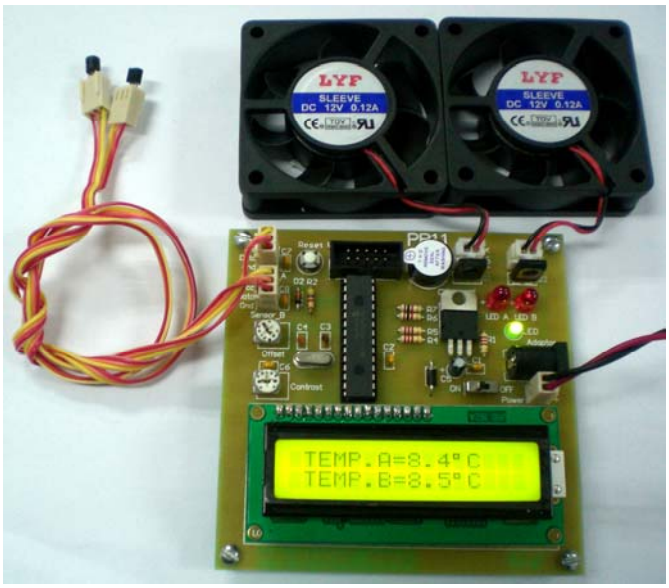
PR11

Temperature Control System using LM35



Version 1.2

Aug 2008

Cytron Technologies Sdn Bhd.

OVERVIEW

This document describes the development of Cytron Technologies DIY (Do It Yourself) Project No.11 (PR11). This project will use PIC16F876A to control NPN power transistor (BD135) further drive DC brushless fans, LEDs and buzzer when the certain temperature was detected. The value of temperature always displayed on a LCD screen. Circuit schematic and PIC source code will be provided.

FEATURES

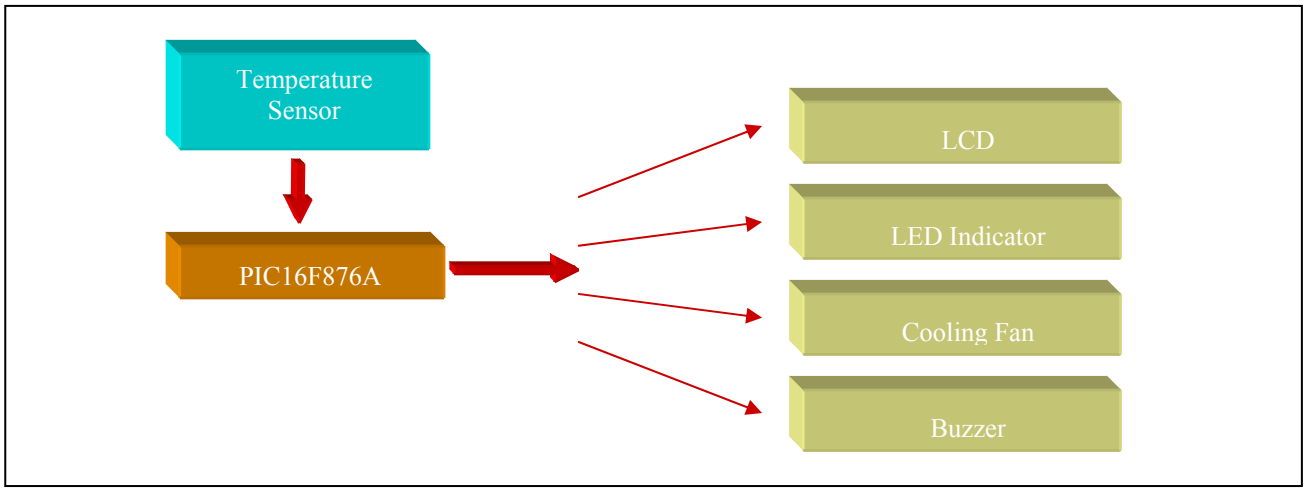
PIC controlled DC brushless fan, LEDs and buzzer
- PIC16F876A with internal ADC read the analog voltage from LM35 (Temperature sensor) and display it on LCD display

BD135 power transistor
- Capable of driver DC brushless fan (0.12A)

Temperature sensor (LM35 DZ)
- Sensor gain, 10mV/°C (average slope)
- 0 - 100°C
- LCD display (always display the value of the temperature)

Fans, LEDs and buzzer are activated when the certain temperature was detected.

SYSTEM OVERVIEW



GENERAL DESCRIPTION

PIC has been used to control the LED or buzzer directly for some previous PRs. But in this project, PIC cannot directly activate the DC brushless fan because it has not enough current. NPN power transistor (BD135) is required, so that DC brushless fan can be controlled by PIC. PIC16F876A will read the analog voltage (using ADC) from LM35 temperature sensor and display it on LCD display. A buzzer, LEDs or fans will be activated when temperature reach a certain temperature value.

Power Transistor (BD135)

BD135 is used for controlling the DC brushless fan with sufficient current. Following Figure 1 shows the pin diagram of BD135.

PINNING

PIN	DESCRIPTION
1	emitter
2	collector, connected to metal part of mounting surface
3	base

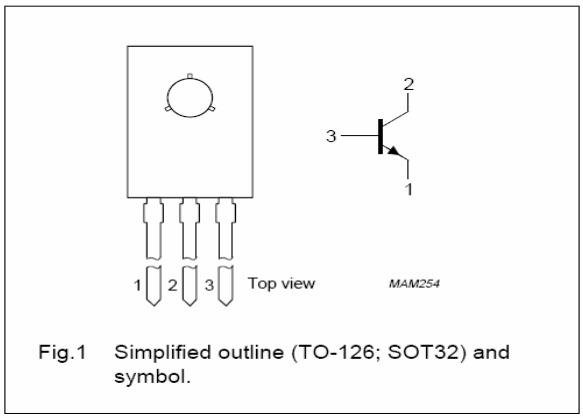


Fig.1 Simplified outline (TO-126; SOT32) and symbol.

Figure 1

Temperature Sensor (LM35)

In this project, two LM35s are used for two difference temperature. Vs of the LM35s are given 5V and the Vout pins are connected to AN0 and AN1 (PIC16F876A) separately.

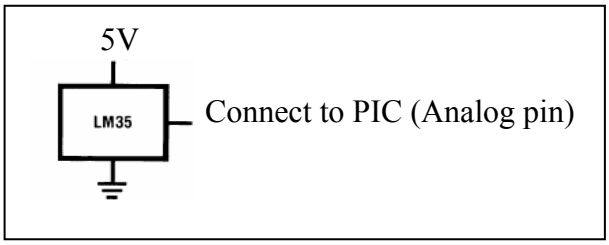


Figure 2

Since the sensor gain (average slope) of the LM35 is $10mV/^{\circ}C$ and ADC has 10 bit ($2^{10} \approx 1000$), so pin 5 (Vref) from PIC16F876A must be given for **1V** by using the voltage divider concept. For preventing the offset, voltage for pin 5 (Vref) should be adjustable (using Preset) as shown in Appendix A.

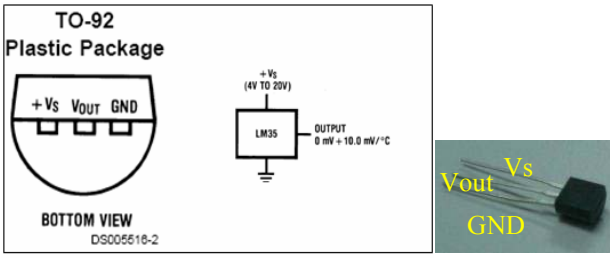


Figure 3

PIC16F876A (Microcontroller)

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 28-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices.
The PIC16F876A features:

- 256 bytes of EEPROM data memory
- Self programming
- An ICD
- 2 Comparators
- 5 channels of 10-bit Analog-to-Digital (A/D) converter
- 2 capture/compare/PWM functions
- The synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I²C™) bus
- A Universal Asynchronous Receiver Transmitter (UART)

All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

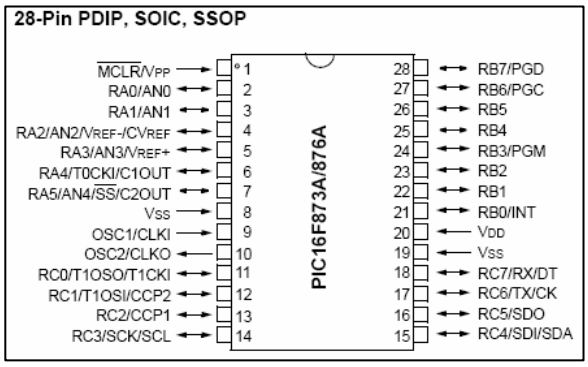


Figure 4

Figure 4 shows the pin diagram for PIC16F876A. For more detail, please download the datasheet from microchip web site at: http://www.microchip.com

HARDWARE

This project will require following hardware:

a. 1 x PIC16F876A
b. 1 x PR11 Printed Circuit Board (PCB)
c. 1 x LCD
d. 1 x Buzzer
e. 2 x Temperature Sensor (LM35)
f. 2 x DC Brushless Fan
g. Other related electronic components

Please refer to the schematic diagram of PR11. The schematic is provided free and therefore Cytron Technologies will not be responsible for any further modification or improvement.

Interface PIC16F876A with Temperature Sensor (LM35)

Signal pin (Vout) from LM35 can be connected to either one of analog input pin (AN0-AN4) except AN3 (pin 5) but make sure the ADC configuration is correct according to the Figure 11 (Software section). In fact, pin 5 (Vref+) from PIC should be given for **1V** but it may has offset, so a variable resistor (VR1) was installed for voltage adjusting. For more stability, user is recommended add a capacitor (104) between the analog signal and GND for every analog input such as signal from LM35 and variable resistor (VR1).

Interface PIC16F876A with LCD (2x16 character)

LCD used in this project is JHD162A, for other type of LCD, please refer to its data sheet.

Figure 5

The 16 header pin should be soldered to the LCD first. The following table shows the LCD (2x16 character) connection:

Pin	Name	Pin function	Connection
1	VSS	Ground	GND
2	VCC	Positive supply for LCD	5V
3	VEE	Contrast adjust	Connected to a preset for contrast adjusting
4	RS	Select register, select instruction or data register	RA2
5	R/W	Select read or write	GND
6	E	Start data read or write	RA5
7	DB0	Data bus pin	RC0
8	DB1	Data bus pin	RC1
9	DB2	Data bus pin	RC2
10	DB3	Data bus pin	RC3
11	DB4	Data bus pin	RC4
12	DB5	Data bus pin	RC5
13	DB6	Data bus pin	RC6
14	DB7	Data bus pin	RC7
15	LED+	Backlight positive input	5V
16	LED-	Backlight negative input	GND

Table 1

Figure 6 shows the schematic of the LCD display.

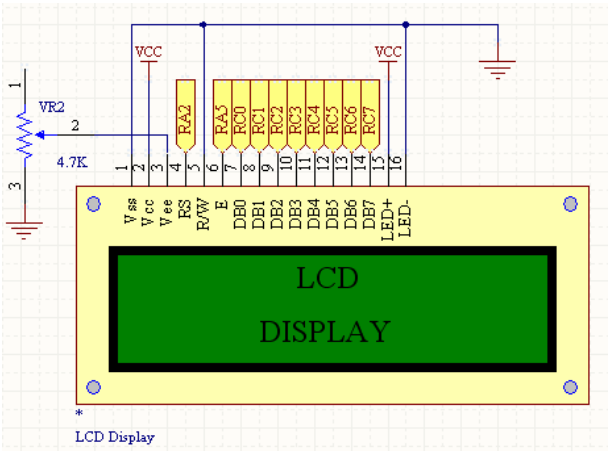


Figure 6

Power Supply for Circuit

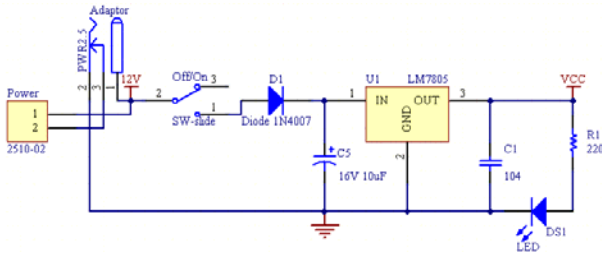


Figure 7

For this project, the voltage range of power source could be given for this circuit board is between 7V and 15V. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 12V. Anyhow, LM7805 will still generate some heat at 12V. There are two type of power connector on the circuit board, DC plug 'Adaptor' is for AC-DC adaptor and 2510-02 'Power' is for battery source. Normally AC to DC adaptor can be plugged to 'Adaptor' type connector. LM7805 (1A maximum) will regulate the given voltage to 5V (VCC) for supplying to the PIC16F876A and pull-up the push button (input). The purpose of using diode (D1) is for circuit protection in case the polarity of the power source is incorrect. Capacitor (C5) and capacitor (C1) is use to stabilize the voltage input and output of the LM7805. DS1 is a green LED (small) as power indicator.

ICSP for Programming PIC Microcontroller

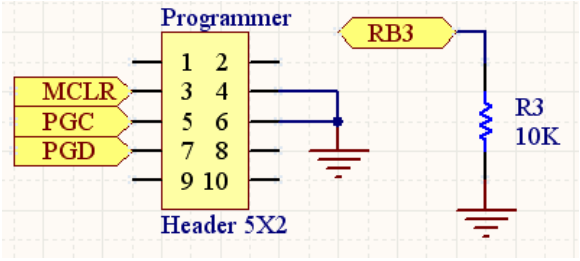


Figure 8

In Circuit Serial Programming (ICSP) is used for loading program in this project. ICSP gives you a convenience way to load program into PIC microcontroller without removing the PIC from the circuit board. So pin 1 (Vpp), pin 27 (PGC) and pin 28 (PGD) from PIC should be connected to Cytron USB In Circuit Programmer (UIC00A) through the external cable. Besides, GND from the circuit board also should be connected with GND from UIC00A and pin 24 (PGM) should be pulled to GND through a 10K resistor as shown in Figure 8. The programmer (UIC00A) is not included in DIY project set since it can be used several times for different project set. User can also choose other type of PIC programmer to load the program. Since the ICSP is used, three I/O pins (RB3, RB6 and RB7) cannot be used as input again but it still can be used for output.

Push Button as Input for PIC microcontroller

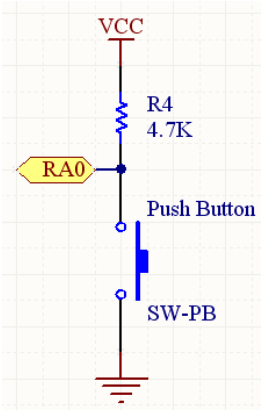


Figure 9

One I/O pin is needed for one push button as input for PIC microcontroller. The connection of the push button to the I/O pin is shown in Figure 9. The I/O pin should be pull up to 5V using a resistor (with value range 1K-10K) and this configuration will result an active-low input. When the button is being pressed, reading of I/O pin will be in logic 0, while when the button is not pressed, reading of that I/O pin will be logic 1.

LED as Output for PIC microcontroller

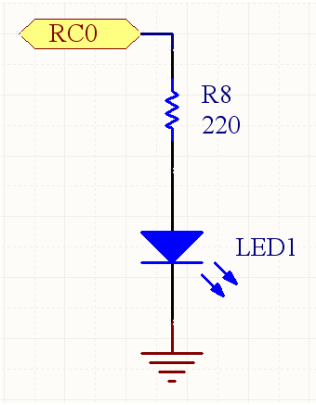


Figure 10

One I/O pin is needed for one LED as output for PIC microcontroller. The connection for a LED to I/O pin is shown in Figure 10. The function of R8 is to protect

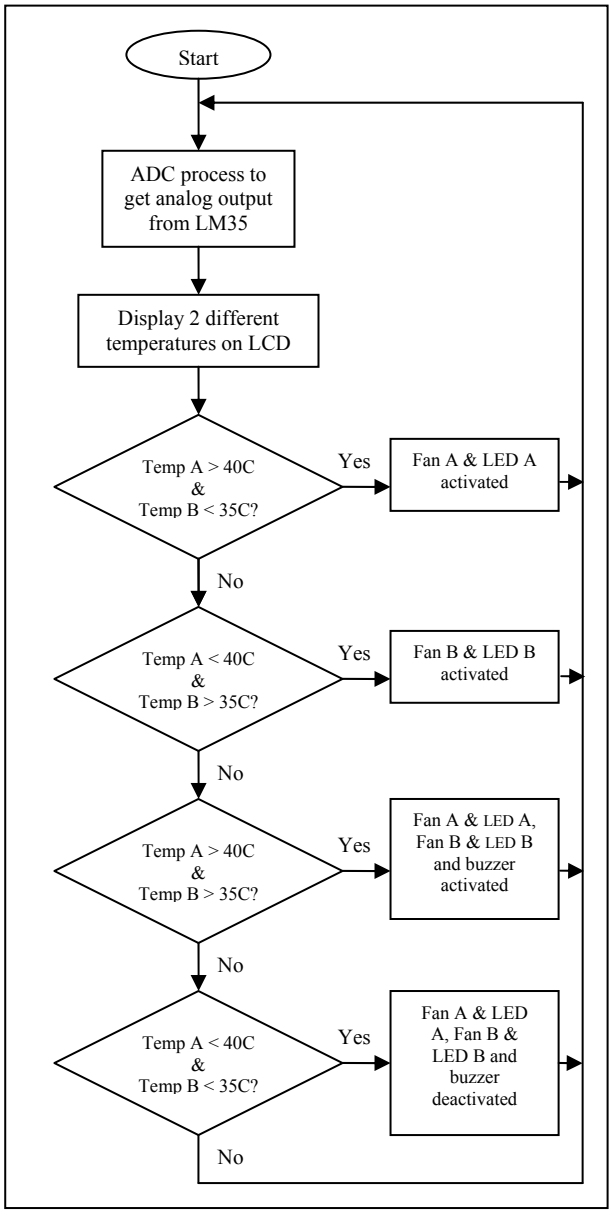
the LED from over current that will burn the LED. When the output is in logic 1, the LED will ON, while when the output is in logic 0, the LED will OFF.

Interface PIC16F876A with DC Brushless Fan

Since the current of I/O pin from PIC is limited to drive a DC Brushless Fan (0.12A), so Power Transistor (BD135) is required for giving current to it sufficiently. The maximum collector current, Ic of BD135 is 1.5A, which means the DC Brushless Fan greater than 1.5A cannot be driven.

SOFTWARE

Flow Chart:



Program

Please download the sample program from Cytron website (same directory as this DIY project)

The source code is provided free and Cytron Technologies will not be responsible for any further modification or improvement.

Analog to Digital Converter (A/D) Module

The Analog-to-Digital (A/D) Converter module has five inputs for PIC16F876A. The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.

The A/D module has four registers. These registers are:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)

The ADCON0 register, shown in Figure 11, controls the operation of the A/D module. For PR11, clock conversion Fosc/64 has been selected. Only two channels (AN0 &AN1) are used for two temperature sensors separately. ADCON0 should be 0b10000001 for channel 0 and 0b10001001 for channel 1.

The ADCON1 register, shown in Figure 12, configures the functions of the port pins. In this project, ADCON1 was set to 0b11000101. Right justified result format was selected (ADFM=1) as shown in Figure 14. The port pins can be configured as analog inputs (RA3 can also be the voltage reference) or as digital I/O. 4 Least Significant bits for ADCON1 were set to 0011, so that the AN0 pin and AN1 pin are configured as analog input but AN3 is set for Vref as shown in Figure 12.

The ADRESH:ADRESL registers contain the 10-bit result of the A/D conversion. When the A/D conversion is complete, the result is loaded into this A/D Result register pair, the GO/DONE bit (ADCON0<2>) is cleared and the A/D interrupt flag bit ADIF is set. After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

bit 7-6 **ADCS1:ADCS0:** A/D Conversion Clock Select bits (ADCON0 bits in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-3 **CHS2:CHS0:** Analog Channel Select bits
000 = Channel 0 (AN0)
001 = Channel 1 (AN1)
010 = Channel 2 (AN2)
011 = Channel 3 (AN3)
100 = Channel 4 (AN4)
101 = Channel 5 (AN5)
110 = Channel 6 (AN6)
111 = Channel 7 (AN7)

Note: The PIC16F873A/876A devices only implement A/D channels 0 through 4; the unimplemented selections are reserved. Do not select any unimplemented channels with these devices.

bit 2 **GO/DONE:** A/D Conversion Status bit
When ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion which is automatically cleared by hardware when the A/D conversion is complete)
0 = A/D conversion not in progress

bit 1 **Unimplemented:** Read as '0'

bit 0 **ADON:** A/D On bit
1 = A/D converter module is powered up
0 = A/D converter module is shut-off and consumes no operating current

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Figure 11

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 7 **ADFM:** A/D Result Format Select bit
1 = Right justified. Six (6) Most Significant bits of ADRESH are read as '0'.
0 = Left justified. Six (6) Least Significant bits of ADRESL are read as '0'.

bit 6 **ADCS2:** A/D Conversion Clock Select bit (ADCON1 bits in shaded area and in **bold**)

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O
C/R = # of analog input channels/# of A/D voltage references

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Figure 12

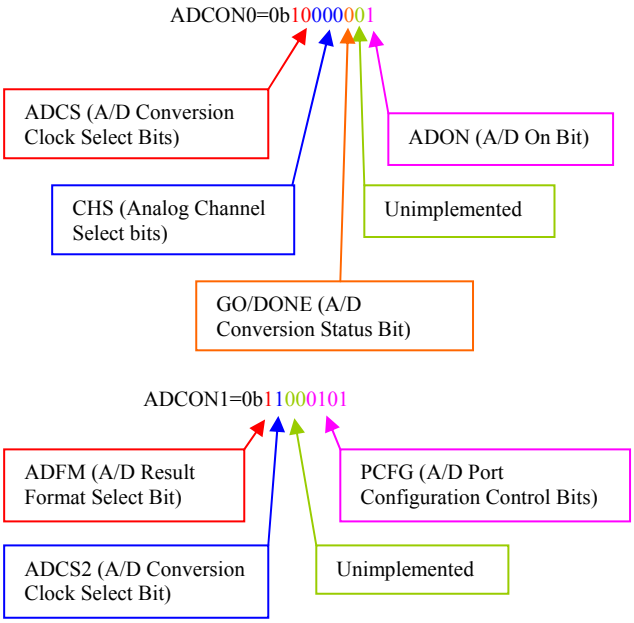


Figure 13

The ADRESH:ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16 bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification.

Figure 14 below shows a subroutine how to get the value from ADC.

```
//====================subroutine ADC==========================

void read_adc(void)
{
    unsigned short i;
    unsigned long result_temp=0;
    for(i=2000;i>0;i-=1)                //looping 2000 times for getting average value
    {
        ADGO = 1;                       //ADGO is the bit 2 of the ADCON0 register
        while(ADGO==1);                 //start ADC, ADGO=0 after finish ADC progress
        result=ADRESH;
        result=result<<8;               //shift to left for 8 bit
        result=result|ADRESL;           //10 bit result from ADC

        result_temp+=result;
    }
    result = result_temp/2000;          //get the average value
}

unsigned short read_temp(void)
{
    unsigned short temp;
    temp=result;
    return temp;
}
```

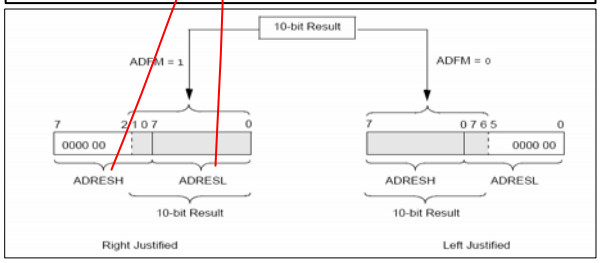


Figure 14

Program Modification

The program can be modified for the desired temperatures to activate the LEDs, fans and buzzer. The sample program is written as shown below. Just change temperature value inside the red circle in Figure 15 below:

```
if((tempA<400)&&(tempB<350))          //  ****************************************
    {                                 //  *   LED A and Fan A activated only for  *
        ledA=1;                       //  *   temperature A greater than 40'C     *
        ledB=0;                       //  *   and temperature B less than 35'C    *
        fanA=1;                       //  ****************************************
        fanB=0;
        buzzer=0;
    }
else if((tempB>350)&&(tempA<400))     //  ****************************************
    {                                 //  *   LED B and Fan B activated only for  *
        ledA=0;                       //  *   temperature A less than 40'C and    *
        ledB=1;                       //  *   temperature B greater than 35'C     *
        fanA=0;                       //  ****************************************
        fanB=1;
        buzzer=0;
    }

else if((tempB>350)&&(tempA>400))     //  ******************************************************
    {                                 //  *   All LED A & LED B, Fan A & Fan B and Buzzer     *
        ledB=1;                       //  *   activated for temperature A greater than 40'C   *
        ledA=1;                       //  *   and temperature B greater than 35'C             *
        fanA=1;                       //  ******************************************************
        fanB=1;
        buzzer=1;
    }

else if((tempB<350)&&(tempA<400))     //  ******************************************************
    {                                 //  *   All LED A & LED B, Fan A & Fan B and Buzzer     *
        ledB=0;                       //  *   disactivated for temperature A less than 40'C   *
        ledA=0;                       //  *   and temperature B less than 35'C                *
        fanA=0;                       //  ******************************************************
        fanB=0;
        buzzer=0;
    }
```

Figure 15

For example, if the desired temperature A is 48°C and temperature B is 55°C, so the value of tempA and tempB should be 480 and 550 respectively.

GETTING START

User can obtain the hardware set for this project (PR11) either by online purchasing (www.cytron.com.my) or purchase it in Cytron Technologies Shop.

1. Once user has the hardware set, soldering process can be started now. Please solder the electronic components one by one according the symbols or overlays on the Printed Circuit Board (PCB). Make sure the component value and polarity is correctly soldered. Please refer to PCB Layout in Appendix A.

Caution: Make sure all the connectors (2510) are soldered in proper side. Those electronic components have polarity such as capacitor, diode, PIC, LM7805, LM35, BD135 and LED should be soldered in right polarity or it may cause the circuit board fail to work.

Warning: Before the battery (Power) is plugged in, make sure the polarity is correct to prevent the explosion. Wrong polarity of capacitor also may cause explosion.

Step for soldering 2510 connector:

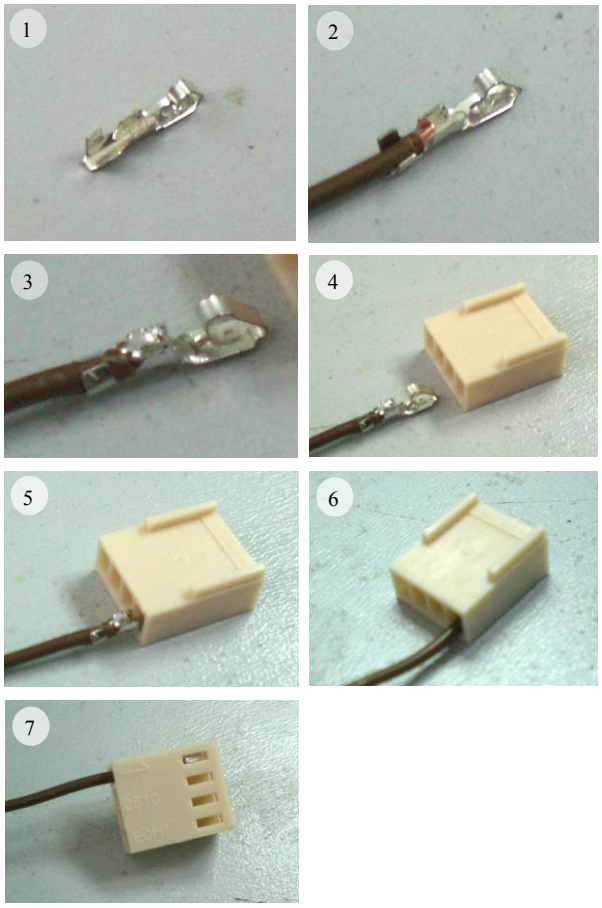


Figure 16

For this project, temperature sensors (LM35) and cooling fans are connected to the circuit board through some wires and connectors. The length of the wires is various depend on the distance of a certain area where would be measured. Each polarity should be correctly connected! Differentiate the types of the connector and please use 2510 connector for LM35 and 2021 connector for cooling fan.

2. After soldering process is finished, please plug in the PIC16F876A to the 28 pins IC socket in proper side. The MPLAB IDE and PICC Lite can be downloading from www.cytron.com.my.
3. After the installation complete, open the project file provided using MPLAB IDE. Please refer to PR1 and PR5 for the method to use MPLAB and PICC Lite. Please plug in the power supply and connect the programmer connector to the circuit board to reprogram the PIC. Do not forget to ON the slide switch! User can get the sample program for this project from Cytron website (same directory as this DIY project) and this program can be modified.
4. After modification, build the project and load the hex file into the PIC microcontroller using Cytron USB In Circuit Programmer (UIC00A). Cytron Technologies do offer USB In Circuit Programmer (UIC00A) as an option in the

hardware list. Please tick the option if the programmer is necessary.

5. User can choose either adaptor or battery to provide the power for the circuit board but make sure the given voltage is between 7V and 15V.

AC to DC adaptor:

Figure 17 (not included in DIY project set)

9V battery connector:

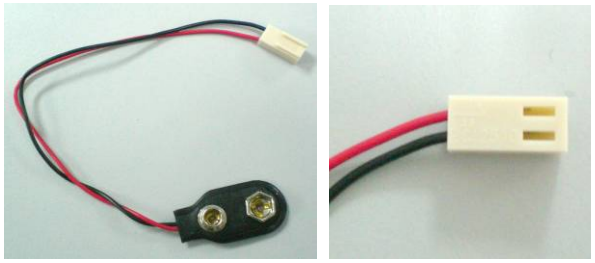


Figure 18 (not included in DIY project set)

Connection to the PCB board:

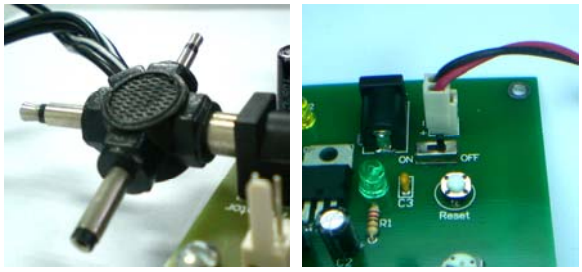


Figure 19

6. When the power is provided, the green LED (small) will turn ON. Let say the sample program (without modification) is being used.
7. First, adjust the 'Contrast' VR2 for desired brightness of LCD Display. Voltage of pin 5 (Vref+) from PIC should adjusted to 1V by rotating the 'Offset' VR1 and using a Multi-meters.
8. If the displayed temperature is different from real temperature, adjust VR1 again to reduce the offset. LED A would light if temperature A reach to 40.0°C.
9. LED B would light to indicate that temperature B is reach to 35.0°C. Both LEDs and buzzer would be activated if temperature A and temperature B is over 40.0°C and 35.0°C respectively. The buzzer will turn ON.

TEST METHOD

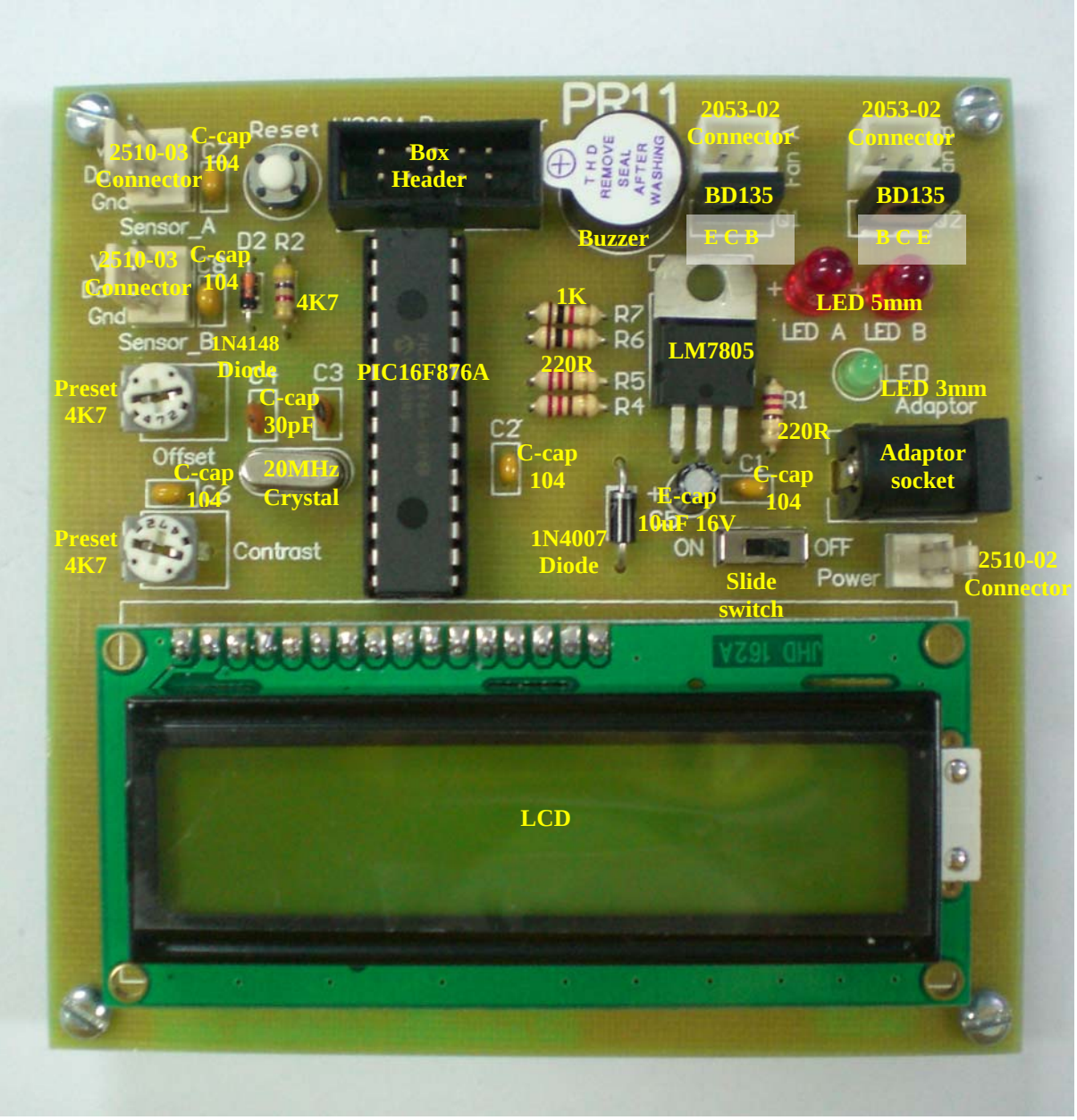
1. Switch ON the power
   - Power Led (green) will turn ON.
   - LCD will display "Temp. A = _ _._ °C Temp. B = _ _._ °C"
2. Adjust preset to get the ambient temperature
   - The temperature reading in LCD will increase to the ambient temperature.
3. Put Sensor B near soldering gun
   - Temp B will increase
   - If Temp B ≥ 35.0°C, LED B (red) will switch on, fan also will turn ON.
4. Then put Sensor A near soldering gun too
   - Temp A will increase
   - If Temp A ≥ 35.0°C, LED A (red) will switch on, fan also will turn ON.
   - If both LED and fan is turn ON, buzzer will activate.
5. If all steps mention above can be executed, your project is done successfully. Congratulations!!

WARRANTY

No warranty will be provided as this is DIY project. Please check the polarity of each electronic component before soldering it to board.

Appendix A

PCB Layout:



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