**Module: R5: RV-fpga**

**Section:** Installations **Task:** Tools

**Task 1.3**

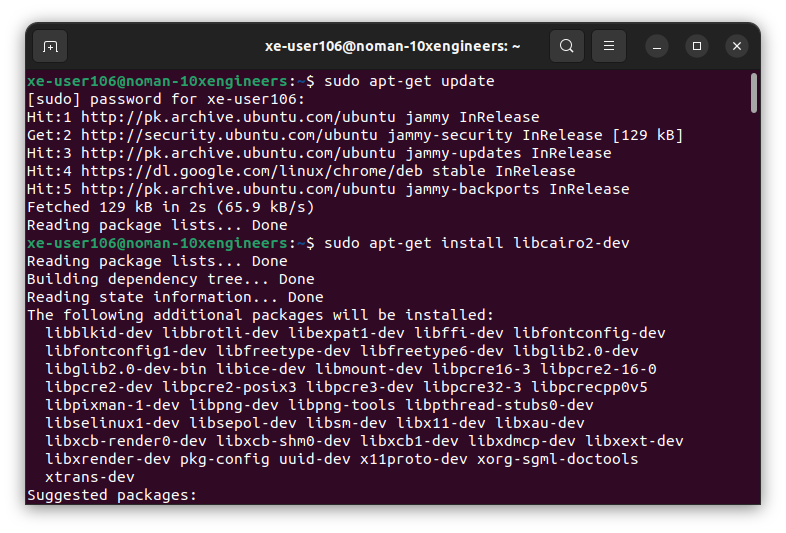
**RVfpga-Pipeline**

* **Testing:**
  + **RVfpga-Pipeline**
    1. In order to build this, we need to install the following packages:

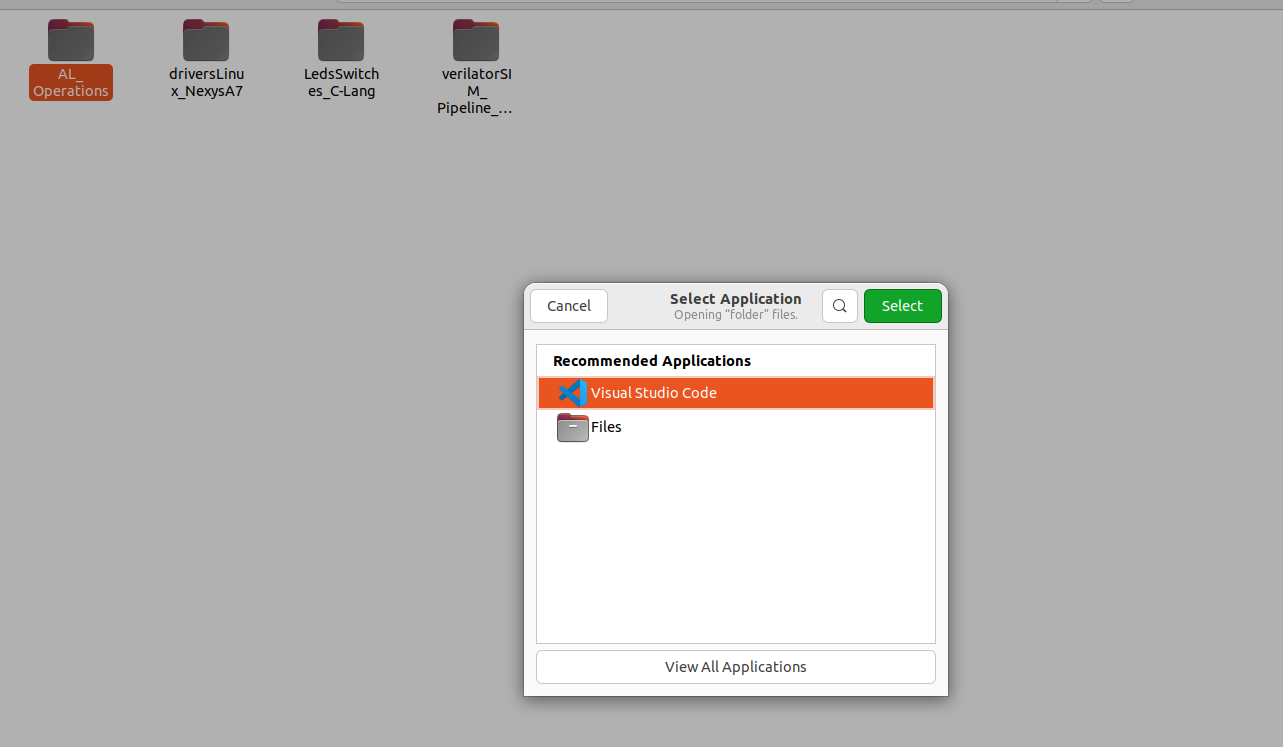
sudo apt-get update

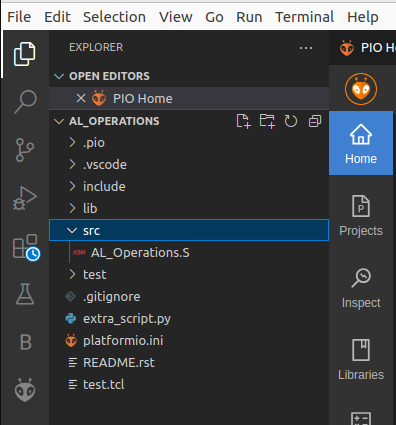
sudo apt-get install libcairo2-dev

sudo apt-get install libgtk-3-dev

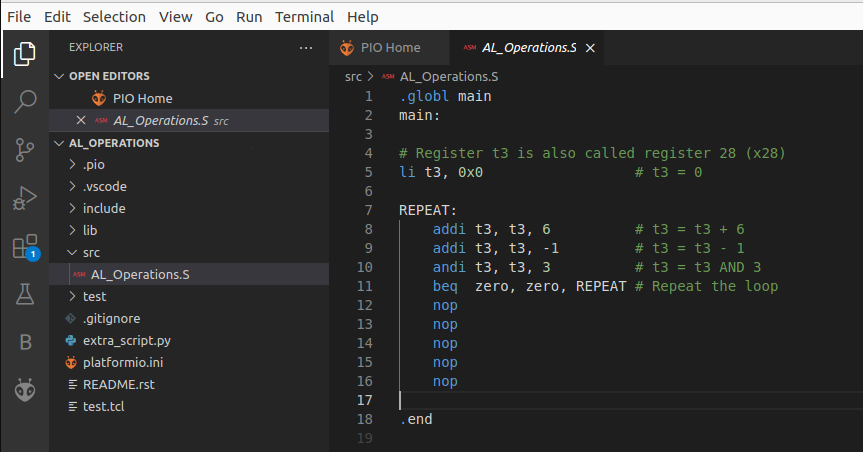


* + 1. Open the specified folder of example program in VS Code:



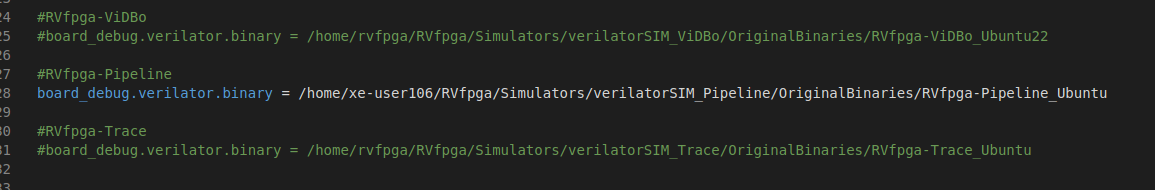


* + 1. PlatformIO will now open this program, which includes three assembly arithmetic-logic instructions (addition, subtraction, and logical and) on the same register, t3 (also called x28), within an infinite loop. We can view the program by expanding the src folder and double-clicking on **AL\_Operations.S**.

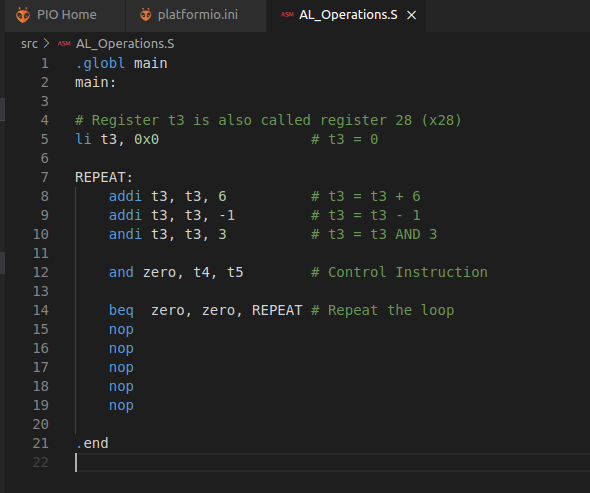


* + 1. Open file **platformio.ini**. Then, established the path to the provided RVfpga-Pipeline simulator binary by editing the following line (replaced [Path-To-RVfpga] with the appropriate path in my system:

board\_debug.verilator.binary = /home/xe-user106/RVfpga/Simulators/verilatorSIM\_Pipeline/OriginalBinaries/RVfpga-Pipeline\_Ubuntu

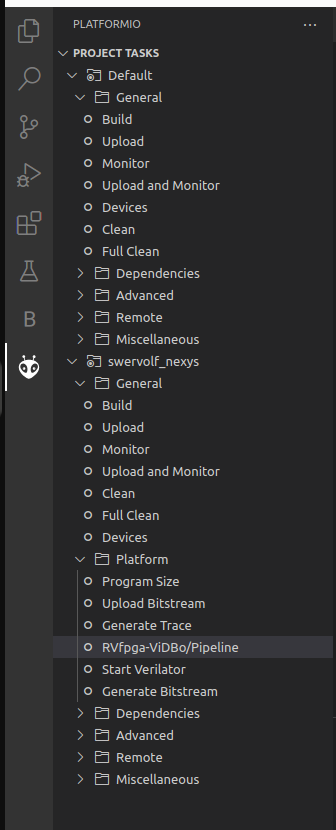


* + 1. Insert the control instruction in order to stop the execution at some point.

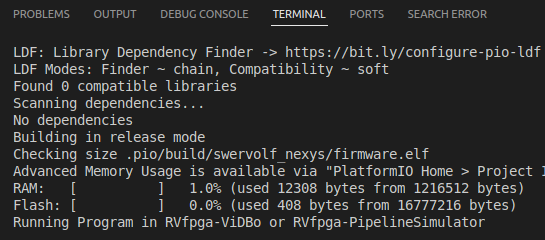


* + 1. Executed the RVfpga-Pipeline simulator from PlatformIO:
       - Click on the PlatformIO button on the left side.
       - Expand Project Tasks > env:swervolf\_nexys > Platform and clicked on **RVfpga-ViDBo/Pipeline**.
       - We have to provide execution rights to the binary by running command

chmod +x RVfpga-Pipeline\_Ubuntu



This first compiles the program and then launches the Verilator simulation of the RVfpga SoC running this program.



* + 1. A new window will open that shows the SweRV EH1 pipeline with a selection of signals for each of the 9 stages.

