



Mit Blöcken Hardware programmieren

Future Learning Lab Wien

Christian Pollek

Herbst 2020, Wien





Inhalt

Vorwort	3
Mit Blöcken Hardware programmieren	3
Kurzbeschreibung	4
Anleitung:	5
Einfach und hübsch – mblock	5
Start	5
Gerät auswählen	6
Warum gerade Arduino Uno?	6
Modus einstellen	7
Programm bauen	8
Programmcode anzeigen	10
Code in die <i>Arduino IDE</i> kopieren	11
Board auswählen	12
Übertragen auf den Microcontroller	13
Mächtig aber nicht so „hübsch“ - ArduBlock	14
Installation	14
Start von ArduBlock	15
Programm bauen	16
Blink-Programm	17
Projekt auf das Board übertragen	18
Expert*innenmodus	19





Vorwort:

Diese Anleitung ist kein Einstieg in das Arbeiten mit Mikrocontrollern. Sie setzt zumindest für die Lehrperson Grundlagen im Umgang mit Mikrocontrollern voraus. Wir empfehlen Ihnen folgende Anleitung für den Einstieg:

<https://funduino.de/anleitung>

Mit Blöcken Hardware programmieren

Schulstufe: 5.-8. Schulstufe, eventuell 4. Klasse
Volksschule

Dauer: Je nach Vorkenntnissen bis zu einer Stunde

Verwendete Geräte und Werkzeuge:

- Microcontroller (z.B. Arduino Uno, Microbit, Elegoo Uno, ...)
- Eingabegerät (Computer, Laptop, ...)
- passendes Verbindungskabel um den Microcontroller mit dem Eingabegerät zu verbinden.

Materialbedarf: kein zusätzliches Material benötigt



Kurzbeschreibung

Wussten Sie, dass auch mit *Arduinos* (und Microcontrollern, die mit der *Arduino IDE* programmiert werden können) mithilfe von Blöcken wie in [Scratch](#) und [Snap!](#) programmiert werden kann?

Wir stellen Ihnen zwei Programme vor, mit denen Sie sich dem Programmieren von Microcontrollern mit Blöcken annähern können.

Voraussetzungen:

- Pro Platz: ein PC mit Internetzugang
- [Arduino IDE](#) installiert
- ein Board/Microcontroller, welcher mit dieser programmiert werden kann (zb Arduino,ESP32,Attiny)
- LEDs, Knöpfe, Widerstände, Sensoren, Breadboard, Steckbrücken, eventuell Krokodilklemmen





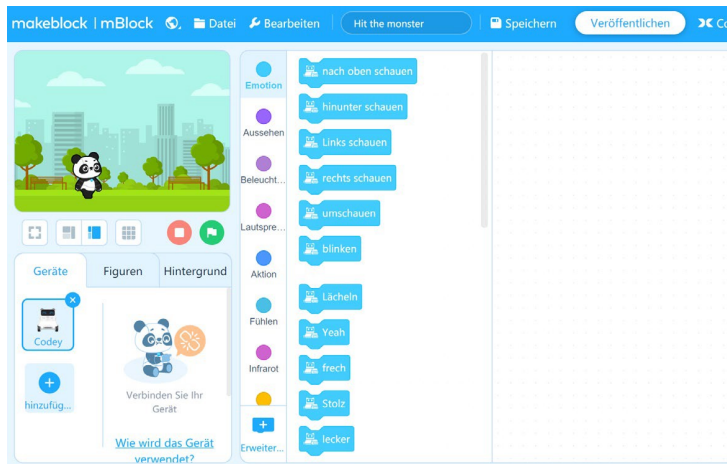
Anleitung:

Einfach und hübsch – mblock

Website: <https://ide.mblock.cc/#/>

Start

Die Website aufrufen und das Popup zu *mLink* ignorieren („Später“ wählen) um zu folgender Anwendung zu gelangen:



Dieses Werk ist unter einer Creative Commons Lizenz vom Typ Namensnennung-Nicht kommerziell 3.0 Österreich zugänglich. Um eine Kopie dieser Lizenz einzusehen, konsultieren Sie <http://creativecommons.org/licenses/by-nc/3.0/at/>.

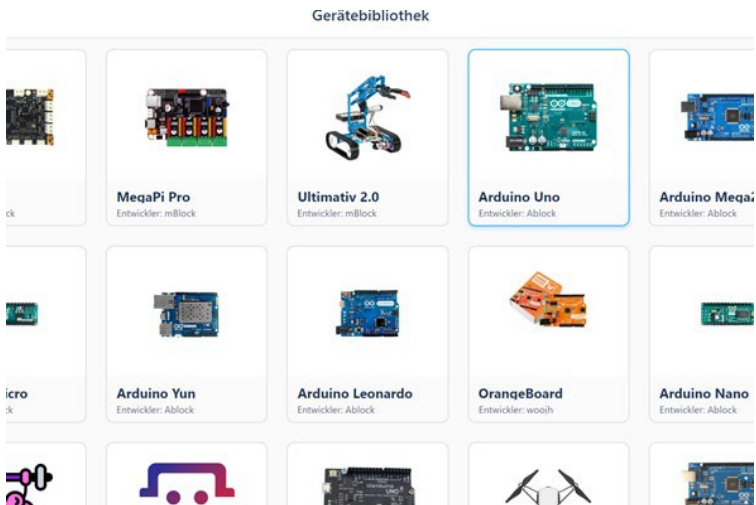


Gerät auswählen

Unten links auf den Tab „Geräte“ klicken und mit „Hinzufügen“ das passende Board auswählen. Ich wähle den *Arduino Uno* aus und drücke OK.

Warum gerade Arduino Uno?

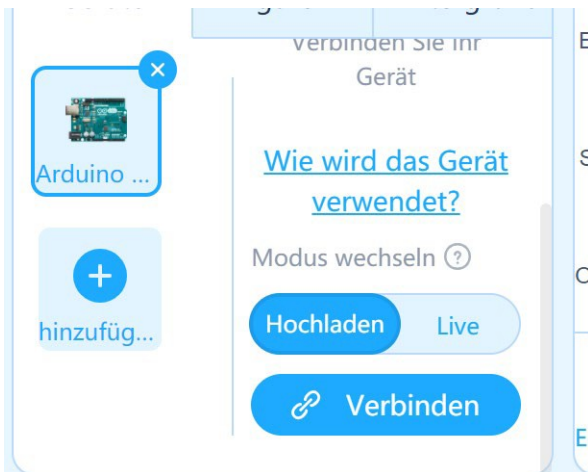
Da ich den fertigen Code mithilfe der *Arduino IDE* übertragen werde, muss die Zielhardware nicht unbedingt ein *Arduino Uno* sein, sondern nur mit der *Arduino IDE* kompatibel.





Modus einstellen

Unter „Wie wird das Gerät verwendet?“ den Modus bei Geräte auf „Hochladen“ stellen.



Dies hat zur Folge dass *mblock* aus meinem Blockprogramm *C Code* generiert der dann später in die *Arduino IDE* übertragen und auf den Microcontroller übertragen werden kann.





Programm bauen

Durch Drag&Drop wird das Programm gebaut, das den Microcontroller steuern soll, aber nicht aus Text, sondern aus Blöcken. Unter „Ereignisse“ kann man den „Wenn Arduino Uno startet“ Block finden und ihn rechts in das freie Feld ziehen. Dieses Feld passt seinen Namen an das gewählte Gerät an.

In der Grafik sehen wir, wie ein einfaches „Blink“-Programm aufgebaut wird. Es sagt dem Microcontroller später:



1. Wiederhole immer wieder folgende Schritte (wiederhole fortlaufend):
2. Die Versorgung mit Spannung (üblicherweise 3 oder 5V) durch den Ausgang 2 fließen lassen (digitalen Pin von Ausgang 2 als hoch setzen)
3. Diesen Zustand eine Sekunde lang beibehalten (warte 1 Sekunden)
4. Die Versorgung mit Spannung (üblicherweise 3 oder 5V) durch den Ausgang 2 stoppen (digitalen Pin von Ausgang 2 als niedrig setzen)
5. Diesen Zustand eine Sekunde lang beibehalten (warte 1 Sekunden)





„Wiederhole fortlaufend“ ist ein sogenannter Loop und unter „Steuerung“ zu finden. Er wiederholt alles, was sich in ihm befindet unaufhörlich.

Die Blöcke für die Ansteuerung der Pins sind unter „Pin“ zu finden und der „Warte“-Befehl findet sich wieder unter „Steuerung“

Damit dieses Programm später auf dem Microcontroller funktioniert, muss eine LED (mit passendem Vorwiderstand) an den Ausgang „2“ angeschlossen sein. Um es zu testen, kann das schnell auf einem Breadboard mithilfe von Steckbrücken und/oder Krokodilklemmen aufgebaut werden. Hierbei ist es wichtig, die Gesetzmäßigkeiten eines elektrischen Stromkreises zu beachten.



Programmcode anzeigen

Damit das Programm den hinter den Blöcken in „C-Code“ geschriebenen Code anzeigt, kann das „</> Symbol“ rechts oben angeklickt werden.



Hier sehen wir den in der Programmiersprache C geschriebenen Code, der von der *Arduino IDE* in den Maschinencode für den Microcontroller umgewandelt (kompiliert) und von diesem ausgeführt werden kann.

```
Arduino C
1 // generated by mBlock5 for <your product>
2 // codes make you happy
3
4 #include <Arduino.h>
5 #include <Wire.h>
6 #include <SoftwareSerial.h>
7
8 void _delay(float seconds) {
9     long endTime = millis() + seconds * 1000;
10    while(millis() < endTime) _loop();
11 }
12
13 void setup() {
14     pinMode(2,OUTPUT);
15     while(1) {
16         digitalWrite(2,1);
17         _delay(1);
18         digitalWrite(2,0);
19         _delay(1);
20     }
21     _loop();
22 }
23
24 }
25
26 void _loop() {
```





Code in die *Arduino IDE* kopieren

Der gesamte Code wird nun ausgewählt (entweder mit der Maus ein Feld über den ganzen Text ziehen oder hinein klicken und Strg+A auf der Tastatur drücken) und kopiert (Rechtsklick+„kopieren“ oder Strg+C auf der Tastatur drücken).

Als nächstes wird die *Arduino IDE* am Eingabegerät gestartet und der kopierte Code eingefügt (Rechtsklick+„Einfügen“ oder Strg+V auf der Tastatur drücken).

Unter „File“ & „Save“ oder „Datei“ & „Speichern“ kann das geschriebene Programm nun auf der Festplatte gespeichert werden.

Hier finden Sie Informationen zur [Installation der *Arduino IDE*](#).

```
sketch_oct23a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_oct23a $
#include <Wire.h>
#include <SoftwareSerial.h>

void _delay(float seconds) {
    long endTime = millis() + seconds * 1000;
    while(millis() < endTime) _loop();
}

void setup() {
    pinMode(2,OUTPUT);
    while(1) {
        digitalWrite(2,1);
        _delay(1);
        digitalWrite(2,0);
        _delay(1);

        _loop();
    }
}

void _loop() {
}

void loop() {
    _loop();
}
```



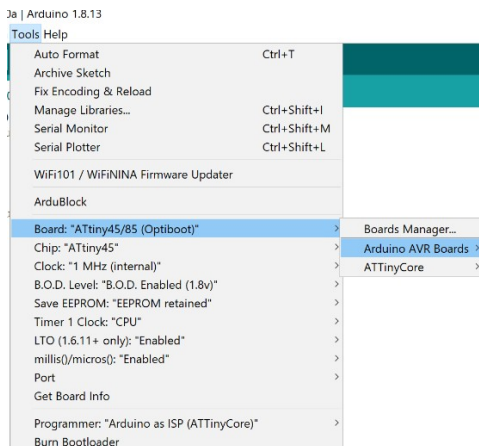


Board auswählen

Der verwendete Microcontroller muss in der Arduino IDE installiert sein. Bei allen Microcontrollern aus der Arduino Familie ist das bereits der Fall. Bei anderen muss dies eventuell nachgeholt werden.

Hier eine Anleitung für die Installation des Microcontrollers [ESP32](#) oder für einen „nackten“ Microcontroller, den [ATiny45](#) (und andere aus der Atiny Familie, für Fortgeschrittene und auf Englisch)

Unter Werkzeuge/Board kann das angeschlossene Board ausgewählt werden.

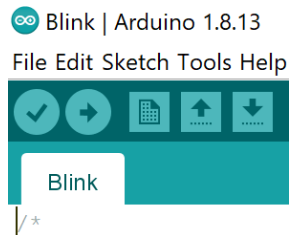




Übertragen auf den Microcontroller

Zum Übertragen des Codes auf den Microcontroller muss das Board mit einem passenden Kabel an den PC angeschlossen werden. Unter „Tools/Port“ kann der Anschluss am Eingabegerät ausgewählt werden, an dem es angeschlossen ist.

Daraufhin kann durch Klick auf das „Haken“-Symbol ganz links den Code auf den Microcontroller übertragen werden.



Ist der Microcontroller nun mit dem auf dem Breadboard vorgesteckten Stromkreis verbunden, so blinkt die LED. Dieses „Blink“ Programm nennt man „Hello World“, weil es ein Testprogramm ist um zu sehen, ob das Programm oder der Microcontroller funktioniert. Es ist wie die erste Kontaktaufnahme des Geräts mit der „Welt“.

Wenn der Microcontroller noch nicht an den Stromkreis angeschlossen ist, dann kann man dies gefahrlos machen. Der Microcontroller benötigt eine Form von Stromversorgung, damit das Programm funktioniert und kann hierfür entweder am Computer angeschlossen bleiben oder an einer anderen Stromquelle angeschlossen werden.





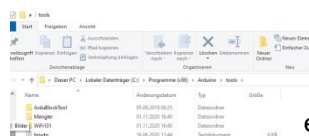
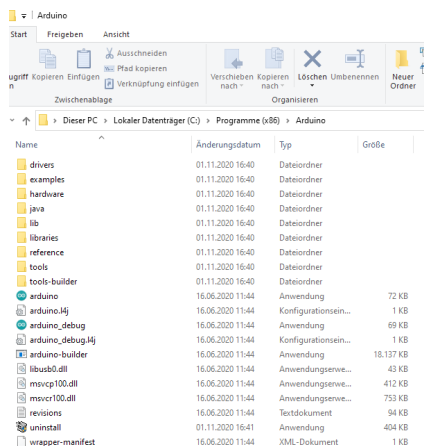
Mächtig aber nicht so „hübsch“ - ArduBlock

ArduBlock eignet sich für Fortgeschrittene, denn es verfügt auch über die Möglichkeit, [Funktionen](#) (Unterprogramme) zu definieren und innerhalb des Hauptprogramms auszuführen. *ArduBlock* ist eine Blocksoftware, die direkt in die *Arduino IDE* als „Tool“ eingebunden werden kann.

Die Namen der Blöcke entsprechen den Namen der verwendeten Codezeilen, folgende [Anleitung](#) kann zur Unterstützung herangezogen werden.

Installation

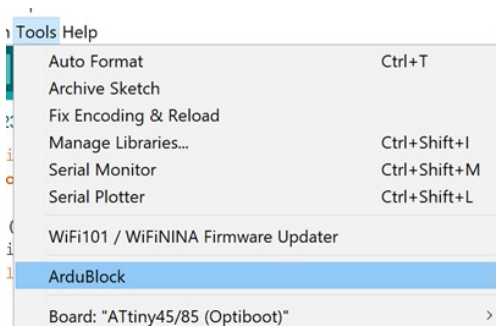
ArduBlock kann [hier](#) als [.ZIP-Datei](#) heruntergeladen werden. Daraufhin den ArduBlockTool-Ordner entpacken und in das Home-Verzeichnis der Arduino IDE in einen neu angelegten oder bereits vorhandenen „tools“-Ordner kopieren.



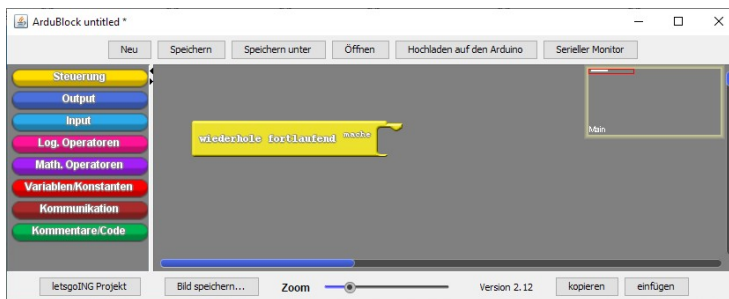


Start von ArduBlock

Nach dem Start der *Arduino IDE* kann nun unter „Tools“ oder „Werkzeuge“ *ArduBlock* in einem neuen Fenster geöffnet werden:



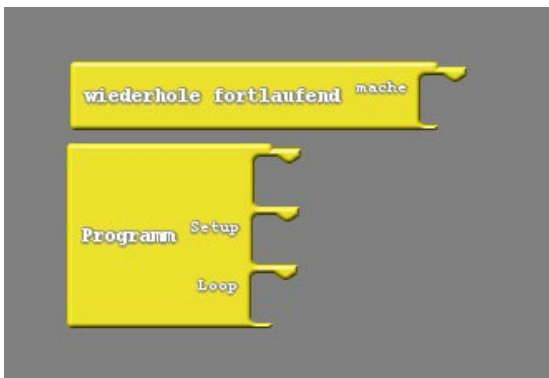
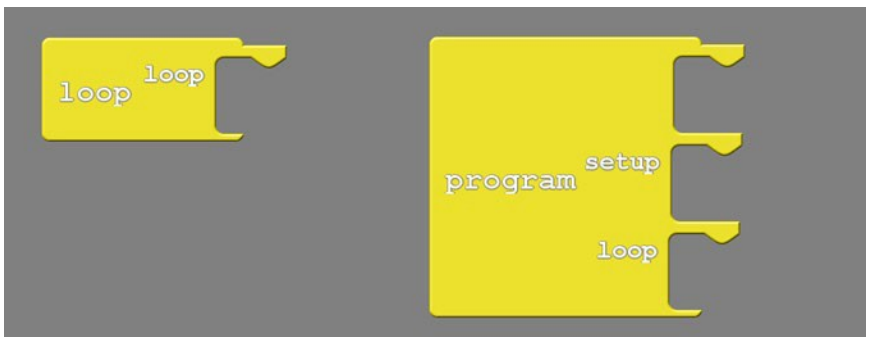
In der *ArduBlock* Oberfläche befinden sich links die Block-Kategorien. Die Blöcke können wie bei *mblock* einfach in das graue Feld rechts daneben gezogen werden.





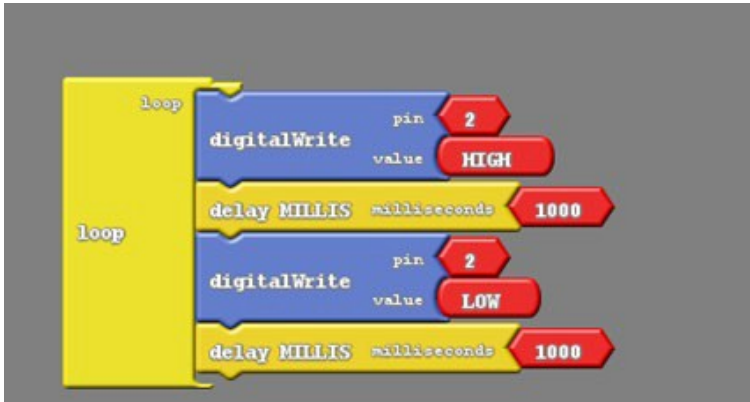
Programm bauen

Hier kann nun das Programm gebaut werden. Es gibt zwei Möglichkeiten zu beginnen - entweder mit dem „Loop“/„wiederhole fortlaufend“-Block oder mit dem „Programm“-Block. Beide befinden sich unter den gelben „Control“/„Steuerung“ Blöcken.





Blink-Programm



Diese Abbildungen zeigen wie bei der vorhergehenden Übung in *mblock* ein „Blink“- oder „Hello

World“ Programm. Eine genauere Anleitung zur Vertiefung ist auch [hier zu finden](#).



Im Vergleich zu *mblock* werden hier auch in der deutschen Version teilweise die englischen Programmierbegriffe dargestellt.

Das Programm kann unter „Speichern“ nun auch gespeichert werden und zu einem späteren Zeitpunkt weiterbearbeitet.

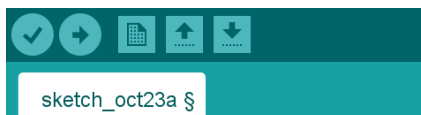


Projekt auf das Board übertragen

Mit dem Knopf „Upload“/„Hochladen auf den Arduino“ wird das Programm gespeichert. Danach öffnet es sich als Code in der *Arduino IDE*, welche sofort versucht, das Programm auf den Microcontroller zu übertragen.

Der Microcontroller muss hierfür an das Eingabegerät angeschlossen sein und in der Arduino IDE unter „Tools“/„Werkzeuge“ ausgewählt und angewählt.

Falls das Übertragen nicht geklappt hat, kann überprüft werden, ob unter „Tools“/„Werkzeuge“ das richtige Board eingestellt wurde.



```
void setup() {  
  pinMode(2, OUTPUT);  
}  
  
void loop() {  
  digitalWrite( 2 , HIGH );  
  delay( 1000 );  
  digitalWrite( 2 , LOW );  
  delay( 1000 );  
}
```



Expert*innenmodus

Zugang zu den „Expert*innen“-Blöcken kann rechts unten über den Button „Project Blocks“/“Projekt Blöcke“ freigeschaltet werden. Mit einem weiteren Klick auf denselben Knopf, der jetzt „Standard Blocks“/“Grundlagen Blöcke“ heißt, kann wieder in den „normalen“ Modus gewechselt werden.



Wir wünschen viel Freude beim Programmieren von Microcontrollern in der Schule!