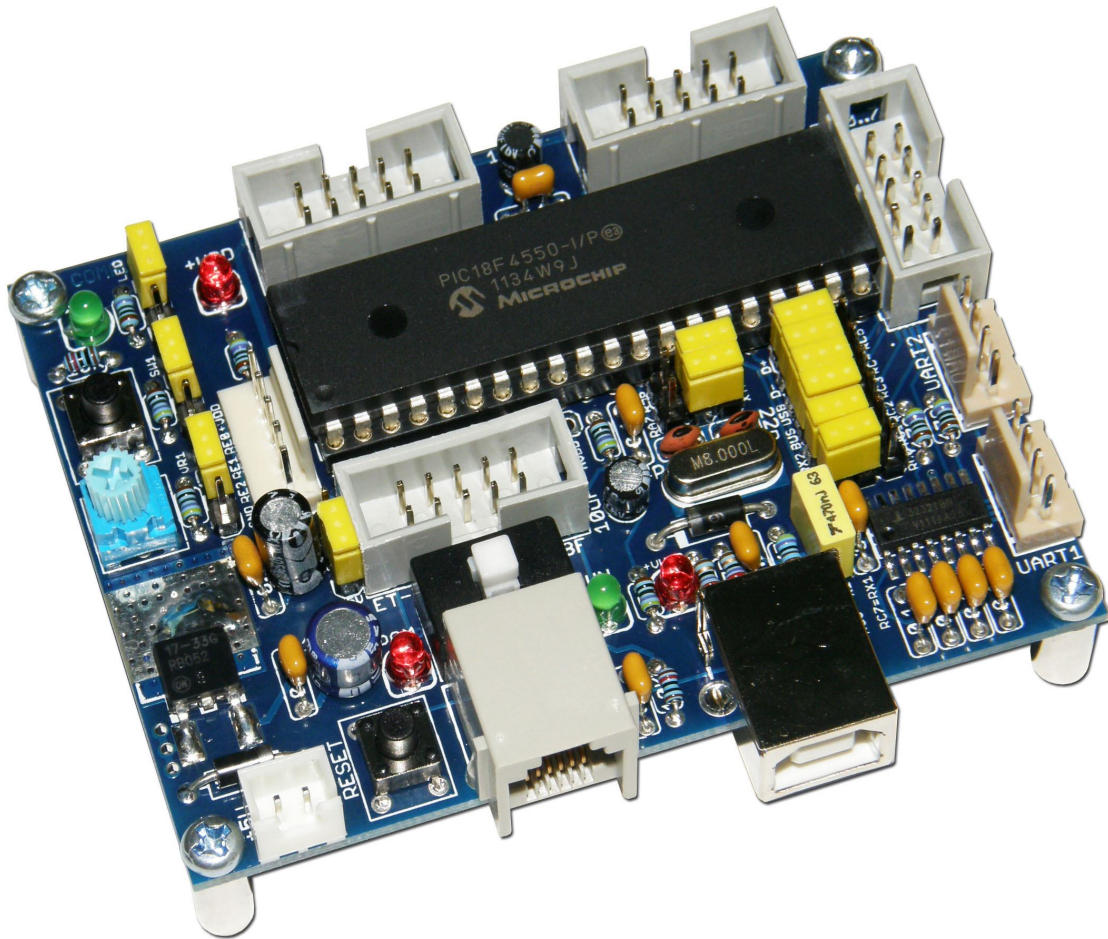


ET-BASE PIC40/4550(ICSP)

ET-BASE PIC40/4550(ICSP) เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล PIC ซึ่งออกแบบให้รองรับการติดตั้งใช้งานกับชิพไมโครคอนโทรลเลอร์ 40 Pin เบอร์ PIC18F4550 รองรับการพัฒนาโปรแกรมของบอร์ดให้ทำหน้าที่เป็นอุปกรณ์ USB Device แบบต่างๆ ได้โดยง่าย

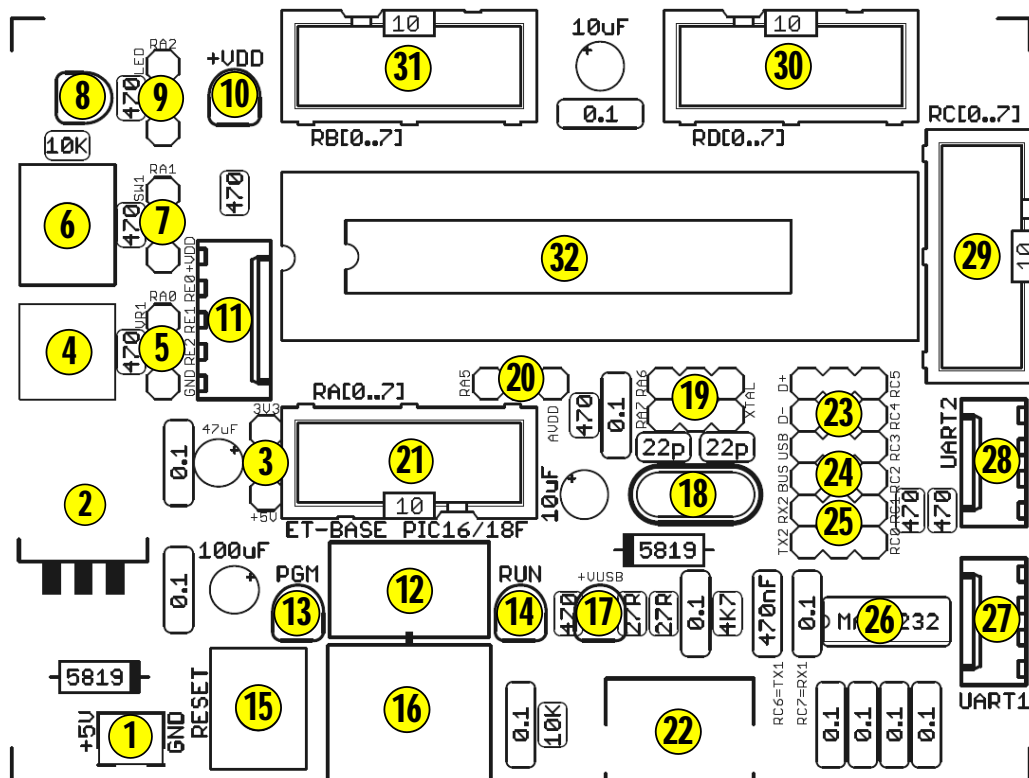
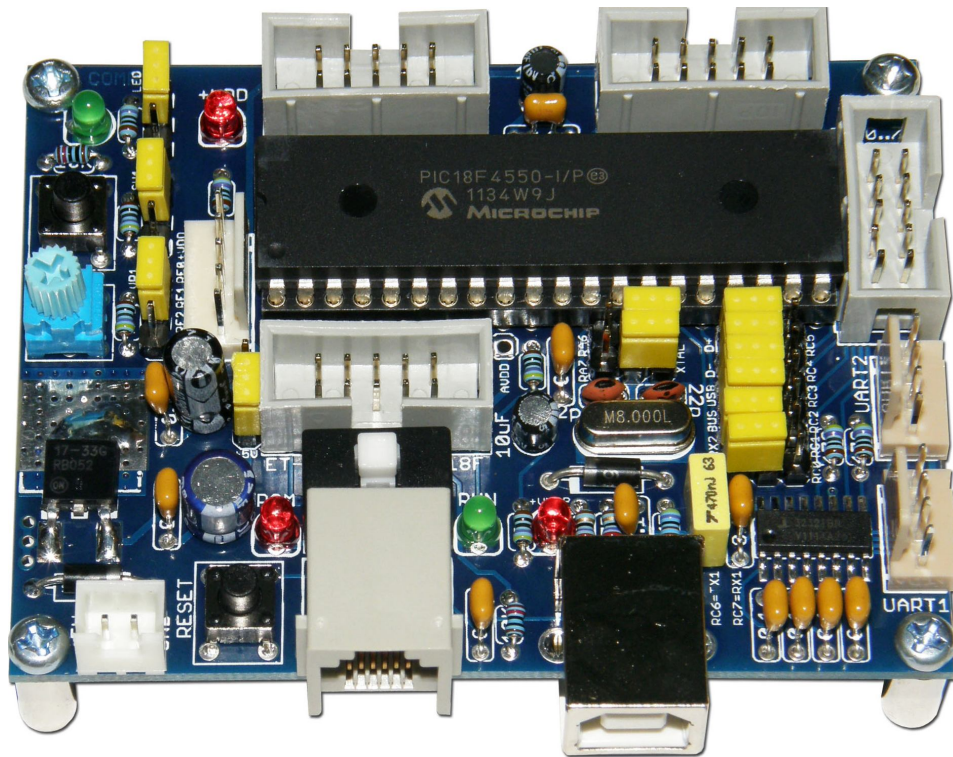
โดยโครงสร้างของบอร์ด ET-BASE PIC40/4550(ICSP) ได้รับการออกแบบให้บอร์ดมีขนาดเล็กเหมาะต่อการนำไปประยุกต์ใช้งานและยังสามารถใช้เป็นชุดทดลองเรียนรู้เบื้องต้นได้อีกด้วย โดยภายในบอร์ดได้บรรจุเอาวางจที่จำเป็นต่อการใช้งาน และสะดวกต่อการพัฒนาโปรแกรม มีความยืดหยุ่น สามารถปรับเปลี่ยนสัญญาณ I/O เพื่อนำไปประยุกต์ใช้งานในลักษณะต่างๆ ให้สอดคล้องและเหมาะสมกับความต้องการใช้งานได้ในหลายๆ ลักษณะตามต้องการ

นอกจากนี้แล้วยังได้เพิ่มเติมอุปกรณ์ทดสอบ Input/Output แบบต่างๆ ในเบื้องต้นไว้ภายในบอร์ด เพื่อให้ผู้ใช้ได้ใช้เป็นเครื่องมือในการทดลองระหว่างการพัฒนาโปรแกรม เช่น VR ปรับค่าได้สำหรับทดสอบการทำงานของ ADC หรือ Push Button Switch สำหรับทดสอบ Input Logic หรือ LED สำหรับใช้ทดสอบการทำงานของ Output Logic เป็นต้น

คุณสมบัติของบอร์ด

- ใช้ MCU ตระกูล PIC ขนาด 40Pin(40PDIP) เบอร์ PIC18F4550 เป็น MCU ประจำบอร์ด
 - 32KByte Flash / 2048 Byte SRAM / 256 Byte EEPROM
 - 35 GPIO
 - 1 Channel USB V2.0 Compliant
 - 13 Channel 10Bit ADC
 - 1 Channel CCP / 1 Channel ECCP
 - 1 Channel I2C / 1 Channel SPI
 - 1 Channel UART
 - 2 Channel Comparator
 - 1 Channel 8 Bit Timer / 3 Channel 16 Bit Timer
- ใช้ Crystal ความถี่ 8.00MHz พร้อม Jumper ตัดต่อสัญญาณใช้หรือไม่ใช้งานได้
- มีวงจร Line Driver สำหรับพอร์ตสื่อสารอนุกรม UART แบบ RS232 จำนวน 2 ช่อง โดยใช้ขั้วต่อ UART แบบ CPA-4 Pin มาตรฐาน อีทีที
 - 1 ช่อง สำหรับ Hardware UART1 โดยใช้ Pin RC6(TX1) และ RC7(RX1) มาตรฐาน PIC
 - 1 ช่อง สำหรับ Software UART โดยใช้ Pin RC0(TX2) และ RC1(RX2) พร้อม Jumper สำหรับเลือกใช้งาน UART2(Software UART) หรือ GPIO ได้ตามต้องการ
- มีขั้ว ICSP มาตรฐาน ICD2 แบบ RJ11 สำหรับใช้ร่วมกับชุดพัฒนาโปรแกรมและ Debugger ที่รองรับการทำงานตามมาตรฐาน ICD2 ของ Microchips เช่น ICD2/ICD3 หรือ Pickit2/Pickit3 ได้
- มี Switch สำหรับสลับสัญญาณระหว่าง Program/Debug(PGM) และ ใช้งานปกติ(RUN) พร้อม LED แสดงโหมดการทำงานของบอร์ด
- มีขั้วต่อสัญญาณ I/O แบบ Header ขนาด 2x5 จำนวน 4 ชุด และ Header CPA-5 Pin อีก 1 ชุด
- มี Switch Reset สำหรับสั่ง Reset การทำงานของ MCU ภายในบอร์ด
- มี VR ปรับค่าสำหรับทดสอบการทำงาน ADC Input โดยใช้ RA0 พร้อม Jumper ตัดต่อสัญญาณ
- มี Switch สำหรับทดสอบการทำงาน Digital Input โดยใช้ RA1 พร้อม Jumper ตัดต่อสัญญาณ
- มี LED สำหรับทดสอบการทำงาน Digital Output โดยใช้ RA2 พร้อม Jumper ตัดต่อสัญญาณ
- Power +5VDC Input พร้อม Regulate แบบ 3.3V/1A และ LED แสดงสถานะแหล่งจ่าย Power พร้อม Jumper สำหรับ เลือกแหล่งจ่ายไฟเลี้ยงให้ MCU ว่าจะใช้เป็น +5VDC หรือ 3.3VDC ซึ่งตามปกติ จะเลือกเป็น +5V เสมอ
- มีวงจรเชื่อมต่อ USB โดยใช้ขั้ว USB-B พร้อม LED แสดงสถานะ +VBUS(USB)
- ขนาด PCB Size เล็กเพียง 8 x 6 cm.

โครงสร้างบอร์ด ET-BASE PIC40/4550(ICSP)



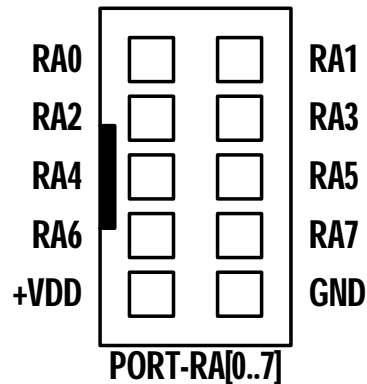
รูปแสดง โครงสร้างของบอร์ด ET-BASE PIC40/4550(ICSP)

- หมายเลข 1 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟ +5VDC
- หมายเลข 2 คือ IC Regulate ขนาด 3.3VDC/1A
- หมายเลข 3 คือ Jumper สำหรับเลือกขนาดแรงดันไฟเลี้ยง MCU(+VDD) ระหว่าง 3.3V หรือ 5V
- หมายเลข 4 คือ VR(VR1) ปรับค่าแรงดัน สำหรับใช้ทดสอบการทำงานของ Input Analog(ADC)
- หมายเลข 5 คือ Jumper สำหรับ ตัดต่อ สัญญาณ RA0 กับแรงดันปรับค่าจาก VR1
- หมายเลข 6 คือ Switch(SW1) กดติดปล่อยดับ สำหรับใช้ทดสอบการทำงานของ Digital Input
- หมายเลข 7 คือ Jumper สำหรับตัดต่อสัญญาณ RA1 กับ Digital Input จาก SW1
- หมายเลข 8 คือ LED สำหรับใช้ทดสอบการทำงานของ Digital Output
- หมายเลข 9 คือ Jumper สำหรับตัดต่อสัญญาณ RA2 กับ Digital Output ให้กับ LED
- หมายเลข 10 คือ LED แสดงสถานะของแหล่งจ่ายไฟ +VDD ของบอร์ด
- หมายเลข 11 คือ ขั้วต่อสัญญาณ RE[0..2] ซึ่งจะถูกใช้งานในกรณีติดตั้ง MCU รุ่น 40Pin เท่านั้น
- หมายเลข 12 คือ สวิตช์ สำหรับเลือกโหมดการทำงานระหว่าง Run(RUN) และ Program(PGM)
- หมายเลข 13 คือ LED สีแดง แสดงสถานะ PGM เมื่อบอร์ดทำงานใน Program Mode
- หมายเลข 14 คือ LED สีเขียว แสดงสถานะ RUN เมื่อบอร์ดทำงานใน Run Mode
- หมายเลข 15 คือ สวิตช์ Reset สำหรับ Reset การทำงานของ MCU เมื่ออยู่ในโหมด Run
- หมายเลข 16 คือ ขั้วต่อ ICSP สำหรับใช้เชื่อมต่อกับเครื่องโปรแกรมและดีบั๊กตามมาตรฐาน ICD2
- หมายเลข 17 คือ LED แสดงสถานะของ +VUSB
- หมายเลข 18 คือ Crystal ค่าความถี่ 8.00MHz
- หมายเลข 19 คือ Jumper สำหรับเลือกขาสัญญาณ RA6,RA7 ของ MCU ว่าจะให้เชื่อมต่อไปเป็น GPIO ที่ขั้ว RA[0..7] หรือวงจร Crystal สำหรับบอร์ด ET-BASE PIC40/4550(ICSP) จะเลือกไว้ทางด้าน Crystal เสมอ
- หมายเลข 20 คือ Jumper สำหรับเลือกหน้าที่ของ Pin7 ของ MCU รุ่น 28Pin ว่าเป็นสัญญาณ RA5 ของ MCU โดยให้เชื่อมต่อไปเป็น GPIO ที่ขั้ว RA[0..7] หรือเป็นขา +AVDD แต่สำหรับในกรณีของบอร์ด ET-BASE PIC40/4550(ICSP) จะกำหนดคงที่เป็น RA5 เสมอ
- หมายเลข 21 คือ ขั้วต่อ IDE10Pin ของ RA[0..7]
- หมายเลข 22 คือ ขั้วต่อ USB-B
- หมายเลข 23 คือ Jumper สำหรับเลือกขาสัญญาณ RC3,RC4,RC5 ของ MCU ว่าจะให้เชื่อมต่อไปเป็น GPIO ที่ขั้ว RC[0..7] หรือใช้เป็นสัญญาณเชื่อมกับกับ USB Bus(VUSB,D-,D+)

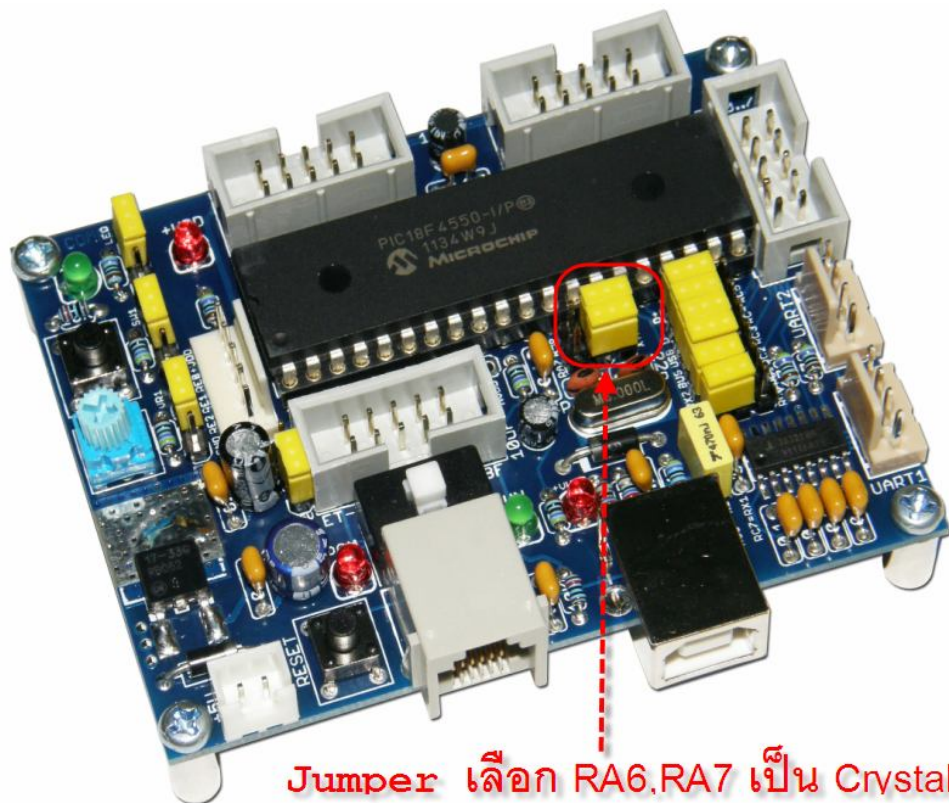
- หมายเลข **24** คือ Jumper สำหรับเลือกขาสัญญาณ RC2 ของ MCU ว่าจะให้เชื่อมต่อไปเป็น GPIO ที่ขั้ว RC[0..7] หรือเป็นขาตรวจจับ VBUS ของ USB
- หมายเลข **25** คือ Jumper สำหรับเลือกขาสัญญาณ RC0,RC1 ของ MCU ว่าจะให้เชื่อมต่อไปเป็น GPIO ที่ขั้ว RC[0..7] หรือ TX2,RX2 ของ UART2(Software UART)
- หมายเลข **26** คือ ไอซี Line Drive ของ RS232(MAX3232) สำหรับแปลง Level ระหว่างสัญญาณของ UART Logic กับ สัญญาณมาตรฐาน RS232
- หมายเลข **27** คือ ขั้วต่อ UART1 โดยเป็นสัญญาณแบบ RS232 รองรับ Hardware UART ซึ่งใช้ Pin ของ RC6(TX1) และ RC7(RX1) เป็นสัญญาณเชื่อมต่อ
- หมายเลข **28** คือ ขั้วต่อ UART2 โดยเป็นสัญญาณแบบ RS232 รองรับ Software UART ซึ่งใช้ Pin ของ RC0(TX2),RC1(RX2) เป็นสัญญาณเชื่อมต่อ
- หมายเลข **29** คือ ขั้วต่อ IDE10Pin ของ RC[0..7]
- หมายเลข **30** คือ ขั้วต่อ IDE10Pin ของ RD[0..7]
- หมายเลข **31** คือ ขั้วต่อ IDE10Pin ของ RB[0..7]
- หมายเลข **32** คือ MCU ปรจําบอร์ด เบอร์ PIC18F4550

หัวข้อสัญญาณต่างๆ

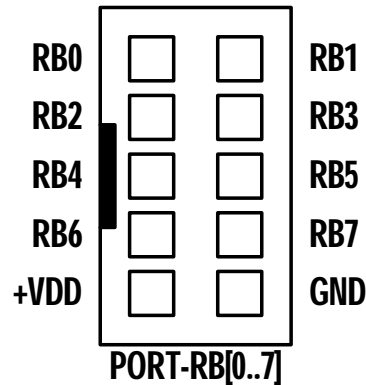
PORT RA[0..7]



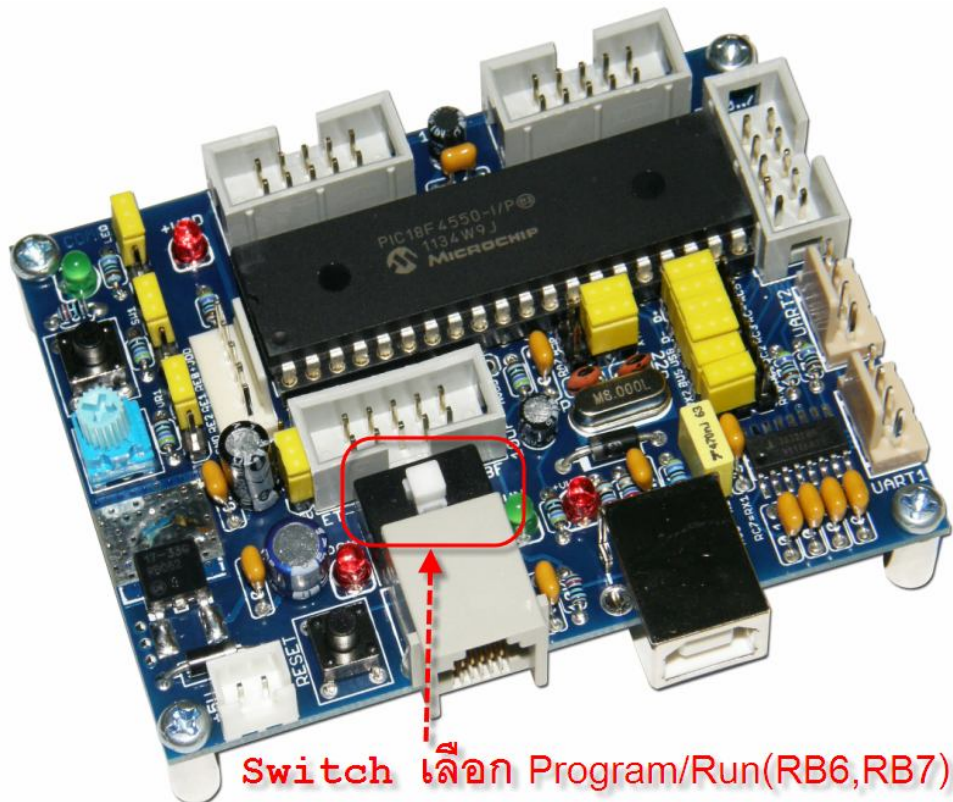
PORT-RA[0..7] เป็นสัญญาณจาก RA0...RA7 ของ MCU โดย RA0...RA5 จะเป็นสัญญาณที่ต่อตรงมาจาก Pin ของ MCU โดยตรง ส่วนสัญญาณ RA6 และ RA7 จะมี Jumper สำหรับเลือกตัดต่อสัญญาณร่วมอยู่ด้วยว่าต้องการใช้ RA6,RA7 เป็น GPIO หรือ ใช้เชื่อมต่อกับวงจรกำเนิดความถี่ Crystal ซึ่งในกรณีของ PIC18F4550 สัญญาณ RA6,RA7 จะถูกใช้ต่อกับวงจรกำเนิดความถี่ Crystal ให้เสมอ

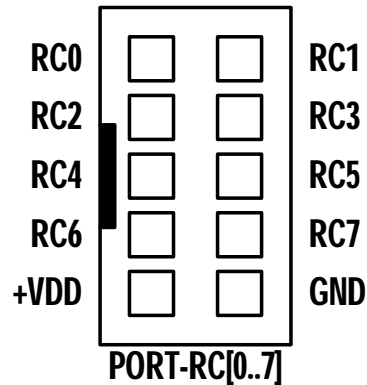


PORT RB[0...7]



PORT-RB[0..7] เป็นสัญญาณจาก RB0...RB7 ของ MCU โดย RB0...RB5 จะเป็นสัญญาณที่ต่อตรงมาจาก Pin ของ MCU โดยตรง ส่วนสัญญาณ RB6 และ RB7 จะมี Switch สำหรับเลือกตัดต่อสัญญาณร่วมอยู่ด้วยว่าต้องการใช้ RB6,RB7 เป็น GPIO หรือ ใช้เชื่อมต่อเครื่องโปรแกรมแบบ ICSP ทางหัว ICSP(RJ11) โดยถ้าเลือกสวิตช์ไว้ในตำแหน่ง RUN จะเป็นการเชื่อมต่อขา RB6,RB7 มายังหัวต่อนี้ แต่ถ้าเลือกสวิตช์ไว้ในตำแหน่ง PGM จะเป็นการเลือกเชื่อมต่อสัญญาณ RB6,RB7 ของ MCU เข้ากับเครื่องโปรแกรม ผ่านทางหัว ICSP(RJ11) แทน

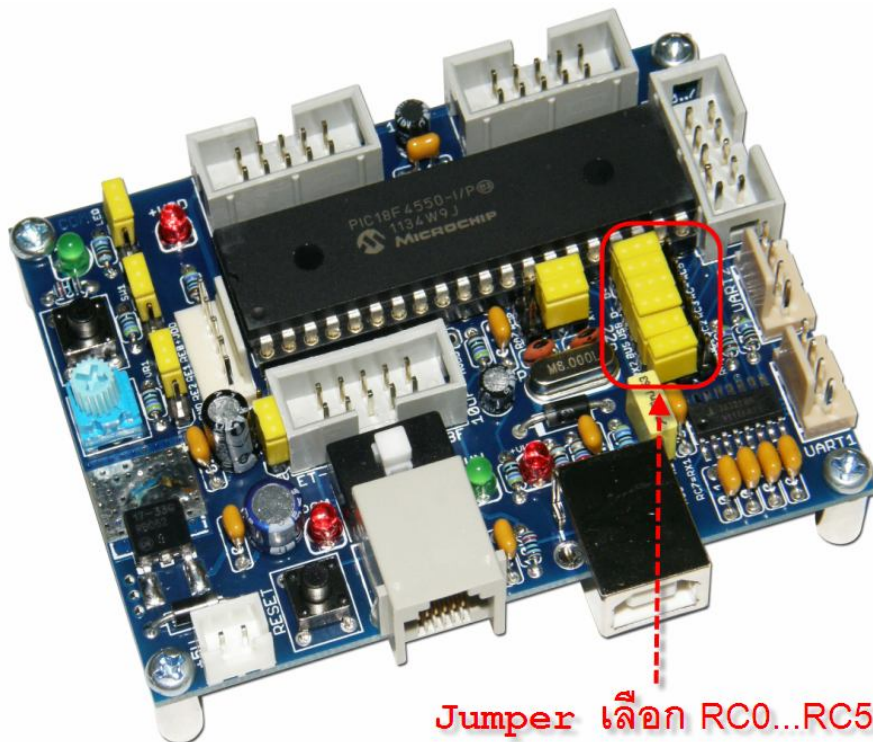


PORT-RC[0...7]

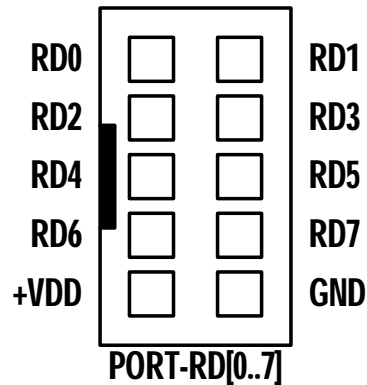
PORT-RC[0..7] เป็นสัญญาณจาก RC0...RC7 ของ MCU โดย RC0...RC5 จะเป็นสัญญาณที่ผ่านการเลือกจาก Jumper มาก่อน ส่วน RC6 และ RC7 จะเป็นสัญญาณที่ต่อตรงจาก Pin ของ MCU

RC0,RC1 จะมี Jumper สำหรับเลือกจะให้ขา RC0 และ RC1 เชื่อมต่อมายัง Connector นี้ หรือจะเชื่อมต่อ RC0,RC1 ไปยังวงจร Line Driver ของ UART2 (Software UART)

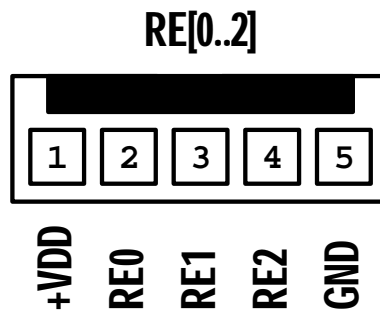
RC2...RC5 จะมี Jumper สำหรับเลือกจะให้ขา RC2...RC5 เชื่อมต่อมายัง Connector นี้ หรือจะเชื่อมต่อไปยังวงจรของการเชื่อมต่อกับ USB ซึ่งในกรณีของ PIC16F1939 ซึ่งโครงสร้างภายใน MCU ไม่มีวงจรการเชื่อมต่อกับ USB อยู่ด้วย ต้องเลือกกำหนด Jumper ของ RC2, RC3, RC4, RC5 ไว้ทางด้าน GPIO เพื่อเชื่อมต่อขา RC2, RC3, RC4, RC5 มายัง Connector นี้เสมอตามรูป



Jumper เลือก RC0...RC5

PORT RD[0...7]

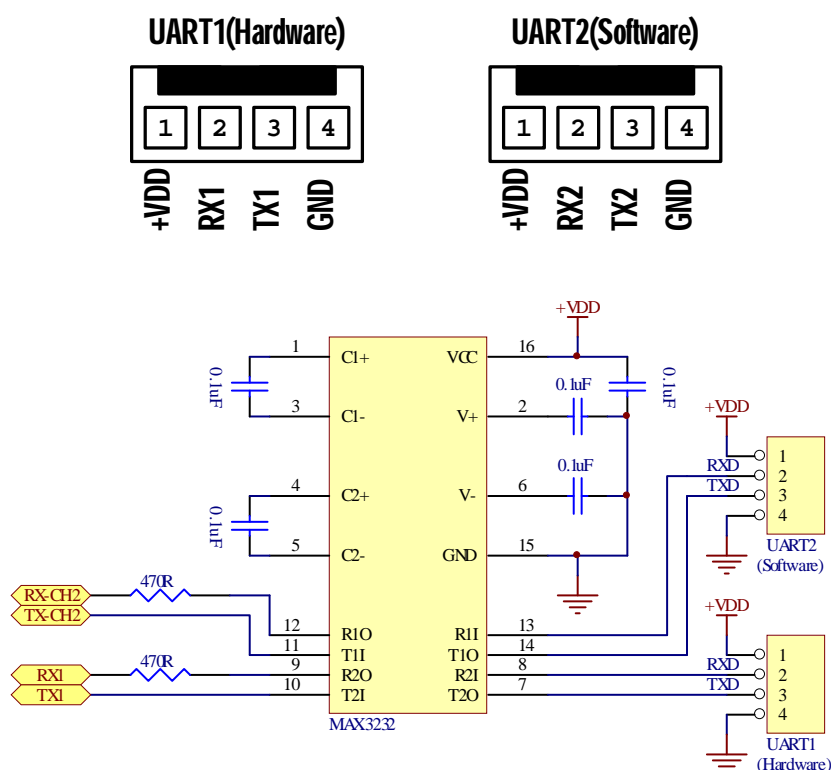
PORT-RD[0..7] เป็นสัญญาณจาก RD0...RD7 ของ MCU โดยเป็นสัญญาณที่ต่อตรงมาจาก Pin ของ MCU โดยตรงทั้ง 8 เส้น

PORT-RE[0..2]

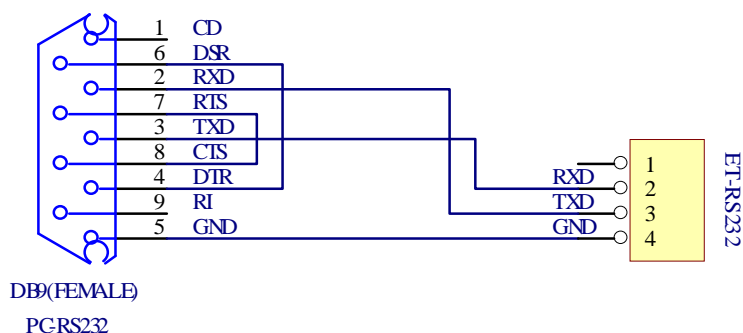
PORT-RE[0..2] เป็นสัญญาณจาก RE0...RE2 ของ MCU โดยเป็นสัญญาณที่ต่อตรงมาจาก Pin ของ MCU โดยตรงทั้ง 3 เส้น

การใช้งาน RS232

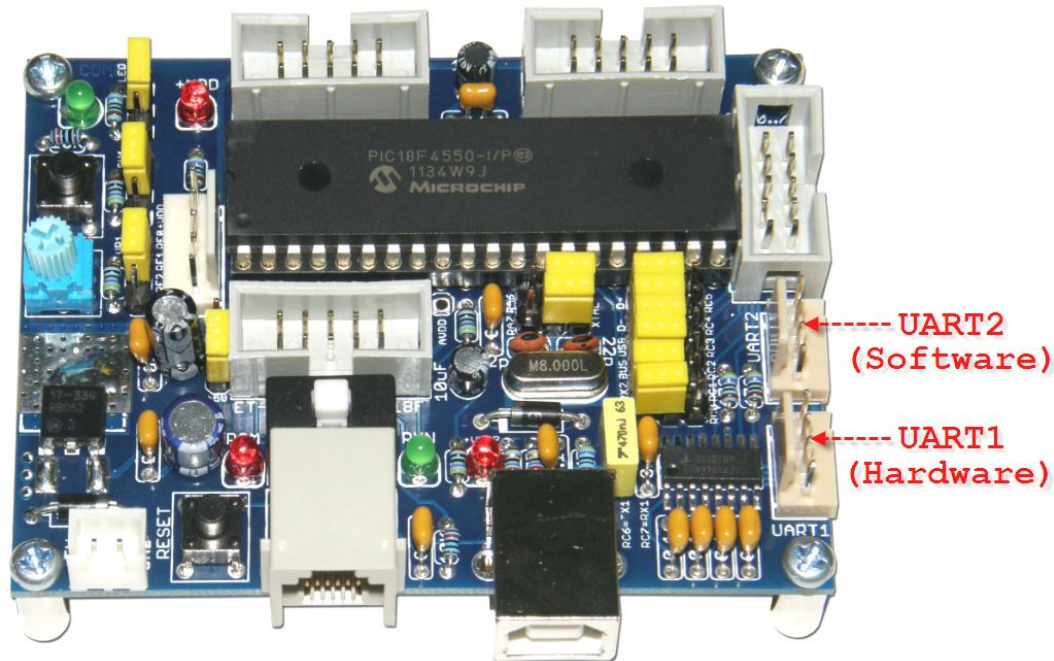
พอร์ต **RS232** เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณจาก MAX3232 เรียบร้อยแล้ว โดยในกรณีของ PIC18F4550 จะมี Hardware UART จำนวน 1 ช่อง โดยใช้ขาสัญญาณ RC6(TX1) และ RC7(RX1) ในการเชื่อมต่อ และวงจรของบอร์ดได้เพิ่มเติม UART2(Software UART) จัดเตรียมเพิ่มเติมไว้ให้ด้วย โดยในช่องที่2 จะใช้ RC0(TX2) และ RC1(RX2) พร้อม Jumper เลือกตัดต่อสัญญาณของ UART2 ว่าจะใช้หรือไม่งานได้ตามต้องการ โดยสัญญาณของ RS232 แต่ละช่อง จะจัดหัวเป็นแบบ CPA-4PIN (RS232) ดังรูป



สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับหัวต่อ RS232 ของบอร์ด ET-BASE PIC40/4550(ICSP) นั้น เป็นดังนี้



รูป แสดงวงจรสาย Cable สำหรับ RS232



หมายเหตุ สำหรับ UART2 (Software UART) นั้น ในการเขียนใช้งานต้องขึ้นอยู่กับความสามารถของ Compiler ด้วยว่าสนับสนุน Library แบบ Software UART ด้วยหรือไม่ ซึ่งกรณีใช้ตัวแปลภาษาของ PIC CCS Compiler เป็นตัวแปลภาษา จะสามารถใช้ความสามารถของ Software UART ได้โดยง่าย ในกรณีไม่ต้องการใช้งานขา RC0/RC1 เป็น UART2 ก็สามารถใช้เลือก Jumper เพื่อกำหนดหน้าที่ของขาสัญญาณทั้ง 2 ไปเป็น GPIO ได้ตามต้องการ เนื่องจากวงจร Driver ในส่วนของ UART2 ถูกออกแบบให้มีความยืดหยุ่นต่อการใช้งาน โดยมี Jumper RC0/TX2 และ RC1/RX2 ในการติดต่อสัญญาณเพื่อให้ผู้ใช้สามารถเลือกใช้หรือไม่ใช้ได้ตามต้องการ ดังตัวอย่าง

```

/*****
/* 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
            }
        }

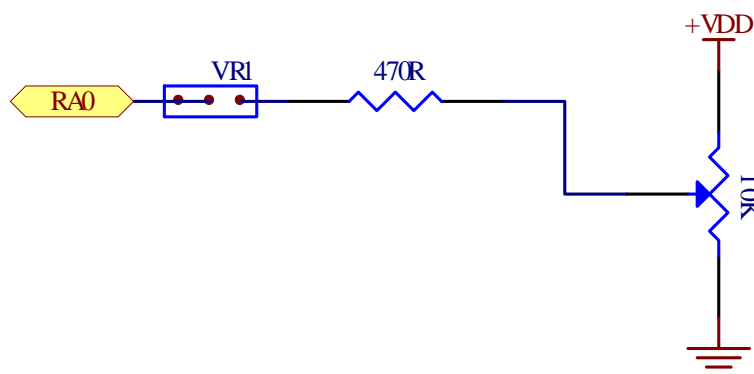
        //Verify & Echo UART2
        if(kbhit(CH2))
        {
            rx_buff = fgetc(CH2);
            if (rx_buff == 0x0D)
            {
                fprintf(CH2, "\n\r");
                fprintf(CH2, "ET-BASE PIC40/4550(ICSP)\n\r");
                fprintf(CH2, "UART2>");
            }
          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

```

แสดงตัวอย่าง Code ติดต่อ UART ภาษาซี (PICC CCS Compiler)

การใช้งาน VR ปรับแรงดัน

VR1 เป็นตัวต้านทานปรับค่าได้ใช้สำหรับปรับค่าแรงดันระหว่าง +VDD และ GND เพื่อใช้สร้างระดับแรงดัน สำหรับใช้ทดสอบการทำงานของ Input แบบ Analog (ADC) โดยแรงดันที่ได้จากการปรับ VR1 จะถูกเชื่อมต่อไปยัง ขาสัญญาณ RA0 ของ MCU โดยมี Jumper ตัดต่อสัญญาณเพื่อให้ผู้ใช้สามารถเลือกใช้หรือไม่ใช้แรงดันทดสอบที่ได้จากการปรับ VR1 นี้ ได้โดยอิสระ



เนื่องจาก Pin RA0 ของ PIC18F4550 สามารถโปรแกรมหน้าที่การทำงานได้หลายหน้าที่ ในกรณีที่ต้องการใช้เป็น ADC(AN0) ต้องทำการกำหนดค่าบิต PCFG ของรีจิสเตอร์ ANCON1 ให้มีค่าเป็น "14" ก่อนเสมอ แล้วจึงสั่ง Initial การทำงานของ ADC ในส่วนอื่นๆตามต้องการ ไม่เช่นนั้นนั้นอาจจะไม่สามารถใช้งาน RA0 เป็น ADC ได้ ดังตัวอย่าง

```
...
void main(void)
{
    unsigned char Result;

    //Initial RA0 = ADC(AN0)
    ADCON1bits.PCFG=14; // RA0 = Analog Function, Other Pin = Digital I/O
    TRISA0=1; // RA0 = Analog Input
    ADCON2bits.ADFM=0; // ADC Result = Left justified
    ADCON2bits.ACQT=7; // ADC Acquisition Time = 20TAD(0x111xxx)
    ADCON2bits.ADCS=6; // ADC Conversion Clock=Fosc/64(0xxxx110)

    ADCON1bits.VCFG=0; // ADC Vref(+)...Vref(-) = VDD...VSS(0x00xxxx)
    ADCON1bits.PCFG=14; // AN0 = Analog (0xxxx110)

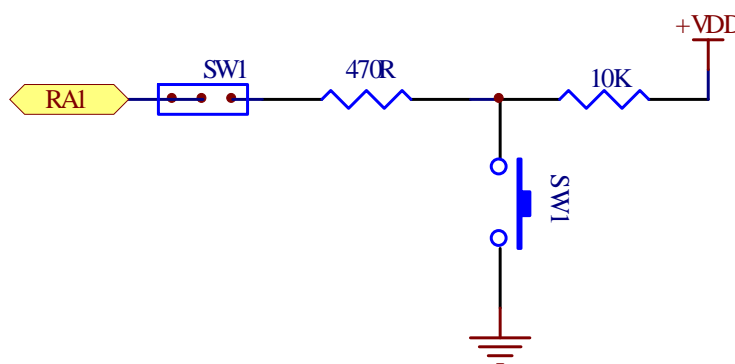
    ADCON0bits.CHS=0; // Select ADC Channel = AN0 (0x0000xx)
    ADCON0bits.ADON=1; // Start ADC

    ...
    ADCON0bits.GO_DONE = 1; // start conversion
    while (ADCON0bits.GO_DONE == 1); // wait for it to complete
    Result = ADRESH; // return high byte of result
}
```

แสดง ตัวอย่าง Code (PIC-C18 Compiler) สำหรับ Initial ADC

การใช้วงจร SW1

SW1 เป็นวงจรสวิตช์ กดติด-ปล่อยดับ ใช้สำหรับสร้างสัญญาณลอจิก "0" และ "1" เพื่อทดสอบการทำงานของ Input แบบ Logic เช่น ทดลองการตรวจจับค่าการกดสวิตช์ โดยเมื่อสวิตช์ไม่ถูกกดจะได้ค่าสถานะทางลอจิกเป็น "1" แต่ถ้าสวิตช์ถูกกดจะได้ค่าสถานะทางลอจิกเป็น "0" โดยสัญญาณลอจิกที่ได้จากวงจรนี้ จะถูกเชื่อมต่อไปยัง ขาสัญญาณ RA1 ของ MCU โดยมี Jumper ตัดต่อสัญญาณเพื่อให้ผู้ใช้สามารถเลือกใช้หรือไม่ใช้ สัญญาณลอจิกที่ได้จากการกดสวิตช์ SW1 นี้ได้โดยอิสระ



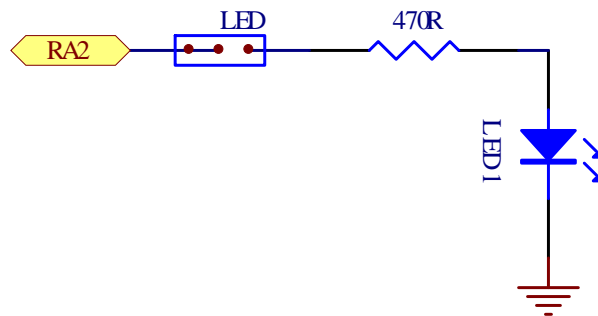
เนื่องจาก Pin RA1 ของ PIC18F4550 สามารถโปรแกรมหน้าที่การทำงานได้หลายหน้าที่ ในกรณีที่ต้องการใช้เป็น SW ต้องทำการกำหนดค่าบิต PCFG ของรีจิสเตอร์ ANCON1 ให้มีค่าเป็น "14" ก่อนเสมอ แล้วจึงสั่ง Initial การทำงานของ RA1 ในส่วนอื่นๆตามต้องการ ไม่เช่นนั้นอาจจะไม่สามารถใช้งาน RA1 เป็น Digital Input ได้ ดังตัวอย่าง

```
#define sw1          PORTAbits.RA1
...
void main(void)
{
    ...
    //Initial RA1 = Digital Input(SW1)
    ADCON1bits.PCFG=14;    // RA0 = Analog Function, Other Pin = Digital I/O
    TRISAbits.TRISA1=1;    // RA1 = Digital Input
    ...
    if(sw1 != 1)
    {
        ...                // SW1 Press(0)
    }
    else
    {
        ...                // SW1 Release(1)
    }
    ...
}
```

แสดง ตัวอย่าง Code (PIC-C18 Compiler) สำหรับ Initial SW1

การใช้งานวงจร LED

LED เป็นวงจรแสดงผลทางโลจิก โดยใช้สำหรับแสดงผลทางโลจิกเพื่อให้ผู้ใช้ได้รับรู้ โดยใช้กับสัญญาณ Output แบบ Logic โดยถ้าได้รับโลจิก "1" จะทำให้ LED ติดสว่าง และ ถ้าได้รับโลจิก "0" จะทำให้ LED ดับ โดยสัญญาณโลจิกที่จะใช้สำหรับการแสดงผลของ LED ในวงจรนี้ จะถูกเชื่อมต่อมาจากขาสัญญาณ RA2 ของ MCU โดยมี Jumper ตัดต่อสัญญาณเพื่อให้ผู้ใช้สามารถเลือกใช้หรือไม่ใช้สัญญาณโลจิกที่ได้จากขาสัญญาณ RA2 มาขับ LED นี้ได้โดยอิสระ

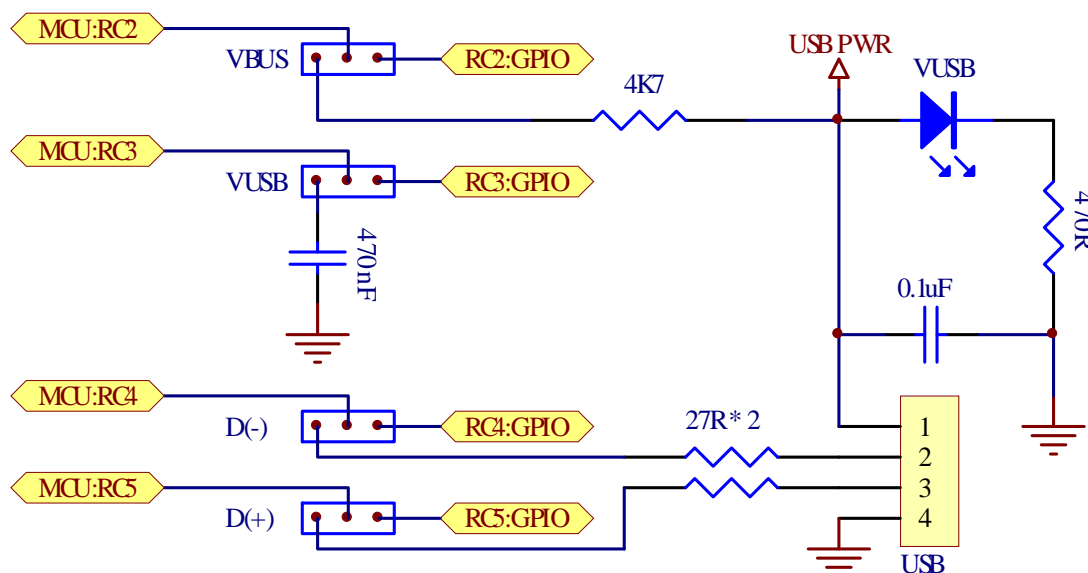


เนื่องจาก Pin RA2 ของ PIC18F4550 สามารถโปรแกรมหน้าที่การทำงานได้หลายหน้าที่ ในกรณีที่ต้องการใช้เป็น LED ต้องทำการกำหนดค่าบิต PCFG ของรีจิสเตอร์ ANCON1 ให้มีค่าเป็น "14" ก่อนเสมอ แล้วจึงสั่ง Initial การทำงานของ RA2 ในส่วนอื่นๆตามต้องการ ไม่เช่นนั้นนั้นอาจจะไม่สามารถใช้งานขา RA2 เป็น Digital Output ได้ ดังตัวอย่าง

```
...
#define LED1                LATAbits.LATA2
#define LED1_On()           LED1 = 1;
#define LED1_Off()          LED1 = 0;
#define LED1_Toggle()       LED1 = !LED1;
...
void main(void)
{
    ...
    //Initial RA2 = Output(LED)
    ADCON1bits.PCFG=14;      // RA0 = Analog Function, Other Pin = Digital I/O
    ANCON0bits.ANSEL2=0;    // RA2 = Digital Function
    TRISAbits.TRISA2=0;     // RA2 = Digital Output
    ...
    LED1_On();              // ON LED
    LED1_Off();             // OFF LED
    LED1_Toggle();          // Toggle LED
    ...
}
```

แสดง ตัวอย่าง Code (PIC-C18 Compiler) สำหรับ Initial LED

การใช้งาน USB



ในกรณีที่เลือกใช้ MCU เบอร์ที่มีระบบฮาร์ดแวร์ USB อยู่ด้วย ในการเชื่อมต่อกับ USB จะใช้ขาสัญญาณ RC3, RC4 และ RC5 เป็นสัญญาณในการเชื่อมต่อกับ USB โดยขา Pin18 (ตรงกับตำแหน่ง Pin ของ RC3 ใน MCU ปรกติ) ของ MCU จะทำหน้าที่เป็น VUSB ส่วน RC4 และ RC5 จะทำหน้าที่เป็น USB Data คือ USB D(-) และ USB D(+) ตามลำดับ

สำหรับ RC2 จะเป็นส่วน Option สำหรับใช้ตรวจจับ VBUS ของ USB ซึ่งจะใช้หรือไม่ใช้ก็ได้ แต่ถ้าต้องการใช้ ต้องกำหนดให้ Pin นี้ ทำหน้าที่เป็น Digital Input เพื่อตรวจจับโลจิกที่ได้จากแรงดัน VBUS ของ USB โดยถ้ามีการเสียบสาย USB เข้ากับขั้วต่อของ USB Host จะทำให้มีแรงดัน +5V ง่ายมาจาก USB Host ซึ่งจะทำให้ขา RC2 ได้รับโลจิก "1" แต่เมื่อมีการถอดสาย USB ออกจะทำให้ขา RC2 ได้รับโลจิก "0" ซึ่งเราจะใช้ขา RC2 สำหรับตรวจจับการว่า ขณะนั้นสายสัญญาณของ USB ยังมีการ เสียบเชื่อมต่ออยู่ หรือ ถูกลอด ออกจาก USB Host แล้ว

โดยวงจรส่วนนี้จะมี LED (+VUSB) ทำหน้าที่แสดงสถานะของแหล่งจ่ายไฟ +5V จากขั้วต่อ USB อยู่ด้วย โดยถ้า LED ติดสว่างแสดงว่ามีการเสียบสาย USB เชื่อมต่อกับ Host อยู่ แต่ถ้า LED ดับก็แสดงว่าไม่มีไฟ +5V ง่ายมาจาก USB Host นั่นเอง

การกำหนดค่า Configuration ของบอร์ด

สำหรับบอร์ด ET-BASE PIC40/4550(ICSP) นั้นจะเลือกใช้ Crystal ค่าความถี่ 8MHz เป็นค่าความถี่พื้นฐานของบอร์ด ดังนั้นในการพัฒนาโปรแกรมของบอร์ดผู้ใช้งานจะต้องกำหนดค่า Configuration ของ MCU ให้สอดคล้องกับค่าความถี่ Crystal (8.00MHz) นี้เพื่อให้ MCU ทำงานได้ถูกต้อง ซึ่ง MCU สามารถประมวลผลได้สูงสุดที่ 48MHz ดังนั้นจะต้องกำหนดค่า Phase Lock Loop(PLL) เพื่อให้คุณความถี่ Crystal 8.00MHz เพื่อให้ได้ 48MHz ซึ่งในการกำหนดค่า Configuration ของแต่ละ Compiler จะมีวิธีการไม่เหมือนกัน ดังตัวอย่าง

```
// 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)
```

ตัวอย่างการกำหนดค่า Configuration ของ PIC18F4550 ด้วย 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 ICPR1       = ON                // ***Enable ICSP/Programming***
#pragma config XINST       = OFF              // Extended Instruction Set
```

ตัวอย่างการกำหนดค่า Configuration ของ PIC18F4550 ด้วย PIC C18 Compiler

การกำหนดค่า Configuration ด้วย PIC Basic PRO Compiler

สำหรับในกรณีใช้ภาษาเบสิก(PIC Basic Pro Compiler) นั้น การกำหนดค่า Configuration เพื่อเลือกกำหนดการทำงานให้กับ MCU จะกำหนดผ่านทางไฟล์ "C:\PBP\18F4550.INC"

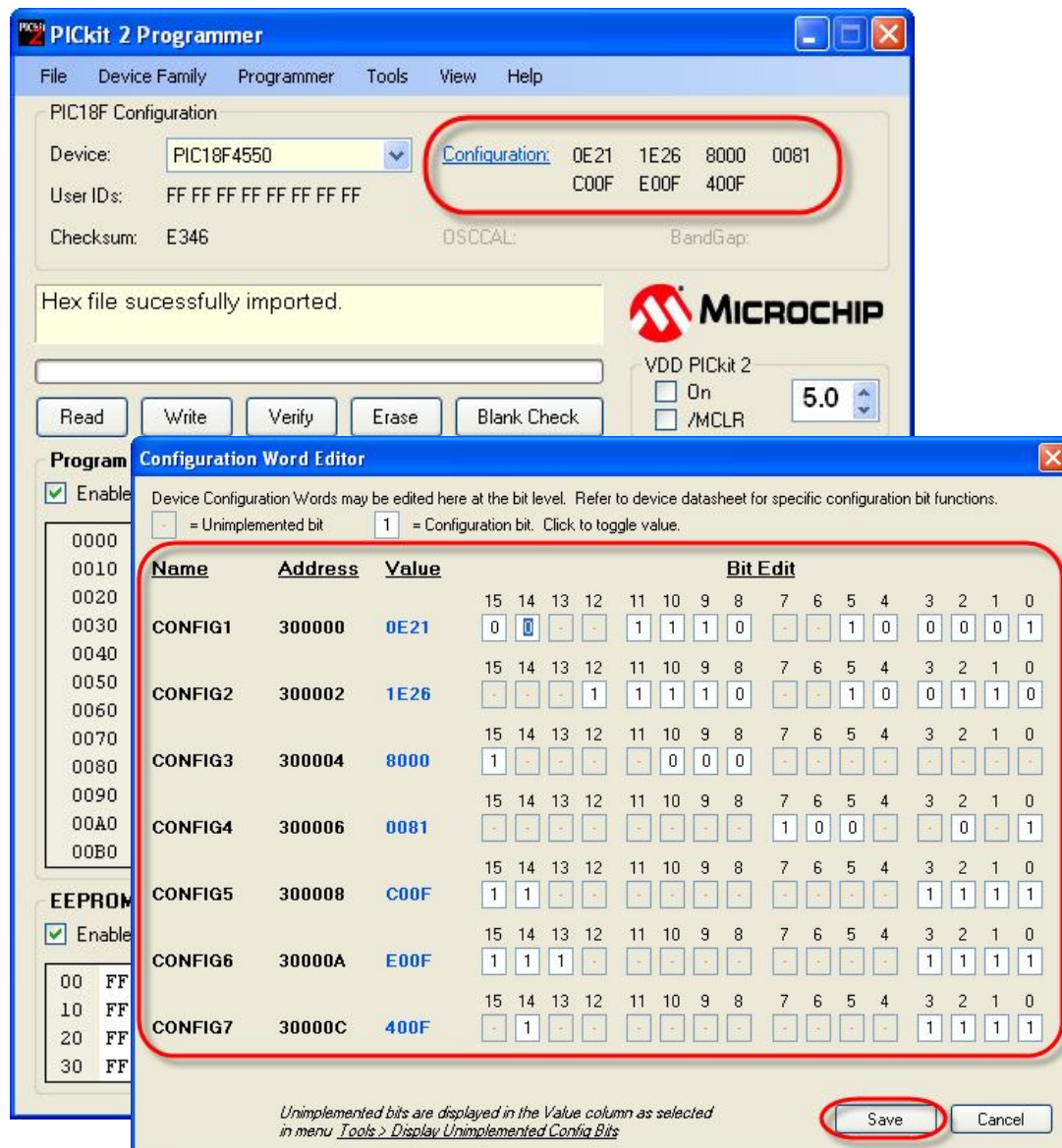
```
'ET-BASE PIC40/4550(ICSP) Hardware used X-TAL=8MHz
'Setup Clock Config -> 8MHz/2=4MHz -> 4MHzxPLL(96MHz) -> 96MHz/2=48MHz
'Config1 = 0x0E21 = 0000 1110 : 0010 0001
'Config2 = 0x1E26 = 0001 1110 : 0010 0110
'Config3 = 0x8000 = 1000 0000 : 0000 0000
'Config4 = 0x0081 = 0000 0000 : 1000 0001
'Config5 = 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 = inhx32
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&_PBDEN_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
```

ตัวอย่างการกำหนดค่า Configuration ของ PIC18F4550 ด้วย PIC Basic PRO Compiler

การกำหนดค่า Configuration ด้วยเครื่องโปรแกรม Pickit2

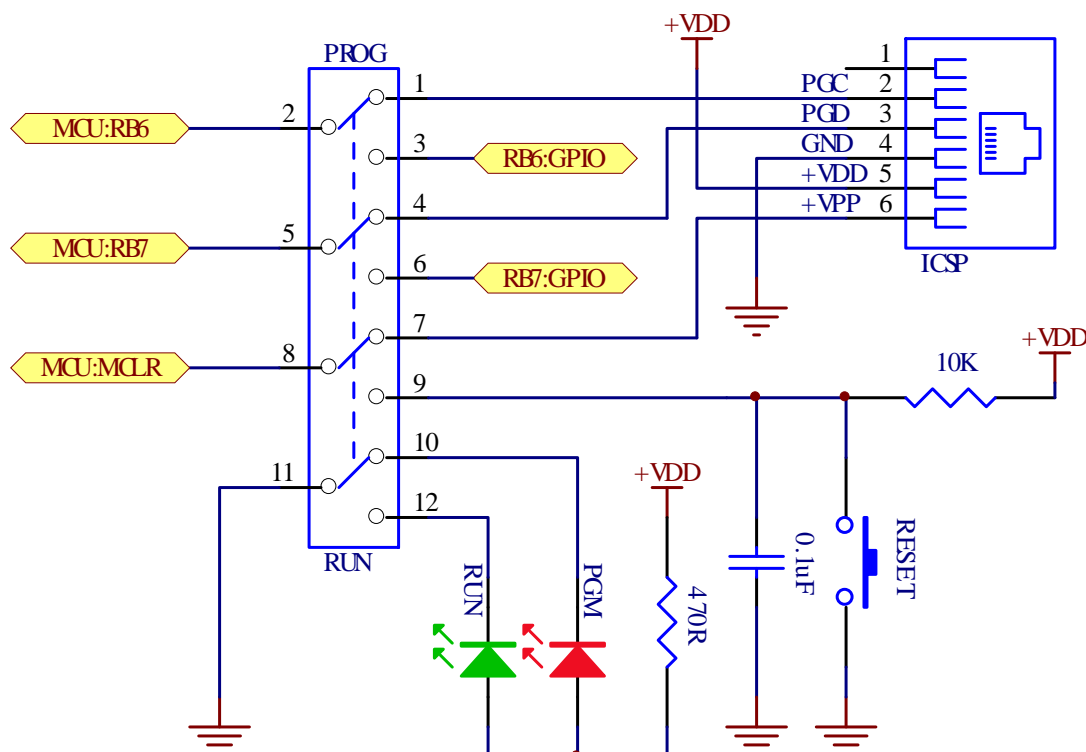
ในกรณีใช้ Pickit2 เป็นเครื่องโปรแกรม ในขั้นตอนของการโปรแกรมถ้าสามารถตรวจสอบค่า Configuration ได้จากโปรแกรม ซึ่งถ้ากำหนดค่า Configuration ถูกแปลและบรรจุไว้ใน Hex File แล้ว เมื่อโหลด Hex เข้ามาในโปรแกรม ค่า Configuration ที่แสดงบนหน้าต่างโปรแกรมควรได้ค่าตามที่กำหนดไว้ ซึ่งถ้าไม่ตรงหรือลืมกำหนดค่า Configuration ไว้ใน Source Code สามารถเข้าไปแก้ไขค่า Configuration ได้เองที่หน้าต่างโปรแกรมของ Pickit2 ดังตัวอย่าง



รูปแสดง ลักษณะของค่า Configuration ของ PIC18F4550 ของโปรแกรม Pickit2

การใช้งาน ICSP

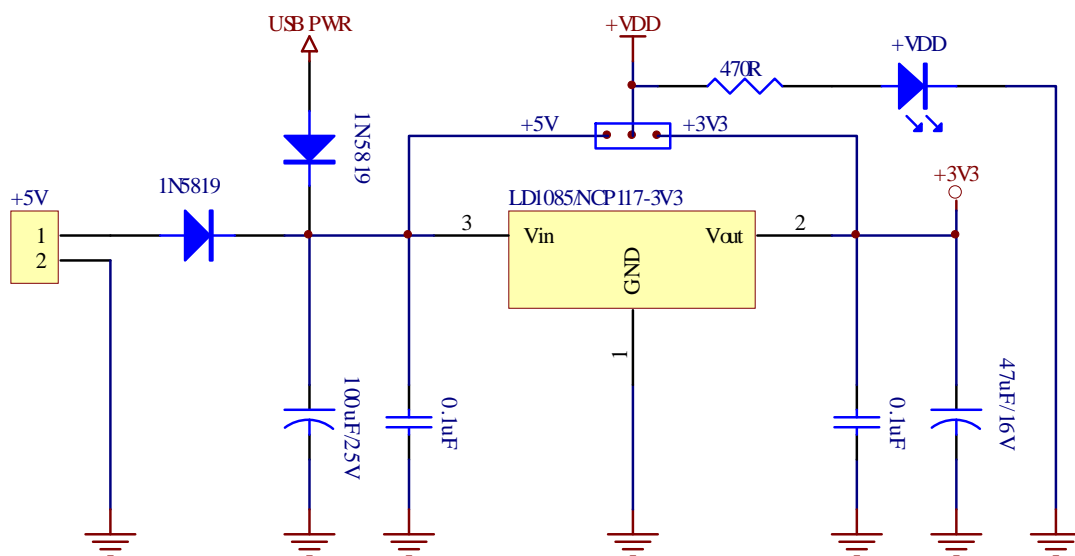
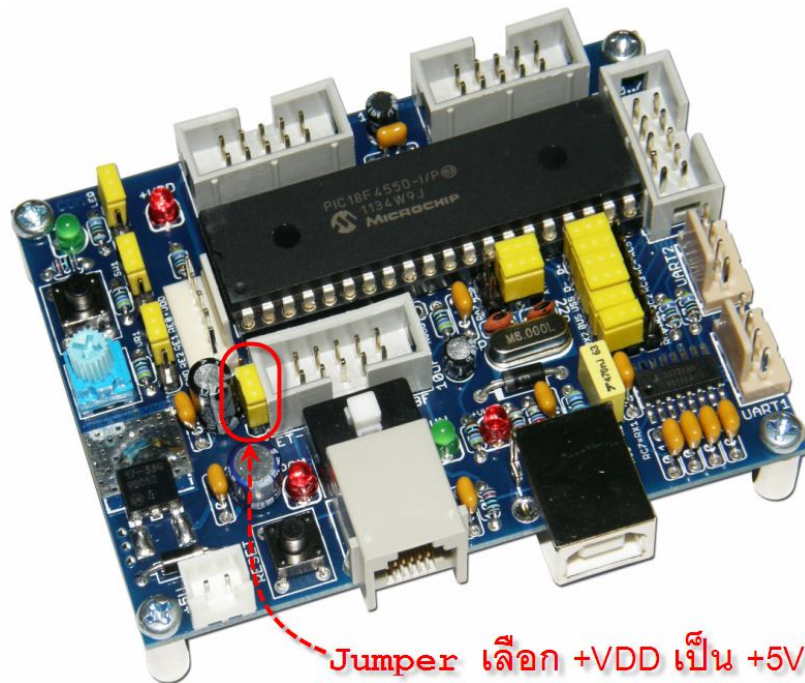
ICSP จะเป็น Connector แบบ RJ11 สำหรับ Interface กับเครื่องมือพัฒนาโปรแกรมตระกูล PIC ที่มีการจัดขั้วตามมาตรฐาน ICSP ของ Microchips เช่น ICD2, ICD3, Pickit2 หรือ Pickit3 ซึ่งสามารถใช้งานร่วมกับเครื่องมือพัฒนาของ Microchips หรือ เทียบเท่า เช่น ET-PGMPIC USB(เทียบเท่า Pickit2) หรือ ET-PGMPIC-PK3(เทียบเท่า Pickit3) หรือ ET-ICDX(เทียบเท่า ICD2) ส่วนการจะเลือกใช้เครื่องมือรุ่นใดนั้นให้พิจารณาจากเบอร์ MCU ที่จะใช้งานเป็นหลักว่า เบอร์ MCU ที่จะใช้งานนั้น มีเครื่องโปรแกรมรุ่นใดรองรับบ้าง เพราะความสามารถของเครื่องโปรแกรมที่ยกตัวอย่างให้ทราบข้างต้นนั้น รองรับการใช้งานกับ MCU ได้จำนวนเบอร์มากน้อยไม่เท่ากัน เพียงแต่มีการจัดเรียงขาสัญญาณในการโปรแกรม MCU เป็นมาตรฐานแบบเดียวกันเท่านั้น โดยวงจรของบอร์ดในส่วนนี้จะมีสวิตช์สำหรับเลือกติดต่อสัญญาณของ RB6, RB7 และ MCLR สำหรับใช้ทำหน้าที่เชื่อมต่อกับ Programmer/Debugger หรือ ใช้งานตามปกติได้พร้อม LED แสดงสถานะ ว่าการทำงานของสวิตช์อยู่ในตำแหน่งใด โดยถ้าเลือกสวิตช์ไว้ทางด้าน Programmer/Debugger จะเห็น LED สีแดงของ PGM ติดสว่าง แต่ถ้าตำแหน่งของสวิตช์อยู่ด้าน Run จะเห็น LED สีเขียว(RUN) ติดสว่าง โดยมีการจัดเรียงขาสัญญาณตามมาตรฐานของ ICSP ดังนี้

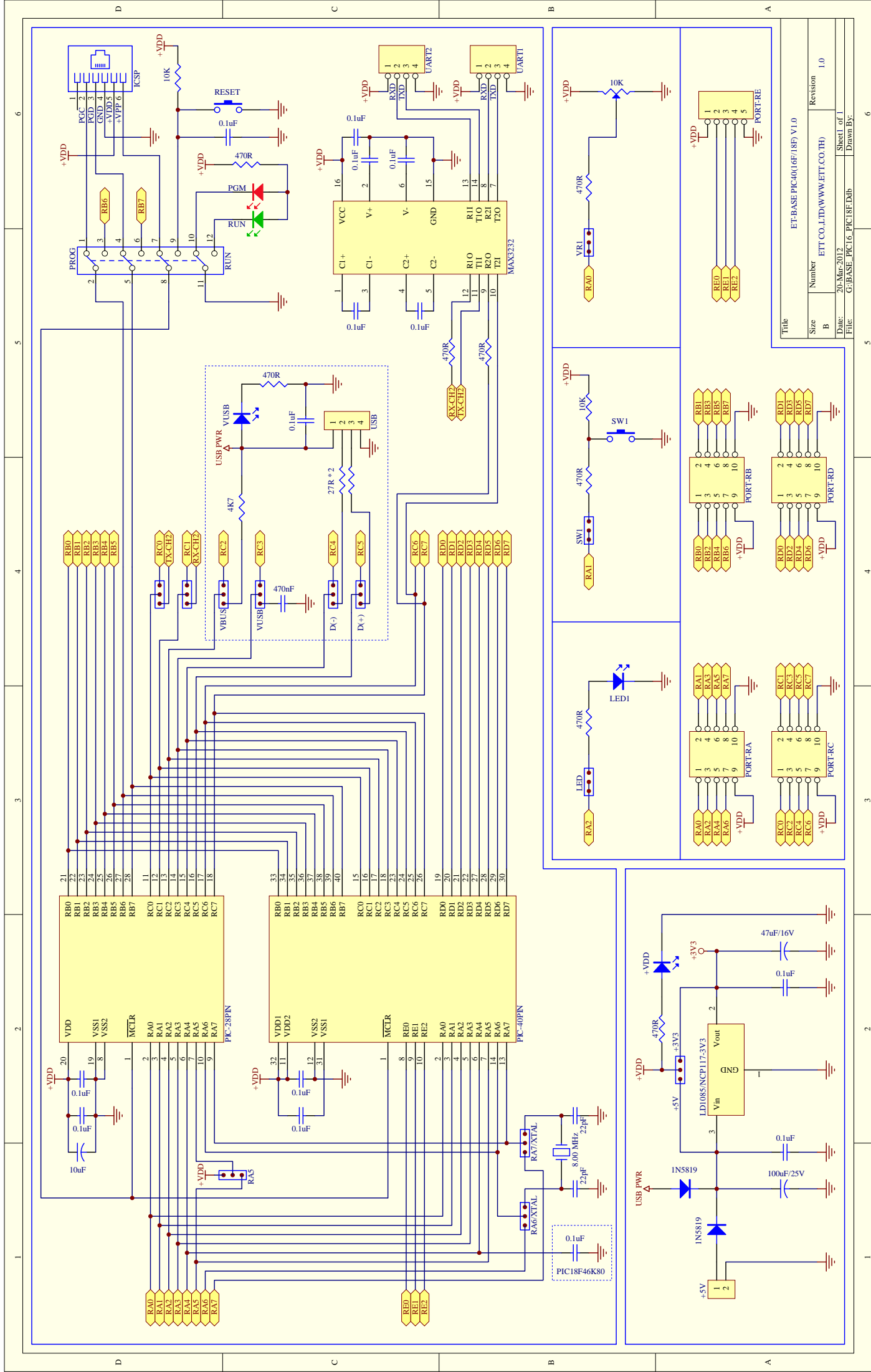


รูปแสดง โครงสร้างวงจรส่วนการเชื่อมต่อ ICSP ของบอร์ด ET-BASE PIC40/4550(ICSP)

วงจรภาคจ่ายไฟ

สำหรับวงจรภาคจ่ายไฟของบอร์ด จะใช้ได้กับแรงดันภายนอกขนาด +5VDC ซึ่งวงจรของบอร์ดสามารถรับแหล่งจ่ายแรงดัน +5VDC ได้จาก 2 แหล่ง คือ จากขั้วต่อ Connector 2 Pin และจากขั้วต่อสัญญาณของ USB (ในกรณีใช้กับ MCU รุ่นมี USB) โดยภายในบอร์ดจะมีวงจร Regulate ขนาด 3.3V/1A พร้อม Jumper สำหรับเลือกใช้แรงดัน +5VDC หรือ +3V3 ให้เป็นแรงดันไฟเลี้ยงของ MCU (+VDD) และวงจร I/O ภายในบอร์ด ซึ่งในกรณีของ ET-BASE40/4550(ICSP) จะเลือกกำหนดเป็น 5V เสมอ





Title				ET-BASE PIC400(6F/18F) V1.0			
Size				Number			
B				ETT CO.,LTD./WWW.ETT.CO.TH			
Date:				20-Mar-2012			
File:				G:\BASE_PIC16_PIC18F.Ddb			
Revision				1.0			