CC430-Based Simple Wireless Mesh Network Mains Switch

Vincent Chan, Steve Underwood, Mars Leung, Roc Yu

MSP430 Systems Application

ABSTRACT

This application report describes a simple wireless mesh network running on a number of wireless power points. Each power point contains a timer switch and a power monitor. The mesh network is self learning. A data concentrator works through the various power points it discovers and forms a network. It can then program the on/off timing of the power point’s switch and monitor its power consumption. The mesh network requires a small amount of system resources and is designed to run on top of existing real-time embedded applications, for example, electricity measurements.

The CC430F6137 is a monolithic MCU+RF transceiver from Texas Instruments. The device combines the ultra-low-power MSP430™ core with a CC1101 RF transceiver on a single piece of silicon. A single CC430 runs the mesh network, electricity measurement, real-time clock and timer switch function.

Project collateral and source code discussed in this application report can be downloaded from the following URL: http://www-s.ti.com/sc/techlit/slaa487.zip.

Contents

1 Introduction ................................................................. 2
2 Network System Description ............................................. 3
3 Concentrator ................................................................. 4
4 Node ........................................................................... 5
5 Forming the Mesh Network .................................................. 6
6 Electricity Measurement ................................................... 22
7 Setting up the RF Interface .................................................. 22
8 Major Software Routines .................................................... 27
9 Reference .................................................................... 27
Appendix A Examples of the HyperTerminal Message Exchanges ..................................................... 29

List of Figures

1 Whole System ................................................................. 3
2 Concentrator ................................................................. 4
3 Node ........................................................................... 5
4 Initial State .................................................................... 6
5 Discovery ....................................................................... 7
6 Node 1 Responds ............................................................. 8
7 Register Node 1 ............................................................... 9
8 Send Discovery Again ....................................................... 10
9 Node 2 Responds ............................................................. 11
10 Register Node 2 .............................................................. 12
11 Ask Node 2 to Discover ...................................................... 13
12 Node 2 Sends Discovery .................................................... 14
13 Node 5 Responds First ...................................................... 15
14 Node 4 Responds ............................................................. 16
15 Node 2 Sends Back the First Response .................................. 17
1 Introduction

In designing a wireless system, there is often a need to form networks of various kinds. This could be a star network with a handheld concentrator talking to several nodes. Or, a mesh network where the range can be extended through hops. Furthermore, there is a need to freely associates nodes in a network or to discover which nodes are active.

This application report describes a mesh-network software that fulfills the above requirements and is implemented in less than 4Kbyte of code and 200 bytes of RAM (depending on network size), which leaves ample system resources for other functions. In this case, a full electricity metering function is working concurrently on the same chip.

The proposed network consists of a reasonably powerful concentrator to master a number of monitored nodes. The monitored nodes use the single chip, CC430, to support the networking protocol.

The concentrator maps all the nodes at regular intervals by probing outwards step-by-step to the extremes of the mesh. At each stage, it gathers and stores all node to node connectivity information worked out by the nodes. Nothing is stored permanently in the nodes that would consume memory resources. The concentrator is in complete control of the mesh; it requests actions of the nodes and expects responses from them. A probing scheme is used to map out the mesh. It re-probes at frequent intervals to find any changes in the mesh, whether they are genuine changes in the meters within the area, changes due to node failures, or communications difficulties. Based on the mapped out mesh, the concentrator regularly reads status, usage and other required information from the nodes in the mesh. It can also set the time and schedule the switching of the mains switches available in the nodes to ON and OFF. The concentrator can be implemented into a handheld unit to facilitate portable applications.

MSP430 is a trademark of Texas Instruments.
Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
All other trademarks are the property of their respective owners.
2 Network System Description

The mesh network consists of a single concentrator and a number of nodes.

The concentrator's tasks are:
- Continuously discover new nodes
- Build a network table
- Calculate the best path to communicate with each node
- Exchange data with each node
- Provide a gateway to the web serve

The node's tasks are:
- Respond to discovery from the concentrator
- Pass on routing messages to other nodes
- Process commands from the concentrator
- Turn on/off the power point
- Provide a calendar function
- Measurement electrical parameters

Master
- Wireless Meter Reading
- Mesh network, greater coverage
- You can get data through the Internet

Slave
- Measurement/RF/Switch/RTC in one single chip
- Voltage & current sample, power & consumption calculate
- Ultra low-power wireless measure data communication
- Schedule on/off: you can leave the electronic equipment connected to the switch work alone

Figure 1. Whole System
3 Concentrator

The concentrator is built using a PC and a MSP-EXP430FG4618 experimental board with a plug-in RF module. The frequency is set to 433 MHz.

![Concentrator Diagram](image-url)

Figure 2. Concentrator
4 Node

The node is a self-contained unit and looks like a common mains-switch product available from many appliance stores (see Figure 3).

Figure 3. Node
5.1 Workflow

5.1.1 Initial State

Using the example where there are initially five new nodes, unknown (orange) to the concentrator (blue), nodes 4 and 5 are outside the concentrator's direct communication path (see Figure 4).

![Figure 4. Initial State](image-url)
5.1.2 Discovery and Finding Nodes

The concentrator sends the first discovery signal (see Figure 5). Upon receiving this signal, the nodes each wait a random number of time slots (a total of 32 possible slots) before replying.

![Diagram showing nodes and discovery process]

**Figure 5. Discovery**
In this example, node 1 responds first (see Figure 6).
Then, the concentrator sends out an acknowledgment telling the node it has been registered (see Figure 7). The concentrator uses the RSSI information in the node’s response package to construct the link quality table, as each node has multiple access paths. This link quality table is used to determine the most optimal path.
The concentrator sends out a fresh discovery (see Figure 8). Since node 1 has been registered, it does not respond to the discovery from the same source again. Note that it can respond to discovery from another source; this is essential for building a multipath network.

Figure 8. Send Discovery Again
Node 2 responds this time after an internally generated random delay (see Figure 9).

Figure 9. Node 2 Responds
Figure 10. Register Node 2
The concentrator now asks node 2 to send a discovery message (see Figure 11), which enables the concentrator to reach nodes that are beyond its direct communication path.

Figure 11. Ask Node 2 to Discover
Node 2 sends out a discovery (see Figure 12).

Figure 12. Node 2 Sends Discovery
In this scenario, node 5 first responds followed by node 4 (see Figure 13 and Figure 14).

Figure 13. Node 5 Responds First
Figure 14. Node 4 Responds
Node 2 waits for a fixed period of time equivalent to $32 \times 64$ ms to respond (64 ms is the response packet length in the current implementation). This allows all other possible responses before Node 2 replies so there is no collision. After this, Node 2 responds to the concentrator with the first node that responded, in this case node 5 (see Figure 15).

![Diagram of mesh network](image)

**Figure 15. Node 2 Sends Back the First Response**
The concentrator updates its link table until all the nodes (consisting of many layers) have been registered (see Figure 16).

Figure 16. Concentrator Registers Node

The final scenario is illustrated in Figure 17. Each node has more than one path to access it and its own link quality so that the concentrator can choose the most reliable path to reach it at all times.

Figure 17. Route Map
5.2 **Concentrator GUI**

In this application, the concentrator is a GUI program that runs on a PC. An example mesh network is shown in [Figure 18](#).

![Figure 18. Concentrator GUI](#)

<table>
<thead>
<tr>
<th>Node to node signal quality</th>
<th>Readings Power</th>
<th>Consumption</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 8 10</td>
<td>2 512.960022</td>
<td>250840</td>
<td>On</td>
</tr>
<tr>
<td></td>
<td>3 255.72001</td>
<td>31250</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>4 603.559988</td>
<td>0.000000</td>
<td>On</td>
</tr>
<tr>
<td></td>
<td>5 423.586031</td>
<td>0.000000</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>6 659.960022</td>
<td>0.000000</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>8 164.559988</td>
<td>0.000000</td>
<td>Off</td>
</tr>
<tr>
<td></td>
<td>10 0.420000</td>
<td>0.420000</td>
<td>On</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Node to node paths (node (signal strength))</th>
<th>Packet exchange</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 1 (0) 2 (267)</td>
<td>&lt;&lt; Probing from node 3</td>
</tr>
<tr>
<td>3 1 (0) 3 (269)</td>
<td>&lt;&lt; P1P3aPM</td>
</tr>
<tr>
<td>4 1 (0) 4 (380)</td>
<td>&gt;&gt;&gt; [P3P1N(26)]</td>
</tr>
<tr>
<td>5 1 (0) 5 (289)</td>
<td>&lt;&lt; Probing from node 4</td>
</tr>
<tr>
<td>6 1 (0) 6 (189)</td>
<td>&lt;&lt; P1P4PM</td>
</tr>
<tr>
<td>8 1 (0) 8 (183)</td>
<td>&gt;&gt;&gt; [P4P1N(5)]</td>
</tr>
<tr>
<td>10 1 (0)</td>
<td>&lt;&lt; Probing from node 6</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt; P1P5aPM</td>
</tr>
<tr>
<td></td>
<td>&gt;&gt;&gt; P6P1N(17)]</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt; Probing from node 8</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt; P1P6aPM</td>
</tr>
<tr>
<td></td>
<td>&gt;&gt;&gt; [P8P1N(5)]</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt; Probing from node 10</td>
</tr>
<tr>
<td></td>
<td>&lt;&lt; P1P1OPM</td>
</tr>
</tbody>
</table>

The following sections are available in the GUI:

- Text communication table listing the commands sent to the nodes and the responses received
- The readings of the nodes' electricity consumption
- The link table listing the number of nodes discovered and their interconnection (mesh) status
- The best path to reach a node
- Set the nodes' real-time clock and calendar functions

The concentrator's source code is developed using the GCC compiler and available at the following URL: [http://www-s.ti.com/sc/techlit/slaa487.zip](http://www-s.ti.com/sc/techlit/slaa487.zip). Using Windows® operating system (OS), install the Cygwin software. The GUI is developed using the Fast Light Toolkit (FLTK).
5.3 Message Format

Figure 19 shows the message format:

<table>
<thead>
<tr>
<th>Sender Address (32 bits)</th>
<th>Recipient Address (32 bits)</th>
<th>Authentication (32 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timeout (8 bits)</td>
<td>Body Length (8 bits)</td>
<td>Paths Stack (e.g., PxPy.Pz)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Command or Response</td>
</tr>
</tbody>
</table>

The message length is variable and consists of several fields.

Note that the authentication and timeout field is reserved for future expansion and is left zero at the moment.

The remaining fields are:

- The unique ID of the sending node (equivalent to a MAC address)
- The unique ID of the receiving node: a zero recipient address is interpreted as a discovery package.
- The body length
- The path information contains the complete message path; the path is initiated by the concentrator. The receiving node uses this information to determine whether it will forward the message to the next node down the path or if it is the terminating node.
- The command and data: only the terminating node interacts with the command data.

The messages are text-based and can be accessed through a serial port into the nodes. The concentrator is essentially a modified node with an ID of 1 and programmed as a wireless pipe to the PC through an RS232 interface.

This is where you can experience the message exchanges by setting up a HyperTerminal window, sending and receiving messages. For example, to send out a discovery type: P1P0.

Examples of the HyperTerminal message exchanges can be found in the Appendix A.
5.4 **Message Passing**

The concentrator uses node to node message passing to extend the reach of the network. In Figure 20, a packet is sent from node 1 (concentrator) to node 5, passing nodes 2, 3, and 4 along the way. After receiving a packet, each node identifies itself in the path queue. If it finds itself in the middle of the queue, it passes the packet to the next node until the packet reaches the end of the queue.

![Figure 20. Message Passing](image)

Note that the order of the path is reversed in the return path. Figure 20 omits the authentication and timeout information in order to make it simpler to understand.

5.5 **Message Type**

Upon reaching the destination node (last one in the path queue), the node will decode the command and react accordingly.

The following list describes the type of commands currently implemented in this software.

- **Discovery:** 'D'
  The node has been asked to send a discovery. It formulates a packet with zero as the recipient, but the queue information is retained.

- **Lock:** 'L'
  This stops the unit from responding to any discovery package.

- **Unlock:** 'U'
  This enables the unit to respond to the discovery packet.

- **Ack:** 'A'
  The node remembers the send ID and will not respond to subsequent discovery requests from the same ID for the next 5 minutes. This lets the other units respond to discovery requests.

- **Nak:** 'N'
  This is used as a test function.

- **Pass:** 'P'
  The unit is part of a multi-unit path; it passes the message along to the next one on the queue. If it is second in the queue, it sends back the received strength from the first one in the queue so the end node always reports in with the path's strength included.

- **Clock:** 'C'
  This is a command to set unit time.

- **Schedule:** 'S'
  This is a schedule command. When scheduled, the switch will automatically turn ON and OFF.
• **On**: 'O'
  This switches ON the mains supply.

• **Off**: 'F'
  This switches OFF the mains supply.

6 **Electricity Measurement**

This application uses the ADC12 module of the CC430 to effectively measure the electricity consumption of the plug. The method outline used here is similar to the method shown in *Three-Phase Electronic Watt-Hour Meter Design Using MSP430 (SLAA391)*, with the exception that the circuit has been simplified to use only one-phase, whereas, the original application report uses a 3-phase implementation.

7 **Setting up the RF Interface**

The CC430 includes a CC1101-based RF transceiver subsystem. The device contains a RF1A module that is used to interface with the CC1101 subsystem. For a description of the RF1A interface, see the *CC430 Family User's Guide (SLAU259)*.

The RF1A treats the CC1101 as a separate silicon block within the device and uses the RF1A as a communication channel to the CC1101. This is similar to a two-device design in which the MCU interfaces with an external CC1101 device through a serial port.

The structure of the mesh network software is written such that receiving a packet data is dealt within an interrupt routine and sending a packet data is dealt with using polling. In this way, the sending and receiving of packets can be handled concurrently with the e-meter function that the device also runs.
Flow charts of the RF1A routines are shown in Figure 21 through Figure 25.

7.1 RF1A Initialization

Figure 21. RF1A Initialization
7.2 Send Message Flow Chart

Figure 22. Send Message Flow Chart
7.3 Receive Message Flow Chart

Figure 23. Send Message Flow Chart

Figure 24. Receive Message Flow Chart
Figure 25. Receive Message Flow Chart
8 Major Software Routines

Section 8.1 and Section 8.2 describe the key software routines used in this application.

8.1 Metering

• Background.c contains the ADC12 interrupt service routine that performs the sample-by-sample metering calculations.
• Communication.c is used in conjunction with the serial port for calibrating the meter (not activated here).
• Foreground.c contains the metrology functions, e.g., power, current, voltage.
• Measure_task.c contains Power_Task(): the state machine for calculating power and forming the universal asynchronous receiver/transmitter (UART) messages.
• RTC.c, Key.c handles the RTC and keyboard interrupt service routines, and the calendar.

8.2 Mesh Network and Radio

• Decode.c decodes incoming messages and formulate the responses. The main part of the mesh network software.
• Meshnet.c contains the RX and TX package routines, the RF1A interrupt service routines and CC1101 initialization routines.
• Rf_task.c contains RF_Task(): the wireless state machine contains the timer-delay interrupt that handles delay timing.
• Hal_RF1A.c, hal_RfRegSettings.c contains the low-level interface with the CC1101 core and the different register settings for different frequencies.
• Host.c allows the node to transmit the packets it receives to an external PC via its UART port (115200-8-N-1). The packets are text based.

8.3 Schematics

The schematics are included in the Schematics folder, which is available at: http://www-s.ti.com/sc/techlit/slaa487.zip.

9 Reference

• Three-Phase Electronic Watt-Hour Meter Design Using MSP430 (SLAA391)
• CC430 Family User's Guide (SLAU259)
• CC1100 Low-Power Sub- 1 GHz RF Transceiver Data Sheet (SWRS038)
• Software for CC430-Based Wireless Mesh Network Mains Switch available at: http://www-s.ti.com/sc/techlit/slaa487.zip
Appendix A Examples of the HyperTerminal Message Exchanges

Using HyperTerminal setup at 115200-8-N-1, you can send commands manually by entering them in text as seen in the following case examples.

A.1 Sending a Discovery: Case 1

P1P0

Write a message to send out a discovery message from node 1 (node 1 is the master node). Authentication, timeout and body length fields are set to zero.

1, 0, 0, 0, 3 [P1]

The master compiles the message into a data frame and sends it out. The body length is back annotated.

12, 1, 0, 0, 9 [P12P1x208]

Node 12 responded after a random delay time. It tells the master the strength it receives. Since this is a signed value, 208 equals -47. This is given in dB using a look up table. For more information, see the CC1100 Low-Power Sub-1 GHz RF Transceiver Data Sheet (SWRS038).

A.2 Sending a Discovery: Case 2

P1P10P3P8D

Write a message to send out a discovery message from node 8 passing through node 10, node 3 (node 1 is the master node). Authentication, timeout and body length fields are set to zero.

1, 10, 0, 0, 10 [P1P10P3P8D]

Master compiles the message into a data frame and sends it out. The body length is back annotated.

10, 1, 0, 0, 15 [P8P10P1N206]

Node 8 responded with NAK (nothing is discovered) after timeout.
A.3 Sending a Discovery: Case 3

P1P4P9D

Write a message to send out a discovery message from node 9 passing through node 4 (node 1 is the master node).
Authentication, timeout and body length fields are set to zero.

1, 4, 0, 0, 7 [P1P4P9D]

Master compiles the message into a data frame and sends it out. The body length is back annotated.

4, 1, 0, 0, 13 [P2P9P4P1x8r14]

Node 2 responded after a random delay time. x8 is the RF strength from node 9 to node 2. r14 is the RF strength from node 2 to node 9.

A.4 Pass a Message

P1P4P9P12L

You type the message on the screen, Authentication, timeout, and body length fields are set to zero.
The message is sent from node 1 to node 4 (node 1 is the master). It asks node 4 to pass a Lock command to node 12 via node 4 and node 9.

1, 4, 0, 0, 10 [P1P4P9P12L]

Master compiles the message into a data frame and sends it out. The body length is back annotated.

4, 1, 0, 0, 16 [P12P9P4P1x74r78]

Node 12 acknowledges this command back to node 1 (this can be seen in the path), the last stop is node 4. The RF strength received by node 12 from node 9 and transmitted by node 12 to node 9 is included to help the master to keep track of the network's status.
IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal and regulatory requirements concerning such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal and regulatory requirements concerning such use.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or “enhanced plastic.” Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Communications and Telecom</td>
</tr>
<tr>
<td>Amplifiers</td>
<td>Computers and Peripherals</td>
</tr>
<tr>
<td>Data Converters</td>
<td>Consumer Electronics</td>
</tr>
<tr>
<td>DLP® Products</td>
<td>Energy and Lighting</td>
</tr>
<tr>
<td>DSP</td>
<td>Industrial</td>
</tr>
<tr>
<td>Clocks and Timers</td>
<td>Medical</td>
</tr>
<tr>
<td>Interface</td>
<td>Security</td>
</tr>
<tr>
<td>Logic</td>
<td>Space, Avionics and Defense</td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Transportation and Automotive</td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Video and Imaging</td>
</tr>
<tr>
<td>RFID</td>
<td>Wireless</td>
</tr>
<tr>
<td>RF/IF and ZigBee® Solutions</td>
<td><a href="http://www.ti.com/cgi">www.ti.com/cgi</a></td>
</tr>
</tbody>
</table>

TI E2E Community Home Page: e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated