BiVo User Manual

Date: April 27, 2021

Team Name: SongBird

Project Sponsor: Paul Flikkema

Team's Faculty Mentor: Andrew Abraham

Team Members: Kevin Imlay, Daniel Mercado, Yasmin Vega-Nuno, Anqi Wang

Table of Contents

1 Introduction

We are thrilled that you have chosen to use BiVo as the solution to your bird vocalization recording needs. BiVo has been designed as the foundation for your custom sensor system, and is ready for you to deploy and add more functionality. This foundation is designed to record the environment and perform basic analysis for bird vocalizations, send only recordings that may contain bird vocalizations to your computer, and bring them into MATLAB for you to view the spectrogram and listen to. The purpose of this user manual is to help you, Dr. Paul Flikkema, successfully install, administer, and maintain the BiVo system to start collecting bird vocalizations.

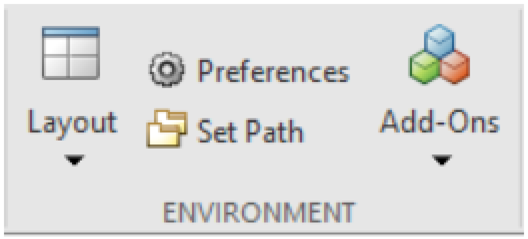
2 Installation

BiVo is made up of a desktop application and sensors. While the desktop application can run on most computers, the sensor is designed to work on the EFM32GG12 Thunderboard. Please follow the instructions below on how to install both parts.

Desktop Application

To install the frontend of the project, follow these steps.

1. Download and install Matlab.
2. Download and install Python 3.8.
3. Open the command prompt and use Pip to install Pyserial.
4. Within Matlab, install the Signal Processing add-on within the Matlab add-on management.

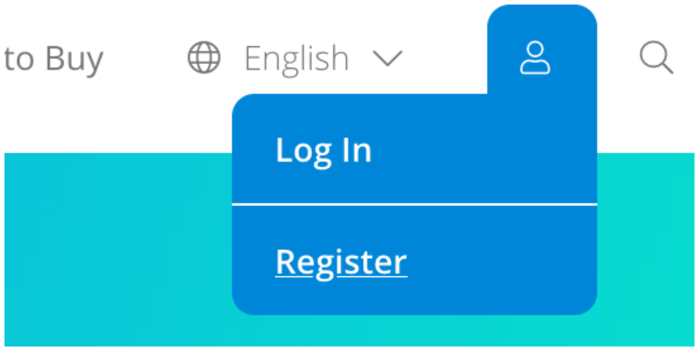


5. Download the source code from the BiVo Github repository.
6. Place the source code download in your Matlab directory in order for it to access the Python Files.
7. Run the .myapp file to load the User Interface.

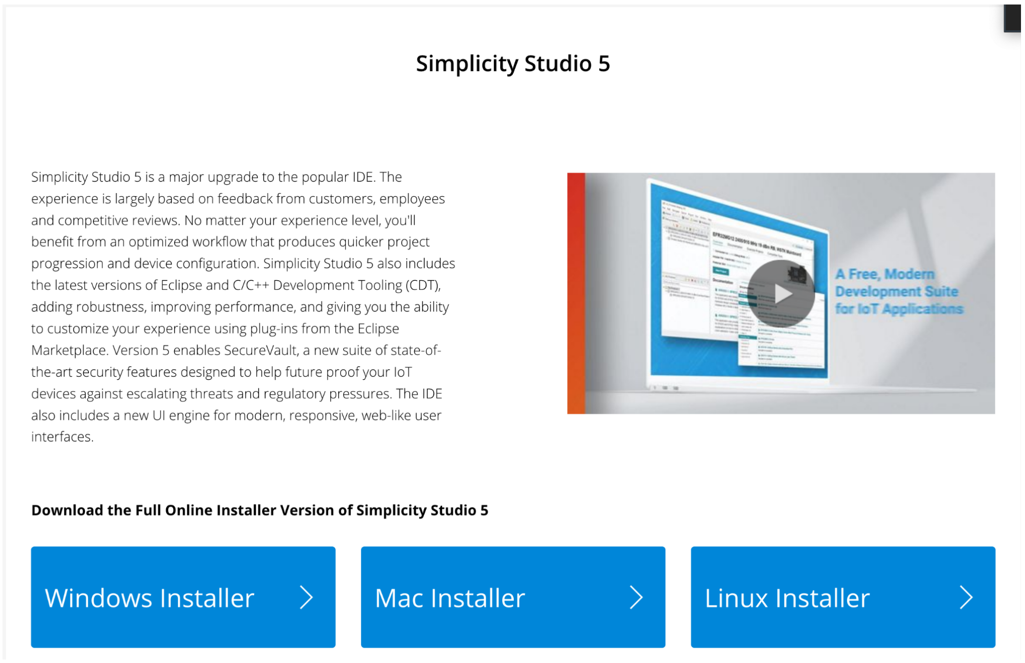
Sensor (Thunderboard)

To install the backend of the project, follow these steps.

1. First, make a Silicon Labs account. Go to Silicon Lab's website at https://www.silabs.com/ and click "Register". Follow the steps to make your account.



2. Then, go to https://www.silabs.com/developers/simplicity-studio and download and install Simplicity Studio 5 appropriate for your system.

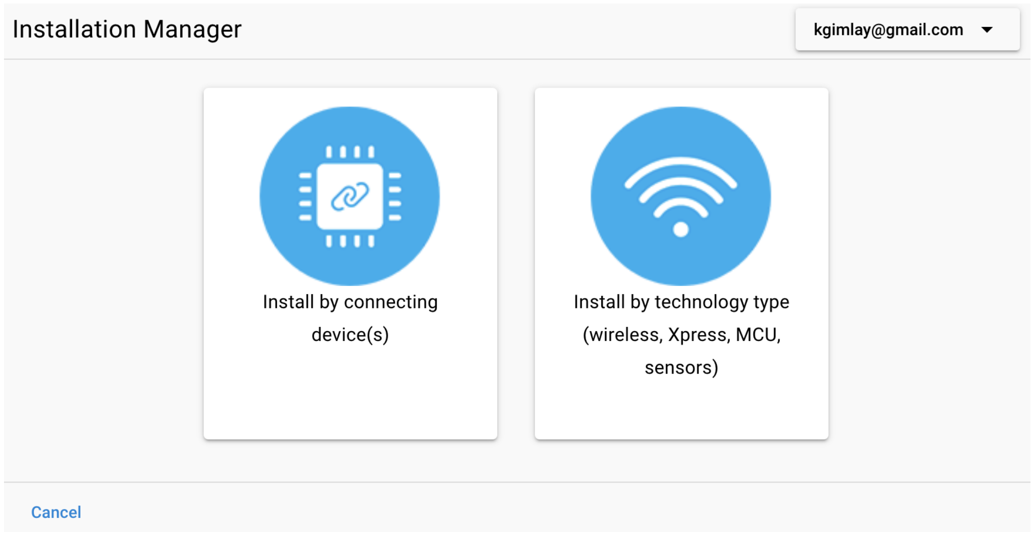


Simplicity Studio 5

Simplicity Studio 5 is a major upgrade to the popular IDE. The experience is largely based on feedback from customers, employees and competitive reviews. No matter your experience level, you'll benefit from an optimized workflow that produces quicker project progression and device configuration. Simplicity Studio 5 also includes the latest versions of Eclipse and C/C++ Development Tooling (CDT), adding robustness, improving performance, and giving you the ability to customize your experience using plug-ins from the Eclipse Marketplace. Version 5 enables SecureVault, a new suite of state-of-the-art security features designed to help future proof your IoT devices against escalating threats and regulatory pressures. The IDE also includes a new UI engine for modern, responsive, web-like user interfaces.

Download the Full Online Installer Version of Simplicity Studio 5

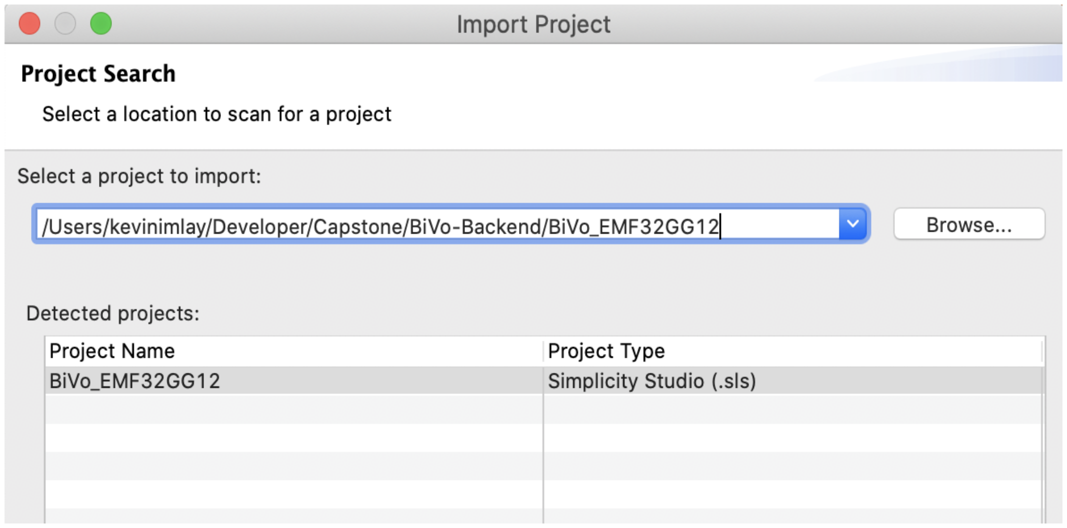
3. Plug in the Thunderboard and follow the prompts to install the Gecko SDK Suite for the EFM32GG12. Instead of plugging in the board, you may also click ‘Install by technology’ and choose ‘32-bit MCUs’. This step may take a long time.


Installation Manager
kgimlay@gmail.com
Install by connecting device(s)
Install by technology type (wireless, Xpress, MCU, sensors)
Cancel

4. Download the BiVo Backend Github repository to a project directory on your computer.

5. From within Simplicity Studio, import the project by going to File>Import and navigating into the project’s top-level folder and selecting open.

6. Select the BiVo_EFM32GG12 sls file and hit ‘next’.


Import Project

Project Search

Select a location to scan for a project

Select a project to import:

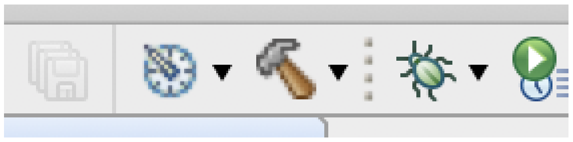
/Users/kevinimlay/Developer/Capstone/BiVo-Backend/BiVo_EMF32GG12

Browse...

Detected projects:

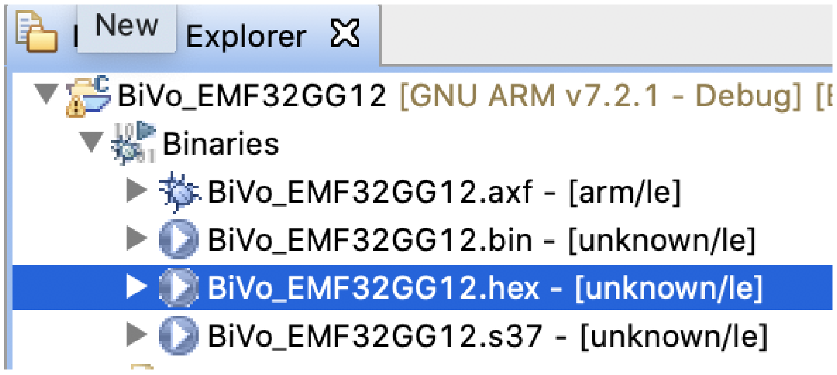
Project Name	Project Type
BiVo_EMF32GG12	Simplicity Studio (.sls)

7. Hit ‘next’ again.
8. Leave ‘Use default location’ unchecked and make sure the directory of your downloaded project is selected in the location field.
9. Hit ‘finish’.
10. If you want to change any sensor settings, see the sensor settings section of daily operation.
11. Build the project by clicking the hammer icon in the toolbar.



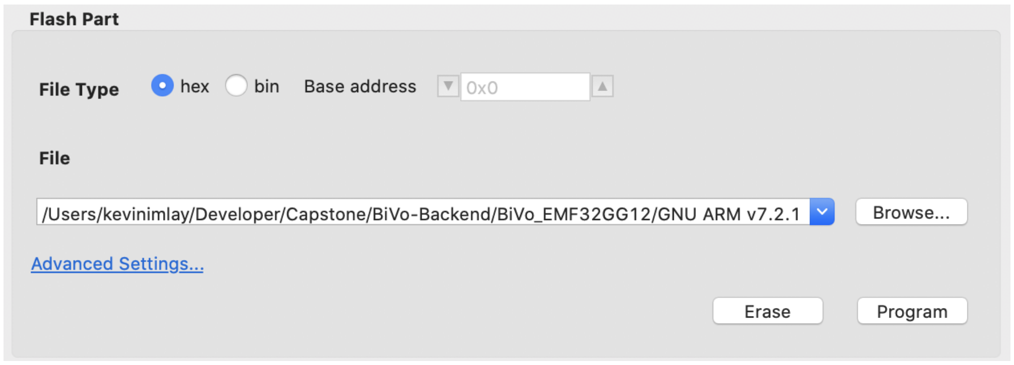
If you encounter build problems, please see the troubleshooting section below.

12. Plug in the board by the micro-USB cable to the DEBUG side of the board and switch the power source switch to ‘DBG’.
13. To flash the project to the board, use the project explorer to navigate to ‘Binaries’, right-click the ‘BiVo_EFM32GG12.hex’, and select ‘Flash to device’.



BiVo_EMF32GG12 [GNU ARM v7.2.1 - Debug]
Binaries
BiVo_EMF32GG12.axf - [arm/le]
BiVo_EMF32GG12.bin - [unknown/le]
BiVo_EMF32GG12.hex - [unknown/le]
BiVo_EMF32GG12.s37 - [unknown/le]

14. Under ‘Flash Part’, hit ‘Program’ and wait for flashing to complete.


Flash Part

File Type [✓ hex] bin Base address 0x0

File

/Users/kevinimlay/Developer/Capstone/BiVo-Backend/BiVo_EMF32GG12/GNU ARM v7.2.1

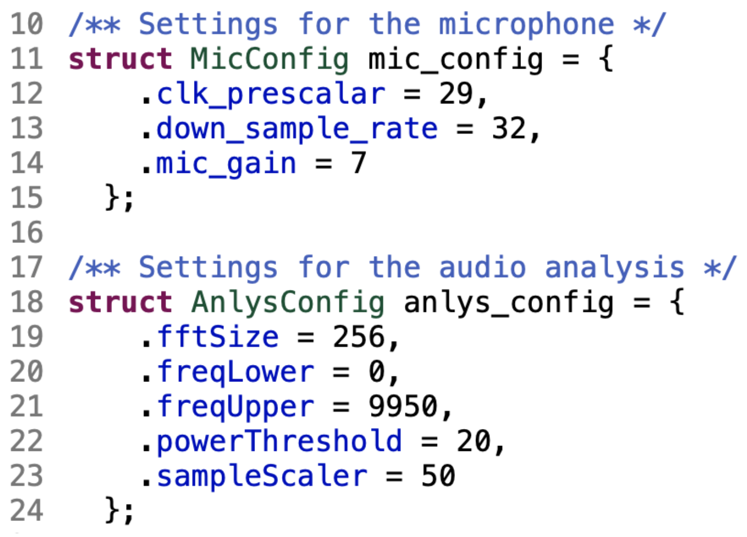
3 Configuration and Daily Operation

In order to get the product deployed and operational, there are a couple of preliminary steps to follow.

- To configure this device you would need to ensure that the program is flashed to the Thunderboard EFM32GG12 as shown in the Installation section.
- To use this device you will need to have the Thunderboard EFM32GG12 plugged into the computer and the User Interface opened. The only way that you can record segments is having the Thunderboard plugged in however if you want to use the spectrogram feature you do not need to plug in the board.

Changing sensor settings, use the file explorer to navigate to Operation Modes > Standard Mode > standard_mode.c and standard_mode.h.

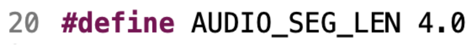
Settings in standard_mode.c:



```c
/** Settings for the microphone */
struct MicConfig mic_config = {
    .clk_prescalar = 29,
    .down_sample_rate = 32,
    .mic_gain = 7
  };

/** Settings for the audio analysis */
struct AnlysConfig anlys_config = {
    .fftSize = 256,
    .freqLower = 0,
    .freqUpper = 9950,
    .powerThreshold = 20,
    .sampleScaler = 50
  };
```

Setting in standard_mode.h:



```c
#define AUDIO_SEG_LEN 4.0
```

- To change the segment length, set the macro called ‘AUDIO_SEG_LEN’ in the standard_mode.h file. The Thunderboard is not capable of segment lengths that result in a buffer much longer than 79600 samples. Please follow the formula:

  $$buffer\ size\ =\ segment\ length\ in\ seconds\ *\ sample\ rate$$

  See (b) below for finding the sample rate. Note that lower down sample rates require higher microphone gain for the same effect.

- To change microphone settings, change the structure variables in the ‘mic_config’ struct on line 11. The sample rate can be found by the following formula:

  $$sample\ rate\ =\ 19104000\ /\ (clock\ prescaler\ +\ 1)\ /down\ sample\ rate$$

  Be careful when setting the clock prescaler and down sample rate as they can change the number of samples in the audio buffer and make it too large! See (a) above.

- To change analysis settings, change the structure variables in ‘anlys_config’ struct on line 18. The fft size variable must be a power of 2 between 32 and 4096. For higher frequency bin resolution, set this higher. To change the frequency bounds on which to test against the threshold, change the ‘freqLower’ and ‘freqUpper’ structure variables. To change the threshold by which a frequency bin needs to meet to pass analysis, change the ‘powerThreshold’ structure variable. The ‘sampleScaler’ variable multiplies each sample by that value. This helps bring out smaller sounds from the audio recorded, but will require the threshold to be raised as well to have the same effect.

4 Maintenance

Desktop Application

Long-term use of the desktop application brings the need for maintenance. The main thing to keep in mind is to get rid of unnecessary audio files.

As of the date of this document, the desktop application creates one .wav file for every 4 seconds of audio recorded. The storage these files take can add up when recording for lengthy periods of time. In order to avoid the hassle of a crowded directory of .wav files, it is recommended to periodically navigate into the directory labeled “Audio” inside of the MATLAB directory, and delete any audio files that may not be needed or useful anymore.

Sensor (Thunderboard)

Board maintenance is also very important. We encourage you to place the board in some sort of waterproof and dustproof housing to prevent environmental damage to the board. Adding a dessicant to this housing would also help prevent water damage from humidity. You should clean this housing and replace the desiccant as needed, depending on the environment the board is placed in. This housing should also be designed to allow for sound to pass through to the microphones.

5 Trouble-shooting

It is possible for the user to run into a few problems while working the desktop application. The solutions to these problems are relatively simple and fast.

Desktop Application

If the MATLAB application claims that the Thunderboard is disconnected when it is connected to the computer, or if the MATLAB application appears to not be completing anything, make sure of the following:

- Check the Thunderboard USB port that the USB cable is connected to. The USB should be connected to a port of the Thunderboard labeled “DBG USB”.
- Check the switch next to the port of the Thunderboard labeled “DBG USB”. The switch should be turned towards the label called “DBG USB”.
- Try closing the GUI, unplugging the board, plugging the board back in, and restarting the GUI in that order.
- Make sure that your Thunderboard is flashed with the BiVo Backend repository’s code. Refer to the installation instructions of how to install the backend of the project.
- Make sure you pip installed the dependency “PySerial”.

You may get an error message from the MATLAB application claiming that it is having trouble locating “PythonScripts”. Be sure to check the following:

- Make sure that the directory labeled “PythonScripts” is inside of the MATLAB directory. To avoid unforeseen problems, be sure that the “PythonScripts” directory is not hidden within more directories inside of the MATLAB one.
- If you do not see the “PythonScripts” directory, you may not have installed it from the BiVo Frontend repository. Refer to the

installation instructions of how to install the frontend of the project.

You may get an error message from the MATLAB application claiming that it does not know what the “spectrogram” function is. Consider the following:

- You may not have installed the add-on called “Signal Processing Toolbox”. You can achieve this by navigating to “Add-Ons” within MATLAB, or via the MATLAB installer.

Sensor (Thunderboard)

There are a number of things that can go wrong on the sensor. The majority of these are a direct cause of misconfiguration of sensor settings. Unfortunately, because changing these settings requires changing code, there are tons of possibilities for things to go ary.

Segments are Never Received/Received When Shouldn’t

If audio segments are never forwarded to the user interface, it is possible that there is an issue with the audio analysis settings or the microphone settings. To address this, check the following:

- The freqLower variable should be set lower than the freqUpper variable. The down sample rate and clock prescaler both need to be positive as well.
- Widen the range between the freqLower and freqUpper variables.
- Increase the sampleScaler variable.
- Lower the threshold variable.
- Increase the microphone’s gain variable.

If audio segments are being sent that don't contain bird vocalizations, the analysis may be too sensitive. Try the following to address this.

- Decrease the range between the freqLower and freqUpper variables.
- Decrease the sampleScaler variable.
- Increase the threshold variable.
- Decrease the microphone's gain variable.

Audio Segments Have Poor Quality

If audio segments are too loud/clipping, try the following.

- Decrease the microphone's gain variable.
- Increase the sample scalar or analysis threshold.

If audio segments are too quiet, try the following.

- Increase the microphone's gain variable.
- Decrease the sample scalar or analysis threshold.

Board Not Recognized When Plugged In

Of course, there are other things that can go wrong with the Thunderboard. If the desktop application is not recognizing the board you may try:

- Make sure that you have plugged in to the board on the DEBUG USB port and the power switch is switched to 'DBG'. This is the most common mistake.
- Close the desktop application and unplug the board, then restart the application and plug in the board again.

Build Troubleshooting

Sometimes importing a project doesn't properly import everything. The majority of times a fresh project does not build is due to missing compiler symbols. To fix this, please follow the these steps:

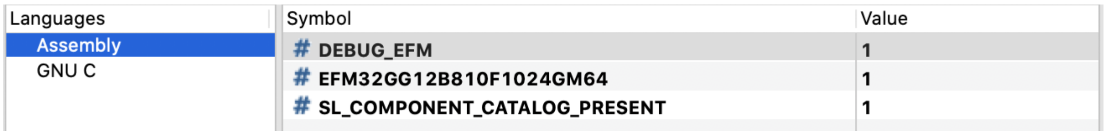
1. Right click on the root project folder in the project explorer.

2. Select 'Properties' at the very bottom.

3. In the sidebar, navigate to C/C++ General > Paths and Symbols. Then select the Symbols tab.

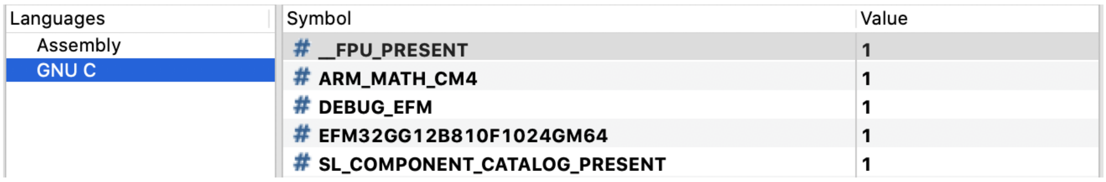
4. On the Languages sidebar, under both Assembly and GNU C, make sure the following symbols are defined:

Assembly tab:



Symbol	Value
# DEBUG_EFM	1
# EFM32GG12B810F1024GM64	1
# SL_COMPONENT_CATALOG_PRESENT	1

GNU C tab:



Symbol	Value
# __FPU_PRESENT	1
# ARM_MATH_CM4	1
# DEBUG_EFM	1
# EFM32GG12B810F1024GM64	1
# SL_COMPONENT_CATALOG_PRESENT	1

5. Click Apply and Close, and try building again.

If none of these seem to fix the issue, you should download a fresh copy of the project and work from there. Follow the installation steps 4-9 of installing the sensor.

6 Conclusion

We are thrilled to have been part of the BiVo project, and hope BiVo will be of good use. We are happy to answer questions in the coming months to help you get BiVo deployed and operating optimally. With best wishes from the SongBird team:

- Kevin Imlay: kgimlay@gmail.com
- Daniel Mercado: 98d.mercado@gmail.com
- Yasmin Vega-Nuno: yasminvega98@gmail.com
- Anqi Wang: aw2393@nau.edu