

Keywords: 1-Wire API, iButton, Application Program Interface (API)

APPLICATION NOTE 1100

White Paper 5: Using 1-Wire APIs for Data Sheet Commands

Jun 07, 2002

Abstract: All 1-Wire® device data sheets describe two sets of commands. The first set referred to as ROM Function Commands are used for device identification and selection. The second set is often called Memory Function Commands but may contain other non-memory operations. A ROM Function Command must be completed each time a device is selected to get it ready for a Memory Function Command. The 1-Wire APIs created by Maxim Integrated utilize these commands to do operations with 1-Wire devices. Sometimes it is not always obvious what commands are being named. This document maps the commands presented in the data sheets to the API functions. Where specific API functions are not available, a technique is presented to translate the commands using the generic communications API functions.

Introduction

All 1-Wire device data sheets describe two sets of commands. The first set referred to as *ROM Function Commands* are used for device identification and selection. The second set is often called *Memory Function Commands* but may contain other non-memory operations. A *ROM Function Command* must be completed each time a device is selected to get it ready for a *Memory Function Command*. The 1-Wire APIs created by Maxim Integrated utilize these commands to do operations with 1-Wire devices. Sometimes it is not always obvious what commands are being named. This document maps the commands presented in the data sheets to the API functions. Where specific API functions are not available, a technique is presented to translate the commands using the generic communications API functions.

See application note 155, "[1-Wire Software Resource Guide Device Description](#)" for a detailed description of the various APIs including the abbreviations discussed in this document (PD, TMEX, OWAPI, OWCOM). (*Special terms, commands, or codes are shown in italics for clarity.*)

ROM Function Commands

The *ROM Function Commands* are used to either discover the ROM ID or use the ROM ID to select a device in various modes. The ROM ID is a unique 64-bit number that contains a family code, serialization field, and a cyclic redundancy check (CRC). The 1-Wire master transmits one of these functions after it has issued a 1-Wire reset and received a presence.

The *Read ROM* command reads the ROM ID directly. It can only be used on a 1-Wire network where there is only one device attached. With networks of more than one device, the ROM ID must be discovered with the *Search ROM* command. This search algorithm is discussed in detail in application note 187, "[1-Wire Search Algorithm](#)." The *Conditional Search ROM* command works the same as *Search ROM* except only 1-Wire devices that are in some kind of alarm state respond. This is used to discover only devices that need attention.

The *Skip ROM* command is used to select all devices regardless of ROM ID. This could be used to gang program memory devices provided there is sufficient energy. The *Overdrive Skip* command is similar but it not only selects all devices it also puts those devices at the Overdrive communication rate. This is most often used to move all capable 1-Wire devices to Overdrive speed. After the devices are communicating in Overdrive, the ROM IDs can be discovered using the conventional *Search ROM sequence*.

The *Match ROM* command selects a specific device by broadcasting a selected ROM ID. The *Overdrive Match* is similar but it also switches the device to the Overdrive communication speed. The *Resume Command* is used to reselect the

last device that was selected. This is a shortcut command when repeatedly accessing the same device.

Table 1 maps the APIs to a particular *ROM Function Command*. Note that since the various 1-Wire APIs are designed with the idea of multiple 1-Wire devices on a network, the commands that require or are most useful in a single device network are not supported directly. However any command can be constructed using the basic communication functions as discussed later in *Custom Commands*.

Table 1. ROM Function Commands		
Command	PD	TMEX
Read ROM	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Match ROM	owAccess	TMAccess
Search ROM	owFirst, owNext	TOMFirst, TMNext
Conditional Search ROM	owFirst, owNext	TMFirstAlarm, TMNextAlarm
Skip ROM	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Overdrive Skip*	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Overdrive Match	owOverdriveAccess	TMOverAccess
Resume Command**	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Command	OWAPI	OWCOM
Read ROM	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Match ROM	(package com.dalsemi.onewire.adapter) DSPortAdapter.select	DSPortAdapter.select
Search ROM	(package com.dalsemi.onewire.adapter) DSPortAdapter.getFirstDeviceContainer, DSPortAdapter.getNextDeviceContainer	DSPortAdapter.getFirstDeviceContainer, DSPortAdapter.getNextDeviceContainer
Conditional Search ROM	(package com.dalsemi.onewire.adapter) DSPortAdapter.setSearchOnlyAlarmingDevices (then same as Search ROM)	DSPortAdapter.setSearchOnlyAlarmingDevices (then same as Search ROM)
Skip ROM	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Overdrive Skip*	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.
Overdrive Match*	(package com.dalsemi.onewire.container) OneWireContainer.setSpeed OneWireContainer.doSpeed	OneWireContainer.setSpeed OneWireContainer.doSpeed
Resume Command**	No predefined API, see Custom Commands.	No predefined API, see Custom Commands.

*Note: Only applies to 1-Wire devices that support Overdrive communication speed.

**Note: Only applies to 1-Wire devices that support the Resume Command.

Memory Function Commands

The *Memory Function Commands* vary slightly from one device type to another. However, their primary objective is the same, which is to read and write the memory areas of the device. To deal with these diverse command structures, the 1-Wire APIs were constructed to abstract out these differences. For example, a generic write memory API may use a Write Scratchpad, Read Scratchpad, and Copy Scratchpad sequence or it may use an EPROM Write Memory sequence. To the API user, it looks the same.

There are three levels of memory commands in most of the APIs. The first allows reading and writing to the memory without any structure (raw). The second uses a packet structure called the Universal Data Packet (UDP). The third type combines multiple UDP structures into a file structure. See application note 114, "[1-Wire File Structure](#)" for a description of the UDP and file structure. **Table 2** maps the APIs to the three types of memory operations.

Table 2. Abstract Memory Functions		
Command	PD	TMEX
Write Raw	owWrite	TMProgramBlock (EPROM only) (see Custom Commands)
Read Raw	owRead	TMProgramBlock (EPROM only) (see Custom Commands)
Write UDP	owWritePagePacket	TMWritePacket
Read UDP	owReadPagePacket	TMReadPacket
Write File	owCreateFile owWriteFile	TMCreateFile TMWriteFile
Read File	owOpenFile owReadFile	TMOpenFile TMReadFile
Command	OWAPI	OWCOM
Write Raw	(package com.dalsemi.onewire.container) MemoryBank.write	MemoryBank.write
Read Raw	(package com.dalsemi.onewire.container) MemoryBank.read	MemoryBank.read
Write UDP	(package com.dalsemi.onewire.container) PagedMemoryBank.writePagePacket	PagedMemoryBank.writePagePacket
Read UDP	(package com.dalsemi.onewire.container) PagedMemoryBank.readPagePacket	PagedMemoryBank.readPagePacket
Write File	(package com.dalsemi.onewire.application.file) OWFileOutputStream.write	OWFileOutputStream.write
Read File	(package com.dalsemi.onewire.application.file) OWFileInputStream.read	OWFileInputStream.read

Some of the commands that are included under *Memory Function Commands* in data sheets are actually custom device commands. See the following section for a guide on how to deal with these commands.

Custom Commands

Almost all of the custom commands can be derived by first selecting the device with a *Match ROM* equivalent API and then send a bidirectional block of data to the 1-Wire network. The block is constructed by putting in the write commands

that are required into the block and putting in FF (hex) bytes into the block that are reads from the 1-Wire device. For example, the DS1922L has a memory-mapped real-time clock (RTC) register that can be accessed with the *Read Memory* command. Figure 1 is taken from the DS1922L's data sheet.

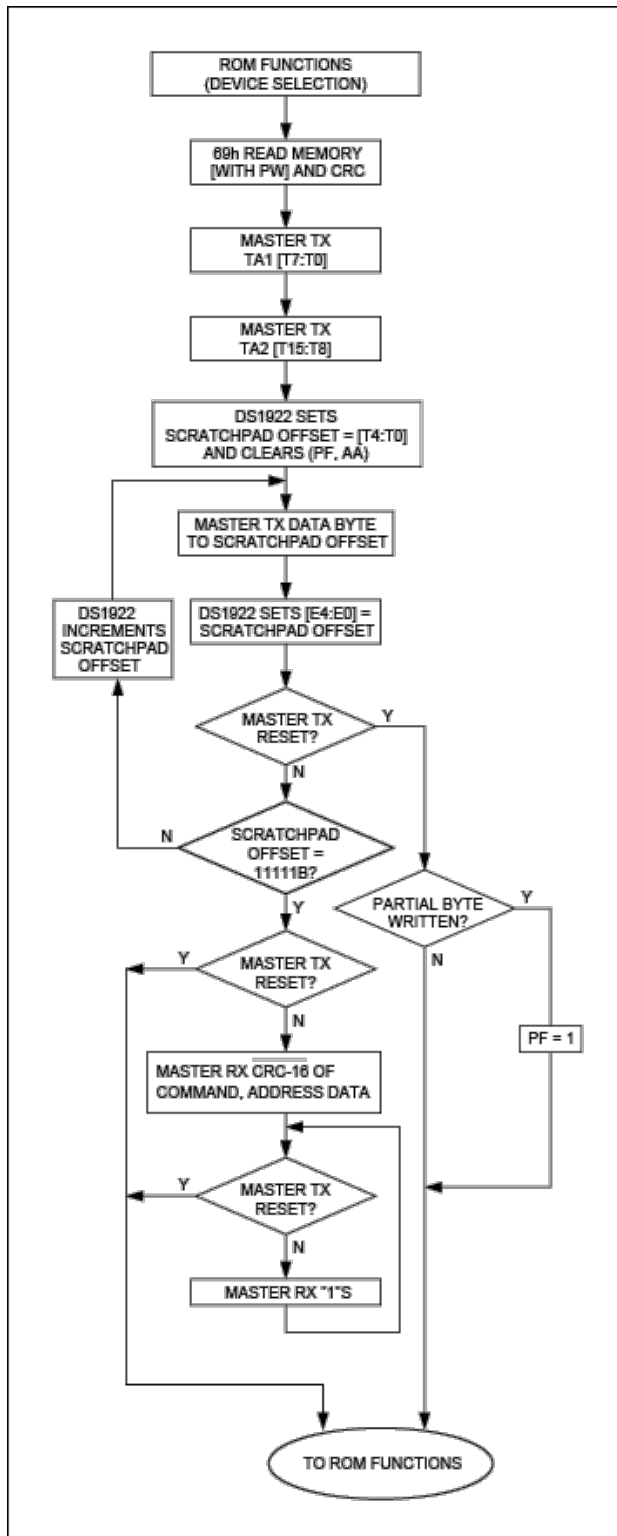


Figure 1. DS1922L read memory (with PW) and CRC flow (data sheet).

As the DS1922L data sheet specifies, the RTC register is five bytes long starting at address 0219(hex). The memory command flow as seen in Figure 1 starts after the device has been selected with a *ROM Function Command* like *Match ROM*.

Table 3 lists the sixteen bytes that make up a block of bidirectional data to be sent to the 1-Wire bus based on the flow chart. While this data could be sent a byte at a time, it is often more efficient to create a block and send it all at once.

Table 3. Read RTC Block		
Block Offset	Byte Value (hex)	Description
0	69	Master Tx Read Memory with (PW) and CRC command
1	19	Master Tx TA1 (address, least significant byte, T7:T0)
2	02	Master Tx TA2 (address, most significant byte, T15:T8)
3	FF	Master Tx Password Byte 0 (assume not set)
4	FF	Master Tx Password Byte 1
5	FF	Master Tx Password Byte 2
6	FF	Master Tx Password Byte 3
7	FF	Master Tx Password Byte 4
8	FF	Master Tx Password Byte 5
9	FF	Master Tx Password Byte 6
10	FF	Master Tx Password Byte 7
11	FF	Master Rx byte 0 of RTC (address 0219h)
12	FF	Master Rx byte 1 of RTC (address 021Ah)
13	FF	Master Rx byte 2 of RTC (address 021Bh)
14	FF	Master Rx byte 3 of RTC (address 021Ch)
15	FF	Master Rx byte 4 of RTC (address 021Dh)

Figure 2 is a 'C' example written for the 1-Wire Public Domain (PD) API that uses the block outlined in Table 3 to read the RTC of the DS1922L.

Figure 2. PD Example Reading RTC

```

unsigned char datablock[] = { 0xF0,0x19,0x02,0xFF,0xFF,0xFF,0xFF,
    0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF };
int portnum=0;

// select the current device (Match ROM)
if (owAccess(portnum))
{
    // send the read memory command and address, receive the RTC value
    if (owBlock(portnum, 1, datablock, 16))
    {
        // RTC is now in bytes 11-15 of datablock
        ;...
    }
}

```

Table 4 shows the bidirectional block commands for each of the APIs. For completeness, the single byte and bit commands are also included.

Table 4. Generic 1-Wire I/O Functions		
Command	PD	TMEX
block (bidirectional)	owBlock	TMBlockStream
byte (bidirectional)	owTouchByte	TMTouchByte
bit (bidirectional)	owTouchBit	TMTouchBit
read byte	owReadByte	TMTouchByte(data = FF hex)
write byte	owWriteByte	TMTouchByte(data to write)
reset	owTouchReset	TMTouchReset
Command	OWAPI	OWCOM
block (bidirectional)	(package com.dalsemi.onewire.adapter) DSPortAdapter.dataBlock	DSPortAdapter.dataBlock
byte (bidirectional)	(package com.dalsemi.onewire.adapter) DSPortAdapter.dataBlock (single byte block)	DSPortAdapter.dataBlock (single byte block)
bit (bidirectional)	Not available	Not available
read byte	(package com.dalsemi.onewire.adapter) DSPortAdapter.getBytes	DSPortAdapter.getBytes
write byte	(package com.dalsemi.onewire.adapter) DSPortAdapter.putByte	DSPortAdapter.putByte
1-Wire reset + presence detect	(package com.dalsemi.onewire.adapter) DSPortAdapter.reset	DSPortAdapter.reset

By looking at the data sheet for each 1-Wire device type, it is possible to do any of the functions by constructing the appropriate block and using the generic 1-Wire I/O functions. Some of the 1-Wire devices require special power delivery constraints or program pulses that are addressed by the special API functions in Table 5.

Table 5. Special 1-Wire Power Functions		
Command	PD	TMEX
EPROM programming pulse	owProgramPulse	TMProgramPulse
Power delivery (strong pullup) after bit	owReadBitPower (read bit only)	TMOneWireLevel (prime for next bit) TMTouchBit
Power delivery (strong pullup) after byte	owWriteBytePower (write byte only)	TMOneWireLevel (prime for next byte) TMTouchByte
Command	OWAPI	OWCOM
EPROM programming pulse	(package com.dalsemi.onewire.adapter) DSPortAdapter.startProgramPulse	DSPortAdapter.startProgramPulse
Power delivery (strong pullup) after bit	(package com.dalsemi.onewire.adapter) DSPortAdapter.startPowerDelivery (prime for next bit) DSPortAdapter.putBit / DSPortAdapter.getBit	DSPortAdapter.startPowerDelivery (prime for next bit) DSPortAdapter.putBit / DSPortAdapter.getBit
Power delivery (strong pullup) after byte	(package com.dalsemi.onewire.adapter) DSPortAdapter.startPowerDelivery (prime for next byte) DSPortAdapter.putByte / DSPortAdapter.getByte	DSPortAdapter.startPowerDelivery (prime for next byte) DSPortAdapter.putByte / DSPortAdapter.getByte

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Related Parts		
DS1822	Econo 1-Wire Digital Thermometer	Free Samples
DS1822-PAR	Econo Parasite-Power Digital Thermometer	
DS18B20	Programmable Resolution 1-Wire Digital Thermometer	Free Samples
DS18B20-PAR	1-Wire Parasite-Power Digital Thermometer	
DS18S20	1-Wire Parasite-Power Digital Thermometer	Free Samples
DS18S20-PAR	Parasite-Power Digital Thermometer	
DS1904	iButton RTC	Free Samples
DS1920	iButton Temperature Logger	
DS1921G	Thermochron iButton Device	
DS1921K	Thermochron iButton Starter Kit	
DS1922L	iButton Temperature Loggers with 8KB Data-Log Memory	
DS1963S	iButton Monetary Device with SHA-1 Function	

DS1971	iButton 256-Bit EEPROM	
DS1973	iButton 4Kb EEPROM	Free Samples
DS1982	iButton 1Kb Add-Only	Free Samples
DS1985	iButton 16Kb Add-Only	Free Samples
DS1990A	iButton Serial Number	Free Samples
DS1992	iButton 1Kb/4Kb Memory	Free Samples
DS1993	iButton 1Kb/4Kb Memory	Free Samples
DS1995	iButton 16Kb Memory	Free Samples
DS1996	iButton 64Kb Memory	Free Samples
DS2401	Silicon Serial Number	Free Samples
DS2406	Dual Addressable Switch Plus 1Kb Memory	Free Samples
DS2417	1-Wire Time Chip With Interrupt	Free Samples
DS2431	1024-Bit 1-Wire EEPROM	Free Samples
DS2432	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	Free Samples
DS2438	Smart Battery Monitor	Free Samples
DS2450	1-Wire Quad A/D Converter	
DS2502	1Kb Add-Only Memory	Free Samples
DS2505	16Kb Add-Only Memory	Free Samples
MAX31820	1-Wire Ambient Temperature Sensor	Free Samples
MAX31820PAR	1-Wire Parasite-Power, Ambient Temperature Sensor	
MAX31826	1-Wire Digital Temperature Sensor with 1Kb Lockable EEPROM	Free Samples

More Information

For Technical Support: <http://www.maximintegrated.com/support>

For Samples: <http://www.maximintegrated.com/samples>

Other Questions and Comments: <http://www.maximintegrated.com/contact>

Application Note 1100: <http://www.maximintegrated.com/an1100>

APPLICATION NOTE 1100, AN1100, AN 1100, APP1100, Appnote1100, Appnote 1100

© 2013 Maxim Integrated Products, Inc.

Additional Legal Notices: <http://www.maximintegrated.com/legal>