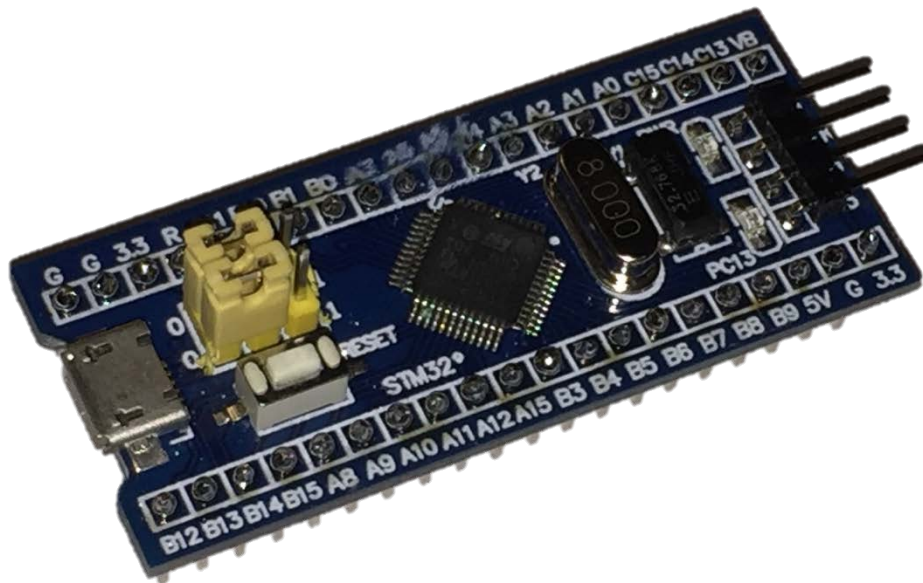


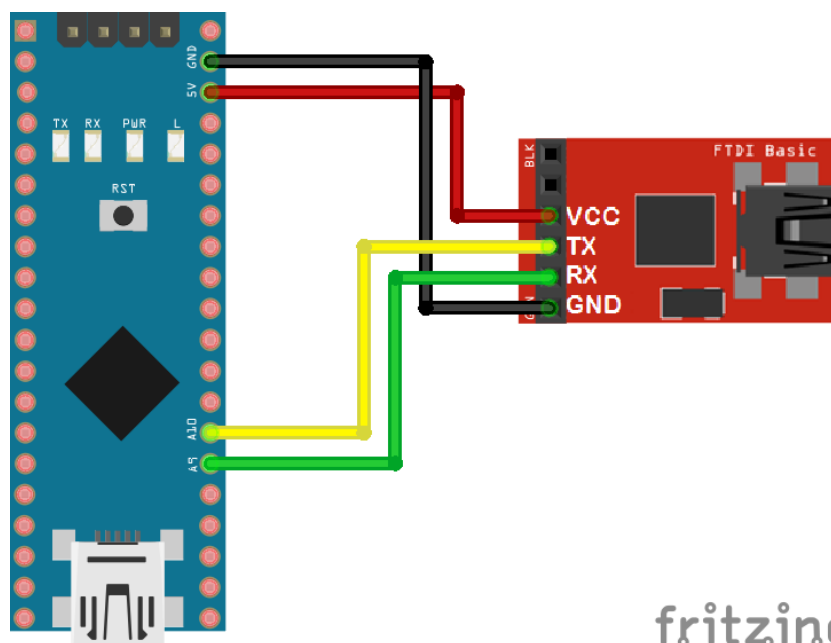
Willkommen!

Und herzlichen Dank für den Kauf unseres AZ-Delivery STM32F103C8T6 Controller. Auf den folgenden Seiten gehen wir die ersten Schritte auf dem Arduino durch. Viel Spaß!



Das STM32 Board lässt sich auch mit Arduino Programmieren, ist aber um ein Vielfaches leistungsfähiger als ein Arduino. Er besitzt einen 72 MHz Prozessor, 64kB Flash und 20kB RAM Speicher. Ebenso unterstützt dieser Controller auch USART, I²C, SPI, USB, CAN und DMA.

Verdrahten des Moduls mit einem Arduino Nano:



fritzing

+5V wird mit **5V** am STM32 verbunden
GND wird mit **GND** verbunden
RX wird mit **A9** verbunden
TX wird mit **A10** verbunden

Rote Leitung
Schwarze Leitung
Grüne Leitung
Gelbe Leitung

Nachdem alles verdrahtet ist kann der STM32 mit Spannung versorgt werden.

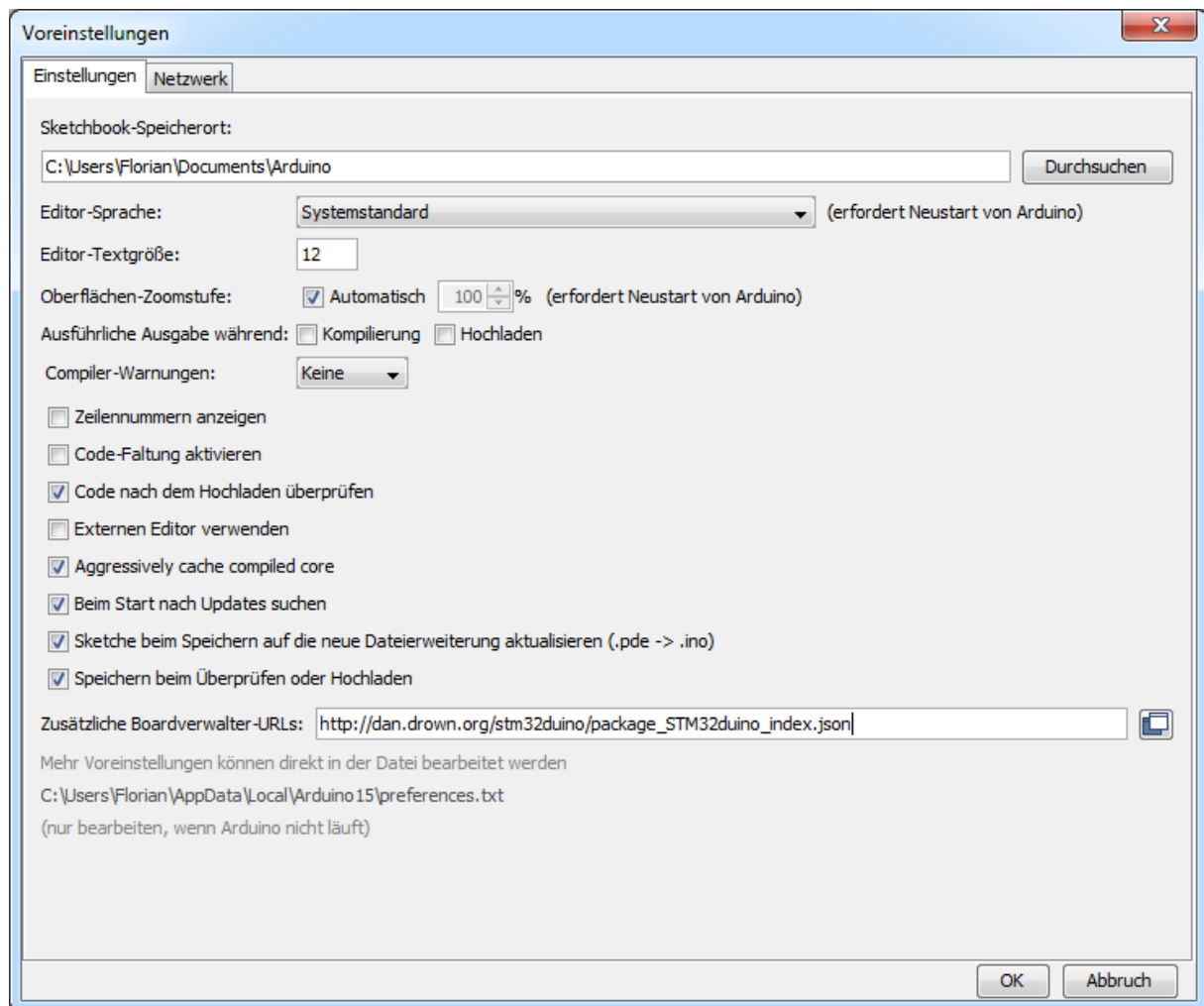
Programmieren des STM32:


Die Arduino-Software setzen wir in diesem Fall als schon installiert voraus.

Bevor wir aber mit dem Programmieren beginnen können, müssen wir zuerst noch ein paar notwendige Bibliotheken einbinden.

Die Arduino Software benötigt zunächst einmal alle Informationen zum STM32F103C8T6, dies können wir tun, indem wir unter dem „Voreinstellungen“ > „Zusätzliche Boardverwalter-URLs“ folgende Adresse eingeben:

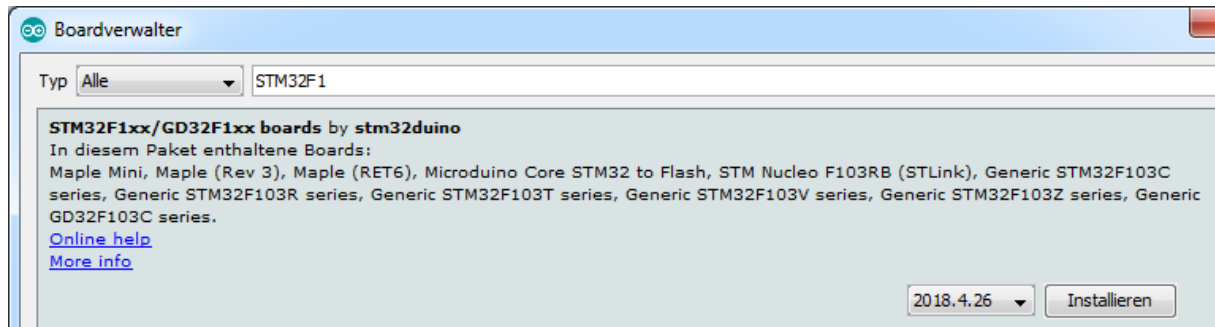
http://dan.drown.org/stm32duino/package_STM32duino_index.json



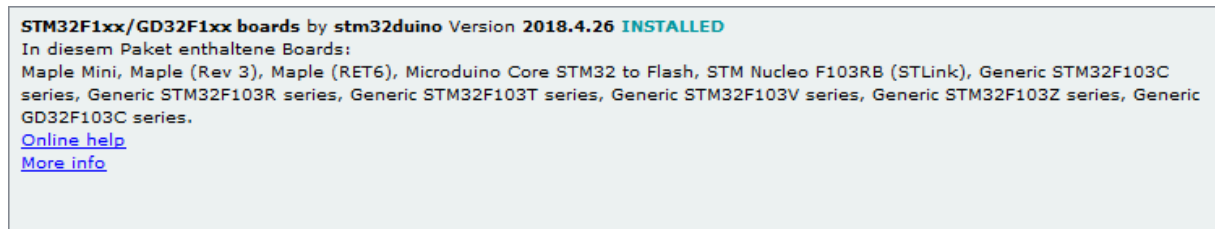
Evtl. wenn du schon einen Link eingetragen hast, auf den Button  klicken und in dem Fenster eine neue Zeile hinzufügen.

Bestätigen wir die Eingabe mit „OK“.

Ist das erledigt, gehen wir auf „Werkzeuge“ > „Board“ > „Boardverwalter“ und installieren die STM32F1 Bibliothek. In dem Boardverwalter geben wir in der Suchleiste oben rechts „STM32F1“ ein, es wird das Paket von stm32duino angezeigt. Dieses wählen wir aus und klicken auf Installieren.



Nach erfolgter Installation steht neben dem Paket nun INSTALLED.

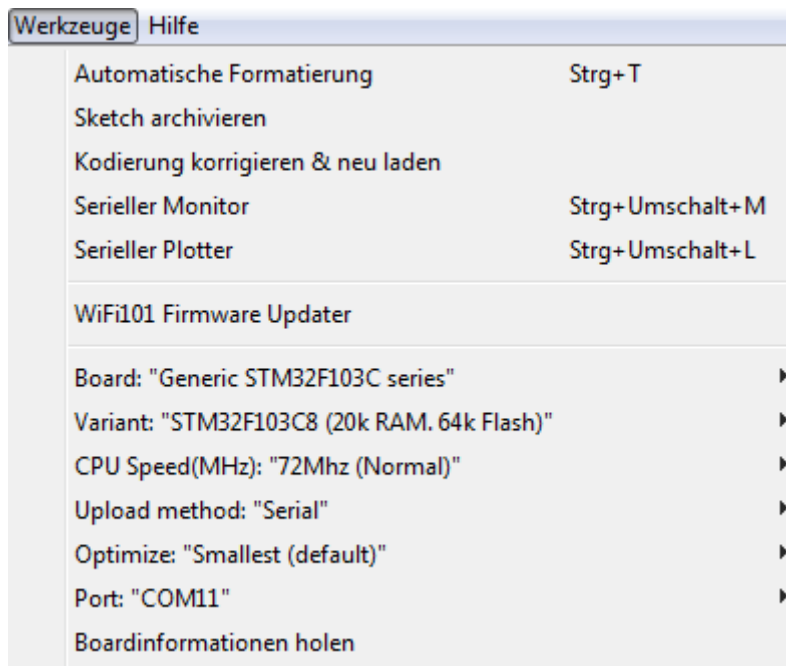


Als nächsten Schritt müssen wir das richtige Board auswählen:

Unter Werkzeuge > Board: „Generic STM32F103C series“

Upload method: „Serial“

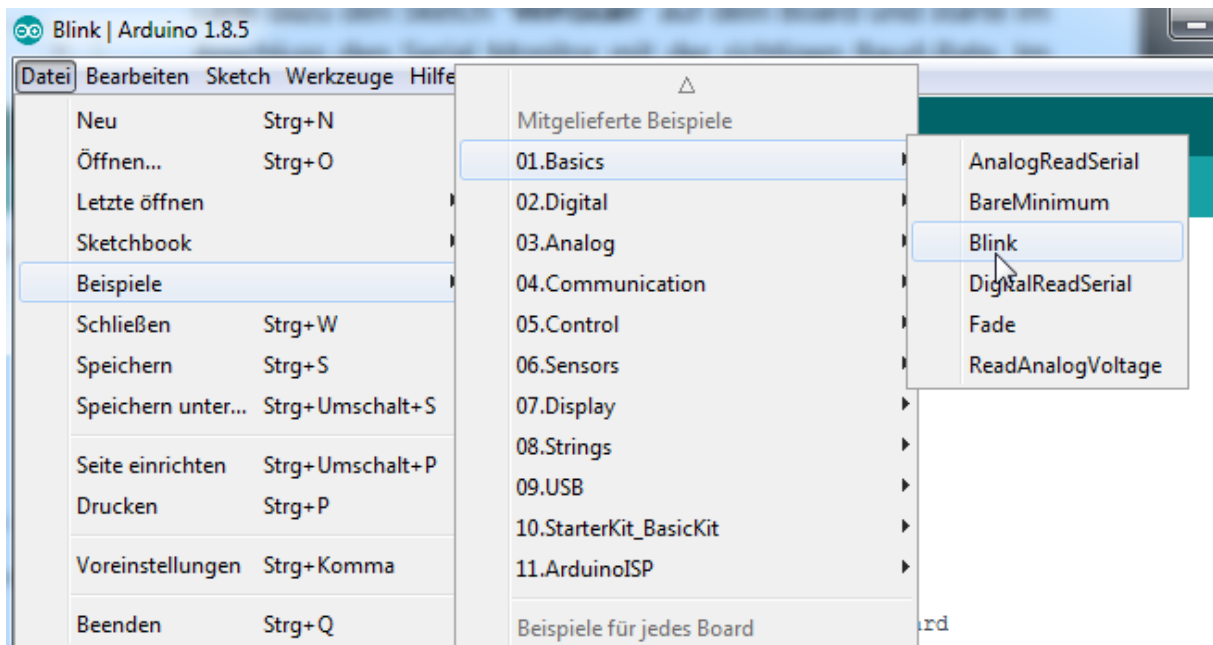
Port: „COMxx“ (hier dein Port des Serial Adapters)



Der Arduino Code:

Nachdem nun die Verdrahtung erledigt wurde, schreiben wir unseren ersten Code. Lassen wir die LED direkt auf dem STM32 blinken.

Wähle dazu unter Datei > Beispiele > 01.Basics > Blink aus.



Den nun angezeigten Code müssen wir noch etwas abändern:

```
// 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
}
```

Und zwar funktioniert die Arduino typische Version „LED_BUILTIN“ nicht.


Möchten wir aber trotzdem die eingebaute LED blinken lassen, so müssen wir LED_BUILTIN mit PC13 ersetzen. Der Pin PC13 ist die eingebaute LED.

Du kannst auch eine LED an jeden anderen GPIO anschließen

Der verbesserte Sketch sieht nun so aus:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(PC13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(PC13, HIGH);  // turn the LED on (HIGH is the voltage
level)
  delay(1000);               // wait for a second
  digitalWrite(PC13, LOW);   // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

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

Wenn alles stimmt und unser Programm keine Fehler enthält

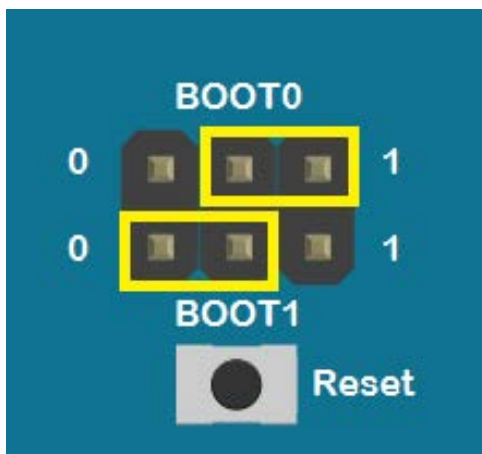
Kompilieren abgeschlossen.
Der Sketch verwendet 7420 Bytes (11%) des Programmspeicherplatzes. Das Maximum sind 65536 Bytes.
Globale Variablen verwenden 1984 Bytes (9%) des dynamischen Speichers, 18496 Bytes für lokale Var

können wir es auf den STM32 hochladen.

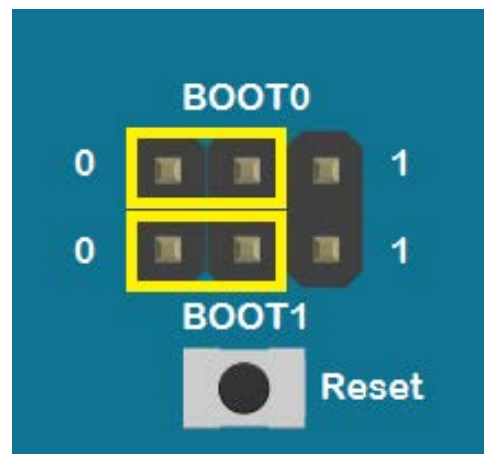
Bevor wir das Programm hochladen können, müssen wir den STM32 in den Programmiermodus booten.

Dazu muss der BOOT0 Jumper umgesteckt werden:

Programmiermodus:



Normaler Modus:



Danach drücken wir Reset und klicken oben auf 

Az-Delivery

Ist unser STM32 erfolgreich programmiert, erscheint folgende Meldung und die LED sollte blinken:

```
Hochladen abgeschlossen.

Using Parser : Raw BINARY
Interface serial_w32: 115200 8E1
Version      : 0x22
Option 1     : 0x00
Option 2     : 0x00
Device ID    : 0x0410 (Medium-density)
- RAM        : 20KiB (512b reserved by bootloader)
- Flash      : 128KiB (sector size: 4x1024)
- Option RAM : 16b
- System RAM : 2KiB
Write to memory
Erasing memory

Wrote address 0x08000100 (3.46%)
Wrote address 0x08000200 (6.92%)
Wrote address 0x08000300 (10.37%)
Wrote address 0x08000400 (13.83%)
Wrote address 0x08000500 (17.29%)
Wrote address 0x08000600 (20.75%)
Wrote address 0x08000700 (24.20%)
Wrote address 0x08000800 (27.66%)
Wrote address 0x08000900 (31.12%)
Wrote address 0x08000a00 (34.58%)
Wrote address 0x08000b00 (38.03%)
Wrote address 0x08000c00 (41.49%)
Wrote address 0x08000d00 (44.95%)
Wrote address 0x08000e00 (48.41%)
Wrote address 0x08000f00 (51.86%)
Wrote address 0x08001000 (55.32%)
Wrote address 0x08001100 (58.78%)
Wrote address 0x08001200 (62.24%)
Wrote address 0x08001300 (65.69%)
Wrote address 0x08001400 (69.15%)
Wrote address 0x08001500 (72.61%)
Wrote address 0x08001600 (76.07%)
Wrote address 0x08001700 (79.52%)
Wrote address 0x08001800 (82.98%)
Wrote address 0x08001900 (86.44%)
Wrote address 0x08001a00 (89.90%)
Wrote address 0x08001b00 (93.35%)
Wrote address 0x08001c00 (96.81%)
Wrote address 0x08001cec (100.00%) Done.

Starting execution at address 0x08000000... done.
```

Dann kann der Jumper wieder auf Normalen Modus umgesteckt werden und ein Micro-USB Kabel, zur Spannungsversorgung, am STM32 angeschlossen werden.



Du hast es geschafft, du kannst deinen STM32F103C8T6 programmieren!

Ab jetzt heißt es lernen und eigene Projekte verwirklichen.
Viel Spaß wünscht dir AZ-Delivery.

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

<https://az-delivery.de>

Viel Spaß!
Impressum

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