

**INSTITUTO TECNOLÓGICO Y DE  
ESTUDIOS SUPERIORES DE MONTERREY  
CAMPUS ESTADO DE MEXICO**

Programación Robot Didáctico del Instituto v1.0

Manual Técnico

## Introducción

El propósito de este manual es facilitar la reutilización del código y archivos documentados para la primera versión del proyecto de programación del robot didáctico del instituto.

Toda la información necesaria podrá ser encontrada dentro de la siguiente página:

<http://rcse-teccem.googlecode.com>

Nota: Es altamente recomendable que las personas encargadas de la siguiente versión consulten y lean en su totalidad este documento.

## Contenido

Dentro de la página se podrán encontrar los siguientes elementos:

- Manual de Usuario
- Manual Técnico
- Código del IDE
- Código de la aplicación móvil
- Librerías necesarias
- Código de comunicación con el robot
- Ficha técnica del robot
- Cliente y servidor RFCOMM
- Cliente y servidor L2CAP
- Plan de comunicación Bluetooth

Todos estos componentes son parte esencial para la comprensión rápida y completa de la primera versión. Todo el código está documentado con el propósito de facilitar al desarrollador la asimilación de éste. A su vez, la página cuenta con ligas importantes a considerar para la comprensión de la comunicación Bluetooth y las librerías de PyBluez.

## Requerimientos

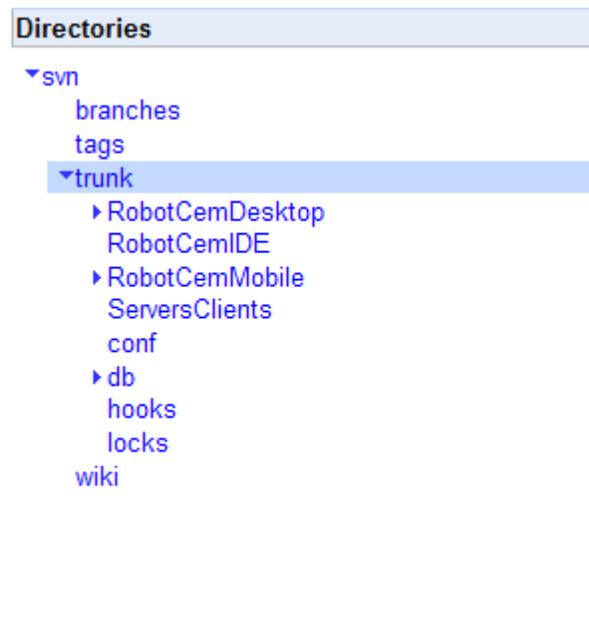
- Python 2.5 en adelante (Excluyendo Python 3 hasta que PyBluez lo soporte).
- PyBluez compatible con la versión de Python que haya sido instalada
- Subversion (Recomendación para Windows: TortoiseSVN)
- JDE Java 2.6 en adelante.
- Driver de Windows o Broadcom para Bluetooth (En el caso de Windows XP en adelante viene integrado o se deberá utilizar la versión disponible del Stack de Windows, si se utiliza alguna versión de Linux se tendrá que agregar manualmente).

## Elementos del código fuente

El código fuente se encuentra dentro de la página anteriormente mencionada. Para poder acceder a él se pueden utilizar dos métodos. El primero es utilizando la opción *Browse* del google code. Por otro lado, se podrá utilizar la herramienta Subversion considerando como semilla la liga que apunta al trunk (<http://rcse-teccem.googlecode.com/svn/trunk/>).

Para mayor referencia de uso y versiones disponibles de dicha herramienta, se puede consultar la siguiente liga: <http://subversion.apache.org/packages.html>.

El árbol está formado de la siguiente manera:



### ❖ **svn/trunk**

La rama trunk es donde se encuentran las 3 principales composiciones del proyecto.

#### ❖ **svn/trunk/RobotCemDesktop (Librerías de comunicación)**

- RobotEmulator.py: Servidor básico para la prueba de conexión con PyBluez.
- robotcem.py: Este archivo contiene las librerías de comunicación entre el robot y el IDE. Es esencial para el funcionamiento del programa.

#### ❖ **svn/trunk/RobotCemIDE (IDE)**

Dentro de este archivo se encuentra el código de la interfaz gráfica y el puente de comunicación entre las librerías y el GUI.

#### ❖ **svn/trunk/RobotCemMobile (Aplicación móvil)**

Esta rama contiene los archivos de java utilizados para la aplicación móvil. El listado de archivos son los siguientes:

- Controller.java
- BluetoothDeviceScanner.java
- Views: MainView.java, MovementView.java, SensorView.java, StartUpView.java.

Cada uno de ellos cuenta con una explicación de sus métodos y clases dentro de los mismos.

Nota: Para poder utilizar y probar esta parte del proyecto se necesita un celular que soporte Java. Se pueden utilizar emuladores pero se debe de tener en cuenta que estos deben de soportar los mismos servicios Bluetooth que los celulares, en particular el servicio de Serial(RFCOMM).

#### ❖ svn/ServersClients (Pruebas de comunicación)

Estos códigos pueden ser utilizados para pruebas de comunicación tanto por RFCOMM como L2CAP.

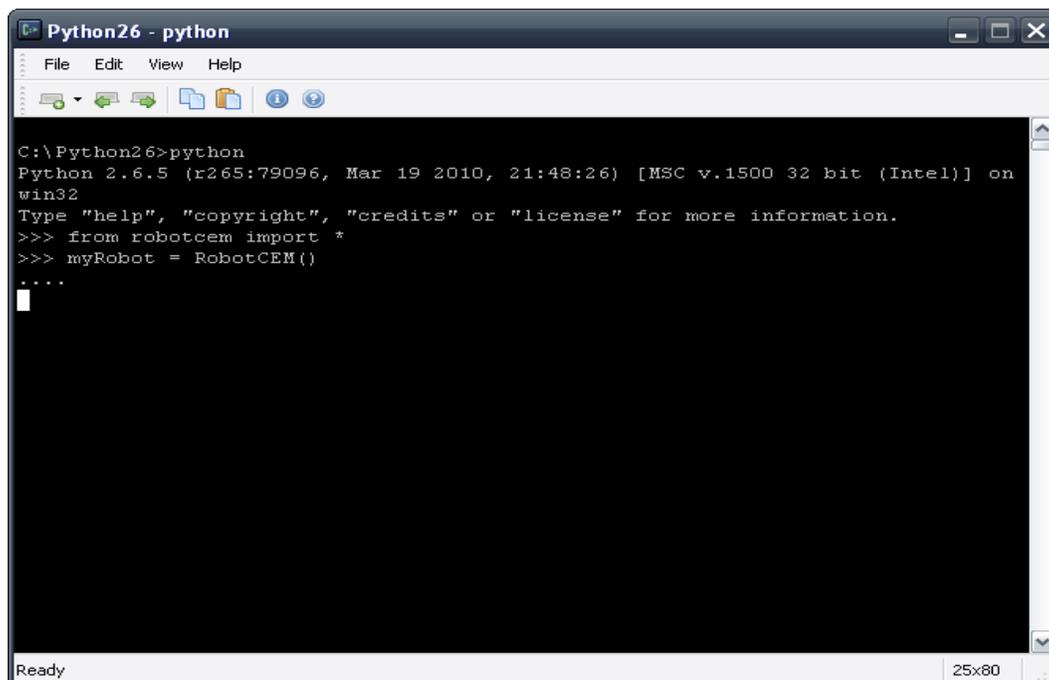
Las demás ramas son parte de la configuración de subversion y por lo tanto no es necesario atender una explicación para éstas.

## Elementos técnicos de las aplicaciones

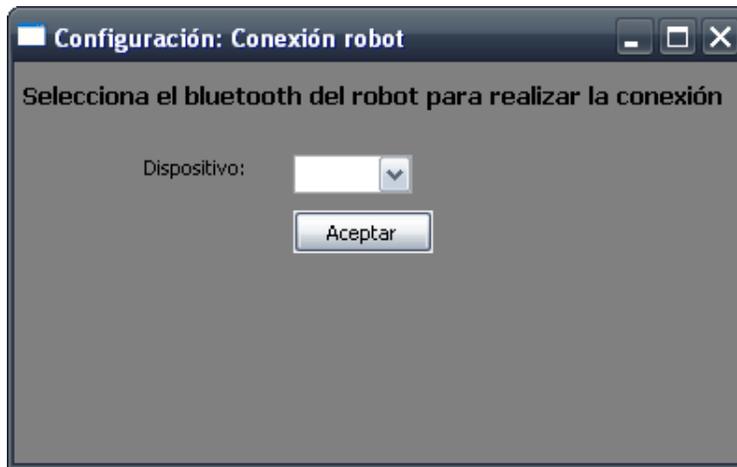
- Librerías de comunicación (RobotCEM)

Estas librerías se encuentran realizadas con la versión de Python 2.6 y utilizan PyBluez para el manejo de las conexiones de bluetooth y WxPython para proveer del ambiente gráfico para la selección del dispositivo.

Al crear una instancia de RobotCEM, en seguida se desplegará una pantalla la cual permita escoger el dispositivo bluetooth a conectar.



```
Python26 - python
File Edit View Help
C:\Python26>python
Python 2.6.5 (r265:79096, Mar 19 2010, 21:48:26) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from robotcem import *
>>> myRobot = RobotCEM()
.....
Ready 25x80
```



Para comprender lo ocurrido en la pantalla anterior, internamente la librería realizará lo siguiente:

→ Buscará los dispositivos cercanos, desplegando el nombre y dirección en forma de tupla de cada dispositivo. Lo anterior generará que se despliegue la lista en la ventana previamente mostrada con los nombres que se tienen asignado en un dispositivo bluetooth (En un futuro, esta acción puede ser omitida al saber las direcciones internas que los robots tengan.

```
bluetooth.discover_devices(lookup_names=True)
```

#### Ejemplo:

```
>>> bluetooth.discover_devices(lookup_names=True)
[('00:05:C9:D2:0A:9A', ''), ('00:0A:3A:2D:7C:64', 'TEC10 '),
 ('00:16:41:C9:C6:9E', 'FALCONS_CORP'), ('00:22:65:1F:C9:81', 'Nozomi'),
 ('00:23:39:F0:6D:28', ''), ('00:24:2C:B8:54:1F', 'SANTIMOMA'),
 ('00:25:CF:86:F8:06', 'KeReN'), ('00:26:08:BF:F7:A7', 'MacBook Pro de
Gustavo Alonso Roman Due\xflas')]
```

→ Buscará los servicios de la dirección escogida por el usuario, tratando de encontrar el puerto que se encuentra asignado para la comunicación. Por el momento el robot tiene solamente un puerto y es el 1 de serial.

```
find_service(address='00:0A:3A:2D:7C:64')
```

#### Ejemplo:

```
>>> services = find_service(address='00:0A:3A:2D:7C:64')
>>> for svc in services:
...     print svc["name"]
...     print svc["port"]
...
Serial Port
1
```

→ Al obtener el puerto, se procederá a establecer la conexión con el robot usando el siguiente comando.

```
>>> socket.connect(('00:0A:3A:2D:7C:64',1))
```

Una vez realizada la conexión, la pantalla se oculta y ya es posible mandar órdenes al robot (Internamente se comunica a través de los protocolos de comunicación del robot, presentados en los anexos de este documento). A continuación se presenta información de las librerías.

- Clase principal de la librería

Classes	
	<a href="#">RobotCEM</a> Class for robot control

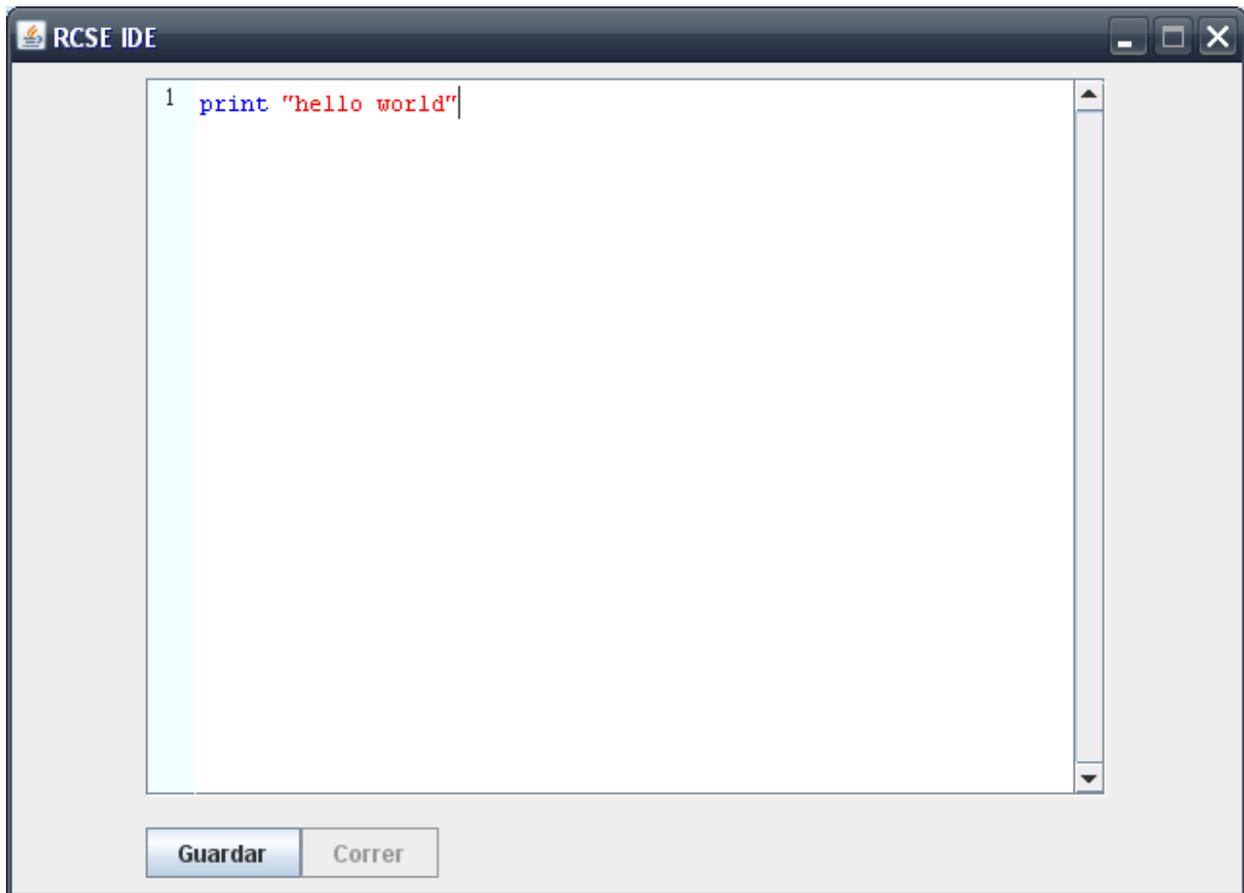
- Métodos de la clase principal

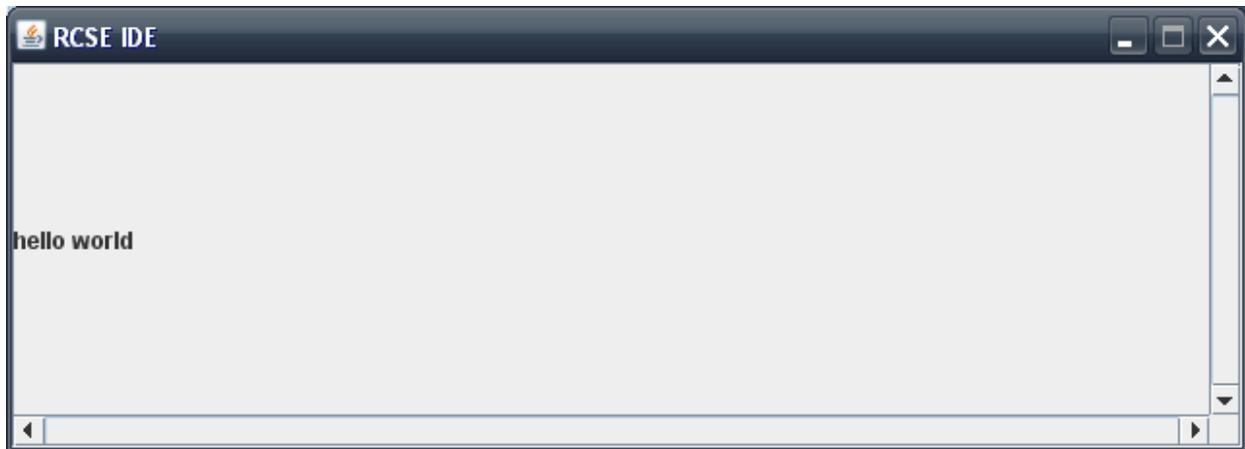
Instance Methods	
	init_graphics(self, devicesList) Starts up the main window for device selection
	select_device(self, event) Device selection event manager
	<u>init</u> (self) RobotCEM constructor
	waitForRobotAction(self) Waits for robot action
	moveAction(self, velocity, command) Sends a moving action to the robot
	moveForward(self, velocity=None) Sends a move forward action
	moveLeft(self, velocity=None) Sends a move left action
	moveRight(self, velocity=None) Sends a move right action
	moveBackward(self, velocity=None) Sends a move backward action
	stop(self) Sends a halt to the current action
	setVelocity(self, value) Sets the robot velocity
	getLuminosity(self) Requests for luminosity information
	getPosition(self) Requests for position information
	getTemperature(self) Requests for temperature information
	getObjectProximity(self)

	Requests for object proximity information
getWheel1Velocity(self)	Requests for wheel 1 velocity information
getWheel2Velocity(self)	Requests for wheel 2 velocity information
turnOnSpeaker(self)	Turns on the speaker
turnOnHighLuminosityLight(self)	Turns on the high luminosity light
turnOnLight(self, lightNumber)	Turns on the light
turnOnLineFollower(self, sensorNumber)	Turns on the line follower

- IDE

Debido a cuestiones de portabilidad, se decidió realizarlo en Java 1.6. Tiene las características de presentar syntax highlight para la parte de codificación del lenguaje python, desplegar los números de línea y permite fácilmente obtener las salidas del código.





Recalcando la parte de syntax highlight, ésta se encuentra realizada por la comparación de cadenas en el documento obtenido. Utilizando el hashSet de palabras claves que se encuentran en el archivo SyntaxDocument.java

```
keywords = new HashSet<String>();
keywords.add( "and" );
keywords.add( "as" );
keywords.add( "assert" );
keywords.add( "break" );
keywords.add( "class" );
keywords.add( "continue" );
keywords.add( "def" );
keywords.add( "del" );
keywords.add( "elif" );
keywords.add( "else" );
keywords.add( "except" );
keywords.add( "exec" );
keywords.add( "finally" );
keywords.add( "for" );
keywords.add( "from" );
keywords.add( "global" );
keywords.add( "if" );
keywords.add( "import" );
keywords.add( "in" );
keywords.add( "is" );
keywords.add( "lambda" );
keywords.add( "not" );
keywords.add( "or" );
keywords.add( "pass" );
keywords.add( "print" );
keywords.add( "raise" );
keywords.add( "return" );
keywords.add( "try" );
keywords.add( "while" );
keywords.add( "with" );
keywords.add( "yield" );
keywords.add( "None" );
keywords.add( "True" );
keywords.add( "False" );
```

Con respecto a la obtención de las salidas en otro lenguaje diferente a Java (En este caso Python), se realiza a través de una llamada al sistema, la cual manda a llamar a python.exe con el código generado como argumento. De la siguiente manera:

```
Process p = r.exec("python.exe " + "'" + file + "'");
```

Es importante recalcar que para que lo anterior funcione, se debe de tener python bien configurado desde el path.

Las características del IDE, permiten abrir la posibilidad de integrar fácilmente cualquier lenguaje de programación.

Importante: A poco tiempo de la realización de la presentación del IDE, se observó que el procesamiento de la salida obtenida, no desplegaba el primer carácter de la salida. Es importante corregirlo.

- Aplicación móvil

Con respecto a la aplicación móvil, se realizó algo similar a las librerías del escaneo de bluetooth, sin embargo presenta la deficiencia que bajo ambientes saturados de dispositivos, el escaneo puede alcanzar tiempos de hasta 15 min. En una habitación, con 4 bluetooth el escaneo tiene una duración aproximada de 1-2 min. Pese a lo anterior, se presentó un problema en relación al robot al momento de realizar la programación para la conexión, la mayoría de los dispositivos móviles, pedían establecer una conexión por intercambio de llaves, sin embargo, el robot no poseía seguridad alguna y no tenía forma de enviar el password, por lo que no fue posible establecer la conexión de manera correcta.

## Aspectos futuros a considerar

Debido a las limitaciones físicas del robot, la primera versión se vio afectada considerablemente por la falta de varios componentes dentro de la composición electrónica del producto. A continuación se mencionarán distintos aspectos a considerar para futuras versiones.

- Memoria física

El robot cuenta con una memoria limitada, la programación de éste es básica y esto afecta en el desarrollo y eficiencia por parte del programa cliente. Es sumamente necesario agregar mayor memoria para que se pueda implementar la utilización de ACKs (Acknowledgements) de las funciones del robot. Esto para evitar la sobresaturación de comandos y pérdida de algunos por el búfer limitado del robot.

- Seguridad

Se estuvieron realizando varios estudios con respecto a la seguridad en la comunicación del robot y la computadora. Como resultado, se pudo lograr el siguiente análisis de vulnerabilidades.

Vulnerabilidad	Definición	Mitigación
Overflow en el número de asociaciones	El atacante puede enviar un número inusual de peticiones para asociarse al dispositivo,	Generar alguna restricción en el programa del robot para restringir el número de

	con lo cual puede utilizar todos los puertos disponibles y negar cualquier tipo de comunicación posterior	conexiones permitidas
DoS por manejo inadecuado de sesiones	Cualquier usuario puede interrumpir la comunicación ya establecida con el robot al intentar establecer una nueva sesión	Integrar un esquema de separación de recursos para cada sesión establecida y/o restringir el número de sesiones establecidas a una sola para control del robot
Overflow de memoria	El atacante envía un número suficiente de peticiones como para saturar la memoria del robot	Restringir el número de direcciones a cierto tipo de dispositivos, que cuentan con el código de seguridad adecuado y cuya dirección física ha sido registrada exitosamente
Descubrimiento de equipos ocultos	El atacante genera peticiones a través de una búsqueda de dispositivos por dirección física, iterando desde cierto rango de direcciones o a través de un diccionario preestablecido, hasta que un dispositivo conteste a la petición	Utilizar un esquema en el que se apague el dispositivo bluetooth cuando no esté siendo utilizado
Rogue Access Point	El atacante puede implementar un servidor bluetooth malicioso al cual los usuarios pueden asociarse, a riesgo de obtener alguna especie de virus o aplicaciones para generar acceso al dispositivo asociado	Generar un esquema de autenticación que incorpore la utilización de llaves dinámicas
Potencial de hijacking	El atacante puede tomar control de una sesión ya establecida por otro usuario, aprovechando el mal manejo de sesiones	Incorporar un esquema de autenticación al robot que se base en un modelo cliente-servidor para mapeo de adecuado de códigos PIN a dispositivos confiables

### Contacto

Para cualquier aclaración o apoyo con respecto al código se podrán consultar a las siguientes personas.

Alfonso Palacios  
[nirvanaapc@gmail.com](mailto:nirvanaapc@gmail.com)

César Gustavo López Zamarripa  
[cesar.lzamarripa@gmail.com](mailto:cesar.lzamarripa@gmail.com)

José Rubén Salazar  
[scarletsoul1024@gmail.com](mailto:scarletsoul1024@gmail.com)

# Anexos

- **Comandos del robotCem**

Al momento de hacer las pruebas con el robot, nos dimos cuenta, que parte de la información de comunicación se encontraba incompleta, por lo que a continuación se presenta la documentación de comunicación resultante.

## MOTORES

Dirección de los motores: (Se puede asociar o no con los comandos de velocidad)

- (w) adelante
- (a) izquierda
- (d) derecha
- (s) atras
- (q) detenerse

Velocidad de los motores: (Se puede asociar o no con los comandos de dirección)

- (0) motores detenidos
- (1) 20% de la velocidad maxima
- (2) 40% de la velocidad maxima
- (3) 60% de la velocidad maxima
- (4) 80% de la velocidad maxima
- (5) Velocidad maxima

## SENSORES

Medir:

- (l) luminosidad
- (p) posicion del carro (acelerometro)
- (t) temperatura
- (u) distancia con ultrasonido
- (v) velocidad de las ruedas: (Los comandos (1) y (2) se asocian con esto)
- (1) motor 1
- (2) motor 2

## SALIDAS

Encender:

- (n) bocina
- (o) leds de alta luminosidad
- (z) Led normal: (Se asocia con los comandos (1),(2),(3) y (4))
- (1) led 1
- (2) led 2
- (3) led 3
- (4) led 4
- (c) Sensor seguidor de línea (Se asocia con los comandos (1) y (2))
- (1) Sensor derecho
- (2) Sensor izquierdo