

1.1 Die senseBox – Was erwartet dich in diesem Buch?

Die senseBox ist ein vielseitiges Toolkit, mit dem ein einfacher Einstieg in die Welt des Programmierens möglich ist. Spannende Projekte rund um Sensoren, Leuchtdioden und weitere elektronische Bauteile zeigen dir, wie du Umweltfragen mit dem Erlernen der Programmierung verknüpfen kannst.

Nachdem du einen Überblick über die verschiedenen Komponenten und Umweltsensoren der senseBox erhalten hast, wird die grafische Programmiersprache Blockly vorgestellt, mit der in diesem Buch gearbeitet wird. Danach wirst du dein erstes Programm erstellen und auf die senseBox übertragen.

Jedes Kapitel entspricht einem Projekt, das du natürlich jederzeit mit deinen eigenen Ideen erweitern oder anpassen kannst. Am Anfang jedes Kapitels werden die benötigten Materialien aufgelistet; Ausnahmen bilden die senseBox MCU (Micro Controller Unit) und das USB-Kabel, beide werden immer benötigt und deshalb nicht jedes Mal aufgeführt.

In den ersten Kapiteln lernst du die grundlegenden Konzepte der senseBox kennen und wirst diese in ersten Projekten anwenden.

Die Projekte sind thematisch sortiert und nach Schwierigkeit geordnet. Im weiteren Verlauf des Buchs wirst du an exponierten Stellen auf Themenboxen stoßen, die wichtige Informationen zu bestimmten Themen enthalten.

1.2 Was steckt drin?

Die senseBox:edu enthält eine Vielzahl von verschiedenen elektronischen Bauteilen, die wir in diesem Abschnitt kurz vorstellen. In den Projekten selbst gibt es oftmals noch mehr Informationen darüber, wie die Sensoren funktionieren oder wie du diese verwendest.

1.2.1 Die senseBox MCU

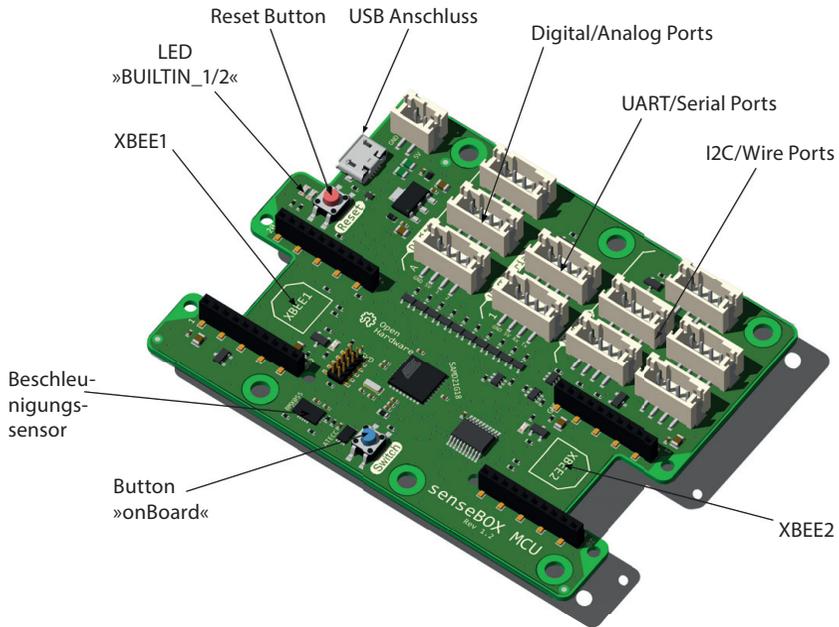


Abb. 1-1 Die senseBox MCU

Die senseBox MCU ist das Gehirn der senseBox und übernimmt alle Aufgaben zum Steuern, Messen und Regeln. Der Mikrokontroller ermöglicht es, verschiedene Sensoren, Ausgabegeräte und Datenübertragungsmodule anzuschließen. In den verschiedenen Projekten wird gezeigt, wie die einzelnen Bestandteile verwendet werden können.

Die senseBox MCU besitzt verschiedene Ports, die für Sensoren und Aktoren verwendet werden können. Der USB-Anschluss wird für die Stromversorgung und für die Datenübertragung verwendet. Direkt auf der senseBox MCU befinden sich zwei Buttons. Der rote Reset-Button kann zum Neustarten eines Programms verwendet werden und wird benötigt, um die senseBox in den Lernmodus zu versetzen. Mit einem Doppelklick auf den roten Reset-Button wird die senseBox MCU in den Lernmodus versetzt. Nur im Lernmodus kann ein neues Programm über den Date Explorer auf die senseBox MCU kopiert werden, anschließend startet die MCU neu und befindet sich im Programmmodus. Wie genau du einen Programmcode überträgst und die senseBox MCU in den Lernmodus versetzt, erfährst du im Abschnitt 1.5.

■ **USB-Anschluss:**

Über diesen Anschluss wird die senseBox MCU mit Strom versorgt und programmiert.

■ **Digital/Analog Ports:**

Über die digitalen und analogen Ports können einfache Sensoren und Aktoren angeschlossen werden.

■ **UART/Serial Ports:**

An diese Ports können noch weitere Sensoren angeschlossen werden, die nicht in der senseBox:edu enthalten sind. Ein Beispiel ist hier der Feinstaubsensor.

■ **I2C/Wire Ports:**

An diese fünf Ports kannst du Sensoren und das Display anschließen. Die Ports erkennen automatisch, an welchem Port ein Sensor angeschlossen ist.

■ **XBEE1/2:**

Diese zwei Steckplätze werden für die Datenübertragungsmodule verwendet.

von ± 1 hPa erfasst werden. Der Luftdruck kann auch zur Berechnung der Höhe verwendet werden. Die Temperatur wird in $^{\circ}\text{C}$ gemessen und mit einer Genauigkeit von $\pm 1,0^{\circ}\text{C}$ erfasst.

Lichtsensor

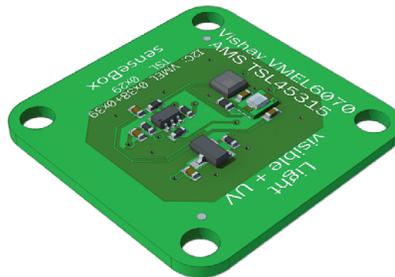


Abb. 1-4 Der Lichtsensor (VEM6070 von Vishay/TSL45315 von AMS)

Der Lichtsensor kombiniert zwei verschiedene Sensoren. Zum einen kann die Helligkeit in Lux und die UV(A)-Intensität in $\mu\text{W}/\text{m}^3$ erfasst werden.

Ultraschall-Distanzsensor

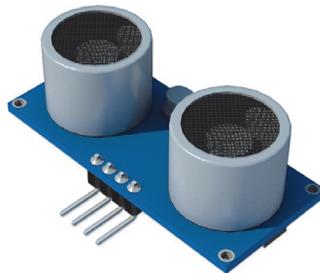


Abb. 1-5 Der Ultraschall-Distanzsensor (HC-SR04)

Der Ultraschall-Distanzsensor kann Distanzen zwischen 5 und 200 cm erfassen. Bei einer größeren Distanz werden die Messungen ungenau oder sprunghaft. Gemessen wird mithilfe eines Ultraschallsignals. Über die Laufdauer des Signals kann dann die Distanz auf Grundlage der Schallgeschwindigkeit bestimmt werden.

Mikrofon

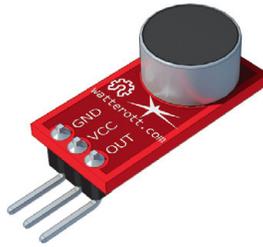


Abb. 1-6 Das Mikrofon (Mic-Breakout)

Das Mikrofon kann genutzt werden, um akustische Signale zu erfassen. Das Mikrofon ist ein analoger Sensor, und der Messwert wird in Volt (Pegel zwischen 0 und 5 V) ausgegeben.

Beschleunigungssensor

Der Beschleunigungssensor (BMX055 von Bosch) ist direkt auf die sense-Box MCU aufgelötet und kann die Beschleunigung der drei Raumachsen erfassen. Die Messwerte werden als Vielfaches der g -Kraft ausgegeben.

1.2.3 Weitere Bauteile

Verbindungskabel

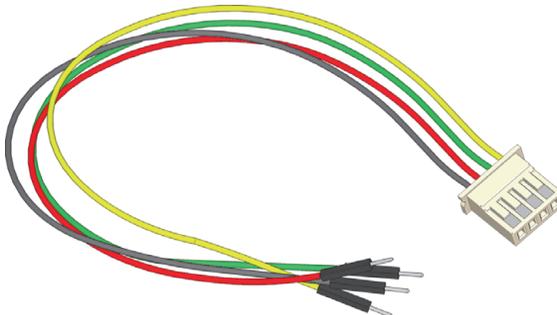


Abb. 1-7 JST-Adapterkabel

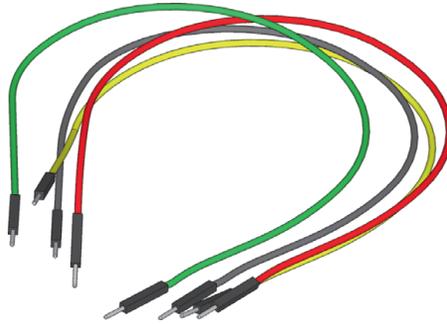


Abb. 1-8 Steckkabel

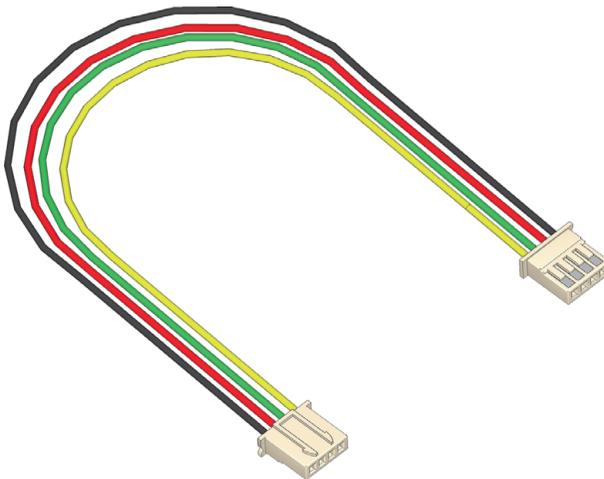


Abb. 1-9 JST-JST-Kabel

Um die Sensoren und die weiteren Bauteile mit der senseBox MCU zu verbinden, stehen dir verschiedene Kabel zur Verfügung. Das **JST-Adapterkabel** wird dann verwendet, wenn du eine Schaltung auf dem Breadboard erstellst. Hierbei unterstützen auch die **Steckkabel**.

Das **JST-JST-Kabel** benötigst du, wenn du einen senseBox-Sensor oder das Display direkt mit der senseBox MCU verbinden willst.

Das MicroSD-Bee wird an Steckplatz XBEE 2 gesteckt und ermöglicht dir, die Daten der Sensoren auf SD-Karte zu speichern oder auch Daten von der SD-Karte zu lesen.

Kleinteile

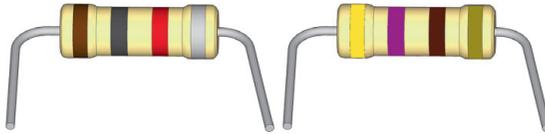


Abb. 1-13
10-k Ω -Widerstand

Abb. 1-14
470- Ω -Widerstand



Abb. 1-15 Piezo



Abb. 1-16 Potenziometer



Abb. 1-17 Buttons

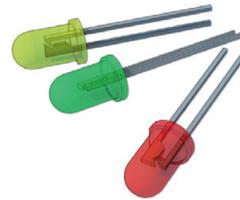


Abb. 1-18
Verschiedene LEDs (fünf rote, fünf gelbe, fünf grüne)



Abb. 1-19 USB-Kabel

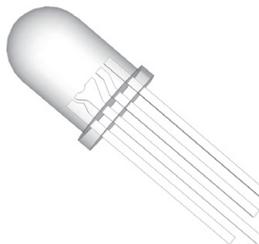


Abb. 1-20 RGB-LED



Abb. 1-21
LDR – lichtabhängiger Widerstand

Zusätzlich ist in der senseBox noch eine Vielzahl von kleinen elektronischen Bauteilen enthalten. **Widerstände** werden für verschiedene Schaltungen benötigt. In der senseBox finden sich zwei verschiedene Widerstände, 10 k Ω und 470 Ω . Erkennen kannst du Widerstände immer anhand der verschiedenen Farbringe, diese geben dir auch an, um welche Stärke des Widerstands es sich handelt.

Der **LDR** ist ein Widerstand, der in Abhängigkeit zur Helligkeit seinen Wert verändert.

Mit dem **Piezo** kannst du Töne erzeugen und die Lautstärke mit dem Potenziometer beeinflussen. Das **Potenziometer** kann auch als analoger Sensor verwendet werden. Die **Knöpfe** sind eine praktische Eingabemöglichkeit für verschiedene Programme.

Das **USB-Kabel** wird benötigt, um die senseBox MCU mit Strom zu versorgen und ein Programm zu übertragen.

Die **RGB-LED** hat 4 Pins und kann in verschiedenen Farben leuchten, während eine normale **LED** nur in einer Farbe leuchtet.

1.3 Die Programmieroberfläche

Die Programmierung der senseBox kann auf verschiedene Art und Weise geschehen. Zum einen kann sie mit der Arduino-IDE programmiert werden. Hier wird in einer Programmiersprache gearbeitet, die von der Syntax ähnlich zu C++ ist. In diesem Buch wollen wir aber zeigen, wie die senseBox mit »Blockly für senseBox« programmiert wird.

Blockly ist eine grafische Programmierumgebung, die es ermöglicht, einen Programmcode mit Bausteinen zusammenzustellen. Ähnlich wie bei einem Puzzle passen die Bausteine ineinander und ergeben bei richtiger Kombination ein funktionsfähiges Programm.

1.3.1 Blockly für senseBox

Blockly für die senseBox erreichst du unter: <https://blockly.sensebox.de>. Es läuft ohne Installation in allen gängigen Webbrowsern (ist aber nicht mit dem Internet Explorer kompatibel).

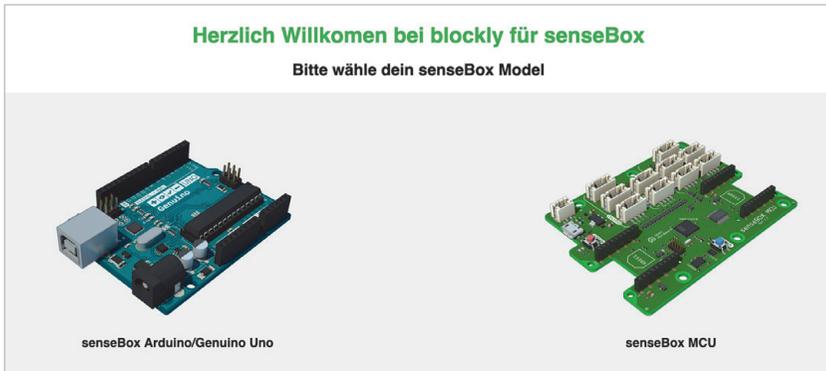


Abb. 1–22 Auswahl zwischen Arduino Uno und senseBox MCU

»Blockly für senseBox« kann auch für Arduino-basierte Projekte verwendet werden, alle Projekte in diesem Buch verwenden allerdings die senseBox MCU. Mit einem Klick auf die senseBox MCU kommst du direkt zur Programmierumgebung.

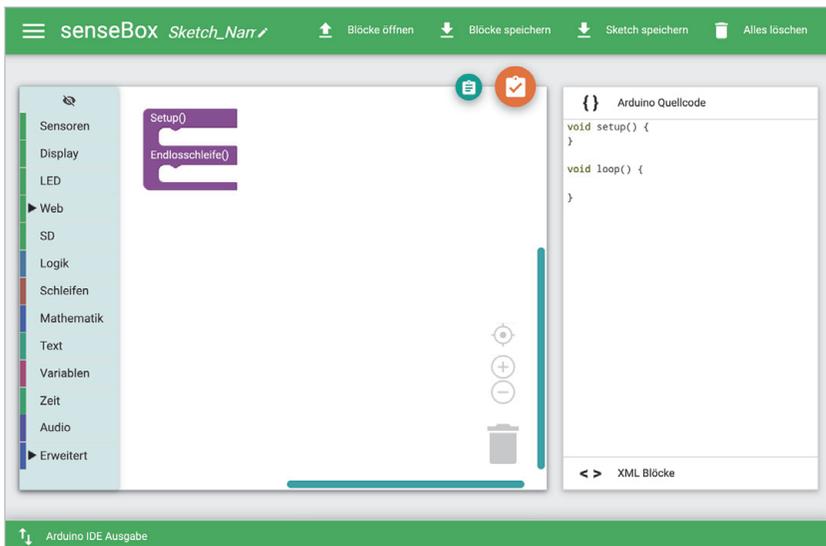


Abb. 1–23 Übersicht »Blockly für senseBox«

Die Programmierumgebung hat verschiedene Bereiche und Funktionen, die dir im Folgenden kurz erklärt werden.

1. Blöcke speichern und öffnen

Hier kannst du deinem Projekt einen Namen geben und es abspeichern. Mit den Buttons *Blöcke öffnen* und *Blöcke speichern* kannst du dein aktuelles Programm als Blöcke speichern oder später auch wieder öffnen. Der Button *Sketch speichern* speichert den Code in Textform ab. Über den Button *Alles löschen* kannst du alle Blöcke aus dem Arbeitsbereich löschen.

2. Der Arbeitsbereich

Im Arbeitsbereich kannst du die Blöcke zusammenbauen und deinen Programmcode erstellen. Die Größe der Blöcke kannst du über Plus und Minus anpassen. Möchtest du einen Block löschen, schiebst du ihn in die kleine Mülltonne.

3. Code kompilieren

Mit dem orangen Button kannst du den Programmcode kompilieren und auf die senseBox MCU übertragen. Mehr dazu in Abschnitt 1.5. Mit dem blauen Button kannst du den Programmcode in Textform in die Zwischenablage kopieren.

4. Programmcode in Textform

Hier wird der Programmcode in Textform angezeigt. Der Programmcode in Textform kann zum Beispiel zum Arbeiten mit der Arduino-IDE verwendet werden. Was genau die Arduino ist, wird dir in Abschnitt 1.3.2 erklärt.

5. Die Toolbox

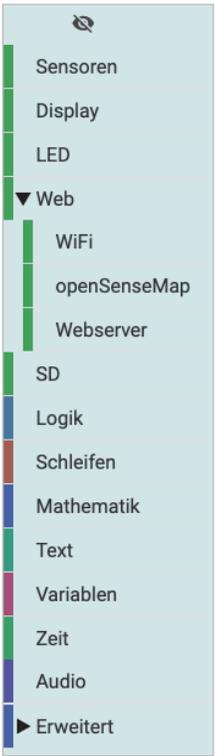
 <p>The image shows a vertical list of categories in a toolbox. From top to bottom: Sensoren (green), Display (green), LED (green), Web (green with a downward arrow), WiFi (green), openSenseMap (green), Webserver (green), SD (green), Logik (blue), Schleifen (orange), Mathematik (blue), Text (green), Variablen (purple), Zeit (green), Audio (blue), and Erweitert (blue with a rightward arrow).</p>	<p>Die Toolbox auf der linken Seite gibt dir Zugriff auf die einzelnen Programmierblöcke. Neben speziellen Blöcken für die senseBox gibt es auch eine Vielzahl von Blöcken, die die Grundlagen der Programmierung abdecken.</p> <p>Die Blöcke speziell für die senseBox gliedern sich wie folgt auf:</p> <p>Sensoren: Hier findest du die Blöcke für alle Sensoren, die in der senseBox enthalten sind bzw. für die senseBox verfügbar sind.</p> <p>Display: Hier findest du alle Blöcke, die für das Display benötigt werden.</p> <p>LED: Hier findest du alle Blöcke für die LED und die RGB-LED.</p> <p>WiFi: Hier findest du die Blöcke für das WiFi-Bee.</p> <p>openSenseMap: Alle Blöcke, um Messwerte an die openSenseMap zu senden, findest du hier.</p> <p>Webserver: Alle Blöcke für den Webserver findest du hier.</p> <p>SD: Hier findest du alle Blöcke, um Messwerte auf einer SD-Karte zu speichern.</p>
--	--

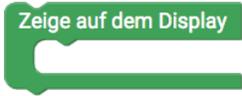
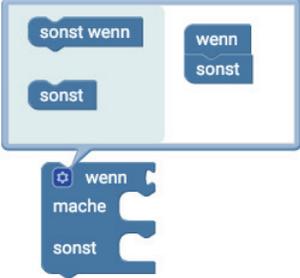
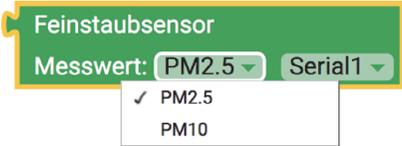
Abb. 1-24
Übersicht Toolbox

Die Blöcke stehen für Programmierbefehle und bieten teilweise verschiedene Einstellungsmöglichkeiten. Die folgende Übersicht hilft dir zu verstehen, welche Art von Blöcken es gibt und wie sie verwendet werden können.

 <p>The image shows a green block with the title 'Temperatur/Luftfeuchtigkeitssensor (HDC1080)'. Below the title, it says 'Messwert: Temperatur in °C' with a small downward arrow next to the text.</p>	<p>Diese Blöcke müssen immer an andere Blöcke angefügt werden. Sie geben einen Wert zurück (zum Beispiel einen Messwert).</p>
---	---

Abb. 1-25 Blöcke mit Rückgabewert

→

 <p>Abb. 1-26 Blöcke mit offenen Blockabschnitten</p>	<p>In diese Blockabschnitte kannst du verschiedene Aktionen einfügen, die dann nacheinander ausgeführt werden.</p>
 <p>Abb. 1-27 Blöcke mit Kontextmenü</p>	<p>Einige Blöcke haben ein kleines Zahnrad an der Seite. Mit einem Klick auf das Zahnrad kannst du das Kontextmenü öffnen und den Block um weitere Funktionen ergänzen. Ein weiterer Klick auf das Zahnrad schließt das Kontextmenü wieder.</p>
 <p>Abb. 1-28 Blöcke mit Drop-down-Menü</p>	<p>Das Drop-down-Menü eines Blocks gibt dir Zugriff auf weitere Funktionen und Rückgabewerte. So kann bei Sensoren zum Beispiel ausgewählt werden, welcher Messwert zurückgegeben oder an welchen Anschluss der Sensor angeschlossen wird.</p>

1.3.2 Arduino und die Arduino-IDE

Arduino ist eine Physical-Computing Plattform, die aus Soft- und Hardware besteht. Sowohl die Software als auch die Hardware werden Open Source angeboten. Die Hardware besteht aus einer Vielzahl von verschiedenen einfachen Mikrocontrollern mit analogen und digitalen Ein- und Ausgängen. Die Entwicklungsumgebung, die Arduino-IDE, arbeitet mit einer Programmiersprache, die viele Ähnlichkeiten zu C bzw. C++ aufweist, allerdings auch an vielen Stellen vereinfacht wurde, um einen leichten Einstieg zu ermöglichen.

Die senseBox MCU kann auch mithilfe der Arduino-IDE programmiert werden. Das Programmieren in der Arduino-IDE erfolgt textbasiert und bietet dir einen größeren Funktionsumfang. Allerdings ist der Einstieg nicht so einfach wie mit einer grafischen Programmieroberfläche, da mehr Fehler in der Syntax gemacht werden können. In »Blockly für sense-Box« siehst du auf der rechten Seite immer auch den Programmcode für die Arduino-IDE. Die Arduino-IDE lässt sich unter folgender Adresse herunterladen: <https://www.arduino.cc/en/Main/Software>. Zur Verwendung der Arduino-IDE muss ein zusätzliches Board Support Package installiert werden. Eine ausführliche Anleitung zum Installieren des Board Support Package findest du auf www.sensebox.de unter *Material*.

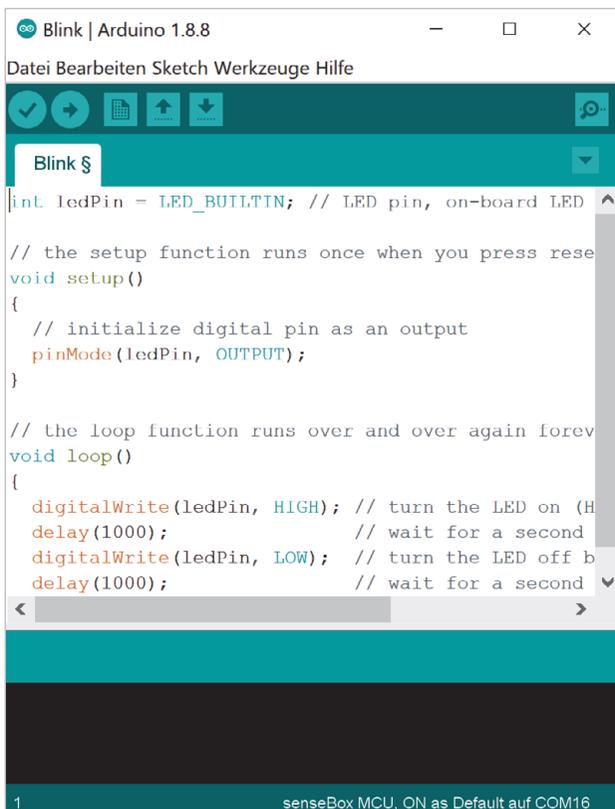


Abb. 1-29 Die Arduino-IDE

1.3.3 NEPO – openRoberta

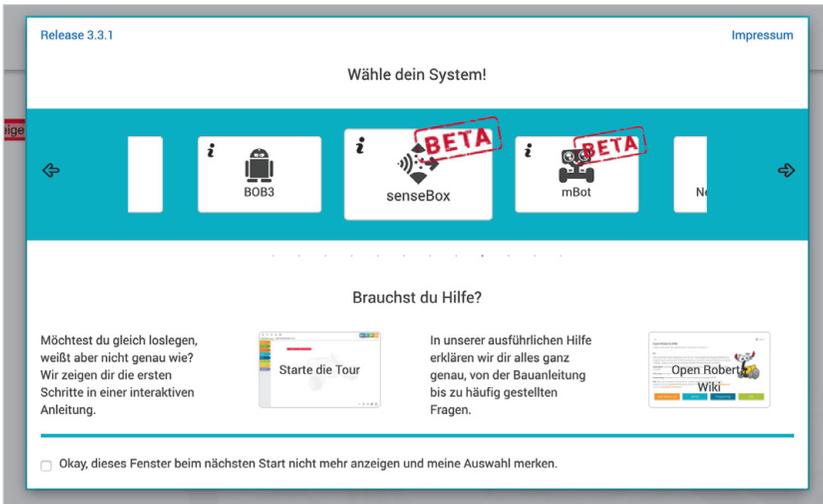


Abb. 1–30 Auswahl des Systems im openRoberta-Lab

Das openRoberta-Lab (<https://lab.open-roberta.org>) ist eine universelle Programmierumgebung für eine Vielzahl von Tools und Bausätzen. Neben Lego-Robotern und dem Calliope mini kann damit auch die senseBox programmiert werden. Viele Funktionen sind sehr ähnlich aufgebaut wie in »Blockly für senseBox«, allerdings ist der Funktionsumfang insgesamt nicht so weitreichend. Einige der Projekte lassen sich trotzdem auch in openRoberta umsetzen.

1.4 Sicherheitshinweise beim Arbeiten mit der senseBox

Die senseBox ist ein elektronisches Gerät. Deshalb ist es wichtig, vorsichtig mit den Komponenten umzugehen und gängige Sicherheitsvorkehrungen beim Umgang mit elektronischen Geräten zu beachten. Dazu gehört das Durchlesen des Produktsicherheitsblatts, das der senseBox beiliegt.

Wir Autoren haben noch zusätzliche Sicherheitstipps:

- Entlade dich vor der Arbeit mit der senseBox (zum Beispiel kann man sich entladen, indem man eine metallische Oberfläche wie ein Heizungsrohr anfasst).
- Achte darauf, dass der Mikrocontroller von der Stromquelle getrennt werden sollte, sobald Komponenten ausgetauscht werden.
- Passe darauf auf, dass du die Projekte in trockenen Umgebungen durchführst und die Bauteile von kleinen Kindern fernhältst.
- Fasse die senseBox MCU und die Sensoren an den Ecken an und berühre die Sensorköpfe und Bauteile nicht direkt, um diese nicht zu beschädigen.
- Wenn dir auffällt, dass die senseBox defekt sein könnte, nutze diese nicht weiter und nimm Kontakt mit dem Hersteller auf.

1.5 Los geht's

Bevor du mit den Projekten anfängst, wird dir im folgenden Abschnitt gezeigt, wie du ein Programmcode auf die senseBox MCU überträgst. Die senseBox MCU kann einen Programmcode speichern, sodass dieser auch nach dem Trennen der Stromverbindung noch vorhanden ist.

Schritt 1: Die Programmierumgebung

Öffne Blockly für die senseBox, indem du mit einem geeigneten Internetbrowser (Chrome, Firefox oder Edge) auf <https://blockly.sensebox.de> gehst. Wähle die senseBox MCU, um zur richtigen Programmieroberfläche zu gelangen.

Schritt 2: Programmieren

In diesem Beispiel wird eine der beiden aufgelöteten LEDs angeschaltet. Du musst also keine weitere Schaltung aufbauen.

Ziehe den Block LED an digitalen Pin in den Blockabschnitt Endloschleife.

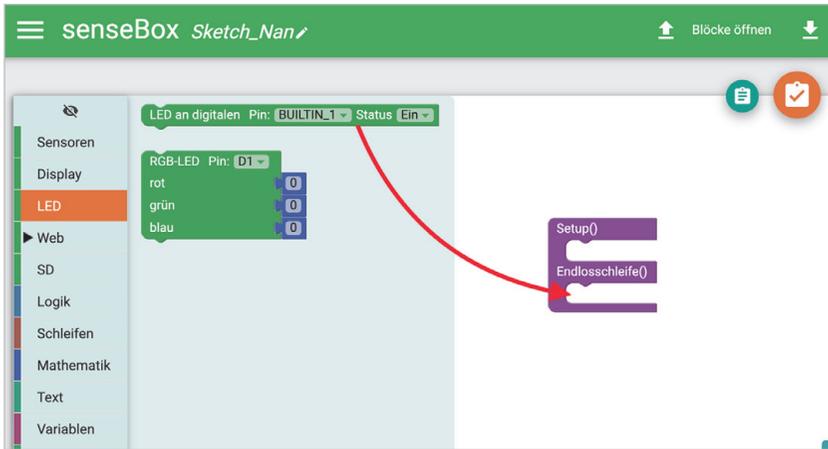


Abb. 1–31 Ziehe den Block in den Abschnitt Endlosschleife.

Welche genaue Funktion die Endlosschleife und auch der Block übernehmen, erfährst du im folgenden Kapitel. Die Programmierung ist an dieser Stelle abgeschlossen, und du kannst mit dem nächsten Schritt fortfahren.

Schritt 3: Programmcode übertragen

1. Verbinden:

Das Programm muss nun auf die senseBox MCU übertragen werden. Verbinde dazu die senseBox MCU mithilfe des USB-Kabels mit deinem Computer.



Abb. 1–32 Anschluss des USB-Kabels an einen Laptop

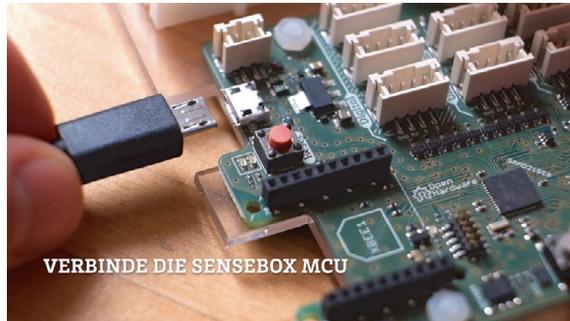


Abb. 1-33 Anschluss des USB-Kabels an die senseBox MCU

Mache anschließend einen Doppelklick auf den Reset-Button, um die senseBox MCU in den Lernmodus zu versetzen. Im Lernmodus leuchtet die kleine rote LED neben dem Reset-Button auf, und die senseBox MCU wird als Wechseldatenträger (zum Beispiel im Windows Explorer) erkannt.

Nur im Lernmodus kannst du einen Programmcode auf die senseBox MCU übertragen. Nach dem erfolgreichen Übertragen startet die senseBox MCU neu und befindet sich anschließend im Programmmodus.

2. Kompilieren:

Bevor der Programmcode übertragen werden kann, muss dieser kompiliert werden. Das bedeutet, dass die Blöcke in eine Maschinsprache übersetzt werden. Klicke dazu auf den orangenen Button in der Oberfläche.

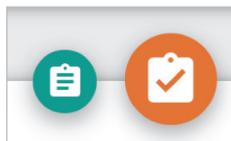


Abb. 1-34 Mit dem orangenen Button kann der Programmcode kompiliert werden.

Der Code wird nun online kompiliert, und anschließend startet ein Download.

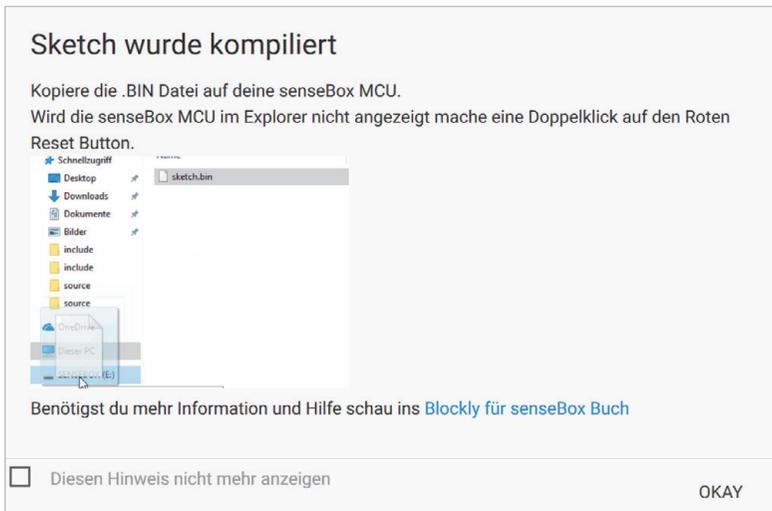


Abb. 1-35 Die Infografik in Blockly zeigt dir, wie du den Code übertragen musst.

3. Übertragen:

Wenn du die senseBox MCU erfolgreich in den Lernmodus versetzt hast, wird diese nun im Windows Explorer angezeigt. Um das zuvor kompilierte Programm zu übertragen, ziehe es per Drag-and-drop auf die senseBox MCU.

Kopieren unter Mac OS X

Das Kopieren der Datei über den Finder in Mac OS X ist leider nicht so einfach möglich wie unter Windows. Ein kleines Tool hilft dir aber dabei, die Datei auf die senseBox MCU zu kopieren. Das Tool führt dich Schritt für Schritt zum Kopieren der `.bin`-Datei auf deine senseBox MCU.

Das Tool findest du zum Download unter: <https://sensebox.de/buch>.

Nach dem erfolgreichen Kopieren verlässt die senseBox MCU den Lernmodus und führt das Programm aus. Hast du alles wie in diesem Abschnitt erläutert durchgeführt, leuchtet die kleine rote LED neben dem Reset-Button auf.

Super, du hast dein erstes Programm übertragen. Immer wenn du Änderungen an dem Programmcode vornimmst, musst du diesen Schritt zum Übertragen des Programms wiederholen.