GRITTY 1.2

iw1ayd Salvatore “Salvo” Irato

18 Apr 2015

GRITTY 1.2

Integrating with N1MM+, general usage report.

It u seems a bit too early, to write about GRITTY. It is at the first version/s, DOC is just a short help file. However, the programmer/s intentions seems defined. Thus as an avid RTTY contester, even if with a short experience that doesn’t cover the last 10 years, and a professional IT roles I am spending yours and mine time here.

Why this paper?

We are gaining new decoding tools and we must thanks who is working so hard to make all us able to run it. There was a launch of GRITTY 1.<2 and then the announce of the latest, now, GRITTY V 1.2. This latest message had given also a very interesting Demo page: link with audio utility pointers, wav recorded files and more stuffs on how to test it off line.

Some more informations are into this document on the same site: RTTY Skimmer Server V 1.0 User manual (PDF).

It relay deeply descriptions of the RTTY Skimmer Server that are part of GRITTY itself. As inherited by this last from the big brother. I.E. the Contest.ini format and tokens, hereafter just summarily presented and shown more than described.

All this is great, also considering that it is a pure new technology for us RTTYers. GRITTY will be further integrated into N1MM+ as a Haddon RX only decoder. What is better?

It received some critics on the same email list and the writer responded promptly. Well done … not cherishing anyone.

This is an attempt to connect what’s new and highly needed with today aspects, all the aspects from roots, of what we use.

This reason let me use such Italian heavy metrics to drill down onto showed aiming of GRITTY e the way as it look. It’s my personal view, fell free to comment and ask to me directly.

Summary

GRITTY is an interesting and promising piece of software. It promise to give inside all of us PC some of the new technologies already in use from the newest RTTY Skimmer application. The one before relied on MMVARI integration. This late relies on the lessons learned with CW skimmer. Unluckily also the installation and data view schema of GRITTY are prone to it’s big parent. But GRITTY is aimed for standard users, by now it lacks real integration with N1MM+ logger and worst it is not made to be portable. Better it couldn’t be made portable. Each installation inside each PC must be copied by hands or scripts. This is the worst case for MS to MM setups. But also for who use 2 PC with two radio.

MMTTY and 2Tone are portable. N1MM+ is de facto portable, even if not supported for this feature. GRITTY is not that. Don’t be fooled like it’s “complexity” of an application would prevent portability. It’s just a choice of the programmers to have all the items needed described or to relay onto the code environment as to the carer for that.

By the way, native installation of N1MM+ aren’t also portable from one PC to another. That’s it just taking care to fully install It on every PC; then making on one PC the whole setup; lastly move one single directory in one single shot onto the second or more PC. But, this lack is easily overcome just taking care of some basic rules. So, definitively N1MM+ will all the setups and the needed basic – now those are to be considered basic – decoders and ancillary applications/files could be moved around the whole group of operating posts.

GRITTY is still not able to write and correct data into N1MM+ DI window. Yes it fetch back corrected data for know and configured/described tokens as callsigns and exchanges. But it seems that the ability to have a back and forward capability to write into a window owned by another process is not reachable now. MS development kit shortcomings? It could be, nobody since now do that in short burst, backing the Cursor and rewriting one or more characters. May be it is not a provisioned callable routine from inside the MS development kit of the season.

GRITTY is not plain user configurable at least for the GUI.

GRITTY still is a very new and promising cake, it is young and some of those issues and maybe other that I didn't seen or could imagine may give use a very productive weapon for our contesting arsenal, when solved.

BTW GRITTY is 87% aimed to contester, not casual user. It's code is designed to give the most valuable functions when contesting.

General note

GRITTY have a new technology and aim to make it deployable at user level into our machines.

It is developed around it's parent application. Making it resembling or perhaps only reusing some aspects. But it's aimed to a new group of potential users. Some parts of the new technology are more prominent, maybe as well known by some. Others living just under the hood of its brand new approach to ours world. It is well mixed and somewhat promising. Attempting to outreach the pure filters and bits correlations limits with data, and the meanings of those data, correlations. That's a new approach leaving old fields and coming inside new ones. It represent a great set of changes and could improve a significant slice of our hobby: RTTY Contesting.

Nonetheless, it is a new one and our habits are much well oriented elsewhere. By now, we use, as real time running decoders, two other applications MMTTY and 2Tone. Somebody of us also trust well-known pieces of HW also with some FW well hidden inside. Spacing with those last from pure decoding to, don't read "simply", Lissajous waveforms check.

Every time we face a new decoder we also face ideas on how to check those. Comparing older and newest, as they could be a full solution for all our problems when in Contest. There is not such a solution, more seasoned contester well know this. Others sometime forgot this truth.

However, out of a nowhere place for such a solution we have to establish some criteria and how to meter around those. That is my personal view about all this.

Well before than leaving he place to whoever is much more technical than me I would apply my nerd abilities to say about integration, ease of use, operational capabilities and so on. Think about it as "how is easy to drive this car".

GRITTY was presented as to be integrated into N1MM. Sided by this I was quick and dirty testing it. I personally would not have too much interest in a not integrated decoder. My focus is where the action is. Action components must stay well tied together N1MM Classic and N1MM+ are the living examples.

Human Correlation

We usually apply some kind of correlation whenever we look at things. Some of us better than others. Some of us try to train their self each time in any contest. It is not that unusual to see at least two decoders windows printing together different words, letters, symbols or simply garbage. That last is not all garbage, often there are mangled word/data vital for our hobby, RTTY Contesting just in case. Decoders by their self apply different technologies for the tones decode and, the most recently developed, bits correlation or lost bit recovery. Aiming to write the most of any group of bits that make a character or a symbol of the Baudot, International Telegraph Alphabet No. 2 (ITA2), we use. Leave out the symbol definition for data communication by now. But those decoder doesn't work out the of very simple to very sophisticated bit recognition technique.

Usually we look at the printing cursors, more than one decoder is printing at a time, and decide what is the real strings that we need, catching it in the way as we prefer or we could. If there are fragments we have to use our hands to write down what's the results, whatever we think it is. Then we have other ways, when the relevant data is already inside the field we desire to valorize. Let's say we are looking for a call sign one decoder print RTVWIUP the other print FSA2UR. We, I in this case, already focused well into all that is on the screen and separated those two fragment of strings: focusing a sliding windows. Than we correlate these strings from different sources and decide for VA2UP. Consider that we have A2 and UP fragments in the received strings. Place that extracted callsigns into the proper filed and look for the checkout result made by screening VA2UP string into databases of known callsign. It is there and valid. Some of us, perhaps the more experienced, fastest or all together check also that the time of the day and the aiming of aerials to have more clue. By example a 18:00 UTC on 40m with a low dipole it's much easy that the call we try to recognize is RA2UP and not VA2UP. Than a check with known call DB and discover that RA2UP is not a valid, well known, call. Now we have at least other two ways to get a clue, but those are out of the scope by now.

What we have done? Correlation. Correlations are useful because they can indicate a predictive relationship that can be exploited in practice. We placed together, scavenged, different things that have nothing of clever. Translating garbage to mangled data and extracting others, the more relevant for us, data. We have applied our knowledge of calls and others skills we have to help ourselves getting out informations from a stream of useless characters we had at the start. All this in a very short time hopefully.

With callsigns it's pretty easy. It's not that easy with exchanges: do you have never seen that NR? NR? PSE NR? ONLY. A very frenetical request. Yes, we are able also to correlate sets of characters that contain numbers. But with those we don't have any clue or any further skill to apply when in needs, 599 0030EQ13P, out to make shorter and clever that retry request and compare again with more data. A data base of common exchange doesn't have any common sense as numbers are a volatile part off the exchange. Still we should apply easy rules. We see “   1” one time, may it's a 001. Then it depends from us if just one time is enough or note. Le's say TIUT12WER12 and TOO312PQW, we have almost a 12 in this case as QW is also 12. That's easy, isn't?

It's much more easy in the contest where exchanges are letters … and letters that make sense, State, Names, locators … those last at least are signaling two letters and two digits. Still this make our RX error prone but not having four letters on hands, JNRE is JN35 at least.

What's GRITTY doing to correlate?

GRITTY correlate in a differed ways. Applying information it has about specific callsign, applying informations it has about generic callsigns, applying information that it has about contests specific exchanges schema.

The first is the well known master.dat binary formatted file.

The second is a file containing the most relevant types of callsign.

The third and maybe the more interesting as it is the driver for all the greatest changes that GRITTY introduces at the user level is a file that contain relevant descriptor signatures for any contest exchange that was taken in account. So not all the fancy ways we have seen not for all the contests.

How I think it could work?

It's working applying the signatures as parameters with internally hardcoded rules. Plain ASCII files may have some sort of user modification.

How this not human correlation will work?

Plainly as a finite state machine doing pattern stuffing recognition at your PC speed quite tireless.

GRITTY aiming I

GRITTY have not only those aspects here portrayed. It has much more. It attempt to logically find and discover from late symbols what was the previous set of symbols and thus the right word since it was transmitted and already gone. It try to receive symbols, recognize patterns and back writing what was already received. Thus it rewrite a word that where mistakenly printed - actual signal decoding capabilities - with a word well recognized by its vocabulary of patterns.

To make this it use a description of exchange for any known contest and a list of call signs.

That's out of any alchemy already done with DSP filters and bit correlation. It is the first, as known by me, attempt to post fetch data for our world. Last but not least, it is really the core engine of the pattern recognition, GRITTY apply statistical concepts to make its choice about words or string that it think we must read instead of the mangled ones. There is a page into the CHM, help file for MS Windows, that explain very quickly but effectively it all.

It doesn't relay only on whatever we know was under the hood of our machine. A lot of care must be exercised looking at those new features. But at least what we need is every time the same:

IW1AYD 599-384-384 IW1AYD

GRITTY search into received data with patterns vocabulary like this

```
+ IW1@@
+ IW1@@@
```

And this:

```
[WAE]
CQ1=CQ|0.3
CQ2=TEST|0.3
CQ3=QRZ|0.3
CQ4=WAE|0.1
Exchange1=599[~]nnn<~nnn>|0.9
Exchange3=5NN[~]nnn<~nnn>|0.1
```

(Some vocabulary to interpret this is inside the Contests.ini, also for Patt3Ch.lst that collaborate for callsign structures checkout. Also remember the RTTY Skimmer manual.)

Then apply statistical analysis using the pattern recognition as a sliding window over the test incoming and already received. As a result it could post fetch and correct data already received as garbage applying such correlation schema. Almost as any of us after having some experience could do. What about the differences in between us and GRITTY:

1) GRITTY even if it is sweet have only a prepared set of elements to be correlated

2) GRITTY doesn't become tired or lazy it couldn't, each correlation schema it has into will be applied from 00:00 UTC until the end of the contest the same way

Point number 1 says also that GRITTY have a finite but modifiable set of patterns. We have to make choices about witch contest will be in use. We have to trust the patterns that are already defined for that contest. Or, we have to find ourselves new pattern/s for new contest not already into is table of recognition.

Point 2, wow! But still we still need to stay bottom in the chair, believe it.

There are already several contests, patterns include obviously, into a file used by GRITTY. The short CHM help say Contest.Ini, that reside by installation into:

%appdata%\Afreet\Products\RttySkimServ\Contests.ini"

The call signs are interpreted by:

%appdata%\Afreet\Reference\MASTER.DTA

That is the master.dta well known file. We will have to maintain it by hands. It is in another directory than Contest.ini.

There are also other files. GRITTY.ini is into %appdata%\Afreet\Products\GRITTY and others. All well displaced into at least four different places with just %appdata%\Afreet in common between each.

That's, %appdata%, a structural tree for Mss standing for

"Wherever-disk_letter:\witcheveristheuser\Appdata\Whereveruser-data-for-this-Programs_are_stored\The-Program" and MS could impose its security by Obscurity rules on it.

This is one of the first issue, %appdata%. It usually correspond to

"C:\users\USER-NAME-OF-THE-TIME\Appdata\Roaming"

And it is a hidden and name variable directory. It is changing as change the USER-NAME of one of the registered USER. On different or even in the same PC. More on it will follow.

But, anyway back on analysis, just looking at those ideas of statistical recognition leave …

From a TX data string like …

IW1AYD 599 004 004 IW1AYD

the RX resulting string could be …

IW1AY 599 034 004 IW1AYD

well valid but still needing operator, human, correlation to be engaged.

GRITTY aiming II

This second aiming is considering to whom GRITTY is aimed to be used. My understood is that GRITTY will be used by contest enthusiasts early adopter. They will have advantages on using it, maybe some. They will have some way to test different GRITTY behavior, not that much. Right now, for common users with low and scarce knowledge of PC GRITTY is just another decoder to look at, when having the screen real estate to use it.

The underlining question is: would be GRITTY aimed to end user with different characteristics than RTTY Skimmer volunteers?

Different answers will require substantial changes to GRITTY.

Also as GRITTY is now free of any charge to use it several other questions may arise from this same points.

Integration and with OS & N1MM+

Too much integrated into the OS, usual for MS hate or love it. It has all the needed files/components into the MS programs sub directories. Just the executable and the ini are moveable. It is extremely not portable by itself. It need care to be moved along, together with is dispersed ancillary files, into different PC, MS, M2, MM installations.

Further attempts to make it portable will follow.

With N1MM+ any attempt to have a DI subwindow populated by incoming data hadn't worked at all. Clicking a single call decoded by GRITTY move that call inside Main Input window call-sign field, OK.

BTWw FLDIGI suffer the same integration syndrome.

Configurability

Out of the portability issue some parameters are not configurable by clicking, i.e., a preference menu.

This seems an early stage of the whole thing, there isn't a central preference menu where to get forth and back clicking buttons so to enforce or not rules or where to enter text as to use new checking schemas.

From one side this could mean: no user setup out of hands on operations on files. From the other side it means that those operations aren't necessarily to be done. It's' hard to say on V 1.2. Speculating on the subject could give different answers but it will be not useful by now.

Logs

Some attempts to use simple strings commands, as explained by the help chm file, though the TCP port went on successfully, but without any effectiveness. Effectiveness is measured having on hands the old but still unique and effective MMTTY log file.

Out of this, it seems that in a slightly real time ambient like SO2R or Mx contesting using the network for a simple task like this should be considered heavy. As usual not by only itself but considering all the other dressing we have to use and what's typically under the hood of our PC.

So we should wait until some good soul will come in help of us making anything of useful to write a file with incoming data. I hope that anybody will recognize the importance to have an automated and clever log output. That's not only for cheating purposes, it seems that everybody write software is preventing logging as a cheating countermeasure. A log, a complete log with TX and RX timestamps is needed also for DXing and on any mode for QSLing. I often ended out a fragment of MMTTY log to say “sorry you are not in the log as I received this”. One or more times I confirmed with LOTW QSO not in my log for the contest but there +callsigns+599+ only for the QSO itself. Having a log is not cheating. Thinking that a log is not useful or it must be a hard wired operation to be done by hands is not a good practice.

Real life unravel testing

It was not a goal to test it that deeply. Nor to simulate it on the run. It will be a task to be accomplished in the future.

Overall impression

From the interesting reading about the subject and testing it on the fly during some QSO, BTW low tones, it doesn’t seem to be used first line now. It is something new, well working, more promising and stable - some hours. It is made to be a great tool.

Summary of issues

Not all, but what was seen until now.

1) Integration and portability, N1MM+ could be rendered truly portable, at least by the file system standpoint. GRITTY will not follow. Nothing that could not be solved by some scripting and checks. Still having more thing to be done, but not to be done the last five minutes before 00:00 UTC or in the middle of a creek. KIS

1.a) metering is how much are portable MMTTY and 2Tone. They don’t need, not even force for, to be sparsely installed into default Windows directories. But could reside, read parameters and write logs into user defined places. It’s easy to tie all those together into well-known places.

2) N1MM+ integration is coming. It is only half done right now. Clicking a call will place it into the main input window. But incoming data aren’t showed into the DI place, that result not useful by now. That is just in case anyone use it. BTW judging on the rate and type of incoming symbols GRITTY will need much more printable space onto its sub DI printable window. Those data are literally flying into the GRITTY window. As a result the incoming string to be interpolated or not are prone to flow out of the readable lines to easy.

2.a) metering is how other decoders work. Changes are welcomed but they should have measurable advantages over previous things. Still changes have to be aimed to several other factor as, i.e., KIS.

3) Preferences and control over the ini file/s is to be done with some help by hands. Not a lack by itself, but a pitfall together with the main issue. Places to be reached are sparse into “you must remember” sub-directories names into the standard – for Microsoft – trees. Further to this there is not a GRITTY directory in view where it live, as to search it by eyes. It is placed down into a AFREET directory you must remember before.

4) Logs aren't just a piece of anyone other cake, QSL data verifying, UBN understanding. Or the pie would result already sliced and somewhat lacking on slices.

4.1) having another networked subsystem under the hood could increase, with high or low tops, the lack of HW(&SW-OS) resources. Carving all that onto the same environment, even if asynchronously
driven by OS buffers, could be too much for two cores two treads
machines.

4.a) metering could be empirically considered having a look to other inter-program communications transposed into Inter-networking communications. The inter-networking communication, making programming code communicate each other, make sense when prototyping – since the UNIX roots of the method – or preparing to slice applications in a Client/Server behavior. It doesn’t make sense for local facilities even if more easy to be done reusing code already made. I.E. syslogd vs. dmesg.

From the same environment as used for any other SO/MO contesting, N1MM+ last rev. MTTY + double 2Tone 24" 1920x1200(*), to adding one copy and then a second of GRITTY 30'.

From that start to some more checks 3 hours, a scarce dozen of HRD and QSO 1 hour.

Platform: HW SFF Mac Mini Early 1012 I7 2.3/4 GHz, 16 GB, 1 GB HD not SSD; OS W 8.1 latest patches as yesterday. Other applications are there not to be used with the logger. To have low latency is and ongoing task, not a simple check when all was installed (<150uS). Radio interfaces are Microham devices, DKII, MKII, MK2R+. Keyboard/s and mouse stuffs via USB/KVM.

Further considerations and close

Going a little bit out of the main topic this is a view that could be shared: isn't?

What should be really needed, as MMVARI is making it a lot but could be made better, is a multi-stream RTTY decoder into the 3 KHz of usual, not SDR, band-pass. But not alone a things like this need some more integration – may be also from the logger side. A pure dream is a DXEpedition machine: a SDR connected to the MF output of a traditional FSK transceiver. Leaving to the operator of that time the abilities: to receive a relevant slice of the pile up – AGC and sub filtering well achieved; controlling with no delays the whole train; having in view the most of the generated pile up. Lags in between the subsidiary, not that much, SDR receiver and the transceiver should be the lowest possible, i.e. not RS-232 replication black holes.

But this is just a dream, as several other I personally have. Maybe GRITTY is not the one but his big parent, the RTTY skimmer is to be used – mod for.

All this is not needed for RTTY contesting. What we would need is measured often into a 300/500 Hz bandwidth and thus the ability to catch on signals slightly out of tracks.

73 de iw1ayd Salvo

(*) screen real estate is a scarce, if not lacking, resource, usually. Well-defined placement of windows is not optional.

IW1AYD's backgrounds

Doing some PSK, JT modes on HF since 2005. Licensed since 1983. Pursuing RTTY contesting as other life aspects permit since 2008. Most of these, the contests, where done from the IQ1RY Club Station, both as part of a team or as a single operator. Since 2008, the switch over RTTY was driving also the switch to N1MM. Working into IT over several different scenarios and matters all over the years. From hardware integration, DC care, Security, FCO & ECO to OS & Application prototyping and pursuing deployments. On each matter, working roles went from Field Service Engineer, Technical Support for Customers, Product Specialist & Support to Consulting, Architect, Trainer, Nerd and Geek. Some thoughts: KIS is the dogma; KISS is not the same; Sustainable maintainability is a must; RTFM is a dogma, too often forgotten; Open Source software need more than windows clicking.

All the document was produced with Pages.