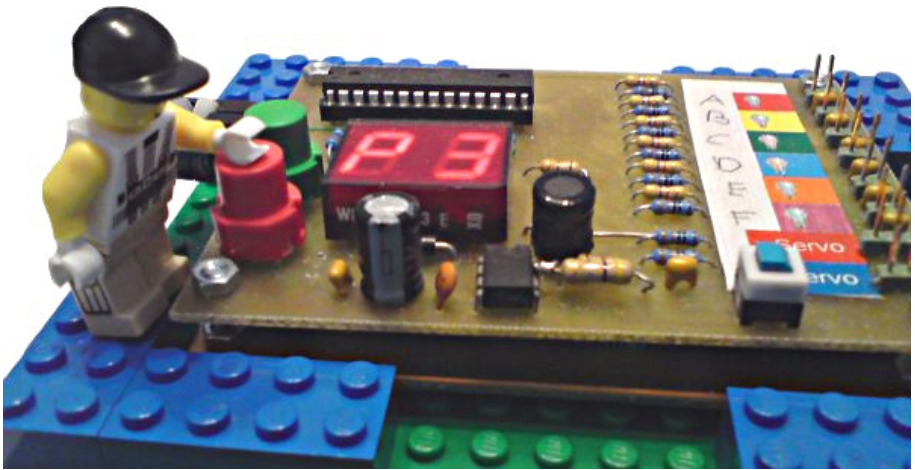


LBlocks:

Programmieren ohne Tastatur



Technisches Handbuch

(Version 3 / Rev. A)

(c) 2012-2013 Jörg Wolfram

Inhaltsverzeichnis

1	Allgemeines	2
1.1	Allgemeines	2
2	Die Software	3
2.1	Installation	3
2.2	Anforderungen	3
2.3	Sourcecode	4
3	Die Hardware	5
3.1	Konzept	5
3.2	Der Controller	6
3.3	Leiterplattenlayout	7
3.4	Die Verbindung zur Aussenwelt	8
3.4.1	Mechanischer Aufbau	8
3.4.2	Lampen (LEDs)	9
3.4.3	Taster	9
3.4.4	Fotowiderstand	10
3.4.5	Reedkontakte	10
3.4.6	Servos	11
4	Changelog	12

Kapitel 1

Allgemeines

1.1 Allgemeines

Die Programme unterliegen der GPL (GNU General Public Licence) Version 3 oder höher, jede Nutzung der Software/Informationen nonkonform zur GPL oder ausserhalb des Geltungsbereiches der GPL ist untersagt!

Die Veröffentlichung dieses Projekts erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGEND EINE GARANTIE, auch ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK.

Alle im Text genannten Marken sind Eigentum des entsprechenden Inhabers.

Kapitel 2

Die Software

2.1 Installation

Das Programm ist monolithisch aufgebaut, das heisst es besteht nur aus einer einzigen Datei. Damit ist eine Installation im eigentlichen Sinne nicht unbedingt nötig, es reicht aus, das Programm ausführbar zu machen und zu starten. Aus diesem Grund habe ich auch auf die Erstellung von Paketen verzichtet.

Das Programm benötigt Schreibrechte im home-Verzeichnis des Benutzers, dort wird beim ersten Start ein Ordner mit Namen **lblocks** angelegt. In diesem befindet sich auch das Projektverzeichnis zum Laden und Speichern von Projekten und ein Verzeichnis mit den nötigen Scripten.

2.2 Anforderungen

Das Programm ist unter und für Linux entwickelt worden, auch unter anderen freien Betriebssystemen (BSD) sollte es recht einfach sein, das Programm zum Laufen zu bringen. Generell ist ein grafische Grundsystem (X11) sowie die **libpng** notwendig, als Toolkit wird ein modifiziertes GTK 1.2 (ist statisch im Programm eingebunden) verwendet. Binaries stehen zur Zeit für x86 (PC-32Bit) und ARM (Raspberry PI) zur Verfügung. Die ARM-Variante wurde allerdings nur in einem Emulator getestet.

Um das Programm in den Controller schreiben und von dort lesen zu können, ist eine Programmer-Software notwendig. Per Default werden bei der Installation **avrdude** und als Programmer der **usbasp** eingerichtet. Das Script **readscript_xx.sh** liest den Flash-Speicher des Controllers in die Datei „lblocks_read.hex“ ein und das Script **progscrip_t_xx.sh** schreibt den Inhalt der Datei „lblocks.hex“ in den Speicher des Controllers.

2.3 Sourcecode

Das Projekt ist unter der GPL V3 lizenziert, was die Verfügbarkeit des Sourcecodes einschließt. Dieser kann in einem separaten Archiv heruntergeladen werden. Der gesamte Compiliervorgang wird durch Scripte gesteuert, Makefiles habe ich nicht benutzt. Neben dem C-Compiler (GCC) sind noch AVRA für die Übersetzung des AVR-Quellcodes und ein Perl-Interpreter notwendig. Ebenso sind die gepatchten Dateien für GTK1.2 mit enthalten.

Kapitel 3

Die Hardware

3.1 Konzept

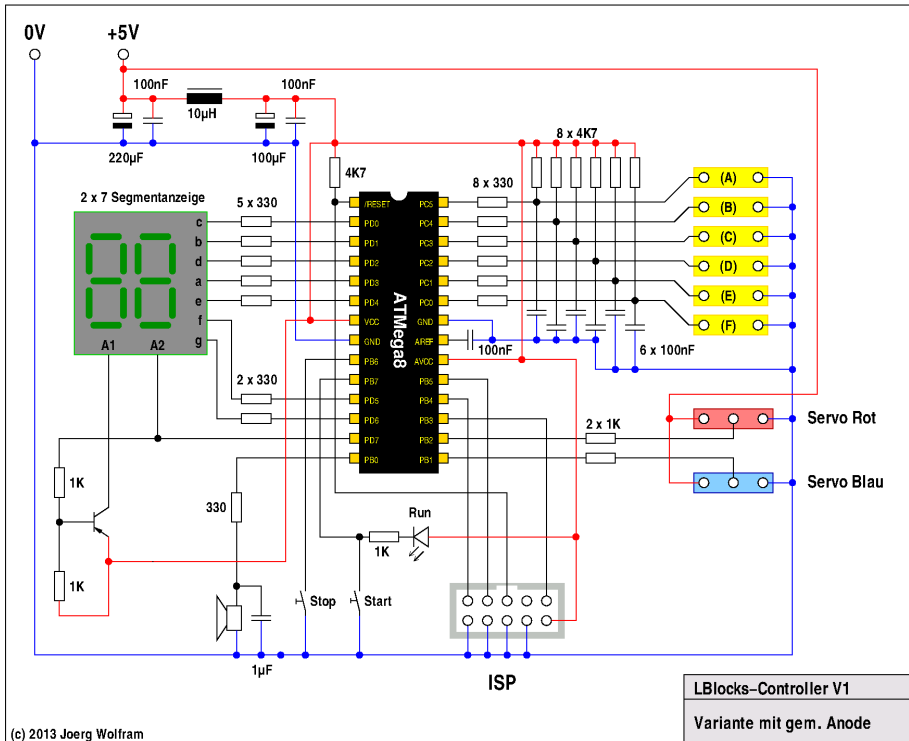
Kernstück des LBlocks-Controllers ist ein ATmega8 oder ATmega88. Der ATmega88 ist aber zur Zeit noch nicht getestet. Für die Stromversorgung kommen sinnvollerweise Akkus in Betracht, ein Beispiel sind 4 NiMH-Zellen in Reihe, die eine Nominalspannung von 4,8V liefern. Der Controller wird über eine Siebglie mit Drossel versrgt versorgt, während die Servos direkt am Akku hängen. Dadurch sollen Rückwirkungen von den Motoren auf den Rest der Schaltung weitgehend vermieden werden.

Eine andere Variante die ich selbst benutze, ist ein alter LiIon- oder LiPo-Akku mit einem Step-Up Schaltregler um auf 5V zu kommen. Hier sind aber zusätzliche Maßnahmen gegen Kurzschlüsse zu ergreifen und es braucht ein spezielles Ladegerät.

Auf dem Board existiert dazu noch eine zweistellige Siebensegmentanzeige, eine LED und zwei Taster. Ebenso ist ein Minilautsprecher vorgesehen. Für externe Peripherie gibt es 6 universelle Ein- / Ausgänge und 2 Servo-Impulsausgänge. Für die Siebensegmentanzeigen lassen sich sowohl welche mit gemeinsamer Anode als auch mit gemeinsamer Kathode verwenden, die Beschaltung wird durch das Programm automatisch erkannt.

3.2 Der Controller

Die Beschaltung des Controllers beschränkt sich auf das notwendige Minimum:



Rechts sind die 6 „Universalanschlüsse“ (A) - (F) mit gelbem Rahmen dargestellt, darunter die beiden Servoanschlüsse. Links ist 2-stellige Siebensegmentanzeige symbolisch dargestellt. Das rechte Digit wird direkt, das linke über einen als Inverter dienenden Transistor angesteuert. Nach dem Einschalten wird der Pegel an PORTD.7 abgefragt und daraus bestimmt, ob es sich um Anzeigen mit gemeinsamer Anode (H-Pegel an PORTD.7) oder um solche mit gemeinsamer Kathode (L-Pegel an PORTD.7) handelt.

3.4 Die Verbindung zur Aussenwelt

Als Verbinder haben sich 2- oder 3-polige Steckerleisten im 2,54mm Rastermaß auf der Leiterplatte bewährt. Das Gegenstück sind dann kurze Abschnitte von Wire-Wrap Buchsenleisten. Das ganze Konzept ist so ausgelegt dass weder der Controller noch die Peripherie bei einer Verpolung Schaden nehmen können, eine farbliche Markierung bei verpolbaren Stecksystemen ist aber auf jeden Fall sinnvoll. Denn z.B. die LEDs und Servos funktionieren natürlich nur bei der richtigen Polung.

Als Grundausstattung für die Experimente wird Folgendes benötigt:

- LED seitlich in den farben rot, gelb, grün und weiss
- 2 Taster
- 1 Fotowiderstand
- 2 Reedkontakte, dazu die passenden Magnete
- 1 Getriebemotor mit Schnurrolle

Das Ganze kann dann noch durch einen zweiten Motor, zweiten Fotowiderstand und z.B. Mikrotaster erweitert werden.

3.4.1 Mechanischer Aufbau

Ich habe für den Aufbau LEGO-Steine benutzt, da unser Sohn genug davon hat und sich die Bausteine so auch recht einfach in „Gesamtkunstwerke“ einfügen lassen. Die Auswahl der Bauelemente erfolgte hauptsächlich nach Bastelkisteninhalt und sind eher als Anregung zu verstehen. Als Verbindungsleitung habe ich dünne abgeschirmte Adern von alten Audiokabeln benutzt, die Buchsen angelötet und mit Schrumpfschlauch gesichert. Alle Steine sind nach Möglichkeit so bearbeitet worden dass die normale Steckfunktion noch gegeben ist. Zum Verbinden und Arretieren habe ich Sekundenkleber benutzt, nach eigener Erfahrung sollte dieser möglichst dickflüssig sein. Als „Zugentlastung“ eignet sich auch etwas Heißkleber an passender Stelle (z.B. in den Mittelrohren der Bausteine).

3.4.2 Lampen (LEDs)



Bei den LEDs wurden nach Verfügbarkeit 5mm bedrahtete LEDs und PLCC4-LEDs mit Linse benutzt. Die Steine sind wieder vom Format 4x4. Die Zuleitung geht (wenn möglich) von hinten durch das Mittelrohr und wird dort im Anschluss mit etwas Heißkleber fixiert. Bei den SMD-LED wurde nur ein Loch für die Linse in den Baustein gebohrt, bei den bedrahteten LED musste ein Teil des Mittelrohrs entfernt werden, damit die LED von innen durch das gebohrte Loch gesteckt werden kann. Die Abbildung zeigt eine rote PLCC-LED, eingebaut in einen roten Baustein. Bei LED mit transparentem Gehäuse ist es sinnvoll, dass LED-Farbe und Bausteinfarbe korrespondieren, bei eingefärbten LED-Gehäusen können auch einfarbige Bausteine (schwarz, grau, weiß) zum Einsatz kommen.

3.4.3 Taster



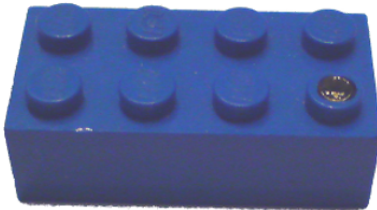
Für die verwendeten Taster wurden wieder zwei 4x4 Steine aufeinandergeklebt und danach das Loch für den Taster gebohrt.

3.4.4 Fotowiderstand



Dazu habe ich 2 4x4 Steine zusammengeklebt und danach mit einem Rollenfräser das Loch für den Fotowiderstand eingebracht. Zum Schluss noch ein Loch in der Rückseite für das Kabel. Nach dem Anlöten und Testen wurde der Fotowiderstand mit Sekundenkleber im „Gehäuse“ fixiert. Um die Empfindlichkeit zu erhöhen, kann noch ein Widerstand parallel geschaltet werden, da erst Widerstandswerte $< 4,7\text{K}\Omega$ überhaupt vom Controller erkannt werden.

3.4.5 Reedkontakte



Die Reedkontakt liegen mittig in 2x4 Steinen. Abhängig von den verwendeten Magneten kann es auch sinnvoll sein, die Röhrchen möglichst an der Seite der Steine zu platzieren. Damit sich die Schalter und Magnete später leichter justieren lassen, sind noch jeweils eine LED + $10\text{K}\Omega$ Vorwiderstand parallel zum Reedkontakt geschaltet. Bei offenen Kontakt leuchtet die LED, durch die Dimensionierung wird der Schalter aber trotzdem als offen erkannt.

Als Magnete wurden kleine quaderförmige, relativ starke Magnete seitlich in Bausteine geklebt. Wenn man die Ausrichtung so wählt dass diese sich anziehen kann man sie auch noch für andere Experimente verwenden.

3.4.6 Servos



Als Servos / Getriebemotoren wurden Modelcraft-Servos von Conrad verwendet. Um sie als Getriebemotoren verwenden zu können, müssen sie umgebaut werden:

1. Entfernen des Anschlages, der die Bewegung begrenzt
2. Entfernen des Rückführ-Potentiometers und Ersatz durch einen Spannungsteiler aus zwei Widerständen
3. Vergrößerung des „Null-Bereiches“, meist durch einen zusätzlichen Kondensator

Die Vorgehensweise ist je nach Modell etwas anders, im Internet finden sich detaillierte Anleitungen für verschiedene Typen. Bei Bedarf

kann ich auch den Umbau des von mir eingesetzten Typs dokumentieren.

Kapitel 4

Changelog

29.10.2013 Version 3.16

- erste öffentliche Version

04.11.2013 Dokumentation Revision A

- Kondensatoren (6x100nF) in den Schaltplänen ergänzt
- Erklärung der Schaltfläche „Controller einrichten“