

# Verbindung schaffen - I



**Kurze Begründung:** Die Arduino IDE arbeitet mit einer C sehr nahen Sprache. In vielen Schulen kommt diese bereits im Fach Technik zum Einsatz. Es macht durchaus Sinn, zur Verknüpfung der beiden Fächer Technik und IMP auf eine gleiche Basis zu setzen. C wird oft nicht für den Einstieg in die Programmierung empfohlen, da diese Sprache zwar einfach aber auch durch sehr viele Bibliotheken, die selbst wieder Wissen brauchen, sehr stark erweiterbar und komplex ist.

Im Folgenden findet sich eine Anleitung, wie der BBC Microbit mit der Arduino IDE programmiert werden kann. Es kann sein, dass man sehr viel Ausdauer und Geduld braucht, bis der erste Sketch glatt durchläuft.

## Ablauf der Installation:

1. Installation der Arduino IDE  
<https://www.arduino.cc/en/Main/Software>
2. Den Board-Manager hinzufügen  
<https://github.com/sandeepmistry/arduino-nRF5>
3. Bibliotheken importieren  
direkt in der Arduino IDE



## Arduino

Das erste Board wurde 2005 von Massimo Banzi und David Cuartielles entwickelt. Der Name „Arduino“ wurde von einer Bar in Ivrea übernommen.



- ① Arduino IDE
  - Laden Sie die Arduino IDE auf Ihren PC ca. 120 MB
  - <https://www.arduino.cc/en/Main/Software>



- ② Öffnen Sie die IDE und wechseln Sie unter Datei zu den Voreinstellungen.
  - kopieren Sie den folgenden Link und fügen Sie diesen im Feld „zusätzliche Boardverwalter URLs“ ein. Auch auf der Seite von Punkt 2. oben kopierbar.
  - [https://sandeepmistry.github.io/arduino-nRF5packages/nRF5boards\\_index.json](https://sandeepmistry.github.io/arduino-nRF5packages/nRF5boards_index.json)



- ③ Wechseln Sie dann in das Menü Werkzeuge und wählen Sie dort  
-> Board -> Boardverwalter und scrollen Sie bis zu  
**Nordic Semiconductor nRF5 Boards**
  - klicken Sie auf installieren
  - dieser Prozess könnte abbrechen - wiederholen sie diesen dann. (ca. 100MB)



[Video dazu...](#)

## Verbindung schaffen - II



Damit die Sketches, die hier gezeigt werden rund laufen, müssen nun noch mind. zwei Bibliotheken nachinstalliert werden.

Wenn dann beim Kompilieren der Compiler eine weitere Bibliothek braucht, sollte diese auch nachinstalliert werden.

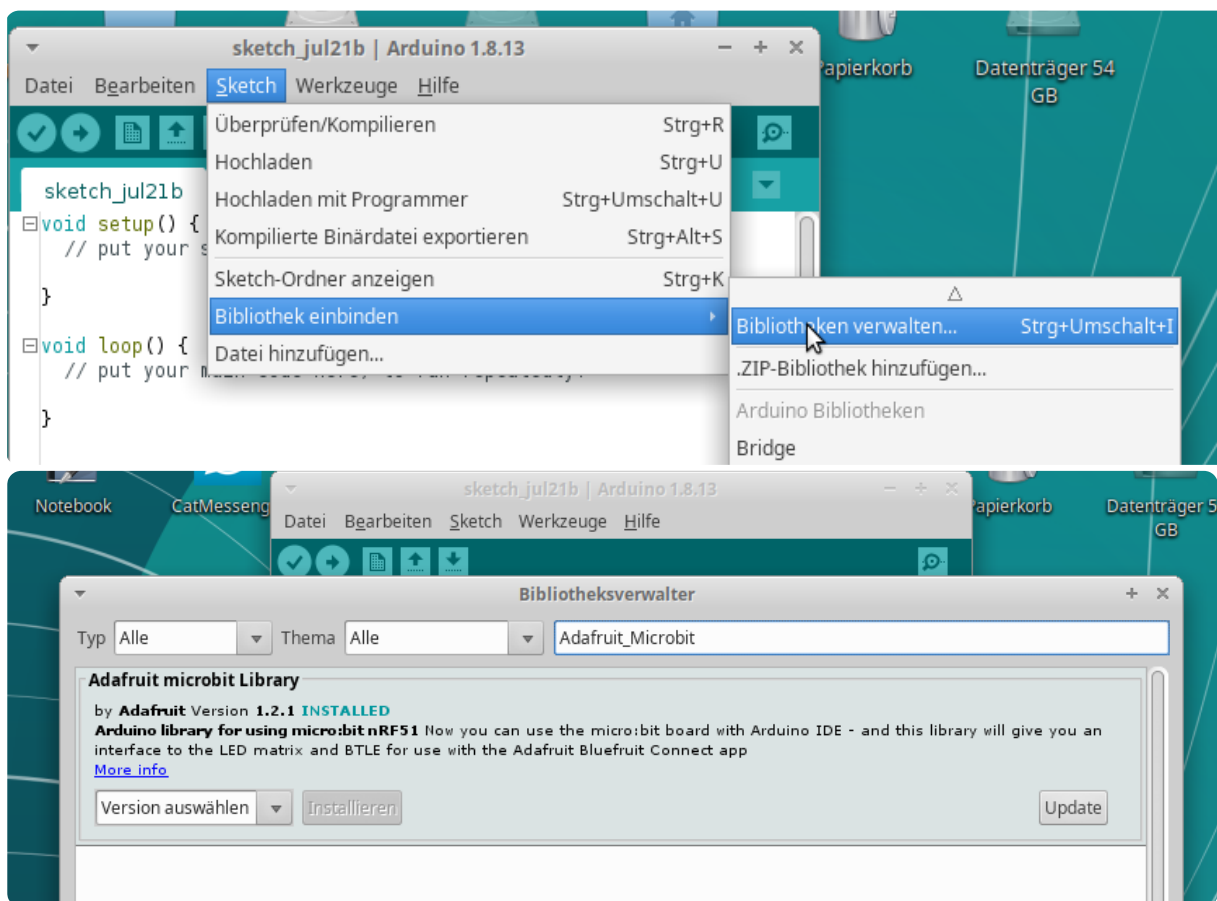
Diese Verbindung zur IDE soll zum einen zeigen, dass diese Entwicklungsumgebung mehr kann, als "nur" Arduinos und zum anderen soll hier die Verbindung zur Technik einfacher gestaltet werden. Microbits und Calliope Mini bieten eben schon ein fertiges System, welches nicht noch mit vielen Kabeln oder ähnlichem zu technisch werden. Es kann direkt losprogrammiert werden.



1. Bibliothek:  
**Adafruit\_Microbit**



2. Bibliothek:  
**BLE\_Peripherals**



**Bitte beide Bibliotheken installieren.**

## Verbindung schaffen - III



"Fast hätte ich es vergessen..."

Damit der Microbit sauber mit der IDE kommunizieren kann ist noch ein Treiber für embedded ARM Systeme notwendig.

Treiberdownload:

### Betriebssystem:

#### OS X

Nichts muss gemacht werden.

#### Linux

Für 64-bit Linux Benutzer:

```
libc6:i386, libstdc++6:i386,
libncurses5:i386
libudev1:i386
```


```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get -y install libc6:i386 libstdc++6:i386
libncurses5:i386 libudev1:i386
```


#### Windows


Driver Setup for mbed devices

Download mbed Windows Serial driver

<https://developer.mbed.org/handbook/Windows-serial-configuration#1-download-the-mbed-windows-serial-port>

 OS X

 Linux

 Win

### Probeketch:

```
C++
1 #include <Adafruit_Microbit.h>
2 Adafruit_Microbit_Matrix microbit;
3
4 void setup() {
5     microbit.begin();
6 }
7
8 void loop() {
9     microbit.fillScreen(LED_ON);
10    delay(1000);
11
12    microbit.fillScreen(LED_OFF);
13    delay(1000);
14 }
```