ECE 375 LAB 2

– C → Assembler → Machine Code → TekBot

Lab Time: Wednesday 12-2

Bradley Martin

INTRODUCTION

The purpose of this lab is write a simple C program for the courses ATmega 128 microcontroller. Using C coding we will initialize and set registers for input and output. Sample code called “Dancebot.c” was provided as an example for creating a C project in Atmel Studio and uploading that code to our board.

PROGRAM OVERVIEW

The C program provides the instructions for how to handle whisker input. Two buttons on the ATmega 128 microcontroller act as whiskers for a tekbot. The tekbot will initially move forward, then depending on what whisker/button is pushed will depend on how the tekbot reacts. If the right whisker is hit, then the tekbot will backup, turn left, and continue forward. If the left whisker is hit, the same routine will happen except with a right turn. If both buttons are hit, then the tekbot will turn away left again.

INITIALIZATION ROUTINE

The initialization routine only runs once initializing registers for the program to start correctly. First PORTB is initialized to all output to tell the tekbot which motor to control. PORTD is initialized for all input to take input from the whiskers. Last is setting PORTB for the motors to start moving forward.

MAIN ROUTINE

The Main routine executes a simple loop that checks to see if a whisker was hit. The loop looks for input from PIND. If PIND detects input, then either the left turn or right turn will execute.

HITRIGHT ROUTINE

For right whisker input the tekbot will set PORTB to move backwards and a delay command to allow the tekbot to move backwards for 1 second. PORTB is then set to turn left for 1 second. Then the tekbot is set to move forward again.

HITLEFT ROUTINE

For left whisker input the tekbot will set PORTB to move backwards and a delay command to allow the tekbot to move backwards for 1 second. PORTB is then set to turn right for 1 second. Then the tekbot is set to move forward again.

HITLEFTRIGHT ROUTINE

If Both the left whisker and right whisker are hit at the same time, then the tekbot will set PORTB to backup for 1 second. Then to turn left for 1 second. Then continue forward.

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 you 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.

Some benefits to writing code in C is that it is easier on the programmer. Code can be visually smaller and organized in a way that is not as complicated as assembly code. This also allows more people to understand the code they are looking at. The code could potentially work on more products than just our mega128 board as well. If other IDE's don't accept assembly code directly then we can use the C code we already have. A drawback to writing in C is that it adds another step for the microcontroller to go from a high-level language down to machine code taking time and memory space.

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 for lab 1 is 485 bytes and the hex file for lab 2 is 958 bytes. This is because C is a higher-level programming language and interprets things differently than assembly that can add to the file size. Assembly is a low-level language and only does what it is told without many extra interpretations.

CONCLUSION

In this lab, we created a simple C program for our ATmega128 boards using Atmel Studio. The resulting program allows the tekbot to handle cases where it receives input from its whiskers. This lab was a good comparison to seeing how a problem could be solved using either C code or Assembly code as seen in Lab1.

SOURCE CODE

```
/*
Lab2.c

Created: 10/13/2020 8:44:54 PM
Author : Bradley M. Martin


This code will cause a TekBot connected to the AVR board to
move forward and when it touches an obstacle, it will reverse
and turn away from the obstacle and resume forward motion.

PORT MAP
Port B, Pin 4 -> Output -> Right Motor Enable
Port B, Pin 5 -> Output -> Right Motor Direction
Port B, Pin 7 -> Output -> Left Motor Enable
Port B, Pin 6 -> Output -> Left Motor Direction
Port D, Pin 1 -> Input -> Left Whisker
Port D, Pin 0 -> Input -> Right Whisker
*/

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

int main(void)
{
        DDRB = 0b11111111; //configure Port B pins for input
        DDRD = 0b00000000; //configure Port D for output
        PORTB = 0b01100000; //set initially forward
        while (1) // loop forever
```

```
	{
		if(PIND == 0b11111110){  //if the Right whisker is hit
			//turn left
			PORTB = 0b00000000; //move backwards
			_delay_ms(1000);  //wait for 1 sec
			PORTB = 0b00100000;	//turn to the left
			_delay_ms(1000);//wait for 1 sec
			PORTB = 0b01100000;	//continue to move forward
		}
		else if(PIND == 0b11111101){//if the Left whisker is hit
			//turn Right
			PORTB = 0b00000000; //move backwards
			_delay_ms(1000);//wait for 1 sec
			PORTB = 0b01000000;	//turn to the right
			_delay_ms(1000);//wait for 1 sec
			PORTB = 0b01100000;	//continue forward
		}

		else if((PIND == 0b11111110) && (PIND == 0b11111101)){//if both whiskers are hit are
pushed
			//turn left
			PORTB = 0b00000000; //move backwards
			_delay_ms(1000);//wait for 1 sec
			PORTB = 0b00100000;	//turn left
			_delay_ms(1000);//wait for 1 sec
			PORTB = 0b01100000;	//continue forward
		}
	}
}
```