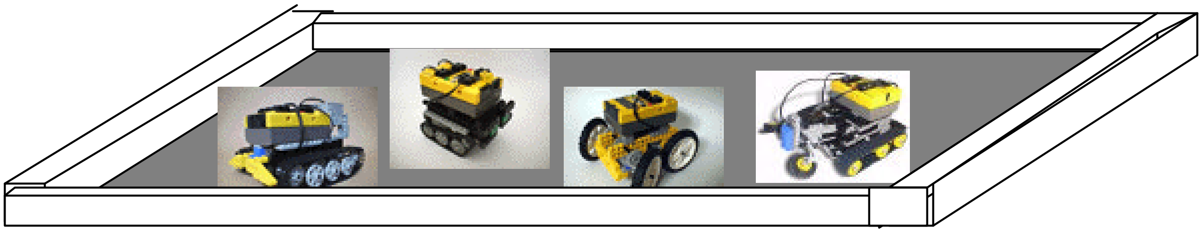
Robotics Design Project

Introduction:

The project will involve working in teams to design and build a robotic wheelchair using LEGO Mindstorms Robotics Invention System to compete in an obstacle course.



Problem Definition:

The objective of this project is to build a computer controlled robot that can safely deliver an immobile person through an obstacle course in the shortest amount of time. To simulate real world situations, the robot must be able to climb a small ramp, cross a street without getting hit by a car, turn corners, fight off aggressive animals, climb stairs and free itself from a sandpit.

Educational Goals:

The goals of this project are as follow:

1. To provide students with hands on experience building a simple programmable robot.
2. To demonstrate that design processes typically involve a multitude of skills and knowledge from many subject areas.
3. To familiarize students with the design process- from brainstorming, initial design, prototyping, testing, revising, to final production and competition.
4. To allow students to experience the perilous designer/builder interface.
5. To spark student's interest in Science and Technology.
6. To win some Money ($100 winners take all)

References:

LEGO MINDSOTRMS Robot , Jonathan B. Kundsen , 1st edition, 1999, O'Reilly
http://www.crynwr.com/lego-robotics/
http://www.plazaearth.com/usr/gasperi/lego.htm#background
http://www.oreilly.com/catalog/lmstorms/resources/index.html
http://www.robotbooks.com/

Competition Guidelines

The objective is to build a computer controlled Robot that can deliver an immobile person through an obstacle course in the shortest time without losing him. To simulate the real world situation, the wheelchair must be able to climb a small ramp, cross a street without getting hit by a car, turn corners, fight off aggressive animals, climb stairs and free itself from a sandpit. For a specific floor plan, please refer to the Playing Field section. The contest is also subject to a few ground rules. Please follow them carefully before begin your design.

Teams

The team will consist of three or four members. Each member is responsible for a part of the design and construction. Great emphasis will be placed on teamwork. Evaluation of contribution from each team member will play a big part in the team's final grade.

Obstacle Course

The layout is shown in Figure 1. The playing field is on a hard concrete floor (somewhere in the Learning Factory). A wall made of stack of two 2x4x8 lumber surrounds the playing field. A 4” tall ramp is placed immediately after the starting block. A crosswalk on a two way traffic is followed. A master controller is placed near the crosswalk that directs the traffic and the wheelchair. It follows by a corner turn. After the turn, robot is confronted a couple unfriendly robots. Wheelchair must be able by pass these robots and proceed onto an up and down stair with 0.25” steps and finally finish off at a sand pit.

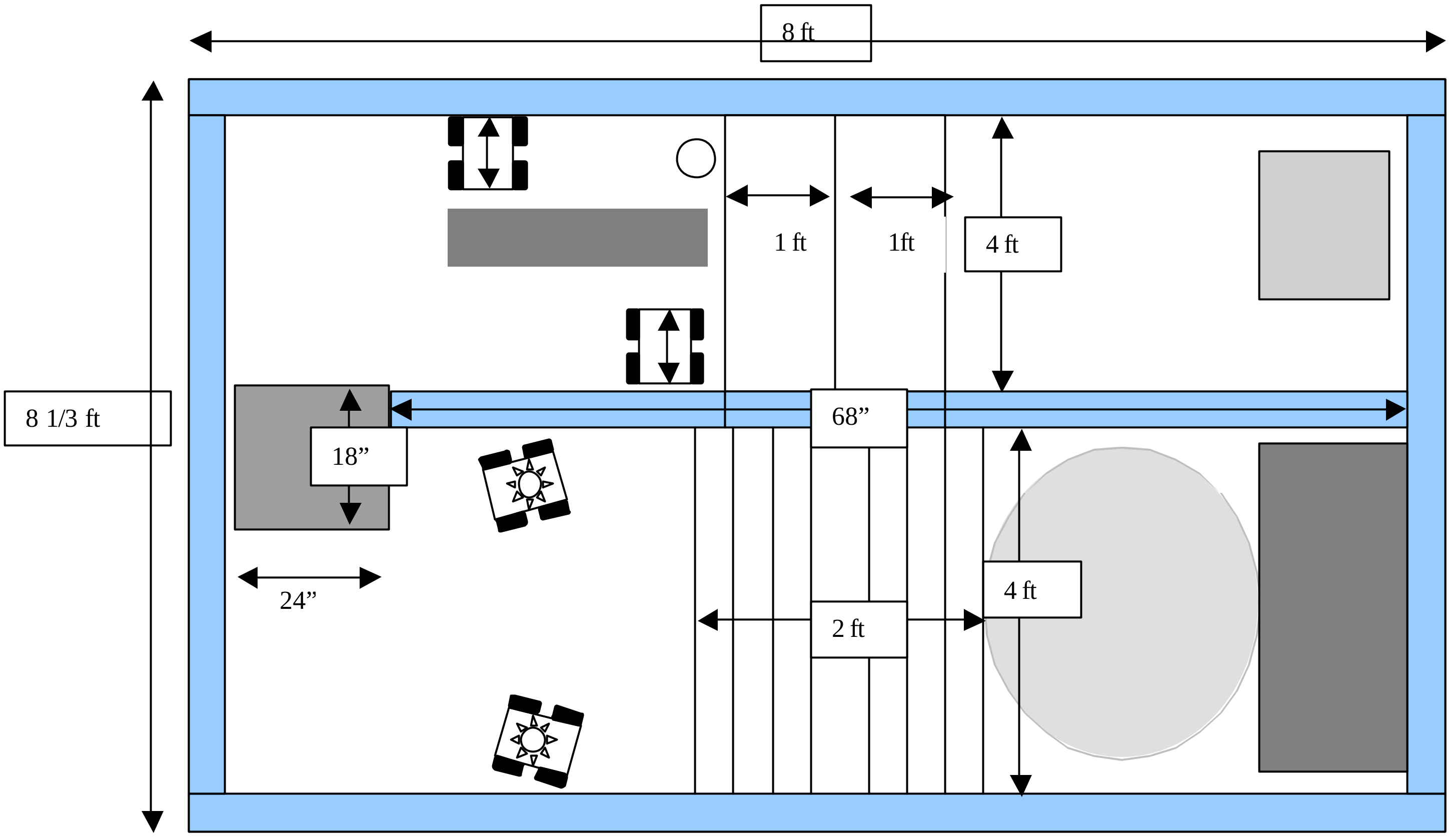


Figure 1

🟦	Non-Painted wooden walls - stack two 2x4x8	■	Cross walk – made of 2 pieces of 8”x11” printer paper
▭	Starting block	▭	Up and down stair – 8 steps, Step- 0.25” height 3” wide Highest step is 6” wide
☰	Pivoted plate form - 4” peak	▬	No defender zone
○	Master controller- send message to stop the traffic and halt the wheelchair	⬭	1/4” deep Sand
		■	Finishing line
⇔	Car moving back and forth	☼	Unfriendly robots

Robot Constraints:

Size:

The maximum size of the Robot shall be 12" by 12" by 12 ". The Robot can not look over the walls of the structure and must never extend itself beyond 12 inches in any dimension. All Robots will be carefully measured. Don't let your Robot be disqualified because it is slightly over the limit.

Weight:

There are no restrictions on the weight of the Robot.

Functions:

The robot must have the comfort and look of a regular wheelchair. The robot must be able to support and hold a passenger - monchichi (will be provided) without losing it during the crossing.

Programs:

The robot must have three programs installed on the RCX:

Program 1 - Robotic Wheelchair

> This program provides the navigation for the robot to maneuver through the obstacle course. In additional to clearing the obstacles, this program needs to ensure safe passage across the crosswalk. It does this by executing Message 1 (stop) and Message 2 (start again) sent from the Master Controller* when the robot reaches the edge of the crosswalk.

Program 2 - Car

> This program causes the robot to behave as a car. A car is defined as a robot that moves forward and backwards across a narrow (4 ft) passage. Additionally, it needs to be programmed to stop for crossing pedestrians (like the robotic wheelchair). It does this by executing Message 1 (move away and stop) sent from the Master Controller* when the robotic wheelchair reaches the crosswalk.

Program 3 - Defender Robot

> This program requires the robot to perform certain actions in attempt to disable competing robots. This robot can have any mechanism or electronic device that can prevent the robotic wheelchair from proceeding through the obstacle course. Additionally it needs to incorporate a value from a light sensor to keep it from entering the "defense free zone".

---

*The Master Controller is an RCX that is programmed by the instructor. It sends out infrared signals when the robotic wheelchair passes in front of its light sensor. The "signals" do not contain directions, but alert robots in the vicinity to review content coded on their RCX. Thus, you will need to program your robot to execute the messages. It will send out a signal for robots to execute Message 1 (Wheelchair – "Stop" and Car - "Move away and stop") and then wait several seconds before sending out Message 2 (Wheelchair - "Go forward").

Ground Rules

The following set of constraints must be adhered to in the implementation of your respective designs. If you need further clarification, ***ask before you implement***!!!

----

1. Robots must start from rest and not to be lifted off the track during starting.

2. No human intervention is allowed during the match.

3. Your machine must be self-contained and self-sufficient. In other words, it must provide its own energy. No "plug-ins" allowed.

4. Robots must be powered by an electric motor. No fuel engine or rocket propulsion is allowed.

5. Robots can be constructed with Lego bricks, or with any type of materials. The structure of the robot, including wheels and legs, can be strengthened or enhanced by any means. Unless otherwise stated, there are no restrictions on the types of motors to be used.

6. You may alter any Lego parts; however, all alterations and manufacturing must be approved by instructor.

7. Using homemade sensors or gadgets (without destroying the existing Lego parts) are encouraged. There are numbers of Lego Mindstorms related web sites deal with homemade sensors. Some are listed in the procedure.

8. The ambient light level in the contest area is impossible to determine until the actual day of the contest. Contestants will be given time on the contest day to make ambient light level readings if necessary to calibrate their Robot. The room will be lit by overhead florescent lamps.

9. All supplemental, not specified parts (if any) that your design uses must be accordingly priced* and approved by your Instructor.

10. All parts necessary for your machine to be built must be itemized.

11. No competitor shall employ devices that compromise the safety of competition spectators or machines. Machines deemed unsafe will be banned from the competition. Unsafe machines will be permitted to re-enter after the unsafe feature is removed.

12. Weapon designed to temporally disable or to throw the other team off balance is allowed. However, device must be operated under the strict safety code of rule # 8. Devices using projectiles (tethered or otherwise), rockets, explosives, open flame, caustic chemicals, fluid or cut-off discs are strictly forbidden. BTW, Flying Robots are not permitted. The cost of the device * must be included in the calculation.

Scoring equation:

Final score will be awarded based on the following equation. It is intended to make the contest as realistic and as fair as possible. Apologies are made if it reminds you of the federal tax code.

$$P = [\frac{1}{2}(\frac{S}{S_{\max}} + \frac{T_{\min x}}{T})] + Ex$$

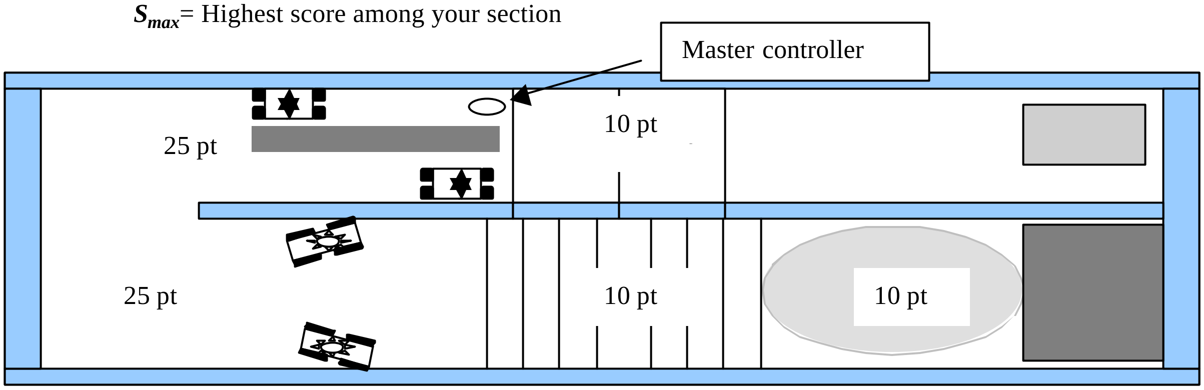
P = Final scores (%)

T= Total time used in the obstacle course
(excluding playoff)

T_{min}= Shortest time used in the obstacle course by any team in the competition

S = Your score from the course (see Figure below to calculate the score)

S_{max}= Highest score among your section



- 25 points are awarded to team that can program the robot to stop at the crosswalk when the "message # 1" signal was received from the master controller and it proceed after receiving the "message # 2".

- 25 points are awarded to team that program the robot to stop and back away from the crosswalk when "message # 1: was received from the master controller.

- 25 points are awarded to team that either fight off the unfriendly robot or stop the robotic wheelchair from moving forward.

- 10 points each are awarded to team that either go over a 3" ramp, ¼" steps or sand pit

* For retail pricing -- any of the following are acceptable:

- A receipt
- A "tear-out" of an ad from a newspaper or magazine, clearly indicating the item price.

Ex = extra credit for ingenious design

RANKING:

1. **The competitor with the best time win the competition and to be awarded the top prize of $100 in cash!!!!!!!!!!!.**
2. The WINNER's score will also get the maximum 20%.
3. The scores of all other teams will be calculated on the basis of the winner's score.

Price List

Gears- 40 cents each

Motors - $3.00 each

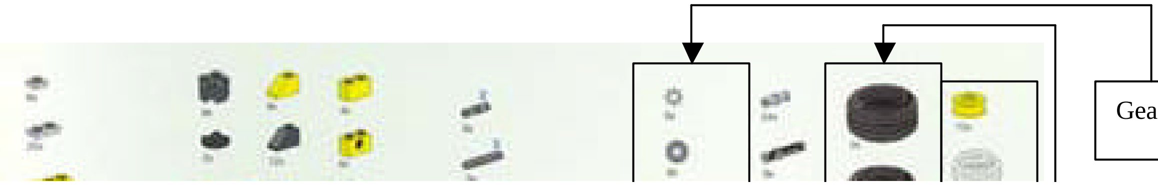
Wheels and belts- 50 cents each

Tires – 60 cents each

- For retail pricing on materials used other than in the kits-- any of the following are acceptable:

  - A receipt
  - A "tear-out" of an ad from a newspaper or magazine, clearly indicating the item price.)

Price List



Gears- 40 cents each

- For retail pricing on materials used other than in the kits-- any of the following are acceptable:
    - A receipt
    - A "tear-out" of an ad from a newspaper or magazine, clearly indicating the item price.)

Tentative Schedule:

	Mon	Wed	Fri
Week 8	**Procedure A & B** - Introduction to final project - Instruction on basic gear mechanics and sensors - Inventory - Training video	**Procedure C** - Instruction on RCX code programming - Instruction on basic gear mechanics and sensors	**Procedure C** - Program a pathfinder for simple tasks - Guest speakers
Week 9	- Career center visit - *Pathfinder Program* due	**Procedure C & D** - Continue programming exercise -Begin Brainstorm and Design	**Holiday**
Week 10	**Procedure D** - Continue Brainstorm and Design - *Design Specification* due	**Procedure E** - Continue prototype construction	**Procedure F** - Trial run
Week 11	**Procedure G** - Revision and Construction -*Performance test* due	**Procedure H** - Competition - Inventory (return robotics Invention kit) - Final Evaluation	

Deliverable and Grading Breakdown:

- Due Monday, week 9 – Construct and Program a simple pathfinder using RCX code (20%)

- Due Monday, week 10 - Design Specification (10%)

- Due Monday, week 11 (extra credit, 20%) – Performance test. Evaluation of average speed and output torque on your robot. This can be included in the your final report.

- Due Friday, week 11 - Each team will evaluate its team member's contribution (10% of your grade). The evaluation should be included in your final report.

- Due Friday, week 11 – A built, competition-ready machine. This grade will be determined by how well you implemented the intended design.

- Competition (30% of grade), Score will be determined by the ground rule equation which is available on the robotics project description.

- Due Friday, week 12- Final Design Project report (30% of grade) with proper layout of a professional document. The report should contain graphs, tables drawings and equations that will clarify the text. You also need to make sure to reference all ideas, equations, figures or quotes that have been taken from other sources.

- Extra credits for using NQC program and homemade sensors

Procedure:

A. Inventory

1. Check to see you have all the parts in the set The set should include:

   - RCX micro-computer
   - CD-ROM software
   - 727 pieces of legos including:

     2 motors

     2 touch sensors

     1 light sensor

   - User Guide
   - Constructopedia
   - Infrared transmitter

2. Record any shortages and overages on a sheet to be turn in to your instructor.

B. Preparation

Students must go through the training exercise provided by the Robotics Invention System. The intention is to familiarize students with the software and the hardware of the kit. Please follow the instructions provided on pages 14 to 33 of the User Guide.

C. Construct and program a simple pathfinder

Toward the end of the training exercise, you will be asked to construct a simple sensor guided robot. Please follow the instruction on the Mindstorns video and complete the task before moving on to the part D.

Tasks to be performed by the pathfinder

- Program robot to go forward for 2.5 second and set the power at 7.
- Program robot to go backward in 3 seconds and set the power at 1 (can your robot move?)
- Program robot to maneuver a 30 ^{o} turn
- Program robot to avoid obstacle in the left or right front direction and also in between the two tactile sensors
- Program robot to halt the robot for 5 seconds when the light intensity reach 50%.

D. Design

1) Investigation

- Define the goal of your project (top-down design)
- Learn about the materials you have to work with (bottom-up design)
- Do research about your project

  There are many Lego related websites that provide helpful tips on making lego structures, and etc. You are highly encouraged to check out the following sites for ideas.
  (many links) http://www.oreilly.com/catalog/lmstorms/resources/index.html
  (interesting sites for ideas) http://www.mi-ra-i.com/JinSato/MindStorms/index.html
  http://www.mi-ra-i.com/JinSato/MindStorms/index-e.html
  http://www.verinet.com/~dlc/botlinks.htm
  http://www.medialab.nl/Company/Crew/daan/legodiff.htm
  http://www.robotbooks.com/
  (good introduction to gear and beam construction)
  http://ldaps.ivv.nasa.gov/Curriculum/legoengineering.html
  http://www.fischermellbin.com/Marcus/Lego/Gear_Mth/gear_math.html
  http://phred.org/~alex/lego/
  (ideas for sensors)
  http://www.plazaearth.com/usr/gasperi/lego.htm#background
  http://www.umbra.demon.co.uk/legopages.html
  http://www.primenet.com/~johnkit/Projects.html
  http://www.io.com/~woodward/lego/
  ftp://ftp.eecs.umich.edu/people/johannb/pos96rep.pdf
  (resources for NQC program)
  http://www.enteract.com/~dbaum/nqc/
  http://www.cs.ruu.nl/people/markov/lego/rcxcc/
  http://students.washington.edu/method13/NQC/

These are some of the sites featuring robots programmed with NQC.

- Kennedy Space Center Crawler/Transporter Model by Mark Haye
- Tic Tac Toe Robot by Marco Beri, Giulio Ferrari, and Mario Ferrari.
- Automatic Brick Sorter by Huw Millington
- My Own Robots, including a Brick Sorter
- Ben's Lego Creations, by Ben Williamson
- Jim Studt's 8 Ball Site
- Dan Danknick brought NQC to the LEGO RoboGaldiator event at E3
- Several interesting robots including a scanner by Simen Svale Skogsrud
- Dave Astolfo's robots
- Sun used NQC in one of their Java demos.
- NQC programs that learn by Robert Munafo
- Mark Miller's MTS program for multiplexing touch sensors.
- A line following robot by Paul Crowley

Do you have some NQC Powered robots? If so, please read about the Powered by NQC link button.

- Generate different ideas for project

Vocabulary:
Top-down design is starting with a goal and then determining how you are going to meet that goal.
Bottom-up design is looking at the materials you have available to you and determining what you can do with them.

2) Invention

- Come up with design criteria - the purpose of the design
- Figure out the good and bad parts of the different ideas - which one fits the purpose the best?
- Plan the project with pictures and sketches (Design Drawings and Materials assignment)

E. Prototype Construction

Construct a prototype of a chosen design. Here designer/builder interface plays a key role in a
Tips: make sure the structure doesn't block the buttons on the RCX.

F. Performance Test (optional)

Test out the design. The working prototype should demonstrate that the design is indeed feasible using the supplies in the kit and the constraints of the competition. To fulfill this, you must complete the given performance tests and estimate the final score of your machine.

G. Implementation

Based on the results of the performance test and observation, make modifications to the design and re-test

H. Final Evaluation

Take part in Competition.
Final score calculation
Present your project in a written form (Final report)

One more thing…..
Throughout this three-week construction project, your team must also document the project in your journal. This documentation will become useful when you are writing your final report.

Components:

RCX micro-computer:

RCX is a programmable micro-computer. It is the brain of the robot. A wireless link to the RCX itself allows robot to sense and to move. RCX can be programmed using a PC via an IR transceiver. The RCX has three input ports for sensors (e.g. push button " touch" sensor or light sensor) and three output ports (e.g. for motors or lights).

IR transceiver:

Downloading the program from your PC to RCX requires a pair of special infrared (IR) transceivers (similar to your TV remotes). One IR-transceiver is connected to a serial port of the personal computer and the other built in RCX. Communication is established via transmission/reception of infrared light. IR transmitter uses a 38kHz carrier, which is 100 times the sampling rate of the RCX (2400bps).

Software:

The programming language used in the Robotics System is a simple string of icon commands. The strings visually describe the response and action of the inputs and outputs of the RCX (micro-controller). The programming language is usually referred as RCX code. As you will see when you go though the training session, RCX code is a computer program environment in which graphics are used to build a program. In RCX's code, each block displayed on the screen represents instruction. You click, grab, and link graphical blocks on the computer screen. The blocks build (stack) on one another, like pieces in a puzzle, to create a program.

Sensors:

Light sensor -Light sensor contains a red LED (light emitting diode) and a PIN photodiode. Light reflected from the environment either due to the LED or from the background is received at the photodiode. The sensor has a 0 to 100% dynamic range. A dim room is about 10% and pointing sensor to a 75 watts incandescent lamp with shade placed some 9 ft away is about 40%.

Touch Sensor - An on/off switch sensor.

Motor:

The DC motor is a device that converts electrical energy into mechanical energy. The motor provided has 8 different speeds and capable of going in 2 different directions (clockwise and counterclockwise) and pulsing.

For more information, please refer to the following Technical Notes or search any Lego Mindstorms related web sites.

Technical Notes:

1. RCX micro-computer

CPU:

Here are the specs on RCX - part number HD6433292 (Ref. Hitachi: website)

Series	H8/3297
Product name	**H8/3292**
Part number	**HD6433292**
ROM size	**16K**
RAM size	**512**
Speed	**16MHz @ 5V**
8-bit Timers	**2**
16-bit Timers	**1**
A/D Conversion	**8 8-bit**
I/O pins	**43**
Input only pins	**8**
Serial port	**1**
10mA outputs	**10**

Serial Pin Connection for the IR Transmitter:

Serial Pin	To IR Block	Name	Description
1	**1**	**-**	**Not Used**
2	**3**	**RX**	**Receive**
3	**2**	**TX**	**Transmit**

4	4	-	Not Used
5	5	SG	Signal Ground
6	6	-	Not Used
7	8	RTS	Ready To Send
8	7	CTS	Clear To Send
9	9	-	Not Used

Hardware:

Numbers for elements on the top (LCD side) of the board (power connector on the right), moving generally from left to right:

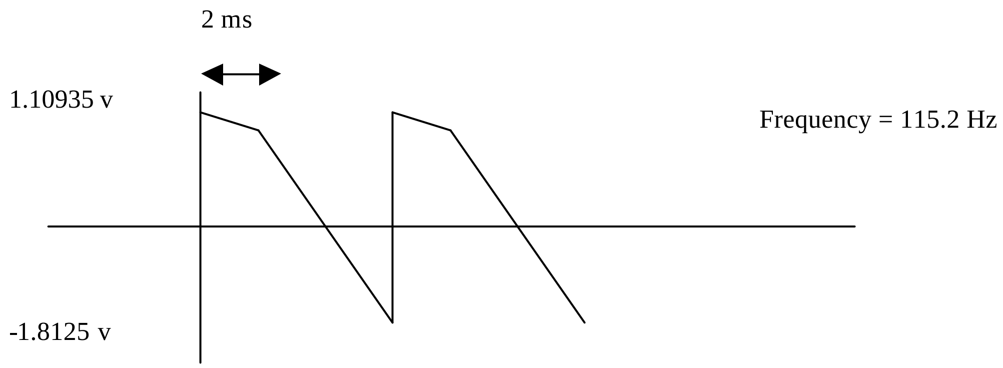
2	IR LEDs	?
1	IR receiver	TK19 / TFM. / 749 / 1380
1	small chip, below IR receiver leads, four leads	CDC / P83
8	small chips scattered on left half, three leads	3K / P / 81
3	small chips scattered on left half, three leads	1K / P / 82
1	capacitor to right of IR receiver leads	4.7uF
3	red cylinders w/ two black lines, two leads	diac
5	resistors, yellow label on black	10K
1	resistor, yellow label on black	1M
5	resistors, yellow label on black	2.2K
1	resistor, yellow label on black	22K
3	resistors, yellow label on black	3.3K
3	resistors, yellow label on black	39
1	resistor, yellow label on black	47K
3	resistors, yellow label on black	10K 1%
1	resistor (?), yellow label on black	IRD (?)
1	resistor (?), white label on green	560
3	gray boxes, two leads, scattered on left half	?
1	LCD display	?
1	speaker (under LCD)	?
1	LCD controller (under LCD)	PCF8566T / 61875 / YSM98061
1	metal can with two (?) leads	H2 / 100 / 16V
1	voltage regulator (?)	LM293
1	power connector	8 (on back) / 22 (on top)
12	scattered brown boxes, two leads	?
24	scattered diodes	SM 4002

12	**clips**	**?**

Numbers for elements on the bottom (micro-controller) side of the board (power connector on left), moving generally from left to right:

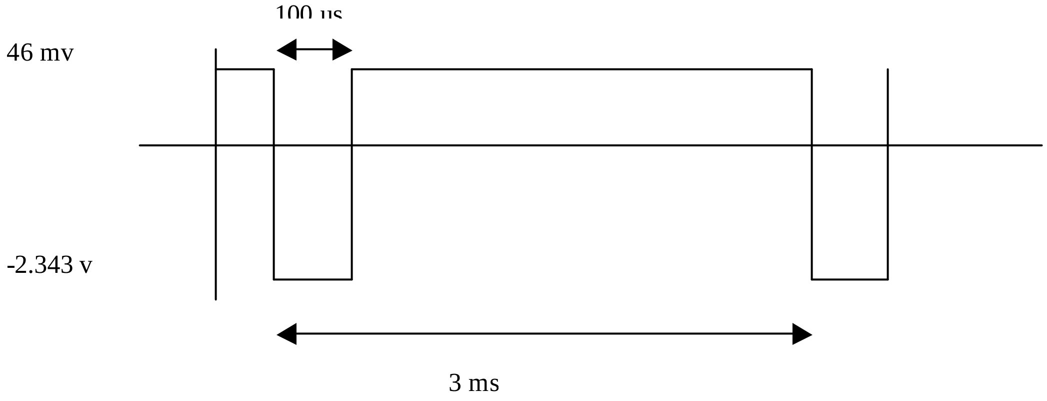
1	small chip, leftmost on board, three leads	Z5 / P / 7d
1	**MOSFET (?)**	**F3055L / H7 / 01**
1	**tantalum cap**	**10uF, 6V**
2	**small chips, right of 3055, three leads**	**3K / P / 81**
1	**gray box, two leads**	**?**
3	**motor controllers (?), 16 leads**	**ELEX 10402B / 9980A 4597**
1	**bank of three input NAND gates, 14 leads**	**74HC10D**
1	**small chip beneath 74HC10, three leads**	**Z4 / P / 7o**
1	**small chip above right of 74HC10, three leads**	**1K / P / 82**
1	**bank of flip flops, 20 leads**	**74HC377D**
1	**crystal, above microcontroller**	**16Mhz**
1	**small chip above microcontroller, three leads**	**P 005**
1	**microcontroller, 64 pin QFP**	**HD6433292B02F**
1	**RAM, 32K (?), 28 pins**	**D43256B60-70LL / 9752XD077**
1	**capacitor, upper right, two leads**	**4.7uF**
1	**small chip, right edge center, three leads**	**3K / P / 81**
1	**small chip, right edge bottom, three leads**	**1K / P / 82**
1	**resistor, yellow label on black**	**10**
1	**resistor, yellow label on black**	**100**
1	**resistor, yellow label on black**	**10K**
10	**resistors, yellow label on black**	**100K**
1	**resistor, yellow label on black**	**12K**
1	**resistor, yellow label on black**	**150K**
4	**resistors, yellow label on black**	**22K**
1	**resistor, yellow label on black**	**220K**
1	**resistor, yellow label on black**	**330**
1	**resistor, yellow label on black**	**390K**
6	**resistors, yellow label on black**	**47K**
1	**resistor, yellow label on black**	**470K**
4	**brown boxes, two leads**	**?**

Output signal from CRX when power level is set at 4



The RCX sends a higher voltage level for higher power level. The frequency remains the same.

Input to light sensor when sensor is not connected



The RCX sends a high (8 VDC) level for around 1.2ms and disconnects it for around 0.1ms which is when it reads the sensor. The 0.1ms voltage is proportional to the measurement where 0% is around 5 VDC and 100% is around 2 VDC.

2. Sensors:

(taken from http://www.plazaearth.com/usr/gasperi/lego.com)

The RCX reads Touch, Temperature and Light sensors in the pretty much the same way. Rotation is a lot like Light but will be covered later. The voltage on the input is converted to an internal RAW value in the range 0V=0 to 5V=1023. Depending on the sensor type, the RAW number is converted into the number you see in the program, Test Panel or View.

-Touch sensors: If the RAW value is less than about 450 it becomes a 1 and if the RAW value is greater than about 565 it becomes a 0.
-Temperature sensors: In degrees C, Temp=(785-RAW)/8 within the range -20C to +70C.

- Light sensors: Light=146-RAW/7 within the range 0 to 100.

Summary Input Table

Volt	Raw	Sensor Ohms	Light	Temp C	Touch
0.0	0	0	-	-	1
1.1	225	2,816	-	70.0	1
1.6	322	4,587	100	57.9	1
2.2	450	7,840	82	41.9	1
2.8	565	12,309	65	27.5	0
3.8	785	32,845	34	0.0	0
4.6	945	119,620	11	-20.0	0
5.0	1023	Inf	0	-	0

For Touch and Temperature type input, the RCX has a 10,000ohm resistor pulling up the input to 5V. The sensor only needs to provide some resistance to create a reading. You can create a fake temperature sensor by hooking a 4,700ohm resistor in series with a 50,000ohm potentiometer (both available from Radio Shack). This will read from about -11C to +60C. With Touch or Temperature sensors there is no reason why you can't overdrive the input to whatever voltage you want within the 0V to 5V range.

You should NOT try to overdrive an input that thinks it has a Light or Rotation Sensor type on it. Use one of the general purpose analog interfaces. The RCX has a 120ohm resistor pulling up to about 8V (probably a diode drop from the battery voltage) to power the red LED for about 3ms and then looks at the sensor voltage during a short 0.1ms time. During the short time the sensor is read just like the Touch or Temperature sensors. The fake Temperature sensor from above will read Light values from 100 down to about 22 without the risk of damaging the RCX since it never loads the input with less than 4,700ohms. I doubt any real damage would occur to the RCX since people could accidentally connect a Touch sensor where a Light sensor should be or even a motor output to a sensor input. I can't imagine LEGO would allow this to destroy the RCX, but don't say "I didn't warn you."