Mestrado em Engenharia Eletrónica e de Computadores

Trabalho preparatório 1

Questão 1 - Qual a estrutura e dimensão standard da mensagem de transmissão do protocolo CAN (ver datasheet MCP2515)? E da mensagem estendida? Indique o que representa cada campo da mensagem.

O circuito integrado MCP2515 contém duas formas de transmitir uma mensagem de protocolo CAN. Uma é a standard e outra é a estendida.

A standard está dividida em 5 partes:

- Um início de trama que passa por transmitir 1 bit de SOF (Start of Frame -);
- Um campo arbitrário que é composto por 12 bits. 11 destes contém informação de identificação, sobrando um destinado a RTR (Remote Transmition Request);
- Um campo de controlo com 6 bits. É iniciado com um bit dedicado ao IDE (Identifier Extension), após outro bit destinado para o RB0 (Reserved bit zero), finalizando com os quatro bits restantes destinados para o tamanho dos dados a transmitir, denominados estes de bits de DLC (Data Length Code);
- Um campo exclusivamente para os dados a transmitir, com o tamanho de bytes igual ao valor do DLC;
- E por fim um campo de término da trama com um ACK (Acknowledge) composto por 2 bits.

A trama estendida está dividida em 5 partes:

- Um início de trama que passa por transmitir 1 bit de SOF (Start of Frame -);
- Um campo arbitrário que é composto por 32 bits. Onze destes contém informação de identificação, seguindo um bit denominado de SRR (Substitute Remote Request) juntamente com outro denominado a IDE (Identifier Extension). Após estes bits sobram 18 bits dos quais 17 são destinados ao Extended ID (identificação extendida) e um para o RTR bit;
- Um campo de controlo com 6 bits. É iniciado com dois bits reservados, finalizando com os quatro bits restantes destinados para o tamanho dos dados a transmitir, denominados bits de DLC (Data Length Code);
- Um campo exclusivamente para os dados a transmitir, com o tamanho de bytes igual ao valor do DLC;
- E por fim um campo de término da trama com um ACK (Acknowledge) composto por 2 bits.

Questão 2 - Quantos buffers de transmissão tem o CI MCP2515? E receção?

O CI MCP2515 tem três buffers de transmissão (TXB0, TXB1 e TXB2) e dois buffers de receção (RXB0 e RXB1).

Questão 3 - Identifique os registos associados à transmissão de mensagem. Para cada um deles indique o endereço e a composição do registo.

REGISTER 3-1: TXBnCTRL – TRANSMIT BUFFER n CONTROL REGISTER (ADDRESS: 30h, 40h, 50h)

U-0	R-0	R-0	R-0	R/W-0	U-0	R/W-0	R/W-0
—	ABTF	MLOA	TXERR	TXREQ	—	TXP1	TXP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-2: TXRTSCTRL – TXnRTS PIN CONTROL AND STATUS REGISTER (ADDRESS: 0Dh)

U-0	U-0	R-x	R-x	R-x	R/W-0	R/W-0	R/W-0
—	—	B2RTS	B1RTS	B0RTS	B2RTSM	B1RTSM	B0RTSM
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-3: TXBnSIDH – TRANSMIT BUFFER n STANDARD IDENTIFIER HIGH (ADDRESS: 31h, 41h, 51h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-4: TXBnSIDL – TRANSMIT BUFFER n STANDARD IDENTIFIER LOW (ADDRESS: 32h, 42h, 52h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-5: TXBnEID8 – TRANSMIT BUFFER n EXTENDED IDENTIFIER HIGH (ADDRESS: 33h, 43h, 53h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-6: TXBnEID0 – TRANSMIT BUFFER n EXTENDED IDENTIFIER LOW (ADDRESS: 34h, 44h, 54h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-7: TXBnDLC - TRANSMIT BUFFER n DATA LENGTH CODE (ADDRESS: 35h, 45h, 55h)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	RTR	—	—	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 3-8: TXBnDm – TRANSMIT BUFFER n DATA BYTE m (ADDRESS: 36h - 3Dh, 46h - 4Dh, 56h - 5Dh)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
TXBnDm7	TXBnDm6	TXBnDm5	TXBnDm4	TXBnDm3	TXBnDm2	TXBnDm1	TXBnDm0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Questão 4 - Que registo deve ser manipulado para definir se a mensagem transmitida é de dados ou de pedido de dados a um identificador externo.

The first byte, TXBnCTRL, is a control register associated with the message buffer. The information in this register determines the conditions under which the message will be transmitted and indicates the status of the message transmission.

Questão 5 - Como deve ser configurado o registo TXBnDLC se pretender transmitir uma mensagem com 6 bytes?

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	RTR	-	-	DLC3	DLC2	DLC1	DLC0
X	0	X	X	0	1	1	0

Questão 6 - Crie uma sequência que permita configurar os registos para o envio de uma mensagem standard?

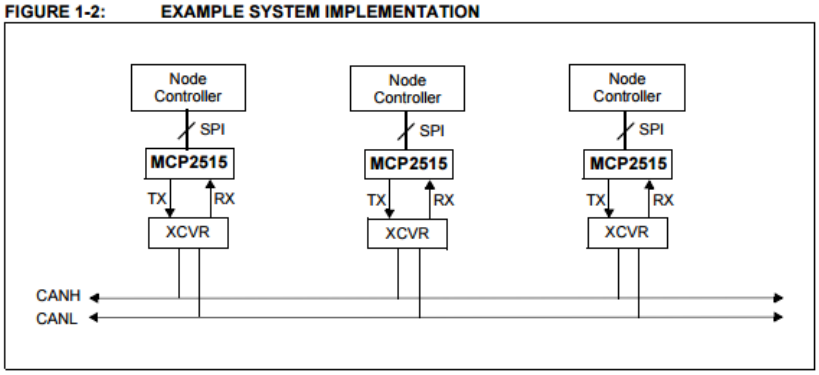
ter certeza que no registo TXBnSIDL , o bit EXIDE (4) está a 0 para que a trama enviada seja standard.

TXRTSCTRL - **--000001**
TXBnSIDL - **XXX-0-XX**

Questão 7 - Quantos condutores são necessários para ligar dois módulos CAN? Como é que estes devem ser ligados?

- CAN HIGH
- CAN LOW

FIGURE 1-2: EXAMPLE SYSTEM IMPLEMENTATION



Questão 8 - Qual a função do buffer MAB?

O buffer Message Assembly Buffer (MAB) tem como função juntar todas as mensagens recebidas. Após a verificação se estas mensagens passam por os critérios de aceitação (critérios para verificar se a mensagem está bem estruturada e não contém erros) estas depois são transferidas para os dois buffers de recepção, denominados de RXB0 e RXB1.

Questão 9 - Qual o nome dos dois buffers de recepção de mensagens? Qual tem maior prioridade?

Os dois buffers de recepção, designados de RXB0 e RXB1 são os buffers que recebem o conteúdo do MAB. O buffer RXB0 tem maior prioridade do que o RXB1.

Questão 10 - Qual o bit (flag) que identifica a chegada de uma nova mensagem? Como é que o microcontrolador deve gerir o seu estado entre mensagens?

O bit CANINTF.RXnIF identifica o momento em que uma mensagem é movida para alguns dos buffers de receção (RXB0 e RXB1).
Se este bit estiver a 1, então uma interrupção é gerada no pino que indica que uma mensagem válida (filtrada por o filtro de aceitação) foi recebida.

Questão 11 - Qual a função dos bits 0:3 do registo RXBnCTRL?

Quando uma mensagem é recebida, os bits 0:3 do registo RXBnCTRL irão indicar o número do filtro de aceitação que desencadeia a receção e se a mensagem recebida é um requerimento de transferência remota.

Questão 12 - Qual a finalidade da aplicação de filtros e máscaras às imagens recebidas?

Os filtros e máscaras de aceitação são utilizados para determinar se uma mensagem recebida no MAB deve ser carregada nalgum dos buffers de recepção (RXB0 ou RXB1). Quando uma mensagem válida é recebida no MAB, os campos de identificação da mensagem são comparados aos valores dos filtros. Se existir uma correspondência, a mensagem recebida é carregada no buffer de recepção respetivo.

Questão 13 - Qual a finalidade dos dois bits RXM do registo RXBnCTRL?

Os bits RXM do registo RXBnCTRL servem para filtrar o tipo de mensagens recebidas. Por defeito estes bits estão ambos a 0, no entanto caso estes bits estejam a 0 1 ou a 1 0, apenas irão ser aceites mensagens standard ou extended, respetivamente. Se ambos os bits estiverem a 1 então todas as mensagens serão recebidas, quer sejam estanderizadas, ou estendidas. Também as mensagens com erros antes do EOF são recebidas, tendo assim alguma utilidade em fazer debug no sistema.

Questão 14 - Qual a flag em que registo é identificado o pedido de transmissão de um módulo remoto?

O pedido é realizado no registo RXBnCTRL através da flag RXRTR: Received Remote Transfer Request bit

1 = Remote Transfer Request Received
0 = No Remote Transfer Request Received

Questão 15 - Identifique os registos associados à recepção de mensagem. Para cada um deles indique o endereço e a composição do registo.

REGISTER 4-1: RXB0CTRL – RECEIVE BUFFER 0 CONTROL (ADDRESS: 60h)

U-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	BUKT	BUKT1	FILHIT0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-5 **RXM**: Receive Buffer Operating Mode bits
11 = Turn mask/filters off; receive any message
10 = Receive only valid messages with extended identifiers that meet filter criteria
01 = Receive only valid messages with standard identifiers that meet filter criteria
00 = Receive all valid messages using either standard or extended identifiers that meet filter criteria

bit 4 **Unimplemented:** Read as '0'

bit 3 **RXRTR**: Received Remote Transfer Request bit
1 = Remote Transfer Request Received
0 = No Remote Transfer Request Received

bit 2 **BUKT**: Rollover Enable bit
1 = RXB0 message will rollover and be written to RXB1 if RXB0 is full
0 = Rollover disabled

bit 1 **BUKT1**: Read-only Copy of BUKT bit (used internally by the MCP2515)

bit 0 **FILHIT**: Filter Hit bit - indicates which acceptance filter enabled reception of message
1 = Acceptance Filter 1 (RXF1)
0 = Acceptance Filter 0 (RXF0)

REGISTER 4-2: RXB1CTRL – RECEIVE BUFFER 1 CONTROL (ADDRESS: 70h)

U-0	R/W-0	R/W-0	U-0	R-0	R-0	R-0	R-0
—	RXM1	RXM0	—	RXRTR	FILHIT2	FILHIT1	FILHIT0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-5 **RXM**: Receive Buffer Operating Mode bits
11 = Turn mask/filters off; receive any message
10 = Receive only valid messages with extended identifiers that meet filter criteria
01 = Receive only valid messages with standard identifiers that meet filter criteria
00 = Receive all valid messages using either standard or extended identifiers that meet filter criteria

bit 4 **Unimplemented:** Read as '0'

bit 3 **RXRTR**: Received Remote Transfer Request bit
1 = Remote Transfer Request Received
0 = No Remote Transfer Request Received

bit 2-0 **FILHIT**: Filter Hit bits - indicates which acceptance filter enabled reception of message
101 = Acceptance Filter 5 (RXF5)
100 = Acceptance Filter 4 (RXF4)
011 = Acceptance Filter 3 (RXF3)
010 = Acceptance Filter 2 (RXF2)
001 = Acceptance Filter 1 (RXF1) (Only if BUKT bit set in RXB0CTRL)
000 = Acceptance Filter 0 (RXF0) (Only if BUKT bit set in RXB0CTRL)

REGISTER 4-3: BFPCTRL – RXnBF PIN CONTROL AND STATUS (ADDRESS: 0Ch)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	B1BFS	B0BFS	B1BFE	B0BFE	B1BFM	B0BFM
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **Unimplemented:** Read as '0'

bit 5 **B1BFS**: $\overline{\text{RX1BF}}$ Pin State bit (Digital Output mode only)
- Reads as '0' when RX1BF is configured as interrupt pin

bit 4 **B0BFS**: $\overline{\text{RX0BF}}$ Pin State bit (Digital Output mode only)
- Reads as '0' when RX0BF is configured as interrupt pin

bit 3 **B1BFE**: $\overline{\text{RX1BF}}$ Pin Function Enable bit
1 = Pin function enabled, operation mode determined by B1BFM bit
0 = Pin function disabled, pin goes to high-impedance state

bit 2 **B0BFE**: $\overline{\text{RX0BF}}$ Pin Function Enable bit
1 = Pin function enabled, operation mode determined by B0BFM bit
0 = Pin function disabled, pin goes to high-impedance state

bit 1 **B1BFM**: $\overline{\text{RX1BF}}$ Pin Operation Mode bit
1 = Pin is used as interrupt when valid message loaded into RXB1
0 = Digital Output mode

bit 0 **B0BFM**: $\overline{\text{RX0BF}}$ Pin Operation Mode bit
1 = Pin is used as interrupt when valid message loaded into RXB0
0 = Digital Output mode

REGISTER 4-4: RXBnSIDH – RECEIVE BUFFER n STANDARD IDENTIFIER HIGH (ADDRESS: 61h, 71h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 7							bit 0

bit 7-0 **SID**: Standard Identifier bits <10:3>
These bits contain the eight most significant bits of the Standard Identifier for the received message

REGISTER 4-5: RXBnSIDL – RECEIVE BUFFER n STANDARD IDENTIFIER LOW (ADDRESS: 62h, 72h)

R-x	R-x	R-x	R-x	R-x	U-0	R-x	R-x
SID2	SID1	SID0	SRR	IDE	—	EID17	EID16
bit 7							bit 0

bit 7-5 **SID**: Standard Identifier bits <2:0>
These bits contain the three least significant bits of the Standard Identifier for the received message

bit 4 **SRR:** Standard Frame Remote Transmit Request bit (valid only if IDE bit = '0')
1 = Standard Frame Remote Transmit Request Received
0 = Standard Data Frame Received

bit 3 **IDE:** Extended Identifier Flag bit
This bit indicates whether the received message was a Standard or an Extended Frame
1 = Received message was an Extended Frame
0 = Received message was a Standard Frame

bit 2 **Unimplemented:** Reads as '0'

bit 1-0 **EID**: Extended Identifier bits <17:16>
These bits contain the two most significant bits of the Extended Identifier for the received message

REGISTER 4-6: RXBnEID8 – RECEIVE BUFFER n EXTENDED IDENTIFIER HIGH (ADDRESS: 63h, 73h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7							bit 0

bit 7-0 **EID**: Extended Identifier bits <15:8>
These bits hold bits 15 through 8 of the Extended Identifier for the received message

REGISTER 4-7: RXBnEID0 – RECEIVE BUFFER n EXTENDED IDENTIFIER LOW (ADDRESS: 64h, 74h)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0

bit 7-0 **EID**: Extended Identifier bits <7:0>
These bits hold the least significant eight bits of the Extended Identifier for the received message

REGISTER 4-8: RXBnDLC – RECEIVE BUFFER n DATA LENGHT CODE (ADDRESS: 65h, 75h)

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

bit 7 **Unimplemented:** Reads as '0'
bit 6 **RTR**: Extended Frame Remote Transmission Request bit (valid only when RXBnSIDL.IDE = '1')
1 = Extended Frame Remote Transmit Request Received
0 = Extended Data Frame Received
bit 5 **RB1:** Reserved Bit 1
bit 4 **RB0:** Reserved Bit 0
bit 3-0 **DLC**: Data Length Code bits <3:0>
Indicates number of data bytes that were received

REGISTER 4-9: RXBnDM – RECEIVE BUFFER n DATA BYTE M (ADDRESS: 66h - 6Dh, 76h - 7Dh)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RBnDm7	RBnDm6	RBnDm5	RBnDm4	RBnDm3	RBnDm2	RBnDm1	RBnDm0
bit 7							bit 0

bit 7-0 **RBnDm7:RBnDm0**: Receive Buffer n Data Field Bytes m
Eight bytes containing the data bytes for the received message

Questão 16 - Em que registo é identificada a dimensão em bytes dos dados recebidos?

É identificado no registo RXBnDLC, nos bits 3:0 onde contém o DLC (Data Length Code).

Questão 17 - Como é estabelecida a ordem dos filtros?

Se um dos filtros de aceitação corresponder, o FILHIT bits irão codificar o valor binário para o menor número filtrado correspondente. Por exemplo, se o filtro RXF2 e o filtro RXF4 corresponderem, FILHIT irá ser carregado com o valor de RXF2. Essencialmente prioriza os filtros de aceitação com uma numeração inferior com maior prioridade. Mensagens são comparadas com filtros numa ordem ascendente do número do filtro. Isto também assegura que a mensagem irá ser apenas recebida para um buffer. Isto implica que o RXB0 terá maior prioridade do que o RXB1.

Questão 18 - Explique como pode ser configurado um filtro para as mensagens recebidas.

Um filtro para mensagens recebidas pode ser configurado a partir da associação dos bits FILHIT do registo RXBnCTRL. Para o buffer 0 temos o registo RXB0CTRL.FILHIT0 for buffer 0, enquanto que para o buffer 1 temos o registo RXB1CTRL.FILHIT<2:0>. Estes três bits FILHIT para o buffer 1 podem ser configurados da seguinte forma:

- 101 = Acceptance Filter 5 (RXF5)
- 100 = Acceptance Filter 4 (RXF4)
- 011 = Acceptance Filter 3 (RXF3)

- 010 = Acceptance Filter 2 (RXF2)
- 001 = Acceptance Filter 1 (RXF1)
- 000 = Acceptance Filter 0 (RXF0)

Nota: 000 and 001 apenas podem ser aplicados se o bit BUKT no registo RXB0CTRL for 1, permitindo que as mensagens do RXB0 se sobreponham às do RXB1.

O registo RXB0CTRL possui duas cópias do bit BUKT e do FILHIT<0> bit. A codificação do bit BUKT permite que esses três bits sejam usados de forma semelhante aos bits RXB1CTRL.FILHIT e para distinguir um hit no filtro RXF0 e RXF1 em RXB0 ou após um roll over em RXB1.

- 111 = Acceptance Filter 1 (RXB1)
- 110 = Acceptance Filter 0 (RXB1)
- 001 = Acceptance Filter 1 (RXB0)
- 000 = Acceptance Filter 0 (RXB0)

Questão 19 - Identifique os registos associados à configuração dos filtros e máscaras.

A lista de registos para a configuração dos filtros e máscaras é a seguinte:
REGISTER 4-10: RXFnSIDH – FILTER n STANDARD IDENTIFIER HIGH
REGISTER 4-11: RXFnSIDL – FILTER n STANDARD IDENTIFIER LOW
REGISTER 4-12: RXFnEID8 – FILTER n EXTENDED IDENTIFIER HIGH
REGISTER 4-13: RXFnEID0 – FILTER n EXTENDED IDENTIFIER LOW
REGISTER 4-14: RXMnSIDH – MASK n STANDARD IDENTIFIER HIGH
REGISTER 4-15: RXMnSIDL – MASK n STANDARD IDENTIFIER LOW
REGISTER 4-16: RXMnEID8 – MASK n EXTENDED IDENTIFIER HIGH
REGISTER 4-17: RXMnEID0 – MASK n EXTENDED IDENTIFIER LOW

Questão 20 - Em que consiste o BIT TIMING? Que cuidados devemos ter com este atributo quando configuramos uma rede CAN? Em que registos podemos configurar?

O Bit Timing consiste em supervisionar a entrada da linha do barramento e processar o tempo de bits relacionado com o bus de acordo com o protocolo CAN.

Os cuidados a ter são:

- Todos os dispositivos do barramento CAN têm de usar o mesmo BITRATE
- No entanto não que todos os dispositivos possuam a mesma frequência do master.
- Para os dispositivos com frequências diferentes o BITRATE é necessário ajustar o prescaler do baud rate e o número do time quanta de forma a serem iguais.

Os registos de configuração são: (CNF1, CNF2, CNF3).

Questão 21 - Quantos tipos de interrupção apresenta o MCP2515? Em que registo podem ser habilitadas? E em que registo estão disponíveis as flag das mesmas quando estas são habilitadas?

O MCP2515 tem oito tipos de interrupção no registo CANINTE. O registo CANINTF contém as flags de interrupção correspondentes para cada fonte de interrupção.

Questão 22 - O que acontece ao pino /INT quando ocorre uma interrupção?

Quando ocorre uma interrupção o pino INT passa para o estado LOW pelo MCP2515 e assim permanece até que a interrupção seja apagada pelo MCU.

Questão 23 - Quais os modos de operação do MCP2515? Em que consistem?

O MCP2515 tem cinco modos de operação que são:

1. Configuration Mode - O modo de configuração é automaticamente selecionado após o power-up, um reset ou ainda através do ajuste dos bits de CANTRL.REQOP para '100'. Quando o modo de configuração é inserido, todos os contadores de erro são apagados.

2. Normal Mode - é o modo de operação padrão do MCP2515. Neste modo, o dispositivo monitoriza ativamente todas as mensagens de barramento e gera bits de confirmação (acknowledge), error frames, etc. É também o único modo no qual o MCP2515 transmite mensagens através do barramento CAN.

3. Sleep Mode - é utilizado para minimizar o consumo de corrente do dispositivo

4. Listen-only mode - pode ser usado para aplicações de monitorização de barramento ou para detetar a taxa de transmissão em situações de 'hot plugging'.

5. Loopback mode - permite a transmissão interna de mensagens dos buffers de transmissão para os buffers de receção sem realmente transmitir mensagens no CAN bus. Este modo pode ser usado no desenvolvimento e teste do sistema.

Questão 24 - Em que registo e bits pode ser definido o modo de operação? (pág 58-59)

REGISTER 10-1: CANCTRL – CAN CONTROL REGISTER (ADDRESS: XFh)

R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
REQOP2	REQOP1	REQOP0	ABAT	OSM	CLKEN	CLKPRE1	CLKPRE0
bit 7							bit 0

REGISTER 10-2: CANSTAT – CAN STATUS REGISTER (ADDRESS: XEh)

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—
bit 7							bit 0

Questão 25 - Em que bits e registo pode ser lido o modo de operação atual?

Nos bits 7-5 do registo CANSTAT pode ser verificado o modo de operação em vigor.

REGISTER 10-2: CANSTAT – CAN STATUS REGISTER (ADDRESS: XEh)

R-1	R-0	R-0	U-0	R-0	R-0	R-0	U-0
OPMOD2	OPMOD1	OPMOD0	—	ICOD2	ICOD1	ICOD0	—
bit 7							bit 0

bit 7-5 **OPMOD**: Operation Mode bits <2:0>
- 000 = Device is in the Normal operation mode
- 001 = Device is in Sleep mode
- 010 = Device is in Loopback mode
- 011 = Device is in Listen-only mode
- 100 = Device is in Configuration mode

bit 4 **Unimplemented:** Read as '0'

bit 3-1 **ICOD**: Interrupt Flag Code bits <2:0>
- 000 = No Interrupt
- 001 = Error Interrupt
- 010 = Wake-up Interrupt
- 011 = TXB0 Interrupt
- 100 = TXB1 Interrupt
- 101 = TXB2 Interrupt
- 110 = RXB0 Interrupt
- 111 = RXB1 Interrupt

bit 0 **Unimplemented:** Read as '0'

Questão 26 - Quais os registos que apenas podem ser modificados no modo de configuração?

No modo configuração é o único modo em que podem ser modificados os registos:

- CNF1, CNF2, CNF3
- TXRTSCTRL
- Filter registers
- Mask registers

Questão 27 - Que instruções estão disponíveis no modo SPI para controlar o funcionamento do MCP2515?

- Reset Instruction
- Read Instruction
- Read RX Buffer Instruction
- Write Instruction
- Load TX Buffer Instruction
- Request-To-Send (RTS) Instruction
- Read Status Instruction
- RX Status Instruction
- Bit Modify Instruction

Questão 28 - Quais os pinos que deverão ser utilizados no MCP2515 para que seja possível estabelecer uma comunicação SPI com um microcontrolador?

- SCK - Clock input pin for SPI interface
- SI - Data input pin for SPI interface
- SO - Data output pin for SPI interface
- CS - Chip select input pin for SPI interface

TABLE 1-1: PINOUT DESCRIPTION

Name	PDIP/SOIC Pin #	TSSOP Pin #	I/O/P Type	Description	Alternate Pin Function
TXCAN	1	1	O	Transmit output pin to CAN bus	—
RXCAN	2	2	I	Receive input pin from CAN bus	—
CLKOUT	3	3	O	Clock output pin with programmable prescaler	Start-of-Frame signal
$\overline{TX0RTS}$	4	4	I	Transmit buffer TXB0 request-to-send. 100 kΩ internal pull-up to VDD	General purpose digital input. 100 kΩ internal pull-up to VDD
$\overline{TX1RTS}$	5	5	I	Transmit buffer TXB1 request-to-send. 100 kΩ internal pull-up to VDD	General purpose digital input. 100 kΩ internal pull-up to VDD
$\overline{TX2RTS}$	6	7	I	Transmit buffer TXB2 request-to-send. 100 kΩ internal pull-up to VDD	General purpose digital input. 100 kΩ internal pull-up to VDD
OSC2	7	8	O	Oscillator output	—
OSC1	8	9	I	Oscillator input	External clock input
Vss	9	10	P	Ground reference for logic and I/O pins	—
$\overline{RX1BF}$	10	11	O	Receive buffer RXB1 interrupt pin or general purpose digital output	General purpose digital output
$\overline{RX0BF}$	11	12	O	Receive buffer RXB0 interrupt pin or general purpose digital output	General purpose digital output
$\overline{INT}$	12	13	O	Interrupt output pin	—
SCK	13	14	I	Clock input pin for SPI interface	—
SI	14	16	I	Data input pin for SPI interface	—
SO	15	17	O	Data output pin for SPI interface	—
$\overline{CS}$	16	18	I	Chip select input pin for SPI interface	—
$\overline{RESET}$	17	19	I	Active low device reset input	—
VDD	18	20	P	Positive supply for logic and I/O pins	—
NC	—	6,15	—	No internal connection	

Note: Type Identification: I = Input; O = Output; P = Power