

Aufgabe 1

1.3)

```
#include <LiquidCrystal.h>
/** INITIALISIERUNG ***/
LiquidCrystal lcd(8, 9, 2, 3, 4, 5); // set up LCD
const int buttons[] = {33, 35, 37};
const int led      = 31;
const int disp     = 51;
/** GETRAENKELISTE ***/
int DRINKS = 6;
char *drink_name[] = {"Wasser", "Limo", "Bier", "Sprudel", "Eistee", "Saft"};
int  drink_price[] = {100, 150, 250, 100, 200, 150 }; // Preise in Cent

/** HAUPMENUE NAVIGATION ***/
int selected_item_in_menu = 0; // Getraenke ID (entspricht Array Position)
int menu_cursor_position  = 0; // 0 oder 1, Zeilenauswahl des Displays

/** TASTER ENTPRELLEN ***/
unsigned long last_select_time[3] = {0};
int pushed[3] = {0};

/** STATEMACHINE IN DER LOOP ***/
int active_menu = 0;

/** CUSTOM CHARS ***/
// Custom Char generiert auf: http://fusion94.org/lcdchargen/
byte custom_char_1[8] = {
    0b00000,
    0b00000,
    0b01000,
    0b01100,
    0b01110,
    0b01100,
    0b01000,
    0b00000
};

byte custom_char_2[8] = {
    0b00000,
    0b00111,
    0b01000,
    0b11110,
    0b10000,
    0b11110,
```

0b00111

```
};  
/*****  
/*** FUNKTIONEN DEFINIEREN */  
/*****/  
  
/*  
    Gibt Wert zurueck, der in gegebenen Grenzen liegt  
*/  
int limit (int value, int min, int max) {  
    if (value < min)  
        return max;  
    else if (value > max)  
        return min;  
  
    return value;  
}  
  
/*  
    Hauptmenue auf dem Display anzeigen, abhaengig von Cursorposition  
    mittels [selected_item_in_menu - menu_cursor_position (+ 1)]  
    wird dafuer gesorgt, dass das Display sich nur "verschiebt", wenn  
    Cursor nicht mehr bewegt werden kann  
*/  
void print_menu() {  
    char buffer[100]; //buffer fuer sprintf  
  
    lcd.clear();  
  
    lcd.setCursor(2, 0); //Setze Curser auf Anfangsposition des oberen Namens  
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position]);  
    sprintf(buffer, "%d,%02d",  
        drink_price[selected_item_in_menu - menu_cursor_position] / 100,  
        drink_price[selected_item_in_menu - menu_cursor_position] % 100);  
    lcd.setCursor(11, 0); //Setze Curser auf Anfangsposition des oberen Preises  
    lcd.print(buffer);  
    lcd.setCursor(15, 0);  
    lcd.write(byte(1));  
  
    lcd.setCursor(2, 1); //Setze Curser auf Anfangsposition des unteren Namens  
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position + 1]);  
    sprintf(buffer, "%d,%02d",  
        drink_price[selected_item_in_menu - menu_cursor_position + 1] / 100,  
        drink_price[selected_item_in_menu - menu_cursor_position + 1] % 100);  
    lcd.setCursor(11, 1); //Setze Curser auf Anfangsposition des unteren Preises  
    lcd.print(buffer);  
    lcd.setCursor(15, 0);
```

```

//setze Auswahldreieck
lcd.setCursor(0, menu_cursor_position);
lcd.write(byte(0)); //die ID des CustomChars (0) muss erst als byte
                      // geparsed werden
}

/*
  Gibt Button Input entprellt als int zurueck, beachte
  Invertierung der Logik durch INPUT_PULLUP
  0: button_up
  1: button_down
  2: button_ok
*/
int read_input () {
  for (int i = 0; i < 3; i++) {
    if (millis() - last_select_time[i] > 25) {
      if (digitalRead(buttons[i]) == LOW && !pushed[i]) {
        pushed[i] = 1;
        last_select_time[i] = millis();

        return i;
      } else if (digitalRead(buttons[i]) == HIGH && pushed[i]) {
        pushed[i] = 0;
        last_select_time[i] = millis();
      }
    }
  }
  return -1;
}

/*
  Aendert Hauptmenue Variablen je nach Eingabe ab, wird nur aufgerufen,
  wenn gerade das Hauptmenue aktiv ist
  Sorgt ausserdem dafuer, dass der Cursor zuerst hoch/runter sprint,
  bevor das Ganze Display "verschoben" wird
*/
void compute_menu_input(int input) {
  if (input == 0) {
    menu_cursor_position = 0; //cursor bewegt sich in obere Reihe
    selected_item_in_menu = limit(selected_item_in_menu - 1, 0, DRINKS - 1);
    //verschiebe Auswahl um 1 nach oben, achte aber auf Grenzen
    if(selected_item_in_menu == DRINKS-1){
      menu_cursor_position = 1;
    }
  } else if (input == 1) {
    menu_cursor_position = 1; //cursor bewegt sich in untere Reihe
    selected_item_in_menu = limit(selected_item_in_menu + 1, 0, DRINKS - 1);
  }
}

```

```

        if(selected_item_in_menu == 0){
            menu_cursor_position = 0;
        }
    }
}

/*
    Gibt Auswahl auf dem Display aus
    Waehrend dieser Anzeige werden jegliche Eingaben ignoriert
*/
void print_selection() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sie kauften:");
    lcd.setCursor(0, 1);
    lcd.print(drink_name[selected_item_in_menu]);

    delay(3000); //Bestaetigung 3 Sekunden anzeigen + Eingaben ignorieren
}

/*****
*** SETUP + LOOP ***
*****/

void setup() {
    //IO einrichten
    for (int i = 0; i < 3; i++)
        pinMode(buttons[i], INPUT_PULLUP);
    pinMode(led, OUTPUT);
    //Setze Schaltpin des Transistors als Ausgang
    // und standardmaessig auf HIGH
    pinMode(disg, OUTPUT);
    digitalWrite(disg, HIGH);
    //lcd einrichten
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.createChar(0, custom_char_1); //definiere ein chustomChar
                                     // (Anzeigen mit Serial.write(byte(0));)
    lcd.createChar(1, custom_char_2);

    print_menu(); //initiales Anzeigen des Haupmenues
}

void loop() {
    //lese und entprelle Taster,
    // bei -1 soll nichts passieren, da keine Eingabe

```

```

if (active_menu == 0) { //Hauptmenue, Getraenkeauswahl

    if (input == 2) {
        //bestaetige Auswahl
        active_menu = 1;
    } else if (input == 0 || input == 1) {
        //veraendere Auswahl
        compute_menu_input(input);
        print_menu();
    }

} else if (active_menu == 1) { //Bestaetige Auswahl auf Display

    print_selection();
    //resette alle Parameter
    active_menu = 0; //setze Statusvariable zurueck; koennte
                    //auch Wechsel in weitere Status initiieren
    selected_item_in_menu = 0;
    menu_cursor_position = 0;
    print_menu();
}
}

```

1.4)

```
#include <LiquidCrystal.h>
/** INITIALISIERUNG ***/
LiquidCrystal lcd(8, 9, 2, 3, 4, 5); // set up LCD
const int buttons[] = {33, 35, 37};
const int led      = 31;
const int disp     = 51;
/** GETRAENKELISTE ***/
int DRINKS = 9;
char *drink_name[] = {"Wasser", "Limo", "Bier", "Sprudel", "Eistee",
                      "Saft", "KornMitEistee", "VodkaEnergy", "MoscowMule"};
int drink_price[] = { 100, 150, 250, 100, 200, 150, 50, 100, 500 };
//Preise in Cent
/** HAUPMENUE NAVIGATION ***/
int selected_item_in_menu = 0; //Getraenke ID (entspricht Array Position)
int menu_cursor_position  = 0; //0 oder 1, Zeilenauswahl des Displays
/** TASTER ENTPRELLLEN ***/
unsigned long last_select_time[3] = {0};
int pushed[3] = {0};
/** STATEMACHINE IN DER LOOP ***/
int active_menu = 0;
/** CUSTOM CHARS ***/
//Custom Char generiert auf: http://fusion94.org/lcdchargen/
byte custom_char_1[8] = {
    0b00000,
    0b00000,
    0b01000,
    0b01100,
    0b01110,
    0b01100,
    0b01000,
    0b00000
};
byte custom_char_2[8] = {
    0b00000,
    0b00111,
    0b01000,
    0b11110,
    0b10000,
    0b11110,
    0b01000,
    0b00111
};
/**
*****
*** FUNKTIONEN DEFINIEREN *
*****
*/
/*
    Gibt Wert zurueck, der in gegebenen Grenzen liegt
*/
```

```

*/
int limit (int value, int min, int max) {
    if (value < min)
        return max;
    else if (value > max)
        return min;
    return value;
}

/*
    Hauptmenue auf dem Display anzeigen, abhaengig von Cursorposition
    mittels [selected_item_in_menu - menu_cursor_position (+ 1)]
    wird dafuer gesorgt, dass das Display sich nur "verschiebt", wenn
    Cursor nicht mehr bewegt werden kann
*/

void print_menu() {
    char buffer[100]; //buffer fuer sprintf
    lcd.clear();
    lcd.setCursor(2, 0); //Setze Curser auf Anfangsposition des oberen Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position]);
    sprintf(buffer, "%d,%02d",
        drink_price[selected_item_in_menu - menu_cursor_position] / 100,
        drink_price[selected_item_in_menu - menu_cursor_position] % 100);
    lcd.setCursor(11, 0); //Setze Curser auf Anfangsposition des oberen Preises
    lcd.print(buffer);
    lcd.setCursor(15, 0);
    lcd.write(byte(1));
    lcd.setCursor(2, 1); //Setze Curser auf Anfangsposition des unteren Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position + 1]);
    sprintf(buffer, "%d,%02d",
        drink_price[selected_item_in_menu - menu_cursor_position + 1] / 100,
        drink_price[selected_item_in_menu - menu_cursor_position + 1] % 100);
    lcd.setCursor(11, 1); //Setze Curser auf Anfangsposition des unteren Preises
    lcd.print(buffer);
    lcd.setCursor(15, 0);
    lcd.write(byte(1));
    //setze Auswahldreieck
    lcd.setCursor(0, menu_cursor_position);
    lcd.write(byte(0)); //die ID des CustomChars (0) muss erst als byte
                        // geparsed werden
}

/*
    Gibt Button Input entprellt als int zurueck, beachte Invertierung
    der Logik durch INPUT_PULLUP
    0: button_up
    1: button_down
    2: button_ok
*/

int read_input () {
    for (int i = 0; i < 3; i++) {

```

```

    if (millis() - last_select_time[i] > 25) {
        if (digitalRead(buttons[i]) == LOW && !pushed[i]) {
            pushed[i] = 1;
            last_select_time[i] = millis();
            return i;
        } else if (digitalRead(buttons[i]) == HIGH && pushed[i]) {
            pushed[i] = 0;
            last_select_time[i] = millis();
        }
    }
}
return -1;
}
/*
Aendert Hauptmenue Variablen je nach Eingabe ab, wird nur aufgerufen,
wenn gerade das Hauptmenue aktiv ist
Sorgt ausserdem dafuer, dass der Cursor zuerst hoch/runter sprint,
bevor das Ganze Display "verschoben" wird
*/
void compute_menu_input(int input) {
    if (input == 0) {
        menu_cursor_position = 0; // cursor bewegt sich in obere Reihe
        selected_item_in_menu = limit(selected_item_in_menu - 1, 0, DRINKS - 1);
        // verschiebe Auswahl um 1 nach oben, achte aber auf Grenzen
        if (selected_item_in_menu == DRINKS - 1) {
            menu_cursor_position = 1;
        }
    } else if (input == 1) {
        menu_cursor_position = 1; // cursor bewegt sich in untere Reihe
        selected_item_in_menu = limit(selected_item_in_menu + 1, 0, DRINKS - 1);
        // verschiebe Auswahl um 1 nach unten, achte aber auf Grenzen
        if (selected_item_in_menu == 0) {
            menu_cursor_position = 0;
        }
    }
}
/*
Gibt Auswahl auf dem Display aus
Waehrend dieser Anzeige werden jegliche Eingaben ignoriert
*/
void print_selection() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sie kauften:");
    lcd.setCursor(0, 1);
    lcd.print(drink_name[selected_item_in_menu]);
    delay(3000); // Bestaetigung 3 Sekunden anzeigen + Eingaben ignorieren
}
int getSize(char* ch) {

```



```

    int tmp=0;
    while (*ch){
        *ch++;
        tmp++;
    }
    return tmp;
}

void checkLength(){
    for(int i = 0; i < DRINKS; i++){
        if(getSize(drink_name[i])>8){
            drink_name[i][7] = '.';
            drink_name[i][8] = '\\0';
        }
    }
}

/***** SETUP + LOOP *****/
/***** SETUP + LOOP *****/

void setup() {
    //IO einrichten
    checkLength();
    for (int i = 0; i < 3; i++)
        pinMode(buttons[i], INPUT_PULLUP);
    pinMode(led, OUTPUT);
    //Setze Schaltpin des Transistors als Ausgang und standardmaessig auf HIGH
    pinMode(disg, OUTPUT);
    digitalWrite(disg, HIGH);
    //lcd einrichten
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.createChar(0, custom_char_1); //definiere ein chustomChar
                                     // (Anzeigen mit Serial.write(byte(0));)
    lcd.createChar(1, custom_char_2);
    print_menu(); //initiales Anzeigen des Hauptmenues
}

void loop() {
    //lese und entprelle Taster, bei -1 soll nichts passieren, da keine Eingabe
    int input = read_input();
    if (active_menu == 0) { //Hauptmenue, Getraenkeauswahl
        if (input == 2) {
            //bestaetige Auswahl
            active_menu = 1;
        } else if (input == 0 || input == 1) {
            //veraendere Auswahl
            compute_menu_input(input);
            print_menu();
        }
    } else if (active_menu == 1) { //Bestaetige Auswahl auf Display
        print_selection();
    }
}

```

```
//resette alle Parameter
active_menu = 0;//setze Statusvariable zurueck; koennte auch
                //Wechsel in weitere Status initiieren
selected_item_in_menu = 0;
menu_cursor_position = 0;
print_menu();
}
}
```

1.5)

```
#include <LiquidCrystal.h>
/** INITIALISIERUNG ***/
LiquidCrystal lcd(8, 9, 2, 3, 4, 5); // set up LCD
const int buttons[] = {33, 35, 37};
const int led      = 31;
const int disp     = 51;

/** GETRAENKELISTE ***/
int DRINKS = 6;
char *drink_name[] = {"Wasser", "Limo", "Bier", "Sprudel", "Eistee", "Saft"};
int  drink_price[] = {100, 150, 250, 100, 200, 150 }; // Preise in Cent

int paid= 0;

/** HAUPMENUE NAVIGATION ***/
int selected_item_in_menu = 0; // Getraenke ID (entspricht Array Position)
int menu_cursor_position  = 0; // 0 oder 1, Zeilenauswahl des Displays

/** TASTER ENTPRELLEN ***/
unsigned long last_select_time[3] = {0};
int pushed[3] = {0};

/** STATEMACHINE IN DER LOOP ***/
int active_menu = 0;

/** CUSTOM CHARS ***/
// Custom Char generiert auf: http://fusion94.org/lcdchargen/
byte custom_char_1[8] = {
    0b00000,
    0b00000,
    0b01000,
    0b01100,
    0b01110,
    0b01100,
    0b01000,
    0b00000
};

byte custom_char_2[8] = {
    0b00000,
    0b00111,
    0b01000,
    0b11110,
    0b10000,
    0b11110,
    0b01000,
    0b00111
};
```

```

};
/*****
/**** FUNKTIONEN DEFINIEREN *****/
/*****/

/*
    Gibt Wert zurueck, der in gegebenen Grenzen liegt
*/
int limit (int value, int min, int max) {
    if (value < min)
        return max;
    else if (value > max)
        return min;

    return value;
}

/*
    Hauptmenue auf dem Display anzeigen, abhaengig von Cursorposition
    mittels [selected_item_in_menu - menu_cursor_position (+ 1)] wird
    dafuer gesorgt,
    dass das Display sich nur "verschiebt", wenn Cursor nicht mehr
    bewegt werden kann
*/
void print_menu() {
    char buffer[100]; //buffer fuer sprintf

    lcd.clear();

    lcd.setCursor(2, 0); //Setze Curser auf Anfangsposition des oberen Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position]);
    sprintf(buffer, "%d,%02d",
        drink_price[selected_item_in_menu - menu_cursor_position] / 100,
        drink_price[selected_item_in_menu - menu_cursor_position] % 100);
    lcd.setCursor(11, 0); //Setze Curser auf Anfangsposition des oberen Preises
    lcd.print(buffer);
    lcd.setCursor(15, 0);
    lcd.write(byte(1));

    lcd.setCursor(2, 1); //Setze Curser auf Anfangsposition des unteren Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position + 1]);
    sprintf(buffer, "%d,%02d",
        drink_price[selected_item_in_menu - menu_cursor_position + 1] / 100,
        drink_price[selected_item_in_menu - menu_cursor_position + 1] % 100);
    lcd.setCursor(11, 1); //Setze Curser auf Anfangsposition des unteren Preises
    lcd.print(buffer);
    lcd.setCursor(15, 1);
    lcd.write(byte(1));
}

```

```

//setze Auswahldreieck
lcd.setCursor(0, menu_cursor_position);
lcd.write(byte(0)); //die ID des CustomChars (0) muss erst als
                    // byte geparsed werden
}

/*
  Gibt Button Input entprellt als int zurueck, beachte
  Invertierung der Logik durch INPUT_PULLUP
  0: button_up
  1: button_down
  2: button_ok
*/
int read_input () {
  for (int i = 0; i < 3; i++) {
    if (millis() - last_select_time[i] > 25) {
      if (digitalRead(buttons[i]) == LOW && !pushed[i]) {
        pushed[i] = 1;
        last_select_time[i] = millis();

        return i;
      } else if (digitalRead(buttons[i]) == HIGH && pushed[i]) {
        pushed[i] = 0;
        last_select_time[i] = millis();
      }
    }
  }
  return -1;
}

/*
  Aendert Hauptmenue Variablen je nach Eingabe ab, wird nur aufgerufen,
  wenn gerade das Hauptmenue aktiv ist
  Sorgt ausserdem dafuer, dass der Cursor zuerst hoch/runter sprint,
  bevor das Ganze Display "verschoben" wird
*/
void compute_menu_input(int input) {
  if (input == 0) {
    menu_cursor_position = 0; //cursor bewegt sich in obere Reihe
    selected_item_in_menu = limit(selected_item_in_menu - 1, 0, DRINKS - 1);
    //verschiebe Auswahl um 1 nach oben, achte aber auf Grenzen
    if(selected_item_in_menu == DRINKS-1){
      menu_cursor_position = 1;
    }
  } else if (input == 1) {
    menu_cursor_position = 1; //cursor bewegt sich in untere Reihe
    selected_item_in_menu = limit(selected_item_in_menu + 1, 0, DRINKS - 1);
    //verschiebe Auswahl um 1 nach unten, achte aber auf Grenzen

```

```

        if(selected_item_in_menu == 0){
            menu_cursor_position = 0;
        }
    }
}

/*
    Gibt Auswahl auf dem Display aus
    Waehrend dieser Anzeige werden jegliche Eingaben ignoriert
*/
void print_selection() {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sie kauften:");
    lcd.setCursor(0, 1);
    lcd.print(drink_name[selected_item_in_menu]);

    delay(500); //Bestaetigung 3 Sekunden anzeigen + Eingaben ignorieren
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
    delay(500);
    digitalWrite(led,HIGH);
    digitalWrite(led,LOW);
}

void print_payment() {
    char buffer[100];

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(drink_name[selected_item_in_menu]);
    lcd.setCursor(0, 1);
    sprintf(buffer, "%d,%02d",
            drink_price[selected_item_in_menu] / 100,
            drink_price[selected_item_in_menu] % 100);

    lcd.print(buffer);

```

```

    lcd.setCursor(4, 1);
    lcd.write(byte(1));

    lcd.setCursor(7, 1);
    sprintf(buffer, "%d,%02d",
            paid / 100,
            paid % 100);

    lcd.print(buffer);
    lcd.setCursor(13, 1);
    lcd.write(byte(1));

}

/*****
*** SETUP + LOOP */
*****/

void setup() {
    //IO einrichten
    for (int i = 0; i < 3; i++)
        pinMode(buttons[i], INPUT_PULLUP);
    pinMode(led, OUTPUT);

    //Setze Schaltpin des Transistors als Ausgang und standardmaessig auf HIGH
    pinMode(dis, OUTPUT);
    digitalWrite(dis, HIGH);

    //lcd einrichten
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.createChar(0, custom_char_1); //definiere ein chustomChar
                                     // (Anzeigen mit Serial.write(byte(0));)
    lcd.createChar(1, custom_char_2);

    print_menu(); //initiales Anzeigen des Hauptmenues
}

void loop() {
    //lese und entprelle Taster, bei -1 soll nichts passieren, da keine Eingabe
    int input = read_input();

    if (active_menu == 0) { //Hauptmenue, Getraenkeauswahl

        if (input == 2) {
            //bestaetige Auswahl
            active_menu = 1;

```

```

    print_payment();
}else if (input == 0 || input == 1) {
    //veraendere Auswahl
    compute_menu_input(input);
    print_menu();
}

}else if (active_menu == 1) { //Bestaetige Auswahl auf Display

    if( input == 0){
        paid = paid+50;
        print_payment();
    }else if(input == 1){
        paid = paid+100;
        print_payment();
    }else if(input == 2 && paid >= drink_price[selected_item_in_menu]){
        print_selection();
        active_menu = 0; //setze Statusvariable zurueck; koennte auch
                        //Wechsel in weitere Status initiieren

        selected_item_in_menu = 0;
        menu_cursor_position = 0;
        paid = 0;
        print_menu();
    }
}
}
}

```


1.6)

```
#include <LiquidCrystal.h>
/** INITIALISIERUNG **/
LiquidCrystal lcd(8, 9, 2, 3, 4, 5); // set up LCD
const int buttons[] = {33, 35, 37};
const int led      = 31;
const int disp     = 51;

/** GETRAENKELISTE **/
int DRINKS = 6;
char *drink_name[] = {"Wasser", "Limo", "Bier", "Sprudel", "Eistee", "Saft"};
int  drink_price[] = {100, 150, 250, 100, 200, 150 }; // Preise in Cent
int  drink_amount[] = {3,3,3,3,3,3};
int paid= 0;
int time= 0;

/** HAUPMENUE NAVIGATION **/
int selected_item_in_menu = 0; //Getraenke ID (entspricht Array Position)
int menu_cursor_position  = 0; //0 oder 1, Zeilenauswahl des Displays

/** TASTER ENTPRELLEN **/
unsigned long last_select_time[3] = {0};
int pushed[3] = {0};

/** STATEMACHINE IN DER LOOP **/
int active_menu = 0;

/** CUSTOM CHARS **/
//Custom Char generiert auf: http://fusion94.org/lcdchargen/
byte custom_char_1[8] = {
    0b00000,
    0b00000,
    0b01000,
    0b01100,
    0b01110,
    0b01100,
    0b01000,
    0b00000
};

byte custom_char_2[8] = {
    0b00000,
    0b00111,
    0b01000,
    0b11110,
    0b10000,
    0b11110,
```

```

0b01000,
0b00111
};
/*****
***  FUNKTIONEN DEFINIEREN  */
*****/

/*
    Gibt Wert zurueck, der in gegebenen Grenzen liegt
*/
int limit (int value, int min, int max) {
    if (value < min)
        return max;
    else if (value > max)
        return min;

    return value;
}

/*
    Hauptmenue auf dem Display anzeigen, abhaengig von Cursorposition
    mittels [selected_item_in_menu - menu_cursor_position (+ 1)] wird
    dafuer gesorgt,
    dass das Display sich nur "verschiebt", wenn Cursor nicht mehr
    bewegt werden kann
*/
void print_menu() {
    char buffer[100]; //buffer fuer sprintf

    lcd.clear();

    lcd.setCursor(2, 0); //Setze Curser auf Anfangsposition des oberen Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position]);
    if(drink_amount[selected_item_in_menu - menu_cursor_position] <= 0) {
        lcd.setCursor(11, 0); //Setze Curser auf Anfangsposition des oberen Preises
        lcd.print("----");
    } else {
        sprintf(buffer, "%d,%02d",
            drink_price[selected_item_in_menu - menu_cursor_position] / 100,
            drink_price[selected_item_in_menu - menu_cursor_position] % 100);
        lcd.setCursor(11, 0); //Setze Curser auf Anfangsposition des oberen Preises
        lcd.print(buffer);
    }
    lcd.setCursor(15, 0);
    lcd.write(byte(1));

    lcd.setCursor(2, 1); //Setze Curser auf Anfangsposition des unteren Namens
    lcd.print(drink_name[selected_item_in_menu - menu_cursor_position + 1]);
    if(drink_amount[selected_item_in_menu - menu_cursor_position + 1] <= 0) {

```

```

        lcd.setCursor(11, 1); //Setze Curser auf Anfangsposition des unteren Preises
        lcd.print("----");
    }else{
        sprintf(buffer, "%d,%02d",
            drink_price[selected_item_in_menu - menu_cursor_position + 1] / 100,
            drink_price[selected_item_in_menu - menu_cursor_position + 1] % 100);
        lcd.setCursor(11, 1); //Setze Curser auf Anfangsposition des unteren Preises
        lcd.print(buffer);
    }
    lcd.setCursor(15, 0);
    lcd.write(byte(1));

    //setze Auswahldreieck
    lcd.setCursor(0, menu_cursor_position);
    lcd.write(byte(0)); //die ID des CustomChars (0) muss erst als byte
                        // geparsed werden
}

/*
    Gibt Button Input entprellt als int zurueck, beachte Invertierung
    der Logik durch INPUT_PULLUP
    0: button_up
    1: button_down
    2: button_ok
*/
int read_input () {
    for (int i = 0; i < 3; i++) {
        if (millis() - last_select_time[i] > 25) {
            if (digitalRead(buttons[i]) == LOW && !pushed[i]) {
                pushed[i] = 1;
                last_select_time[i] = millis();

                return i;
            }else if (digitalRead(buttons[i]) == HIGH && pushed[i]) {
                pushed[i] = 0;
                last_select_time[i] = millis();
            }
        }
    }
    return -1;
}

/*
    Aendert Hauptmenue Variablen je nach Eingabe ab, wird nur aufgerufen,
    wenn gerade das Hauptmenue aktiv ist
    Sorgt ausserdem dafuer, dass der Cursor zuerst hoch/runter sprint,
    bevor das Ganze Display "verschoben" wird
*/
void compute_menu_input(int input) {

```

[illegible]

```

char buffer[100];

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(drink_name[selected_item_in_menu]);
lcd.setCursor(0, 1);
sprintf(buffer, "%d,%02d",
        drink_price[selected_item_in_menu] / 100,
        drink_price[selected_item_in_menu] % 100);

lcd.print(buffer);
lcd.setCursor(4, 1);
lcd.write(byte(1));

lcd.setCursor(7, 1);
sprintf(buffer, "%d,%02d",
        paid / 100,
        paid % 100);

lcd.print(buffer);
lcd.setCursor(13, 1);
lcd.write(byte(1));

}

/*****
*** SETUP + LOOP */
*****/

void setup() {
    //IO einrichten
    for (int i = 0; i < 3; i++)
        pinMode(buttons[i], INPUT_PULLUP);
    pinMode(led, OUTPUT);

    //Setze Schaltpin des Transistors als Ausgang und standardmaessig auf HIGH
    pinMode(dis, OUTPUT);
    digitalWrite(dis, HIGH);

    //lcd einrichten
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.createChar(0, custom_char_1); //definiere ein chustomChar
                                     // (Anzeigen mit Serial.write(byte(0)));
    lcd.createChar(1, custom_char_2);

    print_menu(); //initiales Anzeigen des Hauptmenues

```

```

}

void loop() {
//lese und entprelle Taster, bei -1 soll nichts passieren, da keine Eingabe
int input = read_input();

if (active_menu == 0) { //Hauptmenue, Getraenkeauswahl

    if (input == 2) {
        //bestaetige Auswahl
        if(drink_amount[selected_item_in_menu] > 0){
            active_menu = 1;
            print_payment();
        }else{
            active_menu = 2;
            time =millis();
            digitalWrite(dis, LOW);
        }
    }else if (input == 0 || input == 1) {
        //veraendere Auswahl
        compute_menu_input(input);
        print_menu();
    }

}

}else if (active_menu == 1) { //Bestaetige Auswahl auf Display

    if( input == 0){
        paid = paid+50;
        print_payment();
    }else if(input == 1){
        paid = paid+100;
        print_payment();
    }else if(input == 2 && paid >=drink_price[selected_item_in_menu]){
        print_selection();
        active_menu = 0; //setze Statusvariable zurueck; koennte auch
        // Wechsel in weitere Status initiieren

        selected_item_in_menu = 0;
        menu_cursor_position = 0;
        paid = 0;
        drink_amount[selected_item_in_menu]=
        drink_amount[selected_item_in_menu]-1;
        print_menu();
    }
}

}else if (active_menu==2){
    if(millis()-time >= 150){
        digitalWrite(dis, HIGH);
        active_menu=0;
        print_menu();
    }
}
}

```

}

}

Aufgabe 2

2.1)

a) Das Programm wird nicht übersetzt.

b) /

c)

```
void setup() {  
    char str[] = {'P','I','N','G','\0'};  
    strcat(str, str);  
}
```

```
void loop() {
```

```
}
```

2.2)

a) Das Programm wird nicht übersetzt

b) /

c)

```
int taster = 2;  
int led = 3;  
int tasterGedrueckt = 0;  
int zaehler = 0;  
void setup() {  
  
}  
  
void loop () {  
    tasterGedrueckt = digitalRead ( taster ) ;  
    if ( tasterGedrueckt == 1) {  
        zaehler ++;  
    }  
    if (5 < zaehler < 10) {  
        digitalWrite ( led ,HIGH ) ;  
    } else {  
        digitalWrite ( led ,LOW ) ;  
    }  
}
```

2.3)

a) Das Programm wird nicht übersetzt

b) /

c)

```
int led = 3;  
int naechsterWechsel = 0;  
void setup () {  
    naechsterWechsel =millis() + 300;  
    pinMode(led, OUTPUT);
```



```

    pinMode(led, INPUT);
}
void loop () {
    if (millis() > naechsterWechsel ) {
        digitalWrite (led,!digitalRead(led)) ;
        naechsterWechsel =millis() + 300;
    }
}

```

2.4)

a) Das Programm wird übersetzt, wenn man die leere setupmethode einfügt

b) Nein macht es nicht, da nicht mit einem

`digitalRead` auch wirklich etwas abgefragt wird.

c)

```

int taster = 2;
int led = 3;
void setup(){
    pinMode(taster, OUTPUT);
}
void loop () {
    if (digitalRead(taster) == HIGH ) {
        led =HIGH ;
    } else {
        led =LOW ;
    }
}

```

2.5)

a) Das Programm wird übersetzt wenn man eine leere loopmethode einfügt.

b) Nein macht es nicht, weil so nur led 7 und 8 als `OUTPUT` gepinnt werden, denn die `while` schleife überprüft auf kleiner 9.

c)

```

int led[] = {7, 8, 9};
void setup () {
    int i = 7;
    while (i < 10) {
        pinMode ( leds[i++],OUTPUT) ;
    }
}

void loop() {

}

```