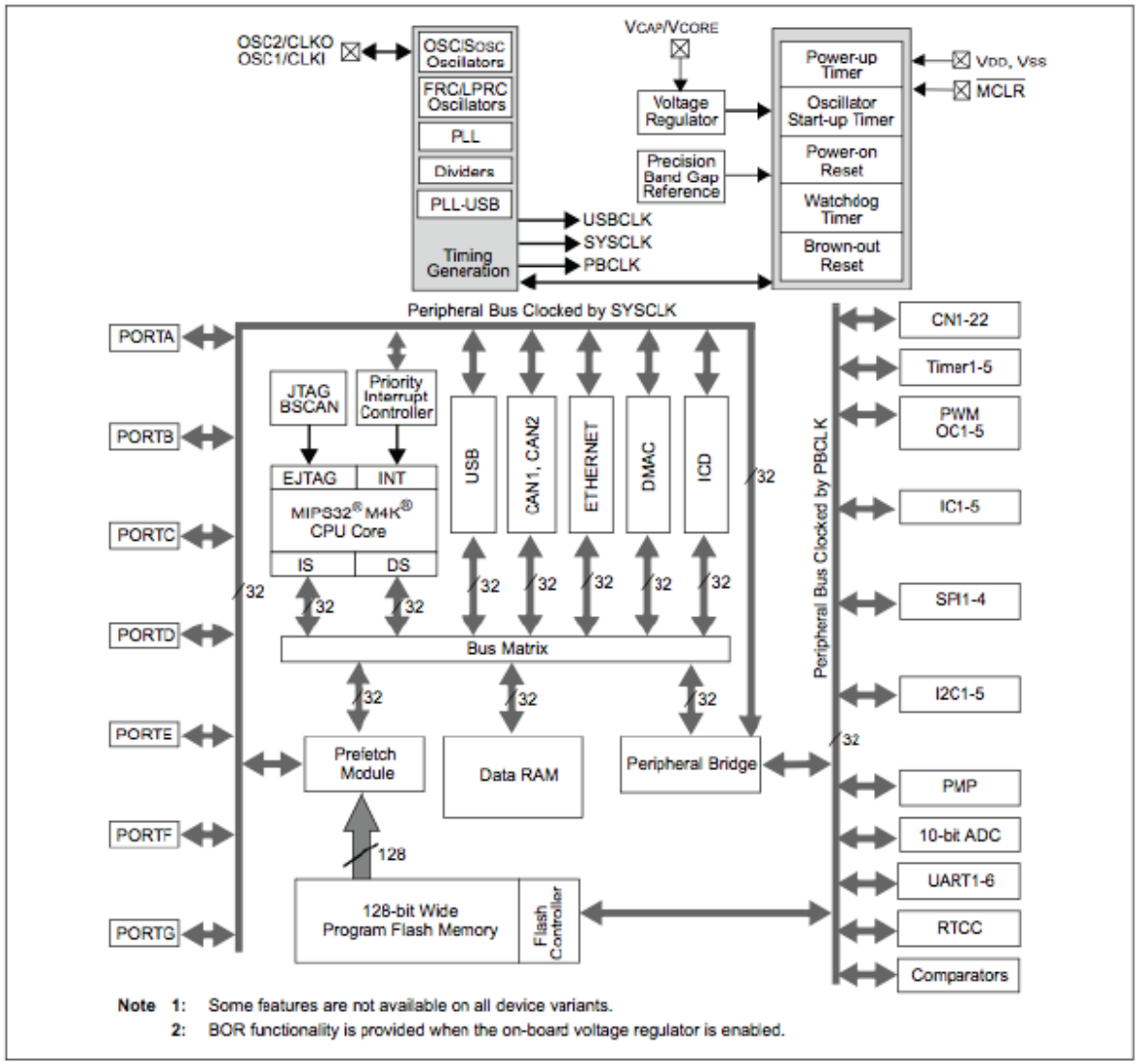
Writing software to communicate over the I^2C bus

ECE 2534

I²C implementation on the PIC32

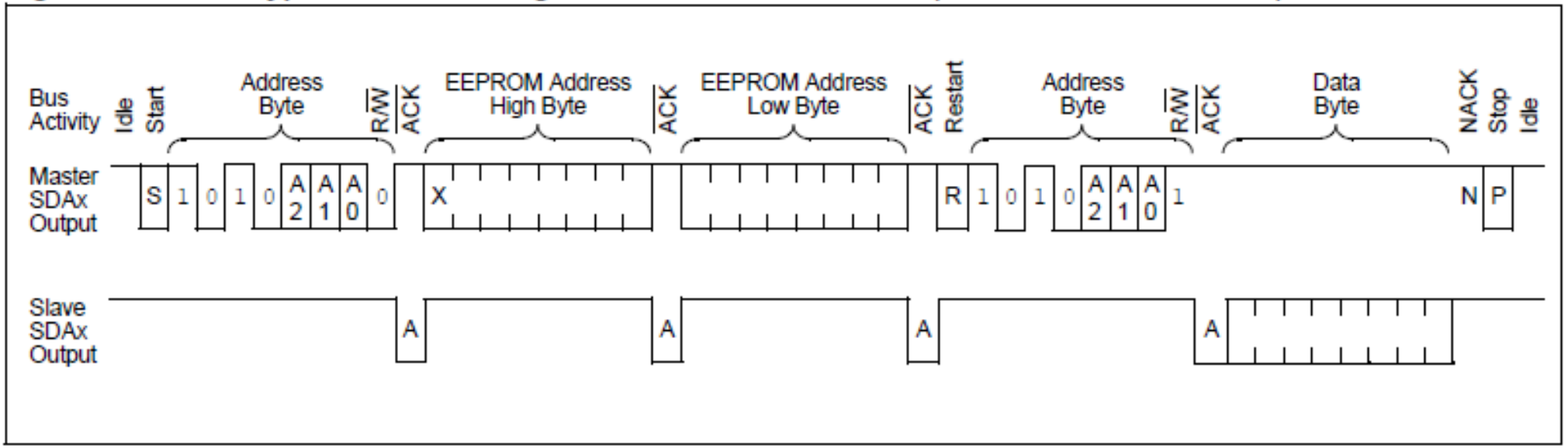
FIGURE 1-1: BLOCK DIAGRAM(1,2)



Note 1: Some features are not available on all device variants.
2: BOR functionality is provided when the on-board voltage regulator is enabled.

I²C data transfers – reminder

Figure 24-4: A Typical I²C™ Message: Read of Serial EEPROM (Random Address Mode)



Source: **61116E.pdf – PIC32 Reference Manual**

I²C data transfer sequence - reminder

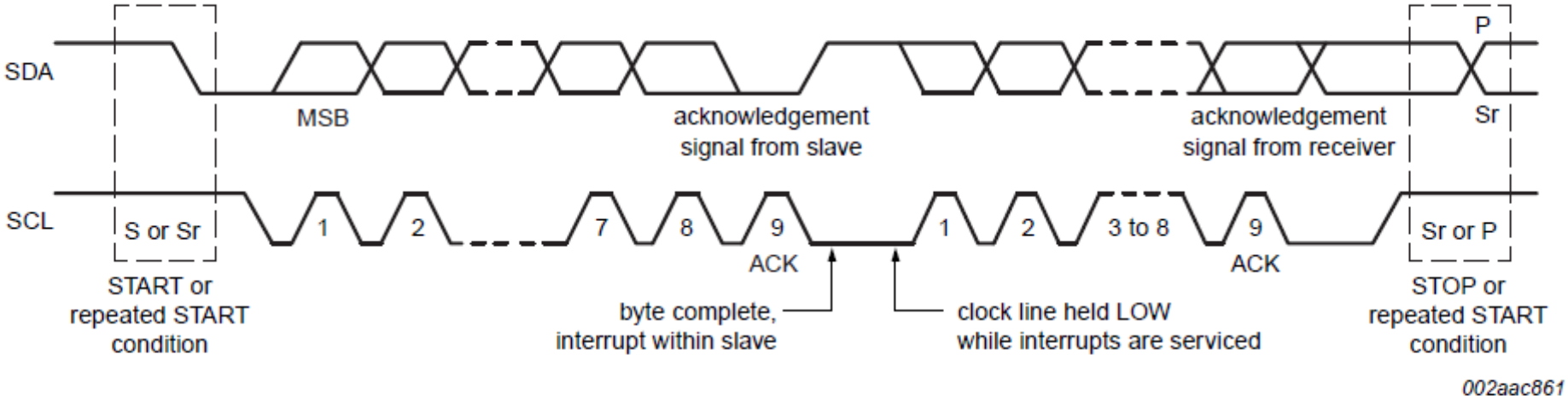
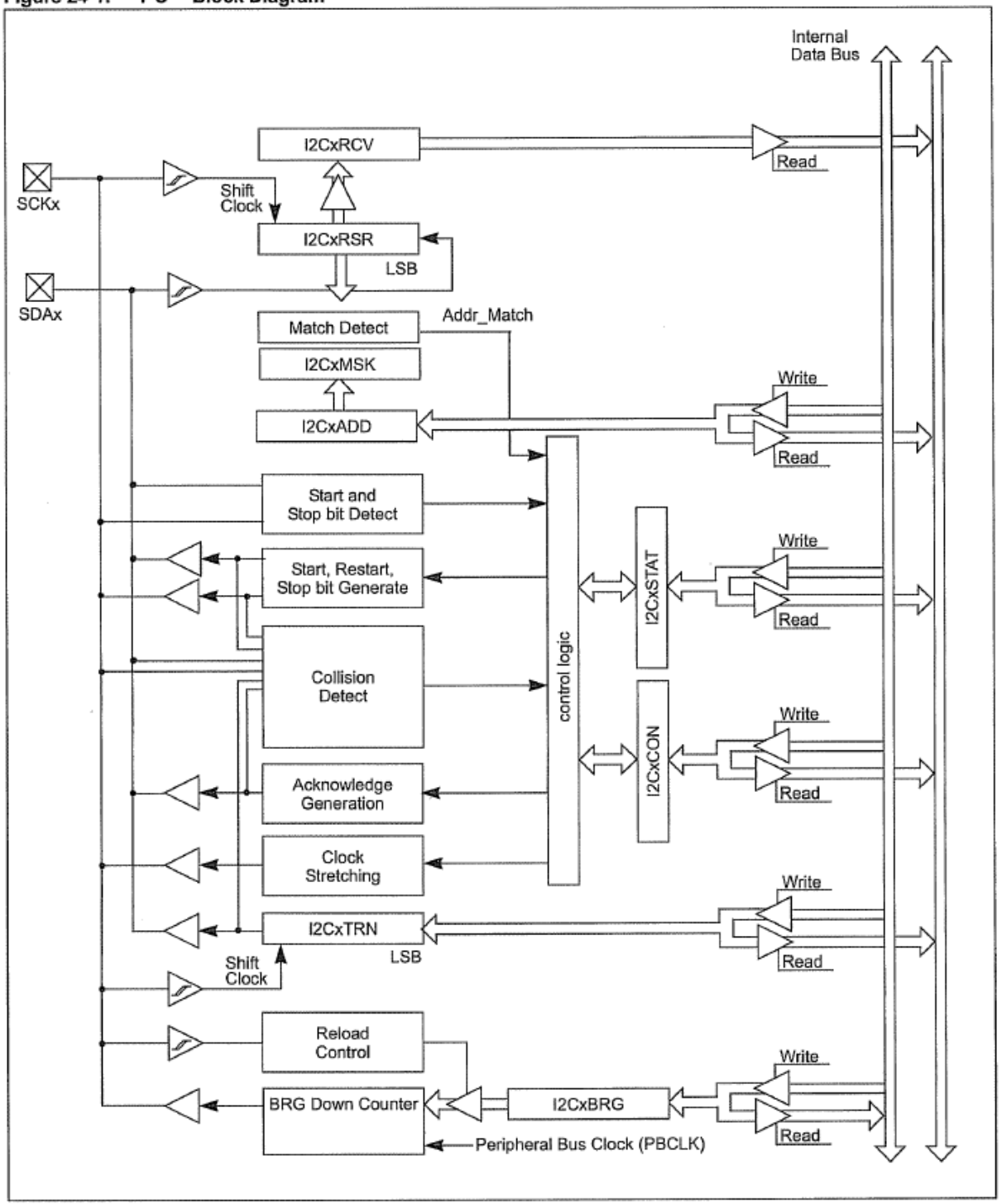


Fig 6. Data transfer on the I²C-bus

Figure 24-1: I²C™ Block Diagram



CONTROL AND STATUS REGISTERS

The I2C module consists of the following Special Function Registers (SFRs):

- **I2CxTRN: I2C Transmit Data Register**

  This read-only register is the transmit register. Bytes are written to this register during a transmit operation.

- **I2CxRCV: I2C Receive Data Register**

  This read-only register is the buffer register from which data bytes can be read.

- **I2CxADD: I2C Slave Address Register**

  This register holds the slave device address (for PIC32 acting in slave mode)

- **I2CxMSK: I2C Address Mask Register**

  This register designates which bit positions in the I2CxADD register can be ignored (allows PIC32 to respond to multiple addresses)

- **I2CxCON: I2C Control Register**

  This register enables control of the I2C module's operation.

- **I2CxSTAT: I2C Status Register**

  This register contains status flags indicating the I2C module's state during operation.

- **I2CxBRG: I2C Baud Rate Generator Register**

  This register holds the Baud Rate Generator (BRG) reload value for the I2C module Baud Rate Generator.

Control Register

Register 24-1: I2CxCON: I²C™ Control Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
	ON(1)	—	SIDL	SCLREL	STRICT	A10M	DISSLW	SMEN
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN

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

Source: **61116E.pdf – PIC32 Reference Manual**

Control Register

Register 24-1: I2CxCON: I²C™ Control Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
	ON(1)	—	SIDL	SCLREL	STRICT	A10M	DISSLW	SMEN
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN

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

Source: **61116E.pdf – PIC32 Reference Manual**

Status Register

Register 24-2: I2CxSTAT: I²C™ Status Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	U-0	U-0	U-0	R/W-0	R-0	R-0
	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
7:0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R-0	R-0
	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF

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

Source: **61116E.pdf – PIC32 Reference Manual**

Status Register

Register 24-2: I2CxSTAT: I²C™ Status Register

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	U-0	U-0	U-0	R/W-0	R-0	R-0
	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
7:0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R-0	R-0
	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF

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

Source: **61116E.pdf – PIC32 Reference Manual**

I²C Plib Functions: Initialization

- `void` **`I2CConfigure`**`(I2C_MODULE` *`id`*`,`
  `I2C_CONFIGURATION` *`flags`*`);`

- `UINT32` **`I2CSetFrequency`**`(I2C_MODULE` *`id`*`,`
  `UINT32 sourceClock, UINT32 i2cClock);`

- `void` **`I2CEnable`**`(I2C_MODULE` *`id`*`,`
  `BOOL` *`enable`*`);`

I²C Plib Functions: Starts & Stops

- `I2C_RESULT` **`I2CStart`**`(I2C_MODULE` *`id`*`);`
- `I2C_RESULT` **`I2CRepeatStart`**`(I2C_MODULE` *`id`*`);`
- `I2C_RESULT` **`I2CStop`**`(I2C_MODULE` *`id`*`);`

- *Need to check I2C status flags!*
- `BOOL` **`I2CBusIsIdle`**`(I2C_MODULE` *`id`*`);`
- `I2C_STATUS` **`I2CGetStatus`**`(I2C_MODULE` *`id`*`);`

I²C Plib Functions: Transmission (by master)

- `I2C_RESULT` **`I2CSendByte`**`(I2C_MODULE` *`id`*`,`
  `BYTE` *`data`*`);`
- `BOOL` **`I2CTransmissionHasCompleted`**`(`
  `I2C_MODULE` *`id`*`);`
- `BOOL` **`I2CByteWasAcknowledged`**`(`
  `I2C_MODULE` *`id`*`);`

I²C Plib Functions: Reception (by master)

- I2C_RESULT **I2CReceiverEnable**(
  I2C_MODULE *id*, BOOL *enable*);
- BOOL **I2CReceivedDataIsAvailable**(
  I2C_MODULE *id*);
- BOOL **I2CAcknowledgeByte**(
  I2C_MODULE *id*, BOOL *ack*);
- BOOL **I2CAcknowledgeHasCompleted**(
  I2C_MODULE *id*);
- BYTE **I2CGetByte**(I2C_MODULE *id*);

Where to find I²C Plib descriptions

c:

→ Program Files (x86)

→ Microchip

→ xc32

→ v1.31

→ pic32mx

→ include

→ peripheral

→ i2c.h