

Programm zum Programmieren

Programme für Mikrocontroller schreibt man in sogenannten Entwicklungsumgebungen (IDE = integrated development environment) auf einem Computer und überträgt sie dann z.B. per USB-Schnittstelle. Die Entwicklungsumgebung für die vielen verschiedenen Arduino-Mikrocontroller heißt „Arduino“ und sieht aus wie ein ganz normales Programm mit einigen Sonderfunktionen.

Überblick

In diesem großen Feld schreibt man das Programm, das beim Arduino auch als **Sketch** bezeichnet wird...

...und mit diesem Button kannst du es probeweise kompilieren (siehe links) und so auf Schreibfehler prüfen.

Mit einem Klick auf den Pfeil wird das Programm per USB auf den Arduino übertragen.

In vielen Programmiersprachen werden Programme in „Anweisungsblöcke“ strukturiert. Ein **Block** besteht hier aus einem Paar **geschweifeter Klammern** (AltGr+7 bzw. 0) sowie den Zeilen dazwischen.

The screenshot shows the Arduino IDE window titled 'sketch_feb03a | Arduino 1.6.7'. The menu bar includes 'Datei', 'Bearbeiten', 'Sketch', 'Werkzeuge', and 'Hilfe'. The toolbar contains icons for saving, running, and other functions. The main editor area shows the following code:

```

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here,
}

```


Callout boxes provide additional information:

- A grey box points to the code editor area, stating: 'In diesem großen Feld schreibt man das Programm, das beim Arduino auch als **Sketch** bezeichnet wird...'.
- A grey box points to the 'Verify' button (a checkmark icon), stating: '...und mit diesem Button kannst du es probeweise kompilieren (siehe links) und so auf Schreibfehler prüfen.'.
- A grey box points to the 'Upload' button (a right-pointing arrow icon), stating: 'Mit einem Klick auf den Pfeil wird das Programm per USB auf den Arduino übertragen.'.
- A grey box points to the bottom status bar, stating: 'Hier werden Hinweise zu auftretenden Fehlern angezeigt.'.
- A grey box points to the code editor, stating: 'In vielen Programmiersprachen werden Programme in „Anweisungsblöcke“ strukturiert. Ein **Block** besteht hier aus einem Paar **geschweifeter Klammern** (AltGr+7 bzw. 0) sowie den Zeilen dazwischen.'
- A grey box points to the code editor, stating: 'Zwei Anweisungsblöcke sind vorgeschrieben: Die Anweisungen im Block **setup(){...}** werden direkt ausgeführt, sobald man den Arduino startet. Danach werden die Anweisungen im Block **loop(){...}** ausgeführt und immer wieder wiederholt...Das nennt man auch „Endlosschleife“.'
- A grey box points to the code editor, stating: 'Alles was in einer Zeile hinter zwei Schrägstrichen („/“, **slash**) steht, gilt als **Kommentarzeile** und wird vom μ C ignoriert. Dieses Programm enthält also keine einzige echte Anweisung, sondern nur zwei leere Blöcke.'

Zwei Anweisungsblöcke sind vorgeschrieben: Die Anweisungen im Block **setup(){...}** werden direkt ausgeführt, sobald man den Arduino startet. Danach werden die Anweisungen im Block **loop(){...}** ausgeführt und immer wieder wiederholt...Das nennt man auch „Endlosschleife“.

Alles was in einer Zeile hinter zwei Schrägstrichen („/“, **slash**) steht, gilt als **Kommentarzeile** und wird vom μ C ignoriert. Dieses Programm enthält also keine einzige echte Anweisung, sondern nur zwei leere Blöcke.

Der grundlegende Aufbau jedes Arduino-Programms ist also **immer** so:



```
bsp | Arduino 1:1.0.5+dfsg2-2
Datei Bearbeiten Sketch Tools Hilfe
bsp 5
/*
  Das ist ein
  mehrzeiliger Kommentar
  der von Kommentarstart- und
  Kommentarende-Markierungen begrenzt wird
*/
// Das ist ein einzeliger Kommentar
int led=13; // das ist eine Anweisung, die etwas macht, was später in beiden Blöcken
// the setup routine runs once when you press reset:
void setup() {
  // Hier kommen die Anweisungen für den setup-Block hinein
  // der setup Block wird einmal durchlaufen
}
// the loop routine runs over and over again forever:
void loop() {
  // Hier kommen die Anweisungen für den loop-Block hinein
  // der loop-Block wird unendlich oft durchlaufen
}

Kompilierung abgeschlossen.

Binäre Sketchgröße: 466 Bytes (von einem Maximum von 32.256 Bytes)

21 Arduino Uno on /dev/ttyACM0
```

The screenshot shows the Arduino IDE interface with a sketch named 'bsp 5'. The code is divided into three highlighted regions with red borders and red text labels on the right side:

- Globaler Bereich 1x**: This region contains the multi-line comment and the variable declaration `int led=13;`.
- setup-Bereich 1x**: This region contains the `void setup()` function, which is executed once when the board is reset.
- loop-Bereich unendlich oft**: This region contains the `void loop()` function, which is executed repeatedly in a loop.

Below the code editor, the IDE shows the compilation status: 'Kompilierung abgeschlossen.' and the binary size: 'Binäre Sketchgröße: 466 Bytes (von einem Maximum von 32.256 Bytes)'. The bottom status bar indicates '21 Arduino Uno on /dev/ttyACM0'.

Aufgabe 3.1 - Erster Kontakt

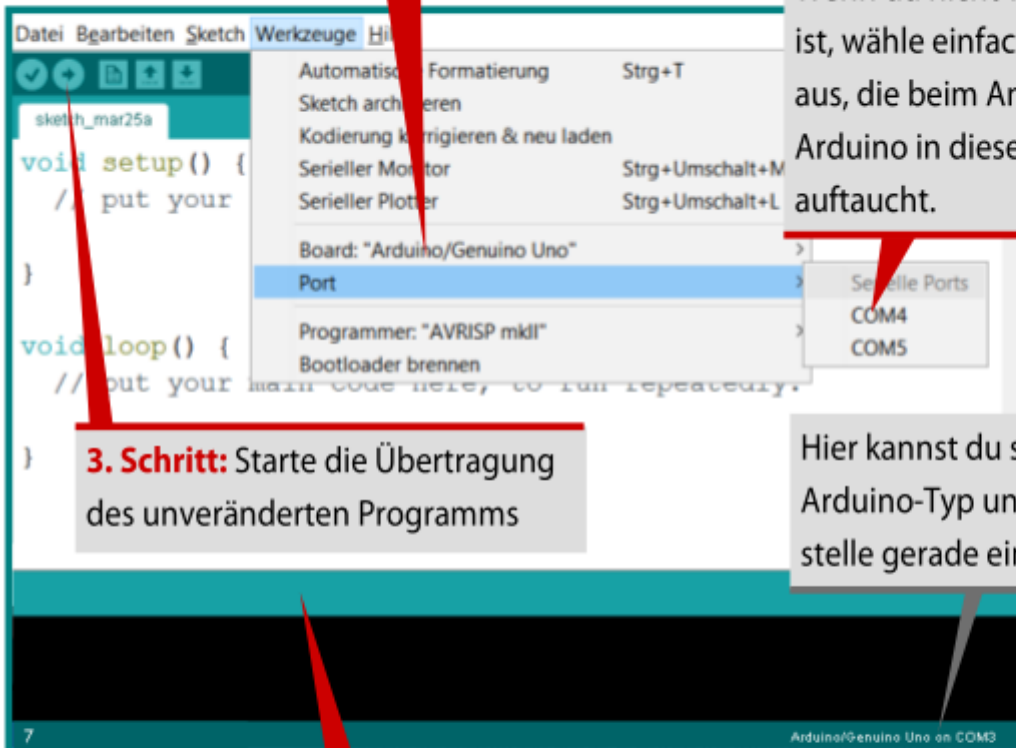
In dieser Aufgabe lernst du, wie das unveränderte „leere“ Programm auf den Arduino übertragen wird. Da die Entwicklungsumgebung für viele verschiedene Arduino-Boards geeignet ist, die an allen möglichen Schnittstellen des Computers angeschlossen werden.

Deshalb sind erst einmal zwei Einstellungen notwendig¹⁾:

1. Einstellung: Wähle hier den richtigen Arduino-Typ aus. Dein „Board“ heißt „Arduino Uno“.

2. Einstellung: Wähle die richtige Schnittstelle (engl. Port) aus.

Wenn du nicht weißt, welche das ist, wähle einfach die Schnittstelle aus, die beim Anschließen des Arduino in dieser Auflistung neu auftaucht.



3. Schritt: Starte die Übertragung des unveränderten Programms

Hier kannst du sehen, welcher Arduino-Typ und welche Schnittstelle gerade eingestellt sind.

Nach der erfolgreichen Übertragung steht in der Statuszeile so etwas wie „Upload done“, „Upload abgeschlossen“ oder „Hochladen abgeschlossen“. Wenn es nicht funktionierte, steht hier **„Upload failed“** oder **„Probleme beim Hochladen“**. Dann solltest du die Einstellungen noch einmal prüfen.

[ardulb1](#)

1)

In der Schule stimmen die meistens schon, weil der Administrator das passend eingestellt hat, auf deinem eigenen Computer musst du dich da selber drum kümmern

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
<https://wiki.qg-moessingen.de/faecher:nwt:arduino:lernbaustein1:ide:start>

Last update: **14.09.2020 19:09**

