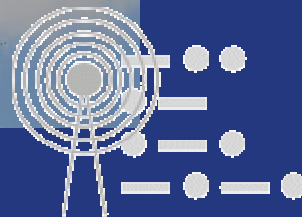


ATMEGA – Mikrocontroller Einführung TEIL 1



DH5HS
4. November 2009
Mönchengladbach

Was ist ein MCU



- Als Mikrocontroller (auch μ Controller, μ C, MCU) werden Halbleiterchips bezeichnet, die mit dem Prozessor mindestens Peripheriefunktionen auf einem Chip vereinen. In vielen Fällen befindet sich der Arbeits- und Programmspeicher ebenfalls teilweise oder komplett auf dem gleichen Chip. Ein Mikrocontroller ist praktisch ein Ein-Chip-Computersystem.

Einsatzgebiete von MCU



- **Der Mikrocontroller tritt in Gestalt von eingebetteten Systemen im Alltag oft unbemerkt in technischen Gebrauchsartikeln auf, zum Beispiel in Waschmaschinen, Chipkarten (Geld-, Telefonkarten), Unterhaltungselektronik (Videorekordern, CD-/DVD-Playern, Radios, Fernsehgeräten, Fernbedienungen), Büroelektronik, Kraftfahrzeugen (Steuergeräte für z. B. ABS, Airbag, Motor, Kombiinstrument, ESP usw.), Mobiltelefonen und sogar in Uhren und Armbanduhren. Darüber hinaus sind sie in praktisch allen Computer-Peripheriegeräten enthalten (Tastatur, Maus, Drucker, Monitor, Scanner uvm.).**

Atmel AVR

- **Atmel AVR ist eine 8-Bit-Mikrocontroller-Familie des US-amerikanischen Herstellers Atmel. Die Controller dieser Familie sind wegen ihres einfachen Aufbaus und ihrer leichten Programmierbarkeit auch bei Hobby-Anwendern weit verbreitet.**
- **Die Typen unterteilen sich in die Gruppen**
- **ATmega: große AVR-Controller mit bis zu 256 KB Flash-Speicher in 28- bis 100-poligen Gehäusen und mit integriertem Hardware-Multiplizierer.**
- **ATtiny: kleinere AVR-Controller mit bis zu 16 KB Flash-Speicher in 6- bis 20-poligen Gehäusen.**
- **AT90USB: ATmega mit integriertem USB-Controller**
- **AT90CAN: ATmega mit CAN-Controller**
- **Sondertypen: einige Spezialmodelle, z. B. zur Ansteuerung von Akku-Ladegeräten, LC-Displays und für Motoren- oder Lichtsteuerungen.**
- **AT90S: veraltete Typen, die „klassischen“ AVR**



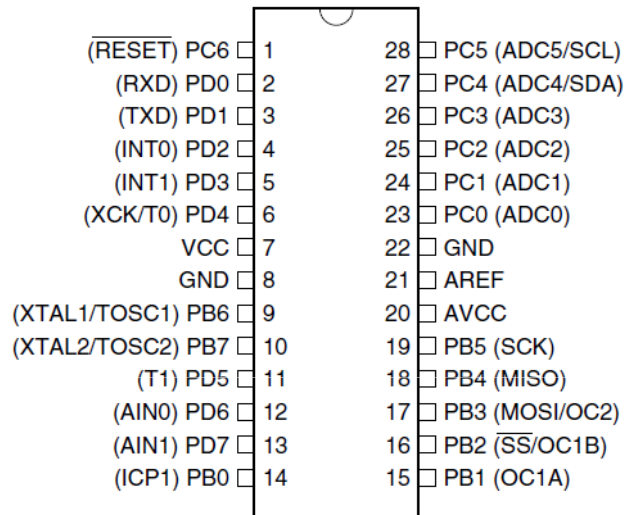
ATMEGA: Architektur



- 32 größtenteils gleichwertige Register
- 3 Pointerregister
- ca. 110 Befehle, die meist 1-2 Taktzyklen dauern
- Taktfrequenz bis 20MHz
- Betriebsspannung von 1,8-5,5 V
- Speicher
- 1-256 kB Flash-ROM
- 0-4 kB EEPROM
- 0-8 kB RAM
- Peripherie: AD-Wandler, 8- und 16-Bit-Timer mit PWM, SPI, I²C (TWI), UART, Analog-Komparator, Watchdog, externer SRAM
- JTAG bei den größeren ATmegas
- debugWire bei den neueren AVR

ATMEGA 8

PDIP



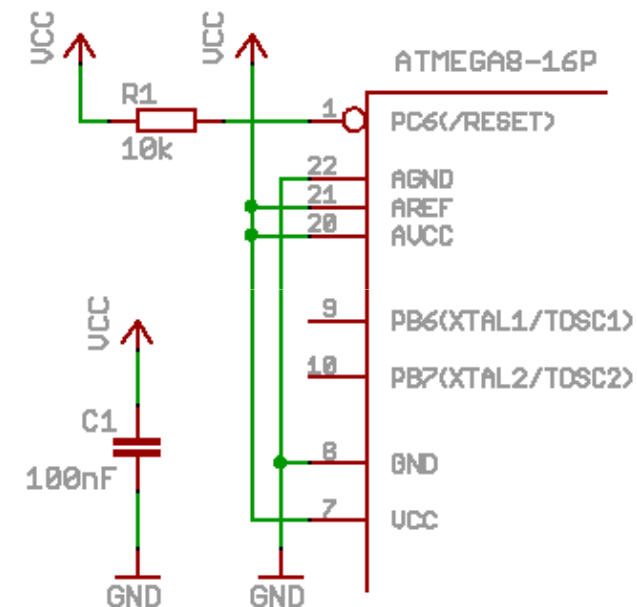
Auszug aus dem Datenblatt



Flash (Kbytes)	EEPROM (Bytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	V _{cc} (V)	A/D Channels	16-bit Timer	8-bit Timer	On Chip Oscillator	PWM Channels	UART
8	512	1K	23	16	2.7-5.5	6 10bit PDIP 8 10bit TQFP QFN/MLF	1	2	Ja	3	Ja

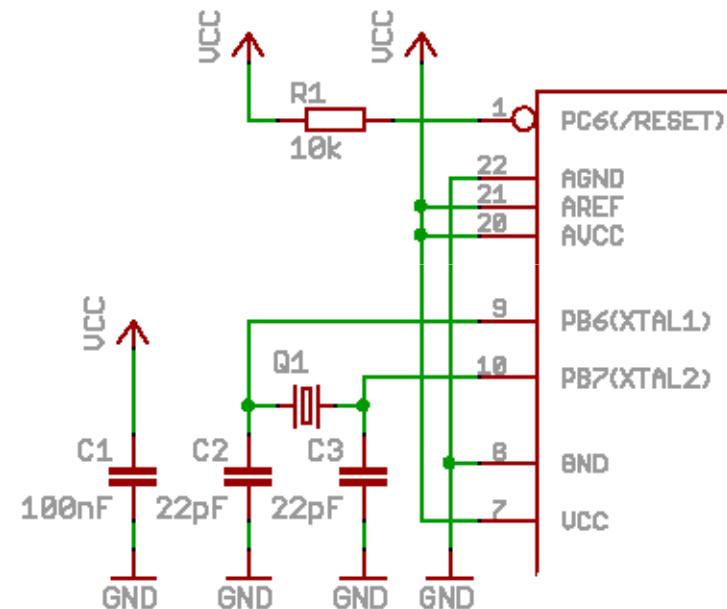
Grundsaltungen Minimal

- Um einen AVR zu betreiben ist erst einmal nicht viel Zubehör nötig. Die minimale Beschaltung besteht nur aus einer stabilen Versorgungsspannung (2,7 bis 5 Volt, je nach AVR und Taktfrequenz), einen Stützkondensator (ca. 100nF) der möglichst nahe am AVR angebracht werden sollte und einem Pull-Up Widerstand für den Reset-Pin. Dieser Pull-Up kann im Prinzip sogar weggelassen werden, allerdings wird dadurch der Reset sehr anfällig für Störungen, da der interne Widerstand sehr hochohmig (30-60 kOhm) ist.
- Selbst wenn man den Analog/Digital Wandler nicht nutzt muss man, bei den AVR Typen die ihn integriert haben, die Pins A_{VCC} und A_{GND} anschließen, da es sonst zu Fehlfunktionen der entsprechenden Ports kommen kann.



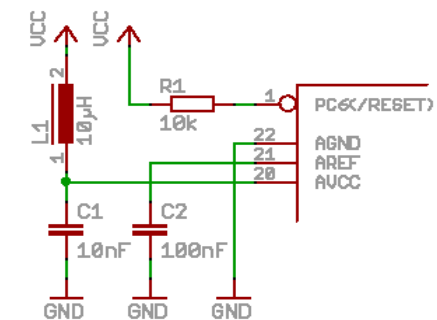
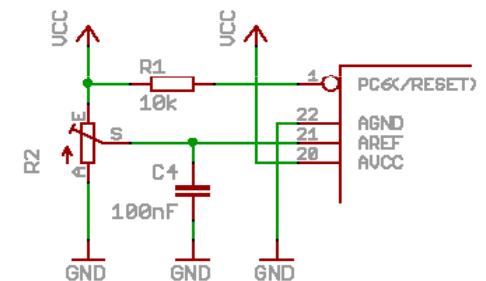
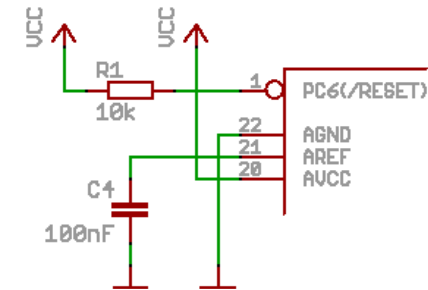
Grundsaltungen mit Quarz

- Im Normalfall reicht der interne Oszillator für den Betrieb vollkommen aus, lediglich wenn man eine höhere Taktfrequenz als die normalerweise verfügbaren 8 MHz oder eine größere Genauigkeit (zum Beispiel wenn man den UART verwenden will) benötigt lohnt es sich einen externen Quarz anzuschließen.
- Die Kondensatoren C2 und C3 sind abhängig von der verwendeten Taktfrequenz. Bei den normalerweise verwendeten Quarzen im Bereich bis 16 MHz sind 22pF ein guter Ausgangswert.
- Die Taktfrequenz von Q1 darf maximal 16 MHz (teilweise auch 20 MHz bei einigen neueren Typen) bei 5 Volt betragen. Wird eine geringere Spannung verwendet muss eventuell auch eine geringere Taktfrequenz benutzt werden. Am besten nochmal im Datenblatt nach schauen, dort gibt es meist gleich auf der ersten Seite einige Richtwerte dazu.



Grundsaltungen für A/D-Wandler

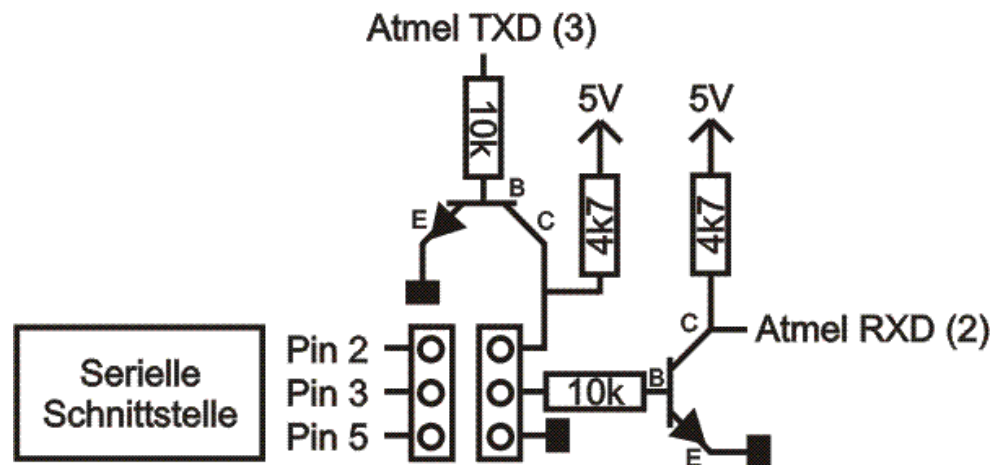
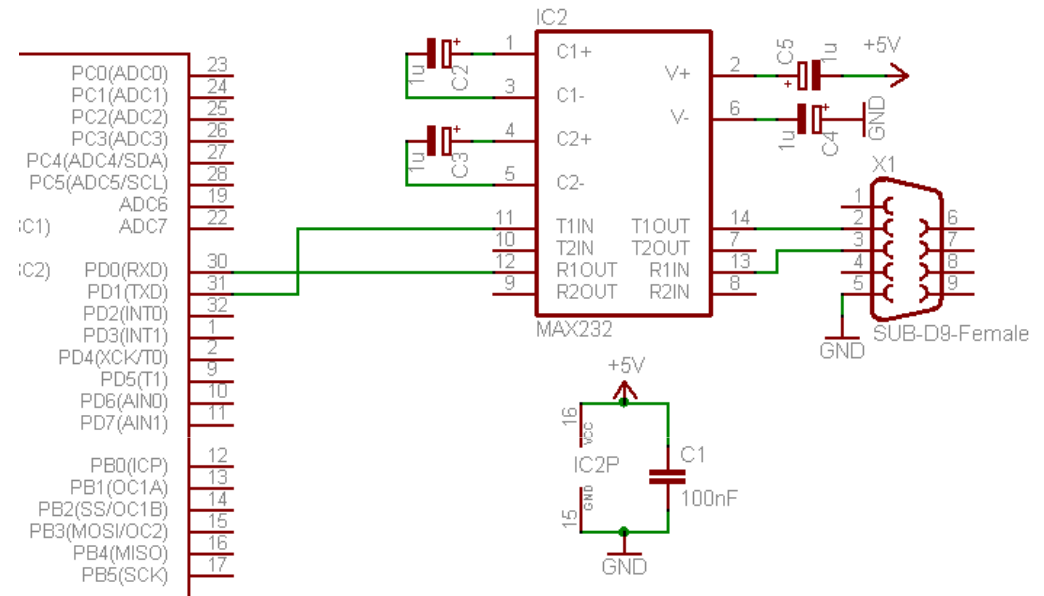
- Der ATMEGA 8 sowie die meisten neueren AVR's wie die ATMegas oder ATTinys bieten eine interne Referenzspannung von 2,56 Volt oder 5 Volt. Diese lässt sich dann über die entsprechenden ADC Register aktivieren. In diesem Modus sollte man einfach einen Kondensator von A_{REF} nach GND legen um die Referenzspannung nochmals zu stabilisieren.
- Weiterhin kann man als Referenzspannung eine beliebige, externe Spannung zwischen V_{CC} und ca. 2 Volt wählen. Wird eine Referenzspannung geringer als 2 Volt verwendet kann es sein das die Analog/Digital-Wandlung keine korrekten Ergebnisse liefert. Im zweiten Bild oben ist die Spannung über ein Poti einstellbar. Genauso gut kann man einen Spannungsteiler aus zwei Widerständen oder ähnliches benutzen. Auch hier sollte die Spannung über einen Kondensator stabilisiert werden.
- Um die Genauigkeit der Messung weiter zu erhöhen kann man noch zusätzlich die Spule L1 zwischen A_{CC} und V_{CC} schalten. Dies stabilisiert die Versorgung des ADC zusätzlich, so dass sich Störungen der Versorgungsspannung nicht so stark auf den Messwert auswirken. Weitere Techniken sind in den jeweiligen Datenblatt in dem Kapitel: "Analog Noise Canceling Techniques" beschrieben.
- An den ADCn Pins kann man jetzt die anliegende Spannung messen. Allerdings muss man aufpassen, dass diese nicht höher als die Spannung an A_{REF} wird.



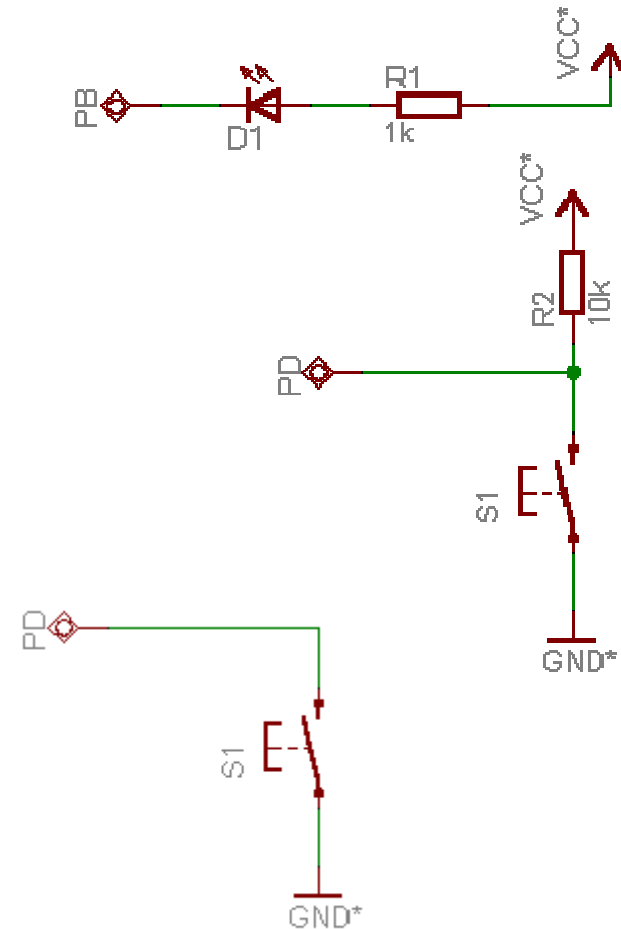
Grundsaltungen

UART – serielle Schnittstelle

- Der MAX232 ist ein Pegelwandler, der die -12V/+12V Signale an der seriellen Schnittstelle des PCs zu den 5V/0V des AVR's kompatibel macht.
- Oder einfach mit Transistoren

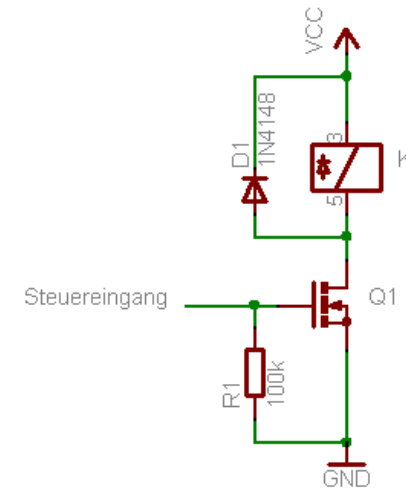
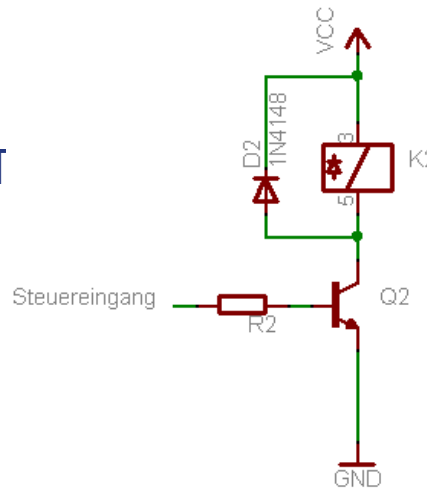
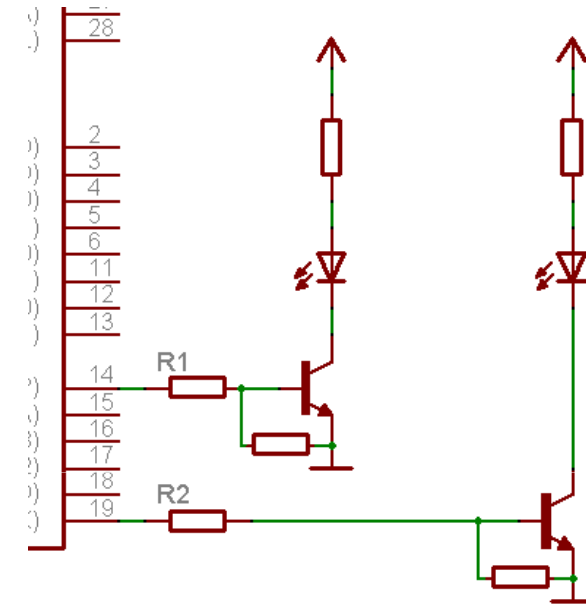


- Für die ersten Versuche braucht man nur ein paar Taster und LEDs an die IO-Ports des AVR anzuschließen. An PB0-PB5 schließt man 6 LEDs über einen Vorwiderstand von je 1 k Ω gegen Vcc (5V) an. In der Praxis ist es unerheblich, ob der Widerstand vor oder nach der Diode liegt, wichtig ist nur, dass er da ist.
- Taster mit je einem 10 k Ω Pullup-Widerstand
- Taster mit internen Pullup



I/O für mehr Strom

- Man kann nicht jeden beliebigen Verbraucher nach dem LED-Vorbild an einen μC anschließen. Die Ausgänge des ATmega8 können nur eine begrenzte Menge Strom liefern, so dass der Chip schnell überfordert ist, wenn eine nachgeschaltete Schaltung mehr Strom benötigt. Die Ausgangstreiber des μC würden in solchen Fällen den Dienst quittieren und durchbrennen.
- Z.B. bei kräftigen Relais, PTT Schaltung,...



ATMEGA was geht noch

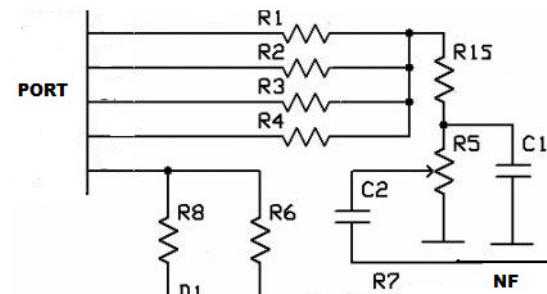
- Die Aussage – alles geht ist so nicht richtig und zu unspezifisch!
- Wir können:

Messen

- digitale Eingänge
- analoge Eingänge (positiv/negativ Komparator)
- D/A Wandler

Steuern

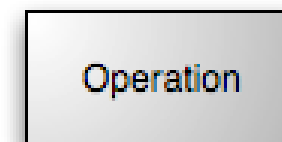
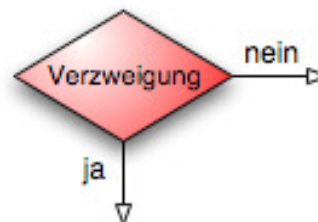
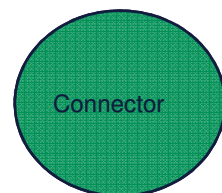
- digitale Ausgänge
- PWM Ausgang (Pulsweitenmodulation für Dimmen, Drehzahl Motor,...)
- analog mit Widerständen, R2R Netz



ATMEGA Programmieren



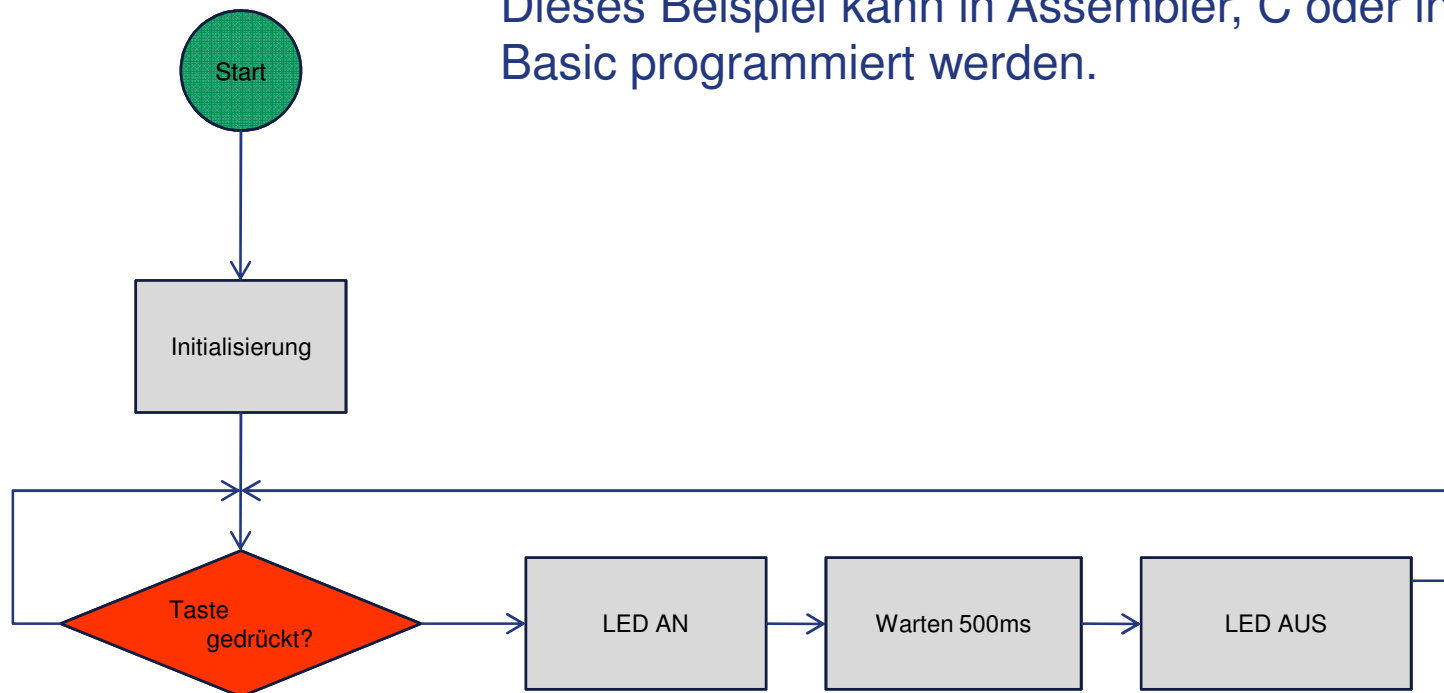
- Der ATMEGA wird hauptsächlich in Assembler, C oder Basic programmiert.
- Wir werden hier zuerst unabhängig von Sprachen unter Verwendung eines minimalistischen PAP (Programm Ablauf Plan) programmieren lernen.
- Ein Programmablaufplan (PAP) ist ein Ablaufdiagramm für ein Computerprogramm, das auch als Flussdiagramm (engl. flowchart) oder Programmstrukturplan bezeichnet wird. Es ist eine graphische Darstellung zur Umsetzung eines Algorithmus in einem Programm und beschreibt die Folge von Operationen zur Lösung einer Aufgabe.
- Die Symbole für Programmablaufpläne sind in der DIN 66001 genormt.
- Wir werden nur verwenden:



ATMEGA Programmieren

- Beispiel: Taste drücken, LED blinkt, Taste loslassen LED aus

Dieses Beispiel kann in Assembler, C oder in Basic programmiert werden.



ATMEGA Programmierung

Ablaufsteuerung PA und VV



ÜBUNG

PTT wird gedrückt:

VV aus, PA an, TRX senden (jeweils 100ms Schaltzeit berücksichtigen).

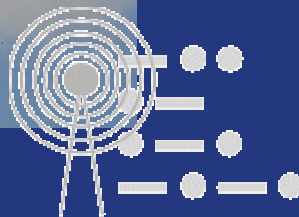
PTT wird losgelassen:

PA aus, TRX auf Empfang, VV an

Bitte erstellt einen PAP.

VIEL ERFOLG!

ATMEGA – Mikrocontroller Einführung TEIL 2

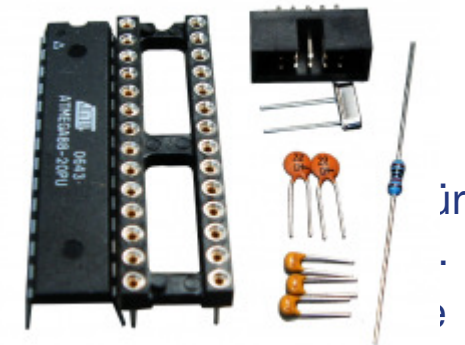


DH5HS
9. November 2009
Mönchengladbach

ATMEGA Programmierung HARDWARE – was wird benötigt

Das Startset:

- Das ATmega8 Controller Set besteht aus einem ATmega8-16PU passendem IC Sockel mit gedrehten Kontakten, 16MHz Quarz, 3x 100nF Keramikkondensatoren, 2x 22pF Keramikkondensatoren, einen Reset PullUp und einer 10 poligen Stiftleiste für den Aufbau
- Damit enthält das Set alles wichtige um die Grundschialtung auf dem Schaltplan erklärt den Aufbau
- ATmega8-16PU Mikrocontroller
- 28 poliger IC Sockel mit gedrehten Kontakten
- 16MHz Quarz
- 3x 100nF Kondensatoren
- 2x 22pF Keramik Kondensator
- 10 poliger Wannenstecker für ISP
- 10k Ohm Widerstand für Reset PullUp



[C] embedit.de

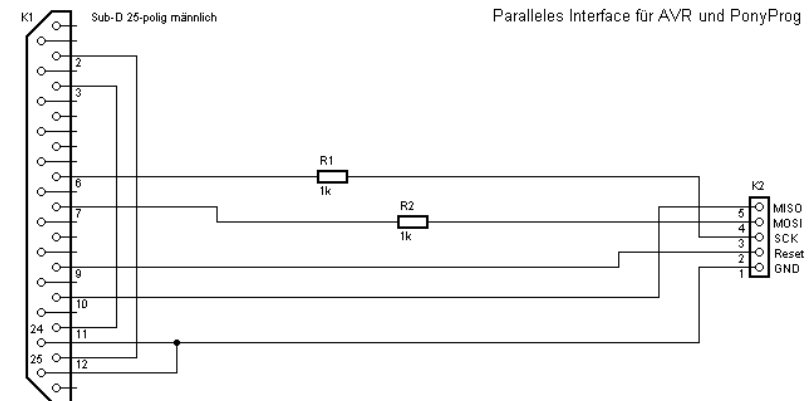
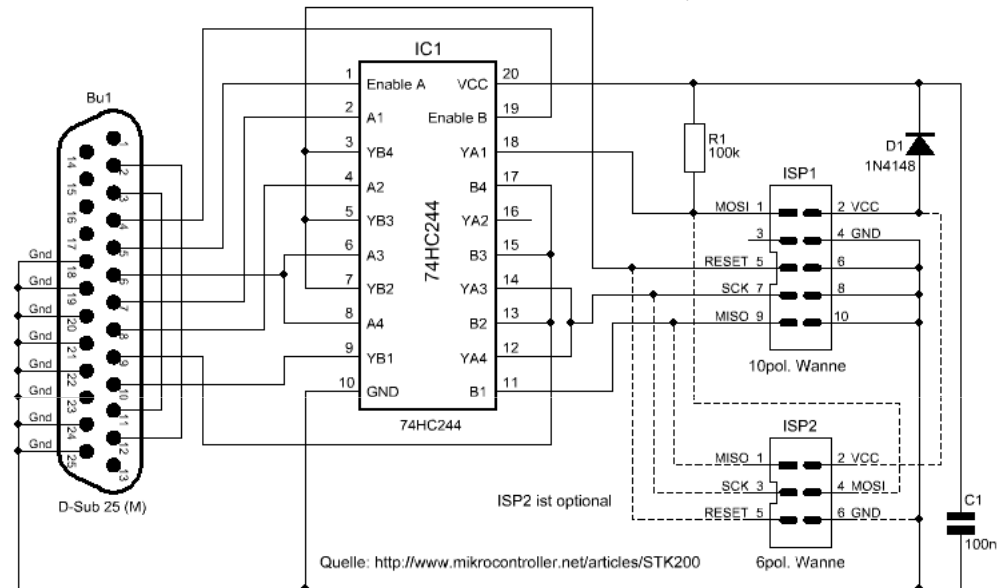
http://shop.embedit.de/product__92.php

ATMEGA Programmierung HARDWARE – was wird benötigt

Der Programmer:

- Gepufferte Version
- Ungepufferte Version
 - Das nebenstehende Bild zeigt die Schaltungsversion für den parallelen Port. Hier werden neben den erforderlichen Steckverbindern nur 2 Widerstände benötigt. Diese sind notwendig, um die Ausgänge des LPT-Port sowie die Programmieranschlüsse des angeschlossenen Controllers zu schützen. Der Vorteil dieser Version ist die einfache Schaltung. Nachteilig ist allerdings, dass LPT-Ports beim PC nicht besonders gut geschützt sind und man sehr vorsichtig arbeiten muss.

STK200 Programmer (mit 74HC244) <http://halvar.at/>



ATMEGA Programmierung Software & Anleitung



Wir werden in BASCOM, einem Basic Dialekt Programme schreiben. Ein sehr schöne Anleitung mit Code Schnipseln findet sich hier :

http://halvar.at/elektronik/kleiner_bascom_avr_kurs/