Tritech Sonar Protocol – Frequently Asked Questions

Q1) Which fields are related to start and stop scanning? How I can start and stop sonar head?

A1) There are no Start and Stop Scanning fields/commands. The surface program must send an 'mtSendData' trigger command down to ping the Sonar. The surface must first send parameters (in the 'mtHeadCommand') to the Sonar before it can start to send the 'mtSendData' triggers. Once it starts sending 'mtSendData' commands, the surface program just simply stops sending these to stop the scanning. If the surface program stops sending 'mtSendData' commands for a period of 1 minute then the Sonar will do a reset – it will return to the centre/ahead position and the stepper motor will go into standby until another 'mtSendData' is received to wake it up.

Q2) Can the Sonar head be sent to a specific position (bearing) by a command?

A2) There is no mode to fire a ping at a specific bearing. The Sonar scanning is set within a Left and Right Limit. 'Left Limit' and 'Right Limit' are set in the 'mtHeadCommand'. At any time during scanning, if you want to change the Left and Right Limit then you simply send down another 'mtHeadCommand' with the new Limits set. The Sonar will then re-position itself to these new Limits after it completes it's current ping.

Here is a procedure to perform the point and ping functionality that is required…

1. On power up the Sonar head requires a set of parameters from the surface. These parameters are sent to the Sonar head using the 'mtHeadCommand'. When the Sonar is not scanning/pinging, and/or awaiting parameters, it will always send an 'mtAlive' broadcast message to the surface. This broadcast includes flags stating parameter status (Byte 21, Bits 6&7).

2. Once the Sonar receives and accepts the 'mtHeadCommand' it will reflect this in the next 'mtAlive' broadcast it sends. When Bit 6 is set to 0 ("Has Parameters") then the Sonar is ready to be triggered.

3. The Sonar is triggered by sending an 'mtSendData' command to the Sonar from the surface. Depending on the comms duplex mode of the head (Half/Full) a single 'mtSendData' will trigger one ping (Half Duplex) or two consecutive pings (Full Duplex) in the Sonar head. The 2 ping Full Duplex operation allows the Sonar to run one ping ahead thus discounting any data turnaround delays encountered with a Half Duplex mode – we can thus keep scan speeds up. The Duplex mode can be configured in the Sonar using the Setup program on the surface computer. The Setup program is installed as a desktop icon when the installer is run from the software CD that is supplied with the Sonar.

4. When the Sonar receives the 'mtSendData' and fires a ping (or two pings if F.D.), once the ping routine is complete the Sonar head will send an 'mtHeadData' reply to the surface which includes the scanline data samples for that ping.

5. The normal operation is to configure the Sonar with Left and Right polar limits to set the width of the required scan sector (0 to 360 deg). These Limits are set within the parameters in the 'mtHeadCommand'. Along with Limits, the

Step angle parameter defines the angular separation between successive pings within the scan sector. N.B. In Full Duplex operation, 1 x ‘mtSendData’ trigger will fire 2 pings at adjacent bearing points within the scan sector.

6. It is possible to operate a point and ping mode whereby you can set a particular bearing and trigger one ping at this point. Then, move to a new bearing and trigger another ping and so on. In this mode you need to send an ‘mtHeadCommand’ before each ‘mtSendData’ trigger. The ‘mtHeadCommand’ is used to set the bearing point before the ‘mtSendData’ trigger is sent. In the ‘mtHeadCommand’, set the Left and Right Limits to the required bearing point, i.e. LL = RL which sets the Sonar into a stare mode.

Q3) Do I have to send 'mtSendData' in order to get sonar data?
If so, do I have to send next 'mtSendData' after receiving 'mtHeadData'? Or, just send 'mtSendData' with timer?
If you use timer, what is a typical interval?

A3) The Sonar can be set to Half or Full Duplex modes - there is a ‘Half Duplex’ check-box that is set using the Setup program (in that program, click on ‘Action‘ - ‘Setup’ for Node 2).

In Full Duplex mode (i.e. ‘Half Duplex’ is un-checked), the Sonar will fire 2 pings for the first ‘mtSendData’ it receives (and so return 2 x ‘mtHeadData’ replies). After the surface receives the 1st ‘mtHeadData’ reply it should immediately send another ‘mtSendData’ to the Sonar. Here is sequence when you do this…

```
-> mtSendData
<- mtHeadData
-> mtSendData
<- mtHeadData
-> mtSendData
<- mtHeadData (note if mtSendData stops then Sonar will still send one more
mtHeadData after this).
```

The above sequence ensures the Sonar is always ready to fire the next ping and there are no delays in telemetry Tx/Rx turnaround. The Sonar cannot store ‘mtSendData’ commands so if you send 5 of these commands in sequence then the Sonar will only fire 2 pings (i.e. it can only store 1 ping at any time).

If you send just 1 ‘mtSendData’ then 2 ‘mtHeadData’ replies are returned for this, i.e.

```
-> mtSendData
<- mtHeadData
<- mtHeadData
```

Sending another ‘mtSendData’ after this, the Sonar will again fire 2 pings and you will receive 2 ‘mtHeadData’ replies, i.e.

```
-> mtSendData
<- mtHeadData
<- mtHeadData
```

So, the aim is to send the next ‘mtHeadData’ immediately after a ‘mtSendData‘ is received and so ensure the Sonar always has 1 ping stored. That way there is no delay in telemetry turnaround as would be the case with a Half Duplex operation.

In ‘Half Duplex’ mode, sending 1 ‘mtSendData’ will instruct the Sonar to fire 1 ping (and not 2 pings as in Full Duplex). So the surface needs to wait for the ‘mtHeadData’ before it sends the next ‘mtSendData’. But in this mode the Sonar does not have 1 ping stored and so it has to wait to receive the next ‘mtSendData’ before it can fire the next ping.

When you send an ‘mtSendData’ to the Sonar, it is recommended to set a 1 second timer. If an ‘mtHeadData’ is not received in this 1 second period then try sending another ‘mtSendData’ and wait another 1 second. This can continue but remember that if the communications with the Sonar stops then the Sonar will start to send ‘mtAlive’ broadcasts at a 1 second interval. So, after trying to re-send the ‘mtSendData’ for a few attempts, you should then stop and listen for the ‘mtAlive’ broadcasts from the Sonar. If you do not receive an ‘mtAlive’ then you should flag a ‘No Communications’ error at the surface and so prompt a check on the system power and cabling.

Q4) During operation, at what instant can I update the scan parameters by sending an 'mtHeadCommand' instead of a 'mtSendData'?
Once the parameters are updated, do I then have to restart the enquire/reply loop?

A4) Yes, that is correct. At any time in between the mtSendData and mtHeadData interchange an mtHeadCommand can be sent to the Sonar. It will be buffered by the Sonar and processed once the Sonar completes its current ping routine. For the full duplex mode the mtHeadCommand can be sent at any time. For half duplex then you need to wait until the Sonar returns the current mtHeadData reply then send the mtHeadCommand followed by another mtSendData (thus re-continuing the trigger sequence).

Note that the Sonar does not send an Acknowledgment when it receives a parameter update with a new mtHeadCommand. But there is a method to check that a parameter change has been applied. For instance if your software application has screen controls for Gain, Range etc, and you change say the Gain control then this change will be sent to the Sonar in an mtHeadCommand without a returned Ack as I've said. To check the Gain parameter has indeed been set in the Sonar you must check the copy of the parameters, including Gain, that are returned in the next mtHeadData reply.

Q5) Do scan parameters such as Nbins and 4-bit / 8-bit ADC resolution affect the amount of data in each acoustic roundtrip?
Also, being the throughput constant, do they affect the communication delay? Do other parameters like Range, Dynamic Range etc affect the communication overhead too?

A5) Generally speaking, if you operate the Sonar on Range Scale settings of 30 metres and below then the communications overhead is the governing factor in the scan timing. That is, the communications can't clear the transmission of the current mtHeadData reply before the next ping has completed and consequently the Sonar has to wait for the comms buffer to clear of the current mtHeadData before it can send the next mtHeadData which it has in waiting.

At Range Scale settings over 30 metres then the communications buffer has usually cleared before the next ping has completed it's acoustic round trip. Of course if you reduce the communications speed below 115,200 Baud then the communications may not clear again before the next ping is completed.

If you operate on Range Scales below 30 metres or have a need to reduce communications speed below 115,200 then to counter this you could take less bin samples or reduce sampling from 8 to 4 bits which of course will reduce the size of the mtHeadData replies and so take less time to clear through the comms channel.

Parameters such as Dynamic Range and other Receiver parameters such as Gain for that matter will not affect the communication overhead. The 4/8 bit mode, number of bins and range scale are the parameters that affect timing as well as communications Baud rate of course. Also, my comments in Q1 about the timing in the mtSendData and mtHeadData interchange as well as the duplex settings also contribute to these timings.

Q6) Our requirement is to send all RS-232 Sonar telemetry via a data multiplexer. What data and packet sizes can we expect to handle?

A6) Firstly, a bit about the protocol used between Sonar and the surface PC. When the Sonar comes on-line its starts to send 'mtAlive' broadcasts at 1Hz so that the surface is aware of it. The surface then requests Version and User data in separate commands ('mtSendVersion' and 'mtSendBBUser'). The surface uses these for verification mainly.

The surface must then send Parameters (Gain, Scan Angles etc) to the Sonar before it can begin to scan. The 'mtHeadCommand' is sent to the Sonar for this purpose. Thereafter, the surface then triggers the Sonar and controls its scanning and ping update rate. An 'mtSendData' command is sent to the Sonar, the Sonar fires a ping and then returns the 'mtHeadData' reply with the scanline data samples ("Bins") contained.

In full duplex mode (RS-232 mode) The Sonar actually will trigger 2 pings for the first 'mtSendData' that it receives. This ensures the Sonar is always ready to fire another ping after the current ping is completed and there is no waiting for telemetry handshaking between surface and Sonar, i.e. waiting for the next 'mtSendData' to be sent down. The Sonar is topped up with 'mtSendData' commands after every reply it sends to the surface, so it is always processing 1 ping with a trigger for the next ping in the locker.
If the Sonar scanning is paused or the surface connection is broken (i.e. surface software shut-down or telemetry disconnected) the Sonar will start to re-send the 1Hz 'mtAlive' broadcasts again.

Therefore, with all considered, the multiplexer would need to handle data packets between 22 ('mtAlive') and up to 1000 bytes ('mtHeadData' with 800 Bins, i.e. ExtraData can be tagged at end of Bins).

Q7) We are running an RS-232 model Sonar in the Full Duplex mode. If we can increase the baudrate to the max (115200) is there any other parameters that should be set to get the Sonar to run at acoustic limits?

A7) To achieve better scan speeds you may need to set a larger (stepper motor) step size which is the angle between adjacent pings in the sweep. 'Step Size' is set in the 'mtHeadCommand' and is in 1/16 Gradian units.

The Tritech surface programs give option of running in 3 or 4 resolutions which vary the scanning speed and picture resolution. Here are some recommended settings which you can adjust to suit your own requirements...

Lo Res. : StepSize = 32 (in 1/16 Grads), Nbins = 10 * RangeScale
Med Res. : Step Size = 16, Nbins = 15 * RangeScale
High Res. : Step Size = 8, Nbins = 20 * RangeScale

...'NBins' is the number of samples over the Range Scale which are taken for each ping. This figure, as well as 'RangeScale' is also set in the 'mtHeadCommand'.

The Lo Res will produce the fastest scanning but the Sonar image will be quite blocky. There will always be a trade off between image resolution and scan speeds so it is recommended to have 3 or 4 Resolution settings as shown above.

With the RS-232 mode you don't have to operate in Half Duplex which would incorporate overheads and delays in you having to wait for an 'mtHeadData' reply before issuing another 'mtSendData' to trigger the next ping.
Instead, with RS-232 you can operate as Full Duplex as follows;

1. Send 1 x 'mtSendData' to Sonar. The Sonar will keep a count of the number of triggers (1 trigger = ping routine). When the trigger count is 0 and the Sonar receives 1 x 'mtSendData' it will increase the trigger count to 2 (The Sonar trigger count is 2 maximum = 1 to process now and 1 to buffer).

2. When the Sonar has a trigger count of 2 it will process 1 trigger while it buffers the other. At then end of the 1st trigger it will send the 'mtHeadData' reply for that ping and at the same time fire off the 2nd buffered trigger (N.B. count now = 0).

3. Once the 'mtHeadData' Reply is received at the surface, another 'mtSendData' should be sent to the Sonar immediately before the next 'mtHeadData' Reply (for 2nd trigger) is received. This will top up the Sonar trigger count so that it always has 1 buffered trigger which can then be executed at the end of the current ping routine.

4. This process should continue so that the Sonar always has 1 trigger buffered which it can process immediately after the current ping routine is completed.

Therefore sequence is...
mtSendData ->, mtHeadData <-, mtSendData ->, mtHeadData <-, mtSendData ->, mtHeadData <-, ...etc.

For an RS-485 Sonar, there is no trigger count or buffering inside the Sonar because the telemetry is Half Duplex and not bi-directional as with the RS-232 Sonar mode. Therefore in Half Duplex mode there will be turnaround delays induced in the data packet interchange between Sonar and surface and Sonar scanning will become evidently slower as a result.