|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | http://sky.geocities.jp/home_iwamoto/img/base/mark01.gif**Peripheral Pin Select (PPS)機能** | [**Next**](http://sky.geocities.jp/home_iwamoto/page/P26J50/P26_A17.htm) |   　18F26J50の特記するべき特徴の一つはPeripheral Pin Select (PPS)機能を持っていることです。PPSにより内蔵するモジュールの入出力ピンをある程度自由に再配置することができます。これは、ピン数の少ないPICでは大きな福音となります。  http://sky.geocities.jp/home_iwamoto/img/base/mark01.gif**入力ピンの配置指定** 　以下のペリフェラル入力ピンを任意の１６本の“RPn”ピンのどれかに指定することができます。左辺に**RPINRx**で、どの周辺デバイスかを表記し、右辺に指定する**“RPn”ピン番号**を指定します。たとえば　EUSART2 Rxピンを　RP1(RA1)ピンに指定するには、   |  |  |  |  | | --- | --- | --- | --- | |  | **RPINR16** | **=** | **1;** | |  | ↑RX2を示す |  | ↑RP1ピンを示す |   とプログラムします。さらに   |  |  | | --- | --- | |  | **ANCONｘ**のビットを「１」にセット | |  | **TRISｘ**のビットも「１」にセット |   該当する“RPn”ピンをデジタル入力にします。 |
| |  |  |  | | --- | --- | --- | | 周辺デバイス名称 | 記号 | 表記 | | External Interrupt 1 | INT1 | RPINR1 | | External Interrupt 2 | INT2 | RPINR2 | | External Interrupt 3 | INT3 | RPINR3 | | Timer0 Ext Clock Input | T0CKI | RPINR4 | | Timer3 Ext Clock Inpuｔ | T3CKI | RPINR6 | | Input Capture 1 | CCP1 | RPINR7 | | Input Capture 2 | CCP2 | RPINR8 | | Timer1 Gate Input | T1G | RPINR12 | | Timer3 Gate Input | T3G | RPINR13 | | EUSART2 Asynchronous Rx | RX2/DT2 | RPINR16 | | EUSART2 Synchronous Clock Input | CK2 | RPINR17 | | SPI2 Data Input | SDI2 | RPINR21 | | SPI2 Clock Input | SCK2IN | RPINR22 | | SPI2 Slave Select Input | SS2IN | RPINR23 | | PWM Fault Input | FLT0 | RPINR24 | |
| http://sky.geocities.jp/home_iwamoto/img/base/mark01.gif**出力ピンの配置指定** 　以下のペリフェラル出力ピンを任意の１６本の“RPn”ピンのどれかに指定することができます。左辺に**RPORx**で“RPn”ピン番号を示し、右辺に周辺デバイスを**表記番号**で指定します。  たとえば　EUSART2 Txピンを　RP0(RA0)ピンに指定するには、   |  |  |  |  | | --- | --- | --- | --- | |  | **RPOR0** | **=** | **5;** | |  | ↑RP0ピンを示す |  | ↑TX2を示す |   とプログラムします。さらに   |  |  | | --- | --- | |  | **ANCOｘ**のビットを「１」にセット | |  | **TRISｘ**のビットは「０」にリセット |   該当する“RPn”ピンをデジタル出力にします。 |
| |  |  |  | | --- | --- | --- | | 周辺デバイス名称 | 記号 | 表記番号 | | NULL | NULL | 0 | | Comparator 1 Output | C1OUT | 1 | | Comparator 2 Output | C2OUT | 2 | | EUSART2 Asynchronous Tx /Asynchronous Clock Output | TX2/CK2 | 5 | | EUSART2 Synchronous Transmit | DT2 | 6 | | SPI2 Data Output | SDO2 | 9 | | SPI2 Data Output | SCK2 | 10 | | SPI DMA Slave Select | SSDMA | 12 | | Ultra Low-Power Wake-up Event | ULPOUT | 13 | | ECCP1 Compare or PWM Output Channel A | CCP1/P1A | 14 | | ECCP1 Enhanced PWM Output, Channel B | P1B | 15 | | ECCP1 Enhanced PWM Output, Channel C | P1C | 16 | | ECCP1 Enhanced PWM Output, Channel D | P1D | 17 | | ECCP2 Compare or PWM Output Channel A | CCP2/P2A | 18 | | ECCP2 Enhanced PWM Output, Channel B | P2B | 19 | | ECCP2 Enhanced PWM Output, Channel C | P2C | 20 | | ECCP2 Enhanced PWM Output, Channel D | P2D | 21 | |
| http://sky.geocities.jp/home_iwamoto/img/base/mark01.gif**PPS機能の例として、** 　　Peripheral Pin Select (PPS)機能の例として、USART2のプログラムです。 UART2 TX Outputは、 RP0(RA0)に、 UART2 RX Inputを RP1(RA1)に設定しています。 |
| ＜プログラム＞ 　　C18 プロジェクトファイル：　[J05c\_UART2\_RxLED.zip](http://sky.geocities.jp/home_iwamoto/pgm/P26J50/J05c_UART2_RxLED.zip) |
| |  | | --- | | //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // PIC18F26J50 UART2 受信 Program// : MPLAB C18  // 内部クロック4Mhz Baud Rate:9600BPS, 16-bit Baud Rate  // C18ライブラリを使用している。  // 内部クロック4Mhz Baud Rate:9600BPS, 16-bit Baud Rate  // シリアルデータの受信で以下の動作をする  // 受信データ、  // 0 ： 全LED OFF  // 1 ： LED0 ON  // 2 ： LED1 ON  // 3 ： LED2 ON  // 4 ： LED3 ON  // ポートの設定  // UART Input : RP1(RA1)  // UART Output : RP0(RA0)  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  #include <p18cxxx.h>  #include <usart.h>  #include <timers.h>  #define LED0 LATBbits.LATB0  #define LED1 LATBbits.LATB1  #define LED2 LATBbits.LATB2  #define LED3 LATBbits.LATB3  //-------------- コンフィグレーション ----------------------  //  #pragma config WDTEN = OFF, PLLDIV = 2, STVREN = ON, XINST = OFF  #pragma config CPUDIV = OSC1, CP0 = OFF  #pragma config OSC = INTOSC, T1DIG = OFF  #pragma config LPT1OSC = OFF, FCMEN = OFF, IESO = OFF  #pragma config WDTPS = 32768  #pragma config DSWDTOSC = INTOSCREF, RTCOSC = T1OSCREF  #pragma config DSBOREN = OFF, DSWDTEN = OFF, DSWDTPS = 8192  #pragma config IOL1WAY = OFF, MSSP7B\_EN = MSK7  #pragma config WPFP = PAGE\_1, WPEND = PAGE\_0, WPCFG = OFF  #pragma config WPDIS = OFF  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  void main(void){  char RxData;  OSCCON = 0b01100011; // 内部クロック4Mhz  LATB = 0; // PortBのすべてのビットを「０」  TRISA = 0b11111110; // TX2のポートを出力に設定  TRISB = 0b11110000; // LEDのポートを出力に設定  ANCON0 = 0b00011111; // すべてのポートをデジタルに設定  ANCON1 = 0b00011111; //  // PPSの設定  RPOR0 = 5; //RP0 = TX2 (RA0)  RPINR16 = 1; //RX2 = RP1 (RA1)  Open2USART(  USART\_TX\_INT\_OFF &  USART\_RX\_INT\_OFF &  USART\_ASYNCH\_MODE &  USART\_EIGHT\_BIT &  USART\_CONT\_RX &  USART\_BRGH\_HIGH,  103); //9600 bps  baud2USART (  BAUD\_IDLE\_RX\_PIN\_STATE\_HIGH &  BAUD\_IDLE\_TX\_PIN\_STATE\_HIGH &  BAUD\_16\_BIT\_RATE &  BAUD\_WAKEUP\_OFF &  BAUD\_AUTO\_OFF);  while(1){  while (!DataRdy2USART()); // 受信するまで待つ  RxData = Read2USART(); // 受信データを取り込む  switch(RxData){ // データに対応したLEDを点灯  case '0':  LED0 = 0; LED1 = 0; LED2 = 0; LED3 = 0; break;  case '1':  LED0 = 1; LED1 = 0; LED2 = 0; LED3 = 0; break;  case '2':  LED0 = 0; LED1 = 1; LED2 = 0; LED3 = 0; break;  case '3':  LED0 = 0; LED1 = 0; LED2 = 1; LED3 = 0; break;  case '4':  LED0 = 0; LED1 = 0; LED2 = 0; LED3 = 1; break;  }  }  } | |
|  |