

1997

6.270 Autonomous LEGO Robot  
Design Competition  
Course Notes

The authors hereby grant to M.I.T. permission to reproduce and to distribute copies of this document in whole or in part.

The authors hereby grant to any other individuals or organizations permission to reproduce and distribute copies of this document, in whole or in part, for personal or non-commercial educational use only. This copyright notice must accompany any copy that is distributed.

Other than these cases, no part of this document may be reproduced in any form without written permission of the authors. For example, the authors do *not* grant the right to make derivative works based on this document without written consent.

The authors grant permission to future organizers and instructors of the MIT 6.270 course to revise the notes as they see fit.

To obtain additional copies of this document, write to:

6.270 Organizers  
The EECS Department  
The Massachusetts Institute of Technology  
50 Vassar Street Room  
Cambridge, MA 02139

or by sending electronic mail to `6.270-organizers@mit.edu`

Chapter 2 was revised for 1997 by Yonah Schmeidler.

Chapter 3 was written by Fred Martin and Pankaj Oberoi, and has been revised by Pankaj Oberoi and Yonah Schmeidler from previous 6.270 notes.

Chapter 4 was written by Fred Martin and Pankaj Oberoi and has been revised by Pankaj Oberoi and Anne Wright to include parts locations, functionality of parts, and to reflect changes in the hardware. The section has been modified by Matt Domsch, Karsten Ulland, and Pankaj Oberoi to reflect the modifications to the board.

Chapter 5 was written by Matt Domsch, Pankaj Oberoi, and Karsten Ulland.

Chapter 6 was revised by Anne Wright from Fred Martin's *Robot Builder's Guide* to reflect changes in the compiler and libraries.

Chapter 7 was written by Randy Sargent.

Chapter 8 was written by Sanjay Vakil and edited by Pankaj Oberoi.

Appendix A was written by Fred Martin and has been painstakingly revised by Anne Wright, Karsten Ulland, Matt Domsch, and Pankaj Oberoi to reflect changes in the hardware.

Appendix B was written by Fred Martin and revised by Matt Domsch and Pankaj Oberoi to reflect changes in the hardware.

Appendix C was written by Fred Martin, *Robot Builder's Guide*.

All other work was co-authored by the organizers of the 1992 6.270 course.

The format for the course notes is based on previous 6.270 course notes.



# Preface

The origins of this course begin with Woody Flowers in MIT's Mechanical Engineering department. Woody Flowers had the idea that teaching should be interactive and not just lecturing. He developed the famous "Introduction to Design" class (course number 2.70). In 2.70, undergraduates use scrap parts—metal, plastic, and wood—to build machines that go on to compete in a head-to-head contest at the end of the course.

Michael B. Parker, an undergraduate in MIT's Electrical Engineering and Computer Science (Course Six) department, had just taken 2.70. Mike liked the course so much that he was jealous: "Why should there be a course like this for Mechanical Engineering students, but not for the students in his department?" he thought.

So in 1987, Mike organized the first 6.270 contest as "Course Six's answer" to the 2.70 course. The contest was a programming competition in which students wrote programs to control computer-simulated robots. In the first two years of the contest, the goal was to design a simulated robot that tried to find and destroy other robots. Unlike the machines that are built in the 2.70 course, there was no human control of the simulated robots (in 2.70 the students control the machines through a joystick and some switches). This was what separated the 2.70 course and the 6.270 contest.

A couple of years later, Mike saw a project at MIT's Media Laboratory called "LEGO/Logo," in which children build robots and other mechanized devices out of LEGO bricks, motors, and electronic sensors, and then write programs to control them using a special version of the Logo programming language. Mike wanted to provide the 6.270 students with similar technology so they could build *real* robots, not just the computer-simulated robots that been done in the past.

Mike recruited Fred Martin and Randy Sargent to be the technical consultants to the upcoming 6.270 contest (which was starting in just a few weeks). It sounded like a fun way to spend IAP.<sup>1</sup>

Randy and Fred spent most of their holiday break designing an interface board that connected to a PC or Macintosh computer, controlling motors and providing input from a few simple sensors. The budget was tight and time was short as they

---

<sup>1</sup>MIT's "Independent Activities Period" is a one-month break between the fall and spring semesters.

scurried about the local Radio Shack stores, buying electronic parts for the twenty teams of students who had preregistered for the course.

Everything went wrong that month: the LEGO parts showed up late; Athena – the campus computer network – would not give approval for 6.270 to use its workstation and mix-ups in room scheduling forced groups of students wielding soldering irons to wander about campus looking for available classrooms.

The robots were powered and controlled through a tether connected to a personal computer. A lounge was transformed into a workshop, and only the students' excitement carried the contest through the month and into a competition at the end of IAP. Students who were building robots contributed their time to making the contest a reality. The contest lasted about four hours—it was a long, drawn-out affair—but the students enjoyed it.

The unique feature of 6.270 was that the students were running the course themselves. The learning was through mutual help. No professors or faculty members were involved with the organization or teaching of the course. Students learned from interaction with other students, and this formed the basis for future 6.270 contests.

After the contest was over, Pankaj (“P.K.”) Oberoi, who had been a student in the contest, called a meeting of interested past participants. P.K. thought the class had great potential as a learning project and wanted to help organize it for the upcoming year. He felt that given some organization and structure, the contest could be transformed into a course in which the students could learn more than just how to “hack” something together.

P.K. had already worked on key administrative aspects, like recruiting corporate sponsors. Microsoft had donated some money to the course previously and was willing to up the ante for another year. P.K. also got the support of Motorola, which agreed to donate valuable semiconductor parts. He also gained the support of the Course 6 faculty to allow the contest for another year, with a more structured environment.

Randy and Fred were recruited once again by P.K. to help develop technology for the course. Hesitant at first, they laid out artwork for a custom printed circuit board that used a new Motorola microprocessor with more control features than the tethered machines. The original board was based on a microprocessor board designed at the MIT AI Laboratory by Henry Minsky.

P.K. and Fred wrote handouts for the students telling them how to build sensors, a battery charger, and other robotics components. The contest was transformed from a group of students wandering around looking for a place to work into a course with lectures and labs in where the students could learn more technical aspects in addition to the hands-on exposure they were getting.

Eighty students, organized into thirty teams, took the course that year. Even before the contest that year, it was evident that the course was a success. As students carried their robot kits around campus, interest and excitement spread. The contest was a hit.

By the end of that year, Fred was interested in the course not only from a technical perspective, but from a special educational perspective. Students at MIT were choosing to pull all-nighters building robots rather than taking ski trips. In doing so, they were learning about engineering design and robotic technologies from first-hand, experiential involvement in a project-based course. The course seemed to fill a gap in the students' education.

By providing the tools and materials for students to work with complex electronic, mechanical, and software ideas, 6.270 gives the students a place to explore and learn about key ideas in technology, engineering, and design. Teamwork, learning-by-doing, and learning from one's peers is primary. 6.270 provides a hands-on style of learning experience for MIT students who are used to the theoretical lecturing style presented in most of the core classes.

The 1990 class was a big success but the organizers wanted to make the class even better. The class was hampered by a controller board which had to be programmed in assembly language and only had a small amount of memory. After the contest, work began on technology that would be more powerful and more useful to 6.270 students, allowing them to get even deeper into robot design and other technological issues.

By the start of the 1991 class, Fred and Randy had developed a robot-building kit with the high degree of power and flexibility we had wanted. Students were able to develop software for their robot using a subset of the C programming language called IC. The new embedded controller board they developed had a number of new features, including a small display screen that could be used to print debugging messages. Students were able to use powerful Unix workstations all across campus to develop programs for their robots.

The 1992 contest was organized by Fred, Randy, and P.K. joined by Karsten Ulland and Matt Domsch, two sophomores who had taken the course the previous year. New features continued to be added. Fred changed the layout of the board to include more channels of analog input for sensors; Randy improved IC to allow multiple C and binary modules to be loaded to the board; and PK, Karsten, and Matt worked staffing the labs during the day and building the contest tables at night.

The enthusiasm in the course was increasing by enormous proportions. This was evident in the increase in enrollment. Over 300 students signed up to take the course for 150 spots. Several new sponsors were recruited to participate in the course. As in previous years, many participants in the course also contributed towards helping prepare for the contest. The rapid change in technology did not allow much time for debugging or completion of the software. Anne Wright, a student in the course helped take up the slack by reporting bugs and writing software updates.

1993 organizers consisted of P.K., Matt, and Karsten joined by Anne Wright and Sanjay Vakil, two participants from the 1992 contest. They worked to update the

technology and notes for the course and to organize the course for a new group of approximately 170 students divided into 55 teams. With Fred and Randy no longer active organizers, they set about to establish a new division of labor which would allow the standards of progress set by the previous contests to continue.

Matt took responsibility of class registration; P.K., Karsten, and Sanjay contacted greater support from the sponsors and ordered the kits; Karsten and Anne made changes to the controller board to fix hardware “bugs” from the previous year and some additional features (a servo); and Anne rewrote portions of the IC compiler to give it more of the features of real C. These course notes have come about from a combination of past experiences and each organizer took on the section of the notes with which they felt most comfortable.

In 1993, over 125 teams signed up to take the course. Because of the popularity and demand, the lab was kept open most evenings. To handle the demands of the course, eight TAs were hired from 6.270 alumni. Tripling the staff helped to offload the organizers. In addition, Motorola sent a film crew to the contest to document the contest from the distribution of the kits to the final round on contest night.

1994’s contest expanded once more: 10 TAs and five organizers helped to create the contest. Of the 265 registrants came fifty teams. A new weighting system was also used to try and give past registrants who were not accepted a better chance at getting into the course. Given that the boards changed minimally, the extra time was spent updating and clarifying the course notes.

Unfortunately, due to budget and staff constraints, participation was back to forty teams (about 110 students) in 1996. If you’re taking the class now and would like to see it grow again, speak to an organizer about helping organize next year’s contest.

Over the years, the organizers of the 6.270 contest have put a great deal of thought into how to organize the class to maximize the students learning potential and enjoyment of the course. We have tried at all times to provide the best educational and technical environment we could. We have tried to schedule events such that teams have a maximum amount of time to experiment with design and programming issues with a functioning robot. But the thing which really sets 6.270 apart from all the other technical and lab courses at MIT is that despite the enormity, this project remains to be one of the largest student-run activities due to the enthusiasm of the students who have taken the course. We believe that this is 6.270’s greatest strength, and we hope that this enthusiasm will continue to attract generations of student organizers to keep 6.270 alive.

The 6.270 Organizers  
January 2, 1997



# MIT Departments

Two departments have played a major role in the development and running of the 6.270 project. We are grateful to both departments who have helped us, and for the encouragement given to us by many individuals in those departments.

## ELECTRICAL ENGINEERING AND COMPUTER SCIENCE (EECS) DEPARTMENT

The EECS department is the main supporter for the course and provides the lecture and laboratory space to run the course for 120 students. The EECS department provides for lab technicians to ensure that the labs are open during late hours. The department also funds some of the organizers. In addition, the EECS department provides funding in excess of \$16,000 so that students are able to keep their kits when the course is over. The department offers this course for 6 units of credit to students wishing to get credit for the course.

## MIT MEDIA LABORATORY

The Media Lab has been a past supporter by assisting in providing development resources for the 6.270 technology from 1990 to 1992. The Media Lab has sponsored Fred Martin's work on the development of the course and his research on the learning process of the students in the course.



# Sponsors

Many companies both large and small have contributed generously to the 6.270 project. We are grateful to both the corporations who have supported us, and for the encouragement given to us by many individuals who have worked for them.

MICROSOFT CORP.

MOTOROLA SEMICONDUCTOR INC.

LEGO SYSTEMS INC.

HAWKER ENERGY PRODUCTS INC.

POLAROID INC.

MIT EECS DEPARTMENT



# Introduction to this Guide

These course notes contain a lot of information that should assist you in developing a working robot by the end of the course. The notes have been arranged in the order in which you should proceed. Many of the topics can be done simultaneously by different members of the team so as to multitask. We suggest that you follow a strict approach through these notes and that you read before you make a mistake.

- Chapter 1 is an introduction to the course. It contains all the administrative information you need.
- Chapter 2 contains the rules and guidelines for this year's contest.
- Chapter 3 explains good assembly technique and how to determine different component types.
- Chapter 4 is the assembly manual for nearly all of the components used to construct a 6.270 robot—including the printed circuit boards, electronic sensors, motors, and battery packs.
- Chapter 5 delves into various robotic sensors, explaining the principles of operation and applications of various sensors in the 6.270 kit. This section also include assembly instructions for the sensors.
- Chapter 6 is a reference manual for the C language software that has been developed for the 6.270 contest.
- Chapter 7 is a chapter which will help you get started with designing your LEGO robot. It goes into to intricacies of building strong LEGO machines.
- Chapter 8 investigates how to program a mobile robot to face up to the uncertainties and challenges of practical operation.

Three appendix sections provide additional material:

- Appendix A explains the workings of the 6.270 hardware, including the micro-processor board, the expansion board, and the infrared circuitry. This section is written with the assumption of some prior background in electronics.
- Appendix B contains copies of the printed circuit board artwork patterns for the 6.270 Rev. 2.21 boards.
- Appendix C discusses battery technology and battery charger operation.

# Contents

<b>1</b>	<b>Introduction to 6.270</b>	<b>1</b>
1.1	Registration Policy . . . . .	1
1.2	Kit Fee and Toolkit Fee . . . . .	2
1.3	Credit Guidelines . . . . .	3
1.3.1	Credit Guidelines . . . . .	3
1.3.2	Design Units . . . . .	4
1.4	Schedule . . . . .	5
1.4.1	Important Dates . . . . .	7
1.4.2	Progress Schedule . . . . .	8
1.5	Computer Facilities . . . . .	14
1.5.1	6th Floor Laboratory . . . . .	14
1.5.2	Athena Clusters . . . . .	14
1.5.3	Athena Etiquette . . . . .	14
<b>2</b>	<b>RoboRats</b>	<b>17</b>
2.1	The Table . . . . .	17
2.2	Scoring . . . . .	19
2.3	Period of Play . . . . .	20
2.4	The Competition . . . . .	21
2.5	Infrared Beacon and Light Sources . . . . .	22
2.6	Structure . . . . .	23
2.6.1	The \$10 Electronics Rule . . . . .	24
2.7	Organizers . . . . .	25
<b>3</b>	<b>Electronic Assembly Technique</b>	<b>27</b>
3.1	Electronic Assembly Technique . . . . .	27
3.1.1	Soldering Technique . . . . .	27
3.1.2	Desoldering Technique . . . . .	30
3.1.3	Component Mounting . . . . .	30
3.1.4	Component Types, Polarity, and Markings . . . . .	31

<b>4</b>	<b>Assembly Manual</b>	<b>39</b>
4.1	The Battery System . . . . .	40
4.1.1	Battery Pack Construction . . . . .	41
4.1.2	The Battery Charger . . . . .	43
4.2	The Motor Switch Board . . . . .	46
4.2.1	Assembly Instructions . . . . .	47
4.3	The Expansion Board . . . . .	48
4.3.1	Assembling the Expansion Board . . . . .	49
4.3.2	Testing the Expansion Board . . . . .	54
4.4	The LCD Display . . . . .	56
4.5	The Microprocessor Board . . . . .	56
4.5.1	Assembling the Microprocessor Board . . . . .	56
4.5.2	Testing the Microprocessor Board . . . . .	66
4.5.3	Board Checkoff . . . . .	67
4.5.4	After Board Checkout . . . . .	70
4.6	The Infrared Transmitter . . . . .	70
4.6.1	Assembly Instructions . . . . .	70
4.7	Cable and Connector Wiring . . . . .	72
4.8	Motor Wiring . . . . .	78
4.8.1	The Polaroid Motor . . . . .	78
4.8.2	Servo Motor . . . . .	82
<b>5</b>	<b>Sensor Design</b>	<b>83</b>
5.1	Sensor Assembly . . . . .	84
5.2	Analog vs. Digital Sensors . . . . .	84
5.3	Location of Digital and Analog Ports . . . . .	85
5.4	Digital Sensors . . . . .	85
5.4.1	Dip Switches . . . . .	86
5.4.2	Micro-Switches . . . . .	86
5.4.3	Sharp IR Detector . . . . .	88
5.5	Analog Sensors . . . . .	89
5.5.1	Resistive Sensors . . . . .	90
5.5.2	Transistive Analog Sensor . . . . .	90
5.5.3	Potentiometers . . . . .	91
5.5.4	Photoresistors . . . . .	93
5.5.5	Photo Transistors and IR LEDs . . . . .	94
5.5.6	Polarizing Film . . . . .	96
5.5.7	Reflectance Sensors . . . . .	98
5.5.8	Motor-Force Sensors . . . . .	98
5.5.9	Breakbeam Sensors . . . . .	98



<b>6</b>	<b>IC Manual</b>	<b>105</b>
6.1	Getting Started . . . . .	106
6.2	Using IC . . . . .	107
6.2.1	IC Commands . . . . .	108
6.2.2	Line Editing . . . . .	108
6.2.3	The <code>main()</code> Function . . . . .	109
6.3	A Quick C Tutorial . . . . .	109
6.4	Variables, Constants, and Data Types . . . . .	111
6.4.1	Variables . . . . .	111
6.4.2	Constants . . . . .	113
6.4.3	Data Types . . . . .	114
6.4.4	Pointers . . . . .	115
6.4.5	Arrays . . . . .	118
6.4.6	Structures . . . . .	121
6.4.7	Complex Initialization examples . . . . .	123
6.5	Operators, Expressions, and Statements . . . . .	124
6.5.1	Operators . . . . .	124
6.5.2	Assignment Operators and Expressions . . . . .	125
6.5.3	Increment and Decrement Operators . . . . .	126
6.5.4	Data Access Operators . . . . .	126
6.5.5	Precedence and Order of Evaluation . . . . .	127
6.6	Control Flow . . . . .	127
6.6.1	Statements and Blocks . . . . .	127
6.6.2	If-Else . . . . .	127
6.6.3	While . . . . .	128
6.6.4	For . . . . .	128
6.6.5	Break . . . . .	129
6.7	LCD Screen Printing . . . . .	129
6.7.1	Printing Examples . . . . .	129
6.7.2	Formatting Command Summary . . . . .	130
6.7.3	Special Notes . . . . .	130
6.8	Preprocessor . . . . .	130
6.8.1	Preprocessor Macros . . . . .	131
6.8.2	Conditional compilation . . . . .	132
6.8.3	Comparison with regular C preprocessors . . . . .	132
6.9	The IC Library File . . . . .	133
6.9.1	Output Control . . . . .	134
6.9.2	Sensor Input . . . . .	137
6.9.3	Analog Inputs . . . . .	138
6.9.4	Time Commands . . . . .	141
6.9.5	Tone Functions . . . . .	142

6.9.6	Menuing and Diagnostics Functions . . . . .	143
6.10	Multi-Tasking . . . . .	145
6.10.1	Overview . . . . .	145
6.10.2	Creating New Processes . . . . .	145
6.10.3	Destroying Processes . . . . .	146
6.10.4	Process Management Commands . . . . .	147
6.10.5	Process Management Library Functions . . . . .	147
6.11	Floating Point Functions . . . . .	148
6.12	Memory Access Functions . . . . .	148
6.13	Error Handling . . . . .	149
6.13.1	Compile-Time Errors . . . . .	149
6.13.2	Run-Time Errors . . . . .	150
6.14	Binary Programs . . . . .	150
6.14.1	The Binary Source File . . . . .	151
6.14.2	Interrupt-Driven Binary Programs . . . . .	153
6.14.3	The Binary Object File . . . . .	157
6.14.4	Loading an <code>icb</code> File . . . . .	158
6.14.5	Passing Array Pointers to a Binary Program . . . . .	158
6.15	IC File Formats and Management . . . . .	158
6.15.1	C Programs . . . . .	158
6.15.2	List Files . . . . .	159
6.15.3	File and Function Management . . . . .	159
6.16	Configuring IC . . . . .	160
<b>7</b>	<b>LEGO Design</b>	<b>161</b>
7.1	The Magic LEGO Dimension . . . . .	162
7.2	LEGO Gears . . . . .	162
7.2.1	It's all the motor's fault . . . . .	164
7.2.2	Pointers for an efficient geartrain . . . . .	164
7.2.3	How to know if your geartrain is really good . . . . .	166
7.2.4	You don't always need an efficient geartrain! . . . . .	166
7.3	Making extremely strong structures . . . . .	167
7.3.1	How to know if your structure is extremely strong . . . . .	167
7.4	LEGO Challenges . . . . .	167
<b>8</b>	<b>Robot Control</b>	<b>169</b>
8.1	Types of Control . . . . .	169
8.2	Open Loop Control . . . . .	169
8.3	Closed Loop Control . . . . .	171
8.4	Closed Loop Control with Coarser Sensors . . . . .	175
8.4.1	Analog Sensor Problems . . . . .	175

8.5	Feed Forward Control . . . . .	177
8.6	Sensor Integration . . . . .	178
<b>A</b>	<b>6.270 Hardware</b>	<b>179</b>
A.1	The Microprocessor and Memory . . . . .	179
A.1.1	Multiplexing Data and Address Signals . . . . .	181
A.2	Memory Mapping . . . . .	182
A.2.1	Memory-Mapping the RAM . . . . .	184
A.2.2	Memory-Mapping with the 74HC138 Chip . . . . .	185
A.2.3	System Memory Map . . . . .	187
A.2.4	Digital Inputs . . . . .	187
A.2.5	Digital Outputs . . . . .	190
A.2.6	6811 and Memory Schematic . . . . .	190
A.3	The Motor Drivers . . . . .	190
A.3.1	The H-Bridge Circuit . . . . .	190
A.3.2	The H-Bridge with Enable Circuitry . . . . .	192
A.3.3	The SGS-Thomson Motor Driver Chip . . . . .	193
A.3.4	Power Considerations . . . . .	196
A.3.5	Expansion Board Motor and LED Circuitry . . . . .	198
A.4	Analog Inputs . . . . .	198
A.4.1	Motor Current Monitoring Circuit . . . . .	199
A.4.2	Analog Input Multiplexing on the Expansion Board . . . . .	200
A.5	The Serial Line Circuit . . . . .	201
A.5.1	Serial Output . . . . .	202
A.5.2	Serial Input . . . . .	203
A.6	Battery-Backing the Static RAM . . . . .	203
A.6.1	Powering the Memory Chip . . . . .	204
A.6.2	The Power-Off Interrupt . . . . .	204
A.6.3	The Power-Up Delays . . . . .	205
A.7	The Infrared Transmission Circuit . . . . .	206
A.7.1	The IR Beacon . . . . .	209
A.8	The LCD Display . . . . .	209
A.9	The Low-Battery Indicator . . . . .	211
A.10	Fun Hacks . . . . .	212
A.10.1	Adding a Loudspeaker . . . . .	212
<b>B</b>	<b>Printed Circuit Layouts</b>	<b>215</b>
B.1	Microprocessor Board . . . . .	216
B.2	Expansion Board . . . . .	217
B.3	Battery Charger Board . . . . .	218
B.4	Motor Switch Board . . . . .	218

B.5	Infrared Beacon Board . . . . .	219
<b>C</b>	<b>Batteries</b>	<b>221</b>
C.1	Cell Characteristics . . . . .	221
C.1.1	Voltage . . . . .	221
C.1.2	Capacity . . . . .	221
C.1.3	Power Density . . . . .	222
C.1.4	Discharge Curve . . . . .	222
C.1.5	Internal Resistance . . . . .	222
C.1.6	Rechargeability . . . . .	223
C.1.7	Cost . . . . .	223
C.2	Battery Packs . . . . .	224
C.3	6.270 Battery Charger . . . . .	224

# List of Figures

2.1	The contest table for RoboRats . . . . .	18
2.2	A food cube (holes not to scale) . . . . .	19
3.1	Proper Soldering Technique . . . . .	28
3.2	Improper Soldering Technique . . . . .	28
3.3	Flat Component Mounting . . . . .	31
3.4	Upright Component Mounting . . . . .	31
3.5	Resistor Color Code Table . . . . .	33
3.6	Resistor Pack Internal Wiring . . . . .	33
3.7	Typical Diode Package . . . . .	34
3.8	Identifying LED Leads . . . . .	34
3.9	Top View of 14-pin DIP . . . . .	35
3.10	Top View of 52-pin PLCC . . . . .	35
3.11	Summary of Polarization Effects . . . . .	38
4.1	Three Battery Pack Configurations . . . . .	41
4.2	Battery Plug and Cable Wiring Diagram . . . . .	42
4.3	Battery Pack Wiring Diagram . . . . .	43
4.4	Battery Charger Component Placement . . . . .	45
4.5	Motor Switch Board Component Placement . . . . .	47
4.6	Expansion Board Component Placement . . . . .	49
4.7	Mounting Method for Male Header Pins . . . . .	52
4.8	Expansion Board Male Header Pin Placement . . . . .	53
4.9	Expansion Board Female Header Mounting . . . . .	54
4.10	Heat Sink on Motor Chip . . . . .	55
4.11	LCD Connector Mounting . . . . .	56
4.12	6.270 Microprocessor Board Component Placement . . . . .	57
4.13	6.270 Microprocessor Board Header Placement . . . . .	62
4.14	Motor Chip Stacking Technique . . . . .	65
4.15	Infrared Transmitter Component Placement . . . . .	71
4.16	Standard Connector Plug Configurations . . . . .	73
4.17	Step One of Connector Wiring . . . . .	74

4.18	Step Two of Connector Wiring . . . . .	75
4.19	Step Three of Connector Wiring . . . . .	76
4.20	Step Four of Connector Wiring . . . . .	77
4.21	LEGO Jig for Mounting Polaroid Motor . . . . .	80
4.22	LEGO Jig for Mounting Polaroid Motor . . . . .	81
4.23	Servo Motor and Integral Connector Plug . . . . .	82
5.1	Generic Digital Sensor Schematics. . . . .	86
5.2	Microswitch Assemblies . . . . .	87
5.3	Sharp IR sensor assembly . . . . .	89
5.4	Analog Sensors Schematics . . . . .	90
5.5	Potentiometer Assemblies . . . . .	92
5.6	Photocell Light Sensor . . . . .	93
5.7	The LEDs . . . . .	95
5.8	Phototransistor body and connector . . . . .	97
5.9	Reflectance Sensor . . . . .	99
5.10	Breakbeam Sensor using discrete components. . . . .	100
5.11	Breakbeam Assembly . . . . .	101
5.12	Shaft encoding using a LEGO pulley Wheel . . . . .	103
6.1	IC Command-Line Keystroke Mappings . . . . .	109
6.2	Sample IC Binary Source File: <code>testicb.asm</code> . . . . .	152
6.3	Interrupt Structure Before User Program Installation . . . . .	154
6.4	Interrupt Structure After User Program Installation . . . . .	155
6.5	<code>sysibeep.asm</code> : Binary Program that Installs into System Interrupt . . . . .	156
6.6	Sample IC Binary Object File: <code>testicb.icb</code> . . . . .	157
7.1	LEGO Spacing . . . . .	162
7.2	Magic LEGO Dimensions . . . . .	163
7.3	Gear Sizing . . . . .	165
7.4	Bracing for Success . . . . .	167
8.1	Open loop control diagram . . . . .	170
8.2	Closed loop control diagram . . . . .	172
8.3	Robot with a Bend Sensor . . . . .	172
8.4	Robot moving . . . . .	173
8.5	Jammed robot . . . . .	174
8.6	Robot with two bend sensors . . . . .	174
8.7	Sensor Problems that can arise. . . . .	176
A.1	Block Diagram of Microprocessor and Memory . . . . .	180
A.2	Block Diagram of Microprocessor and Memory with Latch . . . . .	181

A.3	6.270 System Block Diagram . . . . .	183
A.4	Enabling the Memory . . . . .	184
A.5	Wiring the 'HC138 Address Decoder . . . . .	185
A.6	6811 System Memory Map . . . . .	188
A.7	Digital Input Circuit . . . . .	189
A.8	6811, Memory, Address Decoding and Miscellaneous Circuitry . . . . .	191
A.9	The H-Bridge Circuit . . . . .	192
A.10	The H-Bridge with Left-to-Right Current Flow . . . . .	193
A.11	The H-Bridge with Enable Circuitry . . . . .	194
A.12	The SGS-Thomson L293 Motor Driver IC . . . . .	195
A.13	Motor Driver Circuit . . . . .	197
A.14	Power Filtering and Switching Circuit . . . . .	198
A.15	Expansion Board Motor and LED Circuitry . . . . .	199
A.16	Expansion Board Analog Input Circuitry . . . . .	200
A.17	Host and Board Communications over 3-Wire Serial Link . . . . .	201
A.18	Serial Line Circuit . . . . .	202
A.19	Reset Circuitry . . . . .	205
A.20	Square Wave Consisting of Bursts of 40 Khz Signals . . . . .	206
A.21	Sharp IR Sensor Decoding IR-Encoded Square Wave . . . . .	207
A.22	Block Diagram of Infrared Circuitry . . . . .	207
A.23	Infrared Transmission Circuit . . . . .	208
A.24	Infrared Beacon Circuit . . . . .	209
A.25	Low Battery Indicator Circuit . . . . .	211
A.26	Speaker Hack, v1.0 . . . . .	212
A.27	Speaker Hack, v2.0 . . . . .	213
B.1	Microprocessor Board, Component Side . . . . .	216
B.2	Microprocessor Board, Solder Side . . . . .	216
B.3	Expansion Board, Component Side . . . . .	217
B.4	Expansion Board, Solder Side . . . . .	217
B.5	Battery Charger Board, Component and Solder Sides . . . . .	218
B.6	Motor Switch Board, Component and Solder Sides . . . . .	218
B.7	Infrared Beacon Board, Component and Solder Sides . . . . .	219
C.1	Table of Cell Characteristics . . . . .	224
C.2	Battery Charger Schematic Diagram . . . . .	224

