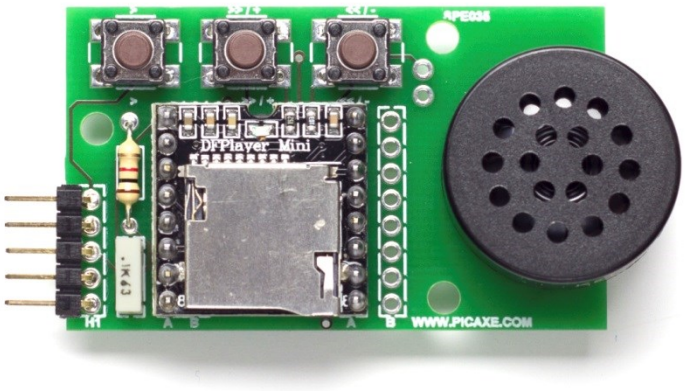
SPE035 Serial MP3 Player Module



The SPE035 Serial MP3 player consists of a small MP3 player module mounted on a PCB with serial connector, test switches and 8 ohm speaker. It provides a simple and low cost way to add MP3 tune playback to any PICAXE project.

The MP3 audio files (music, speech etc.) are copied onto a microSD card (not included) which is then inserted into the MP3 player. A simple 3 wire connection to the PICAXE project then allows playback and control of the audio tunes.

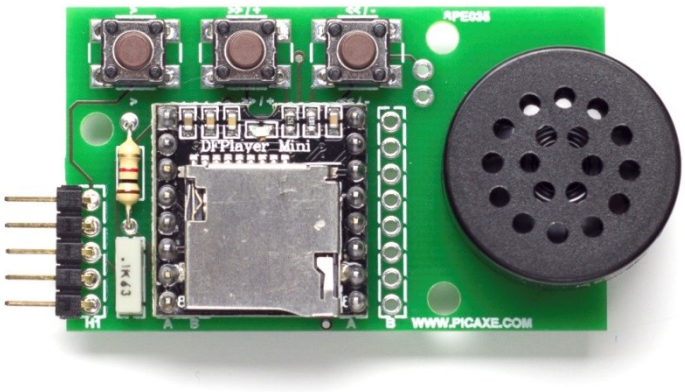
Features:

1. Supports MP3 files with all most common file sampling frequencies (kHz):
   8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48
2. High quality playback on 8 ohm speaker (included) with 90dB dynamic range
3. PCB pads for alternate stereo line out audio connection
4. Supported file format: MP3 / WAV
5. Supports 1GB to 32GB microSD card (FAT16 or FAT32)
6. 30 volume settings (1= quiet, 30 = full volume)
7. 6 equalizer settings (0=Normal, 1=Pop, 2=Rock, 3=Jazz, 4=Classic, 5=Bass)
8. TTL serial control playback mode, at 9600 baud rate (PICAXE serout at T9600_8)
9. Power supply can be 3.3 to 5.2V DC
10. Control logic interface can be at 3.3V or 4.5V or 5V TTL
11. On board switches for playback testing
12. Small PCB size - 52 x 32mm

For the full datasheet see **www.picaxe.com/docs/spe035.pdf**

1.0 Assembling the SPE035 Serial MP3 Module

1.1 Overview



Note that there are two positions (A or B) on the PCB for the MP3 player module. This is because MP3 playback modules using the YX5300 MP3 chip are available in various different types and sizes. Carefully check the position of your module before soldering on to the board.

For all MP3 module types the microSD card connector is placed at the bottom of the SPE035 PCB. Note the microSD card socket is a 'push to eject' style socket (so you must press the card inwards slightly to eject it). If you simply pull hard on the card without ejecting it you may cause physical damage to the socket and/or MP3 module.

The SPE035 PCB was primarily designed to be used with a 'DF Player Mini' (or fully compatible clone such as BY-TF-16P). This is the recommended module and is used in position 'A'.

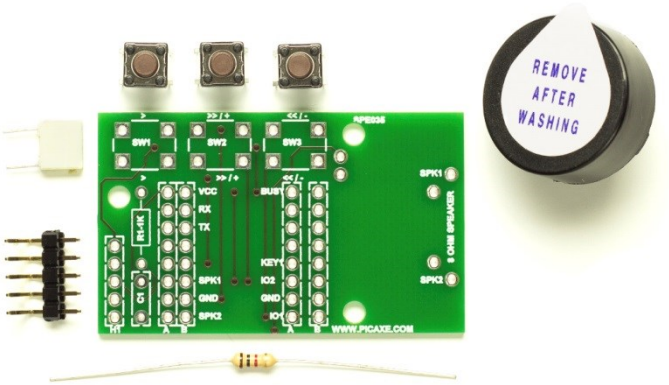
www.picaxe.com/docs/spe033.pdf

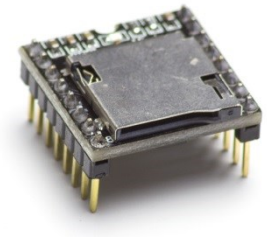
The PCB is also compatible with the 'BY8001-16P' module. This is used in position 'B' as it has a different size/pinout. Note that for serial control of the BY8001 it **must** be put into 'serial mode 111' by **removing all 3** resistors shown in the red circle (the default is normally a resistor in positions 'A' and 'C' with 'B' already removed).

www.picaxe.com/docs/by8001.pdf

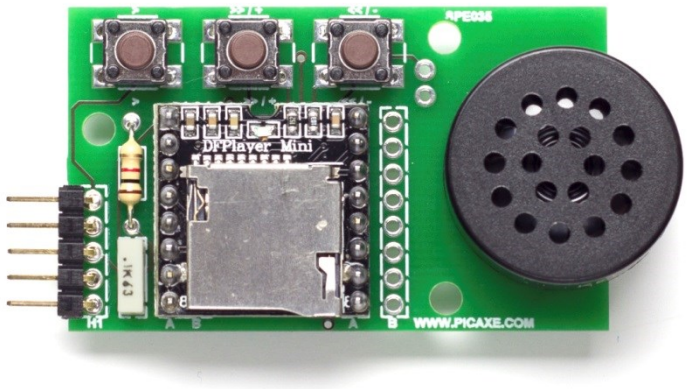
The PCB is not compatible with SPI based MP3 modules such as the WTV020M01 or WDV020-SD-16

1.2 SPE035 Kit Contents

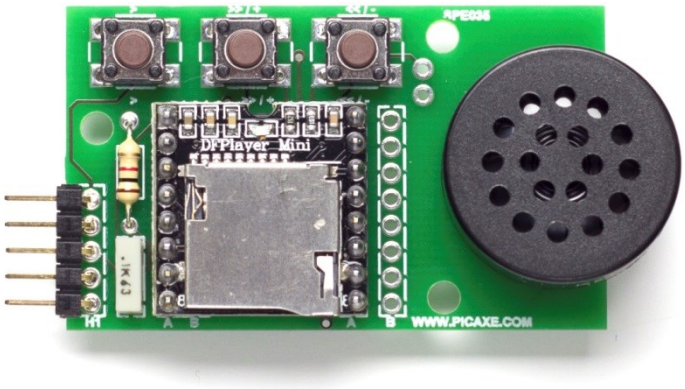




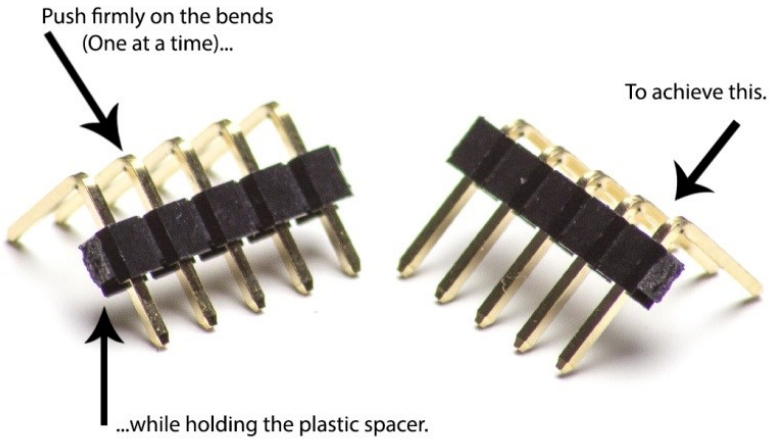
1	PCB		SPE035 pcb
3	SW1-3	SEN030	6mm push switch
1	R1	RES-1K	1k resistor (brown black red gold)
1	C1	CAP001	100nF capacitor
1	H1	CON042	5 pin r/a header
1	SPK1	SPE015	8 ohm PCB mount speaker
2	MP3 socket	CON050	8 way 2.54mm sockets (optional)
1	MP3 player	SPE033	DFPlayer Mini (or similar clone)



1.3 Assembly



- Solder the 1k resistor in position R1 and capacitor in position C1.
- Solder the three switches in positions SW1 to SW3
- Solder the MP3 module in the appropriate position A or B (see note 1.1 above). The module may be soldered directly onto the board or the optional 8 way connectors supplied may be used (if you wish to be able to remove the module from the PCB in the future).



- On the 5 way connector carefully push hard on the corner of the first pin (e.g. with a coin) so that the short end slides through the plastic holder, therefore making the short end longer. Do this for all 5 pins. Place the longer end through the PCB in position H1 and solder in position. The connector is adjusted and used like this so it lies closer to the top of the PCB.
- Solder the speaker wires into positions SPK1 and SPK2. There are 3 different solder pad positions to allow for different speaker sizes and shapes. If you wish to use a different size of speaker it should be an '8 ohm' type. The speaker can be connected either way round.

1.4 Loading MP3 files onto your micro SD card (not included)

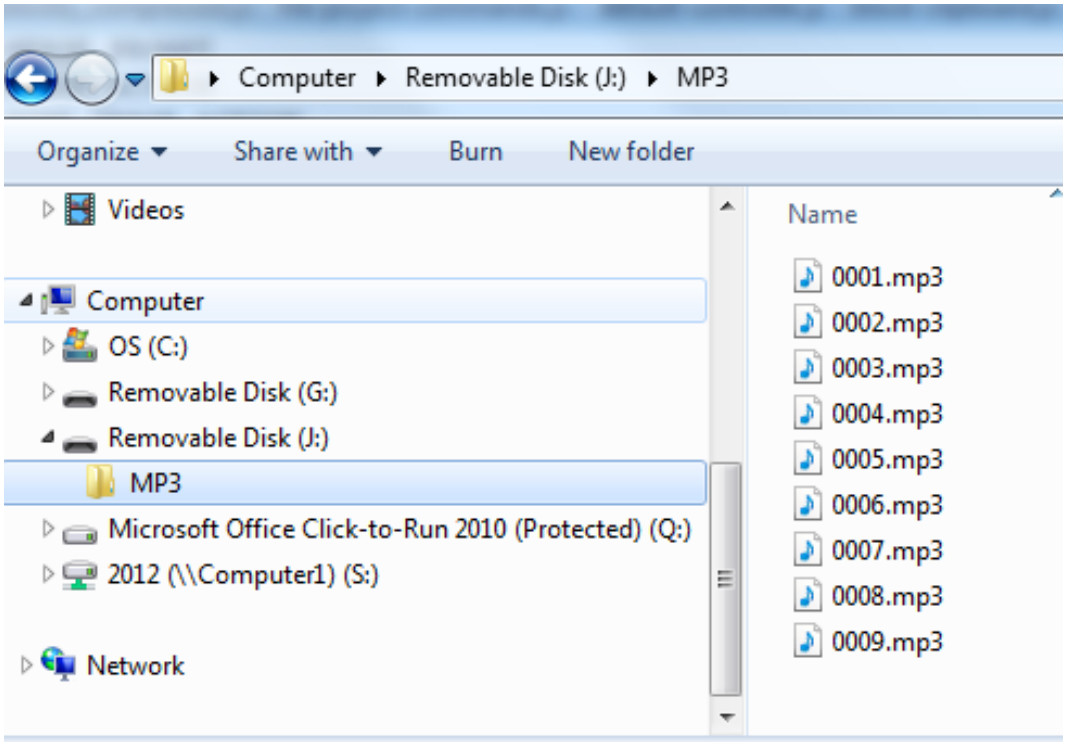
IMPORTANT – Note the microSD card socket on the top of the MP3 module is a 'push to eject' style socket (so you must press the card inwards slightly to eject it). If you simply pull hard on the card without ejecting it you may cause physical damage to the socket and/or module!

MP3 music/speech files may be copied onto the card using a mobile phone, tablet or computer (a separate SD card adapter may be required to insert the microSD card into a computer). MicroSD cards from 1GB to 32GB are supported. Many people may already have a surplus micro SD card from an old mobile phone which will be ideal.

The music filenames **must** have filenames starting with 0001.mp3, 0002.mp3, 0003.mp3 etc. and be saved in a subfolder called \MP3 on the microSD card.

A set of sample MP3 files for testing may be downloaded from

www.picaxe.com/downloads/mp3.zip



So the MP3/WAV files copied onto the microSD card should:

- Use a filename that starts with 4 numeric characters e.g. 0001.mp3, 0002.wav
- Use a filename that starts with numbers between 0001.mp3 and 0255.mp3
- End with .mp3 or .wav
- Be saved within a folder called \MP3 on the microSD card
- Be unprotected files - DRM 'copyright protected' music files are not supported
- See Appendix 2 for more details about MP3 filenames.

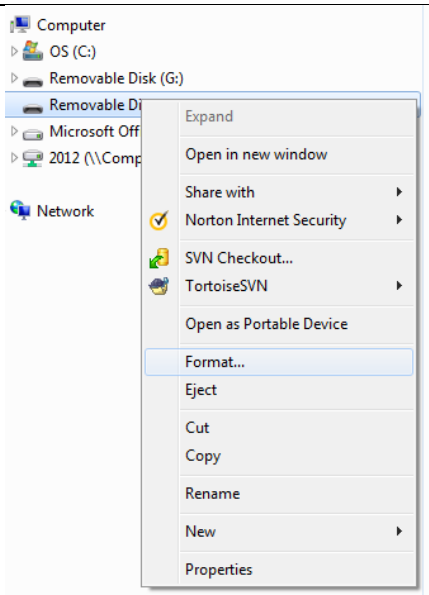
Once the music and speech files have been copied onto the micro SD card carefully and gently insert the card into the socket on the MP3 module.

1.5 Step by step guide to copying MP3 files on to the SD card:

If using an old recycled microSD card it is best to 'Quick Format' it before use.

Step 1.

Insert the microSD card into the computer (an adapter may be required), right click over the microSD card drive and select 'Format…'

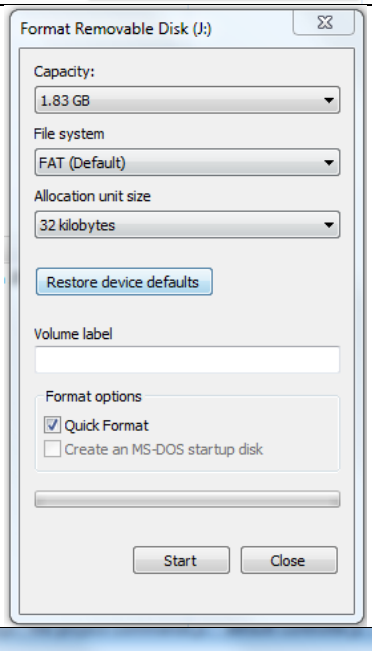


Step 2.

Make sure 'Quick Format' Is selected.

Double check that it is **definitely** the microSD card drive that is selected (as formatting **permanently** deletes all data)

Click Start

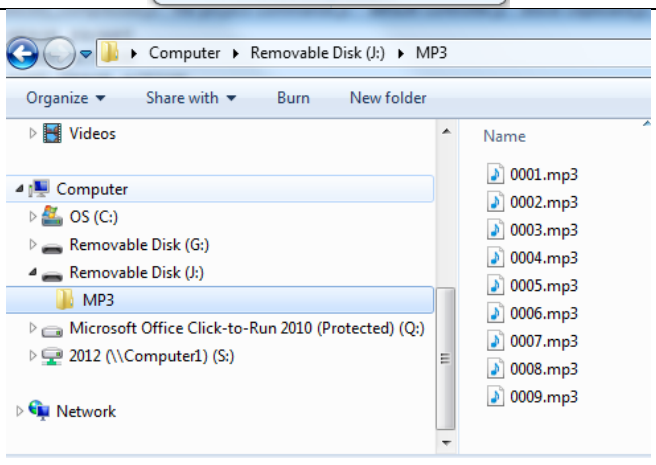


Step 3.

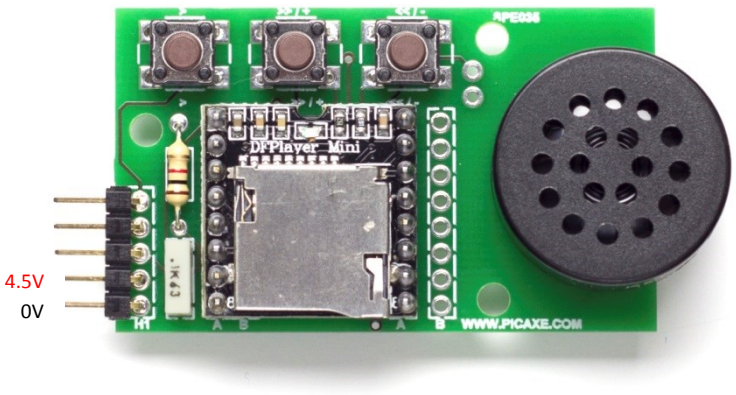
Once formatting is complete create a new folder called MP3 on the microSD card

Step 4.

Copy the MP3 files into the \MP3 sub folder. It is best to copy the files one at a time in order (see notes in 2.0)



2.0 Testing the SPE035 project board (using the on-board switches).



1. Make sure the microSD card containing MP3 files has been inserted into the MP3 player.
2. Connect power (4.5V or 5V) to the bottom two pins of the header H1 (pin 4 = V+, pin 5 = 0V).
3. Press the left play (>) switch, the first tune found will then play.
4. A short press of the centre (>> / +) switch will move to the next track.
5. A short press of the right (<< / -) switch will move to the previous track.
6. A long press of the centre (>> / +) switch will increase the volume.
7. A long press of the right (<< / -) switch will decrease the volume.

Note that, as with most MP3 players, the next/previous track push buttons operate on the FAT file system sort, not the alphabetical filename sort. So the next/previous track played will be according to the order that the files were originally saved onto the card, ***not*** *via an alphabetical sort of the filenames currently on the card. Use a search engine search as Google to find a 'FAT filename sorter' utility program if the push switch playback order is essential and must be amended.*

When playback is controlled by serial commands from a PICAXE chip the files are called directly by filename number, so the sort order on the microSD card is then not important.

3.0 Connecting to a PICAXE project board.

For simplest control only 3 wires are required, these are the lowest 3 connections on the 5 way connector H1. A servo extension lead (part DAG001) may be a useful cable for making this connection. Note that the PICAXE output connection must always be made directly to the PICAXE pin on the project board (not, for instance, via a Darlington driver buffered output).

PICAXE-14M2	Pin	H1 on SPE035
-	1	Output Busy
-	2	Output TX
Output TX	3	Input RX
V+	4	V+
0V	5	0V

For more advanced control the BUSY output signal from the MP3 module may also be connected

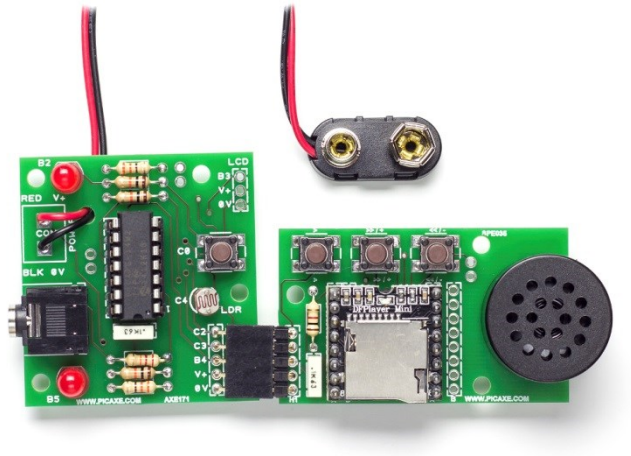
PICAXE-14M2	Pin	H1 on SPE035
Input Busy	1	Output Busy
-	2	Output TX
Output TX	3	Input RX
V+	4	V+
0V	5	0V

For full feedback control the TX output from the MP3 module may also be connected.

When used with the AXE171 PICAXE 14 Audio Kit the connections are as follows:

PICAXE-14M2	Pin	H1 on SPE035
Input C.2	1	Output Busy
Input C.3	2	Output TX
Output B.4	3	Input RX
V+	4	V+
0V	5	0V

4.0 Example PICAXE programs



The following examples assume use of the AXE171 PICAXE 14 Audio Kit, so if using a different board you may need to alter the input / output pins in these examples (which are 'serial transmit' on output B.4 and 'busy signal' receive on input pinC.2).

When used with the AXE171 PICAXE 14 Audio Kit the connections are as follows:

PICAXE-14M2	Pin	H1 on SPE035
Input C.2	1	Output Busy
Input C.3	2	Output TX
Output B.4	3	Input RX
V+	4	V+
0V	5	0V

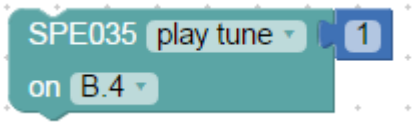
Note that the MP3 module requires up to 4 seconds (pause 4000) at power on to read the tune filename data from the microSD card. Depending on the quantity of MP3 files this can take up to 4 seconds, but may be much quicker. Therefore do not send any commands within this initial period.

Examples are provided for all Blockly, Flowchart and BASIC programming options.

To view the included samples files within 'PICAXE Editor 6' use the File>Open Samples menu and select 'AXE171 – PICAXE 14 Audio Kit'

4.1 Using Blockly

Blockly v1.0.5 (and later) has a special SPE035 command within the Outputs menu. This simplifies use of all the most common SPE035 commands. Simply drag out the block and select the desired command from the drop down list.

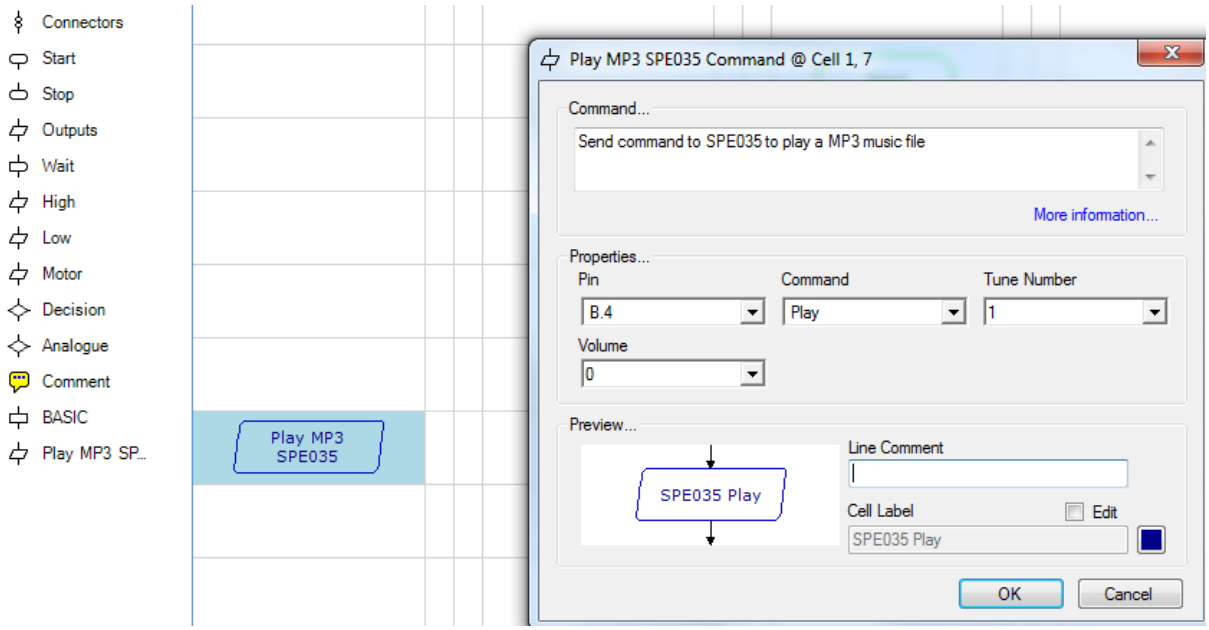


A number (or variable) block only needs to be attached to the SPE035 block for the 'play tune' and 'set volume' commands.

4.2 Using Flowcharts

PE6 v6.0.8.4 (and later) has a special SPE035 flowchart command. To add this command right click over any whitespace area in the Flowchart Toolbox and select 'Add New Item' > 'Play MP3 SPE035'

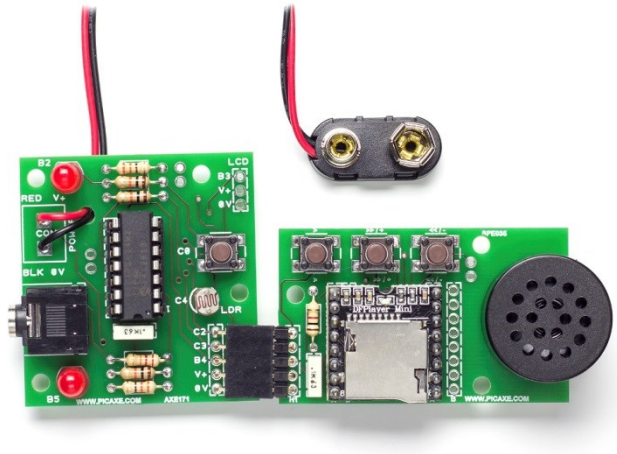
This special flowchart block simplifies use of all the most common SPE035 commands. Simply drag out the block, double click on it to edit and select the desired command from the drop down list.



The 'tune number' value is only used when the 'Play' command is selected.

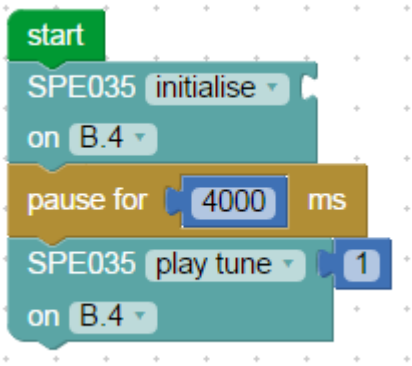
The 'volume' value is only used when the 'Set Volume' command is selected.

4.3 Simple 'Play Tune' Example

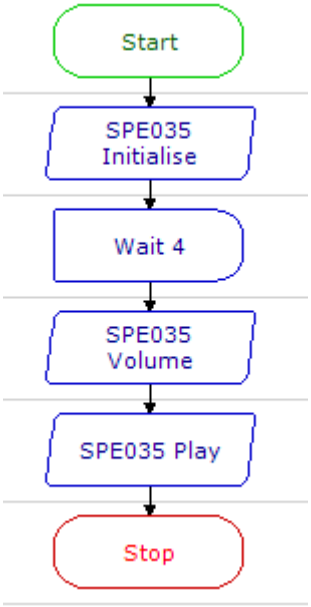


To use the SPE035 it is first necessary to issue the 'initialise' command to the module (this tells the module to read files from the microSD card). Depending on the quantity of MP3 files this can take up to 4 seconds, but may be much quicker. The play command is then used to play the desired tune (0001.mp3).

Blockly



Flowchart



BASIC

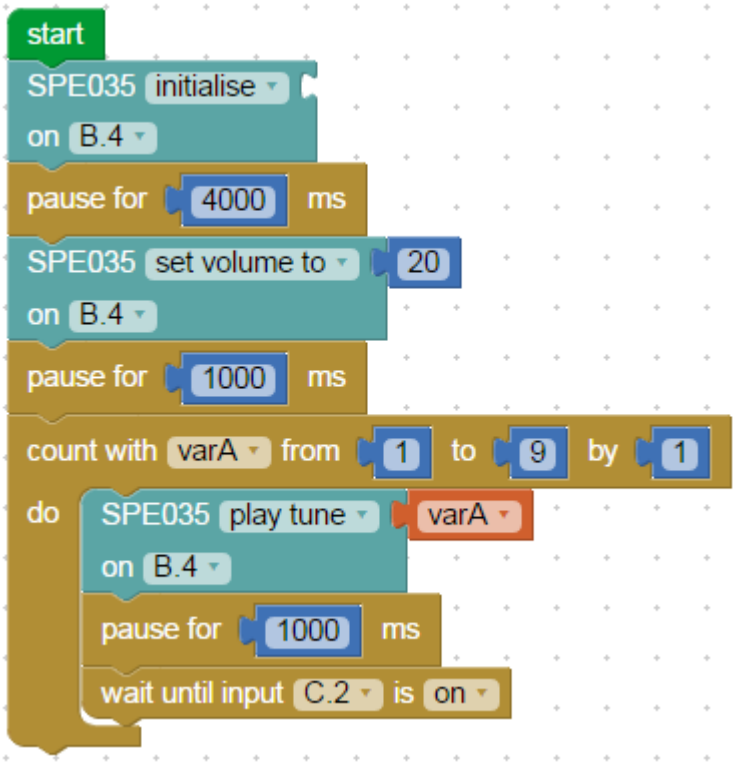
```
Symbol TX         = B.4

Symbol RX         = C.3
Symbol BUSY_PIN   = pinC.2

Symbol BAUD_FREQ = M8
Symbol BAUD       = T9600_8

Symbol cmd        = b0

Symbol arg        = w1 ; b3:b2
Symbol arg.lsb    = b2
Symbol arg.msb    = b3

High TX      ; set TX pin high for idle high serial

Pause 2000
SerTxd("Starting", CR, LF )

SerTxd("Select microSD Card", CR, LF )
cmd = $09 : arg = $0002 : Gosub Send
Pause 4000

SerTxd("Play MP3 folder song 0001.mp3", CR, LF )
cmd = $12 : arg = 0001 : Gosub Send
Pause 1000

Stop

Send:
  SetFreq BAUD_FREQ
  Pause 10
  SerOut TX, BAUD, ( $7E, $FF, $06, cmd, $00, arg.msb, arg.lsb, $EF )
  SetFreq MDEFAULT
  Return
```

4.4 Play Multiple Tunes using a Variable

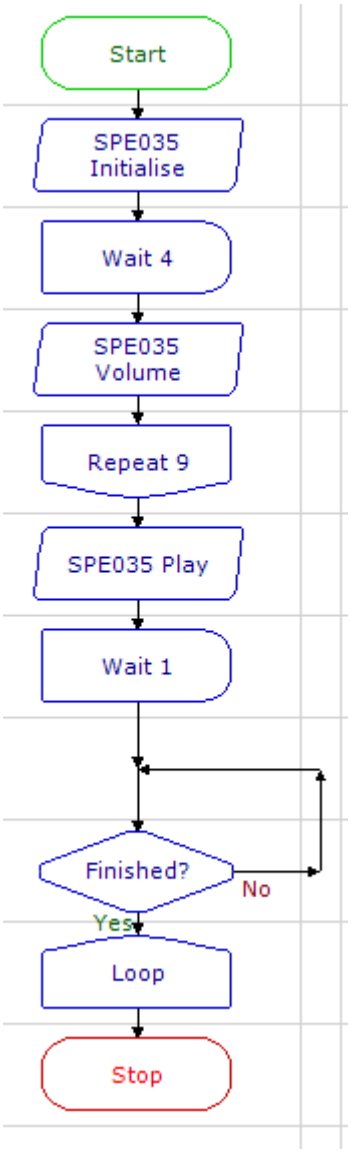
This example shows how to set the volume (valid values are 1 to 30) and then use a variable (varA) to play the tunes 0001.mp3 to 0009.mp3 in turn.

The end of the previous tune is detected by checking whether the MP3 BUSY signal (connected to PICAXE input C.2) is high. The BUSY signal is low (off) when the tune is playing and high (on) at all other times. Therefore the program waits for input pinC.2 to go high (on) before playing the next tune.

Blockly



Flowchart



BASIC

```
Symbol TX         = B.4

Symbol RX         = C.3
Symbol BUSY_PIN   = pinC.2

Symbol BAUD_FREQ = M8
Symbol BAUD       = T9600_8

Symbol cmd        = b0

Symbol arg        = w1 ; b3:b2
Symbol arg.lsb    = b2
Symbol arg.msb    = b3

Symbol varA       = w2

High TX      ; set TX pin high for idle high serial

Pause 2000
SerTxd("Starting", CR, LF )

SerTxd("Select microSD Card", CR, LF )
cmd = $09 : arg = $0002 : Gosub Send
Pause 4000

SerTxd("Set volume 20", CR, LF )
cmd = $06 : arg = 20 : Gosub Send
Pause 1000

For varA = 1 To 9
  SerTxd("Play MP3 folder song 000", #varA, CR, LF )
  cmd = $12 : arg = varA : Gosub Send
  Pause 1000
  Do While BUSY_PIN = 0
    Pause 100
  Loop
Next

Sertxd("Done", CR, LF )
Stop

Send:
  SetFreq BAUD_FREQ
  Pause 10
  SerOut TX, BAUD, ( $7E, $FF, $06, cmd, $00, arg.msb, arg.lsb, $EF )
  SetFreq MDEFAULT
  Return
```

5.0 MP3 Module (DFPlayer Mini) Pinout

Pin	Pin Name	Function	Notes
1	VCC	Power Supply	3.3V - 5V , recommended 5 V, max 5.2V
2	RX	UART serial data input	Protected with 1k series resistor
3	TX	UART serial data output	
4	DAC_R	Audio line output R	Drive amplifier, See appendix 3
5	DAC_L	Audio line output L	Drive amplifier, See appendix 3
6	SPK1	Speaker -	Speaker - (8 ohm)
7	GND	Ground	Power 0V Ground
8	SPK2	Speaker +	Speaker + (8 ohm)
9	IO1	Switch 3	Previous track (long press = volume -)
10	GND	Ground	Power Ground
11	IO2	Switch 2	Next track (long press = volume +)
12	ADKEY1	Switch 1	Play track 0001
13	ADKEY2		Not used
14	USB +		See appendix 4
15	USB -		See appendix 4
16	Busy	Playback indicator	Playing = 0, not playing = 1

Stereo line out (GND, DAC_L and DAC_R) is accessible via the 3 solder pads at the bottom of the PCB. See Appendix 3 for more information.

5.1 Header H1 Pinout

Pin	Pin Name	Dir	Function	Notes
1	Busy	Out	Playback indicator	Playing = 0, not playing = 1
2	TX	Out	UART serial data output	
3	RX	In	UART serial data input	Protected with 1k series resistor
4	VCC		Power Supply	3.3V - 5V , recommended 5 V, max 5.2V
5	GND		Ground	Power 0V Ground

6.0 Circuit Diagram

TODO - Diagram to be added.

6.1 Circuit Explanation

The MP3 module's BUSY, TX and RX pins are connected to header H1. RX is protected via a 1k series resistor so that a 5V microcontroller signal may be applied to the module (which runs at 3.3V).

IO1, IO2 and ADKEY1 are each connected via a miniature push switch to 0V.

The 8 ohm speaker is connected to SPK1 and SPK2.

The 'LIne Out' signal (GND, DAC_L, DAC_R) is also available via 3 unpopulated pads at the bottom of the PCB.

Appendix 1 - MP3 Player Command Summary

General Information:

Note that filenames and folders are saved on microSD card in decimal ASCII digit format e.g.

```
\MP3\0001.mp3
\MP3\0123.mp3
```

When using the default \MP3 and \ADVERT folders filenames must start with 4 digits e.g.

```
\MP3\0001.mp3
\ADVERT\0123.mp3
```

Command **0x12** is used to play tracks from the default \MP3 folder
Command **0x13** is used to play tracks from the \ADVERT folder

When using numbered sub-folders instead of \MP3 the folder name must have 2 digits (01 - 99) and the filename must start with 3 digits (001 - 255) e.g.

```
\01\001.wav
\02\222.mp3
```

Command **0x0F** is used to play tracks from a numbered folder

Advert Tracks

An 'advert' can be started whilst a different MP3 track is already playing. Therefore, for instance, a normal track from the \MP3 folder can be started with command 0x12. Whilst this track is still playing an advert can then be started with command 0x13.

The original track will then be temporarily suspended whilst the advert track plays. Once the advert is complete the original track will automatically resume playing.

Adverts must be saved in the \ADVERT sub folder and are controlled with these commands:

play advert **0x13**
stop advert **0x15**

FAT File System

Tracks can also be played via the file order reference within the FAT file system on the memory card (ie the order that the new files were saved onto the card after it was formatted). This FAT file order system completely ignores the actual filename. Valid FAT file number values are 01-255.

Command **0x03** is used to play a particular FAT file number
Command **0x01** and **0x02** are used to play the next/previous FAT file numbers

Note that the physical switches (play track 1, next track, previous track) on the SPE035 PCB also use the FAT file number, ignoring the actual filenames.

Serial Command Format

Baud rate is 9600,N,8,1 (idle high, which is PICAXE serout command setting T9600_8)

The general command format is

```
    .----------- (vv) Version = FF
    |
    |  .-------- (ln) Number of bytes vv+ln+cm+fb+dh+dl = 06
    |  |
    |  |     .-- (fb) Use 01 if one wants an echo response
    |  |     |        back from the module TX serial pin
    |  |     |
7E FF 06 cm 00 dh dl ch cl EF

      cm    = command
      dh dl = data for the command
      ch cl = checksum = 0 - vv - ln - cm - fb - dh - dl
```

Some technical datasheets display a checksum 'ch' and 'cl' as part of the serial data. However the module seems to completely ignore the checksum and does not care whether the checksum is given or not. Therefore for simplicity it is omitted in all PICAXE examples, which therefore gives the command sequence format

```
      7E FF 06 cm 00 dh dl EF
```

Note that filename numbers in the serial command list below is given in hex (not decimal)
e.g. to play file

```
      \02\222.mp3
```

which is folder 02 (0x02) track 222 (0xDE) - the command sequence of bytes would be

```
      serout pin, T9600_8, ($7E,$FF,$06,$0F,$02,$DE,$EF)
```

Command Summary

```
7E FF 06 cm 00 dh dl EF

          01    00 00       Play next FAT file number
          02    00 00       Play previous FAT file number
          03    00 nn       Play FAT file number nn=01-FF
          04    00 00       Increase volume
          05    00 00       Decrease volume
          06    00 vv       Volume, vv=00-1E (1E=full volume)

          07    00 00       Specify EQ Normal
          07    00 01       Specify EQ Pop
          07    00 02       Specify EQ Rock
          07    00 03       Specify EQ Jazz
          07    00 04       Specify EQ Classic
          07    00 05       Specify EQ Base

          08    00 nn       Repeat play FAT file number nn=01-FF

          09    00 01       Source is USB memory stick
          09    00 02       Source is TF (microSD) Card
          09    00 03       Source is Aux (not used)
          09    00 04       Source is PC (debug mode)
          09    00 05       Source is Flash memory
          09    00 06       Source is Sleep mode

          0A    00 00       Standby mode (enter sleep mode)
          0B    00 00       Normal mode (exit sleep mode)
          0C    00 00       Reset module
          0D    00 00       Resume play (un-pause)
          0E    00 00       Pause
          0F    ff tt       Play from folder ff=01-63, track tt=01-FF

          10    00 vv       Volume gain, vv=00-1F
          10    01 vv       Open volume adjust, vv=00-1F

          11    00 00       Normal mode
          11    00 01       Repeat play (next track to be played loops)

          12    tt tt       Play from MP3 folder, track tt=0001-270F
          13    tt tt       Play from ADVERT folder, track tt=0001-270F

          14    ft tt       Play from folder f=1-F, track ttt=001-3E7

          15    00 00       Stop advert, resume original track
          16    00 00       Stop playback
```

```
(continued)

7E FF 06 cm 00 dh dl EF

         17    00 ff       Repeat play tracks in folder ff. 0x16 to stop

         18    00 00       Shuffle play all tracks. Ox16 to stop

         19    00 00       Start looping of the currently playing track
         19    00 01       End looping of currently playing track

         1A    00 00       Cancel mute (enable DAC)
         1A    00 01       Mute (disable DAC, make DAC high-Z)

where
         vv = volume
         ff = folder
         tt = track
         nn = FAT file number
```

See www.picaxe.com/docs/spe033.pdf for the full technical datasheet of the module.

Response Summary

The module automatically sends out the following serial data upon certain events (described below). This data can be detected on TX (header H1 pin 2). However in practice most people use the simpler high/low BUSY (header H1 pin 1) signal to detect the end of the current track.

```
7E FF 06 cm 00 dh dl EF

          3A    00 01       USB memory stick inserted
          3A    00 02       TF (micro SD) card inserted
          3A    00 04       PC connected

          3B    00 01       USB memory stick removed
          3B    00 02       TF (micro SD) card ejected
          3B    00 04       PC disconnected

          3C    00 tt       USB     - End of track, track number tt=01-FF
          3D    00 tt       TF Card - End of track, track number tt=01-FF
          3E    00 tt       Flash   - End of track, track number tt=01-FF

          3F    00 00       Initialisation - No memory found
          3F    00 01       Initialisation - USB memory stick found
          3F    00 02       Initialisation - TF (micro SD) card found
          3F    00 04       Initialisation - PC is connected
          3F    00 05       Initialisation - Flash memory connected
```

See www.picaxe.com/docs/spe033.pdf for the full technical datasheet of the module.

Appendix 2 – Using Longer MP3 Filenames

The MP3 player module recognises the file number via the first 3 or 4 digits in the filename.

When using the default \MP3 and \ADVERT folders filenames must start with 4 digits e.g.

```
\MP3\0001.mp3
\ADVERT\0123.mp3
```

Valid file numbers are 0001 to 9999

When using numbered sub-folders the folder name must have 2 digits and the filename must start with 3 digits e.g.

```
\01\001.wav
\02\123.mp3
```

Valid folder numbers are 01 to 99
Valid file numbers are 001 to 255

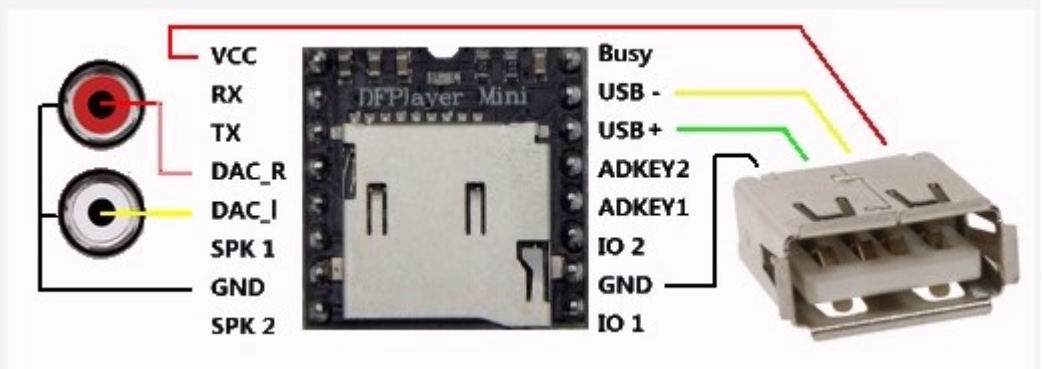
However the MP3 Player also ignores any extra characters after the initial 3 or 4 digits in the filename. Therefore it is also acceptable to add the full song names after the initial 3 or 4 characters if desired e.g. the following filenames are still valid for use in the \MP3 sub folder

Appendix 3 – Using Line Out

To use 'Line out' to an amplifier instead of the 8 ohm speaker the 3 connections GND, DAC-L and DAC_R are available at the bottom centre of the SPE035 PCB (under the MP3 module).

From left to right the 3 pads are

GND DAC_L DAC_R



Appendix 4 – Using a USB Memory Stick

It is assumed that most people will use a microSD memory card to keep the physical size of the unit as small as possible.

However it is also possible to 'hack' the PCB to connect a USB style memory stick instead of the microSD card if desired. The 4 connections required for a USB 'Type A' socket (to accept the USB memory stick) are:

DFPlayer Mini	Pin
Vcc	1
USB -	15
USB +	14
GND	10

Pin	USB Socket
1	Vcc
2	USB -
3	USB +
4	GND

When using a USB memory stick the initialisation command (0x09) sent at the start of the program must now use the data value 00 01 (select USB) instead of 00 02 (select microSD).

The files should still be saved in a folder called \MP3 on the USB memory stick and require the same file naming structure as when using the microSD card.