

SECTION TWO C → ASSEMBLER → MACHINE CODE → TEKBOT

SECTION OVERVIEW

Complete the following objectives:

- Look at a sample C program
- Write a sample C program for the AVR microcontroller
- Compile the code using the Atmel Studio GCC compiler
- Download this code to your AVR board and verify its function

PRELAB

Remember—no late prelabs will be accepted.

Write a pseudo code program for making your TekBot move the way it did in Lab 1. Your robot should detect objects using its whiskers, reverse, and turn away from the object.

PROCEDURE

Looking at C Code in AVR Studio

1. Download the sample code available on the web page. This is simple C code that is well commented and ready to compile. All code that you produce should be as well commented as this code. Save this code where you can find it.
2. Open Atmel Studio 6. Create a new project following the instructions in Section 2.1.2 of the AVR Starter Guide, except choose C/C++ under Installed Templates, and select **GCC C Executable Project** as the project type.
3. We now need to add the source file you downloaded earlier into the project you just created. Again, follow the instructions in Section 2.1.2 of the AVR Starter Guide to complete this step.
4. Examine the source file you just included in your project, so you understand what is happening. It is written in ‘C’ (you should have learned C in CS151, CS261, or elsewhere). If you are having difficulties, check the web and teach yourself or ask for assistance from your TA.

Pseudo Code for Lab 2 ‘Dance’ C Code.

- Initialize Port(s)
- Loop Forever
 - Forward for 500mS
 - Reverse for 500mS
 - Right for 1000mS
 - Left for 2000mS
 - Right for 1000mS

Compiling and Loading the Code to the TekBot

1. To compile the code simply go to 'Build->Build Solution' or press F7. The log at the bottom of the screen should tell you that there are no errors, assuming you have done everything correctly up to this point. You will notice that a **.hex** file was created from this operation.

Open the Universal Programmer and Program the TekBot as you did in Lab 1. Of course your TekBot needs to be connected to your PC and powered on first.

Your Own Code

You need to write a simple C program that will make your TekBot perform the “BumpBot” routine, as described below. Your TekBot should travel forward until it encounters an object, then back up and turn away from the object. If for some reason both whiskers are triggered at the same time, your TekBot needs to back up and turn to the RIGHT. HINT: Use the ‘_delay_ms()’ function.

Theory of Operation for TekBot “BumpBot”

- Initializes key components of the ATmega128
- Starts the TekBot moving forward
- Polls the whiskers for input
- If right whisker is hit
 - Backs up for a second
 - Turns left for a second
 - Continues Forward
- If left whisker is hit
 - Backs up for a second
 - Turns right for a second
 - Continues Forward

1. Write a simple C program that performs this function. You will probably want to use the skeleton code available on the web and modify it as needed. Remember which version of the TekBot you have when designing the motor control logic. It is recommended that you use the pin out in the skeleton file, but it is not necessary.
2. Demonstrate the operation of your TekBot to your TA for credit. Have him/her sign below.

TA Signature: _____

STUDY QUESTIONS/ REPORT

Write a short summary that details **what you did and why, explain any problems you may have encountered, and answer the questions below**. Your write up and code must be submitted to your TA at the beginning of class the week following the lab. **NO LATE WORK IS ACCEPTED.**

Study Questions

1. This lab required you to begin using new tools for compiling and downloading code to your AVR-enabled TekBot using the C language. Explain why it is beneficial to write code in a language that can be ‘cross compiled.’ Also explain some of the problems of writing in this way.
2. Your program does essentially the same thing as the assembly program you downloaded for Lab 1. Compare the size of the output hex files and explain the differences in size between them.

CHALLENGE

2. Modify your TekBot so it can move objects across a tabletop. Your TekBot needs to push objects that it touches a short distance. An example of how your TekBot should work is:
 - a. TekBot hits object.
 - b. TekBot continues forwards for a short period of time.
 - c. TekBot backs up slightly.
 - d. TekBot turns slightly towards the object.
 - e. TekBot repeats steps a through d.

There are several ways to approach this problem, but you must accomplish it by revising your firmware (change the C program or rewrite it). To get credit for this challenge, you must demonstrate the operation of your TekBot to your TA and **submit a copy of the code used**.

TA Signature: _____