EE 5314 Embedded Microcontroller System Design
Fall 2015 Project

1 Overview

The goal of this project is to build a device capable of acting as either a controller or device for a timing intensive asynchronous communications interface based on the DMX512-A protocol with EF1 topology. The PC transmitter will accept commands from a PC via an RS-232 interface and will continuously transmit a serial stream to control up to 512 devices on a RS-485 communication bus. The PC receiver will forward data received from devices on a communications bus and send these to the PC with the RS-232 interface.

Devices on the bus will extract information out of the asynchronous data stream and will control one or more devices. They will also send an acknowledgement to the controller when requested.

2 DMX-512-A Background

The discussion in this section is incomplete and is intended to convey only the minimum information required to build the project. The user should read the complete specification to ensure that information in this document is accurate if building a real product.

The DMX512-A protocol was originally designed to control dimmers on stage lamps replacing the earlier voltage controlled, remote, motor-driven autotransformers used for light control. This interface has expanded to encompass control of other devices such as lamp color control, lamp tilt and pan control, smoke and snow machine activation, and other devices. It has also been introduced into numerous commercial and residential applications as well. The protocol has no error correction and as such is not designed to control devices whose disoperation could cause a safety concern.

The protocol for this interface was originally published in 1986 and was revised most recently to ANSI E1.11-2008 (R2013) by PLASA North America (https://www.plasa.org/).

The DMX-512-A protocol uses a RS-485 physical layer. RS-485 specifies the electrical characteristics of the multiple-drivers and multiple-receivers on a differential bus. A 120ohm resistor is typically used to terminate each end of the physical RS-485 links when using the DMX512-A protocol.

For this project, the DMX-512-A spec with EF1 protocol is used (single differential pair (primary link) for communications).

On the primary link, a controller continuously sends a cadence of break, mark-after-break (MAB), and N+1 data frames, where N is the number of addresses used in the DMX universe (N is always 512 or less). A start code (usually 0) is sent before the N addresses, yielding the N+1 term. The principle unit of time is 4us on the primary link (250 kbaud). The break signal is 92-176us. The MAB is 12us or longer. The data frames consist of a start bit (space), 8 data bits (LSb first), and 2 stop bits (marks). The time for spaces, marks, and data bits are 4us. After the second stop bit of each data frame, additional time may be inserted. The maximum period of any break, MAB, or mark period shall not exceed 1s. The BREAK-to-BREAK time is defined between 1204 us and 1s.

It is also possible for a device to respond to commands from the controller. While the DMX-512-A standard does not stipulate the content of the data (just 4 possible topologies (EF1-4), other specifications, such as the Remote Device Management (RDM), define this operation. Our project will define a much simpler talkback feature to allowing polling of devices.

3 Hardware

Microcontroller:
An ARM M4F core (TM4C123GH6PMI microcontroller) is required.

Serial interface:
If using the EK-TM4C123GXL evaluation board, then the UART0 tx/rx pair is routed to the ICDI that provides a virtual COM port through a USB endpoint. Otherwise, a SP3232E interface can be used.

DMX-512-A interface:
When operating as a controller, the transmitter continuously outputs data at the 250 kbaud rate on the primary link unless it is awaiting a response for a device. The CMOS-level signals are translated to RS-485 differential format with a SN75HVD12P. The SN75HVD12P transmits and receives differential primary link data. The transmitter is controlled by a GPIO line and the received is enables at all times. Please provide bypassing on the power pins.

Address Switch:
The device shall read a 9-pole DIP switch to determine the starting address for reception when operating in device mode..

Activity LED:
The blue LED on the board will act as a RX LED to indicate when RS-232 data has been received when operating in controller mode.

Transmit LED:
The red LED on the board will indicate when DMX-512-A data is transmitted. In controller mode, it will glow continuously when data is being sent. In device mode, it will blink to indicate that a response was sent on the bus.

Receive LED:
The green LED on the board will indicate when DMX-512-A data is received. In controller mode, it will blink to indicate when a response is received. In device mode, it will stay illuminated to indicate that DMX-512-A data is has been received in the maximum allowed time for the protocol. If the data values being monitored change value, then the LED will temporarily blink off.

4 Suggested Parts List

Part	**Quantity**
EK-TM4C123GXL (rs-232 receiver/driver)	1
SN75HVD12 (rs-485 transceiver)	1
0.1uF capacitor (bypassing)	1
9-pos DIP switch	1
8pin 300mil socket (sn75hvd12)	1
10x2 100mil pitch unshrouded header	2
2-pin terminal block	1
Wire (22-24 AWG solid wire, 3 colors)	1
Power supply of your choosing if not using the EK-TM4C123GXL	1
PC boards (line driver and LED)	1
Device under control and interfacing	Varies
Solder, iron, needle-nose pliers, diagonal cutters, safety glasses, wire, cable, …	1 each

5 Controller Software

Commands:
On startup, the controller software should send “Ready” to the host machine on the serial interface.

The user of the host machine can send several commands to the controller:

If “clear” is sent, then the data value of all 512 DMX addresses sent on the primary data link shall be set to zero.

If “set A, V” is sent, where A is the DMX address and V is the value, the controller should update the value being transmitted to address A to a value of V.

If “get A” is sent, where A is the DMX address, the controller should return the value being sent to address A.

If “on” is sent, then the DMX stream shall be sent continuously.

If “off” is sent, then the DMX stream will not be sent.

If “max M” is sent, then the maximum address transmitted shall be M. The default value is 512.

If “poll” is sent, then a series of experimental poll requests are sent to find what devices are on the bus.

When a device responds to the poll, the address of the device is sent to the PC.

After each command is received from the PC, the controller shall respond with “Ready” or “Error”.

Transceiver:
The controller shall transmit continuous DMX packets with a null start code (00h) containing M frames on the primary link, except when the link is “off” or when an alternative start code (F0h) is being sent. The controller shall also receive DMX packets on the primary link.

To poll devices, a binary search can be used to isolate the device present on the bus. The experimental poll function used an alternative start code of F0h (not a valid number for real products), followed by 512 values aligned with each address. If the byte is 1, then a poll response is requested for the corresponding address. If the byte is 0, then no poll response is requested for the corresponding address. The transmitted should then release the bus within 8us after transmission is complete.

All devices on the bus with a mask byte set to 1 should respond to the poll with a break The transmitter will be disabled for a period of time necessary to receive the poll responses (see device section) and should then resume normal DMX transmission if “on”. The binary search continues until the devices are uniquely identified.

Activity LED:
The activity LED should blink on to indicate that PC data has been received.

Transmit LED:
The transmit LED should blink on to indicate that DMX data has been transmitted.

Receive LED:
The receive LED should blink on to indicate that DMX data has been received.

6 Device Software

Address:
The device shall read the address switch to determine the starting address (ADD).

Link:
When a null start code (00h) is received, the corresponding values are extracted from the DMX stream on the primary data link to control the device as appropriate.

When an alternative start code of F0h is received, then the device will look at the byte with an offset equal to ADD. If the byte value is 1, then it will wait 16us, enable the transmitter, send a break, wait 8us, and disable the transmitter.

Receive LED:
The activity LED should light to indicate that a DMX signal has been detected. It should blink off if a change occurs in the value(s) being monitored by the device.

Transmit LED:
The transmit LED should blink on to indicate that DMX data has been sent.

Controlled Device:
The software should control one or more devices. .

7 Testing

It is required that you complete construction of the hardware by 10/5.

Your controller and device will be tested with multiple DMX512-A devices on a common bus. Please verify operation with other devices prior to the defense date.

Computers and lab equipment will be provided on campus in Rm 148 NH for you to work on this project. If you do plan on plugging your project into your own machine, do so at your own risk and only after having the hardware tested. Again, you are responsible for anything that happens to your personal machine. Do not connect your project to any machines in the UTA computing labs or in other EE labs.

8 Deadlines

Project is due on the date shown in the syllabus, with an oral defense, electronic copy of your code, written report (containing theory of operation and software printout), and demonstration of hardware and software (including compilation on site). You may work in teams of up to three members. All members of the team should participate equally and be prepared to answer any question about the project to avoid a deduction in points. Each member of the team will be graded independently, although only one report, hardware, and software submission is needed.

9 Safety Issues

While far beyond the scope of this document, it is important to use tools safely. Safety goggles are a good idea, since you can cause yourself great injury if a wire that is being cut flies toward you. Another good reason for safety goggles is that if unsoldering wires under some strain, the solder can be fling toward you. Soldering entails some care to prevent burning yourself or a burning down a building if you forget to turn it off. If you choose to use solder containing lead, then care should be taken to dispose of lead properly (don't cool off the iron in a drinking fountain, etc.). OSHA also recommends washing your hands after using solder to prevent the build-up of heavy metals. These are a few helpful suggestions and are a very incomplete listing. Please read and understand all safety labels and exercise caution.

Please utilize the supervised lab resources in Rm 148 NH when working on the project for your safety. You may only use the resources in Rm 148 NH when the GTA or other E.E. staff is present. If another class is meeting in the lab, then the time cannot be used for project work.

Have fun!