Computer Technology I

Lab. 6: CyberTech Wall Display

Author: LOIC GALLAND,
LEONARDO PEDRO
Supervisor:
Semester: Autumn 2019
Area: Computer Science
Course code: 1DT301

Contents

1 Task 1 - Write a program that writes a character on the CyberTech Display 1

2 Task 2 - Write a program that writes characters on all text-lines on the CyberTech Display 4

3 Task 3 -Write a program that change text strings on the display 7

4 Task 4 - Write a program that communicates with both the terminal and the display 10

5 Task 5 - Write a program for text input. 17

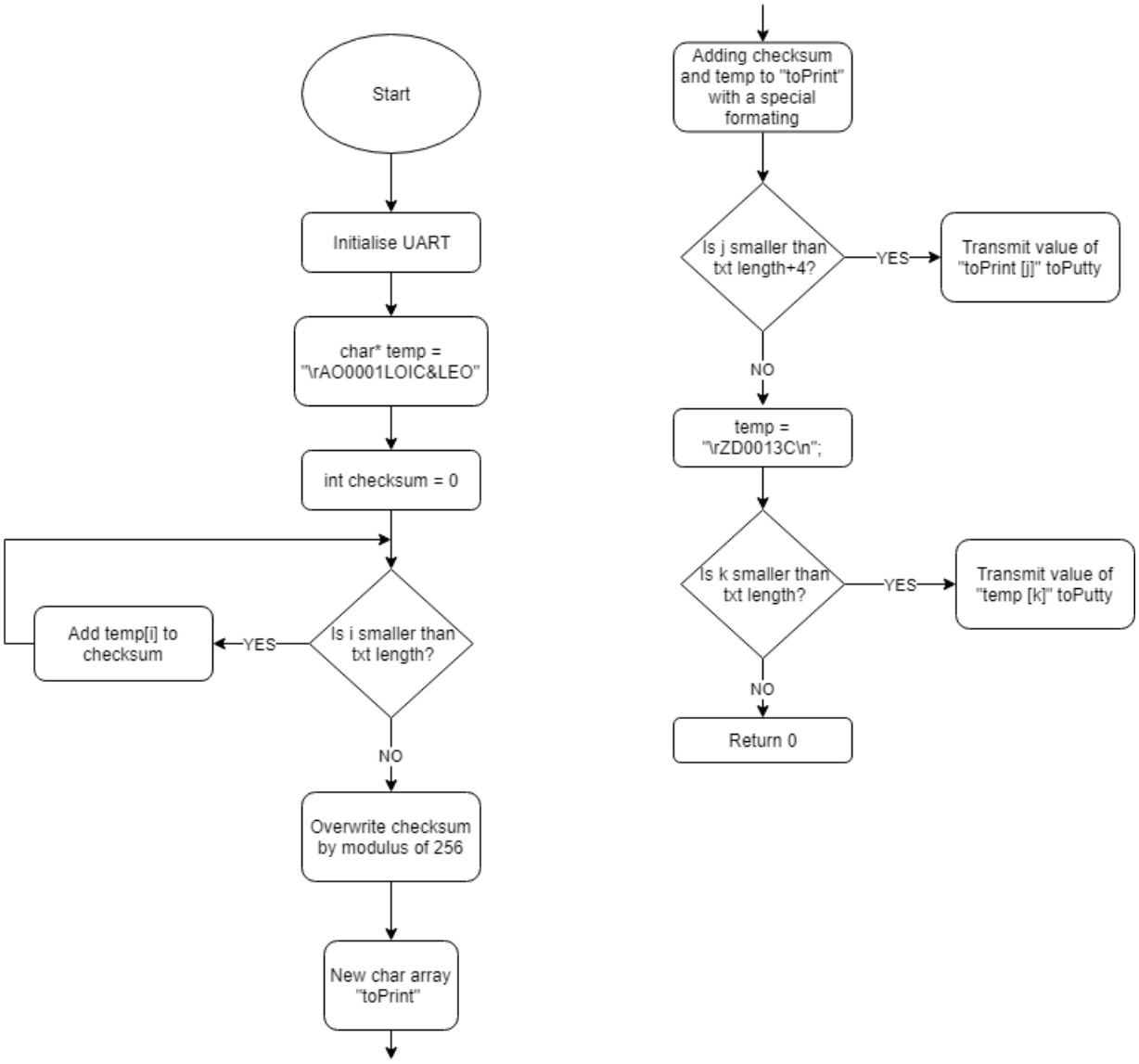
1 Task 1 - Write a program that writes a character on the CyberTech Display

Any character can be displayed. The display is connected to the serial port (RS232) on the STK600. Communication speed is 2400 bps.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2019-10-20
; Author:
; Loic GALLAND
; Leonardo PEDRO
;
; Lab number: 6
; Title: CyberTech Wall Display
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Write a program that writes a character on the CyberTech
    Display
;
; Input ports: Port0 (RS232) VGA
;
; Output ports: Port0 (RS232) VGA
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#include <avr/io.h>
#include <stdio.h>
#include <string.h>
#define F_CPU 1843200 //Clock Speed
#define BAUD 2400
#define MYUBRR (F_CPU/16/BAUD-1)

void USART_Unit(unsigned int ubrr);
void toPutty(unsigned char data);

int main(void)
{
        USART_Unit(MYUBRR);
        char* temp = "\rAO0001LOIC&LEO"; //Begiining code to send to
           the display + actual text.
        int length = 15;
        int checksum = 0;

        ;Use checksum to make sure that what was sent and was received
           is the same.
        for (int i=0;i<strlen(temp);i++ )
        {
                checksum+=temp[i];
        }

        checksum%=256;

        char toPrint[strlen(temp)+3];
```

```
        sprintf(toPrint,"%s%02X\n",temp,checksum);

        for(int j=0;j<length+4;j++){
                toPutty(toPrint[j]);
        }

        temp = "\rZD0013C\n";//End code to be sent to the display to
           say that what needed to be sent has been sent.

        for (int k=0;k<strlen(temp);k++)
        {
                toPutty(temp[k]);
        }

}
;TO INITIALIZE THE USART CONNECTION
void USART_Unit(unsigned int ubrr){
        UBRR1L = ubrr;

        ;/*     Enable receiver and transmitter */
        UCSR1B = (1<<RXEN1)|(1<<TXEN1);
}

;To send to the the display the character
void toPutty(unsigned char data){
        ; Wait for data to be received
        while ( !(UCSR1A & (1 << UDRE1))); //Receive Complete
        ;RXCn) flag //Return received data from buffer
        UDR1 = data;
}
```

This is the flowchart of the task 1:



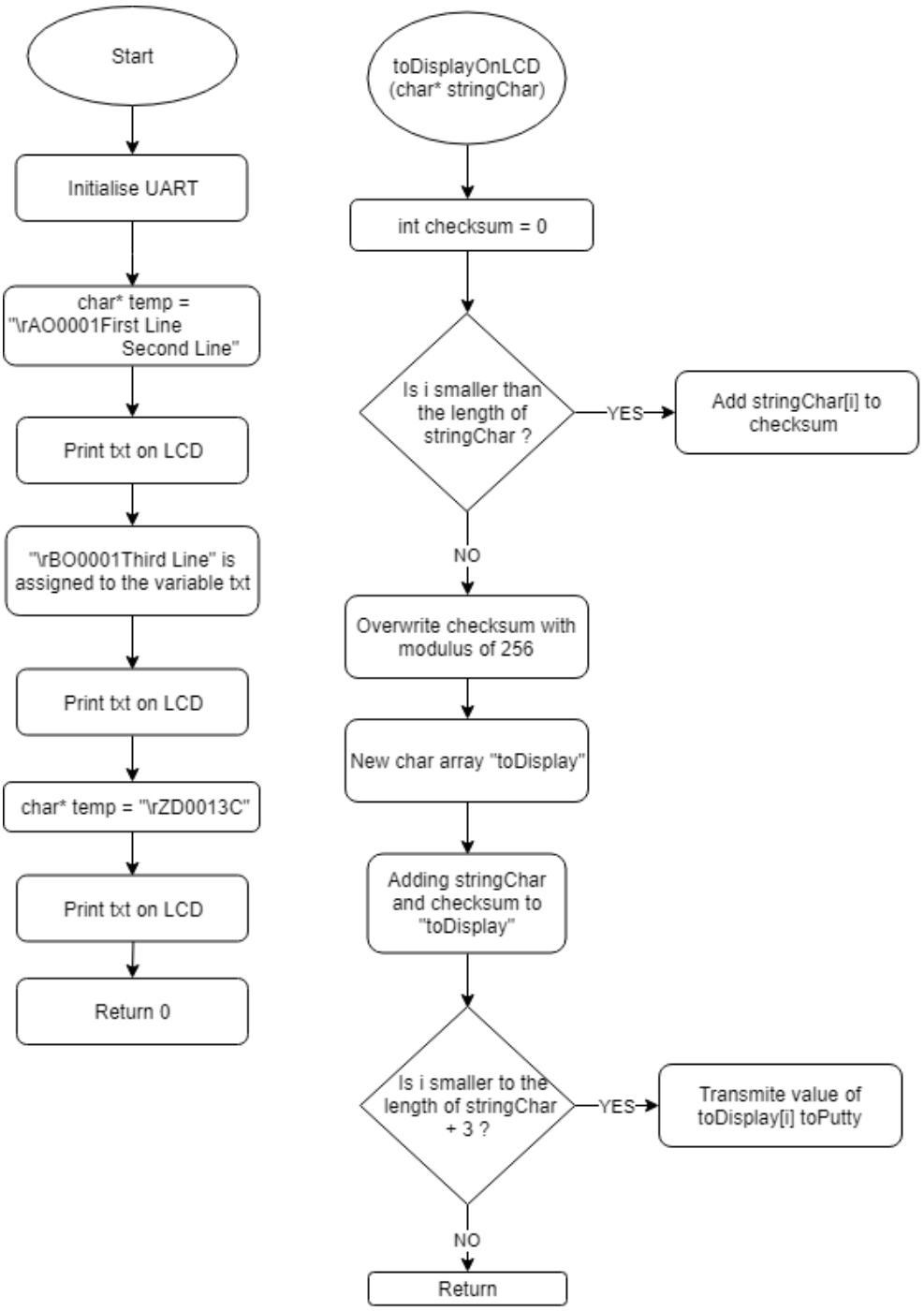
2 Task 2 - Write a program that writes characters on all text-lines on the CyberTech Display

The program will write to all three rows.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2019-10-20
; Author:
; Loic GALLAND
; Leonardo PEDRO
;
; Lab number: 6
; Title: CyberTech Wall Display
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Write a program that writes characters on all text-lines on
    the CyberTech
;                        Display.
;
; Input ports: Port0 (RS232) VGA
;
; Output ports: Port0 (RS232) VGA
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
#include <avr/io.h>
#include <stdio.h>
#include <string.h>
#define FCPU 1000000// Clock Speed
#define BAUD 2400 //Communication Speed Display rate 2400
#define MYUBBRR (FCPU/16/BAUD-1) ;UBBRR = 25 -> osc = 1MHz

void uart_int(void);
void toPutty(unsigned char data);
void toDisplayOnLCD(char* stringChar);

int main(void)
{
        uart_int();

        char* txt = "\rA00001First Line              Second Line";

        toDisplayOnLCD(txt);

        txt = "\rB00001Third Line"; //Begining combination for the
            third line
        toDisplayOnLCD(txt);

        txt = "\rZD0013C\n"; //Ending combination to tell the display
            everything been sent.
        toDisplayOnLCD(txt);

        return 0;
}
```

```
void toDisplayOnLCD(char* stringChar){

        int checksum = 0;

        for(int i =0; i<strlen(stringChar);i++){
                checksum += stringChar[i];
        }

        checksum%=256;

        char toDisplay [strlen(stringChar)+3];
        sprintf(toDisplay, "%s%02X\n", stringChar, checksum); //%02x
           means print at least 2 digits, prepends it with 0's if
           there's less.
        ;%02x is used to convert one character to a hexadecimal string

        for (int i = 0; i<strlen(stringChar)+3;i++){
                toPutty(toDisplay[i]);
        }
}

void toPutty(unsigned char data){

        while(!(UCSR1A & (1<<UDRE1)));
        UDR1 = data;
}

void uart_int(void) {
        UBRR1L = MYUBBRR; //25 because we are setting the board at 1MHz

        UCSR1B = (1<<RXEN1|1<<TXEN1); // Enable receive and transmit
           bit
}
```

This is the flowcharts of the task 2:

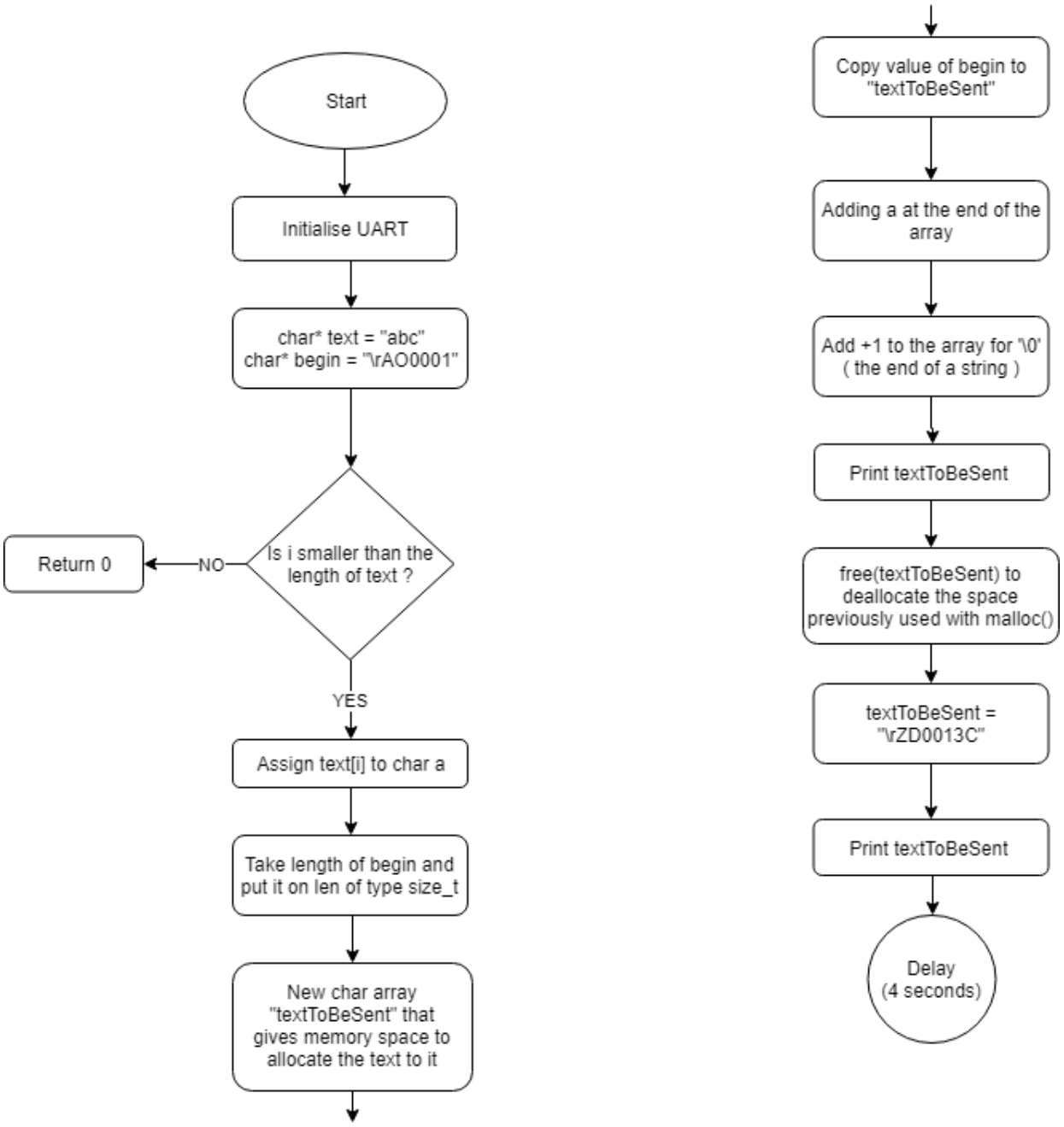


3 Task 3 -Write a program that change text strings on the display

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2019-10-20
; Author:
; Loic GALLAND
; Leonardo PEDRO
;
; Lab number: 6
; Title: CyberTech Wall Display
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Write a program that change text strings on the display.
;
; Input ports: Port0 (RS232) VGA
;
; Output ports: Port0 (RS232) VGA
;
; Subroutines: If applicable.
; Included files: m2560def.inc
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#include <avr/io.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define ClockSpeed 1000000 ;Clock Speed
#include <util/delay.h>
#define BAUD 2400 ;Communication Speed Display rate 2400
#define MYUBBRR (ClockSpeed/16/BAUD-1) ;UBBRR = 25 -> osc = 1MHz

void uart_int(void);
void toPutty(unsigned char);
void toDisplayOnLCD(char*);


void  toDisplayOnLCD(char* character){  ;Showing data into the big
    display

        int checksum = 0;

        for(int i =0;i<strlen(character);i++){
                checksum += character[i];
        }

        checksum%=256;

        char sendingToDisplay [strlen(character)+3];
        sprintf(sendingToDisplay, "%s%02X\n", character, checksum);
           //%02x means print at least 2 digits, prepends it with 0's
           if there's less.
        ;%02x is used to convert one character to a hexadecimal string

        for (int i = 0; i<strlen(character)+3;i++){
                toPutty(sendingToDisplay[i]);
        }
```

```
}

void toPutty(unsigned char data){
        while(!(UCSR1A & (1<<UDRE1)));
        UDR1 = data;
}

void uart_int(void) {
        UBRR1L = MYUBBRR; //25 --> board at 1MHz

        UCSR1B = (1<<RXEN1|1<<TXEN1); // Receive Enable (RXEN) bit //
            Transmit Enable (TXEN) bit
}

int main(void)
{
        uart_int();


        char* text = "abc";
        char* begin = "\rAO0001";

        for(int i =0;i<strlen(text);i++){       //Go through every
            character and add it to the string
             char a = text[i];
             size_t length = strlen(begin);

             char* textToBeSent = malloc(length + 1 + 1); //Giving
                memory space to allocate the data to textToBeSent
             strcpy(textToBeSent, begin); // copy txt to
                textToBeSent

             textToBeSent[length] = a;
             textToBeSent[length + 1] = '\0'; // adding the end char
                 \0
              toDisplayOnLCD(textToBeSent);
             free(textToBeSent); // deallocate the memory space used
                 by malloc()

             ;Ending combination to tell the Display that everything
                 was sent.
             textToBeSent = "\rZD0013C";
              toDisplayOnLCD(textToBeSent);

             _delay_ms(4000); //wait 4s between each letter so that
                we actually have time to see the change.
        }


        return 0;
}
```

This is the flowchart of the task 3:



4 Task 4 - Write a program that communicates with both the terminal and the display

Since we only have one serial port, we must make a special cable, so that the STK600 receive unit is connected to the terminal (PuTTY, for instance) and transmit is connected to the display. Text can be entered at the terminal. End of line with a special character that you choose. It should also be possible to enter address on the screen to display text.

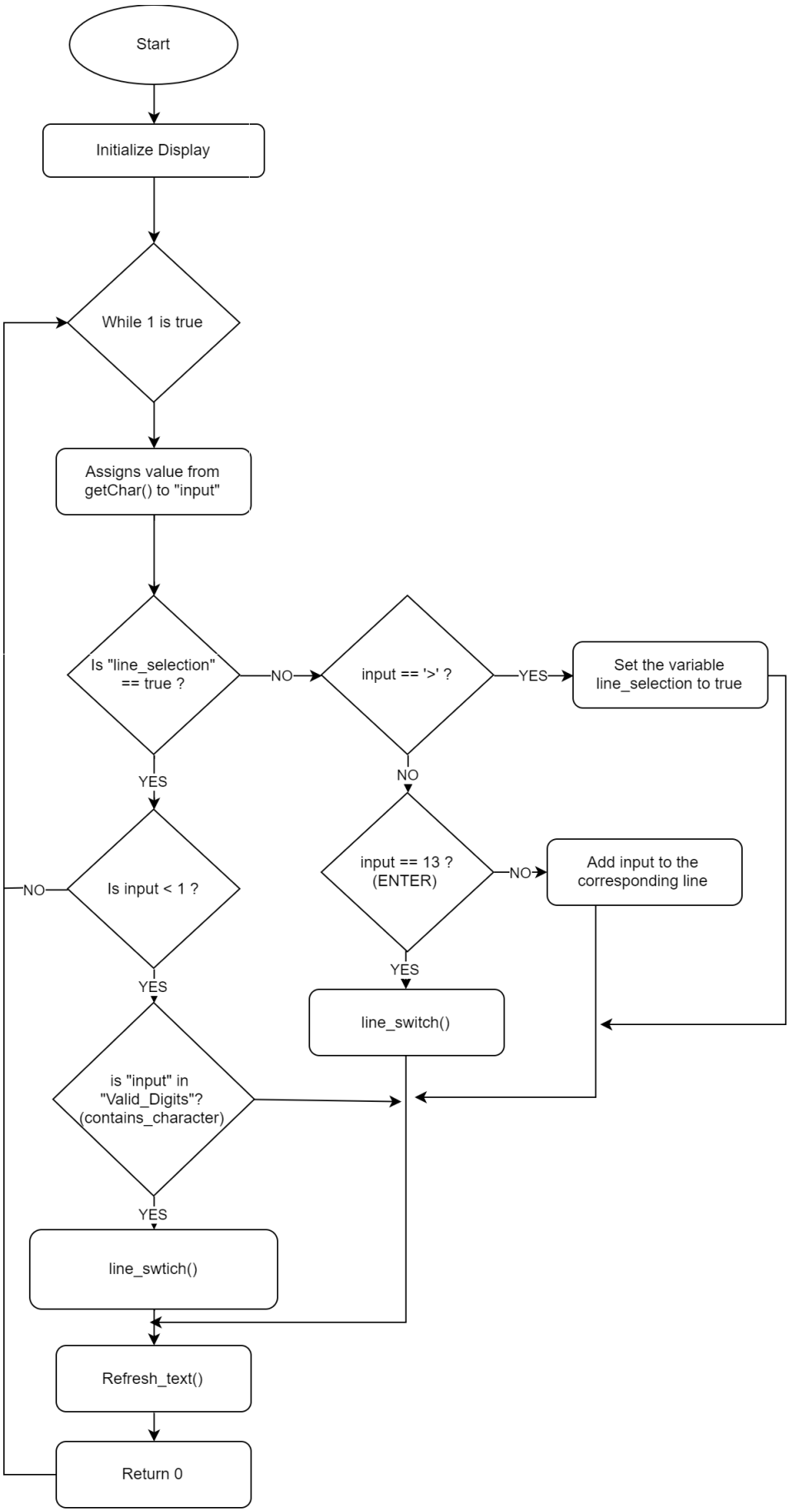
```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <avr/io.h>

#define Card_CPU 1000000 ;Clock Speed of the CPU

#define BAUDRATE 2400 ;Display rate of 2400
#define MY_UBBRR (Card_CPU/16/BAUDRATE-1) ;Baud Rate = 25 -> osc = 1MHz
    -> Display rate speed = 2400
#define Valid_Digits "123"
#define SELECT_LINE '>' ;Char for line selection
#define TOTAL_AMOUNT_CHARS      24
#define TOTAL_AMOUNT_LINES 3 ;change this to 8 for Task 5


int line = 0; //To keep track of the current line
char line_selection = 0;
char text[8][TOTAL_AMOUNT_CHARS] = { "", "", "", "", "", "", "", "" };

;Declaration of the differents methods that will be used
void toPutty(unsigned char);
void uart_int(void);
void line_switch(int);
char getChar();
char contains_character(char*, char);
void refresh_text(void);
void toSendToDisplay(char, char*, char*);


int main(void)
{
        uart_int(); //Method to initialize the display
        refresh_text();

        while (1) {
                ;Get the input from PuTTY
                char input = getChar();


                if (line_selection) {;If the line selection is selected
                    then do nothing and wait for a line number
                        if (input < '1') {
                                ;if the input is lower than 1 than do
                                   nothing and wait for another input
                                continue;
                        }
```

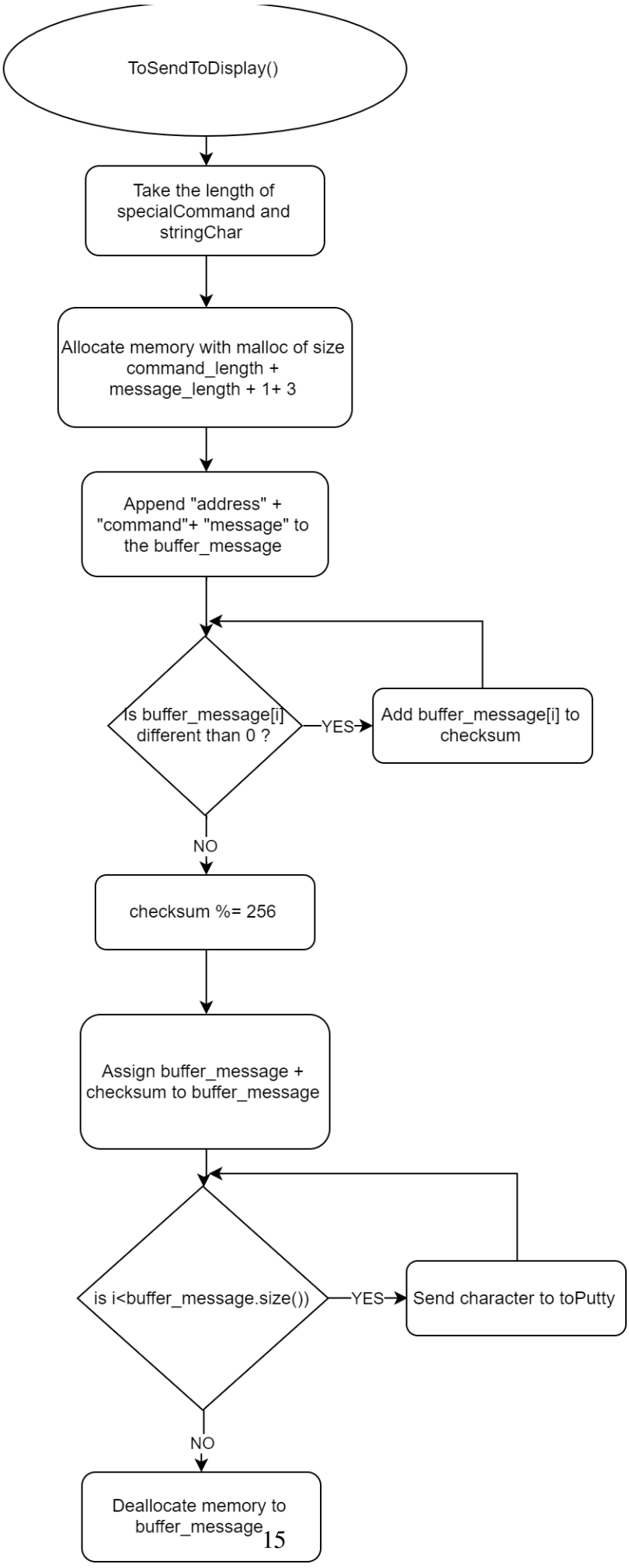
```
                        ;Check if the input is included in the valid
                            digits if it is then go inside if otherwise
                             skip the if statement
                        if (contains_character(Valid_Digits, input)) {

                                line_selection = 0;
                                line_switch((input - '1')); // turn the
                                     input into sterile int

                        }
                } ;if there is no line selection then the code goes
                    here
                else {
                        ;if the input is equal to '>' then change the
                            line_selection to 1 (true)
                        if (input == SELECT_LINE){

                                line_selection = 1;

                        }else if (input == 13 ){ ;Otherwise if the
                            input is the carriage return(ENTER) then
                            switch line

                                line_switch(-1);

                        }else {
                                ;Otherwise add the character to the
                                    corresponding lane
                                char* line = text[line];
                                sprintf(line, "%s%c", line, input);
                        }
                }
                ;Update the screen with with the corresponding changes
                refresh_text();
        }

        return 0;
}

//
void toPutty(unsigned char data){

        while(!(UCSR1A & (1<<UDRE1))){
                ;Do nothing while no data has been sent
        }
        UDR1 = data;
}

;To initialize the display
void uart_int(void) {
        UBRR1L = MY_UBBRR; //Set the Baud Rate to 25.

        UCSR1B = (1<<RXEN1|1<<TXEN1); //Enable Receive and Transmit bit
}

char getChar(){

        while(!(UCSR1A & (1<<RXC1))){
                ;While no data has been received, do nothing
```

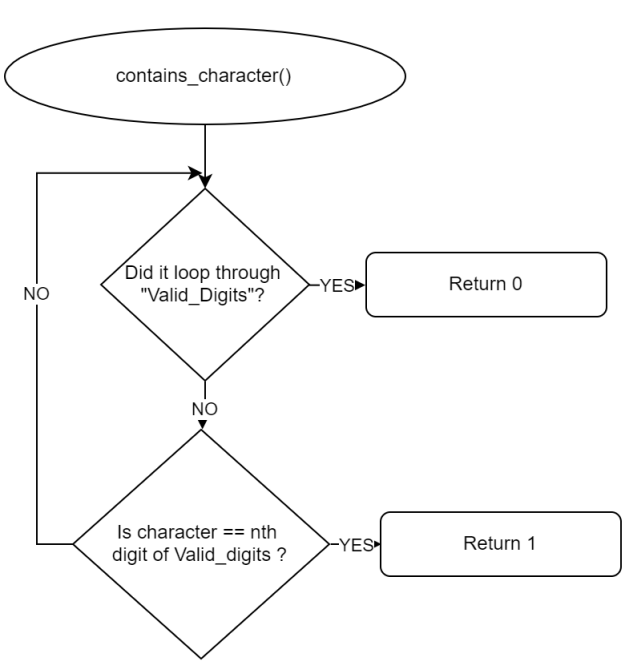
```
        }
        return UDR1; //return the received char.
}


;//Method to send the characters to the Display
void toSendToDisplay(char address, char* command, char* message)
{
        ; Get the lengths of the command characters and of the message
        int command_length = sizeof(command);
        int message_length = sizeof(message);

        ; Calculate how big the buffer needs to be depending on the
           message, command.
        int buffer_length = 1 + command_length + message_length + 3;

        ;Will add the adress + command + message + checksum, together
           to then send it to the screen
        char* buffer_message = malloc(buffer_length);

        ;Create the buffer with all the info needed
        sprintf(buffer_message, "\r%c%s%s", address, command, message);

        ;Checksum calculation
        unsigned int checksum = 0;
        for (int i = 0; (buffer_message[i] != 0); i++){
                checksum += buffer_message[i];
        }


        checksum %= 256;

        ;Add the checksum to the buffer
        sprintf(buffer_message, "%s%02X\n", buffer_message, checksum);


        for (int i = 0; buffer_message[i]; i++){
                toPutty(buffer_message[i]);
        }

        ;To free the space from memory
        free(buffer_message);
}

;Method to check if the char "character" is in the "string". If it is
   return 1 otherwise return 0.
char contains_character(char* string, char character)
{
        char t;
        while ((t = *string++)){
                if (t == character) {
                        return 1;
                }
        }
        return 0;
}

;Method to change between each line. if "-1" is sent then it will
   change the line.
```

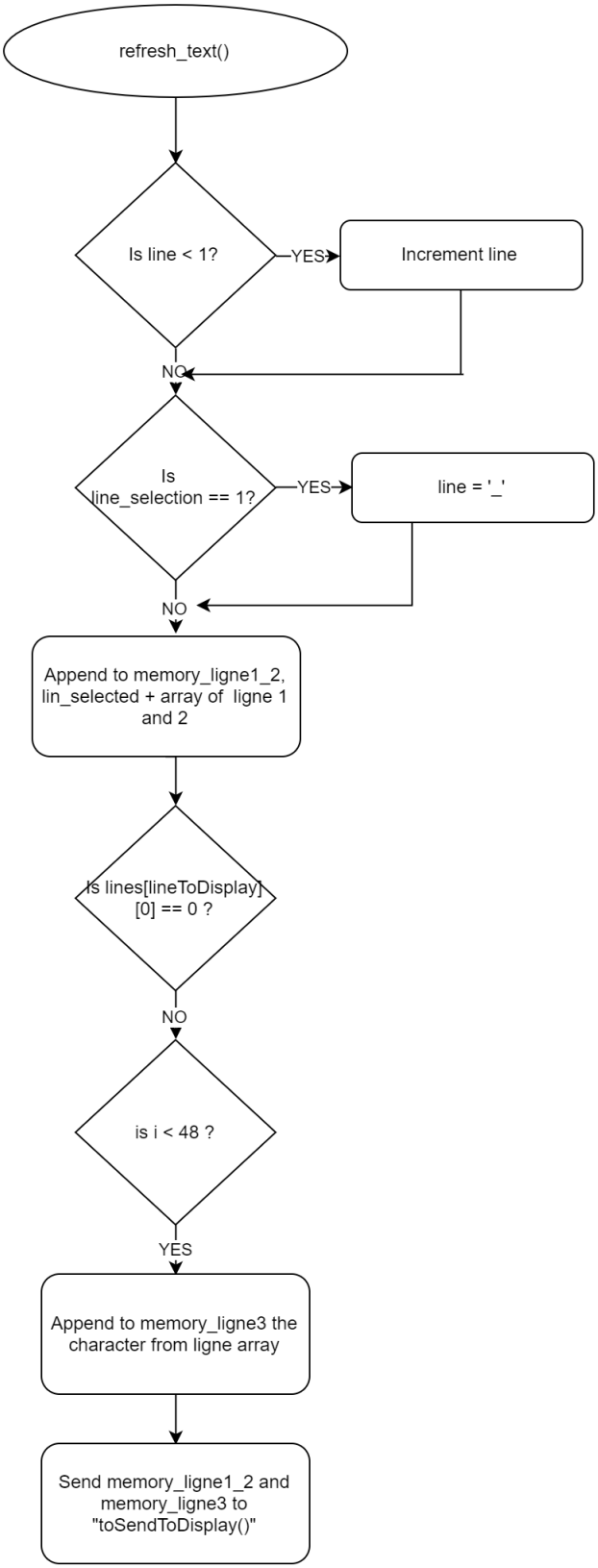
```
void line_switch(int number)
{
        ;if numver =-1 then increment the current line
        if (number == -1) {
                line++;
                if (line >= TOTAL_AMOUNT_LINES)
                line = 0;
        }else {
                line = number;
        }

}

;To update the text on the display
void refresh_text()
{
        ;To have the line to display
        int lineToDisplay = line;
        if (lineToDisplay < 1){
        lineToDisplay++;
        }
        ;variable to set up the first and second line
        char memory_ligne1_2[48] = "";
        char line_selected = line_selection ? '_' : (line + '1');


        sprintf(memory_ligne1_2, "Choose line: %c          %s",
           line_selected, text[lineToDisplay-1]);

        ;Creates the third ligne
        char memory_ligne3[48] = " ";
        if (text[lineToDisplay][0]){ //if the character is "0" do
           nothing otherwise send to "toSendToDisplay"
                ;DO nothing
        }else {
                for (int i = 0; i < 48; i++){
                        memory_ligne3[i] = text[lineToDisplay][i];
                }
        }
        ; Updates all the lignes of the screen.
        toSendToDisplay('A', "O0001", memory_ligne1_2);
        toSendToDisplay('B', "O0001", memory_ligne3);
        toSendToDisplay('Z', "D001", 0);
}
```

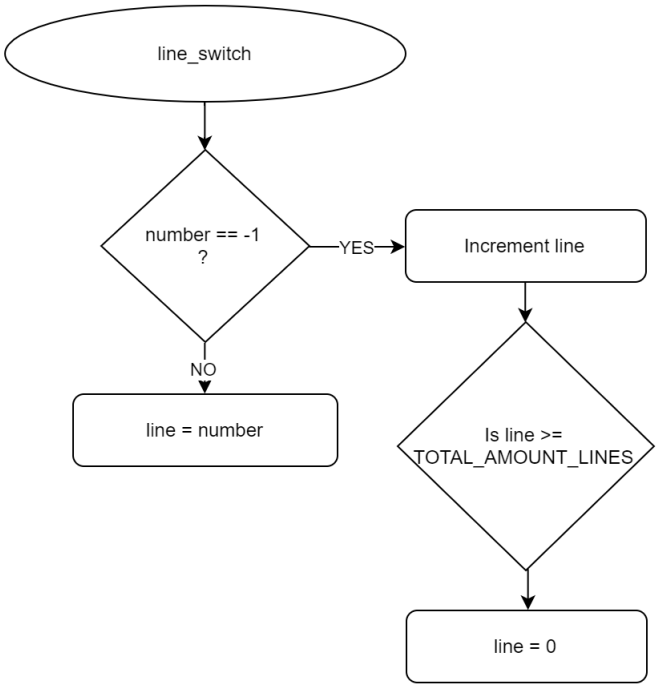
This is the flowcharts of the task 4:


Start
Initialize Display
While 1 is true
Assigns value from getChar() to "input"
Is "line_selection" == true ?
NO
input == '>' ?
YES
Set the variable line_selection to true
YES
NO
Is input < 1 ?
NO
input == 13 ? (ENTER)
NO
Add input to the corresponding line
YES
YES
line_switch()
is "input" in "Valid_Digits"? (contains_character)
YES
line_swtich()
Refresh_text()
Return 0


ToSendToDisplay()
Take the length of specialCommand and stringChar
Allocate memory with malloc of size command_length + message_length + 1+ 3
Append "address" + "command"+ "message" to the buffer_message
Is buffer_message[i] different than 0 ?
YES
Add buffer_message[i] to checksum
NO
checksum %= 256
Assign buffer_message + checksum to buffer_message
is i<buffer_message.size())
YES
Send character to toPutty
NO
Deallocate memory to buffer_message


contains_character()
Did it loop through "Valid_Digits"?
YES
Return 0
NO
NO
Is character == nth digit of Valid_digits ?
YES
Return 1


refresh_text()
Is line < 1?
YES
Increment line
NO
Is line_selection == 1?
YES
line = '_'
NO
Append to memory_ligne1_2, lin_selected + array of ligne 1 and 2
Is lines[lineToDisplay] [0] == 0 ?
NO
is i < 48 ?
YES
Append to memory_ligne3 the character from ligne array
Send memory_ligne1_2 and memory_ligne3 to "toSendToDisplay()"


line_switch
number == -1 ?
YES
Increment line
NO
line = number
Is line >= TOTAL_AMOUNT_LINES
line = 0

5 Task 5 - Write a program for text input.

This exercise works exactly like Task 4. Therefore please refer to task 4.