RoadNarrows Intelligent Systems
RoboSight User Manual v1.3.0

March 29th 2010

www.rnisys.com

Authors

Dave Hygh & Robin Knight
RoadNarrows Intelligent Systems
1151 Eagle Drive #140
Loveland, Colorado USA 80537

Editor

Kim Wheeler
RoadNarrows Intelligent Systems

Trademark Acknowledgements

Legal Notice

Table of Contents

Figure Index

1 INTRODUCTION

1.1 Overview

RoboSight™ by RoadNarrows Intelligent Systems offers an affordable, but powerful vision recognition system, ideal for robotic platforms and small embedded applications. Powered by the CogniMem CM-1K neural network chip by Recognetics, complex visual objects are readily detected and categorized from patterns previously trained by the user or the embedding system. The recognized visual categories can then be fed to the higher-level functions of the controlling system to guide goal decisions, object localization, platform movement and navigation, and environment manipulation.

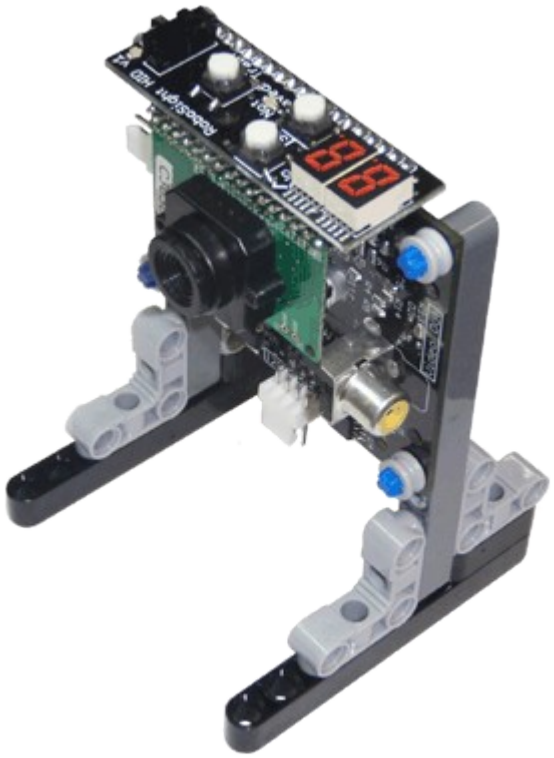


Figure 1: RoboSight

1.1.1 No Programming Required

Rather than require custom programming for each specific visual application, a simple train, then run cycle is all that is needed. Training can be accomplished either by the user using the attached RoboSight Human Interface Device, or through the serial interface connected to either a PC or the embedding system.

1.1.2 Key Features

- Powerful embedded CogniMem CM-1K neural network chip.
- OmniVision OV6630 camera, 352x288 resolution at 60 frames per second.
- Attached Human Interface Device (HID) with push buttons and display LEDs for stand-alone use (N/A for Basic).
- On-board non-volatile memory to store trained neural network patterns and configuration parameters.
- Read-only I^2C interface (N/A for Basic).
- RS-232 command-line interface for expanded features of RoboSight.
- PIC18F4620 main processor with 64KB of program memory.
- Host Software Development Kit (SDK) for host controlled applications.
- Open source main firmware, host library and examples.

- Accessories.

1.2 Safety Precautions

- **Power**: The RoboSight requires a regulated 5VDC, 1Amp+ power supply. Overvoltage or overcurrent can damage the electronics.
- **Connections**: All connection must be made with the RoboSight powered off. Otherwise damage may occur.
- **Heat**: The RoboSight, like all electronic devices, generates heat when powered. Heat can present a fire hazard. Keep the RoboSight in a well ventilated area and away from flammable materials. Exercise Fire Prevention.
- **Mechanical**: Do not apply excessive force to the RoboSight. Otherwise damage may occur.
- **Use**: Power off the RoboSight when it is not in use.

1.3 A Cleaner Environment

Think about the end-of-life event for your RoboSight and its accessories.

1.3.1 Reusing

Donating used (but still operating) electronics for reuse extends the lives of valuable products and keeps them out of the waste stream for a longer period of time. Reuse, in addition to being an environmentally preferable alternative, also benefits society. By donating your used electronics, you allow schools, nonprofit organizations, and lower-income families to obtain equipment that they otherwise could not afford.[epa1]

1.3.2 Recycling

If reusing or donating the RoboSight is not a viable option, then send the components to a recycling center. Recyclers recover more than 100 million pounds of materials from electronics each year in the U.S. Recycling electronics helps reduce pollution that would be generated while manufacturing a new product and the need to extract valuable and limited virgin resources. It also reduces the energy used in new product manufacturing.[epa1]

1.4 Packages

The RoboSight comes in several different packages. Also, various accessories can be purchased separately.

- **Basic**: The RobotSight Basic is a bare-bones, but fully capable system intended for hobbyist and enthusiast comfortable electronics and/or firmware development.
- **Standard**: The RoboSight Standard is fulled featured RoboSight that can be readily adapted to the user's applications.

- **Professional**: The RoboSight Standard plus additional accessories to support standalone mode and host software development.

Table 1 Below itemizes the three supported RoboSight package configurations.

Note: Packages and accessories are subject to change. See www.roadnarrowsrobotics.com for the latest information on available RoboSight products.

RoboSight Item	Basic	Standard	Professional
RoboSight Standard Main Board		✓	✓
RoboSight Basic Main Board	✓		
OmniVision OV6630 camera	✓	✓	✓
HID		✓	✓
Serial port	✓	✓	✓
Power/Ground + I^2C port	Power/Ground Only	✓	✓
Video out port	✓	✓	✓
RS-232 DB9 cable		✓	✓
4-Wire Power/I^2C cable		✓	✓
RCA cinch video composite out cable			✓
Composite video to USB converter			✓
Serial to USB converter			✓
5VDC Power Supply			✓
Simple Stand		✓	
Tripod + Enclosure			✓
Preloaded Firmware	Bootloader + Main Standard	Bootloader + Main Standard	Bootloader + Main Standard
Firmware and software source code available on-line	✓	✓	✓
Support CD			✓
90-Day Warranty Against Hardware Defects	✓	✓	✓

Table 1: RoboSight Package Configurations

1.5 RoboSight Hardware

1.5.1 Main Board

Figure 2 shows the front side of the RoboSight Standard Main Board without the camera and HID attached. Figure 3 depicts the layout of the Main Board with callouts.

Components:

- PIC18F4620 main processor
  - 8MHz clock
  - 3968 bytes SRAM Memory
  - 64KB Flash Program Memory
  - 1KB EEPROM Data Memory
- PIC16F766 HID co-processor (not applicable for Basic)
- CogniMem CM-1K neural network chip
  - 1024 neurons
  - parallel processing
- Non-volatile memory to preserve register configuration and neural network data
  - 512KB Flash
  - Maximum of 992 neurons
- RS-232 serial port
- Power and I^2C port (I^2C not applicable for Basic)
- Composite video out port
  - PAL format
  - Monochrome gray scale
- Camera header
  - Compatible with COMedia C30xx boards with OmniVision camera systems
- Power Indicator LED
- Mechanical attachment holes

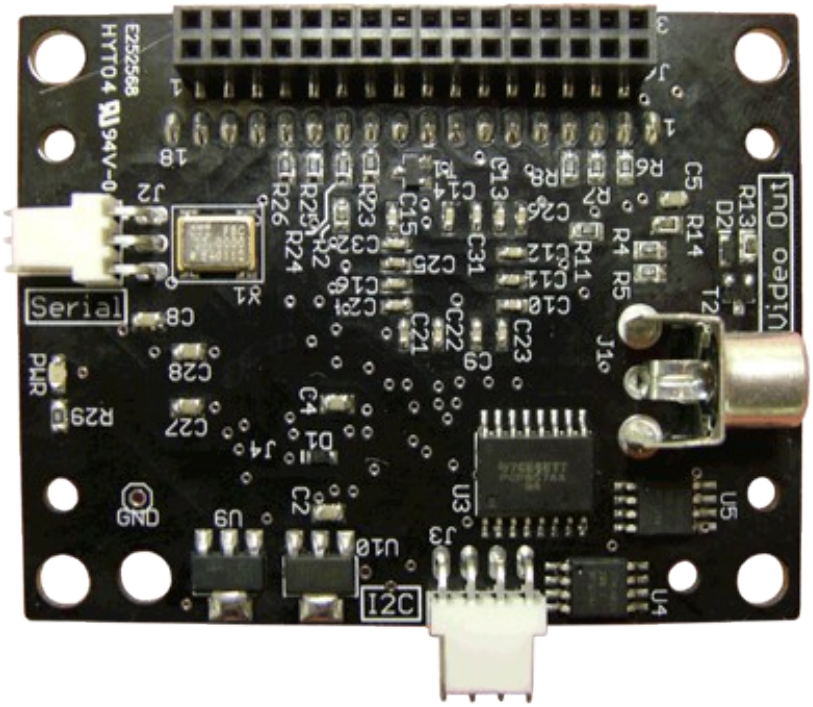


Figure 2: RoboSight Standard Main Board Front View

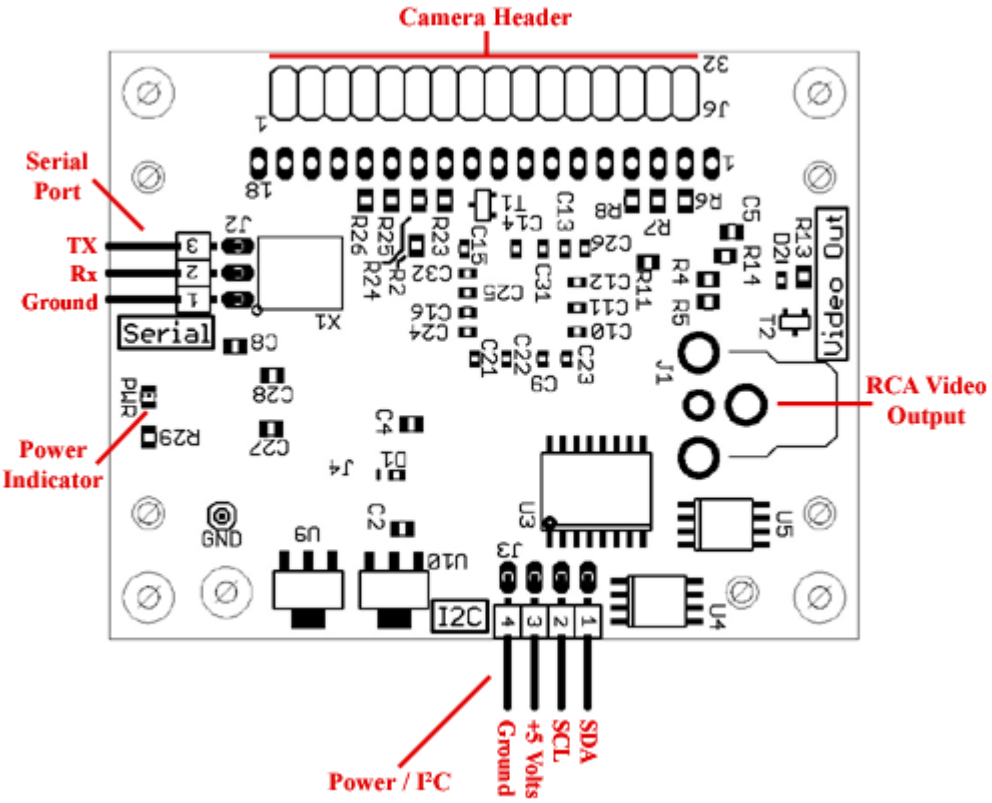


Figure 3: RoboSight Main Board Layout

1.5.2 HID Board

The Human Interface Device (HID) Board provides standalone (i.e. without a host) operation of the RoboSight. Figure 4 shows a of top view image of the HID and Figure 5 the HID layout.

Note: HID is not applicable for RoboSight Basic.

Components:

- Learn/Run slider switch
- Learn Mode indicator LED
- Run Mode indicator LED
- Scroll up and down buttons
- Train button
- Not Saved indicator LED
- Two 7-segment LEDs to provide user feedback and display status information

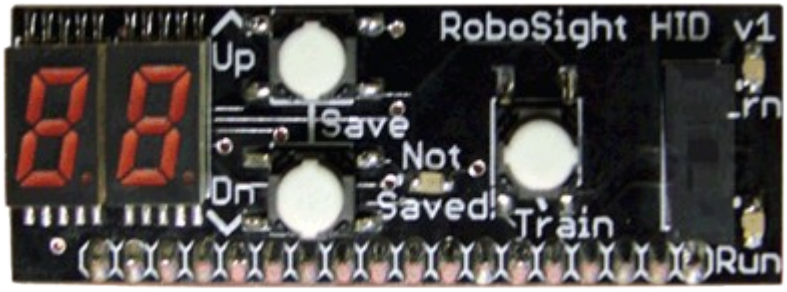


Figure 4: HID Board Top View

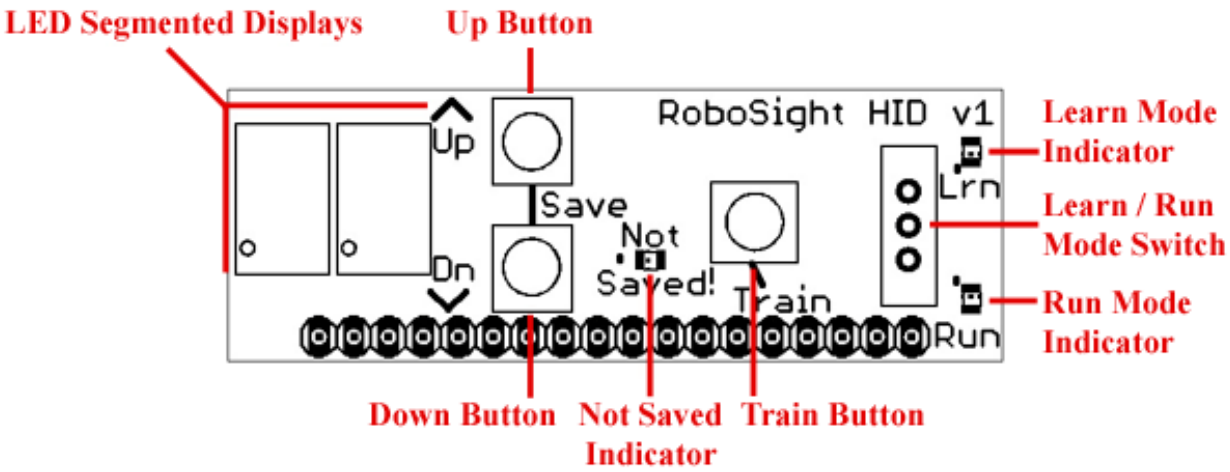


Figure 5: HID Layout

1.5.3 Camera Board

Figure 6 is an image of the C3038 Camera Board. The Camera Board interfaces to the RoboSight Main Board through the camera header.

Components:

- OmniVision OV6630 CMOS image single-chip video/imaging camera sensor
- I^2C control interface
- Digital video output UV bus
- Vertical Sync output
- Video analog output (75 ▪ monochrome)
- f4.9mm, F2.8 FOV34.4x20.7º
- Variety of lens with different performances are available

Figure 6: C3038 Camera Board

1.6 RoboSight Firmware

1.6.1 Bootloader Firmware

The RoboSight is preloaded with a bootloader. The bootloader allows firmware updates or new firmware applications to be downloaded to the Main Board's PIC18F4620 through the serial interfaces. No specialized programmer is need. See Appendix Firmware Bootloader.

For advance users that will develop their own firmware applications, the RoboSight Programming Board accessory is available. The Programming Board is compatible with the Microchip ICD 2 programmer.

Warning: Programming the RoboSight with the Programming Board will erase the bootloader. This is a design limitation of the PIC18F4620 designed.

1.6.2 Main Standard Firmware

The Main Standard firmware comes preloaded on the RoboSight Standard and Basic Main Boards. The firmware automatically detects the underlining hardware. If a Basic Main is detected, the firmware will automatically disabled some features since the features cannot be supported by the board hardware.

Features:

- Auto-detection of RoboSight Standard or Basic hardware.
- Boot-up initialization and restoring configuration from non-volatile memory.
- Command line shell.
- I^2C category output addressing (N/A for Basic Main).
- CogniMem configuration management.
- Camera configuration management.
- Operational parameters for controlling RoboSight behavior.
- Neural network data management.
- HID control master (N/A for Basic Main).

1.7 Host Software

A suite of host software is available to interface with the RoboSight. The software provides extended functionality to the RoboSight for more specific or powerful applications. The software has been designed to run on both embedded processors and full-featured computers.

1.7.1 SDK

The Software Development Kit (SDK) provides programmers with a quick start to develop applications.

- rnr/*.h - RoadNarrows Robotics common project header files.
- librnr - RoadNarrows Robotics common library.
- libserial - RoadNarrows Robotics RS-232 serial library.
- RoboSight/*.h - RoboSight specific header files.
- libRoboSight - The core RoboSight serial interface library.
- Examples - Example applications.

1.7.2 Applications

A full set of host applications have been or are being developed. Some are open-source and can be modified as per the EULA detailed below. Table 2 Summarizes the status of some of the identified applications.

Application	Status	Description
rsFwDownloader	Available	A command-line application to download firmware Hex files through the serial interface to the RoboSight via the RoboSight bootloader.
rsShell	To Be Written	A command-line shell providing raw and expanded commands to the RoboSight over the serial interface. The shell can also serve as a console.
rsVideo	To Be Written	A GUI application to provide video display and camera control of the RoboSight's video. The video stream is fed to the host by the RoboSight composite video to USB converter accessory.
simple_searcher	Available	The *simple_searcher* example drives the panning and zooming of the RoboSight's Region of Interest. The program searches for the specified neural network categories in the video stream. An SDK Example Application.
simple_nav	Available	The current version of *simple_nav* reads the output from the *simple_searcher* application to execute robotic behaviors based on the identification and location of objects. The current version only supports the Khepera III robot with a KoreBot extension. An SDK Example Application.
rsProxy	To Be Written	An IP proxy server daemon that provides proxied hardware access to applications over an IP network. Multiple applications may connect to a single proxy server.

Table 2: Future RoboSight Host Applications

1.7.3 Supported Host Platforms

Supported host platforms require POSIX compatible headers and libraries;

Architecture	Description
i386	Linux with any Intel (backwards) compatible 32-bit architectures.
x86_64	Linux with any AMD 64-bit architectures.
armpxa	Linux XScale PXA Arm architectures.
cygwin	Windows systems with installed cygwin.
osx	Future support for Mac OS-X platforms.

1.8 Free Software Public License and Warranties

RoadNarrows Intelligent Systems supports the concepts of open and free software. Open source protects customer investment and facilitates a viable user community. As defined and specified below, the RoboSight core firmware and software are licensed under version 3 of the GNU Lesser General Public License (LGPLv3).

The LGPL is intended for software libraries, rather than executable programs. Basically it protects RoadNarrows Intelligent Systems RoboSight open firmware and software while granting end users the freedom to convey their *Combined Work* under terms of their choice.

1.8.1 Definitions

Library	The RoboSight Main and HID firmware and the host SDK are open source and herein referred to as the *Library*. The *Library* is fully owned by RoadNarrows LLC or its subsidiaries. The *Library* is the covered work governed by the license and warranty summarized in Section 1.8.2.
Application	An *Application* is any work that makes use of an interface provided by the *Library*, but which is not otherwise based on the *Library*.
Combined Work	A *Combined Work* is a work produced by combining or linking an *Application* with the *Library*. The particular version of the *Library* with which the *Combined Work* was made is also called the “Linked Version”.

1.8.2 The License and Warranties

[see next page]

RoadNarrows Intelligent Systesm End User License Agreement

LGPLv3 License:

Open source provided by RoadNarrows Intelligent Systems is covered under the GNU Lesser General Public License (LGPLv3).

See http://www.opensource.org/licenses/lgpl-3.0.html for the complete specification of the license

In summary:

The LGPLv3 grants to you the four following freedoms:

1. The freedom to run the *Library* for any purpose.
2. The freedom to study how the *Library* works and adapt it to your needs.
3. The freedom to redistribute copies so you can help your neighbor.
4. The freedom to improve the *Library* and release your improvements to the public, so that the whole community benefits.

You may exercise the freedoms specified here provided that you comply with the express conditions of this license. The principal conditions are:

1. You must conspicuously and appropriately publish on each copy distributed an appropriate copyright notice and disclaimer of warranty and keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the *Library* a copy of the GNU Lesser General Public License along with the Program. Any translation of the GNU Lesser General Public License must be accompanied by the GNU Lesser General Public License.
2. If you modify your copy or copies of the *Library* or any portion of it, you may distribute the resulting *Library* provided you do so under the GNU Lesser General Public License. However, programs that link to the *Library* may be licensed under terms of your choice, so long as the *Library* itself can be changed. Any translation of the GNU Lesser General Public License must be accompanied by the GNU Lesser General Public License.
3. If you copy or distribute the *Library*, you must accompany it with the complete corresponding machine-readable source code or with a written offer, valid for at least three years, to furnish the complete corresponding machine-readable source code.

RoadNarrows Intelligent Systems EULA (Continued)

Warranties:

In no event shall the authors, RoadNarrows Intelligent Systems or members/employees of RoadNarrows Intelligent Systems or distributors of this software be liable to any party for direct, indirect, special, incidental, or consequential damages arising out of the use of this software and its documentation, even if the authors or any of the above parties have been advised of the possibility of such damage.

The authors and RoadNarrows Intelligent Systems specifically disclaim any warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The software provided hereunder is on an “as-is” basis, and the authors and distributors have no obligation to provide maintenance, support, updates, enhancements, or modifications.

2 RoboSight Setup

2.1 Unpacking And Inspection

Open the packaging and verify that the contents contain all of the items expected and listed on the enclosed invoice. The exact list of items depends on the RoboSight package and accessories order. Visually check each item for any shipping and handling damages.

2.2 Hardware Assembly

The following assembly instructions steps are for the RoboSight Standard Package. The steps will vary slight for other RoboSight packages and accessories,

- **Warning**: Handle the boards from the edges.
- **Warning**: Avoid static discharge to prevent damage to any electronics.

2.2.1 Assembly Step 1: Camera Board

Attach the Camera Board to the RoboSight Main Board. There is a double row female header near the top of the front side of the Main Board. Insert the double row of pins found on the back side of the Camera Board into the header. Apply even force when inserting. Do not apply any torque or undue force while inserting to prevent any pins from bending or breaking.

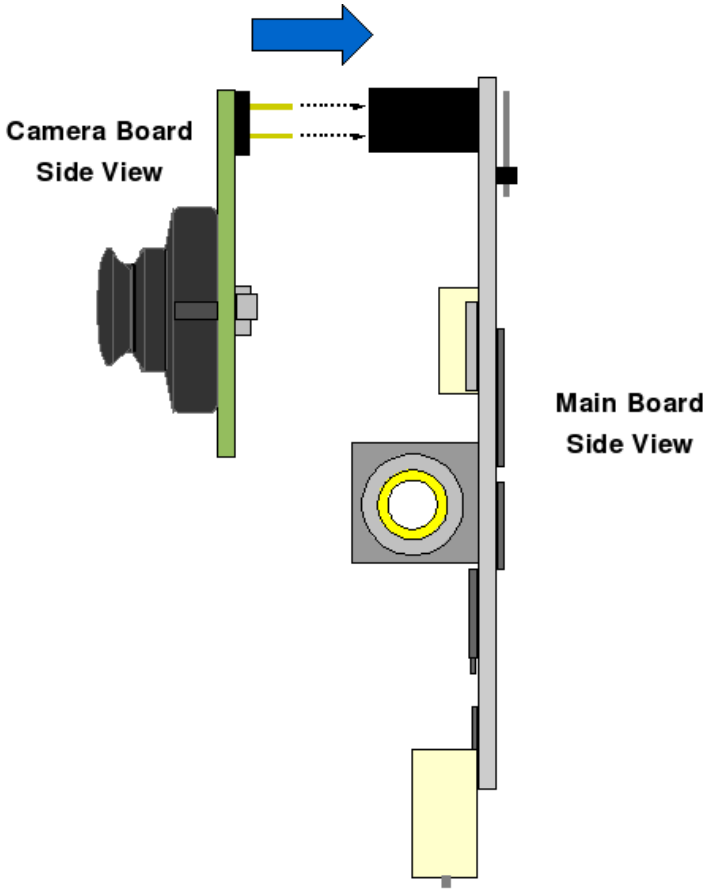


Figure 7: Camera Board - Main Board

2.2.2 Assembly Step 2: HID Board

Attach the Human Interface Device (HID) Board to the RoboSight Main Board. There is a single row male header pins near the top of the back side of the Main Board. Insert the single row female header found on the bottom side of the HID Board onto the pins. Apply even force when inserting. Do not apply any torque or undue force while inserting to prevent any pins from bending or breaking.

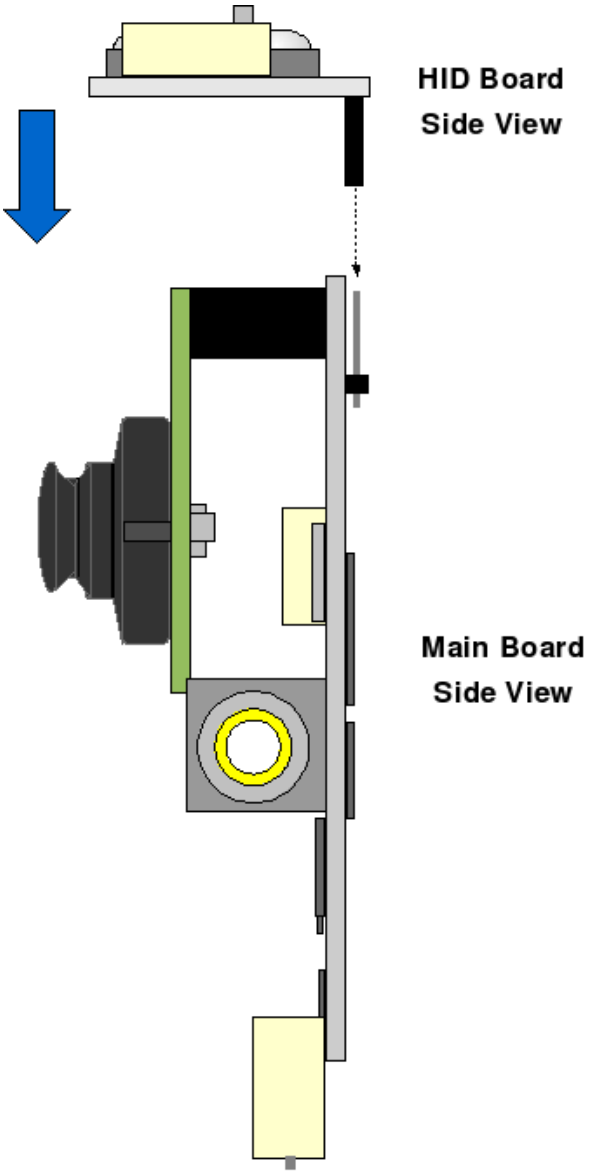


Figure 8: HID Board - Main Board

2.2.3 Assembly Step 3: Cables

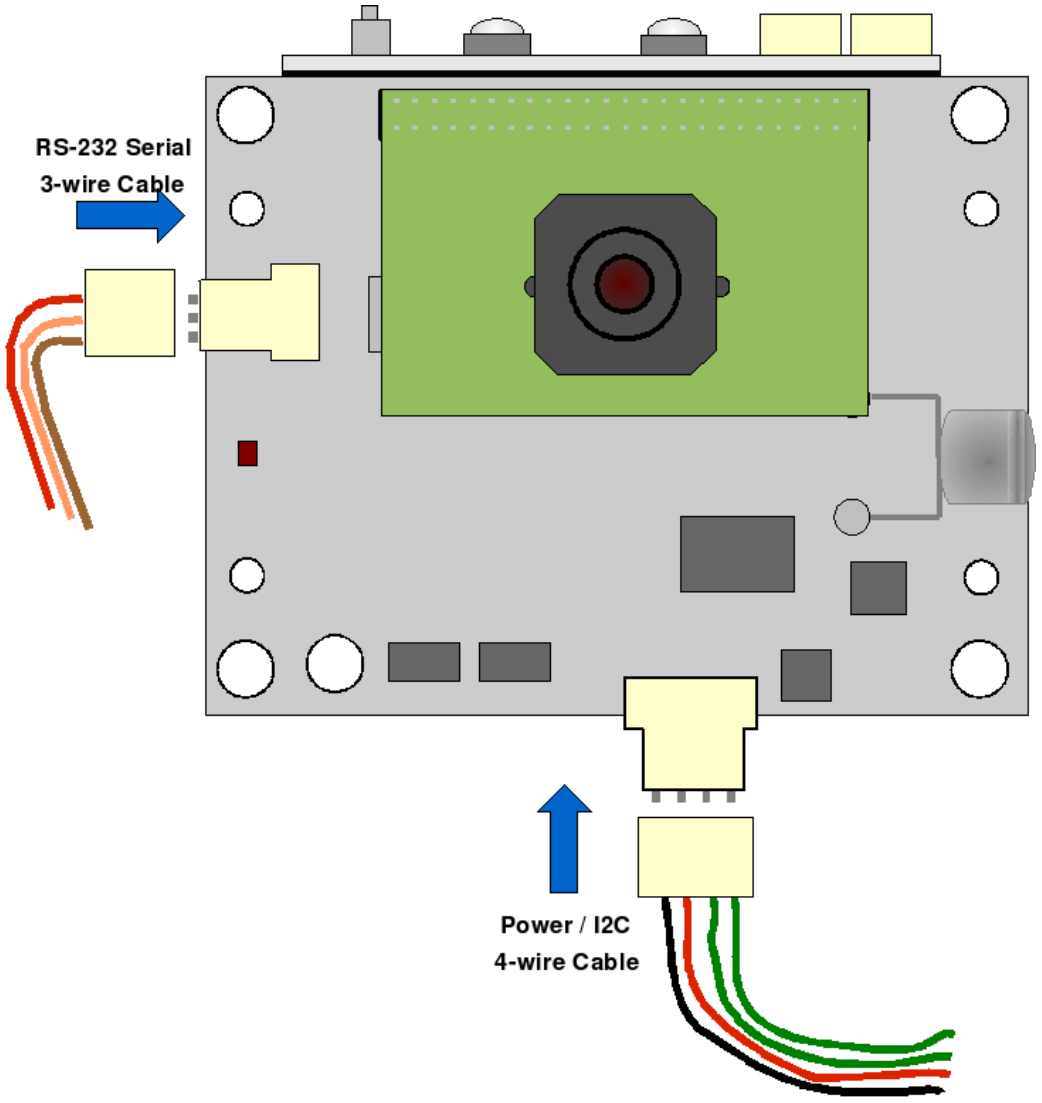


Figure 9: Cables - Main Board

Connect the RS-232 3-wire cable to the RoboSight serial port. A female DB9 connector connects to a null-modem (cross-over) serial cable (not provided) attached to the host computer. That is, Ground to Ground, RX to TX, TX to RX.

Connect the Power/I^{2}C 4-wire cable to the RoboSight Power/I^{2}C port.

- **Warning**: Do not have any electrical power supplied to the RoboSight while inserting the cables.

Connect the +5V power wire (pin 3) to the +5V terminal of the power supply (power supply is not included). Next, connect the ground wire (pin 4) to the ground terminal of the power supply. The 5VDC power supply should be rated at the minimum of 1 amp and be regulated to prevent overvoltages and/or overcurrents, which can damage the RoboSight electronics and to provide sufficient power to prevent undervoltage and/or undercurrent “brownouts”.

2.2.4 Assembly Step 4: RoboSight Stand

Using the provided set of plastic construction pieces, erect a stand for the RoboSight. The large through holes on the RoboSight Main Board are compatible with the plastic pegs. There are various ways to do it – be creative.

2.3 Verify

2.3.1 Power On

Power on the RoboSight. The Power Indicator red LED should be lit on the front side of the Main Board. After a brief period, the HID shows the following.

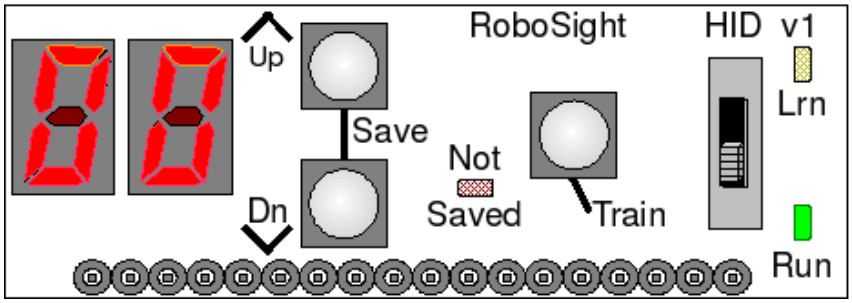


Figure 10: HID Initial Run Display

2.3.2 Serial Interface

With the RoboSight powered off, connect the RoboSight serial DB9 connector to a cable connected to a host computer. Start your favorite terminal emulation program (e.g. picocom, Tera Term, HyperTerminal). Configure the serial device (comm port) to **115200-8-N-1**. Power on the RoboSight. The RoboSight boot up strings should be displayed followed by the RoboSight '->' prompt. Enter a few commands. See Appendix Serial Interface Reference for a complete description of the RoboSight serial interface.

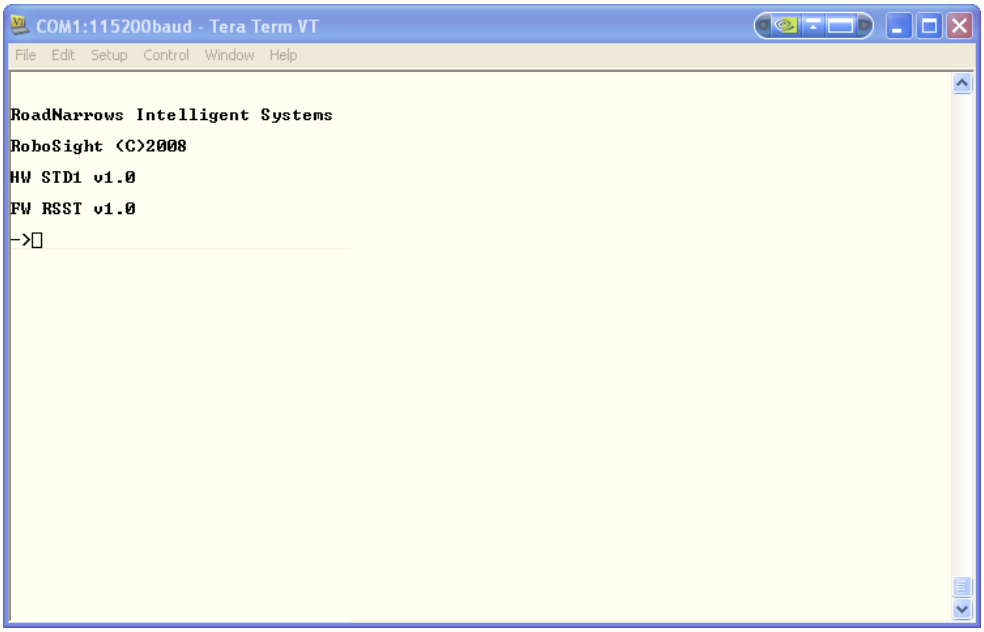


Figure 11: Tera Term

2.3.3 Video

Connect an RCA cinch video cable (not supplied) between the RoboSight video out port and the video display device (not supplied). The video display must accept PAL 50Hz composite signaling.

Adjust the focus of the RoboSight camera by twisting the camera lens left or right as appropriate. For more advance video image controls see the Camera Registers in Appendix RoboSight Registers.

2.4 Done

You are good to go. Happy Visions!

3 CogniMem

3.1 Overview

The CogniMem CM-1K is a fully parallel silicon neural network ASIC. It has an automatic adaptive learning model and n best category pattern matching output. The parallel architecture provides constant learning and recognition time regardless of the number of connected neurons. Rendered in silicon and not running a single line of code, the CogniMem provides true real-time pattern recognition capabilities for a wide range of applications.

Figure 12: CM-1K

The CogniMem artificial neural network is a multilayer perceptron, feed forward network that maps the set of input patterns to an output set of n best categories with associated distances (confidences). The Radial Basis Function[rbf] or K-Nearest Neighbors algorithm[knn] serves as the hidden neural network layer. [mlp][ann]

The CogniMem was invented by Guy Pailet of General Vision. It is the successor of the ZISC (Zero Instruction Set Computing) chip invented also by Guy Paillet and jointly developed and patented with IBM[zisc]. IBM released the first ZISC chip with 36 neurons in 1993, The CogniMem CM-1K chip has 1024 neurons. For more details on the CogniMem see [cm1] [cm2].

3.2 CogniMem Features

3.2.1 Knowledge Representation

- Choice of two non-linear classifier:
  - Radial Basis Function (RBF)
  - K-Nearest Neighbor (KNN)

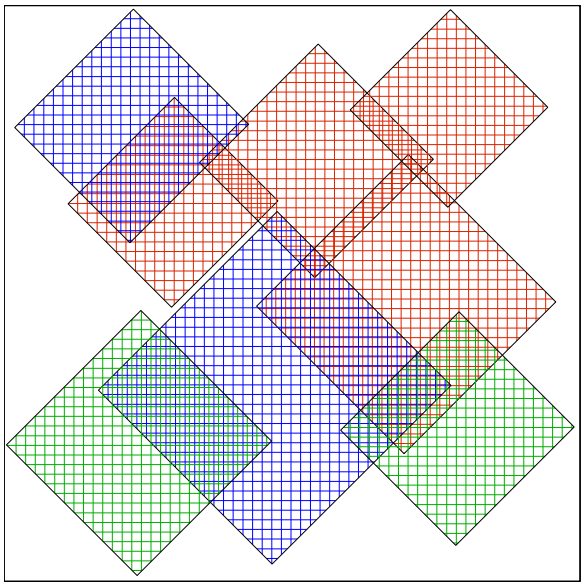


Figure 13: Radial Basis Function Space

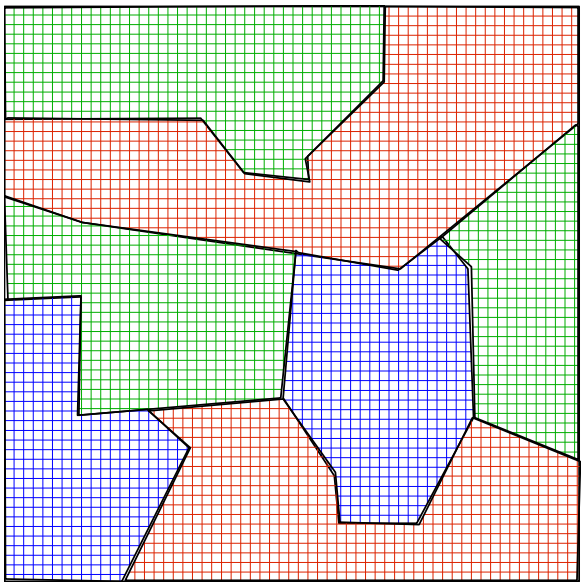


Figure 14: K-Nearest Neighbor Space

- Two different vector distance norms:

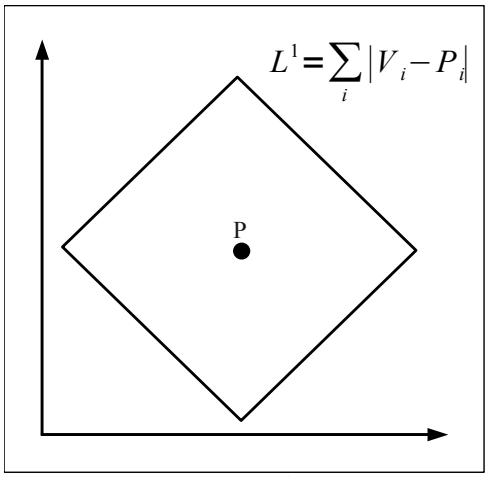


Figure 15: L^1 Norm

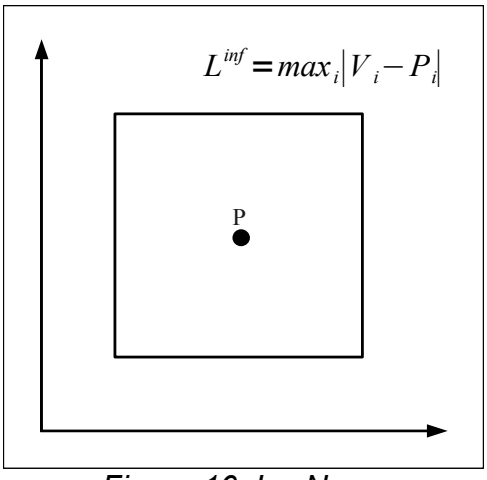


Figure 16: L^∞ Norm

- Automatic adaptive model generator.
  - Minimum Influence Field
  - Maximum Influence Field

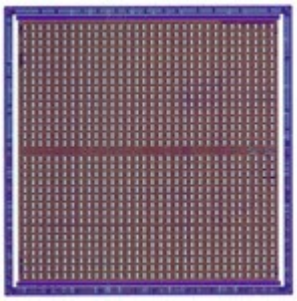
3.2.2 Training

- 256 byte input train pattern vectors.
- 32,767 categories.
- 128 contexts.
- Train by example.

3.2.3 Recognition

- 256 byte input match pattern vectors.
- Classification status can be identified, uncertain or unknown
- Response of all the firing neurons is accessible leading to uncertainty management and hypothesis generation

3.2.4 CM-1K Silicon



- Parallel network architecture.
- 1024 neurons (RoboSight is restricted to 992 neurons).
- Built-in video input recognition engine.
- Save and restore neural network functions.
- Learn in 18 clock cycles.
- Recognize in 36 clock cycles.
- 3.3 V I/O operation with 1.2 V core
- 500 mW @ 15MHz.

4 CAMERA

4.1 OmniVision OV6630 Overview

Figure 17 is an image of the C3038 Camera Board. The Camera Board interfaces to the RoboSight Main Board through front header block. The video signals are feed directly to the CogniMem. The control signals are connected to the main PIC controller. The camera is an OmniVision OV6630 CMOS image single-chip video/imaging camera sensors. See [ov6630] and [c3038] for more complete details.

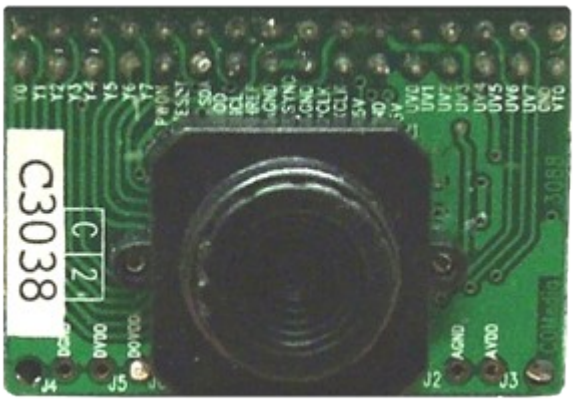


Figure 17: Camera Board - Front View

Imager	OV6630, CMOS image sensor
Array Size	352 x 288 pixels
Frame Rate	60 fps
Pixel size	9 X 8.2 µm
Scanning	Progressive
Effective image area	3.1mm x 2.5mm
Electronic Exposure	500:1
Gamma Correction	0.45/0.55/1.0
S/N Ratio	>48dB
Min Illumination	3lux @F1.2
Lens	f4.9mm, F2.8 FOV34.4x20.7^{o}
Video Analog Output	50Hz PAL monochrome composite at 75 ▪ .

Table 3: OV6630 Specification

4.2 Capabilities

- Automatic Gain Control (AGC).
- Automatic (Electronic) Exposure Control (AEC).
- Automatic White Balance control (AWB)
- Video adjustments: brightness, contrast, saturation, sharpness.
- Wide dynamic range, anti blooming, zero smearing.
- Low power dissipation
- Manual focus.

4.3 Operation Environment Tuning

The key to any vision processing application, including the RoboSight, is optimizing the front-end image quality, minimizing environmental variance, and reducing background clutter.

RoboSight provides a defined interface to control the camera to optimize the video stream for the target viewing conditions. Optimization can improve the neural network performance by:

- reducing the number of committed neurons when training.
- reducing both false positive or false negative categorization when running.

Optimal video images should have:

- sharp focus
- sufficient brightness
- very good contrast

Variable lighting conditions adversely effect performance. The light source should be sufficiently bright, fixed in respect to the camera, and provide a constant photon stream. Florescent light flickering adds variability with the camera image capture brightness and contrast. Incandescent, halogen, and LED lighting are the best light sources.

When possible, keep the visual background and foreground information to a minimum. Widely varying background colors and patterns and foreground object clutter or occlusion reduces object identification. The RoboSight is a pattern recognition device, identifying the pattern from the total Region of Interest. There are no built-in higher artificial intelligence functions that “know” the 3D shape of the target objects nor their spatial relationships.

5 Non-Volatile Memory

5.1 Hardware IDs

RoboSight hardware identification and version information are programmed into non-volatile memory of each manufactured RoboSight. RoadNarrows Intelligent Systems reserved hardware identification strings are:

Hardware ID	Description
STD1	Standard RoboSight hardware.
BAS1	Basic RoboSight hardware.

Table 4: Hardware IDs

5.2 Application Ids

RoboSight main firmware application identification and version information are programmed into non-volatile memory by the loaded firmware application. The specific firmware application is preloaded into each manufactured RoboSight. The specific application is determined by the type of RoboSight hardware and by customer requests.

Customers writing their own specific application are encouraged to use the RoadNarrows Intelligent Systems application identification and version scheme to aid identification and version management.

Application IDs are 4 characters long. RoadNarrows Intelligent Systems reserves all strings beginning with “RS” or “rs”. Current reserved full strings are:

Application ID	Description	Status
RSST	Standard RoboSight firmware.	Available
RSBA	Basic RoboSight firmware. (Superseded by RSST)	xxxx
RSSE	RoboSight firmware ideal of mobile robots. The video Region of Interest (ROI) is panned and zoomed searching for the specified trained categories. The ROI location is reported back to the robot controller.	Under Development
RSM3	RoboSight firmware that interfaces with a MP3 player board. A recognized category triggers the playing of a pre-programed sound file.	Future
RSTT	RoboSight firmware that interfaces with a Text-To-Speech board. A recognized category triggers the playing of a pre-programed string to the text-to-speech engine.	Future

Table 5: Application IDs

5.3 K-Sets

RoboSight working hardware parameters are divided into parameter sets (k-sets). All k-sets can be saved to and restored from RoboSight non-volatile memory. Table 4 list the supported k-sets. See RoboSight Registers Appendix for detailed information on the register k-sets.

K-Set	Mnemonic
CogniMem Registers	`cm`
Camera Registers	`cam`
RoboSight Operational Registers	`op`
Neural Network Trained Data	`nn`
All K-Sets	`all`

Table 6: K-Sets

5.4 Operations

RoboSight execution state is controlled by the working hardware k-set parameters. Figure 18 illustrates the relationship between the working parameters, the stored parameters in non-volatile memory, and the serial command operations.

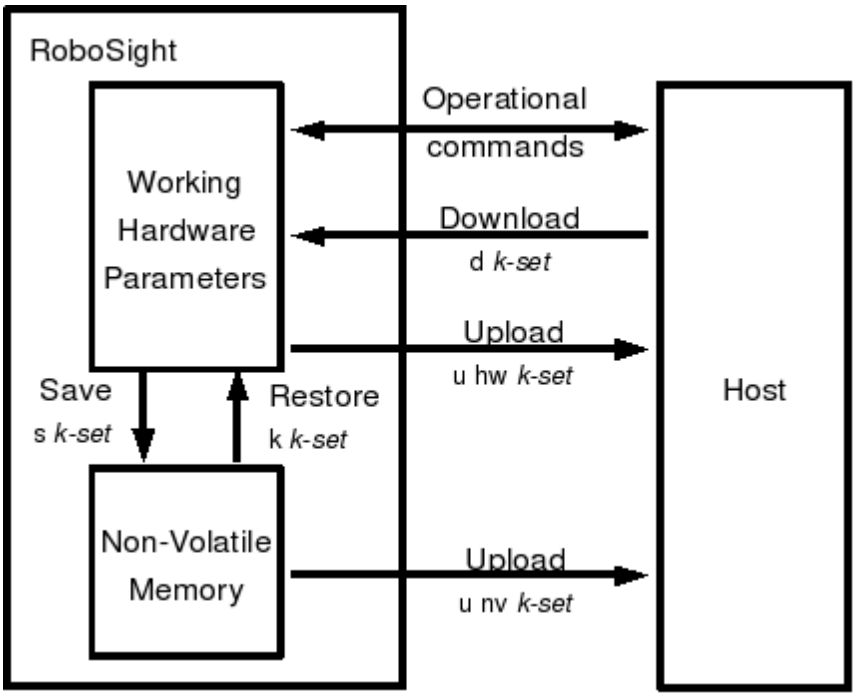


Figure 18: RoboSight Memory Model

5.4.1 Bulk Binary Transfers

The bulk k-set binary transfers are invoked by the Download Command and the Upload Command. See the Bulk Binary Transfer Protocol in the Serial Interface Reference Appendix for complete details on the binary transfer.

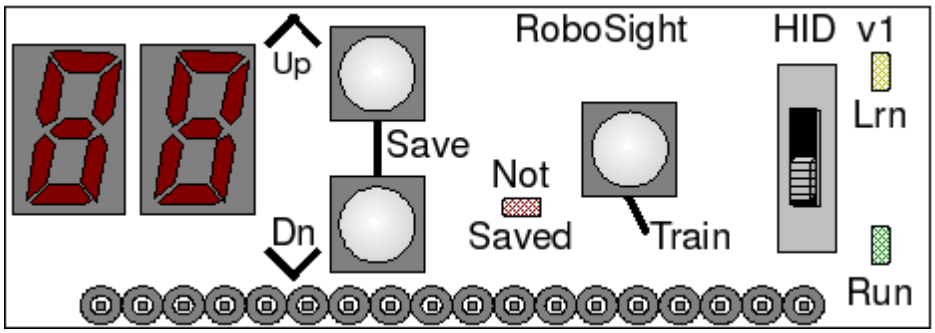
5.4.2 Save and Restore

The Save Parameters Command and the Restore Parameters Command operate between the working hardware k-sets and their respective non-volatile counterparts.

6 Human Interface Device

6.1 HID Overview

The RoboSight HID consists of one slider switch, three pushbuttons (Up, Dn, Train), three LEDs (Run, Lrn, Not Saved), and two 7-segment displays. The slider switch controls the mode of operation for RoboSight, while the buttons perform different functions based on the mode.

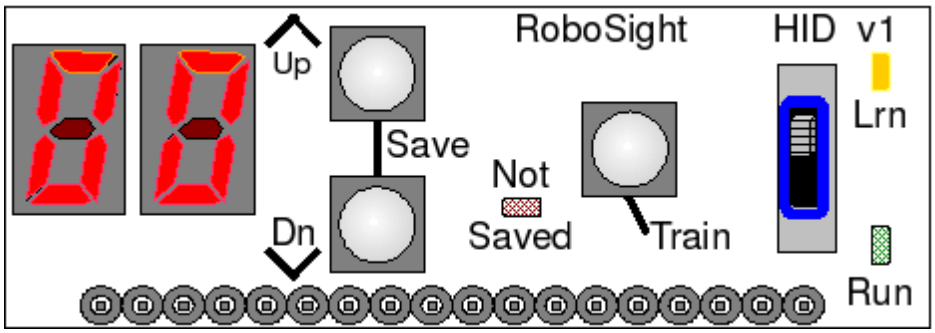


6.2 Operational Modes

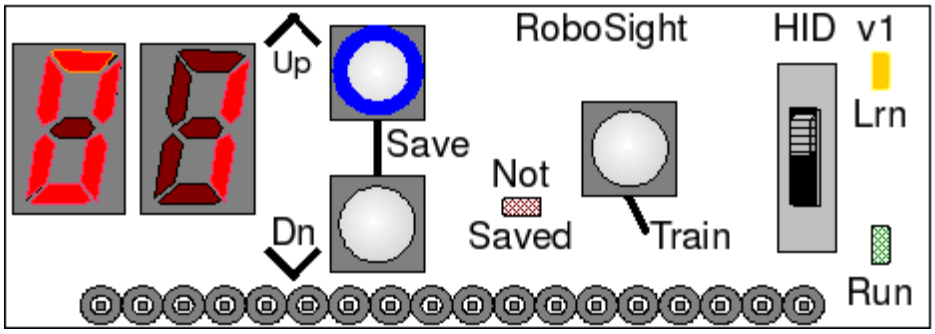
RoboSight has two operational modes; Learn and Run. In Learn mode, RoboSight is ready to train neurons. In Run mode, RoboSight is in actively recognizing patterns.

6.2.1 Learn Mode

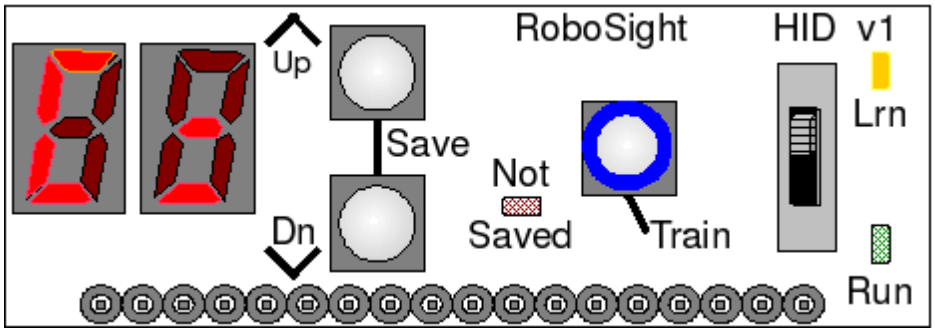
When the slider switch is set to Lrn (learn), the Lrn LED is illuminated and RoboSight is ready to train neurons. The 7-segment display shows the category to be trained.



Pressing the Up button increments the category, while pressing the Dn (down) button decrements the category to be trained.

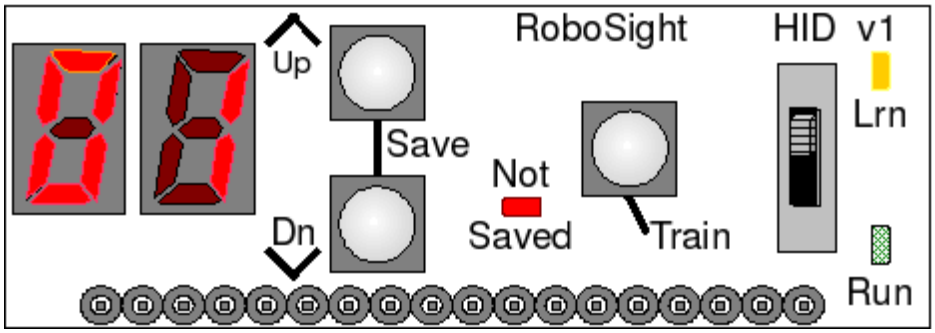


To train a new neuron, position RoboSight on the image to be trained and press the Train button. The 7-segment display will show “C=” followed by the neuron count.



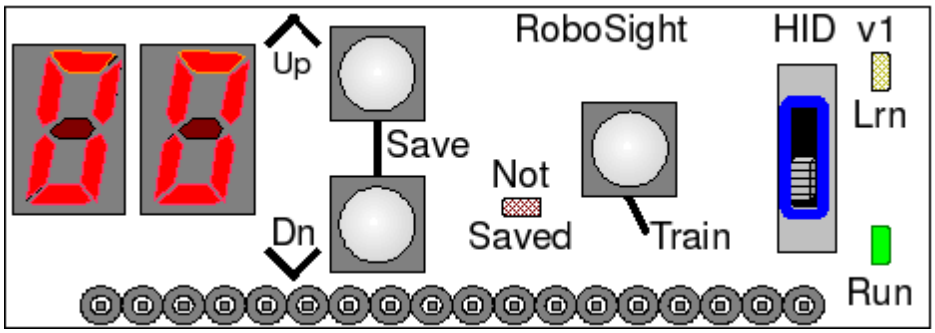
Afterward, the display reverts to showing the next category to be trained.

The Not Saved LED will illuminate to indicate trained CogniMen neurons have not been saved to non-volatile memory. See “Run Mode” for the procedure on saving neurons to non-volatile memory.

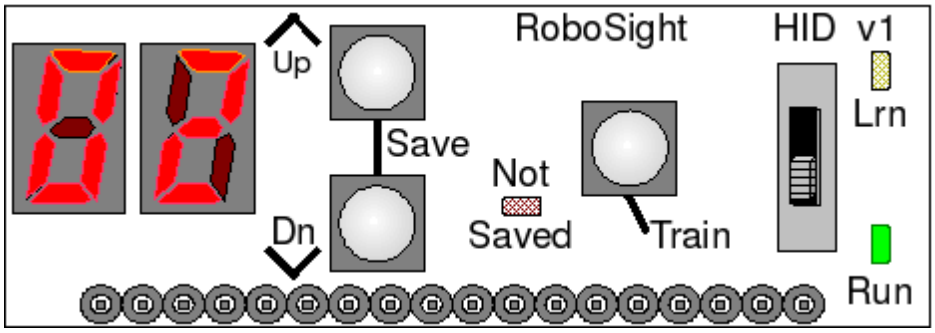


6.2.2 Run Mode

When the slider switch is set to Run, the Run LED is illuminated and RoboSight is in active pattern recognition mode. The 7-segment display shows the matching category.

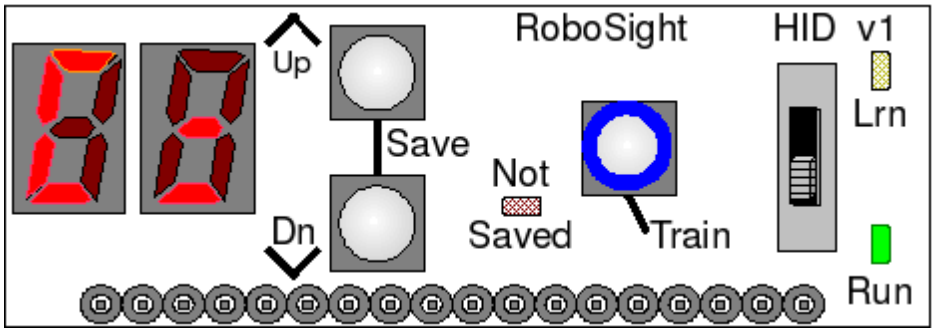


For instance, the following image shows RoboSight detecting category 2.



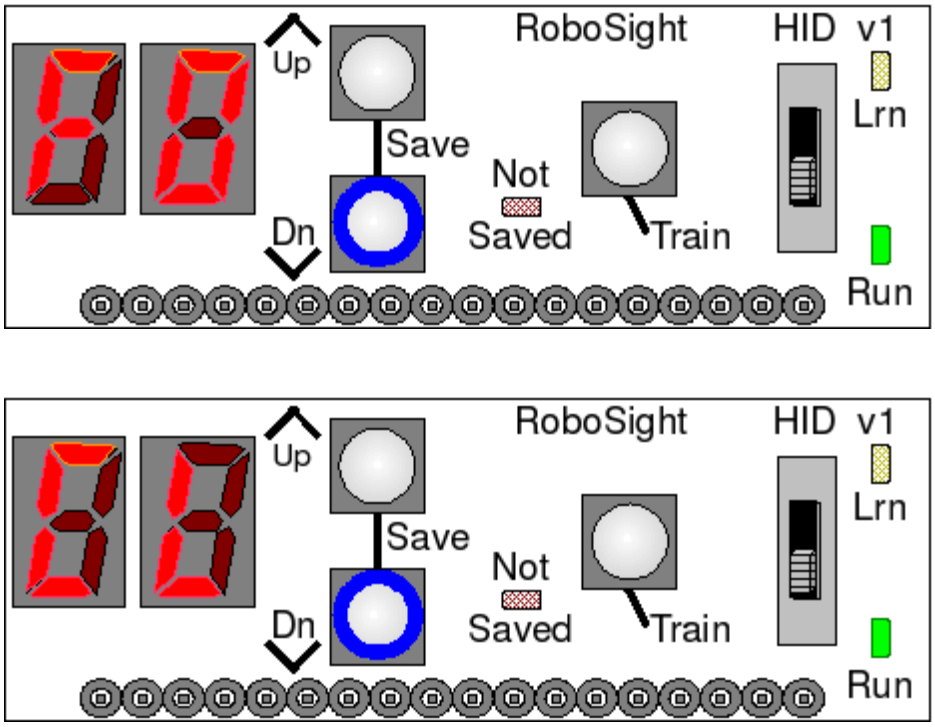
6.2.2.1 Display Count of Trained Neurons

To show the count of trained neurons in RoboSight, press and release the Train button. The 7-segment display will show “C=” followed by the neuron count. Afterward, the display reverts to showing the current matching category.



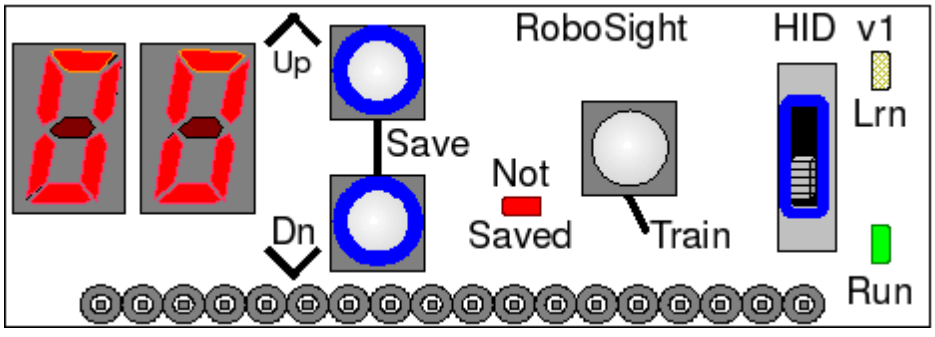
6.2.2.2 Clearing Neurons in CogniMem

To clear all neuron memory in RoboSight, press and hold the Dn (down) button for 2 seconds. The 7-segment display will show “FO” (forget) during the two seconds, followed by “CL” (clear) to indicate the neurons have been cleared. This function only clears the neurons in the CogniMen neural network chip and does not clear the non-volatile memory.

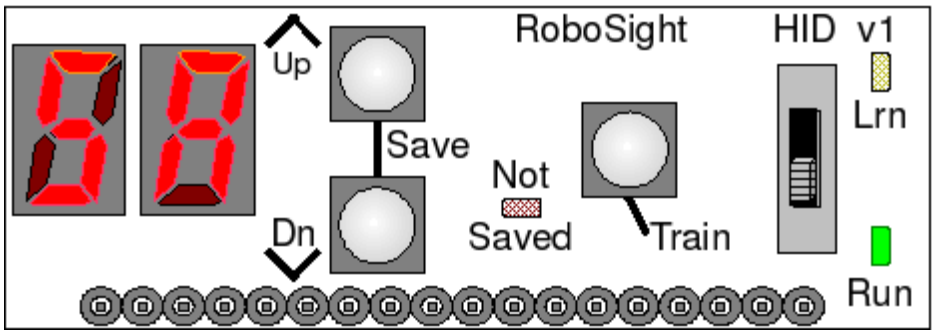


6.2.2.3 Saving Neurons to non-volatile Memory

To save the current trained neurons into non-volatile memory, press and hold the Up button, then press and release the Dn (down) button.



The 7-segment display will show “SA” while the neurons are being saved. Afterward, the display reverts to showing the current matching category.



List of Appendices

APPENDIX A: ROBOSIGHT REGISTERS

A.1 RoboSight Registers Overview

RoboSight registers are accessed via the Read Register ('r') and Write Register ('w') serial commands. There are three register banks, each addressed by an offset from the base address.

Bank	Base Address
CogniMem	0x0000
Camera	0x0100
Operational	0x0200

CogniMem registers allow direct access to the chip's registers while Camera registers allow direct access to the OV6630 camera registers. Operational registers control configuration parameters in RoboSight.

A.2 CogniMem Register Set

CogniMem registers allow direct access to the CogniMem neural network chip.

SR = Save/Restore mode LR = Learn/Recognize mode

Register	Address	Access	Description
NCR	0x000	SR – R/W	Neuron Context Register NCR[15:8] = Neuron Identifier bits [23:16] NCR[7] = Norm NCR[6:0] = Neuron Context
COMP	0x001	SR – R/W LR – W	Neuron Component. Writes to the neuron memory at the current index, updates the distance register and increments the index. COMP[7:0] = Component
LCOMP	0x002	LR – W	Last Neuron Component. Writes to the neuron memory at the current index, updates the distance register and launches the neuron associative logic. LCOMP[7:0] = Last Component
ICOMP	0x003	SR – W LR – W	Neuron Component Index. Sets the neuron memory index. ICOMP[7:0] = Component Index
DIST	0x003	SR – R LR – R	Distance DIST[15:0] = Distance
CAT	0x004	SR – R/W	Category

		LR – R/W	CAT[15] = 1 - Neuron degenerated CAT[14:0] = Category
AIF	0x005	SR – R/W	Active Influence Field AIF[15:0] = Active Influence Field
MINIF	0x006	SR – R LR – R/W	Minimum Influence Field MINIF[15:0] = Minimum Influence Field
MAXIF	0x007	LR – R/W	Maximum Influence Field MAXIF[15:0] = Maximum Influence Field
NID	0x00A	SR – R LR – R	Neuron Identifier NID[15:0] = Neuron Identifier
GCR	0x00B	LR – R/W	Global Context Register GCR[7] = Norm GCR[6:0] = Global Context
RSTCHAIN	0x00C	SR – W	Reset Neuron Chain during Save/Restore. Points to the first neuron of the chain.
NSR	0x00D	LR – R/W	Network Status Register NSR[5] = Model, 0 – RBF, 1 – KNN NSR[4] = Mode, 0 – LR, 1 – SR NSR[3] = 1 – Identified NSR[2] = 1 – Uncertain
FORGET	0x00F	LR – W	Forget. Clear the neurons' category, but not their entire contents.
NCOUNT	0x00F	LR – R	Count of committed neurons NCOUNT[15:0] = Count
LEFT	0x011	LR – R/W	Left corner of the Region of Interest in pixels LEFT[15:0] = Left corner
TOP	0x012	LR – R/W	Top corner of the Region of Interest in pixels TOP[15:0] = Top corner
WIDTH	0x013	LR – R/W	Width of the Region of Interest in pixels WIDTH[15:0] = Width
HEIGHT	0x014	LR – R/W	Height of the Region of Interest in pixels HEIGHT[15:0] = Height
BWIDTH	0x015	LR – R/W	Width of a primitive block in pixels BWIDTH[15:0] = Block Width
BHEIGHT	0x016	LR – R/W	Height of a primitive block in pixels BHEIGHT[15:0] = Block Height
RSR	0x01C	LR – R/W	Recognition Status RSR[5] = 1 – Recognition active RSR[4] = Copy of Frame Valid signal RSR[3] = Last recognition was an uncertain classification

			RSR[2] = Last recognition was an identified classification RSR[1] = 1 – Enable category to the Data Bus RSR[0] = 1 – Enable Real-time Recognition Engine
FEATDIST	0x01D	LR – R	Feature Distance FEATDIST[15:0] = Smallest neuron distance value
FEATCAT	0x01E	LR – R	Feature Category FEATCAT[15:0] = First category value
ROIINIT	0x01F	LR – W	Reset Region of Interest to default CogniMem values

A.3 OmniVision OV6630 Register Set

Camera registers allow direct access to the OV6630 camera registers.

Register	Address	Access	Description
GAIN	0x100	R/W	AGC Gain Control GAIN[5:0] – The current gain setting.
BLUE	0x101	R/W	Blue Gain Control BLUE[7] – Blue gain direction. “1” - increase, “0” - decrease. BLUE[6:0] = Blue gain.
RED	0x102	R/W	Red Gain Control RED[7] – Red gain direction. “1” - increase, “0” - decrease. RED[6:0] = Red gain.
SAT	0x103	R/W	Color Saturation Control SAT[7:3] – Saturation adjustment. “F8h” is highest and “00h” is lowest.
CTR	0x105	R/W	Contrast Control CTR[7] – selects gain at the dark area. “0” – gain=0.5 and “1” – gain=1. CTR[3:0] – Contrast adjustment. “0xF” is highest and “0x0” is lowest.
BRT	0x106	R/W	Brightness Control BRT[7:0] – Brightness adjustment. “0xFF” is highest and “0x00h” is lowest.
SHP	0x107	R/W	Sharpness Control SHP[7:4] – Coring adjustment. Range: 0~80mV with step 5mV. SHP[3:0] – Strength adjustment. Range: 0~8× with step 0.5×.
COMR	0x10E	R/W	Common control R COMR[7] – Analog signal 2x gain control bit. “1” - Additional 2x gain, “0” - normal.
COMS	0x10F	R/W	Common control S COMS[5:4] – select the lowest signal used in automatic black level expanding (or automatic brightness). “00” – lowest, “11” – highest and image is brighter. COMS[3:2] – select highest luminance level to be available in AWB control. “00” – lowest, “11” – highest. COMS[1:0] – select lowest luminance level to be

			available in AWB control. “00” – lowest, “11” – highest.
AEC	0x110	R/W	Automatic exposure control AEC[7:0] - Set exposure time TEX = 2 × TLINE × AEC[7:0]
COMA	0x112	R/W	Common control A COMA[7] – SRST, “1” initiates soft reset. All registers are set to default values and chip is reset to known state and resumes normal operation. This bit is automatically cleared after reset. COMA[6] – MIRR, “1” selects mirror image COMA[5] – AGCEN, “1” enables AGC, COMA[4] – Digital output format, “1” selects 8-bit: Y U Y V Y U Y V COMA[3] – Select video data output: “1” - select RGB, “0” - select YCrCb COMA[2] – Auto white balance “1” - Enable AWB, “0” - Disable AWB COMA[1] – Color bar test pattern: “1” - Enable color bar test pattern
COMB	0x113	R/W	Common control B. COMB[0] – Auto-Adjust Mode. “0” - disable, “1” - enable.

A.4 RoboSight Operational Resister Set

Operational registers display status and set configuration parameters in RoboSight.

Register	Address	Access	Description
FWVER	0x200	R	Firmware Version Holds the two-byte decimal major and minor version numbers of the firmware FWVER[15:8] = Major version (decimal) FWVER[7:0] = Minor version (decimal)
APPID	0x201 0x202	R	Application ID Holds four ASCII characters identifying the application ID that has been loaded onto the hardware. The high register holds the left two characters. The low register holds the right two characters.
CTL	0x203	R/W	RoboSight Control Provides host control and status of the RoboSight operation. CTL[15] = Setting this bit resets RoboSight CTL[14] = Setting this bit causes RoboSight hardware to enter sleep mode and reduce power consumption. CTL[13] = CogniMem Video Recognition Ownership Control, 0 - RoboSight is the owner, 1 - Host is the owner CTL[12] = HID LED Ownership Control, 0 - RoboSight is the owner, 1 - Host is the owner CTL[11] = Lock (disable) HID Push Buttons, 0 - HID buttons are unlocked.1 - HID buttons are locked. CTL[7] = Learn/Run Switch Position Status, 0 - Switch is in run mode position,1 - Switch is in learn mode position
I2CA	0x204	R/W	External I2C Address I2CA[7:1] – External I2C Address. Valid range is 0x70 - 0x7E
LEDB	0x205	W	LED Binary Sets individual LED segments on the 7-segment displays LEDB[14] = Left LED segment A LEDB[13] = Left LED segment B LEDB[12] = Left LED segment C LEDB[11] = Left LED segment D LEDB[10] = Left LED segment E LEDB[9] = Left LED segment F LEDB[8] = Left LED segment G LEDB[6] = Right LED segment A

<table>
<tr><td></td><td></td><td></td><td>LEDB[5] = Right LED segment B
LEDB[4] = Right LED segment C
LEDB[3] = Right LED segment D
LEDB[2] = Right LED segment E
LEDB[1] = Right LED segment F
LEDB[0] = Right LED segment G</td></tr>
<tr><td>LEDA</td><td>0x206</td><td>W</td><td>LED ASCII
Provides a limited mapping of ASCII characters applied to the LED display.
LEDA[15:8] = Set left LED ASCII value
LEDA[7:0] = Set right LED ASCII value</td></tr>
<tr><td>LEDD</td><td>0x207</td><td>W</td><td>LED Decimal
Provides 0 – 99 decimal LED display.
LEDD[15:8] = Set left LED number (0-99)
LEDD[7:0] = Set right LED number (0-99)</td></tr>
<tr><td>CIT</td><td>0x208</td><td>R/W</td><td>Category Integration Time
RoboSight integrates the CogniMem output category over the integration time to help reduce environment jitter.
CIT[15:0] = Integration time in milliseconds (2000 max)</td></tr>
<tr><td>NTN</td><td>0x209</td><td>R</td><td>Number of Trained Neurons
Returns the current number of trained neurons stored in RoboSight non-volatile memory.
NTN[15:0] = Number of trained neurons</td></tr>
<tr><td>LABEL</td><td>0x20A
0x20B
0x20C
0x20D</td><td>R/W</td><td>User Label
Four registers (8 bytes) are allocated for user application use. RoboSight does not interpret any of the contents but simply provides basic read/write and save/restore operations. A possible use of this register set is to store a 4 character label and a 4 byte Unix time stamp to identify the learned neural network and configuration data sets loaded on RoboSight.</td></tr>
<tr><td>FWREV</td><td>0x20E</td><td>R</td><td>Firmware Revision Number
Along with the FWVER register, this identifies the complete ID of the current firmware in a three dot format.
<Major Version>.<Minor Version>.<Revision>
FWREV[7:0] = Firmware revision number (decimal)</td></tr>
<tr><td>BAUDRATE</td><td>0x20F</td><td>R/W</td><td>Serial Baud Rate
Get or set the baud rate of the RoboSight serial port. The register enumeration – baud rate mapping is:
<table><tr><td>Enum Value</td><td>Baud Rate</td></tr><tr><td>1</td><td>1200</td></tr><tr><td>2</td><td>4800</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table>
<tr><td>3</td><td>9600</td></tr>
<tr><td>4</td><td>19200</td></tr>
<tr><td>5</td><td>38400</td></tr>
<tr><td>6</td><td>57600</td></tr>
<tr><td>7</td><td>115200
(default)</td></tr>
</table></td></tr>
<tr><td>HWVER</td><td>0x210</td><td>R</td><td>Hardware Version Register
The hardware version register holds the two-byte decimal major and minor version numbers of the hardware.
HWVER[15:8] = Major version (decimal)
HWVER[7:0] = Minor version (decimal)</td></tr>
<tr><td>HWID</td><td>0x211
0x212</td><td>R</td><td>Hardware ID Registers
RoboSight hardware id registers hold four ASCII characters identifying the RoboSight hardware. The high register holds the left two characters. The low register holds the right two characters. Example: “STD1”</td></tr>
<tr><td>HWSN</td><td>0x213
0x214</td><td>R</td><td>Hardware Serial Number Registers
RoboSight serial number registers hold a 32-bit decimal serial number identifying the specific RoboSight hardware.</td></tr>
</table>

APPENDIX B: HID 7-SEGMENT LEDS

B.1 LED Related Registers

The RoboSight HID contains two 7-segment LEDs to provide user feed back and status information. The RoboSight main firmware typically controls these registers. However, control may be relinquished by the RoboSight and control given to the requesting host application. This feature is useful for host applications that require special HID display needs.

There are four operational registers that provide an interface to the two LEDs:

Register	Address	Description
RoboSight Control and Status	0x203	LED ownership
LED Binary	0x205	Sets individual LED segments for complete control of the display.
LED ASCII	0x206	Provides a limited mapping of ASCII characters applied to the LED display.
LED Decimal	0x027	Provides 0 – 99 decimal LED display.

Table 7: LED Control and Display Registers

B.2 LED 7-Segment Mapping

Figure 19 provides a callout of the individual LED segments and how it maps to the LED Binary operational register.

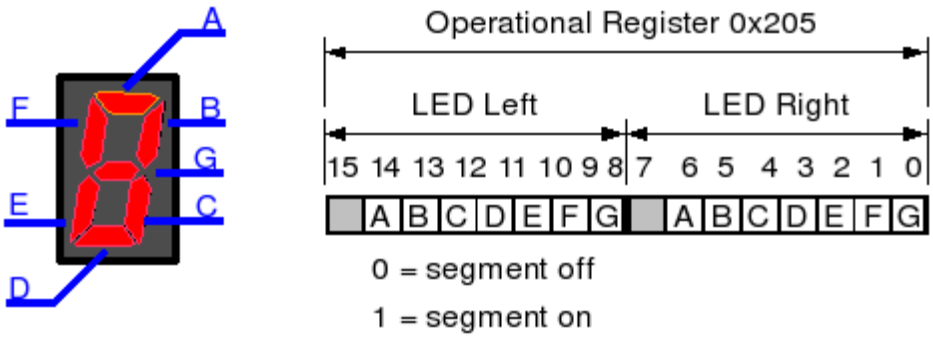


Figure 19: 7-Segment LED Mapping

B.3 LED ASCII Table

Table 7 lists the supported ASCII characters to LED display mapping. See Operational registers 0x206.

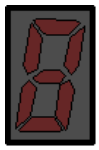
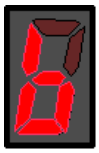
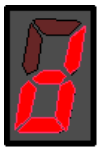
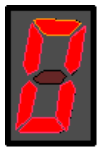
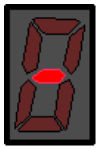
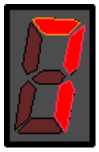
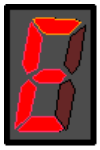
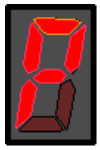
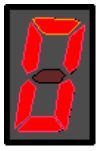
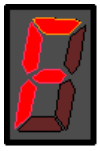
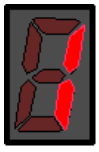
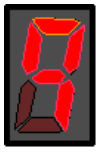
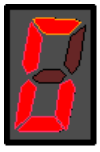
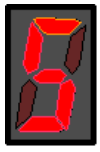
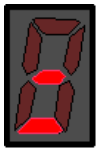
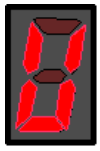
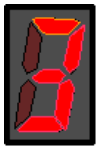
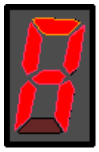
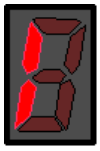
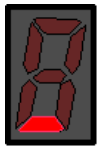
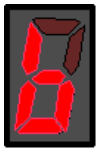
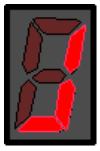
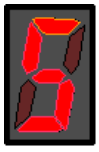
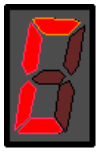
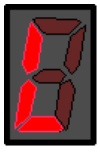
Char	Hex	LED	Char	Hex	LED	Char	Hex	LED	Char	Hex	LED
space	20		6	36		D	44		O	4F	
-	2D		7	37		E	45		P	50	
0	30		8	38		F	46		R	52	
1	31		9	39		G	47		S	53	
2	32		=	3D		H	48		U	55	
3	33		A	41		I	49		_	5F	
4	34		B	42		J	4A				
5	35		C	43		L	4C				

Table 8: ASCII Character to LED Mapping

APPENDIX C: SERIAL INTERFACE REFERENCE

Intro

The RoboSight serial interface, except for the binary transfer and bootloader protocols, is an ASCII, case-sensitive interface. The host sends commands and the RoboSight replies with the appropriate responses. The RoboSight does not send any asynchronous notifications.

Serial Device Settings:

Baud Rate:	Value of register BAUDRATE (0x20f). Factory default: 115200
Data Size:	8
Parity:	N
Stop Bits:	1
Hardware Flow Control:	off
Software Flow Control:	off
Remote Echo:	on
Local Echo:	off
Summary String:	Current: BAUDRATE,8,N,1 Factory default: 115200,8,N,1

RS-232 Pin Out:

The RoboSight serial port supports the minimum 3-pin configuration. Reading bottom to top, the pin out is as follows:

Pin	Mnemonic	Description
1	GND	Ground
2	RX	Receive
3	TX	Transmit

Common Command Syntax:

All host commands have the following basic syntax structure:

```
cmd_id [arg1 ...]
```

The *cmd_id* is a one ASCII character command identifier. All command line arguments are white-space separated. A command terminates with a carriage-return `CR` (0x0D). Numeric

arguments may be entered in either hexadecimal or decimal format. For hexadecimal numbers, the string "0x" must prepend the hexadecimal sequence.

Common Response Syntax:

RoboSight generates a response for each command. Response formats are:

Case	Syntax	Comments
Success with returned data:	*arg1 [arg2 ...]*	Arguments are command specific.
Success with no returned data:	ok	Success.
Parse or execution error:	? *ecode*	See RoboSight Error Codes

Responses are terminated with the character sequence CR+LF (0x0D 0x0A). The RoboSight prompt string "->" is sent by RoboSight after the end of every response, except for the binary protocols initiation. In that case the prompt is delayed until the binary protocol is terminated. A response may span multiple lines.

Notes:

The HID state is ignored by the RoboSight when processing a received serial command. For example, retrieving the current recognized category is valid even though the HID slider is positioned at the Learn Mode.

Ownership of particular RoboSight assets is controlled through the Operational Control and Status Register. By default the RoboSight owns these assets. RoboSight will relinquish ownership at the request of the host.

Parameter K-Sets

RoboSight working parameters are divided into parameter sets (k-sets). All k-sets can be saved to and restored from RoboSight non-volatile memory.

K-Set	Mnemonic
CogniMem Registers	cm
Camera Registers	cam
RoboSight Operational Registers	op
Neural Network Trained Data	nn
All K-Sets	all

Table 9: Parameter K-Sets

RoboSight Error Codes

Table 9 lists the error codes returned to the user in the event of a parsing or execution error.

Error Code	Description
1	General, unspecified error
2	Invalid command syntax
3	Resources busy
4	Input buffer too big
5	Transfer aborted
6	Invalid argument syntax
7	Argument out of range
8	Unknown or invalid command id
9	Bad checksum
10	Wrong number of arguments
11	Execution error
12	Operation timed out
13	Operation not permitted

Table 10: RoboSight Error Codes

Command Summary

ID	Operation	Command	Response
c	Category and Distance Command	`c [n]`	*`cat dist[,...]`*
d	Download Command	`d` *`k_set`*	[binary transfer]
f	Forget Command	`f`	*`Forgetting. Press y to confirm`*
h	Help Command	`h`	List of commands
i	Info Command	`i`	Neural network information
k	Restore Parameters Command	`k` *`k_set`*	`ok`
n	Show Neurons Command	`n`	Neuron data.
r	Read Register Command	`r` *`reg_addr`*	*`reg_val`*
s	Save Parameters Command	`s` *`k_set`*	`ok`
t	Train Command	`t` *`cat`*	*`cat`*
u	Upload Command	`u` *`src k_set`*	[binary transfer]
w	Write Register Command	`w` *`reg_addr reg_val`*	`ok`
x	Software Reset Command	`x`	[reboot]

Table 11: RoboSight Command Summary

Category and Distance Command

Synopsis:

Command:	`c [n]`	
Response:	*cat dist*`[,...]`	
Arguments:	*n*	Number of category distance pairs to retrieve [1 – 4]. Default: 1.
	cat	Recognized category. 0 = unknown category recognized. 65535 = N/A
	dist	Distance from center of category [0 – 65535]. The larger the distance, the less confident is the categorization.

Description:

Get the current *n* best category and distance (confidence level) pairs.

Examples:

Category 5 is recognized with high confidence.

```
->c
5 4033

->c 4
5 4033, 8 12909, 5 16323, 65535 65535
->
```

See Also:

Train Command

Download Command

Synopsis:

Command:	`d k_set`	
Response:	[binary protocol]	Enter bulk binary transfer mode.
Arguments:	*k_set*	Parameter k-set. One of: `cm cam op nn.`

Description:

The download command initiates a bulk binary transfer of the specified k-set data.

The download invokes the bulk, binary transfer protocol to download the parameter k-set data from the host application to the RoboSight. Unlike the upload, the data can only be be downloaded to the working parameters of the hardware.

Notes:

Only the RoboSight working values are altered. Changes are not saved to non-volatile memory until the Save Command is issued.

Downloading a fully populated neural network can take 90 seconds to complete.

Examples:

Download a set of trained neural network data.

```
->d nn
[bulk binary transfer...]
->
```

See Also:

Bulk Binary Transfer Protocol, Save Parameters Command, Parameter K-Sets

Forget Command

Synopsis:

Command:	`f`	
Response:	*`Forgetting. Press y to confirm`*	
Confirm Command:	`y`	No <Enter> required.
Confirm Response:	`ok`	

Description:

Forget all trained data (delete) for the neural network. The command requires a confirmation.

Notes:

This command only effects the working parameters, not the non-volatile memory.

Examples:

Forget the current trained data. The pressed '`y`' is overwritten by the '`ok`' on the display.

```
->f
Forgetting. Press y to confirm
ok
->
```

See Also:

Train Command

Help Command

Synopsis:

Command: `h`

Response:
```
Single letter commands
c - show category and distance
d - download data from host
f - force forget
h - display help
i - display Cognimem information
k - restore data from EEPROM
n - show all neurons
r - read register
s - save data to EEPROM
t - train video
u - upload data to host
w - write register
x - reset RoboSight
```

Description:

Display list of available serial commands.

Info Command

Synopsis:

Command: `i`

Response:
```
Committed neurons=n
Reco status=status
Feature Dist=dist
Category=cat
Region of Interest:
Left=roi_left
Top=roi_top
Width=roi_width
Height=roi_height
Block width=roi_bwidth
Block height=roi_bheight
```

Arguments:		
	n	Number of trained neurons.
	status	Recognition engine status bits.
	dist	Recognized category distance.
	cat	Recognized category.
	roi_left	Region of Interest upper left corner left pixel position in video stream.
	roi_top	Region of Interest upper left corner top pixel position in video stream.
	roi_width	Region of Interest width in pixels.
	roi_height	Region of Interest height in pixels.
	roi_bwidth	Region of Interest primitive block width in pixels.
	roi_bheight	Region of Interest primitive block height in pixels.

Description:

Display a summary of CogniMem key information. This command is useful for the user while interacting with the RoboSight to quickly determine the state of the neural network.

Restore Parameters Command

Synopsis:

Command: `k` *`k_set`*

Response: `ok`

Arguments: *`k_set`* Parameter k-set. One of: `cm cam op nn all`.

Description:

Restore RoboSight working parameters from values stored in non-volatile memory.

Notes:

Restoring a fully populated neural network to the CogniMem can take 90 seconds to complete.

Examples:

Restore working hardware data from parameter k-sets stored in non-volatile memory.

```
->k all
ok
->
```

See Also:

Save Parameters Command

Show Neurons Command

Synopsis:

Command: `n`

Response:
```
NCR=ncr
Vector=
pat_data[0-15]
...
pat_data[200-255]
AIF=aif
NID=nid
Cat=cat
...
```

Arguments:		
	ncr	The neuron's Neuron Context Register bit values.
	pat_data	The 256 byte hex dump of neuron's pattern vector.
	aif	The neuron's Active Influence Field.
	nid	The neuron's Id.
	cat	The category assigned to the neuron.

Description:

Dumps the parameter data of the committed, trained neural network. This command is useful for the user while interacting with the RoboSight to quickly determine the loaded configuration of the neural network.

For each trained neuron, the response repeats the "`NCR ... Cat`" lines.

Notes:

Only the working neural network is read. The RoboSight non-volatile memory is not accessed.

Read Register Command

Synopsis:

Command:	`r reg_addr`	
Response:	*reg_val*	
Arguments:	*reg_addr*	Decimal or hexadecimal register address.
	reg_val	Decimal or hexadecimal register value.

Description:

Read the current value from a RoboSight register. There are three register banks, each addressed by an offset from the base address.

Bank	Mnemonic	Base Address
CogniMem	`cm`	0x0000
Camera	`cam`	0x0100
Operational	`op`	0x0200

Notes:

Only the working values are read. The RoboSight non-volatile memory is not accessed.

Examples:

Read the current category from the CogniMem FEATCAT register. Next, read the RoboSight Operational Control Register.

```
->r 0x1E
0x05
->r 0x203
0x0080
->
```

See Also:

Write Register Command, Save Parameters Command, Bulk Binary Transfer Protocol

Save Parameters Command

Synopsis:

Command: `s k_set`

Response: `ok`

Arguments: *k_set* Parameter k-set. One of: `cm cam op nn all.`

Description:

Save the RoboSight working parameters to non-volatile memory.

Notes:

Saving a fully populated neural network to the CogniMem can take 90 seconds to complete.

Examples:

Save all working camera parameters non-volatile memory.

```
->s cam
ok
->
```

See Also:

Restore Parameters Command

Train Command

Synopsis:

Command:	`t cat`	
Response:	*cat*	
Arguments:	*cat*	Assigned category.

Description:

Train the current video to the given category.

When the CogniMem recognition engine is enabled, the RoboSight camera continuously feeds the video stream to the CogniMem. The train command informs the CogniMem to associate the current image features to the specified category.

Notes:

To properly train on a video stream, keep the imaged object still and the video conditions constant.

Examples:

Train the current video image as category 2.

```
->t 2
2
->
```

See Also:

Category and Distance Command

Upload Command

Synopsis:

Command:	`u src k_set`	
Response:	[binary protocol]	Enter bulk binary transfer mode.
Arguments:	*src*	Parameter k-set source. One of: `hw nv`.
	k_set	Parameter k-set. One of: `cm cam op nn`.

Description:

Upload the parameter k-set data.

The upload invokes the bulk, binary transfer protocol to upload the parameter k-set data from the RoboSight to the host application. The data may be uploaded from either the working parameters of the hardware or from the parameters stored in non-volatile memory.

Notes:

Uploading a fully populated neural network can take 90 seconds to complete.

Examples:

Upload the RoboSight operational registers currently saved in non-volatile memory.

```
->u nv op
[bulk binary transfer...]
->
```

See Also:

Bulk Binary Transfer Protocol, Download Command, Parameter K-Sets

Write Register Command

Synopsis:

Command:	`w reg_addr reg_val`	
Response:	`ok`	
Arguments:	*reg_addr*	Decimal or hexadecimal register address.
	reg_val	Decimal or hexadecimal register value to write.

Description:

Write a value to a RoboSight register. There are three register banks, each addressed by an offset from the base address.

Bank	Mnemonic	Base Address
CogniMem	`cm`	0x0000
Camera	`cam`	0x0100
Operational	`op`	0x0200

Notes:

Only the working values are modified. The RoboSight non-volatile memory is not accessed.

Examples:

Write “A9” to the HID LEDs.

```
->w 0x206 0x4139
ok
->
```

See Also:

Read Register Command, Save Parameters Command, RoboSight Registers

Software Reset Command

Synopsis:

Command: `x`

Response: [reboot]

Description:

Reset the RoboSight.

Force the RoboSight into a software reset. The RoboSight will reboot, performing system checks and initializing hardware from parameters stored in the non-volatile memory.

Notes:

If there are significant number of neurons to load from non-volatile memory, then the boot-up sequence can take as long as 90 seconds.

Bulk Binary Transfer Protocol

Description:

RoboSight binary protocol transfers bulk parameter data between a host and the RoboSight. It is a simple ack/abort protocol. All binary data are big endian at both the byte and bit level.

The nomenclature is RoboSight centric.

- upload: RoboSight transfers parameter data to the requesting host.
- download: The host transfers parameter data to the RoboSight.

Binary Transfer Protocol:

With minor exceptions, the Binary Transfer Protocol is symmetric in regards to the upload or download direction. For uploading, the source is the RoboSight and the destination is the host. Similarly, for downloading, the source is the host and the destination is the RoboSight. Either the source or the destination side may abort the transfer.

The ping pong diagram shown below depicts the messaging flow between source and destination. The type of records sent is determined by the arguments to the download/upload command that initiated the binary transfer protocol and to the *ack* field value.

At the termination of the binary protocol the RoboSight reverts back to ASCII command mode.

Source		Destination	Comments
cnt	⇒		Send the number of records to be transferred.
ack record	⇒		First record, prefaced by an *ack* field
	⇐	*ack* [*abort_record*]	Respond with an ack/abort.
	…		
ack record	⇒		*Send the cnt*th (last) record, prefaced by an *ack* field
	⇐	*ack* [*abort_record*]	Respond with an ack/abort.
checksum	⇒		Transfer checksum.
	⇐	*ack* [*abort_record*]	Sent only by RoboSight on downloads.

[End of Binary Transfer Protocol. Revert back to ASCII command mode.]

Table 12: Binary Transfer Protocol Messaging Diagram

Binary Transfer Protocol Record Format:

Record	Format	Description
reg_record	*reg_addr reg_val*	Register K-Set (`cm`, `cam`, `op`)
nn_record	*ncr aif nid cat pat_0 … pat_255*	Trained Neural Network K-Set (`nn`)
abort_record	*ecode eol*	Error or Abort Transfer. Sent only on abort (*ack* = '`?`') and only if the source is the RoboSight.

Table 13: Binary Transfer Protocol Record Format

Binary Transfer Protocol Field Format:

Field	Type	Description
cnt	*16-bit unsigned binary decimal*	Number of records to transfer.
ack	*8-bit*	Ack/Abort field. One of ASCII: '`g`' '`?`' `g`: Acknowledge and continue. `?`: Abort
reg_addr	*16-bit unsigned binary decimal*	The register address.
reg_val	*16-bit unsigned binary decimal*	The register value.
ncr	*16-bit unsigned binary decimal*	Neuron context.
aif	*16-bit unsigned binary decimal*	Neuron's Active Influence Field.
nid	*16-bit unsigned binary decimal*	Neuron Id.
cat	*16-bit unsigned binary decimal*	Neuron category.
pat_k	8-bit binary	The k^{th} pattern byte for the neural network
ecode	*ASCII number*	See RoboSight Error Codes
eol	*ASCII sequence*	End-Of-Line sequence: host originated: `CR` (0x0D) RoboSight originated: `CR-LF` (0x0D 0x0A)
checksum	*16-bit unsigned binary decimal*	Additive 16-bit check sum over transmitted bytes from the first byte in *cnt* to the last byte in the cnt^{th} (last) record.

Table 14: Binary Transfer Protocol Field Format

APPENDIX D: I²C MESSAGE REFERENCE

Intro

The RoboSight I²C Serial Bus interface provides a read-only simple interface to retrieve the current recognized category. For simple robots or other applications with very little processing capabilities, the I²C is an alternative RoboSight interface to the serial interface. The user can train the neural network using the HID. In Run Mode, the current category is always readable on the I²C interface. For simple robots in particular, the read category can trigger simple behavior responses.

Example pseudo-code:

```
category = i2cread()   // read category
if category == 2       // trained on a apple
then
    go forward         // go to the apple
else if category == 3 // lemon
then
    turn 180           // avoid lemons
else
    make random move   // search
endif
```

I²C Serial Bus Settings:

Mode:	slave
Rate:	100 Kbits/s
Signaling Voltage:	5VDC
Address:	7-bit programable
Data:	8-bit

I²C Pin Out:

The I²C Serial Bus is a 2-wire interface. The RoboSight connector has additional, independent power and ground pins to power the RoboSight. The pin out reads right to left when facing the front of the RoboSight.

Pin	Mnemonic	Description
1	SDA	Serial Data
2	SCL	Serial Clock
3	PWR	+5 Volts
4	GND	Ground

Messaging:

The host (master) initiates a read command by placing the address on the bus. The RoboSight (slave) responds with a 1 byte decimal value specifying the current category.

APPENDIX E: FIRMWARE BOOTLOADER

Intro

The bootloader is a separate RobotSight application saved in protected non-volatile memory. The bootloader provides a serial interface to download updated RoboSight firmware or specific RNIS and user-written applications.

The host downloader application can either be a terminal emulation application (e.g. *picocom, Tera Term,* or *HyperTerminal*) or the command-line *rsFwDownloader* application provided by RoadNarrows Intelligent Systems.

The format of the download image file is Intel 8-bit HEX[hex].

The CCS compiler can generate Intel HEX files from PIC 18F4620 project build output. These generated files are compatible with the RoboSight bootloader.

The new firmware image is saved in RoboSight's non-volatile memory.

Starting The Bootloader:

To enter into the bootloader application, enter the ASCII 'b' (0x62) within the first 2 seconds of the RoboSight boot sequence. The RoboSight will then jump to the bootloader. Power cycling or entering the Software Reset Command will initiate a (re)boot. The bootloader displays the following output and prompts the user for bootloader commands:

```
RNIS bootloader v1.0 (C)2008
h for help
>
```

Note that the version and date strings may vary.

BootLoader Commands:

- `l` Enter the file download protocol.
- `j` Jump to the (new) main application.
- `h` Print list of commands.

Bootloader commands require no end-of-line sequences.

Download Firmware By rsFwDownloader

The **rsFwDownloader** is a command-line RoboSight application to download new RobotSight firmware over the serial connection to the RoboSight. The **rsFwDownloader** executes the host side of the download protocol used by the RoboSight's bootloader. The bootloader reads and decodes each Hex record into binary. The image is saved in non-volatile memeory. The downlaod protocol is a receiver generated ack'ed base protocol with software XON/XOFF flow control.

RsFwDownLoader Man Page

Name:

rsFwDownloader - RoboSight firmware downloader.

Synopsis:

`rsFwDownloader [OPTIONS]` *`hexfile`*

Description:

The **rsFwDownloader** downloads the specified RoboSight PIC18F4620 main processor image file. The image file must be in Intel Hex format. The download is through an RS-232 connection between the host platform and the RoboSight bootloader.

The **rsFwDownloader** automatically places the RoboSight into the RoboSight bootloader prior to downloading. After a successful download, the RoboSight will be commanded to jump to the new image.

The **rsFwDownloader** is compatible with any RoboSight standard+ main application, where standard+ is any (extended) application supporting the RoadNarrows Intelligent System RoboSight core command set.

Options:

Mandatory arguments to long options are also mandatory for short options.

Downloader

`-s, --silent`

Download silently. No status messages are printed to *stdout*.
DEFAULT: false (print status messages)

Communication

`-b` *rate*, `--baudrate=`*rate*

RoboSight serial baudrate. One of:
`1200`, `4800`, `9600`, `19200`, `38400`, `57600`, `115200`
DEFAULT: `115200`

`-d` *device*, `--device=`*device*

RS-232 serial device port attached to the RoboSight.
DEFAULT: `/dev/ttyS0`

Logging

`-l` *level*, `--log=`*level*

Set logging threshold level. All logging events with priority ≤ *level* will be logged. All others will be ignored. Error events are always logged. The *level* is one of:

'off' or 0	Disable all logging.
'error' or 1	Enable error logging.
'diag1' or 2	Enable diagnostics 1 logging.
'diag2' or 3	Enable diagnostics 2 logging.
'diag3' or 4	Enable diagnostics 3 logging. The libRoboSight library logs diagnostics at this level.
'diag4' or 5	Enable diagnostics 4 logging. The libserial library logs diagnostics at this level.
'diag5' or 6	Enable diagnostics 5 logging.
>6	Enable user-defined logging.

DEFAULT: off

`--logfile=`*file*

Set log output file to *file*. Special *file* names are:

'stderr' log to standard error.

'stdout' log to standard output.

DEFAULT: stderr

Information

`--help`

Display command help and exit.

`--version`

Output version information and exit.

Arguments:

Following any command-line options is the required path name to the PIC4620 main image file in Intel Hex format.

Example:

```
# Example
# Download the new version of the RoboSight standard main
# Hex file through the host port associated with serial
# device /dev/ttyUSB0.
```

```
$ rsFwDownloader --device=/dev/ttyUSB0 \
/public/RoboSight4620_std.hex
Hex file: /public/RoboSight4620_std.hex
   size: 54817
   date: Thu May 21 14:59:33 2009
Connecting to RoboSight on device /dev/ttyUSB0 @ 115200.
Connected.
Determining RoboSight application.
RoboSight determined to be executing standard main.
Current RoboSight firmware is RSST v0.101.0.
Resetting RoboSight to bootloader.
Re-connecting to RoboSight bootloader.
Re-connected to RoboSight bootloader.
Communication established with bootloader.
Downloading.
..................................................
..................
..................................................
..................
<more lines>
.........................
Hex file downloaded successfully.
  records count: 1225.
Jumping to RoboSight main.
Re-connecting to RoboSight main.
Re-connected to RoboSight main.
New RoboSight firmware is RSST v1.2.1.
Firmware updated successfully
$
```

Download Firmware By Terminal Program

The following instructions provide an example on how to download RoboSight firmware HEX file using the RoboSight bootloader and TeraTerm[ttssh] terminal emulation program executing on Windows XP.

Step 1: Connect

Connect the RoboSight serial cable to the Windows PC.

Step 2: Device Manager Port Settings

Configure the Windows serial port to buffer only 2 bytes on transmit and 1 byte on receive. The RoboSight bootloader protocol uses XON/XOFF for flow control. If the buffers are too big, the XOFF can sit in the input FIFO and not be seen by the host until too late and transmit data has been lost.

- **Note**: Serial over USB or Bluetooth have buffering that may not be possible to adjust or shrink sufficiently.

In the Windows **Control Panel**, bring up the **System** entry and select the **Hardware** tab. Push the **Device Manager** button. In the Device Manager find the appropriate device in the **Ports** list. In this example, it is **COM1**.

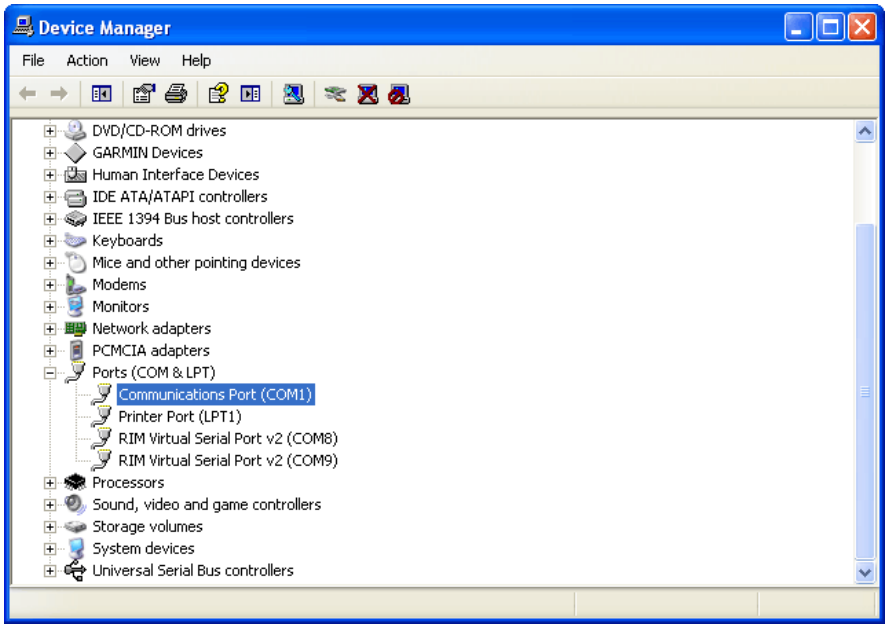


Figure 20: Windows Device Manager

Right click and select **Properties** item in the pop up menu. The Properties Dialog will be displayed.

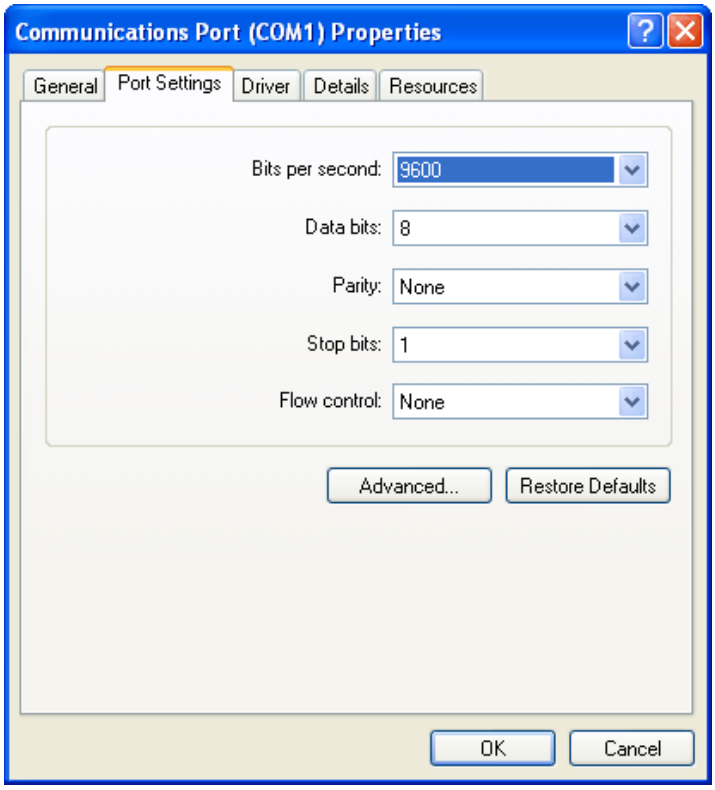


Figure 21: Windows Device Manager Port Properties

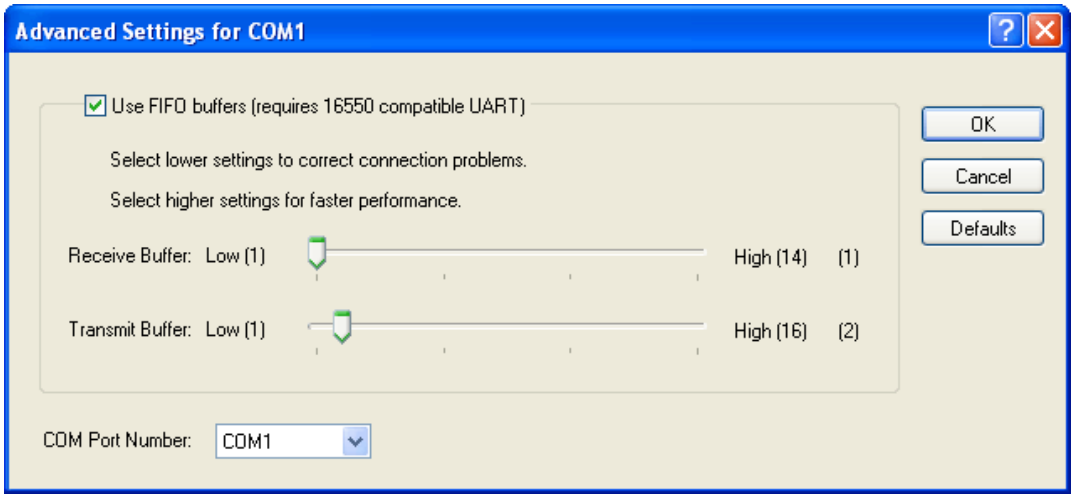


Figure 22: Windows Device Manager Advance Port Settings

Push the **Advanced...** button. In the **Advance Settings** dialog position the **Receive Buffer Low** slider to 1 and the **Transmit Buffer Low** slider to 2. Save the configuration.

Step 3: Tera Term Port Settings

Bring up Tera Term and select the **Port Settings...** item under the **Setup** menu.

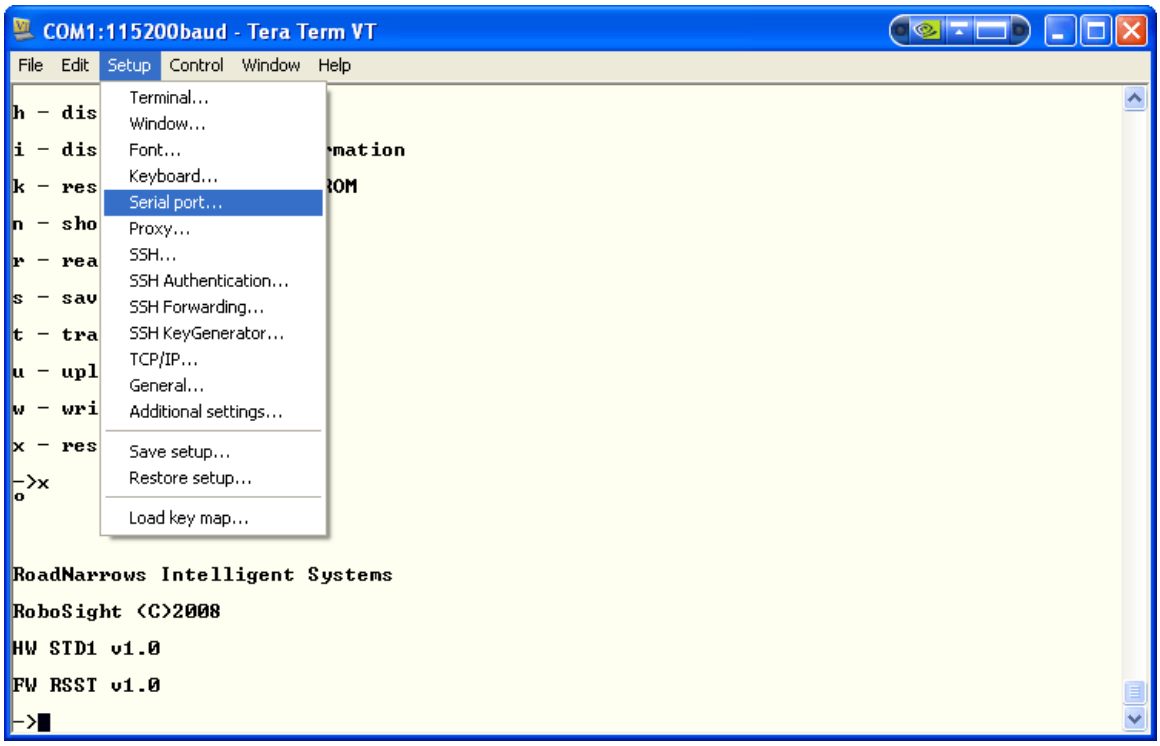


Figure 23: Tera Term - Setup Menu

Configure the port as shown. The flow control must be set to Xon/Xoff. The 15msec line delay between transmitted lines gives the RoboSight a chance to program each line of code into Flash. Save the settings.

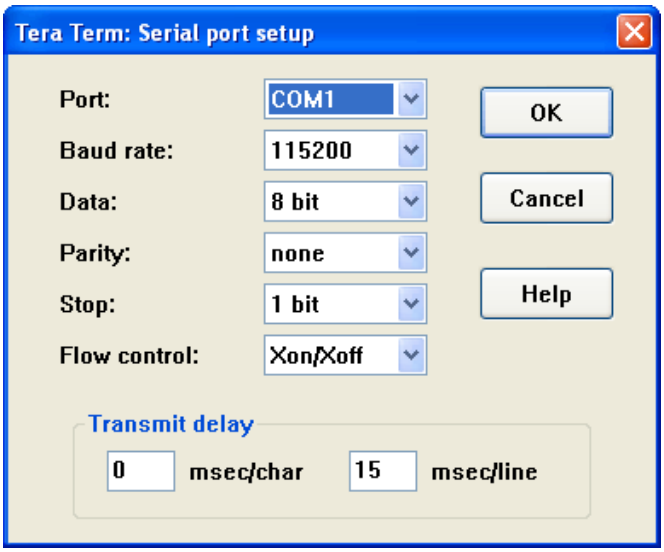


Figure 24: Tera Term - Port Settings

Step 4: Establish Communication

Power on the RoboSight. In the Tera Term window you should see the boot up strings terminated by the user prompt. If you pressed '`b`' during the boot up, go to **Step 6**.

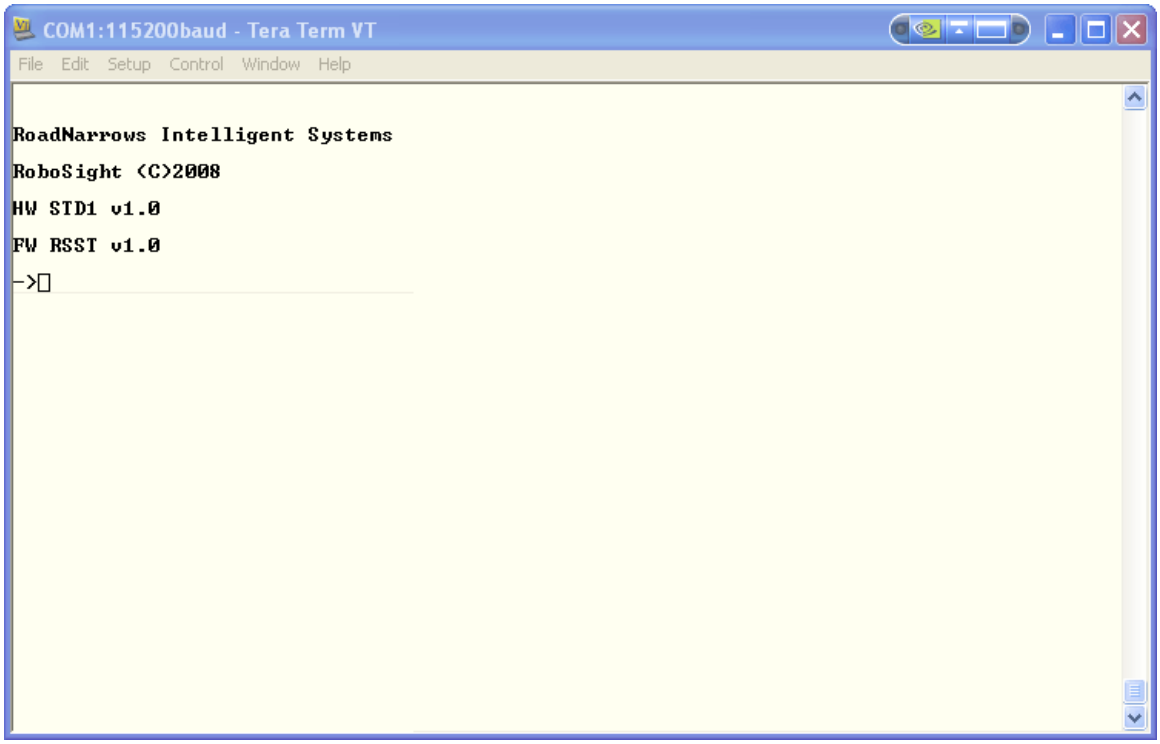


Figure 25: Tera Term - RoboSight Boot Strings

Step 5: Enter the Bootloader

Enter the Reset command ('x') and press 'b' within the first 2 seconds of the reset.

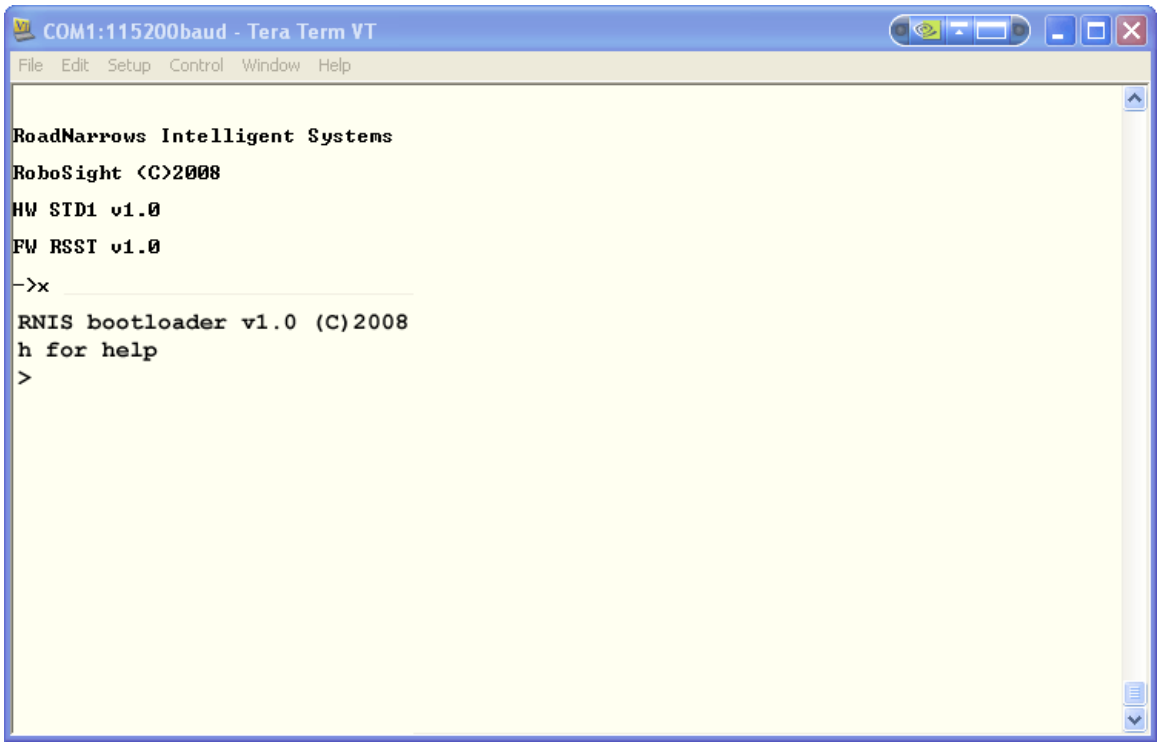


Figure 26: Tera Term - Reset and Enter Bootloader

Step 6: Download the HEX File

At the bootloader prompt enter the load file command '`l`'. The bootloader is now waiting for the file to be sent. At the Tera Term window, select the **Send file...** item in the **File** menu.

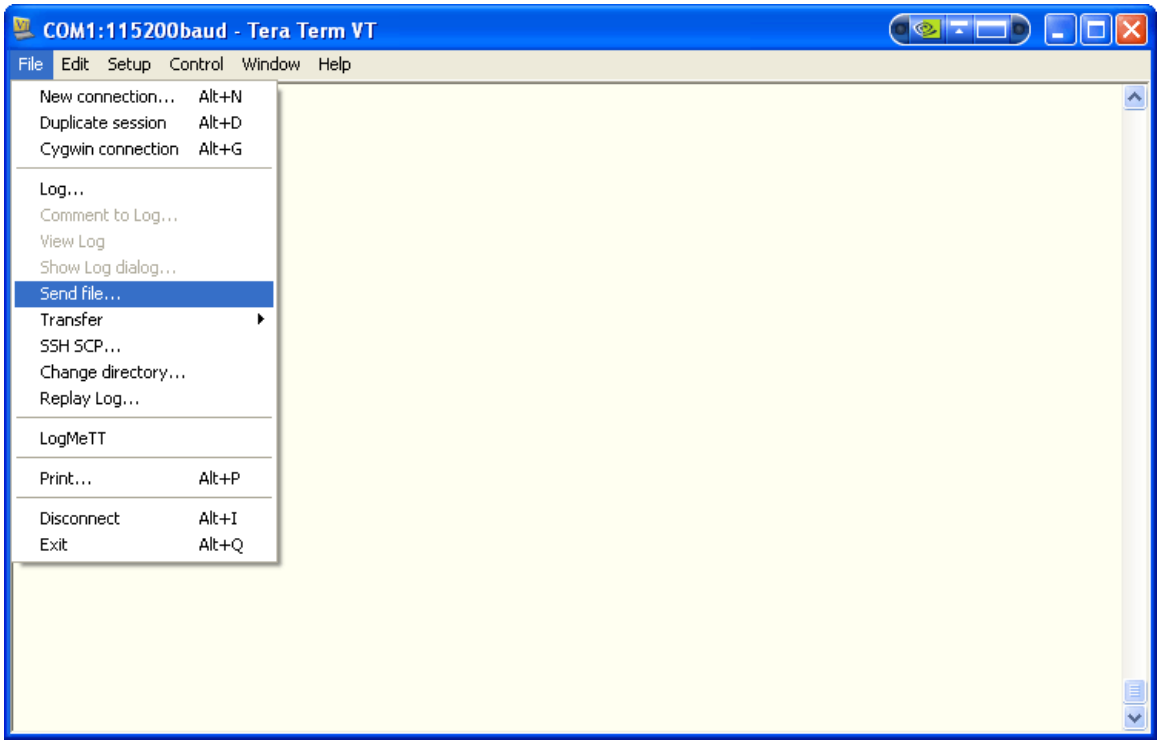


Figure 27: Tera Term - File Send Menu

Select the HEX file. A **Send File** progress dialog box will appear show the progress of the file transfer.

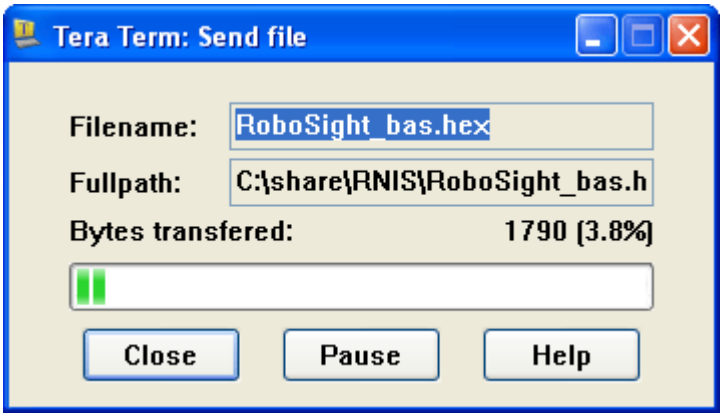


Figure 28: Tera Term - Send File Progress Box

After the file has been transmitted, the bootloader prompt will show. Enter the Jump command '`j`' to jump to the newly downloaded main firmware.

REFERENCES

[epa1] Basic Information | eCycle | US EPA. 3 Nov. 2008. United States Environment Protection Agency. 18 Dec. 2008. <www.epa.gov/epawaste/conserve/materials/ecycling/basic.htm>.

[ov6630] "OV6630/OV6130 Data Sheet". Version 1.0. 4 Mar 2000. OmniVision. <www.roadnarrows.com/customer/RoboSight/latest/doc/OV6630DataSheet.pdf>.

[c3038] "C3038 Data Sheet". COMedia LTD. <www.roadnarrows.com/customer/RoboSight/latest/doc/C3038DataSheet.pdf>.

[hex] Intel HEX – Wikipedia, the free encyclopedia. 30 Sep. 2008. Wikimedia Foundation, Inc. 21 Dec. 2008. <http://en.wikipedia.org/wiki/Intel_hex>.

[ttssh] Tera Term Open Source Project. 15 Dec. 2008. TeraTerm Project. 21 Dec. 2008. <http://ttssh2.sourceforge.jp/>. <http://sourceforge.jp/projects/ttssh2/>.

[zisc] Zero Instruction Set Computer – Wikipedia, the free encyclopedia. 10 Nov. 2008. Wikimedia Foundation Inc. 23 Dec. 2008. <http://en.wikipedia.org/wiki/Zero_Instruction_Set_Computerr>.

[cm1] "CogniMem CM-1K Data Sheet". Version 1.2. General Vision. <www.roadnarrows.com/customer/CogniMem/latest/doc/CM1K_datasheet.pdf>.

[cm2] "CogniMem Referenced Guide". Version 1.2.1. General Vision. 1 Feb. 2008. <www.roadnarrows.com/customer/CogniMem/latest/doc/CogniMemReferenceGuide.pdf>.

[rbf] Radial basis function– Wikipedia, the free encyclopedia. 21 Dec. 2008. Wikimedia Foundation, Inc. 23 Dec. 2008. <http://en.wikipedia.org/wiki/Radial_basis_function>.

[knn] k-nearest neighbor algorithm – Wikipedia, the free encyclopedia. 7 Dec 2008. Wikimedia Foundation Inc. 23 Dec. 2008 <http://en.wikipedia.org/wiki/KNN>.

[norm] Vector Norm – from Wolfram MathWorld. MathWorld. 23 Dec. 2008. <http://mathworld.wolfram.com/VectorNorm.html>

[mlp] Multilayer perceptron – Wikipedia, the free encyclopedia. 12 Dec. 2008. Wikemedia Foundation Inc. 23 Dec. 2008. <http://en.wikipedia.org/wiki/Multilayer_perceptron>.

[ann] Artificial neural network – Wikipedia, the free encyclopedia. 6 Dec. 2008. Wikimedia Foundation Inc. 23 Dec. 2008. <http://en.wikipedia.org/wiki/Artificial_neural_network>.