

Hacklace Software

Das Hacklace ist bereits mit dem Arduino-Bootloader und der Firmware vorprogrammiert. Du kannst es also gleich nach dem Zusammenbau verwenden. Wie du das Hacklace nach deinen Wünschen anpassen kannst, zeigen dir die folgenden Abschnitte.

Für die Programmierung gibt es eine eigene Seite.

Aufspielen eigener Texte und Animationen



Hacklace2 mit Hack-Key

Eigene Texte und Animationen lassen sich als Textdatei auf das Hacklace laden. Dabei bleibt der Programmcode auf dem Hacklace unverändert. Es wird nur der EEPROM-Speicher im Hacklace beschrieben. Verbinde dein Hacklace über einen USB-Seriell-Wandler mit dem PC oder Laptop (siehe "[Verbindung mit dem PC](#)"). Für den Einstieg kannst du die [Standard-Konfigurationsdatei](#) anpassen und auf das Hacklace übertragen.

Die ersten Schritte zeigt dir das Video.



Video

Zur Konvertierung gängiger Bildformate gibt es im [Downloadbereich](#) ein geeignetes Tool.

Download-Modus

Bevor das Hacklace Daten akzeptiert, muss es in den Download-Modus versetzt werden. Dies kann

auf zwei Arten erfolgen:

- Man löst einen Reset aus. Bei Verwendung des **Hack-Key** geschieht dies automatisch, wenn man die serielle Verbindung öffnet.
- Während des Einschaltens des Hacklace wird der rechte Taster gedrückt gehalten.

Befindet sich das Hacklace im Download-Modus erscheint im Display „DL“. Nun kann man Zeichen an das Hacklace senden. Jedes empfangene (druckbare) Zeichen wird zur Kontrolle im Display angezeigt. Wenn der EEPROM-Speicher voll ist, wird bei weiteren Zeichen im Display „EE“ angezeigt (Speicherüberlauf-Fehler). Der Download-Modus kann jederzeit durch Drücken der linken Taste verlassen werden. Das Hacklace schaltet dann in den normalen Betriebsmodus zurück und zeigt den ersten Text bzw. die erste Animation an.

Programme für die serielle Schnittstelle

Es gibt zahlreiche Programme, mit denen sich über die serielle Schnittstelle Daten an das Hacklace verschicken lassen. Für erste Versuche eignet sich der „Serial Monitor“, der in die Arduino-Entwicklungsumgebung integriert ist. Nach dem Start (Tools → Serial Monitor) muss rechts unten noch die Baudrate auf 2400 Baud eingestellt werden und schon kannst du Kommandos an das Hacklace senden.

Um ganze Dateien auf das Hacklace zu übertragen, können u. a. folgende Programme hilfreich sein.

- Hyperterminal (Windows XP)
- Coolterm (Windows, Linux, OS X)
- HTerm (Windows 7 / 8)

Unter Unix-artigen Betriebssystemen kann man auch die Kommandozeile benutzen. Dazu stellst du zunächst die Übertragungsparameter ein und sendest dann z. B. mit „printf“ einen String. (Bei manchen Betriebssystemen muss „-f“ durch „-F“ ersetzt werden.)

```
stty -f /dev/ttyUSB0 2400
printf '\eHL\\00 85 01\\ Hallo Welt!  \\00 FF\\' > /dev/ttyUSB0
```

Kommandos

Das Escape-Zeichen <ESC> (Charatercode 27) hat immer zur Folge, dass der Empfangsteil des Hacklace zurückgesetzt wird. Es empfiehlt sich daher, jede Übertragung mit einem <ESC> zu beginnen. Danach folgt ein Kommando aus zwei Großbuchstaben. Es gibt folgende Kommandos:

- „HL“ (Hacklace) - Übertragen von Texten und Animationen in den EEPROM-Speicher.
- „RD“ (restore defaults) - Beschreibt das EEPROM mit dem vorgefertigten Standardinhalt.
- „VN“ (version number) - Zeigt die Versionsnummer der Firmware auf dem Display an.

Datenformat

Der EEPROM-Speicher enthält eine Liste von App-Einträgen. Im normalen Betrieb folgt das Hacklace dieser Liste, wenn man mit der linken Taste durch die einzelnen Apps blättert. Jeder Eintrag enthält im

ersten Byte die sogenannte App-ID, eine Nummer, welche die aufzurufende App identifiziert. Die darauf folgenden Bytes können von der App als Parameter gelesen und ausgewertet werden. Wieviele Bytes gelesen werden und welche Bedeutung sie haben, hängt von der jeweiligen App ab (siehe App-Beschreibung).

Wichtig: Das Ende der Liste muss durch die App-ID 255 (hexadezimal FF) angezeigt werden, sonst wird man beim Blättern irgendwann die Fehlermeldung „bad ID“ erhalten.

Speichern von Zeichen

Sobald das „HL“-Kommando empfangen wurde, werden alle nachfolgenden druckbaren Zeichen (Charactercode ≥ 32) im EEPROM abgelegt. Nicht-druckbare Zeichen (Charactercode < 32) werden dabei jedoch ignoriert. So kann man sein Konfigurationsfile z. B. durch Tabulatoren oder leere Zeilen übersichtlicher gestalten.

Eingabe von Hex-Werten

Daten lassen sich nicht nur als Zeichen, sondern auch als Hexadezimalwerte eingeben. Dazu müssen sie als zweistellige Hexadezimalzahlen angegeben werden, die von '\' eingeschlossen sind. Mehrere Bytes müssen innerhalb der '\' durch Leerzeichen (genauer: Nicht-Hexadezimalzeichen) getrennt werden. Durch die Hex-Eingabe ist es möglich, auch die Codes für nicht-druckbare Zeichen oder die Codes von Animationen im EEPROM abzulegen. Möchte man den Rückwärtsschrägstrich dagegen als normales Zeichen verwenden, so muss dieser doppelt gesendet werden. Beispiele:

- Die Zeichenfolge 'JKLM' kann man auch als '\4a 4b 4C 4D\' eingeben.
- Mit '\05 06 07 08\' werden die Sonderzeichen chr(5) bis chr(8) erzeugt, welche das Hacklace als Space-Invader-Monster darstellt.
- Die Sequenz '\\' führt dazu, dass das Zeichen '\' = chr(92) einmal im EEPROM abgelegt wird.

Kommentare

Die Kombination '\#' bewirkt, dass das Hacklace den Rest der aktuellen Zeile komplett ignoriert. Auf diese Weise lassen sich einzeilige Kommentare realisieren.

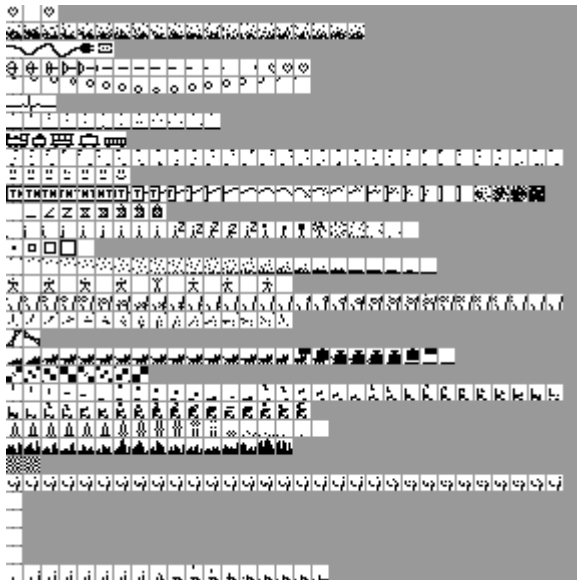
Wichtig: Kommentare müssen durch Tabulatoren und nicht durch Leerzeichen vom Rest der Zeile abgesetzt werden. Leerzeichen sind druckbare Zeichen und werden somit ins EEPROM übernommen werden, was zu einem „bad ID“-Fehler führt.

Zeichensatz



Das Hacklace besitzt einen internationalen Zeichensatz nach ISO8859-15. Damit lassen sich die westeuropäischen Sprachen gut darstellen. Es handelt sich um einen proportionalen Zeichensatz, d. h. die Zeichen besitzen unterschiedliche Länge, z. B. ist ein 'i' kürzer als ein 'w'. Außerdem stehen noch einige Sonderzeichen zur Verfügung. Der Zeichensatz gliedert sich wie folgt:

- Zeichen 0: Systeminternes Steuerzeichen (bitte nicht verwenden)
- Zeichen 1 - 30: Durch den Benutzer verwendbare Sonderzeichen
- Zeichen 31: Systeminternes Steuerzeichen (bitte nicht verwenden)
- Zeichen 32 - 127: Standard-ASCII-Zeichen
- Zeichen 128 - 159: Daten für fertige Animationen
- Zeichen 160 - 255: Nationale Zeichen gemäß ISO8859



fertige Animationen

Apps

Das Hacklace kann bis zu 32 Apps verwalten. Jede App ist durch ihre App-ID eindeutig gekennzeichnet. Die App-ID wird durch den Eintrag in der HL_AppRegistry.h bestimmt. Im

Auslieferungszustand sind bereits einige fertige Apps vorhanden, die als Grundlage für eigene Entwicklungen dienen können.

Animation-App

App-ID	Speed + Delay	Dir + Inc	Text / Animationsdaten	Ende-Flag
0	1 Byte	1 Byte	n Bytes	0

Dies ist sicherlich die am häufigsten verwendete App, denn sie sorgt für die Darstellung von Laufschriften und Animationen. Während die App läuft, kann mit dem rechten Taster die Helligkeit erhöht werden.

Parameter

- Der Speed+Delay-Parameter enthält im oberen **Nibble** die Scrollgeschwindigkeit. Das untere Nibble gibt an, wie lange die Pause am Ende eines Scrolldurchgangs ist. Die Pausendauer ist proportional zur Scrollgeschwindigkeit.
- Das obere Nibble des Dir+Inc-Parameters legt die Scrollrichtung fest. Es kann die Werte 0 (vorwärts), 1 (rückwärts) und 2 (hin und her) annehmen. Das untere Nibble bestimmt die Schrittweite beim Scrollen. Sinnvolle Werte sind 1 für Laufschriften und 8 für Animationen.
- Der Datenbereich kann unterschiedlich lang sein. Er muss durch eine Null am Ende abgeschlossen werden (Ende-Flag). Jedes Byte wird als Zeichencode interpretiert. Mit Hilfe des Zeichensatzes wird die Darstellung des Zeichens in den Bildschirmspeicher geladen. Man kann Daten auch direkt in den Bildschirmspeicher schreiben ohne den Zeichensatz zu benutzen (z. B. für Animationen). Dazu wird der Datenblock durch den Hex-Wert 1F eingeleitet. Das nächste Byte gibt an, wieviele Datenbytes nun folgen (maximal 255). Anschließend muss genau diese Anzahl von Datenbytes aufgeführt werden. Jedes Byte repräsentiert eine Displayspalte. Die acht Bits entsprechen den acht Pixeln dieser Spalte. Dabei steht Bit0 für das oberste und Bit7 für das unterste Pixel.

SubOptimal hat ein [Kommandozeilentool](#) geschrieben, das übliche Grafikformate in einen Hacklace2-String übersetzt. Man kann das Tool direkt nutzen oder den Quellcode als Beispiel für eigene Entwicklungen heranziehen.

Von Johannes wurde eine [Excel- bzw. Calc-Tabelle](#) entwickelt, mit der man auf einfache Weise Pixelgrafiken erstellen und daraus die passenden Hex-Codes erzeugen kann.

Beispiel

```
\00 7A 01\It's hip to be \1F 08 00 7E 42 42 42 42 7E 00\! \00 FF\
```

Dieser String wird wie folgt interpretiert:

```
00          --> App-ID (Animation-App)
7A          --> Speed=7, Delay=10 (= A hex)
01          --> vorwärts scrollen (0), jeweils 1 Spalte pro
Schritt (1)
"It's hip to be " --> wird als Text übernommen
```

```

1F      --> Kennzeichen für einen Datenblock
08      --> Anzahl (es folgen noch 8 Datenbytes)
00 7E 42 42 42 42 7E 00 --> Daten (ergeben ein Quadrat)
"! "    --> wird als Text übernommen
00      --> Ende-Flag
FF      --> App-ID 255 = Ende der App-Liste

```

Spiral-App

App-ID
1

Diese App erzeugt eine Spirale. Sie hat keine Parameter.

Counter-App

App-ID
2

Ein einfacher Handzähler. Durch Drücken der rechten Taste zählt man um eins hoch. Durch längeres Drücken der rechten Taste löscht man den Zählerstand wieder. Die App hat keine Parameter.

Game-of-Life-App

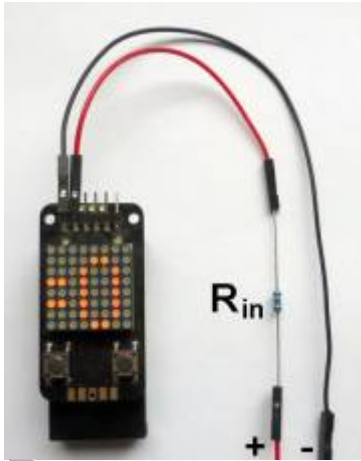
App-ID	Iterationen	Intervall	Start Konfiguration
3	1 Byte	1 Byte	8 Bytes

Eine Implementierung von Conway's [W "Game of Life"](#)

Parameter

- Der erste Parameter gibt an, wieviele Durchläufe simuliert werden sollen, bevor das Spiel wieder von neuem mit der Startkonfiguration beginnt.
- Der Intervall-Parameter bestimmt das Zeitintervall zwischen zwei Simulationsschritten. Es beträgt Intervall * 10 Millisekunden.
- Die Startkonfiguration wird durch 8 Bytes angegeben. Jedes Byte entspricht einer Spalte. Die einzelnen Bits geben den Zustand der Zelle an (1 = lebend, 0 = tot).

Voltmeter-App



Hacklace als Voltmeter

App-ID Eingangswiderstand	
4	1 Byte

Mit dieser App lassen sich Spannungen messen. Das Bezugspotential wird mit Pin 1 des Kommunikationsports (GND) verbunden. Die unbekannte Spannung legt man über einen Widerstand an Pin 2. Die Größe des Widerstands wird als Parameter angegeben. Dieser Widerstand bildet zusammen mit R14 (= 10 k Ω , siehe [Schaltplan](#)) einen Spannungsteiler und legt somit den Messbereich fest. Ohne Vorwiderstand ist der Eingangsbereich des A/D-Wandlers 0 - 1,1 Volt. Bei einem Vorwiderstand von 39 k Ω ergibt sich ein Messbereich von 0 - 5 Volt. Zur Absicherung des Eingangs empfiehlt es sich, zwischen Pin 1 und Pin 2 eine 3,6-Volt-Zenerdiode zu legen (Kathode an Pin 2).

Parameter

- Größe des Vorwiderstands (in k Ω)

Battery-Monitor-App



Zwei Widerstände zur Überwachung der Batteriespannung

App-ID
5

Mit dem Batteriemonitor kann das Hacklace seine eigene Batteriespannung überwachen. Je nach Spannungswert wird ein mehr oder weniger ausgefülltes Batteriesymbol angezeigt. Damit es funktioniert müssen am Sensorport noch zwei Widerstände angelötet werden. Am besten passen SMD-Widerstände der Bauform 0805. Zwischen Pin 1 und Pin 2 kommen 22 k Ω , zwischen Pin 4 und Pin 5 10 k Ω .

Frequencymeter-App

App-ID
6

Wie der Name schon sagt, lassen sich mit dieser App Frequenzen messen. Das Signal sollte rechteckig sein und wird an Pin 5 des Sensorports (PD3) erwartet. Der Arbeitsbereich geht von 16 Hz bis ca. 10 kHz. Die App hat keine Parameter.

Thermometer-App

App-ID
7

Mit nur einem Widerstand und einem NTC wird das Hacklace zum Thermometer. Die Temperaturanzeige erfolgt zweistellig in Grad Celsius. Die App hat keine Parameter.

Stopwatch-App

App-ID
8

Die rechte Taste startet und stoppt die Stoppuhr. Die Uhr zählt Sekunden und zeigt die Zehntelsekunden als Bargraph an. Ein langer Druck setzt die Uhr auf Null zurück. Die App hat keine Parameter.

Speedometer-App

App-ID	Radumfang
9	2 Bytes

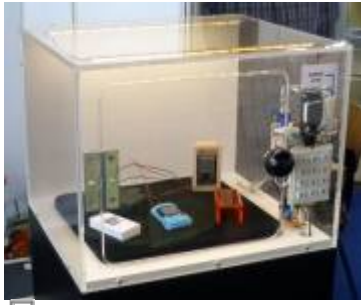
Ermittelt die Pulsfrequenz an Pin 5 des Sensorports und rechnet sie in eine Geschwindigkeit um. Dazu wird als Parameter der Radumfang in Millimetern als 16-Bit-Wert (Low-Byte, High-Byte) angegeben.

Jukebox-App

App-ID
12

Spielt Lieder über den PWM-Ausgang am Sensorport ab (Pin 5). Die rechte Taste schaltet zum nächsten Lied weiter. Die App hat keine Parameter.

Codelock-App



Ein Safe mit Hacklace-Codeschloss

App-ID	Anzahl gültiger Codes	Codes
20	1 Byte	jeweils 2 Bytes

An den Analogeingang des Sensorports wird eine Zehnertastatur angeschlossen. Der digitale Ausgang erzeugt ein PWM-Signal, mit dem sich ein Servo ansteuern lässt. Zum Öffnen muss man einen vierstelligen Code und dann „#“ eingeben. Nach drei Fehlversuchen wird eine Pause eingelegt. Ist der Code korrekt, entriegelt der Servo die Tür. Drücken von „*“ verriegelt die Tür wieder. Jeder gültige Code wird als 2-Byte-Parameter gespeichert wobei jedes **Nibble** eine der vier Ziffern repräsentiert. Standardcodes zum Öffnen sind 1234 und 9999.

Displaytest-App

App-ID
21

Es werden nacheinander alle Zeilen und Spalten des Display eingeschaltet. Dies wiederholt sich bei minimaler und maximaler Helligkeit. Die App hat keine Parameter.

Example-App

App-ID
30

Die Example-App hat keine Parameter. Sie erzeugt eine einfache Grafik (Lauftext aus ASCII-Zeichen), deren Helligkeit pulsiert. Die Example-App dient als Beispiel und Vorlage zur Entwicklung eigener Apps.

Reset-App

App-ID
31

Die Reset-App ist eigentlich keine App, sondern nur ein Platzhalter. Die App, welche auf der Position der Reset-App (App-ID 31) eingetragen ist, wird nach einem System-Reset ausgeführt. Normalerweise ist das die Download-App. Zum Beispiel wird mit dem Aufbau der seriellen Verbindung über den Hack-Key ein Reset ausgelöst, was dann automatisch die Download-App startet. Die Reset-App muss eine parameterlose App sein.

Display-App

Mit der Display-App kann das Hacklace als serielles Display verwendet werden. Baudrate (9600 baud) und Scrollgeschwindigkeit lassen sich im Sourcecode der App leicht anpassen. Alle Zeichen, die das Hacklace empfängt, werden als String auf dem Display angezeigt. <CR> oder <LF> löschen den Bildschirm. Ein Raw-Modus steht auch zur Verfügung. Stellt man einem Byte den Wert 0x1F voran, wird das Byte nicht als Zeichencode interpretiert, sondern direkt in den Bildschirmspeicher geschrieben. Der Display-App ist keine App-ID zugeordnet. Oft wird es sinnvoll sein, sie auf der Position der Reset-App (App-ID 31) einzutragen, damit das Hacklace gleich nach dem Reset als Display arbeitet.

Programmierung

Du möchtest noch mehr als nur Laufschriften und Animationen erstellen? Dann schreib dir deine eigene App oder lerne, wie du die Hacklace-Firmware verändern kannst. Hierzu benötigst du die [Arduino-Entwicklungsumgebung](#). Außerdem empfiehlt sich der Einsatz eines [Hack-Key](#) oder eines anderen USB-Seriell-Wandlers mit FT232-Chip. Alles weitere wird auf der Seite „[Hacklace - Programmierung](#)“ beschrieben.

[Hacklace](#), [Elektronik](#), [Bausatz](#), [Software](#)

From:

<http://www.doku.fab4u.de/> - **fab4U**

Permanent link:

<http://www.doku.fab4u.de/de/kits/hacklace/software>

Last update: **2018/07/03 21:12**

