The Robot2CNC Protocol - v1.0.0

Last Updated: 6 November 2018

Overview

The Robot2CNC Protocol (v1.0) is designed to be a simple API level abstraction of CNC Primitives. While the terminology varies between cnc manufacturers, their functionality is more or less the same. This protocol provides a standard API to control a cnc, regardless of it's make. This should enable a robot automation expert to control various cnc's with little effort.

Versioning

This document is versioned in accordance to the Semantic Versioning 2.0.0 standard. Each update must be reflected in the version at the title of this file. Libraries targeting this protocol must label the exact version they are implement.

Network Channel

- Protocol: TCP/IP
- Port: 9002 (Default)
- Timeout: None
- Encoding: UTF-8

The Robot2CNC will listen for new connections on port 9002 by default on startup, after a TCP/IP close on an active connection or after a CLOSE command is completed. By default, there is no read timeout, a timeout value can and should be set by the client to prevent a rebooted client from holding on to the connection. The socket is bi-directional. In general, the Robot2CNC receives commands from the client and sends responses as events occur. Commands are queued up if the previous command has not completed. Queued commands and responses are guaranteed to be sent in the order received.

Robot2CNC Protocol Format

Commands and responses over the Robot2CNC sockets are semicolon terminated text strings. The parameters of commands and responses are delineated by commas. The first parameter is the action and the remaining parameters define the action. Capital letters are used in string parameters except where noted. Numeric parameters can be integer, decimal or hexadecimal. Hexadecimal values are only supported and preceded by ‘0x’. Whitespace, CR and LF are ignored in the protocol. All commands return a response. If a command returns a response, the response must start with the exact command it is responding to.

Example Command

- Command: `VERSION;`
- Response: `VERSION,1.0.0;`

Error Messages

Messages from the Robot2CNC to the client. These can represent various errors and events the Robot2CNC experiences. When a command is sent by the client that the Robot2CNC cannot complete for some reason, an `ERROR` response is sent back to the client.

ERROR

- Response: `ERROR,{ERROR_MESSAGE},{COMMAND},{COMMAND_PARAMETERS};`

Example Responses

```
ERROR,Invalid command,SCURRY;
```

```
ERROR,CNC Communication Error,CNC_STATUS;
```

```
ERROR,CNC Communication Error,SELECT_PROGRAM,81004;
```

General Commands

Version

- Command: `VERSION;`
- Response: `VERSION,{Protocol SemVer number};`

Get the version of this protocol the Robot2CNC is running.

Example Response

```
VERSION,1.0.0;
```

CNC Status

- Command: `CNC_STATUS;`
- Response: `CNC_STATUS,{STATUS};`

Status (depending on CNC make): {IDLE / RUNNING / COMPLETE / ALARM}

Trigger an status message from the Robot2CNC. Allows the client to get the value of the status on demand.

Back to top

Select Program

- Command: `SELECT_PROGRAM,{PROGRAM};`
- Response: `SELECT_PROGRAM,{PROGRAM};`

Select the given program on the CNC.

Back to top

Run Program

- Command: `RUN_PROGRAM,{PROGRAM};`
- Response: `RUN_PROGRAM,{PROGRAM};`

Select the given program on the CNC and then Cycle Start the CNC.

Back to top

Cycle Start

- Command: `CYCLE_START;`
- Response: `CYCLE_START;`

Cycle Start the CNC.

Back to top

Read Macro Variable

- Command: `READ_MACRO,{VARIABLE_ADDR};`
- Response: `READ_MACRO,{VARIABLE_ADDR},{VALUE};`

Read macro varible value from a given address.

Back to top

Write Macro Variable

- Command: `WRITE_MACRO,{VARIABLE_ADDR},{VALUE};`
- Response: `WRITE_MACRO,{VARIABLE_ADDR},{VALUE};`

Write macro varible value base on a given address.

Back to top

Get IO Value

- Command: `GET_IO,{IO_Address}`
- Command: `GET_IO,{IO_Address},{IO_Value}`

Get the value of a given IO address.

Back to top

Set IO Value

- Command: `SET_IO,{IO_Address},{IO_Value}`
- Command: `SET_IO,{IO_Address},{IO_Value}`

Set the value of a given IO address to the passed value.

Back to top