

Einführung in die Welt der Microcontroller



Inhaltsverzeichnis

- 1 Übersicht
 - Möglichkeiten
 - Einsatz

- 2 Microcontroller
 - Grundlegendes
 - Peripherie
 - Interrupts
 - ADC
 - Timer
 - Rest

- 3 Schaltungen
 - Grundsaltungen für ATmega32
 - Stromversorgung

- 4 Sonstiges
 - Fuse-Bits, Sprachen, Programmer
 - Hilfe, weitere Informationen

Was ist ein Microcontroller?

Microcontroller

Ein Microcontroller ist ein Chip, der neben Prozessor und Arbeitsspeicher weitere Peripherie mitbringt.

Komponenten:

- Prozessor
- Arbeitsspeicher
- Flash-Speicher
- Eingabe-/Ausgabe
- Analog-/Digitalwandler
- EEPROM
- Timer/Counter
- Schnittstellen: UART, I2C (TWI), UART, USB. . .

Wie schnell ist ein Mikrocontroller?

Eckdaten:

- Taktrate: bis 20MHz (8Bit)
- SRAM: bis 4kB
- FLASH: bis 128kB
- EEPROM: einige kB

Wie schnell ist ein Mikrocontroller?

Eckdaten:

- Taktrate: bis 20MHz (8Bit)
- SRAM: bis 4kB
- FLASH: bis 128kB
- EEPROM: einige kB

LANGSAM?

Wie schnell ist ein Mikrocontroller?

Eckdaten:

- Taktrate: bis 20MHz (8Bit)
- SRAM: bis 4kB
- FLASH: bis 128kB
- EEPROM: einige kB

LANGSAM?

kommt drauf an, was man machen will. . .

Wofür kann man einen Microcontroller einsetzen?

allgemein:

- Messen
- Steuern
- Kommunikation mit anderen Geräten (PCs/Microcontroller)

konkret (Beispiele):

- MP3-Player
- Lampe
- Alarmanlage
- Motorsteuerung
- Alkoholsensor
- USB-Geräte (Tastatur, Massenspeicher)
- Temperaturregler
- ...

Atmega32

(XCK/T0) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIN0) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

Prozessor

- Herz des Microcontrollers
- beinhaltet Arithmetikfunktion, Stack Pointer, Register. . .
- RISC-Architektur; gute Optimierungsmöglichkeit für Compiler
- Unterscheidung anhand der Breite des internen Datenbusses: 4bit, 8bit, 16bit. . .
- Taktraten von 1MHz (intern) bis hin zu 16 MHz mit externem Quarz (z.B. Atmega32)
- 32 8-bit-breite Register; R26 bis R31 aber auch 3 16-bit breite Register

Speicher und Register

Programmspeicher:

- Flash-Speicher mit dem Programm
- mind. 1.000-mal wieder beschreibbar
- Programm kann somit aktualisiert/erweitert/ersetzt werden
- ISP: MC kann innerhalb der Schaltung programmiert werden

Datenspeicher (SRAM):

- Speicher für temporäre Daten, die während der Laufzeit anfallen
- Unterscheidung von SRAM-Bereich und Registerbereich
- Registerbereich: Spiegelung der der Register, sowie Port-Zustände
- SRAM-Bereich: eigentlicher SRAM mit Stack

EEPROM:

- Daten bleiben stromlos erhalten
- 100.000 Schreib-/Löschzyklen
- ideal z.B. für Konfigurationsdateien

Ports

- Ports (und Peripherie im allgemeinen) erlaubt dem MC, mit der Außenwelt zu kommunizieren
- jeweils 8-Datenleitungen werden zu einem Port zusammengefasst
- Ports werden alphabetisch nummeriert: PORTA, PORTB, PORTC...
- Datenleitungen erlauben bidirektionale Kommunikation: senden und empfangen
- **vor** der Verwendung muss die Datenrichtung im Data-Direction-Register (DDR) angegeben werden

Beispiel

```
DDRA = 21;   ⇒ PA4, PA2, PA0 Ausgang, Rest Eingang  
PORTA = 3;   ⇒ PA0, PA2 high, PA4 low
```

Interrupts

sequenzielle Abarbeitung von Befehlen

Microcontroller verarbeiten Befehle sequenziell, nicht parallel

Problem: schnelles Reagieren auf Ereignisse

1 Pollen:

in einer Schleife Zustand überprüfen, bis er sich geändert hat

2 Interrupt:

tritt Ereignis ein, unterbricht der Prozessor seine Ausführung, merkt sich die Stelle, an der er gerade war, und führt eine Interrupt-Service-Routine aus

AD-Wandler

- 10-Bit-ADC zum messen von Spannungen
- $2^{10} = 1024 \Rightarrow 1024$ unterschiedliche Spannungsstufen messbar
- Vergleich mit verschiedenen Spannungen
- Unterstützung von Interrupts

Timer/Counter

- Einheit, die eine Zahl mit einer speziellen Frequenz erhöht
- Unterstützung für interne und externe Zählfrequenz
- Atmega32: drei Timer/Counter mit 8-Bit und 16-Bit
- 10-Bit Prescaler
- Input Capture Unit (Zählen externer Ereignisse)
- CTC-Modus (Clear Timer on Compare Match)
- Unterstützung von PWM



Der Rest

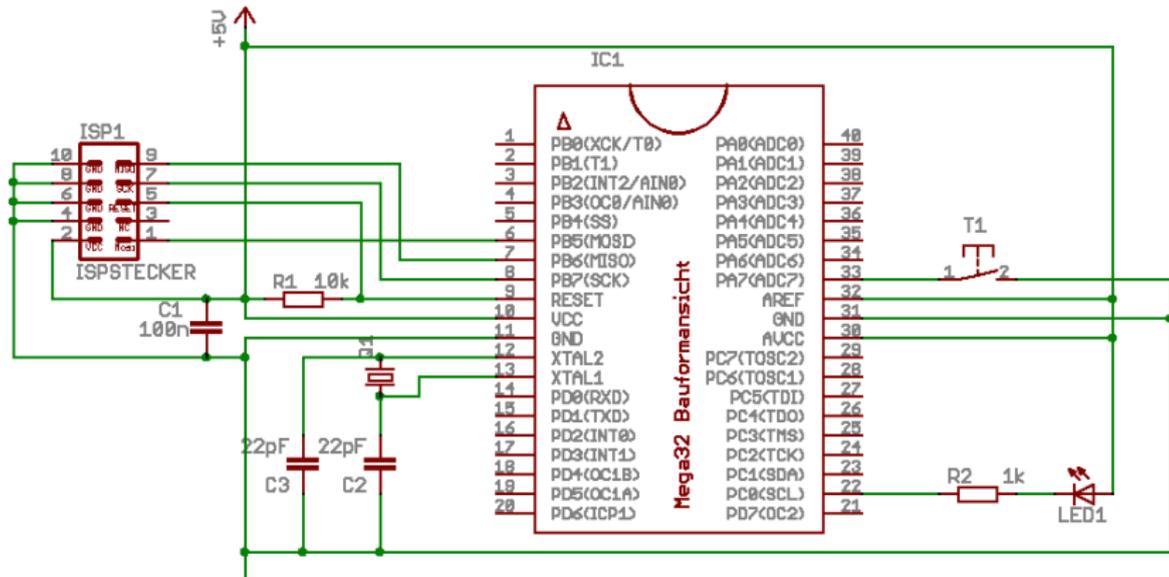
- **Watchdog:**
automatischer Reset des Programms, falls es "hängt"
- **I2C:**
serieller Zwei-Draht-Bus zur Datenübertragung
- **SPI:**
serieller Datenbus zur Datenübertragung; wird für ISP verwendet
- **UART:**
Senden und Empfangen von Daten

Grundschaltung



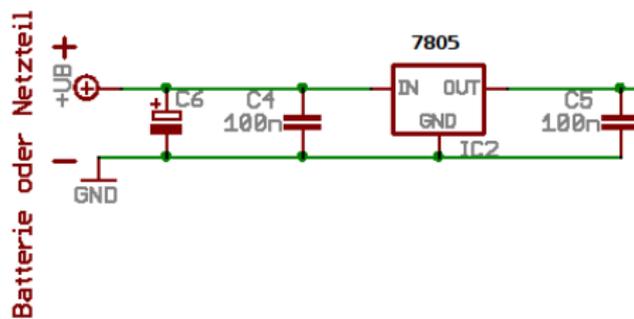
Zeichnung: roboternetz.de

Grundsaltung



Zeichnung: roboternetz.de

Stromversorgung



Dies und Das. . .

- AVR-Fuses: Konfigurationsbits, die das Verhalten des Microcontrollers ändern:
 - Fuse-Bit **setzen**: Wert **0** schreiben
 - Fuse-Bit **löschen**: Wert **1** schreiben
- verwendete Programmiersprachen: Basic (Bascom), C (gcc-avr), Exoten wie Python. . .
- Programmer: usbprog, AVRISP MKII, Atmel AVR Dragon, parallel/seriell
- Liste von Programmern:
<http://www.mikrocontroller.net/articles/AVR.In.System.Programmer>

Demo?

Hilfe/weitere Informationen

- Mailingliste der Mikrocontroller Projektgruppe des ACF
- mikrocontroller.net: Forum, Codesammlung, Artikelsammlung
- roboternetz.de: Wissensbereich mit Wiki, Forum
- AVR-Risc — Embedded Software selbst entwickeln von Roman Mittermayr