

**Technische Universität
München**

**Fakultät für Informatik
Forschungs- und Lehrereinheit Informatik VI**

**Prozessrechner-Praktikum
Echtzeitsysteme**

Aufgabe 6 - Kugelbahn

Matthias Regensburger
regensbu@in.tum.de

Christian Buckl
buckl@in.tum.de

Sommersemester 2008

1 Abläufe auf der Kugelbahn

In der folgenden kurzen Einführung werden die Abläufe auf der Kugelbahn anhand des Weges, den eine Kugel auf der Bahn zurücklegt, vorgestellt. In Abbildung 1 ist die Kugelbahn mit ihren Komponenten zu sehen.

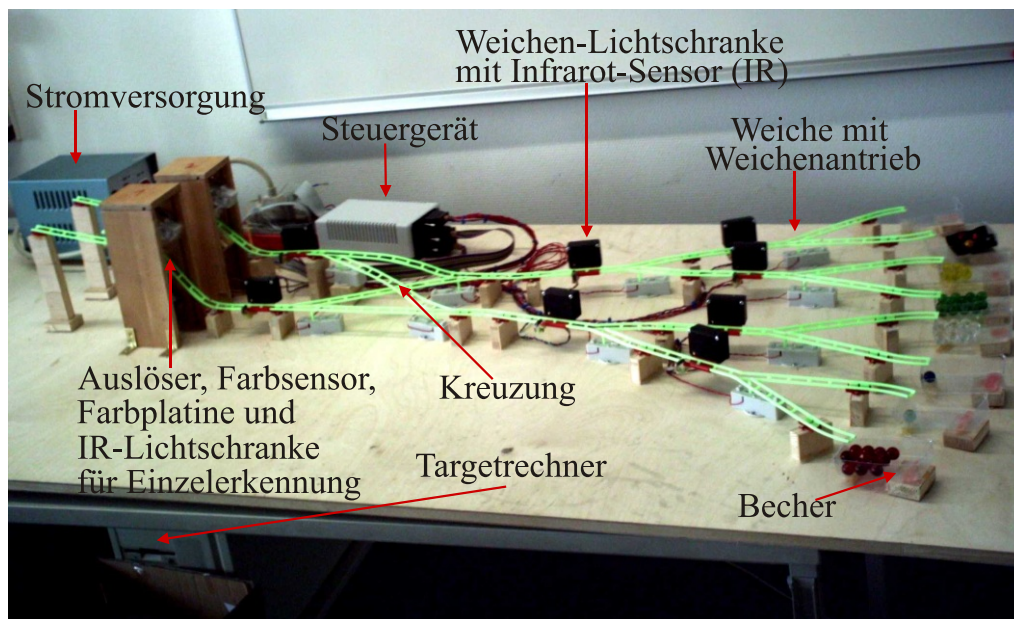


Abbildung 1: Aufbau des Kugelbahn-Versuchs

Die Wegstrecke einer Kugel zum Zielbecher beginnt vor einem der Auslöser. Beide Auslöser befinden sich unter den Holzgehäusen. Direkt vor den Auslösern sind seitlich jeweils Weisslicht-LEDs und ihnen gegenüber Farbsensoren angebracht. Die Farbsensoren liefern drei Farbwerte (Rot, Grün und Blau), aus denen sich jede beliebige Farbe zusammensetzen lässt.

Vor den Weichen befinden sich IR-Lichtschanken. Eine Kugel, die durch die Lichtschanke läuft, unterbricht den IR-Lichtstrahl und sendet ein Signal an das Steuergerät. Vom Steuergerät können Signale dann durch den Targetrechner abgeholt werden. Über das Steuergerät werden auch die Weichenantriebe gesteuert.

Erst wenn eine Kugelfarbe erkannt worden ist, kann entschieden werden, ob die betreffende Kugel ausgelöst werden soll (bzw. ob eine Weiche beim Eintreffen der Kugel geschaltet werden soll). Die Farberkennungssensoren befinden sich deshalb vor den Auslösern gleich am Anfang der Kugel-Wegstrecke.

Nach Passieren eines Auslösers gelangt eine Kugel stets zur ersten Weichen-Lichtschanke. Durch die IR-Lichtschanke wird ein Signal erzeugt. Solange das Signal anliegt, leuchtet die grüne LED im schwarzen Lichtschankegehäuse auf (ca. 30 ms). Wurde das Signal

vom Targetrechner verarbeitet, kann bei Bedarf über das Steuergerät ein Schaltvorgang der betreffenden Weiche veranlasst werden. Die Kugel rollt über die eventuell bewegte Weiche bis zur nächsten IR-Lichtschanke. Auf diese Weise setzt eine Kugel ihren Weg bis zum Zielbecher fort. Beide Auslöser können übrigens unabhängig voneinander bewegt werden. Alle Weichen lassen sich ebenfalls unabhängig voneinander schalten.

Die Kugelbahn kann in zwei Modi betrieben werden:

1. Im Magazinmodus liegen mehrere Kugeln hinter jedem Auslöser. Die Kugeln können stückweise freigegeben werden. Die Farbe kann im Ruhezustand einer Kugel ermittelt werden.
2. Im Einzelmodus werden die Kugeln einzeln auf die Bahn gelegt und rollen dann sofort los. Die Kugelfarbe muss während des Vorbeirollens der Kugel am Farbsensor ermittelt werden.

2 Aufgabenziel

Schreiben Sie ein Programm, mit dem mehrere Kugeln, die vor jedem der Auslöser liegen, stückweise ausgelöst werden können. Jede Kugel soll entsprechend ihres Farbwertes in einen der Becher rollen. Achten Sie darauf, dass zwei Kugeln in Kreuzungsnähe nicht kollidieren. Kugeln, die sich auf dem Weg zu Zielbecher befinden, dürfen sich nicht zu nahe kommen, damit eine separate Erkennung durch die Weichen-Lichtschraken gewährleistet ist.

Es soll außerdem möglich sein, Kugelfarben während des Vorbeilaufens am Farbsensor zu erkennen. Dabei können die Auslöser stets geöffnet sein. Einzelne Kugeln rollen nach der Farberkennung sofort zum entsprechenden Becher.

3 Ansteuerung der Kugelbahn

Der Targetrechner *atknoll133* steht unter dem Tisch, auf dem sich die Kugelbahn befindet. Im Target befindet sich eine digitale I/O-Karte ME1400B. Über die I/O-Karte kann das Steuergerät der Kugelbahn angesprochen werden. Einerseits lassen sich über das Steuergerät Weichen stellen, Farbmessungen starten und Kugeln auslösen. Andererseits können gewandelte, digitale Farbwerte und aufbereitete Signale der Weichen-Lichtschraken von den Ausgängen des Steuergeräts eingelesen werden. Die Anordnung ist in Abbildung 2 zu sehen.

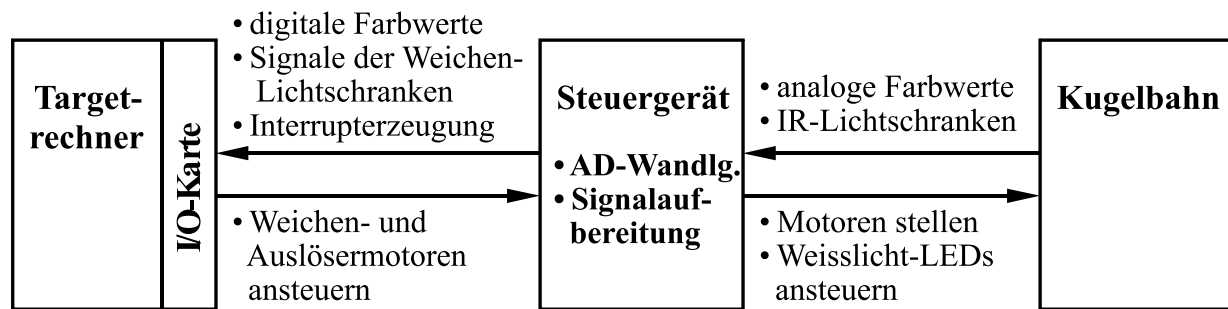


Abbildung 2: Signalfluss zwischen Targetrechner, Steuergerät und Kugelbahn

Zum Lesen und Schreiben der sechs 8-bit Ports der I/O-Karte existieren im Modul *ME1400.c* zwei Routinen. Der Routine *portLesen()* wird als erster Parameter eines der sechs Ports übergeben. Sie sollten jedoch niemals auf die Ports *PORT_A_x* zugreifen. Die A-Ports dienen zum Ansteuern der Weichen- und Auslösermotoren, was immer durch die bereits vorhandenen Routinen (vgl. Bewegen der Weichen und der Auslöser) bewerkstelligt werden sollte. Auf die B-Ports hingegen muss Ihr Programm lesend bzw. schreibend zugreifen. Mit dem zweiten Parameter muss beim lesenden Zugriff ein Byte per Referenz übergeben werden. In das Byte wird der eingelesene Wert des Ports geschrieben. Beim schreibenden Zugriff wird das ins Port zu schreibende Byte übergeben. Rückgabewert der Routinen ist jeweils *OK*, wenn die Routine ordnungsgemäß aufgerufen worden ist, oder *ERROR* im Fehlerfall.

3.1 Kugelfarbe messen

Eine Farbmessung im Magazinmodus wird durch Setzen von **Bit 0** an *PORT_B_C* der I/O-Karte veranlasst. Der erste gewandelte Farbwert ist immer der Blauanteil von Farbsensor 1 (vgl. Abbildung 3). Eine Numerierungen aller Komponenten der Kugelbahn, die auch für Ihr Programm verwendet werden sollte, finden Sie in Abbildung 5 am Ende des Manuals. Nachdem ein gewandelter Farbwert an *PORT_B_A* anliegt, wird ein Interrupt im Targetrechner ausgelöst und die Interruptroutine *user_isr()* im Modul *Rahmen.c* aufgerufen. Mit der Interruptroutine kann der Farbwert von *PORT_B_A* eingelesen werden. Wurden alle 3 Farbwerte zu einer Kugel abgeholt, dann kann **Bit 0** an *PORT_B_C* wieder zurückgesetzt werden, was die AD-Wandlung im Steuergerät stoppt. Die digitalen Farbwerte am Steuergerät liegen in folgender Sequenz vor:

1. Blau-/Grün- und Rotwert der Kugel vor Farbsensor 1, Jeder der Werte löst einen Interrupt im Targetrechner aus. Jeder der Werte liegt in negierter Form an (AD-Wandler kann nur negierte Werte abgeben).
2. Blau-/Grün- und Rotwert der Kugel vor Farbsensor 2, Jeder der Werte löst einen Interrupt im Targetrechner aus. Jeder der Werte liegt in negierter Form an (AD-Wandler kann nur negierte Werte abgeben).

- Es wird ein minimaler und ein maximaler Wandlungswert (AD-Wandler-Wert) vom Steuergerät geliefert. Diese liegen in negierter Form an und werden durch einen Interrupt signalisiert. Sie dienen dazu ein Byte voll zu bekommen.

Insgesamt besteht ein Durchlauf im Steuergerät also aus acht Messungen. Jede der Messungen liegt negiert an der I/O-Karte des Targets an. Jede abgeschlossene Messung wird im Targetrechner durch den selben Interrupt angezeigt. Es obliegt Ihrem Programm die Farbwerte zur richtigen Kugel auszuwerten. Eine Farbmessung zu einer Kugel muss nicht zwangsläufig alle acht Einzelmessungen durchlaufen. Der AD-Wandler im Steuergerät kann beispielsweise nach der dritten Messung abgeschaltet werden, wenn nur die Kugel vor Auslöser 1 gemessen wird. Wird der AD-Wandler nach der achten Messung nicht abgeschaltet, dann beginnt die Sequenz wieder von vorn, d.h. mit dem ersten Farbwert (Blauanteil) für die Kugel vor Farbsensor 1. In Abbildung 3 wird der komplette Durchlauf aller acht Einzelmessungen nochmals verdeutlicht.

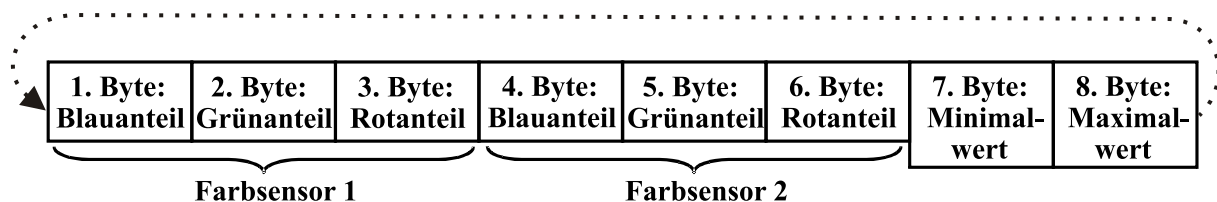


Abbildung 3: kompletter Durchlauf der acht vom Steuergerät erzeugten Einzelmessungen

WICHTIG: Ebenfalls durch den selben Interrupt, durch den gewandelte Farbwerte angezeigt werden, werden dem Targetrechner von den Weichen-Lichtschranken erkannte Kugeln mitgeteilt. Ihr Programm muss in solchen Fällen das *PORT_B_B* der I/O-Karte auslesen und entsprechende Massnahmen zum Stellen der zugehörigen Weiche einleiten. Signale von Weichen-Lichtschranken können zeitgleich zu gewandelten Farbwerten vorliegen! Um Weichen-Lichtschranken-Signale zu erzeugen, muss das **Bit 0** an *PORT_B_C* nicht unbedingt gesetzt sein! Wenn **Bit 0** jedoch gesetzt ist, ist bei jedem Aufruf der Routine *user_isr()* ein neuer Farbwert, gemäss der Reihenfolge aus Abbildung 3, abrufbar!

Für die Farberkennung während des Vorbeilaufs der Kugeln an den Farbsensoren sind spezielle Einstellungen zu berücksichtigen. An *PORT_B_C* ist **Bit 1** und **5** bzw. **Bit 2** und **4** in Abhängigkeit von Auslöser 1 bzw. 2 zu setzen oder zu löschen. Mit gesetztem **Bit 1** bzw. **2** wird die Einzelerkennung am entsprechenden Auslöser freigeschaltet. Nach Freischalten meldet eine IR-Lichtschranke unter den Auslösergehäusen eine vorbeirollende Kugel an das Steuergerät. Im Steuergerät wird daraufhin der AD-Wandler angeschaltet und wandelt nun ununterbrochen analoge Farbwerte für beide(!) Farbsensoren. Sie können nun den Datenstrom von Bytes über die I/O-Karte abholen (wie oben beschrieben). Wurde ein gültiger Farbwert für eine Kugel erkannt, dann kann

die frei geschaltete Lichtschranke durch Rücksetzen (LOW-aktiv) des **Bits 5** bzw. **4** wieder abgeschaltet werden. Der AD-Wandler im Steuergerät stoppt übrigens erst dann, wenn beide IR-Lichtschranken unter den Auslösergehäusen abgeschaltet sind. Es obliegt wiederum Ihnen, die richtigen Daten aus der Sequenz von acht Bytes, die das Steuergerät liefert, herauszuholen. Weitere Aktionen, die eine Kugel auf dem Weg zum Becher auslösen soll, entsprechen genau denen, die im Magazinmodus nötig sind. Eine Ausnahme besteht dennoch. Im Einzelmodus ist es möglich je eine Kugel vor jedem Auslöser zum selben Zeitpunkt freizugeben. D.h., die Kugeln könnten an Weiche 2 oder 3 kollidieren, was sich leider nicht verhindern lässt, wenn die Auslöser ständig geöffnet sind.

Zu beachten ist, dass Glaskugeln Licht, das durch sie hindurch tritt, brechen. Wegen der Lichtbrechung wird bei Vorbeigehen einer Kugel am Farbsensor eine bestimmte Folge von Hell- und Dunkelphasen erzeugt. Zuerst wird das Licht nahezu vollständig vom Farbsensor weg gelenkt. Anschließend trifft ein fokussierter Lichtstrahl auf den Sensor und schließlich wird das Licht wieder nahezu weg gelenkt. In Abbildung 4 ist dieser Effekt verdeutlicht.

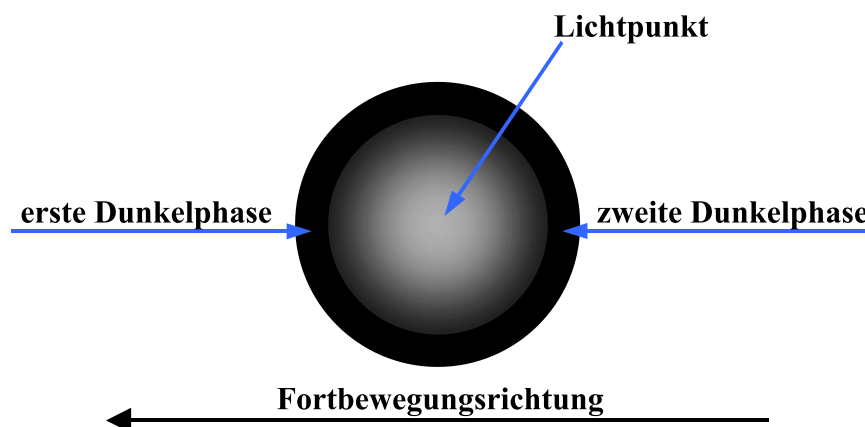


Abbildung 4: Dunkel- und Hellphasen einer durchleuchteten Glaskugel

Die Dunkel- und Hellphasen der Glaskugeln können ausgenutzt werden, um im Einzelmodus "brauchbare" Messungen oder eine vorbeigerollte Kugel zu erkennen.

3.2 Bewegen der Weichen und der Auslöser

Weichen und Auslöser lassen sich mit Hilfe der Routinen in *Rahmen.c* steuern. Verwenden Sie in jedem Fall nur diese Routinen zum Stellen der Weichen und Auslöser! Der Aufbau könnte durch Beeinflussung der Ports *PORT_A_A*, *PORT_A_B* oder *PORT_A_C* zerstört werden. Durch die gegebenen Routinen ist kein explizites Setzen der A-Ports in Ihrem

Programm nötig, um Weichen und Auslöser zu stellen. Setzen Sie weder Bits der Ports *PORT_A_A*, *PORT_A_B* und *PORT_A_C* noch ändern Sie die Werte der gegebenen define-Anweisungen im Modul *Rahmen.c* !

Der Routine *WeicheStellen()* wird als erster Parameter eine Weichennummer (0 bis 9) und als zweiter Parameter eine Richtung (*LINKS* bzw. *RECHTS*) übergeben. Das Stellen einer Weiche dauert ca. 17 ms. Vor dem erstmaligen Stellen der Weichen sind diese in einen definierten Zustand zu versetzen. Dazu muss die Routine *WeicheInitialisieren()* für jede Weiche einmalig aufgerufen werden.

Zum Steuern der Auslöser dienen die beiden Routinen *KugelAusloesen()* und *KorrekturAusloeser()*. Die Routine *KugelAusloesen()* veranlasst, dass der Auslöser nach oben bewegt wird, eine gewisse Zeit im geöffneten Zustand verweilt und dann wieder schließt. Der gesamte Vorgang dauert ca. 100 ms. Wurde keine gültige Kugelfarbe - gültige Farben sind in den Feldern *farben1[]* und *farben2[]* abgelegt - erkannt, dann kann die Kugel durch Bewegen des betreffenden Auslösers etwas umpositioniert werden. Eine erneute Farbmessung liefert dann meist gültige Werte. Eine Positionskorrektur wird mit Hilfe der Routine *KorrekturAusloeser()* durchgeführt. Erster Parameter ist der Auslöser (1 oder 2). Zweiter Parameter beinhaltet die Zeit in Millisekunden, für die der Auslöser nach oben bewegt werden soll. Für die Zeit ist ein Wert von 0 bis 5 ms zulässig; üblicherweise sollten 2 oder 3 ms für eine leichte Bewegung der Kugel ausreichen. Kugelfarben werden übrigens mit sehr hoher Wahrscheinlichkeit bei der ersten Messung erkannt. Ihr Programm kann somit erst im fortgeschrittenen Stadium mit einer zusätzlichen Korrekturfunktion ausgestattet werden. Wurde nach der Farbmessung eine gültige Kugelfarbe erkannt, dann lässt sich die Kugel durch die Routine *KugelAusloesen()* zum Becher schicken. Der erste Parameter beinhaltet wieder den Auslöser (1 oder 2). Der zweite Parameter steht für die Zeit, die bisher insgesamt für Positionskorrekturen der Kugel aufgewendet wurde. Der zweite Parameter muss deshalb übergeben werden, weil der Auslöser eventuell schon nach oben bewegt wurde und genau dieser Weg beim Auslösen berücksichtigt werden muss. Wurde keine Positionskorrektur durchgeführt, was der Regelfall sein dürfte, dann wird 0 übergeben.

4 Auswerten der eingelesenen Farbwerte

Für jede Kugelfarbe und jeden Auslöser wurden auf der Boot-Diskette des Targetrechners bereits gültige Farbwerte hinterlegt. Genauer gesagt handelt es sich um Farbbereiche für jede Kugelfarbe. So umfassen die Farbbereiche für eine gelbe Kugel einem bestimmten Bereich für Rot, z.B. 25 bis 150, und einem bestimmten Bereich für Grün, z.B. 10 bis 210, sowie einen Bereich für Blau, z.B. 0 bis 10. Die von der I/O-Karte eingelesenen Messwerte für Rot, Grün und Blau einer gelben Kugel sollten innerhalb dieser Bereiche liegen, damit die Kugel eindeutig als "Gelb" identifiziert werden kann. Auf diese Weise sind auch die

	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
PORT_B_A	LSB gewandelter Rot-/Grün- oder Blau-Wert (negiert)							MSB
PORT_B_B	Lichts. Weiche 4	Lichts. Weiche 5	Lichts. Weiche 6	Lichts. Weiche 7	Lichts. Weiche 8	Lichts. Weiche 9	Lichts. Weiche 1	Lichts. Weiche 0
PORT_B_C	AD- Wandler starten (HIGH- aktiv)	Einzel- erkenng. 1 frei (HIGH- aktiv)	Einzel- erkenng. 2 frei (HIGH- aktiv)	nicht benutzt	Reset Einzel- erkenng. 2 (LOW- aktiv)	Reset Einzel- erkenng. 1 (LOW- aktiv)	nicht benutzt	nicht benutzt

Tabelle 1: Übersicht über die Belegung des B-Ports der I/O-Karte

anderen Kugelfarben in den Farbdateien auf der Boot-Diskette hinterlegt.

Die Farbdateien können mit der Routine *FarbenEinlesen()*, die sich im Modul *Rahmen.c* befindet, vollständig eingelesen werden. Nach dem Einlesen liegen Kopien der Farbbereiche für jede Kugelfarbe in den beiden Tabellen *farben1[]* und *farben2[]*, die wie folgt aufgebaut sind.

Zeilen:

- Zeile 1: Farbbereiche für rote Kugeln,
- Zeile 2: Farbbereiche für hellblaue Kugeln,
- Zeile 3: Farbbereiche für dunkelblaue Kugeln,
- Zeile 4: Farbbereiche für klare (farblose) Kugeln,
- Zeile 5: Farbbereiche für grüne Kugeln,
- Zeile 6: Farbbereiche für gelbe Kugeln,
- Zeile 7: Farbbereiche für violette Kugeln,
- Zeile 8: Farbbereiche für nicht-erkannte Kugeln,

Spalten:

- Spalte 1: minimaler Wert für Rotanteil,
- Spalte 2: maximaler Wert für Rotanteil,
- Spalte 3: minimaler Wert für Grünanteil,
- Spalte 4: maximaler Wert für Grünanteil,
- Spalte 5: minimaler Wert für Blauanteil,
- Spalte 6: maximaler Wert für Blauanteil,

Die Werte für die Farbbereiche liegen nicht negiert in den Feldern *farben1[]* und *farben2[]* vor. Kleinster Wert in den Feldern ist 0 und größter Wert 255. Es handelt sich um Integer-Felder. Die von der I/O-Karte gelieferten Farbwerte sind vorzeichenlose Bytes. Mit Integeren lassen sich gewisse Vergleiche jedoch unkomplizierter durchführen.

Zeile 8 ist unbenutzt, weil momentan nur sieben unterschiedliche Kugelfarben für die Kugelbahn existieren. Deshalb wurden in die Spalten der Zeile minimale Werte von 255 und maximale Werte von 0 eingetragen.

Metallkugeln sind lichtundurchlässig und verursachen für alle drei Grundfarben Messwerte von 0xFF (negiert). Die Messwerte passen genau in die Farbbereiche für violette Kugeln. Metallkugeln können deshalb dem selben Becher wie violette Kugeln zugeordnet werden. Für beide Modi, den Magazinmodus und den Einzelmodus, gibt es getrennte Farbdateien. Die Dateien für den Magazinmodus liegen im Verzeichnis *Magazinmodus* auf der Boot-Diskette. Die Dateien für den Einzelmodus liegen im Verzeichnis *Einzelmodus*.

5 Beschreibung der Routine *main()* im Modul *Rahmen.c*

Nach Initialisierung der drei Bytes für Port A (nicht der Karten-Ports direkt), wird die I/O-Karte in das Betriebssystem eingebunden und initialisiert. Erst nach der Initialisierung der I/O-Karte sind die Ports als Eingabe- bzw. Ausgabeports beschaltet und in einen definierten Zustand versetzt. Der Interrupteingang der I/O-Karte ist noch nicht aktiviert. Die Motoren der Weichen- und Auslöser werden direkt vom Targetrechner aus gesteuert. Das Steuergerät wandelt lediglich Digitalwerte von der I/O-Karte für die Motoren um. Die Zeitdauer der Steuerimpulse wird ebenfalls vom Targetrechner aus beeinflusst. Für genanntes sind unter anderem die eingerückt gekennzeichneten Codestellen im Modul *Rahmen.c* zuständig. Um die Motorsteuerung auf Millisekunden genau durchführen zu können, ist die Taktung des Betriebssystems auf 1 ms gesetzt. Dieser Wert darf auf keinem Fall verändert werden. Er wirkt sich direkt auf die Motorensteuerzeiten aus. Motoren die zu lange angesteuert werden, könnten durchbrennen oder Teile der Weichen bzw. der Bahn zerstören. Mit den nächsten drei Zeilen wird eine durch vorbeigelaufene Kugeln gestartete Farbmessung im Einzelmodus zurückgesetzt. Da nach dem Einschalten des Steuergerätes dessen

Zustände nicht garantiert sind, könnte es durchaus sein, dass die Kugelerkennung für den Einzelmodus aktiv ist. Infolgedessen kann eine an einem IR-Sensor unter den Holzgehäusen vorbeilaufende Kugel die AD-Wandlung im Steuergerät starten. Wenn nun im Steuergerät die Kugelerkennung für den Einzelmodus deaktiviert wurde, ist ein versehentlicher Start des AD-Wandlers im Steuergerät auszuschließen.

Schließlich wird der Interrupteingang der I/O-Karte frei geschaltet. Im Anschluß können Codezeilen für Ihr Programm eingefügt werden. Damit sich das Target beim nächsten Programmstart in einem definierten Zustand befindet, sollten vorgenommene Änderungen für die Target-Hardware beim Beenden Ihres Programms wieder rückgängig gemacht werden. Dies wird mit den restlichen Zeilen in *main()* erledigt.

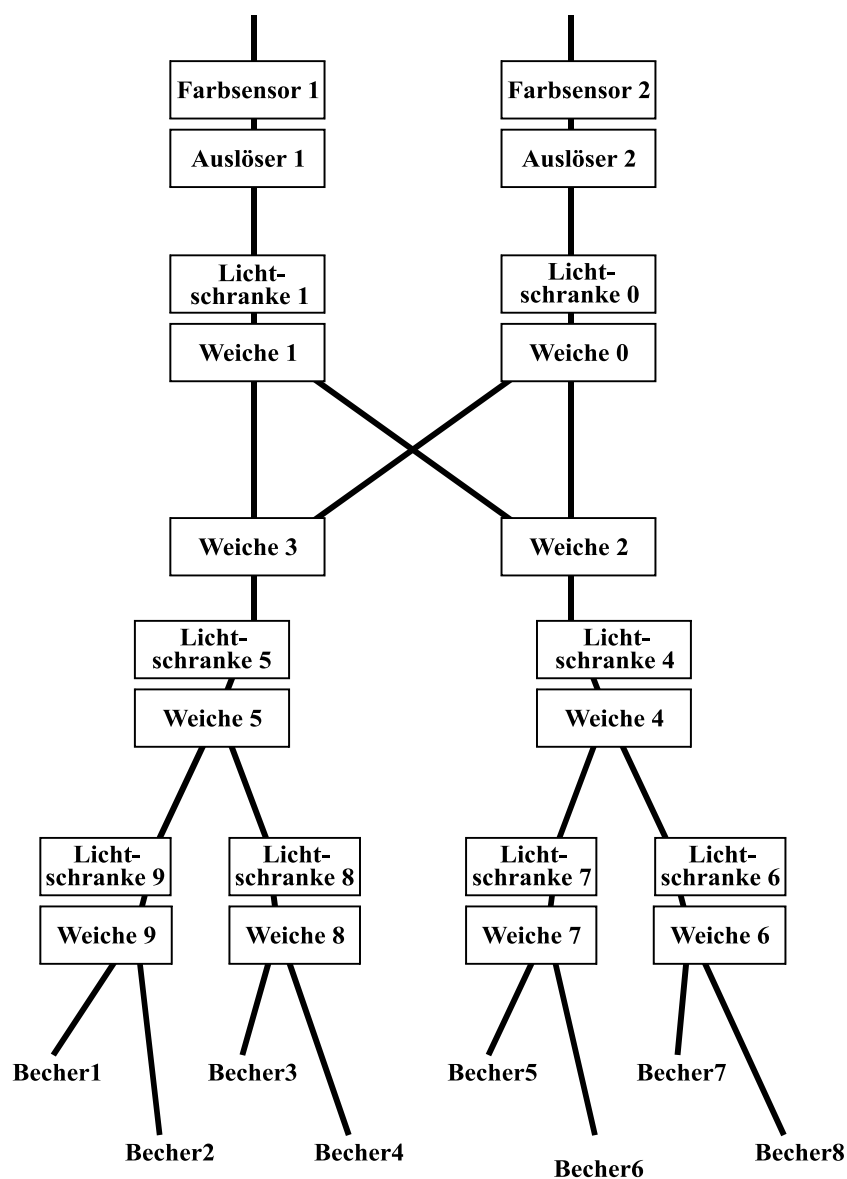


Abbildung 5: Komponenten der Kugelbahn