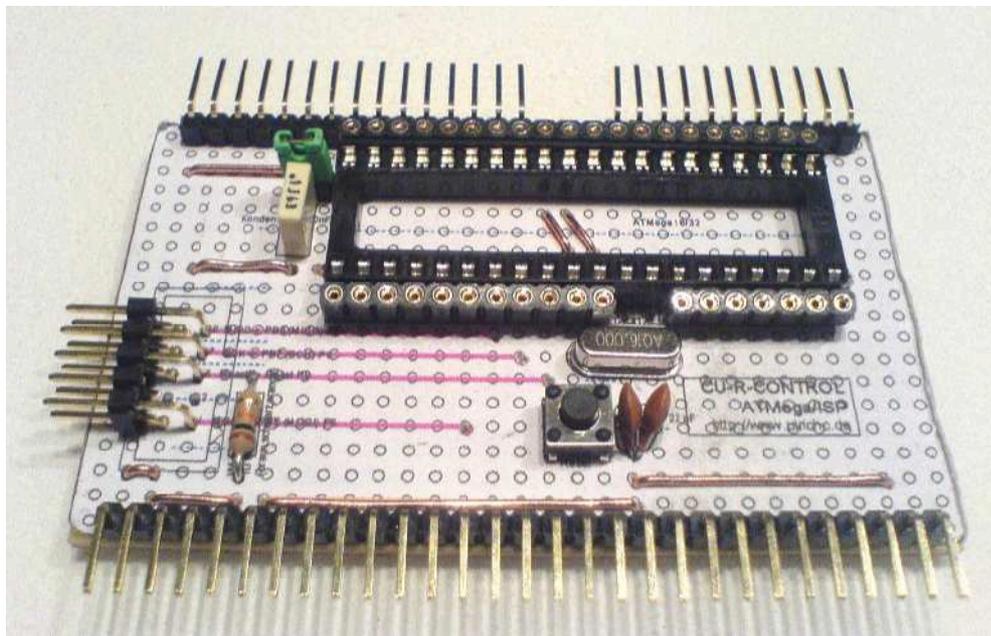


## CU-R-CONTROL

Beschreibung zur Schaltung **ATMega16-32+ISP**  
*MC-Controller Steuerung auf Basis ATMEL Mega16/32*



Autor: Christian Ulrich

Datum: 08.12.2007

Version: 1.00

## Inhalt

Historie	3
Einleitung	4
Beschreibung	5
Funktionen der Schaltung	5
Schaltplan	6
Einstellungen zur Schaltung	7
<b>Hardware</b>	<b>7</b>
Jumper J1	7
<b>Software</b>	<b>8</b>
Fuses für Quarz	8
Fuses für JTAG	8
Programmierung	8
Tipps und Hinweise	9
<b>Controller sockeln</b>	<b>9</b>
<b>Takt optimieren</b>	<b>9</b>

## Historie

Erstellt 08.12.2007 von Christian Ulrich

## Einleitung

Dies ist die Beschreibung zu einer Schaltung des CU-R-CONTROLS. Bei CU-R-CONTROL handelt es sich um einen modularisierter Schaltungsaufbau der es ermöglicht komplexe Steuerungen zu konfektionieren. Weiter Informationen und Schaltungen zum CU-R-CONTROL finden Sie bei <http://www.ulrichc.de/>.

## Beschreibung

Die Schaltung ATMega16/32 Controllerschaltung zum CU-R-CONTROL entspricht einer ATMega - Mikrokontroller- Grundschialtung mit den entsprechenden Anschlüssen zur Programmierung und Nutzung des Controllers.

## Funktionen der Schaltung

### - Freie Wahl aus ATMega-Serie.

Der Mikrokontroller, kann je nach Anforderungen aus der 40 Pin ATMega Kontroller-Serie von Atmel gewählt werden. Passend zu der in CU-R-CONTROL vorgesehene Controllerserie, im DIP40 Gehäuse, sind (stand 2007) Mega164, Mega324, Mega644 und Mega1284 kompatibel.

*Im Falle ATMega32 stehen 32K Speicher, 2K RAM und 1K EEPROM sowie 32 programmierbare I/O Pins inklusive 8 AD Ports zur freien Verfügung.*

### - Externer Oszillator

Ein vorgesehener Quarz, kann die Geschwindigkeit des Controllers auf ca. 16 Mhz steigern.

*Hinweis: Falls der Quarz nicht eingelötet wird, taktet der Prozessor intern mit ca. mit 1 Mhz.*

### - Externe Portleitungen

Alle nutzbaren Ports des Controllers, sind über Buchsenleiste erreichbar. So können alle Ports via Draht, ohne zwingendes Löten, genutzt werden. Die Buchsen eignen sich zudem gut zum Ansetzen von Prüfspitzen zur Messdatenerfassung oder auch Diagnosen bei der Fehlersuche.

### - Programmierschnittstelle

Die ISP- Programmierschnittstelle des ATMega-Controllers kann mit gewöhnlichem AVR- Programmieradapter (10polig) programmiert werden.

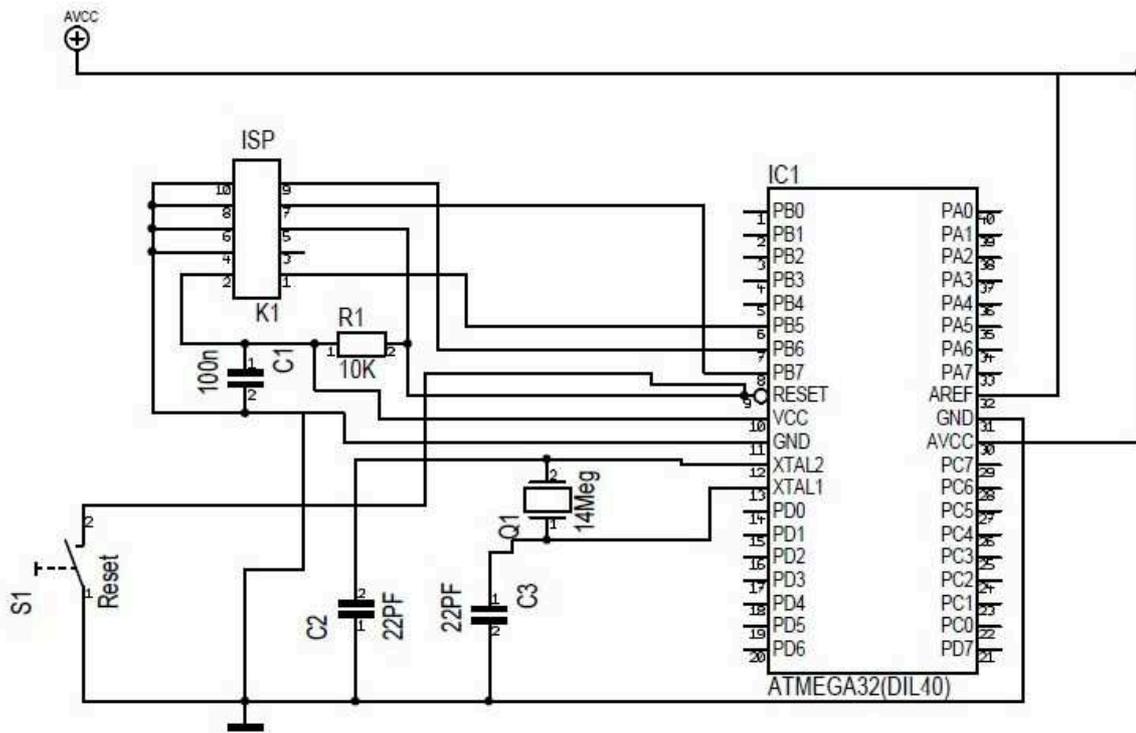
Hinweis: Zum Programmieren des Controllers, wird ein Programmieradapter (ISP- Programmer) benötigt.

### - Resettaster

Ein Reset des Controllers, kann bequem über Taster auf der Platine ausgelöst werden.

## Schaltplan

Die unterhalb abgebildete Schaltskizze entspricht dem Aufbau der Schaltung.



## Einstellungen zur Schaltung

### *Hardware*

**Jumper J1** : Wenn der Jumper gesetzt ist, hat der Controller und alle als „5V“ eingezeichneten PINs Strom. Andernfalls, wird der Controller nicht genutzt. In diesem Fall funktioniert lediglich der ISP-Anschluß , der dann in Verbindung mit einer anderen Controllerplatine dieser Art genutzt werden kann.

## **Software**

### **Fuses für Quarz**

Standardmäßig, nutzt der Controller lediglich den internen Oszillator (1Mhz) zur Taktung. Falls der extern vorgesehene Quarz der Schaltung genutzt werden soll, muss dies über die Einstellung der so genannten Fuses erfolgen.

### **Fuses für JTAG**

Standardmäßig, ist der JTAG des Controller aktiviert. Falls man keinen JTAG benötigt, sollte man diesen via Fuses deaktivieren. Auf diese Weise , werden PortC.2 und PortC.5 frei und nutzbar.

## **Programmierung**

Der verwendete AVR-Controller kann in vielen Sprachen programmiert werden.

Unter anderem in:

- C
- C++
- Basic (auch BASCOM)
- Pascal
- ASM (Assembler)

Die Wahl der Programmiersprache hat für gewöhnlich keinen direkten Einfluss auf das Laufzeitverhalten des Controllers. Sodass die Sprache frei nach individuellen Bedürfnissen gewählt werden kann.

## Tipps und Hinweise

### ***Controller sockeln***

Der Controller IC muss gesockelt werden. Die Controller sind im Allgemeinen sehr empfindlich. Zudem müssen diese, gerade beim Experimentieren, öfter mal ausgetauscht werden.

### ***Takt optimieren***

Der Takt des Controllers kann seiner Aufgabe angepasst werden.

Je kleiner der Takt, desto geringer der Stromverbrauch des Controllers.

Bei RS232 Verbindungen, sind ungerade Taktungen geeignet. Gerade Taktfrequenzen (4, 8, 16 Mhz) führen dazu dass vereinzelt Bytes nicht Empfang werden können.

Folgende Quarze empfehlen sich falls mit RS232 gearbeitet werden soll.

#### **U(S)ART-Quarze**

1,8432MHz,  
3,6864MHz,  
7,3728MHz,  
14,7456MHz,  
18,4320MHz

Dieses Dokument gehört zum Projekt [CU-R-CONTROL](#) von UlrichC.DE. Weitere Informationen, Dokumente sowie Bilder zum Projekt sind auf der Internetpräsenz <http://www.ulrichc.de/> zum Download bereitgestellt.