NERD (Network Enabled Resource Device)

Thomas Denewiler, Robin T. Laird

Space and Naval Warfare Systems Center, San Diego (SSC-SD)

Code 2371, 53406 Woodward Road, San Diego, CA 92152-7383

denewile@spawar.navy.mil, 619-767-4272, http://www.spawar.navy.mil/robots/

Abstract

Network Enabled Resource Devices (NERDs) combine the most common electronic components used in robotic applications into a standard electronics box with “plug-n-play” capabilities. Risk reduction efforts, systems testing and integration, and modifying the functionality of evolving systems becomes greatly simplified by standardizing core hardware and software components; in many cases, minimal software modifications are required to adapt an existing NERD for an emergent application. Internal components include an integral DC-DC converter, a wireless bridge and hub allowing point-to-multipoint communications, an audio/video hardware CODEC, a RISC-based processor with FPGA-based I/O, and a GPS receiver. NERDs can accept 12-36VDC to power all components and are compatible with standard military batteries. Two 8-pin input/output data ports connected directly to the embedded processor allow for a wide range of control flexibility in a variety of applications. Implementations to date include controlling a non-lethal weapons pod on the MDARS-Exterior robot, an intelligent garage-door opener for the exterior robot refueling area, a communications/GPS module for a security team response vehicle, and an embedded controller for intruder detection systems.

1 Introduction

The field of robotics requires the combination of many diverse sensors and actuators working to define the robots' environment and to power and control the robotic system. Historically, for each new application, a different set of components is designed and built to interface to a unique combination of sensors and actuators. These components are required to be integrated and packaged in an enclosure to protect them from dust, water, and other elements harmful to electronics materials.

Experience in research, design, testing and production of robotics systems has shown that common interface components are used for a wide variety of applications. A more efficient process for designing electronics boxes is to develop an architecture that would include the common components in a standard package.

To this end, the Network Enabled Resource Device (NERD) was created. NERDs are electronics boxes which contain six basic components: a RISC-based ipEngine processor with FPGA-based I/O, a wireless bridge and hub allowing point-to-multipoint communications, a VP500 audio/video hardware CODEC, an integral DC-DC converter for power, and the enclosure. Additional room is available in the enclosure for peripheral circuitry such as a GPS receiver or motor controller.

There are numerous benefits of having a standard design for an electronics box:

- Standardized production units – to modify the functionality of a NERD only the firmware needs to be reprogrammed.
- Interchangeability/interoperability – NERDs are self-contained and support "plug-n-play" of payloads, subsystems and platforms.

- Reduced testing costs – systems used in risk reduction efforts for large, expensive robots can be tested on small, inexpensive robots using the same interfaces.

Applications currently using NERDs include:

- Non-Lethal Weapons Pod,
- Remote Start Interface to initiate engine and open garage door for exterior robot,
- Intrusion Detection System (IDS) for portable, unmanned sensors, and
- Response Truck to monitor and coordinate mixed unmanned/manned forces.

The Multiple Resource Host Architecture (MRHA) via a wired or wireless local area network (WLAN) provides command and control for all of these systems.

2 Design Requirements

The NERD is designed to be a low power, compact, and cost-effective system that is robust enough to support existing and future applications. All of the essential input/output components and connectors were included with additional resources reserved for future expansion.

2.1 Power

All of the common components in the NERD require either 5VDC or 12VDC.

Component	**12V**	**5V**
IpEngine	<250mA or	<500mA
Cisco AIR-BR340	1.5A	N/A
BreezeCom	N/A	1.2A
VP500	20mA (audio)	<1.3A (video)
Hub	320mA	N/A

Table 1 – Component Power Requirements

MDARS-E platforms are able to supply approximately 27VDC. The NERDs have two DC-DC converters in them. Both can have 9-36VDC inputs; one supplies 5VDC output and the other supplies 12VDC output.

2.2 Size

Obviously NERDs should be as small as possible. The size of the box is determined mainly by the size of the components that go in it, and to a lesser degree by ease of use and ability to troubleshoot the box. The current box used is 9.0” x 12.0” x 4.4”. In the future, as the commercial market is driving the miniaturization of Commercial-Off-The-Shelf (COTS) equipment, components will be replaced to reduce the size and increase the capabilities of the NERD.

2.3 Functionality

By providing a common set of components used in a variety of robotics applications, the NERD can be employed to address emergent project needs in a timely fashion; it can be easily adapted to meet the needs of a broad range of applications. NERDs are simple to install, configure, use, and troubleshoot.

NERDs have a “plug-n-play” capability that allows them to be swapped out on an as-needed basis. Two NERDs having the same software programmed into them are interchangeable with minimal effort. The IP addresses of the ipEngine, VP500, and Cisco AIR-BR340 are configurable.

The wireless bridges in the NERDs have the ability to act as point-to-multipoint repeaters. In this regard, the NERD performs a dual function: when deployed, it provides a robotics capability for the particular application, and it can increase the range of the WLAN by acting as a network repeater. The WLAN components support operation of the

MRHA. This is very important as the MRHA has the ability to control a variety of the robotics platforms using protocols based on Ethernet UDP/IP and TCP/IP.

3 Components

With the exception of the DC-DC converter board, all of the components found in NERDs are available as COTS products.

3.1 ipEngine

Currently, the main processor used in the NERD is the ipEngine made by BrightStar Engineering. The ipEngine is based on Motorola's PowerPC MPC823 processor. It offers a 50MHz processor, 16MB DRAM, 4MB Flash memory, 10Base-T Ethernet, 16W power output, 16,000 gate FPGA, 132 pin virtual I/O interface, USB host/slave controller, LCD/Video controller and dual RS-232 ports. The ipEngine draws a maximum of 250mA at 12VDC.

3.2 Cisco AIR-BR340

Most mobile robotics applications make use of wireless communications. NERDs use point-to-multipoint 2.4GHz Cisco AIR-BR340 wireless bridges. The bridges offer line-of-sight communications up to 18 miles based on Cisco's tests; experimentally, these devices have a typical range of 0.25 miles on-site due to hilly terrain and loss of line-of-sight. They support 11Mbps maximum transmission. The AIR-BR340 uses direct sequence spread spectrum (DSSS) technology, and does not require an FCC license. The bridges have an output power of 100mW.

The AIR-BR340 requires an input voltage of 12-18VDC. NERDs supply 12VDC to the AIR-BR340. In tests performed by SSC San Diego using a spectrum analyzer, it was found that the bridges transmit at the same output power when supplied either

10.00VDC or 12.50VDC. It is not recommended that the AIR-BR340s be powered by 10VDC. However, 12VDC is sufficient even with the possibility of slight voltage drops due too power consumption by other NERD devices.

3.3 Indigo VP500 Video and Audio Card

The VP500, made by IndigoVision, is a network video device that has video and audio inputs, audio output, and back-channel RS-232 I/O. NERDs use two of the four video inputs and the single audio input. The VP500 is an encoder that grabs frames from a camera or microphone, digitizes and compresses those frames and then packetizes them for Ethernet transmission over IP networks. A VideoBridge 6000 decodes the packets and displays real-time video at the MRHA or another suitable monitor. This system eliminates the need for multiplexors, transmission amplifiers, and correction amplifiers.

3.4 DC-DC Converters

NERDs contain two DC-DC converters, both in the 30W series made by Wall Industries with input voltages of 9 – 36VDC. The SIW24S5-30 has an output voltage of 5VDC and is rated at 6A. This converter powers the VP500, the BreezeCom and the GPS card (if present). The SIW24S12-30 has an output voltage of 12VDC and is rated at 2.5A. It is used to power the ipEngine, the Cisco AIR-BR340, the audio circuit of the VP500, and the hub.

Both DC-DC converters are mounted on a custom interface circuit board designed by SSC San Diego. The converter board has six 5VDC outputs and eight 12VDC outputs. There is an LED power indicator on the outside of the box. In recent testing, the DC-DC converters were put under load using power resistors designed to draw the maximum rated current for each device, and the output voltages showed no drop. In order to conserve power, an ipEngine control line is connected to the DC-DC interface board to turn power on and off to specific components depending on whether they are being used or not. The ipEngine also controls a relay located on the converter board.

3.5 Enclosure

The current NERD enclosure is a waterproof polycarbonate box with dimensions of 9.0” x 12.0” x 4.4”. For the enclosure to be water resistant, a small amount of marine sealant was applied along the outer edges of the connectors to keep dust and water out of the enclosure.

3.6 Connectors

There are seventeen connectors on the outside of the enclosure to interface with all of the components in the NERDs.

Number	Connector	Function
2	DB-9	Comm-1 and Comm-2 for IP Engine
6	SMA	Audio In, GPS Antenna, 4 for BreezeCom Antennas
2	8-pin round	Digital and/or Analog I/O
1	RJ-45 Straight-thru	Ethernet to Hub
1	RP-TNC	Cisco AIR-BR340 Antenna
1	4-pin round	Power In
1	6-pin round	Power Out to 2 Cameras and 1 Microphone
2	BNC	Video In
1	On/Off Switch with LED	Power In Control and Indicator

Table 2 – NERD Connectors

4 SMART Software Architecture

NERDs are programmed using the MPRS SMART (Small Robot Technology) software architecture. The SMART software is an extension of the MDARS MRHA and is compatible with MRHA-compliant systems (platforms and controllers).

4.1 Framework

The SMART software architecture borrows concepts from both the MRHA and the Joint Architecture for Unmanned Ground Systems (JAUGS). It uses the underlying MDARS MRHA message format and a similar approach to function-oriented operation. From JAUGS it borrows the concept of software components as functional agents that are responsible for executing predefined operations such as driving, navigating, communicating, etc. The MPRS SMART software architecture is intended to be efficient, adaptable, and modular: efficient in terms of message processing, adaptable to a variety of applications, and modular in terms of support for adding capabilities in response to new requirements.

4.1.1 Functional Agents

A *functional agent* is a conceptual entity that is capable of performing a specific set of operations. An agent must perform application-specific processing (e.g., monitor sensors, sample input devices, etc.), and it must also respond to incoming messages received from other agents. An agent is implemented as a computational process. Multiple agents can execute on a single computer as multiple concurrent processes.

4.1.2 Domains

A *domain* is a logical collection of functional agents that inter-operate and is typically represented by a complete system such as a robot and its controller(s). Agents represent subsystems such as a drive controller, an operator control unit, or a sensor data collector.

4.1.3 Messages

Messages are requests for information or requests for operation. There are three categories of messages: standard MRHA application messages (e.g., status request, set mode, play sound, etc.)[1], standard SMART network management messages (e.g., register and unregister), and non-standard application-specific messages that extend the MRHA message set. All agents must process the standard MRHA application message set and the SMART network management message set.

4.1.4 Dynamic Resource Discovery

The SMART architecture supports dynamic discovery of resources (i.e., agents). This allows SMART systems to dynamically configure themselves to form networks of cooperating agents within and across domain boundaries. The process is straightforward and does not rely upon a single "coordinating" entity such as a supervisor or master controller[2]. This avoids the rather large problem of what to do if the coordinating entity dies. Dynamic resource discovery under the SMART architecture is based upon the concept of a *registration table* that maintains the current known state of the system in terms of available (alive) agents.

4.1.4-1 Registering

At startup, each agent broadcasts its presence to the network.[3] The broadcast message is transmitted a predefined number of times or until it is acknowledged. The

[1] The MRHA Interface Design Document (IDD) lists all of the standard MRHA application messages.
[2] The only exception to this is when a Portal agent is used. All agents would require the Portal to be running in order to register with other agents.
[3] This assumes the use of some sort of network (not necessarily Ethernet, but a network that supports broadcasting of data).

broadcast data includes the agent's domain, class, network address, and application message port identifier.

When an agent receives a registration request, it adds the registration data to a table. Duplicate entries are simply overwritten. The registration data is used to route future message transmissions to the registering agent. The agent that has successfully received the registration request then sends an acknowledgement to the registering agent. The acknowledgement message includes the entire registration table of the agent that responds to the registration request. By including the entire registration table in the acknowledgement message, the odds are increased further that the registering agent will be made aware of all other agents on the network.

4.1.4-2 Unregistering

When an agent anticipates leaving the network, it sends a broadcast message to all other agents indicating that it is terminating. All remaining agents will remove the terminating agent from their registration table, and the terminating agent will no longer be reachable. The broadcast message is not acknowledged as the originating agent has terminated.

4.1.4-3 Registering Across the Internet

Inter-agent communications assumes the capability of sending broadcast messages across the LAN that connects SMART components. Most routers used on LANs do not allow broadcast messages to propagate beyond the local subnet. In order for the resource discovery process to succeed across the Internet, a *portal* is used that collects and distributes registration information between remote agents. The portal is a pre-defined agent that is located at a well-known communications (IP) address. Instead of broadcasting registration requests to the local network, agents will just send registration requests to the portal. The portal will respond with registration data on all known agents.

4.1.5 Pre-Defined Agents

There are a number of pre-defined SMART agent classes: *User*, *Controller*, *Driver*, *Observer*, *Navigator*, *Monitor*, *RADAC*, and *Portal*. Each agent class has a specific (implied) set of operational requirements in terms of the messages it must process and the capabilities it must provide. Agents are typically configured in an a priori fashion to rely upon other agents for services or information. For example, a Controller expects to find and use a Driver within its domain that manages platform mobility for the robot defined by the domain. In general, however, an agent will accept data requests and commands from any other agent within its domain. This allows for cooperative and subsumptive relationships between agents in a completely dynamic fashion. A Driver agent, for example, can be commanded by any other agent in the network to move the robot to a specified location.

4.1.5 Agent Configuration

SMART agents are configurable. Configuration information is typically stored on non-volatile media such as a disk file or flash memory. At start-up, an agent will read the configuration settings and adjust operation accordingly. The concept is similar to supplying run-time parameters to executable programs under popular desktop operating systems.

4.1.5-5 Sample Configuration

Below is a sample configuration file for an Observer agent in the SMART1 domain using a portal to access the Internet:

```
mydomain=”smart1”
```

```
myclass="observer"
mrhadev="PORT(5002)"  // IPV4 by default
netmandev="IPV4(255.255.255.255):PORT(5123)"   // broadcast
address
```

4.1.6 MRHA and JAUGS Compatibility

SMART messages use the same protocol as MRHA-to-remote resource messages. The *Control Flags* field has been extended to include the use of a *Command Message* flag and a *Response Message* flag. Outgoing SMART command messages have the *Command Message* bit (2) set, and incoming SMART response messages have the *Response Message* bit (3) set.[4]

SMART agents communicate using standard BSD Unix (UDP-based) sockets over well-known port numbers. By default, MRHA messages are received over port number 5001, while SMART network management messages are received over port 5123. The port numbers are configurable.

The SMART network management software implements the Message Routing Service (MRS) described in the JAUGS Reference Architecture (GOA class 3L interface). Agents (which relate to JAUGS components) use the SMART network management services to send and receive application-specific messages. Application messages are defined by the MRHA IDD. Communication at the agent level using the MRHA IDD messages corresponds to the GOA class 4L interface described by JAUGS for component level communications.

Under the SMART architecture there is no logical distinction between intra-agent and inter-agent communication as there is with JAUGS at the node level. All agent communication uses the same message protocol no matter the physical or logical

[4] The *Control Flags* field is eight (8) bits wide. The least significant bit (LSB) is bit zero (0).

decomposition of the system – an agent exists at the atomic level and cannot be logically decomposed.

5 Applications

5.1 Non-Lethal Weapons Pod

The initial NERD was used to control a paintball gun pod located on MDARS-E #3. The box is located inside the shroud that houses the gun, the motor and the electronics. The NERD controls the pan and tilt functions of the assembly as well sending fire commands to the paintball gun. It receives approximately 27VDC from the platform.

5.2 Remote Start Interface

A NERD is located inside the garage of the Butler Hut on-site. A power supply is in place to give 24VDC to the NERD. One camera inside the garage gives a view of what is happening inside and another camera and a microphone are mounted outside the Butler Hut. The NERD is hardwired into the backbone of the LAN in order to receive commands from the MRHA. The Cisco AIR-BR340 in the NERD can then continue communications to the MDARS-E platforms when they enter the garage

5.3 Intrusion Detection Systems (IDS)

NERDs will be installed at the corner of Robart Rd. and Woodward Rd., at the north end of the test track loop on-site and near Battery Woodward in order to control IDS systems. The NERD on the corner of Robart Rd. and Woodward Rd. will also be capable of turning on the emergency lights along Woodward Rd. from the MRHA whenever robots are patrolling the area. At the north end of the test track loop is a trailer that has power installed that would be a good site for installing a Perimeter Security Radar System (PSRS) because of the relative flatness of the terrain in that area

5.4 IDS Controlled Non-lethal Weapons Pod

A NERD will be used to control the coordinated actions of an IDS and a non-lethal weapons pod. The bearing and range received from the IDS will be used to control the bearing and tilt of the non-lethal weapons pod. A message will be sent to the human operator located in the MRHA Command Center with an explanation of the situation before any response is made concerning the suspected intruder.