

Sehr geehrte Leserin, sehr geehrter Leser,

sofern Sie die Artikel in der Modellbahnschule Nr. 23 und 24 gelesen haben, sind Ihnen die Grundlagen bekannt, die folgenden Absätze können Sie in diesem Fall im Wesentlichen überspringen, wirklich interessant dürfte für Sie die unten dargestellte Dokumentation sein. Diese Datei stellt einen Entwurf dar und muss noch in einigen Punkten ergänzt und korrigiert werden. Die Programme werden wir schnellstmöglich zum Download bereitstellen.

- Inhalt: I. Allgemeine Beschreibung: Das Modell
Ausleuchtung der Kanäle
Elektronik
Geräusch
- II. Dokumentation: Aufgabe
Umsetzung
Quelltexte
- III. Stückliste für eine Platine
- IV. Erläuterungen zum Schaltplan
- V. Installationsanleitung für die Programme

I. Allgemeine Beschreibung

Das Modell

Der Hochofen – Bausatz von Trix/Walters ist schon vor einigen Jahren erschienen. Für unsere Anlage haben wir die Kanäle mittels LEDs ausgeleuchtet. Zunächst konnte allerdings die gesamte Beleuchtung nur ein- bzw. ausgeschaltet werden, die Illusion des beim Abstich herablaufenden Roheisens war nicht möglich. Genau das wollten wir natürlich unseren Besuchern bieten. Das Roheisen sollte fließen und am Ende langsam ausglühen. Dazu war es zum ersten notwendig jede LED nacheinander aufleuchten zu lassen und zum zweiten diese LEDs dann langsam herunterzudimmen.

Die Neuauflage des Hochofenmodells wurde von Walters um einige Plattformen, Treppen und Geländer ergänzt. Die zusätzlichen Teile sind auch einzeln erhältlich, so dass ältere Modelle aufgerüstet werden können. Wer die Abbildungen genauer betrachtet, kann erkennen, dass wir bislang erst einen Teil nachgerüstet haben.

Das Modell ist für Transportzwecke vollständig von der Grundplatte abnehmbar. Die Umriss wurden mittels 2x2 mm Leisten passgenau abgebildet, diese dann mit Spachtelmasse bzw. Sand eingefasst. Die grosse Rohrverbindung zwischen Hochofen und Staubsilo mußte natürlich ebenfalls unterbrochen werden. Hierzu haben wir ein Stück Polystyrolstange auf der Drehbank eingepasst, so dass eine stabile Verbindung besteht.

Ausleuchtung der Kanäle

Als Leuchtmittel haben wir LEDs verwendet, da sie gegenüber Mikrobirnen keine nennenswerte Wärme produzieren. Die Kanäle wurden zunächst mit Alufolie als Lichtreflektor ausgelegt. Für eine gleichmäßige und wirkungsvolle Beleuchtung wurden die Leuchtdioden so eng wie möglich gesetzt. Verwendet haben wir rechteckige, gelbe LEDs. An diesen wurde ein Stufenversatz herausgeschnitten, so dass man sie überlappend montieren kann. Zuvor wurden die LEDs mit orangem Lampenlack bemalt. Über die Anschlußbeinchen haben wir Schrumpfschlauch gezogen. Den gesamte Kanal haben wir mit Folie abgedeckt, so dass eine einheitliche Oberfläche entstand. Die Folie wurde ebenfalls mit dem orangefarbenem Lampenlack bemalt. An die Anschlussbeinchen der LEDs mussten natürlich Drähte angelötet werden, die unter der Platine zwischen den Kanälen in Richtung des Hochofen verlaufen, um dort unter die Anlage geleitet zu werden.

Elektronik

Wie oben erwähnt genügte uns eine einfache Beleuchtung der Kanäle nicht. Wir wollten erreichen, dass möglichst vorbildgerecht ein Abstich dargestellt wird. Die Realisierung war bei den uns zur Verfügung stehenden Möglichkeiten am besten mittels Microcontroller zu erreichen. Steuert man die Angelegenheit mittels PC, ist man immer darauf angewiesen, ein Gerät zur Hand zu haben, die Platinen können dagegen unter die Anlage geschraubt werden.

Das Modell verfügt über zwei Kanäle, die beim Vorbild unabhängig voneinander benutzt werden können. Einzelheiten zum Aufbau und Funktion einer Abstichrinne sind in Modellbahnschule 24 dargestellt. Für eine vorbildgerechte Handhabung hätte man die Ausleuchtung beider Kanäle getrennt steuern müssen. Hier haben wir stark vereinfacht: Das „Roheisen“ läuft auf unserer Anlage simultan durch beide Kanäle. Sofern man dies bei einem eventuellen Nachbau nicht wünscht, kann man sich auf die Ausleuchtung eines Kanals beschränken, oder man muss für jeden Kanal eine eigene Steuerung vorsehen. Jeweils eine LED im vorderen Kanal ist bei unserem Modell mit einer LED im Kanal in der Gießhalle gleichgeschaltet, an den Verzweigungen in den Kanälen kommt eine weitere LED hinzu, so dass teilweise drei LEDs parallel mit einem Ausgangskanal unserer Schaltung gesteuert werden. In diesem Fall fließt nur 1/3 des Stroms durch jede LED; daher haben wir den Strom durch Parallelschaltung von Widerständen (im Huckepackverfahren) verdreifacht, so dass jede LED wieder den ursprünglichen Strom erhält.

Unsere Schaltung ist modular aufgebaut. Ein Modul ist für 16 LEDs ausgelegt. Es besteht die Möglichkeit, mehrere dieser Module parallel zu betreiben, wir haben uns für drei Module (Platinen) entschieden. Dadurch können maximal 48 LEDs gesteuert werden. Die Schaltung kann auch für andere Zwecke eingesetzt werden, jede Anwendung bei der mehrere Lichtquellen nacheinander angesteuert werden sollen ist denkbar. Die Module können mit einem PC kommunizieren und somit eingestellt werden. Außerdem ist der Start des Abstichs mittels Taster zu steuern.

Wer sich schon einmal mit LEDs beschäftigt hat, weiß, dass eine Dimmung regelmäßig nicht funktioniert, die Dinger leuchten entweder oder sie sind aus, die Regulierung der anliegenden Spannung zeigt nur geringe Wirkung. Um LEDs schwächer leuchten zu lassen, müssen sie mit einer Spannung angesteuert werden, die ähnlich wie die Motorsteuerung bei digital gesteuerten Lokomotiven ständig an und aus geht. Hierfür ist unsere Schaltung ausgelegt, die Phase, in der die LEDs heller werden, ist relativ kurz, dann folgt die zweite Phase, in der die LEDs voll beleuchtet sind, das Abfallen der Helligkeit ist verhältnismäßig lang eingestellt, um das Nachglühen im Kanal zu simulieren. Insgesamt dauert die Angelegenheit etwas über eine Minute, aber die einzelnen Zeitparameter können über die Programmierung problemlos verändert werden.

Wir haben bewußt darauf verzichtet, die Platinen zu ätzen. Der Aufbau sollte möglichst einfach zu realisieren sein. Die Verbindung der einzelnen Elemente erfolgte mittels Drähten, die Verbindungen zwischen den einzelnen Lötunkten wurde mit reichlich Lötzinn hergestellt. Wir sind uns völlig darüber im Klaren, dass eine professionelle Schaltung anders aussieht, aber wie sagt der Amerikaner? It works!

Verwendet wurden Experimentier Platinen, bei denen eine Bohrung einem Lötunkt auf der Rückseite entspricht. Verbindungen zwischen den Lötunkten bestehen nicht. Allerdings sind bei verschiedenen Ausführungen am Rand Verbindungen vorhanden, man sollte sich die einzelne Platine daher genau ansehen.

Im Zentrum der Schaltung steht natürlich jeweils ein Microcontroller, der die 16 LEDs einzeln ansteuert. Auf jeder Platine haben wir ein Trimpoti vorgesehen, um die Gesamthelligkeit etwas nachregeln zu können. Die Wirkung geht allerdings gegen Null, so dass man gut und gerne darauf verzichten kann. Zur Stromversorgung wird ein einfaches Netzteil von Voltcraft verwendet, das auf 9V Gleichstrom eingestellt werden kann. Die Verbindung der Platinen untereinander wird jeweils mittels dreier Steckverbindungen realisiert. Diese sind aus Stabilitätsgründen sechspolig ausgeführt, wobei allerdings maximal drei Pole verwendet werden. Der gesamte Aufbau wurde getrennt von der Anlage auf einem Brettchen aufgeschraubt, um nicht bei jedem Test das ganze Arrangement bewegen zu müssen.

Auf den Platinen wurden jeweils zwei Block- LEDs aufgelötet um die Schaltung kontrollieren zu können, mittels eines Jumpers können sie aktiviert bzw. deaktiviert werden. Ein gleichzeitiger Betrieb parallel zur Ausleuchtung der Kanäle ist nicht vorgesehen, der Strom ist zu gering.

Die Kabel- Verbindungen zwischen Modell und Platine wurden farblich geordnet zu je 8 Stück zusammengefaßt. Die Steckverbindungen auf den Platinen mußten recht klein ausfallen, da für robustere Teile einfach kein Platz vorhanden war. Wir haben die Lötunkte an den Steckern daher mit Schrumpfschlauch überzogen und das Ganze mit reichlich Heißkleber vergossen.

Geräusch

Zusätzlich haben wir den Abstich noch mit einem Sound – Director von Uhlenbrock gekoppelt, um auch eine entsprechende Geräuschkulisse zu schaffen. Da beide Anwendungen über verschiedene Steuerspannungen verfügen, mußte ein Relais zwischen geschaltet werden, um beide mittels eines einzigen Knopfdrucks zu starten.

Der Sound stammt aus einem auf DVD erhältlichen Dokumentarfilm. Er wurde mittels eines Transit USB der Firma M-Audio (Bezug: Musikhaus Thomann, Nürnberg) auf die Festplatte gebracht. Als Aufnahmeprogramm

diente Audacity, dieses kann kostenlos aus dem Internet heruntergeladen werden. Damit war es auch möglich den Sound auf die gewünschte Länge zu schneiden und eine Ein- und Ausblendung vorzunehmen. Anschließend wurde die Datei in das benötigte MP3 Format konvertiert.

Ein eventueller Nachbau sollte von niemand in Erwägung gezogen werden, der noch niemals einen Lötkolben in der Hand hatte. Für ein Gelingen können wir keinerlei Garantie übernehmen. Sämtliche Gewährleistungsansprüche werden ausgeschlossen. Für konstruktive Kritik und Verbesserungsvorschläge sind wir dankbar.

Nachstehend ist die Dokumentation wiedergegeben.

Mannheim, im April 2011

Modulbaufreunde Ladenburg

II. Dokumentation

Aufgabe

Beim Abstich eines Hochofens fließt der glühende Stahl durch Rinnen zu Auffangbehältern hin, wobei es auch zu Verzweigungen kommen kann, z.B. durch das Abscheiden der Schlacke vom Roheisen. Es sollte das Fließen des glühenden Stahls mit einer Reihe von LEDs nachgebildet werden:

- mit Beginn des Abstichs fließt das Roheisen durch die Rinne bis es den Auffangbehälter erreicht – nacheinander soll eine LED nach der anderen hell aufleuchten. Falls es eine Verzweigung gibt, müssen ab dem Punkt der Verzweigung zwei LEDs gleichzeitig aufleuchten.
- solange der Hochofen sich entleert fließt glühendes Eisen durch die Rinne, alle LEDs leuchten hell
- Wenn der Hochofen leer ist fließt kein glühendes Eisen mehr nach und der in der Rinne anhaftende Rest kühlt langsam ab und glüht zunehmend dunkler. Die LEDs müssen beginnend vom Hochofen her zunehmend gedimmt werden, bis sie schließlich ganz aus sind. Die Farbänderung von weißglühendem Eisen über gelb, orange hin zu rot lässt sich mit LEDs nur einer Farbe leider nicht erzielen.

Mit welcher Geschwindigkeit das Eisen fließt – die LEDs nacheinander eingeschaltet werden – muss im nachhinein subjektiv eingeschätzt werden. Daher muss diese Geschwindigkeit flexibel einstellbar sein. Es muss auch sichergestellt werden, dass die LEDs auch tatsächlich einen Fließvorgang simulieren, das heißt die Reihenfolge der LEDs muss strikt eingehalten werden. Wenn allerdings jede LED völlig unabhängig angesteuert werden kann, und die Art und Weise wie und wann die LED leuchtet flexibel einstellbar ist, dann ist ein versehentlicher Verdrahtungsfehler leicht zu umgehen.

Daraus leitet sich jetzt die Aufgabenstellung ab:

Anforderungen

- es soll eine große Anzahl von LEDs ansteuerbar sein
- die Art und Weise wie eine einzelne LED leuchtet soll flexibel einstellbar sein, das heißt
- wann ist eine LED an, wann aus
- wie hell leuchtet eine LED zu einem bestimmten Zeitpunkt
- Der Ablauf sollte über ein äußeres Signal steuerbar sein (Anforderung durch 'Kunden')
- Der Ablauf sollte, einmal gestartet ablaufen bis zum Ende

Und wenn diese Aufgabe wie oben beschrieben gelöst ist, dann ist mit dem System noch vieles mehr darstellbar als nur fließendes Roheisen:

- ein Feuerwerk
- die brennende Flamme eines Feuerspuckers
- eine Diskobeleuchtung

Konkretisierung

- die LEDs werden über ein Puls-Pause Verhältnis gedimmt; dabei muss die Frequenz so hoch sein, dass diese LED für das Auge nicht flimmert (> 100 Hz)
- Die Ansteuerung erfolgt mittels eines Microcontrollers. Die Firmware des Microcontrollers bietet die notwendige und gewünschte Flexibilität
- Der Microcontroller kommuniziert über eine Standard-RS232 Schnittstelle mit einem PC; diese

- sind bei vielen PCs vorhanden, es gibt jedoch auch günstige USB-RS232-Konverter
- Der gewählte Microcontroller ist ein ATmega8L von Atmel
 - Hat Programmspeicher (Flash) auf dem Chip
 - Hat Arbeitsspeicher (SRAM) auf dem Chip
 - Hat EEPROM-Speicher auf dem Chip (Konfiguration bleibt erhalten)
 - kann mit einem internen Oszillator arbeiten (8 MHz)
 - Kann (über einen entsprechenden Programmieradapter) im System programmiert werden
 - eine Programmierumgebung ist kostenlos verfügbar (WinAVR)
 - ein Simulationsprogramm/Debugger ist kostenlos verfügbar (AVRStudio von Atmel)
 - Das Programm wird auf die Ansteuerung von 16 LEDs hin ausgelegt
 - Das System wird so ausgelegt, dass mehrere gleichartige Systeme synchron zusammenarbeiten können; die Anzahl der LEDs kann auf diese Weise (in gewissem Rahmen) beliebig erhöht werden
 - Die Ansteuerung eines Rot/Grün leuchtenden Drucktasters wird vorgesehen

Umsetzung

Schaltungstechnik

Steuerung der LEDs

Jede LED wird von einem durch eine Stromspiegel eingestellten Strom durchflossen. Über ein Trimpoti kann dieser Strom für alle LEDs eingestellt werden. Ein vom Microcontroller geschalteter Transistor kann diesen Strom schalten, so daß die LED über das Puls-Pauseverhältnis effektiv gedimmt werden kann.

Ablaufsteuerung

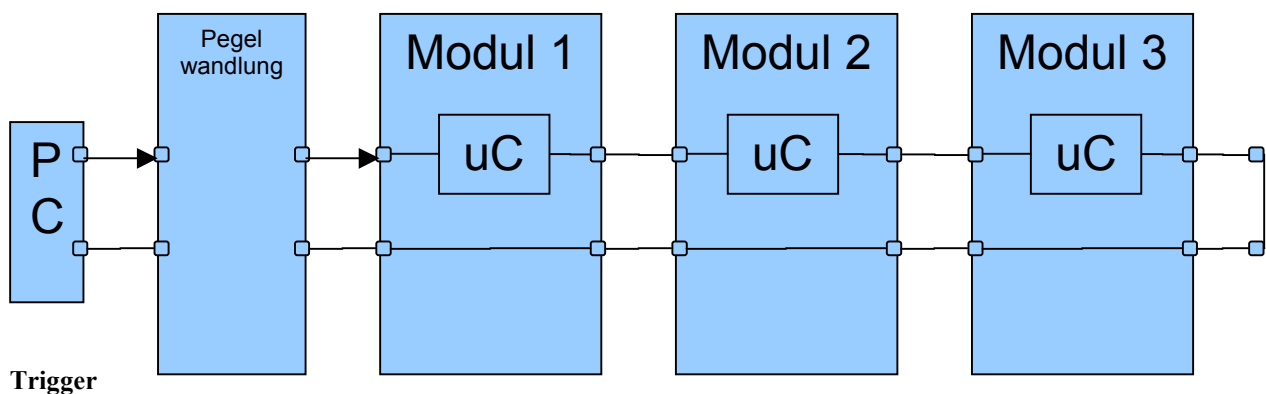
Das Programm soll nicht automatisch ablaufen, sondern erst auf Anforderung des Gastes über einen Drucktaster gestartet werden. Wenn es dann jedoch läuft soll es nicht abgebrochen werden können. Beides soll angezeigt werden: wenn das Programm startbereit ist soll eine grüne LED am Drucktaster leuchten, wenn das Programm läuft und somit nicht neu gestartet werden kann soll eine rote LED dies anzeigen.

Wenn der Drucktaster gedrückt wird, so wird ein Pin des Microcontrollers auf Masse gezogen.

Kommunikation

Eine typischer Pegelwandlerbaustein konvertiert die Spannungspegel der RS232-Schnittstelle von bis zu $\pm 15V$ in die für den Microcontroller zulässigen Pegel 0 und 5V und umgekehrt. Dies muss nur einmal realisiert werden, da zwischen den einzelnen Modulen die Kommunikation mit den Pegeln 0 und 5V erfolgt.

Alle Module werden in einer Ring-Topologie verschaltet: die Ausgangsleitung des PC geht (nach dem Pegelwandler) auf den Eingang des ersten Moduls. Dessen Ausgang geht unmittelbar auf den Eingang des nächsten Moduls und so weiter. Erst am Ausgang des letzten vorhandenen Moduls muss dieser mit der Eingangsleitung des PC (wiederum über den Pegelwandler) verbunden werden.



Der Ablauf soll vom Betrachter auf Wunsch ausgelöst werden, die Möglichkeit diesen auszulösen soll durch eine grüne LED dargestellt werden. Falls der Ablauf gerade nicht neu ausgelöst werden kann – weil er gerade abläuft – soll ein rote LED dies anzeigen. Der Anschluß eines Drucktasters mit einer integrierten rot und grünen LED ist vorgesehen. Dieser Drucktaster muss nur an einem Modul angeschlossen werden, alle weiteren Module werden automatisch synchronisiert.

Synchronisation

Da mehrere Systeme zusammen einen gemeinsamen Ablauf darstellen sollen muss der zeitliche Ablauf synchronisiert werden. Dies geschieht über eine gemeinsame Synchronisationsleitung: sobald bei einem System der Trigger ausgelöst wird, wird eine Signal ausgegeben, das alle Module zum gleichzeitigen Start des Ablaufs triggert.

Programmierung

Der Microcontroller hat insgesamt folgende Einzelaufgaben:

- Initialisierung
einmalig nach Programmstart
- Ansteuerung der LEDs
Die Kernaufgabe des Programms, Berücksichtigung von Verzögerungen und Helligkeit sowie Helligkeitsänderungen über der Zeit
- Kommunikation
Senden und Empfangen von Zeichen über die Schnittstelle
Verarbeitung der Kommandos
- Ablaufsteuerung
zusätzlicher Programmservice um weitere Schaltungskomponenten durch Programmcode zu ersetzen

Initialisierung

Der uC stellt die Kommunikationsbereitschaft her, indem er die Kommunikationsparameter auf die Einstellungen 9600 Baud, 8 Datenbits, keine Parität, 1 Stopbit setzt (9600,8,N,1). Außerdem wird der Empfangsinterrupt freigegeben.

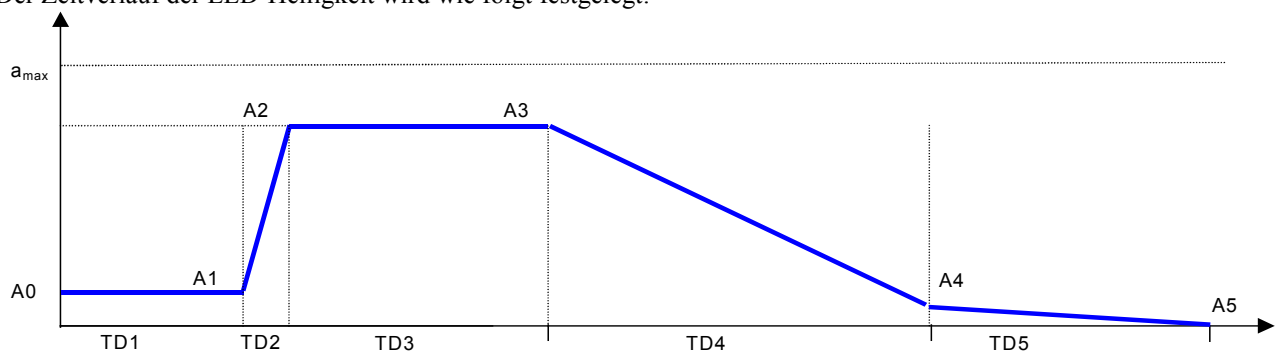
Es werden die aktuelle Einstellungen aus dem EEprom geladen und für die korrekte Ansteuerung der LEDs vorbereitet.

Die Pins werden entsprechend ihrer Funktion konfiguriert:
zur Ansteuerung der LEDs als Ausgänge
zur Erkennung von Eingaben als Eingänge mit Pullup

Zeitverlauf der LED-Helligkeit

Die Helligkeit der LEDs soll sich mit der Zeit ändern. Für die Hochofenanwendung muss eine LED erst nach einer bestimmten Verzögerungszeit aufleuchten, dann für eine Weile konstant hell leuchten um dann über einen längeren Zeitraum nach und nach dunkler zu werden. Allerdings ist dieser typische Zeitverlauf nicht zwingend vorgegeben,

Der Zeitverlauf der LED-Helligkeit wird wie folgt festgelegt:



- Zu Beginn hat eine LED einer Helligkeit von A_0 %
- Während einer Verzögerungszeit T_1 ändert sich die Amplitude zu A_1 %
- Während einer nächsten Verzögerungszeit T_2 ändert sich die Amplitude zu A_2 %
- es sind insgesamt 14 solcher T_x/A_x Paare pro LED möglich

Ansteuerung der LEDs

Die Helligkeit einer LED wird gesteuert über ein Puls-Pausverhältnis, welches von 0% (LED ist ganz aus) bis 100% (LED leuchtet kontinuierlich) betragen kann.

Jede aktive LED wird eingeschaltet, dann zählt der Microcontroller einen Schleifenzähler von 0 bis 99 und für jede LED wird festgestellt, ob der entsprechende Prozentsatz noch über dem aktuellen Zählerstand liegt. Sobald das nicht mehr der Fall ist wird die entsprechende LED ausgeschaltet. Wenn der Zählerstand 100 erreicht ist wird für jede LED bestimmt, ob sich ihre Helligkeit ändern soll und ggf. der Prozentsatz angepasst. Dann beginnt eine weitere Schleife.

Kommunikation

Stichworte:

- *Master Slave*
- *ASCII-Protokoll*
- *Ring-Topologie*
- *Adressierung*
- *Kommandokennzeichnung*

Die Kommunikation erfolgt über ein Master-Slave-Kommunikationsprotokoll. Der Master steht hier für den PC, an dem das Modul angeschlossen ist, das Modul selbst ist demnach der Slave, und das heißt der PC sendet eine Anfrage an das Modul, welche von diesem beantwortet wird. Das Modul sendet auf keinen Fall von sich aus.

Außerdem wird ein ASCII-Protokoll verwendet: alle verwendeten Zeichen können über eine Tastatur eingegeben werden, ein Kommando wird mit einem Carriage Return <CR> abgeschlossen (Enter-, Eingabetaste). Das Format dieses Protokolls wird weiter unten detailliert beschrieben.

Alle Module werden in einer Ring-Topologie verschaltet (siehe oben), und daher müssen die Kommandos adressierbar sein, um gezielt eines der Module im Ring anzusprechen; des Weiteren muss ein Kommando als Anfrage des PC oder als Antwort eines Moduls gekennzeichnet werden.

Um die verschiedenen Module nicht einzeln manuell mit je einer eigenen Adresse versehen zu müssen wird eine automatische Adressvergabe vorgesehen.

Die UART unterbricht die laufende Programmausführung durch einen Interrupt; daher ist die Kommunikation von der restlichen Programmfunktion unabhängig, es kann jedoch während der Kommunikation zu unregelmäßigem Leuchten kommen – insbesondere wenn die Parameter geändert werden. Wenn Parameter geändert werden, dann werden sie erst mit dem nächsten Programmablauf aktiv.

Protokoll

Kodierung

Jedes Telegramm wird als Anfrage oder als Antwort gekennzeichnet. Anfragen werden verarbeitet während Antworten unverändert an das nächste Modul weitergesendet werden.

- , > ' Kennzeichnung einer Anfrage
- , < ' Kennzeichnung einer Antwort

Wenn eine Anfrage von Modul 1 beantwortet wird, so kennzeichnet es die Antwort und alle folgenden Module reichen die Antwort nur weiter, bis letztendlich zum PC. Es kommt dabei zu leichten Verzögerungen pro Modul, allerdings nur um wenige Millisekunden.

Adressierung

Jedes Modul besitzt eine Adresse, welche zu Beginn auf Null gesetzt wird. Sobald ein Modul mit Adresse Null ein Telegramm mit einer gültigen Adresse empfängt, übernimmt es diese Adresse für sich und antwortet auf alle Telegramme für diese Adresse.

Die Adresse Null ist keine gültige Adresse, sondern dient zum Broadcast.

Telegramme für alle anderen Adressen werden unverändert weitergesendet, so dass sich das nächste Device in gleicher Weise verhalten kann.

Adressen können alle alphanumerischen Zeichen sein :

,1'...'9' ,a'...'z' ,A'...'Z'

Colon

Zur gefälligen Formatierung kann unmittelbar nach der Adresse ein Doppelpunkt gesendet werden. Dieser Doppelpunkt ist optional.

Terminierung

Bis zum Empfang des Terminierungszeichens <CR> werden alle weiteren Zeichen in einem Puffer zwischengespeichert, maximal 20 Zeichen. Mit Empfang des Terminierungszeichens wird der empfangene Code interpretiert und beantwortet.

Befehlsatz

Es stehen Kommandos zu Identifikation des Moduls als solches als auch Kommandos zur Konfiguration der Leuchtfunktion zur Verfügung. Die Kommandos zu Identifikation sind nur zur Abfrage möglich.

Bei allen Kommandos zur Konfiguration ist stets ein Setzen und auch ein Lesen möglich.

Identifizierung

Folgende Befehle zur Identifizierung stehen zur Verfügung:

| Befehl | Antwort | Kommentar |
|--------|-----------------------------|----------------------------|
| >1:ID | <1:AVR-LED flashing control | Gerätebezeichnung |
| >1:IDV | <1:1.0.P1 | Version: Major.Minor.Build |

Konfiguration

Es kann der Verlauf der LED-Helligkeit konfiguriert werden.

Um die Komplexität und Länge der Kommandos zu begrenzen werden recht einfache Kommandos definiert. Um die komplexe Funktion der Module zu konfigurieren ist daher eine große Zahl von Kommandos in einer spezifischen Reihenfolge nötig...die Komplexität tritt somit auf der PC-Seite auf.

Ein zusätzliches Testkommando ist dazu vorgesehen, die Anordnung und Reihenfolge der LEDs nach der Verdrahtung zu prüfen.

| Befehl | Antwort | Kommentar |
|---------------------------|-----------------|--|
| >1:LED=0 | <1:+ | Selektion der zu konfigurierenden LED |
| >1:LED=15 | <1:+ | |
| >1:LED | <1:15 | Abfrage der zu konfigurierenden LED |
| >1:MODE | <1:ON <1:OFF | Die aktive LED wird komplett ein- oder ausgeschaltet |
| >1:MODE=ON >1:MODE=OFF | <1:+ | Abfrage der zu konfigurierenden LED |
| >1:AMP0 >1:AMP1 | <1:0 <1:0 | Abfragen der Amplitude AMPx (der selektierten LED) |

| | | |
|---|----------------------|---|
| >1:AMP12 | <1:100 | |
| >1:AMP0=10 >1:AMP1=100 >1:AMP12=0 | <1:+ <1:+ <1:+ | setzen der Amplitude AMPx (der selektierten LED) alle Werte von 0 bis 100 sind gültig |
| >1:T1 ... >1:T12 | <1:10 | Abfragen der Verzögerungszeit TDx (der selektierten LED) 1 → 0.1 Sekunden 10 → 1 Sekunde |
| >1:T1=100 ... >1:T12=1 | <1:+ | setzen der Verzögerungszeit TDx (der selektierten LED) 1 → 0.1 Sekunden alle Werte von 1 bis 255 sind gültig |

LED selektieren

Es wird die LED ausgewählt, auf die sich alle folgenden Kommandos MODE, T* und AMP* beziehen. Zulässig sind die Werte 0 bis 15.

MODE

Eine LED wird grundsätzlich aus- oder eingeschaltet. Eine eingeschaltete LED folgt dem eingestellten Helligkeitsverlauf. Eine ausgeschaltete LED ist schlicht aus.

AMP0**AMP1...AMP12**

Die Helligkeit der gewählten LED wird eingestellt von 0% bis 100%. AMP0 repräsentiert die Helligkeit bei Programmstart, AMP1 die Helligkeit zum Zeitpunkt T1

T1...T12

Die Verzögerungszeit bis der nächste Amplitudenwert erreicht wird. Wenn eine LED ihren Helligkeitswert in 500ms von 20% auf 100% verändern soll, so leuchtet sie die nächsten 100ms mit 20%, dann für 100ms mit $20\%+16\% = 36\%$, dann für 100ms mit 52%, 68%, 84%. Nach 500ms leuchtet die LED mit 100% Helligkeit.

FRQ Frequenzteiler

| Befehl | Antwort | Kommentar |
|--------------|----------|-------------------------|
| >1:FRQ<cr> | <1:1<cr> | Frequenzteiler auslesen |
| >1:FRQ=2<cr> | <1:+<cr> | Frequenzteiler setzen |

Die zeitliche Ablaufsteuerung erfolgt durch einen Timer, der voreingestellt alle 100ms einen Interrupt auslöst. Der voreingestellte Wert hierfür ist 2. Es können Werte 0 bis 3 eingestellt werden. Bei 0 wird alle 3.125 ms ein Interrupt ausgelöst, bei 1 wird alle 25 ms ein Interrupt ausgelöst; der zeitliche Ablauf ist dementsprechend um das 32fach bzw. vierfache beschleunigt. Bei 3 wird alle 400ms Interrupt ausgelöst; der zeitliche Ablauf ist dementsprechend um das vierfache verlangsamt.

Test

Der Test dient dazu einzelne LEDs einfach ein und ausschalten zu können

Der Test-Befehl wird als Broadcast verwendet.

Vorraussetzung für die korrekte Funktion des TST Kommandos ist eine zuvor erfolgte Adressierung auf die Adressen ,0', '1', ...!

| Befehl | Antwort | Kommentar |
|--------|------------------------|---|
| >0:TST | Entfällt bei Broadcast | |
| + | + | Jedes Zeichen wird weitergeleitet |
| - | - | Jedes Zeichen wird weitergeleitet |
| <cr> | <cr> | Jedes Zeichen wird weitergeleitet Ende des Testmodus |

Wenn alle Sensoren in den Testmodus gesetzt werden, werden folgende Aktionen durchgeführt (bei jedem Gerät):

Ein interner LED-Zähler wird auf 0 gesetzt.

Alle LEDs werden gelöscht

Bei Empfang eines ‚+‘ Zeichens wird der interne LED-Zähler aller angeschlossener Einheiten erhöht, bei Empfang eines ‚-‘ Zeichens verringert – innerhalb der Grenzen 0...MAX.
Die Einheit mit Adresse ‚0‘ schaltet bei LED-Zählerstand 1...16 die LEDs 0...15; Die Einheit mit Adresse ‚1‘ schaltet bei LED-Zählerstand 17...32 die LEDs 0...15, und so weiter.
Auf diese Weise können alle LEDs sukzessive durchprobiert werden.

Ein <CR> Zeichen beendet den Testmodus

Links

Diese Links verweisen auf frei verfügbare Programmpakete

WinAVR

Entwicklungsumgebung: C-Compiler, Assembler, Linker etc.

Dieses Programm ist notwendig, wenn die Quelltexte geändert und neu übersetzt werden sollen.

<http://winavr.sourceforge.net/>

AVRStudio

Simulator, Debugger, Projektverwaltung

Auch dieses Programm ist notwendig, wenn die Quelltexte geändert und neu übersetzt werden sollen. Das Programm WinAVR kann hier eingebunden werden.

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

BrayTerminal

Terminal-Programm, Ersatz für Hyperterminal

Dieses Programm ist nicht unbedingt notwendig, aber es erleichtert die Arbeit ungemein, wenn der Ablauf des LED-Leuchtens per Parametrierung angepasst werden soll

<http://sites.google.com/site/terminalbpp/>

AT AVR ISP: Programmierinterface

AVR-Starterkit STK500

Das AT AVR ISP ist nur zum Programmieren des Microcontrollers zu verwenden.

Das STK500 ist ein Entwicklungsboard das auch zur Programmierung verwendet werden kann.

Wer weitere Microcontroller-Programme erstellen und ausprobieren möchte kann das STK500 verwenden; wer nur die bestehende Software verwenden möchte wie sie ist, kann das Programmierinterface verwenden.

Quelltexte

Das Programm besteht aus folgenden einzelnen Quelltexten:

Stahlwerk.c
stahlwerk.data.c
UART.c
utilities.c
eeprom.s
PWMloop.s

stahlwerk.h
UART.h
utilities.h
EEProm.h
PWMENTRY.h

ioex.h
iom8ex.h
macros16.inc

III. Stückliste für eine Platine

| Anzahl | Bezeichnung | | |
|--------|---------------------------------------|----------------|--|
| 1 | Microcontroller | ATmega 8L8 DIP | |
| 1 | Operationsverstärker | LT 1013 DI | |
| 1 | Spannungsregler | 7805 CT | |
| 1 | RS 232 Schnittstelle | | nur auf der ersten Platine |
| 1 | Pegelwandler | MAX 232 A | nur auf der ersten Platine |
| 4 | Kondensator (Stützko) | 0,1 µF | nur auf der ersten Platine, für MAX 232 A |
| 1 | Kondensator (Stützko) | 1 µF | nur auf der ersten Platine, für MAX 232 A |
| 1 | LED grün | | Kontrollleuchte |
| 1 | Stiftenleiste 2 polig mit Jumper | | für Block LEDs |
| 2 | Stiftenleiste 2x3 polig | | Anschluss des Tasters und Programmierung der Microcontroller |
| 1 | Stiftenleiste 1 polig | | nur auf der ersten Platine, Masseanschluss zum Messen |
| 3 | Stiftenleisten abgewinkelt, 2x3 polig | | nicht auf der letzten Platine |
| 3 | Buchsenleisten abgewinkelt, 2x3 polig | | nicht auf der ersten Platine = Steckverbindungen der Platinen |
| 1 | Diode | 1 N 4148 | |
| 1 | Spindel-Trimmpoti | 10k Ohm | verzichtbar |
| 17 | Widerstand | 100 Ohm | |
| 17 | Widerstand | 100kOhm | |
| 1 | Widerstand | 510 Ohm | |
| 3 | Widerstand | 10 kOhm | |
| 2 | Widerstand | 5 kOhm | |
| 1 | Widerstand | 750 Ohm | |
| 2 | BlockLED | z.B. HDSP 4840 | LED Leuchtbalken Array |
| 18 | Transistor | BC 547 | |
| 18 | Transistor | BC 557 | |

IV. Erläuterungen zum Schaltplan

Auf die RS 232 Schnittstelle kann verzichtet werden, dann entfällt allerdings die Kommunikation mit den Microcontrollern, dh., dass beispielsweise eine Änderung der Zeitparameter nicht möglich ist.

Wir haben den Schaltplan mit Eagle erstellt, dieses kann als Freeware heruntergeladen werden. Der Schaltplan wird als PDF und alternativ als Eagle Datei angeboten. Wer damit arbeitet, muss gegebenenfalls die Lizenzbedingungen von Eagle beachten. Die in den Bibliotheken vorhandenen Symbole haben wir weitgehend unverändert verwendet. Die Platzierung der Symbole auf dem Schaltplan ist im Wesentlichen an den tatsächlichen Gegebenheiten orientiert, um die Zuordnung zu erleichtern. Die übliche professionelle Abstrahierung fehlt dem Plan damit natürlich. Im Grunde haben wir zwei Pläne in einem integriert, da die Steckverbindungen links auf der ersten Platine fehlen, während die RS 232 Schnittstelle und der MAX 232 nur auf der ersten Platine vorhanden sind.

Auf jeder Platine ist ein Anschluß für einen Taster vorgesehen (unten in der Mitte). Der verwendete Taster hat einen rot/grünen Lichtwechsel, um dem Besucher anzuzeigen, wann der Abstich neu gestartet werden kann. Grundsätzlich kann der Anschluß beliebig auf einer der drei Platinen frei ausgewählt werden, es ist nur eine Verbindung notwendig. Die Steuerung des Rot/Grün Wechsels auf der Platine verwenden wir zwischenzeitlich nicht mehr, da wegen des Sounds ohnehin eine zusätzliche Schaltung eingebaut werden mußte. Wir haben diese auch für den Rot/Grün Wechsel vorgesehen.

Noch einige Hinweise für Nicht-Elektronik-Profis: Die Symbole der Bauteile stimmen in aller Regel nicht mit den physikalischen Gegebenheiten überein. Der ATmega 8L hat beispielsweise tatsächlich zwei Reihen zu je 8 Pins (Anschlüssen), die sich symmetrisch gegenüberstehen und von 1 bis 16 durchnummeriert sind. Das Symbol des LT 1013 hat keinerlei Ähnlichkeit mit dem Bauteil, wird es doch durch ein Dreieck dargestellt. Wir haben die SMD Version verwendet und diese zunächst auf einen Sockel gelötet. Auf dem Bauteil stehen sich je 4 Pins gegenüber. Hilfreich sind in aller Regel die Datenblätter der Hersteller, dort ist die Belegung dargestellt. Der ATmega stammt von Atmel, der Max 232 ist bei Maxim zu finden, beim LT 1013 handelt es sich um ein Produkt von Texas Instruments.

Beim Stromanschluß links oben (9V Gleichstrom) ist natürlich auf die richtige Polarität zu achten, werden plus und minus vertauscht, leuchten die LEDs nicht. Der einpolige Pin daneben ist mit Masse verbunden und dient dem Anschluß eines Meßgeräts zur Fehlersuche.

Der Max 232 ist auf dem Symbol von Eagle nur mit 14 Pins dargestellt, wir haben die Nummern 15 und 16 zeichnerisch ergänzt, da beide Anschlüsse benötigt werden. Die eigentliche Steuerung der LEDs (zwischen Microcontroller und LED) ist beispielhaft nur für eine LED eingezeichnet (LED 1), sie ist natürlich 16 mal auf einer Platine vorhanden. Die Nummerierung der LEDs beginnt mit „0“.

Vorsicht ist auch bei den Symbolen der Transistoren geboten. Der 547 ist ein npn Transistor, der 557 dagegen ein pnp. Kollektor und Emitter sind vertauscht. Ein Blick in die Datenblätter kann nicht schaden (zB. Datasheetcatalog.com).

Die Block LEDs sind nur pro forma am oberen Rand dargestellt, die Kathode ist an Masse (Verbindung kann über den Jumper unterbrochen werden) anzuschließen, die Anode ist über die Steuerung mit dem ATmega 8L verbunden. Bei den Symbolen der Steckanschlüsse sind je zwei Pole zeichnerisch verbunden, dies ist real nicht gegeben.

Die Widerstände:

R 1 5 kOhm
R 3 100 kOhm
R 4 510 Ohm
R 5 10 kOhm
R 6 10 kOhm
R 7 10 kOhm
R 8 100 Ohm (16 Stück)
R 9 100 kOhm (16 Stück)
R 10 750 Ohm
R 11 5 kOhm
R 13 100 Ohm

V. Installationsanleitung für die Programme

Die Programme sind Textdateien, die kopiert werden können. *Wird noch weiter ausgeführt.*

Verbindung des Boards mit der Platine: *Wird noch weiter ausgeführt*