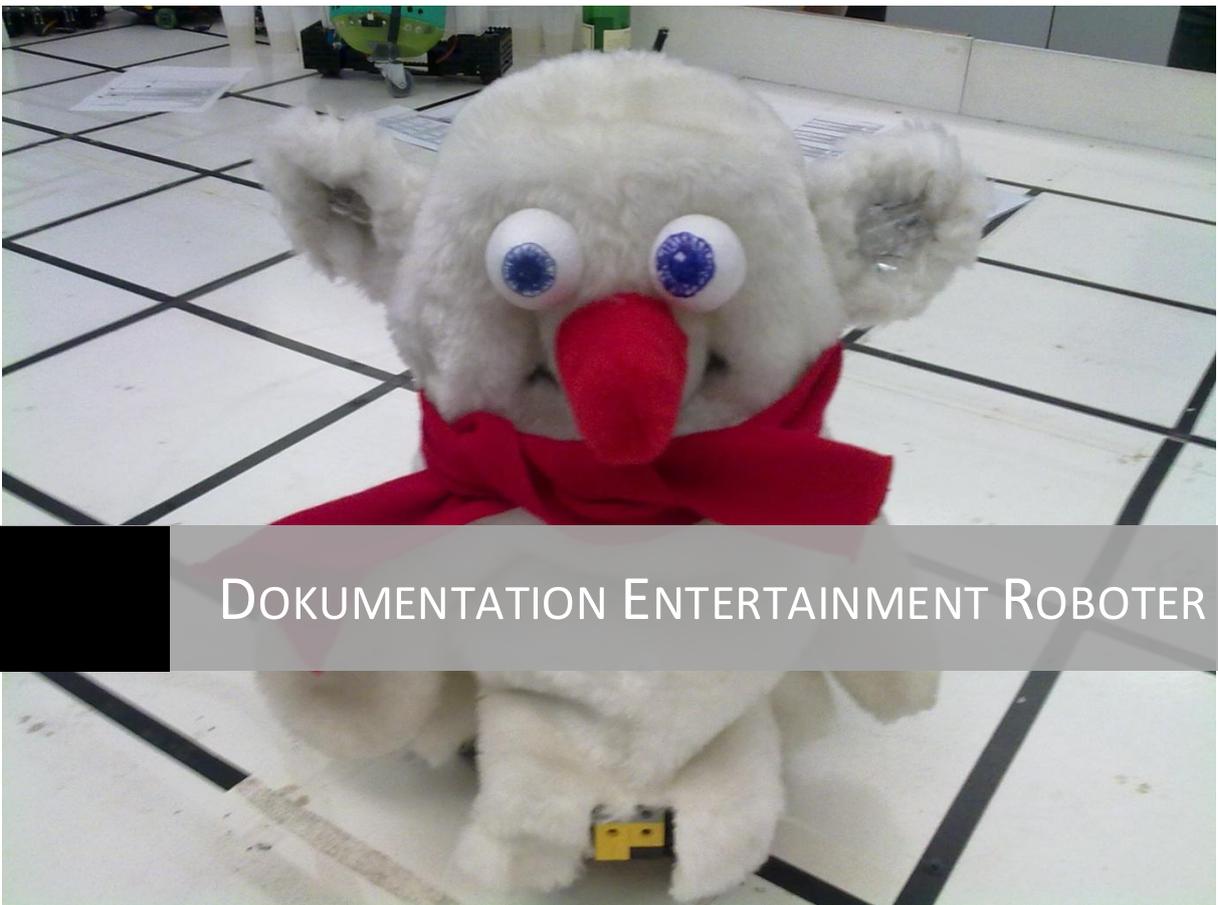


27.1.2010



DOKUMENTATION ENTERTAINMENT ROBOTER

Erstellt von Andre Morgenthal, Marcel Geßner, Stephan Lapoehn

Inhaltsverzeichnis

1. Konzept:	3
2. Umsetzung:	3
3. Inbetriebnahme	4
4. Material/Bauteile	4
3.1 Sensoren:	4
3.2 Aktoren:	4
3.3 Datenverarbeitung:	4
3.4 Chassis	5
4. Der Bewegungszyklus:	5
4.1 Priorität der Abläufe	5
4.2 Definition von Aktionen und Mustern:	5
4.3 Umsetzung des Bewegungszyklus	5
5. Augenbewegungen	7
6. Ohrenbewegungen	7
7. Schaltplan und Belegungen	8

1. Konzept:

Ziel des Projektes war es einen Entertainment-Roboter zu bauen. Daher dachten wir uns, dass ein interaktives Resultat das unterhaltsamste wäre. Wir kamen zu der Überlegung mit dem Roboter über ein Lichtsignal zu kommunizieren. Er sollte "menschens ähnliches" Aussehen bekommen und auf das vom Spieler ausgesandte Licht reagieren. Die Lichtsensoren in unserem Projekt sollten in den Ohren untergebracht werden, welche nach einem "Treffer" sich "schützend" weg drehen. Nachdem beide Ohren getroffen wurden ergreift er die Flucht. Während er auf der Flucht ist, ist er in der Lage Hindernissen auszuweichen oder anderweitig darauf zu reagieren z.B. rückwärts fahren oder Richtung ändern. Sollte sich niemand für ihn interessieren fährt er sinnlos im Raum herum und wartet auf eine Interaktion eines Spielwilligen.

2. Umsetzung:

Begonnen haben wir mit dem Bau des Chassis aus Lego. Dabei haben wir uns zuerst auf das Fahrwerk konzentriert und sind dann über den Kopf zur Lenkung übergegangen. Dabei haben wir versucht den Stromverbrauch, Lärmpegel und die Belastbarkeit des Roboters zu optimieren. Aus diesem Grund haben wir ein Getriebe gebaut, welches die Geschwindigkeit des Roboters reduzieren sollte und ihn ein wenig kräftiger machen sollte. Dieses Getriebe verkantete anfangs sehr stark, da es nicht sehr starr war und den antreibenden Motor nicht fixieren konnte. Aus diesem Grund verbrauchte er auch enorm viel Strom, was nicht nur zu einer schnellen Entladung des Akkus führte, sondern auch zu einer Überhitzung des AKSEN-Boards. Nachdem wir die Starrheit des Getriebes und die Fixierung des Motors behoben hatten, war unser Ziel die Geräuschreduktion. Dies haben wir mit Silikon Spray versucht, was sich als nicht sonderlich geeignet erwies, da es sich überall verteilte. Danach gingen wir zum Kopf über. Dieser sollte 4 Servomotoren enthalten, welche die Ohren des Roboters drehen, einen Motor für die Drehung der Augen und vier optische Widerstände um die Treffer der Ohren zu realisieren. Der Kopf erhielt dadurch so viel Gewicht, dass dieser enorm am Roboter befestigt und verstärkt werden musste. Nun kamen wir zur Lenkung. Diese befindet sich am Heck des Roboters und musste somit noch einen Schalter beherbergen um auf Kollisionen bei rückwärtsfahren reagieren zu können.

Parallel haben wir uns in Grundzügen mit der Programmierung des AKSEN-Boards auseinandergesetzt. Dabei haben wir versucht die Lichtreaktion des Roboters umzusetzen. Dies hat bis dato ausgezeichnet funktioniert, jedoch mussten wir dieses Programm aufgrund des autonomen Fahrens noch mehrfach umstoßen.

Nun haben wir begonnen die Ohren des Roboters zu bauen. Diese sollten dem späteren Schneemannchassis entsprechen, mussten einen optischen Widerstand enthalten und drehbar sein. Nachdem diese fast vollständig installiert waren kamen wir auf die Idee unserem Roboter Augen zu spendieren. Diese sollten sich später bei Treffern der Ohren drehen. Als diese angebracht waren und auf die Lichtsignale reagierten gingen wir zum autonomen Fahren über.

Abschließend galt es noch den Roboter zu verschönern. Dazu nahmen wir den anfangs erwähnten Schneemann aus, um seinen Außenstoff als Verkleidung zu benutzen. Diese musste nicht nur an verschiedenen Stellen Öffnungen für Sensoren und Aktoren haben, sondern auch gut angebracht sein damit sich der Stoff nicht in den beweglichen Teilen verfängt.

3. Inbetriebnahme

Um den Entertainment-Roboter in Betrieb zu nehmen sollte er zuerst auf einer Ebenen Fläche stehen. Zuerst sollte man darauf achten, dass das Fell keine der Sensoren vorne, rechts, links und oben überdeckt. Oft reagiert der Roboter auch auf die herunterhängenden Arme, diese sollten aus dem Sichtbereich entfernt werden. Damit sich die Augen drehen können, sollten sie möglichst wenig Kontakt mit dem Fell haben. Um den Roboter nun zu starten, muss das Fell an der Rückseite hochgenommen werden und der auf dem AKSEN-Board befindliche Schalter umgelegt werden. Nun müsste der Roboter anfangen zu agieren. Falls nicht sind vermutlich die Batterien zu schwach oder das AKSEN-Board kaputt.

4. Material/Bauteile

Die verwendeten Bauteile teilen sich in Sensoren, Aktoren, Datenverwaltung und Chassis auf.

3.1 Sensoren:

1. Abstandssensoren:

1. Infrarotsensor vorne
2. Infrarotsensor vorne links
3. Infrarotsensor vorne rechts
4. Schalter hinten

2. Ohrensensoren:

1. Optischer Widerstand im linken Ohr
2. Optischer Widerstand im rechten Ohr
3. Optischer mittig auf dem Kopf zur Umgebungslichtmessung

3.2 Aktoren:

1. 4 Servomotoren im Kopf
2. Servomotor am Heck
3. Motor als Antrieb vorn
4. Motor zum Antrieb der Augen am Kopf
5. 2 Infrarotdioden an den Seiten
6. 2 Infrarotdioden vorn

3.3 Datenverarbeitung:

1. Die Sensordaten werden durch das AKSEN-Board erkannt, verarbeitet und auf die Aktoren geleitet.

3.4 Chassis

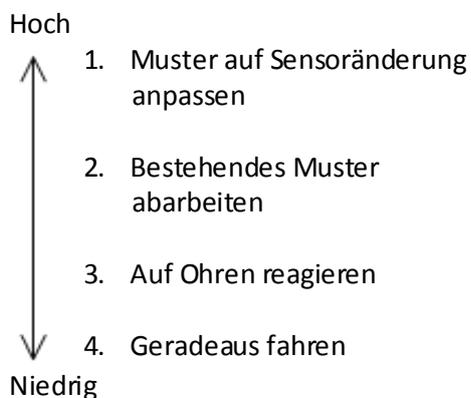
1. Plüschschneemann als Überzug
2. Lego zum Bau des Rahmens und Halterung der Sensoren, Aktoren und des AKSEN-Boards
3. Ohren aus den Resten des Schneemanns, welche die Sensoren zur Kommunikation enthalten
4. Stützräder zur seitlichen Stabilisierung der Roboters (LEGO)
5. 2 Antriebsräder an der Front (LEGO)
6. Lenkbares Rad am Heck (LEGO)
7. Getriebe zur Reduktion der Geschwindigkeit (LEGO)
8. Kabelbinder
9. Zwei Styroporkugeln die als Augen dienen

4. Der Bewegungszyklus:

Die Bewegungsschleife arbeitet mit Bewegungsmustern. Für jede Art von möglichen Ereignissen (z.B. Sensor vorne reagiert oder Sensor links reagiert) existieren ein oder mehrere Bewegungsmuster. Mit den Bewegungsmustern lassen sich komplexe Bewegungen ausführen, die über mehrere Aktionen verfügen können. In jeder Einzelaktion des Musters wird Zeit, Richtung (rechts, links) und Richtung (vorwärts, rückwärts) gespeichert. Wenn einer der 4 Abstandssensoren reagiert wird ein Muster gestartet und dieses ausgeführt. Diese ist solange aktiv bis entweder das Ende des Musters erreicht ist oder ein anderer Sensor ein neues Muster aktiviert. Ist keine Muster aktiv kann der Roboter seine Fahrtweise auf Ereignisse wie beispielsweise Treffer in die Ohren anpassen. Sollte ein solches Ereignis nicht stattfinden fährt der Roboter einfach geradeaus.

Um Deadlocks (in Ecken hängen bleiben) zu vermeiden gibt es für einzelne Ereignisse mehrere Bewegungsmuster. Welches davon verwendet wird entscheidet eine Zufallsfunktion.

4.1 Priorität der Abläufe



4.3 Umsetzung des Bewegungszyklus

4.2 Definition von Aktionen und Mustern:

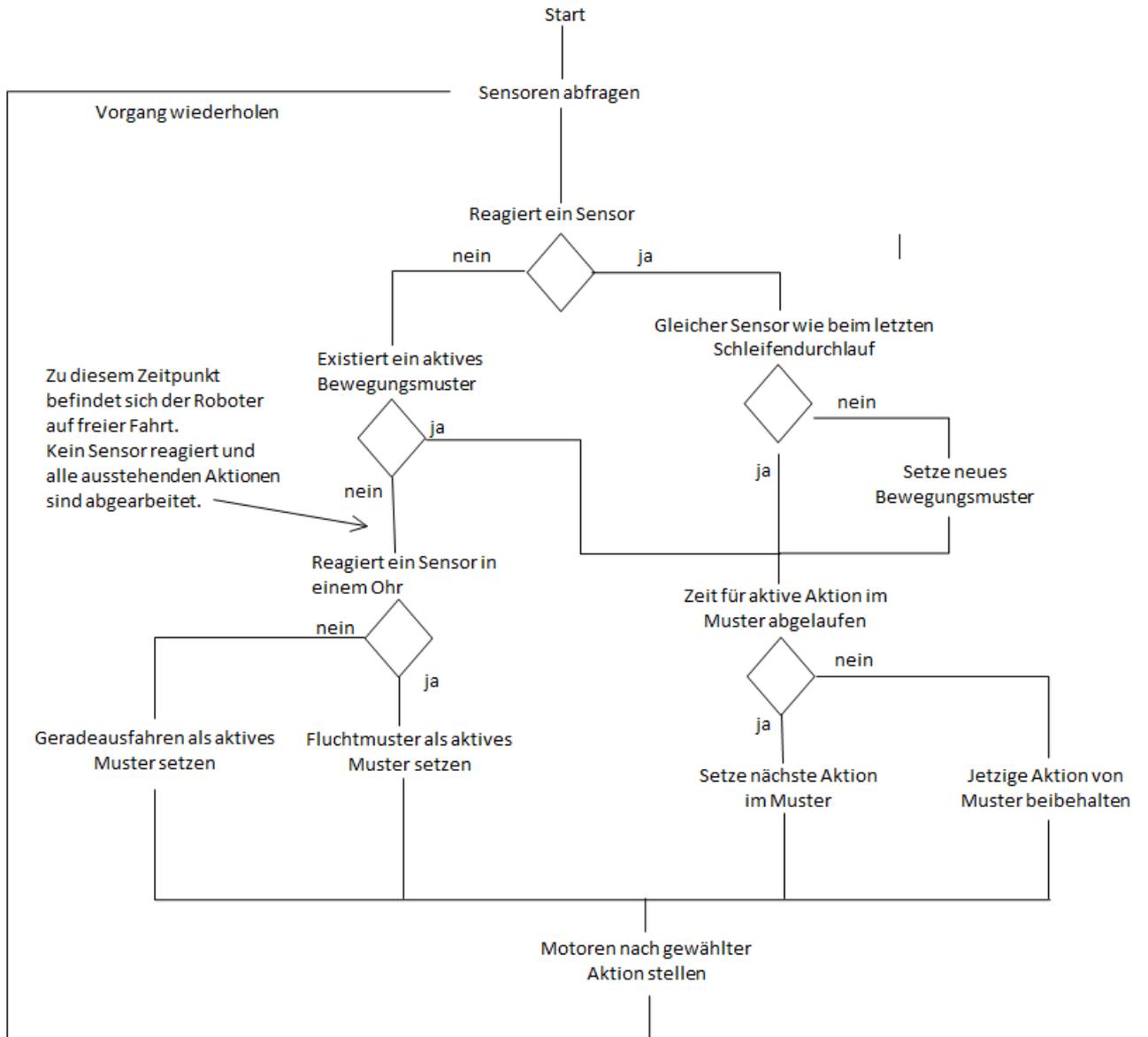
```
#define UP 1
#define DOWN 0

#define LEFT 0
#define RIGHT 1
#define STRAIGHT 2
#define HALFLEFT 3
#define HALFRIGHT 4

//Einzelaktion
typedef struct action{
    int time; //Aktionszeit in ms
    int motorSpeed; // vorwärts, rückwärts
    int direction; // rechts ,links, halbrechts,
    halblinks
    //..zusätzliche Einstellungen
}Action;

//Bewegungsmuster
Action vorneLinksHindernis[] = { {1500, DOWN,
HALFLEFT} , {1000, UP, HALFRIGHT} };

Action vorneRechtsHindernis[] = { {1500, DOWN,
HALFRIGHT} , {1000, UP, HALFLEFT} };
```



5. Augenbewegungen

Die Augen lassen sich für eine dynamisch einstellbare Zeit nach links oder rechts drehen. Dabei kann der Aufrufer entscheiden, ob sich die Augen nach der Ausführung der Aktion zurückdrehen sollen oder nicht.

Die Augen werden durch eine Methode gesteuert, welche als Parameter Zeit und Richtung erwartet sowie einen Parameter der bestimmt ob sich die Augen Hin und Her oder nur im Kreis drehen sollen (`moveEyes(int _richtung, int zeit, int _hinher)`). Die Umsetzung der Ausführung wird in einer weiteren Methode gesteuert. Diese Methode muss jeden Schleifendurchlauf des AKSEN-Boards aufgerufen werden (`eyeControl()`). Während der Ausführung einer Aktion blockiert die Methode `moveEyes` die Einstellungen. Erst nach der Abarbeitung kann eine neue Augenbewegung mit anderen Parametern gestartet werden.

6. Ohrenbewegungen

Um die Ohren zu bewegen wird zuerst der Umgebungslichtsensor am Kopf abgefragt. Dieser Wert wird mit der Lichtstärke in den jeweiligen Ohren verglichen. Wenn der Wert der Sensoren in den Ohren um ein vielfaches größer ist als der Wert am Kopfsensor, hat ein Treffer stattgefunden. Der Servomotor am Ohr wird nun auf eine bestimmte Position gestellt und bleibt solange in dieser, bis eine gewisse Zeit abgelaufen oder der Unterschied zwischen den Sensoren nicht mehr so extrem.

7. Schaltplan und Belegungen

LED Ports:

- 0 -> Infrarot Vorn
- 1 -> 2te Infrarot vorn
- 2 -> rechte Infrarot LED
- 3 -> linke Infrarot LED

Motor Ports:

- 0 -> Antriebsmotor
- 1 -> Augenmotor

Servomotoren Ports:

- 2 -> rechtes Ohr
- 1 -> Ohr oben rechts
- 0 -> Ohr oben links

Digital Ports:

- 0 -> Kollisionsschalter hinten
- 9 -> Antrieb linkes Ohr
- 8 -> Lenkung

Analog Ports:

- 0 -> Infrarotsensor vorn
- 1 -> Infrarotsensor rechts
- 2 -> Infrarotsensor links
- 8 -> linkes Ohr
- 9 -> rechtes Ohr
- 10 -> Umgebungslicht

