HandE communication protocol

This document describes the communication protocol between the HandE host and device. Here host is a PC running Windows OS and device is the instrument (embedded unit) running HandE firmware.

The host and device connected using serial(UART) interface with a baudrate of 115200.

All the messages exchanged between host and device are text messages. Each message is terminated by a newline character '\n'. All text fields in the message are seperated by a comma (','). The starting character of the message is '#' and end character is '*'. After end character there is a 4 hex digits giving the 16 bit CRC. This CRC is calculated for the characters present between '#' and* ''. CRC does not use these boundary characters.

1. Generic format of command

Following is the generic format of command messages that are sent form host to device.

```
#<seqNo>,<fCode>,<arg_1>,<arg_2>...<arg_n>*hhhh\n
```

The first field of message after '#' is 'sequence number' (seqNo). This sequence number is a decimal number string and starts from 0 for the first command message. Next it increments by one for each subsequent command message. After reaching 255, it restarts from 0.

The second field 'function code' (fCode) is a decimal number string identifying the command to execute.

The function code field is followed by zero or more number of command argument fields. Each argument could be either a decimal number string or hex number string or any text string.

2. Generic format of response

Following is the format of response messages that are sent from device to host.

```
#<seqNo>,<fCode>,<status>,<resp_1>,<resp_2> ... <resp_n>*hhhh\n
```

All most all response messages are responses generated by the device for the commands received from the host. However there is some exception. It is possible for a device to send some responses, whenever some unexpected events have occured. These are called notification messages.

The response messages are also classified into two types as intermediate responses and final responses.

The sequence number and function code of response messages should have same values of command message, for which it is a response.

Here status is 4 character string. First character is a letter and remaining three chracters are decimal digits. Following are valid first character for status string.

```
  I : Intermediate response status
  F : Final response status
  N : Notification message number
```

The character 'I' indicates that this response is an intermediate response and final response is pending. This is because some commands take more time to execute. For such commands as soon as command execution is started, intermediate response is sent first. Once command execution is completed final response is sent.

Following are the valid three digit decimal numbers for I and F response status.

```
000 : Command execution success
001 : Command accepted, pending execution
002 : Unknown command
003 : Insufficient arguments
004 : Invalid argument(s)
005 : Robo ARM is busy
006 : Path may cause collision of ARMS
007 : Command under development
008 : Fatal error
009 : Motor is busy
010 : Motor not rotating //When commanded to stop
011 : Heater already off //When commanded to off
012 : Heater already on  //When commanded to on
```

The character 'N' indicates that this message is not a response to any command. It is independent nofification messages. The three digit decimal number gives the notification ID or number. Both the 'seqNo' and 'fCode' fields of notification message are set to zero. Note that zero is not a valid function code..

Following are the valid three digit decimal numbers for notification messages.

```
001 : Device rebooted event
002 : Loading/unloading door opened/closed
```

Status field is followed by a zero or more number of response or notification data items.

3. Description of commands

Following are the list commands supported by the current version of software.

1. RA move

This is a command to move only the robotic ARM.

```
  #seqNo,fCode,ra_no,x,y,z*hhhh\n
```

```
  #seqNo,fCode,status,ra_no*hhhh\n
```

2. RA pick

```
  #seqNo,fCode,ra_no,x,y,z,isHeater*hhhh\n
```

```
  #seqNo,fCode,status,ra_no*hhhh\n
```

3. RA place

```
  #seqNo,fCode,ra_no,x,y,z,isHeater*hhhh\n
```

```
  #seqNo,fCode,status,ra_no*hhhh\n
```

4. Level sensing with RA

Level sensing command for liquide level sensing in all jaars

```
#seqNo,fCode,x,y,z,isHeater*hhhh\n
```

```
#seqNo,fCode,status,level*hhhh\n
```

5. Homing command

```
#seqNo,fCode,motor_bit_string*hhhh\n
```

6. Get home status

```
#seqNo,fCode*hhhh\n
```

7. Rack holding status

```
#seqNo,fCode*hhhh\n
```

#seqNo,fCode,status,rack_state_bit_string*hhhh\n

8. Door control

```
#seqNo,fCode, door_type, operation*hhhh\n
```

door_type : 1 for heater; Other reserved for future use operation : 0 Close 1 Open

9. Door status

```
#seqNo,fCode*hhhh\n
```

Status of heater door and loading/unloading door

```
#seqNo,fCode,status,door_state_bit_string *hhhh\n
```

10. Set temperature

```
#seqNo,fCode, temperature_deg_celsius*hhhh\n
```

11. Get temperature

```
#seqNo,fCode*hhhh\n
```

The response contains the temperature in degrees celsius.

```
#seqNo,fCode, status, temperature_deg_celsius*hhhh\n
```

12. Heater control

```
#seqNo,fCode, operation*hhhh\n
```

operation: 1 ON 0 OFF

13. Valve control

This command is for controlling the water inlet valve

```
#seqNo,fCode, valve_type, operation*hhhh\n

valve_no  : 1 Water Inlet valve
operation : 1 ON
            0 OFF
```

14. Valve status

The purpose of this command is to find the on/off status of all water valves like water inlet and outlet valves.

```
#seqNo,fCode*hhhh\n

#seqNo,fCode,status,valve_state_bit_string*hhhh\n
```

Here bit string is two character string representing two bits. Each character is either '0' or '1'. The LSB character (right side) represent the state of inlet valve and MSB character (left side) represent the state of outlet valve.

15. Poll/version command,

This command is for just checking that HandE device is responding to the host commands are not. Possibly this could be the first command that can be used by the host to ensure that device is present and responding.

```
#seqNo,fCode*hhhh\n
```

The response contains status field as "F000" and one response data field with version string.

```
#seqNo,fCode,status,version_string*hhhh\n
```

16. Agitation control

This command is used start and stop the agitation movement. When agitation movement is stopped, the device ensures that, stop position is home position.

```
#seqNo,fCode,<operation>*hhhh\n

operation: 1 START
           0 STOP
```

17. RA multiple dip

This command is for dipping the rack in a jar for one or more times. In the following format 'dip_count' is the number of dips to perform. The 'dip_delay' is the time in seconds, the rack is dipped in the jar. The 'dry_delay' is the number of seconds to keep the rack for drying above the jar.

```
#seqNo,fCode,ra_no,x,y,dip_count,dip_delay,dry_delay*hhhh\n
```

The response to the above command may have both intermediate response and final response.

```
#seqNo,fCode,status,ra_no*hhhh\n
```