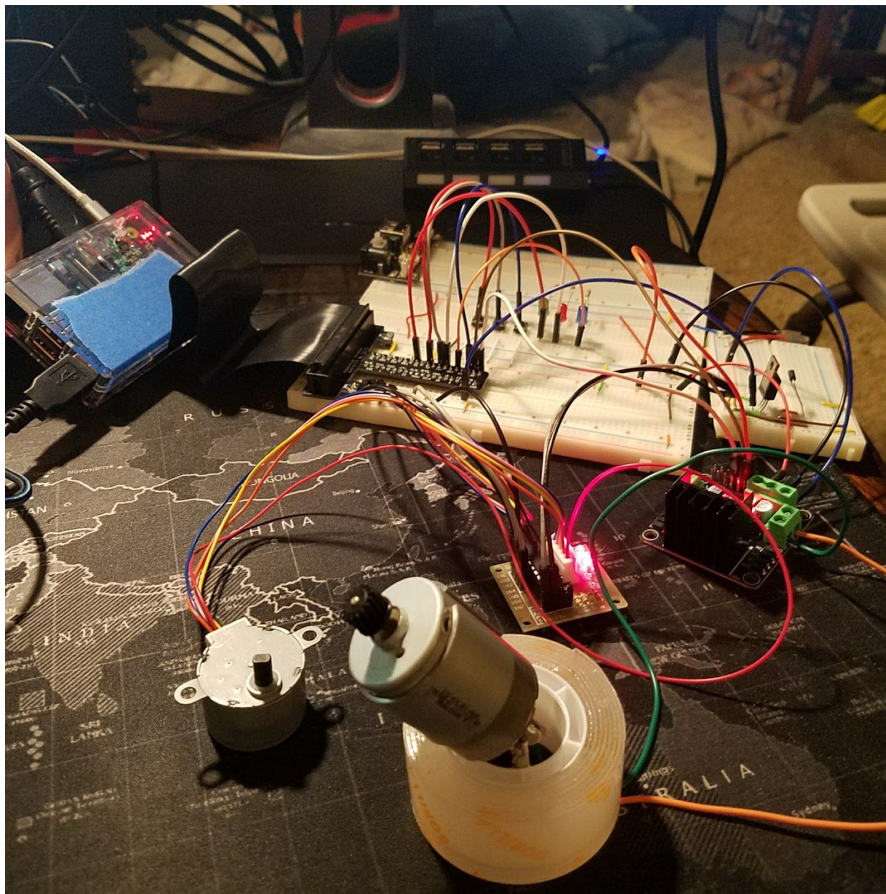


### EEE 174 Final Combined

#### Components:

- 12 Vdc power supply
- Vdc stepper motor
- Motor Breakout board
- 5Vdc power supply
- 12V DC Water Valve
- NPN transistor
- Diode
- Capacitive Touch sensor
- 3 x 470  $\Omega$  resistor
- 3 x LED
- L298 dual H-bridge on breakout board
- 12 Vdc motor



I ran into issues integrating every part of this lab. First, I was unable to get the LCD from lab 10 to function. The next lab I integrated was lab 8 with the touch sensors. I purchased a 4 digit touch sensor, but I could not get the pi to detect the sensor, so for the rest of the lab I used push buttons instead of touch sensors. When I wanted one of the motors or the valve to turn on, I could press the corresponding button. A LED would light up and some status text would print to the terminal. The next lab I added was the motor and H-bridge, lab 9, this lab worked as

expected. The last lab I tried to integrate was lab 6. I ran into major issues and I could not get the stepper motor code to work within the lab 10 code. Lab 6 worked fine on its own, but I could not get the code to work from pushing a button. Below is the code I wrote to try to get everything working together. Considering this is my first time working with python, I think I did okay learning some basic troubleshooting, with some time and guidance I could figure out how to get the rest of the project working.

```
#combined.py
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
try:
    StepPins = [2, 3, 4, 17]
    GPIO.setup(StepPins, GPIO.OUT)
    Seq = [[1, 0, 0, 0],
           [1, 1, 0, 0],
           [0, 1, 0, 0],
           [0, 1, 1, 0],
           [0, 0, 1, 0],
           [0, 0, 1, 1],
           [0, 0, 0, 1],
           [1, 0, 0, 1]]
    StepCount = len(Seq)
    StepDir = 1
    WaitTime = 0.005
    # Initialize variables
    StepCounter = 0
    LoopCounter = 0
    loopmultiplier = 1600 #approx 1 rev
    TotalLoops = 2*loopmultiplier # gives approx n turns
    Loops = True # statement to break loop

    # set pin for touch sensor
    PadPin1=16
    LedPin1=12
    GPIO.setup(PadPin1, GPIO.IN)
    GPIO.setup(LedPin1, GPIO.OUT)
    pressed1 = False

    PadPin2=7
    LedPin2=1
    GPIO.setup(PadPin2, GPIO.IN)
    GPIO.setup(LedPin2, GPIO.OUT)
```

```
pressed2 = False
```

```
PadPin3=25
```

```
LedPin3=24
```

```
GPIO.setup(PadPin3, GPIO.IN)
```

```
GPIO.setup(LedPin3, GPIO.OUT)
```

```
pressed3 = False
```

```
button1=False
```

```
button1on="stepper motor on"
```

```
button1off="stepper motor off"
```

```
button2=False
```

```
button2on="valve opened"
```

```
button2off="valve closed"
```

```
button3=False
```

```
button3on="forward/reverse motor on"
```

```
button3off="forward/reverse motor off"
```

```
while True:
```

```
    PadPressed1=GPIO.input(PadPin1)
```

```
    PadPressed2=GPIO.input(PadPin2)
```

```
    PadPressed3=GPIO.input(PadPin3)
```

```
    if pressed1 and not PadPressed1:
```

```
        button1=False
```

```
        print(button1off)
```

```
        GPIO.output(LedPin1, GPIO.LOW)
```

```
    if PadPressed1 and not pressed1:
```

```
        button1=True
```

```
        print(button1on)
```

```
        GPIO.output(LedPin1, GPIO.HIGH)
```

```
    pressed1=PadPressed1
```

```
    time.sleep(0.1)
```

```
    if pressed2 and not PadPressed2:
```

```
        button2=False
```

```
        print(button2off)
```

```
        GPIO.output(LedPin2, GPIO.LOW)
```

```
    if PadPressed2 and not pressed2:
```

```
        button2=True
```

```
        print(button2on)
```

```
GPIO.output(LedPin2, GPIO.HIGH)
pressed2=PadPressed2
time.sleep(0.1)

if pressed3 and not PadPressed3:
    button3=True
    GPIO.output(LedPin3, GPIO.LOW)
if PadPressed3 and not pressed3:
    print(button3on)
    GPIO.output(LedPin3, GPIO.HIGH)
pressed3=PadPressed3
time.sleep(0.1)

if(button1==True):

    while(Loops):

        for pin in range(0,4):
            xpin = StepPins[pin] #Get GPIO

            if Seq[StepCounter][pin]!=0:
                GPIO.output(xpin, True)
            else:
                GPIO.output(xpin, False)
            StepCounter += StepDir
            LoopCounter += StepDir
            # statement to stop motor
            if (LoopCounter==TotalLoops):
                Loops = False
            # If the sequence ends
            # start again
            if (StepCounter>=StepCount):
                StepCounter = 0
            if (StepCounter<0):
                StepCounter = StepCount+StepDir

            # Wait before moving on
            time.sleep(WaitTime)

if(button2==True):
    ValveTime=10 # set time for valve to operate
    basepin = 26 #set pin for base
    GPIO.setup(basepin, GPIO.OUT)
```

```
GPIO.output(basepin, GPIO.HIGH)
time.sleep(ValveTime) # valve leave valve open for set time
GPIO.output(basepin, GPIO.LOW)

if(button3==True):
    forward = 20 #GPio pin
    backward = 21 #GPio pin
    sleeptime=1 #wait timer
    ftime = 2 #forward timer
    rtime = 2 #reverse timer
    loopcount = 2 #loop counter

GPIO.setup(forward, GPIO.OUT) # assign pin as output
GPIO.setup(backward, GPIO.OUT) # assign pin as output

def Forward(): #define function forward
    print("forward") # print forward
    GPIO.output(forward, GPIO.HIGH) # set pin assigned to forward high
    time.sleep(ftime) # leave pin high for some set time
    GPIO.output(forward, GPIO.LOW) # set pin low

def Reverse(): # define function reverse
    print("reverse") # print reverse
    GPIO.output(backward, GPIO.HIGH) # set in assigned to reverse high
    time.sleep(rtime) # leave pin high for some set time
    GPIO.output(backward, GPIO.LOW) # set pin low

for x in range(0,loopcount):
    x += 1
    Forward() # turn motor in forward direction
    time.sleep(sleeptime) # wait
    Reverse() # turn motor in reverse direction
    time.sleep(sleeptime) # wait
    button3=False
    print(button3off)

except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

From Terminal: