ABSTRACT

This technical brief provides information about the flexible and very useful Signal Measurement Timer (SMT). The SMT is a 24-bit counter/timer with advanced clock and gating logic and is available on a range of PIC® microcontrollers. It can measure a wide variety of digital signal parameters such as pulse width, frequency, duty cycle and time difference between the edges of two signals.

Table 1 shows various modes of operation of SMT. Selecting between the modes and input clock resources is accomplished by setting or clearing the bits in SMT control registers. For more details on SMT and selecting between modes of operation, refer to the respective device data sheet.

FIGURE 1: SMT BLOCK DIAGRAM

SMT ON PIC MICROCONTROLLER

As shown in Figure 1, the SMT clock has five different clock sources all derived from the internal oscillator, namely Fosc, Fosc/4, HFINTOSC, LFINTOSC, MFINTOSC/16. The prescaler further reduces the clock to smaller clock frequencies set using the control register. The SMT has two external inputs, SMT_signal and SMT_window, which get their inputs either from the other on-chip peripherals like op amp, comparator, CCP, or from the external pin. The SMT_signal is used to provide signal to be measured by the SMT. The SMT_window is used as another signal source to measure the time difference, either as an event signal or as a clock in Counter mode, etc.
The SMT GO bit has to be set to start the timer. The SMT GO bit, the SMT signal input and the SMT window input are synchronized to the SMT clock. This synchronization requires one to two clock cycles, as shown in Figure 2. The synchronized signals (SMTxSIG_sync, SMTxWIN_sync, SMTxGO_sync) are used by the SMT further for timing and measurement.

### FIGURE 2: SMT SIGNAL SYNCHRONIZATION

The SMTxTMR register counts for the timer and SMTxPR register holds the count for comparison. The SMTxCPR and SMTxCPW are the SMT capture period and SMT capture pulse-width registers used to capture the timer register value on any SMT signal event.

### Modes of operation

The SMTxCON0 register contains an Enable bit to enable or disable the SMT. Enabling will supply power to the SMT peripheral. The STP bit in SMTxCON0, if set, will generate an overflow interrupt when SMTxTMR is equal to SMTxPR, and if reset, the SMTxTMR will overflow for a whole 24-bit count in SMTxTMR with the corresponding overflow interrupt. The WPOL, SPOL and CPOL bits in SMTxCON0 select the active state of SMT_window, SMT_signal and SMT clock input, respectively (i.e., setting them will make these inputs active-low signals with the falling edge as the trigger point). The SMTxPS bits in SMTxCON0 select the clock prescaler.

The SMTxGO bit in the SMTxCON1 register acts as a run control bit for the Timer (i.e., setting it will start the timer counter operation). The REPEAT bit in SMTxCON1, if set, repeats the acquisition until stopped by software using the SMTxGO bit and, if the bit is reset, the SMT peripheral will perform the acquisition one time and stop. The MODE bits in SMTxCON1 select the mode of operation for the SMT listed in Table 1.

### TABLE 1: SMT MODES OF OPERATION

<table>
<thead>
<tr>
<th>Mode</th>
<th>Modes of Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Timer</td>
</tr>
<tr>
<td>0001</td>
<td>Gated Timer</td>
</tr>
<tr>
<td>0010</td>
<td>Period and Duty Cycle Acquisition</td>
</tr>
<tr>
<td>0011</td>
<td>High and Low-Time Measurement</td>
</tr>
<tr>
<td>0100</td>
<td>Windowed Measurement</td>
</tr>
<tr>
<td>0101</td>
<td>Gated Windowed Measurement Mode</td>
</tr>
<tr>
<td>0110</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>0111</td>
<td>24-Bit Capture</td>
</tr>
<tr>
<td>1000</td>
<td>24-Bit Counter</td>
</tr>
<tr>
<td>1001</td>
<td>Gated Counter</td>
</tr>
<tr>
<td>1010</td>
<td>Windowed Counter</td>
</tr>
</tbody>
</table>
The SMTxSTAT register is the SMT status register. The CPRUP bit is used to update the SMT period and captured period register and the CPWUP bit is to update the SMT captured pulse-width register. This requires setting these bits in software. The RST bit, if set, requests the reset of the SMT timer counter register (SMTxTMR). The TS bit is a status bit; if set, it states that the SMT timer is incrementing and, if reset, it states the SMT timer is not incrementing. The WS bit states if the SMT window input is open when set; if reset, it states that the SMT window is closed. The AS bit determines that the SMT acquisition is in progress when set and not in progress when reset.

The SMTxCLK register selects the clock input for the SMT peripheral, as shown in Figure 1. The SMTxWIN and SMTxSIG registers select the window and signal input for SMT_window and SMT_signal. The SMTxTMR register is a SMT timer counter register used for up counting. The SMTxCPW and SMTxCPR are the SMT captured pulse-width register and SMT captured period register to store the value of SMTxTMR on signal events from the SMT inputs. The SMTxPR register is the SMT period register to hold the maximum count for the SMT timer up to counter the value of which is matched after every increment to SMTxTMR, and if matched, the period match interrupt is generated.

The SMT has 11 modes for signal measurement, as described briefly below:

1. **Timer**

The Timer mode is a simple timer operation with timing value of 24-bit. This mode is similar to the Timer mode of Timer0 or Timer1.

![Figure 3: Timer Mode](image)

2. **Gated Timer**

In this mode, the signal is provided on SMT_signal input and only the high or ON time of the signal is measured cumulatively while the timer is running. The timer value is captured whenever the input signal goes low or OFF in captured pulse-width register (SMTxCPW) with corresponding interrupt generated.

![Figure 4: Gated Timer Mode](image)

This mode can be used to find the total time for which equipment with a feedback control system is in operation. For example, to measure the total power consumption of a compressor motor of a refrigerator or a power supply for a consumer electronic equipment over a particular period.
3. **Period and Duty Cycle Measurement**

The Period and Duty Cycle acquisition mode is capable of detecting rising and falling edge events and measuring the period and duty cycle of an input signal on the SMT signal input pin. The timing register (SMTxTMR) starts incrementing when a rising edge event is detected on SMT signal input (SMTxSIG_sync), with the GO bit (SMTxGO) set. When a falling edge event occurs, the captured pulse-width register (SMTxCPW) is updated with the current value of the timing register (SMTxTMR). On the next rising edge event on the SMT signal input, the captured period register (SMTxCPR) is updated with the current value of the timing register (SMTxTMR) and the timing register (SMTxTMR) is reset. Individual interrupts are generated for each capture event. The selection of edge-to-start counting is programmable through the SMT control register. Setting the WPOL and SPOL bits in the SMTxCON0 register to '0' will enable rising edge control, while setting them to '1' will enable falling edge control. The timing diagram for this mode is shown in Figure 3.

![FIGURE 5: PERIOD AND DUTY CYCLE MODE (SINGLE ACQUISITION)](image)

This mode can be used to find the characteristics of a carrier signal on the receiver side of a communication system and demodulate signal with the help of a comparator and other peripherals. This can also be used for the signal measurement mode in a simple oscilloscope.

4. **High and Low Measurement**

This mode is similar to Period and Duty Cycle measurement mode. The difference in this mode is that the timer count value resets on every signal edge and restarts from '0'.

![FIGURE 6: HIGH AND LOW MEASUREMENT MODE (SINGLE ACQUISITION)](image)
5. Windowed Measurement
This mode measures the window duration of the SMTxWIN input. The timer starts counting on the rising edge and keeps counting until next successive rising edge. The timer then resets and the timer value is captured into the capture period register with corresponding interrupt being generated. The selection of edge-to-start counting is programmable through the SMT control register.

FIGURE 7: WINDOWED MEASUREMENT MODE (SINGLE ACQUISITION)

6. Gated Windowed Measurement
In Gated Window Measurement mode, the ON time of the SMT signal input (SMTxSIG) is measured between two consecutive rising edges of the SMT window input (SMTxWIN) called the measurement window. On every rising edge event on the SMT_window input, the SMT captured period register is updated with the current value of the timer and the corresponding interrupt is generated. The timer also resets and restarts counting.

FIGURE 8: GATED WINDOWED MEASUREMENT MODE (SINGLE ACQUISITION)

7. Time of Flight
In Time-of-Flight mode, the SMT measures the time between a window input event and a signal input event, that is the time between first rising edges of the window input (SMTxWIN) and the signal input (SMTxSIG). The input edges can be configured either to be rising or falling. The timer starts counting when the first rising edge on the SMT_window input is detected. The timer stops on the first rising edge of the SMT_signal input and the timer value is captured into the capture pulse-width register with corresponding interrupt generated. Setting the WPOL and SPOL bits in the SMTxCON0 register to '0' will enable rising edge control and setting them to '1' will enable falling edge control. The time of flight restarts after this.
FIGURE 9: TIME OF FLIGHT (REPEAT ACQUISITION)

This mode can be used in SONAR, RADAR, collision detection for vehicles and Ultrasonic Anemometer (wind direction and speed measurement), etc.

8. 24-bit Capture

In this mode, the signal is given on the SMT_window input. The timer starts counting as soon as the timer GO bit (SMTxGO) is set and stops when GO bit is reset. When a rising edge is detected on the SMT_window input, the current timer value is captured into the capture period register with the corresponding interrupt generated. When a falling edge is detected on the SMT_window input, the current timer value is captured into the capture pulse-width register with the corresponding interrupt generated.

FIGURE 10: 24-BIT CAPTURE (REPEAT ACQUISITION)

9. 24-bit Counter

In this mode, the internal SMT clock is disabled and the signal from SMT_signal is taken as the clock. The timer starts counting after the GO bit is set and stops counting when the GO bit is reset. On every rising edge event on the SMT_window input signal, the current value of timer is captured into the capture period register with the corresponding interrupt generated.

FIGURE 11: 24-BIT COUNTER
10. Gated Counter

In this mode, the SMT_signal input acts as a clock for SMT. The timer starts counting on a rising edge on the SMT_window input and stops counting on a falling edge on the SMT_window input. The timer value does not reset on any window input edge but is held in timer register. The timer resets only when the GO bit is reset. On every falling edge on the SMT_window input, the current timer value is captured into the capture pulse-width register with the corresponding interrupt generated.

FIGURE 12: GATED COUNTER MODE (REPEAT ACQUISITION)

11. Windowed Counter

In this mode, the SMT_signal input acts as a clock for the SMT. The timer starts counting on a rising edge of the SMT_window input. On the next falling edge, on the SMT_window input, the current value of the timer is captured into the capture pulse-width register with the corresponding interrupt generated. On the next rising edge on the SMT_window input, the current timer value is captured into the capture period register with the corresponding interrupt generated and the timer value resets to ‘0’ to restart counting.

FIGURE 13: WINDOWED COUNTER MODE (REPEAT ACQUISITION)
CONFIGURATION OF THE SMT PERIPHERAL

The SMT SFR list is given in Appendix A: “SMT SFRs”.

Example 1 is a simple code snippet to configure the SMT for Time-of-Flight mode for the XC8 compiler.

The MCC or MPLAB® X Code Configurator is a plug-in for MPLAB X to simplify code generation for the peripherals in a short amount of time.

After installing MCC, go to Menu “Tools -> Embedded -> MPLAB Code Configurator” to select the SMT peripheral, follow the left side panel under box “Search for modules…” textbox as shown in the below Figure 14.

Figure 15 shows the main MCC GUI for SMT with Timer of Flight in Capture mode selected.

FIGURE 14: SMT SELECTION IN MCC

Description

There are three modes for each SMT module within a device Capture, Counter and Timer.


2. The Counter mode contains a 24-bit counter, gated counter and Windowed Counter modes.

3. The Timer mode contains timer, gated timer, period and duty cycle measurement and High and Low measure mode.

FIGURE 15: MCC MAIN GUI FOR SMT
An example for configuration of SMT in MCC as shown in Figure 15 is described below:

1. The operation mode selected is Time of flight.
2. The Clock section selects the clock for Capture mode with prescaler and edge selection for counting operation.
3. The Window section selects the SMT Window input from the drop-down menu with polarity selection of input.
4. The Signal section selects the SMT Signal input from the drop-down menu with polarity selection of input.
5. The Period text box sets the SMTxPR register which is a 24-bit value.
6. There are two checkboxes at the top to enable SMT and start SMT after initialization.
7. These checkboxes at the bottom select the type of interrupts to be included in the Interrupt Service Routine.

8. The right side of GUI shows Pin Manager for SMT (i.e., if the SMT Window and Signal inputs are provided input from outside the device; this section configures the port pin to be connected for SMT input). Package selection is also available at the top for the device.

9. The unlock sign indicates that the input is not configured, while lock sign indicates the input is configured.

The “Generate Code” at the top is clicked when all settings are done to generate code automatically. After clicking the Generate Code button, the project in MPLAB X generates all the APIs for SMT as listed in Table 2:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void SMT1_Inititalize()</td>
<td>Initializes SMT registers according to user settings in MCC</td>
</tr>
<tr>
<td>void SMT1_DataAcquisitionEnable()</td>
<td>Sets the SMTxGO bit to start data acquisition</td>
</tr>
<tr>
<td>void SMT1_HaltCounter()</td>
<td>SMTxTMR Counter stops at SMTxPR and overflow interrupt occurs</td>
</tr>
<tr>
<td>void SMT1_RepeatDataAcquisition()</td>
<td>Data acquisition is repeated</td>
</tr>
<tr>
<td>void SMT1_ManualPeriodBufferUpdate()</td>
<td>Changes the value of the SMTxPR register</td>
</tr>
<tr>
<td>void SMT1_ManualPulseWidthBufferUpdate()</td>
<td>Changes the value of the SMTxCPW register</td>
</tr>
<tr>
<td>void SMT1_ManualTimerReset()</td>
<td>Resets the SMTxTMR register</td>
</tr>
<tr>
<td>bool SMT1_IsWindowOpen()</td>
<td>Determines if SMT window input is available</td>
</tr>
<tr>
<td>bool SMT1_IsSignalAcquisitionInProgress()</td>
<td>Determines if acquisition is in progress</td>
</tr>
<tr>
<td>bool SMT1_IsTimerIncrementing()</td>
<td>Determines if Timer counting operation is in progress</td>
</tr>
<tr>
<td>uint32_t SMT1_GetPulseWidth()</td>
<td>Returns the value from the SMTxCPW register</td>
</tr>
<tr>
<td>uint32_t SMT1_GetPeriod()</td>
<td>Returns the value of the SMTxCPR register</td>
</tr>
<tr>
<td>uint32_t SMT1_GetTimerValue()</td>
<td>Returns the current value of the SMTxTMR register</td>
</tr>
<tr>
<td>void SMT1_PR_ACQ_ISR()</td>
<td>Implements Period Acquisition Interrupt ISR</td>
</tr>
<tr>
<td>void SMT1_PW_ACQ_ISR()</td>
<td>Implements Pulse Width Acquisition Interrupt ISR</td>
</tr>
<tr>
<td>void SMT1_Overflow_ISR()</td>
<td>Implements Counter Overflow Interrupt ISR</td>
</tr>
</tbody>
</table>

**Note 1:** For the latest version of the MCC GUI for SMT, visit Microchip’s web site.
EXAMPLE 1:  SMT CONFIGURATION FOR TIME-OF-FLIGHT MODE

```
SMT1CON0bits.EN = 1;        //Enable SMT 1
SMT1CON0bits.STP = 1;       //SMT1 will halt at period=counter and restart
SMT1CON0bits.WPOL = 0;      //SMT1 window input is active when high or during rising edge
SMT1CON0bits.SPOL = 0;      //SMT1 signal input is active when high or during rising edge
SMT1CON0bits.CPOL = 0;      //SMT1TMR increments on rising edge
SMT1CON0bits.SMT1PS = 0b00; //Last two prescale bit set to 0 i.e. 1:1 clock prescale
SMT1PR = 0xFFFFFF;          //Period register set to high to be able to acquire longest signal
SMT1CON1bits.MODE = 0b0110; //SMT1 in time of flight mode
SMT1CLK = 0x00;             //Clock input is FOSC/4
SMT1SIG = 0x02;             //SMT1 signal is inputted from comparator 2
SMT1WIN = 0x01;             //SMT1 window is inputted from comparator 1
SMT1CON1bits.REPEAT = 1;    //Acquisition will continue
SMT1IE = 1;                 //SMT1 interrupt is enabled
SMT1PWAE = 1;               //SMT1 pulse width interrupt is enabled
SMT1PRAIE = 1;              //SMT1 period interrupt is enabled
SMT1GO = 1;                 //SMT1 is started to run
```

CONCLUSION

This technical brief familiarizes the reader with the Signal Measurement Timer (SMT) on PIC microcontrollers. The technical brief also explains the various modes of operation with a few of its possible applications. For certain important modes like Period and Duty Cycle Measurement, Gated Window Measurement and Time of Flight, illustrations with timing diagrams are also provided. For Time-of-Flight mode, the configuration of SMT is provided with a code snippet, showing the settings of the different registers.
APPENDIX A: SMT SFRs

Below is the list of registers (SFRs) for the SMT peripheral (see Register A-1).

### REGISTER A-1: SUMMARY OF SFR REGISTERS ASSOCIATED WITH SMT

<table>
<thead>
<tr>
<th>Reg Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTxCLK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMTxCON0</td>
<td>EN</td>
<td></td>
<td>STP</td>
<td>WPOL</td>
<td>SPOL</td>
<td>CPOL</td>
<td></td>
<td>SMTxPS&lt;1:0&gt;</td>
</tr>
<tr>
<td>SMTxCON1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMTxCPRH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxCPR&lt;15:8&gt;</td>
</tr>
<tr>
<td>SMTxCPRU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxCPR&lt;23:16&gt;</td>
</tr>
<tr>
<td>SMTxCPWH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxCPW&lt;15:8&gt;</td>
</tr>
<tr>
<td>SMTxCPWL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxCPW&lt;7:0&gt;</td>
</tr>
<tr>
<td>SMTxCPWU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxCPW&lt;23:16&gt;</td>
</tr>
<tr>
<td>SMTxPRH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxPR&lt;15:8&gt;</td>
</tr>
<tr>
<td>SMTxPRL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxPR&lt;7:0&gt;</td>
</tr>
<tr>
<td>SMTxPRU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SMTxPR&lt;23:16&gt;</td>
</tr>
<tr>
<td>SMTxSIG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SSEL&lt;2:0&gt;</td>
</tr>
<tr>
<td>SMTxSTAT</td>
<td>CPRUP</td>
<td>CPWUP</td>
<td>RST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TS</td>
</tr>
<tr>
<td>SMTxTMRH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WS</td>
</tr>
<tr>
<td>SMTxTMRL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AS</td>
</tr>
<tr>
<td>SMTxTMRU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMTxWIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WSEL&lt;3:0&gt;</td>
</tr>
</tbody>
</table>
Note the following details of the code protection feature on Microchip devices:

• Microchip products meet the specification contained in their particular Microchip Data Sheet.

• Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.

• There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.

• Microchip is willing to work with the customer who is concerned about the integrity of their code.

• Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “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.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBox, KEELoQ, KEELoQ logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC® logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MFPI, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SFI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestiC is a registered trademark of Microchip Technology Germany GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-051-6