UART to DMX Bridge

AN49187

Author: Krishna Prasad

Associated Project: Yes
Associated Part Family: CY8C29x66, CY8C27x43, CY8C21xxx
GET FREE SAMPLES HERE
Software Version: PSoC Designer™ 5.0
Associated Application Notes: AN45022

Application Note Abstract

This application note discusses how PSoC® can act as a bridge between UART and DMX 512 protocols. An example program for UART to DMX Bridge and a PC based user interface to control the DMX device are also demonstrated.

Introduction

DMX 512 is an industry standard protocol based on RS 485, which is used in lighting control applications in concert halls, theatre lighting, and others. UART is a commonly used standard serial communication protocol for short distance communication applications. This application note demonstrates how to bridge these two protocols, so that existing products using UART can be used to control lighting devices.

This application note discusses the firmware, hardware, and user interface required to demonstrate the protocol bridge solution using PSoC®. The user interface application at the PC sends out the command for brightness control of a particular lighting device. This command is sent through the UART to the PSoC Bridge, which then transfers the data in DMX format to the selected DMX device.

Features

The key advantages of using PSoC to bridge UART and DMX are:

- Configurable UART baud rates
- Configurable DMX address ranges
- Programmable GPIOs
- Additional digital and analog resources that can be configured as PWMs, counters, timers, IrDA, SPI, I2C ADCs, DACs, and more.
- Can shift to sleep Mode when idle and wake up on a GPIO interrupt
- Internal oscillator (option for connecting external crystal)

DMX Protocol

The DMX protocol was developed by the USA Institute of Theatres Technologies (USITT) in 1986. This protocol was represented as the standard interface for data communication between devices that are used for lighting. This protocol has been modified since 1990, and is now known as USITT DMX512.

On the physical layer, the DMX protocol uses the RS485 standard, which defines the electrical interface levels, voltages, and currents on the bus. The RS485 standard provides receiver and transmitter “ground” connections to the cable core screen. This reduces external noise influences and improves electrical safety.

However, the voltages of the transmitter and receiver “ground” lines can be different, resulting in current through the cable wire shield. If the current is too great, the DMX cable and the electronic circuit may be destroyed. To ensure safety and the proper operation of devices that use DMX, it is necessary to implement galvanic isolation of DMX receiver devices.

The DMX data stream is passed in the form of a burst that is repeated continually. This data burst consists of a synchro preamble, which informs the receiver that a message is about to start. After the preamble, a stream of serial data frames follows. This stream contains the values of each channel from 1 to 512 or less (Figure 1 on page 2). The bit rate in DMX protocol is 250 kbaud. The duration of every bit is 4 µsec. The number of channels is not fixed, but it is limited to 24-512 by the DMX 512 standard. If the information is given over all 512 channels, then the maximum update rate is 44 Hz. The maximum update rate is 836 Hz, which corresponds to 24 channels.

Figure 1. Structure of DMX 512 Signal

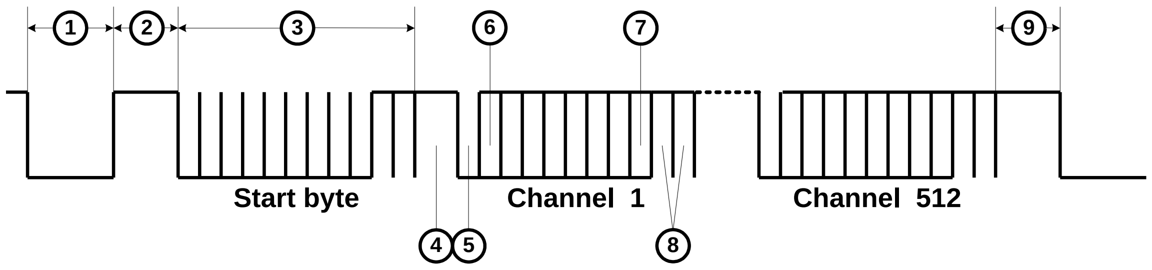


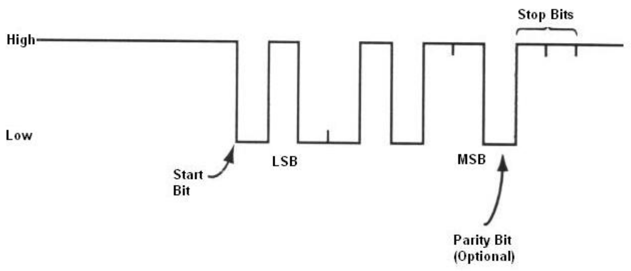
Table 1. DMX Package Field

Reference	Description	Duration
1	Space for break (reset)	Minimum 88 µs
2	Mark after break (MAB)	8 µs – 1 s
3	Slot time	44 µs
4	Mark time between slots	0 µs –1 s
5	Start bit	4 µs
6	Least significant data bit	4 µs
7	Most significant data bit	4 µs
8	Stop bit	4 µs
9	Mark before break (MBB)	0 µs –1 s

UART Protocol

UART is an 8-bit asynchronous serial receiver transmitter that supports duplex RS-232-compliant data format serial communications over two wires. The received and transmitted data format includes a start bit, eight data bits, an optional parity bit, and a stop bit as shown in Figure 2. Each bit interval id is decided by the speed of the clock used for communication (baud rate).

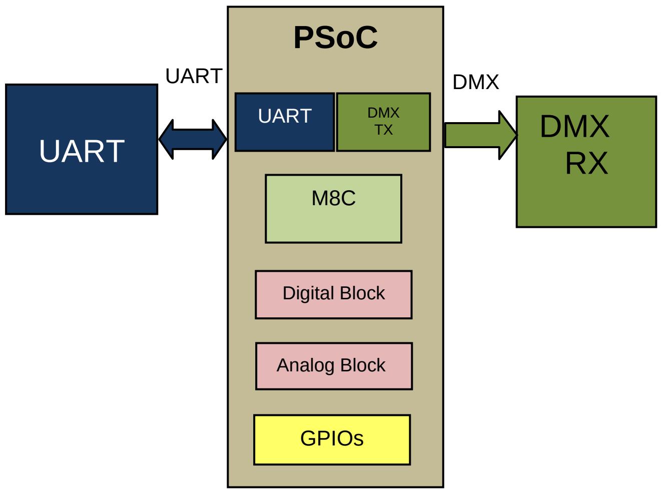
Figure 2. Structure of a UART Signal



Block Diagram

The block level representation of a UART to DMX Bridge application is shown in Figure 3.

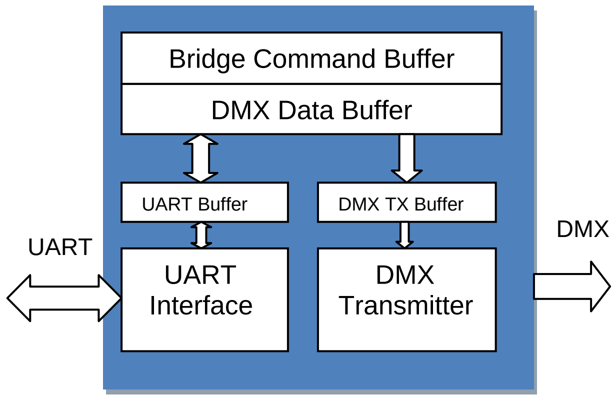
Figure 3. Block Diagram of UART to DMX Bridge



[+] Feedback

As shown in Figure 3 on page 2, the UART device sends DMX device control data for each channel in the form of a packet. This packet contains the channel number and the data corresponding to that channel. Since DMX is a one way protocol and does not involve any feedback about data reception on the receiver, the DMX data is sent out repeatedly over the DMX bus. As a result, all the DMX channel data must be stored. The Bridge contains a DMX data buffer of 512 bytes for this purpose. There are other control commands sent from the UART device to the Bridge, such as sleep command, wake up command, and DMX channel range command that are stored in a separate command buffer. A block diagram of the Buffer structure inside the Bridge is as shown in Figure 4. Depending on the command buffer content, the data present in the DMX data buffer is sent out on the DMX bus.

Figure 4. Block Diagram of DMX Buffer Structure

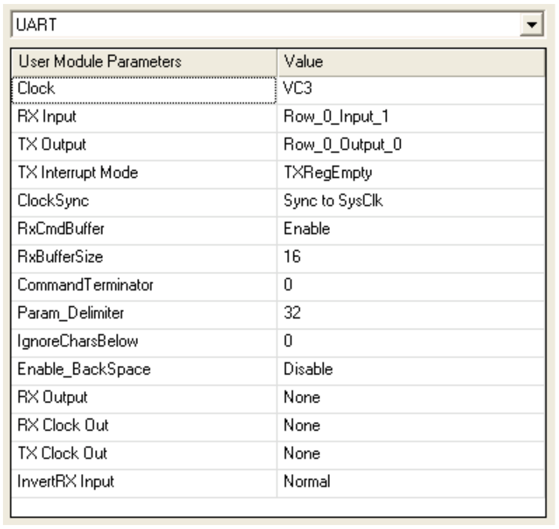


Configuring PSoC as UART to DMX Bridge

UART Interface

The UART interface on PSoC uses two digital blocks: one block configured as a transmitter and the other as a receiver. An external UART level translator IC, which converts RS232 signals to TTL signals and vice versa, must connect to an RS232 bus. A similar level translator is on the PSoC Eval1 Board (CY3210). Commands from the external UART device are received by the UART User Module and the command is stored in a UART receiver buffer. The reception of UART data is acknowledged by the PSoC Bridge that sends a packet to the external UART device. The command received is then decoded and the data is stored in the DMX data buffer or the Bridge Command buffer. The UART receive operation at the PSoC is interrupt driven, that is, an interrupt is generated for every byte received over UART. The configuration of the UART in PSoC is shown in Figure 5.

Figure 5. UART Configuration


UART

User Module Parameters	Value
Clock	VC3
RX Input	Row_0_Input_1
TX Output	Row_0_Output_0
TX Interrupt Mode	TXRegEmpty
ClockSync	Sync to SysClk
RxCmdBuffer	Enable
RxBufferSize	16
CommandTerminator	0
Param_Delimiter	32
IgnoreCharsBelow	0
Enable_BackSpace	Disable
RX Output	None
RX Clock Out	None
TX Clock Out	None
InvertRX Input	Normal

DMX Transmitter

The DMX Transmitter in PSoC is constructed using a TX8 User Module and a PWM8 User Module. The output of the TX8 passes through an optoisolator to prevent the ground loop in DMX communication described in an earlier section. The signal then passes through an RS485 level translator (for example, 75LBC176A). The PWM8 User Module is used to generate a break signal of minimum 88 µs duration. At the start of a DMX frame, the following sequence of actions are performed:

- Disconnect the TX8 output from the DMX output line
- Connect the PWM8 output to the DMX output line
- Stop the TX8 user module
- Start the PWM8 user module

The PWM8 generates a compare true interrupt at approximately 100 µs, generating the Break signal for DMX communication. After the compare true interrupt is generated (that is, the Break signal is sent), the following sequence of actions are performed:

1. Disconnect the output of PWM8 from the DMX output line.
2. Connect the output of TX8 to the DMX output line.
3. Stop the PWM8 user module.
4. Start the TX8 user module.

A Mark After Break (MAB) signal at a minimum of 8 µs is generated by using a Software delay routine. After the MAB is generated, the DMX channel data stored in the DMX data buffer is sent one channel after the other using the TX8 user module.

DMX Data Buffer

The DMX protocol supports a minimum of 24 and a maximum of 512 devices. As a result, the data buffer for DMX must be 512 bytes long. This buffer is updated every time a new data packet is received through UART. If the Bridge is not put into sleep mode by the external UART controller, the data values for each channel are sent continuously over the DMX line.

DMX Command Buffer

The Bridge supports the following three types of commands:

- **Data Command**. This command indicates that the packet is a data packet.
- **Sleep Command**. This command puts the Bridge into sleep mode.
- **Channel Limit Command**. This command indicates the maximum number of DMX devices supported.

These commands are stored in separate variables inside the firmware, and are used to control the data flow over the DMX line.

Firmware

The UART to DMX 512 Bridge firmware has the following two major blocks:

- UART
- DMX Transmitter

The UART receive operation is interrupt driven, that is, an interrupt is generated for every byte of data received over the serial link. Data is transferred from the UART transmitter to the Bridge in the form of a predefined data packet structures. These structures of data are decoded using the UART Command buffer in PSoC. The following four types of packets are supported by the Bridge:

- Data Packet
- Channel Limit Packet
- Sleep Packet
- Connection Check Packet

The structures of these packets are shown in Figure 6 to Figure 9.

Figure 6. Structure of Data Packet

SOP=0xFF	Data Packet	Channel MSB	Channel LSB	Data	EOP = 0xAA	Terminator

Figure 7. Structure of Channel Limit Packet

SOP=0xFF	Channel Limit	Channel MSB	Channel LSB	EOP = 0xAA	Terminator

Figure 8. Structure of Sleep Packet

SOP=0xFF	Sleep Packet	EOP = 0xAA	Terminator

Figure 9. Structure of Connection Check Packet

“H”	“I”	Terminator

The Bridge acknowledges the reception of these packets by sending a Receive Acknowledge packet (in case of a Data, Channel Limit, and Sleep packet) or a Device Connected Packet (in case of a Connection Check packet). All the packets are terminated by a command terminator, which is 0 in this case. Because 0 is used as a command terminator, the occurrence of 0 must be prevented anywhere else in the packet. This is done using Consistent Overhead Byte Stuffing (COBS) encoding and decoding. For more information, refer the IEEE April 1999 Transactions on Networking Paper.

When the Bridge receives a Sleep command, it turns off the UART and TX8 user modules and configures a GPIO interrupt on UART RX pin. Next, the Bridge puts the Bridge device to Sleep and turns off the power to the external DMX level translator and isolation circuit. To wake up the Bridge, a dummy byte of data must be sent to the Bridge over the UART RX line. This dummy byte is not received, but triggers a rising edge interrupt on the RX line and wakes up the Bridge device. At the GPIO ISR, the user modules are again started. The GPIO interrupt is disabled on the RX line so that the UART operates normally after wakeup, and the external isolation circuit and level translator are turned on.

GUI Application

A GUI Application written in C# is used to send data to the Bridge to control its operation. A screenshot of this GUI application is shown in Figure 10.

The transmit checkbox must be checked before transmitting data to any channel, otherwise none of the changes made to the channel data are sent to the Bridge. The packet size field is used to limit the number of channels supported by the Bridge. Block control is used to move between different channel blocks, where a set of 16 channels constitutes a block. Check the sleep checkbox when the Bridge needs to be put into sleep mode. Conversely, to wake up the Bridge, uncheck the Sleep Checkbox. You can change the data for any particular channel using the scrollbars provided below the channel number. There are two connection status indicators on the GUI: one is the Bridge connection status and the other is the COM port connection status.

Summary

This application note demonstrates using PSoC as a UART to DMX Bridge IC. This solution uses PSoC to provide the flexibility of configuring the Bridge for different user requirements.

About the Author

Name: Krishna Prasad

Title: Applications Engineer

Contact: kris@cypress.com

Appendix

Figure 10. GUI Application Screenshot

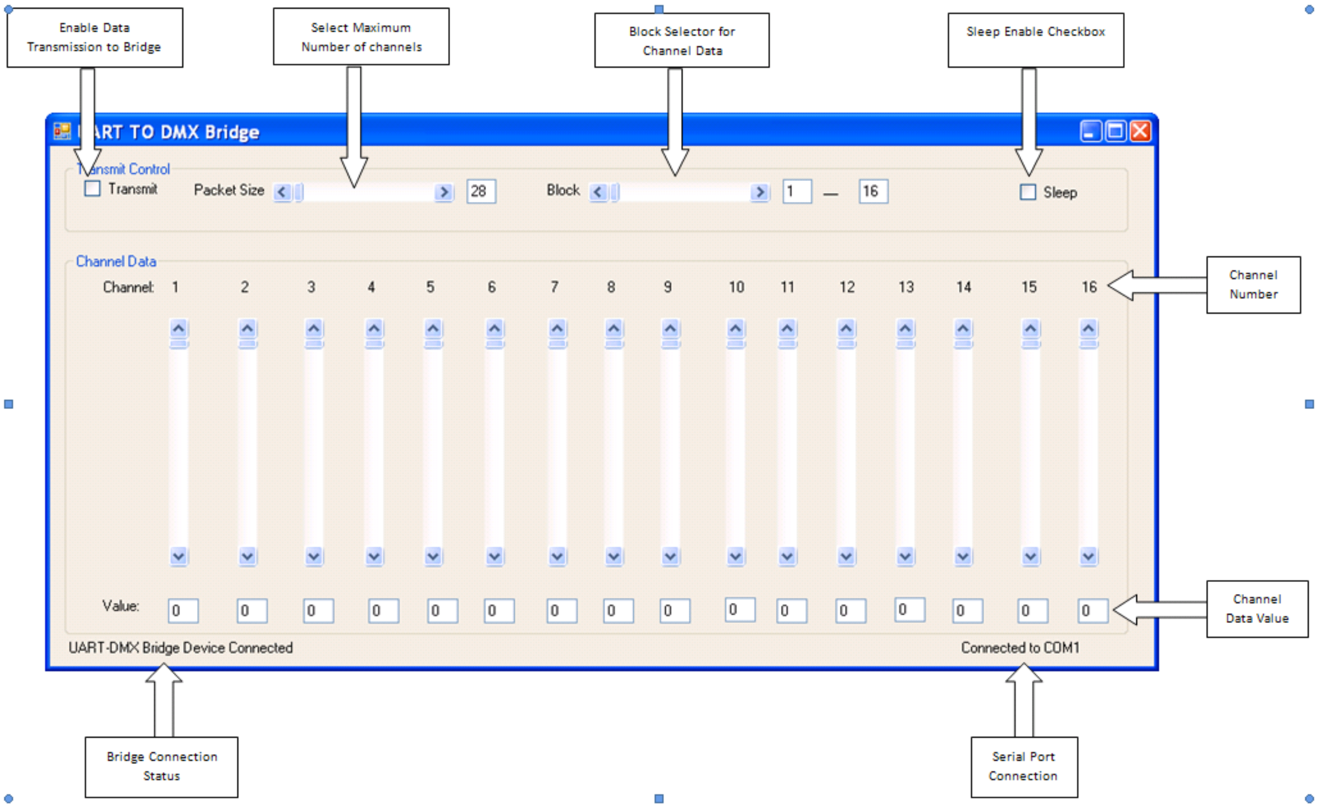
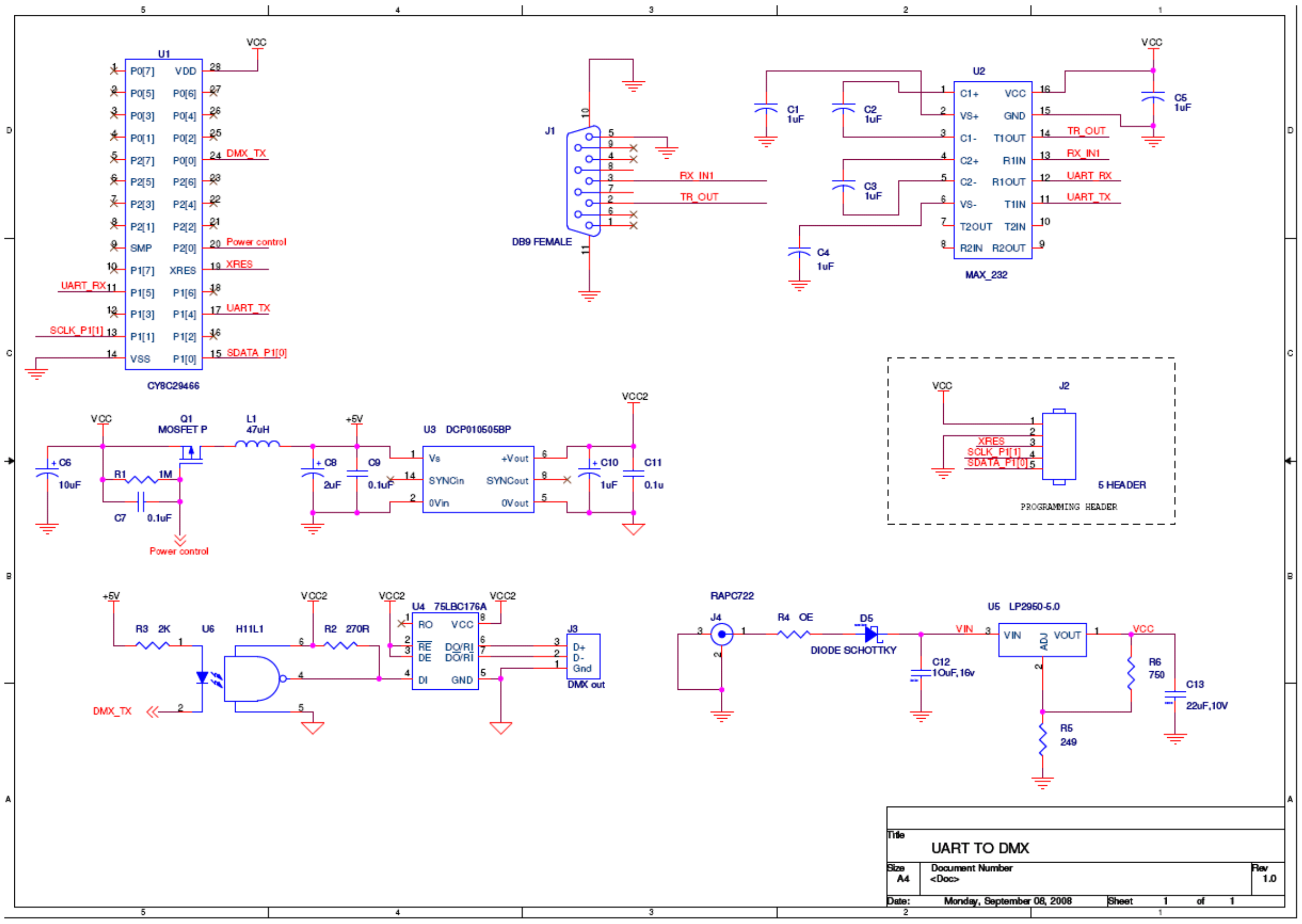

Enable Data Transmission to Bridge
Select Maximum Number of channels
Block Selector for Channel Data
Sleep Enable Checkbox
UART TO DMX Bridge
Transmit Control
Transmit
Packet Size
28
Block
1
16
Sleep
Channel Data
Channel: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Channel Number
Value: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Channel Data Value
UART-DMX Bridge Device Connected
Connected to COM1
Bridge Connection Status
Serial Port Connection

Figure 11. Schematic



Document History

Document Title: UART to DMX Bridge

Document Number: 001-49187

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2575690	KRIS/AESA	10/01/2008	New Application Note.

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