Owen Richards
CS3501: Embedded Circuits
9/18/2021

Verification for Reflectance Program

Table of Content

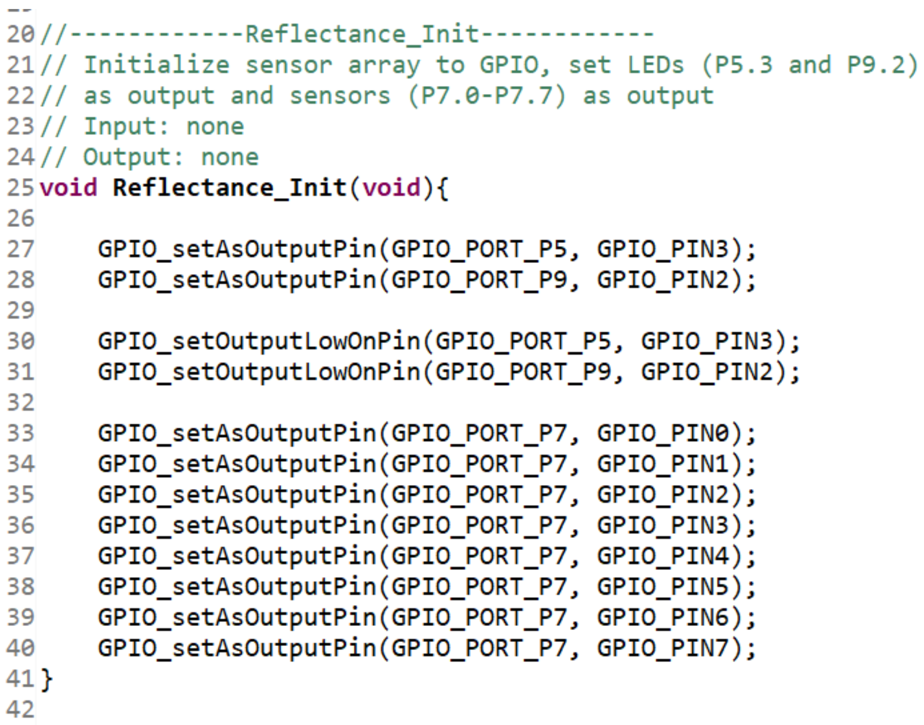
Table of Figures

Introduction

In this module, I was tasked with programming the sensor array so that it is able to identify the difference between a black surface and a white surface. To program the sensor array, four functions were written to support the line sensor. These four functions is the Reflectance_Init(), Reflectance_Read(uint16_t time), Reflectance_Center(uint32_t time**)** and Reflectance_Position(). Throughout this report figure and verification will be give to show that all these functions were correctly implemented and ultimately that the task was completed.

Verification

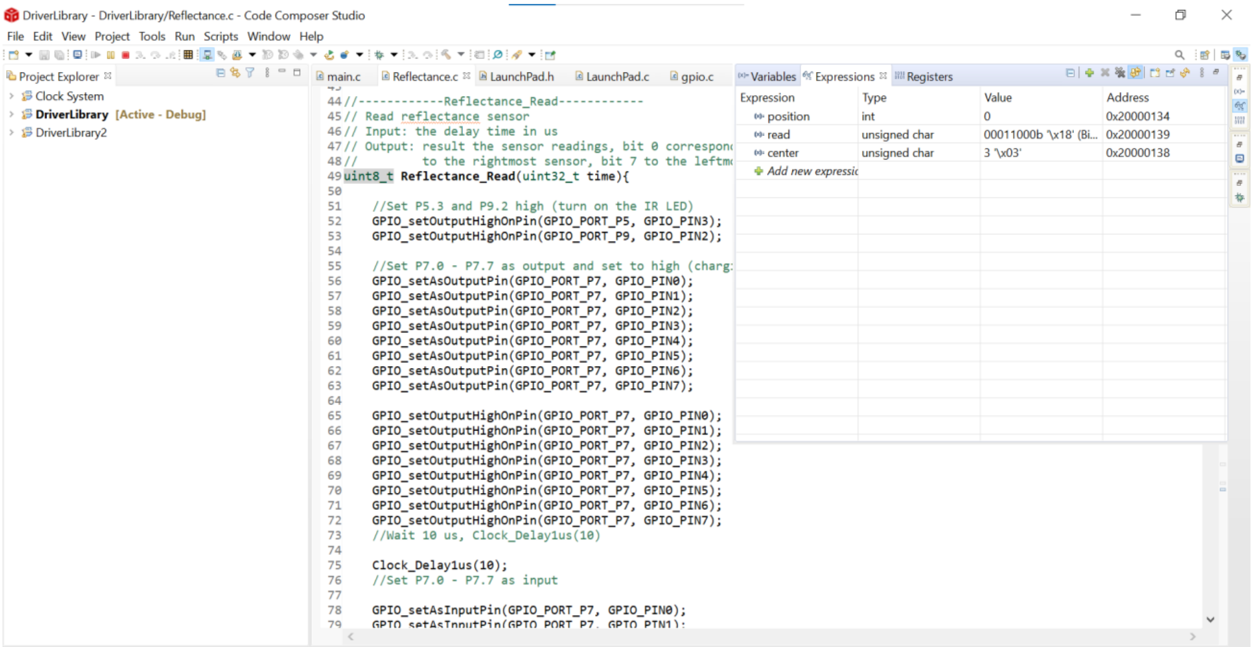
The first function that was implemented to create this program was the Reflectance_Init() function. The Reflectance_Init() function is shown in Figure 1 below. However, this function doesn't have any tests to prove it was correctly programmed. The proof that this function was correctly implemented is in the other functions producing the correct expressions. Despite this, I will work though the code shown in figure 1 to display it is correct. Line 27 and 28 initialize sensor array to GPIO, then line 30 and 31 sets the LEDs for port 5 pin 3, and port 9 pin 2. Lines 33 - 40 sets sensors (port 7 and pins 0-7) as outputs.



```
20//------------Reflectance_Init------------
21// Initialize sensor array to GPIO, set LEDs (P5.3 and P9.2)
22// as output and sensors (P7.0-P7.7) as output
23// Input: none
24// Output: none
25void Reflectance_Init(void){
26
27    GPIO_setAsOutputPin(GPIO_PORT_P5, GPIO_PIN3);
28    GPIO_setAsOutputPin(GPIO_PORT_P9, GPIO_PIN2);
29
30    GPIO_setOutputLowOnPin(GPIO_PORT_P5, GPIO_PIN3);
31    GPIO_setOutputLowOnPin(GPIO_PORT_P9, GPIO_PIN2);
32
33    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN0);
34    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN1);
35    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN2);
36    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN3);
37    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN4);
38    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN5);
39    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN6);
40    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN7);
41}
42
```

Figure 1: Relectance_Init function

The expressions and placement of the sensor array is shown below in Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, Figure 7. The figures illustrate the correct read expressions for the center, left and right placement of the sensor array. The read sensor will turn the bits 1 for the sensors above the black line, otherwise, the bits will be 0. The figures below verify the function since the correct bits are 1 and 0 depending on the position of the sensor array. Therefore, the Reflectance_Read function was correctly implemented. In addition, Figures 2 and 3 highlight that both the position and center expression is correct meaning that both the Reflectance_Center(uint32_t time**)** and Reflectance_Position() functions were completed correctly. Both these functions will be verified later in this report.


DriverLibrary - DriverLibrary/Reflectance.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Project Explorer
Clock System
DriverLibrary [Active - Debug]
DriverLibrary2
main.c
Reflectance.c
LaunchPad.h
LaunchPad.c
gpio.c
Variables
Expressions
Registers

```
//------------Reflectance_Read------------
// Read reflectance sensor
// Input: the delay time in us
// Output: result the sensor readings, bit 0 correspon
//          to the rightmost sensor, bit 7 to the leftm
uint8_t Reflectance_Read(uint32_t time){

    //Set P5.3 and P9.2 high (turn on the IR LED)
    GPIO_setOutputHighOnPin(GPIO_PORT_P5, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P9, GPIO_PIN2);

    //Set P7.0 - P7.7 as output and set to high (charg:
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN7);

    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN7);
    //Wait 10 us, Clock_Delay1us(10)

    Clock_Delay1us(10);
    //Set P7.0 - P7.7 as input

    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN1);
```

Expression	Type	Value	Address
position	int	0	0x20000134
read	unsigned char	00011000b '\x18' (Bi...	0x20000139
center	unsigned char	3 '\x03'	0x20000138
Add new expressic			

Figure 2: Read Expression for Center Sensor Array

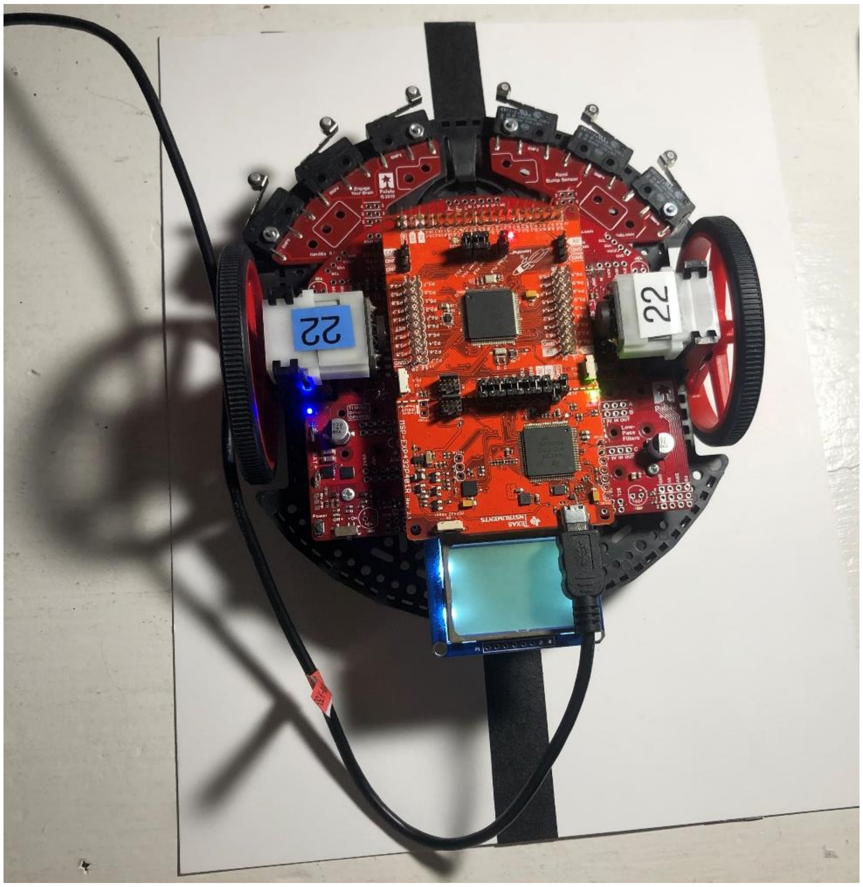
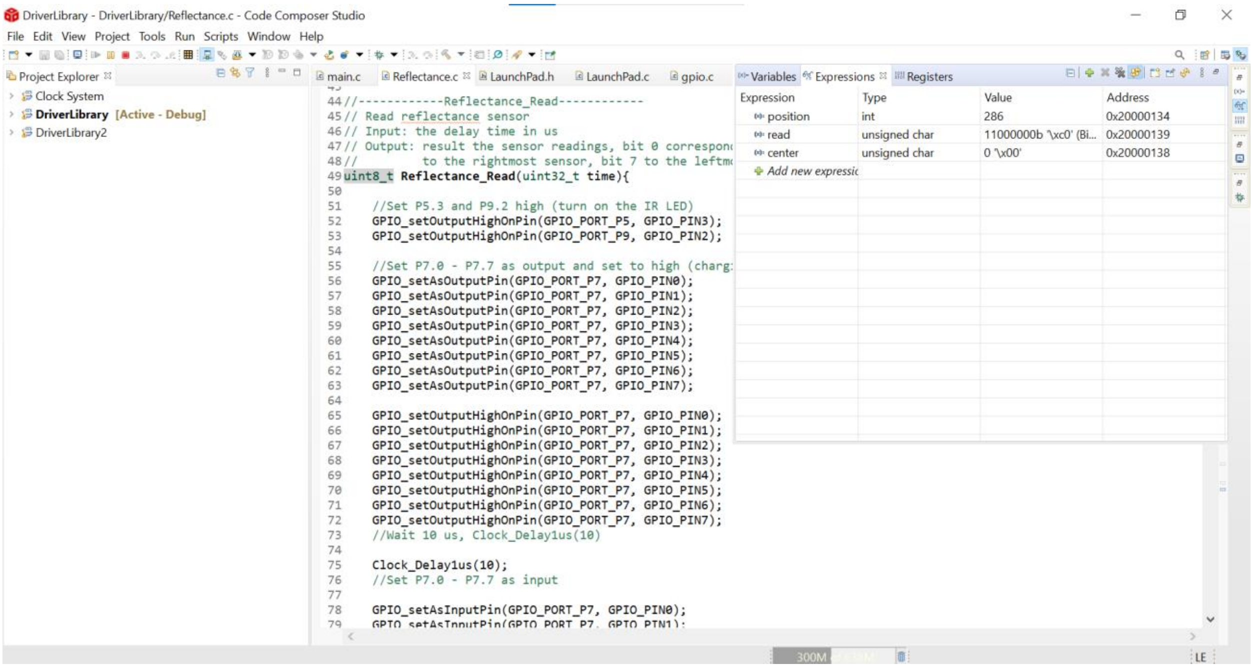


Figure 3: Sensor Array Centered



DriverLibrary - DriverLibrary/Reflectance.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Project Explorer
Clock System
DriverLibrary [Active - Debug]
DriverLibrary2
main.c
Reflectance.c
LaunchPad.h
LaunchPad.c
gpio.c
Variables
Expressions
Registers
300M
LE

```
//------------Reflectance_Read------------
// Read reflectance sensor
// Input: the delay time in us
// Output: result the sensor readings, bit 0 correspon
//         to the rightmost sensor, bit 7 to the leftm
uint8_t Reflectance_Read(uint32_t time){

    //Set P5.3 and P9.2 high (turn on the IR LED)
    GPIO_setOutputHighOnPin(GPIO_PORT_P5, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P9, GPIO_PIN2);

    //Set P7.0 - P7.7 as output and set to high (charg:
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN7);

    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN7);
    //Wait 10 us, Clock_Delay1us(10)

    Clock_Delay1us(10);
    //Set P7.0 - P7.7 as input

    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN1);
```

Expression	Type	Value	Address
position	int	286	0x20000134
read	unsigned char	11000000b '\xc0' (Bi...	0x20000139
center	unsigned char	0 '\x00'	0x20000138
Add new expressic			

Sensor 4: Read Expression for Left Sensor Array

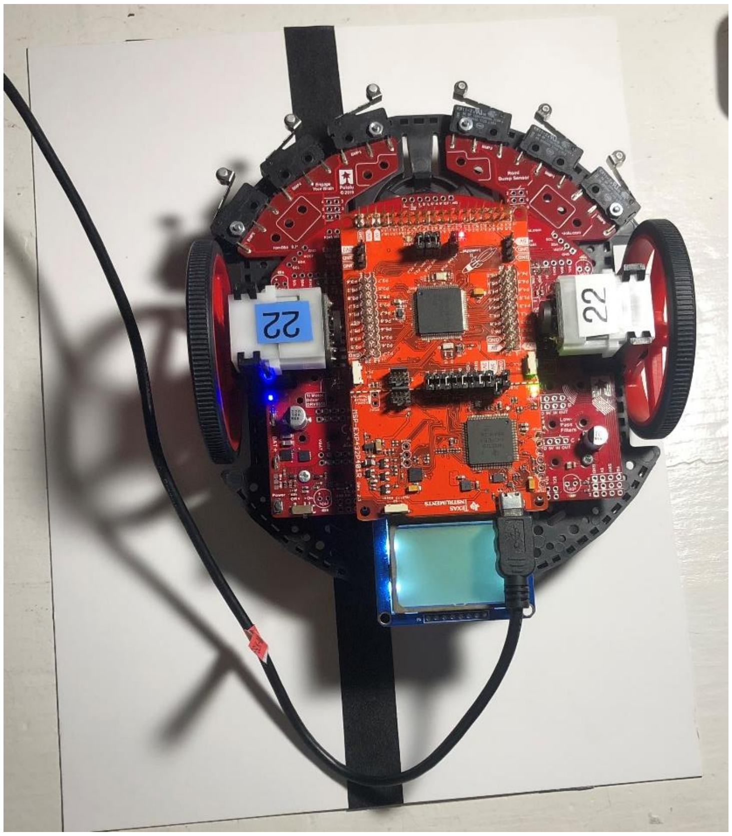
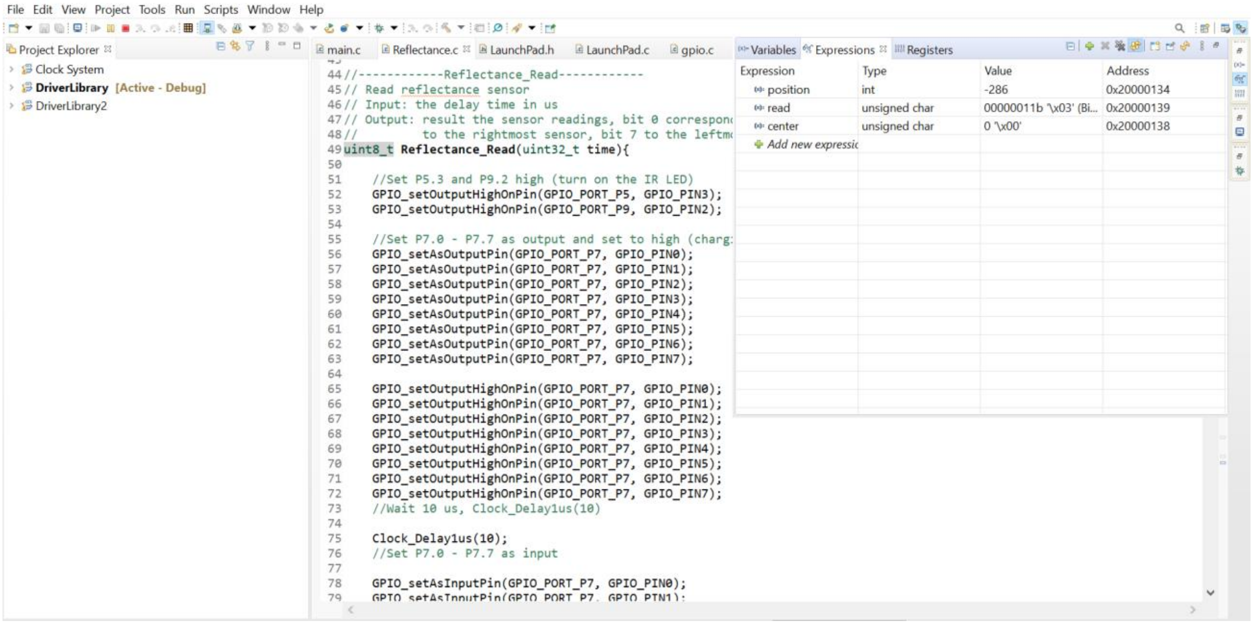


Figure 5: Sensor Array to the Left



```c
//------------Reflectance_Read------------
// Read reflectance sensor
// Input: the delay time in us
// Output: result the sensor readings, bit 0 correspon
//         to the rightmost sensor, bit 7 to the leftm
uint8_t Reflectance_Read(uint32_t time){

    //Set P5.3 and P9.2 high (turn on the IR LED)
    GPIO_setOutputHighOnPin(GPIO_PORT_P5, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P9, GPIO_PIN2);

    //Set P7.0 - P7.7 as output and set to high (charg:
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setAsOutputPin(GPIO_PORT_P7, GPIO_PIN7);

    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN1);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN2);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN3);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN4);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN5);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN6);
    GPIO_setOutputHighOnPin(GPIO_PORT_P7, GPIO_PIN7);
    //Wait 10 us, Clock_Delay1us(10)

    Clock_Delay1us(10);
    //Set P7.0 - P7.7 as input

    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN0);
    GPIO_setAsInputPin(GPIO_PORT_P7, GPIO_PIN1);
```

Expression	Type	Value	Address
position	int	-286	0x20000134
read	unsigned char	00000011b '\x03' (Bi...	0x20000139
center	unsigned char	0 '\x00'	0x20000138
Add new expressic			

Figure 6: Read Expression for Right Sensor Array

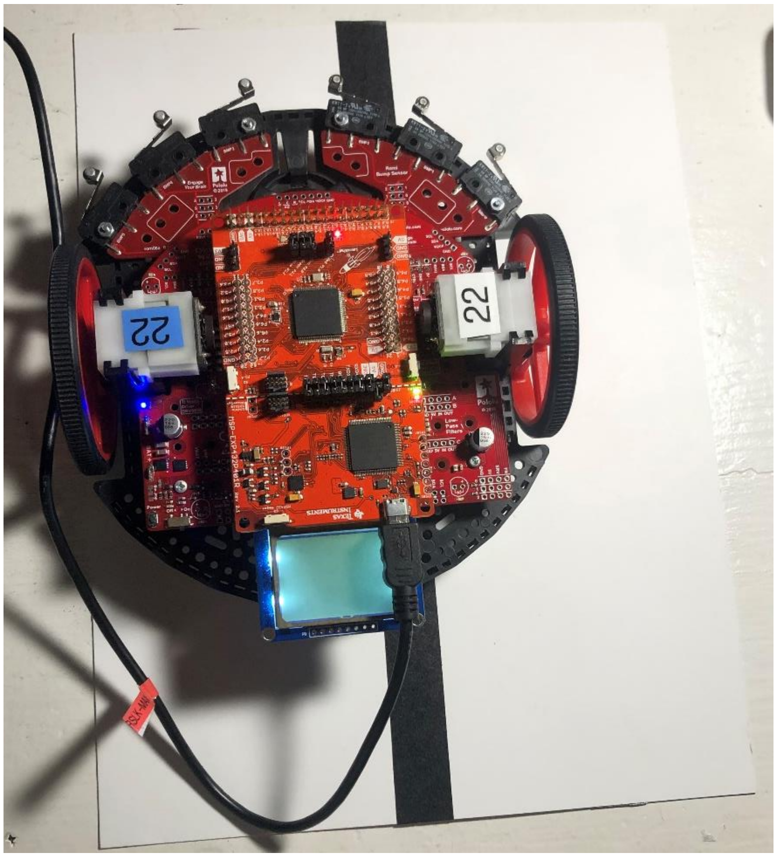


Figure 7: Sensor Array to the Right

To verify that both the Reflectance_Center(uint32_t time**)** and Reflectance_Position() are correct, Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16 and Figure 17 shows more verification of these functions. Figure 8, 9, 10 and 11 illustrate the Reflectance_Center(uint32_t time**)** function. I programmed this function so that if the robot is left of the line the expression will display a 2-meaning left. Additional, if the robot is right of the line the expression will display a 1 meaning right. Figures 12, 13, 14 and 15 show that if only the end sensor on the sensor array, the expression value is either -334 or 334 which demonstrates that the function is correctly implemented. The final figures 16 and 17 displays the position value if the sensor is completely off.

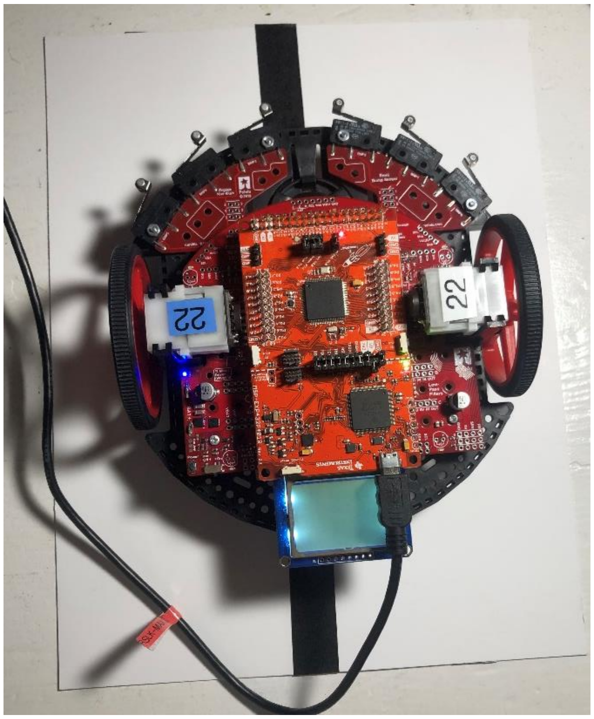
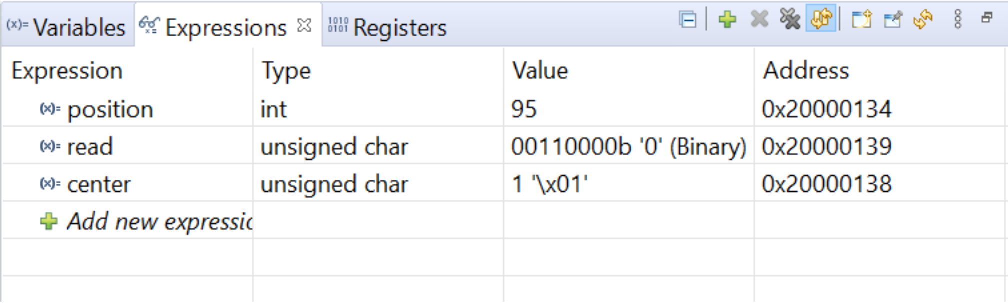


Figure 8: Center Left of Line



Expression	Type	Value	Address
position	int	95	0x20000134
read	unsigned char	00110000b '0' (Binary)	0x20000139
center	unsigned char	1 '\x01'	0x20000138
Add new expressic			

Figure 9: Left Center Expression Value

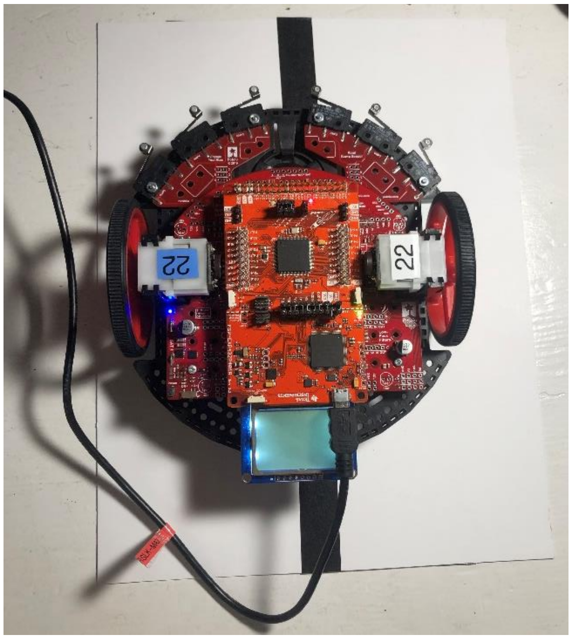
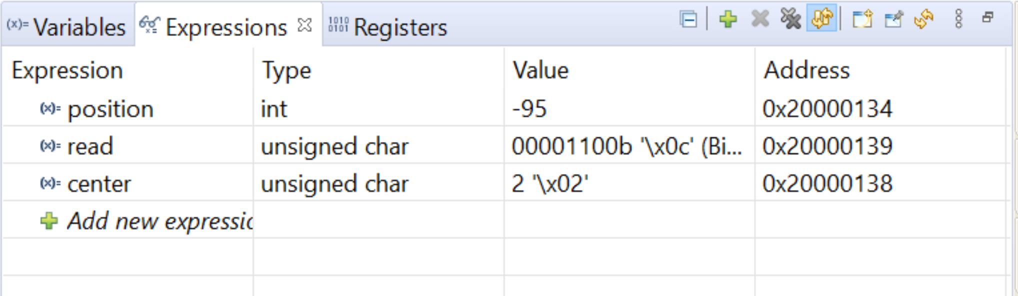


Figure 10: Center Right of Line



Expression	Type	Value	Address
position	int	-95	0x20000134
read	unsigned char	00001100b '\x0c' (Bi...	0x20000139
center	unsigned char	2 '\x02'	0x20000138
Add new expressic			

Figure 11: Right Center Expression Value

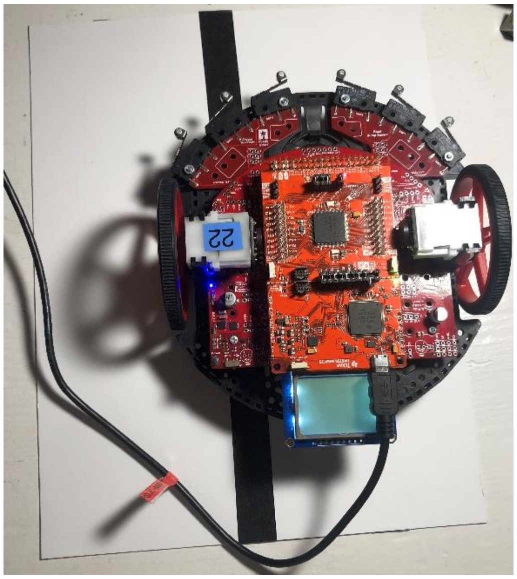
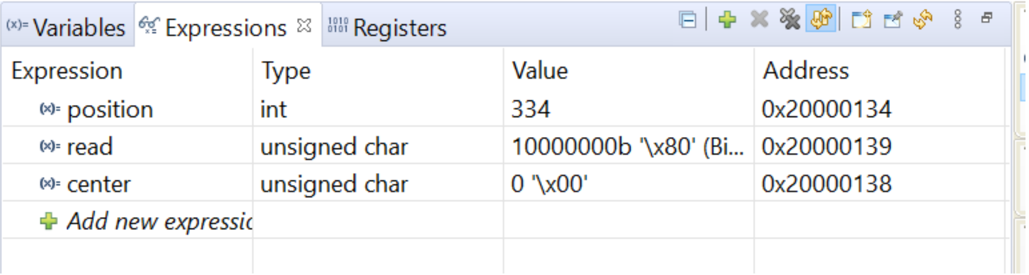


Figure 12: Left Sensor Array Bit



Expression	Type	Value	Address
position	int	334	0x20000134
read	unsigned char	10000000b '\x80' (Bi...	0x20000139
center	unsigned char	0 '\x00'	0x20000138
Add new expressic			

Figure 13: Left Position Expression Value

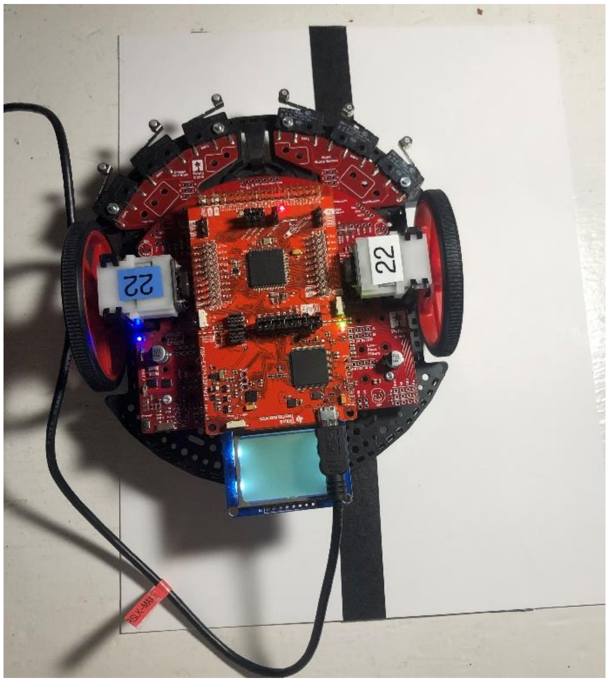
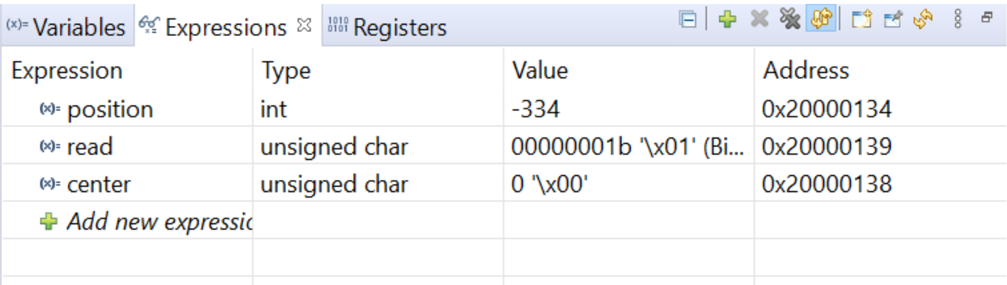


Figure 14: Right Sensor Array Bit



Expression	Type	Value	Address
position	int	-334	0x20000134
read	unsigned char	00000001b '\x01' (Bi...	0x20000139
center	unsigned char	0 '\x00'	0x20000138
Add new expressic			

Figure 15: Right Position Expression Value

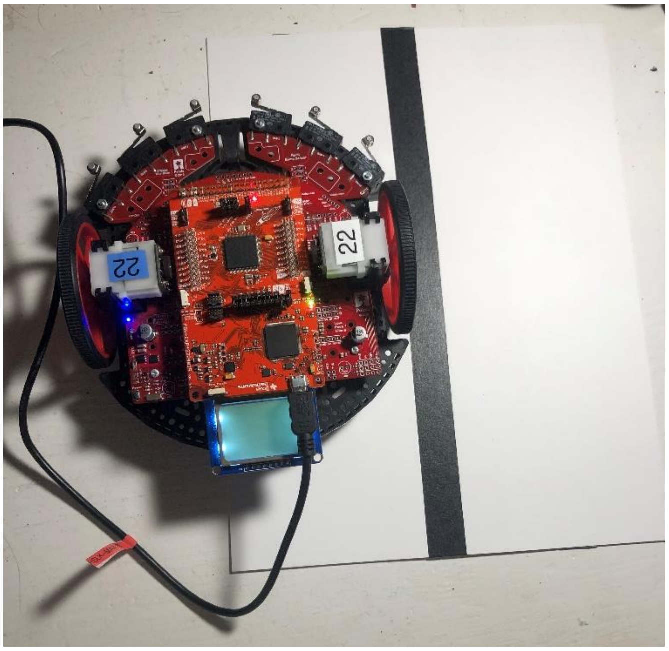
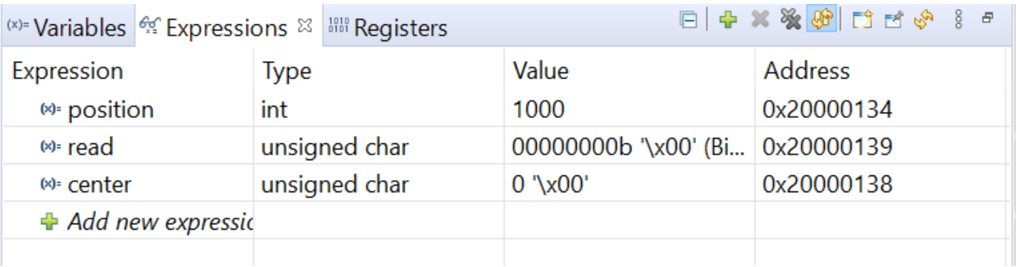


Figure 16: Sensor Array Off Black Line



Expression	Type	Value	Address
position	int	1000	0x20000134
read	unsigned char	00000000b '\x00' (Bi...	0x20000139
center	unsigned char	0 '\x00'	0x20000138
Add new expressic			

Figure 17: Position Expression Value when Off Black Line

Summary

A plethora of images of the sensor array along with expression verification was given for all three functions: read, position and center. Additionally, due to the correct values in each case it verifies that the initialization of the reflectance program was correct. Consequently, the task was entirely finished and outputs all the expected values. Therefore, the assignment was faultlessly finished.