

Arduino

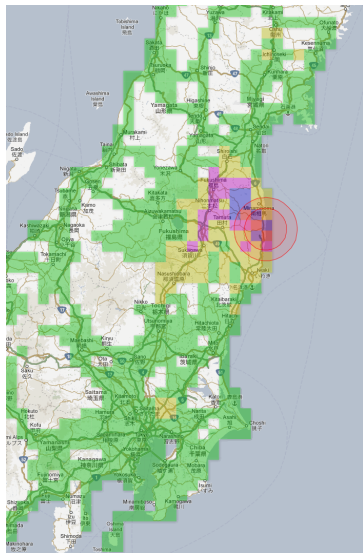
Einführung in die Mikrocontroller-Programmierung

Uwe Ziegenhagen

26. August 2012

Fukushima

- ▶ 11.03.2011: Erdbeben, Tsunami in Fukushima, drei Kernschmelzen
- ▶ öffentliche Daten „ungenau“, Prozess sehr mühselig
- ▶ Lösung: Arduino liest Werte vom Geigerzähler, liest GPS Koordinaten
- ▶ beides wird alle 5 Sekunden auf eine SD Karte geschrieben.
- ▶ Ergebnis: unabhängige (!) Strahlungsdaten in hoher Dichte, öffentlich zugänglich
- ▶ [PBS Video bei Youtube](#)



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
- ▶ \Rightarrow „Elektronik funktioniert mit Rauch!“
- ▶ Blog unter www.uweziegenhagen.de, dort liegen auch die (in \LaTeX erstellten) Folien zu diesem Vortrag
- ▶ Was ist \LaTeX ? 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 IDII (Interaction Design Institute Ivrea)
- ▶ Projekt, um Studenten einfaches Prototyping mit Mikrocontrollern zu ermöglichen
- ▶ Arduino = italienischer König, Name einer Bar
- ▶ mehr dazu in <http://vimeo.com/18539129>

Es gibt verschiedene Arten von Arduinos:

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

Wir konzentrieren uns auf den Arduino Uno!

Arduino Uno

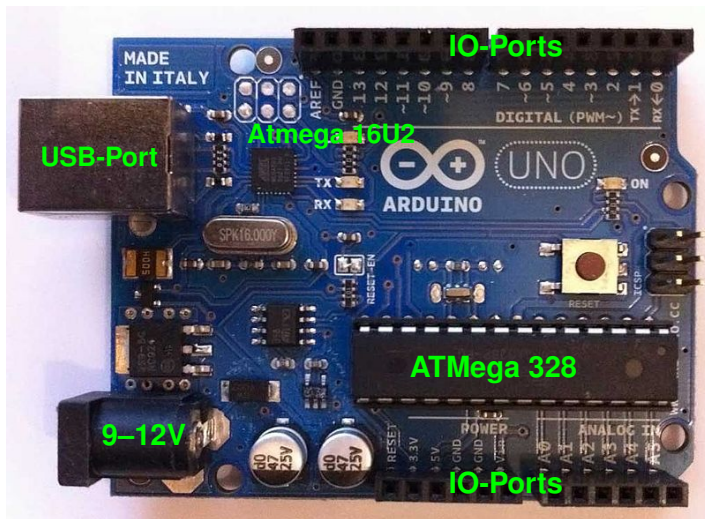


Abbildung : Arduino Uno Quelle: Wikipedia

Arduino Uno

- ▶ ATmega328 Mikrocontroller
- ▶ 32KB Flash (davon 0,5KB für Bootloader)
- ▶ 2KB 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 = Platinen, die direkt auf den Arduino gesteckt werden können
- ▶ erweitern Arduino um viele interessante Möglichkeiten
- ▶ Liste der Shield sehr lang: shieldlist.org
- ▶ für Einsteiger gut geeignet: Danger Shield, Proto Shield, Ethernet Shield
- ▶ Ich habe noch: WLAN Shield, GPRS Shield, LoL-Shield

Danger Shield

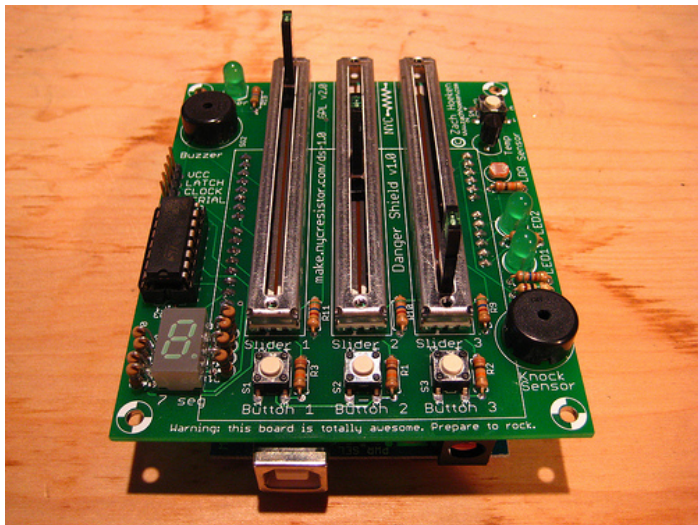


Abbildung : Das Danger Shield Quelle: www.zachhoeken.com

Proto Shield

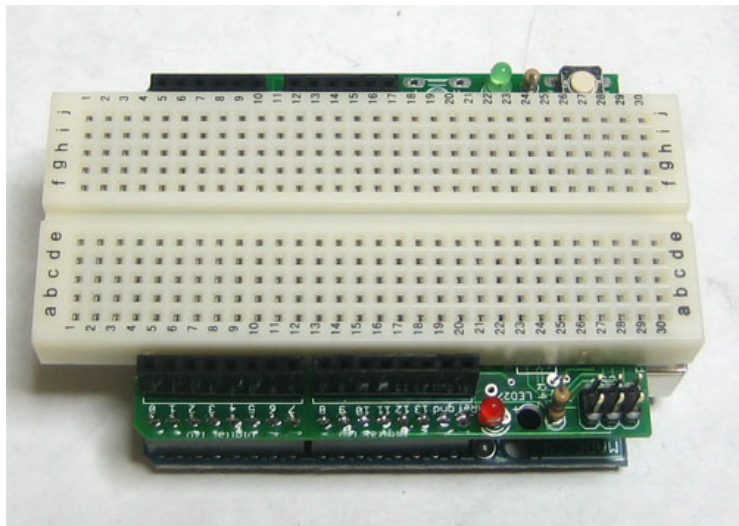


Abbildung : Das Proto Shield Quelle: www.ladyada.net

Ethernet Shield



Abbildung : Das Ethernet Shield mit MicroSD Quelle: www.watterott.com

- ▶ erhältlich für Windows, Linux, Mac
- ▶ aktuell momentan Version 1.0.1
- ▶ benötigt installierte JRE
- ▶ Mac und Windows Version bringen Compiler mit
- ▶ Linux benötigt GCC mit einigen AVR Bibliotheken

am besten auch gleich „Fritzing“ installieren

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

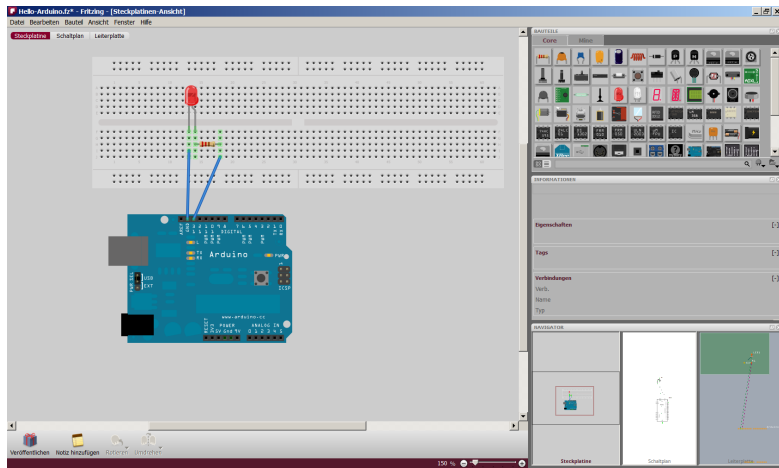


Abbildung : Screenshot Fritzing

Aufgabe: LED blinken lassen

- ▶ LEDs haben Kathode (-) und Anode (+, längeres Beinchen)
- ▶ Merkregel: „**K**ürzeres Bein = **K**athode“
- ▶ nicht direkt anschließen¹, Vorwiderstand nutzen
- ▶ Widerstandsrechner im Internet (<http://www.elektronik-kompodium.de/sites/bau/1109111.htm>)
- ▶ 20 mA bei 5 V Betriebsspannung = 220 Ω

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

Nächster Wert \Rightarrow 220 Ω

¹Erklärung warum...

Verkabelung

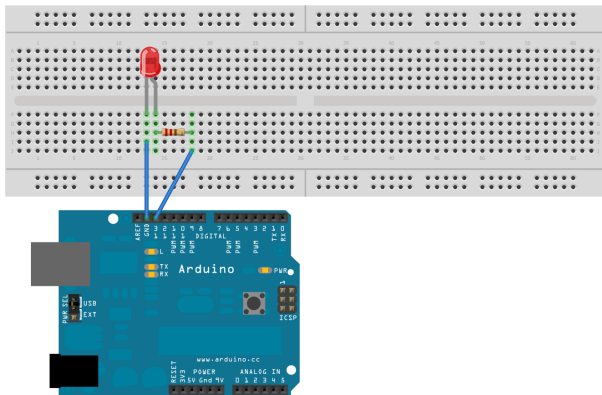


Abbildung : Verkabelungsschema, exportiert aus Fritzing

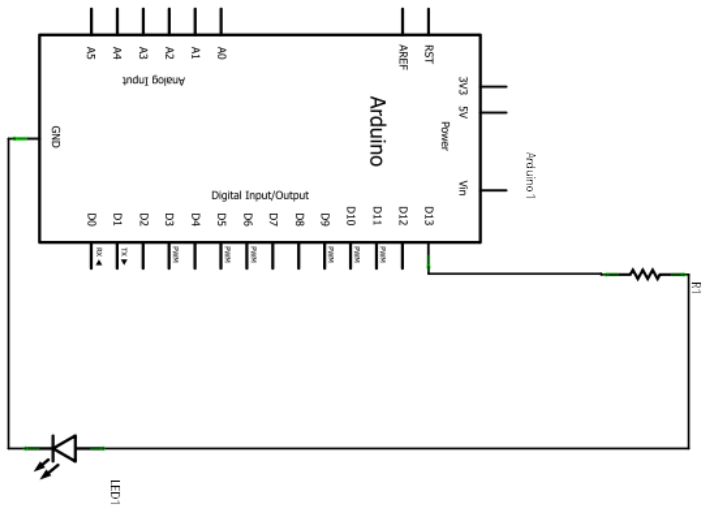


Abbildung : Schaltplan, exportiert aus Fritzing

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  
}
```

Das LoL-Shield

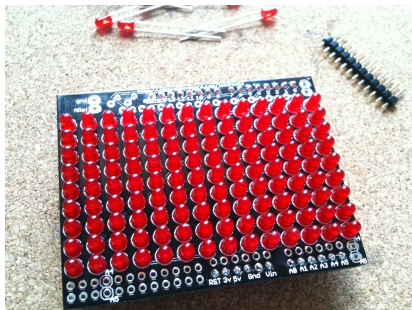


Abbildung : LoLshield, Quelle www.makethisdothat.com

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

Das LoL-Shield – Hilfstabelle

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1																				
2		0	1	2	3	4	5	6	7	8	9	10	11	12	13					
3		2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^10	2^11	2^12	2^13					
4		1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192					
5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0					
6	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0					
7	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0					
8	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0					
9	4	1	0	1	0	1	0	1	0	1	0	1	0	1	0					
10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
11	6	1	1	1	0	0	0	0	0	0	0	0	0	0	0					
12	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
13	8	0	0	0	0	0	0	0	0	0	0	0	0	0	1					
14																				

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://jimmieproducers.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 <Charliplexing.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++; }}
```


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

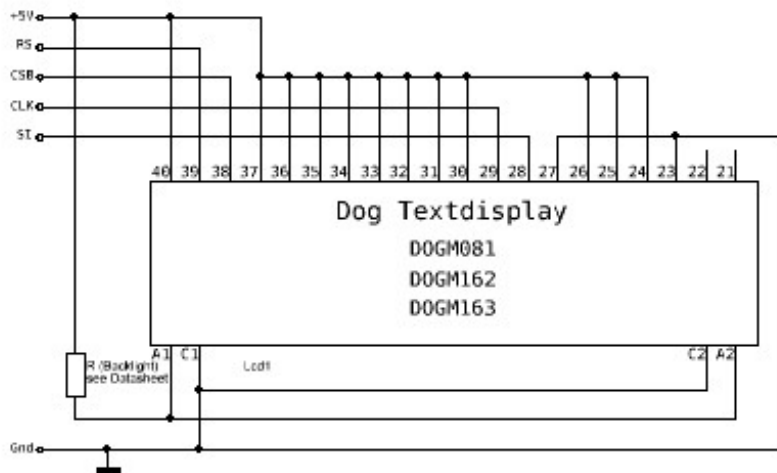


Abbildung : Schaltplan

DOG M 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);
}
```

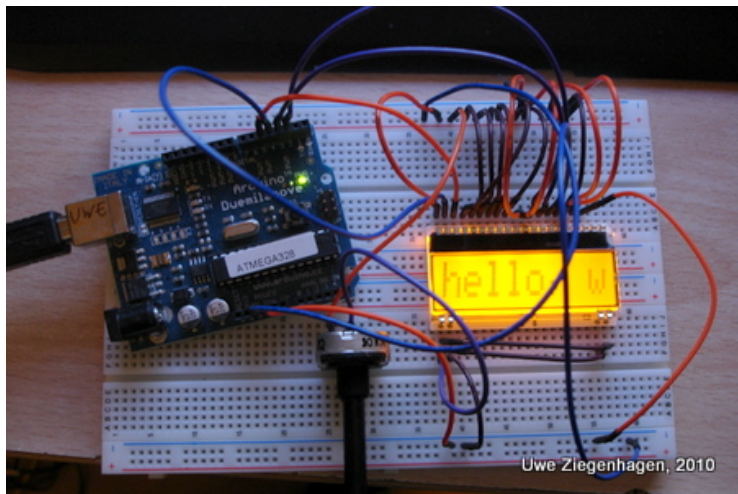


Abbildung : Beschaltung DOGM Display

- ▶ 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

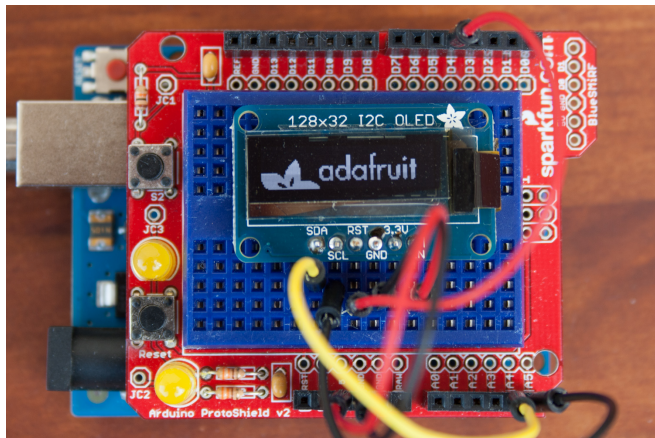
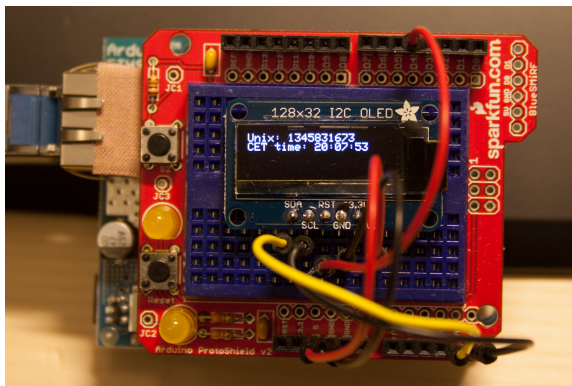


Abbildung : Monochrome 128x32 I2C OLED

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

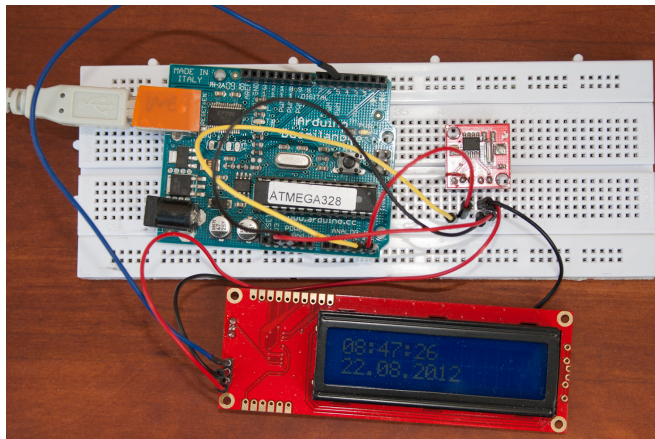


Abbildung : Sparkfun-Display mit Sparkfun RTC

„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 Ganze 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 erworben.“

Quelle: <http://www.heise.de/newsticker/meldung/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 Autopilotensystem. 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!

- ▶ <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 Ozer 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
- ▶ [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
 - ▶ Coolnessfaktor² ziemlich hoch...
 - ▶ mit Versand knapp 45 Euro
 - ▶ Nachteil: Beschaffung aktuell³ „schwierig“

²Im Nerd-Umfeld...

³Stand August 2012