```
/*_____________________________________________________________________
|
| Date:                  2019-10-23
|_____________________________________________________________________
|
| Author:
|   Student name 1:      Einar van de Velde
|   Student name 2:      Abdulla Mehdi
|_____________________________________________________________________
|
| Lab number:            6.
| Title:         Task 2.
|_____________________________________________________________________
|
| Other information:
| Changes in program:
|       File Created (2019-10-23)
|        Program is runnable (2019-10-23)
|_____________________________________________________________________*/
#include<avr/io.h>
#define F_CPU 1000000UL
#include<util/delay.h>

#include<stdio.h>
#include<string.h>

#define CLOCK 1000000
#define SPEED 2400
#define VALUE ((CLOCK)/16/SPEED - 1)

void Transmit(unsigned char data);
int CheckSum(char arr[]);
void Display(char arr[]);
char Receive();
void Change(char arr1[], char arr2[], char displayStr[], bool
doTransmit);
void Initialize(unsigned int ubrr);
void ChooseArray(int num);
void AddCharacter(char data);
void DisplayCharacter(char *arr);
void JudgeLine();
void Start();
void FillLine();
void ChooseLine(char num);

char _saveArr1[25];
char _saveArr2[25];
char _saveArr3[25];
char *_curArr;
char _curArrNum;
char *_curLineStr;
int _counter = 0;

char lineStr1[] = "\rAO0001";
```

```
char lineStr2[] = "\rBO0001";
char displayStr[] = "\rZD0013C";

int main() {
	UBRR0L = 24;
	UCSR0B = 0x08;
	Initialize(VALUE);

	DDRB = 0xff;
	PORTB = 0x00;

	Start();
	char data;
	while(1) {
		data = Receive();
		if(data == 0x2F) {
			data = Receive();
			ChooseLine(data);
			data = Receive();
		}

		if(data == 0x0d) {
			FillLine();
			JudgeLine();
			data = Receive();
		}

		AddCharacter(data);
		DisplayCharacter(_curArr);
	}

	return 0;
}

void ChooseLine(char num) {
	switch(num) {
		case 0x31:
			_curArrNum = 1;
			break;
		case 0x32:
			FillLine();
			_curArrNum = 2;
			break;
		case 0x33:
			_curArrNum = 3;
			break;
	}

	ChooseArray(_curArrNum);
}

void Start() {
	_curArrNum = 1;
	ChooseArray(_curArrNum);
```

```
}

void ChooseArray(int num) {
        if(num == 1) {
                memset(_saveArr1,0,25*sizeof(char)        );
                _curArr = &_saveArr1;
                _curLineStr = &lineStr1;
                _counter = 0;

        } else if(num == 2) {
                memset(_saveArr2,0,25*sizeof(char));
                _curArr = &_saveArr2;
                _curLineStr = &lineStr1;
                _counter = 0;

        } else if(num == 3) {
                memset(_saveArr3,0,25*sizeof(char));
                _curArr = &_saveArr3;
                _curLineStr = &lineStr2;
                _counter = 0;
        }
}

void FillLine() {
        int length = 24 - _counter;
        int i = 0;
        while(i < length) {
                _curArr[i + _counter] = 0x20;
                i++;
        }
}

void AddCharacter(char data) {
        if(_counter >= 24) {
                JudgeLine();
                ChooseArray(_curArrNum);
        }

        _curArr[_counter] = data;
        _counter++;
        if(_curArrNum == 1 && _counter < 24) { FillLine(); }
}

void JudgeLine() {
        if(_curArrNum == 1) { _curArrNum = 2; }
        else if(_curArrNum == 2) { _curArrNum = 3; }
        else if(_curArrNum == 3) { _curArrNum = 1; }

        ChooseArray(_curArrNum);
}


void DisplayCharacter(char *arr) {
        char result[200];
```

```
        memset(result,0,200*sizeof(char));

        if(_curArrNum == 1) {
                strcat(result, _curLineStr);
                strcat(result, arr);
                strcat(result, _saveArr2);

        } else if(_curArrNum == 2) {
                strcat(result, _curLineStr);
                strcat(result, _saveArr1);
                strcat(result, arr);

        } else if(_curArrNum == 3) {
                strcat(result, _curLineStr);
                strcat(result, arr);
        }

        Change(result, result, displayStr, false);
}

void Change(char arr1[], char arr2[], char displayStr[], bool
doTransmit) {
        sprintf(arr2, "%s%2X", arr1, CheckSum(arr1));
        Display(arr2);
        if(doTransmit) { Transmit(0x0A); }
        Display(displayStr);
}

void Initialize(unsigned int ubrr) {
        UBRR0H = (unsigned char)(ubrr >> 8);
        UBRR0L = (unsigned char)ubrr;
        UCSR0B = (1 << RXEN0) | (1 << TXEN0);
        UCSR0C = (1 << UCSZ00) | (1 <<  UCSZ01);
}

void Transmit(unsigned char data) {
        while ( !( UCSR0A & (1<<UDRE0)) );
        UDR0 = data;
}

void Display(char arr[]) {
        int i = 0;
        int length = strlen(arr);
        while(i < length) {
                char data = arr[i];
                Transmit(data);
                i++;
        }

        Transmit(0x0A);
}

char Receive() {
        /* Wait for data to be received */
```

```
        while ( !(UCSR0A & (1<<RXC0)) );
        /* Get and return received data from buffer */
        return UDR0;
}

int CheckSum(char arr[]) {
        int csum = 0;
        int length = strlen(arr);
        int i = 0;
        while(i < length) {
                csum += arr[i];
                i++;
        }

        csum= csum%256;
        Transmit('csum');
        return csum; // modulo 256
}
```