



# Physical Computing

Ralf Nolte

Bielefeld, 13. Dezember 2011

# Überblick

- Motivation
- Definition
- Vorstellung der Hardware
- Vorstellung der Arduino IDE
- Hello World
- LC Display
- Gravity Plug an TWI Bus

# Motivation

- Einwirken auf die Umwelt
- Messen, steuern, regeln oft vor Ort nötig
- Keine Energieversorgung möglich
- Aufgaben oft einfach zu automatisieren, bei hohem Komfortgewinn
- Neue Gerätekonzepte möglich
- Erkenntnisgewinn

# Definition Microcontroller

- Prozessor
- Peripheriefunktionen
- Programmspeicher (optional)
- Arbeitsspeicher (optional)
- Sind alle 4 Punkte gegeben spricht man auch von eine „System on a Chip“ (SOC)
- Grenze zu Mikroprozessoren fließend

- Atmel AVR Microcontroller
  - Geringe Leistungsaufnahme
  - Viele verschiedene Modelle
  - Gute Dokumentation
  - Oft in DIL lieferbar
  - Umfangreiche Bibliotheken
  - Robust
  - Preiswert

- Pollin NetIO
  - LAN Onboard
  - Verlässliche Steckverbinder
  - Teilbestückung möglich
  - Sehr preiswert (benötigt Programmier: +10Eur)
- Freeduino
  - Simple IDE
  - Kein Programmier nötig
  - Klein, leicht
  - Via shields erweiterbar (LAN, WLAN)
  - Preiswert

- Beide Plattformen als Bausatz oder fertig aufgebaut lieferbar.
- Bauzeit:
  - Freeduino ca. 1h
  - NetIO ca. 2h
- Keine komplexen Lötaufgaben, SMDs kommen vormontiert, DILs sind gesockelt
- Spannungsversorgung:
  - Freeduino: 5V, 500mA via USB
  - NetIO: 9V, 200mA

- Frei verfügbar, nur für Windows
- mit jedem AVR-MC nutzbar
- C und Assembler, mit AVRlibc
- On- und Offline Debugger und Emulator
- Freier Zugriff auf alle Optionen des MC
- Alternativ als CLI nutzbar
- Programmierer notwendig



# Programmer

- USB AVR Lab von Christian Ullrich
- Preiswerte Alternative zu Original Programmern (15 Euro)
- Durch verschiedene Firmwares flexibel nutzbar
- Mailsupport direkt vom Entwickler
- Verkauf als „Bausatz“, es muss nur ein Quarz eingelötet werden

# Programmer (cont'd)

- Einbindung unter Win simpel, in VM nicht
- Die aktuellen Softwarekombination erlaubt kein sinnvolles Arbeiten mit USB AVR Lab unter Virtual Box
- Das Folgende daher nur als Beschreibung

# Pollin NetIO an Programmer

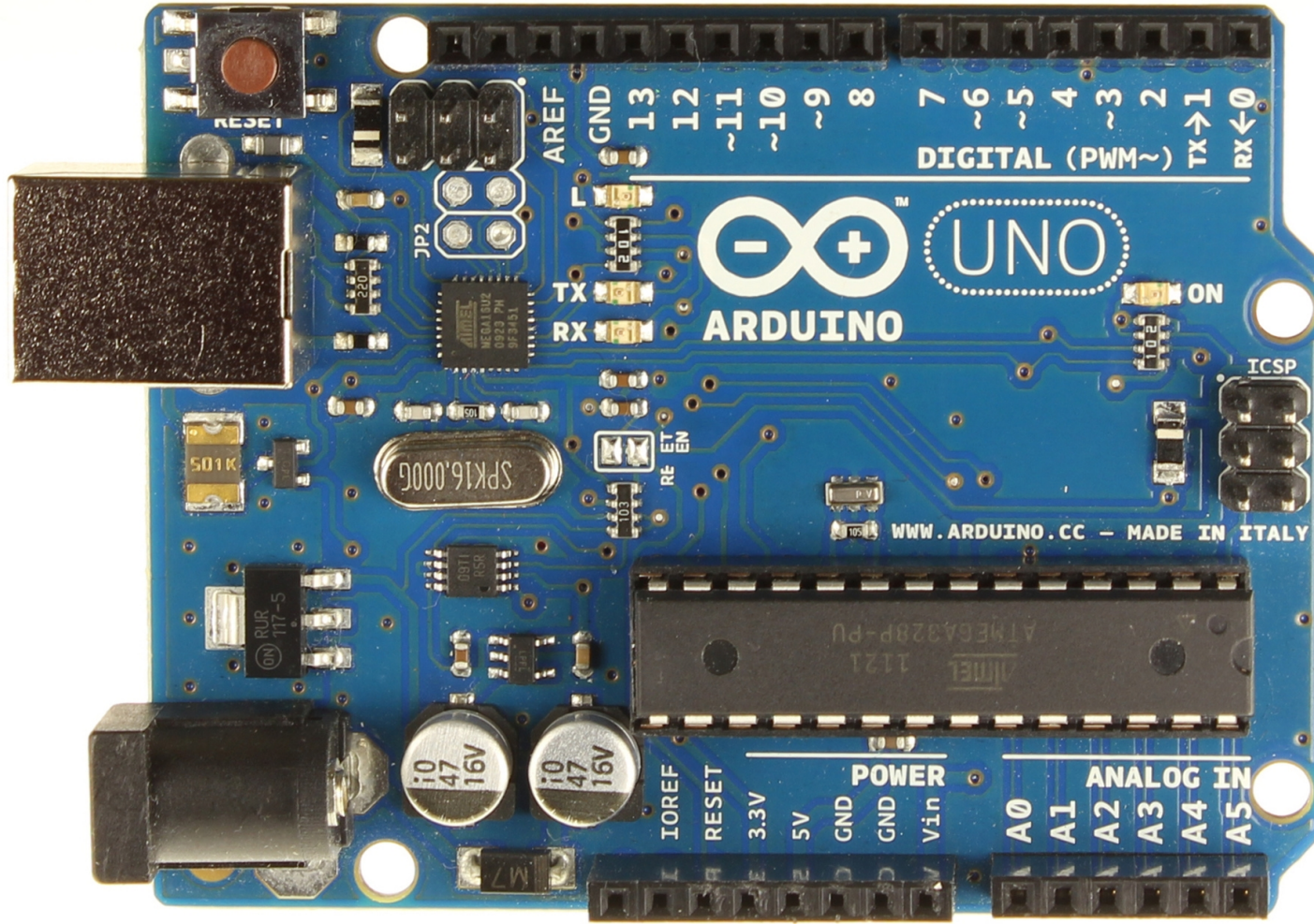
- Programmer und AVR werden mit 10 adrigem 1:1 Kabel verbunden
- NetIO mit 9V versorgen (Stabis werden warm!)
- Programmer an USB anschliessen
- Blaue LED heisst, daß die FW läuft
- Wenn der Programmer einen AVR erkannt hat, zeigt er das mit grüner LED an.

# Arduino IDE

- frei verfügbar für Win/MacOS/Linux
- basierend auf „Processing“
- umfangreiche Libraries integriert
- Fremdlibraries integrierbar
- an C angelehnte Sprache
- AVR-libc ansprechbar

- Hello World auf „Microcontroller“
- Serielle Ausgaben
- Serielle Eingaben
- Anbindung eines LCD
  - Dokumentation mit Beispielcode
  - Herstellerinfo
  - Hardware-Aufbau

# Arduino Hardware



## PINBELEGUNG

Pin	Symbol	Level	Funktion	Pin	Symbol	Level	Funktion
1	VSS	L	Stromversorgung 0V (GND)	10	D3	H / L	Display Data
2	VDD	H	Stromversorgung +5V	11	D4 (D0)	H / L	Display Data
3	VEE	-	Kontrastspannung, Eingang	12	D5 (D1)	H / L	Display Data
4	RS (CS)	H / L	Umschaltung Daten / Befehl	13	D6 (D2)	H / L	Display Data
5	R/W (SID)	H / L	H=Read, L=Write	14	D7 (D3)	H / L	Display Data, MSB
6	E (SCLK)	H	Enable (fallende Flanke)	15	-	-	frei (siehe EA DIP122-5N)
7	D0 (SOD)	H / L	Display Data, LSB	16	RES	L	Reset (interner Pullup 10k)
8	D1	H / L	Display Data	17	A	-	LED-Bel. + (RV erford.)
9	D2	H / L	Display Data	18	C	-	LED-Bel. -

## BELEUCHTUNG

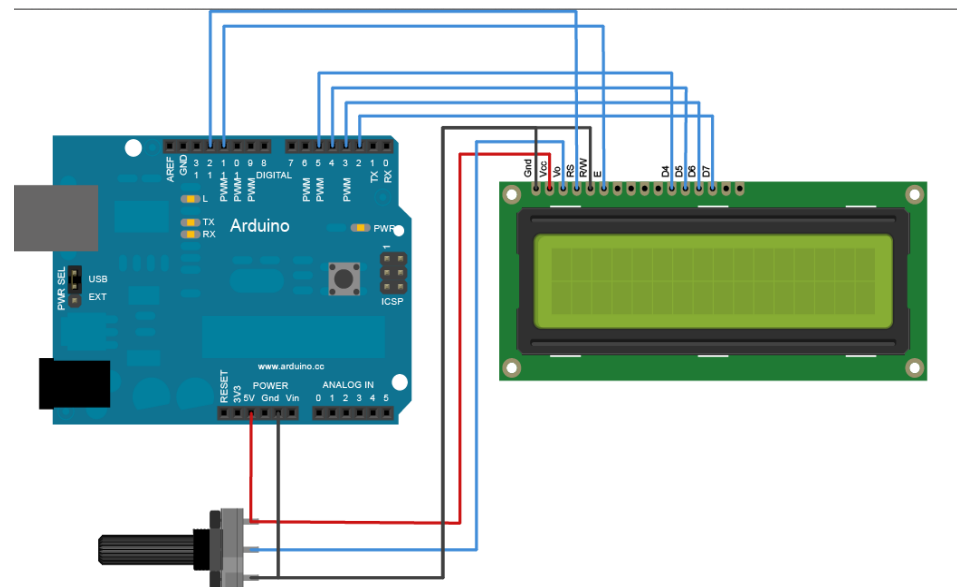
Der Betrieb der Hintergrundbeleuchtung erfordert eine Stromquelle oder einen externen Vorwiderstand zur Strombegrenzung. Die Flussspannung der gelb/grünen Beleuchtung liegt zwischen 3,9V und 4,2V und die der weißen Beleuchtung zwischen 3,0V und 3,6V. Bitte beachten Sie ein Derating für den Betrieb bei Temperaturen > +25°C!

**Achtung:** betreiben Sie die Beleuchtung nie direkt an 5V; das kann zur sofortigen Zerstörung führen!

# Dokumentation zur Library

<http://arduino.cc/en/Tutorial/LiquidCrystal>

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2





# Arduiono Hands-on (cont'd)

- Zur Kommunikation zwischen Bauteilen werden oft Bussysteme verwendet
- Sehr verbreitet hier SPI und I2C (auch TWI genannt)
- Beide Systeme werden von vielen Bauteilen unterstützt
- Als Beispiel wird hier die Einbindung des Gravity Plug von JeeLabs gezeigt

- Das Gravity Plug ist ein von JeeLabs hergestellter Schaltkreis, auf dem sich der Beschleunigungssensor BMA020 von Bosch befindet
- Das Layout des Schaltkreises konfiguriert den Sensor bereits auf TWI
- Optional kann der Chip Interrupts generieren

# Herstellerinfo Bosch

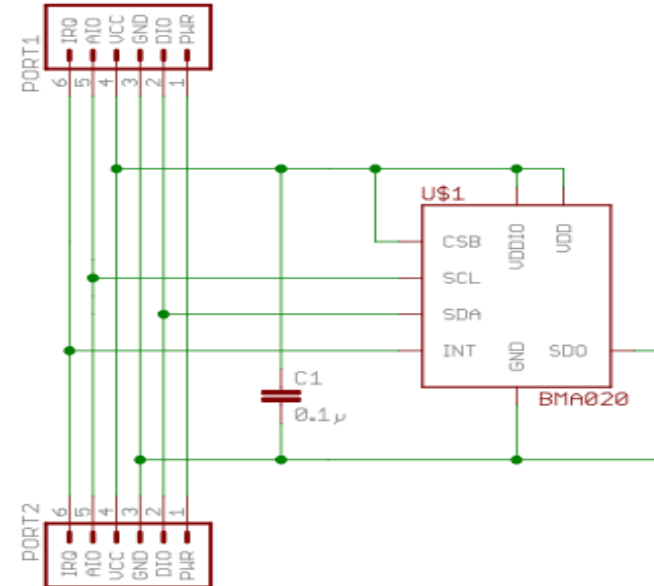
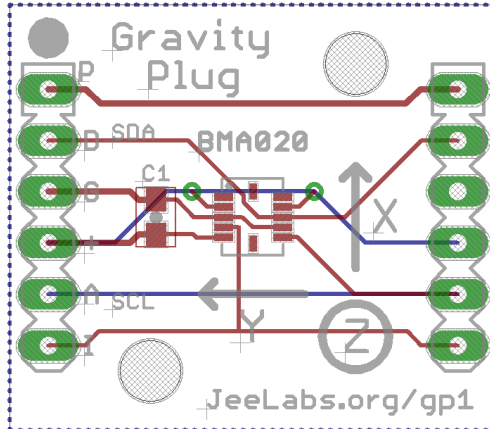
Memory Region	Register Address (hexadecimal)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	type	Default setting	
Default Settings	16h to 7Fh	reserved								reserved	NA	
	15h	SPI4	enable_adv_INT	new_data_INT	latch_INT	shadow_dis	wake_up_pause		wake_up	control	1 0 0 0 0 0 0 0b	
Operational Registers	14h	reserved			range<1:0>		bandwidth<2:0>			control	XXX 00 000b	
	13h	customer_reserved 2 <7:0>								status	NA	
	12h	customer_reserved 1 <7:0>								status	NA	
	11h	any_motion_dur		HG_hyst<2:0>			LG_hyst<2:0>			settings	NA	
	10h	any_motion_thres<7:0>								settings	NA	
	0Fh	HG_dur<7:0>								settings	NA	
	0Eh	HG_thres<7:0>								settings	NA	
	0Dh	LG_dur<7:0>								settings	NA	
	0Ch	LG_thres<7:0>								settings	NA	
	0Bh	alert	any motion	counter_HG		counter_LG		enable_HG	enable_LG	control	0	
	0Ah	reserved	reset_INT	reserved	reserved	self_test_1	self_test_0	soft_reset	sleep	control	0	
	09h	st_result	reserved		alert_phase	LG_latched	HG_latched	status_LG	status_HG	status	NA	
	08h	unused								data	NA	
	07h	acc_z<9:2> (msb)								data	NA	
	06h	acc_z<1:0> (lsb)		unused					new_data_z		data	NA
	05h	acc_y<9:2> (msb)								data	NA	
	04h	acc_y<1:0> (lsb)		unused					new_data_y		data	NA
	03h	acc_x<9:2> (msb)								data	NA	
	02h	acc_x<1:0> (lsb)		unused					new_data_x		data	NA
	01h	al_version<3:0>				ml_version<3:0>				data	NA	
00h	unused					chip_id<2:0>			data	----- 010b		

- Slave address 38h (fest kodiert)
- Schreiben: 2Byte an Slave:
  - Register address (send)
  - Register data (send)
- Selftest\_0: Initialisiert und recalibriert
- Lesen: 1Byte an Slave, dann nByte lesen
  - Register address (send)
  - Dann von Slave address lesen, Register wird automatisch erhöht

# Herstellerinfo Bosch

- Chip id                    reg 00h[bits 0,1,2]
  - fix 010b
- Range                    reg 14h[bits 3,4]
  - 00            +/- 2g
  - 01            +/- 4g
  - 10            +/- 8g
- Selftest\_0            reg 0Ah[bit 3]
  - reset, recal wenn 100b gesetzt
- Acc x,y,z            reg 03,05,07
  - Msbs der jeweiligen Acc-Werte

# Foto, Layout, Schaltplan



# Wire Library

- Aus dem Schaltplan ist ersichtlich, dass nur die 3.3V Spannung verwendet wird
- Laut Arduino-Referenz wird innerhalb der Wire Library SDA auf analog input pin 4, SCL auf aip 5 gelegt
- Damit sind die 4 Leitungen, die benutzt werden klar

# Links

- <http://arduino.cc/en/>
- <http://www.sparkfun.com/>
- <http://www.circuitsathome.com/>
- <http://jeelabs.net/>
- <http://www.pollin.de/>
- <http://www.atmel.com/>
- <http://shop.ullihome.de/wiki/USBAVRLab/ullihome>
- <http://www.mikrocontroller.net/>
- <http://www.procyonengineering.com/embedded/avr/avr-lib/>
- <http://www.watterott.com/>
- <http://www.reichelt.de/>



Vielen Dank für eure Aufmerksamkeit!