Master Synchronous Serial Port (MSSP) to the Stand-Alone I²C Module Migration

Introduction

Microchip Technology Inc. has introduced an 8-bit stand-alone I²C communications module. The module includes new hardware features, which allow for more design flexibility and less software overhead. The stand-alone module differs from previous versions of the I²C module, which were included in the Master Synchronous Serial Port (MSSP) module. This migration document details the differences between the stand-alone module and the MSSP version.

A Brief Review of the I²C and SMBus Specifications

The I²C specification was developed by Phillips Semiconductors (now NXP Semiconductors) to communicate between devices connected to a two-wire bus. Phillips recognized that there were many similarities between consumer electronics, industrial electronics, and telecommunications designs.

Since the various designs often contained similar components, such as Analog-to-Digital Converters (ADCs), LCDs, or EEPROMs, Phillips determined that they could simplify system design and maximize hardware efficiency by creating a communication bus that could be used to transfer data between any device connected to the bus. This allowed designers to use devices from multiple manufacturers, or use one device in several designs. The specification also solved interfacing problems by creating a scheme that is now held as an industry standard, meaning any I²C device could communicate with any other I²C device without having to change the hardware or firmware of either device.

The I²C specification defines the bus as a two-wire, bidirectional communications scheme. One line carries the Serial Data (SDA), and one line carries the Serial Clock (SCL). Each I²C device has its unique address: either 7-bits or 10-bits in length. An I²C device can operate as a bus master, bus slave, or both, depending on the device and application. The specification defines the data transfer rates as follows:

- Standard mode – transfer rates up to 100 Kbits/s
- Fast mode – transfer rates up to 400 Kbits/s
- Fast mode Plus – transfer rates up to 1 Mbit/s
- High-Speed mode – transfer rates up to 3.4 Mbits/s

The SMBus specification is a derivative of the I²C specification. Both buses are basically compatible with each other, but it is important to understand the subtle differences.

The important differences between bus configurations are:

- **Bus time-out and clock speed** – A bus time-out occurs when the Serial Clock (SCL) is held low beyond a specified time limit. The I²C specification does not specify a time-out period, meaning that a slave device can hold (stretch) the clock line low indefinitely. The SMBus specification defines a time-out period of 35 ms, which means that if the SCL line is held low for longer than 35 ms, a bus time-out will occur. In this case, the device hardware must Reset its interface. Since the SMBus incorporates the time-out function, it also has a minimum clock speed of 10 kHz, while the I²C bus

has no minimum clock speed. Of course, in real world applications, the I^2C bus will not communicate without a clock signal present. The SMBus specification defines the maximum clock speed at 100 kHz, while the I^2C specification defines the maximum clock speed at 3.4 MHz.

- **Voltage levels** – The SMBus specification's logic high voltage level is set at 2.1V, while the typical I^2C specification's logic high voltage level is dependent on V_{DD}. The I^2C specification's logic high voltage is defined as 70% of V_{DD}, or 0.7*V_{DD}. SMBus specification's logic low voltage level is set to 0.8V, while the I^2C specification's logic low voltage level is dependent on V_{DD}. The I^2C specification's logic low voltage level is defined as 30% of V_{DD}, or 0.3*V_{DD}.
- **Pull-up resistor values and current levels** – Minimum pull-up resistor values are determined by the minimum sink current allowable on the bus. The SMBus specification defines the minimum sink current as 100 µA, while the I^2C specification sets the minimum sink current at 3 mA. Based on these values, the minimum pull-up values for the SMBus are 8.5 kΩ at 3.0V V_{DD}, and 14 kΩ at 5.0V V_{DD}. The minimum pull-up values for the I^2C bus are 1 kΩ at 3.0V V_{DD}, and 1.6 kΩ at 5.0V V_{DD}.

Table 1. I^2C and SMBus Protocol Differences

Specification	I^2C	SMBus
Bus time-out period	Not required	35 ms
Maximum clock frequency	3.4 MHz	100 kHz
Voltage:		
V_{IH}	0.7 * V_{DD}	2.1V
V_{IL}	0.3 * V_{DD}	0.8V
Minimum sink current	3 mA	100 µA
Minimum pull-up value:		
@ 3.0V	1 kΩ	8.5 kΩ
@ 5.0V	1.6 kΩ	14 kΩ

Table of Contents

1. I²C Module Differences

The stand-alone I²C module has several important differences when compared to Microchip's previous Master Synchronous Serial Port (MSSP) module:

- Dedicated Address, Receive, and Transmit Buffers
- Interrupts for Address Match, Transmit Buffer Empty, Receive Buffer Full, Bus Time-out, Data Byte Count, Acknowledge and Not Acknowledge
- Clock Stretching Hardware for:
  - RX buffer full
  - TX buffer empty
  - Address match
  - Acknowledgment
- Hardware Supported Bus Time-out Detection
- Data Byte Counter
- Programmable SDA Hold Time
- Programmable Bus Free Time
- Dedicated I²C Pad Control Registers
- TRIS Register Setting

1.1 Dedicated Address, Receive, and Transmit Buffers

The stand-alone I²C module contains new buffer registers that are dedicated to individual functions.

1.1.1 Address Buffers

The I²C module has two Address Buffer registers, I2CADB0 and I2CADB1, which can be used as a receive buffer in Slave mode, a transmit buffer in Master mode, or both transmit and receive buffers in Multi-Master mode. This differs from the MSSP module in that the MSSP module only used the SSPBUF to receive or transmit an address (or data). The address buffers are enabled via the Address Buffer Disable (ABD) bit. When ABD is clear, the address buffers are enabled; when the ABD is set, the address buffers are disabled.

In 7-bit Slave mode, I2CADB0 is loaded with a matching address and R/$\overline{W}$ information. The I2CADB1 register is not used in 7-bit Slave mode.

In 10-bit Slave mode, I2CADB0 is loaded with the lower eight bits of the matching received address, while the I2CADB1 is loaded with the upper eight bits of the matching received address, including the R/$\overline{W}$ bit.

In 7-bit Master mode, I2CADB1 is used to store a slave address, while I2CADB0 is unused. When the address buffers are enabled (the ABD bit of I2CCON2 = `0`), the address loaded into I2CADB1 is copied into the Transmit Shift register automatically by the hardware. Conversely, when the address buffers are disabled (ABD = `1`), neither I2CADB0 nor I2CADB1 is used, and the slave address is loaded into the I2CTXB register by user software.

In 10-bit Master mode, I2CADB0 is used to store the lower eight bits of the slave address, while I2CADB1 is used to store the upper bits and R/$\overline{W}$ value of the slave address. When the address buffers are enabled (ABD = `0`), the upper byte of the 10-bit address loaded into I2CADB1 is copied into the Transmit Shift register automatically by hardware. Once the master receives the $\overline{ACK}$ from the slave, the lower byte of the 10-bit address loaded into I2CADB0 is copied into the Transmit Shift register automatically by the hardware.

In Multi-Master mode, only 7-bit addresses are used. If the device is addressed as a slave, the received matching slave address is copied into the I2CADB0 register. If the device is communicating as a master, the contents of the I2CADB1 register are copied into the Transmit Shift register to address the slave.

In addition to the two Address Buffer registers, the I^{2}C module dedicates four Slave mode address registers, I2CADR0/1/2/3. These registers hold up to four independent slave addresses.

In 7-bit Slave mode, the first byte received after a Start or Restart condition is compared independently to the values loaded in all four address registers. If the received byte matches a value in one of the registers, it is loaded into the I2CADB0 register, and the Slave Mode Active (SMA) and Address Interrupt Flag (ADRIF) bits are set. If the received byte does not match, the module remains idle.

In 7-bit Slave mode with address masking, the I2CADR0 and I2CADR2 registers are loaded with slave addresses, while the I2CADR1 and I2CADR3 registers are loaded with mask values. The I2CADR1 value is used to mask the slave address in I2CADR0, while the I2CADR3 value is used to mask the slave address in I2CADR2. This allows for full 7-bit address masking for up to two slave addresses.

In 10-bit Slave mode, I2CADR0 and I2CADR1 are combined to form a 10-bit slave address, as are I2CADR2 and I2CADR3. The first byte (10-bit high-address byte) received after a Start or Restart condition is compared to the values stored in I2CADR1 and I2CADR3, which hold the upper byte of the 10-bit address. If there is an address match, the byte is stored in the I2CADB1 register, and the slave issues an $\overline{\text{ACK}}$. The second received byte (10-bit low-address byte) is then compared to the values in I2CADR0 and I2CADR2, which hold the lower byte of the 10-bit address. If the low-address byte matches, the low-address byte is copied into the I2CADB0 register, and the SMA and ADRIF bits are set. If neither the upper nor lower address bytes match the values in the I2CADR0/1 or I2CADR2/3, the module remains idle.

In 10-bit Slave mode with address masking, the I2CADR0 and I2CADR1 registers are combined to hold the 10-bit slave address, while the I2CADR2 and I2CADR3 registers are combined to hold the 10-bit mask values. The received upper and lower address bytes are compared to the masked upper and lower address bytes. If a match occurs, the upper and lower address bytes are loaded into I2CADB1 and I2CADB0, respectively.

It is important to note that in 10-bit Addressing mode, all seven bits of the received upper address byte are compared to the values in I2CADR1 and I2CADR3. The first five most significant bits of the upper address byte are set by the I^{2}C specification as '`11110`', while the remaining three bits act as the upper two most significant bits of the 10-bit address and the R/$\overline{\text{W}}$ bit. The 5-bit '`11110`' format is not enforced by hardware. User software must define these bits as they are compared to the upper five bits of the I2CADR1 and I2CADR3 registers to determine an address match.

1.1.2 Receive Buffer

The stand-alone I^{2}C module has a dedicated receive buffer, I2CRXB, which operates independently from the transmit buffer. This differs from the MSSP module in that the MSSP module had only one buffer, SSPBUF, to handle both transmit and receive functions.

The receive buffer holds one byte of data that is shifted in from the receive shift register. User software can read the byte through the I2CRXB register. When a new byte is received, the Receive Buffer Full (RXBF) Status bit is set. The RXBF bit replaces the Buffer Full (BF) bit used in the MSSP module upon reception of a full byte. Reading I2CRXB will clear the RXBF bit. If the buffer is read while empty (RXBF = `0`), the Receive Read Error (RXRE) bit is set, and the module generates a $\overline{\text{NACK}}$. User software must clear the RXRE bit to resume normal operation. Additionally, setting the Clear Buffer (CLRBF) bit clears both the receive and transmit buffers, as well as the Receive Interrupt Flag (RXIF) bit and Transmit Interrupt Flag (TXIF) bit.

1.1.3 Transmit Buffer

The stand-alone I^{2}C module has a dedicated transmit buffer, I2CTXB, which operates independently from the receive buffer. This differs from the MSSP module in that the MSSP module had only one buffer, SSPBUF, to handle both transmit and receive functions.

The transmit buffer is loaded with an address or data byte that is to be shifted into the transmit shift register and transmitted onto the bus. When the I2CTXB is empty, the Transmit Buffer Empty (TXBE) Status bit is set, allowing user software to load another byte into the buffer. Once the data is transmitted from the I2CTXB register, the TXBE bit is cleared. If user software attempts to load the I2CTXB while it is full, the Transmit Write Error (TXRE) Flag bit is set, a $\overline{\text{NACK}}$ is generated, and the new data is ignored. If the TXRE flag is set, software must clear this bit before attempting to load the buffer again. Additionally, setting the Clear Buffer (CLRBF) bit clears both the transmit and receive buffers, as well as the Transmit Interrupt Flag (TXIF) bit and Receive Interrupt Flag (RXIF) bit.

1.2 Interrupts for Address Match, Transmit Buffer Empty, Receive Buffer Full, Bus Time-Out, Data Byte Count, Acknowledge and Not Acknowledge

The stand-alone I^{2}C module contains additional interrupt features designed to assist with communication functions. In addition to the MSSP module's Start/Restart condition (SCIF), Stop condition (PCIF), Bus Collision (BCLIF), and transmit, receive and acknowledge (SSPIF) interrupts, the stand-alone I^{2}C module adds an Address Match (ADRIF), Transmit Buffer Empty (TXBE), Receive Buffer Full (RXBF), Bus Time-Out (BTOIF), Data Byte Count (CNTIF), Acknowledge Status Time (ACKTIF) and Not Acknowledge Detect (NACKIF).

The stand-alone I^{2}C module incorporates a new register, the I^{2}C Interrupt Flag register (I2CPIR), which handles several I^{2}C related interrupts. Additionally, when any of the Flag bits in I2CPIR become set, the generic I^{2}C Interrupt Flag (I2CIF) is also set. It is important to note that the generic I2CIF bit is read-only and can only be cleared when all bits in the I2CPIR register are clear. The individual interrupts are enabled through the I2CPIE register. If the matching Interrupt Enable bit is set, an interrupt is generated whenever the Interrupt Flag bit is set. If the appropriate Interrupt Enable bit is clear, the Interrupt Flag will still be set when the Interrupt condition occurs. However, no interrupt will be triggered.

The I2CPIR contains the following Interrupt Flag bits:

- CNTIF – Byte Count Interrupt Flag
- ACKTIF – Acknowledge Status Time Interrupt Flag
- WRIF – Data Write Interrupt Flag
- ADRIF – Address Interrupt Flag
- PCIF – Stop Condition Interrupt Flag
- RSCIF – Restart Condition Interrupt Flag
- SCIF – Start Condition Interrupt Flag

The CNTIF becomes set (CNTIF = `1`) when the I2CCNT register value reaches zero, indicating that all bytes in the data frame have been transmitted or received. CNTIF is set after the 9th falling edge of SCL when the I2CCNT = `0`. The MSSP module did not have a byte counter in hardware; byte counts would have to be controlled through user software alone.

The ACKTIF becomes set (ACKTIF = `1`) after the 9th falling edge of SCL for any byte when the device is addressed as a slave in any I^{2}C Slave mode or I^{2}C Multi-Master mode whenever an $\overline{\text{ACK}}$ is detected. The MSSP module used the ACKSTAT bit in the SSPCON2 register to indicate whether an $\overline{\text{ACK}}$ was

received. The ACKSTAT bit did not generate an interrupt, so the bit had to be polled to determine its state.

The WRIF becomes set (WRIF = 1) after the 8th falling edge of SCL when the module receives a data byte. This bit is only active in any I2C Slave mode or I2C Multi-Master mode. Once the data byte is received, the WRIF is set, as are the Receive Buffer Full (RXBF) Status bit, the I2C Receive Interrupt Flag (I2CRXIF) bit and the generic I2CIF bit. The WRIF bit is read/write and must be cleared by user software, while the RXBF, I2CRXIF and I2CIF are read-only, and are only cleared by reading the I2CRXB. In comparison, the MSSP has the Buffer Full (BF) bit in the SSPSTAT register to indicate that the SSPBUF was full. When a byte is received, the BF and SSPIF bits are set on the 8th falling edge of SCL.

The ADRIF becomes set on the 8th falling edge of SCL after the module has received either a matching 7-bit address byte or the matching upper or lower bytes of a 10-bit address. This bit is only active in Slave mode or Multi-Master mode. Upon receiving a matching address byte, the ADRIF and I2CIF bits are set. This differs from the MSSP module that uses the BF bit. When an address or data byte is received, both the BF and SSPIF bits are set, indicating that a byte was received, but did not individualize the difference between an address or a data byte.

The PCIF is set whenever a stop condition is detected on the bus. The PCIF bit is present in both the stand-alone and MSSP modules, and work the same way.

The RSCIF is set upon the detection of a restart condition. The MSSP module used the Start Condition Interrupt Flag (SCIF) for both start and restart conditions.

The SCIF is set upon the detection of a start condition. Both the stand-alone and MSSP modules use this bit, although the MSSP uses it to detect both start and restart conditions.

In addition to the I2CPIR register, the stand-alone module incorporates the I2C Error register (I2CERR). The I2CERR register contains three Interrupt Flag bits that are used to detect bus errors. These bits are read/write and must be cleared by user software. The I2CERR register also includes the Enable bits for these three functions.

The I2CERR register contains the following Interrupt Flag bits:

- BTOIF – Bus Time-Out Interrupt Flag
- BCLIF – Bus Collision Interrupt Flag
- NACKIF – $\overline{\text{NACK}}$ Detect Interrupt Flag

The BTOIF is set when a bus time-out occurs. The bus time-out time frame is controlled by the I2C Bus Time-Out (I2CBTO) register. When a bus time-out event occurs and the module is configured as a slave and is active (SMA = 1), the BTOIF is set, the SMA and CSTR bits are cleared, and the module is immediately Reset. If a bus time-out event occurs and the module is configured as a master and is active (MMA = 1), the BTOIF is set and the module immediately tries to issue a stop condition. In neither case, when the BTOIF becomes set, the generic I2C Error Interrupt Flag (I2CEIF) bit is also set. The I2CEIF bit is read-only, and is cleared by hardware when all error Interrupt Flag bits in the I2CERR register are clear. The MSSP module does not include a hardware-based bus time-out feature. However, software may use an available timer to monitor for a time-out event.

The BCLIF is set whenever a bus collision is detected. A bus collision occurs anytime the SDA input is sampled low while both the SDA and SCL outputs are high. When a bus collision event occurs, the BCLIF and I2CEIF bits are set. Both the stand-alone and MSSP modules use the BCLIF to detect bus collisions.

The NACKIF is set when either the master or the slave is active (SMA = 1 || MMA = 1) and a $\overline{\text{NACK}}$ is detected on the bus. A $\overline{\text{NACK}}$ response occurs on the 9th SCL pulse when the SDA line is released high. When the module is in Master mode, a $\overline{\text{NACK}}$ can be issued when the master has finished receiving data

from the slave, or in the event it did not receive a byte. In Slave mode, the slave issues a $\overline{NACK}$ when it does not receive a matching address, or did not receive the last data byte. A $\overline{NACK}$ can also be automatically sent if any of the following bits is set, which will set both the NACKIF and I2CEIF:

- TXWE – Transmit Write Error Status bit
- RXRE – Receive Read Error Status bit
- TXU – Transmit Underflow Status bit
- RXO – Receive Overflow Status bit

1.3 Clock Stretching Hardware

Clock stretching allows a slave device to hold the clock line, preventing further bus communication. This allows time for the slave device, which may be operating at a slower speed than the master, to process data. The slave device can hold the clock line low until it is ready to continue communications.

Clock stretching in the stand-alone I^2C module can be enabled or disabled via the Clock Stretching Disable (CSD) bit. The CSD bit defaults to a zero value, meaning that clock stretching is enabled. Setting the CSD bit disables clock stretching.

Clock stretching can be performed under several conditions:

- **Buffer read/write operations** – If the RXBF bit is set (receive buffer full) or the TXBE is set (transmit buffer empty) and clock stretching is enabled, the clock will be stretched on the 8^{th} falling SCL edge, allowing to either read the incoming data from the receive buffer or write data to the transmit buffer.
- **Incoming address match interrupt** – When a matching address is received and the Address Interrupt and Hold (ADRIE) Enable bit is set, the CSTR bit is set by hardware, the Address Interrupt Flag (ADRIF) bit is set, the generic I^2C Interrupt Flag (I2CIF) bit is set, and the SCL line is stretched following the 8^{th} falling clock edge. This allows software to read the address from the I2CADB0/1 registers and choose whether to $\overline{ACK}$ or $\overline{NACK}$ the incoming address. When the operation is complete, software clears the CSTR and ADRIF bits. The incoming address match interrupt is similar to the MSSP's Address Hold Enable (AHEN) bit functionality.
- **Data write interrupt** – The Data Write Interrupt and Hold (WRIE) Enable bit enables clock stretching upon the reception of a data byte. When the WRIE is set and a data byte is received, the CSTR bit is set by hardware, the Data Write Interrupt Flag (WRIF) is set, the generic I^2C Interrupt Flag (I2CIF) bit is set, and the SCL line is stretched following the 8^{th} falling clock edge. This allows for software to determine whether to $\overline{ACK}$ or $\overline{NACK}$ the incoming data byte. Clock stretching is released by software clearing the CSTR bit. Software must also clear the WRIF bit to resume normal operation. The data write interrupt is similar to the MSSP's Data Hold Enable (DHEN) bit functionality.
- **Acknowledge status** – The Acknowledge Status Time Interrupt and Hold (ACKTIE) Enable bit is used to enable clock stretching after the $\overline{ACK}$ phase of transmission. The ACKTIE bit enables clock stretching for all address/data transactions – address, write or read. Following the $\overline{ACK}$ sequence (9^{th} falling edge of SCL), the CSTR bit is set by hardware, the Acknowledge Status Time Interrupt Flag (ACKTIF) bit is set, the generic I^2C Interrupt Flag (I2CIF) bit is set, and the SCL line is stretched. Once the slave has completed its processing of the $\overline{ACK}$ sequence, software must clear CSTR to release the SCL line. Software must also clear the ACKTIF bit to resume normal operation. The Acknowledge status feature is similar to the MSSP's Stretch Enable (SEN) bit, which would stretch the clock upon the 9^{th} falling edge of SCL.

1.4 Hardware Supported Bus Time-Out Detection

The stand-alone I²C module provides a hardware supported bus time-out feature to help prevent bus failure. The bus time-out feature also can be used to meet SMBus compatibility, which requires devices on the bus to respond within a certain time frame, typically 35 ms.

The I²C Bus Time-Out Selection (I2CBTO) register selects the time-out source, which is typically a timer or CLC. The time-out source must be configured with the desired time-out time frame. For example, if a Timer2 is selected as the time-out source, Timer2 must be configured to count when the SCL line is low, and if the timer rolls over before the SCL line goes high, the rollover event will Reset the module.

If the module is configured as a slave and a time-out event occurs while the slave is active (SMA = 1), the module is immediately Reset, the generic I²C Error Interrupt Flag (I2CEIF) is set, the Bus Time-Out Interrupt Flag (BTOIF) bit is set, and both the SMA and CSTR bits are cleared. The BTOIF bit must be cleared by software to continue operation.

If the module is configured as a master and a time-out event occurs while the master is active (MMA = 1), the module immediately attempts to generate a stop condition and sets both the BTOIF and I2CEIF Flag bits. The actual generation of the stop condition may be delayed if the SCL line is being stretched by a slave device. The MMA bit will be cleared only after the stop condition has been generated. The BTOIF bit must be cleared by software to continue operation.

1.5 Data Byte Counting

The stand-alone I²C module adds a hardware data byte count feature to specify how many data bytes are in a complete I²C packet. The I²C Byte Count (I2CCNT) register holds the number of data bytes and will decrement every time a data byte is transmitted or received from the module.

Upon the 8th falling edge in Receive mode or the 9th falling edge in Transmit mode of the data byte that caused the I2CCNT to reach a zero value, the Count Interrupt Flag (CNTIF) and generic I2CIF bits are set. The CNTIF bit must be cleared in order to resume normal operations. It is important to note that the I2CCNT register will not decrement past zero.

The I2CCNT register can be read, but since I2CCNT is decremented on the asynchronous SCL signal, a double read is required to ensure a valid read.

The I2CCNT register can be written to, but writing must not occur during the 8th falling edge in Receive mode or during the 9th falling edge in Transmit mode. Writing during either of those bit times may corrupt the I2CCNT register. The I2CCNT register can be safely written to when either the P (stop) bit or CSTR bit is set in Slave mode. In Master mode, I2CCNT can be safely written to when either the MDR bit or BFRE bit is set. If a packet count is larger than 255, the I2CCNT register can be updated mid-message with a new value to prevent the count from prematurely reaching zero.

When in a mode that is receiving data, an acknowledge value is sent back to the transmitter. When the I2CCNT is not at a zero value and receives a data byte, the module will transmit the value of the Acknowledge Data (ACKDT) bit (0 = $\overline{\text{ACK}}$, 1 = $\overline{\text{NACK}}$) during the acknowledge sequence. When the I2CCNT register reaches zero, the module will transmit the value of the Acknowledge End of Count (ACKCNT) bit (0 = $\overline{\text{ACK}}$, 1 = $\overline{\text{NACK}}$).

In Master mode, if the Restart Enable (RSEN) bit is set and I2CCNT = 0, the master will stretch the clock and wait for the S (start) bit to be set before sending a restart condition and the address of the next slave. If both I2CCNT and RSEN = 0, the Master mode state machine will generate a stop condition.

The I2CCNT register can be automatically loaded with a count value when the Auto-Load I^{2}C Count Register (ACNT) Enable bit is set. When ACNT is set, the first received or transmitted byte after the address is automatically loaded into I2CCNT. This first byte following the address should not be included in the count value loaded into I2CCNT. In other words, if the packet includes the address byte, the count value byte and five data bytes, the value of the byte following the address should be 5.

When ACNT = 1, the value of the ACKDT bit is always used for the $\overline{\text{ACK}}$/$\overline{\text{NACK}}$ response to the byte following the address byte. This prevents false $\overline{\text{NACK}}$ responses before the I2CCNT register has been updated with a non-zero value.

1.6 Programmable SDA Hold Time

The stand-alone I^{2}C module has three SDA hold time selections (30, 100, or 300 ns), using the SDA Hold Time Selection (SDAHT) bits. The hold time is the time the SDA signal is held valid after the falling edge of SCL. Additional hold time may be useful for buses with higher bus capacitance. The MSSP module has two SDA hold time selections – 100 or 300 ns.

1.7 Programmable Bus Free Time

When operating in Master mode, the Bus Free (BFRE) bit is used to indicate an idle bus. The master is prevented from issuing a start condition until the BFRE bit is set, indicating an idle bus. This bit helps to prevent collisions from other master devices that may be already active on the bus.

The Bus Free Time (BFRET) bits are used to select the amount of I^{2}C clock cycles used to delay hardware from setting the BFRE bit. The BFRET bits can be used to meet the minimum stop hold time as defined by the I^{2}C specification.

1.8 Dedicated I^{2}C Pad Control Registers

The PIC16 and PIC18 families dedicate specific I/O pins for use with each internal peripheral, including I^{2}C. The stand-alone I^{2}C module has dedicated control registers, RxyI2C, that configure the I^{2}C slew rate, internal pull-ups, and input threshold levels for the dedicated default I^{2}C pins. The Open-Drain Control (ODCON) must be properly configured for each pin. It is important to note that devices that have the Peripheral Pin Select (PPS) feature allow for the default pins to be relocated to other pins. The RxyI2C registers are mapped to the individual device's I^{2}C pins; if the pins are relocated using PPS, the RxyI2C registers cannot be used. Instead, individual Weak Pull-Up (WPU), Slew Rate Control (SLRCON), ODCON and Input Level Control (INLVL) registers must be properly configured. The MSSP module did not include dedicated Pad Control registers. When the MSSP is enabled and the default pins are used, open-drain control and I^{2}C input level control is handled by the hardware automatically. Slew rate control is handled by the SMP bit of the SSPSTAT register, and SMBus input levels are handled by the CKE bit of the SSPSTAT register.

Slew rate control is handled by the SLEW bit of the RxyI2C register. When SLEW = 1, I^{2}C specific slew rate limiting is enabled, while the standard pin slew rate limiting is disabled, and the SLRCONxy bit setting for the pin is ignored. When SLEW = 0, I^{2}C specific slew rate limiting is disabled, and slew rate control is reverted back to the SLRCONxy bit.

I^{2}C internal pull-up selection is handled by the PU bits of the RxyI2C register. The PU bits select either two times or ten times the current of the standard weak pull-up, or allow for the standard weak pull-up control to be enabled/disabled via the WPUxy bit associated with the pin. This feature allows the internal pull-up resistors to be used instead of external resistors. The MSSP module did not have configurable

internal pull-ups, and it was not recommended to use the internal weak pull-ups in place of external resistors.

The TH bits of the RxyI2C register control the input threshold values. These bits allow for SMBus 3.0, SMBus 2.0, I²C specific, or standard I/O input levels to be used, depending on the application requirements. When the standard I/O input level is selected, threshold control reverts back to the INLVLxy bit associated with the pin. The MSSP module's CKE bit controls the SMBus input levels. Instead, the bit only allows for either SMBus 2.0 or I²C specific levels.

1.9 TRIS Register Setting

The stand-alone I²C module requires the TRIS bits controlling the I/O direction of each I²C pin to be initialized clear (TRISxy = 0). The MSSP module required the TRIS bit for each pin to be initialized set (TRISxy = 1). In both cases, the module will take control of the TRIS settings after initialization and once the module is enabled.

1.10 Comparison Between Stand-Alone I²C and MSSP Modules

Table 1-1. Comparison Between Stand-Alone I²C and MSSP Modules

Feature	I²C		MSSP
Buffers/Registers			
Address Buffers	I2CADB0	I2CADB1	SSPADD
	I2CADR0	I2CADR1	
	I2CADR2	I2CADR3	
Transmit Buffer	I2CTXB		SSPBUF
Receive Buffer	I2CRXB		SSPBUF
Byte Count	I2CCNT		None
Bus Time-Out	I2CBTO		None
I²C Module Control	I2CCON0		SSPCON1
	I2CCON1		SSPCON2
	I2CCON2		SSPCON3
I²C Interrupt Control	I2CPIE	PIE	PIE
	I2CPIR	PIR	PIR
I²C Error Control	I2CERR		PIE
			PIR
			SSPCON1
I²C Status	I2CSTAT0	I2CSTAT1	SSPSTAT
	I2CCON1		SSPCON2

Feature	I²C	MSSP
I²C Pad Control	I2CRXY	Limited Hardware Control
Interrupt bits		
Byte Count	CNTIF, I2CIF	None
$\overline{\text{ACK}}$ Detect	ACKTIF, I2CIF	SSPIF
Data Write	WRIF, RXIF, I2CIF	SSPIF
Address Match	ADRIF, I2CIF	None
Start Condition	SCIF, I2CIF	SCIF, SSPIF
Restart Condition	RSCIF, I2CIF	SCIF, SSPIF
Stop Condition	PCIF, I2CIF	PCIF, SSPIF
Transmit Buffer Empty	TXIF, I2CIF	SSPIF
Receive Buffer Full	RXIF, I2CIF	SSPIF
Bus Time-Out	BTOIF, I2CEIF	None
Bus Collision	BCLIF, I2CEIF	BCLIF, SSPIF
$\overline{\text{NACK}}$ Detect	NACKIF, I2CEIF	SSPIF
Status Bits		
Bus Free Indication	BFRE	None
Master Mode Active	MMA	None
Slave Mode Active	SMA	None
Master Data Request	MDR	None
Acknowledge Status	ACKSTAT	ACKSTAT
Acknowledge Time Status	ACKT	ACKTIM
Receive Overflow	RXO	SSPOV
Transmit Underflow	TXU	None
Transmit Write Error	TXWE	WCOL
Transmit Buffer Empty	TXBE	BF
Receive Read Error	RXRE	None
Receive Buffer Full	RXBF	BF
Data/Address Indicator	D	D/$\overline{\text{A}}$
Read/Write Indicator	R	R/$\overline{\text{W}}$
Enable/Action Bits		
Module Enable	EN	SSPEN

Feature	I²C	MSSP
Restart Enable	RSEN	RSEN
Initiate Start Condition	When:	SEN
	ABD = 0: Write to I2CTXB - Hardware sets S	
	ABD = 1: Software sets S	
Initiate Restart Condition	When:	RSEN
	ABD = 0: Write to I2CTXB - Hardware sets S (when RSEN = 1)	
	ABD = 1: Software sets S (when RSEN = 1)	
Initiate Stop Condition	Hardware	PEN
Enable Clock Stretching	CSD	SEN
Clock Stretch Release	CSTR	CKP
General Call Enable	GCEN	GCEN
Enable Acknowledge Sequence	Hardware	ACKEN
Receive Enable	EN	RCEN
Clock Source and Configuration	I2CCLK, FME	MODE, SSPADD
SDA Hold Time	SDAHT	SDAHT
Bus Free Time Configuration	BFRET	None
Bus Time-Out Configuration	I2CBTO	None
Byte Count Configuration	I2CCNT	None
Address Hold Enable	ADRIE	AHEN
Data Hold Enable	WRIE	DHEN
SCL/SDA Pin Control		
Weak Pull-up Control	PU<1:0> bits of I2CRXY	WPU Register
Use of internal pull-ups on bus?	Yes	Not recommended
Input Threshold Control	TH<1:0> bits of I2CRXY	INLVL Register
SMBus Threshold Control	TH<1:0> bits of I2CRXY	CKE bit of SSPSTAT Register
Slew Rate Control	SLEW bit of I2CRXY	SMP bit of SSPSAT Register
Pin Direction Control	TRIS (TRIS = 0)	TRIS (TRIS = 1)
Open-Drain Control	ODCON Register	Hardware Control

2. Conclusion

The 8-bit stand-alone I^2C module contains new hardware additions, such as individual address, transmit, and receive buffers, interrupts, improved clock stretching hardware, bus time-out, data byte count and I^2C pad control. These features remove software overhead for things like tracking the byte count of a packet or a bus time-out. Configurable internal weak pull-ups can be used in place of external pull-up resistors, saving PCB space and reducing component count. Additional clock stretching hardware allows a device in Receive mode more time for data processing.

3. Revision History

Doc.Rev.	Date	Comments
DS40002020A	03/2018	Initial release of this document with the introduction of the 8-bit stand-alone I^2C communications module and the details on migrating from the Master Synchronous Serial Port (MSSP)

The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **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 (FAQ), technical support requests, online discussion groups, Microchip consultant 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

Customer Change Notification Service

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

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://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 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.

Legal Notice

Trademarks

ISBN: 978-1-5224-2808-4

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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: http://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: 45-4450-2828 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: 49-7131-67-3636 **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: 47-7289-7561 **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