

Technische Dokumentation

für die Elektronikbaugruppe

Mikrocontroller-Kopplung per USB MKU (RL)

Version 1.11

Dateiname: BA19-MKU.ZIP Ausgabe 01.11.2007

Inhaltsverzeichnis

	Seite
1 Beschreibung der Baugruppe	3
2 Stromlaufplan	3
2.1 Aufbau	3
2.2 Beschreibung	4
3 Beispiel für eine unkomplizierte PC-Kopplung	4
3.1 Übersichtsplan	4
3.2 Anschlußschemas	4
3.2.1 Testschaltung	4
3.2.2 MC-Kopplung	5
3.3 Softwarevoraussetzungen	5
3.3.1 Treiber für den USB-Adapter	5
3.3.2 Software für MC und PC	6
3.4 Programmiermodul	6
4 Steuersoftware und Programmierung	6
5 Beschreibung der Anschlüsse	8
6 Allgemeines	9
6.1 Hinweise zum PC-Anschluß	9
6.2 Literaturverzeichnis	9
6.3 Dateienverzeichnis	10
6.4 Elektronische Bauelemente	10
6.5 Hinweise zu den Nutzungsrechten	10
6.6 Technische Hinweise	12
6.6.1 Ausdrucken der Dateien	12
6.6.2 Anfertigung von Leiterplatten	12
6.6.3 Gehäuse und Frontplattenbeschriftungen	12
6.6.4 Bauelementeauswahl	12
6.6.5 SMD-Bestückung	12

1 Beschreibung der Baugruppe

Früher oder später kommt man beim Einsatz von Mikrocontrollern (MC) zu der Erkenntnis, daß eine PC-Kopplung eine wünschenswerte Erweiterung darstellen würde. Zum Beispiel würde sich die Auswertung eines Datenerfassungssystems am PC bedeutend einfacher und ansprechender gestalten lassen als auf einem LCD-Modul und die Speichergrenzen für die Datenspeicherung eines MCs könnten mühelos überschritten werden.

Die Zeiten, in der die parallelen und seriellen Schnittstellen noch zu den Standardinterfacen von PC und Notebook zählten, sind mittlerweile Geschichte. Die damaligen Entwicklungen mit I/O-Expandern der Typen 82C43 und 82C55 [11] könnten zwar für die MC-Kopplung genutzt werden, aber bald werden auch keine PC-Karten mehr zur Verfügung stehen, da die Gegenwart **Universal Serial Bus** (USB) heißt.

Die Nutzung dieses Bussystems ist sehr einfach, wenn man Stick und Co. mit dem PC verbindet, gestaltet sich aber um ein Vielfaches komplizierter, wenn die eigene Hardware mit einem PC verbunden werden soll. Aber warum sollte man als Amateur auf die Vorteile (einheitlicher Standard, Plug & Play, etc.) verzichten? Bei Recherchen zum Thema stößt man jedoch schnell auf unausweichliche Anforderungen (z.B. Treiber), die die Anwendung der USB-Schnittstelle mit sich bringt. Den Leser möchte ich hier nicht mit den vielen Details, die es zu beachten gibt, konfrontieren. Für Interessenten gibt es im Internet zum Thema entsprechendes Material; hier kommentiert würde es den Rahmen des Beitrages sprengen. Mittlerweile haben einige Firmen dies erkannt und an der ständigen Weiterentwicklung eigenständiger USB-Controller gearbeitet.

Bei der Recherche zum Thema ist mir u. a. der Artikel von Roland Walter [3] aufgefallen. Der im Beitrag vorgestellte USB-Controller einschließlich einer Beispielschaltung unterstützen meiner Meinung nach optimal die Möglichkeiten, die einem Amateur zur Verfügung stehen. Frei verfügbare Treiber für alle gängigen Betriebssysteme und ein Preis (Fa. Reichelt, ca. 6 €), der die Haushaltskasse nicht zu stark strapaziert, waren die Gründe, sich intensiver mit den Produkten der Firma Future Technology Devices (FTDI) [1] zu beschäftigen.

Nach Aufbau einer Testschaltung mit dem im Artikel erwähnten FDTI232BL, der eine Weiterentwicklung gegenüber dem FDTI232AM darstellt, kamen die Controller der Familie FDTI232Rx auf den Markt, die weitere Einsparungen an externen Bauelementen (z. B. kein Quarz, EEPROM und Schutzwiderstände der USB-Datenleitungen mehr) und die Möglichkeit der Programmierung von diversen Funktionen mit sich brachten. Ein Blick in das Datenblatt verdeutlicht die Vorteile gegenüber den Vorgängern. Interessenten, die noch im Besitz von Schaltkreisen der Reihe FDTI232Bx sind, können per Email den Schaltplan der geprüften Testschaltung anfordern. Auf die Entwicklung einer Leiterplatte wurde auf Grund der guten Verfügbarkeit des weiterentwickelten Types jedoch verzichtet.

Wenn man sich mit dem Datenblatt einschließlich der Applikationen des Herstellers zur aktuellen Version des USB-Controllers beschäftigt, wird einem auffallen, daß es große Ähnlichkeiten gibt. Warum also in Anlehnung an die Herstellervorschläge nicht eine Schaltung von einem USB-Adapter entwickeln, der per Jumper (oder Drahtbrücken) viele der vorgestellten Einsatzfälle abdecken kann? Entstanden ist ein Modul, daß mit seinen Leiterplattenabmessungen von 52 x 35 mm² auch nachträglich ohne größeren Aufwand integriert werden kann.

Eine Hürde, die es zu meistern gibt, wenn man sich für den FDTI232RL oder FDTI232RQ entscheidet, muß aber noch erwähnt werden. Bei den Schaltkreisen handelt es sich nämlich um Silizium, was in einem total amateurfeindlichen Gehäuse (SSOP-28 bzw. QFN-32) untergebracht ist. Nur mit ruhiger Hand, einer starken Lupe und Geduld bekommt man mit einem Lötkolben den IC aufgelötet. Einige Techniken für das Einlöten von SMD-Bauelementen sind im Internet beschrieben.

Eine Alternative wären zum Beispiel die komplett aufgebauten Evaluation-Kits und DIP-Module (z. B. EVAL232R, MM232R, UM232R, u.v.a.) von FDTI [1], die in reiner SMD-Technik gefertigt werden; Preise, Bezugsmöglichkeiten und -bedingungen sind mir bisher nicht bekannt.

2 Stromlaufplan

2.1 Aufbau

Aus dem FTDI-Sortiment von USB-Controllern, das neben Typen für eine seriellen auch die parallele Signalverarbeitung ermöglicht (z. B. FTDI245RL/RQ mit parallelen FIFO-Interface), wurde für den Aufbau des USB-Adapters der FTDI232RL ausgewählt. Bei ihm handelt es sich um einen Schaltkreis, der den Aufbau seriell arbeitender Schaltungen unterstützt und, da die meisten MC der Firma Atmel [4] eine Hardware-UART (**Universal Asynchronous Receiver Transmitter**: asynchrone Schnittstelle zur Datenübertragung mittels Sen-

der-Empfänger) besitzen, eine einfache Kopplung ermöglicht.

Der Anschluß an den PC erfolgt laut FDTI-Empfehlung über eine USB-Buchse Typ „B“. Für die Kopplung mit der anzuschließenden Schaltung stehen zwei parallel verlaufende 14-polige Adapterleisten zur Verfügung, über die alle I/O-Kanäle des FDTI232RL anschließbar sind. Verwendet man auf der anzuschließenden Schaltung Kontaktbuchsen mit Präzisionskontakten, kann bei Bedarf der USB-Adapter auch für eine andere Aufgabe verwendet werden. Natürlich kann der USB-Adapter auch gleich in das Layout eingebunden werden.

2.2 Beschreibung

Da die Baugruppe im wesentlichen aus dem USB-Controller FDTI232RL und Steckverbindern aufgebaut ist, ist eine ausführliche Schaltungsbeschreibung nicht erforderlich. Die Werte der eingesetzten Kondensatoren entsprechen den Empfehlungen des Herstellers. Die Vorwiderstände für die LED wurden bewußt relativ hoch gewählt und können ggf. reduziert werden (Datenblatt beachten). Wer keine Anzeigen benötigt, kann sie auch komplett bei der Bestückung weglassen.

VD1 und VD2 zeigen das Vorhandensein der Betriebsspannungen an, die entweder komplett aus der USB-Schnittstelle (VD1; Modus: bus powered) oder aus der angeschlossenen Schaltung (VD2; Modus: self powered) entnommen werden kann. In diesem Fall ist auch ein Betrieb mit einer Spannung von 3,3V möglich.

Mit Jumpern auf den Stiftleisten X12/X13 kann der jeweilige Modus festgelegt werden.

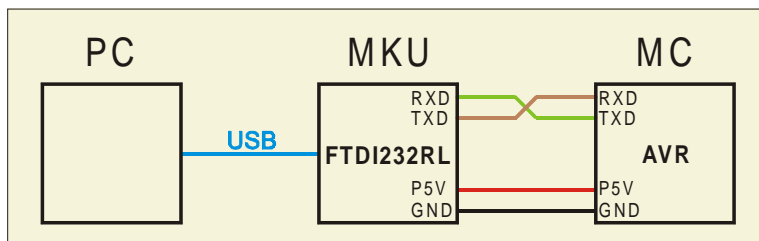
Für die Art und Weise des Signalanschlusses /RESET dienen die Stiftleisten X10/X11. Weitergehende Hinweise können den Applikationen des Herstellers entnommen werden.

VD3 und VD4 sind als Anzeige für den Datenverkehr vorgesehen. Sie sind nicht direkt mit dem FDTI232RL verbunden, da nur im Auslieferungszustand die Kanäle CBUS0 bzw. CBUS1 (TXD, RXD) für die Anzeigefunktionen konfiguriert sind.

3 Beispiel für eine unkomplizierte PC-Kopplung

3.1 Übersichtsplan

Wie einfach sich die PC-Kopplung an einen MC gestalten kann, zeigt folgende Abbildung:



Die Kopplung des USB-Adapters (MKU) kann direkt oder über einen USB-Hub erfolgen. Eine USB-Schnittstelle liefert ohne Weiteres erst einmal maximal 100 mA. Unter bestimmten Voraussetzungen (z. B. self-powered Hub, spezieller USB-Anschluß eines PCs) können auch bis zu 500 mA entnommen werden. Entsprechende Einstellungen können im EEprom, der sich im FDTI232RL befindet, vorgenommen werden. Zu beachten ist die Kreuzung der RXD/TXD-Leitungen.

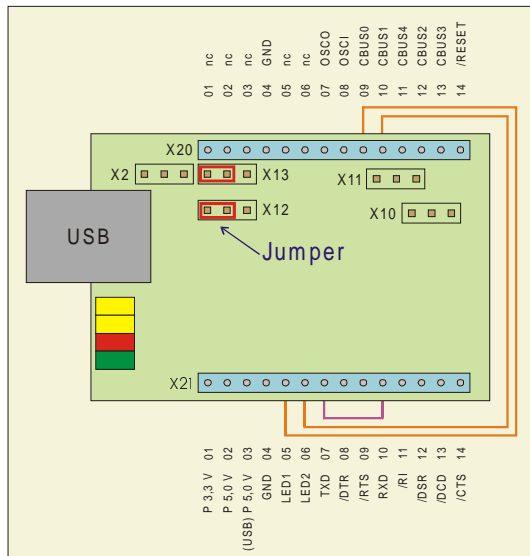
An dieser Stelle soll auch auf die Möglichkeit hingewiesen werden, daß ein Port des FDTI232RL als Takt- ausgang (6, 12, 24, 48 [MHz]) konfigurierbar ist, der dem MC dann zur Verfügung gestellt werden kann.

3.2 Anschlußschemas

3.2.1 Testschaltung

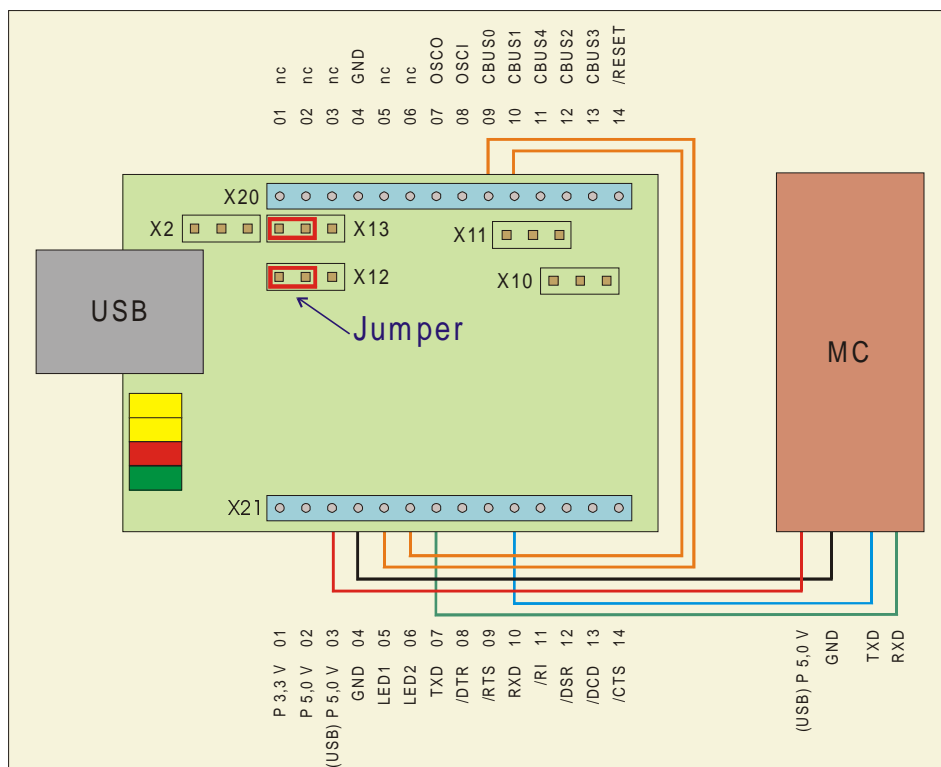
Durch die Realisierung von nur drei Verbindungen kann man überprüfen, ob die Schaltung funktioniert. Dazu sind zwei Jumper so gesteckt, daß die Stromversorgung aus der USB-Schnittstelle entnommen wird. Dies entspricht auch der Basis-Konfiguration des FDTI232RL.

Durch ein Terminalprogramm gesendete Daten werden direkt wieder zurückgeschickt (Verbindung von RXD mit TXD; X21:07-X21:10).



3.2.2 MC-Kopplung

In der Basiskonfiguration des FTDI232RL ist eine maximale Stromentnahme von 90 mA eingestellt (internes EEPROM). Damit kann auch eine einfache MC-Schaltung versorgt werden. Die Jumperbestückung entspricht daher der, die schon bei der Testschaltung zum Einsatz gekommen ist.



3.3 Softwarevoraussetzungen

3.3.1 Treiber für den USB-Adapter

Da der Betrieb an einer USB-Schnittstelle nicht ohne einen Treiber möglich ist, muß dieser zunächst installiert werden. Dazu können vom Hersteller Installations- und Einsatzhinweise als PDF-Datei unter Beachtung des eingesetzten Betriebssystems bezogen werden.

Wie bereits oben erwähnt, stellt die Firma FDTI für die meisten PC-Betriebssysteme zwei Arten von Treibern zur Verfügung, wobei die Art der verwendeten Programmiersprache von untergeordneter Rolle ist. **Wichtig: Es kann immer nur ein Treibertyp installiert sein!**

Beim ersten Typ handelt es sich um einen **Virtuellen COM Porttreiber (VCP)**. Ist dieser installiert, kann zwar nicht auf die Daten des EEPROMs im FDTI232Rx zugegriffen werden, aber ein Anwendungsprogramm „merkt“ nicht, daß es sich um eine USB-Schnittstelle handelt, sondern behandelt die Schnittstelle nach dem RS232-Standard, wobei auch höhere Datentransferraten möglich sind. Die diversen Port-DLL für eine serielle Datenübertragung, die im Internet teils kostenfrei zur Verfügung stehen, können daher ohne Probleme zum Einsatz kommen.

Beim zweiten Typ handelt es sich um einen sogenannten „Direkttreiber“ (D2XX). Das Anwendungsprogramm greift in diesem Fall über DLL-Funktionen auf den USB-Controller zu. Dieser Treiber ist wesentlich kleiner, erfordert aber, daß man sich mit den Funktionen zunächst vertraut machen muß.

Für einen ersten Funktionstest können die RXD/TXD-Leitungen kurzgeschlossen werden (Abschnitt 3.2.1). Dazu kann das unter BASCOM [5] verfügbare Terminalprogramm verwendet werden.

Etwas komfortabler, aber kostenfrei leider nur zeitlich begrenzt nutzbar, ist das Terminalprogramm ZOC [8].

3.3.2 Software für MC und PC

Ein bescheidenes Programm für den AVR ist dieser Dokumentation beigelegt. Es erwartet auf der Schnittstelle einen definierten Zeichencode und sendet daraufhin eine bestimmte Anzahl von Daten an den PC.

Auf dem PC muß ein Programm laufen, daß diesen Anforderungscode senden und die Daten empfangen kann.

Am Beispiel von Excel wird gezeigt, wie komfortabel die Daten mit einer Tabellenkalkulation über den virtuellen COM-Port eingelesen werden können. Sind sie erst einmal in Zeilen und Spalten eingetragen, kann die volle Funktionalität von Excel für die Auswertung der Daten verwendet werden.

Da Excel von Haus aus keine Möglichkeit besitzt, Daten über eine serielle Schnittstelle einlesen zu können, gibt es über die Windows API eine Variante den Zugriff zu ermöglichen. Vorzuziehen ist aber auch hier der Einsatz einer DLL (Geschwindigkeitsvorteile). Man kann sich diese DLL selbst schreiben, oder aber frei verfügbare einsetzen.

Es gibt eine große Anzahl entsprechender Lösungen. Für die Einbindung in Excel erschien die RSAPI.dll [10] als eine günstige Lösung, da die Anzahl der Funktionen überschaubar und eine deutsche Beschreibung verfügbar ist.

Kopiert man die DLL im Windowsverzeichnis und stattet eine Excel-Arbeitsmappe mit dem beiliegenden Makro aus, steht der Datenübertragung zwischen PC und AVR nichts mehr im Weg.

3.4 Programmiermodul

Soll der MC außerhalb der Schaltung programmiert werden, kann jedes Programmiergerät verwendet werden, daß die Möglichkeit besitzt, diesen Schaltkreistyp zu beschreiben.

Möchte man eine neuen Software aufspielen, ohne den MC aus der Schaltung zu entfernen (in-vivo-Programmierung), eignen sich nur solche Geräte, die die serielle Programmierung mit den Signalen Reset, MISO, MOSI und SCK unterstützen, da diese zusammen mit der Schaltungsmasse auf den Programmieran-schluß gelegt sind.

Dieser Steckverbinder ist so ausgelegt, daß über ihn direkt das Programmiermodul PRM [11] angeschlossen werden kann. Dieses bezieht seine Stromversorgung aus dem Modul und wird über ein 25-poliges Verbindungskabel mit der parallelen Schnittstelle eines PCs verbunden.

4 Steuersoftware und Programmierung

Zur Erstellung der Software gibt es die verschiedensten Möglichkeiten, da mittlerweile für die AVR Entwicklungstools für C, Basic, Assembler und anderen Programmiersprachen unter Windows oder Linux verfügbar sind. Für jemanden, dem die Programmiersprache C geläufig ist, sollte sich zum Beispiel einmal WinAVR ansehen.

Etwas einfacher läuft es mit BASCOM, dessen Testversion zwar nur maximal 4 kByte Programmcode zuläßt, für einfache Steuerungsabläufe ist diese Größe sicherlich ausreichend. Mit BASCOM ist es zum Beispiel auch möglich, vor dem Compilieren des in einem Basicdialekt verfaßten Quellcodes eine bestimmte LCD-Anzeige auszuwählen. Ein Beispiel für den geringen Aufwand zur Programmierung kann dem beiliegenden

Quelltext entnommen werden. An dieser Stelle soll erwähnt werden, daß sich bei BASCOM in den Quelltext auch Programmteile in Assemblercode einbinden lassen.

Genannt werden soll auch die Möglichkeit der Programmierung in Assembler. Vielleicht kann damit Quellcode erzeugt werden, der in Bezug auf die Codelänge eine optimalere Lösung darstellt, als bei Verwendung einer Hochsprache. Jedoch sollte man den Aufwand für die Erstellung des Quelltextes nicht unterschätzen.

Hat man seinen Quellcode geschrieben, kann es ans Testen gehen. Einerseits bieten Simulatoren die Möglichkeit, dies ohne den Aufbau einer Testschaltung durchzuführen. Andererseits gibt es bei den Simulatoren auch einige kritische Dinge, wo sie einfach überfordert sind und eine Anbindung an die reale Umgebung unerläßlich ist, als Beispiele soll hier der Timer- und Interruptbetrieb genannt werden. Laut Datenblatt ist zwar die Anzahl der möglichen Programmierzyklen eines MCs begrenzt, aber ob man jemals im Hobbybereich die angegebenen Grenzen überschreitet, möchte ich bezweifeln.

Da zur Zeit beim Autor (noch) keine Erfahrungen der Programmierung mit C oder Assembler vorhanden sind, können hier auch keine Erkenntnisse zur Programmierung der AVR's mit den dazugehörigen Entwicklungstools vermittelt werden. Informationen dazu finden sich sicherlich in den einzelnen Programmbeschreibungen und dem Internet [6]. Die weiteren Hinweise beziehen sich daher alle auf den Einsatz von BASCOM.

Leider gibt es für BASCOM keine deutsche Bedienoberfläche. Mit ein wenig Ergeiz lernt man aber schnell, das Programm zu bedienen, da dem Entwicklungstool eine umfangreiche Hilfefunktion beigelegt ist. Neben den Menüs für die Einstellung verschiedener LCD-Anzeigen, Prozessortypen und anderen Dingen muß man sich auch für eine interne oder externe Software zum Brennen der MC entscheiden. Fast jede Software erfordert einen speziellen Adapter. Man muß sich also schon überlegen, auf welche Art und Weise man seine Steuerung programmieren möchte. Ebenso ist jeder besserer Eprombrenner in der Lage, AVR zu beschreiben.

An dieser Stelle soll kurz die Brennsoftware TwinAvr vorgestellt werden. Dabei handelt es sich um ein Freewareprogramm [1], was seit einiger Zeit auch in der Lage ist, die AVR der MEGA-Serie beschreiben zu können. Mit einigen Einstellungen ist es unter BASCOM direkt verfügbar.

5 Beschreibung der Anschlüsse

Folgende Übersicht erleichtert den Anschluß des Moduls:

- USB-Anschluß

X01	Funktion	Erläuterung
1	P5 V	USB-Spannung P5 V
2	DM	Datenleitung -
3	DP	Datenleitung +
4	GND	Masse

- Anschluß

X20	Funktion	Erläuterung
1	P3,3 V	Spannungsversorgung P3,3 V
2	P5,0 V	Spannungsversorgung P5,0 V
3	P5,0 V	Spannungsversorgung P5,0 V (USB)
4	GND	Masse
5	LED1	Anzeige 1
6	LED2	Anzeige 2
7	TXD	FDTI232RL-Kanal
8	/DTR	FDTI232RL-Kanal
9	/RTS	FDTI232RL-Kanal
10	RXD	FDTI232RL-Kanal
11	/RI	FDTI232RL-Kanal
12	/DSR	FDTI232RL-Kanal
13	/DCD	FDTI232RL-Kanal
14	/CTS	FDTI232RL-Kanal

X21	Funktion	Erläuterung
1	nc	-
2	nc	-
3	nc	-
4	GND	Masse
5	nc	-
6	nc	-
7	OSCO	FDTI232RL-Kanal
8	OSCI	FDTI232RL-Kanal
9	CBUS0	FDTI232RL-Kanal
10	CBUS1	FDTI232RL-Kanal
11	CBUS4	FDTI232RL-Kanal
12	CBUS2	FDTI232RL-Kanal
13	CBUS3	FDTI232RL-Kanal
14	/RESET	FDTI232RL-Kanal

6 Allgemeines

6.1 Hinweise zum PC-Anschluß

Bei sachgerechter Ausführung und Prüfung der Baugruppe vor Anschluß an die USB-Schnittstelle eines PCs gibt es keine Probleme.

Ein Ausbilder meinte einmal: „Eine Schaltung, die auf Anhieb funktioniert, ist fehlerbehaftet!“ Nun, dies stimmt selbstverständlich in vielen Fällen nicht. Man ist aber trotzdem gut beraten, gewisse Vorsichtsmaßnahmen zu ergreifen (z. B. eine ausgiebige Prüfung der Schaltung vor dem Anschluß).

6.2 Literaturverzeichnis

- [1] Fa. FDTI: Future Technology Devices (USB-Schnittstellen-IC)
<http://www.fdtichip.com>
- [2] Walter, R.: Thema MC (AVR, Aufbau und Programmierung mit TwinAvr)
<http://www.rowalt.de/>
bzw. Beitragsreihe im Funkamateure 2002/ 2003
- [3] Walter, R.: Thema AVR und USB
<http://www.rowalt.de/>
Funkamateure 3/2003, Seite 253
- [4] Fa. ATMEL: Datenblatt von Atmel für den ATmega16
<http://www.atmel.com/>
ATMEL 8-bit with 16K Bytes In-System Programmable Flash
- [5] Fa. McSelec: Entwicklungsumgebung BASCOM
<http://www.mcselec.com/>
- [6] Fa. Reichelt: Bezug von Bauelementen
<http://www.reichelt.de/>
- [7] (ohne): Infos zu MC
<http://www.mikrocontroller.net/>
- [8] Fa. EmTec: Terminalprogramm ZOC/Pro 5.08
<http://www.emtec.com>
- [9] (ohne): Arduino - USB Board (Homepage)
[http:// www.arduino.cc](http://www.arduino.cc)

Arduino - deutsche Bauanleitung
http://incom.org/code/projekte/projekt_anzeigen.php?4,98,0,0,0,102
- [10] H.-J. Berndt: RS232-Application-Programming-Interface – RSAPI (RSAPI.DLL)
H.-J. Berndt / B. Kainka
Messen, Steuern und Regeln mit Word und Excel
VBA-Makros für die serielle Schnittstelle, 2., neubearb. und erw. Aufl.
Franzis-Verlag GmbH, 85586 Poing, 1999 ISBN 3-7723-4093-8
Franzis-Verlag: <http://www.franzis.de/> Suchstichwort "BERNDT"

Hommage (RSAPI.DLL)
<http://www.b-kainka.de/msrwefaq.htm>
- [11] Bauer, I.: Funkuhr- modular aufgebaut
siehe Homepage
bzw. Funkamateure 12/2004, Seite 1238

Ein- und Ausgabemodul EAP (I/O-Baustein 82C43)

siehe Homepage
bzw. Funkamateure 7/1999, Seite 763

Modulares Bussystem MBS (I/O-Baustein 82C55)
siehe Homepage
bzw. Funkamateure 2/2000, Seite 158

6.3 Dateienverzeichnis

Für das Arbeiten mit der Dokumentation können der kostenfreie Adobe- oder Foxit-Reader verwendet werden, da alle Dateien im *.pdf Format veröffentlicht sind:

Dateien der Dokumentation

01. MKU (001 Kurzinformation).pdf	Vorstellung der Elektronikbaugruppe
02. MKU (002 Dokumentation).pdf	Dokumentation (geöffnet)
03. MKU (101 Stromlaufplan).pdf	
04. MKU (102 Stückliste).pdf	
05. MKU (103 Bestückungsplan).pdf	
06. MKU (104 LP-Lötseite).pdf	Leiterplatte LS
07. MKU (105 LP-Bestückungsseite).pdf	Leiterplatte BS

Programmbeispiel (Quelltext)

08. MKU_MC (V 1.0).bas	Quellcode des Programms für den MC
09. MKU_PC (V 1.0).bas	Quellcode (Makro) für Excel

6.4 Elektronische Bauelemente

Für Elektronikamateure, die in Ihrer Nähe keinen Fachhandel für elektronische Bauteile haben oder denen die Bestellung, zum Beispiel beim Elektronikversand Reichelt (<http://www.reichelt.de/>) für ein oder zwei Fehlerteile zu aufwendig sein sollte, können wir (nur innerhalb von Deutschland) eventuell Hilfe bei der Beschaffung anbieten.

Das Angebot an Leiterplatten ist abhängig von der Nachfrage und der jeweiligen Baugruppe. Die Leiterplatten sind ein- oder zweiseitig, gebohrt und verzinkt, aber nicht durchkontaktiert.

6.5 Hinweise zu den Nutzungsrechten

Die Bauanleitungen (einschließlich eventueller Software) darf unter den folgenden Bedingungen frei kopiert oder weitergegeben werden:

- es darf kein Preis für die Dokumentation (*.zip) erhoben werden, außer einer angemessenen Kopiergebühr
- für einen kommerziellen Vertrieb der Baugruppen ist das Einverständnis der Autoren erforderlich
- die Verwendung von Teilen der Dokumentation in eigenen Publikationen ist erlaubt, sofern ein eindeutiger Hinweis auf die Quelle erfolgt
- eine eventuell der Baugruppe beigelegte Software kann als Shareware konzipiert sein; dann ist sie zeitlich nicht limitiert, besitzt jedoch nur in der registrierten Version ihren vollen Funktionsumfang

Die Registrierung der Software ermöglicht Ihnen, Hinweise über die neueste (eventuell fehlerbehebene) Version zu erhalten. Wir haben uns bemüht, fehlerfrei zu programmieren. Die Erfahrung mit vielen Programmen zeigt aber, daß Software der „Version 1.0“ selten fehlerfrei funktioniert.

Hardwarefehler sind auf Grund der Anfertigung von Mustergeräten weitestgehend ausgeschlossen; Maßabweichungen bei der Bestückung können durch verschiedene Bauformen der Bauelemente auftreten.

Sollten Sie glauben, einen Fehler gefunden zu haben, senden Sie uns bitte per Email folgendes zu:



- eine kurze, aber präzise Beschreibung für die Reproduzierbarkeit des Fehlers
- eine Beschreibung der eingesetzten Hard- und Software

Ohne diese Informationen ist eine Bearbeitung nicht möglich.

Einige Programme laufen unter Visual Basic (VB) der Versionen 4/ 5. Die jeweilige *.exe funktioniert daher nur unter Windows, wenn VB oder ein zusätzliches Softwarepaket mit den erforderlichen Dateien installiert ist.

Anwender, die darüber nicht verfügen, können sich eine Übersicht über die Funktionalität des Programms verschaffen. Das zusammengestellte Material beschreibt in Kurzform die Funktionalität der Software.

Da die entsprechenden Installationsroutinen für Anwender ohne VB recht groß sind (ca. 1 ... 5 MB), macht eine Verteilung über Online-Dienste kaum einen Sinn. In diesem Fall sollte per Email unter Angabe des Elektronikmoduls ein Informationsblatt angefordert werden.

In der Zukunft entstehende Programme werden mit der Programmiersprache PROFAN entwickelt, der Grund dafür hat mehrere Ursachen.

Für Baugruppen, in denen Mikrocontroller (AVR von Atmel) eingesetzt werden, dient als Programmiersprache BASCOM (Basicdialekt). BASCOM ist bis zu einer Programmgröße von 4 kByte ohne Einschränkungen verwendbar und unserer Meinung nach recht schnell zu beherrschen.

6.6 Technische Hinweise

6.6.1 Ausdrucken der Dateien

Der Ausdruck der Dateien sollte keine Probleme bereiten.

Im Gegensatz zu den Frontplattenbeschriftungen, die im Maßstab 1:1 vorliegen, sind die Leiterplattenzeichnungen der ersten Module im Maßstab 2:1 erstellt, was bei der Fertigung beachtet werden muß.

Für Leiterplatten jüngerer Datums wurde TARGET als Layouter verwendet. Nähere Informationen dazu auf der Homepage.

6.6.2 Anfertigung von Leiterplatten

Bei der Erstellung des Layouts der Leiterplatten wurde auf eine einfache Leitungsführung geachtet, so daß auch manuell gezeichnete Leiterplatten kaum Probleme bereiten.

6.6.3 Gehäuse und Frontplattenbeschriftungen

Sofern bei einer Baugruppe ein größeres Gehäuse verwendet wurde, handelt es sich um ein Kunststoff-Halbschalen-Gehäuse, was zum Beispiel über den CONRAD-Elektronikversand bezogen werden kann.

Diese Gehäuseform hat den Vorteil, daß mit geringem Aufwand eine ansprechende Frontplatte gestaltet werden kann. Die den jeweiligen Baugruppen beigelegten Datei „* (Frontplatte).pdf“ beinhaltet jeweils einen Vorschlag. Sie können auf entsprechenden Papier (nicht zu dickes verwenden) ausgedruckt und unter Verwendung eines Klebestiftes auf die Frontplatte geklebt werden. Nach dem Trocknen kann noch eine selbstklebende Transparentfolie aufgeklebt werden. Mit einer Rasierklinge lassen sich jetzt überstehende Papier- und Folienreste schnell entfernen. Entsprechend den Konturen kann nun gekörnt und vorhandene Ausschnitte mit einer Laubsäge angefertigt werden. Nach Montage der Bedienelemente und der Verdrahtung mit der Leiterplatte kann die Montage des Gehäuses durchgeführt werden.

Stecker- oder Hutschienengehäuse können vom gleichen Anbieter, Reichelt-Elektronik oder direkt vom Hersteller (BOPLA) bezogen werden.

6.6.4 Bauelementeauswahl

Bauelemente ohne besonderen Hinweis sind als Vorschläge zu verstehen. An diesen Stellen können natürlich Typen verschiedener Hersteller zum Einsatz kommen, sofern sich ihre elektrischen Parameter gleichen.

Weitergehende Erläuterungen und Hinweise zum Aufbau elektronischer Schaltungen werden als bekannt vorausgesetzt bzw. können aus entsprechenden Literaturquellen bezogen werden.

6.6.5 SMD-Bestückung

Leider gibt es einige Bauelemente, die nicht in einem DIL-Gehäuse angeboten werden. Die Bestückung der Leiterplatte gestaltet sich daher komplizierter. Im Internet findet man zum Thema auch mehrere Anregungen und Tipps.

Zur Bestückung mit dem FTDI232RL wurde die LötKolbenspitze durch einen angespitzten 1,5 mm² Kupferdraht ersetzt. Die Leistung des LötKolbens mit 30 Watt ermöglicht ein gutes Fließen des Lötzinns. Eine Lupe, ausreichende Beleuchtung und eine ruhige Hand sind weitere Voraussetzungen für die Montage des ICs.

Wer es sich absolut nicht zutraut, den IC einzulöten, dem kann auch geholfen werden, da der Bezug einer teilbestückten LP möglich ist.

Anfragen, egal zu welcher Problematik, beantworten wir gerne per Email.

Viel Spaß beim Aufbau und Einsatz der Baugruppe wünscht Ihnen das

Amatronik Entwicklungsteam