

APRStt Interface for SARTrack

Rough Draft 2 - WB2OSZ – July 24, 2015

Very few hams have portable equipment for APRS but nearly everyone has a handheld radio that can send DTMF tones. The APRStt concept allows a user, equipped with only DTMF (commonly known as Touch Tone) generation capability, to enter information into the global APRS data network.

This example illustrates how this capability can be integrated with other applications such as SARTrack, APRSISCE/32, YAAC, or Xastir.

In this hypothetical scenario, a 5 year old girl with a pink Red Sox cap, wandered off from a family picnic and disappeared into the forest. A hastily assembled search group has a couple APRS trackers and a laptop running SARTrack to track their locations. Teams without special APRS equipment will use APRStt to report their locations.

The area of interest is covered by UTM coordinates of this form:

19T 30xxx0 472yyy0

In this specific situation it was agreed that only the xxx and yyy digits would be sent. The 3 digit easting (xxx) and northing (yyy) values cover a 10 km square region with 10 meter resolution. Single digits were assigned to the search teams:

1 = Team A
2 = Team B
3 = Team C

Search teams will report their positions by sending a 7 digit sequence of the form:

team-number location-easting location-northing #

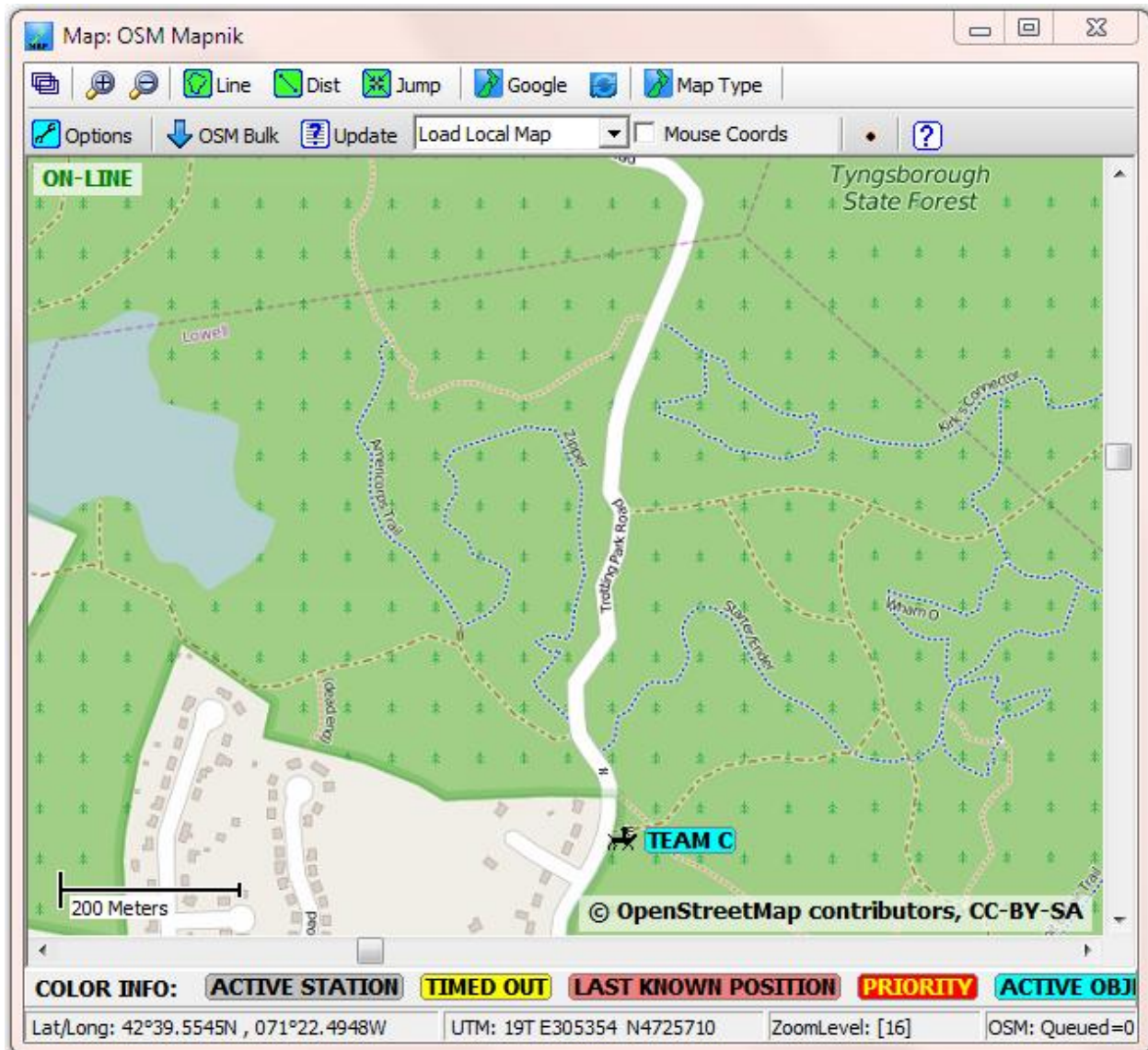
If Team C was at location 19T 305710 4725310 this would be reported by sending the tone sequence:

3 571 531 #

They hear a voice response, confirming that the transmission was received correctly:

“Team Charlie 5 7 1 east 5 3 1 north.”

The APRStt software decodes the tones, constructs an APRS “Object Report” and sends it to the attached application, SARTrack in this example. Team C appears on the map.



Team C found a pink Red Sox cap. A clue! It was also agreed at the beginning that a single digit could be used to report a status. Among them,

- 5 = Clue found
- 6 = Subject found

In this case, they can send a single digit field for the status and a 7 digit field for the team and location:

5 * 3 544 583 #

The voice response includes the new status:

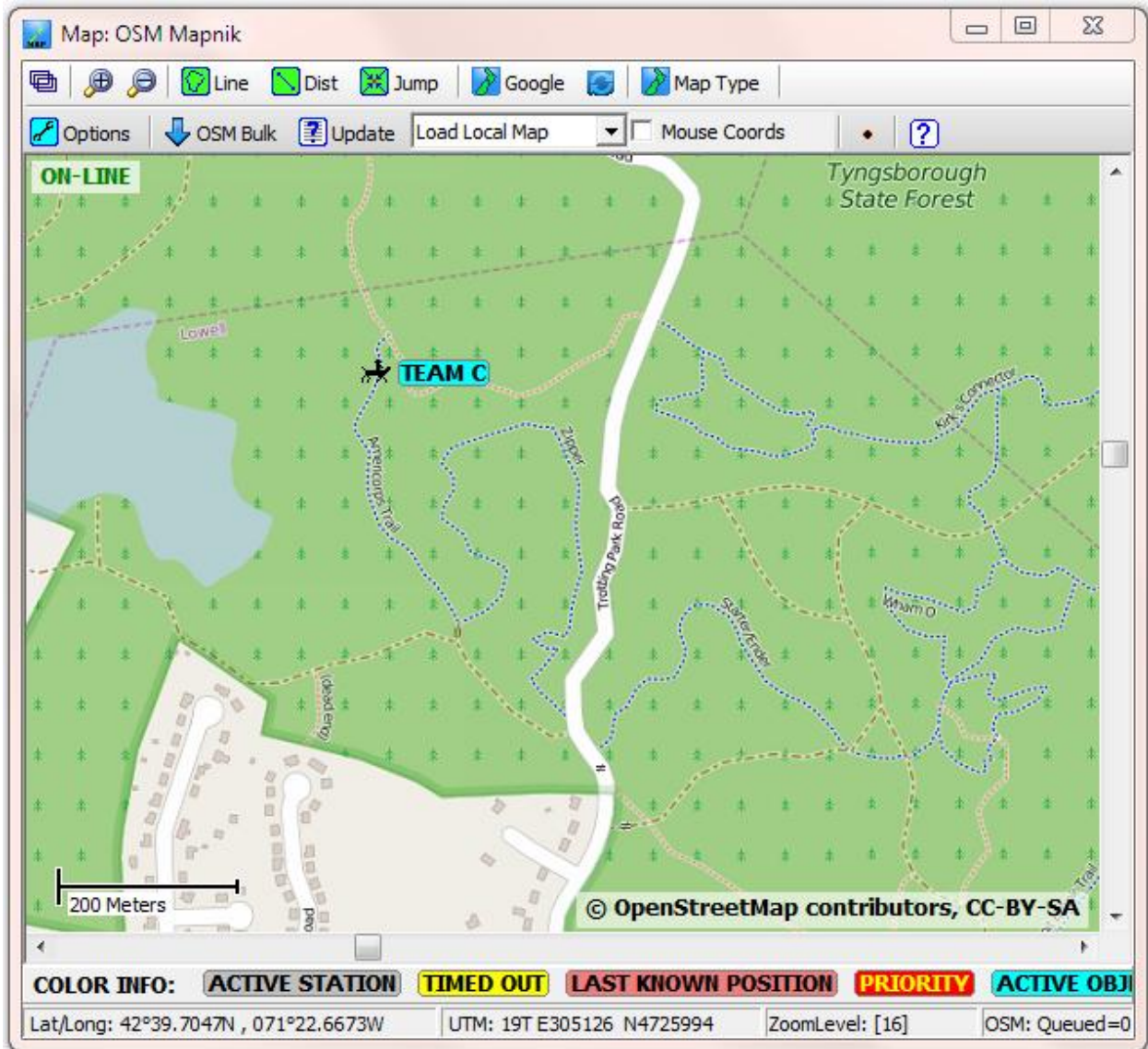
“Team Charlie 5 4 4 east 5 8 3 north. Clue found.”

The Dire Wolf display includes the APRS Object Report which was generated.

```
WB2OSZ-13>APDW13:;TEAM C *151247z4239.62N/07122.43We[19T 305440  
4725830] /Clue found !TB6!
```

Notice how the status is part of the comment, preceded by a slash.

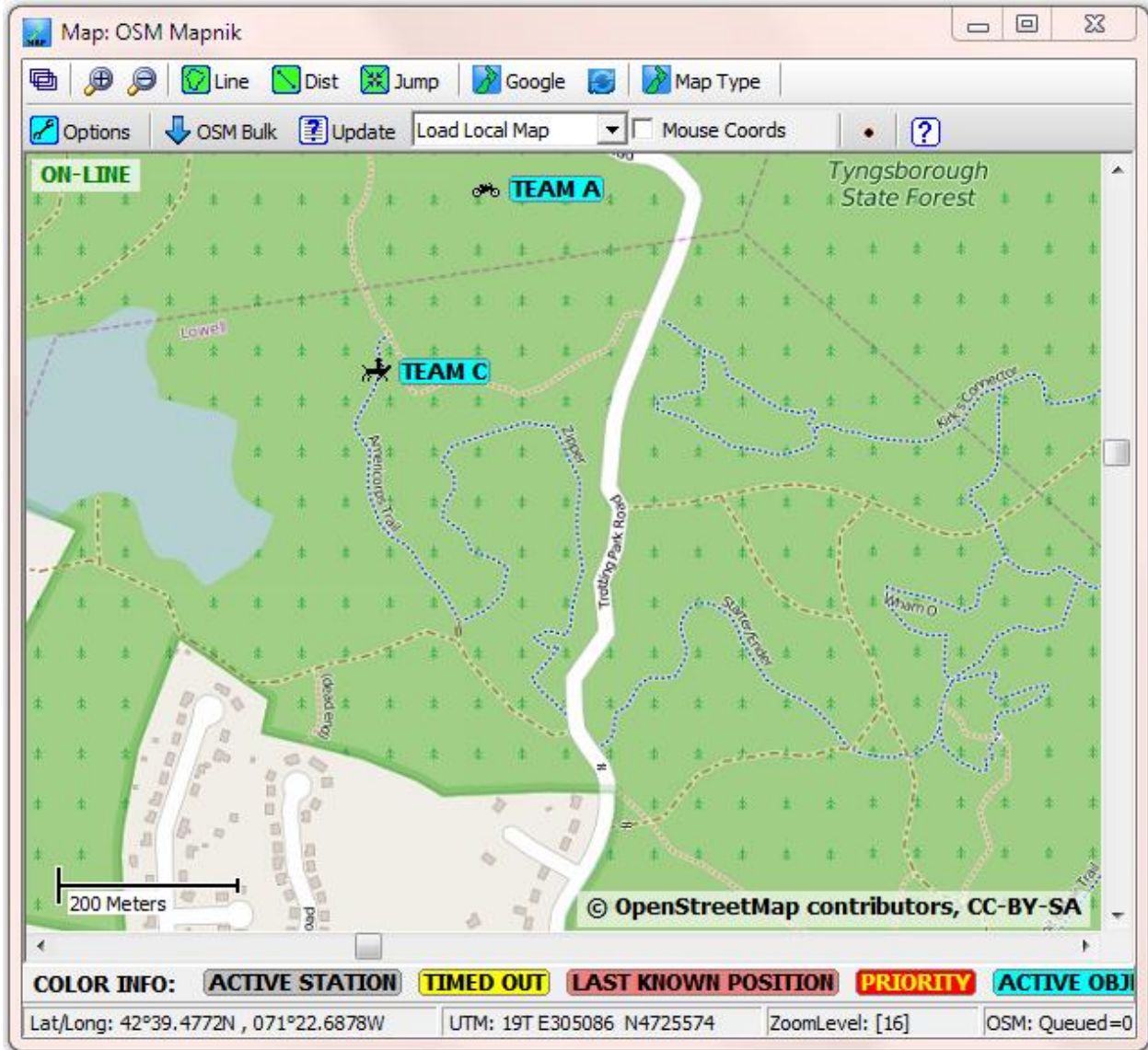
This new location appears in SARTrack.



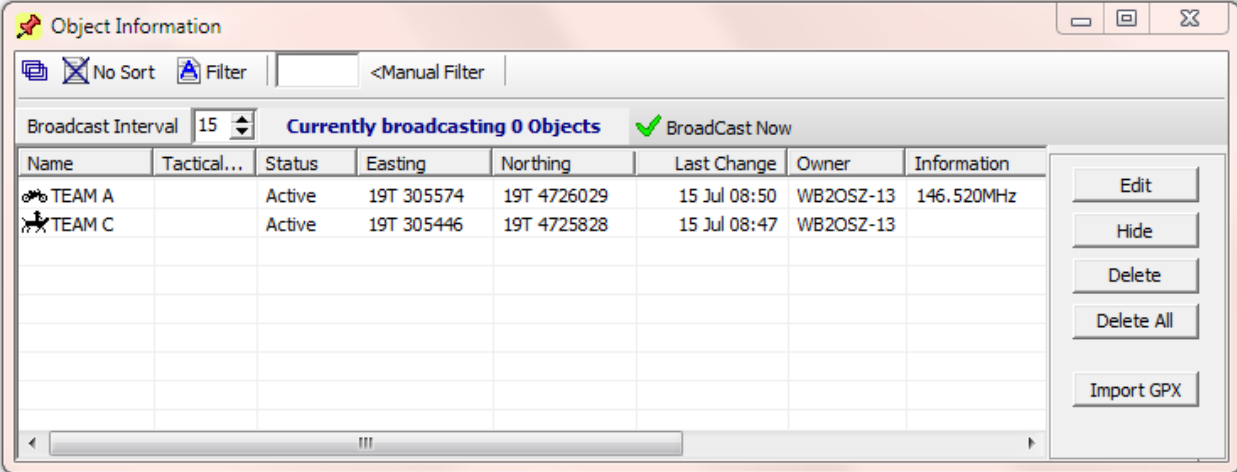
Team A finds the lost girl, safe, and happily watching rabbits nibbling clover. They also want to switch to a different voice frequency due to interference. The APRStt representation for this is the C button followed by exactly 6 digits. They combine all of this into a single transmission with 3 fields.

6 * 1 558 603 * C146550 #

They hear the usual voice confirmation the transmission was received and their information is updated in SARTrack.



Object information is also available in table format:



The screenshot shows a software window titled "Object Information". At the top, there are window control buttons (minimize, maximize, close) and a toolbar with icons for "No Sort" and "Filter". Below the toolbar, there is a "Broadcast Interval" dropdown set to "15", a status indicator "Currently broadcasting 0 Objects" with a green checkmark, and a "BroadCast Now" button. The main area contains a table with the following columns: Name, Tactical..., Status, Easting, Northing, Last Change, Owner, and Information. Two rows are visible: "TEAM A" and "TEAM C", both with a status of "Active". To the right of the table is a vertical stack of buttons: "Edit", "Hide", "Delete", "Delete All", and "Import GPX".

Name	Tactical...	Status	Easting	Northing	Last Change	Owner	Information
TEAM A		Active	19T 305574	19T 4726029	15 Jul 08:50	WB2OSZ-13	146.520MHz
TEAM C		Active	19T 305446	19T 4725828	15 Jul 08:47	WB2OSZ-13	

The incident described here was fictional but the software is quite real and available now at no cost.

Don't just read about it. Try it for yourself! Start with the sample configuration provided and try modifying it for your own requirements.

Hardware Required

- Laptop computer, Microsoft Windows in this case, as required by SARTrack.
- USB Audio adapter if laptop does not have audio in & out jacks.
- Ham radio transceiver. Enable the VOX so we don't need separate PTT wiring.
- Cable connecting them together.

Audio out from the radio goes into the PC microphone input.

Audio out from the PC goes to the radio microphone input. If level is too high you might want to add a voltage divider with two resistors.

Software Required

- Dire Wolf <https://home.comcast.net/~wb2osz/site/>

The example here uses some new features in version 1.3 which has not yet been released at the time this is being written. Download it from this hidden location:

Windows, prebuilt:

<http://home.comcast.net/~wb2osz/Version%201.3/direwolf-1.3-dev-D-win.zip>

Source for Linux:

<http://home.comcast.net/~wb2osz/Version%201.3/direwolf-1.3-dev-D-src.zip>

- SARTrack <http://www.sartrack.co.nz/>

In this example we used SARTrack. You could also use APRISCE/32, YAAC, Xastir, etc. Connect to Dire Wolf with AGW network protocol or KISS.

- eSpeak <http://espeak.sourceforge.net/download.html>

- perl <http://strawberryperl.com/>

This is required to run the provided script that composes the speech response based on name, location, status, etc. You could do the same thing in any other language that can examine environment variables and produce text output.

Dire Wolf Configuration File

The following is a complete configuration file for the scenario above. The transmission format of 1 digit for the team and 6 digits for location is not hardcoded into the application. It is completely configurable.

Paste the rest of this section into a file, such as sar.conf, and then run Dire Wolf pointing to this file.

```
direwolf -c sar.conf

# Set callsign for first radio channel -- Use your own call!!!

CHANNEL 0
MYCALL WB2OSZ-13

# Specify script for speech output. See User Guide for more details.
# A sample dwespeak.bat is included with the distribution.
# Use dwespeak.sh instead if using Linux.

SPEECH dwespeak.bat

# Define UTM coordinate convention used for this incident.
# All locations can fit in the range 19T 30xxx0 472yyy0.
# When B6xxxxyy is received,
#   xxx is multiplied by 10 and added to 300000 to get the easting value.
#   yyy is multiplied by 10 and added to 4720000 to get the northing value.

TTUTM B6xxxxyy 19 10 300000 4720000

# Define macros for reception of 7 digit fields.
# The first digit is the team identifier,
#   1 = Team A, motorcycle icon
#   2 = Team B, canoe icon
#   3 = Team C, horse icon
# The other 6 digits are the UTM location as described above.

TTMACRO 1xxxxyy "AA{Team A}*AB{motorcycle}*B6xxxxyy"
TTMACRO 2xxxxyy "AA{Team B}*AB{canoe}*B6xxxxyy"
TTMACRO 3xxxxyy "AA{Team C}*AB{horse}*B6xxxxyy"

# Define macro for reception of single digit field.
# Override default definitions for status codes 5 & 6.

TTMACRO z Cz
TTSTATUS 5 "Clue found"
TTSTATUS 6 "Subject found"

# We want speech response for any valid tone sequences received.
# Keep the default Morse code "?" for any errors.

TTERR OK SPEECH "Message Received O K. "
```

```
# Invoke the sar.pl script when a valid tone sequence is received.
# The current name, location, status, etc. are passed in thru
# environment variables.  If the script produces text output, that
# is used for the response in place of the string set above.
# For Linux, use "TTCMD sar.pl" without specifying the perl location
# and be sure to "chmod +x sar.pl" to make it executable.
```

```
TTCMD c:\strawberry\perl\bin\perl.exe sar.pl
```

```
# Finally, we enable the APRStt conversion.
# This means enable the DTMF decoder on channel 0.
# Send the result to any attached applications.
# Other possibilities are transmitting a packet over the radio or
# sending information thru the IGate.
```

```
TTOBJ 0 APP
```


Script to Generate Custom Speech Responses

This script gathers information about object name, location, status, etc. and uses this to compose a custom speech response to the tone sequence.

Paste the rest of this section into a file, such as sar.pl. This same file needs to be mentioned in the earlier configuration file. e.g.

```
TTCMD c:\strawberry\perl\bin\perl.exe sar.pl
```

```
#!/usr/bin/perl

# Simple script to generate custom speech responses for APRStt.
# First, gather information of interest.

$name = $ENV{TTCALL};
$loc = $ENV{TTLOC};
$status = $ENV{TTSTATUS};

# Change pronunciation of team names.

if ($name eq "TEAM A") { $name = "Team Alpha"; }
if ($name eq "TEAM B") { $name = "Team Bravo"; }
if ($name eq "TEAM C") { $name = "Team Charlie"; }

# Extract the desired digits from the location.

($zone,$east,$north) = split " ", $loc;
$east = substr("000000" . $east, -4, 3);
$east =~ s// /g;
$north = substr("000000" . $north, -4, 3);
$north =~ s// /g;

# Combine into string to be spoken.

print "$name, $east east, $north north. $status\n";
```

Summary

This simple example demonstrates how the Dire Wolf implementation of the APRStt concept can be used to receive Touch Tone sequences and convert them to a format usable by many other existing APRS applications.

Macro definitions (which evolved from earlier “runner mode” discussions) allow you to define short numeric fields which are internally expanded into the longer standard representation before the final processing. For example single digits can be used to identify teams instead of spelling them out like AA8A3B2A6A0A2A.

Custom speech responses can be composed by a user-supplied script which can make decisions based on callsign / object name, location, status, etc.

A more complete configuration might include two more audio interfaces and two more radios, one for the national APRS frequency and another for a local event frequency. Dire Wolf can digipeat between them with different packet filtering rules for each direction. If Internet connectivity is available, it can also serve as an Internet Gateway (IGate) between the RF channels and global servers.

The APRStt concept has been around since the turn of the Century but saw usage in only a few limited situations due to the need for specialized hardware. This implementation provides a new level of functionality and flexibility which will allow the APRStt concept to be used in many new situations by changing only a configuration file or two.