MATLAB INTERFACE FOR RPL DODAG VISUALIZATION

Software Recommended: NetSim Standard v13.0(64bit), Visual Studio 2019, MATLAB R2015b or higher

Project Download Link:

https://github.com/NetSim-TETCOS/RPL_DODAG_Visualization_in_IOT_v12.2.git

Follow the instructions specified in the following link to download and setup the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Introduction:

RPL (Routing Protocol for Low-Power and Lossy Networks) is a routing protocol for wireless networks with low power consumption and generally susceptible to packet loss. It is a proactive protocol based on distance vectors and operates on IEEE 802.15.4, optimized for multi-hop and many-to-one communication, but also supports one-to-one messages.

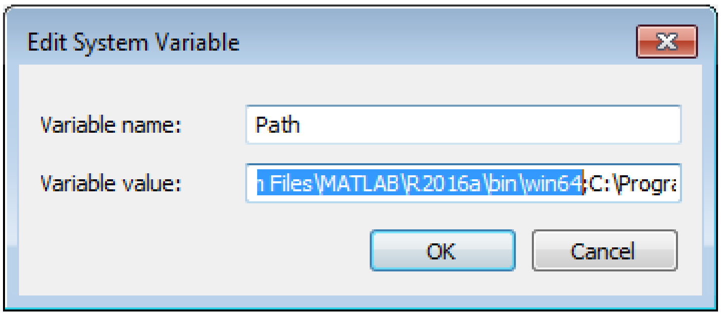
This protocol is specified in RFC 6550 with special applications in RFCs 5867, 5826, 5673 and 5548. RPL can support a wide variety of link layers, including those with limitations, with potential losses or that are used in devices with limited resources. This protocol can quickly create network routes, share routing knowledge and adapt the topology in an efficient way.

RPL creates a topology similar to a tree (DAG or directed acyclic graph). Each node within the network has an assigned rank (Rank), which increases as the teams move away from the root node (DODAG). The nodes resend packets using the lowest range as the route selection criteria.

The DODAG formed as part of RPL protocol in NetSim can be visualized by interfacing NetSim with MATLAB.

Steps to run MATLAB interface

1. Make sure that the following directory is in the PATH(Environment variable)
   <Path where MATLAB is installed>\bin\win64



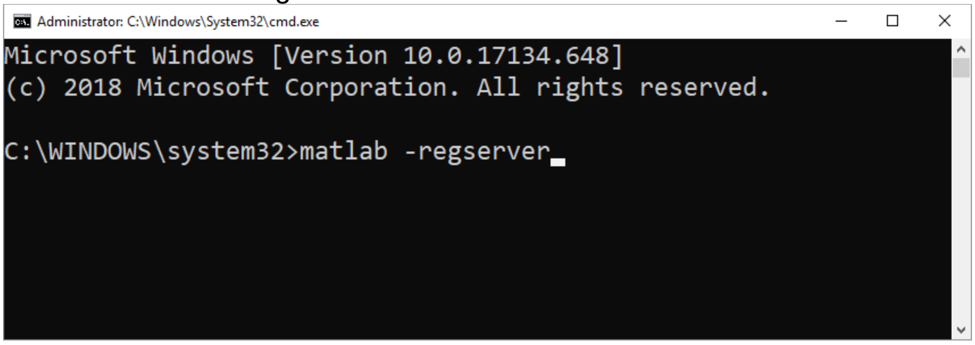
Variable name: Path

Variable value: n Files\MATLAB\R2016a\bin\win64;C:\Progra

   Note: If the machine has more than one MATLAB installed, the directory for the target platform must be ahead of any other MATLAB directory (for instance, when compiling a 64-bit application, the directory in the MATLAB 64-bit installation must be the first one on the PATH).

2. Open Command prompt as admin and execute the command “matlab -regserver”. This will

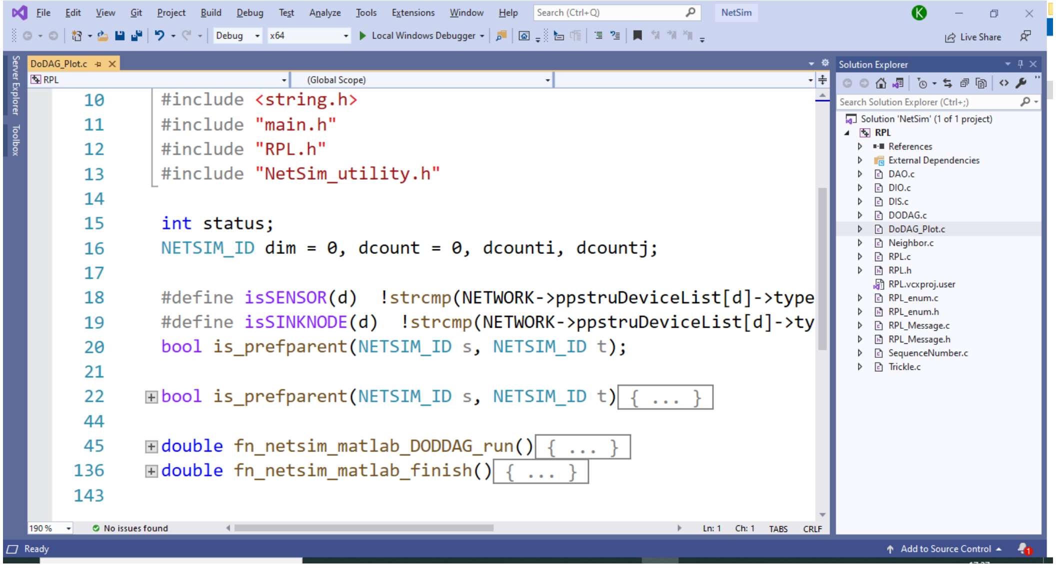
register MATLAB as a COM automation server and is required for NetSim to start MATLAB automation server during runtime.



```
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>matlab -regserver
```

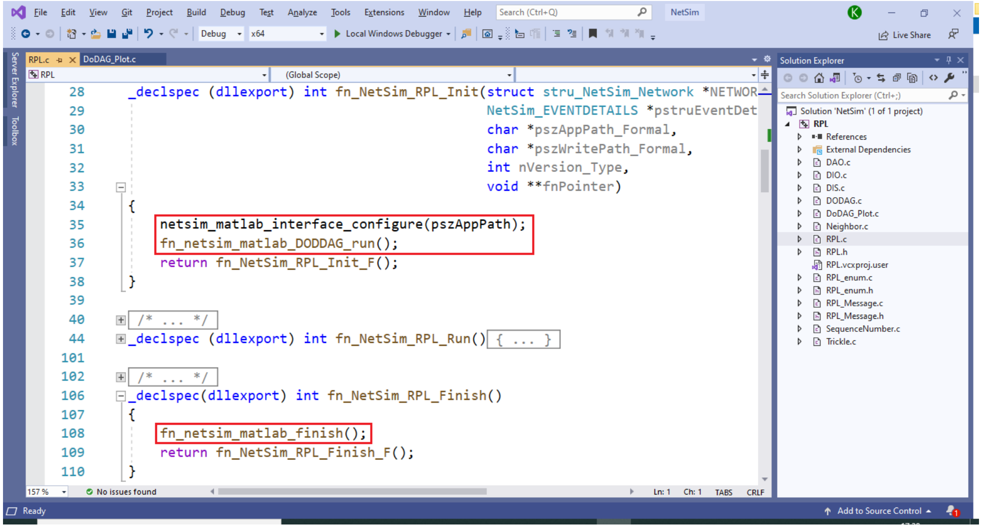
3. Go to home screen of NetSim , Click on **Your Work** ⟶ **Workspace options** ⟶ **Open code**

4. A DoDAG_Plot.c file has been added to the RPL.c file which takes care of interactions with MATLAB.



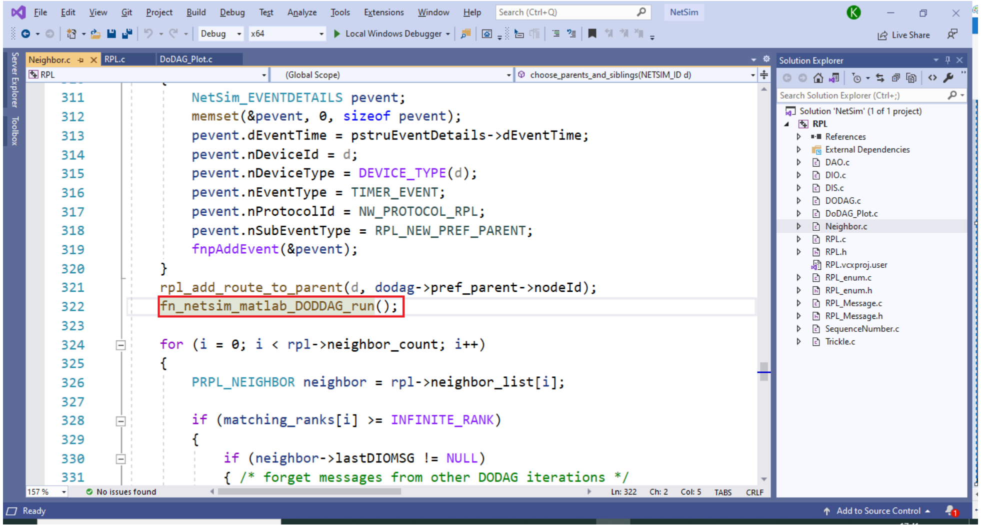
```
#include <string.h>
#include "main.h"
#include "RPL.h"
#include "NetSim_utility.h"

int status;
NETSIM_ID dim = 0, dcount = 0, dcounti, dcountj;

#define isSENSOR(d)  !strcmp(NETWORK->ppstruDeviceList[d]->type
#define isSINKNODE(d)  !strcmp(NETWORK->ppstruDeviceList[d]->ty
bool is_prefparent(NETSIM_ID s, NETSIM_ID t);

bool is_prefparent(NETSIM_ID s, NETSIM_ID t){ ... }

double fn_netsim_matlab_DODDAG_run(){ ... }
double fn_netsim_matlab_finish(){ ... }
```

5. In the RPL.c file the functions **fn_netsim_matlab_init()** and **fn_NetSim_RPL_Finish ()** are modified as shown below:



```
_declspec (dllexport) int fn_NetSim_RPL_Init(struct stru_NetSim_Network *NETWOR
                                             NetSim_EVENTDETAILS *pstruEventDet
                                             char *pszAppPath_Formal,
                                             char *pszWritePath_Formal,
                                             int nVersion_Type,
                                             void **fnPointer)
{
    netsim_matlab_interface_configure(pszAppPath);
    fn_netsim_matlab_DODDAG_run();
    return fn_NetSim_RPL_Init_F();
}

/* ... */
_declspec (dllexport) int fn_NetSim_RPL_Run(){ ... }

/* ... */
_declspec(dllexport) int fn_NetSim_RPL_Finish()
{
    fn_netsim_matlab_finish();
    return fn_NetSim_RPL_Finish_F();
}
```

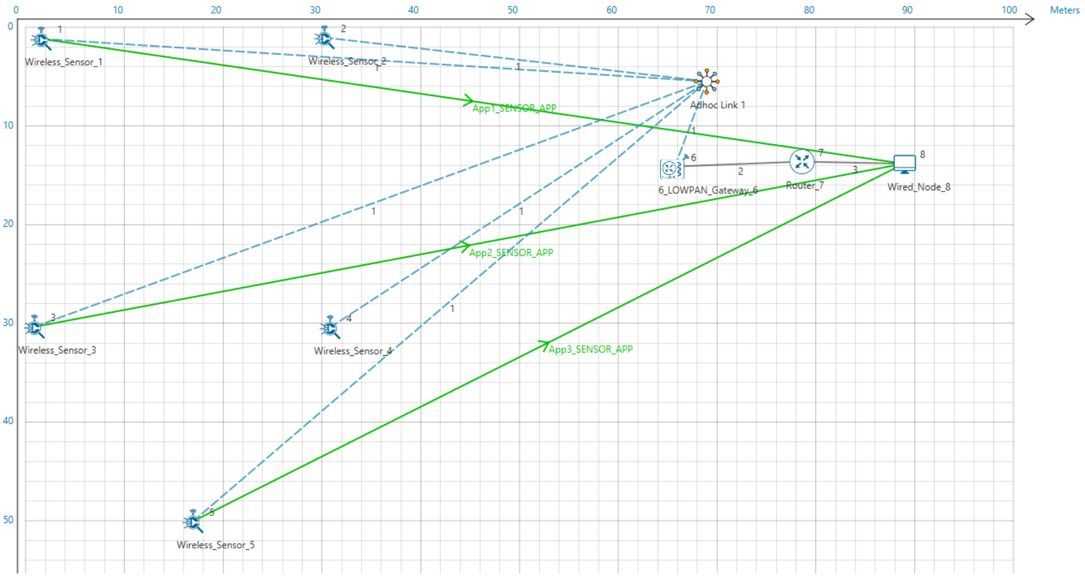
6. Following changes were done to the **Neighbor.c** file.



```
        NetSim_EVENTDETAILS pevent;
        memset(&pevent, 0, sizeof pevent);
        pevent.dEventTime = pstruEventDetails->dEventTime;
        pevent.nDeviceId = d;
        pevent.nDeviceType = DEVICE_TYPE(d);
        pevent.nEventType = TIMER_EVENT;
        pevent.nProtocolId = NW_PROTOCOL_RPL;
        pevent.nSubEventType = RPL_NEW_PREF_PARENT;
        fnpAddEvent(&pevent);
    }
    rpl_add_route_to_parent(d, dodag->pref_parent->nodeId);
    fn_netsim_matlab_DODDAG_run();

    for (i = 0; i < rpl->neighbor_count; i++)
    {
        PRPL_NEIGHBOR neighbor = rpl->neighbor_list[i];

        if (matching_ranks[i] >= INFINITE_RANK)
        {
            if (neighbor->lastDIOMSG != NULL)
            { /* forget messages from other DODAG iterations */
```

7. Now Right Click on RPL project and select Rebuild. Upon rebuilding, **RPL.dll** will automatically be updated in the respective bin folder of the current workspace.

8. Go to NetSim home page, click on **Your Work**, Click on **DODAG_Visualization_Example**.



9. Run Simulation.

Output:

A plot will open, showing the DODAG when the simulation is started, and the first route is formed between sink node and the sensor. And the DODAG will be dynamically updated. The DoDAG gets plotted in MATLAB as shown below:

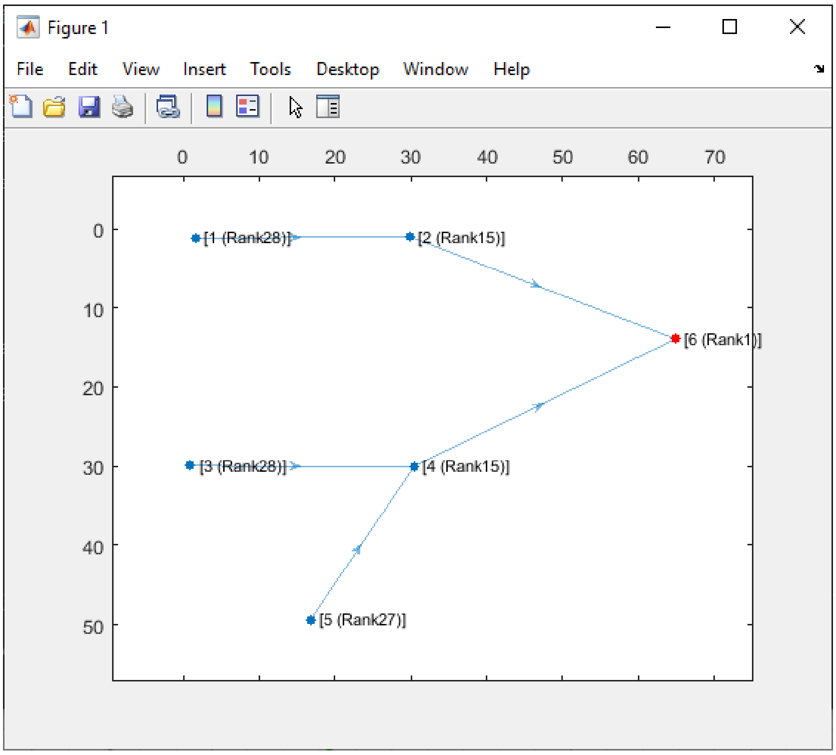


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
0
10
20
30
40
50
60
70
0
10
20
30
40
50
[1 (Rank28)]
[2 (Rank15)]
[6 (Rank1)]
[3 (Rank28)]
[4 (Rank15)]
[5 (Rank27)]