INTRODUCTION

This application note introduces the user to the 1-Wire® communication protocol and describes how a 1-Wire device can be interfaced to the PIC® microcontrollers. 1-Wire protocol is a registered trade mark of Maxim/Dallas Semiconductor.

A software stack for the basic, standard speed, 1-Wire master communication is provided with this application note along with an example application.

OVERVIEW OF THE 1-WIRE BUS

The PIC microcontrollers have multiple General Purpose Input/Output (GPIO) pins, and can be easily configured to implement Maxim/Dallas Semiconductor’s 1-Wire protocol.

The 1-Wire protocol allows interaction with many Maxim/Dallas Semiconductor parts, including battery and thermal management devices, memory, iButtons®, etc.

1-Wire devices provide solutions for identification, memory, timekeeping, measurement and control. The 1-Wire data interface is reduced to the absolute minimum (single data line with a ground reference). As most 1-Wire devices provide a relatively small amount of data, the typical data rate of 16 kbps is sufficient for the intended tasks. It is often convenient to use a GPIO pin of an 8-bit or 16-bit microcontroller in a “bit banging” manner to act as the bus master. 1-Wire devices communicate using a single data line and well-defined, time tested protocols.

1-Wire Protocol

- The protocol is called 1-Wire because it uses 1 wire to transfer data. 1-Wire architecture uses a pull-up resistor to pull voltage off the data line at the master side.
- 1-Wire protocol uses CMOS/TTL logic and operates at a supply voltage ranging from 2.8V to 6V.
- Master and slave can be receivers and transmitters, but transfer only one direction at a time (half duplex). The master initiates and controls all 1-Wire operations.
- It is a bit-oriented operation with data read and write, Least Significant bit (LSb) first, and is transferred in time slots.
- The system clock is not required as each part is self-clocked and synchronized by the falling edge of the master.

Prerequisites

The requirements of any 1-Wire bus are:

- The system must be capable of generating an accurate and repeatable 1 μs delay for standard speed and 0.25 μs delay for overdrive speed.
- The communication port must be bidirectional; its output must be open-drain and there should be a weak pull-up on the line.
- The communication operations should not be interrupted while being generated.

Note: Most PIC microcontrollers allow the user to configure any I/O pin to open-drain as it is one of the prerequisites. For recommended pull-up resistance value, refer to the specific slave device data sheet.
OPERATIONS OF THE 1-Wire BUS

The four basic operations of a 1-Wire bus are Reset, Write 0 bit, Write 1 bit and Read bit.

Using these bit operations, one has to derive a byte or a frame of bytes.

The bus master initiates and controls all of the 1-Wire communication. Figure 2 illustrates the 1-Wire communication timing diagram. It is similar to Pulse-Width Modulation (PWM) because, the data is transmitted by wide (logic ‘0’) and narrow (logic ‘1’) pulse widths during data bit time periods or time slots. The timing diagram also contains the recommended time values for robust communication across various line conditions.

Table 1 provides a list of operations with descriptions and also implementation steps; this is for standard speed.

A communication sequence starts when the bus master drives a defined length “Reset” pulse that synchronizes the entire bus. Every slave responds to the “Reset” pulse with a logic-low “Presence” pulse.

To write the data, the master first initiates a time slot by driving the 1-Wire line low, and then, either holds the line low (wide pulse) to transmit a logic ‘0’ or releases the line (short pulse) to allow the bus to return to the logic ‘1’ state. To read the data, the master again initiates a time slot by driving the line with a narrow low pulse. A slave can then either return a logic ‘0’ by turning on its open-drain output and holding the line low to extend the pulse, or return a logic ‘1’ by leaving its open-drain output off to allow the line to recover.

Most 1-Wire devices support two data rates: standard speed of about 15 kbps and overdrive speed of about 111 kbps.

The protocol is self-clocking and tolerates long inter-bit delays, which ensures smooth operation in interrupted software environments.

TABLE 1: 1-Wire® OPERATIONS

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reset</td>
<td>Reset the 1-Wire bus slave devices and get them ready for a command.</td>
<td>Drive bus low, delay 480 μs. Release bus, delay 70 μs. Sample bus: 0 = device(s) present, 1 = no device present Delay 410 μs.</td>
</tr>
<tr>
<td>Write 0 bit</td>
<td>Send ‘0’ bit to the 1-Wire slaves (Write 0 slot time).</td>
<td>Drive bus low, delay 60 μs. Release bus, delay 10 μs.</td>
</tr>
<tr>
<td>Write 1 bit</td>
<td>Send ‘1’ bit to the 1-Wire slaves (Write 1 slot time).</td>
<td>Drive bus low, delay 6 μs. Release bus, delay 64 μs.</td>
</tr>
<tr>
<td>Read bit</td>
<td>Read a bit from the 1-Wire slaves (Read time slot).</td>
<td>Drive bus low, delay 6 μs. Release bus, delay 9 μs. Sample bus to read bit from slave. Delay 55 μs.</td>
</tr>
</tbody>
</table>
FIGURE 2: 1-Wire® TIMING DIAGRAM

- **Reset**
  - 480 μs
  - 550 μs
  - 960 μs

- **Write 0**

- **Write 1**

- **Read**
  - 6 μs
  - 15 μs
  - 60 μs
  - Recovery Time Between Each Slot
  - Master Sample

Legend:
- Master
- Slave
- Register
- Pull-Up
1-Wire APIs FOR PIC MICROCONTROLLERS

Table 2 provides the 1-Wire functions.

**TABLE 2: 1-Wire® API FUNCTIONS**

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>drive_OW_low</td>
<td>This function configures the 1-Wire port pin as an output and drives the</td>
</tr>
<tr>
<td></td>
<td>port pin to LOW.</td>
</tr>
<tr>
<td>drive_OW_high</td>
<td>This function configures the 1-Wire port pin as an output and drives the</td>
</tr>
<tr>
<td></td>
<td>port pin to HIGH.</td>
</tr>
<tr>
<td>read_OW</td>
<td>This function configures the 1-Wire port pin as an input and reads the</td>
</tr>
<tr>
<td></td>
<td>status of the port pin.</td>
</tr>
<tr>
<td>OW_write_byte</td>
<td>This function is used to transmit a byte of data to a slave device.</td>
</tr>
<tr>
<td>OW_read_byte</td>
<td>This function is used for reading a complete byte from the slave device.</td>
</tr>
<tr>
<td>OW_reset_pulse</td>
<td>This function describes the protocol to produce a Reset pulse to a slave</td>
</tr>
<tr>
<td></td>
<td>device and also to detect the presence pulse from the slave device. The</td>
</tr>
<tr>
<td></td>
<td>1-Wire slave device is identified using this function.</td>
</tr>
<tr>
<td>OW_write_bit</td>
<td>This function describes the protocol to write bit information to a slave</td>
</tr>
<tr>
<td></td>
<td>device.</td>
</tr>
<tr>
<td>OW_read_bit</td>
<td>This function describes the protocol to read bit information from a slave</td>
</tr>
<tr>
<td></td>
<td>device.</td>
</tr>
</tbody>
</table>
CONCLUSION
This application note provides an overview of a 1-Wire protocol and also can be used as a building block to develop a sophisticated 1-Wire application using API developed on PIC microcontrollers.

REFERENCES
• http://www.maxim-ic.com/1-Wire
• http://www.maxim-ic.com/appnotes.cfm?appnote_number=126
• http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3711/t/al
APPENDIX A: 1-Wire FUNCTIONS

drive_OW_low
Configures the 1-Wire port pin as an output and drives the port pin to LOW.

Syntax
void drive_OW_low (void)

Parameter
None

Return Values
None

Precondition
None

Side Effects
None

Example
// Driving the 1-Wire bus low
drive_OW_low();

drive_OW_high
Configures the 1-Wire port pin as an output and drives the port pin to HIGH.

Syntax
void drive_OW_high (void)

Parameter
None

Return Values
None

Precondition
None

Side Effects
None

Example
// Driving the 1-Wire bus High
drive_OW_high();
**read_OW**

Configures the 1-Wire port pin as an input and reads the status of the port pin.

**Syntax**

```
unsigned char read_OW (void)
```

**Parameters**

None

**Return Values**

Return the status of OW pin.

**Precondition**

None

**Side Effects**

None

**Example**

```
unsigned char presence_detect;

// Return the status of OW pin.
presence_detect = read_OW(); // Get the presence pulse from 1-Wire slave device.
```
**OW_write_byte**

Transmits 8-bit data to the 1-Wire slave device.

**Syntax**

```c
void OW_write_byte (unsigned char write_data)
```

**Parameters**

Send byte to the 1-Wire slave device.

**Return Values**

None

**Precondition**

None

**Side Effects**

None

**Example**

```c
#define READ_COMMAND_DS2411 0x33

//Send read command to 1-Wire Device DS2411 to get serial number.
OW_write_byte (READ_COMMAND_DS2411);
```
**OW_read_byte**

Reads the 8-bit information from the 1-Wire slave device.

**Syntax**

```c
unsigned char OW_read_byte (void)
```

**Parameters**

None

**Return Values**

Returns the read byte from the slave device.

**Precondition**

None

**Side Effects**

None

**Example**

```c
// To receive 64-bit registration number  ( 8-bit CRC Code, 48-bit Serial
//Number, 8-bit family code) from the 1-Wire slave device.

unsigned char serial_number [8];
unsigned char temp;

for(temp = 0; temp<8; temp++)
    serial_number[temp] = OW_read_byte();
```
**OW_reset_pulse**

Describes 1-Wire protocol to generate Reset pulse to detect the presence of the 1-Wire slave device.

**Syntax**

```c
unsigned char OW_reset_pulse(void)
```

**Parameters**

None

**Return Values**

Return ‘0’ if the slave device presence pulse is detected, return ‘1’ otherwise.

**Precondition**

None

**Side Effects**

None

**Example**

// OW_reset_pulse function return the presence pulse from the slave device

```c
if (!OW_reset_pulse())
    return HIGH; // Slave Device is detected
else
    return LOW; // Slave Device is not detected
```
**OW_write_bit**

Describes 1-Wire protocol to write 1 bit of information to the 1-Wire slave device.

**Syntax**

```c
void OW_write_bit (unsigned char write_bit)
```

**Parameters**

Send one bit to the 1-Wire slave device.

**Return Values**

None

**Precondition**

None

**Side Effects**

None

**Example**

```c
unsigned char loop;

for (loop = 0; loop < 8; loop++)
{
    OW_write_bit(write_data & 0x01);  //Sending LS-bit first
    write_data >>= 1;                // shift the data byte for the next bit to send
}
```
**OW_read_bit**

Describes 1-Wire protocol to read 1 bit of information from the 1-Wire slave device.

**Syntax**

```
unsigned char OW_read_bit (void)
```

**Parameters**

None

**Return Values**

Return the read bit transmitted by a slave device.

**Precondition**

None

**Side Effects**

None

**Example**

```c
unsigned char loop;
unsigned char result = 0;

for (loop = 0; loop < 8; loop++)
{
    result >>= 1;     // shift the result to get it ready for the next bit to receive
    if (OW_read_bit())
        result |= 0x80;   // if result is one, then set MS-bit
}
return (result);
```
APPENDIX B: APPLICATION FLOWCHART

This flowchart illustrates how to use the library functions.

FIGURE B-1: LIBRARY USE FLOWCHART

START

Initialize the USART to Display the Data Read from 1-Wire® Slave Device (DS2411)

Send Reset Pulse using OW_reset_pulse Function to Detect the Slave Device (DS2411)

Send READ_COMMAND (33h) using OW_write_byte Function to get 64-Bit Serial Number

Read 64-Bit Serial Number from DS2411 using OW_read_byte Function

Display 64-Bit Serial Number to Terminal

END

Note: The source code provided with this application note contains an implementation of this flowchart which can be customized to your needs.
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, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SSEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PICtail logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rLAB, Select Mode, Total Endurance, UNI/O, WiperLock 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.

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

© 2008, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

Microchip received ISO/TS-16949:2002 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**

**Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ  85224-6199
Tel:  480-792-7200
Fax:  480-792-7277

Technical Support:
http://support.microchip.com

Web Address:
www.microchip.com

**Atlanta**
Duluth, GA
Tel:  678-957-9614
Fax:  678-957-1455

**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**
Farmington Hills, MI
Tel:  248-538-0071
Fax:  248-538-0075

**Kokomo**
Kokomo, IN
Tel:  765-864-8360
Fax:  765-864-8387

**Los Angeles**
Mission Viejo, CA
Tel:  949-462-9523
Fax:  949-462-9608

**Santa Clara**
Santa Clara, CA
Tel:  408-961-6444
Fax:  408-961-6445

**Toronto**
Mississauga, Ontario, Canada
Tel:  905-673-0699
Fax:  905-673-6509

**ASIA/PACIFIC**

**Asia Pacific Office**
Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong
Tel:  852-2401-1200
Fax:  852-2401-3431

**Australia - Sydney**
Tel:  61-2-9868-6733
Fax:  61-2-9868-6755

**China - Beijing**
Tel:  86-10-8528-2100
Fax:  86-10-8528-2104

**China - Chengdu**
Tel:  86-28-8665-5511
Fax:  86-28-8665-7889

**China - Hong Kong SAR**
Tel:  852-2401-1200
Fax:  852-2401-3431

**China - Nanjing**
Tel:  86-25-8473-2460
Fax:  86-25-8473-2470

**China - Qingdao**
Tel:  86-532-8502-7355
Fax:  86-532-8502-7205

**China - Shanghai**
Tel:  86-21-5407-5533
Fax:  86-21-5407-5066

**China - Shenyang**
Tel:  86-24-2334-2829
Fax:  86-24-2334-2393

**China - Shenzhen**
Tel:  86-755-8203-2660
Fax:  86-755-8203-1760

**China - Wuhan**
Tel:  86-27-5980-5300
Fax:  86-27-5980-5118

**China - Xiamen**
Tel:  86-592-2388138
Fax:  86-592-2388130

**China - Xian**
Tel:  86-29-8833-7252
Fax:  86-29-8833-7256

**China - Zhuhai**
Tel:  86-756-3210040
Fax:  86-756-3210049

**ASIA/PACIFIC**

**India - Bangalore**
Tel:  91-80-4182-8400
Fax:  91-80-4182-8422

**India - New Delhi**
Tel:  91-11-4160-8631
Fax:  91-11-4160-8632

**India - Pune**
Tel:  91-20-2566-1512
Fax:  91-20-2566-1513

**Japan - Yokohama**
Tel:  81-45-741-6166
Fax:  81-45-741-6122

**Korea - Daegu**
Tel:  82-53-744-4301
Fax:  82-53-744-4302

**Korea - Seoul**
Tel:  82-2-554-7200
Fax:  82-2-558-5932 or 82-2-558-5934

**Malaysia - Kuala Lumpur**
Tel:  60-3-6201-9857
Fax:  60-3-6201-9859

**Malaysia - Penang**
Tel:  60-4-227-8870
Fax:  60-4-227-4068

**Philippines - Manila**
Tel:  63-2-634-9065
Fax:  63-2-634-9069

**Singapore**
Tel:  65-6334-8870
Fax:  65-6334-8850

**Taiwan - Hsin Chu**
Tel:  886-3-572-9526
Fax:  886-3-572-6459

**Taiwan - Kaohsiung**
Tel:  886-7-536-4818
Fax:  886-7-536-4803

**Taiwan - Taipei**
Tel:  886-2-2500-6610
Fax:  886-2-2508-0102

**Thailand - Bangkok**
Tel:  66-2-694-1351
Fax:  66-2-694-1350

**EUROPE**

**Austria - Wels**
Tel:  43-7242-2244-39
Fax:  43-7242-2244-393

**Denmark - Copenhagen**
Tel:  45-4450-2828
Fax:  45-4485-2829

**France - Paris**
Tel:  33-1-69-53-63-20
Fax:  33-1-69-30-90-79

**Germany - Munich**
Tel:  49-89-627-144-0
Fax:  49-89-627-144-44

**Italy - Milan**
Tel:  39-0331-742611
Fax:  39-0331-466781

**Netherlands - Drunen**
Tel:  31-416-690399
Fax:  31-416-690340

**Spain - Madrid**
Tel:  34-91-708-08-90
Fax:  34-91-708-08-91

**UK - Wokingham**
Tel:  44-118-921-5869
Fax:  44-118-921-5820

01/02/08