SenseBoard - Light Harp

You will construct an instrument similar to a harp using the SenseBoard. We will use the IR detector to sense when the note (or string on the harp) should be played. We will also use the button to simulate the lever or pedal on a harp to adjust the tuning.

# Setting up the SenseBoards

## You will need:

* 7 or 8 (full octave) SenseBoards each with and IR LED attachment
* Python installed (with the pysense SenseBoard library installed)
* Enough USB ports to plug them all in (a USB hub can be used)

## Method:

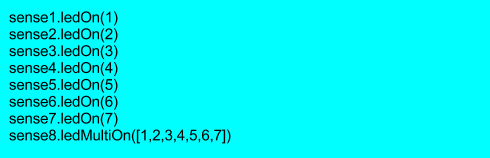
1. Plug in each SenseBoard to a computer with a USB cable. Line them up either in a straight line if you have space, or in two rows
2. From the ‘Sensors and infrared LED’ box within your SenseBoard Package, take the infrared LEDs (blue twisted cable) and plug them into the port on the senseboard (just above the LED array next to LED number 7)
3. Check each SenseBoard should have the red ON LED lit up 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’, ‘winsound’ 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 work. Plug in the SenseBoards and 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 led 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:

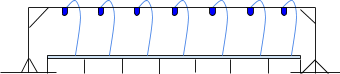
* Sticky Tape or glue
* Scissors
* stiff Cardboard

Method:

What you need to make now is a frame to hold the LEDs above the board. You should make a frame that looks something like this:



You then need to make holes in the top above each of the sensors on the boards to drop the LEDs through. 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:

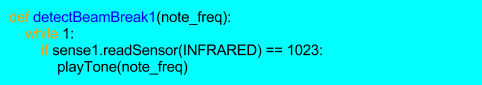
1. The start of the program remains the same apart from where we tell the SenseBoards to turn on LEDs corresponding to their position. We need to change this so that they turn on the IR LED, which is LED 7. To do this you need to change those numbers so that they are all 7, if you have 8 SenseBoards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



1. Next we need to define a new method which will play the sound. We will call this method **playTone()**, and tell it to take in one parameter called **freq**. Then to produce the sound we will use winsound to make a beep, we will set the frequency of the beep to **freq**, and the duration to 1000 milliseconds.



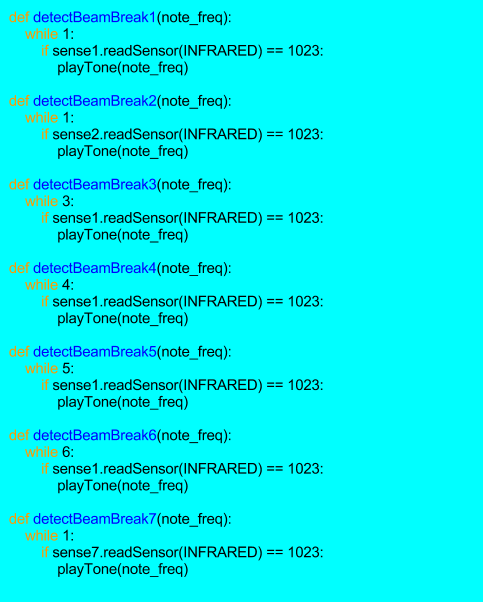
1. Now we need to make a method to constantly check if a SenseBoard is detecting the infrared beam. We will call this method **detectBeamBreak1()** with the parameter **note\_freq**. Then we will create an infinite while loop, which will contain an if statement testing if the IR sensor on **sense1** is picking up the IR from the LED. Inside this if statement will call the **playTone()** method that we made earlier and pass into it **note\_freq**.



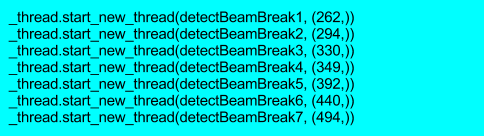
Note that if the SenseBoard detects IR it will return a value of 0 when

**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 **detectBreamBreak2(),** **detectBreamBreak3()** 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 frequencies 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, to run program and put your hand between the LED and the Senseboard to break the beam and play the corresponding note.

