

¿Que es Icaro?





Resumen:

Breve introducción al proyecto Ícaro que usaremos a lo largo del curso. Un repaso rápido por las características del software, el hardware y el firmware de la plataforma.



Índice:

¿Qué es Icaro?.....	3
Introducción	3
Software.....	4
Tortúcaro.....	5
Módulo Apicaro (consola Python).....	5
Icaro Bloques.....	6
Hardware.....	7
Firmware.....	11



¿Qué es Ícaro?

ICARO es un proyecto colaborativo sin fines de lucro, que trata de acercar de manera sencilla las nociones básicas de la electrónica y la programación en un entorno robótico para utilizarlo dentro del aula como una herramienta de aprendizaje desde el **Software y Hardware Libre**.

Introducción

La robótica pedagógica, dado su carácter multidisciplinario, permite el abordaje de conocimientos variados como la electrónica, informática, física y matemática mediante la construcción de un juguete-objeto como puede ser un robot. El desarrollo de estos juguetes-objetos implica una experiencia que contribuye a expandir la creatividad y el pensamiento reflexivo y científico de los alumnos, en relación a la formulación de hipótesis, la experimentación y la elaboración de conclusiones, los cuales al enfrentarse a un “problema” dado, aprenden a experimentar, diseñar y resolver situaciones de carácter constructivista. En el proceso de “pensar el robot”, se generan las condiciones de apropiación del conocimiento por parte del alumno. Se trata de otorgar a los alumnos un rol activo en sus aprendizajes, colocándolos como diseñadores de sus propios proyectos y constructores de conocimientos. El uso de software libre en el ámbito escolar, permite tener control sobre las características del mismo, permitiendo adaptarlo a las necesidades concretas del ámbito escolar y las realidades socio-económicas de la institución, es neutro frente a fabricantes (el alumno no es un “potencial cliente”) y todo el material usado puede ponerse a disposición de otros docentes.¹

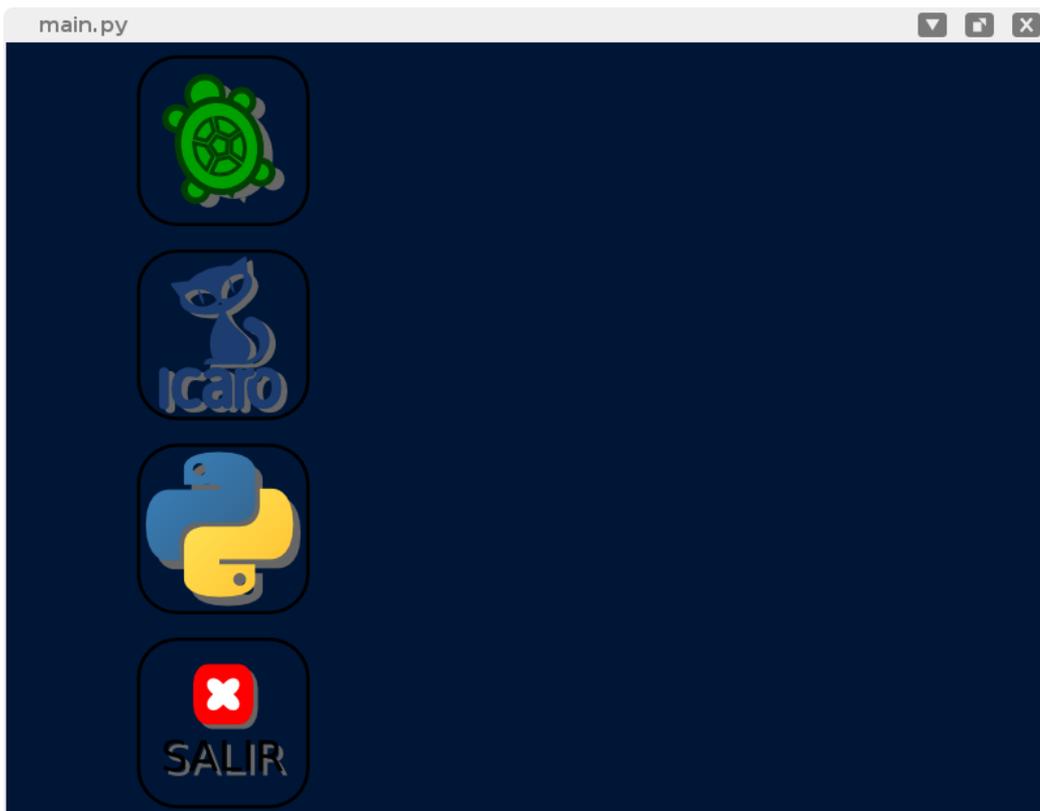
¹ Extraído de <http://roboticaro.org/de-que-se-trata-icaro/>.



Software.

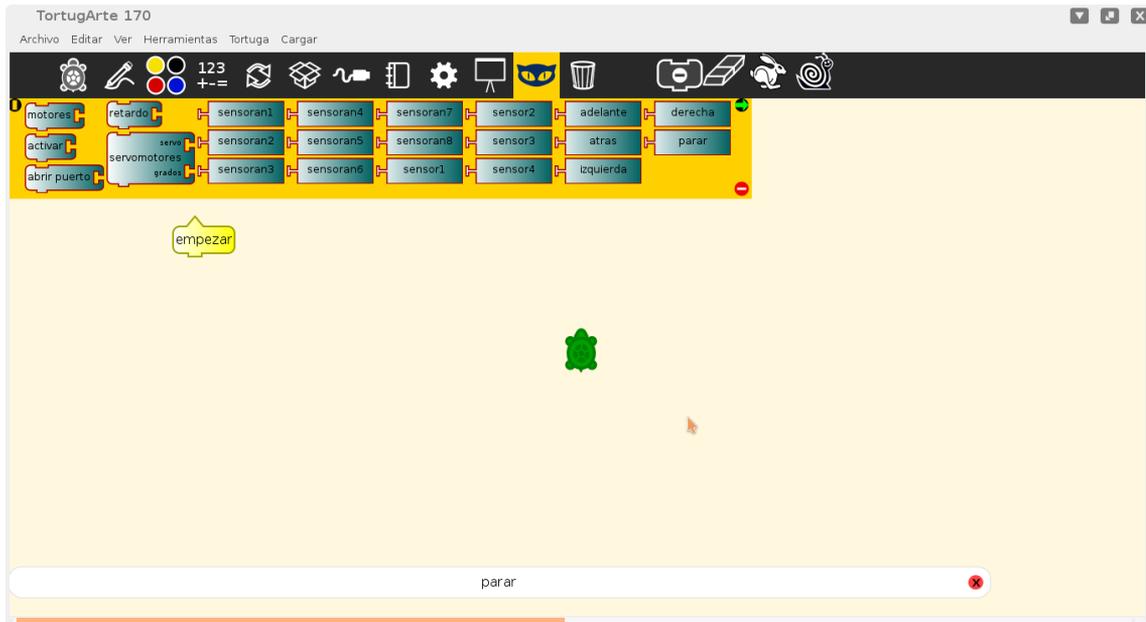
Icaro es una plataforma de software libre, en tanto su código fuente está abierto, es decir que podemos ver cómo está programado y además no responde a ninguna empresa o marca. Su desarrollo inicial fue realizado o por Valentin Basel, programador cordobés que liberó no sólo el software sino todos los planos de construcción del hardware, constituyendo así uno de los primeros proyectos de soft/hardware libre de la Argentina ya que su lanzamiento inicial se remonta al 2005. Desde el 2009 es parte de los repositorios oficiales de la distribución de Linux Fedora y desde su lanzamiento es parte también de Huayra Linux en 2012, la distribución oficial del Ministerio de Educación instalado en millones de netbooks en Argentina. Actualmente cuenta con una comunidad de colaboradores en toda latinoamérica, con experiencias concretas en Nicaragua, El Salvador, Panamá, Venezuela y está siendo usado ya en los cursos que dictó la coordinación TIC Tucumán del Ministerio de educación para las 36 escuelas Técnicas de la provincia. Se trata de una plataforma de programación que ofrece tres entornos de trabajo:

Tortúcaro, Icaro Bloques y Modulo Apicaro (Consola Python).





Tortúcaro



Se trata de TurtleArt+Icaro. Esto es, sumar al entorno de programación gráfica TurtleArt bloques que permiten enviar órdenes a una placa Icaro conectada a la computadora. Requiere cargarle un firmware a la placa que establece la comunicación en tiempo real con la computadora vía USB. De este modo es posible controlar motores y recibir datos de sensores desde la computadora. A su vez la placa depende de esta conexión para funcionar, no es independiente en este caso.

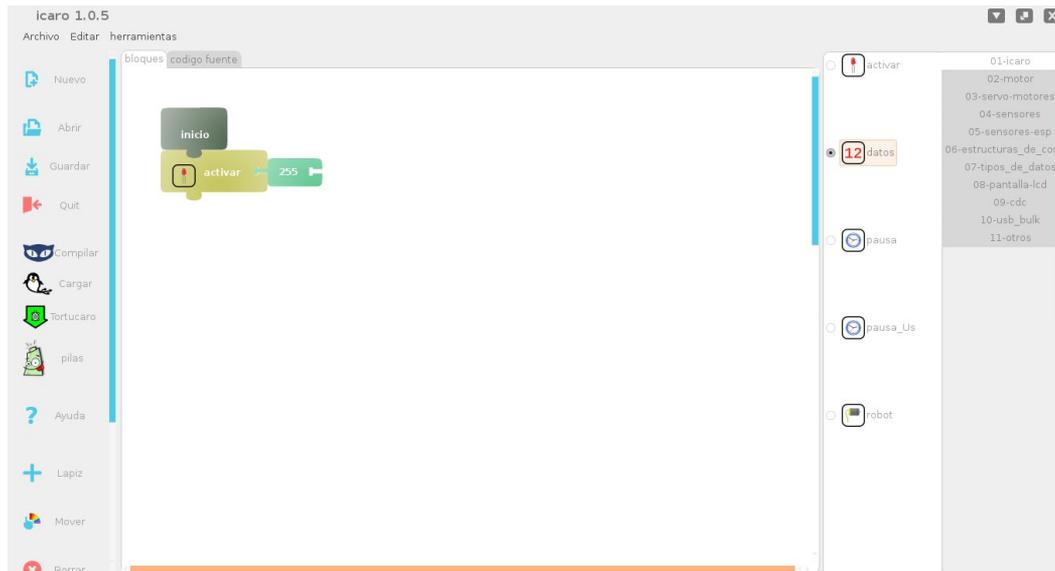
Modulo Apicaro (Consola Python).



Permite controlar la placa Icaro desde una consola Python, con toda la potencia de las librerías y recursos disponible en este lenguaje de programación.



Icaro Bloques



Este es el entorno que usaremos dentro de los talleres. Se trata de una interfaz de programación completa, que nos posibilita crear robots autónomos, que alimentados por una fuente de energía podrán trabajar en forma independiente a la computadora. Este entorno permite mediante bloques programar el microcontrolador de la placa Icaro (PIC18F4550, más detalles en la sección Hardware). Los programas son compilados o sea chequeados y traducidos a binario, para luego ser cargados en la placa via cable USB. La interfaz cuenta con 11 menús donde cada bloque corresponde a una entrada o salida de la placa, es decir cada bloque corresponde a comandos a un motor, a un led o nos permite saber qué valores se está recibiendo desde un sensor. Los bloques también representan estructuras de programación, operaciones matemáticas o asignación de variables. Para más referencias, consultar el manual de Icaro Bloques.

Icaro Bloques cuenta además con un entorno de programación en código (comandos escritos) que se va autogenerando cada vez que compilamos los bloques. Este código es básicamente lenguaje C, que guarda cierta compatibilidad con el lenguaje Arduino ya que es derivado del lenguaje Pinguino (Arduino para PIC).

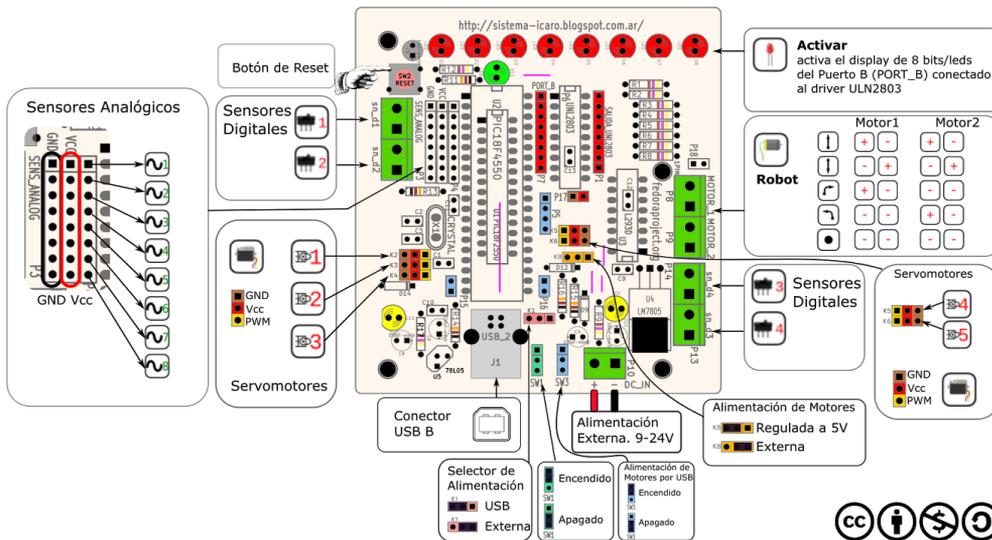


```
icaro 1.0.5
Archivo Editar herramientas
bloques codigo fuente
recargar
48: pinMode(ICR_D1G4_ECHO);
49: pinMode(ICR_L293_P1.OUTPUT);
50: pinMode(ICR_L293_P2.OUTPUT);
51: pinMode(ICR_L293_P3.OUTPUT);
52: pinMode(ICR_L293_P4.OUTPUT);
53: ServoAttach(ICR_SRV1);
54: ServoAttach(ICR_SRV2);
55: ServoAttach(ICR_SRV3);
56: ServoAttach(ICR_SRV4);
57: ServoAttach(ICR_SRV5);
58: #if defined( _LCD_ )
59: //Uso el PORTB para el LCD (usando los primeros 4bits y los
60: // otros dos para RS y E
61: lcd(4, 5, 0, 1, 2, 3, 0, 0, 0); // RS, E, D4 ~ D8
62: // Defino el numero de columnas y filas del LCD:
63: begin(8, 2);
64: home();
65: #endif
66:
67:
68: int sensordigital(int valor)
69: {
70: //funcion para cambiar el valor de los sens digitales (estan invertidos con respecto a la placa)*
71: int temp=0;
72: temp= digitalread(valor);
73: if (temp==0)
74: {
75: return 1;
76: }
77: else
78: {
79: return 0;
80: }
81: }
82:
83: *funciones*/
84: void loop()
85: {
86: PORTB=255;
87: }
```

Hardware.



Placa Icaro



La placa Icaro cuenta con 4 entradas de sensor digital (snd_1, snd_2, snd_3, snd_d4), 8 entradas para sensores analógicos, 2 salidas a motor de continua



bidireccionales (MOTOR_1, MOTOR_2), 5 salidas para servo-motores (K2, K3, K4, K5, K6) y la posibilidad de conectar simultáneamente 2 motores paso a paso. Cuenta con 8 leds de estado que permiten visualizar la salida de un puerto de 8 bits a modo de display binario. Puede ser alimentada via USB o con una fuente de alimentación una vez programada.

El elemento principal de la placa es su microcontrolador PIC18F4550, que cuenta con un puerto USB, lo cual facilita grandemente la comunicación con la placa Icaro, en tanto no requiere de ningún hardware extra para cargar los programas desde una computadora que solo cuente con estos puertos. Por cierto la gestión de la comunicación y la posibilidad de grabar los programas que hacemos con Icaro-Bloques se da gracias a que al PIC se le ha grabado previamente en su memoria tipo flash un firmware llamado bootloader (cargador de arranque) del cual hablaremos en el apartado siguiente.

Las entradas de sensor analógicos correspondientes al Conversor Analógico Digital (ADC) tienen una resolución de 10 bits, esto significa que puede traducir las tensiones que se aplican en su entrada (de 0 a 5V) en 1024 valores binarios diferentes.

Además del microcontrolador PIC la placa cuenta con 2 circuitos integrados auxiliares llamados drivers de potencia, el **L293D/B** que se encarga de proporcionar la potencia necesaria (36V y 0,6 A máximo) para mover en ambos los motores de corriente continua conectados a la salidas de motor (MOTOR_1 y MOTOR_2) y el **ULN2803** que se encarga de activar los leds y todo lo que conectemos a su salida (0,5 Ampere máximo), pudiendo ser como mencionamos anteriormente, dos motores paso a paso.

Así también la placa cuenta con un regulador, el LM7805 que puede proporcionar a los motores o cualquier salida un voltaje estable de 5V y hasta 1A de corriente. Un segundo regulador, el 78L05 se encarga de estabilizar también en 5V la tensión suministrada al PIC en caso de que se use alimentación externa.

PIC18F4550

Referencia rápida:

<http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC18F4550>

Hoja de datos:

<http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>

L293B/D

Referencia rápida:

<http://www.ti.com/product/l293>

Hoja de datos:

<http://users.ece.utexas.edu/~valvano/Datasheets/L293d.pdf>



ULN2803

Referencia rápida:

<http://www.ti.com/product/uln2803a>

Hoja de datos:

<http://www.ti.com/lit/ds/symlink/uln2803a.pdf>

LM7805

Referencia rápida:

<https://www.fairchildsemi.com/pf/Lm/LM7805.html>

Hoja de datos:

<https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf>

LM78L05

Referencia rápida:

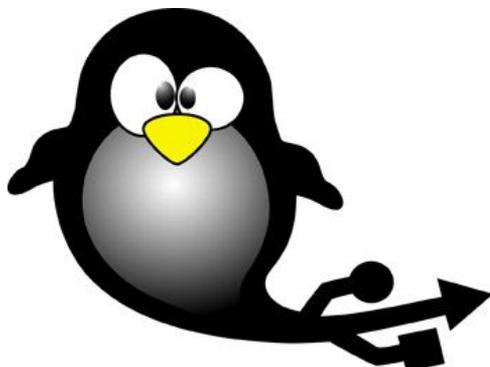
<http://www.ti.com/product/lm78l05>

Hoja de datos:

<http://www.ti.com/lit/ds/symlink/lm78l05.pdf>

Anexo a este documento se puede consultar la lista completa de componentes de la placa Icaro.

Firmware



El *firmware* es un bloque de instrucciones de máquina para propósitos específicos, grabado en un chip, normalmente de lectura/escritura (ROM,EEPROM,



flash, etc.), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.²

Finalmente como antes mencionamos, para poder gestionar la comunicación USB con Icaro Bloques y guardar los programas en su memoria flash es necesario cargar previamente en la misma memoria flash un Firmware llamado Bootloader o cargador de arranque. El bootloader de Icaro está basado en el firmware del proyecto francés de hardware y software libre Pinguino (<http://www.pinguino.cc/>) que intenta acercar el lenguaje y la programación de Arduino originalmente pensada para chips de marca Atmel a los microcontroladores PIC de las marca Microchip. Este cargador de arranque permite que el chip se “autoprograme”, esto es que no haga falta un hardware extra para grabar en la memoria flash los programas que uno realiza en el Icaro Bloques, solo hace falta compilarlos y luego “cargarlos” en la placa. Un hardware programador de PICs extra solo es necesario para grabar inicialmente el bootloader en la memoria Flash.

² <https://es.wikipedia.org/wiki/Firmware>.



Mateo Carabajal para la Coordinación TIC del
Ministerio de Educación de Tucumán.

