

Musical Vegetables

Using a Raspberry Pi3, Explorer HAT Pro, Sonic Pi and Edublocks/Python

Worksheet designed and written by Paul Fretwell
Twitter: @drfootleg

In this worksheet we are going to discover how to play sounds using Sonic Pi, by writing Python code using EduBlocks. Then we make a musical instrument out of some vegetables which play sounds when you touch them.

Introduction to Sonic Pi Samples

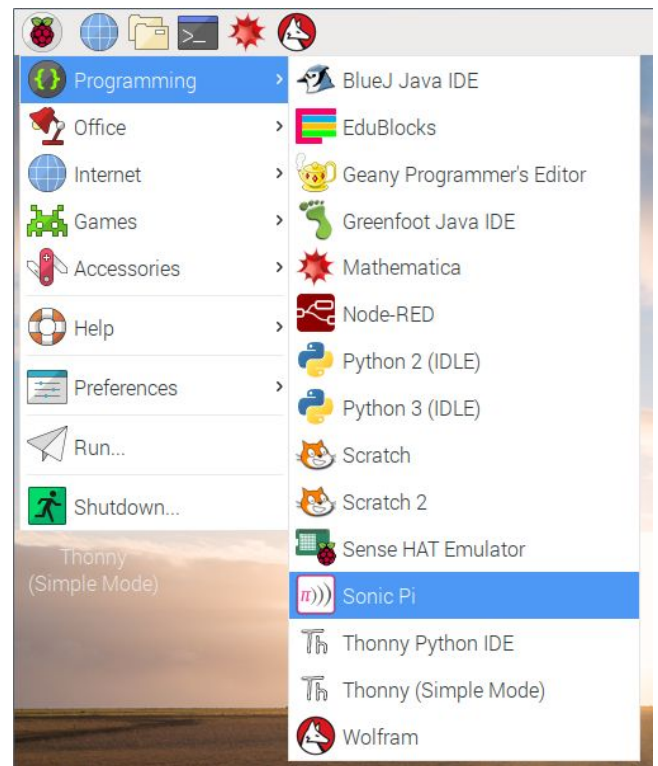
Sonic Pi is a program which enables you to make electronic music from code. You can find it under the Programming menu on the Raspberry Pi.

Launch the Sonic Pi program (you may have to wait for a while as it loads into memory before the main window is displayed, so be patient).

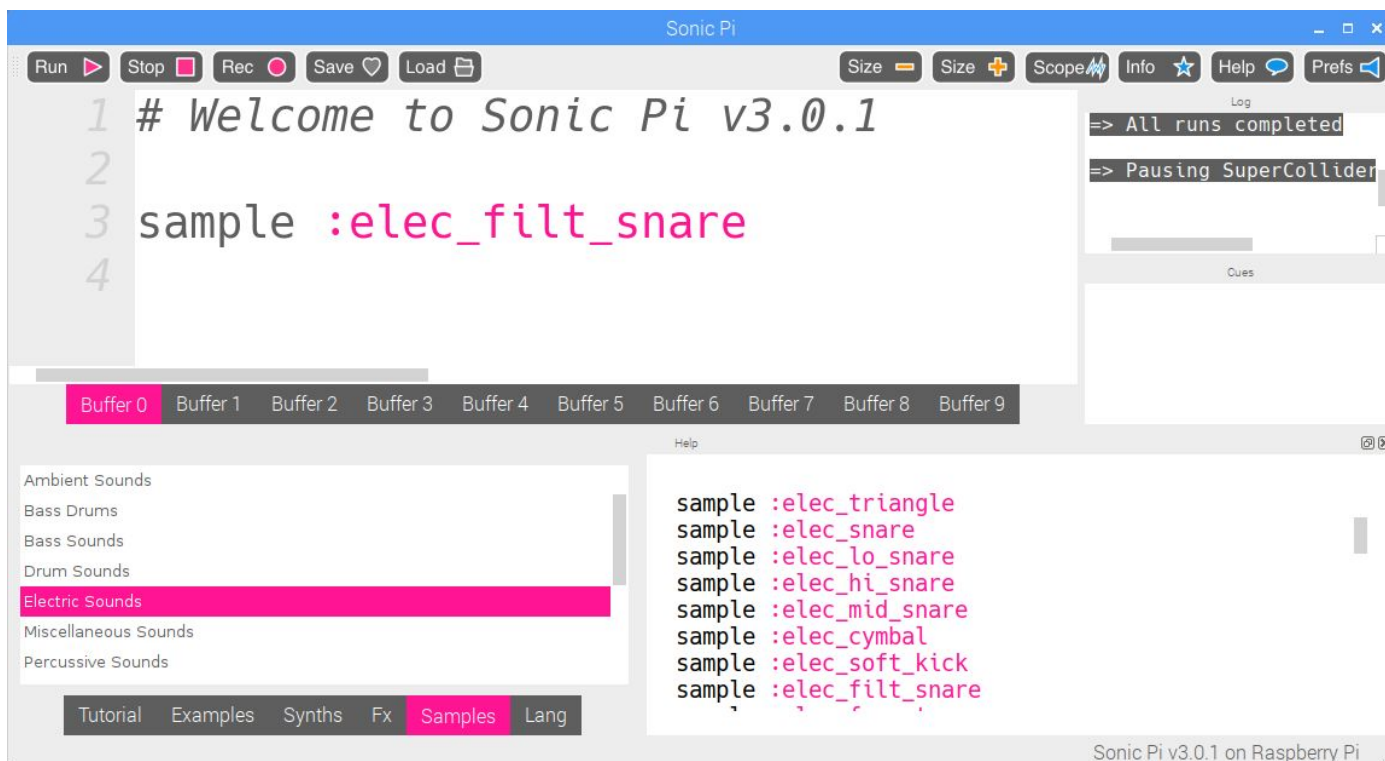
When you see the main window (which looks like the picture on the following page), you should test that the sound is working on your Raspberry Pi.

Type the following code into the coding area, underneath the message *# Welcome to Sonic Pi*.

```
sample :elect_filt_snare
```



Now click on the 'Play' button. If you do not hear any sound then check the volume in the bar at the top of the Raspbian desktop. You can also set the volume in Sonic Pi by clicking on the 'Prefs' button. If you are still not hearing any sound then check you have a speaker plugged in!



Once you have tested that you can play a sound, take a look at the samples provided in Sonic Pi. Click on the **Samples** button on the menu bar near the bottom of the Sonic Pi window (highlighted pink above). You will see the list of categories which the sound samples are grouped under. Here we have clicked on the ‘**Electric Sounds**’ category. The details of the samples are displayed in the bottom right pane of the Sonic Pi window. You need to scroll down the information in the bottom right pane to find a list of the sample names. Any of these sample names can be used to play a sound in Sonic Pi. We will use the same names in our EduBlocks code.

Playing a Sound Sample from EduBlocks

Keep the Sonic Pi window open, otherwise our sounds will not play when we try to run our code in EduBlocks. Launch EduBlocks from the **Programming** menu.

The code blocks we are going to use to play sounds are found in the ‘**Sonic Pi**’ blocks group. You will need to import the psonic code library to use the Sonic Pi code blocks in our programs. Then you can play sound samples using the **sample** block.

```
from psonic import *
sample( "guit_e_fifths" )
```

You can play any sample that is available in Sonic Pi. Here we have chosen the 'guit_e_fifths' sample. Make sure you put the sample name inside double-quotes as shown in the code block above. Run your code in EduBlocks and you should hear your sample sound play through your speakers.

Using a Function

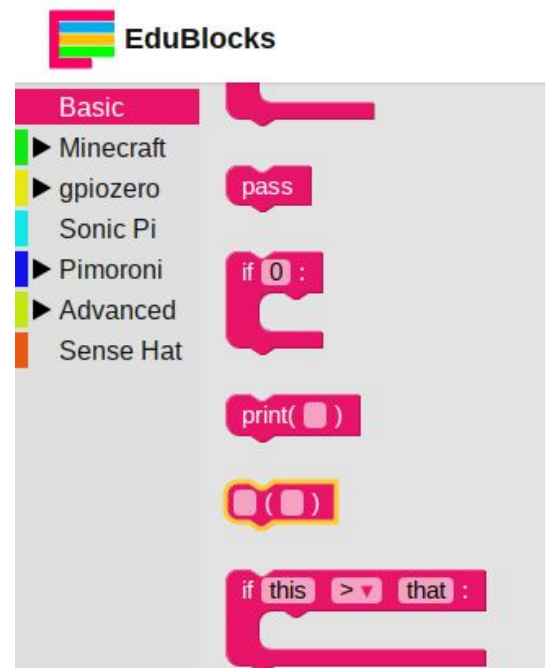
A function is a block of code which you can call by name. Functions are a powerful feature of coding which can be used to make code easier to read, and they allow you to call the same code from different places in your program without needing to write several copies of the same code. This is important because it means you can change the code in the function in one place, and all the places in your program which call that code will now use the updated function code.

In EduBlocks we define functions using a **def** block. Create a function named 'guitar' and put your **sample** block inside this function definition. If you run your code now nothing will happen. You have defined a code function which plays a guitar sample, but your program is not calling the function. You need to add a block to actually call your function, and type the name of your function into this block.

```
from psonic import *
def guitar ( ):
    sample( "guit_e_fifths" )
guitar ( )
```

The block to use for calling a function can be found in the **Basic** blocks group. It is the block highlighted with a yellow border seen in the picture on the right --->

Run the code above to check you hear your sound play. Now we have a function, we can call it more than once in our code!



Before we call our function multiple times, we need to allow the sound to finish playing from the last time we called it. Import the **time** library, which will allow you to use the **time.sleep** block in your program. The guitar sample takes about 2 seconds to finish playing, so set the sleep time to 2 and then call your **guitar** function a second time in your program.

```
from psonic import *  
import time  
def guitar ( ):  
    sample( "guit_e_fifths" )  
guitar ( )  
time.sleep( 2 )  
guitar ( )
```

Run your code, and you should hear the sound sample play twice. What happens if you change the sleep time to less than 2 seconds?

Let's Use a Loop!

When we want to run the same code several times in a row, it is often better to use a loop. Change the code which calls your function and waits for 2 seconds, so that it is contained in a **for** loop. We are using the '**for i in range**' loop here. This will run the code inside the loop the same number of times as we enter inside the brackets after the **range** keyword. The following code will call your **guitar** function 5 times in a row, waiting 2 seconds after each call so the sound sample has time to finish playing before it is called again.

```
from psonic import *  
import time  
def guitar ( ):  
    sample( "guit_e_fifths" )  
for i in range( 5 ):  
    guitar ( )  
    time.sleep( 2 )
```

Have a play with your code. You could try changing the name of the sample, the sleep time or the number of times the loop is run.

Calling a Function from a Touch Event

We can link a function to the touch pads on the Explorer Hat Pro. The function has to match the number of arguments which an event handler for a touch pad requires. Arguments are information sent to a function when it is called. The Explorer Hat Pro code library sends two arguments to touch event handler functions. These are the number of the touch pad which triggered the event, and the type of event. We don't need to use this information in our code, but our function still requires a pair of variables to be declared as arguments for the code to run. So you need to add two arguments in the brackets of the def block. We are using **pad** and **event** here (separated by a comma).

```
def guitar ( pad,event ):  
    sample( "guit_e_fifths" )
```

Before we can use code from the **explorerhat** library in our program, we need to import the library. All the **Explorer Hat** code blocks can be found under the **Pimoroni** blocks group in EduBlocks.

Once you have imported the **explorerhat** library you can use the **explorerhat.touch** block to link a touch event to your function. Set the touch pad number to **one**, the event type to **pressed** and then type the name of your function into the field between the brackets.

We also need to stop our program from ending while it waits for touch events. So you need to add a forever loop at the end. We are using a '**while True:**' block for our forever loop. We do not need to call any more code inside the loop, but we do want to give the computer time to process other events (like updating the screen and responding to the keyboard and mouse!). So add a **time.sleep** block inside the loop with a short sleep time.

```
from psonic import *  
import explorerhat  
import time  
def guitar ( pad,event ):  
    sample( "guit_e_fifths" )  
explorerhat.touch. one . pressed ( guitar )  
while True:  
    time.sleep( 0.05 )
```

You should now be able to run your code. Nothing will appear to happen when it is running, but it is listening for touch events on the pad 1. Touch the pad numbered 1 on the Explorer Hat and you should hear your sound play.

Now you know how to write a code function to play a sound sample, you can try writing several of them. Give each function a different name, and make each one call a different sound sample. You can use the sample names in the example below, or find other samples by looking at the samples lists in the Sonic Pi program. Once you have 4 different functions which each play a different sound, link them to the touch pads numbered 5 to 8. These touch pads work the same way as pads 1 to 4, but they are bare metal which will allow you to connect them to some objects. The objects will then become touch sensors, and touching them will result in your code functions being run.

```
from psonic import *
import explorerhat
import time

def guitar ( pad,event ):
    sample( "guit_e_fifths" )

def choir ( pad,event ):
    sample( "ambi_choir" )

def snare ( pad,event ):
    sample( "elec_lo_snare" )

def cymbal ( pad,event ):
    sample( "elec_cymbal" )

explorerhat.touch. five . pressed ( guitar )
explorerhat.touch. six . pressed ( choir )
explorerhat.touch. seven . pressed ( snare )
explorerhat.touch. eight . pressed ( cymbal )

while True:
    time.sleep( 0.05 )
```

Try running your code, and check that your sounds are played when you touch the pads with the numbers 5 - 8 on the Explorer Hat Pro.

Now we can attach leads to these touch pads and make touch sensitive fruit or vegetables (or any other object which conducts electricity!)