Parallax Say It Module (#30080)

The Parallax Say It Modules provides voice recognition functions for built-in Speaker Independent (SI) pre-programmed commands and up to 32 user-defined Speaker Dependent (SD) keywords (triggers, commands, or passwords).

When you speak into this module, it will match the spoken word to a set of keywords that it has been programmed to recognize. Once the module has determined if there is a match, it will take a defined action, either listening for the next keyword in another "wordset" or executing the commands associated with the word that was said. You can create up to 32 user-definable keywords.

The Say It GUI software for the BASIC Stamp 2 provides an easy interface for training the module and producing template code. Or, the simple and robust serial protocol provided can be used to access the Say It module functions from other Parallax microcontrollers. The 10-pin SIP header makes the module breadboard friendly, and is designed to fit in one row of the AppMod header found on the Board of Education and Boe-Bot Robot.

Features

- 23 Pre-programmed commands
- Up to 32 user-definable commands
- SIP for breadboard friendly projects (0.1" spacing)
- GUI provides training and template code for BASIC Stamp 2 modules
- On-board LED and microphone
- Voice controlled Boe-Bot examples

Key Specifications

- Power requirements: 3.3 to 5.5 VDC
- Communication: Adjustable Asynchonous Serial (9600 (default), 19200, 38700, 57600, 115200)
- Operating temperature: 32 to 158 °F (0 to 70 °C)
- Dimensions: 1.02 x 2.47 x .38 in (26 x 62.93 x 9.70 mm)

Application Ideas

- Voice-controlled entry systems
- Automated house applications
- Voice-activated robotics

Precaution

- Do not solely rely on the Say It module to recognize a command for a safety stop if your project requires one; take all appropriate precautions when implementing this module to maintain a safe project.

Using the Say It GUI Software

With the Say It GUI software for your PC and your BASIC Stamp 2 development board, you can test the Say It module and train it to recognize your custom commands. During training, the BASIC Stamp 2 handles the Say It module-to-PC communication through the provided PBASIC "bridge" program. Once your you have defined and tested your commands, the GUI software will generate a new PBASIC template program ready for you to add the actions to take when your voice commands are received.

Follow the steps below to connect to the Say It module via the GUI software. This example assumes you are using a Board of Education with BASIC Stamp 2, and you have previously installed the BASIC Stamp Editor and tested the programming connection.

1. With the power to your board turned off (switch position 0) Plug the Say It module into the AppMod header of the Board of Education (as seen in Figure 12); be careful to insert the module in the left row of the header and in the correct orientation (Vss at top, Vdd at bottom, RX to P0, TX to P2 and LED to P4).

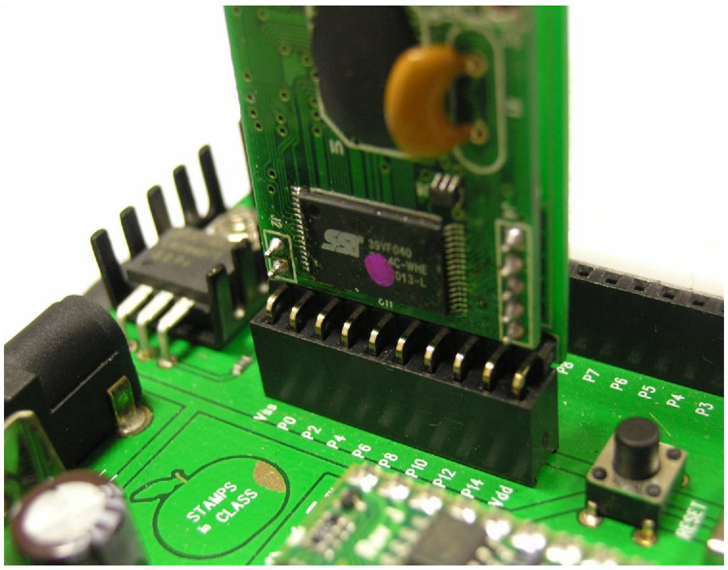


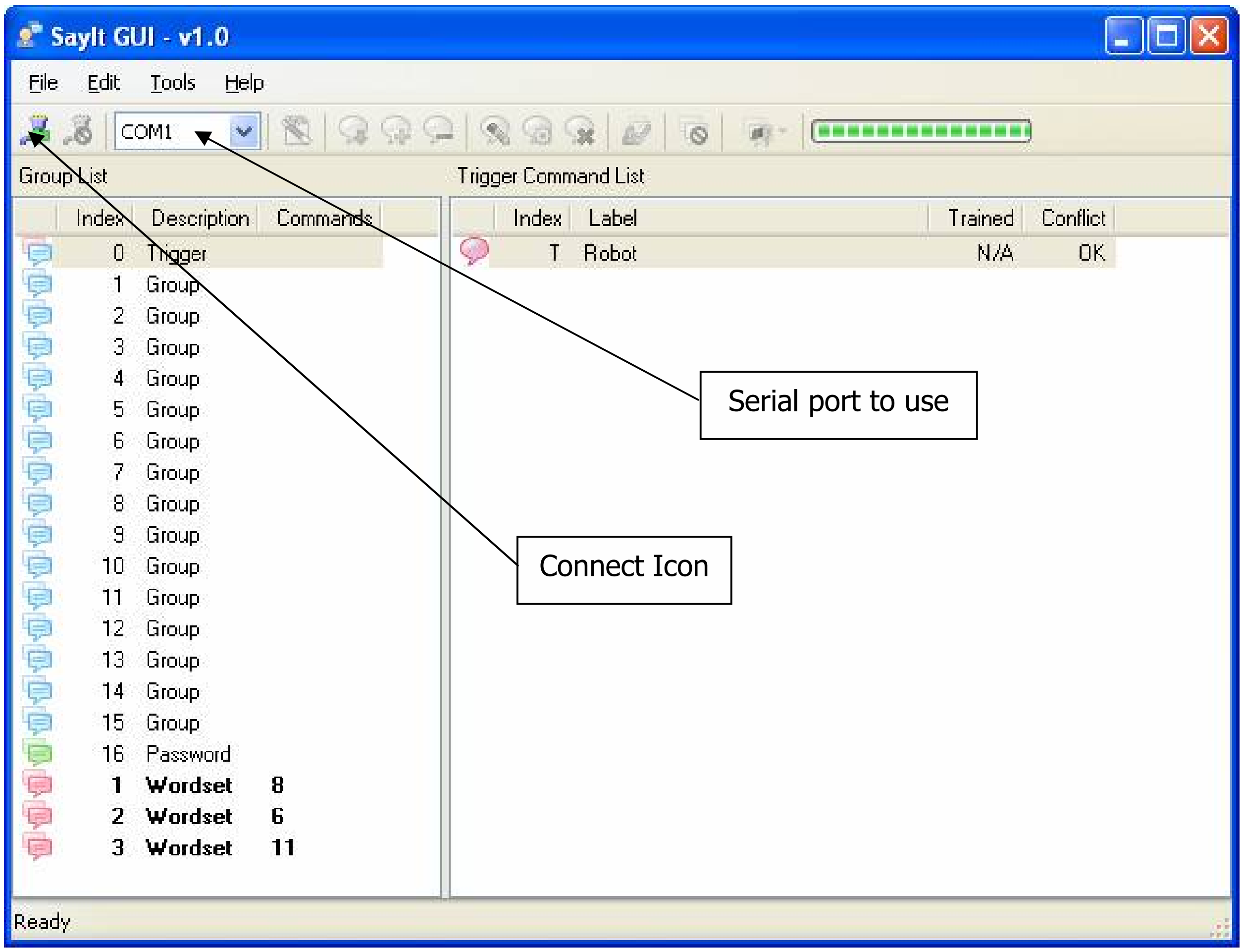
Figure 1

2. Download and install the Say It GUI software from the 30080 product page at www.parallax.com. Use the default installation path. For Windows Vista users, install as administrator.

3. Start the Say It GUI software, and then connect the Board of Education to your PC and turn the power switch on (position 1).

4. Select the serial port that the Board of Education/Say It module is connected from the toolbar (Figure 1) or File from the menu, then Connect. See Figure 2.

BASIC Stamp Editor Debug Terminal must be closed before selecting "Connect" in Say It GUI

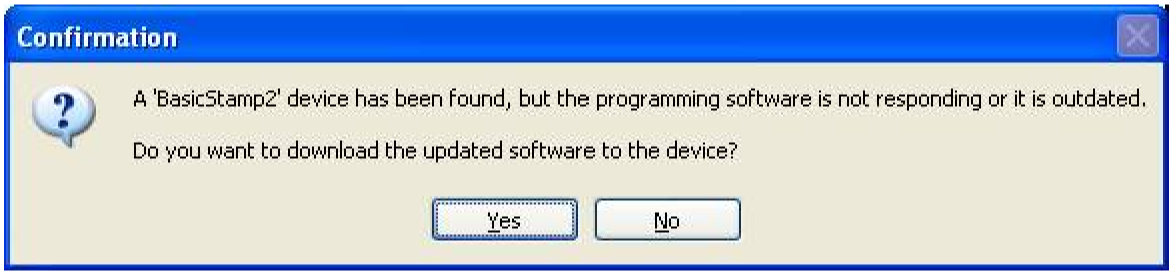


Index	Description	Commands
0	Trigger	
1	Group	
2	Group	
3	Group	
4	Group	
5	Group	
6	Group	
7	Group	
8	Group	
9	Group	
10	Group	
11	Group	
12	Group	
13	Group	
14	Group	
15	Group	
16	Password	
1	**Wordset**	**8**
2	**Wordset**	**6**
3	**Wordset**	**11**

Index	Label	Trained	Conflict
T	Robot	N/A	OK

Figure 2

5. Once connected, the Say It software prompts you to download the PBASIC "bridge" program to the controller board, and switches to the programming mode (Figure 3). Choose Yes when prompted.



A 'BasicStamp2' device has been found, but the programming software is not responding or it is outdated.

Do you want to download the updated software to the device?

Figure 3

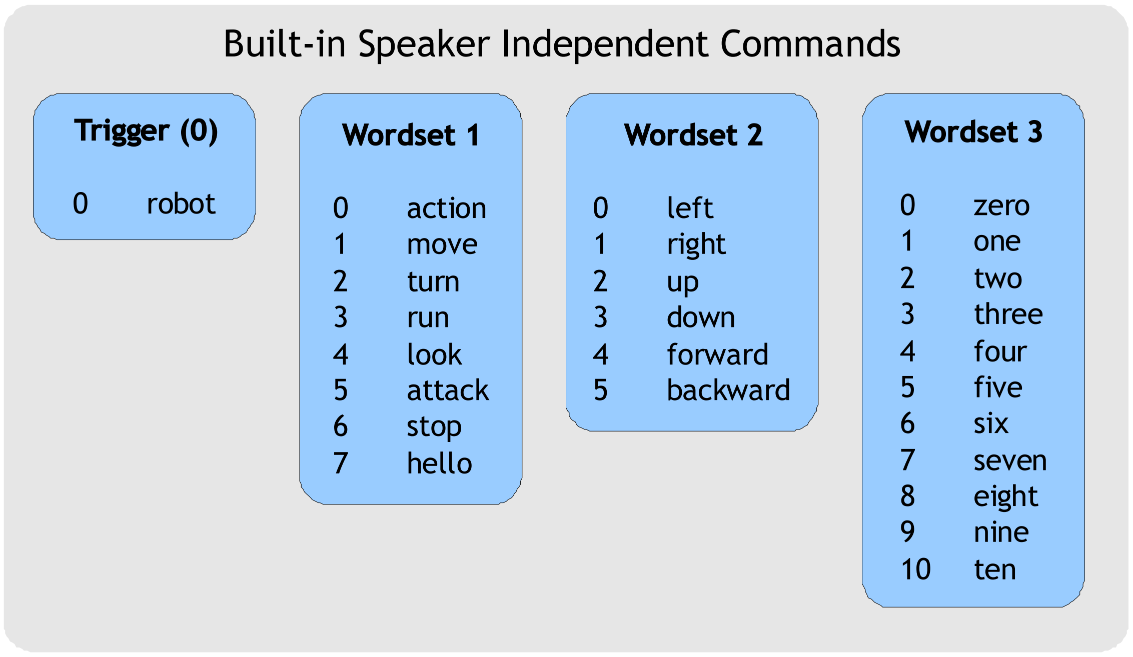
A PBASIC "bridge" program will automatically be downloaded to the BASIC Stamp 2. This bridge program allows the user to work with the set of SI commands the Say It module provides, as well as defining new commands.

6. Verify that the bridge download has been completed by the green status bar in the top right of the GUI; it should remain full.

Once you have successfully connected to the module you can insert, add, remove, rename, train, erase, test, reset all the commands, set the language used, or disconnect from the GUI. First, let's cover testing the pre-existing commands.

Testing Commands

Let's begin by testing the words that are already programmed in the Say It module. These are grouped under Trigger and three Wordsets (Figure 4).



Built-in Speaker Independent Commands

Trigger (0)	
0	robot

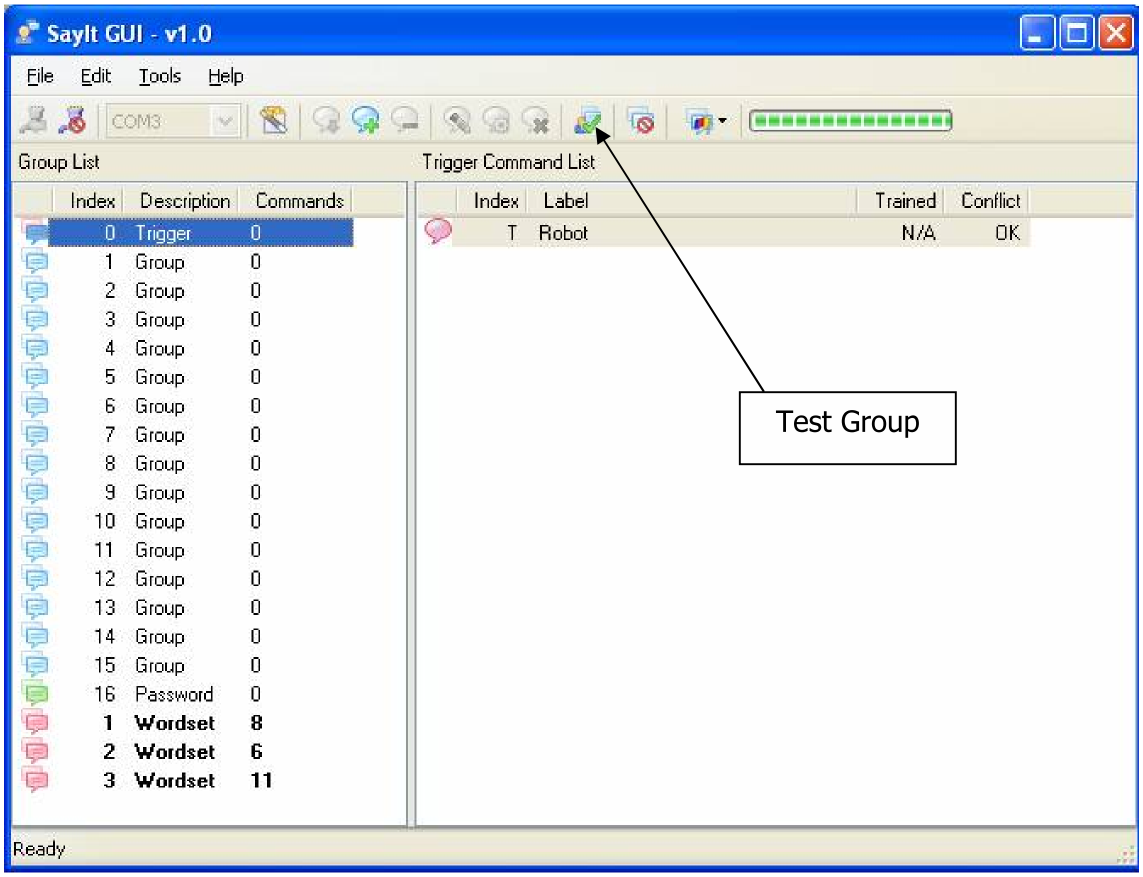
Wordset 1	
0	action
1	move
2	turn
3	run
4	look
5	attack
6	stop
7	hello

Wordset 2	
0	left
1	right
2	up
3	down
4	forward
5	backward

Wordset 3	
0	zero
1	one
2	two
3	three
4	four
5	five
6	six
7	seven
8	eight
9	nine
10	ten

Figure 4

1. Select a Trigger or Wordset to test by highlighting the option in the left window pane (Figure 5) and then click "Test Group" from the tool bar. This example, I chose "Trigger" to test.



Index	Description	Commands
0	Trigger	0
1	Group	0
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	**Wordset**	**8**
2	**Wordset**	**6**
3	**Wordset**	**11**

Index	Label	Trained	Conflict
T	Robot	N/A	OK

Figure 5

2. When the red LED indicator light on the module and the software window prompt you to speak, speak clearly and directly at the microphone on the module. If the module understands, you will see the command highlighted in green.

You can continue this with all the words that need to be tested. If the module does not understand the word or there is nothing said, an information window will pop up indicating an error of a timeout. Later you will want to use the same process to test any new commands that you train it to recognize.

Adding or Deleting Commands

When you want to create your own command, you can do so by using the Say It GUI. There are 4 types of commands in the GUI:

- Trigger – Trigger words are used to start the voice recognition process; all spoken command phrases will begin with a trigger word. "Robot" is the SI trigger word, and you may train one additional trigger word.
- Group – Groups of user-definable SD commands. You may add up to 32 commands total (31 if you also define a trigger word).
- Password – A special group for "vocal passwords," up to 5 may be defined.
- Wordset – Built-in groups of Speaker Independent (SI) commands (Figure 3)

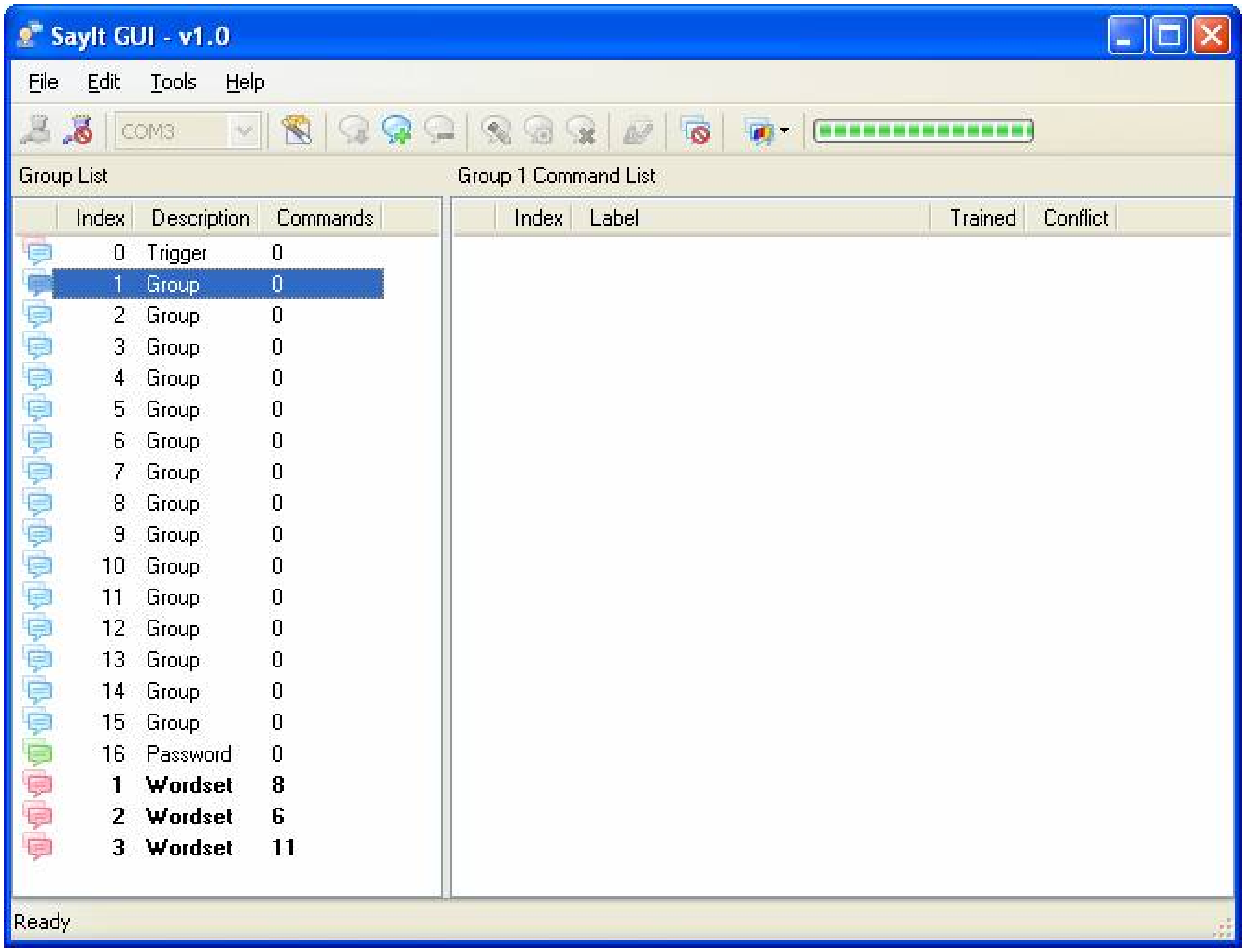
The user can define groups of SD commands or passwords and generate a PBASIC code template to handle them. The recognition function of Say It modules works on a single group at a time, so that users need to group together all the commands that they want to be able to use at the same time.

When Say It GUI connects to the module, it reads back all the user-defined commands and groups, which are stored into the Say It module's non-volatile memory for later review and editing.

When training SI commands, simulate the environmental background noise in which you want to use this module for the best results for recognition.

Adding a SD command can be completed by doing the following while the Say It "bridge" program is running.

1. Select a group that you would like to add the word(s) to (Figure 6).

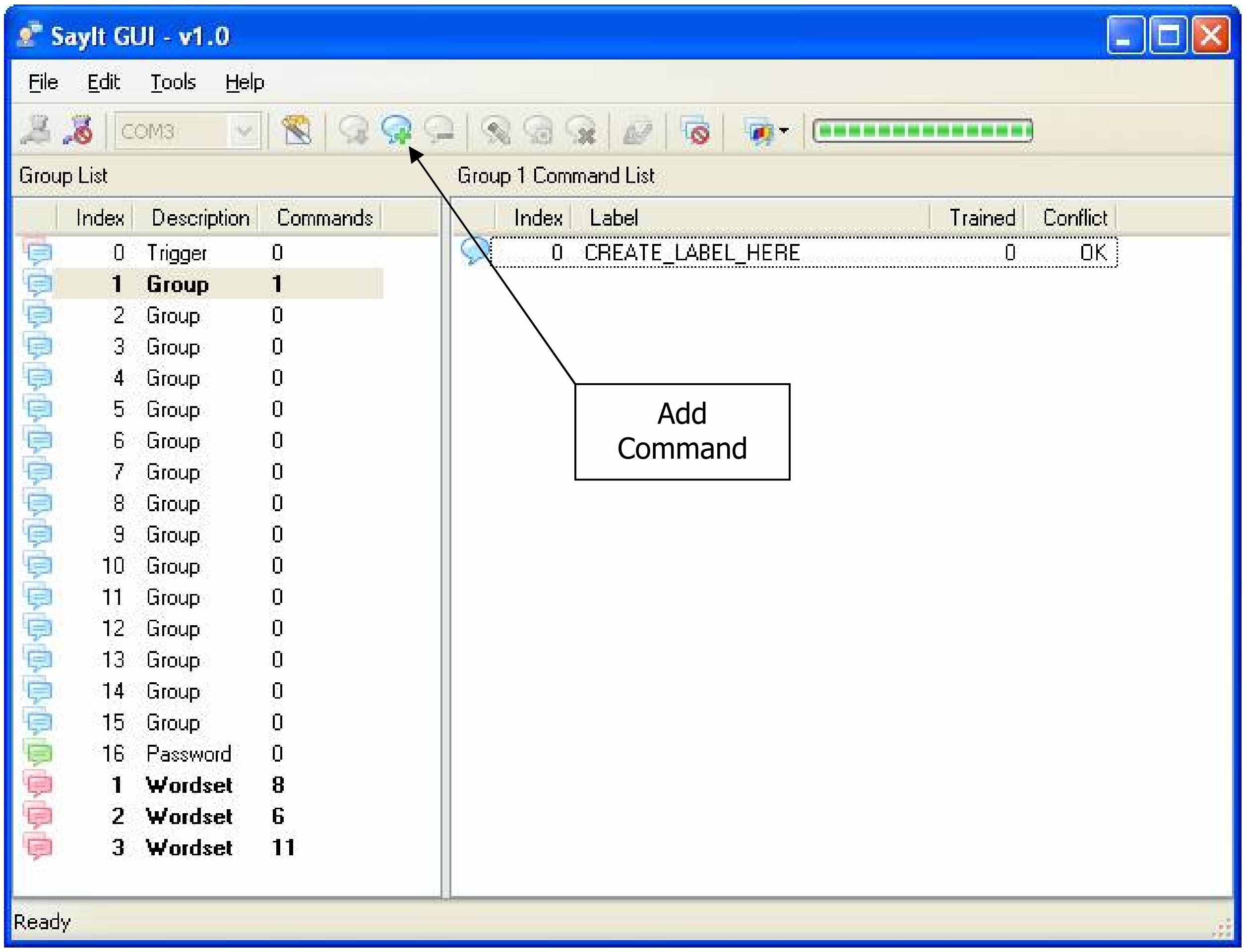


Index	Description	Commands
0	Trigger	0
1	Group	0
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	**Wordset**	**8**
2	**Wordset**	**6**
3	**Wordset**	**11**

Index	Label	Trained	Conflict

Figure 6

2. Click "Add Command" from the tool bar or menu (Figure 7), and provide a label. In this example, the label "CREATE_LABEL_HERE" has been created; however it is suggested that you use a label that you can later review and know what the word is.



Group List

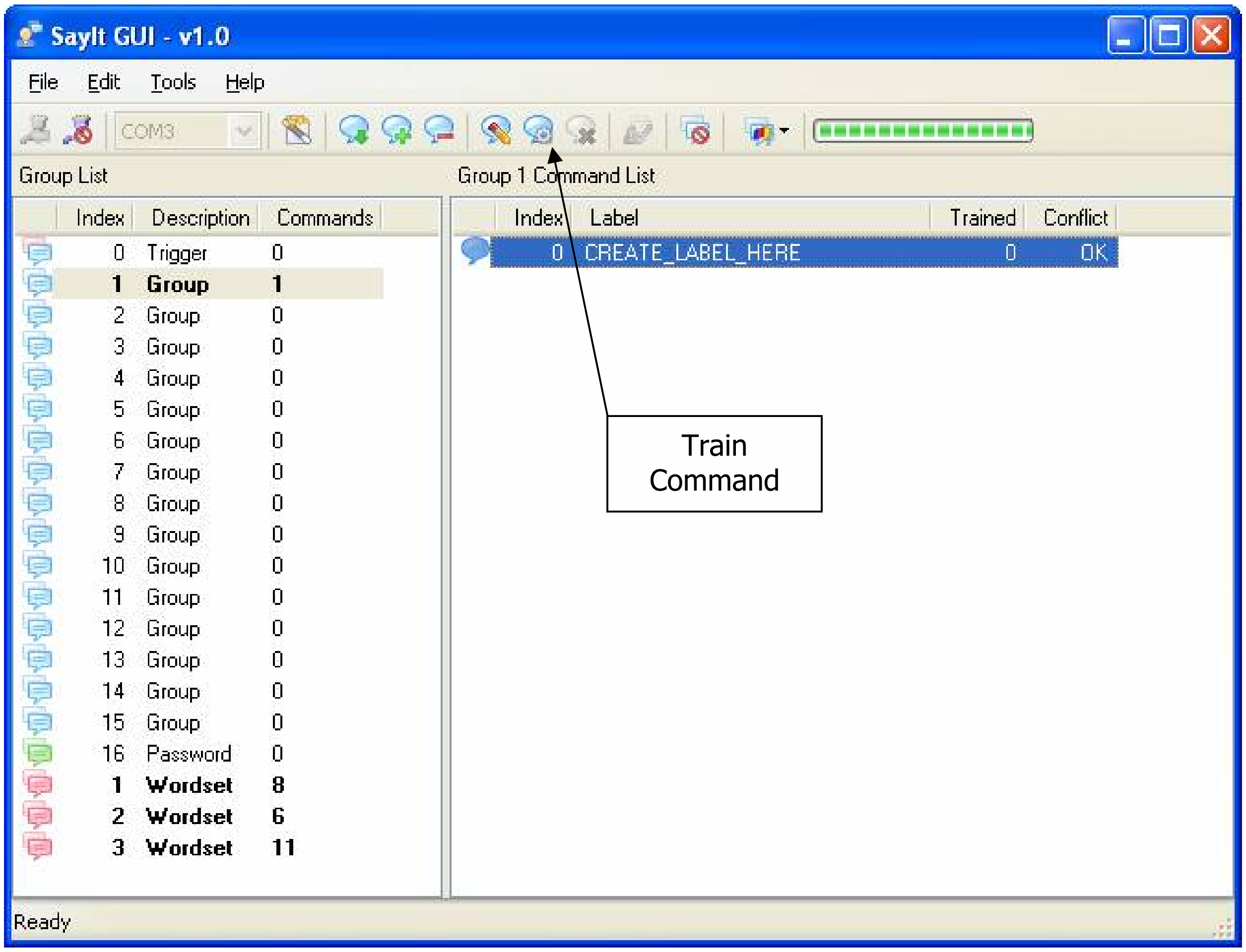
Index	Description	Commands
0	Trigger	0
1	Group	1
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	Wordset	8
2	Wordset	6
3	Wordset	11

Group 1 Command List

Index	Label	Trained	Conflict
0	CREATE_LABEL_HERE	0	OK

Figure 7

3. Select the label that in the right window pane, and click "Train" from the tool bar or menu (Figure 8).

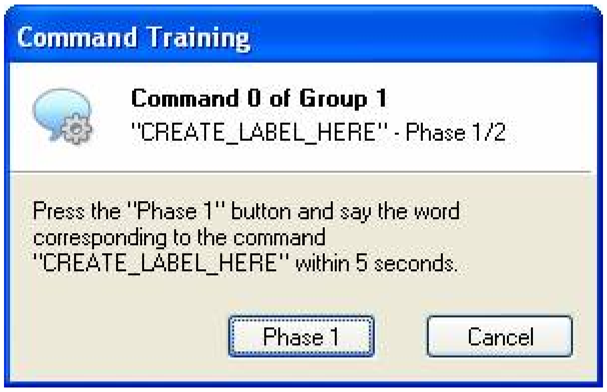


Index	Description	Commands
0	Trigger	0
1	Group	1
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	Wordset	8
2	Wordset	6
3	Wordset	11

Index	Label	Trained	Conflict
0	CREATE_LABEL_HERE	0	OK

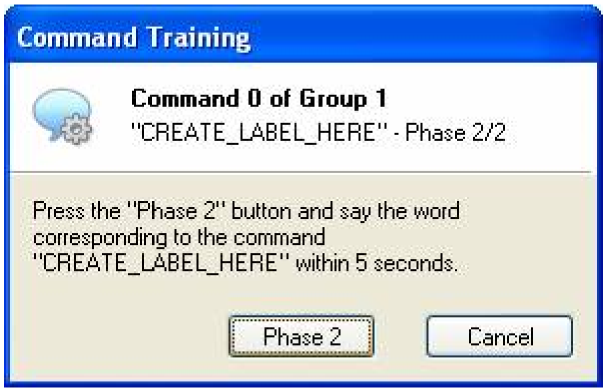
Figure 8

4. Once you have selected Train Command; you will be prompted to say the phrase twice (figure 8) to complete the training of a specific word; keep the words simplistic for optimal recognition. If you are unhappy with the training, select erase training, and start the training process over from step 3 until satisfied.



Command 0 of Group 1
"CREATE_LABEL_HERE" - Phase 1/2

Press the "Phase 1" button and say the word corresponding to the command "CREATE_LABEL_HERE" within 5 seconds.

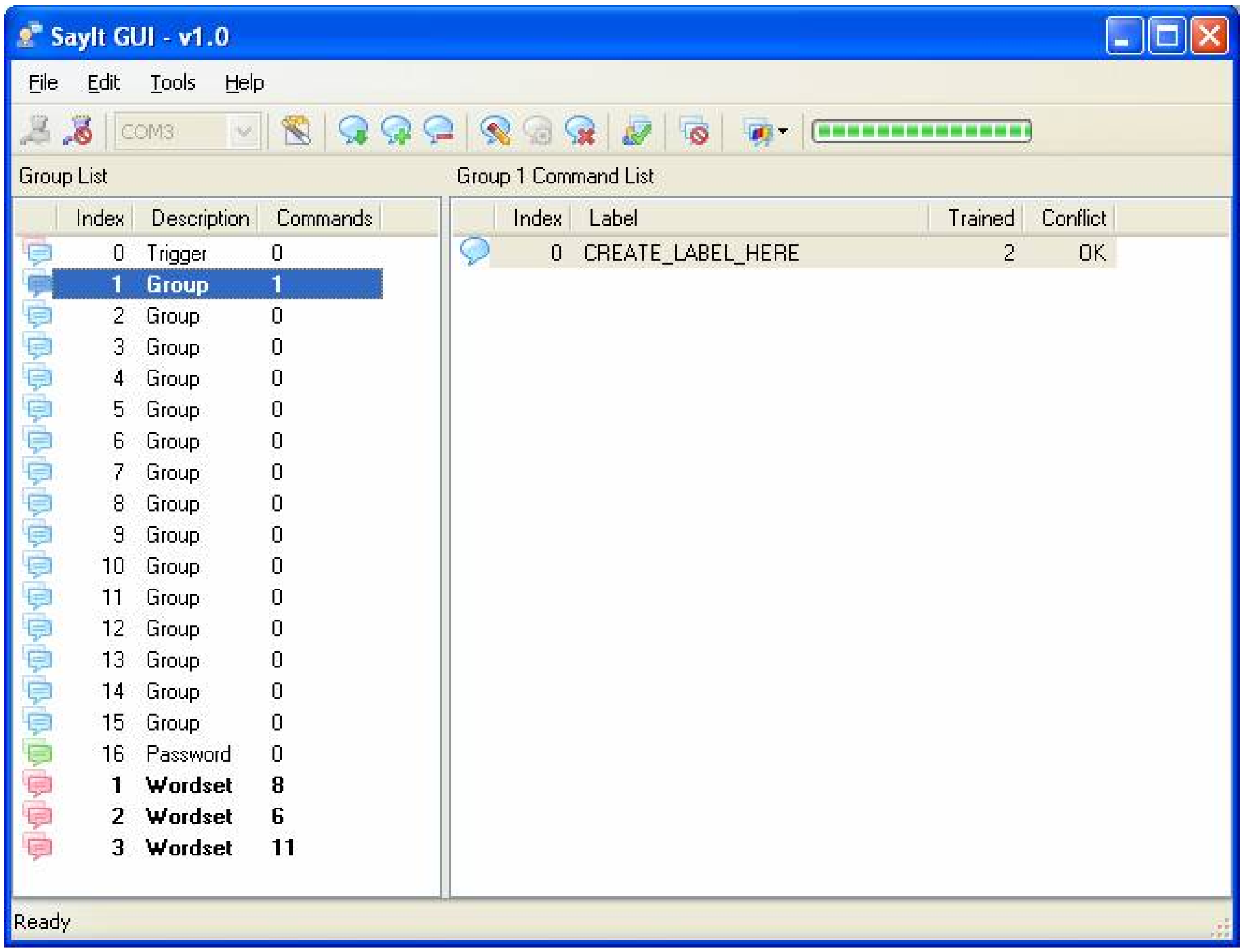


Command 0 of Group 1
"CREATE_LABEL_HERE" - Phase 2/2

Press the "Phase 2" button and say the word corresponding to the command "CREATE_LABEL_HERE" within 5 seconds.

Figure 9

5. Once you have successfully created a phrase, you can test to confirm that it will recognize it, it is suggested that you test each group after you are finished to ensure successful training. Once finished you will see a number next to the group you trained; indicating how many words belong to that group (Figure 9).



Group List

Index	Description	Commands
0	Trigger	0
1	Group	1
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	Wordset	8
2	Wordset	6
3	Wordset	11

Group 1 Command List

Index	Label	Trained	Conflict
0	CREATE_LABEL_HERE	2	OK

Ready

Figure 10

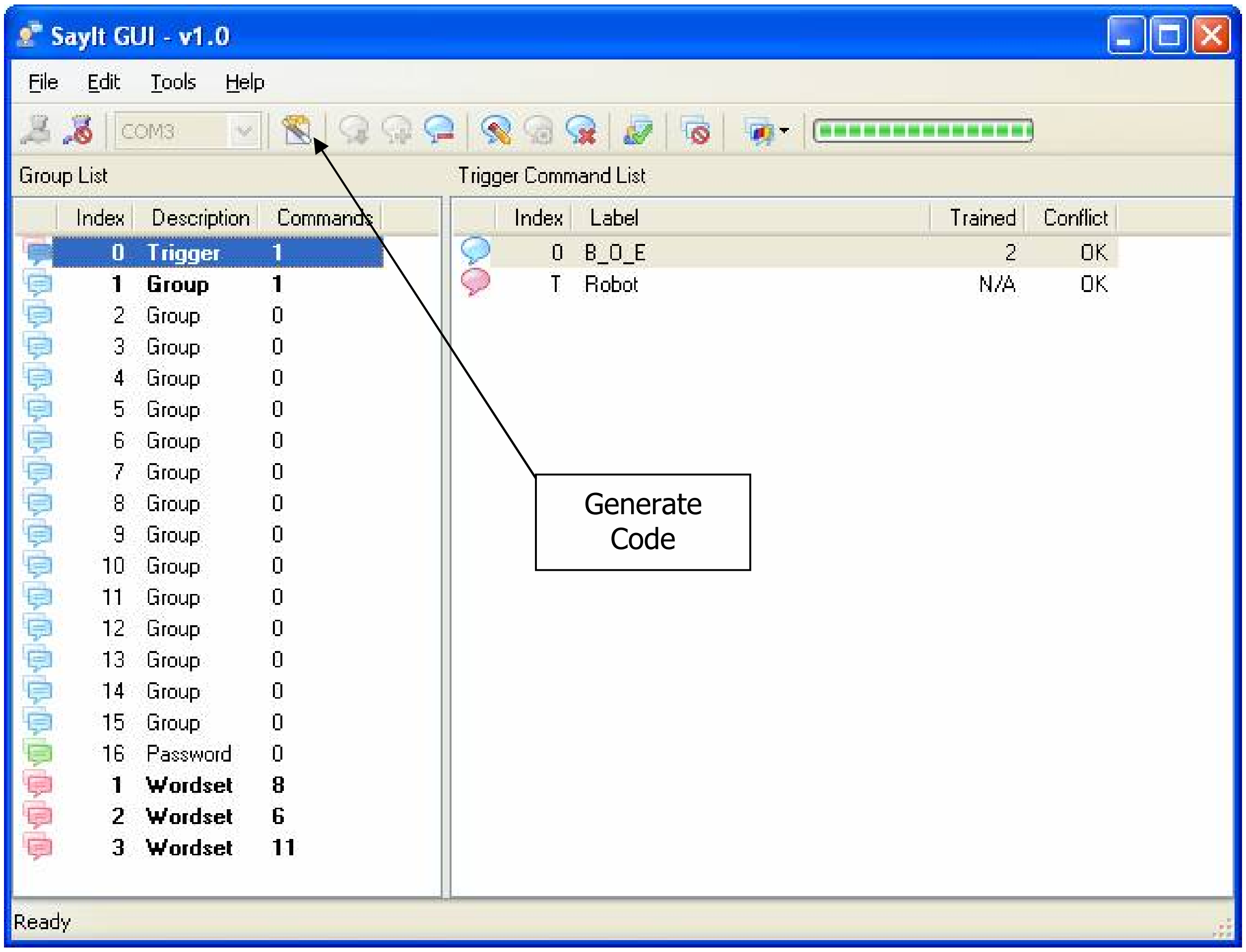
If you want to remove a command, you can use the "remove command" from the tool bar or menu and it will remove the selected command; once this is done it can not be undone so be sure you want to remove a command prior to clicking this action.

Each of the Group, Password, and Trigger words are created and edited in the same manner that these steps cover. Note: The Passwords (group 16) are much more sensitive to background environment noises and distance from the microphone; but sure to train the password in conditions similar to where it will be used.

Generating Code

Once you have created and trained all your desired commands, you can generate the PBASIC code to then edit and assign actions to each of the words created. You can do that be completing the following:

1. Select the "Generate Code…" icon on the toolbar or from the menu (Figure 11)



Group List

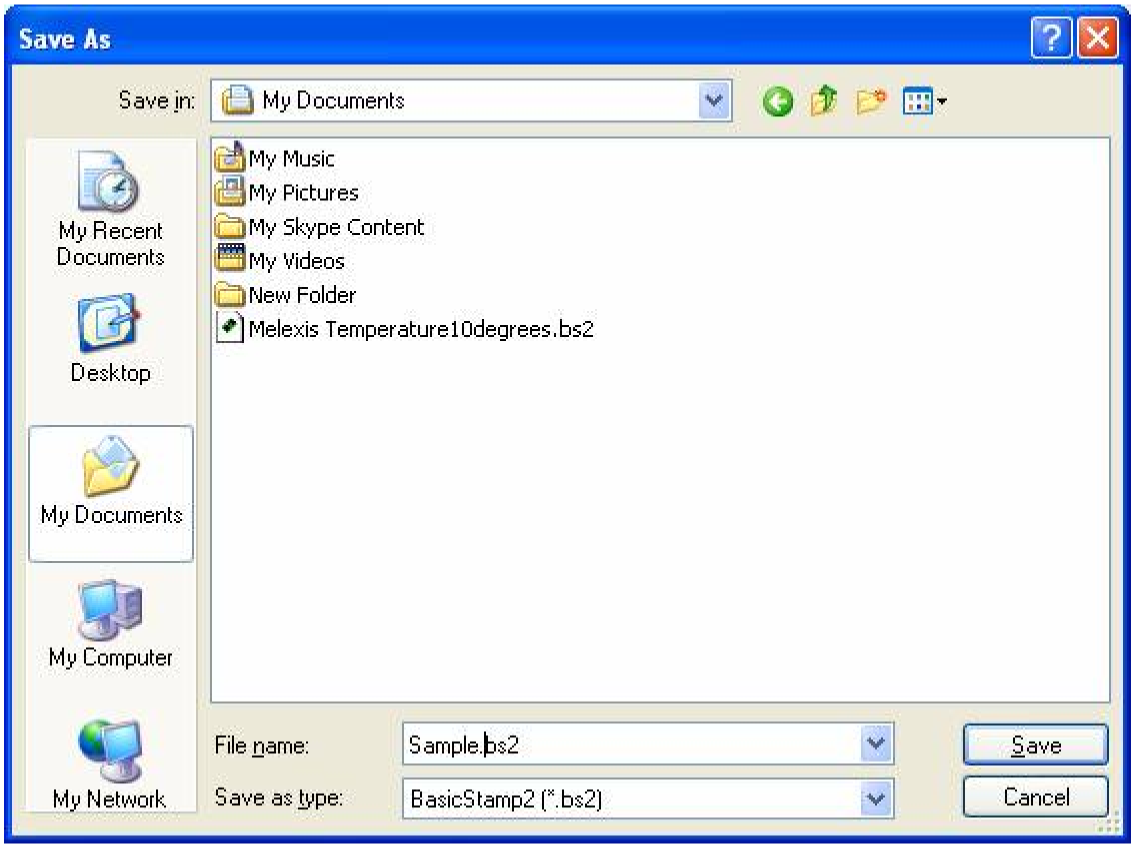
Index	Description	Commands
0	Trigger	1
1	Group	1
2	Group	0
3	Group	0
4	Group	0
5	Group	0
6	Group	0
7	Group	0
8	Group	0
9	Group	0
10	Group	0
11	Group	0
12	Group	0
13	Group	0
14	Group	0
15	Group	0
16	Password	0
1	Wordset	8
2	Wordset	6
3	Wordset	11

Trigger Command List

Index	Label	Trained	Conflict
0	B_O_E	2	OK
T	Robot	N/A	OK

Figure 11

2. You will be prompted to save the file to then edit within the BASIC Stamp Editor (Figure 12).



My Music
My Pictures
My Skype Content
My Videos
New Folder
Melexis Temperature10degrees.bs2

File name: Sample.bs2
Save as type: BasicStamp2 (*.bs2)

Figure 12

3. Click "Disconnect" in the GUI and open the file with the BASIC Stamp Editor.

4. Once the program is opened in the BASIC Stamp Editor, there will be portions of the code that will indicate where you will place the commands that will be used with the trained words. You will see a PAUSE 1 with "'-- write your code here" comments.

5. Save your program, and then download to the BASIC Stamp 2 module and enjoy playing with your new voice recognition module.

Sample Application for the Boe-Bot® Robot

Here is the sample applications that uses the Say It module to control a Boe-Bot robot with a BASIC Stamp 2 on a Board of Education platform. The sample code for this application is available for download on the Say It Module product page at www.parallax.com.

1. Plug the Say It module into the AppMod header of the Board of Education (as seen in Figure 1 on page 2); be careful to insert the module in the left row of the header and in the correct orientation (Vss at top, Vdd at bottom, RX to P0, TX to P2 and LED to P4).

2. Open the sample code labeled "SayIt_Demo.BS2" in the BASIC Stamp Editor.

3. Install any batteries as needed, plug in the battery pack, and move the Board of Education power switch to position 1

4. Download the program to the BASIC Stamp 2 module by clicking Run from the menu, and click Run from the dropdown (ctrl + r)

5. Move the power switch to position 0, and unplug the communication cable; then move the power switch to position 2.

6. Using the command list above (Figure 3); Say the trigger word (robot), and select then select a word from Wordset 1, 2 and then 3 if needed. You can verify the word has been correctly recognized by the red LED indicator on the Say It module.

When you say "robot", the red LED will turn on for a short moment; once the LED is on, you can say the next word. Once the Say It module has received the last Wordset command, it will execute the proper routine associated with that command. Here are some samples that could be used and the descriptions of the actions.

Try saying the following examples:

Robot -> Move -> Forward	(This will move the robot forward)
Robot -> Hello	(Module will say hello on the debug screen, if one is open)
Robot -> Action -> Three	(Module will display 3 on debug screen, if one is open)
Robot -> Turn -> Right	(This will turn the robot right)
Robot -> Run -> Backwards	(This will move the robot backwards)
Robot -> Stop	(stops all movement)

(-> = small pause)

After disconnecting from the Say It GUI, you can still verify that the Say It Module is detecting the right word by using the Debug Terminal. By leaving the Board of Education connected to the computer, each recognized verbal command will be printed to the Debug Terminal.

Note that not all commands will use a word from all 3 Wordsets to be a valid command. For example, "Hello" uses a Trigger word (Robot) and Hello from Wordset 1, which will end the command to then execute; that debugs "Hello" on the BASIC Stamp Debug Terminal.

Troubleshooting

From time to time there may be some snags that can cause what would seem like malfunctions in the module. If you experience any of the symptoms listed below, here are some quick fixes to try.

Q1. Keep getting a time-out error
A1. Make sure the power has not been cycled since the last time the GUI was connected.

Q2. Can't connect to my Say It Module
A2.1 Be sure to close all terminal windows including the debug screen before connecting the GUI software.
A2.2 Check power and make sure it has ample voltage and current to turn on modules

Q3. Will not power up
A3. Check to make sure that all the connections are correct; if using an AppMod header be sure the orientation is correct.

Q4. I am running Windows Vista, and the Say It GUI will not install properly.
A4. Right Click on installer exe, and select "run as administrator"

Device Information

Specifications

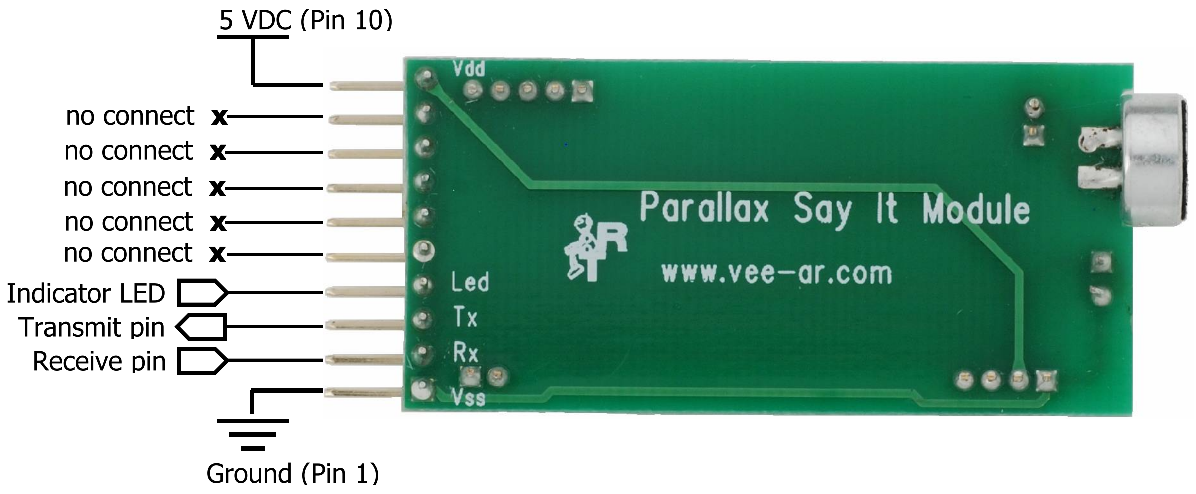
Symbol	Quantity	Minimum	Typical	Maximum	Units
Vdd	Supply Voltage	3.3	5.0	5.5	V

Pin Definitions

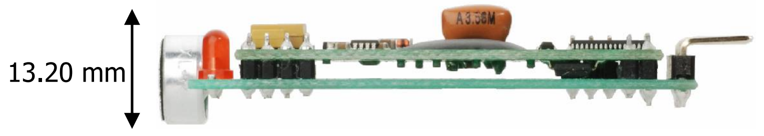
Pin	Label	Function
1	Vss	Ground
2	Rx	Receive I/O Pin (TTL & CMOS compatible)
3	Tx	Transmit I/O Pin (TTL & CMOS compatible)
4	Led	Red LED indicator
5	-	No Connection
6	-	No Connection
7	-	No Connection
8	-	No Connection
9	-	No Connection
10	Vdd	5 V regulated DC

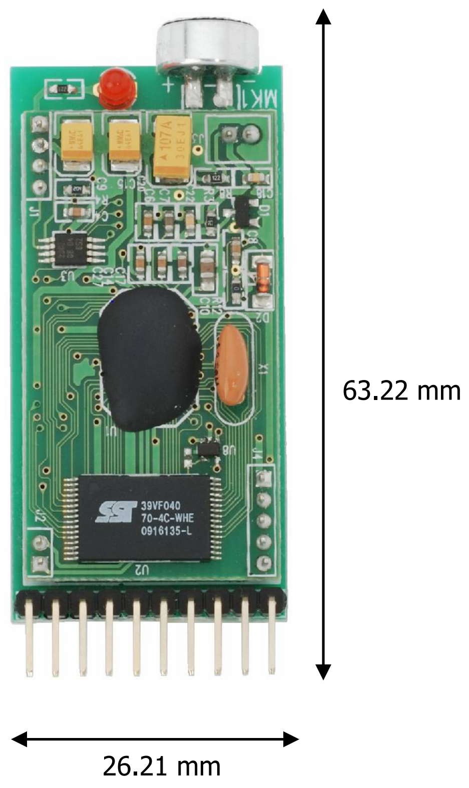
Connection Diagrams

This is the back view of the module, the connection pins are indicated on the silkscreen.



Module Dimensions





Communication Protocol

Communication with the Say It module uses a standard UART interface compatible with 3.3V to 5V TTL logical levels. The initial configuration at power-on is 9600 baud, 8 bit data, No parity, 1 bit stop. The baud rate can be changed later to operate in the range 9600 - 115200 baud.

The communication protocol only uses printable ASCII characters, which can be divided in two main groups:

- Command and status characters, respectively on the TX and RX lines, chosen among lower-case letters
- Command arguments or status details, again on the TX and RX lines, spanning the range of capital letters

Each command sent on the TX line, with zero or more additional argument bytes, receives an answer on the RX line in the form of a status byte followed by zero or more arguments.

There is a minimum delay before each byte sent out from the Say It module to the RX line, that is initially set to 20 ms and can be selected later in the ranges 0 - 9 ms, 10 - 90 ms, 100 ms - 1 s.

The communication is host-driven and each byte of the reply to a command has to be acknowledged by the host to receive additional status data, using the space character. The reply is aborted if any other character is received and so there is no need to read all the bytes of a reply.

Invalid combinations of commands or arguments are signaled by a specific status byte, that the host should be prepared to receive if the communication fails. Also a reasonable timeout should be used to recover from unexpected failures.

The module automatically goes to lowest power sleep mode after power on. To initiate communication, send any character to wake-up the module.

Command Details

CMD_BREAK	
“b”	Abort recognition in progress if any or do nothing
Expected replies: STS_SUCCESS, STS_INTERR	
CMD_SLEEP	
“s”	Go to the specified power-down mode
[1]	Sleep mode (0-8)
Expected replies: STS_SUCCESS	
CMD_KNOB	
“k”	Set Speaker Independent (pre-programmed commands) knob to specific level
[1]	Knob level (0-4)
Expected replies: STS_SUCCESS	
CMD_LEVEL	
“v”	Sets Speaker Dependent (custom programmed commands) to specific level
[1]	Threshold (1-5)
Expected replies: STS_SUCCESS	
CMD_LANGUAGE	
“l”	Set Speaker Independent (pre-programmed commands) language
[1]	Language (0 = English, 1 = Italian, 2 = Japanese, 3 = German
Expected replies: STS_SUCCESS	
CMD_TIMEOUT	
“o”	Set Speaker Independent (pre-programmed commands) language
[1]	Timeout (-1 = default, 0 = infinite, 1-30 = seconds
Expected replies: STS_SUCCESS	
CMD_RECOG_SI	
“i”	Activate Speaker Independent (pre-programmed commands) recognition from specified wordset
[1]	Wordset Index (0-3)
Expected replies: STS_SUCCESS, STS_TIMEOUT, STS_ERROR	
CMD_TRAIN_SD	
“t”	Train specified Speaker Dependent (custom programmed commands) or Password command
[1]	Group index (0 = trigger, 1-15 generic, 16 = password
[2]	Command position (0-31)
Expected replies: STS_SUCCESS, STS_RESULT, STS_SIMILAR, STS_TIMEOUT, STS_ERROR	
CMD_GROUP_SD	
“g”	Insert new Speaker Dependent (custom programmed commands) or Password command
[1]	Group index (0 = trigger, 1-15 generic, 16 = password
[2]	Command position (0-31)
Expected replies: STS_SUCCESS, STS_OUT_OF_MEM	

CMD_UNGROUP_SD	
“u”	Remove Speaker Dependent (custom programmed commands) or Password command
[1]	Group index (0 = trigger, 1-15 generic, 16 = password
[2]	Command position (0-31)
Expected replies: STS_SUCCESS	
CMD_RECOG_SD	
“d”	Activate Speaker Dependent (custom command) or Password recognition
[1]	Group index (0 = trigger, 1-15 = generic, 16 = password
Expected replies: STS_SUCCESS, STS_RESULT, STS_SIMILAR, STS_TIMEOUT, STS_ERROR	
CMD_ERASE_SD	
“e”	Remove Speaker Dependent (custom command) or Password recognition
[1]	Command position (0-31)
Expected replies: STS_SUCCESS	
CMD_NAME_SD	
“n”	Give a label for a Speaker Dependent (custom programmed commands) or Password command
[1]	Group index (0 = trigger, 1-15 = generic, 16 = password
[2]	Command position (0-31)
[3]	Length of label (0-31)
[4-n]	Text for label (ASCII characters from “A” to “`”
Expected replies: STS_SUCCESS	
CMD_COUNT_SD	
“c”	Request count of Speaker Dependent (custom programmed commands) or Password commands in a specified group
[1]	Group index (0 = trigger, 1-15 = generic, 16 = password)
Expected replies: STS_COUNT	
CMD_DUMP_SD	
“p”	Read Speaker Dependent (custom programmed commands) or Password command label (label and training)
[1]	Group index (0 = trigger, 1-15 = generic, 16 = password)
[2]	Command position (0-31)
Expected replies: STS_DATA	
CMD_MASK_SD	
“m”	Request a bit-mask of non-empty groups
Expected replies: STS_MASK	
CMD_RESETALL	
“r”	Reset all commands and groups
“R”	Confirmation character
Expected replies: STS_SUCCESS	
CMD_ID	
“x”	Request firmware ID

Expected replies: STS_ID

CMD_DELAY	
“y”	Set Transmit delay
[1]	Time (0-10 = 0 – 10ms, 11-19 = 20-100ms, 28-28 = 200 to 1000ms)
Expected replies: STS_SUCCESS	
CMD_BAUDRATE	
“a”	Set communication baud-rate
[1]	Speed mode (1 = 115200, 2 = 57600, 3 = 38400, 6 = 19200, 12 = 9600
Expected replies: STS_SUCCESS	

Status Details

STS_MASK	
“k”	Mask of non-empty groups
[1-8]	4-bit value that form a 32-bit mask, LSB first
In replay to: CMD_MASK_SD	
STS_COUNT	
“c”	Count of commands
[1]	Integer (0-31)
In replay to: CMD_COUNT_SD	
STS_AWAKEN	
“w”	Wake-up (back from power-down mode)
In replay to: Any character after power on or sleep mode	
STS_DATA	
“d”	Provide command data
[1]	Training information (0-7 = training count, +8 = SD/Password conflicts, +16 = SI conflict
[2]	Conflicting command position (0-31)
[3]	Length of label (0-31)
[4-n]	Text for label (ASCII characters from “A” to “`”
In replay to: CMD_DUMP_SD	
STS_ERROR	
“e”	Signal recognition error
[1-2]	Two 4-bit values that form 8-bit error code (80h = NOTA, otherwise see FluentChip error codes)
In replay to: CMD_RECOG_SI, CMD_RECOG_SD, CMD_TRAIN_SD	
STS_INVALID	
“v”	Invalid command or argument
In replay to: Any invalid command or argument	
STS_TIMEOUT	
“t”	Timeout expired
In replay to: CMD_RECOG_SI, CMD_RECOG_SD, CMD_TRAIN_SD	
STS_INTERR	

“i”	Interrupted recognition
In replay to: CMD_BREAK while in training or recognition	

STS_SUCCESS	
“o”	OK or no error status
In replay to: CMD_BREAK, CMD_DELAY, CMD_BAUDRATE, CMD_TIMEOUT, CMD_KNOB, CMD_LEVEL, CMD_LANGUAGE, CMD_SLEEP, CMD_GROUP_SD, CMD_UNGROUP_SD, CMD_ERASE, CMD_NAME_SD, CMD_RESETALL	
STS_RESULT	
“r”	Recognized Speaker Dependent (custom commands), Password or training similar to Speaker Dependent (custom commands) and Password commands
[1]	Command position (0-31)
In replay to: CMD_RECOG_SD, CMD_TRAIN_SD	
STS_SIMILAR	
“s”	Recognized Speaker Independent (pre-programmed commands) work or training a similar Speaker Independent (pre-programmed commands) command
[1]	Wordset indext (0-31)
In replay to: CMD_RECOG_SD, CMD_TRAIN_SD,CMD_RECOG_SI	
STS_OUT_OF_MEM	
“m”	Memory Full Error
In replay to: CMD_GROUP_SD	
STS_ID	
“x”	Provide firmware ID
[1]	Version ID (0)
In replay to: CMD_ID	

Argument Mapping

ARG_MIN	
“@”	Minimum argument value (-1)
ARG_MAX	
“ ‘ “	Maximum argument value (-1)
ARG_ZERO	
“A”	Zero argument value