

Name: Yingqi Chen

Project: Timed Lock Box

Libraries used:

BfButton.h – for long press and single press detection of the rolling encoder.

EnableInterrupt.h – running out the INT pin, need this library in order to use PCINT pin for interrupt purpose

Servo.h – for 360 degree continuous rotation servo.

Adafruit_SSD1306.h – for the OLED display basic functionality

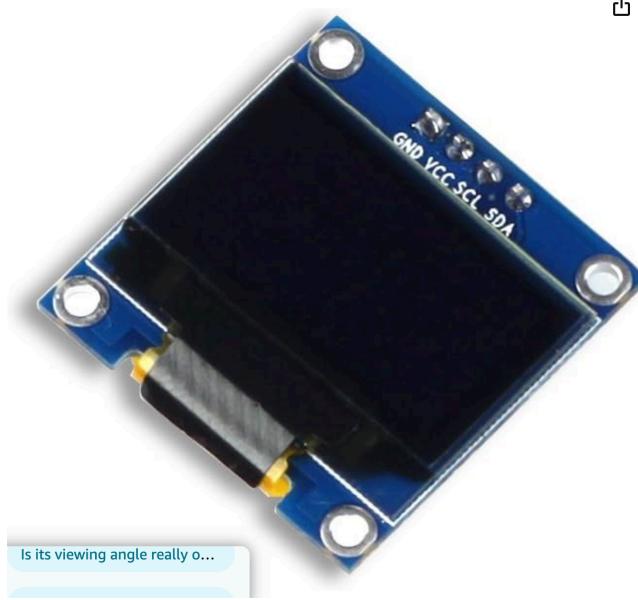
Fonts/FreeMonoBoldOblique12pt7b.h – The sub library from Adafruit to adjust the font on the OLED display for better visualization purpose.

Hardware Components Used:

3- 360 degree continuous rotation servos



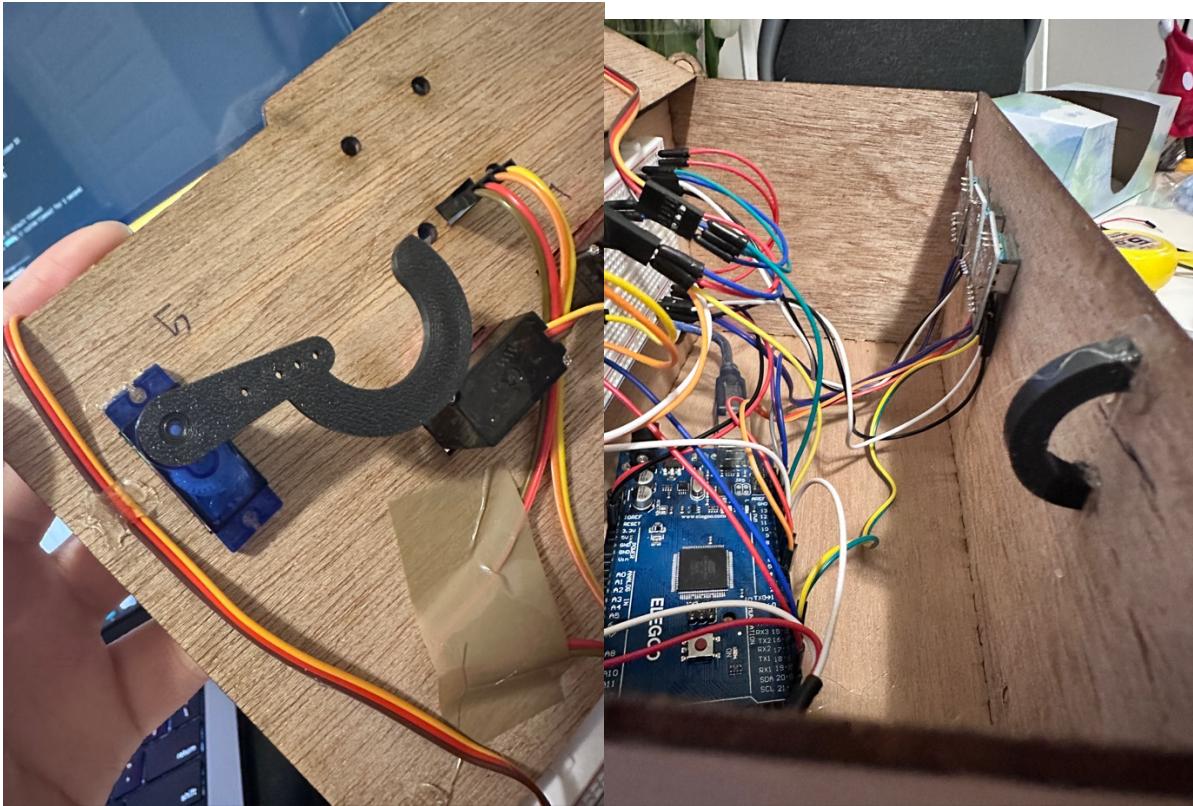
0.96 Inch OLED Module with pixel of 128 X 64 Yellow Blue SSD1306 Driver



3 KY-040 Rotary encoders module with 15 x 16.5 knobs cap



180 degrees built in servo from Arduino kits with 3D printing design component/ to open the door



Skills must learn beyond the skill demos:

The rotary encoder - The rotary encoder CLK and DT works differently, DT is after the CLK. And need to PULLUP to make pin always high to detect the change by using interrupt.

OLED display implementation – Need to know how the pixel works and separate the section on the screen

Continuous rotation servo – it stops at 90, and only controlling the speed. 0 and 180 will be the full speed. Not able to identify the position at all, not able to know the angle it rotated.

Process

Tested the rotary encoder first by using regular general purpose input output (digital pin). It works fine at first. But it faced the issue while testing with oled display. The result from the rotary encoder acts weird. The frequency is wrong. The reason it caused this issue is because both oled and the rotary encoder is using the same method and they have the conflict. Instead, using the interrupt to detect the change of the encoder works fine and no conflict with the oled display any more.

After the succeeded connection between oled display and the rotary encoder. I worked on figuring out the servos, trying to see what's the time to run 1 full round of the servo, in that case will be able to find the initial position. But unfortunately, this factor changed every time I recompile the project and did not have a good solution. That led me to change the pathway a bit, utilized the advantage of continuous rotation servo and using the speed as the identifier of each number, i.e. Hour, Minutes, and second.

The part to lock the latch is easier since it is control by stable PWM with a range to identify the angle, and successfully lock the box with customized 3D printed latch.