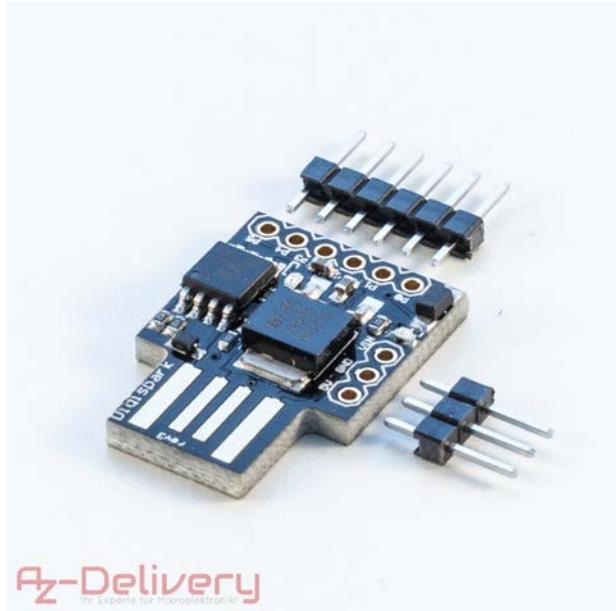


## Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery ATTiny85 - Digispark Rev.3!  
Auf den folgenden Seiten gehen wir mit dir gemeinsam die ersten Schritte von der Einrichtung bis zum Programmieren eines eigenen Scriptes.  
Viel Spaß!



Der Digispark von der Firma Digistump ist wohl der kompakteste und kleinste Arduino den es aktuell gibt. Die Platine misst gerade mal 17,5mm x 19mm (ohne USB-Anschluss) (27mmx19mm mit USB). Er hat einen USB-Stecker direkt mit on Board und kann ohne weiteres Kabel direkt in einen USB-Anschluss gesteckt werden. Der integrierte ATTiny85 wird direkt von dem USB-Anschluss mit Spannung versorgt. Außerdem besitzt der Digispark 6 freiprogrammierbare GPIO Anschlüsse, die mit der bekannten Arduino Programmierumgebung programmiert werden können. Für die eigenen Programme stehen 6kB Speicher zur Verfügung. Neben den 3 weiteren Pins auf der Platine, die für eine Externe Spannungsversorgung genutzt werden können, unterstützt der ATTiny85 volle SPI und i2c Schnittstellen. Der „Arduino“ in größer eines 1-Euro Stückes bringt also schon sehr viel mit. Dann beginnen wir gleich mal mit dem Einrichten und Programmieren.

Digispark ATTiny85 Schaltbild:

<https://s3.amazonaws.com/digispark/DigisparkSchematicFinal.pdf>

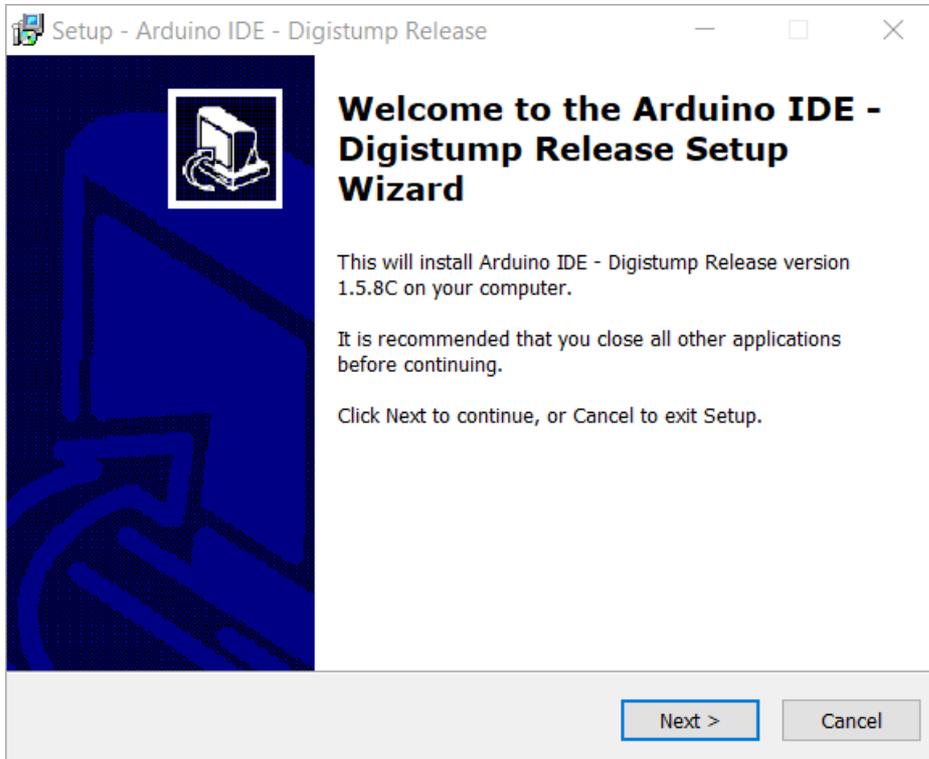
Treiber und Arduino-Software:

<https://github.com/digistump/DigistumpArduino/releases/download/v1.5.8C/DigistumpArduinoInstall1.5.8C.exe>

Die Arduino-Software von Digistump enthält alles was für den Digispark benötigt wird.

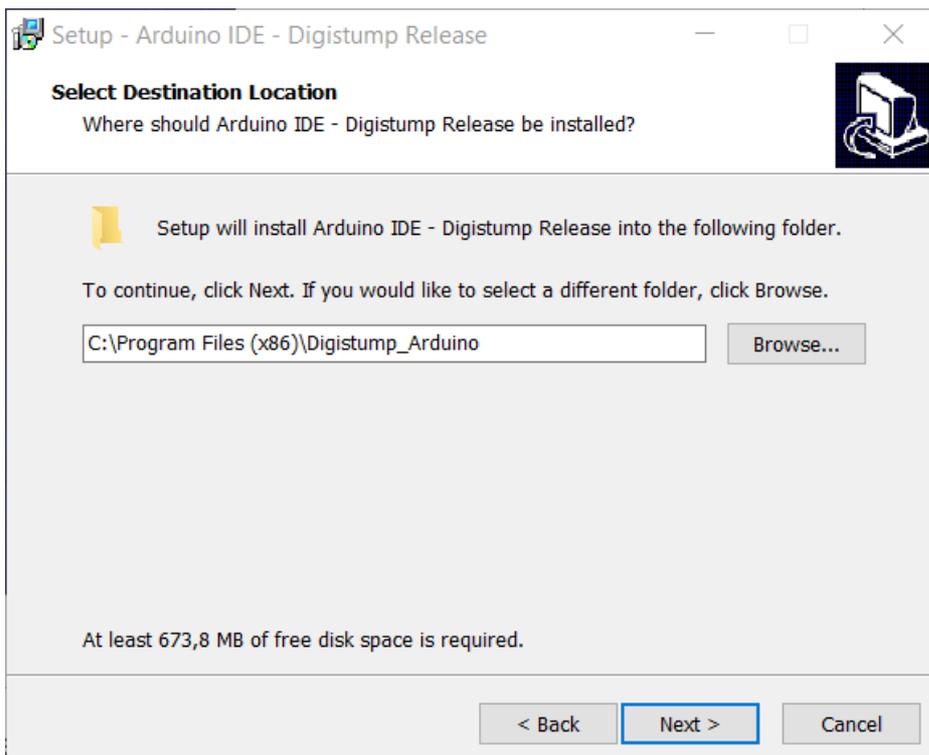
## Installation der Software:

Nach dem Download und Starten des Installers erscheint folgender Startbildschirm:

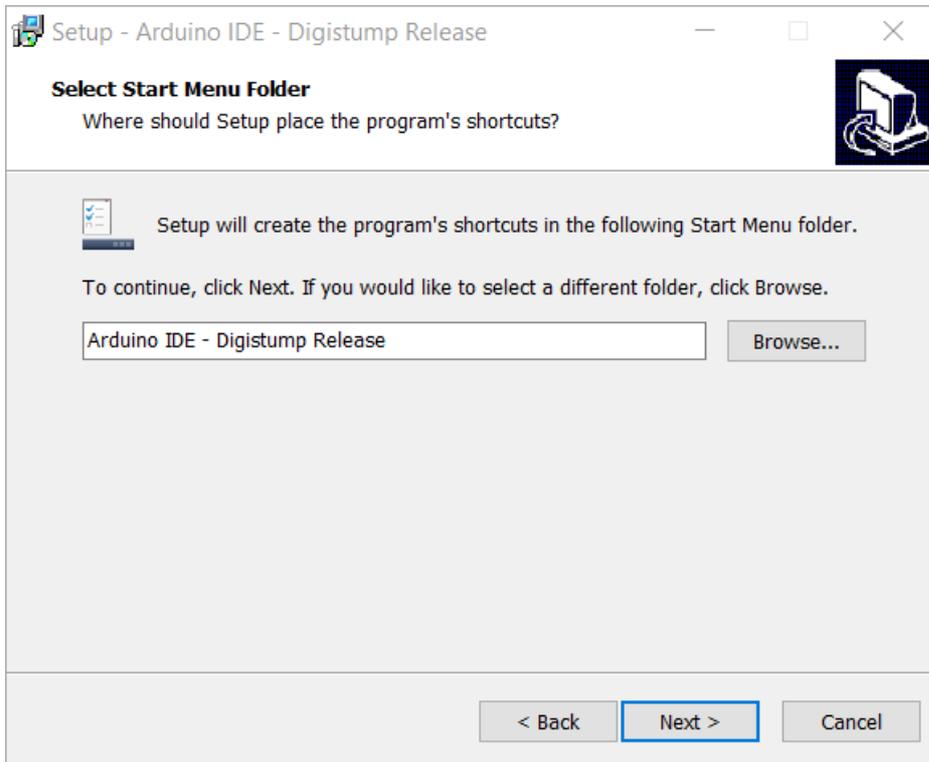


Hier einfach auf „**Next >**“ klicken.

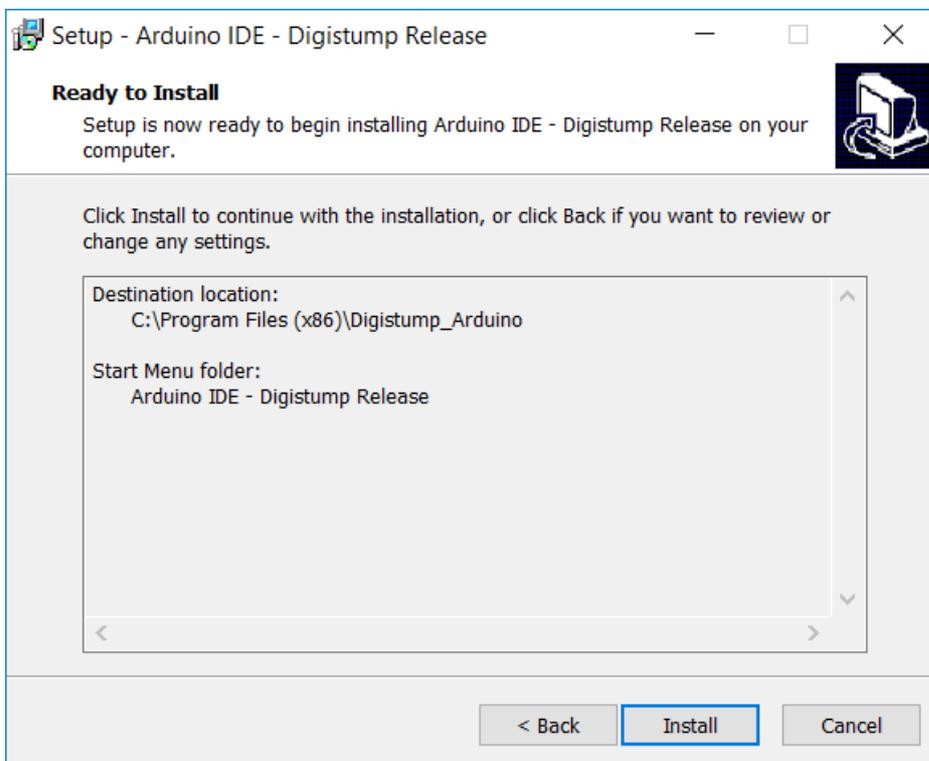
Im nächsten Fenster kann der Speicherort der Software geändert werden. In der Regel kann man hier den Vorschlag ungeändert übernehmen und **Next** klicken:



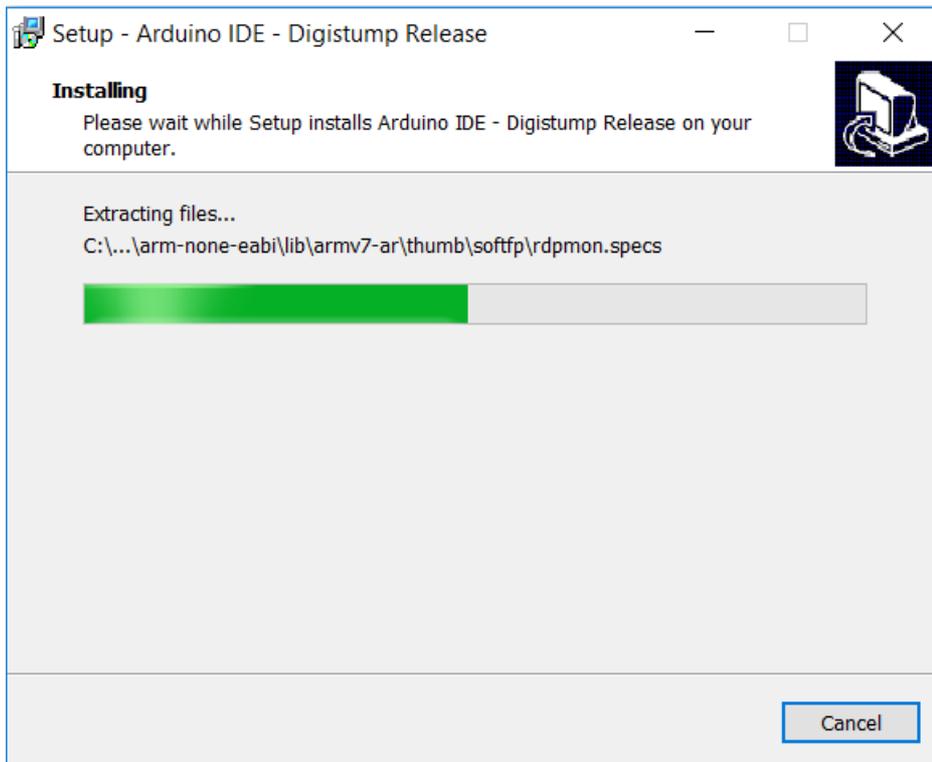
Ebenso bei der Auswahl des Startmenüeintrages:



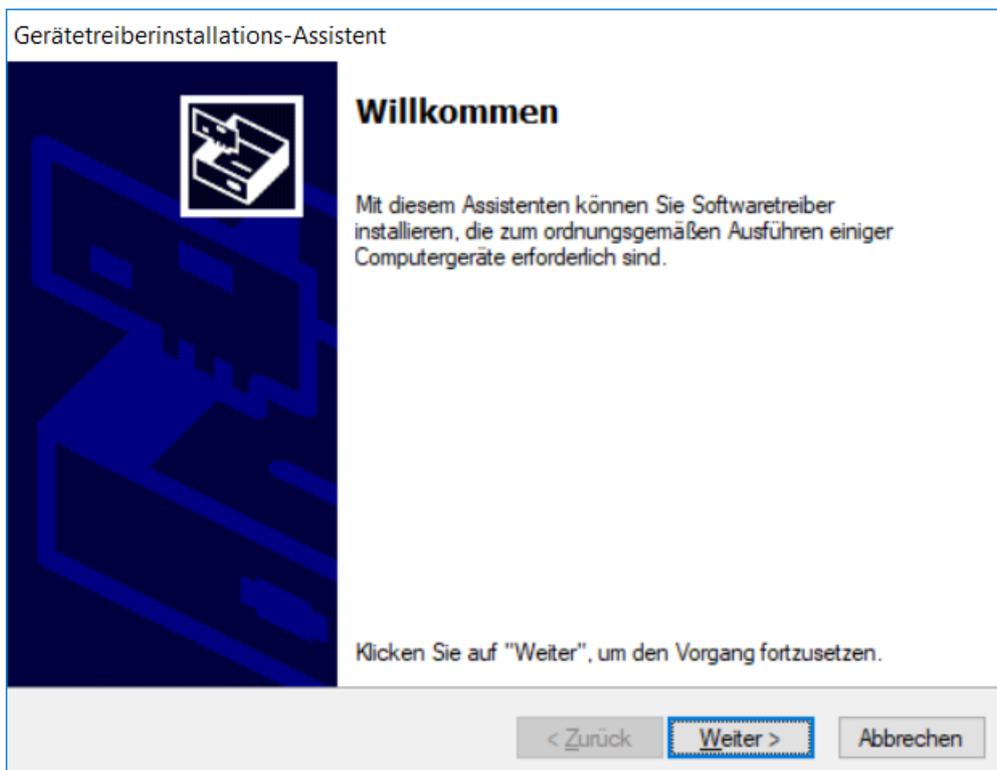
Falls gewünscht, kann im nächsten Fenster ein Desktopsymbol erstellt werden. Und zuletzt wählt man **Install** aus.



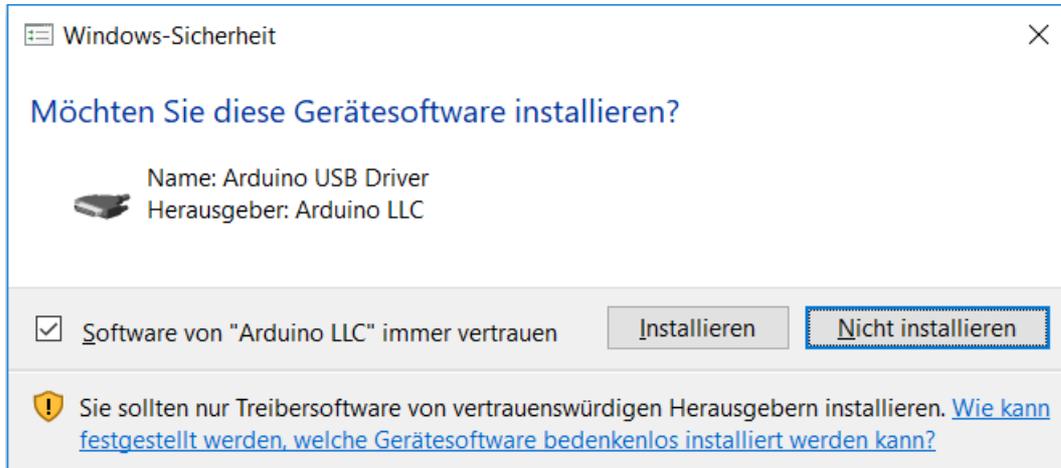
Die Installation dauert einen kurzen Moment...



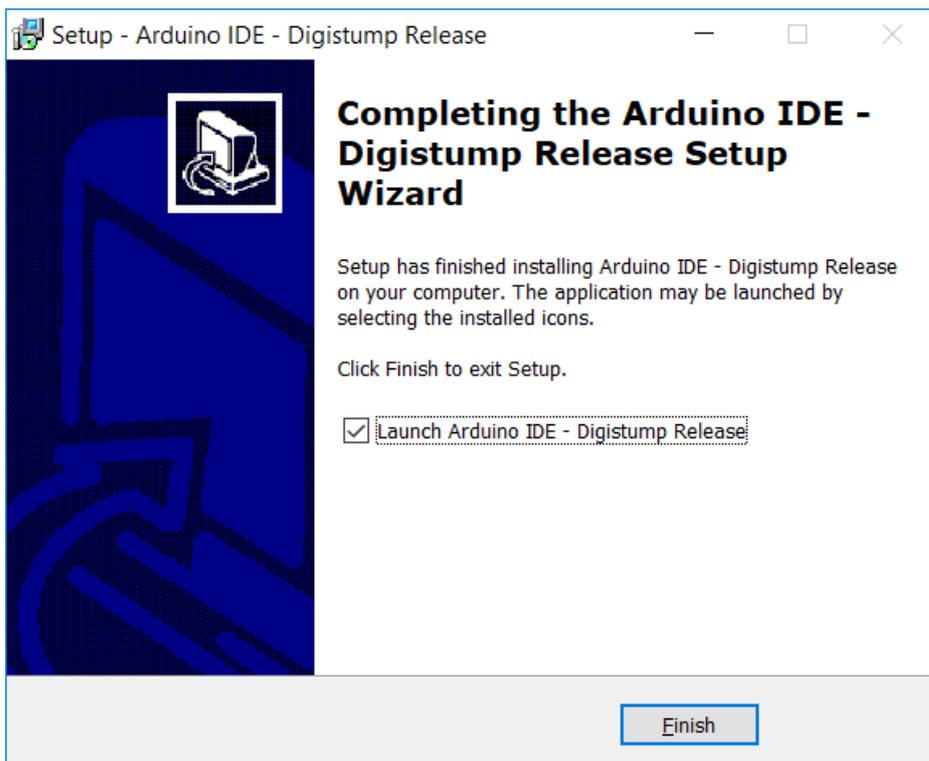
Anschließend meldet sich der Gerätetreiberinstallations-Assistent:  
Hier immer wieder mit **Weiter** bestätigen



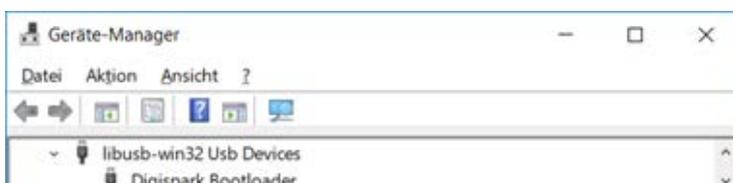
Falls man die Gerätetreiberinstallation bestätigen muss, dies bitte auch mit **Installieren** machen:



Zum Schluss die Treiberinstallation **Fertigstellen** und mit **Finish** ist die Installation abgeschlossen.

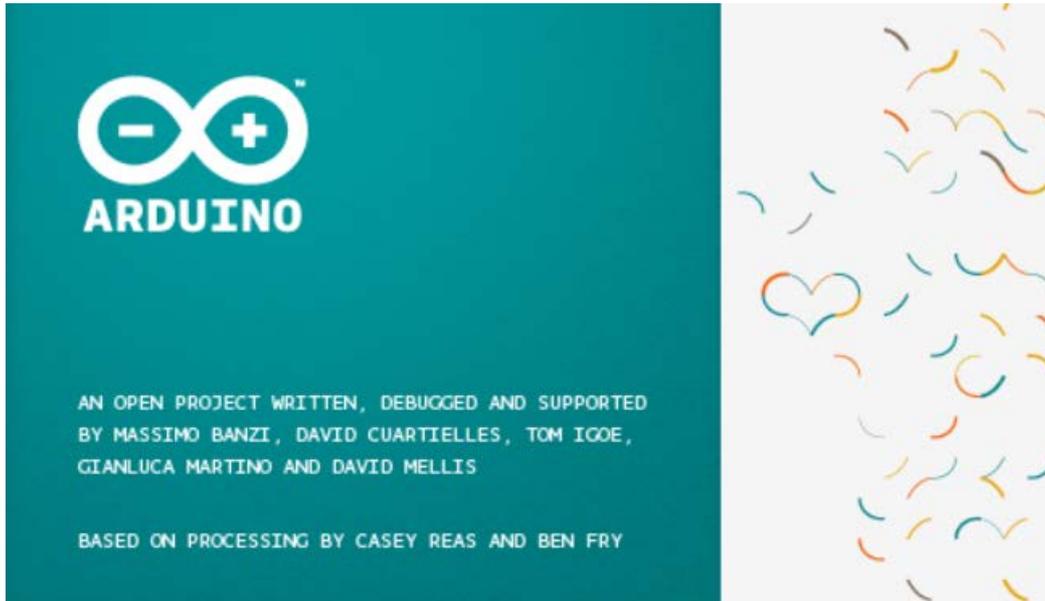


Schließt man jetzt den Digispark am Computer an und kontrolliert die Treiberinstallation im Gerätemanager, erscheint er als Digispark Bootloader.



# AZ-Delivery

Nach der Installation startet normalerweise die Arduino-Software automatisch.

The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_jan24a | Arduino 1.5.8". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". The toolbar contains icons for check, run, upload, and download. The main editor area shows the following code:

```
sketch_jan24a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

The status bar at the bottom indicates "1" on the left and "Arduino Uno on COM1" on the right.

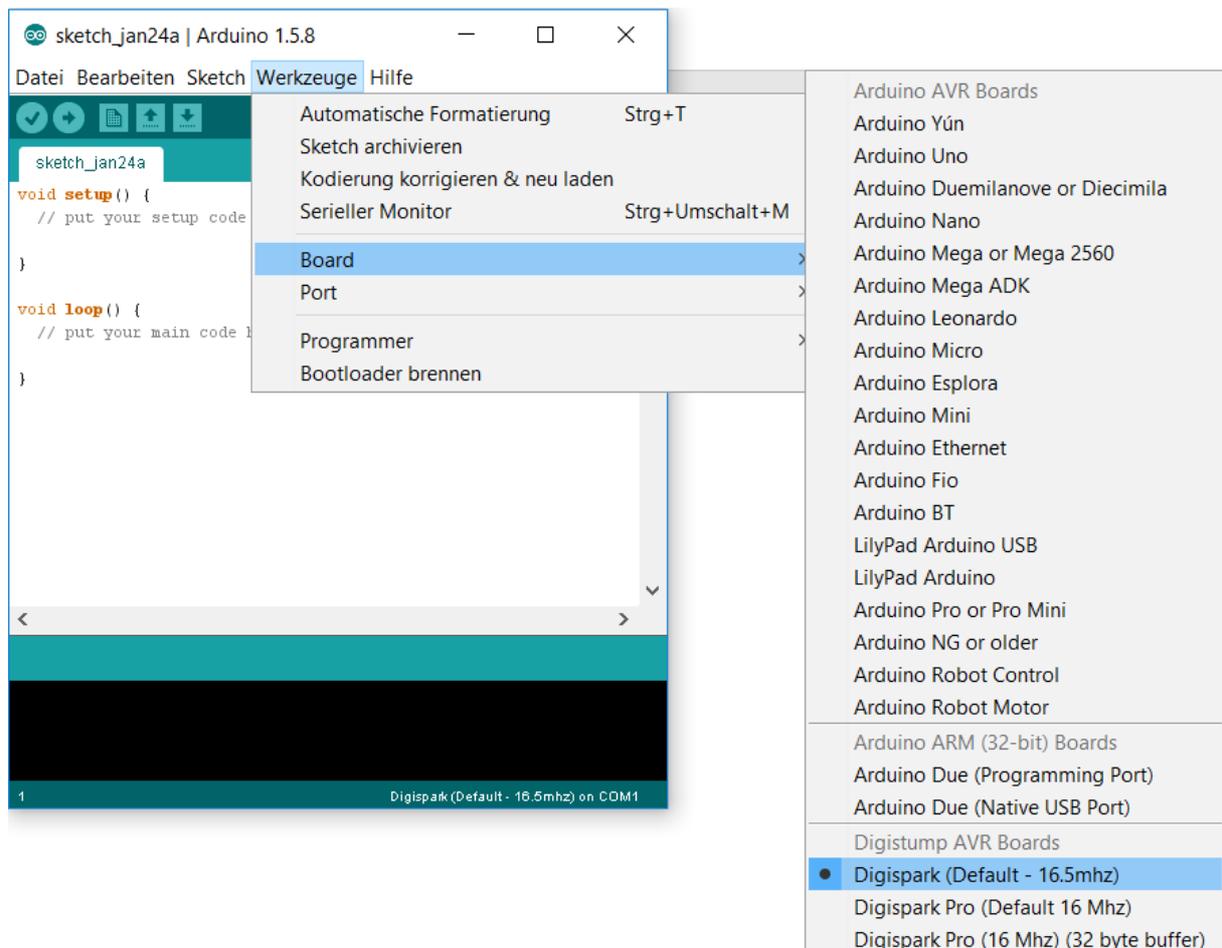
Die Installation ist erfolgreich abgeschlossen und wir können mit dem programmieren beginnen.

## Erste Schritte in der Arduino Programmiersoftware

Bevor wir den Digispark programmieren können, müssen wir in der Software auch den Digispark als Zielgerät definieren.

Dazu wählen wir in der Software:

Werkzeuge > Board > Digispark (Default – 16.5mhz)



Nachdem wir unser Zielgerät ausgewählt haben, schreiben wir unsere ersten Codezeilen, zu Beginn lassen wir die LED blinken:

```
void setup() {  
  
    pinMode(1, OUTPUT);    // Pin 1 (onBoard LED) als Ausgang definieren  
}  
  
void loop() {             //Schleife  
    digitalWrite(1, HIGH); // Ausgang auf HIGH schalten  
    delay(50);            // eine 50ms warten  
    digitalWrite(1, LOW); // Ausgang auf LOW schalten  
    delay(50);           // eine 100ms warten  
}
```

Nachdem wir den Code geschrieben haben klicken wir oben auf  und Verifizieren unser Programm:

Wenn alles stimmt und unser Programm keine Fehler enthält

```
Der Sketch verwendet 650 Bytes (10%) des Programmspeicherplatzes. Das Maximum sind 6.012 Bytes.  
Globale Variablen verwenden 9 Bytes des dynamischen Speichers.
```

können wir es auf den Digispark hochladen. Dazu klicken wir oben auf 

Wir bekommen eine Meldung, dass wir den Digispark einstecken sollen...  
Please plug in the device

```
Please contact the application's support team for more information.  
> Please plug in the device ...  
> Press CTRL+C to terminate the program.
```

Da dieser bereits eingesteckt ist, müssen wir ihn kurz vom Computer trennen und wieder einstecken, erst dann ist er wieder für kurze Zeit im Programmiermodus.

Jetzt starten wir noch einmal das Hochladen des Programms mit  und wenn die Software erfolgreich das Programm übertragen hat, kommt folgende Ausgabe:

```
Der Sketch verwendet 650 Bytes (10%) des Programmspeicherplatzes. Das Maximum sind 6.012 Bytes.  
Globale Variablen verwenden 9 Bytes des dynamischen Speichers.  
Running Digispark Uploader...  
Plug in device now... (will timeout in 60 seconds)  
> Please plug in the device ...  
> Press CTRL+C to terminate the program.  
> Device is found!  
connecting: 16% complete  
connecting: 22% complete  
connecting: 28% complete  
connecting: 33% complete  
> Device has firmware version 1.6  
> Available space for user applications: 6012 bytes  
> Suggested sleep time between sending pages: 8ms  
> Whole page count: 94 page size: 64  
> Erase function sleep duration: 752ms  
parsing: 50% complete  
> Erasing the memory ...  
erasing: 55% complete  
erasing: 60% complete  
erasing: 65% complete  
>> Eep! Connection to device lost during erase! Not to worry  
>> This happens on some computers - reconnecting...  
>> Reconnected! Continuing upload sequence...  
> Starting to upload ...  
writing: 70% complete  
writing: 75% complete  
writing: 80% complete  
> Starting the user app ...  
running: 100% complete  
>> Micronucleus done. Thank you!
```

# AZ-Delivery

Nun können wir mit **pinMode(0, OUTPUT)**; den Anschluss P0 programmieren. Uns stehen von 0 – 5 (pinMode(6, OUTPUT);) insgesamt 6 Pins zur Verfügung. Diese kannst du individuell als Digitalen Eingang, i2c, SPI oder auch als Analogen Eingang programmieren. Viel Spaß beim Experimentieren.

## **Du hast es geschafft, du kannst nun deinen Digispark programmieren!**

Ab jetzt heißt es lernen und eigene Projekte verwirklichen.

Und für mehr Hardware sorgt natürlich dein Online-Shop auf:

<https://az-delivery.de>

Viel Spaß!  
Impressum

<https://az-delivery.de/pages/about-us>