

2. Wettbewerb

2010 auf dem Mars

Aufgabenstellung

MARS PATHFINDER MISSION STATUS

December 4, 1996

NASA's Mars Pathfinder is reported to be performing well on the first day of a seven-month journey to the red planet following a perfect launch today from Cape Canaveral, FL at 1:58 a.m. Eastern time.

"The command and data telecommunications subsystems are working perfectly, sending down data at 1,183 bits per second. The temperature control and propulsion subsystems reported all temperatures and pressures are within expected ranges. All systems are healthy," he said. Pathfinder is traveling away from Earth at a speed of 3.9 kilometers (2.4 miles) per second ...

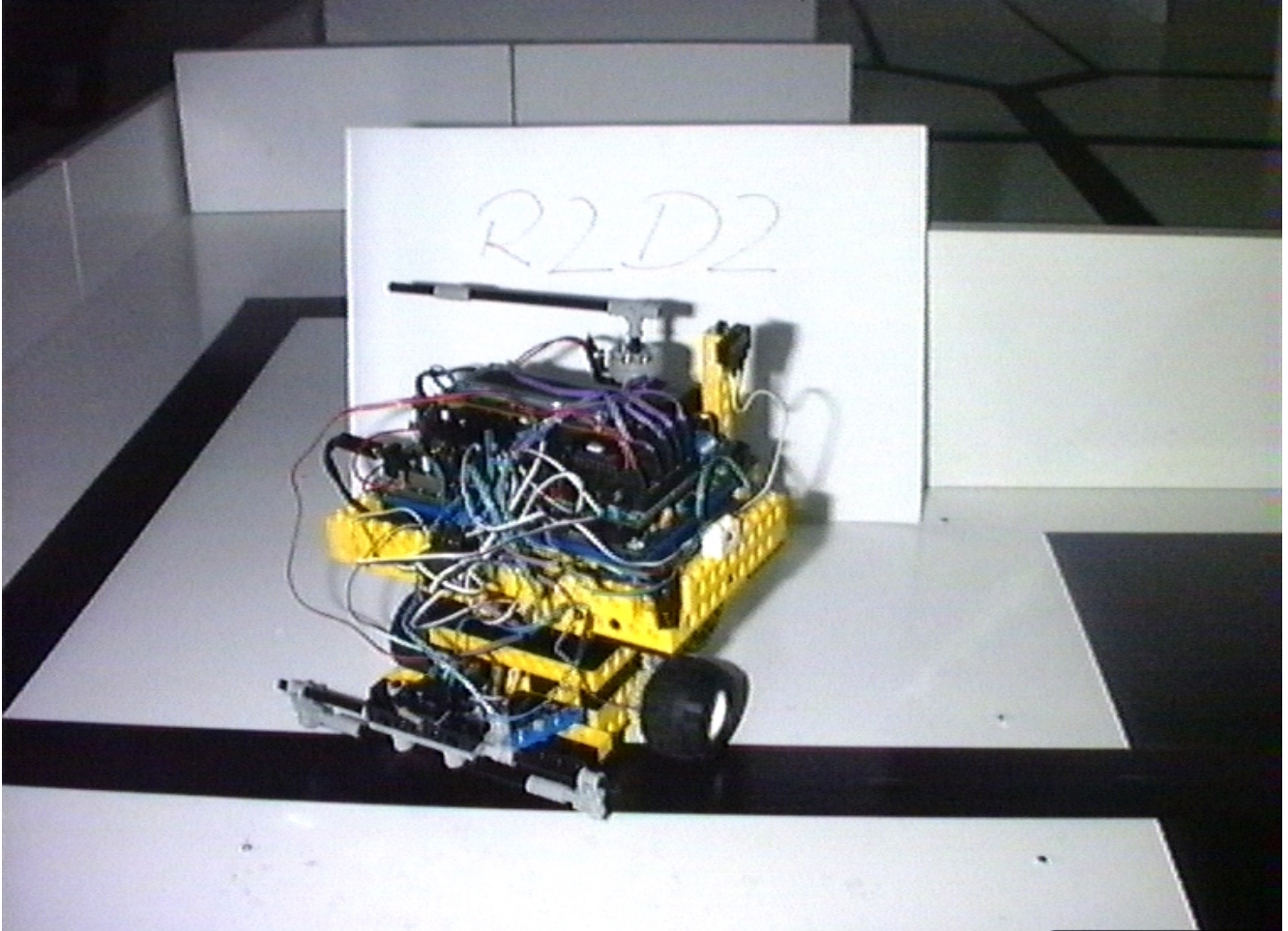
MARS PATHFINDER MISSION STATUS

May 2, 1997

The spacecraft remains in good health and is currently about 98 million kilometers from Earth. No significant spacecraft activities were performed this week. The total flight time since launch is now 149 days, and we have 63 days until Mars arrival ...

An Bord des PATHFINDER wartet ein autonomes mobiles System auf die Landung am 4. Juli 1997 - der sechsrädrige *Sojourner* des Jet Propulsion Laboratory in Kalifornien.

Denken sie 13 Jahre weiter - im Jahre 2010: wieder auf dem Mars, wieder AMS, aber diesmal ist die Aufgabe *Ihres Mars-Roboters* das Finden von Marsbasen und ihre Versorgung.



Regeln

Eine Runde, in der 2 AMS gegeneinander starten, besteht aus 2 Teilen:

- Teil EINS: Erkundung der Marsoberfläche
- Teil ZWEI: Versorgung der Marsbasen

Jedes AMS bestreitet 2 Runden, so daß der gesamte Wettbewerb aus 4 Runden besteht.

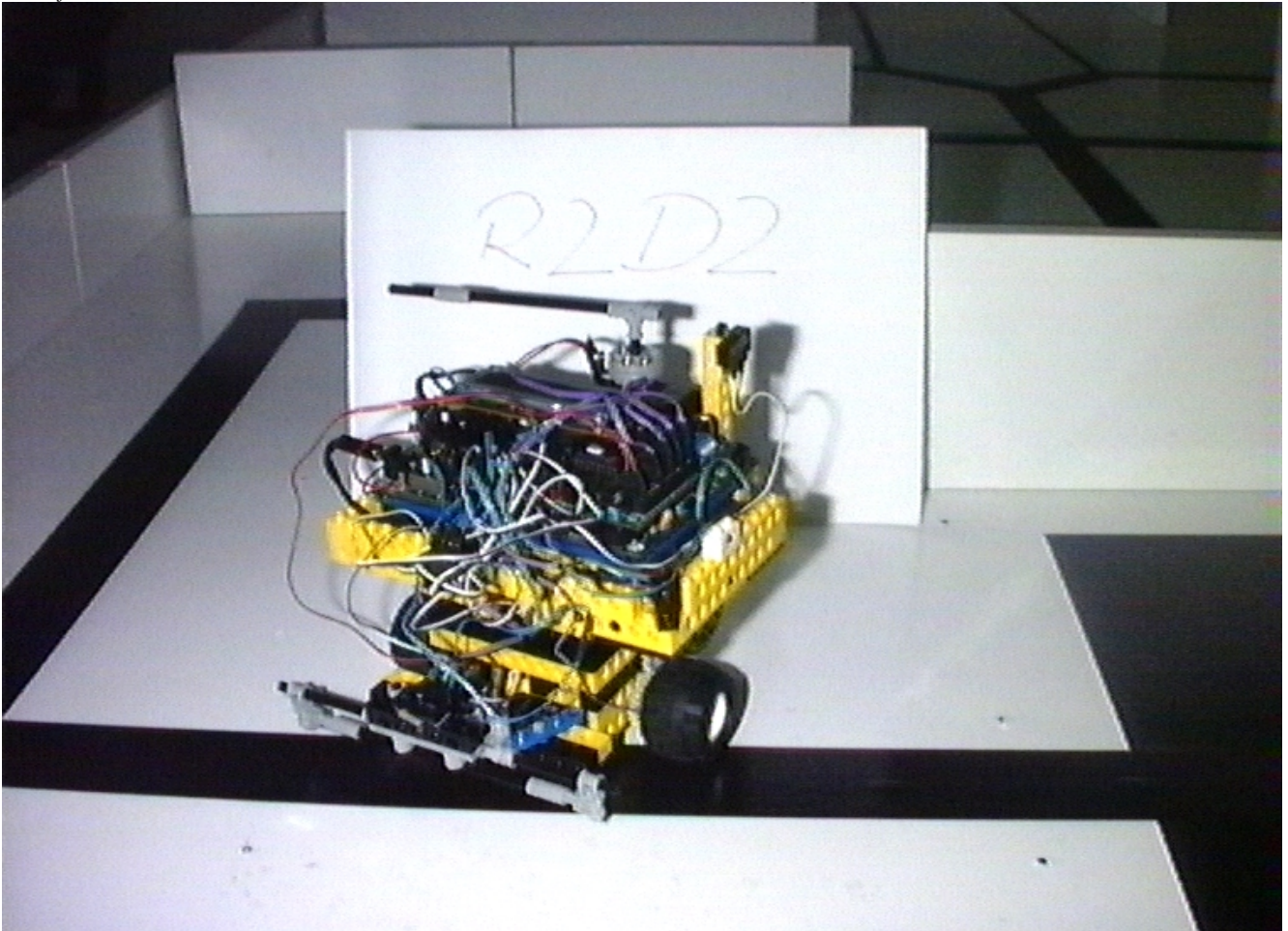
Die Startpositionen werden durch die Jury bestimmt, die Startrichtung ist den Teams überlassen. Gestartet wird mit dem üblichen Lichtsignal. Um *starkes* Gerangel zu vermeiden, sollte Ihr AMS in der Ausdehnung die Fläche DIN A4 nicht überschreiten.

Aufgabe im Teil EINS: Erkundung der Umgebung

Ziel dieses Teils ist die Erkundung einer unbekanntem Umgebung - der *Marsoberfläche*, dargestellt durch ein rechtwinkliges Labyrinth, welches erst kurz vor dem Wettbewerb realisiert wird.

Die AMS starten zeitgleich in disjunkten Labyrinth, und versuchen eine interne Repräsentation ihres Labyrinth aufzubauen. Die maximale Erkundungsdauer beträgt 120 Sekunden, dann sollten die AMS ihre Aktionen einstellen. Falls ein AMS die Erkundung früher beendet, wird dies durch das Heben einer Fahne angezeigt. Die Schnelligkeit und der Umfang der Erkundung werden wie folgt bewertet:

- **Umfangspunkte:** 5 * Anzahl gefundener 30cm-Strecken
- **Siegpunkte:** 20 Punkte für das AMS, welches als erstes das gesamte Labyrinth erkundet hat (Fahne!)
- **Qualitätspunkte:** 50 Punkte, wenn die erzeugte Repräsentation im Teil ZWEI verwendet wird.



Aufgabe im Teil ZWEI: Versorgung von 5 Marsbasen

Das intern erzeugte Abbild des Labyrinthes aus dem Teil EINS sollte im Teil ZWEI verwendet werden. Falls jedoch die Ausnahme eintreten sollte und sich ein Team entscheidet, auf die **Qualitätspunkte** zu verzichten, so steht eine nachladbare Version des Labyrinthes in IC zur Verfügung (s.u.).

Es starten beide AMS direkt im Anschluß an Teil EINS (ggf. nach dem Nachladen des Labyrinthes) zum Teil ZWEI.

Ziel ist die möglichst schnelle Versorgung **aller 5 Marsbasen** mit lebenswichtigen Materialien (Energie, Wasser, Benutzerhandbücher). Dazu sind alle Marsbasen in beliebiger Reihenfolge anzufahren. Das Laden/Entladen an einer Marsbasis ist durch Heben und Senken der Fahne anzuzeigen. Das Zeitlimit beträgt wie immer 120 Sekunden. Folgende Punkte werden vergeben:

- **Erfüllungsgrad:** 10 * Anzahl angefahrener (verschiedener) Marsbasen
- **Geschwindigkeit:** 20 Punkte für das AMS, welches die Versorgung als erstes beendet hat
- **Qualitätspunkte:** 50 Punkte für den optimalen Weg

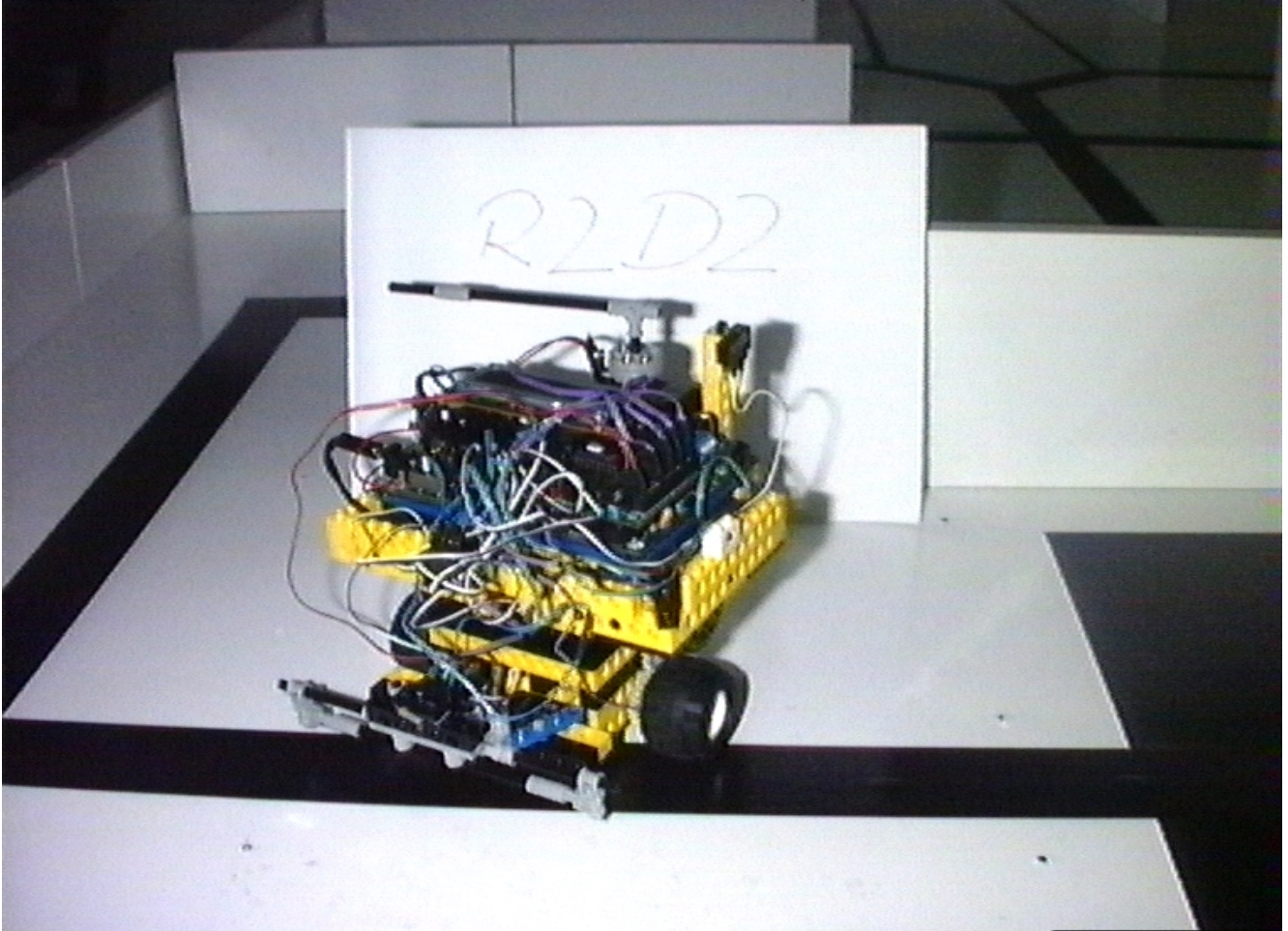
Disqualifikation

Ein Roboter wird aus der Runde entfernt (physisch), wenn er in das Labyrinth des anderen fährt! Bis dahin erreichte Punkte bleiben erhalten.

Ein Roboter wird disqualifiziert, wenn:

- er nach Zeitablauf nicht anhält oder vor dem Startsignal losfährt
- er von Teammitgliedern während der Runde Informationen erhält

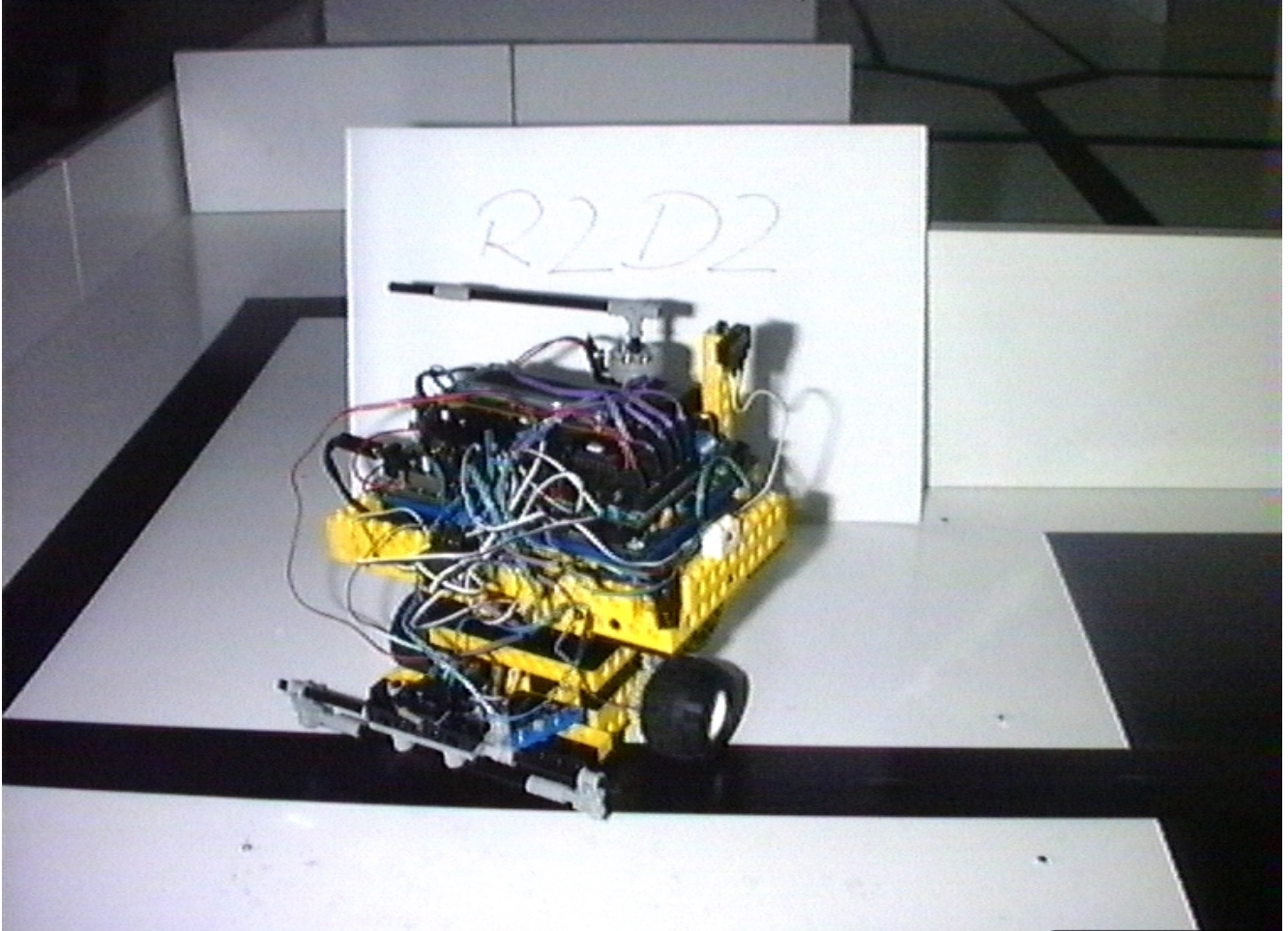
Ein disqualifizierter Roboter erhält für diese Runde keine Punkte.



Das Labyrinth

unterliegt folgenden Konstruktionsregeln:

- nur rechtwinklige Kreuzungen, Ecken, Abzweigungen
- Sackgassen vorhanden
- alle Kreuzungen, Abzweigungen, Ecken, Sackgassen, Marsbasen im 30cm-Raster
- keine Wandberührung
- 2 disjunkte Teillabyrinth für gleichzeitigen Start von 2 AMS
- maximale Anzahl von 30cm-Kanten in einem Labyrinth: 30
- maximale Anzahl von 30cm-Kanten im optimalen Weg: 20
- Darstellung der Marsbasen :

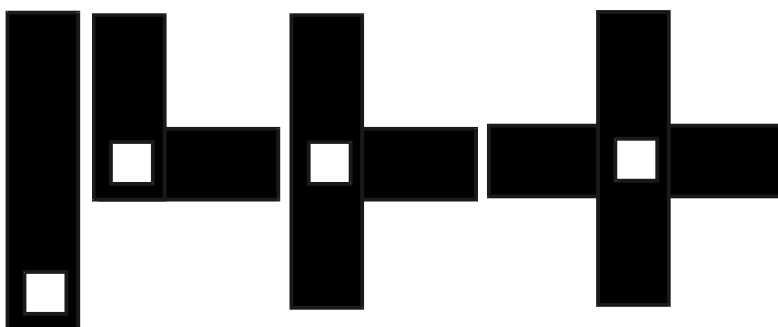


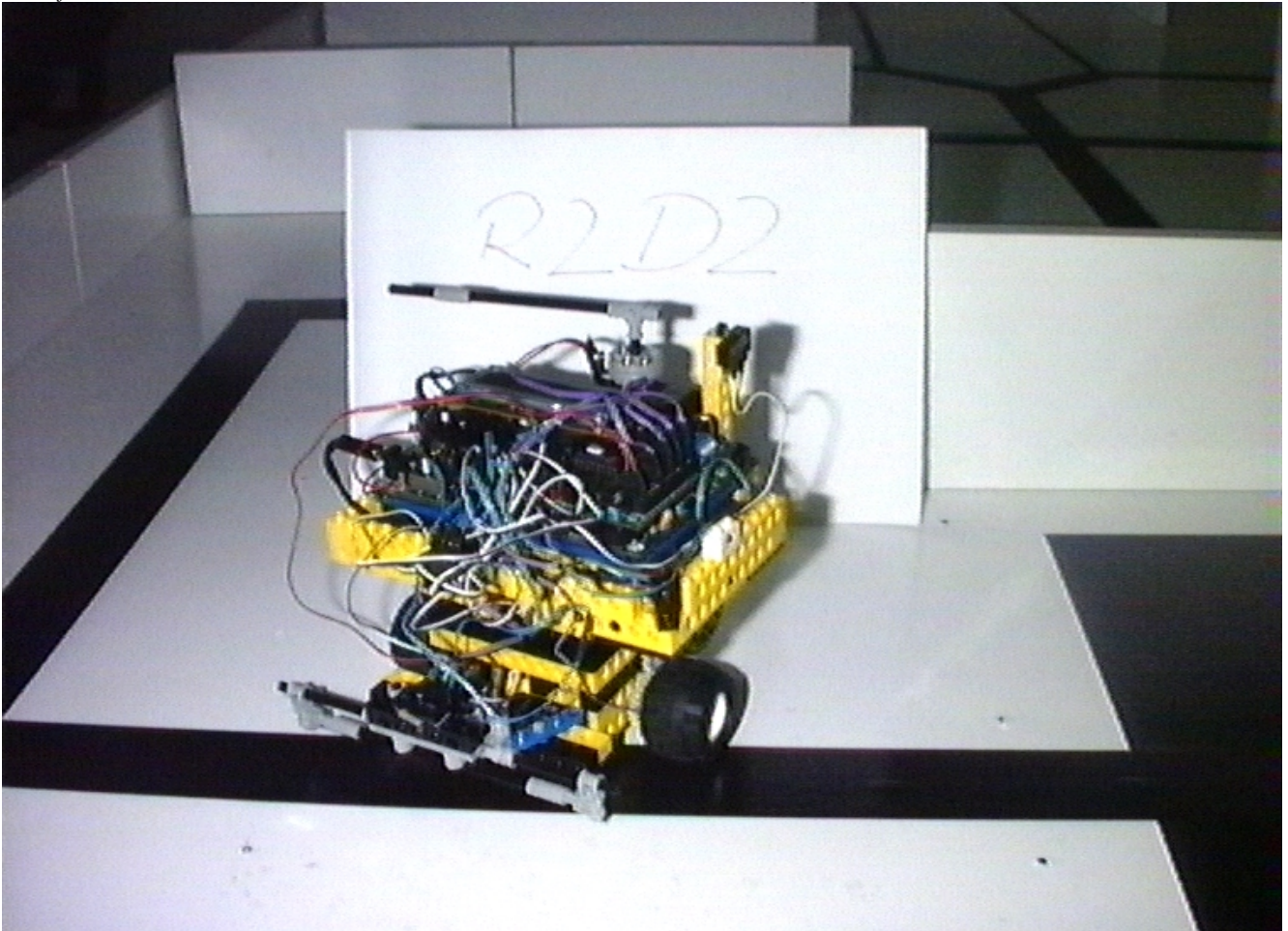
Hinweise

Hier einige Tips, die Sie beachten könnten:

- Shaftencoder nutzen
- perfektes Linienfolgen und Abbiegen nötig
- Wissensrepräsentation des Labyrinthes: persistente Variablen (ICM S.123)
- in IC nicht möglich: Array von Arrays, dynamische Speicherausfassung, Strukturen (struct), char-Variablen (nur als Array)
- Grenze des lokalen Stack beachten (ICM S. 142)
- Speicherplatz sparen!
- Breitensuche

Eine Marsbasis ist durch weißes Viereck auf dem schwarzen Labyrinth symbolisiert, also z.B.





- Aufgabe Teil ZWEI in 2 Phasen lösen:
1. Plan erstellen und
 2. Plan ausführen

Laden des Labyrinthes als C-Code

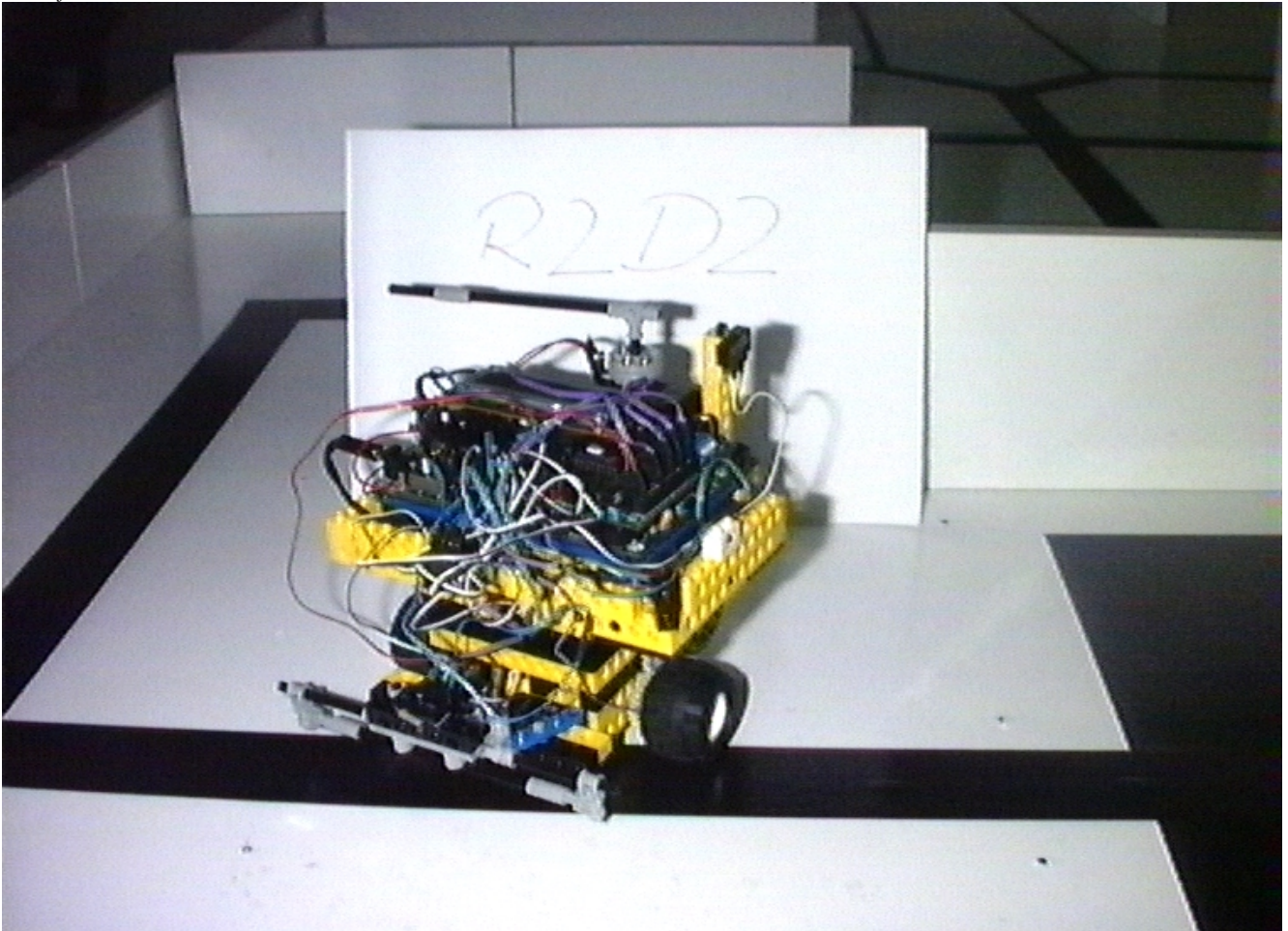
Auf der Platte befinden sich zwei Labyrinth - bezeichnen wir das Labyrinth zur Fensterseite mit **LabF** und das zur Wandseite der Platte mit **LabW**. Für beide Labyrinth gibt es je eine C-Datei, in der es beschrieben ist: M:\GET_LABF.C und M:\GET_LABW.C. In diesen Dateien ist jeweils die Funktion

void Get_Lab(int LabA[30], int LabB[30], int LabS[5])

definiert. Die Funktion füllt die übergebenen Arrays (siehe auch ICM S. 131) mit den Daten eines Labyrinthes. Wir stellen uns dazu alle Punkte der Platte durchnummeriert von 1 .. 45 beginnend in der NW-Ecke des 30cm-Rasters vor. Eine existierende Kante ist beschrieben durch 2 Punkte, also z.B. Punkt 2 und Punkt 3. Der kleinere der beiden Punkte einer Kante wird in das Array **LabA**, der andere in das Array **LabB** eingetragen. Die Reihenfolge der Kanten ist primär durch die A-Punkte und, falls diese gleich sind, durch die B-Punkte definiert. Das dritte Array **LabS** enthält die Punkte, an denen sich Marsbasen befinden (genau 5).

Diese Datei muß geladen werden, bevor Sie die Funktion im anderen Quelltext benutzen können (für Fortgeschrittene: Eintragen in das File lib_r22.lis zum Laden beim IC-Start).

Sie sollten das Einlesen des Labyrinthes mit Get_Lab(..) von einem externen Sensor (z.B. DIP-Schalter) abhängig machen, denn dieses Einlesen kostet Sie 50 (in Worten: fünfzig) Qualitätspunkte im Teil EINS.



Wann

Abschlußwettbewerb: 25. Kalenderwoche, Donnerstag, den 19. Juni 1997, 14:30 Uhr im KI-Labor