

# Mikrocontroller-Programmierung

Ruben 14felgenh

24. Januar 2023

# Disclaimer

Ich laber möglicherweise Müll



iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™



iStock  
by Getty Images

iStock  
by Getty Images

iStock  
by Getty Images

iStock  
by Getty Images

iStock

iStock

iStock



iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock  
by Getty Images™

iStock

iStock



*Coca-Cola*

This Dispenser is temporarily Out of Operation  
Please wait while the dispenser performs required nightly maintenance.

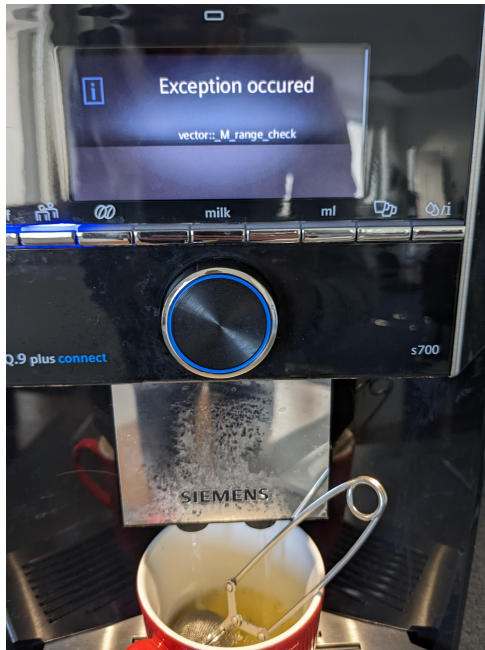
**DO NOT SHUT DOWN THE DISPENSER.**



915:48:58

Downloading Updates...





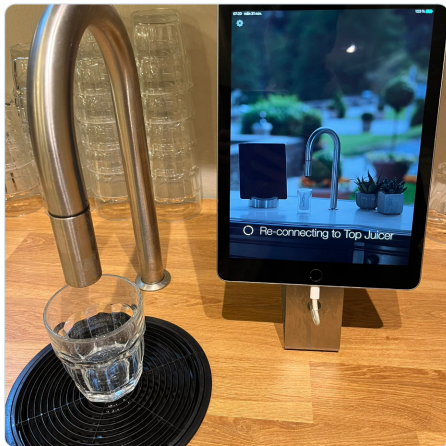




Hanna Gunnarsson   
@gunnarssonhanna · [Follow](#)



Inget internet = ingen juice. Världens sämsta uppfinning detta, att juicen ska komma ur en ipad.



6:23 AM · Nov 21, 2022



[Read the full conversation on Twitter](#)

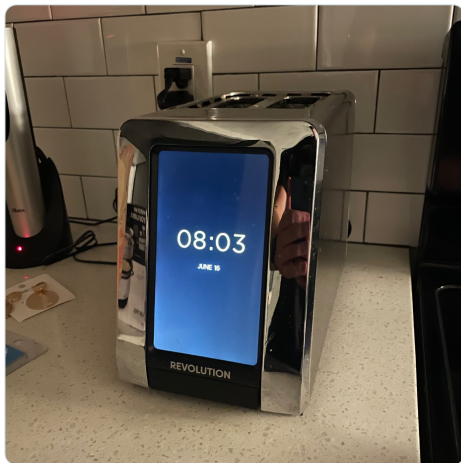


**Evan Hill**

@EvanHillHB · [Follow](#)



THIS TOASTER HAS A FUCKING HOME SCREEN



3:39 AM · Jun 17, 2022



[Read the full conversation on Twitter](#)



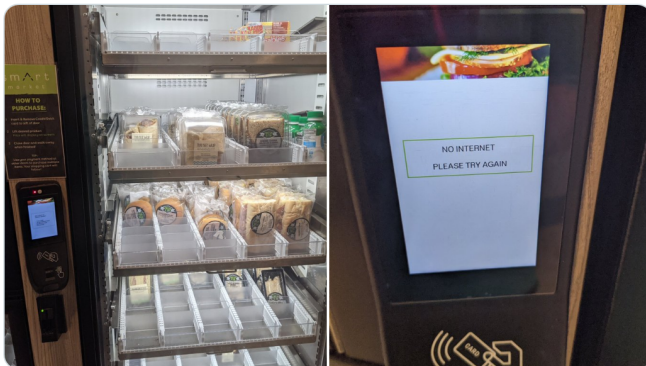
**MoonmanND**

@MoonmanND · [Follow](#)



The only vending machine has to connect to the internet apparently, so today I just don't get to eat breakfast I guess.

[@internetofshit](#)



9:26 AM · Jun 16, 2022





**pictures of the end**  
@neighbours\_wifi · [Follow](#)



6:00 PM · May 17, 2022





**Hard Pass**  
@HardPass4 · [Follow](#)



Hey [@TraegerGrills](#) - who the HELL sends a software update on thanksgiving?!?



3:03 PM · Nov 25, 2021



[Read the full conversation on Twitter](#)



**Cursed Architecture**  
@CursedArchitect · [Follow](#)



12:07 PM · Aug 15, 2019





Josh Barro   
@jbarro · Follow



People make fun of smart appliances, but don't you wish your dryer sent you a monthly digest email?

## [ Usage Summary ]



### Your Favorite Cycles

Cotton/Normal  
(43%)

Heavy duty (21%)

Perm.Press/Casual  
(21%)

*See what you're missing - Learn more about available cycles and options, including what our favorites are!*

2:35 PM · May 5, 2022



**Advanced Pikelet Threat** 🧑‍💻 🍷 ✨ 🐉 🇺🇦

@pikelet · [Follow](#)



Are you kidding me



2:19 AM · Dec 31, 2021 from Auckland, New Zealand



[Read the full conversation on Twitter](#)



## DDoS-Attacke legt Twitter, Netflix, Paypal, Spotify und andere Dienste lahm

Wegen einer massiven DDoS-Attacke sind die Services großer US-Internetdienste, darunter unter anderem Twitter, Paypal, Netflix und Spotify, am Freitagabend in Teilen der USA und Europas zeitweise nicht zu erreichen.

Leszeit: 2 Min.  speichern

  440

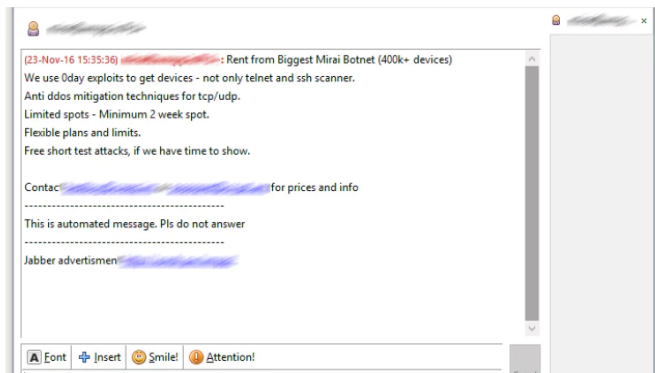


## Kriminelle bieten Mirai-Botnetz mit 400.000 IoT-Geräten zur Miete an

Zwei Hacker sollen derzeit auf Kundenfang gehen und Mirai-Botnetze zur Miete anbieten. Dabei werben sie unter anderem mit einer optimierten Version des DDoS-Tools Mirai. Sicherheitsforscher zeigen den Live-Status des gesamten Mirai-Botnetzes an.

Lesezeit: 2 Min.  speichern

  61



(Bild: [Bleepingcomputer.com](#))

# Internet of Things

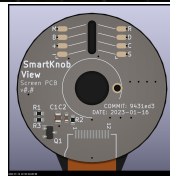
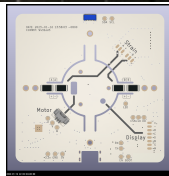
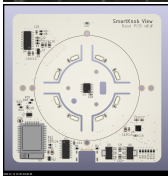
- IOT-Geräte sind...
  - vooooooooooll cooooooooooool!!!!!!!111

# Internet of Shit

- IOT-Geräte sind...
  - vooooooooooll cooooooooooool!!!!!!!111
  - unsicher
  - kurzlebig
  - stromhungrig für unnötigen Scheiß, den kein Mensch braucht
- Die Hersteller von IOT-Geräten sind...
  - grob fahrlässig / inkompetent
  - selten an offenen Standards interessiert
- Mehr nützliche Gadgets:
  - <https://twitter.com/internetofshit>
  - <https://www.reddit.com/r/theinternetofshit/>



# Positives Beispiel





# Begriffserklärung

## **Microprocessor (MPU)**

CPU ohne (bzw. mit externer) Peripherie.

## **Microcontroller (MCU)**

CPU mit etwas Peripherie im selben Package.

## **System on a Chip (SoC)**

Rechnersystem bestehend aus nur einem IC.

## **Integrated Curcuit (IC)**

Schaltkreis in einem Gehäuse mit Pins.

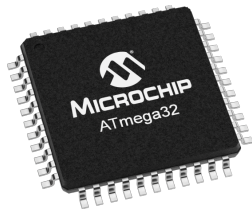
## **Embedded System**

Rechner, der in ein anderes Produkt eingebettet ist.



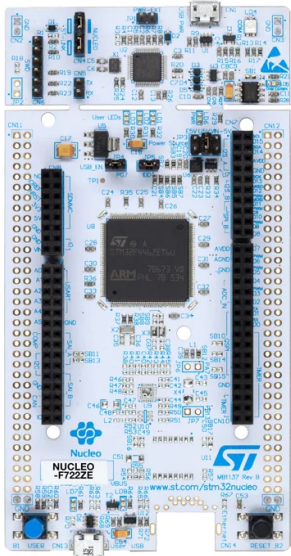
# Atmel AVR / microchip ATmega1281

- 8-Bit ARM Cortex-M4 CPU
- Bis zu 16 MHz
- 8 KB RAM
- 128 KB Flash
- Counter / Timer, PWM, ADC, DAC, USART, I2C, SPI, JTAG
- 0,9 mW (Active Mode, 1 MHz)
- 0,18  $\mu$ W (Power-down Mode)



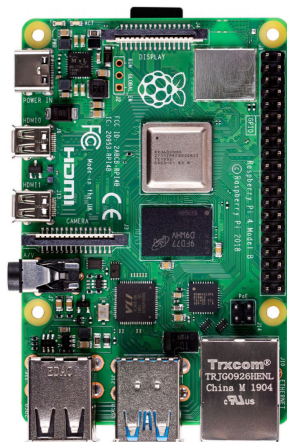
# STMicroelectronics STM32F722ZE

- 32-bit ARM Cortex-M7 CPU
- SP-FPU
- Bis zu 216 MHz
- 256 KB RAM
- 16 KB L1 Cache
- 512 KB Flash
- Counter / Timer, PWM, ADC, DAC, USART, I2C, SPI, JTAG, CAN, I2S, USB



# Broadcom BCM2711

- 64-Bit ARM Cortex-A72
- DP-FPU
- Quad Core
- Bis zu 1,5 GHz
- Hier 8 GB RAM
- Counter / Timer, PWM, ADC, DAC, USART, I2C, SPI, JTAG, HDMI, CAN, I2S, USB, Ethernet, WiFi, Bluetooth
- 2,7 W (Idle)
- 6,4 W (Max)



# Was brauche ich?

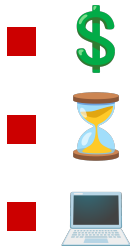
# Was brauche ich?



# Was brauche ich?



# Was brauche ich?








# Was brauche ich?

- 
- 
- 
- 



# Was brauche ich?

- 
- 
- 
- 
- 

```

\_(ツ)_/
  
```

amazon Deliver to Germany    EN -

[All](#) [Today's Deals](#) [Customer Service](#) [Gift Cards](#) [Registry](#) [Sell](#)

[Back to results](#)



## Raspberry Pi 4 Model B 8GB

Brand: [Raspberry Pi](#)

★★★★☆ 267 ratings

\$218<sup>00</sup>

\$54.77 Shipping & Import Fees Deposit to Germany [Details](#)  
 Available at a lower price from [other sellers](#) that may not offer free Prime shipping.

Brand	Raspberry Pi
Operating System	Linux
Memory Storage Capacity	8 GB
Ram Memory Installed Size	8 GB
<a href="#">See more</a>	

**Note:** Products with electrical plugs are designed for use in the US. Outlets and voltage differ internationally and this product may require an adapter or converter for use in your destination. Please check compatibility before purchasing.

~\\_(`ツ)\\_/~

Z.B. mouser.com:

(Listenpreis bei 10k Stück:  
9,7361 USD)

### Verfügbarkeit

Lagerbestand: 0 [Benachrichtigen Sie mich, wenn das Produkt auf Lager ist.](#)  
**Sie können dieses Produkt immer noch nachbestellen.**

Auf Bestellung: 8.913

Lieferzeit ab Hersteller: 104 Wochen [?](#)

Lange Lieferzeit für dieses Produkt.

Minimum: 1 Vielfache: 1

Menge eingeben:  [Kaufen](#)

### Preis (EUR)

Menge	Stückpreis	Erw. Preis
1	22,32 €	22,32 €
10	20,52 €	205,20 €
25	19,67 €	491,75 €
100	17,33 €	1.733,00 €
250	16,48 €	4.120,00 €
360	15,41 €	5.547,60 €
1.080	<a href="#">Kostenvoranschlag</a>	

# Grundsätzliche Erwägungen

## Bare Metal

Euer Prozessor führt nur eure `main()` aus (eigentlich noch etwas mehr...).

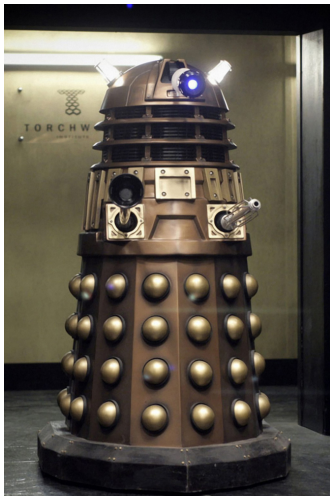
## Betriebssystem

z.B. Linux, ...

## Grenzfälle

z.B. arm Mbed OS, FreeRTOS, ...

# Grundsätzliche Erwägungen



Average Bare Metal  
Programmer



Average Embedded Linux  
Programmer

# The Classic Way

- C
- I/O direkt über Register
- Crosscompiling-Toolchain, z.B. `arm-none-eabi-gcc`
- Hard- und Software zum Flashen via JTAG



```
1  #include <avr/io.h>
2  #include <util/delay.h>
3
4  int main() {
5      DDRG |= 0x02;
6      while (true) {
7          _delay_ms(1000);
8          PORTG ^= 0x02;
9      }
10 }
```

# The STM32 Way

- Still C
- Hardware Abstraction Library (HAL)
- Kompilieren und Flashen via STLINK mit STM32CubeIDE

```
1  #include "hal.h"
2
3  int main() {
4
5      HAL_Init();
6      SystemClock_Config();
7      LED_Init(LED1);
8
9      while (true) {
10         LED_Toggle(LED1);
11         HAL_Delay(1000);
12     }
13 }
```

# The STM32 Way

## Intermission



# The STM32 Way

- + All-in-One-Lösung
- + Grafische Hardware-Konfiguration
- + Viele Beispiele
- + Keine zusätzliche Hardware erforderlich
  
- Eclipse
- Bugs
- Beispiele sind schlechter Code
- Schlechte / falsche Dokumentation der HAL / BSP
- Schlecht durchdachte Features (z.B. `USER CODE BEGIN ...`)
- Leichter Vendor Lock-In

# Hardware-Interaktion

Mit Polling:

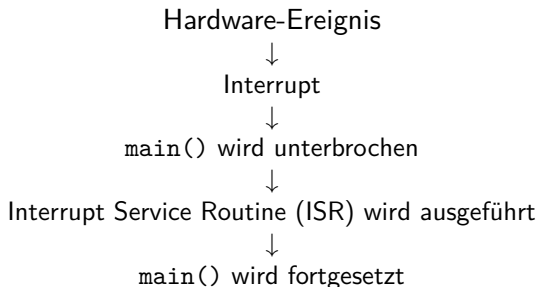
```
1  int main() {
2
3      ...
4
5      GPIO_PinState last_state = GPIO_PIN_RESET;
6
7      while (true) {
8
9          const GPIO_PinState current_state
10             = HAL_GPIO_ReadPin(USER_BUTTON_GPIO_PORT, USER_BUTTON_PIN);
11
12             if (current_state == GPIO_PIN_SET
13                 && last_state == GPIO_PIN_RESET) {
14                 LED_Toggle(LED1);
15             }
16
17             last_state = current_state;
18         }
19     }
```

# Hardware-Interaktion

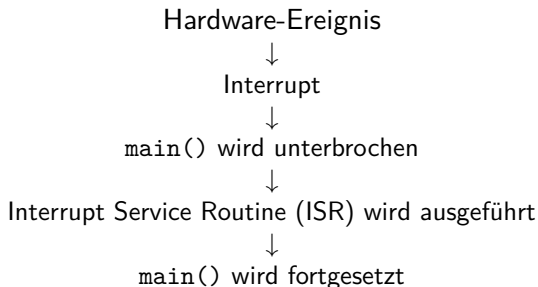
## Mit Interrupts:

```
1  int main() {
2
3      ...
4
5      while (true) {
6          /* nothing */
7      }
8  }
9
10 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
11     if (GPIO_Pin == USER_BUTTON_PIN) {
12         BSP_LED_Toggle(LED1);
13     }
14 }
```

# Interrupts



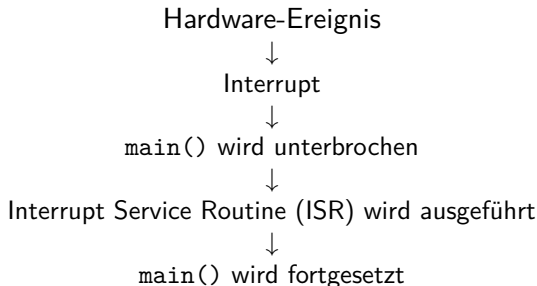
# Interrupts



Anmerkungen:

- I.A. müssen ISRs registriert werden.

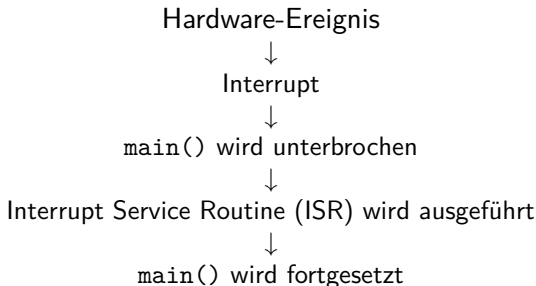
# Interrupts



Anmerkungen:

- I.A. müssen ISRs registriert werden.
- Das Nested Interrupt Vector Control (NVIC) steuert, welche Interrupts aktiviert („maskiert“) sind.

# Interrupts



Anmerkungen:

- I.A. müssen ISRs registriert werden.
- Das Nested Interrupt Vector Control (NVIC) steuert, welche Interrupts aktiviert („maskiert“) sind.
- **Was passiert, wenn ein Interrupt während eines Interrupts ausgelöst wird?**

# Programm-Flow

Woher weiß die CPU, welche Instruktionen sie ausführen muss?



# Programm-Flow

Woher weiß die CPU, welche Instruktionen sie ausführen muss?

Program Counter (PC) / Instruction Pointer (IP):

- Register mit Pointer auf die nächste Instruktion
- Wird meistens stumpf inkrementiert
- Jump-Statements setzen den PC explizit
- Bei Funktionsaufrufen und Interrupts:  
Schiebe außerdem PC+1 auf den Stack.

# Programm-Flow

```
1 int main() {
2     asm("nop");
3     for (int i = 10; i != 0; i--) {
4         asm("nop");
5     }
6     asm("nop");
7 }
```

# Programm-Flow

```
1 0000820c <main>:  
2      820c:      e1a00000      nop  
3      8210:      e3a0300a      mov     r3, #10  
4      8214:      e1a00000      nop  
5      8218:      e2533001      subs   r3, r3, #1  
6      821c:      1afffffc      bne    8214  
7      8220:      e1a00000      nop  
8      8224:      e3a00000      mov     r0, #0  
9      8228:      e12fff1e      bx     lr
```

# Programm-Flow

```
1 void dummy() {
2     asm("nop");
3 }
4
5 int main() {
6     asm("nop");
7     dummy();
8     asm("nop");
9 }
```

# Programm-Flow

```
1 0000820c <dummy>:  
2     820c:      e1a00000      nop  
3     8210:      e12fff1e      bx      lr  
4  
5 00008214 <main>:  
6     8214:      e92d4010      push   {r4, lr}  
7     8218:      e1a00000      nop  
8     821c:      ebfffffa      bl     820c  
9     8220:      e1a00000      nop  
10    8224:      e3a00000      mov    r0, #0  
11    8228:      e8bd4010      pop    {r4, lr}  
12    822c:      e12fff1e      bx     lr
```

# Wieso eigentlich kein C++?

C++ hat viele tolle Features:

- Klassen
- Mehr Memory Safety
- Mehr Type Safety
- Exceptions
- Virtual Functions
- Templates

# Wieso eigentlich kein C++?

C++ hat viele tolle Features:

- Klassen
- Mehr Memory Safety
- Mehr Type Safety
- **Exceptions**
- **Virtual Functions**
- **Templates**

# Embedded Linux

- Mit Package-Manager
  - Raspberry OS / Raspbian (apt)
  - Ubuntu Core (apt)
- Ohne Package-Manager
  - Yocto
  - Buildroot
- Honorable Mentions
  - Android
  - OpenWRT



# Buildroot

- Tool zum Erstellen von Linux-Distributionen
- Abgespecktes Linux:
  - Kein Paketmanager
  - Meistens keine GNU Core Utils, stattdessen busybox (cat, cp, ls, ...)
  - Meistens kein systemd, stattdessen z.B. systemV
- Basiert auf make
- Viele Pakete (z.B. X.org, Qt, GStreamer, verschiedene Linux-Kernel)
- Erstellt tar.gz-Archive von boot- und root-Partition.
- Stellt auch Toolchain für Cross-Compiling zur Verfügung.

# Buildroot

- Bauen dauert ewig
- Schwierig zu konfigurieren
- Änderungen machen gerne was kaputt
- Es gibt nicht alles als Paket und Pakete selber bauen nervt
- Pakete Updaten ist super nervig

# Fazit

- IOT-Geräte sind häufig schlecht.
- Aber: Wir können das besser!
- Aktuell ist Mikrocontroller Kaufen schwieriger als Mikrocontroller Programmieren.
- STM32 ist einfach, aber nervig.