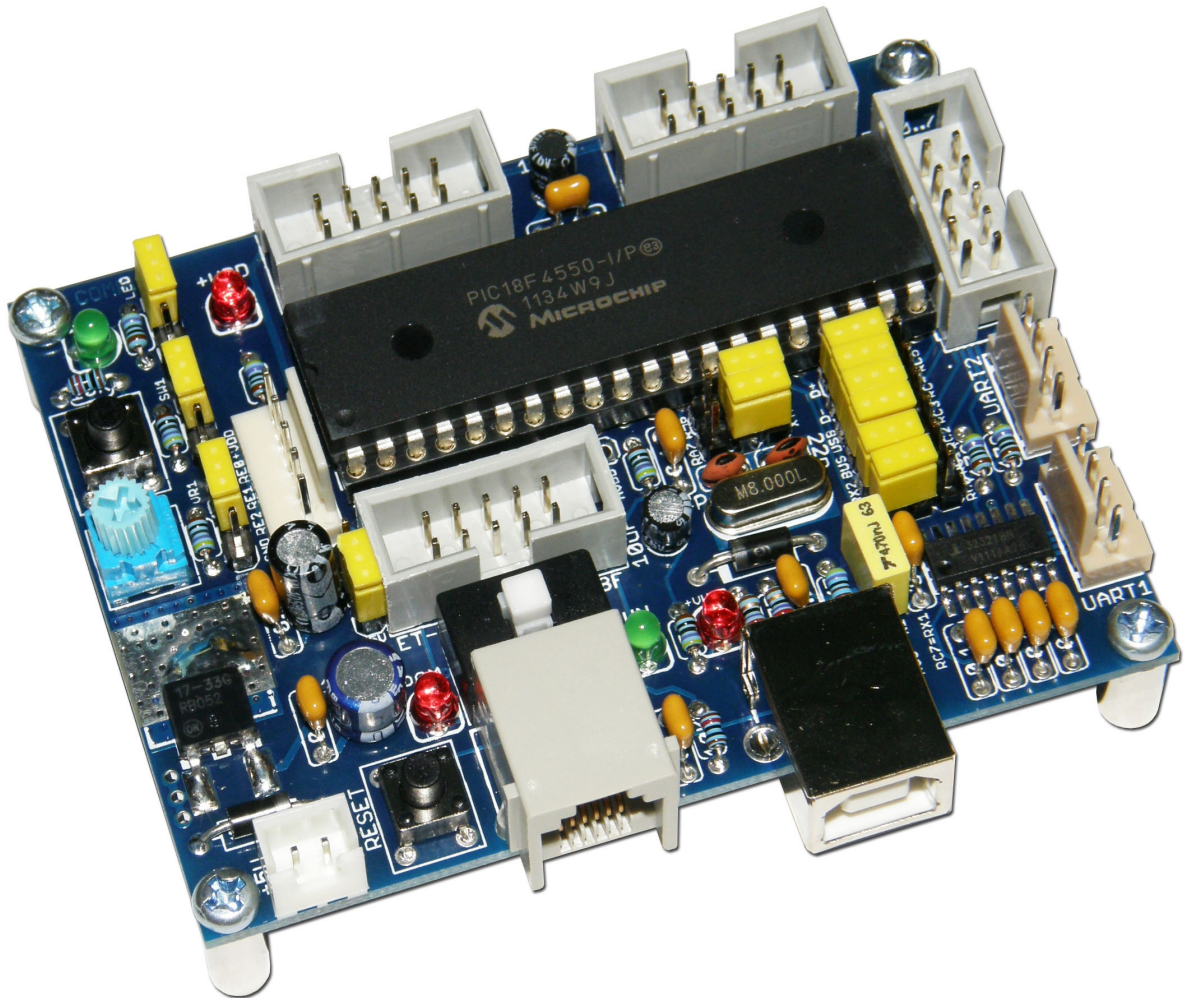


ET-BASE PIC40/4550 (ICSP)

ET-BASE PIC40/4550 (ICSP) is Board Microcontroller in the series of PIC that is designed to use and install Chip Microcontroller 40PIN No. PIC18F4550; it can support program development of board to be USB Devices easily.

Structure of Board **ET-BASE PIC40-4550 (ICSP)** is designed to be mini board that is suitable for various applications and basic Training Kit. Internal board provides circuits that are necessary and convenient to use and develop program. It is flexible because it can adjust Signal I/O for various applications according to preferable objectives suitably.

Moreover, it adds more devices for initially testing Input/Output internal board, so user can use these devices to be tools for testing operation while developing program. For example, it uses Adjustable VR to test the operation of ADC; or it uses Push-Button Switch to test the operation of Input Logic; or it uses LED to test the operation of Output Logic.

Specifications of Board

- Use MCU PIC 40PIN (40PDIP) No.PIC18F4550 on board
 - 32KByte Flash/ 2048 Byte SRAM/ 256 Byte EEPROM
 - 35 GPIO
 - 13 Channel 10Bit ADC
 - 1 Channel CCP/ 1 Channel ECCP
 - 1 Channel I2C/ 1 Channel SPI
 - 1 Channel EUART
 - 2 Channel Comparator
 - 1 Channel 8 Bit Timer/ 3 Channel 16 Bit Timer
- Use Crystal 8.00MHz with Jumper to connect/disconnect signal
- Has 2-Channel Circuit Line Driver for UART RS232 Serial Port Communication; it uses Connector UART CPA-4Pin according to ETT Standard
 - 1-Channel for Hardware UART1; it uses Pin RC6(TX1) and RC7(RX1) according to PIC Standard.
 - 1-Channel for Software UART; it uses Pin RC0(TX2) and RC1(RX2); moreover, there is Jumper to choose the operation modes either UART2(Software UART) or GPIO as required.
- Has Connector ICSP RJ11 according to ICD2 Standard; it is used with Programmer and Debugger that supports the operation according to the ICD2 Standard of MICROCHIPS such as ICD2/ICD3 or Pickit2/Pickit3.
- Has Switch to alternate the operation modes between Programmer/Debugger(PGM) and Normal Run (RUN); moreover, there is LED to display the operation mode of board.
- Has 4 of Header I/O 2x5 and 1 of Header CPA-5Pin
- Has Switch RESET to reset the operation of MCU internal board
- Has Adjustable VR to test the operation of ADC Input; it uses RA0 with Jumper to connect/disconnect signal.
- Has Switch to test the operation of Digital Input; it uses RA1 with Jumper to connect/disconnect signal.
- Has LED to test the operation of Digital Output; it uses RA2 with Jumper to connect/disconnect signal.
- Has Power +5VDC Input with Regulate 3.3V/1A and LED to display operating status of Power Supply; moreover, there is Jumper to choose either +5VDC or 3.3VDC to be source of Power Supply for MCU.
- Has Circuit USB Interface; it uses Connector USB-B with LED to display status and Jumper to connect/disconnect signal. It supports the operation with MCU number that has the HARDWARE System as USB Device internal MCU.
- Be Mini PCB Size: 8 x 6 cm.

Structure of Board ET-BASE PIC40/4550 (ICSP)

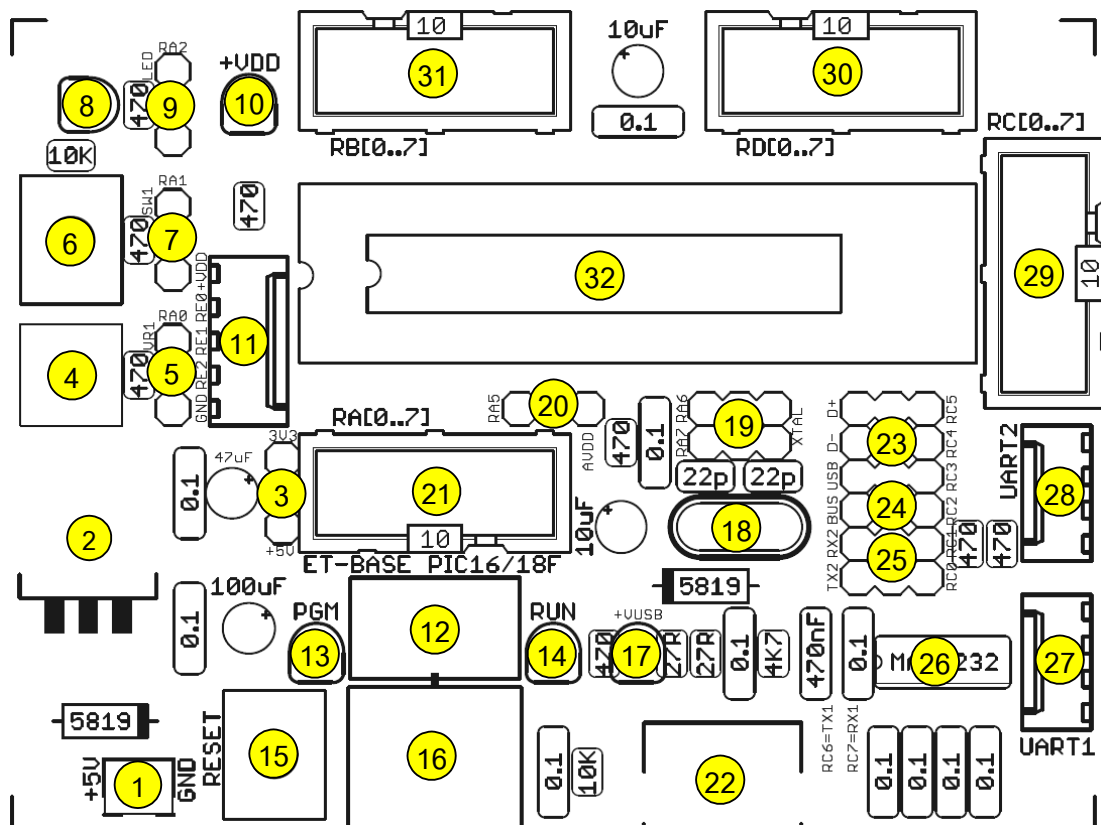
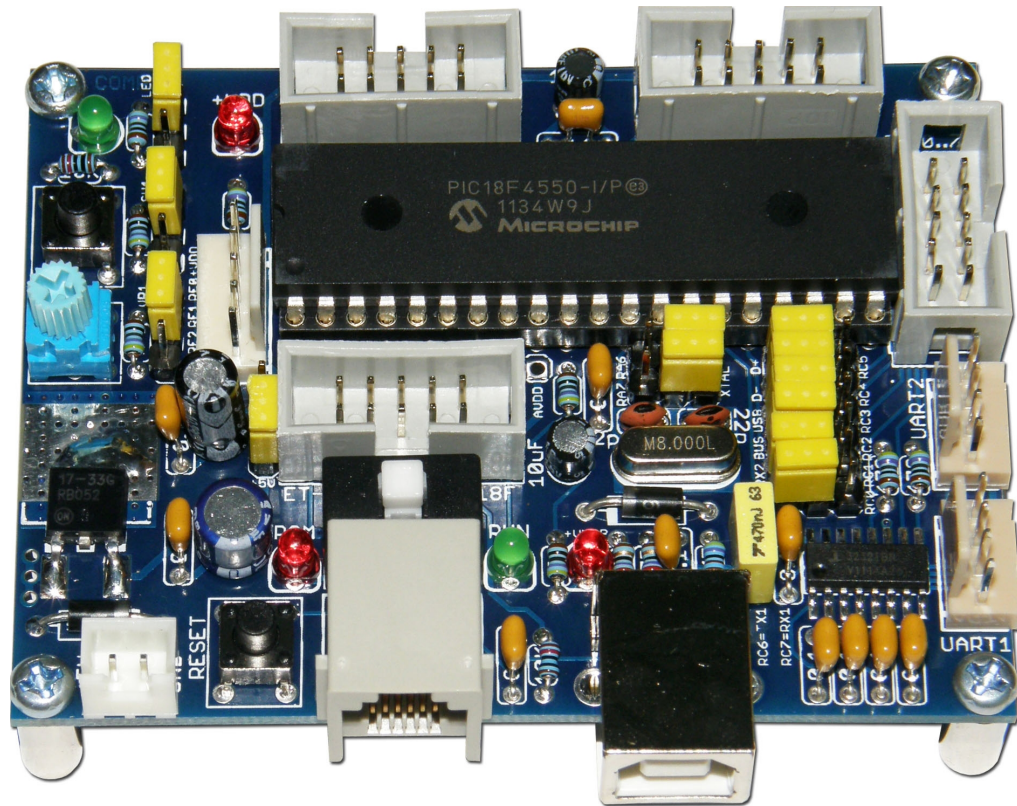


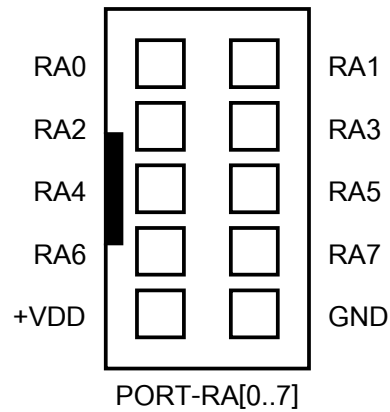
Figure shows the structure of Board ET-BASE PIC40/4550(ICSP) .

- **No.1:** It is Connector +5VDC Power Supply of board.
- **No.2:** It is IC Regulate 3.3VDC/1A.
- **No.3:** It is Jumper to choose either 3.3V or 5V that is the voltage level of Power Supply for MCU(+VDD) .
- **No.4:** It is Adjustable VR(VR1) to test the operation of Input Analog(ADC) .
- **No.5:** It is Jumper to connect/disconnect Signal RA0 and the adjusted voltage from VR1.
- **No.6:** It is Push-Button Switch(SW1) to test the operation of Digital Input.
- **No.7:** It is Jumper to connect/disconnect Signal RA1 and Digital Input from SW1.
- **No.8:** It is LED to test the operation of Digital Output.
- **No.9:** It is Jumper to connect/disconnect Signal RA2 and Digital Output for LED.
- **No.10:** It is LED to display the status of +VDD Power Supply.
- **No.11:** It is Connector RE[0...2] that is used in case of installing MCU 40Pin only.
- **No.12:** It is Switch to choose operation modes either Run(RUN) and Program(PGM) .
- **No.13:** It is red LED to show operating status of PGM when board is running in Program Mode.
- **No.14:** It is green LED to show operating status of RUN when board is running in Run Mode.
- **No.15:** It is Switch RESET to reset the operation of MCU when it is running in RUN Mode.
- **No.16:** It is Connector ICSP to interface with Programmer and Debugger according to the ICD2 Standard.
- **No.17:** It is LED to display status of +VUSB.
- **No.18:** It is Crystal 8.00MHz.
- **No.19:** It is Jumper to choose Pin RA6,RA7 of MCU either to interface to be GPIO at Connector RA[0...7] or Circuit Crystal. In case of Board ET-BASE PIC40/4550(ICSP), it always sets Jumper to the side of Crystal.
- **No.20:** It is Jumper to choose function of Pin7 of MCU 28Pin to be either Signal RA5 of MCU by interfacing to be GPIO at Connector RA[0...7] or to be Pin +AVDD. In case of Board ET-BASE PIC40/4550(ICSP), it always sets Jumper to be RA5.
- **No.21:** It is Connector IDE10Pin of RA[0...7] .
- **No.22:** It is Connector USB.

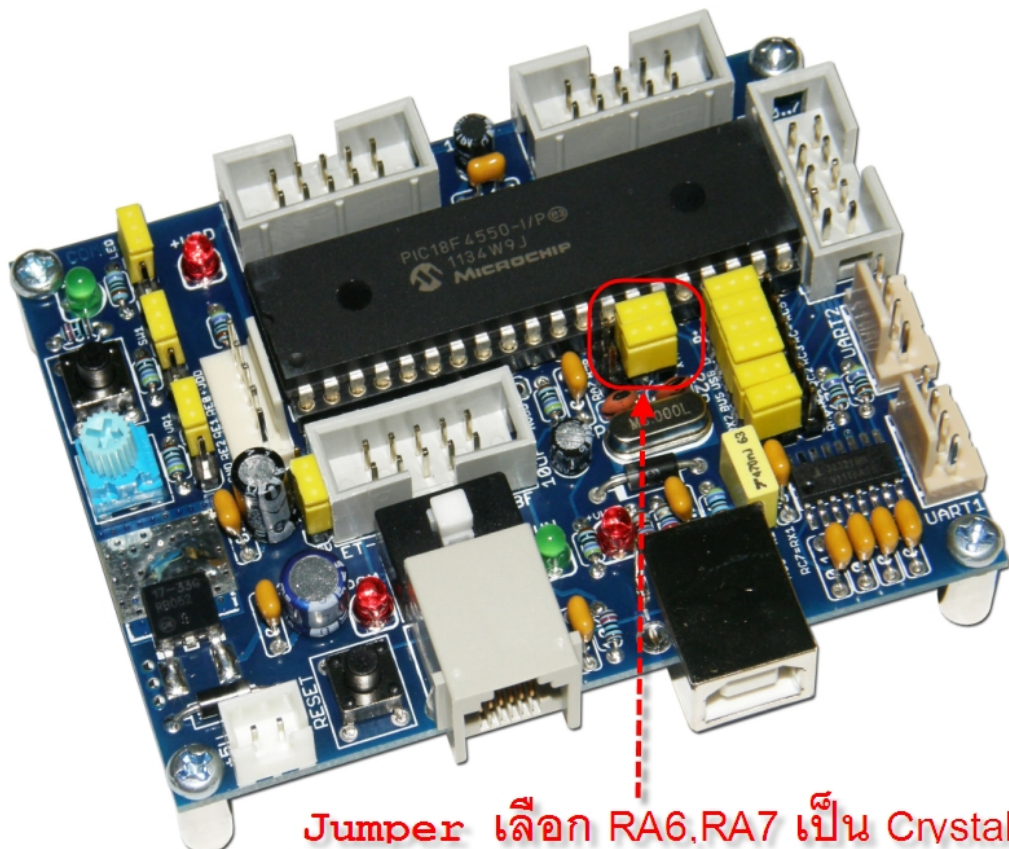
- **No.23:** It is Jumper to choose Pin RC3,RC4,RC5 of MCU either to interface to be GPIO at Connector RC[0...7] or to be Signal to interface with USB Bus(VBUS,D-,D+).
- **No.24:** It is Jumper to choose Pin RC2 of MCU either to interface to be GPIO at Connector RX[0...7] or to be Pin VBUS of USB.
- **No.25:** It is Jumper to choose Pin RC0,RC1 of MCU either to interface to be GPIO at Connector RC[0...7] or TX2,RX2 of UART2.
- **No.26:** It is IC Line Driver of RS232(MAX3232) to convert Signal Level between Signal UART Logic and Standard Signal RS232.
- **No.27:** It is Connector UART1 that is Signal RS232; it supports Hardware UART that uses Pin of RC6(TX1) and RC7(RX1) to be signal for connection.
- **No.28:** It is Connector UART2 that is Signal RS232; it supports Software UART that uses Pin of RC0(TX2),RC1(RX2) to be signal for connection.
- **No.29:** It is Connector IDE10Pin of RC[0...7].
- **No.30:** It is Connector IDE10Pin of RD[0...7].
- **No.31:** It is Connector IDE10Pin of RB[0...7].
- **No.32:** It is MCU No.PIC18F4550 on board.

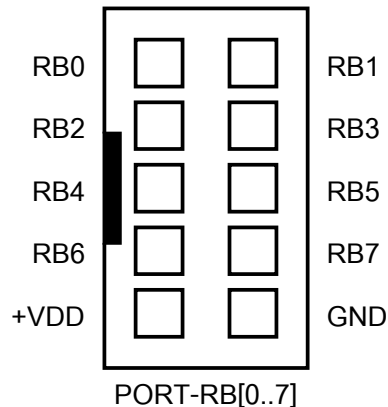
CONNECTORS

PORT RA[0..7]

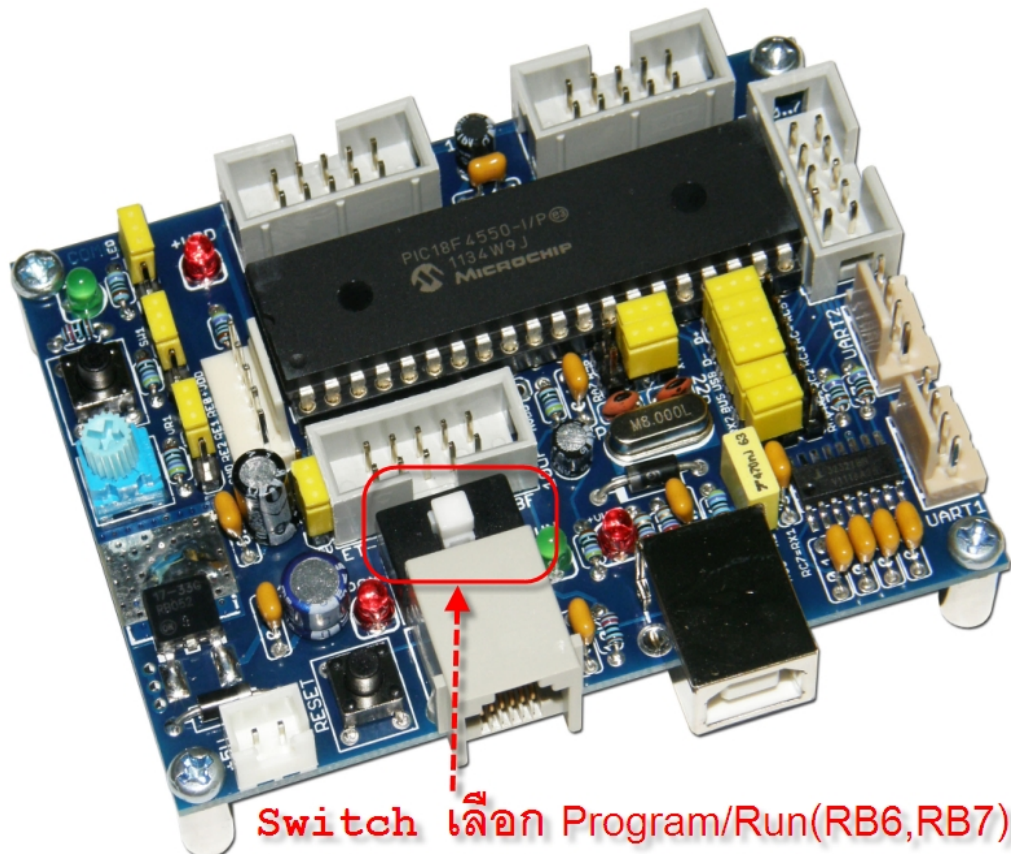


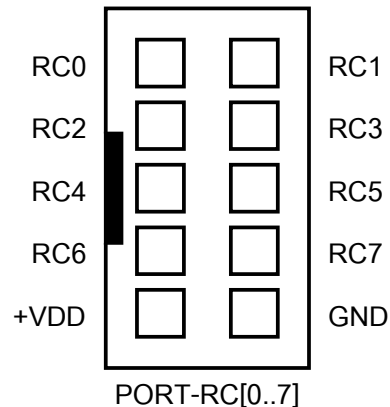
PORT-RA[0..7]: It is Signal from RA0...RA7 of MCU. Signal RA0...RA5 is directly interfaced from Pin of MCU. Signal RA6 and RA7 has Jumper to connect/disconnect signal; in this case, user can choose either to use Signal RA6,RA7 to be GPIO or to connect with Circuit Crystal Generator. In case of using PIC18F4550, Signal RA6,RA7 is always interfaced with Circuit Crystal Generator.



PORT RB[0...7]

PORT-RB[0...7]: It is Signal from RB0...RB7 of MCU. Signal RB0...RB5 is directly interfaced from Pin of MCU. Signal RB6 and RB7 has Switch to connect/disconnect signal; in this case, user can choose either to use Signal RB6,RB7 to be GPIO or to connect with ICSP Programmer through Connector ICSP(RJ11). If it shifts the Switch to position of RUN, it interfaces Pin RB6,RB7 to this Connector; on the other hand, if it shifts the Switch to position of PGM, it interfaces Signal RB6,RB7 of MCU to the Programmer through Connector ICSP(RJ11) instead.

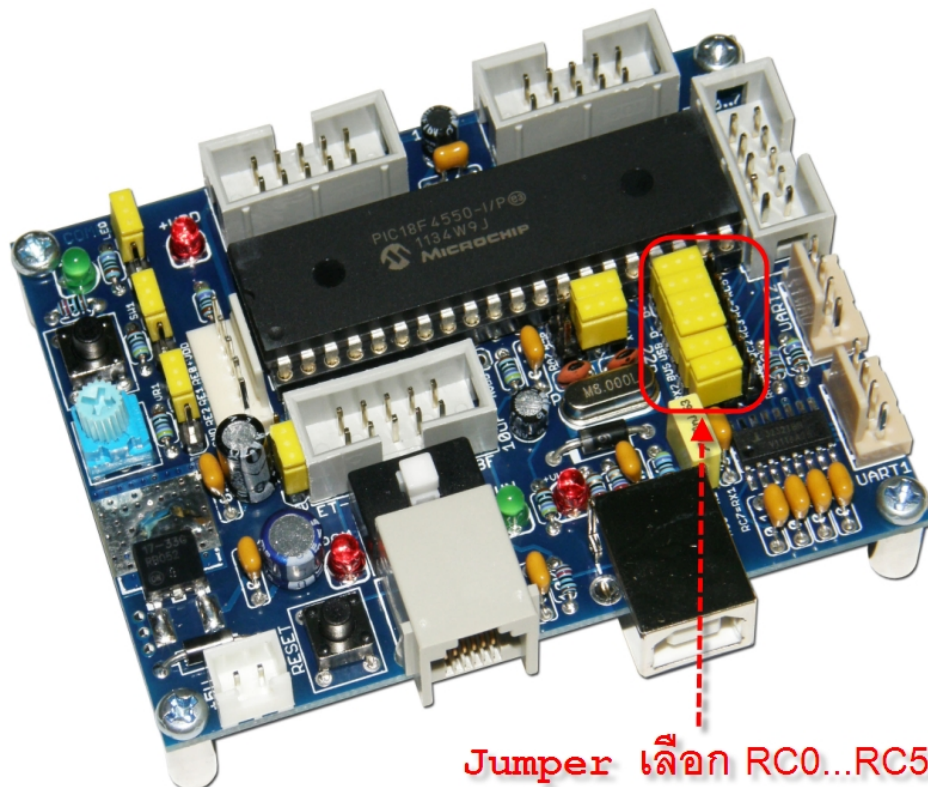


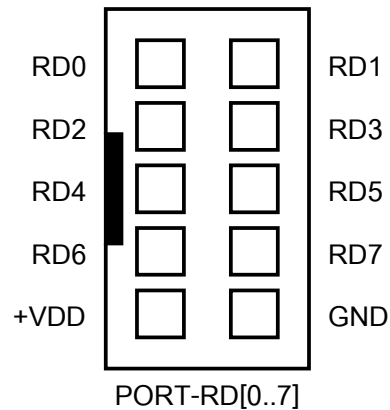
PORT-RC[0...7]

PORT-RC[0..7]: It is Signal from RC0...RC7 of MCU. Signal RC0...RC5 has been set by Jumper. Signal RC6 and RC7 are directly interfaced from Pin of MCU.

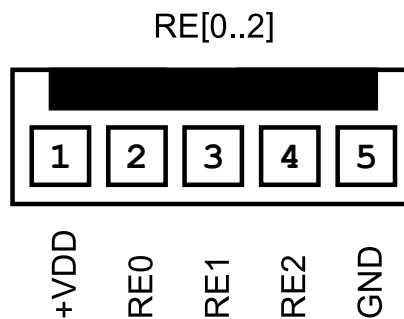
RC0,RC1: There is Jumper to set Pin RC0 and RC1 either to interface to this Connector or to interface RC0,RC1 to Circuit Line Driver of UART2(Software UART).

RC2...RC5: There is Jumper to set Pin RC2...RC5 either to interface to this Connector or to interfaces to Circuit of USB Interface. In case of PIC16F1939, there is no any Circuit of USB Interface internal structure of MCU; so, user has to set Jumper of RC2, RC3, RC4, RC5 to the side of GPIO. In this case, it always interfaces with Pin RC2, RC3, RC4, RC5 of this Connector as shown in picture below;



PORT RD[0...7]

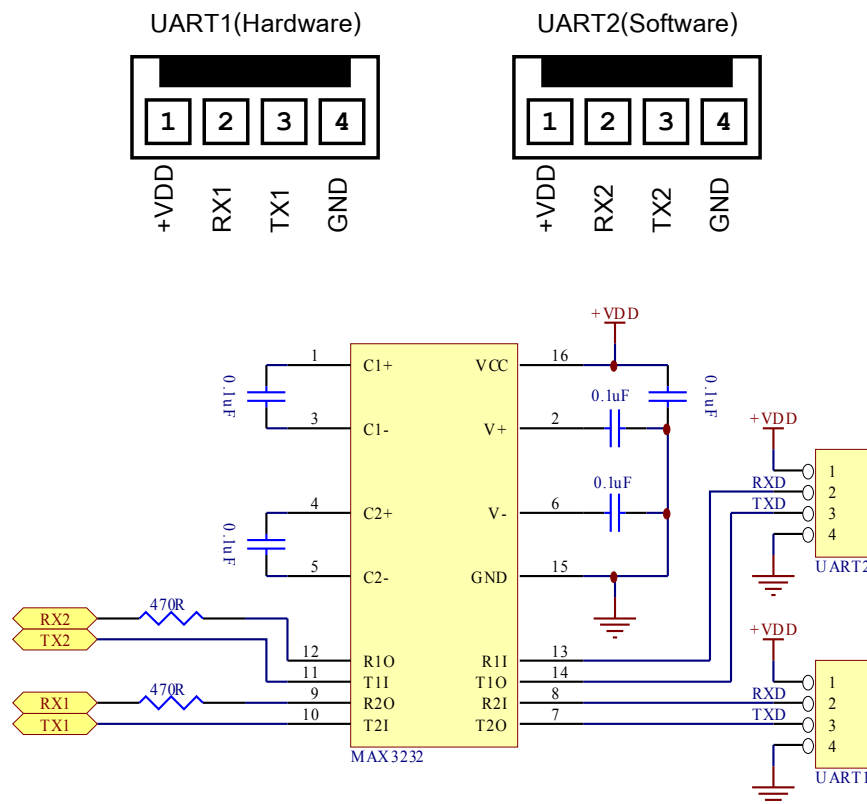
PORT-RD[0...7]: It is Signal from RD0...RD7 of MCU. All 8 Signals are directly interfaced from Pin of MCU.

PORT-RE[0..2]

PORT-RE[0...2]: It is Signal from RE0...RE2 of MCU. All 3 Signals are directly interfaced from Pin of MCU.

PORT RS232

PORT RS232: It is Signal RS232 that has converted Signal Level by MAX3232 completely. In case of PIC18F4550, there is 1-CH Hardware UART; it uses Pin RC6(TX1) and RC7(RX1) for connection. Moreover, it has added UART2(Software UART) for the circuit of board. The second channel uses RC0(TX2) and RC1(RX2) with Jumper to connect/disconnect Signal UART2 as required. Signal RS232 in each channel arranges Connector to be CPA-4PIN(RS232) as show in the picture below;



Cable RS232 is connected between ComPort of computer PC and Connector RS232 of Board ET-BASE PIC40/4550(ICSP) as follows;

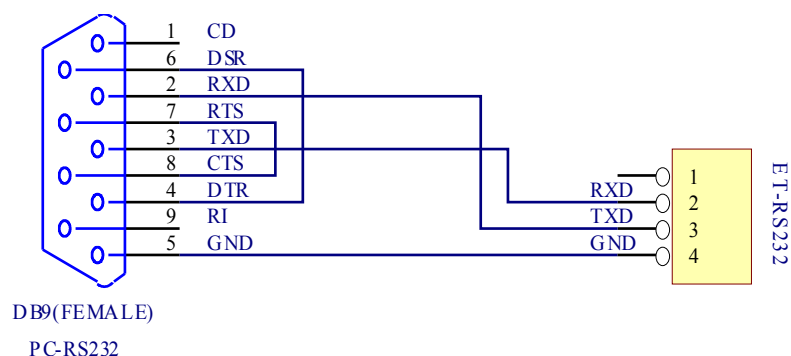
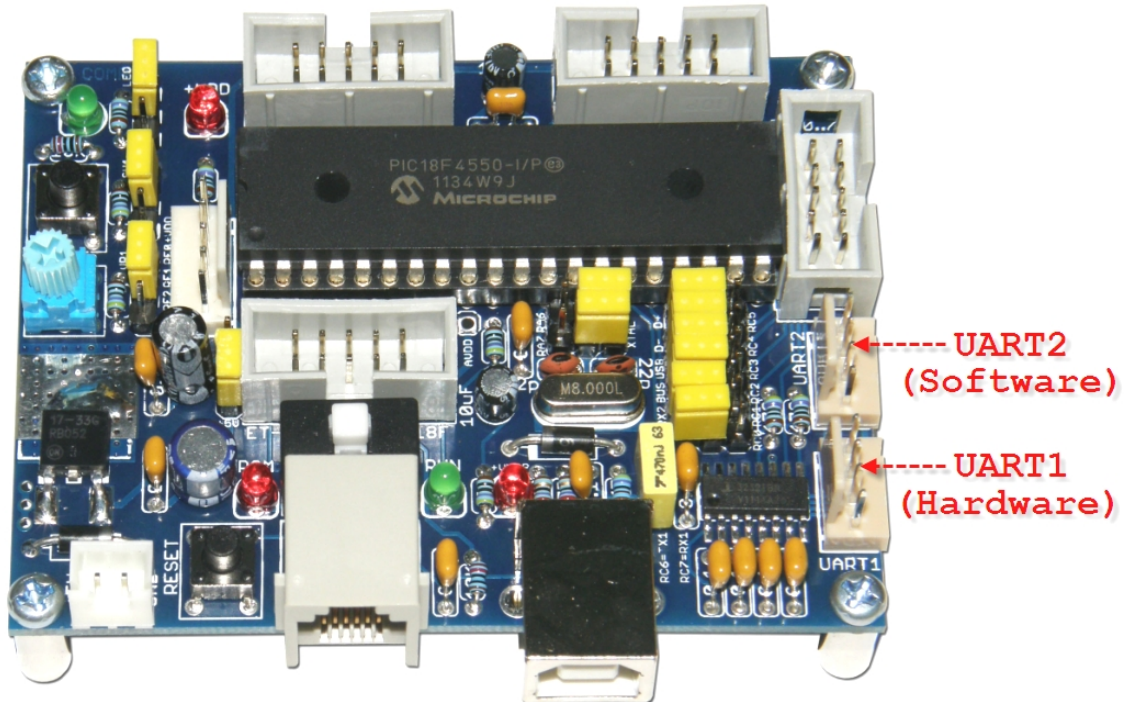


Figure shows circuit of CABLE RS232.



NOTE: When using UART2(Software UART) to write program, user should consider the capability of Compiler to see if it can support the Library that is Software UART. If using PIC CCS Compiler to compile the language, it can use the capability of Software UART easily. If user does not require using Pin RC0/RC1 to be UART2, user can set the function of both pins to be GPIO by setting Jumper as required. Because Circuit Driver in the part of UART2 is designed to be flexible structure; moreover, there is Jumper RC0/TX2 and RC1/RX2 to connect/disconnect signal as required.

```

/*****
/* Demo Program For ET-BASE PIC40/4550 UART Demo */
/* MCU Control : PIC18F4550 */
/*          : Run 48MHz(X-TAL 8.00 MHz + PLL) */
/* +VDD Power : +5V Operate Voltage */
/* Function   : Demo UART1, UART2 Echo Test */
/*****
#include <18F4550.h>
#include <stdlib.h>

// Fuses: PIC18F4550(CCS Compiler)
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL2, CPUDIV1, VREGEN
#use delay(clock=48000000)

/* Config and Enable Hardware UART1(RC6=TX1, RC7=RX1) */
#use rs232(uart1, baud=9600, stream=CH1)

/* Config and Enable Software UART2(RC0=TX2, RC1=RX2) */
#define TX2 PIN_C0
#define RX2 PIN_C1
#use rs232(baud=9600, xmit=TX2, rcv=RX2, stream=CH2)

```

```

/*****
/* Main Program */
*****/
void main()
{
    char rx_buff;

    //Start-Up UART1
    fprintf(CH1, "\n\n\n\r");
    fprintf(CH1, "Demo UART1 ET-BASE PIC40/4550(ICSP)\n\r");
    fprintf(CH1, "Run 48.00 MHz (Crystal 8 MHz + PLL)\n\r");
    fprintf(CH1, "UART1>");

    //Start-Up UART2
    fprintf(CH2, "\n\n\n\r");
    fprintf(CH2, "Demo UART2 ET-BASE PIC40/4550(ICSP)\n\r");
    fprintf(CH2, "Run 48.00 MHz (Crystal 8 MHz + PLL)\n\r");
    fprintf(CH2, "UART2>");

    while(true)
    {
        //Verify & Echo UART1
        if(kbhit(CH1))
        {
            rx_buff = fgetc(CH1);
            if (rx_buff == 0x0D)
            {
                fprintf(CH1, "\n\r");
                fprintf(CH1, "ET-BASE PIC40/4550(ICSP)\n\r");
                fprintf(CH1, "UART1>");
            }
            else
            {
                fputc(rx_buff, CH1);           // Echo Received Characters
            }
        }

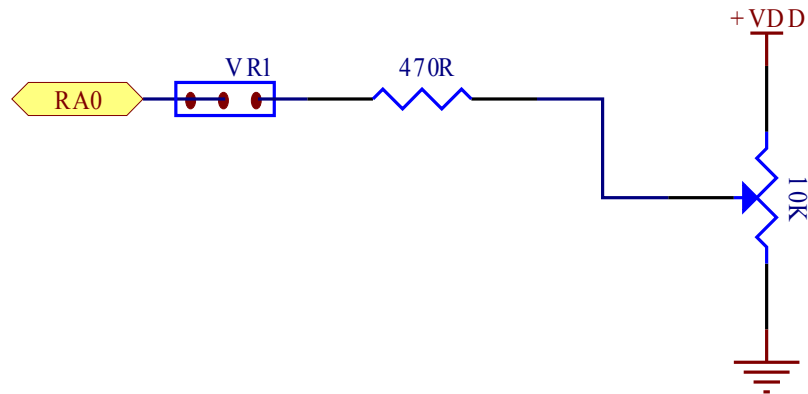
        //Verify & Echo UART2
        if(kbhit(CH2))
        {
            rx_buff = fgetc(CH2);
            if (rx_buff == 0x0D)
            {
                fprintf(CH2, "\n\r");
                fprintf(CH2, "ET-BASE PIC40/4550(ICSP)\n\r");
                fprintf(CH2, "UART2>");
            }
            else
            {
                fputc(rx_buff, CH2);           // Echo Received Characters
            }
        }
    }
}

```

This example shows Code to communicate with C Language UART (PICC CCS Compiler).

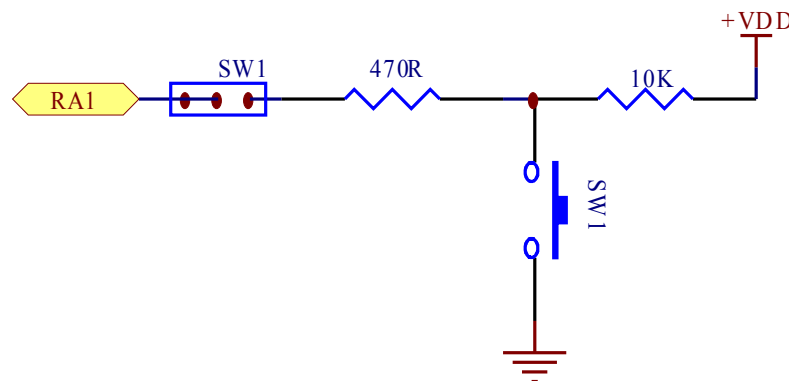
Adjustable VR

This VR1 is Adjustable Resistor to adjust voltage level between +VDD and GND to create the voltage level for testing the operation of Analog Input (ADC). The adjustable voltage of VR1 is connected to Pin RA0 of MCU and there is Jumper to connect/disconnect signal; so, user can set this Jumper for testing the operation independently.



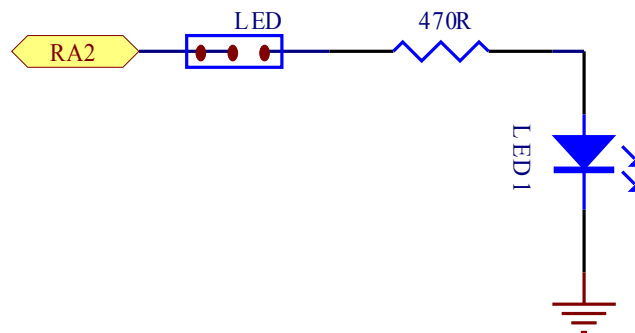
Circuit SW1

This SW1 is Circuit Push-Button Switch to create Signal LOGIC "0" and "1" to test the operation of LOGIC INPUT; for example, it is used to detect the value of pressing Switch. If the Switch is pressed, it is Logic "1"; on the other hand, if the Switch is released, it is Logic "0". The Signal Logic from this circuit is connected to Pin RA1 of MCU; moreover, there is Jumper to connect/disconnect Signal Logic from pressing Switch SW1 independently.

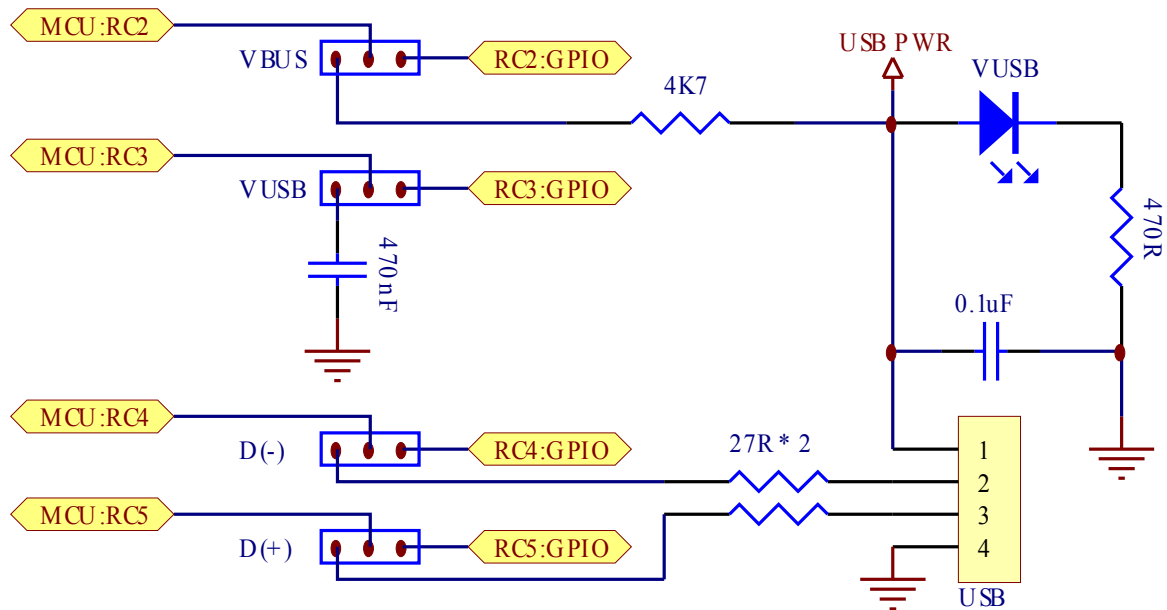


Circuit LED

LED is Circuit LOGIC Display that displays the logic result to user. It is used with Signal Logic Output; if it receives the Signal Logic "1", it makes LED ON; on the other hand, it receives the Signal Logic "0", it makes LED OFF. This Signal Logic is used to drive the LED Display in this circuit; it is connected from Pin RA2 of MCU. Moreover, there is Jumper to connect/disconnect Signal Logic from Pin RA2 to drive this LED independently.



USB



In case of using MCU that has the USB Hardware System; it uses Pin RC3, RC4 and RC5 to be signal for interfacing with USB. In this case, Pin 18 (correspond with pin position of RC3 in normal MCU) of MCU is used to be function VUSB; RC4 and RC5 are used to be function USB Data that are USB D(-) and USB D(+), respectively.

RC2 is Option that is used to detect VBUS of USB; user can use or does not use this option as preferred. If using this option, it has to set this Pin to be Function Digital Input to detect Signal Logic from VBUS of USB. If interfacing USB Cable with Connector USB Host, it supplies +5V from USB Host; so, Pin RC2 receives Signal Logic "1". On the other hand, if removing the USB Cable from the connector, Pin RC2 receives Signal Logic "0". In this case, it uses Pin RC2 to detect if USB Cable is interfacing or it is removed from USB Host.

This Circuit has LED(+VBUS) to display status of +5V Power Supply from Connector USB. If this LED it status ON, it means that USB Cable is interfacing with Host; on the other hand, if this LED is in status OFF, it means that there is no any +5V Power Supply from USB Host.

How to Setup Configuration of Board

It uses Crystal 8MHz to be basic frequency for Board ET-BASE PIC40/4550(ICSP); when developing program, user has to setup the Configuration value of MCU according to this Crystal Frequency (8.00MHz) and finally, it makes MCU operate correctly. MCU can process data with the maximum of 48MHz; so, user has to set Phrase Lock Loop (PLL) to multiply Crystal 8.00MHz and it gets the result of 48MHz. There are several different methods to setup the Configuration value of each Compiler as shown in the example below;

```
// Fuses: PIC18F4550(CCS Compiler)
//////// Fuses: PLL1,PLL2,PLL3,PLL4,PLL5,PLL6,PLL10,PLL12,CPUDIV1,CPUDIV2
//////// Fuses: CPUDIV3,CPUDIV4,NOUSB DIV,USB DIV,XT,XTPLL,EC_IO,EC,ECPLL_IO
//////// Fuses: ECPLL,INTEC_IO,INTEC,INTXT,INTHS,HS,HSPLL,NOFCMEN,FCMEN
//////// Fuses: NOIESO,IESO,PUT,NOPUT,NOBROWNOUT,BROWNOUT_SW,BROWNOUT_NOSL
//////// Fuses: BROWNOUT,BORV45,BORV43,BORV27,BORV20,NOVREGEN,VREGEN,NOWDT
//////// Fuses: WDT,WDT1,WDT2,WDT4,WDT8,WDT16,WDT32,WDT64,WDT128,WDT256
//////// Fuses: WDT512,WDT1024,WDT2048,WDT4096,WDT8192,WDT16384,WDT32768
//////// Fuses: CCP2B3,CCP2C1,NOPBADEN,PBADEN,NOLPT1OSC,LPT1OSC,NOMCLR,MCLR
//////// Fuses: NOSTVREN,STVREN,NOLVP,LVP,ICSP1,ICSP2,NOXINST,XINST,DEBUG
//////// Fuses: NODEBUG,PROTECT,NOPROTECT,CPB,NOCPB,CPD,NOCPD,WRT,NOWRT,WRTC
//////// Fuses: NOWRTC,WRTB,NOWRTB,WRTD,NOWRTD,EBTR,NOEBTR,EBTRB,NOEBTRB

//Select Jumper RA6,RA7(XTAL/GPIO) = X-TAL
//if used XTAL=8MHz -> 8MHz/2=4MHz -> 4MHzxPLL(96MHz)/2=48MHz
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USB DIV,PLL2,CPUDIV1,VREGEN
#use delay(clock=48000000)
```

This example shows how to setup Configuration values of PIC 18F4550 by PIC CCS Compiler.

```
#pragma config PLLDIV = 2 // (8 MHz crystal)
#pragma config CPUDIV = OSC1_PLL2
#pragma config USB DIV = 2 // Clock source from 96MHz PLL/2
#pragma config FOSC = HSPLL_HS // Clock Source = HS+PLL
#pragma config FCMEN = OFF
#pragma config IESO = OFF
#pragma config PWRT = ON
#pragma config BOR = ON
#pragma config BORV = 0
#pragma config VREGEN = ON // USB Voltage Regulator
#pragma config WDT = OFF
#pragma config WDTPS = 32768
#pragma config MCLRE = ON
#pragma config LPT1OSC = OFF
#pragma config PBADEN = OFF
#pragma config CCP2MX = OFF
#pragma config STVREN = ON
#pragma config LVP = OFF
#pragma config ICPRT = ON // ***Enable ICSP/Programming***
#pragma config XINST = OFF // Extended Instruction Set
```

This example shows how to setup Configuration values of PIC 18F4550 by PIC C18 Compiler.

How to Setup Configuration by PIC Basic PRO Compiler

In case of using BASIC Language (PIC Basic Pro Compiler), it has to setup Configuration values for setting operation of MCU through File "C:\PBP\18F4550.INC".

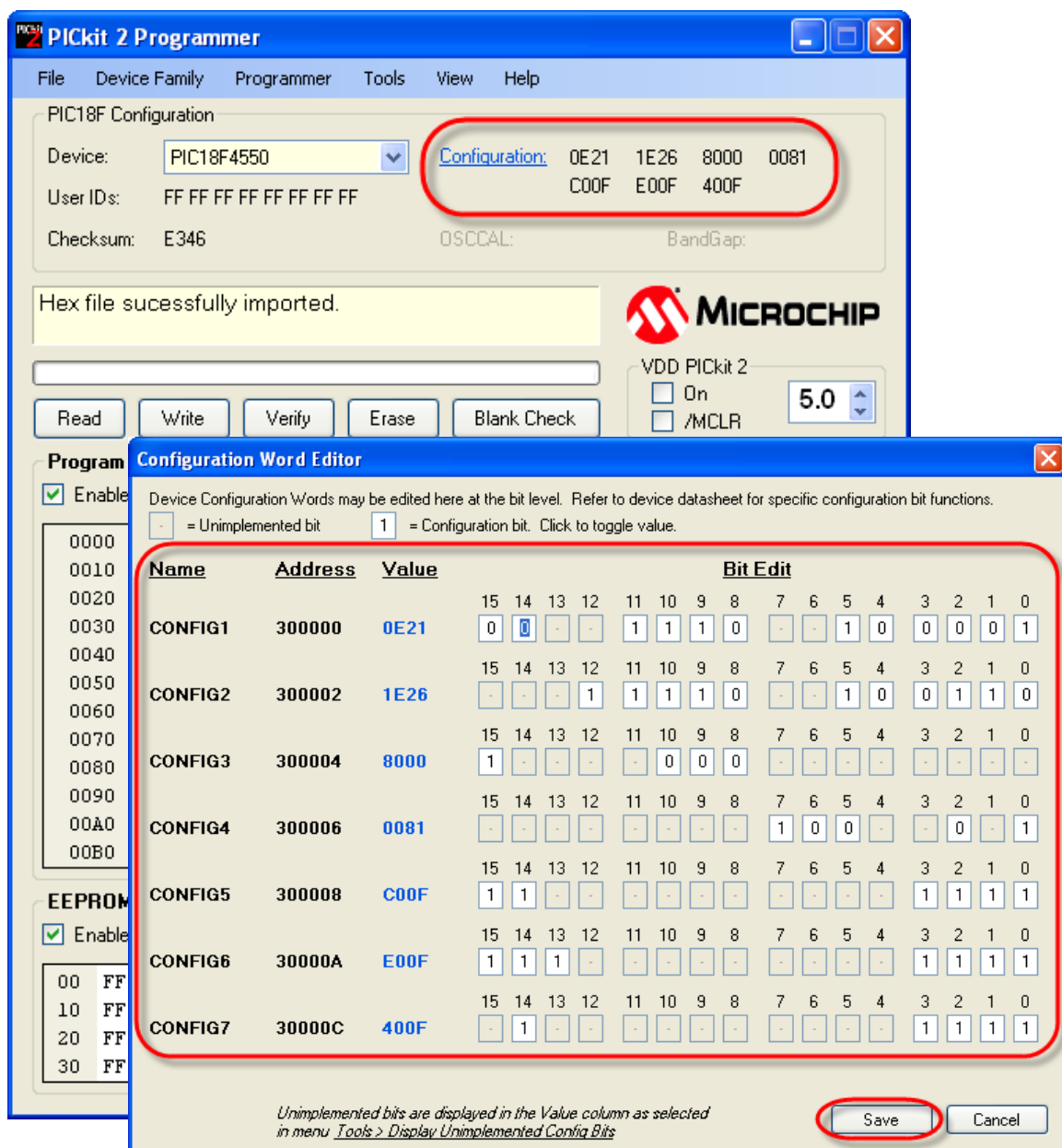
```
'ET-BASE PIC40/4550(ICSP) Hardware used X-TAL=8MHz
'Setup Clock Config -> 8MHz/2=4MHz -> 4MHzxPLL(96MHz)->96MHz/2=48MHz
'Config1 = 0x0E21 = 0000 1110 : 0010 0001
'Config2 = 0x1E26 = 0001 1110 : 0010 0110
'Config3 = 0x8000 = 1000 0000 : 0000 0000
'Config4 = 0x0081 = 0000 0000 : 1000 0001
'Config5 = 0xC00F = 1100 0000 : 0000 1111
'Config6 = 0xE00F = 1110 0000 : 0000 1111
'Config7 = 0x400F = 0100 0000 : 0000 1111

'Start of Config ET-BASE PIC40/4550 Hardware For PIC18F4550 USBHID Demo
NOLIST
#ifdef PM_USED
LIST
"Error: PM does not support this device. Use MPASM."
NOLIST
else
LIST
LIST p = 18F4550, r = dec, w = -311, w = -230, f = inh32
INCLUDE "P18F4550.INC" ; MPASM Header
__CONFIG __CONFIG1L,_PLLDIV_2_1L& CPUDIV_OSC1_PLL2_1L& USBDIV_2_1L
__CONFIG __CONFIG1H,_FOSC_HSPLL_HS_1H & _FCMEN_OFF_1H& _IESO_OFF_1H
__CONFIG __CONFIG2L,_PWRT_ON_2L& BOR_ON_2L& BORV_0_2L& VREGEN_ON_2L
__CONFIG __CONFIG2H,_WDT_OFF_2H & _WDTPS_32768_2H
__CONFIG __CONFIG3H,_CCP2MX_OFF_3H& _PBADEN_OFF_3H& _LPT1OSC_OFF_3H& _MCLRE_ON_3H
__CONFIG __CONFIG4L,_STVREN_ON_4L& _LVP_OFF_4L& _ICPRT_OFF_4L& _XINST_OFF_4L& _DEBUG_OFF_4L
__CONFIG __CONFIG5L,_CP0_OFF_5L& _CP1_OFF_5L& _CP2_OFF_5L& _CP3_OFF_5L
__CONFIG __CONFIG5H,_CPB_OFF_5H& _CPD_OFF_5H
__CONFIG __CONFIG6L,_WRT0_OFF_6L& _WRT1_OFF_6L& _WRT2_OFF_6L& _WRT3_OFF_6L
__CONFIG __CONFIG6H,_WRTB_OFF_6H& _WRTC_OFF_6H& _WRTD_OFF_6H
__CONFIG __CONFIG7L,_EBTR0_OFF_7L& _EBTR1_OFF_7L& _EBTR2_OFF_7L& _EBTR3_OFF_7L
__CONFIG __CONFIG7H,_EBTRB_OFF_7H
NOLIST
#endif
LIST
```

This example shows how to setup Configuration values of PIC 18F4550 by PIC Basic PRO Compiler.

How to Setup Configuration by Pickit2 Programmer

In case of using Pickit2 Programmer, user can check the Configuration values from program in step of programming. If the Configuration value that has been compiled and included in HEX File has been loaded into the program, the Configuration value on the window program should be changed according to the setting value. If it does not accord with the setting values or user forgets setting the Configuration values in Source Code, user can edit and change the new Configuration values by self in the window of Program Pickit2 as shown in the example below.



Picture shows the feature of Configuration values of PIC18F4550 of Program Pickit2.

ICSP

ICSP is Connector RJ11 to interface with Program Developer Kit in the series of PIC and it arranges the connector according the standard of ICSP of MICROCHIPS such as ICD2, ICD3, Pickit2 or Pickit3. It is compatible with Program Developer kit of MICROCHIPS or equivalence such as ET-PGMPIC USB (be equivalent to PICKit2), or ET-PGMPIC PK3 (be equivalent to PICKit3), or ET-ICDX (be equivalent to ICD2). When user requires using any device version, user should consider the MCU number mainly because each MCU number supports different programmer device and it has different capability; however, it only arranges pin for programming MCU in the same standard. The circuit of board in this part has Switch to connect/disconnect Signal of RB6, RB7, and MCLR for interfacing with Programmer/Debugger or Normal Run; moreover, there is LED to display the current position of LED that is running. If it shifts the Switch to the side of Programmer/Debugger, the red LED of PGM is in status ON; on the other hand, if it shifts the Switch to the side of Normal Run, the green LED of RUN is in status of ON. It arranges pins according to the standard of ICSP as follows;

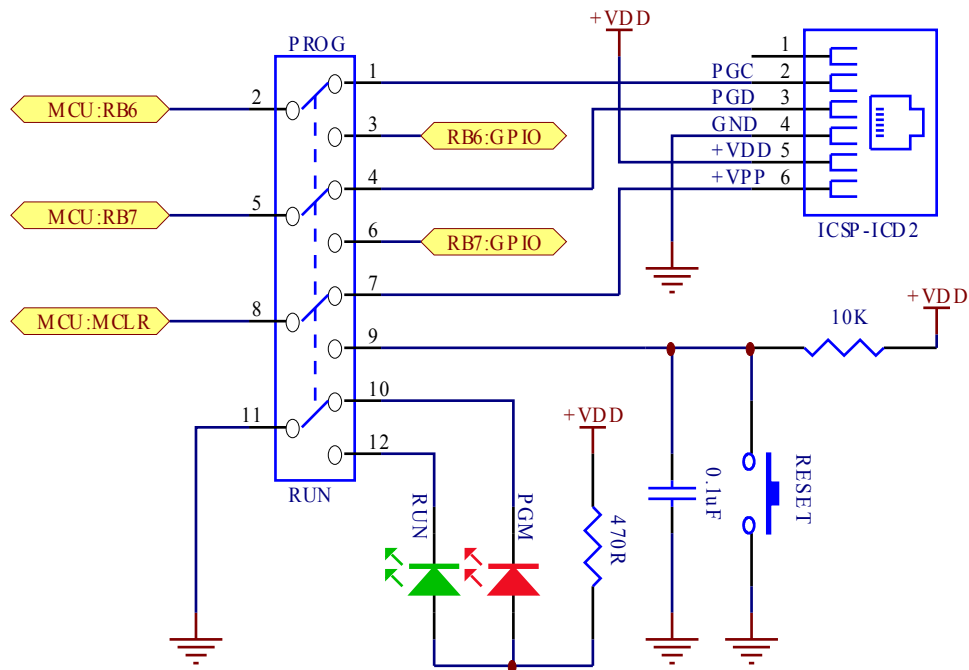


Figure displays structure of circuit in the part of connecting with ICSP of Board ET-BASE PIC40/4550 (ICSP) .

POWER SUPPLY

This Circuit Power Supply of this board is compatible with external +5VDC. The circuit of board can receive 2 sources of +5VDC; from Connector 2Pin and Connector USB (in case of using with MCU that has USB). Internal board has Circuit Regulate 3.3V/1A with Jumper to choose the voltage types between +5VDC and +3V3 to be voltage for MCU(+VDD) and Circuit I/O internal board. In case of Board ET-BASE PIC40/4550(ICSP), it always sets as 5V.

