Communication - Intel. Network - Sensor Applications

AN2347

Author: Andrew Smetana

Associated Project: Yes
Associated Part Family: CY8C24xxxA, CY8C27xxx, CY8C29xxx
GET FREE SAMPLES HERE
Software Version: PSoC® Designer™ 4.2 SP3
Associated Application Notes: AN2346, AN2105, AN2245

Application Note Abstract

This Application Note expands on *AN2346 Communication - Intel. Network - Physical and Data Layers*, which described the network operation principles and protocol. This note provides examples of the network terminal nodes. The hardware and firmware aspects of terminal node development are examined in detail.

Introduction

Sensor networks are developed for many uses. The principal tasks of sensor networks are monitoring, collecting ambient parameters, and making decisions about control of the parameters. The main network device is the network host, which performs all these activities. The host is very important, but so is another kind of node – a terminal network device that executes a particular function. Such nodes are numerous and differ according to function. They include sensors that collect environmental parameters, control levers (buttons) for remote control of other nodes, and actuating mechanisms that provide for control of power machines, lighting systems, and so on. The most widespread sensors are those that monitor and help control temperature, lighting, humidity, pressure, motion (PIR), smoke, and liquid levels.

This Application Note describes the following systems in detail, and places their implementation in the context of the sensor network that was proposed in AN2346:

- Remote Temperature Sensor
- Lighting Control System
- Infrared Motion Sensor
- Smoke Detector
- UART and Network Controller

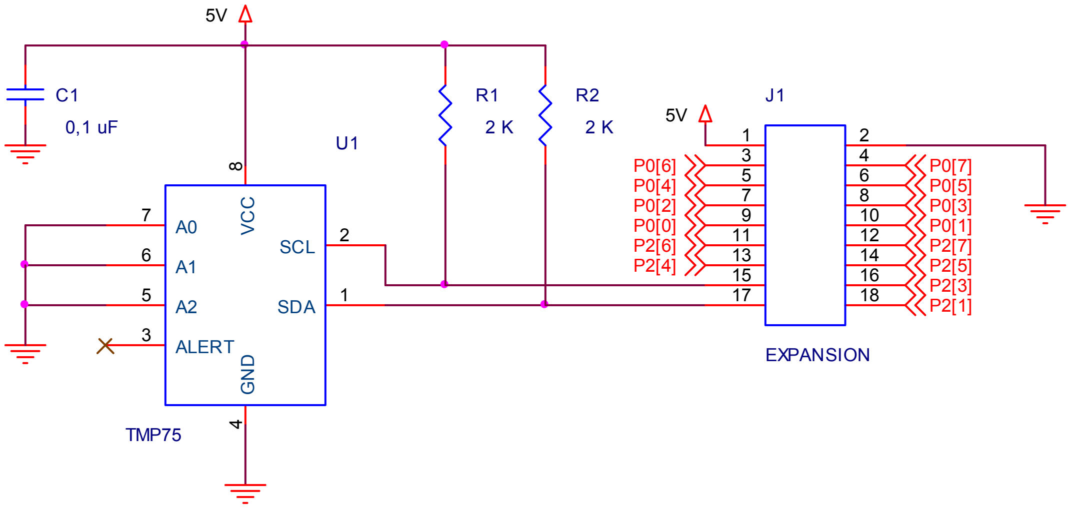
The schematics for these devices were developed as expansion boards to be plugged into the basic node board (refer to Application Note AN2346, Figure 13).

Remote Temperature Sensor

The intelligent wired sensor can be used for remote temperature monitoring of homes, offices, warehouses, greenhouses, hothouses, granaries and many industrial systems. The sensor expansion board schematic is very simple, as shown in Figure 1.

The sensor uses a (Texas Instruments) TMP75 equipped with an I2C interface. The sensor is configured through the I2C interface and returns temperature code. In this implementation, the code must be multiplied by 0.0625 to get the temperature value in degrees Celsius. The sensor is accurate within ±1.5°C in a temperature range of -25 to +80°C.

Figure 1. Temperature Sensor Schematic



The PSoC device utilizes only one I2C User Module in this implementation. The received temperature code is translated to the host and is represented in the terminal program. This node is assigned to address 5. The address is coded in the hardware.

Lighting Control System

The intelligent wired sensor can be used for remote lighting control of offices, high-rise buildings, large garages, warehouses, backup lighting systems, and so on. At present, lighting control systems in these venues usually have lights that must be switched on and off separately in many places. Such systems typically require multiple wires and are not cost effective.

The proposed wired sensor network can be used for these varied home and workplace applications. The structure of the lighting control network is shown in Figure 2. Each node in the network has two buttons to switch on or off one or more lights. The buttons also can be implemented by using CapSense technology (capacitive sensing powered by PSoC). The lighting can be managed centrally from the network controller. The broadcast messages can be used for all lights in the system simultaneously. Within each 24-hour cycle, the light sensor automatically turns on a light when darkness falls and turns off the light when daylight returns. The state of each light can be queried remotely to easily identify nonfunctioning lights.

The electric circuit of the lighting control node is shown in Figure 3. The illumination current in the light is controlled by opto-triac U1. The presence of the current in the light is sensed by the T1 current transformer.

The output of the transformer is a sinusoidal signal, the amplitude of which is proportional to the current in the light. The T1 sensor signal is limited by diodes D2 and D3, after which it feeds the internal PGA. An 8-bit ADC converts the amplified signal to digital representation. To remove the DC component, the ADC data stream is passed to the digital HPF. After this, the ADC data stream is rectified and integrated during one AC power period. If the integrator result exceeds the defined threshold, the light is considered on. The block diagram of lighting node PSoC internals is shown in Figure 4.

The (Microsemi) LX1972 is a low-cost silicon light sensor with spectral response that closely emulates the human eye. In the present implementation, this light sensor is designed on another board to eliminate influence of light on the sensor. It measures the current illumination state and sends the corresponding ADC code to the network.

The node firmware sends the current state of the light to the host, and analyzes on/off commands from the network controller. This node is assigned to address 4, in this implementation. The light sensor is address 2.

Figure 2. Structure of the Lighting Control Network

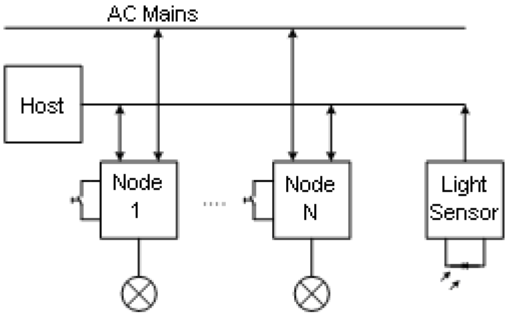


Figure 3. Lighting Control Expansion Board Schematic

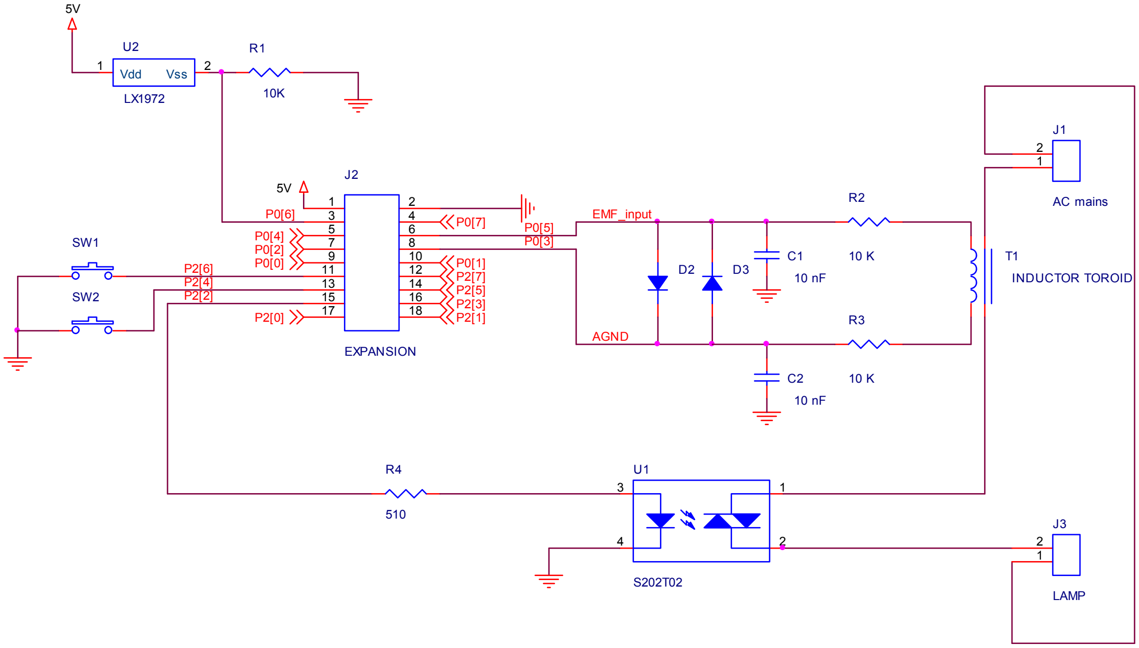
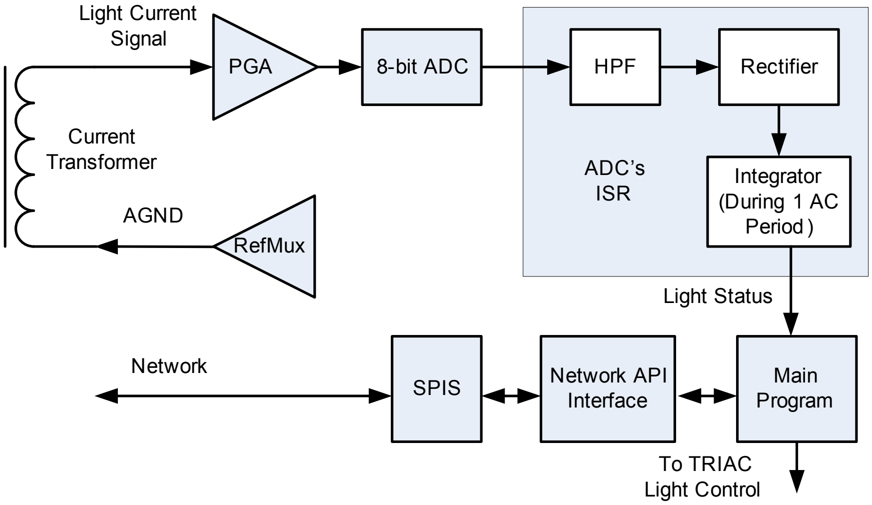


Figure 4. Lighting Node Block Diagram (PSoC Internals)



Infrared Motion Sensor

Infrared motion sensors are widely used in security systems, and a guarded office or residence has at least one detector of this type. The sensor consists of a pyroelectric infrared (PIR) detector, signal amplifier, filters, several comparators, and an alarm switch. The PIR detector is sensitive to human heat emission. A conventional sensor requires one wire pair for the power connection, and another pair for the alarm switch. Dedicated concentrators are used to process data from several sensors. These conventional sensor systems require additional installation and servicing of separate alarm switches, wiring and concentrators. Also, in these conventional systems, PIR detectors are grouped, so identification of a faulty detector is a very complex task.

The proposed network of addressable sensors does not have these drawbacks. Also, the wired sensor network can include other kinds of sensors. Application Note AN2105 “Pyroelectric Infrared Motion Detector, PSoC Style,” describes the operating principles of PIR detectors and their implementation using PSoC. The node schematic is shown in Figure 5. The PIR detector U1 generates a signal that is passed through the LPF assembled from C3 and R3, at a frequency of 10 Hz. The HPF consists of the components C2 and R2. This circuit shifts the signal so that it is centered at the AGND circuit. The signal is amplified, digitized, and processed in the firmware. The sample data is passed through the HPF and the decimator. The result is compared to allowed limits. If the result is out of range, then movement is detected and the L1 LED turns on for 3 seconds. The firmware continually translates the current environmental state to the host (the presence or absence of movement).

A block diagram of the PSoC configuration is shown in Figure 6. Generally, the PIR node transmits current sensor data (filtered) to the host, and decisions about threshold crossing are made on the host side.

Figure 5. Infrared Motion Sensor Expansion Board Schematic

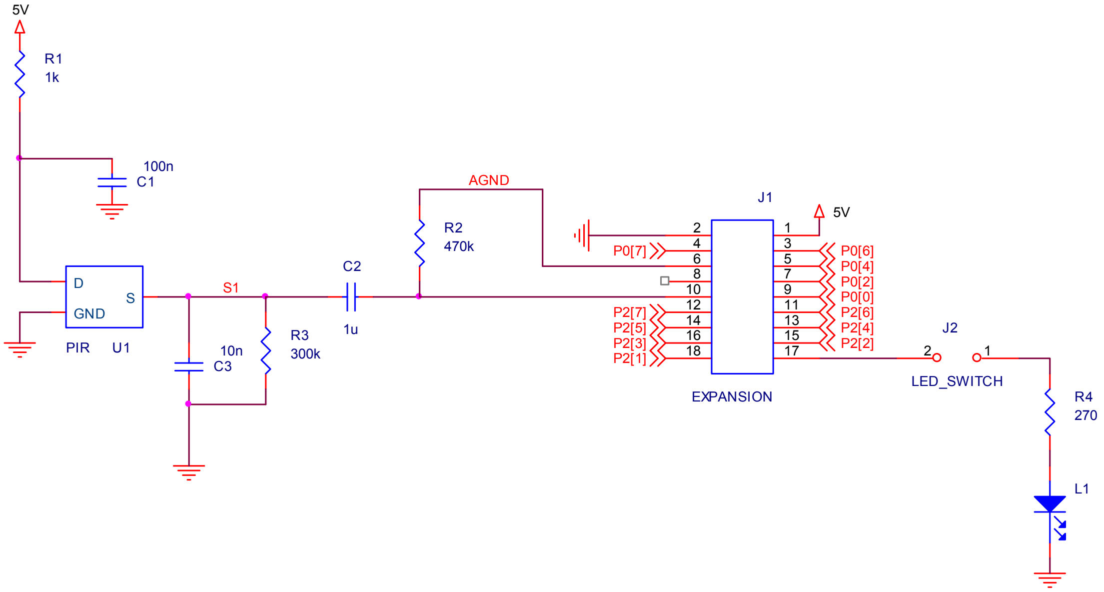
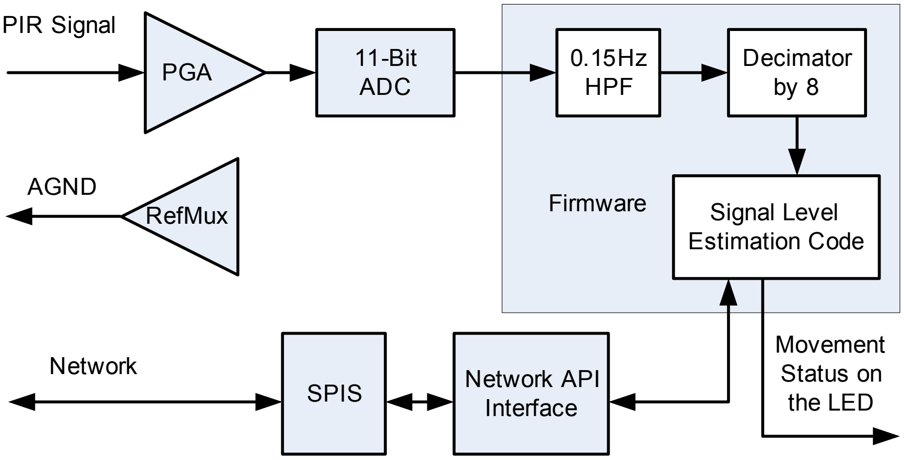


Figure 6. Infrared Motion Sensor Block Diagram (PSoC Internals)

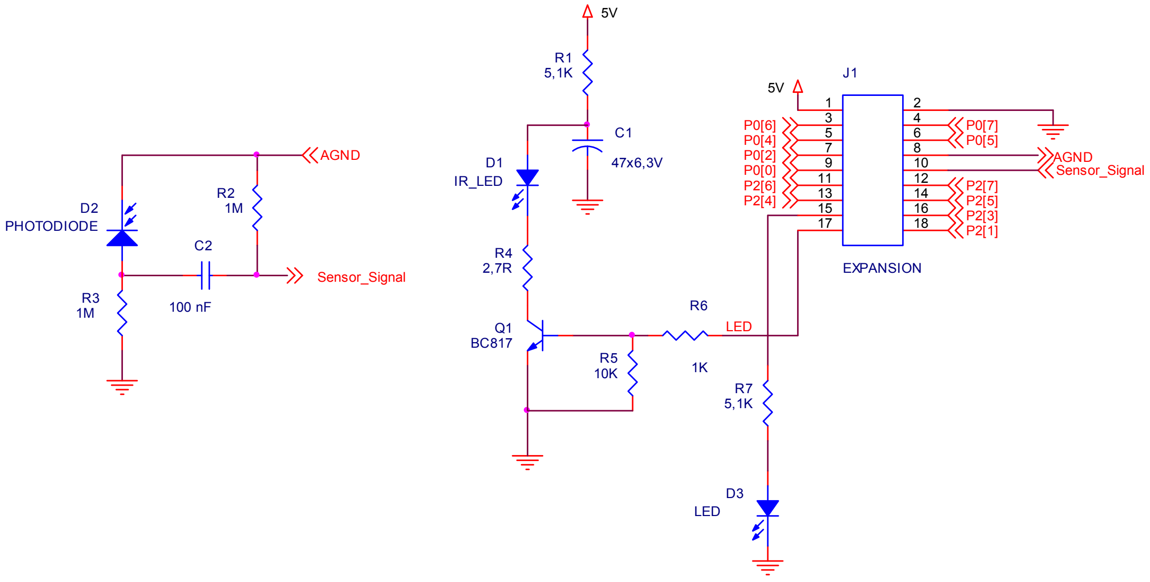


Smoke Detector

Modern fire alarm systems use electronic smoke detectors to detect fire. International safety standards require that every home, office and industrial room be equipped with a smoke detector. Most fire alarm systems use smoke detection to detect the presence of fire. There are two types of smoke detectors: ionization and photoelectric. The proposed smoke detector uses photoelectric detection.

Application Note AN2245 "Smart Smoke Detector," describes this device implementation. It addresses theoretical aspects of smoke detectors and describes implementation of the smart smoke detector in detail. AN2245 describes the foundation of the networked smoke sensor design. The network features of the node require only one SPIS User Module, PSoC port 1, and appropriate firmware. The device schematic is shown in Figure 7.

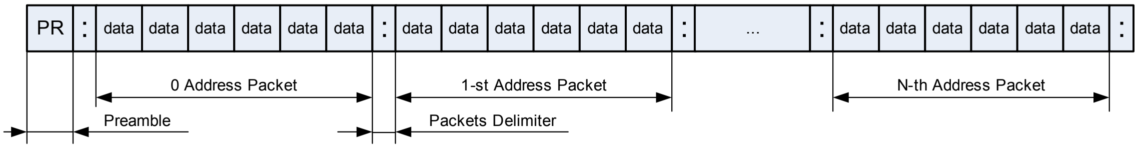
Figure 7. Smoke Detector Schematic



UART and Network Controller

The universal asynchronous receiver/transmitter (UART) connects the network controller device to the host. Data packets are passed back and forth through this channel between the controller and the host. On the host side, the packets are buffered and retransmitted to the line; the host does not analyze them. The information in the packets is analyzed only by the network controller and nodes. However, the host examines certain kinds of service packets. The host uses the service packets to transmit its current status to the controller, and the controller sends commands to the host. The particular features of UART protocol construction are examined in this section.

Figure 8. UART Information Content



The structure of a single transaction between host and controller is shown in Figure 8. The diagram is true for both RxD lines and TxD lines. The transmission sequence during one working network cycle is as follows:

After the preamble is detected on the host, the protocol is initialized and sends the controller a “PR” signature to inform the controller of the start of a new cycle. The boundary between packets is determined by the ":" delimiter. Then, each time the host receives a packet from the network, the data is translated through the UART. The controller, after receiving the 0 packet from the host, processes it, and sends the preamble “PR” to the host and the response to the 0 address. Then the controller receives the next packet, forms the response, and sends it to the host. In Figure 9, a flowchart of network controller functioning is shown.

Note that each packet byte is translated through the UART in hexadecimal. That is, two ASCII symbols in the UART correspond to one packet byte. For example, 0x5A byte is translated into the symbols ‘5’ and ‘A’. This approach distinguishes the control protocol symbols from the informational symbols.

In addition to broadcasting messages, the 0 packet is used as a service slot. The host uses the packet to inform the controller about its current state, and the controller sends control commands (stop/restart network) through the packet. The control/status information is put into the service packets.

The packets cycle within the UART channel until the carrier is present in the line. The service packets can be sent at any time; they need not wait their turn in the working cycle.

Network controller implementation in a Windows (Win32) API is shown in Figure 10. A Win32 API is useful for debugging during development of the sensor network.

Figure 9. Network Controller Algorithm

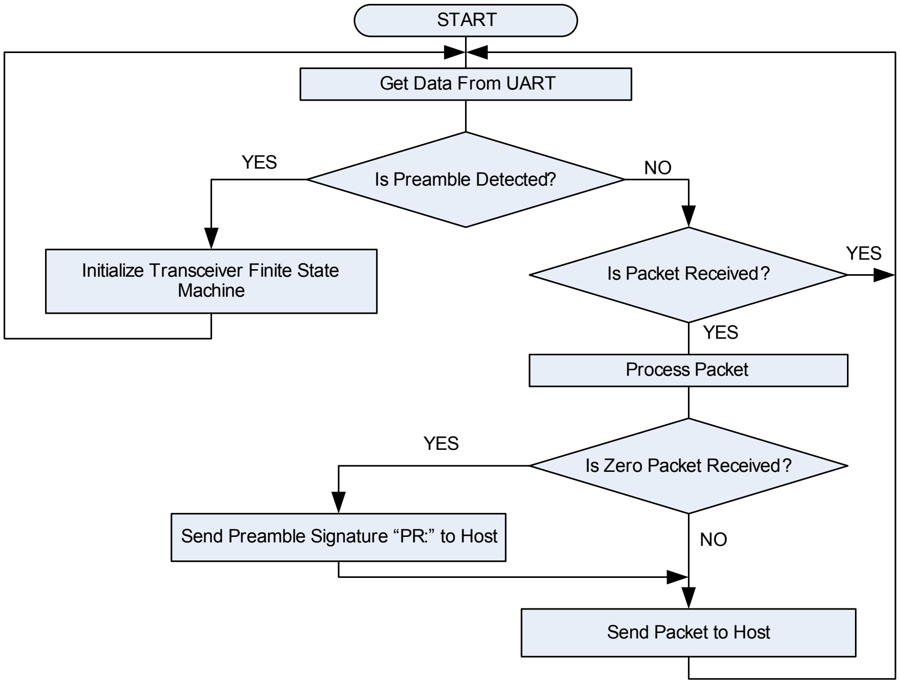
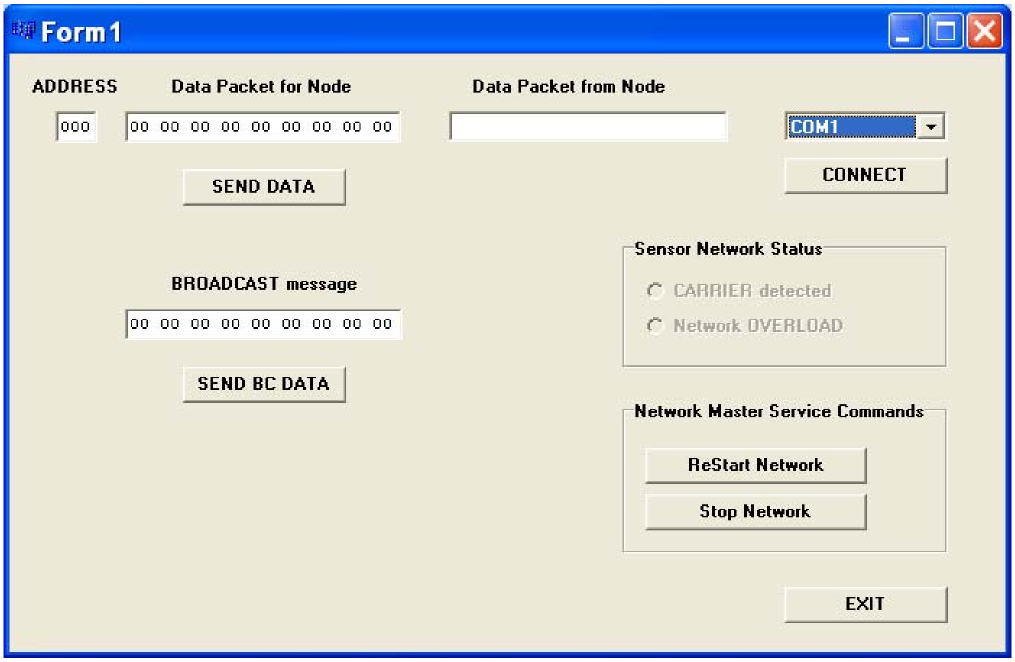


Figure 10. A Network Controller Win32 API



ADDRESS 000

Data Packet for Node 00 00 00 00 00 00 00 00 00

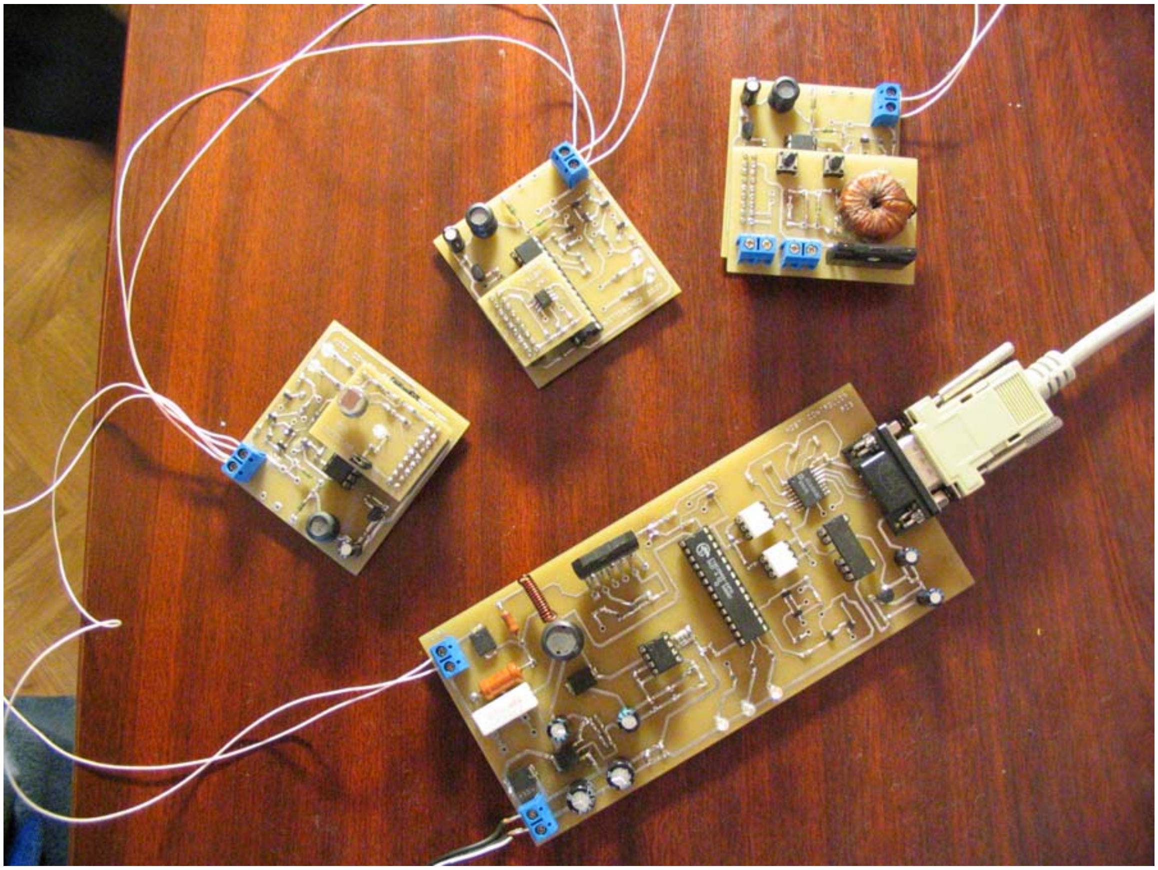
BROADCAST message 00 00 00 00 00 00 00 00 00

COM1

Summary

The network application area of the low-cost sensor network is by no means limited to the applications described in this Application Note. A nearly infinite number of applications can be found. The node implementation method (for basic and expansion boards) described in this note is very suitable for node prototyping, but the finalized devices should have a customized form factor. Multi-purposed sensor networks can be designed by using this Application Note and Application Note AN2346.

Figure 11. Implementation of Proposed Low-Cost Sensor Network



About the Author

Name: Andrew Smetana
Title: Electrical Engineer
Background: Andrew received a Master of Science degree from National University “Lvivska Polytechnika“ (Ukraine) in 2004. His interests include various aspects of embedded systems development.

Contact: smetana@ukrwest.net

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable System-on-Chip," PSoC Designer, and PSoC Express are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2006-2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress’ product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.