Reference Design

Implement a DMX512-A Controller Using the Z51F3220 MCU

RD001601-0415

Overview

The reference design described in this document implements a DMX512-A controller using Zilog's Z51F3220 MCU toward providing a basic understanding of the DMX512-A protocol, including its theory of operation, by using a common lighting application example.

This reference design operates in one of three modes:

Local Mode. In this mode, five control potentiometers are used to set transmitted light levels for each related Digital Multiplex (DMX) channel; each potentiometer corresponds to a single DMX channel.

PC Mode. In this mode, a PC application is required. This PC application has two main functions:

- DMX Controller Function – This function provides two methods of updating and transmitting light levels to a *DMX Universe* through the DMX512-A controller:
  - Auto-Fade: In this method, light levels are set through sliders. All values – from the set light levels down to zero (0) and from zero (0) up to the set light levels – are transmitted continuously when played.
  - Real-Time: In this method, light levels are transmitted each time a slider value changes. The Write to EEPROM function is used to download lighting sequences into the DMX512-A controller's EEPROM.
- Write to EEPROM Function – This function is used to download lighting sequences into the DMX512-A controller's EEPROM.

Demo Mode. In this mode, stored light sequences in the DMX controller's EEPROM are transmitted to a DMX Universe.

The DMX512-A controller reference design is shown in Figure 1.

Figure 1. The DMX512-A Controller Reference Design

> **Note:** The source code file associated with this reference design, RD0016-SC01.zip, is available free for download from the Zilog website.

Features

This DMX512-A controller reference design offers the following feature:

- Three modes of operation: Local, PC, and Demo
  - Local Mode allows control via five on-board potentiometers (1 pot per DMX slot; base address is hard coded in the firmware)
  - PC Mode allows control via a PC application
  - Demo mode allows pre-stored sequences to be generated

Potential Applications

Use this DMX512-A controller reference design to develop a variety of lighting applications, as the following examples suggest:

- Dimmer light controls
- Changing color light controls
- Moving light controls
- Light sequence controls
- Goes Before Optics (GOBO) lighting devices

Discussion

This section describes the DMX512-A protocol, its theory of operation, hardware and firmware design, and a PC application for the PC Mode of operation.

DMX512-A Protocol and Theory of Operation

DMX is an asynchronous serial communication protocol used in controlling remotely-operated lighting devices such as lighting dimmers and intelligent fixtures.

A DMX512-A controller is capable of transmitting data across 512 channels via a DMX cable. Even if the receiver contains only one channel, all of the data for this controller's 512 channels will be transmitted.

Figure 2 shows an example of a DMX network with a DMX receiver/decoder for a single light fixture.

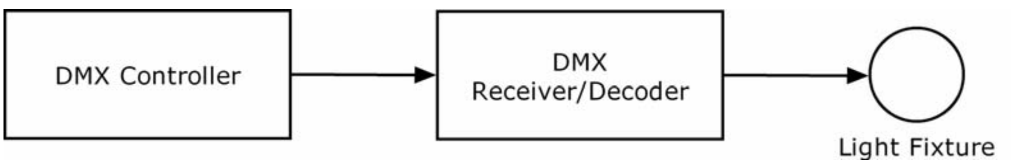

Figure 2. A DMX Network with a Single-Channel DMX Receiver/Decoder

The DMX Receiver/Decoder can be set to any channel ranging from 1 to 512. In this application, the preferred channel can be selected using a DIP switch. If the DMX Receiver/Decoder is set to Channel 1, the Channel 1 data from the data packet transmitted by the DMX512-A Controller will be decoded by the DMX Receiver. This data will adjust the intensity of the light fixture.

Figure 3 shows an example of a DMX network with a DMX receiver/decoder for four light fixtures.

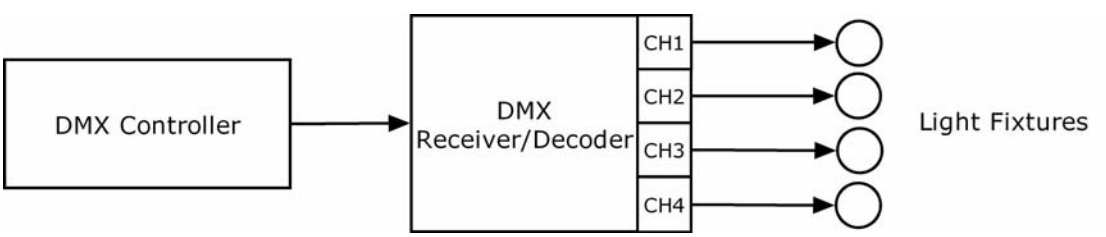

Figure 3. A DMX Network with a Four-Channel DMX Receiver/Decoder

In this Figure 3 example, if the DMX Receiver/Decoder channel is set to Channel 5, its base address will be Channel 5. Data from channels 5, 6, 7, and 8 from the data packet transmitted by the DMX512-A Controller will be applied to the light fixture's CH1, CH2, CH3, and CH4 settings, respectively, as indicated in Table 1.

Table 1. Sample Channel Data Assignment on a Multi-Channel DMX Receiver/Decoder

Base Address	DMX Controller Channel Data	Light Fixtures DMX Receiver
5	Channel 5 data	CH1
	Channel 6 data	CH2
	Channel 7 data	CH3
	Channel 8 data	CH4

Figure 4 shows an example of a DMX network with one DMX controller and two DMX receivers/decoders that can operate a total of eight light fixtures.

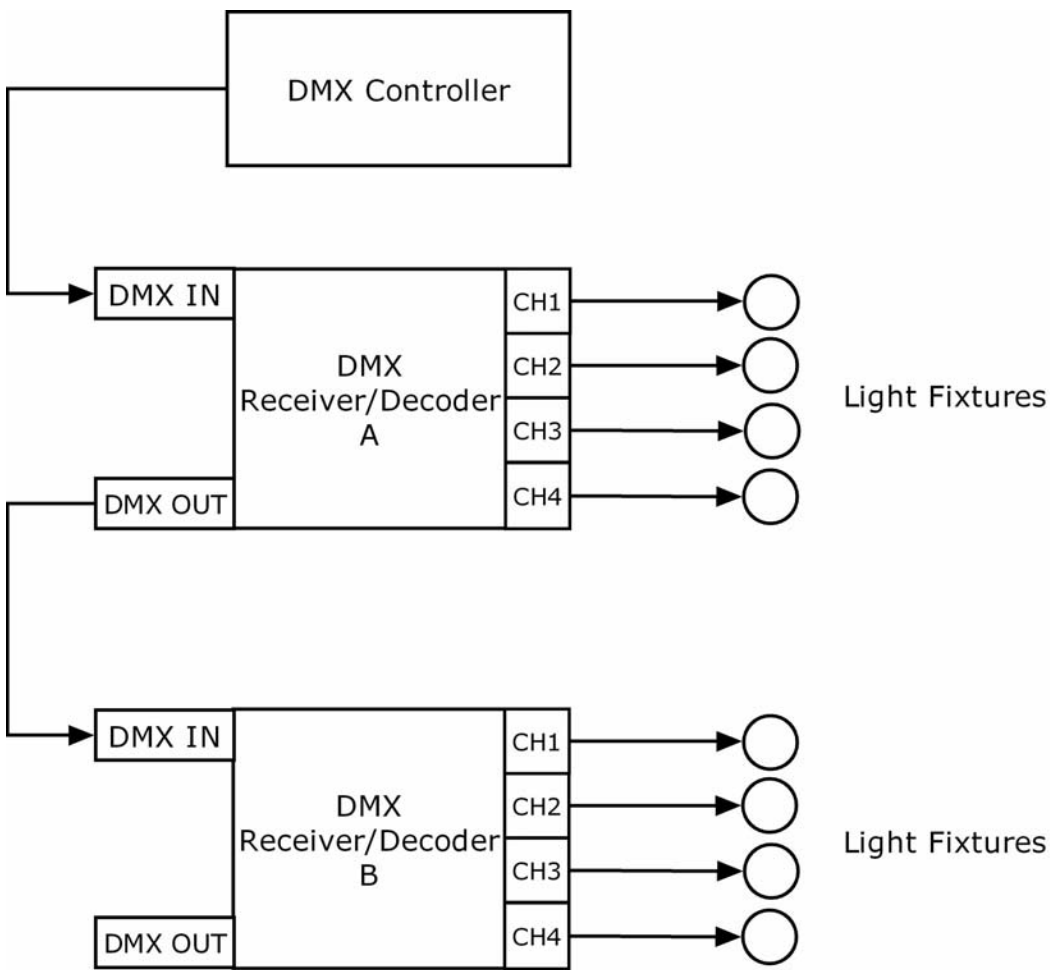

Figure 4. ADMX Network with Two Multi-Channel DMX Receivers/Decoders

In the setup shown in Figure 4, the DMX data packet from the DMX Controller will be transmitted to DMX Receiver/Decoder A and DMX Receiver/Decoder B. If the base address of DMX Receiver/Decoder A is set to Channel 1 and the base address of DMX Receiver/Decoder B is set to Channel 8, the Channel 1–Channel 4 data from the DMX Controller data packet will be applied to the CH1–CH4 light fixtures of DMX Receiver/Decoder A and the Channel 8–Channel 11 data of the data packet will be applied to the CH1–CH4 light fixtures of DMX Receiver/Decoder B. Table 2 shows this example.

Table 2. Sample Channel Data Assignment on Two Multi-Channel DMX Receivers/Decoders

DMX Receiver/ Decoder	Base Address	DMX Controller Channel Data	Light Fixtures DMX Receiver
A	1	Channel 1 data	CH1
		Channel 2 data	CH2
		Channel 3 data	CH3
		Channel 4 data	CH4
B	8	Channel 8 data	CH1
		Channel 9 data	CH2
		Channel 10 data	CH3
		Channel 11 data	CH4

Connectors

The DMX standard specifies the use of 5-pin Cannon X connectors with Latch and Rubber guard (XLR). Connectors are terminals that provide an interface between a DMX controller and a DMX receiver/decoder. Figure 5 and Table 3 show the pin assignments for the cable and XLR connector.

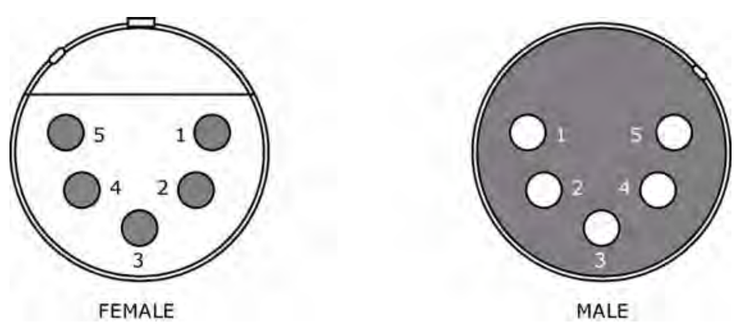

Figure 5. 5-Pin XLR Connectors

Table 3. 5-Pin XLR Connector Pinout

Pin	Wire	Signal
1	Shield Drain	Ground
2	Inner Conductor (Black)	Data -
3	Inner Conductor (White)	Data +
4	Inner Conductor (Green)	Data - (Spare)
5	Inner Conductor (Red)	Data + (Spare)

Data Format

DMX data is transmitted at a frequency of 250 kHz, or at a duration of 4 µs per bit. This data can be produced using the standard UART format, as implemented in this reference design.

Figure 6 illustrates a DMX packet. It consists of the following elements:

- Break and Mark After Break (Reset Sequence)
- Start Code
- Channel Data
- Mark Time Between Frames and Mark Time Between Packets

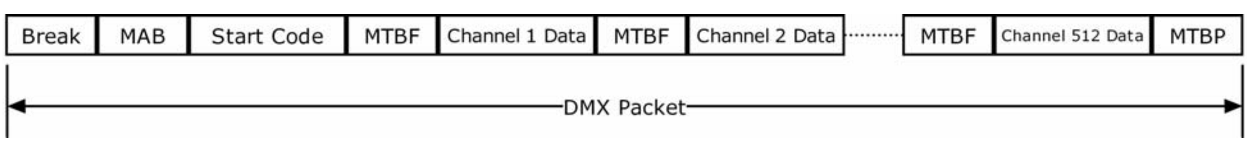

Figure 6. A DMX Data Packet

Figure 7 shows the format for the Start Code and the Channel Data. Both consist of 1 start bit, 8 bits of data, and 2 stop bits.

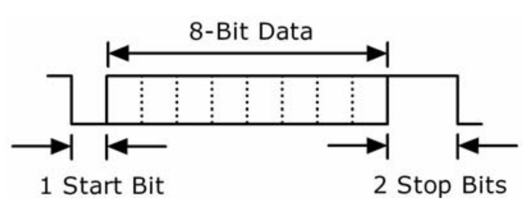

Figure 7. DMX Channel Data

The Start Code identifies the function of the data in the packet. There are two types of Start Code:

- Null Start Code
- Non-Null Start Code or Alternate Start Code

The Null Start Code (0) informs the receiver that the data in Channels 1 to 512 of the packet represents a dimmer value. An Alternate Start Code is used in applications such as an ASCII text packet, a standard test packet, a UTF-8 text packet, and a system information packet. For this reference design, only the Null Start Code is used.

Table 4 provides the logic levels (i.e., states), the number of bits, and the timing requirements for all the components of a DMX packet.

Table 4. Components of a DMX Packet

Element	Description	State	Size	Duration
Break	Break resets the line Indicates new DMX Packet	Logic 0	22 bits–250 kbits	88 µs–1 sec
Mark After Break (MAB)	Informs the receiver to start reading the data	Logic 1	2 bits–250 kbits	8 µs–1 sec
Start Code	Informs receiver of the function of data in the packet Consists of 1 start bit, 8 bits data and 2 stop bits	Mixed	11 bits	44 µs
Mark Time Between Frames (MTBF)	Space between data bytes Transmitted before channel data	Logic 1	0–25 kbits	Up to 1 sec
Channel Data	DMX value for each channel Consists of 1 start bit, 8 bits data, and 2 stop bits	Mixed	11 bits	44 µs
Mark Time Between Packets (MTBP)	Transmitted after Channel 512 data End of DMX Packet	Logic 1	0–250 kbits	Up to 1 sec

Hardware

All DMX512-A controller functions are implemented using a single circuit board, as shown in Figure 8. An external wall mount with a 7–12V power supply is used to supply power to the Board. Alternatively, a USB serial interface can also be used to supply power to the Board.

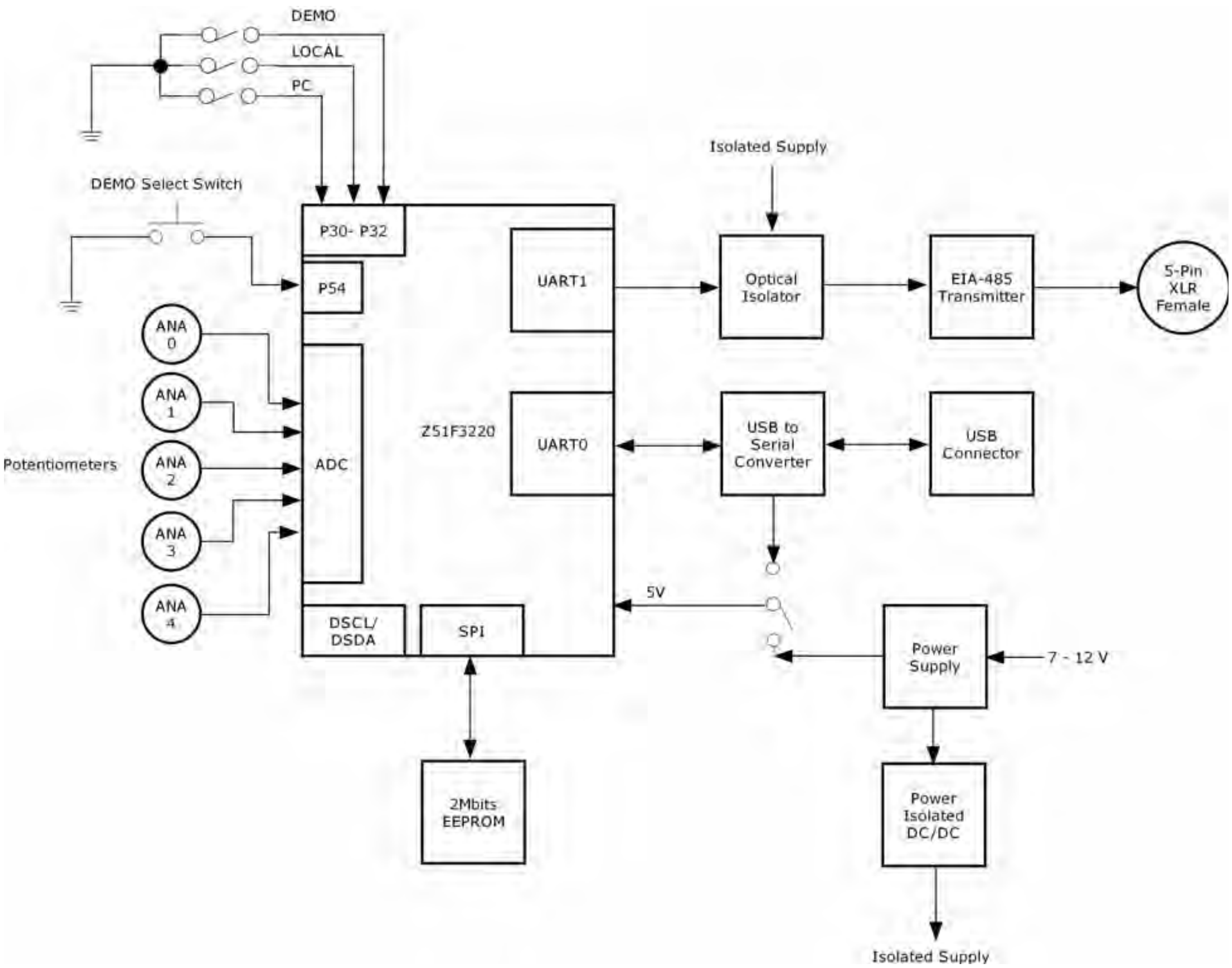

Figure 8. DMX Controller Block Diagram

The DMX512-A controller hardware is based on the Z51F3220 MCU. The features included on this reference design take advantage of the following specific peripherals on this device:

- Five ADC channel inputs for the potentiometers
- SPI for the serial EEPROM
- UART for DMX
- UART for USB

A DMX512-A controller board is capable of operating in three modes. Each of these modes can be selected with this Board's DIP switch. The DIP switch has four positions:

Position 1–Position 3 are for Demo, Local, and PC Mode, respectively. Position 4 is unused.

PC Mode

In PC Mode, UART0 of the Z51F3220 MCU is used to communicate with the PC through the USB chip FT232RL on the USB serial interface. Up to four DMX data light sequences can be saved to the SPI EEPROM (M95M02) when operating in PC Mode.

Local Mode

In Local Mode, the Board can operate without a USB serial interface. The DMX data is sent to the DMX receiver via the five potentiometers. The DMX data will depend on a potentiometer’s angular position. The DMX interface of the controller is connected to the DMX network with a digital optical isolator (CPC5002) and an RS-485 chip (MAX845CSA).

Demo Mode

In Demo Mode, the Board can operate without a USB serial interface. The DMX data that was saved to the EEPROM can be retrieved and sent to the DMX Receiver when operating in Demo Mode. The Demo Select pushbutton is used to retrieve the DMX data.

Firmware

The DMX512-A standard was originally intended for lighting systems. However, due to its simplicity and flexibility, it is commonly used in applications for other devices such as directional control projectors with pan and tilt features, Goes Before Optics (GOBO) lighting, and fog machines.

This reference design was developed to run in three modes: PC Mode, Local Mode, and Demo Mode.

In PC Mode, a Windows-based application is designed to communicate with the DMX512-A controller. It allows the user to control the data to be transmitted for each DMX channel and sequences to be loaded onto the DMX512-A controller's on-board EEPROM.

In Local Mode, the five potentiometers represent the data to be transmitted for each DMX channel. The DMX512-A controller creates a continuous DMX data stream based on these potentiometer values.

In Demo Mode, sequences preloaded onto the EEPROM are interpreted and transmitted as a DMX data stream. The Demo Select pushbutton switch selects which of the four pre-loaded sequences will be used.

PC Mode Routine

The DMX Console, an application running on a Windows PC, was developed to communicate with a DMX512-A controller via a USB interface. To learn more about the DMX Console application, refer to the PC Software Application section on page 17.

A serial protocol between the PC and the DMX512-A controller is devised to provide for this interface. Figure 9 shows a simple command and response scheme used in PC Mode.

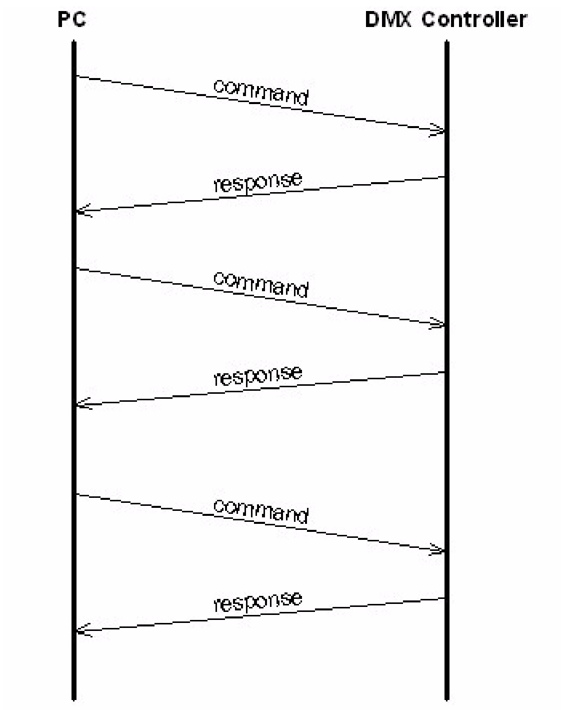

Figure 9. Communication Protocol Scheme

In PC Mode, the DMX512-A controller accepts commands from the PC via the USB interface. The DMX512-A controller can accept two types of commands: a TRANSMIT command and a SAVE command. When a command is received, the DMX512-A controller replies with an ACK or a NAK.

Figure 10 shows the command format understood by the DMX512-A controller. The COMMAND frame identifies the type of command being issued, NUMBYTES determine the number of bytes in the PAYLOAD, and PAYLOAD contains the data bytes. The PAYLOAD length can go up to a maximum of 512 bytes.

Byte#	0	1	2	3	4	5	6	7	...	...	...
	COMMAND			NUMBYTES		PAYLOAD					
TRANSMIT	CMD	SC1	SC2	MSB	LSB	DB0	DB1	DB2	...	...	...
SAVE	CMD	SN	DV	MSB	LSB	DB0	DB1	DB2	...	...	...

Figure 10. Command Format

TRANSMIT Command

A TRANSMIT command feeds the DMX512-A controller with data representing values for each DMX channel. The DMX512-A controller uses these data values to create a continuous DMX data stream which will then be sent to the DMX receiver/decoder through the DMX cable.

A TRANSMIT command frame consists of three bytes: Command byte (CMD), Start Channel MSB (SC1), and Start Channel LSB (SC2). The Command byte is a constant value, 01h, that identifies a TRANSMIT command. The start channel (SC1 and SC2) is a two-byte value that identifies the starting DMX channel in which the payload data bytes begin. The payload contains the DMX levels for each DMX channel starting at the DMX channel specified in SC1 and SC2.

The example in Figure 11 shows a TRANSMIT command issued by the DMX console containing DMX levels from DMX Channel 4 up to Channel 7.

COMMAND			NUMBYTES		PAYLOAD			
01h	00h	04h	00h	04h	55h	75h	32h	FFh

Figure 11. Sample Transmit Command

The DMX512-A controller interprets this command, modifies it, and sends a DMX data packet with channels 4–7 changed to `55h`, `75h`, `32h`, and `FFh`, respectively. Values for other DMX channels remain unchanged. The resulting DMX packet that is sent to the DMX Receiver/Decoder is shown in Figure 12.

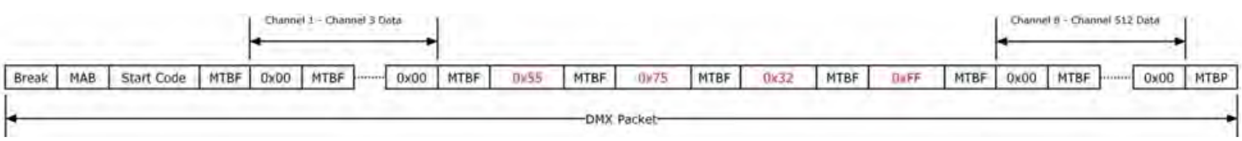

Figure 12. Sample DMX Data Packet Produced by the Transmit Command in Figure 11

SAVE Command

A SAVE command instructs the DMX512-A controller to write sequence information into its on-board EEPROM. A SAVE command frame consists of 3 bytes: a Command Byte (CMD), a Sequence Number (SN), and a Delay Value (DV). The payload contains a series of DMX channel-and-level pairs, as shown in Figure 13.

PAYLOAD								
Channel-and-Level Pair			...			Channel-and-Level Pair		
CN1	CN2	LVL	CN1	CN2	LVL	CN1	CN2	LVL

Figure 13. Payload Data Format for Save Command

Each DMX channel-and-level pair consists of 3-byte data: Channel Number MSB (CN1), Channel Number LSB (CN2), and DMX level (LVL). Altogether, the command frame and the payload describe one EEPROM entry that is used when the DMX512-A controller runs in Demo Mode. To learn more about EEPROM entries, refer to the Demo Mode Routine section on page 13.

The example in Figure 14 shows a SAVE command issued by the DMX Console that stores DMX levels to DMX channels 1, 4, and 8.

COMMAND			NUMBYTES		PAYLOAD								
02h	01h	04h	00h	09h	00h	01h	55h	00h	04h	35h	00h	08h	FFh

Figure 14. Sample SAVE Command

If the command shown in Figure 14 is the first SAVE command received by the DMX512-A controller, it will be interpreted as the first entry in a light sequence with the following details:

Light Sequence:	1
Entry:	1
Time Delay:	NA
Entry Data:	Channel #1, Data = 55h
	Channel #4, Data = 35h
	Channel #8, Data = FFh
	Channel #8, Data = FFh

If the command shown in Figure 14 is not the first SAVE command received by the DMX512-A controller, the same command will be interpreted as follows:

Light Sequence:	1
Entry:	1
Time Delay:	4
Entry Data:	Channel #1, Data = 55h

Channel #4, Data = 35h

Channel #8, Data = FFh

Channel #8, Data = FFh

Local Mode Routine

In Local Mode, the five potentiometers represent the data to be transmitted for each DMX channel. These potentiometers are connected to analog inputs ANA0–ANA4 of the Z51F3220 MCU. ADC data from these analog inputs are the Channel 1–Channel 5 data of the DMX data packet. Unused data in DMX Channel 6–Channel 512 are `0x00`. Figure 15 illustrates this DMX data packet for Local Mode.

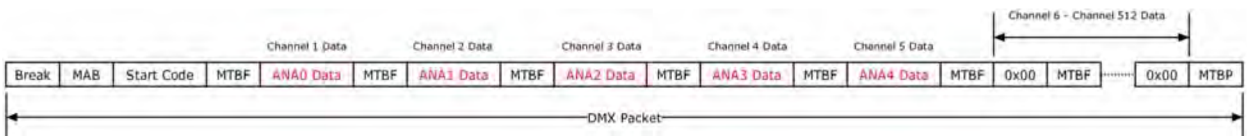

Figure 15. Local Mode DMX Data Packet

The `LOCAL_DMXTransmit` routine is executed in Local Mode. This routine performs the following functions:

- Collects ADC Data from ANA0–ANA4
- Transmits the DMX Data Packet

These functions are continuously performed when the system is in Local Mode.

Inside the `LOCAL_DMXTransmit routine`, the 12-bit ADC-equivalent data of the voltage from the potentiometers will be fetched from the ADC data registers and scaled down to 8-bit data. ADC data are scaled down to 8-bit data because DMX channel data requires 8 bits. After the ADC data from ANA0–ANA5 are collected, the DMX Data Packet will be transmitted.

Demo Mode Routine

In Demo Mode, the `DEMO_ReadLightSequence` routine is executed. This routine performs the following functions:

- Checks the light sequence selected by the Demo Select pushbutton
- Decodes the light sequence stored in EEPROM
- Transmits the light sequence as a DMX data packet

The Demo Select pushbutton switches between the stored sequences. The pushbutton is connected to Port 54 of the Z51F3220 MCU and is configured as an interrupt source with rising edge polarity. The pushbutton normally provides logic 1 to Port 54, which indicates that an interrupt will be generated upon releasing the switch and will not be generated when the switch is pressed.

When a Port 54 interrupt is generated, the `DEMOSELECT_ISR` routine is executed. This routine increments the value of `ucDemoLightSeq`. The value of `ucDemoLightSeq` will be reset to its default value of 1 if it is equal to or greater than 5. Because the `ucDemoLightSeq` default value is 1, Light Sequence 1 is transmitted to the DMX Receiver when selecting Demo Mode.

A maximum of four sequences can be loaded onto the EEPROM via the PC application. These sequences are decoded from the EEPROM through the `DEMO_ReadLSfromEEPROM` routine before transmitting over the DMX interface.

The `DEMO_ReadLSfromEEPROM` routine uses the value of `ucDemoLightSeq` to determine the light sequence that will be accessed from the EEPROM. If the value of `ucDemoLightSeq` is 2, `DEMO_ReadLSfromEEPROM` will access Light Sequence 2 with starting EEPROM address of `0x010000.`

Table 5 lists the address ranges for the light sequences on the EEPROM. The size of the EEPROM is 256KB thereby allowing each light sequence to access 65,536 locations. Storing DMX data for five channels on the EEPROM will be equivalent to a maximum of 3,854 entries in each light sequence.

Table 5. Light Sequence Address Range

Light Sequence	Address Range
1	0x000000–0x00FFFF
2	0x010000–0x01FFFF
3	0x020000–0x02FFFF
4	0x030000–0x03FFFF

Table 6 lists the EEPROM data format. The table lists light sequences, entries, and data. The EEPROM address locations are divided to handle four light sequences. The X variable in the Light Sequence column represents light sequences 1–4. Each light sequence consists of a number of entries. Each entry consists of time delay, channel, and channel data except for the first entry of each light sequence, which does not include a time delay.

Table 6. EEPROM Data Format

Light Sequence	Entry	Data
X	1	Channel m1
		Channel data m1
		Channel m2
		Channel data m2
		...
		...
		Channel mn
		Channel data mn
		Channel mn
		Channel data mn
	2	Time delay
		Channel m1
		Channel data m1
		Channel m2
		Channel data m2
		...
		...
		Channel mn
		Channel data mn
		Channel mn
		Channel data mn
	...	...
	...	...
	...	...
	n	Time delay
		Channel m1
		Channel data m1
		Channel m2
		Channel data m2
		...
		...
		Channel mn
		Channel data mn
		Channel mn
		Channel data mn

Table 6. EEPROM Data Format (Continued)

Light Sequence	Entry	Data
X	n	0xFF
		0xFF
		0xFF
		0xFF

The decoding process of the light sequence from the EEPROM by the `DEMO_ReadLSfromEEPROM` routine includes the following elements:

- Identification of the channel and the channel data
  - The channel variable contains two bytes of data and represents Channel 1–Channel 512
  - The channel data variable contains one byte of data and represents the DMX data value with a range of 0–255
- Identification of the end of an entry is indicated by a repeat of the previous channel and channel data values
- Identification of the end of a light sequence is indicated by four `0xFF` values
- Identification of a time delay is indicated by the end of an entry, after which the next byte of data indicates the time delay
  - A time delay defines how long the previous entry will continue to be transmitted to the DMX receiver. The DMX controller will start transmitting the current entry after this time delay expires.

A light sequence stored in EEPROM is not always destined for channels in the range 1 to n. As shown in Table 6, it is possible that Channel m1 represents Channel 8 and that Channel mn represents Channel 11. If a light sequence for Channels 8–11 is decoded by the `DEMO_ReadLSfromEEPROM` routine from EEPROM, a DMX data packet will be transmitted to the DMX512-A receiver, this data packet is illustrated in Figure 16.

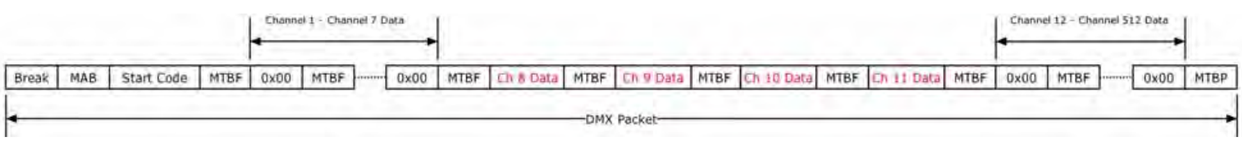

Figure 16. Demo Mode DMX Data Packet

DMX Transmit Routine

The DMX transmit routine includes the following two main parts which are executed each time a DMX transmit signal is requested:

- Transmission of a DMX preamble (i.e., a DMX reset sequence)
- DMX channel data

The DMX transmit line uses pin P20 of the Z51F3220 MCU. When transmitting a DMX preamble, this line is set to GPIO output. A DMX preamble requires transmitting Break and Mark After Break (MAB) signals. Timer1 is used for the timing of these signals.

After sending a DMX preamble to the DMX receiver, pin P20 is configured as `UART1 TX` and `DMX_PREAMBLE_TX` is set to the `ucDMX_States` flag. This action informs the `UART1_TX_ISR` routine that if the transmit buffer is empty, the `DMX_Data[]` buffer data can be transmitted to the DMX receiver.

After transmitting `DMX_Data[]`, `DMX_PREAMBLE_TX` is reset and `DMX_PACKET_TX` is set to the `ucDMX_States` byte flag. This action indicates that `DMX_Data[]` can be updated with new data before a DMX transmit signal is requested again.

`DMX_Data[]` is a RAM data buffer that contains 513 bytes: 1 byte for the START code and 512 bytes for the DMX channel data. `UART1_TX` data is configured to include 2 STOP bits. These 2 STOP bits represent the Mark Time Between Frames (MTBF). The Mark Time Between Packets (MTBP) is the time allotted to update `DMX_Data[]` before another DMX transmit routine.

PC Software Application

DMX Console is a Windows-based application designed to communicate with the DMX512-A controller via a USB interface. This section describes the DMX Console user interface. Figure 17 displays an image of a DMX Console main window.

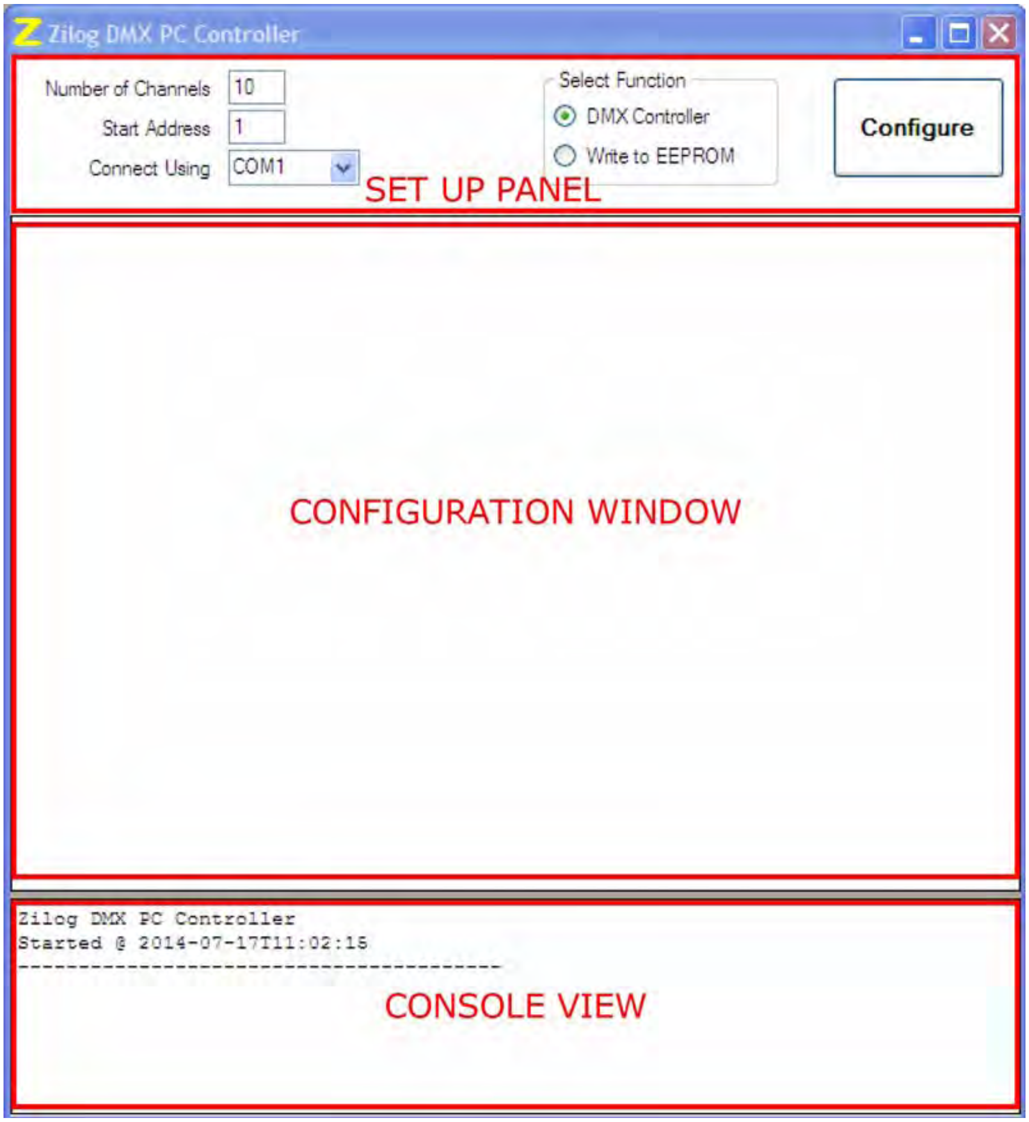

Figure 17. DMX Console Main Window

Setup Panel

The Setup Panel handles the DMX setup and determines the type of operation to use. The parameters specified in this panel reflect the DMX system that the DMX512-A controller operates on. Figure 18 shows the fields on the Setup Panel.

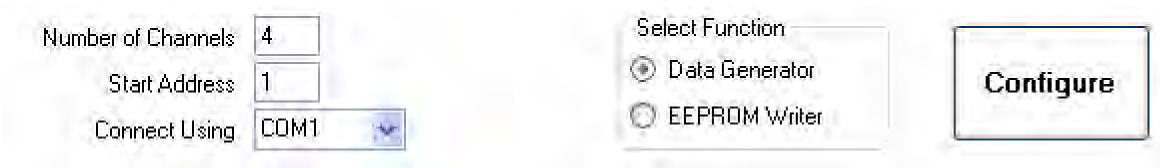

Figure 18. Setup Panel

Number of Channels

Enter the number of DMX channels to be controlled by this application in the Number of Channels field. Valid values range from 1–512. The default value is 4.

Start Address

Specify the address of the first DMX channel present in the DMX system in the Start Address field. Valid values range from 1–512. The default value is 1.

Note: The values indicated in the Number of Channels and Start Address fields must be within the 512 addresses available in the DMX protocol. For example, if the Number of Channels value is 4 and the Start Address value is 5, the resulting DMX channels to be used will be 5, 6, 7, and 8.

Connect Using

Select the serial COM number in which the DMX512-A controller is connected to the PC. This dropdown menu displays all available ports in the machine.

Select Function

Select the type of operation to perform from the following two options:

- Data Generator: Issues a TRANSMIT command to the DMX512-A controller with the data specified in the sliders. When this item is selected, the DMX configuration panel is displayed when the Configure button is clicked.
- EEPROM Writer: Issues a SAVE command to the DMX512-A controller with the EEPROM entries specified in the EEP configuration panel. When this item is selected, the EEP configuration panel is displayed when the Configure button is clicked.

Configure Button

Click the Configure button to display the configuration panel and to start communication with the DMX512-A controller.

Note: If the configuration panel is not displayed after clicking the Configure button, go to Console View to check if any errors or invalid input values are listed.

Configuration Panel

The configuration panel appears in one of two forms, DMX configuration panel or EEP configuration panel, depending on the option selected in the Select Function parameter.

DMX Configuration Panel

The DMX configuration panel handles the constant update of values for each DMX channel connected to the DMX512-A controller. Each slider represents a DMX channel value. All settings specified in this panel can be saved in a file for later use.

There are two update methods available: Auto-Fade update and Real-Time update. If the Auto-Fade update method is selected, the application automatically updates and sends DMX values to the DMX512-A controller. If the Real-Time update method is selected, the application sends DMX values to the DMX512-A controller each time a slider value changes.

This section describes the fields on the DMX configuration panel, which is shown in Figure 19.

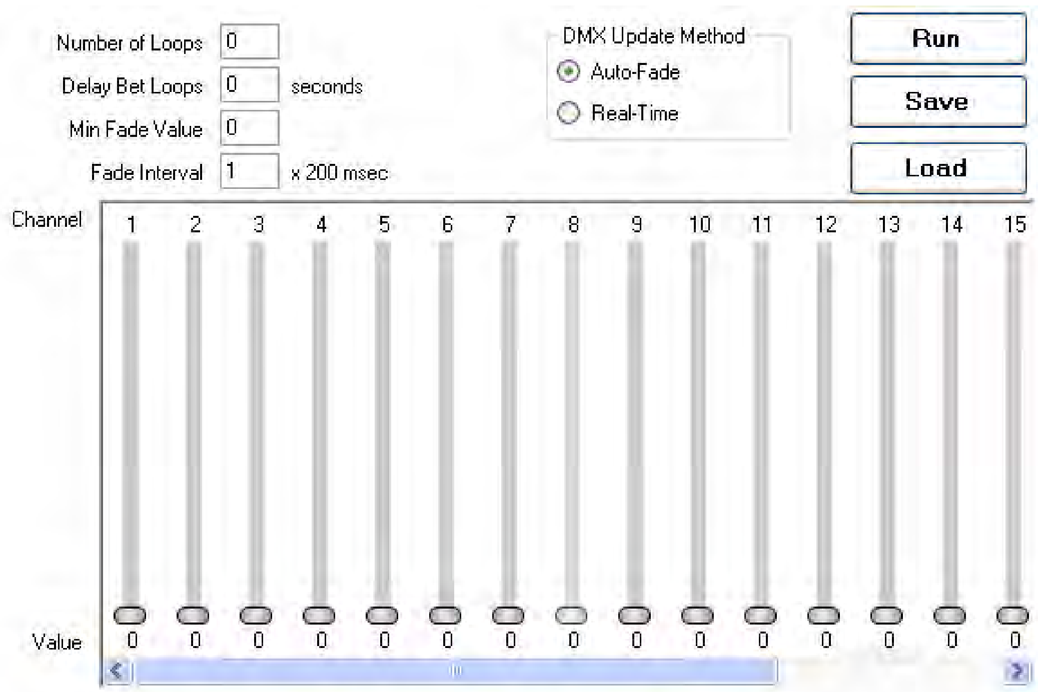

Figure 19. DMX Configuration Panel

DMX Channels/Sliders

The sliders represent the values of the DMX channels associated with them. The DMX channel number associated with a slider is shown at the top of the slider while its current value is shown below it. If the Auto-Fade update method is selected, these sliders update automatically after the Run button is clicked, to reflect the current DMX values sent to the DMX512-A Controller.

DMX Update Method

Select an update method to use for this operation from the following two options:

- Auto-Fade: DMX values are automatically updated and sent to the DMX512-A controller at a specified time interval.
- Real-Time: DMX values are automaticaly sent to the DMX512-A controller each time the slider's value changes.

Run Button

Click the **Run** button to start an Auto-Update sequence. This button is not available for the Real-Time update method.

Save Button

Click the **Save** button to save current settings to a file.

Load Button

Click the **Load** button to load previously saved settings from a file.

Auto-Fade Update Method

In the Auto-Fade update method, DMX values are automatically updated and sent to the DMX512-A controller. The update interval is defined as the fade interval value x 200 ms. The starting values for each DMX channel are defined by the sliders, which are decremented during the update interval. When all DMX values reach the minimum fade value, the values are incremented at every update interval until they reach the original starting values. This sequence completes one loop.

Number of Loops

The value in the Number of Loops field determines the maximum number of times the fade sequence loops to its original/starting value. Specify 0 to disable counting of loops and to enable continuous looping.

Delay between Loops

The Delay between Loops value determines the time delay between each loop, before the DMX values start to decrement again. Specify 0 to disable delay between loops.

Minimum Fade Value

The number in the Minimum Fade Value field determines the minimum value at which DMX values will stop decrementing.

Fade Interval

The number entered in the Fade Interval field helps to determine the update interval of DMX values, in which the actual update interval is defined as the Fade Interval value x 200 ms.

Real-Time Update Method

In the Real-Time update method, DMX values are automatically updated each time a slider value changes.

EEP Configuration Panel

The EEP configuration panel aids in configuring and loading of sequences to the DMX512-A controller's on-board EEPROM. It connects directly to the DMX512-A controller via the serial COM port specified in the Setup Panel. The EEP configuration panel includes the EEPROM Entry List window, which contains a list of EEPROM entries. The values of the sliders represent the values of their associated DMX channel. This section describes the fields on the EEP configuration panel, which is shown in Figure 20.

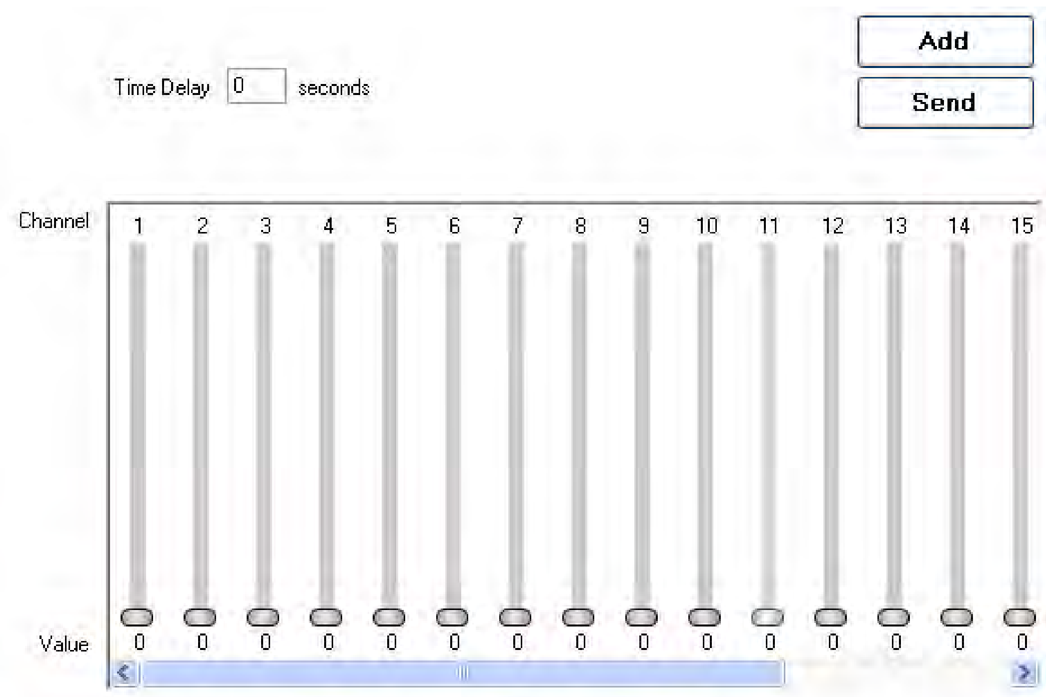

Figure 20. EEP Configuration Panel

Time Delay

The value in the Time Delay field determines the time delay parameter defined in the EEPROM entry format. This value is valid only for EEPROM entries starting from #2. There is no time delay parameter for EEPROM entry #1.

DMX Channels/Sliders

The sliders represent the values of the DMX channels associated with them. The DMX channel number associated with a slider is shown at the top and its current value is shown below it.

Add Button

Click the Add button to create an EEPROM entry, with the current DMX values and time delay value as its parameters. The resulting EEPROM entry is reflected in the EEPROM Entry List window.

Send Button

Click the Send button to send all EEPROM entries listed in the EEPROM Entry List window to the DMX512-A controller.

EEPROM Entry List Window

The EEPROM Entry List window shows a list of all EEPROM entries that are included in the selected sequence number. Each EEPROM entry and its associated sequence number can be saved to a file for later use. This section describes the fields on the EEPROM Entry List window, which is shown in Figure 21.

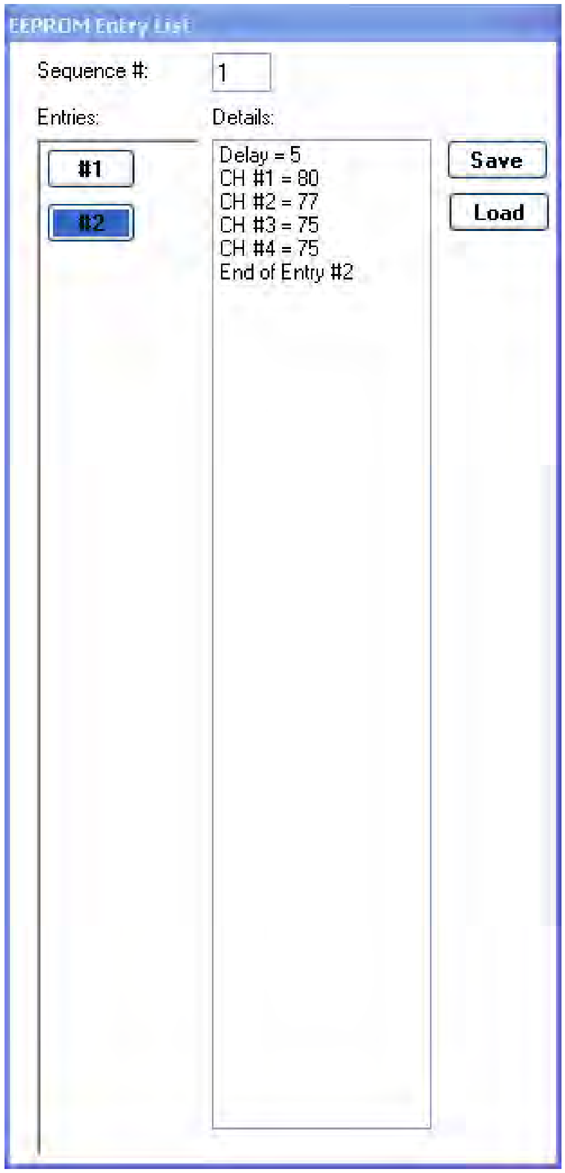

Figure 21. EEPROM Entry List Window

Sequence

The value in the Sequence # field shows the sequence number that the current EEPROM entry is included in. Valid values range from 1–4.

EEPROM Entries List

EEPROM entries that are currently included in the sequence are displayed. If there are no entries listed, this list is blank. When an EEPROM entry is added to the sequence, the list is populated with an entry number button.

EEPROM Entry Buttons

Click on an EEPROM entry button to view its details. The highlighted button indicates that its contents are currently on display in the EEPROM Details panel.

EEPROM Entry Details

The EEPROM entry details panel shows the contents of the EEPROM entry currently selected in the EEPROM entries list.

Save Button

Click the **Save** button to save the EEPROM entries to a file.

Load Button

Click the Load button to load the EEPROM entries from a previously saved file.

Console View

The console view is a display of event logs such as when the system started or when a program encountered an error. It also displays the DMX data bytes sent through the serial COM port. Figure 22 shows a sample console view.

```
Zilog DMX PC Controller
Started @ 2013-09-06T11:42:38
----------------------------------------
Number of Channels Please enter values from 1 - 512.
Number of Channels Please enter values from 1 - 512.
Number of DMX channels cannot exceed 512.
```

Figure 22. Console View

Testing

This DMX512-A Controller works with the DMX512-A Receiver (RD0015) and DMX-Pro Network Checker, available at http://www.bpesolutions.com/dmxproduct.html#anchor1138975.

Software Installation and Programming the Z51F3220 MCU

For instructions on installing the Z8051 OCD software and driver; using Keil µVision IDE; and building, debugging, and executing the Z51F3220 project, refer to the Z8051 Tools Product User Guide (PUG0033) and the Z51F3220 Development Kit User Manual (UM0243).

Enable RESET Input of the Z51F3220 MCU through Keil µVision IDE

In this Reference Design, Pin 1 of the Z51F3220 MCU is used as the RESET input. To enable the RESET input, observe the following procedure prior to building this RD0016_SC01 project.

1. From the Project menu of Keil µVison IDE, click to open the **Options for Target 'RD0016'** dialog.

2. Click the **Utilities** tab.
3. In the **Use Target Driver for Flash Programming** drop-down menu within the **Configure Flash Menu Command** pane; ensure that **Zilog Z8051 Target Driver** is selected.
4. Click **Settings**.
5. Select the **Flash Options** tab.
6. Within the **Configuration Options** pane, select or check the **Enable/Reset input** box.
7. Click **OK** to exit the **Settings** dialog.
8. Click **OK** to exit the **Options for Target 'RD0016'** dialog.

DMX512-A Controller Testing Procedures

The DMX512-A controller can be tested using the DMX-Pro network checker or the DMX512-A receiver. The testing procedures for both options are described in this section.

Using the DMX-Pro Network Checker, Local Mode

Observe the following procedure to test the DMX512-A controller using the DMX-Pro network checker.

1. Set the DMX512-A controller to Local Mode through the DIP switch. Set S1 Position2 to ON and Position1 and Position3 to OFF.
2. Set the DMX-Pro network checker to RX mode.
3. Connect a DMX cable to the DMX512-A controller (XLR connector) and the DMX-Pro network checker (DMX-IN).
4. Apply power to the DMX512-A controller and the DMX-Pro network checker.
5. The DMX-Pro network checker can only display a single channel. To view the active channel, press the + or – **Chan** button. Select Channel 1.
6. On the DMX512-A controller, adjust the ANA0 potentiometer. This action will change the DMX value, ranging from 000–255, displayed on the DMX-Pro network checker. Figure 23 shows a sample DMX value on the DMX-Pro network checker.

Figure 23. DMX-Pro Network Checker

7. To check the values of other channels, repeat Steps 5 and 6.

Table 7 lists the DMX-Pro network checker channel and its source of value (POTs of DMX controller).

Table 7. DMX-Pro Network Checker Channel Data Source

DMX Controller Potentiometers	DMX-Pro Network Checker Channels
ANA0	Channel 1
ANA1	Channel 2
ANA2	Channel 3
ANA3	Channel 4
ANA4	Channel 5

Using the DMX512-A Receiver

The DMX512-A controller can be tested using multiple modes of the DMX512-A Receive. This section describes the testing procedure for each mode.

Local Mode

1. Set the DMX512-A controller to Local Mode through the DIP switch. Set S1 Position2 to ON and Position1 and Position3 to OFF.
2. Set the DMX512-A receiver base address to 1. To learn more about setting the base address, refer to A DMX512-A Receiver Using a Z51F0811 MCU Reference Design (RD0015).

3. Connect a DMX cable to the DMX512-A controller and the DMX512-A receiver.
4. Apply power to the DMX512-A controller and the DMX512-A receiver.
5. Adjust the potentiometers on the DMX512-A controller and observe the LEDs on the DMX512-A receiver. The light intensity of the LEDs varies as the potentiometers are adjusted. Table 8 lists the pairing of potentiometers and LEDs.

Table 8. DMX Receiver LEDs Data Source

DMX Controller Potentiometers	DMX Receiver LEDs
ANA0	LED2 (Red)
ANA1	LED3 (Green)
ANA2	LED4 (Blue)
ANA3	LED5 (White)

PC Mode, DMX Transmit

1. Set the DMX512-A controller to PC Mode through the DIP switch. Set S1 Position3 to ON and Position1 and Position2 to OFF.
2. Set the base address of the DMX512-A receiver to 1. To learn more about setting the base address, refer to A DMX512-A Receiver Using a Z51F0811 MCU Reference Design (RD0015).
3. Connect a DMX cable to the DMX512-A controller and the DMX512-A receiver.
4. Connect a USB cable to the DMX512-A controller and the PC.
5. Apply power to the DMX512-A controller and the DMX512-A receiver.
6. Run the DMX console application on the PC.
7. Set up the DMX console as follows:
   - Number of Channels = 4
   - Start Address =1
   - Data Generator
   - Configure port and connect to the DMX controller

   To learn more about setting up the DMX console, refer to the PC Software Application section on page 17.
8. Start sending data and observe how the light intensity of the LEDs on the DMX512-A receiver varies.

PC Mode, Save to EEPROM

1. Set the DMX512-A controller to PC Mode through the DIP switch. Set S1 Position3 to ON and Position1 and Position2 to OFF.

2. Connect a USB cable to the DMX512-A controller and the PC.
3. Apply power to the DMX512-A controller.
4. Run the DMX console application on the PC.
5. Start saving light sequences to EEPROM. Refer to the PC Software Application section on page 17.
6. After saving a light sequence to EEPROM, reset the DMX board and the DMX console application to prepare them for the next light sequence.

Demo Mode

Note: Ensure that light sequences are stored in EEPROM before you begin the testing procedure in Demo Mode.

1. Set the DMX512-A controller to Demo Mode through the DIP switch. Set S1 Position3 to ON and Position1 and Position2 to OFF.
2. Set the base address of the DMX512-A receiver. To learn more about setting the base address, refer to A DMX512-A Receiver Using a Z51F0811 MCU Reference Design (RD0015). The base address is also dependent on the setup made on the DMX console before saving the light sequences to EEPROM.
3. Connect a DMX cable to the DMX512-A controller and the DMX512-A receiver.
4. Apply power to the DMX512-A controller and the DMX512-A receiver.
5. Observe how the light intensity of the LEDs on the DMX512-A receiver varies. This default sequence is Light Sequence 1.
6. Press the Demo Select pushbutton switch on the DMX512-A controller to change the light sequence.

Electrical Specifications

Table 9 shows the specifications of the DMX512-A controller reference design. All values are typical.

Table 9. Electrical Specifications for the DMX512-A Controller Reference Design

Parameter	Values	Units
Power Input (External)	7–12	V
Current Consumption	54	mA
DMX Packet Timing:		
Break	88.11	µs
Mark After Break	11.54	µs
Data (Start Code and Channel Data)	44	µs

Software Implementation

The source code for this application, RD0016-SC01.zip, is available free for download from the Zilog website.

Equipment Used

The following equipment was used to build and test this reference design:

- Fluke 87V multimeter
- Intronix logic analyzer
- DMX-Pro Network Checker (http://www.bpesolutions.com/dmxproduct.html#anchor1883330)

For a list of materials used to build this reference design, see Appendix C. Bill of Materials on page 49.

Kit Contents

The DMX Reference Design Kit contains the following items:

- DMX Receiver Printed Circuit Board (PCB)
- DMX Controller PCB
- USB A-Male to Mini B-Male 2 m cable
- 9V 2A wall power adapter
- Hosa XLR (Male-Female) 10 feet cable
- DMX Kit Insert (FL0175)

Ordering Information

The products associated with this DMX512-A Controller Using the Z51F3220 MCU Reference Design (RD0016) and the DMX512-A Receiver Using a Z51F0811 MCU Reference Design (RD0015) are available as a single DMX Reference Design Kit. This kit can be ordered from the Zilog Store, using the part number listed in Table 10.

Table 10. DMX512-A Controller Ordering Information

Part Number	Description	Store Product ID
Z51DMX00100KITG	DMX Reference Design Kit	RD10037

Results

This DMX512-A Controller reference design was successfully tested with a commercially available DMX product (DMX-Pro Network Checker) and with Zilog's DMX512-A Receiver (RD0015).

This reference design is based on established DMX standards which are listed in Table 11. The duration of the components of a DMX Packet, as prescribed by the standards, was achieved. For more information about the components of a DMX Packet, including duration, refer to Table 4.

The DMX512-A Controller board is designed with standard DMX connectors to interface seamlessly with a DMX universe. An opto-isolator between Zilog's Z51F3220 MCU and MAX485 (DMX output) is added for safety and robustness.

Summary

This reference design describes the implementation of a DMX512-A controller using Zilog's Z51F3220 MCU, and includes hardware, firmware, and DMX PC controller software.

The hardware includes the following components:

- Zilog's Z51F3220 MCU with debug interface
- DMX output interface (optoisolator and MAX485)
- PC interface (FT232RL serial to USB)
- Demo code storage (M95M02 SPI EEPROM)
- Local Mode potentiometers
- Power supply (external 7–12V power and USB)

Core functions of the firmware include `DMX_PREAMBLE_TX` and `UART1_TX_ISR`. These functions transmit the DMX data packet to a DMX universe.

The DMX PC controller software is available for free. However, the source code is not provided.

References

Documents associated with the DMX512-A controller are listed in Table 11. Each of the linked documents can be obtained from the Zilog website by clicking the link associated with its Document Number.

Table 11. DMX512-A Controller Documentation

Document Number	Description
PS0299	Z51F3220 Product Specification
PUG0033	Z8051 Tools Product User Guide
RD0015	DMX512-A Receiver Reference Design
RD0016	This DMX512-A Controller Reference Design document
UM0243	Z51F3220 Development Kit User Manual
	DMX 101 Handbook. http://elationlighting.com/pdffiles/dmx-101-handbook.pdf
	DMX-Pro Network Checker Manual. http://www.bpesolutions.com/bpemanuals/NewDMX.Checker.pdf
	Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories. Entertainment Services and Technology Association (ESTA) American National Standard E1.11; 2008. Entertainment Technology USITT DMX512-A.

Appendix A. Flow Charts

Figure 24 presents the flow of the main routine.

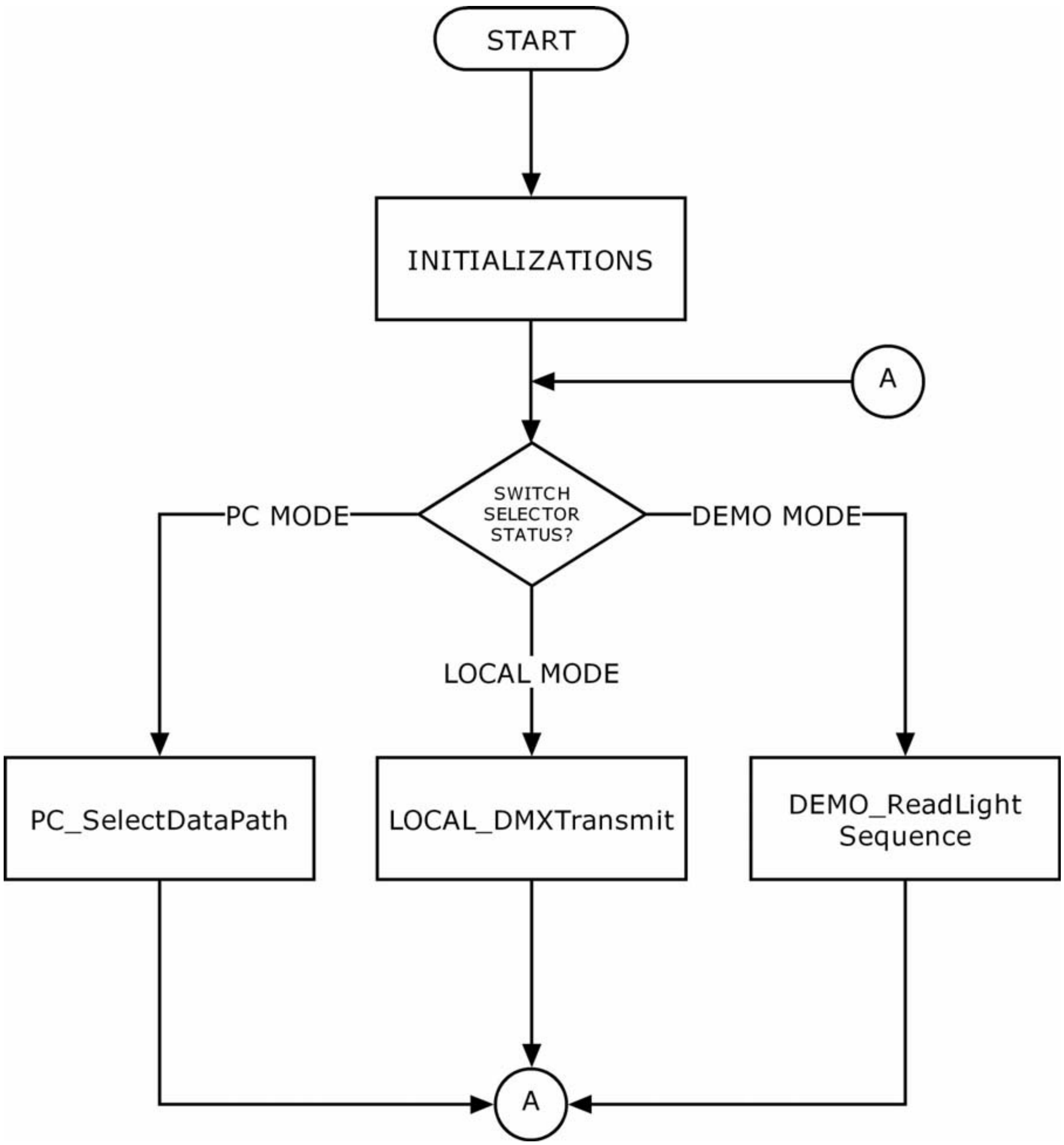

Figure 24. Main Flowchart

Figures 25 through 27 present the flows of the Local Mode routines.

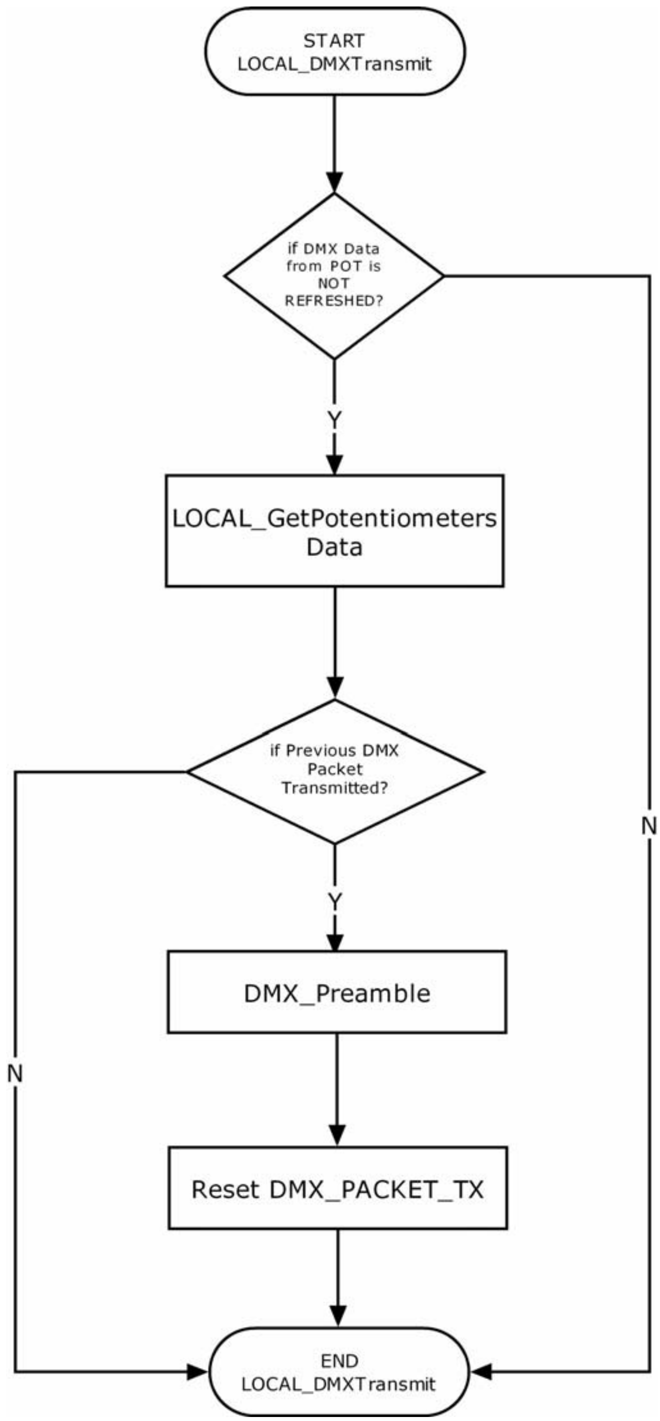

Figure 25. Local Mode Flowchart

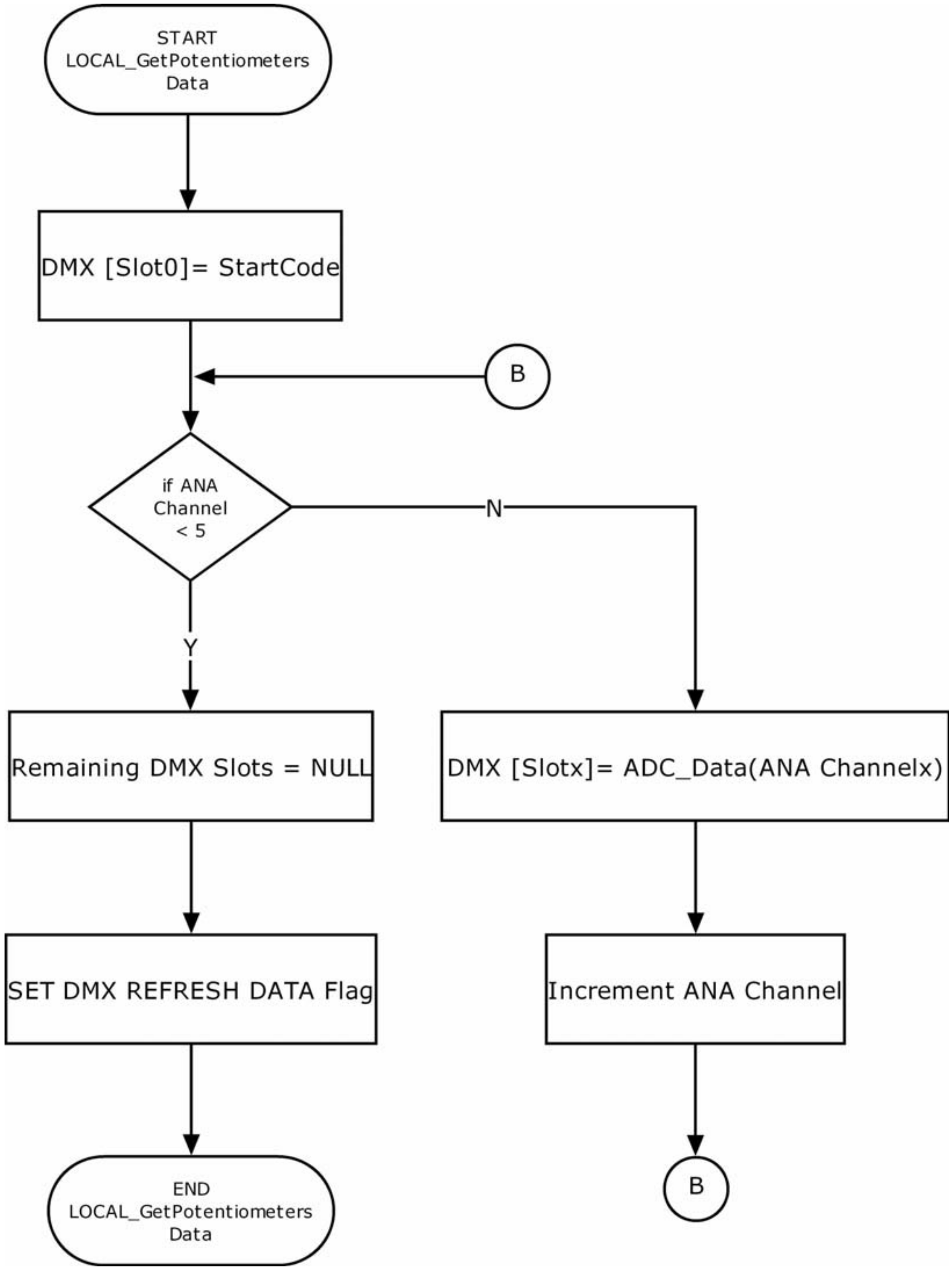

Figure 26. Updating the DMX Data Buffer with the Potentiometer's Data Flowchart

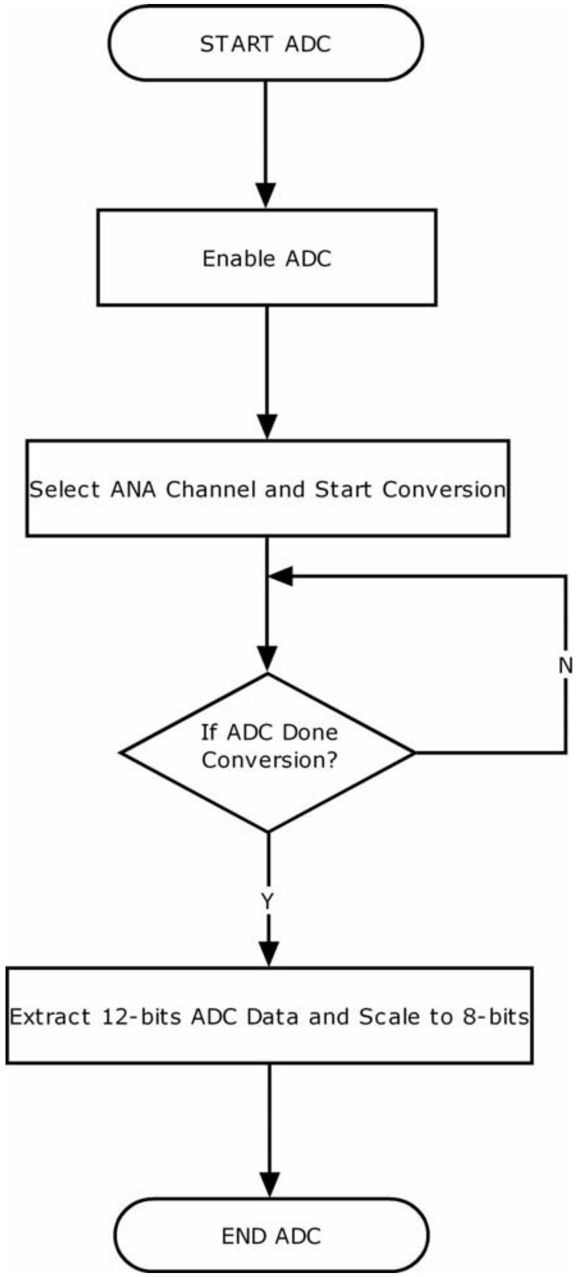

Figure 27. ADC Flowchart

Figures 28 through 32 present the flow of the PC Mode routines.

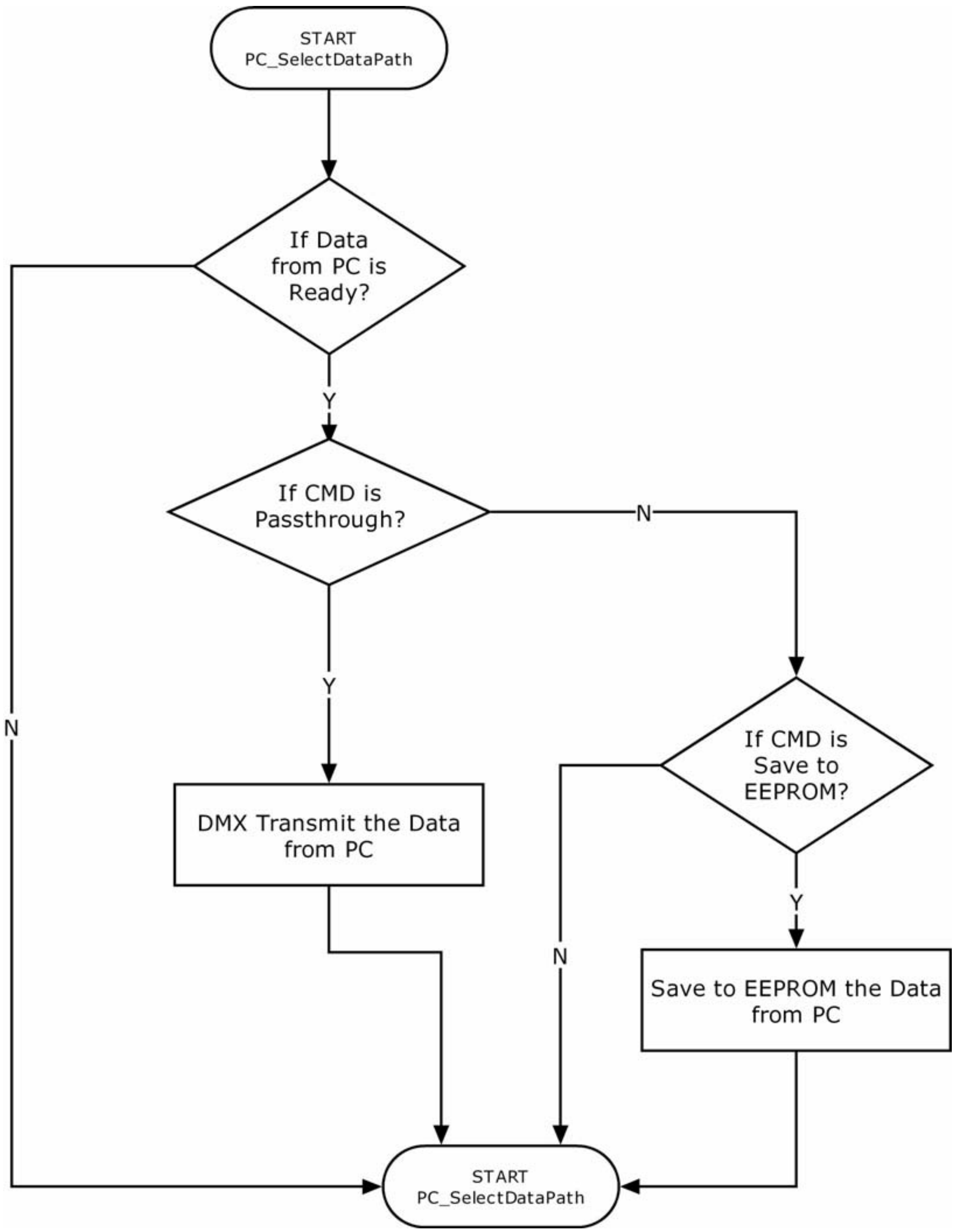

Figure 28. PC Mode Flowchart

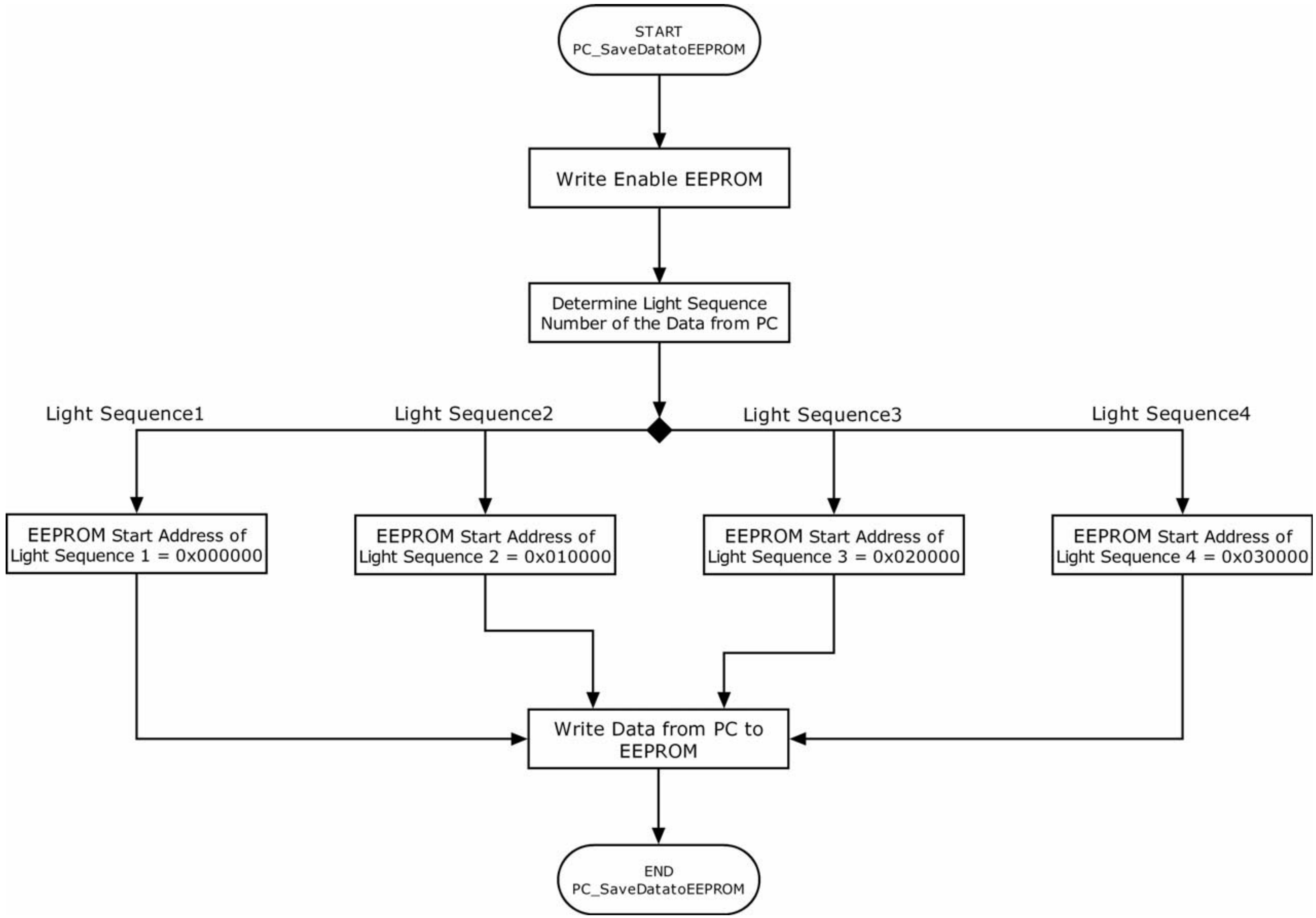

Figure 29. Saving Light Sequence to EEPROM Flowchart

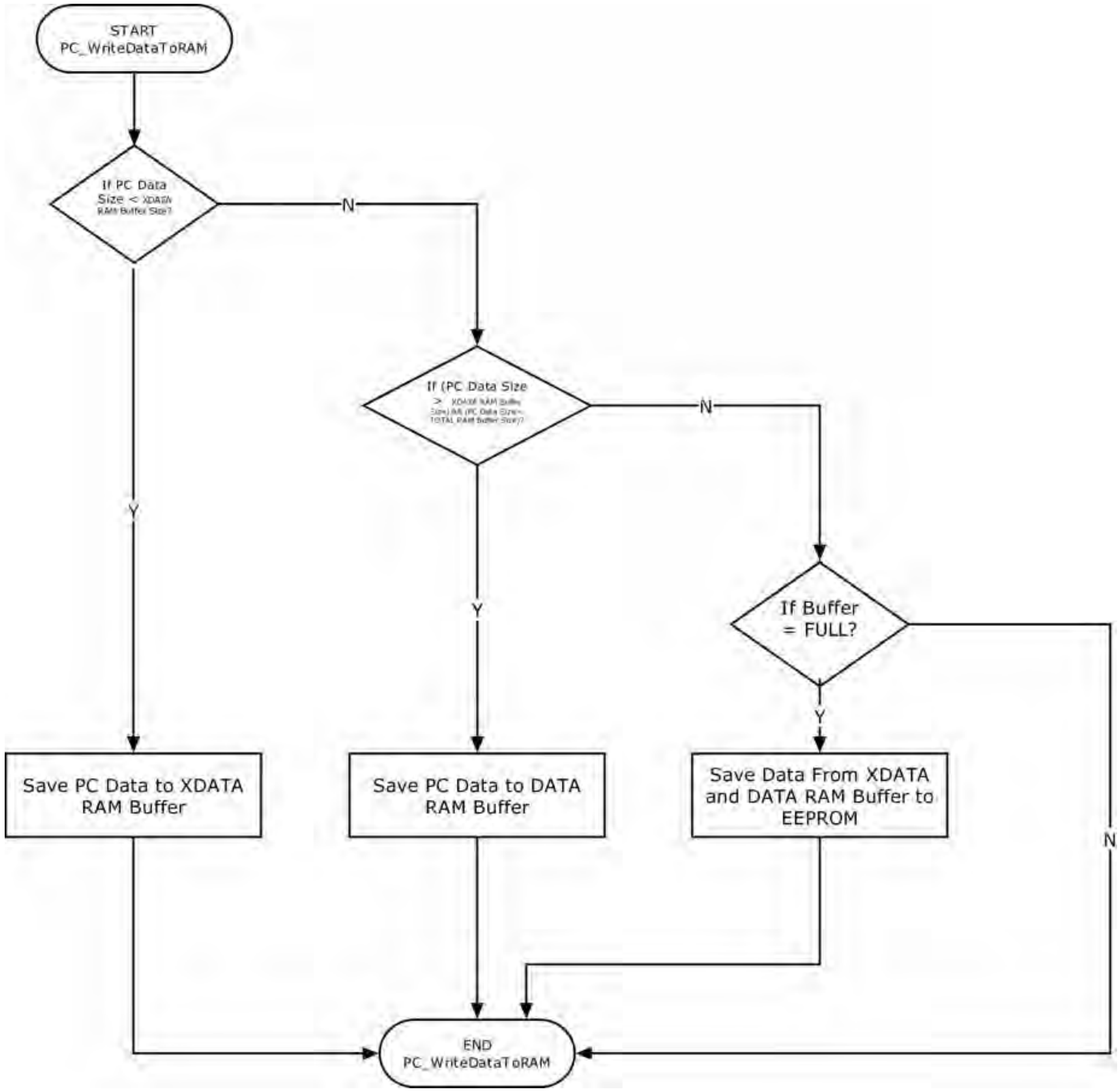

Figure 30. Saving PC Data to RAM Flowchart

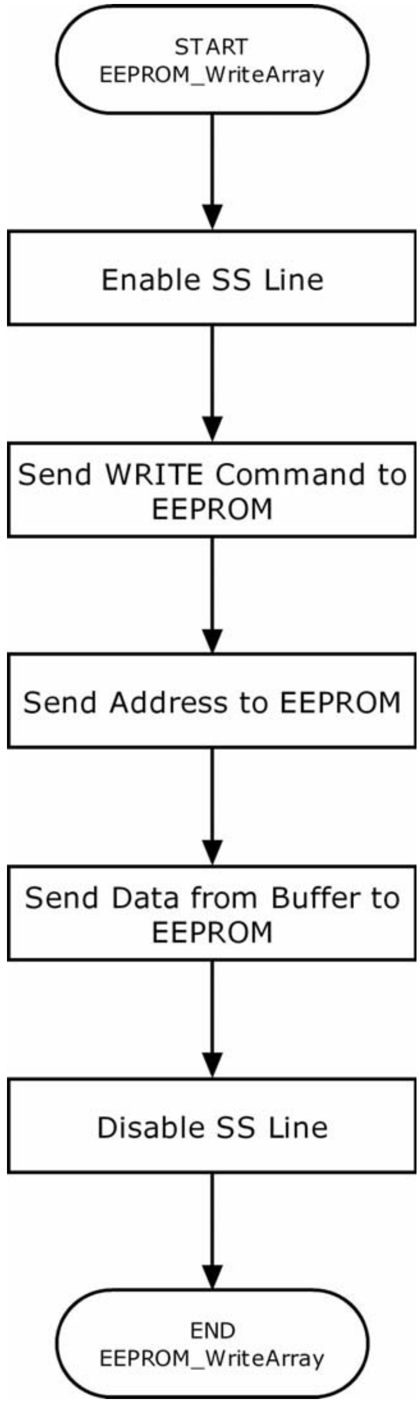

Figure 31. Saving Data from Buffer to EEPROM Flowchart

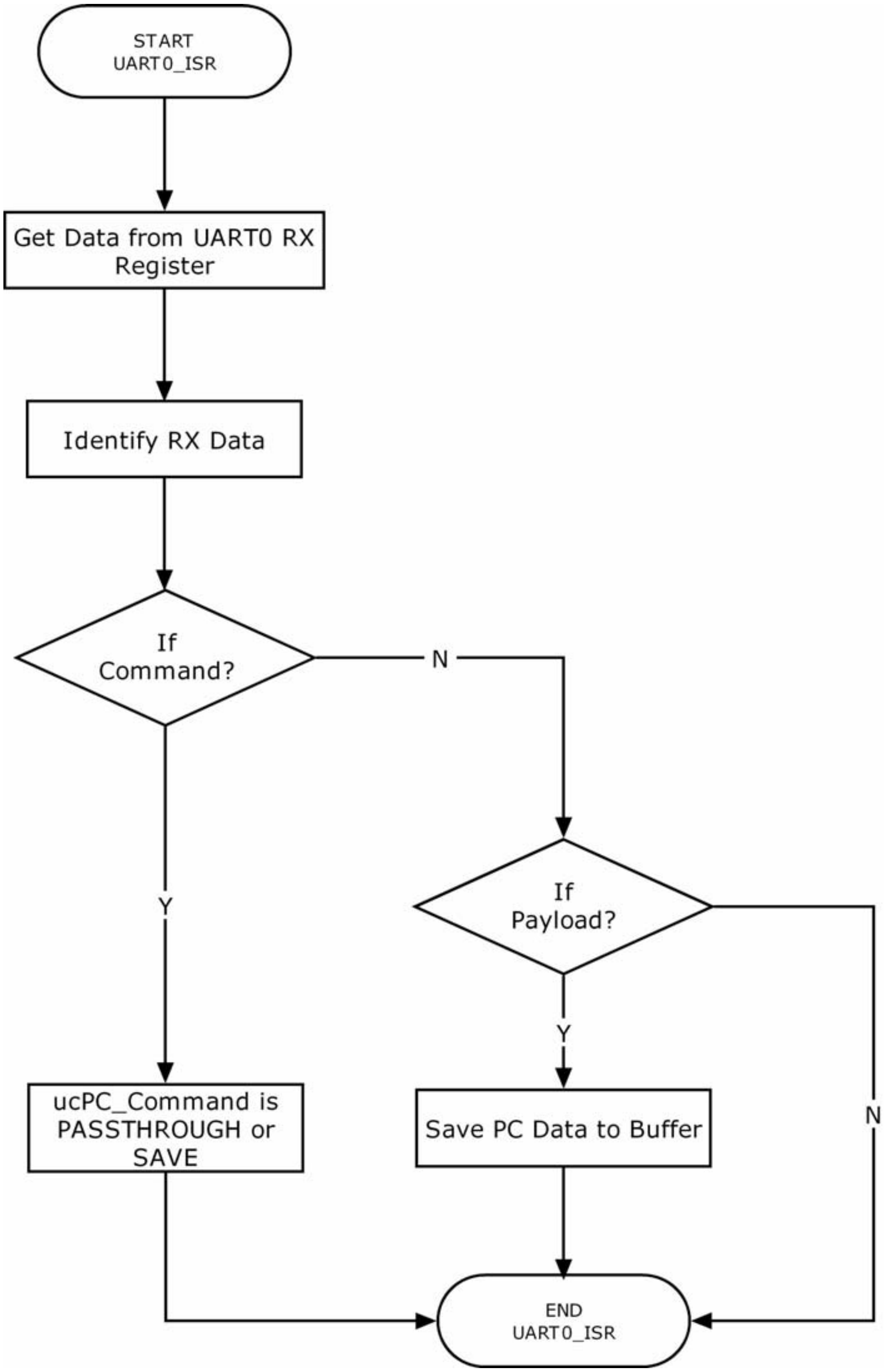

Figure 32. Receiving PC Data (UART0 ISR) Flowchart

Figures 33 through 36 display the flow of the Demo Mode routines.

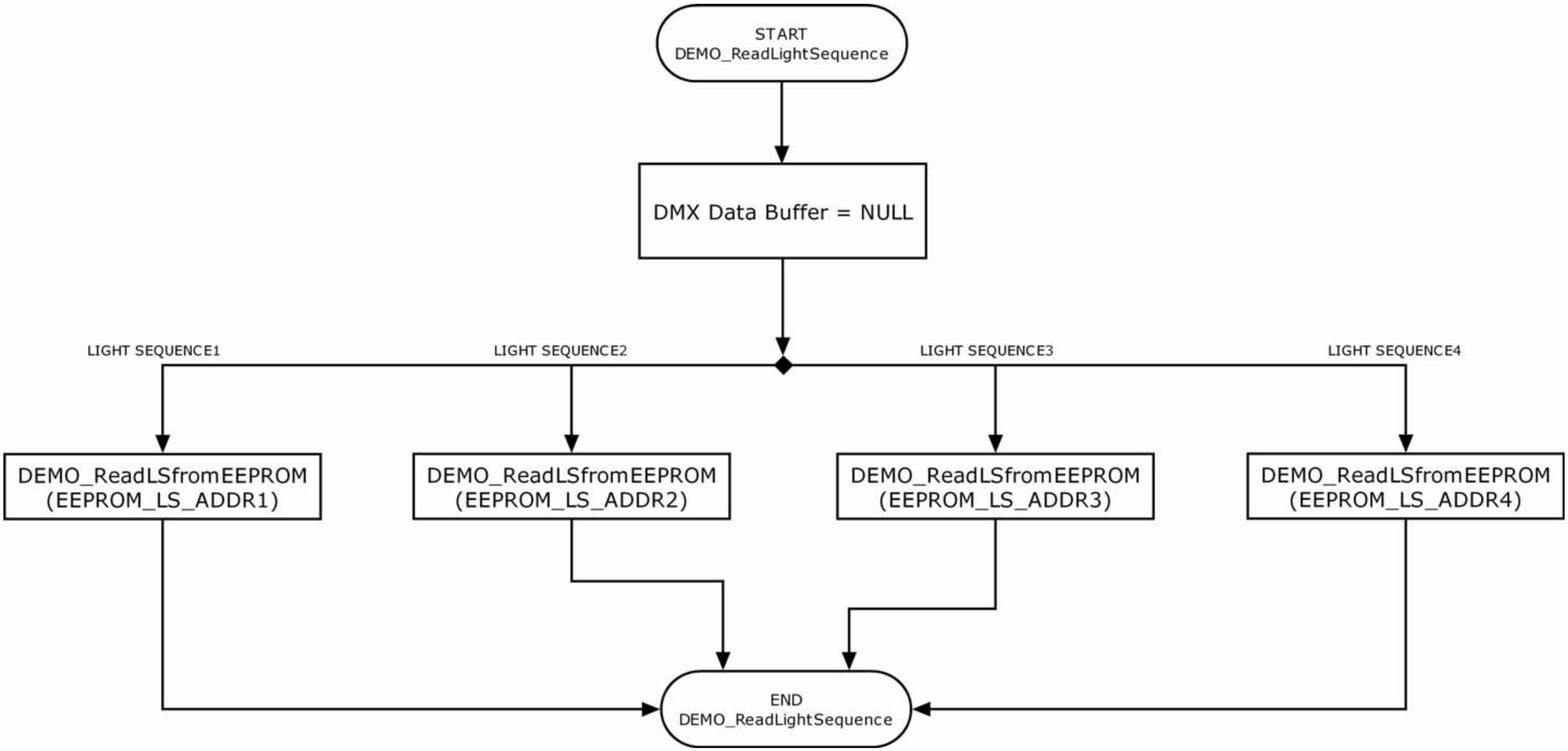

Figure 33. Demo Mode Flowchart

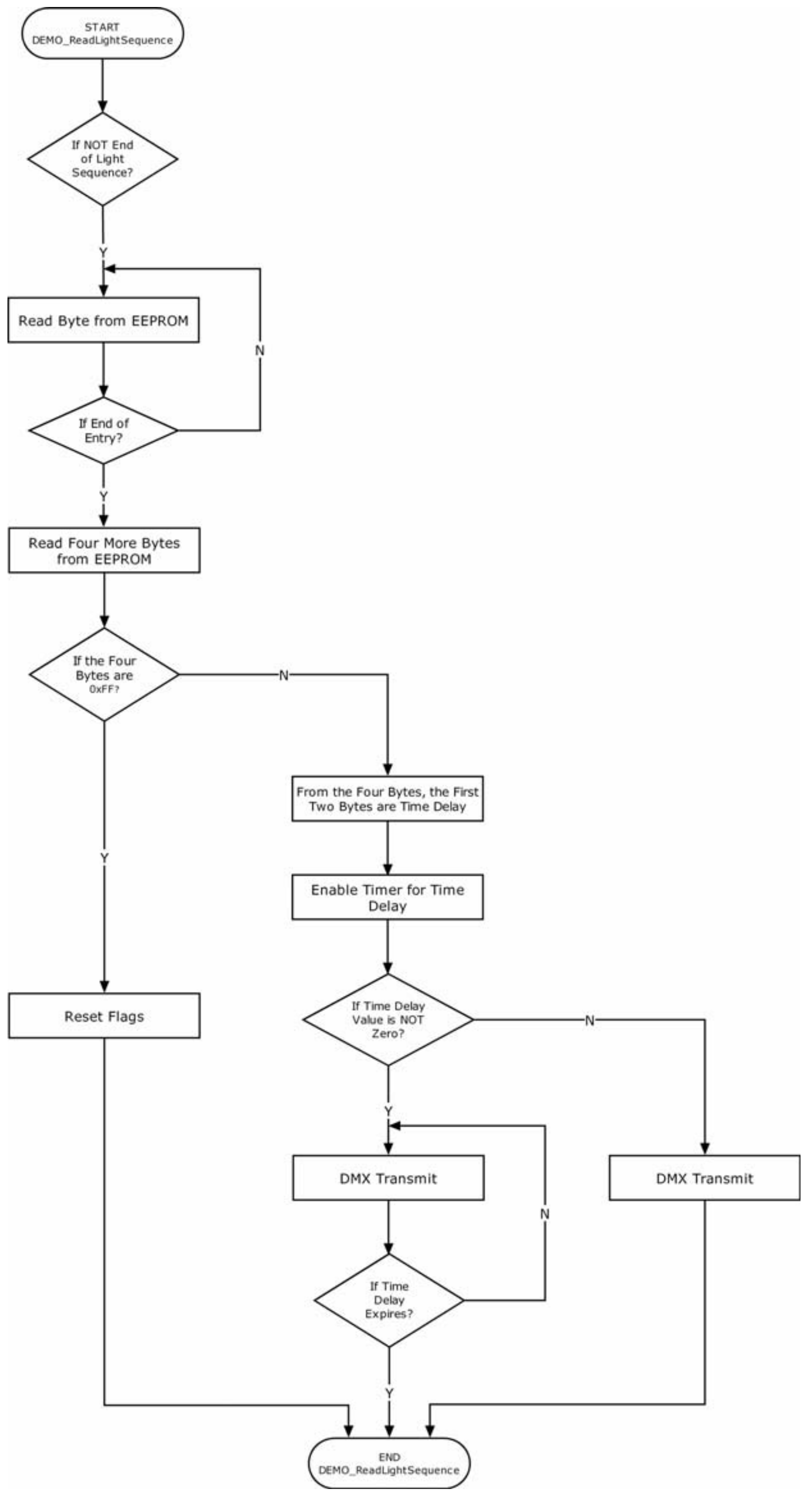

Figure 34. Reading Light Sequence Flowchart

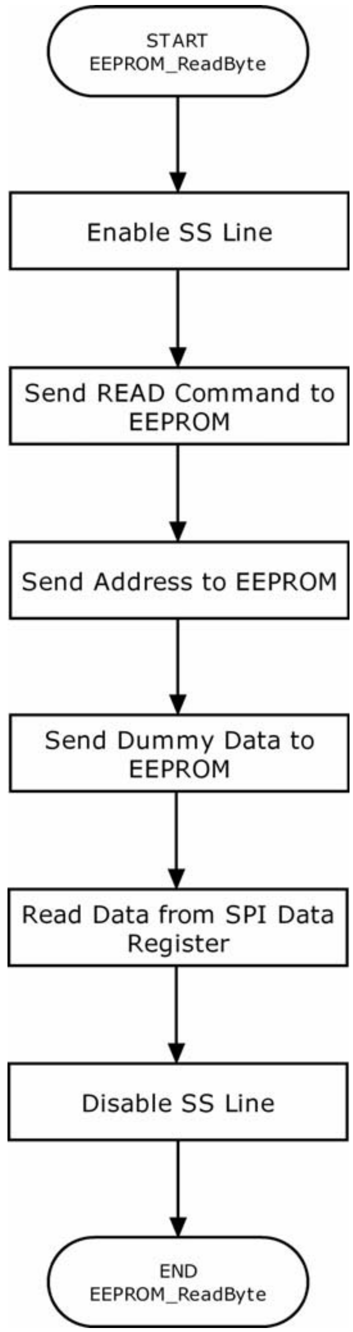

Figure 35. Read Data from EEPROM Flowchart

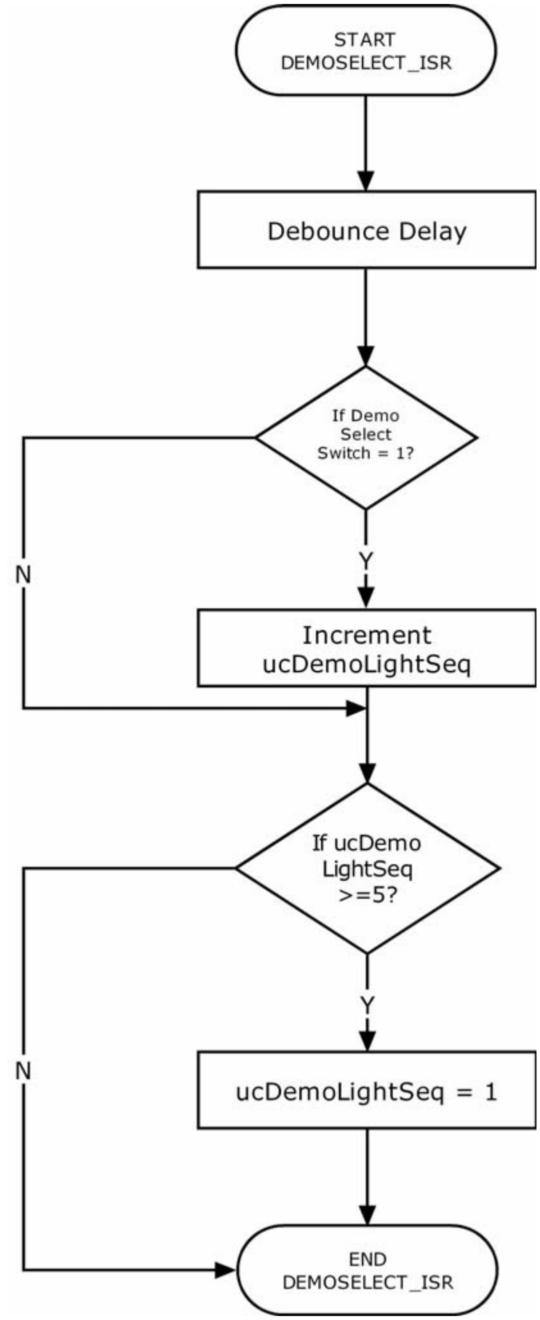

Figure 36. Demo Select Pushbutton Switch Interrupt (DEMOSELECT_ISR) Flowchart

Figures 37 and 38 present the DMX Transmit routine flows.

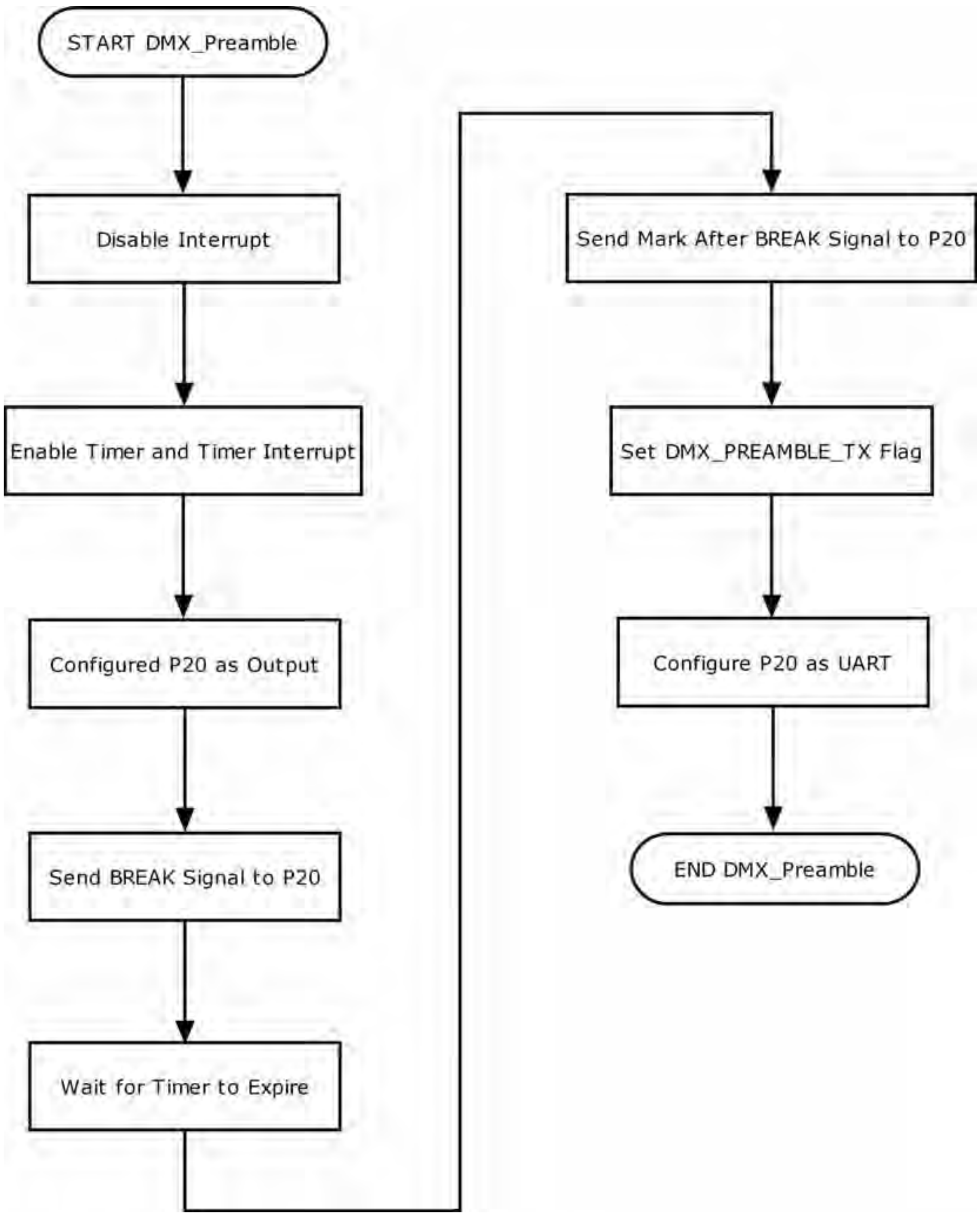

Figure 37. DMX Preamble Flowchart

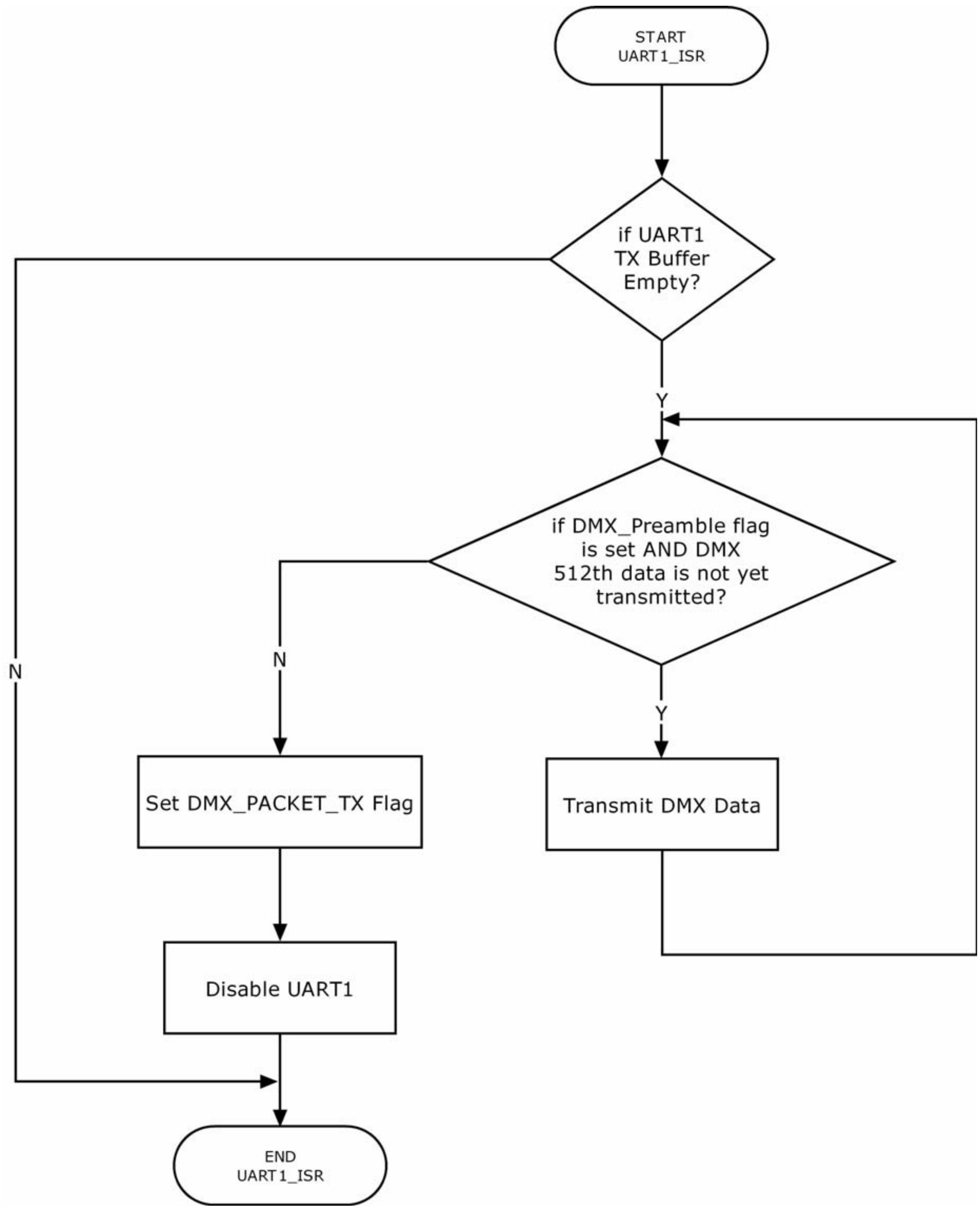

Figure 38. : DMX Data Transmit (UART1_ISR) Flowchart

Appendix B. Schematic Diagram

Figure 39 shows a schematic diagram of a DMX512-A controller.

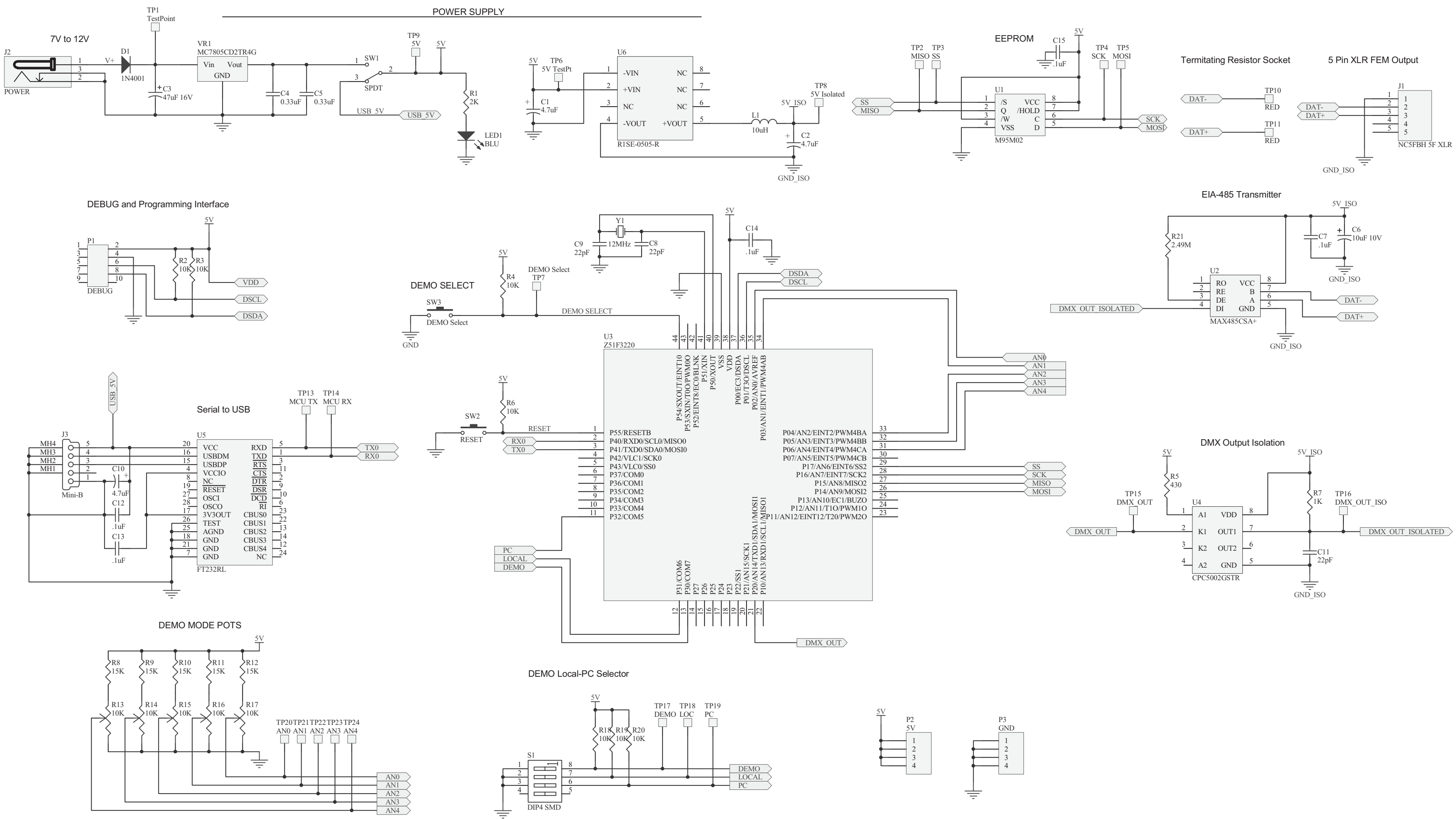

Figure 39. DMX512-A Controller Schematic Diagram

Appendix C. Bill of Materials

Table 12 lists the components used to build this reference design.

Table 12. DMX512-A Controller Bill of Materials

Item #	Designator	Value	Footprint	Quantity	Description	Manufacturer	Mfg. Part Number
1	C1, C2, C10	4.7uF	CAP 1206	3	CAP 4.7uF 16V A	AVX	TAJA475M016RNJ
2	C3	47uF 16V	CAP 5MM	1	Poly Alum 16V	Nichicon	PCG1C470MCL1GS
3	C4, C5	0.33uF	0603	2	CAP CER 0.33UF 25V 10% X5R	TDK	C1608X5R1E334K080AC
4	C6	10uF 10V	B	1	CAP 10uF 10V 10% B	AVX	TAJB106K010RNJ
5	C7, C12, C13, C14, C15	.1uF	0603	5	CAP CER 25V 0603	Murata	GRM188F51E104ZA01D
6	C8, C9, C11	22pF	0603	3	CAP CER 5% 50V	Murata	GRM1885C1H220JA01D
7	D1	1N4001	DO-221	1	Diode 1A 50V	Micro Commercial	SM4001PL-TP
8	J1	NC5FBH 5F XLR	NC5F XLR	1	XLR Connectors 5C PCB FEMALE RECEPT	Neutrik	NC5FBH
9	J2	POWER	JACKPWR 2.1x5.5MM	1	CONN POWER JACK 2.1x5.5 MM	PJ-002A	PJ-002A
10	J3	Mini-B	USB_MINI_B	1	CONN RECPT USB MiniB	Hirose	UX60-MB-5ST
11	L1	10uH	1206	1	100 Ohm 3A	Lair Signal Integrity	HI1206N101R-10
12	LED1	BLU	LED0603	1	LED BLUE 0603 2.9V 5mA	Panasonic	LNJ937W8CRA
13	MH#1, MH#2, MH#3, MH#4	#4	MH#4	4	MTG HOLE #4	Keystone Elect	2025
14	P1	DEBUG	HDR5X2 M	1	Header Male 10 Pin	FCI	67997-210HLF
15	P2	5V	HDR1X4	1	Header, 4-Pin	FCI	68000-104HLF
16	P3	GND	HDR1X4	1	Header, 4-Pin	FCI	68000-104HLF
17	R1	2K	0603	1	RES 2.0K 1%	Yageo	RC0603FR-072KL
18	R2, R3, R4, R6, R18, R19, R20	10K	0603	7	RES 10K 1/10W 1% 0603	Yageo	RC0603FR-0710KL

Table 12. DMX512-A Controller Bill of Materials (Continued)

Item #	Designator	Value	Footprint	Quantity	Description	Manufacturer	Mfg. Part Number
19	R5	430	0603	1	RES 430 OHM 1/10W 1%	Panasonic	ERJ-3EKF4300V
20	R7	1K	0603	1	RES 1K 0603	Vishay	CRCW06031K00FKEA
21	R8, R9, R10, R11, R12	0	0603	5	RES 0 Ohm 0603	Yageo	CRCW06030000Z0EA
22	R13, R14, R15, R16, R17	10K	POT3386F	5	POT 10K THMBADJ	Bourns Inc	3386F-1-103TLF
23	R21	2.49M	0603	1	RES 2.49M OHM 1/10W 1%	Yageo	RC0603FR-072M49L
24	S1	DIP4 SMD	SW DIP4 SMD	1	SMALL SW	Copal Elect	CHS-04TA
25	SW1	SPDT	SW SPDT SLIDE	1	Switch Slide SPDT	E-SWITCH	EG1218
26	SW2	RESET	SW_PB2	1	Switch P10891S-ND, Alt EG2644-ND	Panasonic	EVQ-PAD07K
27	SW3	DEMO Select	SW P/B 1/2"D	1	SW P/B 1/2" D	C&K	D6R40 F1 LFS
28	TP10, TP11	RED	TEST POINT 063	2	Test Point	Keystone Elect	5011
29	U1	M95M02	SO8N	1	SPI_EEPROM 2MBIT 5MHZ	ST Micro	M95M02-DRMN6TP
30	U2	MAX485CSA+	SO8	1	IC TXRX RS485/RS422 LOWPWR	Maxim	MAX485CSA+
31	U3	Z51F3220	Z51F3220_MQFP 44	1	Z51F3220 44PQFN	Zilog	Z51F3220FNX
32	U4	CPC5002GSTR	SMP8	1	ISOLAT DGTL 3.75KVRMS 2CH	IXYS	CPC5002GSTR
33	U5	FT232RL	SSOP28	1	IC USB TO SERIAL UART SSOP28	FTDI	FT232RL-REEL
34	U6	R1SE-0505-R	RISE-0505-R	1	DC/DC CONVERTER 1W 5VIN 5VOUT	RECOM POWER	R1SE-0505-R
35	VR1	MC7805CD2TR4 G	D2PAK-3	1	IC REG LDO 5V 1A D2PAK	ON Semi	MC7805CD2TR4G
36	Y1	12MHz	Crystal 5x3.2mm	1	XTAL 12MHz	ABRACON	ABM3-12.000MHZ-B2-T
37	MH#1, MH#2, MH#3, MH#4	4-40x1/4"	Screw Phillips	4	Pan Head	B&F Fastener Supply	PMSSS 440 0025 PH

Customer Support

To share comments, get your technical questions answered, or report issues you may be experiencing with our products, please visit Zilog's Technical Support page at http://support.zilog.com.

To learn more about this product, find additional documentation, or to discover other facets about Zilog product offerings, please visit the Zilog Knowledge Base at http://zilog.com/kb or consider participating in the Zilog Forum at http://zilog.com/forum.

This publication is subject to replacement by a later edition. To determine whether a later edition exists, please visit the Zilog website at http://www.zilog.com.

Warning: DO NOT USE this product IN LIFE SUPPORT systems.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8051 is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.