

Bartoschek · Fehrenbach
Fehrenbach · Pesch · Steinmann

DAS SENSEBOX BUCH

12 Projekte rund um
Sensoren, Umwelt und IoT



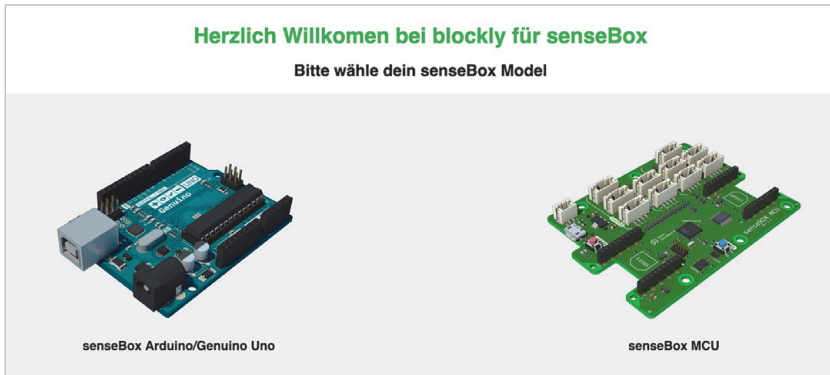


Abb. 1–22 Auswahl zwischen Arduino Uno und senseBox MCU

»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.

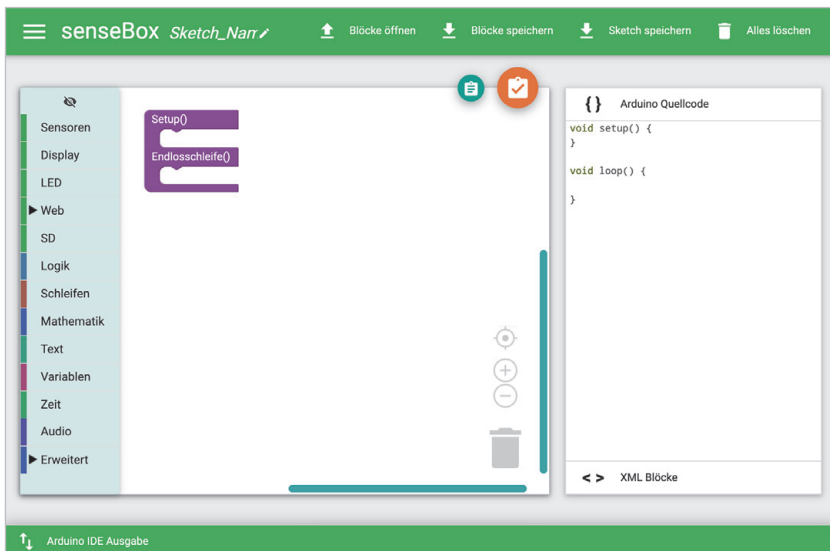


Abb. 1–23 Übersicht »Blockly für senseBox«

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

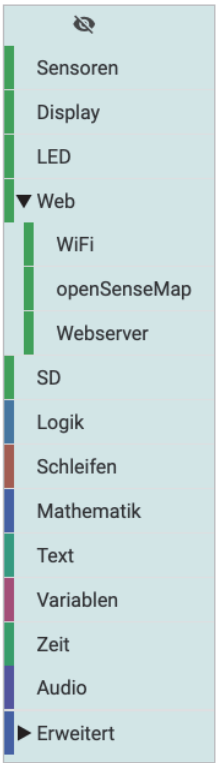
 <p>The screenshot shows a vertical list of toolbox categories. From top to bottom: Sensoren (green), Display (green), LED (green), Web (green with a downward arrow), WiFi (green), openSenseMap (green), Webserver (green), SD (green), Logik (blue), Schleifen (orange), Mathematik (blue), Text (green), Variablen (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.

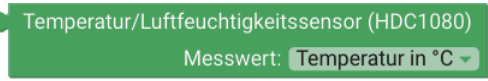
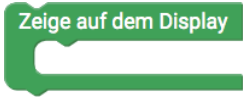
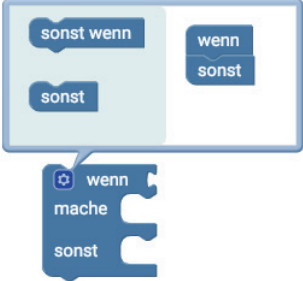
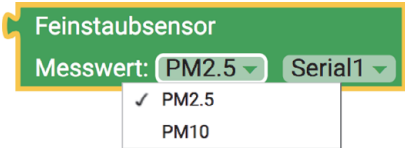
 <p>The screenshot shows a green block titled 'Temperatur/Luftfeuchtigkeitssensor (HDC1080)'. Below the title, it says 'Messwert:' followed by a dropdown menu currently displaying 'Temperatur in °C'.</p>	<p>Diese Blöcke müssen immer an andere Blöcke angefügt werden. Sie geben einen Wert zurück (zum Beispiel 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 senseBox« 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*.



The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.8". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". Below the menu bar is a toolbar with icons for saving, running, uploading, and downloading. The main editor area displays the following C++ code for a Blink sketch:

```
int ledPin = LED_BUILTIN; // LED pin, on-board LED

// the setup function runs once when you press reset
void setup()
{
  // initialize digital pin as an output
  pinMode(ledPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop()
{
  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the positive voltage)
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // turn the LED off by making the pin LOW (no voltage)
  delay(1000); // wait for a second
}
```

At the bottom of the IDE, a status bar indicates "1 senseBox MCU, ON as Default auf COM16".

Abb. 1-29 Die Arduino-IDE