

SPEECH SYNTHESIZER

Contents:

- 1) SPE030 speech synthesizer module
 - 2) Programming adapter kit (pcb, 2 connectors, battery clip)
- Also required (for programming) : 4.5V battery pack
AXE026 PICAXE download cable



Specification:

The SPE030 speech synthesizer module is a complete text-to-speech 'talking' module, containing an on-board amplifier, speaker and electronic circuit based upon the WTS701 single chip synthesizer (which also includes a text to speech processor). Given the near impossibility of producing a definitive set of rules for text to speech processing, the WTS701 performs impressively.



Voltage	- 5V dc
Current	- 35mA Standby, 60-100mA Speaking.
Stored Speech	- 30 pre-defined phrases with up to 1925 total characters.
Text to Speech	- Unlimited via I2C interface.
Programming	- RS232 interface to PC with free Windows software
Control Mode 1	- 5 pre-defined messages via push switches
Control Mode 2	- 30 pre-defined messages via any PICAXE microcontroller
Control Mode 3	- I2C interface for easy wiring to PICAXE-18X or 28X
Synthesizer	- Winbond WTS701EM/T
Speaker	- 40mm Speaker built in.
Amplifier	- 325mW LM386 on-board.
Small Size	- 40mm x 40mm excluding connectors

Operation:

The 30 predefined phrases are downloaded from a PC into the SPE030 module. These 30 phrases can then be selected for playback by either the digital input port or an I2C signal. Additional text can also be spoken via the I2C interface. On power up, the module will speak phrase #1, if one has been stored.

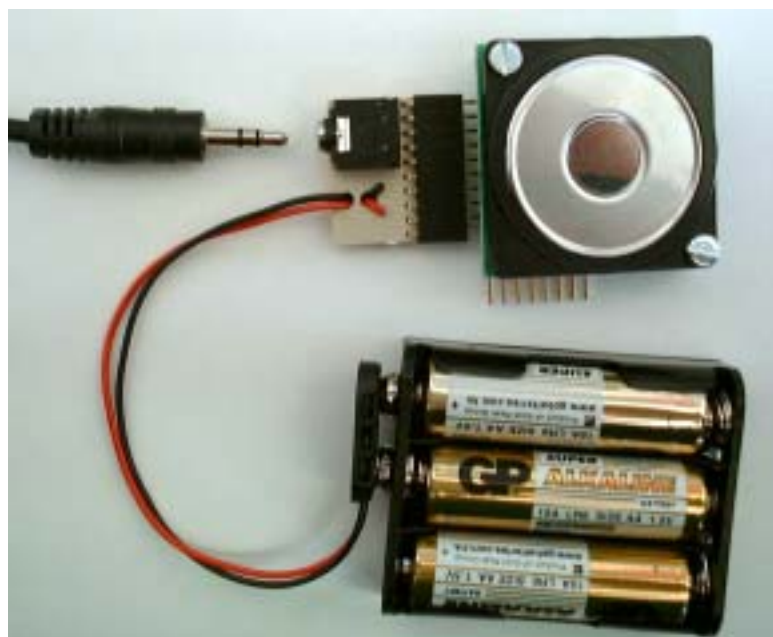
Initial Programming Adapter Assembly:

The self-assembly adapter allows a normal PICAXE cable to be used for programming the pre-defined messages.

Solder the 10 pin connector and stereo socket to the programming adapter PCB. Ensure the stereo socket clicks flat on to the PCB before soldering. Thread the battery clip through the large holes and solder in position.

Note the clip is to connect a 4.5V AA battery pack (not supplied).

DO NOT USE A 9V PP3 BATTERY AS THIS WILL DAMAGE THE SPE030 MODULE.



Electrical connection:

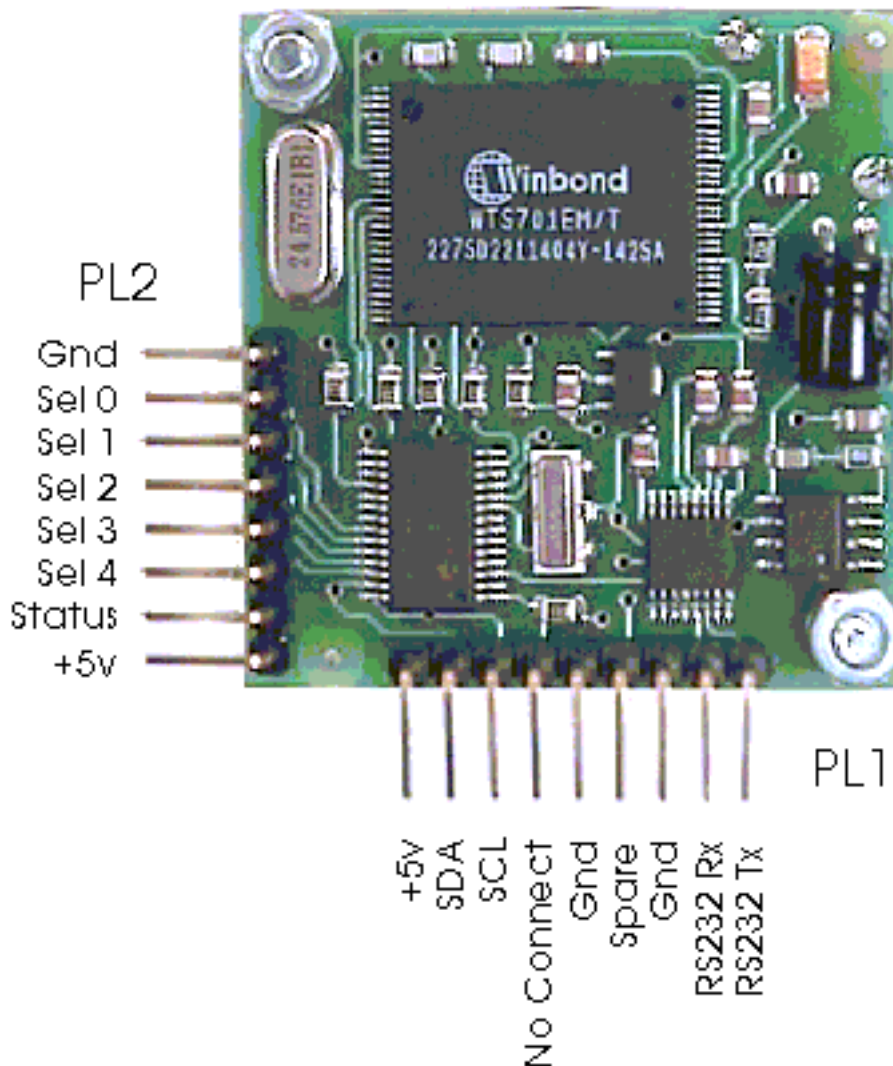
There are two connectors on the SPE030 module. The 5V power supply may be applied to either of them.

PL1 (9 pin)

- +5V - Power Supply - up to 100mA
- SDA - I2C bus SDA connection
- SCL - I2C bus SCL connection
- N/C - Do not connect this pin
- GND - The 0 volt Ground line
- Spare - Do not connect this pin
- GND - The 0 volt Ground line
- RS232 Rx - Connect to Tx on the PC
- RS232 Tx - Connect to Rx on the PC

PL2 (8 pin)

- GND - The 0 volt Ground line
- Sel 0 - Sel 0 to Sel 4 are the
- Sel 1 - digital binary select
- Sel 2 - input port. They select
- Sel 3 - one of the 30 predefined
- Sel 4 - phrases.
- Status - High when speaking, Low when done
- +5V - Power Supply - up to 100mA

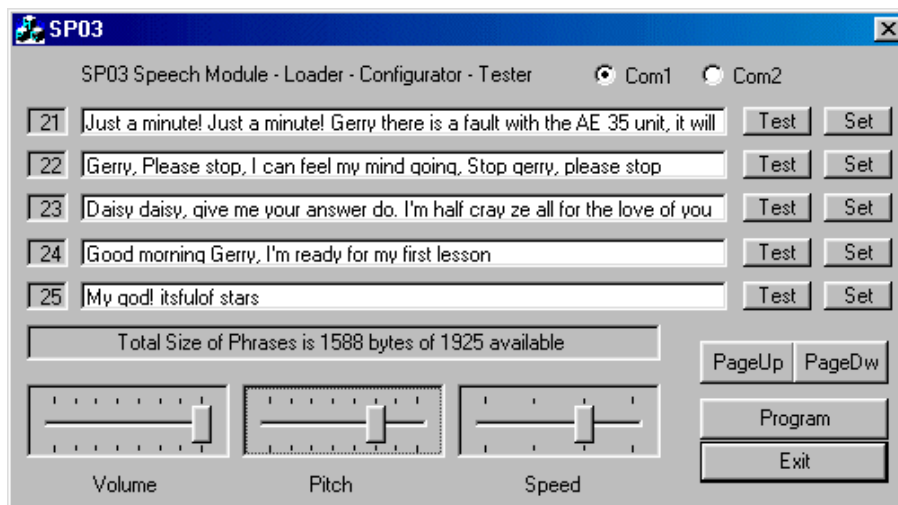


Programming the pre-defined phrases:

Before use, up to 30 pre-defined phrases can be downloaded into the SPE030 module. This is achieved by connecting the programming connector PCB (provided in self assembly kit form) to connector PL1. The connector PCB allows the standard PICAXE cable (AXE026 or AXE027) to be used to download the phrases. Note the power supply connected to the connector PCB should be 4.5V only, a 9V PP3 battery will damage the module.

When connecting the SPE030 module to the connector PCB take care to ensure the 9 pins of PL1 align correctly.

The configuration software is called SP03.exe and can be found in SP03 folder on the PICAXE CDROM, or downloaded from www.picaxe.co.uk



The operation of the program is fairly straight forward. Start by selecting the COM port (1 or 2) and setting the Volume, Pitch & Speed sliders as shown above. Now type something into one of the edit boxes and press the "Test" button to the right of that edit box. The words you typed will be instantly spoken. Notice that the message bar keeps track of the number of characters used so far. This is the total for all 30 phrases.

The "Test" and "Set" buttons

Both of these buttons cause the phrase to be spoken. The "Test" button uses, and also stores, the Volume, Pitch & Speed values set on the sliders. The "Set" button uses the Volume, Pitch & Speed values stored from the previous use of the "Test" button. So "test" saves and uses the screen value into the unit, "set" uses the value already in the unit (ie ignores the screen value). Therefore you can have different Volume, Pitch & Speed settings for each of the 30 phrases. When you've all your phrases set-up and tested using the "Set" buttons you're ready to program them into the SPE030 module.

Programming the Phrases

To program the SPE030 module just press the "Program" button. Your phrases will be compressed and stored in the Flash memory of the SPE030 module. The message bar will report the progress of programming. When programming is complete you will notice that the "Test" buttons are de-selected. The remaining "Set" buttons have now changed mode and when pressed will cause the phrase to be spoken directly from the pre-defined phrase in the SPE030 memory. Click in any of the edit boxes to restore the "Test" and "Set" buttons to normal. Note the "program" button DOES NOT download volume, pitch or speed. Therefore you need to use the 'test' button on each phrase before 'program'.

MODE1: Operation with switches (digital input port):

The simplest way to use the SPE030 module is to connect 5 push switches, one between each of SEL0 to SEL4 and 0V. Unused pins are pulled high by the on-board pull-up resistors. Pressing one of the switches will play back one of five messages as follows.

- Pressing SEL0 will play back message number 30 (%11110)
- Pressing SEL1 will play back message number 29 (%11101)
- Pressing SEL2 will play back message number 27 (%11011)
- Pressing SEL3 will play back message number 23 (%10111)
- Pressing SEL4 will play back message number 15 (%01111)

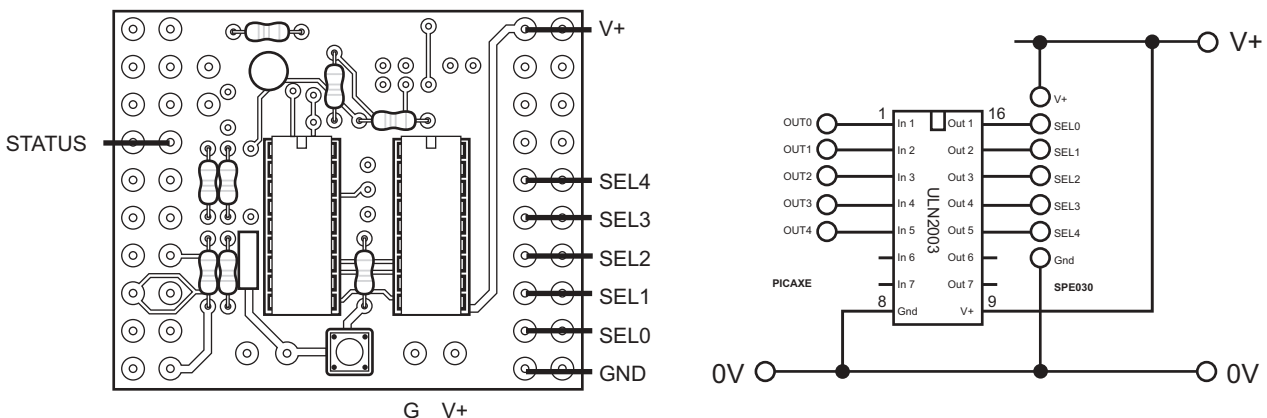
If desired, an LED (and 330R resistor) can be connected between the STATUS pin and 0V. The LED will light whilst the phrase is being spoken.

MODE2: Operation with any PICAXE microcontroller (digital input port):

The digital input port on PL2 is used to speak one of the 30 pre-defined phrases. Text cannot be sent to the digital input port for speaking, only the pre-defined phrases can be selected in this manner.

To speak any of the 30 (1-30) pre-defined phrases, apply the phrase number to the 5 bit input port SEL0 - SEL4. The numbers 0 and 31 (\$00 and \$1F) are ignored and do not cause any phrase to be spoken. The SP03 has pull-up resistors on the inputs so they can be left unconnected if not used. As soon as the CPU has recognised and confirmed the new signal it will raise the STATUS bit to a logic 1 (high) and speak the phrase. As soon as the STATUS bit goes high, the input code may be removed by returning the input to 0. This must happen before the unit has finished speaking or the phrase will be repeated. The STATUS bit will go to a logic 0 (low) when the SP03 has finished speaking, and this can be monitored by the PICAXE microcontroller on an input (e.g. input 2).

Note that although there are 5 select pins, they do not all have to be used. Therefore if you only require 3 phrases you would only need to use SEL0 and SEL1. The select pins are active low (ie the signal must be taken low to activate them). Note this has the action of inverting the selection (ie 00001 plays back 11110 = message 30). This inverted logic is so they can be wired directly to the 'transistor' outputs found on most PICAXE project boards. The STATUS pin can be connected to a PICAXE input (e.g. input 2) to indicate when speaking has finished.



Sample connection (PICAXE-18 projectboard CH1030):

Note that on the project board the wire is connected to the inner hole of the output/input pairs.

Sample PICAXE-18 Program:

```

main:
' *** playback message 30***
  let pins=%00000001          ' select message 11110 = 30
  gosub wait4shutup          ' wait for speech to finish

' *** playback message 21 ***
  let pins=%00001001          ' select message 10110 = 21
  gosub wait4shutup          ' wait for speech to finish

  goto main

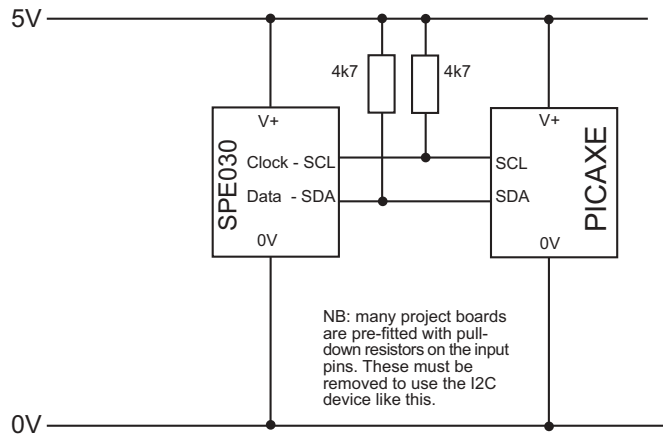
' *** sub to wait until SP03 has stopped talking ***
wait4shutup:
  pause 10                    ' short delay
  let pins = %00000000        ' reset select pins
  if pin2 = 1 then wait4shutup ' test status pin
  return
    
```

MODE 3: Operation with the PICAXE-18X/28X microcontroller (i2c bus):

Along with the 5 Volt power supply, the I2C bus connection just requires the SDA and SCL lines.

The I2C bus option also allows the SPE030 module to speak text directly, rather than just the pre-defined messages (which are also available).

The I2C interface does not have any 4k7 pull-up resistors on the SPE030 module, these should be provided in the PICAXE interface circuit. 4k7 resistors are required on both the SCL and SDA lines as shown.



The i2c communication protocol with the speech module is the same as popular eeprom's such as the 24C04. The SPE030 family code is \$C4, operates at 400kHz (i2cfast) and has a single byte (i2cbyte) address size.

Therefore the PICAXE i2c setup command (required before readi2c or writei2c is used) is

```
i2cslave $C4,i2cfast,i2cbyte
```

The SP03 only has two registers, the command register (address 00) and the software revision number (address 01).

To read the software revision number, read address 01

```

e.g.      readi2c 1,(b2)
          debug b2
    
```

All commands and text to be spoken are written to the command register (address 00). There are 32 valid commands, listed below:

Command	Action
LOAD BUFFER (\$00)	Load buffer with text to be spoken
SPKPRE 1 to 30 (\$01 to \$1E)	Speak pre-defined phrase
SPKBUF 64 (\$40)	Speak text previously stored in buffer

The LOAD BUFFER command is followed by the text that you want spoken. You may send as little or as much (up to the 85 byte limit) as you wish. A number of LOAD BUFFER sequences may be used to build up the buffer before the SPKBUF command is issued. The buffer is flushed empty every time a SPKPRE or SPKBUF command is used. A text sequence is the 3 control bytes (volume, pitch, speed), then the ASCII text, and finally a null zero (\$00). The text in the buffer may then be spoken by sending a SPKBUF command. The SPE030 module's text buffer is 85 bytes in size, so that is the limit for a single phrase. The volume, pitch, speed and trailing NULL characters take 4 bytes leaving 81 for the text. Therefore to say "PICAXE" send the following sequence over the I2C bus

```
writei2c 0,(0,0,5,3,"pickacks",0)
writei2c 0,(64)
```

where the data is broken down as follows

0	SP03 LOAD BUFFER Command
0	Volume (Max.) (0 to 3, 0 is max)
5	Speech Pitch (0 to 7)
3	Speech Speed (0 to 7)
"pickacks"	Text to speak
0	NULL

Note that on uncommon words (such as PICAXE) better results can often be gained by experimenting with a 'phoentic' spelling (e.g. pickacks).

To check to see when the SPE030 has finished speaking, you can read back the command register (address 0) -

```
readi2c 0,b1
```

Whilst speaking, the register will contain the command that initiated the speaking, either 1-30 (\$01-\$1E) or 64 (\$40). It will be cleared to zero (\$00) when speaking is complete and the module is ready for the next phrase.

Sample PICAXE Program:

```
' Define i2c slave address for the SPE030 Speech Module
  i2cslave $C4,i2cfast,i2cbyte

' *** firmware ***
  ' read firmware version into b1 and display on screen
  readi2c 1,(b2)
  debug b2

main:
' *** message playback ***
  'first play back a pre-recorded message (msg 1)
  'remember to use gosub to wait for message to stop
  writei2c 0,(1)
  gosub wait4shutup

' *** instant speech ***
  ' now download text to instantly speak
  ' first write text to speak to buffer (max 81 characters)
  ' load buffer(0), volume(0), pitch(5), speed(1), "text", null(0)
  ' note creative spelling makes picaxe sound better!

  writei2c 0,(0,0,5,1,"Hello pickacks user",0)

  ' now actually say it
  writei2c 0,(64)
  gosub wait4shutup

  'loop forever
  goto main

' *** sub to wait until SP03 has stopped talking ***
wait4shutup:
  readi2c 0,(b1)
  if b1 > 0 then wait4shutup
  return
```

SP03 Mounting Dimensions

