SenseBoard - Light Harp

You will construct an instrument similar to a harp using the SenseBoard. You will use the IR detectors to sense when the notes (or strings on the harp) should be played.

# Setting up the SenseBoards

## You will need:

* 7 or 8 (full octave) SenseBoards each with an IR LED attachment
* Python 3 installed (with the pysense SenseBoard library installed)
* Enough USB ports to plug them all in (a USB hub is recommended)

## Method:

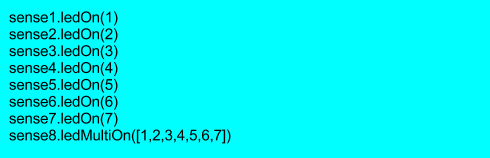
1. Plug in the SenseBoards with USB cables. Line them up in a straight line.
2. From the box named ‘Sensors and infrared LED’ in your SenseBoard Package, take the infrared LEDs (the ones with blue twisted cables) and plug them into the port on the SenseBoards (just above the LED array next to LED number 7)
3. Check each SenseBoard has the red ‘ON’ LED lit up, which is to the right of the USB port on the SenseBoard
4. To make sure that you have the SenseBoards in the correct order, we need to see which SenseBoard is connected in which position. To do this we need to write some python code. In IDLE, or your text editor of choice, open a new file, and save it as **Harp.py** .
   1. To begin we need to tell python what libraries to include. We will be using the, ‘\_thread’, ‘pygame.mixer’ and ‘SenseBoard’ libraries. To include these libraries, type:



* 1. Next we need to make an object for each SenseBoard making it a part of the PySense() class. (The serial port is opened when the class is created)



* 1. Now we will tell each SenseBoard to switch on the LED corresponding to what object it has been defined as, for example tell sense1 to turn on LED 1 and sense2 to turn on LED 2 etc. when we get to sense8 we will turn all of them on



* 1. Finally, save your file. Run the program (press F5 if using IDLE). You should now be able to see it connecting to SenseBoards at serial ports such as COM5. It may take some time to connect to all of the SenseBoards. Once they have all connected look at the LEDs and see if they are in the correct order, if they are, great! If not then rearrange them on the table so that they are in the correct order.

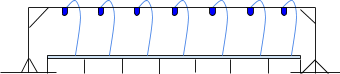
Creating a frame to hold the IR LEDs

## You will need:

* Sticky tape or glue
* Scissors
* Stiff cardboard / wood, or similar

Method:

What you need to make now is a frame to hold the LEDs above the boards. You should make a frame, about 15cm tall and long enough to span all the SenseBoards. You will need to make holes in the top of the frame above each of the IR sensors on the SenseBoards to put the LEDs in. You should make these holes just large enough so that the heads of the LEDs just fit through so that it looks like this



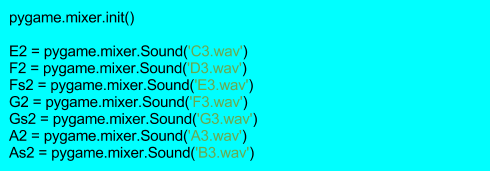
Writing the program to make the harp

Method:

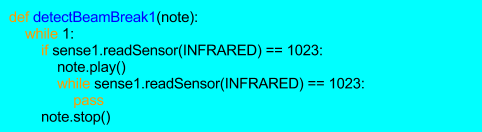
1. The start of the program remains the same apart from where we tell the SenseBoards to turn on LEDs corresponding to their position. We need to change this so that we turn on the IR LEDs, which are labelled LED 7 on the SenseBoards. To do this you need to change the **ledOn()** numbers so that they are all 7, if you have 8 SenseBoards then you need to remove the Multi from **ledMultiOn()** and set the parameters to 7 like the rest.



1. Next we need to initialise **pygame.mixer**, by calling its **init()** method. Then we need to define objects representing the wav files we are going to use. For this example I have just used music notes but you could use any wav file.



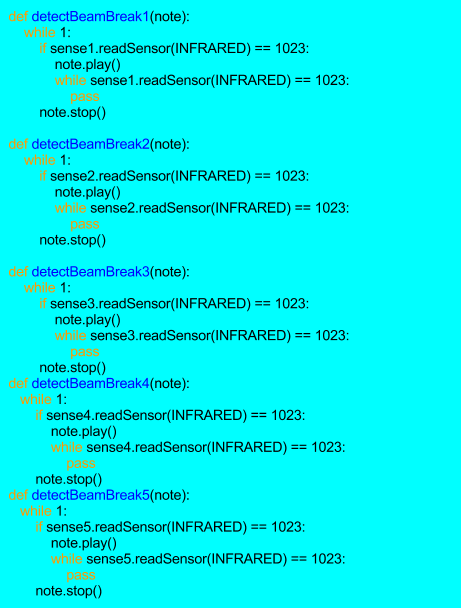
1. Now we need to make a method to constantly check if a SenseBoard is detecting the infrared beam. We will call this method **detectBeamBreak1()** with the parameter **note**. Then we will create an infinite while loop, which will contain an if statement testing if the IR sensor on **sense1** is picking up the IR light from the LED. Inside this if statement we will call **note.play()** to play the file **note** is pointing to, and a while loop testing for IR detection which will just **pass**. Then outside of the if statement we need to call **note.stop()**.

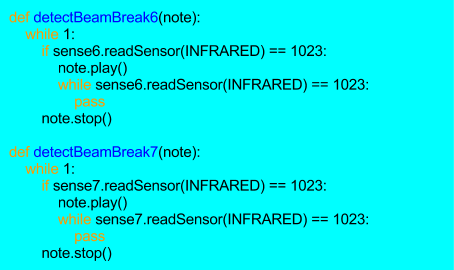


NB: if the SenseBoard detects IR it will return a value of 0 when

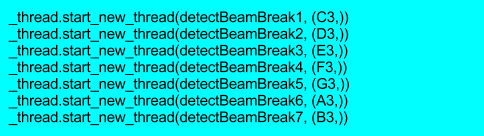
**readSensor(INFRARED)** is called, and a value of 1023 when it isn’t detected.

1. Now we need to create a similar method for all the other SenseBoards, calling them **detectBeamBreak2(),** **detectBeamBreak3()** etc.





1. Next we need to make a new Thread for each method to run in, so that they can all run simultaneously. We will pass in the notes of a C major scale when we create the thread of the method.



1. Finally, the threads will run until the program ends, so to stop the program from ending, we will make an infinite while loop which will run **pass**.



## Thats it, run the program and put your hand between the LED and the Senseboard to break the beam and play the corresponding note.

