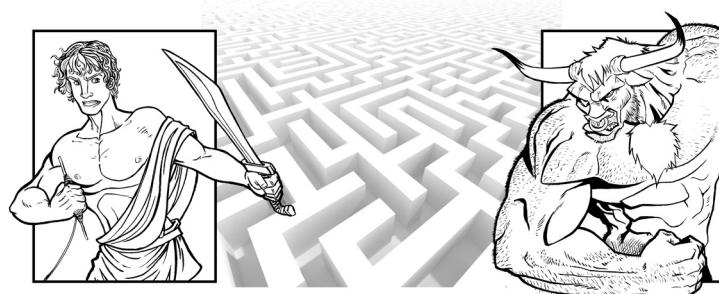




## Laboratorio de Microcomputadoras [66.09]

### Proyecto TCO



**Primer Cuatrimestre 2013**

Apellido	Nombre	Padrón	Mail
Carballeda	Ignacio L. J.	91646	carballeda.ignacio@gmail.com
Cerone	Sebastian	90259	sebascerone@gmail.com
Farace	Gisela	92457	gisela.farace@gmail.com

# Índice

<b>1. Motivación</b>	<b>3</b>
<b>2. Objetivo</b>	<b>3</b>
<b>3. Descripción general</b>	<b>4</b>
Robot . . . . .	4
Laberinto . . . . .	5
Algoritmo . . . . .	6
<b>4. Resumen de funcionamiento</b>	<b>7</b>
PWM y Motores . . . . .	7
Funcionamiento de los encoders . . . . .	7
Medición de distancia . . . . .	7
Identificación del color del piso . . . . .	8
Comunicación serie . . . . .	8
<b>5. Desarrollo del proyecto</b>	<b>9</b>
Decisiones de diseño . . . . .	9
Armado del robot . . . . .	9
Problemas hallados . . . . .	9
Experiencia competencia Megaman 2013 . . . . .	9
<b>6. Conclusiones</b>	<b>10</b>
<b>7. Listado de componentes</b>	<b>11</b>
Robot . . . . .	11
<b>8. Herramientas de software</b>	<b>12</b>
<b>9. Bibliografía</b>	<b>12</b>
<b>10. Apéndice</b>	<b>12</b>
Esquemáticos . . . . .	12
Código fuente (*) . . . . .	14

## 1. Motivación

Este desarrollo surge como un proyecto extracurricular del *Club de Robótica* de la *Facultad de Ingeniería de la Universidad de Buenos Aires* a realizarse dentro del ámbito del *Laboratorio Abierto (LABI)*.

Desde el año 2011, el *Club de Robótica* realiza una competencia anual de robótica. En un primer momento, empezó con una carrera de robots velocistas; que consiste en que un robot realice por sus propios medios, en el menor tiempo posible, un recorrido delimitado por una línea blanca sobre una superficie negra. Al año siguiente, se incorporó la competencia de "Mini-sumo"; en donde dos robots se enfrentan, cual luchadores de sumo, y deben empujar al oponente fuera del "tatami". Finalmente, este año, se incorporó la categoría "Laberinto", en donde un robot, en el menor tiempo posible, debe encontrar la salida de un laberinto.

Es por esta razón, que se optó por realizar un proyecto que sirva para aprender los temas de la asignatura *Laboratorio de microcomputadores* y pueda representar al *Club de Robótica* en la competencia anual.

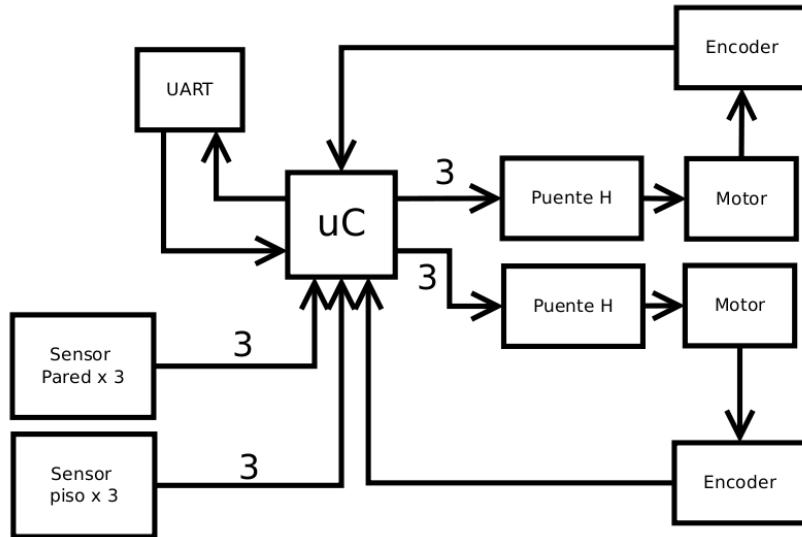
Cabe aclarar, la elección del nombre *TCO*, está inspirado en el personaje de la mitología griega *Teseo*, quien logró salir de un laberinto luego de derrotar al Minotauro.

## 2. Objetivo

Por las razones planteadas anteriormente, el objetivo de este trabajo es el diseño, fabricación y programación de un robot capaz de salir de un laberinto por sus propios medios.

### 3. Descripción general

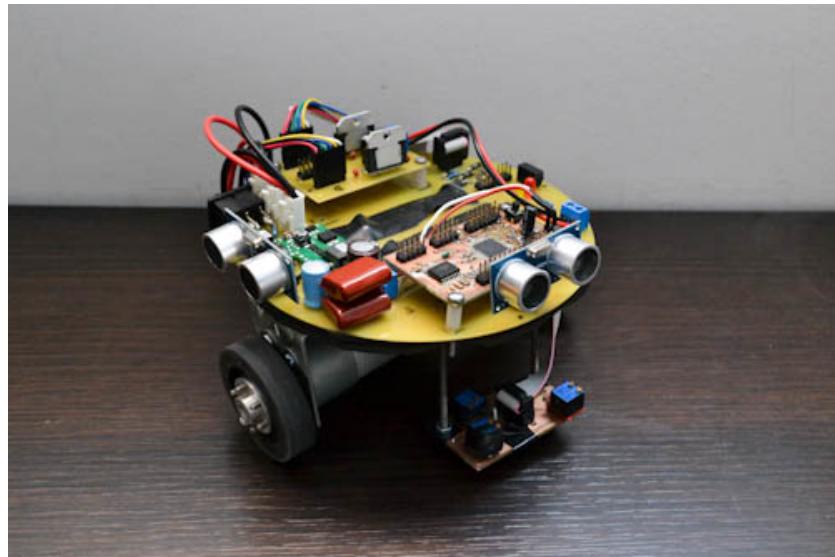
#### Robot



**Figura 1:** Diagrama de bloques del robot.

Para llevar a cabo este proyecto se fabricó un robot. El mismo está compuesto de las siguientes partes:

- Microcontrolador: Se eligió un *ATMEGA16* de *AVR*. Este microcontrolador, muy sencillo, tiene la cantidad adecuada de periféricos necesarios para la realización de este proyecto.
- Motores: Se optó por dos motores de la marca *pololu*. Éstos tienen encoders integrados, que facilitan un mejor control del desplazamiento.
- Puente H LMD18200: Este puente H simple, maneja cargas de hasta 3 Amperes y posee un flag de advertencia de alta temperatura.
- Sensores HC - SR04: Estos sensores de distancia por ultrasonido, sencillos de utilizar, y muy estables, detectan distancia en un rango de 2 cm a 450 cm, con un margen de error de 0,3 cm.
- Sensores CNY70: Estos sensores ópticos, se utilizan para diferenciar el color del piso entre blanco o negro. En este caso, se utilizarán para detectar si el robot llegó al final del laberinto.
- Baterías y fuente de alimentación: El robót cuenta con 3 baterías de Li-ion, 2 de 7,2 v y 1 de 3,7 v. Junto con dos fuentes DC/DC de *texas* proporcionando 12 v y 5 v.

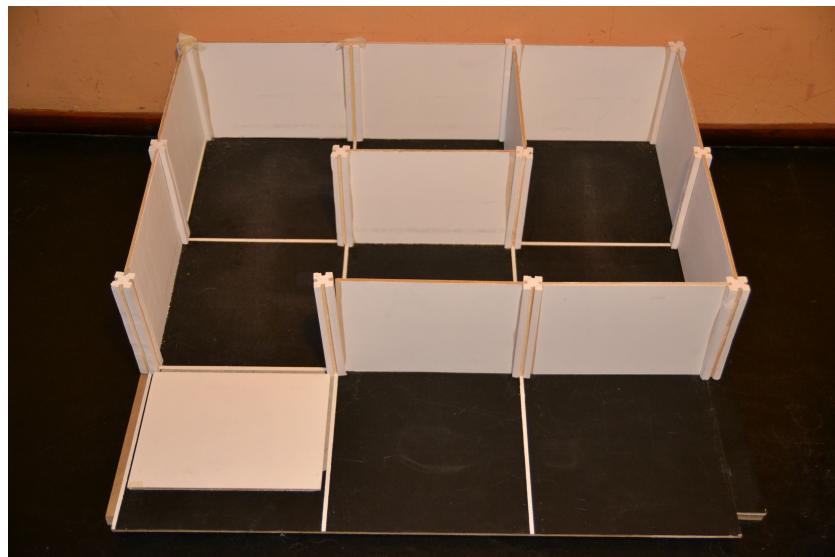


**Figura 2:** TCO

## Laberinto

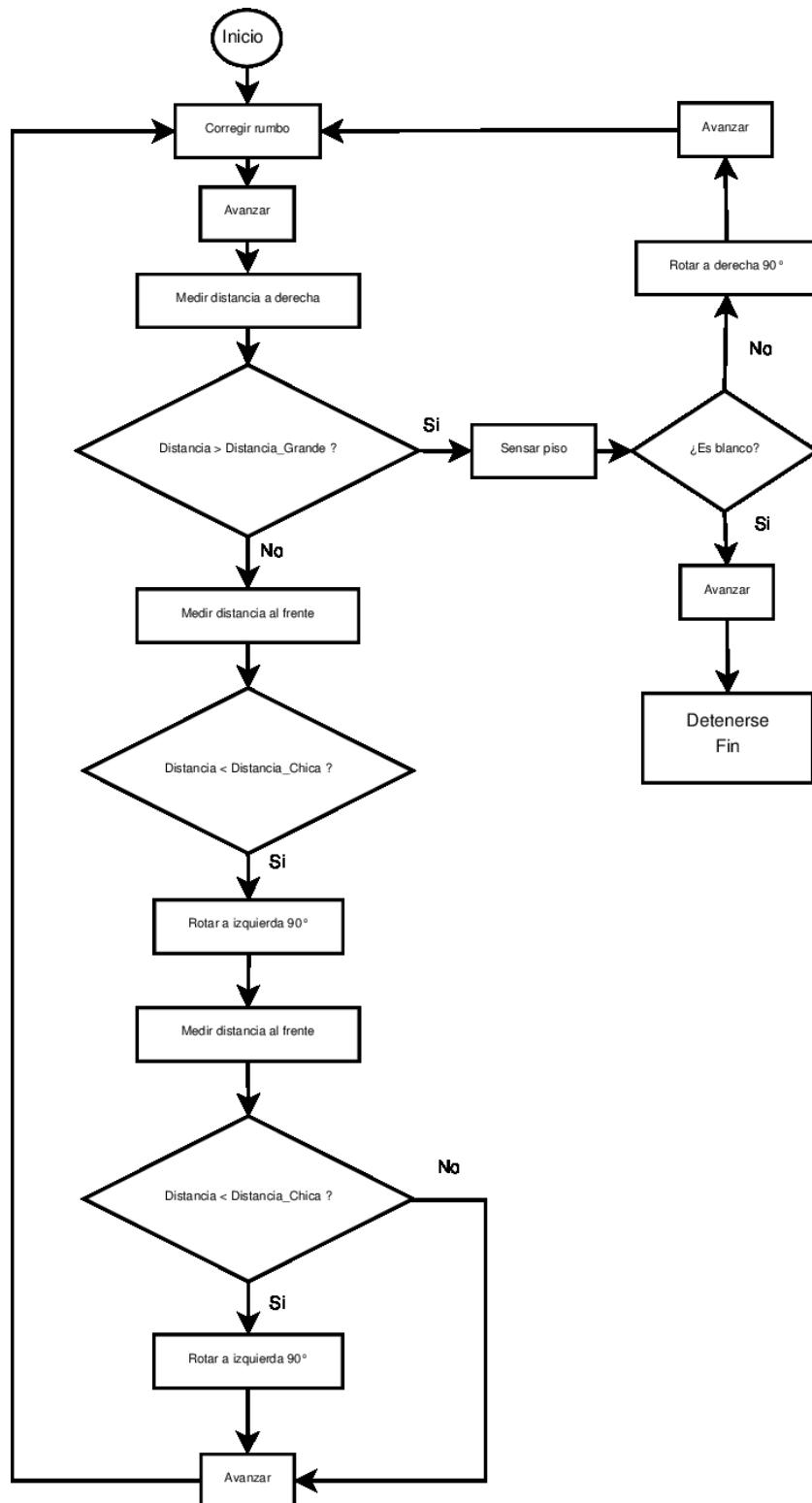
El laberinto se fabricó como un módulo modelizable como una matriz perteneciente a  $\mathbb{R}^{3x3}$ , donde sus paredes internas, definen *celdas* que se pueden interpretar como posiciones dentro de la matriz.

El mismo se construyó con paredes color blanco y piso color negro, dividiendo cada celda por una línea blanca, siendo que sobre este color el rebote de luz infrarroja resulta óptimo para los sensores del piso, y facilita la detección entre una celda y otra.



**Figura 3:** Laberinto

## Algoritmo



**Figura 4:** Laberinto

## 4. Resumen de funcionamiento

### PWM y Motores

### Funcionamiento de los encoders

### Medición de distancia

Para medir la distancia del robot a las paredes se utilizaron tres módulos *HC-SR04*, los mismos funcionan por ultrasonido.

Se eligieron estos sensores por varias razones:

- Su rango de 2 centímetros a 4 metros.
- Su resolución de 3 milímetros.
- Simplicidad y costo reducido.

Se desconoce la circuitería interna del mismo, por lo que será tratado como a una caja negra con el cual se puede interactuar mediante sus pines de Entrada/Salida.

El HC-SR04 además de sus pines para alimentación (VCC y GND), tiene otros dos pines: Trigger y Echo. Para obtener una medición, el módulo sólo requiere un pulso de  $10\mu S$  en el pin Trigger, luego el módulo pone en alto el pin de Echo con un tiempo proporcional a la distancia medida.

En el código fuente de este proyecto, se configuró una interrupción para que se active con cada desbordamiento del *timer/contador1*. Esta interrupción en cada iteración incrementa en 1 la variable **interrupciones\_timer\_1**. También se configuran los pines Echo y Trigger como entrada y salida respectivamente.

Para obtener una medición, primero se pone en alto el pin de Trigger por un tiempo mayor a  $10\mu S$ , luego se espera con un ciclo “while” a obtener un “uno” en pin de Echo; acto seguido se toma el valor actual del contador/timer1 y se guarda en la variable **cuentas\_inicial**. A continuación, se espera a que el pin de Echo vuelva a cero con otro ciclo “while” y se guarda el valor del registro del contador/timer1 en otra variable llamada **cuentas\_final**.

Luego, teniendo en cuenta que el timer1 fué configurado para que desborde a las 256 cuentas; para obtener el número total de cuentas se utiliza el siguiente código:

```
1 cuentas = (256 - cuentas_inicial) + 256 * interrupciones_timer_1 +
    cuentas_final;
```

Finalmente se realiza una conversión para relacionar las cuentas con una distancia aproximada.

## Identificación del color del piso

Con el objetivo de detectar la celda blanca, se utilizaron tres sensores detectores de color en la parte baja del mismo, que le indican al robot que salió correctamente del laberinto. Los sensores utilizados son los *CNY70*, compuestos por un emisor (led infrarrojo) y un receptor (transistor TBJ NPN).

Estos sensores pueden ser encendidos y apagados a voluntad por el microcontrolador.

Cuando el robot está situado en una celda blanca (salida), la luz emitida por el led del sensor se refleja en la superficie y excita la base del transistor, permitiendo de esta forma un flujo de corriente de emisor a colector.

Se ubicó un resistor entre el emisor del transistor y la referencia del circuito (GND), para detectar los cambios de tensión.

En la hoja de datos del *ATMEGA16* se especifica que la máxima tensión admisible para un cero lógico (con el pin configurado como entrada), es de  $0,2 \times VCC$ , en nuestro caso  $VCC = 5V \Rightarrow V_{IL_{max}} = 1V$ . Por lo tanto una tensión que no supere 1 Volt, será considerada como un cero lógico. Por otro lado,  $V_{IH_{min}} = 0,6 \times VCC = 0,6 \times 5V = 3V$  es la mínima tensión para detectar un uno lógico en un pin configurado como entrada.

Cuando un sensor es situado sobre una celda blanca, la tensión en su emisor alcanza los 4.5 V, valor considerado por el microcontrolador como un "uno" lógico. En cambio, ubicado sobre una celda negra, la tensión medida es aproximadamente cero, ya que la luz no es reflejada en esta superficie y esto es interpretado como cero lógico.

El microcontrolador enciende los sensores durante un tiempo muy breve cada vez que el robot decide girar a su derecha, de esta forma se evita que las líneas blancas que dividen las celdas sean interpretadas como la salida del laberinto.

## Comunicación serie

El presente proyecto se puede conectar a un PC, por medio de un puerto RS232. Esta función es utilizada para *debugging*.

Se configuró la *UART* para enviar y recibir datos, y se configuró una interrupción, que se activa cuando se recibe un dato por el puerto serie.

## **5. Desarrollo del proyecto**

**Decisiones de diseño**

**Armado del robot**

**Problemas hallados**

**Experiencia competencia Megaman 2013**

## 6. Conclusiones

## 7. Listado de componentes

### Robot

Referencia	Descripción	Cantidad	Precio unitario	Total
C3, C9, C11	100 nF	3		
C25	100 uF	1		
C14, C15, C18, C19	10 n	4		
C4, C5, C6, C7, C8, C13, C17	1 uF	7		
C1, C12	12 p	2		
C12, C16	220 uF	2		
C20, C21	2,2 uF	2		
C23	330 uF	1		
R1	10 k	1		
R11	21 k	1		
R10	732	1		
R2, R3, R4, R5, R6, R7	R	6		
X1	8 MHz	1		
IC1	ATMEGA16-A	1		
U1	MAX232	1		
U2	PTN78000W	1		
U3	PTN78020W	1		
Puente1, Puente2	LMD18200	2		
L1	Inductor	1		
Conecotor 2x1	Conecotor 2x1	7		
Conecotor 3x2	Conecotor 3x2	1		
Conectores	Conectores	9		
SW1, SW2,	Pulsador NA	2		
SW3	Llave dos vías	1		
Q1	NPN	1		

## 8. Herramientas de software

- KiCAD (Build: 20100406 SVN-R2508)-final: Diseño de esquemáticos y ruteo de placas
- CuteCom (v0.22.0): Captura de datos por puerto serie (S.O. Linux - Ubuntu)
- L<sup>A</sup>T<sub>E</sub>XLenguaje para paginado de este informe

## 9. Bibliografía

- I.Scott Mackenzie, "The 8051 Microcontroller",Prentice Hall, 1995
- Datasheet del Microcontrolador ATMEGA16

## 10. Apéndice

**Esquemáticos (\*)**

**Código fuente (\*)**

\*Ver páginas adjuntas a este informe.