**Arduino - Theraminimal**

**Arduinos can be used to create many interactive objects. In this example, we will be using an Arduino to create an instrument that utilises a potentiometer in order to change the pitch, and a switch to play the note.**

Length: 1 hour

*Difficulty: Beginner*

**You will need:**

* Computer with Arduino installed (follow the guide at <http://arduino.cc/en/Guide/windows> up to step 5 to install the Arduino software and set it up)
* Breadboard
* Potentiometer
* Button
* 10k ohm resistor
* Speaker or piezoelectric buzzer
* Jump leads or stripped wires
* Arduino Uno
* USB cable

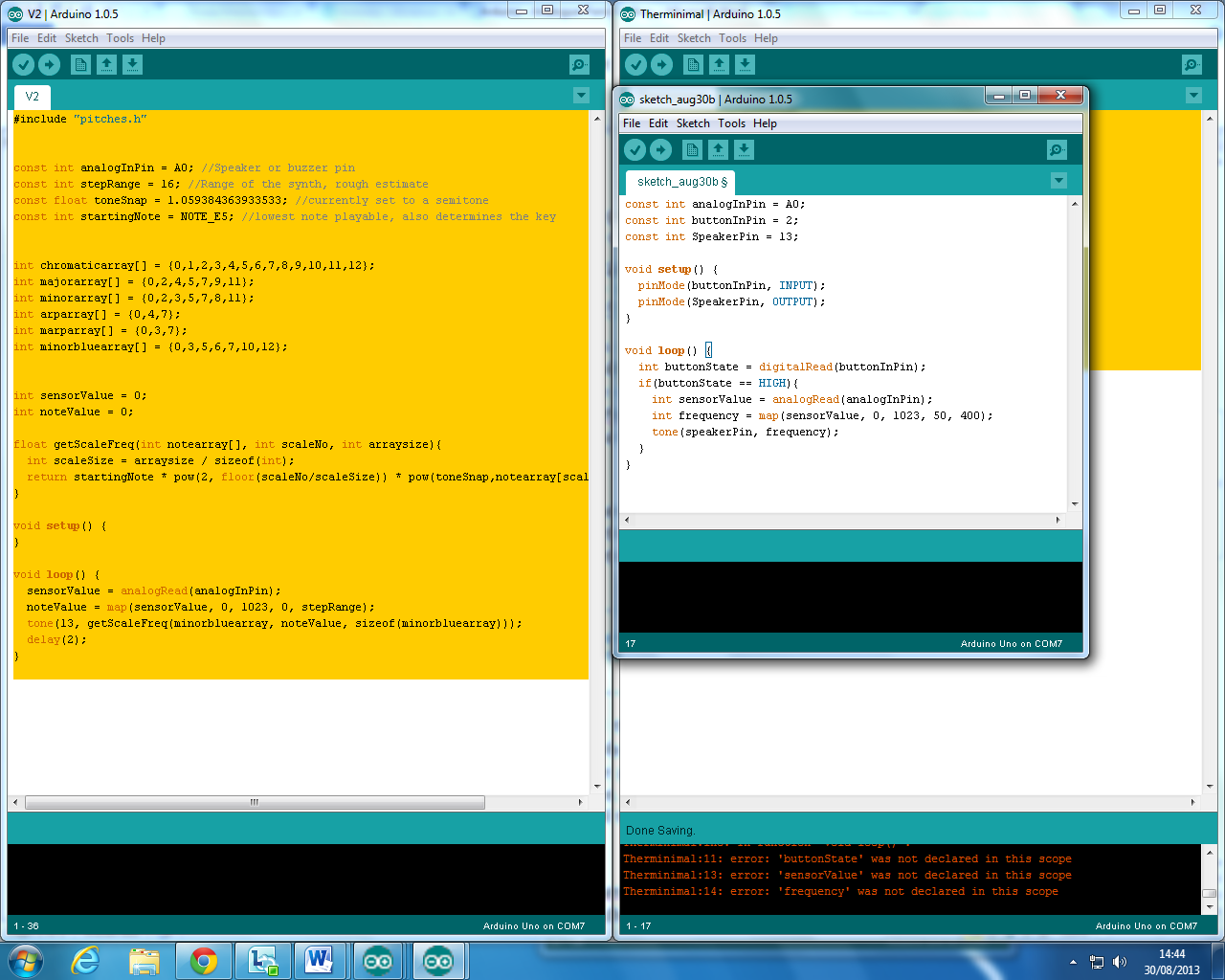
1

**Writing the Code**

Before we create the physical Thereminimal we will first need to write the code for it in order to work. Every one of us uses computers in everyday life, whether it is a desktop, a phone, or even a calculator. Each of these objects would be nothing but shells if they weren’t programmed to do anything.

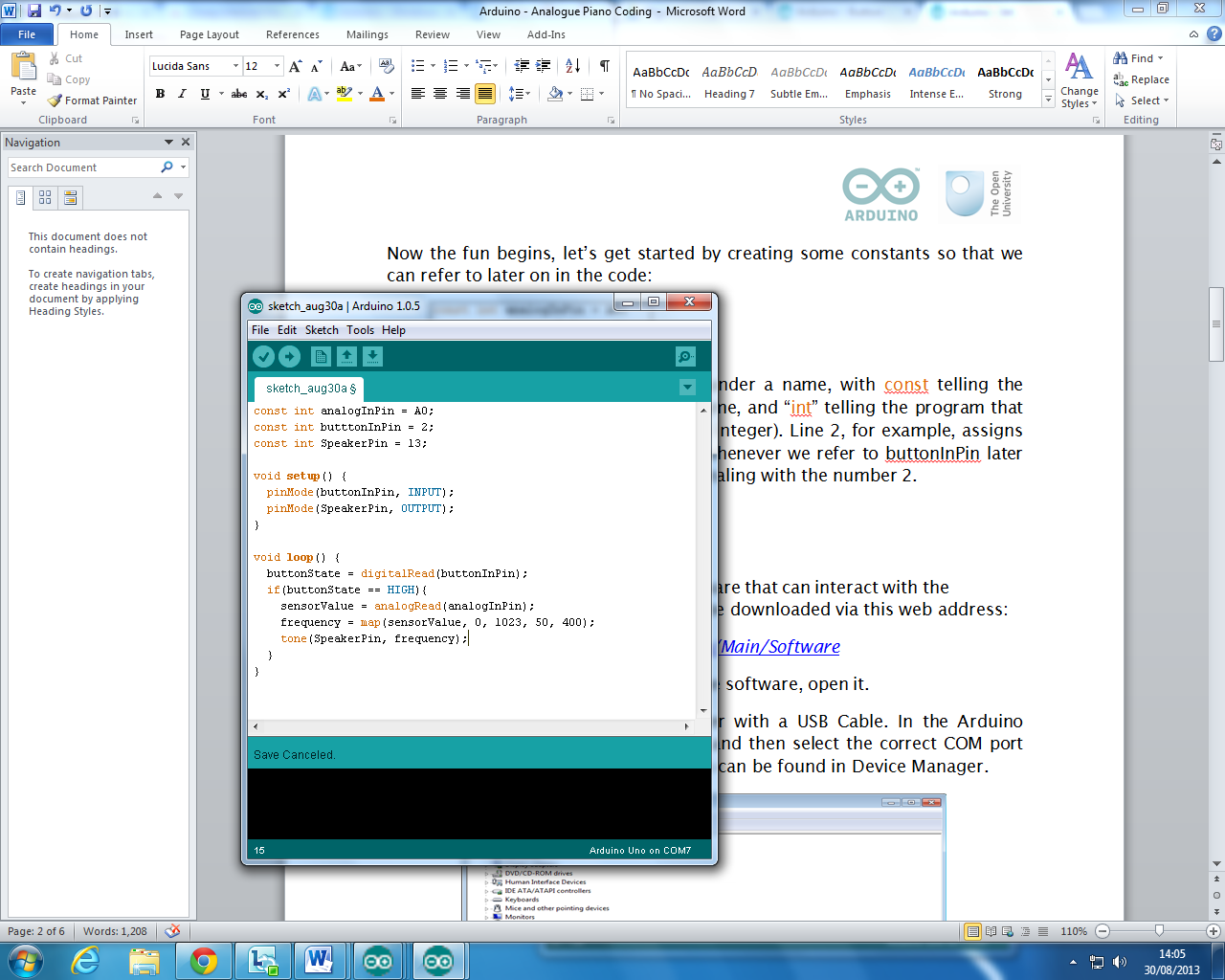
Firstly, open up the Arduino software and it should automatically create a blank project. Now go to the top left menu and click File->Save As, and save your project with a name that you will remember and in a memorable place, I saved mine as “Thereminimal” in my Arduino folder located in Documents, for example. I won’t remind you to continually save throughout the project, however it is a very good idea to save after making any changes (pressing CTRL & s together saves and is quicker than going to the menu). No previous programming experience is required in order to create the Thereminimal, however be aware that the language used to program the Arduino (C) is case sensitive, so remember to type carefully!

Now the fun begins, let’s get started by creating some constants so that we can refer to later on in the code, so type:



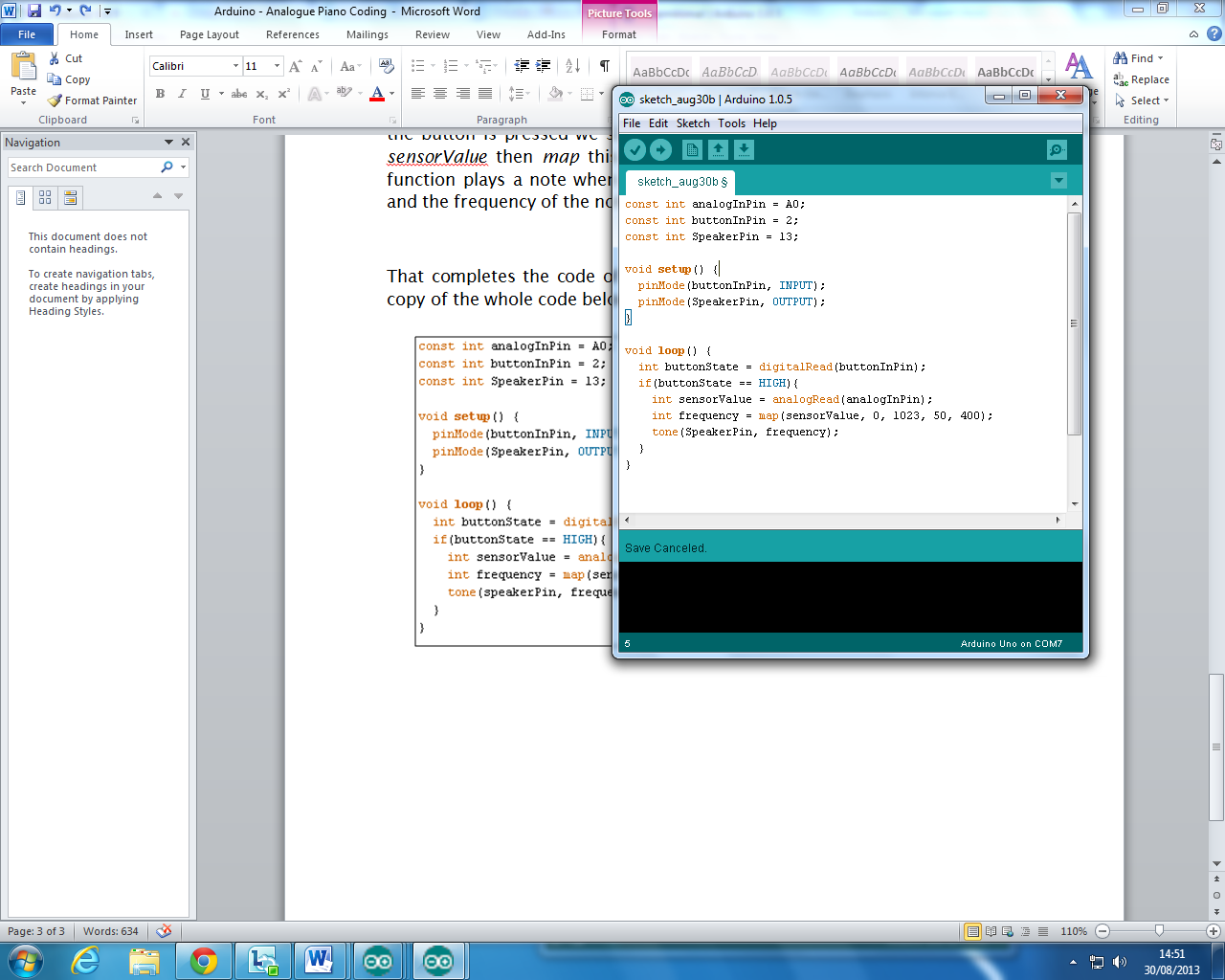
Each of these lines stores a number under a name, with const telling the program that this value will stay the same, and int telling the program that the type of data is a whole number (an integer). Line 2, for example, assigns the number 2 to buttonInPin, so now whenever we refer to buttonInPin later on in the program we will actually be dealing with the number 2.

After setting up the constants we will need to write the code that the Arduino will run automatically at the start, only once. We do this through what is called a “function” or “method”, and there is one called *setup* built into the Arduino:



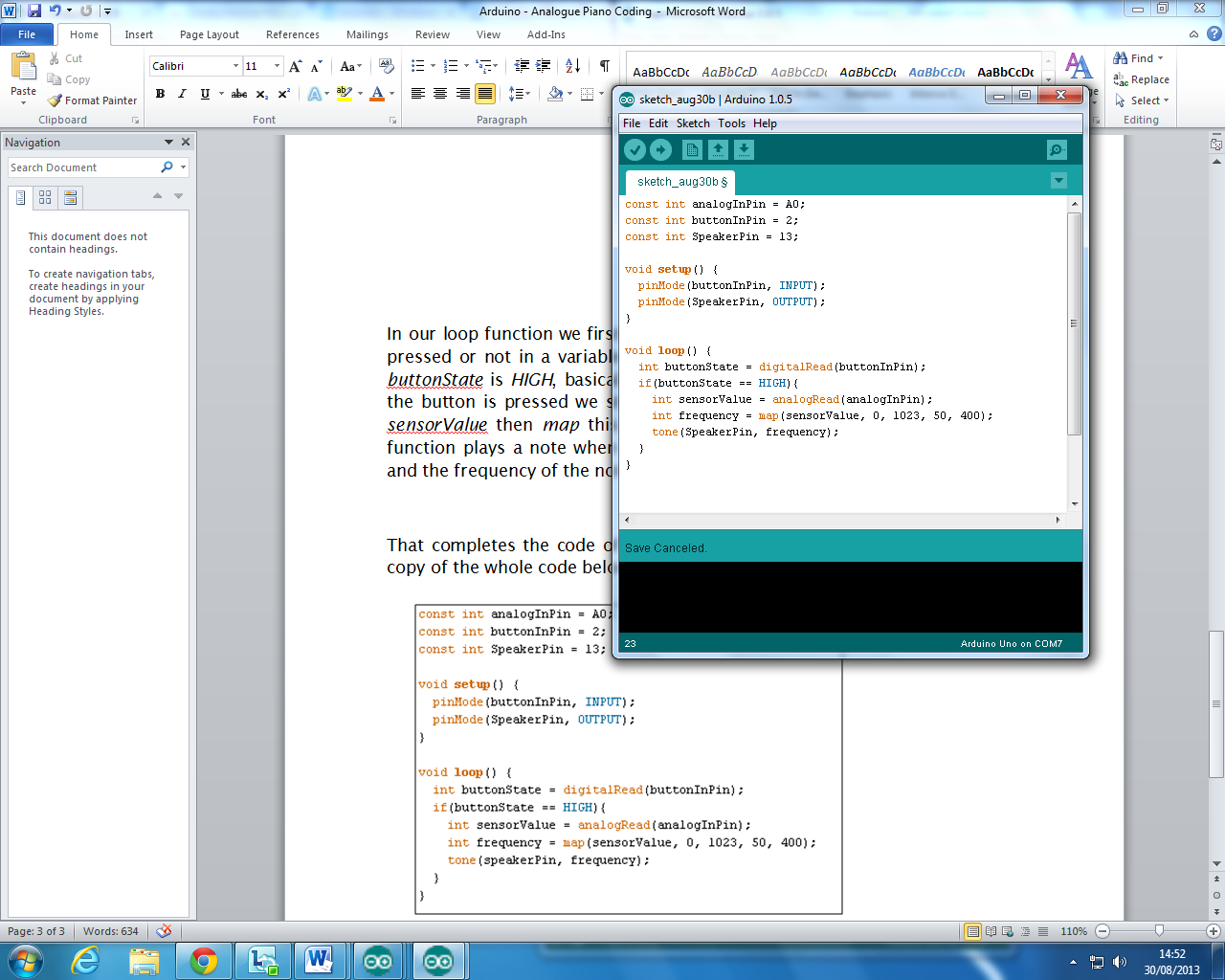
The curly braces just tell the program the boundaries of the function, so when it is called it will only perform tasks written between these two lines, and the void just tells the program that it does not expect anything to be returned from the function. In this case we have two lines within the setup function, both of which tell the program which pins are inputs and outputs.

The last part of our Theraminimal program is a function called *loop*, this is automatically run repeatedly by the Arduino while the program is running, and is generally the place ‘where stuff happens’ in an Arduino program:



In our loop function we firstly tell the Arduino to store whether the button is pressed or not in a variable *buttonState*. Then on the next line we check if *buttonState* is *HIGH*, basically whether the button is pressed or not. Then if the button is pressed we store the value of the potentiometer in a variable *sensorValue* then *map* this value to the range 50 – 400. Finally the tone function plays a note when you give it the pin the speaker is connected to and the frequency of the note to play.

That completes the code of our project, now onto its creation! I will post a copy of the whole code below for reference:

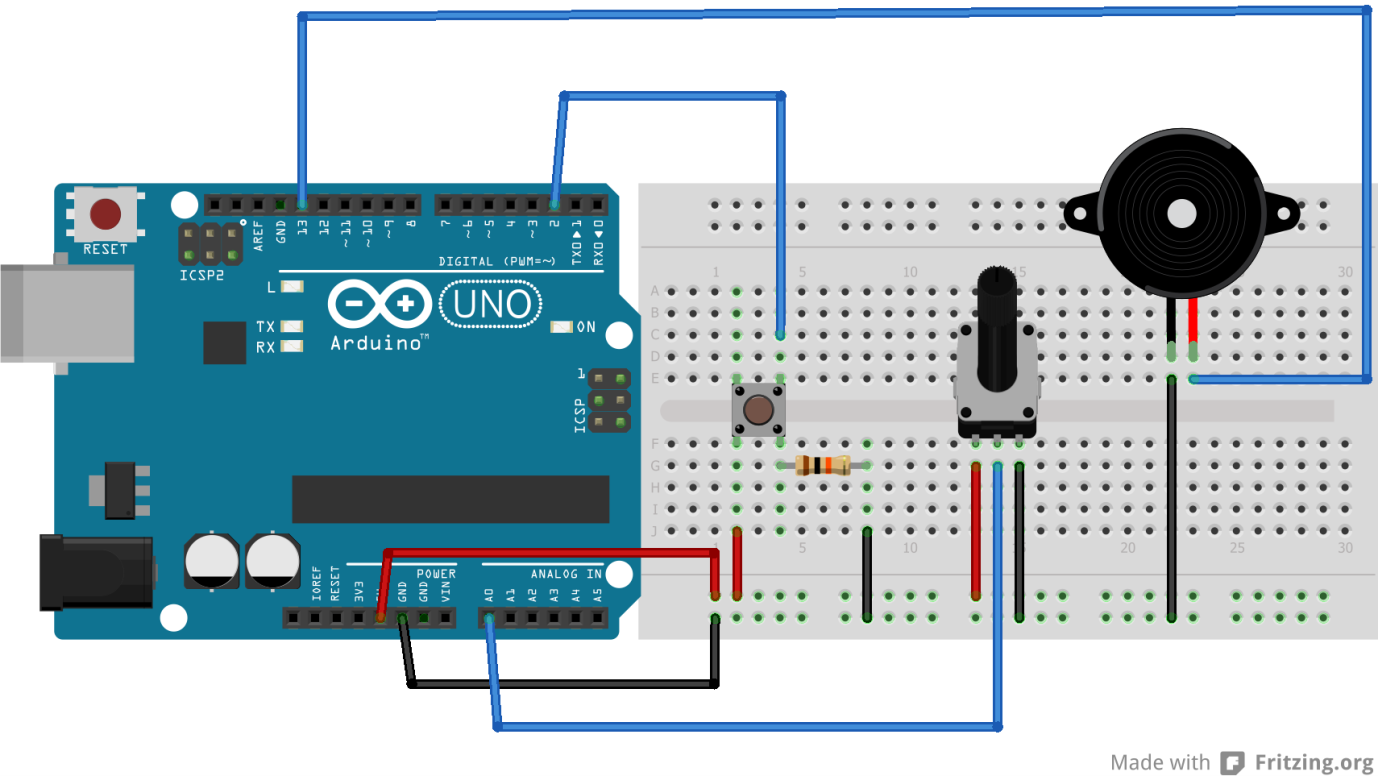


**Creating the Thereminimal**

2

This section is easier to explain using pictures than words, so follow the picture below and assuming you have all the parts listed at the start it should be very simple! Remember to attach the potentiometer to the pin “A0”, the button to pin “2” and the speaker to pin “13” though, otherwise your program won’t work.

The blue wires represent going from a component to a pin, the black wires represent going from a component to ground and the red wires represent going from a component to 5V output.



**Customise!**

3

Well Done! Now you should have a working instrument that plays a note when you press a button, and the pitch of that note should change dependant on the potentiometer, however don’t stop there! There is a variety of different things you can now do to improve or customise your instrument, for example try making it snap to different notes rather than playing continuous frequency, or perhaps try putting a capacitor between the speaker and the Arduino, so that instead of playing a square wave it will play a more rounded wave and produce a different tone.