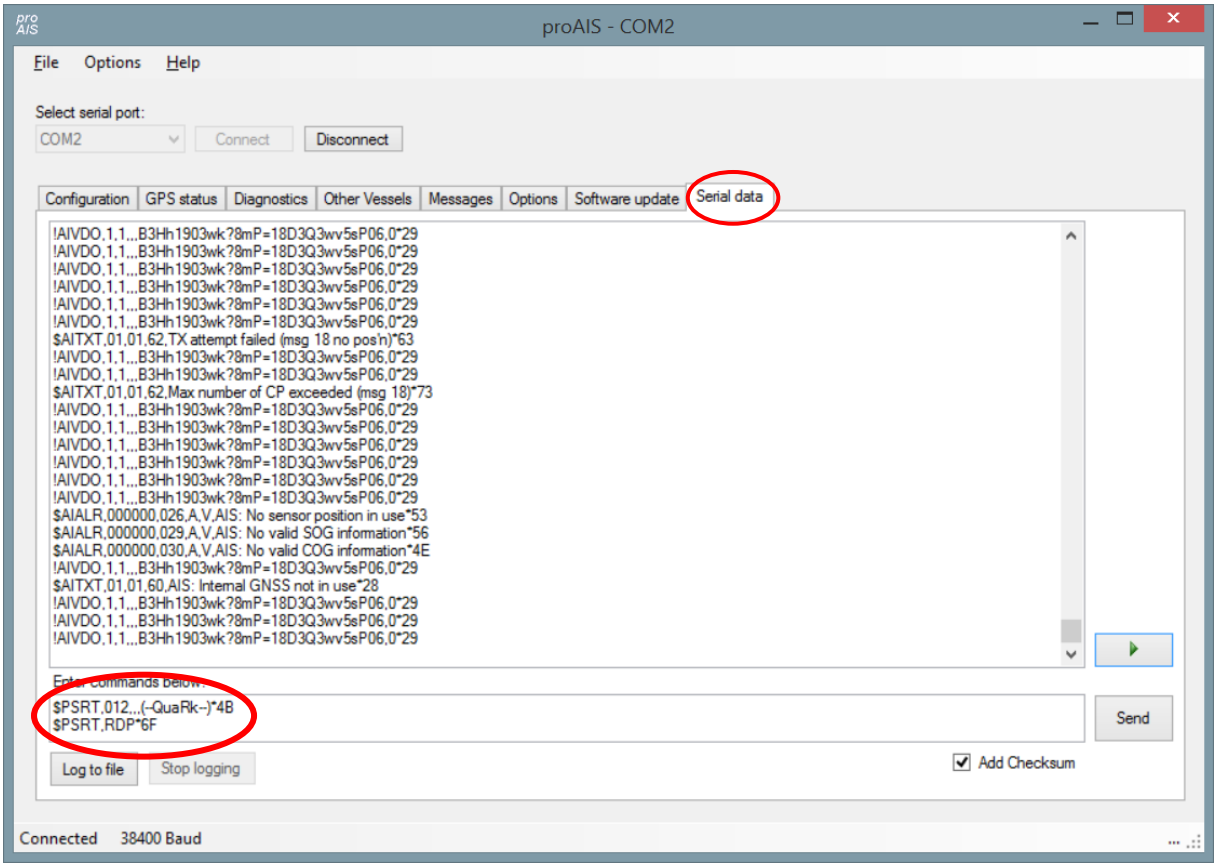
Commandes à introduire (Logiciel ProAIS) pour faire un reset d'usine du transpondeur

```
$PSRT,012,,,(--QuaRk--)*4B
$PSRT,RDP*6F
```



SRT proprietary AIS commands

My Navico NAIS-300, like most other commercial AIS class B products that have been available for a while, uses the SRT AIS class B OEM board. I know the same board is used in the Raymarine 500, True Heading AIS-CTRX, Transas, Digital Yacht AIT250/1000, and many more. The tell-tale seems to be that the AIS transponder comes with *Pro AIS* software. I have hooked up a serial data logger so that I could find out what NMEA-0183 commands the Pro AIS software sends to the transponder. Unfortunately neither SRT nor any of the vendors publish the protocol.

My research into this started because I wanted to control my AIS transponder's transmit setting using software. For some reason there are many functions that you can assign to the *remote button* input, but not a reliable level-style on/off of the transmit functionality. This has been added to recent firmware versions, but mine contains older firmware. However, an hour's worth of analysis shows that controlling the transmitter is actually quite easy. In fact, almost everything that you can set up using Pro AIS is easy as pie to implement.

Basics

The SRT product, logically, uses the 'P' prefix to indicate Proprietary, followed by SRT. Here is an example of a PSRT message:
`$PSRT,LED,01*49`
Just like all other sentences I document on this page the shown message format is inclusive of the NMEA style checksum bytes at the end of the line, e.g. `*49` in the message above.

For some reason some sentences start with a prefix `$DUAIQ`. I have no theory as to why this is so, maybe the data is intercepted by a different microcontroller.

Authorization

Some sentences require the passing of a password. Luckily, the protocol to do this is very simple -- just send the following sentence before every *authorized* message:
`$PSRT,012,,,(--QuaRk--)*4B`

Silent mode

To make the AIS silent (not transmit its own position) send the following authorized (prefix with the command above) sentence:
`$PSRT,TRG,02,33*6A`

To make the AIS transmit its own position send the following authorized sentence:
`$PSRT,TRG,02,00*6A`

Alarm mode

To make the AIS output all alarms every minute send the following authorized sentence:
`$PSRT,ALM,0000*45`

To make the AIS output only the active alarms send the following authorized sentence:
`$PSRT,ALM,0001*44`

GPS update speed

To make the AIS output GPS data every second send the following authorized sentence:
`$PSRT,GER,01*54`

To make the AIS output GPS data every four seconds send the following authorized sentence:
`$PSRT,GER,00*55`

GPS data

The SRT board has a complete GPS on-board. For some reason it only sends out two GPS sentences: RMC and GBS. See the GPSD source for more information on these sentences (as well as those below.)

To get the GPS to send out more GPS sentences send the following authorized sentence:
`$PSRT,GPSDATA,,,1*60`
This will cause the board to send out VTG, GGA, GSV, GLL and ZDA sentences as well as RMC and GBS. My particular board has a small bug in that it also starts sending out two copies each of the RMC and the GBS commands.

To get it to stop sending the additional GPS messages, send the following authorized command:
`$PSRT,GPSDATA,,,0*61`

Interrogating the board

There is a whole stack of sentences that can be used to read out system information. These are

LED status

```
Send: $DUAIQ,LED*29
Recv: $PSRT,LED,a*hh
a bit 1: Power On
a bit 2: TX timeout
a bit 3: Error
a bit 4: SRM status
hh: checksum
```

Internal data

```
Send: $DUAIQ,ADC*22
Recv: $PSRT,ADC,a,b,c,d,e,f,g*hh
a: Tx forward power
b: Tx reverse power
c: RSSI Rx 1
d: RSSI Rx 2
e: Internal 3V3 supply
f: Internal 6V supply
g: Supply voltage
hh: Checksum
```

Station Static Data (AISSD)

```
Send: $DUAIQ,SSD*20
Recv: $AISSD,a,b,c,d,e,f,g,h*hh
a: Callsign, 8 bytes fixed length; @ for unused bytes
b: Ship's name,20 bytes fixed length; @ for unused bytes
c: GPS antenna distance from bow, in m
d: GPS antenna distance from stern, in m
e: GPS antenna distance from port side, in m
f: GPS antenna distance from SB side, in m
g: DTE
h: Source Identifier
hh: Checksum
```

MMSI

```
Send: $DUAIQ,010*55
Recv: $PSRT,010,,,c*hh
a: ?
b: ?
c: MMSI
```

OEM name

```
Send: $DUAIQ,SRM*28
Recv: $PSRT,SRM,a,b,c*hh
a: ?
b: ?
c: OEM name used in AIS messages, 7 bytes fixed length; unused bytes are
filled with @.
hh: Checksum
```

Vessel Static Data (VSD)

```
Send: $DUAIQ,VSD*25
Recv: $AIVSD,a,b,c,d,e,f,g,h*hh
a: Type of ship and cargo. For recreational use: 36 = Sailing vessel, 37 =
Pleasure craft
```

```
b: Maximum present draught, always 00.0
c: Persons on board, always 0000
d: Destination, always @@@@@@@@@@@@@@@@@@@@
e: Est. UTC of arrival, always 000000
f: Est. day of arrival, always 00
g: Est. month of arrival, always 00
h: Navigational status, always 00
i: Regional application flags, always 00
hh: Checksum
```

Software version

```
Send: $DUAIQ,SWF*26
Recv: $PSRT,SWF,a,b*hh
a: AIS software version
b: FPGA version
hh: Checksum
```

Transponder Serial Number

```
Send: $DUAIQ,SNO*36
Recv: $PSRT,SNO,a*hh
a: Serial number in ASCII, 10 digits. Usually all 0.
```

Reset Data Programming

The following command can be used to reset the AIS back to its factory settings, inclusive of the MMSI number, so that it can be reprogrammed, for instance when you want to sell your AIS transceiver.

```
$PSRT,RDP*6F
```

I have received numerous reports from people who tell me that this works fine, over a range of devices. Just make sure your AIS came with ProAIS and you should be OK.