FEM II GUI User Guide

Author: Ben Edwards

Issue: 1.0

Published:

Contents

[Introduction 3](#_Toc42039317)

[Setting up the GUI 3](#_Toc42039318)

[Using the GUI 4](#_Toc42039319)

[Starting the GUI 4](#_Toc42039320)

[Registers 4](#_Toc42039321)

[Status (Read Only) 4](#_Toc42039322)

[Reset (Write Only) 4](#_Toc42039323)

[Control (Read / Write) 4](#_Toc42039324)

[Selected Flash Device 5](#_Toc42039325)

[Upload FPGA File 5](#_Toc42039326)

[Configure Flash 5](#_Toc42039327)

[Editing the GUI 6](#_Toc42039328)

[Changing FPGA File Location 6](#_Toc42039329)

[Renaming Controls 6](#_Toc42039330)

[Adding Controls 6](#_Toc42039331)

# Introduction

The aim of this document is to provide supporting documentation to the Graphical User Interface (GUI) embedded control system for the FEM-II module. The underlying control system that is used to control and monitor the module is based on a detector control system called ODIN CONTROL, a collaborative project between DIAMOND and STFC. Further details of ODIN CONTROL can be found [here](https://github.com/stfc-aeg/odin-workshop) where a workshop on how to use the software can be found.

## Odin Control

Here we shall discuss a brief outline of odin control concepts.

# Setting up the GUI

To access the FEM-II GUI an instance of Odin server with the FEM GUI adapter must be installed and running on the FEM itself. As such the first step is to open a terminal session on the FEM, by SSHing in to it. In order to successfully install this software, full access rights are required so ensure you are logged in as root.

*ssh username@192.168.0.1*

*su root*

Next you must create a new directory to hold all necessary code and clone the github code repositories for both Odin Server and the GUI adapter, part of the FEM-II embedded systems repository, into this new directory.

*mkdir FEM-II*

*cd FEM-II*

*git clone git@github.com:odin-detector/odin-control.git*

g*it clone git@github.com:stfc-aeg/fem-ii-embedded-system.git*

In order to ensure that this software will not impact any other software, it is best to install it within a virtual environment. Odin Server is designed to run with python version 2.7. As such, next create and activate a new virtual environment then install odin control, followed by setting up the adapter.

*$ virtualenv -p /usr/bin/python2.7 venv*

*$ source venv2.7/bin/activate*

*$ cd odin-control$ python setup.py $ cd ../fem-ii-embedded-system/Odin\_Control\_Adapter/$ python setup.py*

Finally, before running the software ensure that the configuration is set up correctly, as some defaults will likely need to be updated. In particular the default configuration assumes that the FPGA files used to configure the FEM will be stored in the root/FPGA directory

*Insert Screenshot*

# Using the GUI

## Starting the GUI

In order to access the GUI, the odin server must be running on the fem-ii module. To start odin server you must first access the FEM via terminal and ensure the user profile has sufficient access rights. The easiest way to ensure this is to use the root identity.

*ssh username@192.168.0.1*

*su root*

To run the server first navigate to the folder where the code has been cloned. If the server and adapter were installed into a virtual environment, then said environment must be activated before the server can be run.

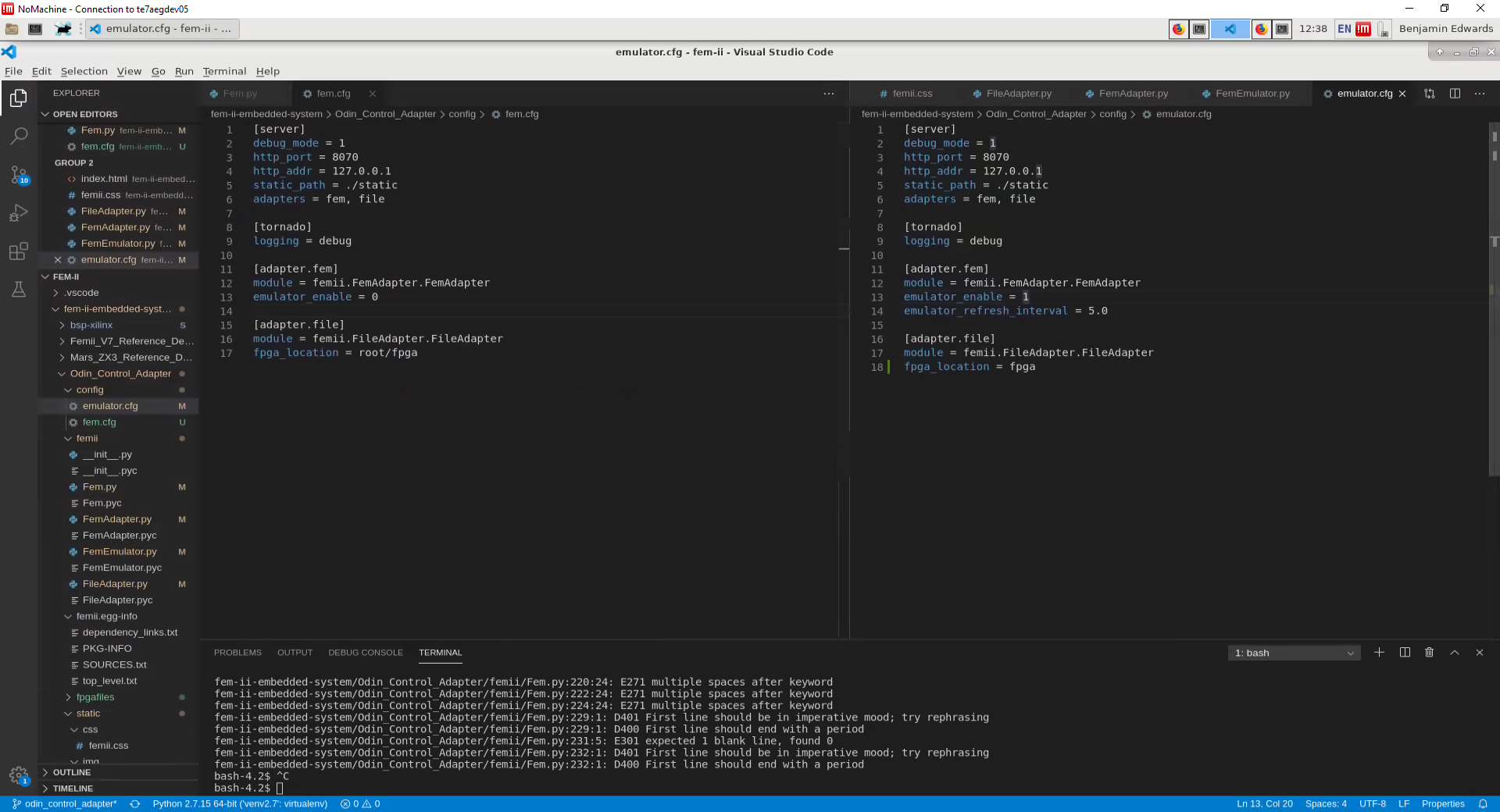
*cd FEM-II*

*$ source venv2.7/bin/activate*

The server can then be run with the configuration file for the adapter as an argument

*$ odin\_server --config Odin\_Control\_Adapter/config/fem.cfg*

Now to use the GUI on any machine that can see the FEM on the network, just open a browser and enter into the address bar the ip address for the FEM, colon then the port designated for the server in the config file (by default 8070)



e.g. 192.168.0.1:8070

## Registers

There are three main registers within the FEM-II firmware that can be read / controlled and written to. These registers are the status, control and reset register.

### Status (Read Only)

This register reads the status of the module and contains the status of the power supplies on the module and whether the FPGA is “DONE”

### Reset (Write Only)

This is a write-only register and when a ‘1’ is sent to this register, the signal on the module is kept “high” or “low” resetting the appropriate logic and or requesting the FPGA to re-program.

### Control (Read / Write)

This register allows control of the FEM-II module and some peripherals. Here you can force the module into low-power mode, forcing off the power to the main FPGA.

#### FEM\_MODE-LPOWER

This puts the module into a low-power state switching of the main processing FPGA but leaving essential power supplies on so that communication via ethernet is maintained

#### FPGA\_FLASH\_CLK

This changes the source clock used for the FLASH devices. When programming the FLASH devices, this is changed automatically so that it can be programmed from the logic that does the programming. When programming is complete, this is automatically changed back so that when the FPA is re-programmed the default clock is used. If this it is not changed back, the FPGA will not be able to program itself from the FLASH memories.

#### FPGA\_FLASH\_SEL0/1

There are four FLASH memories that can be used to program the FPGA. They are selected using two logic signals into a de-multiplexer to ensure that two FLASH memories cannot be selected at the same time.

#### QSFP\_I2C\_BUS\_SEL

There is one I2C bus connected to both QSFP devices. This bit is used to select which I2C bus is enabled on the bus so that both modules do not respond to commands sent on this bus. Addressing is not available on the QSFP devices and hence the need for this setup.

QSFP\_U13/U

### Selected Flash Device

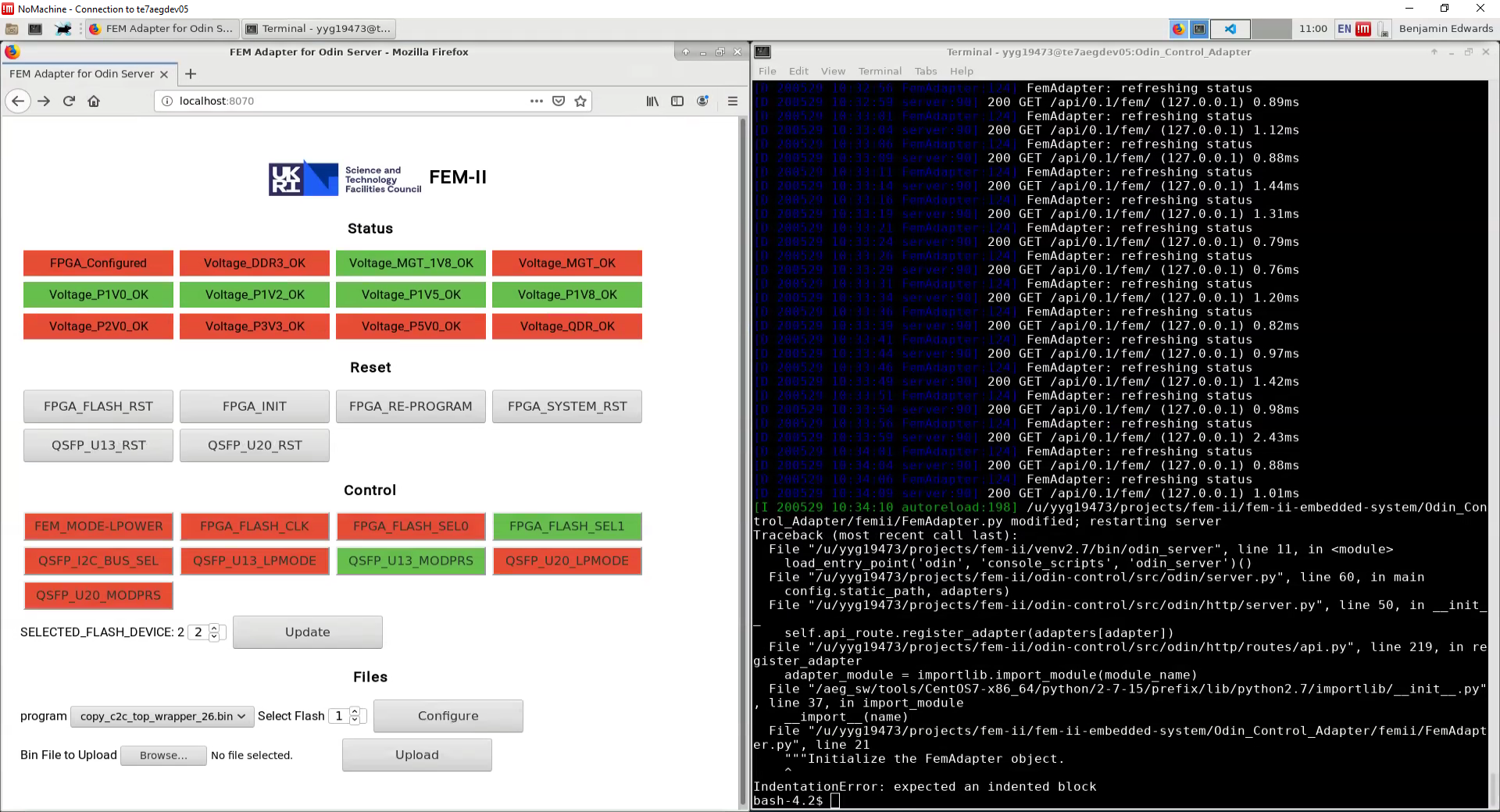


Figure 1

## File Section

### Upload FPGA File

Within the file section there is the ability to upload new bin files to be used in configuring the FPGA. To do so, first use the browse button to find the desired bin file on the machine currently in use accessing the GUI. Now when the upload button is pressed the GUI will enter a disabled state until the system has finished uploading the file to the fem-ii module. Note that this should only take a few seconds unless the file is a few gigabytes or larger. When this process has finished a copy of the bin file will be stored in the FPGA files directory as setup in the adapter config file.

### Configure Flash

Within the file section there is also the ability to configure the FEM with a new bin file. To do so first select the desired bin file from the drop-down list. This list is populated from the FPGA file directory as setup in the adapter config file, and so will automatically update to include any newly uploaded FPGA files as laid out above.

Next select the flash you wish to configure. Please note that the selection is from 1 to 4, as the selection is being done in a human readable format, and not the 0 to 3 it would be in coding terms as determined by the FPGA\_FLASH\_SEL controls.

Finally, once the upload button has been pressed the relevant flash will then be configured with the chosen file. Whilst the configuration is taking place, the GUI will be disabled but a progress bar will keep track of the configuration. Please note that this may take several minutes.

# Editing the GUI

## Changing FPGA File Location

If the FPGA files on the FEM are stored in a different directory on the FEM than that expected by the GUI, then to correct this it is a simple matter of updating the GUI adapters’ config file. This file can be found at

*FEM-II/fem-ii-embedded-system/Odin\_Control\_Adapter/config/fem.cfg*

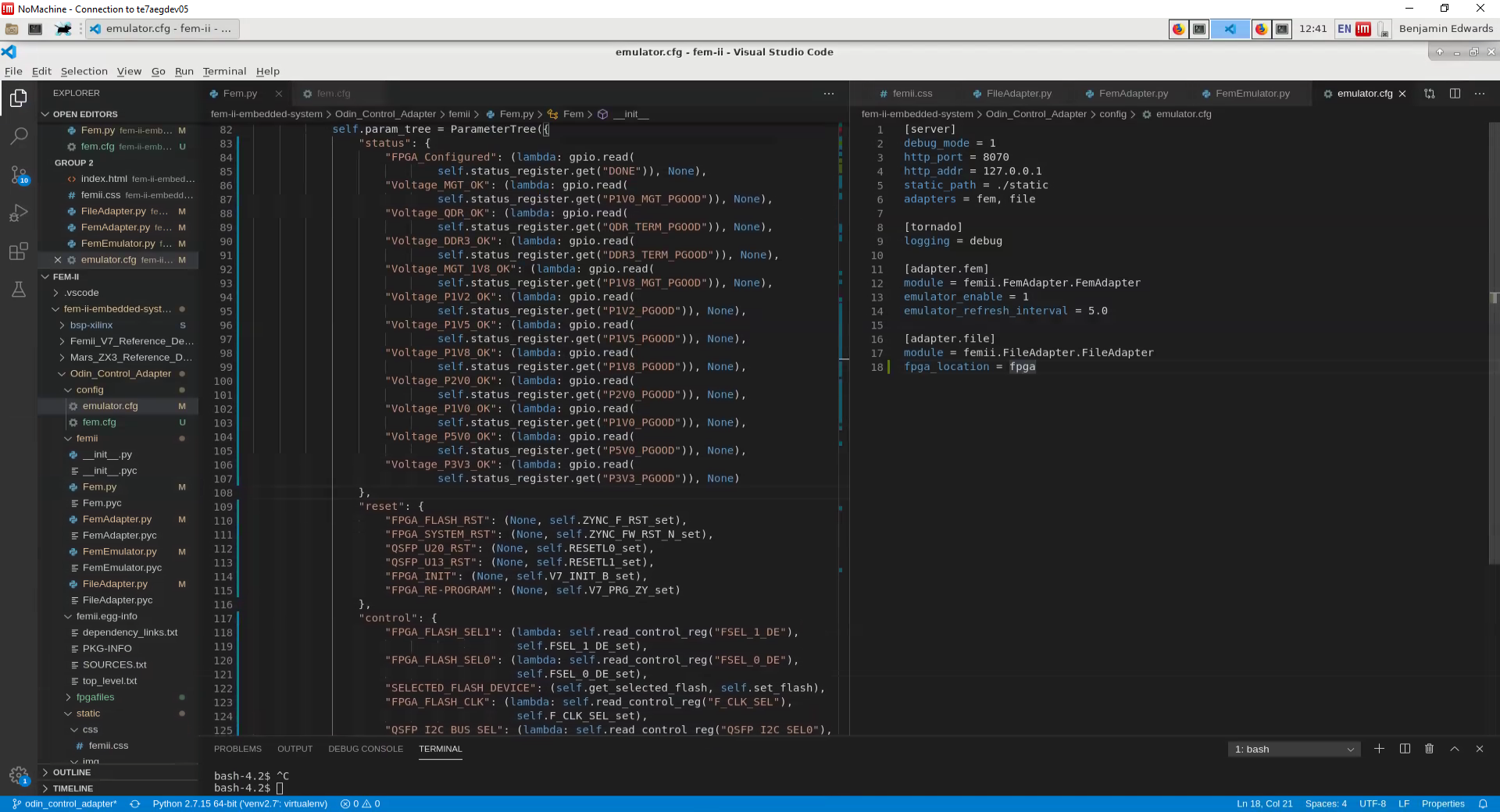
where FEM-II is the base directory where the GUI has been installed. Now within the config file just update the FPGA file location under the file section with the desired directory location for the FPGA files. Please note that unless an absolute file location is entered, the system will use the relative file location to wherever the current directory is at the time the Odin server is started.

## Renaming Registers

In order to change how any of the registers are named in the GUI itself, first open the fem python module responsible for communication between the fem and the adapter.

*vi FEM-II/fem-ii-embedded-system/Odin\_Control\_Adapter/fem\_ii/fem.py*

Now within this file navigate to the parameter tree which starts around line 80.



As shown above, the parameter tree is divided into the same sections as the GUI, so it should be easy to find the register name you wish to update. Now simply change the name as it first appears in the line of code. Please note that the different name used later in the line is the name for the register as outlined in the hardware schematics which is used elsewhere in the code and so should remain unchanged.

## Adding Registers

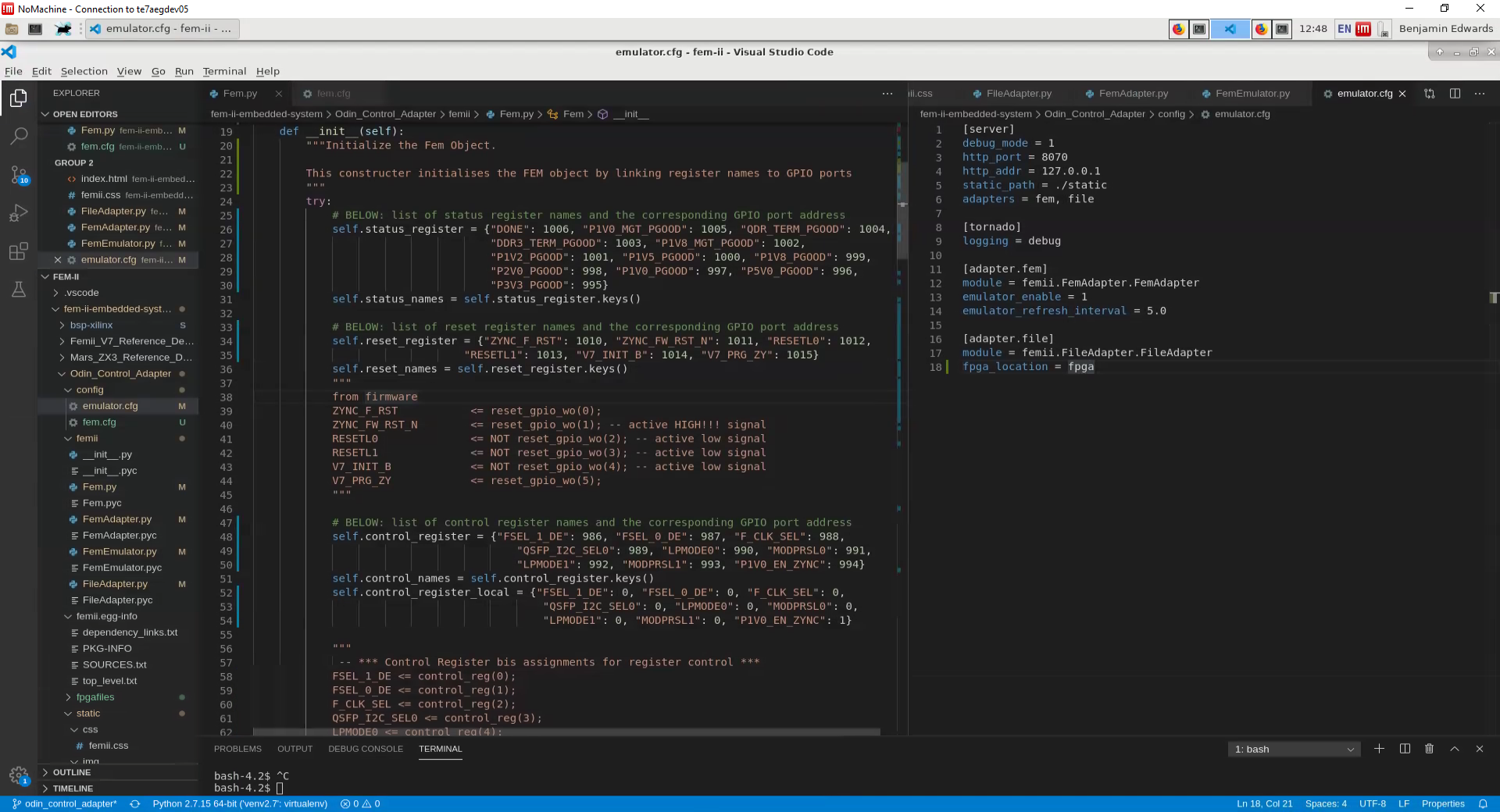
In order to add a new register to the GUI, first ensure that the GPIO address is known, alongside whether the control is read only (Status), write only (Reset), or read/write (Control), a name meaningful to users of the GUI, and optionally a separate name that more clearly links to the hardware itself.

Now open the fem python module responsible for communication between the fem and the adapter.

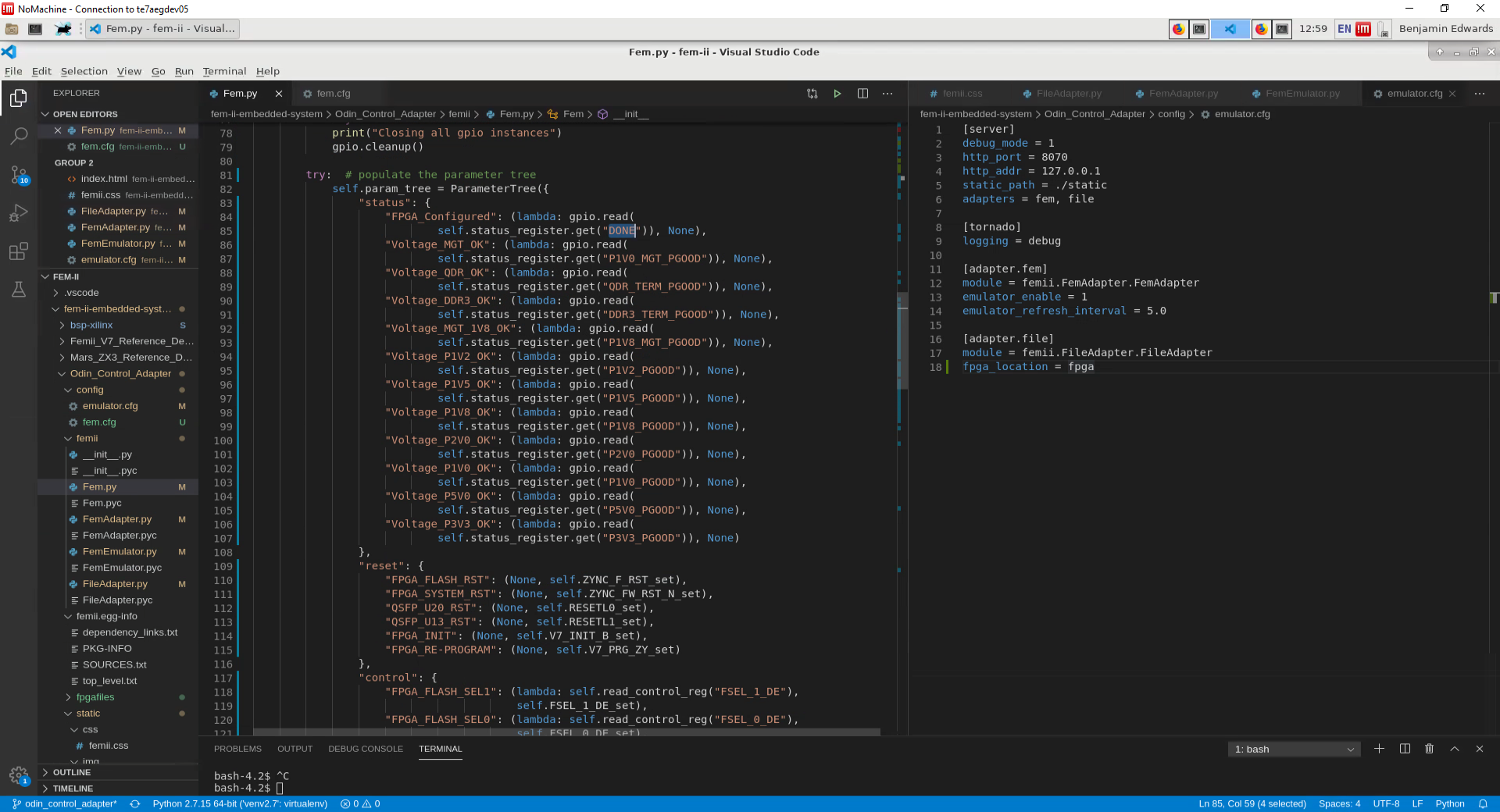
*vi FEM-II/fem-ii-embedded-system/Odin\_Control\_Adapter/fem\_ii/fem.py*

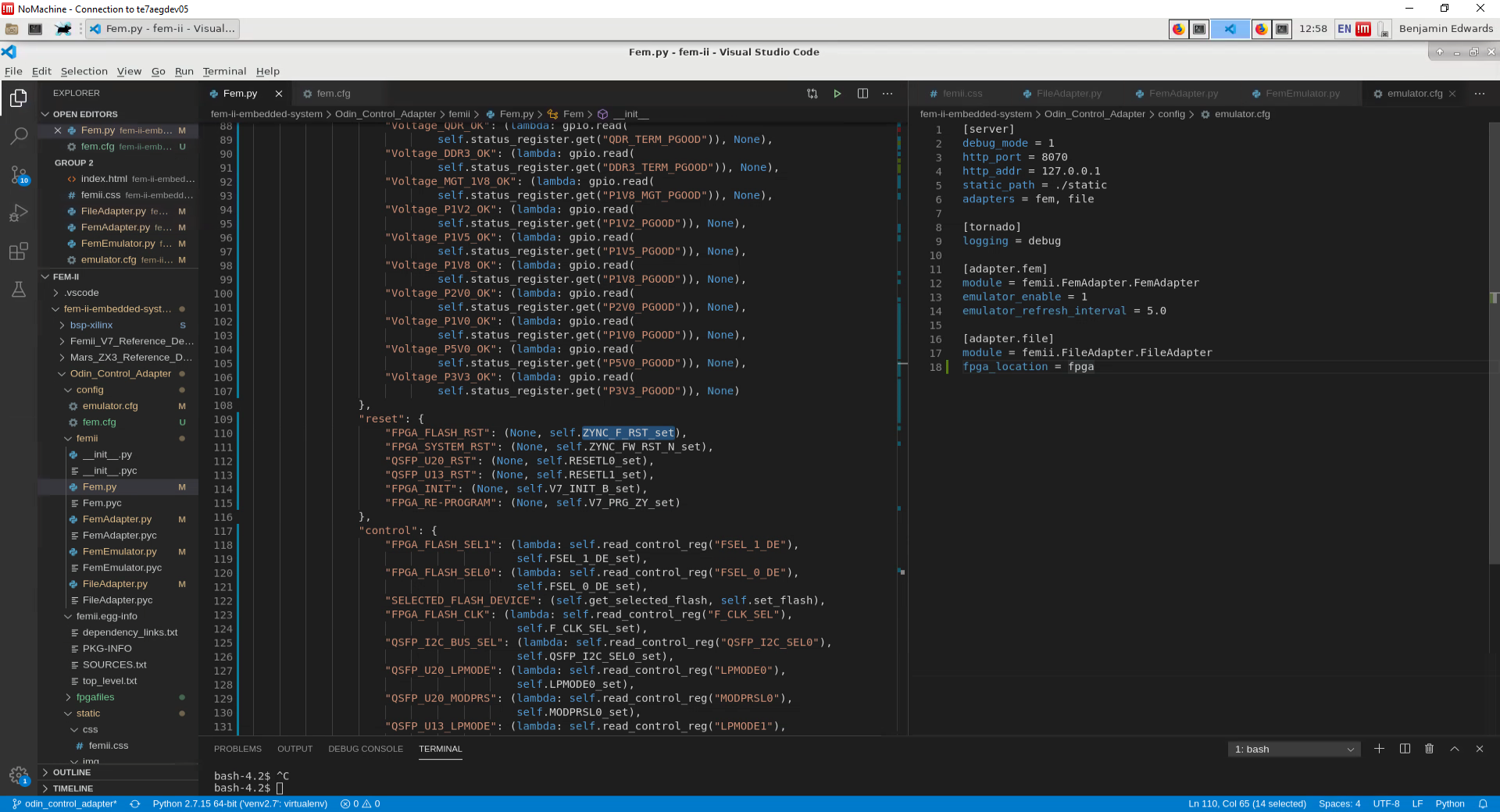
Add hardware name to list of registers with GPIO address

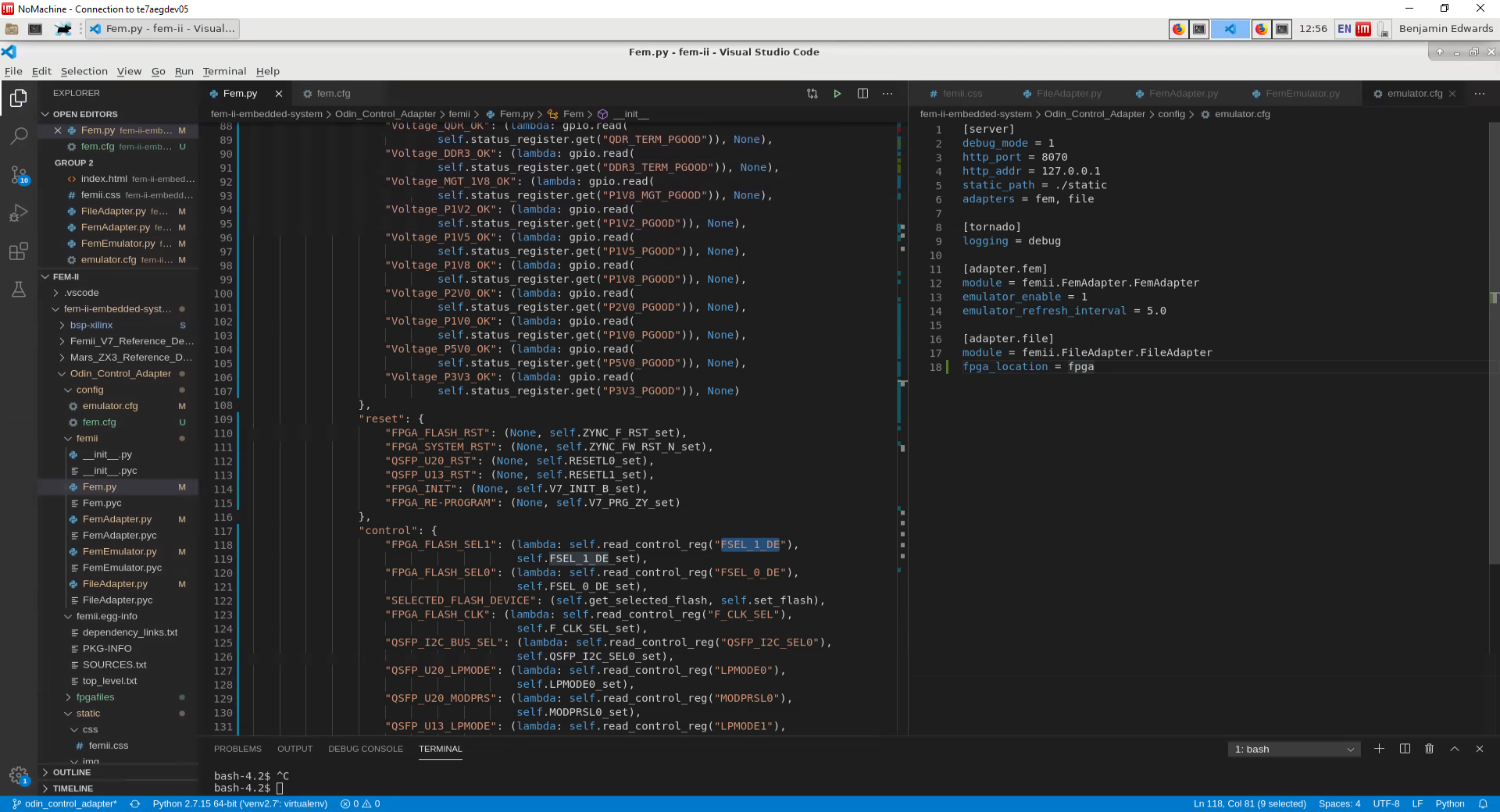
Scrolling down to the \_\_init\_\_ method of the Fem class there are list of registers for each type of control linking names to GPIO ports. Within the relevant list append a new item in the form “Hardware Name”: GPIO Port as seen below.



Next the register must be added to the parameter tree that will be used to build the GUI. For the relevant branch of the tree, copy one of the previous names then edit the name before the colon to be the desired name for the GUI, and the names after (shown highlighted in the figures below) the colon to the same hardware name that was added to the list of registers earlier.







If the new register has a set method in the parameter tree (being a writable register), then the set method will need to be created. There are groups of set methods for both control and reset registers, so navigate to the relevant grouping, copy one of the methods, then replace all instances of the name with the hardware name of the new register.

