

Inbetriebnahme – AmbiController v0.4.1

Autor: Kai Lauterbach

Datum: 16.7.2015

Version: 0.1

Für die Inbetriebnahme sind folgende Hilfsmittel notwendig bzw. hilfreich:

- Multimeter
- USB-Label Typ-A-Stecker zu Mini-B-Stecker
- Lupe
- Software: <http://github.com/klaute/AmbiController>
- AVR XMeta Programmieradapter der per JTAG flashen kann (z.B. AVR Dragon)
- PC mit Linux oder Windows mit Cygwin

Ich wünsche viel Erfolg und würde mich über jede Rückmeldung freuen!

1. Sichtkontrolle

Um sicher zu stellen das keine Kurzschlüsse auf der Platine vorhanden sind sollten zunächst nochmals alle Lötstellen mit der Lupe geprüft werden. Sind hier ungewollte Kontaktierungen zu erkennen müssen diese korrigiert bzw. entfernt werden.

2. Erstes anschließen an USB

Mit einem passenden Kabel wird nun der fertig aufgebaute AmbiController mit einem USB-Anschluss verbunden. Dies muss kein PC sein. Es ist auch möglich ein USB Ladegerät zu und einen entsprechenden Adapter verwenden.

Wird nach dem einstecken des USB-Anschlusses an die Spannungsquelle einer oder mehrere Teile der Platine bzw. der Bauteile darauf heiß könnte ein Kurzschluss vorliegen. Dieser muss gefunden und korrigiert werden bevor mit der Inbetriebnahme weiter vorangeschritten werden kann.

3. Spannungen messen

In diesem Schritt sollen alle Spannungen gemessen und kontrolliert werden.

Ist eine aktive Spannungsquelle an den AmbiController angeschlossen so sind zwischen den folgenden Pads – beispielsweise mit dem Multimeter – die danebenstehenden Spannungen zu messen.

PAD7 (GND) – PAD2 = 3,3V

PAD7 (GND) – PAD4 = 2,5V

PAD7 (GND) – PAD5 = 1,8V

PAD7 (GND) – PAD6 = 1,2V

Stimmen die gemessenen Spannungen mit der Liste überein dann kann mit Schritt 4 fortgefahren werden.

4. Flashen des Bootloaders

Damit das System arbeiten kann muss zunächst ein Bootloader auf den Mikrocontroller installiert werden. Mittels dieses Bootloader ist es später möglich die Software des Xmega192A3 zu aktualisieren.

Zum flashen benötigt man einen oben genannten Programmieradapter/Debugger der per JTAG AVR XMeta Mikrocontroller flashen kann. Wie das Hex-File *Xmega_Bootloader.hex* geflasht wird kann dem Handbuch des Programmieradapters/Debuggers entnommen werden.

Ist der Bootloader auf dem AVR, startet dieser entweder automatisch, oder muss per Tastenkombination aktiviert werden.

Die beiden roten LEDs zeigen dabei an ob der Bootloader aktiv ist oder nicht. Leuchtet die „untere“ der beiden LEDs, welche weiter vom USB-Anschluss entfernt ist. Dann ist der Bootloader aktiv. Leuchtet keine LED ist keine Spannung angeschlossen oder keine Firmware auf dem Mikrocontroller vorhanden.

Um den Bootloader mittels Tastenkombination zu aktivieren hält man die Taste *FN* gedrückt und betätigt kurz den *Reset*-Taster. Daraufhin wird die eben beschriebene LED eingeschaltet.

ACHTUNG! Die LED wird nach einiger Zeit wieder ausgeschaltet. Ist eine Firmware auf dem XMeta installiert wird diese nach dem abschalten der LED geladen.

5. Flashen der Firmware

Damit der AmbiController seinen Dienst aufnehmen kann benötigt dieser neben dem Bootloader noch die Firmware. Auch diese liegt im „Release“-Ordner bei welche von Github kopiert werden kann.

Zum flashen der Firmware wird nun AVRDUDE mit den folgenden Einstellungen verwendet.

Linux:

```
$ ./rtbt /dev/ttyUSB?
```

```
$ avrdude -v -v -P /dev/ttyUSB? -b 115200 -p atxmega192a3 -c avr911 -U  
flash:w:AmbiController.hex:i -U eeprom:w:eeprom.hex:is
```

Windows (cygwin):

```
$ rtbt.exe COM?
```

```
$ avrdude -v -v -P COM? -b 115200 -p atxmega192a3 -c avr911 -U  
flash:w:AmbiController.hex:i -U eeprom:w:eeprom.hex:is
```

Drauf hin sollte man einmal den AmbiController resettet. Dazu kann der Reset-Button verwendet werden.

ACHTUNG! Während des Flashens sollten die beiden grünen LEDs beide abwechseln und rhythmisch aufleuchtet. Diejenige LED welche nicht ein einziges Mal aufgeleuchtet hat sollte auf Verpolung oder einen Defekt geprüft werden.

Nach einigen wenigen Sekunden sollte nun bei den roten LEDs Aktivität angezeigt werden.

Der Lange bzw. verzögerte Start kommt daher das der AmbiController noch nicht vollständig konfiguriert ist.

6. Flashen des Serial Flash

In diesem Schritt wird nun das Design des FPGA auf den Seriellen Flash übertragen. Hierzu wird bereits die Firmware auf dem AVR verwendet. Dazu gibt es ein Tool welches ebenfalls im Release-Ordner bereits vorkompiliert vorhanden ist.

Linux:

```
$ ./fpga_flash -p /dev/ttyUSB? -f main_ep4ce6.img -s -w
```

Windows (cygwin):

```
$ fpga_flash.exe -p com? -f main_ep4ce6.img -s -w
```

Wie bereits beim Flashen der Firmware dauert es eine Weile bis die Daten übertragen wurden.

Der AVR wird nach der Übertragung automatisch resettet, was durch beobachten der LEDs beobachtet werden kann. Der Start ist nun deutlich schneller als zuvor.

Achtung! Ist ein aktives Eingangssignal am Cinch-Anschluss angelegt ist kein Update des Serial-Flash möglich.

7. Channel Configuration

Damit der AmbiController die vorhandenen WS2811 – LED – Streifen korrekt ansteuern kann benötigt dieser Informationen wie viele LEDs an welcher Stelle des Bildschirms angebracht sind. Dies wird durch das Schreiben der entsprechenden Channel Configuration durchgeführt.

Hierzu gibt es bereits einige Dateien welche verwendet werden können. Hat man spezielle Wünsche kann man das „Channel config – Tool“ verwenden um eine neue Konfigurationsdatei zu erstellen.

Die Konfiguration kann mit dem „bctool“ aus dem „Release“-Ordner übertragen werden.

Linux:

```
$ bctool -p /dev/ttyUSB? -U -L config_file_name.cfg
```

Windows (cygwin):

```
$ bctool -p com? -U -L config_file_name.cfg
```

8. Erweiterter Software-/System-Test

Nachdem nun mit den letzten Schritten die Software und der FPGA in Betrieb genommen wurde, kann das Verhalten des Systems und der Software geprüft werden.

Hierzu können mit dem BCTool beispielsweise weitere Daten abgerufen werden.

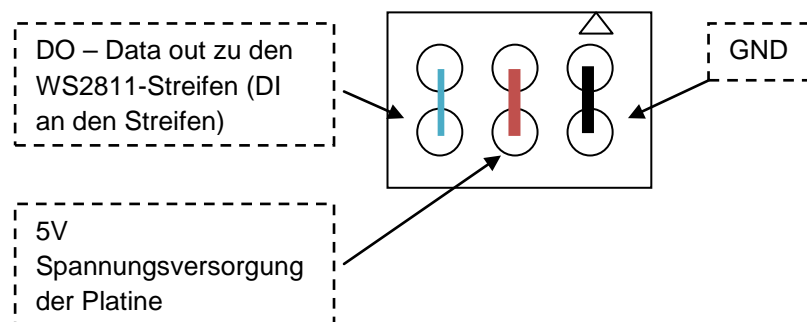
Die Kommandozeilen-Hilfe des BCTool beschreibt welche Einstellungen angepasst werden dürfen und welche Informationen aus dem System ausgelesen werden können.

Hier sollte geprüft werden – nachdem ein CVBS-Signal angeschlossen wurde – ob der Aufruf der Option „-S“ ausgibt das das System ein Signal erkannt hat. Ein weiterer Indikator für ein aktives Eingangssignal ist zudem das aufleuchten der rechten gelben LED. Die linke gelbe LED leuchtet auf der v0.4.1 Hardware in der Regel nicht auf.

9. Steckverbindung zu den LED-Streifen

Beim AmbiController werden die LED-Streifen mit einer 6-Pin Pfostenbuchse verbunden. In diese Buchse ist ein übliches 6-Pin Flachbandkabel eingelegt.

Die Belegung kann der folgenden Skizze entnommen werden. Das Dreieck symbolisiert *Pin 1* bei Draufsicht auf den Anschluss auf der Platine. Die Senkrechten Markierungen zeigen welche der Pins durch die Platine miteinander verbunden sind. Die Pinbelegung der drei Pin-Paare ist der Skizze zu entnehmen.



Das Flachbandkabel der Pfostenbuchse wird direkt an die LED-Streifen angelötet. Achtung, die Platine sollte die LED-Streifen nicht mit Strom aus der USB-Buchse versorgen. Dies kann

mit wenigen LEDs gut gehen, führt aber zu einem durch den fließenden hohen Strom zu einer hohen Wärme-/Hitze-Entwicklung der die Platine und den USB-Port zerstören kann.

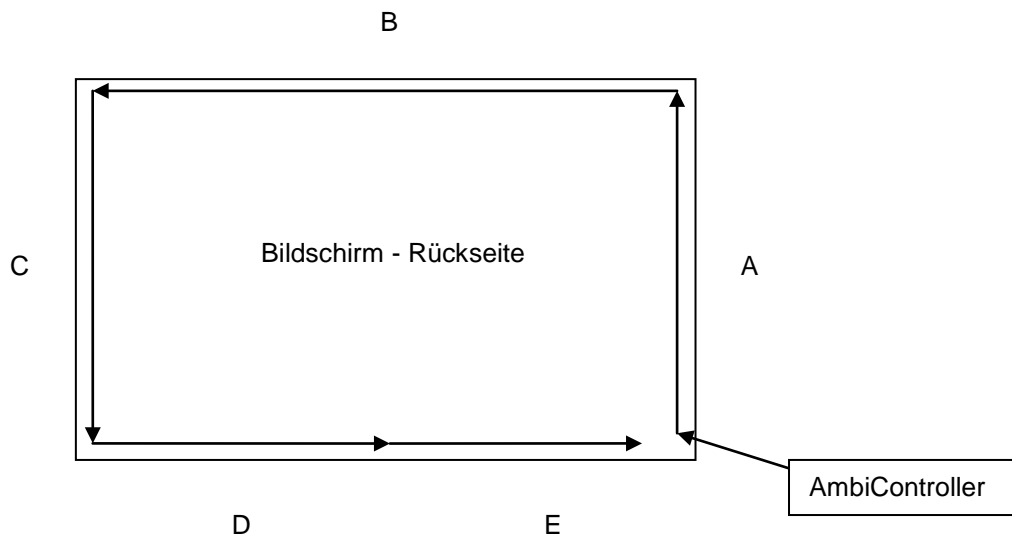
Im Idealfall besitzt der AmbiController eine Spannungsquelle welche per USB angeschlossen ist und die LED-Streifen eine eigene welche direkt an diesen angeschlossen ist.

10. Aufbau der LED-Streifen

Um die LED-Streifen kompatibel mit dem AmbiController an den Bildschirm anzubringen müssen die folgenden Notwendigkeiten beachtet werden.

1. Der zu verwendete LED-Streifen wird in 5 Segmente unterteilt (A,B,C,D,E).
2. Alle folgenden Punkte beziehen sich darauf das man auf den Rücken des Bildschirms sieht.
3. Zwischen allen nebeneinander liegenden Segmenten müssen GND und 5V jeweils verbunden werden, so dass sich ein Ring bildet.
4. Segment A befindet sich auf der rechten Seite, B oben, C links, D links unten und E rechts unten
5. Der AmbiController wird mit Segment A an der unteren rechten Ecke des Bildschirms verbunden (Pin DI am LED-Streifen).
6. Segment A wird an der oberen rechten Ecke des Bildschirms (DO) mit Segment B (DI) verbunden.
7. Segment B (DO) an der linken oberen Ecke mit Segment C (DI).
8. Segment C (DO) an der linken unteren Ecke mit Segment D (DI).
9. Segment D (DO) an in der Mitte der unteren Kante mit Segment E (DI). Diese beiden Segmente wurden eingeführt falls der Bildschirm einen Sockel besitzt der bei der Berechnung ausgelassen werden muss.
10. Das Ende von Segment E verbleibt unverbunden, abgesehen von GND und 5V
11. Bei längeren Streifen sollte nach einer aus dem Handbuch/Datenblatt zu entnehmenden Länge erneut eine 5V Spannung angelegt werden.

Die folgende Skizze zeigt den Aufbau/Umlauf der LED-Streifen schematisch.



Sind die LEDs am Bildschirm auf diese Weise angebracht und miteinander verbunden worden, können diese an den AmbiController angeschlossen werden.

Wird dieser nun eingeschaltet wird zunächst die mittels *BCTool* eingestellte Standard-Farbe auf den LEDs eingestellt.

Hat der AmbiController am Cinch-Anschluss ein Eingangssignal erkannt beginnt dieser automatisch mit der Verarbeitung und Ausgabe der Farben.

11. Abschluss

Ich wünsche viel Spaß und Freude an Ihrem nun vollständig aufgebauten AmbiController.

Bei Fragen und auch für Anregungen stehen ich gerne zur per Email an klaute@web.de zur Verfügung.