

Appendix A

Guide to the Extra Materials

A.1 Folders

The online materials (<http://extras.springer.com>) that supplement this book are divided into 2 folders, ARIA and Player. The ARIA folder contains a control program “control.cpp” that uses a number of simple behaviours contained in the files “RobotModes.cpp” and “RobotModes.h”. These behaviours do not inherit from the ArAction class. The control program also uses data processing techniques found in the files “SensorData.cpp” and “SensorData.h”. The control program can perform various demos depending on which macros are defined at the beginning of the program. For example, if you want to run the arm movement demo, uncomment the ARM definition and recompile. The ARIA folder also contains a single blob finding action (the file “BlobFind.cpp” is the action and “single_control.cpp” is the file that uses it), a group blob finding action (the file “BlobFindGroup.cpp” is the action group and “group_control.cpp” is the file that uses them) and a blob finding mode (the file “BlobFindMode.cpp” is the mode and “mode_control.cpp” is the file that uses it). The ARIA folder also includes a bash script “acompile” for quick compilation of ARIA client programs, and a test ACTS configuration file “actsconfig” with its associated .lut files “channel1.lut” and “channel2.lut”. A subfolder Images comprises images of a test robot environment and cans for ACTS training. It also includes a test map for use with MobileSim.

The Player folder contains a configuration file for a real robot “config-Player2.cfg” and world, include and configuration files for Stage 2 simulations “simple.world”, “mysimple.cfg”, “map.inc”, “sick.inc” and “pioneer.inc”. There are four demo programs, “goalseek.cc”, which performs goal seeking in the world described by “simple.world”, “gripper.cc”, where the robot collects blocks and transports them through the goal, “griptrack.cc”, where the robot tracks only the red blocks and transports them through the goal, and “joint.cc” which controls the 5D arm. The first three programs make use of behaviours and data processing methods contained in the files “Robot.cpp” and “Robot.h”. “WorldReader.h” is used with “goalseek.cc” when simulated robots are required to go to specific co-ordinates in

the world rather than find the gate and travel through it. It simply reads in the starting position of the robot from the world file. User documentation for these user-written “Robot” and “WorldReader” classes is available in [13]. The “joint.cc” program requires the file “args.h”, a standard test file supplied with Player. The files “miniRobot.h”, “miniRobot.cpp” and “minigoal.cc” are more simplified versions of “Robot.h”, “Robot.cpp” and “goalseek.cc”. The Player folder also includes a bash script “pcomp” for quick compilation of Player client programs and a subfolder bitmaps with the image “pen.png” needed for the test Stage simulations.

A.2 Testing the Programs

The programs in the ARIA folder were tested using a Pioneer P3-DX robot running Debian Linux 2.6.10 with ARIA version 2.4.1 and ACTS version 2.2.1. The programs in the Player folder were tested by running the Player server on the same robot with Player 2.0.1 installed in the default location. The client programs were run on a remote PC running Debian Linux 2.6.10 with Player 2.0.1 also installed in the default location. The program “joint.cc” was tested on a different Pioneer P3-DX robot with a 5D arm, running the same operating system and with Player 2.0.1 installed in the default location. The programs “goalseek.cc”, “gripper.cc” and “griptrack.cc” were tested in simulation using the remote PC described and Stage 2.0.1 installed in the default location. The program “joint.cc” was not tested in simulation as Stage does not support the 5D arm device.

References

1. Gerkey, B.: The Player Robot Device Interface (2005)
<http://playerstage.sourceforge.net/doc/Player-2.0.0/player/index.html>. Cited 3rd June 2009
2. Vaughan, R.: The Stage Robot Simulator (2007)
<http://playerstage.sourceforge.net/doc/Stage-2.0.0/>. Cited 3rd June 2009
3. Bryant, J. L.: Instructions for Rewiring a Pioneer Robot so that the PTZ Camera Device can be Connected to a Serial Port (ttyS1) on the On-board Computer (2006)
http://playerstage.sourceforge.net/index.php?src=faq#evid30_wiring. Cited 3rd June 2009
4. Vaughan, R. T., Gerkey, B., Howard, A.: On Device Abstractions For Portable, Reusable Robot Code. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robot Systems, pp. 2121-2427, Las Vegas, USA (2003)
5. Gerkey, B., Vaughan, R. T., Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: Proceedings of the 11th International Conference on Advanced Robotics, pp. 317-323, Coimbra, Portugal (2003)
6. Gerkey, B. P., Vaughan, R. T., Sty, K., Howard, A., Sukhatme, G., S., Mataric, M. J.: Most Valuable Player: A Robot Device Server for Distributed Control. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1226-1231, Wailea, Hawaii (2001)
7. SourceForge.Inc.: SourceForge.net email archive: PlayerStage-users (2009)
http://sourceforge.net/mailarchive/forum.php?forum_name=playerstage-users. Cited 3rd June 2009
8. Wilson, D.: RGB/YUV Pixel Conversion (2007)
<http://www.fourcc.org/fccyvrgb.php#BOURGEOIS>. Cited 3rd June 2009
9. Bruce, J.: CMVision Realtime Color Vision (2006)
<http://www.cs.cmu.edu/~jbruce/cmvision/>. Cited 3rd June 2009
10. Niku, S.B.: Introduction to Robotics: Analysis, Systems, Applications. Prentice Hall (2001)
11. Gan, J. Q., Oyama, E., Rosales, E. M., Hu, H.: A complete analytical solution to the inverse kinematics of the Pioneer 2 robotic arm. *Robotica* **23**, 123-129 (2005)
12. Oates, R.: Bob's Guide To using ARIA (2006)
<http://www.boboates.co.uk/aria.pdf>. Cited 3rd June 2009
13. Whitbrook, A. M.: An Idiotypic Immune Network for Mobile Robot Control. MSc Dissertation, School of Computer Science, University of Nottingham (2005)
14. ActivMedia Robotics Interface for Application (ARIA) ActivMedia, NH, USA, (2005), available with ARIA software download
15. Pioneer 3-SH Operations Manual, Version 1, ActivMedia, NH, USA, (2004)
16. ACTS User Manual, Version 6, ActivMedia, NH, USA, (2006)
17. ActivMedia Robotics' Pioneer Gripper Manual, Version 6, ActivMedia, NH, USA, (2004)
18. ActivMedia Robotics' Pioneer Arm Manual, Version 5, ActivMedia, NH, USA, (2004)

19. Laser Range-Finder Installation and Operations Manual, Version 1, ActivMedia, NH, USA, (2002)
20. PTZ Robotic Cameras Manual, Version 3, ActivMedia, NH, USA, (2003)
21. VC-C4/VC-C4R Instruction Manual, Version 3, Canon, Japan (2000)

Index

- 5D arm, 4, 18, 27, 28, 79, 80
- 5D arm approach vector, 80, 83
- 5D arm end effector position, 80, 83
- 5D arm limb offset, 80, 83
- 5D arm orientation vector, 80, 83
- 5D arm servos, 28, 30

- accelerated Stage simulations, 104
- actarray interface, 79
- ActArrayProxy, 80
- action group, 57
- action resolver, 1, 53
- activate() method, 57, 60
- ACTS, 4, 11, 18, 37–40, 42–44, 55, 91
- ACTS configuration file, 37, 38, 42, 43, 91
- acts driver, 91
- ACTS pref file, 43, 44
- actuator array interface, 79
- addAction() method, 57, 58
- addRangeDevice() method, 16
- Alien software, 5, 7
- alwayson attribute, 65
- ArAction class, 1, 19, 53, 55, 57
- ArAction subclass, 53, 57
- ArActionConstantVelocity subclass, 57
- ArActionDesired class, 53, 56
- ArActionGroup class, 57
- ArActionGroup subclass, 57, 58, 60
- ArActionLimiterBackwards subclass, 57
- ArActionLimiterForwards subclass, 57
- ArACTS_1_2 class, 44
- ArACTS_1_2::getBlob() method, 44
- ArACTS_1_2::getNumBlobs() method, 44
- ArACTSBlob class, 44
- ArACTSBlob::getArea() method, 44
- ArACTSBlob::getXCG() method, 44
- ArACTSBlob::getYCG() method, 44

- ArArction class, 55
- ArArgumentParser class, 14, 15
- ARCOS, 1
- ArForbiddenRangeDevice class, 50
- ArGripper class, 30
- ArKeyHandler class, 62
- ArMode subclass, 60
- ArNetworking, 2, 3, 14, 50
- ARNL, 50
- AROS, 1, 3, 27
- ArP2Arm class, 27–29
- ArP2Arm::moveTo() method, 29
- ArP2Arm::NumJoints attribute, 29
- ArPTZ class, 18, 33
- ArRangeDevice class, 16, 18, 21, 22
- ArRangeDeviceThreaded class, 16
- ArRobot class, 15, 16, 18, 19, 21, 23–25, 28, 53, 56, 62
- ArRobot::clearDirectMotion() method, 53
- ArRobot::run() method, 15
- ArRobot::runAsync() method, 15
- ArRobot::setVel() method, 53
- ArSensorReading class, 21, 23
- ArSick class, 23
- ArSimpleConnector class, 14, 16, 17, 47
- ArTcpConnection class, 47
- ArUtil::sleep() method, 26, 29
- ArVCC4 class, 18, 33, 34
- ATRV Jr, 3
- autoconf, 9
- automake, 9
- AUX port, 3

- B21r robot, 3
- baud rate, 65
- bitmap files (bmp), 100

- blob finding device, 4, 65, 67, 86, 88, 90, 93, 100
- BlobfinderProxy, 86, 88, 90, 91
- bump sensors, 4, 16, 24, 25, 77–79, 93
- BumperProxy, 77–79
- camerav41 driver, 65, 86–88, 91
- Canon VC-C4 camera, 4, 18, 33, 35, 37, 64, 85, 86
- canonvcc4 driver, 64, 85
- checkRangeDevicesCurrentPolar() method, 23
- Clodbuster robot, 3
- CMUcam, 3
- CMVision, 86–88
- CMVision configuration file, 65, 87, 88
- cmvision driver, 65, 86, 88, 91
- Color Machine Vision, 86, 87
- Configure() method, 76
- connectLaser() method, 17
- connectRobot() method, 16
- currentDesired attribute, 56
- currentReadingPolar() method, 21, 22
- deactivate() method, 60
- Debian Linux, 6, 7, 9
- DTOR() function, 74
- dynamic linker, 68
- endAngle attribute, 21
- ERI robot, 3
- ERRATIC robots, 3
- ERSDK robot, 4
- EZ-train software, 37–39
- failed dependencies error, 7
- fire() method, 53, 55, 56
- forward kinematics, 79
- framegrabber, 65
- FrameGrabberChannel variable, 44
- Garcia robot, 3
- getAbsoluteMaxRotVel() method, 19
- getAbsoluteMaxTransVel() method, 19
- GetActuatorGeom() method, 80
- GetBlob() method, 88, 89
- getBottom() method, 44
- getBreakBeamState() method, 31
- GetCount() method, 75, 76, 80, 88
- getGripState() method, 30, 31
- GetHeight() method, 89
- GetInnerBreakBeam() method, 85
- GetIntensity() method, 76
- getJoint() method, 29
- getLeft() method, 44
- GetLiftDown() method, 85
- GetLiftMoving() method, 85
- GetLiftUp() method, 85
- GetMaxAngle() method, 76
- getMaxNegPan() method, 34
- getMaxNegTilt() method, 34
- getMaxPosPan() method, 34
- getMaxPosTilt() method, 34
- GetMinAngle() method, 76
- getNumFrontBumpers() method, 25
- getNumRearBumpers() method, 25
- getNumSonar() method, 21
- GetOuterBreakbeam() method, 85
- GetPaddlesClosed() method, 85
- GetPaddlesMoving() method, 85
- GetPaddlesOpen() method, 85
- getPaddleState() method, 31
- GetPan() method, 86
- getPan() method, 34
- GetPose() method, 75, 79
- GetRange() method, 75, 76
- getRange() method, 21, 23
- GetRangeRes() method, 76
- getRawReadings() method, 23
- getRight() method, 44
- getRobotRadius() method, 21
- getRotVel() method, 19
- GetScan() method, 75
- GetScanRes() method, 76
- getSensorTh() method, 21, 23
- getSonarReading() method, 21
- getStallValue() method, 24, 25
- GetTilt() method, 86
- getTilt() method, 34
- getTop() method, 44
- getType() method, 31
- getVel() method, 19
- GetWidth() method, 89
- getXCG() method, 44
- GetXPos() method, 74
- GetYaw() method, 74
- getYCG() method, 44
- GetYPos() method, 74
- GetZoom() method, 86
- getZoom() method, 34
- GIMP toolkit (GTK), 9
- GNU Compiler Collection (gcc), 9
- grip sensor, 30
- gripClose() method, 31
- gripOpen() method, 31
- gripper infrared break beams, 30, 31, 84
- gripper lift mechanism, 30
- gripper paddles, 30, 31, 84
- gripperDeploy() method, 31

- gripperHalt() method, 31
- GripperProxy, 83, 84
- grippers, 4, 16, 18, 30, 31, 33, 67, 83, 85, 93, 100
- gripStop() method, 31

- hasFrontBumpers() method, 25
- hasRearBumpers() method, 25
- home() method, 30
- host argument, 71–73

- init() method, 34
- inverse kinematics, 79, 80
- inverse kinematics interface, 79
- IsAnyBumped() method, 77, 79
- IsBumped() method, 78, 79
- iterator object, 23

- jpeg files, 67, 103

- Kameleon 376BC robot, 3
- Khepera robot, 3

- laser range resolution, 65
- laser resolution, 65
- laserincrement half argument, 24
- LaserProxy, 73, 75, 76
- LD_LIBRARY_PATH variable, 10, 68
- ldconfig, 68
- libplayerdrivers library, 63
- libpng.so file, 8
- librtk2 library, 10
- libtool, 9
- liftDown() method, 31
- liftStop() method, 31
- liftUp() method, 31
- limb interface, 79
- limb_links attribute, 79
- limb_offsets attribute, 79
- limb_pos attribute, 79
- LimbProxy, 80, 83
- lists, 23
- loadDefaultArguments() method, 16
- lookup table (lut) files, 37, 39, 42, 43, 91

- make, 9
- Mapper3Basic, 7, 49
- Mapper3Basic dock points, 49, 50
- Mapper3Basic forbidden areas, 49, 50
- Mapper3Basic forbidden lines, 49
- Mapper3Basic goals, 49, 50
- Mapper3Basic home areas, 49, 50
- Mapper3Basic home points, 49
- Mapper3Basic map files, 46, 50
- Mapper3Basic map lines, 49
- MinimumRunLengthWidth variable, 44
- MobileEyes, 50
- MobileSim, 4, 5, 7, 16, 22, 24, 26, 30, 33, 34, 46–50
- MobileSim position data, 48
- mode attribute, 65
- motors, 18, 74
- move() method, 19
- MoveAtSpeed() method, 80
- MoveHome() method, 80, 83
- moveStep() method, 30
- moveStepTicks() method, 30
- moveStepTo() method, 30
- MoveTo() method, 80
- moveTo() method, 29, 30
- moveToTicks() method, 30
- moveVel() method, 30
- mpeg files, 103
- myCenter attribute, 29
- myHome attribute, 29
- myMax attribute, 29
- myMin attribute, 29
- myPos attribute, 29
- myTicksPer90 attribute, 29
- myVel attribute, 29

- NOMAD200 robot, 3
- non-ranged devices, 16, 18, 27, 28, 30, 33
- norm attribute, 65
- NTSC capture format, 65

- Obot d100 robot, 3
- odometric pose, 48
- openPort() method, 18, 44
- overlay mode, 40

- P2ArmJoint class, 29
- P2OS, 1, 3, 27
- p2os driver, 64, 79
- PAL capture format, 65
- pan attribute, 86
- pan() method, 34
- pan-tilt-zoom camera, 4, 16, 18, 33, 35, 64, 67, 85, 86, 93
- panRel() method, 34
- panTilt() method, 34
- panTiltRel() method, 34
- park() method, 30
- parseArgs() method, 16
- PATH variable, 10
- PKG_CONFIG_PATH variable, 10
- Player C++ client library, 68, 70, 72, 73, 77
- Player C++ client library proxies, 70
- Player configuration file, 63–67, 76, 79, 91

- Player debugging information, 66
- Player drivers, 9, 11, 63, 64
- Player PULL mode, 72
- Player PUSH mode, 72
- Player stage driver, 101, 102
- PLAYER_LIMB_STATE_OOR status, 80
- PlayerCc namespace, 72
- PlayerClient proxy, 70, 72–75, 77, 80, 83, 86, 88
- PlayerViewer, 10, 66, 67
- PlayerViewer command feature, 67
- port argument, 71–73
- portable network graphics (png) files, 100, 103
- portable pix map (ppm) files, 67, 100
- Position2dProxy, 71, 74
- powerOff() method, 30
- powerOn() method, 29
- PSOS, 3
- PtzProxy, 86

- ranged devices, 16, 21, 22, 24
- Read() method, 75, 80
- Red Hat Linux, 6, 7
- Red Hat Package Manager, 5–7
- RequestBumperConfig() method, 79
- RequestGeometry() method, 80
- requires key word, 65
- ResetOdometry() method, 74
- RFLEX-based robots, 3
- RGB data, 65, 87–90
- robot toolkit (RTK), 10
- robot trails, 48, 104
- Robotic Mobility Platform, 3
- Roomba robot, 3
- root permissions, 8, 9, 11
- RS-232, 1, 2
- runAsync() method, 28

- savClient software, 38, 39
- savServer software, 38
- segmentation fault, 16
- server information packets (SIPs), 15
- setAbsoluteMaxRotVel() method, 19
- setAbsoluteMaxTransVel() method, 19
- SetCam() method, 86
- setDeltaHeading() method, 19, 56
- SetGrip() method, 85
- setHeading() method, 19
- SetMotorEnable() method, 74
- SetOdometry() method, 74
- SetPose() method, 80
- setRobot() method, 18
- setRotVel() method, 18
- SetSpeed() method, 74, 86

- setVel() method, 18
- setVel2() method, 18
- SICK LMS200 laser sensor, 4, 16, 22–24, 48, 65, 67, 73, 75, 76, 93
- sicklms200 driver, 65
- size attribute, 65
- sonar sensors, 4, 16, 21–23, 25, 48, 67, 73, 75, 76, 93
- SonarProxy, 73, 75
- source attribute, 65
- Stage bcount attribute, 98
- Stage bitmap attribute, 100
- Stage blob_return attribute, 94
- Stage blobfinder model, 97
- Stage blobfinder_return attribute, 101
- Stage bpose attribute, 98
- Stage bumper model, 98
- Stage center attribute, 100
- Stage channel_count attribute, 97
- Stage channels attribute, 97
- Stage color attribute, 94, 97
- Stage configuration file, 88, 100–102, 104
- Stage define statement, 97
- Stage drive attribute, 98
- Stage fov attribute, 97
- Stage gripper model, 97
- Stage gripper_return attribute, 94, 98, 101
- Stage gui_boundary attribute, 94
- Stage gui_grid attribute, 94
- Stage gui_interval attribute, 100, 103, 105
- Stage gui_movemask attribute, 94, 101
- Stage gui_nose attribute, 94, 98
- Stage image attribute, 97
- Stage include file, 94, 97
- Stage include keyword, 94
- Stage interval_real attribute, 100, 103–105
- Stage interval_sim attribute, 100, 103–105
- Stage laser model, 97
- Stage laser_return attribute, 94, 100
- Stage map model, 100
- Stage mass attribute, 94, 98
- Stage models, 93
- Stage obstacle_return attribute, 94
- Stage origin attribute, 94, 98
- Stage point attribute, 98
- Stage points attribute, 98
- Stage polygons model, 98
- Stage pose attribute, 94
- Stage position model, 98
- Stage ptz attribute, 97, 98
- Stage ptz model, 98
- Stage ptz_speed attribute, 98
- Stage range_max attribute, 97
- Stage range_min attribute, 97

- Stage ranger model, 97
- Stage ranger_return attribute, 94, 100
- Stage resolution attribute, 100
- Stage samples attribute, 97
- Stage scale attribute, 100
- Stage scout attribute, 97
- Stage size attribute, 94, 97, 98, 100
- Stage spouse attribute, 97
- Stage ssize attribute, 97
- Stage sview attribute, 97
- Stage window entity, 100
- Stage world file, 93, 94, 97, 100–105
- startAngle attribute, 21
- Stop() method, 83
- stop() method, 19, 30
- symbolic link, 8

- TCP socket, 2
- thresh mode, 40

- tick, 30
- tilt attribute, 86
- tilt() method, 34
- tiltRel() method, 34
- training an ACTS channel, 41
- true pose, 48

- unix sleep() function, 105
- unix usleep() function, 105

- visible mode, 40

- X-Windows, 66

- YUV data, 65, 87, 88

- zoom attribute, 86
- zoom() method, 34