**Arduino – Analogue Piano Coding**

**Arduinos can be used to create many interactive objects. In this example, you will be using an Arduino to create an interactive keyboard.**

**Before you create a piano, you will need to setup the code in which the piano can work. This can be performed by following the instructions on this page. Another page will provide instructions on setting up the electronic side of the piano.**

**You will need:**

* Computer

1

* Arduino Uno
* USB Cable

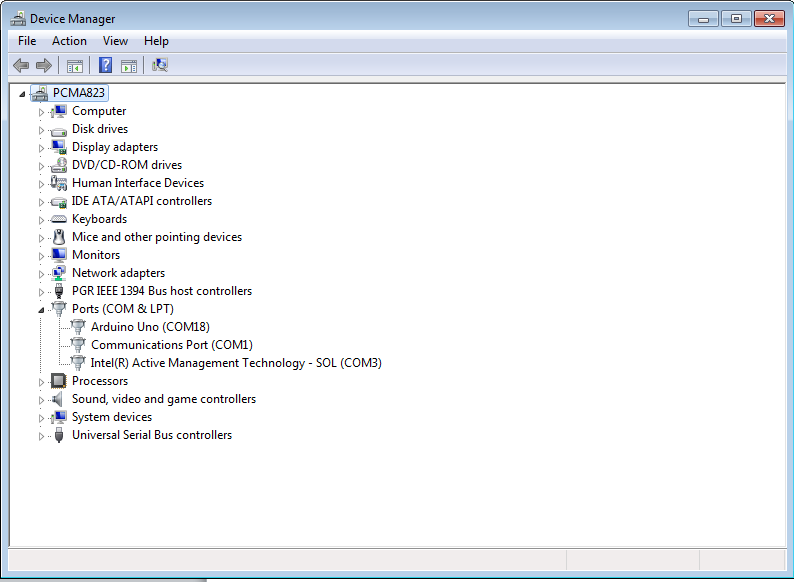
**Connecting the Arduino to your computer**

Firstly, you will need to download software that can interact with the Arduino. This software is free and can be downloaded via this web address:

[*http://arduino.cc/en/Main/Software*](http://arduino.cc/en/Main/Software)

Once you have finished downloading the software, open it.

Connect the Arduino to your computer with a USB Cable. In the Arduino program, select ‘Tools’ -> ‘Serial Port’ and then select the correct COM port which is connected to the Arduino. This can be found in Device Manager.



Device Manager can be found in ‘Control Panel’ -> ‘Device Manager’. The ports can be found under ‘Ports (COM & LPT)’.

**Writing the Code**

2

You don’t need to have previous experience with programming. These instructions will give you the coding needed for the Arduino to function. However, it is important to know that the programming is very case-sensitive, so type in the code carefully.

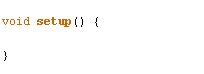
In the Arduino software, type in each piece of code listed below:



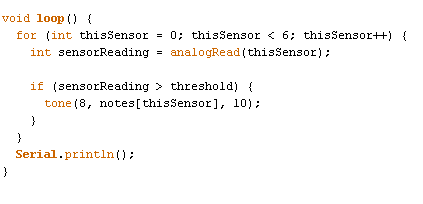
The minimum reading that the sensor needs in order to generate a noise.



Determines what note is played when a button is pressed. Each note will correspond to each button depending on the order of the button.



This code activates the Arduino so that it can interact with the sensors.



This code is in a loop command, this is means that the code will always be on regardless when the Arduino is working. The first part of the code will make the Arduino detect a specific number of sensors, in this case 6 sensors. Then any sensor readings will be converted to a reading that the Arduino can use.

The second part of the code will detect if the reading is greater than the threshold used in the first piece of the code then it will play a tone depending on the order of the sensor.

The third part of the code is not important and has not effect on the Arduino.

**Setting up the notes**

3

Arduino does not have any commands for playing specific notes, it needs the frequencies of each notes in order to play them. This is where the Internet is useful as you will need to copy the code that has frequencies of each note. The instructions on how to do this is explained below:

Firstly, click on the down arrow on the top right in the Arduino software and then click on ‘New Tab’.



You will get the option to name your tab, call it ‘pitches.h’.



Next, you will need to find the codes that contain frequencies for all notes so that the Arduino can play it through the speaker. The link below contains this code, or you could find other sources by searching ‘pitches.h’ in your browser.

[*http://arduino.cc/en/Tutorial/Tone?action=sourceblock&num=2*](http://arduino.cc/en/Tutorial/Tone?action=sourceblock&num=2)

Once you have found the ‘pitches.h’ code. Copy and paste it to the new tab in the Arduino software.

Back in the original tab, you will need to type in an extra command above everything else in that tab.

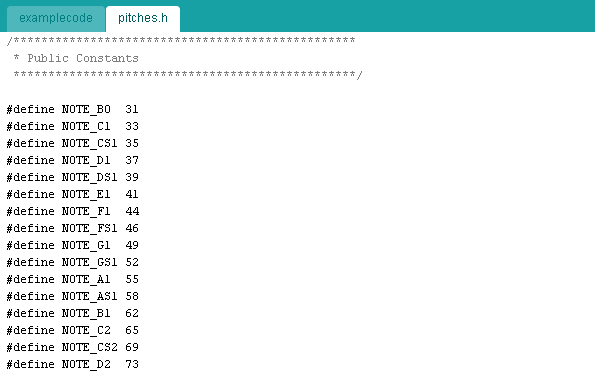
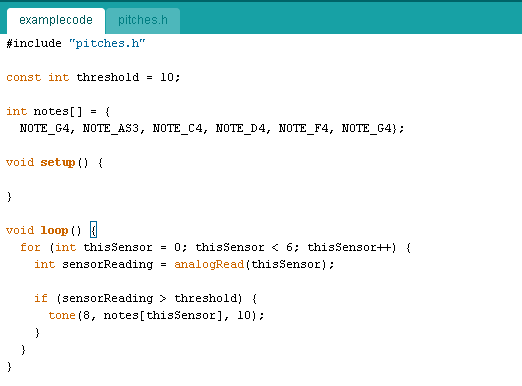


This is used so that the Arduino will include the content in the new tab. The reason why new tabs are used is so that the coding doesn’t get too large and easier changing.

**Uploading the coding to the Arduino**

4

At this point, you should have a finished code that allows the Arduino to play sounds when a sensor is activated. Next, you will need to check that the code can run on the Arduino. The first step is to check your code matches the coding shown below.



**Things to watch out for:**

* **Spelling Errors**
* **Character and Paragraph Spacing**
* **Indentations**

Once, you have found that your coding matches the code shown above with no errors. You can upload the code to the Arduino which will store it onto the Arduino. This allows portable usability as you don’t need to use a computer to run the piano. However, you will still need power from your computer if you don’t have another way to power the Arduino.

To upload the code onto the Arduino, click on the button shown below in the Arduino software:



You will get a loading bar on the bottom of the program which shows the progress of the upload. During this, the program will also check for any mistakes and highlight it if any mistakes are found.

**More things to watch out for:**

* **If the right COM port is used**

Once the software has finished uploading the code onto the Arduino, you will be ready to construct the electronics of the piano.

Sometimes, there might be errors during upload even if the code is correct. This is might because there is nothing for the Arduino to interact with. The best solution to this is to construct the electronics then upload the code afterwards.

\*

**Things that students could do:**

* Allow students to choose what notes to play by removing the notes in the code.
* Remove parts of the code to allow students to finish it off.
* Change the number of buttons available to allow students to manipulate the code to match with the change in sensors by providing an example code.