

```
#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", "Cola", "Grundels", "Kaffee" };
```

```
int drink_price[] = { 100, 150, 250, 300, 300, 200 }; //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/
```

```
// Arrow sign
```

```
byte custom_char_1[8] = {
```

```
    0b000000,
```

```
    0b000000,
```

```
    0b01000,
```

```
    0b01100,
```

```
    0b01110,
```

```
    0b01100,
```

```
    0b01000,
```

```
    0b000000
```

```
};
```

```
/**/ ***** /**/
```

```
/**/ FUNKTIONEN DEFINIEREN */
```

```
/**/ ***** /**/
```

```
/*
```

```
    Gibt Wert zurueck, der in gegebenen Grenzen liegt
```

```
*/
```

```
int limit (int value, int min, int max) {
```

```
    if (value < min)
```

```
        return min;
```

```
    else if (value > max)
```

```
        return max;
```

```
    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(12, 0); //Setze Curser auf Anfangsposition des oberen Preises
    lcd.print(buffer);

    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(12, 1); //Setze Curser auf Anfangsposition des unteren Preises
    lcd.print(buffer);

    //setze Auswahl dreieck
    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 INI
    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
Sorgt ausserdem dafuer, dass der Cursor zuerst hoch/runter sprint, bevor das Ganze D
*/
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); //verschieb
    } 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); //verschieb
    }
}

/*
    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(dis, OUTPUT);
    digitalWrite(dis, HIGH);

    //lcd einrichten
    lcd.begin(16, 2);
    lcd.setCursor(0, 0);
    lcd.createChar(0, custom_char_1); //definiere ein customChar (Anzeigen mit Serial.wr

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

```