

Wie Software laufen lernt

Übersicht

1. Einleitung
2. Grundkenntnisse
3. Ausstattung
4. Wo fange ich an?

Grundsichtplan

Steckbrett, AVR-Studio und Datenblätter

Debugging: Oscilloscope und Seriellport

Tutorials

5. Besonderheiten bei der uC Programmierung
6. Bezugsquellen, Linksammlung

Einleitung

Wozu kann man Microcontroller einsetzen:

ServoControl: Ansteuerung eines Monitors und von Servomotoren

Quadcopter: Ansteuerung der Rotoren, Sensorauswertung

Hexapod: Steuerung der Servos, Umwandlung der Funksignale

ServoControl

ServoControl:

Ansteuerung von Servos vom PC aus, mit Hilfe der 3D-
Animationssoftware Maya

Mehrere Servos mit einem Controller

Aufnahme der Bewegungsabläufe

Schnelles Setup

ServoControl

Erste Schritte:

Aufbau einer einfachen Schaltung um einen Servo mit einem Poti zu steuern

Ersetzen des Potis durch einen Digipoti

Senden von Positionen über den Parallelport

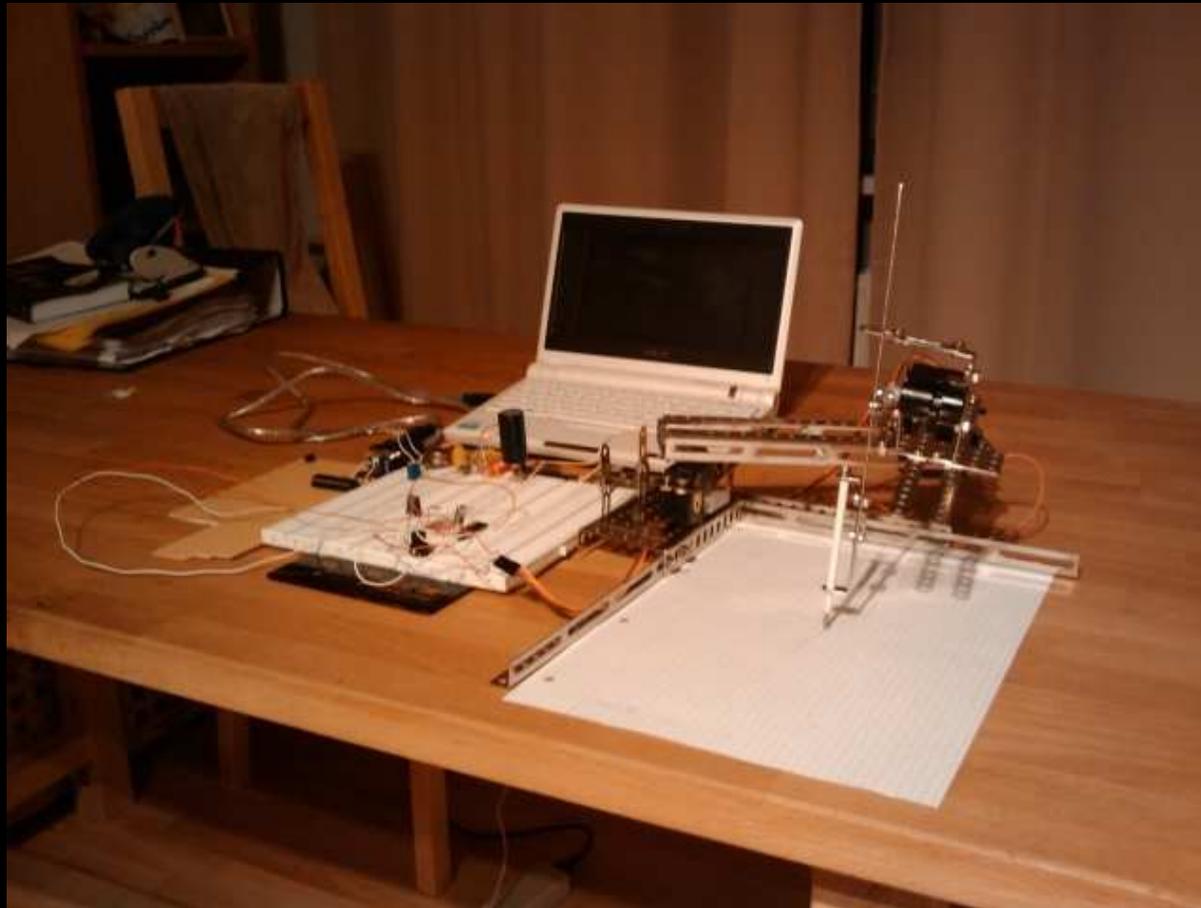
Parallelport durch Seriellport ersetzen

ServoControl



ServoControl

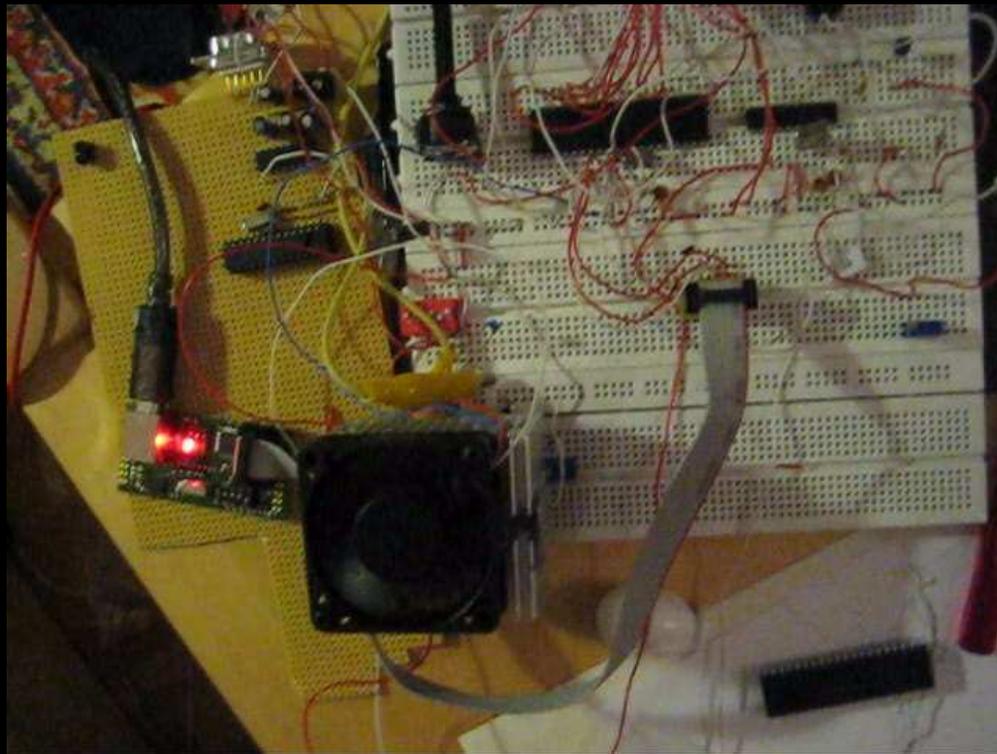
Aufbau eines Testarms mit mehreren Servos



ServoControl

Einbau Bildschirm

Hinzufügen von selbstgebauten Joysticks



ServoControl

Was man mit der Servocontrol alles anstellen kann könnt ihr nachher selbst probieren

Quadrocopter

Precise Aggressive Maneuvers for Autonomous Quadrotors

**Daniel Mellinger, Nathan Michael, Vijay Kumar
GRASP Lab, University of Pennsylvania**

Hexapod

A-Pod part II

Ant inspired hexapod robot

made by

Kåre Halvorsen
aka Zenta

Grundkenntnisse

Inhalte TI – I

Spannung, Strom, Erdung, Stromkreis

Kondensatoren, Widerstand, Transistoren, Dioden

Programmierung:

C und Grundlagen in Assembler (TI-II MMIX reicht)

Die Werkstatt



Ausstattung

Dringend benötigt

Lötstation

Steckbrett

Programmer

Voltmeter

Bauteile (siehe Einkaufsliste)

Praktisch

STK500

Oszilloskop – am besten digital (aber teuer)

Abisolierzange

Einkaufliste

Micrcontroller:	ATMEGA 8 oder 32
Widerstände:	470 Ohm, 4,7 kOhm – in guten Stückzahlen (>50)
Kondesatoren:	22pF, 100nF, 1uF (>20)
Klingeldraht :	am besten in verschiedenen Farben
Transistoren:	nnp(BC547), pnp(BC557)
Dioden:	1n4148
Quarze:	16 MHz, 14,4764 MHz
VCC:	7805 (Spannungsregler auf 5V)
Seriell Buchse:	D-Sub9
Spannungswandler:	Max232
Netzteil:	Altes ausm Keller (>7V <12V) / Rechnernetzteil

Icespray, Leuchtdioden, Schalter, Piezzotaster, USB-Seriellwandler

Cola, Energydrink, Junk-Food

Abisolierzange



Spart unglaublich Zeit, Nerven und Kabel

Preis: ca.: 15,00 €

Stückzahl: 1

Lötbesteck



Dritte Hand / Helfende Hand

Lötstation: 400° , mindesten 75 W

Lötzinn: verbleit, mit Seele

alten PC Ventilator zum Lötdampf wegpusten

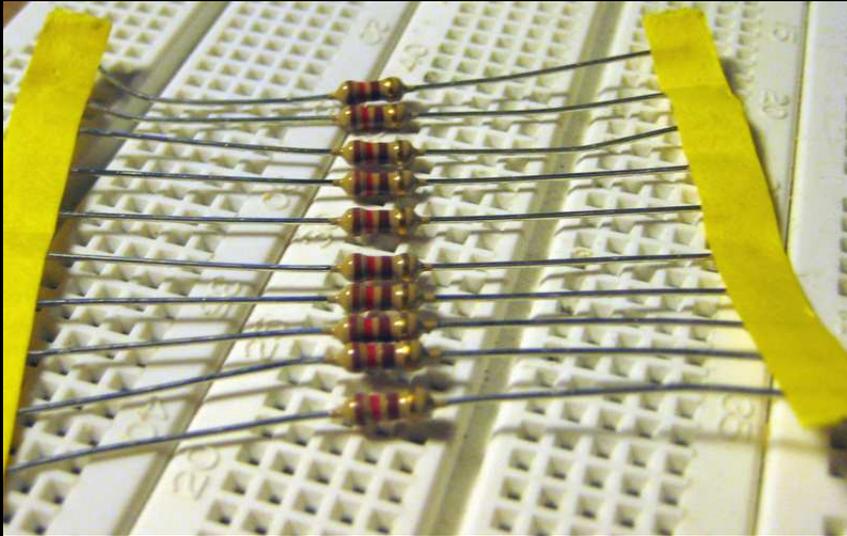


Lötstation: 40,00 € aufwärts

Lötzinn: 2,00 €

Helfende Hand: 2,45 €

Widerstände



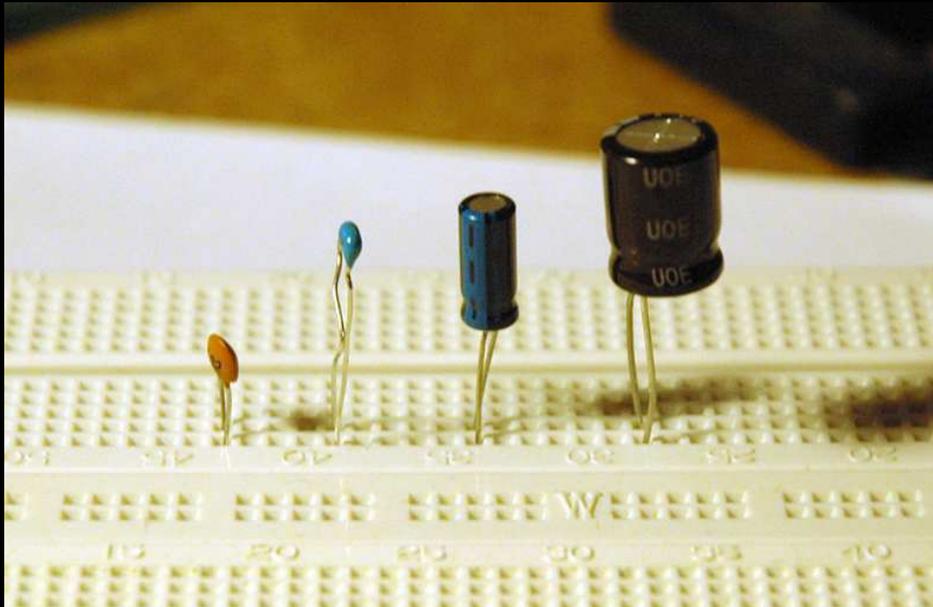
Einzel:	10 cent
Ab 10 Stk.:	3 cent
Ab 100 Stk.:	1,9 cent
Ab 1000 Stk	0,5 cent

470 Ohm 4,7 kOhm

Preis: ca.: siehe oben

Stückzahl: 10-50

Kondensatoren



Puffern, Filtern, RC-Glied, Max232
Beschaltung

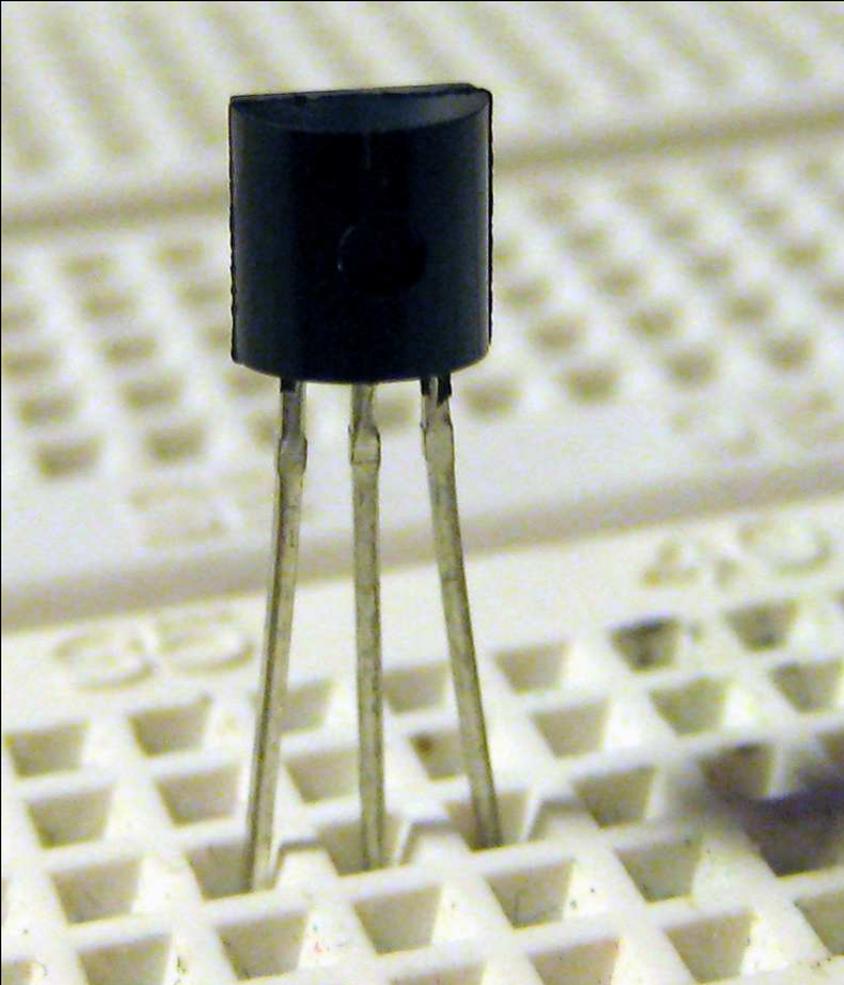
22pF, 100nF, 1uF

Preis: ca.: 0,06 €

Stückzahl: 10

Transistoren

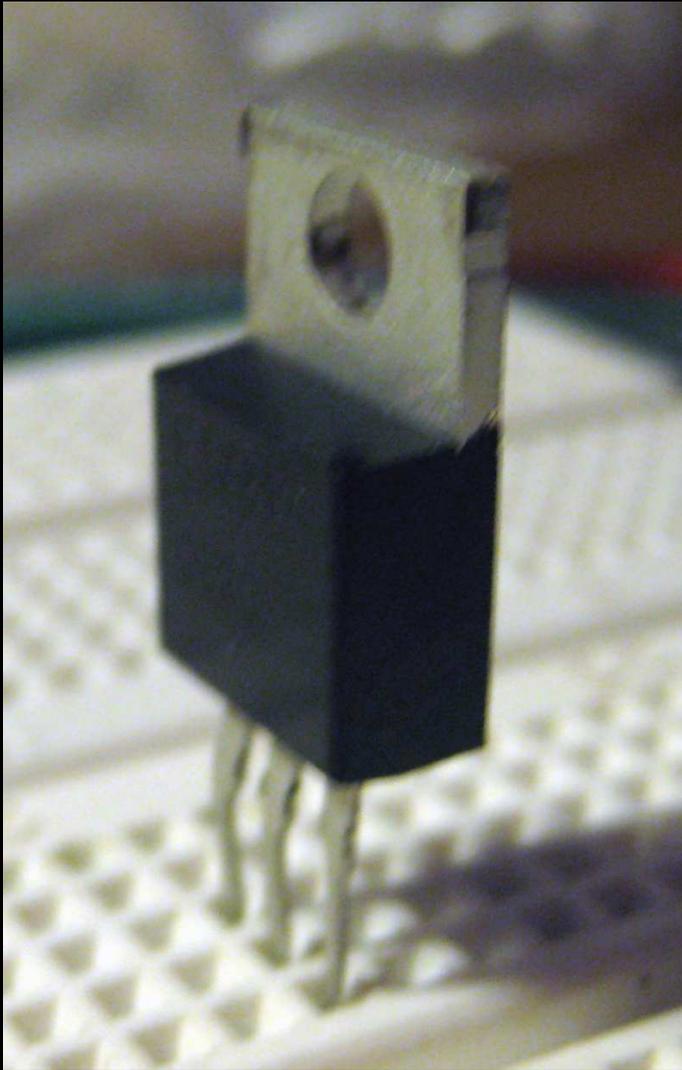
Braucht man für Logikschaltungen



Preis: ca.: 0,04 €

Stückzahl: 15

7805



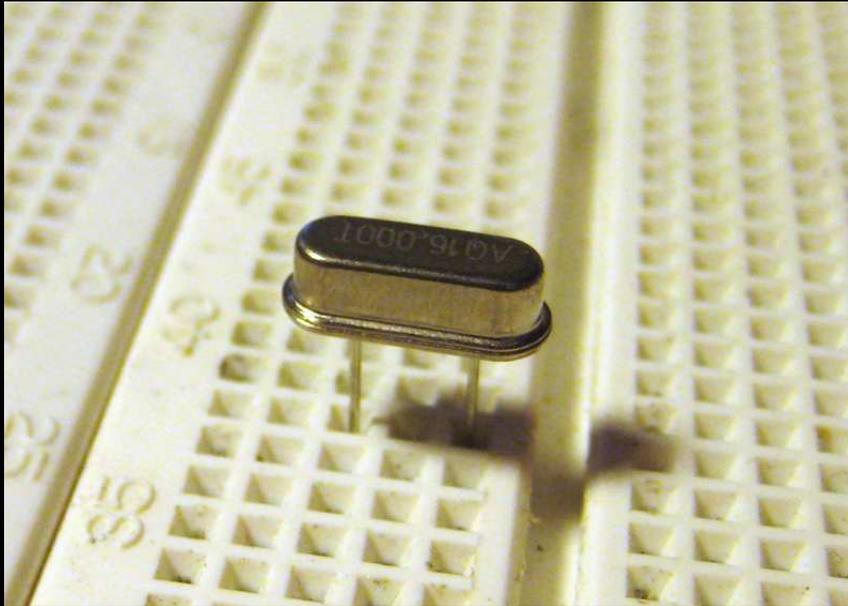
Hält Spannung auf 5V

Input 7 bis 12 V

Preis: ca.: 0,25 €

Stückzahl: 5

Quarz

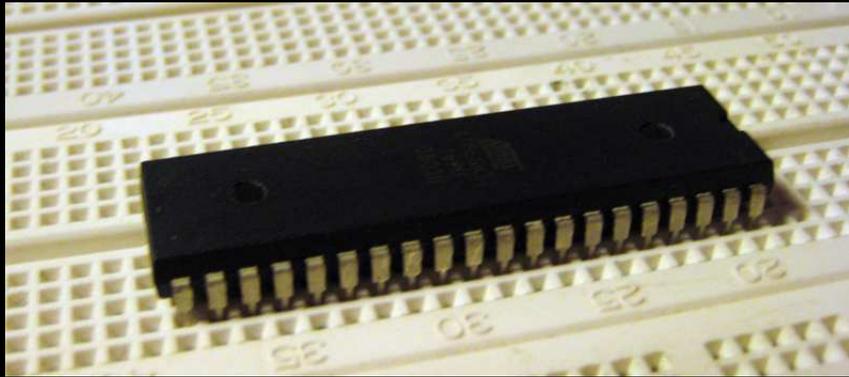


Genauer Takt für Microcontroller
Vorallem wichtig bei RS232

Preis: ca.: 0,20 €

Stückzahl: 1-3

Microcontroller

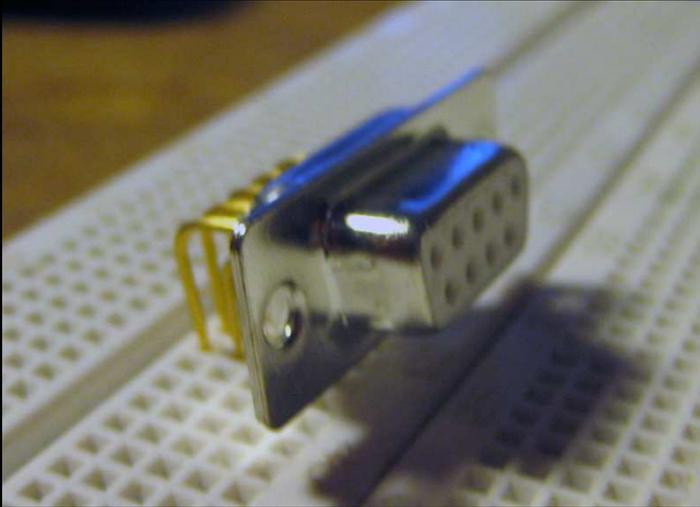


Atmega32, oder Atmega8 (Auslaufmodelle
aber im Netz gut dokumentiert)

Atmega8515 (Neue Pinbelegung)

Atmega8:	Preis: ca.: 2,40 €	Stückzahl: 5
Atmega8515:	Preis: ca.: 2,70 €	Stückzahl: 5
Atmega32:	Preis: ca.: 3,80 €	Stückzahl: 5

RS232

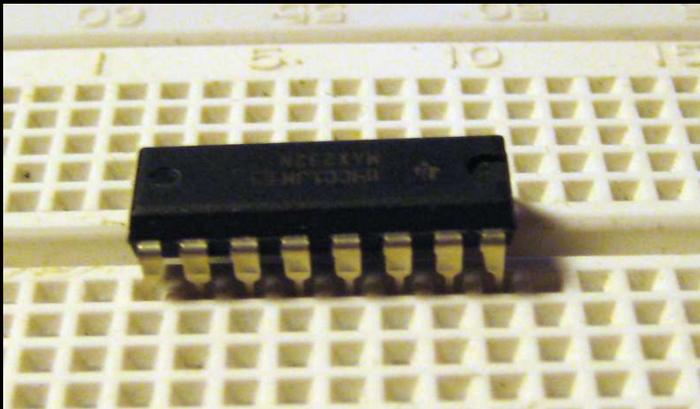


RS-232 Buchse

Braucht man wenn man ohne das STK500 arbeitet zum Datenaustausch über Seriellbuchse

Preis: ca.: 0,50 €

Stückzahl: 1



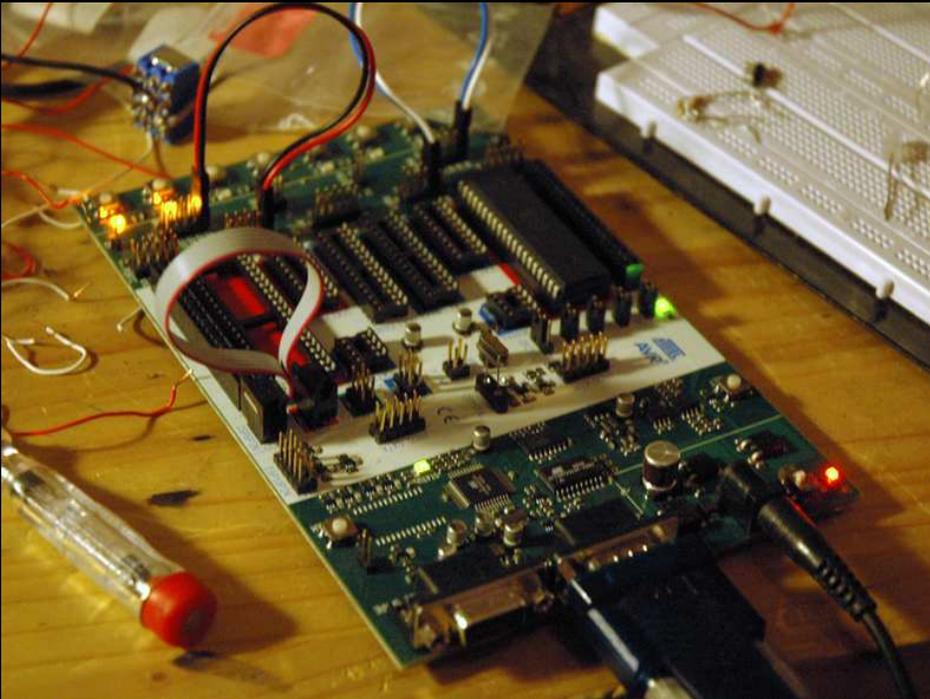
Max-232

Signalpegelwandler von 16V auf 5 V und zurück

Preis: ca.: 0,40 €

Stückzahl: 2-4

STK500



Developerboard von Atmel,
Hat Programmer, Taster, Lampen,
Seriellanschluss.

Unterstützt alle Atmel RISC Controller
Kann auch nur als Programmer für
Steckbrett benutzt werden

**Großer Vorteil: Man weiß das die
Schaltung richtig ist!**

Bei fast allen neuen Rechnern: USB-Seriellwandler: 5 €

Preis normal: ca. 80 €, im education-shop von embedded-projects ca. 40 €

Hello World

Page 1

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

ISR(INT0_vect) {
    PORTB ^= 0x02;
}

int main() {
    cli(); // Disable all interrupts
    DDRB=0xFF; // Port B is output
    DDRD=0x00; // Port D is input

    MCUCR |= (1<<ISC01) | (1<<ISC00); // Interrupt mode: any change
    GICR |= (1<<INT0); // Activate interrupt
    sei(); // Allow interrupts

    while(1) {
        PORTB ^= 0x01;
        _delay_ms(500);
    }
}
```

Datenblätter

Debugging

Keine Textausgabe keinen Debugger (nur AVR Simulation)

Debuggen nur über Spannungswechsel an den Pins

Leds kann man generelle Statuscodes ausgeben

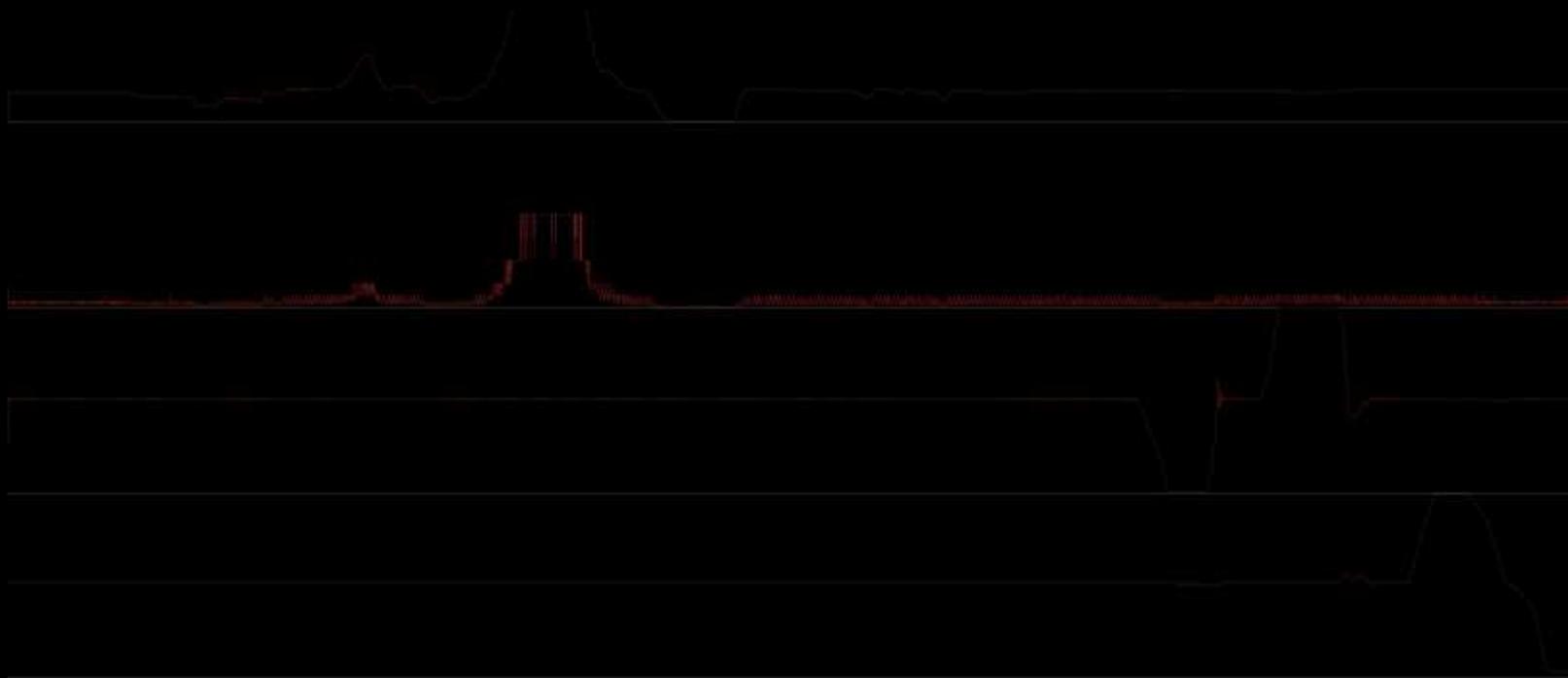
Mit einem Oszilloskop kann man auch Flanken anschauen und Zeiten messen kann

Über Seriell kann man Text an den PC schicken – das verändert aber das Timing da der UC ziemlich lange braucht um die Zeichen zu schicken

Selbst lese Programme schreiben – grafische Darstellung von seriell werten

Debugging

Ausgabe seriell Werte als Graph



Besonderheiten

Fehler sind nicht mehr nur in der Programmierung zu suchen.

Sie treten mindestens genauso häufig in der Schaltung auf.

Externe Störquellen

Funkstrahlung: Lötkolben, Boxen, Oszi, Funkmäuse, Handys

Kaputte Hardware:

Kabel austauschen

uCs brennen gerne mal durch oder haben kaputte Pins – durchmessen

Im Zweifelsfalle – Teile Austauschen.

Wiki hat nicht immer Recht!

Im Zweifel lieber Datenblätter lesen im Netz kursieren leider sehr viele Halbwahrheiten.

Tutorials

Generelle Einleitung:

<http://www.mikrocontroller.net/articles/AVR-Tutorial>

C Programmierung – Komplett Übersicht

<http://www.mikrocontroller.net/articles/AVR-GCC-Tutorial>

Wichtigsten Tutorials:

Erzeugen von Maschinencode

Programmieren mit Interrupts (14)

Timer (16)

UART (17) (Serielle Kommunikation)

Assembler Programmierung

<http://www.mikrocontroller.net/articles/AVR-Tutorial>

Linkliste

Projektblog:

<http://hackaday.com/>

Shops:

<http://www.reichelt.de> (Einzelteile, Werkzeug)

<http://www.eproo.net/> (Edu Rabatt – STK500)

<http://www.nodna.com/> (fertige Boards für Servo, Motoren etc)

<http://www.roundsolutions.com/> (fertige Board GSM, GPS etc.)

<http://www.pcb-pool.com/ppde/index.html> (Platinen ätzen)

Foren:

<http://www.mikrocontroller.net/>

<http://www.avrfreaks.net/>

<http://www.roboternetz.de/>