TB3264

Using the Universal Timer (UTMR) Module

Introduction

Author: Max Prasad, Microchip Technology Inc.

The Universal Timer (UTMR) module is the new timer module introduced in newer PIC® devices, and combines most of the operations of all the legacy timers (TMR0/1/2, SMT, CCP) into one single timer. The inbuilt capture and compare features, along with customizable Start, Reset and Stop options, make the UTMR universal. Multiple UTMR modules can be chained together to form a larger sized timer as well. Figure 1 shows the block diagram of the module.

Figure 1. UTMR Module Block Diagram

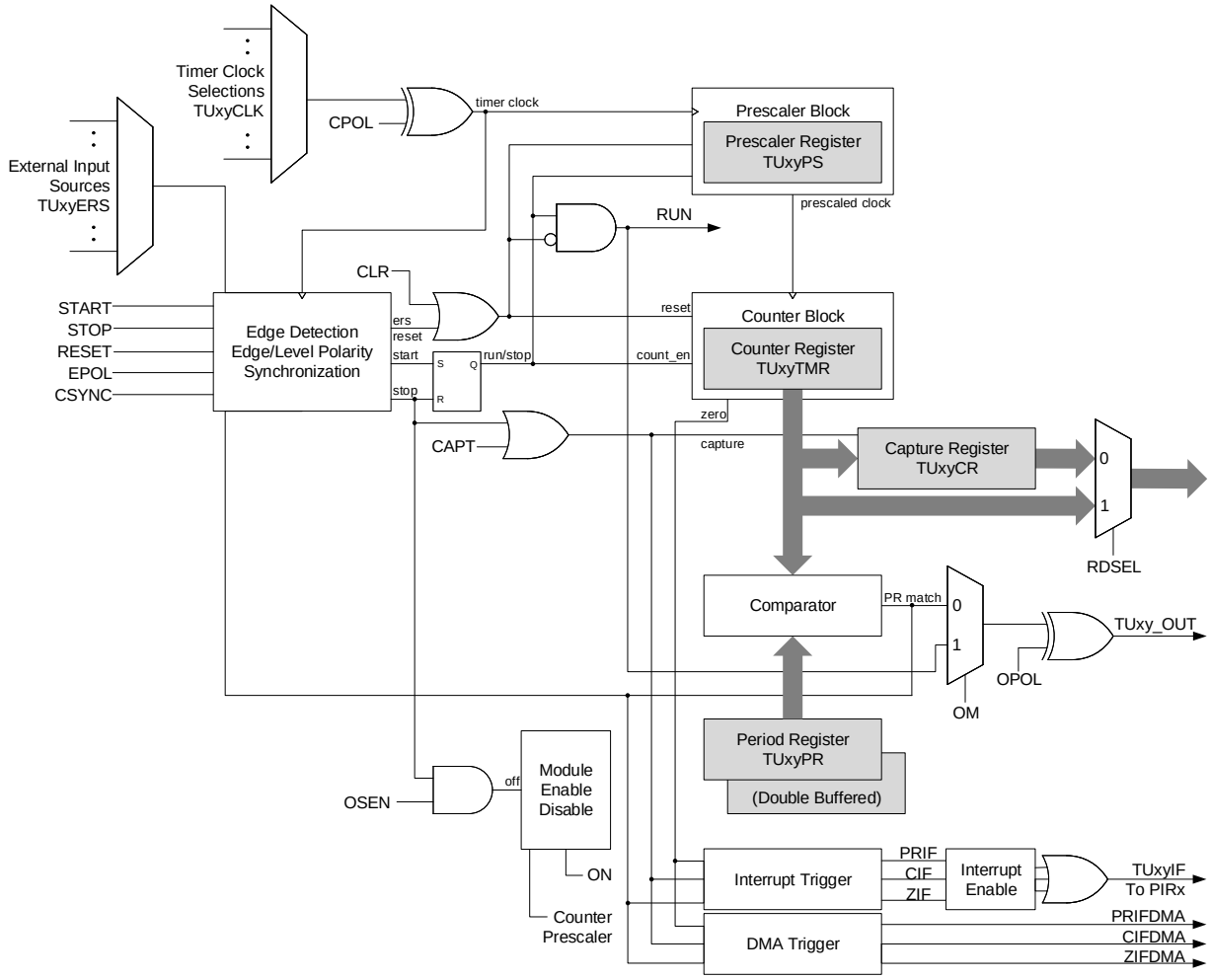


Table of Contents

1. Basic Operation

The operation of the UTMR revolves around the three n-bit counter, capture and period registers, and the Start, Reset and Stop conditions for hardware limit timing. While the basic operation of the timer is explained in this section, refer to the device data sheet for full description and operation of the UTMR module.

1.1 Timer, Capture and Period Registers

The n-bit timer/counter register (TUxyTMR) is the actual register that contains the timer value. This value increments at every timer clock tick. Use the CPOL bit to select the active timer clock edge. The TUxyTMR timer/counter register is not double buffered in hardware and no read/write protection is offered in hardware. Hence, the user is encouraged to access or alter the timer value only using the capture and period registers described below. The raw counter may be directly accessed, but the access is not double buffered and may corrupt the data.

The n-bit period register (TUxyPR) contains the value that determines the period of the timer. At every timer clock tick, the value in the TUxyTMR timer counter register is compared to the TUxyPR period register. When the TUxyTMR timer counter value increments from PR-1 to PR, a period register match (PR match) condition is generated. The TUxyPR period register is double buffered in hardware to allow the user to change the period of the timer while the timer is running. To change the period while the timer is running, write to the higher bytes of the TUxyPR register first (does not apply in 8-bit configuration of the module), which buffers the new data. Then write to the least significant byte of the TUxyPR register to arm the buffer to be loaded. Once the buffer is armed, an upcoming Reset event will load the new period value.

The n-bit capture register (TUxyCR) is a read-only register that is provided to safely and atomically read the value of the timer counter. Any of the following capture events will capture the timer counter value in that instant (clock synchronization delay applies) and store it in the TUxyCR capture register.

1. Setting the CAPT command bit.
2. When STOP ≠ None, the timer value is captured whenever a stop event is generated.
3. When STOP = None, the timer value is captured at every rising edge of the ERS signal (or falling ERS edge if EPOL = 1). Refer to Table 1-3 for more information.

Both TUxyTMR timer counter and the TUxyCR capture registers share the same memory address. When RDSEL = 1, the TUxyTMR timer counter register is accessed to read, and when RDSEL = 0, the TUxyCR capture register is accessed to read. The TUxyCR capture register gets updated whenever a capture event occurs regardless of the RDSEL bit setting.

1.2 Start, Reset and Stop Hardware Limit Timing

The UTMR module consists of an External Reset Signal (ERS) source that can be used to Start, Reset and/or Stop the timer without any software intervention. The different Start, Reset and Stop conditions are listed in the tables below. The EPOL bit controls the polarity of the ERS signal and changes how the Start, Reset, and Stop conditions are handled in the hardware.

Table 1-1. Start Conditions

Start Condition	Operation when EPOL = 0	Operation when EPOL = 1
None	No hardware control – timer runs when ON = 1	Same as EPOL = 0
Either ERS Edge	Timer starts at either edge (rising or falling) of ERS	Same as EPOL = 0
Rising ERS Edge	Timer starts at rising edge of ERS	Timer starts at falling edge of ERS

.....continued		
Start Condition	**Operation when EPOL = 0**	**Operation when EPOL = 1**
ERS Level-1	Timer starts at the first clock where ERS has a high level and the Start condition remains asserted as long as the subsequent clocks register a high ERS level.	Timer starts at the first clock where ERS has a low level and the Start condition remains asserted as long as the subsequent clocks register a low ERS level.

Table 1-2. Reset Conditions

Reset Condition	Operation when EPOL = 0	Operation when EPOL = 1
None	No hardware reset	Same as EPOL = 0
ERS Level-0 + PR Match	Timer resets at the first clock where ERS has a low level and the Reset condition remains asserted as long as the subsequent clocks register a low ERS level. Level output and RUN bit become false while the Reset condition is active. A PR match condition also triggers a Reset, in which case the timer resets at the next clock after PR match. Level output and RUN bit remain true in this case.	Timer resets at the first clock where ERS has a high level and the Reset condition remains asserted as long as the subsequent clocks register a high ERS level. Level output and RUN bit become false while the Reset condition is active. A PR match condition also triggers a Reset, in which case the timer resets at the next clock after PR match. Level output and RUN bit remain true in this case.
At Start + PR Match	Timer resets at the first clock whenever a Start condition is generated. A PR match condition also triggers a Reset, in which case the timer resets at the next clock after PR match. Level output and RUN bit remain true in either case.	Same as EPOL = 0
At PR Match	Timer resets at the next clock after a PR match condition is generated. Level output and RUN bit remain true in this case.	Same as EPOL = 0

Table 1-3. Stop Conditions

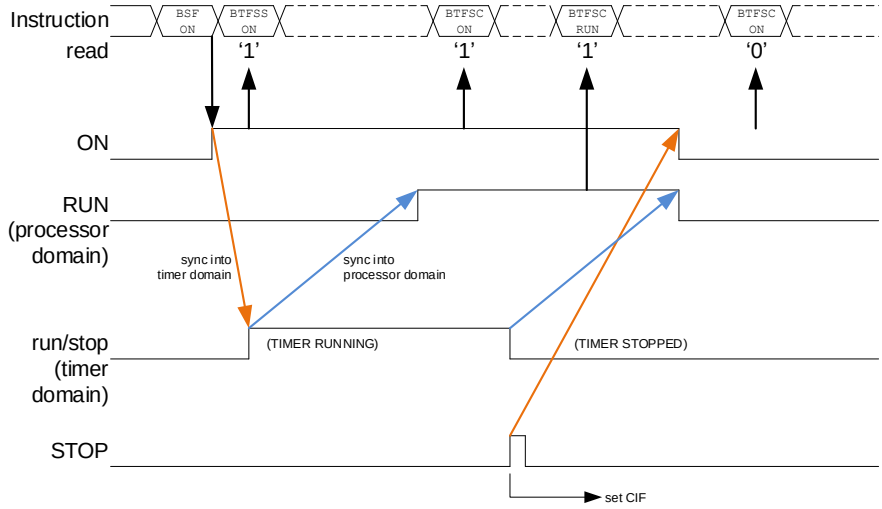
Reset Condition	Operation when EPOL = 0	Operation when EPOL = 1
None	No hardware control – timer stops when ON = 0. Capture occurs at every rising ERS edge.	No hardware control – timer stops when ON = 0. Capture occurs at every falling ERS edge.
Either ERS Edge	Timer stops at either edge (rising or falling) of ERS and causes a capture event	Same as EPOL = 0
Rising ERS Edge	Timer stops at rising edge of ERS and causes a capture event.	Timer stops at falling edge of ERS and causes a capture event
At PR Match	Timer stops at the next clock after a PR match condition occurs and causes a capture event.	Same as EPOL = 0

1.3 Synchronous vs. Asynchronous Operation

The UTMR module is designed to operate completely isolated from the main system which simplifies the timing, especially when running off of an external clock. The timer being isolated means that any input or output signal or command needs to be synchronized to the timer domain. This synchronization is controlled by the CSYNC bit.

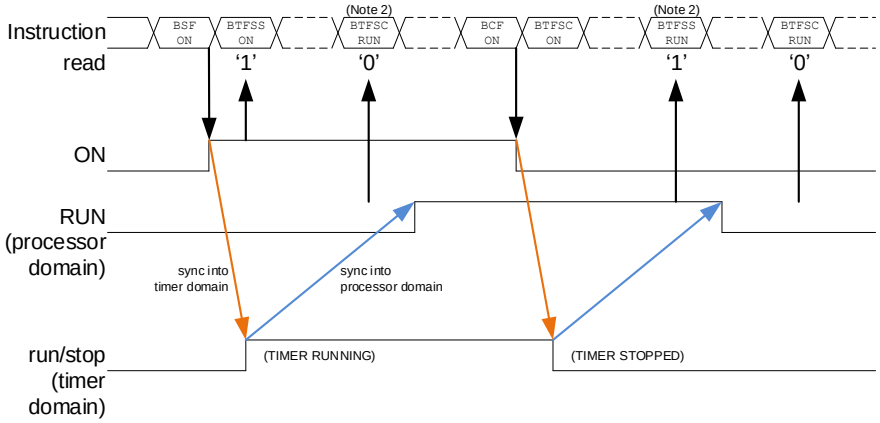
When the CSYNC bit is set, any input signal or command that goes from the system domain into the timer domain is synchronized to the timer clock, and any output signal that goes from the timer domain to the system domain is synchronized to the system clock. There will be a synchronization delay of up to three cycles of the clock that the signal is being synchronized to, i.e., an input signal or command will be delayed by three *timer* clocks, whereas an output signal will be delayed by three *system* clocks. This synchronization ensures that any ERS edges or SFR bit changes are not missed by the timer. Because the delay is the same for any Start, Reset or Stop condition, the counter value of the timer will remain unaffected. See Figure 1-1 and Figure 1-2 for examples of clock synchronization.

Figure 1-1. Clock Synchronization with ON bit and Stop Condition



Note:
1. Not to scale; clocks are not shown.

Figure 1-2. Clock Synchronization with ON bit and Off Condition



Note:
1. Not to scale; clocks are not shown.
2. It is possible to read RUN before clock synchronization is complete.

When the CSYNC bit is cleared, all input signals and commands are fed directly into the timer module, and all output signals go to the system domain immediately. There is no synchronization delay because the signals are registered immediately. If the edge of the input signal or command occurs too close to the active clock edge, a clock collision may occur and there is a chance that the input signal or command will be missed. To avoid this, consider setting CSYNC = 1 to synchronize the signals. Use of CSYNC = 0 is only recommended when the clock is non-continuous (like push button, or a counter) and it is expected that the clock and ERS edges won't collide.

1.4 One-Shot and Limit Modes

The UTMR module contains two different modes that can be applied on top of the selected Start, Reset, and Stop conditions – One-Shot and Limit modes.

The One-Shot mode is selected by setting the OSEN bit. When One-Shot mode is active, a Stop condition will stop a running timer and clear the ON bit, even if it coincides with a Start condition. This disables the timer module and prevents it from being restarted again (unless the user turns the timer on again). When ON = 1 and the timer is not running (RUN = 0), if a stop event occurs prior to the start, the ON bit is not cleared.

The Limit mode is selected by setting the LIMIT bit, and is relevant only when RESET = None and counter equals PR value. Limit mode prevents the counter value in TUxyTMR timer register from exceeding the period value in TUxyPR period register. In the clocks following the PR match condition, while the timer remains in a Run state (RUN = 1), the counter value is held at PR and does not advance to PR+1. Any subsequent Start conditions will not have any effect on the counter value, unless the counter is reset.

Both the One-Shot and Limit modes have effects exclusive of one another and can be used together in an application.

1.5 Output Modes

The UTMR module can have two different types of outputs, level and pulse, and are selectable using the OM bit.

When level output is selected (OM = 1), the output is high as long as the timer is in a run state and goes low when the timer stops. The output remains high through all Reset conditions, except when Reset = ERS Level-0. This run or stop state is also reflected in the RUN status bit. When Start = Rising/Either ERS Edge, the level output is asserted as soon as the qualified ERS edge is registered without any synchronization delays (even when CSYNC = 1).

When pulse output is selected (OM = 0), the output is pulsed high for one timer clock cycle whenever a PR match occurs. If prescaler is enabled, then the output is pulsed high during the final timer clock cycle of the PR match duration, thus ensuring that the output pulse width is the same regardless of prescaler setting.

The polarity of the output is inverted when the OPOL bit is set. The OPOL bit controls the output polarity even when the module is disabled (ON = 0). The OPOL bit does not affect the polarity of the RUN status bit.

1.6 Interrupts and DMA Triggers

The UTMR module offers three interrupts and DMA triggers:

1. Period Match Interrupt – occurs when a PR match condition is generated.(1)
2. Zero Interrupt – occurs when timer counter value becomes zero either because of a Reset condition, rollover, or software clear.
3. Capture Interrupt – occurs when a capture event is generated.(2)

Each of these interrupts has separate interrupt enable and flag bits in the timer module (PRIE/PRIF, ZIE/ZIF, CIE/CIF), which are combined together to have one top system level interrupt (TUxyIE/TUxyIF) in the PIRx register.

Important:

1. A PR match condition is generated only when the timer counter increments from PR-1 to PR. The condition is not triggered when the user writes the PR value directly to the counter.
2. Refer to Timer, Capture and Period Registers for details about the different capture events.

1.7 Chained Operation

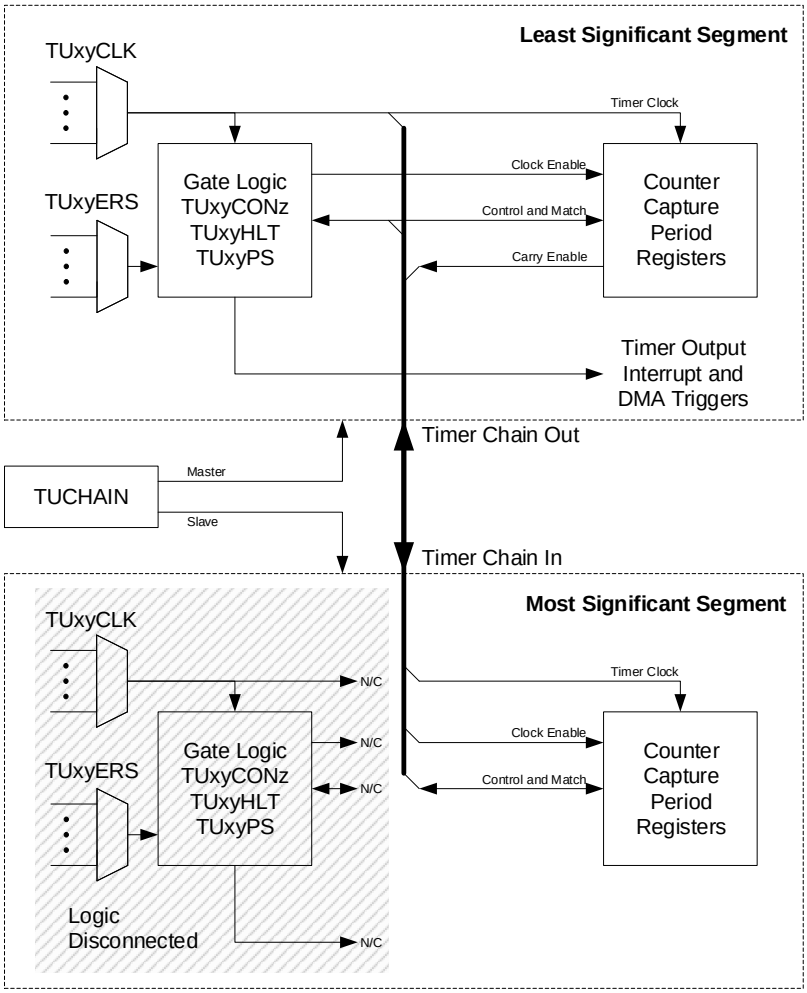
A special feature of the UTMR module is the ability to chain two timers together to operate as one big timer. This is done primarily to increase the size of the timer. An n-bit UTMR module can be chained with an m-bit UTMR module to operate as a single (n+m)-bit UTMR.

For example, two 8-bit versions of the timer can be chained together to form a 16-bit timer. Or, two 16-bit timers can be chained together to form a 32-bit timer. These timer combinations are specific to each device. Please refer to the device data sheet for specific chaining combinations available. These combinations can be enabled using appropriate bits in the TUCHAIN register.

When two timers are chained, one of them is designated as a Master whereas the other is designated as a Slave. The Master forms the LSBs of the chained timer, whereas the Slave forms the MSBs. These Master/Slave combinations are also device specific. Please refer to the device data sheet for specific details.

When chained, the control registers of the Slave timer become defunct. The timer, period, and capture registers of the Slave timer are added as MSBs to the timer, period, and capture registers of the Master timer. The control registers of the Master timer control the operation of the entire chained timer. See Figure 1-3 below for more information.

Figure 1-3. UTMR Chaining



Note:

1. This is a conceptual diagram only.
2. Control registers, state machine, prescaler and input ERS and clock for slave is not used. Rather they are derived from the master segment.

2. Use Cases

The UTMR can operate in many applications typically falling into the following categories:

- Periodic Operation
- External Clock Gating
- Hardware Limit Timing
- External Clock Operation

2.1 Periodic Operation

The timer can operate periodically by removing the Stop condition, allowing it to run freely. The following use cases demonstrates this:

- Periodic Rollover
- Software Triggered Time Delay
- Hardware Triggered Time Delay
- Variable Time Delay

2.1.1 Periodic Rollover

The timer can be configured for continuous periodic operation as per the settings shown in Table 2-1.

Table 2-1. Timer Configuration

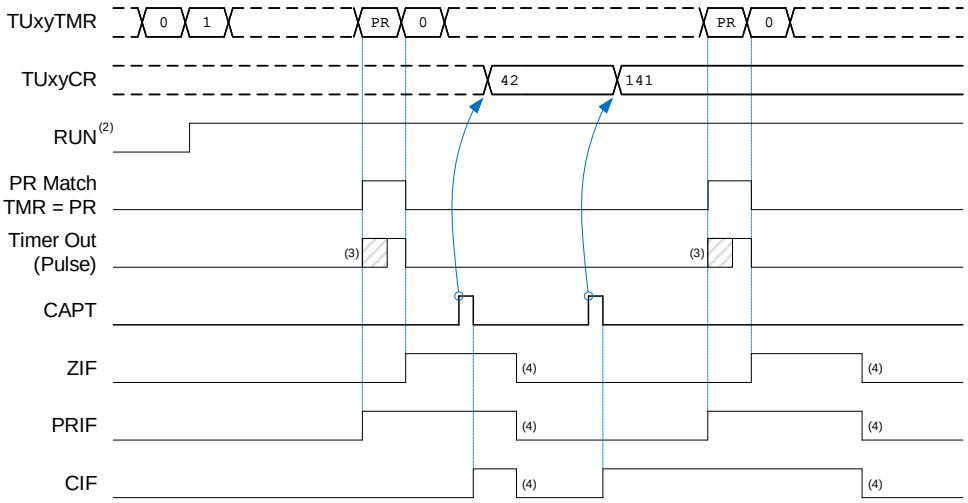
Timer Setting	Value
START	None (ON = 1)
RESET	At PR Match
STOP	None
CSYNC (Clock Sync)	Sync
PR (Period Register)	Desired Period Value - 1 (e.g., for a desired period of 20, PR = 19)

The counter counts until a PR match occurs, and then rolls over to zero and continues counting. At PR match, the PRIF interrupt and output pulse occur. At the following clock, the counter returns to zero and the ZIF interrupt occurs. This is shown in Figure 2-1.

If RESET = None, the counter will rollover at the maximum value of the timer (0xF...FF) with ZIF, but the output pulse and PRIF will occur at PR match. This is shown in Figure 2-2.

The software can capture (read) the counter value using CAPT command at any time.

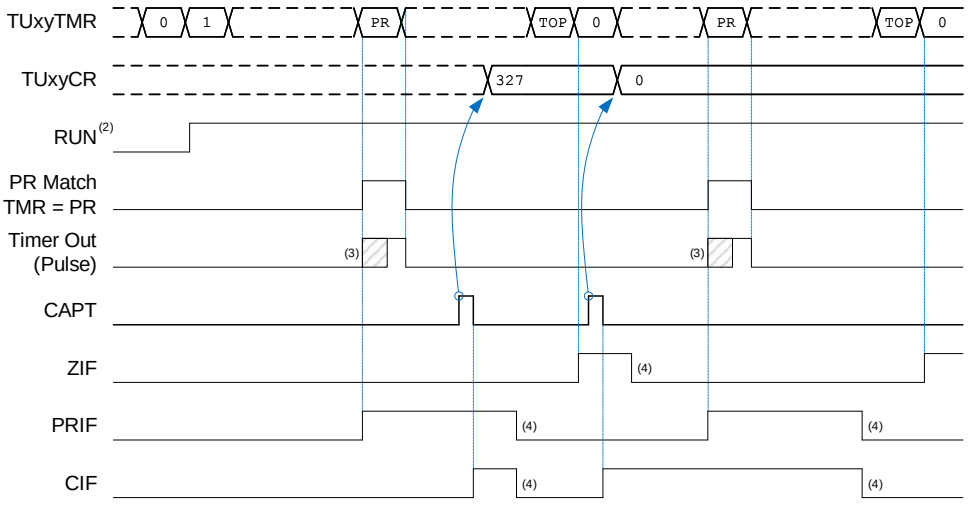
Figure 2-1. Periodic Rollover With Reset



Timer Setup:
START = None (ON = 1) RESET = At PR Match STOP = None
CSYNC = Sync

Note:
1. Cross-domain clock synchronization applies as required but is not highlighted.
2. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
3. The uncertainty of the output is due to the prescaler setting.
4. Cleared by software.

Figure 2-2. Periodic Rollover Without Reset



Timer Setup:
START = None (ON = 1) RESET = None STOP = None
CSYNC = Sync

Note:
1. Cross-domain clock synchronization applies as required but is not highlighted.
2. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
3. The uncertainty of the output is due to the prescaler setting.
4. Cleared by software.

2.1.2 Software Triggered Time Delay

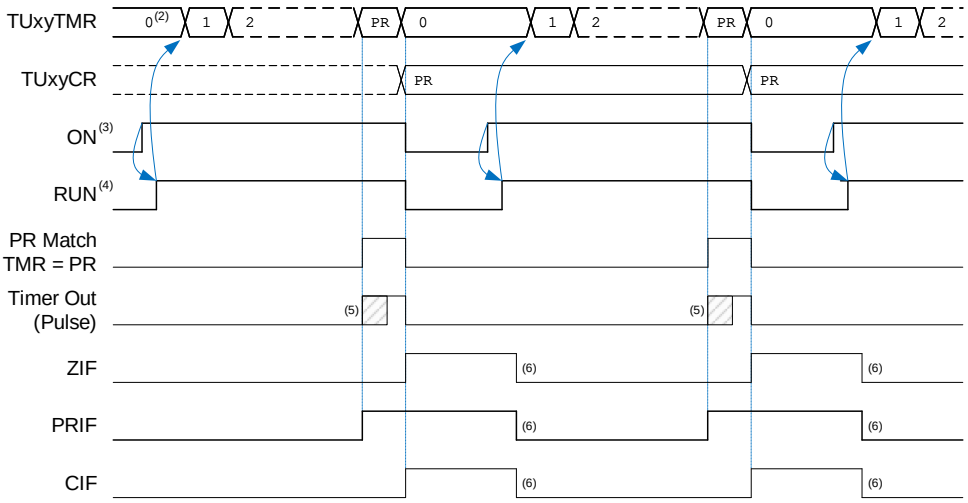
The timer can be configured for a simple software-triggered timeout operation as per the settings shown in Table 2-2.

Table 2-2. Timer Configuration

Timer Setting	Value
START	None (ON = 1)
RESET	At PR Match
STOP	At PR Match
CSYNC (Clock Sync)	Sync
OSEN (One-shot)	Enabled
PR (Period Register)	Desired Period Value - 1 (e.g., for a desired period of 20, PR = 19)

The counter counts until a PR match occurs, and then rolls over to zero and stops. The ON bit is cleared by one-shot operation. At PR match, the PRIF interrupt and output pulse occur, indicating the timeout. The software can set the ON bit again to start another time delay. This is shown in Figure 2-3.

Figure 2-3. Software Triggered Time Delay



Timer Setup:

START = None (ON = 1) RESET = At PR Match STOP = At PR Match
CSYNC = Sync OSEN = Enabled

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. The ON bit is set in the software and cleared by hardware upon Stop (one-shot mode).
4. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
5. The uncertainty of the output is due to the prescaler setting.
6. Cleared by software.

2.1.3 Hardware Triggered Time Delay

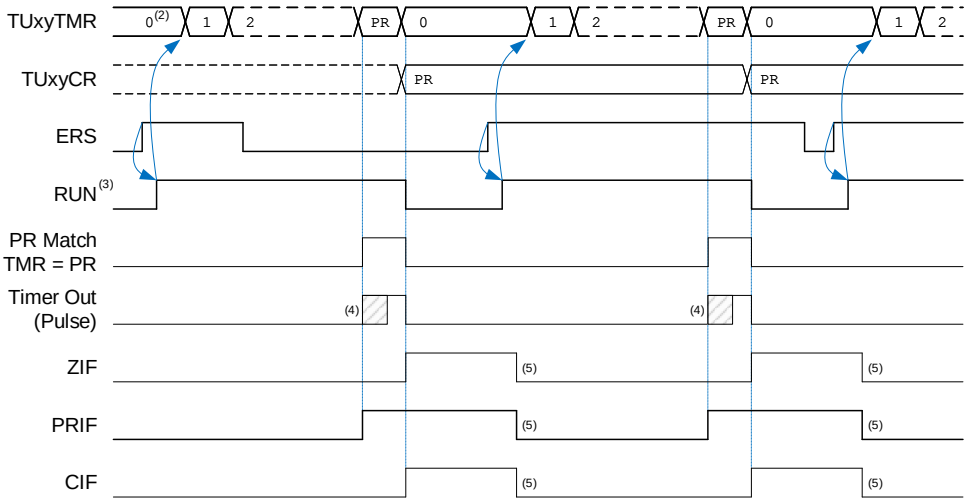
The timer can be configured for a simple hardware-triggered timeout operation as per the settings shown in Table 2-3.

Table 2-3. Timer Configuration

Timer Setting	Value
START	Rising ERS Edge
RESET	At PR Match
STOP	At PR Match
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to start at rising ERS edge) Inverted Level (to start at falling ERS edge)
PR (Period Register)	Desired Period Value - 1 (e.g., for a desired period of 20, PR = 19)

The counter starts counting when a rising ERS edge is detected, and counts until a PR match happens, then rolls over to zero and stops. At PR match, the PRIF interrupt and output pulse occur, indicating the timeout. The next rising ERS edge will restart the time delay. This is shown in Figure 2-4. To start the time delay at falling ERS edge, set the EPOL bit to invert ERS polarity. One-Shot mode can be enabled by setting the OSEN bit to add a layer of software control, if needed.

Figure 2-4. Hardware Triggered Time Delay



Timer Setup:

START = Rising ERS Edge RESET = At PR Match STOP = At PR Match
CSYNC = Sync

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. The uncertainty of the output is due to the prescaler setting.
5. Cleared by software.

2.1.4 Variable Time Delay

The Direct Memory Access (DMA) peripheral can be used in conjunction with the timer to create a variable output pulse period that updates at rollover. The timer can be configured as per the settings shown in Table 2-4. A DMA channel is configured to read from a program flash table with post-increment and rollover without termination. The PRIF timer interrupt can be used as a trigger to start the DMA transfer. Refer to TB3164: Direct Memory Access on 8-Bit PIC Microprocessor to learn how to configure the DMA peripheral.

Table 2-4. Timer Configuration

Timer Setting	Value
START	None (ON = 1)
RESET	At PR Match
STOP	None
CSYNC (Clock Sync)	Sync
PR (Period Register)	First period value; updated by DMA subsequently

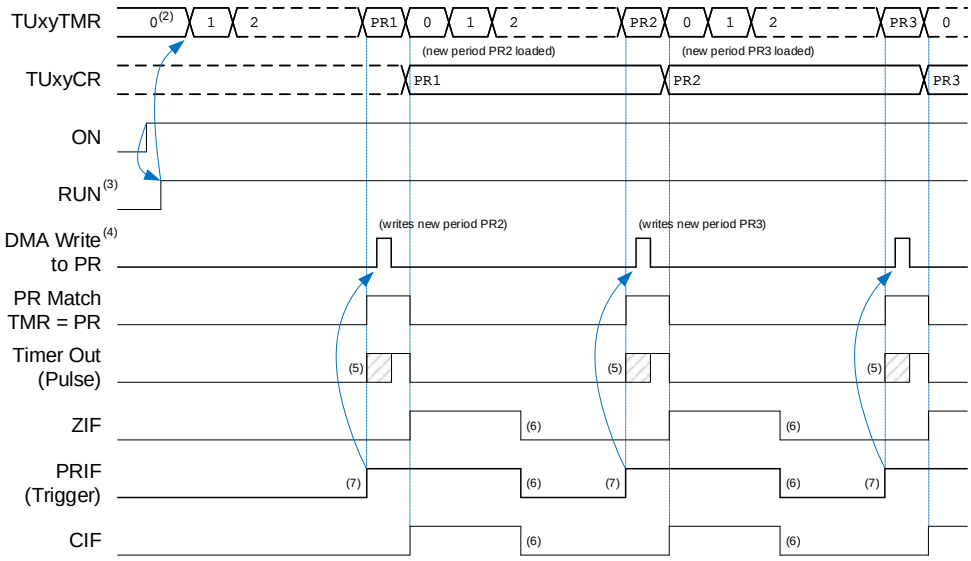
The counter counts until the first PR match occurs and then rolls over to zero and continues counting. At PR match, the PRIF interrupt triggers the DMA to transfer the new period value into the PR register and arms the loading of the new period value. The effective PR register is updated with the new period value at the next PR match. This is shown in Figure 2-5. Clock synchronization delays apply.

If an immediate PR load is desired, then set:

- ERS = PRL_Write
- RESET = ERS Level-0 + PR Match
- EPOL = 1 (Inverted Polarity)

In this case, as soon as the DMA triggers a transfer and writes to the TUxyPRL register, it arms the update. The write to the TUxyPRL register also generates a high ERS input which can be used to force a timer reset and load the new PR value. This is shown in Figure 2-6.

Figure 2-5. Variable Time Delay (Period Update at PR Match)



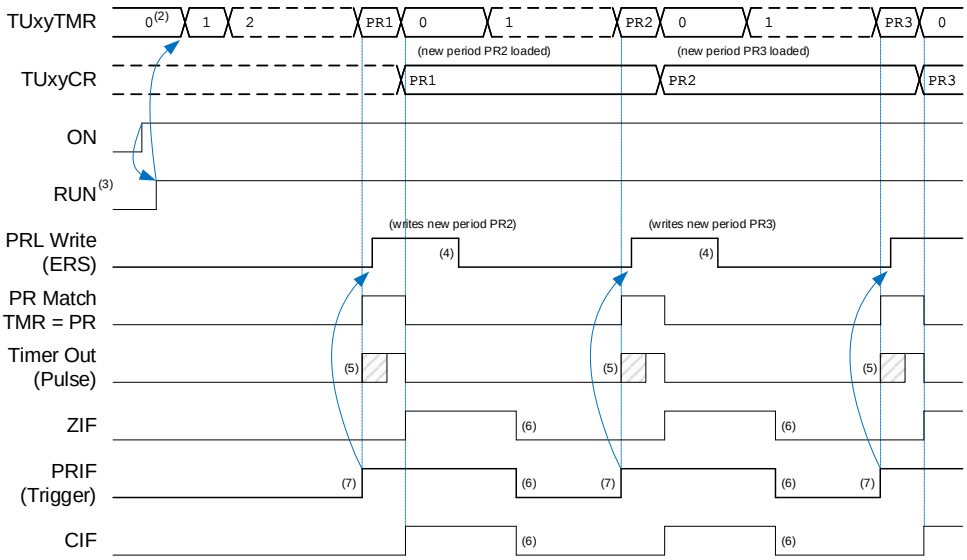
Timer Setup:

START = None (ON = 1) RESET = At PR Match STOP = None
CSYNC = Sync

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. DMA write signal is based on instruction clock. In the figure, the instruction clock (Fosc/4) is assumed to be faster than the timer clock. If the timer clock is faster than the instruction clock, then the DMA write will take longer than the PR match duration and the period may not update until the next PR match event.
5. The uncertainty of the output is due to the prescaler setting.
6. Cleared by software.
7. PRIF is used as a DMA trigger to write to TUxyPR register.

Figure 2-6. Variable Time Delay (ERS Update)



Timer Setup:

START = None (ON = 1)	RESET = ERS Level-0+PR Match	STOP = None
CSYNC = Sync	ERS = PRL_Write	EPOL = Inverted

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. The PRL_Write signal stays asserted for up to two timer clock cycles internally.
5. The uncertainty of the output is due to the prescaler setting.
6. Cleared by software.
7. PRIF is used as a DMA trigger to write to TUxyPR register.

2.2 External Clock Gating

The timer can measure the pulse width or period of the ERS input signal. The following use cases demonstrate this.

- Measure Input High/Low Pulse Width
- Measure Input Period
- Measure Input Frequency

2.2.1 Measure Input High/Low Pulse Width

The timer can be configured to measure the pulse width of the ERS signal as per the settings shown in Table 2-5.

Table 2-5. Timer Configuration

Timer Setting	Value
START	ERS Level-1
RESET	At Start + PR Match
STOP	Either ERS Edge
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to measure “high” pulse width) Inverted Level (to measure “low” pulse width)

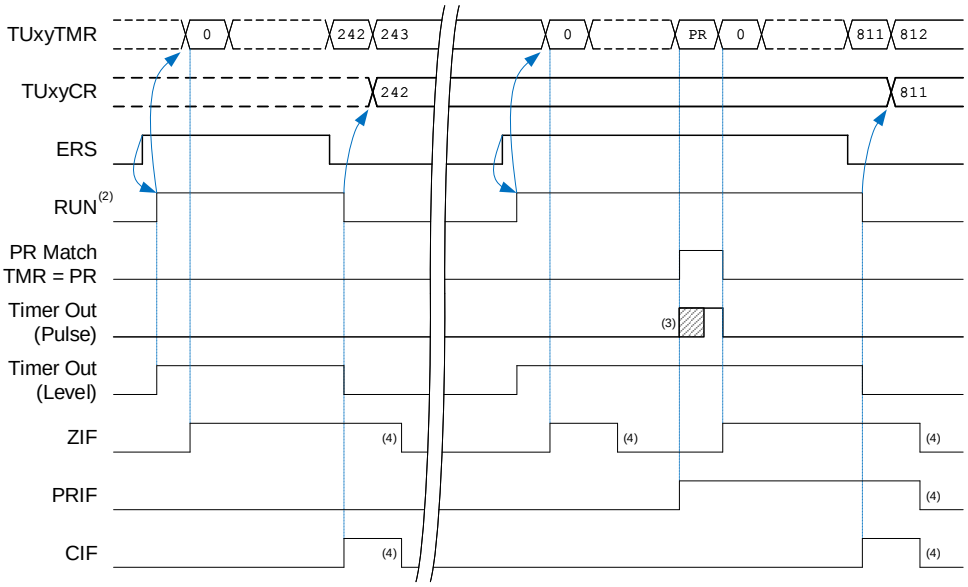
.....continued	
Timer Setting	**Value**
OSEN (One-shot)	True Level (to start at rising ERS edge) Inverted Level (to start at falling ERS edge)
PR (Period Register)	Desired Period Value

When the ERS input rises (changes from 0 to 1), the counter resets to zero and begins counting. When the counter resets to zero, ZIF interrupt occurs. At the following falling edge of ERS, the counter value is captured into the TUxyCR capture register, the counter stops, and CIF interrupt occurs. The captured value in the TUxyCR register is the 'high' pulse-width time. This is shown in Figure 2-7. To measure the 'low' pulse-width time, invert the ERS polarity by setting the EPOL bit.

The software must read the captured value before the completion of the next pulse. The One-Shot mode can be used to disable the subsequent start edge, allowing more time for the software to read the captured value.

This mode can be used to determine if the RS232, I^2C or the SPI bus has been idle for an extended amount of time (with EPOL = 1 to measure 'low' time). The PRIF interrupt occurs only if the idle time exceeds the PR value. Alternatively, this can also be measured using the use case in Input Stuck High/Low.

Figure 2-7. Measure Input Pulse Width



Timer Setup:
START = ERS Level-1 RESET = At Start+PR Match STOP = Either ERS Edge
CSYNC = Sync

Note:
1. Cross-domain clock synchronization applies as required but is not highlighted.
2. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
3. The uncertainty of the output is due to the prescaler setting.
4. Cleared by software.

2.2.2 Measure Input Period

The timer can be configured to measure the period of the ERS signal as per the settings shown in Table 2-6.

Table 2-6. Timer Configuration

Timer Setting	Value
START	Rising ERS Edge
RESET	At Start + PR Match
STOP	Rising ERS Edge
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to measure rising-to-rising edges period) Inverted Level (to measure falling-to-falling edges period)
OSEN (One-shot)	Enabled (for single measurement) Disabled (for continuous measurement)
PR (Period Register)	Desired Period Value

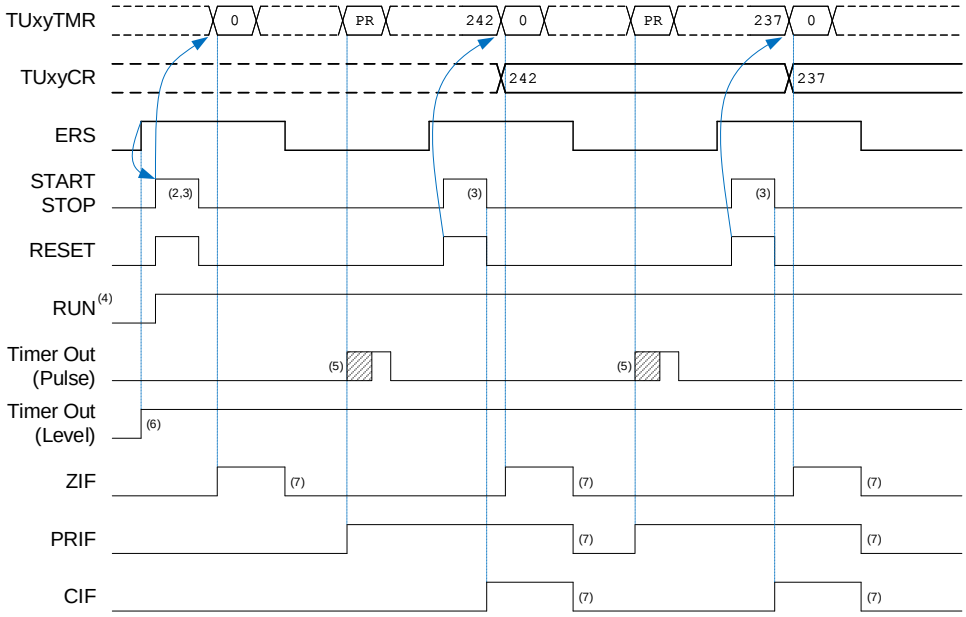
When the ERS input rises (changes from 0 to 1), the counter resets to zero and begins counting. When the counter resets to zero, ZIF interrupt occurs. At the next rising edge, the counter value is captured into the TUxyCR capture register, where it can be read by software, and CIF is set. This is shown in Figure 2-8.

Because Start, Stop and Reset all occur at the same time, the counter is immediately reset and resumes counting. The software must read the captured value before the next input period is complete or the TUxyCR capture register will be overwritten with the next measurement. Alternatively, use One-Shot mode by setting the OSEN bit to stop after the first measurement and prevent a data overrun.

The CIF interrupt occurs when each measurement completes. Technically, the CIF interrupt and timer capture should occur with the Stop condition that is present when first starting, but is withheld because it does not mark a completed measurement.

To measure the input period from falling-edge to falling-edge, invert ERS polarity by setting the EPOL bit.

Figure 2-8. Measure Input Period



Timer Setup:
START = Rising ERS Edge RESET = At Start+PR Match STOP = Rising ERS Edge
CSYNC = Sync

Note:
1. Cross-domain clock synchronization applies as required but is not highlighted.
2. A coincident Start/Stop condition that starts the counter does not cause either a capture or CIF to be set.
3. A synchronous edge-triggered Start/Stop condition is one timer clock cycle wide internally.
4. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
5. The uncertainty of the output is due to the prescaler setting.
6. Timer Out (Level) rises along with ERS when START = Rising/Either ERS Edge.
7. Cleared by software.

2.2.3 Measure Input Frequency

The setup for Measure Input Period to measure the input period can also be used to measure the frequency of an unknown clock. Set up the timer as per Table 2-6. But instead of selecting a system oscillator as a timer clock, connect the unknown signal from the I/O pin (PPS input) or CLC. Then select the ERS input as a pulse train or clock of known period T_{SAMPLE} (for example, the output of a crystal-clocked timer or a Pulse-Width Modulator (PWM)). The CIF interrupt will occur with each cycle of ERS, and the frequency of the input is then the captured value in $TUxyCR/T_{SAMPLE}$.

As an example, suppose the ERS-input timer is set to overflow 100 times every second, making T_{SAMPLE} = 10 ms. Let's say that when CIF occurs, the value of TUxyCR = 7300. Upon calculation, the input frequency comes out to be 7300/10 ms = 730 kHz.

2.3 Hardware Limit Timing (HLT)

The timer hardware can determine if the ERS input has been high, low, or unchanged for more than a specified amount of time. The following use cases demonstrate these:

- Input Stuck High/Low
- Pulse-Width Detector
- Pulse Stretcher
- Switch Debouncer

- Pulse-Width Modulation Generator

2.3.1 Input Stuck High/Low

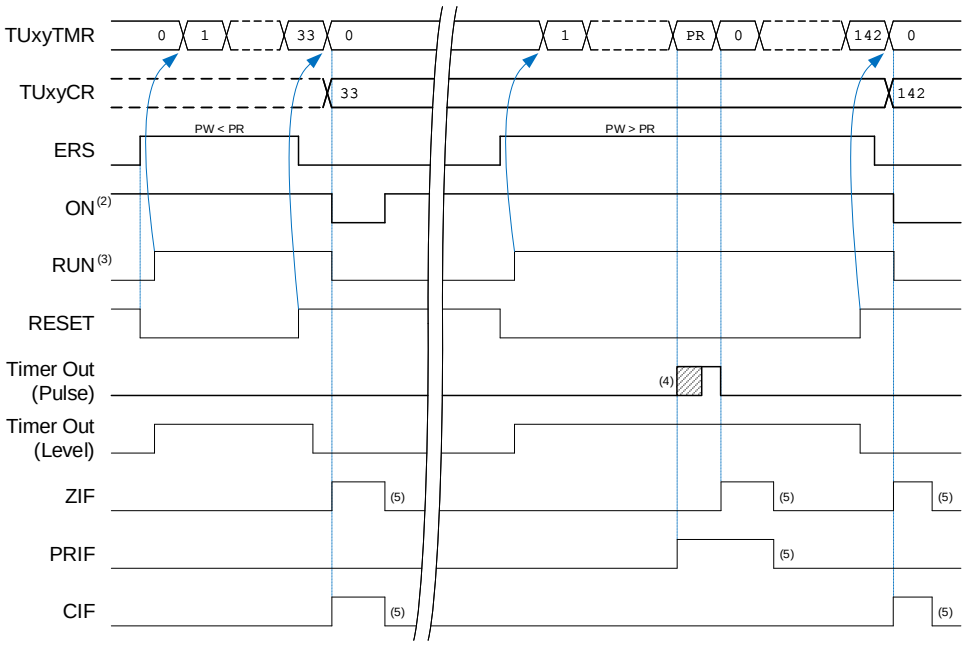
The timer can be configured to test if the ERS input has been high/low for a certain period of time as per the settings shown in Table 2-7.

Table 2-7. Timer Configuration

Timer Setting	Value
START	None (ON = 1)
RESET	ERS Level-0 + PR Match
STOP	Either ERS Edge
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to test for stuck-high) Inverted Level (to test for stuck-low)
OSEN (One-shot)	Enabled (for one-shot) Disabled (for auto-repeat)
PR (Period Register)	Desired time period to test

When the input rises (from 0 to 1), the counter will begin to count. When a PR match occurs, PRIF will occur indicating that the input has been stuck at high for a desired long enough time period. The counter resets and continues counting, and ZIF is set. When ERS falls, the timer stops counting, CIF is set, and TUxyCR captures the length of the pulse. The OSEN bit can be set to make it a one-shot operation, and cleared for auto-repeat operation. This is shown in Figure 2-9. If testing for stuck-low, set EPOL = 1.

Figure 2-9. Input Stuck High



Timer Setup:

START = None (ON = 1)
CSYNC = Sync
RESET = ERS Level-0+PR Match
OSEN = Enabled
STOP = Either ERS Edge

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. The ON bit is set in the software and cleared by hardware upon Stop (one-shot mode).
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. The uncertainty of the output is due to the prescaler setting.
5. Cleared by software.

2.3.2 Pulse-Width Detector

The timer can be configured to detect a minimum pulse width when combined with a D Flip Flop in a CLC peripheral as per the settings shown in Table 2-8. The output of the timer is connected to the clock input of the D flip flop. Set and Clear of the D flip flop are tied low. The D input of the flip flop is connected to the same ERS input signal. This setup is shown in Figure 2-11.

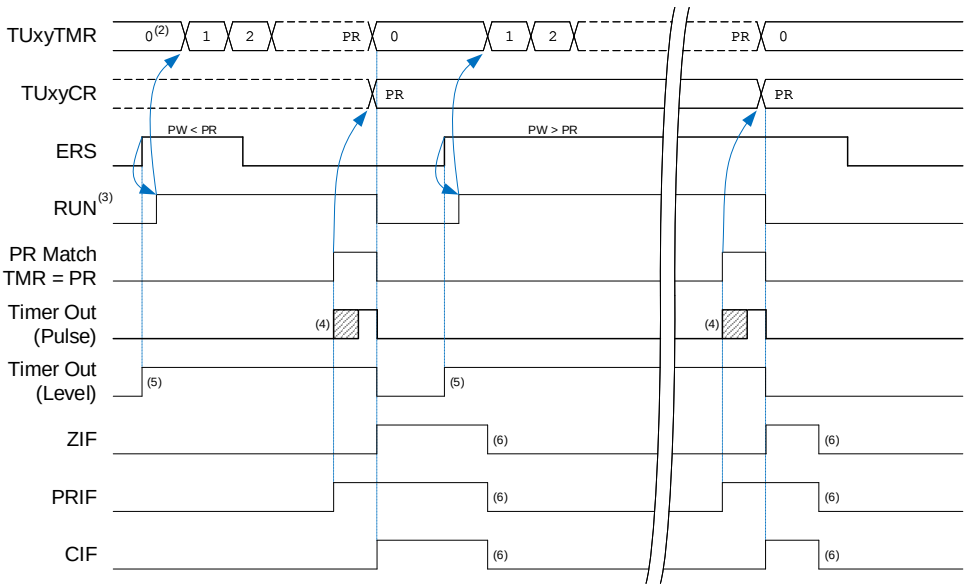
Table 2-8. Timer Configuration

Timer Setting	Value
START	Rising ERS Edge
RESET	At PR Match
STOP	At PR Match
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to test for high pulse width) Inverted Level (to test for low pulse width)
OSEN (One-shot)	Enabled (for one-shot) Disabled (for auto-repeat)
PR (Period Register)	Desired minimum pulse width

When the input rises (from 0 to 1), the counter will begin to count. When PR match occurs, PRIF is set and an output pulse will occur. This is shown in Figure 2-10. This output pulse triggers the clock input of the D Flip Flop in the CLC peripheral to latch the state of D-input. If the pulse width is longer than the PR match period, the D flip flop will latch a high output. If the pulse width is shorter than the PR match period, the D flip flop will latch a low output.

The OSEN bit can be set to make it a one-shot operation, and cleared for auto-repeat operation. If testing for low pulse-width time, set EPOL = 1 and invert the output of the CLC.

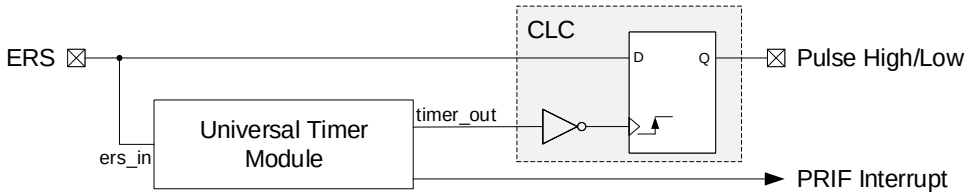
Figure 2-10. Pulse-Width Detector



Timer Setup:
START = Rising ERS Edge RESET = At PR Match STOP = At PR Match
CSYNC = Sync

Note:
1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. The uncertainty of the output is due to the prescaler setting.
5. Timer Out (Level) rises along with ERS when START = Rising/Either ERS Edge.
6. Cleared by software.

Figure 2-11. UTMR and CLC Connection



2.3.3 Pulse Stretcher

The timer can be configured to produce a minimum pulse width when the ERS input rises as per the settings shown in Table 2-9.

Table 2-9. Timer Configuration

Timer Setting	Value
START	Rising ERS Edge

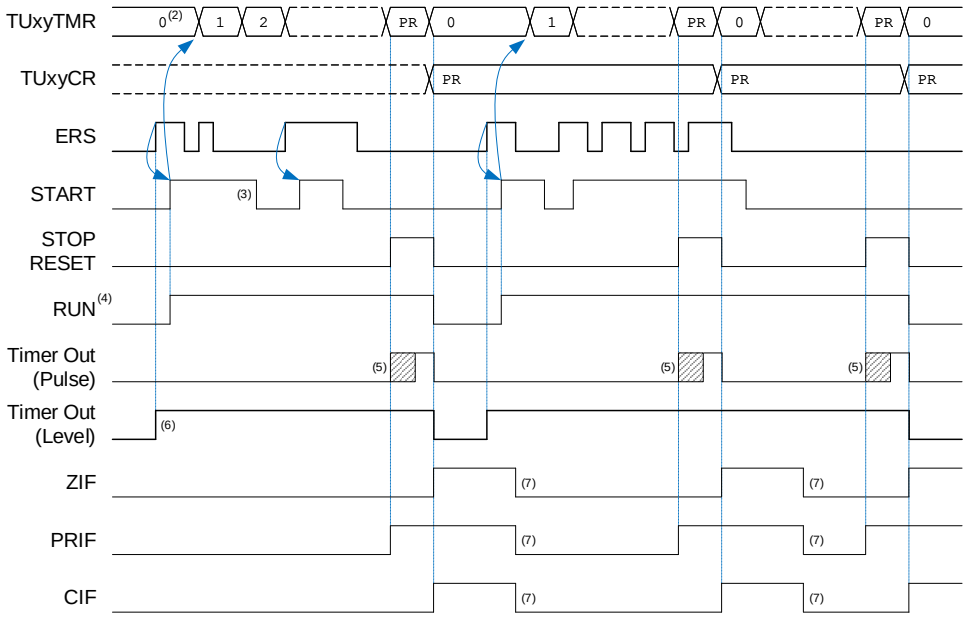
.....continued	
Timer Setting	**Value**
RESET	At PR Match
STOP	At PR Match
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to trigger to rising edge) Inverted Level (to trigger to falling edge)
PR (Period Register)	Desired minimum pulse width

Initialize the timer counter by setting the CLR command. When the input rises (changes from 0 to 1), the counter will begin counting. Any more input changes will be ignored unless a rising edge occurs near the PR match, which will override the Stop condition, reset the counter, and continue counting. This is shown in Figure 2-12.

If OM = 1 (level), the output will be a pulse of duration (timer clock period) * (PR+1), beginning immediately upon the input edge, and continuing as long as ERS toggles. If OM = 0 (pulse), a pulse will appear some time after the original rising edge when the PR match occurs, and possibly repeat periodically.

Triggering from falling edges or either edge is accomplished by setting EPOL = 1 or START = Either ERS Edge.

Figure 2-12. Pulse Stretcher



Timer Setup:

START = Rising ERS Edge RESET = At PR Match STOP = At PR Match
CSYNC = Sync

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. Ensure that TUxyTMR counter is reset to zero by setting CLR command.
3. A synchronous edge-triggered Start/Stop condition is one timer clock cycle wide internally. Multiple consecutive edges in short interval of time may cause the internal Start/Stop condition to remain set for a longer time duration.
4. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
5. The uncertainty of the output is due to the prescaler setting.
6. Timer Out (Level) rises along with ERS when START = Rising/Either ERS Edge.
7. Cleared by software.

2.3.4 Switch Debouncer

The timer can be configured to be used as a hardware-based switch debouncer as per the settings shown in Table 2-10.

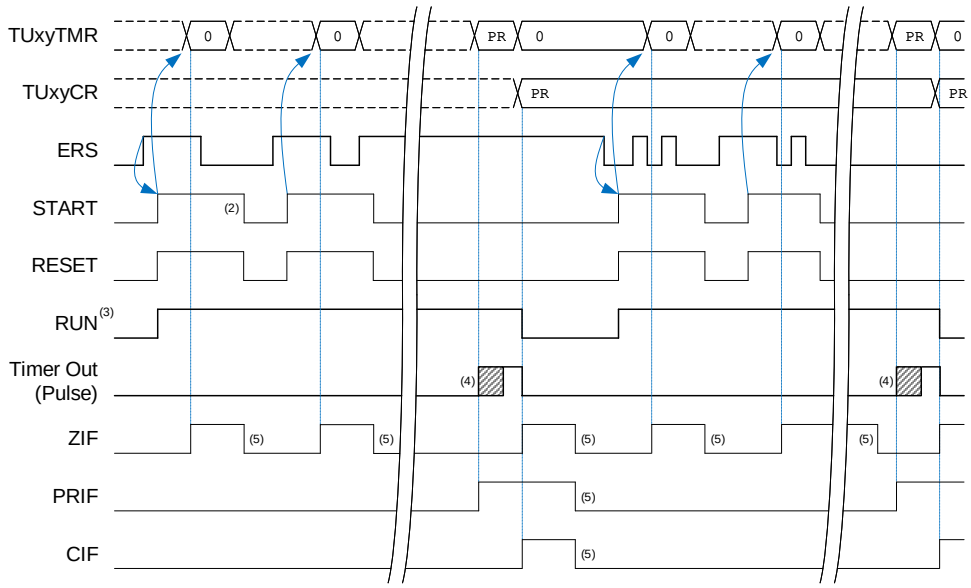
Table 2-10. Timer Configuration

Timer Setting	Value
START	Either ERS Edge
RESET	At Start + PR Match
STOP	PR Match
CSYNC (Clock Sync)	Sync
EPOL (ERS Polarity)	True Level (to trigger to rising edge) Inverted Level (to trigger to falling edge)
PR (Period Register)	Desired minimum pulse width

Initialize the timer counter by setting the CLR command. When the input rises (changes from 0 to 1), the counter will begin counting. Any more input changes will be ignored unless a rising edge occurs near the PR match, which will override the Stop condition, reset the counter, and continue counting. This is shown in Figure 2-13.

With any input change, the counter is reset and begins counting. When the counter reaches PR, the output changes and causes an interrupt. This is shown in Figure 2-13 below. The interrupt software, knowing that the input is stable, can read the raw switch input to determine if the switch is open or closed. Alternatively, the D flip-flop in a CLC may be used to sample the input, as illustrated below in Figure 2-14.

Figure 2-13. Switch Debouncer



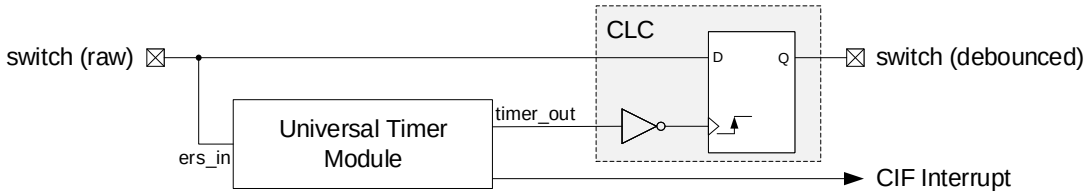
Timer Setup:

START = Either ERS Edge RESET = At Start+PR Match STOP = At PR Match
CSYNC = Sync

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. A synchronous edge-triggered Start/Stop condition is one timer clock cycle wide internally. Multiple consecutive edges in short interval of time may cause the internal Start/Stop condition to remain set for a longer time duration.
3. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
4. The uncertainty of the output is due to the prescaler setting.
5. Cleared by software.

Figure 2-14. UTMR and CLC Connection



2.3.5 Pulse-Width Modulation (PWM) Generator

The timer can be configured to generate a PWM signal when combined with a periodic ERS input signal. The timer rollover period must be greater than or equal to the ERS input period for correct operation. The PWM period is the period of the ERS input signal. The PWM duty cycle is determined by the PR value in the TUxyPR period register. The TUxyPR period register is also double-buffered to avoid glitches when changing the duty cycle. Use OM = Level mode to generate a PWM output. The OPOL bit controls whether the PWM signal is left-aligned or right-aligned.

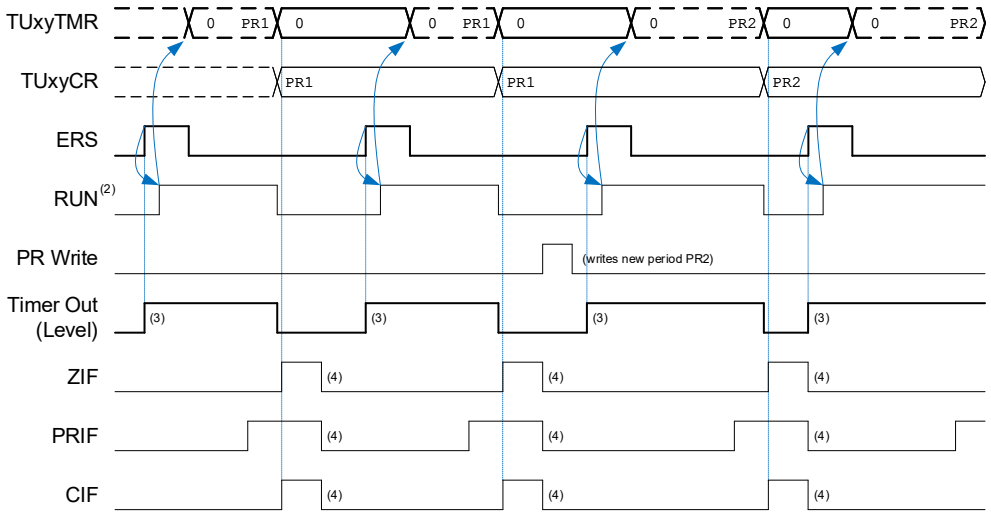
The timer configuration settings are shown in Table 2-11.

Table 2-11. Timer Configuration

Timer Setting	Value
START	Rising ERS Edge
RESET	At Start + PR Match
STOP	At PR Match
CSYNC (Clock Sync)	Sync
ERS (Ext. Reset Source)	Periodic input of desired PWM frequency
OM (Output Mode)	Level mode
OPOL (Output Polarity)	Low when Idle (for left-aligned PWM) High when Idle (for right-aligned PWM)
PR (Period Register)	Calculated as per desired duty cycle and OPOL setting

When the periodic input rises (changes from `0` to `1`), the timer is reset and starts counting. This process asserts the level output. The timer stops counting and resets at PR match, which deactivates the level output. This process repeats when the next rising input edge is detected. The level output generated is a PWM waveform. When OPOL = `0` (low when idle), the waveform is left-aligned PWM while, when OPOL = `1` (high when idle), the waveform is right-aligned PWM. A left-aligned PWM is shown in Figure 2-15 below.

Figure 2-15. PWM Generator



Timer Setup:

START = Rising ERS Edge
CSYNC = Sync
RESET = At PR Match
OPOL = Low when Idle
STOP = At PR Match

Note:

1. Cross-domain clock synchronization applies as required but is not highlighted.
2. The RUN trace illustrates the internal Timer Clock domain run/stop signal. Clock sync delays apply before the value appears in the RUN SFR bit.
3. Timer Out (Level) rises along with ERS when START = Rising/Either ERS Edge.
4. Cleared by software.

The period of the PWM is dependent on the period of the ERS input signal:

$$PWM\ Period\ =\ ERS\ Input\ Period$$

When OPOL = 0 (left-aligned PWM), the PR value determines the ON time. Hence, pulse width of the signal is:

$$Pulse\ Width\ =\ (PR\ value\ +1)\ \times\ TUCLK\ period$$

When OPOL = 1 (right-aligned PWM), the PR value determines the OFF time. Hence, pulse width of the signal is:

$$Pulse\ Width\ =\ PWM\ Period\ -\ ((PR\ value + 1) \times TUCLK\ period)$$

The duty cycle ratio is a function of the pulse width and period of the PWM signal:

$$Duty\ Cycle\ Ratio\ =\ \frac{Pulse\ Width}{PWM\ Period}$$

The resolution of the PWM is a function of the PWM period, UTMR rollover period, and UTMR size. The maximum PWM resolution is the size of the UTMR when the UTMR rollover period matches the PWM period.

$$PWM\ Resolution\ =\ \left\lfloor \log_2\left(\frac{PWM\ Period}{UTMR\ Rollover\ Period}\ \times\ 2^{UTMR\ Size}\right)\right\rfloor bits$$

Important: The UTMR rollover period must be greater than or equal to the ERS input period for proper PWM operation.

As an example, the following calculations show how to determine PR value and PWM resolution to generate a left-aligned 1 kHz PWM signal with 30% duty cycle. For this calculation, it is assumed that the UTMR clock, TUCLK = F_{OSC} = 16 MHz and UTMR size = 16 bits.

$$PWM\ Period\ =\ ERS\ Input\ Period\ =\ 1/(1\ kHz)\ =\ 1\ ms$$

$$Pulse\ \ Width\ =\ Duty\ Cycle\ Ratio \times PWM\ Period\ =\ 30\,\%\ \times\ 1\ ms\ =\ 300\ \mu s$$

$$PR\ value\ =\ \frac{Pulse\ \ Width}{TUCLK\ Period} - 1 = \frac{300\,\mu s}{1/16\,MHz} - 1 = \frac{300\,\mu s}{62.5\ ns} - 1 = 4799$$

$$PWM\ Resolution = \left\lfloor log_2\left(\frac{1\ \ ms}{4.096\ \ ms}\ \times\ 2^{16}\right)\right\rfloor = 13\ bits$$

2.4 External Clock Operation

The timer can operate from an external aperiodic clock by setting CSYNC = 0. Refer to Synchronous vs. Asynchronous Operation for more information about clock synchronization. Note that with CSYNC = 1, clock synchronization delays will apply, which is unreliable since the input signal is aperiodic and may not be toggling. With CSYNC = 0, the effect of the gate is immediate, without synchronization delays. For reliable operation the user must ensure that the inputs meet reasonable setup and hold times of a few nanoseconds. The following use case demonstrates this.

2.4.1 Input Counter

The timer can be configured to be clocked from an external aperiodic source as per the settings shown in Table 2-11.

Table 2-12. Timer Configuration

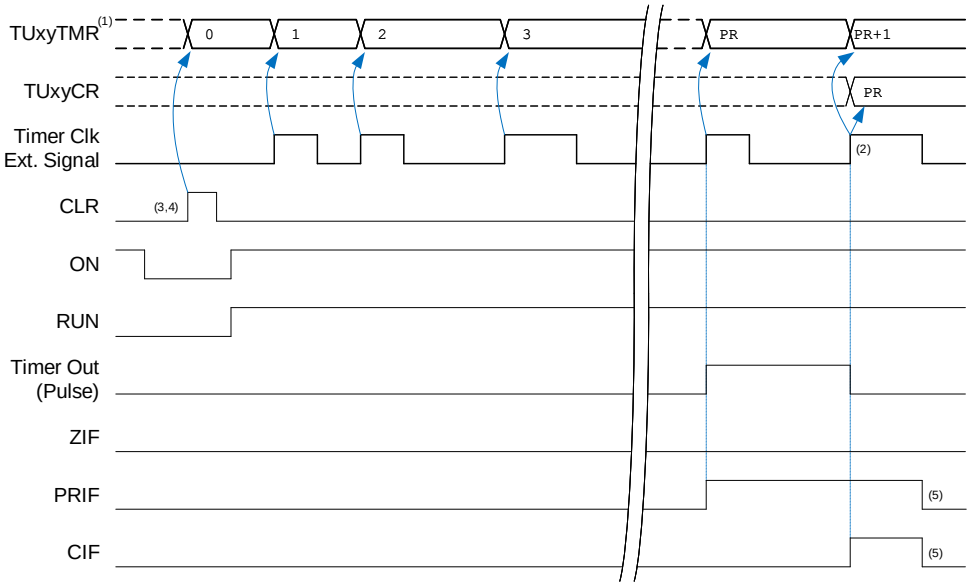
Timer Setting	Value
START	None (ON = 1)
RESET	None
STOP	At PR Match
CSYNC (Clock Sync)	Async
CLK (Timer Clock)	External Input
RDSEL (Read Select)	TUxyTMR Access

With the external input being used as the clock, the counter will advance with each clock after the timer is enabled (ON = 1). The rest of the timer module will operate the way it is configured – this includes ERS input, Stop, Reset, PR match, interrupts, etc.

Reading from the TUxyCR capture register is not desirable in this use case because the capture event requires at least one timer clock to happen, which is not ensured in an aperiodic operation. Hence, the user can read the TUxyTMR counter register directly with RDSEL = 1, keeping in mind that the TUxyTMR register is not buffered and is not guarded against clock collisions when read by software.

The clock input may be filtered using a separate debouncer, as shown in Figure 2-16.

Figure 2-16. Input Counter



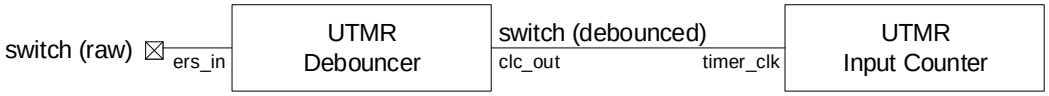
Timer Setup:

START = None (ON = 1)	RESET = None	STOP = At PR Match
CSYNC = Async	RDSEL = TUxyTMR access	

Note:

1. TUxyTMR is not double buffered and software access is not guarded for clock collisions.
2. Capture and CIF interrupt occur only with an additional input clock after pulse output is asserted.
3. The effect of CLR command is immediate when CSYNC = 0 and ON = 0.
4. ZIF interrupt does not occur because timer is not running (ON = 0).
5. Cleared by software.

Figure 2-17. Cascading Input Counter with Debouncer



3. Conclusion

The UTMR module can be configured in many different ways and be customized to operate in a variety of applications like periodic operation, external clock gating, hardware limit timing, and external clock operation. With features like the inbuilt capture, compare, and customizable Start, Reset, Stop conditions, the UTMR is quite versatile.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-6843-1

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820