repair café

Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Ditzinger Makerspace

Arduino Upload

Ein Projekt von Repair-Café und Stadtbücherei Ditzingen

10.02.2018 Stadtbücherei Ditzingen

Bürgerstiftung Ditzingen

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Inhalt

- Arduino Build-Prozess
- Wie kommt das Programm in den Arduino?
- Übertragung vom PC zum Arduino
- `avrdude`
- Upload-Vorgang
- Fuses und Lock-Bits
- Warum Programmierprozess beeinflussen?
- Behebung von Problemen beim Programmieren
- Arduino Uno USB-Interface-Prozessor programmieren
- Arduino als ISP
- ISP-Stecker
- Arduino-Bootloader programmieren
- Fuses oder Lock-Bits ändern / Board-Definition ändern

Bürgerstiftung Ditzingen

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Arduino Build-Prozess

1) Arduino-Präprozessierung
   - Fügt zur setup- und loop-Funktion des Sketches den Code hinzu, der notwendig ist, damit das Programm compilierbar wird.
2) Compilieren – nicht nur Sketch, sondern auch hinzugefügte Bibliotheken.
3) Linken – macht alles zu *einem* Programm.
4) Laden auf den Arduino (auch „Hochladen“, „Upload“, „Programmieren“, „Flashen“, „Brennen“).

Bürgerstiftung Ditzingen

Wie kommt das Programm in den Arduino (I)?

Es gibt (für alle Mikroprozessoren) grundsätzlich zwei Möglichkeiten:

ISP-Methode

A) Ohne aktive Beteiligung des Prozessors

- Das Programm wird durch die Prozessor-HW von Prozessor-Pins entgegengenommen und ins Flash-ROM geschrieben.
- Bei ATMEL-AVR-Prozessoren z.B. durch
  - Parallel-Programmierung
    - schnell
    - geht i.d.R. nicht in der Schaltung ("In Circuit"), da sowohl der Datenbus als auch zusätzlich Steuerleitungen benötigt werden.
  - Serielle Programmierung
    - meist (wie auch bei den Arduinos) in der Schaltung, dann **ISP (In Circuit Seriell Programming)**

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Wie kommt das Programm in den Arduino (II)?

Bootloader-Methode

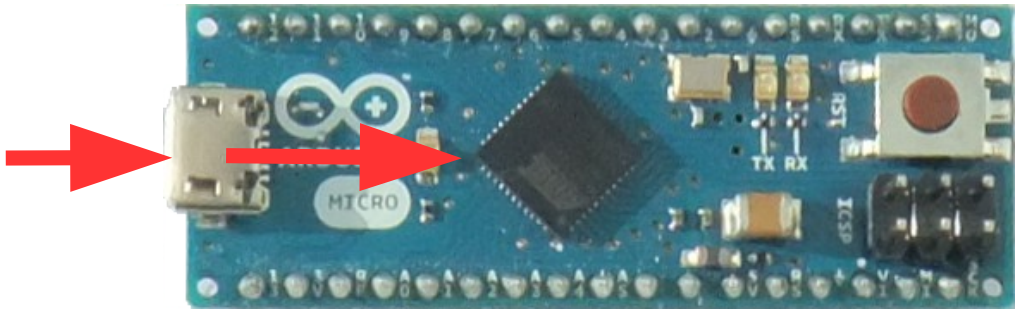
B) Der Prozessor hilft mit

- Dazu muss er in seinem Flash-ROM einen **Bootloader** enthalten, der das Programm z.B. von der seriellen Schnittstelle entgegen nimmt und ins Flash-ROM schreibt.
- Der Bootloader selber kann nur über Methode A) über ein ISP-Interface in den Prozessor gelangen.
- Der Bootloader wird immer beim Einschalten oder Reset ausgeführt.
- Wenn er merkt, dass geflasht werden soll, startet er den Programmiervorgang, sonst startet er das Applikations-programm.

Übertragung vom PC zum Arduino (I)

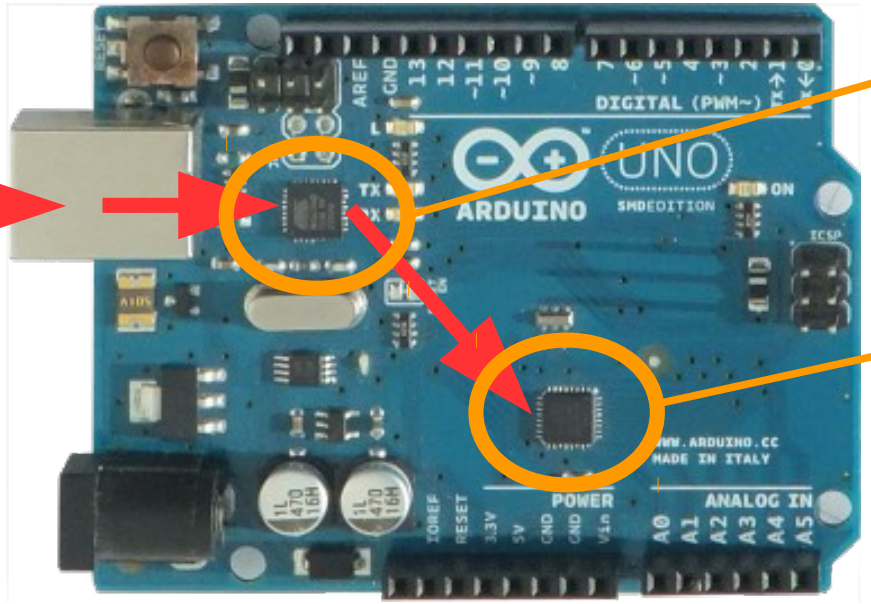
Das Programm muss vom Host-PC zum Mikroprozessor übertragen werden.

- Bei **Methode B)** („mittels Bootloader“):
  - ➔ **über USB**.
  - Einige Mikroprozessoren (z.B. `ATmega32u4` im Arduino Micro) haben einen USB-Treiber im Bootloader integriert.



Übertragung vom PC zum Arduino (II)

- Andere haben keinen USB-Stack, z.B. der `ATmega328P` im Arduino Uno.
- Dort **zweiter Prozessor** (`ATmega16u2`) mit USB-Interface – wird durch ein eigenes ISP-Interface programmiert.

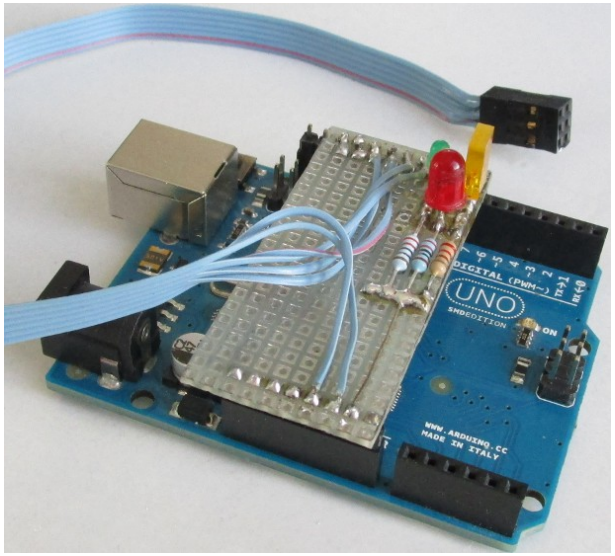


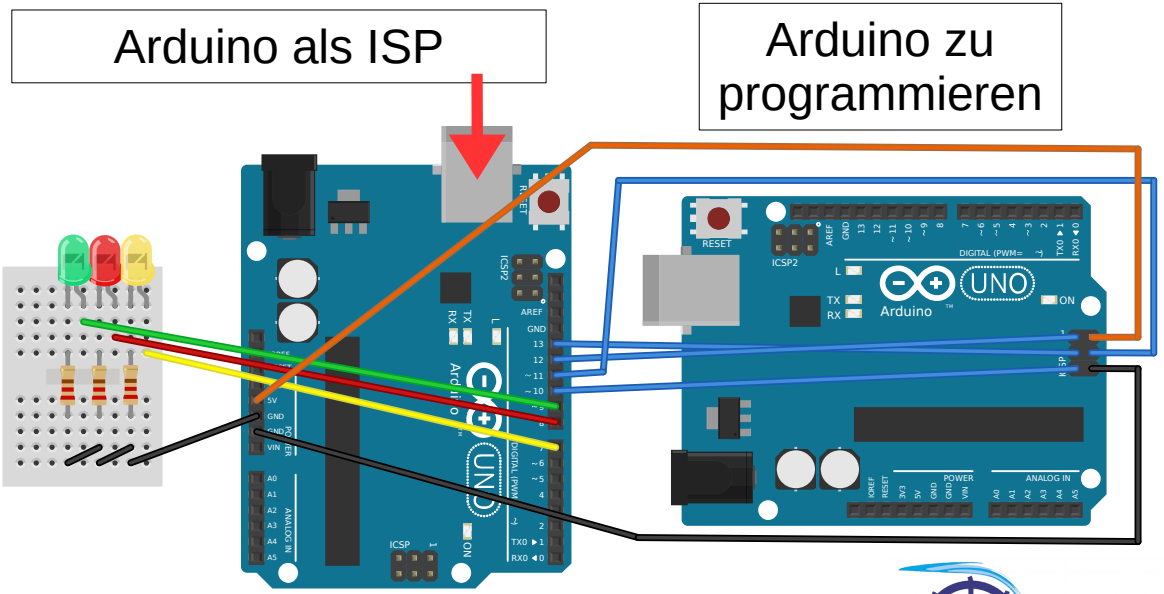
USB-Interface-Prozessor `ATmega16U2`

Hauptprozessor `ATmega328P`

Übertragung vom PC zum Arduino (III)

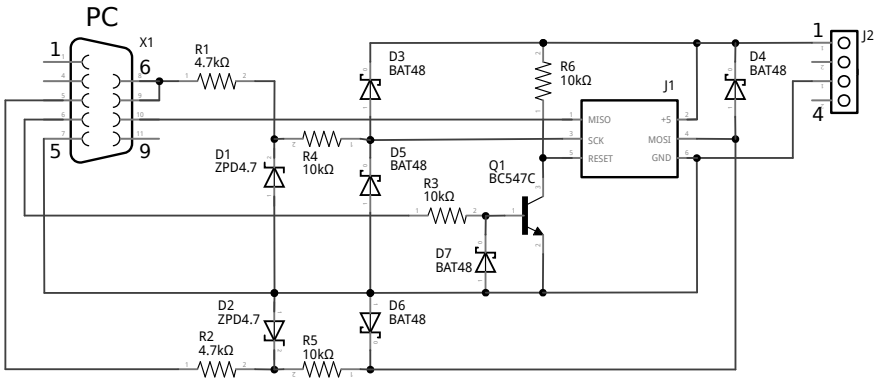
- Bei **Methode A)** („über ISP“)
  - ➔ **Programmieradapter**, der i.d.R. auch über USB an den PC angeschlossen ist.
  - Viele kommerzielle Fabrikate
  - Man kann auch einen Arduino Uno verwenden (später mehr):

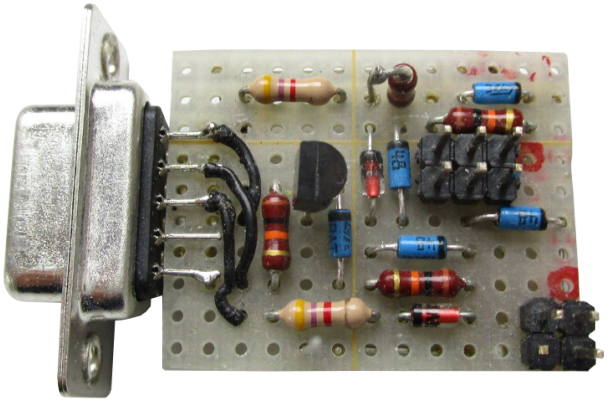




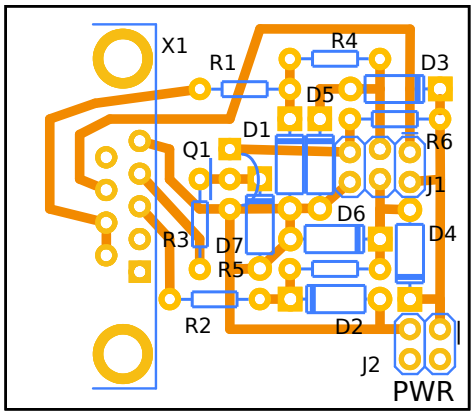
Übertragung vom PC zum Arduino (IV)

- … oder Eigenbau-Programmer, z.B. „Einfachst-Programmer“ DASA3 (mit separatem USB-RS232-Adapter):





Prototyp



Layout (doppelseitig)

Siehe AVR-DASA3-1.0.tgz

avrdude

Host-PC

- Auf dem PC muss das Programm zum Arduino oder zum Programmieradapter übertragen werden.
- Das passiert mit **`avrdude`**:
  - Open-Source-Universalprogramm zum Programmieren von ATMEL-AVR-Prozessoren
  - wird mit Arduino IDE automatisch mit installiert.
- Man kann `avrdude` auch ohne Arduino-IDE zum Programmieren von Arduinos oder allgemein ATMEL-AVR-Prozessoren verwenden.

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Upload-Vorgang

- Vor dem Upload müssen einige Parameter eingestellt werden:
  - Menü **Werkzeuge ► Board**
    - Wenn man die genaue Bezeichnung des Boards nicht kennt:
    - **Werkzeuge ► Boardinformationen holen**.
  - **Werkzeuge ► Port**
    - muss auf das Arduino-USB-Device eingestellt werden
    - Unter Linux liefert Kommandozeilenbefehl

      ```
      ls -lrt /dev/tty*
      ```

      in der letzten Zeile das zuletzt angeschlossene Arduino-USB-Device.
  - Beim Programmieren über Tool-Bar-Upload-Button ist Option **Werkzeuge ► Programmer** egal (zu der kommen wir später).

- Für Detailinformationen während des Upload-Prozesses:
  - **Datei ► Voreinstellungen ► Ausführliche Ausgabe während: ☑Hochladen**
  - Damit gibt `avrdude` auch die Lock-Bits und Fuses aus (später mehr).

Bürgerstiftung Ditzingen

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Fuses und Lock-Bits

- Die **Grundkonfiguration** eines Mikroprozessors wird durch „**Fuses**“ bestimmt.
  - Einstellungs-Bits,
  - z.B. Clock-Grundkonfiguration, Startup-Verhalten, etc.
- **Zugriffsberechtigungen** auf den Mikroprozessor-Inhalt werden durch „**Lock-Bits**“ bestimmt,
  - z.B. Lese-/Schreibzugriffsrechte auf das Flash-ROM.
- Fuses und Lock-Bits können nur über **ISP** programmiert werden
  - also nur mit einem **Programmieradapter**.
- Die Fuses und Lock-Bits unterscheiden sich für unterschiedliche Prozessoren.
- Beschreibung im jeweiligen **Prozessorhandbuch**, siehe Anhang.

repair café
Entsorgen? Nein, danke!

Warum Programmierprozess beeinflussen?

STADTBIBLIOTHEK DITZINGEN

- **Beheben von Problemen** beim Programmieren
- **Boot-Verhalten verändern**
  - Beispiel:
    - Frühere Bootloader des Arduino Micro brauchten sehr lange, bis das Applikationsprogramm gestartet wurde.

      Bei neueren geht das erheblich schneller.
- **Bootloader ganz entfernen**
  - Startet den Arduino schneller.
  - Spart 1KB ein.
- Um **Fuses** oder **Lock-Bits** zu **verändern**

Bürgerstiftung Ditzingen

Behebung von Problemen beim Programmieren (I)

Wenn das Programmieren eines Arduinos nicht funktioniert (und außerhalb des Arduinos alles i.O. ist) ...

- kann der **Bootloader** defekt sein. Das kommt gelegentlich vor, evt. durch SW-Fehler im Applikationsprogramm oder durch Entladung von Zellen im Flash-ROM oder (wahrscheinlicher) durch Störungen beim Programmieren.
  → Bootloader neu flashen.
- können **Fuses** oder **Lock-Bits** sich verändert haben.
  - Wenn noch Zugriff auf den Prozessor-Inhalt möglich ist
    → Locks bzw. Fuses neu programmieren. Das geht nur über ISP.
- kann die **Hardware** des Hauptprozessors defekt sein.
  - Wenn es sich um eine Version mit DIL-Prozessor handelt
    → Prozessor gegen einen fabrikneuen tauschen.
  - Bei diesem müssen dann sowohl Bootloader, als auch Fuses programmiert werden.
  - Bei Programmieren eines Bootloaders in der Arduino IDE werden Fuses und Lock-Bits immer automatisch mit programmiert.

repair café
Entsorgen? Nein, danke!

Behebung von Problemen beim Programmieren (II)

STADTBIBLIOTHEK DITZINGEN

Wenn das Programmieren eines Arduinos nicht funktioniert (und außerhalb des Arduinos alles i.O. ist) ...

- kann der **zweite** (USB-Interface-)**Prozessor** defekt sein (im Arduino Uno `ATmega16U2`).
  - Das ist jedoch sehr selten der Fall.
  - Prüfung mit Loop-Back-Test (http://forum.arduino.cc/index.php?topic=73748.0):
    - Reset mit Gnd verbinden
    - RX mit TX verbinden
    - Auf dem Host-PC ein Terminalprogramm ausführen mit dem Arduino-USB-Device
    - z.B. Linux auf der Kommandozeile

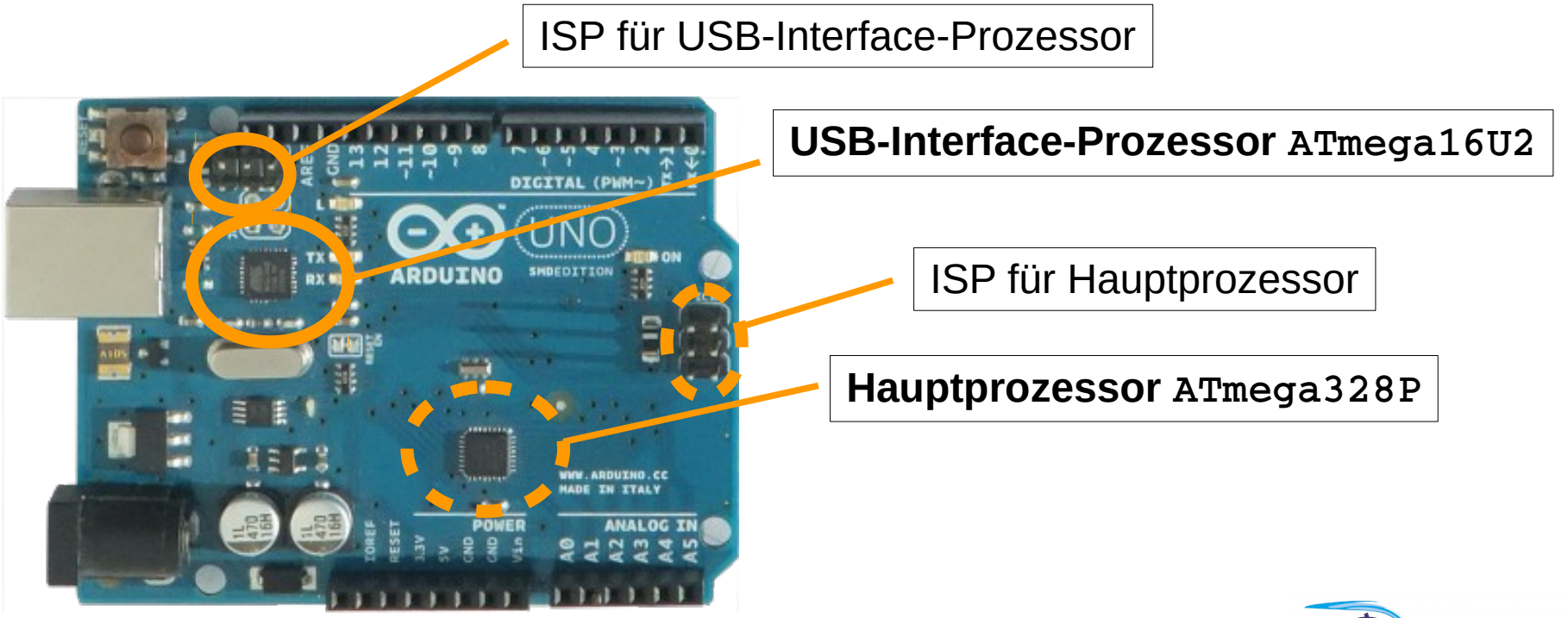
      ```
      screen /dev/ttyACM0
      ```

    - Dann muss der USB-Interface-Prozessor das Eingetippte unverändert zurück senden.
  - Wenn nur das Programm defekt ist
    → USB-Interface-Prozessor neu flashen.

Bürgerstiftung Ditzingen

Arduino Uno USB-Interface-Prozessor programmieren

Zweiter (USB-Interface)-Prozesser (`Atmega16U2`) bei Arduino Uno kann nur über seinen **ISP-Stecker** programmiert werden.



repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Arduino als ISP (I)

- Zum Programmieren von Bootloadern, Fuses und Lock-Bits wird ein separater ISP-Programmer benötigt.
- Das geht auch mit einem separaten Arduino.
  - Bei Arduino-IDE-Installation wird ein ISP-Sketch mit installiert in `examples/11.ArduinoISP`
  - **Achtung**: beim Programmieren in der mit Arduino IDE über separate Programmer (auch Arduino als ISP) **wird der Bootloader überschrieben** – er ist dann weg.
    - Damit der normale Upload wieder funktioniert, muss der Bootloader erst wieder re-programmiert werden.
  - Beim separaten Programmer werden **nicht** automatisch die **Fuses** und **Lock-Bits** mit programmiert.

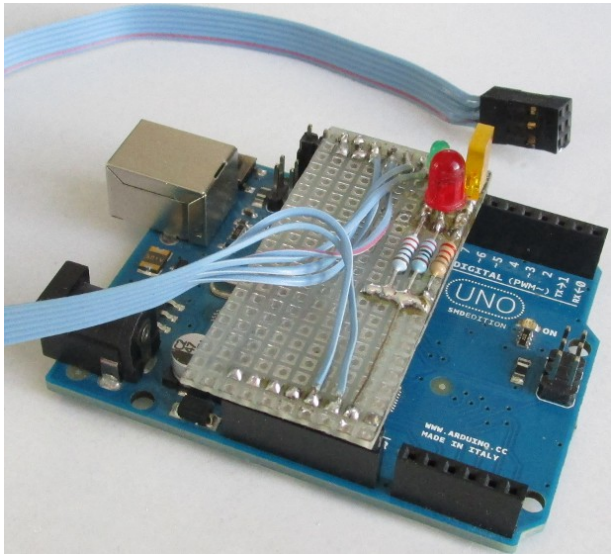
Bürgerstiftung Ditzingen

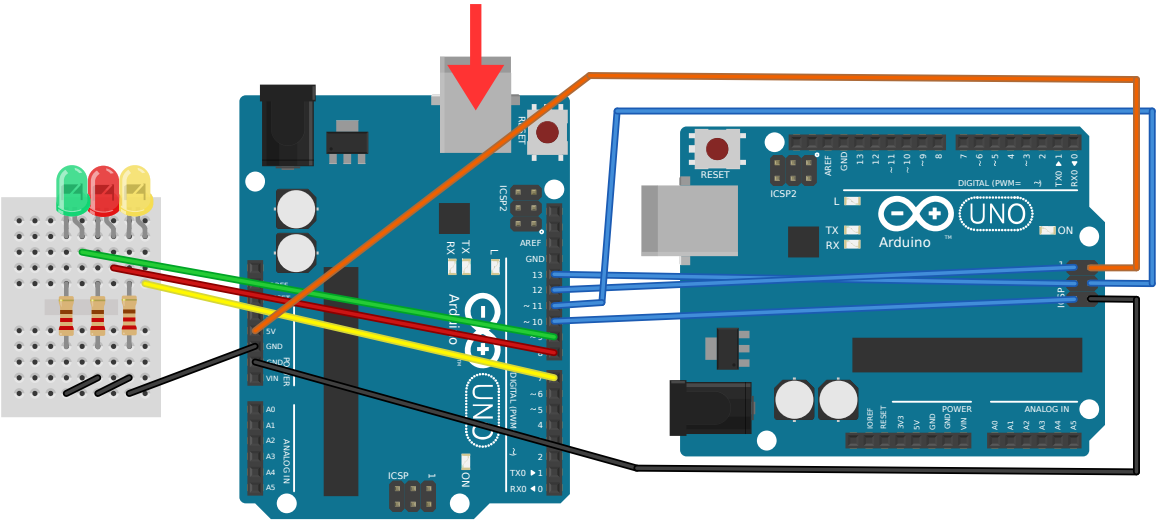
repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Arduino als ISP (II)

Arduino als ISP mit Prototyp-Shield	Arduino als ISP mit Steckbrett	Arduino zu programmieren





Bürgerstiftung Ditzingen

repair Entsorgen? Nein, danke! café

Arduino als ISP (III) Ablauf

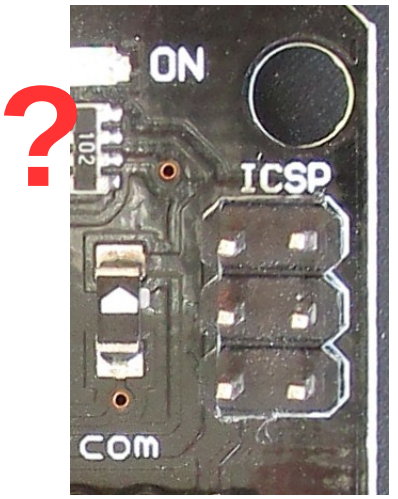
STADTBIBLIOTHEK DITZINGEN

1) Arduino-als-ISP-Hardware erstellen (das sind 3 LEDs mit Vorwiderständen) – auf Steckbrett oder Shield, siehe Bilder (es geht prinzipiell auch ohne).
2) ISP-Arduino mit ISP-Sketch über normalen Upload programmieren.
   - **Werkzeuge ► Board** ist das Board des ISP-Arduinos
   - **Werkzeuge ► Port** ist das USB-Device des ISP-Arduinos
   - Programmieren mit **Sketch ► Hochladen** (oder Upload-Button) 
3) ISP-Arduino mit zu programmierendem Arduino verbinden.
   - Arduino Uno als ISP: siehe Bild, bei anderen siehe Beschreibung im Sketch.
   - Der zu programmierende Arduino wird **über den ISP-Arduino mit Strom versorgt**.
4) Zu programmierenden Arduino programmieren.
   - **Werkzeuge ► Board** ist das Board des zu programmierenden Arduino
   - **Werkzeuge ► Port** ist das USB-Device des ISP-Arduinos
   - **Werkzeuge ► Programmer** ist „Arduino as ISP“ (nicht ArduinoISP, ArduinoISP.org)
   - Programmieren mit **Sketch ► Hochladen mit Programmer** (nicht Upload-Button) 

Bürgerstiftung Ditzingen

ISP-Stecker

- Achtung: ISP-Stecker sind **nicht gegen Verpolen geschützt**!
- Bei verpolt aufgesteckten Adapter werden sowohl **Arduino**, als auch **Programmer beschädigt**.
- **Pin-1-Markierung** beachten!
- Beispiele:



repair café
Entsorgen? Nein, danke!

Arduino-Bootloader programmieren

STADTBIBLIOTHEK DITZINGEN

- Der Bootloader in einem Arduino kann nur über einen **separaten ISP-Programmer** programmiert werden,
  - z.B. auch Arduino als ISP.
- Bootloader mit Arduino-IDE programmieren:
  - Parameter wie in Abschnitt „Arduino als ISP“ einstellen.
    - Je nachdem den richtigen Programmer einstellen.
  - **Werkzeuge ► Bootloader brennen**
  - Beim Upload des Bootloaders programmiert die Arduino-IDE immer **auch die Fuses und Lock-Bits** mit.

- Die Arduino-Bootloader liegen im Arduino-IDE-Installationsverzeichnis in
  - `hardware/arduino/avr/bootloaders`

Bürgerstiftung Ditzingen

Fuses / Lock-Bits ändern
Board-Definition ändern (I)

Beispiel: Arduino Uno soll System-Clock auf Pin ausgeben

- Dazu muss in **`lfuse`** das zweithöchste Bit resettet werden.
- Im Board-Definitionsfile
  `hardware/arduino/avr/boards.txt`
  die Definition für das Board kopieren und die Namen vor dem ersten Punkt ändern.
- In diesem Beispiel den Bereich
  `uno.…`
  kopieren und umbenennen nach
  `unoclkout.…`
- Einen neuen Namen vergeben, z.B.:
  `unoclkout.name=Arduino Uno Clockout`
  statt vorher `Arduino/Genuino Uno`.
- Außerdem den Wert für `low_fuses` ändern in:
  `unoclkout.bootloader.low_fuses=0xBF`
  statt vorher `0xFF`.

Fuses / Lock-Bits ändern
Board-Definition ändern (II)

- Nach dem Neustart der Arduino IDE ist jetzt bei
  Werkzeuge ► Board
  zusätzlich der Auswahlpunkt
  Arduino Uno Clockout
  vorhanden.
- Damit das neue Fuse-Bit wirksam wird, muss zunächst mit einem separaten Programmer der Bootloader programmiert werden, wie im Abschnitt „Arduino-Bootloader programmieren“ beschrieben.
- Danach kann mit dem normalen Upload der Sketch programmiert werden.
- Der Arduino gibt dann an Pin 8 den System-Clock aus.

repair café
Entsorgen? Nein, danke!

Neuen Programmer definieren

STADTBIBLIOTHEK DITZINGEN

- Falls ein neuer Programmer verwendet werden soll, muss dieser in die Programmer-Definitionsdatei aufgenommen werden.
- Ähnliches Vorgehen, wie im letzten Beispiel.

Beispiel: DASA3-Programmer einfügen

- In Programmerdatei
  `hardware/arduino/avr/programmers.txt`
  Abschnitt eines möglichst ähnlichen Programmers suchen, kopieren und den Namen vor dem ersten Punkt ändern.
- In diesem Beispiel den Bereich
  `avrisp.…`
  kopieren und umbenennen nach
  `dasa3.…`
- Außerdem die restlichen Werte anpassen (nach den Informationen aus der `avrdude`-Dokumentation):

```
dasa3.name=DASA3
dasa3.protocol=dasa3
dasa3.program.protocol=dasa3
dasa3.program.extra_params=-P{serial.port} -i500
```

- Nach dem Neustart der Arduino IDE ist jetzt bei Werkzeuge ► Programmer zusätzlich der Auswahlpunkt DASA3 vorhanden.

Bürgerstiftung Ditzingen

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Anhang
Wo ist was?

- **`avrdude:`**
  Inst-Verz`/hardware/tools/avr/bin`
- **`avrdude`-Konfiguration:**
  Inst-Verz`/hardware/tools/avr/etc/avrdude.conf`
- **Board-Beschreibungen (mit Fuse-Bits):**
  Inst-Verz`/hardware/arduino/avr/boards.txt`
- **Liste der Programmer:**
  Inst-Verz`/hardware/arduino/avr/programmers.txt`
- **Bootloader:**
  Inst-Verz`/hardware/arduino/avr/bootloaders`
  - **Arduino Uno:** `…/optiboot/optiboot_atmega328.hex`
  - **Arduino Micro:** `…/caterina/Caterina-Micro.hex`
- **Arduino Uno ATmega16u2-Firmware:**
  Inst-Verz`/hardware/arduino/avr/firmwares`

repair café
Entsorgen? Nein, danke!

STADTBIBLIOTHEK DITZINGEN

Anhang
Links

- **Arduino.cc Webseite:**
  https://www.arduino.cc/
- **IDE Download:**
  https://www.arduino.cc/en/Main/Software#
- **Arduino Reference:**
  https://www.arduino.cc/en/Reference/HomePage
- **ATMEL Prozessorhandbücher**
  http://www.microchip.com
  - z.B. ATmega328P
    http://www.microchip.com/wwwproducts/en/atmega328p
  - z.B. ATmega32u4
    http://www.microchip.com/wwwproducts/en/atmega32u4
- **ATMEL Application Notes**
  - Atmel-0943-In-System-Programming_ApplicationNote_AVR910
    http://ww1.microchip.com/downloads/en/AppNotes/Atmel-0943-In-System-Programming_ApplicationNote_AVR910.pdf
  - Atmel USB DFU Bootloader Datasheet
    http://ww1.microchip.com/downloads/en/devicedoc/doc7618.pdf
- **Downloads**
  http://www.buergerstiftung-ditzingen.de/index.php/downloads/makerspace
  - AVR-DASA3-1.0.tgz
  - FusesLocksATmega328P-1.0.tgz
  - FusesLocksATmega32U4-1.0.tgz

Anhang
Lizenz

- Dieses Werk ist lizenziert unter einer
  Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International
  (https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de)
  Lizenz.