

Program Documentation

Chimera 3.2

**The Real-Time Operating System
for Reconfigurable Sensor-Based Control Systems**

February 8, 1995

Darin Ingimarson
David B. Stewart
Pradeep K. Khosla

The Robotics Institute • Dept. of Electrical and Computer Engineering
Carnegie Mellon University

ABB v ROY-G-BIV
TRIAL IPR2013-00062
TRIAL IPR2013-00282
ABB - EXHIBIT 1133

Chimera 3.2

Copyright © 1993, 1994, 1995 – all rights reserved.

Chimera 3.2 was developed in the Advanced Manipulators Laboratory of the Robotics Institute, Carnegie Mellon University (CMU).. The authors and CMU have used their best efforts in preparing the documentation and the corresponding software. These efforts include development, research, and testing of the theories and programs to determine their effectiveness. See Section 0.1 (in Preface) for general terms, licensing, and limitation of liability.

Advanced Manipulators Laboratory, The Robotics Institute
and
Department of Electrical and Computer Engineering

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213–3890

Email address for bugs, fixes, questions, and comments: *chimera@cmu.edu* (internet)

Chimera 3.2

Table of Contents

Preface	ix
0.1 Licensing and other Terms.....	ix
0.1.1 <i>Chimera 3.2</i> General License Agreement.....	ix
0.1.2 Limitation of Liability	xii
0.1.3 BSD 4.3 Unix Terms.....	xii
0.1.4 GNU General Public License.....	xii
0.1.5 GNU Library General Public License.....	xvi
0.2 History of Chimera	xxii
0.3 Overview of Documentation.....	xxiii
0.4 List of New Features	xxiv
0.4.1 <i>Chimera 3.0 (released 1993)</i>	xxiv
0.4.2 Chimera 3.1 (November 1993)	xxv
0.4.3 Chimera 3.2 (April 1994).....	xxv
0.5 Acknowledgements.....	xxvi
1. Beginner's Tutorial	1

1.1	Setting up your Chimera 3.2 Environment	2
1.2	A Simple Program Example	2
1.3	A Multitasking Example	5
1.4	A Real-Time Example	8
1.5	Error Detection and Handling	11
1.6	Terminating a Chimera 3.2 Session	11
1.7	Summary	11
2.	User's Manual	13
2.1	Compiling and Linking Programs	13
2.1.1	C Code for RTPUs	14
2.1.2	C++ Code for RTPUs	15
2.1.3	C Code for Host Workstation	15
2.1.4	Makefiles	15
2.2	chim Interface Commands	16
2.2.1	attach: Attach to an RTPU	18
2.2.2	bg: Execute Program in Background	18
2.2.3	cd: Change Working Directory	18
2.2.4	detach: Detach from an RTPU	19
2.2.5	download: Download a File	19
2.2.6	execute: Download and Execute Default RTPU	19
2.2.7	fg: Continue Execution in Foreground	20
2.2.8	go: Execute Program	20
2.2.9	kill: Interrupt Execution	21
2.2.10	killserver: Kill the qs Server	21
2.2.11	monitor: Use Monitor of Default RTPU	21
2.2.12	pwd: Print Working Directory	21
2.2.13	quit: Quit from Console Process	21
2.2.14	rcd: Change Default Directory of RTPU	22
2.2.15	select: Select the Default RTPU	22
2.2.16	status: Print Out Status of System	22
2.2.17	tables: Start the tables Interface	23
2.2.18	version: Print Current Version Information	24
2.3	Ugraph	24
2.3.1	Starting ugraph	24
2.3.2	The Main Window	24
2.3.3	Plotting a Function	25
2.3.4	Plotting UXDF Data	26
2.3.5	Viewing the Raw Data	26
2.3.6	Legends and Labels	28
2.3.7	Line Types	30
2.3.8	Resizing the Graph	30
2.3.9	Ranges and Zooming	31
2.3.10	Curve Interpolation Method	31
2.3.11	Graph Sampling Period	31
2.3.12	Live Mode	32

2.4	Debugging Chimera Programs.....	33
2.4.1	Embedded I/O statements and macros.....	33
2.4.2	The VMEbus Analyzer.....	34
2.4.3	Post-mortem debugging with the UNIX nm utility.....	34
2.4.4	Post-mortem debugging with gidb.....	35
2.5	Monitors.....	43
2.5.1	Ironics IMON Monitor.....	43
2.5.2	Chimera IMONICA Monitor.....	43
2.6	tables Interface Commands.....	45
2.6.1	cpus: System CPU Tables.....	45
2.6.2	devices: Device Driver Tables.....	46
2.6.3	fds: File Descriptor Tables.....	46
2.6.4	host: Host Interface and Address Windows.....	46
2.6.5	local: Host Mappings to VMEbus Devices.....	47
2.6.6	log: Log Configuration Information.....	48
2.6.7	map: Mapping of VME Address Ranges to Host Windows.....	48
2.6.8	memory: Memory Expansion Boards.....	48
2.6.9	options: Current Option Settings.....	49
2.6.10	quit: Quit from tables Interface.....	49
2.6.11	resources: Resource Allocation.....	49
2.6.12	xm: Express Mail Configuration.....	49
3.	Programming Guidelines.....	51
3.1	File System Organization.....	52
3.1.1	Subdirectory Structure.....	52
3.1.2	Supporting Multiple Computing Architectures.....	53
3.1.3	Makefiles.....	54
3.2	Constants, Variables, and Subroutine Declarations.....	56
3.2.1	Register Variables.....	56
3.2.2	Variables.....	57
3.2.3	Global Variables.....	58
3.2.4	Subroutines.....	60
3.3	Naming Conventions.....	60
3.3.1	Macros and Constants.....	61
3.3.2	Local Variables and Subroutines.....	61
3.3.3	Types.....	62
3.3.4	Global State Variables.....	62
3.4	Libraries.....	63
3.4.1	Subroutine Libraries.....	64
3.4.2	Device Driver Libraries.....	64
3.4.3	Reconfigurable Module Libraries.....	64
3.5	General Coding Tips.....	65
3.5.1	Loops.....	65
3.5.2	State Variable Table.....	65
3.5.3	Generating the Framework for a Reconfigurable Module.....	66
3.5.4	Special Purpose Processors.....	66

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.