RS232Cについてのまとめ

PIC16F877AとXBEE用USBインターフェイス基板と通信を行った．

苦労したのは文字が飛んできても文字化けしていたことと基板（AE-XBEE-USB）のTXとRXが反対だったことである。

文字化けしていた理由はGNDが共通化していなかった（基板とPICのGNDがつながっていなかった）ことが原因である．

このプログラムではLEDを１秒おきにチカチカさせながらPICから「Yohoo」と文字が飛んでくる使用です。

プログラムのコードは次に示す

ここから始まる

//ヘッダファイルの読み込み

#include <16f877a.h> //16F873のヘッダファイル

#include <stdio.h> //標準入出力のヘッダファイル

// 入出力ピンの設定

#define TRIS\_A 0x00 //ALL,OUT

#define TRIS\_B 0x00 //ALL,OUT

#define TRIS\_C 0x80 //PIN\_C7:IN

#define TRIS\_D 0x00 //ALL,OUT

#define TRIS\_E 0x00 //ALL,OUT

//ピンのdefine

#define RUN\_LED PIN\_C3

//RS232Cの設定コマンド

#define RS\_BAUD 9600 //Baud-Reatは9600bps

#define RS\_TX PIN\_C6 //TXピンはPIN\_C6

#define RS\_RX PIN\_C7 //RXピンはPIN\_C7

//コンフィギュレーションビットの設定

#fuses HS,NOWDT,NOPROTECT

//詳細は一番下のメモ書きにて

//クロック速度の指定（20MHz)

#use delay(clock = 20000000)

//RS232Cの設定

#use rs232(BAUD = RS\_BAUD , XMIT = RS\_TX, RCV = RS\_RX)

//初期設定

#use fast\_io(a)

#use fast\_io(b)

#use fast\_io(c)

#use fast\_io(d)

#use fast\_io(e)

#byte port\_a = 5

#byte port\_b = 6

#byte port\_c = 7

#byte port\_d = 8

#byte port\_e = 9

//ここからmain関数

main()

{

long int i = 0;

char j = 0;

//初期化

set\_tris\_a(TRIS\_A);

set\_tris\_b(TRIS\_B);

set\_tris\_c(TRIS\_C);

set\_tris\_d(TRIS\_D);

set\_tris\_e(TRIS\_E);

port\_a = 0;

port\_b = 0;

//port\_c = 0;

port\_d = 0;

port\_e = 0;

printf("start");

output\_low(RUN\_LED); //動作確認用LEDを光らす

delay\_ms(50);

while(1)

{

delay\_ms(1000);

output\_high(PIN\_B5);

printf("yahoo");

delay\_ms(1000);

output\_low(PIN\_B5);

//j = getc();

printf("yahoo");

//printf("Hell, sofmap would!!No.%ld\n\r",i);

//i++;

}

}

/\*ｃ⌒っﾟдﾟ)っφ ﾒﾓﾒﾓ...

コンフィグレーションビットについて

HS(High Speed)

NOWDT(no-Watch Dog Time)

NOPROTECT

NOLVP(no Low-Voltage-Programming)

PUT(Power-Up-Timer)

BROWNOUT\*/

ここで終わり