

---

# ECE 375 LAB 2

C -> Assembler -> Machine Code -> TekBot

Lab Time: Tuesday 6-8

*Aditya Kothari (Adit Kotharii)*

*McIntyre Santa Cruz*

## INTRODUCTION

The goal of this lab was to rewrite the Lab 1's assembly program "BumpBot" in C and Learn how to configure I/O ports of the ATmega128 Microcontroller. We were provided with a sample program "DanceBot" in C, which we compiled and loaded on to the ATmega128 board. The sample program helped us understand how to code in C for the microcontroller, which in turn helped us write the "BumpBot" program in C.

## PROGRAM OVERVIEW

The BumpBot program, makes the TekBot react to input to its whiskers (two buttons). By default, the TekBot is meant to move forward until it meets with an obstacle, that triggers its whiskers. If the collision with the obstacle is head on, which triggers both of its whiskers (both the buttons are pressed), the TekBot reacts to the input by moving backwards a bit then turning right and continuing to move forward. If either of the two whiskers (either of the two buttons pressed) are triggered the TekBot backs up a little, changes the direction and continues to move forward.

## MAIN ROUTINE

The main function contains an infinite loop that executes a series of commands that control the signals to the motors of the Tekbot, depending on the input from the whiskers. We also used the "\_delay\_ms" function to time the signals to the motors.

## ADDITIONAL QUESTIONS

*1) This lab required you to compile two C programs (one given as a sample, and another that you wrote) into a binary representation that allows them to run directly on your mega128 board. Explain some of the benefits of writing code in a language like C that can be "cross compiled". Also, explain some of the drawbacks of writing this way*

The main advantage of C is that it makes it lot easier to read and write the code compared to assembly. In addition, C with its ability to be cross compiled allows us to run the program on multiple devices without requiring us to re-write the code. One of the drawbacks of C is that it is a high-level language doesn't give us access to processor/ hardware as much as assembly does.

*2)The C program you just wrote does basically the same thing as the sample assembly program you looked at in Lab 1. What is the size (in bytes) of your Lab 1 & Lab 2 output .hex files ? Can you explain why there is a size difference between these two files, even though they both perform the same BumpBot behavior?*

The hex file of BumpBot program from lab1 is 490 bytes, while the C version of BumpBot is 938 bytes. In spite of the two programs functioning being identical, the C hex file is almost twice the size of assembly hex file. The reason for the size difference seems to be all the additional libraries that the C program is packs in to make the code more easily readable. We trade code density for ease of readability.

## CONCLUSION

In this lab, we learnt how to code in C for the ATmega128 microcontroller and how to configure I/o ports by writing code for Bumpbot from assembly to C.

## SOURCE CODE

```
Port B, Pin 7 -> Output -> Left Motor Enable
Port B, Pin 6 -> Output -> Left Motor Direction
*/

#define F_CPU 16000000
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>

int main(void)
{
    DDRB = 0b11110000;    // configure Port B pins for input/output
    PORTB = 0b01100000;    // set initial value for Port B outputs
    DDRD = 0b00000000;    // set it up for input
    PIND = 0b11111111;    //1 = none of the buttons are pressed, 0 = button pressed

    while (1) { // loop forever

        PORTB = 0b01100000;    // make TekBot move forward

        if(PIND == 0b11111100) //bump head on, both whiskers hit
        {
            PORTB = 0b00000000; //move backwards
            _delay_ms(500); // wait 500 ms
            PORTB = 0b01000000; //turn right
            _delay_ms(500);
        }

        if(PIND == 0b11111110) //bump right, right whisker hit
        {
            PORTB = 0b00000000; //move backwards
            _delay_ms(500); //wait 500 ms
            PORTB = 0b00100000; //turn left
            _delay_ms(500);
        }

        if(PIND == 0b11111101) //bump left on, left whisker hit
        {
            PORTB = 0b00000000; //move backwards
            _delay_ms(500); // wait 500 ms
            PORTB = 0b01000000; //turn right
            _delay_ms(500);
        }
    }
}
```