LED Lighting Control Using Powerline Communication

Author: Rohan Gandhi

Associated Project: Yes

Associated Part Family: CY8CPLC20

Software Version: PSoC® Designer™ 5.4

This application note describes LED control using Powerline Communication. It also discusses the Auto Node Discovery algorithm, which is a part of LED control.

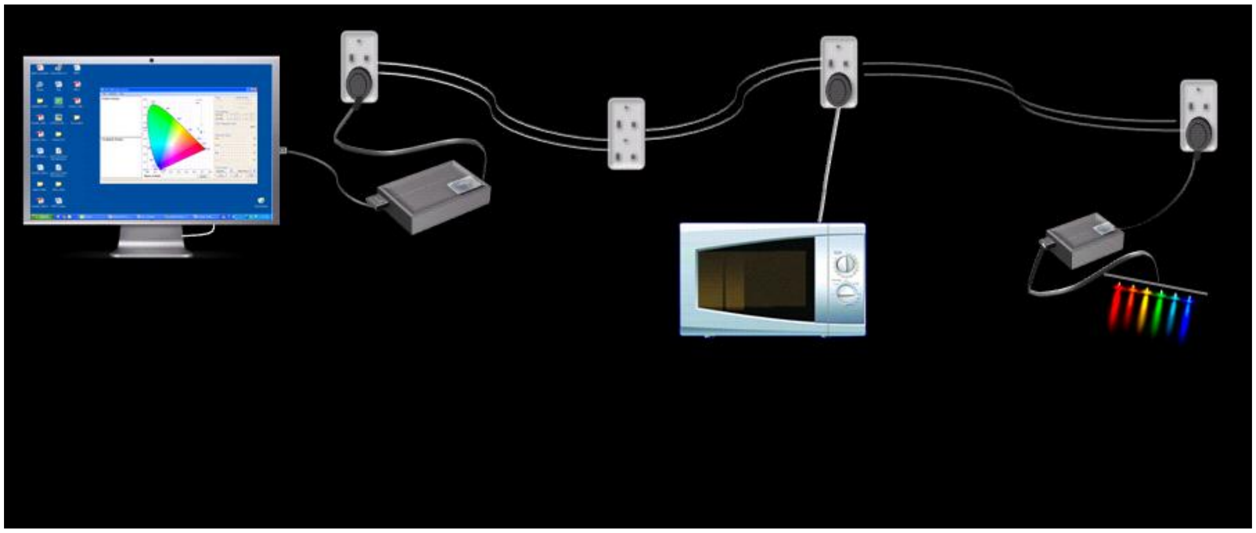
Contents

Introduction

Cypress provides a robust solution to implement Powerline Communication (PLC) for command and control applications such as lighting control. The CY8CPLC20 device combines the robustness and ease-of-use of the PLC solution with the configurability and flexibility of the PSoC® core. The CY8CPLC20 device allows precise color control with a high brightness (HB) LED controller, based on the Cypress EZ-Color™ technology. The CY8CPLC20 device gives the ability to run your own application. The Powerline Transceiver (PLT) User Module manages the network protocol and Physical Layer FSK modem, which transmits and receives messages over the Powerline.

This application note describes a LED lighting control system. The master for this system is either an RGB control GUI or an embedded controller based on CY8CPLC20. The slave node is the CY3276 board with the CY8CPLC20 part, which receives the commands from the master and accordingly controls the LEDs.

Figure 1. LED Control



Powerline Communication

Powerlines are a widely available communication medium all over the world for PLC technology. The pervasiveness of Powerline also makes it difficult to predict the characteristics and operation of PLC products. Because of the variable quality of Powerlines around the world, implementing robust communication over Powerline has been an engineering challenge for years. The Cypress PLC solution enables secure and reliable communication over Powerline. The features of the Cypress PLC solution include:

- Integrated Powerline PHY modem with optimized filters and amplifiers to work with high voltage and low voltage Powerlines.
- Powerline optimized network protocol that supports bidirectional communication with acknowledgement based signaling. In case of data packet loss due to louder noise on the Powerline, the transmitter has the capability to retransmit the data.
- The Powerline network protocol also supports 8-bit CRC for error detection and data packet retransmission.
- A Carrier Sense Multiple Access (CSMA) scheme is built into the network protocol; it minimizes collision between packet transmissions on the Powerline, supports multiple masters, and enables reliable communication on a bigger network.

HB LED Controller

The HB LED Controller is based on the Cypress EZ-Color™ technology. EZ-Color offers the ideal control solution for high-brightness LED applications requiring intelligent dimming control. EZ-Color devices combine the power and flexibility of PSoC (Programmable System-on-Chip) with Cypress PrISM™ (Precise Illumination Signal Modulation) technology that provides lighting designers with a fully customizable and integrated lighting solution platform.

The CY8CPLC20 device supports up to 16 independent LED channels with up to 32 bits of resolution per channel. This provides lighting designers the flexibility to choose the LED array size and color quality. The PSoC Designer software, with lighting-specific user modules, significantly cuts development time and simplifies implementation of fixed color points through temperature and LED binning compensation. The virtually limitless analog and digital customization of EZ-Color enables simple integration of features in addition to intelligent lighting, such as battery charging, image stabilization, and motor control. During the development process, these features, along with the Cypress best-in-class quality and design support, make EZ-Color the ideal choice for intelligent HB LED control applications.

Networking for LED Lighting Control

In a traditional lighting system, each lighting fixture requires a separate controller (for example, switch). Powerline networks use a bus topology, which provides the ability to control more than one device from one controller. The controller can manage all of the lights in a room or in the home.

Powerline networks use existing Powerlines for communication; therefore, very little cost and effort is required to convert to an LED lighting control system. With a bus protocol, the system is easily reconfigurable and initial setup becomes easier. The packets transmitted contain color and brightness information at a minimum, but can also contain more complex information such as color scenes (pre-defined color patterns of different fixtures) and fading (transitioning between colors).

Functional Overview

The LED lighting control system consists of two main functions:

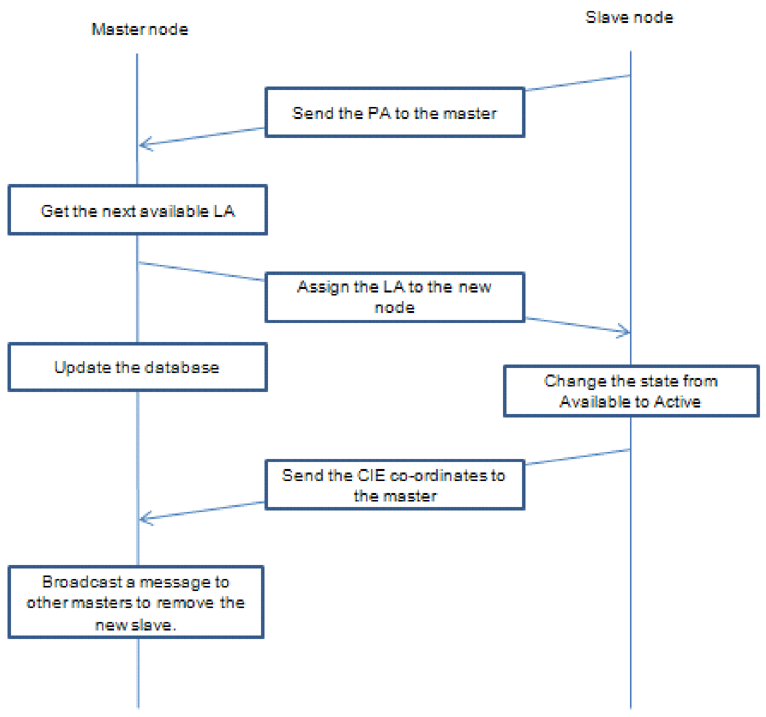
- Node discovery
- RGB LED control

Node Discovery

When a slave node comes on the Powerline network, it is not tethered to any of the master nodes. The slave node has a default logical address of 0x00. It broadcasts its physical address (64-bit physical address is unique to every PLC device) every two seconds so that the master nodes on the network can notice the availability of a new slave node. As the slave node is not bound to any of the masters, it is called an “Available node”.

When you select this slave to control it, the node no longer remains available and is called an “Active node”. To bind with the slave node, the master sends a packet to the slave to configure the logical address for the slave node. The master node also stores the assigned logical address and ensures that this address does not get assigned to any other slave node. After the slave node receives the packet with its logical address, it is no longer available to other master nodes and can be controlled only by the master that assigned the logical address. When the slave binds to the master, it also sends its CIE color gamut for the red, green, and blue coordinates.

Figure 2. Message Passing Diagram for Binding


Master node
Slave node
Send the PA to the master
Get the next available LA
Assign the LA to the new node
Update the database
Change the state from Available to Active
Send the CIE co-ordinates to the master
Broadcast a message to other masters to remove the new slave.

RGB Control

When a slave is tethered to its master, its RGB LEDs are controlled by the master. The color of the slave node LED light can be selected through the CIE coordinates, direct LED control, or temperature control functions available in the master RGB Control GUI.

Note: For demonstration of the example project in the AN, three onboard LEDs on CY3274 kit will be used. There is no actual RGB LEDs on this kit.

Features of LED Control Method

- **Binding to the master:** When you select the available node to control its color, the master node binds to the slave node so that only that particular master can control it. The binding algorithm consists of the following:
- **Dynamic address allocation:** The master node assigns a unique and non-zero logical address to the slave node. The slave node becomes active when it has a non-zero logical address. The slave node stores the logical address of the master node and only processes packets that are received from the master node.
- **Sending the CIE color gamut coordinates:** The slave node sends the CIE color gamut coordinates to the master node. If the master node uses the RGB GUI, then the coordinates are used to form the color gamut which eases the color selection using the CIE coordinates option.
- **Node search:** If there are multiple slave nodes on the network, it becomes difficult to distinguish between all the slave nodes. To overcome this difficulty, all three LEDs flashes on the slave node for a second when the available node is selected in the RGB Control GUI.
- **Node state information:** This feature is provided so that the master nodes can get the status of any slave node (active or available) during run-time.
- **Unbinding:** This feature is provided to unbind the slave node from its current master node. There are two ways the node can unbind:
  - ❑ Press the reconnect key on the slave node.
  - ❑ At the master, select active node and unbind it.
- **RGB control:** This feature is provided to control the LEDs of the slave node. You can control the color of the slave node through one of the following methods:
- **CIE coordinates:** The master node sends the CIE coordinates to the slave node. The slave node runs the color mixing algorithm and calculates the dimming values for red, green, and blue LEDs. In the case of Color Temperature Control, the GUI sends the CIE coordinates.
- **Direct LED control:** The master sends the 8-bit dimming values for each of the red, green, and blue LEDs which are used by the slave node to control the LEDs.
- **Scene information:** This feature is provided to set different scenes.
- **Save scene:** When you set the scene at the master node, the master node sends a packet to the slave node. The slave node then saves the current LED dimming values at the scene number received.
- **Load scene:** To load the scene, the master sends a corresponding packet and the slave retrieves the dimming values for the scene number received and controls the LEDs accordingly.
- **Reset scene:** This feature resets the scene information at the slave node.
- **Fading:** There are three different fading values provided–10, 5, or 2 seconds. The current dimming values are faded depending on the fading rate.
- **Save configuration:** You can set the default scenes stored in a text file, which can be used later.
- **Load configuration:** You can load default saved scenes that can be sent to the slave nodes.

Memory Array

CY8CPLC20 have the user interface through memory array shown in the following table. For more information on individual bits, refer to the data sheet of any of the parts.

Offset	Register Name	Access	7	6	5	4	3	2	1	0
0x 00	INT_Enable	RW	INT_Clear	INT_ Polarity	INT_UnableToTX	INT_TX_ NO_ACK	INT_TX_NO_ RESP	INT_RX_ Packet_ Dropped	INT_RX_Data_Available	INT_TX_ Data_Sent
0x 01	Local_LA_ LSB	RW	8-bit Logical Address / LSB for extended 16-bit address							
0x 02	Local_LA_ MSB	RW	MSB for 16-bit Extended Address							
0x 03	Local_Group	RW	8-bit Group Address							
0x 04	Local_Group_Hot	RW	One Hot Encoded (e.g. if byte = 0b00010001, then member of groups #5 and #1)							
0x 05	PLC_Mode	RW	TX_ Enable	RX_Enable	Lock_ Configuration	Disable_ BIU	RX_Override	Set_Ext_ Address	Promiscuous _MASK	Promiscuous_ CRC_MASK
0x 06	TX_Message_Length	RW	Send _ Message	Reserved		Payload_Length_MASK				
0x 07	TX_Config	RW	TX_ SA_ Type	TX_DA_Type		TX_Service_Type	TX_Retry			
0x 08	TX_DA	RW	Remote Node Destination Address (8 bytes)							
0x 10	TX_CommandID	RW	TX Command ID							
0x 11	TX_Data	RW	TX Data (31 bytes)							
0x 30	Threshold_ Noise	RW	Reserved	Auto_BIU_Threshold	Reserved			BIU_Threshold_Constant		
0x 31	Modem_ Config	RW	Reserved	Modem_TXDelay		Reserved	Modem_ FSKBW	Reserved	Modem_BPS	
0x 32	TX_Gain	RW	Reserved				TX_Gain_Mask			
0x 33	RX_Gain	RW	Reserved					RX_Gain_Mask		
0x 34- 0x 3F	Reserved	RW								
0x 40	RX_Message_INFO	R	New _RX _ Msg	RX_DA_ Type	RX_SA_Type	RX_Msg_Length				
0x 41	RX_SA	R	Remote Node Source Address(8 Bytes)							
0x 49	RX_CommandID	R	RX Command ID							
0x 4a	RX_Data	R	RX Data (31 bytes)							

Off set	Register Name	Acc ess	7	6	5	4	3	2	1	0
0x 69	INT_Status	R	Status _ Value_ Chang e	Reserved	Status_ BUSY	Status_TX _NO_ACK	Status_TX _ NO_RESP	Status_RX_ Packet_ Dropped	Status_RX_ Data_ Available	Status_TX_ Data_Sent
0x 6A	Local_PA	R	Physical Address (8 bytes), "0x6A -> MSB"							
0x 72	Local_FW	R	Version Number							
0x 73	Noise_ Strength	R	Noise Strength on the Powerline							

Use of Custom Command IDs

Cypress proprietary network protocol has a byte reserved for command IDs. The command IDs are divided into two sets:

- For network management (0x01 to 0x0F)
- For communication (0x30 to 0xFF) (also called custom command IDs)

This application uses command IDs from both of the sets. To assign the logical address to the slave node during binding, the master node uses command ID 0x04 (SetRemote_Logical_Address). The remaining features use custom command IDs. For example, the node search feature uses command ID 0x45. When any node receives the packet with this command ID, it flashes the white light. Appendix A covers the command IDs and packet structure for different features. The following table summarizes the command IDs and associated feature.

Command ID	Function	Sent By
0x04	To assign the logical address to the slave node	Master
0x31	To save the scene	Master
0x32	To load the scene	Master
0x33	To reset the scene	Master
0x34	To request/send the RGB coordinates	Master/Slave
0x36	To control the LEDs	Master
0x39	To send the node state information	Slave
0x40	To get the node state information	Master
0x41	To broadcast the physical address	Slave
0x42	To remove the slave node from other master's available nodes list.	Master
0x43	To unbind from the master	Slave
Command ID	**Function**	**Sent By**
0x44	To unbind the slave	Master
0x45	For node search	Master
0x48	To save the default scenes	Master

Associated Projects

The application note contains two projects:

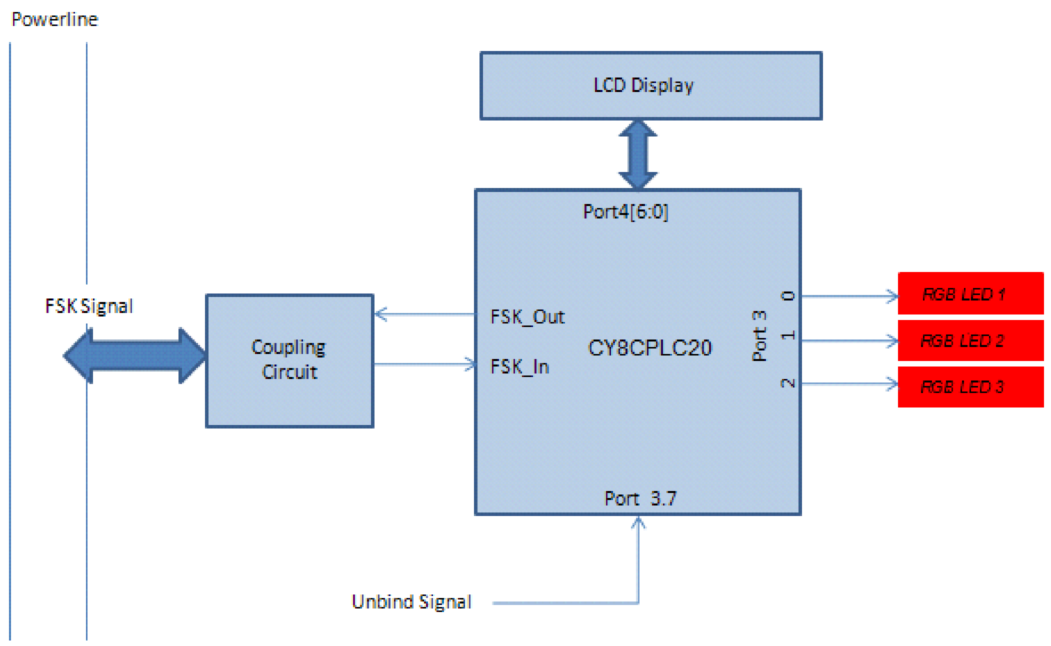
- Embedded master
- Slave device

Slave Device

For this implementation, the CY8CPLC20 part is selected because it has PLC modem and SSDM blocks to control the LEDs. This example project is designed to run on the CY3274 High Voltage PLC DVK with onboard LEDs (not RGB)..

When the slave node receives the color information, it controls the three high-brightness LED drivers using SSDM via Port3[2:0]. The LCD used to display the status information is interfaced through Port 4[6:0]. Port 3.7 is connected to the unbind signal and is used as a GPIO interrupt.

Figure 3. Block Diagram for Slave Node



The following user modules are used in this project:

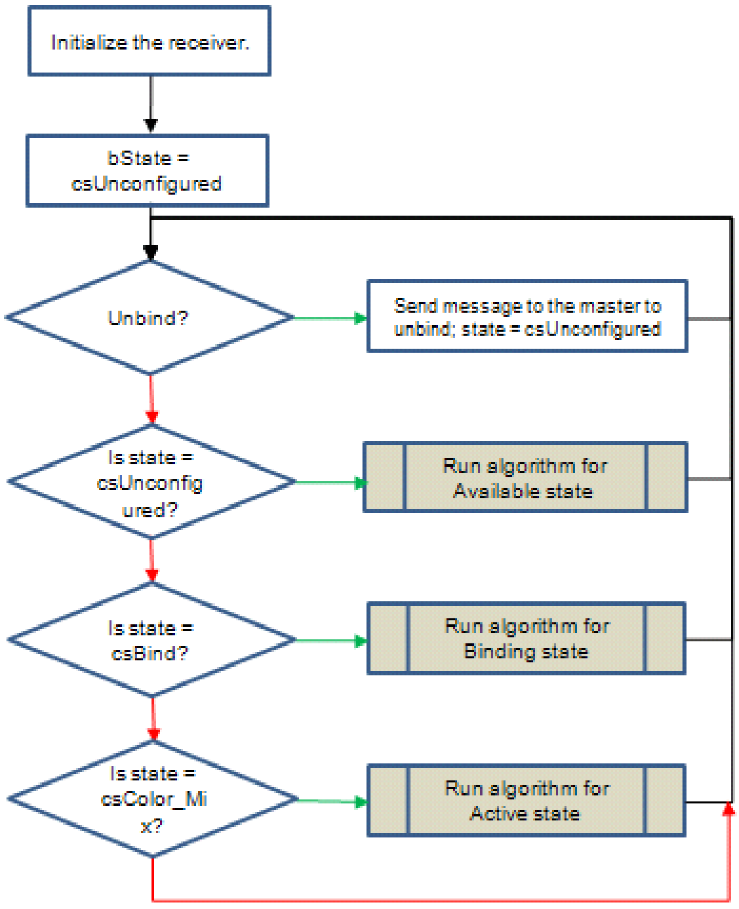
- PLT (“FSK Modem + Network Stack” user module option)
- SSDM (3 UMs) – One UM each for red, blue, and green colors.
- Counter16 – To implement different timings required for
  - Broadcasting physical address every two seconds.
  - Flashing white light for 500 msec (node search)
  - Fading (when loading the scene)

Implementation

The slave node can either be available or active; therefore, overall implementation on the slave side is a state diagram with the following states:

```
csUnconfigured (When the slave node is in available state)
csBind (When the slave node is binding to the master node)
csColor_mix (When the slave node is in the active state)
```

Figure 4. Overall Flowchart for Slave Node

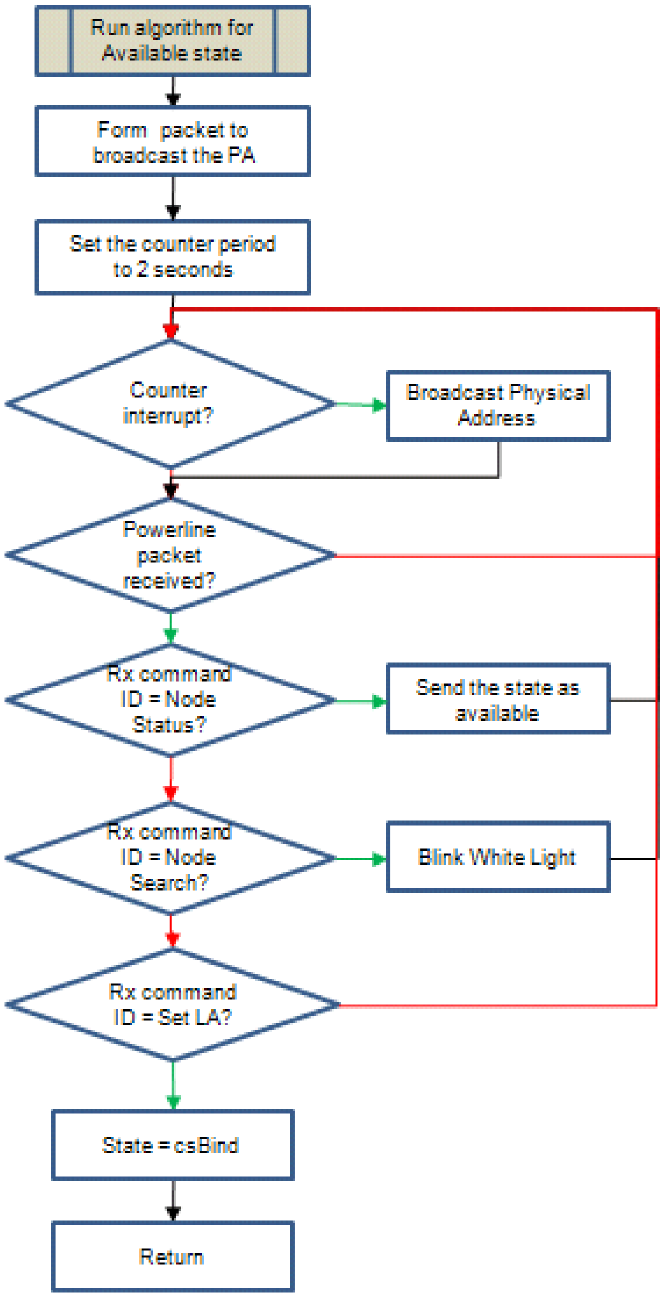

Initialize the receiver.
bState = csUnconfigured
Unbind?
Send message to the master to unbind; state = csUnconfigured
Is state = csUnconfigured?
Run algorithm for Available state
Is state = csBind?
Run algorithm for Binding state
Is state = csColor_Mix?
Run algorithm for Active state

In the flowchart in Figure 4:

- The slave node checks whether the Unbind button is pressed. If the button is pressed, then it notifies the master that it is no longer bound to the master. At the same time, the slave node moves to the csUnconfigured state because it is again available.
- Until the slave node receives binding command from the master, it remains in the csUnconfigured state.
- In the csBind state, the slave node runs the algorithm for binding and goes to the state csColor_mix.

The following figure shows the algorithm for the `csUnconfigured` state.

Figure 5. Flowchart for Slave Node when Unconfigured

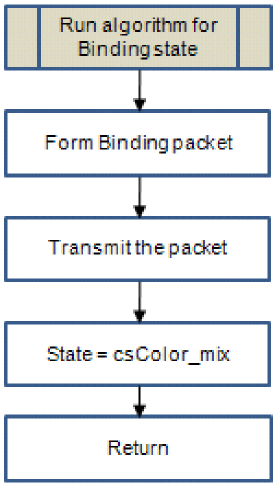

Run algorithm for Available state
Form packet to broadcast the PA
Set the counter period to 2 seconds
Counter interrupt?
Broadcast Physical Address
Powerline packet received?
Rx command ID = Node Status?
Send the state as available
Rx command ID = Node Search?
Blink White Light
Rx command ID = Set LA?
State = csBind
Return

- In this state, the slave device sets the counter period to 2 seconds and prepares the packet to broadcast its physical address.
- On every counter interrupt, the slave node broadcasts its physical address.
- When any packet is received from the Powerline, the slave node checks for the command ID. If the command matches 0x40, then the slave node sends the state information to the node that requested it. The information states that the slave is in the “Available” state.

- If the received command ID is 0x45 (Node Search), then the slave node flashes a white light for 1 second.
- If the received command ID is 0x04 (SetRemote_Logical_Address), then it changes its state to csBind because the master has started the binding algorithm and the slave node is in the process of changing from available to active.

The following figure shows the algorithm for the **`csBind`** state.

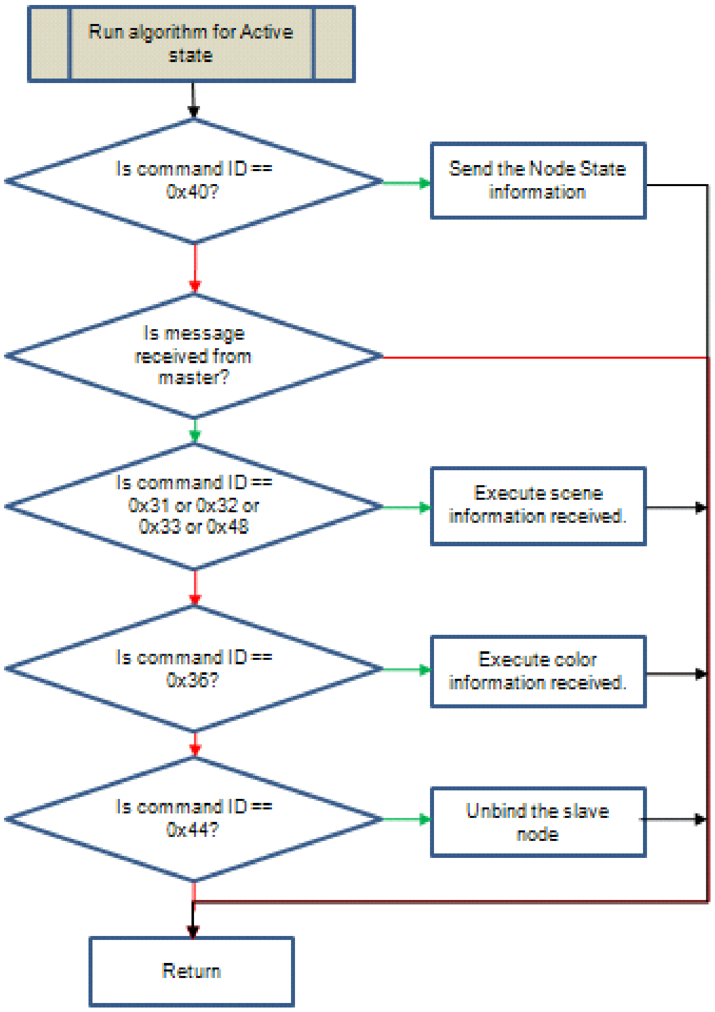
Figure 6. Flowchart for Slave Node in csBind State



In this state, the slave node prepares and transmits the binding packet. The binding packet contains the CIE color gamut coordinates, which completes the binding algorithm. The slave node becomes the active node.

The following figure shows the algorithm for the `csColor_mix` state.

Figure 7. Flowchart for Slave Node in csColor_Mix State


Run algorithm for Active state
Is command ID == 0x40?
Send the Node State information
Is message received from master?
Is command ID == 0x31 or 0x32 or 0x33 or 0x48
Execute scene information received.
Is command ID == 0x36?
Execute color information received.
Is command ID == 0x44?
Unbind the slave node
Return

This state corresponds to the active state of the slave node.

- Any message received is checked against the address of the master node to which the slave node is bound. If the addresses match, then only the message is processed. The only exception for this is command ID 0x40 (node state information). If this command ID is received, then the slave sends its node state information to the master node that requested it.
- If the received command ID is 0x31, 0x32, 0x33 or 0x48, then the slave node runs the scene information code.
- If the slave node receives the command ID 0x36, then it determines whether the information corresponds to the CIE coordinates or the direct LED control based on the first bit.
- If the information is related to the CIE coordinates, then the slave node runs the color mix algorithm and finds out the maximum dimming values for the CIE coordinates received. Next, the slave multiplies the maximum dimming values with the percentage value received and gets the exact dimming values and controls the LEDs accordingly.
- If the command ID 0x44 (Unbind) is received, then the slave node becomes available (cs_Unconfigured).

Master Project

The master for this application can run on an embedded host (such as the CY8CPLC20 device) or on a PC (with the Cypress RGB Control GUI). This section describes the firmware project developed for the embedded master node. The master for the LED Controller is developed using the CY8CPLC20 device on the CY3274 High Voltage Powerline development board. For more information on the development board, see CY3274 Programmable High Voltage Powerline Communication Development Kit.

When using the RGB GUI, the CY8CPLC10 device, along with an USB-I2C bridge acts as the master node. The CY3274 can also be made to work with the RGB GUI.

For information on how to do this, see Making the PLC Control Panel work with the CY3274, CY3275, CY3276 and CY3277 boards.

For more information on using the GUI, see the PLC RGB Control Application User's Guide. You can download the RGB Control GUI from the same location.

The embedded master node does not have all the features of the GUI. However, the following essential features are implemented:

- Binding
- Unbinding
- Direct LED control
- Node search

The CY3274 board has four LEDs, eight DIP switches, one push button, and a potentiometer. These are used by the master project for following functions:

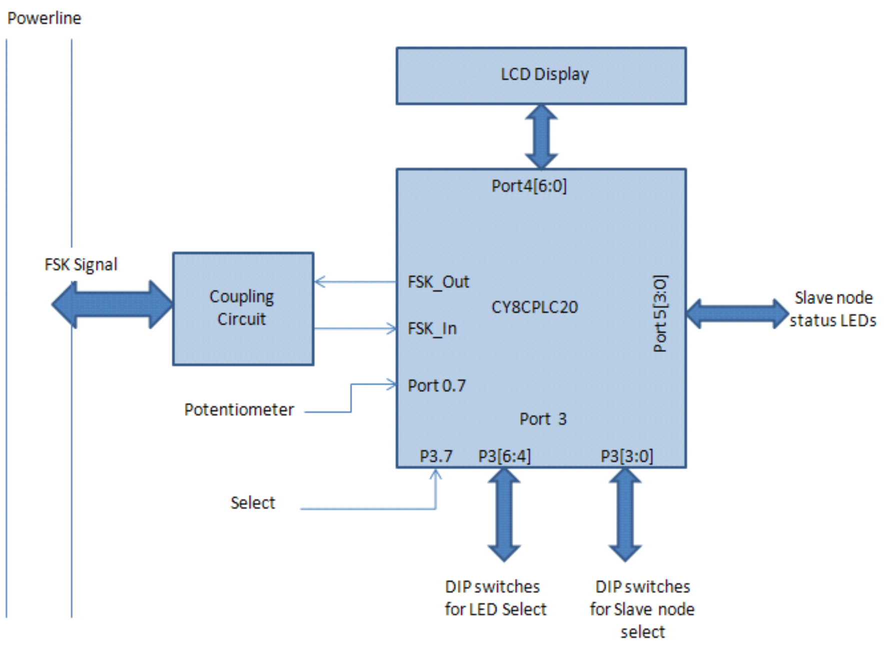
- DIP Switches (1:4) –To select the slave device. The slave selection is one-hot encoded (for example, DIP1 = Node 1, DIP2 = Node 2)
- DIP Switch (5:7) –To select red, green, and blue LEDs.
- Potentiometer – To vary the dimming values of the LEDs on the slave node.
- LED – To display the status of the corresponding slave node.

LED Status	Slave Status
On	Active
Off	Not present
Blinking	Available

- LCD - For user interaction

.

Figure 8. Block Diagram for Master Node

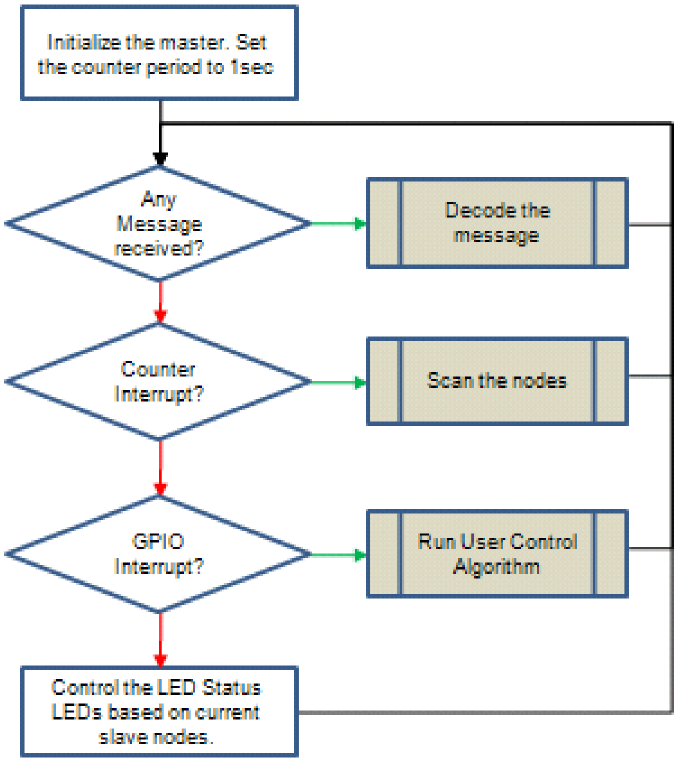


Implementation

The master performs the following functions continuously:

- It waits to receive messages from the slave nodes.
- Every 20 seconds, it polls for all the nodes (active and available). If the node is not found, then the master node removes that node from the database.
- When you press the select button, it runs the user control available to control the active node or to bind the available node.

Figure 9. Flowchart for Master Node


Initialize the master. Set the counter period to 1sec
Any Message received?
Decode the message
Counter Interrupt?
Scan the nodes
GPIO Interrupt?
Run User Control Algorithm
Control the LED Status LEDs based on current slave nodes.

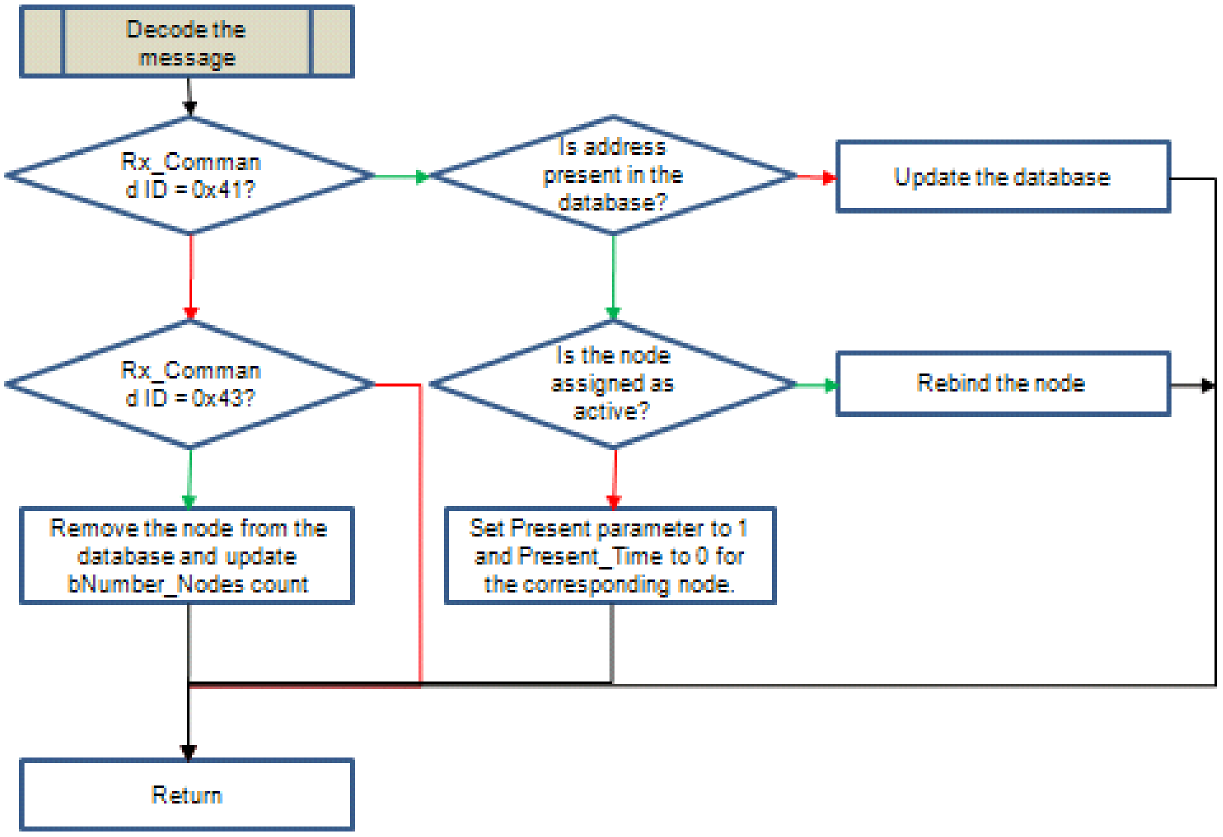
The master has the following structure for each of the four potential nodes:

```
typedef struct{
BYTE PA[8];
BYTE LA;
BYTE State;
BYTE Color[3];
BYTE Enabled;
BYTE Present;
BYTE Present_Time;
BYTE User_Selected;
}PLC_Nodes;
```

The description for the parameters is as follows:

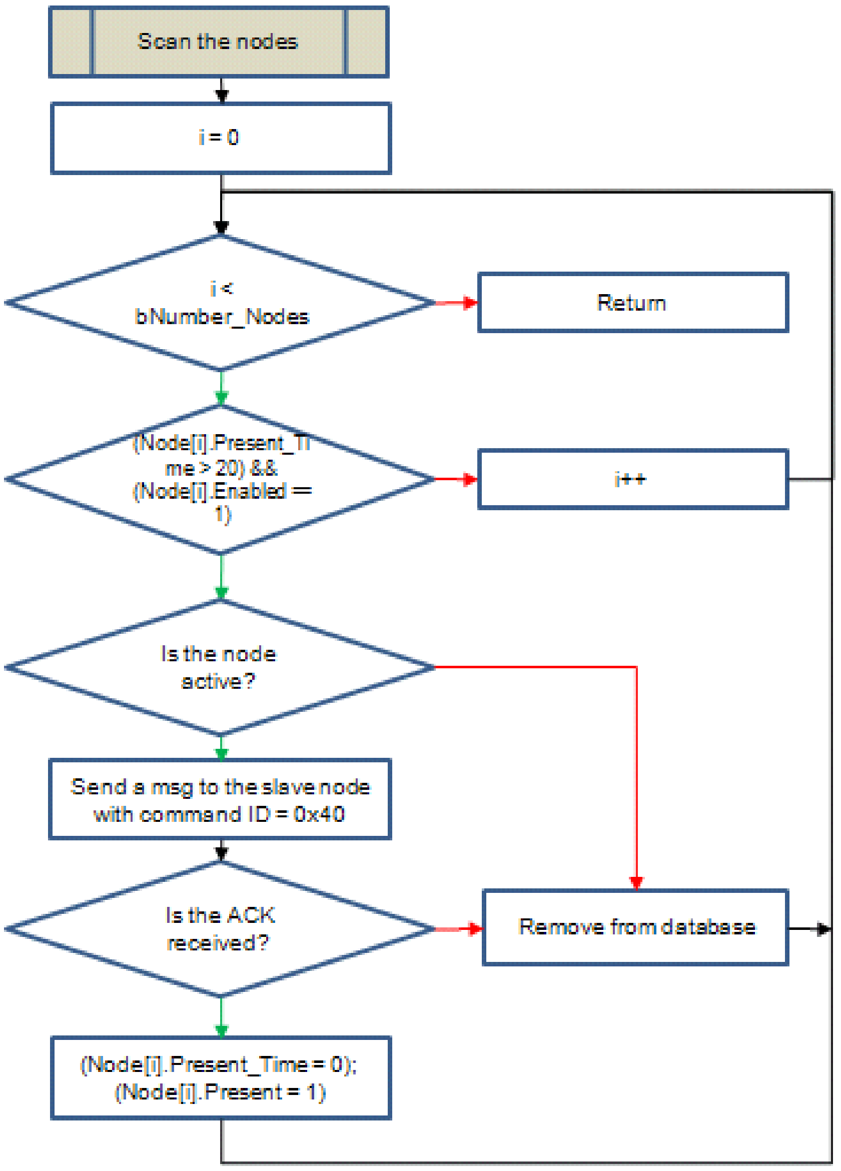
- `PA[8]` – Stores the physical address of the node.
- `LA` – Stores the current logical address of the node.
- State – Stores the state of the node. The possible states and their corresponding values are:
  - enNo_Node = 0x01
  - enAvailable = 0x02
  - enactive = 0x03
- `Color[3]` – This array is used to store the current dimming values for red, green, and blue LEDs.
- `Enabled` – If this parameter is set, then you are allowed to select a particular node. The parameter is set after receiving a first message from the new node on the network and is reset only after the corresponding node is found to be disconnected at the time of polling.
- `Present` – This parameter shows if the node is present on the network. After polling an available node, this parameter for the polled available node is reset. The parameter is set when a broadcast message is received from the corresponding available slave node.
- `Present_Time` – This parameter is incremented every second (in the counter ISR) if the node is enabled (active or available). The master polls for the node if its Present_Time is equal to 20.
- `User_Selected` – This parameter is used to determine which node is selected by the user.

Figure 10. Flowchart for Master Node for Decoding Message


Decode the message
Rx_Command ID = 0x41?
Is address present in the database?
Update the database
Rx_Command ID = 0x43?
Is the node assigned as active?
Rebind the node
Remove the node from the database and update bNumber_Nodes count
Set Present parameter to 1 and Present_Time to 0 for the corresponding node.
Return

- If the master has received command ID 0x41 (which means that the master has received a message from an available node), it checks the physical address of the node in its database.
- If the address is not found, then the master adds the new node to the database. Otherwise, the parameter `Present_Time` is set to '0' and variable `Present` to '1' for that particular node. The master also checks whether the node is active in the database. If the node is active, then the master rebinds the node by setting the logical address of that particular node to the one stored in its database.
- If the master receives command ID 0x43 (Unbind) from the slave, then the master removes the corresponding slave node from its database.

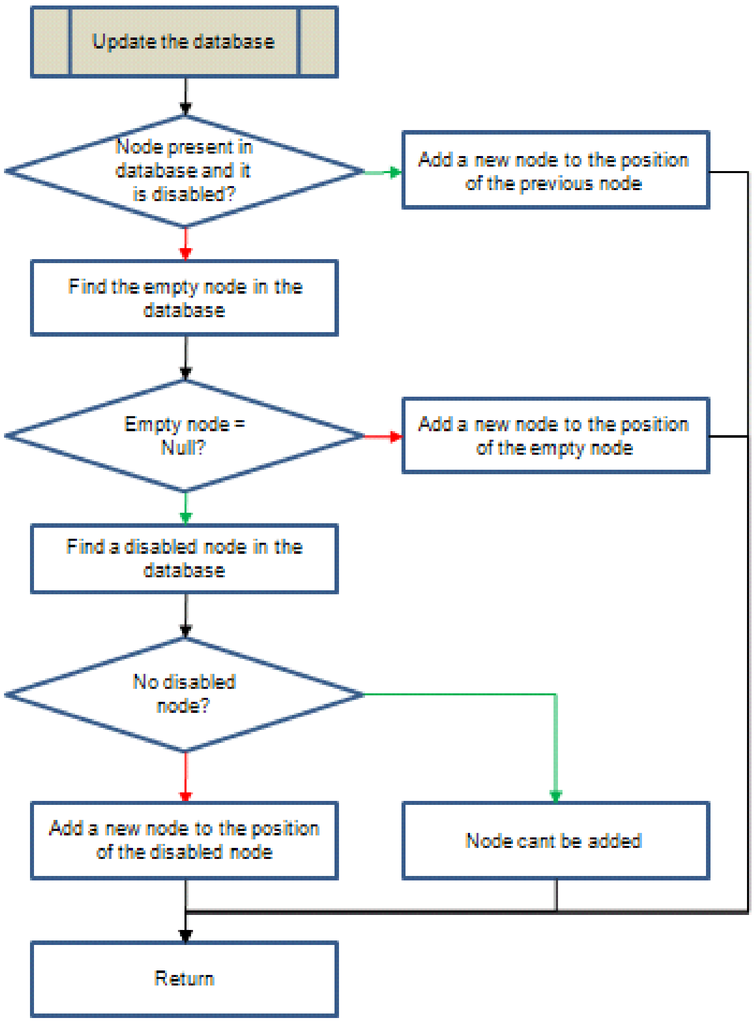
Figure 11. Flowchart for Master Node for Scanning Nodes


Scan the nodes
i = 0
i < bNumber_Nodes
Return
(Node[i].Present_Time > 20) && (Node[i].Enabled == 1)
i++
Is the node active?
Send a msg to the slave node with command ID = 0x40
Is the ACK received?
Remove from database
(Node[i].Present_Time = 0); (Node[i].Present = 1)

- The master node keeps track of all the nodes. At every second, it checks for the nodes whose Present_Time is greater than 20. This means that there is no activity between the master and the slave for 20 seconds.
- If the node is available, the master simply removes the node from the database. An available node is supposed to broadcast its physical address every 2 seconds. Therefore, no activity for 20 seconds means that the available node is no longer present on the network

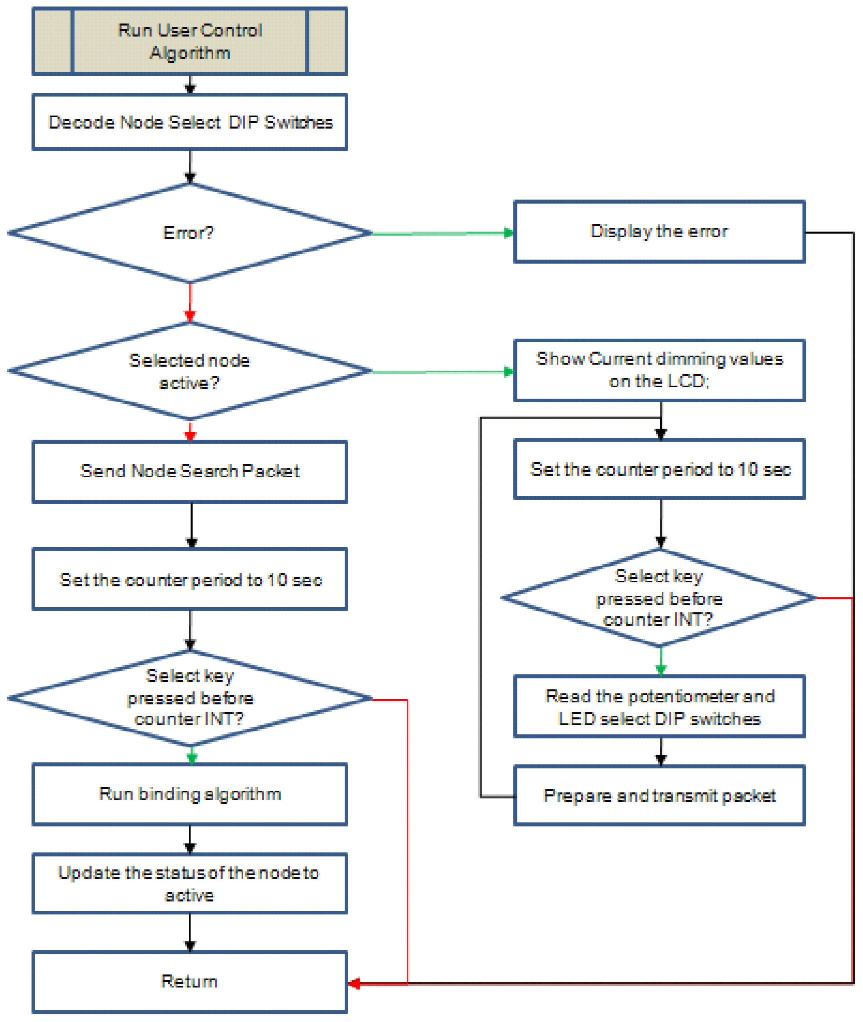
  However, if the node is active, the master sends a packet to that node with command ID = 0x40. If the master receives an acknowledgement, it confirms that the active slave node is present and therefore, the master resets the Present_Time parameter for the node. If the acknowledgement is not received, the master discards the node from the database.

Figure 12. Flowchart for Master Node for Adding a Node


Update the database
Node present in database and it is disabled?
Add a new node to the position of the previous node
Find the empty node in the database
Empty node = Null?
Add a new node to the position of the empty node
Find a disabled node in the database
No disabled node?
Add a new node to the position of the disabled node
Node cant be added
Return

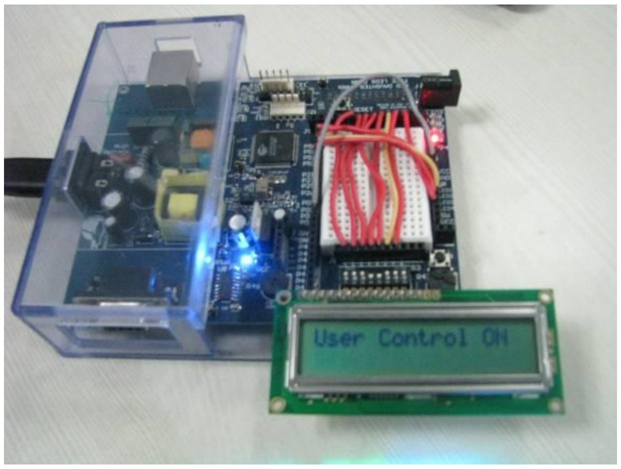
- The master searches the node (by its physical address) in its database and finds its offset (The offset is between 0 and 3). If the node is found and is disabled, then the master adds the new node to the offset found. Therefore, if the node is temporarily disconnected from the network, after reconnection, it is available at the same offset as before.
- However, if the node is not found in the database, then the master node adds the new node to the node that is empty. An empty node means that there is no node added to the corresponding offset.
- If there is no empty node, then the master adds the new node to the disabled node with least offset. If no node is disabled, it means that all the four networks are enabled and there is no space to add the new node. The master shows an error message on the LCD.

Figure 13. Flowchart for Master Node for User Control Algorithm


Run User Control Algorithm
Decode Node Select DIP Switches
Error?
Display the error
Selected node active?
Show Current dimming values on the LCD;
Send Node Search Packet
Set the counter period to 10 sec
Set the counter period to 10 sec
Select key pressed before counter INT?
Select key pressed before counter INT?
Read the potentiometer and LED select DIP switches
Run binding algorithm
Prepare and transmit packet
Update the status of the node to active
Return

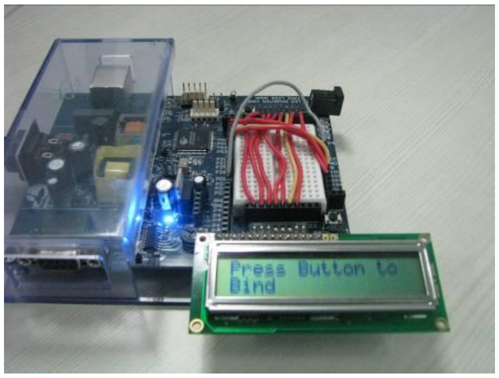
- When “User Control ON” appears on the LCD as shown in Figure 14, you can adjust the slave select DIP switches and press the button to control the selected nodes.

Figure 14. Master Node when User Control is ON



- When the select button is pressed, based on the position of the slave select DIP switches, the master decides whether an error has occurred.
- The error occurs in one or more of the following cases:
  - ❑ More than two available nodes are selected.
  - ❑ More than two active nodes are selected.
  - ❑ One available and one or more active nodes are selected.
  - ❑ No node is selected.
- If no error has occurred, then the master checks the status of the selected node.
  - ❑ If the selected node is available, then the master asks whether to bind the selected slave device; at the same time, the master node sends the “Node Search” command so that the white light flashes on the slave node. A message appears on the LCD stating “Press Button to Bind” as shown in Figure 15.
  - ❑ To bind the slave device, press the switch again; the master runs the binding algorithm and updates the selected slave node’s status to active. In the binding algorithm, the master checks whether there is a node present with the next logical address by sending a test packet. If an acknowledgement is received, the master runs the same algorithm; otherwise, the master changes the address of the new slave to the available logical address.
  - ❑ If the selected node is active (using slave select DIP switches), the master sends the current dimming values. Users get 10 seconds to set new dimming values. When there is a change in potentiometer position (if the current ADC value changes by more than 5) and/or there is a change in LED Select DIP switches, then the master sends a packet with the dimming values to the particular slave. After sending a packet, the master resets the counter and allows 10 more seconds to control the slave node.

Figure 15. Master Node when Binding



- During this 10 second interval, line 0 of the LCD displays message ‘Unbind:Press Btn’ and line 1 displays ‘LED Dim:Vary Pot’. To unbind the selected slave node, the button has to be pressed again during this 10-second interval
- After controlling the slave node, if nothing is done for 10 seconds, the master exits the “User Control” algorithm.

Evaluating the Example Projects on Hardware

The slave in this example is designed to run on the CY3274 High Voltage PLC. The master is designed to work on CY3274 High Voltage PLC DVK.

On the master node, make the following connections on the board:

1. Connect slave select switches – Using wires provided with the kit, connect DIP switch [1:4] to P3[0:3]. DIP switch 1 is connected to P3[0] and DIP switch 4 is connected to P3[3].
2. Connect LED select switches – Using wires provided with the kit, connect DIP switch [5:7] to P3[4:6]. DIP switch 5 is connected to P3[4] and DIP switch 7 is connected to P3[6].
3. Connect potentiometer – Connect “VR” to P0[7].
4. Connect select switch – Connect “SW” to P3[7].
5. Connect slave status LEDs – Connect P5[3:0] to LED[1:4] on the board. P5[3] is connected to LED4 and P5[0] is connected to LED1.
6. Connect LCD – Insert the LCD provided with the kit in the LCD header on the board.

On the slave node, make the following connections:

1. Connect Unbind switch – Using a wire provided with the kit, connect “SW” to P0[1].
2. Connect the LED Driving Pins to onboard LEDs. Connect P5[0:3] to LED1-LED4.

Figure 16. Master Node

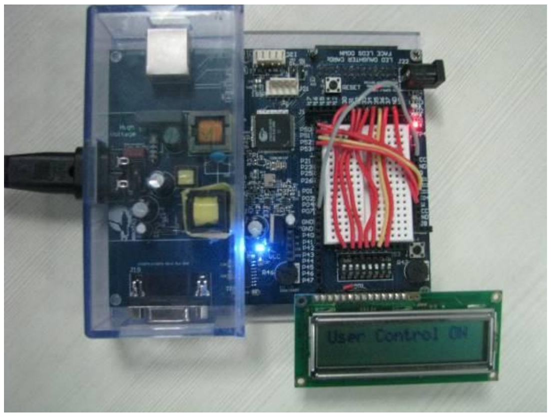
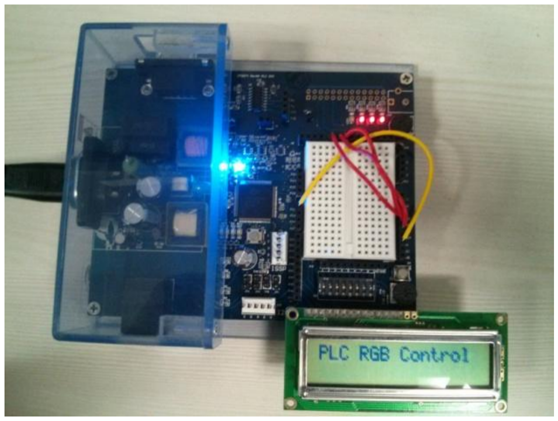


Figure 17. Slave Node



About the Author

Name:	Rohan Gandhi
Title:	Applications Engineer

Appendix A: Packet Structures

Node Discovery

Binding to the Master

When available, the slave transmits a packet with the physical address in its payload for the master nodes to notice that the slave is “Available”. The following table summarizes the transmit parameters for this packet.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Group	Logical	0x01*	0x41	8	PA of the slave

*Every master on the network has group address of 0x01.

On receiving this packet, the master displays the slave in an available node.

When Bound to the Master

When the slave becomes “Active”, the master node sends a packet with the following transmit parameters to bind to slave node.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11)
Physical	Logical	8-byte physical address of the Slave	0x04	1	non-zero logical address

Also, the master transmits a packet with the following transmit parameters for the other masters on the network to remove the selected node from their available node’s list.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID(Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Group	Logical	0x01	0x42	8	8-byte physical address of selected slave

After receiving the packet from the master, slave node sends information about the CIE coordinates for red, green, and blue LEDs to the master. The following table summarizes the transmit parameters.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Logical	Logical	Of the master	0x34	12 or 16*	Table 3

Packet lengths vary depending on the number of LEDs the slave can control: 12 bytes for three LEDs. If the slave node controls four LEDs, the packet length becomes 16 bytes.

Table 3. Packet Structure for the CIE Coordinates

Byte Offset	Content
1	Maximum possible x coordinate value for RED LED
2	
3	Maximum possible y coordinate value for RED LED
4	
5	Maximum possible x coordinate value for GREEN LED
6	
7	Maximum possible y coordinate value for GREEN LED
8	
9	Maximum possible x coordinate value for BLUE LED
10	

Byte Offset	Content
11	Maximum possible y coordinate value for BLUE LED
12	

This completes the “Binding to the Master” function of the slave. The master now controls the RGB colors of the slave. When the packet with RGB CIE coordinates from the slave node does not reach the master node, the master can also request the CIE coordinates. For this, the master sends a packet with the following transmit parameters.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Logical	Logical	Address of the slave node	0x34	0	None

*Logical address of the slave node.

RGB Control

The master has three ways to control the R, G, and B dimming values of the slave:

- CIE coordinates
- Direct LED control
- Color temperature (slave receives a packet with CIE coordinates)

The slave has a structure consisting of information about the current x, y, and Y position and dimming values for each of the red, green, and blue colors. If the slave receives the packet with CIE coordinates, it runs the color mix algorithm to calculate corresponding red, green, and blue dimming values.

CIE Coordinates

When any point inside the color gamut is clicked, the master sends the packet with x, y coordinates and percentage of the intensity (0 to 100) to the slave. The slave first calculates the dimming values corresponding to the color and maximum intensity by running a color-mix algorithm. When it gets the maximum dimming values, it multiplies these values with the percentage value received to get exact dimming values for RGB LEDs. Because of the 8-bit calculations, the LEDs sometimes get saturated at 98-99%.

The master sends a packet with following payload and command ID:

Command ID = 0x36.

Payload:

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	x	x	x	x	x	x	0*	1**
1	x coordinate							
2								
3	y coordinate							
4								
5	Percentage value Y							

*Bit 1 – This bit in the control byte (Byte Offset 0) determines whether the payload contains the direct LED control or CIE coordinates information.

**Bit 0 – This bit in the control byte (Byte Offset 0) is the On/Off bit. Resetting this bit switches off all LEDs on the slave node, irrespective of remaining payload.

Direct LED Control

In this method, the master directly sends the dimming values for R, G, and B colors.

The master sends a packet with following payload for Direct LED control (The command ID remains to be 0x36).

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	X	x	x	x	X	x	1	1
1	Dimming value for the red LED							
2	Dimming value for the green LED							
3	Dimming value for the blue LED							

Node Search

When there are multiple available slave nodes, it becomes difficult to see which button in the available node’s frame corresponds to actual slave node. To help identify the node, a white light flashes on the slave when its button is clicked in the RGB GUI.

When the slave node button (in the available node’s frame) is clicked, the master sends a packet with the following transmit parameters.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Physical	Logical	Address of the slave node	0x45	0	None

Node State Information

Any master on the network can get the information from the slave node. This functionality helps masters to poll slaves when the packets keep dropping on the Powerline or the slave node is powered off. To get the node information, the master sends the packet with the following transmit parameters.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
*	Logical	Address of the slave node.	0x40	0	None

* The Tx DA Type is logical if the slave node is active, otherwise, the DA type is physical.

\If the slave is “Available” (not bound to any master), it sends the packet with the following payload.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Physical	Logical	Address of the master node.	0x39	1	0x00

But, if the slave is “Active” (bound to one of the masters), the following payload is sent back to the master.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Logical	Logical	Address of the master node.	0x39	2	Payload[0] = 0x01; Payload[1] = LA of the master to which the slave is bound to

Unbinding

When the node is bound the master, it can be unbound in two ways

By the master (move the node from “Active” to “Available” space): To disconnect the slave, the master sends a packet with the following parameters.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Logical	Logical	Address of the slave node	0x44	0	None

When the slave disconnects from the master, it sends the following packet to the master.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11:0x19)
Logical	Logical	Of the master	0x43	0	None

After the master disconnects the slave node, it has no control over the R, G, and B of the slave. The slave node gets logical address = 0x00 and starts broadcasting its physical address every two seconds.

Save Scene

With this function, the slave node saves the RGB dimming values at the scene number received in the packet. There are eight scenes available. The master sends a packet with following parameters.

Tx Command ID – 0x31

Payload

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	x	x	x	x	x	Scene Number		

On receiving this packet, the slave node stores the current dimming values for the RGB LEDs at the scene number specified.

Load Scene

With this function the stored scene is loaded; this means that the stored dimming values are loaded on the LEDs. The color-mixing algorithm is not run because the dimming values are already known.

Tx Command ID – 0x32

Payload

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Fade Rate		x	x	x	Scene Number		

Bit 7	Bit 6	Fade Rate
1	1	2 Seconds
1	0	5 Seconds
0	1	10 Seconds
0	0	0 Seconds (No fading)

If the fade rate is non-zero, the current color is faded until the duration specified (by the fade rate) and then the scene is loaded. The fading is implemented using Counter16 ISR, which is called 20 times every second.

Reset Scene

The slave node changes the stored dimming values for the R, G, and B colors for the scene numbers specified by the master to 0x00. The master sends a packet with following payload.

Tx Command ID – 0x33

Tx Payload

Byte Offset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Scene 7	Scene 6	Scene 5	Scene 4	Scene 3	Scene 2	Scene 1	Scene 0

If the bit is set to '1', then that particular scene is reset.

Save Configuration

This option allows you to set default scenes. To save the configuration, save six scenes (Scenes 1 to 6) for one active node. (If there are two or more active nodes, the configuration is saved based on the scenes for the first active node). When you click 'Save Configuration' or 'Save Configuration As', the GUI saves a text file with the current scene information. Scenes 7 and 8 are hardcoded in the slave nodes.

This way, you can store the scene information, which can be loaded at any time. The text is stored depending on whether the scene is selected through CIE/temperature control or direct LED control. For CIE/temperature control, the GUI writes '1' to the text file followed by five lines of data. For direct LED control, the GUI writes '2' and the next three lines contain the dimming values for RGB LEDs. This way, the GUI stores all eight scenes in order in the same file. This option is available only with RGB Color Control GUI.

CIE Coordinates/ Temperature Control	Direct LED control
'1'	'2'
MSB of x coordinate	Dimming value for Red LED
LSB of x coordinate	Dimming value for Green LED
MSB of y coordinate	Dimming value for Blue LED
LSB of y coordinate	
Percentage of the maximum intensity	

Load Configuration

When you click 'Load Configuration', the scene information is sent out to all active nodes. The first byte in the payload is used as the control byte. In the last six bits (bit 5:0), every bit specifies whether the scene information is according to CIE/temperature control coordinates or according to the direct LED control. This option is available only with the RGB Color Control GUI.

The transmit parameters for this packet are as follows.

Tx DA Type (Offset 0x07)	Tx SA Type (Offset 0x07)	Tx DA (Offset 0x08)	Tx Command ID (Offset 0x10)	Tx Data Length (Offset 0x06)	Tx Data (Offset 0x11)
Group	Logical	0x00	0x48	**	*

*Contains the scene information.

**The length varies depending on whether the scene information is specified according to the CIE/temperature control or according to the direct LED control. The minimum length is 19 bytes (1 control byte + 18 bytes for direct LED control). The maximum size is 31 bytes (1 control byte + 30 bytes for CIE/temperature control).

Document History

Document Title: LED Lighting Control Using Powerline Communication - AN58717

Document Number: 001-58717

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2871859	ROSG	02/26/2010	New application note.
*A	3198479	MKKU	03/16/2011	Projects updated to PD5.1. File Headers added for Main.C in both the Master and Slave projects. LCD display modified for both the Master and Slave projects. Issue of low LED brightness while using with the GUI fixed in the Slave project. BIU, TX, RX LEDs added for both the Master and Slave projects. More description added for using the RGB control panel GUI. ISR calls moved from boot.tpl to user module INT.asm files. 'Transmit_Packet' function added in the Slave project. The RGb control panel GUI setup and users guide added to the attachments.
*B	4370128	ROIT	05/05/2014	CY3272, CY3273, CY3275 and CY3276 are obsolete. All projects and hardware ported to CY3274 kit. Also, no EZ-Color DVK used. Onboard LEDs on CY3274 kit used to demonstrate the Dimming and Switching of the three LEDs (emulating RGB). Projects updated to Designer 5.4 Updated to new document template.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

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