

## 4.0.3. Die Entwicklungsumgebung

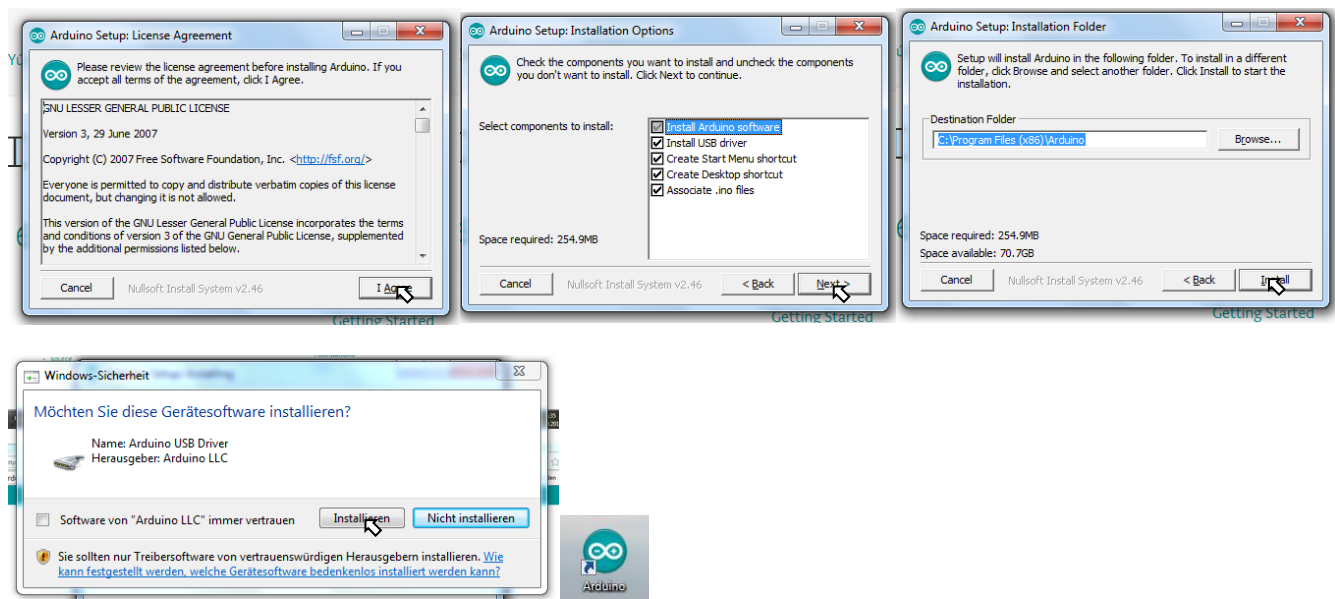
### 4.0.3.1. Herunterladen und Installieren der Entwicklungsumgebung

Das Arduino Board und die Entwicklungsumgebung wurde 2005 am **Interaction Design Institute Ivrea** in der Nähe von Turin von **Massimo Banzhi** und **David Cuartielles** entwickelt, um den Studenten einfaches und billiges Lernmaterial zur Verfügung zu stellen. **Arduino** war der Name eines mittelalterlichen Burgherren der Stadt Ivrea.

Unter dem Suchbegriff **Arduino IDE download** findet man mit einer Suchmaschine wie z.B. **Google** die Downloadseite für die aktuelle Version, z.B. 1.0.6. Je nach Betriebssystem wählt man die passende **Installationsdatei** wie z.B. den **windows installer** aus, lädt sie auf den eigenen Computer herunter und führt sie aus. Für die Installation von Programmen benötigt man in der Regel **Administratorrechte**.



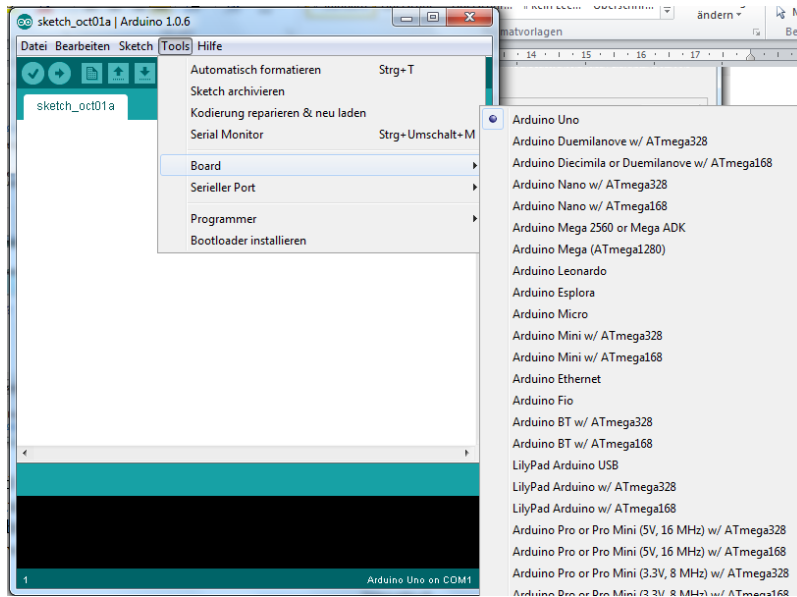
Man folgt allen Anweisungen bzw. übernimmt alle Vorschläge einschließlich der Installation des Treibers (bei MAC OS und Linux nicht notwendig, da schon vorhanden) und erkennt zum Schluss das Symbol auf der Arbeitsfläche:



### 4.0.3.2. Anschluss des Arduino-Boards

Nun wird das Arduino-Board über USB-Kabel mit dem PC verbunden. In der Regel wird die Erkennung bzw. die Initialisierung des Treibers auf der Statusleiste angezeigt. Die **grüne LED (PWR)** zeigt die **Stromversorgung** an. Bei erfolgreicher **Erkennung** leuchtet die **gelbe LED (L)** auf PIN 13 im Sekundentakt. Als **Test** wird nämlich standardmäßig das Blinkprogramm von Seite 3 an das Board übertragen.

Zur Kontrolle öffnen wir die Arduino-IDE und überprüfen die Auswahl des **Boards**:

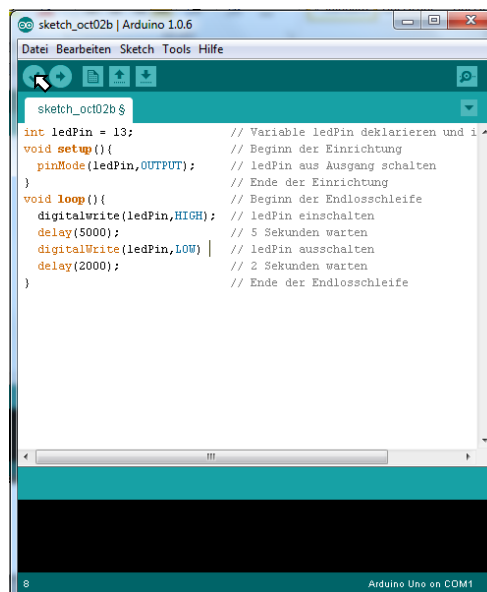
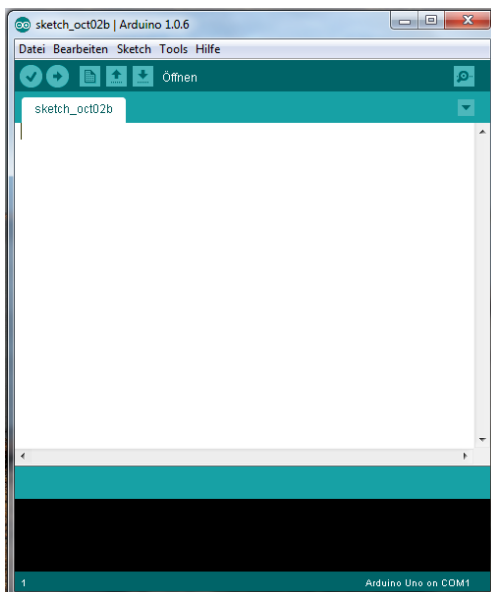



Anschließend kontrollieren wir die **serielle Port-Adresse**, d.h. die Nummer des verwendeten usb-Ausgangs. Unter Windows sollte es **COM ?** heißen, unter MAC OS und Linux **/dev/tty.usbmodem?????**. Die Fragezeichen stehen für die Ziffern, die der Computer seinen Anschlüssen selbständig zuordnet.

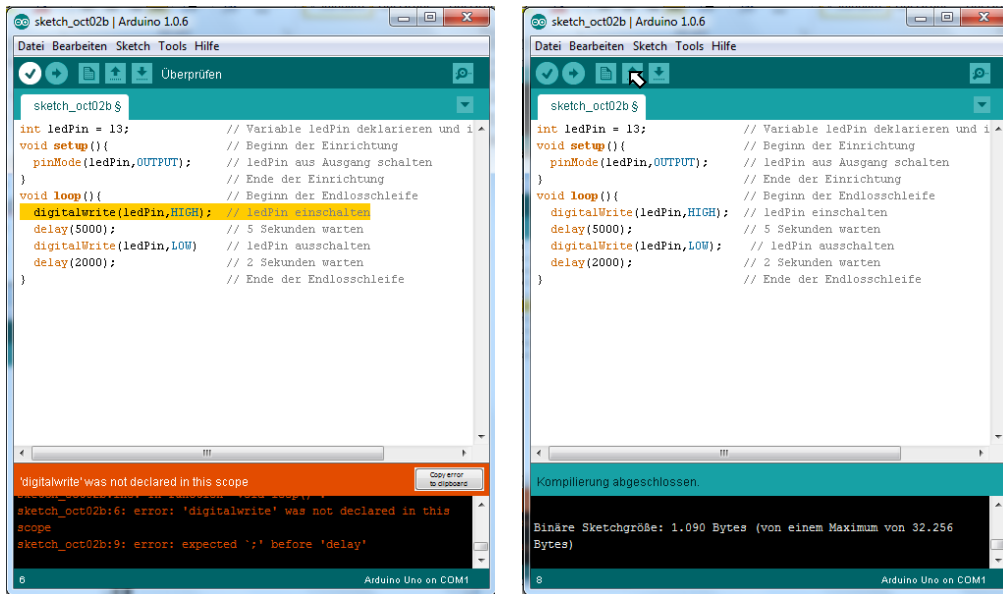
### 4.0.3.3. Der erste Sketch


Zum Testen des Ablaufs öffnen wir die IDE und schreiben den Blinkskech in leicht veränderter Form in das Fenster:

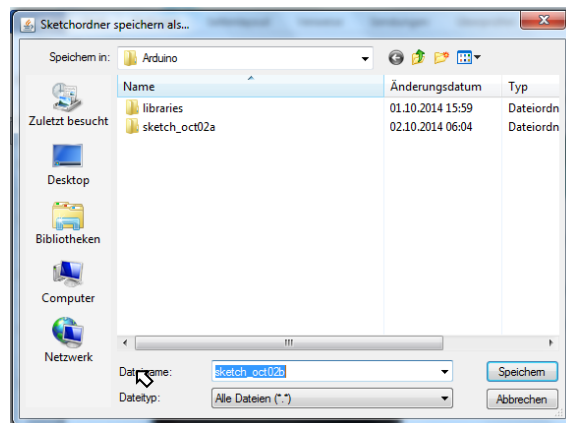
```
int ledPin = 13; // Variable für Pin 13 initialisieren
void setup() { // Beginn der Variablendefinition
    pinMode(ledPin, OUTPUT); // Pin 13 als Ausgang definieren
} // Ende der Variablendefinition
void loop() { // Beginn der Endlosschleife
    digitalWrite(ledPin, HIGH); // LED auf High-Pegel (5V)
    delay(5000); // Eine Sekunde warten
    digitalWrite(ledPin, LOW); // LED auf Low-Pegel (0V)
    delay(2000); // Eine Sekunde warten
} // Ende der Endlosschleife
```




Der Editor markiert alle bekannten Schlüsselwörter farbig. Jetzt schon verdächtig ist die fehlende Markierung bzw. Erkennung des ersten `digitalwrite`-Befehls. Wir klicken auf den Haken  links oben und lassen den Sketch probierhalber kompilieren und damit überprüfen:



Die beiden Fehler (Grossbuchstabe bei `digitalwrite`) und im Fall des fehlenden Semikolons auch die erwartete Verbesserung werden im Kommentarfenster unten angezeigt. Nachdem wir die Fehler korrigiert und nochmal kompiliert haben, erscheint die Meldung „Kompilierung abgeschlossen“. Nun speichern wir den Sketch mit  auf dem eigenen usb-stick in einem neuen Ordner Arduino:



Zum Schluss übertragen wir mit  das kompilierte Maschinenprogramm an das Board. Die LED sollte nun immer abwechselnd 5 Sekunden an und 2 Sekunden aus sein.