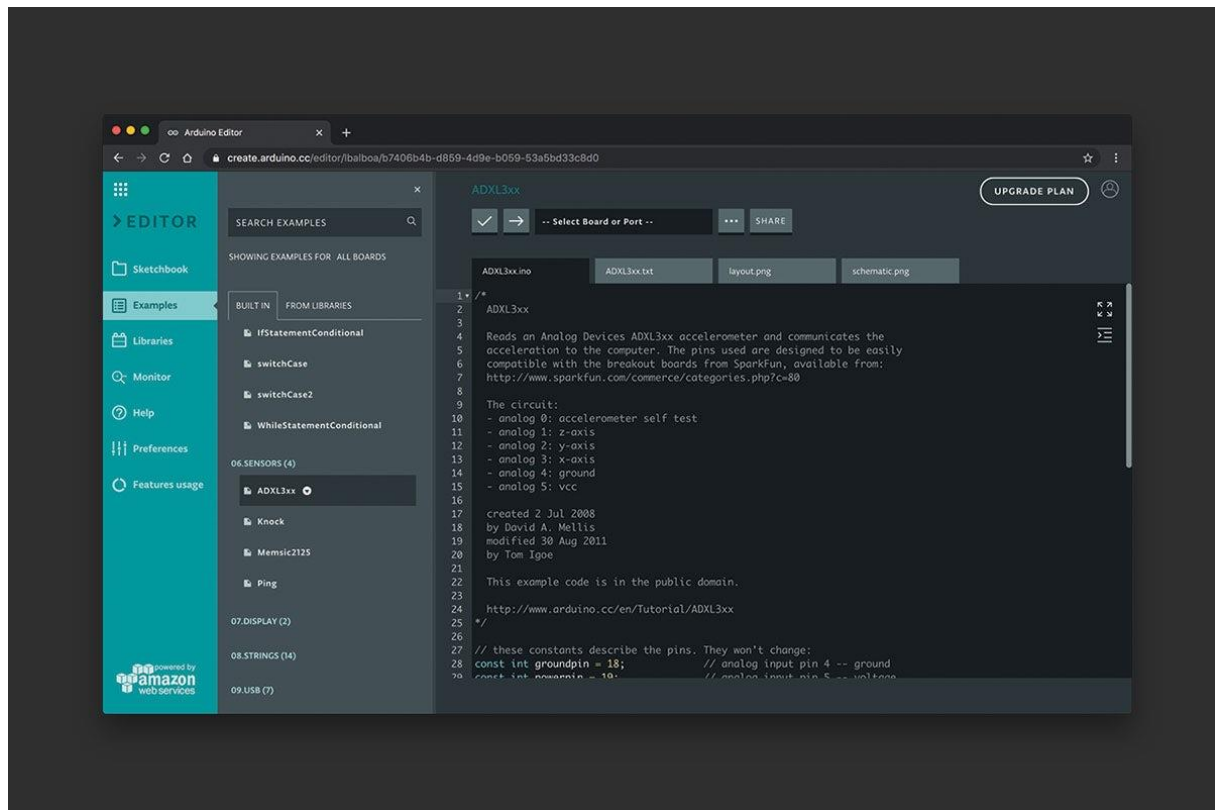
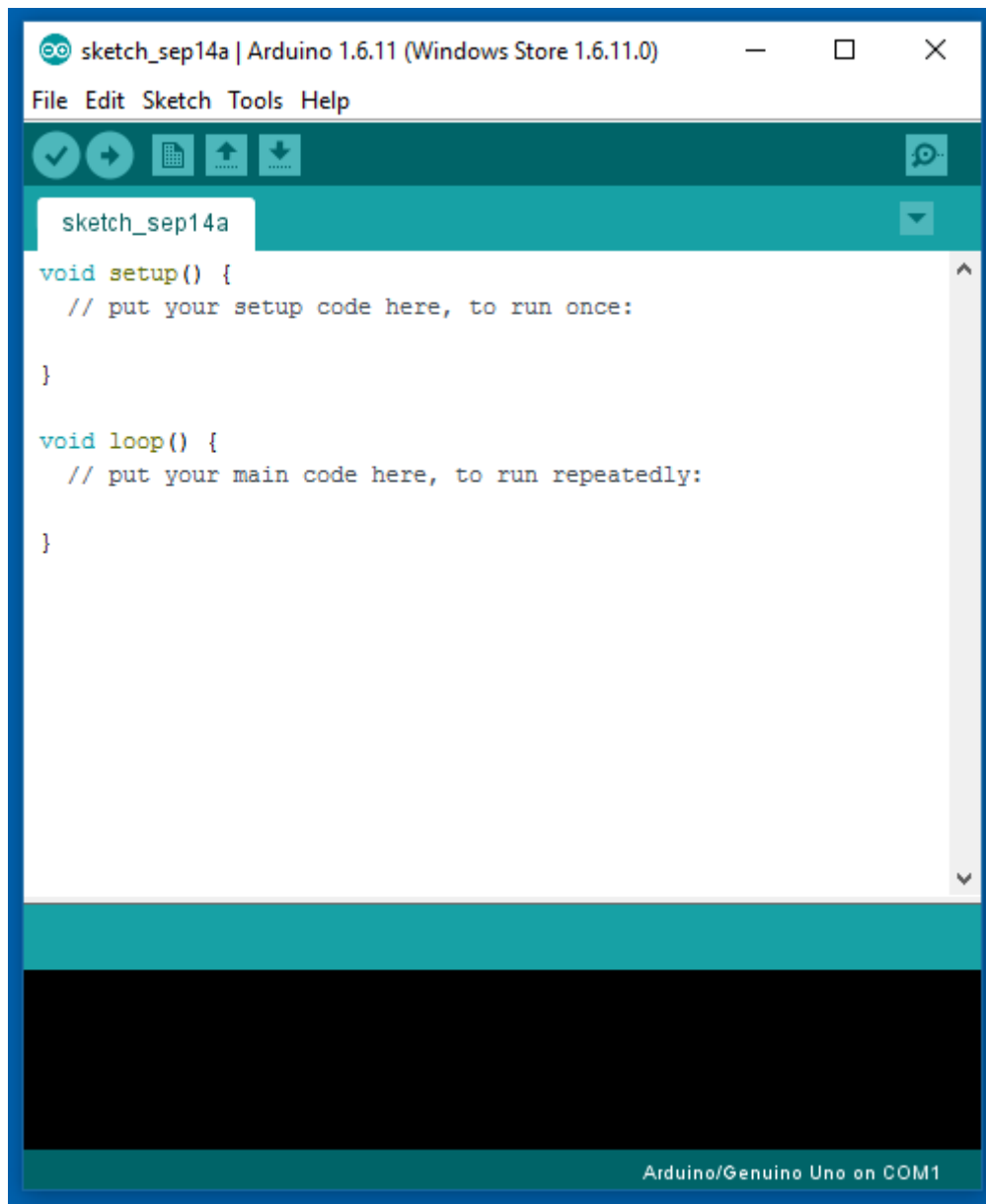



Schritt 1: Arduino IDE herunterladen

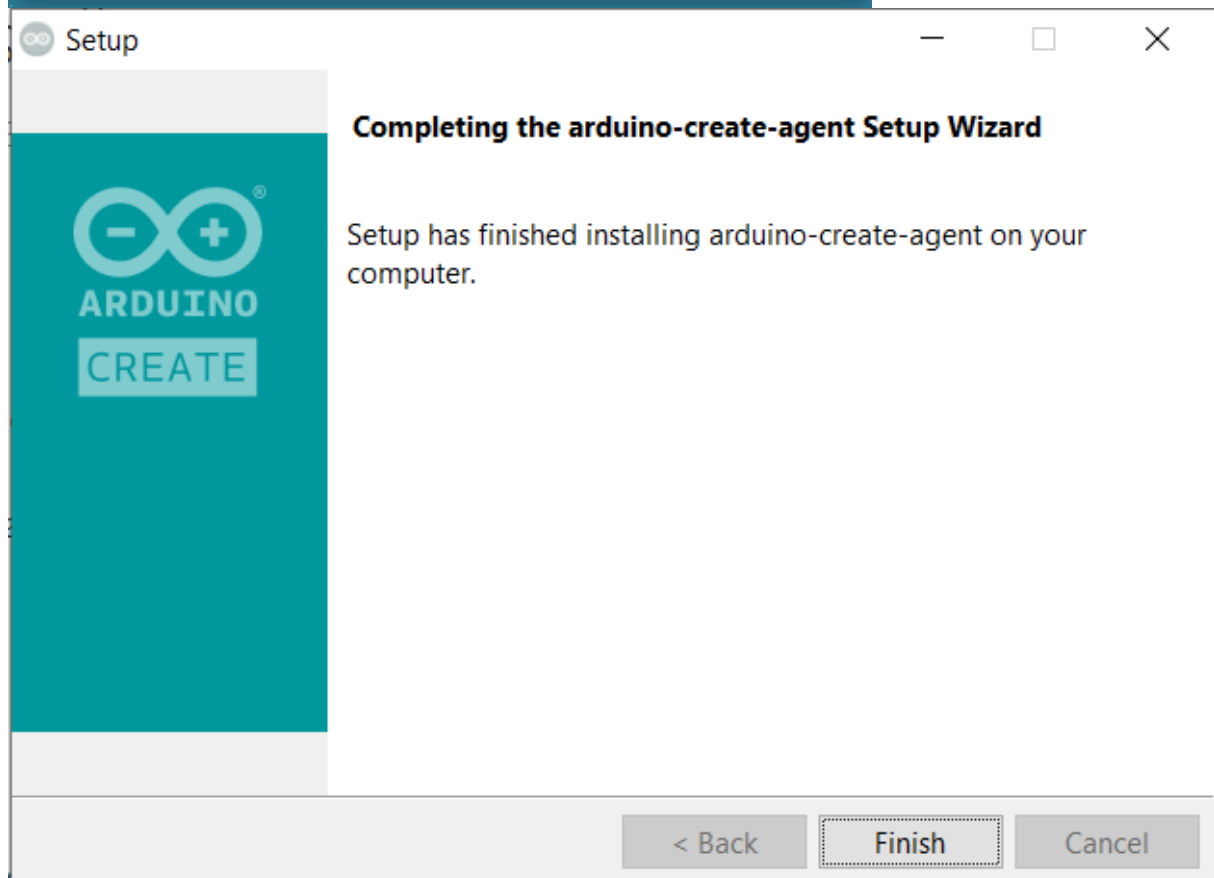
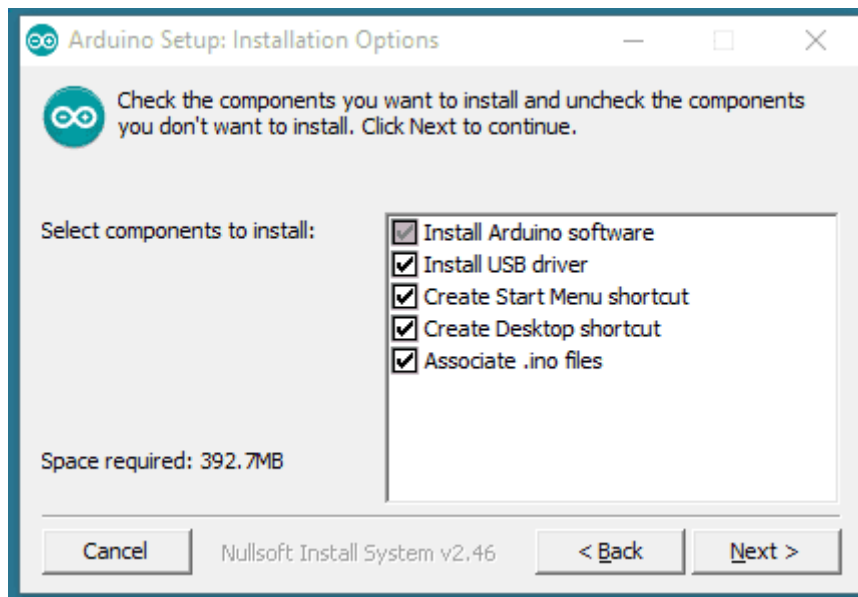


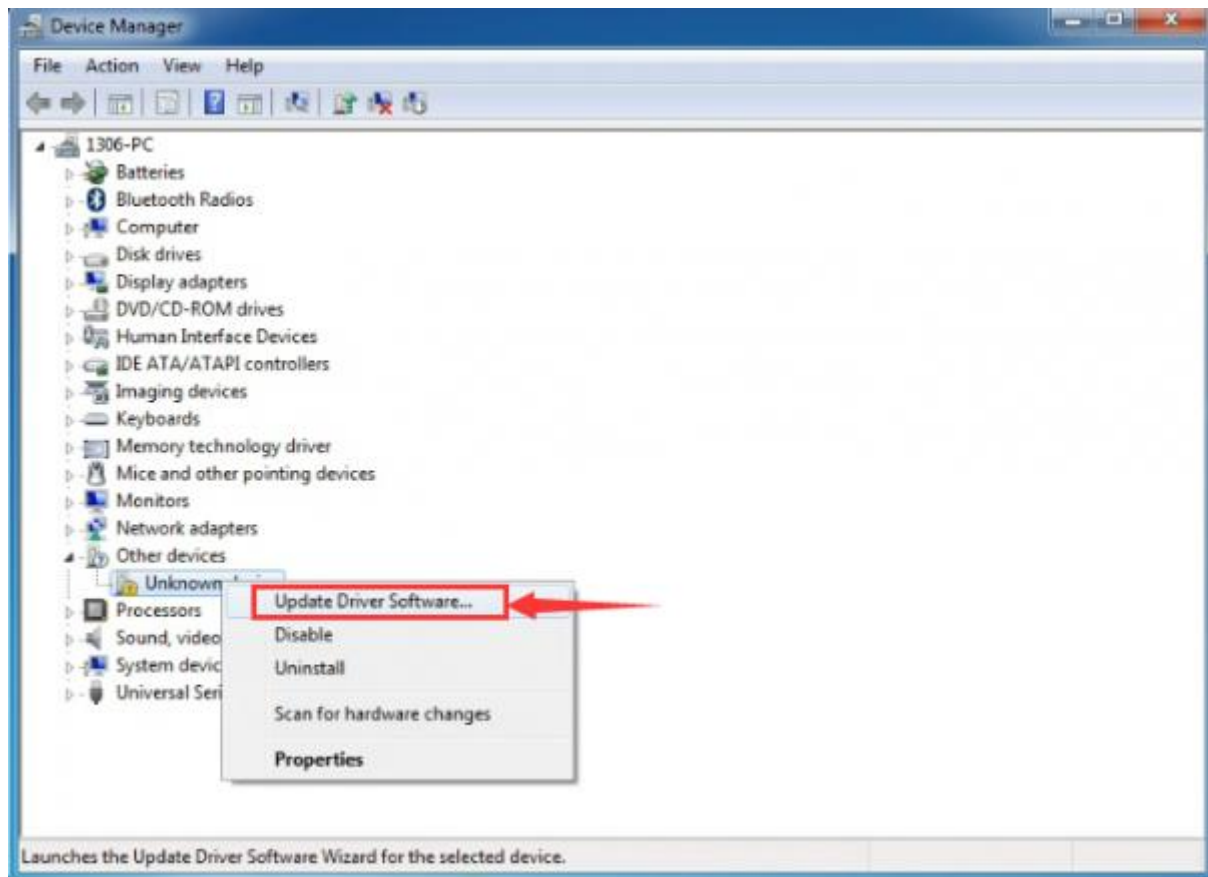


1. Öffne einen Webbrowser
2. Gehe auf die offizielle Arduino-Seite:
 <https://www.arduino.cc/en/software>
3. Wähle **Windows**
4. Klicke auf **Just Download**
(Spenden sind freiwillig)

 Die Installationsdatei wird heruntergeladen

◆ Schritt 2: Arduino IDE installieren

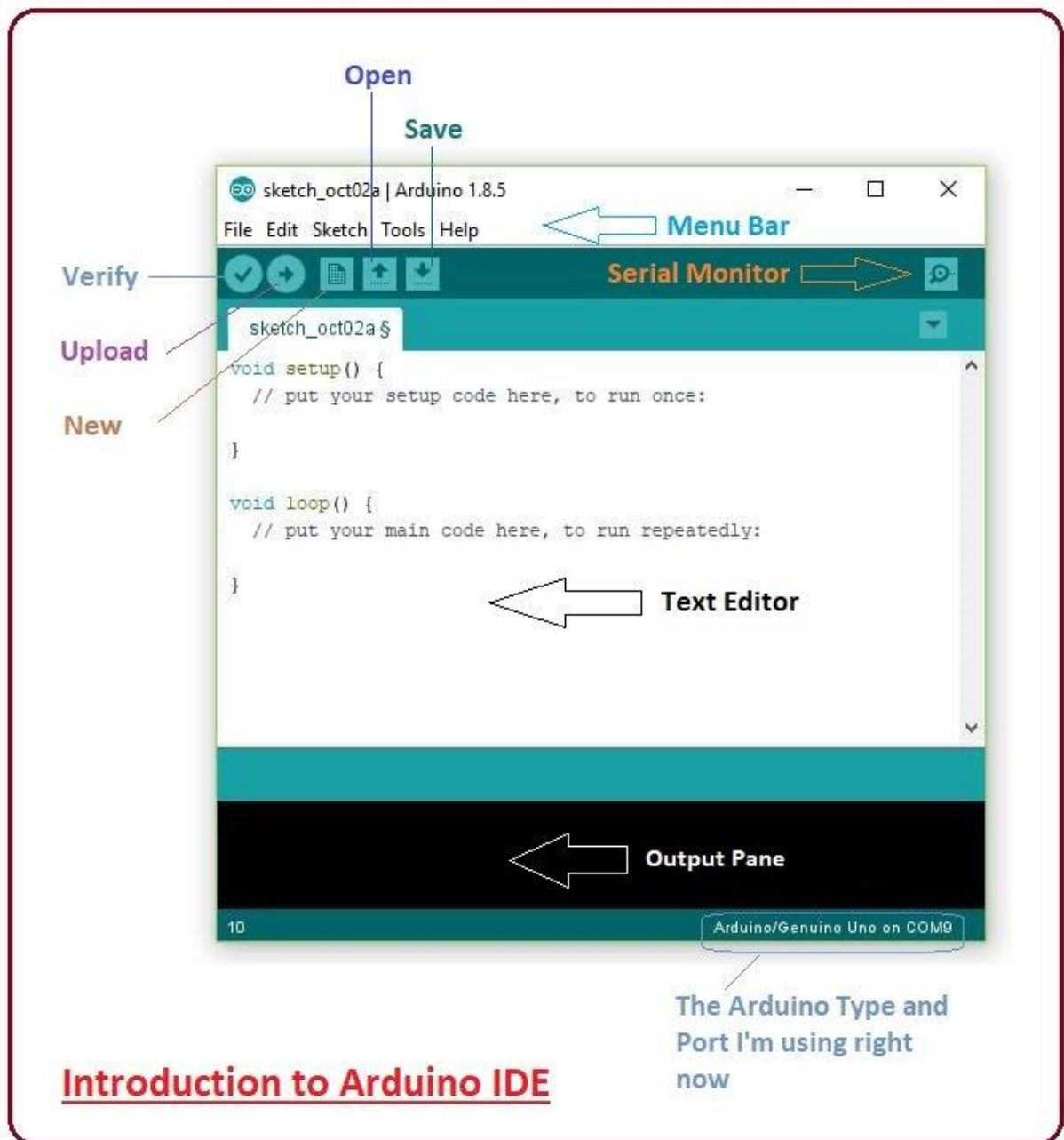




1. Doppelklick auf die heruntergeladene Datei
2. Sicherheitsabfrage → **Ja**
3. Alle Optionen **aktiviert lassen**
(Treiber sind wichtig!)
4. Installation abschließen
5. Arduino IDE starten

☒ Jetzt ist alles bereit

◆ Schritt 3: Die Arduino IDE verstehen (sehr wichtig!)



Die wichtigsten Bereiche:

1. Code-Fenster (Mitte)

Hier schreibst und liest du deinen Programmcode.

2. „Hochladen“-Button

- Kompiliert den Code
- Überträgt ihn auf das Arduino-Board

3. „Überprüfen“-Button

- Prüft nur den Code
- Lädt **nichts** hoch
- Ideal für Anfänger

🔑 4. Board & Port

- Hier wird ausgewählt:
 - welches Arduino-Board
 - welcher USB-Anschluss

◆ Schritt 4: Erstes Beispiel öffnen (empfohlen!)



 A screenshot of the Arduino IDE interface. The title bar reads 'Blink | Arduino 1.0.1'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar shows icons for opening files, saving, compiling, and uploading. The main text area displays the 'Blink' example code, which is a simple program to turn an LED on and off. The code is as follows:


```

/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.

 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialise the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
  
```

 At the bottom of the window, a status bar indicates '1' and 'Arduino Uno on COM4'.



Blink

Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct LED pin independent of which board is used. If you want to know what pin the on-board LED is connected to on your Arduino model, check the Technical Specs of your board at: <https://www.arduino.cc/en/Main/Products>

modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/Blink>

*/

// the setup function runs once when you press reset or power the board

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

// the loop function runs over and over again forever

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```



1. Menü **Datei** → **Beispiele**
2. **01.Basics**
3. **Blink** auswählen

● Das klassische Anfängerbeispiel:

- LED an
- LED aus
- immer wieder

◆ Schritt 5: Einen Sketch verstehen (didaktisch)

Ein Arduino-Programm heißt **Sketch**.

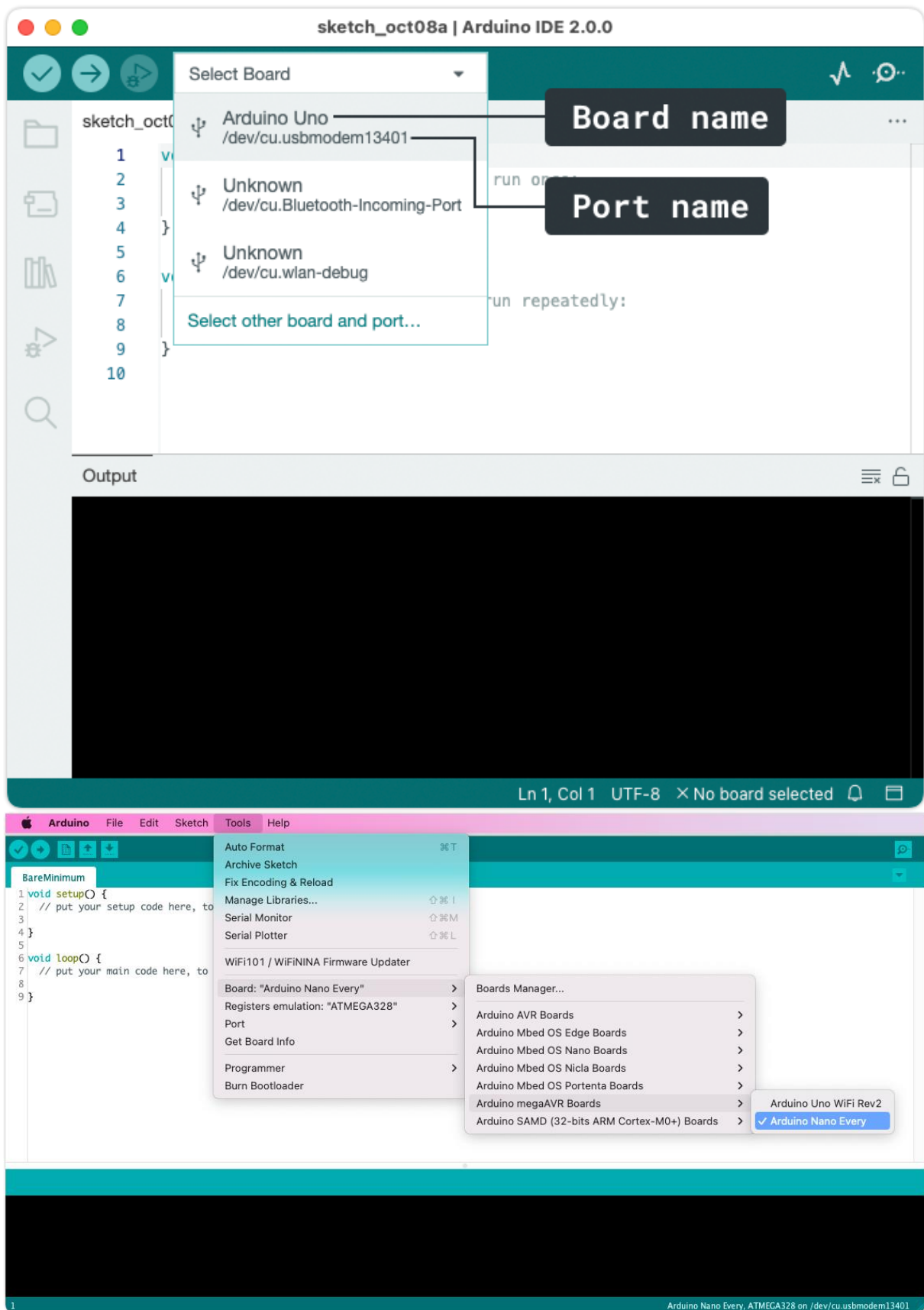
Jeder Sketch besteht aus 2 Pflichtteilen:

```
void setup() {  
  // wird EINMAL beim Start ausgeführt  
}  
  
void loop() {  
  // läuft IMMER wieder  
}
```

🧠 Merkhilfe:

- `setup()` = **Vorbereitung**
 - `loop()` = **Dauerschleife**
-

◆ Schritt 6: Board und Anschluss auswählen

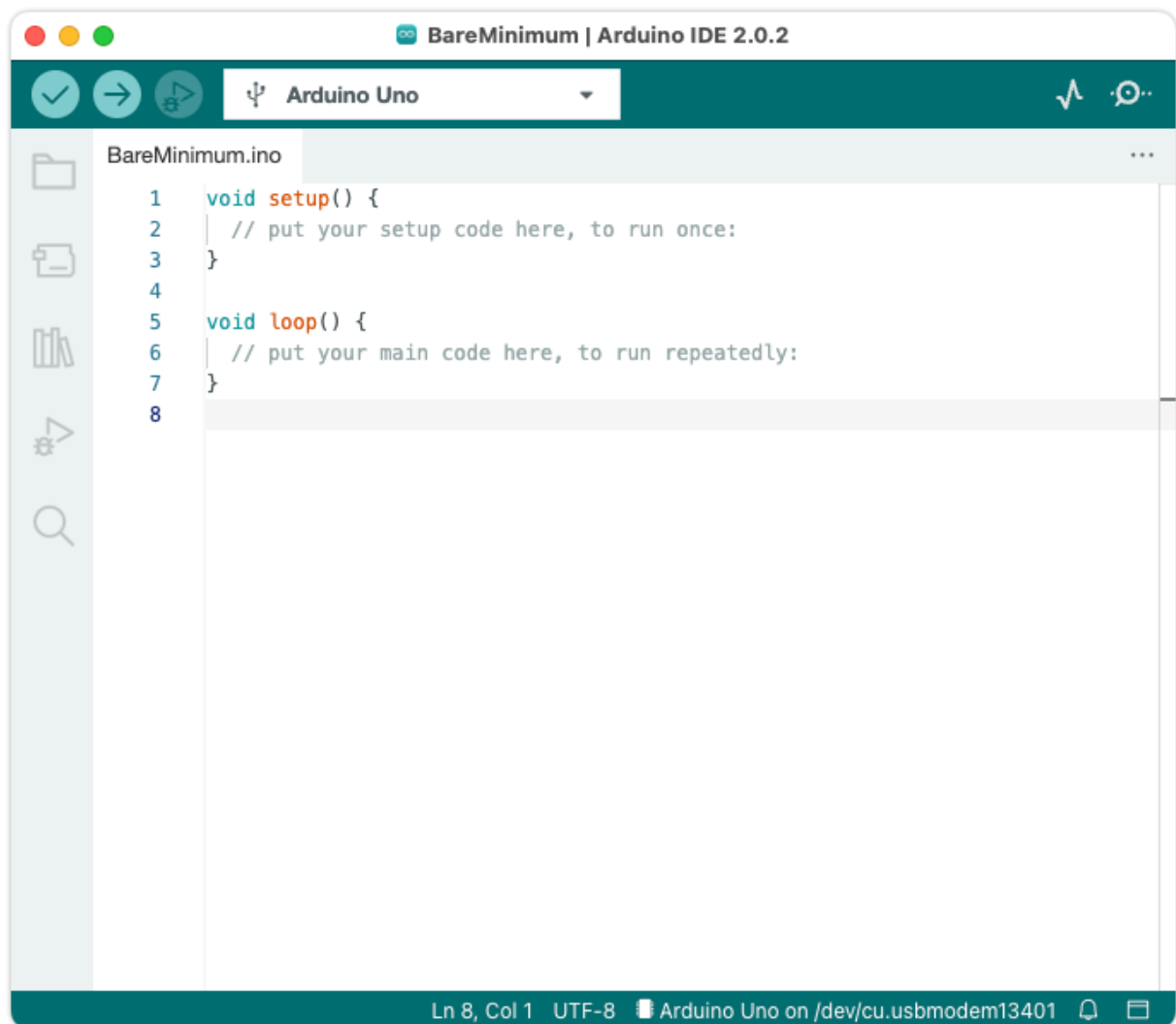


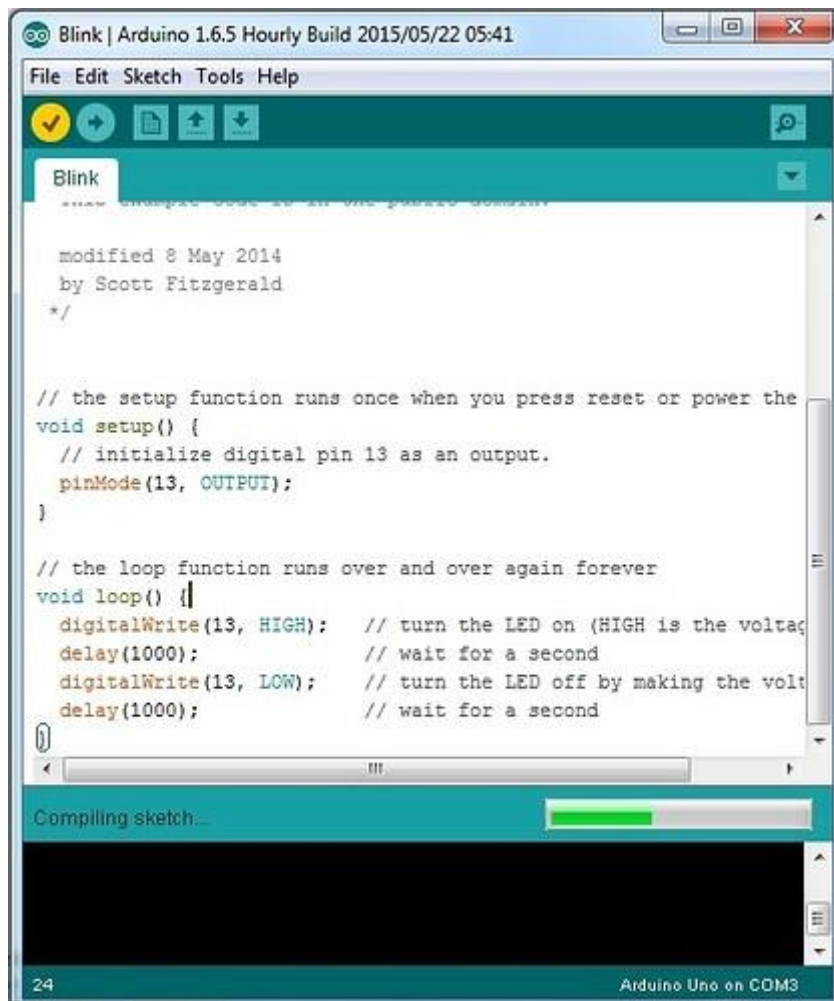
1. Arduino per USB anschließen
2. Menü **Werkzeuge** → **Board**
 - z. B. **Arduino Uno**
3. Menü **Werkzeuge** → **Port**

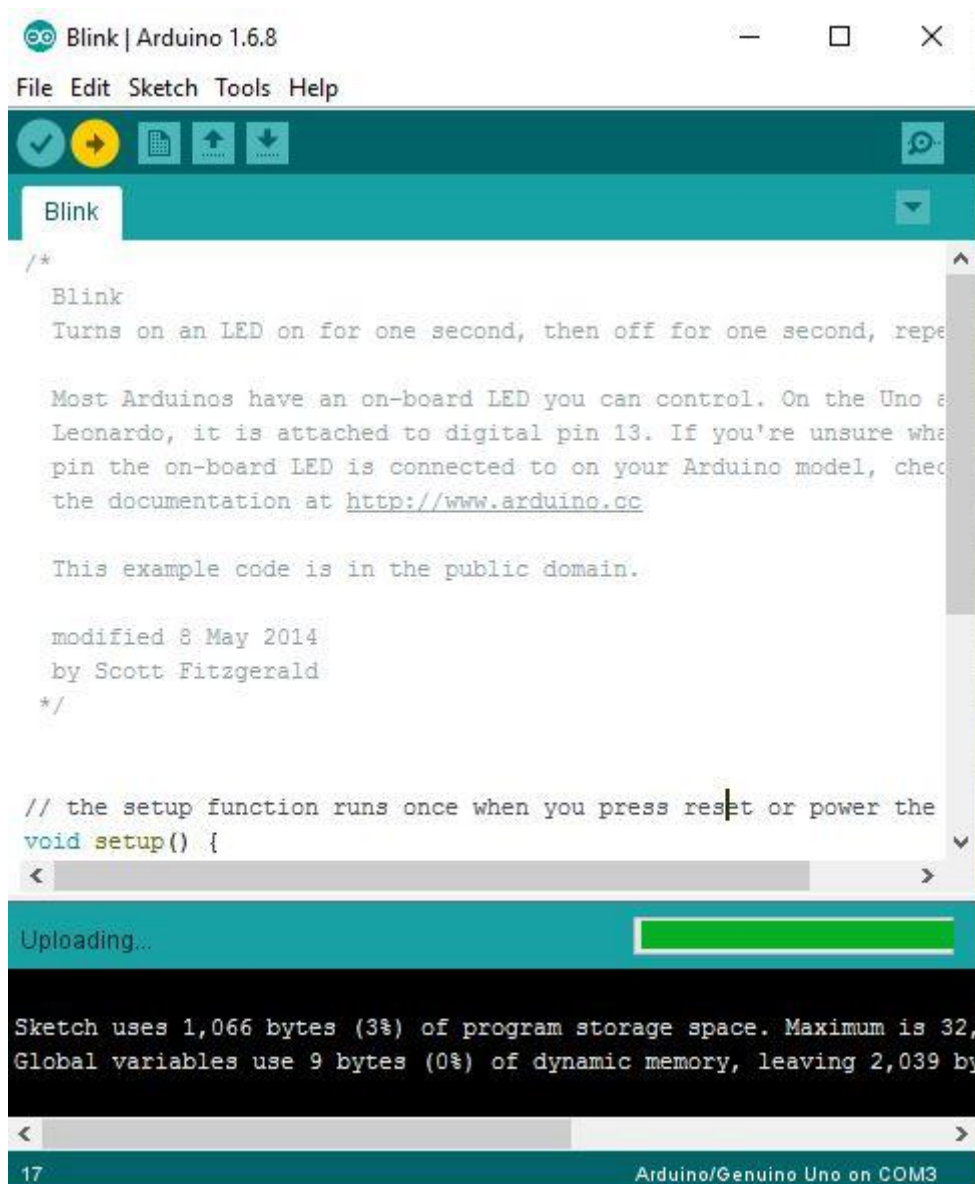
- COM-Port auswählen (meist nur einer sichtbar)

⚠ Ohne richtigen Port funktioniert nichts!

◆ Schritt 7: Sketch hochladen







1. Klick auf **Überprüfen** ✓
→ Fehler? Dann Meldung lesen!
2. Klick auf **Hochladen** ▶
3. Warten ...
4. Meldung unten:
„Upload abgeschlossen“

🎉 Der Code läuft jetzt auf dem Arduino

◆ Schritt 8: Typische Anfängerfragen (ehrlich beantwortet)

❓ „Ich verstehe den Code nicht“

→ Normal!

Arduino lernt man **durch Ausprobieren**, nicht durch Lesen allein.

? „Ich habe etwas kaputt gemacht“

→ Nein.

Du kannst jederzeit:

- den Sketch schließen
- ein Beispiel neu öffnen

? „Es kommt eine rote Fehlermeldung“

→ Meist:

- Klammer vergessen
- falsches Board
- falscher Port

Wichtige Grundregeln für Anfänger

✓ Nur eine Sache auf einmal ändern

✓ Beispiele nutzen

✓ Fehlermeldungen lesen

✗ Nicht alles gleichzeitig verstehen wollen
