Plug'n'Play Technology in SpaceWire Network

L. Koblyakova, A. Setkov, K. Khramenkova
SUAI
Saint-Petersburg, Russia
luda_o@rambler.ru, alexsetkov@gmail.com, ksu_khramenkova@mail.ru

Abstract

Modern Embedded Networked Systems are frequently consist of a considerable quantity of devices (nodes) allocated on different distances from each other. It's necessary to organize interconnection between them and to maintain working correctness during all work-time.

Plug'n'Play technology means, that when all system devices had been connected, it will be able to discover and identify network structure by special algorithm, to set logical addresses to nodes, to set routing tables and other necessary system parameters and to provide with this information all others needed devices. At regular work connection and disconnection of new network devices recognized automatically.

The philosophy of the SpaceWire Plug'n'Play technology is to provide maximum compatibility with legacy SpaceWire devices (nodes and routers); to support all forms of SpaceWire network addressing. In most cases, it can be reached by using Plug'n'Play technology.

INDEX TERMS: EMBEDDED NETWORKED SYSTEMS, SPACEWIRE, PLUG'N'PLAY, RMAP.

I. INTRODUCTION

Modern Embedded Networked Systems are frequently consist of a considerable quantity of devices (nodes) allocated on different distances from each other. It's necessary to organize interconnection between them and to maintain working correctness during all work-time.

When all system devices has been connected, it will be able to discover and identify network structure by special algorithm, to set logical addresses to nodes, to set routing tables and other necessary system parameters and provide with this information all others needed devices. At regular work connections and disconnections of new network devices are recognizing automatically. In most cases, it can be reached by using Plug'n'Play technology.[4]

II. MAIN PART

A. SpaceWire

SpaceWire technology is assigned to make communication networks for on-board systems. One of the main aims is provision compatibility with different types of equipment and multifunctional usage of terminal devices and subsystems.

SpaceWire is assigned to transmit data and to control information for on-board systems. It's high-speed standard (2-400 Mbits per second).

SpaceWire is rapidly becoming a de facto communications standard within the space industry. SpaceWire represents a suitable point of convergence for data communication onboard spacecraft due to its simplicity, speed, low-power and reliability, providing the basis

for interoperability between devices and components from different manufacturers and organizations. [2]

SpaceWire network consists of 3 types of elements: node, link, router. Nodes are connected through router and admit to exchange data packets (messages).

B. RMAP

The remote memory access protocol (RMAP) has been designed to support a wide range of SpaceWire applications. Its primary purpose however is to configure a SpaceWire network, to control SpaceWire units and to gather data and status information from those units. RMAP may operate alongside other communications protocols running over SpaceWire. RMAP may be used to configure SpaceWire routing switches, setting their operating parameters and routing table information. It may also be used to monitor the status of those routing switches. RMAP may be used to configure and read the status of nodes on the SpaceWire network. For example, the operating data rate of a node may be set to 100 Mbits/s and the interface may be set to auto-start mode. For simple SpaceWire units without an embedded processor, RMAP may be used to set application configuration registers, to read status information and to read or write data into memory in the unit. For intelligent SpaceWire units RMAP can provide the basis for a wide range of communications services. Configuration, status gathering, and data transfer to and from memory or mailboxes can be supported. [3]

C. SpaceWire & Plug'n'Play

The SpaceWire Plug-and-Play (SpW PnP) is a joint effort together with NASA and other partners in the frame of the SpW PnP Working Group.

The philosophy of the SpaceWire Plug'n'Play technology is to provide maximum compatibility with legacy SpaceWire devices (nodes and routers); to support all forms of SpaceWire network addressing. Designers do not want to try to include all present and future applications by predicting what they will be and specifying their behavior before they exist. History teaches that the future will always bring things you did not predict. Designers attempts to avoid this trap by only defining a minimum required functionality and allowing freedom, within a general framework, for unforeseen devices to be added. Designers also try to keep the protocol as simple as possible in order to minimize design effort and implementation 'cost' (minimizes validation, silicon area, code size and power consumption). Plug'n'Play technology must fulfill the following conditions:

1. The Plug'n'Play technology must provide each device in the system with information of what it needs to route and to interpret packets received from each other device in network.
2. The algorithms must not result in undetermined results or deadlock. Plug'n'Play must allow the network to function as well as possible for as long as possible. The effect of failures should be minimized.
3. Operating with only two types of devices (a router and node), we build arbitrarily big structured network and algorithm Plug'n'Play should work correctly on it.
4. Technology should provide ability, in future, to add new devices. Thus, there shouldn't be any overload in presents.
5. This technology should be implemented with minimal cost; device equipment should take minimal place on a integrated circuit and minimal energy consumption.
6. Plug'n'Play technology should work quickly and shouldn't have great influence on performance of a network.

The main condition is that SpW must be unchangeable. Plug'n'Play is above SpW.

Plug'n'Play isn't standardized. There are several different ways how Plug'n'Play could be implemented. All Plug'n'Play algorithms can be divided on two main types:

- Centralized – a network has one device, which initiates and controls algorithm Plug'n'Play.

- Decentralized – a network has several centers, which controls algorithm Plug'n'Play. In this case, if one or some devices are out-of-work system is workable.

Creating SpaceWire Plug'n'Play technology, we suppose that:

Network structure is arbitrary and unknown and can change at any time due to failures or user intervention. New devices plugged may not be in reset status.

Network discovery shall be executed by an intelligent node, called Network Manager, which detects plug/unplug events of any SpaceWire link, device or subnet and uniquely identifies all devices in the network. [1]

D. Algorithm

Algorithm described below is centralized algorithm (using only one NM). NM can be implemented as PC, node or other SpW device. RMAP protocol is using to configure remote devices.

Fig. 1 represents sequence of commands during the initial network discovery. Digits mean order of commands.

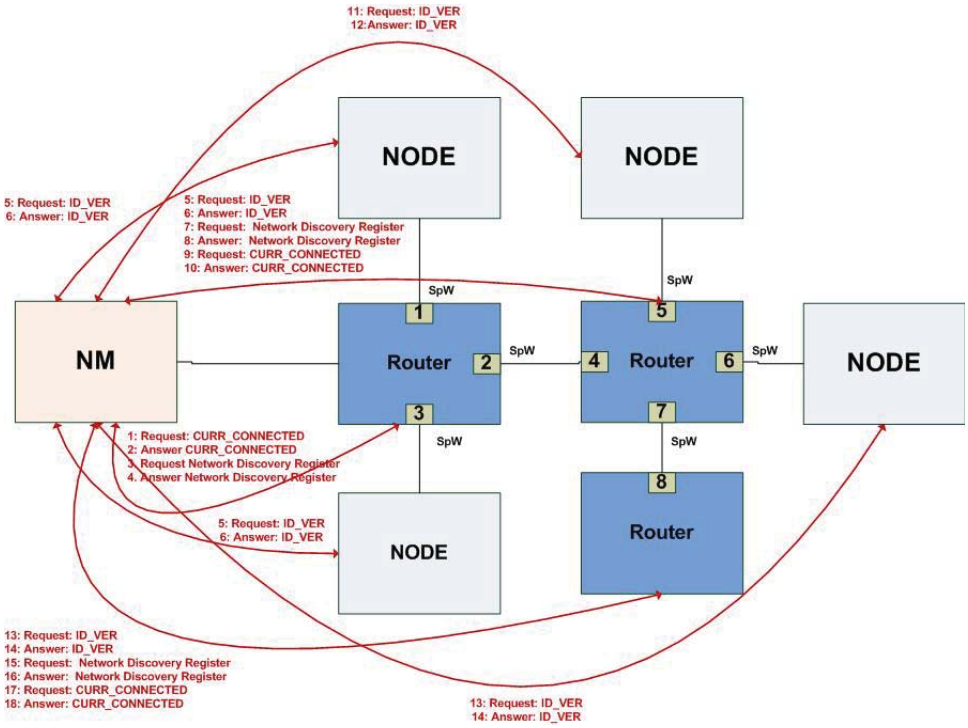

Fig. 1. Algorithm is divided into 2 stages: initial network discovery and work state.

Initial network discovery: NM (Network Manager) is connected to at least one router. NM can be connected through SpW link, COM port or others interfaces.

At the beginning we read version register (ID_VER). If there is a router, than we read Network Discovery Register (indication, that device has been already configured). If it contains 0 value, than write 1 to it. Then we read register of current connections (CURR_CONNECTED). Using this value we recognize what ports of router are in active state. After that, we send packets with version register reading command through all active ports. By reading all received packets, NM configures routing table with following values: for

node - Logical Address and port number to which it has attached; in case of router, address which consists of port number, by which it has been connected, and 0, which is accorded to sending packet to NM.

Whereat, logical address of NM (already reserved) and port number, by what it has been attached, are written.

At each further step, all already discovered routers are being configured.

When all network devices founded and configured, initial network discovery stage is finished, Work state is starting.

During work state NM repeats cycle of actions from Initial stage after some defined period. By comparing CURR_CONNECTED router register, NM can found out disconnection/connection in some network part. If disconnection occurred by some port NM clears all entries this port number in routing tables of all routers. If connection occurred NM configures routing tables according to the type device connected. Work state lasts all working time. Intervals between retries longer than time needed to discovery network.

If during discovery cycle NM receiving no response to its interrogation waits for time-out, and then tries to identify which link occurred disconnection. NM starting network discovery from the beginning with interrogation first device connected.

Disconnection means:

1. Device reset
2. Lost connection (for some time connection doesn't recover)
3. Device switched-off

Temporary disconnection occurred by link failure goes beyond the scope of the article.

Requirements and limitations, putted on our realization:

1. One of the logical addresses should be reserved for NM, for nodes extracted from the remaining.
2. Number of nodes in network is limited by routing table size.
3. Node can be connected only to one router.
4. Nodes should be set up in a such way, to be able to send response packet when it received an interrogation.

NM uses specially allocated memory for intermediate operations to discovery network structure. Each router uses Routing Table and Network Discovery Register (1 byte).

Following variants are foreseen:

1. Detaching device group
2. Attaching device group
3. Cycles

Main aspects:

1. Eliminates all manual configuration
2. Only one NM is used in this algorithm
3. Regionally-logical addressing is not used
4. Router shouldn't be programmed specially
5. Using RMAP (there are no need to use new protocols)
6. Users can manually configure any device just using RMAP commands

Shortcoming: When NM is out-of-working, devices can't be uniquely discovered and configured.

As a result, we have configured network consisting of nodes and routers and using one NM.

III. CONCLUSION

NM admits to identify attached/detached devices in automatic mode, without any human intervention. Thus, we can configure any network, which applied to our demands and limitations. at the expense of simultaneously sending packets to all active ports device identification and setting performed quickly.

REFERENCES

[1] A. A. Setkov, K. N. Khramenkova, article “Plug'n'Play architecture in SpaceWire network”, SUAI, St. Petersburg, 2009.

[2] ESA (European Space Agency), standard ECSS-E-50-12A, “Space engineering. SpaceWire – Links, nodes, routers and networks. European cooperation for space standardization”, ESA Publications Division ESTEC, Noordwijk, The Netherlands, 2003.

[3] ESA, standard ECSS-E-50-11 Draft F, “Remote memory access protocol”

[4] V. Olenev, I. Korobkov, L. Koblyakova, F. Shutenko, article “Remote Memory Access in Embedded Networked Systems”, SUAI, St. Petersburg, 2009.