

```

//*****
// PIC16F1829 UART SUPPORT
// Program Name : uart.c
// Programmer  : nobcha
// Program date : Version 1.0 2014.11.1
// History    :
// Language   : HI_TECH C Version 9.83
//*****
//
#include <htc.h>
#define PI_NORMAL 0
#define PI_FAIL 1

//*****
// Note for Initializing
//*****
// Set OSC (16MHz, IntOSC)
// OSCCON = 0b11111010;
//   RX ==> RB5
//   TX ==> RB7
//
// Initializing EUART
//   init_UART();
//*****
// Function : init_UART
// Title    : initializing EUSTAT
// Input    : None
// Output   : None
// Note     : Fosc:16MHz, Baud Rate:9600BPS
//*****
// TXSTA Register Setting
// b7 CSRC: 0 Don't care
// b6 TX9: 0 Not 9bits
// b5 TXEN: 1 Transmit ENable
// b4 SYNC: 0 EUSART Asynchronous mode
// b3 SENDB:0 No
// b2 BRGH: 1 High Baud Rate Select
// b1 TRMT: 0 TSR full
// b0 TX9D: No Ninth bit
// *****
// RCSTA Register Setting
// b7 SPEN: 1 Serial Port ENable
// b6 RX9: 0 9-bit Receive Disable
// b5 SREN: 0 Don't care
// b4 CREN: 1 Continuous Receive ENable
// b3 ADDEN:0 Don't care
// b2 FERR: 0 Framing Error bit (May be updated by reading RCREG register)
// b1 OERR: 0 Overrun Error bit
// b0 RX9D: Ninth bit of Received Data (Don't care)
// *****
// BAUDCON Register Setting
// b7 ABDOVF:0 Do'nt Auto-Baud Detect Overflow bit
// b6 RCIDL: 0 If 1, Receive is Idle
// b5 Unimplemented: 0
// b4 SCKP: 0 Transmit not inverted data
// b3 BRG16: 1 16-bit Baud Rate Generator bit used (SPBRGH:SPBRG)
// b2Unimplemented: 0
// b1 WUE: 0 No Wake-up Enable bit
// b0 ABDEN: Auto-Baud Detect Disable

```

```

// *****
// Setting BRG Register
// BRG = 1600000(Hz)/4/9600(BPS)-1 ==> 416d ==> 0x1A0
// *****
void init_UART(void){
    TXSTA = 0b00100100;
    RCSTA = 0b10010000;
    BAUDCON = 0b00001000;

    SPBRGH = 0x01;
    SPBRG = 0xA0;
}

//*****
// Program      : get_UART
// Title       : Recive from EUSTAT
// Input      : RX data pointer
// Output     : error flag
//*****
char get_UART(unsigned char *rx_data){

// Waiting received by checking Flag
// Before checking Clear RCIF
    PIR1 = PIR1 & 0b1111011;

// Check RCIF
    while((PIR1&0x02) == 0x00);

// Check error
    if(RCSTA & 0x06 != 0x00){ //0000,0110 RCSTA(FERR,OERR)
        *rx_data = 0x00;
        return(PI_FAIL);
    }

// Return Recived Data
    *rx_data = RCREG;
    return(PI_NORMAL);
}

//*****
// Program      : put_UART
// Title       : Sending data via EUSTAT
// Input      : Transmitting data
// Output     : Error flag
//*****
char put_UART(unsigned char tx_data){
// Checking Send Flag
    while((TXSTA & 0x02) == 0x00); //0b00000010 TXSTA(TRMT)

// Write data
    TXREG = tx_data;
    return(PI_NORMAL);
}

//*****
// Function : puts_UART
// Title   : Send Data String via UART
// Input  : data pointer until 0x00
// Output : None

```

```
/**
 * *****
 */
void puts_UART(char *string){
    while(*string!=0x00){
        put_UART(*string);
        string++;
    }
}
```