



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	ECANbus, I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46k80-i-p

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Pin Name	Pin Num	Pin Type	Buffer Type	Description
MCLR/RE3	28			
MCLR		Ι	ST	Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.
RE3		Ι	ST	General purpose, input only pin.
OSC1/CLKIN/RA7	46			
OSC1		Ι	ST	Oscillator crystal input.
CLKIN		I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7		I/O	ST/ CMOS	General purpose I/O pin.
OSC2/CLKOUT/RA6	47			
OSC2		0	_	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKOUT		0	_	In certain oscillator modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6		I/O	ST/ CMOS	General purpose I/O pin.
Legend: $l^2C^{TM} = l^2C/Sl$	MBus ir	put buff	er	CMOS = CMOS compatible input or output

TABLE 1-6:	PIC18F6XK80 PINOUT I/O DESCRIPTIONS

ST= Schmitt Trigger input with CMOS levelsAnalog= Analog inputI= InputO= OutputP= PowerO= Output

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROON		ROSSLP	ROSEL ⁽¹⁾	RODIV3	RODIV2	RODIV1	RODIV0
bit 7							bit 0
Legend:	L.:4		L 14	ll llaimealan			
R = Readable		vv = vvritable	DIL	0 = 0	nented bit, rea	uas u	
-n = value at	PUR	I = BILIS SEL		0 = Bit is cie	ared	x = Bit is unki	IOWN
bit 7	ROON: Refe	rence Oscillato	r Output Enab	le bit			
	1 = Referenc 0 = Referenc	e oscillator out e oscillator out	put is available put is disabled	e on REFO pin			
bit 6	Unimplemen	ted: Read as '	0'				
bit 5	ROSSLP: Re	eference Oscilla	ator Output Sto	op in Sleep bit			
	1 = Referenc 0 = Referenc	e oscillator con e oscillator is d	tinues to run ii isabled in Slee	n Sleep ep			
bit 4	ROSEL: Refe	erence Oscillate	or Source Sele	ect bit ⁽¹⁾			
	1 = Primary c 0 = System c	scillator (EC or lock is used as	HS) is used a the base cloc	as the base clo k; base clock r	ck eflects any clo	ck switching of	the device
bit 3-0	RODIV<3:0>	: Reference Os	cillator Divisor	r Select bits			
1111 = Base clock value divided by 32,768 1110 = Base clock value divided by 8,192 1100 = Base clock value divided by 4,096 1011 = Base clock value divided by 2,048 1010 = Base clock value divided by 1,024 1001 = Base clock value divided by 512 1000 = Base clock value divided by 256 0111 = Base clock value divided by 128 0110 = Base clock value divided by 64 0101 = Base clock value divided by 32 0100 = Base clock value divided by 32 0101 = Base clock value divided by 4 0101 = Base clock value divided by 32 0100 = Base clock value divided by 32 0100 = Base clock value divided by 4 0011 = Base clock value divided by 4 0010 = Base clock value divided by 4 0011 = Base clock value divided by 4 0001 = Base clock value divided by 2 0000 = Base clock value divided by 2 0000 = Base clock value							

REGISTER 3-4: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

Note 1: For ROSEL (REFOCON<4>), the primary oscillator is available only when configured as the default via the FOSCx settings. This is regardless of whether the device is in Sleep mode.

5.6.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (Parameter 33). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes and only on Power-on Reset or on exit from most power-managed modes.

5.6.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the oscillator start-up time-out.

5.6.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

- 1. After the POR pulse has cleared, PWRT time-out is invoked (if enabled).
- 2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 5-3, Figure 5-4, Figure 5-5, Figure 5-6 and Figure 5-7 all depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figures 5-3 through 5-6 also apply to devices operating in XT or LP modes. For devices in RC mode and with the PWRT disabled, on the other hand, there will be no time-out at all.

Since the time-outs occur from the POR pulse, if MCLR is kept low long enough, all time-outs will expire. Bringing MCLR high will begin execution immediately (Figure 5-5). This is useful for testing purposes or to synchronize more than one PIC18FXXXX device operating in parallel.

Oscillator	Power-up an	Exit from		
Configuration	PWRTEN = 0	PWRTEN = 1	Power-Managed Mode	
HSPLL	66 ms ⁽¹⁾ + 1024 Tosc + 2 ms ⁽²⁾	1024 Tosc + 2 ms ⁽²⁾	1024 Tosc + 2 ms ⁽²⁾	
HS, XT, LP	66 ms ⁽¹⁾ + 1024 Tosc	1024 Tosc	1024 Tosc	
EC, ECIO	66 ms ⁽¹⁾	_	—	
RC, RCIO	66 ms ⁽¹⁾	_	—	
INTIO1, INTIO2	66 ms ⁽¹⁾	_	—	

TABLE 5-2:TIME-OUT IN VARIOUS SITUATIONS

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.

2: 2 ms is the nominal time required for the PLL to lock.

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD, VDD RISE < TPWRT)



6.1.3.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 6-1) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off of the stack. On Reset, the Stack Pointer value will be zero.

The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

What happens when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (For a description of the device Configuration bits, see **Section 28.1** "**Configuration Bits**".) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and the STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and set the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or until a POR occurs.

Note:	Returning a value of zero to the PC on an
	underflow has the effect of vectoring the
	program to the Reset vector, where the
	stack conditions can be verified and
	appropriate actions can be taken. This is
	not the same as a Reset, as the contents
	of the SFRs are not affected.

6.1.3.3 PUSH and POP Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off of the stack, without disturbing normal program execution, is a desirable feature. The PIC18 instruction set includes two instructions, PUSH and POP, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The PUSH instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
STKFUL ⁽¹⁾	STKUNF ⁽¹⁾	—	SP4	SP3	SP2	SP1	SP0	
bit 7			•				bit 0	
Legend:		C = Clearable	bit					
R = Readable bit W = Writable bit			bit	U = Unimplemented bit, read as '0'				
-n = Value at POR '1' = Bit is set				'0' = Bit is cleared x = Bit is unknown				

REGISTER 6-1: STKPTR: STACK POINTER REGISTER

bit 7	STKFUL: Stack Full Flag bit ⁽¹⁾ 1 = Stack has become full or overflowed 0 = Stack has not become full or overflowed
bit 6	STKUNF: Stack Underflow Flag bit ⁽¹⁾ 1 = Stack underflow has occurred 0 = Stack underflow did not occur
bit 5 bit 4-0	Unimplemented: Read as '0' SP<4:0>: Stack Pointer Location bits

Note 1: Bit 7 and bit 6 are cleared by user software or by a POR.

6.1.3.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit (CONFIG4L<0>). When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

6.1.4 FAST REGISTER STACK

A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a "fast return" option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the Stack registers. The values in the registers are then loaded back into the working registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the Fast Register Stack for returns from interrupt. If no interrupts are used, the Fast Register Stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A RETURN, FAST instruction is then executed to restore these registers from the Fast Register Stack.

Example 6-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

EXAMPLE 6-1: FAST REGISTER STACK CODE EXAMPLE

CALL SUB1, FAST • •	;STATUS, WREG, BSR ;SAVED IN FAST REGISTER ;STACK
SUB1 •	
RETURN FAST	;RESTORE VALUES SAVED ;IN FAST REGISTER STACK

6.1.5 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

6.1.5.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the Program Counter. An example is shown in Example 6-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions that returns the value, 'nn', to the calling function.

The offset value (in WREG) specifies the number of bytes that the Program Counter should advance and should be multiples of two (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 6-2: COMPUTED GOTO USING AN OFFSET VALUE

	MOVF	OFFSET,	W
	CALL	TABLE	
ORG	nn00h		
TABLE	ADDWF	PCL	
	RETLW	nnh	
	RETLW	nnh	
	RETLW	nnh	
	•		

6.1.5.2 Table Reads

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory one byte at a time.

The table read operation is discussed further in **Section 7.1 "Table Reads and Table Writes**".

REGISTER 10-15: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP		—	—	BCLIP	HLVDIP	TMR3IP	TMR3GIP
bit 7		•		•			bit 0
Legend:							
R = Readable	e bit	W = Writable	bit	U = Unimplem	nented bit, read	d as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	nown
bit 7	OSCFIP: Osc	illator Fail Inter	rupt Priority bit	:			
	1 = High prio	rity					
	0 = Low prior	ity					
bit 6-4	Unimplemen	ted: Read as ')'				
bit 3	BCLIP: Bus C	Collision Interru	pt Priority bit				
	1 = High prio	rity					
	0 = Low prior	ity					
bit 2	HLVDIP: High	n/Low-Voltage [Detect Interrupt	t Priority bit			
	1 = High prio	rity					
	0 = Low prior	ity					
bit 1	TMR3IP: TMF	R3 Overflow Int	errupt Priority I	oit			
	1 = High priority						
	0 = Low prior	ity					
bit 0	TMR3GIP: TN	/IR3 Gate Inter	rupt Priority bit				
	1 = High prio	rity					
	0 = Low priority						

11.7 PORTF, LATF and TRISF Registers

PORTF is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISF and LATF. All pins on PORTF are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: PORTF is only available on 64-pin devices.

Each of the PORTF pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is done by clearing bit, RFPU (PADCFG1<5>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

Note: On device Resets, pins, RF<7:1>, are configured as analog inputs and are read as '0'.

EXAMP	LE 11-6:		INITIALIZING PORTF
CLRF	PORTF	;	Initialize PORTF by
		;	clearing output
		;	data latches
CLRF	LATF	;	Alternate method
		;	to clear output
		;	data latches
MOVLW	OCEh	;	Value used to
		;	initialize data
		;	direction
MOVWF	TRISF	;	Set RF3:RF1 as inputs
		;	RF5:RF4 as outputs
		;	RF7:RF6 as inputs
1			

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RF0/MDMIN	RF0	0	0	DIG	LATF<0> data output.
		1	Ι	ST	PORTF<0> data input.
	MDMIN	1	Ι	ST	Modulator source input.
RF1	RF1	0	0	DIG	LATF<1> data output.
		1	Ι	ST	PORTF<1> data input.
RF2/MDCIN1	RF2	0	0	DIG	LATF<2> data output.
		1	Ι	ST	PORTF<2> data input.
	MDCIN1	1	Ι	ST	Modulator Carrier Input 1.
RF3	RF3	0	0	DIG	LATF<3> data output.
		1	Ι	ST	PORTF<3> data input.
RF4/MDCIN2	RF4	0	0	DIG	LATF<4> data output.
		1	Ι	ST	PORTF<4> data input.
	MDCIN2	1	Ι	ST	Modulator Carrier Input 2.
RF5	RF5	0	0	DIG	LATF<5> data output.
		1	Ι	ST	PORTF<5> data input.
RF6/MDOUT	RF6	0	0	DIG	LATF<6> data output.
		1	I	ST	PORTF<6> data input.
	MDOUT	0	0	DIG	Modulator output.
RF7	RF7	0	0	DIG	LATF<7> data output.
		1	Ι	ST	PORTF<7> data input.

TABLE 11-11: PORTF FUNCTIONS

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input; x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

TABLE 11-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0
PADCFG1	RDPU	REPU	RFPU ⁽¹⁾	RGPU ⁽¹⁾	—	_	_	CTMUDS

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTF.

Note 1: These bits are unimplemented on 28-pin devices, read as '0'.

17.2 Timer4 Interrupt

The Timer4 module has an eight-bit Period register, PR4, that is both readable and writable. Timer4 increment from 00h until it matches PR4 and then resets to 00h on the next increment cycle. The PR4 register is initialized to FFh upon Reset.



17.3

Output of TMR4

as is the Timer2 output.

The outputs of TMR4 (before the postscaler) are used

only as a PWM time base for the ECCP modules. They

are not used as baud rate clocks for the MSSP module

FIGURE 17-1: TIMER4 BLOCK DIAGRAM

TABLE 17-1: REGISTERS ASSOCIATED WITH TIMER4 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP		CCP5IP	CCP4IP	CCP3IP
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF
PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE
TMR4	Timer4 Reg	ister						
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
PR4	Timer4 Period Register							
PMD1	PSPMD	CTMUMD	ADCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer4 module.

EXAMPLE 18-1: SETUP FOR CTMU CALIBRATION ROUTINES

```
#include "pl8cxxx.h"
void setup(void)
{ //CTMUCON - CTMU Control register
  CTMUCONH = 0 \times 00;
                         //make sure CTMU is disabled
  CTMUCONL = 0x90;
  //CTMU continues to run when emulator is stopped,CTMU continues
  //to run in idle mode, Time Generation mode disabled, Edges are blocked
  //No edge sequence order, Analog current source not grounded, trigger
  //output disabled, Edge2 polarity = positive level, Edge2 source =
  //source 0, Edgel polarity = positive level, Edgel source = source 0,
  // Set Edge status bits to zero
   //CTMUICON - CTMU Current Control Register
  CTMUICON = 0 \times 01;
                         //0.55uA, Nominal - No Adjustment
//Setup AD converter;
TRISA=0x04;
                         //set channel 2 as an input
  // Configured AN2 as an analog channel
  // ANCON1
  ANCON1 = 0 \times 04;
  // ADCON1
  ADCON2bits.ADFM=1;
                       // Result format 1= Right justified
                        // Acquisition time 7 = 20TAD 2 = 4TAD 1=2TAD
  ADCON2bits.ACQT=1;
  ADCON2bits.ADCS=2;
                         // Clock conversion bits 6= FOSC/64 2=FOSC/32
  // ADCON1
                         // Vref+ = AVdd
  ADCON1bits.VCFG0 =0;
  ADCON1bits.VCFG1 =0;
                         // Vref+ = AVdd
                        // Vref- = AVss
  ADCON1bits.VNCFG = 0;
                         // Select ADC channel
  ADCON1bits.CHS=2;
  ADCON0bits.ADON=1;
                         // Turn on ADC
}
```

19.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE bit (PIE4<x>) clear to avoid false interrupts and should clear the flag bit, CCPxIF, following any such change in operating mode.

19.2.4 CCP PRESCALER

There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCPxM<3:0>). Whenever the CCP module is turned off, or the CCP module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Doing that also will not clear the prescaler counter – meaning the first capture may be from a non-zero prescaler.

Example 19-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 19-1: CHANGING BETWEEN CAPTURE PRESCALERS

CLRF	CCPxCON	;	Turn CCP module off
MOVLW	NEW_CAPT_PS	;	Load WREG with the
		;	new prescaler mode
		;	value and CCP ON
MOVWF	CCPxCON	;	Load CCPxCON with
		;	this value

19.2.5 CAN MESSAGE TIME-STAMP (CCP2 ONLY)

For CCP2, only the CAN capture event occurs when a message is received in any of the receive buffers. When configured, the CAN module provides the trigger to the CCP2 module to cause a capture event. This feature is provided to "time-stamp" the received CAN messages.

This feature is enabled by setting the CANCAP bit of the CAN I/O Control register (CIOCON<4>). The message receive signal from the CAN module then takes the place of the events on RC2/CCP2.

If this feature is selected, then four different capture options for CCP2M<3:0> are available:

- 0100 Every time a CAN message is received
- 0101 Every time a CAN message is received
- 0110 Every 4th time a CAN message is received
- 0111 Capture mode, every 16th time a CAN message is received

20.4.2.1 Direction Change in Full-Bridge Mode

In the Full-Bridge mode, the P1M1 bit in the CCP1CON register allows users to control the forward/reverse direction. When the application firmware changes this direction control bit, the module will change to the new direction on the next PWM cycle.

A direction change is initiated in software by changing the P1M1 bit of the CCP1CON register. The following sequence occurs prior to the end of the current PWM period:

- The modulated outputs (P1B and P1D) are placed in their inactive state.
- The associated unmodulated outputs (P1A and P1C) are switched to drive in the opposite direction.
- PWM modulation resumes at the beginning of the next period.

For an illustration of this sequence, see Figure 20-10.

The Full-Bridge mode does not provide a dead-band delay. As one output is modulated at a time, a dead-band delay is generally not required. There is a situation where a dead-band delay is required. This situation occurs when both of the following conditions are true:

- The direction of the PWM output changes when the duty cycle of the output is at or near 100%.
- The turn-off time of the power switch, including the power device and driver circuit, is greater than the turn-on time.

Figure 20-11 shows an example of the PWM direction changing from forward to reverse, at a near 100% duty cycle. In this example, at time, t1, the P1A and P1D outputs become inactive, while the P1C output becomes active. Since the turn-off time of the power devices is longer than the turn-on time, a shoot-through current will flow through power devices, QC and QD (see Figure 20-8), for the duration of 't'. The same phenomenon will occur to power devices, QA and QB, for PWM direction change from reverse to forward.

If changing PWM direction at high duty cycle is required for an application, two possible solutions for eliminating the shoot-through current are:

- Reduce PWM duty cycle for one PWM period before changing directions.
- Use switch drivers that can drive the switches off faster than they can drive them on.

Other options to prevent shoot-through current may exist.

FIGURE 20-10: EXAMPLE OF PWM DIRECTION CHANGE



(1/FOSC) • TMR2 Prescale Value.

22.4 EUSARTx Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTAx<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTAx<4>). In addition, enable bit, SPEN (RCSTAx<7>), is set in order to configure the TXx and RXx pins to CKx (clock) and DTx (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CKx line. Clock polarity is selected with the TXCKP bit (BAUDCONx<4>). Setting TXCKP sets the Idle state on CKx as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

22.4.1 EUSARTx SYNCHRONOUS MASTER TRANSMISSION

The EUSARTx transmitter block diagram is shown in Figure 22-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREGx (if available).

Once the TXREGx register transfers the data to the TSR register (occurs in one TcY), the TXREGx is empty and the TXxIF flag bit is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxIE. TXxIF is set regardless of the state of enable bit, TXxIE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREGx register.

While flag bit, TXxIF, indicates the status of the TXREGx register, another bit, TRMT (TXSTAx<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user must poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

- Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
- 2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
- 3. If interrupts are desired, set enable bit, TXxIE.
- 4. If 9-bit transmission is desired, set bit, TX9.
- 5. Enable the transmission by setting bit, TXEN.
- 6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
- 7. Start transmission by loading data to the TXREGx register.
- 8. If using interrupts, ensure that the GIE and PEIE bits (INTCON<7:6>) are set.



FIGURE 22-11: SYNCHRONOUS TRANSMISSION

REGISTER 23-9: ANCON1: A/D PORT CONFIGURATION REGISTER 1

U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	ANSEL14 ⁽¹⁾	ANSEL13 ⁽¹⁾	ANSEL12 ⁽¹⁾	ANSEL11 ⁽¹⁾	ANSEL10	ANSEL9	ANSEL8
bit 7							bit 0

1								
Legend:								
R = Readable bit		W = Writable bit	U = Unimplemented bit	, read as '0'				
-n = Value a	t POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown				
bit 7	Unimpleme	nted: Read as '0'						
bit 6	ANSEL14:	RD3/C2INB Pin Analog Er	nable bit ⁽¹⁾					
	1 = Pin is co 0 = Pin is co	ifigured as an analog channel; digital input is disabled and any inputs read as '0' ifigured as a digital port						
bit 5	ANSEL13:	RD2/C2INA Pin Analog Er	nable bit ⁽¹⁾					
	1 = Pin is co 0 = Pin is co	onfigured as an analog cha onfigured as a digital port	annel; digital input is disabled	and any inputs read as '0'				
bit 4	ANSEL12:	RD1/C1INB Pin Analog Er	nable bit ⁽¹⁾					
	1 = Pin is co	onfigured as an analog cha	annel; digital input is disabled	and any inputs read as '0'				

0 = Pin is configured as a digital port

bit 3 ANSEL11: RD0/C1INA Pin Analog Enable bit⁽¹⁾

- 1 = Pin is configured as an analog channel: digital input disabled and any inputs read as '0'
 - 0 = Pin is configured as a digital port

bit 2-0 **ANSEL11<10:8>:** Analog Port Configuration bits (AN10 through AN8)

- 1 = Pin is configured as an analog channel; digital input is disabled and any inputs read as '0'
 0 = Pin configured as a digital port
- **Note 1:** AN14 through AN11 and AN7 to AN5 are implemented only on 40/44-pin and 64-pin devices. For 28-pin devices, the corresponding ANSELx bits are still implemented for these channels, but have no effect.

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AVDD and AVSS) or the voltage level on the RA3/VREF+/AN3 and RA2/VREF-/AN2 pins. VREF+ has two additional internal voltage reference selections: 2.0V and 4.1V.

The A/D Converter can uniquely operate while the device is in Sleep mode. To operate in **Sleep**, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

Each port pin associated with the A/D Converter can be configured as an analog input or a digital I/O. The ADRESH and ADRESL registers contain the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/DONE bit (ADCON0<1>) is cleared and the A/D Interrupt Flag bit, ADIF (PIR1<6>), is set.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. The value in the ADRESH:ADRESL register pair is not modified for a Power-on Reset. These registers will contain unknown data after a Power-on Reset.

The block diagram of the A/D module is shown in Figure 23-4.

REGISTER 27-34: BnDLC: TX/RX BUFFER 'n' DATA LENGTH CODE REGISTERS IN RECEIVE MODE $[0 \le n \le 5, TXnEN (BSEL < n >) = 0]^{(1)}$

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0
bit 7						-	bit 0
Legend:							
R = Readable I	bit	W = Writable	bit	U = Unimpler	nented bit, rea	d as '0'	
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unk	nown
bit 7	Unimplemen	ted: Read as '	0'				
bit 6	RXRTR: Rece	eiver Remote T	ransmission	Request bit			
	1 = This is a r 0 = This is no	emote transmi t a remote tran	ssion request smission req	uest			
bit 5	RB1: Reserve	ed bit 1					
	Reserved by	CAN Spec and	read as '0'.				
bit 4	RB0: Reserve	ed bit 0					
	Reserved by	CAN Spec and	read as '0'.				
bit 3-0	DLC<3:0>: D	ata Length Coo	de bits				
	1111 = Rese	rved					
	1110 = Reser	rved					
	1101 = Rese	rved					
	1100 = Rese i	rved					
	1011 = Resei	rved					
	1010 = Resei	rved					
	1001 = Resei	Ved	-				
	1000 = Data	length = 8 byte	S				
	0111 - Data	engli – 7 byle ength – 6 byte	5				
	0110 = Data	length = 5 byte	5 C				
	0101 = Data	length = 4 byte	3 9				
	0011 = Data	length = 3 byte	s				
	0010 = Data	enath = 2 byte	S				
	0001 = Data	length = 1 byte					
	0000 = Data	length = 0 byte	S				

Note 1: These registers are available in Mode 1 and 2 only.

BTG	Bit Toggle	f		BOV		Branch if C	Overflow		
Syntax:	BTG f, b {,a	a}		Synta	ax:	BOV n			
Operands:	$0 \le f \le 255$			Oper	ands:	-128 ≤ n ≤ 127			
	0 ≤ b < 7 a ∈ [0,1]			Oper	ation:	if Overflow (PC) + 2 + 2	bit is '1', 2n \rightarrow PC		
Operation:	$(\overline{f}) \to f$			Statu	is Affected:	None			
Status Affected:	None			Enco	odina:	1110	0100 nn:	nn nnnn	
Encoding:	0111	bbba ff	ff ffff	Desc	ription:	If the Overf	ow bit is '1'. tl	nen the	
Description:	Bit 'b' in da inverted	ta memory loc	ation 'f' is			program wi	I branch.		
	lf 'a' is '0', t lf 'a' is '1', t GPR bank.	he Access Ba he BSR is use	nk is selected. d to select the			added to the incremente instruction,	Plement num PC. Since th to fetch the the new addre	iber, '2n', is ie PC will have next ess will be	
	If 'a' is '0' a	nd the extend	ed instruction			PC + 2 + 2r two-cycle ir	 I his instruction 	tion is then a	
	set is enab in Indexed	led, this instru Literal Offset A	ction operates	Word	ls:	1			
	mode wher	never f ≤ 95 (5	Fh). See	Cycle	es:	1(2)			
	Section 29 Bit-Oriente Literal Offs	2.3 "Byte-Or ed Instruction set Mode" for	iented and is in Indexed details.	Q C If Ju	ycle Activity: ımp:				
Words:	1				Q1	Q2	Q3	Q4	
Cvcles:	1				Decode	Read literal	Process	Write to PC	
Q Cycle Activity					No	n No	Data	No	
Q1	Q2	Q3	Q4		operation	operation	operation	operation	
Decode	Read	Process	Write	lf No	Jump:				
	register 'f'	Data	register 'f'		Q1	Q2	Q3	Q4	
					Decode	Read literal	Process	No	
Example:	BTG P	ORTC, 4, ()			'n'	Data	operation	
Before Instruc	ction:								
PORTC	= 0111	0101 [75h]		Exan	<u>nple:</u>	HERE	BOV Jump		
PORTC	on: = 0110 (0101 [65h]			Before Instruct PC	ction = ad	dress (HERE)	
					If Overflo PC	ow = 1; = ad	dress (Jumo)	
					lf Overflo PC	ow = 0; = ad	dress (HERE	+ 2)	

CLRF		Clear f				с	LRWDT		Clear \	Vato	hdog Ti	imer		
Syntax:		CLRF f{,;	a}			S	yntax:		CLRW	DT				
Operands	3 :	0 ≤ f ≤ 255 a ∈ [0,1]				0	perands:		None	<u>\</u> \//	דר			
Operatior	1:	$\begin{array}{l} 000h \rightarrow f, \\ 1 \rightarrow Z \end{array}$				0			$000h = 000h = 1 \rightarrow T(1)$	→ W[→ W[→ →	DT, DT posts	scaler,		
Status Aff	fected:	Z			1	1 0	atus Affectes	J.	$1 \rightarrow Pl$	5				
Encoding	:	0110	101a	ffff	ffff	5	atus Anecteo	1:	TO, PL	,			_	
Descriptio	on:	Clears the	contents	of the spe	ecified	E	ncoding:		000		0000	000		0100
		If 'a' is '0', t If 'a' is '1', t GPR bank.	he Acces he BSR i	ss Bank is is used to	s selected. select the	D	escription.		Watche post <u>sca</u> and PE	$\log \frac{1}{2}$	Timer. It of the W e set.	also res DT. Sta	sets th tus bit	e s, TO
		lf 'a' is '0' a	nd the e	xtended ir	nstruction	W	/ords:		1					
		set is enabl	ed, this i	instructior	operates	С	ycles:		1					
		in Indexed	Literal Of	ffset Addr 95 (5Eb)	essing	(Q Cycle Activi	ity:						
		Section 29	.2.3 "By	te-Orient	ed and		Q1		Q2		Q	3	C	<u>)</u> 4
		Bit-Oriente Literal Offs	ed Instru set Mode	ictions in e" for deta	Indexed ails.		Decode	е	No operatio	on	Proce Dat	ess a	N opera	o ation
Words:		1												
Cycles:		1				<u>E</u>	xample:		CLRWD	Г				
Q Cycle	Activity:						Before Ins	structi	on		0			
,	Q1	Q2	Q3	3	Q4		WD I After Instr	I Cour	nter	=	?			
D	ecode	Read register 'f'	Proce Data	ess a re	Write gister 'f']	WDT WDT	T Cour T Post	nter scaler	= =	00h 0			
							TO			=	1			
Example:		CLRF	FLAG_	_REG,1			PD			=	1			
Befc After	ore Instruc FLAG_RI r Instructic FLAG_RI	tion EG = 5A on EG = 00	.h h											

IORL	w	V Inclusive OR Literal with W								
Synta	ax:	IORLW k	IORLW k							
Oper	ands:	$0 \le k \le 25$	5							
Oper	ation:	(W) .OR. k	$x \rightarrow W$							
Statu	s Affected:	N, Z								
Enco	oding:	0000	1001	kkk	k	kkkk				
Desc	escription: The contents of W are ORed with the eight-bit literal 'k'. The result is placed in W.									
Word	ls:	1								
Cycle	es:	1								
QC	ycle Activity:									
	Q1	Q2	Q3	3		Q4				
	Decode	Read literal 'k'	Proce Data	ess a	N	/rite to W				
Exan	nple:	IORLW	35h							
	Before Instruc W	tion = 9Ah								

BFh

=

After Instruction W

IOR	NF	Inclusive	OR W wit	h f					
Synt	ax:	IORWF	f {,d {,a}}						
Oper	rands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$							
Oper	ration:	(W) .OR. (1	(W) .OR. (f) \rightarrow dest						
Statu	is Affected:	N, Z	N, Z						
Enco	oding:	0001	00da	ffff	ffff				
Desc	cription:	Inclusive C '0', the res the result is (default).	Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).						
		If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.							
		If 'a' is '0' a set is enab in Indexed mode when Section 29 Bit-Oriente Literal Off	and the ex led, this i Literal Of never f < 9 9.2.3 "By ed Instru set Mode	tended i nstructio fset Add 95 (5Fh) te-Orien ctions in e ² for def	instruction n operates ressing . See ted and n Indexed tails.				
Word	ds:	1							
Cycle	es:	1							
QC	vcle Activity:								
	Q1	Q2	Q3		Q4				
	Decode	Read register 'f'	Proce Data	ss a de	Write to estination				
<u>Exar</u>	<u>nple:</u> Before Instruc RESULT W	IORWF R tion = 13h = 91h	ESULT,	0, 1					

13h 93h

After Instruction RESULT = W =

_	1	1		i	i	t	
Param. No.	Symbol	Charac	teristic	Min	Max	Units	Conditions
100	Thigh	Clock High	100 kHz mode	2(Tosc)(BRG + 1)	_	—	
		Time	400 kHz mode	2(Tosc)(BRG + 1)	—	_	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	_	
101	TLOW	Clock Low Time	100 kHz mode	2(Tosc)(BRG + 1)	—	_	
			400 kHz mode	2(Tosc)(BRG + 1)	—	_	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	_	
102	TR	SDA and SCL	100 kHz mode		1000	ns	CB is specified to be from
		Rise Time	400 kHz mode	20 + 0.1 Св	300	ns	10 to 400 pF
			1 MHz mode ⁽¹⁾		300	ns	
103	TF	SDA and SCL	100 kHz mode		300	ns	CB is specified to be from
		Fall Time	400 kHz mode	20 + 0.1 Св	300	ns	10 to 400 pF
			1 MHz mode ⁽¹⁾	_	100	ns	
90	TSU:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	_	Only relevant for Repeated
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_	_	Start condition
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	_	_	
91	THD:STA	Start Condition	100 kHz mode	2(Tosc)(BRG + 1)	_	_	After this period, the first
		Hold Time	400 kHz mode	2(Tosc)(BRG + 1)	_	_	clock pulse is generated
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	_	_	
106	THD:DAT	Data Input	100 kHz mode	0	—	_	
		Hold Time	400 kHz mode	0	0.9	μs	
			1 MHz mode ⁽¹⁾	_	μS	ns	
107	TSU:DAT	Data Input	100 kHz mode	250	_	ns	(Note 2)
		Setup Time	400 kHz mode	100	_	ns	
			1 MHz mode ⁽¹⁾	_	_	ns	
92	Tsu:sto	Stop Condition	100 kHz mode	2(Tosc)(BRG + 1)	—	_	
		Setup Time	400 kHz mode	2(Tosc)(BRG + 1)	_		
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	_	_	
109	ΤΑΑ	Output Valid	100 kHz mode	_	3500	ns	
		from Clock	400 kHz mode	_	1000	ns	
			1 MHz mode ⁽¹⁾	_	_	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	_	μs	Time the bus must be free
			400 kHz mode	1.3	_	μS	before a new transmission
			1 MHz mode ⁽¹⁾	_	_	μS	can start
D102	Св	Bus Capacitive I	oading	—	400	pF	
			-				1

TABLE 31-22:	MSSP I ² C™	BUS DATA	REQUIREMENTS

Note 1: Maximum pin capacitance = 10 pF for all I^2C^{TM} pins.

2: A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but Parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, Parameter #102 + Parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCL line is released.

TBLPTR (Table Pointer) Register	132
Erase Sequence	134
Erasing	134
Operation During Code-Protect	137
Reading	133
Table Pointer	
Boundaries Based on Operation	132
Table Pointer Boundaries	132
Table Reads and Table Writes	129
Write Sequence	135
Writing	135
Protection Against Spurious Writes	137
Unexpected Termination	137
Write Verify	137
FSCM. See Fail-Safe Clock Monitor.	

G

GOTO	 	 	 504

Н

Hardware Multiplier	145
8 x 8 Multiplication Algorithms	145
Oneration	145
Operation	145
Performance Comparison (table)	145
High/Low-Voltage Detect	385
Applications	389
Associated Registers	390
Current Consumption	387
Effects of a Reset	390
Operation	386
During Sleep	390
Setup	387
Start-up Time	387
Typical Application	389
HLVD. See High/Low-Voltage Detect.	385

I

I/O Descriptions	
PIC18F2XK80	18
PIC18F4XK80	24
PIC18F6XK80	
I/O Ports	171
Analog/Digital Ports	174
Open-Drain Outputs	173
Output Pin Drive	171
Pin Capabilities	171
Port Slew Rate	174
Pull-up Configuration	171
I ² C Mode (MSSP)	
Acknowledge Sequence Timing	325
Associated Registers	331
Baud Rate Generator	
Bus Collision	
During a Repeated Start Condition	329
During a Stop Condition	330
Clock Arbitration	
Clock Stretching	
10-Bit Slave Receive Mode (SEN = 1)	
10-Bit Slave Transmit Mode	
7-Bit Slave Receive Mode (SEN = 1)	
7-Bit Slave Transmit Mode	
Clock Synchronization and the CKP bit	
Effects of a Reset	
General Call Address Support	
I ² C Clock Rate w/BRG	
Master Mode	

-		- · -
C	Operation	317
R	Perention	322
		022
R	Repeated Start Condition Timing	321
S	Start Condition Timing	320
-		020
I	ransmission	322
Multi-N	Master Communication Bus Collision and Arb	itra-
		000
ti	on	320
Multi-N	Vaster Mode	326
0	tion.	204
Opera	<u>tion</u>	301
Read/	Write Bit Information (R/W Bit)	304
Dogiat	toro (206
Regisi	EIS	290
Serial	Clock (RC3/REFO//SCL/SCK)	304
Slave	Mada	201
Slave		301
A	ddress Masking Modes	
	E Dit	202
	J-DIL	30Z
	7-Bit	303
^	ddrooping	201
A	luuressing	301
R	Reception	304
т	ranamiasian	204
1	14115111551011	304
Sleep	Operation	326
Ston C	andition Timing	225
Stop C		325
ID Location	s	482
Idlo Modeo		70
Idle Modes		. 70
INCF		504
INCER7		EOE
INCF52		505
In-Circuit D	ebuager	482
	arial Dragonaniag (ICCD) 457	400
In-Circuit Se	enal Programming (ICSP)	40Z
Indexed Lite	eral Offset Addressing	
and Ct	tenderd DIC19 Instructions	E 2 0
and St		530
Indexed Lite	eral Offset Mode	530
Indiract Add	drooping	104
mullect Aut	Jiessing	124
INFSNZ		505
Initialization	Conditions for all Desisters	22
Initialization	n Conditions for all Registers 88	-??
Initialization	n Conditions for all Registers	-?? 106
Initialization	n Conditions for all Registers	106
Initialization Instruction Clocki	n Conditions for all Registers	—?? 106 106
Initialization Instruction Clockin Flow/F	n Conditions for all Registers	-?? 106 106 106
Initialization Instruction (Clocking Flow/F	n Conditions for all Registers	106 106 106 483
Initialization Instruction (Clockin Flow/F Instruction (n Conditions for all Registers	-?? 106 106 106 483
Initialization Instruction (Clockin Flow/F Instruction S ADDL	n Conditions for all Registers	-?? 106 106 106 483 489
Initialization Instruction (Clockin Flow/F Instruction (ADDL)	n Conditions for all Registers	-?? 106 106 106 483 489 489
Initialization Instruction (Clockii Flow/F Instruction (ADDL) ADDL	n Conditions for all Registers	-?? 106 106 106 483 489 489
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW	n Conditions for all Registers	-?? 106 106 106 483 489 489 531
Initialization Instruction (Clockin Flow/F Instruction (ADDL ADDW ADDW	n Conditions for all Registers	?? 106 106 106 483 489 489 531 490
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW	n Conditions for all Registers	?? 106 106 106 483 489 489 531 490
Initialization Instruction (Clockin Flow/F Instruction (ADDL ADDW ADDW ADDW ANDL	n Conditions for all Registers	?? 106 106 483 489 489 531 490 490
Initialization Instruction (Clockin Flow/F Instruction (ADDL ADDW ADDW ADDW ANDL ANDL	n Conditions for all Registers	?? 106 106 483 489 489 531 490 490 491
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ANDL) ANDL	n Conditions for all Registers	?? 106 106 483 489 489 531 490 490 491
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ANDL) ANDW BC	Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 W. 9 /F 10 /F (Indexed Literal Offset Mode) 10 /FC 10 W. 10	5-?? 106 106 483 489 489 531 490 490 491 491
Initialization Instruction (Clockin Flow/F Instruction (ADDL ADDW ADDW ADDW ADDW ANDL ANDW BC BCF	n Conditions for all Registers	5-?? 106 106 106 483 489 489 531 490 490 491 491 492
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDU ADDW ADDW ADDW ANDU ANDU BC BCF	n Conditions for all Registers	5-?? 106 106 106 483 489 489 531 490 490 491 491 492
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ANDL) ANDW BC BCF BN	n Conditions for all Registers	5-?? 106 106 483 489 489 531 490 490 491 491 492 492
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDU ADDW ADDW ADDW ANDL ANDW BC BN BNC	n Conditions for all Registers	?? 106 106 483 489 531 490 490 491 491 492 492 493
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDU ADDW ADDW ADDW ADDW ADDW BC BCF BN BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 490 491 491 492 492 493
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ADDW BC BCF BN BNC BNN	n Conditions for all Registers	-??? 106 106 483 489 489 531 490 490 491 492 492 492 493 493
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCC BNC BNC BNN BNN	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 491 492 492 493 493 494
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDU ADDW ADDW ADDW ADDW ADDW BC BCF BNN BNN BNN BNN	n Conditions for all Registers	-??? 106 106 483 489 489 531 490 491 491 492 492 493 493 494
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ADDW BC BCF BN BNC BNN BNOV BNZ	n Conditions for all Registers	-??? 106 106 483 489 489 531 490 491 492 493 493 493 494 494
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BC BNC BNC BNC BNC BNC BNOV BNZ BOV	n Conditions for all Registers	-??? 106 106 483 489 531 490 491 491 492 493 494 493 494 494 494
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDW ADDW ADDW ADDW ADDW ADDW BC BCF BNN BNN BNN BNN BNV	n Conditions for all Registers	?? 106 106 483 489 531 490 491 491 492 493 493 494 493 494 494
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ADDW ADDW BC BNC BNC BNN BNOV BNZ BNOV BNZ BNZ	Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 VF 9 VF (Indexed Literal Offset Mode) 9 VFC 9 VF 9 VF 9 VFC 9 VF 9	-?? 106 106 483 489 531 490 491 492 493 494 492 493 494 494 494 494
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC	n Conditions for all Registers	-?? 106 106 483 489 531 490 491 492 493 494 492 493 494 494 494 494 495
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDU ADDW ADDW ADDW ADDW ADDW ADDW	h Conditions for all Registers 88 Cycle	-??? 106 106 483 489 531 490 491 492 493 494 494 494 494 494 495 531
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ADDW ADDW BC BNC BNC BNN BNC BNN BNOV BNZ BNOV BNZ	n Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 W. 9 VF 9 VF (Indexed Literal Offset Mode) 9 VF 9 VF 9 W. 9 VF 9 With the set of t	-?? 106 106 483 489 489 531 490 491 491 492 493 494 494 494 494 495 531
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BC BCF BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 491 492 493 494 494 494 494 495 531 495 531
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDW ADDW ADDW ADDW ADDW ADDW ADDW	n Conditions for all Registers 88 Cycle	-?? 106 106 483 489 489 531 490 491 491 492 493 494 493 494 494 495 531 495 531
Initialization Instruction (Clockin Flow/F Instruction 3 ADDL ADDU ADDW ADDW ADDW ADDW BC BN BNC	n Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 W. 9 VF 9 VF (Indexed Literal Offset Mode) 9 VF 9 W. 9 VFC 9 W. 9 VF 9 Main and the set of the	-?? 106 106 483 489 489 531 490 491 492 493 494 492 493 494 494 495 531 495 531 496 496
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW ADDW	n Conditions for all Registers 88 Cycle	-?? 106 106 483 489 489 531 490 491 491 492 493 494 494 494 494 495 531 496 497
Initialization Instruction (Clockin Flow/F Instruction (ADDL) ADDW ADDW ADDW ADDW ADDW ADDW ADDW BC BCF BNC BNC BNN BNC BNC BNN BNC B	n Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 W 9 VF 10 VFC 9 W 9 VFC 9 W 9 VFC 9 W 9 VF 9 Indexed Literal Offset Mode) 9 C 9 S 9	-?? 106 106 483 489 489 531 490 491 492 493 494 493 494 494 495 531 496 496 497
Initialization Instruction (Clockin Flow/F Instruction 3 ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC	n Conditions for all Registers 88 Cycle 9 ng Scheme 9 Pipelining 9 Set 9 VF 9 VF (Indexed Literal Offset Mode) 9 VFC 9 VFC 9 VF 9	-?? 106 106 483 489 489 531 490 491 492 493 494 494 494 494 495 531 496 496 497 498
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 491 492 493 494 494 494 494 495 531 496 496 497 498 498
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDW ADDW ADDW ADDW ADDW ADDW ADDW	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 493 494 494 495 531 496 496 497 498 498
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC	n Conditions for all Registers	-?? 106 106 483 489 531 490 491 492 493 494 492 493 494 494 495 531 496 495 531 496 497 498 498
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 494 493 494 494 495 531 494 495 531 495 531 496 495 495 495 496 497 498 499 499
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDW ADDW ADDW ADDW ADDW ADDW ADDW	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 492 493 494 494 495 531 496 495 531 496 497 498 499 499 531
Initialization Instruction (Clockin Flow/F Instruction 3 ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCC BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 491 492 493 494 494 495 531 496 495 531 496 497 498 499 499 500
Initialization Instruction (Clockin Flow/F Instruction S ADDL ADDW ADDW ADDW ADDW ADDW ADDW BCF BNC CLRW COMF	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 492 493 494 494 495 531 494 495 531 495 531 496 497 498 499 499 500 500
Initialization Instruction (Clockin Flow/F Instruction (ADDU ADDW ADDW ADDW ADDW ADDW ADDW ADDW	n Conditions for all Registers 88 Cycle	-?? 106 106 483 489 489 531 490 491 492 493 494 497 495 531 496 497 495 531 496 497 498 499 500 500 500
Initialization Instruction (Clockin Flow/F Instruction 3 ADDL ADDW ADDW ADDW ADDW ADDW BCW BCC BNC BNC BNC BNC BNC BNC BNC BNC BNC	n Conditions for all Registers	-?? 106 106 483 489 489 531 490 491 492 493 494 492 493 494 494 495 531 496 495 531 496 497 498 499 499 500 500 500 501

BnDLC (TX/RX Buffer n Data Length Code in Transmit Mode) 420
BnDm (TX/RX Buffer n Data Field Byte m in Receive
BnDm (TX/RX Buffer n Data Field Byte m in Transmit
Mode)418 BnEIDH (TX/RX Buffer n Extended Identifier, High Byte
in Receive Mode)417 BnEIDH (TX/RX Buffer n Extended Identifier, High Byte
in Transmit Mode)417 BnEIDL (TX/RX Buffer n Extended Identifier, Low Byte in
Receive Mode)
BnSIDH (TX/RX Buffer n Standard Identifier, High Byte
BnSIDH (TX/RX Buffer n Standard Identifier. High Byte
in Transmit Mode)
BnSIDL (TX/RX Buffer n Standard Identifier, Low Byte in
Receive Mode)
BRGCON1 (Baud Rate Control 2) 431
BRGCON3 (Baud Rate Control 3)
BSEL0 (Buffer Select 0)
CANCON (CAN Control)
CANSTAT (CAN Status)
CCP1CON (Enhanced Capture/Compare/PWM1 Con-
trol)
CCPPRXL (CCPX Period Low Byte)
CCPTMPS (CCP Timer Select) 254, 267
CCPxCON (CCPx Control CCP2-CCP5) 253
CIOCON (CAN I/O Control)
CMSTAT (Comparator Status)
CMxCON (Comparator Control x)
COMSTAT (CAN Communication Status) 399
CONFIG1H (Configuration 1 High) 460
CONFIG1L (Configuration 1 Low)
CONFIG2H (Configuration 2 Hign)
CONFIG2L (Configuration 2 Low)
CONFIG4L (Configuration 4 Low)
CONFIG5H (Configuration 5 High)
CONFIG5L (Configuration 5 Low)
CONFIG6H (Configuration 6 High) 468
CONFIG6L (Configuration 6 Low)
CONFIG7H (Configuration 7 High)
CONFIG/L (Configuration / Low)
CTMUCONI (CTMU Control Low) 237
CTMUICON (CTMU Current Control)
CVRCON (Comparator Voltage Reference Control). 381
DEVID1 (Device ID 1)
DEVID2 (Device ID 2)471
ECANCON (Enhanced CAN Control)
ECCP1AS (ECCP1 Auto-Shutdown Control)
ECCP IDEL (EIIIanced PWW Control 1)
FECON1 (FEPROM Control 1) 131
HLVDCON (High/Low-Voltage Detect Control)
INTCON (Interrupt Control)149
INTCON2 (Interrupt Control 2)150
INTCON3 (Interrupt Control 3) 151
IOCB (Interrupt-on-Change PORTB Control)
IPR1 (Peripheral Interrupt Priority 1)
IPR3 (Peripheral Interrupt Priority 3) 164

IPR4 (Peripheral Interrupt Priority 4)	
	. 165
IPR5 (Peripheral Interrupt Priority 5) 166	436
	, 400
MDCARH (Modulation High Carrier Control)	. 203
MDCARL (Modulation Low Carrier Control)	204
	0 1
MDCON (Modulation Control Register)	. 201
MDSRC (Modulation Source Control)	202
	400
MSELU (Mask Select 0)	. 426
MSEL1 (Mask Select 1)	427
MSEL2 (Mask Select 2)	. 428
MSEL3 (Mask Select 3)	429
ODCON (Peripheral Open-Drain Control)	. 173
OSCCON (Oscillator Control)	53
OSCCON2 (Oscillator Control 2) 54	, 225
OSCTUNE (Oscillator Tuning)	55
PADCFG1 (Pad Configuration)	. 172
PIE1 (Perinheral Interrunt Enable 1)	157
	. 107
PIE2 (Peripheral Interrupt Enable 2)	. 158
PIE3 (Perinheral Interrunt Enable 3)	150
	. 100
PIE4 (Peripheral Interrupt Enable 4)	. 160
PIE5 (Perinheral Interrunt Enable 5) 161	435
	, 400
PIR1 (Peripheral Interrupt Request (Flag) 1)	. 152
PIR2 (Perinheral Interrunt Request (Flag) 2)	153
	. 100
PIR3 (Peripheral Interrupt Request (Flag) 3)	. 154
PIR4 (Perinheral Interrunt Request (Flag) 4)	155
Tiller (Teripheral Interrupt Request (Tag) +)	. 155
PIR5 (Peripheral Interrupt Request (Flag) 5) 156	6, 434
PMD0 (Perinheral Module Disable 0)	75
	75
PMD1 (Peripheral Module Disable 1)	74
PMD2 (Perinheral Module Disable 2)	73
	75
PSPCON (Parallel Slave Port Control)	. 193
PSTR1CON (Pulse Steering Control)	283
	. 205
RCON (Reset Control) 80	, 167
RCSTAx (Receive Status and Control)	335
	. 000
REFOCON (Reference Oscillator Control)	62
RXB0CON (Receive Buffer 0 Control)	406
	. 400
RXB1CON (Receive Buffer 1 Control)	. 408
RXBnDLC (Receive Buffer n Data Length Code)	411
RXBnDLC (Receive Buffer n Data Length Code)	. 411
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m)	. 411 . 411
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnFIDH (Receive Buffer n Extended Identifier	. 411 . 411 High
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier,	. 411 . 411 High
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte)	. 411 . 411 High . 410
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier	. 411 . 411 High . 410
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier,	. 411 . 411 High 410 Low
RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte)	. 411 . 411 High . 410 Low . 410
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, 	. 411 . 411 High . 410 Low . 410 High
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, 	. 411 . 411 High . 410 Low . 410 High
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) 	. 411 High . 410 Low . 410 High . 409
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) 	. 411 . 411 High 410 Low 410 High 409
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) 	. 411 High . 410 Low . 410 High . 409 Low
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) 	411 High 410 Low 410 High 409 Low 410
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Frror Count) 	. 411 High . 410 Low . 410 High . 409 Low . 410 412
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte). RXBnSIDL (Receive Buffer n Standard Identifier, Byte). RXBnSIDL (Receive Error Count) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 412
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Error Count) RXFBCONn (Receive Filter Buffer Control n) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 412 . 425
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firther Count) RXFBCONn (Receive Filter Control n) 	. 411 High . 410 Low . 410 High . 409 Low . 409 Low . 410 . 412 . 425 424
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) 	. 411 High . 410 Low . 410 High . 409 Low . 409 Low . 410 . 412 . 425 . 424
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Acceptance Filter n Extended 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 409 Low . 410 . 425 . 424 Iden-
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firor Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 409 Low . 410 . 425 . 424 Iden- 422
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 409 Low . 410 . 425 . 424 Iden- . 422
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Error Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 409 Low . 410 . 425 . 424 Iden- . 422 Iden-
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter Neutron Count) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFCONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Extended tifier, Lew Butc) 	. 411 High . 410 Low . 410 High . 409 Low . 409 Low . 410 . 425 . 424 Iden- . 422 Iden- . 422
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter Notice Control n) RXFBCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 412 . 425 . 424 Iden . 422 Iden . 422
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Error Count) RXFBCONn (Receive Filter Buffer Control n) RXFRCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Standard 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 409 Low . 410 . 425 . 424 Iden . 422 Iden . 422 Iden
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter Neutron Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFCONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Standard tifier, Low Byte) 	. 411 . 411 High . 410 Low . 410 High . 409 Low . 410 . 409 . 409 . 410 . 422 Iden- . 422 Iden- . 422 Iden- . 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Filter Notice Control n) RXFBCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier, High Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 420 Iden- 422 Iden- 422 Iden- 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFRCONn (Receive Filter Buffer Control n) RXFRCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) 	. 411 . 411 High . 410 Low . 410 High . 409 Low . 410 . 412 . 425 Iden- . 422 Iden- . 422 Iden- . 421 Iden-
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 422 Iden- 422 Iden- 422 Iden- 421 Iden- 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Filter n Standard Identifier, Byte) RXFBCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Standard tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 420 Iden- 422 Iden- 422 Iden- 421 Iden- 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firor Count) RXFBCONn (Receive Filter Buffer Control n) RXFRCONn (Receive Filter Control n) RXFRCONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 410 . 410 . 412 . 425 Iden- . 422 Iden- . 421 Iden- . 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnEIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firter Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask High Putc). 	. 411 . 411 High 410 Low 410 High 409 Low 410 421 Iden- 422 Iden- 421 Iden- 421 Iden- 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Filter n Standard Identifier, Byte) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 420 Iden- 422 Iden- 422 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 421 Iden- 422 Iden- 421 Iden- 421 Iden- 422 Iden- 421 Iden- 422 Iden- 421 Iden- 422 Iden- 421 Iden- 422 Iden- 421 Iden- 422 Iden- 422 Iden- 421 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 423 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 43
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter Suffer Control n) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFCONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask, High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 412 . 425 . 424 Iden- . 422 Iden- . 421 Iden- . 421 Iden- . 421 Iden- . 421
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter n Standard Identifier, Byte) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Standard tifier, Low Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask, High Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 421 Iden- 422 Iden- 422 Iden- 421 Iden- 421 Iden- 421 Iden- 422 Iden- 421 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Ide
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Filter n Standard Identifier, Byte) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended tifier Mask, High Byte) 	. 411 . 411 High . 410 Low . 410 High . 409 Low . 410 . 422 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 423 Iden- . 423 Iden- . 423
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Buffer n Standard Identifier, Byte) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Mask, High Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Mask, Low Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 425 . 424 Iden- . 422 Iden- . 422 Iden- . 421 Iden- . 423 Iden- . 433 Iden- . 433 Iden- / 433 Iden- / 433 Iden- / 433 Iden- / 433 Iden- / 433 Iden- / / / / / / / / / / / / / / / / / / /
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter n Standard Identifier, Byte) RXFBCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnEIDL (Receive Acceptance Filter n Standard tifier, Low Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask, High Byte) RXMnEIDL (Receive Acceptance Mask n Extended Identifier Mask, Low Byte) RXMnSIDH (Receive Acceptance Mask n Stan Identifier Mask Low Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 421 Iden- 422 Iden- 422 Iden- 422 Iden- 422 Iden- 423 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Iden- 433 Id
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Firter Count) RXFBCONn (Receive Filter Buffer Control n) RXFRCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended Identifier Mask, High Byte) RXMnEIDL (Receive Acceptance Mask n Extended Identifier Mask, Low Byte) RXMnSIDH (Receive Acceptance Mask n Stan Identifier Mask, High Byte) 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 410 . 421 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 421 Iden- . 422 Iden- . 423 iden- . 423 iden . 424 iden . 4244 iden . 424 iden . 424 iden . 424 iden . 424 iden . 424 iden . 42
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Filter Count) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Filter, Low Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Filter, Migh Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Mask, High Byte) RXMnSIDL (Receive Acceptance Mask n Extended Identifier Mask, Low Byte) RXMnSIDL (Receive Acceptance Mask n Extended Identifier Mask, Low Byte) RXMnSIDL (Receive Acceptance Mask n Standard RXMnSIDL (Receive Acceptance Mask n Standard RXMnSIDL (Receive Acceptance Mask n Standard 	. 411 High . 410 Low . 410 High . 409 Low . 410 . 425 . 424 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 422 Iden- . 421 Iden- . 422 Iden- . 423 Iden- . 433 Iden- . 433 Iden- . 433 Iden- / 433 Iden- / 433 Iden- / 433 Iden- / 433
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERCNT (Receive Filter Control n) RXFBCONn (Receive Filter Control n) RXFONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnSIDL (Receive Acceptance Mask n Extended Identifier Mask, High Byte) RXMnEIDL (Receive Acceptance Mask n Standard tifier Mask, Low Byte) RXMnSIDL (Receive Acceptance Mask n Standard tifier Mask, High Byte) 	. 411 . 411 High . 410 Low . 410 High . 409 Low . 410 . 422 Iden- . 422 Iden- 42
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Firor Count) RXFBCONn (Receive Filter Buffer Control n) RXFRCONn (Receive Filter Control n) RXFRCONn (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, Low Byte) RXMnEIDH (Receive Acceptance Mask n Extended tifier Mask, High Byte) RXMnEIDL (Receive Acceptance Mask n Standard tifier Mask, Low Byte) RXMnSIDL (Receive Acceptance Mask n Standard tifier Mask, Low Byte) 	. 411 . 411 High . 410 Low . 410 High . 409 Low . 410 . 412 . 425 Iden- . 422 Iden- . 421 Iden- . 421 Iden- . 421 Iden- . 421 Iden- . 421 Iden- . 423 Iden- . 423
 RXBnDLC (Receive Buffer n Data Length Code) RXBnDm (Receive Buffer n Data Field Byte m) RXBnEIDH (Receive Buffer n Extended Identifier, Byte) RXBnSIDL (Receive Buffer n Extended Identifier, Byte) RXBnSIDH (Receive Buffer n Standard Identifier, Byte) RXBnSIDL (Receive Buffer n Standard Identifier, Byte) RXERRCNT (Receive Buffer n Standard Identifier, Byte) RXFBCONn (Receive Filter Buffer Control n) RXFCONn (Receive Filter Control n) RXFCONn (Receive Filter Control n) RXFnEIDH (Receive Acceptance Filter n Extended tifier, High Byte) RXFnSIDH (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXFnSIDL (Receive Acceptance Filter n Standard tifier Filter, High Byte) RXMnSIDL (Receive Acceptance Mask n Extended tifier Filter, Low Byte) RXMnEIDL (Receive Acceptance Mask n Extended tifier Filter, Mask, High Byte) RXMnSIDH (Receive Acceptance Mask n Extended Identifier Mask, Low Byte) RXMnSIDL (Receive Acceptance Mask n Standard tifier Mask, Low Byte) RXMNSIDH (Receive Acceptance Mask n Standard tifier Mask, Low Byte) RXMNSIDL (Receive Acceptance Mask n Standard tifier Mask, Low Byte) 	. 411 . 411 High 410 Low 410 High 409 Low 410 425 424 Iden- 422 Iden- 421 Iden- 422 Iden- 422 Iden- 422 Iden- 423 Iden- 424 Iden- 423 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Iden- 424 Id