Communication between Various Processors

Revision 1

A. Pramesh Rao
2 April 91

The GMRT Online system requires communication between 3 kinds of computers- the Central Unix machine (Sun via the communication handler), 30 Antenna Base Computers (ABC) (80186 microprocesser) and the 12 (?) Monitor and Control Modules (MCMs) at each antenna (80351 microprocessors). The purpose of this note is to set specifications for the first version of the Online system so that if one or more antennas become available in the next few months, we will have some software available for using and studying the system.

At the Central station we will have a single operator system, with the operator controlling at most 4 antennas. We will still retain the proposed distributed processes model on the Unix System. However since in the first version there will be only a single user, the interprocess communication should be simplified. In due course this system can be made to evolve into the final multiuser system. Some of the essential processes required for the first phase are a) User interface program to parse and convert user commands to machine intelligible form, b) User display programs - one for the standard operators console and another for nonstandard user requested displays c) A simple Operating System Supervisor (OSS) for inter process communication, d) a subarray controller to calculate the required elevation and azimuth of the antenna e) a Unix based Communication handler to form packets for communication and transmit and receive packets from the available antennas f) a data logger for recording the monitor and control information coming from the antennas. In the first phase the available user commands can be quite limited - set and read MCM values for antenna n, issue servo commands to antenna n and so forth, and the complexity of the instruction set has to evolve with use.

In the first phase the Central computer will transmit sequentially, a packet containing ABC, MCMs, Servo or Null instructions once a second to each antenna that is available. The ABC that receives this packet immediately returns a packet containing the accumulated MCMs and Servo information for the previous second, which the data logger logs and required information is displayed on the operators console. Thus every second a packet of information is exchanged between the central computer and every ABC. The remaining free part of the second can used for repeat requests in case of transmission errors.

The ABC also has a program that operates on a one second cycle. Every second, the ABC performs the following tasks - 1) Receive a packet of instructions from the central computer and while it has the attention of the Central computer, initiate the sending of a packet containing the previous seconds monitor and servo information (this is done within the interrupt servicing routine) ; 2) Parse the received packet and put the received commands on the appropriate queues (local command, servo command or MCM command queue); 3) Execute the commands in the queues and clear the queues (priority local commands, servo commands, MCM commands); 4) Once a second request a display status from thr servo computer; 5) Once a second talk to all the MCMs and get their monitor data; 6) Record all commands issued to the servo computer and also all its expected and unexpected responses; 7) Once a second form a packet out of all the acquired data and put it in the transmission buffer, waiting for the interrupt from the central computer.

The actions taken by the Servo computer are described in the BARC document.

The MCMs performs (in the first phase) an endless loop reading and storing the ADC values of the specified channels, as also the 1 word of TTL controls bits. This loop is interrupted when it recieves a packet from the ABC, at which

instant the MCM sends a packet consisting of the current values of the monitored points.

Given below are possible packet structure between the various units.

1. ABC to MCM

a) Preamble (STX control characters, byte count etc) control character to select appropriate MCM.
b) MCM number, a control word , 4 word bit mask , n (from control word) words of control bits for setting switches.
c) Post script (ETX, Parity etc)

The control word can be used for indicating parity error in incoming data, whether the TTL levels should be set, for indicating the number of words of control bits for the particular MCM.

The 4 word bit mask is user selectable at the operators console. Each MCM will return the ADC values of only those multiplexer channels for which the bit is on. In the first phase, we will assume that the bit mask will correspond to all valid monitor points for the MCM and that the mask will remain constant for the observation. (Later the constant bit mask will correspond to only critical monitoring points and further bits can be set on or off by the user.)

2. MCM to ABC

a) Preamble
b) MCM number, a control word, 4 word bit mask, n (defined in control word) words of control bits used.
c) N x (ADC values) where N= no of bits on in 4 word bit mask.
(Item removed from previous version)
d) Post script.

As before, the control word can be used to indicate parity errors, whether all is OK with the MCM, etc

3. ABC to Central Computer

a) Preamble.
b) Station ID, time stamp and one or more control and status words (use to show error on links to central computers , servo computers and MCMs, Servo OK, MCMn OK etc), a fixed data structure showing all display points (analog and digital) returned by the servo computer. Depending on the packet mode (specified in the control words), this data structure may be ASCI or binary. In the first phase we will keep it as ASCI.
c) For each MCM items 2b,c and d (just 2b if particular MCM not OK).
d) A variable length sequence showing all the significant commands sent to servo computer and its expected and unexpected responses (with individual fields separated by delimiters). (In later phases, one can define different levels of significance that the user can select).
e) A variable length string showing responses from ABC itself.
e) Postscript

4. Central computer to ABC

a) Preamble with station select code.
b) Station ID, MODE (broadcast, individual command, repeat request, etc).

c) Sequence of commands separated by delimiters. There are 3 type of commands -

1) Station commands meant for the ABC itself like SET TIME, SET 4 word bit mask etc. The vocabulary of the station commands should evolve with time. In later versions, we should have a system where brief binary instructions sent from the centre are converted into longer commands that are required by the MCMs and the Servo computer. In the first version, we will keep this command set minimal.

2) Servo commands in the form required by the servo computer

3) MCM commands, essentially the same as item 1b

Each type of command will have a different label.

d) Postscript