National Centre for Radio Astrophysics

Tata Institute of Fundamental Research,

Pune University Campus, Pune, INDIA

Internal Technical Report
GMRT/R-268

Technical Report
on
Enabling Serial communication in Online_V2: Rabbit MCM to the FPS

Charudatta Kanade

GMRT – Khodad

Email : cpk@gmrt.ncra.tifr.res.in

Author : Charudatta Kanade	Date of issue : 19th May 2015	Scope : Testing of FPS system over Serial Interface under Online_V2 CMS.
Verified by : Dr. Nimisha Kantharia		
Approved by : Dr. Nimisha Kantharia	Status/ Version : 1	Internal Technical Report No.: GMRT/R-268

INDEX

Abstract :

The Giant Meter wave Radio Telescope (GMRT) is being upgraded to improve the overall performance of telescope along with the modern servo systems for antennas and feeds. To control and coordinate the newly upgraded GMRT systems for performing astronomical observations, the new GMRT Monitor and Control (M&C) system is being developed. The new M&C system has a modern hardware and software architectural features as compared to the existing GMRT control system. The M&C upgrade will speed up the antenna communication by exploiting the power of the upgraded Ethernet over the OF. This along with the upgraded servo PC104 would result in reduced overhead times for several antenna control operations. The high end server class machines in the Central Electronics Building (CEB) will be used to run the central supervisory M&C system, which communicates to all thirty antennas spread over a radial distance about ~ 15 km using a dedicated Ethernet/Optical-Fiber link at 1 Gbps.

The existing GMRT sub systems are controlled by using RS485 serial interface at 9600 bps rate, half duplex mode and multi drop configuration. The Rabbit MCM card supports RS485 connectivity to provide backward compatibility for controlling existing hardware as well as upgraded GMRT sub systems, if necessary. The SPI interface (Serial Peripheral Interface) is provided for control of FSW unit in GMRT Analog Backend system. The rest of upgraded GMRT sub systems including Servo, FPS, fiber optics, Sentinel, GAB can be controlled over Ethernet uses the recently developed In-House M&C system Online_V2.

Background

The block diagram of existing M&C system is given below.

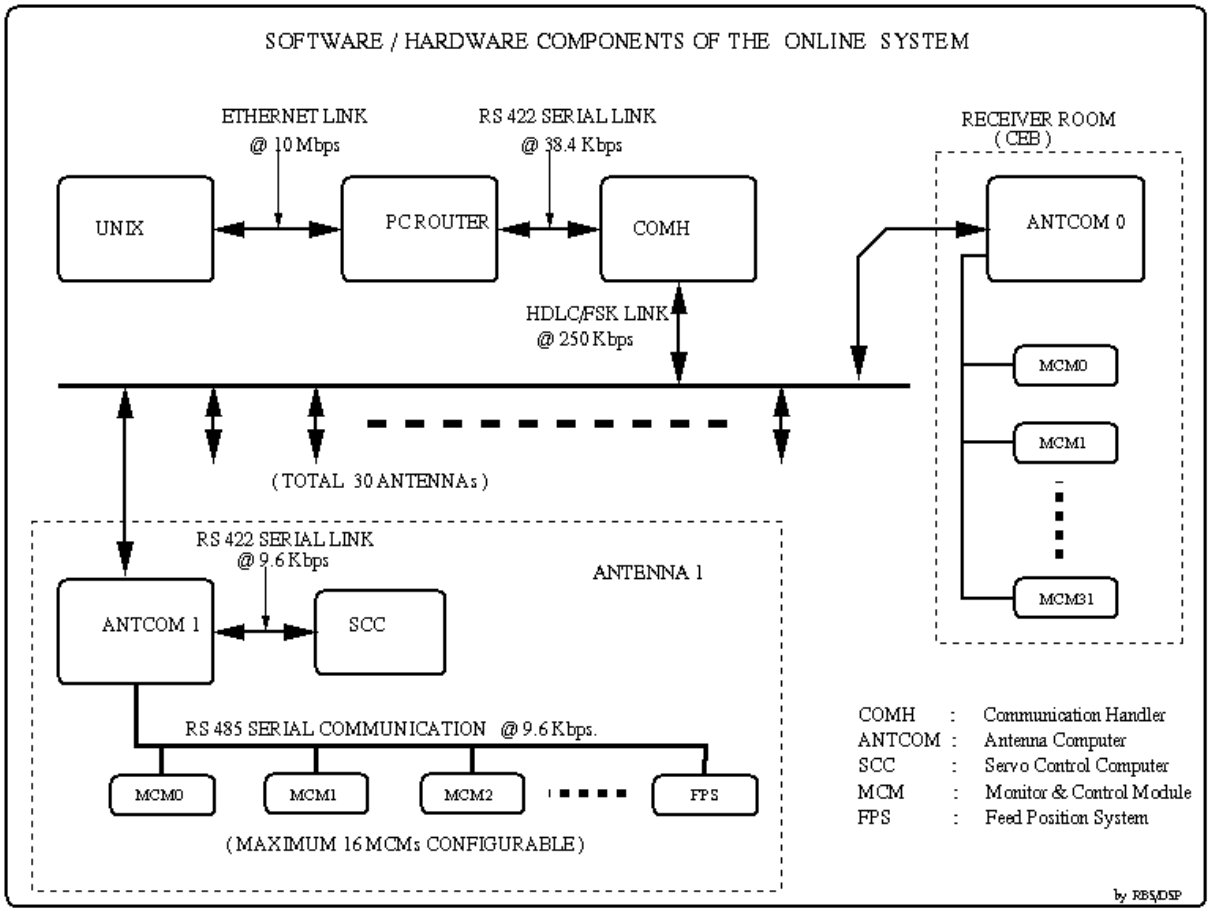

Figure #1. Block Diagram of existing M&C system.

In existing ONLINE system there is chain of different hardware components and different technologies are being used to exchange data with GMRT sub systems. These include 10 Mbps Ethernet link in CEB, HDLC/FSK link @250 Kbps from CEB to ABC, RS422 serial link @9.6 Kbps for controlling servo and RS485 serial link @ 9.6 Kbps for controlling MCM at antenna base through Antenna Base Computer (ANTCOM or ABCCOM).

The ANTCOM talks to MCM 0, 2, 3, 5, 10 and 14 over RS485, half duplex, multi drop point @9.6 Kbps communication link. The MCM 0 is used for monitoring Sentinel and telemetry system and recently being used for providing attenuation to RF signal in Fiber Optics system at antenna base. The MCM 2, 3 and 10 are used for Antenna base receiver system. The Front End system is controlled by MCM 5 through MCM 2. The Feed Positioning System is controlled by MCM 14. The existing MCM cards do not have Ethernet port and SPI ports for talking to latest hardware.

The upgrade of GMRT hardware and Software is going on in a phased manner. Some systems are fully upgraded and some are in different phases of completion. The Rabbit MCM cards are ready for taking care of all GMRT control and monitoring requirements.

The new Monitor and Control Modules (MCM) developed based on Rabbit RCM 4300 micro-controller are used to tune the RF signal receiver chain systems like Front-end, signal conditioning and Analog-backend. The Rabbit MCM card supports 1 GB mini SD memory card, 10/100T base Ethernet port for the communication along with a configurable 32 TTL control outputs and 64 analog monitoring channels. Dynamic C Integrated development environment support provided on Rabbit processor is used to develop and run the embedded control software. This software handles low level M&C functionalities like implementation of control logic, monitoring interpretation and safety of the instrument. For more information refer to the report by Mr. Naresh Sisodiya.

Similarly, servo system of antenna and Feed-Positioning System plan to be controlled by PC104 embedded computer with lightweight Puppy Linux OS. The M&C system will thus coordinate and remotely control all thirty antennas and a total of 180 sub-systems using the IP based interface.

Now, in upgraded GMRT hardware RF signals are coming directly to the Central Electronics Building (CEB), over broadband Optics Fiber system. The Rabbit MCMs are deployed for controlling and monitoring of upgraded GAB system in Receiver room, over the Ethernet link.

The Front End system which sits at the focus of the antenna has few practical challenges for adopting right kind of physical connection medium. The physical connection medium could be Fiber Optics link or shielded CAT 5 cable or serial connectivity from base of the antenna to FE System. Ideally, like any other upgraded GMRT subsystem, FE system would be controlled over Ethernet but in case of any problem the Rabbit MCM card will supports serial link as a fallback option.

This report describes efforts taken to control the existing FPS under Online_V2 M&C using combination of Ethernet and Serial connectivity.

The Overview of GMRT FPS system

The GMRT operates in five frequency bands centered at 150 MHz, 233 MHz, 325 MHz, 610 MHz and 1420 MHz and the feeds are mounted on the four side of a turret. The 233 MHz and 610 MHz feeds are concentric and mounted on the same side, thus five feeds are accommodated. The feed Positioning System (FPS) is used to precisely focus the selected feed before start of observation. The feed positioning system needs to be calibrated before use by moving feed from current position to 270 degrees which acts as the reference position for FPS.

The feed drive circuit consists of a Half HP DC servo motor with inbuilt brakes. The power MOSFETS are used in 'H' configuration to enable bidirectional rotation. The speed of the motor is controlled by using Pulse Width Modulation technique. The appropriate software produces the trapezoidal speed profile for achieving high accuracy of 1.054 arc min.

The feed rotation is restricted to 275 degree as feed cable wrap hose carries the front end, RF Control and Monitor cables. The End limit switches are provided at -15 deg. and 285 deg. along with stoppers for ensuring safety from over travel of feed turret. The Non Volatile RAM (IC DS1220Y) is provided to retain current position in case of power failure and thereby avoiding frequent recalibration.

The MCM 14(FPS MCM) facilitates communication with ANTCOM PIU and uses RS485 serial link for exchanging data with existing ONLINE system.

The feed drive motor is coupled to the feed turret through two stages of gear reduction, first is a worm gear with 300:1 reduction and second one is primary bull gear with 5:1 reduction. Thus total gear reduction is 1500:1. The incremental encoder is used for marking the position. This means that for one complete rotation of the turret, the encoder rotates for 5 times. The encoder gives the direction and count signals which acts as a position feedback to the FPS CPU card, an 8051 microcontroller based card. The CPU card generates signals like direction, RGSO and PWM to the drive card to control operations of motor and set the feed with an accuracy of one arc minute. These encoder pulses are counted by the microcontroller to determine the angular displacement of the feed system. More details on FPS can be obtained from GMRT FPS Group.

The MCM 14 card always send back TWO logical packets in the response for any fired command from ONLINE system, including NULL command, which is different from other MCM cards used in existing GMRT setup.

In MTAC-April 2015 at C06 Antenna, the FPS was receiving commands from the CEB via a Rabbit MCM at the antenna base. The other Rabbit card was used to control and monitor the OF system and the sentinel system (both system from same Rabbit MCM card). In the final configuration when the old FPS MCM is upgraded to an Ethernet based card, it is envisaged that the commands from the CEB will directly control the new FPS.

Implementation:

The testing of Feed Positioning System using Rabbit MCM card.

The Online_V2 was used to control the FPS system at C06 antenna base with help of Rabbit MCM card and existing MCM 14 (FPS MCM). The role of ANTCOM, talking to ONLINE system as well as existing MCM card, was taken over by Rabbit MCM card. The Online_V2 was talking to Rabbit MCM over 100 Mbps Ethernet link and Rabbit MCM was communicating with MCM 14 card over RS485 serial link. The CISCO layer 2 Ethernet switch, SG300-28, was set up at antenna base to provide Ethernet connectivity to two Rabbit MCM cards. The other Rabbit MCM card was set up to control and monitor Fiber Optics system and Sentinel system.

The high level commands for FPS system was issued from Online_V2 machine (Teleset PC) GUI as well as command Terminal. The main high level commands for FPS system are fpsnull, Set commands, Read commands, Run command, Reboot command and Stop command. The text string command was sent to antenna based Rabbit MCM card over Ethernet in structure based format. The firmware for FPS Rabbit

MCM card converts text string commands into a hex code which was sent to existing MCM 14 over serial link. Then MCM 14 executes the command as if it coming from ANTCOM and sends back the response packet to the FPS Rabbit MCM over serial link in hex code format. The FPS Rabbit MCM firmware decodes the response packet and converts data into structure based format before sending it to Online_V2 over Ethernet.

So in short, this involves following steps and the hardware architecture is shown in figure 2.

(a) Sending command from OnlineV2 to Rabbit MCM card over Ethernet.

(b) Rabbit MCM card converts command to hex format and sends the command to FPS MCM over serial interface.

(c) FPS MCM card sends a response to Rabbit MCM over serial interface.

(d) Rabbit MCM which in turn converts response to the structure required by OnlineV2 and sends it across.

Hardware Architecture:

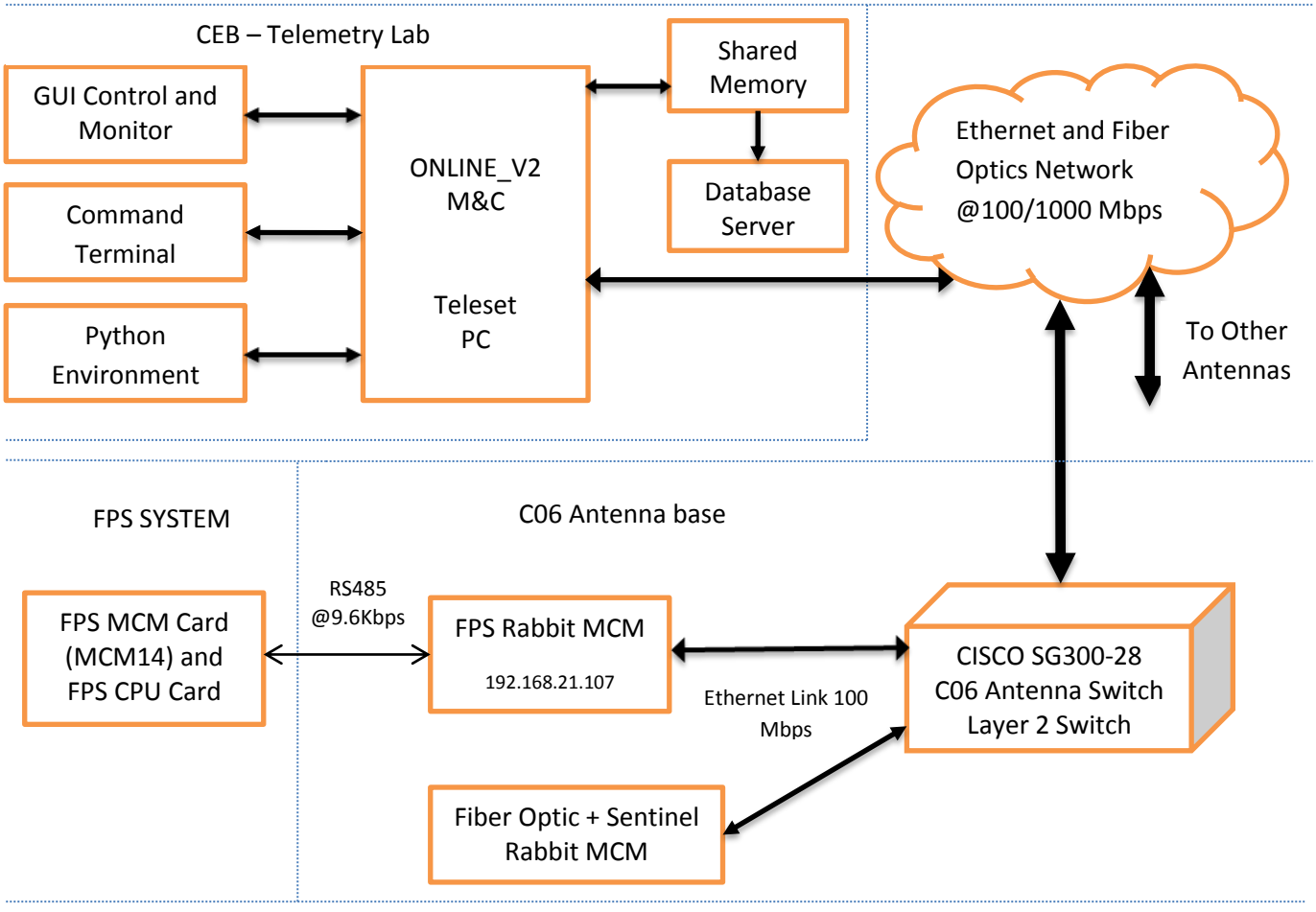

Figure #2 shows hardware architecture used for testing of FPS under Online_V2 M&S.

As show in the block diagram in figure 2, the Online_V2 PC is located in Telemetry Lab and the basic L3 Ethernet switch and broadband Fiber Optics system is in receiver room of Central Electronics Building (CEB). The Broadband fiber optics system supports 1Gbps connectivity to all 30 antennas. At antenna base, a layer 2 Ethernet switch, CISCO SG300-28, is installed to provide Ethernet connectivity to two Rabbit MCM cards. The FPS system Rabbit MCM card's IP address was set to 192.168.21.107. The Online_V2 machine IP was set to 192.168.8.45 for the testing. This test was carried out on 20th March 2015.

The high level command, a text string command, is passed to targeted Rabbit MCM card over Ethernet and Fiber optics network. The Rabbit MCM then decode and send command to FPS MCM (MCM 14) over RS485 serial link. After executing command MCM 14 send response packet to the Rabbit MCM over RS485 serial link which in turn send high level response packet to Online_V2. While MCM14 and Online_V2 were exchanging data between them other MCM cards, MCM 2, 3, 5, and 10 were working as usual. The response packet received by the Online_V2 was displayed on the Online_V2 GUI, python environment and terminal environment. The details of FPS test results are attached to this report.

Software architecture:

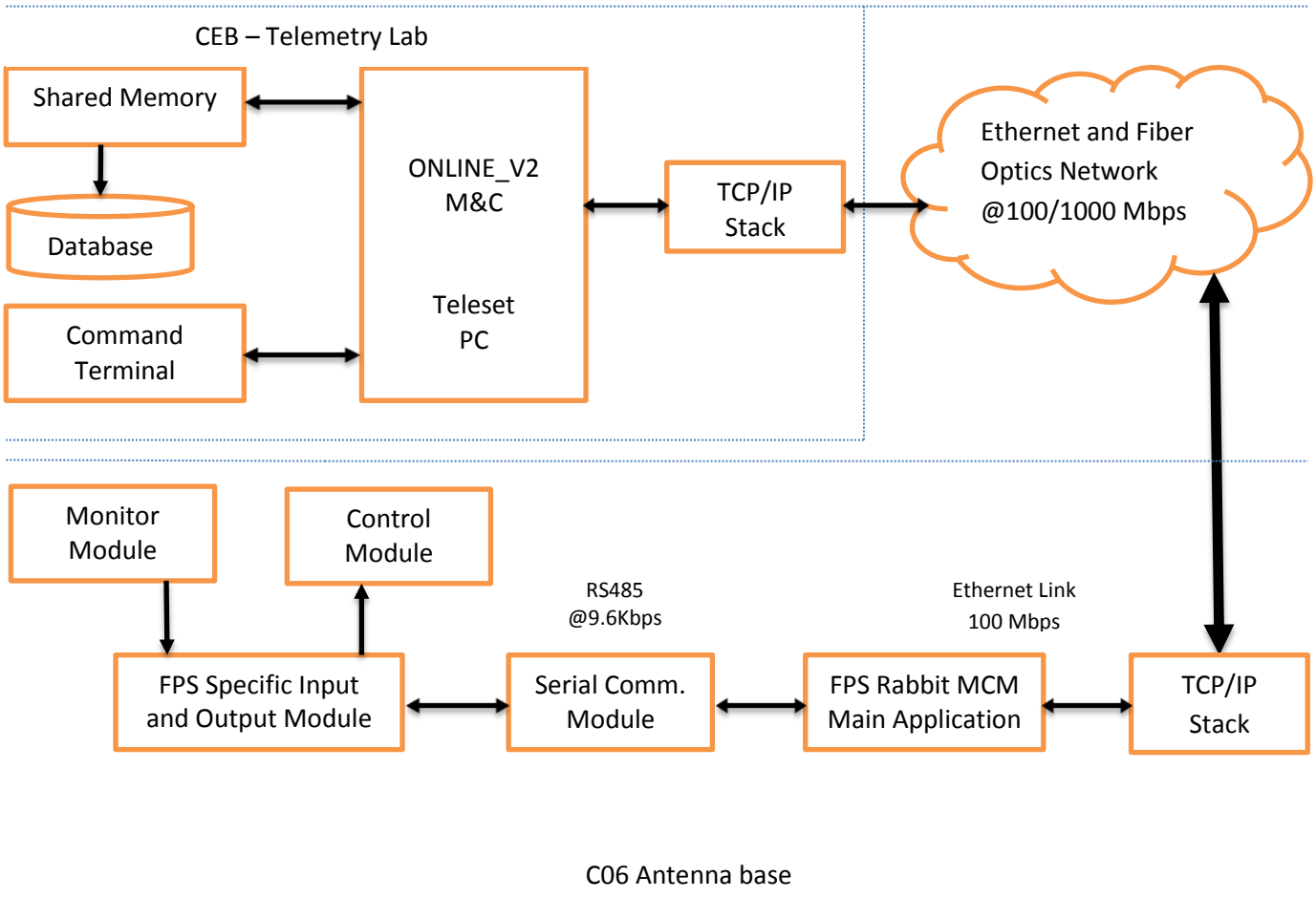

Figure #3 shows software architecture of test setup for testing FPS system under Online_V2 M&C.

The details of communication protocol and high commands used by Online_V2 for GMRT subsystems are available in technical report “Online_v2 core software & Testing done” written by Mr. Raju Uprade.

The Online_V2 uses structure based communication protocol to exchange data with Rabbit MCM cards. The command structure has details of system, command, sub command, parameters and other details but all in alphanumeric string. The command is send over socket to the Rabbit MCM firmware. The firmware decodes the high level command and generates hex codes understood by MCM 14 card. The Rabbit MCM card (or ANTCOM) and MCM 14 uses fixed packet communication protocol for exchanging data over serial link.

The detail of communication protocol used by existing ONLINE system and existing MCM card over serial link is as follows.

The communication protocol used for sending Command packet to FPS system is as follows.

0th byte : MCM Address
1st byte : command Packet Length lower byte
2nd byte : command Packet Length higher byte
3rd byte : Id_Code
4th byte : Main Command lower byte : Null , Set, Read ,Run, Reboot, Stop
5th byte : Main Command higher byte
6th byte : Argument length lower byte
7th byte: Argument length higher byte
8th byte : Sub command of main command
9th byte : Actual argument/s , if any
10th byte : Checksum.

The command packet length can vary depending on actual arguments in the command.
The communication protocol used for Response packet from MCM 14 to existing ONLINE system is as follows.

0th byte : MCM Address
1st byte : Response Packet Length lower byte
2nd byte : Response Packet Length higher byte
3rd byte : Id_Code
4th byte : Command Status
5th byte : Number of Logical packets.
6th byte : First Logical Packet length lower byte
7th byte : First Logical Packet length higher byte
8th byte: Code for Self test
9th byte: Status byte
10th byte : Encoder lower byte
11th byte : Encoder higher byte
12th byte: RPM byte
13th byte : Second Logical Packet length lower byte
14th byte : Second Logical Packet length higher byte
15th byte : Main command to which this response is send by MCM14.
16th byte: Arguments/parameters of main command, if any
17th byte : Checksum.

In a response packet, the first logical packet (from 5th byte to 12th byte) gives position and motor RPM feedback. The second logical packet relates command issued from ONLINE to MCM14.

The command structure and response structure used by Online_V2 and Rabbit MCM for exchanging data is as follows.

Command Structure used sending command from Online_V2 to Rabbit MCM.

```
typedef struct
{
int seq; // Unique Sequence
char timestamp[64]; // Timestamp of command
char system_name[16]; // System Name for which command is
char op_name[16]; // Operation to perform ( Init/Set/Mon/Reset )
short int number_param; // Number of parameter
char parameter_name[32][16]; // Parameter Name
char Argument_Ch1[32][16]; // Channel One argument
char Argument_Ch2[32][16]; // Channel Two argument
} cmd;
```

Response Structure used for sending response from Rabbit MCM to Online_V2.

```
typedef struct
{
int response_type; // Response type
int seq; // Sequence number
char timestamp[64]; // Time stamp
char system_name[16]; // System name
char Mon_raw[64][8]; // 64 channel raw data
char Mon_sum[32][64]; // Monitoring summary prepared from 64 channel raw data
short int num_resp_msg; // Number of Response Message
char response_message[32][64]; // Response message from MCM
} resp;
```

Here is the example of fpsnull command which was fired from Online_V2 to demonstrate how communication protocols are used for exchanging data.

The fpsnull Command is a general purpose command used to communicate with the FPS hardware. The response packet for the fpsnull command sends back the current position of feed and motor RPM.

The First step is to send high level command “fpsnull” from Online_V2 to the Rabbit MCM
High level command: fpsnull

The Second step is to convert high level command into hex format which is recognized by FPS MCM 14. The hex format for fpsnull command send by Rabbit MCM to FPS system over serial link is as follows.

0E 08 00 00 00 00 00 00 EA

The Third Step is to send back the response packet over serial link to Rabbit MCM after executing command by FPS MCM. The response packet send by FPS MCM is as follows.

0E OD 00 00 00 01 07 00 04 40 4D 6F 00 CB

The fourth step is to convert response packet to the structure required by Online_V2 and send it across over Ethernet to Teleset PC. The high level response on Teleset PC looks as follows. This shows that feed is calibrated and idle. The motor is off.

High level response:
Exec. OK
Feed Calibrated and Idle
EncCount = 17000
Rpm = 0

The following FPS Commands were tested at C06 antenna from Online_V2 M&C.

All FPS commands listed below have been ported for execution from Online_V2 .

Null Command This command is a general purpose command used to communicate with the FPS hardware. The response packet for the NULL command has logical packet which send back the current position of feed and motor RPM.

High level command: fpsnull

High level response:
Exec. OK
Feed Calibrated and Idle
EncCount = 17000
Rpm = 0

FPS Set Commands:

Set Turning Point Turning point difference is the difference between the current position and target position at which the rpm of the feed starts ramping down.

High level command: set_tpoint
Enter turning point position difference: 200

High level response:
Exec. OK
Set Turning Point, target: 200

Set Ramp Down Time Count: Once the turning point is reached, motor rpm has to be reduced with a uniform rate. This rate can be specified by issuing this command .The motor rpm is controlled by pwm register of 8051.The Ramp Down Time Count is nothing but the time difference in milliseconds after which the count in pwm register is incremented by one to decrement the duty cycle and hence the rpm.

High level command: set_rampdcnt
Enter ramp down time count (msec) : 20

High level response:
Exec. OK
Set Ramp down count, Slope : 20

Set Lower RPM Limit: This command is used to set the lower limit of the motor rpm when it is ramping down. Once the pulse count lower rpm limit reached, further decrement in rpm is stopped.

High level command: set_low_rpm
Enter Lower RPM limit: 630
Enter Check-Interval (ms): 20
High level response:
Exec. OK
Set Lower Ramp Limit, 0 int 20

Set Brake Count Difference: Brake count difference is the difference between current angle and the target angle at which brakes should be applied.

High level command: set_Brake_dd
Enter Break Cnt difference: 6

High level response:
Exec. OK
Set Break Count Diff, 6

Set Ramp Up Time count: As feed starts rotating ,it is not wise to increase the rpm to its maximum allowed value .This transition in the count can be made smooth by decrementing count in the pwm register of 8051 microcontroller at regular interval. Set ramp up time count command is used to specify this time.

High level command: set_rampupcnt
Enter ramp up time count (msec) : 20

High level response:
Exec. OK
Set Ramp up count, Slope : 20

Set Stop Time Count: This is the rate at which the pwm register count is incremented by one when a stop command is issued by user.

High level command: set_stoptimecnt
Enter stop time count (msec) : 10

High level response:
Exec. OK
Set Stop time count : 10

Set Max PWM Count: This command controls the maximum rpm of motor that can be achieved in any run command. Argument to this command is a byte that is nothing but the minimum value in hex in the PWM register.

High level command: set_Max_pwm_cnt

Enter max PWM cnt:50
High level response:
Exec. OK
Set Max PWM Count, 32

Set Max Angle: As the feed starts moving towards 270 deg. limit switch angle count goes up.Upper limit on this angle is set up by this command so that software does not allow the feed to cross this limit.

High level command: set_Max_angle
Enter angle count: 17300

High level response:
Exec. OK
Set Max Angle, 17300

Set Minimum Angle: Similarly ,as feed moves toward -15 deg. limit switch angle count goes down.This command is used to set the lower limit on the minimum angle position that feed can attain.

High level command: set_min_angle
Enter angle count: 1450

High level response:
Exec. OK
Set Min Angle, 1450

FPS Read Commands.

The read commands give either default status value or latest value set by previous “set” command of a parameter.

Read Version:

High level command: read_version

High level response:

Exec. OK
Read Version: 8.5

Read Turning Point:

High level command: read_tpoint

High level response:
Exec. OK
Read Turning Point, target:200

Read Max Angle:

High level command: read_Max_angle

High level response:

Exec. OK
Read Max Angle, 17300

Read Minimum Angle:
High level command: read_Min_angle

High level response:

Exec. OK
Read Min Angle, 1450

Read Brake Count Difference

High level command: read_Brake_dd

High level response:

Exec. OK
Read Brake Count Diff, 4

Read Lower RPM Limit
High level command: read_low_rpm

High level response:

Exec. OK
Read Lower Ramp Limit, 649 int 325

Read Ramp Up Time Count
High level command: read_rampupcnt

High level response:

Exec. OK

Read Ramp Up Count, Slope: 20

Read Ramp Down Time Count
High level command: read_rampdcnt

High level response:

Exec. OK

Read Ramp Down Count, Slope: 20

Read Max PWM Count
High level command: read_Max_pwm_cnt

High level response:

Exec. OK
Read Max PWM Count, 50

Read Stop Time Count
High level command: read_stoptimecnt

High level response:

Exec. OK

Read Stop Count: 10

FPS Run Commands:

Run to Calibrate:
This is always the first command to be issued. It forms the part of the calibration process. When issued it causes the feed to move towards the 270 deg limit switch .When the feed hits the 270 deg limit switch, the rotation is stopped and the current encoder count is initialized to 17000.

High level command: run_to_cal

High level response:

Exec. OK

Run to Preset :
Using the preset run command the feed can be rotated to the target encoder count .This can used only after calibrating the feed i.e. when feed is calibrated and idle. This command will get rejected if The target encoder count entered is less than the minimum angle or more than the maximum angle.
The speed begins to ramp up from the starting point until it reaches to the maximum PWM where it remains constant till the turning point difference is reached. When the turning point difference is reached, the speed begins to ramp down till lower RPM reaches. Once the lower RPM is reached , speed remains constant till it reaches the brake count difference, then the brakes are applied at the target position.

High level command: run_to_preset
Enter target encoder value: 15000

High level response:

Exec. OK
Run to Preset

Run to Fine Tune
This command is used to rotate the feed through small amount of counts generally less than 200 counts i.e. this command gets executed only when the difference between the current and target encoder count is less than 200 counts.

High level command: run_fine_tune
Enter target encoder value: 15050
Enter PWM cnt: 70

High level response:

Exec. OK
Run to Fine Tune

Free Run
The feed can be rotated in desired direction with this command by passing argument, 0 for anticlockwise direction and 1 for clockwise direction. The feed rotates till software limit is reached defined by Max angle and Minimum angle.

High level command: free_run_tow
Enter 0-towards 270deg / 1-towards -10deg: 1

High level response:

Exec. OK

Run Free

FPS Reboot command
High level command: reboot

High level response:

Exec. OK
Reboot

FPS Stop command
The feed rotation can be stopped by this command. The motor RPM starts reducing at the rate specified by the command “set stop time count” until the RPM reduced to ZERO and then brakes are applied.

High level command: fpsstop

High level response:

Exec. OK
Stop

Summary

The testing of FPS system over serial link under Online_V2 was completed successfully. The Hardware components (Rabbit MCM card, Ethernet switch and Broadband Fiber Optics) and Software components of Online_V2 has provided backward compatibility to existing setup. This will help in future for further development of Online_V2 irrespective of underlying hardware.

The same test methodology can be used for controlling existing Front End System under Online_V2.

Reference

1.GMRT Web site : www. http://gmrt.ncra.tifr.res.in/
2.The technical report “Online_v2 core software & Testing done” written by Mr. Raju Uprade.
3.FPS system documents available with FPS Lab.
4.TELESET-ABCCOM software Technical report by Laurent Pommier .

Acknowledgement

I would like to thank my telemetry group colleague, Mr. Raju Uprade, Shri. C. Sateesh, Mr. Mahadeo Misal, for their help in completion of this part of work. Also thank Mr. Bharat Shete and Mr. Anil Muley for their contribution in making interface cables and installation of test setup at 16 antennas.

I would like to thank FPS group, especially Mr. Abhay Bhumkar, for their help in testing of FPS system in the lab as well as antenna base with Rabbit MCM card.

I would like to avail this opportunity to express my heartfelt gratitude towards Dr. Nimisha Kanthariya, for her constant encouragement and guidance in achieving the set target.

I am also thankful to Mr. S Nayak for his support and guidance.

I also acknowledge gratitude towards Mr. R. Balasubramanium, former Group Head - Operations Group, for initiating In-House Online_V2 activities and set off all of us in right path.

In addition, I also acknowledge the contribution of Shri Naresh Sisodia, former colleague Telemetry Group, in development of Rabbit MCM hardware and Firmware.

Appendix:

The data exchanged between Online_V2 and FPS system during the testing as seen on Online_V2 terminal on teleset PC.

Online_v2- FPS system testing over rabbit mcm serial link in C06 antenna

Date : 20/03/2015 Time : 11.45 Am to 13.10 PM

Test Done by: Charu Kanade, Abhay Bhumkar, Mahadev Misal & Raju Uprade

Rabbit card with Device IP 192.168.21.107 connected to FPS system over serial link

Online_v2 machine IP : 192.168.8.45

```
[teleset@tellab2 Online]$ ./online_v2
HIGHUSER thread CREATED=> 0
SERVO thread CREATED=> 0
GUI INTERFACE thread CREATED=> 0
PYTHON INTERFACE thread CREATED=> 0
MCM SYSTEM thread CREATED=> 0

msgget: Calling msgget(0xc9,01600)
msgget: msgget succeeded: msqid = 0
Successfully Created MESSAGE QUEUE ID=0
$$$ SERVER WANTING FOR PYTHON ENVIRONMENT CLIENT TO CONNECT $$$$

 >>  ACCEPTED CONNECTION FROM FPS MCM DEVICE 192.168.21.107
FPS thread opened successfully=> 0
############## SERVER WANTING FOR CLIENT CONNECTION #####

C06 fps reboot                    // Command from Online_V2 terminal

 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => reboot
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  reboot
we wrote on the socket 35  fps reboot
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 35 20-Mar-2015 12:16:35 fps reboot
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 35
 20-Mar-2015 12:16:35
 fps
```

NUmber of RESPONSE MSG is 1
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Reboot
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_to_cal   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => run_to_cal
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_to_cal
we wrote on the socket 30 fps run_to_cal
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 30 20-Mar-2015 12:16:51 fps run_to_cal
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
30
20-Mar-2015 12:16:51
fps
NUmber of RESPONSE MSG is 1
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Run to calibrate
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_to_preset   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => run_to_preset
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_to_preset

Enter target encoder value:
15000
we wrote on the socket 32 fps run_to_preset
tar_encr_v 76 29
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

```
>> we wrote on the socket 32 20-Mar-2015 12:19:07 fps run_to_preset
tar_encr_v 76 29
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
32
20-Mar-2015 12:19:07
fps
###### NUmber of RESPONSE MSG is 1
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Run to Reset
Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps free_run_tow   // Command from Online_V2 terminal

```
CMD[0] => C06
CMD[1] => fps
CMD[2] => free_run_tow
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  free_run_tow
Enter 0-towards 270deg / 1-towards -10deg::
1
we wrote on the socket 31  fps free_run_tow
1 0
Size of Struct is ######## 1638
###### Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 31 20-Mar-2015 12:21:40 fps free_run_tow
1 0
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
31
20-Mar-2015 12:21:40
fps
###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
Run Free
Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps fpsnull   // Command from Online_V2 terminal

```
CMD[0] => C06
CMD[1] => fps
CMD[2] => fpsnull
```

Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME fpsnull
we wrote on the socket 10 fps null
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 10 20-Mar-2015 12:25:15 fps null
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 10
 20-Mar-2015 12:25:15
 fps
  ###### NUmber of RESPONSE MSG is 4
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
 Feed Calibrated and Idle
 EncCount = 1508
 Rpm = 0
 Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_version // Command from Online_V2 terminal

 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => read_version
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_version
we wrote on the socket 25 fps read_version
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 25 20-Mar-2015 12:25:58 fps read_version
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 25
 20-Mar-2015 12:25:58
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
 Read Version: 8.5

Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_Max_angle // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_Max_angle
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_Max_angle
we wrote on the socket 28 fps read_Max_angle
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 28 20-Mar-2015 12:26:22 fps read_Max_angle
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
28
20-Mar-2015 12:26:22
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Max Angle, 17284
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_Min_angle // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_Min_angle
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_Min_angle
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 29 20-Mar-2015 12:27:05 fps read_Min_angle
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
29
20-Mar-2015 12:27:05
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888

888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Min Angle, 1468
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_Brake_dd // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_Brake_dd
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_Brake_dd
we wrote on the socket 23 fps read_Brake_dd
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 23 20-Mar-2015 12:27:54 fps read_Brake_dd
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
23
20-Mar-2015 12:27:54
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Break Count Diff, 4
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_tpoint // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_tpoint
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_tpoint
we wrote on the socket 20 fps read_tpoint
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 20 20-Mar-2015 12:28:22 fps read_tpoint
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1

20
20-Mar-2015 12:28:22
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Turning Point, target: 300
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_low_rpm // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_low_rpm
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_low_rpm
we wrote on the socket 22 fps read_low_rpm
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 22 20-Mar-2015 12:28:45 fps read_low_rpm
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
22
20-Mar-2015 12:28:45
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Lower Ramp Limit, 649 int 325
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps read_rampupcnt // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_rampupcnt
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_rampupcnt
we wrote on the socket 24 fps read_rampupcnt
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

```
 >> we wrote on the socket 24 20-Mar-2015 12:29:14 fps read_rampupcnt
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 24
 20-Mar-2015 12:29:14
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
 Read Ramp up Count, 20
 Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps read_rampdcnt   // Command from Online_V2 terminal

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => read_rampdcnt
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  read_rampdcnt
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 21 20-Mar-2015 12:29:47 fps read_rampdcnt
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 21
 20-Mar-2015 12:29:47
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
 Read Ramp Down Count, Slope: 80
 Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps read_Max_pwm_cnt   // Command from Online_V2 terminal

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => read_Max_pwm_cnt
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  read_Max_pwm_cnt
 ###### Element in Command Queue fps
```

INSERTING in Command Queue fps

>> we wrote on the socket 27 20-Mar-2015 12:30:21 fps read_Max_pwm_cnt
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
27
20-Mar-2015 12:30:21
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Max PWM Count, 80
Writing to ONLINE from FPS THREAD SUCCESSFUL
C06 fps read_stoptimecnt   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_stoptimecnt
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_stoptimecnt
we wrote on the socket 26 fps read_stoptimecnt
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 26 20-Mar-2015 12:30:44 fps read_stoptimecnt
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
26
20-Mar-2015 12:30:44
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Read Stop Count, 20
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_tpoint   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_tpoint
Command for C06 ANTENNA
ANTENNA C06 C06
System fps

OP NAME set_tpoint

Enter turning point position difference:
200
we wrote on the socket 11 fps set_tpoint
set_tpoint 100 0
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 11 20-Mar-2015 12:31:58 fps set_tpoint
set_tpoint 100 0
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 11
 20-Mar-2015 12:31:58
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
 Set Turning Point, target: 200
 Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_Max_pwm_cnt // Command from Online_V2 terminal

 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => set_Max_pwm_cnt
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME set_Max_pwm_cnt
Enter max PWM cnt:
50
we wrote on the socket 17 fps set_Max_pwm_cnt
set_Max_pwm_cnt 50 0
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 17 20-Mar-2015 12:32:46 fps set_Max_pwm_cnt
set_Max_pwm_cnt 50 0
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 17
 20-Mar-2015 12:32:46
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888

888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Max PWM Count, 32
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_Max_pwm_cnt   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_Max_pwm_cnt
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME set_Max_pwm_cnt
Enter max PWM cnt:
80
we wrote on the socket 17 fps set_Max_pwm_cnt
set_Max_pwm_cnt 80 0
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 17 20-Mar-2015 12:34:41 fps set_Max_pwm_cnt
set_Max_pwm_cnt 80 0
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
17
20-Mar-2015 12:34:41
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Max PWM Count, 50
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_Max_pwm_cnt   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_Max_pwm_cnt
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME set_Max_pwm_cnt
Enter max PWM cnt:
80
we wrote on the socket 17 fps set_Max_pwm_cnt
set_Max_pwm_cnt 80 0
Size of Struct is ######## 1638

Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 17 20-Mar-2015 12:35:14 fps set_Max_pwm_cnt
set_Max_pwm_cnt 80 0
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
17
20-Mar-2015 12:35:14
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Max PWM Count, 50
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_Max_angle // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_Max_angle
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME set_Max_angle

Enter angle count:
17300
we wrote on the socket 18 fps set_Max_angle
set_Max_angle 202 33
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 18 20-Mar-2015 12:37:28 fps set_Max_angle
set_Max_angle 202 33
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
18
20-Mar-2015 12:37:28
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Max Angle, 17300
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps set_min_angle // Command from Online_V2 terminal

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => set_min_angle
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  set_min_angle

Enter angle count:
1450
we wrote on the socket 19  fps set_min_angle
set_min_angle 213 2
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 19 20-Mar-2015 12:38:07 fps set_min_angle
set_min_angle 213 2
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 19
 20-Mar-2015 12:38:07
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
 Set Min Angle, 1450
 Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps set_Brake_dd // Command from Online_V2 terminal

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => set_Brake_dd
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  set_Brake_dd
Enter Break Cnt difference::
6
we wrote on the socket 14  fps set_Brake_dd
set_Brake_dd 3 0
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 14 20-Mar-2015 12:38:45 fps set_Brake_dd
set_Brake_dd 3 0
```

Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
14
20-Mar-2015 12:38:45
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Break Count Diff, 6
Writing to ONLINE from FPS THREAD SUCCESSFUL
C06 fps set_low_rpm

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_low_rpm
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME set_low_rpm
Enter Lower RPM limit:
630
Enter Check-Interval(ms)::
20
we wrote on the socket 13 fps set_low_rpm
set_low_rpm 4 0
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 13 20-Mar-2015 12:39:29 fps set_low_rpm
set_low_rpm 4 0
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
13
20-Mar-2015 12:39:29
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Lower Ramp Limit, 0 int 20
Writing to ONLINE from FPS THREAD SUCCESSFUL
C06 fps set_low_rpm

CMD[0] => C06
CMD[1] => fps
CMD[2] => set_low_rpm
Command for C06 ANTENNA

ANTENNA C06 C06
System fps
OP NAME set_low_rpm
Enter Lower RPM limit:
300
Enter Check-Interval(ms)::
20
we wrote on the socket 13 fps set_low_rpm
set_low_rpm 4 0
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 13 20-Mar-2015 12:40:22 fps set_low_rpm
set_low_rpm 4 0
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
13
20-Mar-2015 12:40:22
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Set Lower Ramp Limit, 0 int 20
Writing to ONLINE from FPS THREAD SUCCESSFUL
C06 fps read_low_rpm

CMD[0] => C06
CMD[1] => fps
CMD[2] => read_low_rpm
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME read_low_rpm
we wrote on the socket 22 fps read_low_rpm
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 22 20-Mar-2015 12:40:41 fps read_low_rpm
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
22
20-Mar-2015 12:40:41
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999

```
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
 Read Lower Ramp Limit, 0 int 20
 Writing to ONLINE from FPS THREAD SUCCESSFUL
```

>> C06 fps reboot

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => reboot
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  reboot
we wrote on the socket 35  fps reboot
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 35 20-Mar-2015 12:44:41 fps reboot
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 35
 20-Mar-2015 12:44:41
 fps
  ###### NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999  Exec. OK
 Reboot
 Writing to ONLINE from FPS THREAD SUCCESSFUL
```

C06 fps read_Max_angle

```
 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => read_Max_angle
Command for C06 ANTENNA
ANTENNA C06 C06
System  fps
OP NAME  read_Max_angle
we wrote on the socket 28  fps read_Max_angle
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 28 20-Mar-2015 12:45:05 fps read_Max_angle
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 28
 20-Mar-2015 12:45:05
 fps
  ###### NUmber of RESPONSE MSG is 1
```

888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Read Max Angle, 17284
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_fine_tune   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => run_fine_tune
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_fine_tune

Enter target encoder value:
1550

Enter PWM cnt:
90
we wrote on the socket 33 fps run_fine_tune
tar_encr_v 7 3
pwm_cnt 90 144
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 33 20-Mar-2015 12:45:59 fps run_fine_tune
tar_encr_v 7 3
pwm_cnt 90 144
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
33
20-Mar-2015 12:45:59
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 FPS rejected command
Read Max Angle, 17284
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_to_cal   // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => run_to_cal
Command for C06 ANTENNA
ANTENNA C06 C06
System fps

OP NAME run_to_cal
we wrote on the socket 30 fps run_to_cal
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 30 20-Mar-2015 12:46:22 fps run_to_cal
 Size of Struct is ######## 1638
Size of Response Struct => 4698
 MCM => 1
 30
 20-Mar-2015 12:46:22
 fps
 ###### NUmber of RESPONSE MSG is 1
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
 Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_to_preset   // Command from Online_V2 terminal

 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => run_fine_tuneC06
 CMD[3] => fps
 CMD[4] => run_to_preset
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_fine_tuneC06

 >> C06 fps run_to_preset

 CMD[0] => C06
 CMD[1] => fps
 CMD[2] => run_to_preset
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_to_preset

 Enter target encoder value:
15000
we wrote on the socket 32 fps run_to_preset
tar_encr_v 76 29
 Size of Struct is ######## 1638
 ###### Element in Command Queue fps
 INSERTING in Command Queue fps

 >> we wrote on the socket 32 20-Mar-2015 12:48:30 fps run_to_preset
tar_encr_v 76 29
 Size of Struct is ######## 1638

Size of Response Struct => 4698
MCM => 1
32
20-Mar-2015 12:48:30
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Run to Reset
Writing to ONLINE from FPS THREAD SUCCESSFUL

C06 fps run_fine_tune // Command from Online_V2 terminal

CMD[0] => C06
CMD[1] => fps
CMD[2] => run_fine_tune
Command for C06 ANTENNA
ANTENNA C06 C06
System fps
OP NAME run_fine_tune

Enter target encoder value:
15050

Enter PWM cnt:
70
we wrote on the socket 33 fps run_fine_tune
tar_encr_v 101 29
pwm_cnt 70 112
Size of Struct is ######## 1638
Element in Command Queue fps
INSERTING in Command Queue fps

>> we wrote on the socket 33 20-Mar-2015 12:49:54 fps run_fine_tune
tar_encr_v 101 29
pwm_cnt 70 112
Size of Struct is ######## 1638
Size of Response Struct => 4698
MCM => 1
33
20-Mar-2015 12:49:54
fps
NUmber of RESPONSE MSG is 2
888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888 888
888 888 888 888 888 888 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999
999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 999 Exec. OK
Run to Fine Tune
Writing to ONLINE from FPS THREAD SUCCESSFUL