

LittleWire

Mikrokontroller mit Ruby - Teil 2

28. November 2014

Andreas Steinel
A.Steinel@gmail.com



Schon wieder Hardware?

Ja ... IoT ist im kommen

"According to Gartner:
there will be nearly 26 billion devices on the Internet of Things by 2020."

Gartner-Quelle <http://www.gartner.com/newsroom/id/2636073>

2

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

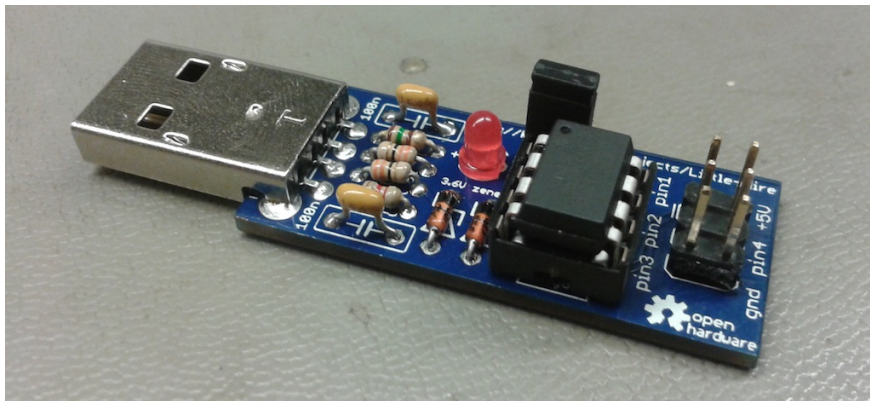
Schluß

Literatur

Das ist LittleWire?

Hardware

Zum einen das Referenzdesign und die Hardwareimplementierung von Ihsan Kehribar



Bildquelle: <http://littlewire.cc/>

Hardwareübersicht

Schon wieder?

3

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

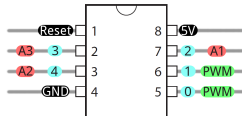
Literatur

Was treibt das ganze an?

Hardware

Andreas Steinel
LittleWire und Ruby

- ▶ ATTiny45 bzw. ATTiny85
- ▶ 256/512 Bytes EEPROM nichtflüchtiger Speicher (100k Schreibzyklen)
- ▶ 256/512 Bytes RAM
- ▶ 6 Ein-/Ausgabeeleitungen - 4 ADC (10-bit), 6xPWM
- ▶ Schnittstellen
 - ▶ GPIO
 - ▶ Serial Peripheral Interface (SPI)
 - ▶ Inter-Integrated Circuit (I²C)



4

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Bildquelle: <https://learn.sparkfun.com/tutorials/tiny-avr-programmer-hookup-guide/attiny85-use-hints>

Wichtig ist das Schema

Hardware



Andreas Steinel
LittleWire und Ruby

Hardwareübersicht

Schon wieder?

5

Überblick

Anschlüsse

Rubyintegration

Wie gehts los?

Hello World

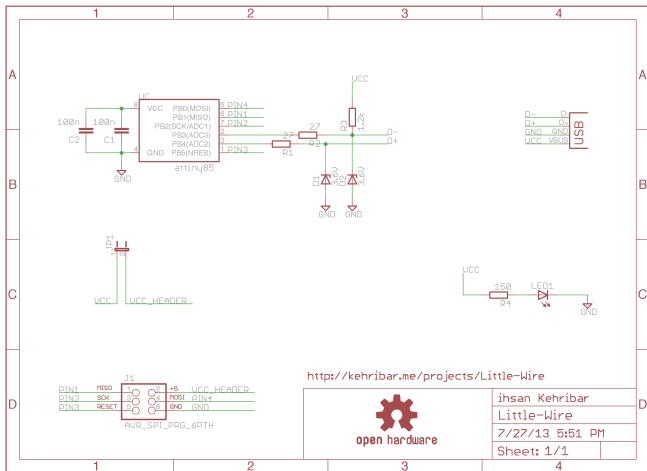
Fading

WS281

Ausblick

0110

Schluß



Bildquelle: <http://littlewire.cc/>

Neben Hardware gibts noch Software

Software



Andreas Steinel
LittleWire und Ruby

Zur Hardware gehört natürlich die Software (auf μC):

- ▶ 2012 von Ihsan Kehribar entworfen und über Seeedstudio vertrieben
- ▶ Kann auch selbst geflasht werden, alles OpenSource
- ▶ Software USB 1.1 mittels vUSB (Leute von LittleSnitch)
- ▶ LittleWire Firmware (Aktuell Version 1.3)
- ▶ Firmware dient der Schnittstelle zum Rechner, aber auch zur Peripherie
- ▶ Gerät funktioniert nur mit einem Rechner, da es selbst keine Programme speichert

Hardwareübersicht

Schon wieder?

Überblick

6

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Warum kein ...

Software



Andreas Steinel
LittleWire und Ruby

- ▶ Arduino?
- ▶ Raspberry Pi?
- ▶ Name-your-thing?

7

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

LittleWire bietet folgende Anschlußmöglichkeiten out-of-the-box:

- ▶ Vierkanal GPIO (digital)
- ▶ 2x 10-bit Analog-Digitalwandler
- ▶ 2x parallele Hardware PWMs (Digital-Analogwandler)
- ▶ 3x software PWMs
- ▶ SPI Protokoll
- ▶ Philips I^2C Protokoll
- ▶ Dallas 1-Wire Protokoll
- ▶ 4-Kanal WS2811 und WS2812 RGB LED Unterstützung
- ▶ USBTinyISP - Mikrocontrollerprogrammierer

Hardwareübersicht

Schon wieder?

Überblick

Software

8

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

- ▶ GPIO für
 - ▶ Blinken von LEDs
 - ▶ Auslesen von Schalterzuständen
 - ▶ manuelles (nicht zeitkritisches) Bit-Banging (z.B Sensoren)
- ▶ Analog-Digitalwandler zum Auslesen
 - ▶ von Potentiometern
 - ▶ Sensoren
 - ▶ Spannungsmessung (z.B. Akku)
- ▶ Digital-Analogwandler (Hardware oder Software)
 - ▶ Appleskes LED Fading
 - ▶ Servo-Ansteuerung
 - ▶ Spannungserzeugung

Hardwareübersicht

Schon wieder?

Überblick

Software

9

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Die Wesentlich spannenderen Dinge sind aber die High-Level-Protokolle, die einfach "nach oben" durchgereicht werden:

- ▶ Serial Peripheral Interface (SPI)
 - ▶ 4 Kanal Schnittstelle (ohne Strom, insgesamt 6 Leitungen)
 - ▶ eigenes Clock-Signal
- ▶ Inter Integrated Circuit (I^2C) Schnittstelle
 - ▶ 2 Kanal Schnittstelle (ohne Strom, insgesamt 4 Leitungen)
 - ▶ eigenes Clock-Signal
- ▶ 1-Wire
 - ▶ Geteilter Kanal inkl. Strom (insgesamt 2 Leitungen)

Warum das Ganze?

Anschlüsse



Andreas Steinel
LittleWire und Ruby

Warum brauch man diese High-Level-Protokolle?

- ▶ SPI (1 Geräte an LittleWire möglich)
RF-Module (RFM12, nRF24L, etc.), RTC, Displays, SD-Karten, Ethernet, Sensoren, EEPROMs, ISP
- ▶ I^2C (ein Bus-Master, 112 Geräte möglich)
Displays, Sensoren, EEPROMs
- ▶ 1-Wire (Ein Master, beliebig viele Geräte über adressierbare Clients mit 64-bit)
Sensoren, Türöffner, EEPROMs

Das schöne an allen Protokollen ist, dass die Taktung individuell erfolgt.

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

11

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Was wird wo angeschlossen?

Anschlüsse



Andreas Steinel
LittleWire und Ruby

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Silkscreen	Hard PWM	Soft PWM	ADC	I2C	SPI	1-Wire
pin1	CH_B	CH_B	-	-	DO	-
pin2	-	CH_C	CH_1	SCL	SCK	DATA
pin3	-	-	CH_0	-	CS	-
pin4	CH_A	CH_A	-	SDA	DI	-

Wie gehts los?

Rubyintegration



Andreas Steinel
LittleWire und Ruby

Entweder man kümmert sich in plain-old-C um die Kommunikation mit LittleWire, oder man nimmt einfach Ruby!

Mittels des `littlewire` Gem erfolgt die Integration, die auch die Abhängigkeiten `libusb` installiert.

Danach kann es losgehen direkt interaktiv rein

```
require 'littlewire'  
require 'pry'
```

```
wire = LittleWire.connect
```

```
pry
```

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

13

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Wie könnte es anders sein ... wir bauen als erstes 'Hello World' in Hardware - oder das Blink-Beispiel:

Wir bringen eine LED zum Leuchten (GPIO-Aufgabe):

- ▶ GPIO Pin SCK (Pin 2) als Ausgang konfigurieren
- ▶ Ausgang auf HIGH (also 1) setzen
- ▶ warten
- ▶ Ausgang auf LOW (also 0) setzen

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

14

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Hello World

Rubyintegration



Andreas Steinel
LittleWire und Ruby

```
require 'littlewire'

wire = LittleWire.connect
pin = :sck

wire.pin_mode pin, :out

loop do
  wire.digital_write pin, :vcc
  sleep 1
  wire.digital_write pin, :gnd
  sleep 1
end
```

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

15

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Einfaches Appleskes Fading einer LED mittels PWM

- ▶ Schleife iterieren
 - ▶ Fadewert mittels Sinus berechnen (Bytewert als Ergebnis)
 - ▶ Schreiben des Bytewerts an PWM Pin

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

16

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur


```
require 'littlewire'

wire = LittleWire.connect

FPS = 60
fader = 0.0

loop do
  wire.softpwm_c = (Math.sin(fader) + 1.0) * 100 + 27

  fader += 0.025
  sleep 1.0 / FPS
end
```

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

17

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

Sinnvolle und massentaugliche Anwendung von LEDs über LED Strips:

- ▶ feste Farbe (Rot, Grün, Blau, Warmweiß, Kaltweiß)
- ▶ RGB-LEDs (feste Farbe abwechselnd)
- ▶ RGB-LEDs (Tricolor LED), Strip steuerbar
- ▶ RGB-LEDs mit einzeln steuerbaren Farben

und genau den letzten Typ schauen wir uns an:

- ▶ einzeln ansteuerbar (sehr cool!!)
- ▶ aktuell noch sehr teuer (144 LED pro Meter, ca. 50 Euro)
- ▶ sehr Energieintensiv (35 W pro Meter)

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur

```
require 'littlewire'

wire = LittleWire.connect
pin = :mosi
speed_in_seconds = 0.025
num_leds = 5
hue = 0.0

loop do
  red = 1.0 + (Math.sin(hue) * 127)
  green = 1.0 + (Math.sin(hue + ((Math::PI*2.0) / 3)) * 127)
  blue = 1.0 + (Math.sin(hue - ((Math::PI*2.0) / 3)) * 127)

  wire.ws2811(pin).send([Colorist::Color.from_rgb(red, green, blue)] * num_leds)

  hue += 0.01
  sleep speed_in_seconds
end
```

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

19

WS2811/WS2812

Ausblick

Schluß

Literatur

Was haben wir erreicht?

- ▶ Simples LED Handling
- ▶ PWM für Servos anhand von LED gezeigt
- ▶ LED Strip bezwungen

Was steht noch auf dem Plan? Bzw. was machen wir das nächste Mal?

- ▶ SPI Bus programmierung
- ▶ I^2C programmierung
- ▶ Ruby Netzwerken über nRF24L01 2.4 GHz Netzwerk

Hardwareübersicht

Schon wieder?
Überblick
Software
Anschlüsse
Warum das Ganze?

Rubyintegration

Wie gehts los?
Hello World
Fading
WS2811/WS2812
Ausblick

Schluß

Literatur

- ▶ LittleWire Webseite
- ▶ LittleWire C-Dokumentation
- ▶ LittleWire Ruby GEM
- ▶ Difference between WS2801, WS2811 and WS2812

Hardwareübersicht

Schon wieder?

Überblick

Software

Anschlüsse

Warum das Ganze?

Rubyintegration

Wie gehts los?

Hello World

Fading

WS2811/WS2812

Ausblick

Schluß

Literatur