

EE3576 Tutorials

Communications Engineering

All answers should provide suitable diagrams and explanations. Unless the question explicitly requests a sketch of a diagram, diagrams alone shall not be considered sufficient to provide a full answer, be sure to explain the purpose of any diagram and to explain how this may be used to answer the question.

Students should review the material at: erg.abdn.ac.uk/users/gorry/eg3576/

List of Tutorials

- 1 *Asynchronous Transmission and Reception*
- 2 *Time Division Multiplexing (TDM)*
- 3 *EIA-485 Transmission*
- 4 *DMX Frame Transmission*
- 5 *DMX Microcontroller Algorithms*
- 6 *DMX Receiver Processing*
- 8 *Remote Device Management (RDM)*
- 9 *Controller Area Network (CAN) Bus*

Tutorial 1: Asynchronous Transmission and Reception

(a) An ASCII character is transmitted using the 7-bit ASCII code with one parity bit, followed by two stop bauds.

Start of sequence:



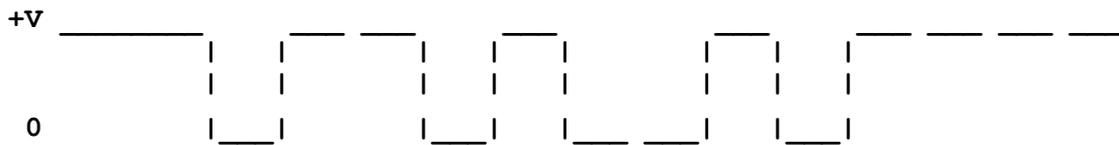
What is the decimal value of the byte sent in this sequence?

What is the ASCII character represented by this byte?

What form of parity was used?

(b) The following byte was sent using 8 bits, no parity, 2 stop bauds.

Start of sequence:



What is the hexadecimal value of the byte sent in this sequence?

What ASCII character represented by this byte?

(3) What is the sequence of bauds transmitted to send an ASCII character 'j' (hexadecimal 0x6A) at 4800 bps using no parity and two stop bits?

What period of time is required to send this character?

(A follow-up question for later in the course: what signal bandwidth is needed in kHz?)

Tutorial 2: Time Division Multiplexing (TDM)

- (a) A sender generates a Time Division Multiplex stream in which the 32 bytes of each frame are represented as characters, and the start of a frame is denoted by the 3 characters “\$GP” inserted in the first 3 positions of each frame. Explain how a receiver extracts the data sixth position of the frame from the stream of characters shown below. [5 marks]

```
FRAMEA$GPHHHNOVALUEFRAMERXFRAMDF
OXKYPQ$GPEEEHTHOINTERMSOFCASHTHG
PJOLWZ$GPMAAYXHOINTERCIBJKDTGSDF
POKLLY$GPMTTYFRAMECANCASHITNICHG
```

- (b) How can someone add an integrity check to detect corruption of the data? [3 marks]
- (c) What is the percentage overhead for this framing method (i.e., what percentage of capacity is used for bytes other than the frame payload data), assuming:
- Assuming asynchronous transmission of slots is used.
 - The Frame Alignment method above is used with a one byte integrity check.

[5 marks]

Tutorial 3: EIA-485 Transmission

- (a) A bus uses *asynchronous character framing*, with *differential transmission*, in which one slot is sent with 2 stop bits, no parity baud included. Sketch the sequence of signals sent on both wires, when sending the hexadecimal value 0x05. [6 marks]
- (b) If the baud rate is 250 k baud, what is the maximum frame rate for 512 slot frames? [5 marks] How is this rate changed if slots are of size 52B?
- (c) Asynchronous transmission does not require a clock signal to be sent - use diagrams to show how it is possible for a receiver to work when a 250 000 baud transmit clock is 2% lower than the normal receive clock frequency? Show also that this can not work at a 10% lower baud rate. [10 marks]
- (d) What is the purpose of the *Stop bits*? [4 marks]

Tutorial 4: DMX Frame Transmission

- (a) The DMX multiplex may be sent on an interface that uses a 3 or 5 pin XLR connector. What are the properties of this interface that make it suited for use in an industrial environment? [4 marks]
- (b) A DMX-512 control bus uses a *balanced transmission line*, draw a diagram showing the way equipment is connected to the bus and the waveform when the bus is used to send a 100 kHz square wave. [8 marks]
- (c) Explain how a receiver determines the *start* of a received DMX-512 frame. [8 marks]
- (d) Estimate the maximum number of receivers allowed on the bus, given a nominal receiver input impedance of 12 k Ohms [5 marks]
- (e) What is the effect of two 4 channel DMX fixtures being configured with a start address of 5? Explain how these process a received DMX frame to extract the values that represent each channel. [8 marks]

Tutorial 5: DMX Microcontroller Algorithms

- (a) A microprocessor-based DMX receiver uses software that includes the following Interrupt Service Routine. Draw a flow chart to capture and explain the operation of this algorithm. [10 marks]

```

ISR (UART_RX_vect)
{
    static uint16_t DmxCount;           //global counter
    uint8_t USARTstate= UCSRA;         //get USART state before data!
    uint8_t DmxByte = UDR;             //get USART data
    uint8_t DmxState = gDmxState;     //copy global to increase speed

    if (USARTstate & (1<<FE))         //check for break
    {
        UCSRA &= ~(1<<FE);             //reset flag in USART
        DmxCount = DmxAddress;         //reset slots counter
                                        //(count slots before start address)
        gDmxState= BREAK;
    }
    else if (DmxState == BREAK)
    {
        if (DmxByte == 0) gDmxState= STARTB; //normal start code detected
        else gDmxState= IDLE;
    }
    else if (DmxState == STARTB)
    {
        if (--DmxCount == 0)           //start address reached?
        {
            DmxCount= 1;               //set up counter for
                                        // required slots
            DmxRxField[0]= DmxByte;    //get 1st DMX channel of device
            gDmxState= STARTADR;
        }
    }
    else if (DmxState == STARTADR)
    {
        DmxRxField[DmxCount++]= DmxByte; //get channel
        if (DmxCount >= sizeof(DmxRxField)) //all slots received for device?
        {
            gDmxState= IDLE;           //wait for next break
        }
    }
    return i;
}

```

- (b) The Receiver has a *base address* of 6 and supports four 8-bit channels. Explain how the receiver uses the DMX frame to set the output value of the 2nd channel. [4 marks]
- (c) The value of a channel has an initial DMX slot value of 0, later a value of 50 and then finally a value of 100. Sketch the waveform for each of the values that is used to *drive* a TRIAC dimmer circuit, and the resulting voltage on the mains power line. [10 marks]

Additional thoughts:

- (i) What is the purpose of the return i statement? What does it do?
- (ii) Can you represent the algorithm as a state transition diagram? What variable is used to store the state?
- (iii) Why is the code written so that the global state is copied to a local variable during the interrupt service?

Tutorial 6: DMX Receiver Processing

Example: This example considers a DMX receiver with a start address of 6, and a field size of four slots. It traces the values of key variables after completing each interrupt service when the receiver receives a DMX frame with the following first ten data slots {11,12,13,14,15,16,17,18,19,20,...}.

Sample data including start code: DMX data = {0,11,12,13,14,15,16,17,18,19,20,...}

DMXAddr = 6 (configured start address)

DMXRxFld = 4 Bytes (given receive size).

Variable	Note
gDMXstate	{0=Idle;1=Break;2=StartB;3=StartAddr;}
USART Status	{FE = Break; 0 = Data} - flag triggering ISR
DMXByte	Contents of receive byte from USART
DMXAddr	6 (configured from DIP Switch)
DMXCount	Global slot counter
DMXRxFld[0..1]	Output Array of size 4 bytes

Here is a trace reporting key variable values after completion of the interrupt service routine.

Interrupt causing ISR	DMXByte	gDMXstate	DMXcount	DMXRxFld			
				[0]	[1]	[2]	[3]
Break	N/A	1	6	?	?	?	
Data	0	2					
Data	11						
Data							
Data							
Data							
Data							
Data							
Data							
Data							
Data	...	0	4	16	17	18	19

Tutorial 7: Control Networks

No current Tutorial

Tutorial 8: Remote Device Management (RDM)

- (a) Remote Device Management is an extension to DMX that allows configuration of remote devices and retrieval of data from them. What changes are needed in a device's electronics to enable RDM operation? [4 marks]
- (b) Explain the sequence of interactions required when a controller wishes to retrieve data from a remote device. [6 marks]
- (c) In RDM, the bus may be unpowered for periods of time. Explain how a bias circuit may be used to prevent the bus floating to an arbitrary value, and calculate the values of the bias components. [15 marks]

Tutorial 9: Controller Area Network (CAN) Bus

- (a) A CAN bus uses a variant of differential transmission, what is the key difference to methods such as RS-485 and DMX? [5 marks]
- (b) Explain how CAN uses bit stuffing to achieve redundancy. [10 marks]
- (c) Consider two nodes with two message IDs: 415 and 455, what is the sequence for the first 12 bits if each is sent individually, and what is the resulting arbitration when the two messages are sent simultaneously? [10 marks]