Lab 13: Programming 8051 Microcontroller

EE222: Microprocessor Systems

1 Administrivia

Learning Outcomes

By the end of this lab you will be able to;

1. Write an assembly language code for x86 architecture
2. Create .hex file using Keil µVision IDE
3. Burn your code onto 8051 microcontroller to run your application successfully

Deliverable

You are required to submit

- Appropriately Commented Code
- Explicit Calculations for Timer Values
- Issues in Developing the Solution and your Response

in the beginning of next lab

Hardware Resources

- This is a Proteus based simulation lab and does not require any hardware.

Introduction

The AT89C51 is an age old 8-bit microcontroller from the Atmel family. It works with the popular 8051 architecture and hence is used by most beginners till date. It is a 40 pin IC package with **4 Kb flash memory**. It has **four ports** and all together provides **32 Programmable GPIO pins**. It does not have in-built ADC module and supports only USART communication. Figure 1 shows the pin-out of 8051 microcontroller.

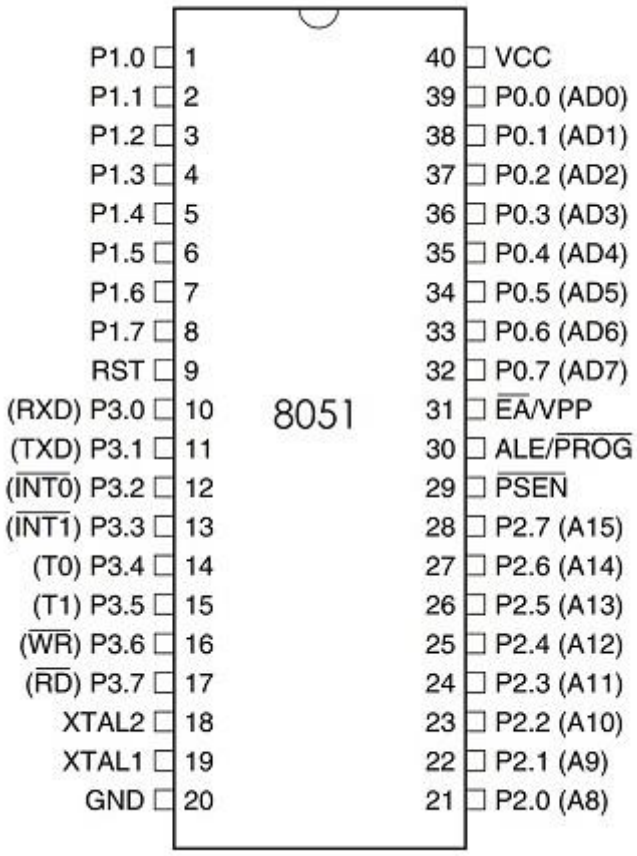

Figure 1. 8051 Microcontroller Pin-out

In order to program the Atmel microcontroller we will need

- **IDE** (Integrated Development Environment), where the programming takes place.
- **Compiler**, where our program gets converted into MCU readable form called HEX files.
- **IPE** (Integrated Programming Environment), which is used to dump our hex file into our MCUs.

We will be using **Keil uVision IDE** throughout this lab. Use the provided "*How to Program with Keil*" file to familiarize yourself with the Keil environment.

LED Interfacing with 8051

For any microcontroller, LED interfacing is equivalent to "Hello world!" program to see a physical output. We will begin by learning how to interface an LED with 8051 microcontroller.

Figure 2 shows the circuit diagram of two LEDs interfaced with AT89C51 on two different Port Pins. One is connected to Pin7 of Port0 and second one is connected to Pin0 of Port2. Both LEDs are driven through resistors to limit the current.

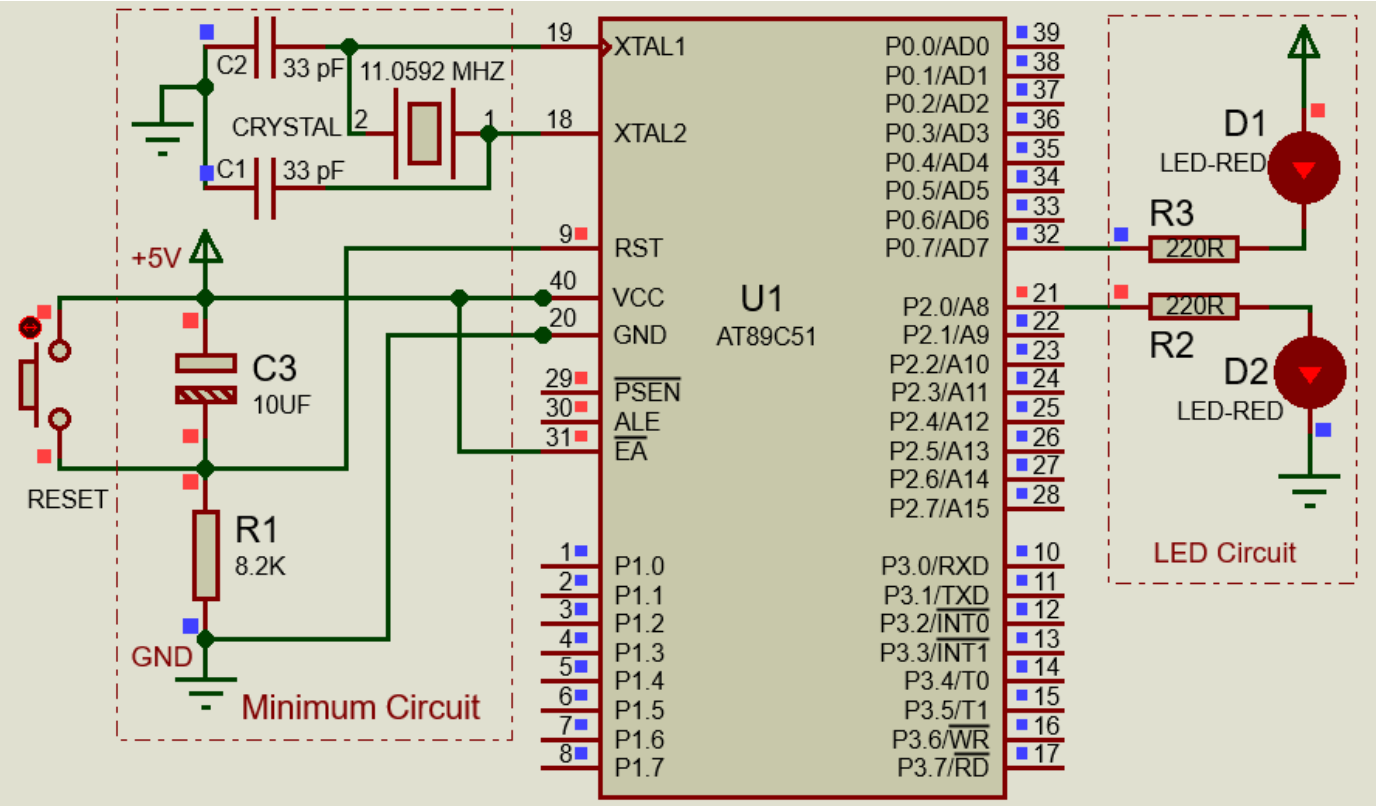

Figure 2. LED interfacing with 8051

Note that the "Minimum Circuit" on the left is required to provide external clock to the 8051 microcontroller.

Lab Task

Build the following code in Keil to generate a hex file. Understand its functionality. Then construct the circuit of figure 2 in Proteus and burn the created hex file. Observe the output.

```
        ORG     00H             ; PROGRAM STORE LOCATION
        MOV     P0,#00H         ; REFRESHING PORT-0
        MOV     P1,#00H         ; REFRESHING PORT-1
        MOV     P2,#00H         ; REFRESHING PORT-2
        MOV     P3,#00H         ; REFRESHING PORT-3

AGAIN:
        CLR     P0.7            ; PORT-0 PIN-7 LOW (LED1-ON)
        SETB    P2.0            ; PORT-2 PIN-0 HIGH (LED2-ON)
        LCALL   DELAY           ; CALL DELAY SUBROUTINE
        SETB    P0.7            ; PORT-0 PIN-7 HIGH (LED1-OFF)
        CLR     P2.0            ; PORT-2 PIN-0 LOW (LED2-OFF)
        LCALL   DELAY           ; CALL DELAY SUBROUTINE
        LJMP    AGAIN           ; LOOP FOREVER TO LABEL AGAIN

DELAY:                          ; DELAY SUBROUTINE
        MOV     R7,#50
L3 :    MOV     R6,#100
L2 :    MOV     R5,#100
L1 :    DJNZ    R5,L1
        DJNZ    R6,L2
        DJNZ    R7,L3
        RET                     ; END DELAY SUBROUTINE

        END
```

ADC Interfacing with 8051

Refer to the following link to interface an AD0804 ADC chip with 8051 microcontroller to read analog signals. As a part of **lab task**, build the following circuit in Proteus and load the provided assembly code to faithfully run the analog-to-digital conversion.