

Open Hardware Workshop - Chemnitzer
Linux-Tage 2014

2014-03-11

Contents

1	Allgemeine Hinweise	3
1.1	Funktion der Schaltung	3
1.2	Hinweise zum Löten	3
2	Bestückung der Platine	4
2.1	Bauteilliste	4
2.2	Bestückungsplan	5
2.3	Reihenfolge der Bestückung	5
2.3.1	ICs	5
2.3.2	Sockel und Quarzoszillator	6
2.3.3	Buchsenleisten	7
2.3.4	Widerstände und Kondensatoren	8
2.3.5	Externe Anschlüsse	10
2.3.6	Sensor	12
2.4	Inbetriebnahme	14
3	Schaltungsaufbau	15
3.1	Mikrocontroller	15
3.2	Serielle Schnittstelle	15
3.3	Temperatur-Feuchtigkeitssensor	15
3.4	Ansteuerung der Matrix	15
3.5	Stromversorgung	16

4	Ansteuerung	17
4.1	Verbindung	17
4.1.1	BlueZ 4.x (Debian/Ubuntu/Raspbian)	17
4.1.2	BlueZ 5.x (Arch Linux, Fedora 20, ...)	19
4.2	ArduRPC	21
4.2.1	Installation	21
4.2.2	Verwendung	22
5	Software	23
5.1	Programmierung	23
5.2	Bootloader	23
5.3	Arduino IDE	23

Chapter 1

Allgemeine Hinweise

1.1 Funktion der Schaltung

Die Schaltung wurde mehrfach aufgebaut und getestet. Trotzdem können wir nicht sicherstellen, dass die Schaltung unter allen Bedingungen einwandfrei funktioniert. Der Toleranzbereich der Treiber und der LEDs kann unter bestimmten Umständen leicht überschritten werden. Für Schäden, z.B. an Hardware, können wir keine Haftung übernehmen!

1.2 Hinweise zum Löten

- LötKolben werden bis zu 400°C heiß, Verbrennungs- und Brandgefahr!
- Flußmitteldämpfe nicht einatmen!
- LötZinn kann Blei enthalten, während der Arbeit nicht Essen oder Trinken und nach der Arbeit Hände waschen!
- Augen schützen, Spritzgefahr durch flüssiges Zinn (Vorsicht beim Auslöten)!

Chapter 2

Bestückung der Platine

2.1 Bauteilliste

Name	Beschreibung
C1,C2	KERKO 100nF
IC1	IC-Sockel 28-polig mit ATmega328P
IC2, IC4, IC6	ULN2803A (8 Darlington Arrays)
IC3	UDN2983A (Treiber)
IC5, IC7, IC8	74HC595N (8-Bit Schieberegister)
JP1	Anschluss: Stromversorgung
JP2	Anschluss: Sensor
JP3	Anschluss: Bluetooth-Modul
JP4	Stiftleiste 1X2 (Programmier-Modus)
JP5, JP6, JP7	Anschluss: nicht verwendet
QG1	Quarzoszillator 20MHz
R1-R16	Widerstand 90,9 Ohm
R17-R24	Widerstand 120 Ohm
R25-R29,R31-R34	Widerstand 10k Ohm
R30	Widerstand 4.99k Ohm
SV1-4	Buchsenleiste für 8x8 RGB-LED-Matrix
V1	74HC237N (3-to-8 Line Decoder)
-	Temperatur-Feuchtigkeitssensor: DHT11

Name	Beschreibung
-	8x8 RGB-LED-Matrix
-	Bluetooth-Modul: BT_BOARD mit HC-06
-	Stecker-Schaltnetzteil: 5V/1A
-	Schrumpfschlauch
-	Jumper

2.2 Bestückungsplan

Für eine grobe Orientierung befindet sich auf der Platine ein Bestückungsdruck. Weitere Informationen können dem Bestückungsplan entnommen werden.

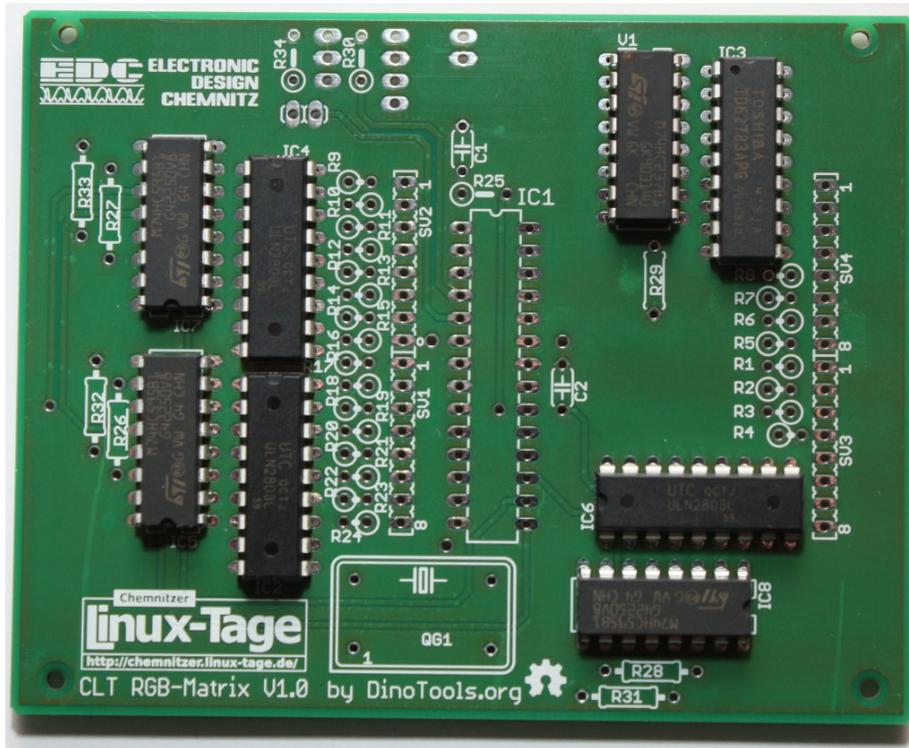
2.3 Reihenfolge der Bestückung

Achtung: Beim Auflöten der Bauteile ist insbesondere bei den ICs und dem Quarzoszillator auf die richtige Richtung zu achten.

Die im Folgenden beschriebenen Schritte haben sich bei unseren Testaufbauten als günstig erwiesen.

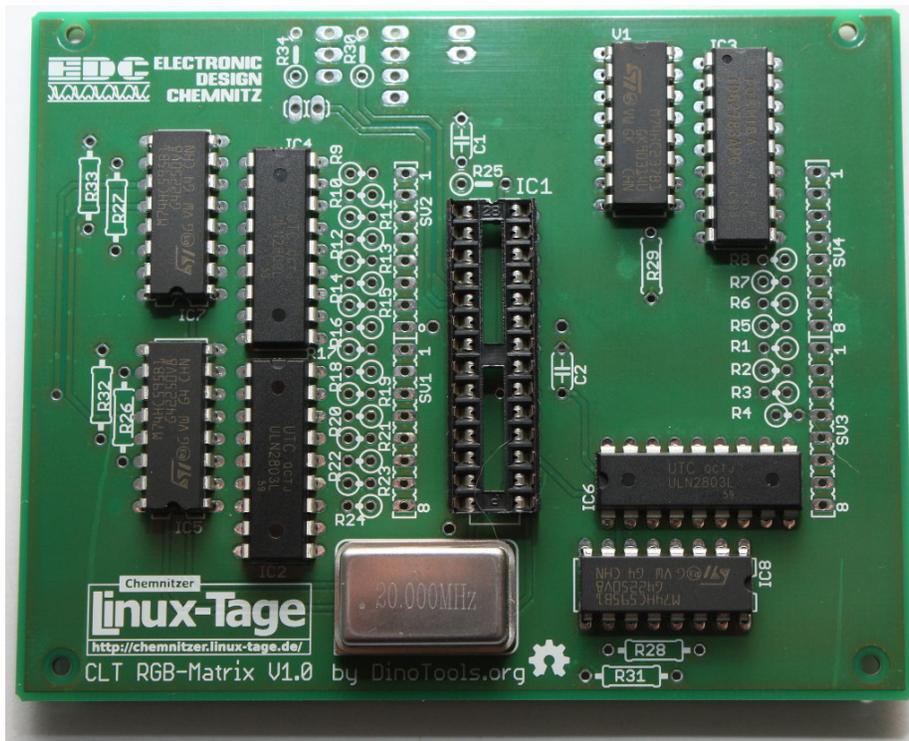
2.3.1 ICs

Im ersten Schritt werden die ICs aufgelötet. Jedoch noch nicht der Sockel für den Mikrocontroller.



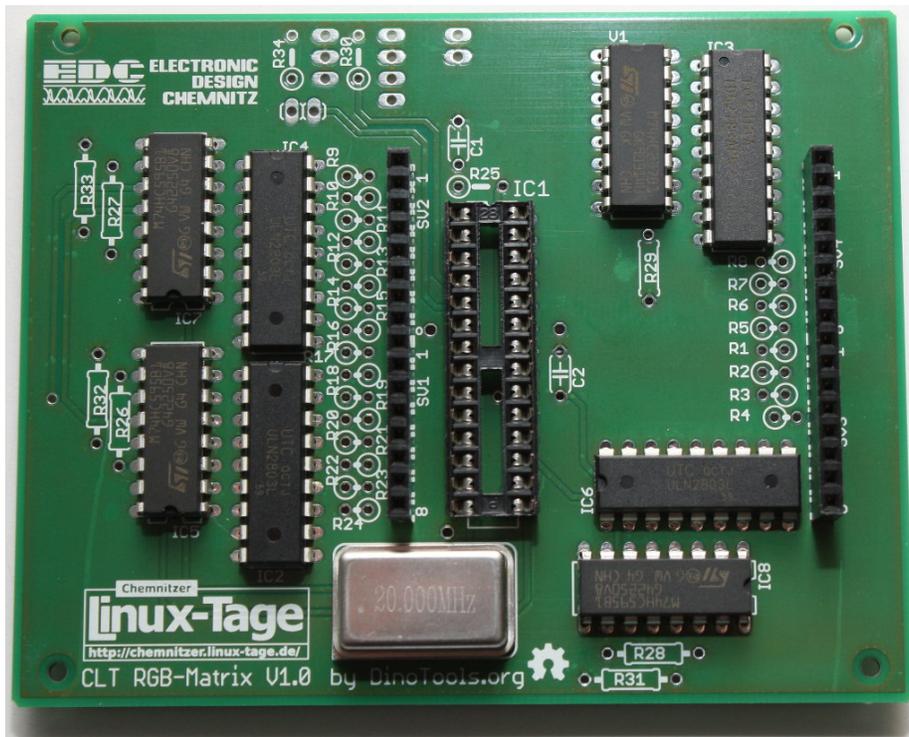
2.3.2 Sockel und Quarzoszillator

Anschließend sollten der Sockel und der Quarzoszillator aufgelötet werden.



2.3.3 Buchsenleisten

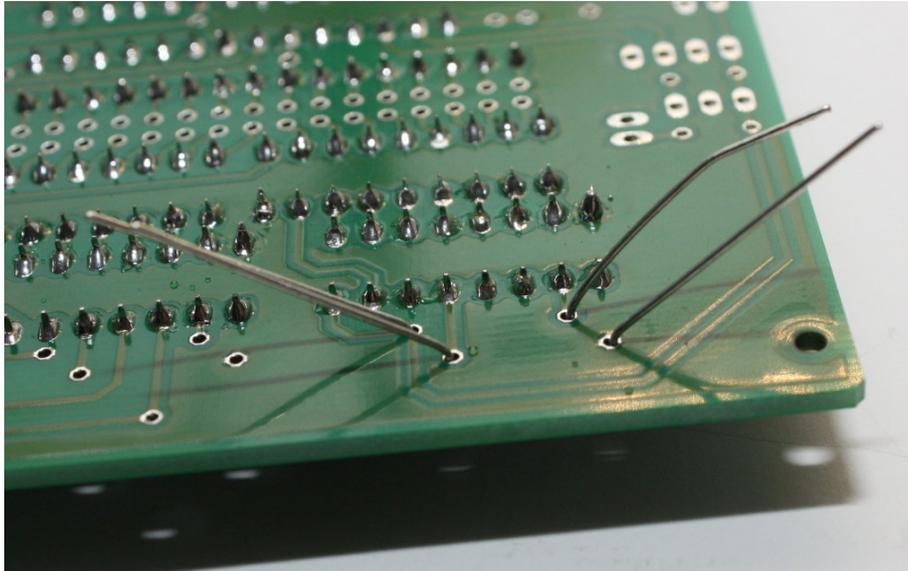
Vor dem Anlöten der Widerstände und der Kondensatoren empfiehlt es sich, die Buchsenleisten zu befestigen.



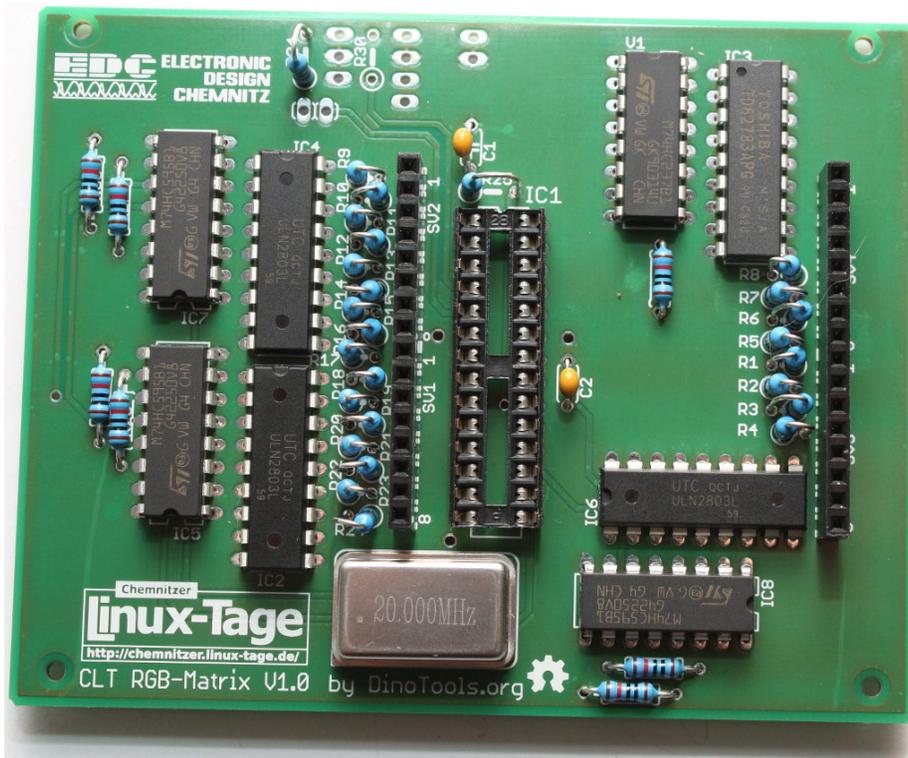
2.3.4 Widerstände und Kondensatoren

Bevor die Befestigung der Widerstände und der Kondensatoren möglich ist, müssen die Beinchen vorsichtig in die passende Form gebogen werden. Nach dem die Bauteile durch die Leiterplatte gesteckt wurden, können die Beinchen zur temporären Befestigung und vor dem Lötens, wie auf dem folgenden Bild zu sehen, **leicht** umgebogen werden.

Achtung: Es bietet sich an, den Widerstand R30 erst nach dem nächsten Schritt zu befestigen.



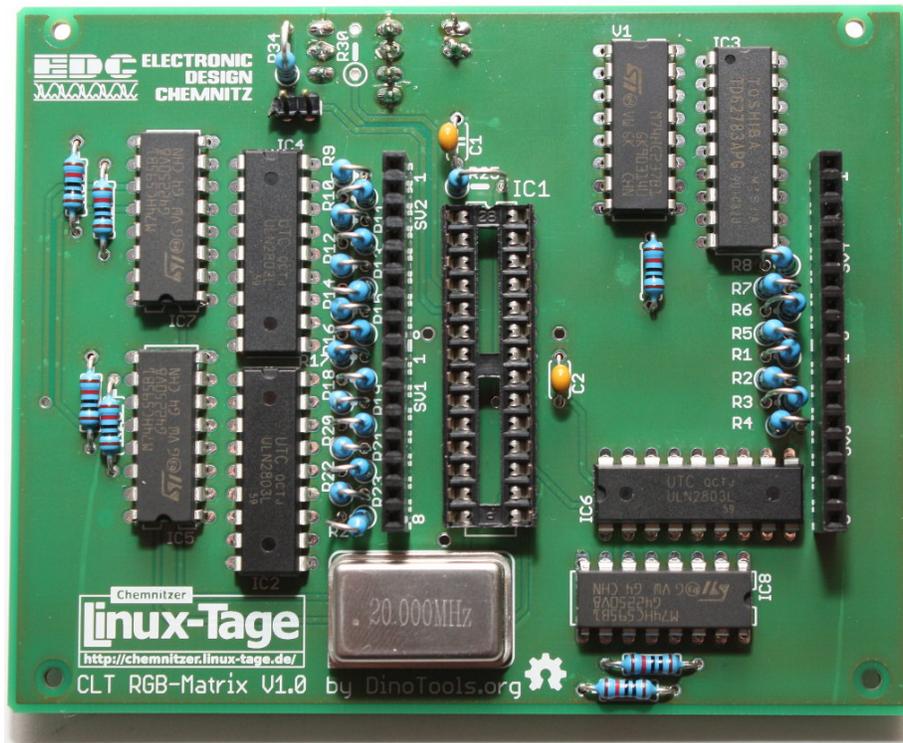
Anschließend sollte die Platine wie folgt aussehen.

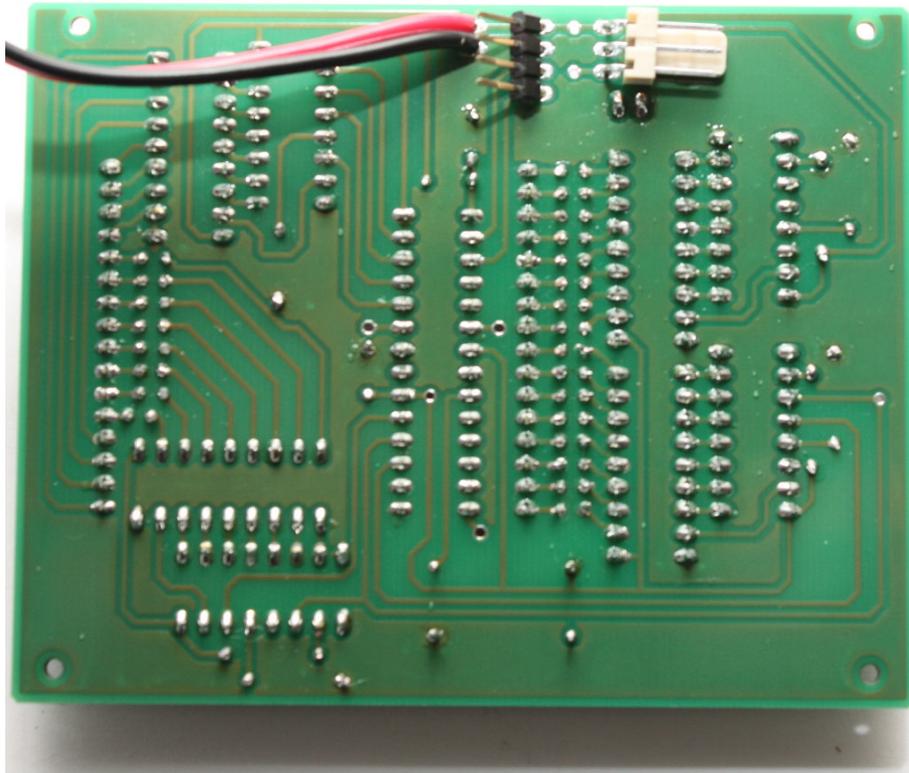


2.3.5 Externe Anschlüsse

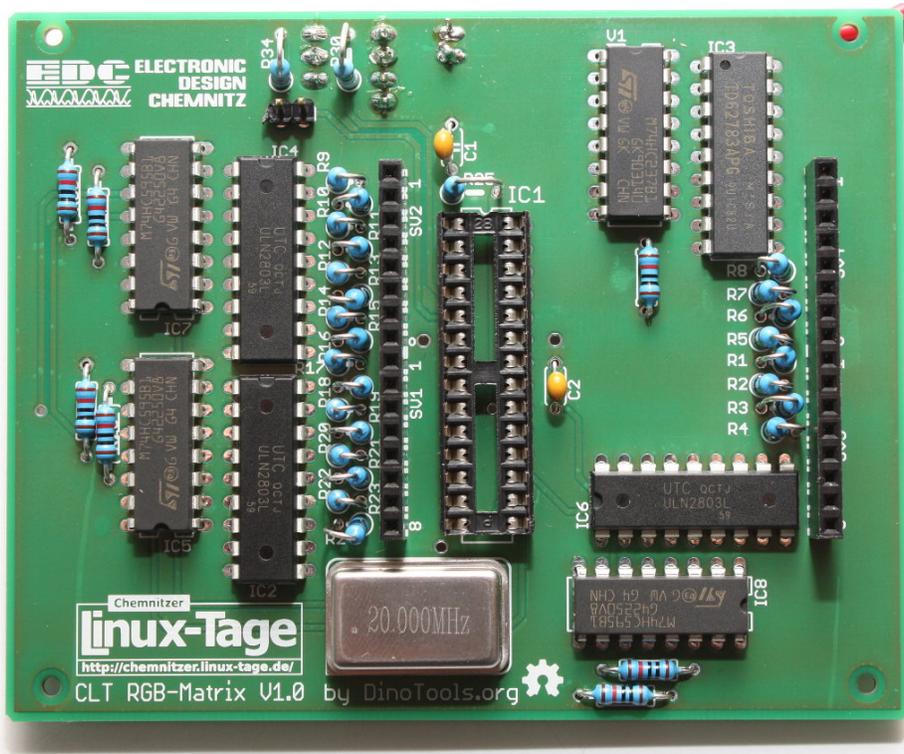
In diesem Schritt werden die Steckverbinder für die externen Komponenten wie das Bluetooth-Modul und den Sensor befestigt. Gleichzeitig werden der Buchsenstecker für die Stromversorgung und die Pins für den Programmier-Jumper aufgelötet.

Hinweis: Es gibt verschiedene Möglichkeiten welche Art Verbinder zum Einsatz kommt. Die Betreuer vor Ort stellen diese gern kurz vor.





Achtung: Wurde der Widerstand R30 noch nicht befestigt, sollte es jetzt nachgeholt werden.

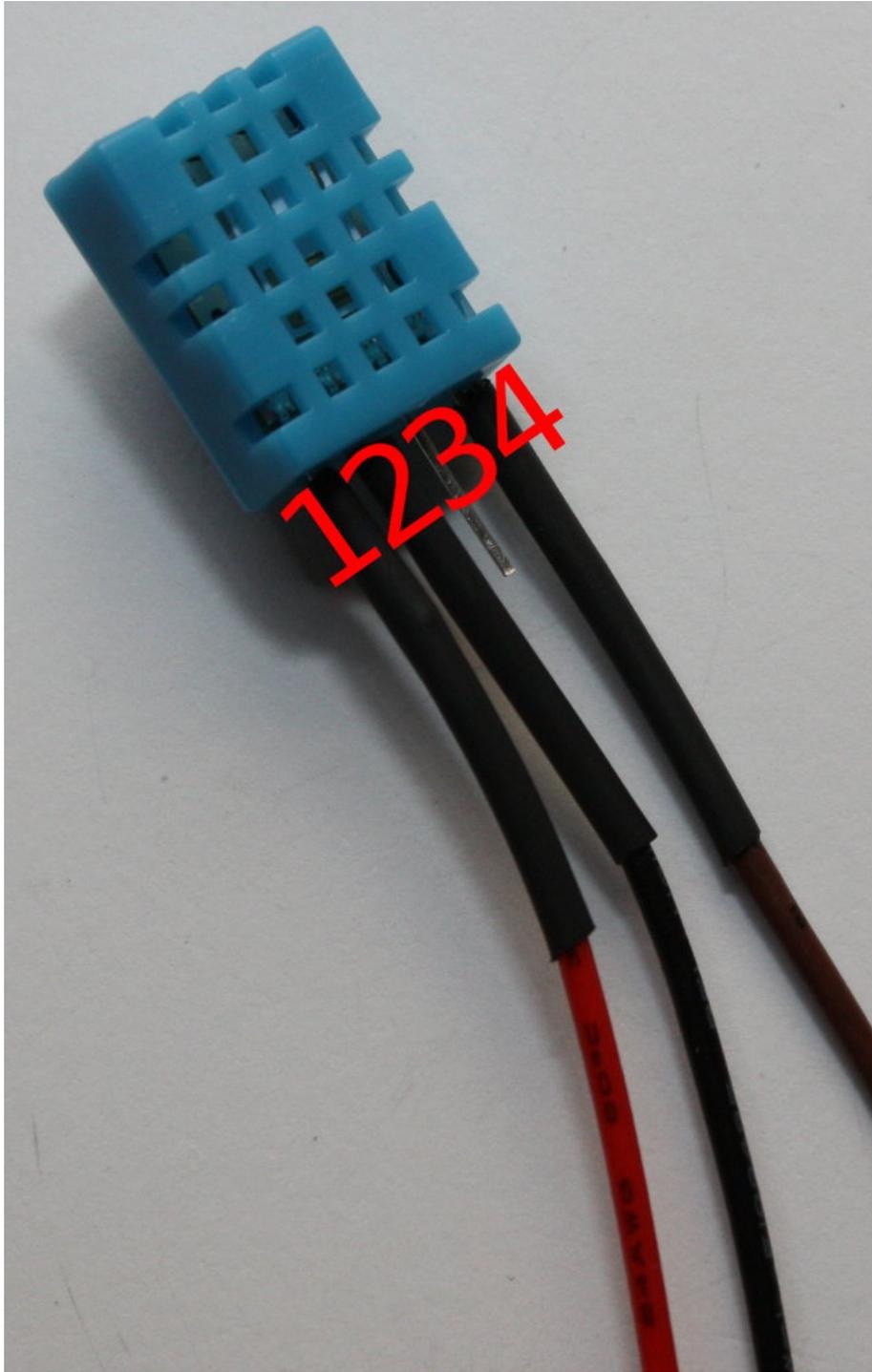


2.3.6 Sensor

Nachdem die Platine für die Matrix fertig ist, muss abschließend noch der Sensor an den entsprechenden Stecker gelötet werden.

Hinweis: Nicht vergessen vor dem Anlöten den Schrumpfschlauch zu kürzen und auf die Kabel zu stecken.

Pin	Farbe	Funktion
1	rot	VCC
2	schwarz	Daten
3	nicht verwendet	wendet
4	braun	GND



2.4 Inbetriebnahme

Der Bootloader und das Grundprogramm befinden sich für einen schnellen Start bereits auf dem Mikrocontroller. Daher sind nur noch wenige Schritte für die eigentliche Inbetriebnahme durchzuführen.

1. Funktionstest über ein bereits konfiguriertes Bluetooth-Modul
2. Konfiguration und Personalisierung des im Bausatz enthaltenen Bluetooth-Modules
 - Setzen der Baud-Rate auf 19200
 - Festlegen des Namen (Standardmäßig: HC-06)
 - Festlegen der PIN
3. Erneuter Funktionstest mit dem personalisierten Bluetooth-Modul

Chapter 3

Schaltungsaufbau

3.1 Mikrocontroller

Zum Einsatz kommt ein AVR Mikrocontroller vom Type ATmega328P. Dieser verfügt über 32kB Flash Programmspeicher, 2kB SRAM und 1kB EEPROM für die nichtflüchtige Speicherung von Daten. Als Taktgeber wird ein externer Oszillator mit 20MHz verwendet.

3.2 Serielle Schnittstelle

Der im Mikrocontroller integrierte USART-Baustein übernimmt die Funktion einer seriellen Schnittstelle. Darüber kann entweder direkt das im Bausatz enthaltene Bluetooth-Modul oder ein zusätzlicher USB auf Serial Adapter mit Unterstützung des TTL 5V Pegels angeschlossen werden.

3.3 Temperatur-Feuchtigkeitssensor

Bei dem Temperatur-Feuchtigkeitssensor handelt es sich um einen DHT11. Dieser ist über drei Pins, von denen zwei für die Stromversorgung und einer für die Datenübertragung verwendet werden, mit dem Board verbunden. Zur Datenübertragung kommt ein proprietäres 1-wire Protokoll zum Einsatz.

3.4 Ansteuerung der Matrix

Die Ansteuerung der Matrix erfolgt zeilenweise über ein Multiplexing-Verfahren. Mit Hilfe von drei 8-Bit Schieberegistern wird die Ansteuerung der acht RGB-

LEDs pro Zeile realisiert. Die Auswahl der anzusteuernden Zeile erfolgt über einen 3-zu-8 Multiplexer. Mit diesem Verfahren ist es möglich, die Anzahl der benötigten Ausgänge des Mikrocontrollers möglichst gering zu halten. Über Pulsweitenmodulation (PWM) kann die Helligkeit der einzelnen LEDs in mehreren Schritten beeinflusst werden.

Sind in einer Zeile alle LEDs aktiv, treten Ströme von mehr als 500mA auf. Um diesen Strom zuverlässig schalten zu können, kommen weitere Treiber-ICs zum Einsatz.

3.5 Stromversorgung

Die Schaltung ist für eine Betriebsspannung von 5V ausgelegt. Sind in einer Zeile alle LEDs an, kann die Stromaufnahme 500mA übersteigen. Das im Bausatz enthaltene Netzteil kann bei 5V bis zu 1A liefern und kann über die Holkupplung(5,5/2,1mm) direkt mit dem Board verbunden werden.

Chapter 4

Ansteuerung

4.1 Verbindung

Zur Ansteuerung der CLTMatrix muss eine serielle Verbindung zu dem Bluetooth-Modul bestehen. Dies kann mithilfe des Bluetooth-Protokolls RFCOMM erfolgen. Mit RFCOMM können eine oder mehrere serielle Schnittstellen emulieren werden. Das im Bausatz enthaltene Bluetooth-Modul kann jedoch nur eine serielle Schnittstelle bereitstellen, was jedoch für den Einsatz vollkommen ausreichend ist.

Nach einem erfolgreichen Verbindungsaufbau, sind folgende Parameter für eine Kommunikation notwendig:

- Baud-Rate: 19200
- Parität: 8N1

4.1.1 BlueZ 4.x (Debian/Ubuntu/Raspbian)

Zunächst müssen die benötigten Pakete installiert werden.

```
$ sudo apt-get update
$ sudo apt-get install bluetooth bluez-utils
...
[ ok ] Starting bluetooth: bluetoothd rfcomm.
...
```

Das Bluetooth-Gerät sollte vom Computer richtig erkannt werden. Hier wird davon ausgegangen, dass das Bluetooth-Gerät per USB angeschlossen ist. In der Liste der Geräte ist der Bluetooth Adapter aufgeführt.

```
$ lsusb
```

```
...
```

```
Bus 001 Device 004: ID 1131:1001 Integrated System Solution Corp. KY-BT100 Bluetooth Adapter
```

Sieht soweit alles funktionstüchtig aus, kann nach neuen Bluetooth-Geräten in der Umgebung gesucht werden. Das Beispiel Bluetooth-Modul meldet sich mit der MAC 11:22:33:44:55:66 und dem Namen HC-06.

```
$ hcitool scan
```

```
Scanning ...
```

```
11:22:33:44:55:66 HC-06
```

Bei den ersten Versuchen bietet es sich an, eine Sicherheitskopie der Konfiguration anzulegen.

```
$ cp /etc/bluetooth/rfcomm.conf rfcomm.conf.orig
```

Ist das geschafft, sollte der Konfiguration nichts mehr im Weg stehen. Hierfür wird die Konfigurationsdatei wie folgt angepasst. Wichtig ist, dass die Adresse von dem Gerät angegeben wird, welches beim Scannvorgang erkannt wurde.

```
#
```

```
# RFCOMM configuration file.
```

```
#
```

```
rfcomm0 {
```

```
    # Automatically bind the device at startup  
    bind yes;
```

```
    # Bluetooth address of the device  
    device 11:22:33:44:55:66;
```

```
    # RFCOMM channel for the connection  
    channel 1;
```

```
    # Description of the connection  
    comment "Arduino";
```

```
}
```

Da für die Kommunikation mit dem Modul eine Pin erforderlich ist, muss diese angegeben werden. In diesem Fall handelt es sich um die Pin 1234.

```
$ echo "11:22:33:44:55:66 1234" | sudo tee /var/lib/bluetooth/66\:55\:44\:33\:22\:11/pincode  
11:22:33:44:55:66 1234
```

Anschließend ist ein Neustart des Bluetooth-Dienstes notwendig, damit die Konfiguration übernommen wird.

```
$ sudo service bluetooth restart
[ ok ] Stopping bluetooth: rfcomm /usr/sbin/bluetoothd.
[ ok ] Starting bluetooth: bluetoothd rfcomm.
```

Ist dies geschafft, sollte ein neuer serieller Port `/dev/rfcomm0` verfügbar sein.

Achtung: Zunächst sind das Bluetooth-Gerät des Computers und das Modul noch nicht miteinander verbunden. Die Verbindung wird erst aufgebaut, wenn der serielle Port `/dev/rfcomm0` geöffnet wird. Die Verbindung bleibt so lange bestehen, bis der serielle Port wieder geschlossen wird.

4.1.2 BlueZ 5.x (Arch Linux, Fedora 20, ...)

Bevor mit der Installation und Konfiguration begonnen wird, sollte geprüft werden, ob der Nutzer in der `uucp` Gruppe ist. Dies ist notwendig, damit er später auf die serielle Schnittstelle zugreifen kann.

```
$ sudo usermod -a -G uucp phibo
```

Tipp: Ist der Nutzer während er zur Gruppe hinzugefügt wird angemeldet, sollte er sich zur Sicherheit noch einmal ab- und wieder anmelden, damit der Zugriff möglich wird.

Im nächsten Schritt wird die Installation aller Pakete durchgeführt.

```
$ sudo pacman -S bluez bluez-utils
```

Anschließend wird der Bluetooth Dienst gestartet.

```
$ sudo systemctl start bluetooth
```

Damit dieser beim Systemstart automatisch gestartet wird, kann folgender Befehl ausgeführt werden.

```
$ sudo systemctl enable bluetooth
```

Jetzt kann mit Hilfe der **bluetoothctl** Anwendung per Konsole nach Bluetooth-Geräten gesucht werden. Dazu wird die Anwendung gestartet und die Ausgabe signalisiert, dass der lokale Computer bereits erkannt wurde.

```
$ bluetoothctl
[NEW] Controller 66:55:44:33:22:11 arch-linux [default]
```

Anschließend wird der Agent registriert.

```
[bluetooth]# agent on
Agent registered
```

Danach wird der Bluetooth Adapter bzw. Dongle aktiviert.

```
[bluetooth]# power on
[CHG] Controller 66:55:44:33:22:11 Class: 0x00010c
Changing power on succeeded
[CHG] Controller 66:55:44:33:22:11 Powered: yes
```

War bis jetzt alles erfolgreich, kann jetzt mit der Suche nach Bluetooth-Geräten in der Umgebung begonnen werden. Zu sehen ist, dass das gewünschte Gerät mit dem Namen **HC-06** gefunden wurde.

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 66:55:44:33:22:11 Discovering: yes
[NEW] Device 11:22:33:44:55:66 11-22-33-44-55-66
[CHG] Device 11:22:33:44:55:66 LegacyPairing: no
[CHG] Device 11:22:33:44:55:66 Name: HC-06
[CHG] Device 11:22:33:44:55:66 Alias: HC-06
[CHG] Device 11:22:33:44:55:66 LegacyPairing: yes
```

Nach einem kurzen Moment kann mit dem Befehl **devices** eine Liste der gefundenen Bluetooth-Geräte angezeigt werden.

```
[bluetooth]# devices
Device 11:22:33:44:55:66 HC-06
```

Anschließend ist es möglich die Verbindung zwischen Computer und Bluetooth-Gerät anzulegen. Bei diesem Schritt ist bei den meisten Geräten die Eingabe einer PIN notwendig. Jedoch muss dieser Schritt pro Gerät nur beim ersten Mal durchgeführt werden.

```
[bluetooth]# pair 11:22:33:44:55:66
Attempting to pair with 11:22:33:44:55:66
[CHG] Device 11:22:33:44:55:66 Connected: yes
Request PIN code
```

```
[agent] Enter PIN code: 1234
[CHG] Device 11:22:33:44:55:66 UUIDs has unsupported type
Pairing successful
[CHG] Device 11:22:33:44:55:66 Connected: no
```

War bis hier hin alles erfolgreich, kann die serielle Schnittstelle über Bluetooth angelegt werden.

```
$ sudo rfcomm bind 0 11:22:33:44:55:66
```

Anschließend sollte die serielle Schnittstelle als `/dev/rfcomm0` zur Verfügung stehen.

Hinweis: Zunächst sind der Computer und das Bluetooth-Modul noch nicht miteinander verbunden. Die Verbindung wird erst aufgebaut, wenn der serielle Port `/dev/rfcomm0` geöffnet wird. Die Verbindung bleibt so lange bestehen, bis der serielle Port wieder geschlossen wird.

4.2 ArduRPC

Zur Ansteuerung kommt das ArduRPC Protokoll zum Einsatz. Die Matrix meldet sich als Matrix-Display vom Typ Extended Matrix und ist mit dieser API kompatibel. Es können daher alle Beispiele, die für den Matrix Typ verfügbar sind, verwendet werden.

Es steht eine Python Bibliothek zur Verfügung, die eine einfache Ansteuerung ermöglicht.

4.2.1 Installation

Aktuelle Version der ArduRPC Python Bibliothek von GitHub¹ herunterladen und entpacken. Anschließend folgende Befehle ausführen:

```
$ python setup.py build
$ sudo python setup.py install
```

Hinweis: Entwickelt wurde die Bibliothek für Python 3.x, es sollte jedoch auch mit Python 2.7 laufen.

Für den Verbindungsaufbau wird zusätzlich noch pySerial² benötigt.

¹<https://github.com/DinoTools/ArduRPC-python>

²<http://pyserial.sourceforge.net/>

4.2.2 Verwendung

Wie der Verbindungsaufbau und die Ansteuerung im Detail funktionieren, kann den verfügbaren Beispielen entnommen werden.

Chapter 5

Software

In diesem Abschnitt wird beschrieben, wie die Firmware angepasst und erweitert werden kann. Der im Folgenden beschriebene Vorgang ist für den reinen Betrieb der CLTMatrix jedoch nicht notwendig, da die benötigte Firmware bereits vor dem Workshop auf den Mikrocontroller aufgespielt wurde.

5.1 Programmierung

Um auch Anfängern einen einfachen Einstieg zu ermöglichen, wurde beim Entwurf des Boards und der Software darauf geachtet, dass Beides kompatibel zur Arduino IDE ist. Die Programmierung kann daher direkt mit der Arduino IDE über das Bluetooth-Modul oder über einen zusätzlichen USB auf Seriell Adapter erfolgen.

5.2 Bootloader

Bei dem verwendeten Bootloader handelt es sich um eine leicht angepasste Version des optiboot¹ Bootloaders. Er belegt weniger als 512 Byte im Flash-Speicher und unterstützt die für die Programmierung notwendigen Befehle des STK500 Protokolls.

5.3 Arduino IDE

Zum Erstellen der Firmware werden folgende Bibliotheken benötigt:

¹<https://code.google.com/p/optiboot/>

- [DHT](#)²
- [Adafruit_GFX](#)³
- [CLTMatrix_GFX](#)⁴
- [ArduRPC](#)⁵
- [ArduRPC.CLTMatrix_GFX](#)⁶
- [ArduRPC_Adafruit_DHT](#)⁷

Da die CLTMatrix im Gegensatz zu momentan verfügbaren Arduino-Boards statt mit 8 oder 16 MHz mit 20 MHz betrieben wird, ist das Einspielen einer zusätzlichen Board-Konfiguration notwendig. Hierfür muss das hardware-Verzeichnis aus den CLTMatrix Quellen⁸ in das entsprechende Arduino-Verzeichnis kopiert werden. Mit der Arduino IDE 1.5 unter Linux ist es im Home-Verzeichnis das Verzeichnis `~/Arduino/hardware`.

War alles erfolgreich, kann das CLTMatrix Board im Menü unter *Werkzeuge* \rightarrow *Boards* \rightarrow *CLTMatrix* ausgewählt werden. Gegebenenfalls muss ein Neustart der Anwendung durchgeführt werden, bevor das CLTMatrix Board und die Bibliotheken verfügbar sind.

²<https://github.com/adafruit/DHT-sensor-library>
³<https://github.com/adafruit/Adafruit-GFX-Library>
⁴<https://github.com/DinoTools/CLTMatrix>
⁵<https://github.com/DinoTools/ArduRPC>
⁶<https://github.com/DinoTools/CLTMatrix>
⁷<https://github.com/DinoTools/ArduRPC-handlers>
⁸<https://github.com/DinoTools/CLTMatrix>