

ICmega8 / ICmega8-USB

I. EIGENSCHAFTEN

- **Controller:** ATmega8 von Atmel, 8Bit RISC Micro-controller mit 8 MHz Quarztakt
 - 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
- **USB-Schnittstelle:** 1xDevice (CP2102: 300bps - 1Mbps)
- **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)
- **Versorgungsspannung (V):** 2.7 - 5.5
- **Stromaufnahme:** ca. 10mA bei 8 MHz, Testsoftware, *ICmega8-USB* von USB gespeist
- **Größe L x B x H (mm³):** 68.8 x 26.0 x 15.1
- **Gewicht (g):** 15



Abbildung 1. *ICmega8-USB*

II. SCHNITTSTELLEN

A. USB

Das *ICmega8-USB* besitzt einen USB 2.0 full-speed Controller (Silicon Laboratories¹ USB 2.0 full-speed function controller CP2102), der seriell mit dem ATmega8 kommuniziert und die Signale an der USB-B-Buchse J11 bereitstellt. Der USB-Controller wird aus dem USB-Bus des PCs gespeist. Der CP2102 enthält einen 3.3V Spannungsregler. Diese 3.3V können dem Board durch das Schließen des Lötjumpers *J12* auf der Unterseite des Boards zugänglich gemacht werden

Die Software des ATmega8 zum Übertragen von Daten per USB ist identisch mit der für eine Kommunikation via RS232, da vom Treiber für den CP2102 eine virtuelleCOM-Schnittstelle zum PC eingerichtet wird. Detaillierte Informationen sind den Datenblätter des ATmega8 und des CP2102 zu entnehmen. Windows-Treiber für den PC ist im Internet unter <http://www.ic-board.de/> oder Silicon Laboratories erhältlich. Für Linux existieren in den gängigen Distributionen Treiber für den CP2102.

B. Stiftleisten J1

Die Pinbelegung an den Stiftleisten des Moduls ist der Tabelle 1 zu entnehmen. Die Funktion der Pins läßt sich aus dem Schaltplan (Abbildung 4) und der Beschreibung des ATmega8 erschließen. Die Anordnung der Pins zeigt Abbildung 2.

| Pin# | Name | Pin# | Name |
|------|----------------|------|-------------|
| 1 | /RESET | 38 | Vcc |
| 2 | NC | 37 | PC0(ADC0) |
| 3 | NC | 36 | PC1(ADC1) |
| 4 | NC | 35 | PC2(ADC2) |
| 5 | NC | 34 | PC3(ADC3) |
| 6 | NC | 33 | NC |
| 7 | NC | 32 | NC |
| 8 | ADC6 | 31 | NC |
| 9 | ADC7 | 30 | NC |
| 10 | GND | 29 | GND |
| 11 | PB0(ICP1) | 28 | NC |
| 12 | PB1(OC1A) | 27 | NC |
| 13 | PB2(SS, OC1B) | 26 | PD2(INT0) |
| 14 | PB3(MOSI, OC2) | 25 | PD3(INT1) |
| 15 | PB4(MISO) | 24 | PD4(XCK/T0) |
| 16 | PB5(SCK, OC2) | 23 | PD5(T1) |
| 17 | NC | 22 | PD6(AIN0) |
| 18 | NC | 21 | PD7(AIN1) |
| 19 | GND | 20 | GND |

Tabelle 1. Pinbelegung der Stiftleisten J1

¹<http://www.silabs.com/>

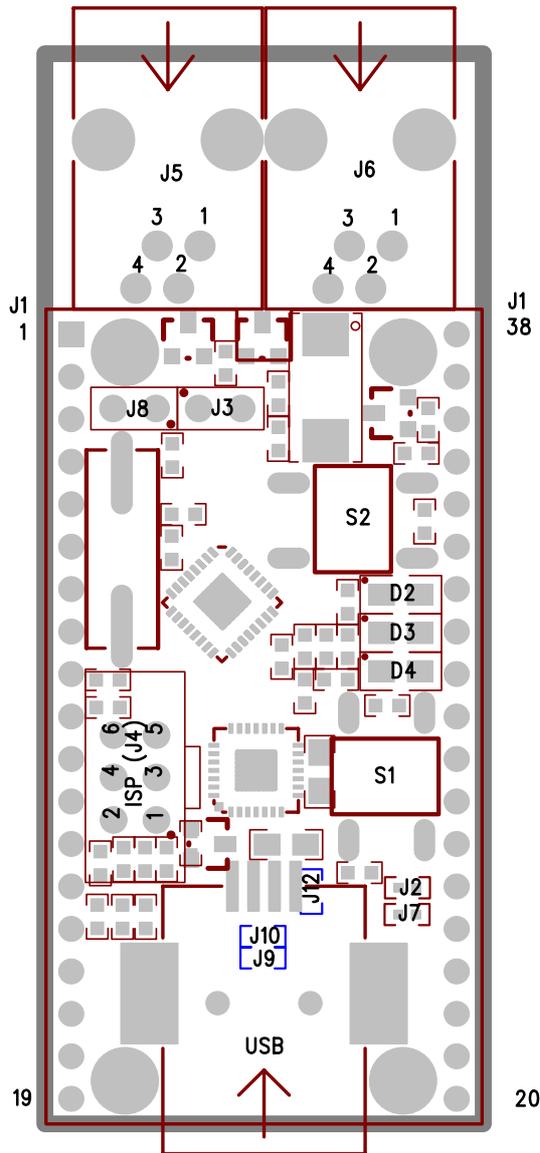


Abbildung 2. Pinbelegungen, Ansicht von oben

| Pin# | Name | Pin# | Name |
|------|--------|------|------|
| 1 | MISO | 2 | Vcc |
| 3 | SCK | 4 | MOSI |
| 5 | /RESET | 6 | GND |

Tabelle 2. Pinbelegung ISP Programmierinterface, J4

| 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

I²C-Buchsen J5, J6 oder über die Stiftleisten J1. Das *ICmega8-USB* Modul kann zusätzlich die Versorgungsspannung aus dem USB-Baustein CP2102 gewinnen. Dazu muss der Lötjumper J12 auf der Unterseite des Moduls geschlossen werden.

A. Versorgung über I²C-Buchse

Soll das *ICmega8-USB* 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 Jumpern 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. Versorgung über Stiftleisten J1

Die Versorgungsspannung kann auch über Vcc (38) und GND (10,19,20,29) 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.

C. Versorgung über USB

Das *ICmega8-USB* Modul besitzt einen USB 2.0 full-speed Controller CP2102. Der USB-Controller wird aus dem USB-Bus des PCs gespeist. Er enthält einen 3.3V Spannungsregler. Diese 3.3V können dem Board durch das Schließen des Lötjumpers J12 auf der Unterseite des Boards zugänglich gemacht werden. Ist dieser Lötjumper geschlossen, leuchtet LED D4.

C. Programmierschnittstelle

Die doppelreihige Pinleiste J4 (Tabelle 2) stellt das ISP Programmier-Interface des Moduls dar. Die Pinbelegung entspricht dem AVR910 Standard. Pin 1 ist durch einen kleinen Punkt auf der Oberseite des Boards markiert.

D. I²C-Schnittstelle

Am *ICmega8-USB* 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 der I²C-Signale und der Versorgungsspannung gewährleistet.

III. SPANNUNGSVERSORGUNG

Für die Spannungsversorgung existieren mehrere Alternativen - entweder erfolgt die Zuführung über eine der

IV. LEDs, TASTER UND JUMPER

A. LEDs

Das Modul besitzt zwei LEDs. LED D2 dient als Power-Indikator, die bei angelegter Betriebsspannung permanent leuchtet, die LED D3 ist an dem Port PD3 des ATmega8 über den Lötjumper J7 angeschlossen. Die LED D3 ist direkt vom ATmega8 zu schalten und leuchtet, wenn PD3 Low-Pegel führt. Die LED D4 leuchtet, wenn der Lötjumper J12 geschlossen ist und das Modul somit über den USB Port gespeist wird (nur beim ICmega8-USB).

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 *offen* sein sollten. Die verbinden die Pins RX und TX des Controllers entsprechend mit den Pins PD0(Pin 28) und PD1(Pin 27) der Stiftleiste J1.

Die geregelten 3.3 Volt des CP2102 können beim ICmega8-USB durch das Schließen des Lötjumpers J12 auf der Unterseite des Boards mit der Versorgungsspannung verbunden werden. Dadurch ist eine Speisung von USB möglich.

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. Im Auslieferungszustand des Modules sind die Lötjumper geschlossen.

Die Jumper J3 und J8 verbinden die Versorgungsspannung des ICmega8-USB mit den Spannungen auf dem V_{cc2} und V_{cc1} des I²C-Bus. Wie bereits in Abschnitt III beschrieben, kann damit die Zuführung und Weiterleitung der Betriebsspannung ausgewählt werden.

ACHTUNG - Bitte achten Sie darauf, durch das Schließen der Lötjumper(J12) oder Jumper(J3, J8) keine Kurzschlüsse zu erzeugen, bzw. Versorgungsspannungen verschiedener Beträge untereinander zu verbinden!

V. TESTSOFTWARE UND DESIGNFLOW

A. Testsoftware

Das ICmega8-USB 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-USB 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++, PASCAL oder BASIC programmiert werden.

1) *Assembler*: Für die Programmierung in Assembler bietet sich das *AVR Studio* an welches auf der Atmel-Homepage kostenlos bereitgestellt wird.

2) *Hochsprachen*: Die am weitesten verbreitete Hochsprache ist C/C++, für die freie Compiler im Internet angeboten werden. Wir empfehlen den GCC-port *WinAVR* zu nutzen, da dieser auch vom AVR Studio aus benutzbar ist.

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 ISP Programmierstecker J4 in den Flash des ATmega8 geschrieben werden. Zum programmieren benötigt man PC-seitige Software und eine Hardware, die am PC an der seriellen bzw. parallelen Schnittstelle oder an einem USB-Port angeschlossen ist.

Wir empfehlen für die Programmierung den *ICprog-AVR*. Der *ICprog-AVR* von IN-CIRCUIT ist ein kleiner, sehr schneller und komfortabler USB-Programmer. PC-Softwareseitig unterstützt er alle AVR910-Kompatiblen Programmierer. Wir empfehlen entweder das im AVR Studio enthaltene Programm *avrprog.exe* oder das ebenfalls freie Programm *avrdude* zu nutzen. Einen Link zum download finden Sie unter www.ic-board.de.

VI. MECHANISCHE DATEN

Die wichtigsten Abmessungen² des *ICmega8-USB* Moduls sind in Abbildung 3 angegeben (Modulhöhe: 15.1mm).

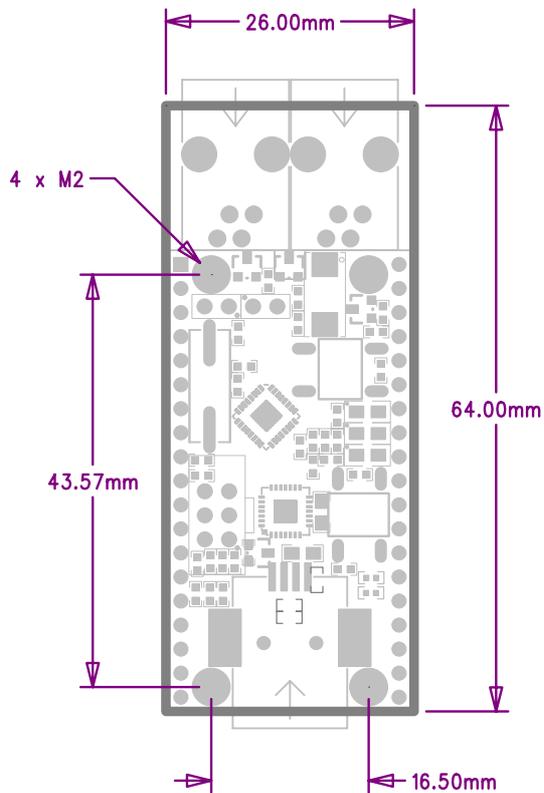


Abbildung 3. Mechanische Abmessungen, Ansicht von oben

VII. SCHALTPLAN

Der Schaltplan des *ICmega8-USB* ist in Abbildung 4 dargestellt.

Copyright ©2006, In-Circuit GmbH

²alle Angaben in mm

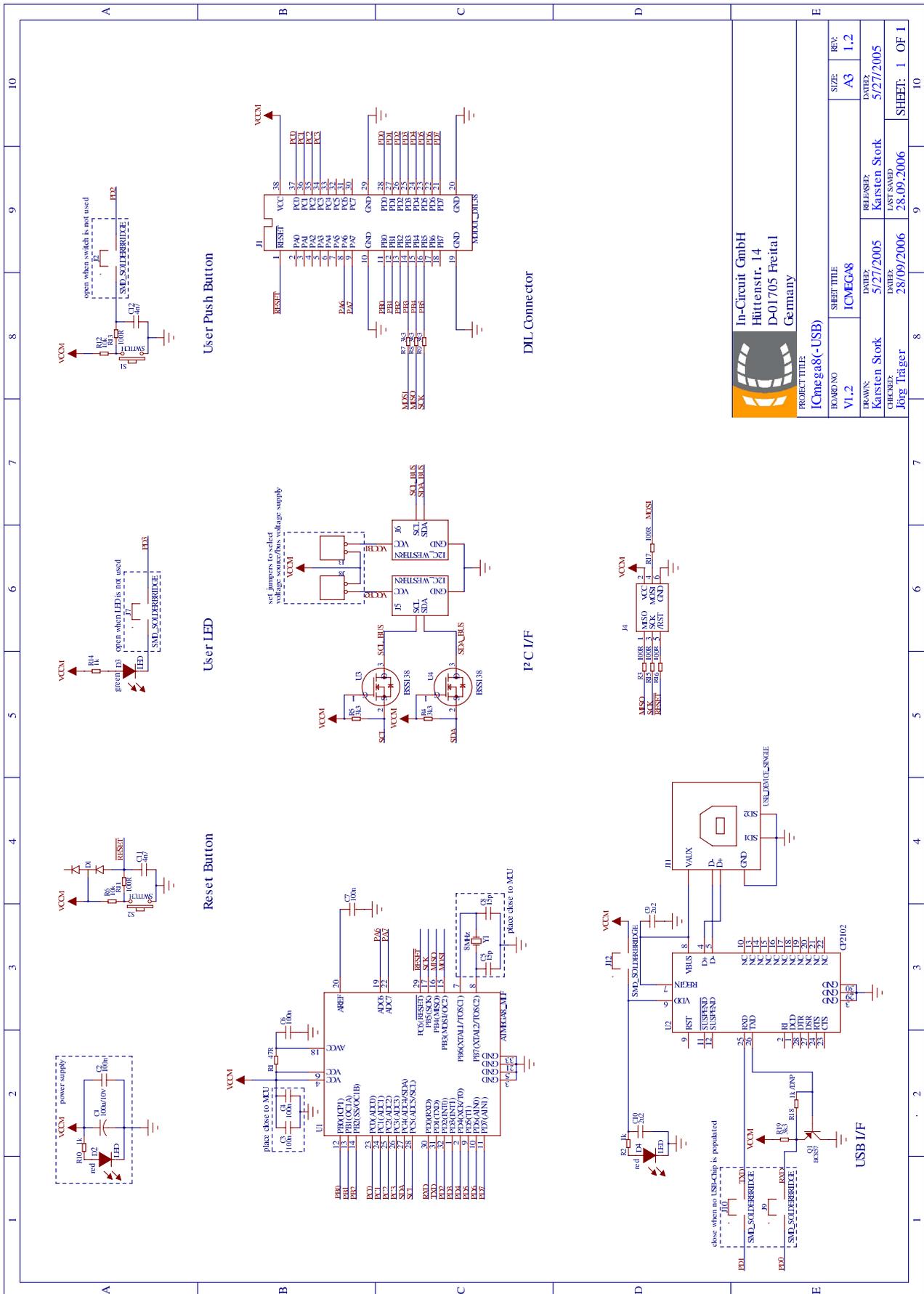


Abbildung 4. Schematic