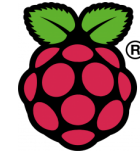


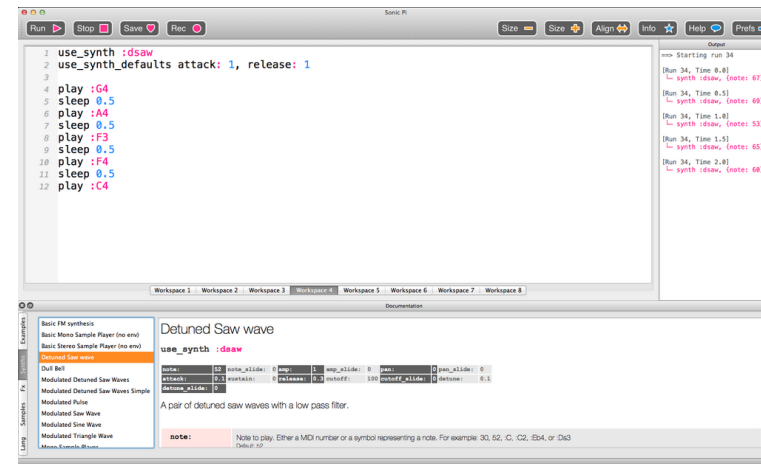


# Sonic Pi



What is it?

Sonic Pi is a music making program written specifically for the Raspberry Pi. It mixes the creation of music with a hint of programming, allowing you to create the next pop/rock/RnB chart topper - Pet Shop Boys eat your heart out! [or something like that].



## Table of Contents

Getting Started

Task 1 - You hum it, I'll play it

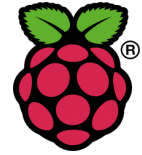
Task 2 - Pure and Synth-le

Task 3 - Play it again Sam

Task 4 - Easy on the Threads, man



# Sonic Pi



I know a song that'll get on your nerves.... Let's get started

Step 1 - Power up and log on (probably done already)

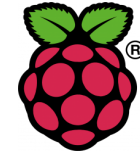
1. If it's not already running plug the micro-USB (phone charger) lead into the connector on the board
2. If prompted for a login use **pi** as the user and **raspberry** as the password, don't panic; you won't see the password as you type, just press the Enter key when you're done
3. Type **startx** and press Enter
4. The "magic" will start to happen and eventually you'll get to a desktop with a giant raspberry on it. Congratulations, Step 1 complete!

Step 2 - Starting Sonic Pi (aka tuning up)

1. If there's not already some headphones plugged in ask for some and plug them in (feel free to use your own, just don't forget them!)
2. Double click the **LXTerminal** icon - a small black console window will appear
3. Type the following then press Enter  
***sonic-pi/bin/sonic-pi***
4. You may find some of the Workspaces already have text in them. If so, select all the text and press Delete
5. Aaaand breathe - you're ready to get your funk on



# Sonic Pi



## Task 1 - You hum it, I'll play it

First off, let's make sure we can play a couple of notes and, more importantly, hear them!

1. Select Workspace 1, delete its content if necessary

2. Type:

```
play 60
```

3. Click on the run icon at the top of the screen.

4. What happens if you type `pley 60` and click on the play icon?

5. Now type:

```
play 60
```

```
play 67
```

```
play 69
```

6. Click on the run icon - What happens?

7. We need to tell the computer to pause between each note. We can do this by typing the following after each `play`:

```
sleep 1
```

8. Now write a sequence of `play` and `sleep` to make a cool sounding tune!

If you can't hear anything and are getting no errors, check the connections!

*Hopefully you should hear a note!*

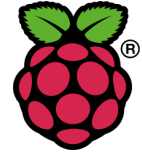
*This is an example of a bug in your code. In later activities if the error panel displays text you will know that you have a bug that you need to fix. It could be that you have misspelled a word like `play`.*

*The computer is playing each note in sequence (one after the other), but it is happening so fast that to us they sound like they are playing at the same time.*

*The value entered after the word `sleep` represents time in seconds. Using the value 1 represents one second. What would you type for half a second?*



# Sonic Pi



## Task 1 - Part 2

Three notes don't make a song (unless you're Status Quo, they only played 3 chords in their songs!). The values that you have been typing after the word `play` represent notes; in fact, they are MIDI note numbers. This means we can translate songs played on a piano into Sonic Pi! Let's try the French nursery rhyme "Frère Jacques". The song begins with:

**C D E C** or **60 62 64 60** in MIDI notes.

1. Select Workspace 2.
2. Type the following code:

```
play 60
sleep 0.5
play 62
sleep 0.5
play 64
sleep 0.5
play 60
sleep 0.5
```

3. Now click on the play icon at the top of the screen and it will play the first part of Frère Jacques\*  
*This first section plays twice. How could you repeat it? You could type the same section out again, or we could start to introduce loops to your code.*

4. At the top of your code, above the first `play 60`, type:

```
2.times do
```

5. And at the bottom of your code, below `sleep 0.5`, type:

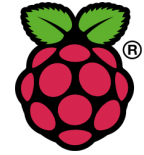
```
end
```

6. Click on the play icon at the top of the screen. What happens?

*Hopefully you should now have the notes repeated*



# Sonic Pi



Your code should look something like this

```
2.times do
  play 60
  sleep 0.5
  play 62
  sleep 0.5
  play 64
  sleep 0.5
  play 60
  sleep 0.5
end
```

*You can see that some lines of code are indented. This makes it easier to read your code and check for any bugs if it does not work when you press the run button. You can press the space bar twice to indent a line of code.*

## What next?

You could complete the song. The notes are as follows:

```
C D E C C D E C
E F G E F G
G A G F E C G A G F E C
C D E C C D E C
```

Notes to Midi Numbers

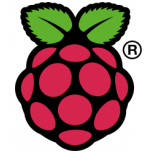
C	D	E	F	G	A	B
60	62	64	65	67	69	71

Or put together a tune of your own

Or move onto the next task



# Sonic Pi



## Task 2 - Pure and Synth-le

It's time to make your tune sound more interesting! We can do this by changing the synthesizer sounds it is using. The default Sonic Pi synth is called `:beep`

1. At the top of your code, above the `2.times do`, add the following

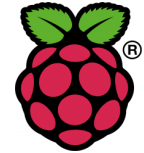
```
use_synth :fm
```

2. Click on the run icon at the top of the screen.
3. Now play with the synths, on the right is a few to try. For a complete list click the **Help** button (top right) and select **Synths**

```
:dull_bell  
:pretty_bell  
:beep  
:dsaw  
:fm  
:prophet  
:pulse  
:saw  
:tri  
:zawa
```



# Sonic Pi



## Task 3 - Play it again Sam

Currently, your tune is playing in a set key; it always plays the same notes from the same octave. To give your program the ability to use a different set of notes we can use a function. A function is just a named group of statements.

1. Open a new workspace in which to store your code.
2. Let's begin by creating a function for the first part of the song:

```
def frere(n)
  play n
  sleep 0.5
  play n + 2
  sleep 0.5
  play n + 4
  sleep 0.5
  play n
end
```

3. To make this now work add the following to the end

```
frere 60
```

4. Change the value of `n` by typing `frere 40` or `frere 80`.

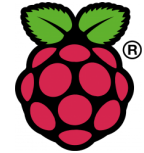
*If you press the run button, nothing will happen. You need to call the function underneath*

*The value of the parameter `n` here is 60, but you can change this.*

*You will hear the tune play in a different key each time.*



# Sonic Pi



## Task 4 - Easy on the Threads, man!

Music often has a repeating backing track, with a separate melody played over the top. So we have played just one tune. Let's try playing two tunes at the same time!

1. Click on a new workspace tab.
2. The code we use to play two tunes at the same time needs to be between `in_thread do` and `end`.
3. Underneath `in_thread do`, type your tune. Let's use Frère Jacques from the earlier activity:

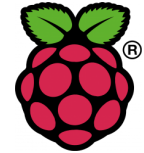
```
in_thread do
  with_synth :beep do
    2.times do
      play 60
      sleep 0.5
      play 62
      sleep 0.5
      play 64
      sleep 0.5
      play 60
      sleep 0.5
    end
  end
end
```

*This first 'thread' will act as the melody of your music. Underneath, you can type the code for your backing track or baseline.*





# Sonic Pi



4. Now add:

```
in_thread do
  with_synth :pretty_bell do
    30.times do
      play 49
      sleep 1
    end
  end
end
```

## Achievement Unlocked!

Assuming everything up to this point has gone according to plan good job, you've hopefully made something that passes for a tune and learnt some basic programming constructs along the way.

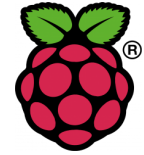
## What's Next?

There's another set of examples you can work through; look for the Sonic Pi article which should be around somewhere (if not, ask). There are also a bunch of examples in the Help section of Sonic Pi, which you can copy/paste and hack away at.

Other than that, do your own thing and have fun! There's a table to convert notes to midi numbers on the next page if you need it



# Sonic Pi



## Full Note to Midi Number Table

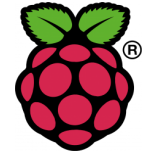
Octave	C	C#/Db	D	D#/Eb	E	F	F#/Gb	G	G#/Ab	A	A#/Bb	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
-1	12	13	14	15	16	17	18	19	20	21	22	23
0	24	25	26	27	28	29	30	31	32	33	34	35
1	36	37	38	39	40	41	42	43	44	45	46	47
2	48	49	50	51	52	53	54	55	56	57	58	59
3	60	61	62	63	64	65	66	67	68	69	70	71
4	72	73	74	75	76	77	78	79	80	81	82	83
5	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	104	106	107
7	108	109	110	111	112	113	114	115	116	117	118	119
8	120	121	122	123	124	125	126	127	-	-	-	-

Source - <http://computermusicresource.com/midikeys.html>

NB: We have been playing in octave 3 (in yellow) in these examples



# Sonic Pi



## Acknowledgements

Raspberry Pi and the Raspberry Pi logo are trademarks owned by the Raspberry Pi Foundation (<http://www.raspberrypi.org>)

Sonic-Pi is the work of Dr Sam Aaron <http://www.sonic-pi.net> Twitter @sonic\_pi

The tasks are reproduced from <http://www.raspberrypi.org/learning/sonic-pi-taster/> and modified where necessary for flow or v2 compatability

Additional resources from <http://issuu.com/themagpi/docs/issue23final/44>

Notes to Midi Numbers table reproduced from <http://computermusicresource.com/midikeys.html>