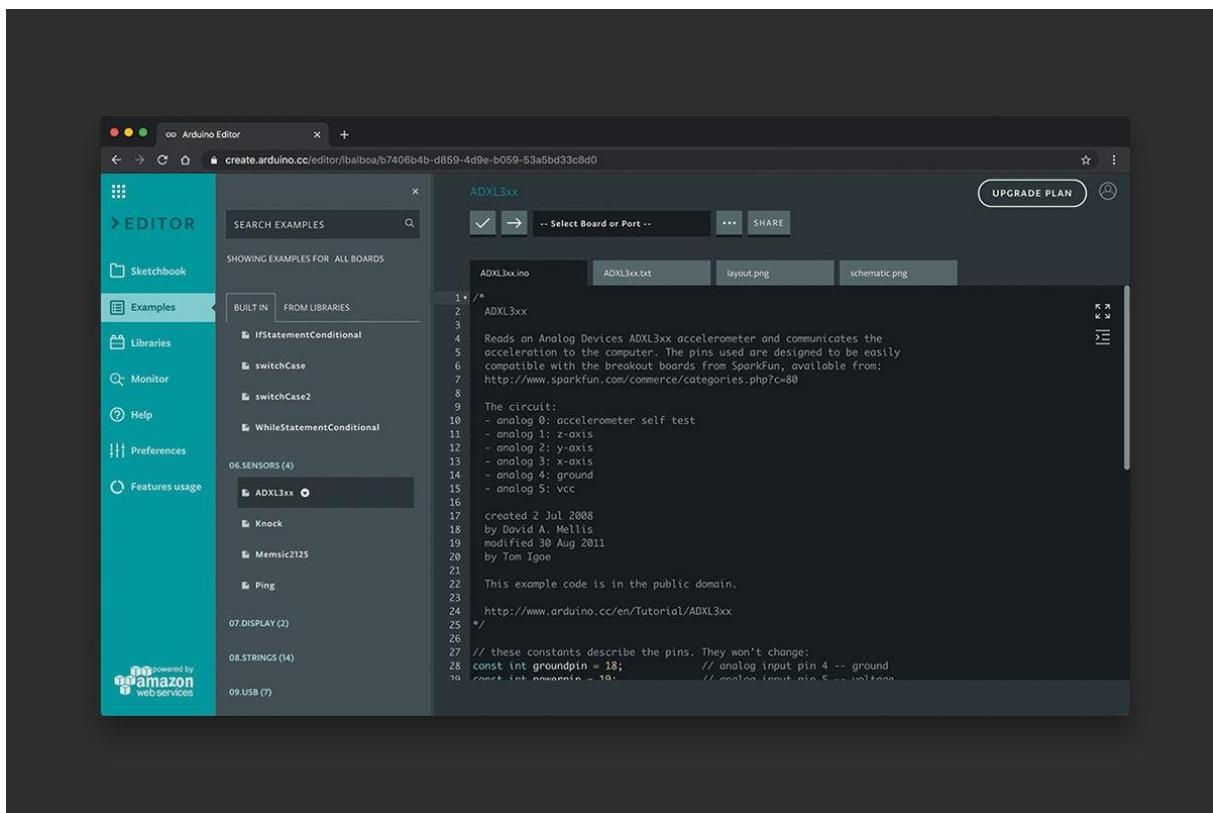
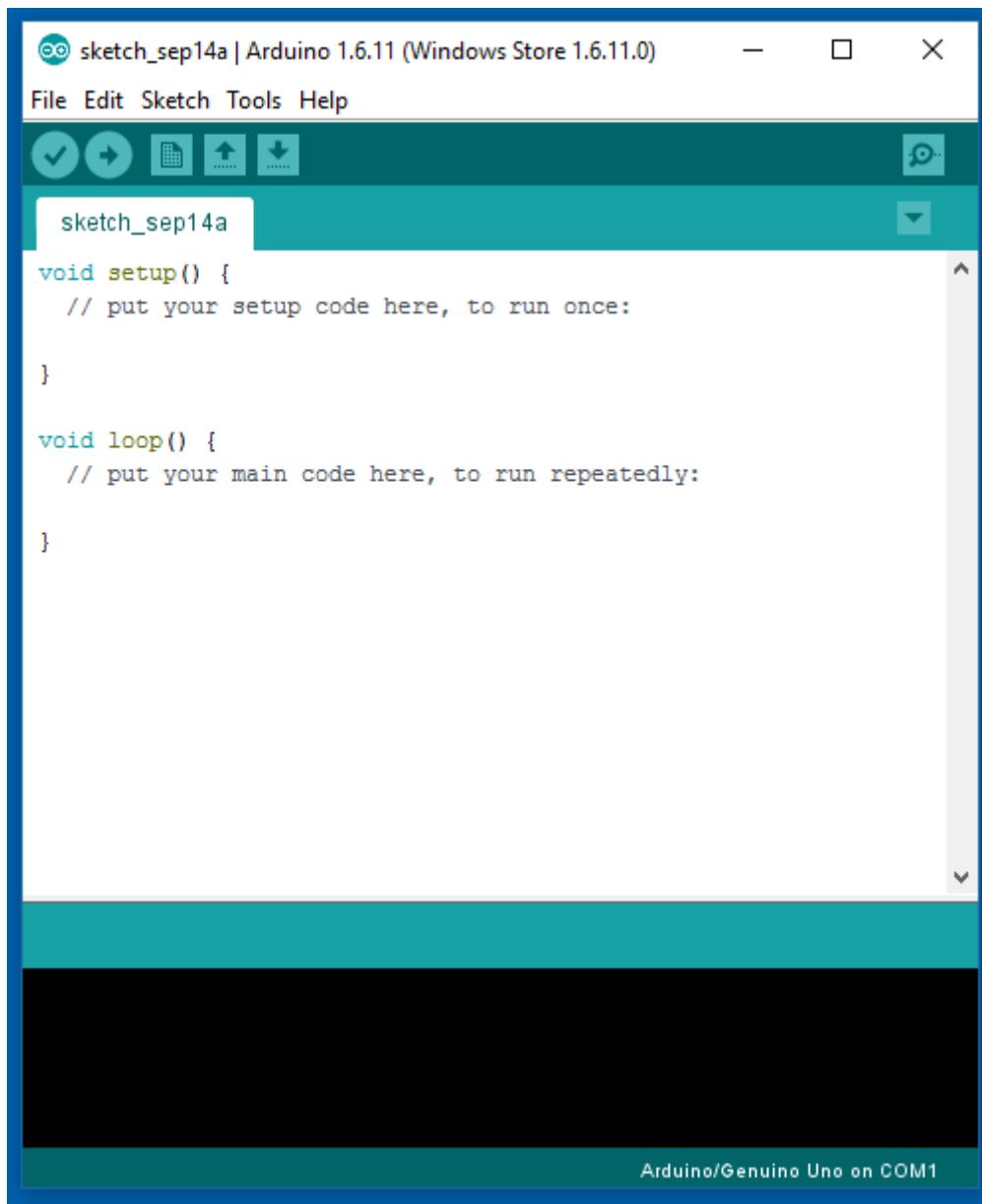


Schritt 1: Arduino IDE herunterladen

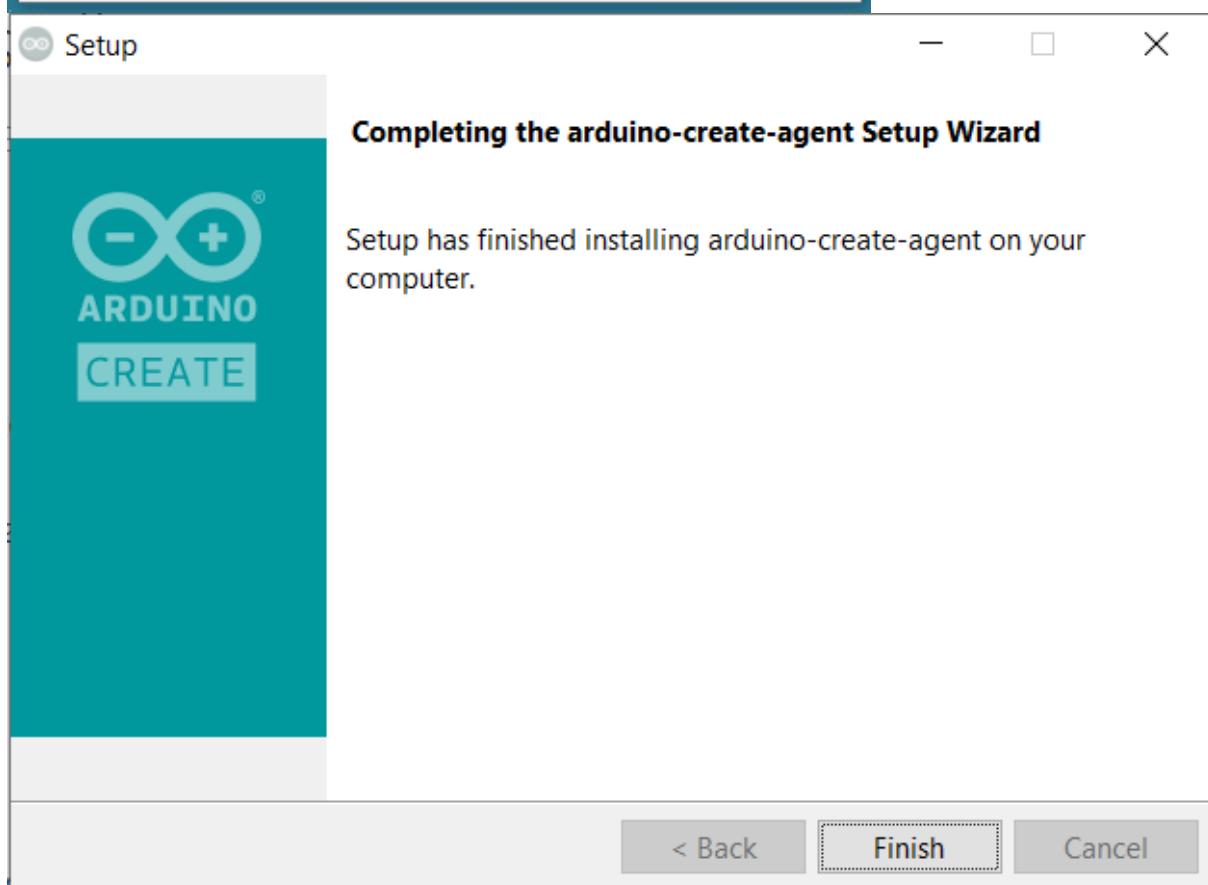
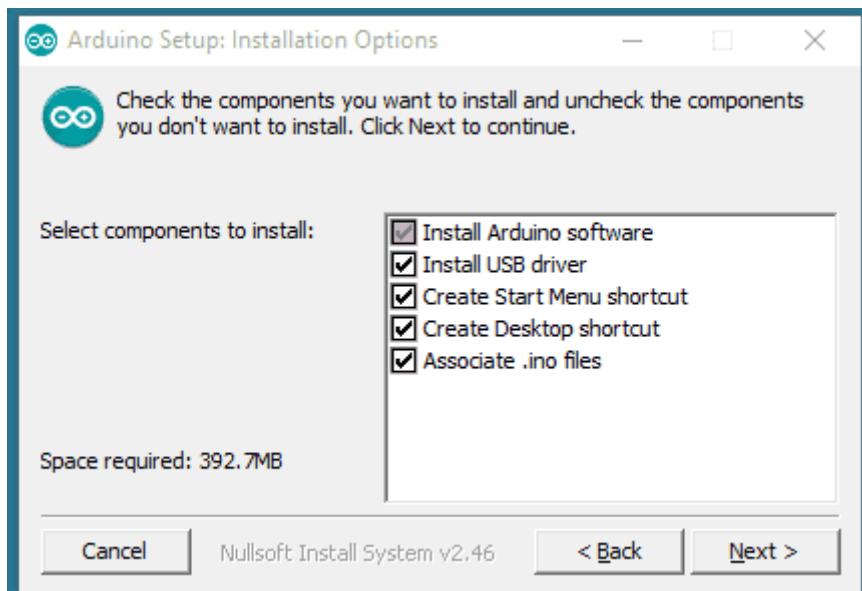


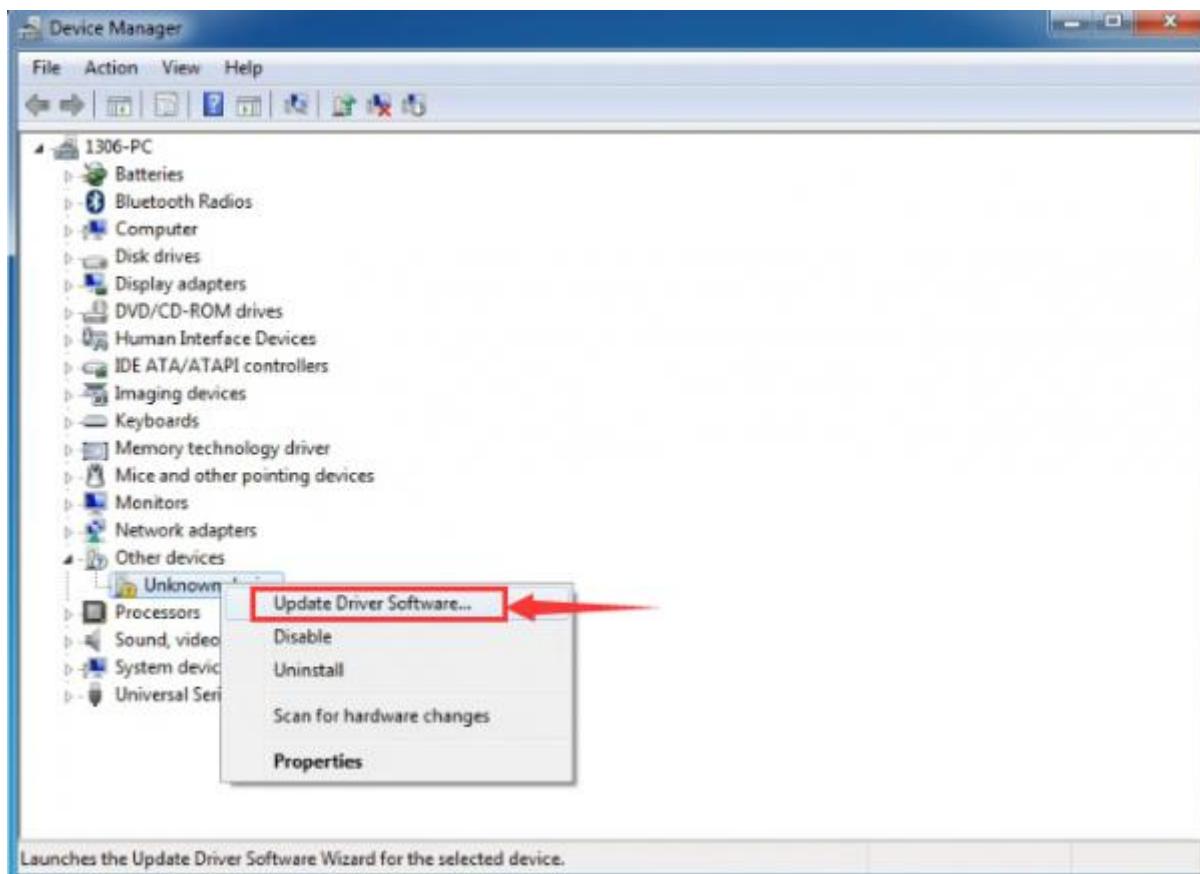


1. Öffne einen Webbrowser
2. Gehe auf die offizielle Arduino-Seite:
👉 <https://www.arduino.cc/en/software>
3. Wähle Windows
4. Klicke auf Just Download
(Spenden sind freiwillig)

⬇️ Die Installationsdatei wird heruntergeladen

◆ Schritt 2: Arduino IDE installieren

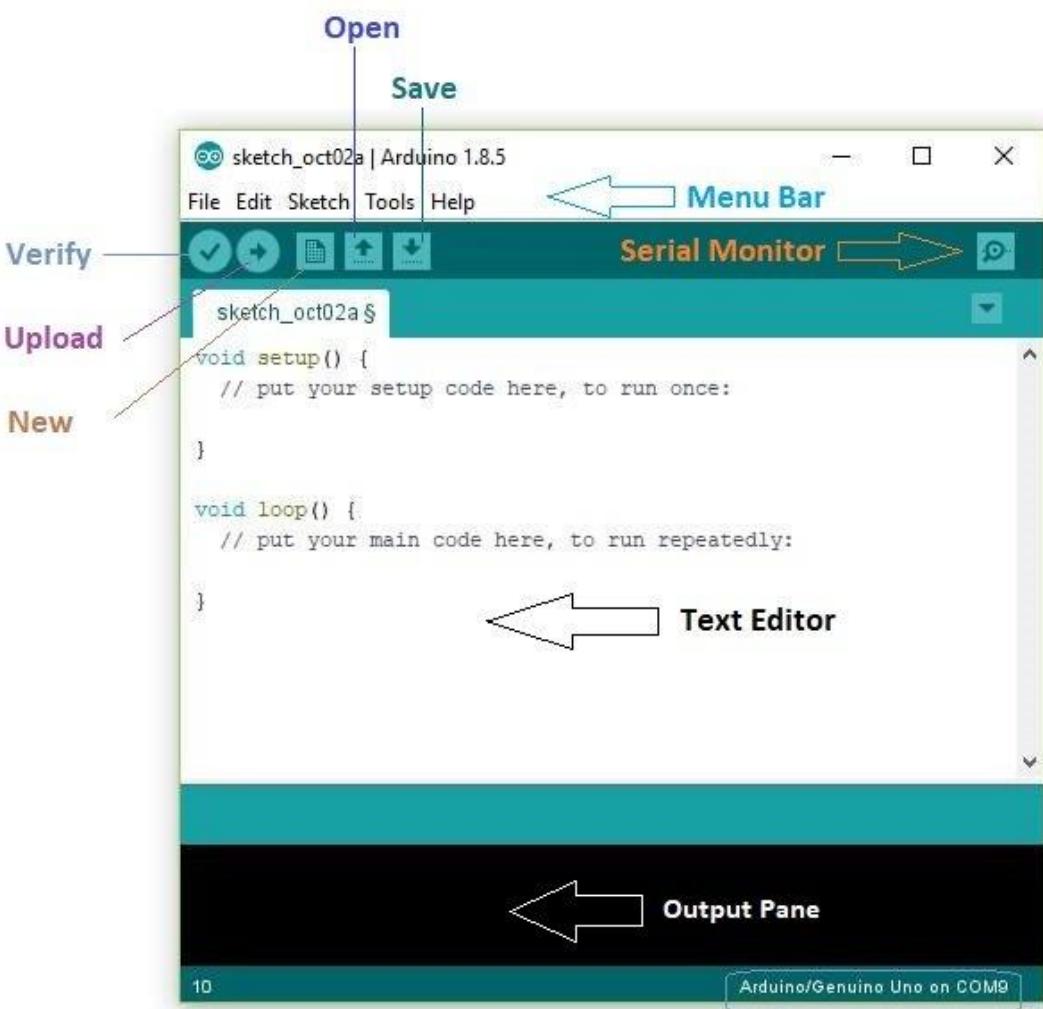




1. Doppelklick auf die heruntergeladene Datei
2. Sicherheitsabfrage → **Ja**
3. Alle Optionen **aktiviert lassen**
(Treiber sind wichtig!)
4. Installation abschließen
5. Arduino IDE starten

Jetzt ist alles bereit

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



Introduction to Arduino IDE

The Arduino Type and
Port I'm using right
now

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!)



The screenshot shows the Arduino IDE interface. The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for back, forward, file operations, and a search function. The title bar says "Blink". The main area contains the code for the Blink sketch, which controls the built-in LED on most Arduinos. The code is as follows:

```
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
```

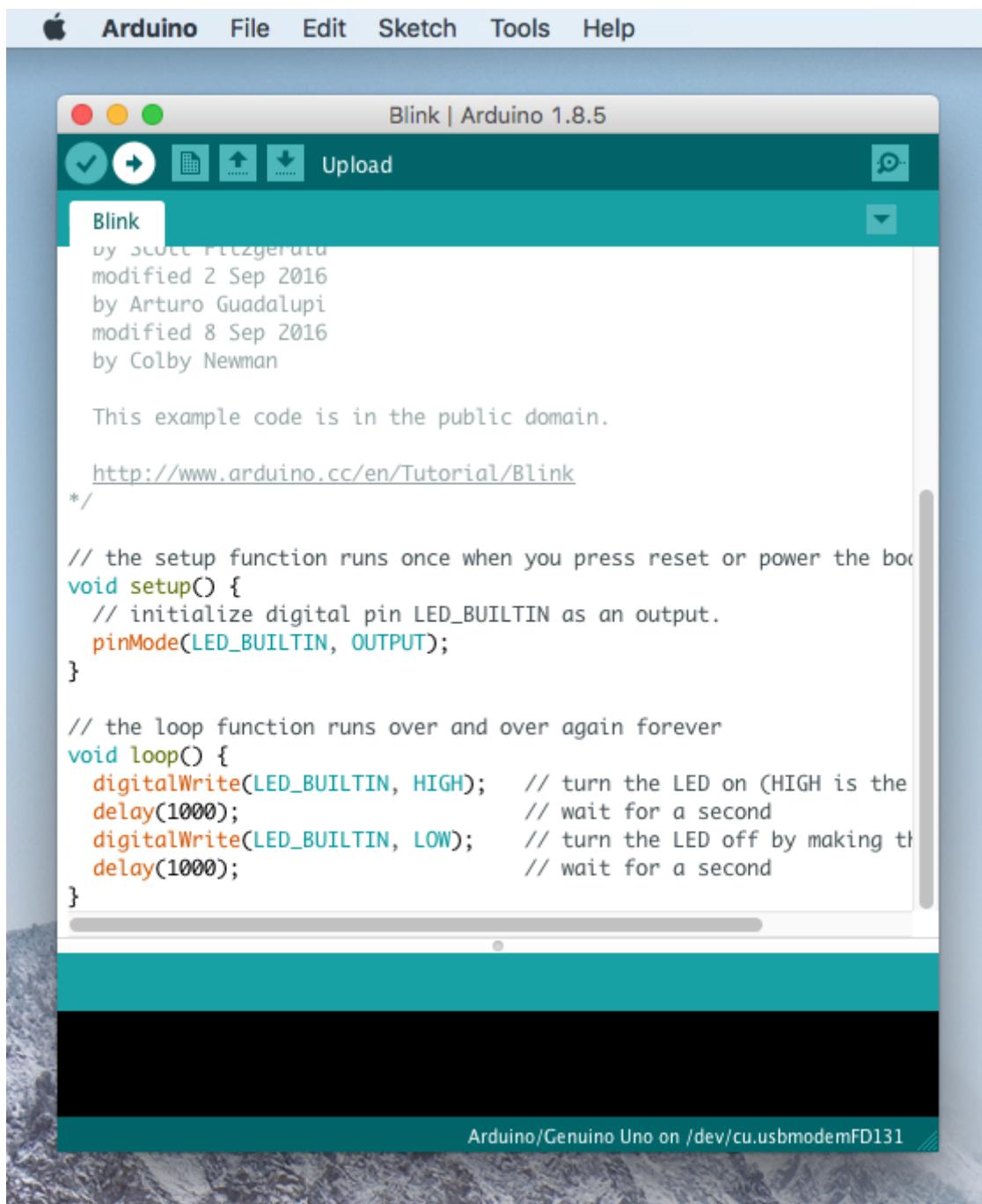
```
modified 8 May 2014  
by Scott Fitzgerald  
modified 2 Sep 2016  
by Arturo Guadalupi  
modified 8 Sep 2016  
by Colby Newman
```

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  
}
```

At the bottom right of the code editor, it says "Arduino/Genuino Uno on /dev/ttyUSB0".



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**.

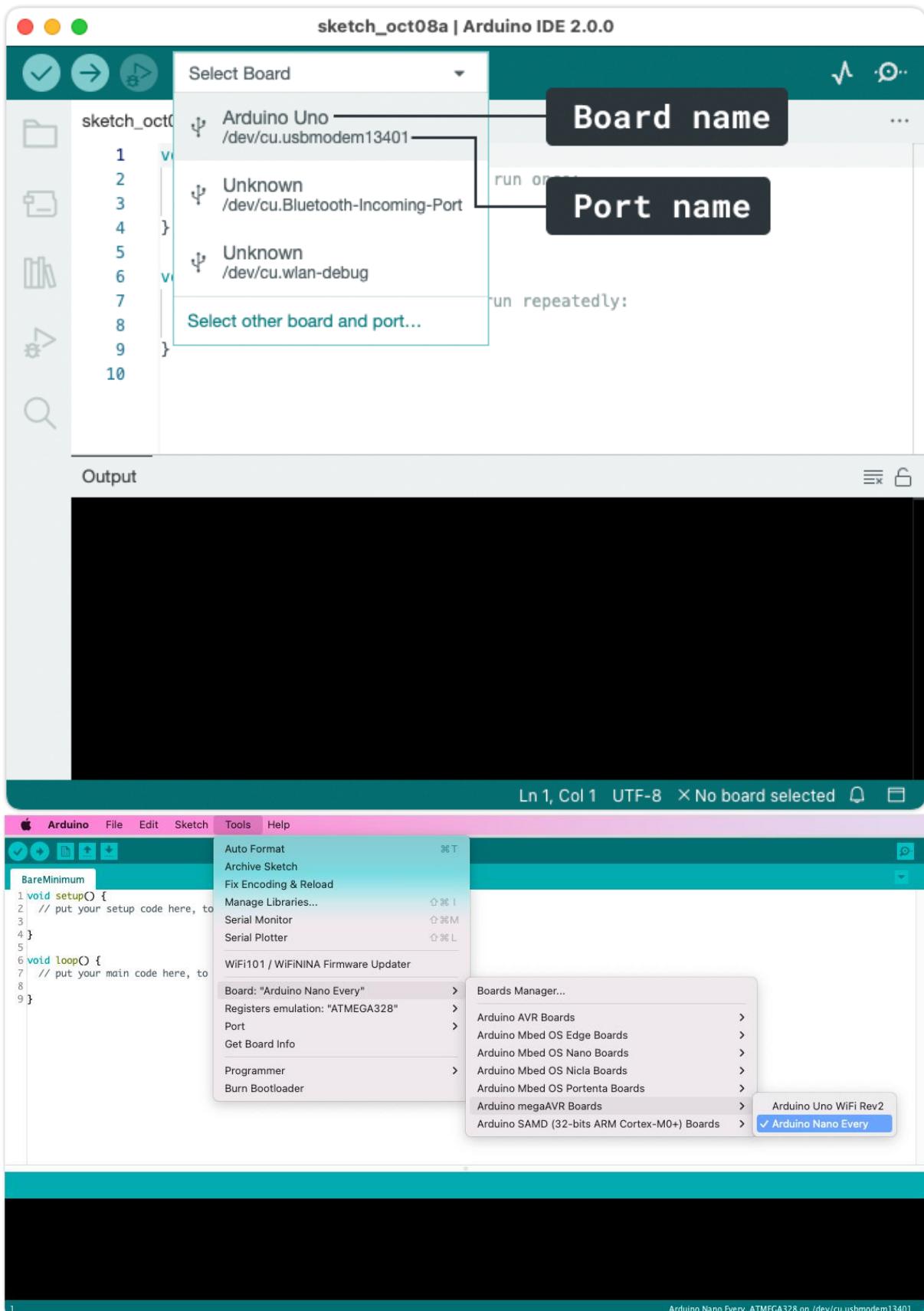
Jeder Sketch besteht aus 2 Pflichtteilen:

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

 Merkhilfe:

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

◆ Schritt 6: Board und Anschluss auswählen

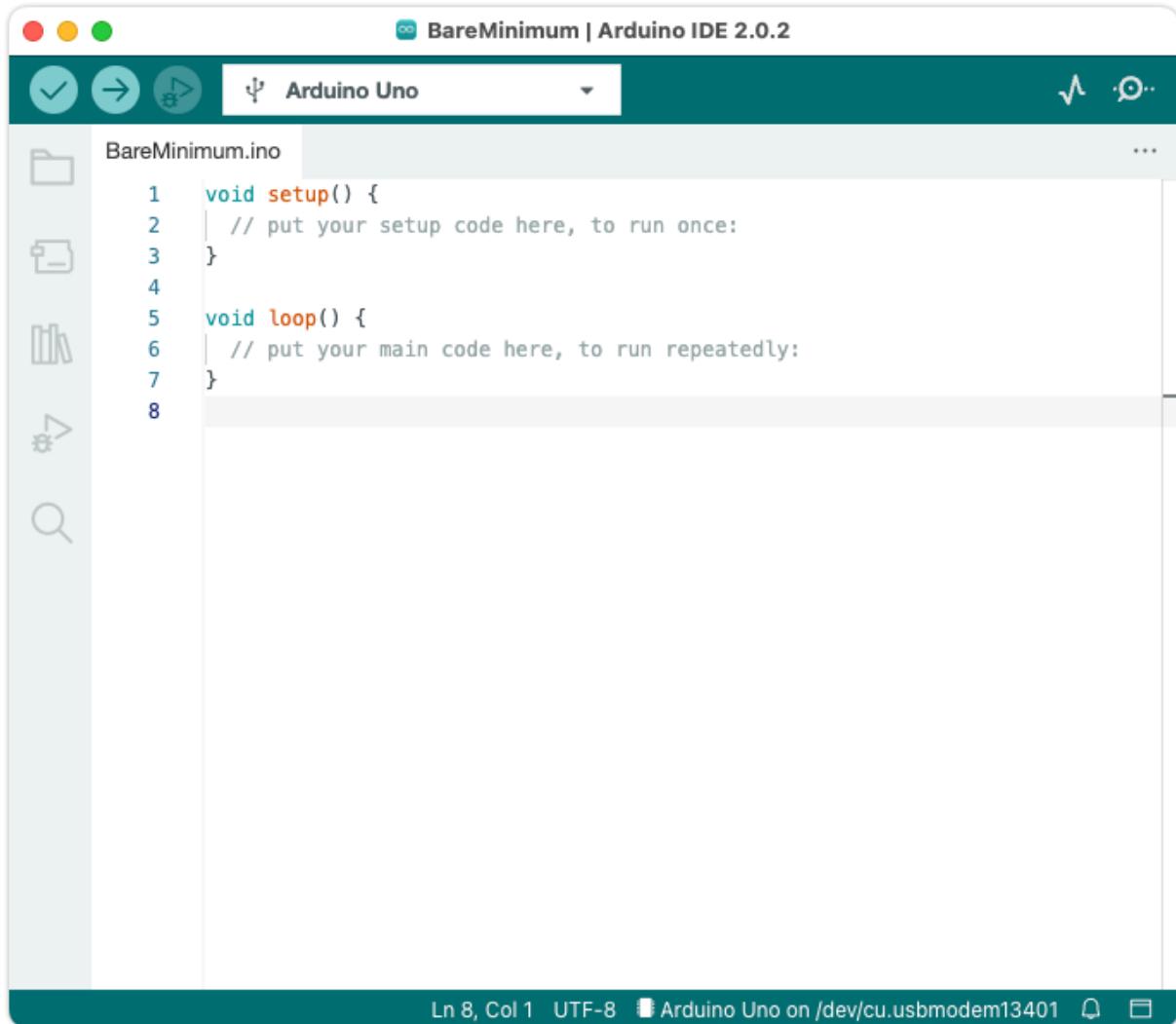


1. Arduino per USB anschließen
2. Menü **Werkzeuge** → **Board**
 - z. B. **Arduino Uno**
3. Menü **Werkzeuge** → **Port**

- COM-Port auswählen (meist nur einer sichtbar)

 Ohne richtigen Port funktioniert nichts!

◆ Schritt 7: Sketch hochladen



The screenshot shows the Arduino IDE interface with the title bar "BareMinimum | Arduino IDE 2.0.2". The central code editor window displays the "BareMinimum.ino" file. The code contains the standard bare minimum sketch:

```
1 void setup() {
2     // put your setup code here, to run once:
3 }
4
5 void loop() {
6     // put your main code here, to run repeatedly:
7 }
8
```

The status bar at the bottom indicates "Ln 8, Col 1 UTF-8" and "Arduino Uno on /dev/cu.usbmodem13401".

Blink | Arduino 1.6.5 Hourly Build 2015/05/22 05:41

File Edit Sketch Tools Help

Blink

```
modified 8 May 2014
by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // turn the LED off by making the volt
    delay(1000);                // wait for a second
}
```

Compiling sketch...

24 Arduino Uno on COM3

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeating.  
  
Most Arduinos have an on-board LED you can control. On the Uno and Leonardo, it is attached to digital pin 13. If you're unsure what pin the on-board LED is connected to on your Arduino model, check the documentation at http://www.arduino.cc  
  
This example code is in the public domain.  
  
modified 8 May 2014  
by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
}  
  
Uploading...  
  
Sketch uses 1,066 bytes (3%) of program storage space. Maximum is 32,256 bytes.  
Global variables use 9 bytes (0%) of dynamic memory, leaving 2,039 bytes free.  
  
17 Arduino/Genuino Uno on COM3
```

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
