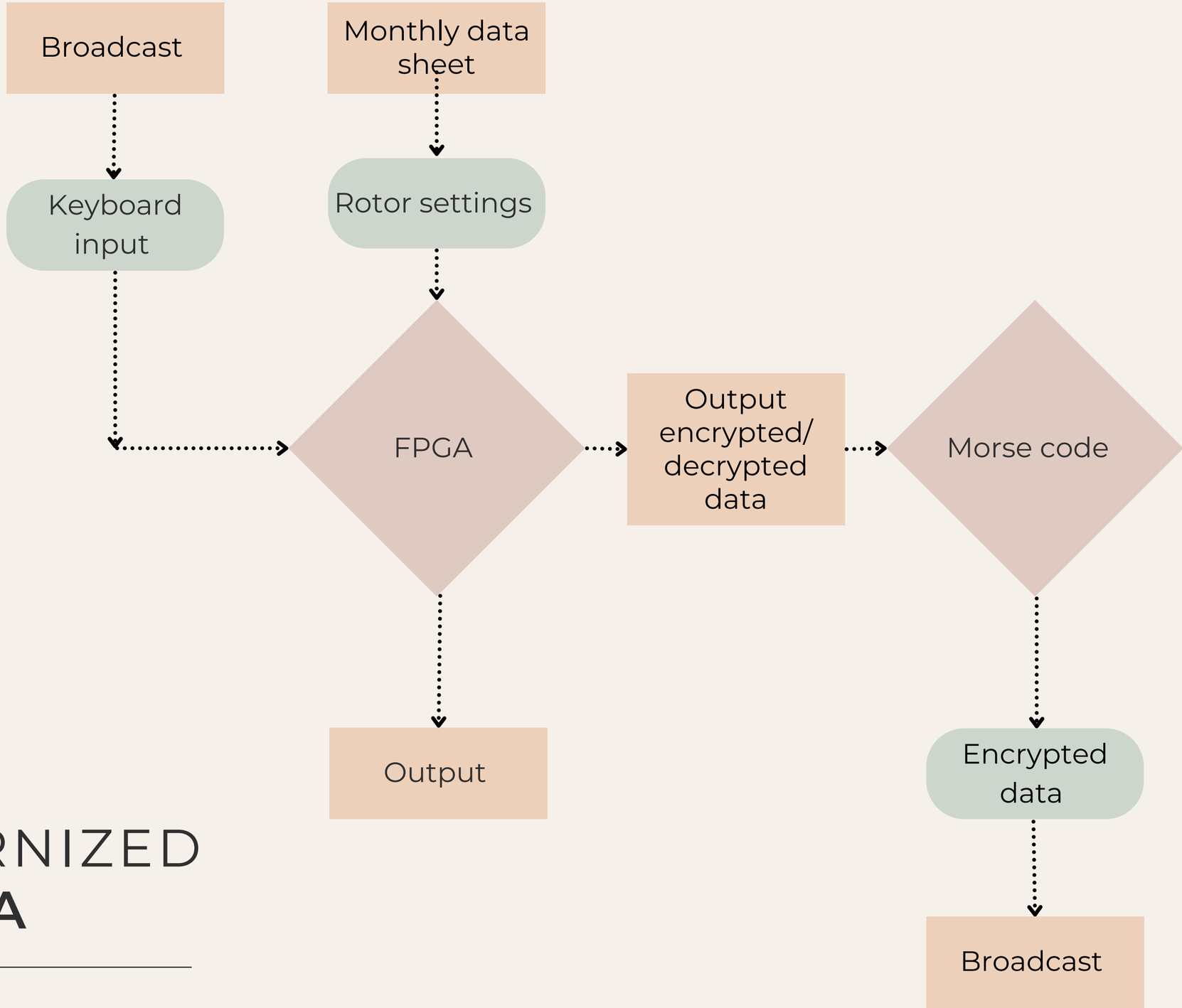
Modernized Enigma

Contents

What is an Enigma

The Enigma machine was a cipher device used primarily by Nazi Germany during World War II (1939-1945) to encode and decode secret messages.

The Enigma machine used rotors to scramble letters, making messages nearly impossible to decode without the correct settings. Each keystroke changed the rotor positions, creating unique encryption patterns that helped secure wartime communication.



MODERNIZED **ENIGMA**

Objective

This project aims to retain the core cryptographic principles of the original Enigma while enhancing its functionality, user interface, and visual feedback, thereby creating a more interactive and accessible version of the machine.

To modernize the Enigma machine by designing and implementing a version using FPGA technology

This modernized Enigma will provide a hands-on understanding of historical encryption techniques within a contemporary technological framework

Our Approach

FPGA

SEGMENT DISPALYS

LED

KEYBOARD

BREAD BOARD

WIRES

DIODES

MUX

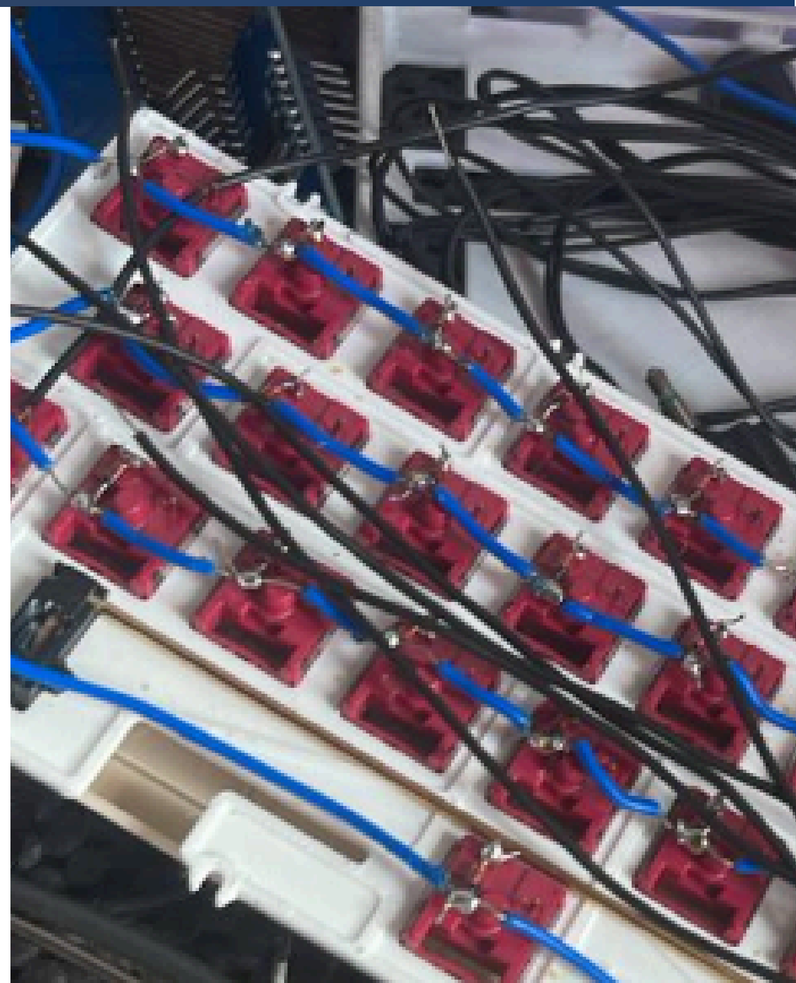
BCD DRIVE

KEYBOARD

THE INPUT SIDE

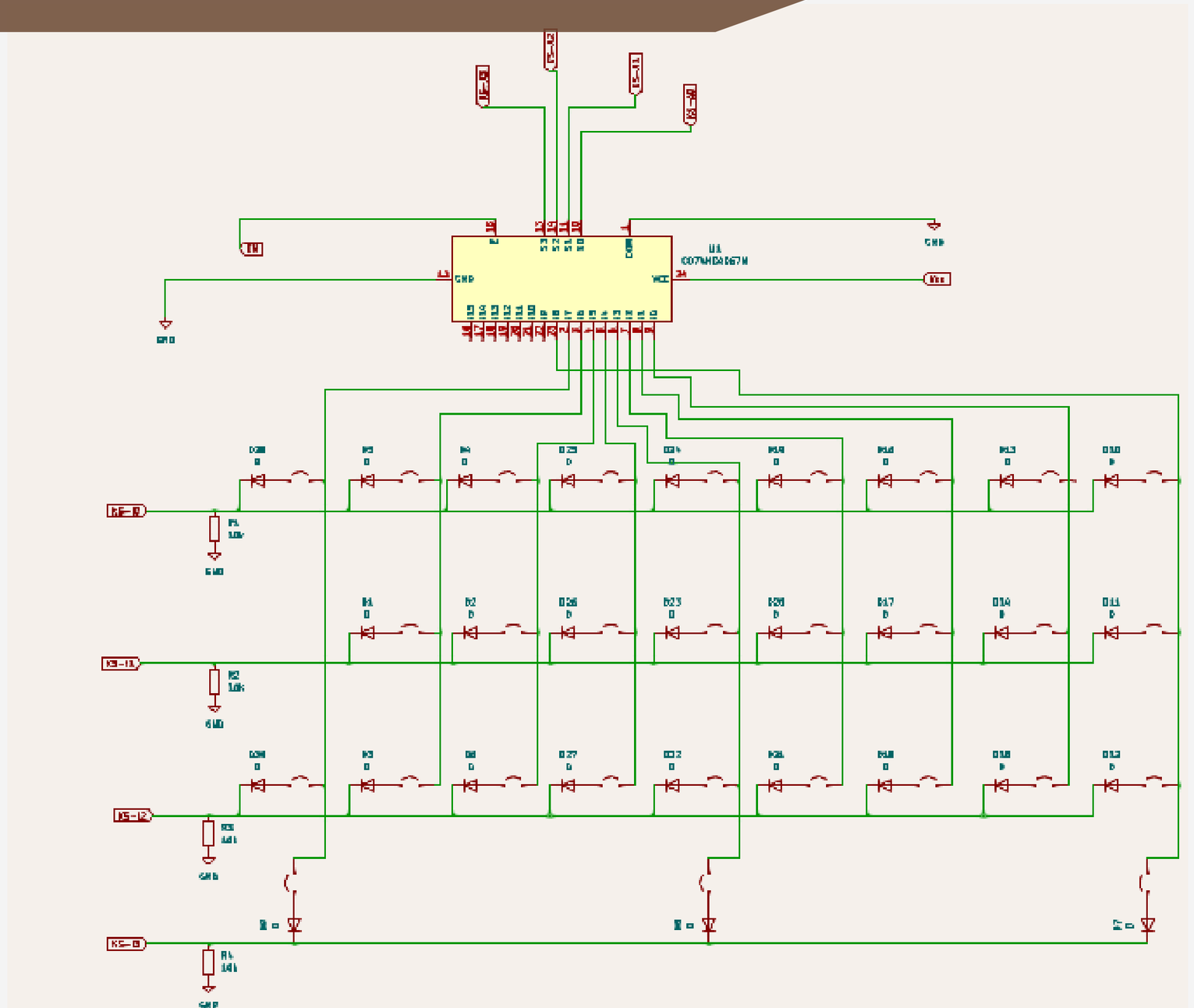
The keyboard made is of matrix-style setup (9x4), where keys are arranged in rows and columns with diodes in series for each key. This layout allows each key press to connect a specific row to a specific column, creating a unique signal path for each key.

A multiplexer (74HC4051) is used to read the inputs from the keyboard matrix while saving pins on the microcontroller. The multiplexer scans each row (or column) sequentially, allowing it to detect which key in that row or column is pressed by checking for signals across the columns (or rows).



This setup is analogous to the input side of a modernized Enigma machine, where pressing a key would create a unique electrical path that maps to a specific output (or letter). The multiplexer and matrix arrangement streamline the process, making it easy to detect and process key presses in real time, similar to how an Enigma machine would take each input sequentially to create coded output.

SCHEMATIC OF KEYBOARD

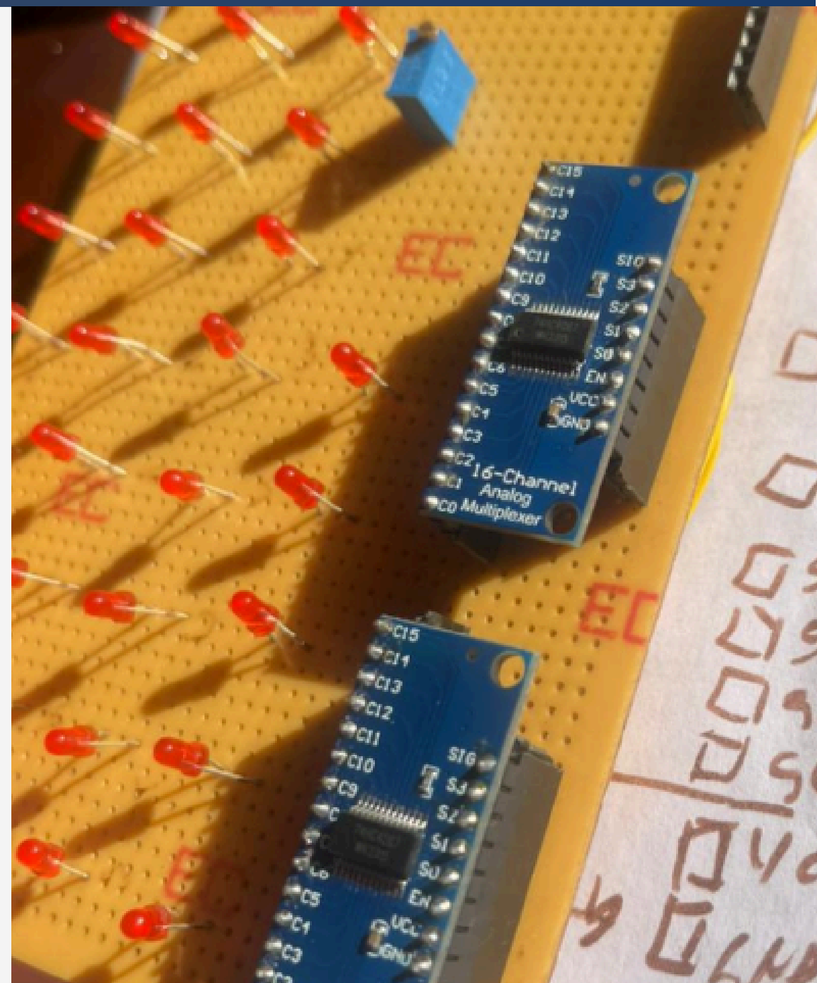


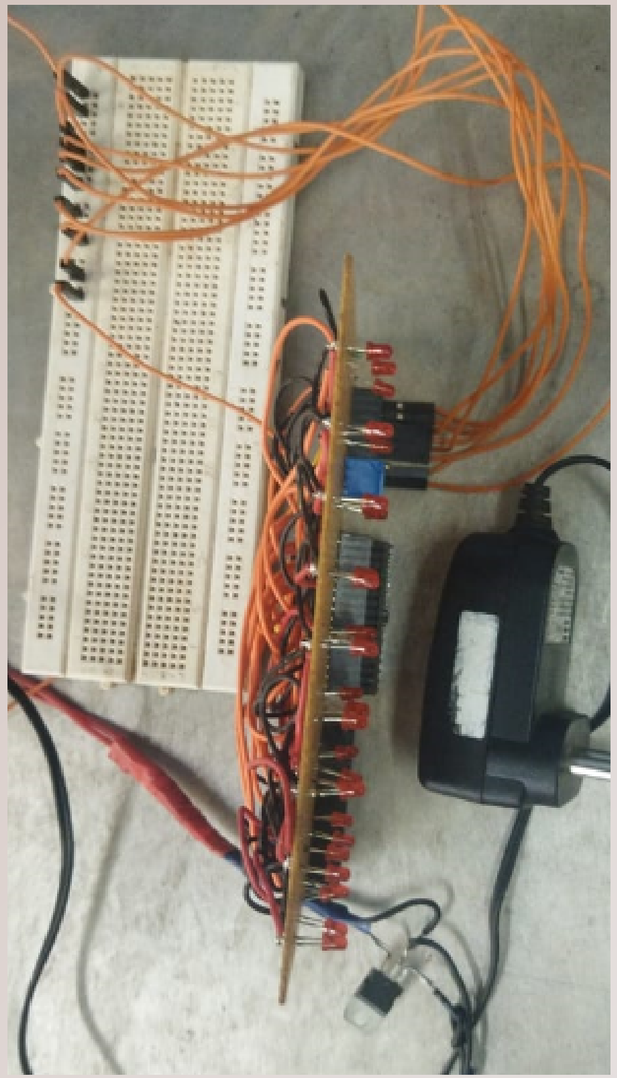
LIGHT UP BOARD

THE OUTPUT SIDE

A custom output board featuring an array of LEDs controlled by two 16-channel analog multiplexers. This type of board could serve as an effective display system for an Enigma machine replica. By using multiplexers, multiple LEDs can be controlled with fewer input lines, making it easier to manage a large number of indicators without complex wiring.

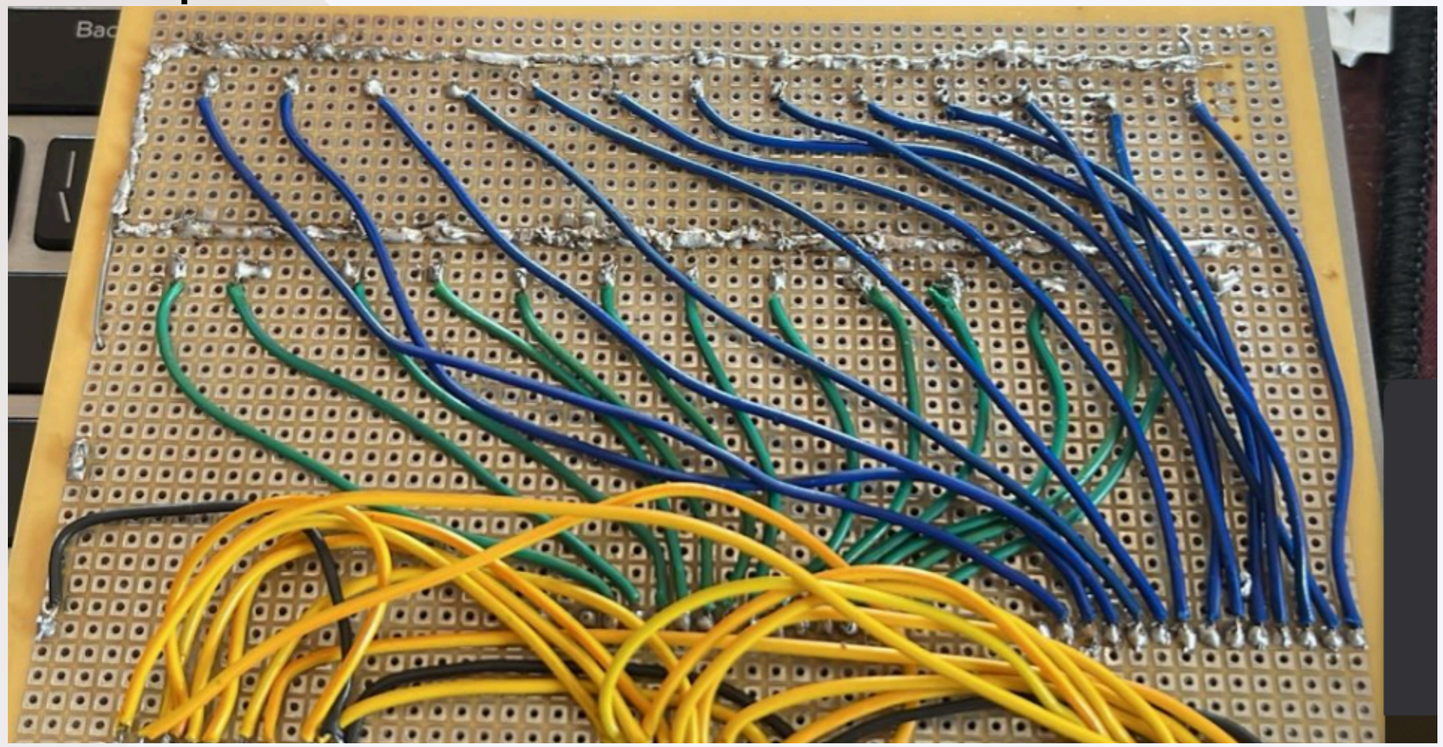
Each LED can represent a letter, lighting up when a corresponding letter is decoded, similar to the lampboard in historical Enigma machines



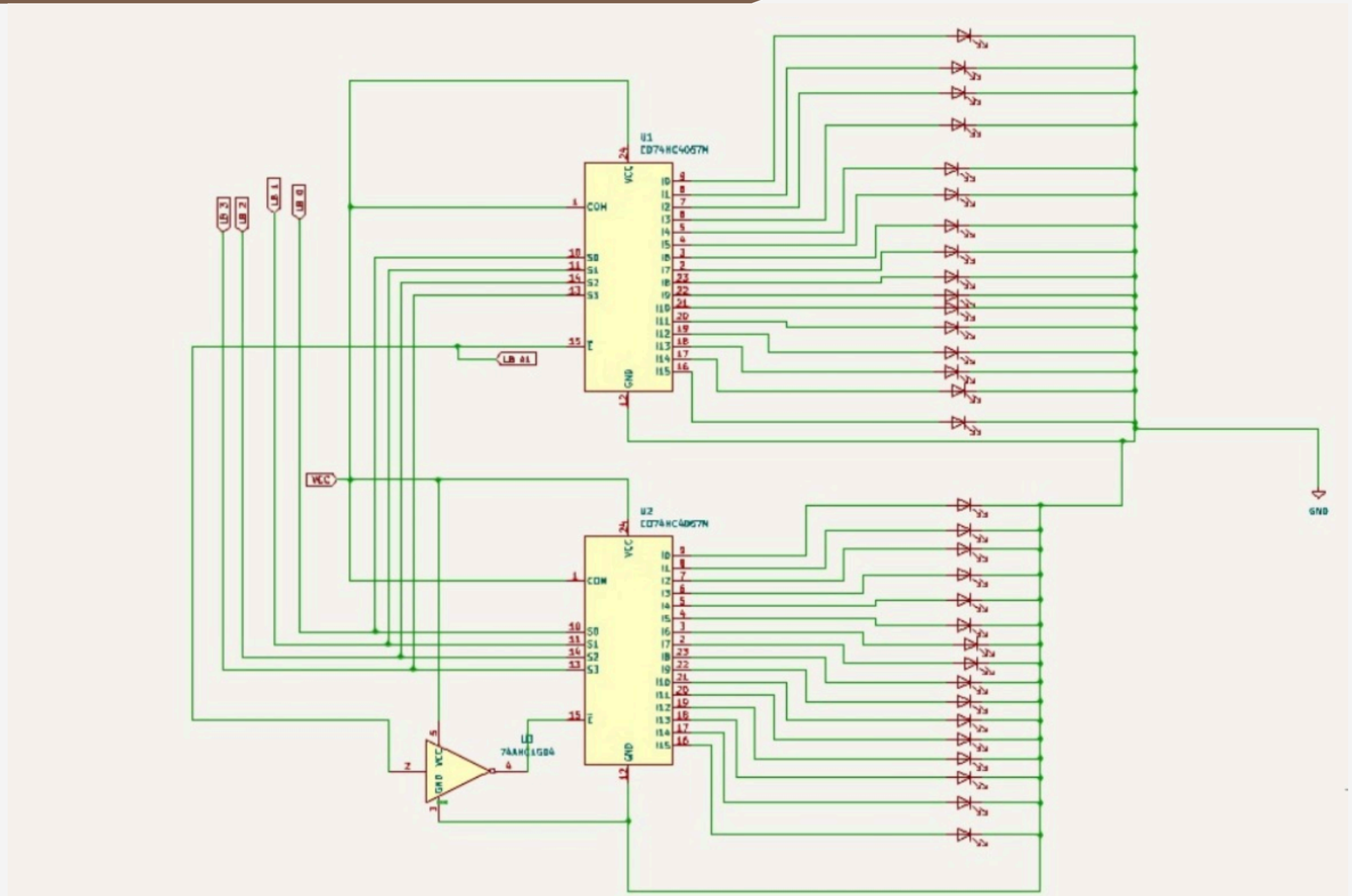


In this setup, the multiplexers allow selection of individual LEDs based on input signals, enabling specific LEDs to light up for specific letters. This setup mimics the output function of an Enigma machine, where each keystroke activates a unique lamp corresponding to the encoded letter.

When a character is decoded, the microcontroller sends a signal through the multiplexer to select the LED on the lightup board that should illuminate, providing a visual representation of the machine's output.



SCHEMATIC OF LIGHT UP BOARD

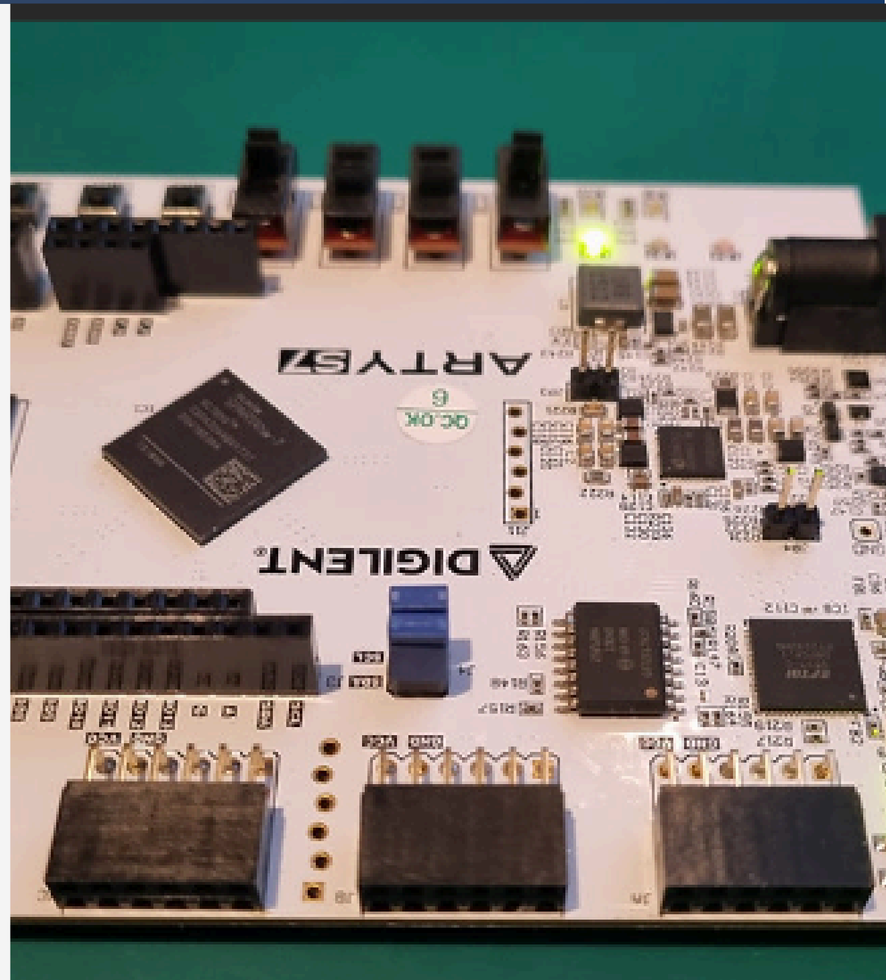


SCHEMATIC

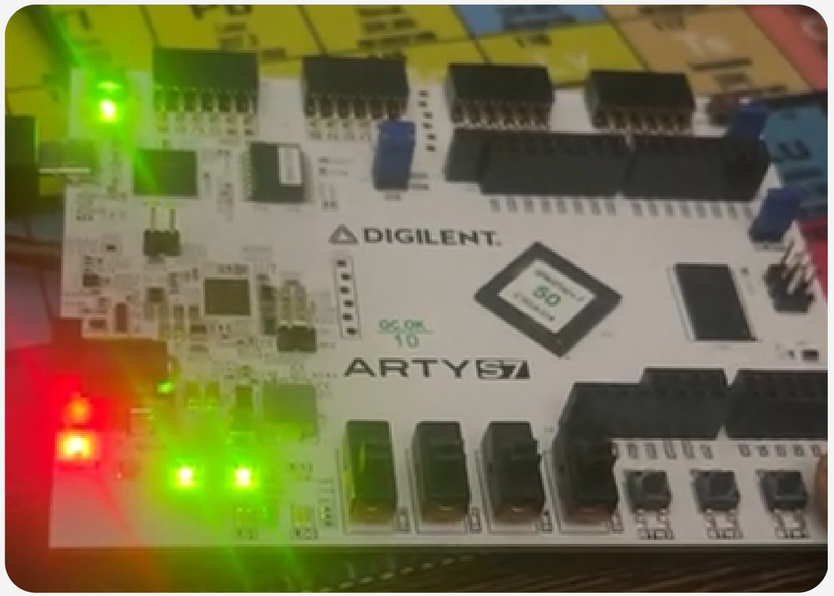
For designing the schematics, we used KiCad due to its comprehensive library of electronic components and intuitive schematic editor. The software allowed us to organize and interconnect components efficiently while adhering to design rules. With KiCad's simulation support, we could validate circuit functionality before moving to layout. The tool's user-friendly interface made it easy to manage complex designs.

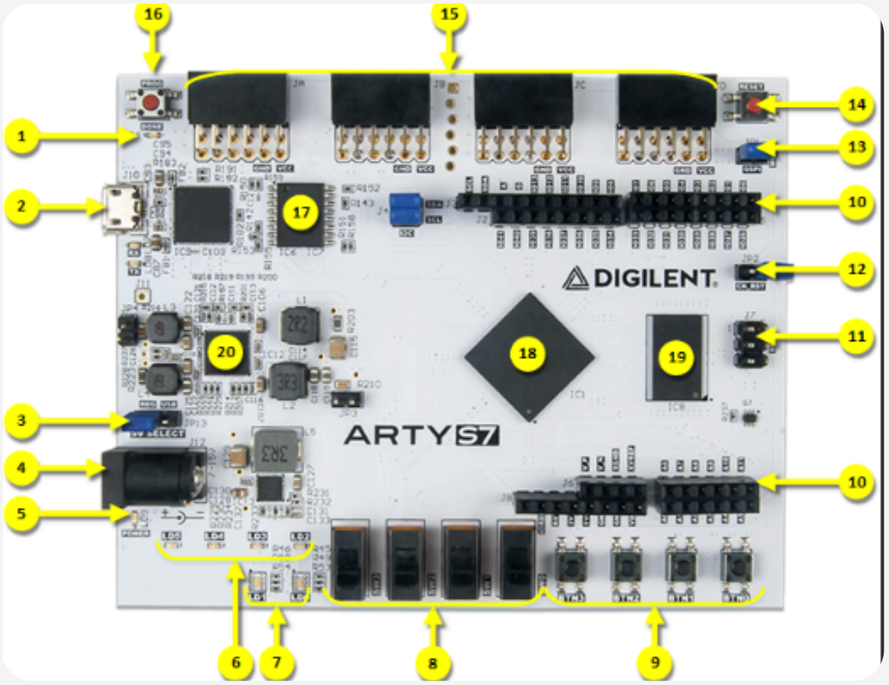
FPGA - Field programmable gate array

The Arty S7 FPGA will act as the main controller that processes encryption and decryption based on the Enigma logic. Each key press from the self-built keyboard is fed to the FPGA, which handles all the necessary transformations through a series of virtual rotors and a reflector implemented as lookup tables. After the FPGA processes each character, it sends the output signal to a multiplexer (MUX) to direct it to the appropriate output.



The multiplexer routes signals based on the current rotor position. After each character is encrypted, the MUX guides the output to the light-up board, where LEDs represent the encrypted characters. The light-up board serves as a visual display, lighting up each letter according to the FPGA's processed output, simulating the behavior of an authentic Enigma machine’s lampboard.



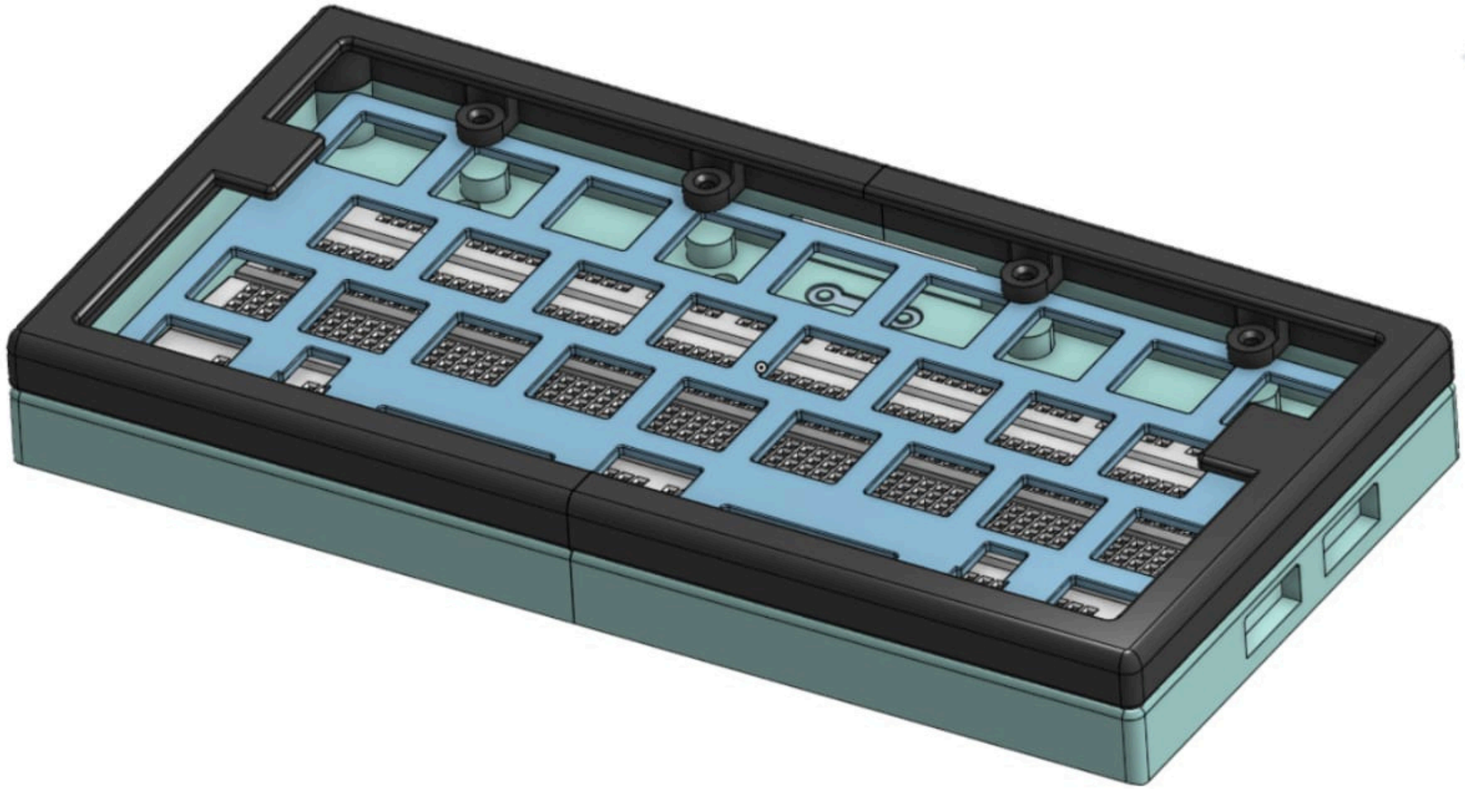


The FPGA’s role as the main controller is essential, as it manages the logic and state for each key press, advancing the rotor positions dynamically. By using the FPGA’s logic gates and memory resources, this setup efficiently replicates the complex, real-time computations of the Enigma machine, providing seamless encryption with each keystroke.

CAD for keyboard

Keyboard based on enigma layout for this project. Has space for a full-size breadboard, and can be configured in 2 mounting styles(with additional parts, currently is top-mount). Large parts have been split into two and can be glued together Components:

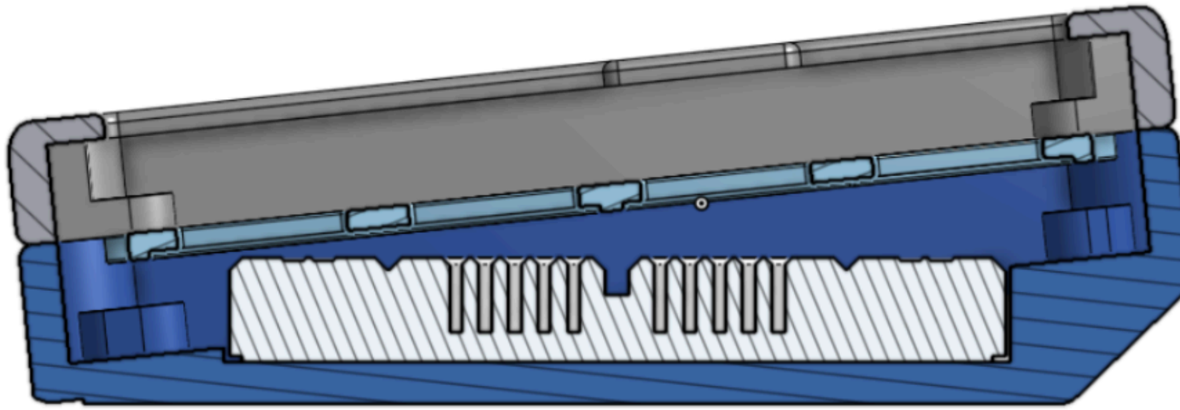
- 3dp plate
- Left top
- Left bottom
- Right top
- Right bottom



Screws needed(with nuts):

size	length	Qty
M2	5mm	8
M2.5	15mm	4

SIDE VIEW:



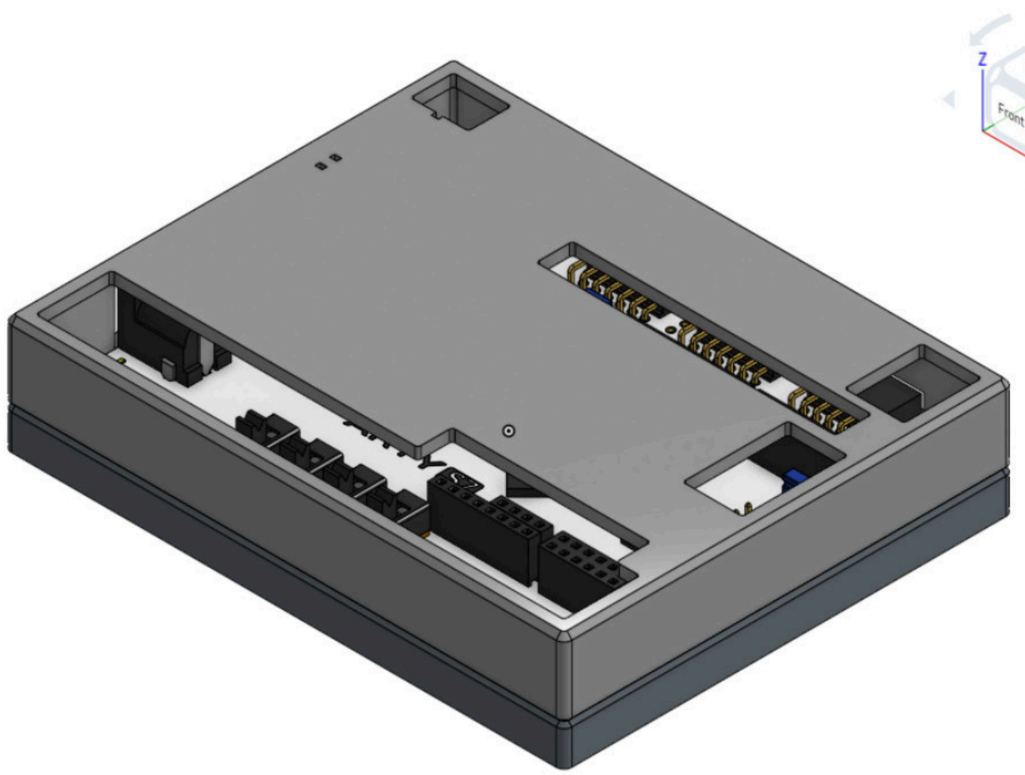
views are also present in the respective folder:

https://cad.onshape.com/documents/61e4573137cff3cd69e74c09/v/754eaec4891ee67ad769792b/e/6e0d1207b123d6a84890a30c?renderMode=0&uiState=6728f3efcf02b05b8225c897

CAD FOR FPGA-ARTY-S7 CASE

Arty S7 Case
Case for the Arty S7 Spartan FPGA board
Components:

- Case top
- Case bottom



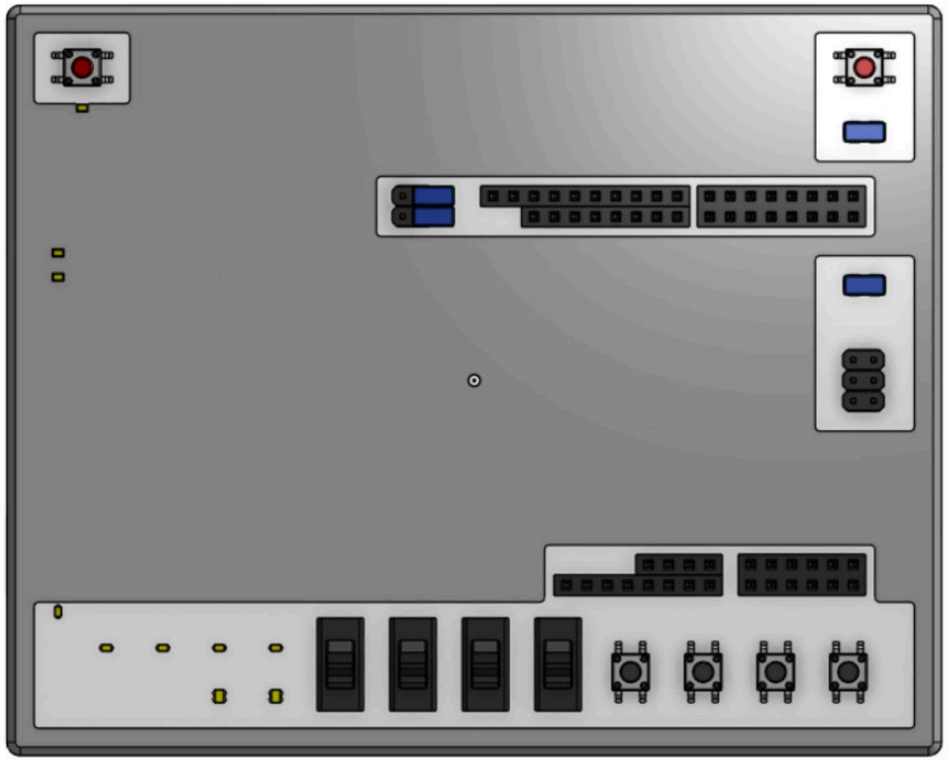
Screws needed:

size	length	Qty
M2	10mm	4

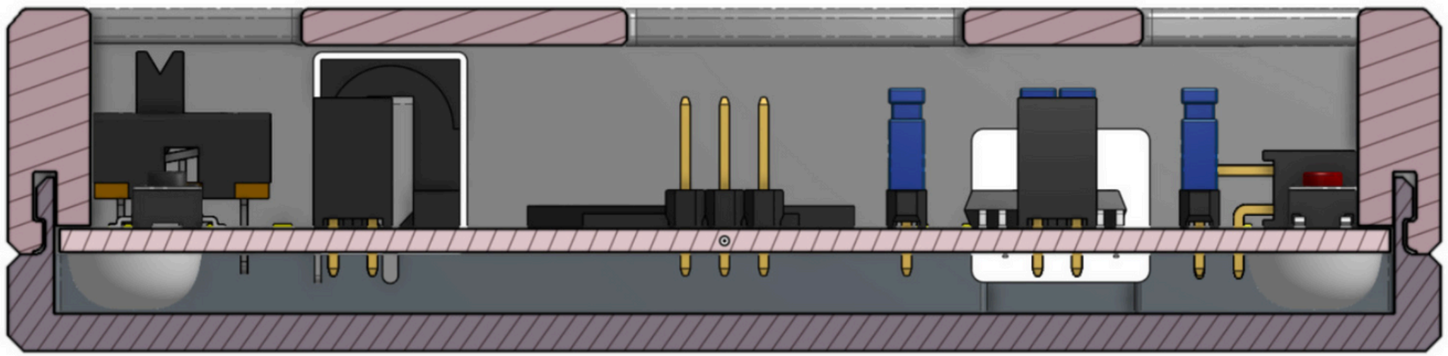
views are also present in the respective folder:

https://cad.onshape.com/documents/9edc83e212fe6055c19b0e6f/v/f7f69ae74b499a4b4b45839b/e/570bb53666f2b7a60168de72?renderMode=0&uiState=6728efd880a4715c97a79fe2

TOP VIEW:

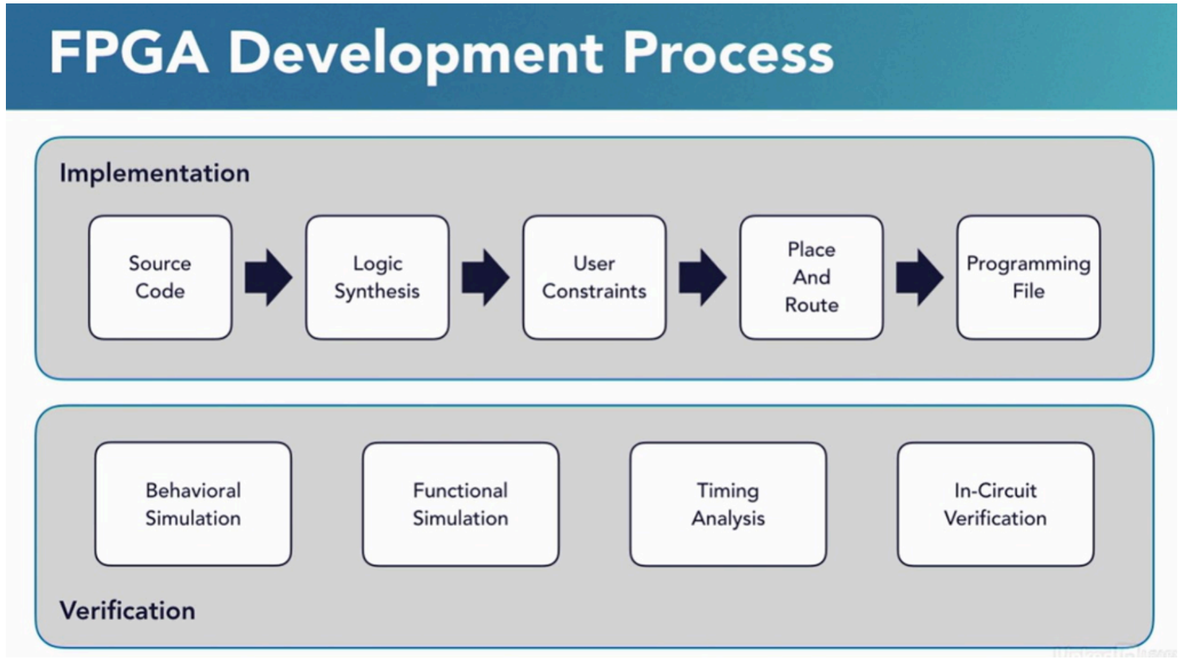


SIDE VIEW:



Programming Files

The programming files can be accessed at:
https://github.com/KL-Mithunvel/MyOwnEnigma/tree/master



VIVADO®

LEARNING OUTCOME

Understanding FPGA Implementation: Gain experience in using FPGAs for implementing digital logic systems, including designing and programming with hardware description languages (HDLs) like VHDL or Verilog

Digital Logic Design: Develop skills in designing and using digital logic gates to build functional components of the Enigma machine, including the substitution and permutation processes.

Interface Design: Learn to interface various hardware components, such as LEDs and 7-segment displays, with the FPGA to visually represent the state of the Enigma machine and user inputs

Making schematics: understnood how to draw schematics of various analog and digital components including keyboard, light up board and FPGA's.

Reference Page

https://www.cs.cornell.edu/courses/cs3110/2020fa/a1/

https://youtu.be/G2_Q9FoD-oQ?feature=shared

https://www.quora.com/What-digital-logic-circuit-do-I-use-to-make-an-Enigma-machine?ch=17&oid=60486229&share=82ea9b00&srid=u2GQ3&target_type=question

https://www.ciphermachinesandcryptology.com/en/enigmatech.htm

https://www.cryptomuseum.com/crypto/enigma/msg/p1030681.htm

Reference Page 2

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/2280/410-352_Web.pdf

https://www.emo.org.tr/ekler/a70aa1cbbf26e9c_ek.pdf

https://en.wikipedia.org/wiki/Enigma_machine?wprov=sfti1#Mathematical_analysis

https://crypto.stackexchange.com/questions/29315/how-does-the-ring-settings-of-enigma-change-wiring-tables?newreg=18aae1da1b1e4b8481d01c7d44bcaff5#

Mithunvel KL	23BMH1029
Kiran T	23BMH1028
Subbu	23BCE1259
Josika P	23BMH1129
Madhumitha v	23BMH1019