[IV.1 Hardware 1](#_Toc1623847783)

[IV.1.1 I/O Mapping 1](#_Toc492389718)

[IV.1.1.1 ECP5 Chip 7](#_Toc1197038798)

[IV.1.1.2 Memory 7](#_Toc1115699982)

[IV.1.1.1 Power Design 8](#_Toc1935306684)

[IV.1.1.4 Clock Generator 9](#_Toc113242595)

[IV.1.2.1 Ordering Parts/Board 9](#_Toc22573799)

[IV.2 Software 9](#_Toc1153434220)

[IV.2.1 User Code 9](#_Toc241078774)

[IV.2.1.1 Open-Source Toolchain Configuration 9](#_Toc495074578)

[IV.2.1.2 Flash Simple Program 9](#_Toc1641859619)

[IV.2.2 Peripheral Board Design 9](#_Toc179103363)

[IV.2.2.1 Peripheral Board Hardware 9](#_Toc1161968192)

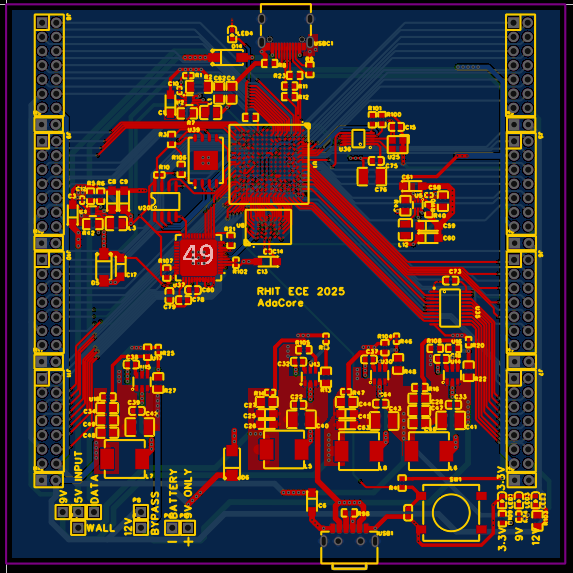
[IV.2.2.2 Peripheral Board Software 9](#_Toc1478688360)

Github Link: <https://github.com/rhit-olsonkc1/AdaCore-Protoboard>

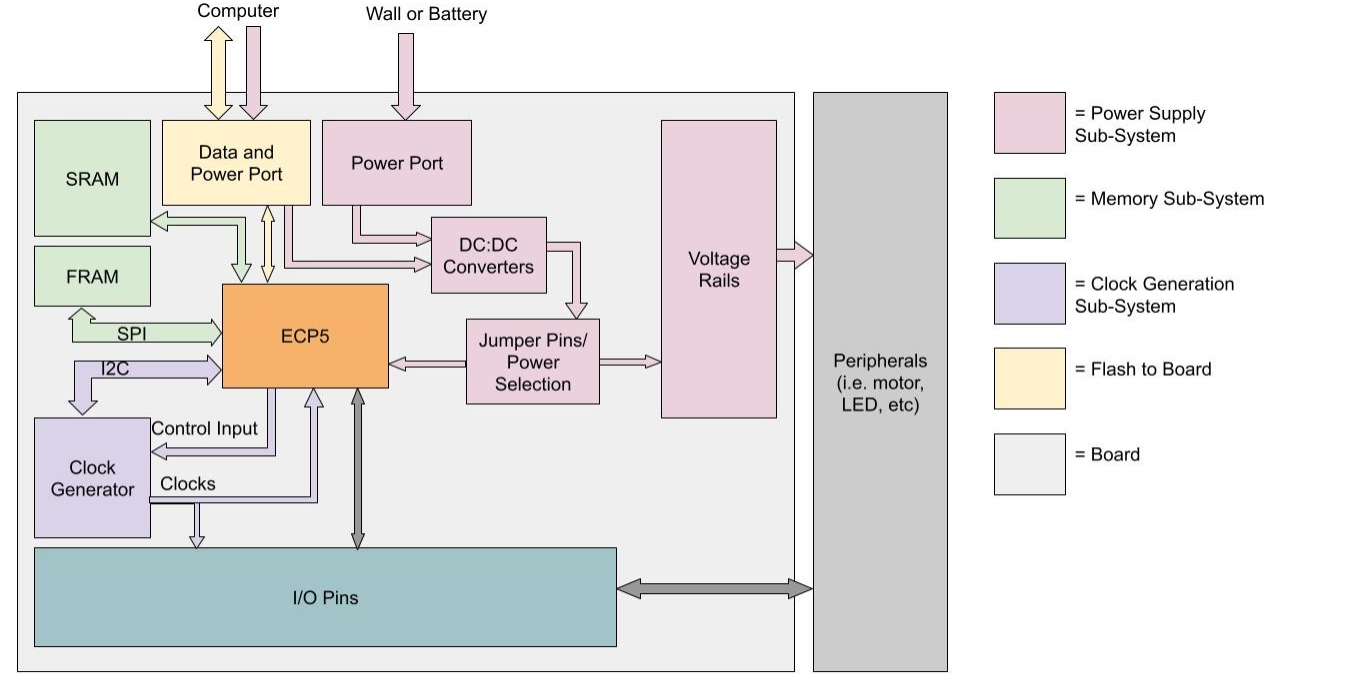
EasyEDA Project: [https://pro.easyeda.com/editor#id=b775dcf8fb034abe816e62a757abf690,tab=\*2293fd57060d48eda9ecb6699bc6adf4@b775dcf8fb034abe816e62a757abf690](https://pro.easyeda.com/editor#id=b775dcf8fb034abe816e62a757abf690,tab=*2293fd57060d48eda9ecb6699bc6adf4@b775dcf8fb034abe816e62a757abf690)

## IV.1 Hardware

The hardware section describes the elements that make up the physical board. The two subsections of this are I/O mapping and PCB.



*Figure 1: PCB*

*Figure 2: Overall hardware interaction and I/O for the board*

### IV.1.1 I/O Mapping

This section discusses the hardware components of the board and their interactions with each other. Figure 2 demonstrates the overall depiction of what the board will look like. This section can be further broken down into the following: Chip selection, memory, power design, and the clock generator

Table 1a to 1f defines the pins of the ECP5 and their purposes

|  |  |  |
| --- | --- | --- |
| ECP5 PIN | NAME | DESCRIPTION |
| R9 | fpga\_reset | Resets the fpga |
| C8 | y3 | Clock divider output |
| J1 | y2 | Clock divider output |
| K1 | y1 | Clock divider output |
| C4 | scl | I2C SCL (for clock generator configuration) |
| D5 | sda | I2C SDA (for clock generator configuration) |

*Table 1a: I2C Controlled Clock and Reset Pin Definitions*

|  |  |  |
| --- | --- | --- |
| ECP5 PIN(S) | NAME | DESCRIPTION |
| R6 | cs | FRAM chip select |
| R7 | so | FRAM serial output (MISO) |
| P7 | hold | FRAM hold |
| T7 | sck | FRAM serial clock |
| T6 | si | FRAM serial in |
| M13, M14, M15, K13, K14, J12, J13, J14, E16, E15, F16, F15, F14, E13, F13, H14, H13, G13 | a | SRAM address pins A0-A17 |
| L15, L16, K15, K16, J15, H15, J16, G15, L13, G12, K12, H12, F12, E12, G14, D13 | io | SRAM data pins I/O0 to I/O15 |
| M16 | ce | SRAM chip enable |
| L12 | oe | SRAM output enable |
| G5 | we | SRAM write enable |
| A1 | lb | SRAM lower byte |
| B2 | ub | SRAM upper byte |

*Table 1b: Memory Pin Definitions (FRAM and SRAM)*

|  |  |  |
| --- | --- | --- |
| ECP5 PIN | NAME | DESCRIPTION |
| N8 | cs | QSPI chip select |
| T7 | miso | QSPI master in slave out |
| M7 | d2 | QSPI data line 2 |
| N7 | d3 | QSPI data line 3 |
| N9 | sck | QSPI serial clock |
| T8 | mosi | QSPI master out slave in |

*Table 1c: QSPI Flash Pin Definitions*

|  |  |  |
| --- | --- | --- |
| ECP5 PIN | NAME | DESCRIPTION |
| B1 | d\_p | USB data positive |
| B2 | d\_n | USB data negative |
| C2 | pullup | USB pullup |

*Table 1d: USB Pin Definitions*

|  |  |  |
| --- | --- | --- |
| ECP5 PIN | NAME | DESCRIPTION |
| M10 | tdo | QSPI test data out |
| T10 | tck | QSPI test clock |
| R11 | tdi | QSPI test data in |
| T11 | tms | QSPI test mode select |
| B2 | d\_n | QSPI data positive |
| B1 | d\_p | QSPI data negative |

*Table 1e: FTDI Pin Definitions*

|  |  |  |
| --- | --- | --- |
| ECP5 PIN | NAME | DESCRIPTION |
| A13, B14, A14, B15, C15, B16, C16, D16 | tolerant\_io | 5V tolerant I/O pins |
| C13 | oe | 5V tolerant I/O output enable |
| E3 C1 E2 D1 F2 E1 G2 F1 H2 G1 F3 D3 C3 B3 A3 B4 A4 B5 A5 B6 A6 | gpio\_1 | General purpose pins |
| P1 R1 T2 T3 K2 L2 N1 P2 M3 P3 N3 B7 D6 C5 D7 C6 C7 B8 C9 D9 B9 C10 D11 | gpio\_2 | General purpose pins |
| R4 T4 P4 N5 M5 N11 R12 P12 R13 T13 P13 A8 B10 A9 C11 A10 B12 A11 B11 A12 B13 C12 | gpio\_3 | General purpose pins |
| R15 P14 N12 T14 R14 T15 R16 P15 P16 N14 N16 C14 D14 E14 A13\_i B14\_i A14\_i B15\_i C15\_i B16\_i C16\_i D16\_i | gpio\_4 | General purpose pins |

*Table 1f: GPIO Pin Definitions*

#### *IV.1.1.1 ECP5 Chip*

This board uses the ECP5, more specifically the LFE5U-25F-6BG256C.

Key Features of the LFE5U-25F-6BG256C:

* 24 Look-Up Tables(LUTs)
* 1,008 Kb of Embedded Memory
* 194 Kb of Distributed RAM

More information can be found in the ECP5 Datasheet: <https://www.lcsc.com/datasheet/lcsc_datasheet_2403111456_Lattice-LFE5U-25F-6BG256C_C1521614.pdf>

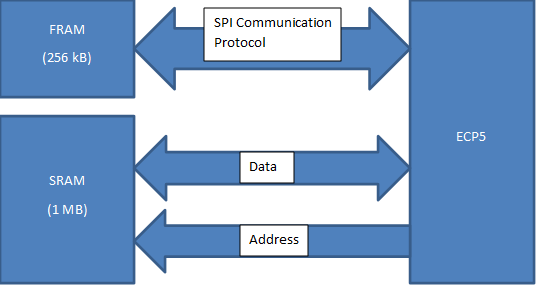
#### *IV.1.1.2 Memory*

The board’s memory consists of SRAM and FRAM.

|  |  |
| --- | --- |
| **Memory Type** | **Quantity** |
| SRAM | 1Mbits |
| FRAM | 256kbits |

Table 2: Memory Types and Quantities

As seen in Figure 3, the ECP5 communicates with FRAM using a SPI communication protocol. The ECP5 communicates with SRAM through address pins and I/O pins. The address pins carry the memory address, and the I/O pins carry data to and from memory

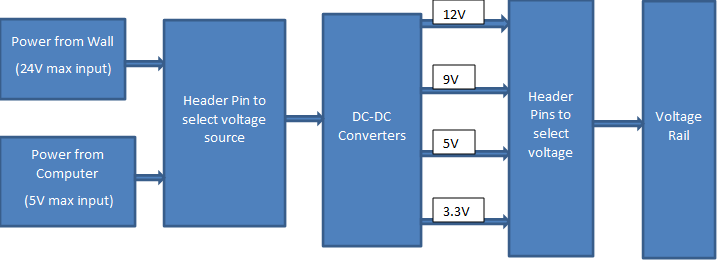


*Figure 3: Memory Interactions Diagram*

#### *IV.1.1.1 Power Design*

As seen in Figure 4, power can be supplied by either a computer, battery, or wall. However, power from the computer can only supply a voltage to the power rail that is less than 5V.

This design requires the user to heavily interact with the board to get the desired voltage, but it contributes to the low cost and modularity of the design.



*Figure 4: Power Design*

*IV.1.1.3.2 Power Configuration*

This board has the capability to produce voltage rails at 12V, 9V, 5V, and 3.3V. It can get its input power from the wall, a battery, or directly from the computer. Voltage amounts can only be produced by a source providing more than the desired output, and the header pins must be configured by the user.

Figure 5, shows where the header pins are on the board, and Table 2 demonstrates the proper configuration of the header pins for each voltage source, and

*Figure 5: Header Pin Location*

*Figure 6: Header Pins “Zoomed-In”*

|  |  |  |  |
| --- | --- | --- | --- |
| **Source** | **Output Voltage on Rails** | **Header Configuration** | **Additional Comments** |
| Wall | 12V\*, 9V, 5V, 3.3V |  | Due to the specifications of the DC-DC converters, using the wall input without the bypass will result in a voltage slightly less than 12V |
| 12V, 9V, 5V, 3.3V |  |  |
| Battery | 9V, 5V, 3.3V |  |  |
| Data | 5V, 3.3V |  |  |

*Table 3: Header Pin Configuration*

*Figure 7: 1 of 8 I/O Sections*

*Figure 8: 1 I/O Section “Zoomed-In”*

#### *IV.1.1.4 Clock Generator*

The component used to generate the clocks is the MS5351M. This component was chosen based on price, availability, and frequency range.

Figure 9: Clock Generation Diagram

The output of an onboard 27MHz Crystal Oscillator is fed into the MS5351M, which contains PLLs and dividers that can be configured with I2C and control signals. The generator outputs 3 clocks that can range from 2.5kHz to 200MHz

More information about the MS5351M can be found in its datasheet:

<https://qrp-labs.com/images/synth/ms5351m.pdf>

IV.1.2 PCB

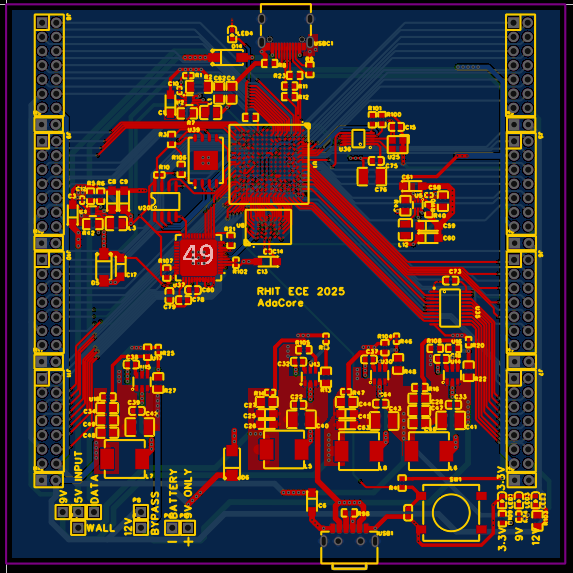


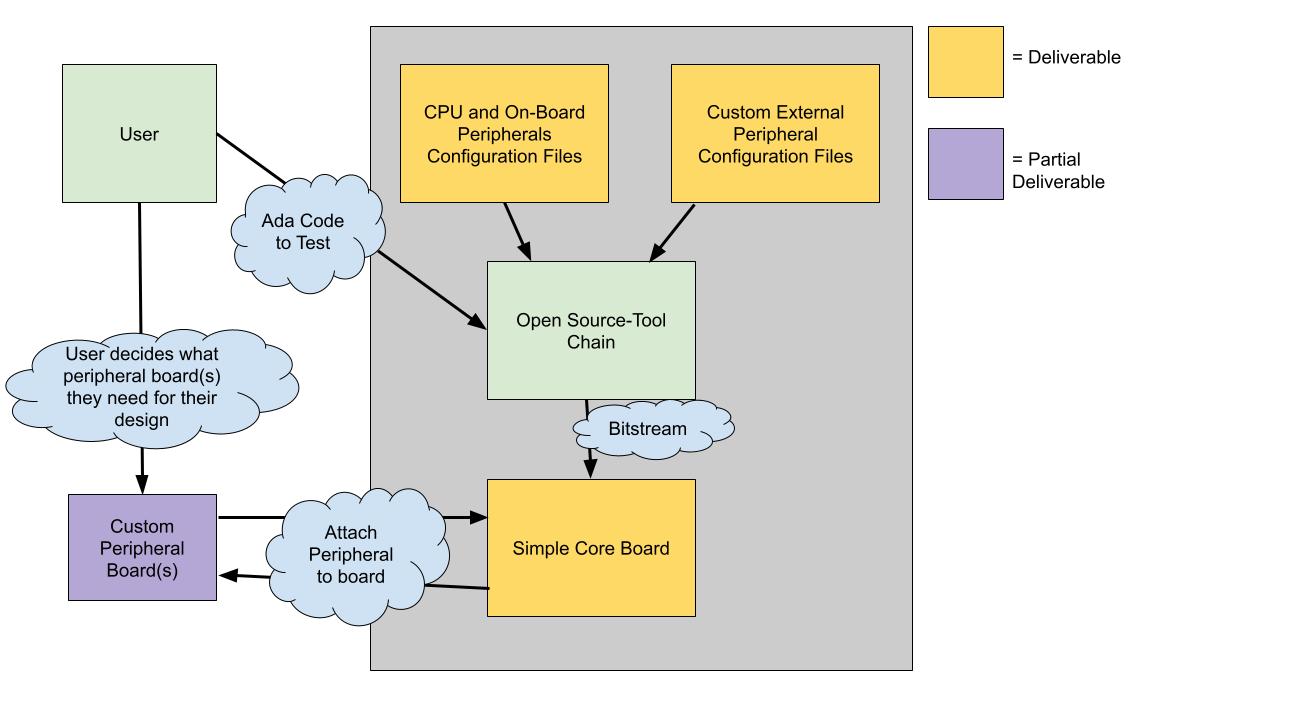
Figure 10: Top View of the PCB.

#### *IV.1.2.1 Ordering Parts/Board*

All parts used in the board design are available from LCSC (<https://www.lcsc.com/>) and ordered through EasyEDA. Doing so means that the board will be soldered by the manufacturers, which minimizes any extra steps for the user.

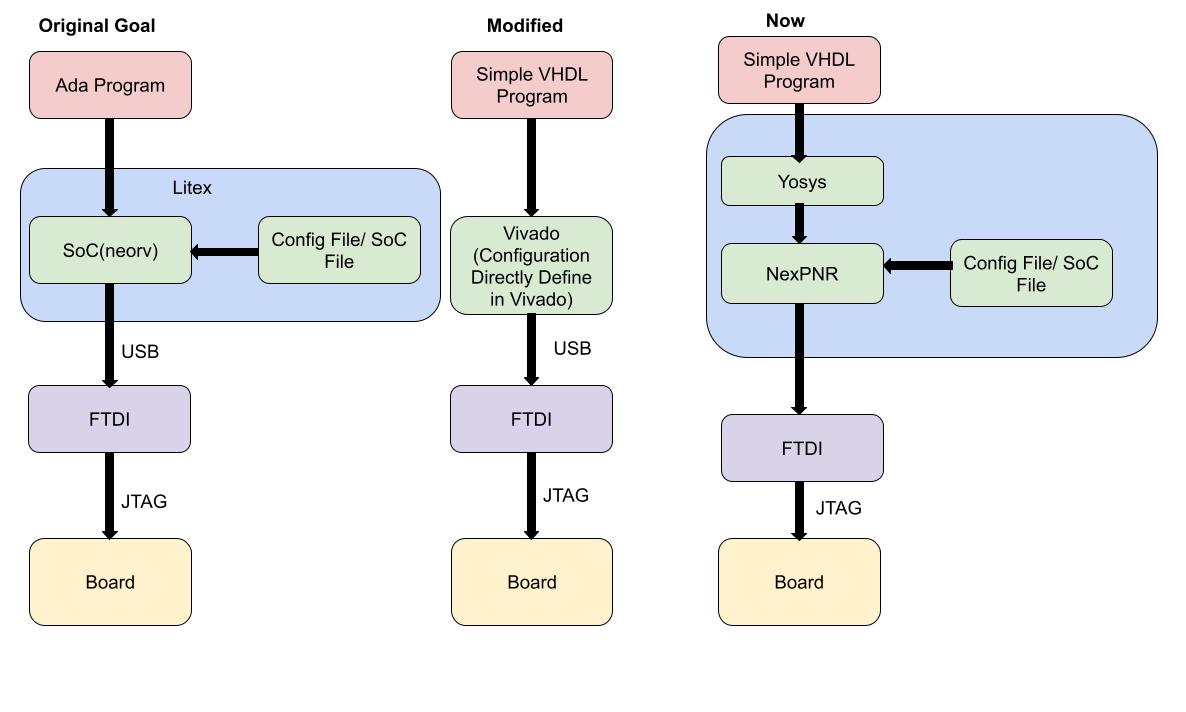
## IV.2 Software

The software describes how the users will be able to program and interact with the board. The two subcategories of this section are the user code and peripheral board design



*Figure 11: Software Interactions (End Goal)*

### IV.2.1 User Code

Figure 12, shows the open-source tool chain used for the project

*Figure 12: Open-Source Toolchain*

#### *IV.2.1.1 Open-Source Toolchain Configuration*

Currently, we are still working on integrating the open-source tool chain with the hardware, but we have created configuration files based on the Litex Orange\_Crab files to have a starting point. These files are unimplemented/tested.

The SoC and platform file can be found in the “configuration-files/” directory. They are titled “our\_board\_platform\_file.py” and “our\_board\_SoC.py”. In addition, the Orange crab files can be found in the same directory.

#### *IV.2.1.2 Flash Simple Program*

In the “Blink-LED-Example" folder are the files to flash a simple program that blinks the onboard LED.

### IV.2.2 Peripheral Board Design

Peripheral Board EasyEDA Project(Has the base schematic for peripheral boards): [https://pro.easyeda.com/editor#id=272f9f23de424d80810937cb97ee5a7e,tab=\*1d1858f54d4143b0bfea385ff95b55e8@272f9f23de424d80810937cb97ee5a7e](https://pro.easyeda.com/editor#id=272f9f23de424d80810937cb97ee5a7e,tab=*1d1858f54d4143b0bfea385ff95b55e8@272f9f23de424d80810937cb97ee5a7e)

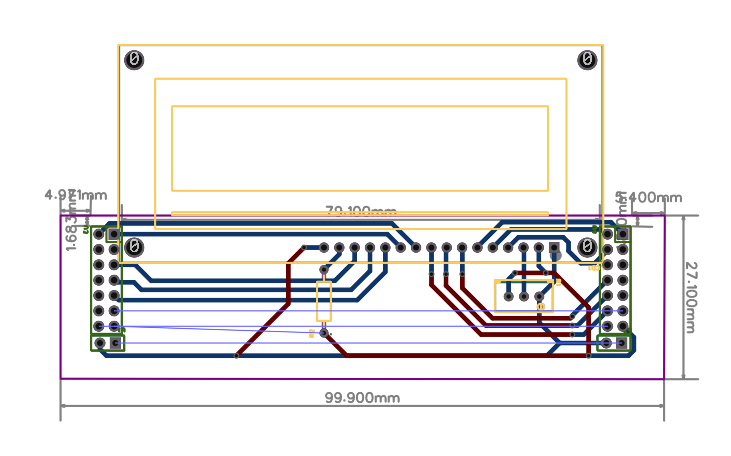
LCD Extension Board Example:

[https://pro.easyeda.com/editor#id=a81874e131d34a9fa7ddbfd5b8a12d08,tab=\*bc98c65ea31043a2bb4d7faf4bdc5280@a81874e131d34a9fa7ddbfd5b8a12d08](https://pro.easyeda.com/editor#id=a81874e131d34a9fa7ddbfd5b8a12d08,tab=*bc98c65ea31043a2bb4d7faf4bdc5280@a81874e131d34a9fa7ddbfd5b8a12d08)

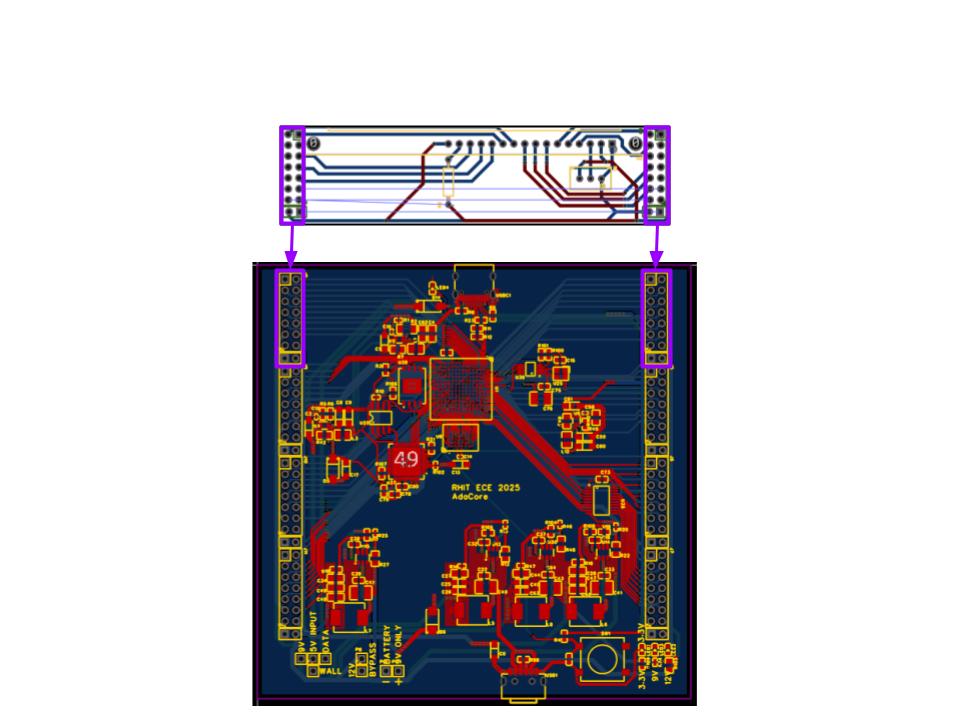
Schematic can be found in the “Board/Peripheral-Board/” directory

#### *IV.2.2.1 Peripheral Board Hardware*

The goal of the project is that peripherals can easily be “slotted” onto the board. Figure x shows an example of an LCD peripheral board



*Figure 13: LCD Extension Board*



*Figure 14: “Slotting” the Peripheral onto the Board*

#### *IV.2.2.2 Peripheral Board Software*

For each additional peripheral designed to slot on the board, we plan to create a driver. The code for a driver for the previous LCD module can be found in the “peripherals/ directory”. It is currently untested