Marvelmind UDP C library and example.

Version 2016.11.30

1. About the library.

Marvelmind UDP C library provides an API and example of receiving coordinates of mobile beacons via UDP from the running Dashboard software.
Supported operating systems:

- Microsoft Windows
- GNU/Linux

2. Building the example.

To build the example on GNU/Linux or another *nix-OS you need to have installed GCC. Then unpack the archive, change directory to unpacked library and run make in console. Then you can execute ./marvelmind_udp_c to watch data from Marvelmind beacons being received.
Prebuilt example for Microsoft Windows is included in the archive. If you want to rebuild it, you may use integrated development environment (such a MS Visual Studio, Code::Blocks etc.): create empty console project and add 3 source files (udp_example.c, udp_marvelmind.h, udp_example.c) into the project and run build. You may need to change the project settings to successfully build it.
Windows version of library uses “winsock” library, you should have it installed on your computer and link to the project. Linkage of winsock is included in library code for MS Visual Studio and CodeBlocks project.

3. Command line options of the example.

You may specify up to three command line parameters to select following options:

Number of parameter	Function	Default	Description
1	Address of beacon	0	Address of beacon whose coordinates should be received. If the address is zero, the example will listen the selected UDP port for streaming data about positions of all mobile beacons without requests. Second parameter in this mode has no effect. If the address is not zero, the example will send requests to selected IP address and UDP port to get position of selected beacon.
2	IP address (URL)	127.0.0.1	Address of host with the running dashboard to send requests in “request-reply” mode (if first parameter not zero). Default value 127.0.0.1 means local host.
3	UDP port	49100	Address of UDP port for sending requests in “request-reply” mode (if first parameter not zero), or for listening in streaming mode (if first parameter is zero).

As you see, running the example without command line parameters starts listening of UDP stream on port 49100.
See examples of using of the command line parameters in appendix.

4. Using the library.

Example of library usage see in the file **udp_example.c**. You can use the library in your own projects by adding file **udp_marvelmind**.c into build, including **udp_marvelmind.h**:

```
#include "udp_marvelmind.h"
```

and your code may follow the sequence:

4.1. Call **createMarvelmindUDP** to allocate memory for library structure (struct MarvelmindUDP). You need to call it before any other usage of the library

4.2. Modify some variables in created structure, if needed. For example, you can change IP address of Dashboard host, UDP port, address of mobile beacon to read, rate of requests of data (in request-reply mode), list of them you can found in file **udp_marvelmind.h**

4.3. Call **startMarvelmindUDP** to tell library to start collecting and parsing data received from Dashboard

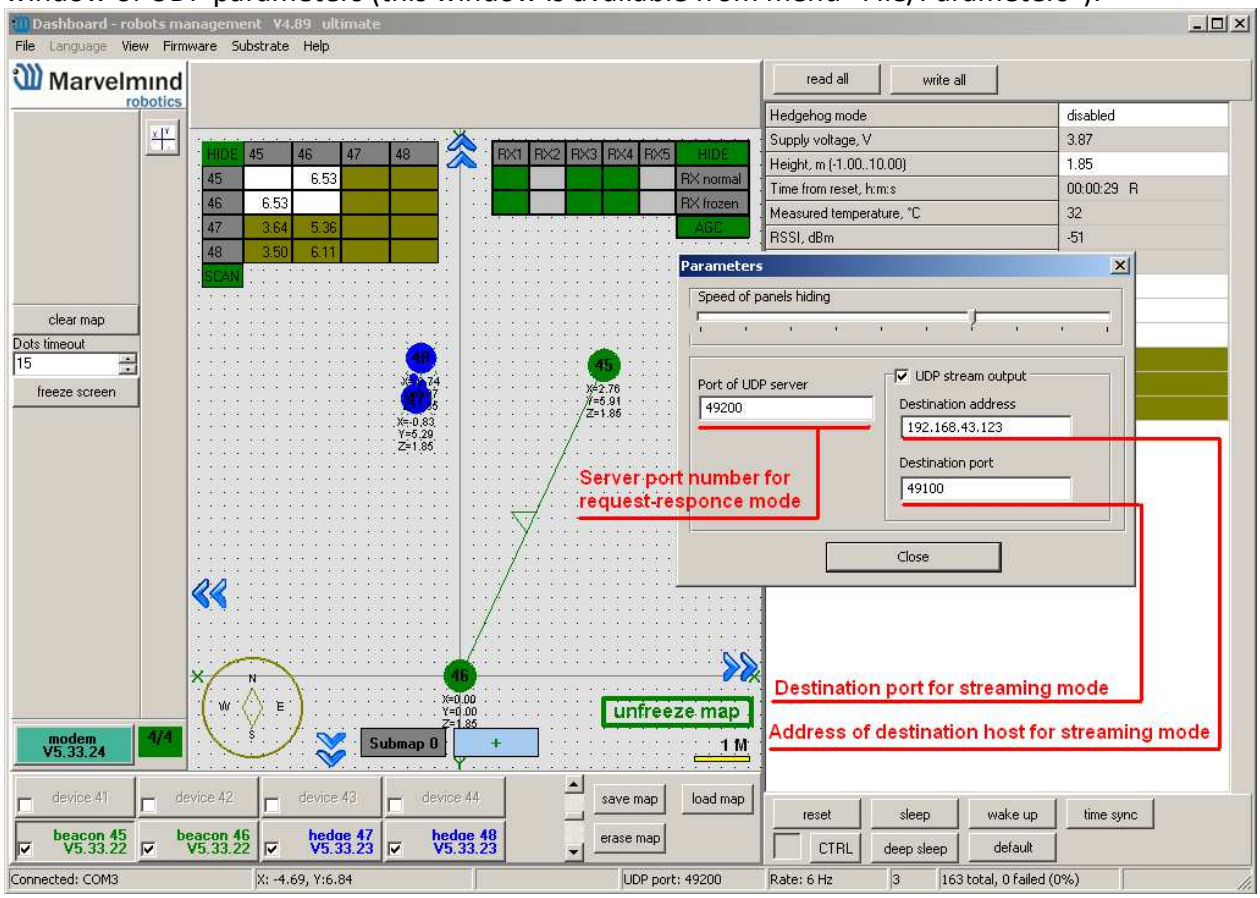
4.4. Get the data: call **getPositionFromMarvelmindUDP** to get 3-axis coordinates of selected beacon or call **printPositionFromMarvelmindUDP** to print it on console output. You can repeat this step

4.5. After usage call **stopMarvelmindUDP** to stop the collecting thread

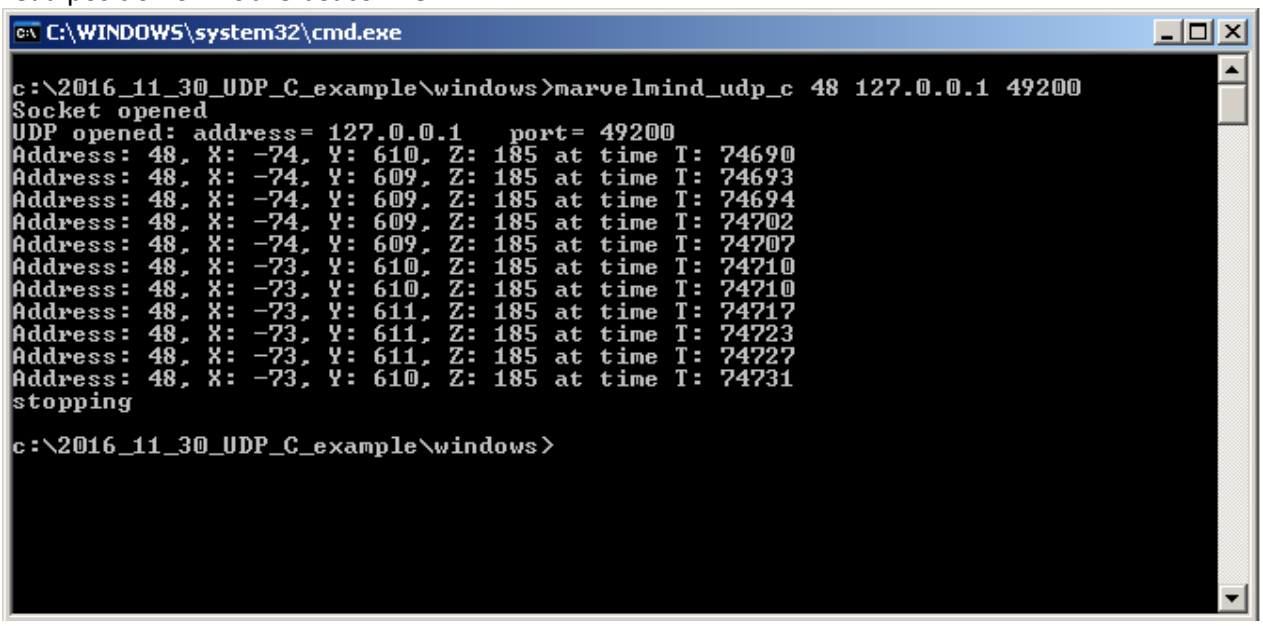
4.6. Call **destroyMarvelmindUDP** to free memory, used by Marvelmind UDP library

Appendix. Examples of usage of the example.

The following screenshot shows the window of Marvelmind Dashboard with test system and window of UDP parameters (this window is available from menu “File/Parameters”):



The next screenshot shows running the example on the same MS Windows computer were the Dashboard is running. According to the command line parameters, the example sends requests to read position of mobile beacon 48.



The next screenshot shows building and running the example on another computer under Linux. Two runs are shown.
First run sends requests to get positions of mobile beacon 48 to the computer with the executing Dashboard.
Second run starts listening of port 49100 for incoming streaming data. The IP address of this Linux computer is selected in dashboard streaming parameters, see the screenshot above.