

Workshop: Datenbrille

Georg Jehle

Juni 2020

Übersicht

Teil 1: Datenbrillen und Linsenoptik

Teil 2: Microcontroller und Arduino

Teil 3: Programmieren

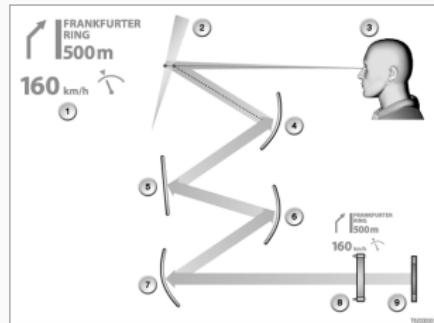
Datenbrillen

Idee hinter Datenbrillen

- Computergestützte Erweiterung des realen Sichtfelds (*augmented reality*)
- Zentrales Element: Überblendung von mehreren Bildprojektionen
Bsp.: Sichtfeld überblendet mit Meta-Informationen
- Zusätzliche Informationen werden genutzt als Unterstützung bei bestimmten Arbeiten, als Orientierungshilfe, Live-Analyse,
...
- Technische Umsetzung: viele Varianten

Datenbrillen: Beispiele i

Head-Up Display¹

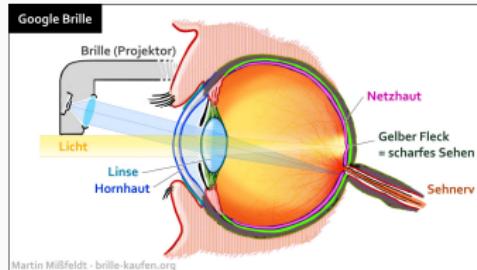


- Licht des Displays wird über ein Spiegelsystem auf die Windschutzscheibe projiziert und ist als Reflexion erkennbar
- An einer bestimmten Position der Fahreraugen ist alles sichtbar und scharf
- Meta-Informationen kommen aus Fahrdiagnostik und Internet

¹next-mobility.news, 28.6.2020

Datenbrillen: Beispiele ii

Google-Brille²



- Licht wird direkt auf die Netzhaut projiziert
- Anwender kann auf beliebige Stelle in Umgebung schauen, dann ist das Bild der Brille am Rand des Sichtfelds
- Datenbrille ist mit Internet verbunden

²notebookcheck.de, 28.6.2020

Datenbrillen: Beispiele iii

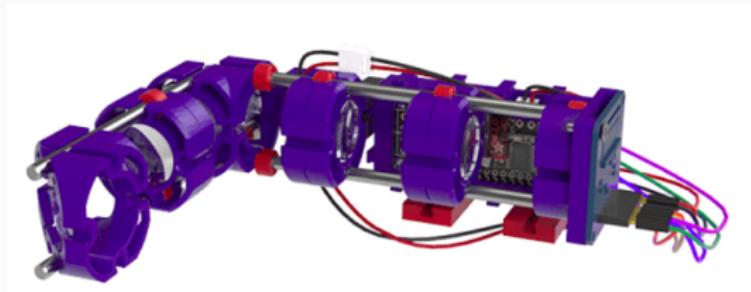
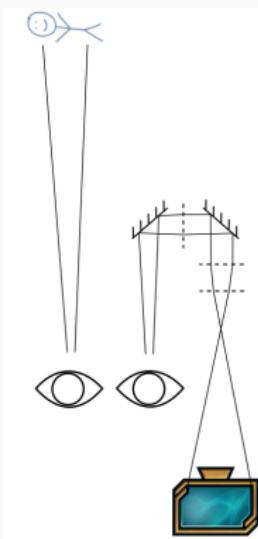
Datenbrille für Montagetechniker³



- Zusätzliche Kamera zur Übertragung des Sichtfelds
- Ziel: Kontrolle (Fehlervermeidung)
- Kommunikation über Bild und Audio

³vdi-nachrichten.de, 28.6.2020

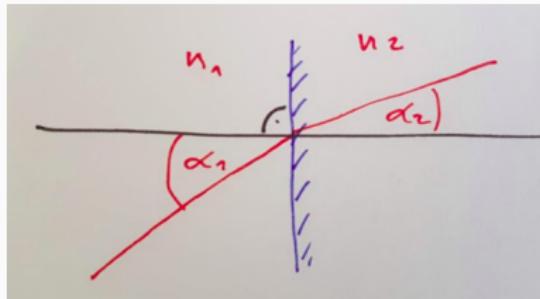
Datenbrille mit Fischertechnik



- Ein Auge sieht Umgebung, das andere ein OLED-Display
→ Überblendung findet im Kopf statt
- Umlenkkoptik auf OLED-Display durch drei plankonvexe Linsen und zwei Spiegel
- Anzeige muss von uns programmiert werden

Linsenoptik i

Lichtbrechung an einer Grenzfläche



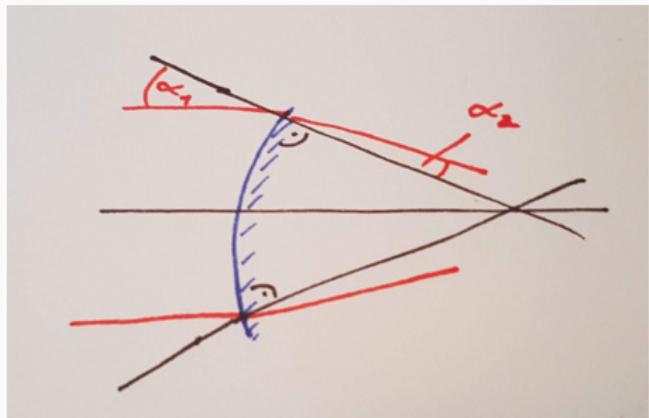
Physikalische Eigenschaften von lichtdurchlässigen Medien beeinflussen die Lichtgeschwindigkeit. An Grenzflächen führt das zur Lichtbrechung.
Beispiel: Person im Wasser

- Brechungsindex: n_1, n_2
- Lichtgeschwindigkeit: c_1, c_2
- Winkel: α_1, α_2

$$\frac{\sin(\alpha_1)}{\sin(\alpha_2)} = \frac{n_2}{n_1} = \frac{c_1}{c_2}$$

Linsenoptik ii

Auswirkung auf Strahlengang durch konvexe Grenzfläche

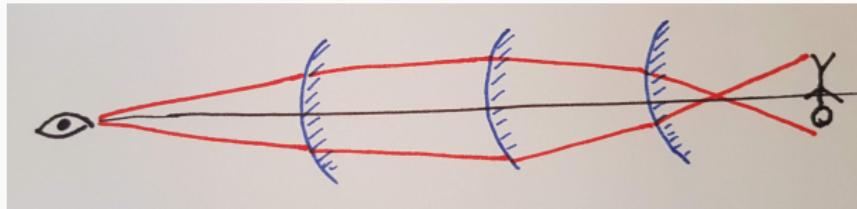


Medium 1: Luft, Medium 2: Glas. $n_2 > n_1 \rightarrow \alpha_1 > \alpha_2$

Licht wird gesammelt

Linsenoptik iii

Mehrere Linsen hintereinander wie in unserer Datenbrille



Das Licht wird mehrmals gebündelt und dadurch nach der dritten Stufe in einem Fokus invertiert. Ein zweiter Fokus der Lichtstrahlen vom Display befindet sich im Auge, sodass das Bild hier scharf ist. Gleichzeitig kann eine Vergrößerung eingestellt werden.

Parameter: Abstand zwischen den Linsen

Spiegelung vs. Invertierung

Spiegelung: Lichtstrahlen werden an einer Ebene vertauscht
(z.B. rechts wird zu links, oben wird zu unten)

Invertierung: Lichtstrahlen werden in einem Punkt (Fokus)
vertauscht. Dann wird rechts oben zu links unten etc.

Microcontroller und Arduino i

Microcontroller: Ein-Chip-Computersystem mit Peripheriefunktionen (Anschlüsse: USB, Ethernet, speziellere Anschlüsse). Einsatz heutzutage in fast allen technischen Geräten wie Drucker, Smartphone, Waschmaschine, ... Vorteile: sparsam (geringer Energieverbrauch im Vergleich mit PC), kostengünstig, wenig Bauraum. Nachteile: Leistung der Datenverarbeitung und Geschwindigkeit sind nicht so hoch.

Arduino: Open-Source-Platform bestehend aus Hardware (Board, z.B. Uno oder Feather) und Software (Arduino IDE)

Download der Installationsdatei der aktuellen Softwareversion:
<https://www.arduino.cc/en/main/software>

Konfiguration: Bibliotheken für OLED-Display (`adafruit_ss1306`) und Board (Feather ist in `arduino_samd` enthalten) einbinden

Microcontroller und Arduino ii

Übersicht

The screenshot shows the Arduino IDE interface with the following components:

- Menüleiste:** Datei, Bearbeiten, Sketch, Werkzeuge, Hilfe.
- Werkzeuge:** Standard icons for upload, refresh, and search.
- Textfeld (Programm):** The main code editor window containing the following C++ code for a BLE dataglass module:

```
1 //based on the Adafruit BluefruitLE library (SPI)
2 // Plan:
3 // Die Brille soll Daten wie zum Beispiel die Temperatur (von einem
4
5
6 //https://github.com/adafruit/Adafruit\_BluefruitLE\_nRF51
7
8 //include header
9 #include <Adafruit_SSD1306.h>
10
11 //edit me via bandura - CDT
```
- Ausgabefeld:** Shows the message "Kompilieren abgeschlossen." (Compile finished).
- Schittstelle:** Shows the message "Der Sketch verwendet 39784 Bytes (15%) des Programmspeicherplatzes. Das ist viel!". Below it, the serial port information is shown: "1" and "Adafruit Feather M0, Arduino, Off auf /dev/ttyACM0".

Menüleiste

Werkzeuge

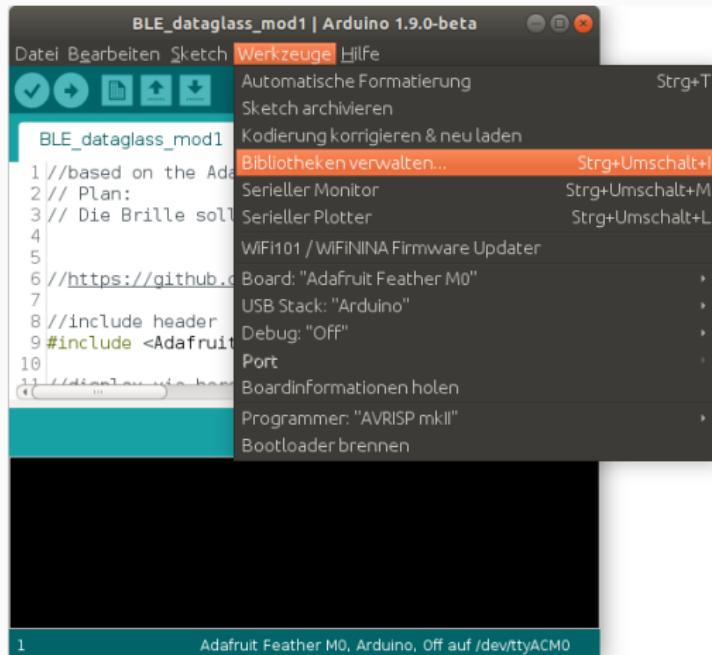
Textfeld
(Programm)

Ausgabefeld

Schittstelle

Microcontroller und Arduino iii

Konfigurieren: Bibliothek adafruit_ssdl1306 hinzufügen



The screenshot shows the Arduino IDE interface. The title bar reads "BLE_dataglass_mod1 | Arduino 1.9.0-beta". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge" (which is highlighted in orange), and "Hilfe". A context menu is open under "Werkzeuge", listing options like "Automatische Formatierung", "Sketch archivieren", "Kodierung korrigieren & neu laden", "Bibliotheken verwalten...", "Serieller Monitor", "Serieller Plotter", "WIFI101 / WIFININA Firmware Updater", "Board: 'Adafruit Feather M0'", "USB Stack: 'Arduino'", "Debug: 'OFF'", "Port", "Boardinformationen holen", "Programmer: 'AVRISP mkII'", and "Bootloader brennen". The "Bibliotheken verwalten..." option is currently selected. The main code editor window contains the following sketch code:

```
1 //based on the Ada
2 // Plan:
3 // Die Brille soll
4
5
6 //https://github.com
7
8 //include header
9 #include <Adafruit
10
```

The status bar at the bottom left shows "1" and "Adafruit Feather M0, Arduino, Off auf /dev/ttyACM0".

Menüleiste

Werkzeuge

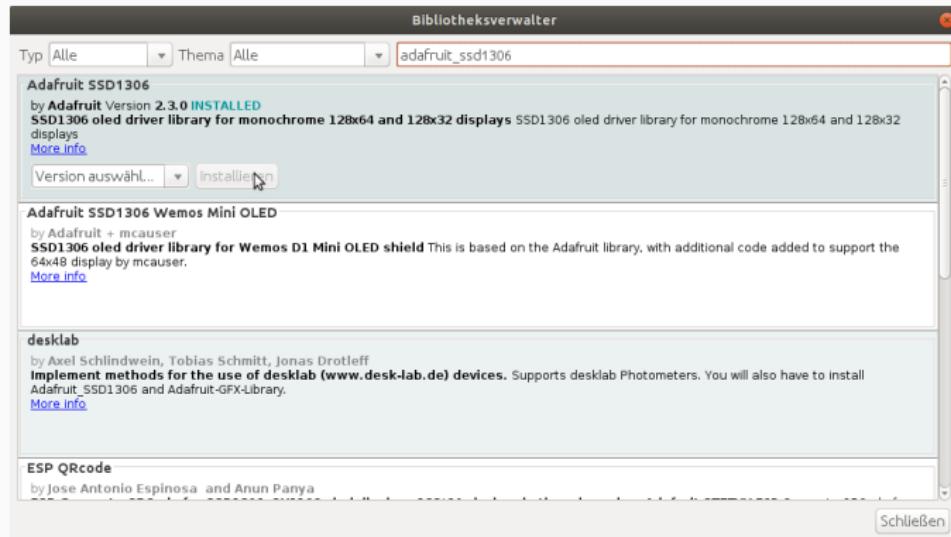
Textfeld
(Programm)

Ausgabefeld

Schittstelle

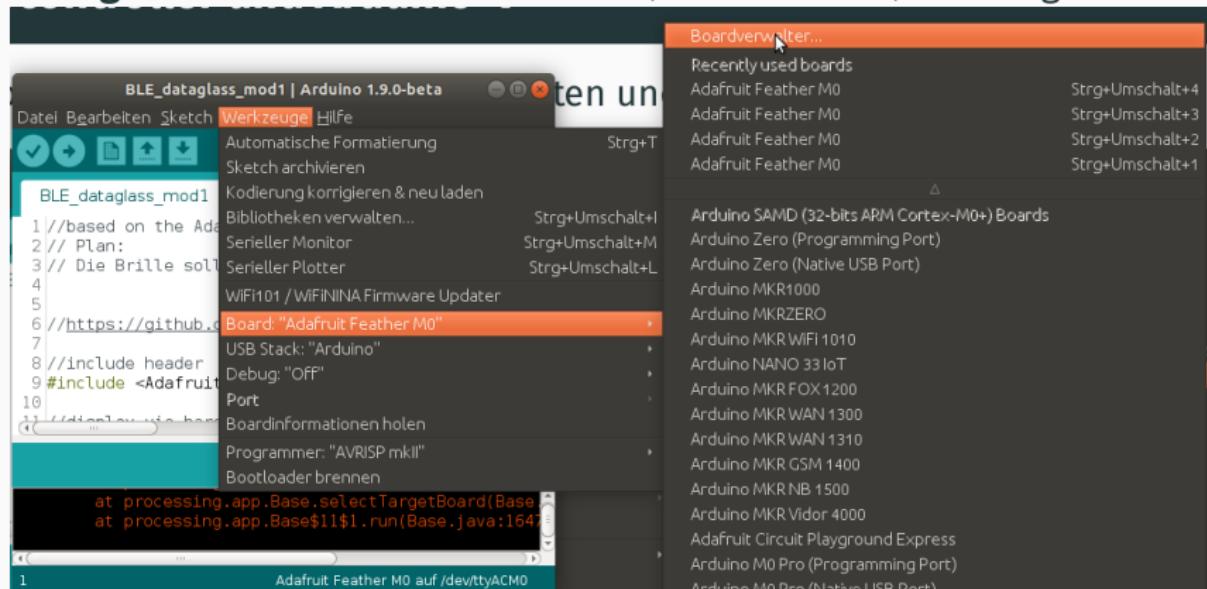
Microcontroller und Arduino iv

Konfigurieren: Bibliothek adafruit_ssdl306 hinzufügen
Suche adafruit_ssdl306 → aktuelle Version installieren



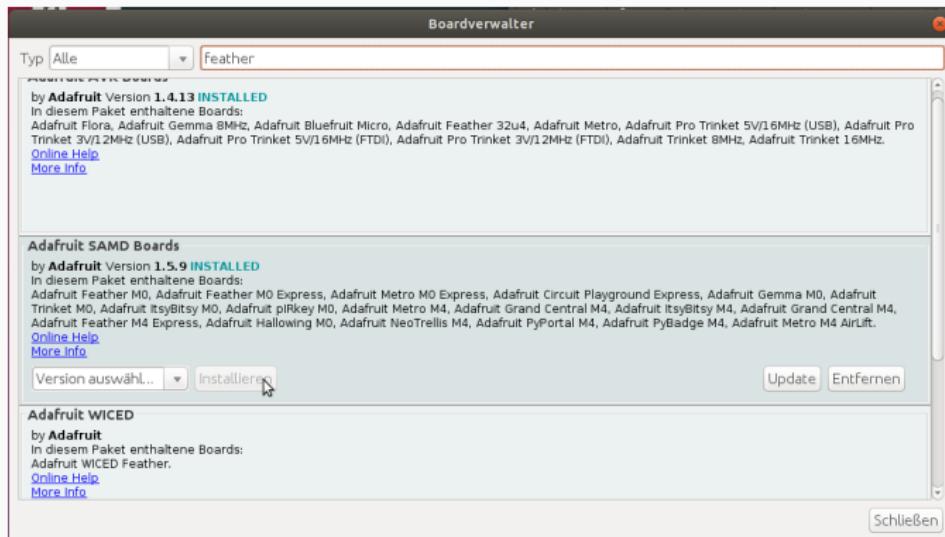
Microcontroller und Arduino v

Konfigurieren: Bibliothek für Feather (arduino_samd) hinzufügen



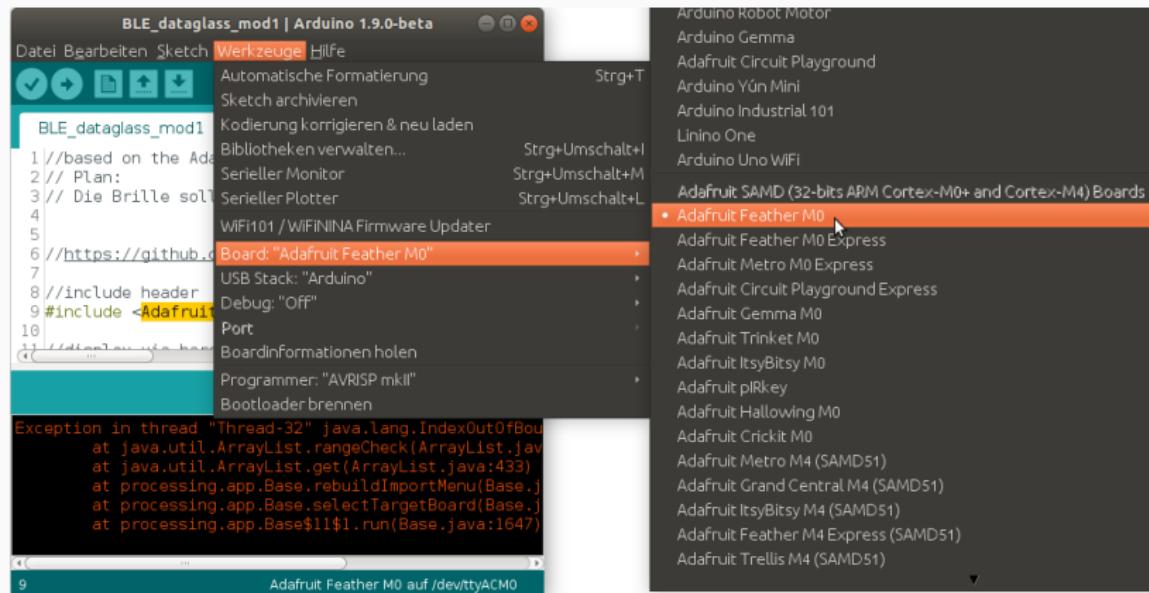
Microcontroller und Arduino vi

Konfigurieren: Bibliothek für Feather (arduino_samd) hinzufügen
Suche feather oder arduino_samd → aktuelle Version installieren



Microcontroller und Arduino vii

Konfigurieren: Richtiges Board einstellen



Spätestens danach steht als Schnittstelle "Adafruit Feather Mo"

Microcontroller und Arduino viii

Code überprüfen: (kompilieren, engl. *to compile*: zusammensetzen)

The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** BLE_dataglass_mod1 | Arduino 1.9.0-beta
- Menu Bar:** Datei Bearbeiten Sketch Werkzeuge Hilfe
- Toolbar:** Includes icons for Save, Open, Upload, Download, and Check.
- Sketch Name:** BLE_dataglass_mod1
- Code Area:** Displays the following C++ code:

```
1 //based on the Adafruit BluefruitLE library (SPI)
2 // Plan:
3 // Die Brille soll Daten wie zum Beispiel die Temperatur anzeigen
4 //
5 //
6 //https://github.com/adafruit/Adafruit_BluefruitLE
7 //
8 //include header
9 #include <Adafruit_SSD1306.h>
10 //
```
- Status Bar:** Komplizieren abgeschlossen.
- Bottom Status:** Der Sketch verwendet 39784 Bytes (15%) des Programmspeichers.
- Serial Monitor:** Shows the connection information: Adafruit Feather M0, Arduino, Off auf /dev/ttyACM0.

Menüleiste

Werkzeuge

Textfeld (Programm)

Ausgabefeld

Schittstelle

Kompilieren: Übersetzen des Programms in Maschinencode

Microcontroller und Arduino ix

Fehler nach kompilieren:

The screenshot shows the Arduino IDE interface with the title bar "BLE_dataglass_mod1 | Arduino 1.9.0-beta". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". The toolbar has icons for back, forward, search, and file operations. The code editor window contains the following code:

```
17 Adafruit_SSD1306 display(OLED_DC, OLED_RESET, OLED_CS);
18 uint8_t displaybuffer[DISPLAY_X*DISPLAY_Y/8]; //linear binary buffer
19
20
21 int i;
22 int z;// Zahler
23 int a=1; //Schrittweite
24 int b=10; //Pause
25 int c=5; //Kreisabstand
26 int d=200; //Max Anzahl Kreise
27 //int e=-0; //Startwert Kreise
28 unsigned long time;
29
30 #if (SSD1306_LCDHEIGHT != 64)
31 #error("Height incorrect, please fix Adafruit_SSD1306.h!");
32
33 #endif
```

A red error highlight covers the line "#error("Height incorrect, please fix Adafruit_SSD1306.h!");". Below the code editor, the "Fehlermeldungen kopieren" (Copy Error Messages) button is visible. The bottom status bar shows "core/core.a(main.cpp.o): In function `main':". The terminal window displays the following error messages:

```
BLE_dataglass_mod1:31:2: error: #error ("Height incorrect, please fix Adafruit_SSD1306.h")
^~~~~
core/core.a(main.cpp.o): In function `main':
Mehrere Bibliotheken wurden für "Adafruit_SSD1306.h" gefunden
Benutzt: /home/georg/Arduino/libraries/Adafruit_SSD1306
Mehrere Bibliotheken wurden für "Adafruit_GFX.h" gefunden
Benutzt: /home/georg/Arduino/libraries/Adafruit_GFX_Library
Mehrere Bibliotheken wurden für "SPI.h" gefunden
Benutzt: /home/georg/Arduino/libraries/WiFi/libraries/SPI.h
```

The terminal window also shows the build progress: "31" and "Adafruit Feather M0 auf /dev/ttyACM0".

- Ausgabefeld: rote Meldungen
- Textfeld: Zeile 31 mit Fehler (Bug) rot markiert
- Hier: die Fehlermeldung wurde vom Programmierer eingebaut

Microcontroller und Arduino x

Fehler nach kompilieren: Falsche Parameter in adafruit_ssdl306

Problembehebung: Öffne Sketchbook-Speicherort → Arduino/
libraries/Adafruit_SSD1306/adafruit_ssdl306.h und ersetze

adafruit_ssdl306.h

```
//#define SSD1306_128_64 //;< DEPRECATED: ...
#define SSD1306_128_32 //;< DEPRECATED: ...
```

durch

adafruit_ssdl306.h

```
#define SSD1306_128_64 //;< DEPRECATED: ...
//#define SSD1306_128_32 //;< DEPRECATED: ...
```

Microcontroller und Arduino xi

Hochladen: Code kompilieren und auf Board überspielen

The screenshot shows the Arduino IDE interface with the following components:

- Menüleiste:** Contains File, Bearbeiten, Sketch, Werkzeuge, and Hilfe.
- Werkzeuge:** Includes icons for upload, download, and search.
- Textfeld (Programm):** Displays the C++ code for the sketch. The code includes comments about using the Adafruit BluefruitLE library via SPI, defining a plan, and including the Adafruit SSD1306 header. It also includes a hardware SPI display configuration.
- Ausgabefeld:** A large black area at the bottom of the interface.
- Schittstelle:** Shows the board name "Adafruit Feather M0" and the port "/dev/ttyACM0".

```
//based on the Adafruit BluefruitLE library (SPI)
// Plan:
// Die Brille soll Daten wie zum Beispiel die Temperatur anzeigen
// https://github.com/adafruit/Adafruit_BluefruitLE
//include header
#include <Adafruit_SSD1306.h>
//display via hardware SPI
```

Menüleiste

Werkzeuge

Textfeld (Programm)

Ausgabefeld

Schittstelle

Code: Programmbausteine i

// Kommentar

```
// Alles was in einer Zeile nach // folgt ist ein Kommentar. Ein Kommentar ist vom Kompilieren ausgenommen.
```

Compiler-Optionen

```
// Compiler-Optionen erkennt man am vorgestellten #
// Bsp.: Bibliothek einbinden mit #include
#include <adafruit_ssdl1306.h>
// Bsp.: Variable OLED_DC auf Wert 5 setzen mit #define
#define OLED_DC 5
```

Code: Programmbausteine ii

Zuweisung

```
// Deklaration der Ganzzahl i und Zuweisung des Werts  
int i; // "Es gibt die Variable i mit Datentyp int".  
i = 1; // int hat den Wert 1  
  
// Beide Schritte in einer Zeile geht auch:  
int i=1;  
  
// Am Ende einer Zuweisung muss ein ; stehen. Danach kommt  
// der nächste Befehl.
```

Code: Programmbausteine iii

Objekte mit Methode ansprechen

```
// Ein Objekt hat verschiedene Eigenschaften, die ange-
// sprochen werden können

// Syntax: Objekt.Methode(Argumente)

display.begin(SSD1306_SWITCHCAPVCC); // Display starten
display.setTextColor(WHITE); // Textfarbe auf weiß setzen
```

Weitere Methoden von display: <https://learn.adafruit.com/adafruit-gfx-graphics-library/graphics-primitives>

Text auf Display darstellen mit display.print() i

display.ino

```
// Statischen Text auf Display ausgeben:  
void Targeter() {  
    display.setTextColor(WHITE);      // Textfarbe  
    display.setCursor(30, 10);        // Cursorposition  
    display.setTextSize(2);          // Textgröße  
    display.setTextWrap(0);          // kein Zeilenumbruch  
    display.clearDisplay();         // Display löschen  
    display.print("Hallo Georg!");   // Textausgabe definieren  
    display.display();              // Display aktualisieren  
    delay(500);                   // Pause für 500 ms = 0.5 s  
}
```

Text auf Display darstellen mit `display.print()` ii

display.ino

```
// statt display.print("Hallo Georg!") ab jetzt Folgendes:  
  
display.print("Zeit="); // statische Textausgabe  
time = millis()/1000; // Zeit in s ab Start berechnen  
display.print(time); // Zeit auf Display werfen
```

Achtung: die Variable `time` gibt es noch nicht. Sie muss zuerst in der Hauptfunktion deklariert werden:

BLE_dataglass_mod1.ino

```
unsigned long time;
```

Kreis auf Display darstellen i

display.ino

```
x = (display.width()-50)/2;  
y = (display.height()-50)/2;  
r = 50;  
display.drawCircle(x, y, r, WHITE);
```

Deklarieren (int) nicht vergessen!

Kreis auf Display darstellen ii

display.ino

```
for (r_i = 0; r_i <= 10; r_i = r_i + 2) {  
    display.clearDisplay();  
    display.drawCircle(x, y, r_i, WHITE);  
    display.display();  
    delay(500);  
}
```

In einer `for`-Schleife werde die beschriebenen Operationen für jeden Wert der Schleifenvariable `r_i` ausgeführt. Der Startwert ist 0, der Endwert 10, die Schrittweite 2. Also nimmt der Radius die Werte `r_i=0,2,4,6,8,10` an, für den jeweils ein Kreis gezeichnet wird.

Anregungen für Erweiterungen

- Lass die Kreise/Rechtecke von außen nach innen laufen!
- Lass den Mittelpunkt der Kreise wandern!
- Recherchiere im Internet, welche weiteren geometrischen Elemente darstellbar sind, und teste sie!