Adept Motivity Core
User's Guide

Adept Motivity Core
User's Guide

10757-000 Rev. A
May, 2011

5960 Inglewood Drive • Pleasanton, CA 94588 • USA • Phone 925.245.3400 • Fax 925.960.0452
Otto-Hahn-Strasse 23 • 44227 Dortmund • Germany • Phone +49.231.75.89.40 • Fax +49.231.75.89.450
Block 5000 Ang Mo Kio Avenue 5 • #05-12 Techplace II • Singapore 569870 • Phone +65.6755 2258 • Fax +65.6755 0598

Copyright Notice

Table of Contents

Chapter 1: Introduction

This Adept Motivity Core User's Guide provides both the general and technical details you need to integrate the Motivity Core mobility controller hardware and software with your own mobile robot, and to begin developing your own robotics hardware and software.

For additional documentation and software, refer to Support on page 11.

1.1 Core Operating Environment

Robot

The Motivity Core mobility controller is designed to fit inside a user-designed robot. It takes care of navigation and motor control (but not motor power). It also manages the robot's sensors. These include the laser range finder (LRF), which is included with the Core, and bumper sensors, which are user-supplied. A gyroscope is built into the Core.

It is assumed that your robot will have two drive wheels, with an encoder on each. Skid-steer robots may have four drive wheels, but it is assumed that the two wheels on each side are mechanically joined, requiring just one encoder per side.

Operating Environment

The navigation and avoidance systems built into the Core are designed for indoor robots, generally in a wheelchair-accessible environment. You can adjust parameters to accommodate the Core to different robot specifications and operating environments.

Care must be taken to avoid:

- glass doors/walls
- pits without railings or low bumpers
- floors with access panels removed
- loose cables, hoses, etc.
- large, highly-reflective objects

The Motivity Core is not intended for use in hazardous environments, such as near explosive gas, water, dust, or oil mist.

The Motivity Core is not intended for air- or ship-based applications, because the gyroscope output won't be accurate if the environment is moving.

Temperature, Water, Vibration

The Core can operate in a temperature range of 5 - 50° C (41 - 122° F) ambient, measured at its air-circulation inlet. It will automatically shut down at 50° C.

The Core is not waterproof. It is rated IP-51.

Humidity should be 0 - 95%, non-condensing.

The Core is generally rugged enough for vibration likely to be encountered in normal use. Care should be taken to avoid falls, which could produce excessive shock.

Air Quality

Dirty or dusty environments will require more frequent cleaning or replacement of the fan filter, and possibly more frequent replacement of the fan as well.

Power

The Core can guide your robot to a charging station as needed, but the charging station itself and the entire power system onboard the robot are your responsibility.

The Core typically requires 65 W at 24 ± 10% VDC. This is in addition to the power needed for the robot's drive wheels, sensors, and any accessories.

Mounting

The Core needs to be mounted upright, unless arrangements have been made to mount the gyroscope in an alternate configuration.

1.2 Core Package

Adept manufacturing put your Motivity Core mobility controller and accessory laser range finder (LRF) through a burn-in period, and carefully tested them before shipping the products to you. Although we've made every effort to make your Motivity Core package complete, please check the components carefully after you unpack them.

Basic Components (all shipments)

- One Core with Motivity software installed
- One SICK LMS200 LRF with signal and power cables
- Two radio Ethernet antennas
- Software CD
- Documentation

User-Supplied Components / System Requirements

- Core connectors and hardware
- Client PC, preferably a portable
  - Microsoft Windows®
  - One Ethernet and one serial port
- Auto-recharging system

  On-board batteries and a charging station

Robot Requirements

- Symmetrical design preferred: Round or nearly square footprint, with drive wheels centered, or skid-steer, rotation-centered for near-zero turn radius
- Differential drive with PWM, DIRECTION, and ENABLE digital-controlled motor drivers
- Quadrature shaft encoder for each drive wheel, suitable for dead reckoning (100 - 200 resulting ticks/mm of travel) or one encoder per robot side for skid-steer robots

  Encoders should be single-ended, with 5 V digital logic.
- Forward mount with 180° unobstructed view (planar) for SICK LMS200
- 24 ± 2.4 VDC power with external motor-power emergency stop
- Power switches and operations indicators

Optional Requirements (for ± 15 mm positioning accuracy)

- Triangular-shaped target to be detected by laser sensor
- Additional tactile or position sensor on or off the robot to detect exact stop position

Environmental Retrofitting Requirements

- Highly-dynamic areas may require laser-height rail
  If there are too many objects in the environment that change their positions (such as chairs), the Core will have difficulty localizing.
- Highly-reflective or transparent surfaces, such as glass doors, some glass walls, and glossy painted surfaces require a 75-100 mm stripe at laser height.

1.3 Support

Refer to the How to Get Help Resource Guide (Adept P/N 00961-00700) for details on getting assistance with your Adept software and hardware. Additionally, you can access information sources on Adept's corporate web site:

http://www.adept.com

For details on getting assistance with your Adept software or hardware, you can access the following information sources on the Adept corporate website:

- For contact information: http://www.adept.com/contact/americas
- For product support information:
  http://www.adept.com/support/service-and-support/main
- For user discussions, support, and programming examples:
  http://www.adept.com/forum/
- For further information about Adept Technology, Inc.: http://www.adept.com
- For product support email, use:

support@adept.com

Be sure to tell us your Core's Serial Number.

Information on the Advanced Robotics Command Language (ARCL) can be found in the referenece manual for ARAM versions 3.0 and later.

Chapter 2: Safety

2.1 Dangers, Warnings, Cautions, and Notes

There are six levels of special alert notation used in Adept manuals. In descending order of importance, they are:

DANGER: This indicates an imminently hazardous electrical situation which, if not avoided, will result in death or serious injury.

DANGER: This indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING: This indicates a potentially hazardous electrical situation which, if not avoided, could result in serious injury or major damage to the equipment.

WARNING: This indicates a potentially hazardous situation which, if not avoided, could result in serious injury or major damage to the equipment.

CAUTION: This indicates a situation which, if not avoided, could result in minor injury or damage to the equipment.

NOTE: Notes provide supplementary information, emphasizes a point or procedure, or gives a tip for easier operation.

2.2 Safety Standards

CAUTION: Adept Technology strictly prohibits installation, commissioning, or operation of an Adept robot without adequate safeguards according to applicable local and national standards. Installations in EU and EEA countries must comply with EN 775/ISO 10218, especially sections 5,6; EN 292-2; and EN 60204-1, especially section 13.

Adept recommends that you maintain a similar level of safety for any robot using the Adept Motivity Core mobility controller.

2.3 Additional Safety Information

The standards and regulations listed in this manual contain additional guidelines for robot system installation, safeguarding, maintenance, testing, startup, and operator training. The following table lists some sources for the various standards.

Table 2-1. Standards and Directives

SEMI International Standards 3081 Zanker Road San Jose, CA 95134 USA Phone: 1.408.943.6900 Fax: 1.408.428.9600 http://www.semi.org	American National Standards Institute (ANSI) 11 West 42nd Street, 13th Floor New York, NY 10036 USA Phone 212-642-4900 Fax 212-398-0023 http://www.ansi.org
Underwriters Laboratories Inc. 333 Pfingsten Road Northbrook, IL 60062-2096 USA Phone: +1-847-272-8800 Fax: +1-847-272-8129 http://www.ul.com/info/standard.htm	BSI Group (British Standards) 389 Chiswick High Road London W4 4AL United Kingdom Phone +44 (0)20 8996 9000 Fax +44 (0)20 8996 7400 http://www.bsi-global.com
Global Engineering Documents 15 Inverness Way East Englewood, CO 80112 USA Phone 800-854-7179 Fax 303-397-2740 http://global.ihs.com	Document Center, Inc. 1504 Industrial Way, Unit 9 Belmont, CA 94002 USA Phone 415-591-7600 Fax 415-591-7617 http://www.document-center.com
IEC, International Electrotechnical Commission Rue de Varembe 3 PO Box 131 CH-1211 Geneva 20 Switzerland Phone 41 22 919-0211 Fax 41 22 919-0300 http://www.iec.ch	Robotic Industries Association (RIA) 900 Victors Way PO Box 3724 Ann Arbor, MI 48106 USA Phone 313-994-6088 Fax 313-994-3338 http://www.robotics.org
DIN, Deutsches Institut fur Normung e.V. German Institute for Standardization Burggrafenstrasse 6 10787 Berlin Germany Phone.: +49 30 2601-0 Fax: +49 30 2601-1231 http://www.din.de http://www2.beuth.de/ (publishing)	

2.4 Intended Use of the Robot

The installation and use of Adept products must comply with all safety instructions and warnings in this manual. Installation and use must also comply with all applicable local and national requirements and safety.

The Adept equipment is not intended for use in any of the following situations:

- In hazardous (explosive) atmospheres
- In mobile, portable, marine, or aircraft systems
- In life-support systems
- In residential installations
- In situations where the Adept equipment will be subject to extremes of heat or humidity.

CAUTION: The instructions for installation, operation, and maintenance given in this manual must be strictly observed.

Non-intended use of an Adept Motivity Core mobility controller can:

- Cause injury to personnel
- Damage the robot or other equipment
- Reduce system reliability and performance

If there is any doubt concerning the application, ask Adept to determine if it is an intended use or not.

2.5 Safety Requirements for Additional Equipment

If the robot is to be used in an EU or EEA member country, all accessories used with the robot must comply with the safety requirements in the European Machine Directive 89/392/EEC (and subsequent amendments) and related harmonized European, international, and national standards. For robot systems, these include: EN 775/ISO 10218, sections 5,6; EN 292-2; and EN 60204.

- In other countries, Adept strongly recommends, in addition to complying with the applicable local and national regulations, that a similar level of safety be obtained.
- In the USA, applicable standards include ANSI/RIA R15.06 and ANSI/UL 1740.
- In Canada, applicable standards include CAN/CSA Z434.

2.6 Qualification of Personnel

This manual assumes that all personnel have attended an Adept training course and have a working knowledge of the system. The user must provide the necessary additional training for all personnel who will be working with the system.

As noted in this user's guide, certain procedures should be performed only by skilled or instructed persons. For a description of the level of qualification, Adept uses the standard terms:

- Skilled persons have technical knowledge or sufficient experience to enable them to avoid the dangers, electrical and/or mechanical.
- Instructed persons are adequately advised or supervised by skilled persons to enable them to avoid the dangers, electrical and/or mechanical.

All personnel must observe sound safety practices during the installation, operation, and testing of all electrically powered equipment. To avoid injury or damage to equipment, always remove power by disconnecting the power from the source before attempting any repair or upgrade activity. Use appropriate lockout procedures to reduce the risk of power being restored by another person while you are working on the system.

DANGER: Any person who programs, teaches, operates, maintains, or repairs the robot system must be trained and must demonstrate the competence to safely perform the assigned task.

2.7 Safety Aspects While Performing Maintenance

Only skilled persons with the necessary knowledge about safety and the equipment are allowed to maintain the robot.

DANGER: During maintenance and repair, the power to the robot and charging station must be turned off. Unauthorized third parties must be prevented from turning on power through the use of lockout measures.

DANGER: During maintenance and repair, disconnect the batteries of the robot as soon as possible. Avoid shorting the terminals of the batteries.

2.8 Risks Due to Incorrect Installation or Operation

Take precautions to ensure that these situations do not occur.

- Purposely defeating any aspect of the safety E-Stop system
- Improper installation or programming of the robot system

2.9 What to Do in an Emergency Situation

Press the E-Stop button (a red push-button on a yellow background/field) and then follow the internal procedures of your company or organization for an emergency situation. If a fire occurs, use CO_2 to extinguish the fire.

Chapter 3: What Is Motivity?

Motivity is a collection of technologies—sensors and data-processing hardware with state-of-the-art robotics software—that enable and support autonomous, intelligently responsive and safe mobility of motorized vehicles in indoor environments. A Motivity robot uses a map of the physical features of its workspace—walls, doors, hallways, shelves, and so on—to keep track of where it is and how to get from one place to another. Because of this, deploying a Motivity robot doesn’t require costly and inflexible retrofitting of the workspace.

Moreover, Motivity comes with the MobilePlanner and MobileEyes software tools that let you quickly and easily develop, deploy, modify, and maintain an entire mobile application without having to write or compile a single line of code.

3.1 Motivity Hardware

The essential Motivity hardware, embodied in the Core package, consists of:

- Robot microcontroller with µARCS firmware that tightly manages mobility and sensors for differential-drive robots. It includes a gyroscope and power management.
- SICK LMS200 laser range finder (LRF) for detailed, long-range (16 meters) scanning of the robot's operating environment and for "virtual bumper" obstacle avoidance.
- A single-board computer (SBC) to host the Motivity software, custom applications, and accessory drivers.
- Wired and wireless Ethernet for remote command and monitoring of the robot and its onboard payload over a LAN or WAN.
- Optional accessory LCD for status and interactive, manual control of onboard systems power.
- Optional accessory joystick for manually driving the robot.

3.2 Motivity Software

The power of a Motivity Core mobility controller resides in its software. Motivity software enables physical features-based, self-guided mobility and intelligent operation of the robot, including:

- At the lowest level, a microcontroller running µARCS firmware handles the details of mobility, including maintaining the robot's drive speed and heading, as well as acquiring sensor readings, such as from the encoders, and gyroscope, and managing the robot's emergency stop systems and bumpers. The µARCS controller computes and reports the robot's odometry (X, Y, and heading), sonar readings, and a variety of other low-level operating conditions to ARAM.
- Advanced Robotics Automation Manager (ARAM) is the core of the Core. It provides the high-level robotics intelligence for robot mobility and the performance of application tasks.

  For intelligent, autonomous mobility, ARAM:

  - combines and interprets the robot's sensor readings (LRF, encoders, gyro) and compares them with an onboard map to locate the robot in its operating environment.
  - plans and executes paths to reach accessible goals within the environment.
  - interacts with the microcontroller to drive the robot automatically.
  - re-plans and executes local variations in the path to avoid obstacles.
  - supports tasks, routes and goals:
    - Goals are named locations in the Motivity map corresponding with a real place in the operating environment to which the robot may be sent on command or as part of an automated sequence of goals (route).
    - Routes are series of goals that the robot may patrol automatically.
    - Tasks are routines and maneuvers the robot and/or its accessories may perform. Tasks may be performed as part of a route or when associated with a goal. Tasks are performed upon departure towards ("before") or arrival at ("after") a goal. Some tasks are universal, like robot motion control. These include turning the robot to a particular heading, or moving a prescribed distance, for instance. Other tasks are robot-specific, depending on supported accessories, like turning on a digital relay or making pan-tilt-zoom camera adjustments.
- SetNetGo™ also runs in the Motivity Core. It is used for configuring, maintaining, and detailed monitoring of the Motivity onboard systems and software.

The Motivity Core mobility controller comes with off-board software, too: applications that operate on a remote PC, either over the LAN or plugged into the Ethernet port on the Core. Motivity off-board software includes:

- µARCScf, an application used to configure various µARCS parameters in the Core's flash. It can also be used to upgrade the version of µARCS.
- MobileEyes, a comprehensive Graphical User Interface (GUI) application for remote

monitoring and temporary direct control of the robot and its various accessories and interfaces. Use it locally to initiate map-building, too. Over the network, MobileEyes lets you:

  - Track, in an onscreen display, the location and travels of a Motivity robot
  - Monitor sensor and robot conditions, including live alerts
  - Manually and safely drive the robot with keyboard or joystick
  - Send the robot to a specific place or goal, where the robot and/or its accessories may perform scheduled tasks
  - Enable a route or schedule of routes and tasks
  - Remotely operate the robot's accessories and payloads
  - Modify many of the robot's operating parameters, including maps, so it can adapt to changing environments

- MobilePlanner, a GUI map and task-planning application, which lets you do:
  - Mapping. Introduce your Motivity robot to its new operating environment with map-scanning mode in MobileEyes. Drive the robot around its workspace with the joystick so that the robot's LRF captures its physical features. Then input the scan into MobilePlanner to generate a map that your and other Motivity robots can use for localization and navigation in that workspace.
  - Editing. Use MobilePlanner to add virtual features to the map, including "forbidden" lines and areas where you do not want the robot to drive, to mark goals, and to stitch maps together.
  - Route Builder. Use MobilePlanner's built-in Route Builder utility to create patrol routes and to associate tasks, including accessory and sensor operations, to the robot's departure towards and/or arrival at a goal or goals and along routes.
  - Mini-MobileEyes. Use MobilePlanner's built-in MobileEyes monitor and control feature to perfect routes, tasks, and goals by having the robot actually go to a goal. Then adjust its pose there to get just the right position for a delivery, for example.
  - Distribution. MobilePlanner gives you the tools to manage distribution and implementation of the map over the network, as well as other files the robot may need or use.

Chapter 4: Connectors and Signals

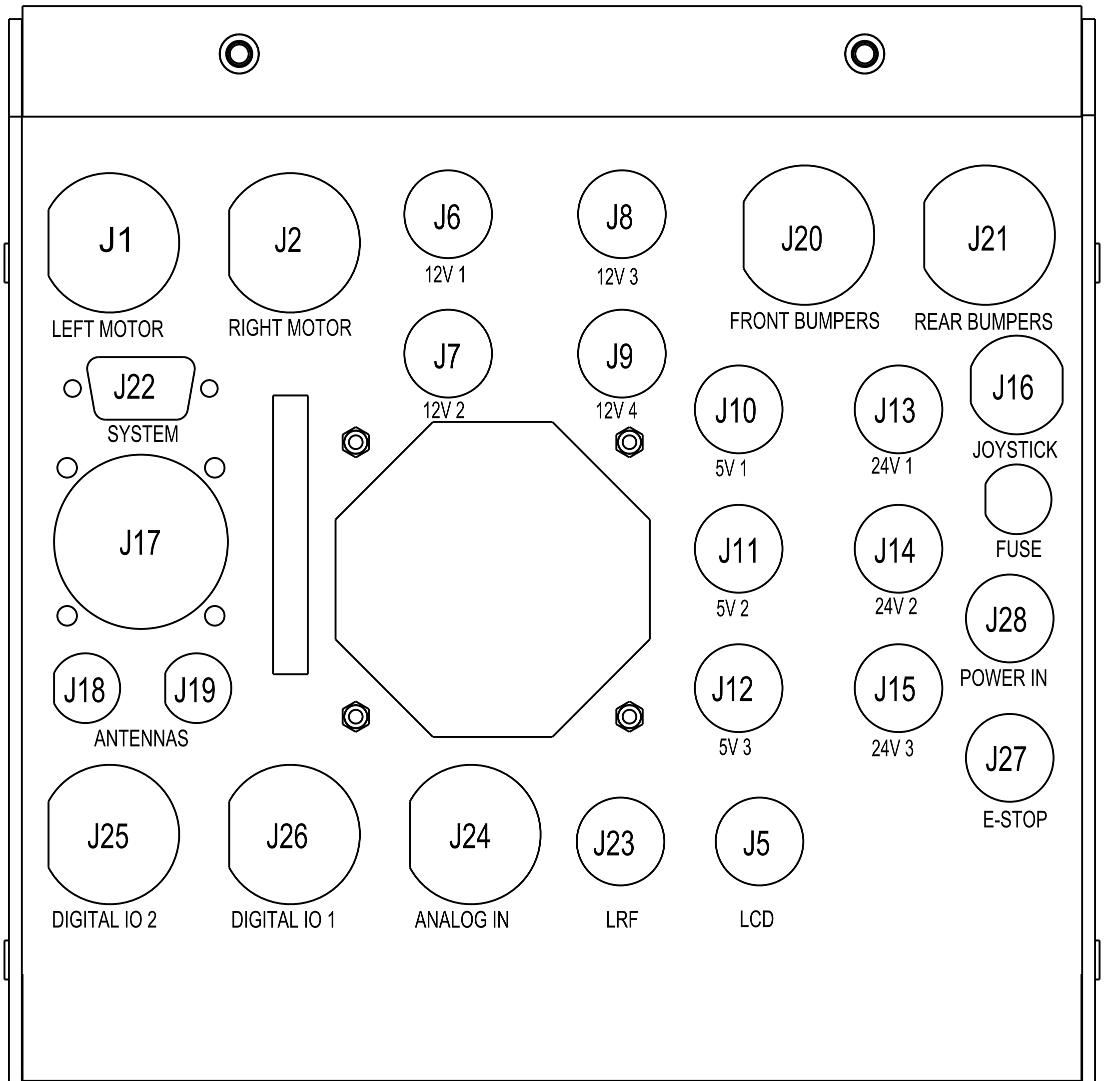


Figure 4-1. Motivity Core Connector Panel

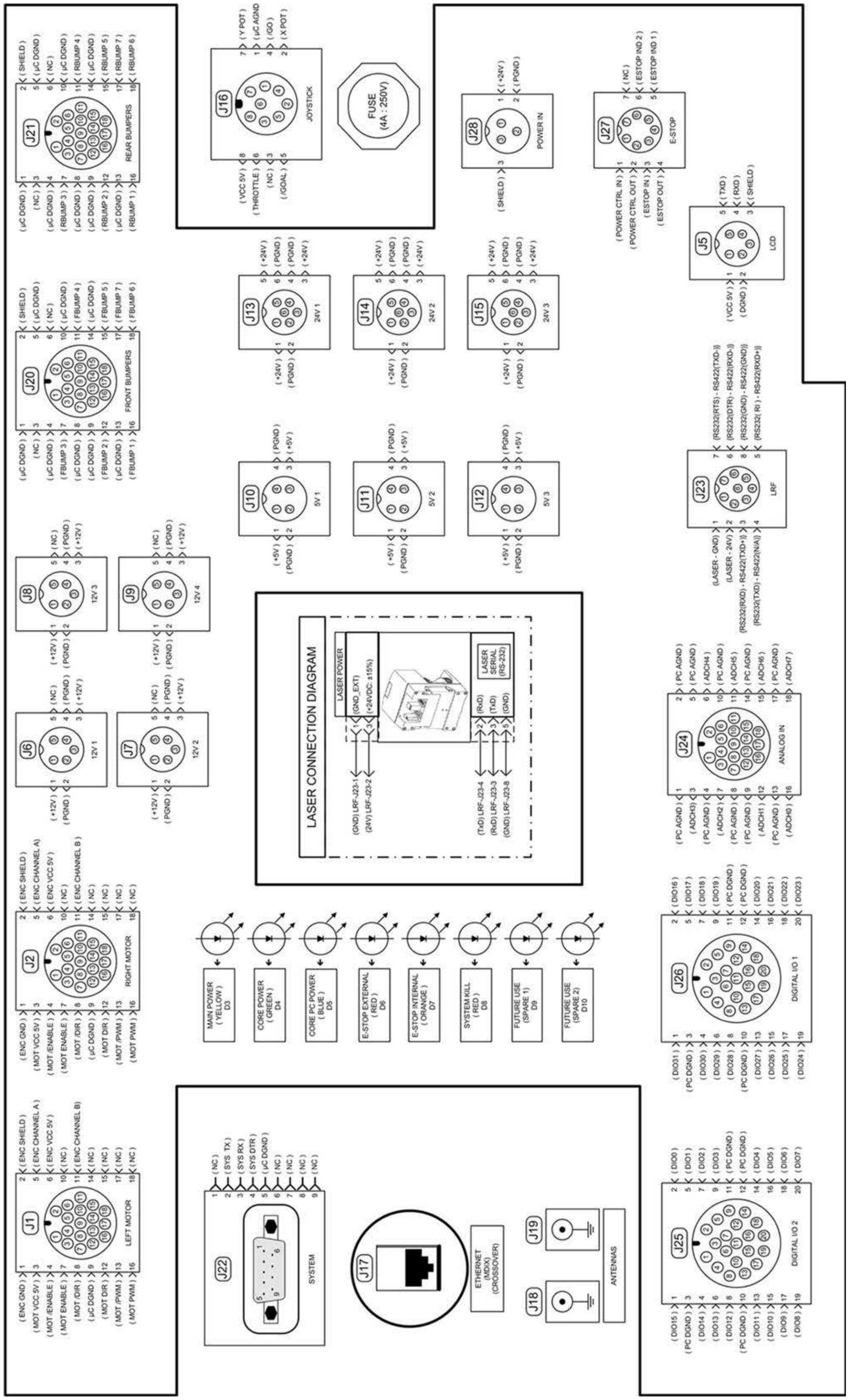


Figure 4-2. External Connections Diagram

4.1 Source Power, CTRL and Emergency Stop (J27-28)

Core electronics, including the LRF, but not additional accessories, operate from a single 65 W (max), 24 ± 2.4 VDC source applied at J28. It is up to the external power control systems to switch and condition power safely in the entire robot, including to the Core.

A 4 A 250 V fuse is located just above J28. The Adept part number is 10745-000.

Power CTRL and E-Stop

Relay-isolated switches on J27 provide ways for the Core to indicate to the external systems of a Core-requested power shutdown (between POWER CTRL J27:1 and J27:2) or a Core-initiated emergency stop (ESTOP between J27:3 and J27:4). The relay contacts normally are closed. They open to indicate Core-requested power shutdown or E-Stop. Contacts are rated at <3 A and should not be used to actually switch Core or other systems power.

Reciprocally, the robot notifies the Core of an E-Stop condition by opening a normally-closed switch, and shorting the ESTOP IND pins across J27:5 and J27:6.

4.2 Power Ports (J6-15)

The Core supplies and switches under software control 5 and 12 VDC conditioned and 24 ± 2.4 VDC (source) auxiliary power at connectors J6 through J15. Refer to the following table for current ratings and software addresses. The total Core system load should not exceed 100 W, including the laser and controllers. This leaves about 35 W to be used from the auxiliary ports.

SBC, LRF, and Auxiliary Power

µARCS allocates power control ports and states as firmware parameters. This way, you can allocate or change a power port for attached devices, including the LRF, one each auxiliary 5, 12, and 24 V, as well as for motion, fault, and battery state indicators. Once allocated, the power port state appears on the LCD and you can turn power on or off from LCD Interactive Mode.

FLASH state parameters also let you specify whether the SBC and other allocated power control ports stay off or come on when the robot starts up or when a connection starts or stops. (The SBC power port always is 0, and because connections depend on the SBC, you can only specify if it powers up on power restart.)

For example, the LRFPort parameter is set to 11 because that is where the LRF gets its power. You can set its LRFPower state parameter to 1 so that its power switches on as soon as the microcontroller starts up. Or set it to 4, in order to start/stop the LRF when a connection starts/stops. See µARCS.

Similarly allocate the Aux5VPort, Aux12VPort, and Aux24VPort to your accessory devices, and set their respective Aux5VPower, Aux12VPower, and Aux24VPower state parameters.

4.1Auxiliary Power Connectors and Control Ports

Core Name	CORE Junction Number	Software Address Number	Actual Voltage	Current limit	Normal Shutoff Response Time
5V #1	J10	4	5.0 ± 0.2 VDC	2 A	29 ms
5V #2	J11	5	5.0 ± 0.2 VDC	2 A	29 ms
5V #3	J12	6	5.0 ± 0.2 VDC	2 A	29 ms
12V #1	J6	7	12.0 ± 0.5 VDC	2 A	29 ms
12V #2	J7	8	12.0 ± 0.5 VDC	2 A	29 ms
12V #3	J8	9	12.0 ± 0.5 VDC	4 A	29 ms
12V #4	J9	10	12.0 ± 0.5 VDC	2 A	29 ms
24V #1	J13	12	24 ± 2.4 VDC	1.125 A	21 ms
24V #2	J14	13	24 ± 2.4 VDC	4.5 A	21 ms
24V #3	J15	14	24 ± 1 VDC	4 A	21 ms

MotionLampPort, FaultLampPort, and BattLampPort

If the Core-based vehicle provides lamps on a user control panel, for example, µARCS supports three special indicators:

- The MotionLampPort is the FLASH parameter allocated for the mobile robot's motion indicator. It toggles power at that designated auxiliary control port on and off quickly just before the robot starts to move, and rhythmically while it moves. The MotionLampPort goes off when the robot is stopped or idle. Set the parameter to zero to disable the behavior.
- The FaultLampPort monitors the Motivity connection between the SBC and microcontroller, as well as E-STOP. When allocated and attached, the fault lamp flashes whenever a robot E-STOP button gets pressed. Otherwise, on startup, the lamp is on continuously until ARAM starts up and establishes a connection with the microcontroller. While the connection remains active or when in Joy-Only drive mode, the lamp is off.
- The BattLampPort monitors power incoming to the Core and the Core's internal temperature sensors. When allocated and attached, the associated indicator lamp turns on continuously any time any of the Core's temperature sensors exceed their rated limit. Otherwise, if the Core's power drops to or below the lowBattery threshold, the indicator lamp flashes.

4.3 Motors and Encoders (J1-2)

The Core runs a differential-drive, motorized mobile robot using pulse-width modulated (PWM) left and right motor-control output signals available on connectors J1 and J2, respectively. PWM, direction (DIR), and enable (ENABLE) signals all are digital outputs 0 or 5 VDC, normal logic (0=OFF; 5 V=ON) and inverted logic (/ENABLE, for example: 5 V=OFF; 0=ON).

Motor power amplifiers are separate and supplied by the end user. Encoders should be single-ended, with 5 V digital logic.

The Core controller's proportional-integral-differential (PID) control loop, speed, and heading computations require left and right quadrature encoder signal inputs, channels A and B, on the same connectors, and also supplied separately.

Emerging PWM signals operate at a 20 KHz duty cycle, proportionally high (PWM) or low (/PWM) at the end of the cycle relative to the desired wheel speed and torque. Motor direction is controlled via the respective active high (DIR) or low (/DIR) signals. The respective motors-enable signals, either active high (ENABLE) or low (/ENABLE), can be used to activate or disable motor amplifiers and, with additional circuitry, be used to control brakes.

Differential-drive logic and low-level odometry are functions of the Core microcontroller's firmware. Adjustable parameters in FLASH include:

- RevCount: the differential number of encoder ticks for a 360-degree rotation and is used to measure heading and rotation speed
- Ticks/mm: the number of encoder ticks per millimeter of travel for measuring translation distance and speed
- DriftFactor: adjusts for differences between left and right tire circumference
- PWMMax: lets you limit the maximum PWM output pulse
- A number of various speed, acceleration and PID factors and limits that affect the translation and rotation of the robot. See µARCS on page 43 for more details.

4.2Example Core Connections with an Amplifier
(Advanced Motion Controls model BD30A8 and Motor)

	Amplifier Signals	**Connections**
Core Left (J1)		
	PWM +	+5 Volts (J1:3)
	PWM -	/PWM (J1:13)
	DIR +	+5 Volts (J1:3)
	DIR -	DIR (J1:12)
	INH +	+5 Volts (J1:3)
	INH -	/ENABLE (J1:4)
	+ 5 IN	+5 Volts (J1:3)
Motor and Power		
	+ FAULT	no connection
	- FAULT	no connection
	+ V HALL OUT	Red Hall Effect Wire
	HALL 1	White Hall Effect Wire
	HALL 2	Green Hall Effect Wire

	Amplifier Signals	Connections
	HALL 3	Brown Hall Effect Wire
	CURR. MONITOR OUT	no connection
	MOTOR A	White Motor Wire
	MOTOR B	Black Motor Wire
	MOTOR C	Red Motor Wire
	POWER GND	Ground

4.4 System Serial (J22)

A common 9-pin DSUB connector provides RS-232 serial access to the Core controller for COREcf-mediated controller FLASH parameter and software maintenance. See µARCS on page 43 for details.

4.5 LRF Serial and Power (J23)

The RS-232 serial port pins for the Core's accessory LRF (J23) go to a serial port on the internal SBC. The power pins on the Core LRF connector go to the 24 VDC software-switched power port number 11. The Core comes with an LRF-to-Core cable and connectors sized to the application's specifications.

4.6 Ethernet (J17-19)

Two Core connectors, J18 and J19, are for external radio-Ethernet antennas.

Because the other Ethernet connector crosses the signals, J17 accepts a standard CAT-5E cable and connectors for direct attachment of the Core SBC's Ethernet to an Ethernet port on an external PC. Some hubs and switches auto-detect the cable type, but you may have to supply a CAT-5E crossover cable to connect the Core SBC for those switches or hubs that don't auto-detect.

4.7 Joystick (J16)

The JOYSTICK connector (J16) is for the joystick accessory and connects directly with the Core controller for manual drive operation of the robot.

4.8 Bumpers (J20-21)

Two connectors, J20 and J21, provide seven digital inputs each for front and rear contact bumpers. The ports are pulled digitally high (5 VDC) internally. Bumper switches may be populated front and/or rear, but each must be wired in order, starting with Bumper 1. Enter the number of active segments front and rear in their respective COREcf-mediated controller FLASH parameters, and whether active high or active low in the InvertBumps FLASH setting.

4.9 Analog IN (J24) and Digital IO (J25-26)

Eight pins on the J24 connector go to the Core SBC's analog input ports. Analog resolution is 12 bits (0-4095), measured at the port versus PC AGND, software-selectable per channel (16 V maximum input) for 0-to-5 VDC.

Thirty-two pins on two connectors, J25 and J26, go to the Digital I/O ports in the Core's SBC. Blocks of 8-bits each get designated input or output by the controlling application (all default to inputs on startup.) The outputs push-pull TTL drivers with the following characteristics:

Parameter	Description	Test Condition	Min.	Typ.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC}=Min., I_{OH}=-24 mA	2.4	3.3		V
V_{OL}	Output LOW Voltage	V_{CC}=Min., I_{OL}=24 mA		0.3	0.55	

Consult the VersaLogic EBX-12 PC manual for details.

4.10 Indicators

Behind a clear strip panel to the left of the Core's fan are a set of LED Core status indicators. From the top down, these are:

Function	Color	Description
Main Power	Yellow	24 V supply to Core is good. Lights even if the fuse is blown.
Core Power	Green	Core is powered on. Lights even if the fuse is blown.
Core SBC Power	Blue	Onboard computer supply is on (5 V is being supplied to the SBC). Does not indicate that the SBC itself if on.
E-Stop External	Red	External wiring loop for E-Stop circuitry has been tripped.
not used	Orange	
System Kill	Red	Lights momentarily when the onboard software or micro-controller decide to turn the Core off.

NOTE: The two bottom indicators (below System Kill) are not used.

4.11 LCD

The liquid crystal display (LCD) module and two rocker switches (SCROLL and SELECT) are available as an option.

The LCD module provides a window to the operation and status of your Core's micro-controller. Attach it to the LCD port on the Core. It should start automatically, including hot-plug. The module supports four lines of 20 characters for status messages, and has SCROLL and SELECT rocker switches that let you review additional operating values as well as enable and disable server functions.

In normal operating mode, the first line of the LCD has a spinning line segment that acts as a microcontroller activity indicator, and the source voltage. The value flashes when the source voltage drops below the LowBattery FLASH setting.

The following indicators are summarized in the table that follows.

When the SBC has power, a ‘C’ character appears. Similarly, ‘L’ is the LRF. A left angle bracket (<) indicates that the allocated Aux5VPort has power; an asterisk (*) is for Aux12VPower; and a right ange bracket (>) indicates Aux24VPower.

When the motors are engaged, an animated cart also appears in the upper left corner of the LCD. Its wheels change to the characters ‘S’ or ‘B’ if the robot stalls due to motor or bumpers, respectively.

An ‘E’ character appears and flashes when the robot’s E-STOP is engaged. A ‘G’ character appears and flashes if the integrated gyro is disabled for some reason.

The middle two lines are used at times to display error information, such as during a system shutdown due to very low souce voltage (ShutDownVolts FLASH parameter). The bottom line displays microcontroller status messages, such as when connected with a client or when the source voltage is low.

	Symbol	Meaning
Power	C	SBC power ON
	L	LRF power ON
	<	5 VDC ON
	*	12 VDC ON
	>	24 VDC ON
Errors	S	Wheel(s) stalled
	B	Bumper(s) activated
	E (flashing)	E-Stop engaged
	G (flashing)	Gyro disabled

LCD Interactive Mode

Press either the SCROLL or SELECT switch to activate LCD interactive mode. SCROLL switches the display message and acts to cancel a pending change. In all cases, the display reverts to normal mode, canceling any pending options, if you don’t press a SCROLL or SELECT button after a few seconds.

4.3LCD interactive display messages and options

Display	Select Options
μARCS version and build date	none
Robot Type and Serial Number	none
Engage uC Maintenance Mode?	Verify before activating maintenance mode
Client connection status	Disconnect if connected with a client

Display	Select Options
Motors status	Enable/disable if connected with a client
Joystick drive mode	(UN)SAFE if connected with a client
Scanning	Start/stop map scanning mode and looping
Calibration	Enable calibration mode
PC status	Enable/disable power to the SBC
LRF status	Enable/disable power to the LRF
Aux5V status	Enable/disable power to the allocated 5 V Port
Aux12V status	Enable/disable power to the allocated 12 V Port
Aux24V status	Enable/disable power to the allocated 24 V Port

4.12 Analog Temperature Sensor

Internally, the Core is protected against overheating by digital temperature sensors. In addition, the Core supports an analog-based temperature sensor attached to the microcontroller. The threshold is set in FLASH. Set the FLASH power (see Chapter 5) parameter tempThreshold to the degrees Celsius (°C) above which the sensor will trigger a system shutdown. A one degree hysteresis is built in, so the temperature has to drop below the tempThreshold in order to cancel a system shutdown. A tempThreshold of 0 or -128 disables the feature.

Chapter 5: Configuration and Operation

The Motivity Core mobility controller comes with on-board and off-board software. On-board software running on the Core SBC includes ARAM, the robotics applications core of Motivity, and SetNetGo, a web-based server for Core configuration and maintenance. Off-board software includes two networked GUI applications: MobileEyes, for network-based monitoring and executive-level control of ARAM, and MobilePlanner, for the creation of robotics maps and associated goals, routes, and tasks.

This chapter provides an operational overview. Consult online documentation for details of the software. A third off-board application, for update and management of the microcontroller firmware and operational parameters, is covered in the next chapter.

5.1 Startup Logic

Upon initial application of power, the Core starts up in an order of events dependent on its current operating condition:

Step 1: Core controller (µARCS firmware-mediated)

The Core's microcontroller monitors voltage at connector J28, as well as the internal temperature sensors.

- If the supplied voltage is above the value set by the LowBattery FLASH parameter (23.5 VDC by default) and the temperature reading is below the fixed rating for the digital temperature sensors, and the variable tempThreshold value for the analog temperature sensor accessory,
    - If a serial connector is attached to the SYSTEM serial port and its Pin 4 (DTR) is active, the Core starts up in Maintenance mode.
    - If a joystick is attached at the JOYSTICK connector and the operator holds down its green GO button, the Core starts up in Joy-Only drive mode.
    - If neither of the two preceding conditions apply, the Core startup proceeds to Step 2: SBC and LRF Power.
- If power at J28 is below ShutdownVolts (22.5 VDC by default), or if the Core's temperature sensor at the air-circulation inlet exceeds 50° C, or if the analog temperature sensor exceeds the tempThreshold, after approximately three seconds the POWER CTRL relay switch opens. The Motivity robot's power systems may detect the event and remove power from the entire system, including the Core, in an effort to save the batteries from deep-discharge or the onboard systems from over-temperature damage. If the external system does not remove power, the Core microcontroller remains on and continues to draw power, albeit little. The System Kill indicator LED is lit when the POWER CTRL relay is open.
- If power at J28 is below LowBattery, but above ShutdownVolts, the Core's microcontroller allows you to drive the robot with the joystick, presumably to move it somewhere where it can get its batteries recharged. The Core's Main Power and System

Power status LEDs are lit, but the SBC Power LED remains off. The battLampPort-related indicator is on continuously when in the low-battery state on startup.

Step 1a: Maintenance Mode

If, upon initial application of power, a serial connector is attached to the SYSTEM serial port and its Pin 4 (DTR) is active, the Core microcontroller starts up in Maintenance mode. No other Core systems start up. The Core's Main Power and System Power status LEDs are lit, but the SBC Power LED remains off. Refer to µARCS.

Step 1b: Joy-Only Drive Mode

If, upon initial application of power, a joystick is attached at the JOYSTICK connector and the operator holds down its green GO button, the Core microcontroller starts up in Joy-Only drive mode, subject to the power conditions described in Step 1. In this mode, the Core's Main Power and System Power status LEDs are lit, but the SBC and related SBC Power LED remain off.

In Joy-Only drive mode, only the low-level drive systems of the Core are operational. The robot is not protected by the Core's obstacle-avoidance systems. Bumpers are operational in this mode only if bumpStall is turned on. Consequently, you can cause serious damage and injury in Joy-Only drive mode, so restrict access to qualified personnel and use it with extreme caution.

WARNING: Use Joy-Only drive mode carefully.

ARAM's collision-avoidance behaviors are not active in this mode.

Step 2: SBC and LRF Power

If the Core microcontroller starts normally, it automatically supplies power to the Core SBC, to the LRF, and to the allocated auxiliary power ports as specified by their FLASH power parameters. The SBC Power status LED lights up if power gets supplied to the SBC, and the 'C', 'L', angle brackets, and asterisk characters appear on the LCD accessory display. The Core SBC starts its OS, which in turn automatically starts ARAM and the web-server services needed by SetNetGo. Note that ARAM starts up with options according to a list of optional arguments that are editable through SetNetGo (See "Ethernet Connections").

Step 3: LRF and Client Connections

After startup, ARAM makes serial connections with the Core microcontroller and with the LRF. If either connection fails, the switched power port specified by FaultLampPort turns on continuously, LRF power goes OFF, but the SBC status LED remains lit.

5.2 Shutdown Logic

Under normal circumstances, use MobileEyes **Tools > Robot Tools > Shutdown** or the ARCL shutdown command to gracefully shut down the Core. Because its FLASH is protected, removing power manually will not harm the system. In addition, the microcontroller firmware contains voltage and temperature monitors that can activate system-wide power shutdown.

The controller firmware automatically initiates a shutdown sequence if source power at J28 equals or falls below shutdownVolts or if the Core's internal temperature sensors detect overheating. The Core initially waits a short period of time for the condition to improve, and returns to normal operation if the temperature and voltage return to an acceptable state. If the condition does not improve, the Core initiates a non-recoverable shutdown sequence. If the SBC and ARAM are both running, the event will be logged in the ARAM log files. If allocated, the faultLampPort indicated is lit during shutdown.

5.3 Ethernet Connections

Communicate with the Core SBC for operation and SetNetGo-mediated configuration and maintenance through its Ethernet ports. The Core SBC supports two Ethernet ports: one wireless (antennas) and one 10/100 RJ-45 wired (J27). You do not need wireless to operate the Core. You can connect MobileEyes and MobilePlanner off-board clients directly through the wired interface. In fact, first-time configuration of the Core requires a wired Ethernet connection.

The wired port has a fixed IP address: 1.2.3.4, which you may access from an external PC Ethernet port set to any class-C IP address 1.2.3.xxx (where xxx is not 0, 4 or 255). Use a standard CAT-5E cable.

Connect with SetNetGo from an off-board PC web browser through the wired Ethernet connection: https://1.2.3.4. Use SetNetGo's **Networking > General** form to set up wireless network connectivity.

5.4 Access Passwords

Except when connecting through the https://1.2.3.4 port, all Ethernet connections support the use of access control. For SetNetGo, the username is "admin", but you must provide a password when enabling it on the wireless Ethernet interface. For ARCL, you must always provide a password to enable it. For MobileEyes and MobilePlanner connections, you can enable access control by checking the "-enableAccounts" flag in the ARAM section of SetNetGo. If you do so, you can configure and modify the usernames, passwords, and access privileges using the UserAccounts section of SetNetGo.

Default usernames/passwords for MobileEyes and MobilePlanner are user/user and administrator/administrator.

5.5 ARAM Configuration

ARAM automatically starts up soon after the Core SBC starts up. The software accepts command-line arguments that alter its startup behavior, mostly to specify device and service connections. Use SetNetGo's Configuration page checkboxes to select options, and the custom startup argument text box for special features. Click Apply, and ARAM restarts to apply the options.

5.6 Core Status

ARAM will not start or restart if there is a problem. Accordingly, it is useful to monitor the Core's status AramLog, especially after restarting ARAM. Review it and other logs, such as ConfigLog and Dmesg through SetNetGo's Status feature. The main AramLog tracks the progress of your Motivity robot, too.

Use SetNetGo's **Configuration > Download Files** feature to download all map and ARAM configuration files that have been saved on the robot. Alternatively, use SetNetGo's **Configuration > DebugInfo** feature to download a file containing the status logs and information about the Core that can be useful for troubleshooting. Provide this file when requesting support from Adept.

5.7 Core Operating Parameters

Three sets of parameter values affect the operation of your Motivity robot through ARAM and its associated microcontroller. The lowest-level settings are found in:

- the aramConfig.txt file
- the robot's parameters (".p") file
- the microcontroller's FLASH

Some parameters are redundant, and follow a specific order of precedence:

- parameter values found in the aramConfig.txt take precedence over all others
- the robot's .p parameters take precedence over microcontroller FLASH values

In the aramConfig.txt and the robot's .p files, a blank or null value means the parameter label is a placeholder; it does not set the parameter to value 0 and has no effect on operation. Blank or missing parameters are resolved by system defaults.

aramConfig.txt is one of the files that you may download and examine with SetNetGo's Download Files feature or from the MobileEyes Robot Configuration tool (see the next section for details). It mostly affects ARAM and, therefore, the highest levels of mobility and functionality of your Motivity robot.

The microcontroller FLASH parameters are set and managed with a separate tool, COREcf. Your robot's .p parameters file is accessible via SetNetGo's **Configuration > Upload Parameters File**. The filename must be identical (although case-insensitive) with the Subtype value found in your Core microcontroller's FLASH parameters, and with a .p file-extension suffix. For example, if the Subtype FLASH parameter is "Platform", the uploaded parameters file must be "platform.p." If you change the subtype, upload a correspondingly-named parameters file, too.

Always use the MobileEyes **Tools > Robot Configuration** dialog to manage ARAM's configuration parameters. Applied changes immediately take effect. Manually edited aramConfig.txt parameters only get read on ARAM startup.

5.8 MobileEyes and MobilePlanner

Use the networked GUI application MobileEyes to not only monitor and control your Core-enabled Motivity robot, but also to manage its many operating parameters. Use MobilePlanner to create and edit maps, including embedded goals, patrol routes, and advanced robotic tasks.

Installation

MobileEyes and MobilePlanner are Windows PC applications that run offboard the Motivity robot. When installed, by default the applications get stored in your PC's file system in their respectively-named directories. The application executable is in a bin\ subdirectory, and also accessible from Windows' **Start > All Programs**.

Connecting

MobileEyes and MobilePlanner communicate with ARAM through the Core's Ethernet, either wirelessly or by direct connection. Their connection dialogs each ask for a Robot Server host-name or IP address (1.2.3.4, for example, if you are connecting through the wire).

If connecting to the Core from another address, you may need to input a username and password. Manage network access, usernames, and passwords through SetNetGo. By default, administrators have special privileges for creating and managing maps and configurations. Other users have restricted access for operation of the robot.

5.9 Operating MobileEyes

Once connected, MobileEyes displays the robot's map, sensor readings, and other operating conditions, and locates the robot within the map. Use the Map and View options to customize the display. For instance, select **Map > Laser** to have LRF range readings appear as blue dots in the map display.

Robot Configuration

Select the MobileEyes **Tools > Robot Configuration** menu option, and in the dialog verify that the map filename (Sections:Files) corresponds to the real workspace. You also might take the opportunity now to review and modify other details of the robot's configuration, such as those governing its automated recharging behaviors, localization parameters, and others.

At the bottom of the MobileEyes configuration dialog, the File Option menu lets you manage several configurations for the same robot. Use its Save option to extract the current operating parameters from the Core and save them in an aramConfig.txt-like file that you name and store on your MobileEyes PC or other accessible file system. You can Load that file into the MobileEyes configuration dialog File Option menu, if needed.

Note that no parameter changes are applied on the Core, including those loaded from a local file, until you click Apply from the MobileEyes configuration dialog. Changes made from MobileEyes are saved on the robot immediately but don't take effect until the next time the robot is idle. ARAM also saves a copy of the current parameters onboard in the aramConfig.txt file.

Localization

Once you have verified the map, verify that the robot is properly located in the map relative to its actual position in real space. If it's not where it should be, it won't work.

To relocate the robot, scroll the MobileEyes map (right mouse button; you may have to relieve the Map:Center on Robot option) to bring the true location into display, then select Localize to Point. Place the special mouse pointer where the robot really is in the map, then left-click and drag the mouse in the direction the robot is facing before releasing the mouse button.

When successfully localized, your Motivity robot's sensor reflections should match the map's features. It's best to achieve a LocScore of 80% or better. The robot periodically refines its position while navigating by a statistical comparison of its sensor readings to the map. It remembers where it was when you turned off its power, so you will rarely have to manually relocalize.

The robot will automatically localize whenever it has moved TriggerDistance or TriggerAngle, or when TriggerTime has elapsed (refer to Parameters and Interactions, which follows).

Navigation

Click the robot-like button just to the left of the MobileEyes Status textbox or select Robot:Arrow Key Drive to engage manual driving mode. Then click the corresponding toolbar buttons, press and hold the keyboard arrow keys, or operate a joystick that you have connected to your MobileEyes PC in order to manually drive and turn the robot at its adjusted speeds (**Robot Configuration > Teleop** settings or from the slider on the menu bar).

Normally, manual drive is in SAFE mode with automatic obstacle avoidance. Select and drive unsafely temporarily, for example to extricate a trapped robot. The MobileEyes map background turns yellow to indicate UNSAFE mode.

Alternatively, to have ARAM plan a path and automatically drive the Motivity robot to some place in the map, select Send Robot and click the special mouse pointer at the destination in the map. (MobileEyes will tell you if that place is inaccessible.)

Select **View > Routes/Goals** to display the MobileEyes Routes and All Goals panes and their contents. Double-click a goal to send the robot to that particular place in the map, or double-click a route to have it travel from goal to goal. In either case, your Motivity robot performs the set of tasks that you associated with a goal or route with MobilePlanner.

5.10 Parameters and Interactions

There are two classes of parameters that interact: those that control motion and those that affect CPU usage.

Robot Motion

There are six main motion-control parameters:

Parameter	Function	Units	Default
TransVelMax	Maximum Translational Velocity	mm/sec	2000
TransAccel	Maximum Translational Acceleration	mm/sec^2	650
TransDecel	Maximum Translational Deceleration	mm/sec^2	650
RotVelMax	Maximum Rotational Velocity	deg/sec.	50
RotAccel	Maximum Rotational Acceleration	deg/sec^2	600
RotDecel	Maximum Rotational Deceleration	deg/sec^2	600

CPU Usage

The effects of the parameters that influence CPU usage are complex, although there are only two CPU-intensive parts to ARAM:

- Localization (Localization Settings in the MobileEyes **Tools > Robot Configuration** dialog) and

- Path Planning (Path Planning Settings).

Localization keeps track of the robot's place in its operating environment, including correcting for errors in the robot's odometry. Path planning figures out how to get from one place to another.

Localization Effects on CPU Usage

Parameter	Description
NumSamples	Number of hypotheses used to resolve localization
KMmPerMm	Expected translation error per mm of travel
KDegPerDeg	Expected rotation error per degree of rotation
KDegPerMm	Expected rotation error per mm of travel
TriggerDistance	Maximum distance traveled before localizing
TriggerAngle	Maximum angle turned before localizing
TriggerTime	Maximum time between localizations, if enabled
TriggerTimeEnabled	True enables TriggerTime
IdleTimeTriggerX	Distance to search in X after TriggerTime elapses
IdleTimeTriggerY	Distance to search in Y after TriggerTime elapses
IdleTimeTriggerTh	Angle to search after TriggerTime elapses
PassThreshold	Minimum LocScore to accept localization
LostThresholdDistance	Maximum distance to search before reporting Lost
RecoverOnFail	True enables expanded search after failed localization
FailedX	Distance to search in X after failed localization
FailedY	Distance to search in Y after failed localization
FailedTh	Angle to search after failed localization

NumSamples

NumSamples specifies how many different hypotheses ARAM uses to resolve localization. The number scales linearly, so increasing NumSamples from 2000 to 3000 takes 1.5 times more processing.

Expected Error Parameters (K-Values)

KMmPerMm, KDegPerDeg, and KDegPerMm set the expected error of the robot's odometry. The higher the values, the more ARAM expects the odometry to be in error.

KMmPerMm and KDegPerDeg are, respectively, the expected translation and rotation odometry error (how many millimeters of error are expected per millimeter of travel and how many degrees error per degree of travel). KDegPerMm is how many degrees of error are expected in one mm of travel.

For example, if:

- KMmPerMm is 0.04
- KDegPerDeg is 0.04, and
- KDegPerMM is 0.0025

when the robot travels a meter forward without turning, ARAM computes its odometry to be in error by up to 40 mm (0.04 (KMmPerMm) * 1000 (mm traveled)) and 2.5 degrees (0.04 (KDegPerDeg) * 0 (degrees turned) + 0.0025 (KDegPerMM) * 1000 (mm traveled)).

ARAM puts NumSamples readings in this search area, and selects the best match. This is why these values affect how large you should make NumSamples.

If you have the K-values too high, the samples will be too spread out to be able to localize, while K-values too low may not search a large enough area to find the robot's actual position in the map.

TriggerDistance and TriggerAngle

ARAM tries to localize whenever the robot moves TriggerDistance and/or turns TriggerAngle. The computations take a lot of the CPU time, so if you have the value of NumSamples set too high, the robot will be forced to localize less often. It's usually better to localize less precisely more often.

TriggerTime Parameters

Localization also happens at TriggerTime since the last localization if TriggerTimeEnabled is true. TriggerTime is used when the robot is stopped or moving slowly. This localization event uses IdleTimeTriggerX, IdleTimeTriggerY, and IdleTimeTriggerTh to determine the localization search area.

IdleTimeTriggerX and IdleTimeTriggerY should generally be the same. IdleTimeTriggerTh is the angle, relative to the robot's current heading, that Motivity will search for a localization.

Failed Localization Parameters

Just for completeness, if localization fails, typically when LocScore goes below PassThreshold and stays low for a while, and RecoverOnFail is true, then localization will spread samples out FailedX-, FailedY- and FailedTh-wide from the last good position to try to find the robot's location.

FailedX and FailedY should generally be the same. FailedTh is the angle, relative to the robot's current heading, that Motivity will search for a localization.

Path-Planning Effects on CPU Usage

Path planning also affects CPU usage. The most important path-planning parameter is PlanRes. It sets the resolution of the grid ARAM uses for path planning. PlanRes, like GridRes, subdivides a square area, so increasing the resolution (decreasing the PlanRes value) increases the processing load by the square of the difference. If you reduce PlanRes from 100 to 50, you make the CPU work four times harder; 100 to 25 results in a 16-fold increase in processing time.

ARAM's path planner computes one plan from start to finish at the outset of going to a goal or point in the map. Along the way to the goal, ARAM frequently re-plans at each local section. An important factor, in terms of CPU usage, is how large an area gets re-planned each cycle. The area is set by CollisionRange (how far ahead to look for collisions) plus FreeSpacing (how far to stay away from obstacles). The additive value (CollisionRange + FreeSpacing) specifies a square, so doubling the sum of the two increases CPU usage by a factor of four.

The PlanEverytime parameter, if true, causes ARAM's path planner to re-plan a local path every 100 ms. When false, ARAM only re-plans when encountering obstacles.

Performance and Conflicts

Parameter	Description
FreeSpacing	How far to stay away from obstacles
CollisionRange	How far ahead to look for collisions

In terms of accuracy, changes to several goal-related parameters affect how smoothly and accurately the robot moves.

Increasing FreeSpacing and/or CollisionRange gives ARAM more opportunity for more gentle curves, but at the expense of more CPU usage.

There are three competing robot performance-related factors: speed, smoothness, and accuracy. The more you try to improve one factor, the greater potential it has to affect the other factors detrimentally. Generally you need to work on all three factors together, rather than just one or two of them. See also μARCS on page 43.

5.11 Core Maintenance

The only user-serviceable part on the Motivity Core is the fan filter. The replacement interval is dependent on the environment in which the robot is operating. This will typically range from replacement every three months, in very dirty environments, to replacement every year.

Chapter 6: µARCS

The Core's microcontroller runs µARCS firmware that drives the robot. It also contains a set of low-level operating parameters in FLASH ROM.

The latest version of µARCS comes installed in your new Core. Its FLASH-based operating parameters are preset to defaults that may or may not accurately reflect your robot and its environment. Also, periodically, Adept MobileRobots issues updates to the firmware. Use the COREcf upload and configuration tool to update µARCS and manage its operating parameters.

6.1 Core Maintenance Mode

The COREcf program runs on an off-board Windows PC and connects with the microcontroller through the Core's SYSTEM serial port. Attach a straight-through serial cable between your PC and the DSUB-9 SYSTEM serial port connector on the Core.

You have several ways in which to put the microcontroller into µARCS Maintenance mode without starting the onboard SBC (ARAM interferes with COREcf). One way is to start up the Core in Joystick mode. Another is to connect with SetNetGo and stop ARAM. Or use LCD Interactive mode and select the option to enter Maintenance mode.

The best way, however, is to power down the Core, connect your PC to the Core's SYSTEM port, start the COREcf utility, then apply power to the Core. If, for any other reason, the Core SBC doesn't start up (Joystick mode and having the SYSTEM port active on power-up prevents SBC startup), the microcontroller automatically starts up in Maintenance-Ready mode.

6.2 COREcf Installation and Startup

The COREcf firmware update and configuration program is the comprehensive manager for your Core microcontroller and its FLASH-based operating parameters. The distribution archive for the software is named simply COREV_v (V and v are the version major and minor numbers, respectively, such as 4_2), with a ".exe" suffix.

Install the utility on a Windows PC by double-clicking the distribution software's onscreen icon or otherwise executing the self-extracting, self-installing package.

The expanded archive creates a COREV_v/ directory in the selected Windows path and stores the µARCS and COREcf software there.

WARNING: Activate the E-Stop or otherwise disengage your robot's motors before starting COREcf Maintenance mode.

The COREcf program is a text-based console application. It runs in two stages: startup followed by interactive mode. Optionally, start COREcf with one or more command-line arguments (see the list of options in the following table).

You can double-click the COREcf icon, to automatically open a console window and start the program without any options. To start with command-line options, Run the program from the

Start menu, or run cmd from the Start menu, navigate to the CORE directory, and start COREcf with the options.

For example (after invoking the command (cmd) window):

> C:\> cd Program Files\MobileRobots\CORE
>
> C:\Program Files\MobileRobots\CORE\> COREcf –rp COM3

This will specify COM3 as the serial port for your PC.

When successfully connected with the Core microcontroller after startup, COREcf automatically retrieves the FLASH-stored operating parameters and enters interactive mode. You also can operate many of COREcf's interactive features without a connection ('-n' command-line startup option), such as to maintain disk-based copies of your robot's operating parameters.

6.1COREcf startup options

Key	Argument	Description
-b	command arguments	Batch mode executes a list of COREcf interactive-like mode commands with arguments, then exits automatically.
-u	motfile	Automatically upload a µARCS (.mot) motfile after connecting with the controller.
-l	paramsfile	Load the disk-stored params (.rop) file instead of the robot's copy.
-n	none	Don't automatically connect with controller.
-rp	serial-device	Use specified serial port for connection.
-rb	baudrate	Specify the port connection baud rate.
-s	paramsfile	On exit from COREcf, automatically save the current parameter values to the named .rop paramsfile.

6.3 Exiting COREcf Maintenance Mode

What happens with your Motivity robot after you exit COREcf and its associated Maintenance mode with the Core controller depends on how you started up and how you exit. First, either use the '-b' argument on COREcf startup, in which case the program exits immediately after completing its startup arguments, or type the quit interactive key command. Either of these methods gracefully exits Maintenance mode and restarts the microcontroller. If you had stopped ARAM with SetNetGo, you'll need to restart it to get your robot moving again. Otherwise, the robot should behave as if you had just applied power to the Core.

Alternatively, if you ungracefully exit COREcf (Control-C out of the program, for example, or shut off your PC), your Core microcontroller will remain in Maintenance mode and will not restart or operate normally. You have two options: either cycle power or restart COREcf and exit gracefully.

6.4 Updating µARCS

Updated versions of the Core microcontroller's firmware get distributed as Motorola®-encoded text files, typically named µARCSV_v.mot and stored inside the COREV_v directory of the new distribution. Updating µARCS does not alter the controller's operating FLASH parameters. Please carefully read the README text file, which accompanies the update, before updating the µARCS firmware.

You have two ways to upload the new firmware: on startup, or interactively, after you have started COREcf. Either method lets you examine and alter the FLASH parameters, unless you include the '-b' argument in the startup command line, in which case the program exits once its command line tasks are completed.

To update µARCS to rev 4_2 through your PC's COM3 port and exit immediately:

Invoke the command (cmd) window, then enter:

C:\> cd Program Files\MobileRobots\CORE

C:\Program Files\MobileRobots\CORE\> COREcf –rp COM3 –u µARCS4_2.mot –b

Use the 'u' option and provide a pathname for the new firmware when in interactive mode.

6.5 Configuring the Controller

Your Core has several parameters stored in FLASH that µARCS and ARAM use to configure its connections and auxiliary attachments and to uniquely identify your robot. For instance, the default maximum translation velocity for your Motivity robot is stored in the TransVelMax FLASH parameter. Its value takes effect when you first start up your robot, but may be changed temporarily by ARAM. Use COREcf's batch or interactive modes to modify the operating parameters and, therefore, your robot's default operating characteristics.

Start up COREcf as described in the previous section. The program normally downloads the set of operating parameters from your Core controller's FLASH for your review and modification. Or, you may load a disk-stored version of those parameters.

Interactive Commands

To operate COREcf in interactive mode, type a keyword at its command line. For instance, to review the list of the common µARCS FLASH variables, type 'v' or 'view' followed by Enter. Type 'p' to review the power-related parameters. Each successive Enter will display additional variable keywords and current values. Type '?' or 'help' to see a list of COREcf interactive commands.

To review the value of a particular parameter, enter its keyword name (case insensitive) and press Enter.

Changing Parameters

Provide an argument after the parameter's keyword, separated by a space, to change its value. That value may be a string (no quotes or spaces) or a decimal or hexadecimal ("0xN") number. For example, to change the WatchDog timeout to four seconds, type:

> watchdog 4000

or

> watchdog 0xfa0

See the control command and parameter tables following for a full description of COREcf keywords and values.

6.6 Save Your Work

While changing parameter values in COREcf interactive mode, you are editing a temporary copy on your PC. Your changes are not put into effect in your Core microcontroller's FLASH until you explicitly save them.

Also use the COREcf save command to save a copy of the parameters to a disk file for later upload. We strongly recommend that you save each version of your Core's parameter values to disk for later retrieval should your microcontroller get damaged or its FLASH inadvertently erased. Default parameter files come with each CORE distribution, but it is tedious to reconstruct an individual robot's unique configuration.

6.2COREcf control commands

Command	Description
<keyword> <value>	Keyword alone: display current, edited value. Add argument to change current value.
v or view	Display common FLASH parameters.
p or power	Display power-related FLASH parameters.
u or upload <motfile>	Upload specified µARCS image (.mot) file to the controller.
r or restore <paramsfile>	Restore parameters to values currently stored in FLASH or, if given, from a paramsfile (.rop) on disk.
save <paramsfile>	Save current edited values to FLASH or save current edited values to paramsfile (.rop) on disk for later reference.
q or quit	Exit COREcf.
? or help	Display these commands and descriptions.

Keyword	Data	Default	Description
name	str	ADAM	Unique name for your robot. Maximum of 20 characters, no spaces.
sernum	str	#######	Serial number of the robot.
ticksmm	int	150	Encoder ticks/mm: (4 x ticks per rev x gear-ratio)/(wheel_ diameter x Π).
revCount	int	38173	The number of encoder ticks for a 360- degree revolution of the robot.
driftFactor	int	0	Value in 1/8192 increments to be added or subtracted from the left encoder ticks in order to compensate for tire differences.
lowBattery	int	230	In 1/10 volts; µARCS controller alarm activated when- source voltage falls below this value.
shutdownVolts	int	220	In 1/10 volts; µARCS controller disconnects client and sig- nals onboard SBC to shut down when source voltage falls below this value.
gyroCW	int	970	Firmware gyro clockwise calibration value.
gyroCCW	int	970	Firmware gyro counter-clockwise calibration value.
hasBrakes	byte	1	1 if your robot has brakes; 0 if not.
frontBumps	byte	1	Number of front bumper segments.
rearBumps	byte	1	Number of rear bumper segments.
invertBump	byte	3	0=none; 1=front; 2=rear; or 3=invert both; affects STALL bits in std. SIP only.
bumpStall	byte	0	0=enable all bump stall; 1=enable rear; 2=enable front; 3=disable both front and rear bump stalls.
StallOnBumpHit	byte		0=decelerate;1=stall. Both disable motor amplifiers.
stallVal	int	494	Maximum PWM before stall. If > PwmMax, never.
stallCount	int	500	Ms time motors disabled after a stall for recovery.
pwmmax	int	495	Maximum motor PWM (500 maximum).
rotveltop	int	150	Maximum rotation velocity; deg/sec.
transveltop	int	2000	Maximum translation speed; mm/sec.
rotacctop	int	600	Maximum rotation (de)acceleration; deg/sec2.
transacctop	int	650	Maximum translation (de)acceleration; mm/sec2.
rotvelmax	int	50	Max rotation speed; deg/sec.
transvelmax	int	1900	Max translation speed; mm/sec.
rotacc	int	100	Rotation acceleration; deg/sec2.
rotdecel	int	150	Rotation deceleration; deg/sec2.

Keyword	Data	Default	Description
rotkp	int	10	Proportional PID for rotation.
rotkv	int	10	Differential PID for rotation.
rotki	int	0	Integral PID for rotation.
transacc	int	650	Translation acceleration; mm/sec2.
transdecel	int	650	Translation deceleration; mm/sec2.
transkp	int	10	Proportional PID for translation.
transkv	int	10	Differential PID for translation.
transki	int	0	Integral PID for translation.
joystick	byte	1	Joystick type: 0=analog, 1=inductive.
joyvelmax	int	1000	Joydrive maximum translation velocity.
joyrvelmax	int	50	Joydrive maximum rotation velocity.

6.3Power-related Core Parameters with Typical Values ('p' command)

Keyword	Data	Default	Description
motionLampPort	byte	13	Power port for motion-related lamp.
faultLampPort	byte	12	SBC-controller connection and E-Stop indicator.
battLampPort	byte	14	Power port for low battery voltage and high-temperature sensor indicator.
aux5VPort	byte	0	Power port associated the aux 5 VDC power.
aux5V	byte	0	Power 0=off, 1=on at startup, state persists through multiple connection sessions; 2=off, state forced after each session, 3=on, state forced after each session; 4=on when connected, off when not connected.
aux12VPort	byte	0	Power port associated with the aux 12 VDC power.
aux12V	byte	0	See aux5V.
aux24VPort	byte	0	Power port associated with the aux 24 VDC power.
aux24V	byte	0	See aux5V.

6.7 Key Parameters

PIDs

The FLASH PID parameters affect the smoothness and responsiveness of the µARCS classic proportional-integral-differential motor-control loop. The Proportional PID values for translation (transKp) and rotation (rotKp) control the responsiveness of your mobile robot. Lower values make for a slower system; higher values make the motion "zippier", but can lead to overshoot and oscillation.

The Derivative PIDs (transKv and rotKv) dampen oscillation and overshoot. Increasing values gives better control of oscillation and overshoot, but they also make the robot's movements more sluggish.

The Integral PIDs (transKi and rotKi) adjust residual error in turning and velocity. Higher values make the robot correct increasingly smaller errors between its desired and actual angular position and speed.

DriftFactor, TicksMM, and RevCount

µARCS uses the TicksMM and RevCount parameters to convert the robot-independent speed and rotation commands—typically expressed in millimeters or degrees, respectively—into robot-dependent units. It uses DriftFactor to compensate for diameter differences between left and right tires.

The TicksMM value is the number of encoder pulses, or ticks, per millimeter of travel. The value is dependent upon the wheel encoder's resolution, the motor-to-wheel gear ratio, and the wheel's diameter. For a 1000-line quadrature-encoder, with a 29:1 gear reduction, and 200 mm diameter wheels, the solution is:

(4 x 1000 x 29) / (pi x 200) = 185 ticks/mm

The RevCount value is the number of encoder ticks for one revolution of the robot. RevCount can change based on changes in tire wear, payload, track width, and operating surface.

DriftFactor is a value in 1/8192 units that gets added or subtracted from the left-wheel encoder count at each motor cycle every 5 ms. In doing so, it compensates for tire differences and thereby straightens the robot's translation forward and backward.

The TicksMM and RevCount parameters affect the conversion of your motion command arguments into robot-dependent values used by µARCS. µARCS also uses TicksMM and RevCount to convert its internal measures into robot-independent position, heading, and velocity values reported back, such as X-Pos and Th. In most situations, the ".p" file parameters will have many conversion factors, like DistConvFactor, set to 1.0.

StallVal and StallCount

A µARCS stall monitor maintains a running average of PWM values for each motor over a 500 ms integration period. PWM values get added to the sum if the wheel speed is below 100 mm/sec. The average is then compared with the StallVal FLASH value. If it exceeds that value—in other words, if the motors are being given lots of power, but are barely moving, if at all—a stall happens. When stalled, µARCS removes power and the motors relax for the StallWait period, after which µARCS re-applies power. If brakes have been associated with the ENABLE line, they will be toggled when power is removed from the motors.

GyroCW, GyroCCW, and GyroRateLimit

A Core internal gyroscope refines the rotational component of your Motivity robot's odometry. The gyroscope is attached to an analog port on the Core microcontroller and provides a maximum 300 degrees-per-second rotational rate reading at 20 ms intervals.

µARCS firmware automatically uses the gyro to correct rotational errors. Set gyroCW and gyroCCW calibration values to engage the firmware-based gyro.

Bumpers

Use the BumpStall FLASH parameter to set the default for your Motivity robot's behavior when any of its front and/or rear bumpers get triggered. Normally, BumpStall is engaged for both front and rear (default value of 0). Reset it to 3 to disengage bump stalls altogether; 1 to trigger stalls only when the rear bumpers engage; or 2 for front bumps only. Your robot's BumpStall behavior reverts to the FLASH default on reset and upon disconnection from ARAM.

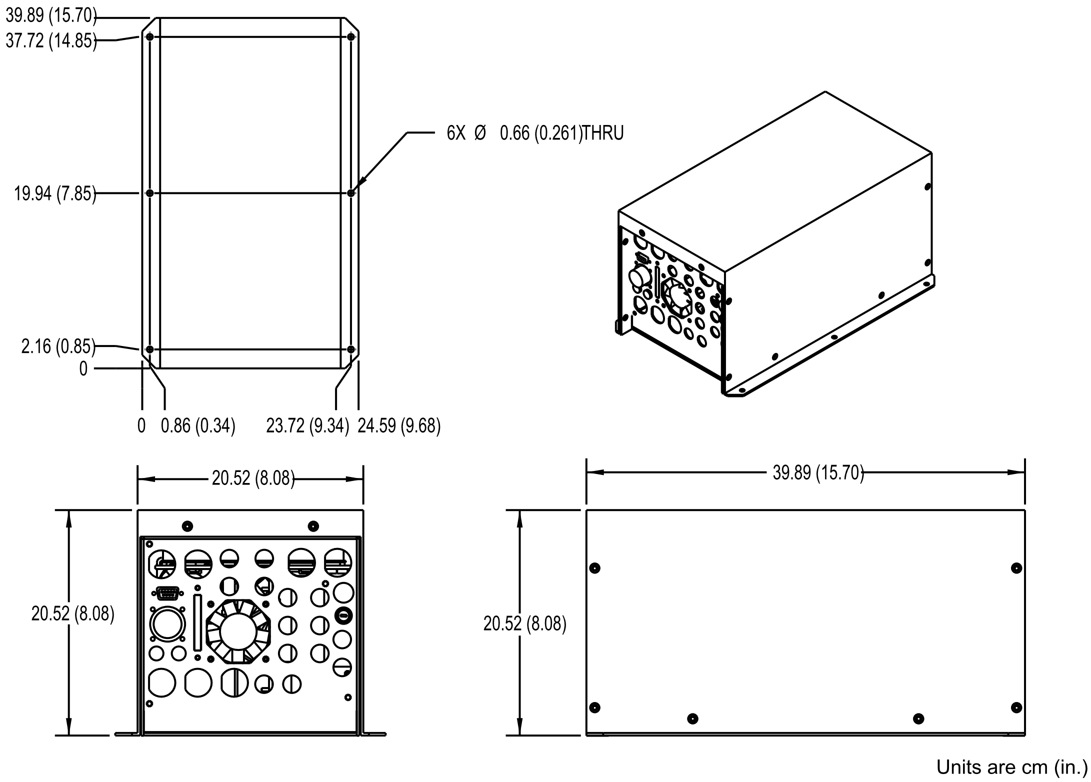
μARCS implements three FLASH parameters that specify states and numbers of front and rear bumper segments. Set the FrontBumps and RearBumps parameters to the number of bumper segments for the front and rear bumpers, respectively. The number of segments is used to isolate the bumper bits, if any, so that a triggered bumper event is reported correctly in the STALL value ARAM uses to detect and respond to stalls. Use the InvertBump FLASH parameter's value to invert those bumper-related STALL values.

StallOnBumpHit affects how the motors behave when a bumper has been triggered. If set to 1, the motors will stall and disable the motor amplifiers. If set to 0, the velocity commands will decelerate at transVelTop and then disable the motor amplifiers. If you have a brake system that uses the ENABLE or /ENABLE signals, we suggest setting StallOnBumpHit to 1.

If, for any reason, you remove a bumper from your robot, you MUST reset the associated FrontBumps or RearBumps FLASH value. Otherwise, the robot will stall incessantly and won't drive.

Chapter 7: Technical Specifications

7.1 Dimension Drawings



7.1Arc Core Dimensions

7.2 Motivity Core Specifications

Description	Specification
Physical (electronics may be re-packaged if necessary)	
Length	39.89 cm (15.70 in.)
Width	20.52 cm (8.08 in.)
Height	20.52 cm (8.08 in.)
Weight	6.6 kg (14.6 lb)
Environment/Rating	
Temperature Range	5 - 50° C (41 - 122° F) ambient at air-circulation inlet (automatic Core shutdown @ 50° C)
IP Rating	IP-51
Power	
Input	24 VDC ± 10% (65 W typical)
Auxiliary	4 @ 12 VDC regulated 5 A max. software switched
	3 @ 5 VDC regulated 5 A max. software switched
	3 @ 24 VDC ± 10% (source) 5 A max. software switched
Sensors	
	Internal angular rate-sensing gyroscope
	External 180º scanning laser range-finder (SICK LMS200)
	2 (front/rear) x 7-bit segmented bumper inputs
Software	
Onboard	µARCS microcontroller operating system
	SetNetGo network setup and maintenance tool
	ARAM Core robot-control, localization and navigation
Offboard	COREcf microcontroller FLASH manager
	MobilePlanner workspace map editing software
	MobileEyes robot command, configuration and monitor GUI
Navigation and Delivery	
	Autonomous, guarded or unguarded teleoperation drive

Description	Specification
	Localization based on existing features of the environment
Travel repeatability	± 150 mm
Docking accuracy	± 15 mm
Goal repeatability	± 100 mm
Electronics and Interfaces	
Embedded (EBX) computer	Pentium M-class computer with FLASH storage
Robot-drive microcontroller	Embedded 32-bit RISC robot-drive microcontroller
Joystick	2-axis analog with analog throttle, engage and mark pushbutton inputs
Microcontroller maintenance	SYSTEM serial port
Motor-control output ports	Differential-drive (left and right), signals inverted
Motor-power PWM signals	20 KHz digital (0, 5 VDC)
Drive forward-reverse direction signals	Digital (0, 5 VDC)
Motor-enable signal	Digital (0, 5 VDC)
Quadrature encoder	Left and right signal input ports
Ethernet port	10BaseT/100BaseTX CAT-5E
Antenna ports	Two wireless Ethernet (802.11b)
Laser range finder signal and power	SICK LMS200
Digital bumper signal inputs	2x7 digital (0, 5 VDC; normally high)
Analog inputs	8-channel, software-selectable
I/O ports	32-bits digital (0,5 VDC); software-selectable input or output
Support for LCD	4x20 character microcontroller signal and power

5960 Inglewood Drive
Pleasanton, CA 94588
925•245•3400

PN: 10757-000 Rev A