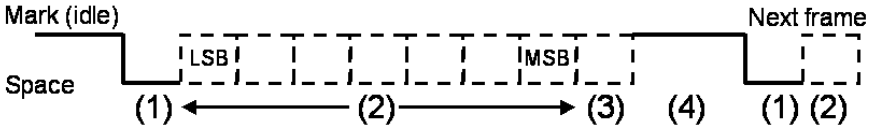

You may discuss these questions with your classmates, as it may be helpful to gain a good understanding of the topic. Nonetheless you should always submit your own work!
Use a word processor to answer these questions, attach this questions page to your answers.

Question 1.
Explain the RS-232C frame format. Depict a diagram indicating the various components of this format and explain each one of them.



(1) *Start bit: It indicates the starting time, i.e. mark level to space level transition, of a new data frame.*
(2) *Data bits: These bits generally encode 7-bit ASCII or ISO characters, or 8-bit EBCDIC characters. However format can also be used.*
(3) *Parity bit: For error checking, either odd or even parity. This is an optional bit.*
(4) *Stop bit: Indicates the end of the frame. It consists of a spacer after a character.*

Question 2.
Explain the Queued Serial Module (QSM). Indicate its main components/submodules. Specify the general purpose/functions of each of the submodules. Explain what the "port D" is and the signals it incorporates.
NOTE: This is a general question about the QSM, you do not have to talk about the registers of the QSM.

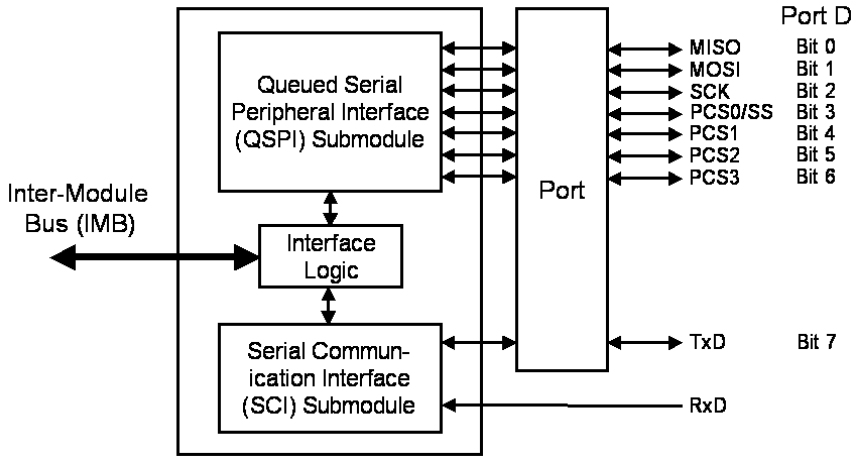
The queued serial module provides the MC32 with two serial communication interfaces. There are two independent submodules:

1. *The Serial Communication Interface (SCI)*
2. *The Queued Serial Peripheral Interface (QSPI)*

The SCI is a Universal Asynchronous Receiver Transmitter (UART) interface. In many applications the SCI transfers ASCII characters between the MC32 and other external devices that also have a UART interface.

The QSPI provides synchronous transfer of data for communication I/O expansion. The QSPI can act as a master or slave device in a network of devices with compatible serial interfaces. As a master, the QSPI controls the data format and timing of transfers between the QSPI and slave devices. This submodule contains a queue that allows up to 16 independent data transfers without intervention of the MC32. As master, the QSPI chip select signals lines select the specified peripheral device involved in each transfer.

The following diagram depicts the QSM and its I/O signal lines, and its submodules.



The following table describes the U/O signal lines.

Signal mnemonic	***Signal name***	***Function***
SCI		
RxD	*Receive data*	*Serial data to SCI*
TxD	*Transmit data*	*Serial data from SCI*
QSPI		
MISO	*Master in / slave out*	*Serial data to QSPI (master) / Serial data from QSPI (slave)*
MOSI	*Master out / slave in*	*Serial data from QSPI (master) / Serial data to QSPI (slave)*
SCK	*Serial clock*	*Clock output (master) / Clock input (slave)*
PCS1-PCS3	*Peripheral chip select signals*	*Output to select peripheral*
PCS0 / SS	*Peripheral chip select / slave select signals*	*Select peripheral (master) / Select QSPI (slave)*

The QSM signal lines can also form an 8-bit port called "port D" if they are not used for serial communication. Port D is formed by the following signals ordered from left (bit 0) to right (bit7):
MISO, MOSI, SCK, PCS0 / SS, PCS1-PCS3, and TxD.

Question 3.
Mention the four operating modes of the DUART. Briefly explain the purpose of each of them.

The four serial operating modes of the DUART are:

1. *Full duplex (normal mode): It is possible to transmit and receive data simultaneously.*
2. *Automatic echo: the information that the remote CPU sends to the local CPU is sent back to the remote CPU for verification purposes. It is also possible to send information to the remote CPU.*
3. *Local loop-back: The information that the local CPU sends over the DUART is sent back to the local CPU. Information from the remote CPU is also received. This mode is used for testing and debugging purposes.*
4. *Remote loop-back: The remote CPU sends information to the DUART connected to the local CPU. The information is sent back to the remote CPU without being received by the local CPU. Used to test the channel of communication.*

Question 4.

a) Describe the Interrupt Mask Register (IMR) of the DUART.

This is a write-only register. The bit settings in the register indicate which of the eight possible events can actually cause an interrupt request signal to be asserted.
The bit configuration of the IMR is described in the next table.

Bit in IMR	***Symbol***	***Function***
IMR0	*TxRDYA*	*Channel A transmitter is ready for more data*
IMR1	*RxRDYA / FFULLA*	*Channel A receiver has data to be read, or channel A FIFO buffer is full (Mode selected according to bit MR1A6)*
IMR2	*Delta break A*	*Start or end of break detected on channel A*
IMR3	*Counter / timer rdy*	*Counter / timer interrupt*
IMR4	*TxRDYB*	*Analogous to IMR0*
IMR5	*RxRDYB / FFULLB*	*Analogous to IMR1*
IMR6	*Delta break B*	*Analogous to IMR2*
IMR7	*Input port change*	*One of the parallel inputs with bit set in Aux. Control Register has undergone a transition*

b) Describe the Interrupt Status Register (ISR) of the DUART.

This is a read-only register. The bit format is the same as that for the IMR. Whenever an interrupt is capable of generating an active IRQ, the corresponding bit in the Interrupt Status Register is set. The shared IRQ signal is asserted if for at least one bit position, both IMR and IST bits are set.

c) Describe the Auxiliary Control Register (ACR) of the DUART.

The Auxiliary Control Register is used to configure the behaviour of the DUART interrupt interface:

- *The lower four bits ACR0 –ACR3 are used to enable change of state detectors on parallel inputs IP0-IP3.*
- *Bit ACR7 is used to select between two possible sets of baud rates that can be generated by the baud rate generator.*
- *Bits ACR4-ACR6 are used to select the operating mode and clock sources of the counter / timer function.*

d) Consider the following piece of code. It initializes the interrupts of the DUART. Indicate what interrupts would trigger an interrupt request signal.

```
.equ  INT_MASK,  0b11101110
.equ  ACR_SET,   0b00000011
      MOVE.B  #INT_MASK,IMR(A0)
      MOVE.B  #ACR_SET,ACR(A0)
```

RxRDYA / FFULLA, Delta break A, Counter / Timer rdy, RxRDYB / FFULLB, Delta break B, a change in the input ports IP0 and IP1.

Question 5.
Describe the layers of the ISO OSI 7-layer protocol.

It is a protocol developed in 1980. It defines the 7 layers of a reliable serial communication protocol as follows:

Layer	***Name***	***Services***
7	*Application*	*Generation, processing, and consumption of information*
6	*Presentation*	*Data format conversion, compression, and encryption*
5	*Session*	*Opening and closing of communication channels*
4	*Transport*	*Control of end-to-end flow of data byte streams*
3	*Network*	*Addressing and routing of packets through network*
2	*Data Link*	*Transmission of data frames from one node to the next*
1	*Physical*	*Transmission of bits over the communication medium*

Question 6.
Describes the fields of the TCP header.

- *The sequence number is a number that corresponds to the first data byte. All data packets have sequence numbers.*
- *The acknowledgement number identifies the sequence number of the next data byte that is expected to be received.*
- *The data offset gives the number of 32-bit words in the header:*
  - *Six flags are used for various purposes:*
  - *URG: Urgent pointer field enabled (MSB)*
  - *ACK: Acknowledgement field enabled*
  - *PSH: Push function. Force transmission of segment.*
  - *RST: Reset the connection*
  - *SYN: Synchronize the sequence numbers*
  - *FIN: No more data will be sent from sender (LSB)*
- *The window indicates the maximum number of data bytes that the other TCP entity must be prepared to accept without an acknowledgement.*
- *The checksum is computed over the TCP segment and allows the receiving TCP entity to detect transmission errors.*
- *The urgent pointer gives the sequence number of the first nonurgent data byte. Only used if URG = 1.*
- *Only one option is currently defined in TCP. It specifies the maximum segment size, in bytes, that will be accepted. The value is determined when the connection is established.*