

# Der Arduino reicht mir aus: Microcontroller statt Microprocessor

Mattias Schlenker

28. Juni 2014

# Inhalt

- 1 Arduino vs. RaspberryPi
  - Technische Daten
  - Einsatzbereiche
- 2 Die Arduino-Zukunft
  - „Starke“ Arduinos mit Linux
  - Neue Microcontroller mit ARM-Kern
- 3 Gemeinsam stark!
  - Raspberry Pi mit Arduino huckepack
  - Raspberry Pi als Zentrale
- 4 Abspann

# Worin unterscheiden sich die beiden Bastlerplatinen?

Auf den ersten Blick bietet der Raspberry Pi mehr als 1000 mal soviel Rechenleistung wie ein Arduino Uno für zehn Euro Aufpreis. Da fällt die Entscheidung leicht, oder?  
Sehen wir uns die Unterschiede doch einmal näher an:

# Speicher und Rechenleistung

Plattform	Arduino	Raspberry Pi	Faktor
RAM/Variablenspeicher	2048	536870912	262144

# Speicher und Rechenleistung

Plattform	Arduino	Raspberry Pi	Faktor
RAM/Variablenspeicher	2048	536870912	262144
Festspeicher	32768	8589934592	262144

# Speicher und Rechenleistung

Plattform	Arduino	Raspberry Pi	Faktor
RAM/Variablenspeicher	2048	536870912	262144
Festspeicher	32768	8589934592	262144
Wortbreite	8Bit	32Bit	4

# Speicher und Rechenleistung

Plattform	Arduino	Raspberry Pi	Faktor
RAM/Variablenspeicher	2048	536870912	262144
Festspeicher	32768	8589934592	262144
Wortbreite	8Bit	32Bit	4
„Rechenleistung“ (MIPS)	16	1000	64

# Speicher und Rechenleistung

Plattform	Arduino	Raspberry Pi	Faktor
RAM/Variablenspeicher	2048	536870912	262144
Festspeicher	32768	8589934592	262144
Wortbreite	8Bit	32Bit	4
„Rechenleistung“ (MIPS)	16	1000	64



# IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!

# IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!
analog	6	0	0

# IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!
analog	6	0	0
SPI	1	2	2

## IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!
analog	6	0	0
SPI	1	2	2
UART (seriell)	1	1	1

# IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!
analog	6	0	0
SPI	1	2	2
UART (seriell)	1	1	1
I2C	1	1	1

# IO und Buses

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
GPIO digital	17	8	unfair! Äpfel vs Birnen!
analog	6	0	0
SPI	1	2	2
UART (seriell)	1	1	1
I2C	1	1	1

# Interrupts

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
Steigend/Fallend	2	8? Publikum?	?

# Interrupts

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
Steigend/Fallend	2	8? Publikum?	?
Wakeup-Timer	2	beliebig viele	?



# Interrupts

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
Steigend/Fallend	2	8? Publikum?	?
Wakeup-Timer	2	beliebig viele	?

# Leistungsaufnahme

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
ohne Optimierungen	40mA bei 5V	300mA bei 5V	7,5

# Leistungsaufnahme

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
ohne Optimierungen	40mA bei 5V	300mA bei 5V	7,5
optimiert	4 $\mu$ A bei 3,0V	60mA bei 3,6V	15000

# Leistungsaufnahme

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
ohne Optimierungen	40mA bei 5V	300mA bei 5V	7,5
optimiert	4 $\mu$ A bei 3,0V	60mA bei 3,6V	15000

- Optimierungen beim Arduino: BOD abschalten, ADC abschalten, Tiefschlaf, Batteriebetrieb mit 2xAAA, Takt auf 8MHz reduziert.
- Optimierungen beim Raspberry Pi: Abschalten nicht benötigter Schnittstellen, Auflöten eines optimierten Spannungswandlers, Akkubetrieb mit 3,6V.

# Leistungsaufnahme

Arduino heisst in diesem Kontext Atmega328P, Raspberry Pi meint das Model B...

Plattform	Arduino	Raspberry Pi	Faktor
ohne Optimierungen	40mA bei 5V	300mA bei 5V	7,5
optimiert	4 $\mu$ A bei 3,0V	60mA bei 3,6V	15000

- Optimierungen beim Arduino: BOD abschalten, ADC abschalten, Tiefschlaf, Batteriebetrieb mit 2xAAA, Takt auf 8MHz reduziert.
- Optimierungen beim Raspberry Pi: Abschalten nicht benötigter Schnittstellen, Auflöten eines optimierten Spannungswandlers, Akkubetrieb mit 3,6V.

# Preise

Auf den ersten Blick liegen Arduino und Raspberry Pi preislich eng beieinander. Auf den zweiten Blick:

- Raspberry Pi: 35 bis 40 €
- Arduino Uno oder Zero: 25 bis 30 €

# Preise

Auf den ersten Blick liegen Arduino und Raspberry Pi preislich eng beieinander. Auf den zweiten Blick:

- Raspberry Pi: 35 bis 40 €
- Arduino Uno oder Zero: 25 bis 30 €
- Arduino Pro Mini (Sparkfun oder Watterott): 10 €

# Preise

Auf den ersten Blick liegen Arduino und Raspberry Pi preislich eng beieinander. Auf den zweiten Blick:

- Raspberry Pi: 35 bis 40 €
- Arduino Uno oder Zero: 25 bis 30 €
- Arduino Pro Mini (Sparkfun oder Watterott): 10 €
- Arduino Pro Mini (China-Klon): 3 bis 4 €



# Preise

Auf den ersten Blick liegen Arduino und Raspberry Pi preislich eng beieinander. Auf den zweiten Blick:

- Raspberry Pi: 35 bis 40 €
- Arduino Uno oder Zero: 25 bis 30 €
- Arduino Pro Mini (Sparkfun oder Watterott): 10 €
- Arduino Pro Mini (China-Klon): 3 bis 4 €
- Atmega328P im DIL28-Gehäuse: 3 bis 5 €

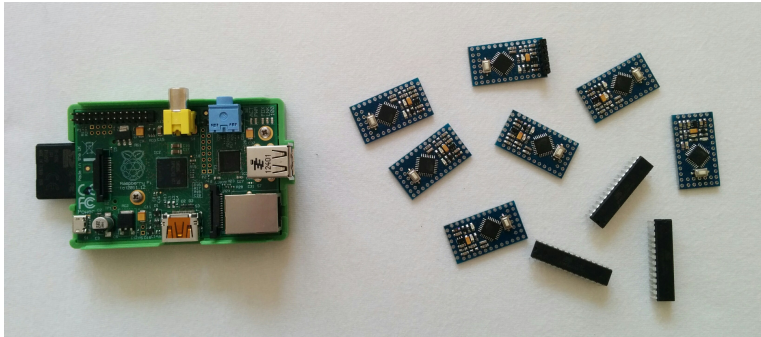
# Preise

Auf den ersten Blick liegen Arduino und Raspberry Pi preislich eng beieinander. Auf den zweiten Blick:

- Raspberry Pi: 35 bis 40 €
- Arduino Uno oder Zero: 25 bis 30 €
- Arduino Pro Mini (Sparkfun oder Watterott): 10 €
- Arduino Pro Mini (China-Klon): 3 bis 4 €
- Atmega328P im DIL28-Gehäuse: 3 bis 5 €

## Preise visualisiert

Der Raspberry links war teurer als die Arduino-Klone und nackten Atmega328P rechts!



# Arduino

- *Energiesparende* Sensoren oder Aktoren: Jahrelanger Betrieb auf zwei AAA-Zellen möglich
- *Billige* Sensoren: Totalverlust ist zu verschmerzen
- *Billige* Peripherie: Der integrierte ADC erlaubt den Anschluss von Thermistoren statt One-Wire-Temperatursensoren
- *Einfache* Sensoren und Aktoren: auch größere Stückzahlen (Sensornetze, Kunstprojekte) lassen sich mit wenigen Bauteilen schnell fertigen
- *Robuste* Softwareentwicklung: Weit weniger Abstraktionsschichten, bessere Vorhersehbarkeit

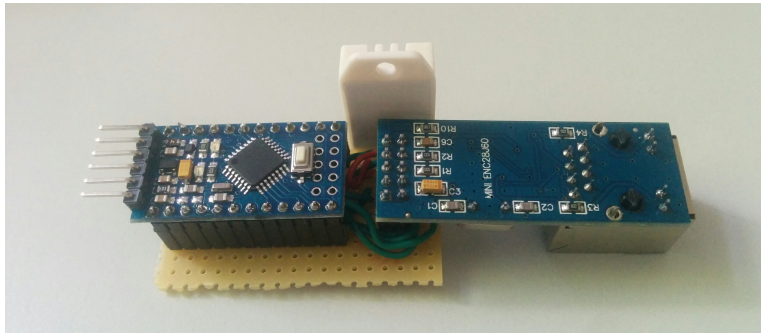
# Temperatursensor mit Webserver

Der simpelste aller Webserver: Jedes auf Port 80 eingehende Paket - egal, was drinsteht - wird einfach mit einer Webseite (Temperatur und Luftfeuchte) beantwortet. Weniger als 20k Binärcode gesamt!

- Arduino Pro Mini oder nackter Atmega328P
- Enc28J60 Ethernet (SPI-Anbindung)
- DHT11/22 Sensor
- Gesamtkosten: ca. 12 €

# Temperatursensor mit Webserver

So sieht der verlötete Aufbau aus:



Code auf <https://github.com/mschlenker>

```
21 #include <UIPEthernet.h>
22 #include <dht.h>
23
24 #define DHT22_PIN 6
25
26 const char header[] = "HTTP/1.0 200 OK\r\n"
27   "Content-Type: text/plain\r\n\r\nLuftfeuchte: ";
28 const byte myMac[6] = { 0x23, 0x23, 0xDE, 0xAD, 0xBE, 0xEF };
29 const IPAddress myIP(192,168,2,5);
30
31 EthernetServer server = EthernetServer(80);
32 dht DHT;
33
34 void setup() {
35   Serial.begin(9600);
36   Ethernet.begin(myMac, myIP);
37   server.begin();
38   Serial.println("\nListening...");
39 }
40
41 void loop() {
42   if (EthernetClient client = server.available()) {
43     Serial.println("\nAnswering...");
44     client.print(header);
45     int chk = DHT.read22(DHT22_PIN);
46     client.print(DHT.humidity, 1);
47     client.print("\r\nTemperatur: ");
48     client.print(DHT.temperature, 1);
49     client.print("\r\n");
50     client.stop();
51     Serial.println("\nDisconnect...");
52   }
53 }
```

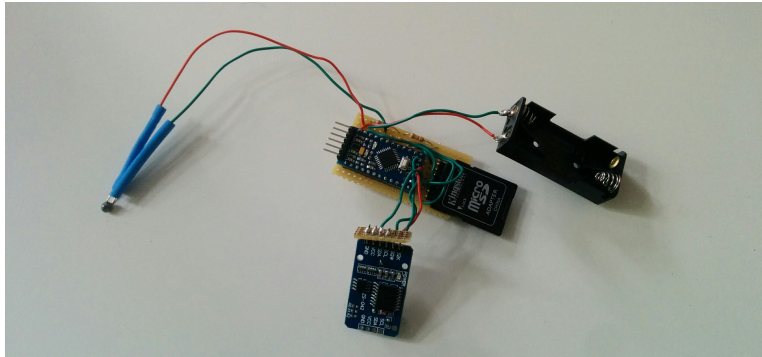
# Logger Sensor (Temperatur, Erdfeuchte, Licht)

- Arduino Pro Mini oder nackter Atmega328P
- $\mu$ SD-Kartenadapter (geschenkt!)
- diverse Analogsensoren (NTC, resistive Erdfeuchte, Photo-Widerstand...)
- hier zusätzlich: RTC
- Gesamtkosten: 5 bis 10 Euro €
- ca. ein Jahr Laufzeit auf 3xAA bei Optimierung



# Sensor mit SD-Logging

So sieht der verlötete Aufbau aus:

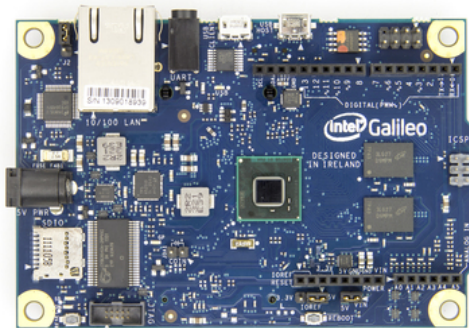


# Raspberry Pi

- *Billige* Basis für Sensornetze
- *Stromsparende* „Außenstelle“: Solarbetrieb mit relativ kleinem Panel möglich
- *Große* Flexibilität dank „normalem“ Linux
- *Kleine Server* für den Hausgebrauch
- *Stark genug* um bspw. DVB-T oder Webcam zu streamen
- *Lautloser* Thin Client für Windows- oder Linux-Server

# Galileo

Intel verschenkt an Unis Galileo-Boards im Arduino-Shield-Layout.  
Das besondere: Statt Microcontroller kommt ein Pentium zum Einsatz:



# Galileo

Intel verschenkt an Unis Galileo-Boards im Arduino-Shield-Layout.  
Das besondere: Statt Microcontroller kommt ein Pentium zum Einsatz:

- Intels Versuch, mit Embedded x86 in die Nähe von Microcontrollern zu kommen
- Referenzboard für Intels Quark SoC x1000
- Arduino-Code wird in ELF-Objekte kompiliert
- recht langsames IO
- *Mattias' Meinung:* The worst of both worlds

Gemeinsam mit TI und dem BeagleBone-Projekt entwickelter „janusköpfiger“ Arduino aus TI Sitara und Atmega32u4, Leistung der Linux-Seite höher als beim Raspberry Pi:



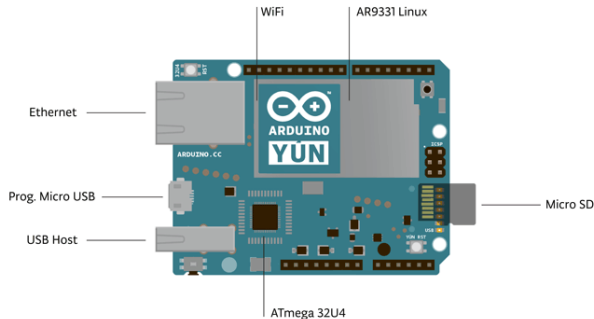
# Arduino Tre

Gemeinsam mit TI und dem BeagleBone-Projekt entwickelter „janusköpfiger“ Arduino aus TI Sitara und Atmega32u4, Leistung der Linux-Seite höher als beim Raspberry Pi:

- Von TI als Referenzplattform betrachtet
- 100% Code-Kompatibilität zu Arduino Leonardo
- 100% Code-Kompatibilität zu Beaglebone Black
- recht teuer - Entwicklerkit 149 €- final 60 bis 80 €
- *Mattias' Meinung:* Tolles Konzept, hohe Leistung, aber hoher Preis und große Platine

# Arduino Yún

Janusköpfiger Arduino aus MIPS-Prozessor und Atmega32u4, Formfaktor des Uno, Leistung der Linux-Seite deutlich kleiner als beim Raspberry Pi, dafür incl. WLAN und schnell angebundenem Ethernet:



# Arduino Yún

Janusköpfiger Arduino aus MIPS-Prozessor und Atmega32u4, Formfaktor des Uno, Leistung der Linux-Seite deutlich kleiner als beim Raspberry Pi, dafür incl. WLAN und schnell angebundenem Ethernet:

- Eigenentwicklung mit Fokus auf guter Nutzbarkeit
- 100% Code-Kompatibilität zu Arduino Leonardo
- OpenWRT basierte Distribution für die MIPS-Seite
- Straßenpreis 60 bis 75 €- kann parallel als WLAN Access Point genutzt werden
- *Mattias' Meinung:* Tolles Konzept, passende Leistung, sollte etwas günstiger sein
- *Der Clou:* Die Software - einfacher geht es nicht!



# Arduino Yún - Software-Integration

Eine neue Sammlung von Bibliotheken für die Arduino-IDE abstrahiert die Kommunikation zwischen Microcontroller- und Microprozessorseite:

- Bridge-Bibliothek für Zugriff auf viele Netzwerkfunktionen aus Arduino-Sketches - nur eine Programmiersprache nötig!
- „Mailbox“ (toter Briefkasten) für den Datenaustausch zwischen Microcontroller und Microprocessor
- Integration von Temboo - abstrahiert verschiedene Webdienste
- Voller Paketumfang von OpenWRT
- Einfaches Webinterface für die Einrichtung - Luci bekannt von OpenWRT

# Arduino Yún - Twitter-Beispiel

```
void loop() {  
  String tweetText("TEST: Der Yun l auft seit " + String(millis()) + " ms.");  
  // Temboo-Objekt initialisieren  
  TembooChoreo StatusesUpdateChoreo;  
  StatusesUpdateChoreo.begin();  
  StatusesUpdateChoreo.setAccountName(TEMBOO_ACCOUNT);  
  StatusesUpdateChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);  
  StatusesUpdateChoreo.setAppKey(TEMBOO_APP_KEY);  
  // Temboo mit Twitter verkn pfen  
  StatusesUpdateChoreo.setChoreo("/Library/Twitter/Tweets/StatusesUpdate");  
  StatusesUpdateChoreo.addInput("AccessToken", TWITTER_ACCESS_TOKEN);  
  StatusesUpdateChoreo.addInput("AccessTokenSecret", TWITTER_ACCESS_TOKEN_SECRET);  
  StatusesUpdateChoreo.addInput("ConsumerKey", TWITTER_CONSUMER_KEY);  
  StatusesUpdateChoreo.addInput("ConsumerSecret", TWITTER_CONSUMER_SECRET);  
  // Tweet absetzen  
  StatusesUpdateChoreo.addInput("StatusUpdate", tweetText);  
  // Return Code ermitteln  
  unsigned int returnCode = StatusesUpdateChoreo.run();  
  if (returnCode == 0) {  
    Serial.println("-D Tweet erfolgreich gesendet!");  
    StatusesUpdateChoreo.close();  
    delay(3600000); // Zehn Stunden warten  
  } else {  
    Serial.println(":-( Fehler!");  
    while (StatusesUpdateChoreo.available()) {  
      char c = StatusesUpdateChoreo.read();  
      Serial.print(c);  
    }  
    StatusesUpdateChoreo.close();  
    delay(90000); // 90 Sekunden warten  
  }  
}
```

Without title somethink is missing.

Without title somethink is missing.