

ICmega8-8, ICmega8-16

Frank Schäfer, In-Circuit GmbH, frank.schaefer@in-circuit.de

I. EIGENSCHAFTEN

- **Controller:** ATmega8 von Atmel, 8Bit RISC Microcontroller mit 8 MHz Quarztakt (*ICmega8-16*: 16 MHz)
 - Flash (Kbytes): 8
 - EEPROM (Kbytes): 0.5
 - SRAM (Bytes): 1024
 - Max I/O Pins: 23
 - Timer: 2x 8Bit, 1x 16Bit
 - PWM-Kanäle: 3
 - RTC: ja
 - Schnittstellen: UART, TWI, SPI
 - 10-bit AD-Wandler Kanäle: 8
 - Interrupts: 18 (3 extern)
 - weitere Eigenschaften: Analoger Komparator, Brown Out Detector, Watchdog, On-Chip Oszillator, Hardware Multiplizierer
- **User-I/O:** Reset-Taster, User-Taster, Power-LED, User-LED, alle I/O-Pins an Stiftleiste verfügbar
- **I²C-Schnittstellen:** 1 (an zwei I²C Buchsen)
- **Vcc (V):** *ICmega8-8*: 2.7-5.5; *ICmega8-16*: 4.5-5.5V
- **Stromaufnahme (mA):** 9.5¹
- **Größe L x B x H (mm³):** 66.9 x 26.0 x 15.1
- **Gewicht (g):** 12

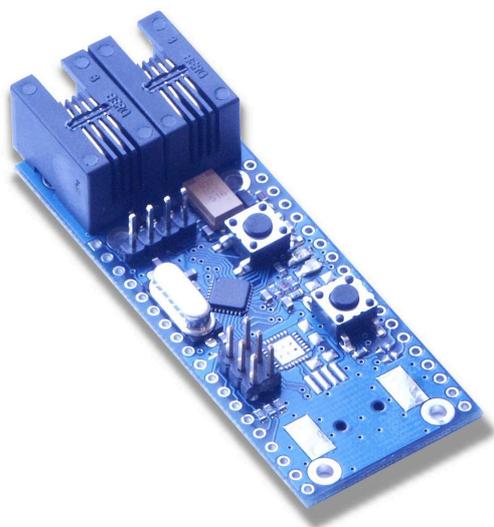


Abbildung 1. *ICmega8-x*

II. SCHNITTSTELLEN

A. Stiftleisten J1

Die Pinbelegung an den Stiftleisten des Moduls ist der Tabelle 1 zu entnehmen. Es ist zu beachten, dass die Pinnummern 20 bis 45 beim *ICmega8-x* nicht vergeben sind. Diese Pins sind Modulen mit einer größeren Pin-Anzahl vorbehalten, wobei die Module in den Bereichen mit gleicher Pin-Nummerierung kompatibel bleiben. Die Funktion der Pins läßt sich aus dem Schematic (Abbildung 3) und der Beschreibung des ATmega8 erschließen. Die Anordnung der Pins zeigt Abbildung 2.

Pin#	Name	Pin#	Name
1	/RESET	64	Vcc
2	NC	63	PC0(ADC0)
3	NC	62	PC1(ADC1)
4	NC	61	PC2(ADC2)
5	NC	60	PC3(ADC3)
6	NC	59	NC
7	NC	58	NC
8	ADC6	57	NC
9	ADC7	56	NC
10	GND	55	GND
11	PB0(ICP1)	54	PD0(RXD)
12	PB1(OC1A)	53	PD1(TXD)
13	PB2(SS, OC1B)	52	PD2(INT0)
14	PB3(MOSI, OC2)	51	PD3(INT1)
15	PB4(MISO)	50	PD4(XCK/T0)
16	PB5(SCK)	49	PD5(T1)
17	NC	48	PD6(AIN0)
18	NC	47	PD7(AIN1)
19	GND	46	GND

Tabelle 1. Pinbelegung der Stiftleisten J1

B. Programmierschnittstelle

Die doppelreihige Pinleiste J4 (Tabelle 2) stellt das Programmier-Interface des Moduls dar.

Pin#	Name
1	Vcc
2	/Reset
3	SCK
4	MISO
5	MOSI
6	GND

Tabelle 2. Pinbelegung Programmierinterface, J4

C. I²C-Schnittstelle

Am *ICmega8-x* befinden sich zwei I²C-Buchsen J6 und J5 (Tabelle 3) die beide am gleichen I²C-Bus angeschlossen sind. Durch die zwei Buchsen ist das Durchschleifen

¹@3.3V, 8 MHz, Testsoftware

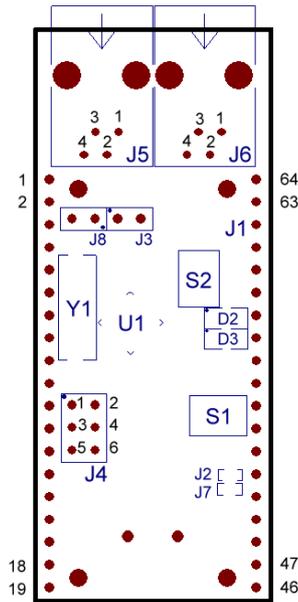


Abbildung 2. Ansicht von oben

der I²C-Signale und der Versorgungsspannung gewährleistet.

Buchse 1, J6		Buchse 2, J5	
Pin#	Name	Pin#	Name
1	Vccb1	1	Vccb2
2	SCL	2	SCL
3	SDA	3	SDA
4	GND	4	GND

Tabelle 3. Pinbelegung I²C-Buchsen J5 und J6

III. SPANNUNGSVERSORGUNG

Für die Spannungsversorgung existieren zwei Alternativen - entweder erfolgt die Zuführung über eine der I²C-Buchsen J5, J6 oder über die Stiftleisten J1.

A. I²C-Buchse

Soll das ICmega8-x Modul mit anderen Modulen oder Komponenten per I²C kommunizieren, so kann die Versorgungsspannung der Module ebenfalls über die I²C-Verbindung erfolgen. Im Idealfall befindet sich im System ein Stromversorgungsmodul, z.B. das IClinpower Modul von IN-CIRCUIT, und stellt mehreren Modulen die Versorgungsspannung zur Verfügung. Mit den Jumpfern J3 und J8 kann festgelegt werden von welcher der I²C-Buchsen die Versorgungsspannung bezogen werden soll. Wenn beide Jumper geschlossen sind, wird die Versorgungsspannung von einer I²C-Buchse zur anderen weitergeleitet und somit anderen Modulen am I²C-Bus zur Verfügung gestellt. Durch entsprechendes Setzen der Jumper ist es möglich,

Teile eines Systems mehrerer Module mit eigenen Spannungen unterschiedlichen Betrags zu speisen und trotzdem die Kommunikationsmöglichkeit per I²C zwischen allen Modulen zu wahren.

B. Stiftleisten

Die Versorgungsspannung kann auch über Vcc (64) und GND (10,19,46,55) an den beiden Stiftleisten J1 eingespeist werden. Soll die Spannung auch andere Module betreiben, die per I²C an den Buchsen J5, J6 angeschlossen sind, so geschieht dies durch schließen der Jumper J3 bzw. J8.

IV. LEDs, TASTER UND JUMPER

A. LEDs

Das Modul besitzt zwei LEDs. Eine davon ist eine Power-Indikator LED D2, die bei angelegter Betriebsspannung permanent leuchtet, die andere LED (D3) ist an dem Port PD3 des ATmega8 angeschlossen. Die LED D3 ist direkt vom ATmega8 zu schalten und leuchtet, wenn PD3 Low-Pegel führt.

B. Taster

Eine Betätigung des Tasters S2 bewirkt ein Reset des Controllers. Der Taster S1 kann vom ATmega8 am Port PD2 abgefragt werden oder einen externen Interrupt auslösen.

C. Jumper

Auf der Unterseite der Platine befinden sich 2 Lötjumper J9, J10 welche geschlossen sein sollten.

Die Lötjumper J2 und J7 auf der Platinenoberseite (Abbildung 2) verbinden den User-Taster S1 und die LED D3 mit den I/O-Ports PD2 und PD3 des ATmega8. Im Falle, dass diese Ports an den Stiftleisten J1 benötigt werden und der Taster und die LED abgetrennt sein sollen, können die beiden Lötjumper geöffnet werden.

Die Jumper J3 und J8 verbinden die Versorgungsspannung des ICmega8-x mit den Spannungen auf dem Vccb2 und Vccb1 des I²C-Bus. Wie bereits in Abschnitt III beschrieben, kann damit die Zuführung und Weiterleitung der Betriebsspannung ausgewählt werden.

V. TESTSOFTWARE UND DESIGNFLOW

A. Testsoftware

Das ICmega8-x Modul wird mit vorprogrammierter Testsoftware ausgeliefert, die unmittelbar nach dem Zuschalten der Betriebsspannung startet. Die Software schaltet alle Ausgänge der Stiftleiste auf High und dann in einer Endlosschleife jeweils ein Pin für ca. 0.5-1.0s auf Low. Die LED D3 blinkt dazu kontinuierlich im Takt. Eine Betätigung des Taster S1 erzwingt einen Low-Pegel an allen Ausgängen für 2 Blinkperioden der LED D3.

B. Designflow

Das *ICmega8-x* Modul ist völlig frei programmierbar und benötigt kein Betriebssystem sowie keinen Bootloader. Der ATmega8 kann entweder direkt in Assembler oder in einer Hochsprache wie z.B. C, C++ oder PASCAL programmiert werden.

1) *Assembler*: Für die Programmierung in Assembler bietet sich das *AVR Studio* an welches auf der Atmel-Homepage kostenlos bereitgestellt wird. Dort können mit Hilfe eines Wizards einfach Programmgerüste erstellt werden. Der Assembler des AVR Studio erzeugt auch den Hex-Code zum Programmieren des ATmega8.

2) *Hochsprachen*: Die am weitesten verbreitete Hochsprache ist C/C++ für die freie Compiler im Internet angeboten werden (GNU WinAVR). Darüber hinaus existieren auch kommerzielle Entwicklungsumgebungen (ICCAVR: <http://www.imagecraft.com/>), von denen teils limitierte Testversionen heruntergeladen werden können. Das Angebot an Quelltexten ist umfangreich wodurch Eigenentwicklungen beschleunigt werden.

3) *Simulator*: Das AVR-Studio beinhaltet ebenfalls einen leistungsfähigen Simulator, mit dem Interrupts und I/O-Signale nachgebildet werden können. Eine Simulation auf C-Ebene ist in Abhängigkeit vom Compiler möglich.

4) *Programmer*: Der Hex-Code, der vom Assembler oder Compiler erzeugt wird, kann mit einem Programmer über den Programmieradapter J4 in den Flash des ATmega8 geschrieben werden. Der Programmer besteht also aus PC-seitiger Software, z.B. dem AVR Studio oder Ponyprog, und einer Hardware, die am PC an der seriellen bzw. parallelen Schnittstelle oder an einem USB-Port angeschlossen ist. Der *ICavrprog-USB* von IN-CIRCUIT ist ein schneller und komfortabler USB-Programmer, der zum AVR Studio kompatibel ist.

VI. SCHALTPLAN

Der Schaltplan des *ICmega8-x* ist in Abbildung 3 dargestellt. Das USB-Interface bestehend aus USB-Controller und USB-Buchse bleibt bei den *ICmega8-x* Modulen unbestückt.

VII. MECHANISCHE DATEN

Die wichtigsten Abmessungen² des *ICmega8-x* Moduls sind in Abbildung 4 angegeben (Modulhöhe: 15.1mm).

²alle Angaben in mm

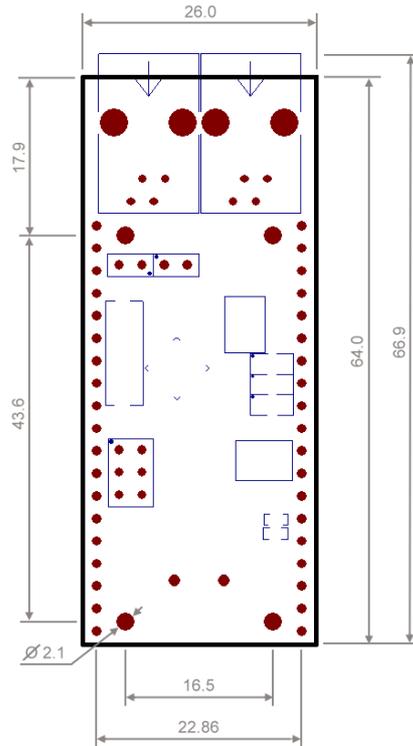


Abbildung 4. Ansicht von oben

VIII. KONTAKT

In-Circuit GmbH
Hüttenstr. 14
D-01705 Freital
Germany

Internet: <http://www.in-circuit.de/>
Email: office@in-circuit.de

Tel: 0351-6485200
Fax: 0351-6485202

Geschäftsführung: Frank Schäfer, Jörg Träger
Handelsregister: HRB 23099, Amtsgericht Dresden

USt.-Id.Nr.: DE237550066

Steuernummer: 206/111/00869, Finanzamt Freital

Vertrieb: <http://www.ic-board.de/>

Copyright ©2005, In-Circuit GmbH

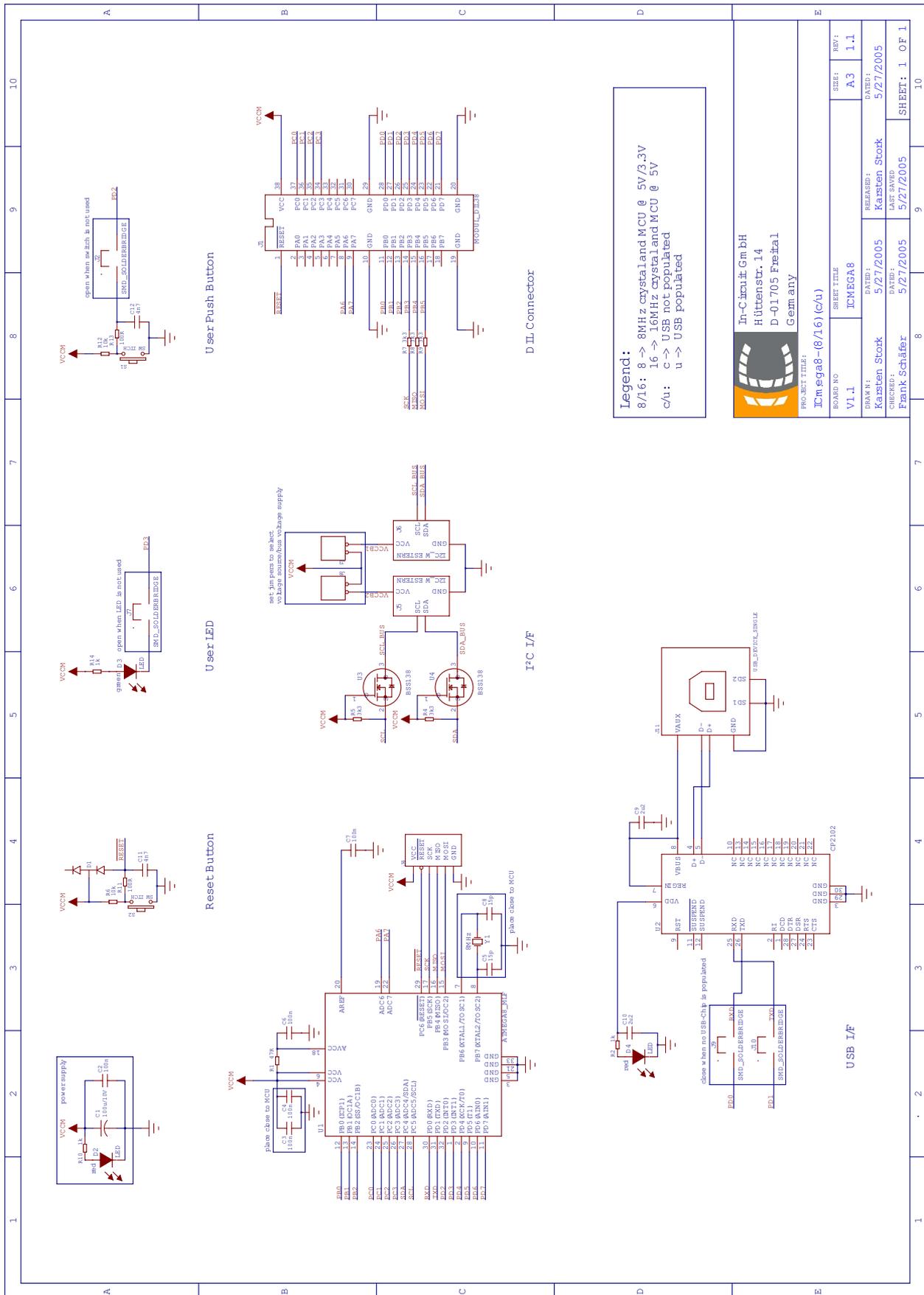


Abbildung 3. Schematic