Basic Roomba

Adithya Shastry

November 6, 2018
CS342:Project 3

1 Overview

The goal of this project was to make a small car that will avoid obstacles by turning. The car will take in sensory input(through the distance sensor), will have one button to turn the car on, and one to switch on a light on the car. This satisfies all of the criteria for the project. This project will incorporate ADCs, interrupts and GPIO in order to correctly work. I designed this project because I think it will be really cool to pull off something like this.

Note about the video: I decided to not include the tires in the video since I could not find anything to place my car on top of. It is possible to hear the motors if viewing them turn is difficult.

2 Part Requirements

Part	Number
Wheels	2
Motors	2
ATmega Board	1
Computer	1
Distance Sensor	1
Switch	1
Button	1
Wires	Copious amounts
Battery Pack with batteries	1
Tape	Copious Amounts
Frame for Car	1
Twist Tie	2

3 Circuit Implementation

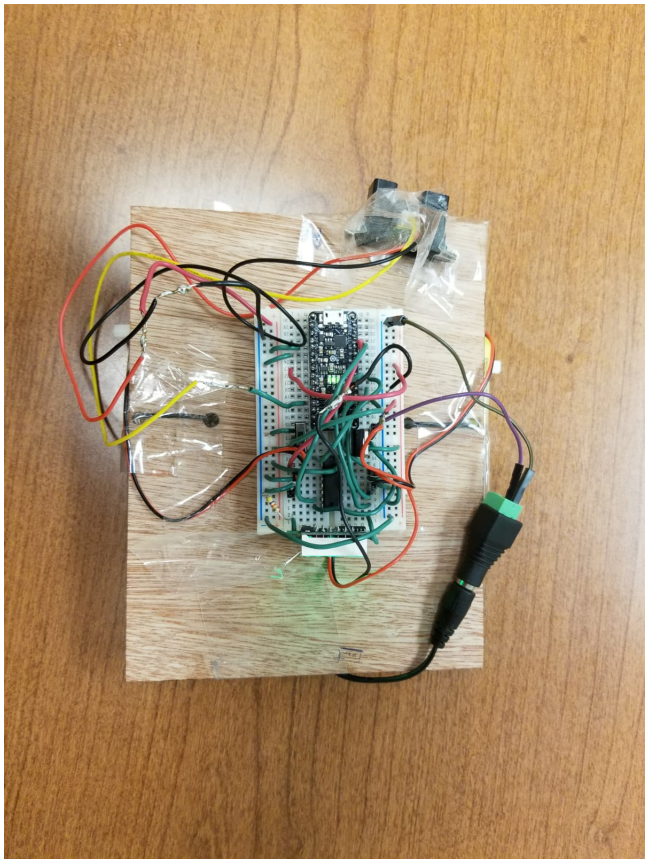

Figure 1: Hardware Implementation

4 Coding Implementation

```
#include <avr/io.h> // contains uint8_t and register
//definitions
#include "USART.h"


uint16_t sensedADC=0;
float mean=0.0;
int count=0;
 char msg[300];

//Enable the External Interrupt
  //Here you need to enable the motor interrupt and the
  //ADC
ISR(INT0_vect){
  //Enable everything
  if(PINB & 0b00000100){
```

```
17      //Enable GPIO
18     //For the Switch and Motor 2
19     //WE will se the motor to go from 3 to 4 and 1 to 2
20     DDRD     = 0b01011000;
21
22  //   motor 1
23
24  DDRB     = 0b00011000;
25
26     PORTB    =    0b00010000;
27     PORTD     =   0b01001000;
28
29     }
30     else if(PINB&0b00000000){
31       printString("Entered Else if");
32     DDRB    =    0;
33     DDRD     =   0;
34    TIMSK1=0b00000100;//Enable COMP1A and COMP1B
35
36  }
37  }
38
39
40
41
42  ISR(ADC_vect){
43  //the ADC ISR
44      //This will have logic for the actual conversion and
45      //the sliding window
46       int N=200;
47
48
49     sensedADC=ADCL;
50     sensedADC|=(ADCH &0x03<<8);
51     mean=mean *float(N-1)/float(N)+float(sensedADC)/float(N);
52
53
54
55
56     //Convert the Mean to a distance in cm
57     int distance= 0.0017*pow(mean,2)-1.0519*mean+197.75;
58
59     sprintf(msg, "online mean: \t%d\n",int(distance));
60     printString(msg);
61
```

```
62      //Have logic for how to handle possible collisions
63      //with distances less than 20cm
64       //Turn on the timer interrupt 1
65       //Switch the direction of the motors
66
67  //I found that the equation I derrived from the data
68  //collected from my sensor was not accurate, thus I
69  //updated
70  //The equation to better fit the error in my equation
71
72    if(distance<=150){
73      printString("Distance Less than 65\n");
74      TIMSK1=0;//Enable COMP1A and COMP1B
75        PORTB   =    0;
76    PORTD    =    0;
77      ADMUX=0;
78    }
79
80  }
81
82
83
84
85
86
87
88  ISR(TIMER1_COMPB_vect){
89
90  }
91
92
93
94  int main(){
95
96    initUSART();
97
98
99
100
101   //ADC
102   //The sensor is connected to A3
103    DDRC    = 0b00000000;
104    PORTC   = 0b00000000;
105
106
```

```
107
108
109
110     TCCR0A  = 0b11000011;
111    TCCR0B  = 0b00000100;
112    OCR0A   = 10;
113
114
115
116    //Enable Timer 1 COMPA for turning duration
117    //COMPB for The ADC
118    //We can have this count to 1ms and use a counter to
119    //count the correct duration. This will make testing easier
120
121    TCCR1A=0b00000000;
122    TCCR1B=0b00001011;//Set to CTC mode and have prescale of 64
123    OCR1AL=250;
124    OCR1BL=250;//Want to be able to use COMP1B
125    TIMSK1=0b00000100;//Enable COMP1A and COMP1B
126
127
128
129    //Enable ADC control
130   ADMUX=0b01000011;
131   ADCSRA=0b11101111;
132   ADCSRB=0b00000101;//Set the ADC to run based on Compare match B
133
134
135    //Enable the Switch with Pin change Interrupt
136    //PCINT18---->PCI2---------->PD2
137
138    //Use EINT0
139    EICRA= 0b00001101;//Falling edge for Eint1 and any
140    //logical change for EINT0
141    EIMSK= 0b00000011;
142
143
144
145
146
147
148
149
150
151
```

```
152
153     //Enable Global Interrupts
154     SREG|=0x80;
155
156     while(true){}
157 }
```

5 Testing and Possible Further Improvements

Further imporvements would include using better motors to allow the car to be fully mobile, even with the added weight of the battery pack. During my testing for this, I found that the normal force on the wheels of the car, were too large for the motors to actually move the car.Therefore, I had to remove the battery pack from the car. A stronger motor, one with greater torque, would greatly help carry the added weight of the battery pack.

6 Conclusion

Overall,this was a great project to work on because it taught me not only how to set up the software, but also how to set up the physical parts of the car in order for it to actually work. In order to do this I had to utilize a lot of the physics I had learned freshman year. This is was a really cool experience for me because I never had to really apply things learned in physics like that before!