

# Guía rápida del módulo de reconocimiento de voz VRbot

## 1.- INTRODUCCION

Se trata de un potente y asequible módulo para el reconocimiento de voz. Estas voces pueden estar predefinidas y grabadas internamente. Se dice que son voces independientes del micrófono (SI). También puede grabar y reconocer voces definidas por el propio usuario y en cualquier idioma. En este caso se tratan de voces dependientes del micrófono (SD), que se supone previamente han sido “enseñadas” y grabadas en la memoria interna del módulo.

Esta guía está realizada a partir de la documentación que facilita el fabricante y trata de explicar las prestaciones y recursos del módulo VRbot. También se comentan una serie de ejercicios que, a modo de ejemplo, tratan de mostrar posibles aplicaciones. Los ejemplos se presentan en un CD-ROM que se adjunta con el módulo VRbot. Son ejemplos que están resueltos tanto en ensamblador como en C para la familia de microcontroladores PIC16F88X y están implementados sobre el laboratorio USB-PIC’SCHOOL de Ingeniería de Microsistemas Programados S.L., aunque pueden ser fácilmente adaptables a cualquier otra plataforma o lenguaje de programación.

### 1.1 Aplicaciones

Son numerosas las posibles aplicaciones del módulo de reconocimiento VRbot. Algunas sugerencias:

- Sistemas de control de propósito general que se deseen gobernar mediante voz
- Automatización de aplicaciones en el ámbito doméstico
- Control de acceso por voz
- Sistemas robóticos controlados por voz

### 1.2 Características

Se pueden destacar las más relevantes:

- 23 comandos y voces pre-programados (SI) en: en inglés, italiano, japonés y alemán
- Hasta 32 comandos y sus voces definidas por el usuario (SD) en cualquier idioma
- Los comandos y voces SD se pueden organizar en hasta 16 grupos
- Se dispone de un interface gráfico de usuario (GUI) para Windows que permite una rápida familiarización con el uso del módulo.
- El módulo se puede conectar fácilmente con cualquier tipo de controlador mediante una sencilla comunicación serie.
- Dispone de un potente protocolo de comandos para el desarrollo de cualquier aplicación basada en el reconocimiento de voz.
- Alimentación de 3.3V a 5V
- Se suministra junto con un CD-ROM que contiene el presente documento y una serie de librerías y ejemplos de aplicación para PIC16F88X desarrollados por Ingeniería de Microsistemas Programados

### 1.3 Presentación

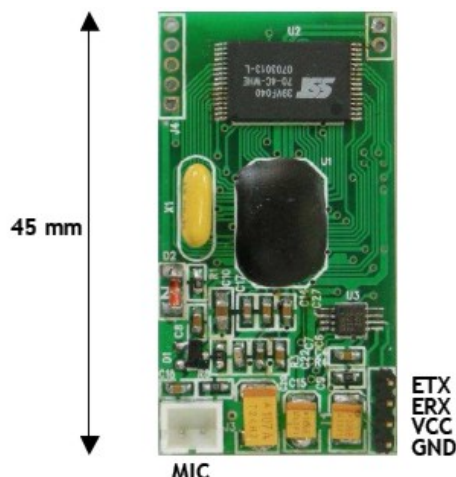
El módulo se presenta totalmente montado y comprobado junto con los cables de alimentación, comunicación y micrófono, tal y como se muestra en la figura 1.



**Figura 1.** El módulo de reconocimiento VRbot

## 1.4 Especificaciones técnicas

La figura 2 muestra las dimensiones físicas del módulo VRbot así como la disposición de las señales que se explican en la siguiente tabla.



Pin Nº	NOMBRE	TIPO	DESCRIPCION
1	GND	-	Tierra
2	VCC	I	Entrada de alimentación
3	ERX	I	Recepción de datos serie con niveles TTL
4	ETX	O	Transmisión de datos serie con niveles TTL

Figura 2. Distribución de señales del VRbot

En la siguiente tabla se resumen las características eléctricas:

SIMBOLO	PARAMETRO	Mín.	Típ.	Máx.	UNIDAD
VCC	Tensión de alimentación	3.3	5	5.5	V
Ta	Temperatura ambiente de trabajo	0	25	7'	°C
ERX	Recepción de datos serie	0	-	VCC	V
ETX	Transmisión de datos serie	0	-	VCC	V
Is	Consumo en el modo sleep		<1		mA
Io	Consumo en el modo normal de operación		12		mA

## 2.- GUI: EL INTERFACE GRAFICO DE USUARIO

Junto con el módulo VRbot se adjunta un software para Windows, el VRbot GUI. Se trata de un interface gráfico de usuario que permite utilizar y analizar de forma rápida, eficaz y sencilla todas las funciones del módulo de reconocimiento VRbot.

Su instalación en el PC es muy sencilla y lo único que hay que realizar son las conexiones adecuadas de las señales ETX, ERX y alimentación. Se recuerda que las señales de comunicación de datos son compatibles con niveles de tensión TTL por lo que **NO** se pueden conectar directamente al puerto COM del PC. Hay que realizar el clásico circuito para la adaptación de niveles TTL a RS232, p.e. mediante el conocido circuito MAX232 como se muestra en el esquema de la figura 3.

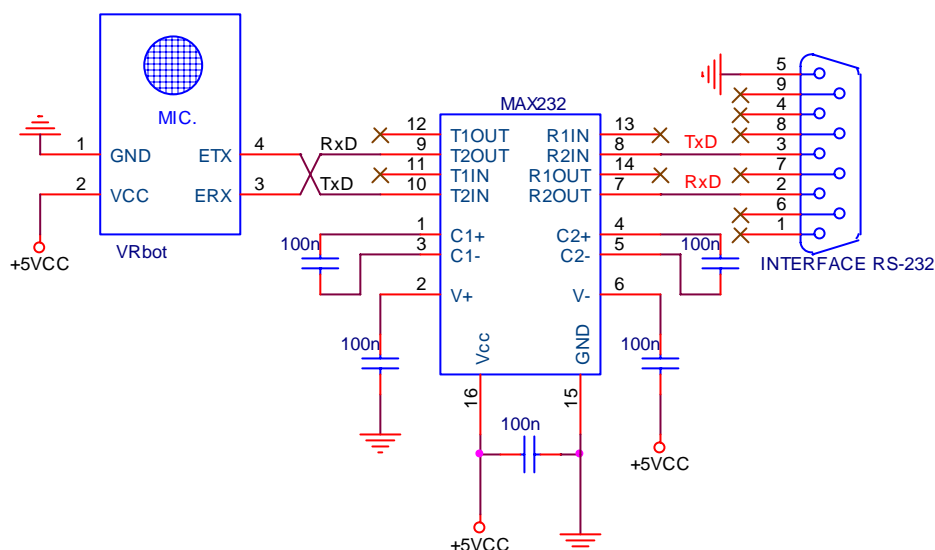
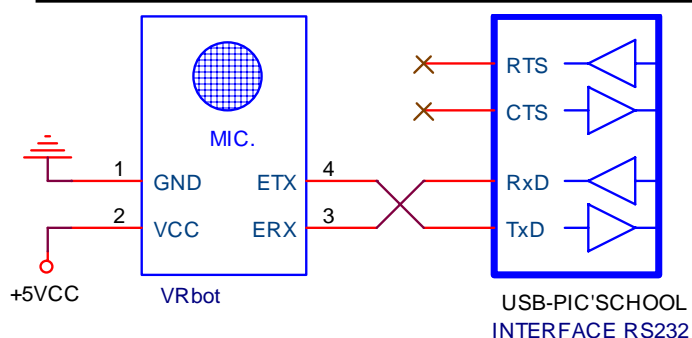
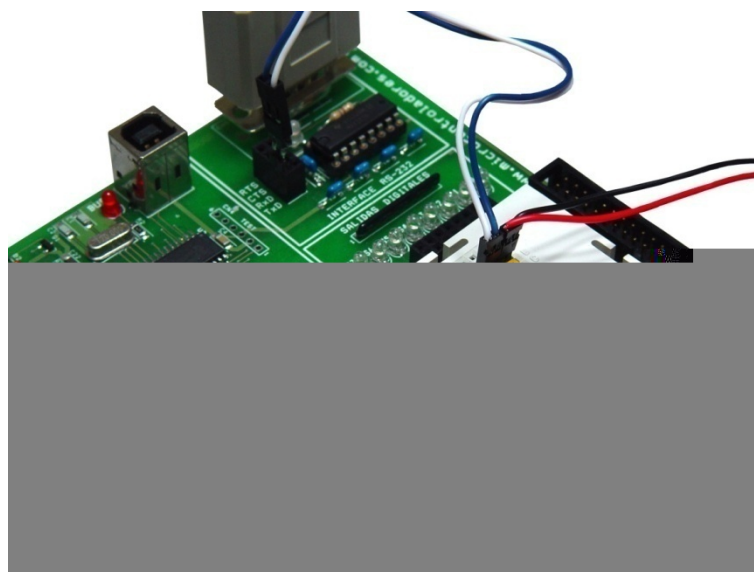


Figura 3. Esquema de interface de TTL a RS232 mediante el MAX232



En Ingeniería de Microsistemas Programados hemos empleado nuestro laboratorio USB-PIC'SCHOOL y el interface RS232 con el que está dotado. El mismo laboratorio también dispone de la alimentación de +5VCC necesaria. Así, en la figura 4 se muestra las conexiones realizadas y, en la figura 5, se muestra una fotografía del montaje.

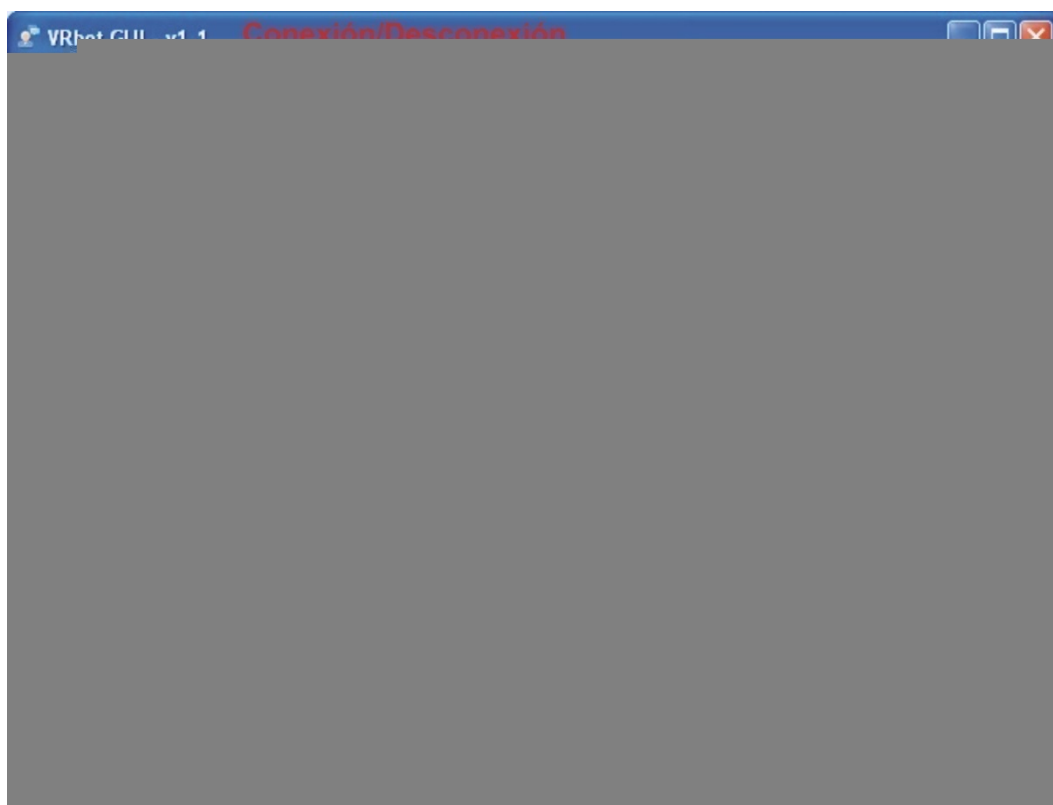
**Figura 4.** Esquema de conexiones



**Figura 5.** Montaje práctico del VRbot sobre USB-PIC'SCHOOL

## 2.1 Estableciendo la comunicación

Una vez instalado el programa VRbot GUI y realizadas las conexiones eléctricas entre el módulo y el COM del PC, ejecutamos la aplicación. Nos aparecerá una ventana de trabajo como la mostrada en la figura 6.



**Figura 6.** El área de trabajo

En el ángulo superior izquierdo tenemos un desplegable que permitirá seleccionar el canal COM disponible en nuestro PC. También disponemos de dos botones para la conexión/desconexión de la comunicación entre el módulo VRbot y el PC.

En la parte superior central existen una serie de botones con las diferentes tareas que podemos realizar con el módulo: añadir/editar/borrar comandos, enseñanza de las voces asociadas a los comandos, reconocimiento de esas voces, etc. Cada vez que se establece la comunicación el software VRbot GUI lee el contenido del módulo en su totalidad y lo traslada a las diferentes áreas de trabajo: el área de grupos y el área de comandos.

En el área de grupos aparecen los grupos 0-16 disponibles para almacenar las voces definidas por el usuario (SD) y otros tres grupos llamados "Wordset 1-3" donde están almacenadas las voces pre grabadas o voces SI. Al lado de cada grupo aparecerá también el nº de voces que contiene el mismo. Se recuerda que en el módulo se pueden grabar un **TOTAL** de 32 voces SD definidas por el usuario. Lo que pasa es que estas voces se pueden almacenar o clasificar en diferentes grupos si fuera necesario.

En el área de comandos se Editan/Visualizan las etiquetas o comandos asociados a las voces y qué posición ocupan dentro de un determinado grupo. Un comando o etiqueta no es ni más ni menos que una cadena ASCII de hasta 32 caracteres. Toda etiqueta o comando debe estar asociado a una voz o viceversa, durante la fase de enseñanza. De esta forma, cuando el módulo reconozca una voz, devuelve el comando o etiqueta asociada a la misma. Sólo se admiten etiquetas formadas por los caracteres ASCII en mayúsculas desde la A a la Z.

## 2.2 Las voces SI pre definidas

Vienen grabadas de fábrica y no pueden ser modificadas. Pueden ser reconocidas en 4 idiomas diferentes: Inglés, Italiano, Japonés y Alemán. Están almacenadas en los grupos llamados Trigger, Wordset 1, Wordset 2 y Wordset 3. La siguiente tabla muestra los grupos con las etiquetas o comandos (y sus voces) que contienen y para visualizarlas basta con seleccionar, en el área de grupos, el grupo deseado.

Trigger	Wordset 1	Wordset 2	Wordset 3
0 ROBOT	0 ACTION	0 LEFT	0 ZERO
	1 MOVE	1 RIGHT	1 ONE
	2 TURN	2 UP	2 TWO
	3 RUN	3 DOWN	3 THREE
	4 LOOK	4 FORWARD	4 FOUR
	5 ATTACK	5 BACKWARD	5 FIVE
	6 STOP		6 SIX
	7 HELLO		7 SEVEN
			8 EIGHT
			9 NINE
			10 TEN

Vamos a realizar un primer ejemplo de reconocimiento de voces pre definidas SI. Para ello:

1. Seleccionamos, mediante el botón "*Set Language*", el idioma deseado: Inglés (por defecto), Italiano, Japonés o Alemán (elige el que mejor pronuncies).
2. Seleccionamos el grupo en que se encuentran las voces a reconocer, por ejemplo el Wordset 3.
3. Mediante el botón "*Test Group*" iniciamos el reconocimiento.
4. Dictamos cualquiera de las voces disponibles en el grupo seleccionado.
5. Si la voz es reconocida, la etiqueta o comando asociada a ella quedará resaltada.
6. Repetir el proceso con diferentes voces de ese grupo o de cualquier otro.
7. Si una vez se inicia el proceso de reconocimiento, se tarda un cierto tiempo en dictar la voz, aparecerá un mensaje de rebasamiento del tiempo "*Timeout expired*".

En la figura 7 se muestra el resultado de reconocer una voz. En primer lugar en el área de grupos aparece resaltado el grupo Wordset 3 seleccionado. En el área de datos aparece resaltada en verde la etiqueta o comando correspondiente a la voz pronunciada (p.e. "*FIVE*") que ocupa la posición 5 del grupo seleccionado.

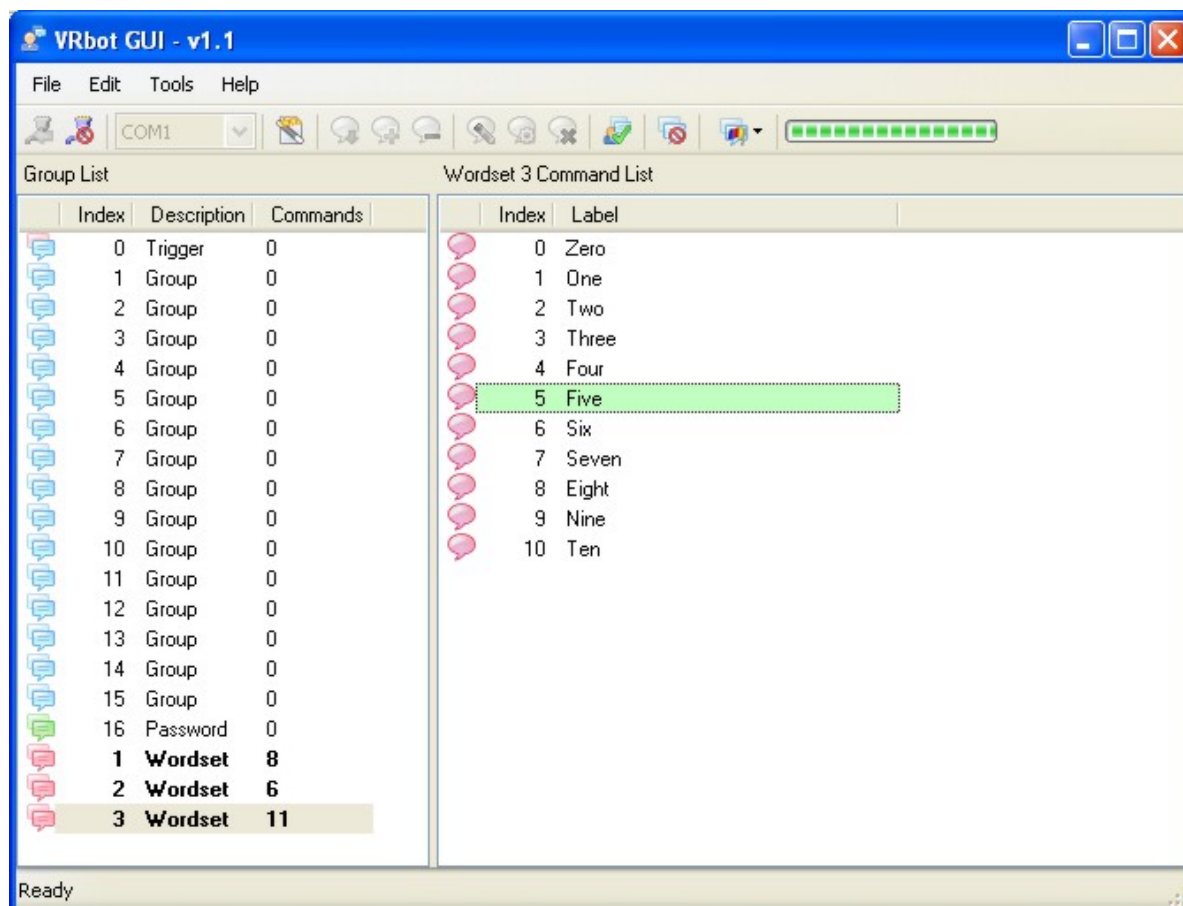


Figura 7. Reconocimiento de una voz

## 2.3 Las voces SD definidas por el usuario

Son las que puede grabar el usuario en cualquier idioma para luego ser reconocidas. El máximo de voces admitidas es de 32 y pueden ser organizadas en diferentes grupos según convenga.

### 2.3.1 Edición de etiquetas y comandos

Como ya se ha comentado, una etiqueta o comando no es más que una cadena ASCII que posteriormente se asociará a una voz de forma que, cuando esta sea reconocida, el sistema nos responda con ese comando asociado. En primer lugar vamos a crear o editar una serie de comandos. Para ello:

1. Seleccionamos el grupo deseado (p.e. el "Group 1")
2. Mediante el botón "Add Command" y/o "Insert Command" vamos añadiendo todas las etiquetas o comandos deseados. Proponemos que sean los mostrados en la figura 8.
3. Podemos emplear los botones "Remove Command" y/o "Rename Command" para borrar y/o renombrar (reeditar) una etiqueta o comando cualquiera de la lista.
4. Para las etiquetas o comandos sólo se admiten los caracteres ASCII de la A a la Z en mayúsculas.
5. Se puede apreciar que a cada comando se le asigna un número que se corresponde con la posición o índice de ese comando respecto al grupo en que se encuentra.
6. En el área de grupos, el grupo seleccionado, va representando el número de comandos que contiene según se van añadiendo.
7. Repetir el proceso para editar los comandos que se muestran en la figura 9 y que se introducen en el grupo 16. Estos grupos y comandos asociados serán empleados en los ejemplos de programación que se propondrán más adelante.

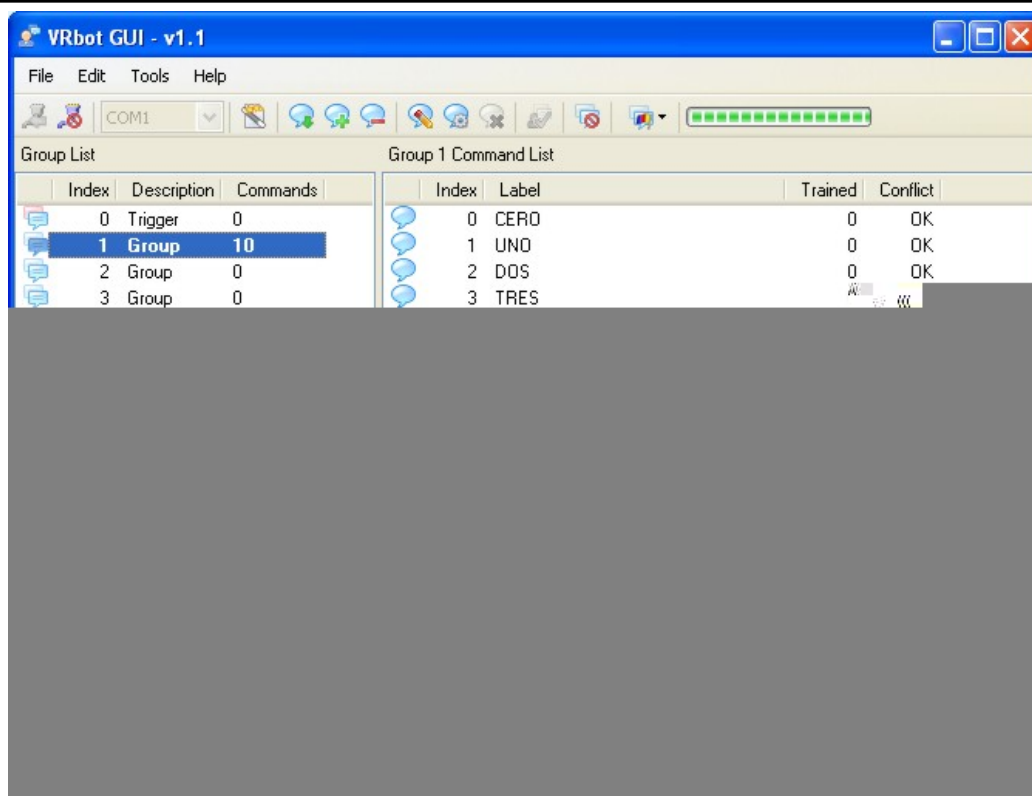


Figura 8. Edición de comandos en el grupo 1

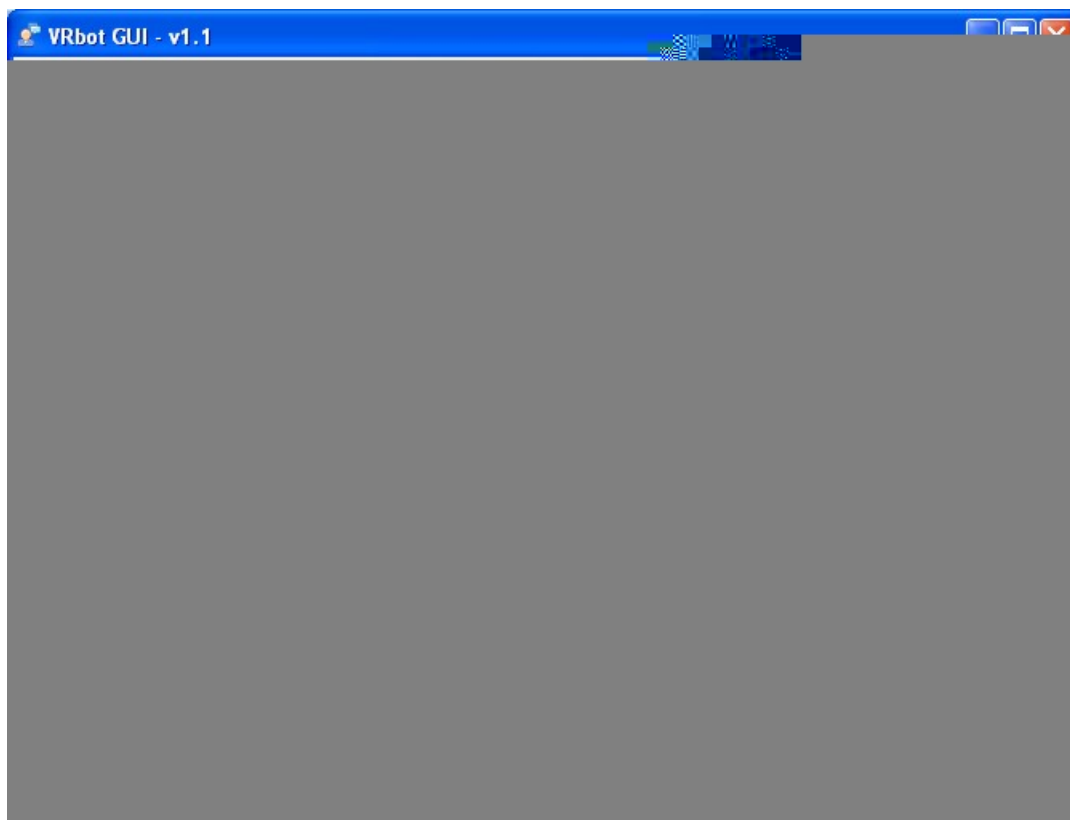


Figura 9. Edición de comandos en el grupo 2



---

### 2.3.2 El ciclo de enseñanza

Consiste en grabar las voces que se asociarán a los comandos editados anteriormente y que se usarán durante el reconocimiento. Para ello:

1. Seleccionar el grupo al que pertenece el comando al que queremos asociar una voz.
2. Seleccionar el comando propiamente dicho
3. Mediante el botón "*Train Command*" se inicia el ciclo de enseñanza.
4. Disponemos de un cierto tiempo para dictar la voz que queramos
5. Esa voz hay que volver a dictarla una 2ª vez para tener una mayor garantía en los posteriores reconocimientos.
6. Observaremos que a la derecha del comando aparece un valor que expresa el nº de veces (2) que la voz ha sido enseñada, seguido de OK cuando finaliza el ciclo.
7. Si rebasamos el tiempo o si la repetición de la voz por 2ª vez no corresponde con la de la 1ª, aparecerá el correspondiente mensaje de inicio.
8. Repetimos el proceso con todos los comandos disponibles en todos los grupos.

Se debe indicar que, aunque es muy frecuente que una voz coincida con su correspondiente comando, esto no tiene porqué ser necesariamente así. Hay una asociación entre voz y comando o etiqueta aunque sean diferentes. Por ejemplo a la etiqueta "UNO" le podemos asociar la voz "HOLA". Durante el posterior, reconocimiento cuando dictemos la voz "HOLA", el módulo nos responderá con la etiqueta asociada "UNO" y la posición que ocupa dentro del grupo.

### 2.3.3 El reconocimiento

A estas alturas ya tenemos nuestro módulo VRbot con una serie de etiquetas y voces SD definidas por nosotros, los usuarios. Para el reconocimiento de las mismas basta seguir los mismos pasos que se hicieron anteriormente con el reconocimiento de voces predefinidas SI. Se resumen a continuación

1. Seleccionamos el grupo en que se encuentran las voces a reconocer, por ejemplo el Group 1.
2. Mediante el botón "Test Group" iniciamos el reconocimiento.
3. Dictamos cualquiera de las voces disponibles en el grupo seleccionado.
4. Si la voz es reconocida, la etiqueta o comando asociada a ella quedará resaltada.
5. Repetir el proceso con diferentes voces de ese grupo o de cualquier otro.
6. Si una vez se inicia el proceso de reconocimiento, se tarda un cierto tiempo en dictar la voz, aparecerá un mensaje de rebasamiento del tiempo "*Timeout expired*".

### 3.- EL PROTOCOLO DE COMUNICACIÓN DEL MODULO VRbot

Este es el tema más espinoso. Sin embargo gracias al protocolo de comunicación del VRbot nos será posible realizar nuestras propias aplicaciones a medida. Efectivamente, el VRbot se puede conectar, además de a un PC, a cualquier otro sistema de control basado en microcontroladores como pueden ser los PIC's, Basic Stamp, Atmel, Arduino, etc... que actuarán como Host.

El VRbot emplea un UART como interface estándar de comunicación con niveles lógicos compatibles TTL de acuerdo a la tensión con la que se le alimente (3.3-5V). Las conexiones entre el VRbot y el microcontrolador principal (Host MCU) son compatibles a nivel eléctrico. Lo único que se debe tener en cuenta es que la señal de transmisión de datos del VRbot ETX debe conectarse con la de recepción RX del controlador y viceversa. Es decir, las señales se deben cruzar como se muestra en la figura 10.

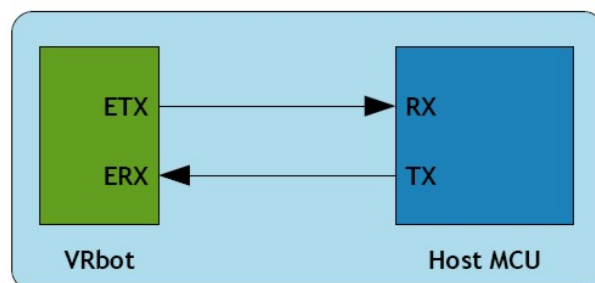


Figura 10. Conexión entre el VRbot y un Host

Por defecto, al conectar la alimentación, la configuración de la comunicación es de 9600 baudios, 8 bits de datos, 1 de stop y sin paridad. Los baudios pueden configurarse en el rango de 9600 hasta 115200.

En la comunicación solo se emplean caracteres ASCII imprimibles, lo que permite el control de VRbot desde cualquier software de comunicación para PCD como, por ejemplo, el hyperterminal de Windows. Se siguen los siguientes criterios:

- Los comandos que se transmiten desde el Host al módulo por la línea TX son siempre caracteres en minúsculas. Por su parte el VRbot responde al Host por la línea RX unos caracteres de estado que también son siempre en minúsculas.
- Los argumentos o parámetros necesarios tanto para los comandos transmitidos como los de estado recibidos, son siempre caracteres en mayúsculas.
- Cada comando que transmite el Host por la línea TX con los argumentos o parámetros necesarios (cero o más caracteres), genera la correspondiente respuesta por parte del VRbot que se recibe por la línea RX. Esta respuesta también puede tener parámetros o argumentos de cero o más caracteres.
- Existe un retardo mínimo entre cada byte enviado al Host por parte del VRbot, que por defecto es de 20ms. Se puede configurar en rangos que van de 0 a 9 ms, de 10 a 90 ms y de 100ms a 1 s. Esto permite adaptarse a la velocidad de proceso alta o baja por parte del Host.
- El flujo de información lo controla el Host en todo momento. Cada byte que manda el VRbot como réplica a un determinado comando, debe ser reconocido por parte del Host que le envía carácter *espacio* (0x20) para recibir así los sucesivos bytes de respuesta si los hubiera.
- Los comandos y/o argumentos incorrectos quedan reflejados por el correspondiente byte de estado que envía el VRbot al Host para su oportuno tratamiento. Por otra parte si el Host no envía todos los bytes de argumentos o parámetros necesarios por un determinado comando, este es ignorado por parte del módulo VRbot
- Cada vez que se conecta la alimentación el módulo VRbot queda en el modo sleep de bajo consumo. Para iniciar la comunicación basta con que el Host le mande cualquier carácter.

#### 3.1 Los argumentos

Algunos comandos necesitan argumentos o parámetros numéricos para su correcta ejecución. Los valores numéricos de esos parámetros están en el rango de -1 hasta 31. Sin embargo, como ya se ha comentado, la comunicación con el módulo VRbot se realiza siempre mediante caracteres ASCII imprimibles. Así pues los valores numéricos de los parámetros habrá que codificarlos en su equivalente carácter ASCII y viceversa. Se sigue el siguiente criterio: partiendo del carácter '@' (0x40) y de forma correlativa se representan los valores del -1 al 31. Para los valores positivos del 0 al 31 basta con emplear el carácter 'A' (0x41) como offset.



<b>ASCII</b>	'@'	'A'	'B'	'C'	'D'	----	' '	'\ '
<b>HEX</b>	0x40	0x41	0x42	0x43	0x44	----	0x5F	0x60
<b>Valor</b>	-1	0	1	2	3	----	30	31

### 3.2 Los comandos

A continuación se exponen todos los comandos aceptados por el módulo VRbot. Recordemos que los argumentos o parámetros numéricos que precisan alguno de ellos deben ser codificados realmente por su equivalente carácter ASCII imprimible, en mayúsculas, según la tabla anterior.

<b>NOMBRE</b>	<b>CMD_BREAK</b>	
<b>COD./DESCRIPCION</b>	'b' (0x62)	Aborta la ejecución de cualquier comando previo
<b>ARGUMENTOS</b>	Ninguno	
<b>RESPUESTA</b>	STS_SUCCESS, STS_INTERR	

<b>NOMBRE</b>	<b>CMD_SLEEP</b>	
<b>COD./DESCRIPCION</b>	's' (0x73)	Coloca al módulo VRbot en uno de los modos sleep de bajo consumo
<b>ARGUMENTOS</b>	[1]	Modo sleep (0-8)  0=se reactiva (wake up) al recibir un carácter 1=se reactiva (wake up) al recibir un sonido agudo o un carácter 2=se reactiva (wake up) al recibir un sonido grave o un carácter 3-5=se reactiva (wake up) al recibir una doble palmada o un carácter 6-8=se reactiva (wake up) al recibir una triple palmada o un carácter
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_KNOW</b>	
<b>COD./DESCRIPCION</b>	'k' (0x6B)	Ajusta el nivel de reconocimiento de las voces pre definidas SI
<b>ARGUMENTOS</b>	[1]	0= nivel bajo: se obtienen muchos resultados válidos 2= típico: valor por defecto 4= nivel alto: se obtienen pocos resultados válidos y hay que pronunciar la voz con precisión
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_LEVEL</b>	
<b>COD./DESCRIPCION</b>	'v' (0x76)	Ajusta el nivel de reconocimiento de las voces SD definidas por el usuario
<b>ARGUMENTOS</b>	[1]	1=fácil, 2=por defecto, 5=difícil  Un valor alto supone una mayor precisión en la dicción y por tanto más errores de reconocimiento
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_LANGUAGE</b>	
<b>COD./DESCRIPCION</b>	'l' (0x6C)	Ajusta el lenguaje empleado en el reconocimiento de las voces SI
<b>ARGUMENTOS</b>	[1]	0= Inglés, 1= Italiano, 2= Japonés y 3= Alemán
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_TIMEOUT</b>	
<b>COD./DESCRIPCION</b>	'o' (0x6F)	Ajusta el tiempo de espera durante el reconocimiento de una voz
<b>ARGUMENTOS</b>	[1]	-1= defecto, 0= infinito, 1 a 31= segundos
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_RECOG_SI</b>	
<b>COD./DESCRIPCION</b>	'i' (0x69)	Activa el ciclo de reconocimiento de voces predefinidas SI en el grupo o wordset especificado
<b>ARGUMENTOS</b>	[1]	0-3= N° del grupo o wordset donde se realizará el reconocimiento
<b>RESPUESTA</b>	STS_SIMILAR, STS_TIMEOUT, STS_ERROR	

<b>NOMBRE</b>	<b>CMD_TRAIN_SD</b>	
<b>COD./DESCRIPCION</b>	't' (0x74)	Activa el ciclo de enseñanza para una nueva voz SD definida por el usuario
<b>ARGUMENTOS</b>	[1]	Grupo donde se grabará la nueva voz: 0= Trigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
	[2]	Posición dentro del grupo en donde se grabará la voz (0 a 31)
<b>RESPUESTA</b>	STS_SUCCESS, STS_RESULT, STS_TIMEOUT, STS_ERROR	

<b>NOMBRE</b>	<b>CMD_GROUP_SD</b>	
<b>COD./DESCRIPCION</b>	'g' (0x67)	Permite insertar un nuevo comando o etiqueta
<b>ARGUMENTOS</b>	[1]	Grupo donde se insertará el nuevo comando: 0= Trigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
	[2]	Posición dentro del grupo a partir de la cual se insertara la nueva etiqueta o comando (0 a 31)
<b>RESPUESTA</b>	STS_SUCCESS, STS_OUT_OF_MEM	

<b>NOMBRE</b>	<b>CMD_UNGROUP_SD</b>	
<b>COD./DESCRIPCION</b>	'u' (0x75)	Borra un comando o etiqueta
<b>ARGUMENTOS</b>	[1]	Grupo de donde se borrará el comando: 0= Tigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
	[2]	Posición dentro del grupo de donde se quita el comando o etiqueta (0 a 31)
<b>RESPUESTA</b>	STS_SUCCESS	

<b>NOMBRE</b>	<b>CMD_RECOG_SD</b>	
<b>COD./DESCRIPCION</b>	'd' (0x64)	Inicia el ciclo de reconocimiento de una voz SD definida por el usuario

NOMBRE	CMD_NAME_SD	
COD./DESCRIPCION	'n' (0x6E)	Almacena los caracteres que forman la etiqueta o comando
ARGUMENTOS	[1]	Grupo donde se almacenará la etiqueta: 0= Tigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
	[2]	Posición dentro del grupo donde se desea grabar los caracteres que definen una etiqueta o comando (0 a 31)
	[3]	Número de caracteres que conforman la etiqueta o comando (0 a 31)
	[4 a n]	Caracteres de la cadena que forman la etiqueta. Deben ser caracteres ASCII de la A al ' '. Máximo 32
RESPUESTA	STS_SUCCESS	

NOMBRE	CMD_COUNT_SD	
COD./DESCRIPCION	'c' (0x63)	Obtiene el número de comandos o etiquetas que tiene almacenadas el grupo indicado
ARGUMENTOS	[1]	Grupo del que se quiere averiguar cuántas etiquetas o comandos tiene: 0= Tigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
RESPUESTA	STS_COUNT	

NOMBRE	CMD_DUMP_SD	
COD./DESCRIPCION	'p' (0x70)	Lee la cadena de caracteres que forman una etiqueta o comando
ARGUMENTOS	[1]	Grupo donde se encuentra el comando: 0= Tigger, 1-15= grupos del 1 al 15, 16=grupo 16 o password
	[2]	Posición dentro del grupo en que se encuentra la etiqueta o comando (0 a 31)
RESPUESTA	STS_DATA	

NOMBRE	CMD_MASK_SD	
COD./DESCRIPCION	'm'(0x6D)	Devuelve una máscara de 32 bits que expresa qué grupos están vacíos o bien contienen alguna etiqueta/comando o voz
ARGUMENTOS	Ninguno	
RESPUESTA	STS_MASK	

NOMBRE	CMD_RESETALL	
COD./DESCRIPCION	'r' (0x72)	Borra todos las etiquetas/comandos y voces SD de todos los grupos (excepto los predefinidos SI)
	'R'(0x52)	Carácter de confirmación. El módulo VRbot queda virgen
ARGUMENTOS	Ninguno	
RESPUESTA	CMD_SUCCESS	

NOMBRE	CMD_ID	
COD./DESCRIPCION	'x' (0x78)	Devuelve el código o versión del firmware interno del módulo VRbot
ARGUMENTOS	Ninguno	
RESPUESTA	STS_ID	

NOMBRE	CMD_DELAY	
COD./DESCRIPCION	'y' (0x79)	Ajusta el retardo en la transmisión de un byte y el siguiente
ARGUMENTOS	[1]	Retardo: 0-10= de 0 a 10ms; 11-19= de 20 a 100mS; 20-28= de 200 a 1000mS
RESPUESTA	STS_SUCCESS	

NOMBRE	CMD_BAUDRATE	
COD./DESCRIPCION	'a' (0x61)	Ajusta la velocidad de comunicación
ARGUMENTOS	[1]	Baudios: 1= 115200; 2= 57600; 3= 38400; 6= 19200 y 12= 9600
RESPUESTA	STS_SUCCESS	

### 3.3 Las respuestas

Todos los comandos transmitidos por el Host genera la correspondiente respuesta por parte del VRbot que debe ser recibida y analizada por el Host. Estas respuestas constan siempre en un carácter o código y a su vez pueden tener también argumentos o parámetros numéricos. Estos se representarán mediante los correspondientes caracteres ASCII imprimibles como se explicó con anterioridad. Se recuerda que cada vez que el VRbot transmite un byte de parámetro al Host, este debe transmitir el carácter espacio 0x20 para recibir el siguiente byte.

NOMBRE	STS_MASK	
COD./DESCRIPCION	'k' (0x6B)	Devuelve 8 bytes que representan la máscara que indica qué grupos están vacíos y cuáles no
ARGUMENTOS	[1 – 8]	De cada byte del argumento sólo se emplean los 4 bits de menos peso. Se obtiene así una máscara de 32 bits. El bit de menos peso representa al grupo 0, el siguiente bit al grupo 1 y así sucesivamente. Sin un bit está a "0" indica que su correspondiente grupo está vacío. Si el bit está a "1" indica que su correspondiente grupo contiene alguna etiqueta/comando y voz.
EN RESPUESTA A	CMD_MASK_SD	

NOMBRE	STS_COUNT	
COD./DESCRIPCION	'c' (0x63)	Devuelve el nº de comandos o etiquetas de un determinado grupo
ARGUMENTOS	[1]	Número de etiquetas o comandos existentes en un grupo (0 a 31)
EN RESPUESTA A	CMD_COUNT_SD	

NOMBRE	STS_AWAKEN	
COD./DESCRIPCION	'w' (0x77)	Re activación o wake up del módulo tras la conexión de la alimentación o tras salir de cualquiera de los modos sleep de bajo consumo
ARGUMENTOS	Ningun	
EN RESPUESTA A	A cualquier carácter recibido tras conectar la alimentación o tras haber ejecutado uno de los modos sleep de bajo consumo	

NOMBRE	STS_DATA	
COD./DESCRIPCION	'd' (0x64)	Devuelve datos
ARGUMENTOS	[1]	Información de la enseñanza: 0 a 7= número de veces que se grabó correctamente una voz SD; >8= Conflicto en la voz SD grabada; >16 conflicto en una voz SI
	[2]	Posición dentro del grupo (0 a 31)
	[3]	Número de caracteres que tiene la etiqueta o comando
	[4 a n]	Caracteres de la cadena que forman la etiqueta.
EN RESPUESTA A	CMD_DUMP_SD	

---

NOMBRE	STS_ERROR
--------	-----------

---

NOMBRE	STS_ID	
COD./DESCRIPCION	'x' (0x78)	Proporciona información del firmware interno del VRbot
ARGUMENTOS	[1]	Identificación de la versión
EN RESPUESTA A	CMD_ID	

#### 4.- LIBRERIAS

En el CD-ROM, junto con los ejemplos propiamente dichos, se proporcionan dos librerías escritas tanto en C (\*.H) como en ensamblador (\*.INC), que incluyen una serie de funciones o rutinas que facilitan el empleo de la pantalla LCD del laboratorio USB-PIC'SCHOOL y del módulo de reconocimiento VRbot. Estas librerías se deben incluir en nuestros programas fuente de aplicación mediante las directivas `<include>`

##### 4.1 Librería LCD4bitsPIC16

Contiene funciones de propósito general para controlar una pantalla LCD alfanumérica mediante un controlador PIC de la familia PIC16FXXX y con un interface de 4 bits. En la siguiente tabla se resumen las funciones o rutinas que contiene esta librería.

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
UP_LCD	Ninguno	Ninguno	Configura las líneas de E/S para adaptarlas a la pantalla LCD
LCD_INI	Ninguno	Ninguno	Rutina para la inicialización de la pantalla LCD según especificaciones del fabricante
LCD_DATO	W=Contiene el carácter a visualizar	Ninguno	Envía a la pantalla el dato a visualizar en la posición actual del cursor
LCD_REG	W=Contiene el comando a ejecutar por parte de la pantalla LCD	Ninguno	Envía a la pantalla el comando que debe ejecutar

##### 4.2 Librería VRbot\_16FXXX

Contiene funciones específicas para controlar el módulo de reconocimiento de voz VRbot desde un controlador PIC16FXXX. La siguiente tabla resume las funciones o rutinas contenidas en esta librería

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
VRbot_Serie	Ninguno	VRbot_Status='o' (0x6F)	Inicia el EUART para la comunicación serie con el módulo de reconocimiento VRbot. Por defecto esta se realizará a 9600 baudios, 8 bits de datos, 1 de stop y sin paridad. También espera el Wake Up del módulo tras un Power Down.
VR_BREAK	Ninguno	VRbot_Status='o' (0x6F) o 'i' (0x69)	Aborta cualquier operación en curso.
VR_ID	Ninguno	VRbot_Dato1=Nº de versión	Devuelve la identificación o versión del firmware interno del módulo de reconocimiento VRbot.
VR_COUNT_SD	VRbot_Grupo=Nº de grupo a inspeccionar	VRbot_Cont=Nº de comandos	Devuelve el número de comandos (0-31) que hay en un determinado grupo
VR_DUMP_SD	VRbot_Grupo=Nº de grupo a inspeccionar (0-16) VRbot_Pos=Nº de posición dentro del grupo (0-32)	VRbot_Buff=etiqueta VRbot_Cont=Nº de caracteres de la etiqueta VRbot_Dato1= Info. De la enseñanza VRbot_Dato2= Posición conflictiva	Devuelve a partir de la posición de VRbot_Buff la etiqueta o comando que haya en el grupo indicado y en la posición señalada dentro de ese grupo.  La etiqueta de un comando puede tener un máximo de 32 caracteres.
VR_KNOW	VRbot_Dato1=Nivel	VRbot_Status='o' (0x6F)	Ajusta el nivel de exigencia de reconocimiento para las voces predefinidas



			(SI). La variable VRbot_Dato1 debe contener el valor del nivel: 0=nivel bajo (muchos resultados válidos), 2=valor típico (por defecto), 4=alto (pocos resultados válidos)
<b>VR_LEVEL</b>	VRbot_Dato1=Nivel	VRbot_Status='o' (0x6F)	Ajusta el nivel de exigencia de reconocimiento para las voces del usuario (SD). La variable VRbot_Dato1 debe contener el valor del nivel: 0=nivel bajo (mínimo), 2=valor típico (por defecto), 5=nivel alto (máximo, pocos resultados válidos)
<b>VR_LANGUAGE</b>	VRbot_Dato1=Nº de idioma	VRbot_Status='o' (0x6F)	Ajusta el lenguaje empleado para el reconocimiento de las voces predefinidas (SI). La variable VRbot_Dato1 debe contener el valor del lenguaje: 0=Ingles, 1=Italiano, 2=Japones, 3=Alemán
<b>VR_TIMEOUT</b>	VRbot_Dato1=segundos	VRbot_Status='o' (0x6F)	Ajusta el tiempo de respuesta o Time Out. Si durante la ejecución de determinados comandos se supera este tiempo, dicha ejecución se da por finalizado. La variable VRbot_Dato1 se supone cargada con el valor a temporizar: -1 (0xFF)= defecto, 0=infinito, de 1 a 32 valor del Time Out en segundos
<b>VR_DELAY</b>	VRbot_Dato1=mili segundos	VRbot_Status='o' (0x6F)	Ajusta el tiempo mínimo que emplea el módulo VRbot entre la transmisión de un byte y el siguiente. La variable VRbot_Dato1 se supone cargada con el valor: 0-10= de 0 a 10mS, 11-19= de 20-100mS, 20-28=de 200mS a 1s.
<b>VR_BAUDRATE</b>	VRbot_Dato1=baudios	VRbot_Status='o' (0x6F)	Ajusta la velocidad de comunicación (baudios) entre el módulo VRbot y el controlador. La variable VRbot_Dato1 se supone cargada con el valor: 1=115200, 2=57600, 3=38400, 6=19200, 12=9600.
<b>VR_SLEEP</b>	VRbot_Dato1=modo	VRbot_Status='o' (0x6F)	Determina el modo power down (dormir) de bajo consumo. La variable VRbot_Dato1 se supone cargada con el modo: 0= Despertar al recibir un carácter, 1= despertar con un silbido o al recibir, 2= despertar con un ruido o al recibir carácter, 3-5= con una doble palmada o al recibir un carácter, 6-8= con una triple palmada o al recibir un carácter.
<b>VR_RECOG_SD</b>	VRbot_Grupo=Nº de grupo (0 a 16)	VRbot_Status='r' o 't' o 'e' VRbot_Pos=Posición de la voz en el grupo VRbot_Cont=Nº de caracteres de la etiqueta VRbot_Buff= etiqueta VRbot_dato1 y VRbot_dato2= Códigos de error	Activa el reconocimiento de una voz definida por el usuario (SD). En VRbot_Grupo debe indicarse el grupo (0-16). En VRbot_Status se devuelve: 'r' si la voz ha sido reconocida, 't' si ha habido sobre pasamiento en el tiempo o 'e' si ha habido error en el reconocimiento. En VRbot_Pos se devuelve la posición dentro del grupo que ocupa la voz, en caso de ser reconocida. En este caso su correspondiente comando o etiqueta se almacena a partir de VRbot_Buffer y VRbot_Cont indicará el número de caracteres. En VRbot_Dato1 y VRbot_Dato2 se indica el código de error en caso de que lo hubiera. En caso de error o de Time Out,

			dicho buffer queda borrado.
<b>VR_RECOG_SI</b>	VRbot_Grupo=Nº de grupo (0-3)	VRbot_Status='s', 't' o 'e' VRbot_Pos= posición dentro del grupo VRbot_dato1 y VRbot_dato2= Códigos de error	Activa el reconocimiento de una voz predefinida (SI). En VRbot_Grupo debe indicarse el grupo (0-3). En VRbot_Status se devuelve: 's' si la voz es similar, 't' si ha habido sobre pasamiento en el tiempo o 'e' si ha habido error en el reconocimiento. En VRbot_Pos se devuelve la posición dentro del grupo que ocupa la voz en caso de ser similar. En VRbot_Dato1 y VRbot_Dato2 se indica el código de error en caso de que lo hubiera.
<b>VR_ADD_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Posición VRbot_Buff=etiqueta	VRbot_Pos='o', 'e', 'm' o 'v'	Añade un nuevo comando o etiqueta SD. En VRbot_Grupo se indicará el grupo donde se va a insertar, en VRbot_Pos se indicará la posición dentro del grupo, En VRbot_Cont se indica el número de caracteres de que consta el comando o etiqueta y, a partir de VRbot_Buff, se supone almacenada la cadena ASCII que define el comando. En VRbot_Status se devuelve 'o' si la ejecución es correcta, 'e' si hay error o 'm' si no hay memoria suficiente para almacenar el nuevo comando o 'v' si el comando o argumentos no son válidos.
<b>VR_TRAIN_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Posición	VRbot_Status='o', 'e' o 't'	Añade una voz definida por el usuario (SD) a una determinada etiqueta o comando previamente añadida al módulo VRbot (enseñanza). En VRbot_Grupo se indica el grupo al que pertenecerá la nueva voz. En VRbot_Pos se indica la posición de la voz dentro del grupo. En VRbot_Status se devuelve 'o' si la función se ejecuta correctamente, 'e' si hay error y 't' si se produce un Timer Out durante la enseñanza.
<b>VR_ERASE_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Nº de posición	VRbot_Status='o'	Borra una de las voces definidas por el usuario (SD). En VRBot_Grupo se indica el grupo al que pertenece la voz a borrar, en VRbot_Pos se indica la posición dentro de ese grupo. En VRbot_Status se devuelve 'o' si la función se ejecuta correctamente.
<b>VR_UNGROUP_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Posición	VRbot_Status='o'	Borra la etiqueta o comando asociado a una de las voces definidas por el usuario (SD). En VRBot_Grupo se indica el grupo al que pertenece la voz a borrar, en VRbot_Pos se indica la posición dentro de ese grupo. En VRbot_Status se devuelve 'o' si la función se ejecuta correctamente.
<b>VR_ADD_TRAIN_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Posición VRbot_Cont=Nº de caracteres	VRbot_Status='o', 'e', 'm' o 'v'	Esta rutina engloba las funciones VR_ADD_SD y VR_TRAIN_SD en una única función: Añade un nuevo comando o etiqueta SD y seguidamente le asocia una nueva voz. En VRbot_Grupo se indicará el grupo donde se va a insertar, en VRbot_Pos se indicará la posición dentro del grupo, En VRbot_Cont se indica el número de caracteres de que consta el comando o etiqueta, cuya cadena ASCII se supone almacenada a partir de VRbot_Buff. En VRbot_Status se devuelve 'o' si la ejecución es correcta, 'e' si hay error o 'm' si no hay memoria suficiente para almacenar el nuevo comando, 't' si se produce un Timer

			Out durante la enseñanza o 'v' si el comando o argumentos no son válidos.
<b>VR_ERASE_UNG_SD</b>	VRbot_Grupo=Nº de grupo VRbot_Pos=Posición	VRbot_Status='o'	:: Esta rutina engloba las funciones VR_ERASE_SD y VR_UNGROUP_SD en una única función: borra una voz y seguidamente borra la etiqueta o comando asociado a la misma. En VRBot_Grupo se indica el grupo al que pertenece la voz a borrar, en VRbot_Pos se indica la posición dentro de ese grupo. En VRbot_Status se devuelve 'o' si la función se ejecuta correctamente.
<b>VR_MASK_SD</b>	Ninguno	VRbot_Status='k' VRbot_Buff=máscara	Devuelve la máscara de los grupos no vacíos. Dicha máscara consta de 4 bytes (32 bits) que se almacenan en hex. a partir de VRbot_Buff, empezando por el de menos peso. Un bit 1 representa grupo usado o no vacío. P.e. la máscara = 00000000 00000000 01001000 00000101 representa que los grupos 0, 2, 11 y 14 están usados, el resto están vacíos. En VRbot_Status se devuelve 'k'
<b>VR_RESETALL</b>	Ninguno	Ninguno	Borra todos los comandos y grupos existentes

## 5.- EJEMPLO PRACTICOS

Junto con el módulo de reconocimiento de voz se adjunta un CD-ROM con una serie de ejemplos y aplicaciones. Estos ejemplos, puramente didácticos, han sido realizados por *Ingeniería de Microsistemas Programados* con la única finalidad de facilitar el empleo del VRbot. Están diseñados para el microcontrolador PIC16F88X y se proporcionan las librerías y los programas fuente escritos tanto en lenguaje C como en ensamblador. Son fácilmente adaptables a cualquier otro dispositivo y se han realizado sobre nuestro laboratorio de prácticas USB-PIC'SCHOOL, aunque se pueden implementar fácilmente sobre cualquier otra plataforma hardware.

Esperamos que sean de ayuda a todos los usuarios que deseen incorporar la potencia y prestaciones de un sistema de reconocimiento de voz en todos sus proyectos y aplicaciones.

### 5.1 Ejemplo 1: Reconocimiento de voces predefinidas SI

#### Objetivos

Realizar una primera toma de contacto con el módulo de reconocimiento de voz VRbot controlado desde un programa de aplicación del usuario. Se trata de reconocer las voces predefinidas SI alojadas en el grupo wordset 3 y que corresponden a los números del 0 al 10 que deberán pronunciarse en inglés.

#### Esquema

La figura 11 muestra el esquema de conexiones necesario para implementar este ejemplo. Empleamos el laboratorio USB-PIC'SCHOOL conectando las líneas de RB7:RB0 del PIC con los leds de salida S7:S0. Por su parte el módulo VRbot se alimenta con +5Vcc. Su línea ETX de transmisión se conecta con RC7/RX del PIC mientras que la línea de recepción ERX se conecta con la de transmisión RC6/TX del PIC.

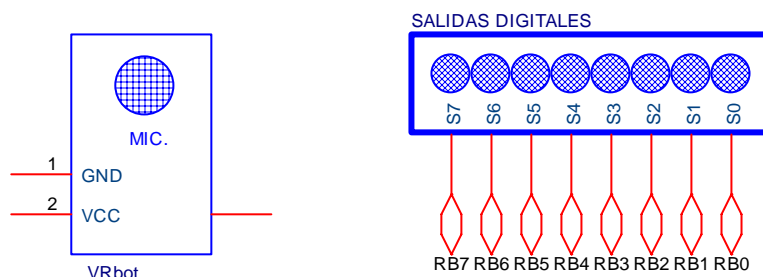


Figura 11. Esquema de conexiones

## Comentarios

El ejemplo controla las salidas RB7:RB0 con el dictado de las voces SI predefinidas “zero” a “seven” (en inglés). Cuando una voz es reconocida la salida correspondiente cambia de estado. Si se detecta un error de reconocimiento, todas las salidas producen una intermitencia. Si la voz no está comprendida entre 0 y 7 (en inglés) no se produce ningún cambio en las salidas.

En el ejemplo se puede observar que, una vez reconocida una voz, se obtiene qué número de posición ocupa esa voz dentro del grupo. Dicho número permite calcular sobre qué salida se debe actuar. En la fotografía de la figura 12 se muestra el montaje realizado sobre el laboratorio USB-PIC’SCHOOL.



**Figura 12.** Montaje práctico del ejemplo 12

A pesar de su sencillez, este ejemplo nos permite intuir las posibilidades del módulo VRbot de cara al control por voz de determinados procesos.

## 5.2 Ejemplo 2: Reconocimiento de voces SD definidas por el usuario

### Objetivos

El ejemplo es similar al anterior. En esta ocasión se trata de realizar el reconocimiento de voces definidas por el usuario y en cualquier lenguaje. Vamos a emplear el grupo 1 donde se supone se han definido las voces del 0 al 9 en cualquier lenguaje (p.e. castellano) mediante el empleo del entorno VRbotGUI, tal y como se explicó en el apartado 2.3 anterior.

### Esquema

La figura 13 muestra el esquema de conexiones. Las salidas RB0:RB7 controlan los segmentos a:dp del display de las unidades del laboratorio. El módulo VRbot se conecta como en el ejemplo anterior: ETX con RC7/RX y ERX con RC6/TX.

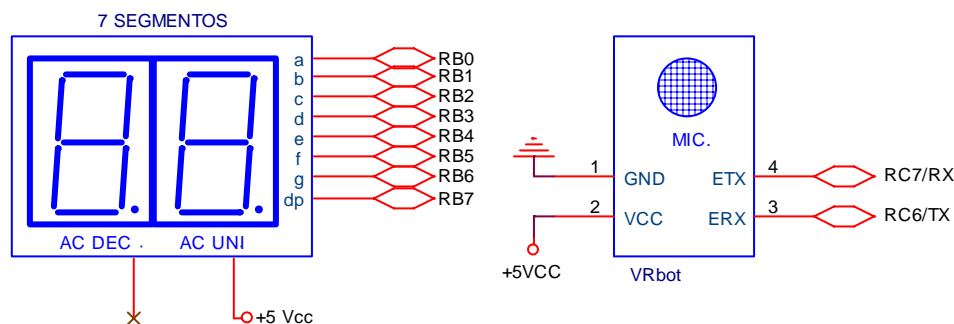


Figura 13. Esquema eléctrico del ejemplo 2

## Comentarios

Cuando se detecta una de las voces SD definidas por el usuario en el grupo 1, el programa extrae qué posición ocupa esa voz dentro de ese grupo. A partir de dicho número de posición se obtiene el código de 7 segmentos que hay que visualizar sobre el display. Si la voz no es reconocida el display se apaga. En la fotografía de la figura 14 se muestra el montaje sobre el laboratorio USB-PIC'SCHOOL. El resultado final es que sobre el display se visualiza el número de la voz dictada.

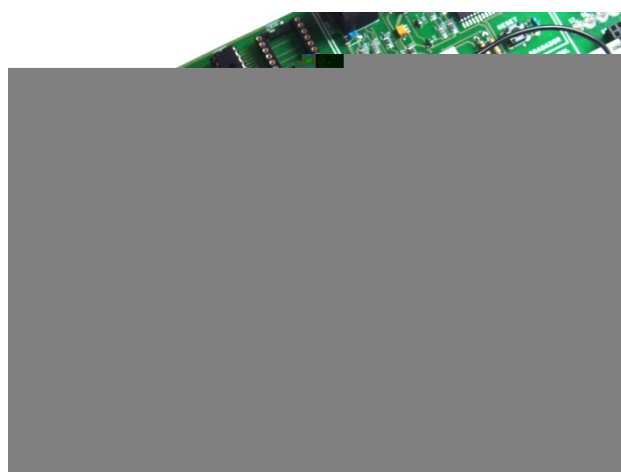


Figura 14. Montaje práctico del ejemplo 2

## 5.3 Ejemplo 3: Reconocimiento de voces SD definidas por el usuario y visualización de las etiquetas

### Objetivos

Este ejemplo reconoce una de las voces definidas por el usuario en el grupo 1. Sobre la pantalla LCD, además de visualizar el número que ocupa la voz dentro del grupo, se visualiza también la etiqueta o comando asociada a la misma.

### Esquema

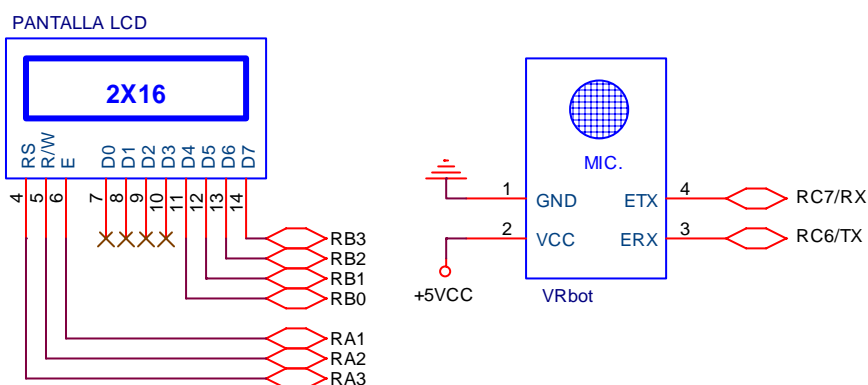


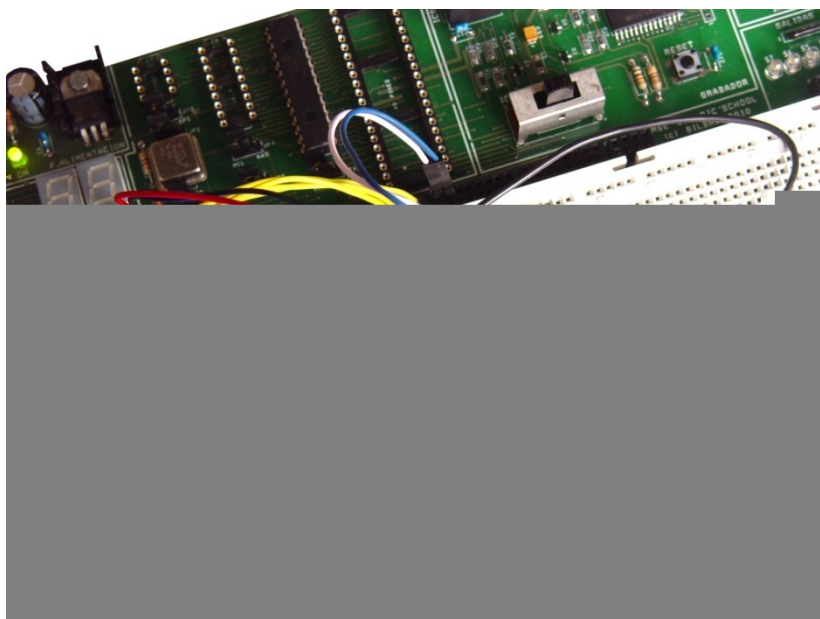
Figura 15. Esquema eléctrico

Se visualiza en la figura 15. La pantalla LCD se gobierna mediante un interface de 4 bits con las líneas RB3:RB0. Las señales de control de la misma las proporcionan RA1, RA2 y RA3 que se conectan con E, R/W y RS respectivamente.

El módulo VRbot se conecta como en anteriores ocasiones.

## Comentarios

La nueva aportación de este ejemplo es el extraer y visualizar el comando o etiqueta asociado a una voz. Como ya se ha explicado dicha etiqueta consiste en una cadena ASCII de caracteres imprimibles y que está asociada a una voz. Efectivamente, cuando se reconoce cualquiera de las voces SD definidas por el usuario en el grupo 1, el programa extrae y visualiza qué número de posición ocupa dentro del grupo. A partir de ese número también obtiene y visualiza la cadena ASCII asociada a esa voz y ese número. La fotografía de la figura 16 muestra el montaje práctico del ejemplo.



**Figura 16.** Montaje práctico del ejemplo 3 sobre el laboratorio USB-PIC'SCHOOL

Tanto las voces como sus correspondientes comandos o etiquetas asociadas se definieron según el apartado 2.3 de este documento. En cualquier caso hay que indicar que esa etiqueta NO tiene porqué coincidir con la voz.

## 5.4 Ejemplo 4: Reconocimiento de voces SD definidas por el usuario

### Objetivos

Este importante ejemplo pretende servir para mostrar cómo los programas de aplicación del usuario también pueden realizar ciclos de enseñanza de voces y grabarlas sobre el módulo VRbot junto con las etiquetas o comandos que queramos asociarlas. Todo ello lo vamos a realizar sobre el grupo 2 y con independencia total de un PC.

### Esquema

Se emplea el mismo esquema de conexiones que el empleado en la figura 15 del ejemplo anterior

## Comentarios

El ejemplo es un poco más complicado que los anteriores, pero merece la pena analizarlo con cuidado ya que puede servir de referencia para posteriores aplicaciones.

Empieza grabando las etiquetas o comandos sobre el grupo 2, en las posiciones 0 al 4 de ese grupo: ADELANTE, ATRÁS, DERECHA, IZQUIERDA y PARA. Seguidamente comienzan los ciclos de enseñanza gracias a los cuales a cada una de esas etiquetas se les asocia una voz (no tiene porqué ser la misma).

El ciclo de enseñanza de cada voz se repite dos veces para obtener una mejor definición de la misma. Sobre la pantalla LCD irán apareciendo una serie de mensajes intuitivos que irán informando de los diferentes pasos. Una vez grabadas las 5 etiquetas y sus voces, el programa queda bloqueado en un bucle cerrado. Posteriormente, podemos conectar el módulo VRbot a nuestro PC y, mediante el entorno VRbotGUI para Windows, ver el estado actual del módulo con las nuevas voces y etiquetas recién incorporadas. La figura 17 muestra el aspecto que debiera tener el contenido del módulo visto con el software VRbotGUI.



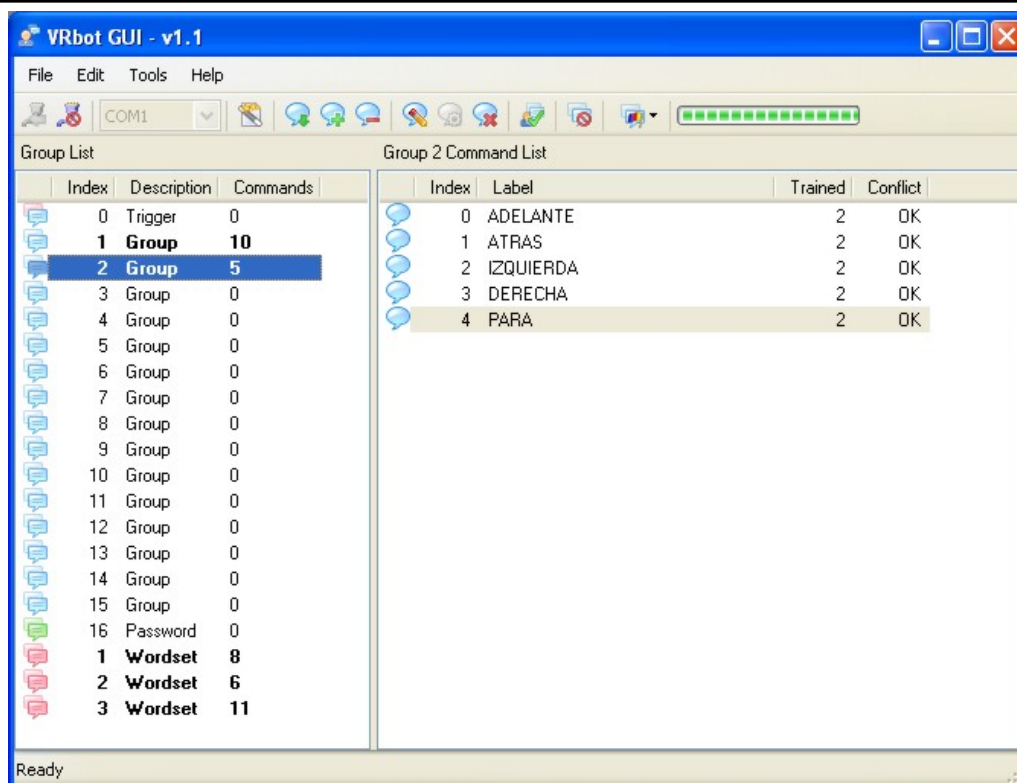


Figura 17. Estado actual del VRbot

Una vez hemos ejecutado este ejemplo podemos luego ejecutar el ejemplo 3 anterior para verificar que tanto las voces que acabamos de enseñar como sus etiquetas son debidamente reconocidas. En ese ejemplo 3 tan solo habrá que modificar el número de grupo (Grupo 2).

## 5.5 Ejemplo 5: Reconocimiento de voces SD definidas por el usuario

### Objetivos

Se trata de reconocer las voces que se definieron y grabaron en el ejemplo anterior sobre el grupo 2. En la pantalla LCD se visualizará la etiqueta y la posición dentro del grupo 2 en que se encuentre cualquiera de las voces reconocidas. Además, para dar un mayor interés, se supone que se está controlando un sistema móvil que reproducirá los movimientos según la voz dictada.

### Esquema

Se muestra en la figura 18. Tanto la pantalla LCD como el VRbot se conectan como en anteriores ocasiones. Las líneas RC0:RC3 se conectan con los leds de salida S0:S3

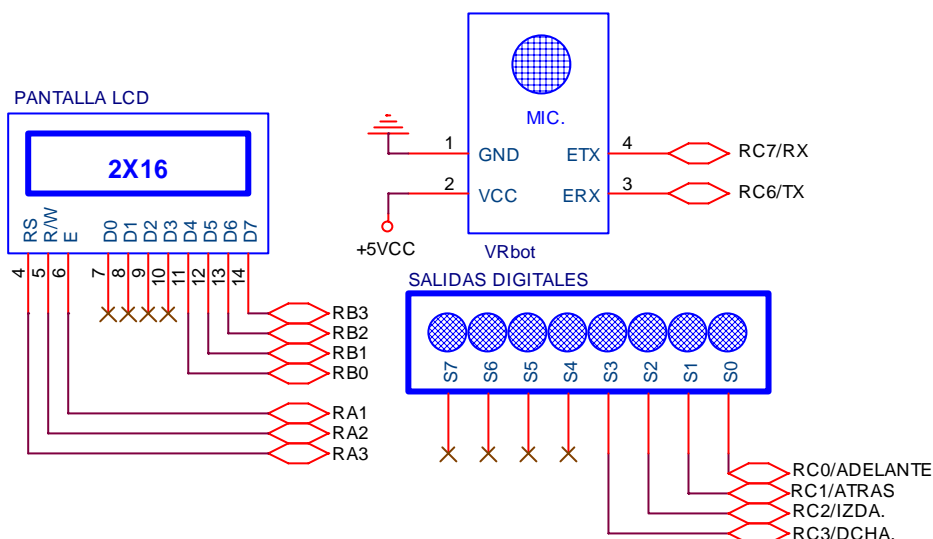
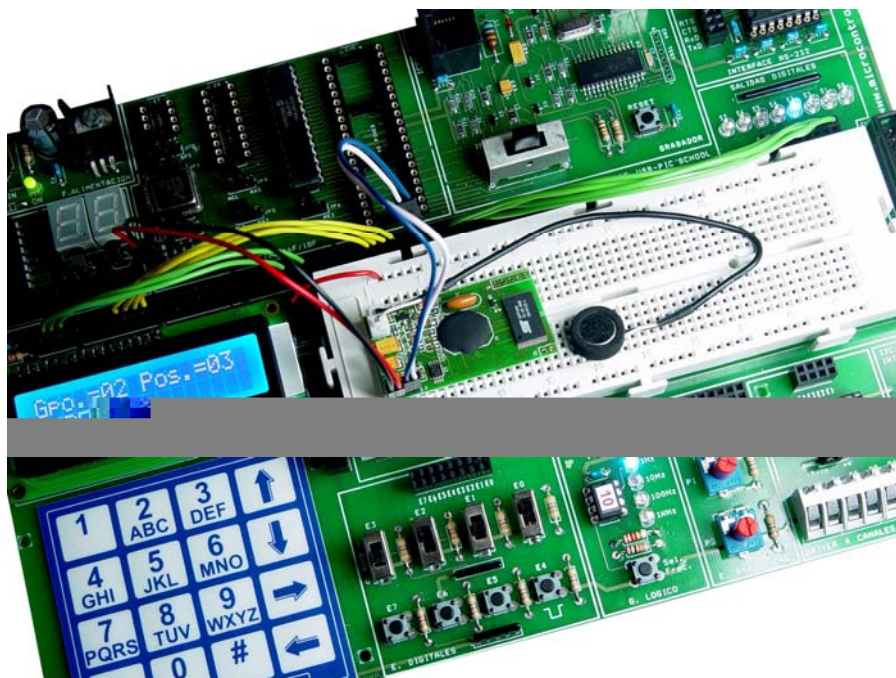


Figura 18. Ejemplo 5

## Comentarios



**Figura 19.** Montaje práctico del ejemplo 5

Con objeto de dar una sencilla idea de aplicación, el ejemplo propone controlar 4 salidas que simulan otros tantos movimientos de un supuesto sistema móvil (p.e. un robot). Para ello las líneas RC0:RC3 conectadas a las salidas S0:S3 simulan activar los mecanismos que reproducirán los movimientos hacia adelante, atrás, giro a la derecha, giro a la izquierda y parada.

Cuando se reconoce una de las voces del grupo 2 (grabadas en el ejemplo anterior), la pantalla LCD visualiza la etiqueta o comando correspondiente y su posición dentro del grupo, al tiempo que se activará el led correspondiente. En la fotografía de la figura 19 se muestra el montaje práctico realizado sobre el laboratorio USB-PIC'SCHOOL.

## 5.6 Ejemplo 6: Reconocimiento de voces SD definidas por el usuario

### Objetivos

El ejemplo trata de reconocer voces SD definidas por el usuario en diferentes grupos. Efectivamente, se combinan las voces del grupo 2 (movimiento) con las del grupo 1 (números 0 al 9) con objeto de realizar una aplicación en la que un sistema móvil realiza un movimiento determinado durante un cierto tiempo. Todo ello controlado por la voz.

### Esquema

Es el mismo esquema que el de la figura 18 del ejemplo anterior.

### Comentarios

El programa empieza tratando de reconocer las voces de movimiento del grupo 2: ADELANTE, ATRÁS, DERECHA, IZQUIERDA y PARO. Cuando una de ellas es reconocida, se visualiza en el LCD y se determina qué tipo de movimiento hay que realizar.

Seguidamente trata de reconocer las voces numéricas del grupo 1 (del 0 al 9). Cuando se reconoce cualquiera de ellas se visualiza en el LCD de forma concatenada con la voz de movimiento (p.e. DERECHA NUEVE). Esta voz numérica representará tiempo en segundos.

Efectivamente, cuando se han reconocido ambas voces, la de movimiento y la de tiempo, se procede a realizar la maniobra. Se activa la salida RC0:RC3 correspondiente durante el tiempo establecido en segundos.

La figura 20 muestra el montaje del ejemplo sobre el laboratorio y su ejecución.

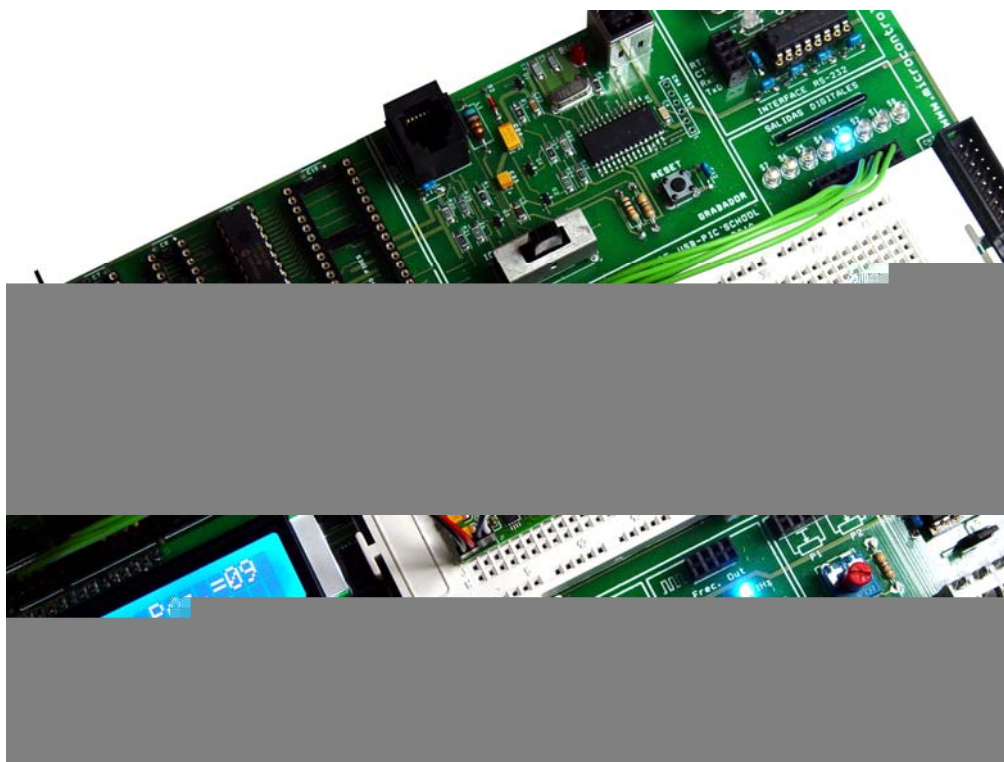


Figura 20. Montaje práctico del ejemplo 6

## 5.7 Ejemplo 7 : Reconocimiento de voces SD definidas por el usuario

### Objetivos

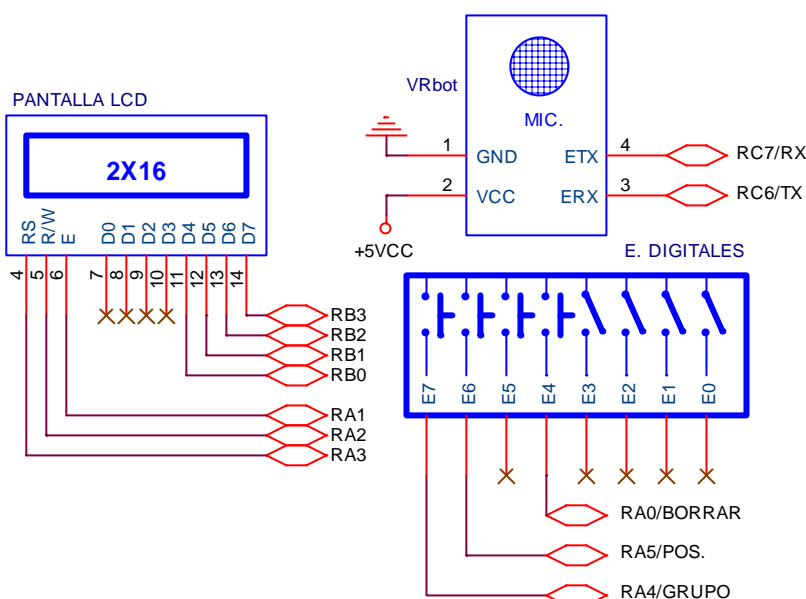
Se trata de un ejemplo meramente didáctico en el que se pretende mostrar algunas de las funciones contenidas en la librería <VRbot\_16FXXX>. De una forma intuitiva, gracias a la pantalla LCD, podremos explorar el contenido interno del módulo VRbot y visualizar qué grupos están ocupados, cuántas voces poseen, que posición ocupan dentro de un grupo y la etiqueta o comando correspondiente.

### Esquema

Se muestra en la figura 21. Tanto la pantalla LCD como el módulo VRbot se conectan como se ha venido realizando en anteriores ejercicios. Los pulsadores E7, E6 y E4 se conectan con las líneas RA4, RA5 y RA0 respectivamente.

RA4 permite seleccionar secuencialmente los grupos que contengan alguna voz. RA5 selecciona la posición dentro de un grupo. RA0 elimina la etiqueta y la voz que haya en la posición y grupo actual.

Figura 21. Esquema de conexiones



## Comentarios

El ejemplo pretende mostrar el empleo de algunas de las funciones contenidas en la librería "VRbot\_16FXXX.inc" y que no han sido usadas hasta ahora:

- VR\_MASK\_SD: Obtiene la máscara con los grupos de voces SD no vacíos
- VR\_COUNT\_SD: Obtiene el nº de comandos y voces de ese grupo
- VR\_ERASE\_UNG\_SD: Borra la voz y etiqueta asociada en la posición y grupo actual

Con cada transición 1-0-1 en RA4 (aplicada desde el pulsador E7 de USB-PIC'School) se selecciona el siguiente grupo que contenga una o varias etiquetas (no vacío)

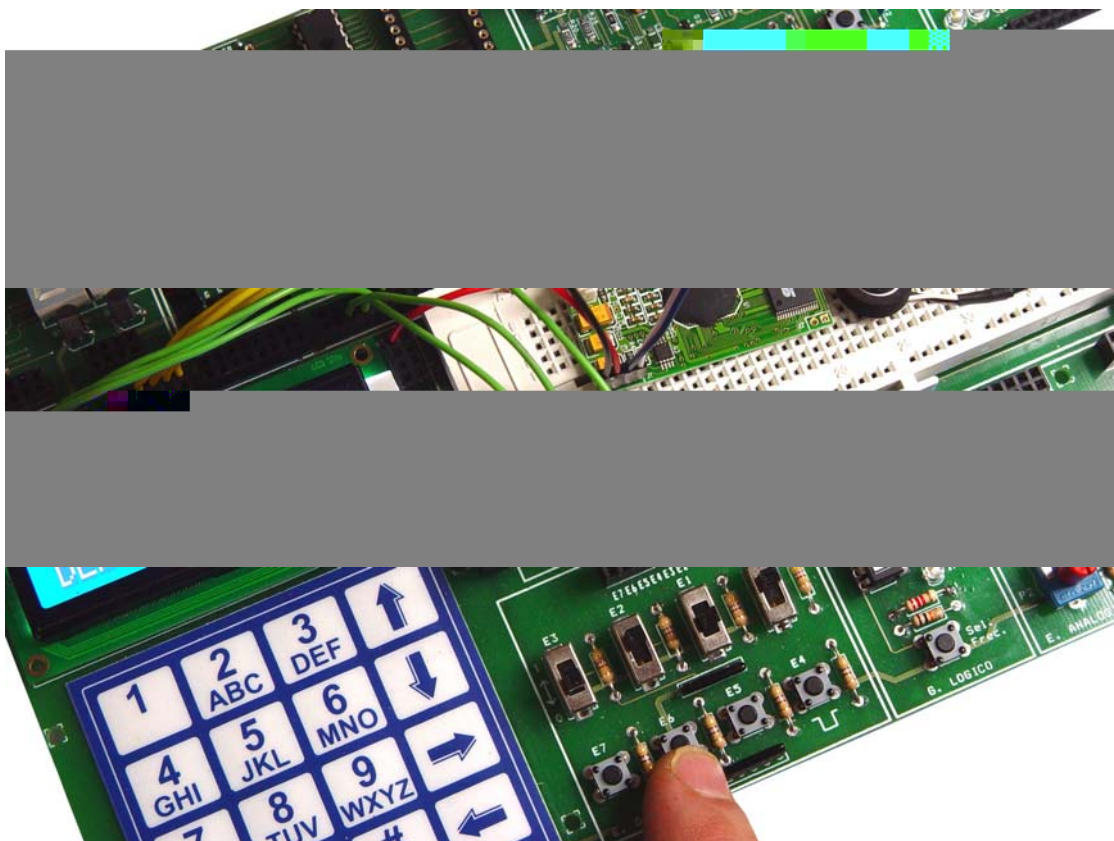
Con cada transición 1-0-1 en RA5 (aplicada desde el pulsador E6) se selecciona la siguiente posición dentro del grupo.

En la pantalla LCD se visualizará el nº de grupo actual (no vacío), el nº de etiquetas contenidas en ese grupo, posición de la etiqueta dentro del grupo y la etiqueta propiamente dicha.

Con una transición 1-0-1 en RA0 (aplicada desde el pulsador E4) se elimina, del módulo VRbot, la etiqueta y voz que en ese momento se esté visualizando en la pantalla LCD

Podemos emplear el entorno VRbot GUI de windows para crear nuevos grupos y etiquetas que luego serán visualizadas y/o borradas mediante la ejecución de este ejemplo.

En la figura 22 se muestra el montaje práctico de este ejemplo con el resultado de su ejecución.



**Figura 22.** Montaje práctico del ejemplo 7