Virginia Tech Ground Station
TNC Interfacing Tutorial

Zach Leffke, MSEE (zleffke@vt.edu)
Research Associate
Aerospace Systems Lab
Ted & Karyn Hume Center for National Security and Technology

3/23/2018
Agenda

• TNC Connection Overview
• KISS Protocol
• AX.25/HDLC Protocol
• AFSK/FSK/GMSK Modulation
• ....System Review....
• OSI Stack
• Remote Connection
• VTGS Remote Interface
• Summary

3/23/2018
Primary Ground Station - UPEC

Host Computer

Client

TC SW
AX.25 KISS
TCP
IP
AAC

Software TNC

TC SW
KISS
TCP
IP
AFSK

SoundCard Interface

RADIO
DATA IN
FM RF

Audio

Remote Ground Station - VTGS

Host Computer

Software TNC

TC SW
KISS
TCP
IP
MAC

SoundCard Interface

Telecommand (TC)

RADIO
DATA IN
FM RF

Audio

Spacecraft

C&DH Comput

RADIO
TC SW
AX.25 KISS
AFSK
FM RF

SERIAL

Audio

INTERNET

VPN CONNECTION
TNC Connection Overview

TNC Implementation Types

Hardware Radio DATA Jack

Specific Interface Examples
TNC Implementation Summary

- Multiple implementation options exist
  - Hardware TNC Ground Station
  - Software TNC + Sound Card
  - Software Defined Radio Receiver
  - Software Defined Radio Transceiver

- Hybrid SDR RX / HW Radio TX implementations
Hardware Radios – Common for Satellite
Hardware TNCs
Radio Sound Card Interfaces

- Good ones offer optical isolation (optocouplers).
- Appear as a soundcard to host computer OS.
- HW control of TX/RX volume.
- Also used for non-packet modes. Make sure one with the proper capabilities is selected.
- Different non-sound related options available (such as PTT control, and CW keying). Again, make sure one with the proper features is selected.
Software Defined Radios

Receive Only

Transceivers (TX & RX)

See for more: https://www.rtl-sdr.com/roundup-software-defined-radios/
The ‘S’ in SDR?...here are a few…

SDR#
GNU Radio

HDSDR
gqrx

GNU Radio

gr-fosphor (GNU Radio)
Putting it all together

LOTS of options exist........

....and combinations of options...

....how to decide?!?!?!.....

....Lets go over some fundamentals........
Hardware TNC Ground Station

Host Computer

HW TNC

HW Radio

Antennas

Host Computer

HW TNC

HW Radio

Antennas
Software TNC + Sound Card Ground Station

Host Computer

Sound Card Interface

HW Radio

Antennas

Host Computer

TCP/IP
 AX.25/KISS

Software TNC

Sound Card Driver

USB Port

USB / Serial 'audio' samples

Audio/NRZ PTT

Hardware Radio

VHF RF Chain

UHF RF Chain

Client Software

Software TNC

Sound Card IC (ADC/DAC)

FTDI IC

USB Hub

COM PORT (PTT)

MIC

RX

TX

DTR

PTT

SPKR

USB Port

Client Software

Software TNC

Sound Card Driver

USB Port

USB / Serial 'audio' samples

Audio/NRZ PTT

Hardware Radio

VHF RF Chain

UHF RF Chain
Software Defined Radio Receiver

Host Computer

USB

Digitized RF Samples (IQ)

Software Radio Dongle

Antennas

TCP/IP

AX.25/KISS

USB Port

Client Software

Host Computer

Software TNC

Audio Samples

SDR Signal Processing

IQ

SDR Driver

USB Port

RF Chain

RF

Software Radio Receiver

USB PHY

RX

RF

TX

IQ

Q

2

10°

90°

D A

D A

RX

USB PHY

TX

IQ

Q

2

10°

90°

D A

D A

RX

RF

USB PHY

TX

IQ

Q

2

10°

90°

D A

D A
Software Defined Radio Transceiver

NOTE: This is basically the chosen VTGS architecture
TNC Connection Overview

TNC Implementation Types

Hardware Radio DATA Jack

Specific Interface Examples
Hardware Radio DATA Jack

ICOM-9100 Example

What is it for?

- Used for packet operations.
- Bypasses audio filtering (which can distort digital communications).
- Standardized PS-2 Connector and pinouts across vendors.
- High speed (9600 baud) and low speed (1200 baud) packet access.
- Connects to ‘Radio Port’ on TNC.
FM Transceiver (simplified)

RF Stage
- RX Channel Filter
- Tuning Control (the BIG knob)
- AGC

IF Stage
- Demodulator (FM Discriminator)
- De-Emphasis
- Volume Control

Audio Stage
- Speaker
- De-Emphasis
- Pre-Emphasis
- Audio Filtering
- ALC

TX/RX Control
- TX Filter
- TX PWR Control
- TX/RX Control

FM Transceiver

3/23/2018
FM Transceiver – DATA Jack Tap Points

*NOTE:*

- Selecting 1200 Baud packet mode on radio menu connects 1a tap point
- Selecting 9600 Baud packet mode on radio menu connects 1b tap point
- Unsure if pins 4 and 5 are affected by this menu selection. (TEST!)
Probably want MAIN assigned to uplink so that when PTT is triggered, the radio transmits on the Uplink Frequency. SUB would then be assigned to Downlink noting the comments below. *Could be totally wrong about this, especially if IC-9100 has a ‘satellite mode’ of operation that overrides these settings.*

---

**Table: DATA Jack Specifications**

<table>
<thead>
<tr>
<th>DATA2</th>
<th>PIN No.</th>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>SPECIFICATIONS</th>
</tr>
</thead>
</table>
|       | 1       | DATA IN | Input terminal for data transmit. (1200 bps: AFSK/9600 bps: G3RUH, GMSK) | Input level (1200 bps) : 100 mV  
Input level (9600 bps) : 0.2 to 0.5 Vp-p |
|       | 2       | GND   | Common ground for DATA IN, DATA OUT and AF OUT. |       |
|       | 3       | PTTP  | PTT terminal for packet operation. Connect to ground to activate the transmitter. | Input voltage (High) : 2.0 V to 20.0 V  
Input voltage (Low) : −0.5 V to +0.8 V |
|       | 4       | DATA OUT* | Data out terminal for 9600 bps operation only. | Output impedance : 10 kΩ  
Output level : 1.0 Vp-p |
|       | 5       | AF OUT* | Data out terminal for 1200 bps operation only. | Output impedance : 4.7 kΩ  
Output level : 100–300 mV rms |
|       | 6       | SQL*   | Squelch out terminal. This pin is grounded when the transceiver receives a signal which opens the squelch.  
• To avoid interfering transmissions, connect squelch to the TNC to inhibit transmission when squelch is open.  
• Keep RF gain at a normal level, otherwise a “SQL” signal will not be output. | SQL open : Less than 0.3 V  
5 mA  
SQL closed : More than 6.0 V  
100 μA |

* The pin 4 (DATA), pin 5 (AF) and pin 6 (SQL) output capabilities are for the MAIN Band’s AF and squelch by default. You can change this setting in “DATA AF/SQ Select” of the Set mode. (p. 166)

---

**From IC-9100 Manual**

---

**52. DATA AF/SQ Select (Default: MAIN)**

Set the [DATA2] socket's pin 4 (DATA), pin 5 (AF) and pin 6 (SQL) output usage.

- **MAIN**: Sends the MAIN Band's receive audio and squelch.
- **SUB**: Sends the SUB Band's receive audio and squelch.
• Each vendor is different.
• Different connector styles, different pinouts, different settings.
• Must consult the manufacturer documentation to properly configure and fabricate interface cable.
• Some TNC/Soundcard vendors sell pre-fabricated radio specific interface cables.
• MUST pay attention to signal levels (volume control) on DATA IN (1), DATA OUT (4), and AF OUT (5) pins to avoid signal distortion (lost packets).
Volume Control

• The AF OUT (5) and DATA OUT (4) Pins bypass the majority of the audio conditioning circuitry. This is to avoid distortion. This means that the VOLUME control knob on the radio has no effect on the voltages output on these pins. The human operator can set the volume of the radio speaker to a comfortable level for monitoring without affecting the voltages output on these pins. Most TNCs have a configurable ‘input gain control’ (or similar) setting to set this at an optimal input level for the TNC. Most sound card interfaces have an ‘RX knob’ which is essentially a volume control for the same purpose.

Squelch Pin (6)

• SQL (6) connection is optional. Purpose is to inhibit Transmit when open (signal detected). For satellite work, recommend leaving disconnected so that the squelch can be completely opened on the radio. This will allow a human operator to listen to signals too weak to decode (like near AOS/LOS) to monitor system performance without inhibiting transmit (uplink) capability.

• This pin is most useful for Terrestrial APRS use when in a cluttered RF environment (many transmitters on same frequency and bursty). This pin is usually connected to a ‘Data Carrier Detect’ Pin on the TNC to inhibit TX when signals are detected on the channel in order to avoid packet collisions.
• The PTT (3) pin is used to enable transmit on the radio.

• It is tempting to connect directly to the TNC (that’s what the manual says to do right!?!?).

• For satellite work, usually this is not actually desirable because there are multiple devices that need to ‘see’ the PTT signal to switch the entire system (not just the radio) into a transmit state.

• Devices that need to see the PTT pin include the Low Noise Amplifiers, The High Power Amplifiers, the Radio and sometimes additional devices (such as coax relays) depending on the specifics of the ground station design.

• Normally, this pin out of the TNC is used to trigger a device called a ‘sequencer’ that has multiple PTT output channels with configurable delays between each channel. The radio PTT signal on the DATA jack is usually the LAST PTT signal to be enabled.

• More on Sequencing later........its important!
TNC Connection Overview

TNC Implementation Types

Hardware Radio DATA Jack

Specific Interface Examples
Some Examples

- Hardware TNC And Radio: KPC-9612+ TNC & ICOM-9100
- Sound Card Interface and ICOM-9100
- Software TNC Examples (for sound card interfaces)
- Software Radio Receiver Example
- ‘Client Software’ Examples
HW TNC To HW Radio Example

- **KPC-9612+**
  - Port 1 – Low Speed (AFSK1200)
  - Port 2 – High Speed (FSK9600)
  - Computer = Serial

- **ICOM-9100**
Cable Fabrication Time!

<table>
<thead>
<tr>
<th>Terminal Name</th>
<th>KPC-9512 Plus Terminal Number</th>
<th>Computer Terminal Number (DB-25)</th>
<th>Computer Terminal Number (DB-9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FG</td>
<td>1</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>TXD</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>RXD</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>RTS</td>
<td>4</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>CTS</td>
<td>5</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>DSR</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>SG</td>
<td>7</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>DCD</td>
<td>8</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>DTR</td>
<td>20</td>
<td>20</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Terminal Nr</th>
<th>Nomenclature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TXA</td>
<td>Transmit audio (AFSK out)</td>
</tr>
<tr>
<td>2</td>
<td>RXA</td>
<td>Receive signal (input)</td>
</tr>
<tr>
<td>3</td>
<td>TXA</td>
<td>Transmit signal (output)</td>
</tr>
<tr>
<td>4</td>
<td>RXD</td>
<td>Receive signal (digital input) (not used)</td>
</tr>
<tr>
<td>5</td>
<td>CTLA 9600</td>
<td>Control line A (output) (High Speed Port)</td>
</tr>
<tr>
<td>6</td>
<td>CTLB 9600</td>
<td>Control line B (output) (High Speed Port)</td>
</tr>
<tr>
<td>7</td>
<td>RX/IN</td>
<td>Receive quality (output)</td>
</tr>
<tr>
<td>8</td>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>9</td>
<td>EXT-IN</td>
<td>External input for Power/Reset*</td>
</tr>
<tr>
<td>10</td>
<td>CTLA(1200)</td>
<td>Control line A (1200 Port)</td>
</tr>
<tr>
<td>11</td>
<td>GND/RST</td>
<td>Ground (may be configured as external reset)*</td>
</tr>
</tbody>
</table>

* Shielded
NOTES
1. Port 1 (AFSK1200) Controls Transmit (PTT) Since this is the uplink rate.
2. Port 1 (AFSK1200) Can also Receive (not needed, but useful for testing, related to KISS Protocol).
3. Recommend NOT Connecting Squelch (SQL/XCD) Lines to Data Jack (used For TX inhibit).
4. Dashed Lines are optional (SQL to DATA and Calibration equipment).
KPC-9612+ TNC to HOST Computer Cable

NOTES:
1. SG/TXD/RXD/Shield Are Required!
2. Other Signals are optional.
3. FTDI Converters can cause EMI/RFI and ‘splatter’ receiver front ends.
   **Keyspan USA-19HS USB to Serial to converter is a recommended ‘known quiet’ device.

USB To Serial Adapter**
SoundCard Interface To HW Radio Example

DIGIBOX2

ICOM-9100

3/23/2018

TNC Interfacing Tutorial
Digibox2 Interface To HW Radio Example

NOTE: In order to test this connection you need an FSK9600 Transmit reference signal. CANNOT use this cable with local APRS transmissions for testing.

NOTE: Can test this configuration with local Terrestrial APRS signals (useful for testing higher layer AX.25/KISS interfaces).
• Option for switchable low speed or high speed connection
• Can select RX1 (AFSK1200) for low speed testing with terrestrial APRS signals. Can also use to test other system functionality against satellites (doppler control, antenna pointing, etc.).
• Once confident with higher level AX.25/KISS processing and satellite tracking (Doppler, pointing) can then switch to RX2 (FSK9600) for testing against 9600 baud satellites.
• NOTE: This is a logical diagram, fabrication details are left to the experimenter to decide.
Dual Digibox2 Interface To HW Radio Example

Low Speed Data Cable:
- Transmit: Low Speed (AFSK1200)
- Receive: Low Speed (AFSK1200)

High Speed Data Cable:
- Receive: High Speed (FSK9600)
Digibox2 COM Ports (Not Soundcards!)

Jumper positions for this Example:
COM A = PTT/CW Key Control (Transmit Signal)
COM B = Yaesu’s Computer Aided Transceiver (CAT) = ICOM’s CI-V = Computer Tuning Control
Software TNCs (for soundcard interfaces)

SoundModem (AFSK1200)

High Speed Soundmodem (FSK 9600)
### Client Software Examples – AMSAT FOX Software

#### Satellite Fox-1A(FM) Mode: TRANSPONDER

<table>
<thead>
<tr>
<th>Last Realtime:</th>
<th>Resets: 0</th>
<th>Uptime: 89991</th>
<th>Max: Resets: 0</th>
<th>Uptime: 89568</th>
<th>Min: Resets: 0</th>
<th>Uptime: 90080</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>RT</td>
<td>MIN</td>
<td>MAX</td>
<td>RT</td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>Temperature (°C)</td>
<td>41.7</td>
<td>19.9</td>
<td>40.8</td>
<td>85.6</td>
<td>41.4</td>
<td>65.4</td>
</tr>
<tr>
<td>Spacecraft Spin (rpm)</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery I2C</td>
<td>FAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSU1 I2C</td>
<td>OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSU2 I2C</td>
<td>OK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground Reset</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic Info</td>
<td>spln: -9357603</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard Error</td>
<td>wc 0 ec 1 nr 0 af 0 in...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soft Error</td>
<td>dac235 i2c 13 spl 0...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery</td>
<td>RT</td>
<td>MIN</td>
<td>MAX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cell A + B + C (V)</td>
<td>4.02</td>
<td>3.94</td>
<td>4.50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Experiments

<table>
<thead>
<tr>
<th>PSU</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current (mA)</td>
<td>6.3</td>
<td>5.9</td>
<td>9297.7</td>
</tr>
<tr>
<td>Board Temp (°C)</td>
<td>43.8</td>
<td>21.1</td>
<td>43.9</td>
</tr>
</tbody>
</table>

#### +X Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.7</td>
<td>41.4</td>
<td>-17.1</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>3.8</td>
<td>0.1</td>
<td>5.6</td>
</tr>
<tr>
<td>Rotation (GPS)</td>
<td>-11.0</td>
<td>-4.4</td>
<td>29.2</td>
</tr>
</tbody>
</table>

#### +Y Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1</td>
<td>3935.0</td>
<td>-14.6</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>1.7</td>
<td>0.0</td>
<td>5.4</td>
</tr>
<tr>
<td>Rotation (GPS)</td>
<td>19.9</td>
<td>-26.0</td>
<td>26.4</td>
</tr>
</tbody>
</table>

#### +Z Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.8</td>
<td>42.5</td>
<td>-18.1</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>3.8</td>
<td>0.0</td>
<td>5.2</td>
</tr>
<tr>
<td>Rotation (GPS)</td>
<td>-11.7</td>
<td>-27.9</td>
<td>26.2</td>
</tr>
</tbody>
</table>

#### -X Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.5</td>
<td>45.3</td>
<td>-17.5</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>3.6</td>
<td>0.0</td>
<td>5.6</td>
</tr>
</tbody>
</table>

#### -Y Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.2</td>
<td>44.4</td>
<td>-14.8</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>3.9</td>
<td>0.0</td>
<td>5.6</td>
</tr>
</tbody>
</table>

#### -Z Panel

<table>
<thead>
<tr>
<th>Temp (°C)</th>
<th>RT</th>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.8</td>
<td>43.1</td>
<td>-14.5</td>
<td></td>
</tr>
<tr>
<td>Voltage (V)</td>
<td>1.7</td>
<td>0.0</td>
<td>5.6</td>
</tr>
</tbody>
</table>
Client Software Examples – COSMOS!

COSMOS is a software suite used for various scientific applications. The image shows a screenshot of the COSMOS interface, including monitoring and adjusting parameters such as temperature and light levels. The software interface includes graphs and data logs for real-time monitoring of various conditions.
**Client Software – The Point!**

Satellite XYZ
Telecommand (TC) And Telemetry (TM) Software

AX.25/KISS Frames
Come Out (TX/Uplink) and Go In (RX/Downlink)
KISS Protocol
Keep It Simple, Stupid (KISS) Protocol

- “KISS (keep it simple, stupid) is a protocol for communicating with a serial terminal node controller (TNC) device used for amateur radio. This allows the TNC to combine more features into a single device and standardizes communications. **KISS was developed** by Mike Chepponis and Phil Karn to allow transmission of AX.25 packet radio frames containing IP packets **over an asynchronous serial link**, for use with the KA9Q NOS program.”  -- [https://en.wikipedia.org/wiki/KISS_(TNC)](https://en.wikipedia.org/wiki/KISS_(TNC))
KISS Protocol – Only 3 Extra Bytes!

Special KISS Characters

<table>
<thead>
<tr>
<th>Hex value</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xC0</td>
<td>FEND</td>
<td>Frame End</td>
</tr>
<tr>
<td>0xDB</td>
<td>FESC</td>
<td>Frame Escape</td>
</tr>
<tr>
<td>0xDC</td>
<td>TFEND</td>
<td>Transposed Frame End</td>
</tr>
<tr>
<td>0xDD</td>
<td>TFESC</td>
<td>Transposed Frame Escape</td>
</tr>
</tbody>
</table>

Why Special Characters?:

- FEND Used to Mark Start/Stop of a KISS Frame (kind of like HDLC Flag bytes)
- If the FEND or FESC codes appear in the data to be transferred, they need to be escaped.
- If DATA contains:
  - FEND $\rightarrow$ FESC, TFEND replaces in byte stream
  - FESC $\rightarrow$ FESC, TFESC replaces in byte stream
- (kind of like HDLC bit stuffing, but at the byte level)
FEND (0xC0) is used to mark the start and stop of a FRAME.
### KISS Protocol – COMMAND Byte

<table>
<thead>
<tr>
<th>FEND (1 Byte)</th>
<th>COMMAND (1 Byte)</th>
<th>DATA (0-N Bytes)</th>
<th>FEND (1 Byte)</th>
</tr>
</thead>
</table>

**Port Index (Hi Nibble*)** | **CMD (Lo Nibble*)**

**Commands for control of TNC**

**Port Index of TNC**

0 | 1

---

**FEND**

1 Byte

**COMMAND**

1 Byte

**DATA**

0-N Bytes

**FEND**

1 Byte

---

### KISS Command Codes

<table>
<thead>
<tr>
<th>Hex value</th>
<th>Name</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Data frame</td>
<td>Varies</td>
<td>This frame contains data that should be sent out of the TNC. The maximum number of bytes is determined by the amount of memory in the TNC. <strong>This is the only allowed command code for a RECEIVED frame</strong></td>
</tr>
<tr>
<td>0x01</td>
<td>TX DELAY</td>
<td>1</td>
<td>The amount of time to wait between keying the transmitter and beginning to send data (in 10 ms units).</td>
</tr>
<tr>
<td>0x02</td>
<td>P</td>
<td>1</td>
<td>The persistence parameter. Persistence=Data*256-1. Used for CSMA.</td>
</tr>
<tr>
<td>0x03</td>
<td>SlotTime</td>
<td>1</td>
<td>Slot time in 10 ms units. Used for CSMA.</td>
</tr>
<tr>
<td>0x04</td>
<td>TXtail</td>
<td>1</td>
<td>The length of time to keep the transmitter keyed after sending the data (in 10 ms units).</td>
</tr>
<tr>
<td>0x05</td>
<td>FullDuplex</td>
<td>1</td>
<td>0 means half duplex, anything else means full duplex.</td>
</tr>
<tr>
<td>0x06</td>
<td>SetHardware</td>
<td>Varies</td>
<td>Device dependent.</td>
</tr>
<tr>
<td>0xFF</td>
<td>Return</td>
<td>1</td>
<td>Exit KISS mode. This applies to all ports and requires a port code of 0xF.</td>
</tr>
</tbody>
</table>

---

*NOTE:

1 Nibble = 4 bits
1 Byte = 2 Nibbles = 8 bits
### KISS Protocol – DATA Byte

<table>
<thead>
<tr>
<th>FEND (1 Byte)</th>
<th>COMMAND (1 Byte)</th>
<th>DATA (0-N Bytes)</th>
<th>FEND (1 Byte)</th>
</tr>
</thead>
</table>

**AX.25 Transfer Frame Header (128 bits)**

<table>
<thead>
<tr>
<th>DESTINATION ADDRESS</th>
<th>SOURCE ADDRESS</th>
<th>Control Bits</th>
<th>Protocol Identifier</th>
<th>Information Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>56</td>
<td>8</td>
<td>8</td>
<td>0-2048</td>
</tr>
</tbody>
</table>

Client Software Outputs/Inputs The AX.25 Frame inside the KISS DATA Field

Note ONLY the AX.25 Frame, NOT the HDLC pieces, are inside the KISS Frame
AX.25/HDLC Protocol
AX.25 Protocol - Overview

• The ‘A’ means ‘Amateur’
• Layer 2 (Link Layer) Specification.
• Derived from HDLC (flags, framing, bit stuffing, NRZ-I encoding)
• Most common amateur radio packet protocol
• Relatively large protocol, with only a small subset of details relevant to satellite
• References:
  • https://www.tapr.org/pdf/AX25.2.2.pdf ← Full AX.25 specification (July 1998, almost 20 years old!)
  • https://www.qb50.eu/index.php/tech-docs/category/17-up-to-date-docs
    • 11- QB50-EPFL-SSC-SCS-ICD-AX.25-TFF-3-1.pdf ← This one describes the relevant pieces of AX.25 for satellite work
  • http://destevez.net/2016/06/kiss-hdlc-ax-25-and-friends/ ← additional useful details and mapping to Physical layer (AFSK, GMSK, etc.)
• Octets (Bytes) are sent Least Significant Bit (LSB) first:

```
<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
```

• Frames are sent Most Significant Byte First (Flag, then Dest, then Src, .... One exception!)

• All fields in between flags are subject to HDLC bit stuffing (more later)

• Satellites use AX.25 Unnumbered Information Frame (UI-Frame)

• UI-Frames are Connectionless

• AX.25 UI-Frame Structure:

<table>
<thead>
<tr>
<th>FLAG</th>
<th>AX.25 Transfer Frame Header (128 bits)</th>
<th>Information Field</th>
<th>Frame Check Sequence</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>DESTINATION ADDRESS</strong></td>
<td><strong>SOURCE ADDRESS</strong></td>
<td><strong>Control Bits</strong></td>
<td><strong>Protocol Identifier</strong></td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>56</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

3/23/2018 TNC Interfacing Tutorial
### AX.25 Protocol – Flag Field

<table>
<thead>
<tr>
<th>Flag (8 bits)</th>
<th>0 1 1 1 1 1 1 0</th>
</tr>
</thead>
</table>

#### AX.25 Transfer Frame Header (128 bits)

<table>
<thead>
<tr>
<th>Flag</th>
<th>Destination Address</th>
<th>Source Address</th>
<th>Control Bits</th>
<th>Protocol Identifier</th>
<th>Information Field</th>
<th>Frame Check Sequence</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>56</td>
<td>56</td>
<td>8</td>
<td>8</td>
<td>0-2048</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

- **Flag (8 bits)**: 01111110 (0x7E)
- **Technically this is NOT part of the AX.25 Frame, its part of the HDLC encapsulation (a detail but important for implementers)**!

- Used to mark beginning and end of frame
- Helps receiver synchronize
- Usually many are sent before and after frame
- Not subject to HDLC Bit Stuffing (more later)
- Fixed Value: 01111110 (0x7E)
### AX.25 Transfer Frame Header (128 bits)

<table>
<thead>
<tr>
<th>FLAG</th>
<th>DESTINATION ADDRESS</th>
<th>SOURCE ADDRESS</th>
<th>Control Bits</th>
<th>Protocol Identifier</th>
<th>Information Field</th>
<th>Frame Check Sequence</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>56</td>
<td>56</td>
<td>8</td>
<td>8</td>
<td>0-2048</td>
<td>16</td>
<td>8</td>
</tr>
</tbody>
</table>

**DESTINATION ADDRESS**

<table>
<thead>
<tr>
<th>C1 (8 bits)</th>
<th>C6 (8 bits)</th>
<th>SSID (8 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X X X X X X X 0</td>
<td>...</td>
<td>X X X X X X X X 0</td>
</tr>
</tbody>
</table>

**SOURCE ADDRESS**

<table>
<thead>
<tr>
<th>C1 (8 bits)</th>
<th>C6 (8 bits)</th>
<th>SSID (8 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X X X X X X X X 0</td>
<td>...</td>
<td>X X X X X X X X 0</td>
</tr>
</tbody>
</table>

- 6 Callsign octets + 1 SSID Octet
- 6 Callsign octets are 7 bit ASCII, but left shifted one bit.
- Last bit of each octet indicates whether there is more data or not:
  - 0 = More Address Data
  - 1 = End of Address Data

- SSID[7:5] fixed bits → 011
- SSID[4:1] → 16 bit integer value, usually 0000

- **NOTE:** 2 to 4 addresses are allowed in AX.25; Destination and Source are required with up to 2 additional ‘paths.’ Usually, for satellites, only the source and destination fields are present. A notable exception to this are the APRS repeater satellites (ISS, PSAT, NO-44, etc.) that include additional ‘digipeat paths’ (According to the APRS protocol).
AX.25 Protocol – Control and Protocol ID

<table>
<thead>
<tr>
<th>FLAG</th>
<th>AX.25 Transfer Frame Header (128 bits)</th>
<th>Information Field</th>
<th>Frame Check Sequence</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>DESTINATION ADDRESS</td>
<td>SOURCE ADDRESS</td>
<td>Control Bits</td>
<td>Protocol Identifier</td>
</tr>
</tbody>
</table>

Control (8 bits)
0 0 0 0 0 1 1

Protocol ID (8 bits)
1 1 1 1 0 0 0

Control Bits:
• Indicates what type of AX.25 Frame is being sent
• Unnumbered Information Frame → Shall be fixed: **0000011 (0x03)**

Protocol Identifier:
• Indicates what type of layer 3 protocol
• No layer 3 protocol implemented → Shall be fixed: **11100000 (0xF0)**
AX.25 Protocol – Information Field

<table>
<thead>
<tr>
<th>FLAG</th>
<th>AX.25 Transfer Frame Header (128 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DESTINATION ADDRESS</td>
</tr>
<tr>
<td>8</td>
<td>56</td>
</tr>
</tbody>
</table>

• Integer multiple of octets
• Maximum Length of 256 octets
• Contains the DATA!!!
  • Telecommand (TC)
  • Telemetry (TM or TLM)
  • Mission Data
• This is where higher layer definitions come into play. QB50 for example elected to use CCSDS packet formats within the Information Field. APRS is an ASCII based protocol that defines this field for APRS messages. **This can be completely custom, but MUST be defined (ICDs recommended)!**
AX.25 Protocol – Frame Check Sequence

<table>
<thead>
<tr>
<th>FLAG</th>
<th>AX.25 Transfer Frame Header (128 bits)</th>
<th>Information Field</th>
<th>Frame Check Sequence</th>
<th>FLAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>DESTINATION ADDRESS</td>
<td>SOURCE ADDRESS</td>
<td>Control Bits</td>
<td>Protocol Identifier</td>
</tr>
</tbody>
</table>

- 16-CCITT based CRC (Cyclic Redundancy Check)
- Used to detect bit errors (doesn’t correct them!, some limited tricks are possible)
- Usually if errors are detected packet is discarded and not passed up the OSI stack for higher level processing.
- NOTE: The FCS is sent least significant byte FIRST!!!! This can be confusing for implementers.
- Technically this is NOT part of the AX.25 Frame, its part of the HDLC encapsulation (a detail but important for implementers)!
HDLC (AX.25) Protocol – Bit Stuffing

• Flags are special...it's how the receiver detects an AX.25 frame in the bit stream.

• It is entirely possible (likely) that 6 consecutive 1s will appear in the data stream of an AX.25 frame. If nothing is done about this, then a false flag detection will occur, resulting in a failed CRC check, resulting in a discarded frame.

• Bit Stuffing to the rescue!
  • As the AX.25 frame is being sent into the modulator, the number of consecutive 1s is monitored.
  • If 5 consecutive 1s are detected in the bit stream, a 0 is inserted.
  • On the receiver side, if 5 consecutive 1s are detected and then the next bit is a 0, the 0 is discarded from the bit stream (‘un-stuffed’). If the next bit is a 1, the receiver should expect a following 0, indicate a FLAG has been received.
The Use of Non Return to Zero Inverted (NRZ-I) means that modulations don’t care about the actual symbol state (say a positive or negative frequency). What matters is the change or lack of change of a symbol to represent the bit value.

NOTE: HDLC/AX.25 uses an inverted form of NRZ-I.

- 0 → Transition
- 1 → No Transition
AFSK & FSK/GMSK Modulation
AFSK and FM Modulation

• AFSK = Audio Frequency Shift Keying
  • Uses audio tones at 1200 Hz (Space) and 2200 Hz (Mark).
    • Derived from Bell 202 dialup modem standard, which is why a ‘burst’ sounds like an old modem.
• NRZ-I bits are used to alternate (or not) between the tones.
• Continuous Phase Frequency Modulation used to generate AFSK
  • No instantaneous phase change → reduced signal bandwidth.
• Audio is then piped into an FM transmitter.
G3RUH FSK/GMSK Modulation

- FSK = Frequency Shift Keying
- GMSK = Gaussian Minimum Shift Keying
- Modulating signal (the ‘message’) bypasses the FM pre-emphasis/de-emphasis circuitry of hardware radio and is applied directly to the FM Modulator (varactor) to produce FSK.
- If the NRZ-I signal is passed through a Gaussian pulse shaping filter before it is sent to the radio then GMSK is produced.
- More on next slide for pulse shaping.
GMSK and Pulse Shaping Filter – more details

- Pulse shaping smooths out instantaneous transitions, which reduces side lobes in the produced signal.
- Rolloff factor of pulse shaping filter controls ‘smoothness’ of curves. Is between 0 and 1 where 0 is a rectangular filter (i.e. no different than no filtering). Sometimes called $\beta$, sometimes $\alpha$, sometimes BT.
- For Amateur Satellites, BT = 0.5 is common.
Summary of process so far......

Uplink Only Scenario (for simplicity)
Agnostic to what type of TNC/Radio interface is used
1. Client Software generates a properly formatted AX.25 Frame with Telecommand in Information Field.
2. Client Software *encapsulates* the AX.25 Frame inside the DATA field of a KISS Frame.
3. Client Software writes the KISS Frame to the TNC via Serial or TCP/IP.

### AX.25 Transfer Frame Header (128 bits)

<table>
<thead>
<tr>
<th>Field</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESTINATION ADDRESS</td>
<td>8</td>
</tr>
<tr>
<td>SOURCE ADDRESS</td>
<td>8</td>
</tr>
<tr>
<td>Control Bits</td>
<td>8</td>
</tr>
<tr>
<td>Protocol Identifier</td>
<td>8</td>
</tr>
<tr>
<td>Information Field</td>
<td>0-2048</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame Components</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEND (1 Byte)</td>
<td></td>
</tr>
<tr>
<td>COMMAND (1 Byte)</td>
<td></td>
</tr>
<tr>
<td>DATA (0-N Bytes)</td>
<td></td>
</tr>
<tr>
<td>FEND (1 Byte)</td>
<td></td>
</tr>
</tbody>
</table>
4. TNC Receives the KISS FRAME (Serial or TCP), removes/replaces escaped bytes if necessary, discards the FEND Bytes
5. The command byte is passed off for TNC parameter control and port routing. It is REMOVED from the byte stream.
6. The DATA field, which contains an AX.25 frame, is extracted and passed off to HDLC steps
7. HDLC Processing receives the AX.25 Frame
8. Frame Check Sequence is computed and appended to the Frame, Bit stuffing occurs (not depicted)
9. FLAGS are appended and prepended to the Frame (usually many flags on both ends)
10. Bit stream of frame is line encoded (HDLC NRZ-I), Most significant byte first (except FCS, flipped), LSB first
11. HDLC NRZ-I encoded bit stream is input to the AFSK Modulator; CPFSK modulation is used to produce an output AFSK signal.

12. TNC triggers push to talk (PTT) line to place radio in transmit mode; AFSK Signal is passed into the DATA IN port of the radio on the DATA JACK (packet mode 1200 tap point).

13. AFSK signal is sent through the FM Modulation process and radiated out of the radio.
Open Systems Interconnect Model
Open Systems Interconnect (OSI) Stack

• Devices logically communicate at the relevant layers
• Encapsulation
• Headers
• Standards (for interchangeable layers)
OSI Stack – Simple Web Browsing Example

1. Physical
   - Bits

2. Data Link
   - Frames

3. Network
   - Segments

4. Transport
   - Packets

5. Session
   - Data

6. Presentation
   - Data

7. Application
   - Data

HTTP (Web Browser)

TCP

IP

MAC

SWITCHES

COPPER

FIBER

RF

ROUTERS
Before We Proceed—a simplification

This is a common simplification which we will use for the rest of this presentation
OSI Stack Logical Communication

Layer | Data | Layer
---|---|---
Application | Application Data | Application
Transport | Segmented Application Data | Transport
Network | L3 Header | Network
Data Link | L4 Header | Data Link
Physical | L2 Header | Physical

Physical Medium
OSI Stack – Smallsat Comms Example

Data Layer

Application

Data

Segments

Transport

Packets

Network

Frames

Data Link

Bits

Physical

Client Application

TCP

IP

VPN

MAC /KISS/AX.25/HDLC

RS-232/AFSK/FSK/GMSK

TNC

Satellite TC/TM Software

3/23/2018

TNC Interfacing Tutorial
Remote Connection
Bring it all together!
Information overload---what’s the point?

- TCP
- IP
- VPN
- KISS
- HDLC
- AX.25
- AFSK
- OSI
- G3RUH GSMK/FSK

Once again.............
OSI Stack for AFSK Uplink – Hardware TNC

Host Computer

HW TNC

HW Radio

Antennas

Ground Station

Host Computer

Client Application

TC SW

AX.25/KISS

SERIAL

RS-232

Audio/NRZ

PTT

Telecommand (TC)

Spacecraft

RADIO

RADIO FIRMWARE

HDLC/AX.25

KISS

RF/FM/AFSK

SERIAL

C&DH Computer

TC SW

KISS/AX.25

SERIAL

TNC FIRMWARE

AX.25/HDLC

AFSK

DATA IN

FM / RF

Radio

TTL Serial

Serial

AX.25/KISS

Audio

3/23/2018

TNC Interfacing Tutorial
Slight tangent - If you are still developing/testing

This is the power of adhering to the OSI Model!
OSI Stack for AFSK Uplink – Software TNC

Host Computer

SoundCard Interface

HW Radio

Antennas

Ground Station

Host Computer

Software TNC

HOST OS NETWORK

Spacecraft

C&DH

Compu
er

Radio

Telecommand (TC)

Audio

TNC SW

KISS

TC SW

AX.25

KISS

TCP

LOCALHOST

HOST OS NETWORK

SoundCard Interface

D to A

USB / Serial

‘audio’ samples

Audio/NRZ

PTT

Radio

Serial

FM

AFSK

HDLC

TCP

IP

Data

IN

FM

RF

RADiO

FIRMWARE

TC SW

AX.25

KISS

HDLC

AFSK

SERIAL

TTL Serial

Audio

3/23/2018

TNC Interfacing Tutorial

76
OSI Stack for AFSK Uplink – Remote Connection
NOTE: the VTGS is actually a bit more complicated than shown below. For simplicity related to OSI stack, copy of SW TNC + Soundcard GS is shown.
Summary

The Key things to remember
Summary – Key points to remember

• Telemetry (TM) processing and Telecommand (TC) generation software specific to the satellite must be written if it does not already exist.

• The structure of the of the Information Field for TC/TM must be defined (ICDs).

• The TC/TM Software must output an AX.25 Frame encapsulated in a KISS Frame for uplink and expect to receive a KISS Frame containing an AX.25 Frame from downlink.

• It is HIGHLY recommended that the TC/TM software utilize TCP/IP sockets for transport of KISS Frame. This interface is common with software TNCs and is necessary for the Internet based remote connection to the VTGS.

• I am NOT an expert on the IC-9100. I’ve given some pointers on the cable fabrication and testing process…..but double checking functionality and configurations specific to the IC-9100 (Menu settings) need to be done at UPEC.

• This presentation is pretty specific to TNC Interfacing (with relevant deviations). Topics not addressed for full ground station design and implementation would have resulted in at least double the slide deck length...(link analysis, antenna design, noise characterization, amp design, tracking, sequencing, message passing, data storage, metadata generation, etc. etc.).......
Thank You!

And remember…..Don’t Panic!…..It’s all gonna work!