PR1

LED Blinking

Version 1.0

April 2009

Cytron Technologies Sdn. Bhd.

OVERVIEW

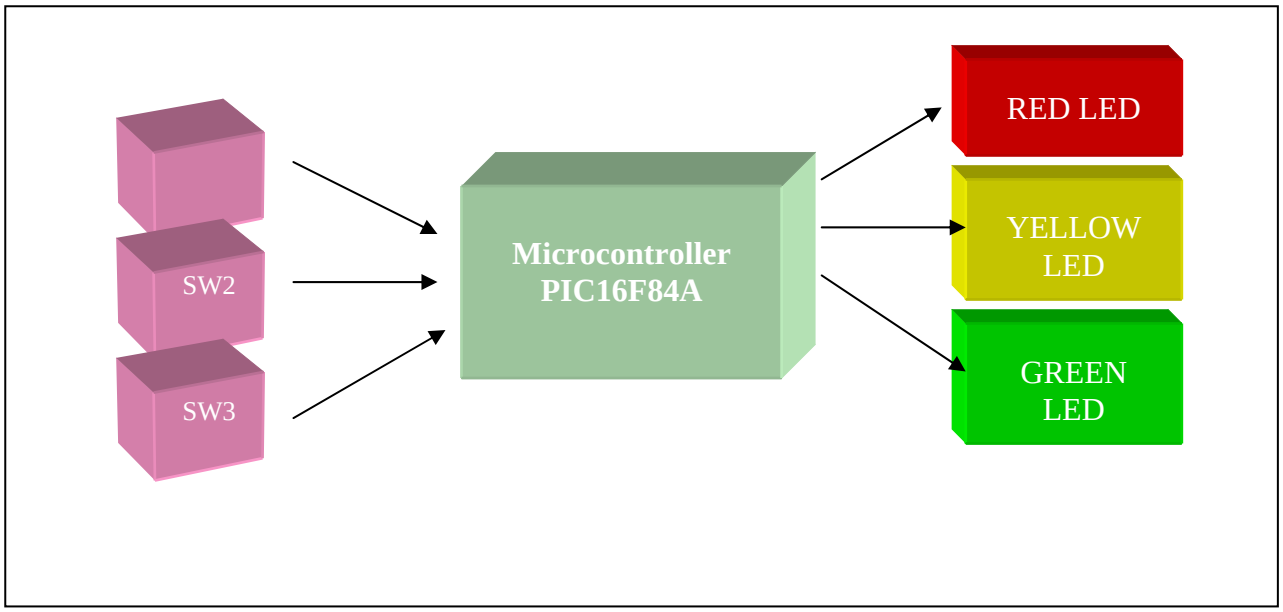
This document describes the development of Cytron Technologies DIY (Do It Yourself) Project PR1. The most efficient way to learn PIC is hands-on. PR1 provide 3 simple steps for beginner to learn PIC with hands-on. A set of electronic components is needed in building a circuit for this project. Circuit schematic and software will be provided.

FEATURES

PIC16F84A

- 8 bit microcontroller with 13 I/O
- Operate with 5V supply.
- Operating speed 20MHz

SYSTEM OVERVIEW



GENERAL DESCRIPTION

LED blinking is the simplest PIC microcontroller preject. With just 3 simple steps, user can make the LEDs blinking by using the PIC16F84A. This project is a very good kick-start to learn about programming microcontroller for the beginners. Information which includes sample source code, schematic and detail description can be downloaded from
www.cytron.com.my

PIC16F84A

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 18- or 20-pin package. Features of the device:

- 64 bytes of EEPROM data memory
- 68 bytes of Data RAM
- 14-bit wide instruction words
- 8-bit wide data bytes

There are also 13 I/O pins that are user-configured on a pin to pin basis. Some pins are multiplexed with other device functions. These functions include:

- External interrupt
- Change on PORTB interrupt
- Timer0 clock input.

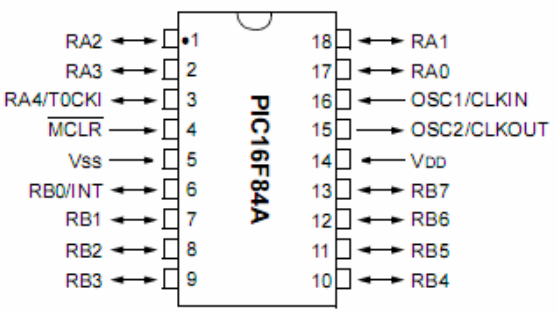


Figure 1

HARDWARE

This project will require following hardware:

PR1-A
a. 1 x PIC16F84A
b. Breadboard
c. 9V battery
d. Jumper wire
e. Related electronic components

PR1-B
a. 1 x PIC16F84A
b. 1 x IC Socket
c. Donut Board
d. 9V Battery
e. Related electronic components

Please refer to Component list to see related electronic components. For a beginner which not familiar with the kind of schematic, please refer Appendix A for the circuit diagram of PR1.

Power supply for the circuit

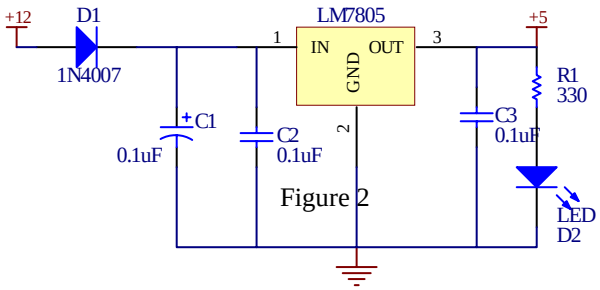


Figure 2

This project will use 6V-9V battery (included in the DIY project set) to power up the circuit. Higher input voltage will result in more heat dissipated at LM7805 voltage regulator. Typical voltage is 6V although LM7805 will still dissipate some heat at 6V. Refer to Figure 2, the D1 is use to protect the circuit from wrong polarity supply. C1 and C2 is use to stabilize the voltage at the input side of the LM7805 voltage regulator, while the C3 is use to stabilize the voltage at the output side of the LM7805 voltage supply. A LED is connected indicate the power status of the circuit. Resistor R1 is used to limit current flow through LED.

Push Button as input for PR1

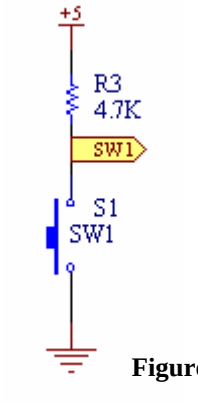


Figure 3

The connection of the push button to the I/O pin is shown in Figure 3. The I/O pin should be pull up to 5V using a resistor (with value range 1K-10K) and this configuration will result in an active-low input. When the button is being pressed, reading of this input pin will be logic 0, while when the button is not pressed, reading of this input pin will be logic 1. The capacitor is used to filter out the noise.

LED as output for PR1

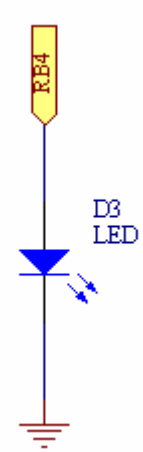
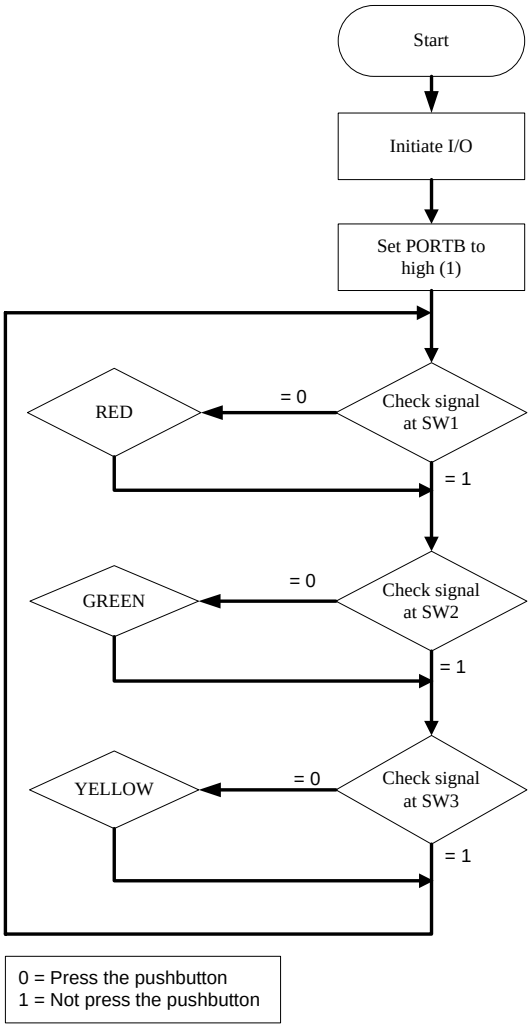


Figure 4

A LED is connected at output pin of PIC as Output device. The connection for the LED to output pin is shown in Figure 4. When the output is logic **1**, the LED will **ON**, while when the output is logic **0**, the LED will **OFF**.

SOFTWARE
Flowchart:



Description of Source Code

This section will discuss the Sample Source Code of PR1. They are 3 Sample Source code provided in PR1. This description is refer to PR1_2.asm Sample Source Code.

```
; initialize of your PIC

    BSF     STATUS,5
    CLRF    TRISB
    BSF     TRISA,2
    BSF     TRISA,3
    BSF     TRISA,4
    BCF     STATUS,5

    MOVLW   B'11111111'
    MOVWF   PORTB
```

Listing 1

Listing 1 show s the initial configuration of PIC. Port B is defined as output by clear the TRISB register. RA2, RA3 and RA4 are defined as input. All 8 pin in PORTB is set to high with sending binary 11111111 to PORTB. User may see the comment at the Sample Source Code.

```
;the main program begin here

START
        BTFSS       PORTA,2
        CALL        RED
        BTFSS       PORTA,3
        CALL        GREEN
        BTFSS       PORTA,4
        CALL        YELLOW
        GOTO        START
```

Listing 2

Let's look at Listing 2, the program starts with check signal at pushbutton1 (SW1). If SW1 is pressed, program will execute the operation at RED subroutine and following line will skip. If SW1 is unpressed, program will check signal at pushbutton2 (SW2). If SW2 is pressed, program will execute the operation at GREEN subroutine and following line will skip. If SW2 is unpressed, program will check signal at pushbutton3 (SW3). If SW3 is pressed, program will execute the operation at YELLOW subroutine. If SW3 is unpressed, program will keep looping to check the pushbuttons signal. User may see the comment at the Sample Source Code.

```
RED
        MOVLW       D'20'
        MOVWF       COUNT
        BCF         PORTB,6
        CALL        DELAY
        BSF         PORTB,6
        CALL        DELAY
        DECFSZ      COUNT
        GOTO        $-5
        RETURN
```

Listing 3

Listing 3 is RED subroutine. Program will execute this operation if SW1 is pressed. Program will store initial value of count at COUNT. RED LED will OFF and after some delay it will ON. Value of count will decrease one and program will check again the value of count. If value of count not equal to zero, it will lead the program to 5 line above which is OFF the LED and then it will execute the following line. It will repeat until value of count reach zero. If value of count reach zero, program will skip the nest.

```
GREEN
        MOVLW       D'20'
        MOVWF       COUNT
        BCF         PORTB,5
        CALL        DELAY
        BSF         PORTB,5
        CALL        DELAY
        DECFSZ      COUNT
        GOTO        $-5
        RETURN
```

Listing 4

Listing 3 is GREEN subroutine. Program will execute this operation if SW2 is pressed. Program will store initial value of count at COUNT. GREEN LED will OFF and after some delay it will ON. Value of count will decrease one and program will check again the value of count. If value of count not equal to zero, it will lead the program to 5 line above which is OFF the LED and then it will execute the following line. It will repeat until value of count reach zero. If value of count reach zero, program will skip the nest.

```
YELLOW
        MOVLW       D'20'
        MOVWF       COUNT
        BCF         PORTB,4
        CALL        DELAY
        BSF         PORTB,4
        CALL        DELAY
        DECFSZ      COUNT
        GOTO        $-5
        RETURN
```

Listing 5

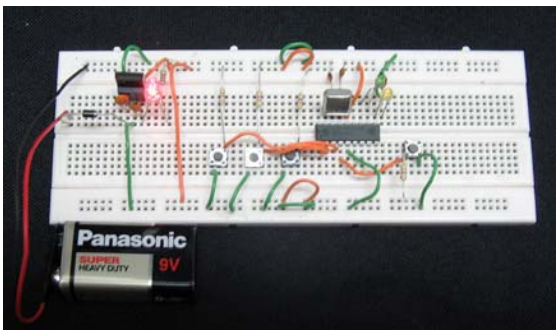
Listing 5 is YELLOW subroutine. Program will execute this operation if SW1 is pressed. Program will store initial value of count at COUNT. YELLOW LED will OFF and after some delay it will ON. Value of count will decrease one and program will check again the value of count. If value of count not equal to zero, it will lead the program to 5 line above which is OFF the LED and then it will execute the following line. It will repeat until value of count reach zero. If value of count reach zero, program will skip the nest.

GETTING START

User can obtain the hardware set for this project (PR1) either by online purchasing (www.cytron.com.my) or purchase it from Cytron Technologies Shop. They are 2 types of PR1 which are PR1-A and PR1-B. For PR1-A, no soldering is required and for PR1-B soldering is required. User may choose one of them.

1. Once user has the hardware set, user can start plug in the electronic components on the protoboard. Protoboard is a fundamental tool for beginner to develop their mini project, where no permanent soldering is required. However, for user who bough PR1-B, soldering process is needed. Plug the electronic components on donut board and solder the electronic components one by one according to the schematic. Schematic is provided in Cytron website.

2. When plug the electronic components either on protoboard or donut board, ensure the components value and polarity is correct.



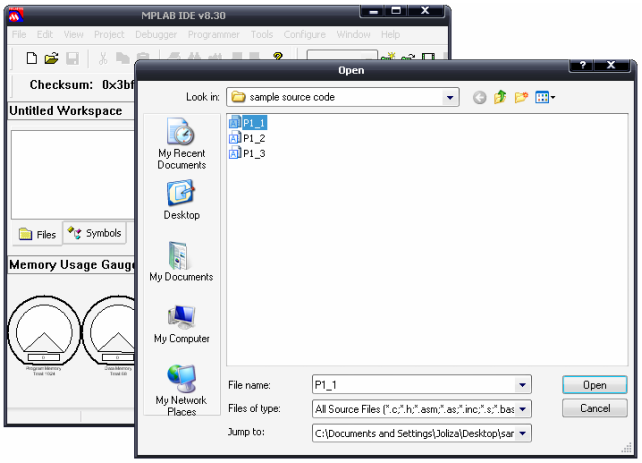
Tips : Start soldering with component like jumpers, resistor, diode which are lower in height, followed by capacitor, LED, button and lastly the connector or other components which are taller.

Warning: Before the plug in the battery (Power), make sure the polarity is correct to prevent permanent damage to the circuit. Electrolytic capacitor mounted in wrong polarity may cause small explosion.

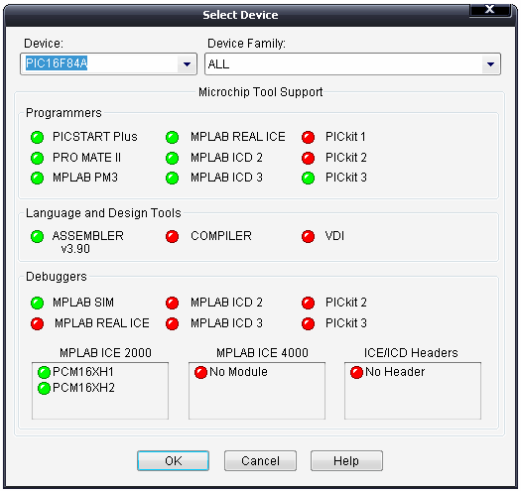
3. Please download the necessary files and documents from Cytron Technologies website. These included documentation, sample source code, schematic, component list and software.

4. The next step is to install MPLAB IDE and Hi-Tech CPRO into a computer. Please refer MPLAB installation step document to install the software.

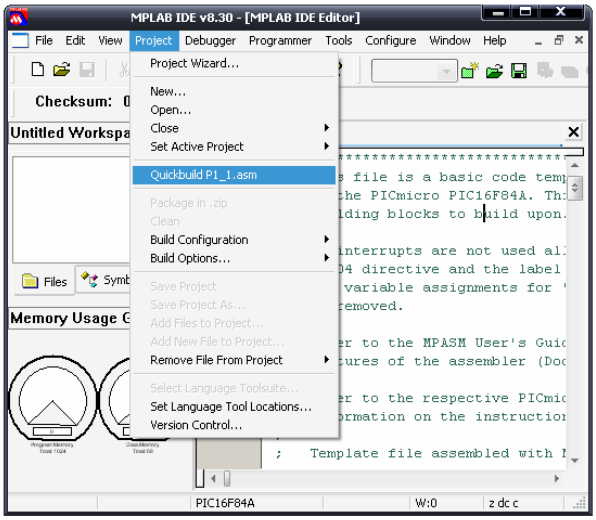
5. After the installation is complete, open the MPLAB program. Open the PR1_1.asm file that user download from Cytron website. To open the file, click *file>open>*browse to *PR1_1.asm file.*



6. From MPLAB menu bar, click *Configure> Select device> Select PIC16F84A>OK.*



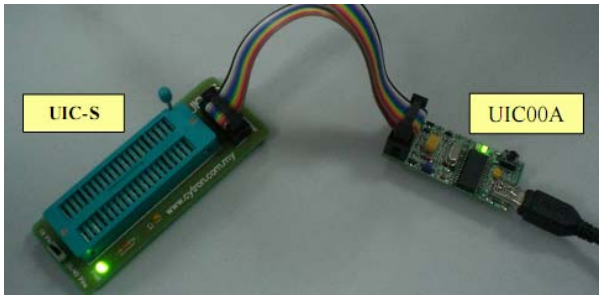
7. Click *Project>Quickbuild P1_1.asm* if user want to compile the program.



8. If no error occurred, MPLAB will auto generate a hex file at the location user put the asm file. Hex file is the file that needed for programmer to burn into PIC.

9. User need PIC programmer to load hex file into PIC. The programmer is not include in this project. User need to buy separately. The low cost programmer that most suitable for students is UIC-S and UIC00A. User need both programmer in this project to load hex file into PIC.

10. Please refer UIC00A User's Manual on how to load hex file into PIC using UIC00A with UIC-S. UIC00A User's Manual and related software is provided in Cytron website.



11. Figure above is hardware connection of UIC00A and UIC-S. The end of USB cable from UIC00A is connect to computer or laptop. PIC16F84A is plug in on UIC-S zip socket.

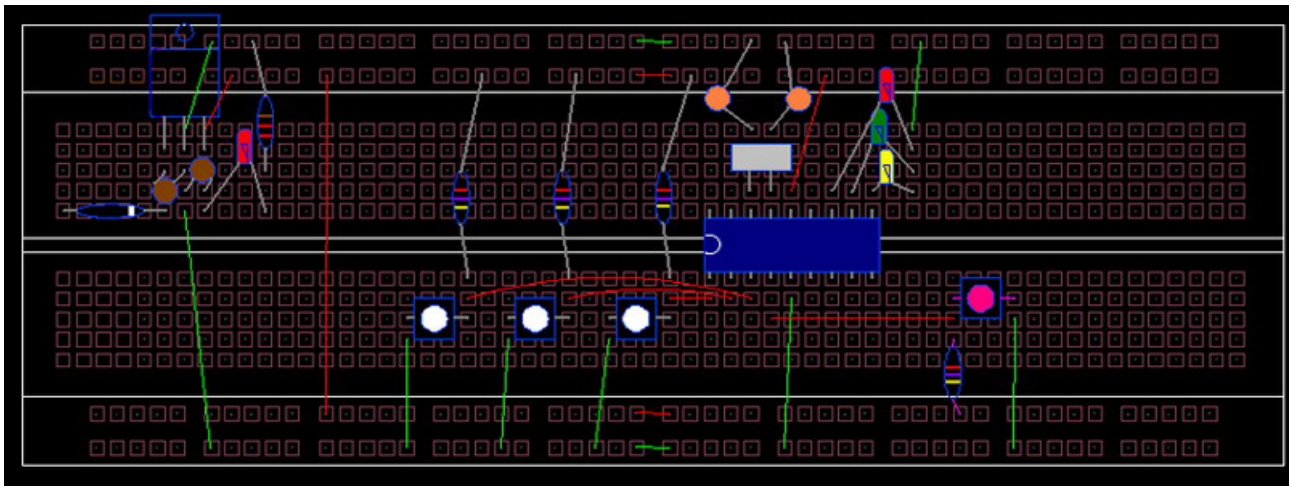
12. Easy and simple steps. User has successfully load the program into PIC.

13. User may modify the program. After modified the program repeat steps 6-9 to compile and load the hex code into PIC.

WARRANTY

No warranty will be provided as this is DIY project. Thus, user is advice to check the polarity of each electronic component before solder it into board.

Appendix A
Circuit Diagram:



Prepared by
Cytron Technologies Sdn. Bhd.
19, Jalan Kebudayaan 1A,
Taman Universiti,
81300 Skudai,
Johor, Malaysia.

Tel: *+607-521 3178*
Fax: *+607-521 1861*

URL: *www.cytron.com.my*
Email: *support@cytron.com.my*
sales@cytron.com.my