Style Guide REMOVE BEFORE HAND IN

Italics - Name of component i.e. naming buttons to differentiate them

Introduction - needs re-written

Combination locks are incredibly useful devices that allow the user to secure containers, doors, bikes etc against theft or damage. Adding to their utility they do not even need a key meaning individuals do not need to worry about carrying or losing one and companies to not need to pay to issue every employee or However many of them are critically flawed, able to be bypassed in seconds with the right tools, another issue is that given enough time an attacker may have time to simply guess the code.

Concept

The basic idea for the user experience is to have three rotary potentiometers and a button, the user can input the code for the lock by changing the rotation of the dials and then pressing the button (referred to as the *Confirmation Button*) to confirm the input. If the combination was correct a solenoid will pull back the bolt unlocking the lock, this bolt will be held back for five seconds, enough time for a person to walk through the door and the door closer to close it behind them before the bolt is extended again and the door is locked. *I may also add a feature allowing the timer to be deactivated allowing the door to remain unlocked indefinitely but this is not a priority at present*

Colour Coding

- Red - Live
- Black - Ground
- Purple - input
- Green - Data within subsystem
- White - Data Between subsystems

Subsystem 1 - Dials

Design

A single dial has to output a logic high signal when the resistance of a potentiometer(R_{VI}) is between two values and output low when the resistance is not between these two values. The values needed to output high are controlled by a separate potentiometer I am calling the setting potentiometer (R_{VS} Symbol may need changing), the setting potentiometer will have the same

maximum resistance as the input potentiometer making the desired value easier to set. To compare the values of the potentiometers I use two comparators, comparator 1 has R_{VS} connected to the

Parts List (Per Dial)

- 1KΩ resistor x3
- Lm311 Comparator x2
- 10KΩ potentiometer x2
- Diode x1
- Quad AND gate (1 per four dials)

Combination REMOVE BEFORE HAND IN

A - right
B - just right of A
C - Pointing at bottom left of confirmation button

Testing

Measurements taking with logic probe and analogue voltmeter with a range of -1 to +5V and resolution of 0.1V.

Dial A

$V_{R_{VS}}$ = 1.9V
Voltage drop between comparators = 2.6 - 1.9= 0.7
Voltage range to give a Logic high output = 2V < V_{RI} ≤ 2.6v
Dial functions correctly. No further testing needed.

Dial B

$V_{R_{VS}}$ = 2.4V
Voltage range to give logic high output = 2.6V < V_{RI} ≤ 3.2V
Dial functions correctly. No further testing needed.

Dial C

$V_{R_{VS}}$ = 3.8V
Voltage range to give logic high output = 3.9 < V_{RI} ≤ above max range
with further testing I found that the output actually acted in accordance with $R_{VI} < V_{RI}$
Dial is built incorrectly. Examination and rebuild is required.

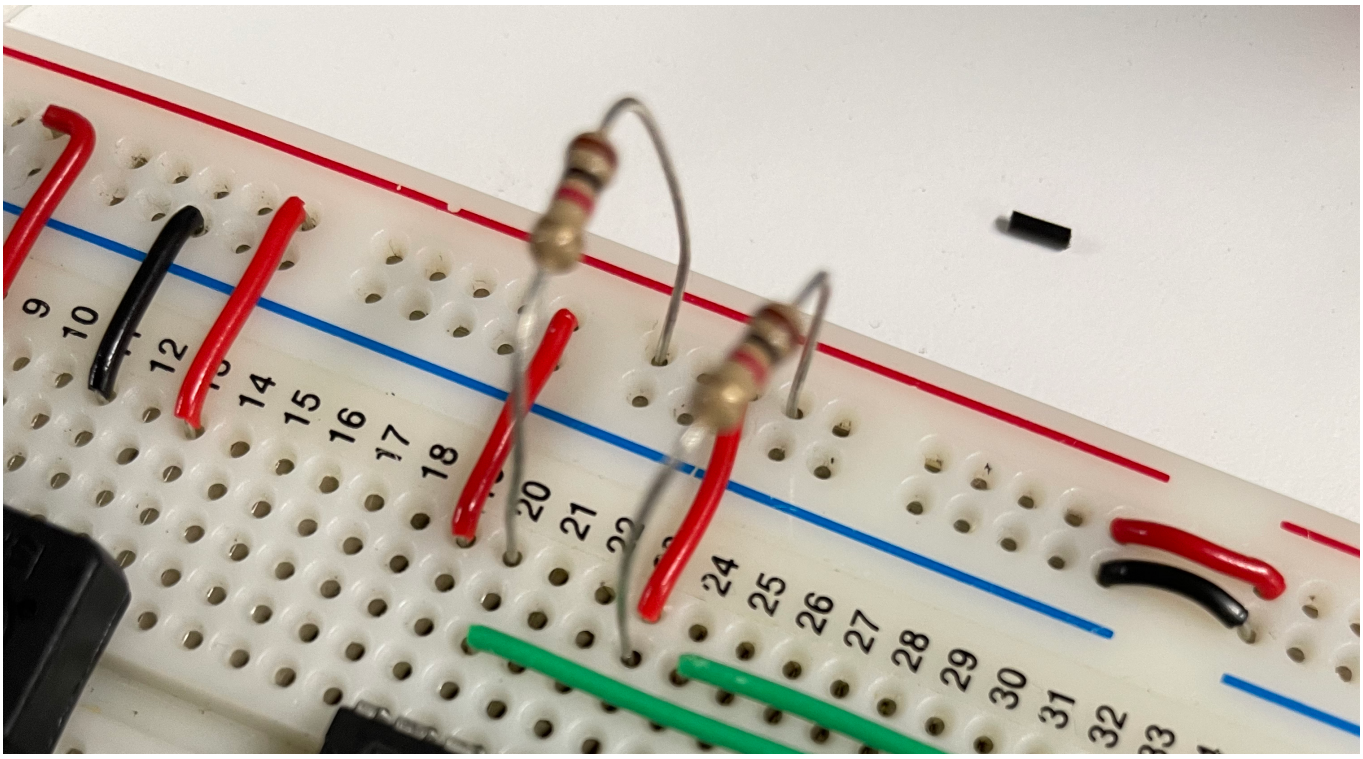
Dial C second test

Connected R_{VS} to the non-inverting input of pot C_{ii} (Fixed Error 2)
$V_{R_{VS}}$ = 3.8V
Voltage range to give logic high output = 3.9 < V_{RI} ≤ 4.6V
Dial now functions correctly. No further testing required.

Building Errors

Error 1

While building the first dial subsystem I placed the collector resistor of the second comparator in the wrong place, instead of placing it between the collector and the +5V rail I placed it between the Vcc of the LM311 chip and the +5V rail. This resulted in the comparator outputting +1.5V when high instead of +5V, this was not enough to trigger the AND gate so the subsystem never had a final output of Logic High as it should when the input potentiometer is in the correct position.
(The Collector resistor in the incorrect place, it should have been in row 24 to the right of the red wire)



Error 2

missing wire from setting pot to comparator cii

Final layout

(highlighted image showing where each subsystem is)

Subsystem 2 - Comparison Logic

Design

This subsystem will I also need to trigger an alarm/alert if the code is wrong *If I have enough time I will add a system allowing for multiple tries*. This will be represented by an LED, in a real world use of the system this would be replaced with either an alarm or another system to send a notification to someone inside the building.
To make the comparison logic I first decided that I wanted two outputs, one for the solenoid and one for the alarm. These will be named Q_1 and Q_2, Q_1 will be connected to the solenoid and Q_2 will be connected to the alarm. To get the outputs I AND together the three inputs from the three dials, this signal is then ANDED with the Confirmation Button so that when all three dials are in the correct positions and the button is pressed Q_1 will be set high until the button is released, at this stage the button is not debounced. To get Q_2 I NOT the output of AND gate 3 and then AND that with the Confirmation button, this signal is then fed into the S input of an S-R latch, another button called $AlarmReset$ is attached to the R input; this button would not be accessible to the user on the outside but could be used by someone on the other side of the door to turn the alarm off once it had been started.
Q_1 will be fed into the input of a 555 monostable, this will extend the signal to 5 seconds.

Parts List

- Quad AND gate
- Quad NAND gate
- Hex Inverter chip
- 555 Timer IC
- 100KΩ Resistor
- 1KΩ resistor x2
- 220Ω Resistor
- 100µF capacitor
- 10nF capacitor
- Tactile switch x2

Testing

Q_1

A = 1
B = 1
C = 1

Q_1 = high when button pressed but does not go low once button is released needs further testing.

Q_1 test two

A = 1
B = 1
C = 1
Q_1 = 1 when button pressed and 0 when button is not pressed

A = 1
B = 1
C = 0
Q_1 = 0

Q_2

A = 1
B = 1
C = 1
Q_2 = 0
A = 1
B = 1
C = 0
Q_2 = 1 when button is pressed and o when it is not.

555 Monostable

test 1

does not work

- needs examination and rebuild

test 2

after connecting the power rails (Error 2 - Wrong rail) and pressing the Confirmation Button the signal changed from low to high and stayed high for 5.2 seconds.

- Monostable functions correctly no further testing needed

SR Latch

test 1

Woke up with both Q and $\overline{Q}$ high, this is an illegal state.
Alarm Reset did not reset the latch

test 2

After making *Alarm Reset* active low (Error 3 - alarm reset) SR latch now reset but set when all dials were in the correct positions

test 3

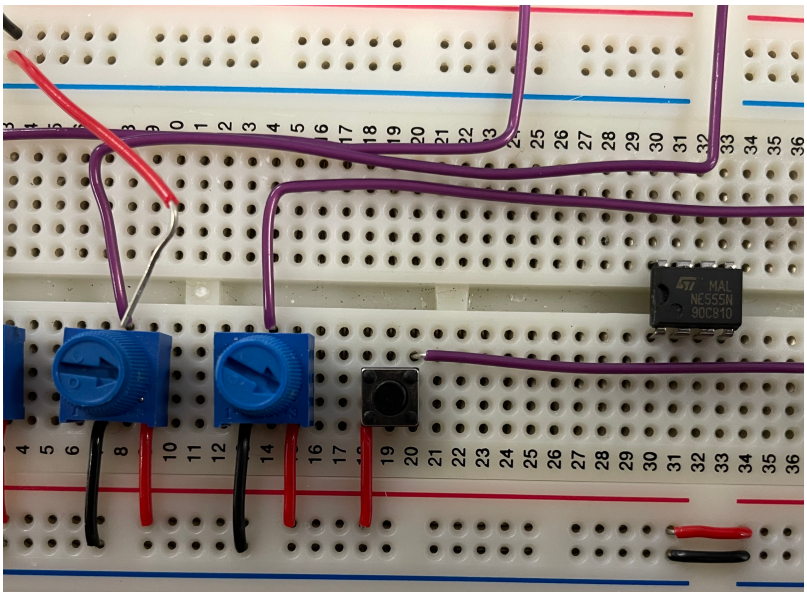
after swapping the output that set the SR latch (Error 4 - wrong output) the SR latch works as intended

- SR latch functions correctly no further testing needed

Building Errors

Error 1 - missing resistor

Forgot to add pull down resistor to the tactile switch



Error 2 - Wrong rail

The live rail of the second breadboard was giving a logic low signal when measured, this is the rail pin 8 of the 555 IC was connected to. This pin is the +V_{CC} so when the rail has no voltage the chip is entirely unpowered.
Upon further inspection I realised I had not connected the left side of the rails (the side with the 555 attached) to the right hand side of the rails, this was a very easy fix just connect the left to the right side of the power rails.

Error 3 - alarm reset

Alarm Reset was active high but it should be active low as it is the reset of an S-R latch, this resulted in the SR latch resetting immediately rendering it useless.

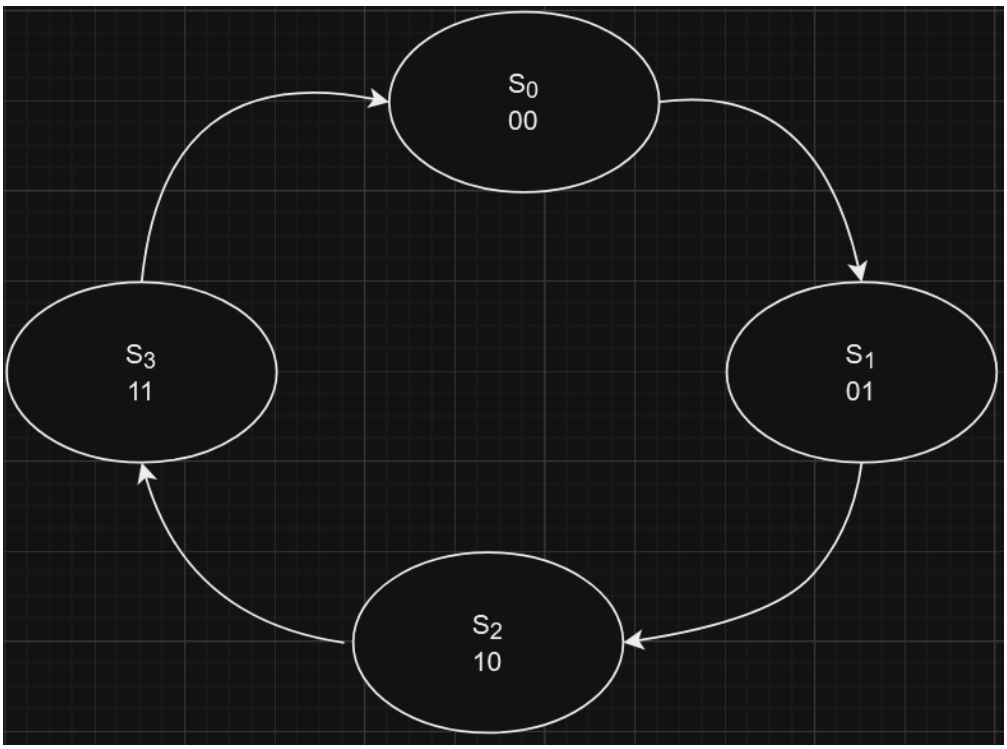
Error 4 - wrong output

Q_1 was fed into the $\overline{S}$ input of the SR latch instead of Q_2 this meant the alarm would go off whenever the combination was correct

Subsystem 3 - Counting logic

Design

One problem with the alarm system above is that the alarm will go off if there is a single wrong combination. In the real world this would cause a very high number of false alarms as the users accidentally input the wrong combination. To counter this I will add a system allowing the user to have three tries to open the lock before the alarm is triggered.
To do this I have decided to use a synchronised counter.
I starter by considering how many states I would need and thus how many counter bits would be required, I determined that I would need four states $4_{denary} = 11_{binary}$; from this I can tell that I need a two bit counter as 11 takes up two bits.
I then determined the state diagram:



As shown there are no illegal states to worry about so there is no logic to change/remove those.
In addition to this the logic to determine when to trigger the alarm will be extremely simple

Subsystem 4 - Solenoid

This goal of this system is to move the bolt of the lock out of the locked position to move the bolt of the door to unlock it. To do this I will use a solenoid, a coil of wire surrounding a ferrous core that is moved by the magnetic field formed when a current moves through the coil. Technically the term solenoid refer to the coil of wire itself however for this report I will be referring to the entire assembly as the solenoid. The solenoid will be driven by a transistor as the current required will be far too high for it do be driven directly from the 555IC it will be controlled by. This 555IC will be configured as a monostable so that the bolt will be held open for a set amount of time before being closed again.
For my purposes a Pull solenoid is needed, in this type of solenoid the armature pulled in by the magnetic flux formed when it is active and them

To begin designing this section of the subsystem I started by finding the values of the resistor and capacitor used to control the To get the values for the resistor and the capacitor I started by assuming the period, T, would be 5 or close to it as I wanted the lock to stay unlocked for 5 seconds.
Target for T = 5

$$T = 1.1RC \approx 5$$

Rearrange to find R

$$5 = 1.1RC \therefore \frac{5}{1.1} = RC \therefore 4.54 = RC \therefore R = \frac{4.54}{C}$$

I then looked at the values of capacitors available to me (E3 or E12) and decided that a 100µF capacitor was the best option, from this I found that I would need a 45454.54Ω resistor.

$$R = \frac{4.54}{C} \therefore R = \frac{4.54}{100 * 10^{-6}} = 45454.54 \approx 45.5K\Omega$$

I only have access to resistors in the E24 series of preferred values so the closest I can get to this value is 47KΩ. This gives a period of 5.17 seconds which is within an acceptable range from 5 seconds.

$$45.5K\Omega \approx 47K\Omega$$

$$1.1 \times 47 \cdot 10^{3} \times 100 \cdot 10^{-6} = 5.17s$$