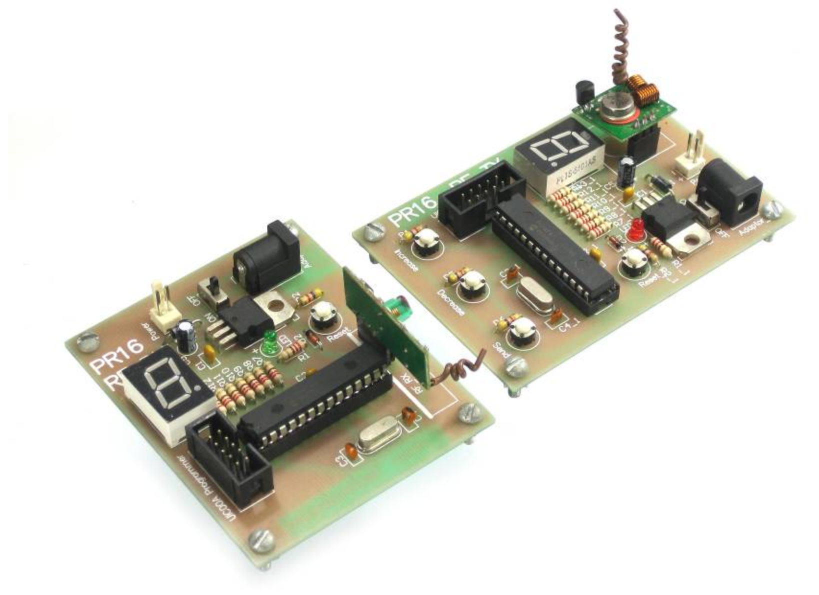
PR16

Sending Data using RF Module



Version 1.2

Aug 2008

Cytron Technologies Sdn. Bhd.

OVERVIEW

This document describes the development of Cytron Technologies DIY (Do It Yourself) Project No.16 (PR16). This project will use two PIC16F876A to control RF Module (Transmitter and Receiver). The transmitter will send 0→F (data in Hexadecimal) and the receiver will receive the data and display the data on 7-segment display. Circuit schematic and PIC source code will be provided.

FEATURES

PIC16F876A

- 8K x 14words of program memory
- On board programming
- Universal Asynchronous Receiver Transmitter (UART)

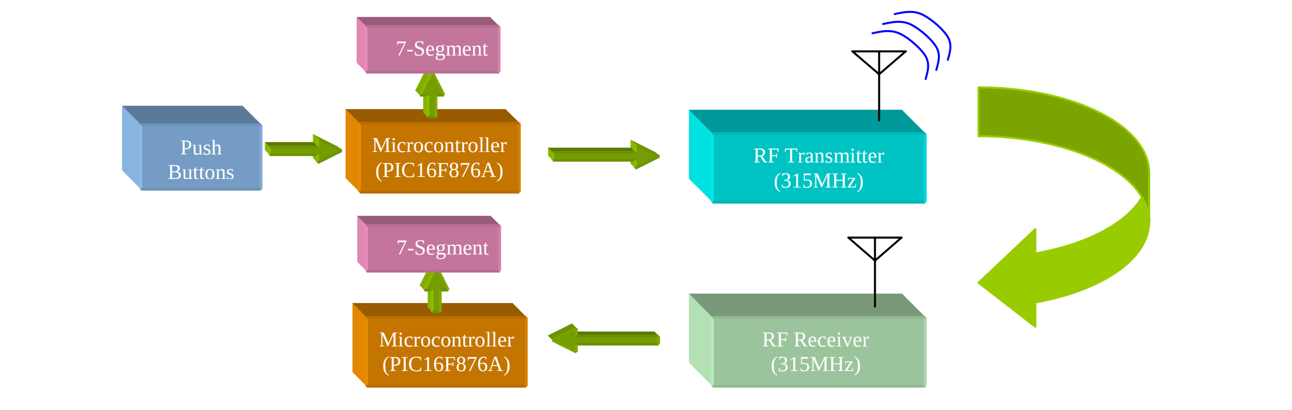
RF Transmitter (315MHz)

- Capable transmit signal up to 100 meters range around the open area
- Wide input supply (2.5-12V)
- Small dimension
- ASK Modulation

RF Receiver (315MHz)

- Super regeneration design ensures sensitive to weak signal
- Low power consumption (4mA)
- Small dimension

SYSTEM OVERVIEW



GENERAL DESCRIPTION

Cytron Technologies offers several wireless modules such as RF Module, Bluetooth Module and ZigBee Module. This project will discuss about how RF module function with a microcontroller. For this project, there are 2 separate circuit boards with a microcontroller (PIC16F876A) each. One board is for transmitter while another for receiver. Hexadecimal number from 0 to F (which display on 7-segment) can be chosen for transmitting. The receiver will receive the sent data and display the number.

RF Module

RF Transmitter (315MHz):

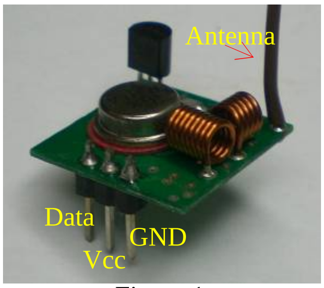


Figure 1

This low cost RF transmitter can be used to transmit signal up to 100 meters (the antenna design, working environment and supply voltage will seriously impact the effective distance). It is good for short distance, battery power device development.

It has benefits of:

- Wide input supply (2.5V-12V)
- Easy to integrate (Data, Vcc and GND)
- Device in deep sleep mode when Data pin is grounded
- Very small dimension

Specification:

Operating Voltage	2.5 V to 12 V
Operating Current	4mA @ 5V, 15mA @ 9V
Quiescent Current	10uA
Operating Temperature	-10°C - 60°C
Modulation	ASK
Max. Data Rate	9.6K
Data Input	TTL
RF Power	20 mW@5V

RF Receiver (315MHz):

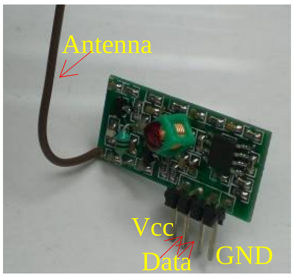


Figure 2

This low cost RF Receiver can be used to receive RF signal from any 315MHz transmitter. Super regeneration design ensure sensitive to weak signal.

It has benefits of:

- Low power consumption (4mA)
- Easy to integrate (Vcc, GND and Data)
- Super regeneration design
- Very small dimension

Specification:

Operating Voltage	4.5V to 5.5V
Operating Current	4mA @ 5V
Operating Temperature	-10°C - 60°C
Sensitivity	-105dBm
Max. Data Rate	4.8K
Data Output	TTL

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)

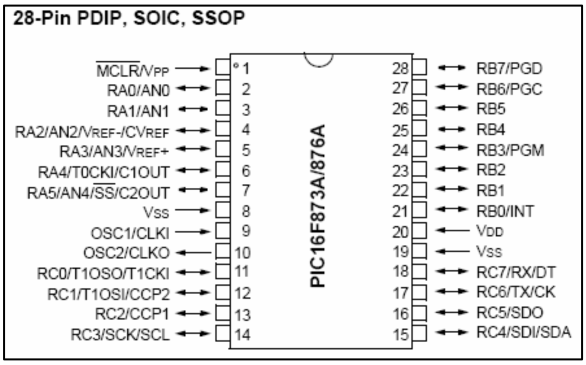


Figure 3

Figure 3 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 RF Module Set (Transmitter and Receiver)
b. 2 x PR16 Printed Circuit Board (RF_RX & RF_TX)
c. 2 x PIC16F876A
d. Other related electronic components

Please refer to the schematic diagram of PR16. The schematic is provided free therefore Cytron Technologies will not be responsible for any further modification or improvement.

Interface PIC16F876A with RF Transmitter

As mention early, this RF Transmitter has 3 pins (Vcc, GND and Data). In this project, Vcc pin is connected to 5V and GND pin is connected to GND of circuit board. The data pin should be connected to pin 17 (RC6/TX/CK) of PIC16F876A.

Interface PIC16F876A with RF Receiver

There are 4 pins for the receiver (Vcc, GND and 2 Data pins). The 2 data pins are internally connected each other, thus connecting either one to PIC is sufficient. Same as transmitter, 5V is given to Vcc pin and GND pin is connected to GND of circuit board. The data pin should be connected to pin 18 (RC7/RX/DT) of PIC16F876A.

Power Supply for Circuit

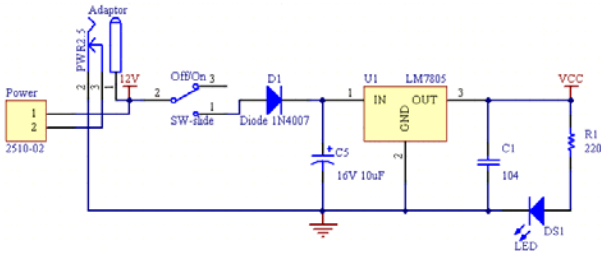


Figure 4

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 power 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

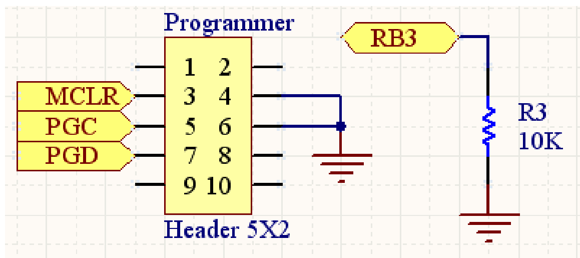


Figure 5

In Circuit Serial Programming (ICSP) is used for loading program in this project. ICSP offers a convenience way to load program into PIC microcontroller without removing the PIC from the circuit board. Thus 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 5. 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) should not be used as input, anyway it still can be used for output.

Push Button as Input for PIC microcontroller

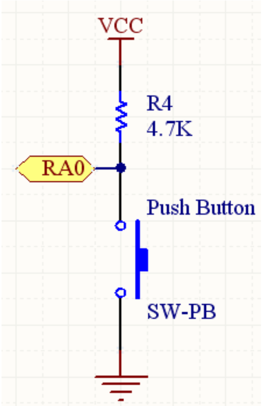


Figure 6

One I/O pin is needed for one push button as input of PIC microcontroller. The connection of the push button to the I/O pin is shown in Figure 6. 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 (inside 7-segment) as Output for PIC microcontroller

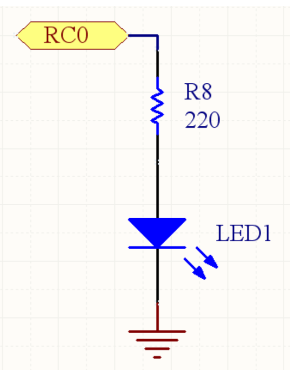


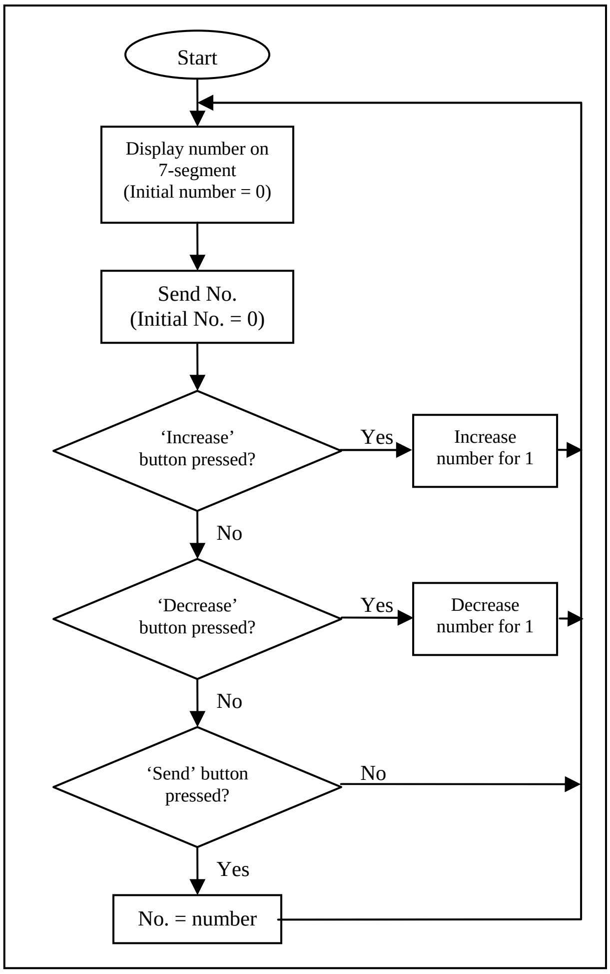
Figure 7

One I/O pin is needed for one LED as output of PIC microcontroller. The connection for a LED to I/O pin is shown in Figure 7. 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.

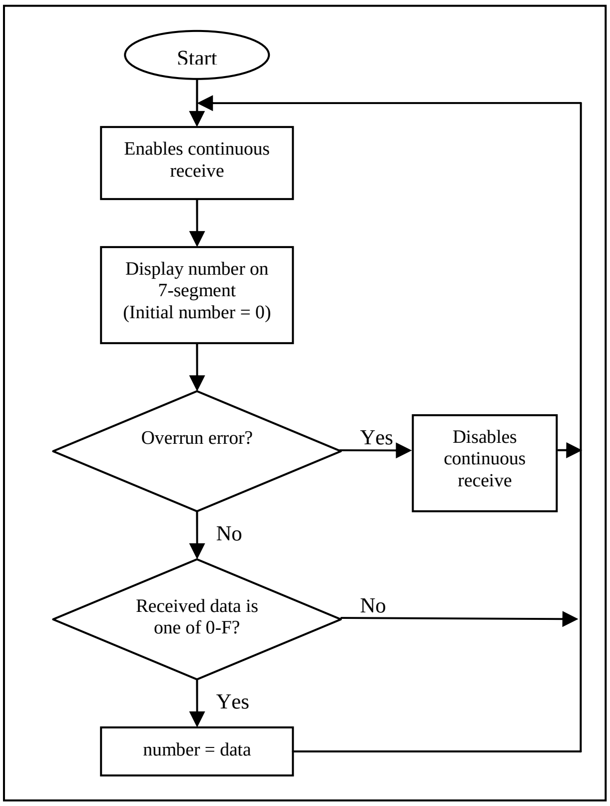
SOFTWARE

Flow Chart:

RF Transmitter:



RF Receiver:



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.

USART Configuration

The Universal Synchronous Asynchronous Receiver Transmitter (USART) Module for PIC16F876A can be configured in the following mode:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Asynchronous receiver/transmitter is used to translate data between parallel and serial interface in this project. When doing a data communications, the condition of '0' and '1' from the side of the sending must be able to be recognized at the receiving side. In the asynchronous communication, it puts a start bit to the head of the transferred data (8 bits or 9 bits) and it puts a stop bit at the end of the data. Recognition in the data block is done by it. The start bit is a Low level and the stop bit is the signal of the High level.

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

bit 7 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

bit 6 **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN**: Single Receive Enable bit
Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.

bit 4 **CREN**: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

bit 2 **FERR**: Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR**: Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data (can be parity bit but must be calculated by user firmware)

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 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
Don't care.
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode.

bit 1 **TRMT**: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

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 9

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64 (X + 1))	Baud Rate = Fosc/(16 (X + 1))
1	(Synchronous) Baud Rate = Fosc/(4 (X + 1))	N/A

Legend: X = value in SPBRG (0 to 255)

Figure 10

BAUD RATE (K)	Fosc = 20 MHz KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-
1.2	1.221	1.75	255
2.4	2.404	0.17	129
9.6	9.766	1.73	31
19.2	19.531	1.72	15
28.8	31.250	8.51	9
33.6	34.722	3.34	8
57.6	62.500	8.51	4
HIGH	1.221	-	255
LOW	312.500	-	0

SPBRG

Figure 11

Figure 8 and Figure 9 shows the TXSTA Register and RCSTA Register respectively. First, all relevant bits from TXSTA Register and RCSTA Register should be configured as shown in Figure 12 below. The given baud rate from PIC microcontroller for this RF module is up to 3 Kbps. However, the baud rate was set to 1.2 Kbps for this project to reduce the error during sending data. By referring to Figure 10 and Figure 11, SPBRG Register was set to 255 due to 1.2 Kbps baud rate.

```
//setup USART
BRGH = 0;                  //baud rate low speed option
SPBRG = 255;               //set boud rate to 1200bps for 20Mhz crystal
TX9 = 0;                   //8-bit transmission
TXEN = 1;                  //enable transmission
SYNC = 0;                  //asynchronous

SPEN = 1;                  //enable serial port
```

For Transmitter

```
//setup USART
BRGH = 0;                  //baud rate low speed option
SPBRG = 255;               //set boud rate to 1200bps for 20Mhz crystal
SPEN = 1;                  //enable serial port
RX9 = 0;                   //8-bit reception
CREN = 1;                  //enable reception
```

For Receiver

From TXSTA Register
From RCSTA Register
SPBRG Register

Figure 12

```
#define display      PORTB
```

Define Port B as 'display'

```
                                    //7 segment display
unsigned char _7seg[16]={0b01111111,0b00001101,0b10110111,0b10011111,   //0,1,2,3
                         0b11001101,0b11011011,0b11111011,0b00001111,   //4,5,6,7
                         0b11111111,0b11011111,0b11101111,0b11111001,   //8,9,A,B
                         0b01110011,0b10111101,0b11110011,0b11100011};  //C,D,E,F
```

```
num=0;
no=0;
display=_7seg[num];
```

PORTB=0b01111111
7-segment display '0'

Figure 13

Refer to the schematic diagram; pin RB1 to RB7 from PIC are connected to 7-segment and RB0 is "don't care" bit. Figure 13 shows the program which displays 0 to F depend on value of 'num'. When the value of 'num' changed from 0 to 15, the displayed number will changed from 0 to F respectively. The pattern of displayed numbers and alphabets should be determined first by referring the pin diagram of 7-segment below.

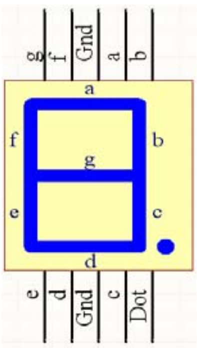


Figure 14

```
//  functions
//==================================================================
void uart_send(unsigned char data)
{
    while(TXIF==0);                  //only send the new data after
    TXREG=data;                      //the previous data finish sent
```

Figure 15

```
//  functions
//==================================================================
unsigned char uart_rec(void)      //receive uart value
{
    unsigned char rec_data;
    while(RCIF==0);               //wait for data
    rec_data = RCREG;
    return rec_data;              //return the received data
```

Figure 16

Figure 15 shows the Function for Data Transmitting while Figure 16 shows the Function for Data Receiving. During the transmitting process, TXIF=0 (cleared) until previous data is totally transmitted. For data receiving, RCIF=1 when reception is complete then RCREG register will store the received data.

GETTING START

User can obtain the hardware set for this project (PR16) 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.

Notice: Do not forget to solder a wire for each transmitter and receiver as an antenna.

Caution: Make sure all the connectors (2510) are soldered in proper side. Those electronic components have polarity such as capacitor, diode, PIC, RF Transmitter, RF Receiver 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:

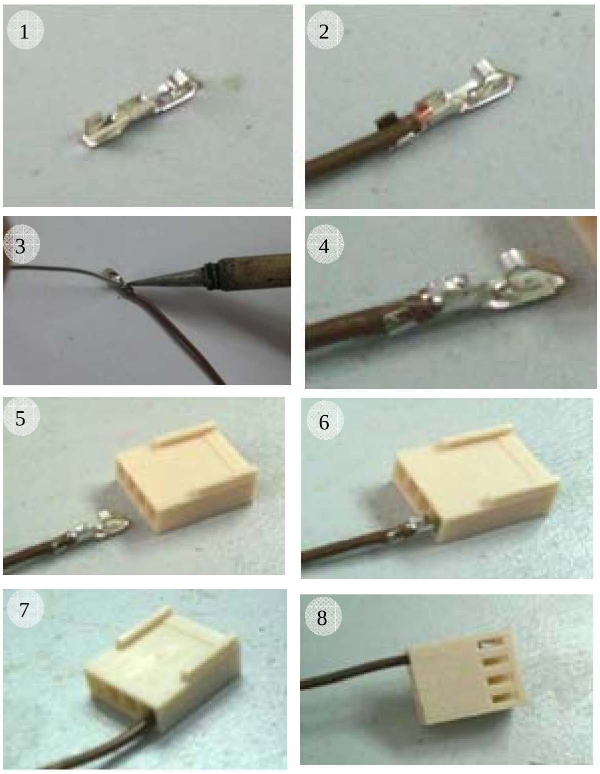


Figure 17

2. After soldering process is finished, please plug in the PIC16F876A to the 28 pins IC socket in proper side.

3. Please download the necessary files and document from Cytron Technologies website, www.cytron.com.my. These included documentation, sample source code, schematic, component list and software.

4. The next step is to install MPLAB IDE and HI-TECC C PRO into a computer. The MPLAB IDE and HI-TECC C PRO can be downloaded from www.cytron.com.my . Please refer MPLAB IDE installation step document to install the software. The documents can be used to any version of MPLAB IDE software.

5. After the installation complete, open the project file provided using MPLAB IDE. Please refer MPLAB Open Project document to open the sample program.

6. Plug in power supply for the circuit. User can choose to use battery or AD to DC adaptor.

AC to DC adaptor:

Figure 18 (not included in DIY project set)

9V battery connector:

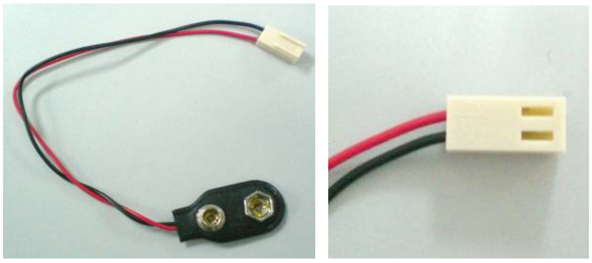


Figure 19 (not included in DIY project set)

Connection to the PCB board:

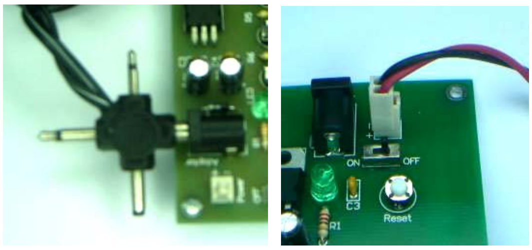


Figure 20

7. Build the project and load the hex file into the PIC microcontroller using the USB In Circuit Programmer (UIC00A). When user build the project, MPLAB IDE will generate hex file. The hex file generated from MPLAB IDE will be named according to project name, not C file name. Cytron Technologies also provide hex file for user. Do not forget to switch ON the power. The programmer is not included in the hardware set but it can be found at Cytron website. (User manual is provided at website).

8. User can modify this program. After modification, build the project once again and load the hex file into the PIC microcontroller using Cytron USB In Circuit Programmer (UIC00A).

7. User can select the desired number by the 'Increase' button and 'Decrease' button. For sending the data, press 'Send' button then the

receiver circuit board will display the sent number.

TEST METHOD

1. Switch ON the power for both RF_TX and RF_RX
    - Power Led of RF_TX (red) will turn ON
    - Power Led of RF_RX (green) will turn ON
    - 7-segment for both RF_TX and RF_RX will display "0"
2. Press Increase button in RF_TX
    - Value in 7-segment will change, it will display the increase number
3. Press Decrease button in RF_TX
    - Value in 7-segment will change, it will display the decrease number
4. Press Send button in RF_TX
    - RF_TX will send the display number to RF_RX
    - 7-segment display of RF_RX will display the same number in 7-segment display of RF_TX
5. Press Reset Button
    - 7-segment display for both RF_TX and RF_RX will display "0"
6. If all steps mention above can be executed, your project is done successfully. Congratulations!!

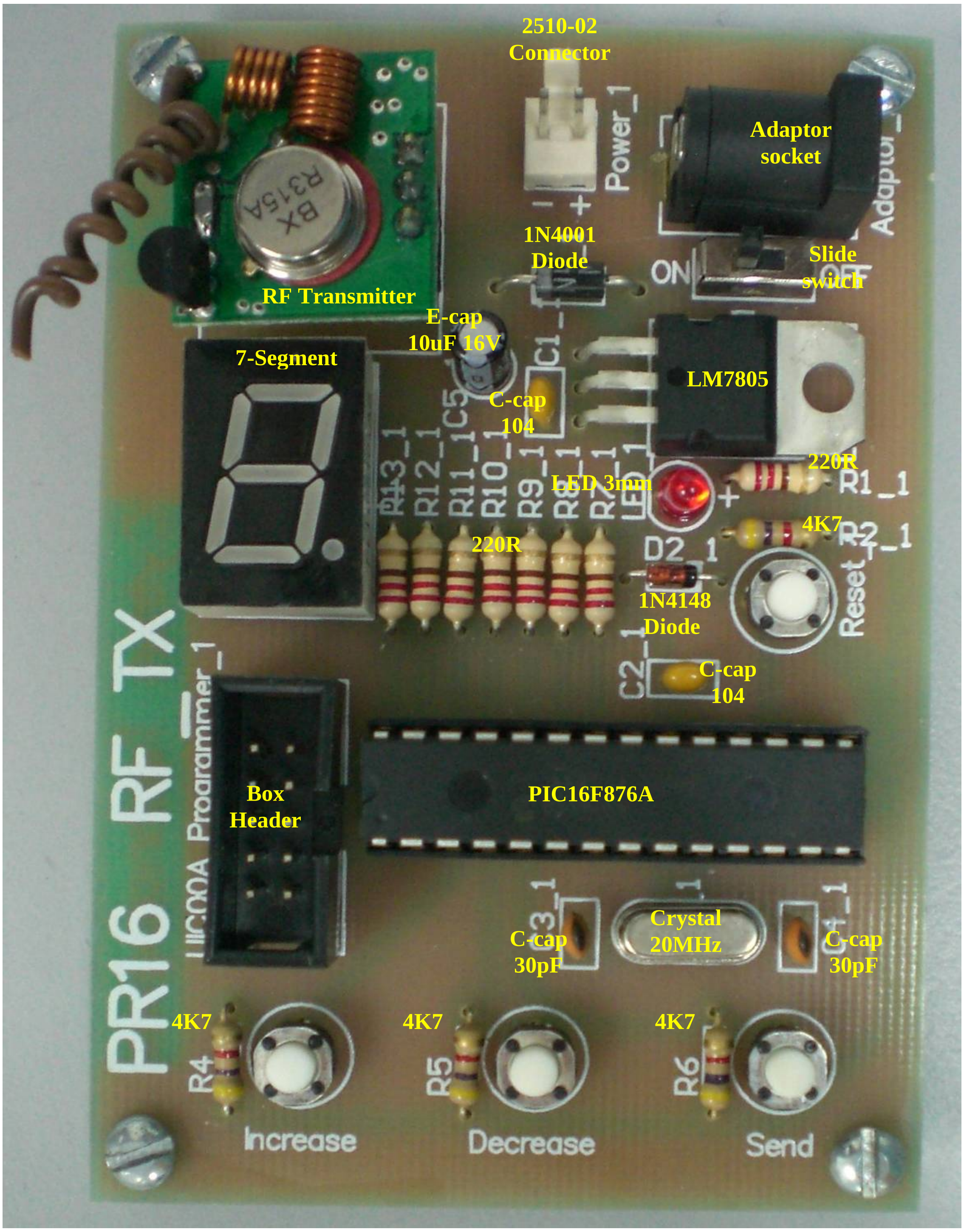
WARRANTY

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

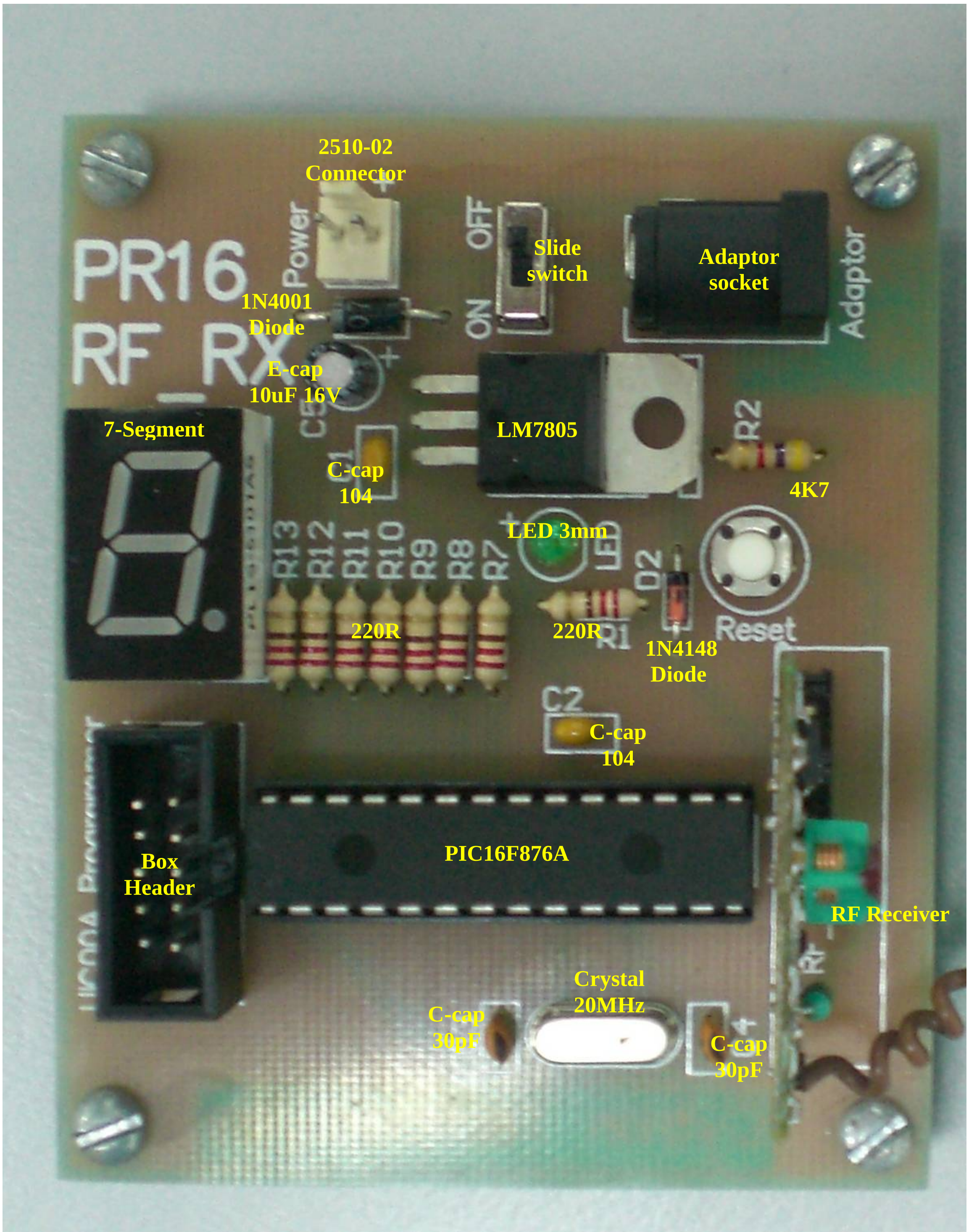
Appendix A

PCB Layout:

TRANSMITTER (TX)



RECEIVER (RX)


2510-02
Connector
Slide
switch
Adaptor
socket
Adaptor
PR16
RF RX
Power
OFF
ON
1N4001
Diode
E-cap
10uF 16V
7-Segment
LM7805
R2
C-cap
104
4K7
LED 3mm
R13 R12 R11 R10 R9 R8 R7
D2
Reset
220R
220R
R1
1N4148
Diode
C2
C-cap
104
Box
Header
PIC16F876A
RF Receiver
Crystal
20MHz
C-cap
30pF
C-cap
30pF

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