

Jade Robot Console Interface



mimetics
digital education

Myke Predko

Last Updated: December 12, 2014

License and Warranty

This document and code was written for the Mimetics Jade Robot™ and follow on products.

This document and code is considered Mimetics Proprietary and may not be released outside of Mimetics except by permission of Mimetics, Inc.

Software Compatibility

The script language described in this document was written to be supported by:

Robot Software Release 39 or later, except where noted

Jade Support Version 0.9.7.1 or later

Robot Tokenizer Version 0.11.12 or later

Syscall API 0.0.4 or later

Panel Processor 0.3.1 or later

Bitmap Processor 0.0.1 or later

Wave Processor 0.2.0 or later

_start.s defined as _Header8.script or later

Conventions, Options and Selections

Example code will be put in monospace font like:

```
A = B + C
```

In the language definition, there are a number of instances where there are optional parameters or multiple parameters for the same task. To make these situations more obvious, the following convention is used:

[] – Optional parameter

| - One parameter or another

... – Previous parameter can be repeated

<none> - indicates that nothing is a possible option

Overview

This document outlines the communications protocols between a host PC and a Jade Robot, either through Bluetooth communications or USB. There are two communications modes available to the developer: “Command Mode” and “Rover Mode”. Command Mode communications are short ASCII strings terminated in an ASCII Carriage Return (“\r” – 0x0D) character but for program downloading and image transfer, data is sent as blocks of 8 bit bytes. Rover Mode communications are designed for passing jpeg data from the Jade Robot to a program that will display them for users and are terminated with Commands maybe passed to the Jade Robot at any time, but they may be rejected or ignored based on the current operating state of the Jade Robot.

The maximum command length (including terminating “\r”) is 63 bytes.

The primary development environment expected to be used for these commands is known as “Jade Support”, a custom Google App that can be downloaded and installed by the user and provides communications between the Jade Robot and Jade Support is via Bluetooth. Bluetooth provides a fast, reliable wireless interface to the Jade Robot that allows application development, download and debug without cables, making a development a faster, more efficient operation.

Secondary uses of the information contained in this document is for remote-controlling the Jade Robot as well as updating the Flash Firmware.

It should be noted that the primary use for the USB interface is for Firmware Updates of the Jade Robot. While it can be used for downloading and debugging applications as well as controlling the Jade Robot, the wired connection makes for a cumbersome developer/user experience.

Support

Along with this document, more comprehensive documentation of the Jade Robot script programming language, syscall APIs is available on the Mimetics web page at:

<http://www.mimetics.ca/knowledge-base-2/>

Mimetics has a forum for news, questions and sample programs. They can be found here:

<http://www.mimetics.ca/forum/>

Mimetics is active on social media, look for “MimeticsCanada” on:

- Twitter: <http://twitter.com/MimeticsCanada>
- Tumblr: <http://www.tumblr.com/blog/mimeticscanada>
- YouTube: <http://www.youtube.com/channel/UCu26ZMBIK9fBgmUOKfwK3lg>

Finally, this document will be undergoing constant updating, please check for the latest version at:

<http://www.mimetics.ca/knowledge-base-2/>

Jade Robot Command Mode:

“Command Mode” is used for basic communications between a host device and the Jade Robot. It provides the ability to:

- Control the operation of a program loaded in the Jade Robot’s file system. This includes debug functions.
- Manage, upload and download contents of the Jade Robot’s file system.
- Access the motors, OLED, sensors and other devices on the Jade Robot. Access to these features is also possible in “Rover Mode”.
- Update the Jade Robot’s Firmware.

Command Mode commands are sent to the Jade Robot terminated with a “Carriage Return” (0x0D or “\r”). Each command is replied to with a string that is terminated with a “Carriage Return”.

If a command is being sent in error, then it can be halted by the use of the “Break” (Ctrl-C or 0x03) character. This feature was put in for user operations in which a terminal emulator is used to communicate with the Jade Robot and the user feels there is an error in the data and wants to restart the operation without the current data being processed by the Jade Robot.

NOTE: Some “Command Mode” commands can be accepted in “Rover Mode”, but most cannot. In the “Rover Mode” section, the commands that can be used in “Rover Mode” are listed. The commands that can be used in “Rover Mode” do not return in the same way as “Command Mode” commands so interface software needs to be able to expect the actual return.

Jade Robot Operating States

Command Mode is the default for the Jade Robot and within it there are four main operating states. The current operating state can be found by using the “status\r” command. The “status\r” command can be used to monitor the status of an executing Jade Robot script application.

For the different operating states the return strings are in the same format:

```
AppName   Offset   LineNumber  STATE\r
```

When the different states are discussed below, only the “STATE” response will be explained (as the Application Name, Offset within the Application and currently executing Line Number are self-explanatory and used for debugging).

No Script Loaded

The script engine has not received the “load scriptname.s\r” and “resume\r” commands necessary to execute a script. The “status\r” command will return the STATE message (with all others having placeholder information):

```
NOPGM
```

Script Loaded or Ready for Execution

After the “load scriptname.s\r” or a “halt\r” command for a running program, the “status\r” command will return the current application, the offset within the application and the current line number along with the STATE message:

```
READY
```

Script Executing

When an application is executing, there are several different status messages that will be returned along with the currently executing application, its offset and line number. If the application is running, then the “status\r” command will return a STATE of:

```
RUN'G
```

If the application is currently executing a “delay” statement, then the “status\r” command will return a state of:

```
DELAY
```

Finally, if the application is waiting for a “synch” loop to complete, the “status\r” command will return a state of:

```
SYNCH
```

Script Finished Executing

There are two reasons why an application will have stopped executing (other than the “halt/r” command). First is that it has completed. If this is the case, the returned state will be:

DONE

If there is an error, then the returned state will be:

ERROR

In the case of the “ERROR”, to have the error returned, the “statusmsg\r” command with “errmsgnxt\r” is used.

Rover Mode

If the “status” message is attempted with the Jade Robot in Rover Mode, the message “Rover Active\r” will be returned - BUT, this message maybe garbled by the continuous stream of jpeg data being sent by Rover Mode.

Commands

The following commands are available to access the Jade Robot. Note that when the Jade Robot is executing an application or it is in Rover Mode, many of the commands are unavailable and errors will be returned if the commands are made when the Jade Robot is in these states – these are the “Restrictions” listed in the command descriptions below. Also note that many commands require follow on commands and are a unique operating state that can only end with the end of the data being returned or sending a “halt\r” command:

*\r	
Ping the Jade Robot to check its operation.	
Passed Argument: none	Returned Data: “_*\r”
Execution Restrictions: Not Available in Rover Mode	
Error: None	

halt\r	
Stop whatever operation is currently active. If nothing is running or no command states requiring additional commands are active, then the “halt\r” is ignored.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: none	
Error: None	

paneloff\r	
Stop the panel driver executing.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode	
Error: None	

ver\r	
Return Jade Robot Firmware Version. This is a long string of numbers separated by dots (“.” 0x2E). Normally, the first number is referred to as the Firmware “Release” number and is the one shown on the Jade Robot’s OLED display.	
Passed Argument: none	Returned Data: ASCII Decimals spaced by dots and “\r”
Execution Restrictions: Not Available in Rover Mode	
Error: None	

hwver\r	
Return Jade Robot Hardware Version. Currently this is “Robot 1.00”.	
Passed Argument: none	Returned Data: “Robot 1.00\r”
Execution Restrictions: Not Available in Rover Mode	
Error: None	

btser\r	
Return the 12 digit Jade Robot Bluetooth MAC Address. NOTE: the string address returned is 14 digits long.	
Passed Argument: none	Returned Data: 14 Digit MAC Address String "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

getname\r	
Return the Jade Robot's Bluetooth "Friendly Name". This name is also used to identify the Jade Robot.	
Passed Argument: none	Returned Data: Bluetooth Friendly Name "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

proctype\r	
Return the Jade Robot's Processor Type in the format: "K##-##(##)P-R(##)".	
Passed Argument: none	Returned Data: Processor Type information
Execution Restrictions: Not Available in Rover Mode	
Error: None	

status\r	
Return the current execution status of the Jade Robot.	
Passed Argument: none	Returned Data: Described elsewhere in this document.
Execution Restrictions: Not Available in Rover Mode	
Error: None	

statusmsg\r	
Return verbose status message. This is normally used for returning current error message when the "status" message has returned "ERROR".	
Passed Argument: none	Returned Data: Current error state "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

errmsgnxt\r	
Normally used to display error message after "stepin\r" returns "Error". When the message has been fully displayed, "errmsgnxt" just returns "\r" indicating there are no more errors to display.	
Passed Argument: none	Returned Data: Current Not Displayed Error Message "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

usbstatus\r	
Indicate whether or not the Jade Robot is connected to the PC via USB.	
Passed Argument: none	Returned Data: "(NO) USB Plugged in\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

dir [*.*]\r	
Return File list for specified files. Windows "*" wildcard can be used in specifying files. If nothing specified (ie "dir\r") then it is assumed to be "dir *.*\r". If no file found, then just "\r" is returned, else the file with its size and CRC value. "dirnxt\r" must be sent afterwards until just a "\r" string is returned. The "dir ..\r" operation can be stopped using the "halt\r" command.	
Passed Argument: none	Returned Data: File Information "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name or Wildcard Specified	

dirnxt\r	
Return next file started by "dir ..\r" command. This command continues until no more files meeting the original description are found (returning just "\r"). The "dir ..\r" operation can be stopped using the "halt\r" command.	
Passed Argument: none	Returned Data: File Information "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing. MUST be after "dir ..\r" or "dirnxt\r" command.	
Error: None	

filemap\r	
Return File list for all specified files in the Jade Robot. "filemapnxt\r" must be sent afterwards until just a "\r" string is returned. The "filemap\r" operation can be stopped using the "halt\r" command.	
Passed Argument: none	Returned Data: File Information "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

filemapnxt\r	
Return next file started by "filemap\r" command. This command continues until no more files are found (returning just "\r"). Note that before returning just "\r", the "END" record with the amount of space left is shown. The "filemap\r" operation can be stopped using the "halt\r" command.	
Passed Argument: none	Returned Data: File Information "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing. MUST be after "dir ..\r" or "dirnxt\r" command.	
Error: None	

avail\r	
Return number of bytes available in the Jade Robot's File system.	
Passed Argument: none	Returned Data: "#\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

del fileName.e\r	
Delete the specified Filename and extension. No wildcards are allowed in the file name. When looking at the space occupied by the file after deleted (using "filemap\r"), it will be marked as "Deleted".	
Passed Argument: fileName.e	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name, File Not Found	

deleteall\r	
Delete ALL the files in the Jade File system. The file system will be wiped clean and will not require garbage collection. This operation will take several seconds to complete.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: none	

garbage\r	
Perform Garbage Collection on Jade Robot File System. This will eliminate any "Deleted" files and free up the space at the end of Flash. This operation will take several seconds to complete.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

meta fileName.s\r	
Start dumping the meta description of the specified script. Data will be returned as an ASCII string terminated in a "\r". the "metanxt\r" command is used until just a "\r" is returned which means that all of the description has been returned. The "halt\r" instruction is used to stop the meta description read. NOTE: for use with the _Header#.s files, the start of the meta description is a short description of the script terminated in a colon (":" 0x3A) with a more complete description following.	
Passed Argument: none	Returned Data: "meta description string\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name, File Not Found	

metanxt\r	
Continue returning the contents of the file's meta description. Data returned is an ASCII string. The command can be stopped using the "halt\r" command. Reading after completion will just return "\r".	
Passed Argument: none	Returned Data: "meta description string\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

download fileName.e size\r	
-----------------------------------	--

Start downloading a file from the host system into the Jade Robot. The size is in the number of bytes.	
Passed Argument: fileName.e & size	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name, File Name Already Exists, Insufficient Space for File	

block ## ## ... ##\r	
Download file contents up to 16 bytes at a time. Download operation stops when all bytes have been received or a “halt\r” command is received. Each data byte is passed as two nibbles (“0” to “F”) with a space separating them.	
Passed Argument: ## ## ... ##	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

block @#data...\r	
Alternative to downloading three bytes for each byte. In this mode, the “@” indicates that actual data will be passed with “#” being the number of bytes (up to 48). This alternative approach is to provide a faster data transfer method.	
Passed Argument: @#data...	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

flashwrite hexAddress\r	
Specify a location in Flash to start downloading code at the specified address (which is a hexadecimal number). This is used for reflashing the Jade Robot’s MCU. The “flashblock ...\r” command is used to pass data at the current address. To stop the flash update, a “halt\r” command is sent.	
Passed Argument: Address	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: none	

flashblock ## ## ... ##\r	
Download flash update contents up to 16 bytes at a time. Download operation stops when all bytes have been received or a “halt\r” command is received. Each data byte is passed as two nibbles (“0” to “F”) with a space separating them.	
Passed Argument: ## ## ... ##	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

flashblock @#data...\r	
Alternative to downloading three bytes for each byte. In this mode, the “@” indicates that actual data will be passed with “#” being the number of bytes (up to 48). This alternative approach is to provide a faster data transfer method.	
Passed Argument: @#data...	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

flashread hexAddress\r

Return 16 bytes starting at the specified, hexadecimal address. Each data byte is returned as two bytes, each representing a nibble (“0” to “F”) and a space separating them. This command is used to verify the contents of the MCU Flash.

Passed Argument: Address

Returned Data: “## # ... ##\r”

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: none

block1crc\r

Return the decimal CRC of Block 1 of the Jade Robot MCU. This command is used to verify the load of the new Flash image before swapping blocks.

Passed Argument: none

Returned Data: “####\r”

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: None

swapon\r

Open Flash Block 1 as a reset Flash Block swap destination. First operation in updating Firmware after new image has been written into Block 1.

Passed Argument: none

Returned Data: “\r”

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: None

do_swap\r

Set internal registers of MCU to accept Block 1 as the new Block 0 start target. Second operation in updating Firmware after new image has been written into Block 1.

Passed Argument: none

Returned Data: “\r”

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: None

swapoff\r

Close Flash Block 1 as a reset Flash Block Swap destination. Third operation in updating Firmware after new image has been written into Block 1.

Passed Argument: none

Returned Data: “\r”

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: None

do_reset\r

Wait 5 seconds before forcing an MCU reset. The final operation in updating Firmware after new image has been written into Block 1. During the 5 second delay, the BT and USB connections need to be broken. The Jade Robot can be reconnected after it has reset.

Passed Argument: none	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

swapstatus\r

Return current status of Flash swap operation. The number returned indicates the current state of the swap operation.

Passed Argument: none	Returned Data: “#\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

cur_copy\r

Copy the contents of MCU Flash Block 0 into Block 1. This function is primarily used for testing the MCU Firmware Update Functions.

Passed Argument: none	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

dump fileName.e\r

Start dumping the contents of the specified filename as hex data. Data will be returned 16 bytes at a time using “hexdumpnxt\r” until the file has been read to completion (at which point just “\r” is returned) or a “halt\r” instruction is received.

Passed Argument: none	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name, File Not Found	

hexdumpnxt\r

Continue dumping the contents of the file specified in the “hexdump” command. Data will be returned 16 bytes at a time using “hexdumpnxt\r” until the file has been read to completion (at which point just “\r” is returned) or a “halt\r” instruction is received. Data returned will be two characters representing each nibble of the hex byte (from “0” to “F”) with a space separating them. If the file length is not a multiple of 16, then just the modulo number of bytes will be returned on the last line. Reading after completion will just return “\r”.

Passed Argument: none	Returned Data: “## ## ... ##\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

getbutton\r

Return the current state of the Jade Robot user interface buttons. String returned in “udlre” for “Up”, “Down”, “Left”, “Right” and “Enter” (Red button). When the buttons are not pressed, the lower case letter is returned, when they are pressed, the upper case letter is returned.

Passed Argument: none	Returned Data: “udlre\r”
Execution Restrictions: Not Available in Rover Mode	
Error: None	

rssl\r	
Return the Bluetooth Signal Strength as measured by the Bluetooth adapter built in to the Jade Robot. Return Values are: NO CONNECTION, WEAK, AVERAGE, STRONG. NOTE: This command takes several (5+) seconds to return.	
Passed Argument: none	Returned Data: "rsslLevel\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

irsense\r	
Read the IR distances sensors and return values (from "0" to "100") in the format "LF-RF-LS-RS-B\r", for "L"eft "F"ront, "R"ight "F"ront, "L"eft "S"ide, "R"ight "S"ide, and "B"ack (rear).	
Passed Argument: none	Returned Data: "LF-RF-LS-RS-B\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

lightsense\r	
Read light sensors and return values (from "0" to "100") in the format "L-R-B\r", for "L"eft, "R"ight and "B"ack (rear).	
Passed Argument: none	Returned Data: "L-R-B\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

linesense\r	
Read the line sensors and return values (from "0" to "100") in the format "L-M-R\r", for "L"eft, "M"iddle and "R"ight.	
Passed Argument: none	Returned Data: "L-M-R\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

batlevel\r	
Return the current battery level as a value from "0" to "100".	
Passed Argument: none	Returned Data: "#\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

spectro\r	
Read the current Spectrometer values. Data is returned as "B"lue, "G"reen, "Y"ellow, "R"ed, "8"80nm (infra-red) and "9"40nm (infra-red).	
Passed Argument: none	Returned Data: "B-G-Y-R-8-9\r"
Execution Restrictions: Not Available when script executing	

Error: None

play fileName.w\r

Play the specified .wav file loaded in the Jade Robot.

Passed Argument: fileName.w	Returned Data: "\r"
------------------------------------	----------------------------

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: File Not Found, Invalid fileName
--

setvol #\r

Specify the volume level for the audio amplifier. Volume levels are: "0" – Off/Muted, "1" – Low, "2" – Medium, "3" – High

Passed Argument: Level	Returned Data: "\r"
-------------------------------	----------------------------

Execution Restrictions: Not Available in Rover Mode
--

Error: Invalid Volume Level Specified
--

gevol\r

Return the volume level of the audio amplifier. Values returned are defined in "setvol #\r".

Passed Argument: none	Returned Data: "#\r"
------------------------------	-----------------------------

Execution Restrictions: Not Available in Rover Mode
--

Error: None

showbmp fileName.b\r

Clear OLED display and put specified bitmap file starting at 0,0 (top left hand corner).

Passed Argument: fileName.b	Returned Data: "\r"
------------------------------------	----------------------------

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: File Not Found, Invalid Filename, Invalid File Contents

screendump\r

Save current OLED display as a bitmap file in the format "____.b" where "#####" starts at zero and increments after ever screendump command. Bitmap file is read using the "dump"/"hexdumpnxt" commands.

Passed Argument: none	Returned Data: "\r"
------------------------------	----------------------------

Execution Restrictions: Not Available in Rover Mode or when script executing

Error: None

panelload fileName.p\r

Load and start executing a panel. This command is used for testing and debugging panels without the need for a script to go along with them.

Passed Argument: fileName.p	Returned Data: "\r"
------------------------------------	----------------------------

Execution Restrictions: Not Available in Rover Mode or script executing
--

Error: None

panelstate\r	
Return the Panel Driver State. This can be done at any time and will reply with, "RUN", "END", "Ready" or "Error".	
Passed Argument: none	Returned Data: "Current Panel Status\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

panelgupdt\r	
Return flag indicating if the OLED has been updated.	
Passed Argument: none	Returned Data: "OLED:0\r" or "OLED:1\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

panelcupdt\r	
Clear the OLED update flag.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

btledon\r	
Enable the two blue LEDs indicating Bluetooth (or USB) communication. These LEDs are turned off during remote control modes when the LEDs could be distracting or add a blue tinge to the built in camera image.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: none	
Error: None	

btledoff\r	
Disable the two blue LEDs indicating Bluetooth (or USB) communication. These LEDs are turned off during remote control modes when the LEDs could be distracting or add a blue tinge to the built in camera image.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: none	
Error: None	

forward\r	
Robot moves forwards for 200ms.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

reverse\r	
Robot moves backwards for 200ms.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

left\r	
Robot turns left for 200ms.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

right\r	
Robot turns right for 200ms.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

mtrstop\r	
Stop Robot movement.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

setleftmtr ##\r	
Specify the value for the left motor from "-100" (full reverse) to "0" (stop) to "100" (full forwards).	
Passed Argument: Motor value	Returned Data: "\r"
Execution Restrictions: Not when script executing	
Error: Invalid Motor value specified	

setritemtr ##\r	
Specify the value for the right motor from "-100" (full reverse) to "0" (stop) to "100" (full forwards).	
Passed Argument: Motor value	Returned Data: "\r"
Execution Restrictions: Not when script executing	
Error: Invalid Motor value specified	

servoon\r	
Turn on power to Gripper and Elevation Servos. NOTE: The Jade Robot should be power by its battery when this command is sent.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: None	
Error: None	

servooff\r	
Turn off power to Gripper and Elevation Servos.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: None	
Error: None	

servocntr\r	
--------------------	--

Place the Gripper and Elevation servos in a “center” position.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

servoclose\r	
Close the Gripper Servo. Fully Closed is “0” and fully Open is “100”. Decrease the current servo value by 10 until value is equal to “0”.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

servoopen\r	
Open the Gripper Servo. Fully Closed is “0” and fully Open is “100”. Increase the current servo value by 10 until value is equal to “100”.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

servoup\r	
Elevate the Gripper Up. Fully Up is “1000” and fully Down is “0”. Increase the current servo value by 10 until value is equal to “100”.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

servodown\r	
Move the Gripper Down. Fully Up is “100” and fully Down is “0”. Decrease the current servo value by 10 until value is equal to “0”.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

gripset ##\r	
Set the Gripper Servo. Fully Closed is “0” and fully Open is “100”.	
Passed Argument: Servo Value	Returned Data: “\r”
Execution Restrictions: None	
Error: Invalid Servo Value	

elaset ##\r	
Set the Gripper Elevation Servo. Fully Up is “100” and fully down is “0”.	
Passed Argument: Servo value	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

camstatus\r	
Return the status of the camera. Status consists of "Camera Off\r" or "Camera Active\r".	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

camon\r	
Apply power to the Camera and initialize it for operation. NOTE: this operation will take approximately 5s.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

camoff\r	
Turn off power to the camera.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

camres #\r	
Set the resolution of the active Camera: "0" – 160x120, "1" – 320x240, "2" – 640x480. Should be set before picture taken or entering Rover Mode.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: None	
Error: None	

camshoot\r	
Take a picture using the active camera. NOTE: jpeg image returned is in "Rover Mode" format with "jpeg:" prefix.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

load fileName.s\r	
Specify script file for execution. When successfully complete, the specified file name and first line number are returned.	
Passed Argument: fileName.s	Returned Data: "fileName.s #\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: Invalid File Name, File Name Not Found, Invalid Script File	

reset\r	
Reset the currently specified script file to restart execution. When successfully complete, the specified file name and first line number are returned.	
Passed Argument: none	Returned Data: "fileName.s #\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified for Execution	

stepin\r	
Single Step regardless of Start Rover Mode. After execution, the next line number is returned.	
Passed Argument: none	Returned Data: "fileName #\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified for Execution, Execution Stopped Due to Error, Execution Complete	

stepover\r	
Single Step except when function call encountered – then executes to statement after the function call unless Error encountered in Execution.	
Passed Argument: none	Returned Data: "fileName #\r" or "Error Message\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified for Execution, Execution Stopped Due to Error, Execution Complete	

stepout\r	
Single Step unless inside a function at which point execute to the statement past the function's "return" statement. After execution, the next line number is returned unless Error encountered in Execution.	
Passed Argument: none	Returned Data: "fileName #\r" or "Error Message"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified for Execution, Execution Stopped Due to Error, Execution Complete	

resume\r	
Start Script Execution. The "status\r" command is used to poll the state of execution.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified for Execution, Execution Stopped Due to Error, Execution Complete	

vardump\r	
Start listing all the active variables and their values in specified script. Data comes as "variableName[arrayIndex] =\r" with the data and follow in variables returned by the "vardumpnxt\r" command. The "arrayIndex" is only provide for array variables.	
Passed Argument: none	Returned Data: "variableName ... =\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified	

vardumpnxt\r	
Continue listing all active variables and their values in the specified script. Return string alternates between variableName, arrayIndex (if appropriate) and values. Operation continues until there are no more variables and data to display (at which point just "\r" is returned) or a "halt\r" command is received.	
Passed Argument: none	Returned Data: "variableName ... = \r" or "\r".
Execution Restrictions: Not Available in Rover Mode or when script executing. MUST be after "vardump\r" or "vardumpnxt\r" command.	
Error: None	

stackdump\r	
Start listing the contents of the variable and RPN stacks. Data comes as “element =\r” with the data and follow in variables returned by the “stackdump\r” command.	
Passed Argument: none	Returned Data: “element ... =\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: No Script File Specified	

stackdumpn\r	
Continue listing the variable and RPN stack data. Return string alternates between element and values. Operation continues until there are no more stack elements to display (at which point just “\r” is returned) or a “halt\r” command is received.	
Passed Argument: none	Returned Data: “element ... = \r” or “\r”.
Execution Restrictions: Not Available in Rover Mode or when script executing. MUST be after “stackdump\r” or “stackdumpn\r” command.	
Error: None	

breakset fileName.s:#\r	
Set a breakpoint in the specified script file at the specified line number. NOTE: The fileName.s does NOT have to be loaded. NOTE: If there are multiple statements on the same line, then execution will stop at each of these statements.	
Passed Argument: fileName, #	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

breakclear\r	
Remove ALL currently stored breakpoints.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

breakclear fileName.s:#\r	
Turn off the breakpoint in the specified script file at the specified line number.	
Passed Argument: fileName, #	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

breaklist\r	
Start listing the breakpoints specified by the user.	
Passed Argument: none	Returned Data: “fileName.s:#\r”
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

breaknext\r	
Continue listing the breakpoints. Operation continues until there are no more breakpoints to display (at which point just “\r” is returned) or a “halt\r” command is received.	
Passed Argument: none	Returned Data: “fileName.s:#\r” or “\r”.

Execution Restrictions: Not Available in Rover Mode or when script executing
Error: None

writevar variableName:newValue\r	
Change the contents of a non-array variable. NOTE: The maximum command length is 63 bytes, if the command, variable name and new value is longer than 63 bytes, the data may not be saved correctly or an error is returned.	
Passed Argument: variableName:#	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode	
Error: Variable Not Found, Array Variable, String Data Too Long	

expsendrx Device:Register:sendLength:receiveLength:## ## ... ##\r	
Send I2C command to Expansion Connectors. “Device” is the device’s I2C address in hex. “Register” is the device’s register address in hex. “sendLength” is the number of bytes to send in decimal. “receiveLength” is the number of expected bytes in reply in decimal. The message follows as hex bytes. Data returned is in hex format.	
Passed Argument: ##:##:##:##:Data	Returned Data: “\r”
Execution Restrictions: Not Available in Rover Mode	
Error: Invalid Data Passed with Command	

exp1pwr on\r	
Turn on 3.3V to Expansion Port 1.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

exp1pwr off\r	
Turn off 3.3V to Expansion Port 1.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

exp2pwr on\r	
Turn on 3.3V to Expansion Port 2.	
Passed Argument: none	Returned Data: “\r”
Execution Restrictions: None	
Error: None	

exp2pwr off\r	
Turn off 3.3V to Expansion Port 2.	
Passed Argument: none	Returned Data: “\r”

Execution Restrictions: None
Error: None

exp1getpwr\r	
Return power state of Expansion Port 1. "0" off, "1" on.	
Passed Argument: none	Returned Data: "#\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

exp2getpwr\r	
Return power state of Expansion Port 2. "0" off, "1" on.	
Passed Argument: none	Returned Data: "#\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

roverstart\r	
Start Rover Mode. This mode and its commands are described in detail in the 'Jade Robot "Rover Mode" Command section.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode or when script executing	
Error: None	

roverstop\r	
End Rover Mode. This mode and its commands are described in detail in the 'Jade Robot "Rover Mode" Command section.	
Passed Argument: none	Returned Data: "\r"
Execution Restrictions: Not Available in Rover Mode	
Error: None	

Jade Robot “Rover Mode” Commands

“Rover Mode” allows the Jade Robot, with a C329 camera attached, to act in remote control operation with the robot continually sending camera images to the host system while the host system sends movement, servo and sensor requests to the Jade Robot. This mode is separate from the normal “command” mode of the Jade Robot.

Rover commands are sent to the Jade Robot terminated with a “Carriage Return” (0x0D or “\r”). Data can be sent to the Jade Robot at any time; the commands are stored within the Jade robot and handled in a serialized manner.

Data returning from the Jade Robot in Rover Mode will consist of “DataPackets” which will have prefix and size information and be in the format:

```
Prefix (4 chars) + ":" + 3 bytes PacketData.length (LSB first) + PacketData (ending in 0xFF/0xD9)
```

Rover Mode DataPackets **ALWAYS** end with 0xFF, 0xD9. The PacketData.length plus 8 will always be the size of the incoming packet except in the case of jpegs, in which the PacketData.length plus 8 will be within 32 bytes which is due to the data being sent from the camera to the Jade Robot in 16 byte blocks which maybe ended prematurely due to the appearance of the 0xFF/0xD9 ending sequence.

The following DataPacket prefixes will be sent from the jade Robot:

- a) “jpeg:” – jpeg image being follows (Automatically sent in Rover Mode)
- b) “objd:” – Object Detection Data follows (Automatically sent in Rover Mode)
- c) “spec:” – Spectrometer Data follows (Sent in response to “spectro\r” Rover Mode Command)
- d) “lite:” – Light sensor Data follows (Automatically sent in Rover Mode)
- e) “batt:” – Battery Level Data follows (Automatically sent in Rover Mode)
- f) “eror:” – Error Message Follows (Response to an invalid command (see below))
- g) “stat:” – Rover Status Follows (Response to “roverstat\r” Rover Mode Command)
- h) “term:” – Rover Mode is Ended (received a “roverstop\r” or “halt\r” Rover Mode Command)

DataPackets are sent continuously from the Jade Robot with **NO** overlapping or embedded DataPackets.

To put the Jade Robot into Rover Mode, first the robot must be connected via Bluetooth to a host PC, secondly camera must be turned on (“camon\r”) and, optionally, the camera resolution (“camres #\r”) is set, before sending the “roverstart\r” command.

Commands

Except for the following Rover Mode Commands, all commands sent to the Jade Robot while it is in Rover Mode will result in a “error:” DataPacket Reply:

roverstart\r	
Start Rover Mode. NOTE: script cannot be executing. NOTE: Camera must be turned on. Following the Carriage Return (0x0D or “\r”) from this command, the first jpeg DataPacket will start to be sent to the PC.	
Passed Argument: value	Returned Data: “\r”
Error: None	

roverstop\r or halt\r	
End Rover Mode. Will return a “term:” DataPacket when Rover Mode has completed.	
Passed Argument: None	Returned Data: “term:” + 0x02 + 0x00 + 0x00 + 0xFF + 0xD9
Error: None	

roverstat\r	
Request the current Rover Mode Status. Will ONLY return a “stat:” DataPacket. See below in “Special Cases”.	
Passed Argument: None	Returned Data: “stat:” + DataPacket.length + DataPacket + 0xFF + 0xD9
Error: None	

spectro\r	
Perform a Spectrometer Reading. Will return a “spec:” DataPacket.	
Passed Argument: None	Returned Data: “spec:” + DataPacket.length + DataPacket + 0xFF + 0xD9
Error: None	

btledon #\r	
Turn on the Specified Green LED (0 – 5) on top of Jade Robot. No Response from this command, User is to assume it completed successfully.	
Passed Argument: “0”-“5”	Returned Data: None
Error: None	

btledoff #\r	
Turn off the Specified Green LED (0 – 5) on top of Jade Robot. No Response from this command, User is to assume it completed successfully.	
Passed Argument: “0”-“5”	Returned Data: None
Error: None	

setleftmtr #\r	
Send a value to the left motor (from -100 (full reverse) to 0 (Stop) to 100 (full forwards)). Motor will automatically stop after 100ms. No Response from this command, User is to assume it completed successfully.	
Passed Argument: value	Returned Data: None
Error: None	

setritemtr #\r

Send a value to the right motor (from -100 (full reverse) to 0 (Stop) to 100 (full forwards)). Motor will automatically stop after 100ms. No Response from this command, User is to assume it completed successfully.

Passed Argument: value	Returned Data: None
-------------------------------	----------------------------

Error: None

servoon\r

Turn on the Servo (Gripper and Elevation) Power Supply and enable PWM signals sent to servos. Initial Servo positions will be "50" (middle). No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servooff\r

Turn off the Servo (Gripper and Elevation) Power Supply and disable PWM signals sent to servos. Initial Servo positions will be "50" (middle). No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servocntr\r

Centre the Servos (Gripper and Elevation to "50") and enable Power Supply and PWM signals if they are off. No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servoclose\r

Close the Gripper Servo by a value of "10" until the servo is fully closed ("0"). No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servoopen\r

Open the Gripper Servo by a value of "10" until the servo is fully opened ("100"). No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servoup\r

Raise the Gripper by a value of "10" until the Gripper is fully up ("100"). No Response from this command, User is to assume it completed successfully.

Passed Argument: None	Returned Data: None
------------------------------	----------------------------

Error: None

servodown\r

Lower the Gripper by a value of "10" until the Gripper is fully down ("0"). No Response from this command, User is to assume it completed successfully.

Passed Argument: None **Returned Data:** None

Error: None

gripset #\r

Set the Gripper position with a value of "0" (fully closed) to "100" (fully open). No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

eleaset #\r

Set the Gripper elevation with a value of "0" (fully down) to "100" (fully up). No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

camres #\r

Set the resolution of the camera – all following images returned in Rover Mode will be at the specified resolution: "0" – 160x120, "1" – 320x240, "2" – 640x480. No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

exp1pwron\r

Turn on the 3.3V LDO to "Expansion 1". No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

exp1pwroff\r

Turn off the 3.3V LDO to "Expansion 1". No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

exp2pwron\r

Turn on the 3.3V LDO to "Expansion 2". No Response from this command, User is to assume it completed successfully.

Passed Argument: value **Returned Data:** None

Error: None

exp2pwroff\r	
Turn off the 3.3V LDO to "Expansion 2". No Response from this command, User is to assume it completed successfully.	
Passed Argument: value	Returned Data: None
Error: None	

exp2pwroff\r	
Turn off the 3.3V LDO to "Expansion 2". No Response from this command, User is to assume it completed successfully.	
Passed Argument: value	Returned Data: None
Error: None	

Special Cases:

1. The Jade Command "camshoot\r" (Take Picture) returns the Rover Mode "jpeg:" DataPacket but is not to be considered a Rover Mode Command. This command will return an error if sent in Rover Mode and is meant for taking still pictures during normal operations.
2. The Rover Command "roverstat\r" **ALWAYS** returns a Rover Mode Packet (starting with "stat:" and ending with 0xFF, 0xD9). The expected payloads are:
 - "roveractive" – In rover mode, everything working fine.
 - "roverstop" – Not in Rover Mode. The Jade Robot should NOT drop out of Rover Mode spontaneously so the only case where this reply is received is if the PC software is in the incorrect mode.
 - "rovererror" – When there has been an unexpected failure in Rover Mode. Right now, the only expected case for this return is if the camera fails for some reason.

Notes:

1. An invalid command (for Jade Robot Operation or Rover Mode Command) will return an "eror:" DataPacket.

Supporting Documents

Jade Robot™ Support Software Installation and Introduction - TBD

Jade Robot™ Script Language Outline – available at <http://www.mimetics.ca/knowledge-base-2/>

Jade Robot™ Panel Driver Outline - TBD

Glossary

ASCII – Standard 8 bit character set. See <http://en.wikipedia.org/wiki/ASCII>

Document Updates

Date	Changes	Author
2014.06.02	Document Started	Myke Predko
2014.07.31	Updates including "Rover Mode" description	Myke Predko
2014.08.02	Expand definition of "Command mode"	Myke Predko
2014.08.03	Finalized description of various Commands	Myke Predko
2014.12.12	Added "proctype" as a console Command – Added in Release 42	Myke Predko