

AVR ATxmega128A1 Experimentierplatine, Benutzerhandbuch

Michael Groß (michael@coremelt.net)

generiert: 7. November 2009 (*Revision* : 75)

Inhaltsverzeichnis

Vorwort

Das vorliegende Handbuch entstand als Begleitwerk zu einem Projekt, das ursprünglich in den Foren auf mikrocontroller.net entwickelt wurde. Es beschreibt den Aufbau und die Funktionsweise einer peripheriereichen und kompakten Mikrocontroller-Experimentierplatine auf Basis des neuen AVR ATxmega128A1-Controllers der Firma Atmel. Da diese Controller noch neu sind und somit noch Erfahrung fehlt, beschreibt dieses Handbuch ebenfalls die Einrichtung und Verwendung der GNU-Toolchain unter Linux als Alternative zum AVR-Studio, der offiziellen Entwicklungsumgebung von Atmel¹.

Das gesamte Projekt, inklusive Dokumentation und CAD-Daten, wird in einem offiziellen SVN-Archiv gepflegt und verwaltet. Dort befindet sich ebenfalls die aktuelle Fassung dieses Handbuchs. Da das Handbuch in Teilen geschrieben wird ist es sinnvoll, sich zunächst die aktuelle Fassung zu besorgen. Wer inhaltliche oder rechtschriftliche Fehler findet, mag diese bitte unter michael@coremelt.net an den Autor melden, so dass sie als Korrekturen einfließen können. Ebenfalls ist jeder willkommen, der beim Verfassen der Dokumentation helfen möchte.

Dieses Handbuch soll dem geneigten Leser einen Einstieg in die Materie ermöglichen und die Funktionen der vorliegenden Experimentierplatine näher beschreiben. Es vermittelt jedoch keine grundlegenden Programmierkenntnisse und Konzepte der Entwicklung mit Mikrocontrollern. Ebenfalls richtet sich die behandelte Platine aufgrund ihres Aufbaus nicht an Einsteiger ohne SMD-Lötkenntnisse. Die schlimmsten Fallstricke werden jedoch bebildert aufgezeigt.

An dieser Stelle ist der Leser aufgerufen, Informationen, Erfahrungen und natürlich Quellcode und Projekte auf Basis dieser Platine der Allgemeinheit zur Verfügung zu stellen. Dies verbessert nicht nur die Qualität dieses Projekts, sondern erweitert auch das Anwendungsspektrum. Eine solide Basis an Beispielcode verhilft Einsteigern zu schnelleren Ergebnissen und mindert den Frust bei missglückten Experimenten. Es sollte jedoch nie vergessen werden, dass ein Experiment ohne ausreichende Dokumentation nur beschränkt von Nutzen für andere ist.

Wir wünschen Dir viel Freude mit diesem Projekt,

Erlangen im Sommer 2009

¹Das AVR-Studio verwendet zur Programmierung in C ebenfalls die GNU Compiler Collection (GCC), welche von dem offenen Projekt WinAVR bereitgestellt wird.

Nomenklatur dieses Handbuchs

Wie in Fachliteratur üblich, bedient sich auch dieses Handbuch gewisser Paradigmen und Darstellungsweisen. Hierbei wurde darauf geachtet, dass das Erscheinungsbild einen vertrauten und üblichen Eindruck bildet. In diesem Abschnitt werden die verwendeten Formate nochmals im Detail beschrieben.

Satzform und Lizenz: Dieses Buch wurde mit dem Satzpaket L^AT_EX als Buch erstellt. Der Dokumenttyp Buch optimiert dieses Schriftstück zur Verwendung als doppelseitig gedrucktes und gebundenes Dokument. Ein einseitiger Ausdruck ist daher nicht zu empfehlen. Wer das Handbuch lieber als Bericht drucken möchte, sollte sich die Quellen aus dem Archiv besorgen und den Dokumenttyp entsprechend ändern – die offizielle Fassung wird jedoch als Buch gepflegt. Das Handbuch wird unter den Bedingungen der GNU Free Document License vertrieben. Die Bedingungen dieser Lizenz können unter <http://www.gnu.org/copyleft/fdl.html> eingesehen werden.

Formatierung von Quellcode: Die verwendete Programmiersprache ist ANSI C. Quellcode wird in folgender Form in den Text zur Veranschaulichung eingebunden:

```
1 #include <stdio.h>
2
3 int main(void) {
4     printf("Hello, world!\n");
5
6     return 0;
7 }
```

Um Bezug auf den Quellcode nehmen zu können, befinden sich an der linken Seite die korrespondierenden Zeilennummern. Farbliche Differenzierung (engl. “syntax highlighting”) vereinfacht den Lesefluss.

Hinweise und Warnungen: Im Umgang mit Hardware ist es oftmals erforderlich, auf Gefahren hinzuweisen, welche zu einer Beschädigung führen können. Ebenfalls sind

oftmals weiterführende und nützliche Hinweise angebracht. Diese werden in entsprechend hervorgehobenen Kästen wie folgt dargestellt:



Achtung: Die Klemme X1 ist nicht verpolungssicher. Bei inkorrekt polarisierter Schaltung kann die Schaltung irreparable Schäden nehmen.

Bezeichner aus Quellen und Schaltplänen: Schlüsselwörter aus Quelltexten und Bezeichner von Bauelementen oder Netzen aus Schaltplänen werden mit **Blocksatz** formatiert, um die besondere Bedeutung hervorzuheben. Die Bezeichner lauten immer gleichnamig mit den Elementen, die sie referenzieren, so dass ein problemloses Auffinden möglich ist. Beispiel: Die Diode D1 sollte erst am Schluss eingelötet werden.

Research is what I'm doing when I don't know what I'm doing.

Wernher von Braun

1

Die Experimentierplatine

Dieses Kapitel behandelt den Aufbau der Platine und die Beschreibung der Funktionseinheiten. Bevor Programme auf der MCU zur Ausführung gebracht werden können, muss der Entwicklungsrechner über eine entsprechend eingerichtete Entwicklungsumgebung verfügen, siehe hierzu in Kapitel 2. Wer im Umgang mit Elektronik und Lötkolben erfahren ist, kann den einleitenden Teil und die Hilfestellung beim Aufbau überspringen und sofort zur Funktionsbeschreibung übergehen.

1.1 Benötigte Ausstattung

Dieser Abschnitt soll die Frage beantworten, welche Ausrüstung für den erfolgreichen Umgang mit der Experimentierplatine unbedingt benötigt wird. Schließlich soll es möglich sein, sinnvolle Experimente durchzuführen und diese dann durch Messungen zu belegen. Gerade im Umgang mit Mikrocontrollern wird es schnell frustrierend, wenn man bei der Entwicklung schätzen und raten muss, um einem Problem auf die Spur zu kommen. Dies hindert die Produktivität und trübt die Freude nicht unerheblich. Es gilt also der Leitfaden, dass man am Ende die Erfolge nur so gut sein können wie die Ausrüstung, mit der sie erzielt wurden.

Minimalausrüstung für den Umgang mit der Experimentierplatine

- Ein PC

Dient zur Entwicklung der Programme und als Host-System zum Übertragen der fertigen Programme auf den Mikrocontroller. Der PC braucht nicht sonderlich stark sein, aber ein Rohrkrepierer ist ebenfalls ungeeignet.

- Ein Programmiergerät

Mit dem Programmiergerät wird das fertige Kompilat vom PC über eine geeignete Schnittstelle (z.B. RS-232 oder USB) an den Controller übertragen.

- Spannungsquelle

Für die Experimentierplatine reicht ein unstabilisiertes Steckernetzteil mit einer Ausgangsspannung von mindestens 12V. Es sollte je nach Anwendung möglich sein, etwa 1A Strom zu entnehmen und auch unter Last nicht unter 12V abfallen. Am besten eignet sich ein entsprechend starkes Labornetzteil. Hinweis: Wer auf die Verstärkung der DAC-Wandler verzichten kann, kommt hier mit 5V aus.

- Etwa zwei bis drei 10-polige Flachbandkabel mit gequetschten Stecker (Schneid-Klemm-Ausführung) zum Verbinden der Funktionseinheiten auf dem Entwicklungsboard mit den Port-Stiftleisten. Die Kabel sollten etwa 10 bis 15cm lang sein.

- Ein Digitalvoltmeter

Zum Messen von Spannungspegeln an den Ports oder Bauteilen wird ein DMM benötigt. Ein einfaches DMM reicht für diesen Zweck, ist aber dringend zu empfehlen, wenn man nicht zu sehr im Dunklen tappen will.

- Ein Speicheroszilloskop *optional*

Für den Anfang ein teurer Ausrüstungsgegenstand. Für erweiterte Experimente und Projekte mit Mikrocontrollern jedoch unverzichtbar. Da die Signale meist nicht kontinuierlich sind, wird ein Speicheroszilloskop benötigt.

Zum verifizieren von rein digitalen Signalen kann ein Logikanalysator sehr von Vorteil sein. Mischsignal-Oszilloskope bieten diese Funktionalität ebenfalls, sind jedoch mitunter sehr teuer.

- Weitere Ausrüstung *optional*

Lötkolben/-station, Verbindungskabel, Zangen, Lochrasterplatinen und elektronische Bauteile, Draht, LCD-Display mit Standardcontroller, weitere Quarze, Funktionsgenerator, Ausrüstung zur Herstellung von zusätzlichen Leiterplatten, uvm.

Welches Programmiergerät? Es gibt eine Vielzahl von Programmiergeräten für die AVR-Familie. Jedoch ist die Zahl derer, die die ATxmega-Familie unterstützen, recht überschaubar. Im folgenden wird eine Aufstellung gegeben:

- Atmel AVRISP mkII (USB)

Einer der offiziellen Programmer. Besitzt eine USB-Schnittstelle und beherrscht den PDI-Modus (In System Programming). Erhältlich bei Reichelt. Der AVRISP mkII kann nicht zum Debuggen verwendet werden, ist jedoch recht günstig. Hinweis:

Um den ATxmega programmieren zu können, benötigt man die neuere Version des AVRISP, welcher eine interne LED besitzt. Ältere Versionen sind für die Programmierung des ATxmega nicht geeignet.

- JTAG ICE mkII (USB)

Das JTAG ICE mkII ist ein JTAG-fähiges Programmiergerät, mit dem auch das laufende System genauer untersucht werden kann (engl. in circuit debugging). Mit diesem Gerät hat man alles Nötige in der Hand, um mit den ATxmegas arbeiten zu können und auch komplexere Szenarien zu beherrschen, es ist jedoch recht teuer. Als Anfänger ist ein AVRISP zunächst ausreichend.

- Atmel STK600 (USB)

Kein reines Programmiergerät, sondern eine komplette Entwicklungsplattform für die AVR-Familie. Um mit dem ATxmega128A1 arbeiten zu können, braucht man einen Adaptersatz für das TQFP-100-Gehäuse, was mit etwa 100EUR zu Buche schlägt. Wer jedoch bereits ein STK600 besitzt, kann per PDI aber auch JTAG die Experimentierplatine ohne das Adapterkit programmieren.



Was tun, wenn keiner der Programmiergeräte aus obiger Liste zur Verfügung steht? Der ATxmega kann mit einem passenden Bootloader auch über die USB-Schnittstelle programmiert werden. Selbst in diesem Szenario wird jedoch ein Programmiergerät zum flashen des Bootloaders selber benötigt und die Konfiguration und Installation sind nicht trivial. Wenig erfahrene sollten sich hier Hilfestellung besorgen.

Die Liste enthält nur offizielle Geräte der Firma Atmel. Dies liegt daran, dass der ATxmega ein neues Protokoll zum Programmieren verwendet (PDI), welches noch keinen weiten Einzug in freie Projekte gefunden hat. Es ist daher auf jeden Fall anzuraten, sich eines der offiziellen Geräte zu besorgen, um unnötigen Ärger durch fehlerhafte Werkzeuge aus dem Weg zu gehen. Außerdem muss die verwendete Hardware auch mit der Programmiersoftware harmonieren. Auch hier stehen die Chancen auf einen reibungslosen Ablauf mit den offiziellen Werkzeugen derzeit am besten.

1.2 Aufbau und Inbetriebnahme

Dieser Abschnitt widmet sich dem Aufbau der gefertigten Platine und gibt einige Hinweise zur Vermeidung grober Fehler, die zu Beschädigungen führen können. Der Abschnitt enthält ebenfalls Orientierungshilfen zum erfolgreichen Bestücken. Unerfahrene Anwender sollten diesen Abschnitt vollständig lesen, *bevor* mit den Lötarbeiten begonnen wird.

Da sich im Laufe der Zeit wahrscheinlich mehrere Platinenversionen im Umlauf befinden, sollte der Leser zunächst prüfen, welche Platinenversion er in den Händen hält. Der entsprechende Aufdruck befindet sich auf der rechten Seite oben:

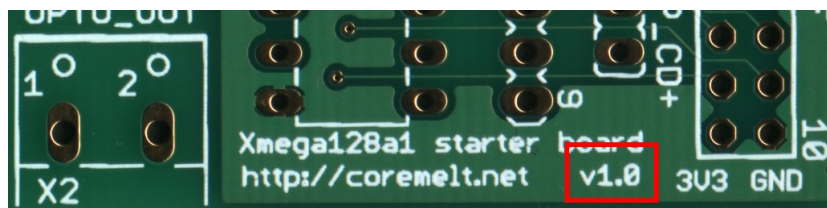


Abbildung 1.1: Versionskennzeichnung der Platine

Der Grundaufbau wird anhand der Platinenversion 1.0 beschrieben. Besonderheiten und Abweichungen finden an geeigneter Stelle Beachtung, sobald weitere Versionen im Umlauf sind.

1.2.1 Version 1.0 (Fertigung im September 2009)

Zunächst die technischen Daten dieser Platine:

technische Daten der Version 1.0	
Abmessungen	140x80 mm
Materialstärke	FR4 (1.55mm)
Kupferstärke	35 μ m
kleinster Bohrdurchmesser (Via)	0.2mm (12mil)
kleinster Leiterbahndurchmesser	0.15mm (6mil)
kleinster Abstand	0.15mm (6mil)
Bestückungsdruck	beidseitig weiß
Lötstopplack	beidseitig grün
Oberflächenveredlung	NiAu (Nickel-Gold)
E-Test	ja

Hinweis: Die Version 1.0 wurde mit Revision 39 des Verzeichnisses `/trunk/pcb/cad` gefertigt. Die CAD-Daten (Eagle \geq Version 5.6) kann man mit folgendem Befehl besorgen:

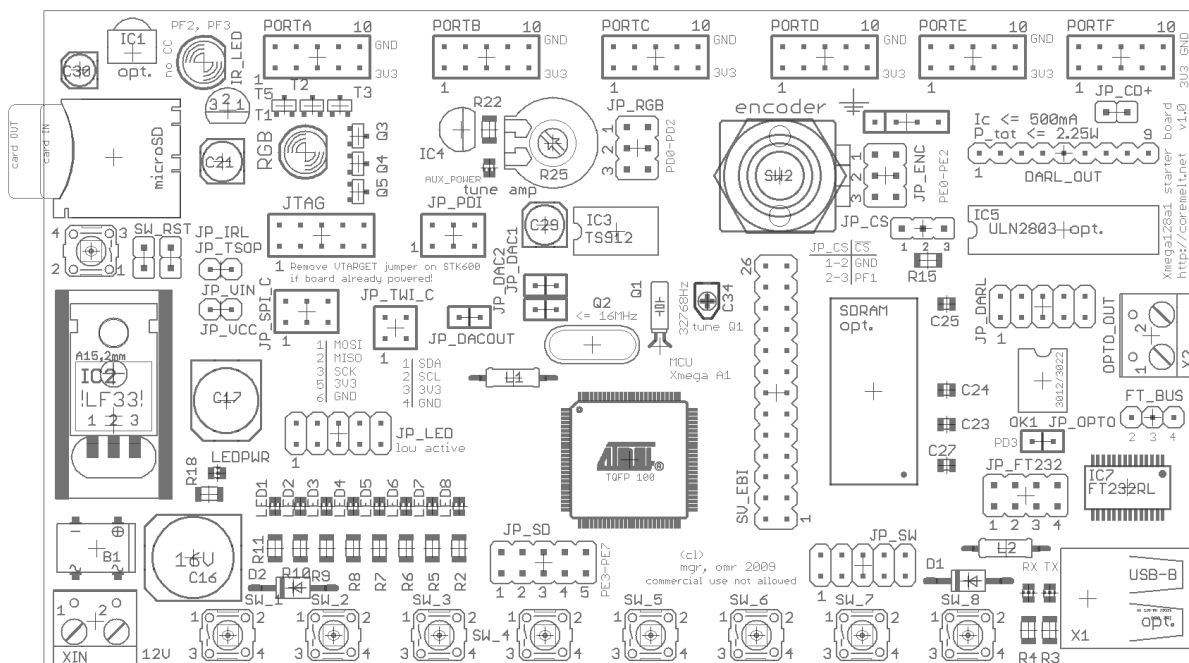
```
$ svn co -r 39 \
http://svn.coremelt.net/atxmega128a1_board/trunk/pcb/cad
```



Da die Platine mit einem E-Test gefertigt wurde, sind herstellungsbedingte Fehler (Ausschuss) fast auszuschließen. Durch die feinen Strukturen muss jedoch in jedem Fall vorsichtig mit der Platine umgegangen werden, um Beschädigungen auszuschließen. Bauteile dürfen maximal einmal wieder ausgelötet werden, da sich sonst die Pads vom Basismaterial ablösen und die Platine dadurch ggf. unbrauchbar wird. Bepinnte SMD-Gehäuse sollten grundsätzlich nur mit einer geeigneten Heißluftstation ausgelötet werden. Gute Lötkenntnisse sind in jedem Fall erforderlich.

Bauteilpositionen

Die folgenden Abbildungen helfen beim Auffinden von Bauteilen. Diese Informationen können der Platine ebenfalls direkt durch den Bestückungsdruck entnommen werden.



Bestückungsreihenfolge Es ist hilfreich, zunächst die SMD-ICs zu bestücken, danach die 1206 und 0805-Bauteile und erst zum Schluss die bedrahteten Bauteile und Stiftleisten. Da die Platine recht dicht besetzt ist, können sich diese Bauteile sonst als hinderlich erweisen und die Arbeit deutlich erschweren. Der Vorgang kann auf einzelne Bereiche der Platine beschränkt werden.

SMD-Bauteile und -Kondensatoren Die SMD-Widerstände und -Kondensatoren sind am Besten mit Lotpaste lötbar. Hierzu trägt man zunächst mit einer Spritze etwas Lotpaste auf, platziert das Bauteil und erhitzt die Pads anschließend mit dem Kolben. Bei den größeren Kondensatoren kann man zusätzlich etwas Druck von oben ausüben, um einen glatten Sitz auf der Platine zu gewährleisten. Dies verringert die Gefahr von Ermüdungsbrüchen an den Bauteilen, die durch das Spiel entstehen können. Die kleinen Bauteile (1206, 0805) platziert und hält man am Besten mit einer entsprechenden SMD-Pinzette. Beim Löten wird das Bauteil mit der Pinzette an seiner Position gehalten, um ein Verrutschen zu verhindern. Mit der Paste sollte hier eher sparsam umgegangen werden. Die Paste hat den entscheidenden Vorteil, auch als schwacher Kleber zu dienen. Daher sind diese Bauteile mit Draht eher schlecht zu löten, im Ernstfall kann jedoch ein 0.5mm-Draht (auch zum Nachlöten) verwendet werden.

Bestückungsoptionen Je nach persönlichen Anforderungen muss nicht die gesamte Platine bestückt werden. Einzelne Schaltungsteile sind optional. Hierunter fallen insbesondere: IrDA, Encoder, externer SDRAM, Optokoppler, ULN2803, microSD-Kartenslot, USB-Teil und RGB-LED. Welche Bauteile hier genau bestückt oder weggelassen werden können sollte der geneigte Leser dem Schaltplan entnehmen. Funktionseinheiten sind dort mit gestrichelten Rahmen versehen und so logisch voneinander abgetrennt. Es ist ratsam, DIL-Bausteine mit einem Sockel zu versehen, so dass man sie im Fehlerfall leicht austauschen kann.

Korrekte Montage des Kühlkörpers Der passende IC-Kühlkörper sollte auf eine der folgenden Arten montiert werden:

- Der Kühlkörper wird mit Wärmeleitpaste und einer Metallschraube mit Mutter und Beilagscheibe auf der Unterseite mit dem IC fixiert.
- Der Kühlkörper wird mit Wärmeleitkleber fixiert. Eine entsprechende Schraube ist auch hier empfehlenswert, um ein versehentliches Abscheren zu verhindern.

Der Kühlkörper kann zwar weggelassen werden, dies ist jedoch auf keinen Fall anzuraten, vor allem bei der regelmäßigen Entnahme größerer Ströme von bis zu 1A oder unter Verwendung der zweiten Spannung der Platine (8V).

Bestückung des SDRAMs und des EBI-Ports Falls ein externes SDRAM verwendet wird, sollte `SV_EBI` nicht bestückt werden, um eine Wirkung als Antennen entgegenzuwirken. `SV_EBI` dient als Erweiterungsport für den EBI, wenn kein SDRAM angeschlossen werden soll. Wenn ein 8-Bit-SDRAM verwendet wird, sollten die Pullups für die Datenleitungen entsprechend bestückt werden (siehe Schaltplan). Bitte beachten: Es ist bisher nicht erprobt, ob statt des 4-Bit-SDRAM auch ein 8-Bit-SDRAM verwendet werden kann. An dieser Stelle tritt ein Design-Fehler des ATxmega zu Tage: 4-Bit-SDRAM ist weitestgehend abgekündigt und teilweise nur noch in hohen Stückzahlen in einer non-RoHS-Fassung zu bekommen. `SV_EBI` ist für eigene Erweiterungen vorgesehen.

Wahl des Widerstands R14 Der Widerstand ist wegen des hohen erlaubten Pulsstroms der IR-Sendediode mit 1 Ohm angegeben. Die Diode darf dann nicht mit einer kontinuierlichen Spannung angesteuert werden, da sie sonst durchbrennt. Wer dieses Risiko ausschließen will, sollte R14 mit 22 Ohm bestücken, um den Strom auf ca. 100 mA zu begrenzen, welchen die Diode auch kontinuierlich verkraftet.

Bestückung von C34 Die Pads des regelbaren Kondensators C34 sind recht klein. Am Besten lötet man den Kondensator daher mit Lotpaste ein. Die Paste wird auf die Pads aufgetragen, das Bauteil platziert und unter Hitzeeinwirkung auf die Pads verlötet.

Orientierung von Q6 (MOSFET) Aus dem Bestückungsdruck von Q6 geht nicht hervor, wie das Bauteil orientiert werden muss. Der MOSFET hat eine abgeflachte Kante. Diese Kante muss nach unten, also zum Rand der Platine, zeigen. Dort befindet sich auch der Aufdruck "IRF7416", der sich dann auf der Seite der abgeflachten Kante befindet.

Bestückung von Q2 Q2 liefert den Haupttakt für die MCU. Manche Quarzgehäuse sind leitend mit dem Innenleben verbunden, deshalb müssen sie vom Rest der Schaltung isoliert werden. Auf der Unterseite des Gehäuses befindet sich jedoch ein unisoliertes Via. Q2 sollte daher entweder mit Klebeband isoliert oder mit einem Isolationsabstand zur Platine eingelötet werden, um diese Probleme zu verhindern.

Polarität der LEDs Die RGB-LED muss mit der abgeflachten Seite nach *rechts* eingelötet werden. Bei den 0805-LEDs ist die Kathode (negativ) auf der Unterseite der LED mit einem kleinen Pfeil markiert. Dieser Pfeil muss in Richtung der Markierung im Bestückungsdruck zeigen. Diese Markierung besteht aus drei Punkten, die näher am Pad der Kathode liegen. Verpolte LEDs wirken wie normale Dioden in Sperr-Richtung und leuchten nicht.

LED9 und LED10 Diese LEDs gehören als RX- und TX-LED an den FT232 und sollten eine Vorwärtsspannung von mind. 1.7V besitzen, da sie mit einem 3V3-Ausgang über eine 5V-Spannungsquelle getrieben werden.

RGB-LED von Reichelt Die RGB-LED von Reichelt stimmt von der Pin-Belegung nicht mit dem Datenblatt überein. Schlimmer noch scheint die Belegung je nach gelieferter Variante nochmals abzuweichen. Es ist daher empfehlenswert, die LED zunächst trocken an ihren Pins mit einem Widerstand auf deren Farbkanäle hin zu untersuchen und die Widerstände entsprechend umzubestücken. Unterlässt man dies, passen die Widerstände ggf. nicht zu den stark abweichenden Diffusionsspannungen der unterschiedlichen Kanäle. Auf der Platine wurden die beiden blauen Kanäle zu einem Signal zusammengefasst. Somit funktioniert die Schaltung nur wie gedacht, falls die beiden blauen Kanäle tatsächlich auf den entsprechenden Pins liegen.

Einlöten des SD-Kartenschachts Der SD-Kartenschacht kann vorsichtig an den seitigen Laschen mit den Fingern oder einem feinen Werkzeug zerlegt werden, da er einen abnehmbaren Deckel besitzt. Bei dieser Prozedur muss man darauf achten, dass sich die Feder nicht aus der unteren Plastikeinlage löst. Danach fixiert man den Slot an den beiden vorderen (nach rechts zeigenden) Pins, so dass alle anderen Pins korrekt ausgerichtet sind. Im nächsten Schritt trägt man Flussmittel auf die verlängerten Pads auf und lässt das Zinn mit dem heißen Lötkolben an die Beine laufen. Hier sollte man mit Lot großzügig sein, um die Stabilität zu verbessern. Am Ende verschließt man das Gehäuse wieder und fixiert dieses durch verlöten der verbleibenden beiden Pads. Wer das Gehäuse nicht zerlegen will, kann den Schacht auch durch das Fenster im Deckel löten, dies ist allerdings mühevoller.

Kurzschlussbrücken (Jumper) Die Platine enthält eine Reihe an Kurzschlussbrücken, die mit normalen Pinleisten bestückt werden. Hierauf werden dann die Jumper gesetzt, um gewisse Peripheriebereiche mit der MCU zu verbinden oder zu trennen. Diese Jumper sollte man immer vorsehen, auch wenn sie nicht geschlossen sind. Sie werden dann an nur einen Pin gesteckt. Auf die richtige Ausrichtung der Jumper ist immer zu achten, da sonst ggf. Ausgänge der MCU miteinander verbunden werden und diese so Schaden nehmen kann.

Löten feiner SMD-Bauteile Die Platine enthält mehrere SMD-Bauteile mit feinem Raster. Hierunter fallen die MCU, das SDRAM und der FT232RL. Solche Bauteile erfordern zum korrekten Einlöten ein spezielles Vorgehen, was im folgenden beschrieben wird. Eine feine Lötspitze wird ebenfalls benötigt. Eine Hohlkehle ist von Vorteil, ist aber nicht zwingend erforderlich. Die folgende Beschreibung bezieht sich auf die Verwendung einer normalen Lötspitze. Am besten eignet sich hier eine feinere Punktspitze. Die Spitze sollte nicht stärker als etwa 0.8mm sein. *Die beschriebene Methode funktioniert nur mit einem SMD-Flussmittel korrekt.*

1. Platzieren und Fixieren

Das Bauteil wird sorgfältig auf dem Fußabdruck ausgerichtet, so dass alle Beine bündig auf den Pads aufliegen. Hierbei muss darauf geachtet werden, dass alle Seiten korrekt ausgerichtet sind. Es ist wichtig, das Bauteil nicht mit scharfen Gegenständen oder den Fingern anzufassen. Ebenfalls sollte es nur an der Oberseite mit einer passenden SMD-Pipette angefasst werden, um das Verbiegen von Pins oder ESD-Schäden zu vermeiden. Das Bauteil kann auch unter Verwendung eines weichen Gummiwerkzeuges vorsichtig an den Pins ausgerichtet werden, jedoch nur als Feinkorrektur.

Schließlich wird das Bauteil an mindestens zwei gegenüberliegenden Seiten mit einem Lötspunkt so fixiert, dass es nicht mehr verrutschen kann. Hierbei ist es nicht wichtig, dass keine Lötbrücken entstehend, sondern lediglich, dass die Ausrichtung auch nach dem Festlöten noch korrekt ist. Nach dem Fixieren kann man das Bauteil nochmals korrekt ausrichten. Eine Taschenlampe oder Lupenlampe hilft nochmals, das Ergebnis zu überprüfen. Dies ist der schwierigste Schritt.

2. Auflöten

Nachdem das Bauteil fixiert ist, wird es von allen bepinnten Seiten großzügig mit einem Flussmittelgel behandelt. Sodann wird mit einem dünnen Lötendraht etwas Zinn aufgetragen, welches mit der heißen Lötspitze an den Beinen entlang gezogen wird. Man sollte hierbei stets in eine Richtung (z.B. nach rechts) arbeiten. Brücken sollten durch das Flussmittel nur an der rechten äußeren Seite entstehen. Diese werden am Ende mit einem Stück Entlötlitze wieder entfernt. Hierbei ist darauf zu achten, dass man nicht zu viel Lot verwendet (weniger ist mehr). Wurde zu wenig Lot aufgetragen, kann man den Vorgang wiederholen, bis alle Pads einen gut verzinnten Eindruck machen. Es ist hierbei wichtig, dass das Bauteil nicht lokal zu stark oder zu lange erhitzt wird, der LötKolben sollte stets in Bewegung bleiben, wenn er über die Pads gezogen wird.

3. Säubern und Prüfen

Am Ende befreit man den Baustein von Flussmittelresten. Hierbei ist Druckluft hilfreich. Verbleibende Reste kann man mit Kontakt LR oder Isopropanol abwaschen. Im Anschluss prüft man nochmals genau, ob keine Brücken entstanden sind. Auch hier ist die Taschenlampe hilfreich: Beleuchtet man den Fußabdruck des ICs nach dem Löten von unten, sind Kurzschlüsse sofort erkennbar. Falls noch Brücken zu sehen sind, werden diese entweder mit Entlötlitze entfernt oder unter Verwendung weiteren Flussmittels mit dem LötKolben entfernt (verwischt). Das Flussmittel verändert die Oberflächeneigenschaften des Lots und unterbindet die Brückenbildung effektiv. Auch sollte man darauf achten, dass sich unter den Pads keine Lotansammlungen gebildet haben. Bei gewissen Bauteilen kann man einen feinen Draht (0.5mm) unter den Beinen hindurch schieben, um es auf Lotfreiheit zu prüfen.

Nach der Prozedur sollte die MCU brückenfrei aufgelötet sein. Untrige Abbildung dient als Orientierungshilfe. Wer keine Adleraugen sein Eigen nennt, kann sich durch

Abfotografieren oder mit einer Lupe und Taschenlampe helfen.

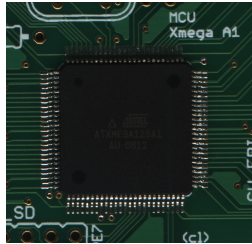


Abbildung 1.4: Eingelötete MCU

Errata

In diesem Abschnitt werden Fehler bzw. Probleme und deren Lösungsmöglichkeiten (workarounds) beschrieben, sobald diese bekannt werden.

Beschaltung des MOSFET Q6 Der MOSFET Q6 stellt einen Teil der USB-Sanft-Start-Schaltung dar, so dass keine zu großen Ströme vom Bus genommen werden, bevor dieser nicht entsprechend enumiert wurde. In Version 1.0 wird das Gate des Transistors nicht korrekt angesteuert. Es wird zusätzlich ein NPN-Transistor (z.B. BC846) als Pegelwandler benötigt. Eine Gefahr der Beschädigung ist mit der gedruckten Schaltung nicht auszuschließen.

Folgende Lösungsmöglichkeiten existieren:

- Der MOSFET Q6 wird nicht bestückt. Dies erfordert keine Reproarbeiten, führt allerdings dazu, dass die Platine nicht mehr per USB mit Spannung versorgt werden kann. Die Platine benötigt dann im USB-Betrieb eine Spannung aus XIN.
- Die Eingangsspannung des USB wird direkt an VIN gebrückt. Keine größeren Reproarbeiten, jedoch nicht mehr USB-Norm konform. Die Erfahrung zeigt jedoch (durch viele auf dem Massenmarkt erhältliche Geräte) dass dies eine nutzbare Option darstellt.
- Es wird ein weiterer Transistor und ein weiterer Widerstand (Pullup) eingelötet. Dies stellt die volle Funktionalität her, erfordert jedoch Reproarbeiten an der Platine.

Zu vervollständigen...

1.3 Beschreibung der Funktionseinheiten

Dieser Teil des Handbuchs muss leider noch geschrieben werden...