

Roboter-Nachrichten 03 / 2014

Vorwort:

Liebe Leser, dies ist die dritte Ausgabe der Roboter-Nachrichten und es ist "Sommer". Sommer, die Roboter stehen in einer Ecke oder in einem Regal und verstauben.

Richtig? NEIN !

die Robotniks (Gerhard, Klaus und ich) haben alle Hände voll zu tun, denn wir haben uns entschieden erstmals als Aussteller auf eine Messe zu gehen, Inspiration Modellbau in Mainz 20/21 Sept. 2014 <http://inspiration-modellbau.de/>. Es gibt also sehr viel vorzubereiten, ein Messestand muss gebaut werden, ein Info-Flyer muss her, die Roboter müssen programmiert werden und, und, und ...

Wir freuen uns, wenn Sie uns auf dem Messestand besuchen werden, um Fachgespräche oder einfach nur „small-talk“ zu führen.

Einen kleinen „Vorgeschmack“ finden Sie in diesem Newsletters.

Inhaltsangabe:

- **Noch ein Hinweis zur Test- und Programmierplatine**
- **Der I²C Bus, eine Einführung**
- **Änderung der Spannungsversorgung der Motoren**
- **Messevorbereitungen**
- **Vorstellen des laufenden Projektes: Wegsuche durch ein Labyrinth**
(wird auf Newsletter Q4 verschoben, Aufgrund der Messevorbereitungen)
- **Kolumne von Klaus Wellmann**
(Die optische Täuschung, oder schau mal ins Datenblatt...)
- **Ausblick auf die nächste Ausgabe des Newsletters**
(Newsletter Q4 2014 wird erst nach der Messe, ca. Oktober / November erscheinen)

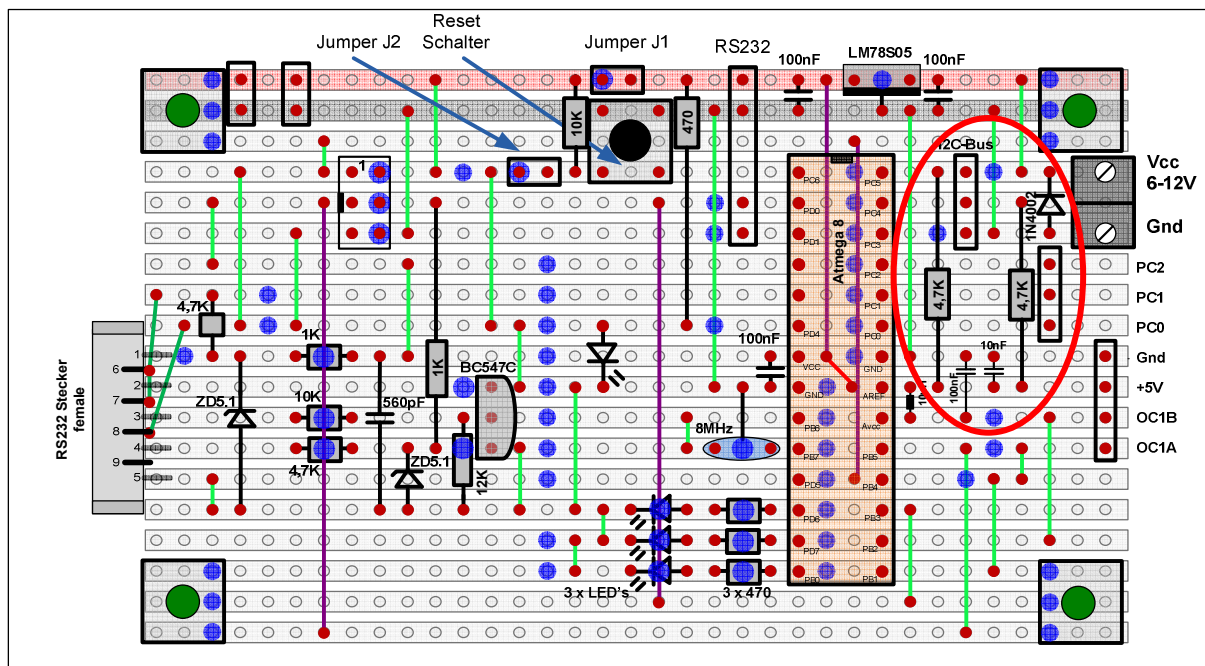
Noch ein Hinweis zur Test- und Programmierplatine:

Die Test- und Programmierplatine lässt uns einfach nicht zur Ruhe kommen. Das liegt wohl daran, dass wir die Platine ständig für neue Projekte benötigen. Auch die Software und Hardware für die Messevorbereitung wird immer erst auf dieser Platine getestet, bevor die Software auf die Roboter geladen wird. Der Roboter hat keine weitere Ausgabemöglichkeit als die Motoren. Es fehlen LEDs, die z.B. einen Status ausgeben könnten. Zurzeit arbeiten wir an der Datenübertragung von PC zum Roboter über WLAN-Module und dafür benötigen wir den I²C-Bus, da an diesem Bus das WLAN-Modul angeschlossen wird. Genauer beschreiben werden wir diese Art der Datenübertragung in einem der nächsten Newsletter. Wer schon jetzt etwas darüber erfahren möchte kann auf www.ps-robotics.de den Bericht über das WLAN-Modul von AVISARO lesen.

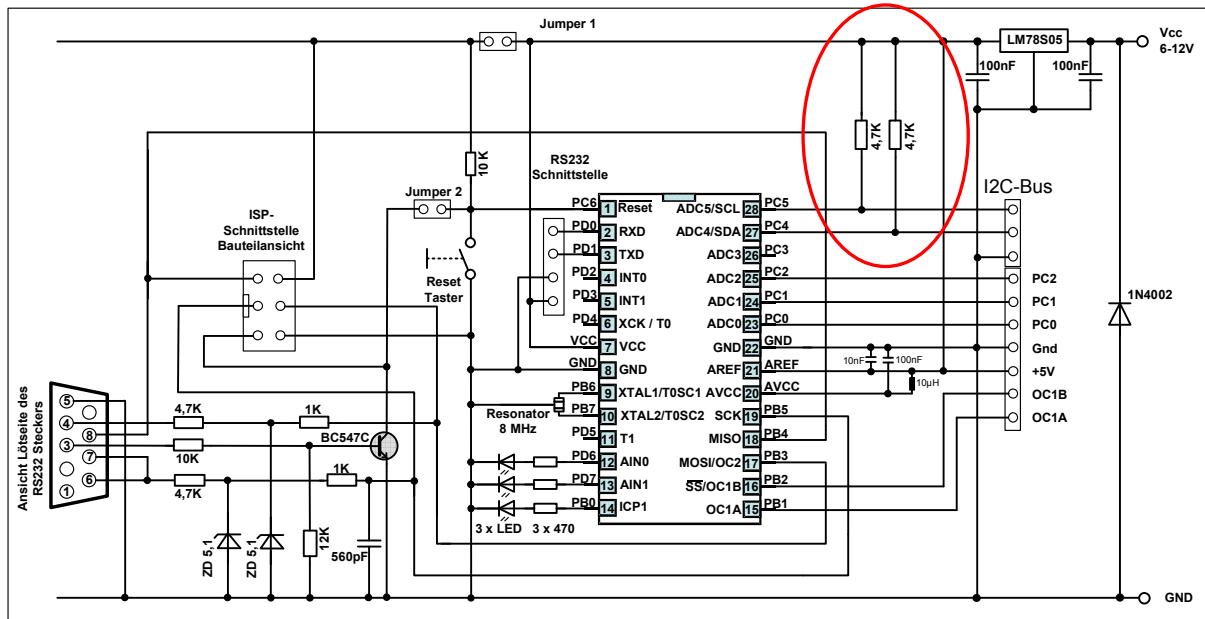
Also, man nehme ein WLAN-Modul schließe es an und nichts geht. Der übliche Effekt, wenn man etwas Neues ausprobiert. Was dieses Mal fehlte, waren die Abschlusswiderstände von 2 x 4,7k für den I²C-Bus. Aus diesem Grund, und damit uns das nicht nochmal passiert, haben wir uns entschlossen, nochmals eine Veränderung an der Test- und Programmierplatine vorzunehmen und mit zwei Abschlusswiderstände (siehe Schaltplan) zu ergänzen. Auf der Steuerplatine ist leider kein Platz mehr, sonst hätten wir diese Veränderung auch an dieser Platine vorgenommen.

Die nächsten zwei Abbildungen zeigen die geänderte Platine (siehe Markierung) in der Bestückungsansicht und den dazugehörigen Schaltplan.

Bestückungsplan Test- und Programmierplatine



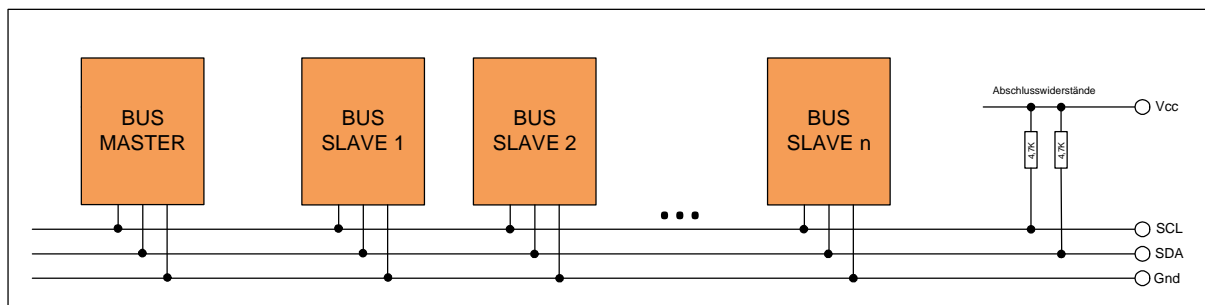
Schaltplan Test- und Programmierplatine



Der I²C Bus, eine Einführung

Wir möchten hier nur einen kurzen Einblick geben, wofür man einen I²C-Bus nutzen kann und wie man diesen Bus ansteuert mittels Programmierung. Es wird hier nicht auf Timing (Zeitverhalten / Laufzeiten der Signale) des Busses o.ä. eingegangen, da dieses viel zu weit führen würde. Ein kleines Schaubild soll die Wirkungsweise des Busses erklären, aber zuvor möchte ich ein paar mögliche Anwendungen aufführen. Genutzt wird der Bus hauptsächlich in der Industrie, aber auch in HIFI Geräten oder in Computern findet man diesen Bus, da er sehr einfach ist und nur zwei Leitungen benötigt. Eigentlich, strenggenommen drei, wenn man Gnd mitzählt. Eine klassische Anwendung für unsere Roboter ist das WLAN-Modul, oder der elektronische Kompass, der auch über I²C-Bus angesteuert bzw. abgefragt wird. Auch wenn man zwei Mikrocontroller miteinander kommunizieren lassen möchte, bietet sich dieser einfache Bus ebenfalls an.

Hier ein Schaubild zum I²C-Bus:



Der I²C-Bus, oft auch I2C geschrieben ist ein synchroner, serieller Datenbus mit einem MASTER und mehreren SLAVES. Es muss mindestens einen Bus-Master geben, der aber alle Aufgaben, genau wie die bis zu 127 Slaves erfüllen kann. Ein Slave kann keine Daten senden, sondern stellt Daten dem Master zur Verfügung, die der Master dann beim Slave abholt.

Jetzt kann man sich das Leben richtig schwer machen und mittels Datasheet und Busbeschreibung eine Programmierung erarbeiten, oder wie wir das normalerweise tun, wir "googeln". Unter I²C-Bus und Peter Fleury wird man schnell fündig: <http://homepage.hispeed.ch/peterfleury/avr-software.html>. Einfach mal lesen und die entsprechenden, fertigen Libraries in das eigene Programm einbinden. Dort findet man auch tolle Libraries für die UART (RS232) Ansteuerung, LCD Display und Bootloader. Es lohnt sich also einen Abstecher auf die Webseiten von Peter Fleury zu unternehmen.

Hier ein minimales Beispiel für die Ansteuerung eines Kompassmoduls:

D.h. das Programmbeispiel würde auf dem Bus-Master laufen z.B. einen ATmega8 der einen Kompass ansteuert

```
1: #include <i2cmaster.h> // Nicht vergessen, Library von Peter Fleury einbinden
2:
3: #define I2C_READ 1 // Binaer eine 1 addieren fuer lesen von device
4: #define I2C_WRITE 0 // Binaer eine 0 addieren fuer schreiben auf device
5:
6: #define kompass 0x33 // device address des kompass (Beispiel Adresse)
7:
8:
9: int main(void)
10: {
11:     int data; // Variable fuer Aufnahme der Kompassdaten
12:
13:     i2c_init(); // initialize I2C bus
14:
15:     // Kompass auslesen
16:     i2c_start_wait(kompass + I2C_WRITE); // setzen device adresse und "write mode"
17:     i2c_write(0x01); // write address gleich register 1 des kompass
18:
19:     i2c_rep_start(kompass + I2C_READ); // setzen device adresse und "read mode"
20:
21:     data = i2c_readNak(); // read one byte from kompass aus register 1
22:     i2c_stop(); // I2C Bus wieder freigeben
23:
24:     for(;;); // Endlos-Schleife
25: }
```

Wichtig ist, jedes Device (Gerät), das den I²C-Bus benutzt hat eine eigene Adresse. Diese kann frei wählbar sein, oder auf fest verdrahtet in der Hardware verankert sein. Im obigen Beispiel lautet die Adresse des Devices 0x33, das angesprochen werden soll, ein Kompassmodul (siehe Zeile 6). Der Adresswert basiert auf einer 8-Bit Adresse, die meistens in hexadezimaler Schreibweise angegeben wird. Möchte man auf ein Device lesen oder schreiben, so muss man erst die Adresse des Devices angeben, inkl. was man tun möchte. In unserem Fall **lesen** von dem Device. Um einen Leseprozess durchzuführen, muss man aber als ersten Schritt dem Device mitteilen aus welchem Register gelesen werden soll. Eine Mitteilung ist aber ein schreibender Prozess. Das erklärt auch, warum in Zeile 16 zuerst auf das Device geschrieben wird, nämlich bitte aus Register 1 lesen. Anschließend wird erst der eigentliche Leseprozess gestartet. Der Programmcode „+ I2C_WRITE“ wird in Zeile 3 und 4 erklärt und bedeutet, dass bei Schreibprozessen eine „0“ und bei lesenden Prozessen eine „1“, **binär** zur Adresse des Devices hinzugefügt werden muss. Durch den Befehl „i2c_start_wait“ wird dem Bus mitgeteilt, dass man auf ein Device zugreifen will und den Bus für einen schreibenden Prozess übernehmen möchte. Der Bus bleibt dann für andere Devices gesperrt, bis er wieder freigegeben wird. In Zeile 19 signalisiert man dem Bus, dass man von einem Device lesen möchte. Anschließend wird der Registerinhalt aus Register 1 ausgelesen und in die Variable „data“ geschrieben. Nicht vergessen, den Bus wieder freigeben für andere Devices (Zeile 22).

Fazit:

Der I²C-Bus stellt eine sehr einfache, schnelle und effektive Kommunikation zwischen Devices zur Verfügung.

Änderung der Spannungsversorgung der Motoren

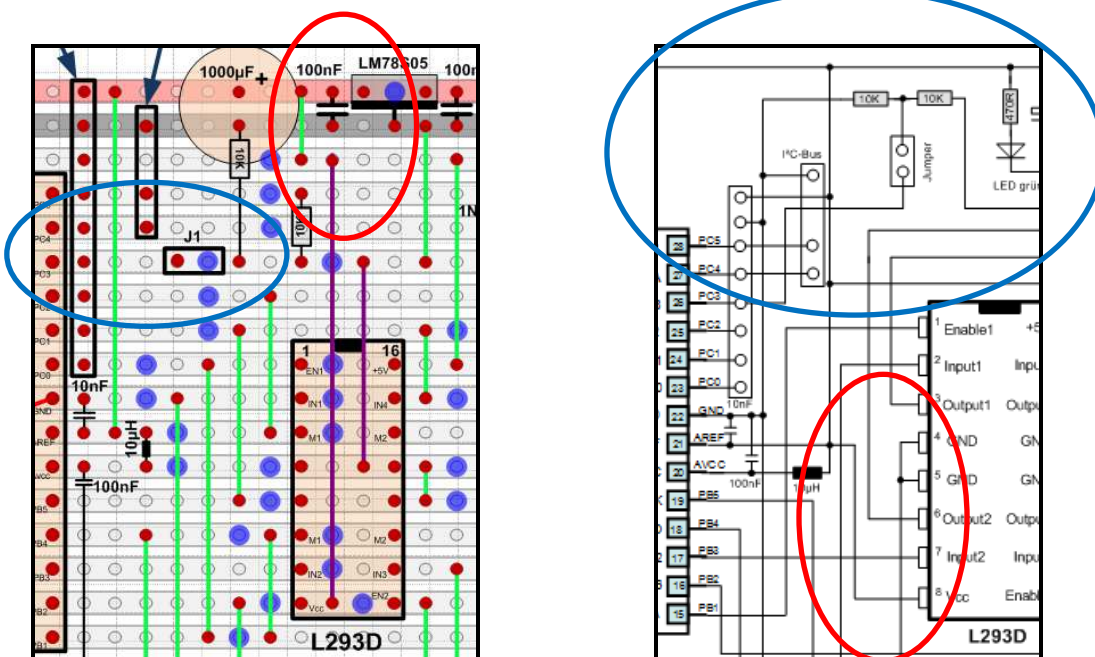
Falls Sie den R2PT3 Roboter bereits nach Anleitung im Buch, oder Bauanleitung des Bausatzes von Schneider-Engineerings gebaut haben, dann könnte Ihnen aufgefallen sein, dass bei nachlassender Akkuspannung der Roboter immer stärker von der Geradeausfahrt abweicht und natürlich auch langsamer fährt.

Einschub : Zu einem späteren Zeitpunkt werden wir erklären, wie man mit den Impulsgebern eine exakte Fahrt geradeaus realisiert.

Hier erst mal eine einfache Lösung, in dem wir die Spannung der Motoren nicht mehr direkt von dem Akku erhalten, sondern den Umweg über den Festspannungsregler wählen. Zurzeit erhält der Motortreiber L293D seine Spannung direkt vom Akku und deshalb wird bei fallender Akkuspannung die Spannung an den Motoren abfallen. Die Lösung ist einfach, wir trennen die bisherige Leitung auf und verlegen mittels flexiblem Draht die Spannungsversorgung Vcc Pin8 des IC L293D an die +5V Festspannung der Steuerplatine.

Wenn Sie ganz genau hinsehen: Wir haben noch einen weiteren Jumper hinzugefügt und den Spannungsteiler mit den zwei 10K Widerständen vom Pin PC3 des Mikrocontrollers getrennt. Folglich wird der Pin PC3 wieder für andere Anwendungen nutzbar. Eigentlich wollten wir die Spannung des Akkus messen und mit Analog/Digital-Wandler (siehe Ausgabe Q2 des Newsletters) auswerten, aber für unsere zukünftigen Projekte wird dieser Port des Mikrocontroller dringender benötigt. Oh ja, Ideen haben wir viele, nur Zeit fehlt uns für die Realisierung ...

Hier nun der Bestückungs- und Schaltplanausschnitt.



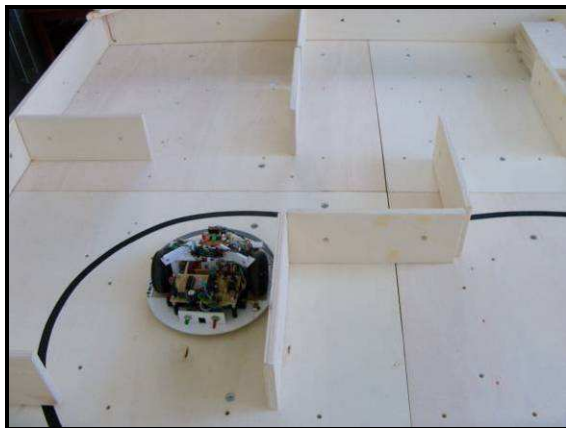
Messevorbereitungen

Auf der Messe möchten wir gerne die Roboter auf Basis R2PT3, in den verschiedenen Ausbaustufen vorstellen. Das heißt, dass wir einen Roboter zeigen werden im Original und zwei weitere Roboter, die mit verschiedenen Sensoren ausgerüstet sind. Wir wollen präsentieren, wie ein Roboter sich selbständig durch ein Labyrinth findet und ein weiterer Roboter wird auf der Platte des Messestandes einer durchgezogenen Linie folgen. Wenn alles klappt, zeigt Beispiel Drei, wie man Befehle über Funkmodule von PC an den Roboter sendet und dieser bestimmte Positionen auf dem Messestand selbstständig ansteuert, d.h. der Roboter betreibt also eine Navigation. Einen Hexabot (Sechsfüßler) werden wir ebenfalls dabei haben und in Funktion (Action) zeigen können.

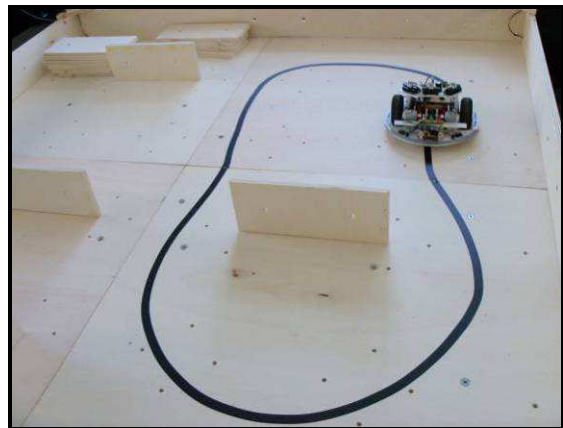
Außerdem werden wir den R2PT3 Roboter in verschiedenen Fertigstellungsstufen und alle zusätzlichen Erweiterungsplatinen präsentieren.

Alles zum anschauen und anfassen!

Hier unser Messestand:



Fahrt durch ein Labyrinth



Einer Linie folgen

Falls noch Buchexemplare benötigt werden, so sind diese käuflich auf dem Messestand erhältlich, so lange der Vorrat reicht. Auch ein paar gebrauchte Roboter haben wir dabei, falls man den günstigen Einstieg in die Roboterwelt wagen möchte. Gerne tauschen wir Adressen und „Know How“ aus und vielleicht finden sich ja noch ein paar neue Robotniks.

Also die Typen, die sich riesig freuen, wenn der Mikrocontroller eine LED an- und ausschaltet und das ganz von alleine 😊😊😊.

Kolumne von Klaus Wellmann (Die optische Täuschung, oder schau mal ins Datenblatt...)

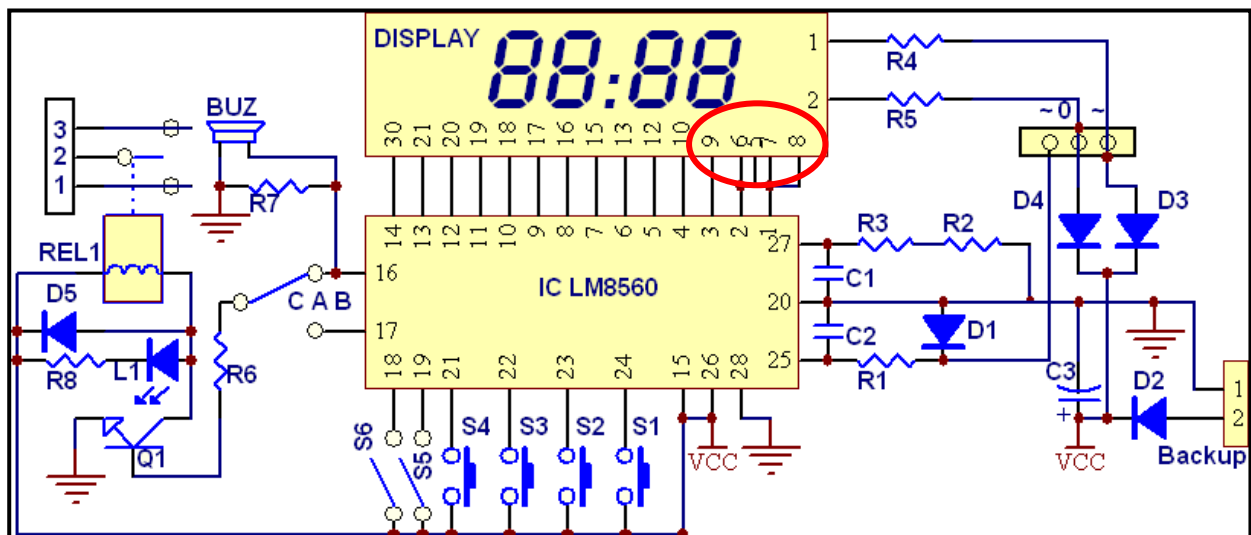
Aus einem alten Radiowecker habe ich ein LED-Display ausgebaut - vielleicht auch gerade deswegen, weil es so herrlich altmodisch aussah... Aber was damit anfangen? Die näheren Bauteile um das Display auf der Platine sahen „verschmort“ aus und somit wenig vertrauensvoll. Trotzdem, irgendwie so dachte ich mir, es muss doch möglich sein die Uhr (inkl. dem Display) in einer neuen Schaltung zu reaktivieren! Wie bin ich vorgegangen?

Das Steuerungs-IC mit direkten Anschlüssen zum Display trägt die Bezeichnung: TMS3450NL. Unter: „google“, habe ich die Typenbezeichnung eingegeben und fand eine Schaltung, die gar nicht so furchtbar kompliziert aussah. <http://www.elfrum.ro/topic/25173-ceas-cu-tms3450nl/>. Meine Kenntnisse der rumänischen Sprache sind gleich Null, aber der Satz *...afiat ca TMS3450NL e echivalent cu LM8560...* ließ mich aufhorchen! Verschiedene ICs mit der Kennzeichnung LM plus irgendeine Zahlenfolge habe ich schon einmal im Katalog gesehen.

Richtig: Gibt man diese Bezeichnung ein, dann findet man einige Artikel, Baubeschreibungen und Berichte von verschiedenen Bastlern. Als besonders gut und informativ möchte dabei die Ausführungen von Gerald Penz www.elektronik-kompodium.de/ und www.ferromel.de/tronic305.htm/ hervorheben. Trotzdem ist die Beschreibung der 14 Übergänge, der Pins vom IC zum Display, mit seinen 30 Anschlüssen etwas zu „knapp“ formuliert. Zitat *...so etwa sollte es anzuschließen sein...* oder Zitat von ferromel: *...weiteres benötigt das IC ein spezielles sog. Duplex-LED-Display mit zwei gemeinsamen Kathoden, das nicht gegen übliche Siebensegmentanzeigen austauschbar ist.*

Aha, das muss also etwas Besonderes sein.

Da auf dem Display keine Artikelkennzeichnung steht habe ich beim Zusammenlöten immer wieder auf die Veröffentlichung der rumänischen Bastelkollegen geschaut. Bitte überzeugen Sie sich selbst:



Der Pin 3 vom IC LM 8560 ist der Nummer 9 vom Display zugeordnet
Der Pin 2 vom IC LM 8560 ist der Nummer 6 vom Display zugeordnet und teilt sich auf in eine Zahl, welche wie eine „verunglückte“ sieben aussieht.



Quelle: google : Zahl 7 , Bilder

Der Pin 1 vom IC LM 8560 ist der Nummer 8 vom Display zugeordnet und teilt sich auf in eine weitere Zahl, welche EBENFALLS, wie eine „verunglückte sieben aussieht. Die Fachleute unter Ihnen werden sofort anmerken, dass eine Ansteuerung von einem IC auf zwei identischen Eingangspins eines Displays nicht möglich ist.

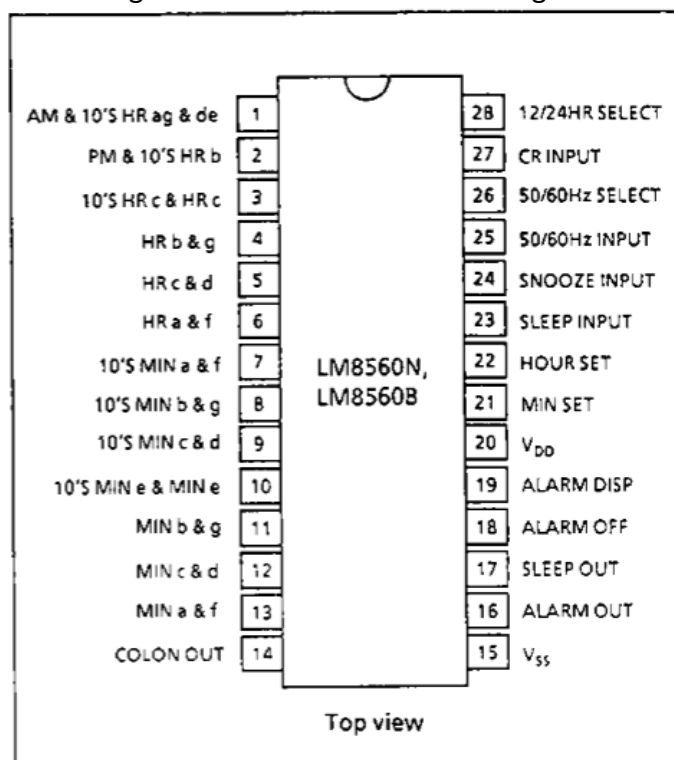


Aber, welche Zahl oder Zahlen sind aber die Richtigen?

Oder anders gefragt: Wer sieht in dem beiliegenden Bild eine junge Frau oder eine alte Frau?

Quelle : quizfragen4kids.de, optische Täuschungen

Die Lösung des Rätsels findet sich im Original-Datenblatt des LM 8560 N:

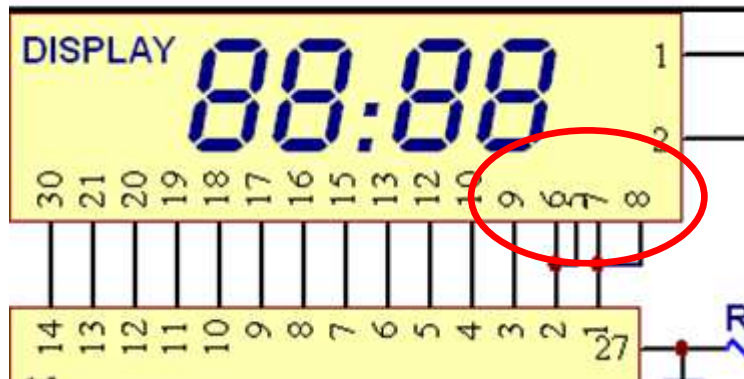


Pin 2 = PM & 10's HR b = übersetzt : LED für Nachmittag im 12 Stunden Modus und Stundenanzeige und das Segment b

Pin 1 = AM & 10's HR ag & de = übersetzt : LED für Morgen im 12 Stunden Modus und Stundenanzeige für die Segmente a , g , d , e

Schaut man mit diesem zusätzlichen Wissen wieder auf die Kopie der 30 poligen Ansteuerung des Displays unserer rumänischen Kollegen + auf die obige Detailvergrößerung, erkennt man die korrekte Ziffernfolge zum Anlöten.

Bild: Detailvergrößerung Display



Fazit: Optische Täuschungen sind gemein und tückisch, wer in der Wüste einer Fata Morgana folgt, wird sein Ziel (die Wasser spendende Oase) nie finden und wer in der Hobbyelektronik den Ansteuerungspin 5 nicht von der Nummer 7 unterscheiden kann, der läuft ebenfalls in die „Irre“.

Ausblick auf die nächste Ausgabe des Newsletters

- Weiter Korrekturen des Buches
- Vorstellen des Projektes: Weg suche (siehe Oben) durch ein Labyrinth
- Die serielle Schnittstelle RS232 oder auch UART genannt

Mit freundlichen Grüßen

Die Robotniks (Gerhard, Klaus und Peter)

www.ps-robotics.de/

Dieser Newsletter enthält Links zu externen Webseiten Dritter, auf deren Inhalte wir keinen Einfluss haben. Deshalb können wir für diese fremden Inhalte auch keine Gewähr übernehmen. Für die Inhalte der verlinkten Seiten ist stets der jeweilige Anbieter oder Betreiber der Seiten verantwortlich. Die verlinkten Seiten wurden zum Zeitpunkt der Verlinkung auf mögliche Rechtsverstöße überprüft. Rechtswidrige Inhalte waren zum Zeitpunkt der Verlinkung nicht erkennbar. Eine permanente inhaltliche Kontrolle der verlinkten Seiten ist jedoch ohne konkrete Anhaltspunkte einer Rechtsverletzung nicht zumutbar.