EG3576 Tutorials
Communications Engineering

Examination Style

The examination shall consist of 5 compulsory questions each 20 marks of a similar style to those used in the tutorials. Marks assigned to tutorial questions are for illustration only.

In the exam students are expected to answer all 5 questions (some will contain optional parts). Each question carries 20 marks, resulting in a maximum score of 100 marks.

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 the diagram and explain how this may be used to answer the question.

List of Tutorials

- *Time Division Multiplexing (TDM)*
- *Asynchronous Transmission and Reception*
- *DMX Slot Transmission*
- *DMX Frame Transmission*
- *DMX Microcontroller Algorithms*
- *Remote Device Management (RDM)*
- *Controller Area Network (CAN) Bus*

Tutorial: 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:

(a) Assuming asynchronous transmission of slots is used.

(b) The Frame Alignment method above is used with a one byte integrity check.

[5 marks]

(d) 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 a distributed frame alignment method in which the characters "F" "R" "A" "M" "E" are inserted in sequence in the first positions of each successive frame. Explain how a receiver extracts the data for the fifth position within the frame from the stream of characters [5 marks]

```
AFRAMEHASFOVALUEFRAMEXYHIPQWERTY
THOINFFRMAOFCASHTRRAXEXYXGOPIGSX
HOINTRRCIRJKDTGSHTARAMEXYPHANTOM
FRAMEACNCMSHITNICELMFRAMEOFXYRGF
HASNOMALUEZREWDTYVEEFHFTHTHEMEEE
HASNOEEALREZREWDTGGEEFHFTOPERAXY
```

Tutorial: Time Division Multiplexing (TDM2)

(a) A system uses a ***distributed*** frame alignment uses a 32 byte frame with one slot reserved for synchronisation and 5 frames per superframe. This may be implemented as a state machine. Can you draw a state machine that could be used to "know" which character of the Frame Alignment Word (FAW) comes next? [5 marks]

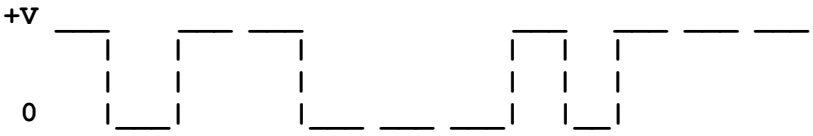
(b) What is the percentage overhead for this framing method? (i.e. what percentage of capacity is used for bytes other than the frame payload data). [2 marks]

(c) Practical systems often require synchronisation to be robust to some level of corruption - e.g. synch is maintained when a single mistake is then followed series of "n" correct values. Can you design a receiver to be robust to corruption of one byte of FAW provided that it is followed by 2 good Frame Alignment Word (FAW) values? [15 marks]

Tutorial: 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:



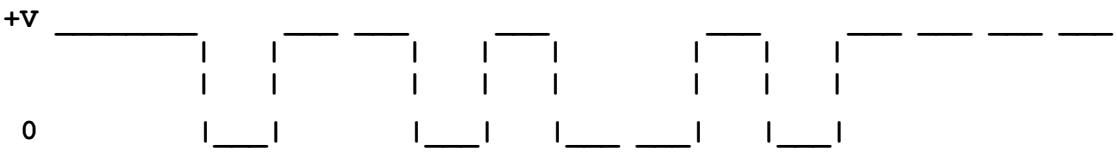
What is the decimal value of the byte sent in this sequence?

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: DMX Slot Transmission

(a) The DMX 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 of DMX 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: 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) An RS-485 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]

(f)

Tutorial: 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
        }
   }
 returni;
}
```

(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 returni 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: 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: Controller Area Network (CAN) Bus

(a) CAN 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]