Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

AN62487

Cypress Powerline Communication (PLC) Repeater Implementation

Author: Aditya Yadav
Associated Project: Yes
Associated Part Family: CY8CPLC20
Software Version: PSoC Designer™ 5.1 SP1 or later
Related Application Notes: AN54416

If you have a question, or need help with this application note, contact the author at adiy@cypress.com.

Abstract

AN62487 explains Cypress's Powerline Communication (PLC) repeater algorithm and the implementation on the CY8CPLC20 device. The attached code example can be programmed to run on the PLC development kits.

Contents

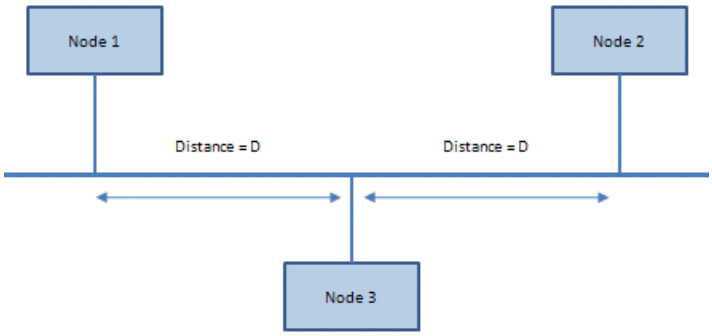
Introduction

Current PLC implementations are limited by the distance between nodes and the loading on the network. If the destination node is out of range from the source node, then communication is not possible between the two nodes.

Cypress has developed a repeater algorithm that can overcome this limitation. It is now possible to reach any node, provided there is at least one node present in range of every other node. In addition, a node that functions as a repeater can also run an application. The Cypress CY8CPLC20 device can be programmed with the PLC repeater algorithm, as well as the user's application (e.g. LED control, power measurement). This application describes the repeater algorithm and how to evaluate it on the CY3274 high voltage PLC or CY3275 low voltage PLC development kits.

For an introduction to using Cypress PLC devices, refer to application note "Using CY8CPLC20 in Powerline Communication (PLC) Applications - AN54416".

Figure 1. Use of the Repeater

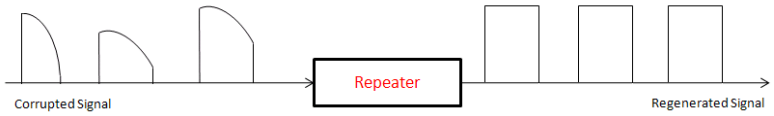


As shown in Figure 1, Node 1 is not able to communicate with Node 2 directly because the powerline signal is attenuated below the Node 2 receive threshold. But, when Node 3 is present, Node 1 can communicate with Node 2 through Node 3. Node 3 acts as a repeater and passes the packet from Node 1 to Node 2.

Traditionally, devices that help communication between nodes located far apart are broadly divided into the following categories.

1. **Repeater** – These devices operate at the physical layer and only amplify the signal. The following figure shows the operation.
2. **Bridge** – Bridges operate at the physical and at data link layer. Bridges pass the packet from one frame to the other depending on the address (logical / MAC) in the packet.
3. **Router** – Routers operate at the network layer and use routing tables for making decisions about the route between the source and the destination node.

Figure 2. Typical Repeater Operation



.

Cypress Repeater

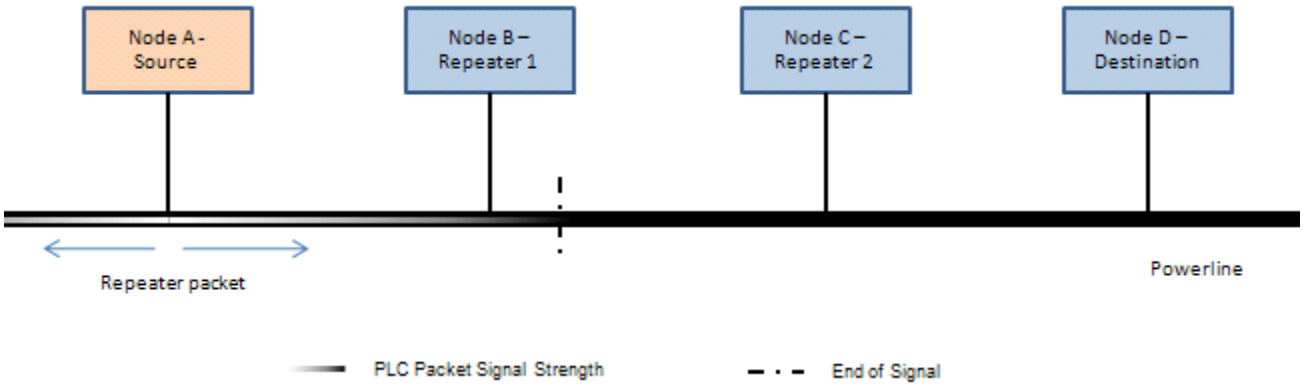
The traditional solutions mentioned in the previous section physically disconnect the communication medium. The repeater developed in this application note, however, does not disconnect the powerlines and enables every node to act as a repeater, eliminating the need for any separate devices. The algorithm operates at the network layer but unlike the router, it requires no routing table and operates as a connectionless service.

The algorithm is developed to repeat the packets reliably without affecting the throughput significantly. The following section explains how the repeater functions.

Stage 1

Consider that Node A (Source) wants to communicate with Node D (Destination). Due to the signal attenuation, the packet from node A is not acknowledged by node D and therefore Node A broadcasts the same packet in repeater mode.

Figure 3. Repeater Stage 1

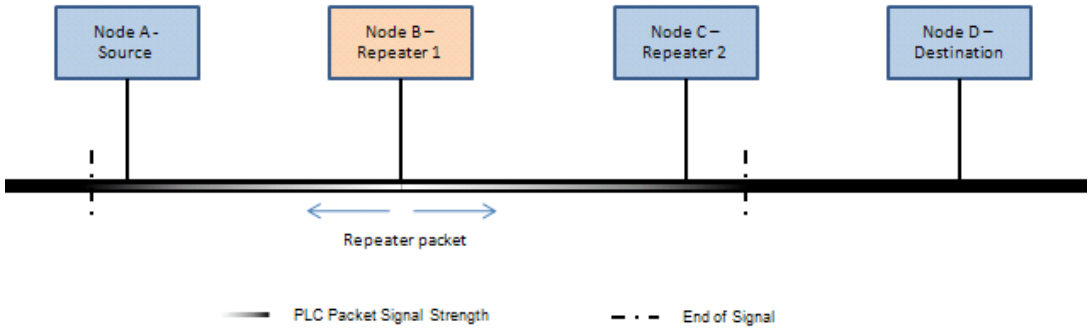


Stage 2

The broadcasted packet is received by Node B (Repeater 1), which runs the repeater algorithm and repeats the same packet.

Repeating a packet is a two-stage process where in the first stage, the node transmits the packet directly to the destination (without using the repeater feature). If an acknowledgement is not received, then the node broadcasts the packet in the repeater mode. In this case, as the destination node is not visible to Node B, it does not get an acknowledgement back and therefore Node B rebroadcasts the repeater packet.

Figure 4. Repeater Stage 2

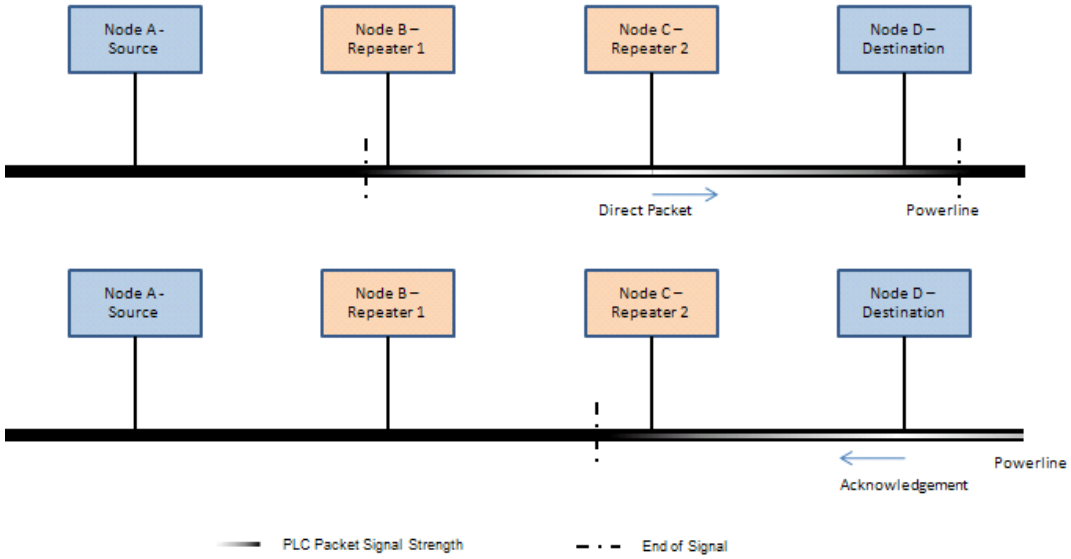


The retransmitted packet is received by Node A and Node C. For Node A, the packet from Node B is the same as the packet it had broadcasted in stage 1 and therefore it considers that the packet has propagated ahead. This packet from node B, therefore, acts as a confirmation to node A. At this stage, node A stops attempting to re-transmit this packet. Now, it can make another packet and transmit.

Stage 3

A packet transmitted by Node B is received by Node C and is retransmitted. As a part of the repeater algorithm, Node C transmits a direct packet to the destination node. Since the destination is visible to node C, node C receives an acknowledgement when it transmits a packet in the direct mode. The acknowledgement acts as a confirmation for node C.

Figure 5. Repeater Stage 3



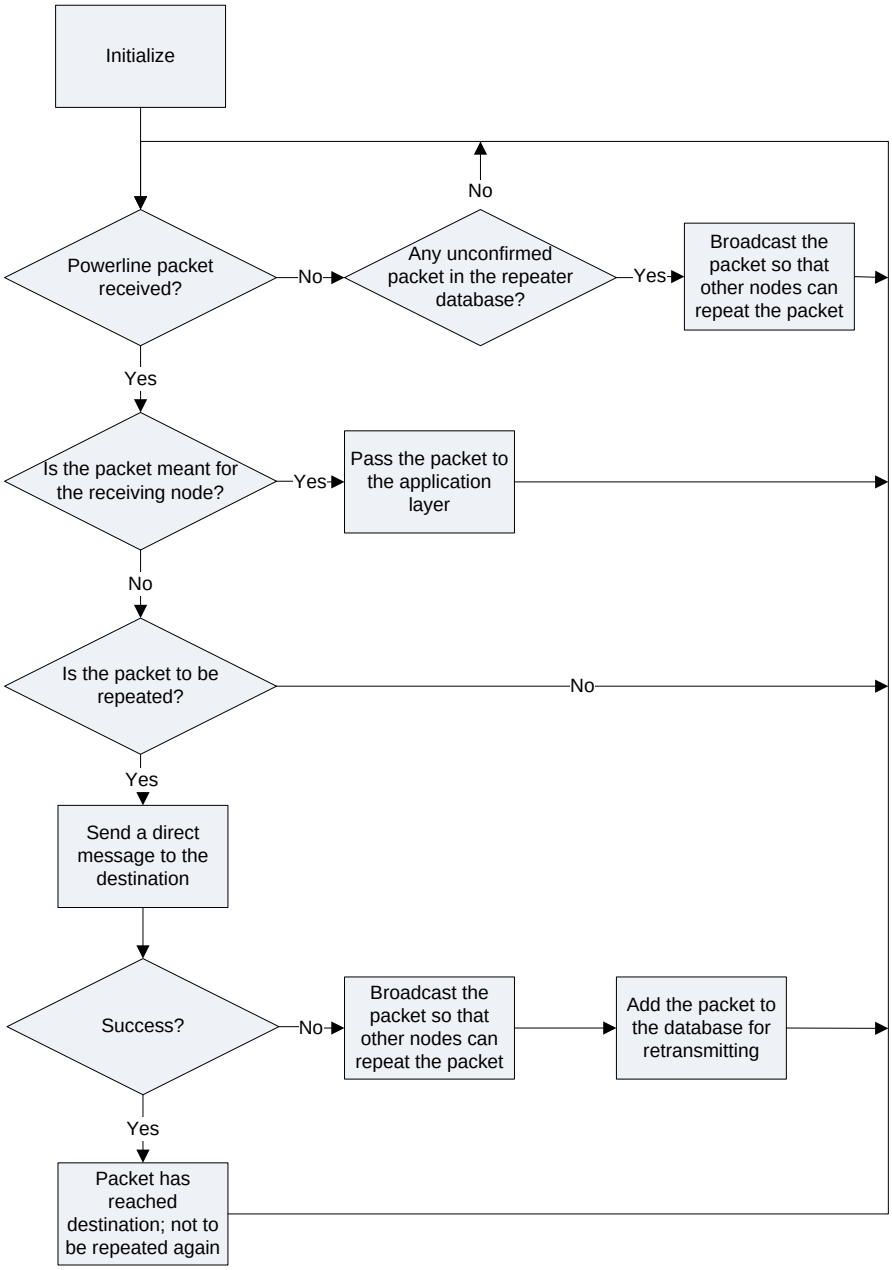
Stage 4

At this stage, the packet is received by the destination. Node A and Node C have the confirmation for the same packet. However, node B has not yet received the confirmation since communication between node C and node D is direct (node to node). Therefore, node B retransmits the packet, which is again received by node A and node C. Node C sends the confirmation to Node B. This way, the packet is received by the destination and all the nodes have the confirmation.

Thus, the repeater nodes transmit packets until a confirmation (retransmission from a different node or acknowledgement) is received, which adds to the robustness. Furthermore, the source node can start transmitting another packet when it has the confirmation, which introduces pipelining of packets and increases throughput.

The following figure explains the system level flowchart for the repeater algorithm.

Figure 6. Flowchart



The algorithm performs two operations:

- Process the received packet.
- Repeat pending packets.

After a packet is received, the node checks if the address in the packet matches with its own address. If there is a match, the packet is then sent to the application layer. In cases when the address does not match and the repeater mode is enabled, the node then repeats the packet.

Periodically, the node also checks if it has any pending packets to be repeated. If so, the node then broadcasts those packets.

Transmitting the Repeater Packet on the Powerline

This solution uses Cypress's proprietary network protocol to transmit the repeater packet. The following figure shows Cypress's PLT packet structure.

Figure 7. PLT Packet Structure

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0x00	SA Type	DA Type		Service Type	RSVD		Response	RSVD
0x01	Destination Address (8-bit Logical, 16-bit Extended Logical or 64-bit Physical)							
0x02	Source Address (8-bit Logical, 16-bit Extended Logical or 64-bit Physical)							
0x03	Command							
0x04	RSVD			Payload Length				
0x05	Seq Num				Powerline Packet Header CRC			
0x06	Payload (0 to 31 Bytes)							
	Powerline Transceiver Packet CRC							

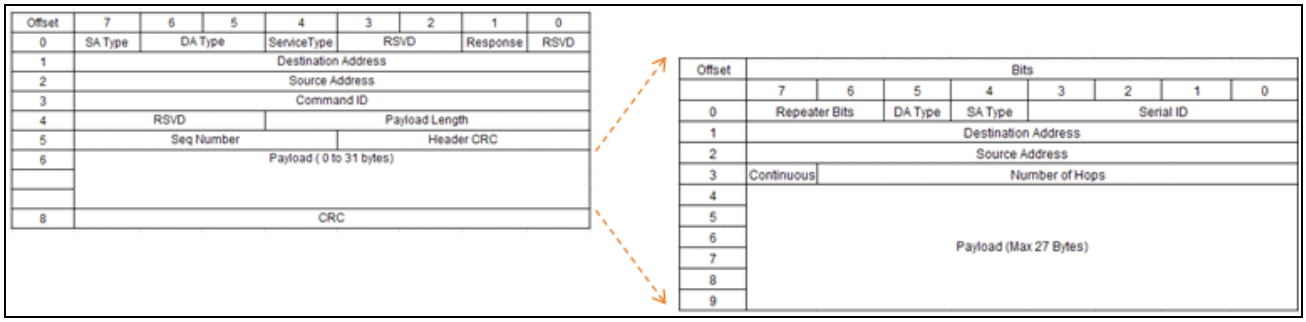
For information on all the fields, refer to the PLT User Module datasheet available with the latest PSoC Designer release or at www.cypress.com .

Repeater packets are of two types.

- **Direct Message** – This message is received by all the nodes but is passed to the application layer only by the node whose address matches the destination address in the PLT packet. The destination address can be either logical or physical (not group).
- **Broadcast Message** – This message is received by all the nodes and they pass the message to the application layer. The destination address type is group and the destination address is 0x00.

As it is not possible to modify the PLT packet structure, the repeater packet uses the payload (0 to 31 bytes) of the PLT packet as shown in the following figure.

Figure 8. Repeater Packet Structure Inside PLT Packet Structure



Repeater Packet Structure

In the repeater algorithm, the following table specifies the overall packet structure. The first two bits of the payload are used to specify the repeater mode. The first four bytes are other required parameters as shown in Table 1.

Table 1. Repeater Packet Structure

Offset	Bits							
	7	6	5	4	3	2	1	0
0	Repeater Bits		DA Type	SA Type	Serial ID			
1	Destination Address							
2	Source Address							
3	Continuous	Number of Hops						
4	Payload (Max 27 Bytes)							
5								
6								
7								
8								
9								

Repeater Modes

There are four modes

- Mode 1 (Repeater Bits = 00)
  - When final destination node is in range of source node (visible)

    The source node will transmit the packet in acknowledgement mode. If it receives the acknowledgement, it means the destination address is in its range and the packet is successfully delivered.
- Mode 2 (Repeater Bits 01)
  - The final destination node is not in range of the repeater and source node

    The source node will broadcast a message that is picked up by the repeater node(s).The broadcasted packet will have repeater bits = 01 stating the other nodes to repeat the packet.
- Mode 3 (Repeater Bits = 10)
  - When the final destination node is out of range of the source but is in range of repeater node

    The repeater node will transmit the packet in acknowledgement service type mode. If it receives the acknowledgement, it means the destination address is in the repeater node's range. The direct message sent by the repeating node will have repeating bits 10 so that the destination node understands that the message has been received through another node and not from the source node itself.
- Mode 4 (Repeater Bits = 11)
  - When the final destination node is out of range of source but is in range of repeater node

    When a repeater broadcasts a packet with repeater bits = 01, it doesn't wait for an automatic acknowledgement. Therefore, when the destination node receives a packet through multiple repeaters with repeater bits = 01, it confirms the reception by sending a packet with repeater bits = 11. This is a direct transmission.

Other Parameters

Table 2. Other Parameters in Payload

Serial ID	Fixed by the source that changes with every new message sent. The serial ID increments for every new message. It doesn't change when the same message is re-transmitted.
Source Logical Address	Logical address of the source that initiated the transmission.
Destination Logical Address	Logical address of the final destination Address.
Number of Hops	When a node receives the packet for the first time, the number of hops is decremented and transmitted to the powerline again. When the number of hops is 0, the packet will not be transmitted.
Continuous	This bit specifies whether the packet is to be repeated by all the nodes or not. If this bit is set, then the packet is repeated by all the nodes irrespective of the number of hops to the powerline.

DA Type, SA Type, Destination Address, Source Address in the repeater packet should not be confused with the fields in the PLT packet.

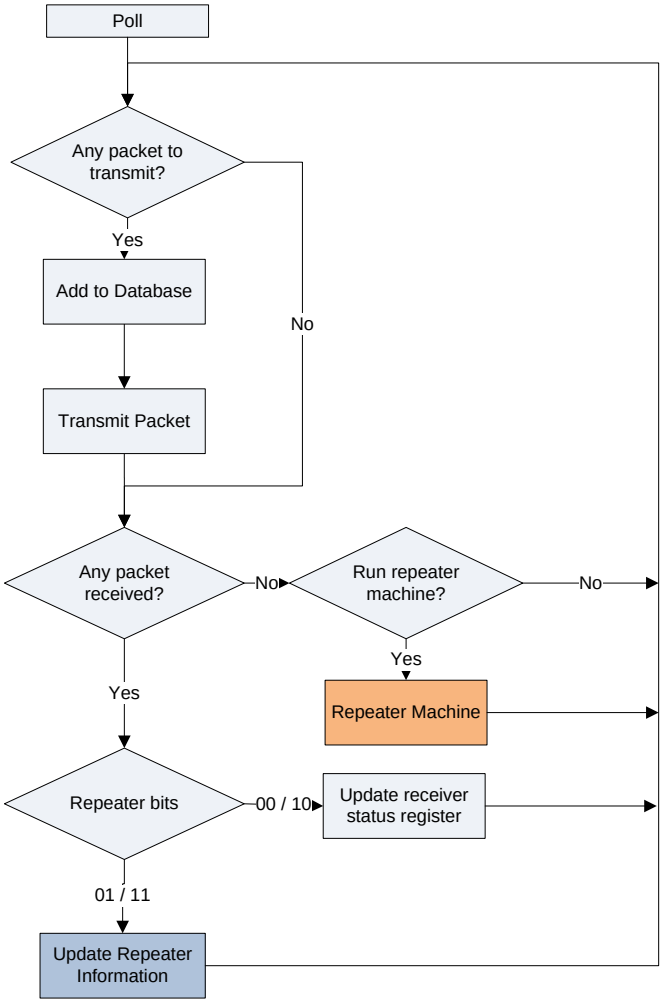
In the repeater packets (mode 2, 3 and 4), the DA Type, SA Type, Source Address and Destination address remain the same. However, these fields are varied in the PLT packet in different repeater modes. The following table summarizes the addressing in the PLT Packet with respect to different repeater modes.

Mode	Repeater Bits		SA Type	DA Type	Source Address	Destination Address
1	0	0	Logical / Physical	Logical / Physical	Address of the source	Address of the destination
2	0	1	Logical	Group	Address of the source	0x00
3	1	0	Logical	Depends on the DA Type in the repeater packet	Address of the repeating node	Final destination address in the repeater packet
4	1	1	Logical	Logical	Address of the destination	Address of the repeating node

Implementation

The following figure shows the overall flowchart for the repeater

Figure 9.- Overall Flowchart for the Repeater



Description

1. The node checks if it has any packet to transmit. In case there is a packet to be transmitted, the node transmits the packet. If the packet is to be transmitted in the repeater mode then it is added to the database.
2. Every packet that is received is checked for the repeater bits.
3. If the repeater bits are 01 (mode 2) or 11 (mode 4), then the node updates the repeater related information. For the packets with repeater bits 00 or 10, the node doesn't do anything and the packet is automatically forwarded to the control panel GUI.
4. The device regularly runs the "Repeater Machine" every 1 second or whenever a packet that requires a response is received over the powerline.

The implementation uses the following structure for every packet:

```
typedef struct{
BYTE Packet[31];
BYTE Command_ID;
BYTE Size;
BYTE Repeater_Count;
BYTE Confirmation;
BYTE Active;
BYTE Tx_Time;
BYTE Hops;
BYTE Host_Request;
BYTE Next_Repeat;
} stPacket_Structure;
```

Description

`Packet [31]` – Stores the packet received over the powerline.

`Command_ID` – Stores the command ID for the packet received.

`Size` –the actual size of the packet. 31 bytes is the maximum size of the packet.

`Repeater_Count` – Keeps track of the number of times a packet has been repeated.

`Confirmation` – Is true if the confirmation is received.

`Active` – Specifies if the packet is to be repeated.

`Tx_Time` – Keeps track of the time for which the packet has been in the buffer.

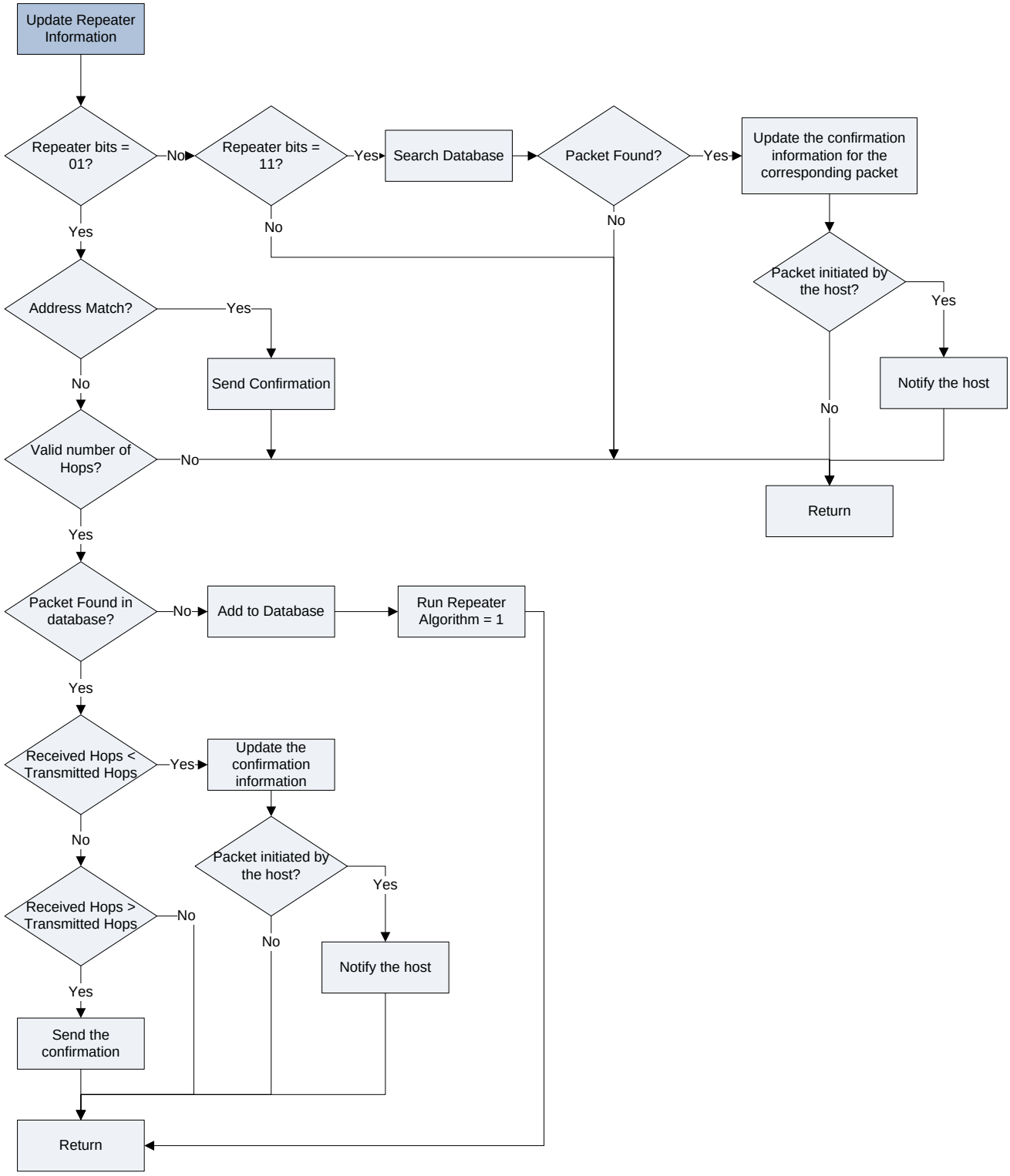
`Hops` – Stores the number of hops used while repeating the packet. This is used later when it receives a packet with a higher hops value, which acts as the confirmation.

`Host_Request` – Specifies whether the corresponding packet was requested by the host application or any other node on the network.

`Next_Repeat` – Specifies the next packet to be repeated. When the node polls to check if there is any repeater packet, it will start from the packet specified by the `Next_Repeat` variable.

The following figure shows the flowchart when the repeater packet is received

Figure 10 - Flowchart for Updating Repeater Information

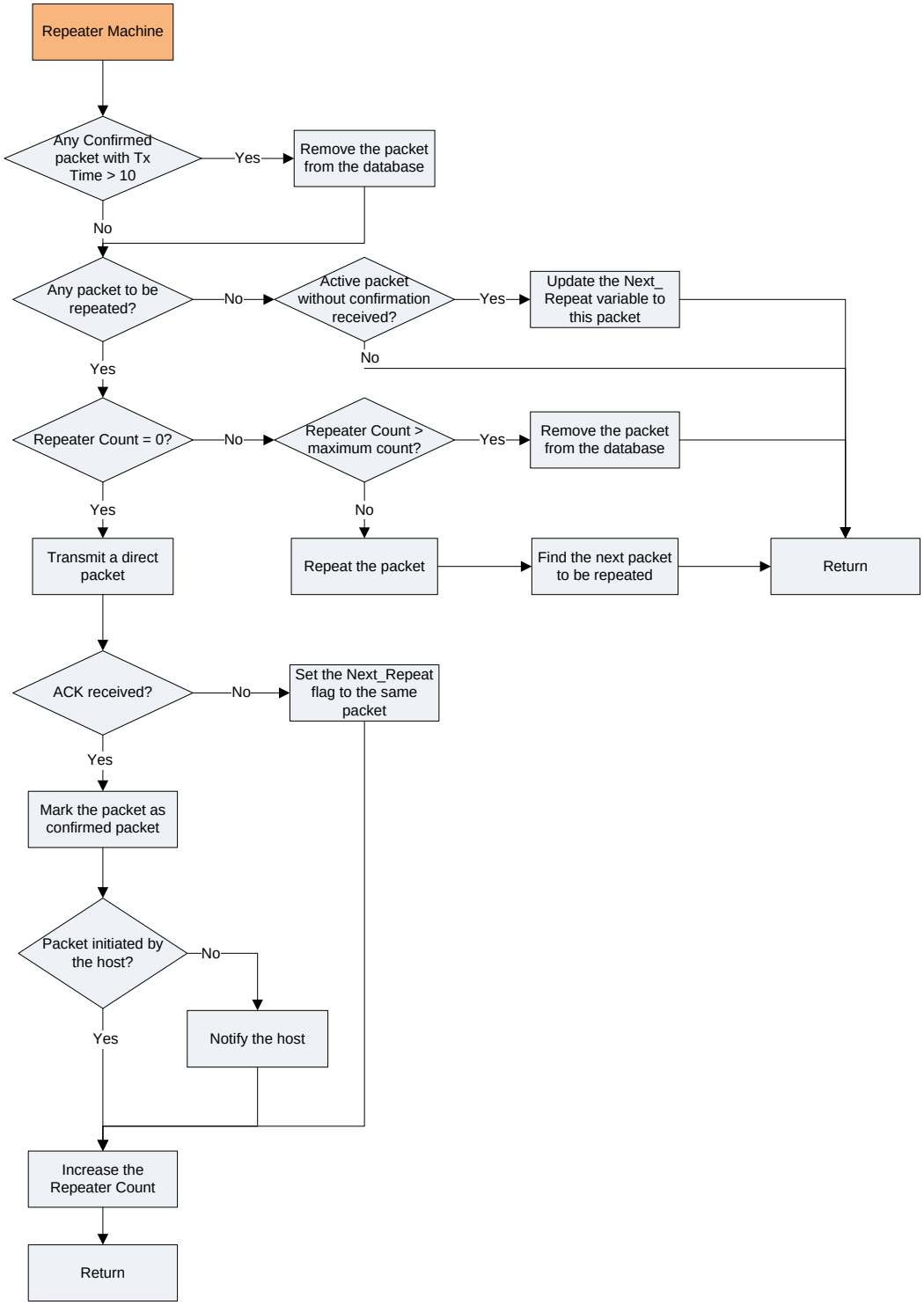


Description

1. “Update Repeater Information”, performs multiple functions related to the received packet based on the repeater bits including, adding the packet to the database, sending the confirmation, updating confirmation information etc.
2. If the repeater bits are 01, the node checks the address of the destination node received in the packet with its own address. If the addresses match, the node sends the confirmation back (with repeater bits = 11) as the packet has reached the destination. The node notifies the host for it to read the packet.
3. In case, if the addresses don’t match, the node looks for the number of hops received in the packet. If the number of hops is valid i.e. the continuous bit in the register is set or the remaining 7 bits are non zero, then the node searches its database for the new packet. In case the number of hops field is found to be 0, then the packet is not repeated. If the packet is found, the node compares the number of hops received with the number of hops stored in the database. If the number of hops received is lower than the number of hops stored, then it means that the packet has been retransmitted by a node further away. In this case, the node marks the corresponding packet as confirmed.
4. If the node finds the received number of hops higher than the stored number of hops, it means that the node closer to the source has retransmitted the packet. Therefore, in this case the node sends a confirmation stating the other node should confirm the packet.
5. If the packet is not found in the database, the node adds the packet to the database and makes the “Run Repeater Algorithm” flag to 1, so that the node runs the “Repeater Machine” immediately.

The following figure shows the flowchart for the repeater machine, which handles the repeater packets

Figure 11 - Flowchart for Repeater Machine



Repeater Machine

1. The repeater machine takes care of transmitting the non-confirmed repeater packets on the powerline.
2. The node checks for any packet in the database for which the confirmation is received and the packet is active for more than 15 seconds. In this case, the packet is removed from the database.
3. The node searches if it has any repeater packet to transmit. The structure for every packet has a variable called Next_Repeat. If the node finds this variable set for any of the packets, then the packet is set to repeat.
4. While repeating the packet, the node first checks for the repeat count for the particular packet. If the repeat count is 0, then the packet is sent as a direct packet with repeater bits = 10. As a direct packet is sent in an acknowledgement mode, the node checks for the result of the transmission.
5. If the packet was acknowledged, it means that the packet has reached the destination node and there is no need to repeat this packet any more. The node at this point also checks if the transmission of the packet was requested by the host. If the request was from the host, then the host is notified with the transmission result as a success.
6. In case the packet is not acknowledged, then the node sets the Next_Repeat variable to true and also sets the Run_Repeater_Machine so that the repeater machine is called immediately.
7. If the repeater count for the packet is nonzero, then the node checks if the repeater count is higher than the maximum (maximum count is 5). If the count is higher than 5, then the packet is discarded from the database.

Implementation

For this implementation, the CY8CPLC20 part is selected. This code example is designed to run on the CY3274 High Voltage PLC DVK or the CY3275 Low Voltage PLC DVK.

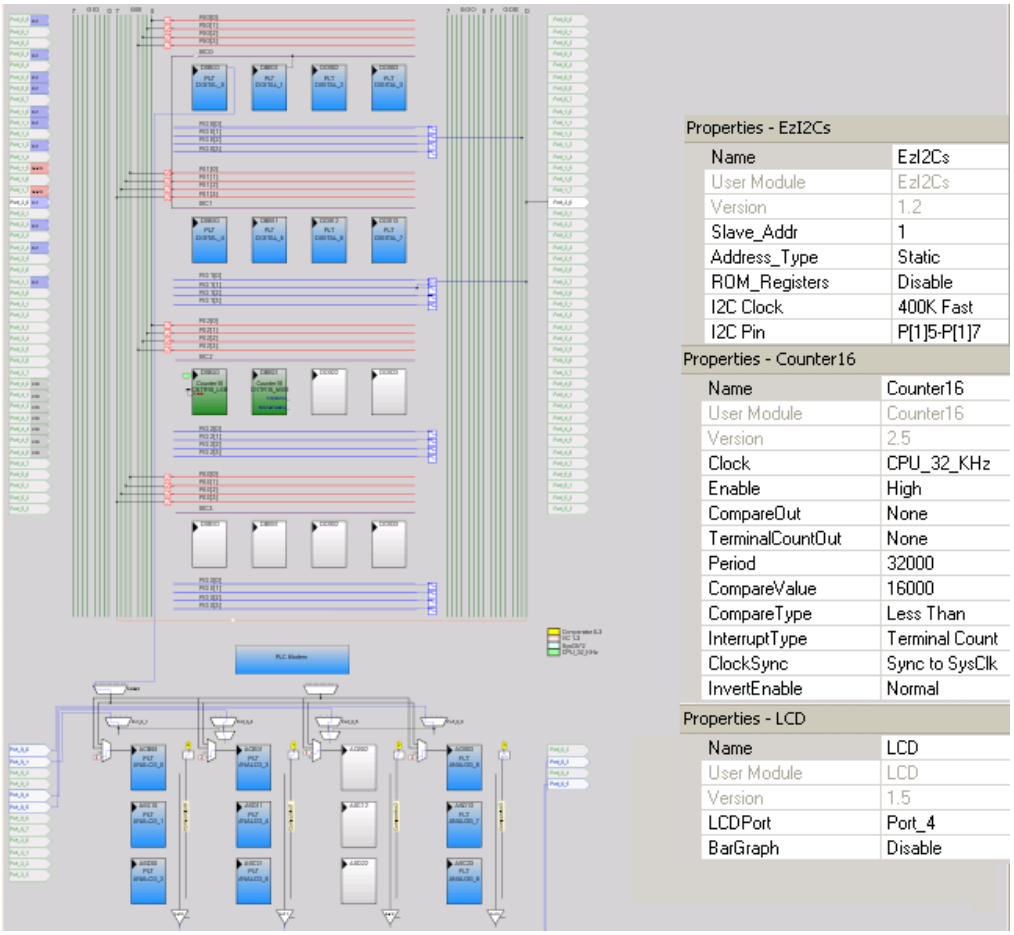
The following user modules are used in this code example:

- PLT (Powerline Transceiver) with the “FSK Modem + Network Stack” user module option
- EzI2Cs – To provide the I2C interface, so that the Control Panel GUI can be connected to the node.
- LCD – To display the repeater statistics.
- Counter16 – To generate a one second interrupt for timing of the repeater algorithm

.

The following figure shows the interconnect view for this application

Figure 12 - Interconnect View



PLT UM properties are set in the application code.

The LCD is used to display the statistics. The top row on the LCD displays the number of packets repeated, the number of packets for which the confirmation is received and the Serial ID of the packet transmitted. The bottom row displays the number of packets received in the repeater mode, the number of packets in the database, and the serial ID for the most recent repeater packet received. The following table shows the structure on the LCD:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Top Row	T	X	=	X	X		C	=	Y	Y		I	D	=	Z	Z
Bottom Row	R	X	=	X	X		D	=	Y	Y		I	D	=	Z	Z

Adding an I²C interface with the "PLC Modem + Network Layer" User Module Option

The "PLC Modem + Network Layer" user module option is used because it allows the application code to initiate packet transmission. However, it does not have an I²C interface. This code example requires the I²C interface to communicate with the PLC Control Panel GUI. The EzI2Cs user module should be placed to add the I²C interface. Also, the following code should be included in the initialization to start the EZI²C and interface it to the memory array.

```
EzI2Cs_SetRamBuffer(sizeof(PLT_Memory_
Array), 0x41, PLT_Memory_Array);
EzI2Cs_Start();
// Turn on I2C
```

Also, the following code should be added in order to interface the PLT user module to the Control Panel GUI.

```
if
(PLT_Memory_Array[TX_Message_Length] &
Send_Message)
{
      PLT_SendMsg();
      PLT_Poll();

}
if (PLT_Memory_Array[Threshold_Noise]
& 0x40)
{
      PLT_AutoSetBIUThreshold();
      PLT_Memory_Array[Threshold_Noise
] &= ~0x40;
}
```

Updating the Result

The repeater code example updates the host application whenever a confirmation is received for the repeater packet or the packet is discarded from the database after 15 seconds. It also updates the host application when a packet is received for the host application.

This code example uses memory array register 0x37 to return the result for the transmitter. The following figure shows different fields in this register.

Bit 7	6	5	4	3:0
Tx Success	Tx No ACK / Tx No Confirmation	BIU Timeout	Tx No Response	Serial ID Tx

Bit Field Description

Tx Success (Bit 7) – This bit specifies if the transmission is a success. When the confirmation is received for any of the packets in the database, this bit is set. The bit is 0 when the confirmation has not been received and the packet is discarded from the database after 15 seconds.

Tx No ACK / Tx No Confirmation (Bit 6) – This bit is set if the acknowlegment is not received in the normal mode or the confirmation is not received in the repeater mode.

BIU Timeout (Bit 5) – In case the band-in-use (BIU) timeout error occurs, this bit is set.

Tx No Response (Bit 4) – In case the response is not received, then this bit is set. This is valid only when using the normal mode and sending a request command.

Serial ID (Bit 3:0) – This field specifies the serial ID of the transmitted packet for which the result is updated. This is required as the database contains up to 6 different repeater packets and the result can be updated for any of the packets. In the normal mode, this field is 0x00 and in the repeater mode it is from 0x01 to 0x0F.

Similarly, memory array register 0x38 is used to return the result for the receiver. If any packet is received over the powerline (normal mode or repeater mode), then the bit 7 in the register 0x38 is set. The Serial ID for the receiver is available in bits 3:0. In normal mode, this field is 0x00 and in the repeater mode, it is from 0x01 to 0x0F.

Bit 7	6	5	4	3:0
Data Received	RSVD			Rx Serial ID

It also uses memory array register 0x39 to store the repeater version.

Use of PLC Control Panel GUI to Evaluate the Repeater

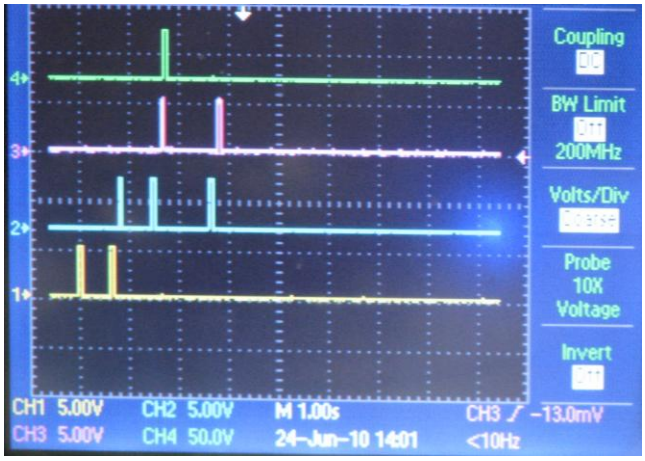
The Control Panel GUI can be used at the source to initiate a packet transmission in the repeater mode. The GUI can also be used at the destination to receive packets in repeater mode. Please refer to the Control Panel User Guide for more information on using the Control Panel GUI in repeater mode.

The PLC Control Panel GUI is available at http://www.cypress.com/?rID=38135

Performance Analysis

The repeater firmware was tested for the system described in Figure 3 on page 2. In this system, Node A communicates with Node D through Node B and Node C. Node B and C act as repeater nodes. The following figure shows the plot of packets transmitted by all the nodes.
The Tx LED pin P2[5] was probed for each of the nodes.

Figure 13 - Repeater Performance For Single Packet



In Figure 13, the yellow plot corresponds to Node A, the blue plot to Node B, the pink plot to Node C and the green plot to Node D.

As a part of the repeater algorithm, Node A firstly transmits a direct packet to Node D (First vertical bar in the yellow plot). As Node A doesn't receive the acknowledgement in 500 msec from Node D, it broadcasts the repeater packet (second vertical bar in the yellow plot.

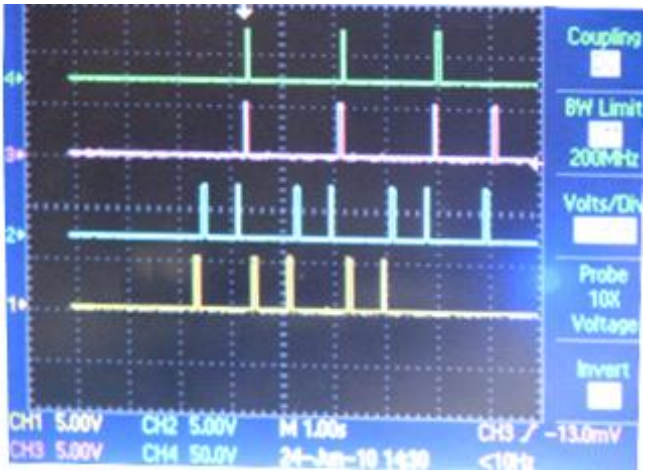
The broadcasted packet is received by Node B, which again sends a direct packet to Node D (First vertical bar in the pink plot). Again, as Node B doesn't receive the acknowledgement, it broadcasts the packet (Second vertical bar in the pink plot). The broadcasted packet from Node B is a confirmation for Node A and therefore, Node A doesn't transmit any further packet.

The broadcasted packet from Node B is also received by Node C, which sends a direct packet to Node D (First vertical bar in the blue plot). As Node D is visible to Node C, Node D receives the packet and sends the acknowledgement back (vertical bar in the green plot), which is a confirmation for Node C. At this point, Node D has received the packet whereas Node A and C have the confirmation.

As Node B doesn't have the confirmation, it again broadcasts the packet after 1 second (Third vertical bar in the pink plot). Node C and node A receive the packet and Node C sends the confirmation back to Node B (Second vertical bar in the blue plot). This way, the packet has been delivered to Node D and all other nodes have the confirmation.

The following figure shows the plot for 3 consecutive packets transmitted by Node A in repeater mode.

Figure 14 - Repeater Performance for 3 Packets



In this figure, the yellow plot corresponds to Node A, the blue plot to Node B, the pink plot to Node C and the green plot to Node D.

The plot is similar to the one shown in Figure 13. However, note that Node A starts preparing the second packet as soon as it receives confirmation from Node B (check the duration between second vertical bar in the pink plot and third vertical bar on the yellow plot). The throughput of the system is increased because Node A doesn't wait for the confirmation from Node D before transmitting a new packet.

Document History

Document Title: AN62487 - Cypress Powerline Communication (PLC) Repeater Implementation

Document Number: 001-62487

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2959674	ROSG	06/25/2010	New Application Note.
*A	3212774	FRE	04/01/2011	Updated code example for PSoC Designer 5.1 SP1 Added a reference to the application note “Using CY8CPLC20 in Powerline Communication (PLC) Applications - AN54416”
*B	4053415	ADIY	07/09/2013	Updated template. No content update.

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
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com