

Arduino

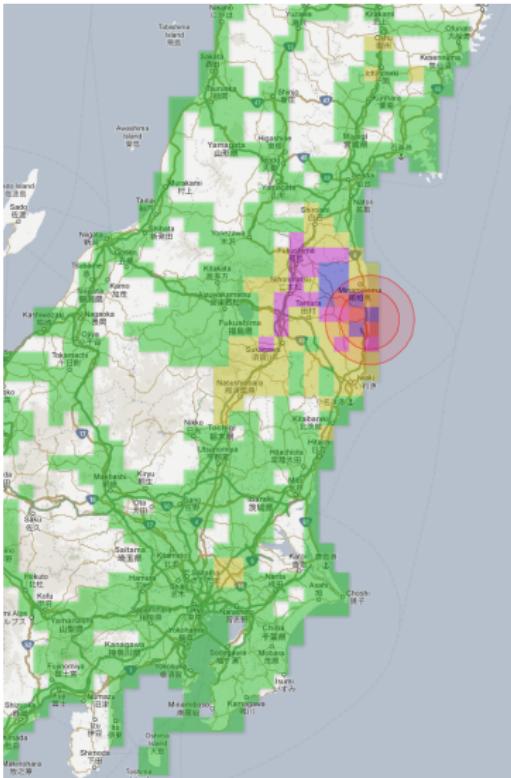
Einführung in die Mikrocontroller-Programmierung

Uwe Ziegenhagen

23. August 2013

Fukushima

- ▶ 11.03.2011: Erdbeben in Fukushima, Kernschmelzen
- ▶ öffentliche Daten „ungenau“, Prozess sehr mühselig
- ▶ Lösung: Arduino liest Werte vom Geigerzähler, liest GPS Koordinaten
- ▶ alle 5 Sekunden auf SD Karte geschrieben
- ▶ Ergebnis: unabhängige Strahlungsdaten in hoher Dichte, öffentlich zugänglich
- ▶ PBS Video bei Youtube
- ▶ mehr zum Thema



Überblick

Was ist „Arduino“?

- Allgemeines
- Hardware
- Software

Mein erstes Programm

Mehr Spaß mit LEDs

LC Displays anschließen

- DOGM-Displays
- I²C-Displays
- Serielle Displays

Mehr Beispiele

Mehr Informationen

Alternativen?

- ▶ lebe und arbeite seit Oktober 2008 in Köln
- ▶ habe BWL & Statistik studiert, beides mit IT-Fokus
- ▶ ⇒ „Elektronik funktioniert mit Rauch!“
- ▶ Blog unter www.uweziegenhagen.de, dort liegen auch die (in L^AT_EX erstellten) Folien zu diesem Vortrag
- ▶ Was ist L^AT_EX? Ein geniales Textsatzsystem, mit dem ihr eure Bachelor-, Master- und sonstigen Arbeiten schreiben wollt! Mehr am Stand von Dante e.V.!

Das Arduino Projekt

- ▶ Arduino = Physical-Computing-Plattform
- ▶ besteht aus Hard- und Software (beides Open Source)
- ▶ Hardware = einfache I/O-Boards mit Mikrocontroller und analogen/digitalen Ein- und Ausgängen
- ▶ Entwicklungsumgebung beruht auf Processing (Java-Dialekt) und Wiring (C-Dialekt)

Geschichte beginnt 2005 in Ivrea, Italien

- ▶ Massimo Banzi & David Cuartielles vom Interaction Design Institute Ivrea (IDI)
- ▶ Projekt, um Studenten **einfaches** Prototyping mit Mikrocontrollern zu ermöglichen
- ▶ Arduino = italienischer König, Name einer Bar
- ▶ mehr dazu in <http://vimeo.com/18539129>

Es gibt mittlerweile 19 Arten von Arduinos, unter anderem

- ▶ **Uno**, der Standard-Arduino (vormals „Duemilanove“)
- ▶ **LilyPad**, zum Auf- und Einnähen in Kleidung
- ▶ **Leonardo** ⇒ Maus/Tastatur
- ▶ **ArduinoBT**, mit eingebautem Bluetooth-Stack
- ▶ **Mega/Mega 2560**, mit mehr I/O Ports
- ▶ **Mini, Micro & Nano**, kleinste Arduinos
- ▶ dazu diverse Clone: Boarduino, Nano, Arduino Pro, etc.
- ▶ Spin-Offs: Netduino, Gadgeteer ⇒ Microsoft .Net

Wir konzentrieren uns auf den Arduino Uno!

Arduino Uno

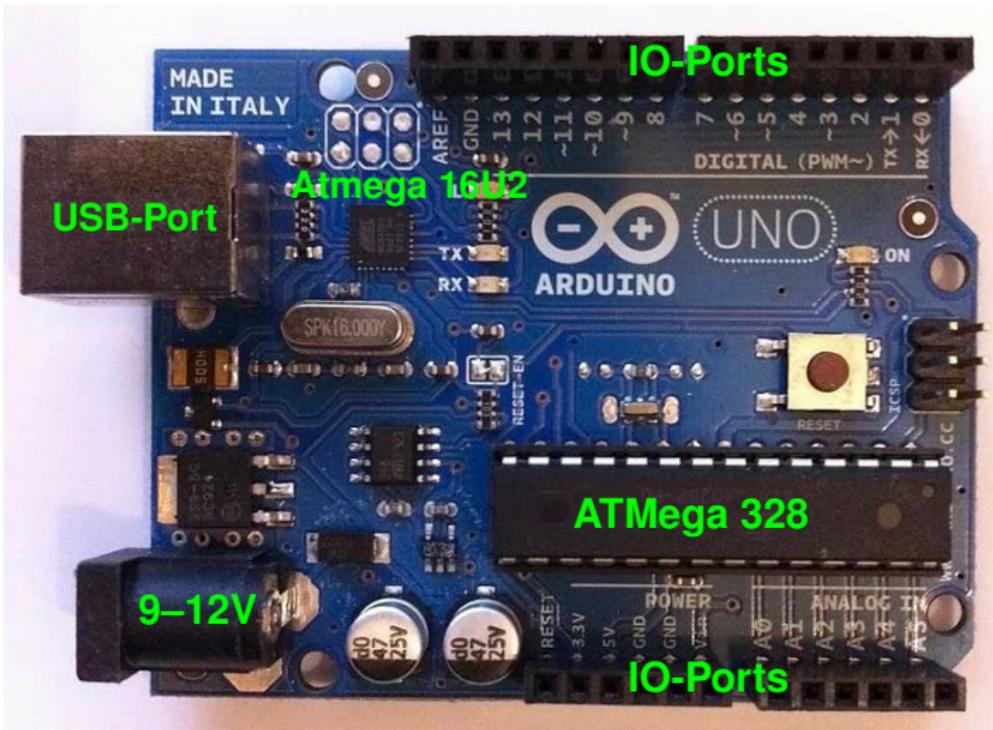


Abbildung : Arduino Uno Quelle: Wikipedia

Arduino Uno

- ▶ ATmega328 Mikrocontroller
- ▶ 32 KB Flash (davon 0,5 KB für Bootloader)
- ▶ 2 KB RAM, 1 KB EEPROM
- ▶ 16 MHz Takt
- ▶ 14 digitale I/O-Pins
- ▶ davon 6 als PWM ([Pulsweitenmodulation](#)) nutzbar
- ▶ 6 analoge Eingänge (10 Bit)
- ▶ On-Board USB-Schnittstelle (mit Atmega8U2 als Seriell-Wandler)
- ▶ 5 V Betriebsspannung, Speisung über USB oder Spannungsregler (7–12V)

- ▶ „Shields“ = Erweiterungsplatinen, die direkt auf den Arduino gesteckt werden können
- ▶ bereichern Arduino um viele interessante Möglichkeiten
- ▶ Liste der Shields sehr lang: shieldlist.org listet¹ 291 Shields von 116 Herstellern auf
- ▶ für Einsteiger gut geeignet: Danger Shield, Proto Shield, Ethernet Shield
- ▶ Ich habe noch: WLAN Shield, GPRS Shield, LoL-Shield

¹Stand: 31.05.2013

Danger Shield

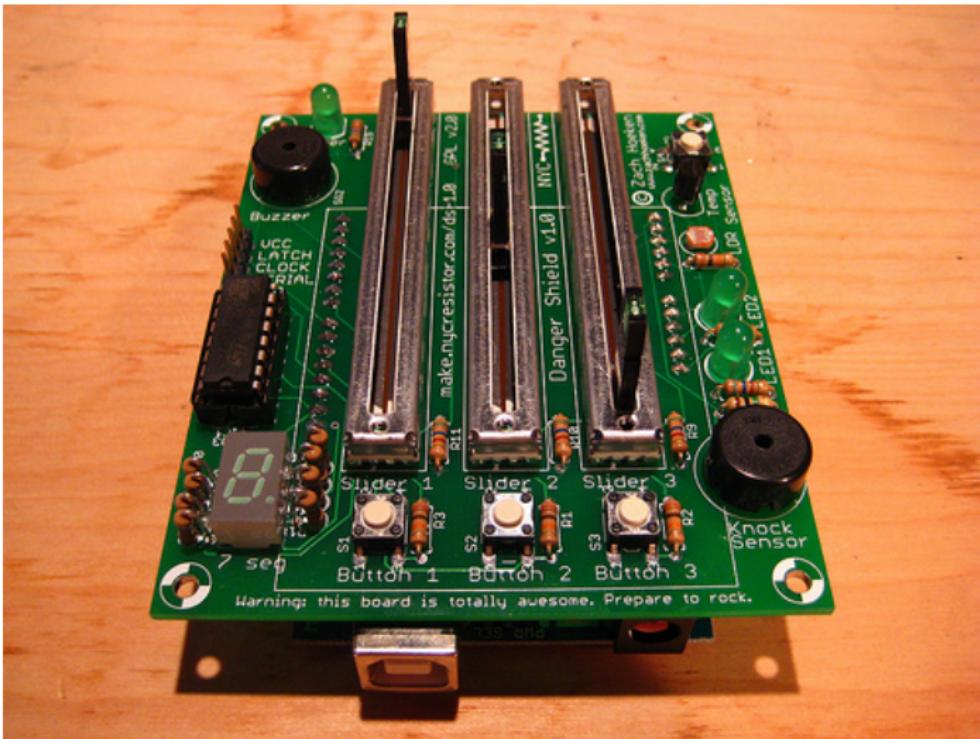


Abbildung : Das Danger Shield Quelle: www.zachhoeken.com

Proto Shield

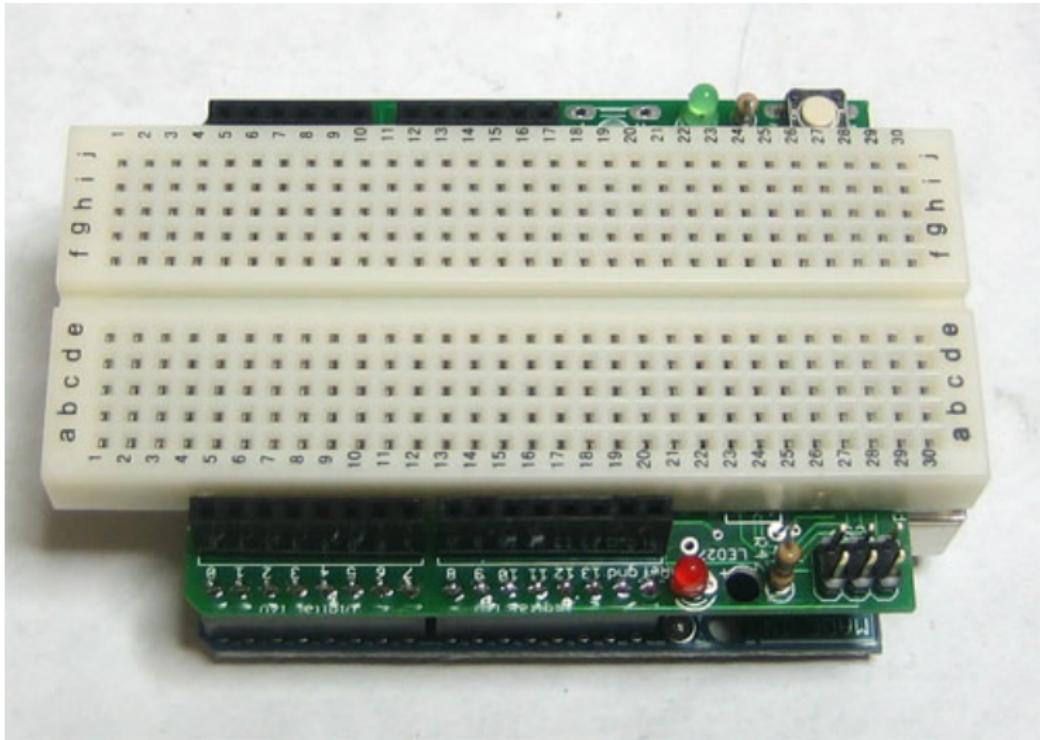


Abbildung : Das Proto Shield Quelle: www.ladyada.net

Steckbretter („Breadboards“)

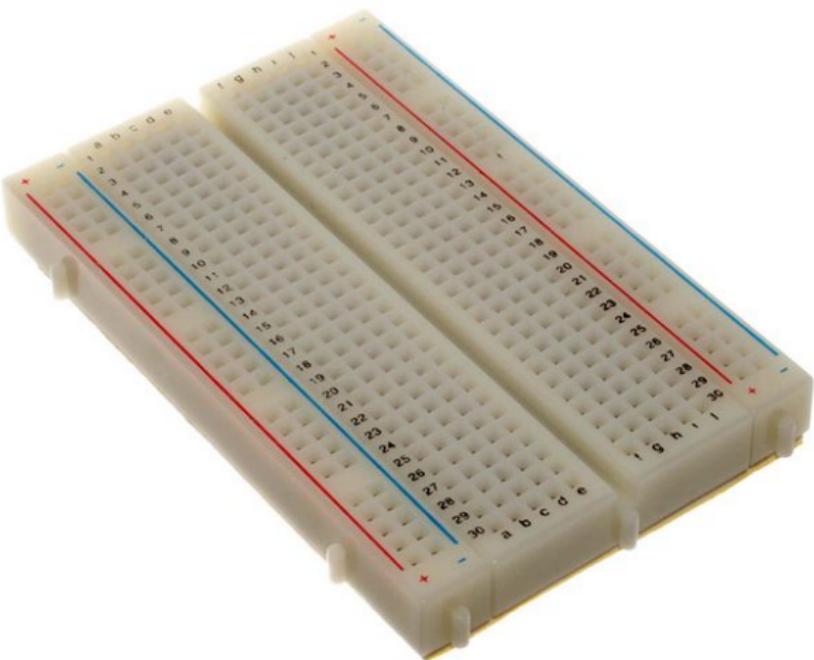


Abbildung : Breadboard, Quelle: Wikipedia

Ethernet Shield



Abbildung : Das Ethernet Shield mit μ SD-Slot Quelle: www.watterott.com

Arduino Software & Fritzing

- ▶ erhältlich für Windows, Linux, Mac
- ▶ aktuell momentan Version 1.0.5
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken
„sudo apt-get install arduino“

Am besten auch gleich „Fritzing“ installieren

- ▶ grafische Design-Umgebung
- ▶ kann auch Schaltplan und Leiterplatte erzeugen
- ▶ Projekt der FH Potsdam, auf Spenden angewiesen

Fritzing

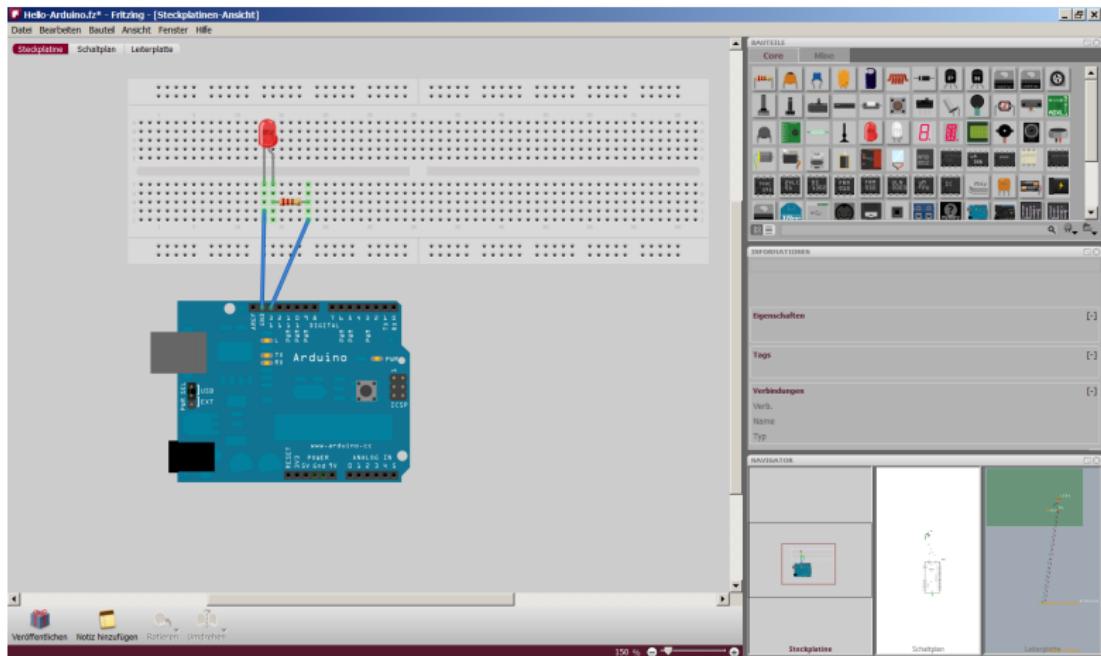


Abbildung : Screenshot Fritzing

Mein erstes Programm

Aufgabe: LED blinken lassen

- ▶ LEDs haben Kathode (–) und Anode (+, längeres Beinchen)
- ▶ Merkregel: „Kürzeres Bein = Kathode“
- ▶ nicht direkt anschließen², Vorwiderstand nutzen
- ▶ Widerstandsrechner im Internet (<http://www.elektronik-kompendium.de/sites/bau/1109111.htm>)
- ▶ 20 mA bei 5 V Betriebsspannung = 220Ω

$$\frac{5 \text{ V} - 1.6 \text{ V}}{0.02 \text{ A}} = 170 \Omega$$

Nächster verfügbarer Wert $\Rightarrow 220 \Omega$

²Erklärung dazu...

Verkabelung

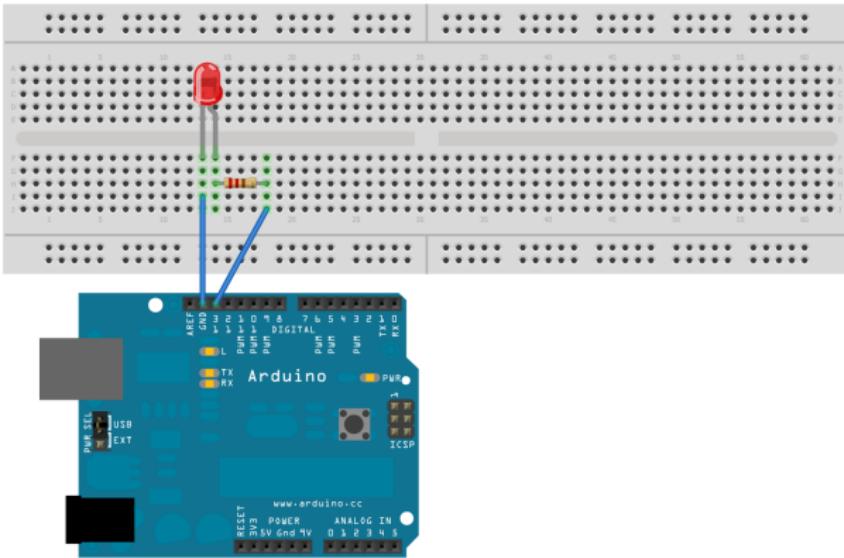


Abbildung : Verkabelungsschema, exportiert aus Fritzing

Fritzing

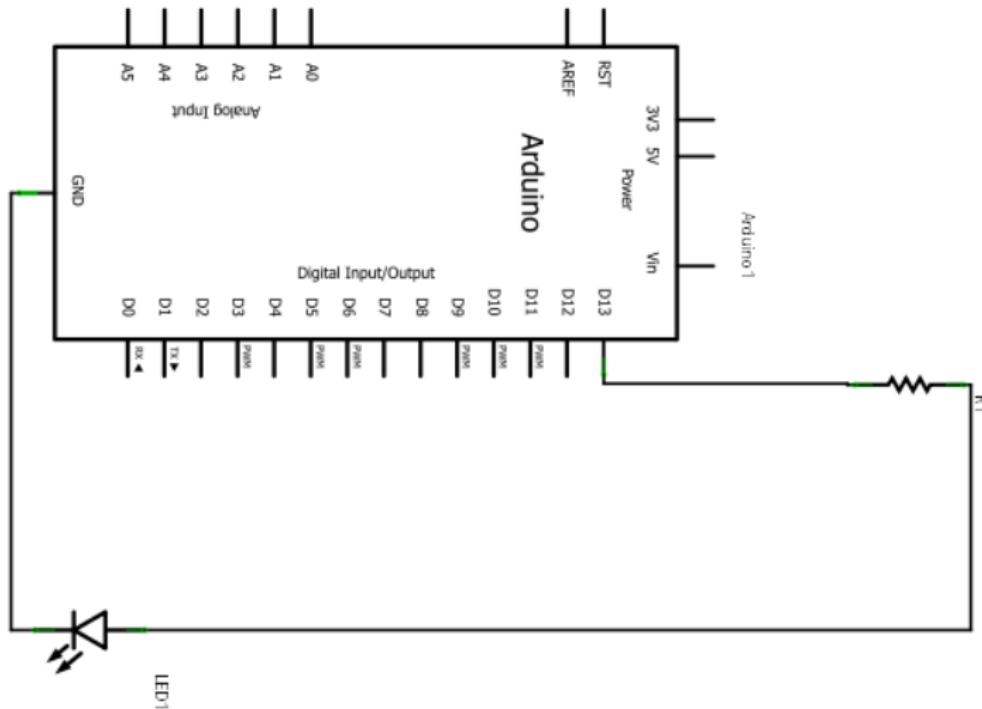


Abbildung : Schaltplan, exportiert aus Fritzing

Programmcode

Siehe Examples ⇒ Basic ⇒ Blink.pde

```
/*
Blink
Turns on an LED on for one second, then off for one second,
repeatedly.
This example code is in the public domain.
*/
void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);      // set the LED on
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // set the LED off
    delay(1000);                // wait for a second
}
```

Mehr LEDs ansteuern mit Schieberegistern - Grundlagen

- ▶ ich nutze einen 74HC164 (28 Cent bei Reichelt Elektronik aka „E-Aldi“)
- ▶ logisches Schaltwerk bla, bla ⇒ Wikipedia
- ▶ Funktion:
 - ▶ nimm seriell eine 8-stellige Binärzahl seriell entgegen
 - ▶ schau an jeder Stelle, ob Ziffer 0 oder 1
 - ▶ schalte den entsprechenden Ausgang oder nicht...
 - ▶ Strom kann ein wenig zu viel sein für den Arduino (nur 16–20mA)
⇒ FETs empfehlenswert

Mehr LEDs ansteuern mit Schieberegistern - Ansteuerung

Pin Arrangement

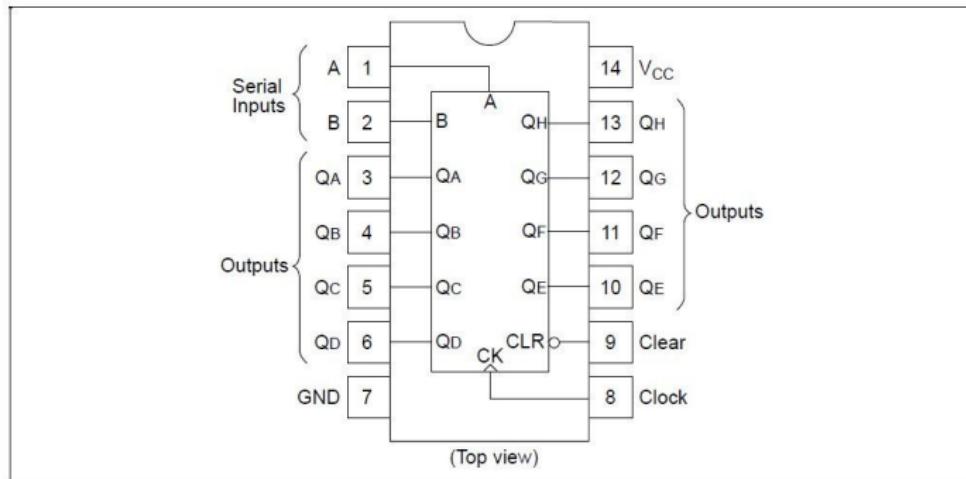


Abbildung : Schieberegister, Quelle: Instructables³

³<http://www.instructables.com/id/The-74HC164-Shift-Register-and-your-Arduino/step2/Basic-wiring-and-operation/>

Mehr LEDs ansteuern mit Schieberegistern - Versuchsaufbau

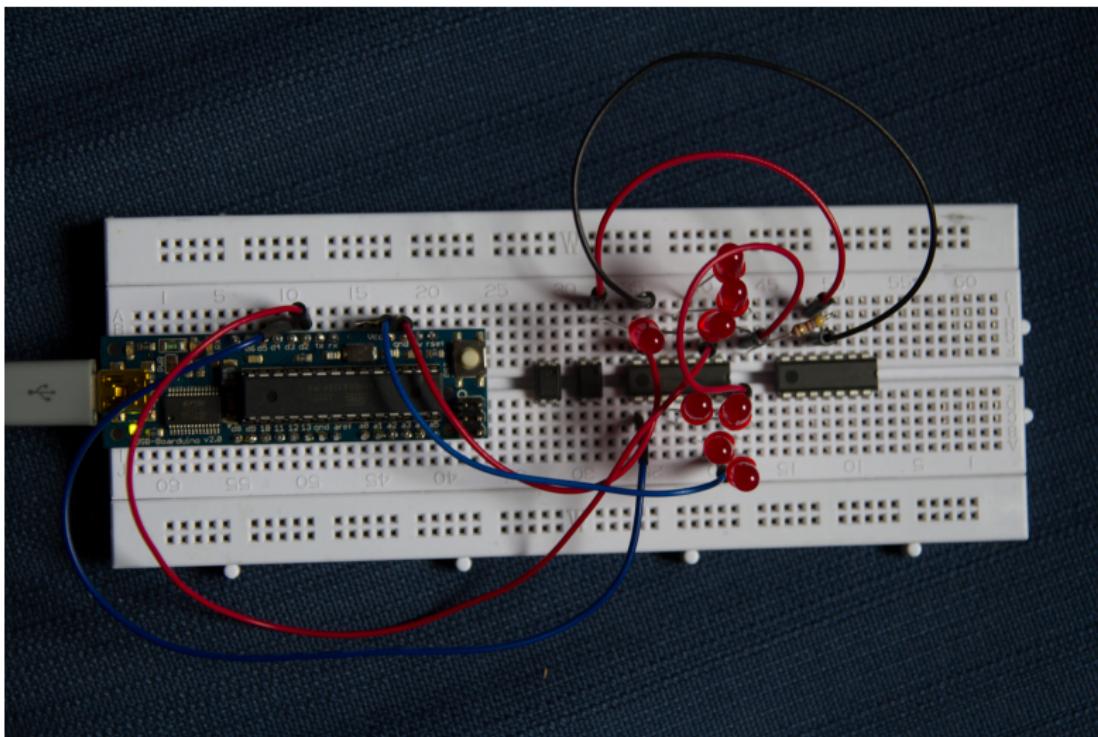


Abbildung : Schieberegister Versuchsaufbau

Schieberegister Programmcode

```
#define data 2
#define clock 3
#define wait 200

byte zero    = B00000001;byte one     = B00000011;
byte two     = B00000111;byte three   = B00001111;
byte four    = B00011111;byte five    = B00111111;
byte six     = B01111111;byte seven   = B11111111;

void setup(){
    pinMode(clock, OUTPUT); // make the clock pin an output
    pinMode(data , OUTPUT); // make the data pin an output3
}

void loop(){
    shiftOut(data, clock, LSBFIRST, zero); delay(wait);
    shiftOut(data, clock, LSBFIRST, one);   delay(wait);
    shiftOut(data, clock, LSBFIRST, two);   delay(wait);
    shiftOut(data, clock, LSBFIRST, three); delay(wait);
    shiftOut(data, clock, LSBFIRST, four);  delay(wait);
    shiftOut(data, clock, LSBFIRST, five);  delay(wait);
    shiftOut(data, clock, LSBFIRST, six);   delay(wait);
    shiftOut(data, clock, LSBFIRST, seven); delay(wait);
}
```

Das LoL-Shield

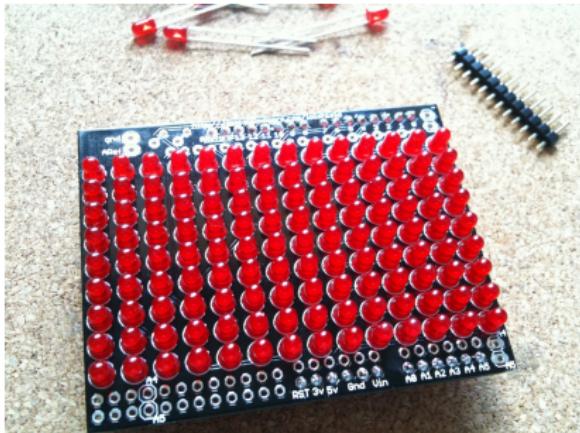


Abbildung : LoLshield, Quelle www.makethisdothat.com

- ▶ 9×14 LEDs im Raster, nutzt „Charlieplexing“
- ▶ Charlieplexing: n Pins steuern $n^2 - n$ LEDs
- ▶ komplette Bibliothek zur einfachen Programmierung
- ▶ Webseite <http://jimmieprodgers.com/kits/lolshield/>

Das LoL-Shield – Hilfstabelle

The screenshot shows an Excel spreadsheet with the following details:

- Worksheet Title:** MyLoLatx - Microsoft Excel
- Cell A1:** Contains the formula `=P$5&Q$5&R$5&Q6&R6&Q7&R7&Q8&R8&Q9&R9&Q10&R10&Q11&R11&Q12&R12&Q13&R13`.
- Cell A2:** Contains the value 0.
- Cell A3:** Contains the value 1.
- Cell A4:** Contains the value 2.
- Cell A5:** Contains the value 3.
- Cell A6:** Contains the value 4.
- Cell A7:** Contains the value 5.
- Cell A8:** Contains the value 6.
- Cell A9:** Contains the value 7.
- Cell A10:** Contains the value 8.
- Cell A11:** Contains the value 9.
- Cell A12:** Contains the value 10.
- Cell A13:** Contains the value 11.
- Cell A14:** Contains the value 12.
- Cell A15:** Contains the value 13.
- Cell B5:** Contains the value 1.
- Cell C5:** Contains the value 0.
- Cell D5:** Contains the value 0.
- Cell E5:** Contains the value 0.
- Cell F5:** Contains the value 0.
- Cell G5:** Contains the value 0.
- Cell H5:** Contains the value 0.
- Cell I5:** Contains the value 0.
- Cell J5:** Contains the value 0.
- Cell K5:** Contains the value 0.
- Cell L5:** Contains the value 0.
- Cell M5:** Contains the value 0.
- Cell N5:** Contains the value 0.
- Cell O5:** Contains the value 0.
- Cell P5:** Contains the value 0.
- Cell Q5:** Contains the value 0.
- Cell R5:** Contains the value 0.
- Cell S5:** Contains the value 0.
- Cell B6:** Contains the value 0.
- Cell C6:** Contains the value 1.
- Cell D6:** Contains the value 1.
- Cell E6:** Contains the value 0.
- Cell F6:** Contains the value 0.
- Cell G6:** Contains the value 0.
- Cell H6:** Contains the value 0.
- Cell I6:** Contains the value 0.
- Cell J6:** Contains the value 0.
- Cell K6:** Contains the value 0.
- Cell L6:** Contains the value 0.
- Cell M6:** Contains the value 0.
- Cell N6:** Contains the value 0.
- Cell O6:** Contains the value 0.
- Cell P6:** Contains the value 0.
- Cell Q6:** Contains the value 0.
- Cell R6:** Contains the value 0.
- Cell S6:** Contains the value 0.
- Cell B7:** Contains the value 2.
- Cell C7:** Contains the value 0.
- Cell D7:** Contains the value 1.
- Cell E7:** Contains the value 0.
- Cell F7:** Contains the value 0.
- Cell G7:** Contains the value 0.
- Cell H7:** Contains the value 0.
- Cell I7:** Contains the value 0.
- Cell J7:** Contains the value 0.
- Cell K7:** Contains the value 0.
- Cell L7:** Contains the value 0.
- Cell M7:** Contains the value 0.
- Cell N7:** Contains the value 0.
- Cell O7:** Contains the value 0.
- Cell P7:** Contains the value 0.
- Cell Q7:** Contains the value 0.
- Cell R7:** Contains the value 0.
- Cell S7:** Contains the value 0.
- Cell B8:** Contains the value 3.
- Cell C8:** Contains the value 0.
- Cell D8:** Contains the value 0.
- Cell E8:** Contains the value 1.
- Cell F8:** Contains the value 0.
- Cell G8:** Contains the value 0.
- Cell H8:** Contains the value 0.
- Cell I8:** Contains the value 0.
- Cell J8:** Contains the value 0.
- Cell K8:** Contains the value 0.
- Cell L8:** Contains the value 0.
- Cell M8:** Contains the value 0.
- Cell N8:** Contains the value 0.
- Cell O8:** Contains the value 0.
- Cell P8:** Contains the value 0.
- Cell Q8:** Contains the value 0.
- Cell R8:** Contains the value 0.
- Cell S8:** Contains the value 0.
- Cell B9:** Contains the value 4.
- Cell C9:** Contains the value 1.
- Cell D9:** Contains the value 0.
- Cell E9:** Contains the value 1.
- Cell F9:** Contains the value 0.
- Cell G9:** Contains the value 1.
- Cell H9:** Contains the value 0.
- Cell I9:** Contains the value 1.
- Cell J9:** Contains the value 0.
- Cell K9:** Contains the value 1.
- Cell L9:** Contains the value 0.
- Cell M9:** Contains the value 1.
- Cell N9:** Contains the value 0.
- Cell O9:** Contains the value 0.
- Cell P9:** Contains the value 0.
- Cell Q9:** Contains the value 0.
- Cell R9:** Contains the value 0.
- Cell S9:** Contains the value 0.
- Cell B10:** Contains the value 5.
- Cell C10:** Contains the value 0.
- Cell D10:** Contains the value 0.
- Cell E10:** Contains the value 0.
- Cell F10:** Contains the value 0.
- Cell G10:** Contains the value 0.
- Cell H10:** Contains the value 0.
- Cell I10:** Contains the value 0.
- Cell J10:** Contains the value 0.
- Cell K10:** Contains the value 0.
- Cell L10:** Contains the value 0.
- Cell M10:** Contains the value 0.
- Cell N10:** Contains the value 0.
- Cell O10:** Contains the value 0.
- Cell P10:** Contains the value 0.
- Cell Q10:** Contains the value 0.
- Cell R10:** Contains the value 0.
- Cell S10:** Contains the value 0.
- Cell B11:** Contains the value 6.
- Cell C11:** Contains the value 1.
- Cell D11:** Contains the value 1.
- Cell E11:** Contains the value 0.
- Cell F11:** Contains the value 0.
- Cell G11:** Contains the value 0.
- Cell H11:** Contains the value 0.
- Cell I11:** Contains the value 0.
- Cell J11:** Contains the value 0.
- Cell K11:** Contains the value 0.
- Cell L11:** Contains the value 0.
- Cell M11:** Contains the value 0.
- Cell N11:** Contains the value 0.
- Cell O11:** Contains the value 0.
- Cell P11:** Contains the value 0.
- Cell Q11:** Contains the value 0.
- Cell R11:** Contains the value 0.
- Cell S11:** Contains the value 0.
- Cell B12:** Contains the value 7.
- Cell C12:** Contains the value 0.
- Cell D12:** Contains the value 0.
- Cell E12:** Contains the value 0.
- Cell F12:** Contains the value 0.
- Cell G12:** Contains the value 0.
- Cell H12:** Contains the value 0.
- Cell I12:** Contains the value 0.
- Cell J12:** Contains the value 0.
- Cell K12:** Contains the value 0.
- Cell L12:** Contains the value 0.
- Cell M12:** Contains the value 0.
- Cell N12:** Contains the value 0.
- Cell O12:** Contains the value 0.
- Cell P12:** Contains the value 0.
- Cell Q12:** Contains the value 0.
- Cell R12:** Contains the value 0.
- Cell S12:** Contains the value 0.
- Cell B13:** Contains the value 8.
- Cell C13:** Contains the value 0.
- Cell D13:** Contains the value 0.
- Cell E13:** Contains the value 0.
- Cell F13:** Contains the value 0.
- Cell G13:** Contains the value 0.
- Cell H13:** Contains the value 0.
- Cell I13:** Contains the value 0.
- Cell J13:** Contains the value 0.
- Cell K13:** Contains the value 0.
- Cell L13:** Contains the value 0.
- Cell M13:** Contains the value 0.
- Cell N13:** Contains the value 0.
- Cell O13:** Contains the value 0.
- Cell P13:** Contains the value 0.
- Cell Q13:** Contains the value 0.
- Cell R13:** Contains the value 0.
- Cell S13:** Contains the value 1.

The formula bar shows the formula `=P$5&Q$5&R$5&Q6&R6&Q7&R7&Q8&R8&Q9&R9&Q10&R10&Q11&R11&Q12&R12&Q13&R13`. The status bar indicates the formula is copied to cell T13.

The right side of the screen shows a snippet of code or a list of values:

```
{ 1,  
 6,  
 4,  
 8,  
 {1,6,4,8,5461,0,7,0,8192}  
 5461,  
 0,  
 7,  
 0,  
 8192 }
```

Abbildung : LoLshield – Excel Mappe zur Erzeugung der Frames

Das LoL-Shield – Let's fire it up! I

```
/*
FrOSCon 2011 Animation
```

Written for the LoL Shield, designed by Jimmie Rodgers:
<http://jimmieprodgers.com/kits/lolshield/>

This needs the Charliplexing library, which you can get at the
LoL Shield project page: <http://code.google.com/p/lolshield/>

Created by Jimmie Rodgers on 12/30/2009.

Adapted from: <http://www.arduino.cc/playground/Code/BitMath>

History:

August 20, 2011 - Animation created for FrOSCon 2011

December 30, 2009 - V1.0 first version written at 26C3/Berlin

This is free software; you can redistribute it and/or
modify it under the terms of the GNU Version 3 General Public
License as published by the Free Software Foundation;
or (at your option) any later version.

Das LoL-Shield – Let's fire it up! II

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

*/

```
#include <avr/pgmspace.h> //AVR library for writing to ROM
#include <Charlplexing.h> //Imports the library, which needs to be
                           //Initialized in setup.
int blinkdelay = 1000;      //Sets the time each frame is shown

uint16_t BitMap[][][9] PROGMEM = {
{1,6,4,8,5461,0,7,0,8192},
{1,0,0,0,0,0,0,0,8192},
{15017,2729,2729,2729,15017,2729,2729,2729,15343},
{2345,2429,2429,2429,2429,2429,2361,2321,1553},
{448,992,2032,2032,4088,2032,2032,992,448},
```

Das LoL-Shield – Let's fire it up! III

```
{18000}
};

void setup() {
    LedSign::Init(); //Initializes the screen
}
void loop() {
    DisplayBitMap(); //Displays the bitmap
}

void DisplayBitMap()
{
    boolean run=true; //While this is true, the screen updates
    byte frame = 0; //Frame counter
    byte line = 0; //Row counter
    unsigned long data; //Temporary storage of the row data

    while(run == true) {
        for(line = 0; line < 9; line++) {

            //Here we fetch data from program memory with a pointer.
            data = pgm_read_word_near (&BitMap[frame][line]);
        }
    }
}
```

Das LoL-Shield – Let's fire it up! IV

```
//Kills the loop if the kill number is found
if (data==18000) {
    run=false;
}
else for (byte led=0; led<14; ++led) {
    if (data & (1<<led)) {
        LedSign::Set(led, line, 1);
    }
    else {
        LedSign::Set(led, line, 0);
    }} delay(blinkdelay);
frame++; }}
```

LC Displays

per LC-Display (Liquid Cristal Display) lassen sich einfach und schnell Informationen ausgeben

- ▶ ideal: Displays kompatibel mit HD44780 wählen,
„Industrie-Standard“
- ▶ DOGM Displays für knapp 11 Euro
- ▶ Bibliothek: <http://code.google.com/p/doglcd/>

⇒ <http://www.reichelt.de/?ACTION=3;ARTICLE=77826;PROVID=2402>

Schaltplan

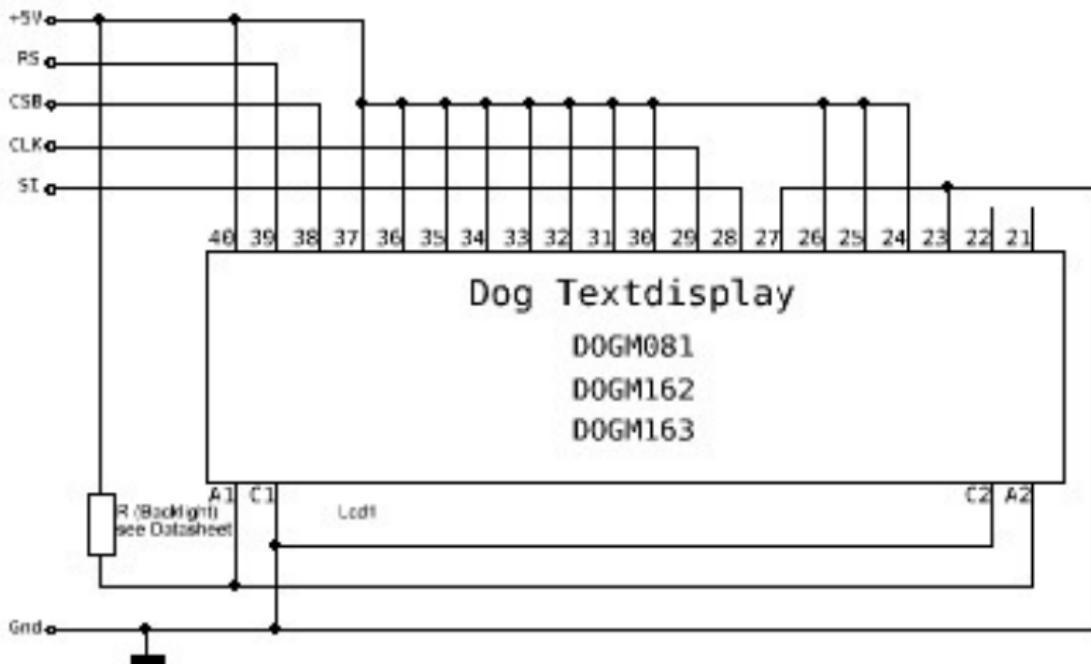


Abbildung : Schaltplan

DOGM Programmcode

```
/* basiert auf DogLcd Library - Hello World
We assume the following pins are connected:
* LCD SI pin to digital pin 2
* LCD CLK pin to digital pin 3
* LCD RS pin to digital pin 4
* LCD CSB pin to digital pin 5
http://www.arduino.cc/en/Tutorial/LiquidCrystal
*/
#include <DogLcd.h>
DogLcd lcd(2, 3, 4, 5);

void setup() {
    lcd.begin(DOG_LCD_M081, 0x20);
    lcd.print("Hallo!");
    delay(500);
    lcd.clear(); lcd.noCursor();
}

void loop() {
    lcd.setCursor(0, 0);
    lcd.print(millis()/1000);
}
```

Schaltplan

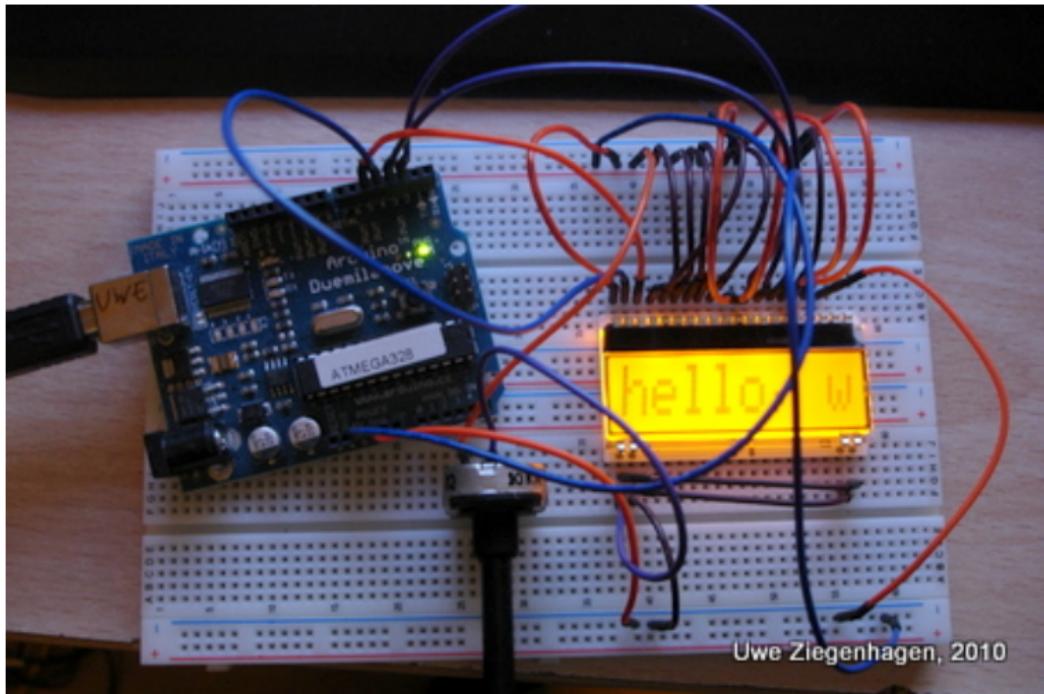


Abbildung : Beschaltung DOGM Display

I²C-Displays

- ▶ I²C: „Inter-Integrated Circuit“, von Philips/NXP entwickelt
- ▶ alternativer Name: TWI, „Two-Wire-Interface“
- ▶ Besonderheit, mehrere Geräte (auch identische) können am Bus hängen
- ▶ Ich nutze das **Monochrome 128x32 I2C OLED graphic display** von Adafruit, ca. 18 Euro bei Watterott
- ▶ 128x32 Pixel auf 0.91 Inch (2.31 cm)
- ▶ Tutorial dazu unter <http://learn.adafruit.com/monochrome-oled-breakouts/wiring-128x32-i2c-display>

Monochrome 128x32 I2C OLED

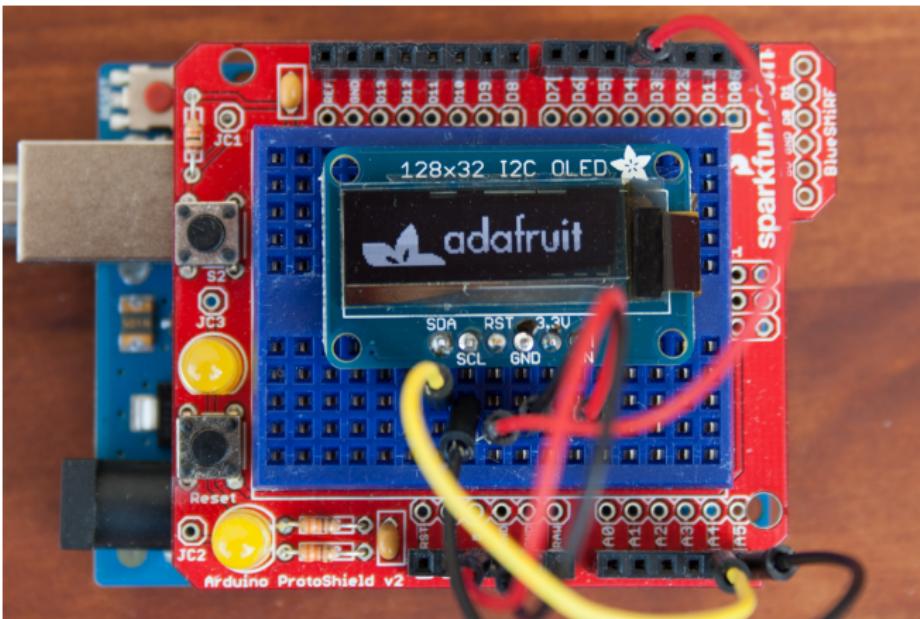
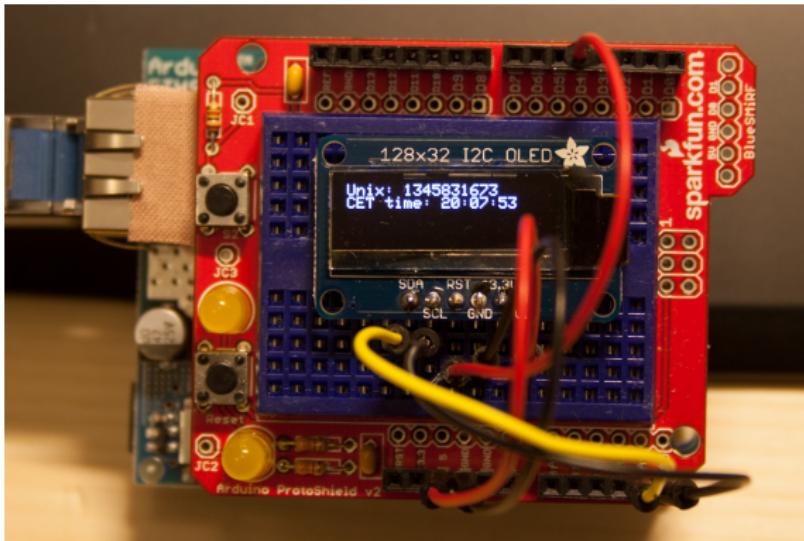


Abbildung : Monochrome 128x32 I2C OLED

NTP-Server abfragen

- ▶ NTP = Network Time Protocol
- ▶ fertiger Client für den Arduino und das Ethernet-Shield im [Arduino Playground](#)
- ▶ unsere Aufgabe: den NTP & Display-Code miteinander verheiraten



Serielle Displays

- ▶ Serielles Display, braucht nur eine Leitung (plus 5 V und GND)
- ▶ Ich nutze <http://www.watterott.com/de/Serial-Enabled-16x2-LCD-Yellow-on-Blue-5V>, Kosten: 20 Euro
- ▶ 2 Zeilen mit jeweils 16 Zeichen
- ▶ hat intern einen PIC16F88 Mikrocontroller, der die Kommunikation abwickelt
- ▶ ⇒ spart wertvolle Pins! (Pins sind meistens Mangelware!)

Serielle Displays



Abbildung : Sparkfun-Display, Quelle: www.watterott.com

Serielle Displays – Beispielanwendung

- ▶ NTP Zeit alle 15 Sekunden nicht für jede Anwendung wichtig
- ▶ RTC („Real Time Clock“) Modul von Sparkfun, ca. 17 Euro
- ▶ enthält einen DS1307 RTC Controller und Lithiumzelle
- ▶ Ansteuerung per I²C
- ▶ Beispiel-Skript unter http://combustory.com/wiki/index.php/RTC1307_-_Real_Time_Clock
- ▶ Display + RTC = Uhr

Serielle Displays

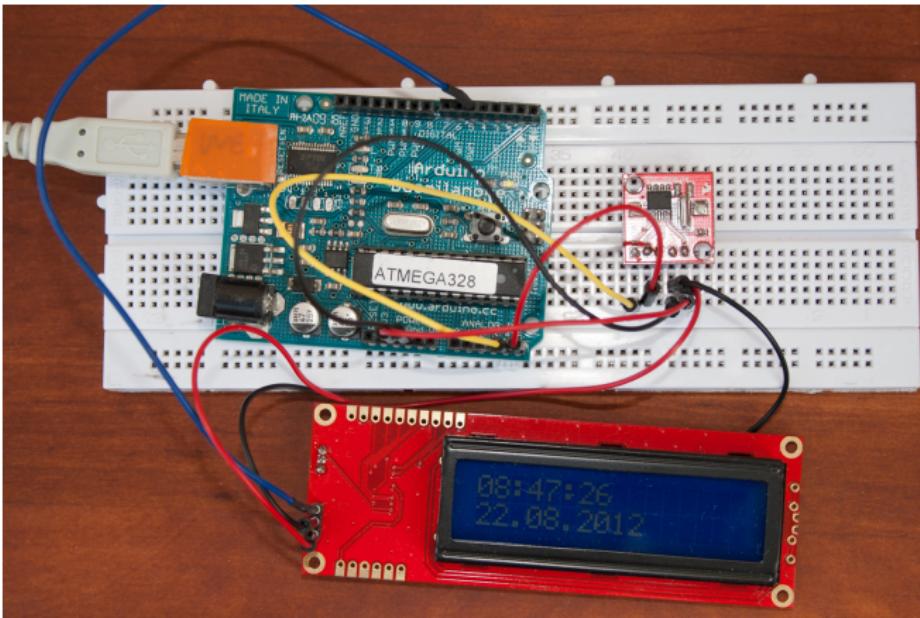


Abbildung : Sparkfun-Display mit Sparkfun RTC

„Ein 14-jähriger Schüler zeigt der Chilenischen Regierung, wie man mit einfachsten Mitteln ein funktionierendes Erdbeben-Warnsystem errichten kann. Sebastian Alegria beschaffte sich einen Erdbeben-Detektor für Haushalte, tauschte dessen integrierte Elektronik mit einem Arduino-Mikrocontroller aus und verband das Ganz mit seinem Webserver. Sekunden, bevor ein Erdbeben droht, wird eine Warnung per Twitter verschickt.“

Quelle: <http://kurier.at/techno/3924051.php>

„Die AEG-Waschmaschine meldet über eine LED, ob die Wäsche fertig gewaschen wurde. Wenn die Maschine jedoch im Keller steht, ist das schwer zu überwachen. Ein Bastler hat nun dieses Signal abgegriffen und kann in seinem LAN nachsehen, ob die Waschmaschine ihre Arbeit beendet hat.“

Quelle: <http://www.golem.de/1108/85541.html>

Bluetooth Morse Code Keyboard

„Das „Bluetooth Morse Code Keyboard“ von Zunkworks ist mit der freien Prototyping-Plattform Arduino entwickelt worden. Es übermittelt per Bluetooth die Tastenkommandos an den Rechner, an den es gekoppelt ist. Vorher werden die eingegebenen Morsesignale noch in normale Tastaturbefehle zurückübersetzt, damit jeder Rechner, der ein Bluetooth-HID-Interface unterstützt, damit funktioniert.“

Quelle: <http://www.golem.de/1108/85465.html>

„Die PLOTS-Forscher nutzen Teile, die regulär im Handel erhältlich sind, wie die Teammitglieder Jae-Ok Lee und Byeongwon Ha erklären. Der Chemikaliensensor MQ 135 und eine farbige LED werden von einem Arduino-Microcontroller-Board gesteuert. All das hängt wiederum samt Akkusatz an einem iRobot Roomba Staubsauger, den die Wissenschaftler gebraucht für 50 Dollar erwarben.“

Quelle: <http://www.heise.de/newstickermeldung/Roomba-als-Raumluftsensor-1282241.html>

WASP: Drohne zum Entern von WiFi und GSM-Netzen

„Die Flugzeugelektronik basiert auf dem Ardupilot, einem auf einem Arduino Mega basierendem Autopilotsystem. Eine HD-Kamera, die an der Drohne befestigt ist, kann zudem Bilder vom überflogenen Gebiet aufnehmen.“

Quelle: <http://www.pro-linux.de/news/1/17328/wasp-drohne-zum-entern-von-wifi-und-gsm-netzen.html>

Wie fängt man an?

- ▶ Arduino
- ▶ USB-Kabel
- ▶ Breadboard
- ▶ Kabelset für das Breadboard
- ▶ Widerstände
- ▶ LEDs
- ▶ Multimeter
- ▶ Lötstation
- ▶ Literatur
- ▶ und ganz wichtig: Zeit!

Bezugsquellen

- ▶ <http://www.arduino.cc>
- ▶ <http://www.watterott.com>
- ▶ <http://www.elmicro.com>
- ▶ <http://www.segor.de>
- ▶ <http://www.sparkfun.com>
- ▶ <http://www.reichelt.de>
- ▶ <http://www.seeedstudio.com/depot/>

- ▶ *Arduino: Praxiseinstieg* von Thomas Brühlmann
- ▶ *Praxisbuch Arduino - Mikrocontroller-Programmierung mit Arduino und Freeduino* von Ulli Sommer
- ▶ *Arduino - Physical Computing für Bastler, Designer und Geeks* von Odendahl, Finn, Wenger
- ▶ *Getting Started with Arduino* von Massimo Banzi
- ▶ *Practical Arduino: Cool Projects for Open Source Hardware* von Oxer und Blemings

- ▶ Hackerspaces, Fablabs, etc.
- ▶ Beispiel in Köln: Dingfabrik (dingfabrik.de)
 - ▶ offene Werkstatt
 - ▶ komplett ausgerüstete Holz/Metall-Werkstatt
 - ▶ 3D-Drucker, Lasercutter
 - ▶ umfangreiche Elektronik-und -Werkzeuge
 - ▶ Freitagabends ab 19 Uhr: freies Basteln



Gibt es Alternativen zum Arduino?

- ▶ Microsoft Gadgeteer, programmierbar in **.NET**
- ▶ Netduino, baut ebenfalls auf **.NET** auf
- ▶ BeagleBone Black, 1GHz Arm Cortex-A8, HDMI, Ethernet, 2x6 Pins
- ▶ Raspberry Pi
 - ▶ Mehr Power: 700 MHz ARM 11 auf dem Level eines Pentium II
 - ▶ Linux onboard: Raspian, Debian, XBMC
 - ▶ Netzwerk onboard: Ethernet RJ45
 - ▶ Video onboard: HDMI Ausgang mit 1080p Support
 - ▶ Python-Support: Aus Python heraus Pins ansteuern
 - ▶ Coolnessfaktor⁴ ziemlich hoch...
 - ▶ mit Versand knapp 45 Euro

⁴Im Nerd-Umfeld...