

# 1. Sitzung: Kennenlernen von LEGO Mindstorms

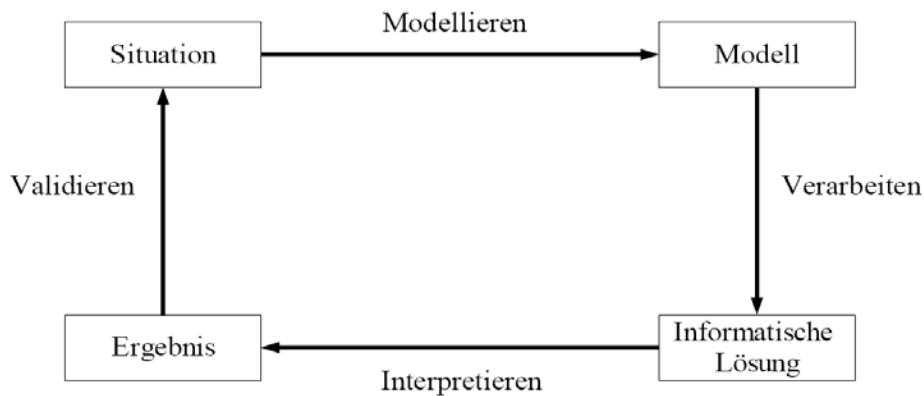
## 1. Erklärung Ablauf

- Projektseminar:  
hauptsächlich Eigenarbeit, zusätzlich gemeinsame Phasen zur Einführung bzw. Zusammenfassung  
Betrachtung der Einheiten sowohl aus Lehrer- als auch aus Schülersicht
- Verschiedene Phasen:  
Einführung, Eigenarbeit, didaktische Reflektion (Lehrplanbezug, Lernziele, Unterrichtsmethodik, ...), Zusammenfassung
- Erstellen eines Seminarberichts:  
Jeweils ein Teilnehmer fasst eine Sitzung bzw. Sitzungsgruppe zusammen. Dabei sollen alle wichtigen Ergebnisse festgehalten werden. Die entstandene Sammlung soll als Vorlage für den eigenen Unterricht dienen können.

## 2. Einführung

Ein moderner Informatikunterricht versucht, fachliche Inhalte durch praxisbezogene Aufbereitung zu vermitteln. Der Lernprozess sollte so individualisiert werden, dass der Schüler im Mittelpunkt steht. Er soll durch Eigenaktivität die Möglichkeit haben, sich mit einem Thema auseinander zu setzen. Die Lehrkraft dient als Ansprechpartner, gibt Anstöße, hilft bei der Reflektion – ist aber kein Dozent.

Gerade die Informatik bietet sich für einen solchen Unterricht an. Ihr praktischer Bezug ist den Schülern aus dem Alltagsleben bekannt. Ein Ziel des Unterrichts ist das Kennenlernen und Verstehen der Funktionsweise von Informatiksystemen. Dieses Ziel ist am besten erreichbar, wenn die Lernenden selbst ausführliche Erfahrungen mit Informatiksystemen sammeln können. Dabei soll es jedoch nicht um den Erwerb bloßer Bedienkompetenz gehen. Vielmehr muss erreicht werden, Systeme von einer höheren Warte zu sehen, bereits erlangtes Wissen auf neue Kontexte anwenden zu können und den informatischen Problemlöseprozess verstehen und anwenden zu lernen. Gerade dieser Problemlöseprozess eignet sich für einen eigenständigen, aktiv entdeckenden Unterricht.



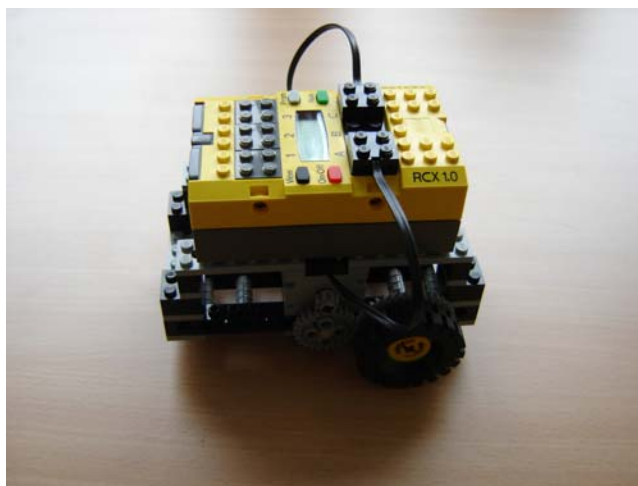
Der Einsatz von Medien besitzt im Informatikunterricht einen besonderen Stellenwert. Ein zentrales Medium stellt das Informatiksystem selbst dar. Der Informatikunterricht sollte dieses Medium selbst nie aus den Augen verfehlen. Besonders in der Sekundarstufe I stellen Informatiksysteme den zentralen Kern des Unterrichts dar.

Diese Veranstaltung befasst sich hauptsächlich mit dem Roboter-System LEGO Mindstorms und mit der Entwicklungsumgebung Squeak. Beide Systeme eignen sich für einen praxisbezogenen, problemorientierten Unterricht. Im Umgang mit diesen Informatiksystemen lernen die Schüler diese zu bedienen und zu verstehen, erwerben darüber hinaus jedoch auch grundsätzliche Fähigkeiten des informatischen Problemlöseprozesses, lernen Algorithmen kennen und diese problembezogen zu entwickeln. Zudem bieten beide Themen einen Einstieg in die Programmierung.

### 3. Kennenlernen von LEGO Mindstorms

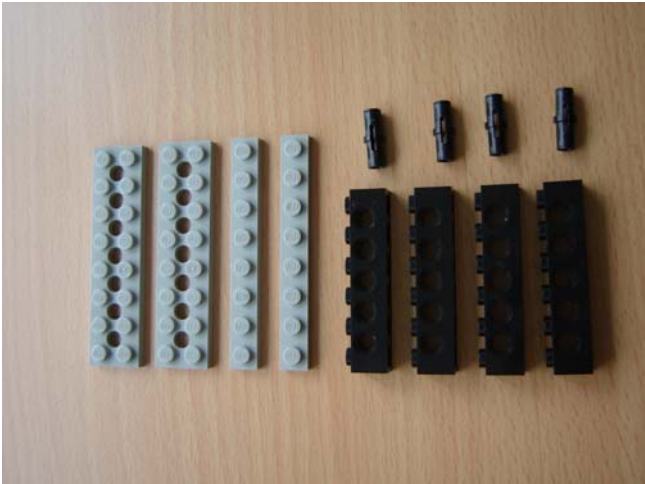
Die erste Einheit dient dem Kennenlernen von Mindstorms. Ziel ist es einen ersten Roboter zu bauen und die Funktionsweise des RCX Bausteins zu erfassen. Dazu wird ein Universalroboter gewählt, weil dieser flexibel mit Sensormodulen ausgestattet werden kann.

#### 3.1. Der Universalroboter

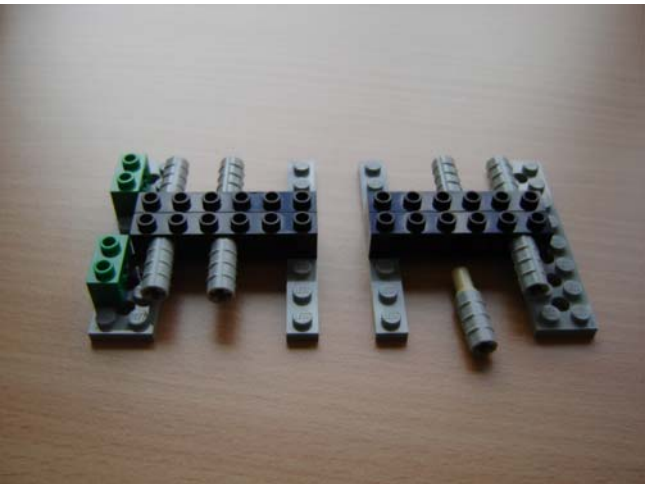


### Konstruktionsschritte

#### Schritt 1

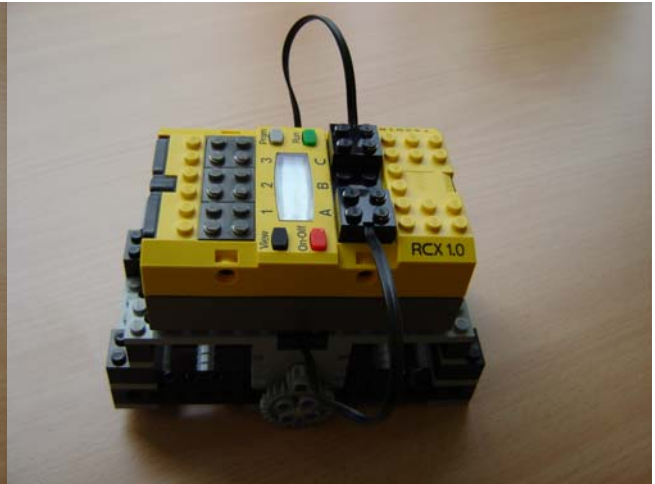
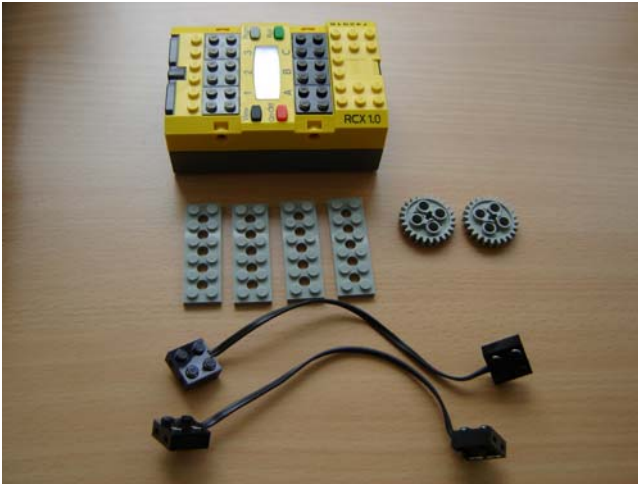


#### Schritt 2

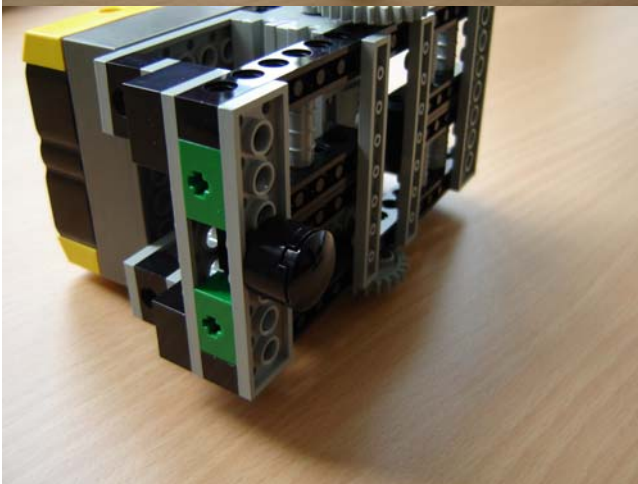




**Schritt 6**



**Schritt 7**



## 3.2. Erste Experimente mit Mindstorms

Nach Fertigstellung des Universalroboters kommen die im RCX ursprünglich vorhandenen Programme zum Einsatz. Dabei wird die grundsätzliche Bedienung des Bausteins eingeübt. Sollten noch keine Programme auf dem RCX vorhanden sein, muss zunächst mit Hilfe von Robolab die Firmware eingespielt werden (siehe auch 3.3. Vorbereitungen).

Die Experimente orientieren sich an den Kopiervorlagen aus dem Mindstorms Lehrerhandbuch. Dort wird der Acrobot verwendet. Die Experimente lassen sich aber ohne Probleme auf den Universalroboter übertragen.

### ***Experiment 1 – Starten von Programmen***

#### **Inhalte/Ziele**

Ziel dieses Experiments ist es den grundsätzlichen Umgang mit dem RCX bekannt zu machen. Durch den *On/Off* Schalter wird der Baustein aktiviert. Drücken des *Prgm* Knopfes wechselt zwischen Programmen. Betätigen der *Run* Taste startet das Programm. Erneutes Drücken der Taste stoppt die Programmausführung. Umpolen des Motors verändert die Laufrichtung.

#### **Arbeitsaufträge**

- Drückt die rote Taste. Was passiert?
- Stellt den Roboter auf den Boden. Drückt die *Prgm* Taste bis auf der Anzeige eine 1 steht. Drückt dann die *Run* Taste und beobachte was geschieht? Beschreibt was ihr seht.
- Drückt erneut die *Run* Taste.
- Dreht die Anschlüsse beider Motoren um 180° und startet den Roboter erneut. Was stellt ihr fest?
- Versucht den Roboter so zu verändern, dass er sich im Kreis dreht.

## **Experiment 2 – Steuern des Roboters**

### **Inhalte/Ziele**

Der RCX Baustein kann mehrere Programme parallel im Speicher halten. Mit der *Prog* Taste kann zum nächsten Programmplatz gewechselt werden. Zur Steuerung des Roboters können Sensoren verwendet werden. Diese werden an den Eingängen 1, 2 und 3 angeschlossen. Die Schüler sollen erkennen, welchen Einfluss das Betätigen der Sensoren auf den Programmablauf hat.

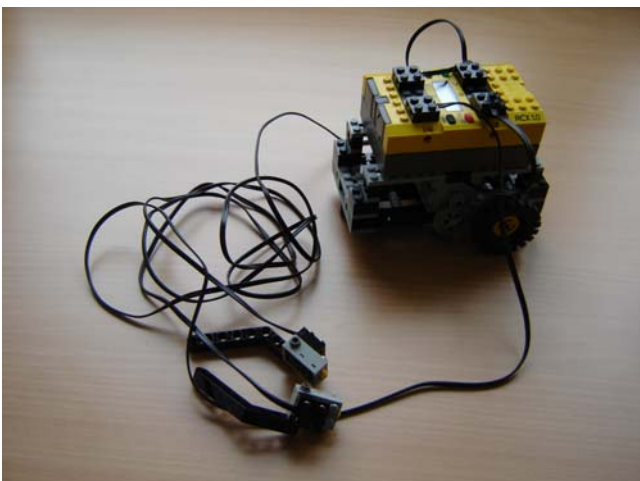
### **Arbeitsaufträge**

- Baut die beiden Griffe - wie in der Anleitung angegeben - zusammen.

### **Schritt 1**



### **Schritt 2**



- Schließt die beiden Griffe an den Anschlüssen 1 und 3 an.
- Stellt den Roboter auf den Boden, schaltet den RCX ein, wechselt zu Programm 2 und startet es. Was geschieht?
- Betätigt die gelben Tasten an den Griffen. Was stellt ihr fest? Wie verhalten sich die

Motoren.

- Lasst euren Roboter herumfahren und steuert ihn mit den Druckschaltern.

### Experiment 3 – Hinzufügen eines optischen Sensors

#### Inhalte/Ziele

Neben den Drucksensoren kann auch mit dem optischen Sensor Einfluss auf den Roboter genommen werden. Die Anschlüsse 1 - 3 dienen als Eingänge für Sensoren. Das Experiment wird mit der farbigen Testvorlage durchgeführt. Die Schüler erkennen, dass der optische Sensor zwischen hell und dunkel unterscheiden kann.

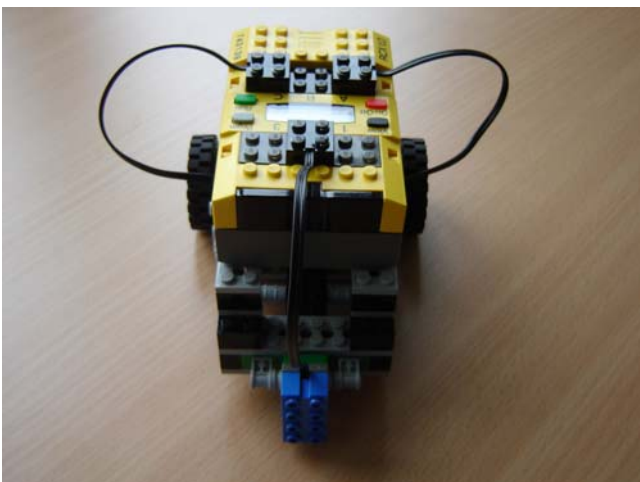
#### Arbeitsaufträge

- Baut den Lichtsensor - wie angegeben - an den Roboter.

#### Schritt 1



#### Schritt 2



- Stellt den Roboter auf die Testvorlage und startet Programm 3. Was stellt ihr fest?
- Wie reagiert der Roboter auf hell und dunkel?

## Experiment 4 – Auf Hindernisse reagieren

### Inhalte/Ziele

Dieses Experiment zeigt ein komplexeres Programm, bei dem der Roboter aufgrund von Daten des Berührungssensors seine Bewegung steuert. Die Schüler sollen den Programmablauf erkennen und mit eigenen Worten wiedergeben können.

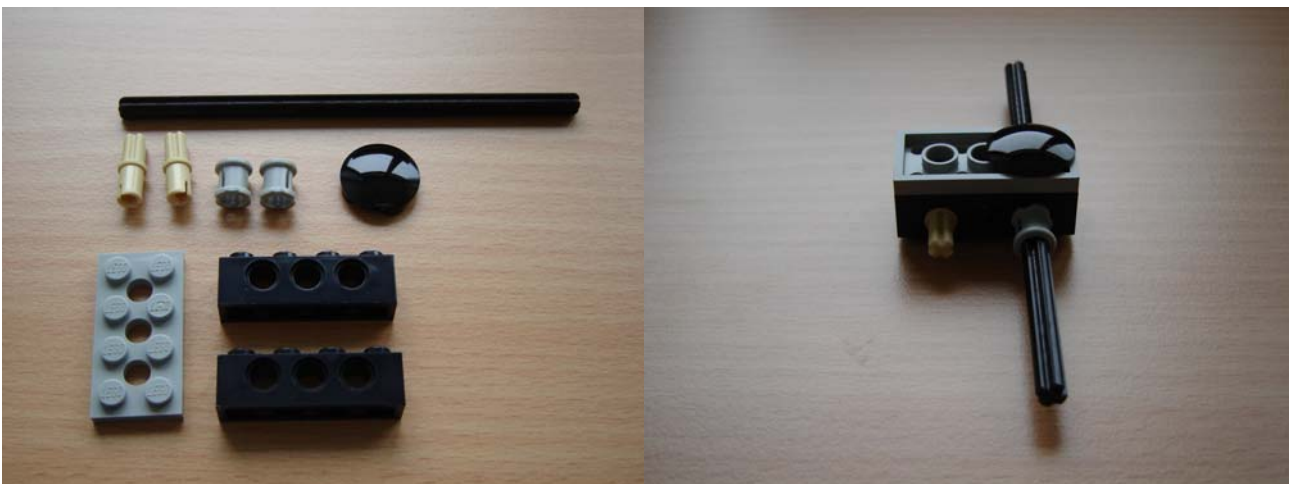
### Arbeitsaufträge

- Baut nach den angegebenen Schritten den Sensor zusammen und schließt ihn an Eingang 1 an.

### Schritt 1



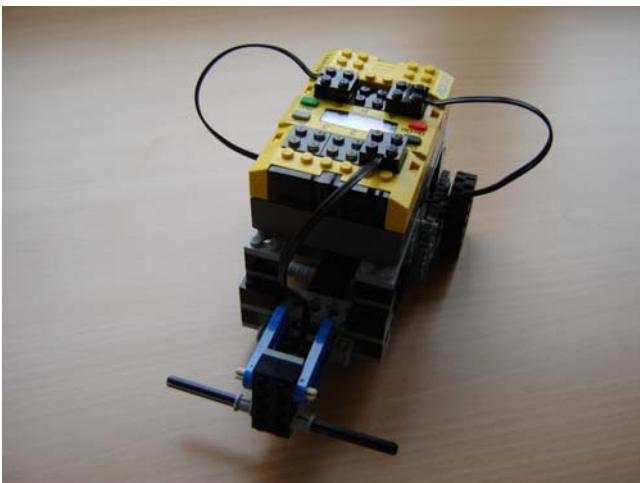
### Schritt 2



### Schritt 3



### Schritt 4



- Stellt den Roboter auf den Boden und führt Programm 5 aus. Was geschieht?
- Beschreibt, wie der Roboter auf Hindernisse reagiert.

### **Didaktische Hinweise**

Als Sozialform mit Mindstorms ist Gruppenarbeit grundsätzlich geeignet. Die Größe der Gruppen richtet sich dabei hauptsächlich nach der Anzahl der vorhandenen Baukästen. Drei Schüler können sehr gut zusammenarbeiten: je einer „verwaltet“ die Teile in einem Materialkasten, der dritte setzt die Teile zusammen. Die Rollen können gegebenenfalls auch bei längeren Konstruktionen getauscht werden.

Erfahrungsgemäß ist die Motivation der Schüler beim Arbeiten mit LEGO Mindstorms sehr groß. Bereits das Zusammenbauen des ersten Roboters ist für die Schüler ein Erlebnis. Dabei werden sowohl ihr „handwerkliches“ Geschick als auch das technische Verständnis gefördert. Es bietet sich an, vor Beginn der Bauphase einen fertigen Roboter zu zeigen. Damit wird ein eindeutiges Ziel für die ersten Arbeitsschritte festgelegt.

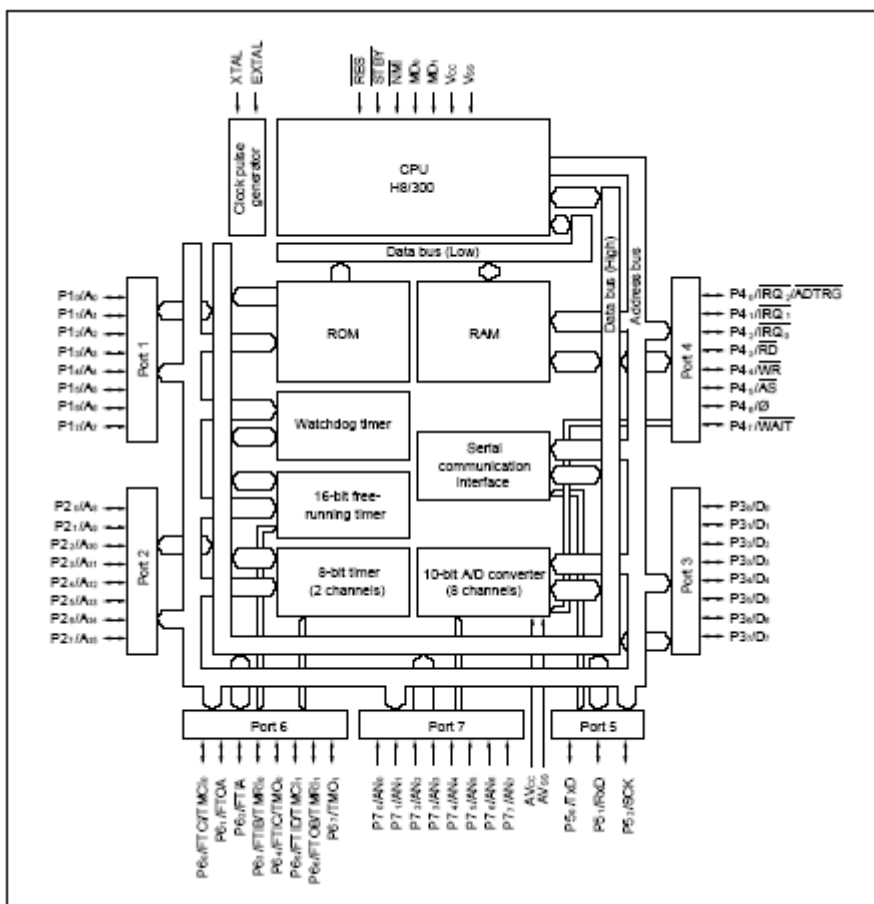
Für die Experimentierphase ist es sinnvoll, die Schüler mit entsprechenden Arbeitsblättern

auszustatten. Auf diesen sind die nötigen Arbeitsanweisungen – beispielsweise die Bauanleitung für einen neuen Sensor – und Aufgabenstellungen aufgeführt. Die Experimentierphase hat zum Ziel, dass die Schüler eigenständig bzw. in der Kleingruppe herausfinden, wie der RCX funktioniert. Alle Beobachtungen bei den Experimenten sollen notiert werden. Dies ist insbesondere beim vierten Experiment interessant, da hier schon ein etwas aufwändiger „Programmablauf“ beschrieben werden soll. Die Lehrkraft fasst die Ergebnisse der einzelnen Experimente zusammen, gibt weitere Hinweise und unterstützt die Schüler bei Problemen.

### 3.3. Technischer Hintergrund

#### 3.3.1. Der RCX Baustein

Der RCX Baustein benutzt einen Hitachi H8/3292 Microcontroller. Dieser besitzt eine H8/300 CPU und zusätzliche On-Chip Komponenten. Er hat 16 kByte ROM, 512 Bytes internes und 32 kByte externes RAM.



Blockdiagramm der Hitachi H8 Serie

Das ROM beinhaltet ein Basis-Betriebssystem für den RCX. Damit ist es insbesondere möglich Daten über die Infrarotschnittstelle zu empfangen und im RAM abzulegen. Darüber hinaus kann das Basis-Betriebssystem auf Sensoreingänge und

Steuerungsausgänge zugreifen. Für die volle Funktionsfähigkeit ist eine Ergänzung des fest vorhandenen Betriebssystems nötig. Dazu wird in einem Teil des RAM eine zusätzliche Firmware gespeichert. Dadurch wird es erst möglich, Programme über die Infrarotschnittstelle zu empfangen und auch auszuführen.

### **3.3.2. Entwicklung von Software für den RCX**

Die Firmware und das Betriebssystem im ROM bestehen aus Maschinencode für den Hitachi H8. Sie ermöglichen den Zugriff auf die Ein- und Ausgänge des RCX. Zudem erlauben sie das Einspielen, Speichern Starten und Stoppen von Programmen.

Die eingespielten Programme selbst bestehen nicht aus H8 Maschinenbefehlen. Viel mehr handelt es sich dabei um Bytecode, der vom Betriebssystem des H8 interpretiert wird (vgl. Java).

Das Erstellen von Programmen für den RCX ist ein typisches Beispiel für Cross Compiling für ein eingebettetes System (Embedded System). Das heißt, ein Programm wird auf einer Plattform – hier ein PC – erstellt und übersetzt und ist auf einem System mit anderer Architektur ausführbar.

Der herkömmliche Weg beim Erstellen von Software für ein Embedded System besteht aus der Benutzung eines Cross Compilers für die Zielplattform. Der Quellcode wird beispielsweise auf einem PC in C oder Assembler geschrieben und von einem Cross Compiler in Maschinencode für das Zielgerät übersetzt. Dieses Maschinenprogramm ist im Allgemeinen nicht auf der Entwicklungsplattform lauffähig. Der Maschinencode wird dann auf die Zielplattform transferiert (Chip, Datenübertragung ...) und kann dort direkt ausgeführt werden.

Der Cross Compiling Schritt für den RCX läuft insofern anders ab, dass nicht direkt Maschinencode, sondern Bytecode generiert wird.

### **3.3.3. Vorbereiten des RCX**

Ein unbenutzter RCX besitzt nur das Basis-Betriebssystem im ROM. Die Firmware muss zunächst mit dem IR-Sender ins RAM übertragen werden. Die nötigen Werkzeuge dafür bringt die Robolab Software mit. Neben der Firmware werden auch die fünf Standardprogramme, auf denen die Experimente dieser Einheit beruhen, übertragen. Da das RAM durch die Batterien des RCX gepuffert wird, geht die Firmware verloren, wenn der RCX länger als 2 Minuten nicht mit Strom versorgt wird. Dann wird ein erneutes Einspielen der Firmware nötig.

### **3.3.4. Schnittstellen des RCX**

Der RCX stellt mehrere Schnittstellen bereit. Zu den Eingabe Ports (Input Ports) zählen die drei Sensoreingänge (A/D Wandler) und die 4 Buttons. Als Ausgänge (Output Ports) stehen die Anschlüsse für drei Aktoren, das LCD und ein Lautsprecher bereit. Die Infrarotschnittstelle erlaubt bidirektionale Kommunikation (I/O Port).