/\*

\* PSerial.c

\*

\* Created: 2/9/2020 2:11:27 PM

\* Author: Nathan Potvin

\* Author: Matthew Glancy

\*/

#include <avr/io.h>

#include "PSerial.h"

/\*\*

\* Based on the given port, choose which to set into PORTn

\*

\* @param port is a given port

\* @param PORTn is what should be set to

\*\*/

void get\_port(*uint8\_t* port, UART\_PORT \*\*PORTn)

{

switch (port)

{

case 0:

\*PORTn = UART\_0;

break;

case 1:

\*PORTn = UART\_1;

break;

case 2:

\*PORTn = UART\_2;

break;

case 3:

\*PORTn = UART\_3;

break;

}

}

/\*\*

\* Initializes a port and sets it up to be read from and write to

\*

\* @param port is the port to open

\* @param speed is the baud rate

\* @param framing is the

\*\*/

void PSerial\_open(*uint8\_t* port, long speed, int framing)

{

volatile UART\_PORT \*PORTn;

get\_port(port, &PORTn);

// Has predetermined UBRR's to pick from

// (makes it faster since it doesn't need to use the calculation)

switch (speed)

{

case 2400:

PORTn->UBRRn = 832;

break;

case 4800:

PORTn->UBRRn = 416;

break;

case 9600:

PORTn->UBRRn = 207;

break;

case 14400:

PORTn->UBRRn = 138;

break;

case 19200:

PORTn->UBRRn = 103;

break;

case 28800:

PORTn->UBRRn = 68;

break;

case 38400:

PORTn->UBRRn = 51;

break;

case 57600:

PORTn->UBRRn = 33;

break;

case 76800:

PORTn->UBRRn = 25;

break;

case 115200:

PORTn->UBRRn = 16;

break;

case 230400:

PORTn->UBRRn = 8;

break;

case 250000:

PORTn->UBRRn = 7;

break;

case 500000:

PORTn->UBRRn = 3;

break;

case 1000000:

PORTn->UBRRn = 1;

break;

default:

// Function used to determine the UBRR

PORTn->UBRRn = (*F\_CPU* / (8 \* speed)) - 1;

}

PORTn->UCSRnA |= (1<<U2X0);

// Set the UCSRnB register for Rx complete interrupt (RXCIE),

// Tx complete interrupt (TXCIE), Data register empty interrupt (UDRIE),

// Receiver enable (RXEN), and Transmitter enable (TXEN)

PORTn->UCSRnB |= (0b0<<RXCIE0) | (0b0<<TXCIE0) | (0b0<<UDRIE0)

| (0b1<<RXEN0) | (0b1<<TXEN0);

// Set the UCSRnC register for USART mode select (UMSELn0),

// Parity mode (UPMn0), Stop bit select (USBSn), Character size (UCSZn0)

PORTn->UCSRnC |= (0b00<<UMSEL00) | (((framing>>PARITYBITS)&0x3)<<UPM00)

| (((framing>>STOPBITS)&0x1)<<USBS0)

| (((framing>>DATABITS)&0x7)<<UCSZ00);

}

/\*\*

\* If there is currently a byte to read from UDR, it will return that

\*

\* @param port is to be read from

\*\*/

int PSerial\_read(*uint8\_t* port)

{

volatile UART\_PORT \*PORTn;

get\_port(port, &PORTn);

if (!(PORTn->UCSRnA & (1 << RXC0)))

{

return -1;

}

return (unsigned int) PORTn->UDRn;

}

/\*\*

\* Will wait to read a byte when one becomes available

\*

\* @param port is to be read from

\*\*/

char PSerial\_readw(*uint8\_t* port)

{

volatile UART\_PORT \*PORTn;

get\_port(port, &PORTn);

while (!(PORTn->UCSRnA & (1 << RXC0)));

return PORTn->UDRn;

}

/\*\*

\* If it's currently allowed to, it will write a byte to UDR

\*

\* @param port is written to

\* @param data is what to write

\*\*/

int PSerial\_write(*uint8\_t* port, *uint8\_t* data)

{

volatile UART\_PORT \*PORTn;

get\_port(port, &PORTn);

if (!(PORTn->UCSRnA & (1 << UDRE0)))

{

return -1;

}

else

{

PORTn->UDRn = data;

return 0;

}

}

/\*\*

\* Will wait to write a byte for when it's allowed to

\*

\* @param port is written to

\* @param data is what to write

\*\*/

void PSerial\_writew(*uint8\_t* port, *uint8\_t* data)

{

volatile UART\_PORT \*PORTn;

get\_port(port, &PORTn);

while (!(PORTn->UCSRnA & (1 << UDRE0)));

PORTn->UDRn = data;

}