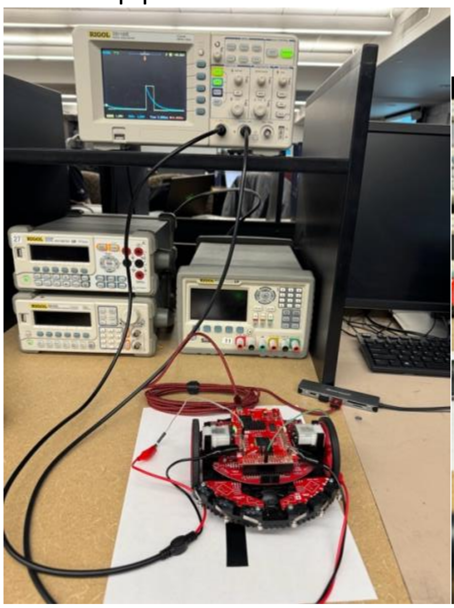
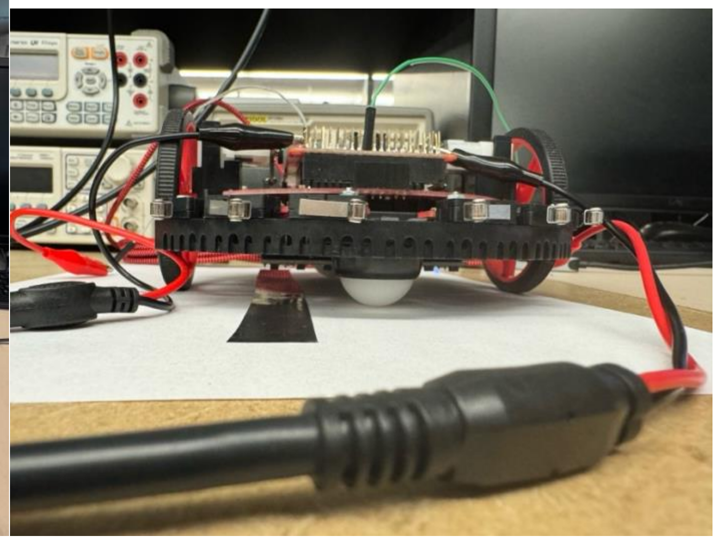
Rick Brophy
ECE1188 Cyberphys
Dr. Dickerson
Due: 2/16/24

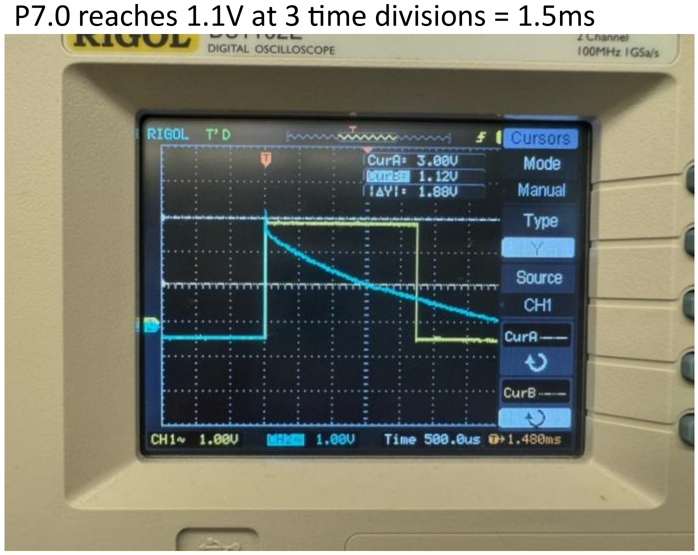
Lab 2 – Reflectance Sensing

1. Set up pictures

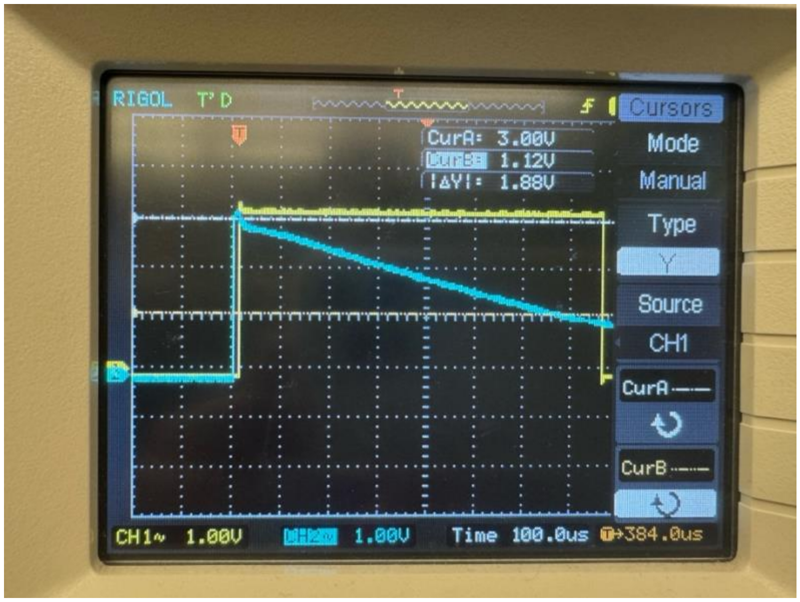




   a. BLACK MATERIAL: 3.0V * 0.368 = 1.1V – <u>This scope did not like having both the x and y cursors on simultaneously, I hope using the divisions is fine.</u>
      i. P7.0 reaches 1.1V at 3 time divisions = 1.5ms



   b. WHITE MATERIAL: 3.0V * 0.368 = 1.1V
      i. P7.0 reaches 1.1V at 6.75 time divisions = 0.675ms



c. Questions

i. A long explanation (sorry) of phototransistor & RCs

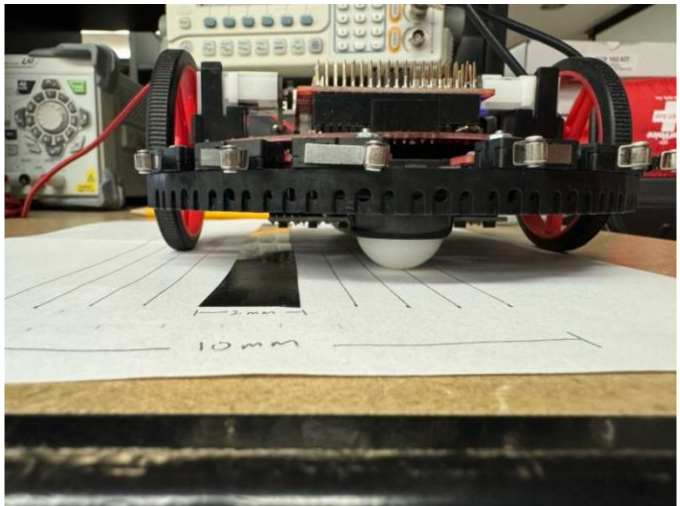
From previous projects, I learned that most phototransistor collector currents are modeled to be linearly dependent on the amount of measured luminous flux. From a schematic standpoint, one can infer that the color of the material reflected on changes the effective impedance ($R_{DS,on}$) seen by the capacitor across the transistor. As more light is reflected (white paper), the phototransistor allows more current to flow, due to a small $R_{DS,on}$, creating a smaller time constant ($\tau = RC$) for the capacitor to discharge. Note that $R_{DS,on}$ and the 220Ω resistor also form a voltage divider. As less light is reflected (black tape), the phototransistor allows small amounts of current to flow, due to a large $R_{DS,on}$, creating a large time constant for the capacitor to discharge. The smaller the time constant, the less amount of time it takes to completely discharge the cap ($\tau \sim 4RC$).

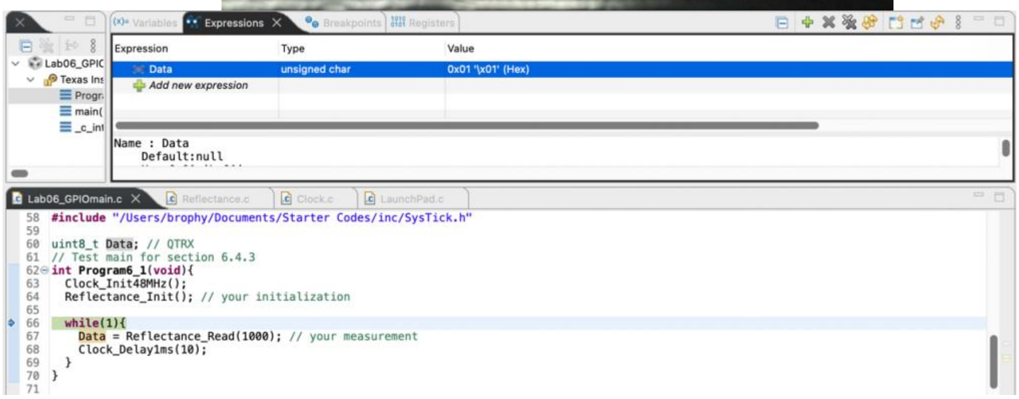
ii. BLACK MATERIAL
1. The microcontroller interprets a voltage of [0, 0.76)V as a binary 0
2. The microcontroller interprets a voltage of [3, 0.76)V as a binary 1

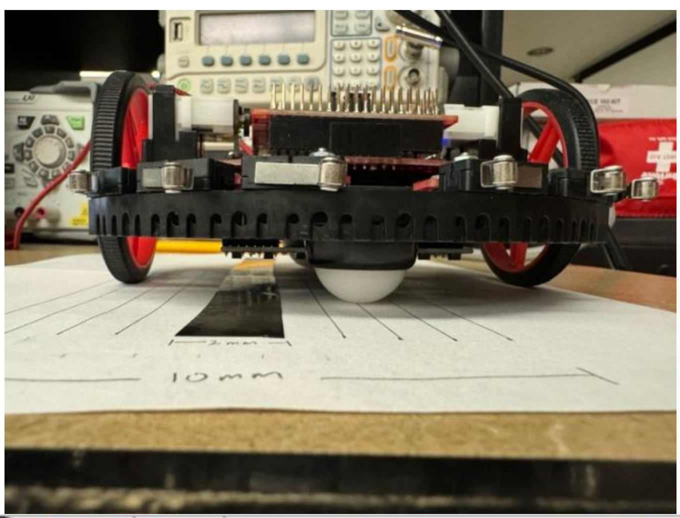
iii. WHITE MATERIAL
1. The microcontroller interprets a voltage of [0, 0.96)V as a binary 0
2. The microcontroller interprets a voltage of [3, 0.96)V as a binary 1

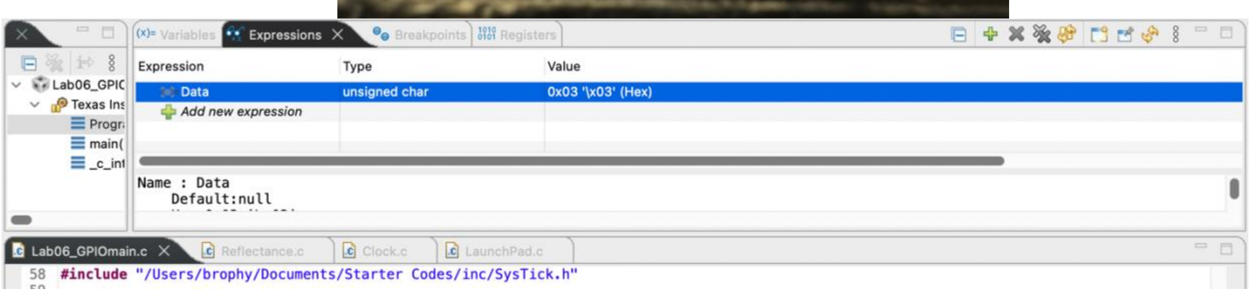
iv. I am sad they do not match.

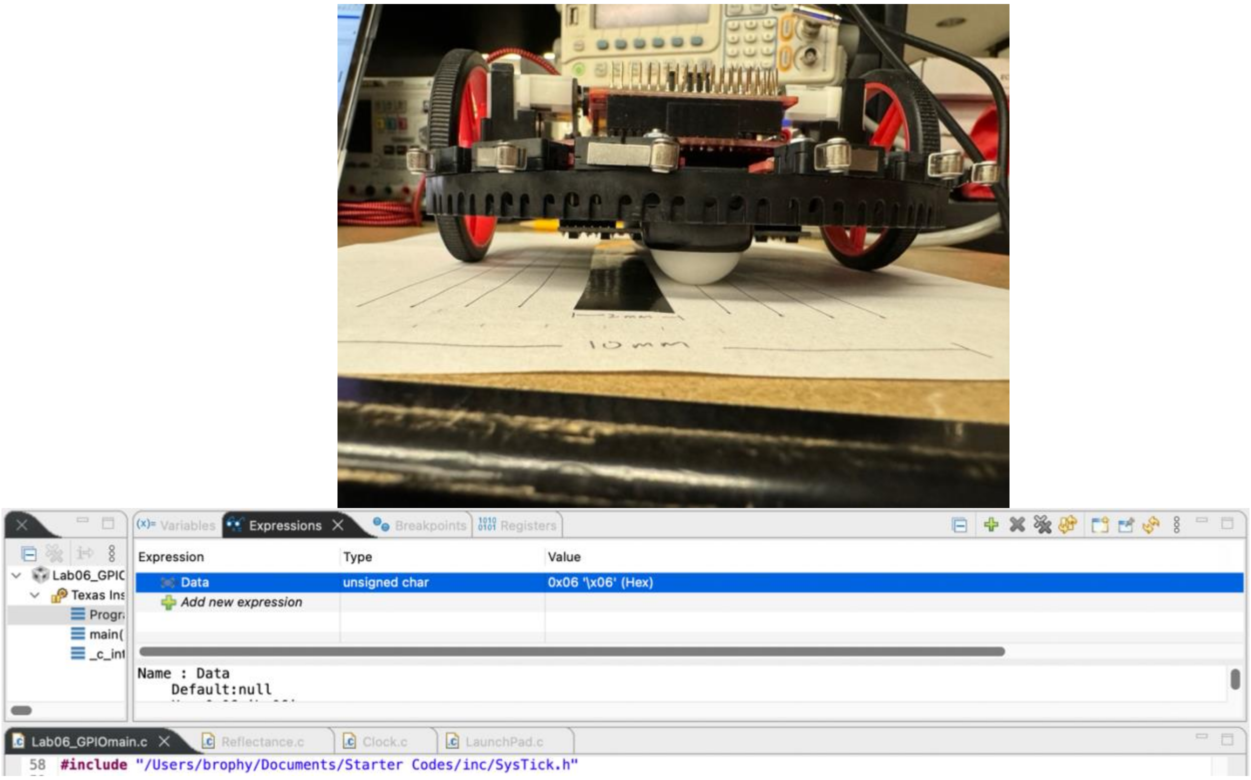
2. Position changes and memory
   a. I used 10mm intervals with ~2mm black tape, starting with the right most IR LED (P7.0)
   b. The first picture only has P7.0 reflecting with shifting in the next IR until the last picture, which only has the left most IR LED, P7.7

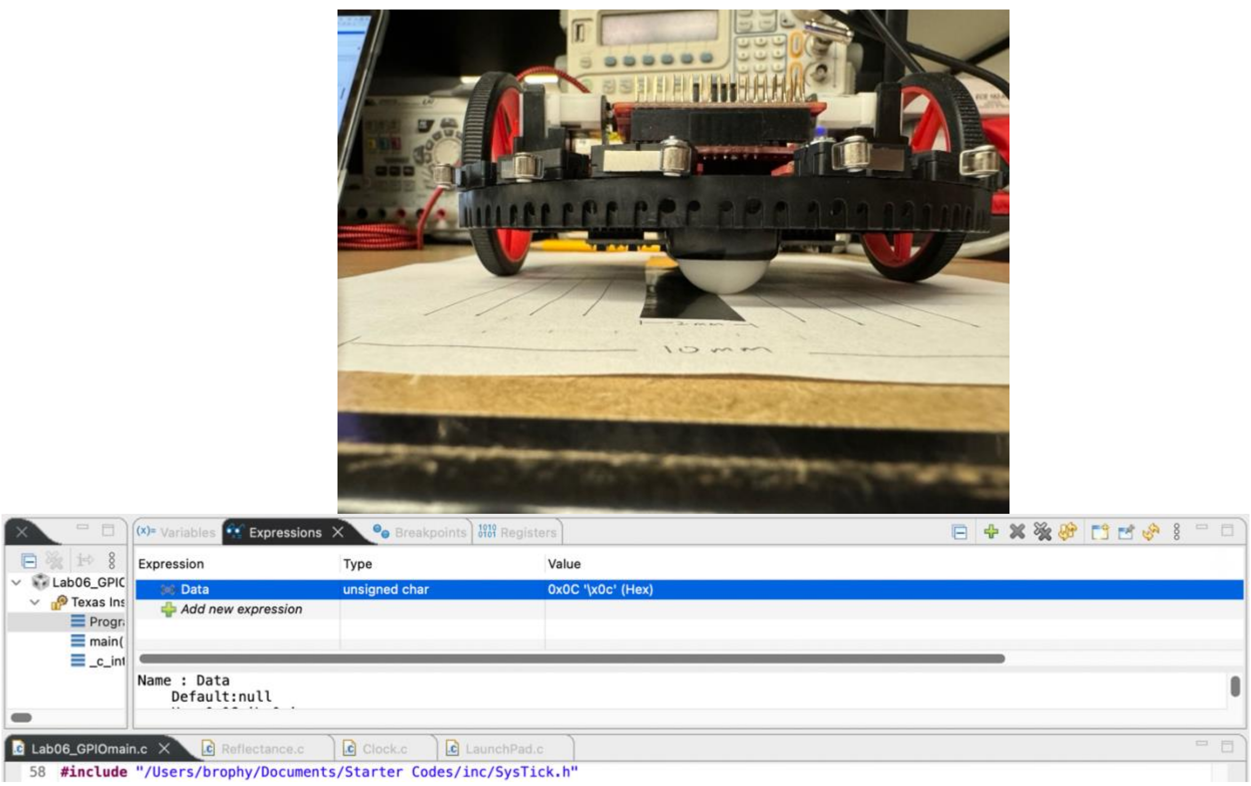


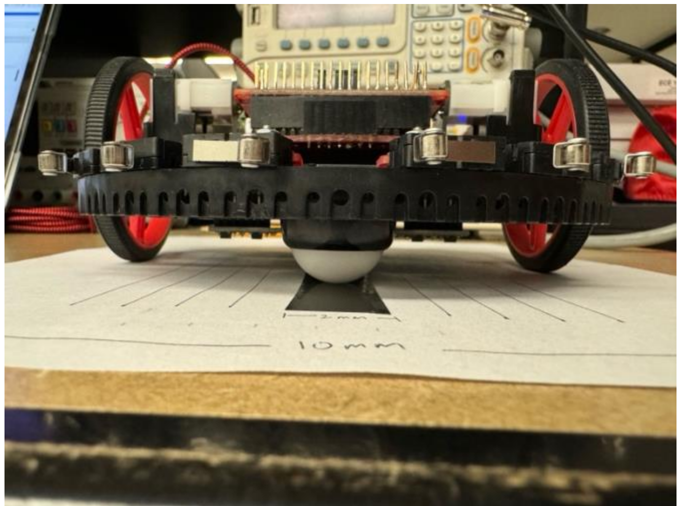

Variables
Expressions
Breakpoints
Registers
Expression
Type
Value
Data
unsigned char
0x01 '\x01' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
#include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"
uint8_t Data; // QTRX
// Test main for section 6.4.3
int Program6_1(void){
Clock_Init48MHz();
Reflectance_Init(); // your initialization
while(1){
Data = Reflectance_Read(1000); // your measurement
Clock_Delay1ms(10);
}
}

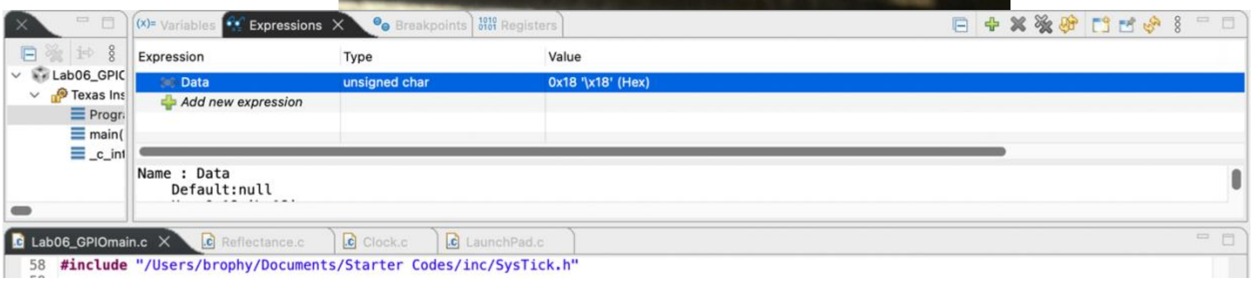


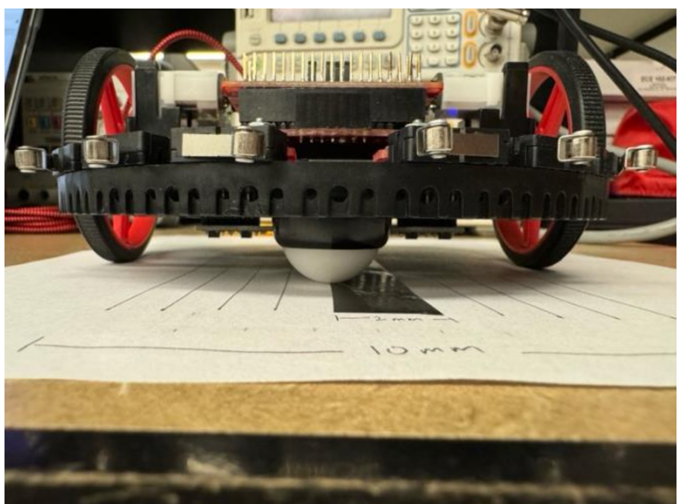

Variables
Expressions
Breakpoints
Registers
Expression
Type
Value
Data
unsigned char
0x03 '\x03' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
#include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"

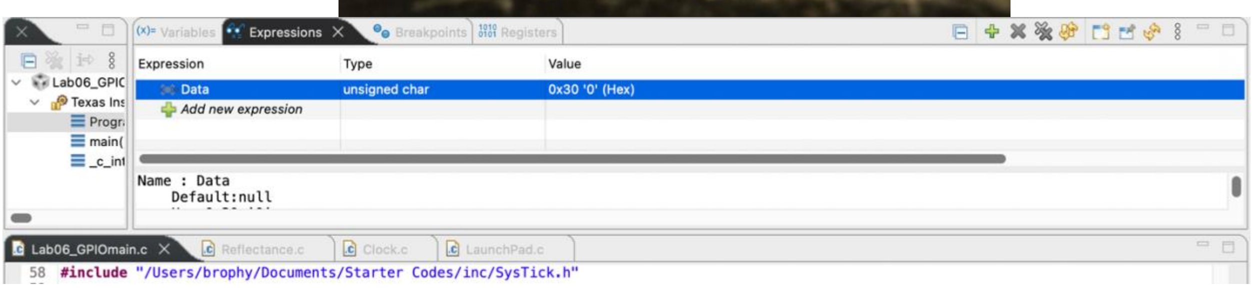

10 mm
Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Progr
main(
_c_int
Expression
Type
Value
Data
unsigned char
0x06 '\x06' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"

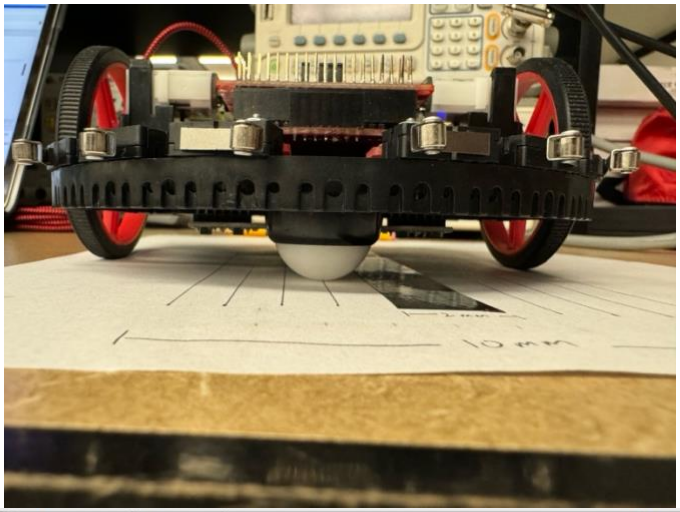

10 mm
Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Progr
main(
_c_int
Expression
Type
Value
Data
unsigned char
0x0C '\x0c' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"

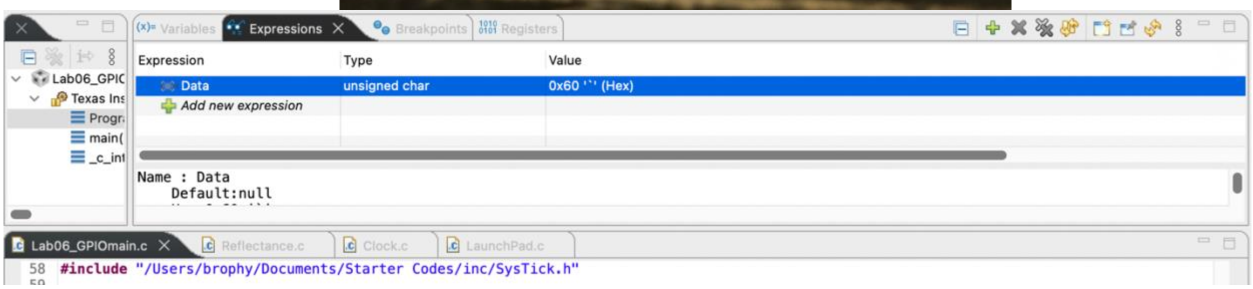

10 mm

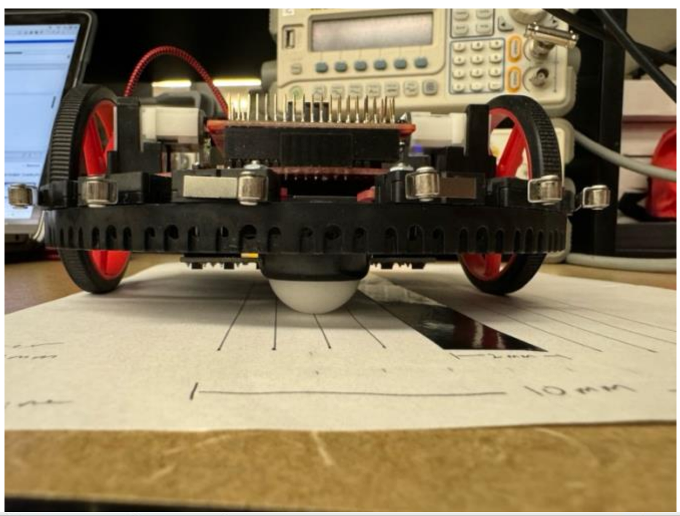

Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Expression
Type
Value
Data
unsigned char
0x18 '\x18' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"

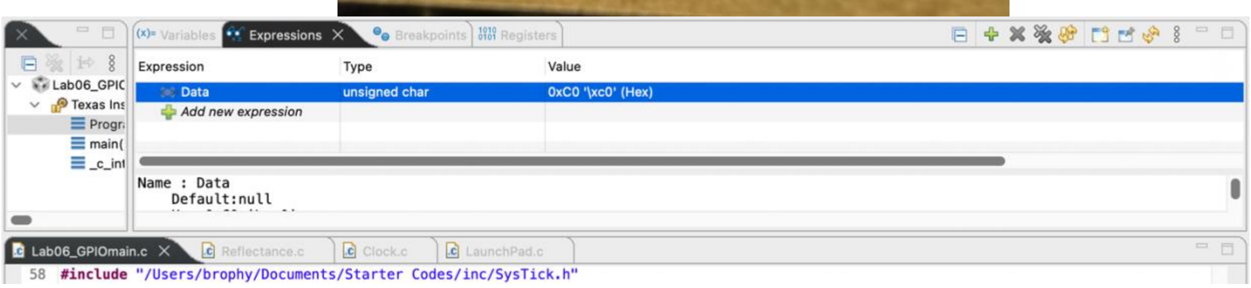

10 mm

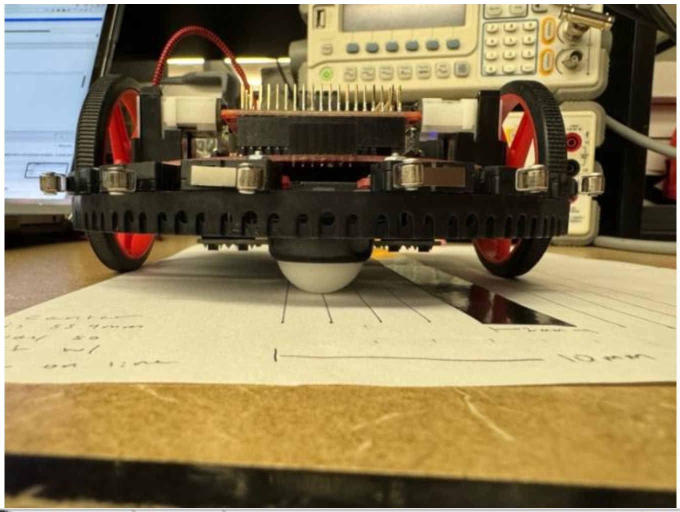

Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Expression
Type
Value
Data
unsigned char
0x30 '0' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"

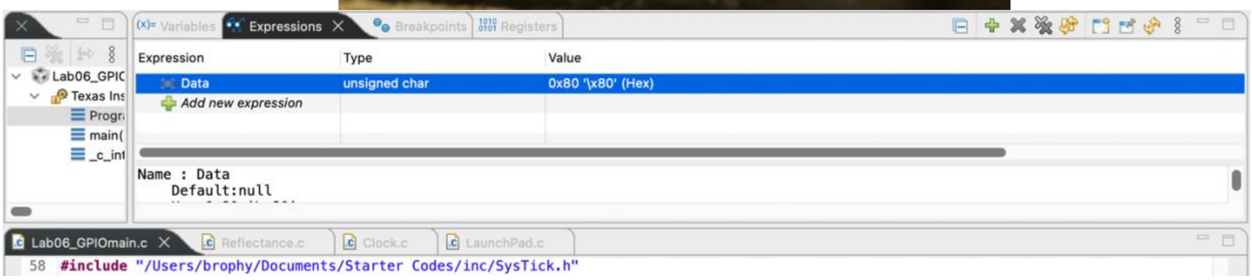



Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Progr
main(
_c_int
Expression
Type
Value
Data
unsigned char
0x60 '`' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"



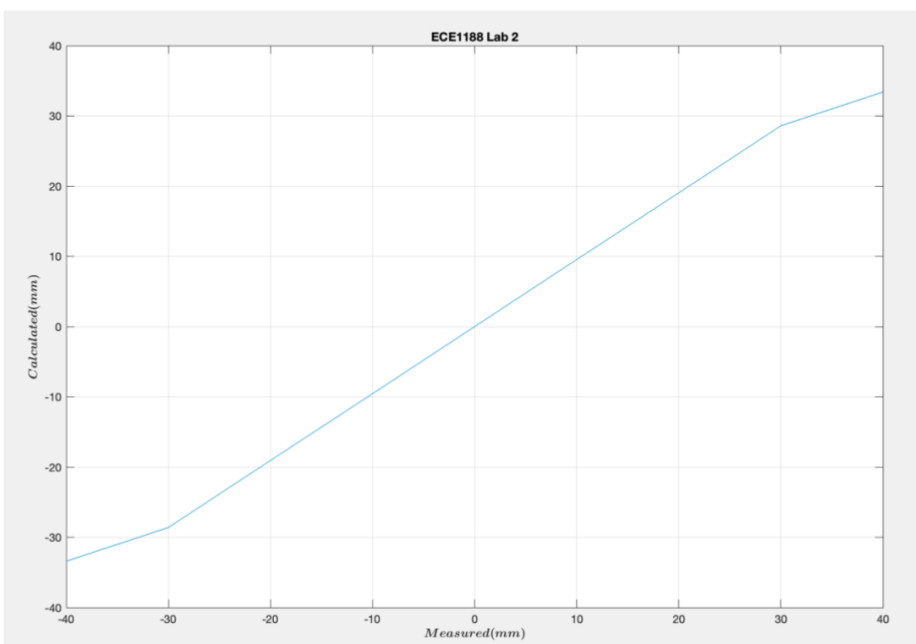

Variables
Expressions
Breakpoints
Registers
Lab06_GPIC
Texas Ins
Progr
main(
_c_int
Expression
Type
Value
Data
unsigned char
0xC0 '\xc0' (Hex)
Add new expression
Name : Data
Default:null
Lab06_GPIOmain.c
Reflectance.c
Clock.c
LaunchPad.c
58 #include "/Users/brophy/Documents/Starter Codes/inc/SysTick.h"



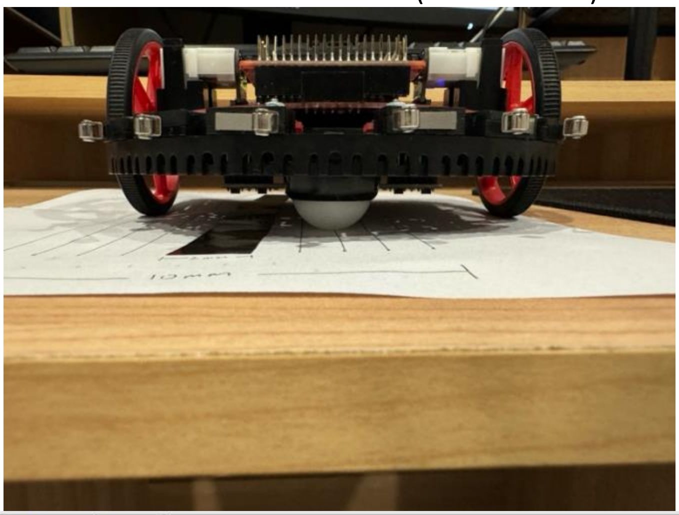


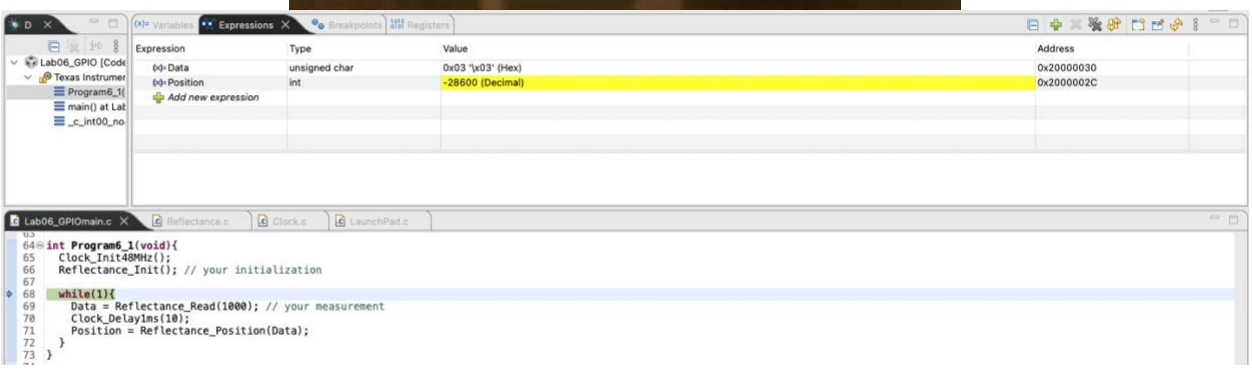
3. Reflectance Position – all ruler measurements were taken from the center of the black line to the center of the robot
   a. Plot of measured vs. calculated positions

```
x = [-40, -30, -20, -10, 0, 10, 20, 30, 40];
y = [-33.4, -28.6, -19.05, -9.55, 0, 9.55, 19.05, 28.6, 33.4];
```



b. Picture of measured vs calculated data (one instance)





c. Questions

i. As you can see from the graph, there are two separate linear relationships between the measured and calculated position data. The end regions show where only 1 IR LED was sensing reflections along the black line and created a less accurate reading. The middle region shows where two IR LEDs were sensing reflections on the black line and had more data to calculate the position. The point-slope form equations to mathematically describe these regions are {y = 0.48x $\pm$ 14.2, y = 0.95x} for the end (negative for IR LED1 and positive for IR LED 2) and middle regions respectively. This relationship shows that as more IR LEDs are present on the black line, the more accurate the calculation is.

ii. The defition of monotonicity is that the value of a function is increasing or decreasing for the entirety of its domain. In this instance, the values read from Reflectance_Position() are monotonic. This is important because this allows us to build trustworthy models of robot position based on our data. We are 100% sure that when the robot is getting further away from the black line, the absolute value of the Reflectance_Position() output will always be increasing. This will come in handy when we potentially implement comparison algorithms later in the course.