IEC 104 Multiple-Masters to Single-Slave (IEC104MM2SS) – Ver 1.0

By M. Medhat

Contents

If this project help you reduce time to develop, please spend some of your time to check link below:

The Fog is Lifting video

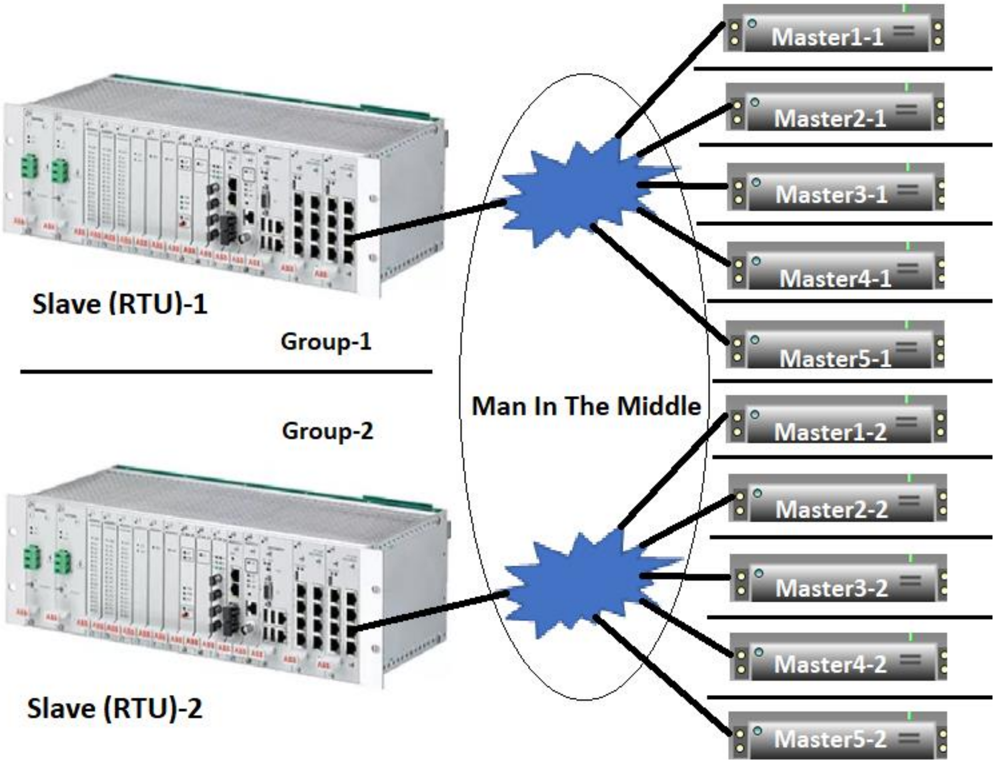
https://www.youtube.com/watch?v=bdH16Hz1naI

The Fog is Lifting' is one of the best holistic presentations a non Muslim can watch about Islam to learn about: its meaning, its beliefs and its concepts.

Introduction and why I wrote this program

This program will connect multiple masters (SCADA master stations) to a single slave (RTU) as defined in protocol IEC 60870-5-104. Although the protocol IEC 104 itself doesn't have a direct way to do that but the program will play the Man In The Middle role to achieve this connection.

Any number of masters connected to a slave is forming a group. You can create any number of groups and the program will establish the communication among each group members independent on the other groups. Each master in each group can receive IO status and send commands to the slave RTU independent on the other masters.



Why may anyone need the IEC104MM2SS program?

For cyber security reasons, the slave RTU is configured to accept connections from specific number of masters' IPs. Some RTUs may have limited number of masters to be configured.

In my company, we were replacing our legacy SCADA system with new one. During the test period we need both two systems' servers to communicate to the RTUs and station control systems (SCS) simultaneously. We have 2 old servers + 2 new servers at the main control center + 1 server at the backup control center. These total of 5 servers so here is the problem:

1- Some RTUs and SCS stations doesn't have enough entries for all the servers.
2- Some old RTUs and SCS stations have many configuration difficulties.
3- For many stations it is tedious to configure the RTUs/SCS many times to add the servers during the test period then to remove the old servers again after that.
4- Connecting all masters over the iec104mm2ss program will save and reduce the data traffic (quota) to at least half which is important for limited bandwidth stations and specially for stations connected over 2G/3G/LTE to reduce monthly payments to ISPs.
5- The program can sequentially try multiple IP:PORT to establish the communication to the slave (RTU) which will save number of channels (specified under master license) required for each RTU on each master.

For the above reasons, I wrote the IEC104MM2SS program.

Program features:

- Can easily create any number of groups of masters + slave to communicate between any master and slave. No communication allowed among masters.
- the program will establish the communication among each group members independent on the other groups.
- Each master in each group can receive IO status and send commands to the slave RTU independent on the other masters.
- To not affect the RTU availability reports, program will not accept connections from the master SCADA systems until the program connected first to the corresponding slave RTU. If disconnected from the RTU then it will disconnect all the masters in the same group.
- Trying to not loss any alarm or event without reaching to at least one master, the program will not send "startdt" packet to the slave RTU until at least one master connected. If all masters disconnected, then "stopdt" packet will be sent to the slave to stop data traffic until at least one master connected again.
- Easy configuration file building (.csv format) by using spreadsheet programs such as MS Excel.
- IP/Network filtration for master in each group or slave entry independently on other groups.
- Multiple IP:PORT numbers for each slave (RTU/SCS). Program will try the IP:PORT one by one until establish the connection to the slave RTU/SCS.
- Linux and windows compatible.
- Python native graphical user interface (GUI) (no need for third party solutions).
- Multithread operation.

- Time synchronization through multiple NTP servers.
- Log file for each group.
- “No GUI” mode of operation in which the program will work in background silently while only update the log files.
- Capability to work as a service under Window OS.

Program is distributed under GPL license and could be found on GitHub:

https://github.com/med7at69/IEC104-MultipleMasters2SingleSlave

It is written in python3 language and code is supporting both Windows and Linux OS.

Package contains the following files:

iec104mm2ss.py: The code in python 3 language.

iec104mm2ss.csv: ini file in comma separated values. Must be in the same folder where program starts in.

Iec104mm2ss.pdf: Help file in pdf format.

Readme.md

LICENSE file.

Program arguments

There are two formats for the program arguments, one to provide parameters to run program normally (not as Windows service) and the other one is for manipulation of service operation. It is not possible to mix the two formats together.

-h or --help	display help message.
-i or --ini	specify init file name.
-t or --ntp_update_every_sec	NTP update interval (seconds). Default = 900 sec.
-s or --ntp_server	NTP server (may repeated for multiple servers).
-n or –nogui	No GUI operation.
-c or –service	Run as Windows service (no GUI operation).

Usage:

usage iec104mm2ss [[-h][--help]] [[-i][--ini] init-file] [[-t][--ntp_update_every_sec] sec] [[-s][--ntp_server] ntpserver]

- Updating local time requires admin/root privilege.
- init file is a comma separated values format, default: iec104mm2ss.csv
- “-s or --ntp_server” could be included multiple times for multiple servers.

example1:

```
iec104mm2ss -i iec104rs1.csv
```

example2:

```
iec104mm2ss --ntp_server pool.ntp.org --ntp_server time.windows.com
```

To install the program as Windows service run 'iec104mm2ss --startup auto install'

When the program run as Windows service it will use the default initial file ‘iec104mm2ss.csv’. To use another initial file, check the part in this file explaining using Windows “sc” tool,

Details of service arguments:

Usage: 'iec104mm2ss [options] install|update|remove|start [...]|stop|restart [...]|debug [...]'

Options for 'install' and 'update' commands only:

--username domain\username : The Username the service is to run under

--password password : The password for the username

--startup [manual|auto|disabled|delayed] : How the service starts, default = manual

--interactive : Allow the service to interact with the desktop.

--perfmonini file: .ini file to use for registering performance monitor data

--perfmondll file: .dll file to use when querying the service for

performance data, default = perfmondata.dll

Options for 'start' and 'stop' commands only:

--wait seconds: Wait for the service to actually start or stop.

If you specify --wait with the 'stop' option, the service

and all dependent services will be stopped, each waiting

the specified period.

Program operation

- Groups are isolated from each other.

- Masters in the same group are isolated from each other.
- In each group:
  - Each master should have originator (org) address = 0. This is usually the default for all IEC 104 masters.
  - Each group entry should have an RTU number or ASDU address independent on other groups so the RTU number could be repeated for multiple groups without problem.
  - U-Format packets (startdt, stopdt, testfr, etc.) and S-Format packet will not be forwarded from master to slave and vice versa.
  - I-Format packets:
    - "End of initialization" will not be forwarded from slave to master.
    - Status (SPI, DPI) and AMI signals received from each slave are forwarded to all masters. This keeps all masters updated all the time.
    - Other I-Format control packets such as general interrogation (GI), Time synchronization, SCO, DCO, etc. will keep isolated among all masters and will be forwarded only from the master who send this packet to the slave. The reply from the slave on these I-Format packets will be forwarded only to the specific master who initiated the transmission at first time.
- Program is collecting the configuration parameters of all groups (masters and slave) from comma separated values file (csv) format for the following reasons:

  1- "csv" format is simple and well known since long time.

  2- Besides supported by Microsoft Excel, there are many freeware programs supporting editing "csv" files.

  3- It is easy to add, delete, copy, and paste large number of data entries to the "csv" files without complications.

Initial file (default name is iec104mm2ss.csv): It is a comma separated values file format or ".csv" which should be available in the same folder where program starts in. In the file you can define the following:

- a. NTP servers to update local time of the PC (requires admin/root privilege).
- b. Number of seconds to periodically update local time from NTP servers.
- c. "nogui" entry will start the program without GUI interface.
- d. Any number of groups. On each group entry, only one slave (RTU/SCS) can be connected to + any number of masters (SCADA master stations) limited by "maximum connection" field. For each group entry, the following parameters could be defined:
  - i. Unique port number to listen on and receive master connections.
  - ii. RTU number.
  - iii. Maximum master connections could be received and accepted.
  - iv. Connection idle time after which program will disconnect and reconnect again.
  - v. IEC 104 constants such as w, k, t1, t2 and t3.
  - vi. Buffer size of packets.
  - vii. Log events enable or disable. Usually you should enable the log events at the beginning and for some period to be sure everything is working fine then you can disable it after that.

viii. How to handle master connections from same IP address (accept “all”, keep “old” connection and reject the new one or keep the “new” connection and disconnect the old).
ix. List of slave IPs:Ports to try connection to until one connected.
x. Filter of accepted masters IPs/Network.

Log file for each RTU and each master are saved in folder “log”. Folder “log” will be created in the same folder where the program starts in.

When the program starts it will:

1- Read the program arguments if provided by user.
2- If initial file is not provided in program arguments, then the program will use the default iec104mm2ss.csv
3- Read the initial file to get the NTP servers and masters/slaves as described later.
4- Each group or slave (RTU) entry should have name, RTU number, port number to listen for coming master connections and IP:Port list separated by “;”. Program will try each one periodically until connected to the slave (RTU).
5- To speed up the loading of group entries, program will not start any connection until load all entries in the memory.
6- For any connection, if idle time (in seconds which configured in the initial file) passed without send/receive data, then the program will disconnect the connection to restart a working connection again.

Initial file format

General notes:

- Initial file format is comma separated values format (csv).
- Initial file default name is iec104mm2ss.csv
- You can provide another name as program argument with “-i” or “--ini”
- Initial file should be in the same folder where the program starts in.
- If first character of first column in any row is “!” Then program will stop reading the initial file and cancel the rest of the rows.
- Initial file will start by defining the following parameters:
    - “nogui”: if exist then program will start without GUI interface. Still the program will update log files for each group silently if logging is enabled.
    - “ntp_server”: it could be repeated in multiple rows for multiple NTP servers. If program has admin/root privilege, then it will try all the NTP servers one by one to synchronize the local time.

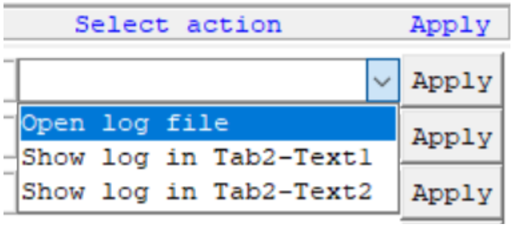
    - “ntp_update_every_sec”: seconds to periodically update local time. If not provided, then the default is 900 seconds.
    - IEC 104 constants: w, k, t1, t2 and t3 as defined by IEC 104 protocol.
- The initial file should contain all groups of masters/slaves’ information one by one (each row contains one complete slave and its masters information).
- Slave (RTU) entry of each group should come first before any master in the same group otherwise master entry will be neglected.
- Each group entry defines the following parameters in separated rows:
    - ID:
        - Should be unique number for each group contains multiple masters + single slave.
        - If “id” field is not number, then it will be considered as a comment line and will be neglected by the program.
        - If first character of “id” column in any row is “!” Then program will stop reading the rest of the rows in the initial file and will cancel the rest of the rows.
    - System/RTU name: Name with maximum of 16 characters length.
    - Port number: Unique (not used for any other group entry) port number (1-65535). Program will listen to this port to accept multiple masters’ connections.
    - RTU number: RTU number (1-65535). RTU number is not unique and multiple groups can have the same RTU number.
    - Max. conn: Maximum master connections could be received and accepted.
    - Idletime: Connection idle time after which program will disconnect and reconnect again.
    - IEC 104 constants such as w, k, t1, t2 and t3.
    - Buffsize: Buffer size of packets. Minimum buffer size is equal “k” constant. Default is 1000 packets.
    - Logevents (Y/N): Log events enable or disable. Usually, you should enable the log events at the beginning and for some period to be sure everything is working fine then you can disable it after that.
    - Keep conn from same IP (all, new, old): How to handle master connections from same IP address:
        - “all”: accept “all” connections even from same IP already connected.
        - “old”: Keep “old” connection and reject the new one.
        - “new”: Keep the “new” connection and disconnect the old.
    - Slave hosts: List of slave IPs:Ports separated by “;”. Program will periodically try them one by one until connected to the slave (RTU). Example: 192.168.2.5:2404;127.0.0.1:2405
    - Masters filter: Filter of accepted masters IPs/Network separated by “;” Program will not accept master connection from same IP which slave connected to. For example if slave is connected to 192.168.1.1 then program will not accept master connection from same IP 192.168.1.1. Example of entry: 192.168.1.0/24;10.1.1.0/29

Please check appendix A for sample initial file format.

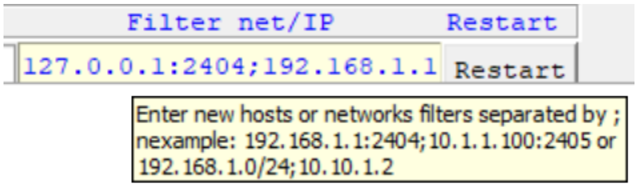
Program GUI

Trying to make the graphical user interface as simple as possible:

1- In each master/slave entry you can select to view the log file or to edit the entry parameters and view its log file in the "data edit" tab.



2- Configuration of all RTUs/Systems are initially read from the initial file (default: iec104mm2ss.csv). However, you can select any group entry and display it in the "data edit tab" then modify some parameters then restart the master/slave connection with the new parameters. Editing and changes of any entry will not be saved in the initial file. Please refer to below screenshot (entries in light yellow color could be changed and it will take effect after restarting the connection).



127.0.0.1:2404;192.168.1.1

Enter new hosts or networks filters separated by ;
nexample: 192.168.1.1:2404;10.1.1.100:2405 or
192.168.1.0/24;10.10.1.2

3- Floating tooltips is displayed whenever possible to explain the GUI part.

More screenshots available in appendix "C".

Using Windows "sc" tool to manipulate service

When using the program to connect multiple SCADA masters to the same slave (RTU/SCS) then usually you will run the program from the SCADA front end servers, and it should run all the time and start running even if no one logged on the server. So, we need to install the program to run as a service (or multiple services) under Windows.

The best way to install and start multiple instances of the program as multiple Windows services is to use the Windows "sc" tool came with Windows.

For example, if you copied the program to the folder “c:\iec104mm2ss_1”, to install the service you can run “sc” as follows:

```
sc create iec104mm2ss binPath=" c:\iec104mm2ss_1\iec104mm2ss.exe -c" start="auto"
DisplayName="IEC104 Multiple Masters to Single Slave_1"
```

For old windows versions such as windows server 2003 and 2008 we have to leave a “space” between the option and its value:

```
sc create iec104mm2ss binPath= " c:\iec104mm2ss_1\iec104mm2ss.exe -c" start= "auto" DisplayName=
"IEC104 Multiple Masters to Single Slave_1"
```

The “-c” is a must to tell the program to start as a service not in program normal operation.

Another example: if you want to specify initial file name other than the default name:

```
sc create iec104mm2ss_1 binPath=" c:\iec104mm2ss_1\iec104mm2ss.exe -c -i iec104mm2ss_1.csv"
start="auto" DisplayName="IEC104 Multiple Masters to Single Slave_1"
```

The initial file “iec104mm2ss_1.csv” should also be in the same folder of the program which is here “c:\iec104mm2ss_1”

You can easily create another service instance in another folder as follows:

```
sc create iec104mm2ss_2 binPath=" c:\iec104mm2ss_2\iec104mm2ss.exe -c -i iec104mm2ss_2.csv"
start="auto" DisplayName="IEC104 Multiple Masters to Single Slave_2"
```

For more information, please check the Windows “sc” help.

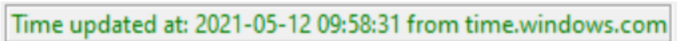
Troubleshooting

Program did not load one or more configured masters/slaves in the initial file:

- If first character of “ID” field equal “!” then current row and all subsequent rows (slaves) will not be loaded.
- If the “ID” field is not number, then master/slave entry will not be loaded.
- If group (slave) has no name, then this entry will not be loaded. Program will read the first 16 characters of the name field.
- “RTU number” field should be in range 1 to 65535.
- “Port number” field for each group entry should be unique (not used for any other group entries) and in range 1 to 65535.
- For any group (slave) entry, if “Hosts” field does not contain proper IP:PORT combinations then this slave entry will not be loaded.

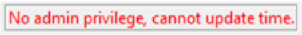
Local time not updated although NTP servers configured, and it is tested normally.

- Updating local time required admin/root privilege under both Windows and Linux so please be sure to start the application with admin/root privilege so it can update the local time normally.
- Program will try the NTP servers one by one then will sleep for the specified period (ntp_update_every_sec = 900 seconds by default) before trying again. So, maybe software couldn’t reach to the servers at the first try so please wait until the software try next time.
- Please notice the local time update status as indicated in the screenshots below:



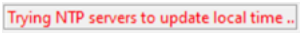
Time updated at: 2021-05-12 09:58:31 from time.windows.com

Local time updated successfully.



No admin privilege, cannot update time.

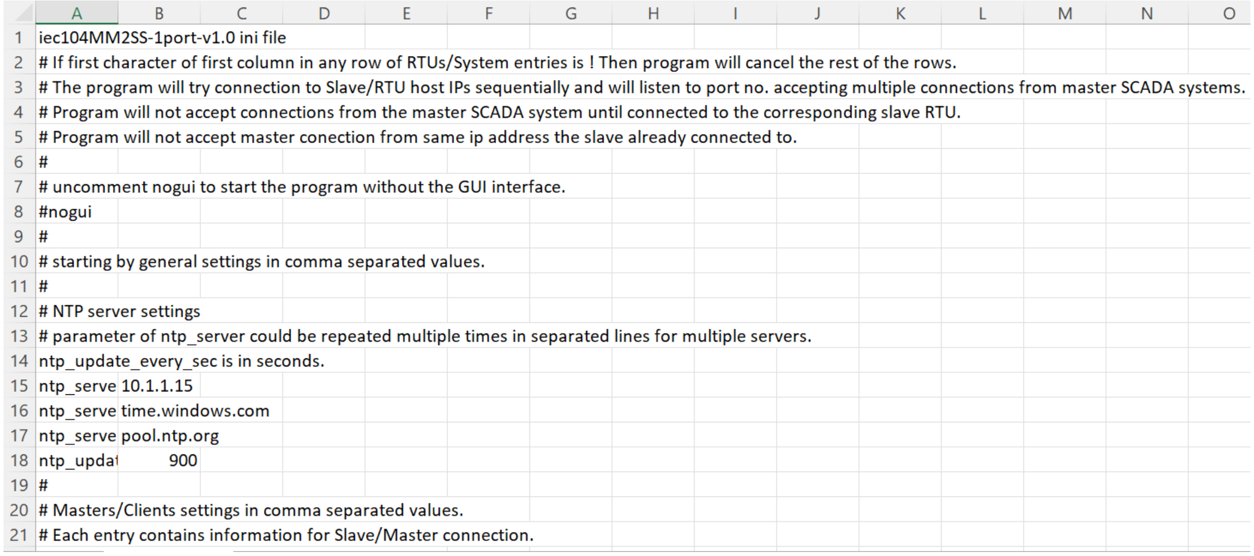
No admin privilege so program cannot update the local time.



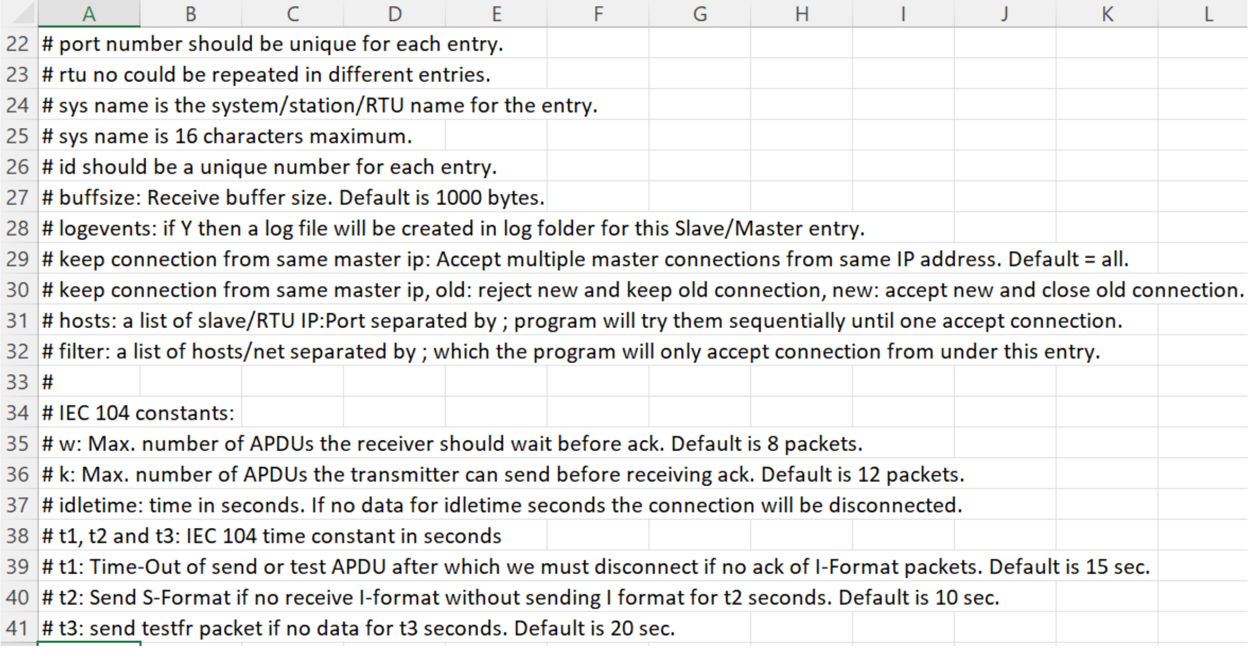
Trying NTP servers to update local time ..

Program is trying but cannot connect to the NTP servers so please check the network connection and the availability of the configured servers.

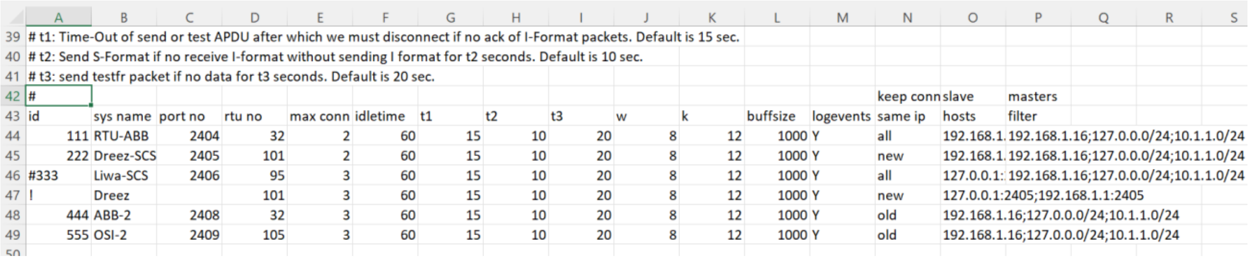
Appendix A - Sample initial file



	A	B
1	iec104MM2SS-1port-v1.0 ini file	
2	# If first character of first column in any row of RTUs/System entries is ! Then program will cancel the rest of the rows.	
3	# The program will try connection to Slave/RTU host IPs sequentially and will listen to port no. accepting multiple connections from master SCADA systems.	
4	# Program will not accept connections from the master SCADA system until connected to the corresponding slave RTU.	
5	# Program will not accept master conection from same ip address the slave already connected to.	
6	#	
7	# uncomment nogui to start the program without the GUI interface.	
8	#nogui	
9	#	
10	# starting by general settings in comma separated values.	
11	#	
12	# NTP server settings	
13	# parameter of ntp_server could be repeated multiple times in separated lines for multiple servers.	
14	ntp_update_every_sec is in seconds.	
15	ntp_serve	10.1.1.15
16	ntp_serve	time.windows.com
17	ntp_serve	pool.ntp.org
18	ntp_updat	900
19	#	
20	# Masters/Clients settings in comma separated values.	
21	# Each entry contains information for Slave/Master connection.	



	A
22	# port number should be unique for each entry.
23	# rtu no could be repeated in different entries.
24	# sys name is the system/station/RTU name for the entry.
25	# sys name is 16 characters maximum.
26	# id should be a unique number for each entry.
27	# buffsize: Receive buffer size. Default is 1000 bytes.
28	# logevents: if Y then a log file will be created in log folder for this Slave/Master entry.
29	# keep connection from same master ip: Accept multiple master connections from same IP address. Default = all.
30	# keep connection from same master ip, old: reject new and keep old connection, new: accept new and close old connection.
31	# hosts: a list of slave/RTU IP:Port separated by ; program will try them sequentially until one accept connection.
32	# filter: a list of hosts/net separated by ; which the program will only accept connection from under this entry.
33	#
34	# IEC 104 constants:
35	# w: Max. number of APDUs the receiver should wait before ack. Default is 8 packets.
36	# k: Max. number of APDUs the transmitter can send before receiving ack. Default is 12 packets.
37	# idletime: time in seconds. If no data for idletime seconds the connection will be disconnected.
38	# t1, t2 and t3: IEC 104 time constant in seconds
39	# t1: Time-Out of send or test APDU after which we must disconnect if no ack of I-Format packets. Default is 15 sec.
40	# t2: Send S-Format if no receive I-format without sending I format for t2 seconds. Default is 10 sec.
41	# t3: send testfr packet if no data for t3 seconds. Default is 20 sec.



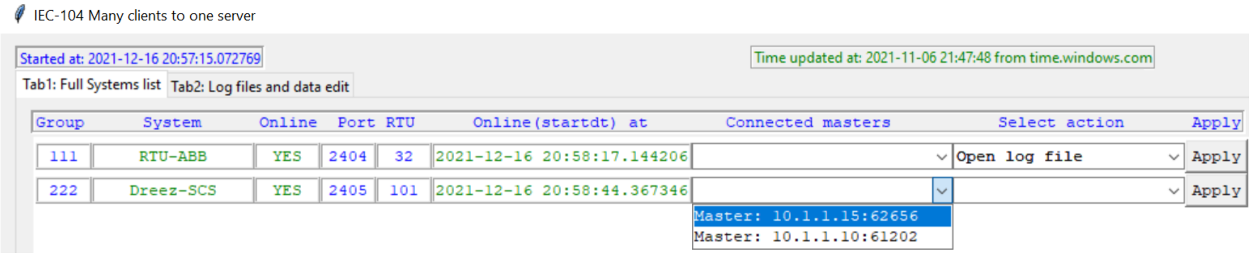
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
39	# t1: Time-Out of send or test APDU after which we must disconnect if no ack of I-Format packets. Default is 15 sec.															
40	# t2: Send S-Format if no receive I-format without sending I format for t2 seconds. Default is 10 sec.															
41	# t3: send testfr packet if no data for t3 seconds. Default is 20 sec.															
42	#													keep conn	slave	masters
43	id	sys name	port no	rtu no	max conn	idletime	t1	t2	t3	w	k	buffsize	logevents	same ip	hosts	filter
44	111	RTU-ABB	2404	32	2	60	15	10	20	8	12	1000	Y	all	192.168.1.	192.168.1.16;127.0.0.0/24;10.1.1.0/24
45	222	Dreez-SCS	2405	101	2	60	15	10	20	8	12	1000	Y	new	192.168.1.	192.168.1.16;127.0.0.0/24;10.1.1.0/24
46	#333	Liwa-SCS	2406	95	3	60	15	10	20	8	12	1000	Y	all	127.0.0.1:	192.168.1.16;127.0.0.0/24;10.1.1.0/24
47	!	Dreez		101	3	60	15	10	20	8	12	1000	Y	new	127.0.0.1:2405;192.168.1.1:2405	
48	444	ABB-2	2408	32	3	60	15	10	20	8	12	1000	Y	old	192.168.1.16;127.0.0.0/24;10.1.1.0/24	
49	555	OSI-2	2409	105	3	60	15	10	20	8	12	1000	Y	old	192.168.1.16;127.0.0.0/24;10.1.1.0/24	

```
iec104MM2SS-1port-v1.0 ini file,,,,,,,,,,,,,,
# If first character of first column in any row of RTUs/System entries is ! Then program will cancel the rest of the rows.,,,,,,,,,,,,,,
# The program will try connection to Slave/RTU host IPs sequentially and will listen to port no. accepting multiple connections from master SCADA systems.,,,,,,,,,,,,,,
# Program will not accept connections from the master SCADA system until connected to the corresponding slave RTU.,,,,,,,,,,,,,,
# Program will not accept master conection from same ip address the slave already connected to.,,,,,,,,,,,,,,
#,,,,,,,,,,,,,,
# uncomment nogui to start the program without the GUI interface.,,,,,,,,,,,,,,
#nogui,,,,,,,,,,,,,,
#,,,,,,,,,,,,,,
# starting by general settings in comma separated values.,,,,,,,,,,,,,,
# ,,,,,,,,,,,,,,
# NTP server settings,,,,,,,,,,,,,,
# parameter of ntp_server could be repeated multiple times in separated lines for multiple servers.,,,,,,,,,,,,,,
#ntp_update_every_sec is in seconds.,,,,,,,,,,,,,,
#ntpserver,10.1.1.15,,,,,,,,,,,,,
#ntp_server,time.windows.com,,,,,,,,,,,,,
#ntp_server,pool.ntp.org,,,,,,,,,,,,,
#ntp_update_every_sec,900,,,,,,,,,,,,,
#,,,,,,,,,,,,,,
# Masters/Clients settings in comma separated values.,,,,,,,,,,,,,,
# Each entry contains information for Slave/Master connection.,,,,,,,,,,,,,,
# port number should be unique for each entry.,,,,,,,,,,,,,,
# rtu no could be repeated in different entries.,,,,,,,,,,,,,,
# sys name is the system/station/RTU name for the entry.,,,,,,,,,,,,,,
# sys name is 16 characters maximum.,,,,,,,,,,,,,,
# id should be a unique number for each entry.,,,,,,,,,,,,,,,

# buffsize: Receive buffer size. Default is 1000 bytes.,,,,,,,,,,,,,,,

# logevents: if Y then a log file will be created in log folder for this Slave/Master entry.,,,,,,,,,,,,,,,

# keep connection from same master ip: Accept multiple master connections from same IP address. Default = all.,,,,,,,,,,,,,,,

"# keep connection from same master ip, old: reject new and keep old connection, new: accept new and close old connection.",,,,,,,,,,,,,,,

# hosts: a list of slave/RTU IP:Port separated by ; program will try them sequentially until one accept connection.,,,,,,,,,,,,,,,

# filter: a list of hosts/net separated by ; which the program will only accept connection from under this entry.,,,,,,,,,,,,,,,

#,,,,,,,,,,,,,,,

# IEC 104 constants:,,,,,,,,,,,,,,,

# w: Max. number of APDUs the receiver should wait before ack. Default is 8 packets.,,,,,,,,,,,,,,,

# k: Max. number of APDUs the transmitter can send before receiving ack. Default is 12 packets.,,,,,,,,,,,,,,,

# idletime: time in seconds. If no data for idletime seconds the connection will be disconnected.,,,,,,,,,,,,,,,

"# t1, t2 and t3: IEC 104 time constant in seconds",,,,,,,,,,,,,,,

# t1: Time-Out of send or test APDU after which we must disconnect if no ack of I-Format packets. Default is 15 sec.,,,,,,,,,,,,,,,

# t2: Send S-Format if no receive I-format without sending I format for t2 seconds. Default is 10 sec.,,,,,,,,,,,,,,,

# t3: send testfr packet if no data for t3 seconds. Default is 20 sec.,,,,,,,,,,,,,,,

#,,,,,,,,,,,,,keep conn,slave,masters

id,sys name,port no,rtu no,max conn,idletime,t1,t2,t3,w,k,buffsize,logevents,same ip,hosts,filter

111,RTU-ABB,2404,32,2,60,15,10,20,8,12,1000,Y,all,192.168.1.1:2413;192.168.1.1:2414;192.168.1.1:2415;127.0.0.1:2410,192.168.1.16;127.0.0.0/24;10.1.1.0/24

222,Dreez-SCS,2405,101,2,60,15,10,20,8,12,1000,Y,new,192.168.1.1:2413;192.168.1.1:2414;127.0.0.1:2411;192.168.1.1:2415,192.168.1.16;127.0.0.0/24;10.1.1.0/24
```

```
#333,Liwa-SCS,2406,95,3,60,15,10,20,8,12,1000,Y,all,127.0.0.1:2413;192.168.1.1:2405,192.168.1.16;127.0.0.0/24;10.1.1.0/24

!,Dreez,,101,3,60,15,10,20,8,12,1000,Y,new,127.0.0.1:2405;192.168.1.1:2405,

444,ABB-2,2408,32,3,60,15,10,20,8,12,1000,Y,old,192.168.1.16;127.0.0.0/24;10.1.1.0/24,

555,OSI-2,2409,105,3,60,15,10,20,8,12,1000,Y,old,192.168.1.16;127.0.0.0/24;10.1.1.0/24,
```

Appendix B – GUI screenshots

Tab1 – Full master/slave list



IEC-104 Many clients to one server

Started at: 2021-12-16 20:57:15.072769

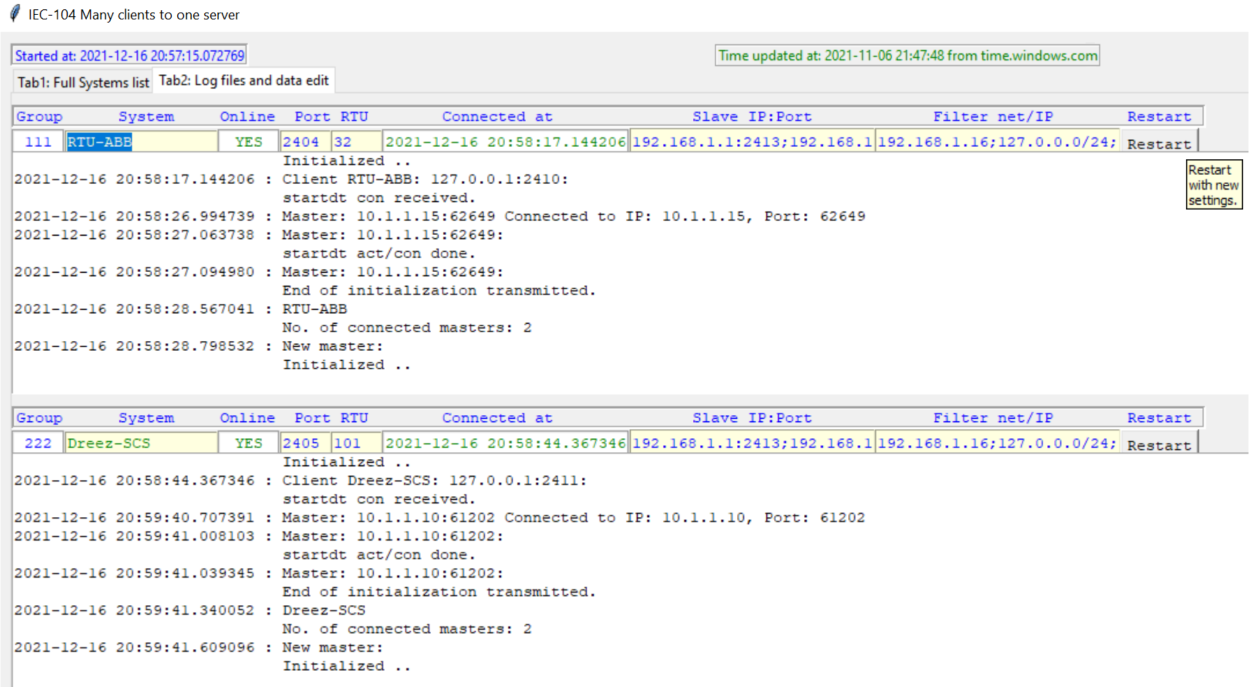
Time updated at: 2021-11-06 21:47:48 from time.windows.com

Tab1: Full Systems list | Tab2: Log files and data edit

Group	System	Online	Port	RTU	Online(startdt) at	Connected masters	Select action	Apply
111	RTU-ABB	YES	2404	32	2021-12-16 20:58:17.144206		Open log file	Apply
222	Dreez-SCS	YES	2405	101	2021-12-16 20:58:44.367346	Master: 10.1.1.15:62656 Master: 10.1.1.10:61202		Apply

2 groups.

Tab2 – Comparisons and parameters editing.



IEC-104 Many clients to one server

Started at: 2021-12-16 20:57:15.072769

Time updated at: 2021-11-06 21:47:48 from time.windows.com

Tab1: Full Systems list | Tab2: Log files and data edit

Group	System	Online	Port	RTU	Connected at	Slave IP:Port	Filter net/IP	Restart
111	RTU-ABB	YES	2404	32	2021-12-16 20:58:17.144206	192.168.1.1:2413;192.168.1	192.168.1.16;127.0.0.0/24;	Restart

Restart with new settings.

```
                             Initialized ..
2021-12-16 20:58:17.144206 : Client RTU-ABB: 127.0.0.1:2410:
                             startdt con received.
2021-12-16 20:58:26.994739 : Master: 10.1.1.15:62649 Connected to IP: 10.1.1.15, Port: 62649
2021-12-16 20:58:27.063738 : Master: 10.1.1.15:62649:
                             startdt act/con done.
2021-12-16 20:58:27.094980 : Master: 10.1.1.15:62649:
                             End of initialization transmitted.
2021-12-16 20:58:28.567041 : RTU-ABB
                             No. of connected masters: 2
2021-12-16 20:58:28.798532 : New master:
                             Initialized ..
```

Group	System	Online	Port	RTU	Connected at	Slave IP:Port	Filter net/IP	Restart
222	Dreez-SCS	YES	2405	101	2021-12-16 20:58:44.367346	192.168.1.1:2413;192.168.1	192.168.1.16;127.0.0.0/24;	Restart

```
                             Initialized ..
2021-12-16 20:58:44.367346 : Client Dreez-SCS: 127.0.0.1:2411:
                             startdt con received.
2021-12-16 20:59:40.707391 : Master: 10.1.1.10:61202 Connected to IP: 10.1.1.10, Port: 61202
2021-12-16 20:59:41.008103 : Master: 10.1.1.10:61202:
                             startdt act/con done.
2021-12-16 20:59:41.039345 : Master: 10.1.1.10:61202:
                             End of initialization transmitted.
2021-12-16 20:59:41.340052 : Dreez-SCS
                             No. of connected masters: 2
2021-12-16 20:59:41.609096 : New master:
                             Initialized ..
```

Appendix C – Windows binary files

Windows binary file is generated by nuitka Python compiler:

https://nuitka.net/

By using the following command:

```
python -m nuitka --windows-file-description="IEC104 Multiple Masters to Single Slave" --windows-file-version="1.0" --windows-product-version="1.0" --windows-company-name="M.M" --onefile --plugin-enable=tk-inter --standalone --mingw64 iec104mm2ss.py
```

Appendix D - Other projects

IEC 104 RTU Simulator

https://github.com/med7at69/IEC104-RTU-Simulator

IEC 104 RTU simulator is a program to simulate the operation of RTU (remote terminal unit) or server as defined by protocol IEC 60870-5-104. It can simulate any number of RTUs or servers. Simulated RTUs could be connected to different or same SCADA master station. IO signals are indexed and grouped by using index numbers. You can send IO signals from all RTUs to the connected SCADA master stations at once by using index number.

Program features:

- Simulation of any number of RTUs simultaneously.
- Connect to multiple SCADA systems at once.
- Can simulate redundant RTU ports. Program will send same IO signal to all RTUs with same RTU number.
- Can simulate redundant SCADA system connections.
- Easy IO database building by using spreadsheet programs such as MS Excel.
- IP address and network filtration for each RTU independently.
- Can repeat sending grouped (by index number) of IO signals for any period (in seconds). Also, a delay time in seconds could be applied after sending each IO.
- Can send any IO signal based on receiving filter conditions such as specific type ID, IOA, etc.
- Accordingly, could be used to simulate any site or factory acceptance tests (SAT/FAT) such as:
    - End to end (E2E) test.
    - Avalanche test (sending full IO signal list from all defined RTUs by using index 0).
    - Worst case (SCADA system emergency case).
    - Steady state test (normal operation).
    - SCADA system switchover time.

    - SCADA system time synchronization.
    - Load shedding.
    - FLISR (fault location, isolation, and service restoration).
    - Self-healing.
    - Etc.
- IEC 104 parameters (timing and k) for each RTU independently.
- Linux and windows compatible.
- Python native graphical user interface (GUI) (no need for third party solutions).
- Multithread operation.
- Time synchronization through multiple NTP servers.
- Detailed logging of all events and signals in comma delimited file format.