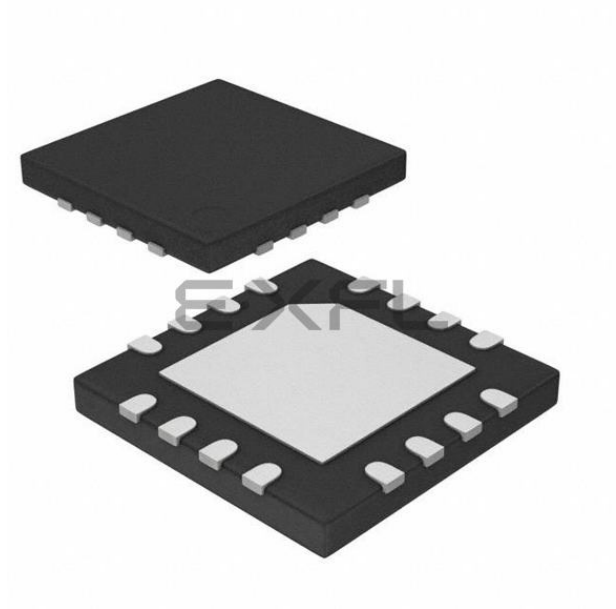


Welcome to E-XFL.COM

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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	16-UQFN Exposed Pad
Supplier Device Package	16-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18426-e-jq

  - Rising and falling edge dead-band control
  - Full-bridge, half-bridge, 1-channel drive
  - Multiple signal sources
- Capture/Compare/PWM (CCP) modules:
  - 4 CCPs
  - 16-bit resolution for Capture/Compare modes
  - 10-bit resolution for PWM mode
- Pulse-Width Modulators (PWM):
  - 2 10-bit PWMs
- Numerically Controlled Oscillator (NCO):
  - Precision linear frequency generator (@50% duty cycle) with 0.0001% step size of source input clock
  - Input Clock: 0 Hz < f_{NCO} < 32 MHz
  - Resolution: $f_{NCO}/2^{20}$
- Peripheral Pin Select (PPS):
  - I/O pin remapping of digital peripherals
- Serial Communications:
  - EUSART
    - 1 EUSART(s)
    - RS-232, RS-485, LIN compatible
    - Auto-Baud Detect, Auto-wake-up on Start.
  - Master Synchronous Serial Port (MSSP)
    - 2 MSSP(s)
    - SPI
    - I^2C, SMBus and PMBus™ compatible
- Data Signal Modulator (DSM)
  - Modulates a carrier signal with digital data to create custom carrier synchronized output waveforms
- Up to 18 I/O Pins:
  - Individually programmable pull-ups
  - Slew rate control
  - Interrupt-on-change with edge-select
  - Input level selection control (ST or TTL)
  - Digital open-drain enable
- Timer modules:
  - Timer0:
    - 8/16-bit timer/counter
    - Synchronous or asynchronous operation
    - Programmable prescaler/postscaler
    - Time base for capture/compare function
  - Timer1/3/5 with gate control:
    - 16-bit timer/counter
    - Programmable internal or external clock sources
    - Multiple gate sources

1.4.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to the following:

- EUSART
- MSSP

1.4.4 Register Legend

The table below describes the conventions for bit types and bit reset values used in the current data sheet.

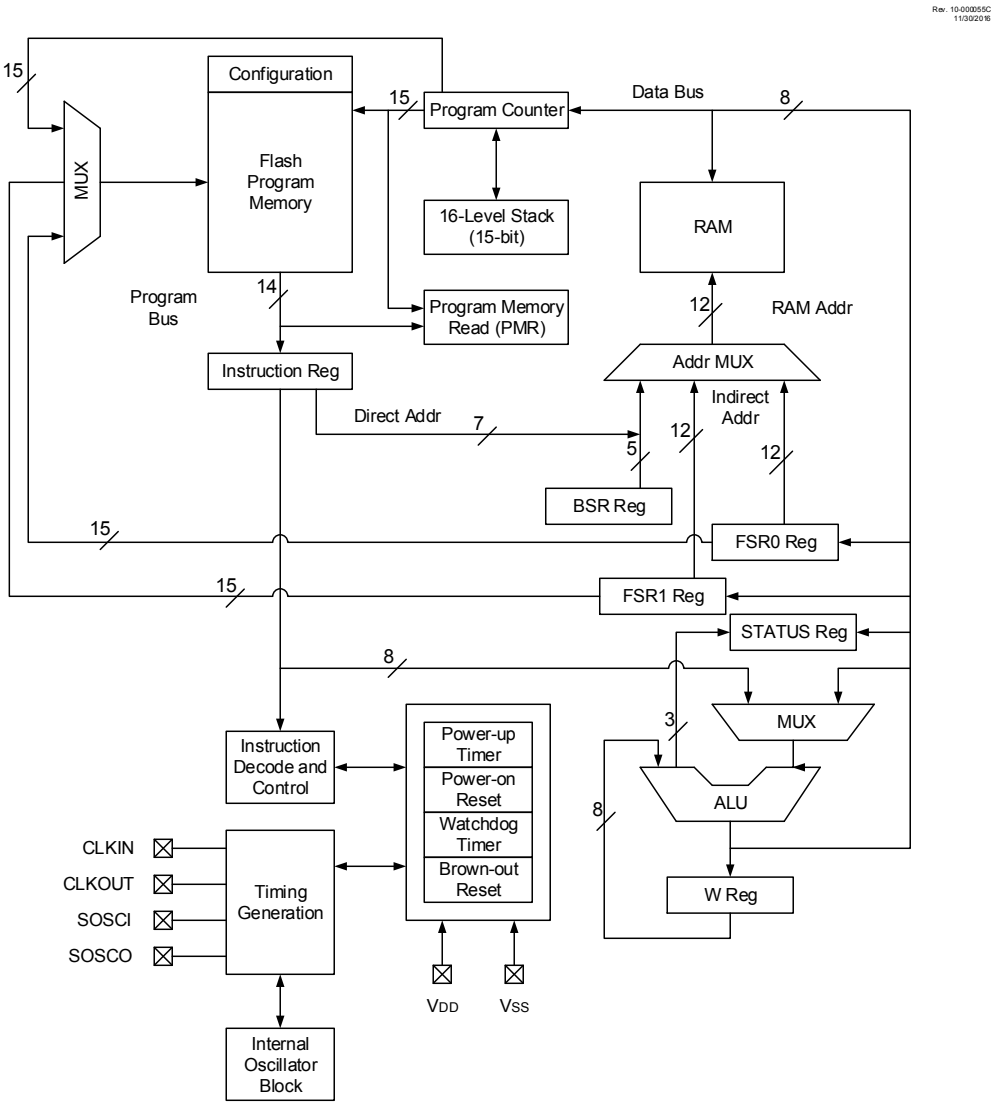
Table 1-2. Register Legend

Value	Description
RO	Read-only bit
W	Writable bit
U	Unimplemented bit, read as '0'
'1'	Bit is set
'0'	Bit is cleared
x	Bit is unknown
u	Bit is unchanged
-n/n	Value at POR and BOR/Value at all other Resets
q	Reset Value is determined by hardware
f	Reset Value is determined by fuse setting
g	Reset Value at POR for PPS re-mappable signals

3. Enhanced Mid-Range CPU

This family of devices contains an enhanced mid-range 8-bit CPU core. The CPU has 50 instructions. Interrupt capability includes automatic context saving. The hardware stack is 16-levels deep and has Overflow and Underflow Reset capability. Direct, Indirect, and Relative Addressing modes are available. The two File Select Registers (FSRs) provide the ability to read program and data memory.

Figure 3-1. Core Data Path Diagram



3.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code.

Related Links

Automatic Context Saving

7.8.8 WREG

Name: WREG
Address: 0x09 + n*0x80 [n=0..63]

Working Data Register

Bit	7	6	5	4	3	2	1	0
	WREG[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – WREG[7:0]

Related Links

Core Registers

12.8.1 WDTCON0

Name: WDTCON0
Address: 0x80C

Watchdog Timer Control Register 0

Bit	7	6	5	4	3	2	1	0
			WDTPS[4:0]					SEN
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			q	q	q	q	q	0

Bits 5:1 – WDTPS[4:0] Watchdog Timer Prescale Select bits(1)
Bit Value = Prescale Rate

Value	Description
11111 to 10011	Reserved. Results in minimum interval (1 ms)
10010	1:8388608 (2^{23}) (Interval 256s nominal)
10001	1:4194304 (2^{22}) (Interval 128s nominal)
10000	1:2097152 (2^{21}) (Interval 64s nominal)
01111	1:1048576 (2^{20}) (Interval 32s nominal)
01110	1:524288 (2^{19}) (Interval 16s nominal)
01101	1:262144 (2^{18}) (Interval 8s nominal)
01100	1:131072 (2^{17}) (Interval 4s nominal)
01011	1:65536 (Interval 2s nominal) (Reset value)
01010	1:32768 (Interval 1s nominal)
01001	1:16384 (Interval 512 ms nominal)
01000	1:8192 (Interval 256 ms nominal)
00111	1:4096 (Interval 128 ms nominal)
00110	1:2048 (Interval 64 ms nominal)
00101	1:1024 (Interval 32 ms nominal)
00100	1:512 (Interval 16 ms nominal)
00011	1:256 (Interval 8 ms nominal)
00010	1:128 (Interval 4 ms nominal)
00001	1:64 (Interval 2 ms nominal)
00000	1:32 (Interval 1 ms nominal)

Bit 0 – SEN Software Enable/Disable for Watchdog Timer bit

Value	Condition	Description
—	If WDTE = 1x	This bit is ignored
1	If WDTE = 01	WDT is turned on
0	If WDTE = 01	WDT is turned off
—	If WDTE = 00	This bit is ignored

Note:

1. Times are approximate. WDT time is based on 31 kHz LFINTOSC.

13.6.1 NVMADR

Name: NVMADR
Address: 0x81A

Nonvolatile Memory Address Register

Bit	15	14	13	12	11	10	9	8
		NVMADRH[6:0]						
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	NVMADRL[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 14:8 – NVMADRH[6:0] NVM Most Significant Address bits
Specifies the Most Significant bits for program memory address.

Bits 7:0 – NVMADRL[7:0] NVM Least Significant Address bits
Specifies the Least Significant bits for program memory address.

Note:

1. Bit <15> is undefined while WR = 1

14.6.1 PORTA

Name: PORTA
Address: 0x00C

PORTA Register

Bit	7	6	5	4	3	2	1	0
			RA5	RA4	RA3	RA2	RA1	RA0
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			x	x	x	x	x	x

Bits 0, 1, 2, 3, 4, 5 – RAn Port I/O Value bits
Reset States: POR/BOR = xxxxxx
All Other Resets = uuuuuu

Value	Description
1	Port pin is ≥ V_{IH}
0	Port pin is ≤ V_{IL}

Note:

- Writes to PORTA are actually written to the corresponding LATA register. Reads from PORTA register return actual I/O pin values.
- Bits RA0, RA1 and RA3 are read-only when $\overline{\text{DEBUG}}$ is enabled, and will read '0'.
- Bit RA3 will read '1' when MCLRE = 1 (master clear enabled) and '0' when $\overline{\text{DEBUG}}$ is enabled.

14.6.23 INLVLB

Name: INLVLB
Address: 0x1F47

Input Level Control Register

Bit	7	6	5	4	3	2	1	0
	INLVLB7	INLVLB6	INLVLB5	INLVLB4				
Access	R/W	R/W	R/W	R/W				
Reset	1	1	1	1				

Bits 4, 5, 6, 7 – INLVLBn Input Level Select on RB Pins

Value	Description
1	ST input used for port reads and interrupt-on-change
0	TTL input used for port reads and interrupt-on-change

15.9.1 Peripheral xxx Input Selection

Name: xxxPPS

Important: The Reset value of this register is determined by the device default for each peripheral.
Refer to the input selection table for a list of available ports and default pin locations.

Bit	7	6	5	4	3	2	1	0
				PORT[1:0]		PIN[2:0]		
Access				R/W	R/W	R/W	R/W	R/W
Reset				g	g	g	g	g

Bits 4:3 – PORT[1:0] Peripheral xxx Input PORT Selection bits
See the input selection table for a list of available ports and default pin locations.

Value	Description
10	PORTC
01	PORTB
00	PORTA

Bits 2:0 – PIN[2:0] Peripheral xxx Input Pin Selection bits

Value	Description
111	Peripheral input is from PORTx Pin 7 (Rx7)
110	Peripheral input is from PORTx Pin 6 (Rx6)
101	Peripheral input is from PORTx Pin 5 (Rx5)
100	Peripheral input is from PORTx Pin 4 (Rx4)
011	Peripheral input is from PORTx Pin 3 (Rx3)
010	Peripheral input is from PORTx Pin 2 (Rx2)
001	Peripheral input is from PORTx Pin 1 (Rx1)
000	Peripheral input is from PORTx Pin 0 (Rx0)

Important:
PORTB is available only for 20-pin or higher pin-count devices.

17.6.6 IOCBF

Name: IOCBF
Address: 0x1F4A

PORTB Interrupt-on-Change Flag Register

Bit	7	6	5	4	3	2	1	0
	IOCBF7	IOCBF6	IOCBF5	IOCBF4				
Access	R/W/HS	R/W/HS	R/W/HS	R/W/HS				
Reset	0	0	0	0				

Bits 4, 5, 6, 7 – IOCBFn Interrupt-on-Change Flag bits

Value	Condition	Description
1	IOCBP[n]=1	A positive edge was detected on the RB[n] pin
1	IOCBN[n]=1	A negative edge was detected on the RB[n] pin
0	IOCBP[n]=x and IOCBN[n]=x	No change was detected, or the user cleared the detected change

Note: PORTB associated registers are available on 20-pin or higher pin-count devices only.

25.6.2 T0CON1

Name: T0CON1
Address: 0x59F

Timer0 Control Register 1

Bit	7	6	5	4	3	2	1	0
	T0CS[2:0]			T0ASYNC	T0CKPS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 7:5 – T0CS[2:0] Timer0 Clock Source Select bits
Refer the clock source selection table

Bit 4 – T0ASYNC TMR0 Input Asynchronization Enable bit

Value	Description
1	The input to the TMR0 counter is not synchronized to system clocks
0	The input to the TMR0 counter is synchronized to Fosc/4

Bits 3:0 – T0CKPS[3:0] Prescaler Rate Select bit

Value	Description
1111	1:32768
1110	1:16384
1101	1:8192
1100	1:4096
1011	1:2048
1010	1:1024
1001	1:512
1000	1:256
0111	1:128
0110	1:64
0101	1:32
0100	1:16
0011	1:8
0010	1:4
0001	1:2
0000	1:1

27. Timer2 Module

The Timer2 module is a 8-bit timer that incorporate the following features:

- 8-bit Timer and Period registers
- Readable and writable
- Software programmable prescaler (1:1 to 1:128)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on T2TMR match with T2PR
- One-shot operation
- Full asynchronous operation
- Includes Hardware Limit Timer (HLT)
- Alternate clock sources
- External Timer Reset signal sources
- Configurable Timer Reset operation

See Figure 27-1 for a block diagram of Timer2. See table below for the clock source selections.

Important: References to module Timer2 apply to all the even numbered timers on this device. (Timer2, Timer4, etc.)

Mode	MODE<4:0>		Output Operation	Operation	Timer Control		
	<4:3>	<2:0>			Start	Reset	Stop
		100	Edge Triggered Start and Hardware Reset (Note 1)	Rising edge start and Rising edge Reset (Figure 27-9)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑	
		101		Falling edge start and Falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers ↓	
		110		Rising edge start and Low level Reset (Figure 27-10)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0	
		111		Falling edge start and High level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1	
Mono-stable	10	000	Reserved				
		001	Edge Triggered Start (Note 1)	Rising edge start (Figure 27-11)	ON = 1 and TMRx_ers ↑	—	ON = 0 or Next clock after TMRx = PRx (Note 3)
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
Reserved		100	Reserved				
Reserved		101	Reserved				
One-shot		110	Level Triggered Start and Hardware Reset	High level start and Low level Reset (Figure 27-12)	ON = 1 and TMRx_ers = 1	TMRx_ers = 0	ON = 0 or Held in Reset (Note 2)
		111		Low level start & High level Reset	ON = 1 and TMRx_ers = 0	TMRx_ers = 1	

- Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
- Enable Timer2 by setting the T2ON bit of the T2CON register.

6. Enable PWM output pin and wait until Timer2 overflows, TMR2IF bit of the PIRx register is set.(2)
7. Enable the PWMx pin output driver(s) by clearing the associated TRIS bit(s) and setting the desired pin PPS control bits.
8. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note:

1. In order to send a complete duty cycle and period on the first PWM output, the above steps must be followed in the order given. If it is not critical to start with a complete PWM signal, then move Step 8 to replace Step 4.
2. For operation with other peripherals only, disable PWMx pin outputs.

30.9.1 PWMx Pin Configuration

All PWM outputs are multiplexed with the PORT data latch. The user must configure the pins as outputs by clearing the associated TRIS bits.

30.10 Setup for PWM Operation to Other Device Peripherals

The following steps should be taken when configuring the module for PWM operation to be used by other device peripherals:

1. Disable the PWMx pin output driver(s) by setting the associated TRIS bit(s).
2. Clear the PWMxCON register.
3. Load the T2PR register with the PWM period value.
4. Load the PWMxDCH register and bits <7:6> of the PWMxDCL register with the PWM duty cycle value.
5. Configure and start Timer2:
   - Clear the TMR2IF interrupt flag bit of the PIRx register.(1)
   - Select the timer clock source to be as $F_{OSC}/4$ using the TxCLKCON register. This is required for correct operation of the PWM module.
   - Configure the T2CKPS bits of the T2CON register with the Timer2 prescale value.
   - Enable Timer2 by setting the T2ON bit of the T2CON register.
6. Wait until Timer2 overflows, TMR2IF bit of the PIRx register is set.(1)
7. Configure the PWM module by loading the PWMxCON register with the appropriate values.

Note:

1. In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then step 6 may be ignored.

MDCARL	
CLS<3:0>	Connection
0001	F_{OSC} (system clock)
0000	Pin selected by MDCARLPPS

32.4 Carrier Synchronization

During the time when the DSM switches between carrier high and carrier low signal sources, the carrier data in the modulated output signal can become truncated. To prevent this, the carrier signal can be synchronized to the modulator signal. When synchronization is enabled, the carrier pulse that is being mixed at the time of the transition is allowed to transition low before the DSM switches over to the next carrier source.

Synchronization is enabled separately for the carrier high and carrier low signal sources. Synchronization for the carrier high signal is enabled by setting the CHSYNC bit. Synchronization for the carrier low signal is enabled by setting the CLSYNC bit.

The figures below show the timing diagrams of using various synchronization methods.

Figure 32-2. On Off Keying (OOK) Synchronization

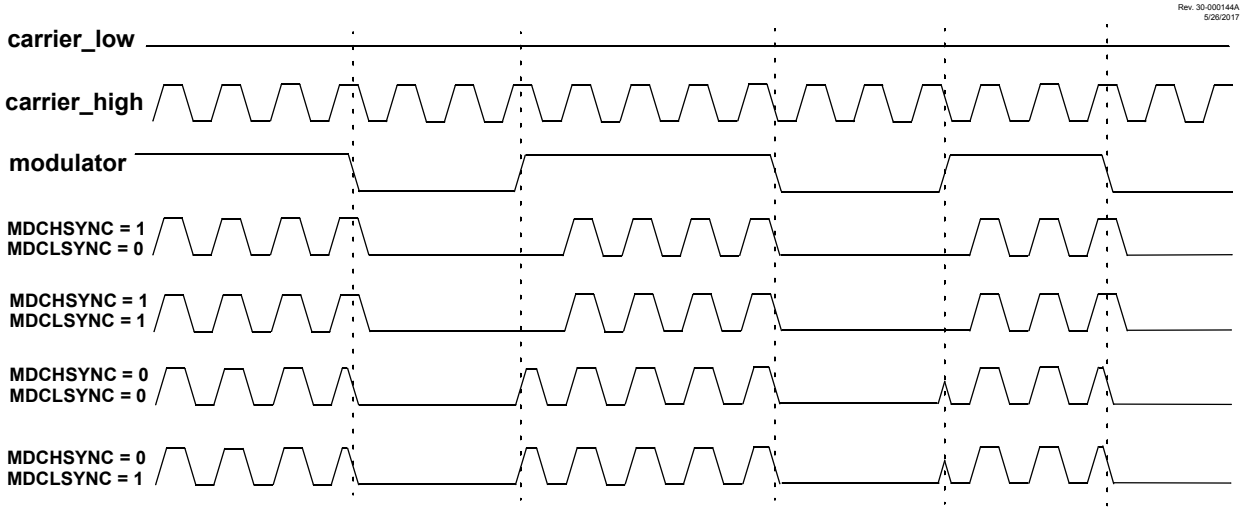
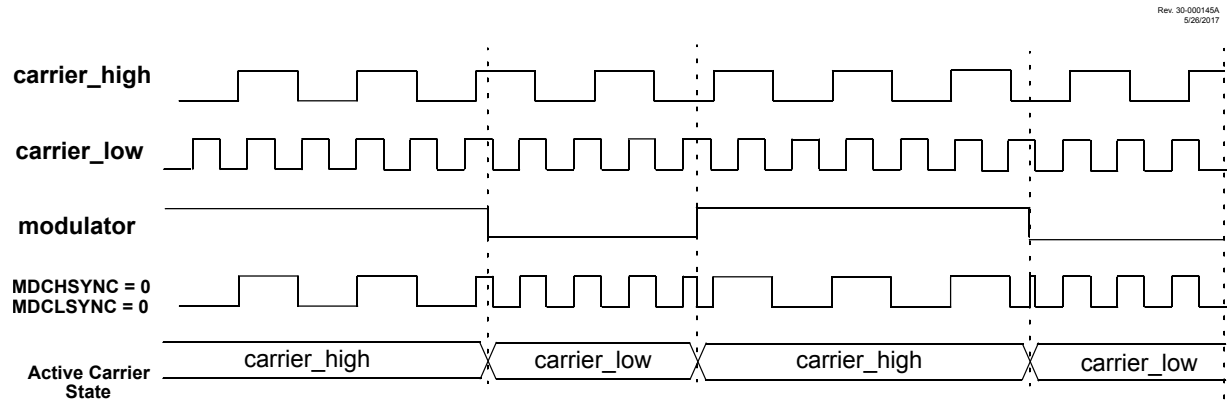


Figure 32-3. No Synchronization (MDCHSYNC = 0, MDCLSYNC = 0)



35.4.2 Arbitration

Each master device must monitor the bus for Start and Stop bits. If the device detects that the bus is busy, it cannot begin a new message until the bus returns to an Idle state.

However, two master devices may try to initiate a transmission on or about the same time. When this occurs, the process of arbitration begins. Each transmitter checks the level of the SDA data line and compares it to the level that it expects to find. The first transmitter to observe that the two levels do not match, loses arbitration, and must stop transmitting on the SDA line.

For example, if one transmitter holds the SDA line to a logical one (lets it float) and a second transmitter holds it to a logical zero (pulls it low), the result is that the SDA line will be low. The first transmitter then observes that the level of the line is different than expected and concludes that another transmitter is communicating.

The first transmitter to notice this difference is the one that loses arbitration and must stop driving the SDA line. If this transmitter is also a master device, it also must stop driving the SCL line. It then can monitor the lines for a Stop condition before trying to reissue its transmission. In the meantime, the other device that has not noticed any difference between the expected and actual levels on the SDA line continues with its original transmission. It can do so without any complications, because so far, the transmission appears exactly as expected with no other transmitter disturbing the message.

Slave Transmit mode can also be arbitrated, when a master addresses multiple slaves, but this is less common.

If two master devices are sending a message to two different slave devices at the address stage, the master sending the lower slave address always wins arbitration. When two master devices send messages to the same slave address, and addresses can sometimes refer to multiple slaves, the arbitration process must continue into the data stage.

Arbitration usually occurs very rarely, but it is a necessary process for proper multi-master support.

35.4.3 Byte Format

All communication in I^2C is done in 9-bit segments. A byte is sent from a master to a slave or vice-versa, followed by an Acknowledge bit sent back. After the eighth falling edge of the SCL line, the device outputting data on the SDA changes that pin to an input and reads in an acknowledge value on the next clock pulse.

The clock signal, SCL, is provided by the master. Data is valid to change while the SCL signal is low, and sampled on the rising edge of the clock. Changes on the SDA line while the SCL line is high define special conditions on the bus, explained below.

35.4.4 Definition of I^2C Terminology

There is language and terminology in the description of I^2C communication that have definitions specific to I^2C. That word usage is defined below and may be used in the rest of this document without explanation. This table was adapted from the Philips I^2C specification.

TERM	Description
Transmitter	The device which shifts data out onto the bus.
Receiver	The device which shifts data in from the bus.
Master	The device that initiates a transfer, generates clock signals and terminates a transfer.

- TXEN = 1 (enables the transmitter circuitry of the EUSART)
- SYNC = 0 (configures the EUSART for asynchronous operation)
- SPEN = 1 (enables the EUSART and automatically enables the output drivers for the RxyPPS selected as the TXx/CKx output)

All other EUSART control bits are assumed to be in their default state.

If the TXx/CKx pin is shared with an analog peripheral, the analog I/O function must be disabled by clearing the corresponding ANSEL bit.

Important: The TXxIF Transmitter Interrupt flag is set when the TXEN enable bit is set and the TSR is idle.

36.1.1.2 Transmitting Data

A transmission is initiated by writing a character to the TXxREG register. If this is the first character, or the previous character has been completely flushed from the TSR, the data in the TXxREG is immediately transferred to the TSR register. If the TSR still contains all or part of a previous character, the new character data is held in the TXxREG until the Stop bit of the previous character has been transmitted. The pending character in the TXxREG is then transferred to the TSR in one T_{CY} immediately following the Stop bit transmission. The transmission of the Start bit, data bits and Stop bit sequence commences immediately following the transfer of the data to the TSR from the TXxREG.

36.1.1.3 Transmit Data Polarity

The polarity of the transmit data can be controlled with the SCKP bit of the BAUDxCON register. The default state of this bit is '0' which selects high true transmit idle and data bits. Setting the SCKP bit to '1' will invert the transmit data resulting in low true idle and data bits. The SCKP bit controls transmit data polarity in Asynchronous mode only. In Synchronous mode, the SCKP bit has a different function. See the Clock Polarity section for more detail.

36.1.1.4 Transmit Interrupt Flag

The TXxIF interrupt flag bit of the PIRx register is set whenever the EUSART transmitter is enabled and no character is being held for transmission in the TXxREG. In other words, the TXxIF bit is only clear when the TSR is busy with a character and a new character has been queued for transmission in the TXxREG. The TXxIF flag bit is not cleared immediately upon writing TXxREG. TXxIF becomes valid in the second instruction cycle following the write execution. Polling TXxIF immediately following the TXxREG write will return invalid results. The TXxIF bit is read-only, it cannot be set or cleared by software.

The TXxIF interrupt can be enabled by setting the TXxIE interrupt enable bit of the PIEx register. However, the TXxIF flag bit will be set whenever the TXxREG is empty, regardless of the state of TXxIE enable bit.

To use interrupts when transmitting data, set the TXxIE bit only when there is more data to send. Clear the TXxIE interrupt enable bit upon writing the last character of the transmission to the TXxREG.

36.1.1.5 TSR Status

The TRMT bit of the TXxSTA register indicates the status of the TSR register. This is a read-only bit. The TRMT bit is set when the TSR register is empty and is cleared when a character is transferred to the TSR register from the TXxREG. The TRMT bit remains clear until all bits have been shifted out of the TSR register. No interrupt logic is tied to this bit, so the user needs to poll this bit to determine the TSR status.

Bit 1 – TRMT Transmit Shift Register (TSR) Status bit

Value	Description
1	TSR is empty
0	TSR is not empty

Bit 0 – TX9D Ninth bit of Transmit Data
Can be address/data bit or a parity bit.

Note:

1. SREN and CREN bits override TXEN in Sync mode.

Figure 37-5. Period and Duty-Cycle, Repeat Acquisition Mode Timing Diagram

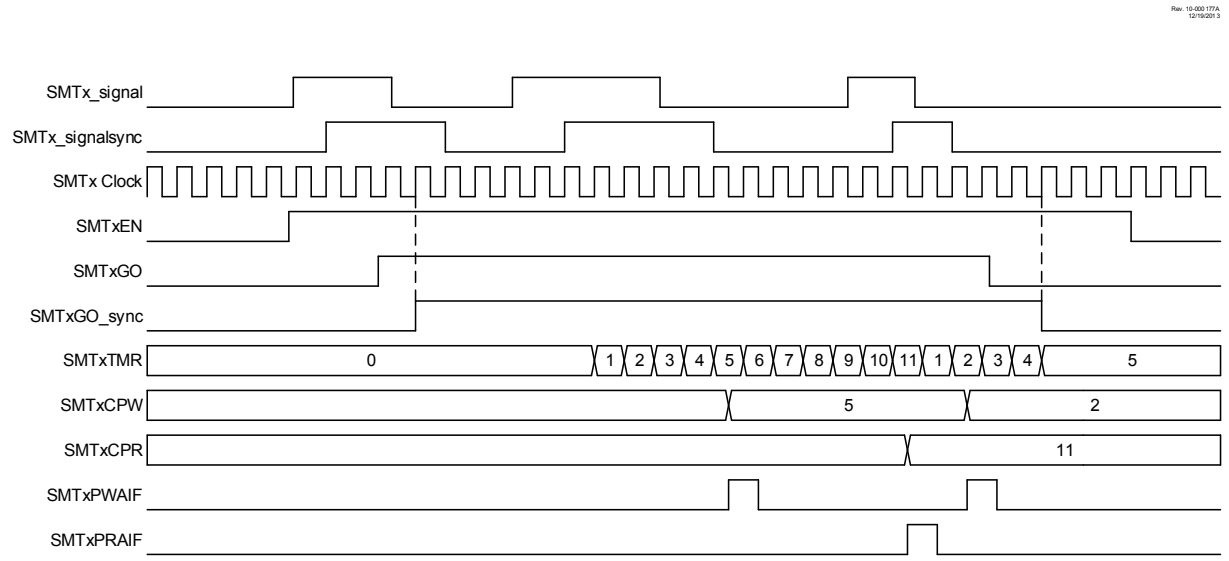
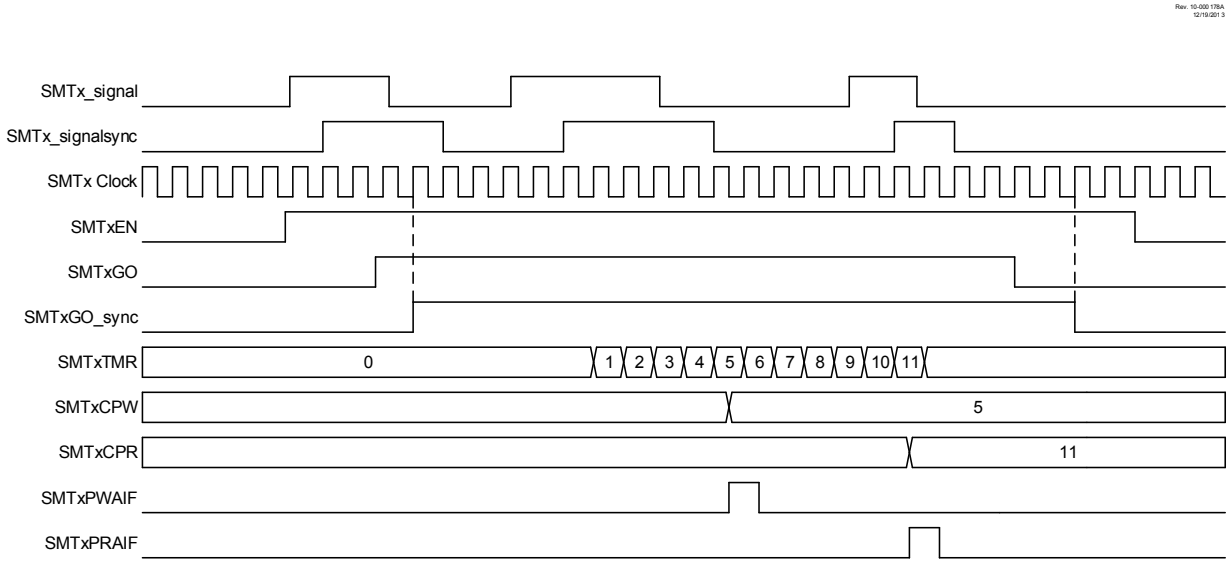


Figure 37-6. Period and Duty-Cycle, Single Acquisition Mode Timing Diagram



37.1.6.4 High and Low Measurement Mode

This mode measures the high and low pulse time of the signal relative to the SMT clock. It begins incrementing the SMTxTMR on a rising edge on the input signal, then updates the SMTxCPW register with the value and resets the SMTxTMR on a falling edge, starting to increment again. Upon observing another rising edge, it updates the SMTxCPR register with its current value and once again resets the SMTxTMR value and begins incrementing again. See the figures below.

Offset	Name	Bit Pos.								
		15:8	LTHH[7:0]							
0x8E	ADUTH	7:0	UTHL[7:0]							
		15:8	UTHH[7:0]							
0x90	ADERR	7:0	ADERRL[7:0]							
		15:8	ERRH[7:0]							
0x92	ADSTPT	7:0	STPTL[7:0]							
		15:8	STPTH[7:0]							
0x94	ADFLTR	7:0	FLTRL[7:0]							
		15:8	FLTRH[7:0]							
0x96	ADACC	7:0	ACCL[7:0]							
		15:8	ACCH[7:0]							
		23:16							ACCU[1:0]	
0x99	ADCNT	7:0	CNT[7:0]							
0x9A	ADRPT	7:0	RPT[7:0]							
0x9B	ADPREV	7:0	PREVL[7:0]							
		15:8	PREVH[7:0]							
0x9D	ADRES	7:0	RESL[7:0]							
		15:8	RESH[7:0]							
0x9F	ADPCH	7:0			PCH[5:0]					
0xA0 ... 0xFF	Reserved									
0x0100	INDF0	7:0	INDF0[7:0]							
0x0101	INDF1	7:0	INDF1[7:0]							
0x0102	PCL	7:0	PCL[7:0]							
0x0103	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x0104	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0106	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x0108	BSR	7:0			BSR[5:0]					
0x0109	WREG	7:0	WREG[7:0]							
0x010A	PCLATH	7:0		PCLATH[6:0]						
0x010B	INTCON	7:0	GIE	PEIE						INTEDG
0x010C	ADACQ	7:0	ACQL[7:0]							
		15:8				ACQH[4:0]				
0x010E	ADCAP	7:0				CAP[4:0]				
0x010F	ADPRE	7:0	PREL[7:0]							
		15:8				PREH[4:0]				
0x0111	ADCON0	7:0	ON	CONT		CS		FRM		GO
0x0112	ADCON1	7:0	PPOL	IPEN	GPOL					DSEN
0x0113	ADCON2	7:0	PSIS	CRS[2:0]			ACLR	MD[2:0]		
0x0114	ADCON3	7:0		CALC[2:0]			SOI	MD[2:0]		
0x0115	ADSTAT	7:0	OV	UTHR	LTHR	MATH		STAT[2:0]		
0x0116	ADREF	7:0				NREF			PREF[1:0]	
0x0117	ADACT	7:0				ACT[4:0]				