FOCUS ASSISTANT

INTRODUCTION

Focus Assistant is powered by a Raspberry Pi Pico. It aims to lock items away from the user and/or others. It allows the user to set a locking time for up to 6 Hours.

USE CASES

There are many use cases for Focus Assistant, such as:

1. Preventing the usage of phones when driving, meetings or studying
2. Preventing overdose of medicine by scheduling an interval, especially so for elderlies who might forget if they already consumed the required amount
3. Hiding "Junk Food" away from the user, allowing him/her to access it for certain time of the day only.

LEARNING POINTS

Through this project, I managed to learn many different skills. Hence, I will split it into 3 parts: Design and Planning, Programming, Circuitry and prototyping.

DESIGN AND PLANNING

Reading the book "Atomic Habits: An Easy & Proven Way to Build Good Habits & Break Bad Ones" inspired me to create Focus Assistant. The book demonstrated that habits are formed through repetition and are influenced by what we see. For example, if someone wants to eat healthily, he should remove unhealthy snacks from sight. This will reduce their temptations for these unhealthy foods. During the exam period, I realised that I am frequently distracted by my phone. However, here comes the problem – I could not throw away my phone. And with that dilemma, I decided to create a device which can hide and even lock our distractions – phones, food or others.

A new microcontroller, Pi Pico, was released around that time. Currently, I only have experience in programming Arduinos. However, I sided towards Pi Pico as it offers many advantages such as faster processor, more accurate sensor readouts (16-bit DAC), larger program space, C++, Micro Python, RTC, and more…

As part of my presentation for Project Funding, I learnt Fusion 360 through Lynda.com and created a preliminary CAD drawing of Focus Assistant. All the files are available in hackster.io

PROGRAMMING

I learnt about the advantages and disadvantages between the two different programming languages, Python and C++. Python have the advantage of being a high-level language, meaning that it is easier to learn and have fewer restrictions as compared to C++. However, this means that the Microcontroller itself, must interpret the codes, which means longer processing time and higher power consumption. Meanwhile, C++ is compiled into machine code directly, which means faster processing time and lower power consumption. Having learnt Python in Semester 1, I will be using the python programming language.

When programming, I learnt communication protocols such as SPI/I2C, which allows for communication between components such as displays. This is useful as it has the potential to reduce the number of wires needed. For example, the display requires 16 pins. With I2C, it is reduced to 4.

As Pi Pico is a newly released microcontroller, some of its advertised features such as low-power mode and real time clock is unavailable. I've managed to overcome these challenges with alternative methods such as adding a manual switch to cut power when not in use and using of a library utime – which acts similarly to a real time clock. However, these alternative methods will result inconvenience to the user.

CIRCUITRY

I started by reading up the data sheet of Pi Pico and planned my circuit appropriately. Through the data sheet, I found out that Pi Pico operates at 3.3V. This means that all my components must be compatible with a 3.3V input. I then planned on how to wire the components by using the software Fritzing (available on hackster.io). When connecting components, I also learnt the importance of colour coding the wires and diagnosing the prototype when it does not work.

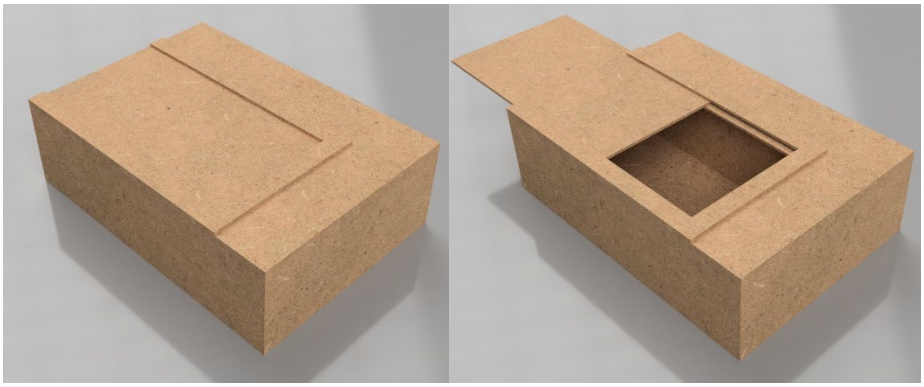
PROTOTYPING

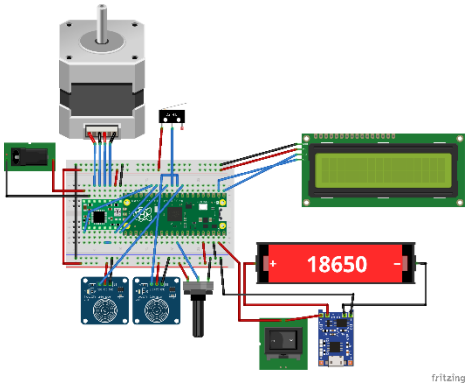
The prototype of Focus Assistant is made with cardboard, and acts as an illustration of what the actual product can do. In the YouTube video, I have demonstrated how Focus Assistant works.

DESIGN AND BUILDING PROCESS

Step 1

I begin by building a CAD rendering using Fusion 360 and designing the circuit.





Step 2

The components I ordered from Cytron arrived in 2 days. I start to explore Micro Python and coding it. After spending a week trying to interface with the E-ink display, it did not work as intended. This is because the official software for controlling it have not been released for Pi Pico yet.

Items includes the Micro Controller, wiring and prototyping items, battery charger, boost converter and a screen

Step 3

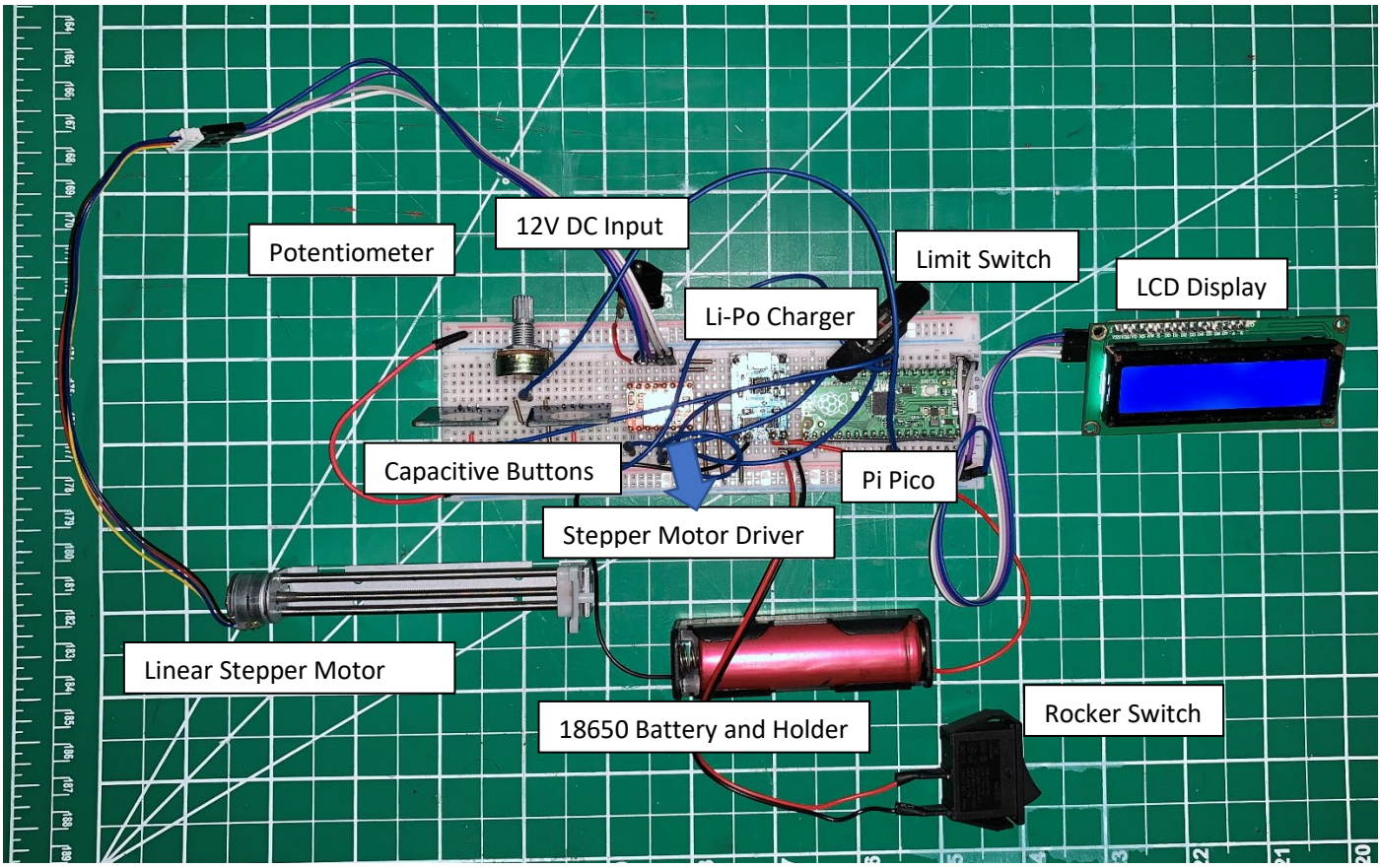
The components I ordered from Shopee arrived in 8 days. By this time, I have written a layout for the display. I Immediately added the components into the circuit, and it made further modifications to the code to integrate the components together.

From left to right:

1. Capacitive touch sensor – allows the user to interact with the microcontroller
2. Stepper motor driver – drives the linear stepper motor
3. Resistor and Potentiometer – regulates current and allows the user to select the time to lock for
4. Linear Stepper motor – lock and unlock the cover, it could not be moved by hand
5. Thermal Paste – Transfer heat away from the motor driver

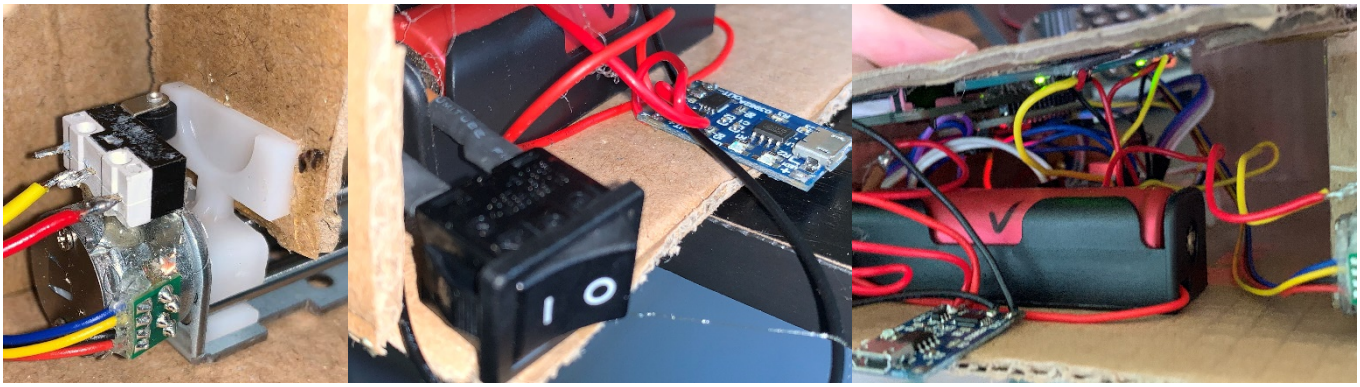
Step 4

This is a picture of the assembled circuit. I used my personal LCD display, 18650 battery holder and 18650 battery cell and a limit switch to assemble the circuit.



Step 5

After doing a final check to ensure that everything is working, I started building the prototype. I modified the cover mechanism to make it easier for the motor to move it.



Left to Right: Linear motor with limit switch to indicate position, Battery circuit with charger, interior of LCD display and buttons.

Step 6

Finally, I tested the circuit to see if it still works. It does not. After some time checking the connections (with the help of a multi meter) I discovered the fault: a 3V wire. And with that, Focus Assistant is complete!

CONCLUSION

This Prototype have the potential to allow people to concentrate and focus on their task better, as well as to build good habits.

I have identified several points of Focus Assistant which can be enhanced:

1. Inclusion of an RTC module to accurately measure time
2. 3D printing a limit switch mount for the motor
3. Laser cutting an enclosure

Links

hackster.io project: https://www.hackster.io/jianquan08/focus-assistant-1693b4
YouTube link demonstrating product: https://www.youtube.com/watch?v=lP9iciva5b0