Christopher Luey

28.4.1 #7, 28.4.7 #7, 28.4.9 #8 And all of your code, well commented!

28.4.1

1. The NU32 communicates with the encoder counter by an SPI channel. Which SPI channel will you use? Which NU32 pins does it use?

The NU32 will use UART2 for communication with the encoder counter, utilizing pins RPB0 (pin 4) for U2RX and RPB1 (pin 5) for U2TX.

2. The NU32 reads the MAX9918 current sensor using an ADC input. Which ADC input will you use? Which NU32 pin is it?

INA219: INA219 SDA to PIC32 SDA1 (pin 18), INA219 SCL to PIC32 SCL1 (pin 17)

3. The NU32 controls the DRV8835 H-bridge using a direction bit (a digital output) and PWM (an output compare and a timer). Which peripherals will you use, and which NU32 pins?

Direction bit: RB10 (pin 21).
PWM signal: Output Compare 1 (OC1) mapped to RB15 (pin 26) using PPS.
PWM Timer: Timer 2 (T2) for the PWM frequency.

4. Which timers will you use to implement the 200 Hz position control ISR and the 5 kHz current control ISR? What priorities will you use?

5 kHz Current Control ISR: Timer 3 (T3) with a higher priority level 5 to ensure timely response for current control, which is critical for motor performance and safety.
200 Hz Position Control ISR: Timer 4 (T4) with an interrupt priority level set 3, since position control typically has less stringent timing requirements compared to current control.

5. Based on your answers to these questions, and your understanding of the project, annotate the block diagram of Figure 27.7. Each block should clearly indicate which devices or peripherals perform the operation in the block, and each signal line should clearly indicate how the signal is carried from one block to the other. (After this step, there should be no question about the hardware involved in the project. The details of wiring the H-bridge, current sensor, and encoder are left to later.)

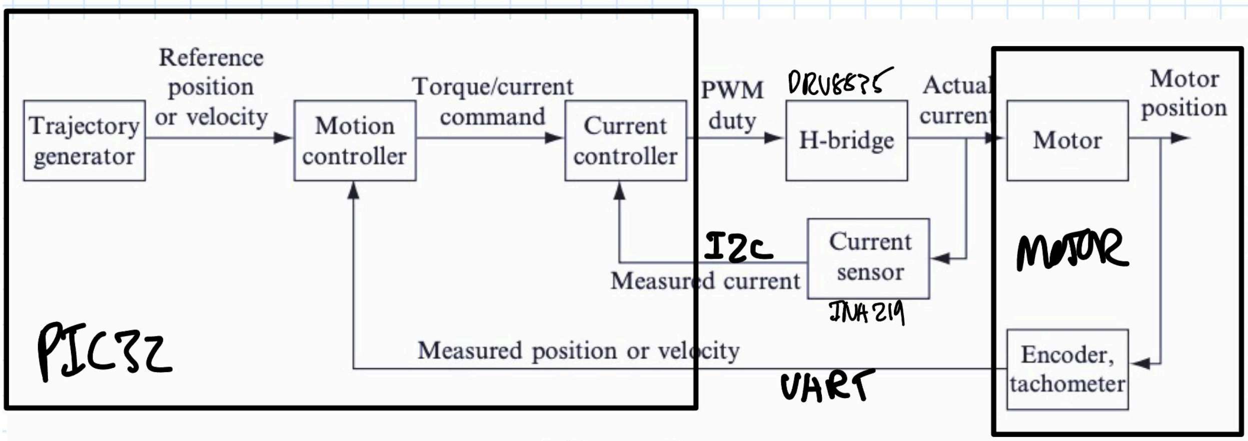
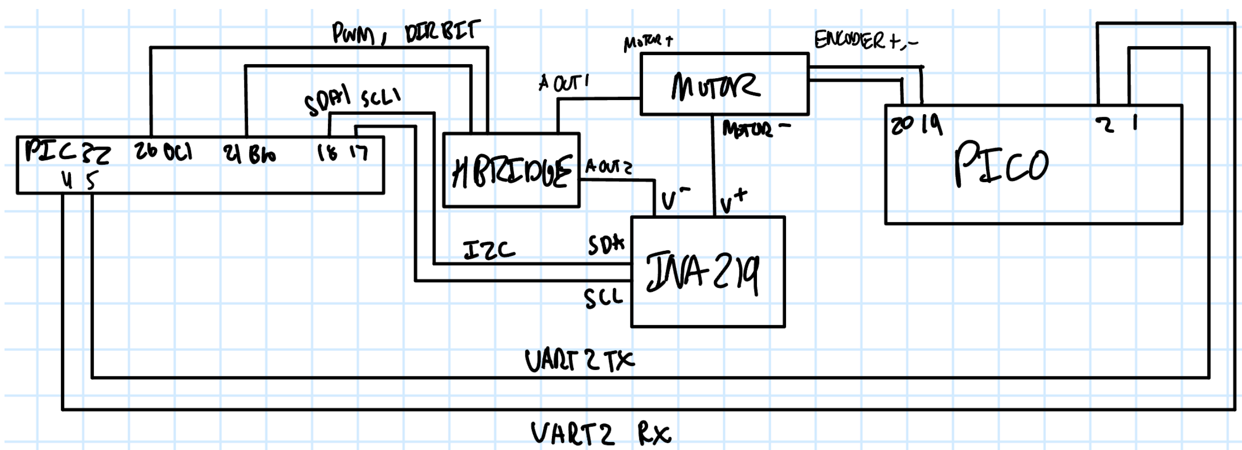


Figure 27.7
A block diagram for motion control.

6. Based on which circuit boards need to be connected to which pins of the NU32, and the connections of the circuit boards to the motor and encoder, sketch a proposed layout of the circuit boards relative to the NU32 so that wire crossing is approximately minimized. (Do not make a full circuit diagram at this time.)



28.4.7

2. Find the maximum current you expect to sense. If the H-bridge's battery voltage is V and the motor resistance is Rmotor, then the maximum current you can expect to see is approximately I_{max} = $2V/R_{motor}$.Record your calculated Imax for your battery and motor.

V = 6V
R_{motor} = 8.7 ohm (empirical measurement)
I_{max} = 2V/R = 12/8.7 = 1.379 A

3. Calculate the voltage across the 15 ohm sense resistor if I_{max} flows through it. Call this V_{max}.

Shunt resistor is R = 0.1 ohm.
$V_{max} = I_{max}R = 1.379 * 0.1 = 0.1379$ V

4. Choose resistors R1 and R2 so the current-sense amplifier gain G = 1 + (R2/R1) approximately satisfies 1.65 V = G × V_{max}. This ensures that the maximum positive motor current yields a 3.3 V output from the current sensor and the maximum negative motor current yields a 0 V output from the current sensor, utilizing the full range of the ADC input. Choose R1 and R2 to be in the range of 10^4-10^6 ohm.

The INA219 automatically sets the gain to measure the voltage across the shunt resistor accurately. You don't have to manually set resistors R1 and R2 to set the gain. Instead, you can use the INA219 calibration register to ensure the correct current measurement range.

5. Choose a resistor R and a capacitor C to make an RC filter on the MAX9918 output with a cutoff frequency fc = 1/(2πRC) in the neighborhood of 200 Hz, to suppress high-frequency components due to the 20 kHz PWM

The INA219 has digital filtering options, which means you don't need an external RC filter for the output.

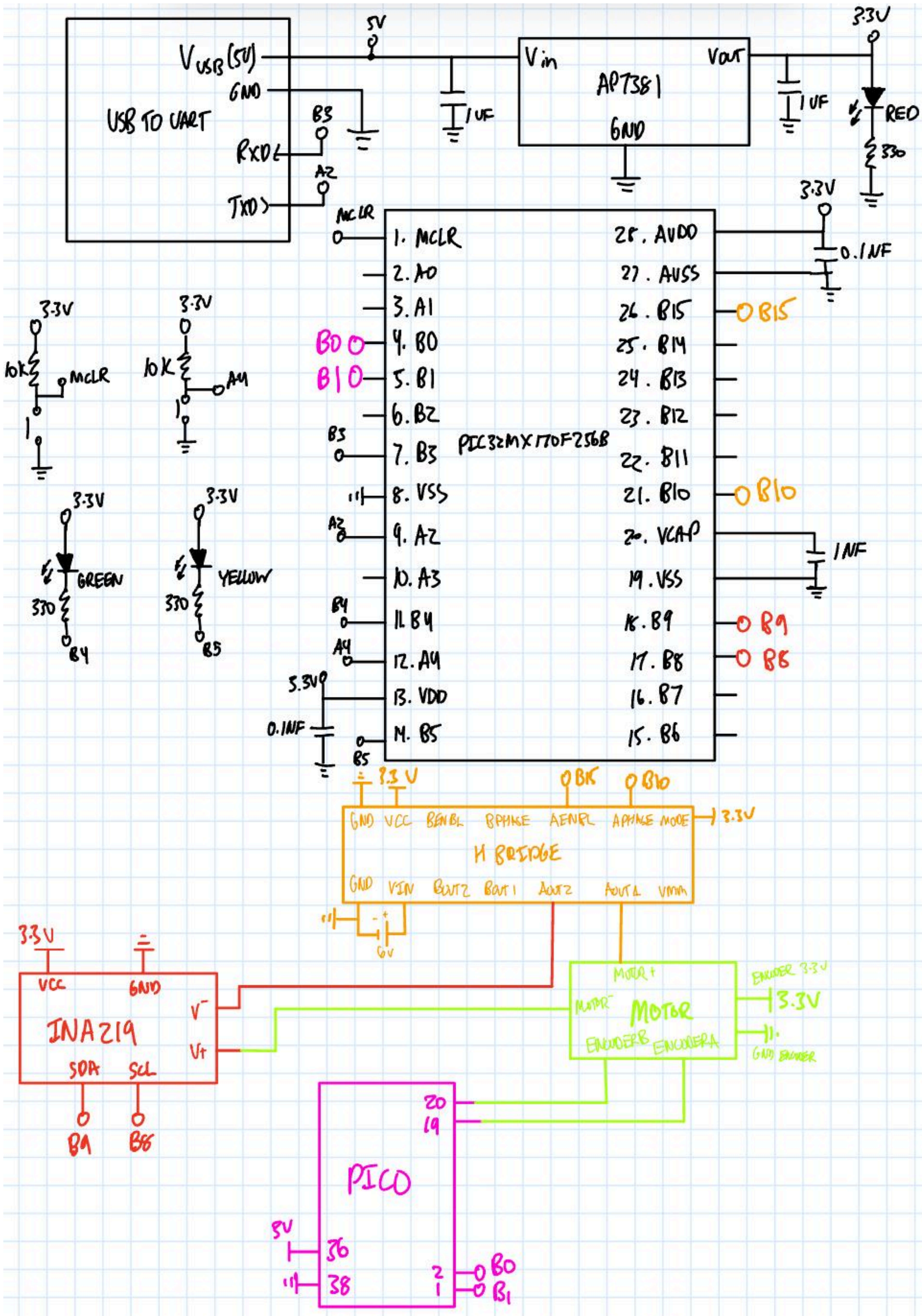
6. Calibrate using resistors.

The INA219 is calibrated in code here are the following lines:

```
// set the INA219 sensitivity - 10 bit, plus/minus160mV, 148us per sample
unsigned short ina219_calValue = 1024;
unsigned short ina219_config = 0b0011000010001111;
writeINA219(INA219_REG_CALIBRATION, ina219_calValue);
writeINA219(INA219_REG_CONFIG, ina219_config);
```

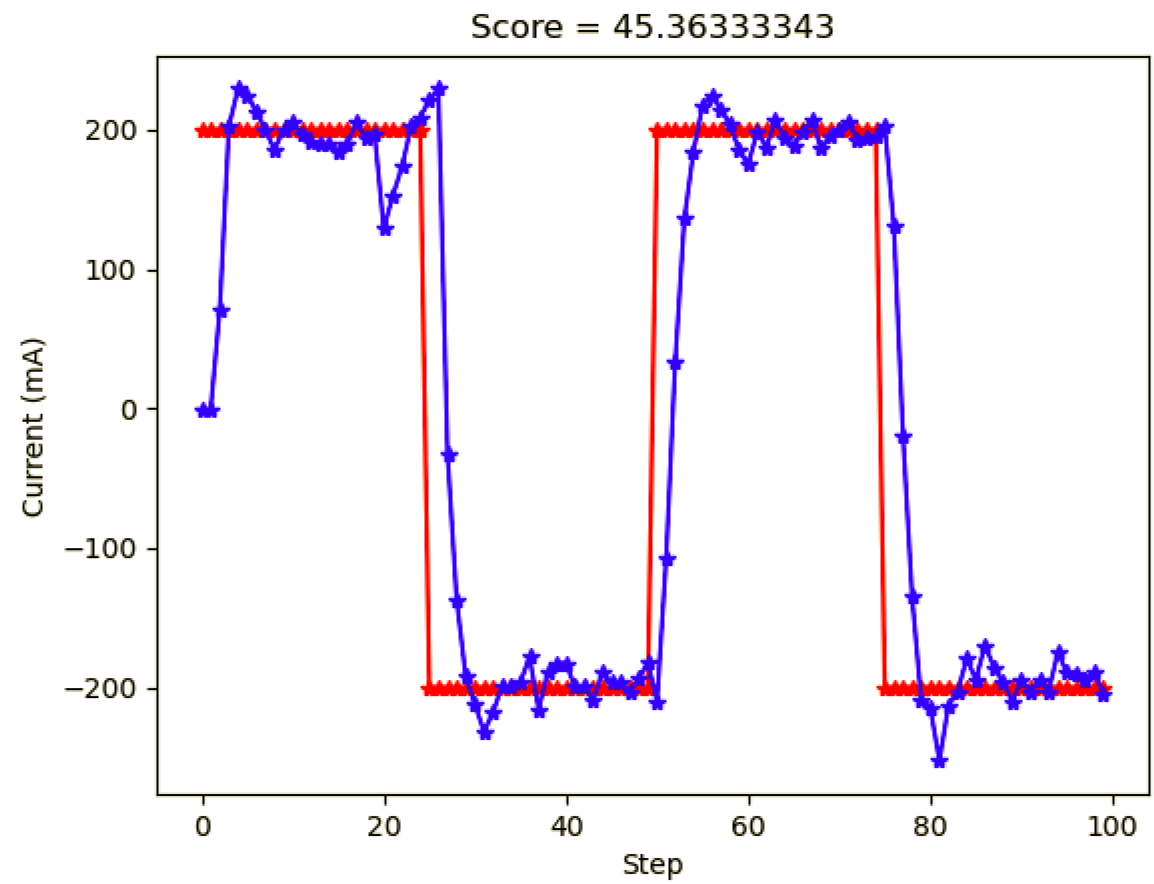
28.4.9

Turn in a circuit diagram showing all connections of the H-bridge to the NU32, motor, and current sensor PCB



USB TO UART
VUSB(5V)
GND
RXD
TXD
5V
Vin
APT381
GND
Vout
1UF
1UF
3.3V
RED
330
PIC32MX170F256B
1. MCLR
2. A0
3. A1
4. B0
5. B1
6. B2
7. B3
8. VSS
9. A2
10. A3
11. B4
12. A4
13. VDD
14. B5
28. AVDD
27. AVSS
26. B15
25. B14
24. B13
23. B12
22. B11
21. B10
20. VCAP
19. VSS
18. B9
17. B8
16. B7
15. B6
0.1NF
1NF
10K
GREEN
YELLOW
H BRIDGE
GND
VCC
BENBL
BPHASE
AENBL
APHASE
MODE
VIN
BOUT2
BOUT1
AOUT2
AOUT1
VMM
6V
INA219
VCC
GND
SDA
SCL
MOTOR
MOTOR+
MOTOR-
ENCODERB
ENCODERA
ENCODER 3.3V
GND ENCODER
PICO

28.4.10 #5




```
ENTER COMMAND: g
Set current gains (Jp, Ji): 0.33 0.06
Jp: 0.330000
Ji: 0.060000
PIC32 MOTOR DRIVER INTERFACE

ENTER COMMAND: k
Data lengeth = 100
```

Proportional Gain Current Jp = 0.33 [1/mA]

Integral Gain Current Ji = 0.06 [1/(mA s)]

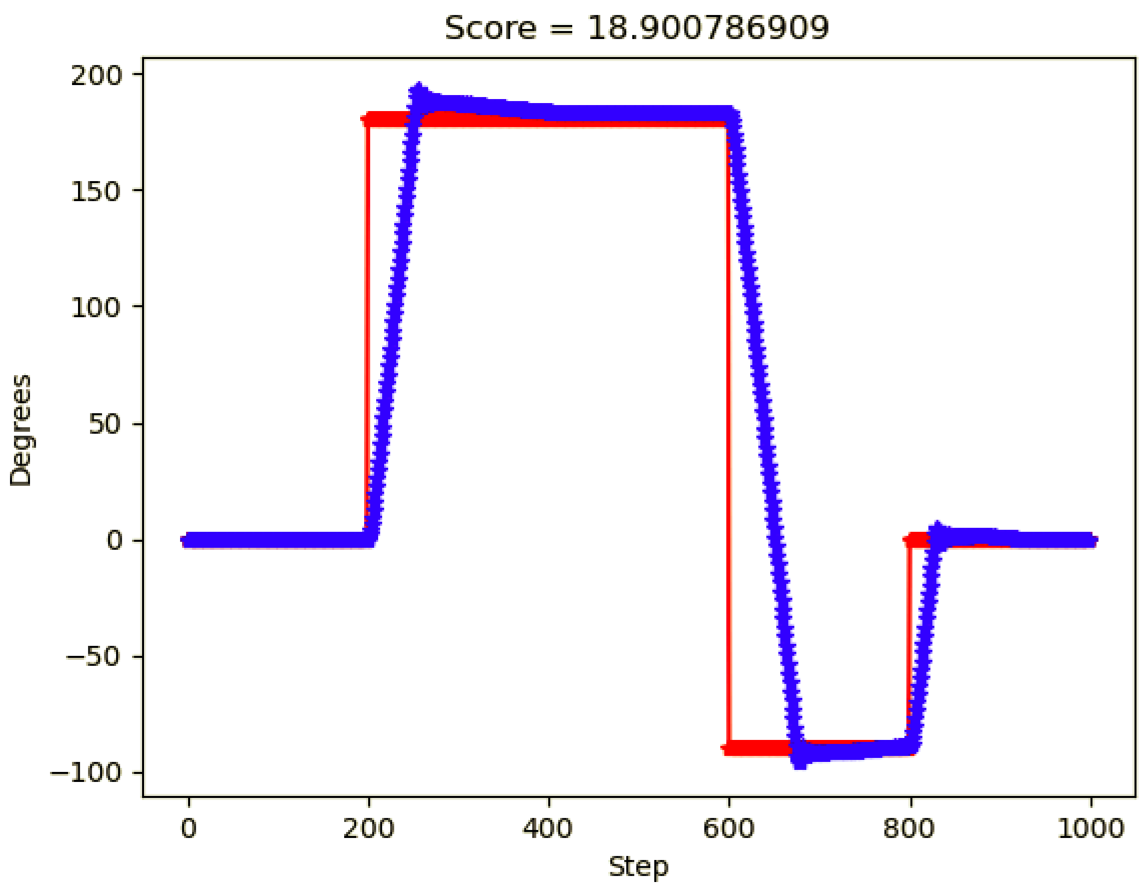
28.4.12 #5

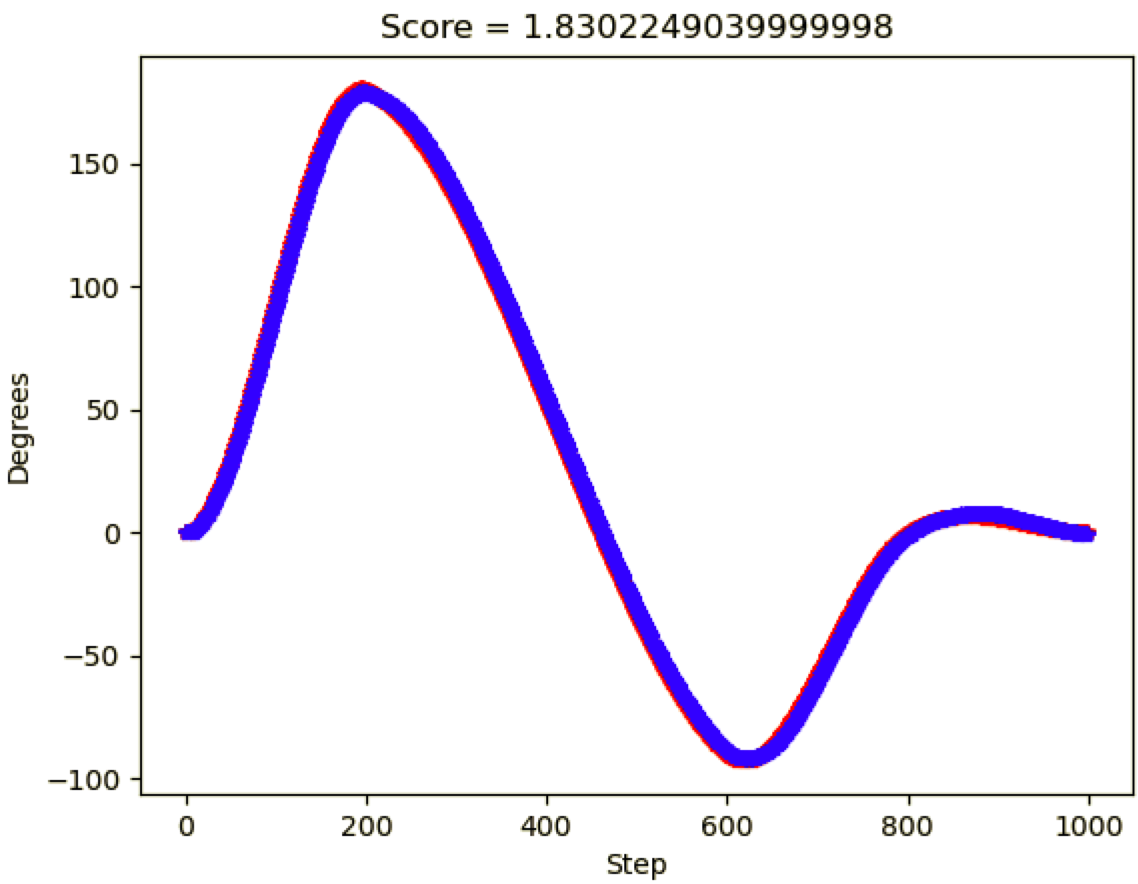
Kp = 0.090000 [mA/count]

Ki = 0.150000 [mA/(count*s)]

Kd = 0.650000 [mA*s/count]

```
(ME333) 1:CH28 christopherluey$ python ch28.py
Opening port:
/dev/tty.SLAB_USBtoUART
PIC32 MOTOR DRIVER INTERFACE

ENTER COMMAND: g
Set current gains (Jp, Ji): .33 .06
Jp: 0.330000
Ji: 0.060000

ENTER COMMAND: i
Set position gains (Kp, Ki, Kd): 0.090000 0.15000 0.650000
Kp: 0.090000
Ki: 0.150000
Kd: 0.650000

ENTER COMMAND: m
Step!
Enter times and angles, starting at t=0 (ex: 0 0 1 90 2 90 3 0): 0 0 1 180 3 -90 4 0 5 0

ENTER COMMAND: o
Data lengeth = 1000
```






```
(ME333) 1:CH28 christopherluey$ python ch28.py
Opening port:
/dev/tty.SLAB_USBtoUART
PIC32 MOTOR DRIVER INTERFACE

ENTER COMMAND: g
Set current gains (Jp, Ji): .33 .06
Jp: 0.330000
Ji: 0.060000

ENTER COMMAND: i
Set position gains (Kp, Ki, Kd): 0.090000 0.15000 0.650000
Kp: 0.090000
Ki: 0.150000
Kd: 0.650000

ENTER COMMAND: n
Cubic!
Enter times and angles, starting at t=0 (ex: 0 0 1 90 2 90 3 0): 0 0 1 180 3 -90 4 0 5 0

ENTER COMMAND: o
Data lengeth = 1000
```