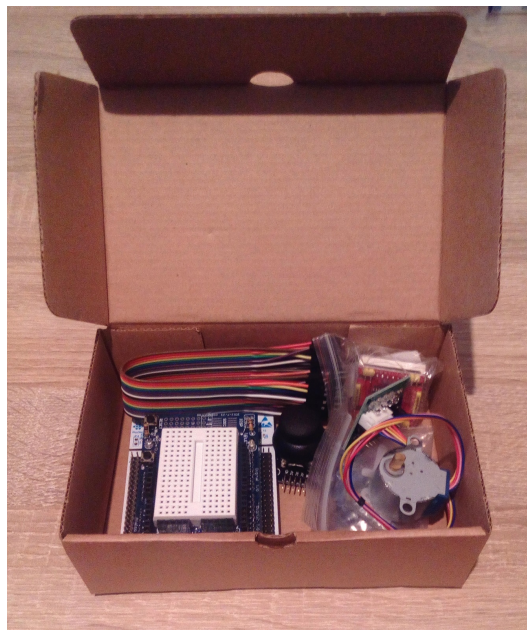


Hinweise & Vorbereitungen

Hinweise zum Kurs

Hier zunächst ein paar Hinweise zum Kurs:

- Jeder Teilnehmer benötigt einen eigenen PC. Aufgrund des Platzangebots im Rosenwerk sind Laptops oder Tablets vorzuziehen, die die Teilnehmer selbst mitbringen. Nach Absprache können 1 bis 2 PCs gestellt werden. Bitte meldet dies vor Beginn des Kurses dem Kursleiter.
- Um Problemen beim Einrichten vorzubeugen, sollte jeder Teilnehmer vor Kursbeginn die Arduino Software auf dem eigenen Laptop / Tablet installieren. Dieser Vorgang benötigt kein Detailwissen und ist innerhalb von 10 Minuten erledigt. Eine Anleitung hierfür befindet sich im nächsten Kapitel. Bitte lest die Anleitung zunächst vollständig durch und startet erst danach die Installation. Bei Problemen oder Fragen kann gern der Kursleiter kontaktiert werden.
- Jeder Teilnehmer bekommt eines der unten abgebildeten Kits. Es beinhaltet alle benötigten Hardwarekomponenten und Kabel. Es ist kein Löten notwendig. Das Kit kann im Zeitraum des Kurses mit nach Hause genommen werden, muss jedoch nach Beendigung des Kurses wieder abgegeben werden. Eine Ausleihe darüber hinaus ist nach Absprache möglich.



- Essen und Trinken ist während des gesamten Kurszeitraums erlaubt. Es wird zusätzlich eine längere Pause eingeplant. Freut euch auf Kaffee, Mate und Kekse. Bitte achtet aber stets auf das richtige Zusammenspiel von Getränken und Elektronik/Laptops.
- Der Kurs ist so konzipiert, dass keine Vorkenntnisse notwendig sind. Eventuell werden Kursteilnehmer mit Erfahrung in der Programmierung hier etwas ausgebremst. Für die weitere Auseinandersetzung mit dem Thema sind Übungsaufgaben geplant. Diese können freiwillig zwischen Kursterminen bearbeitet werden. Hilfestellungen, Kontrolle und Tipps erfolgen per eMail.

- Es wird ein Handout ausgegeben. Zunächst als PDF, am Kursende in Papierform. Alle wichtigen Informationen sind hier zusammengefasst, sodass der Kurs ohne Abschreibübungen auskommt. Es empfiehlt sich dennoch die ein oder andere Notiz bzw. Ergänzung.
- Bitte Arbeitet den Kurs nicht in der Freizeit vor. Die Aufgaben der folgenden Abende sollten nur während der Kursabende bearbeitet werden.
- Die Zusammenarbeit und Austausch mit den anderen Kursteilnehmer ist explizit gewünscht. Besonders in den Zeiträumen der eigenständigen Arbeit und des Schaltungsaufbaus. Helft euch untereinander. Natürlich gilt: Alles Freiwillig, kein Muss.
- Bei Fragen und Unklarheiten während, vor und nach dem Kurs, einfach Fragen. Während des Vortrags, bitte ich um Meldung mit Handzeichen.
- Die Gesundheit der Teilnehmer und ihrer Technok ist mir sehr wichtig, bitte seid Vorsichtig beim Umgang mit den Kits und den anderen Teilnehmern.
- Sollte mal etwas kaputt gehen oder im Kit fehlen, meldet dies bitte sofort. Es ist Ersatz vorhanden. Jeder freut sich über ein vollständiges, funktionsfähiges Kit.
- Der Kurs ist in meiner Freizeit entstanden. Ich habe keinen Lektor und bin pädagogisch nicht ausgebildet. Ich habe an der Universität Übungen geleitet und Nachhilfe gegeben. Bei Problemen mit Rechtschreibfehlern und mit der Kursführung bitte ich dies zu berücksichtigen. Ich bin für Verbesserungen offen und werde jederzeit versuchen diese einzuarbeiten. Dabei steht für mich die Zielgruppe (Anfänger) und Nutzbarkeit des Wissens vor Expertentipps, Optimierungen, Hintergrundwissen und Vollständigkeit.
- Habt Spaß.

Software einrichten

Die sogenannten Arduino IDE (Integrated development enviroment / Integrierte Entwicklungsumgebung) ist der Namensgeber des Kurses. Sie wird uns während der Abende wesentlich bei der Programmierung unterstützen. Sie übernimmt im Hintergrund die nötige Verwaltung und Kommandos zur Übersetzung der Programmiersprache in die Maschinensprache. Dieser Vorgang („Kompilierung“) besteht aus vielen Kommandozeilen-Befehlen, Programmen und Skripten die im Hintergrund ablaufen. Der übersetzte Maschiencode kann anschließend mit der IDE in die entsprechenden Hardware geladen werden („Flashen“.) Zusätzlich bietet die Software zahlreiche Werkzeuge zum vereinfachen des Programmierprozesses auf die wir später im Kurs bei ihrer Benutzung eingehen werden.

Die Arduino IDE ist eine Open Source Anwendung und kann unter <https://www.arduino.cc/en/Main/OldSoftwareReleases#previous> kostenfrei für Linux, Mac und Windows heruntergeladen werden. Für den Kurs wird aus Gründen der Kompatibilität und des einheitlichen Aussehens die Verwendung der Version 1.6.9 (unter „Old Releases“) empfohlen. Die Installationsprozess sollte selbsterklärend sein (Standardinstallation).

Im Anschluss öffnet bitte die Arduino IDE und geht auf "Werkzeuge -> Board -> Boardverwalter" (bzw. in Englisch "Tools -> Board ... -> Boardmanager"). Hier installiert ihr das Paket "Arduino

SAMD Boards (32-bits ARM Cortex-M0+) by Arduino (Arduino/Genuino Zero)“ installieren. Nach anklicken erscheint hier die Schaltfläche zur Installation. Voreingestellt sollte die neueste Version 1.6.6 installiert werden. Hier werden im Hintergrund die richtigen Übersetzer für die Mikrocontroller nachgeladen.

Da wir im Kurs das Nucleo-Board von STM anstelle der klassischen Arduino-Hardware einsetzen werden, müssen wir nun noch das Addon STM32Duino für die Arduino-IDE installieren. Dieses Plugin erweitert die Umgebung, sodass diese Mikrocontroller/Boards ebenfalls angesprochen und programmiert werden können. Hierzu muss folgende gepackte Datei von https://github.com/rogerclarkmelbourne/Arduino_STM32/archive/master.zip heruntergeladen werden. Anschließend ist der Inhalt je nach Betriebssystem in einen der folgenden Ordner zu entpacken:

- Unter Windows: „Dokumente/Arduino/hardware“
(englische Systeme „My Documents/Arduino/hardware“)
- Unter MacOS: ~/Documents/Arduino/hardware/ . Zusätzlich müssen die Rechte für die COM-Schnittstelle freigegeben werden.
- Unter Linux: In den „hardware“-Ordner deiner Arduino-Installation („Sketches“-Ordner). Führe anschließend das Skript „tools/linux/install.sh“ aus. Zusätzlich müssen die Rechte für die COM-Schnittstelle (ttyACM) freigegeben werden.

Für MacOS und Linux sind zusätzlich noch weitere Schritte notwendig, welche ihr unter https://github.com/rogerclarkmelbourne/Arduino_STM32 nachlesen könnt. Nach erfolgreicher Installation der IDE und der Erweiterung sollte die Arduino IDE nach dem Start wie folgt (siehe Bild) aussehen. Im Menüpunkt „Werkzeuge->Board ... -> “ wird nun das „STM Nucleo F103RB“ aufgelistet.

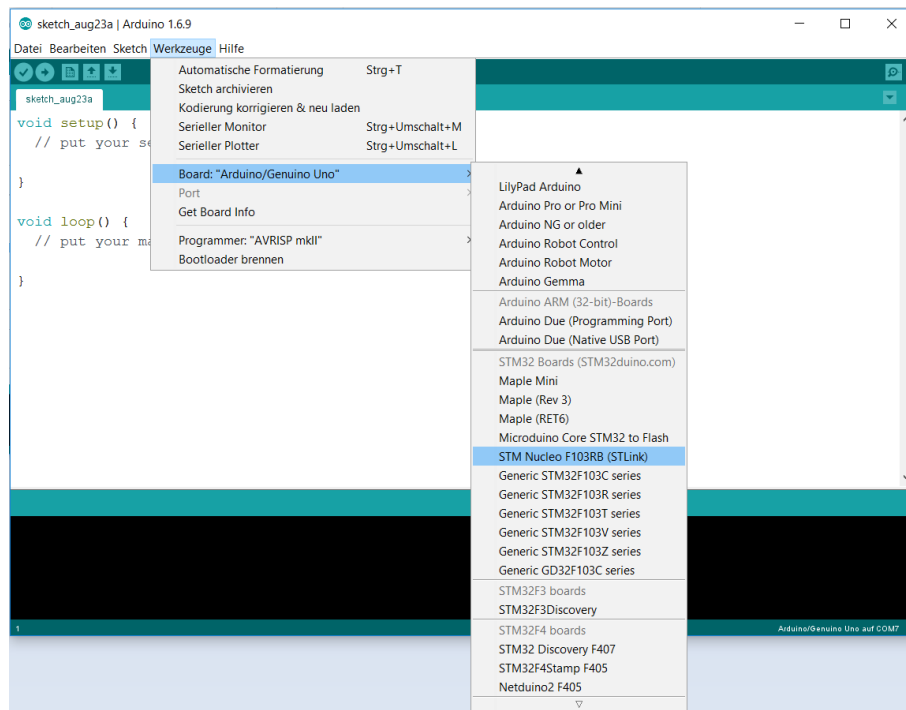


Abbildung 1: Arduino 1.6.9

Abschließend kann die Installation überprüft werden indem ein „leeres“ Projekt in Maschinensprache übersetzt wird. Dazu in der Arduino-IDE „Datei -> Neu“ Ausführen und prüfen, ob unter „Werkzeuge -> Board“ und „Werkzeuge -> Processor“ die richtigen Einstellungen vorgenommen sind. Ein „Sketch -> Überprüfen / Kompilieren“ sollte mit folgender Ausgabe in der Konsole erfolgreich durchlaufen. Anderfalls sollten zunächst die bisherigen Schritte überprüft werden. Sollte die Einrichtung weiterhin nicht funktionieren kann gern auch der Kursleiter kontaktiert werden. In entsprechenden Suchmaschinen, findet man zusätzlich meist Lösungen für einzelne Probleme entsprechend der Ausgabe in der Konsole.

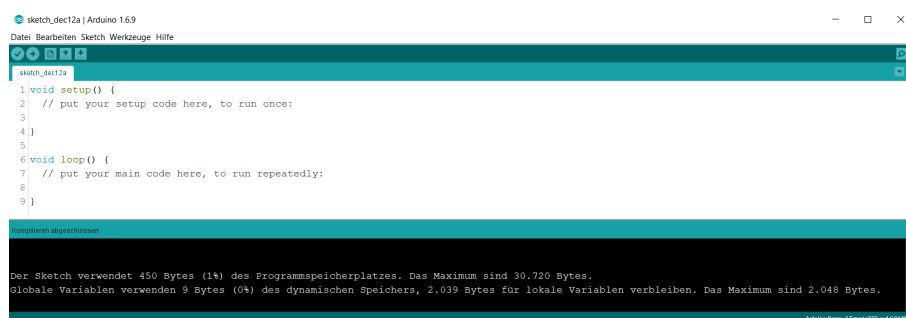


Abbildung 2: Ausgabe bei erfolgreicher Einrichtung

Circuits.io

CircuitIO ist ein Simulationswerkzeug für den Arduino. Es ist in der Standardversion kostenfrei. Die Nutzung erfolgt Web-basiert. D.h. zur Nutzung ist eine Registrierung und Internetverbindung erforderlich. Im Kurs wird besonders für die Vorführung und Erklärung der Funktionsweise von Programmen und Schaltungen CircuitIO eingesetzt. Auch während der eigenen Arbeit lohnt sich die Verwendung. So können Fehler bereits vorher erkannt, Hardwareaufbauten korrigiert und komplexe Programme getestet (Debugging) werden. Besonders die bereitgestellten Beispiele machen den Simulator zu einem einsteigerfreundlichen und mächtigen Helfer. Für Teilnehmer ist die Verwendung freiwillig. Die kostenlose Registrierung mit der persönlichen eMail-Adresse kann unter <https://circuits.io/> erfolgen. Die im Kurs verwendeten Schaltungen sind unter <https://circuits.io/users/763753/designs> zu finden.

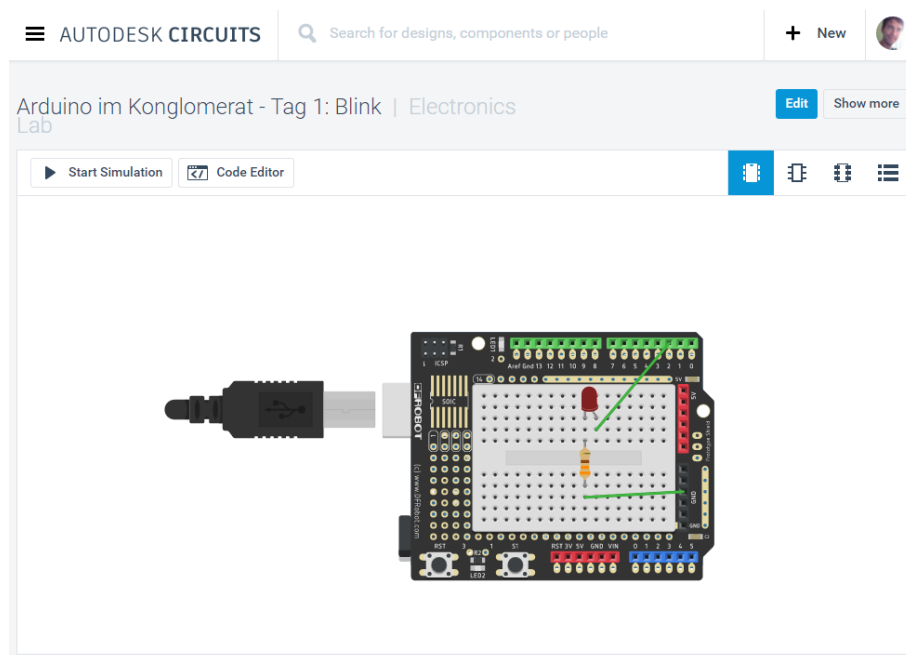


Abbildung 3: CircuitsIO

Einstieg und Auffrischung in die Programmierung:

Während de Kurses wird wenig auf die verwendete Programmiersprache C/C++ eingegangen. Zum Einstieg oder Auffrischung der Programmierkenntnisse empfehle ich, die für Kinder gedachte Einführung „Hour of Code“. Besonders die Reihe Blockly-Games (<https://blockly-games.appspot.com/>) ist eine geeignete Möglichkeit die grundlegenden Programmierkenntnisse zu erwerben. Hierfür sollten die die Teile von links nach rechts durchgearbeitet werden. Die Reihe ist für JavaScript gedacht, kann aber 1 zu 1 für die C/C++-Programmierung übernommen werden. Auf unterschiede wir im Kurs eingegangen.

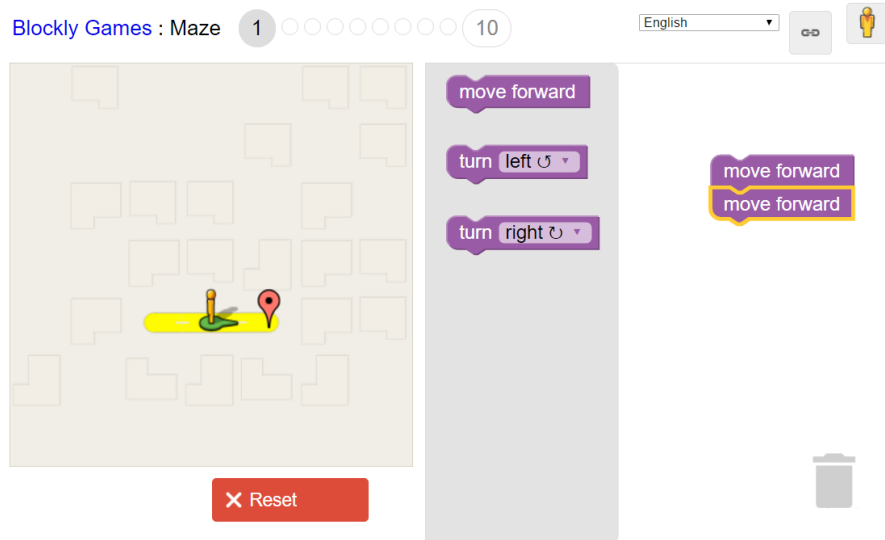


Abbildung 4: CircuitsIO

Github.com

Die Quellcodes die im Kurs verwendet werden findet ihr unter https://github.com/mdrobisch/arduino_kurs. Für den Kurs ist keine Registrierung notwendig.