Assignment 2
Blinking LED

Embedded Logic Design

August 11, 2015

The main part of this assignment is divided into several parts. Please read the instructions carefully and make sure that you understood them.

1 Logistics

1. Find out your group number by checking your entry at `https://docs.google.com/a/iiitd.ac.in/spreadsheets/d/1OQ3JhiSssSjxjrtGrweAejZgAdGqL8YUxCQ1GVZhqO8/`
2. Log in the server at `nanu.iiitd.edu.in` via SSH (ssh in linux, putty in Windows) and change your password. The user name is your group name (e.g. groupXX) with the same password (in this case groupXX). You are forced to change your password before you can access the server. Please note that the server is only reachable from within the IIITD network.
3. Log in into the server and have a look at the README.TXT in your home directory, which gives you the locations to the libraries and header files, if you would like to use the compiler on the server.
4. For every group, pick up one Arduino board from Mrs Aarti, Library Building, 3rd floor, if you have not done so far.

2 Installation

If you have not done so far, download and install the Arduino IDE for your operating system from `https://www.arduino.cc/`.

3 Actual Assignment

1. Have a look at the following code which makes the onboard LED blink in intervals of 2 seconds (repetitively 1 second on, 1 second off). Make sure you understood the workings of the code (you may also try it out on the Arduino).

```
1  // the setup function runs once when you reset/power the board
2  void setup() {
3    pinMode(13, OUTPUT);
4  }
5
6  // the loop function runs over and over again forever
7  void loop() {
8    digitalWrite(13, HIGH);
9    delay(1000);
10   digitalWrite(13, LOW);
11   delay(1000);
12 }
```

2. **From now onwards, you are not allowed to use the Arduino IDE nor its libraries. You will have to use the AVR library only**, which has already been installed on your computer, when you installed the Arduino IDE.

3. Write the same program as in item 1 above by using the AVR library **only**. You will have to think, search and find replacements for the functions such as `pinWrite`, `digitalWrite` and `delay` found in the Arduino library.

   Optimize your code as much as possible in terms of space requirements in the program memory as well as SRAM.

4. Create a `Makefile` so that you can compile your code at the command prompt (i.e. without using the buttons in the Arduino IDE). This `Makefile` has to contain 3 sections at least to separately compile and program your device plus to clean up all created files. Fill these sections with the suitable commands so that someone else can compile the code, upload the hex-file or clean the project by executing `make compile`, `make program` and `make clean` respectively at the command prompt. If no parameter is given, the code is compiled and the Arduino is programmed. In the `Makefile` use variables for the compiler executable, source code, port number, directories for libraries and include files, so that an easy change is possible at a single location. This will be useful for later assignments.

5. What is the amount of space required in the flash for your programs using the Arduino and the AVR library? What is the program that consumes the least amount of space?

4 Deliverables

All files must be submitted to `nanu.iiitd.edu.in` via `git` or `subversion`. Late submissions are not evaluated nor will be submissions through `https://www.usebackpack.com` or mail. Your repository has to contain:

- Source code
- Makefile

4.1 Remarks

Apart from your solution in section 3 (1) you *must not* use the Arduino library, only the AVR library is allowed. If you encounter a problem, ask Google, DuckDuckGo, Bing, etc. first. The TAs will not type the question that you have, into the mask in the search engine for you. Required resources, textbooks, etc. are available on the ELD course website of `https://www.usebackpack.com` or in the Internet (datasheets, AVR library documentation, etc.)