

Produktinformationen

Basis Set Atmel Atmega8 & ISP Programmer

Lieferumfang :

- 1 Steckboard 840 Pin
- 1 USB ISP Programmer mit Kabel
- 1 Kabelsatz 32-teilig
- 1 Atmega8L
- 5 LED 3mm Klar Rot/Gelb/Grün/Blau/Weiss
- 3 Widerstände 330Ohm
- 2 Widerstände 220Ohm
- 2 Widerstände 1K
- 2 Widerstände 10K
- 2 Widerstände 47K
- 2 Kondensatoren 100nF
- 2 Dioden 1N4148
- 1 Transistor NPN
- 1 Transistor PNP
- 2 Mikro Taster
- 1 Ponti 10KOhm (veränderbarer Widerstand)
- 1 Batteriehalter AAA (ohne Batterie)
- 1 CD mit Software / Tools / Links / Erste Schritte / Treiber

Auf der CD :

AVR_PACK\	Software und Anleitungen
AVR_PACK\AVR_IMG\	Bilder
AVR_PACK\AVR_PRO\	Erste Programme
AVR_PACK\DATASHEET\	Datenblätter
AVR_PACK\USBISP_Driver\	Treiber zum USB Programmer
AVR_PACK\USBISP_FIRMWARE\	Bootloader des Programmer

Der Atmega8L

Der Atmega8L ist ein 8-Bit-Mikrocontroller des US-amerikanischen Herstellers Atmel. Die Controller dieser Familie sind wegen ihres einfachen Aufbaus und ihrer leichten Programmierbarkeit auch bei Hobby-Anwendern weit verbreitet.

Wichtige Eckdaten :

Spannungsbereich 2.7 – 5.5 Volt

Geschwindigkeit intern 1 MHz, mit externen Quarz bis 8 MHz

programmierbarer Flash-Speicher 8k (10.000 Schreibzyklen)

flüchtiger SRAM 1k

nicht-flüchtiger EEPROM 512byte (100.000 Schreibzyklen)

Belastung pro Pin: 20mA

Belastung pro Port : 100mA

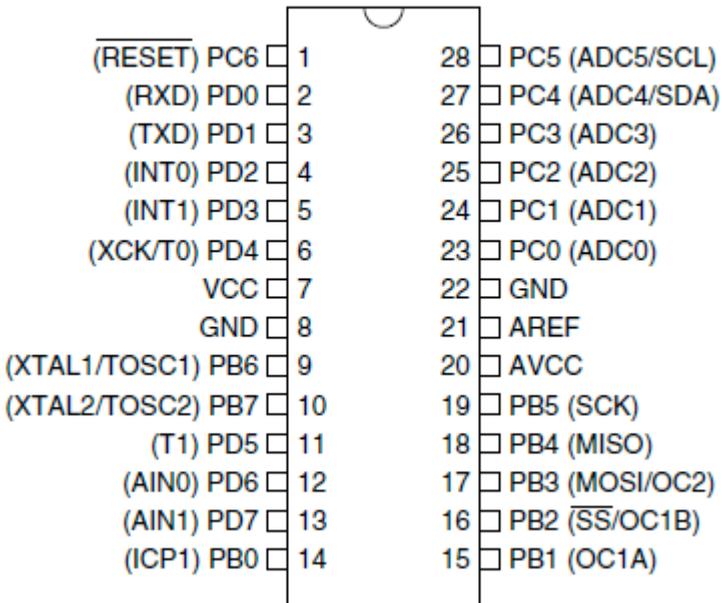
Belastung Total : 400mA

Achtung, das sind allgemeine

Eckdaten. Detailliert auf Seite

237+238 im Datenblatt

PDIP



Der USB Programmer

Bei diesen Programmer handelt es sich um ein OpenSource-Projekts USBasp (USB basiertes Programmiergerät für Atmel AVR Mikrocontroller, siehe <http://www.fischl.de/usbasp>).

Der gelieferte Programmer ist leicht erweitert mit einer 500mA Sicherung für den Schutz des USB Anschlusses. Weiter ist eine Spannungs Weitergabe eingebaut. So lässt sich ein Versuch über den USB Port mit 5 Volt versorgen, oder mit einer externen Batterie oder Netzgerät.



Am besten entfernen Sie diese Brücke für die erste Versuche.

Die zweite Brücke ist nur nötig um den USB Programmer selbst neu zu Flashen.

Liste der unterstützten Mikrocontroller

51Series:

AT89S51, AT89S52, AT89S53, AT89S8252

AVR Series:

ATTiny12 (L), ATTiny13 (V), ATTiny15 (L), ATTiny24 (V), ATTiny25 (V), ATTiny26 (L), ATTiny2313 (V), ATTiny44 (V), ATTiny45 (V), ATTiny84 (V), ATTiny85 (V), AT90S2313 (L), AT90S2323 (L), AT90S2343 (L), AT90S1200 (L), AT90S8515 (L), AT90S8535 (L), ATMEGA48 (V), ATMEGA8 (L), ATMEGA88 (V), ATMEGA8515 (L), ATMEGA8535 (L), ATMEGA16 (L), ATMEGA162 (V), ATMEGA163 (L), ATMEGA164 (V), ATMEGA165 (V), ATMEGA168 (V), ATMEGA169 (V), ATMEGA169P (V), ATMEGA32 (L), ATMEGA324 (V), ATMEGA325 (V), ATMEGA3250 (V), ATMEGA329 (V), ATMEGA3290 (V), ATMEGA64 (L), ATMEGA640 (V), ATMEGA644 (V), ATMEGA645 (V), ATMEGA6450 (V), ATMEGA649 (V), ATMEGA6490 (V), ATMEGA128 (L), ATMEGA1280 (V), ATMEGA1281 (V), ATMEGA2560 (V), ATMEGA2561 (V), AT90CAN32, AT90CAN64, AT90CAN128, AT90PWM2 (B) , AT90PWM3 (B)

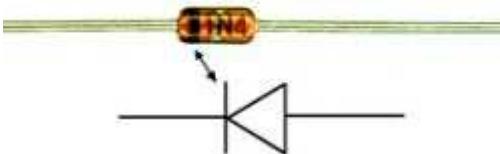
Die Bauteile

Der Widerstand stellt einen elektrischen Widerstand für den Strom dar. Mit folgende Tabelle lässt sich der Wert bestimmen.

Deutsch

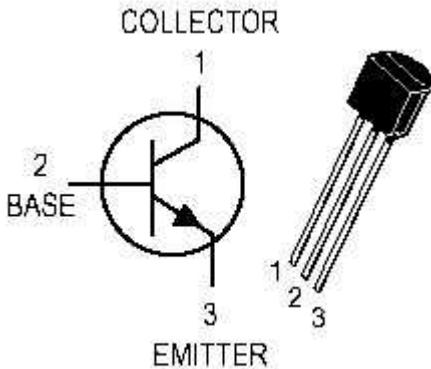
Widerstände mit 4 Farbringen (Din41429)

	1. Ring	2. Ring	3. Ring	4. Ring
	Wert A	Wert B	Multiplikator	Toleranz
Silber			0,01	10 %
Gold			0,1	5 %
Schwarz		0	1	
Braun	1	1	10	1 %
Rot	2	2	100	2 %
Orange	3	3	1.000	
Gelb	4	4	10.000	
Grün	5	5	100.000	0,5 %
Blau	6	6	1.000.000	0,25 %
Lila	7	7	10.000.000	0,1%
Grau	8	8		0,05%
Weiß	9	9		



Die Diode Typ 1N4148 lässt den Strom nur in einer Richtung fließen. Hier ist die Polung wichtig.

Ein Transistor ist ein elektronisches Bauelement zum Schalten und Verstärken von elektrischen Signalen. Mit dem Typ BC547A (NPN) und BC557A (PNP) verstärken Sie bis 0.5 Watt (zB. ein Relais)



Die Leuchtdioden mit 20mA sind optimal für Modell oder Anzeigen. Die LED immer mit dem passenden Vorwiderstand betreiben. Benutzen Sie die blaue und weisse LED mit 220 Ohm und die restlichen mit 330 Ohm. So ist die Strombelastung des Atmega8 optimal. Die LED muss richtig gepolt werden.



Der Keramikkondensator nutzen wir um die Spannung am Atmega8 möglichst zu stabilisieren. D.h. Hochfrequente Spannungsspitzen abzufangen. Die Polung ist egal.

I/O Pin und C Programmierung

Gemäss Atmega8 Schema haben wir 3 Data Direction Register. Wir bezeichnen Sie DDRB , DDRC und DDRD. Jede der 3 Register hat wiederum 8 Pin mit den Bezeichnung PB0, PB1 usw. Nun können wir die I/O Pins wahlweise als Ausgang oder Eingang nutzen.

Dies funktioniert folgenderweise zB. PB0 :

```
DDRB = (1<<PB0); Ausgang ( dem Register eine 1 zuweisen)
DDRB &= ~(1<<PB0); Eingang ( dem Register eine 0 zuweisen)
```

Wenn jetzt eine LED leuchten soll setzen wir den Pin auf Ausgang und danach:

```
PORTB = (1<<PB0); PIN High (es liegen 5 Volt an)
PORTB &= ~(1<<PB0); PIN Low (es liegen 0 Volt also GND an)
```

Nun der Ablauf eines Taster nachdem wir den Pin auf Eingang setzen :

```
PORTB |= (1<<PB0); Interne Pull-Up einschalten
if(!(PINB & (1<<PB0))) Mit dieser Abfrage warten wir bis der Pin
auf GND gezogen wird um danach eine
Aktion zu Starten. (Ampel)
```

Allgemeiner Ablauf eines Mikrokontroller Programm

Mit **Include Dateien** werden dem Compiler verschiedene Informationen oder mehr Funktionsumfang bereitgestellt. Diese erlaubt uns fertige Funktionen einzubinden. Die wichtigste ist:

```
#include <avr/io.h>
```

Die io.h gibt uns Zugriff auf einfache Konstrukte zu verschiedene Funktionen und Variablen des Atmega8. Wie zB. DDRB, PINB usw. Mit der Delay.h

```
#include <util/delay.h>
```

erhalten wir Funktionen auf Wartefunktionen in Milli – oder Mikrosekunden.

Mit **Defines** geben wir Pins, Zahlen oder Einstellungen einen Vernünftigen Namen. Hängt zB. eine rote LED am PIN5 ist es einfacher mit:

```
#define ROT_LED PIN5
```

Das Programm startet immer in der Main Funktion :

```
int main(void) {  
  
}
```

So würde der Atmega8 die Main einmal durchlaufen, wir wollen aber das er fortlaufend neue Befehle entgegennimmt. Aus diesem Grund brauchen wir eine Endlosschleife in der main.

```
While(1) {  
  
}
```

Übersetzt heisst das, wenn das in der Klammer wahr ist wiederhole. Die 1 gilt in solchen Abfragen immer als Wahr darum eine Endlosschleife. Genauso ist eine 0 immer Falsch, mit 0 würde die Schleife sofort verlassen.

Bit, Byte, Hex oder Dezimal

Rasch musste ich feststellen, das man sich mit Bits auseinander setzen muss um die vielen C Beispiele aus dem Internet zu Verstehen. Nicht verzweifeln an den vielen Schreibweisen.

0b00101000 = 0x48 = 72 = H

Die erste Zahl ist binär, indem man ,0b' voranstellt weiss der Compiler das auch. Das selbe in Hex mit ,0x' dann noch Dezimal aber was jetzt das ,H' ? Schauen Sie in eine ASCII Tabelle, das ,H' hat die Nummer 72 und für Textdarstellungen nicht unwichtig.

Mit dieser kurzen Einleitung und den 4 Experimenten haben Sie den Grundstock um eigene Ideen zu verwirklichen. Ein Buch würde nicht reichen um die ganzen Möglichkeiten eines Mikrokontrollers aufzuzeigen.

Auf der CD befindet sich eine Bit.html mit integrierten Javascript. Diese habe ich mir geschrieben für:

- Umrechnungen Dez, Hex und Bit
- einfacher Bitrechner
- Taktberechnungen mit Prescaler
- Anzeigen von Kondensator Codes, zB. 104 / 4n7 / 0.1 je nach Typ

Vielleicht ist Ihnen das Tool eine Hilfe.

Sicherheitshinweise

Der ISP-Programmer und Atmega8 werden geprüft und mit einer Test Firmware geflasht. Der ISP Programmer wird mit einer Gleichspannung von 5V aus dem USB betrieben und besitzt eine Überlastungsschutz. Wir können keine Schäden an Computer übernehmen. Wir empfehlen für erste Versuche ein alter PC oder Laptop mit Windows XP.

Das Modul ist nur für die Verwendung in trockenen Räumen gedacht, ein anderer Einsatz als angegeben ist nicht zulässig.

Ein Anschluss an das 220VAC-Hausnetz ist unzulässig, in diesem Fall wird das Modul zerstört, außerdem besteht Lebensgefahr!

Für alle Schäden, sei es Personen oder Sachschäden, die aus nicht bestimmungsgemäßer Verwendung resultieren, übernimmt der Betreiber die Verantwortung.

Elektronische Kleinteile gehören nicht in Kinderhände!

Bitte beachten Sie, dass wir keinerlei Haftung für Schäden, die aus einer falschen Bestückung, Handhabung oder Anschlussfehler der Baugruppe übernehmen, da diese außerhalb unseres Einflussbereiches liegen.