EEEN202, 2021, Microprocessor Lab 1, Part A

Introduction to microprocessor machine code programming

During this first session, the idea is to become acquainted with the fundamentals of a microprocessor and how it operates at the machine code level. You will also learn about the development environment that you will be using for the next sets of labs where you build circuits and then write, compile, debug, download and run your programs to run on your configured system. The microprocessor you will be using is based on the 8051 series core that can be found in many devices such as the AT89C51AC3.

Your microprocessor development board is connected to the PC's USB port, which has been configured as a RS232 serial port. There are a number of connectors on the development module which will be used to interface to several other devices such as an LCD display and a digital to analogue converter.

Using My Computer/Windows Explorer, create an EEEN202 folder in your H: drive (personal home directory) or on a USB flash drive. During the lab sessions you will save your work here in folders relevant to the lab being undertaken. The skeleton code for the current and future lab sessions is available on blackboard under EEEN202 → lab files. Download the lab1.HEX file into your newly created folder.

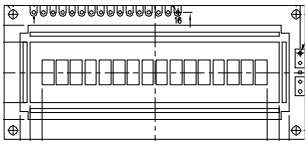
Initially we will be using a simple programmer tool to download and program your code into the development board's FLASH memory. Later we will be using a professional code development tool called KEIL. So things will get a little easier, but for now we are going to jump into the lowest possible level to learn how a microprocessor executes code. The goal of this lab is for you to write microprocessor code so that it will display a message of your choice on an LCD display.

Before a program can be run it must have suitable hardware to run on. As you will have noticed, the microprocessor development board doesn't have an LCD display! We will have to connect one to it by using two ribbon cables.

Please connect one ribbon cable between the “LCD DATA” connector of the display module and “PORT1” of the microprocessor development module. Connect another ribbon cable between the “LCD CONTROL” of the display module and “PORT2” of the microprocessor development module.

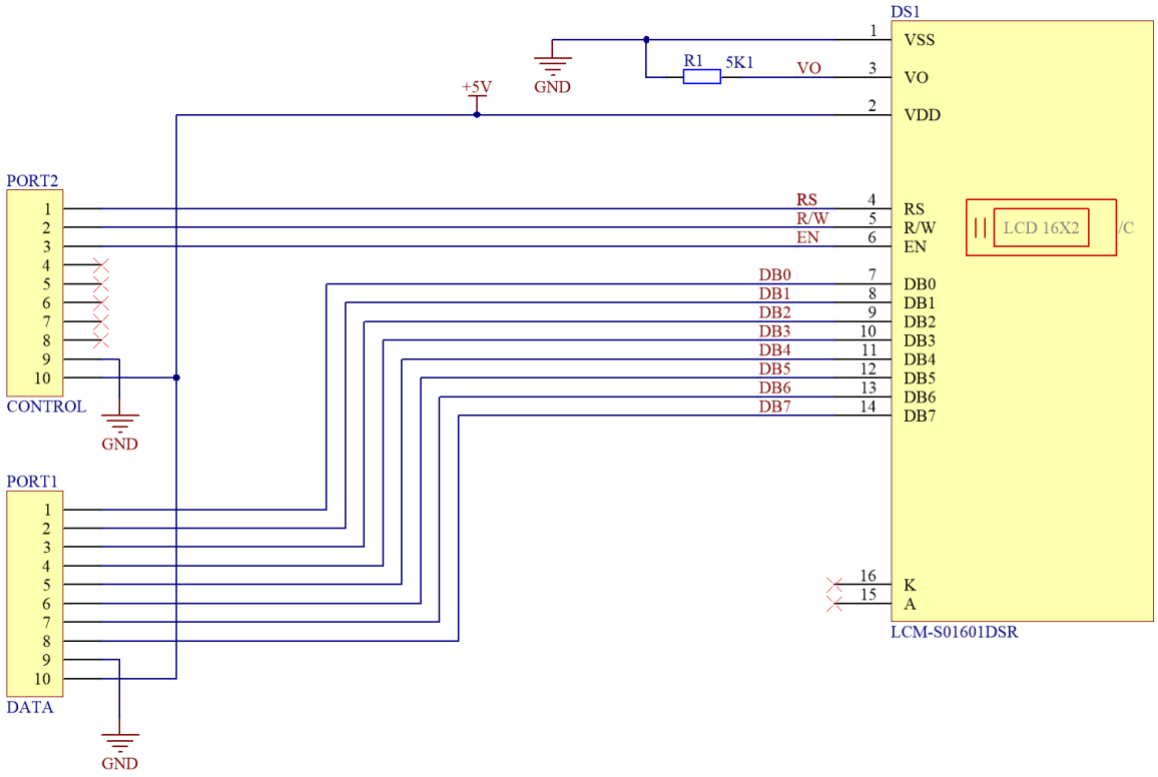
The diagrams below indicate the display circuit wiring as well as the functions of the display connections.

Lumix LCM-S01601DSR, 16x1 LCD display



Pin Description and Wiring Diagram

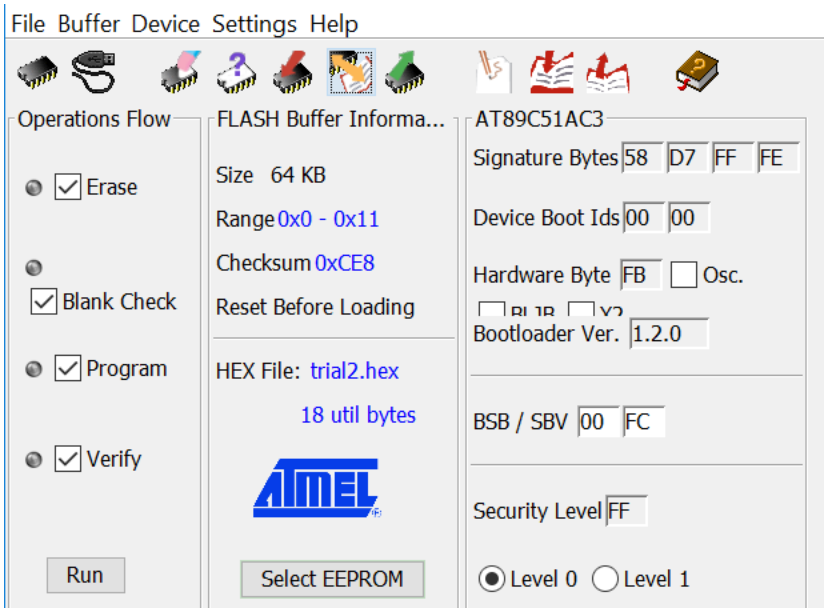
Pin No.	Symbol	External Connection	Function Description
1	V_{SS}	Power Supply	Ground
2	V_{DD}	Power Supply	Supply Voltage for Logic (+5.0V)
3	V_0	Adj. Power Supply	Supply Voltage for Contrast (approx. 0.5V)
4	RS	MPU	Register Select signal. RS=0: Command, RS=1: Data
5	R/W	MPU	Read/Write select signal, R/W=1: Read R/W: =0: Write
6	E	MPU	Operation Enable signal. Falling edge triggered.
7-10	DB0 – DB3	MPU	Four low order bi-directional three-state data bus lines. These four are not used during 4-bit operation.
11-14	DB4 – DB7	MPU	Four high order bi-directional three-state data bus lines.
15	NC	-	No Connect
16	NC	-	No Connect



At this stage, it is useful to connect up the microcontroller to the PC. We need to do this in order to setup the FLASH memory programming tools. Connect the Microcontroller using a 9V DC power supply and use a USB cable and connect it to the PC. Both the power supply and the USB cable must be connected for the micro to communicate to the PC. Ensure that the UART switch on the box is turned to the 'on' position. There are two indicator LEDs, one beside the USB, and one beside the DC jack. **Check that both of them are on before continuing. If you are having problems, ask the demonstrator.**

Launching and Configuring the FLASH memory programming tools

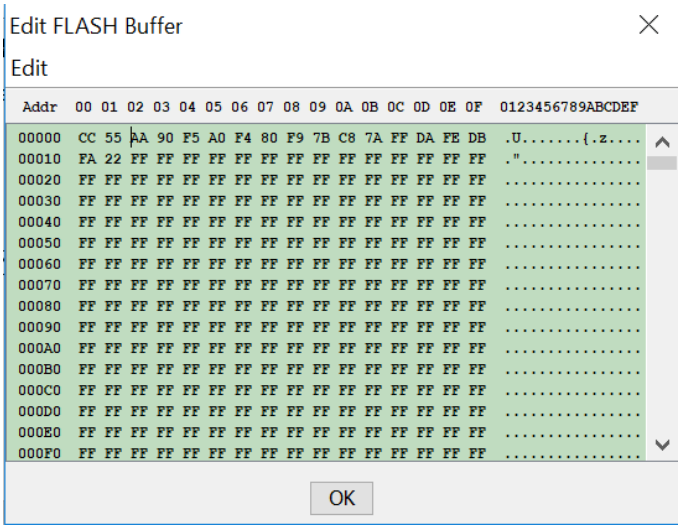
Launch the Atmel "Flip" tool version 3.4.7.



Select the "Device" menu and then "select", choose the AT89C51AC3 device.

Select the "File" menu and then select "Load Hex file" and load the lab1.hex file. This file contains the bit stream (binary file) that will be downloaded to the board.

Select "Buffer" and then select "Edit" to see the contents of the file.



Click “OK” to exit.

The next step is to see the program running, (it is rather useless otherwise!). To do this, we need to download the program into the microprocessor board FLASH memory. The steps for preparing the micro board for programming are:

- Make sure your board has power, and the USB cable is plugged in.
- Make sure the UART is switched on.
- Press and hold the Program button.
- While holding the Program button, press and release the Reset button.
- Release the Program button.

Select “Settings” menu and then “Communications”. Choose RS232. Then choose the COM port that the USB is connected to. Finally, choose “Baudrate” 9600.

To download the program, click on the “Run” box at the bottom left of the window.
After a few seconds, your program should be finished downloading into the FLASH memory.

Press the RESET button to run the program. **You must press reset after each new upload to run your program.**

Verify that the “Hello” message appears in the LCD. (ask the demonstrator if this is not the case).

So, we have now simply "downloaded" a precompiled program onto a microprocessor. Pretty simple stuff so far isn't it? This may seem trivially easy, but it is vital that you get a good feel for the system before you attempt anything too complicated.

Your task now is to decipher the Lab1.hex code and then change it to display your own message. Download the at_c51ism.pdf and AT89C51AC3.pdf files to help you work out what is happening. Don’t Panic!, we will help you through this.

This first part of lab 1 is a “warm up” exercise and you do not need to discuss it in your Lab1 report.