

Übertragungsprotokoll des MMM8x8

Das im ELVjournal 6/2013 vorgestellte batteriebetriebene LED-Mini-Matrix-Modul MMM8x8 kann wie auch andere Produkte aus dem ELVjournal ohne großen Aufwand in eigene Anwendungen und Installationen integriert werden. Ermöglicht wird dies durch die offene und simple Kommunikationsschnittstelle zur Ansteuerung des MMM8x8, die im Folgenden vorgestellt wird.

Anschluss des MMM8x8

Über die Buchsenleiste BU5 des MMM8x8 kann eine Verbindung zum PC mittels eines UART-zu-USB-Umsetzers erfolgen. Hier kann ein beliebiger USB-Umsetzer, z. B. der UM2102 von ELV, eingesetzt werden (Bild 1). Angesprochen wird das Gerät über einen virtuellen COM-Port (VCP).

Die notwendigen Verbindungsparameter zur Kommunikation mit dem MMM8x8 über einen virtuellen COM-Port können der Tabelle 1 entnommen werden. Zum Testen der Verbindung kann ein beliebiges Terminalprogramm (z. B. HTerm [1]) zur Kommunikation mit dem virtuellen COM-Port genutzt werden. Je nach eingesetztem Umsetzer werden unterschiedliche Treiber benötigt.

Grundlagen des Kommunikationsprotokolls

Das Protokoll orientiert sich sehr stark an den zuvor vorgestellten Kommunikationsschnittstellen anderer Produkte aus den ELVjournalen, beispielsweise dem 5x5x5-RGB-Cube RGBC555 oder dem Info-Display

ID100. Das MMM8x8 wird über eine Reihe an das Gerät gesendeter Befehle gesteuert, anschließend wird eine Antwort vom Gerät ausgelesen.

Ein Befehl und die Antwort des MMM8x8 erfolgen innerhalb eines festen Kommunikationsrahmens (siehe Tabelle 2). Ein Befehl startet mit dem Steuerzeichen STX (0x02). Es folgen zwei Byte, die die Anzahl der folgenden Bytes in den eigentlichen Nutzdaten (Befehlsbyte und eventuelle Parameter) angeben. Nun folgt ein Byte für den Befehl und bis zu 100 Byte für die je nach Befehl erforderlichen Parameter. Der Befehl wird mit einer Prüfsumme (CRC16) aller vo-

Tabelle 1

Kommunikationsparameter zur Verbindung mit einem virtuellen COM-Port

Baudrate	38.400
Datenbits	8
Parität	none/keine
Stoppbits	1



rigen Bytes (Startzeichen, Nutzdatenlänge, Befehls- und Parameterbytes) abgeschlossen. Der CRC16 wird auf Basis des Polynoms 0x8005 und des Initialwerts 0xFFFF berechnet. Details zur Berechnung eines CRC16 finden sich beispielsweise in Wikipedia [2]. Auch die Antwort auf den gesendeten Befehl wird anhand dieses Kommunikationsrahmens gesendet. Eine Auflistung und Erläuterung aller möglichen Befehle und Antworten des MMM8x8 findet sich in den Tabellen 3 und 4.

Bei der Übertragung gibt es zwei Steuerzeichen, die in den zu übertragenden Daten nicht vorkommen dürfen: das Startzeichen (0x02) und das Escapezeichen (0x10). Wenn eines dieser Steuerzeichen in der Länge, den Nutzdaten oder der Prüfsumme enthalten ist, wird diesem ein Escapezeichen (0x10) vorangestellt und das höchstwertige Bit (0x80) gesetzt. Hieraus ergeben sich folgende Umsetzungen: 0x02 wird zu 0x1082 und 0x10 wird zu 0x1090.

Senden einer eigenen LED-Matrix

Die naheliegendste Ansteuerung des MMM8x8 nach dem ersten Verbindungsaufbau ist die Übertragung einer eigenen konfigurierten LED-Matrix. Ein Testbefehl zur Darstellung des Buchstabens „L“ auf dem MMM8x8 lautet:

```
0x02 0x00 0x09 0x44 0x00 0x7E 0x40 0x40 0x40 0x40 0x40 0x00 0x25 0x3B
```

Das MMM8x8 antwortet mit folgenden Daten:

```
0x02 0x00 0x01 0x44 0x2F 0xBC
```

Die Antwort setzt sich aus einem Byte Nutzdaten zusammen, dem Byte des zuvor gesendeten Befehls – der Befehl wurde erfolgreich ausgeführt (siehe Tabelle 4 für eine komplette Antwortübersicht). Bild 2 zeigt die Nutzung des Beispielbefehls mit dem Terminalprogramm HTerm.

Bei den folgenden Erläuterungen zur LED-Matrix wird das MMM8x8 von vorne betrachtet, d. h. die UART-Schnittstelle befindet sich auf der gegenüberliegenden bzw. der dem Benutzer zugewandten Seite (siehe Bild 3).

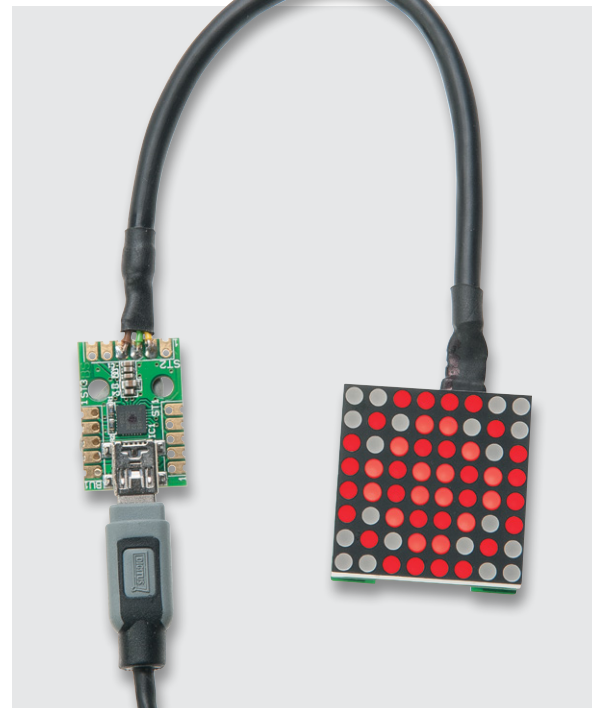


Bild 1: UART-zu-USB-Umsetzer UM2102 mit verbundenem MMM8x8

Die gewünschte LED-Matrix wird als Parameter in Form von acht Byte ausgedrückt. Jede Spalte innerhalb der LED-Matrix des MMM8x8 entspricht hierbei einem zu übertragenden Byte, jede Zeile einem Bit in der dazugehörigen Spalte (siehe Tabelle 5 für ein Beispiel, siehe Bild 4 für eine Visualisierung der Zeilen- und Spaltenzuordnung). Die acht Byte im oberen Befehl zur Definition der LED-Matrix lauten „0x00 0x7E 0x40 0x40 0x40 0x40 0x40 0x00“. Das erste und letzte Byte haben jeweils den Wert 0x00, sprich keine LED der ersten und letzten Spalte soll aktiviert werden. In der zweiten Spalte sollen die LEDs 2 bis 7 aktiviert werden, einzig die LEDs 1 und 8 sollen nicht leuchten. Das zweite Byte 0x7E (binär 0111 1110) drückt diesen gewünschten Zustand aus. Bei den verbleibenden Spalten soll jeweils nur die siebte LED aktiviert werden, um den Buchstaben „L“ auf dem MMM8x8 darzustellen, was mit dem Byte 0x40 (binär 0100 0000) erreicht wird.

Tabelle 2

Kommunikationsrahmen von Befehlen und ihren Antworten

Inhalt	Byteanzahl	Beschreibung
STX (0x02)	1	Startzeichen
Nutzdatenlänge	2	Anzahl der Nutzdaten (Befehl und Parameter)
Befehl	1	Befehl (siehe Tabelle 3)
Parameter	0 bis 100	Parameter des Befehls (siehe Tabelle 3 für Befehle und Tabelle 4 für Antworten)
Prüfsumme	2	CRC16 auf Basis des Polynoms 0x8005 und des Initialwerts 0xFFFF

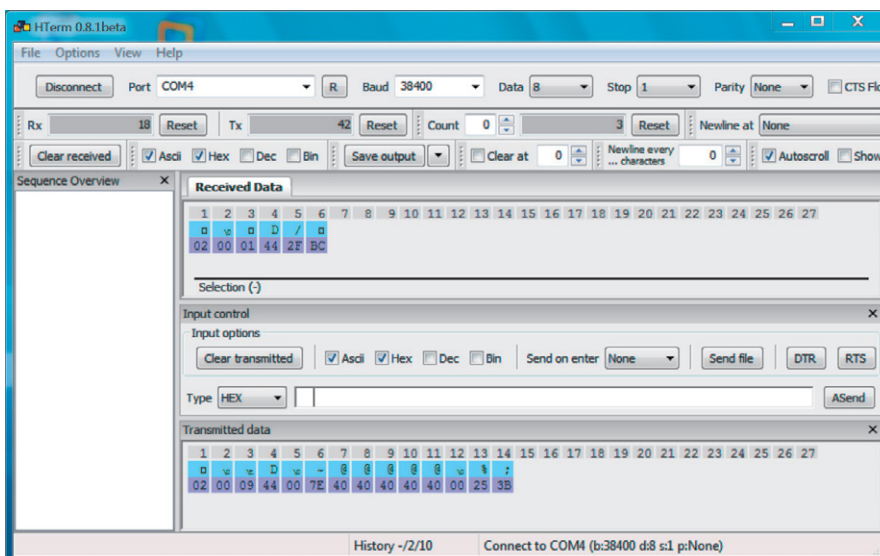


Bild 2: Kommunikation mit dem MMM8x8 mit dem Terminalprogramm HTerm: Senden des Befehls zur Darstellung des Buchstabens „L“ auf dem MMM8x8

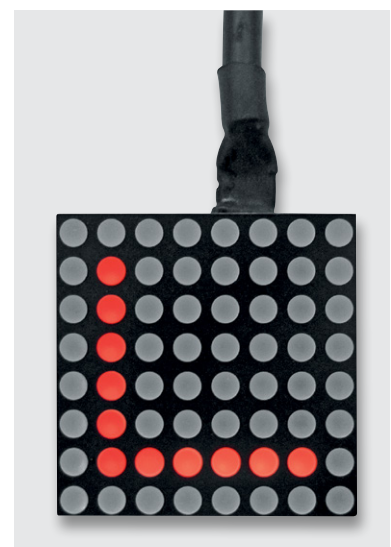


Bild 3: Darstellung des Demobefehls („L“) auf dem MMM8x8

**Befehlsübersicht**

Befehl	Parameter	Parameterlänge	Beschreibung
A (0x41)	-	0	Aktivierung des Normalmodus
B (0x42)	-	0	Aktivierung des Bild-Vorschaumodus
C (0x43)	-	0	Aktivierung des Text-Vorschaumodus
D (0x44)	LED-Matrix als 8 Byte je Spalte (siehe Tabelle 5)	8	Senden eines Vorschaubildes
E (0x45)	ASCII-codierte Zeichenkette	1-100	Senden eines Vorschautes
F (0x46)	ein Byte für die Bestimmung der Text-Scrollgeschwindigkeit, 0: schnell (Minimum), 1-254: jeweilige Abstufung, 255: langsam (Maximum)	1	Setzen der Text-Scrollgeschwindigkeit
G (0x47)	- LED-Matrix als 8 Byte je Spalte (siehe Tabelle 5) - Anzeigedauer im Intervall von 100 ms (z. B. für 3,6 Sekunden 36/0x24)	9	Update des internen Speichers (als Parameter übergebene LED-Matrix wird an Speicheradresse 0 gespeichert)
I (0x49)	gleiche Parameter wie „G“	9	Update des internen Speichers (als Parameter übergebene LED-Matrix wird an die nächste Speicheradresse gespeichert)
J (0x4A)	ASCII-codierte Zeichenkette	1-100	Speicherung einer Zeichenkette
v (0x76)	-	0	Auslesen der Firmwareversion
X (0x58)	-	0	Starten eines Werksresets
! (0x21)	-	0	Starten des Bootloaders

Tabelle 3

Antwortübersicht

Befehl	Parameter	Parameterlänge	Beschreibung
A (0x41)	-	0	Normalmodus wurde gestartet
B (0x42)	-	0	Bild-Vorschaumodus wurde gestartet
C (0x43)	-	0	Text-Vorschaumodus wurde gestartet
D (0x44)	-	0	gesendetes Vorschaubild wurde empfangen
E (0x45)	-	0	gesendeter Vorschautes wurde empfangen
F (0x46)	-	0	gesendete Text-Scrollgeschwindigkeit wurde gesetzt
G (0x47)	-	0	Speicherupdate wurde gestartet, LED-Frame wurde an Adresse 0 gespeichert
I (0x49)	-	0	gesendetes LED-Frame wurde an die nächste Speicheradresse gespeichert
v (0x76)	je 2 Byte pro Versionsstelle	6	Firmware-Version wurde gesendet
X (0x58)	-	0	Werksreset wurde gestartet
! (0x21)	-	0	Bootloader wurde gestartet
NAK (0x15)	-	0	Befehl ist unbekannt oder der Speicher ist voll

Tabelle 4

8 Byte der LED-Matrix

Zeile	Spalte	Byte	Bit
1	1	0	0
2	1	0	1
7	1	0	6
8	1	0	7
1	2	1	0
2	2	1	1
1	7	6	0
2	7	6	1
1	8	7	0
2	8	7	1

Tabelle 5

Betriebsmodi des MMM8x8

Das MMM8x8 besitzt drei Betriebsmodi: Den Normalmodus, den Bild-Vorschaumodus und den Text-Vorschaumodus, die jeweils über die Befehle „A“, „B“ und „C“ gestartet werden können (siehe Tabellen 3 und 4). Der Normalmodus entspricht dem autarken Betrieb ohne Verbindung mit einer ansteuernden Software. Bei der Übertragung einer gewünschten LED-Matrix mittels des Befehls „D“ schaltet das Gerät automatisch in den Bild-Vorschaumodus, ein explizites Umschalten des Modus ist nicht vonnöten.

Speicherung von Bildern für den Normalbetrieb

Damit das MMM8x8 mittels Knopfzelle im Normalmodus genutzt werden kann, müssen die gewünschten Bilder dauerhaft auf dem MMM8x8 gespeichert werden. Für diesen Zweck müssen die Befehle „G“ und „I“ genutzt werden. Bei der Übertragung des ersten Bildes muss ersterer Befehl genutzt werden, für jedes weitere folgende Bild wird der Befehl „I“ benutzt. Die Parameter beider Befehle erwarten dasselbe Muster:



Zuerst folgen acht Byte zur Definition der LED-Matrix (siehe vorige Erläuterung zum Senden einer eigenen LED-Matrix), es folgt ein Byte zur Definition der gewünschten Anzeigedauer des Bildes im Normalmodus. Sie wird in 100 Millisekundenschritten mit diesem Byte ausgedrückt, 3000 Millisekunden entsprechen beispielsweise dem Byte 0x1E (dezimal 30). Die Gesamtzahl an Parametern beim Speichern eines Bildes beträgt somit neun Byte.

Um beispielsweise drei Bilder mit den Buchstaben „E“, „L“ und „V“ mit einem Anzeigeintervall von 1,5 Sekunden (15 * 100 Millisekunden) zu speichern, sind die folgenden drei Befehle vonnöten:

```
- 0x02 0x00 0x0A 0x47 0x00 0x7F 0x49 0x49 0x49 0x49 0x00 0x0F 0x32 0x93
```

```
- 0x02 0x00 0x0A 0x49 0x00 0x7F 0x40 0x40 0x40 0x40 0x00 0x0F 0x74 0xD1
```

```
- 0x02 0x00 0x0A 0x49 0x00 0x0F 0x30 0x40 0x40 0x30 0x0F 0x00 0x0F 0xC7 0x3B
```

Die Bildfolge wird im Normalmodus auf dem MMM8x8 dargestellt, dieser muss nach der Übertragung der Befehle explizit mit dem Befehl „A“ (siehe zuvor) eingestellt werden oder kann durch Trennung und Wiederherstellung der Spannungsversorgung gestartet werden.

Darstellung und Speicherung von Zeichenketten

Ein Lauftext kann über den Befehl „E“ auf dem MMM8x8 angezeigt werden, als Parameter können bis zu 100 ASCII-codierte Zeichen genutzt werden. Auch hier wird bei Nutzung des Befehls automatisch der Text-Vorschaumodus aktiviert. Ein Beispielbefehl, um die Zeichenfolge „MMM8x8“ auf dem Gerät als Lauftext anzuzeigen, lautet:

```
- 0x02 0x00 0x07 0x45 0x4D 0x4D 0x4D 0x38 0x78 0x38 0x4C 0x72
```

Um eine Zeichenkette auf dem Gerät fest abzuspeichern, wird der Befehl „J“ genutzt. Dabei wird der zuvor abgespeicherte Text im Speicher des MMM8x8 überschrieben. Um die Zeichenfolge „MMM8x8“ im internen Speicher abzulegen, muss der folgende Befehl genutzt werden:

```
- 0x02 0x00 0x07 0x4A 0x4D 0x4D 0x4D 0x38 0x78 0x38 0x43 0x82
```

Falls nur Bilder angezeigt werden sollen, lässt sich der Lauftext auch löschen. Dies geschieht mit dem folgenden Befehl, indem ein leerer Text in dem Gerät gespeichert wird:

```
- 0x02 0x00 0x01 0x4A 0xAF 0x9B
```

Bei einem leeren Text springt das Gerät im Normalbetrieb direkt wieder in den Bildmodus und beginnt die Bildfolge von Neuem.

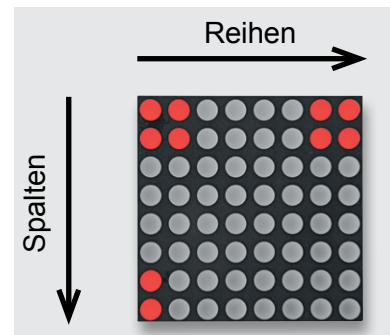
Die Geschwindigkeit des Lauftextes kann über den Befehl „F“ variiert werden, der Parameter drückt das gewünschte Tempo aus. Hierbei ist zu beachten: je höher der eingestellte Wert ist, desto langsamer ist die Geschwindigkeit des Lauftextes. Daraus folgt, dass der Wert 0x00 der schnellsten und 0xFF der langsamsten Text-Scrollgeschwindigkeit entspricht, die dazwischen liegenden Werte entsprechen einer Abstufung der Geschwindigkeit.

Erstellte Demos

Zum besseren Verständnis der Schnittstelle wurden zwei Beispielansteuerungen in der Programmiersprache C# erstellt: Ein digitaler Würfel (Bild 5) und das Spiel „Snake“ (Bild 6). Beide Anwendungen mit den dazugehörigen Quelltexten werden auf der Produktseite zum MMM8x8 im ELV-Web-Shop zum kostenfreien Herunterladen angeboten. Die Projektmappe der Demoanwendung besteht aus drei einzelnen C#-Projekten, den beiden Spielen („MMM8x8DigitalCube“ und „MMM8x8Snake“) und einer Bibliothek zur Kommunikation und Nutzung des MMM8x8 („MMM8x8Utils“). Im Utility-Projekt finden sich vier Klassen:

- ComPortDevice: Verwaltet die Verbindung zum MMM8x8 über einen virtuellen COM-Port und stellt Methoden zur Kommunikation zur Verfügung. Die Befehle werden auf Basis eines übergebenen Befehls und der dazugehörigen Parameter anhand des Kommunikationsrahmens (siehe Tabelle 2) generiert und zum Gerät gesendet. Die Antworten werden entsprechend umgekehrt decodiert.

Bild 4:
MMM8x8 mit
aktivierten LEDs
aus Tabelle 5



- CRC16: Die Klasse wird zur Berechnung der Prüfsumme (CRC16) genutzt (siehe Abschnitt zu den Grundlagen des Kommunikationsprotokolls).
- MMM8x8: Stellt Konstanten für die Nutzung des MMM8x8 bereit, zum Beispiel Kommunikationsparameter zur Ansteuerung des virtuellen COM-Ports und eine Sammlung von möglichen Befehlen.
- MMM8x8LedMatrix: Die LED-Matrix des MMM8x8 wird mit der Klasse repräsentiert. Sie bietet die vereinfachte Generierung der acht Byte der LED-Matrix (siehe Tabelle 5) an, indem gewünschte LEDs anhand von Reihen- und Spaltenangaben ein- bzw. ausgeschaltet werden können.
- Die beiden Demoanwendungen nutzen die zuvor erläuterten Klassen. Der digitale Würfel (MMM8x8 DigitalCube) generiert 30 Zufallszahlen zwischen 1 und 6 und stellt diese nacheinander mittels des Bild-Vorschaumodus auf dem Gerät dar. Auch das Spiel Snake („MMM8x8Snake“) nutzt den Bild-Vorschaumodus zur Darstellung auf dem MMM8x8. **ELV**

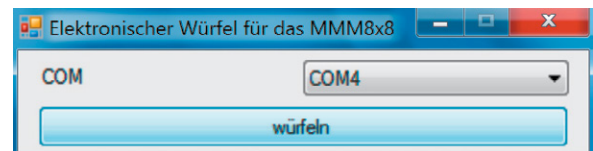


Bild 5: Screenshot vom elektronischen Würfel

Bild 6:
Screenshot
von Snake



Weitere Infos:

[1] www.der-hammer.info/terminal

[2] http://de.wikipedia.org/wiki/Zyklische_Redundanzprüfung

Download der Demoanwendungen:
www.elv.de: Webcode #1302