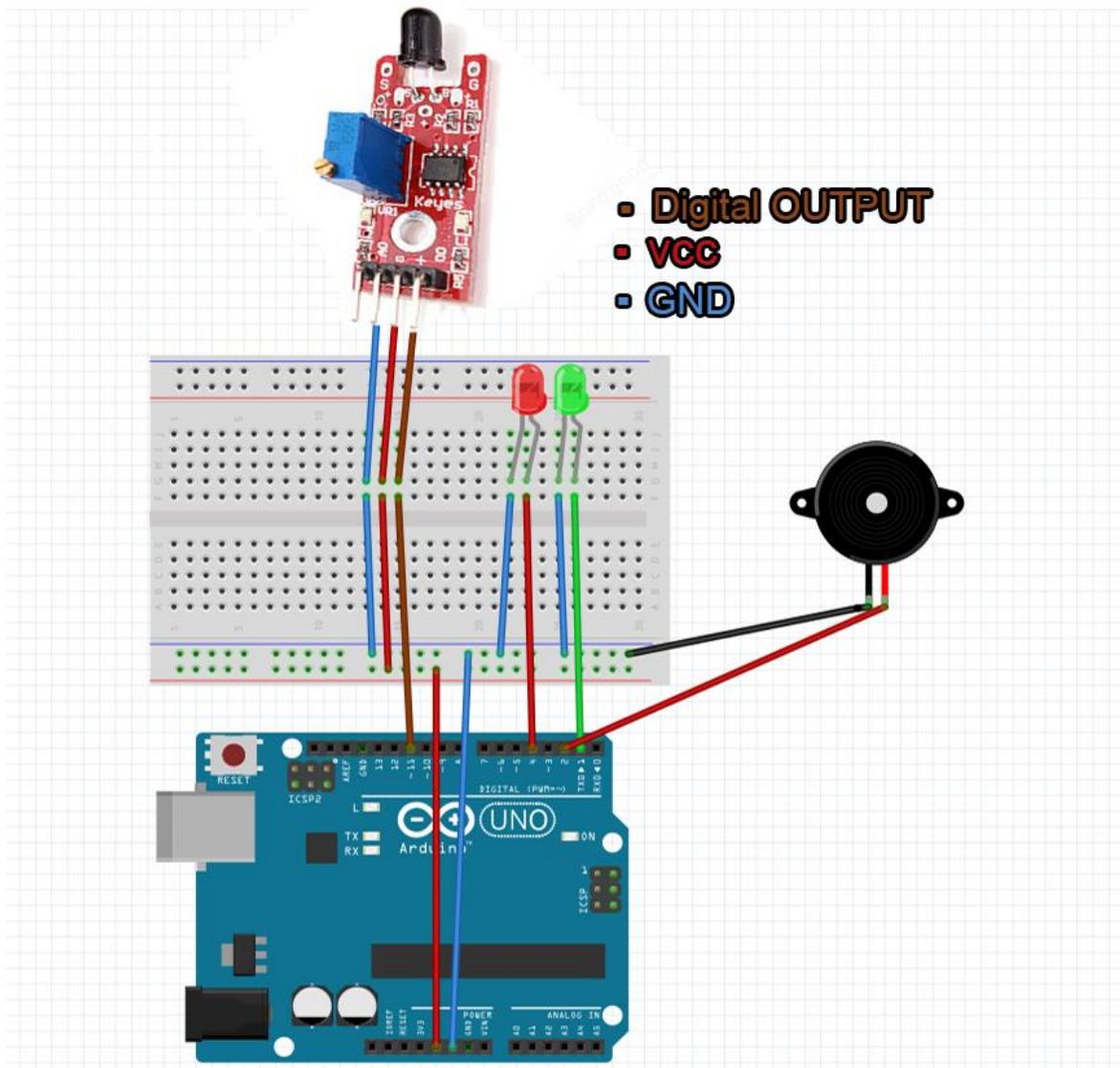


Ilustración 28: Programa de Alarma de Fuego

Ahora, solo falta la conexión:

Conexión



Con solo cargar el programa en nuestro Arduino, ¡la alarma debería funcionar!

Alarma de Humo

El objetivo de este tutorial, es crear una alarma la cual, a partir de la detección de humo o gases, sonara un parlante y se encenderá un led rojo.

Caso contrario (no detectar humo o gases), el parlante no emitirá sonido alguno, y un led verde estará prendido, indicando que no estamos en peligro.

Primero, veremos los bloques necesarios en Mblock para llevar a cabo la alarma:

Bloques

1) “Programa Arduino y “por siempre””: Estos bloques son la base de un programa en Mblock, ya que básicamente nos indican, que al hacer doble click en programa Arduino se ejecutara continuamente (“Por siempre”) lo indicado.



Ilustración 29: Bloques -> Programa Arduino y por siempre

2) Bloque “Si <> entonces -> Sino””: Este bloque, es de los llamados bloques “Selectivos”, sirven para evaluar situaciones. En nuestro caso, si el sensor nos indica que encontró fuego, **prenderemos un led rojo y haremos sonar el parlante**. Dentro del <> ira la condición.



Ilustración 30: Si <> Entonces -> Si no

3) Bloque “leer pin analógico ()”: este bloque me permite, leer información de un Pin en específico ingresado en el espacio en blanco. Como hemos mencionado anteriormente, utilizaremos en la salida Analógica (Analog Output) del sensor de MQ2, por lo tanto en el espacio en blanco colocaremos el pin analógico al que fue conectado el pin AO del sensor.



Ilustración 31: Leer pin analógico

4) Operador “[] > [] “: Nos servirá para comparar la información que nos manda el sensor al pin analógico con un valor dado, por lo tanto si encastro el bloque anterior a este quedaría...



Ilustración 33: Operador Mayor



Ilustración 32: Leer pin analógico + Operador Mayor

Ahora, si encastro con el “Si <> Entonces -> Sino” quedaría

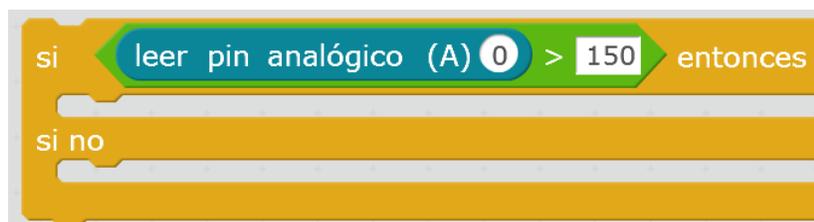


Ilustración 34: Si + Leer Pin Analógico

Este conjunto de Bloques se interpretaría: “Si el pin analógico detecta humo o gases

entonces -> Si no”.

El “Leer Pin Analógico () > 150” me indica que se ha detectado un nivel de humo o gas mayor que el normal en el ambiente. El valor normal en el ambiente detectado por el sensor puede variar entre 150 y 200, es importante chequear eso para el funcionamiento correcto de la alarma. Cuando el sensor detecta llama, envía por su pin DO HIGH o 1.

Si fuera LOW, estaría fijándome el caso que no se detecte llama.

5) “Fijar salida Digital () (Alto/bajo)”: Fijar salida digital, nos permite enviar un pulso digital a cualquier dispositivo que conectemos al Pin que designemos en el primer espacio en blanco. En nuestro caso, prender y apagar los leds que necesitemos



Ilustración 35: Fijar Salida Digital ALTO – BAJO

En este caso, nos servirá para encender un LED, si tenemos el led conectado por ejemplo en el Pin 10 y queremos encenderlo:

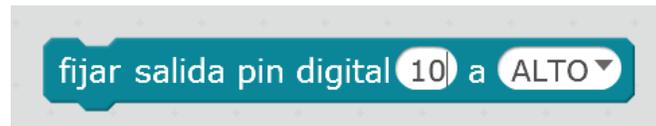


Ilustración 36L: LED 10 Encendido

En caso que deseemos apagarlo, seleccionaremos LOW en lugar de HIGH.

6) “Reproducir tono () en nota () beat ()”: Este bloque nos permitirá enviar un pulso digital al parlante (espacio en blanco indicaremos el pin digital donde hallamos conectado el parlante) y esto hará que suene.

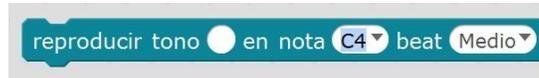


Ilustración 37: Reproducir Tono

Como resultado final de combinar todos estos bloques, nos debería quedar el siguiente Programa:

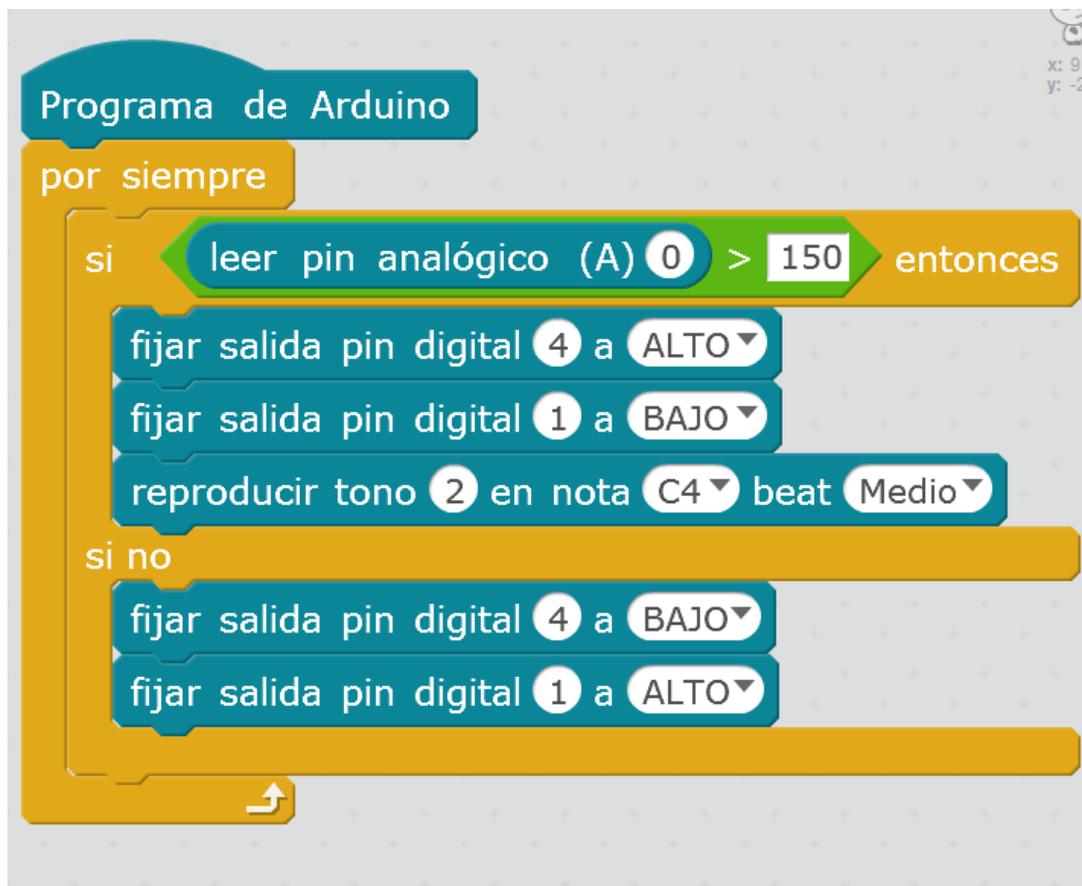
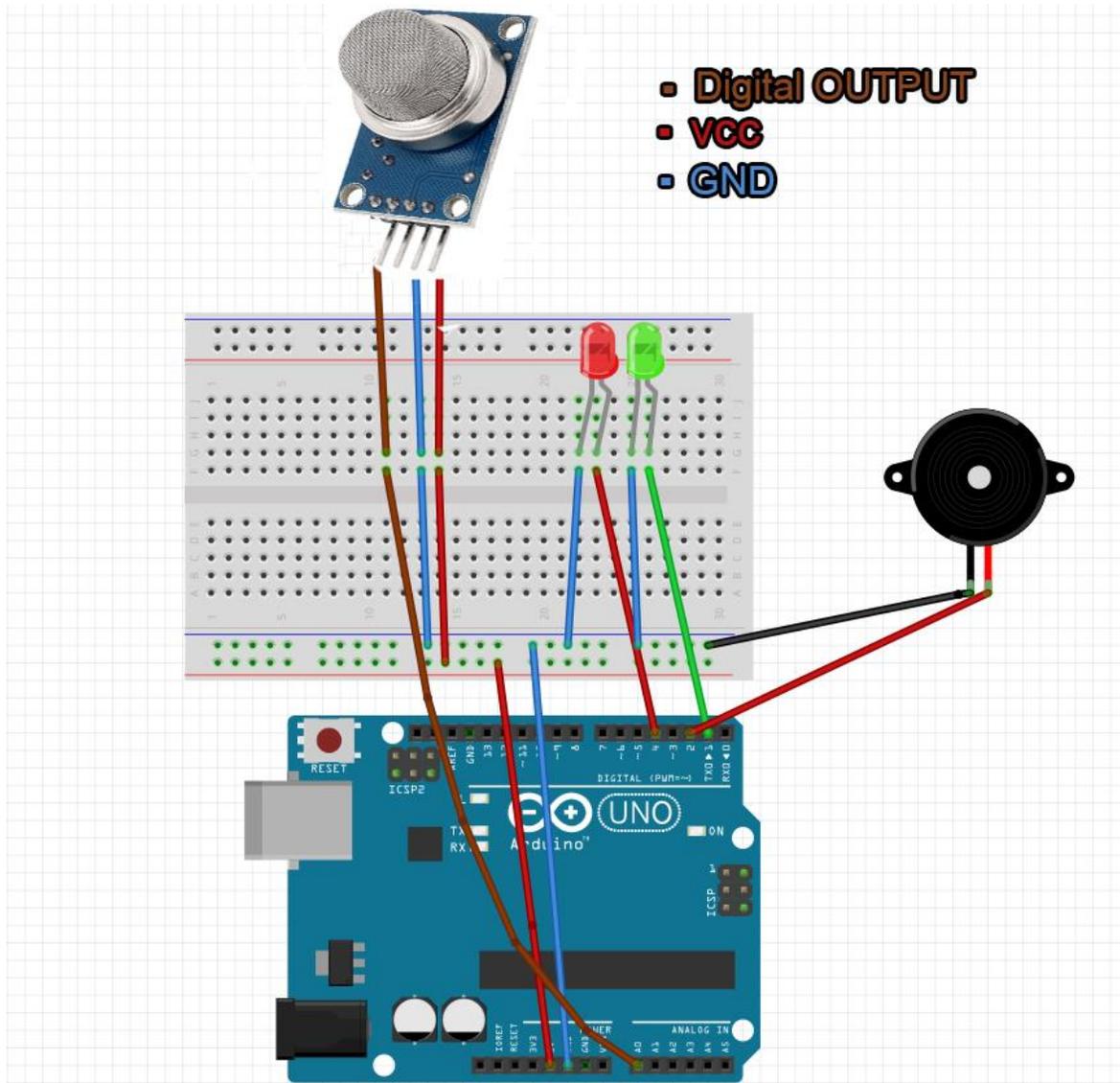


Ilustración 38: Programa Alarma de Humo / Gases

Ahora, solo falta la conexión:

Conexión



Con solo cargar el programa en nuestro Arduino, ¡la alarma debería funcionar!

Robot Group: Robot N8

Descripción

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.

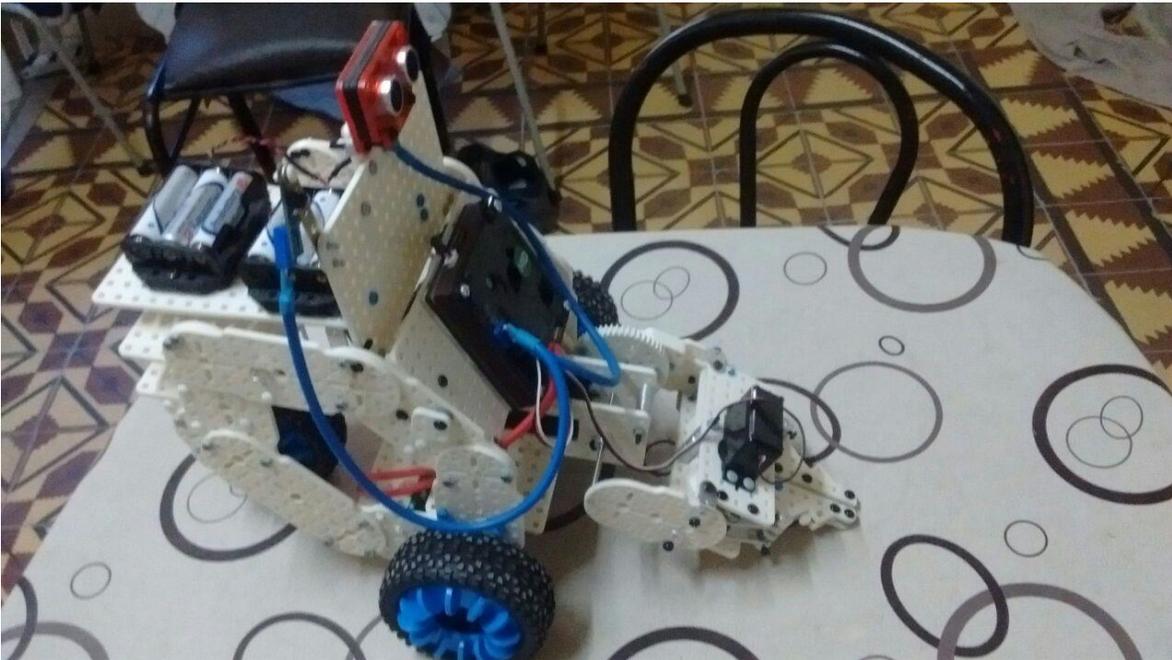


Ilustración 40: Robot N8 ensamblado

El mismo tiene 2 ruedas que se encuentran conectadas al placa DuinoBot v2.3, tiene una pinza con desplazamiento vertical y tiene la función de abrirse o cerrarse.

Con respecto al movimiento de las ruedas y el de la pinza, para la manipulación de los mismos utilizamos un sensor infrarrojo de RobotGroup, el cual recibe señal de un control remoto.

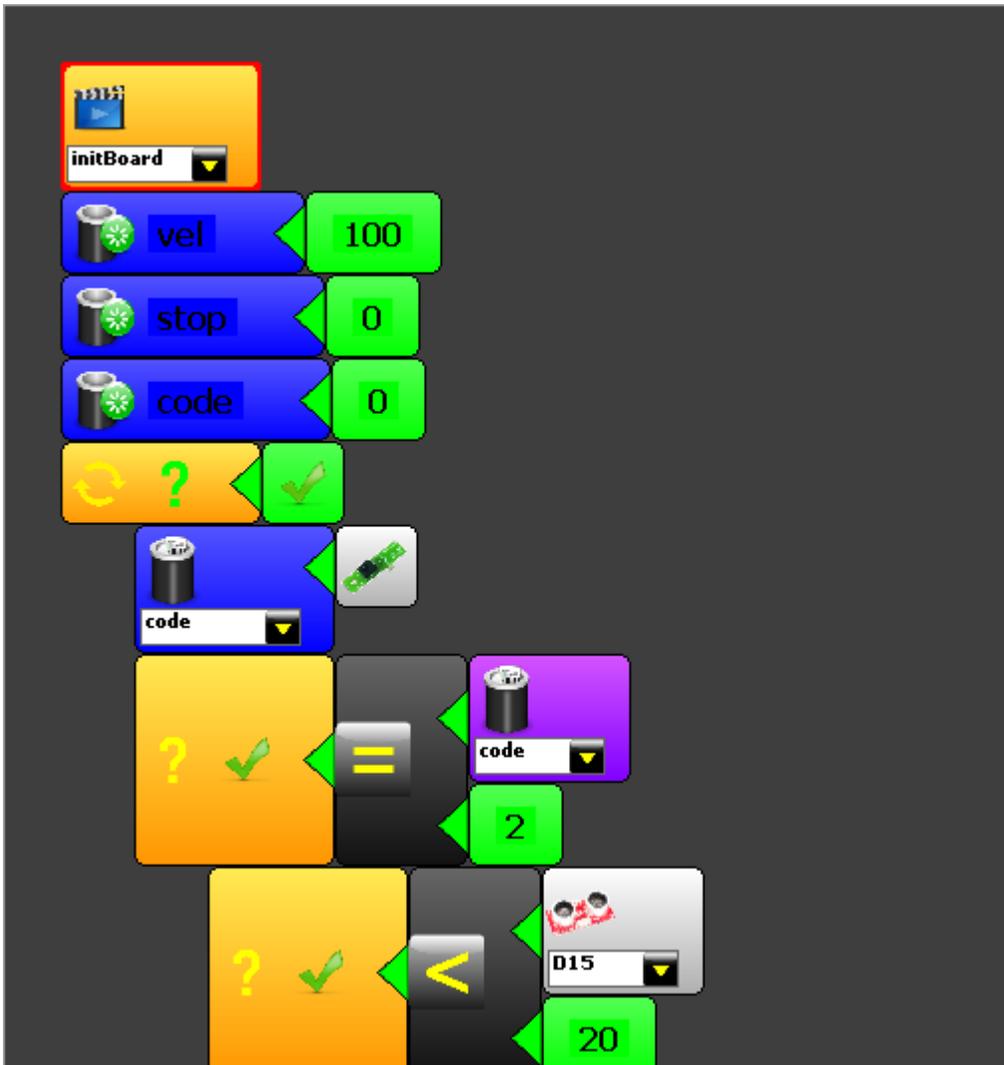
Además se encuentra incluido un sensor ultrasónico en la parte superior con el fin de que el robot se detenga si encuentra un obstáculo a 20 cm de distancia.

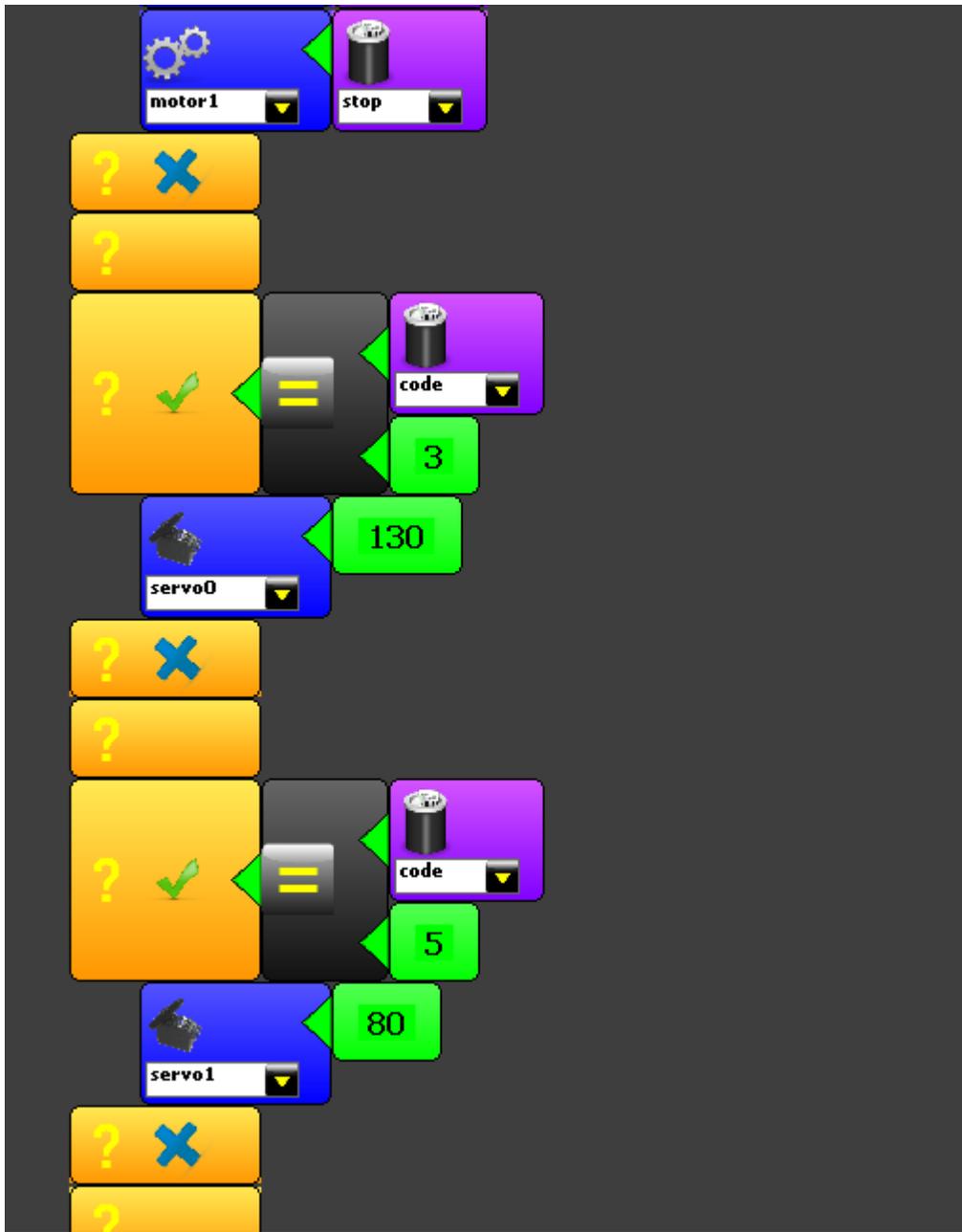


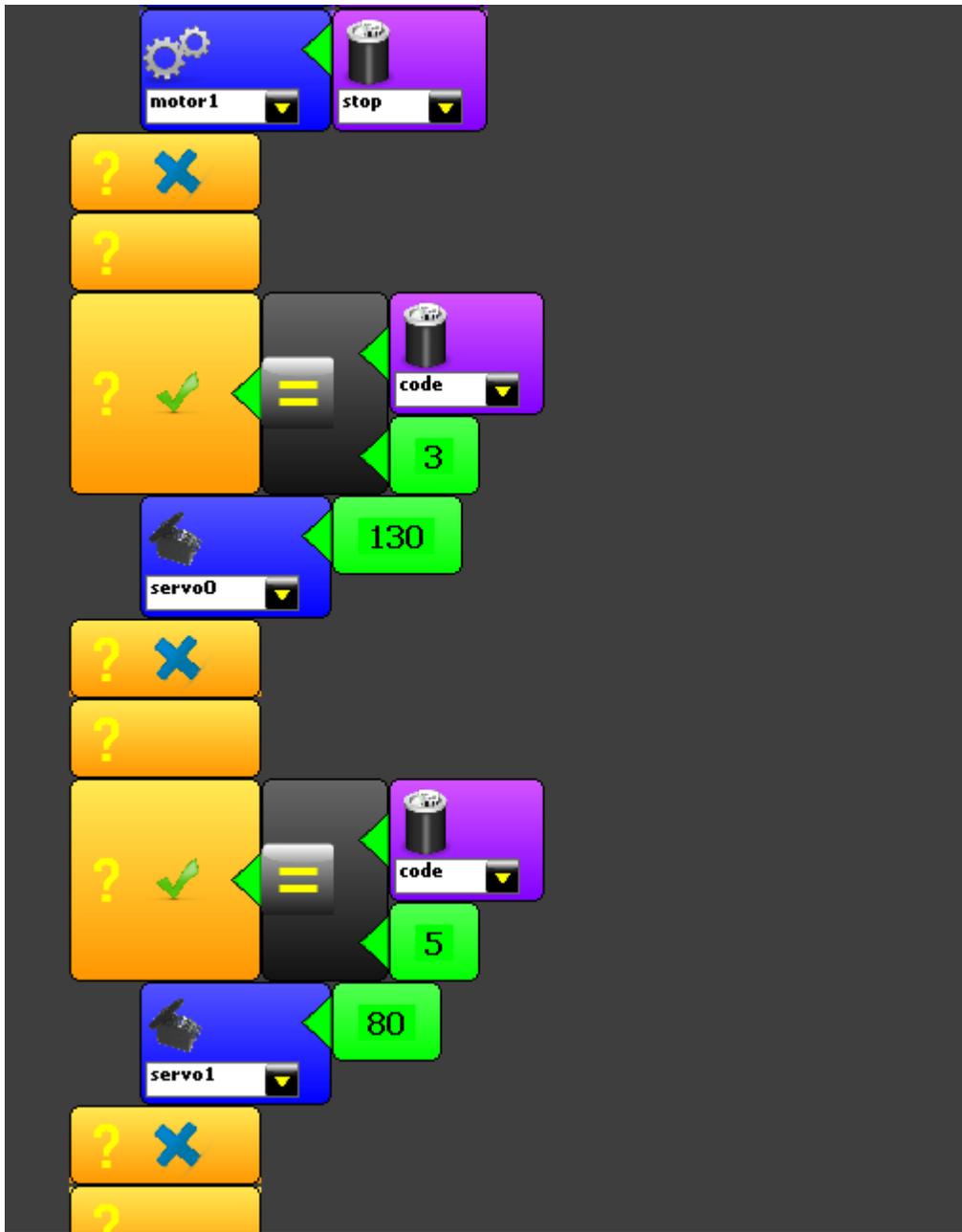
Ilustración 41: Control Remoto

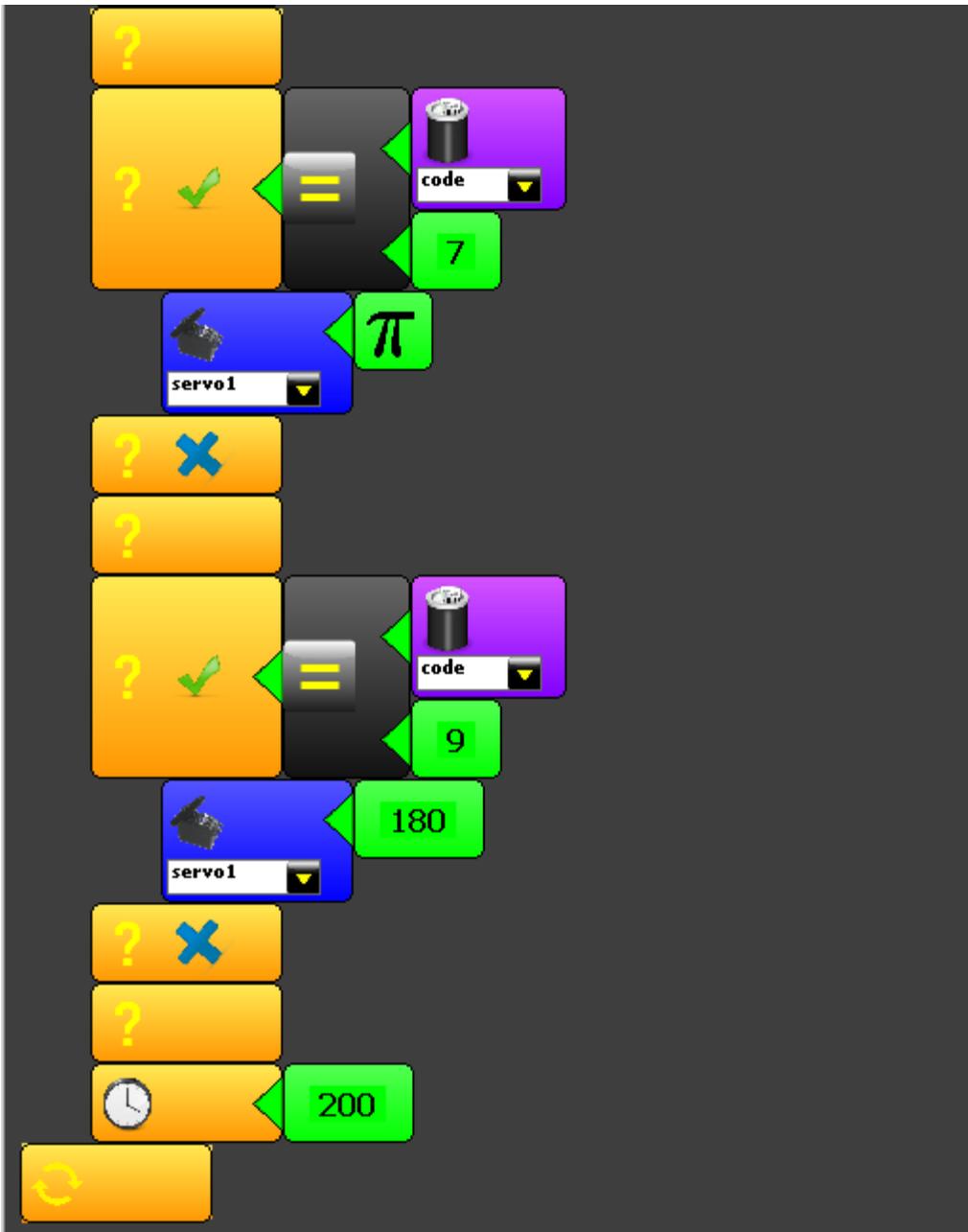
Código

El código en bloques, creado por el IDE Minibloq es el siguiente:









El código en lenguaje 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)(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()
```



{
}

Conclusión

El trabajar con Arduino y con Kit de rebotica de Robot Group (o con rebotica en general) en las escuelas secundarias, sería una excelente manera de llevar la práctica en las diferentes materias informáticas del secundario a un nuevo nivel ya que el contacto directo con el Hardware y la Programación de este, amplia los horizontes de las tareas que le asignamos a nuestros alumnos.

ANEXO 1

[Repositorio de fotos.](#)

Bibliografía

- Pruzzo di Pego, Vilma. (Editorial Espacio). (1998). EVALUACIÓN CURRICULAR: EVALUACIÓN PARA EL APRENDIZAJE. Buenos aires
- <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/>

