Lab 1

Introduction to Microcontroller

Feb. 2016
Objective

1. To be familiar with microcontrollers.
2. Introducing LPC2138 microcontroller.
3. To be familiar with Keil and Proteus software tools.

Introduction

A microcontroller (or MCU) is a computer-on-a-chip. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (the kind used in a PC).

A microcontroller is a single integrated circuit, commonly with the following features:

- Central processing unit (CPU) - ranging from small and simple 4-bit processors to sophisticated 32- or 64-bit processors
- Memory (SRAM and Flash memory)
- Input/output interfaces such as serial ports (UARTs)
- Peripherals such as timers and watchdog
- Many include analog-to-digital converters
- Clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit

Microcontroller versus Microprocessor

A microcontroller differs from a microprocessor in many ways. The first and most important difference is its functionality. In order that the microprocessor may be used, other components such as memory must be added to it. Even though the microprocessors are considered to be powerful computing machines, their weak point is that they are not adjusted to communicating to peripheral equipment. On the other hand, the microcontroller is designed to be all of that in one. No other specialized external components are needed for its application because all necessary circuits are already built into it.
ARM Processor

ARM-Advanced RISC Machine, is one of a family CPUs based on a reduced instruction set computer (RISC) architecture developed by British company ARM Holdings. A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical processors in average computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices—including smart phones, laptops, tablet and notepad computers), and other embedded systems.

Memory Unit

- **Flash memory**
  The contents of this memory can be written and cleared practically an unlimited number of times, the microcontrollers with Flash ROM are ideal for learning, experimentation and small-scale manufacture.

- **Random Access Memory (RAM)**
  Once the power supply is off the contents of RAM (Random Access Memory) is cleared. It is used for temporary storing data and intermediate results created and used during the operation of the microcontroller.

- **Electrically Erasable Programmable ROM (EEPROM)**
  The contents of the EEPROM may be changed during operation (similar to RAM), but remains permanently saved even upon the power supply goes off (similar to ROM). Accordingly, an EEPROM is often used to store values, created during operation, which must be permanently saved.
LPC21xx Microcontrollers

The LPC2131/2132/2134/2136/2138 microcontrollers are based on a 32/16 bit ARM7TDMI-S CPU with real-time emulation and embedded trace support, that combines the microcontroller with 32 kB, 64 kB, 128 kB, 256 kB and 512 kB of embedded high speed Flash memory. A 128-bit wide memory interface and a unique accelerator architecture enable 32-bit code execution at maximum clock rate. Due to their tiny size and low power consumption, these microcontrollers are ideal for applications where miniaturization is a key requirement, such as access control and point-of-sale. With a wide range of serial communications interfaces and on-chip SRAM options of 8/16/32 kB, they are very well suited for communication gateways and protocol converters, soft modems, voice recognition and low end imaging, providing both large buffer size and high processing power. Various 32-bit timers, single or dual 10-bit 8 channel ADC(s), 10-bit DAC, PWM channels and 47 GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers particularly suitable for industrial control and medical systems.

LPC2138 Microcontroller

This is the microcontroller we are going to use during our lab. This microcontroller has the following main features:

- an ARM7TDMI-S based high-performance 32-bit RISC Microcontroller
- 512KB on-chip Flash ROM
- 32KB RAM
- Two 8-ch 10bit ADC
- Two UARTs
- Single 10-bit DAC
- Two I2C serial interfaces
- Two SPI serial interfaces
- Three 32-bit timers
- Watchdog Timer
- Vectored Interrupt Controller
- In-System Programming (ISP)
- In-Application Programming (IAP)
- Up to 47 general purpose I/O pins.
- CPU clock up to 60 MHz
- On-chip crystal oscillator and On-chip PLL.
Software Tools

These programs are the backbone of the microprocessor and microcontroller based systems; since using Keil we can build the software of the project using C, and then we can simulate the project virtually using Proteus, finally we can download the program on the microcontroller and see the practical results.

Keil MDK Program

Keil is a Microcontroller Development Kit “MDK” software program that runs on your PC to provide a development environment for your embedded system design.

Installation of Keil MDK v5

1- Double click on the icon of mdk522.exe.
2- When the Welcome dialog box pops up, click “Next” button.
3- In the next License Agreement dialog box, check “I agree” and click “Next” button.
4- In the next Folder Selection dialog box, accept the defaults and click “Next” button.
5- In the next Information dialog box, fill the information and click “Next” button.
6- Before installing ULINK device drivers, Windows will halt it and asks for confirmation. Click install button for confirmation.
7- When the installation is complete, this dialog box will appear. Click “Finish” button.

Installation of MDK v4 Legacy support

MDK Version 5 uses Software Packs to support a microcontroller device and to use middleware, The LPC21xx devices don’t not supported this. So, you should install Legacy support for it. Repeat the above steps for the MDK79522.exe file.

Create a new project with Keil MDK

1- From menu, select Project -> New uVision Project.
2- In the “Create New Project” dialog box, click “New folder”. Browse to a folder location where you would like to create your project folder.

3- Type in the folder name, for example “lab1_Project” then click the Open button. This will create a folder named lab1_Project to hold all the files for the new project.
4- While in the project folder, type “Blinky” in the File name field and click “Save” button. This creates a project with the project named Blinky. The project file has the file extension of “.uvprojx”.

![Image of project folder with File name field set to Blinky]

5- The Project Wizard will prompt you to select the device type for the project target. As we mentioned, the LPC21xx microcontroller does not supported by Keil Software Packs, so choose “Legacy Device Database” from the device menu.

![Image of device selection menu with Legacy Device Database selected]
6- Next, Search for “LPC2138” in the search field and choose it. Finally, Press “OK” button.

![Select Device for Target 'Target 1'...](image)

7- A dialog will appear to ask copying the Startup.s to the project folder and add file to project, press “Yes”.

![Prompt to copy Startup.s to project folder](image)
8- You should see a target created in the Project pane. If you click the + signs to open the “Target1”, and click right the “Source Group1” and then add a new file by choosing “Add New Item to Group “Source Group1””, this will open a window to add a new file.

9- The Add New Item Wizard will prompt you to select the file name and type, Select file type “C File”, enter the file name “Blinky” and finally click “Add” button.
10- Copy the code that is attached with lab into the new text file window. This is a simple program to toggle the LED on the LPC2138 microcontroller.

11- Click “Build” button or press “F7” to build the project.
You should get a clean build with this project.

Generate the Hex file machine code

1. Right click on the “Source Group1” and choose “Options for Target “Target 1”. Or simply choose “Option for Target” icon as shown.
2. Next, from the “Options for Target “Target 1”” dialog, click on “Output” from the upper menu bar and check on “create HEX file” and press “Ok” button.

![Options for Target “Target 1” dialog]

3. Click on “Rebuild” to rebuild the file to create a hex file.

![Rebuild dialog]

4. You should get a clean build with this project.

![Clean build screen]
Protelus Program

Proteus is a simulation program used to simulate hardware connections and check that they are error-free before connecting them.

Installation of Proteus

1- Run the “Proteus 8 professionals sp0 build 15417” as an administrator.
2- Press next until you reach the window which asks for the key.
3- From browse for key; browse until you find the file where the key exist.
4- Then click on Grassington North Yorkshire.lxk and then press install.
5- Then choose yes and close the window, after that press Next and choose Typical install to install the program.
6- After installation, for Crack, copy (“PIN”, ”Models”, ”Help”, ”LICENCE”) and paste them on the folder where the Protues installed in your PC.
7- Run the LICENCE and press install.
8- Finally, Choose “Ok” and then “Close”

Create a new project with Proteus

1- Open Proteus and click on “New Project” button.
2- Name your project with suitable name and click “Next”.

3- Choose “Create a schematic from the selected template” and click “Next”.
4- Click “Next”.

5- Click “Next”.
6- Click “Finish”. And now your workspace is ready.

7- Press the “Component mode” to choose the components from “P” icon.
8- As an Example, we deal with “LPC2138 Microcontroller” in this lab, from the Keyword field we write the keyword “LPC2138” and choose it then press “Ok” button.

9- From “Terminal Mode” we can find the “Power” and the “Ground”.
10- To install your program to the microcontroller, double click on the “LPC2138”, then from the “Edit Component” dialog click on “search file” field and browse for your .hex file and then press “OK”.

11- You can run the simulation as shown.

Good Luck