

# Daisy Tutorial 1 Script

Holland Sersen

September 15, 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Opening Folders in VSCode</b>	<b>3</b>
<b>3</b>	<b>Compile Files to Daisy</b>	<b>3</b>
<b>4</b>	<b>How to get audio in and out of the Daisy Seed</b>	<b>4</b>
<b>5</b>	<b>C++ File Functions</b>	<b>4</b>
<b>6</b>	<b>Compile code and verify that it is working</b>	<b>5</b>
<b>7</b>	<b>Outro</b>	<b>6</b>

# 1 Introduction

Hello my name is Holland Sersen, and today we will be learning how to use the Electrosmith Daisy Seed.

For this tutorial you will need; a Daisy Seed, a breadboard, and an audio jack to get audio from the daisy board. You will also need to follow along the official electrosmith tutorial on how to install the toolchains required to compile the code ([Linked here](#)).

Although this series will go through how to code the Daisy seed basic knowledge of C++ is required to follow along for this tutorial. With that out of the way let get to it.

## 2 Opening Folders in VSCode

First we have to open the folder in VSCode. There are a couple ways of doing this but the easiest way is to open up VSCode, Click File, then Open. Then go to your DaisyExamples Folder on your computer, go into the seed folder, and then oscillator. Then click open.

VSCode should automatically show all the file folders and directories. You should see a couple different files but the one we are interested in is oscillator.cpp. This file is what our Daisy will run. The file defines what our daisy will do.

## 3 Compile Files to Daisy

To compile the code we will have to use the command line.

When you opened this folder VSCode automatically created a terminal in the current folder of your computer. To open this, click the button called “terminal” on the bottom bar. Next, type in the terminal “make”. This will automatically compile the code, it will also tell you if you have any errors in your code or if it can’t compile.

If you get an error check to make sure your code matches the code in the examples on Github, or verify that you have the toolchain installed correctly. Then to upload the code to your Daisy, first hold the boot button and then press reset while it is connected via USB. The led on the board should start to flash. Then type in the console “make -dfu” this should work without trouble.

## 4 How to get audio in and out of the Daisy Seed

Next we need to get audio from the Daisy and make it audible. I would recommend some sort of bread board so you can test circuits on the Daisy. The first and most important circuit is getting audio out from the circuit. Find the pinout of the Daisy from Github, some wires and your audio jack.

TODO: Finish when breadboarding

## 5 C++ File Functions

Now that we got the Daisy working on a basic level let's take a quick look at what the code is doing. First is the main function:

```
int main(void)
{
    // initialize seed hardware and oscillator daisysp module
    float sample_rate;
    seed.Configure();
    seed.Init();
    sample_rate = seed.AudioSampleRate();
    osc.Init(sample_rate);

    // Set parameters for oscillator
    osc.SetWaveform(osc.WAVE_SIN);
    osc.SetFreq(440);
    osc.SetAmp(0.5);

    // start callback
    seed.StartAudio(AudioCallback);

    while(1) {}
}
```

The main function is what the daisy runs, all of the code is defined here in `someway`, and this is the function the daisy will run. If you want to learn

more about main functions there will be a link in the description below. But for this tutorial all you need to know is that the main function sets the frequency of the oscillator and then starts processing audio.

Next we have the process audio block. The Process audio block is where the audio is actually played back and processed. You can see the oscillator is being played here and it is going into the both the left and right audio output.

```
static void AudioCallback(AudioHandle::InterleavingInputBuffer
in, AudioHandle::InterleavingOutputBuffer out, size_t size)
{
    float sig;
    for(size_t i = 0; i < size; i += 2)
    {
        sig = osc.Process();

        // left out
        out[i] = sig;

        // right out
        out[i + 1] = sig;
    }
}
```

## 6 Compile code and verify that it is working

Let's change the oscillator's pitch so that it sounds different than the default. First let's change the `osc.SetFreq()` to 880 instead of 440.

```
//Changed from 440 to 880
osc.SetFreq(880);
```

This should produce a perfect octave above the previous note. So let's compile this to the Daisy and see this for ourselves.

## 7 Outro

With this tutorial hopefully you were able to get a simple sine wave from the Daisy Seed. If you want to learn more there will be links in the description as to how to get better and as always take it easy!