#include "ASA\_Lib.h" //測試程式 成功 (Bluetooth to UART1)(UART0 to PC)

#include "ASA\_Lib\_DAC00.h"

#include <math.h>

#include <string.h>

#define FOSC 11059200// Clock Speed

#define BAUD0 9600 //

#define BAUD1 115200//UL 115200

#define MYUBRR0 (FOSC/16/BAUD0-1)

#define MYUBRR1 (FOSC/16/BAUD1-1)

volatile uint8\_t i,get[100];

int k,position;

void USART\_Flush( void )

{

unsigned char dummy;

while ( UCSR1A & (1<<RXC1) ) dummy = UDR1;

}

void USART0\_Init( unsigned int ubrr )

{

/\* Set baud rate \*/

UCSR0B|=(1<<RXCIE0)|(1<<TXCIE0); //致能TX，RX complete interrupt

UBRR0H |= (unsigned char)(ubrr>>8); //p.362 // fosc = 11.0592MHz，Baud Rate=9600，U2X=0 =>UBRR=71，U2X=1=>UBRR=143

UBRR0L |= (unsigned char)ubrr;

/\* Enable receiver and transmitter \*/

UCSR0B |= (1<<RXEN0)|(1<<TXEN0); //enables the USARTn Receiver，enables the USARTn Transmitter

/\* Set frame format: 8data, 2stop bit \*/

//UCSR1C = (0<<USBS1)|(3<<UCSZ10); //selects the number of stop bits，USBS1=1=> 2 bits

UCSR0C |= (1<<UPM01)|(0<<UPM00)|(1<<USBS0)|(1<<UCSZ01)|(1<<UCSZ00)|(0<<UCPOL0);//Character Size=8 bits，UCPOL1=上升/下降

}

void USART1\_Init( unsigned int ubrr )

{

/\* Set baud rate \*/

UCSR1B|=(1<<RXCIE1)|(1<<TXCIE1); //致能TX，RX complete interrupt

UBRR1H |= (unsigned char)(ubrr>>8); //p.362 // fosc = 11.0592MHz，Baud Rate=9600，U2X=0 =>UBRR=71，U2X=1=>UBRR=143

UBRR1L |= (unsigned char)ubrr;

/\* Enable receiver and transmitter \*/

UCSR1B |= (1<<RXEN1)|(1<<TXEN1); //enables the USARTn Receiver，enables the USARTn Transmitter

/\* Set frame format: 8data, 2stop bit \*/

//UCSR1C = (0<<USBS1)|(3<<UCSZ10); //selects the number of stop bits，USBS1=1=> 2 bits

UCSR1C |= (1<<UPM11)|(0<<UPM10)|(1<<USBS1)|(1<<UCSZ11)|(1<<UCSZ10)|(0<<UCPOL1);//Character Size=8 bits，UCPOL1=上升/下降

}

void USART1\_Transmit( unsigned char data )

{

/\* Wait for empty transmit buffer \*/

/\* Put data into buffer, sends the data \*/

PORTB|=(1<<PB4);

PORTB&= ~(1<<PB3);

UDR1 = data;

while ( !( UCSR1A & (1<<UDRE1)) ) //If UDREn is one, the buffer is empty

;

}

// uint8\_t uart1\_get() {

// while(!(UCSR1A&(1<<RXC1)));

// // {

// // //printf("stuck\n" );

// // }

//

// a++;

// return UDR1;

// }

int main(void)

{

//ASA\_M128\_set();

//printf("start1111\n");

DDRB |= (1<<DDB7)|(1<<DDB6)|(1<<DDB5); //洞洞板通道開啟

PORTB |= (1<<PB6);//洞洞板通道開啟(洞洞板轉到2)

DDRB=0xff;

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=0;

k=0;

USART1\_Init ( MYUBRR1 );

USART0\_Init ( MYUBRR0 );

sei(); // 開啟所有中斷功能

// while (1) {

// USART1\_Transmit(9);

// }

//USART1\_Transmit(9);

while(1)

{

//printf("1 i=%d\n",i );

//USART1\_Transmit(10);

// while (i<4) {

// ;

// }

if(get[2]==130)

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

//printf("wrong1\n" );

}

if(i==2 || i==4)

{

if(get[0]==50 && get[1]==230 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=1;

}

else if(get[0]==51 && get[1]==204 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

// while (1) {

// k=7400;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

// }

position=2;

}

else if(get[0]==50 && get[1]==204 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=3;

}

else if(get[0]==51 && get[1]==230 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=4;

}

else if(get[0]==49 && get[1]==230 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=5;

}

else if(get[0]==48 && get[1]==204 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=6;

}

else if(get[0]==49 && get[1]==204 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=7;

}

else if(get[0]==48 && get[1]==230 )

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

position=8;

}

else {

//printf("wrong2\n" );

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

}

}

if(i>4)

{

//printf("wrong i=%d\n",i );

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

//printf("wrong3\n" );

}

if(get[3]==254)

{

i=0;

for(int j=0;j<100;j++)

{

get[j]=0;

}

}

while (position>0) {

switch (position) {

case 1:

{

while (1) {

k=7500;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7500;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("UP!!!\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 2:

{

while (1) {

k=7400;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7400;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("RIGHT!!!\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 3:

{

while (1) {

k=7300;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7300;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("DOWN!!!\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 4:

{

while (1) {

k=7200;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7200;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("LEFT!!!\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 5:

{

while (1) {

k=7100;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7100;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("UP~~~\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 6:

{

while (1) {

k=7000;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=7000;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("RIGHT~~~\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 7:

{

while (1) {

k=6900;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=6900;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("DOWN~~~\n");

//\_delay\_ms(500);

//position=0;

}

break;

case 8:

{

while (1) {

k=6800;

USART1\_Transmit(0b10000000);

USART1\_Transmit(k>>7);

USART1\_Transmit(k&127);

}

// k=6800;

// USART1\_Transmit(0b10000000);

// USART1\_Transmit(k>>7);

// USART1\_Transmit(k&127);

//printf("LEFT~~~\n");

//\_delay\_ms(500);

//position=0;

}

break;

}

}

//\_delay\_ms(500);

}

return 0;

}

ISR(USART0\_RX\_vect) {

get[i]=UDR0;

i++;

}