

Lab2

Stack Overflow¹

Arduino Workshop

Objectives

Understand the memory constraints, organization and see what happens when we take Arduino to its limits. Also we will explore the concept of pointers and callbacks for designing event-based system.

General Instructions:

Problem 2.1 is designed to be a demonstration for a possible systemic failure. Problem 2.2 is designed to take you through the process of building callbacks triggered at different pre-defined intervals.

Both these problems might appear to be difficult to a person new in C programming. But the funny thing is, they are not difficult, but unfamiliar! So chill... it's all easy. Just go on doing the sub-problems and see if you enjoy the end result!

Problem 2.1: *Save me Sherlock!!*

Well, this problem does not require you to write a single piece of code! But it expects you to find the bug. A working-fine version of code is provided to you along with a crashing version. Your job is to identify why it is crashing!

Just as side-note, don't go running into code haphazardly trying to find the bug! Try to think of a strategy, how to find where the error is.

Try to make as many possible observations and inferences as possible. Support your inferences. Discuss with your instructor.

Problem 2.2: *Event-Programming...*

Well, I hope you didn't fall asleep during the theory session. We shall take a look the concept of pointers in C/C++ in this problem.

To begin with, let's do a trivial problem → Problem 2.2.1

¹ Author: Abhishek N. Kulkarni (abhishek.kulkarni12@vit.edu)

Problem 2.2.1: *Hello World...*

Hello World is no longer a stranger to us. In this problem, you should declare a string variable using pointers. Discuss with your instructor → if it's so easy to use pointers for declaring strings, then why are arrays commonly used for string representation?

Problem 2.2.2: *The word goes upside down...*

Now, that you can store a string using pointers, write a function which inputs a string and reverses it. For example, *vit-pune* → *enup-tiv*

Do not use *String.h*

Problem 2.2.3: *Function pointers...*

As we discussed, the pointers are just one memory location storing the address of another memory location. But this address may be of any data structure. It could be character, integer or perhaps a function too!

Print the address of function you implemented in Problem 2.2.2

Problem 2.2.4: *Callbacks...*

I hope you are clear with concept of callback/event by now. Don't say no... we discussed it in lecture! So let's create a callback.

In this problem, we'll implement a piece of code that prints a message every 1 second and blinks a led every 500ms. The functions are provided to you. Your job is primarily to understand the code provided to you.

Next, I think now you might be able to resolve the issue of lighting 3 LEDs with different intensities from Lab1. Think and devise a plan of implementation on paper first (I recommend). Then implement it in Arduino and see if it works.

The take-home lessons to be learnt

- What looks to be trivial can trouble the most!
- If you know the theory, life becomes simple!

Have fun...

Keep Learning!