Project 2 Pomodoro!

a study-helper.

IDEA310L

01

Statement of Problem

How can I study more efficiently?

TomatoTomahto

What is Pomodoro?

It’s a time-management technique

Developed by Francesco Cirillo in the 1980’s

Works great for students and people like me, who have trouble staying focused on tasks and with time-management skills

Pomodoro Continued...

It originally started as a kitchen timer.

But it’s more common to find it as a mobile app or browser extension

It’s broken up into 25-35 minute sections, with a small break between.

Generally though, people customize it to work best for them, which is usually includes a longer 15 or 20-minute break after 3 or 4 cycles, which makes a complete pomodoro!

Working in short bursts helps with preventing burnout.

The Pomodoro technique is great for studying too, because it forces you to break things into manageable chunks that can be worked on a little each day

This got me thinking

I really like using Pomodoro, but app and browser notifications tend to become background noise after a while.

How do I solve this issue? (Without a kitchen timer, obviously because I want to make one myself)

Design thinking.

Analysis

What problem did I want to solve? Studying is something I constantly struggle with.

Design

I like the look of tiny displays and well-designed objects that have one purpose and that are pleasing to look at.

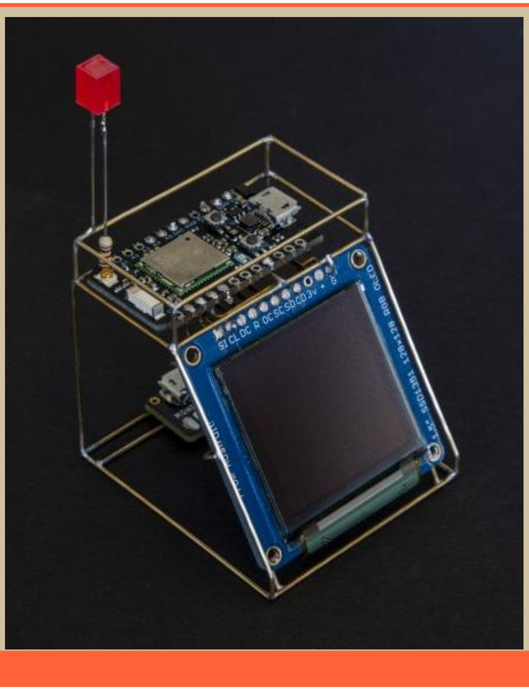
Evaluation

What modules would be best for implementing this? I decided I wanted the prototype to have a fun, toy-like feel

Implementation

I thought the 2.8” TFT we got in class would perfect for this, as well as the joystick*

Inspiration!



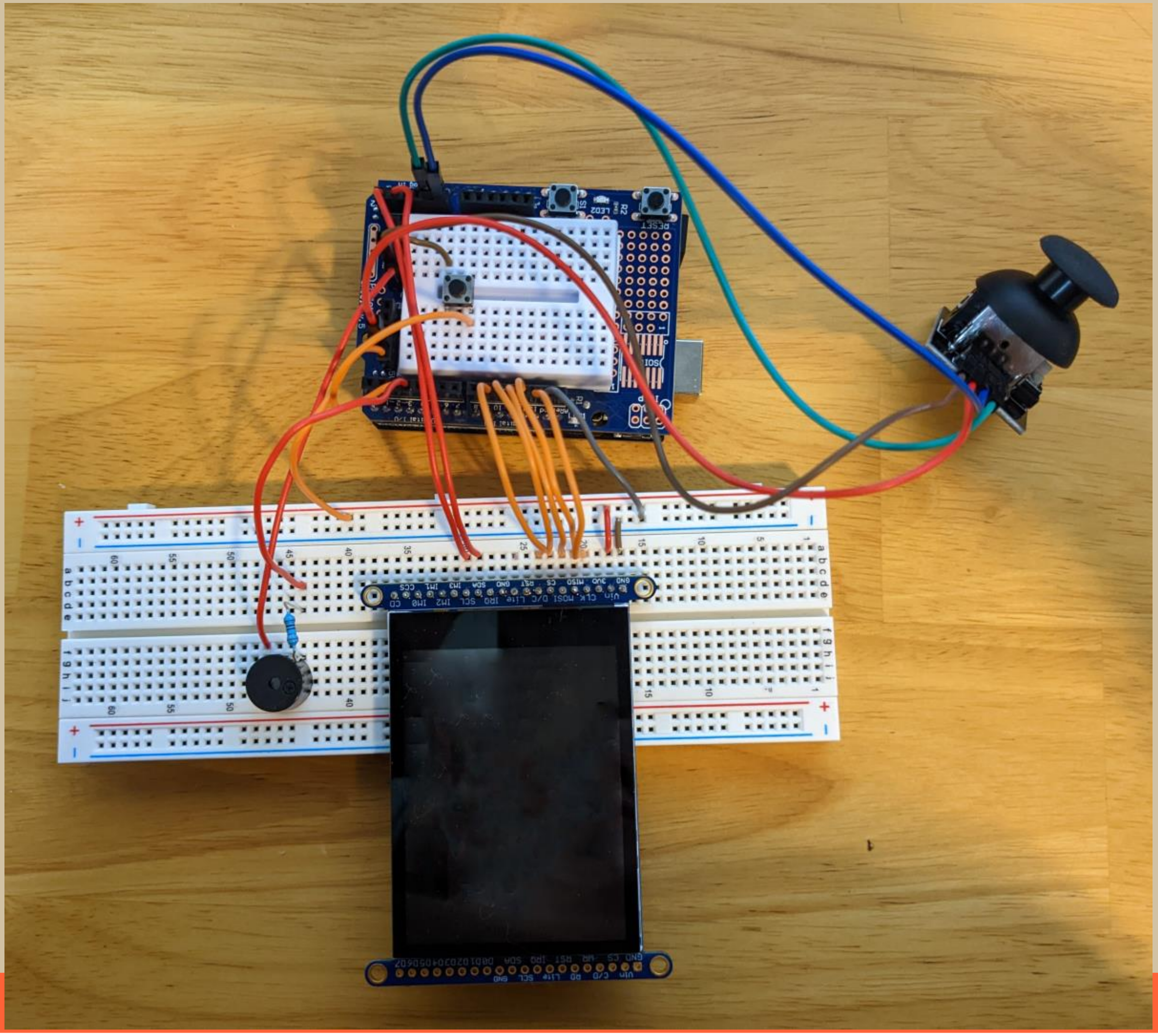
OLED Terminal Sculpture by Mohit Bhoite

Gameboy alarm clock by muid_amoy

PyPortal Titano Weather Station by Ruiz Brothers

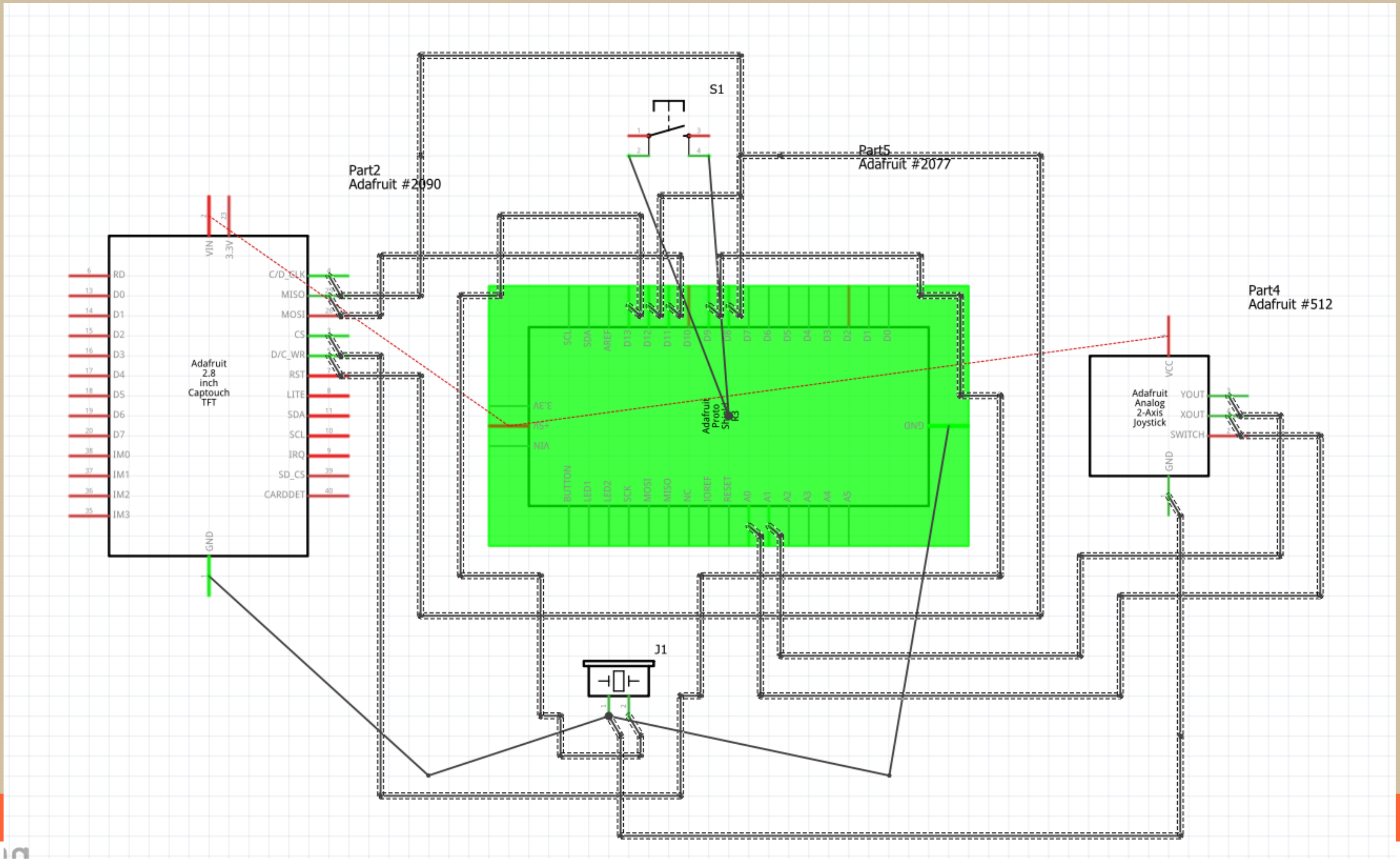
Design results!

Wiring Photo



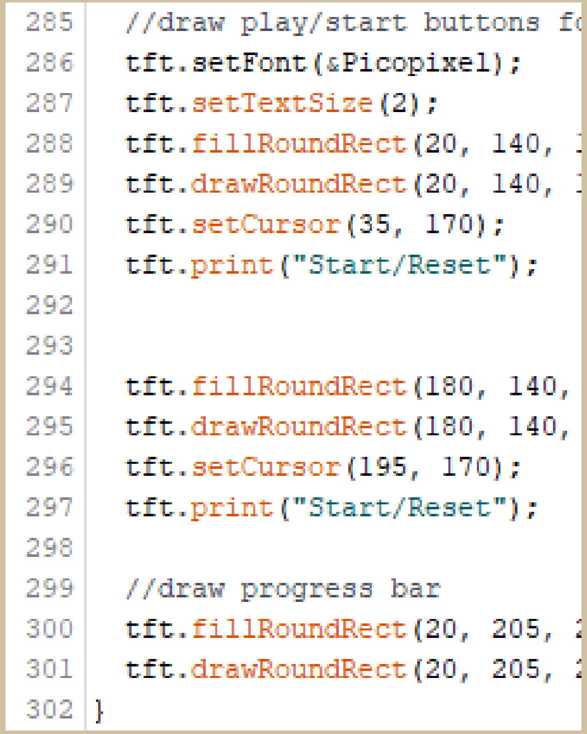
Design results!

Circuit Diagram



Some code...

A lot of this:

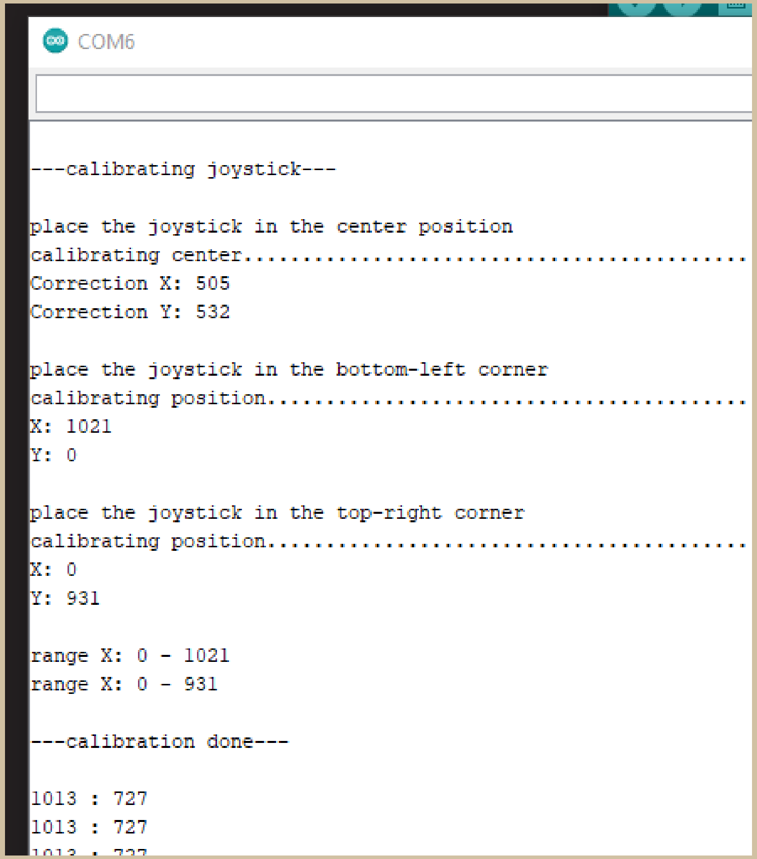


```
  //draw play/start buttons fo
  tft.setFont(&Picopixel);
  tft.setTextSize(2);
  tft.fillRoundRect(20, 140, 1
  tft.drawRoundRect(20, 140, 1
  tft.setCursor(35, 170);
  tft.print("Start/Reset");


  tft.fillRoundRect(180, 140,
  tft.drawRoundRect(180, 140,
  tft.setCursor(195, 170);
  tft.print("Start/Reset");

  //draw progress bar
  tft.fillRoundRect(20, 205, 2
  tft.drawRoundRect(20, 205, 2
}
```

Calibrating the joystick:



```
---calibrating joystick---

place the joystick in the center position
calibrating center.........................................
Correction X: 505
Correction Y: 532

place the joystick in the bottom-left corner
calibrating position......................................
X: 1021
Y: 0

place the joystick in the top-right corner
calibrating position......................................
X: 0
Y: 931

range X: 0 - 1021
range X: 0 - 931

---calibration done---

1013 : 727
1013 : 727
1013 : 727
```

03

Demonstration

I sure hope the joystick debouncing isn’t too bad

04

Challenges and Accomplishments

What I struggled with and what I’m really proud of

Design Challenges

- Making a GUI from scratch
- Joystick not cooperating
    - Same goes for the momentary button switch
- Finding libraries that were compatible with the ILI9341 chip
- Was stubborn and wanted to challenge myself to not use the touchscreen
- Interacting with the GUI
- Had to remove some fonts because of low memory

...and some accomplishments!

- Learned a ton about libraries
- Lots of piecing together and figuring things out
- Used a library that really helped me out (EventJoystick)
- Arduino interrupts
- Loading bar and the ADAGFX library
- Feel a lot more comfortable with breadboards now
- The GUI

Thank you!

Do you have any questions?

THANK YOU

Credits.

Presentation Template: SlidesMania

Fonts used in this presentation:Roboto Mono