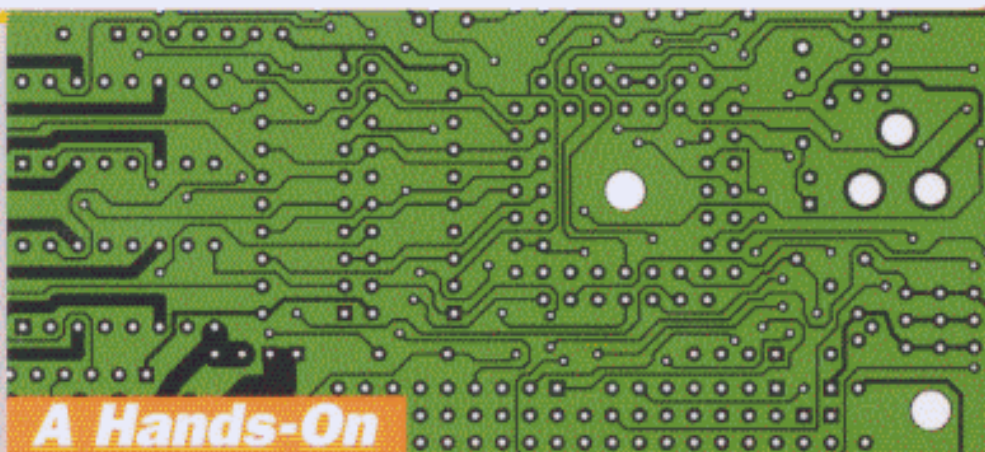
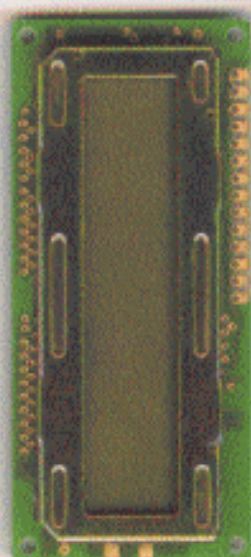
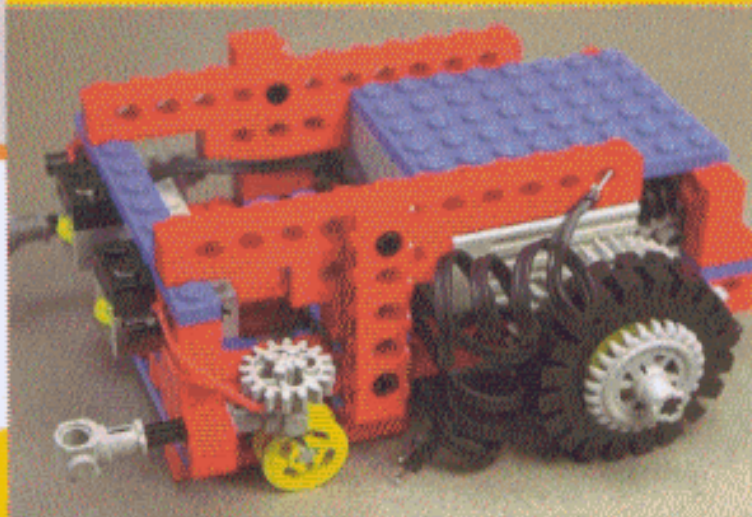


Robotic Explorations



***A Hands-On
Introduction to Engineering***



Fred G. Martin

Contents

Acknowledgments ix

1 *Introduction 1*

- 1.1 Feedback Control, Cybernetics, and Robotics **1**
- 1.2 Toys to Think With **4**
 - 1.2.1 LEGO/Logo 9
 - 1.2.2 The Programmable Brick 10
 - 1.2.3 The MIT 6.270 Robot Design Competition 11
- 1.3 About the Technology **12**
 - 1.3.1 The Handy Board 12
 - 1.3.2 Interactive C 15
 - 1.3.3 LEGO® Technic® 16
- 1.4 An Overview of the Book **17**

2 *A First Robot 20*

- 2.1 Interactive C and the Handy Board **20**
 - 2.1.1 Interactive C Prompt 21
 - 2.1.2 Motors 22
 - 2.1.3 Sensors 23
 - 2.1.4 Files and Functions 25
 - 2.1.5 The Main Function 26
- 2.2 The HandyBug **28**
 - 2.2.1 HandyBug 9645 28
 - 2.2.2 HandyBug 9719 50
- 2.3 A First Program **71**
 - 2.3.1 Plugging in Motors and Sensors 72
 - 2.3.2 Obstacle Avoidance 72
 - 2.3.3 Obstacle-Avoidance Exercises 73
 - 2.3.4 Turtle Movements 74
 - 2.3.5 Turtle Movement Exercises 74
- 2.4 Braitenberg Vehicles **75**
 - 2.4.1 One Motor and One Sensor 76
 - 2.4.2 Two Motors and Two Sensors 76

- 2.4.3 Multisensor Vehicles 78
- 2.4.4 Light Sensors 80
- 2.4.5 Coding Braitenberg Vehicles 80
- 2.5 Light and Touch Sensitivity **86**
 - 2.5.1 Light and Touch Exercises 87
- 2.6 Randomness **87**
 - 2.6.1 Random Exercises 89
- 2.7 Emergence and Meta-Sensing **89**
 - 2.7.1 Emergence 90
 - 2.7.2 Meta-Sensing 91
 - 2.7.3 Emergence and Meta-Sensing Exercises 92
- 2.8 Conclusion **94**

3 *Sensors 96*

- 3.1 Sensor Interfacing **96**
 - 3.1.1 Digital Inputs 96
 - 3.1.2 Analog Inputs 98
 - 3.1.3 Ohm's Law 99
 - 3.1.4 Ohm's Law Exercise 101
- 3.2 Building Sensors **101**
 - 3.2.1 Connector Wiring 101
 - 3.2.2 Sensor Mounting 103
- 3.3 Switch Sensors **104**
 - 3.3.1 Switch Sensor Construction 105
 - 3.3.2 Switch Sensor Applications Examples 107
- 3.4 Light Sensor Circuits **108**
 - 3.4.1 The Single-Photocell Circuit 108
 - 3.4.2 The Differential Photocell Sensor 111
 - 3.4.3 Polarized Light Seeking 114
- 3.5 Resistive Position Sensors **115**
 - 3.5.1 Bend Sensors 115
 - 3.5.2 Potentiometers 116
- 3.6 Reflective Optosensors **118**
 - 3.6.1 Applications 120
 - 3.6.2 Interfacing 121

- 3.6.3 Building It 123
- 3.6.4 Correcting for Ambient Light 125
- 3.6.5 Exercises and Projects 126
- 3.7 Break-Beam Sensors **127**
 - 3.7.1 Interfacing 129
 - 3.7.2 Building the Break-Beam Sensor 129
 - 3.7.3 Object Detection 130
 - 3.7.4 Break-Beam Exercises 130
- 3.8 Shaft Encoding **130**
 - 3.8.1 Counting Encoder Clicks 131
 - 3.8.2 Driver Software 134
 - 3.8.3 Measuring Velocity 136

4 Motors, Gears, and Mechanism 139

- 4.1 DC Motors **139**
 - 4.1.1 Exercises 140
- 4.2 Gearing **143**
 - 4.2.1 Meshing Gears 145
 - 4.2.2 Exercise 146
- 4.3 Electronic Control **146**
 - 4.3.1 The H-Bridge Motor Driver Circuit 146
 - 4.3.2 Enable and Direction Logic 147
 - 4.3.3 Active Braking 147
 - 4.3.4 Speed Control 149
 - 4.3.5 Handy Board Implementation 150
 - 4.3.6 Exercises 151
- 4.4 The Servo Motor **152**
 - 4.4.1 The Servo Control Signal 152
 - 4.4.2 Generating the Control Waveform 155
 - 4.4.3 Continuous Rotation 156
 - 4.4.4 Exercises 157
- 4.5 LEGO Design **157**
 - 4.5.1 Structure 157
 - 4.5.2 Gearing 159
 - 4.5.3 Mechanism 167
 - 4.5.4 Exercises 172

5 Control 174

- 5.1 Simple Feedback Control **174**
 - 5.1.1 Wall Following 175
 - 5.1.2 Gentle Turning Algorithm 179
 - 5.1.3 Exercises 179
- 5.2 Proportional-Derivative Control **179**
 - 5.2.1 Proportional Control 182
 - 5.2.2 The Derivative Term 188
 - 5.2.3 Exercises 189

- 5.3 Sequential Control **190**
 - 5.3.1 The Robo-Pong Contest 190
 - 5.3.2 Sequential Strategies 194
 - 5.3.3 Exercise: Groucho's Program 198
 - 5.3.4 Exit Conditions 199
 - 5.3.5 Exercise: Groucho with Timeouts 206
- 5.4 Reactive Control **206**
 - 5.4.1 A Priority-Based Control Program 208
 - 5.4.2 How the Prioritization Algorithm Works 214
 - 5.4.3 Using Reactive Control 219
 - 5.4.4 Exercises: Reactive Control 221
- 5.5 Conclusion **223**

6 Advanced Sensing 224

- 6.1 Quadrature Shaft Encoding **224**
 - 6.1.1 Construction Notes 226
 - 6.1.2 Driver Code 228
 - 6.1.3 Using the Driver 232
 - 6.1.4 Quadrature Encoder Exercises 232
- 6.2 Infrared Sensing **233**
 - 6.2.1 Modulation and Demodulation 234
 - 6.2.2 Proximity Sensing 237
 - 6.2.3 Using Proximity Sensors 238
 - 6.2.4 IR LED Exercises 239
 - 6.2.5 Advanced IR Proximity Sensing 240
 - 6.2.6 Infrared Communications 251
- 6.3 Ultrasonic Distance Sensing with the Polaroid 6500 **271**
 - 6.3.1 Connection Diagram 275
 - 6.3.2 Basic Driver Code 276
 - 6.3.3 Driver with Close-Up Capability 278
 - 6.3.4 A Test Routine 279
 - 6.3.5 Converting Time Measurements to Actual Distance 279
 - 6.3.6 Sonar Exercises 279
- 6.4 Optical Distance Sensing with the Sharp GP2D02 **280**
 - 6.4.1 Connection Diagram 281
 - 6.4.2 Construction Notes 281
 - 6.4.3 Sensor Communications Timing 282
 - 6.4.4 Driver Code 283
 - 6.4.5 Optical Ranging Exercises 285
- 6.5 Sensor Data Processing **286**
 - 6.5.1 Line Following as a Reference Activity 286
 - 6.5.2 Fixed Thresholding 287
 - 6.5.3 Parameterized Fixed Thresholding 287
 - 6.5.4 Thresholding with Hysteresis 288

- 6.5.5 Calibration by Demonstration 289
- 6.5.6 Persistent Calibration 291
- 6.5.7 Sensor Histories 291

A *Inside the Handy Board Design* 308

- A.1 Introduction to Microprocessors and the 68HC11 308
 - A.1.1 Bits, Bytes, and Characters 308
 - A.1.2 Memory Map 310
 - A.1.3 Registers 312
 - A.1.4 Evaluation Sequence 313
 - A.1.5 Machine Code Versus Assembly Language 313
 - A.1.6 Addressing Modes 314
 - A.1.7 Data Types 316
 - A.1.8 Arithmetic Operations 316
 - A.1.9 Signed and Unsigned Binary Numbers 317
 - A.1.10 Condition Code Register and Conditional Branching 318
 - A.1.11 Stack Pointer and Subroutine Calls 319
 - A.1.12 Interrupts and Interrupt Routines 320
- A.2 Introduction to Assembly Language Programming 321
 - A.2.1 Development Cycle 322
 - A.2.2 Write the Source Code 322
 - A.2.3 Assemble the Source Code 324
 - A.2.4 Download the Object File 326
 - A.2.5 Run the Program 327
 - A.2.6 Delay Loops and Subroutines 327
 - A.2.7 Digital Sensors and the User Buttons 329
- A.3 The 68HC11 with the Handy Board Hardware 330
 - A.3.1 Architecture of the 68HC11 331
 - A.3.2 Microprocessor and Memory 332
 - A.3.3 Peripherals 336
 - A.3.4 Analog Inputs 346
 - A.3.5 The Serial Line Circuit 347
 - A.3.6 The Infrared Transmission Circuit 353
 - A.3.7 The LCD Display 355
 - A.3.8 Piezo Beeper and Interrupt Routines 360
 - A.3.9 Power and Battery Charging 368
- A.4 The AS11 Assembler 372
 - A.4.1 Labels 372
 - A.4.2 Arithmetic Expressions 372
 - A.4.3 Assembler Pseudo-Operations 372
 - A.4.4 Comments 373
- A.5 Differences Between A and E Series 68HC11 CPUs 375

B *Construction Techniques* 376

- B.1 DC Motor 376
 - B.1.1 Motor Compatibility 376
 - B.1.2 Motor Wiring 377
 - B.1.3 Interfacing Hobby Motors to the LEGO Technic System 377
- B.2 Servo Motors 379
 - B.2.1 Electrical Interface 379
 - B.2.2 Continuous Rotation 379
- B.3 Infrared Sensor and Transmitter 381
- B.4 LEGO Adapter Cables 382

C *Serial Communications and Data Collection* 384

- C.1 Serial Line Interaction 384
- C.2 Connecting to a Terminal Program 386
- C.3 Printing to the Serial Line 387
 - C.3.1 Exercises 389
- C.4 Capturing Data 389
 - C.4.1 Exercises 389

D *Handy Board Specification* 391

- D.1 Specifications 391
- D.2 Ports and Connectors 392
- D.3 Battery Maintenance 394
 - D.3.1 Battery Charging 394
 - D.3.2 Adapter Specifications 394
- D.4 Part Listing 395
- D.5 Printed Circuit Board Layouts 396
 - D.5.1 Handy Board Component Side 396
 - D.5.2 Handy Board Solder Side 397
 - D.5.3 Handy Board Silkscreen 398
 - D.5.4 Interface/Charger Board Component Side 399
 - D.5.5 Interface/Charger Board Solder Side 399
 - D.5.6 Interface/Charger Board Silkscreen 400

E *Interactive C Reference* 401

- E.1 Quick Start 401
- E.2 6811 Downloaders 401
 - E.2.1 Overview 401
 - E.2.2 Putting the Handy Board into Bootstrap Download Mode 402

E.2.3	MS-DOS	402
E.2.4	Windows 3.1 and Windows 95	402
E.2.5	Macintosh	402
E.2.6	Unix	403
E.3	Interactive C	403
E.3.1	Using IC	404
E.3.2	A Quick C Tutorial	405
E.3.3	Data Types, Operations, and Expressions	407
E.3.4	Control Flow	410
E.3.5	LCD Screen Printing	411
E.3.6	Arrays and Pointers	413
E.3.7	Library Functions	415
E.3.8	Multitasking	419
E.3.9	Floating Point Functions	421
E.3.10	Memory Access Functions	422
E.3.11	Error Handling	422
E.3.12	Binary Programs	423
E.3.13	IC File Formats and Management	428
E.3.14	Configuring IC	429

F *Robot Contests 430*

F.1	Introduction	430
F.1.1	Why Robot Contests?	430
F.1.2	Why Not Contests?	430
F.1.3	Types of Contests	431
F.1.4	Social Message	432

F.2	Sample Contests	432
F.2.1	Kick the Can	432
F.2.2	Robo-Golf	434
F.2.3	Robo-Pong	436
F.2.4	Egg Hunt	438
F.3	Contest Rule Analysis	441
F.3.1	Materials	441
F.3.2	Design	442
F.3.3	Robot Size	443
F.3.4	Multiple-Robot Entries	443
F.3.5	Creativity Rule	444
F.3.6	Starting the Round	444
F.3.7	Robot Beacons	444
F.3.8	Playing Field Features	445
F.3.9	Practical Concerns	446
F.4	Conclusion	446

G *Resources 447*

References 449

Index 451