

Digital Planner System

Parker James Tribble BS in Computer Engineering 2022
Dr. Joseph T Wunderlich Ph.D.

OPPORTUNITY

Smart phones are quickly becoming more addicting as social media and other applications work to keep the users engaged as much as possible. To combat this, the digital planner system is designed to help increase productivity and improve one's relationship with technology. This planner system will have no distractions while still providing the same capabilities of a smart phone planner application.

DESIGN SPECIFICATIONS

Goal: Create an easy-to-use digital planner that automatically organizes entries and outputs easy to understand GUI

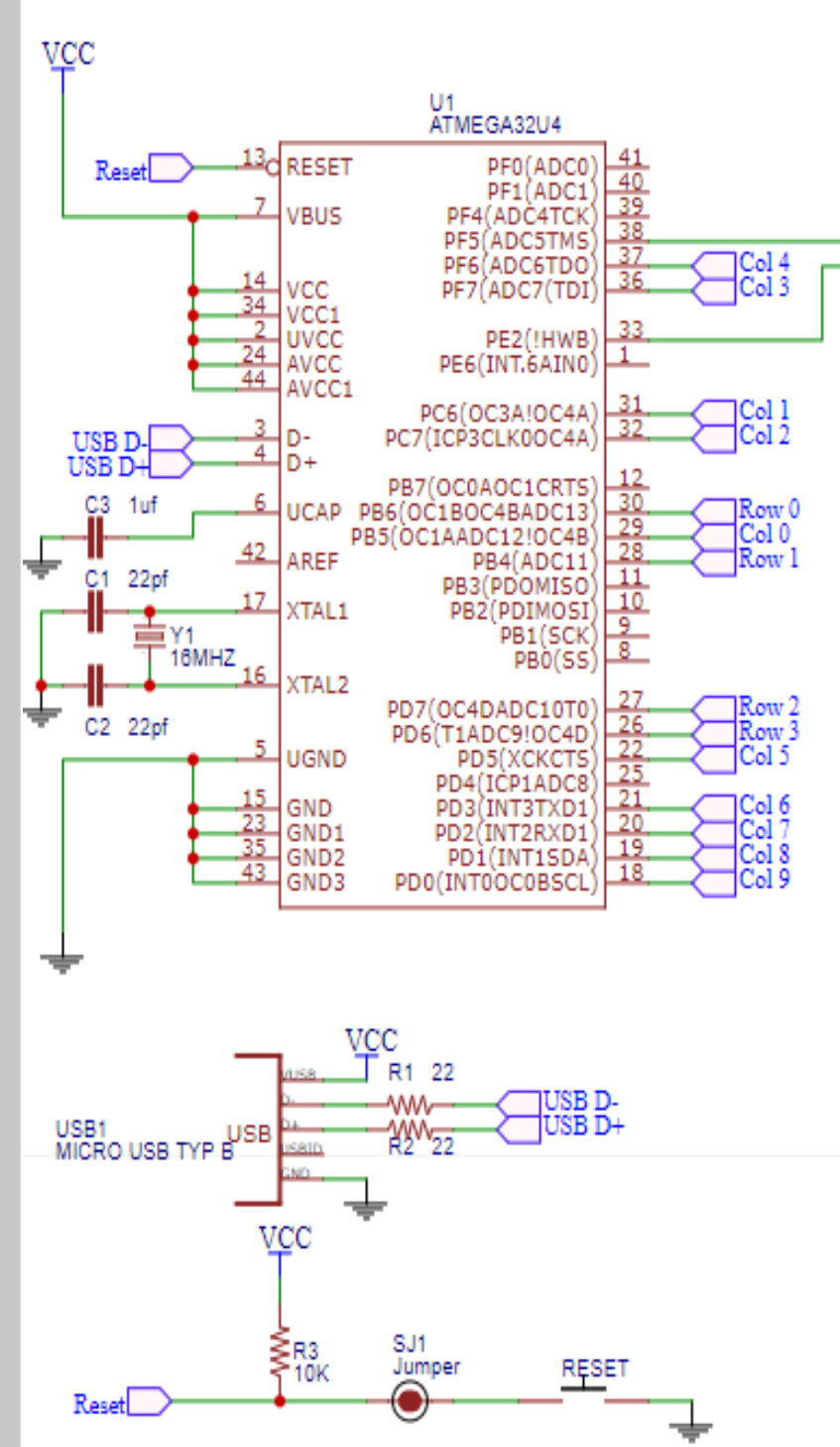
- Terminal:**
- Linux Ubuntu OS on Raspberry PI
 - Receive data from mobile unit via sftp
- Python program:**
- Organize data and output into CSV file and GUI
 - Set reminders to be displayed on mobile unit

- Mobile Unit:**
- Linux Ubuntu OS on Raspberry PI
 - PCB keyboard and 3.5-inch LCD Display
- Python program:**
- Take in planner entries and save them locally
 - Display GUI for user to input planner entries
 - Display user's current reminders

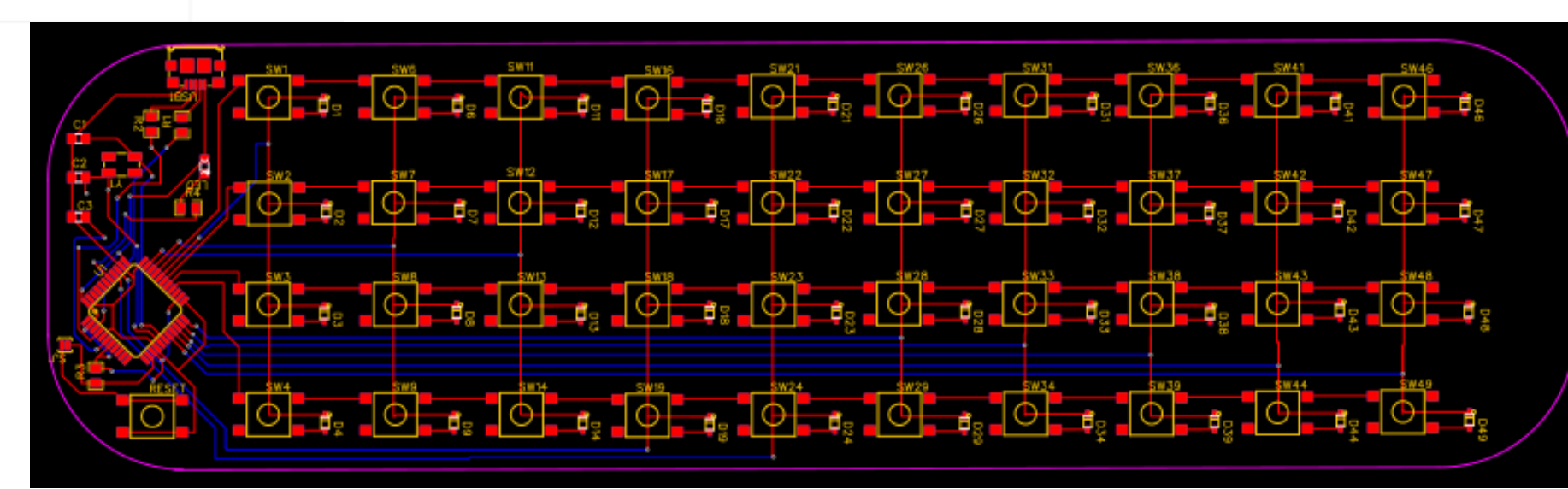
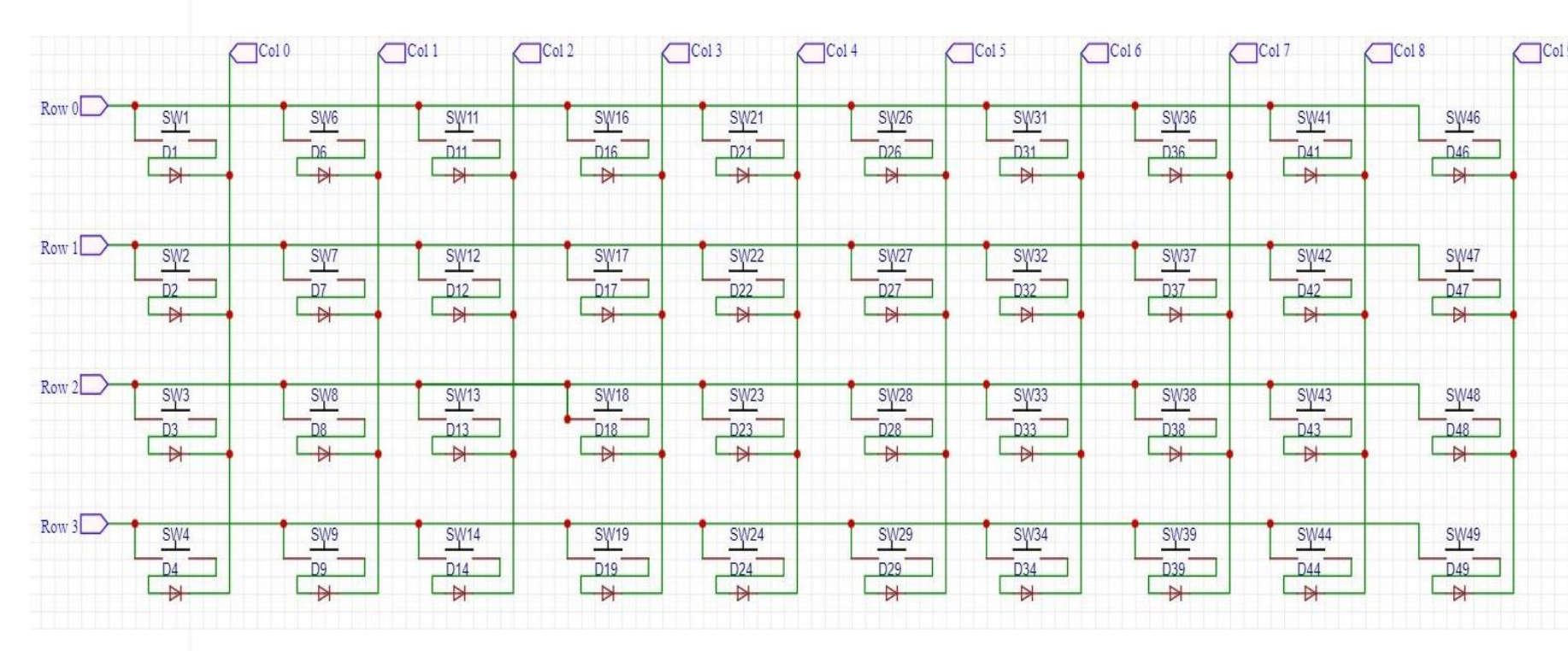
- PCB Keyboard:**
- Keyboard with 40 keys:
 - 26 for the alphabet. 10 for single digit numbers. 3 for space, backspace, and enter. 1 key for navigating the GUI (similar to tab key)

TECHNICAL DESIGN

PCB Keyboard:



Left: Microcontroller ATmega32u4 connections to PCB
Middle: Design of keyboard matrix
Bottom: 2-layer PCB design (Gerber file)



Software:

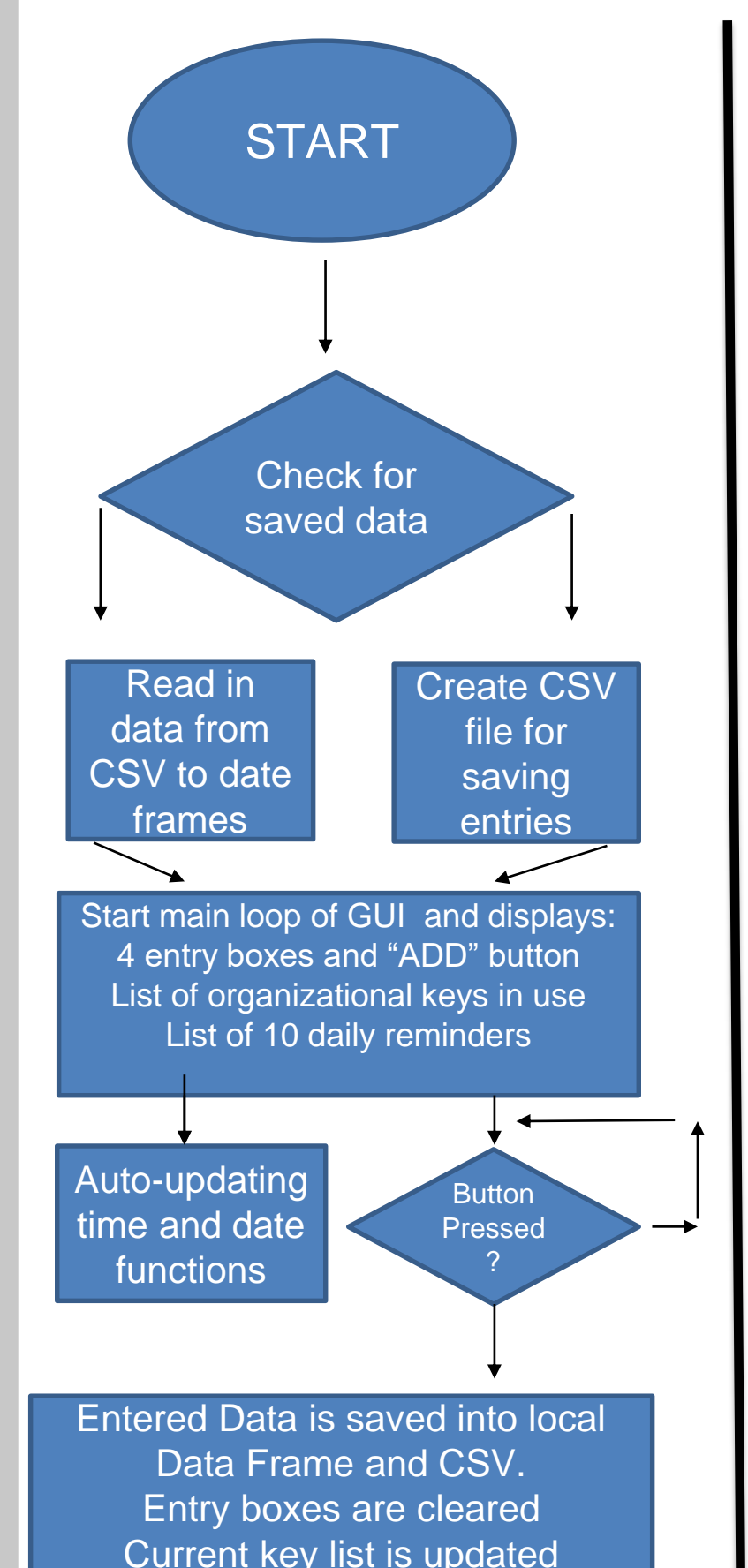


Figure 2: Mobile Unit: Raspberry Pi 3B V1.2 running Ubuntu OS. Utilizes Tkinter library for GUI and Pandas library with objects to handle data entries

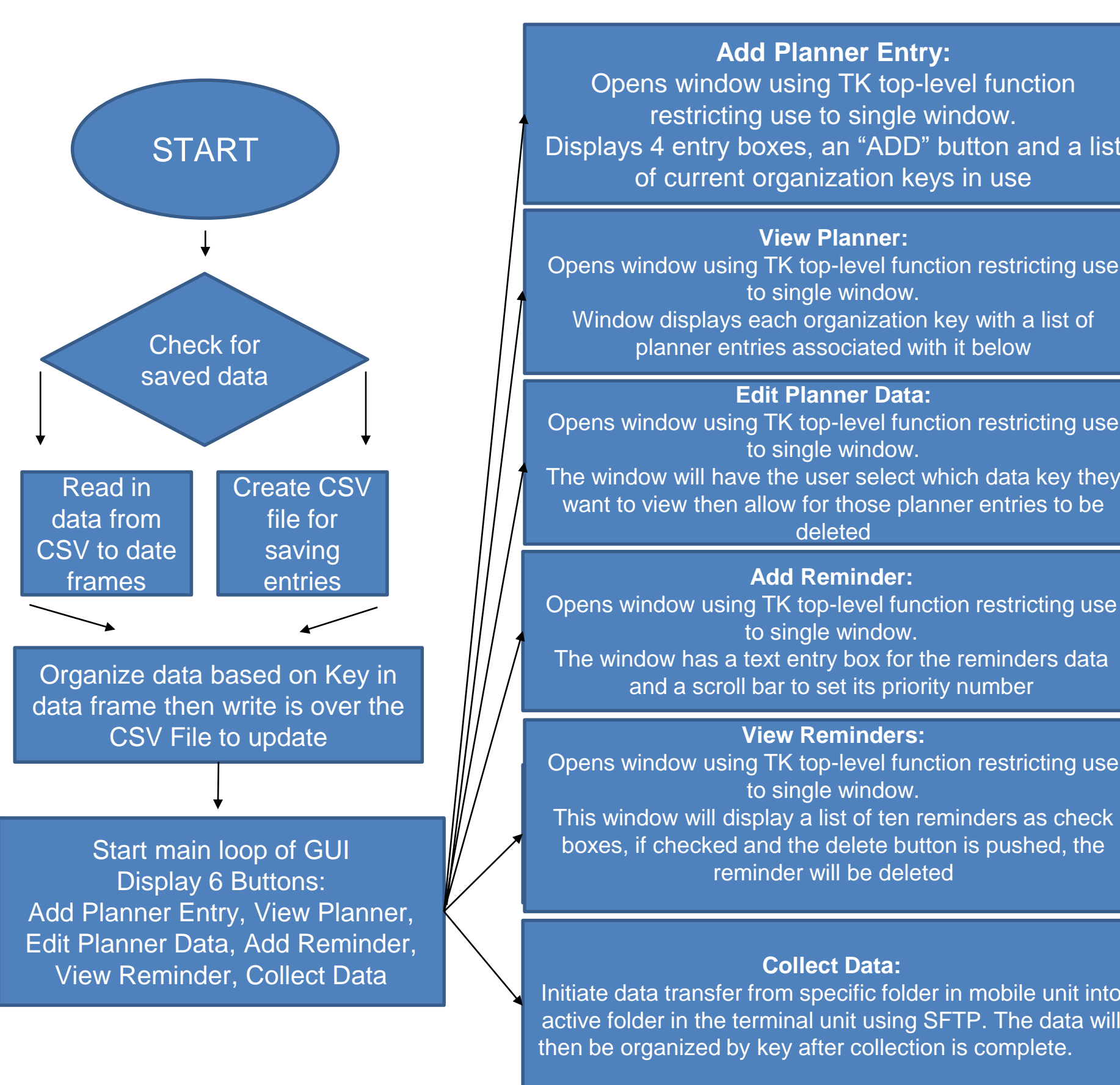
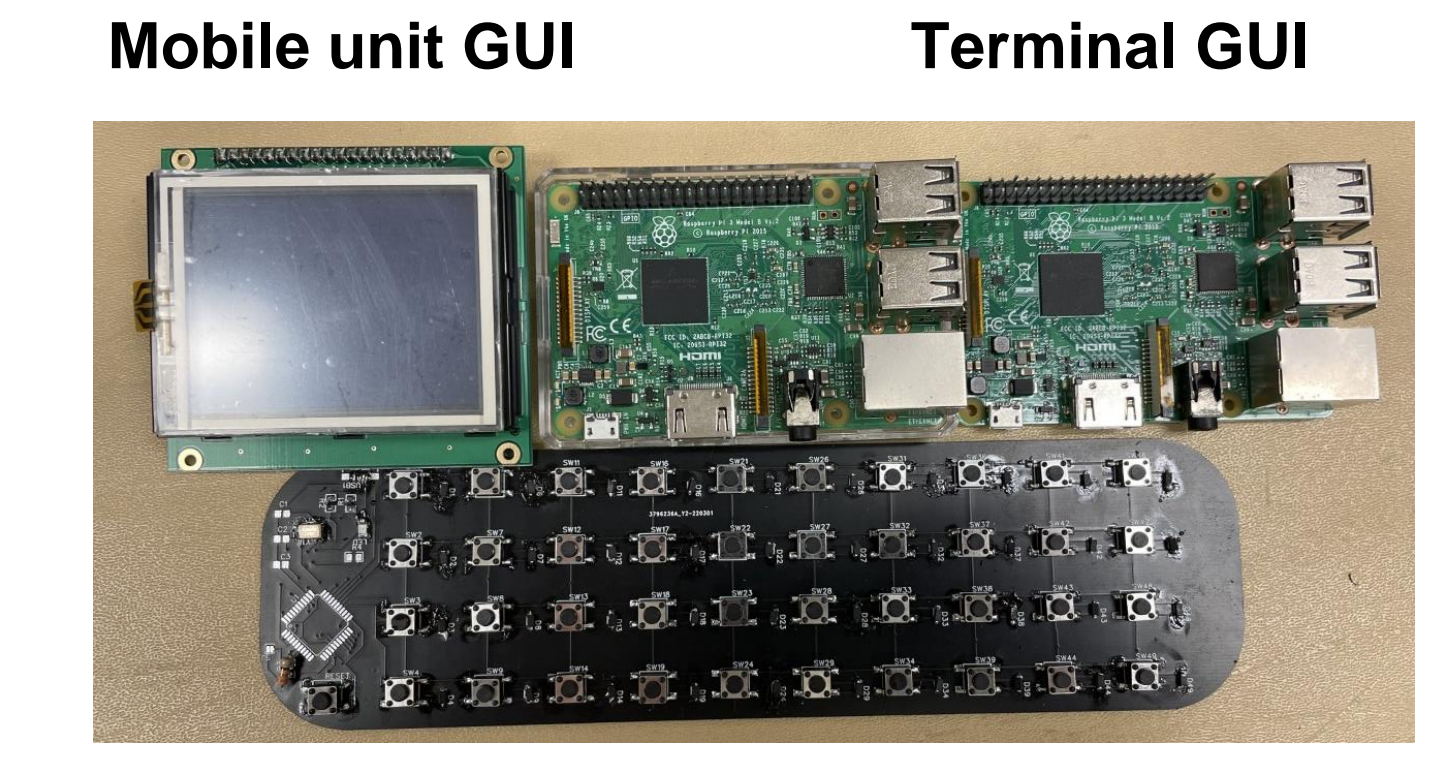
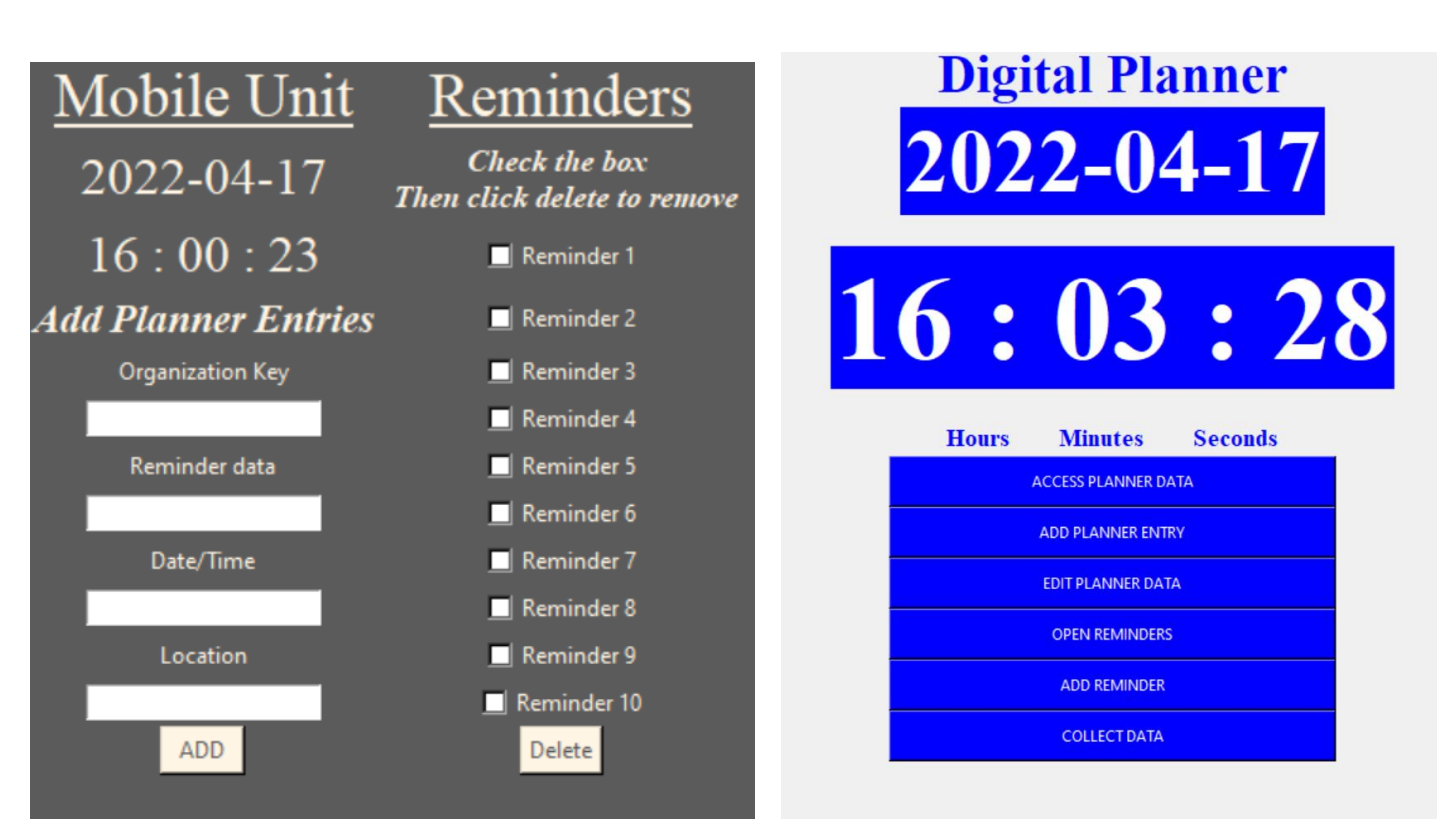


Figure 3: Terminal: Raspberry Pi 3B V1.2 Running Ubuntu OS. Utilizes Tkinter library for GUI and utilizing Panda's library along with objects to handle data entries. The Terminal also acts as the host server for the SFTP data transfer from the Mobile unit.

PROTOTYPE/DEVICE/PROCESS



Top row: LCD Screen, terminal and mobile unit raspberry Pi. **Bottom Row:** PCB Keyboard

Add reminders window **Remove planner entry window**

TESTING

Input testing: feed a series of different strings, integers, and unique characters to test if any do not function with in the code
Results: no characters or length of input would affect the program
Input testing: continue adding entries and reminders until the program stops functioning properly
Results: It was determined that the for functionality, a limit of 10 daily reminders was set. No limit needed for planner entries
Data Transfer: Transfer data from mobile unit to terminal writing over the CSV files that the programs read from
Results: Errors within data transfer come from either of the units lacking network connection or the SFTP server is not reachable
Firmware: testing of the keymap and firmware of the PCB keyboard was preformed on desktop keyboard to ensure functionality during keyboard assembly
Keyboard: after testing the firmware the keyboard should function properly. Any problems present will be hardware related.

IMPACT

This project is aimed to encourage the production of quality digital products that are designed to last and help people. Technology should assist or improve people's lives, not aim to take it over to keep engagement numbers up.

REFERENCES AND ACKNOWLEDGEMENTS

- [1] w4ilun github.com/w4ilun
- [2] ubuntu.com
- [3] kbfirmware.com
- [4] github.com/tmk/tmk_keyboard
- [5] pandas.pydata.org
- [6] docs.python.org/3/library/tkinter.html

Special thanks to:
Dr Joseph T Wunderlich & Matthew Grohotolski for technical support
Gio V, Lizzy Z, Louis S, for brainstorming sessions and creative input
Grant, Jim, Beth, and Ming Tribble for daily inspiration