



# **AX2550**

## **Dual Channel High Power Digital Motor Controller**

### **User's Manual**



v1.3, August 25, 2003

visit [www.roboteq.com](http://www.roboteq.com) to download the latest revision of this manual

©Copyright 2003 Roboteq, Inc.

---

---

## Revision History

Date	Version	Changes
August 25, 2003	1.3	Added CLosed Loop Speed mode Added Data Logging support Removed RC monitoring
August 15, 2003	1.2	Modified to cover AX2550 controller design Changed Power Connection section
April 15, 2003	1.1	Added analog mode section Added position mode section Added RCRC monitoring feature Updated Roborun utility section Modified RS232 watchdog
March 15, 2003	1.0	Initial Release

The information contained in this manual is believed to be accurate and reliable. However, it may contain errors that were not noticed at time of publication. User's are expected to perform their own product validation and not rely solely on data contained in this manual.

Revision History   **2**

**SECTION 1**

AX2500 Quick Start	<b>9</b>
What you will need	<b>9</b>
Locating Switches, Connectors and Wires	<b>10</b>
Connecting to the Batteries and Motors	<b>11</b>
Connecting the R/C Radio	<b>12</b>
Powering On the Controller	<b>12</b>
Button Operation	<b>13</b>
Default Controller Configuration	<b>14</b>
Checking and Changing Configurations	<b>14</b>
Connecting the controller to your PC using Roborun	<b>15</b>
Obtaining the Controller's Software Revision Number	<b>16</b>
Exploring further	<b>17</b>

**SECTION 2**

AX2500 Digital Speed Controller Overview	<b>19</b>
Product Description	<b>19</b>
Technical features	<b>20</b>

**SECTION 3**

Connecting Power and Motors to the Controller	<b>23</b>
Connecting Power	<b>23</b>
Controller Power	<b>24</b>
Powering the Controller from a Separate Battery	<b>25</b>
Powering the Controller using the Motor Batteries	<b>25</b>
Motor Power	<b>26</b>
Power Fuses	<b>27</b>
Wire Length Limits	<b>28</b>
Electrical Noise Reduction Techniques	<b>28</b>
Power Regeneration Considerations	<b>28</b>

**SECTION 4**

Connecting Sensors and Actuators to Input/Outputs	<b>31</b>
AX2500 Connections	<b>31</b>
AX2500's Inputs and Outputs	<b>33</b>
I/O List and Pin Assignment	<b>34</b>
Connecting devices to Output C	<b>35</b>
Connecting devices to Output D	<b>36</b>
Connecting Switches or Devices to Input E	<b>36</b>
Connecting Switches or Devices to Input F	<b>37</b>
Connecting Switches or Devices to EStop/Invert Input	<b>38</b>
Connecting Position Potentiometers to Analog Inputs	<b>39</b>
Connecting Tachometer to Analog Inputs	<b>40</b>

Connecting External Thermistor to Analog Inputs	41
Using the Analog Inputs to Monitor External Voltages	42
Connecting User Devices to Analog Inputs	43

## **SECTION 5**

General Operation	45
Basic Operation	45
Input Command Modes	45
Selecting the Motor Control Modes	46
Open Loop, Separate Speed Control	46
Open Loop, Mixed Speed Control	46
Closed Loop Speed Control	47
Close Loop Position Control	47
Continuous and Extended Current Limitation	48
Peak Current Protection	49
Programmable Acceleration	49
Command Control Curves	50
Left / Right Tuning Adjustment	51
Emergency Shut Down Using Controller Switches	52
Emergency Stop using External Switch	53
Inverted Operation	53
Controller Overheat Condition	53
External Devices Overheat Monitoring	54
Special Use of Accessory Digital Inputs	54
Using the Inputs to Activate the Buffered Output	54
Using the Inputs to turn Off/On the Power MOSFET transistors	55
Self-Test Mode	55

## **SECTION 6**

Normal and Fault Condition LED Messages	57
Use of the LED Display	57
Motor Direction Status	58
Fault Messages	59
No Control	59
Emergency Stop	60
Heat	60
Self-Test Display	60

## **SECTION 7**

R/C Operation	63
Mode Description	63
Selecting the R/C Input Mode	64
Typical Wiring	64
Connector I/O Pin Assignment (R/C Mode)	65

R/C Input Circuit Description	66
Cabling to R/C Receiver using Full Opto-Isolation	67
Cabling to R/C Receiver with Partial Opto-Isolation	68
Powering the Radio from the controller	69
Operating the Controller in R/C mode	70
Reception Watchdog	71
R/C Transmitter/Receiver Quality Considerations	71
Joystick Deadband Programming	72
Command Control Curves	73
Left/Right Tuning Adjustment	73
Joystick Calibration	73
Automatic Joystick Calibration	74
Activating the Accessory Outputs	75
Data Logging in R/C Mode	76

## **SECTION 8**

Serial (RS-232) Controls and Operation	79
Use and benefits of RS232	79
Connector I/O Pin Assignment (RS232 Mode)	80
Cable configuration	81
Extending the RS232 Cable	81
Communication Settings	82
Establishing Manual Communication with a PC	82
Entering RS232 from R/C or Analog mode	83
Data Logging String in R/C or Analog mode	83
RS232 Mode if default	84
RS232 Commands Set	84
Set Motor Command Value	84
Set Accessory Outputs	85
Query Power Applied to Motors	85
Query Amps Consumed by Motors	86
Query Analog Inputs	86
Query Digital Inputs	86
Read and Modify Controller Settings	87
Apply Parameter Changes	88
Reset Controller	88
Commands Acknowledge and Error Messages	88
Character Echo	88
Command Acknowledgement	88
Command Error	89
Overheat On and Off	89
Emergency Stop	89
Watchdog time-out	89
Motor Commands during Stop Condition	89

RS-232 Watchdog	90
RS232 Accessible Parameter Table	90
Analog and R/C Modes Data Logging String Format	92
Decimal to Hexadecimal Conversion Table	93

## **SECTION 9**

Analog Control and Operation	97
Mode Description	97
Connector I/O Pin Assignment (Analog Mode)	98
Connecting to a Voltage Source	99
Connecting a Potentiometer	99
Selecting the Potentiometer Value	100
Analog Deadband Adjustment	101
Power-On Safety	102
Data Logging in Analog Mode	102

## **SECTION 10**

Position Mode	105
Mode Description	105
Selecting the Position Mode	105
Potentiometer Mounting	106
Potentiometer wiring	106
Potentiometer and Motor Polarity	107
Adding Safety Limit Switches	108
Control Loop Description	110
PD tuning	111

## **SECTION 11**

Closed Loop Speed Mode	113
Mode Description	113
Selecting the Position Mode	113
Tachometer Mounting	114
Tachometer wiring	114
Tachometer and Motor Polarity	114
Adjust Offset and Max Speed	115
Control Loop Description	116
PID tuning	117

## **SECTION 12**

Configuring the Controller using the Switches	119
Programming Methods	119
Programming using built-in Switches and Display	119
Entering Programming Mode	120
Changing parameters	121
The Special Case of Joystick Calibration	121
Restoring factory defaults	121

Exiting the Parameter Setting Mode	122
Programmable Parameters List	122

<b>SECTION 13</b>	Using the Roborun Configuration Utility	125
	System Requirements	125
	Downloading and Installing the Utility	125
	Connecting the Controller to the PC	126
	Roborun Frame, Tab and Menu Descriptions	127
	Getting On-Screen Help	128
	Loading, Changing Controller Parameters	128
	Controls Settings	129
	Power Settings	130
	Analog or R/C Specific Settings	131
	Closed Loop Parameters	132
	Viewing the Parameters Summary	133
	Running the Motors	133
	Logging Data to Disk	135
	Viewing and Logging Data in Analog and R/C Modes	137
	Loading and Saving Profiles to Disk	137
	Updating the Controller's Software	137

<b>SECTION 14</b>	Mechanical Specifications	139
	Mechanical Dimensions	139
	Mounting Considerations	140
	Thermal Considerations	140
	Wire Dimensions	141





# AX2550

## Quick Start

---

This section will give you the basic information needed to quickly install, setup and run your AX2550 controller in a minimal configuration.

### **Important Warnings**

**The AX2550 is a high power electronics device. Serious damage, including fire, may occur to the unit, motors, wiring and batteries as a result of its misuse. Please review the User's Manual for added precautions prior to applying full battery or full load power.**

**This product is intended for use with rechargeable batteries only.**

**Damage to the controller may occur if operated with a power supply. See "Power Regeneration Considerations" on page 28 of the Users Manual.**

---

### **What you will need**

For a minimal installation, gather the following components

- One AX2550 Controller and its provided cables
- 12V to 40V high capacity, high current battery
- One or two DC motors
- One R/C to DB15 connector (provided)
- Miscellaneous wires, connectors, fuses and switch

## Locating Switches, Connectors and Wires

Take a moment to familiarize yourself with the controller's wires, switches and connector.

The front side (shown in Figure 1) contains the buttons and display needed to operate and monitor the controller. The 15-pin connector provides the connection to the R/C or micro-computer, as well as connections to optional switches and sensors.

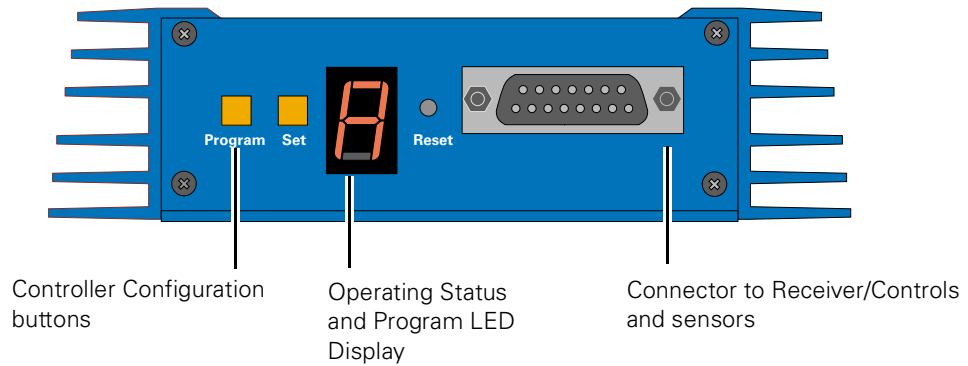


FIGURE 1. Front Controller Layout

At the back of the controller (shown in the figure) are located all the wires that must be connected to the batteries and the motors.

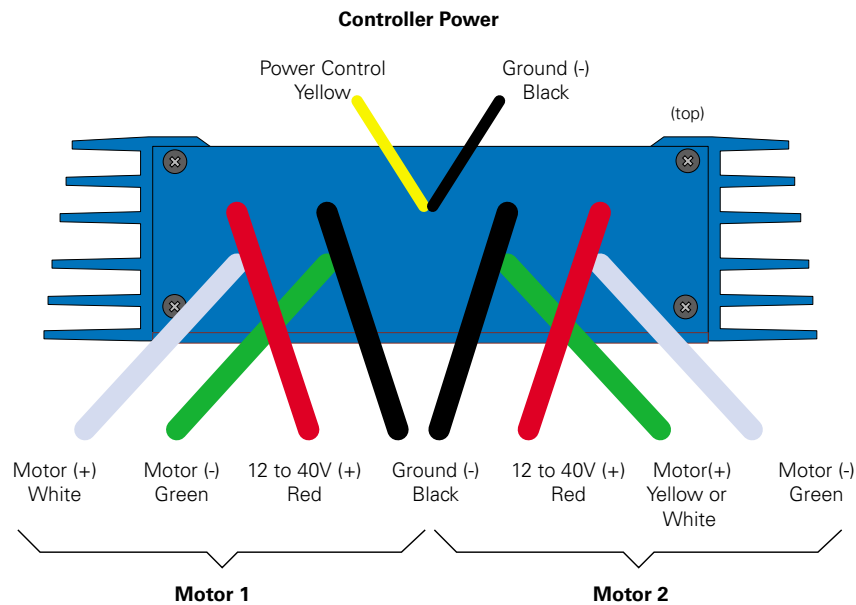


FIGURE 2. Rear Controller Layout

## Connecting to the Batteries and Motors

Connection to the batteries and motors is shown in the figure below and is done by connecting the set of wires coming out from the back of the controller.

1- Connect the two thick black wires to the minus (-) terminal of the battery that will be used to power the motors. Connect the two thick red wires to the plus (+) terminal of the battery. The motor battery may be of 12 to 40 Volts. There is no need to insert a switch on these cables, although one is suggested.

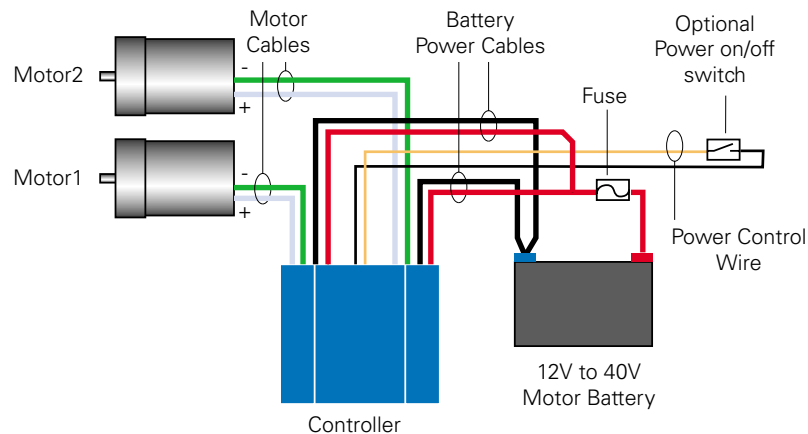
**Avoid extending the length of these wires as the added inductance may cause damage to the controller when operating at high currents. Try extending the motor wires instead.**

**The two red wires are connected to each other inside the controller. The same is true for the black wires. You should wire each pair together as shown in the diagram below.**

2- You may leave the yellow Power Control wire and the thin black wire unconnected, or you may connect them to a power switch. If left floating, protect these wires from touching any metallic part of the controller or chassis.

Refer to the chapter “Connecting Power and Motors to the Controller” on page 23 for more information about batteries and other connection options.

3- Connect each motor to one of the two output cables pair. Make sure to respect the polarity, otherwise the motor(s) may spin in the opposite direction than expected



Notes:

- The Battery Power cables are doubled in order to provide the maximum current to the controller. If only one motor is used, only one set of motor power cables needs to be connected
- Typically, 1, 2 or 3 x 12V batteries are connected in series to reach 12, 24 or 36V respectively
- The Power Control wire may be used to turn On and Off the controller, or to provide a separate and stable 12V supply to the controller's logic (See discussion below)

FIGURE 3. Electrical Power Wiring Diagram

## **Important Warning**

**The controller includes large capacitors. When connecting the Motor Power Cables, a spark will be generated at the connection point. This is a normal occurrence and should be expected.**

## **Using the Power Control Wire**

The AX2550 includes a DC/DC converter that will generate a 12V internal supply from the main +12 to +40V battery. As a result, the controller will turn On as soon as its Battery Wires (thick red and black wires) are connected to the battery.

In order to turn On and Off the controller without the need for a bulky and expensive switch or relay on the high current wires, the AX2550 uses a Power Control wire to enable or disable the internal DC/DC converter. When left unconnected, the DC/DC converter is On. When grounded, the DC/DC converter is Off.

The Power Control wire can also be used to feed a stable 12V supply to the controller so that it will continue to operate if and when the main batteries' voltage dips below 12V. The table below shows the various functions of the Power Controller Wire. See "Connecting Power" on page 23 for more details on the use and operation of the Power Control signal.

TABLE 1. Use of Power Control wire

<b>Power Control wire connected to</b>	<b>Action</b>
Floating	Controller is On
Ground	Controller is Off
Separate 12V supply	Controller is On. Controller will draw power from the Power Control wire if main battery voltage dips below 12V.

## **Important Warning**

**Never exceed 14V on the Power Control wire as this may cause permanent damage to the controller**

## **Connecting the R/C Radio**

Connect the R/C adapter cables to the controller on one side and to two or three channels on the R/C receiver on the other side. The third channel is for activating the accessory outputs and is optional.

When operating the controller in "Separate" mode, the wire labelled Ch1 controls Motor1, and the wire labelled Ch2 controls Motor2.

When operating the controller in “Mixed” mode, Ch1 is used to set the robot’s speed and direction, while Ch2 is used for steering.

See “R/C Operation” on page 63 of the User’s Manual for a more complete discussion on R/C commands, calibration and other options.

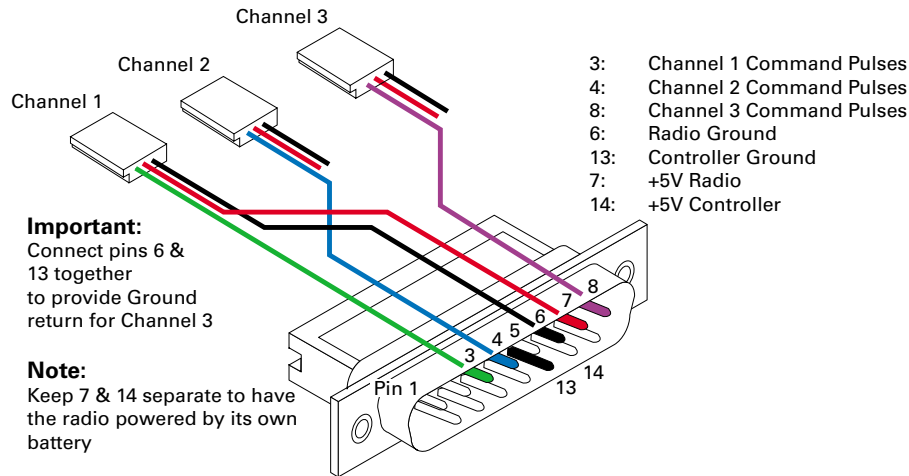


FIGURE 4. R/C connector wiring for 3 channels and battery elimination (BEC)

This wiring assumes that the R/C radio will be powered by the AX2550 controller. Other wiring options are described in “R/C Operation” on page 63 of the User’s Manual.

Connecting the optional channel 3 will enable you to turn on and off two accessory outputs. See “Connecting Sensors and Actuators to Input/Outputs” on page 31 and “Activating the Accessory Outputs” on page 75 of the User’s Manual.

## Powering On the Controller

**Important reminder:** There is no On-Off switch on the controller. You must insert a switch on the controller’s power wire as described in section “Connecting to the Batteries and Motors” on page 11.

To power the controller, center the joystick and trims on the R/C transmitter. Then turn on the switch that you have placed on the Battery Power wire or on the Power Control wire.

If the R/C transmitter and/or receiver is powered off, the display on the controller will alternate the letters spelling “no ctrl” to indicate that it is On but is not receiving a control signal.



FIGURE 5. “no control” scroll message indicates no valid R/C signal is present

Turn the R/C transmitter On. The “no ctrl” scrolling message will disappear and the display will show steady patterns depending on the motors’ selected direction.

Move the joystick on the transmitter to activate the motors to the desired speed and direction.

See “R/C Operation” on page 63 of the User’s Manual for a detailed description of the many features and options available in the R/C mode.

## Button Operation

The AX2550 has three buttons: Set, Program and Reset. These buttons are not needed for normal operation, as the controller is immediately operational upon power up.

The **Reset** button will restart the controller. This button is recessed and you will need a paper clip to press it. Reset is also accomplished by turning the controller’s power Off and back On.

The **Set** and **Program** buttons have the following functions depending how and when they are pressed:

TABLE 2. AX2550 Buttons Function

Prog and Set button status	Function
Press and hold <b>Program</b> alone during reset or power up	Enter the Programming Mode.
Press and hold <b>Set</b> alone during reset of power up	Enter Self-Test mode. See “Self-Test Mode” on page 55 of the User’s Manual
Press and hold <b>Program</b> and <b>Set</b> together during reset or power up	Reset configuration parameters to factory default
Press <b>Program</b> while Programming Mode	Accept previous parameter change and select next parameter
Press <b>Set</b> while in Programming mode	Change value of selected parameter
Press <b>Program</b> pressed alone during normal operation	No effect
Press <b>Set</b> alone during normal operation	No effect
Press <b>Program</b> and <b>Set</b> together during normal operation	Emergency stop

## Default Controller Configuration

Version 1.3 of the AX2550 software is configured with the factory defaults shown in the table below. Although Roboteq strives to keep the same parameters and values from one version to the next, previous and more recent versions match. Make sure that the matching manual and software versions. These may be retrieved from the Roboteq web site. See “Configuring the Controller using the Switches” on page 119 of the User Manual for a complete configuration parameter list and their possible values.

TABLE 3. AX2550 Default Settings

Parameter	Default Values	Letter
Input Command mode:	(0) = R/C Radio mode	I
Motor Control mode	(0) = Separate A, B, speed control, open loop	C
Amp limit	(4) = 105A	A
Acceleration	(2) = medium-slow	S
Input switch function	(3) = no action	U
Brake/Coast	(0) = brake when idle	b
Joystick Deadband	(2) = 16%	d
Exponentiation on channel 1	(0) = Linear (no exponentiation)	E
Exponentiation on channel 2	Same as E, above	F
Heat Detection	(1) = run at 50% for 30 seconds and stop	H
Left / Right Adjust	(7) = no adjustment	L

## Checking and Changing Configurations

Any one of the parameters listed in Table 2, and others not listed, can easily be changed either using the controller’s buttons or your PC with the Roboteq Configuration Utility.

The example below shows how to use the buttons to select and change the Motor Control mode from “separate” to “mixed”. See “Configuring the Controller using the Switches” on page 119 of the User’s Manual for a complete list of all the AX2550’s parameters and their meanings.

Restart  
Press & hold **Prog**



Press and hold the **Prog** button for 10 seconds while resetting or powering on the controller

Program mode entered  
after 10 seconds



After 10 seconds, the controller will enter the programming mode and flash alternatively the current parameter (I= Input Mode) and its value (0= R/C mode).

Press **Prog** to select  
next parameter



Press the **Prog** button to move to the next parameter  
(C= Motor Control Mode) and its value (0= Separate)

Press **Set** to select  
next value for parameter



Press the **Set** button to change the parameter's value  
(1= Combined)

Press **Prog** to store change  
and select next parameter



Press the **Prog** button record the change and move to  
the next parameter (A= Amps limit) and it's value (2=  
75A)

Reset controller  
to exit

Press the **Reset** button or power off/on the control to  
restart the controller using the new parameters.

## Connecting the controller to your PC using Roborun

Connecting the controller to your PC is not necessary for basic R/C operation. However, it is a very simple procedure that is useful for the following purposes:

- to Read and Set the programmable parameters with a user-friendly graphical interface
- to obtain the controller's software revision and date
- to send precise commands to the motors
- to read and plot real-time current consumption value
- Save captured parameters onto disk for later analysis
- to update the controller's software



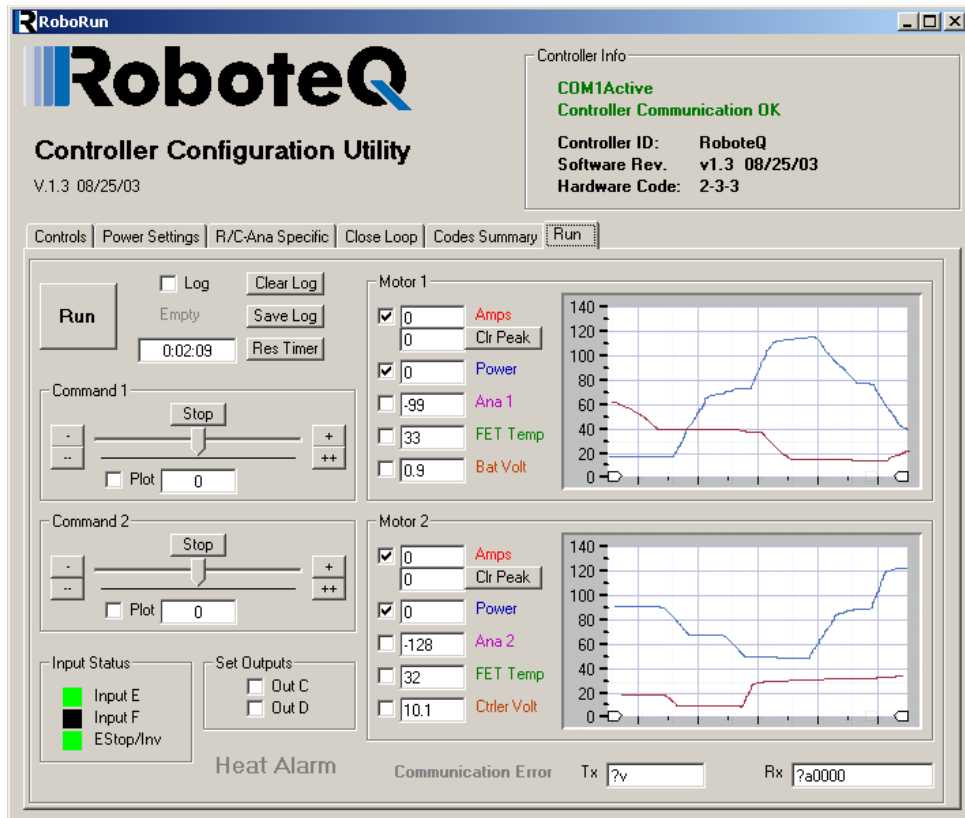


FIGURE 6. Roborun Utility screen layout

To connect the controller to your PC, use the provided cable. Connect the 15-pin connector to the controller. Connect the 9-pin connector to your PC's available port (typically COM1). Apply power to the controller to turn it on.

Download the Roborun software from [www.roboteq.com](http://www.roboteq.com), install it on your PC and launch the program. The software will automatically establish communication with the controller, retrieve the software revision number and present a series of buttons and tabs to enable its various possibilities.

The intuitive Graphical User Interface will let you view and change any of the controller's parameters. The "Run" tab will present a number of buttons, dials and charts that are used for operating and monitoring the motors.

## Obtaining the Controller's Software Revision Number

One of the unique features of the AX2550 is the ability to easily update the controller's operating software with new revisions downloaded from Roboteq's web site at [www.roboteq.com](http://www.roboteq.com). This is useful for adding features and/or improving existing ones.

Each software version is identified with a unique number. Obtaining this number can be done using the PC connection discussed previously.

It is also possible to get the controller to display the software version number by following these simple steps

- Disconnect the power from the motor batteries
- Press and hold the **Set** button while powering or resetting the controller

The LED will display a sequence of two numerical digits and an optional letter separated by dashes as shown in the examples below.



FIGURE 7. Press and hold “Set” to display version number and enter self-test

After these digits are displayed, the controller will attempt to power the motors as part of the self test mode (see “Self-Test Mode” on page 55 of the User’s Manual for a more detailed explanation). This is why the **motor’s battery must be disconnected**. After about 30 seconds, the software revision number will be displayed every 30 seconds.

You will need to reset, or power down and up, the controller to exit and resume normal operations.

Now that you know your controller’s software version number, you will be able to see if a new version is available for download and installation from Roboteq’s web site, and which features have been added or improved.

Installing new software is a simple and secure procedure, fully described in “Updating the Controller’s Software” on page 137 of the User’s Manual.

---

## Exploring further

By following this quick-start section, you should have managed to get your controller to operate in its basic modes within minutes of unpacking.

Each of the features mentioned thus far has numerous options which are discussed further in the complete User’s Manual, including:

- Self test mode
- Emergency stop condition
- Joystick calibration
- Using Inputs/Outputs
- Current limiting
- Software updating
- and much more

# AX2550 Digital Speed Controller Overview

---

Congratulations! By selecting Roboteq's AX2550 you have empowered yourself with the industry's most versatile and programmable DC Motor Controller for mobile robots. This manual will guide you step by step through its many possibilities.

---

## Product Description

The AX2550 is a highly configurable, microcomputer-based, dual-channel digital speed or position controller with built-in high power drivers. The controller is designed to interface directly to high power DC motors in computer controlled or remote controlled mobile robotics and automated vehicle applications.

The AX2550 controller can accept speed or position commands in a variety of ways: pulse-width based control from a standard Radio Control receiver, Analog Voltage commands, or RS-232 commands from a microcontroller or wireless modem.

The controller's two channels can be operated independently or can be combined to set the forward/reverse direction and steering of a vehicle by coordinating the motion on each side of the vehicle. In the speed control mode, the AX2550 can operate in open loop or closed loop. In closed loop operation, actual speed measurements from tachometers are used to verify that the motor is rotating at the desired speed and direction and to adjust the power to the motors accordingly.

The AX2550 can also be configured to operate as a precision, high torque servo controller. When connected to a potentiometer coupled to the motor assembly, the controller will command the motor to rotate up to a desired angular position. Depending on the DC motor's power and gear ratio, the AX2550 can be used to move or rotate steering columns or other physical objects with very high torque.

The AX2550 is fitted with many safety features ensuring a secure power-on start, automatic stop in case of command loss, over current protection on both channels, and overheat protection.

The motors are driven using high-efficiency Power MOSFET transistors controlled using Pulse Width Modulation (PWM) at 16kHz. The AX2550 power stages can operate from 12 to 40VDC and can sustain up to 120A of controlled current, delivering up to 4800W (approximately 6 HP) of useful power to each motor.

The many programmable options of the AX2550 are easily configured using one-touch Program and Set buttons and a 7-segment LED display. Once programmed, the configuration data are stored in the controller's non-volatile memory, eliminating the need for cumbersome and unreliable jumpers.

---

## **Technical features**

### **Fully Digital, Microcontroller-based Design**

- Multiple operating modes
- Fully programmable using either built-in switches and 7 segment display or through connection to a PC
- Non-volatile storage of user configurable settings. No jumpers needed
- Simple operation
- Software upgradable with new features

### **Multiple Command Modes**

- Radio-Control Pulse-Width input
- Serial port (RS-232) input
- 0-5V Analog Command input

### **Multiple Advanced Motor Control Modes**

- Independent operation on each channel
- Mixed control (sum and difference) for tank-like steering
- Open Loop or Closed Loop Speed mode
- Position control mode for building high power position servos
- Modes selectable independently for each channel

### **Automatic Joystick Command Corrections**

- Joystick min, max and center calibration
- Selectable deadband width
- Selectable exponentiation factors for each joystick
- 3rd R/C channel input for weapon and accessory output activation

### **Special Function Inputs/Outputs**

- 2 Analog inputs. Used as
  - Tachometer inputs for closed loop speed control
  - Potentiometer input for position (servo mode)
  - Motor temperature sensor inputs
  - Battery voltage sensor

- User defined purpose (RS232 mode only)
- One Switch input configurable as
  - Emergency stop command
  - Reversing commands when running vehicle inverted
  - General purpose digital input
- Up to 2 general purpose outputs for accessories or weapon
  - One 24V, 2A output
  - One low-level digital output
- Up to 2 general purpose digital inputs

**Internal Sensors**

- Voltage sensor for monitoring the main 12 to 40V battery system operation
- Voltage monitoring of internal 12V
- Temperature sensors on the heat sink of each power output stage
- Sensor information readable via RS232 port

**Low Power Consumption**

- On board DC/DC converter for single 12 to 40V battery system operation
- Optional 12V backup power input for powering safely the controller if the motor batteries are discharged
- Max 200mA at 12V or 100mA at 24V idle current consumption
- Power Control wire for turning On or Off the controller from external microcomputer or switch
- No power consumed by output stage when motors are stopped
- Regulated 5V output for powering R/C radio. Eliminates the need for separate R/C battery

**High Efficiency Motor Power Outputs**

- Two independent power output stages
- Dual H bridge for full forward/reverse operation
- Ultra-efficient 2.5 mOhm ON resistance (RDSon) MOSFET transistors
- 12 to 40 V operation
- High current 8 AWG cable sets for each power stages
- SmartAmps Automatic current limitation based on actually measured transistor temperature.
  - 120A up to 15 seconds (per channel)
  - 100A up to 30 seconds
  - 80A extended
  - High current operation may be extended with forced cooling
- 250A peak Amps per channel
- 16kHz Pulse Width Modulation (PWM) output
- Auxiliary output for brake or clutch
- Heat sink extruded case

**Advanced Safety Features**

- Safe power on mode
- Optical isolation on R/C control inputs
- Automatic Power stage off in case of electrically or software induced program failure
- Overvoltage and Undervoltage protection
- Watchdog for automatic motor shutdown in case of command loss (R/C and RS232 modes)
- Large, bright run/failure diagnostics on 7 segment LED display
- Programmable motor acceleration
- Built-in controller overheat sensor
- Motor temperature sensing and protection
- Emergency Stop input signal and button

**Data Logging Capabilities**

- 12 internal parameters, including battery voltage, captured R/C command, temperature and Amps accessible via RS232 port
- Data may be logged in a PC, PDA or microcomputer

**Sturdy and Compact Mechanical Design**

- Built from aluminum heat sink extrusion with mounting brackets
- Efficient heat sinking. Operates without a fan in most applications.
- 7" (178mm) long (excluding mounting brackets) by 5.5" wide (140mm) by 1.8" (40mm) high
- -20o to +70o C operating environment
- 2 lbs (900g)

## SECTION 3

# Connecting Power and Motors to the Controller

This section describes the AX2550 Controller's connections to power sources and motors.

## **Important Warning**

**Please follow the instructions in this section very carefully. Any problem due to wiring errors may have very serious consequences and will not be covered by the product's warranty.**

---

## **Connecting Power**

The AX2550 has 2 sets of Ground (black) and V+ (red) power cables and a Power Control wire (yellow):

The power cables are located at the back end of the controller. The various power cables are identified by their position, wire thickness and color: Red is positive (+), black is negative or ground (-).

The power cables to the batteries and motors are shown in the figure below.

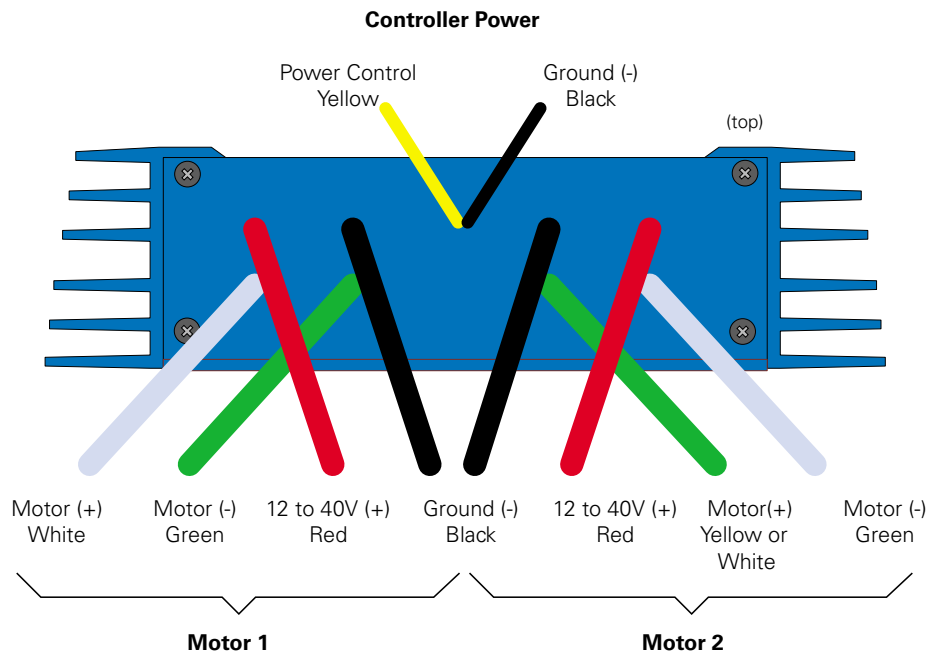


FIGURE 8. Controller rear plate and power wiring

## Controller Power

The AX2550 uses a flexible powering scheme that is best described in Figure 9. On this diagram, it can be seen that the Control Logic requires a stable 12V supply, while the Power Output stage that drives the motors can tolerate a very wide voltage range.

Because of its wide tolerance, the Power Output stage is wired directly to the Main Battery wires.

The control logic is connected to both the DC/DC converter as well as the Power Control wire. If the voltage applied to the DC/DC converter's input is lower than the 12V needed for its proper operation, the Control Logic will stop unless the Power Control wire is connected to a separate 12V power source. The diode circuit is designed to automatically select one power source over the other. The diodes will let through the source that is 0.6V higher than the other. If the voltage difference is less than 0.6V between these two sources, both supplies will source current into the controller.

The Power Control wire also serves as the Enable signal for the DC/DC converter. When floating or pulled to above 1V, the DC/DC converter is active and supplies the AX2550's control logic, thus turning it On. When the Power Control wire is pulled to Ground, the DC/DC converter is stopped and the controller is turned Off.



All 3 ground wires (-) are connected to each other inside the controller. The two main battery wires are also connected to each other internally. However, you must never assume that connecting one wire of a given color will eliminate the need for connecting the other. Malfunction or even damage may occur if one or two wires are assuming the load of all three.

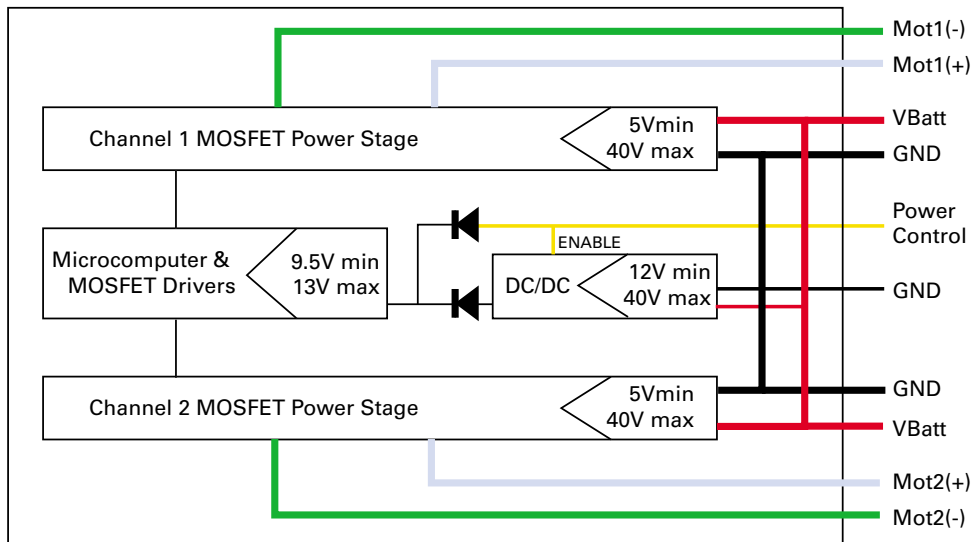


FIGURE 9. Representation of the AX2550's Internal Power Circuits

TABLE 3. Effect of Power Control under various voltage conditions

Power Control wire is connected to	And Main Battery Voltage is	Action
Ground	Any Voltage from 0V to 40V	Controller is Off
Floating	Below 7V	Controller is Off
Floating	Between 7V and 10.5V	Controller Logic is On Power Stage is Disabled (under-voltage condition)
Floating	Between 10.5 and 40V	Controller is On. Drawing power from DC/DC converter Power Stage is Active
12V	Below 12V	Controller is On. Drawing power from external source Power Stage is Active

TABLE 3. Effect of Power Control under various voltage conditions

Power Control wire is connected to	And Main Battery Voltage is	Action
12V	Between 12 and 40V	Controller is On. Drawing power from DC/DC converter Power Stage is Active
12.5 to 14V	Any Voltage from 0V to 40V	Controller is On. Drawing power from external source Power Stage is Active

### Powering the Controller using the Motor Batteries

The AX2550 included a DC/DC converter to generate the internal 12V required for its operation. The diagram on Figure 11 show how to wire the controller to a single battery circuit and the two options for turning the power On and Off. The diagram shows two switches, although switching either one alone will power the controller Off.

In a typical configuration, it is recommended that the Motor Power be always applied and that the Controller's Power be controlled using a switch on the Power Control wire (yellow). When the controller is Off, the output transistors are in the Off position and no power is drawn on the Motor Power battery.

For safety reasons, however, it is highly recommended that a way of quickly disconnecting the Motor Power be provided in the case of loss of control and all of the AX2550 safety features fail to activate

Note, however, that eventually the motor batteries will get weaker and the voltage drop below the level needed for the internal DC/DC converter to properly operate. For all professional applications it is therefore recommended to add a separate 12V power supply to ensure proper powering of the controller under any conditions.

### Important Warning

**Unless you can ensure a steady 12V voltage in all conditions, it is recommended that the battery used to power the controller's electronics must be separate from the one used to power the motors. This is because it is very likely that the motor batteries will be subject to very large current loads, which may cause the voltage to eventually dip below 12V as the batteries' charge drops. The separate backup 12V power supply should be connected to the Power Control wire.**

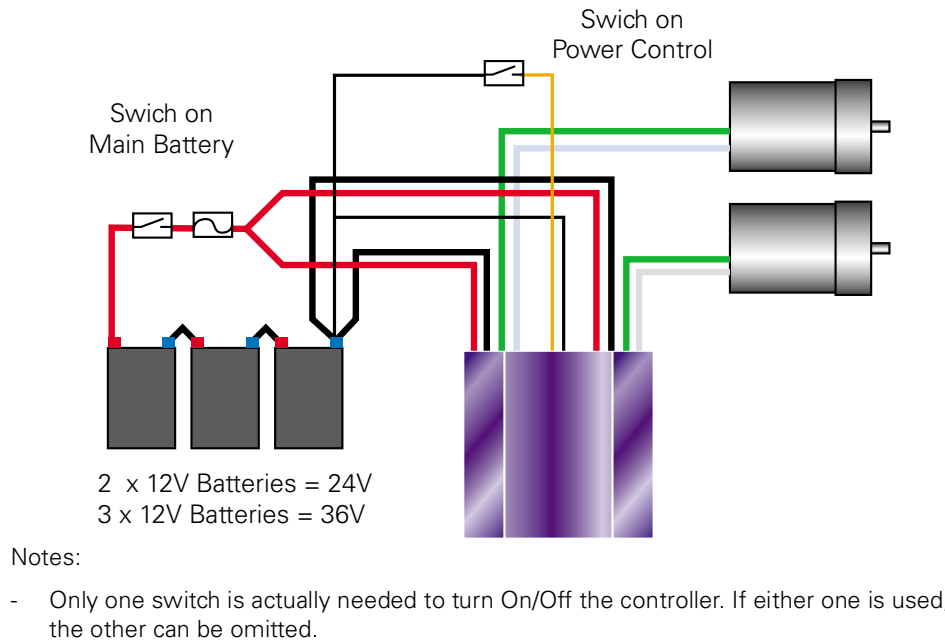


FIGURE 10. Powering the controller from the Motor Batteries

## Using a Backup Battery

To ensure that the controller will always be operational, regardless of the charge left in the main battery, it is recommended to add a 12V backup battery connected as shown in Figure 11.

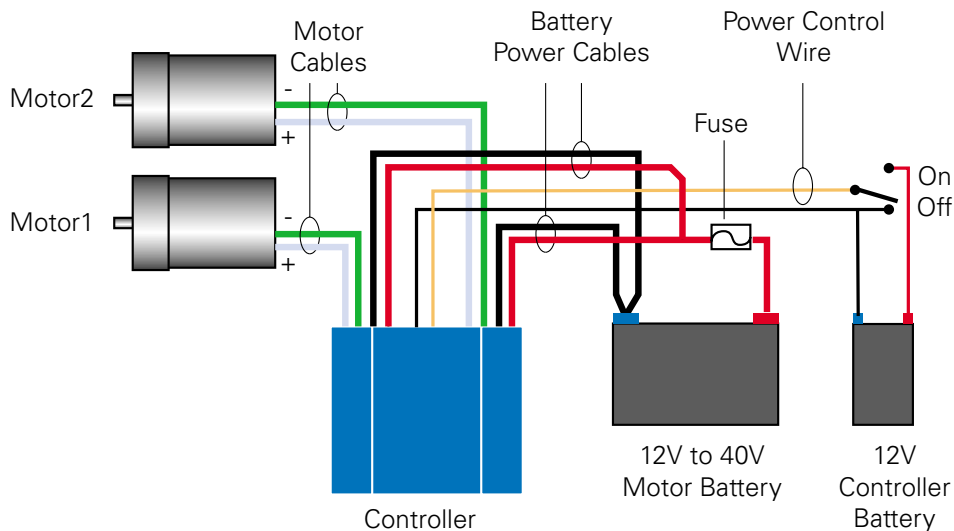


FIGURE 11. Power wiring using a two-battery system

## **Important Warning**

**Make sure that your motors have their wires isolated from the motor casing. Some motors, particularly automotive parts, use only one wire, with the other connected to the motor's frame.**

**If you are using such a motor, make sure that it is mounted on isolators and that its casing will not cause a short circuit with other motors and circuits which may also be inadvertently connected to the same metal chassis.**

## **Power Fuses**

For low Amperage applications (below 30A per motor), it is recommended that a fuse be inserted in series with the main battery circuit as shown in the figures above.

The fuse will be shared by the two output stages and therefore must be placed before the Y connection to the two power wires.

Automotive fuses are generally slow and the following values are recommended depending on the desired level of protection. It should be noted that the current values listed will apply to the sum of the current drawn by each channel. Practically, a fuse will be of limited effectiveness. In very high current application (80A or more per channel), it will simply not be possible to find a fuse with the necessary Amps rating and speed needed to provide adequate protection. In such situation, the fuse may be just as well omitted keeping in mind that the controller's current limiting circuit may not be able to protect it against short circuits and other abnormal situations.

TABLE 4. Fuse selection table

<b>Desired Current Protection (total for Channel 1 + Channel 2)</b>	<b>Fuse</b>
120A	100A
100A	80A
80A	60A
60A	50A
50A	40A
40A	35A

## **Important Warning**

**Fuses are typically slow to blow and will thus allow temporary excess current to flow through them for a time (the higher the excess current, the faster the fuse will blow). This characteristic is desirable in most cases, as it will allow motors to draw surges during acceleration and braking. However, it also means that the fuse may not be able to protect the controller in case of overvoltage or high current spikes.**

---

## Wire Length Limits

The AX2550 regulates the output power by switching the power to the motors On and Off at high frequencies. At such frequencies, the wires' inductance produces undesirable effects such as parasitic RF emissions, ringing and overvoltage peaks. The controller has built-in capacitors and voltage limiters that will reduce these effects. However, should the wire inductance be increased, for example by extending the wire length, these effects will be amplified beyond the controller's capability to correct them. This is particularly the case for the power wires (thick red and black cables).

## Important Warning

**Avoid extending the black and red Motor Power wires beyond their original length as the added inductance may cause damage to the controller when operating at high currents. Try extending the motor wires instead.**

---

## Electrical Noise Reduction Techniques

As discussed in the above section, the AX2550 uses fast switching technology to control the amount of power applied to the motors. While the controller incorporates several circuits to keep electrical noise to a minimum, additional techniques can be used to keep the noise low when installing the AX2550 in an actual robotic vehicle. Below is a list of techniques you can try to keep noise emission low:

- Keep wires as short as possible
- Loop wires through ferrite beads
- Add snubber R/C circuit at motor terminals
- Keep controller, wires and battery enclosed in metallic body

---

## Power Regeneration Considerations

When a motor is spinning faster than it would normally at the applied voltage, such as when moving downhill or decelerating, the motor acts like a generator. In such case, the current will flow in the opposite direction, back to the battery.

It is therefore essential that the AX2550 be connected to rechargeable batteries. If a power supply is used instead, during regeneration the power will attempt to flow back in the power supply potentially damaging it and/or the controller.

Regeneration can also cause potential problems if the battery is disconnected while the motors are still spinning. In such a case, and depending on the command level applied at that time, the regenerated voltage will attempt to flow back to the battery but since none is present, the voltage will rise to potentially unsafe levels. The AX2550 includes an over-voltage protection circuit to prevent damage to the output transistors (see "Overvoltage Protection" on page 30)

## **Important Warning**

**Use the AX2550 only with a rechargeable battery as supply to the Motor Power wires (thick black and red wires). If a power supply is used, damage to the controller and/or power supply may occur during regeneration.**

## **Important Warning**

**Avoid switching Off the main power cables (thick wires) while the motors are spinning. Damage to the controller may occur.**

---

## **Overvoltage Protection**

The AX2550 includes a battery voltage monitoring circuit that will cause the output transistors to be turned off if the main battery voltage applied on the thick red and black wires is above 45V.

This protection is designed to prevent the voltage created by the motors during regeneration to be "amplified" to unsafe levels by the switching circuit.

This circuit will not protect the controller against powering the controller with a power source above 45V.

The controller will resume normal operation when the measured voltage drops below 45V.

---

## **Undervoltage Protection**

In order to ensure that the power MOSFET transistors are switched properly, the AX2550 monitors the internal 12V power supply that is used by the MOSFET drivers. If the internal voltage drops below 9.5V, the controller's output stage is turned Off. The rest of the controller's electronics, including the microcomputer will remain operational as long as the internal voltage is above 7V.

The internal voltage will be the greater value of either the DC/DC converter's output or the external backup voltage applied to the power control wire (see "Controller Power" on page 24)

If the internal voltage comes from the DC/DC converter, its value will be 12V as long as the main battery voltage is above 13V. The DC/DC converter's output will be approximately 1V lower than the Main Battery voltage if the latter is below 13V.

If an external backup power supply is used on the yellow Power Control wire, the internal voltage will be 0.5V lower than the applied voltage.

## SECTION 4

# Connecting Sensors and Actuators to Input/Outputs

This section describes the various inputs and outputs and provides guidance on how to connect sensors, actuators or other accessories to them.

---

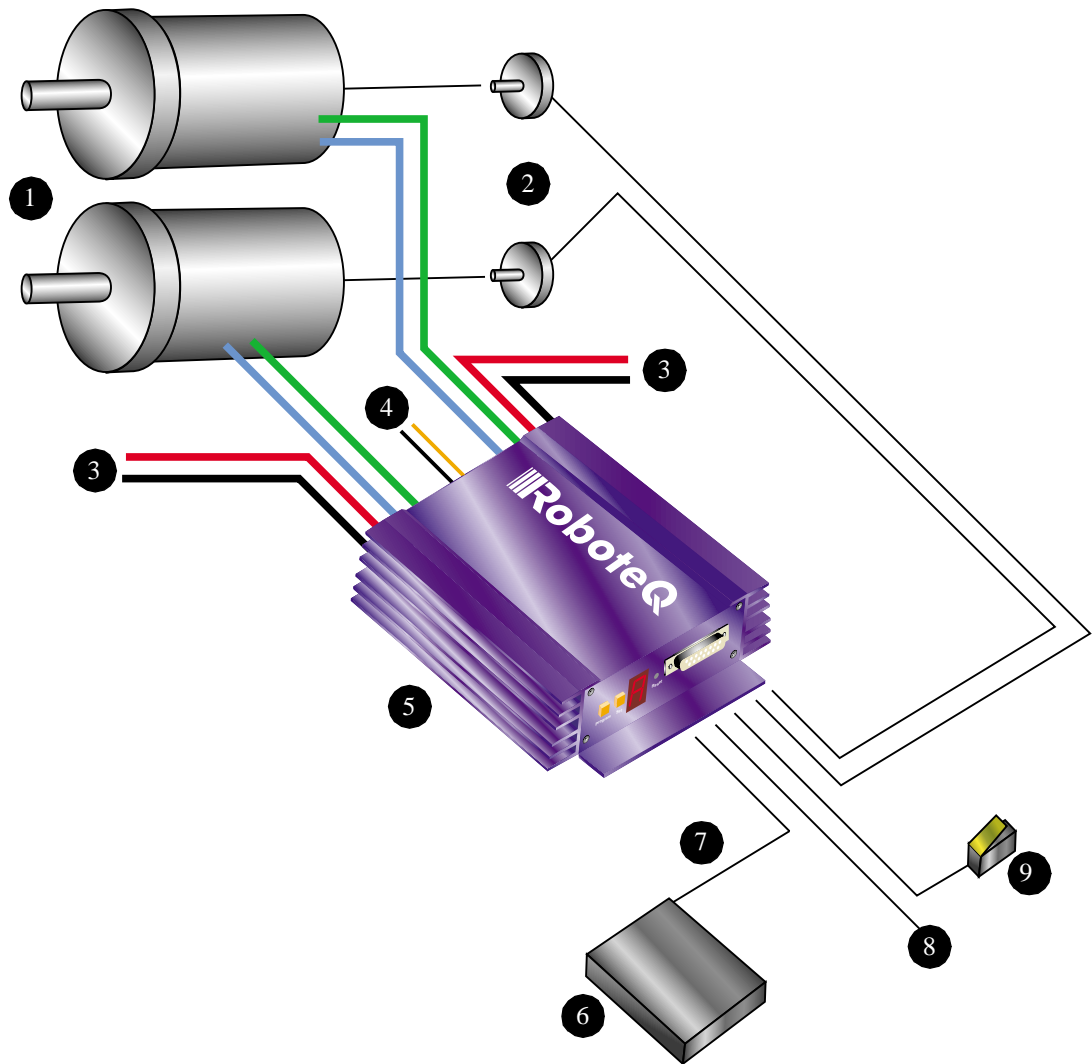
## AX2550 Connections

The AX2550 uses a set of power wires (located on the back of the unit) and a DB15 connector for all necessary connections. The diagram on Figure 11 shows a typical wiring diagram of a mobile robot using the AX2550 controller.

The wires are used for connection to the batteries and motors and will typically carry large current loads. Details on the controller's power wiring can be found at "Connecting Power and Motors to the Controller" on page 23

The DB15 connector is used for all low-voltage, low-current connections to the Radio, Microcontroller, sensors and accessories. This section covers only the connections to sensors and actuators.

For information on how to connect the R/C radio or the RS232 port, see "R/C Operation" on page 63 and "Serial (RS-232) Controls and Operation" on page 79.



- |  |   |
|--|---|
| 1- DC Motors                           | 6- R/C Radio Receiver, microcomputer, or wireless modem |
| 2- Optional sensors:                   | 7- Command: RS-232, R/C Pulse                           |
| - Tachometers (Closed loop speed mode) | 8- Miscellaneous I/O                                    |
| - Potentiometers (Servo mode)          | 9- Running Inverted, or emergency stop switch           |
| - Thermistors (Temperature protection) |   |
| 3- Motor Power supply wires            |   |
| 4- Power Control wire                  |   |
| 5- Controller                          |   |

FIGURE 11. Typical controller connections



## AX2550's Inputs and Outputs

In addition to the RS232 and R/C channel communication lines, the AX2550 includes several inputs and outputs for various sensors and actuators. Depending on the selected operating mode, some of these I/Os provide feedback and/or safety information to the controller.

When the controller operates in modes that do not use these I/O, these signals become available for user application. Below is a summary of the available signals and the modes in which they are used by the controller or available to the user.

TABLE 4. AX2550 I/O functions

Signal	I/O type	Use	Activated
Out C	2A Digital Output	User defined	Always active. Activated using R/C channel 3 (R/C mode), or serial command (RS232 mode)
Out D	Low Current Digital Output	User defined	Always active. Activated using R/C channel 3 (R/C mode), or serial command (RS232 mode)
Inp E	Digital Input	User defined	Active in RS232 mode only. Read with serial command (RS232)
		Activate Output C	When Input is configured to drive Output C
		Turn FETs On/Off	When Input is configured as "dead man switch" input
Inp F	Digital Input	Same as Input E	
EStop/Invert	Digital Input	Emergency stop	When Input is configured as Emergency Stop switch input.
		Invert Controls	When Input is configured as Invert Controls switch input.
		User defined	When input is configured as general purpose. Read with serial command (RS232).
Sensor 1	Analog Input	Tachometers input	When Channel 1 is configured in Closed Loop Speed Control
		Position sensing	When Channel 1 is configured in Closed Loop Position Control
		Temperature sensor	When Channel 1 is configured is Open Loop Speed Control and Motor Temperature protection is enabled
		User defined	When Channel 1 is configured as Open Loop Speed Control and Temperature protection is disabled. Read with serial command (RS232).
Sensor 2	Analog Input	Same as Sensor 1	

## I/O List and Pin Assignment

The figure and table below lists all the inputs and outputs that are available on the AX2550.

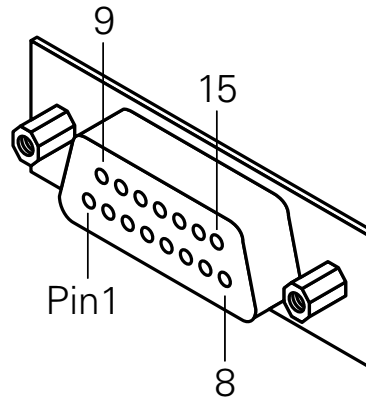


FIGURE 12. Controller's DB15 connector pin numbering

TABLE 5. DB15 connector pin assignment

Pin Number	Wire Color*	Input or Output	Signal	Description
1 and 9	Orange	Output	Output C	2A Accessory Output C
2	White	Output	R/C: RS232 data	RS232 Data Logging Output
			RS232: Data Out	RS232 Data Out
			Analog: RS232 Out	RS232 Data Logging Output
3		Input	R/C: Ch 1	R/C radio Channel 1 pulses
			RS232: Data In	RS232 Data In (from PC/MCU)
			Analog: Unused	Unused
4	Purple	Input	R/C: Ch 2	R/C radio Channel 2 pulses
			Ana/RS232: Input F	Digital Input F in RS232 mode
5 and 13	Black	Power Out	Ground	Controller ground (-)
6		Power In	Ground	Connect to pin 5**
7		Power In	+5V	Connect to pin 14**
8	Gray	Input	R/C: Ch 3	R/C radio Channel 3 pulses
			Input E	Accessory input E in RS232 mode
10	Green	Analog in	RC/RS232: Ana in 1	Channel 1 speed, position or temp input
			Analog: Command 1	Analog command for channel 1

TABLE 5. DB15 connector pin assignment

Pin Number	Wire Color*	Input or Output	Signal	Description
11	Blue	Analog in	RC/RS232: Ana in 2	Channel 2 speed, position or temp input
			Analog: Command 2	Analog command for channel 2
12	Yellow	Output	Output D	Low Current Accessory Output D
14	Red	Power Out	+5V	+5V Power Output (100mA max.)
15	Brown	Input	Input EStop/Inv	Emergency Stop or Invert Switch input

\*The wire colors are those used by Roboteq on our prefabricated cables. It is recommended you use these colors for consistency.

\*\*These connections should only be done in RS232 mode or R/C mode with radio powered from the controller. Otherwise connect to radio battery.

## Connecting devices to Output C

Output C is a buffered, Open Drain MOSFET output capable of driving over 2A at up to 24V.

The diagrams on Figure 13 show how to connect a light or a relay to this output:

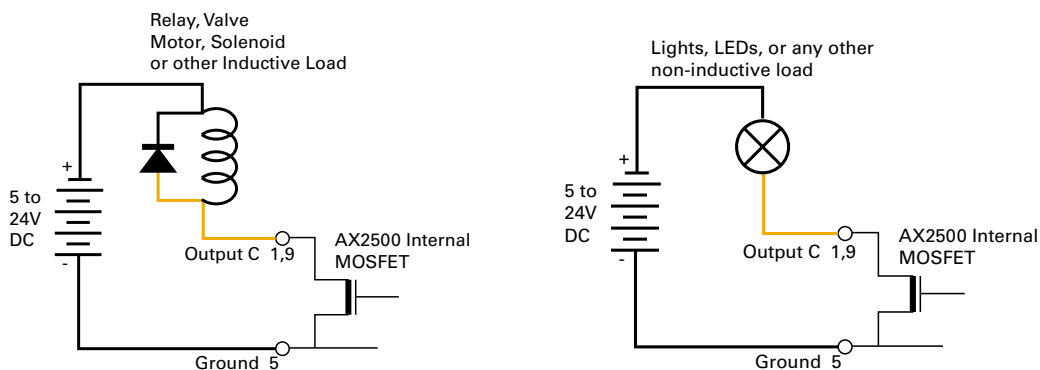


FIGURE 13. Connecting inductive and resistive loads to Output C

This output can be turned On and Off using the Channel 3 Joystick when in the R/C mode. See "Activating the Accessory Outputs" on page 75 for more information.

When the controller is used in RS232 mode, this output can be turned On and Off using the **!c** (On) and **!C** (Off) command strings. See "RS232 Commands Set" on page 84 for more information.

## Important warning:

**This output is unprotected. If your load draws more than 2A, permanent damage may occur to the power transistor inside the controller. A 1A fuse may be used in series with the load for increased protection.**

**Overvoltage spikes induced by switching inductive loads, such as solenoids or relays, will destroy the transistor unless a protection diode is used.**

## Connecting devices to Output D

Output D is a low-current, unbuffered output. When Off, this output will be pulled to ground through a 10Kohm resistor. When On, this output will be pulled to around 4.5V through that same resistor.

The D output cannot drive any load directly and requires an external transistor or solid state relay to drive high current accessories. Because they are external, the size, voltage, and power of these transistors can be selected to best suit a particular need.

The diagram on Figure 14 shows how to drive a large inductive load using the D output and a Power MOSFET widely available at most electronic component distributors. The MOSFET must conduct with a gate voltage of 4V or higher.

The D output can be toggled On and Off using the Channel 3 Joystick when in the R/C mode. See "Activating the Accessory Outputs" on page 75 for more information.

When the controller is used in RS232 mode, this output can be turned On and Off using the **!D** (On) and **!d** (Off) command strings. See "RS232 Commands Set" on page 84 for more information.

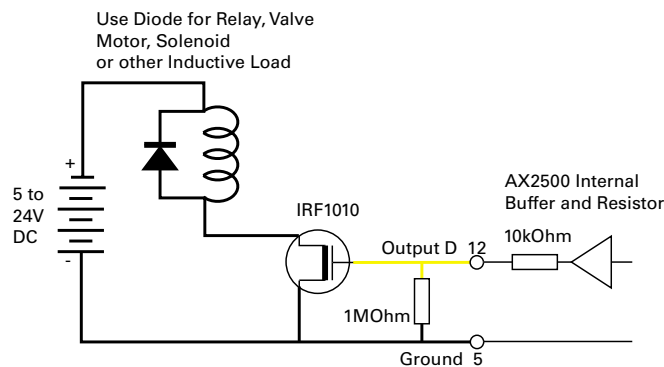


FIGURE 14. Connecting external MOSFET and load to Output D

## Connecting Switches or Devices to Input E

Input E is a general purpose, digital input. This input is only active when in the RS232 and Analog modes. In R/C mode, this line is used as the radio channel 3 input.

Input E is a high impedance input with a pull-up resistor built into the controller. Therefore it will report an On state if unconnected, and a simple switch as shown on Figure 16 is necessary to activate it.

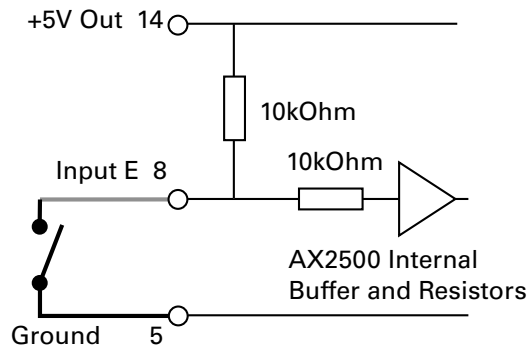


FIGURE 15. Switch wirings to Input E

The status of Input E can be read in the RS232 mode with the **?i** command string. The controller will respond with three sets of 2 digit numbers. The status of Input E is contained in the first set of numbers and may be 00 to indicate an Off state, or 01 to indicate an On state.

## Connecting Switches or Devices to Input F

Input F is a general purpose digital input. This input is only active when in the RS232 mode. In R/C mode, this line is used as the radio channel 2 input.

Input F is an opto-coupled input and requires the opto coupler buffer to be powered for it to work. Figure 16 below shows how to wire the buffer's power inputs (+5V In, GND In). Seen from the outside, Input F is similar to input E, but with a lower impedance of 10kOhm. If left unconnected, this input will report an Off state. As a result, this input may be activated by a simple switch connected to +5V and no external resistor, as show in the Figure.

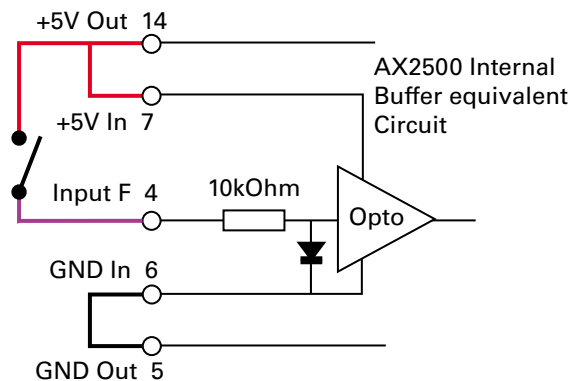


FIGURE 16. Switch wiring to Input F

The status of Input F can be read in the RS232 mode with the **?i** command string. The controller will respond with three sets of 2 digit numbers. The status of Input F is contained in

the second set of numbers and may be 00 to indicate an Off state, or 01 to indicate an On state.

## Connecting Switches or Devices to EStop/Invert Input

This input is used to connect various switches or devices depending on the selected controller configuration.

The factory default for this input is “No Action”.

This input can also be configured to be used with an optional “inverted” sensor switch. When activated, this will cause the controls to be inverted so that the robot may be driven upside-down.

When neither Emergency Stop or Inverted modes are selected, this input becomes a general purpose input like the other two described above.

This input is a high impedance input with a pull-up resistor built into the controller. Therefore it will report an On state (no emergency stop, or not inverted) if unconnected. A simple switch as shown on Figure 17 is necessary to activate it. Note that to trigger an Emergency Stop, or to detect robot inversion this input must be pulled to ground. Figure 17 shows how to wire the switch to this input.

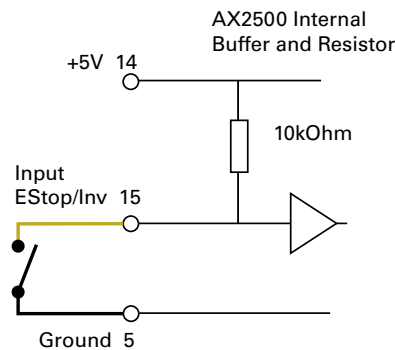


FIGURE 17. Emergency Stop / Invert switch wiring

The status of the EStop/Inv can be read at all times in the RS232 mode with the **?i** command string. The controller will respond with three sets of 2 digit numbers. The status of the ES/Inv Input is contained in the last set of numbers and may be 00 to indicate an Off state, or 01 to indicate an On state.

## Connecting Position Potentiometers to Analog Inputs

When configured in the Position mode, the controller's analog inputs are used to obtain position information from a potentiometer coupled to the motor axle. This feature is useful in order to create very powerful servos as proposed in the figure below:

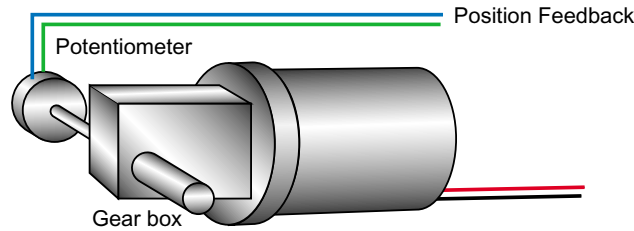


FIGURE 18. Motor and potentiometer assembly for position servo operation

Connecting the potentiometer to the controller is as simple as shown in the diagram on Figure 19.

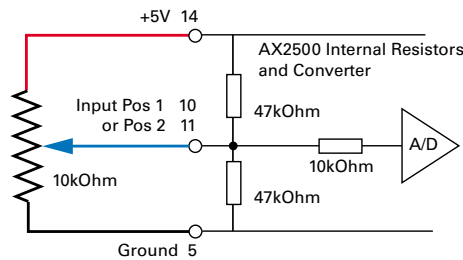


FIGURE 19. Potentiometer wiring in Position mode

The potentiometer must be attached to the motor frame so that its body does not move in relationship with the motor. The potentiometer axle must be firmly connected to the gear box output shaft. The gearbox must be as tight as possible so that rotation of the motor translates into direct changes to the potentiometers, without slack, at the gearbox's output.

See "Position Mode" on page 105 for complete details on Position Mode wiring and operation.

## Important Warning

**Beware that the wrong + and - polarity on the potentiometer will cause the motor to turn in the wrong direction and not stop. The best method to figure out the right potentiometer is try one way and change the polarity if incorrect. Note that while you are doing these tests, the potentiometer must be loosely attached to the motor's axle so that it will not be forced and broken by the motor's uncontrolled rotation in case it was wired wrong.**

## Connecting Tachometer to Analog Inputs

When operating in closed loop speed mode, tachometers must be connected to the controller to report the measured motor speed. The tachometer can be a good quality brushed DC motor used as a generator. The tachometer shaft must be directly tied to that of the motor with the least possible slack.

Since the controller only accepts a 0 to 5V positive voltage as its input, the circuit shown in Figure 20 must be used between the controller and the tachometer: a 10kOhm potentiometer is used to scale the tachometer output voltage to -2.5V (max reverse speed) and +2.5V (max forward speed). The two 1kOhm resistors form a voltage divider that sets the idle voltage at mid-point (2.5V), which is interpreted as the zero position by the controller. The voltage divider resistors should be of 1% tolerance or better. To precisely adjust the 2.5V midpoint value it is recommended to add a 100 ohm trimmer on the voltage divider.

With this circuitry, the controller will see 2.5V at its input when the tachometer is stopped, 0V when running in full reverse, and +5V in full forward.

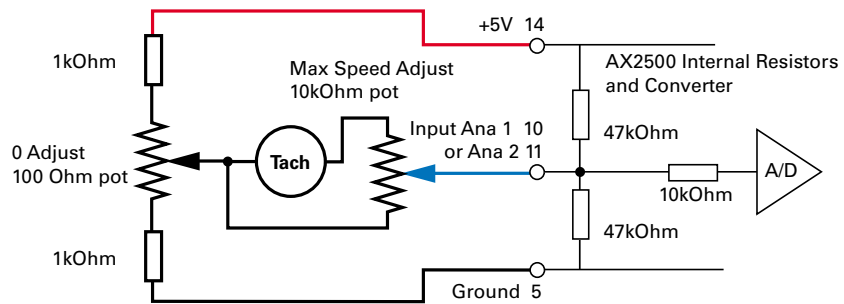


FIGURE 20. Tachometer wiring diagram

The tachometers can generate voltages in excess of 2.5 volts at full speed. It is important, therefore, to set the potentiometer to the minimum value (cursor all the way down per this drawing) during the first installation.

Since in closed loop control the measured speed is the basis for the controller's power output (i.e. deliver more power if slower than desired speed, less if higher), an adjustment and calibration phase is necessary. This procedure is described in "Closed Loop Speed Mode" on page 113.

## Important Warning

**The tachometer's polarity must be such that a positive voltage is generated to the controller's input when the motor is rotating in the forward direction. If the polarity is inverted, this will cause the motor to run away to the maximum speed as soon as the controller is powered with no way of stopping it other than pressing the emergency stop button or disconnecting the power.**



## Connecting External Thermistor to Analog Inputs

Using external thermistors, the AX2550 can be made to supervise the motor's temperature and adjust the power output in case of overheating. Connecting thermistors is done according to the diagram show in Figure 21. The AX2550 is calibrated using a 10kOhm Negative Coefficient Thermistor (NTC) with the temperature/resistance characteristics shown in the table below. This 10kOhm thermistor model can be found at the component section of every Radio Shack stores under the product reference 271-110A.

TABLE 6. Recommended NTC characteristics

Temp (oC)	-25	0	25	50	75	100
Resistance (kOhm)	86.39	27.28	10.00	4.16	1.92	0.93

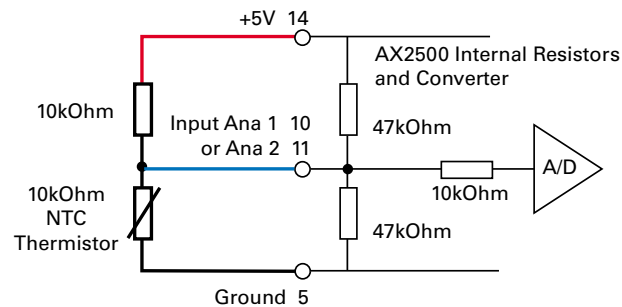


FIGURE 21. NTC Thermistor wiring diagram

Thermistor are non-linear devices. Using the circuit described on Figure 21, the controller will read the following values (represented in signed binary) according to the temperature.

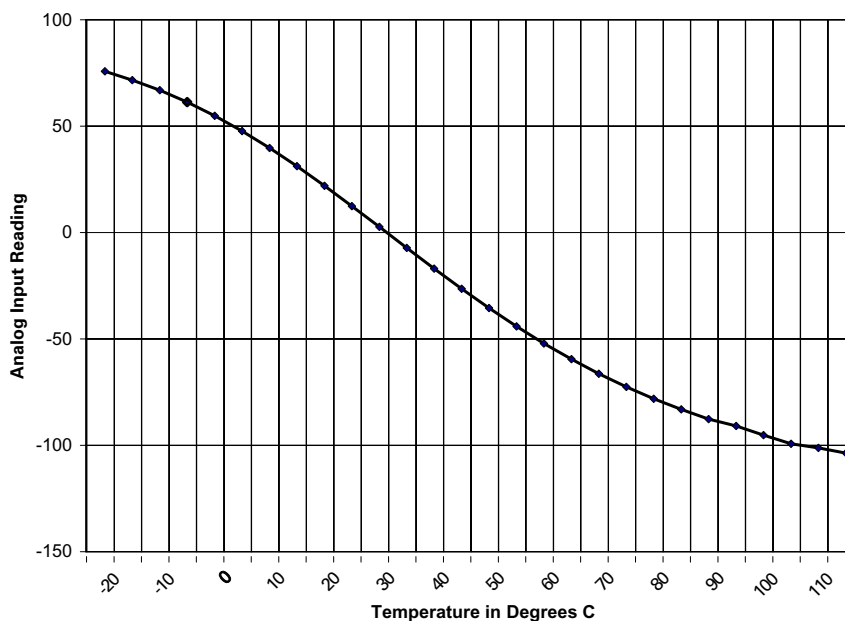


FIGURE 22. Signed binary reading by controller vs. NTC temperature

To read the temperature, use the **?p** command to have the controller return the A/D converter's value. The value is a signed 8-bit hexadecimal value. Use the chart data to convert the raw reading into a temperature value.

## Using the Analog Inputs to Monitor External Voltages

The analog inputs may also be used to monitor the battery level or any other DC voltage. In this mode, the controller does not use the voltage information but merely makes it available to the host microcomputer via the RS232 port. The recommended schematic is shown in Figure 23.

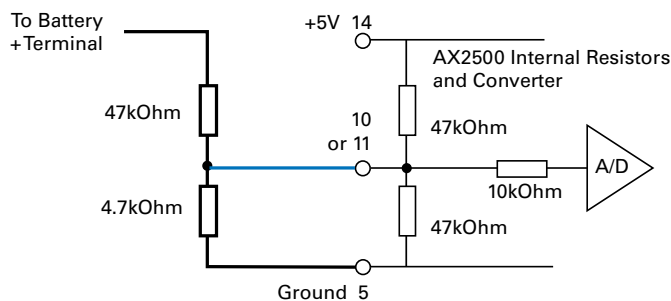


FIGURE 23. Battery voltage monitoring circuit

Using these resistor values, it is possible to measure a voltage ranging from -5V to +60V with a 0.25V resolution. The formula for converting the A/D reading into a voltage value is as follows.

$$\text{Measured volts} = ((\text{controller reading} + 128) * 0.255) - 5$$

Note: The A/D converter's reading is returned by the **?p** command and is a signed 8-bit hexadecimal value. You must add 128 to bring its range from -127/+127 to 0/255.

## Connecting User Devices to Analog Inputs

The two analog inputs can be used for any other purpose. The equivalent circuit for each input is shown in Figure 24. The converter operates with an 8-bit resolution, reporting a value of 0 at 0V and 255 at +5V. Care should be taken that the input voltage is always positive and does not exceed 5V. The converter's intrinsic diodes will clip any negative voltage or voltage above 5V, thus providing limited protection. The value of the analog inputs can be read through the controller's RS232 port.

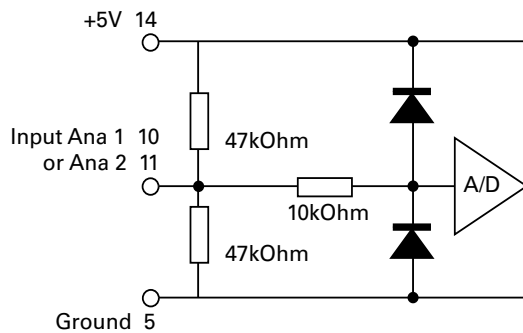


FIGURE 24. AX2550 Analog Input equivalent circuit

## Internal Voltage Monitoring Sensors

The AX2550 incorporates voltage sensors that monitor the Main Battery voltage and the Internal 12V supply. This information is used by the controller to protect it against overvoltage and undervoltage conditions (see "Overvoltage Protection" on page 30 and "Undervoltage Protection" on page 30). These voltages can also be read from the RS232 serial port using the **?e** query.

The returned value are numbers ranging from 0 to 255. To convert these numbers into a Voltage figure, the following formulas must be used:

$$\text{Measured Main Battery Volts} = 55 * \text{Read Value} / 256$$

$$\text{Measured Internal Volts} = 28.5 * \text{Read Value} / 256$$

## Internal Heatsink Temperature Sensors

The AX2550 includes temperature sensors making contact with each of the two Heat-sink sides of the controller.

These sensors are used to automatically reduce the maximum Amps that the controller can deliver as it overheats. However, the temperature can be read using the RS232 port using the `?m` query, or during data logging (see "Analog and R/C Modes Data Logging String Format" on page 93)

The analog value that is reported will range from 0 (warmest) to 255 (coldest). Because of the non-linear characteristics of NTC thermistors, the conversion from measured value to temperature must be done using the correction curve below.

It should be noted that the temperature is measured inside the controller and that it may be temporarily be different than the temperature measured outside the case.

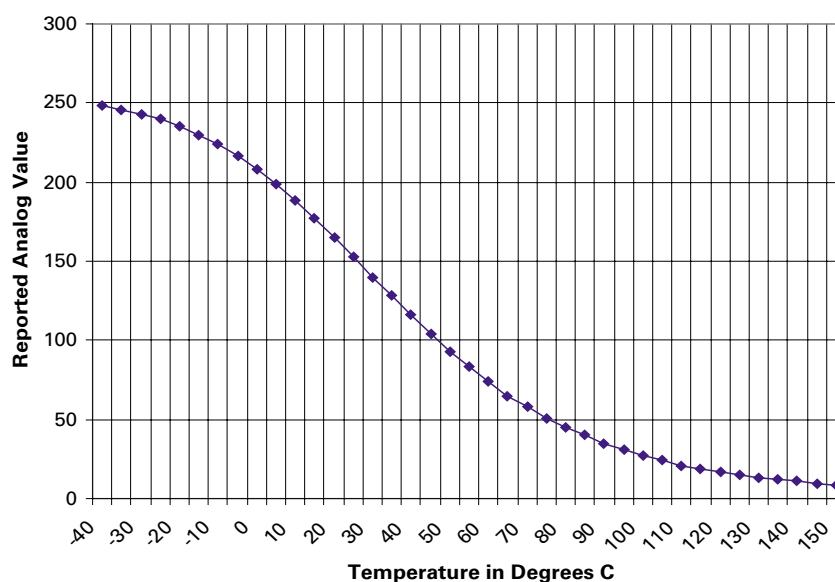


FIGURE 25. Analog reading by controller vs. internal heat sink temperature

## SECTION 5

# General Operation

This section discusses the controller's normal operation in all its supported operating modes.

---

## Basic Operation

The AX2550's operation can be summarized as follows:

- Receive commands from a radio receiver or a microcomputer
- Activate the motors according to the received command
- Perform continuous check of fault conditions and adjust actions accordingly

Multiple options are available for each of the above listed functions which can be combined to produce practically any desired mobile robot configuration.

---

## Input Command Modes

The AX2550 will accept commands from one of the following sources

- R/C radio
- Serial data (RS232)
- Analog signal (0 to 5V)

A detailed discussion on each of these modes and the available commands is provided in the following dedicated chapters: "R/C Operation" on page 63, "Serial (RS-232) Controls and Operation" on page 79, and "Analog Control and Operation" on page 97.

The controller's factory default mode is R/C radio. The mode can be changed using any of the methods described in "Programming using built-in Switches and Display" on page 119 and "Loading, Changing Controller Parameters" on page 128.

Selecting the Motor Control Modes

For each motor, the AX2550 supports multiple motion control modes. The controller’s factory default mode is Open Loop Speed control for each motor. The mode can be changed using any of the methods described in “Programming using built-in Switches and Display” on page 119 and “Loading, Changing Controller Parameters” on page 128.

Open Loop, Separate Speed Control

In this mode, the controller delivers an amount of power proportional to the command information. The actual motor speed is not measured. Therefore the motors will slow down if there is a change in load as when encountering an obstacle and change in slope. This mode is adequate for most applications where the operator maintains a visual contact with the robot.

In the separate speed control mode, channel 1 commands affect only motor 1, while channel 2 commands affect only motor 2. This is illustrated in Figure 25 below.

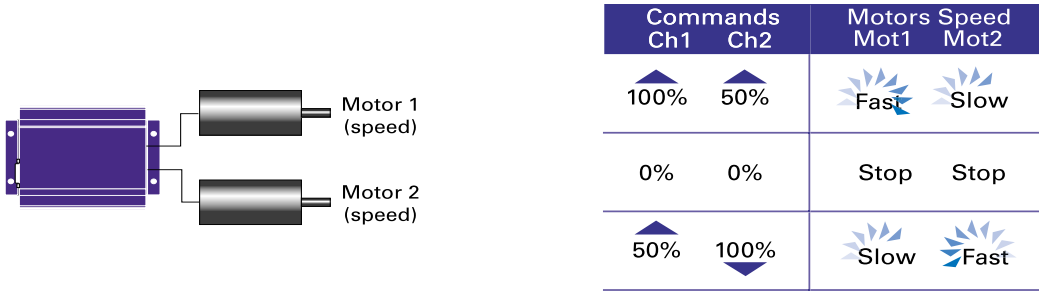


FIGURE 25. Effect of commands to motors examples in separate mode

Open Loop, Mixed Speed Control

This mode has the same open loop characteristics as the previously described mode. However, the two commands are now mixed to create a tank-like steering when one motor is used on each side of the robot: Channel 1 is used for moving the robot in the forward or reverse direction. Channel 2 is used for steering and will change the balance of power on each side to cause the robot to turn.

Figure 26 below illustrates how the mixed mode works.

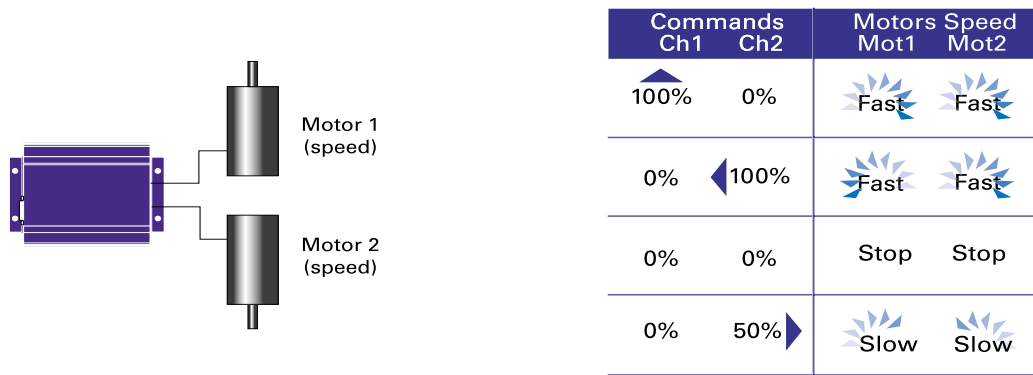


FIGURE 26. Effect of commands to motors examples in mixed mode

## Closed Loop Speed Control

In this mode, illustrated in Figure 28, an analog tachometer is used to measure the actual motor speed. If the speed changes because of changes in load, the controller automatically compensates the power output. This mode is preferred in precision motor control and autonomous robotic applications. Details on how to wire the tachometer can be found in “Connecting Tachometer to Analog Inputs” on page 40. Closed Loop Speed control operation is described in “Closed Loop Speed Mode” on page 113.

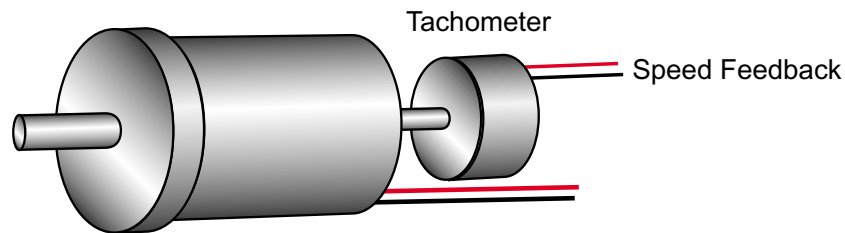


FIGURE 27. Motor with tachometer assembly for Closed Loop Speed operation

## Close Loop Position Control

In this mode, illustrated in Figure 28, the axle of a geared down motor is coupled to a potentiometer that is used to compare the angular position of the axle versus a desired position. This AX2550 feature makes it possible to build ultra-high torque “jumbo servos” that can be used to drive steering columns, robotic arms, life-size models and other heavy loads. Details on how to wire the position sensing potentiometers and operating in this mode can be found in “Position Mode” on page 105.

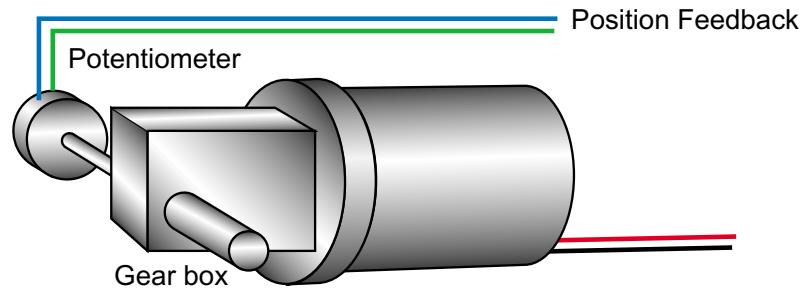


FIGURE 28. Motor with potentiometer assembly for Position operation

### SmartAmps Current Limitation

The AX2550 features active current limitation with a SmartAmps control algorithm. On the AX2550, the current limiting is based on the actual measured temperature of the heat sink, near the mounting point of the Power MOSFET transistors inside the controller. While the measured temperature is below 60°C, the controller will deliver up to 120A Continuous Amps. For every degree of temperature rise above 60°C, the limit is reduced from this absolute maximum amount. Furthermore you can select one of 6 current limit settings based on your robot's component capabilities (motors, battery, wiring), up to the controller's own limit. The current limit that is applied at any given time is the lesser value of your selected value and the maximum Amps the controller is allowed to deliver at its current temperature point. This capability ensures that the controller will be able to work safely with practically all motor types and will adjust itself automatically for the various load and environmental conditions. The chart on Figure 31 shows the maximum current limit depending on the measured heat sink temperature.

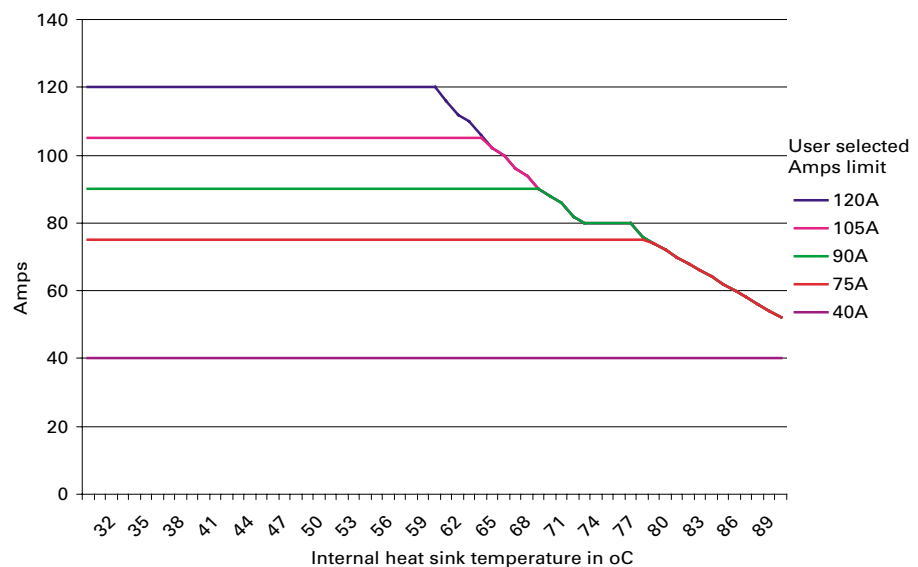


FIGURE 29. Maximum current vs. heat sink temperature at various preset limits



The time it takes for the heat sink's temperature to rise depends on the current output, ambient temperature, and available air flow (natural or forced).

In practice, assuming the maximum Amps limit setting (120A), the AX2550 will typically allow up to 1 minute of operation at 100A or 15 seconds at 120A. These numbers can be significantly improved by providing an air flow to the controller or placing it in contact with a large metallic surface on which it can conduct some of its heat.

Note that the measured temperature is on the inside of the heat sink and will rise and fall faster than the outside surface.

---

## Peak Current Protection

During exceptionally high load conditions such as motor stalls, the motors may make demands on the controller in excess of several hundred Amps. The current limiting features described above will automatically reduce the power output so that the consumption does not exceed the maximum limit.

For example, if the motor attempts to draw 1000A, the controller will throttle back the power to 10% so that the average consumption becomes 100A.

However, the controller's MOSFET transistors will see these 1000A pulses during the 10% ON time of the Pulse Width Modulation. This level of current would be damaging to the controller.

Therefore, the AX2550's will continuously compute the peak current consumption as the average measured Amps divided by the PWM ratio and automatically reduce the power output if the computed peak Amps exceeds 250A.

This current protection scheme operates in addition to the Continuous Current Limitation described above.

---

## Programmable Acceleration

When changing speed command, the AX2550 will go from the present speed to the desired one at a user selectable acceleration. This feature is necessary in order to minimize the surge current and mechanical stress during abrupt speed changes.

Acceleration can be one of 6 available preset values, from very soft(0) to very quick (6). The AX2550's factory default value is medium soft (2).

Numerically speaking, each acceleration value corresponds to a fixed percentage speed increment, applied every 16 milliseconds. The value for each setting is shown in the table below.

TABLE 7. Acceleration setting table

Acceleration Setting	%Acceleration per 16ms	Time from 0 to max speed
0	3.1%	0.51 second
1	6.3%	0.26 second
2 (default)	9.4%	0.17 second

TABLE 7. Acceleration setting table

Acceleration Setting	%Acceleration per 16ms	Time from 0 to max speed
3	12.5%	0.13 second
4	15.6%	0.10 second
5	18.8%	0.09 second

## **Important Warning**

**Depending on the robot's weight and inertia, a quick acceleration can cause considerable current surges. Always experiment with the lowest acceleration value first and settle for the slowest acceptable value.**

## **Command Control Curves**

The AX2550 can also be set to translate the joystick or RS232 motor commands so that the motors respond differently whether the joystick is near the center or near the extremes.

The controller can be configured to use one of 5 different curves independently set for each channel.

The factory default curve is a "linear" straight line, meaning that after the joystick has moved passed the deadband point, the motor's speed will change proportionally to the joystick position.

Two "exponential" curves, a weak and a strong, are supported. Using these curves, and after the joystick has moved past the deadband, the motor speed will first increase slowly, increasing faster as the joystick moves near the extreme position. Exponential curves allow better control at slow speed while maintaining the robot's ability to run at maximum speed.

Two "logarithmic" curves, a weak and a strong, are supported. Using these curves, and after the joystick has moved past the deadpoint, the motor speed will increase rapidly, and then increase less rapidly as the joystick moves near the extreme position.

The graph below shows the details of these curves and their effect on the output power as the joystick is moved from its center position to either extreme. The graph is for one joystick only. The graph also shows the effect of the deadband setting.

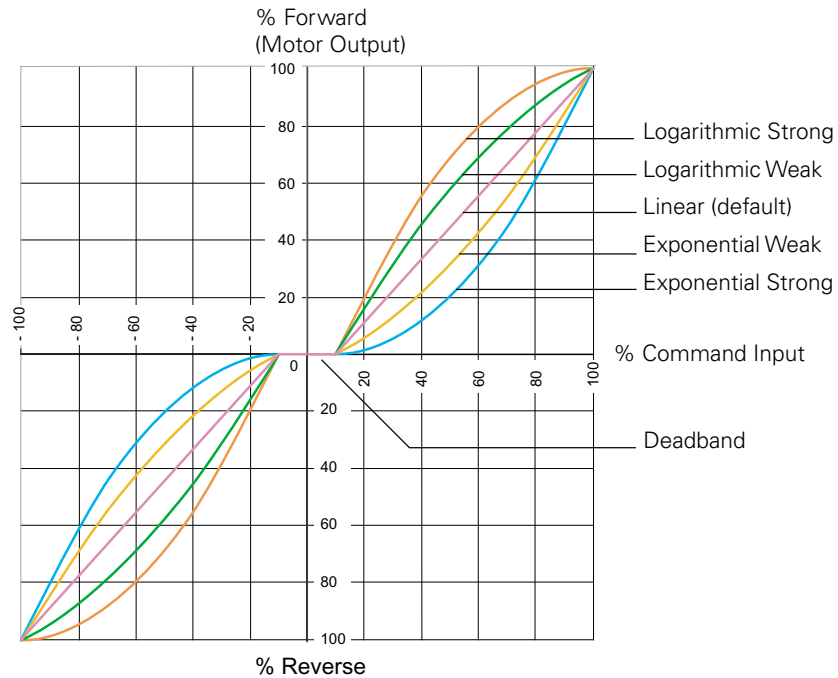


FIGURE 30. Exponentiation curves

The AX2550 is delivered with the “linear” curves selected for both joystick channels. To select different curves, you will need to change the values of “**E**” (channel 1) and “**F**” (channel 2) according to the table below. Refer to the chapter “Configuring the Controller using the Switches” on page 119, or “Using the Roborun Configuration Utility” on page 125 for instructions on how to program parameters into the controller.

TABLE 8. Exponent selection table

Exponentiation Parameter Value	Selected Curve
<b>E</b> or <b>F</b> = 0	Linear (no exponentiation) - default value
<b>E</b> or <b>F</b> = 1	strong exponential
<b>E</b> or <b>F</b> = 2	normal exponential
<b>E</b> or <b>F</b> = 3	normal logarithmic
<b>E</b> or <b>F</b> = 4	strong logarithmic

## Left / Right Tuning Adjustment

By design, DC motors will run more efficiently in one direction than the other. In most situations this is not noticeable. In others, however, it can be an inconvenience. When operating in open loop speed control, the AX2550 can be configured to correct the speed in one direction versus the other by as much as 10%. Unlike the Joystick center trimming tab that

is found on all R/C transmitters, and which is actually an offset correction, the Left/Right Adjustment is a true multiplication factor as shown in Figure 30

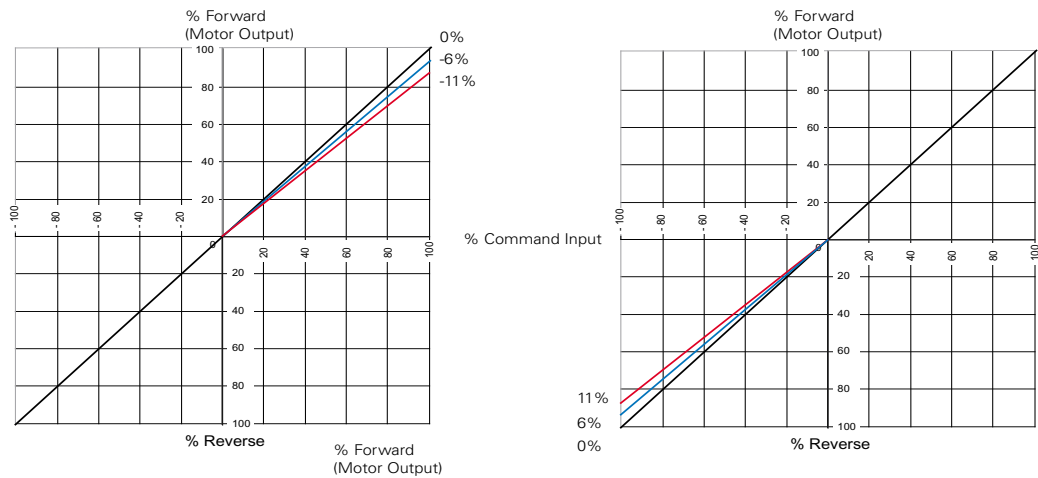


FIGURE 31. Left Right adjustment curves

The curves on the left show how a given forward direction command value will cause the motor to spin 6 or 11 % slower than the same command value applied in the reverse direction. The curves on the right show how the same command applied to the forward direction will cause the motor to spin 6 to 11% faster than the same command applied in the reverse direction. Note that since the motors cannot be made to spin faster than 100%, the reverse direction is the one that is actually slowed down.

In applications where two motors are used in a mixed mode for steering, the Left/Right Adjustment parameter may be used to make the robot go straight in case of a natural tendency to steer slightly to the left or to the right.

The Left/Right adjustment parameter can be set from -11 % to +11 % in seven steps of 1.5%. See "Programmable Parameters List" on page 122 and "Loading, Changing Controller Parameters" on page 128 for details on how to adjust this parameter.

The Left/Right adjustment is performed in addition to the other command curves described in this section. This adjustment is disabled when the controller operates in any of the supported closed loop modes.

TABLE 9. Left/Right Adjustment Parameter selection

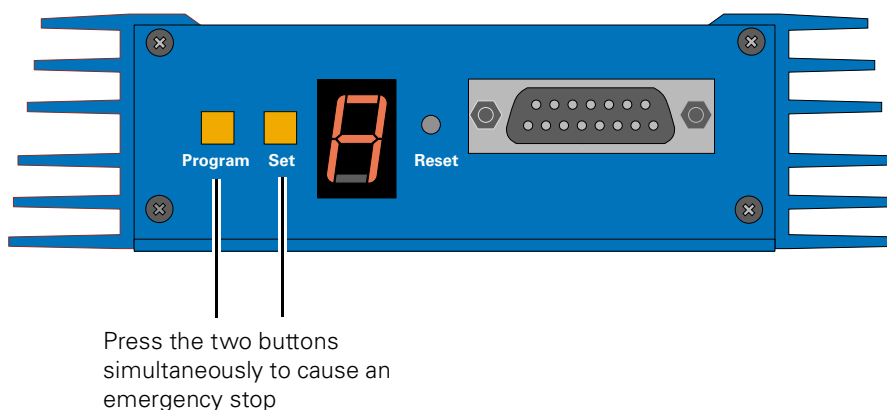
Parameter Value	Speed Adjustment	Parameter Value	Speed Adjustment
		7	None (default)
0	-10.5%	8	1.5%
1	-9%	9	3%
2	-7.5%	10	4.5%
3	-6%	11	6%
4	-4.5%	12	7.5%

TABLE 9. Left/Right Adjustment Parameter selection

Parameter Value	Speed Adjustment	Parameter Value	Speed Adjustment
5	-3%	12	9%
6	-1.5%	14	10.5%

## Emergency Shut Down Using Controller Switches

In case of emergency, it is possible to cause the controller to cut off the power to the motors by depressing the Program and Set buttons simultaneously as shown in the figure below. The controller will stop immediately without delay.



**After and Emergency Stop condition, the controller must be reset or powered Off and On to resume normal operation.**

## Emergency Stop using External Switch

An external switch can be added to the AX2550 to allow the operator to stop the controller's output in case of emergency. This controller input can be configured as the "Inverted" detection instead of Emergency Stop. The factory default for this input is "No Action".

The switch connection is described in "Connecting Switches or Devices to EStop/Invert Input" on page 38. The switch must be such that it is in the open state in the normal situation and closed to signal an emergency stop command.

**After and Emergency Stop condition, the controller must be reset or powered Off and On to resume normal operation.**

## Inverted Operation

For robots that can run upside-down, the controller can be configured to reverse the motor commands using a gravity activated switch when the robot is flipped. This feature is

enabled only in the mixed mode and when the switch is enabled with the proper configuration of the "Input switch function" parameter. See "Programmable Parameters List" on page 122.

The switch connection is described in "Connecting Switches or Devices to EStop/Invert Input" on page 38. The switch must be such that it is in the open state when the robot is in the normal position and closed when inverted. When the status of the switch has changed, the controller will wait until the new status has remained stable for 0.5s before acknowledging it and inverting the commands. This delay is to prevent switch activation triggered by hits and bounces which may cause the controller to erroneously invert the commands.

---

## External Devices Overheat Monitoring

Using low cost and readily available thermistors connected to its analog input ports, the AX2550 can be made to monitor the motor's or any other component's temperature and can trigger an overheat alarm when it reaches a programmable threshold value.

When such an overheat situation is detected, the controller will respond with one of four user-selectable actions:

- Stop Immediately
- Run at 50% power for an additional 30 seconds and stop if the temperature has not cooled off below the programmed threshold oC. This is the factory default mode.
- Run at 25% power for and additional 2 minutes and stop if the temperature has not cooled off below the programmed threshold
- No action

The overheat threshold may be any value between 60 and 120 oC, or be disabled altogether. There is only one temperature threshold setting which will apply equally to both temperature probes.

The wiring of the thermistors is show in "Connecting External Thermistor to Analog Inputs" on page 41.

The threshold temperature value can only be enabled and set using the PC configuration utility. See "Loading, Changing Controller Parameters" on page 128 for details on how to set this parameter.

The controller will resume normal operation when the overhead condition disappears.

---

## Special Use of Accessory Digital Inputs

The AX2550 includes two general purpose digital inputs identified as Input E and Input F. The location of these inputs on the DB15 connector can be found in the section "I/O List and Pin Assignment" on page 34, while the electrical signal needed to activate them is shown on page 36.

By default, these inputs are ignored by the controller. However, the AX2550 may be configured to cause either of the following actions:

- Activate the buffered Output D

- or Turn Off/On the power MOSFET transistors

These alternate modes can only be selected using the Roborun Utility (see “Controls Settings” on page 129. Each of these modes is detailed below.

### Using the Inputs to Activate the Buffered Output

When this setting is selected, the buffered Output C will be On when the Input line is pulled to Ground (0V). The Output will be Off when the Input is pulled high.

This function makes it possible to drive solenoids or other accessories up to 2A at 24V using a very low current switch, for example.

### Using the Inputs to turn Off/On the Power MOSFET transistors

When this setting is selected, the controller’s Power MOSFET transistors will be active, and the controller will be operating normally, only when the input is pulled to ground.

When the input is pulled high, all the power MOSFETs are turned Off so that the motors are effectively disconnected from the controller.

This function is typically used to create a “dead man switch” when the controller is driven using a analog joystick. The motors will be active only while the switch is depressed. If the switch is left off for any reason, the motors will be disconnected and allowed to freewheel rather than coming to an abrupt stop.

## Self-Test Mode

The AX2550 incorporates a simple Self-Test mode that performs the following functions:

- Display the software revision number on the LED display
- Ramp each motor up and down in both directions
- Internal parameters on the serial port output

The Self Test mode can be conveniently initiated using only the controller’s switches so that no radio or computer is needed.

To enter the Self Test mode, press and hold the **Set** button while resetting or powering up the controller.

After a few seconds, the LEDs will display a sequence of two numerical digits and an optional letter separated by dashes as shown in the examples below.



FIGURE 32. Press and hold “Set” to display version number and enter self-test

After these digits are displayed, the controller will attempt to power the motors. Motor 1 will be ramped from stop to full speed forward, to full speed reverse and back to stop. Then the same operation will repeat on motor 2. After both motors have completed their ramps, the software revision will be displayed again and the motors will be ramped again. This sequence will repeat itself indefinitely until the controller is powered off or reset.

While in the Self Test mode, the AX2550 will continuously send a string of characters on the RS232 output line. This string will contain 12 two-digits hexadecimal number representing the 12 following operating parameters.

- Captured R/C Command 1 and 2
- Power Applied to Controller's output stage
- Values applied to Analog inputs 1 and 2
- Amps on channel 1 and 2
- Internal Heat Sink temperatures 1 and 2
- Main Battery voltage
- Internal 12V voltage

The entire string is repeated every 213 milliseconds with the latest internal parameter values. This information can be logged using the Roborun Utility ( see "Viewing and Logging Data in Analog and R/C Modes" on page 137).

The string and data format is described in "Analog and R/C Modes Data Logging String Format" on page 92.

## **Important Warning**

**Disconnect the Motor Power (thick red wires) from the battery and power the controller by applying 12V on the thin yellow wire if you do not wish the motors to be activated while in Self Test mode.**

This could be the case if you only wish to use the Self Test sequence to read the software revision number or to monitor the joystick capture.



## SECTION 6

# Normal and Fault Condition LED Messages

This section discusses the meaning of the various messages and codes that may be displayed on the LED display during normal operation and fault conditions.

---

## Use of the LED Display

The AX2550 uses a single 7-segment LED display to report a number of operating or fault conditions. The type of reported information depends on the controller's operating context:

During normal motor operation:

- Motor direction

During Parameter Settings

- Selected parameter and its value

In R/C mode with Radio off

- No control message

During Error condition

- Error condition (overheat, emergency stop, short circuit)

During Self-Test mode

- Software revision number
- Motor direction

Motor Direction Status

When the controller is running, two pairs of LED segments are directly related to command informations to the Power Output stage. The position and meaning of the segments is shown in the figure below.

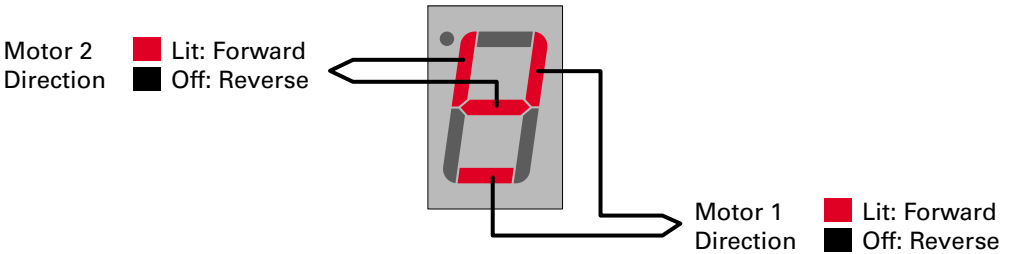


FIGURE 1. Each command bit is wired to 2 LED segments








Note that the display does not provide Power information. Therefore it is possible that the motor be stopped while the display indicates that the direction is forward. In such a situation is set to apply the power in the forward direction to the output stage but the motor is stopped because the applied power is zero.

The LED can display a total of 5 patterns summarized in Table 11.

TABLE 11. Motor Commands and resulting display

Possible Display	Motor 1	Motor 2
	Forward	Forward
	Reversed	Forward

TABLE 11. Motor Commands and resulting display

Possible Display	Motor 1	Motor 2
	 Forward	 Reversed
	 Reversed	 Reversed
	<b>MOSFET Transistors are OFF</b> Motors are freewheeling Will occur during Overvoltage or Undervoltage conditions (see page 29)	

## Fault Messages

The AX2550 uses the LED display to report fault conditions. When these messages are displayed, the motors are normally stopped.

### No Control

This message is displayed in the R/C mode to indicate that no valid radio signal has been detected at its inputs, or that radio signal has been lost. When the controller is configured in the mixed mode, signals on channel 1 and channel 2 must be present to enable the controller. When the controller is configured in the separate mode, a signal received on either channel will enable the controller. When enabled, the controller will display the normal motor status described previously.

The “no control” messages is displayed using the following sequence of digits.



FIGURE 32. No Radio signal scrolling message

## Emergency Stop

The “Emergency Stop” message is displayed after the motors have stopped following an emergency stop condition triggered by any of the methods described in Figure , “Emergency Shut Down Using Controller Switches,” on page 52. The “Emergency Stop” message is displayed using the following sequence of digits.



FIGURE 33. Emergency Stop scrolling message

## Heat

The “Overheat” message is displayed after the motors have stopped following a controller overheat condition. See Figure , “External Devices Overheat Monitoring,” on page 54 for a detailed description of the controller’s behavior in case of overheat. The “Overheat” message is displayed using the following sequence of digits.



FIGURE 34. Overheat condition scrolling message

## Self-Test Display

Self test is a special condition that is entered by holding the Program and Set button immediately after power-on or reset. During self test, the LEDs will display the controller’s software revision number by flashing a sequence of digits as shown in the figure below. Then the each motor will, in turn, be ramped to maximum forward and maximum reverse. When the motors are operating, the LED will display one of the patterns described in Table 11. The cycle will repeat indefinitely until the controller is powered Off or reset.

 = Software version 1.2c

 = Software version 2.0

FIGURE 35. Example of Software revision number display

---



## SECTION 7

# R/C Operation

This section describes the controller's wiring and functions specific to the R/C radio control mode.

## Mode Description

The AX2550 can be directly connected to an R/C receiver. In this mode, the speed or position information is contained in pulses whose width varies proportionally with the joysticks' positions. The AX2550 mode is compatible with all popular brands of R/C transmitters. A third R/C channel can be used to control the On/Off state of two outputs that may be connected to electrical accessories (valves, lights, weapons,...)

The R/C mode provides the simplest method for remotely controlling a robotic vehicle: little else is required other than connecting the controller to the R/C receiver (using the provided cable) and powering it On. For better control and improved safety, the AX2550 can be configured to perform correction on the controls and will continuously monitor the transmission for errors.

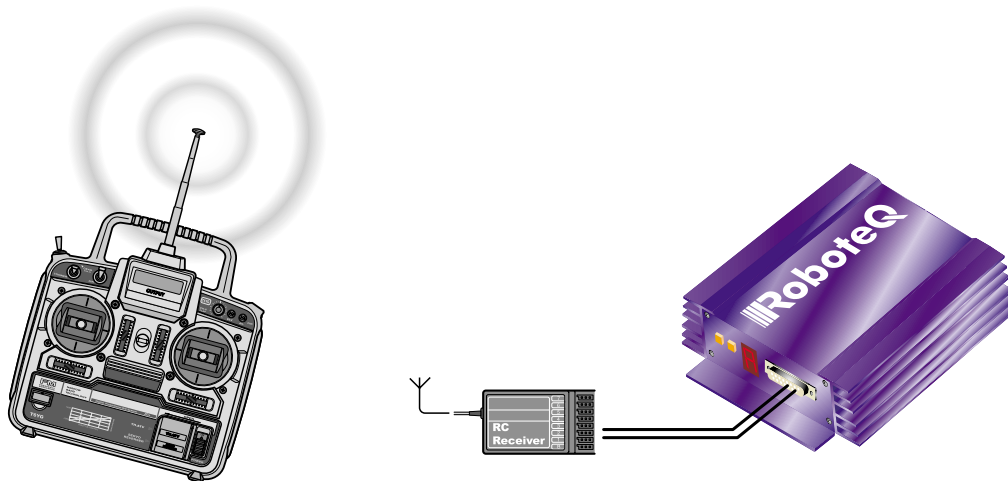


FIGURE 36. R/C radio control mode

## Selecting the R/C Input Mode

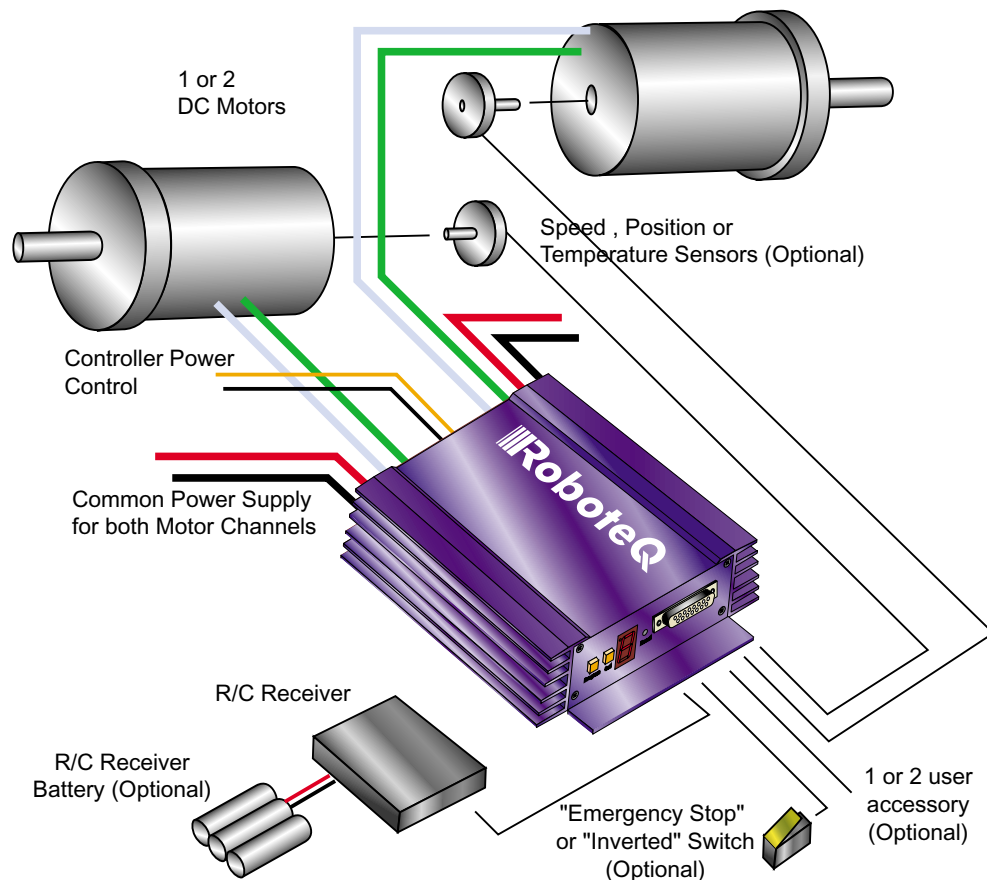
The R/C Input Mode is the factory default setting.

If the controller has been previously set to a different Input Mode, it will be necessary to reset it to the R/C mode using one of the following methods:

- Restoring the factory defaults by pressing and holding the **Program** and **Set** buttons while powering on the controller until the LED display flashes
- Setting the "I" parameter to the value "0" using one of several methods described in the chapters "Configuring the Controller using the Switches" on page 119, "Using the Roborun Configuration Utility" on page 125, and "RS232 Accessible Parameter Table" on page 90.

## Typical Wiring

The diagram below shows all the required and optional connections when using the R/C mode.



The required connections are the following:

- a 12V to 40V to Main Battery



- at least one motor
- optional 12V backup power supply for the controller
- R/C radio

See “Connecting Power and Motors to the Controller” on page 23. The connection to the R/C radio is further detailed in this chapter.

The optional connections are the following:

- optional 12V backup power supply for the controller
- the speed or position sensors required for closed loop speed or position operation
- 1 or 2 electrical accessory (valve, weapon, solenoid, ...)
- thermistor for motor temperature monitoring
- gravity actuated switch for detecting inverted operation
- manual switch for emergency stop
- the R/C receiver’s battery may also be omitted by using the power output available on the controller’s connector

See “Connecting Sensors and Actuators to Input/Outputs” on page 31.

## Connector I/O Pin Assignment (R/C Mode)

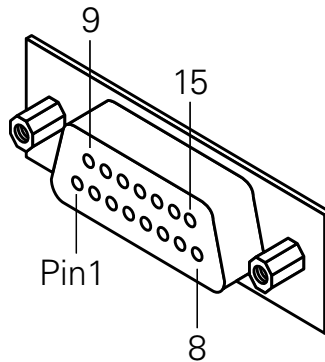


FIGURE 37. Pin locations on the controller’s 15-pin connector

When used in R/C mode, the pins on the controller’s DB15 connector are mapped as described in the table below

TABLE 12. Connector pin-out in R/C mode

Pin Number	Signal	Input or Output	Description
1	Output C	Output	2A Accessory Output C (same as pin 9)
2	RS232 Out	Output	Optional. Used for Data Logging
3	R/C Channel 1	Input	Channel 1 input pulse (isolated)

TABLE 12. Connector pin-out in R/C mode

Pin Number	Signal	Input or Output	Description
4	R/C Channel 2	Input	Channel 1 input pulse (isolated)
5	Ground	Power	Controller ground (-)
6	R/C -	Power Input	R/C Battery input - (isolated)
7	R/C +	Power Input	R/C Battery input + (isolated)
8	R/C Channel 3	Input	Channel 3 input for On/Off accessories
9	Output C	Output	2A Accessory Output C (same as pin 1)
10	Speed/Pos/T 1	Analog in	Channel 1 speed, position or temp feedback
11	Speed/Pos/T 2	Analog in	Channel 2 speed, position or temp feedback
12	Output D	Output	Low Current Accessory Output D
13	Ground	Power	Controller ground (-)
14	+5V	Power Output	+5V Power Output (100mA max.)
15	Switch Input	Input	Emergency Stop or Invert Switch input

## R/C Input Circuit Description

The AX2550 includes an optical isolation barrier on the Channel 1 and Channel 2 R/C radio inputs. Figure 38 shows an electrical representation of the R/C input circuit. The right side is powered from the controller's batteries. The R/C signals 1 and 2 activate a LED whose light then triggers the photo transistors on the microcontroller side. The left side of the opto isolators include an amplifier and thus need to be powered separately. Notice that channel 3 is optional and not opto-isolated.

Optical isolation is designed to protect the R/C radio from damage that may occur inside the controller, and vice-versa. Additionally, the isolation barrier will help prevent some of the electrical "noise" generated inside the controller by the Microcontroller and the high power switching from reaching and interfering with the radio.

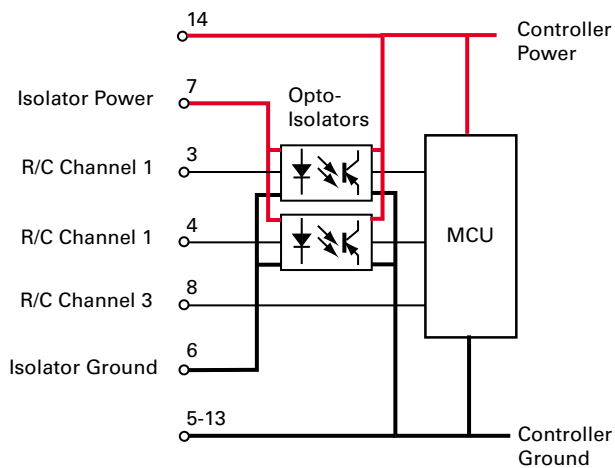


FIGURE 38. AX2550 R/C Input equivalent circuit

## Cabling to R/C Receiver using Full Opto-Isolation

Full opto-isolation is achieved when the radio's power and ground are totally separated from the controller's power and ground.

Figure 39 shows the cabling of the R/C radio to the controller when full optical isolation is required.

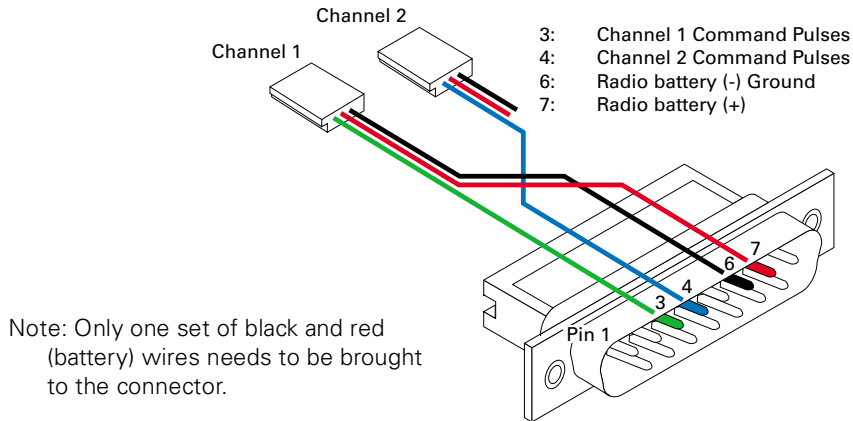


FIGURE 39. Channel 1 & 2 wiring using external battery and full optical isolation

The opto isolators must be powered with a 5V supply for their built-in amplifiers to work. This power is normally supplied from the R/C radio's battery as detailed in the diagram on Figure 40.

Note that if full optical isolation is required, it is not possible to connect the third R/C channel to the radio. Doing so will create a potential path for electricity between the controller and radio, thus breaking the optical barrier.

This wiring, with a separate R/C battery, is the preferred method for connecting the R/C radio.

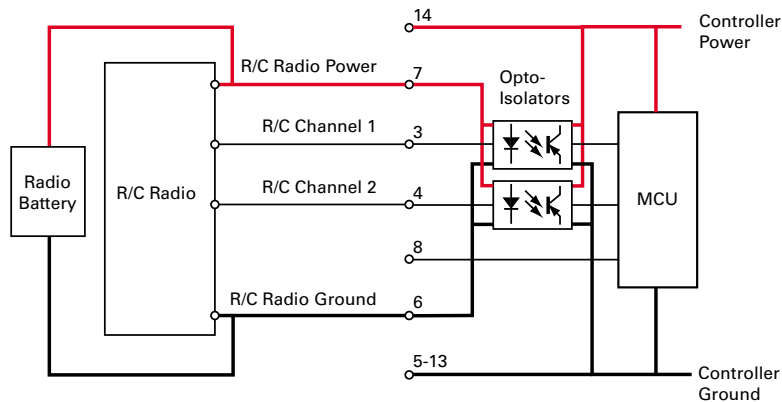


FIGURE 40. Fully opto-isolated connection to R/C radio

## Cabling to R/C Receiver with Partial Opto-Isolation

This wiring option should be considered only when the Accessory R/C channel (Channel 3) is required. Since Channel 3 is directly connected to the AX2550's microcontroller without opto coupler, it is necessary to connect the radio and controller grounds together to create a common ground reference for the signal. The radio will be powered from its own separate battery using this wiring scheme.

By joining the grounds and having the R/C signal connected directly from the radio to the controller, the full electrical barrier is lost. However, since the radio's +5V and controller power voltage are still separate, some protection remains should the controller fail and accidentally generate a high voltage on its +5V output.

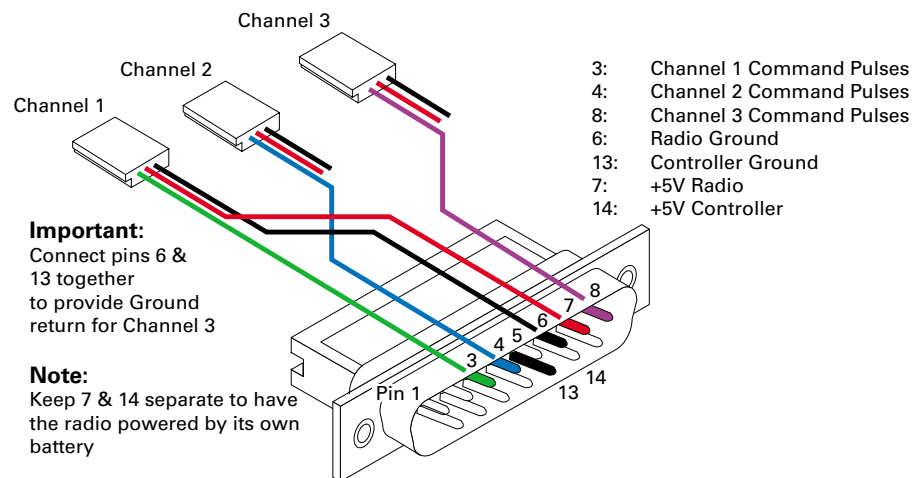


FIGURE 41. Wiring when Ch 3 is used with radio powered by its own separate battery

Figure 41 shows the cabling of the R/C radio to the controller when Channel 3 is used and partial optical isolation is desired. Figure 42 shows the equivalent electrical diagram.

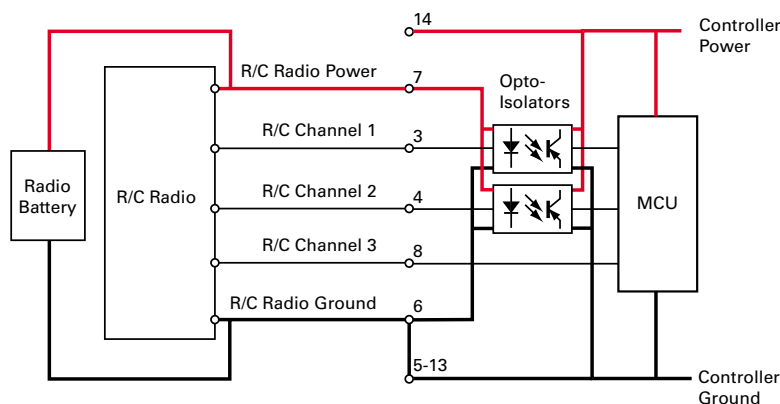


FIGURE 42. Partial opto-isolation with Channel 3 electrical diagram

## Powering the Radio from the controller

The 5V power and ground signals that are available on the controller's connector may be used to power the R/C radio. Figure 43 below shows the connector wiring necessary to do this. Figure 44 shows the equivalent electrical diagram.

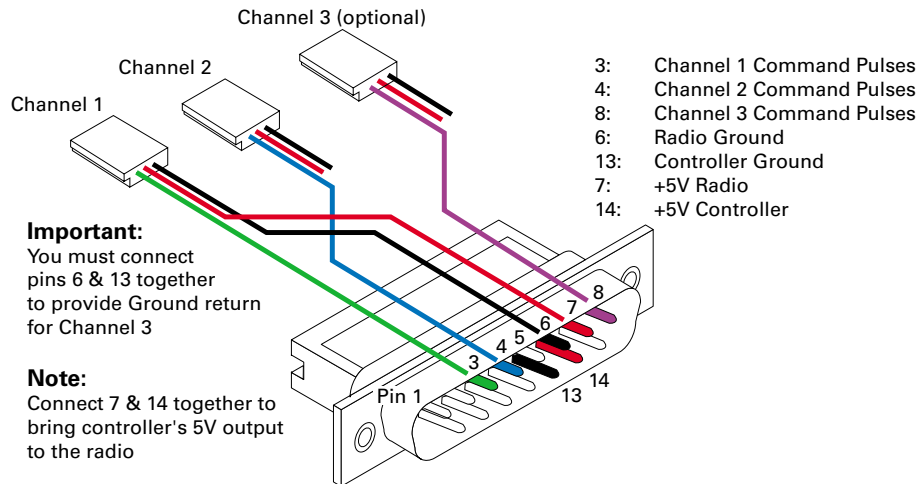


FIGURE 43. Wiring for powering R/C radio from controller

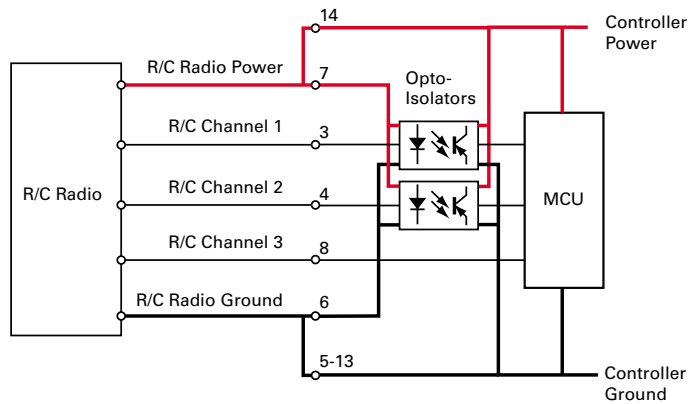


FIGURE 44. R/C Radio powered by controller electrical diagram

Notice that in this configuration, the controller's +5V power and ground are connected to the floating (opto-isolated) R/C power inputs so that now the radio and the controller are no longer optically isolated.

Losing the optical isolation increase the risks that damage in the controller will reach the radio, although these risks remain extremely low. Without optical isolation, the radio is less immune to electrical "noise" and therefore may operate at a reduced range.

This mode of operation is the most convenient and is the one wired in the R/C cable delivered with the controller.

## Operating the Controller in R/C mode

In this operating mode, the AX2550 will accept commands from a Radio Control receiver used for R/C models remote controls. The speed or position information is communicated to the AX2550 by the width of a pulse from the R/C receiver: a pulse width of 1.0 millisecond indicates the minimum joystick position and 2.0 milliseconds indicates the maximum joystick position. When the joystick is in the center position, the pulse should be 1.5ms.

Note that the real pulse-length to joystick-position numbers that are generated by your R/C radio may be different than the ideal 1.0ms to 2.0ms discussed above. To make sure that the controller captures the full joystick movement, the AX2550 defaults to the timing values shown in Figure 45.

For best control accuracy, the AX2550 can be calibrated to capture and use your radio's specific timing characteristics and store them into its internal Flash memory. This is done using a simple calibration procedure described on page 74.

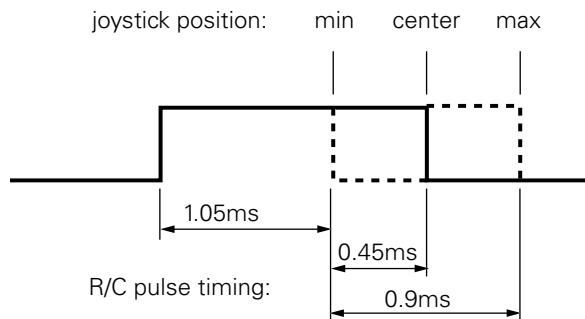


FIGURE 45. Joystick position vs. pulse duration default values

The AX2550 has a very accurate pulse capture input and is capable of detecting changes in joystick position (and therefore pulse width) as small as 0.4%. This resolution is superior to the one usually found in most low cost R/C transmitters. The AX2550 will therefore be able to take advantage of the better precision and better control available from a higher quality R/C radio, although it will work fine with lesser expensive radios as well.

Internally, the measured pulse width is compared to the reference minimum, center and maximum pulse width values. From this is generated a number ranging from -127 (when the joystick is in the min. position), to 0 (when the joystick is in the center position) to +127 (when the joystick is in the max position). This number is then used to set the motors' desired speed or position that the controller will then attempt to reach.

For best results, reliability and safety, the controller will also perform a series of corrections, adjustments, and checks to the R/C commands, as described in the following sections.

## Reception Watchdog

Immediately after it is powered on, if in the R/C mode, the controller is ready to receive pulses from the R/C radio and move the motors accordingly.

If no pulses are present, the motors are disabled, and the controller's display will scroll alternatively the letters "no ctrl" as shown in Figure 46 below.



FIGURE 46. "No control" message will scroll when no valid radio signal is present

After powering on the R/C radio receiver and transmitter, and if the wiring is correct, the controller will start receiving pulses. For a preset amount of time, the controller will monitor the pulse train to make sure that they are regular and therefore genuine R/C radio command pulses. After that, the motors are enabled and the LEDs will display a pattern related to the actual motor direction (see Figure , "Motor Direction Status," on page 58).

This power-on Watchdog feature prevents the controller from becoming active from parasite pulses and from moving the motors erratically as a result.

Similarly, if the pulse train is lost while the motors were enabled, the controller will wait a short preset amount of time before it disables the motors. If the pulses reappear during that time, the controller continues without any breaks. If the communication is confirmed lost, the "no ctrl" message is displayed again.

Note: the Accessory Outputs C and D will be turned Off when radio is lost.

## Important Notice about PCM Radios

**PCM radios have their own watchdog circuitry and will output a signal (normally a "safe condition" value) when radio communication is lost. This signal will be interpreted by the AX2550 as a valid command and the controller will remain active. To benefit from the AX2550's radio detection function, you will need to disable the your PCM radio watchdog.**

## R/C Transmitter/Receiver Quality Considerations

As discussed earlier in this chapter, the AX2550 will capture the R/C's command pulses with great accuracy. It will therefore be able to take advantage of the more precise joysticks and timings that can be found in higher quality R/C radio, if such added precision is desired in your application.

Another important consideration is the R/C receiver's ability to operate in an electrically noisy environment: the AX2550 switches high current at very high frequencies. Such transients along long battery and motor wires will generate radio frequency noise that may interfere with the R/C radio signal. The effects may be reduced remote control range and/or induced errors in the command pulse resulting in jerky motor operation.

A higher quality PCM R/C transmitter/radio is recommended for all professional applications, as these are more immune to noise and interference.

While a more noise-immune radio system is always desirable, it is also recommended to layout the wiring, the controller, radio and antenna so that as little electrical noise is generated in the first place. Section "Electrical Noise Reduction Techniques" on page 28 provides a few suggestions for reducing the amount of electrical noise generated in your robot.

## Joystick Deadband Programming

In order to avoid undesired motor activity while the joysticks are centered, the AX2550 supports a programmable deadband feature. A small deadband is set into the controller by default at the factory. This deadband can be stretched, reduced or eliminated by changing the "**d**" parameter using one of the three methods described in the chapter "Configuring the Controller using the Switches" on page 119. The AX2550 has 8 preset deadband values coded 0 to 7. The value 0 disables the deadband. Other values select a deadband according to the table below. The deadband value applies equally to both joysticks

The deadband is measured as a percentage of total normal joystick travel. For example, a 16% deadband means that the first 16% of joystick motion in either direction will have no effect on the motors. If the joystick is recalibrated to operate using a shorter travel (see "Joystick Calibration" on page 73), the percent value in the table will not be as accurate.

TABLE 13. Selectable deadband values

Deadband Parameter Value	Deadband as Percent of full Joystick Travel
<b>d</b> = 0	No deadband
<b>d</b> = 1	8%
<b>d</b> = 2	16% - default value
<b>d</b> = 3	24%
<b>d</b> = 4	32%
<b>d</b> = 5	40%
<b>d</b> = 6	46%
<b>d</b> = 7	54%

Note that the deadband only affects the start position at which the joystick begins to take effect. The motor will still reach 100% when the joystick is at its full position. An exaggerated illustration of the effect of the deadband on the joystick action is shown in the Figure 47 below.



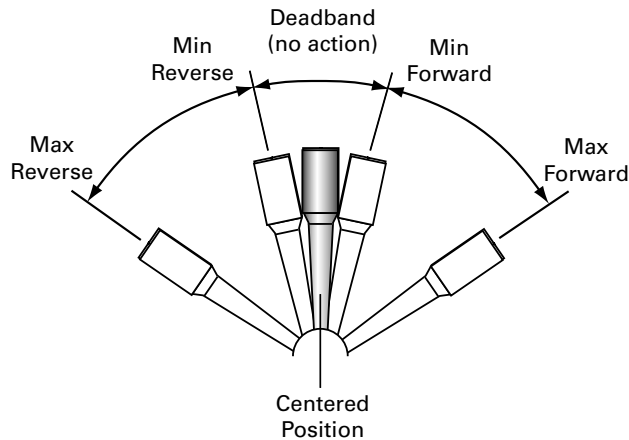


FIGURE 47. Effect of deadband on joystick position vs. motor speed

---

## Command Control Curves

The AX2550 can also be set to translate the joystick motor commands so that the motors respond differently whether the joystick is near the center or near the extremes. Five different exponential or logarithmic translation curves may be applied. Since this feature applies to both R/C and RS232 mode, it is described in detail in “Command Control Curves” on page 50, in the General Operation section of this manual.

## Left/Right Tuning Adjustment

When operating in mixed mode with one motor on each side of the robot, it may happen that one motor is spinning faster than the other one at identically applied power, causing the vehicle to pull to the left or to the right.

To compensate for this, the AX2550 can be made to give one side up to 10% more power than the other at the same settings. This capability is described in detail in “Left / Right Tuning Adjustment” on page 51, in the General Operation section of this manual.

## Joystick Calibration

This feature allows you to program the precise minimum, maximum and center joystick positions of your R/C transmitter into the controller’s memory. This feature will allow you to use the full travel of your joystick (i.e. minimum = 100% reverse, maximum = 100% forward). It also ensures that the joystick’s center position does indeed correspond to a “0” motor command value.

Joystick calibration is also useful for modifying the active joystick travel area. For example, the figure below shows a transmitter whose joystick’s center position has been moved back so that the operator has a finer control of the speed in the forward direction than in the reverse position.

There are two methods for calibrating the joysticks:

- The automatic method is the simplest and is based on the controller “learning” the joystick’s parameters using the simple training sequence described below.
- The manual method lets you enter actual timing numbers directly in the controllers flash memory using your PC running the Roborun configuration utility. This method is described in “Loading, Changing Controller Parameters” on page 128.

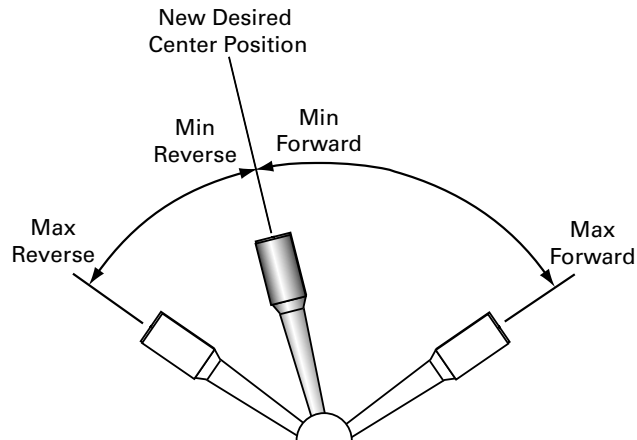


FIGURE 48. Calibration example where more travel is dedicated to forward motion

## Automatic Joystick Calibration

To calibrate the joystick(s) follow these steps:

- Press and hold the **Program** button while resetting or power up the controller
- After 10 seconds, the controller will enter the Program mode and flash the first parameter (**I**) and its value (**0** if set R/C mode)
- Turn the R/C transmitter and receiver On
- Press the **Program** button several times until the letter “**J**” is displayed. If you missed it, keep pressing the **Program** button for another cycle.
- When the display flashes the letter “**J**” followed by the “-” sign, press the **Set** button
- When the display flashes the letter “**J**” followed by “**o**”, the controller has entered the joystick calibration mode
- Move each joystick to the desired minimum and maximum position several times
- Move back the joystick(s) to the desired center position
- With the joystick(s) in the center position, press the **Program** button to record the change permanently in the controller’s Flash memory.
- Restart the controller by pressing the **Reset** button or cycling the power.

To reset the controller to factory default or to program the joystick’s positions using numerical values, see “Configuring the Controller using the Switches” on page 119 and “Using the Roborun Configuration Utility” on page 125.

Notes:

If you attempt to calibrate the joysticks while the radio is off or not connected to the controller, the calibration data will not change and the previously stored information will continue to be used.

If calibration is performed with only one R/C channel connected to the controller, then only the joystick that is active will be calibrated. The other channel will keep its original settings.

A minimum amount of travel is required between the min, max and center joystick positions. If, while calibrating, the joystick has not been moved far enough from either side of the center position, the controller will automatically include a minimum of travel to ensure proper and safe operation. In most cases, this creates no undesired effect to the driving characteristics of the robot.

R/C calibration only applies to the channel 1 and channel 2 inputs. The accessory activation channel (channel 3) is preset at the factory and cannot be changed.

## **Important Notice**

**To ensure that only stable pulses are present, the R/C transmitter and radio must be On before entering joystick calibration.**

---

## **Activating the Accessory Outputs**

The AX2550 has two general purpose outputs that may be turned on and off using a third R/C channel on the radio.

Output C is a buffered output capable of driving a 2A device at up to 24V. Output D is a non-buffered output that requires an external relay to drive high current loads. Details on how to wire these outputs to user accessories can be found at “Connecting Sensors and Actuators to Input/Outputs” on page 31.

Both outputs are controlled from a single joystick on Channel 3.

Output C is activated by pushing the joystick to the maximum position. The output turns back off when the joystick is returned to the center position.

Output D is a toggle output. When the joystick is moved from the center to the minimum position, the output changes state (to On if it was Off, or to Off if it was On). Beware that because this is a toggling output, it may be accidentally activated by noise in the radio reception. Use this output to drive safe accessories. Use only with a good quality radio, such as a PCM radio.

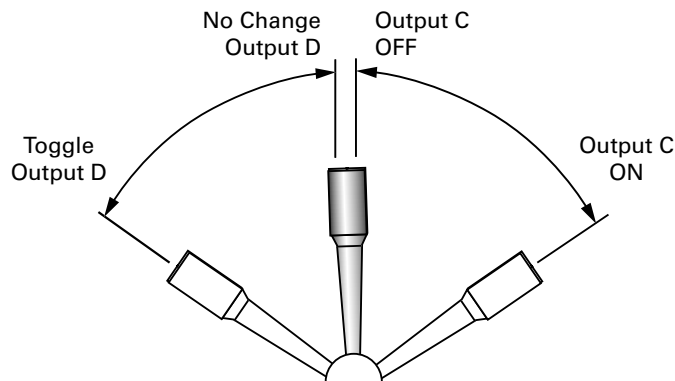


FIGURE 49. Using Channel 3 to activate accessory outputs

## Data Logging in R/C Mode

While in R/C Mode, the AX2550 will continuously send a string of characters on the RS232 output line. This string will contain 12 two-digits hexadecimal number representing the 12 following operating parameters.

- Captured R/C Command 1 and 2
- Power Applied to Controller's output stage
- Values applied to Analog inputs 1 and 2
- Amps on channel 1 and 2
- Internal Heat Sink temperatures 1 and 2
- Main Battery voltage
- Internal 12V voltage

The entire string is repeated every 213 milliseconds with the latest internal parameter values. This information can be logged using the Roborun Utility (see "Viewing and Logging Data in Analog and R/C Modes" on page 137). It may also be stored in a PDA that can be placed in the mobile robot.

The string and data format is described in "Analog and R/C Modes Data Logging String Format" on page 92. The serial port's output can be safely ignored if it is not required in the application.

To read the output string while operating the controller with the R/C radio, you must modify the R/C cable to add an RS232 output wire and connector that will be connected to the PC's communication port. Figure 50 and Figure 50 below shows the wiring diagram of the modified R/C cable for connection to a PC or to a PDA, respectively.

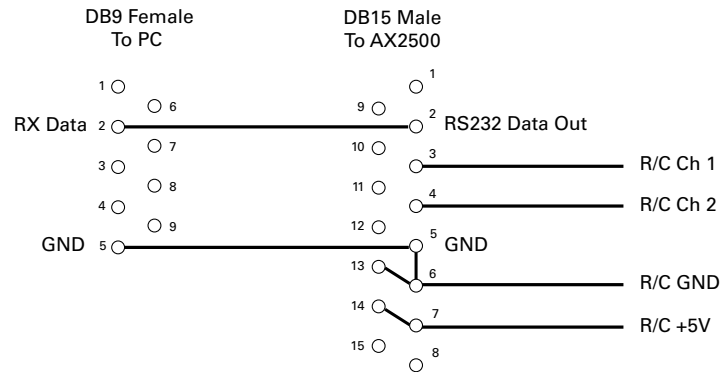


FIGURE 50. Modified R/C cable with RS232 output for data logging to a PC

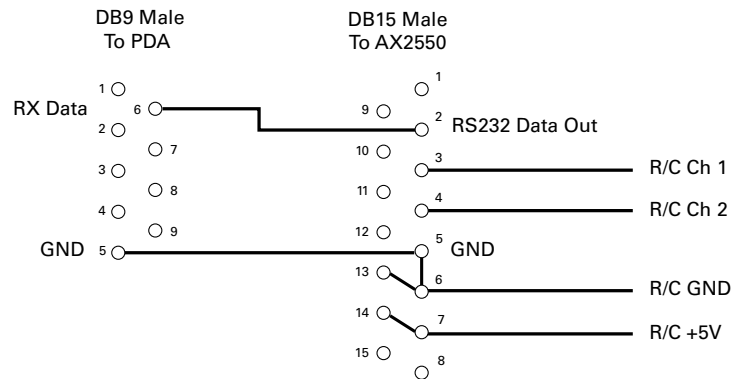


FIGURE 51. Modified R/C cable with RS232 output for data logging to a PDA



## SECTION 8

# Serial (RS-232) Controls and Operation

This section describes the communication settings and the commands accepted by the AX2550 in the RS232 mode of operations. This information is useful if you plan to write your own controlling software on a PC or microcomputer. These commands will also allow you to send commands manually using a terminal emulation program. If you wish to use your PC simply to set parameters and/or to exercise the controller, you should use the Roborun utility described on page 79.

---

## Use and benefits of RS232

The serial port allows the AX2550 to be connected to microcomputers or wireless modems. This connection can be used to both send commands and read various status information in real-time from the controller. The serial mode enables the design of autonomous robots or more sophisticated remote controlled robots than is possible using the R/C mode. RS232 commands are very precise and securely acknowledged by the controller. They are also the method by which the controller's features can be accessed and operated to their fullest extent.

When connecting the controller to a PC, the serial mode makes it easy to perform simple diagnostics and tests, including:

- Sending precise commands to the motors
- Reading the current consumption values and other parameters
- Obtaining the controller's software revision and date
- Reading inputs and activating outputs
- Setting the programmable parameters with a user-friendly graphical interface
- Updating the controller's software

## Connector I/O Pin Assignment (RS232 Mode)

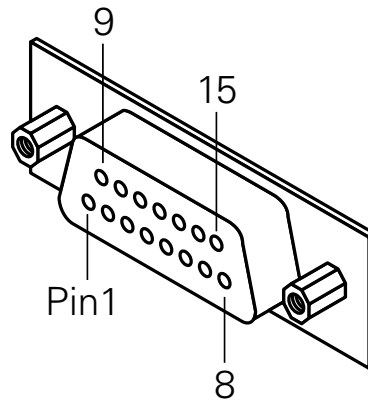


FIGURE 1. Pin locations on the controller's 15-pin connector

When used in the RS232 mode, the pins on the controller's DB15 connector are mapped as described in the table below

TABLE 14. DB15 Connector pin assignment in RS232 mode

Pin Number	Signal	Input or Output	Description
1	Output C	Output	2Amp Accessory Output C (same as pin 9)
2	Data Out	Output	RS232 data output from the controller to the PC
3	Data In	Input	RS232 data input to the controller from the PC
4	Input F	Input	Accessory Input F
5	Ground Out	Power Output	Controller ground (-)
6	Ground In	Power Input	<b>Must be wired to pin 13</b> or pin 5
7	+5V In	Power Input	<b>Must be wired to pin 14</b>
8	Input E	Input	Accessory input E
9	Output C	Output	2Amp Accessory Output C (same as pin 1)
10	Speed/Pos/T 1	Analog in	Channel 1 speed, position or temp feedback
11	Speed/Pos/T 2	Analog in	Channel 2 speed, position or temp feedback
12	Output D	Output	Low Current Accessory Output D
13	Ground Out	Power	Controller ground (-)
14	+5V Out	Power Output	+5V Power Output (100mA max.)
15	Switch Input	Input	Emergency Stop or Invert Switch input



## Cable configuration

The RS232 connection requires the special cabling as described in the figure below. The 9-pin female connector plugs into the PC (or other microcontroller). The 15-pin male connector plugs into the AX2550.

It is critical that you do not confuse the connector's pin numbering. The pin numbers on the drawing are based on viewing the connectors from the front (facing the sockets or pins). Most connectors have pin numbers molded on the plastic.

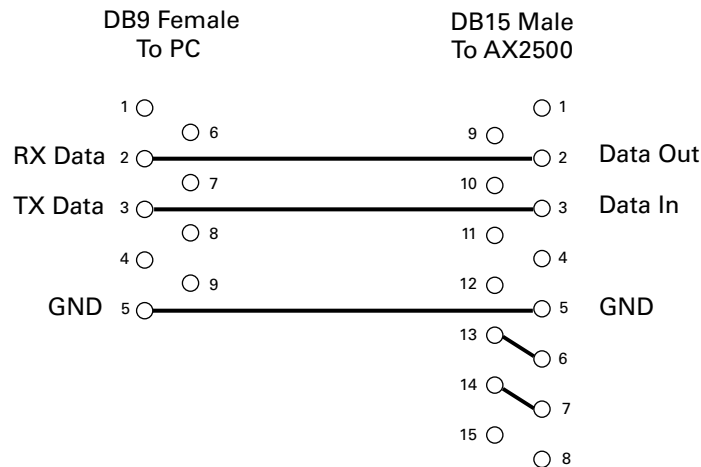


FIGURE 52. PC to AX2550 RS232 cable/connector wiring diagram

## Extending the RS232 Cable

The AX2550 is delivered with a 2 foot cable adapter which may be too short, particularly if you wish to run and monitor the controller inside a moving robot.

RS232 extension cables are available at most computer stores. However, you can easily build one using a 9-pin DB9 male connector, a 9-pin DB9 female connector and any 3-wire cable. These components are available Radio Shack or any electronics distributor. A CAT5 network cable is recommended, and cable length may be up to 100' (30m). Figure 53 shows the wiring diagram of the extension cable.

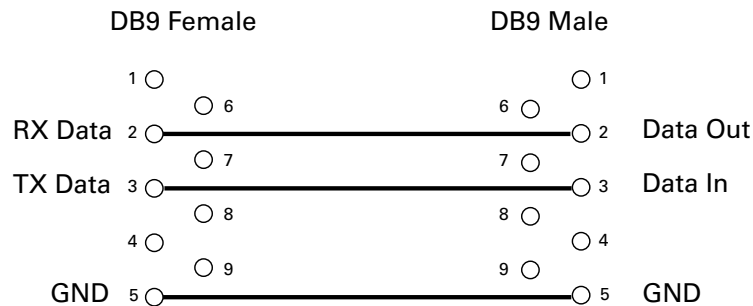


FIGURE 53. RS232 extension cable/connector wiring diagram

## Communication Settings

The AX2550 serial communication port is set as follows:

### 9600 bits/s, 7-bit data, 1 Start bit, 1 Stop bit, Even Parity

Communication is done without flow control, meaning that the controller is always ready to receive data and can send data at any time.

**These settings cannot be changed.** You must therefore adapt the communication setting in your PC or microcomputer to match those of the controller.

## Establishing Manual Communication with a PC

The controller can easily be connected to a PC in order to manually exercise its capabilities. Simply connect the supplied cable to the AX2550 on one end (DB-15 connector) and to a free COM port on the other end (DB-9 connector).

Once connected, you will need a Terminal Emulation program to display the data received from the controller on the PC's screen and to send characters typed on the keyboard to the controller. All Windows PC's come with the Hyperterm terminal emulation software.

Locate the Hyperterm launch icon in the Start button: Programs > Accessories > Communication folder.

You will need to configure Hyperterm to use the COM port to which you have connected the controller (typically COM1) and to configure the communication settings as described in the section above.

To save time and avoid errors, a hyperterm configuration file is automatically installed in your PC's Start button menu when you install Roboteq's Roborun utility (See "Downloading

and Installing the Utility” on page 125). The configuration file is set to use the COM1 port. You can easily change this setting to a different port from the program’s menus.

In all cases, immediately after reset or power up, the controller will output a short identity message followed by a software revision number and software revision date as follows:

**RoboteQ v1.3 08/25/03**  
**G**

The letter below the prompt message is a code that provides information on the hardware and can be ignored.

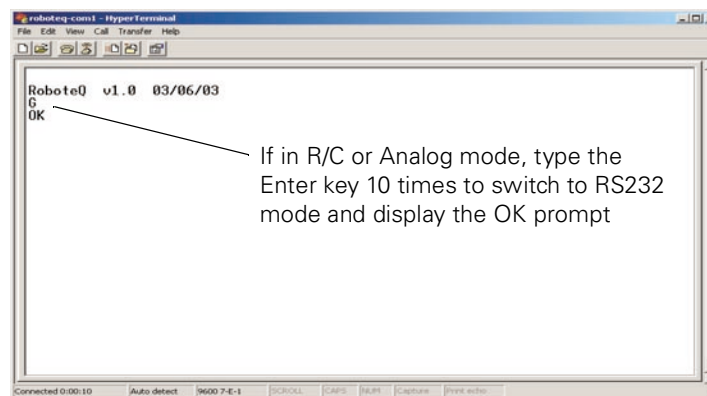


FIGURE 54. Power-on message appearing on Hyperterm

## Entering RS232 from R/C or Analog mode

If the controller is configured in R/C or Analog mode, it will not be able to accept and recognize RS232 commands immediately.

However, the controller will be “listening” to the serial port and will enter the serial mode after it has received 10 continuous “Enter” (Carriage Return) characters. At that point, the controller will output an “OK” message, indicating that it has entered the RS232 mode and that it is now ready to accept commands.

Note that for improved safety, the RS232 watchdog is automatically enabled when entering the RS232 in this way. See “RS-232 Watchdog” on page 90.

When reset again, the controller will revert to the R/C mode or Analog mode, unless the Input Mode parameter has been changed in the meantime.

## Data Logging String in R/C or Analog mode

If the controller is in the R/C or analog mode, immediately after reset it will send a continuous string of characters (one character every 8ms, one entire string every 200ms) containing operating parameters for data logging purposes.

This information can be safely ignored and the controller will still be able to switch to RS232 mode upon receiving 10 continuous Carriage Returns as described above.

The format of the data logging string and its content is described in Figure , "Analog and R/C Modes Data Logging String Format," on page 92

## RS232 Mode if default

If the controller is configured in RS232 mode, it will automatically be in the RS232 mode upon reset or power up.

In this case, the "**OK**" message is sent automatically, indicating that the controller is ready to accept commands through its serial port.

## RS232 Commands Set

AX2550 commands and queries are composed of a series of 2 or 4 characters followed by the "enter" (carriage return) code.

The controller will send back (echo) every character it is receiving. By checking that the returned character is the same as the one sent, it is possible to verify that there has been no error in communication.

After a command has been received and properly executed, the controller will send the "+" character.

If a command has been received with errors or bad parameters, the controller will send the "-" character.

A complete summary of the acknowledge and error messages sent by the controller can be found on page 88.

The following is a list of the AX2550 RS232 commands and queries:

## Set Motor Command Value

Description:

Send a speed or position value from 0 to 127 in the forward or reverse direction for a given channel. In mixed mode, channel 1 value sets the common forward and reverse value for both motors, while channel 2 sets the difference between motor 1 and motor 2 as required for steering. In all other modes, channel 1 commands motor 1 and channel 2 commands motor 2.

Syntax: **!Mnn**

Where **M=**

- A**: channel 1, forward direction
- a**: channel 1, reverse direction
- B**: channel 2, forward direction
- b**: channel 2, reverse direction

Where **nn=** Speed or position value in 2 Hexadecimal digits from 00 to 7F

Examples:

<b>!A00</b>	channel 1 to 0
<b>!B7F</b>	channel 2, 100% forward
<b>!a3F</b>	channel 1, 50% reverse

**Notes:**

The hexadecimal number **must always contain two digits**. For example, !a5 will not be recognized and the controller will respond with a "-" to indicate an error. The proper command in this case should be !a05

Hexadecimal numbers are easy to generate using a microcomputer. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 93.

## Set Accessory Outputs

**Description:**

Turn on or off one of the 2 available digital output lines on the 15-pin connector. See "Using the General Inputs and Outputs" on page 29 for details on how to identify and wire these signals.

Syntax:           **!M**

Where           **M=** **c**: output C off  
                  **C**: output C on  
                  **d**: output D off  
                  **D**: output D on

**Examples:**

**!C**               turn C output on  
**!d**               turn D output off

## Query Power Applied to Motors

**Description:**

This query will cause the controller to return the actual amount of power that is being applied to the motors at that time. The number is a hexadecimal number ranging from 0 to +127 (0 to 7F in Hexadecimal). In most cases, this value is directly related to the command value, except in the conditions described in the notes below.

Syntax:           **?S**

Reply:           **nn**  
                  **mm**

Where:           **nn** = motor 1 applied power value  
                  **mm** = motor 2 applied power value

**Notes:**

The applied power value that is read back from the controller can be different than the command values for any of the following reasons: current limitation is active, motors operate at reduced speed after overheat detection, or mixed mode is currently active.

No forward or reverse direction information is returned by this query.

This query is most useful for providing feedback to a microcontroller commanding the controller. The hexadecimal format is intended to be deciphered by a microcontroller. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 93.

## Query Amps Consumed by Motors

### Description:

This query will cause the controller to return the actual number of Amps being consumed by each motor. The number is an unsigned Hexadecimal number ranging from 0 to 256 (0 to FF in Hexadecimal).

Syntax: **?A**

Reply: ***nn***  
***mm***

Where: ***nn*** = motor 1 Amps  
***mm*** = motor 2 Amps

### Notes:

The Amps measurement has a 10% (worse case) precision. Its main purpose is to provide feedback to the controller's current limitation circuitry.

The hexadecimal format is intended to be deciphered by a microcontroller. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 93.

## Query Analog Inputs

### Description:

This query will cause the controller to return the values of the signals present at its two analog inputs. If the controller is used in close-loop speed mode, the values represent the actual speed measured by the tachometer. When used in position mode, the values represent the actual motor position measured by a potentiometer. In all other modes, the values represent the measured voltage (0 to 5V) applied to the analog inputs. The values are signed Hexadecimal numbers ranging from -127 to +127. The -127 value represents 0V at the input, the 0 value represents 2.5V, and the +127 value represents +5V.

Syntax: **?P**

Reply: ***nn***  
***mm***

Where: ***nn*** = analog input 1 value, speed or position  
***mm*** = analog input 2 value, speed or position

### Notes:

The command returns a signed hexadecimal number where 0 to +127 is represented by 00 to 7F, and -1 to -127 is represented by FF to 80 respectively. The hexadecimal format is intended to be deciphered by a microcontroller. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 93.

## Query Heatsink Temperatures

### Description:

This query will cause the controller to return values based on the temperature measured by internal thermistors located at each heatsink side of the controller. Because NTC thermistors are non-linear devices, the conversion of the read value into a temperature value requires interpolation and a look up table. Figure 26 on page 44 shows this correlation. Sample conversion software code is available from Roboteq upon request. The values are

unsigned Hexadecimal numbers ranging from 0 to 255. The lowest read value represents the highest temperature.

Syntax: **?M**

Reply: **nn**  
**mm**

Where: **nn** = thermistor 1 read value  
**mm** = thermistor 2 read value

Notes:

The hexadecimal format is intended to be deciphered by a microcontroller. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 94.

## Query Battery Voltages

Description:

This query will cause the controller to return values based on two internally measured voltages: the first is the Main Battery voltage present at the thick red and black wires. The second is the internal 12V supply needed for the controller's microcomputer and MOSFET drivers. The values are unsigned Hexadecimal numbers ranging from 0 to 255. To convert these numbers into a voltage figure, use the formulas described in "Internal Voltage Monitoring Sensors" on page 43.

Syntax: **?E**

Reply: **nn**  
**mm**

Where: **nn** = main battery voltage value  
**mm** = internal 12V voltage value

Notes:

The hexadecimal format is intended to be deciphered by a microcontroller. When exercising the controller manually, you may use the Decimal to Hexadecimal conversion table on page 94.

## Query Digital Inputs

Description:

This query will cause the controller to return the state of the controller's two accessory inputs (inputs E and F) and the state of the Emergency Stop/Inverted input. See "Connecting Sensors and Actuators to Input/Outputs" on page 31 for information on how to wire and use these signals. The returned values are three sets of two digits with the values 00 (to indicate a 0 or Off state), or 01 (to indicate a 1 or On state).

Syntax: **?I**

Reply: **nn**  
**mm**  
**oo**

Where: **nn** = Input E status  
**mm** = Input F status  
**oo** = Estop/Invert Switch Input status

Examples:  
**?I** Read Input status query  
**01** Controller replies, Input E is On  
**00** Input F is Off  
**01** Emergency stop switch is high (not triggered)

## Read and Modify Controller Settings

### Description

These commands make it possible to examine and change the controller's parameters stored in Flash. These commands will appear cryptic and difficult to use for manual parameter setting. It is recommended to use the Graphical configuration utility described in "Using the Roborun Configuration Utility" on page 125. Note that many parameters will not take effect until the controller is reset or a special command is sent (see "Apply Parameter Changes" on page 88). The complete list of parameters accessible using these commands is listed in "RS232 Accessible Parameter Table" on page 90.

### Read parameter

Syntax: **^mm**

Reply: **DD**

Where **mm**= parameter number  
**DD**= current parameter value

Example:  
**^00** Read value parameter 0  
**01** Controller replies, value is 01

### Modify parameter

Syntax: **^mm nn**

Reply: **+** if command was executed successfully  
**-** if error

Where **mm**= parameter number  
**nn**= new parameter value

Examples:  
**^02 03** Store 03 into parameter 2

### Notes:

All parameters and values are expressed with 2 hexadecimal digits

No changes will be made and an error will be reported ("-" character) when attempting to read or write a parameter that does not exist or when attempting to store a parameter with an invalid value.



## Apply Parameter Changes

Description:

Many parameters will take effect only after the controller is reset. This command can be used (instead of resetting the controller) to cause these parameters to take effect immediately.

Syntax:           **^FF**

Reply:           **+** Success, changed parameters are now active  
                  **-** if error

## Reset Controller

Description:

This command allows the controller to be reset in the same manner as if the reset button was pressed. This command should be used in exceptional conditions only or after changing the controller's parameters in Flash memory so that they can take effect.

Syntax:           **%rrrrrr**

Reply:           None. Controller will reset and display prompt message

---

## Commands Acknowledge and Error Messages

The AX2550 will output characters in various situations to report acknowledgements or error conditions as listed below.

### Character Echo

At the most fundamental level, the AX2550 will echo back to the PC or Microcontroller every valid character it has received. If no echo is received, one of the following is occurring: the controller is not in the RS232 mode, the controller is Off, or it may be defective.

### Command Acknowledgement

The AX2550 will acknowledge commands in one of two ways:

For commands that cause a reply, such as a speed or amps queries, the reply to the query must be considered as the command acknowledgement.

For commands where no reply is expected, such as speed setting, the controller will issue a "plus" character (+) after every command as an acknowledgment.

### Command Error

If a command or query has been received with errors or is wrong, the control will issue a "minus" character (-) to indicate the error.

If the controller issues the "-" character, it should be assumed that the command was lost and that it should be repeated.

## Overheat On and Off

When the controller detects an overheat condition on its internal temperature sensor, or if the temperature has reached a preset value on one of the (enabled) external sensors, it will send a **"H"** character.

When the offending temperature cools below the overheat threshold, then the controller will issue a **"L"** character and resume normal operation.

## Emergency Stop

If the external emergency switch is pressed or if the Program and Set buttons are pressed simultaneously during normal operation, the controller will stop immediately and issue a **"X"** character.

Note that the emergency stop condition is permanent and can only be exited by powering off the controller, pressing the reset button, or sending the controller reset command via the RS232 port.

## Watchdog time-out

If the RS232 watchdog is enabled, the controller will stop the motors and issue a **"W"** character if it has not received a valid character from the PC or microcontroller within the past 1 seconds.

## Motor Commands during Stop Condition

If a motor command is sent while the controller has stopped due to a overheat or emergency stop condition, the controller will report an error message even though the command may be correct.

In such a case, the controller will issue a code indicating the type of stop condition it is in, followed by a "minus" (-) character,

TABLE 15. Error reported if motor command is received during stop condition

Controller condition	Reported error
Overheat stop	XH
Emergency Stop	XS

## RS-232 Watchdog

For applications demanding the highest operating safety, the controller may be configured to automatically stop the motors (but otherwise remain fully active) if it fails to receive a character on its RS232 port for more than 1 seconds.

The controller will also send a **"W"** character every second to indicate to the microcomputer that such a time-out condition has occurred.

The character does not need to be a specific command, but any valid ASCII code.

The RS232 watchdog is enabled or disabled depending on the value of the "Input Command Mode" parameter.

The RS232 watchdog is automatically enabled when entering the RS232 mode from the R/C mode (see "Entering RS232 from R/C or Analog mode" on page 83)

## RS232 Accessible Parameter Table

Table 16 below lists the complete set of configuration parameters that may be accessed and changed using RS232 commands.

TABLE 16. AX2550 Configuration parameters accessible through RS232

Param nbr	Description	Allowed Values (default)	Active after	See pages
00	Input control mode	(0) = R/C Radio mode (default) 1 = RS232, no watchdog 2 = RS232, with watchdog 3 = Analog mode	Reset	page 63 page 79 page 97
01	Motor Control mode	(0) = Separate A, B, speed control, open loop (default) 1 = Mixed A & B, speed control, open loop 2 = Speed control on A, open loop. Position control on B 3 = A & B Position control 4 = Separate A, B, speed control, closed loop 5 = Mixed A & B, speed control, closed loop 6 = Speed control on A, closed loop. Position control on B	Reset	page 46 page 105
02	Amps limit	0 = 45A 1 = 60A 2 = 75A 3 = 90A (4) = 105A default 5 = 120A	Reset	page 49
03	Acceleration	0 = very slow 1 = slow (2) = medium-slow (default) 3 = medium 4 = fast 5 = fastest	Reset	page 49

TABLE 16. AX2550 Configuration parameters accessible through RS232

Param nbr	Description	Allowed Values (default)	Active after	See pages
04	Input switch func- tion	Bits 0 to 3 contain the EStop/Inv input operating mode 0 = causes emergency stop 1 = invert commands (2) = no action (default) Bits 4,5 and 6,7 contain the Input E and Input F respective configurations (0) = no action 1 = Activate Output C 2 = Turn MOSFETs On or Off	Reset	page 52 page 53  page 54 page 55
05	Brake/Coast	(0) = brake when iddle (default) 1 = coast when idle - not implemented	Reset	
06	Joystick Deadband or Analog Deadband	Values are for Joystick deadband 0 = no deadband 1 = 8% (2) = 16% (default) 3 = 24% 4 = 32% 5 = 40% 6 = 46% 7 = 54%	Reset	page 72 or page 101
07	Exponentiation on channel 1	(0) = Linear (no exponentiation - default) 1 = strong exponential 2 = normal exponential 3 = normal logarithmic 4 = strong logarithmic	Instant	page 73
08	Exponentiation on channel 2	Same as E, above	Instant	
09	External Overheat Detection	0 = disabled 1 to 255 = temperature threshold. See chart on page 42	Reset	page 42
0A	Action upon Exter- nal Heat Detection	0 = immediate stop 1 = 50% power for 30 seconds (default) 2 = 25% power for 2 minutes	Reset	page 54
0B	Left / Right Adjust	(7) = no adjustment (default) 0, 1, ..., 6 = -10.5%, -9%, ..., -1.5% 8, ..., D, E** = +1.5%, ..., +9%, +10.5%	Reset	page 51
0C to 0E	Reserved	Do not use		

TABLE 16. AX2550 Configuration parameters accessible through RS232

<b>Param nbr</b>	<b>Description</b>	<b>Allowed Values (default)</b>	<b>Active after</b>	<b>See pages</b>
0F	Gain Integral for PID	0 to 64 (16) default	Instant	
10	Gain Diff for PID	0 to 64 (16) default	Instant	
11	Gain Prop for PID	0 to 64 (16) default	Instant	
12	Joystick Center 1 MS	00 to FF (06) default	Reset	
13	Joystick Center 1 LS	00 to FF (40) default	Reset	
14	Joystick Center 2 MS	00 to FF (06) default	Reset	
15	Joystick Center 2 LS	00 to FF (40) default	Reset	
16	Joystick Min 1 MS	00 to FF (11) default	Reset	
17	Joystick Min 1 LS	00 to FF (30) default	Reset	
18	Joystick Min 2 MS	00 to FF (11) default	Reset	
19	Joystick Min 2 LS	00 to FF (30) default	Reset	
1A	Joystick Max 1 MS	00 to FF (0C) default	Reset	
1B	Joystick Max 1 LS	00 to FF (80) default	Reset	
1C	Joystick Max 2 MS	00 to FF (0C) default	Reset	
1D	Joystick Max 2 LS	00 to FF (A8) default	Reset	

## Analog and R/C Modes Data Logging String Format

When the controller is configured in R/C or Analog mode, it will automatically and continuously send a string of ASCII characters on the RS232 output.

This feature makes it possible to log the controller's internal parameters while it is used in the actual application. The data may be captured using a PC connected via an RS232 cable or wireless modem, or into a PDA installed in the actual robot. Details on how to wire the DB15 connector is described on page 77 for the R/C mode, and on page 103 for the Analog mode.

This string is composed of a start character delimiter, followed by 12 two-digit Hexadecimal numbers representing 12 internal parameter values, and ending with a Carriage Return character. The figure below shows the structure of this string.

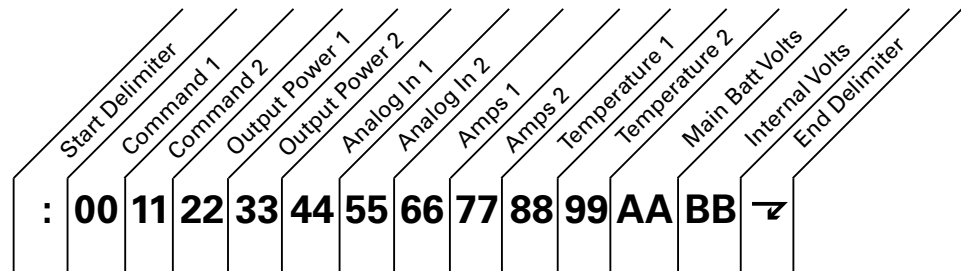


FIGURE 55. ASCII string sent by the controller while in R/C or Analog mode

The hexadecimal values and format for each parameter is the same as the response to RS232 queries described starting on page 85 in this section.

Characters are sent by the controller at the rate of one every 8ms. A complete string is sent in 213ms.

## Decimal to Hexadecimal Conversion Table

The AX2550 uses hexadecimal notation for accepting and responding to numerical commands. Hexadecimal is related to the binary system that is used at the very heart of micro-computers. Functions for converting from decimal to hexadecimal are readily available in high level languages such as C.

If you intend to enter commands manually using the terminal emulation program, you can use the conversion table in Table 17 to do the translation. Note that the table only shows numbers for 0 to 127 decimal (00 to 7F hexadecimal). The AX2550's speed commands are within this range. Table 18 shows the conversion values for numbers between 128 and 255 (unsigned) and between -1 and -128 (signed)

TABLE 17. 0 to +127 signed or unsigned decimal to hexadecimal conversion table

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex
0	00	32	20	64	40	96	60
1	01	33	21	65	41	97	61
2	02	34	22	66	42	98	62
3	03	35	23	67	43	99	63
4	04	36	24	68	44	100	64
5	05	37	25	69	45	101	65
6	06	38	26	70	46	102	66
7	07	39	27	71	47	103	67
8	08	40	28	72	48	104	68
9	09	41	29	73	49	105	69

**TABLE 17. 0 to +127 signed or unsigned decimal to hexadecimal conversion table**

<b>Dec</b>	<b>Hex</b>	<b>Dec</b>	<b>Hex</b>	<b>Dec</b>	<b>Hex</b>	<b>Dec</b>	<b>Hex</b>
10	0A	42	2A	74	4A	106	6A
11	0B	43	2B	75	4B	107	6B
12	0C	44	2C	76	4C	108	6C
13	0D	45	2D	77	4D	109	6D
14	0E	46	2E	78	4E	110	6E
15	0F	47	2F	79	4F	111	6F
16	10	48	30	80	50	112	70
17	11	49	31	81	51	113	71
18	12	50	32	82	52	114	72
19	13	51	33	83	53	115	73
20	14	52	34	84	54	116	74
21	15	53	35	85	55	117	75
22	16	54	36	86	56	118	76
23	17	55	37	87	57	119	77
24	18	56	38	88	58	120	78
25	19	57	39	89	59	121	79
26	1A	58	3A	90	5A	122	7A
27	1B	59	3B	91	5B	123	7B
28	1C	60	3C	92	5C	124	7C
29	1D	61	3D	93	5D	125	7D
30	1E	62	3E	94	5E	126	7E
31	1F	63	3F	95	5F	127	7F

**TABLE 18. +128 to 255 unsigned and -1 to -128 signed decimal to hexadecimal conversion table**

<b>UDec</b>	<b>Dec</b>	<b>Hex</b>	<b>UDec</b>	<b>Dec</b>	<b>Hex</b>	<b>UDec</b>	<b>Dec</b>	<b>Hex</b>	<b>UDec</b>	<b>Dec</b>	<b>Hex</b>
-128	128	80	-96	160	A0	-64	192	C0	-32	224	E0
-127	129	81	-95	161	A1	-63	193	C1	-31	225	E1
-126	130	82	-94	162	A2	-62	194	C2	-30	226	E2
-125	131	83	-93	163	A3	-61	195	C3	-29	227	E3
-124	132	84	-92	164	A4	-60	196	C4	-28	228	E4
-123	133	85	-91	165	A5	-59	197	C5	-27	229	E5
-122	134	86	-90	166	A6	-58	198	C6	-26	230	E6
-121	135	87	-89	167	A7	-57	199	C7	-25	231	E7
-120	136	88	-88	168	A8	-56	200	C8	-24	232	E8
-119	137	89	-87	169	A9	-55	201	C9	-23	233	E9
-118	138	8A	-86	170	AA	-54	202	CA	-22	234	EA
-117	139	8B	-85	171	AB	-53	203	CB	-21	235	EB

TABLE 18. +128 to 255 unsigned and -1 to -128 signed decimal to hexadecimal conversion table

UDec	Dec	Hex	UDec	Dec	Hex	UDec	Dec	Hex	UDec	Dec	Hex
-116	140	8C	-84	172	AC	-52	204	CC	-20	236	EC
-115	141	8D	-83	173	AD	-51	205	CD	-19	237	ED
-114	142	8E	-82	174	AE	-50	206	CE	-18	238	EE
-113	143	8F	-81	175	AF	-49	207	CF	-17	239	EF
-112	144	90	-80	176	B0	-48	208	D0	-16	240	F0
-111	145	91	-79	177	B1	-47	209	D1	-15	241	F1
-110	146	92	-78	178	B2	-46	210	D2	-14	242	F2
-109	147	93	-77	179	B3	-45	211	D3	-13	243	F3
-108	148	94	-76	180	B4	-44	212	D4	-12	244	F4
-107	149	95	-75	181	B5	-43	213	D5	-11	245	F5
-106	150	96	-74	182	B6	-42	214	D6	-10	246	F6
-105	151	97	-73	183	B7	-41	215	D7	-9	247	F7
-104	152	98	-72	184	B8	-40	216	D8	-8	248	F8
-103	153	99	-71	185	B9	-39	217	D9	-7	249	F9
-102	154	9A	-70	186	BA	-38	218	DA	-6	250	FA
-101	155	9B	-69	187	BB	-37	219	DB	-5	251	FB
-100	156	9C	-68	188	BC	-36	220	DC	-4	252	FC
-99	157	9D	-67	189	BD	-35	221	DD	-3	253	FD
-98	158	9E	-66	190	BE	-34	222	DE	-2	254	FE
-97	159	9F	-65	191	BF	-33	223	DF	-1	255	FF



## SECTION 9

# Analog Control and Operation

This section describes how the motors may be operated using analog voltage commands.

---

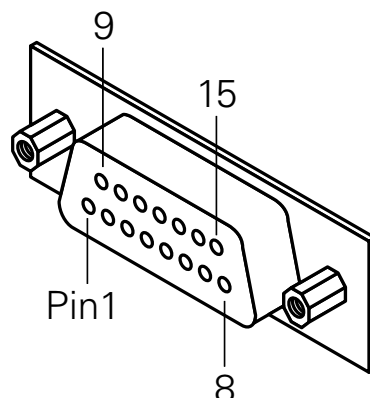
## Mode Description

The AX2550 can be configured to use a 0 to 5V analog voltage, typically produced using a potentiometer, to control each of its two motor channels. The voltage is converted into a digital value of -127 at 0V, 0 at 2.5V and +127 at 5V. This value, in turn, becomes the command input used by the controller. This command input is subject to deadband threshold and exponentiation adjustment. Analog commands can be used to control motors separately (one analog input command for each motor) or in mixed mode.

## Important Notice

**The analog mode can only be used in the open loop speed motor control modes. If the controller is configured in either of the Position or Closed Loop Speed modes, the analog input lines will be used for connection to speed or position sensors and the Analog Input mode will not take effect even if it is enabled.**

## Connector I/O Pin Assignment (Analog Mode)



When used in the Analog mode, the pins on the controller's DB15 connector are mapped as described in the table below

TABLE 19. DB15 Connector pin assignment in Analog mode

Pin Number	Signal	Input or Output	Description
1	Output C	Output	2Amp Accessory Output C (same as pin 9)
2	Data Out	Output	RS232 data output to the PC for data logging
3	Data In	Input	unused
4	Input F	Input	See "Special Use of Accessory Digital Inputs" on page 54
5	Ground Out	Power Output	Controller ground (-)
6	Ground In	Power Input	<b>unused</b>
7	+5V In	Power Input	<b>unused</b>
8	Input E	Input	See "Special Use of Accessory Digital Inputs" on page 54
9	Output C	Output	2Amp Accessory Output C (same as pin 1)
10	Channel 1 In	Analog in	Channel 1 analog input
11	Channel 2 In	Analog in	Channel 2 analog input
12	Output D	Output	Low Current Accessory Output D
13	Ground Out	Power	Controller ground (-)
14	+5V Out	Power Output	+5V Power Output (100mA max.)
15	Switch Input	Input	Emergency Stop or Invert Switch input

## Connecting to a Voltage Source

The analog inputs expect a DC voltage of 0 to 5V which can be sourced by any custom circuitry (potentiometer, Digital to Analog converter).

The controller considers 2.5V to be the zero position (Motor Off). 0V is the maximum reverse command, +5V is the maximum forward command.

The inputs' equivalent circuit is show in Figure 56 below.

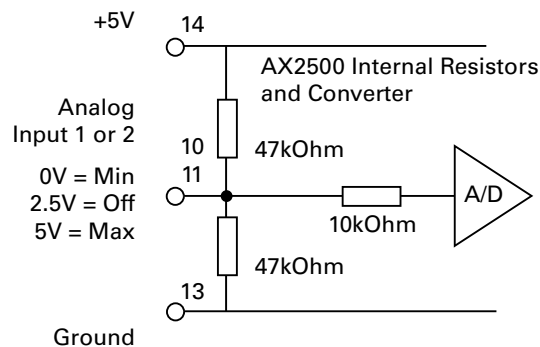


FIGURE 56. Analog input circuit

Notice the two 47K resistors, which are designed to automatically bring the input to a mid-point (Off) position in case the input is not connected. The applied voltage must have sufficient current (low impedance) so that it is not affected by these resistors.

## Connecting a Potentiometer

Figure 57 shows how to wire a potentiometer to the AX2550. By connecting one end to ground and the other to 5V, the potentiometer acts as an adjustable voltage divider. The voltage will thus vary from 0V when the tap is at the minimum position and to 5V when the tap is at the maximum position.

The controller considers 2.5V to be the zero position (Motor Off). 2.5V is the potentiometer's mid point position.

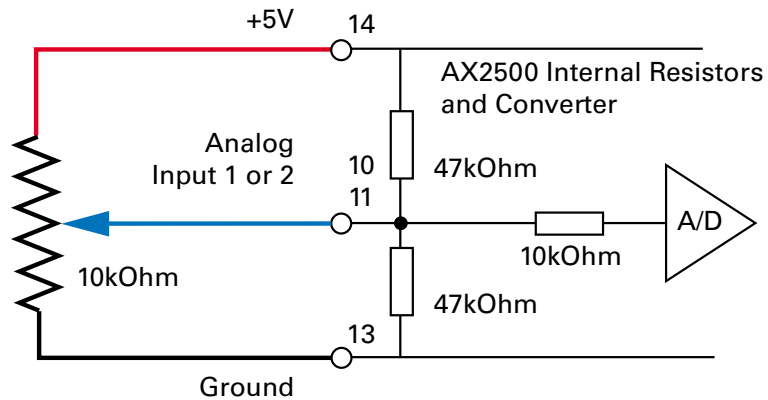


FIGURE 57. Potentiometer connection wiring diagram

The controller includes two 47K ohm resistors pulling the input to a mid-voltage point of 2.5V. When configured in the Analog Input mode, this will cause the motors to be at the Off state if the controller is powered with nothing connected to its analog inputs.

## Selecting the Potentiometer Value

The potentiometer can be of almost any value; however, undesirable effects occur if the value is too low or too high.

If the value is low, an unnecessarily high and potentially damaging current will flow through the potentiometer. The amount of current is computed as the voltage divided by the potentiometer's resistance at its two extremes. For a 1K potentiometer, the current is:

$$I = U/R = 5V / 1000 \text{ Ohms} = 0.005A = 5mA$$

For all practical purposes, a 1K potentiometer is a good minimal value.

If the value of the potentiometer is high, then the two 47K resistors built into the controller will distort the reading. The effect is minimal on a 10K potentiometer but is significant on a 100K or higher potentiometer. Figure 58 shows how the output voltage varies at the various potentiometer positions, for three typical potentiometer values. Note that the effect is an exponentiation that will cause the motors to start moving slowly and accelerate faster as the potentiometer reaches either end.

This curve is actually preferable for most applications. It can be corrected or amplified by changing the controller's exponentiation parameters (see "Command Control Curves" on page 50).

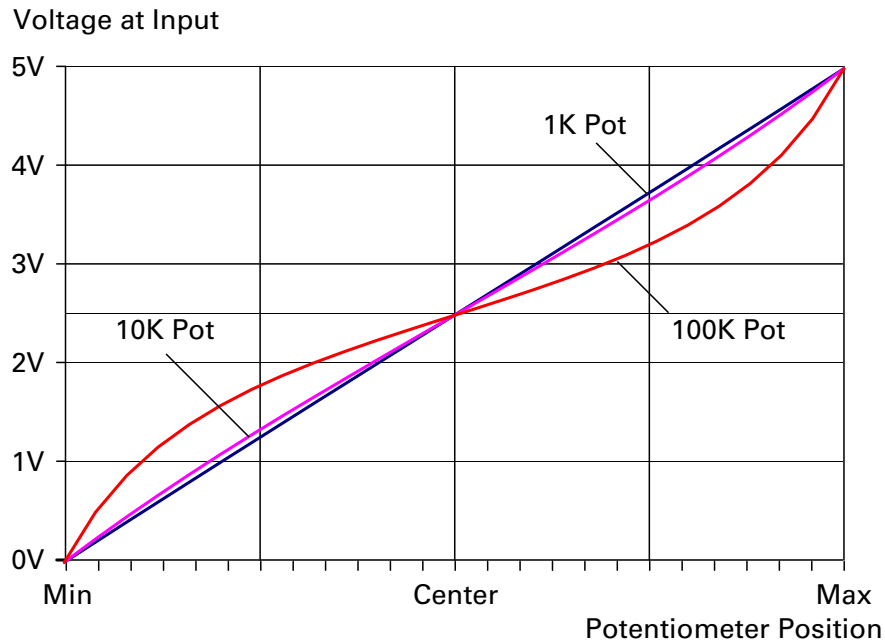


FIGURE 58. Effect of the controller's internal resistors on various potentiometers

## Analog Deadband Adjustment

The controller may be configured so that some amount of potentiometer or joystick travel off its center position is required before the motors activate. The deadband parameter can be one of 8 values, ranging from 0 to 7, which translate into a deadband of 0% to 16%. Even though the deadband will cause some of the potentiometer movement around the center position to be ignored, the controller will scale the remaining potentiometer movement to command the motors from 0 to 100%.

Note that the scaling will also cause the motors to reach 100% at slightly less than 100% of the potentiometer's position. This is to ensure that 100% motor speed is achieved in all circumstances. Table 20 below shows the effect of the different deadband parameter values. Changing the deadband parameter can be done using the controller's switches (see "Configuring the Controller using the Switches" on page 119) or the Roborun utility on a PC (see "Loading, Changing Controller Parameters" on page 128).

TABLE 20. Analog deadband parameters and their effects

Parameter Value	Pot. Position resulting in Motor Power at 0%	Pot. Position resulting in Motor Power at 100%
0	0%	94%
1	0% through 2.4%	96%
2	0% through 4.7%	93%

TABLE 20. Analog deadband parameters and their effects

Parameter Value	Pot. Position resulting in Motor Power at 0%	Pot. Position resulting in Motor Power at 100%
3 (default)	0% though 7.1%	95%
4	0% though 9.4%	93%
5	0% though 11.8%	95%
6	0% though 14.2%	94%
7	0% through 16.5%	96%

## **Important Notice**

**Some analog joysticks do not cause the potentiometer to reach either extremes. This may cause the analog voltage range to be above 0V and below 5V when the stick is moved to the extremes, and therefore the controller will not be able to deliver full forward or reverse power.**

## **Power-On Safety**

When powering on the controller, power will not be applied to the motors until both the Channel 1 and Channel 2 potentiometers have been centered to their middle position. This is to prevent the robot or vehicle from moving in case the joystick was in an active position at the moment the controller was turned on.

## **Data Logging in Analog Mode**

While in Analog Mode, the AX2550 will continuously send a string of characters on the RS232 output line. This string will contain 12 two-digits hexadecimal number representing the 12 following operating parameters.

- Captured Analog Command 1 and 2
- Power Applied to Controller's output stage
- Raw analog command values
- Amps on channel 1 and 2
- Internal Heat Sink temperatures 1 and 2
- Main Battery voltage
- Internal 12V voltage

The entire string is repeated every 213 milliseconds with the latest internal parameter values. This information can be logged using the Roborun Utility (see "Viewing and Logging Data in Analog and R/C Modes" on page 137). It may also be stored in a PDA that can be placed in the mobile robot.

The string and data format is described in "Analog and R/C Modes Data Logging String Format" on page 92. The serial port's output can be safely ignored if it is not required in the application.

To read the output string while operating the controller with an analog command, you must modify the cable to add an RS232 output wire and connector that will be connected to the PC's communication port. Figure 59 below shows the wiring diagram of the modified cable for connection to a PC or to a PDA, respectively.

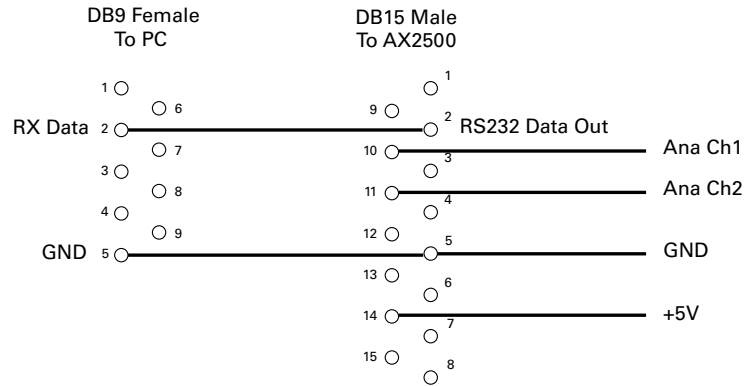


FIGURE 59. Modified Analog cable with RS232 output data logging for PC

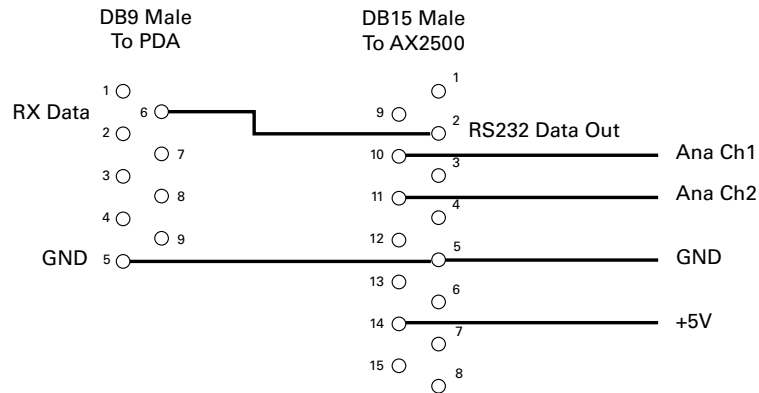


FIGURE 60. Modified Analog cable with RS232 output data logging for PDA





## SECTION 10

# Position Mode

This section describes the AX2550 Position mode, how to wire the motor and position sensor assembly, and how to tune and operate the controller in this mode.

---

## Mode Description

In this mode, the axle of a geared down motor is coupled to a potentiometer that is used to compare the angular position of the axle versus a desired position. The controller will move the motor so that it reaches this position.

This unique feature makes it possible to build ultra-high torque “jumbo servos” that can be used to drive steering columns, robotic arms, life-size models and other heavy loads.

The AX2550 incorporates a full-featured Proportional, Integral, Differential (PID) control algorithm for quick and stable positioning.

---

## Selecting the Position Mode

The position mode is selected by changing the Motor Control parameter in the controller to either

- A Open Loop Speed, B Position
- A Closed Loop Speed, B Position
- A and B Position

Note that in the first two modes, only the second motor will operate in the Position mode.

Changing the parameter is best done using the Roborun Utility. See “Loading, Changing Controller Parameters” on page 128.

For safety reasons and to prevent this mode from being accidentally selected, Position modes **CAN NOT** be selected by configuring the controller using the built-in switches and display.

---

## Potentiometer Mounting

Proper mounting of the potentiometer is critical for an effective and accurate position mode operation. Figure 61 shows a typical motor, gear box, and potentiometer assembly.

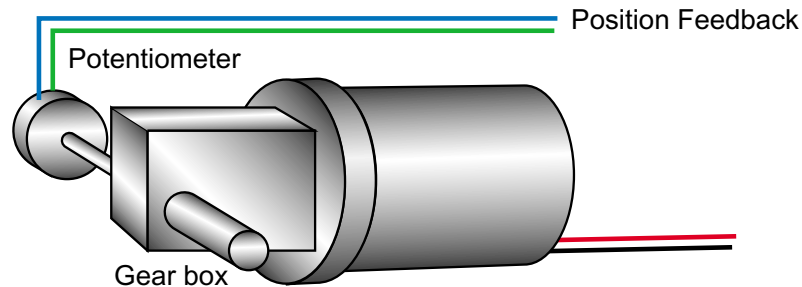


FIGURE 61. Typical motor/potentiometer assembly in Position Mode

---

The potentiometer is composed of two parts:

- a body which must be physically attached to a non-moving part of the motor assembly, or the robot chassis, and
- an axle which must be physically connected to the rotating part of the motor you wish to position.

A gear box is necessary in order to greatly increase the torque of the assembly. It is also necessary in order to slow down the motion so that the controller has the time to perform the position control algorithm. If the gearing ratio is too high, however, the positioning mode will be very sluggish.

A good ratio should be such that the output shaft rotates at 1 to 10 rotations per second (60 to 600 RPM) when the motor is at full speed.

The mechanical coupling between the motor and the potentiometer must be as tight as possible. If the gear box is loose, the positioning will not be accurate and will be unstable, potentially causing the motor to oscillate.

Potentiometers have a limited rotation range of typically 270 degrees (3/4 of a turn), which will in turn limit the mechanical motion of the motor/potentiometer assembly.

---

## Potentiometer wiring

The potentiometer must be wired so that it creates a voltage that is proportional to its angular position: 0V at one extreme, +5V at the other. A 10K potentiometer value is recommended for this use.

Connecting the potentiometer to the controller is as simple as shown in the diagram on Figure 62.

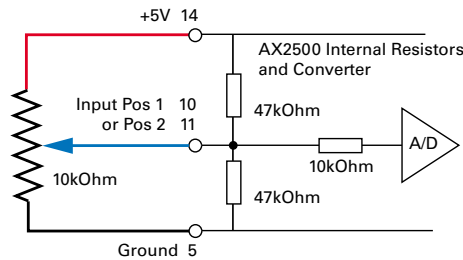


FIGURE 62. Potentiometer wiring in Position mode

## Potentiometer and Motor Polarity

The potentiometer polarity (i.e. which rotation end produces 0 or 5V) is related to the motor's polarity (i.e. which direction the motor turns when power is applied to it).

In the Position mode, the controller compares the actual position, as measured by the potentiometer, to the desired position. If the motor is not at that position, the controller will apply power to the motor so that it turns towards that destination, until reached.

## Important Warning:

**If there is a polarity mismatch, the motor will turn in the wrong direction and the position will never be reached. The motor will turn continuously with no way of stopping it other than cutting the power or hitting the Emergency Stop buttons.**

Determining the right polarity is best done experimentally using the Roborun utility (see "Using the Roborun Configuration Utility" on page 125) and following these steps:

1. Disconnect the controller's Motor Power (thick power wires).
2. Configure the controller in Position Mode using the PC utility.
3. Loosen the potentiometer's axle from the motor assembly.
4. Launch the Roborun utility and click on the Run tab. Click the "Start" button to begin communication with the controller. The potentiometer values will be displayed in the Ana1 and Ana2 boxes.
5. Move the potentiometer manually to the middle position until you measure "0" using Roborun utility
6. Verify that the motor sliders are in the "0" (Stop) position. Since the desired position is 0 and the measured position is 0, the controller will not attempt to move the motors, and the Power graph on the PC must be 0.
7. Apply power to the Motor Power wires (thick wires). The motor will be stopped.

8. With a hand ready to disconnect the Motor Power cable, or ready to press the “Program” and “Set” buttons at the same time (Emergency Stop), SLOWLY move the potentiometer off the center position and observe the motor’s direction of rotation.
9. If the motor turns in the direction you moved the potentiometer, the polarity is correct. You may tighten the potentiometer axle to the motor assembly.
10. If the motor turns in the direction away from the potentiometer, then the polarity is reversed. You should either exchange the wire polarity on the motors or swap the 0 and +5V wires on the potentiometer.
11. Move the potentiometer back to the center point to stop the motor. Cut the power if you feel you are losing control.
12. If the polarity was wrong, invert it and repeat steps 8 to 11.
13. Tighten the potentiometer.

---

## Adding Safety Limit Switches

The Position mode depends on the potentiometer providing accurate position information. If the potentiometer is damaged or one of its wire is cut, the motors may spin continuously in an attempt to reach a fictitious position. In many applications, this may lead to serious mechanical damage.

To limit the risk of such breakage, it is recommended to add limit switches that will cause the motors to stop if unsafe positions have been reached, independently of the potentiometer reading.

Two simple and low cost methods are proposed to implement safety limit switches:

The first, shown in Figure 63, uses Normally Closed limit switches in series on each of the motor terminals. As the motor reaches one of the switches, the lever is pressed, cutting the power to the motor. The diode in parallel with the switch allows the current to flow in the reverse position so that the motor may be restarted and moved away from that limit.

The diode polarity depends on your particular wiring and motor orientation. If the diode is mounted backwards, the motor will not stop once the limit switch lever is pressed. If this is the case, reverse the diode polarity.

The diodes may be eliminated, but then it will not be possible for the controller to move the motor once either of the limit switches has been triggered.

The main benefit of this technique is its total independence on the controller’s electronics and its ability to work in practically all circumstances. Its main limitation is that the switch and diode must be capable of handling the current that flows through the motor.

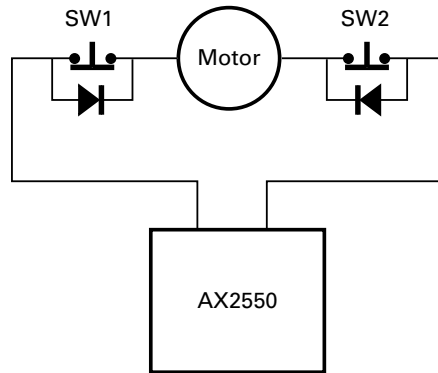


FIGURE 63. Safety limit switches interrupting power to motors

The second method uses the AX2550's Emergency Stop input to shut down the controller if any of the limit switches is tripped. Figure 64 shows the wiring diagram used in this case. Each of the limit switches is a Normally Open switch. Two of these switches are typically required for each motor. Additional switches may be added as needed for the second motor and/or for a manual Emergency Stop. Since very low current flows through the switches, these can be small, low cost switches.

The principal restriction of this technique is that it depends on the controller to be fully functioning, and that once a switch is activated, the controller will remain inactive until the switch is released. In most situations, this will require manual intervention. Another limitation is that both channels will be disabled even if only one channel caused the fault.

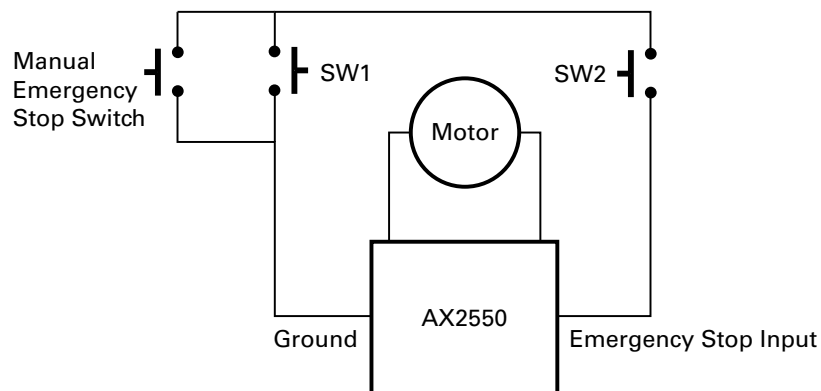


FIGURE 64. Safety limit using AX2550's Emergency Stop input

## **Important Warning**

**Limit switches must be used when operating the controller in Position Mode. This will significantly reduce the risk of mechanical damage and/or injury in case of damage to the position sensor or sensor wiring.**

## Control Loop Description

The AX2550 performs the Position mode using a full featured Proportional and Differential (PD) algorithm. This technique has a long history of usage in control systems and works on performing adjustments to the Power Output based on the difference measured between the desired position (set by the user) and the actual position (captured by the potentiometer).

Figure 65 shows a representation of the PD algorithm. Every 16 milliseconds, the controller measures the actual motor position and subtracts it from the desired position to compute the position error.

The resulting error value is then multiplied by a user selectable Proportional Gain. The resulting value becomes one of the components used to command the motor. The effect of this part of the algorithm is to apply power to the motor that is proportional with the distance between the current and desired positions: when far apart, high power is applied, with the power being gradually reduced and stopped as the motor moves to the final position.

A higher Proportional Gain will cause the algorithm to apply a higher level of power for a given measured error thus making the motor move quicker. However, because of inertia, a faster moving motor will have more difficulty stopping when it reaches its desired position. It will therefore overshoot and possibly oscillate around that end position.

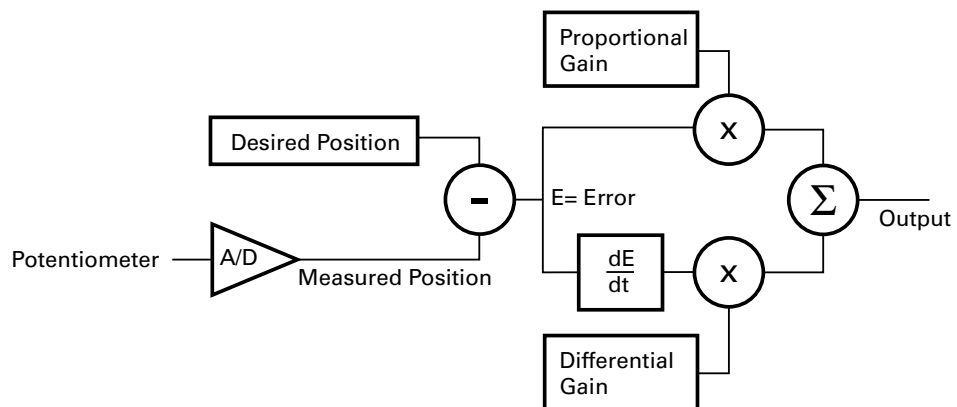


FIGURE 65. PD algorithm used in Position mode

The differential component of the algorithm computes the changes to the error from one 16 ms time period to the next. This change will be a relatively large number every time an abrupt change occurs on the desired position value or the measured position value. The value of that change is then multiplied by a user selectable Differential Gain and added to

the output. The effect of this part of the algorithm is to give a boost of extra power when starting the motor due to changes to the desired position value. The differential component will also greatly help dampen any overshoot and oscillation.

---

## PD tuning

As discussed above, two parameters - Proportional Gain and Differential Gain - can be adjusted to tune the position control algorithm. The ultimate goal in a well tuned PD is a motor that reaches the desired position quickly without overshoot or oscillation.

Because many mechanical parameters, such as motor power, gear ratio, load and inertial are difficult to model, tuning the PD is essentially a manual process that takes experimentation.

The Roborun PC utility makes this experimentation easy by providing a screen for changing the Proportional and Differential gains and another screen for running and monitoring the motors.

When tuning the motor, first start increasing the Proportional Gain until the motor overshoots and oscillates. Then add Differential gain until there is no more overshoot. If the overshoot persists, reduce the Proportional Gain. You may further fine tune the PD by varying the gains from these positions.

In case where the load moved by the motor is not fixed, you will need to tune the PD with the minimum expected load and tune it again with the maximum expected load. Then try to find values that will work in both conditions. If the disparity between minimal and maximal possible loads is large, it may not be possible to find satisfactory tuning values.

Note that the AX2550 uses one set of Proportional and Differential Gains for both motors and therefore assumes that similar motors, mechanical assemblies and loads are present at each channel.





## SECTION 11

# Closed Loop Speed Mode

This section discusses the AX2550 Close Loop Speed mode.

---

## Mode Description

In this mode, an analog tachometer measures the actual motor speed and compares it to the desired speed. If the speed changes because of changes in load, the controller automatically compensates the power output. This mode is preferred in precision motor control and autonomous robotic applications.

The AX2550 incorporates a full-featured Proportional, Integral, Differential (PID) control algorithm for quick and stable speed control.

---

## Selecting the Position Mode

The position mode is selected by changing the Motor Control parameter in the controller to either

- A and B Closed Loop Speed Separate
- A and B Closed Loop Speed Mixed
- A Closed Loop Speed, B Position

Note that in the last selection, only the first motor will operate in the Closed Loop Speed mode.

Changing the parameter to select this mode is done using the Roborun Utility. See “Loading, Changing Controller Parameters” on page 128.

For safety reasons and to prevent this mode from being accidentally selected, Closed Loop Speed modes **CAN NOT** be selected by configuring the controller using the built-in switches and display.

## Tachometer Mounting

Proper mounting of the tachometer is critical for an effective and accurate position mode operation. Figure 66 shows a typical motor and tachometer assembly.

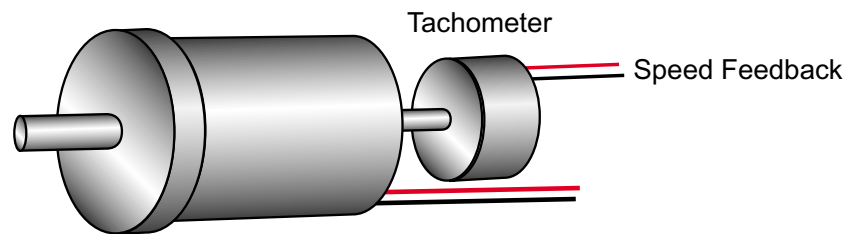


FIGURE 66. Motor + tachometer assembly needed for Close Loop Speed mode

## Tachometer wiring

The tachometer must be wired so that it creates a voltage at the controller's analog input that is proportional to rotation speed: 0V at full reverse, +5V at full forward, and 0 when stopped.

Connecting the tachometer to the controller is as simple as shown in the diagram on Figure 67. See also “Connecting Tachometer to Analog Inputs” on page 40.

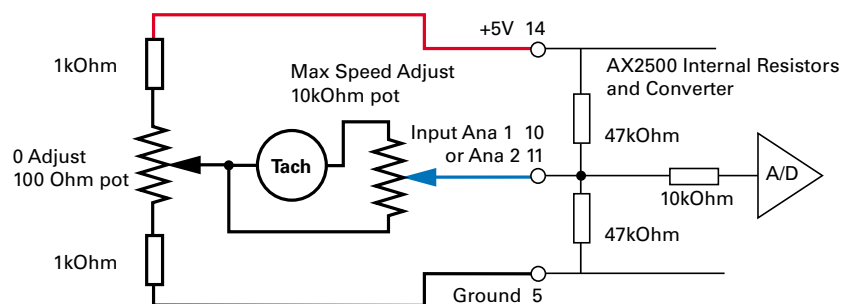


FIGURE 67. Tachometer wiring diagram

## Tachometer and Motor Polarity

The tachometer polarity (i.e. which rotation direction produces a positive or negative voltage, is related to the motor's rotation speed and direction the motor turns when power is applied to it).

In the Closed Loop Speed mode, the controller compares the actual speed, as measured by the tachometer, to the desired speed. If the motor is not at the desired speed and direction, the controller will apply power to the motor so that it turns faster or slower, until reached.

## **Important Warning:**

**If there is a polarity mismatch, the motor will turn in the wrong direction and the speed will never be reached. The motor will turn continuously with no way of stopping it other than cutting the power or hitting the Emergency Stop buttons.**

Determining the right polarity is best done experimentally using the Roborun utility (see "Using the Roborun Configuration Utility" on page 125) and following these steps:

1. Disconnect the controller's Motor Power (thick power wires).
2. Configure the controller in Open Loop Mode using the PC utility. This will cause the motors to run in Open Loop for now
3. Launch the Roborun utility and click on the Run tab. Click the "Start" button to begin communication with the controller. The tachometer values will be displayed in the appropriate Analog input value box(es) which will be labeled Ana 1 and Ana 2.
4. Verify that the motor sliders are in the "0" (Stop) position.
5. Verify that the Analog value read is 0 when the motors are stopped. If not, trim the "0" offset potentiometer.
6. Apply power to the Motor Power wires (thick wires). The motor will be stopped.
7. Move the cursor of the desired motor to the right so that the motor starts rotating, and verify that a positive voltage is read on the tachometer. Move the cursor to the left and verify that a negative voltage is read.
8. If the tachometer polarity is the same of the applied command, then the wiring is correct.
9. If the tachometer polarity is opposite to the command polarity, then reverse the tachometer wires.
10. Proceed to calibrate the Max Closed Loop speed.
11. Set the controller parameter to the desired Closed Loop Speed mode using the Roborun utility

---

## **Adjust Offset and Max Speed**

For proper operation, the controller must see a 0 analog speed value (2.5V voltage on the analog input).

To adjust the 0 value when the motors are stopped, use the Roborun utility to view the analog input value while the tachometer is not turning. Move the 0 offset potentiometer until a stable 0 is read. This should be right around the potentiometer's middle position.

The tachometer must also be calibrated so that it reports a +127 or -127 analog speed value (5V or 0V on the analog input, respectively) when the motors are running at the maximum desired speed in either direction. Since most tachometers will generate more than +/- 2.5V, a 10kOhm potentiometer must be used to scale it's output.

To set the potentiometer, use the Roborun utility to run the motors at the desired maximum speed, while in Open Loop mode (no speed feedback). While the tachometer is spinning, adjust the potentiometer until the analog speed value read is reaching 126.

Note: The maximum desired speed should be lower than the maximum speed that the motors can spin at maximum power and no load. This will ensure that the controller will be able to eventually reach the desired speed under most load conditions.

## **Important Warning:**

**It is critically important that the tachometer and its wiring be extremely robust. If the tachometer reports an erroneous voltage, or no voltage at all, the controller will consider that the motors has not reached the desired speed value and will gradually increase the applied power to the motor to 100% with no way of stopping it until power is cut off or Emergency Stop is activated.**

## **Control Loop Description**

The AX2550 performs the Closed Loop Speed mode using a full featured Proportional, Integral and Differential (PID) algorithm. This technique has a long history of usage in control systems and works on performing adjustments to the Power Output based on the difference measured between the desired speed (set by the user) and the actual position (captured by the tachometer).

Figure 68 shows a representation of the PID algorithm. Every 16 milliseconds, the controller measures the actual motor speed and subtracts it from the desired position to compute the speed error.

The resulting error value is then multiplied by a user selectable Proportional Gain. The resulting value becomes one of the components used to command the motor. The effect of this part of the algorithm is to apply power to the motor that is proportional with the difference between the current and desired speed: when far apart, high power is applied, with the power being gradually reduced as the motor moves to the desired speed.

A higher Proportional Gain will cause the algorithm to apply a higher level of power for a given measured error thus making the motor react more quickly to changes in commands and/or motor load.

The Differential component of the algorithm computes the changes to the error from one 16 ms time period to the next. This change will be a relatively large number every time an abrupt change occurs on the desired speed value or the measured speed value. The value of that change is then multiplied by a user selectable Differential Gain and added to the output. The effect of this part of the algorithm is to give a boost of extra power when starting the motor due to changes to the desired speed value. The differential component will also greatly help dampen any overshoot and oscillation.

The Integral component of the algorithm perform a sum of the error over time. This component helps the controller reach and maintain the exact desired speed when the error is reaching zero (i.e. measured speed is near to, or at the desired value)

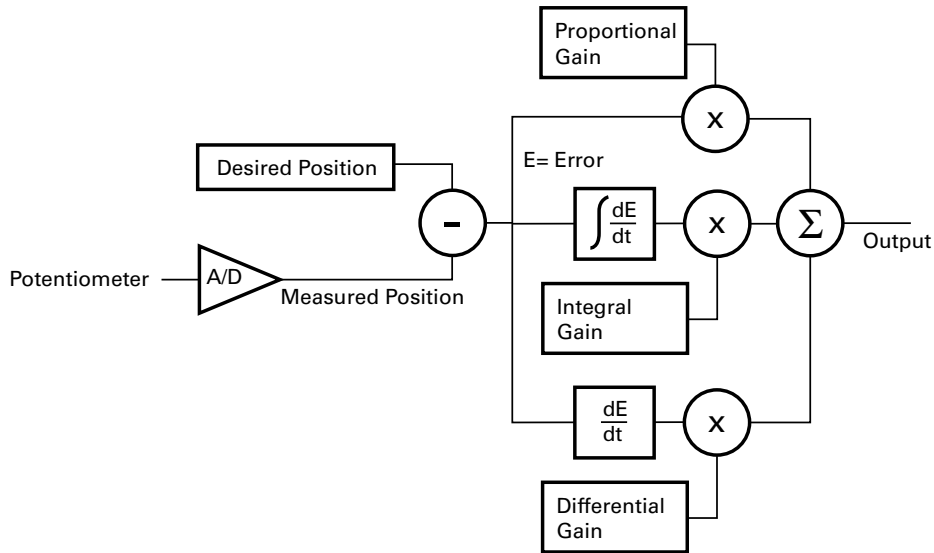


FIGURE 68. PID algorithm used in Speed mode

## PID tuning

As discussed above, three parameters - Proportional Gain, Integral Gain, and Differential Gain - can be adjusted to tune the Closed Loop Speed control algorithm. The ultimate goal in a well tuned PID is a motor that reaches the desired speed quickly without overshoot or oscillation.

Because many mechanical parameters, such as motor power, gear ratio, load and inertia are difficult to model, tuning the PID is essentially a manual process that takes experimentation.

The Roborun PC utility makes this experimentation easy by providing a screen for changing the Proportional, Integral and Differential gains and another screen for running and monitoring the motors. Run first the motor with the preset values. Then experiment different values until a satisfactory behavior is found.

In case where the load moved by the motor is not fixed, you will need to tune the PID with the minimum expected load and tune it again with the maximum expected load. Then try to find values that will work in both conditions. If the disparity between minimal and maximal possible loads is large, it may not be possible to find satisfactory tuning values.

Note that the AX2550 uses one set of Proportional, Integral and Differential Gains for both motors and therefore assumes that similar motors, mechanical assemblies and loads are present at each channel.



# Configuring the Controller using the Switches

---

The AX2550 Speed Controller can be programmed to operate in many modes using a simple set-up procedure. Two buttons and a 7-segment LED display let the user examine and change these settings. Alternatively, the controller may be programmed using a PC connected to the AX2550 controller through the RS232 serial communication port.

The new settings are then permanently stored in the controller's Flash memory so that jumpers are not required, resulting in improved system reliability.

---

## Programming Methods

There are three methods for programming the AX2550's settings:

- Using the controller's built-in switches and display. This method is described in details in this chapter.
- Using the PC-based Configuration Utility. See "Using the Roborun Configuration Utility" on page 125.
- Sending RS232 commands manually. See "RS232 Accessible Parameter Table" on page 90.

---

## Programming using built-in Switches and Display

Two switches and an LED display are provided to let you easily configure the controller in any of its many operating modes. Unlike the RS232 programming mode, the switches will let you configure the controller without the need for external hardware or special connectors. In this mode, the controller may be configured while installed on the robot without the need for special tools or a PC. Figure 69 shows the placement of the switches and display.

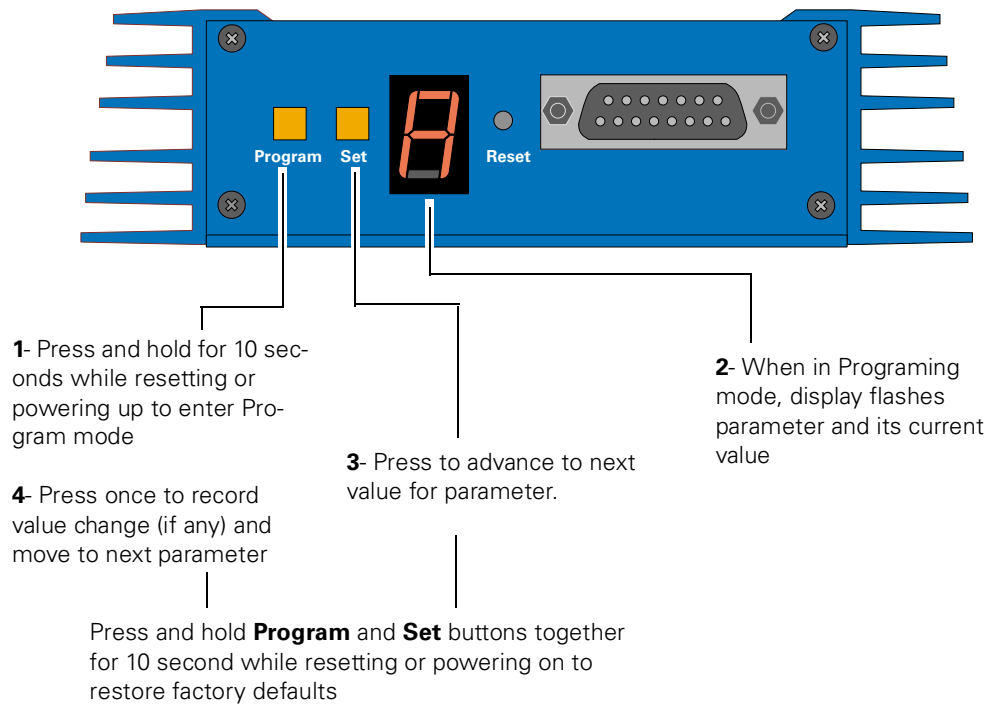


FIGURE 69. Operating the controller's buttons and display

## Entering Programming Mode

Programming mode is entered by pressing and holding the **Program** button for 10 seconds after resetting the controller. The controller can be reset by powering it down and up or by pressing the **Reset** switch inserting a paper clip in the hole.

While the button is pressed and until the controller enters the programming mode, the display will show the following steady pattern.



After 10 seconds, the controller will enter the programming mode and flash a letter representing the first parameter in the list, followed by its numerical value.

**Be careful** not to confuse the **Set** and **Program** button when entering the Programming Mode. Pressing and holding the **Set** button alone for 10 seconds after reset will cause the



controller to enter in self-test mode. This will cause the motors to be turned On and Off according the test sequence described in the Self Test section of this manual. See "Self-Test Mode" on page 55.

## Changing parameters

Pressing the Set button while in Programming Mode will cause the value of the parameter being inspected to be incremented by one. When the maximum valid value is reached, pressing Set again will cause the value to restart at 0.

When the desired value is displayed, press the **Program** button to store it in the controller's non-volatile memory. This will also cause the controller to display the next parameter and its current value.

**Note** that a new parameter value will **ONLY** be saved if the **Program** button is pressed after the value has been changed with the **Set** button.

Additionally, once you have pressed the **Set** button and begin changing a parameter's value, it is not possible to cancel the change. If you wish to leave a parameter value unchanged after you have started changing it, you must press the **Set** button again several times until it goes back to its original value. Alternatively, you can reset or power Off the controller to prevent the new value to be recorded.

## The Special Case of Joystick Calibration

When the Joystick parameter is selected ("J" and "-" flashing), pressing the **Set** key again will cause the controller to enter the Radio Control Joystick calibration sequence.

Once the joystick calibration mode is entered, the display will flash "J" and "o". With the R/C receiver and transmitter On, move both joysticks to their full forward, full back, full left and full right positions a few times. This will cause the controller to capture the min. and max. joystick position values.

Then move the joysticks to their central (rest) positions. Press the **Program** button to save these values in the controller's Flash memory. The min. and max. values saved are these captured when the joystick was moved around. The center values saved are the position of the joystick as it was when **Program** was pressed.

**Note** that once the Joystick calibration mode is entered, you must go through the complete calibration sequence described above. If the joysticks are not moved, incoherent data may be saved in the Flash causing the controller to malfunction in the R/C mode. If bad calibration data is saved in the Flash, try calibrating again by repeating the entire Joystick Programming sequence, otherwise you may restore the factory defaults (this will cause all parameters you have changed to be restored to their default values as well)

The Joystick calibration operation is fully described in the R/C mode chapter. See "Left / Right Tuning Adjustment" on page 51.

## Restoring factory defaults

Should you, for any reason, require to reset the AX2550 controller to its factory defaults value, press and hold the Program and Set button together for 10 seconds while resetting the controller. All parameters, including Joystick limits will be reset to their default values shown in the "Programmable Parameters List" on page 122

## Exiting the Parameter Setting Mode

Exiting the Programming Mode can be done only by pressing the **Reset** button or powering down the controller. The new parameters will be the ones in use after the controller is reset or first powered up.

## Programmable Parameters List

The following table shows the AX2550's controller parameters in the order they appear during programming, as well as their valid values.

## Important Notice

**The parameter table below is guaranteed to be accurate only if your controller software version number matches the one of this manual. See "Obtaining the Controller's Software Revision Number" on page 16 for instructions on how to find this number.**

**This manual is for software version 1.3**

If your controller has a more current software revision, please download an updated version of this manual from the Roboteq web site at [www.roboteq.com](http://www.roboteq.com).

**This table shows only the parameters that can be programmed using the switches and display.** Other less commonly used parameters exist and are only accessible and programmable using the Configuration Utility (page 125) or the RS232 commands (page 90).

TABLE 21. Parameters accessible using the controller's switches and display

Order	Letter	Description	Possible Values (default)	See pages
1	I	Input Command mode:	(0) = R/C Radio mode (default) 1 = RS232 full duplex, no watchdog 2 = RS232 half duplex, with watchdog 3 = Analog mode	page 63 page 79 page 97
2	C	Motor Control mode	(0) = Separate A, B, speed control, open loop (default) 1 = Mixed A & B, speed control, open loop <b>For safety reasons, the modes below cannot be selected using the switches.</b> 2 = Speed control on A, open loop. Position control on B 3 = A & B Position control 4 = Separate A, B, speed control, closed loop 5 = Mixed A & B, speed control, closed loop 6 = Speed control on A, closed loop. Position control on B	page 46

TABLE 21. Parameters accessible using the controller's switches and display

<b>Order</b>	<b>Letter</b>	<b>Description</b>	<b>Possible Values (default)</b>	<b>See pages</b>
3	A	Amp limit	0 = 45A 1 = 60A 2 = 75A 3 = 90A (4) = 105A default 5 = 120A	page 49
4	S	Acceleration	0 = very slow 1 = slow (2) = medium (default) 3 = medium 4 = fast 5 = fastest	page 49
5	U	Input switch function	0 = causes emergency stop 1 = invert commands switch (2) = no action (default)	page 52 page 53
6	b	Brake/Coast	(0*) = brake when idle* (default) 1 = coast when idle - not implemented	
7	d	R/C Joystick Dead-band Analog Input Dead-band	0 = no deadband *** 1 = 8% (2) = 16% (default) 3 = 24% 4 = 32% 5 = 40% 6 = 46% 7 = 54%	page 72 page 101
8	E	Exponentiation on channel 1	(0) = Linear (no exponentiation - default) 1 = strong exponential 2 = normal exponential 3 = normal logarithmic 4 = strong logarithmic	page 73
9	F	Exponentiation on channel 2	Same as E, above	
10	H	Heat Detection on external sensor	(0) = immediate stop (default) 1 = 50% power for 2 minutes 2 = 25% power for 10 minutes	page 54

TABLE 21. Parameters accessible using the controller's switches and display

Order	Letter	Description	Possible Values (default)	See pages
11	L	Left / Right Adjust	(7) = no adjustment (default) 0, 1,..., 6 = -10.5%, -9%,...,-1.5% 8,..., D, E** = +1.5%,..., +9%, +10.5%	page 51
12	J	Joystick calibration	- = not calibrating o = in calibration mode	page 73

\*The coast function is not implemented in this revision of the software. The controller will only operate in brake mode.

\*\* Values are in hexadecimal numbers where the decimal values 10, 11, 12... 15 are represented with the letters A, B, C... F.

\*\*\* Deadband percent values shown are for R/C mode. For analog deadband values, see page 101.

# Using the Roborun Configuration Utility

---

A PC based Configuration Utility is available, free of charge, from Roboteq. This program makes configuring and operating the AX2550 much more intuitive by using pull-down menus, buttons and sliders. The utility can also be used to update the controller's software in the field as described in "Updating the Controller's Software" on page 137.

---

## System Requirements

To run the utility, you will need the following:

- PC compatible computer running Windows 98, Me, 2000 or XP
- An unused serial communication port on your computer with a 9-pin, female connector
- An Internet connection for downloading the latest version of the Roborun Utility or the Controller's Software
- 5 Megabytes of free disk space

If you do not have a free serial port, the Configuration Utility can still run, but you will not be able to communicate with the controller.

---

## Downloading and Installing the Utility

The Configuration Utility is included on the CD that is delivered with the controller or may be obtained from the download page on Roboteq's web site at [www.roboteq.com](http://www.roboteq.com). It is recommended that you use the downloaded version to be sure that you have the latest update.

- download and run the file `robosetup.exe`
- follow the instructions displayed on the screen
- after the installation is complete, you will be able to run the program from your Start Menu > Programs > Roboteq

You do not need to have the controller connected to your PC to start the Utility.

---

## Connecting the Controller to the PC

The controller must be connected to the PC if you plan to use the Utility to perform any of the following functions

- to read the current parameters stored in the controller and display them on the computer
- to store new parameters in the controller
- to exercise the motors using your PC
- to update the controller's software

If the controller is not connected, the Configuration Utility can run and may be used to automatically generate the setting codes for manual entry. See "Viewing the Parameters Summary" on page 133.

Most computers have at least one, but often times two serial ports. Look for one or two connectors resembling the illustration in Figure 70.

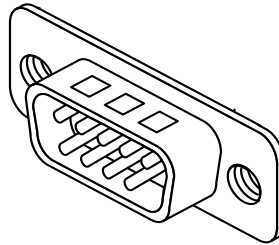


FIGURE 70. Look for a 9-pin male connector on your PC

---

If you have a serial port connector that is already connected to something else, you may be able to unplug your current device and temporarily connect the controller as long as the software operating the current device is not running.

Connect the provided serial cable to the controller on one end and to the PC on the other.

Power the controller, preferably using the yellow Power Control wire, with a 12V power supply. Preferably, you should use a 12V battery. However, a 9VDC battery or 9-12VDC wall-transformer with 200mA of minimum output will work.

Connect the thin black wire to the negative (-) terminal, and the thin yellow wire to the positive (+) of the power supply. The controller will turn On. If it doesn't, verify that you have not reversed the polarity.

Upon powering On, the controller will display "no ctrl" if configured in the R/C mode or a steady pattern if configured in the RS232 mode.

## Roborun Frame, Tab and Menu Descriptions

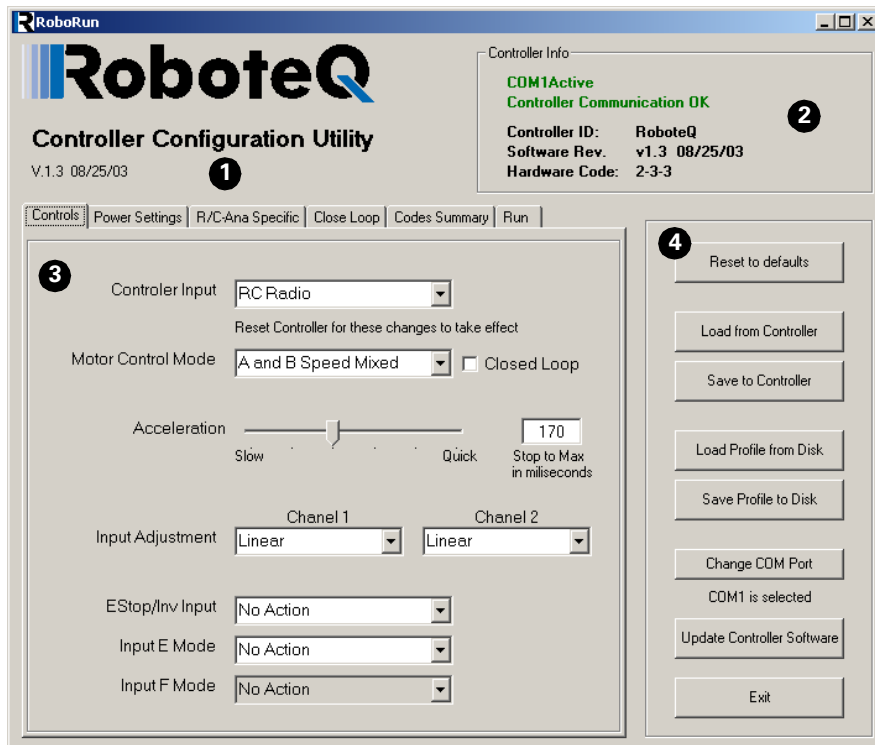


FIGURE 71. Roborun screen layout

The Roborun screen contains the four main set of commands and information frames described below:

### 1- Program Revision Number

This is the revision and date of the Roborun utility. It is recommended that you always verify that you have the latest revision of the utility from Roboteq's web site at [www.roboteq.com](http://www.roboteq.com)

### 2- Controller and Communication Link Information

This frame will automatically be updated with an indication that a free communication port was found and opened by the utility.

If no free communication port is available on your computer, it will be indicated in this window. You should try to select another port using the "Change COM Port" button or try to free the port if it is used by a different device and program.

With the port open, Roborun will try to establish communication with the controller. If successful, this window will display the software revision, the revision date and a set of digits identifying hardware revision of the board inside the controller.

### 3- Parameter Selection and Setting, and Special Functions

This is the program's main frame and includes several types of tabs, each of which has several buttons, menus and other User Interface objects. These tabs and the functions they contain are described in detail in the following sections.

You can go from one set of commands to another by clicking on the desired tab.

#### **4- File and Program Management Commands**

This frame contains a variety of buttons needed to load and save the parameters from and to the controller or disk. This frame also contains the button needed to initiate a software update to the controller.

---

### **Getting On-Screen Help**

The Roborun buttons and fields are very intuitive and self-explanatory. Additional explanations and help is provided by means of ToolTips for each command. Simply move the cursor to a button, tab or other gadget on the screen and a message box will appear after a few seconds.

---

### **Loading, Changing Controller Parameters**

The first set of tabs allows you to view and change the controller's parameters. These tabs are grouped according to the general type of parameters (Controls, Power Setting, and R/C Settings).

When starting Roborun, this screen is filled with the default values. If the controller is connected to your PC, Roborun will automatically detect it and ask you if you wish to read its settings.

You can read the controller's setting in the PC at any other time by pressing the "Load from Controller" button. After changing a parameter, you must save it to the controller manually by pressing the "Save to Controller" button.



## Controls Settings

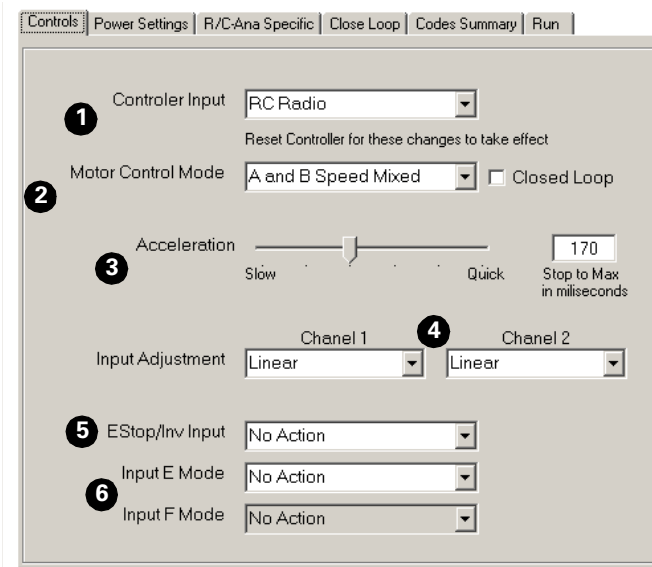


FIGURE 72. Control modes setting screen

The screen shown in Figure 72 is used to view and change the controller's main control modes. Below is the list of the parameters accessible from this screen

### 1- Controller Input:

This pull down menu allows you to select the RS232, R/C or Analog mode of operation. If the RS232 mode is selected, a check box will appear, allowing you to enable or disable the RS232 Watchdog. For more information on these modes, see

- "R/C Operation" on page 63
- "Serial (RS-232) Controls and Operation" on page 79
- "RS-232 Watchdog" on page 90
- "Analog Control and Operation" on page 97

### 2- Motor Control Mode

This pull down menu is used to choose whether the controller will operate in Separate or Mixed mode. For more information on these modes, see "Selecting the Motor Control Modes" on page 46.

### 3- Acceleration Setting

This slider will let you select one of seven preset acceleration values. The label on the right shows a numerical value which represents the amount of time the controller will take to accelerate a motor from idle to maximum speed. See "Programmable Acceleration" on page 49.

### 4- Input Adjustment

These pull down menus will let you select one of five conversion curves on each of the input command values. See “Command Control Curves” on page 50.

### 5- Emergency Stop or Invert Switch Select

This pull down menu will let you select the controller’s response to changes on the optional switch input: Emergency Stop, Invert Commands, or no action. See “Emergency Stop using External Switch” on page 53 and “Inverted Operation” on page 53.

### 6- Effect of Digital Inputs

This pull down menu will let you select the controller’s response to changes on either of the two digital inputs. See “Special Use of Accessory Digital Inputs” on page 54.

## Power Settings

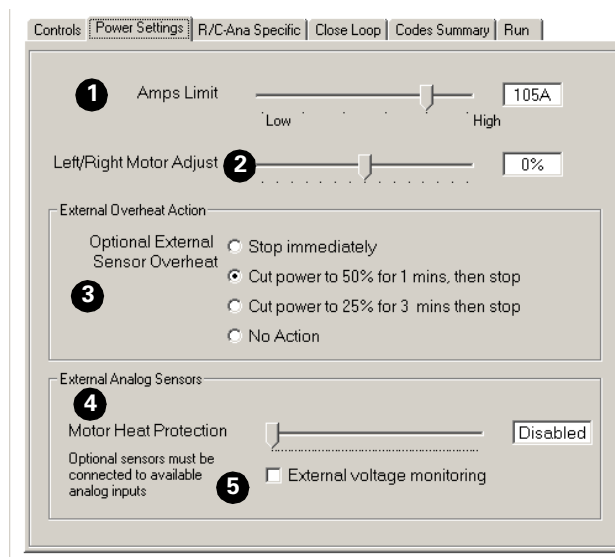


FIGURE 73. Power settings screen

The screen shown in Figure 73 is used to view and change the power parameters of the controller.

#### 1- Amps limit

This slider will let you select the max amps that the controller will deliver to the motor before the current limitation circuit is activated. See “Peak Current Protection” on page 49.

#### 2- Left/Right Adjust

This slider will let you configure the controller so that it applies more power to the motors in one direction than in the other. See “Left / Right Tuning Adjustment” on page 51.

#### 3- Action upon External Overheat Condition

These radio buttons will let you choose the controller's response to an overheat condition detected on any external components on which a temperature sensor is mounted. See "Connecting External Thermistor to Analog Inputs" on page 41 for information on how to wire these sensors.

The AX2550 contains internal temperature sensor which will automatically act on the allowed current limit. The buttons on this screen have no effect on how the controller will respond to an internal overheat situation.

## 4- External Motor Heat Protection

This slider will let you select the temperature at which an external overheat condition will be detected. If the slider is all the way to the left, motor heat detection will be disabled. See "External Devices Overheat Monitoring" on page 54.

## 5- External Voltage Monitoring

This box is enabled if the analog input is not used for an external temperature, speed or position sensor. Checking this box does not affect the controller in any way. Its purpose is to enable the computation and display of the temperature on the Run screen of the Robo-run utility. See "Running the Motors" on page 133. A resistor network must be connected to the voltage source you wish to monitor as shown in "Using the Analog Inputs to Monitor External Voltages" on page 42. Note that internal voltages are monitored using the AX2550's internal sensors.

## Analog or R/C Specific Settings

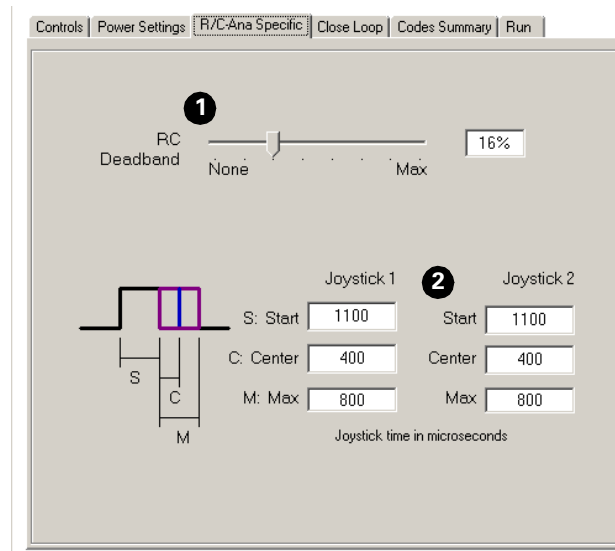


FIGURE 74. Power settings screen

The screen shown in Figure 74 is slightly different whether the Analog Input mode is selected or not.

If the Analog Input mode is selected on the main screen, then this page is used to set the Analog Deadband value. In the R/C mode, this page is used to view and change parameters used in the R/C mode of operation. None of these parameters has any effect when running the controller in RS232 mode.

If the controller is configured in RS232 mode, some of these menus will be grayed-out but will remain active.

### 1- Deadband

This slider will let you set the amount of joystick motion off its center position before the motors start moving. The slider will work identically in the R/C or analog mode, however, the % value will be different. See “Joystick Deadband Programming” on page 72 and “Analog Deadband Adjustment” on page 101.

### 2- Joystick Timing

These fields are enabled only if the R/C mode is selected. These number areas will let you read and modify the R/C pulse timing information used by the controller. New values can be entered manually to create different capture characteristics. They are also useful for viewing the stored values after an automatic joystick calibration sequence. See “Joystick Calibration” on page 73 and “Automatic Joystick Calibration” on page 74.

## Closed Loop Parameters

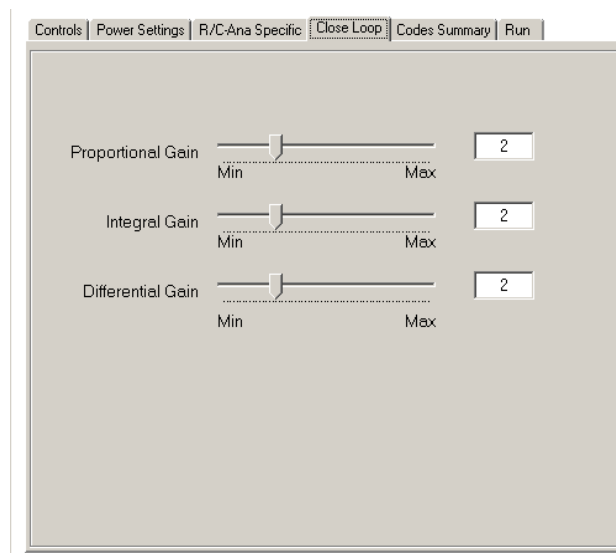


FIGURE 75. Closed Loop parameter setting screen

This screen shown in Figure 75 is used to set the Proportional, Integral and Differential gains needed for the PID algorithm. These parameters are used in the Position mode (see page 105) and the Closed Loop speed mode (see page 113).

## Viewing the Parameters Summary

The summary page will list the command letter and value for each parameter, as defined in the various parameter setting screens.

It is possible to print this screen and bring it to the controller in order to then quickly and accurately update the controller's parameters using the switch and display. See "Programming using built-in Switches and Display" on page 119.

## Important Notes:

**Most but not all configuration parameters can be updated using the switches.**

**The print command will automatically use your default printer. No print setup box will be displayed.**

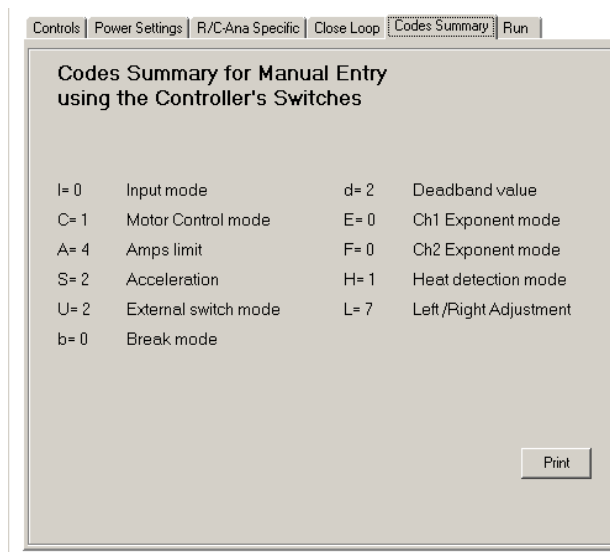


FIGURE 76. Viewing parameters summary

## Running the Motors

The Roborun utility will let you exercise and monitor the motors, sensors and actuators using your computer. This feature is particularly useful during development as you will be able to visualize, in real-time, the robot's Amps consumption and other vital statistics during actual operating conditions.

Figure 77 shows the Run Screen and its numerous buttons and dials.

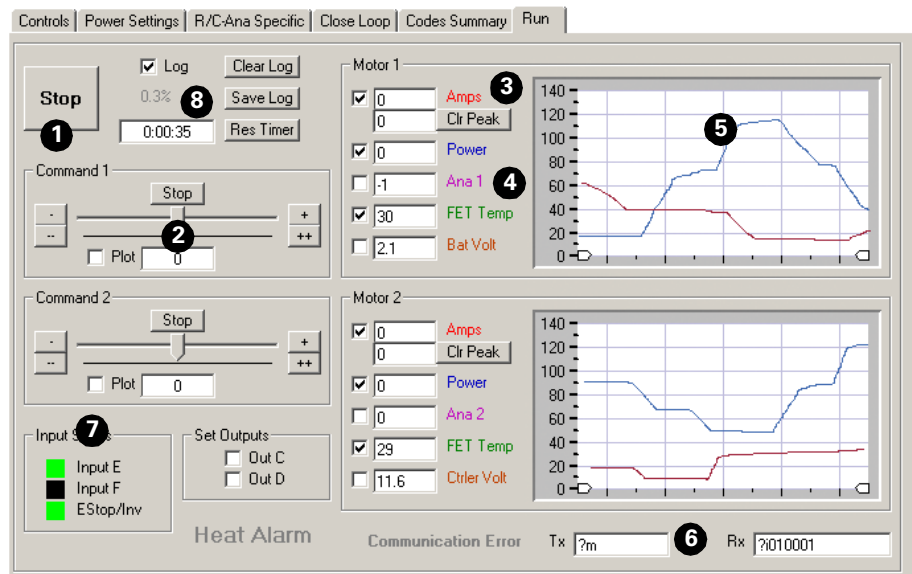


FIGURE 77. Motor exercising and monitoring screen

### 1- Run/Stop Button

This button will cause the PC to send the run commands to the controller and will update the screen with measurements received from the controller.

When the program is running, the button's caption changes to "Stop". Pressing it again will stop the motors and halt the exchange of data between the PC and the controller.

If you select another tab while the program is running, the program will stop as if the Stop button was pressed.

### 2- Motor Power setting

This sub-frame contains a slider and several buttons. Moving the slider in any direction away from the middle (stop) position will cause a power command to be issued to the controller. The value of the command is shown in the text field below the slider.

The stop button will cause the slider to return to the middle (stop) position and a 0-value command to be sent to the controller. The + and ++ buttons will cause the slider to move by 1 or 10 power positions respectively.

### 3- Amps Measurement

These two fields display the Amps measured and reported in real-time by the controller. The peak amp field will store the highest measured Amp value from the moment the program began or from when the peak was reset using the Reset button.

### 4- Power, Analog, Temperature and Battery Measurements

The power field displays the power level that is actually being applied to the motor. This value is directly related to the motor command except during current limitation, in which case the power level will be the one needed to keep the Amps within the limit.

The analog field contains the analog input value that is measured and reported by the controller. If External Motor Heat Protection is enabled, this field will display the temperature measured by the external thermistor. If Battery Monitoring is enabled, this field will display the measured battery voltage.

Note that in order to measure and display external temperature or voltage, the proper external components must be added to the controller. See "Connecting External Thermistor to Analog Inputs" on page 41 and "Using the Analog Inputs to Monitor External Voltages" on page 42.

When used with the AX2550, two more fields are displayed for monitoring the Heatsink temperature, the Main Battery voltage, and the internal 12V voltage.

### **5- Real-Time Strip Chart Recorder**

This chart will plot the actual Amps consumption and other parameters as measured from the controller. When active, the chart will show measurement during the last five seconds.

The two handles at the bottom of the chart are zoom in or out the history sample that is to be displayed on the chart. Using these handles it is possible to go back and examine any event up to 5 minutes into the past. This feature is extremely useful during development and test of the robotic vehicle.

### **6- Transmit and Receive Data**

These two fields show the data being exchanged between the PC and the controller. While these fields are updated too fast to be read by a person, they can be used to verify that a dialog is indeed taking place between the two units.

After the Start button is pressed, the Tx field will show a continuous string of commands and queries sent to the controller.

The Rx field will display the responses sent by the controller. If this field remains blank or is not changing even though the Tx field shows that data is being sent, then the controller is Off or possibly defective. Try resetting the controller and pressing the Run/Stop button.

### **7- Input Status and Output Setting**

This section includes two check boxes and three color squares. The check marks are used to activate either of the controller's two outputs. The color blocks reflect the status of the three digital inputs present on the controller. Black represents a "0" level. Green represents a "1" level.

### **8- Data Logging and Timer**

A timer is provided to keep track of time while running the motors. An additional set of buttons and displays are provided to operate a data logger. The data logger is fully described in the section that follows.

## Logging Data to Disk

While running the motors, it is possible to have Roborun capture all the parameters that were displayed on the various fields and charts, and save them to disk. The log function is capable of recording 32,000 complete sets of parameters, which adds up to approximately 15 minutes of recording time. The figure below details the buttons and check boxes needed to operate this function.

### 1- Log Check Box

When checked, Roborun will capture all the parameters and save them in local RAM. The data is not saved to disk until the "Save to Disk" button is pressed. Data is being captured for as long as the program is in the Run mode, whether or not a motor command is applied.

### 2- Clear Log

This button can be pressed at any time to clear the local RAM from its content. Clearing the log also has the effect of resetting the timer.

### 3- Log Fill Status

This grayed text box indicates whether the local RAM log is empty, full, or in-between.

### 4- Reset Timer button

The timer automatically runs when the Run button is pressed and data is being exchanged with the controller, regardless whether logging is activated. This button resets the timer.

### 5- Save Log to Disk button

Pressing this button will prompt the user to select a filename and location where to copy the logged data. The file format is a regular text file with each parameter saved one after the other, separated by a coma. The file extension automatically defaults to .csv (coma separated values) so that the data can be imported directly into Microsoft Excel. The first line of the save file contains the Header names. Each following line contains a complete set of parameters. The Header name, order and parameter definition is shown in Table 22

TABLE 22. Logged parameters order, type and definition

Parameter Header	Data type/range	Measured Parameter
Seconds	Integer	Timer value expressed in seconds
Command1	-127 to +127	Command applied to channel 1
Command2	-127 to +127	Command applied to channel 2
Power1	0 to 127	Amount of power applied to the output stage of channel 1
Power2	0 to 127	Same for channel 2
Ana 1, Speed 1, Pos 1 or Temp 1 or Volt 1	-127 to + 127 -40 to +150 0 to 55	Value of sensor connected on analog input 1. Data is automatically converted to the right value and format by Roborun according to the sensor that is being used



TABLE 22. Logged parameters order, type and definition

Parameter Header	Data type/range	Measured Parameter
Ana 2, Speed 2, Pos 2	-127 to + 127	Same for channel 2
Temp 2 or	-40 to +150	
Volt 2	0 to 55	
Amps1	0 to 255	Measured Amps on channel 1
Amps2	0 to 255	Measured Amps on channel 2
FET Temp1	-40 to +150	Measured Temperature on channel 1's heatsink.
FET Temp2	-40 to +150	Measured Temperature on channel 2's heatsink.
Batt Volt	0 to 55	Main Battery Voltage.
Ctrl Volt	0 to 28.5	Internal 12V Voltage.

## Viewing and Logging Data in Analog and R/C Modes

When the controller is configured in R/C or Analog mode, it will automatically and continuously send a string of ASCII characters on the RS232 output. "Analog and R/C Modes Data Logging String Format" on page 92 shows the nature and format of this data.

This feature makes it possible to view and log the controller's internal parameters while it is used in the actual application. The data may be captured using a PC connected to the controller via an RS232 cable or wireless modem.

When wired for R/C or Analog controls, the AX2550 will not be able to receive commands from the PC and the Roborun software will not recognize the controller as being present. However, when in the Run tab and the Run button activated, Roborun will be receiving the strings sent by the controller and display the various parameters in the right display box and chart.

## Loading and Saving Profiles to Disk

It is possible to save the configuration parameters that are read from the controller or that have been set/changed using the various menus to the disk. This function will let you easily recall various operating profiles at a later time without having to remember or manually reset all the parameters that are used from one configuration to another.

To save a profile to disk, simply click on the "Save Profile to Disk" button. You will then be prompted to choose a file name and save.

Reading a profile from disk is as simple as clicking on the "Load Profile from Disk" button and selecting the desired profile file. The parameters will be loaded in each of their respective buttons, sliders and text fields on the various Roborun screens. **The parameter will not be transferred to the controller until you press the "Save to Controller" button.**

---

## Updating the Controller's Software

The AX2550's operating software can be easily upgraded after it has left the factory. This feature makes it possible to add new features and enhance existing ones from time to time.

## Important Warning

**Updating the controller will cause all its parameters to reset to their default conditions. You should re-enter these parameters to your desired value prior to re-installing and using the controller.**

The upgrade procedure is quick, easy and error proof:

1- Connect the controller to the PC via the provided RS232 cable.

2- Apply a 12V power supply to the controller's thin yellow and black cables. If you wish to perform the updating procedure near your PC and do not have a 12V power supply, you may use a small and fresh 9V battery for the short duration of the update. **Never exceed 14V as this may cause permanent damage to the controller.**

3- Launch the Roborun utility if it is not already running. Then click on the "Update Controller Software" button.

4- If the controller is On, Roborun will find it and prompt you to select the new software file. It may happen that the controller is not responding properly and you may be asked to reset it while connected.

5- Press the "Program" button to start programming. **Do not interrupt or cut the power to the controller while the new program is loading its Flash memory.**

6- After a verification, you will be notified that the operation was successful and you will see the new software revision and date as reported by the controller.

Notes:

It is a good idea to load the controller's parameters into the PC and save them to disk prior to updating the software. After the new software is transferred to the controller, you can use the "Load Parameters from Disk" function and transfer them to the controller using the "Save to Controller" button.

## SECTION 14

# Mechanical Specifications

This section details the mechanical characteristics of the AX2550 controller.

## Mechanical Dimensions

The AX2550 is built into a extruded aluminum case which also serves as a heat sink for its electronics. Figure 78 and Figure 79 show drawings of the controller and its mechanical dimensions.

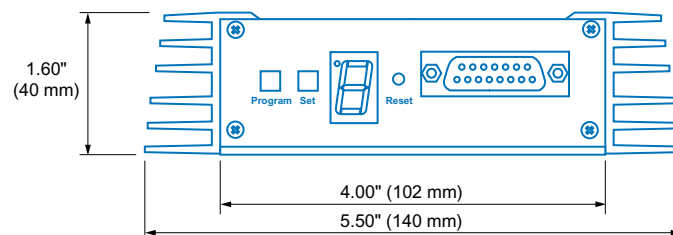


FIGURE 78. AX2550 front view and dimensions

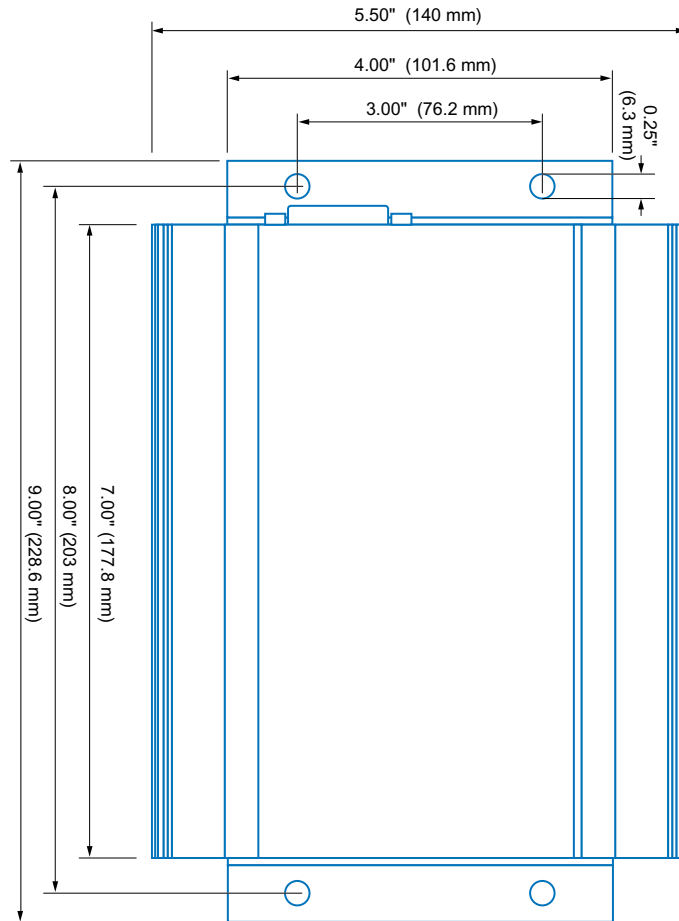


FIGURE 79. AX2550 top view and dimensions

## Mounting Considerations

Mounting brackets on each end are provided for durable assembly of the controller onto any robot chassis.

The AX2550 uses no jumpers and includes no loose parts or wires. Consequently, it will be able to run reliably in severe shock and vibration environment.

Optional shock mount absorbers may be added for use in the most brutal environment.

## Thermal Considerations

The AX2550 is built in a heat-sinking aluminum extrusion. The large fin area on the side of the extrusion ensures sufficient heat dissipation for operation without a fan in most applications.

For high current applications, it is possible that the controller may heat-up faster and to a higher temperature than can be dissipated by the enclosure using natural convection alone.

In these applications, you should ensure that air flow exists to remove the heat from the heat sink. In the most extreme use, you should consider using external fan to circulate air around the controller.

Alternatively, you may mount the controller so that its bottom area is in contact with large metallic parts of the robot's body. This would allow some of the controller's heat to dissipate through conduction to additional metal mass, and therefore remain cooler for longer periods of time. Apply thermal grease between the controller's bottom and the robot's chassis to ensure the best thermal conduction.

## Wire Dimensions

A total of eight heavy gauge wires, and two thin wires are present at the back of the controller. Their length and dimension is shown in Table 23 below.

TABLE 23. AX2550 wire identification and dimensions

Wire	Gauge	Outside Diameter	Color	Length
Motor1 Battery+	AWG8	0.26" (6.6mm)	Red	17"(43mm)
Motor1 Battery-	AWG8	0.26" (6.6mm)	Black	17"(43mm)
Motor2 Battery+	AWG8	0.26" (6.6mm)	Red	17"(43mm)
Motor2 Battery-	AWG8	0.26" (6.6mm)	Black	17"(43mm)
Motor1 Terminal+	AWG8	0.26" (6.6mm)	White	17"(43mm)
Motor1 Terminal-	AWG8	0.26" (6.6mm)	Green	17"(43mm)
Motor2 Terminal+	AWG8	0.26" (6.6mm)	White	17"(43mm)
Motor2 Terminal-	AWG8	0.26" (6.6mm)	Green	17"(43mm)
Power Control and Backup Controller Battery+	AWG22	0.07" (1.8mm)	Yellow	17"(43mm)
Controller Battery-	AWG22	0.07" (1.8mm)	Black	17"(43mm)

