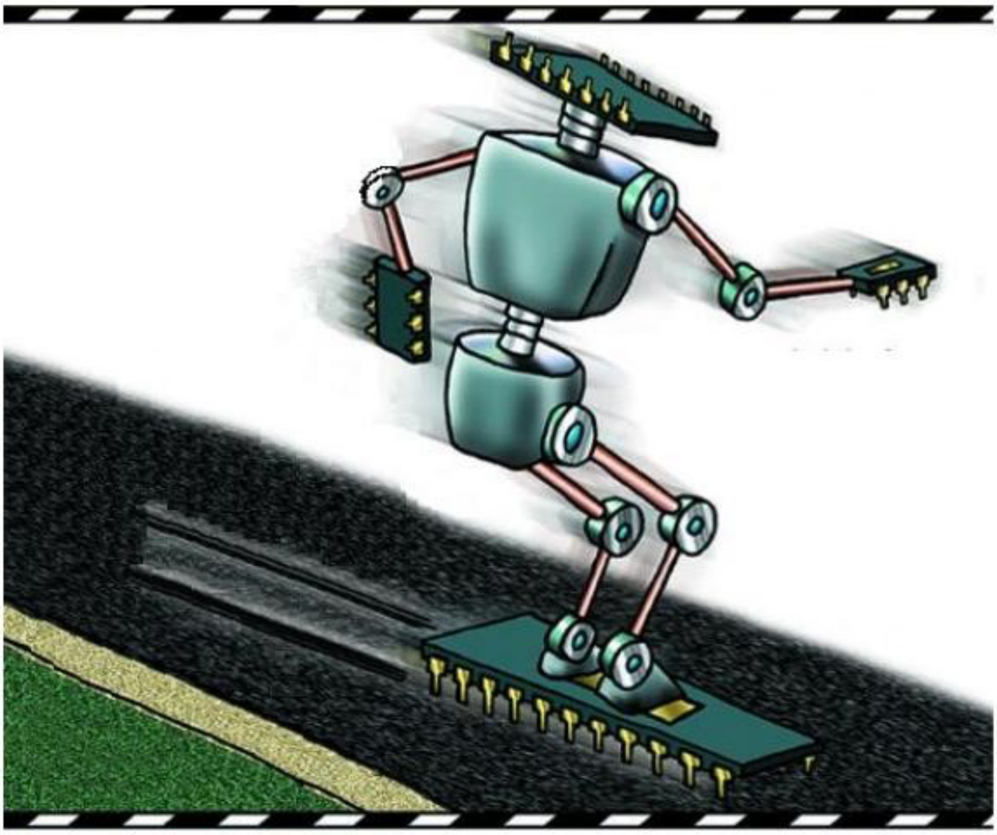
MICROPROCESSOR BASED SYSTEM DESIGN

LAB MANUAL



DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING
UNIVERSITY OF ENGINEERING & TECHNOLOGY
PESHAWAR

Prepared by: Engr. Salim Ullah

Revised by: Engr Amaad & Dr. Bilal Habib (Spring 2019)

Lab Objective

This course is focused on the principles and practices of modern Microprocessor Based Systems Design. We will focus on the Intel’s 8051 Microcontroller and other supporting hardware, to learn how to design, build and test embedded systems.

Contents

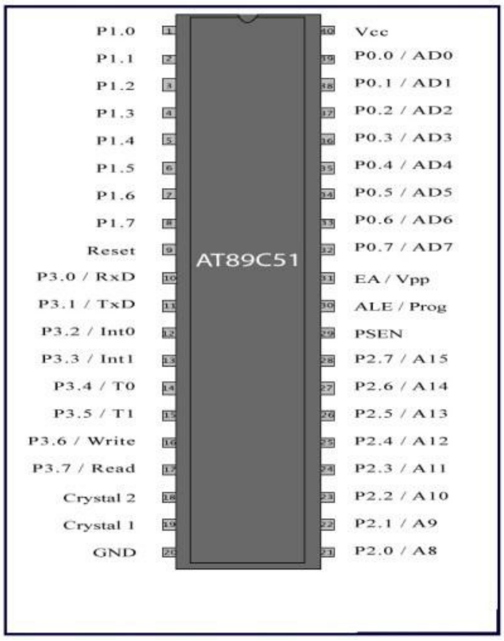
Lab No.1: Introduction to 8051 Microcontroller & Installation of software and hardware tools

Objectives of the lab

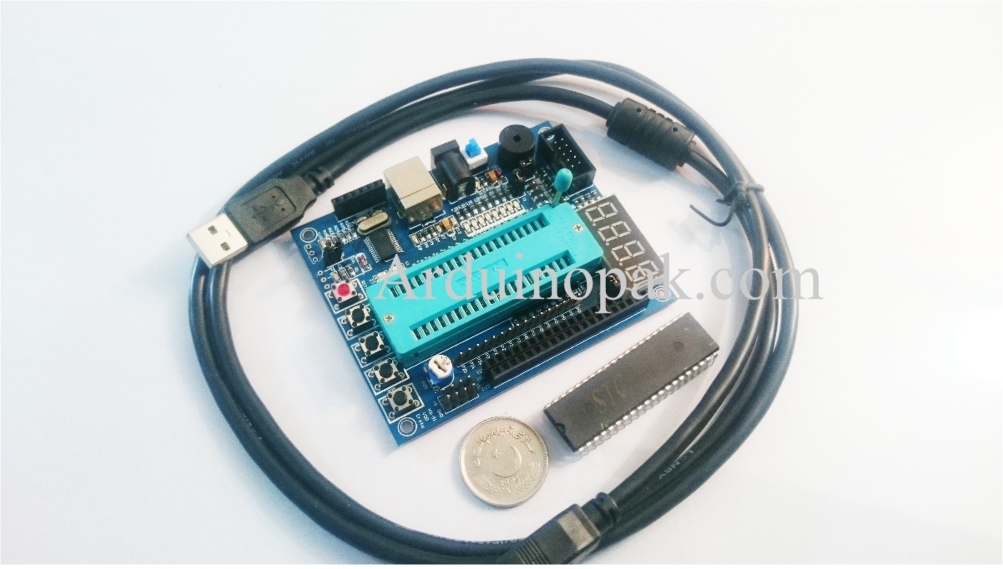
- To get familiar with Keil µVision Integrated Development Environment (IDE).
- Familiarization with 8051 Microcontroller.
- Toggling I/O Port.

Hardware & software requirement

- Breadboard
- Proteus
- 8051 Microcontroller (AT89C51)
- 8051 Assembler (Keil µVision).
- 8051 Programmer.



Training Kit



51/AVR MCU Development Board System Control Panel x 1

The trainer kit consists of

- 8 high-brightness LED
- 4 common Anode LED display;
- LCD1602 and LCD12864 (support color) interfaces;
- 1 channel passive buzzer;
- A channel DS18B20 temperature measurement circuit;
- 1 channel infrared interface circuit;
- 4 independent keys;
- 1 channel PL2303USB communication circuit.

Programmer

Programmer is used to burn the HEX file generated by Keil.

Setup Steps

Complete following step 1 to 3 befor connect the hardware to PC

- Open the folder DISK_1\EnglishVersion
- Double click the File “G540_Setup” to setup the G540 application program.

we can change the setup location ,otherwise the default is
C:\GeniusProgrammer

Return to windows desk after set G540 application program

- Execute the "StgICproUSB_Insteller" at Windows "Start—Program—Genius Programmer" or at the folder DISK_1\EnglishVersion\USB_DRIVER

Note:
This step is to pre-setup the USB driver. May be your computer will consume some minutes to find the USB driver files. Wait patiently please.

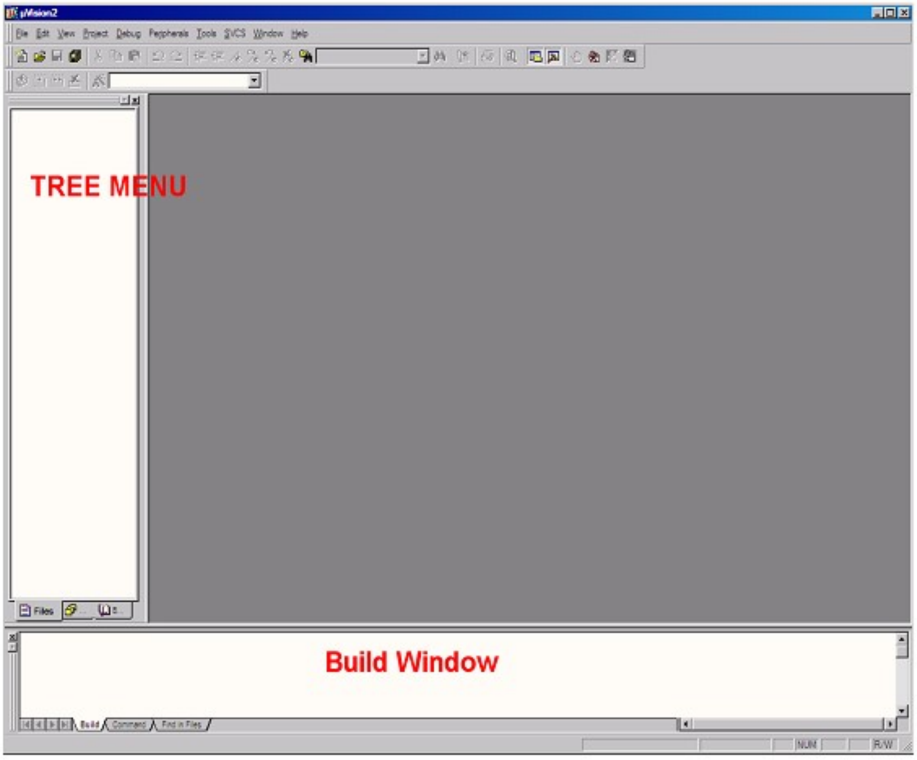
- Connect the G540 hardware to the USB interface of your computer. The "Now hardware…" will be displayed at the right down corner, and then the Finding Now Hardware Wizard dialog window will show.

Select Automatic then click the Next bottom ….till completion.

- ALL the setup steps were finished fully. Now , we can double click the icon G540 on the windows desk to start your usage

Keil µVision Introduction

1. Open KeilµVision from the Start menu
2. The Figure below shows the basic windows referred in this document.

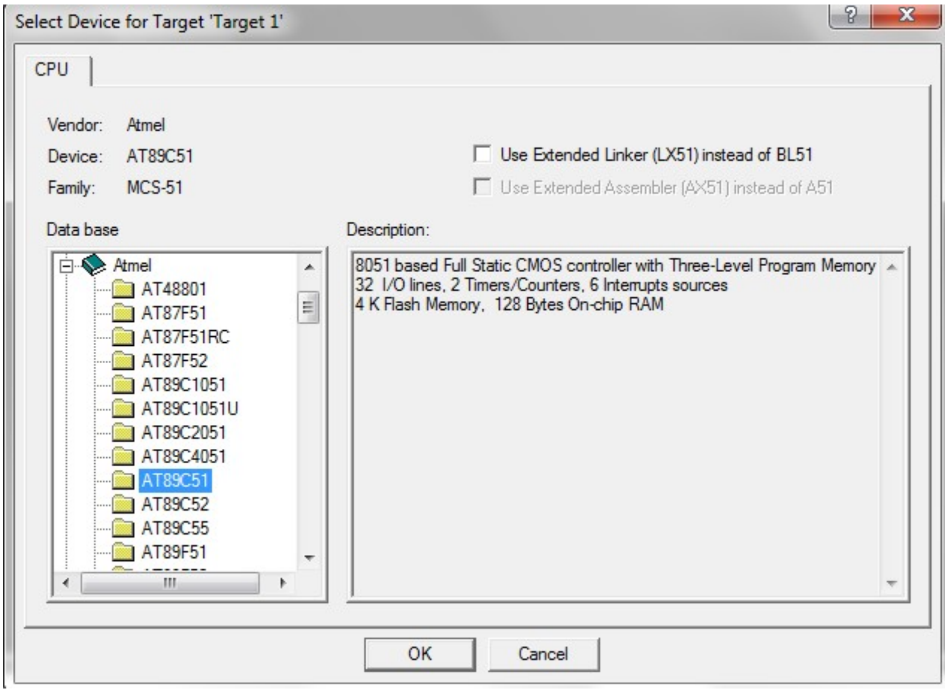


Starting a New Project

3. Select New Project from the Project Menu.
4. Name the project *Toggle*
5. Click on the Save Button

The device window will be displayed.

6. Select the part you will be using to test with. For now we will use the Atmel's AT89C51



Select Device for Target 'Target 1'

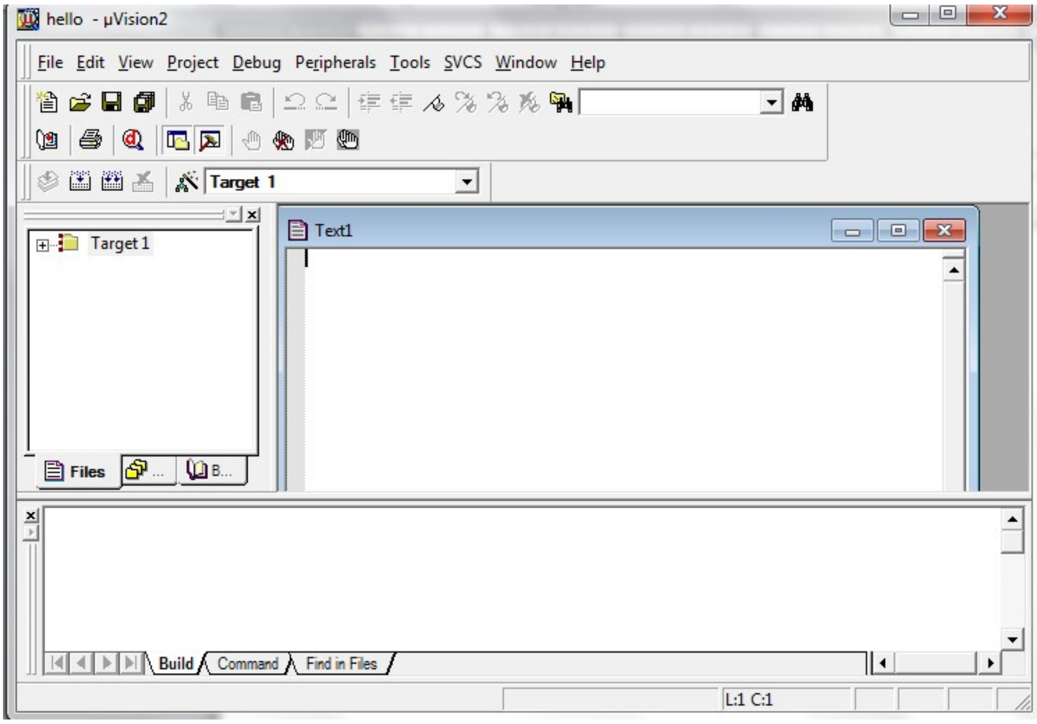
Vendor: Atmel
Device: AT89C51
Family: MCS-51

□ Use Extended Linker (LX51) instead of BL51
□ Use Extended Assembler (AX51) instead of A51

Description:
8051 based Full Static CMOS controller with Three-Level Program Memory
32 I/O lines, 2 Timers/Counters, 6 Interrupts sources
4 K Flash Memory, 128 Bytes On-chip RAM

Creating Source File

7. Click File Menu and select New.
8. A new window will open up in the KeilIDE.
9. Copy the code given below in assembly language in the new window. This code will toggle Port 1 and Port 2 continuously.
10. Click on the file menu and select Save.
11. Name the file Toggle.asm

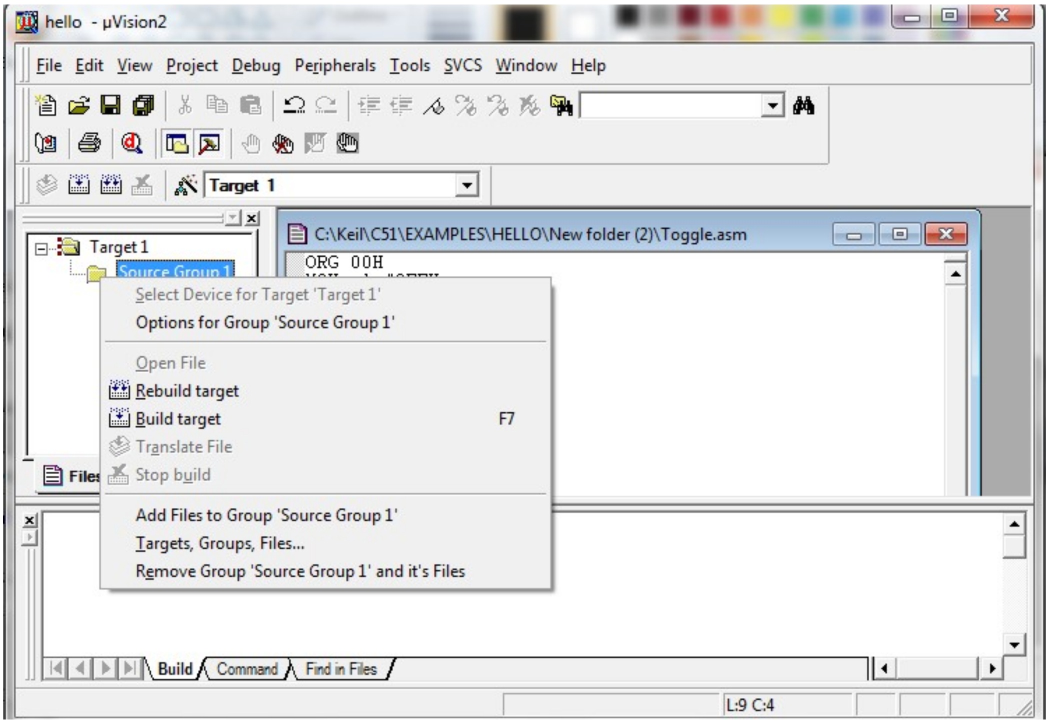


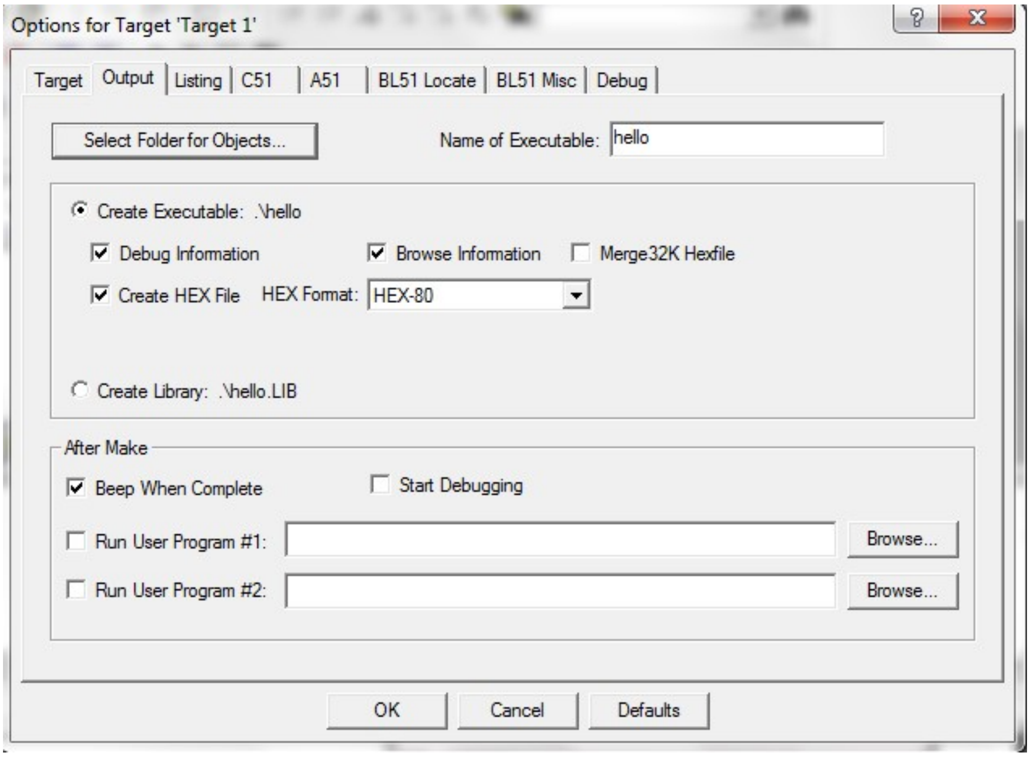
Adding File to the Project

12. Expend Target1 in the tree menu.
13. Click on Source Group1 and select Add file to Group 'Source Group1'.
14. Change file type to Asm Source File (*.a, *.src, *.asm).
15. Click on Toggle.asm and select Add button.
16. Click Close button.
17. Expend the Source Group 1 to ensure that the file was added.

Creating HEX file

18. Select on Target 1 in tree menu.
19. Click on Project menu and select Options for Target 1.
20. Select Output tab and then select Create Hex File check box.
21. Click Ok button.


hello - µVision2
File Edit View Project Debug Peripherals Tools SVCS Window Help
Target 1
Target 1
Source Group 1
C:\Keil\C51\EXAMPLES\HELLO\New folder (2)\Toggle.asm
ORG 00H
Select Device for Target 'Target 1'
Options for Group 'Source Group 1'
Open File
Rebuild target
Build target F7
Translate File
Stop build
Add Files to Group 'Source Group 1'
Targets, Groups, Files...
Remove Group 'Source Group 1' and it's Files
Files
Build
Command
Find in Files
L:9 C:4



Name of Executable: hello

● Create Executable: .\hello

☑ Debug Information ☑ Browse Information □ Merge32K Hexfile

☑ Create HEX File HEX Format: HEX-80

○ Create Library: .\hello.LIB

After Make

☑ Beep When Complete □ Start Debugging

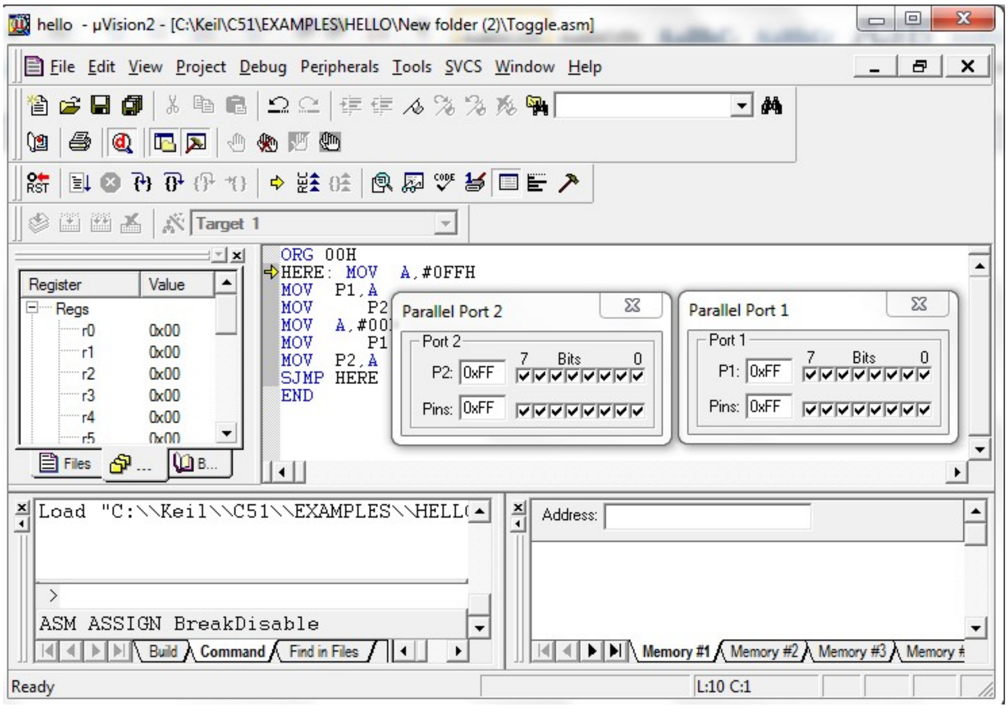
□ Run User Program #1:

□ Run User Program #2:

Testing Program in Debugger

22. Click on the Project menu and select Build all Target Files.
23. In the build window it should report ‘0 Error(s) and 0 Warnings’

24. Click on Debug menu and select ‘Start/Stop Debug Session’.
25. Click on Peripherals and select I/O Ports.
26. Select Port 1 and Port 2.
27. Step through the code by pressing F11 on the keyboard. The Port Boxes change with passing instructions.
28. To exit out, click on Debug menu and select ‘Start/Stop Debug Session’.


hello - µVision2 - [C:\Keil\C51\EXAMPLES\HELLO\New folder (2)\Toggle.asm]
File Edit View Project Debug Peripherals Tools SVCS Window Help
Target 1
Files
B...
Build
Command
Find in Files
Memory #1
Memory #2
Memory #3
Memory #
Ready
L:10 C:1

```
ORG 00H
HERE: MOV  A,#0FFH
MOV  P1,A
MOV     P2
MOV  A,#00
MOV     P1
MOV  P2,A
SJMP HERE
END
```

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00

Parallel Port 2

Port 2

P2: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑

Parallel Port 1

Port 1

P1: 0xFF 7 Bits 0 ☑☑☑☑☑☑☑☑

Pins: 0xFF ☑☑☑☑☑☑☑☑

```
Load "C:\\Keil\\C51\\EXAMPLES\\HELL(
>
ASM ASSIGN BreakDisable
```

Address:

Program Code in Assembly

```
ORG 00H
MOV A,#0FFH
MOV P1,A
MOV P2,A
MOV A,#00H
MOV P1,A
MOV P2,A
SJMP HERE
END
```

Exercise

Implement the assembly code for toggling in C and test it on Keil IDE.

Lab No.2 Interfacing LEDs to 8051and displaying Binary Patterns

Lab Task

Interface Pull-up resistors with Port0 on starter circuit and write a program in assembly language and C language to display the following patterns on Port.

```
00000001
00000010
00000100
…
…
10000000
00000001
00000010
```

Objectives of the Lab

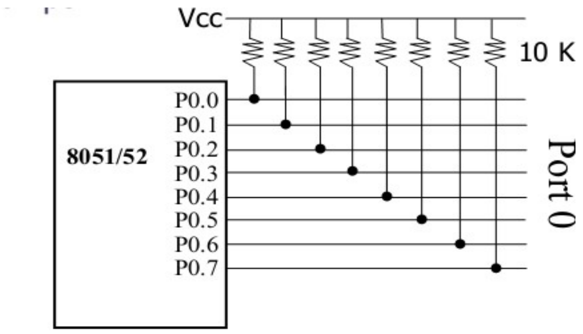
- To understand handling different ports
- Introduce Loops
- To introduce delays

Hardware & software requirement

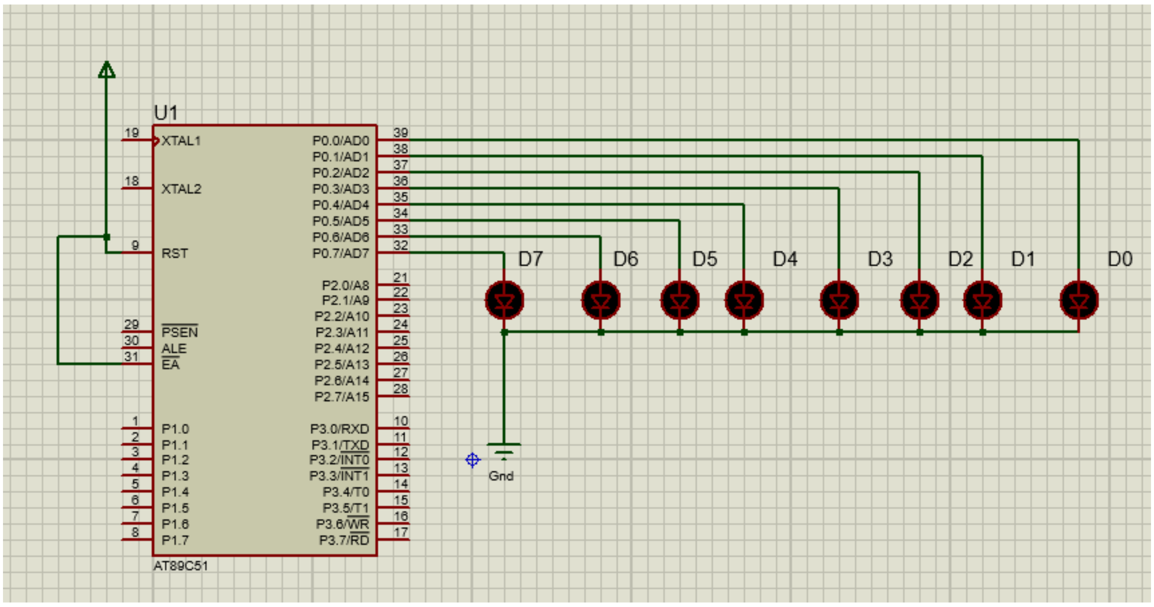
- Proteus
- 8051 Trainer Kit
- 8051 Assembler (Keil µVision).
- 8051 Programmer.
- Resistors
- LEDs for display.

Unlike P1, P2 and P3, the P0 is an open drain. Therefor each pin of P0 must be connected externally to a $10K\Omega$ pull-up resistor.

Interfacing Pull-up Resistors with 8051



Circuit Diagram



Program Code in Assembly

```
ORG 00H
MOV P0, #0FFH
BACK: MOV P0, #01H
ACALL DELAY
MOV P0, #01H
 ACALL DELAY
MOV P0, #03H
ACALL DELAY
MOV P0, #08H
ACALL DELAY
MOV P0, #10H
```

```
ACALL DELAY
MOV P0, #20H
ACALL DELAY
MOV P0, #40H
 ACALL DELAY
MOV P0, #80H
ACALL DELAY
MOV P0, #40H
ACALL DELAY
MOV P0, #20H
ACALL DELAY
MOV P0, #08H
 ACALL DELAY
MOV P0, #04H
ACALL DELAY
MOV P0, #02H
ACALL DELAY
MOV P0, #01H
ACALL DELAY
SJMP BACK DELAY:
MOV R1, #255H
HERE: DJNZ R1, HERE
RET
END
```

Program Code in C

```
#include <REG52.H>
void delay(void)
{
  int i;  for (i=0;i<30000;i++);  for
           (i=0;i<30000;i++);
}
void main (void) // Starting of the main function.
{
  while(1)
           {
                      P0 = 0x01;
delay();
                      P0= 0x02;
                      delay();
                      P0=0x04;
                      delay();
                      P0=0x08;
                      delay();
                      P0=0x10;
                      delay();
                      P0=0x20;
                      delay();
```

```
        P0=0x40;
        delay();
        P0=0x80;
        delay();
        P0=0x40;
        delay();
        P0=0x20;
        delay();
        P0=0x10;
        delay();
        P0=0x08;
        delay();
        P0=0x04;
        delay();
        P0=0x02;  delay();
        P0=0x01;  delay();
    }
}
```

Lab No.3 Interfacing Push buttons to 8051 Development kit using Polling

Lab Task

Write a C language program which will continuously read values from Port 1 of 8051 and display them on LEDs interfaced to Port 0 of 8051.

Objectives of the Lab

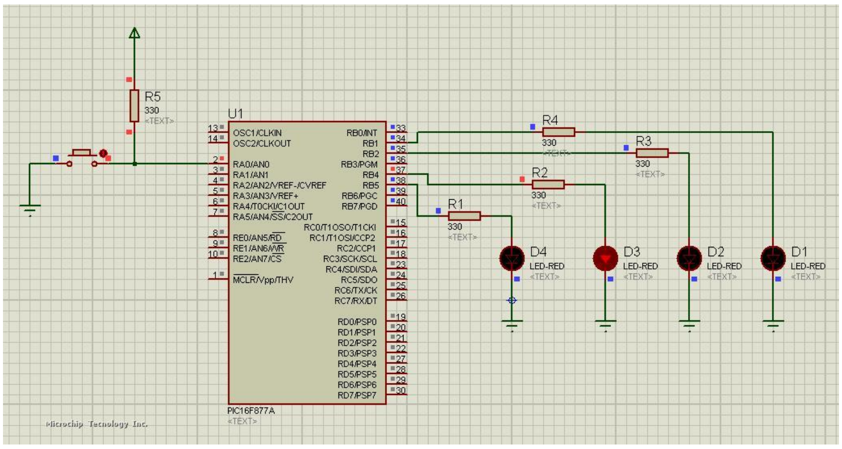
- To understand the operation of DIP Switches
- To interface and program DIP Switches

Hardware & software requirement

- 8051 Starter Circuit from Previous lab.
- 8051 Assembler (Keil µVision).
- 8051 Programmer.
- DIP Switches and LEDs.

It is always best connecting the switch to ground with a pull-up resistor. When the switch is open, the $10K\Omega$ resistor supplies very small current needed for logic 1. When it is closed, the port pin is short to ground. The voltage is 0V and the entire sinking current requirement is met, so it is logic 0.

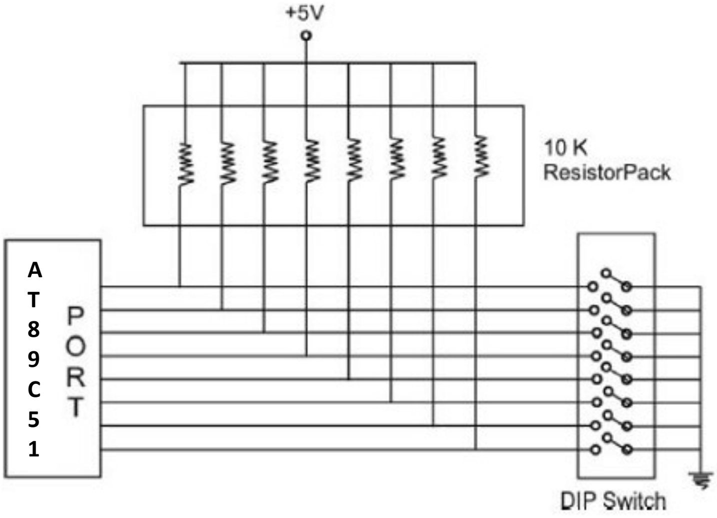
Circuit Diagram



Program Code in C

```
#include<REG52.H> void
main(void)
{
          While(1)
          {
                    Char N1;
                    P1=0xFF;    N1=P1;
                    P0=N1;
          }
}
```

Interfacing DIP Switches with 8051



LAB TASK

- Write a program to input two numbers, find their sum and display the result.
- Write a program to find whether the input number is a prime number or not?