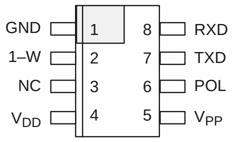
DALLAS SEMICONDUCTOR

DS2480
Serial 1–Wire^{TM} Line Driver

FEATURES

- Universal, common–ground serial port to 1–Wire^{TM} line driver for MicroLAN^{TM} applications
- Works with all iButtons and MicroLAN–compatible 1–Wire slave devices
- Communicates at regular and Overdrive 1–Wire speed and serial port data rates of 9600 (default), 19200, 57600 and 115200 bps
- Supports 12V EPROM programming and stiff 5V pull–up for Crypto iButton, sensors and EEPROM
- Self–calibrating time base with ±5% tolerance for serial and 1–Wire communication
- Slew rate controlled 1–Wire pull–down and active pull–up to accommodate long lines and reduce radiation
- User–selectable RXD/TXD polarity minimizes component count when interfacing to 5V based RS232 systems or directly to UARTs
- Programmable 1–Wire timing and driver characteristics accommodate a wide range of MicroLAN configurations at regular speed
- Smart protocol combines data and control information without requiring extra pins
- Compatible to optical, IR and RF to RS232 converters
- Low cost 8–pin SOIC surface mount package
- Operates over 4.5V to 5.5V from –40°C to +85°C

PIN ASSIGNMENT



8–PIN SOIC
(150 MIL)

PIN DESCRIPTION

GND	Ground
1–W	1–Wire Input/Output
NC	No Connection
V_{DD}	4.5 to 5.5 Volts
V_{PP}	Optional EPROM Programming Voltage
POL	RXD/TXD Polarity Select
TXD	Serial Data from UART
RXD	Serial Data to UART

ORDERING INFORMATION

DS2480S	8–pin SOIC

DESCRIPTION

The DS2480 is a serial port to 1–Wire interface chip that supports standard and Overdrive speeds. It connects directly to UARTs and 5V RS232 systems. Interfacing to RS232C (± 12V levels) requires a passive clamping circuit and one 5V to ± 12V level translator. Internal timers relieve the host of the burden of generating the time–critical 1–Wire communication waveforms. In contrast to the DS9097(E) where a full character must be sent by the host for each 1–Wire time slot, the DS2480 can translate each character into eight 1–Wire time slots thereby increasing the data throughput significantly. In addition, the DS2480 can be set to communicate at four different data rates including 115.2 kbps, 57.6 kbps and 19.2 kbps with 9.6 kbps being the power–on default.

Command codes received from the host's crystal controlled UART serve as a reference to continuously calibrate the on–chip timing generator. The DS2480 uses a unique protocol that merges data and control information without requiring control pins. This approach maintains compatibility to off–the–shelf serial to wireless converters allowing easy realization of 1–Wire media jumpers. The various control functions of the DS2480 are optimized for MicroLAN 1–Wire networks and support the special needs of all current 1–Wire devices including the Crypto iButton, EPROM–based Add–Only Memories, EEPROM devices and 1–Wire Thermometers.

DETAILED PIN DESCRIPTION

PIN	SYMBOL	DESCRIPTION
1	GND	**Ground Pin:** common ground reference and ground return for 1–Wire bus
2	1–W	**1–Wire Input/Output Pin:** 1–Wire bus with slew–rate–controlled pull–down, active pull–up, ability to switch in V_{PP} to program EPROM, and ability to switch in V_{DD} through a low–impedance path to program EEPROM, perform a temperature conversion or operate the Crypto iButton.
3	NC	**No Connection Pin.**
4	V_{DD}	**Power Input Pin:** power supply for the chip and 1–Wire pull–up voltage. 5V ± 10%; may be derived from 12V V_{PP} with an external voltage regulator.
5	V_{PP}	**EPROM Programming Voltage:** 12V supply input for EPROM programming; if EPROM programming is not required, this pin must be tied to V_{DD}. V_{PP} must come up before V_{DD}.
6	POL	**RXD/TXD Polarity Select:** RXD/TXD polarity select; tied to GND for RS232 (12V or 5V) connection, tied to V_{DD} for direct connection to UART chip.
7	TXD	**Serial Data from UART:** data input from host (inverted or true); maximum voltage swing –0.3V to V_{DD} + 0.3V; for logic thresholds see DC specifications.
8	RXD	**Serial Data to UART:** signal output to host; push–pull driver with CMOS compatible levels; for true ±12V RS232 systems an external level–translator must be provided.

OVERVIEW

The DS2480 directly interfaces a 5V serial communication port with its lines TXD (transmit) and RXD (receive) to a 1–Wire bus. In addition the device performs a speed conversion allowing the data rate at the communication port to be different from the 1–Wire date rate. Several parameters relating to the 1–Wire port and its timing as well as the communication speed at both the port and the 1–Wire bus are configurable. The circuit to achieve these functions is outlined in Figure 1, Block Diagram.

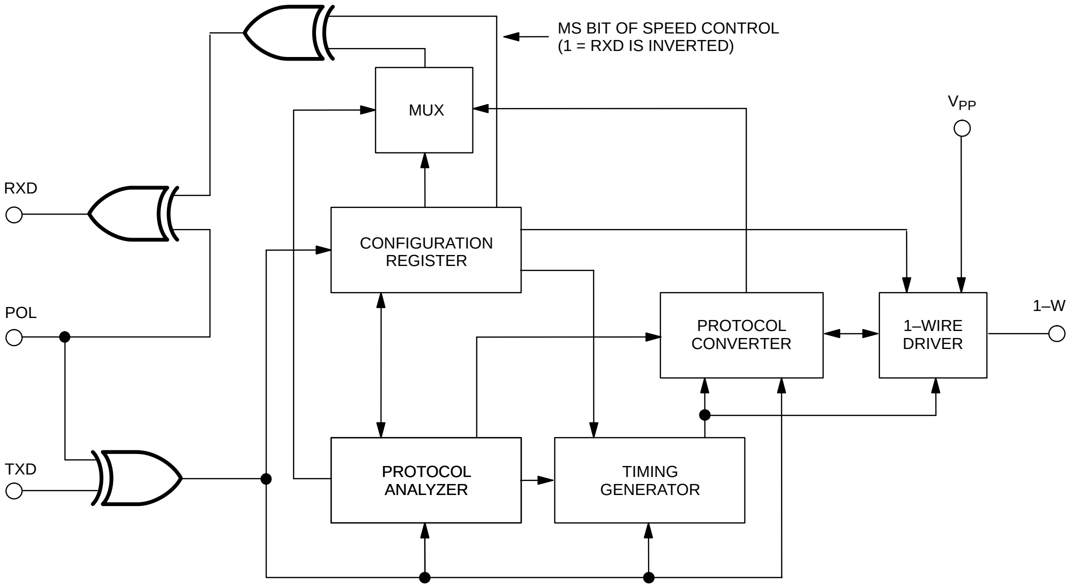
The device gets its input data from the serial communication port of the host computer through pin TXD. For compatibility with active–high as well as active–low systems, the incoming signal can be inverted by means of the polarity input POL. The polarity chosen by hard–wiring the logic level of this pin is also valid for the output pin RXD. If for minimizing the interface hardware an asymmetry between RXD and TXD is desired, this can be achieved by setting the most significant bit of the Speed Control parameter to a 1 (see Configuration Parameter Value Codes). With the MS bit of the speed control set to 1, the polarity at TXD is still selected by the logic level at POL, but the polarity at RXD will be the opposite of what the logic level at POL specifies.

As data enters the core of the DS2480's logic circuitry, it is analyzed to separate data and command bytes and to calibrate the device's timing generator. The timing generator controls all speed relations of the communication interface and the 1–Wire bus as well as the wave forms on the 1–Wire bus.

Command bytes either affect the configuration setting or generate certain wave forms on the 1–Wire bus. Data bytes are simply translated by the protocol converter into the appropriate 1–Wire activities. Each data byte generates a return byte from the 1–Wire bus, that is communicated back to the host through the RXD pin as soon as the activity on the 1–Wire bus is completed.

The 1–Wire driver shapes the slopes of the 1–Wire wave forms, applies programming pulses or strong pull–up to 5 volts and reads the 1–Wire bus using a non–TTL threshold to maximize the noise margin for best performance on large 1–Wire MicroLAN networks.

DS2480 BLOCK DIAGRAM Figure 1



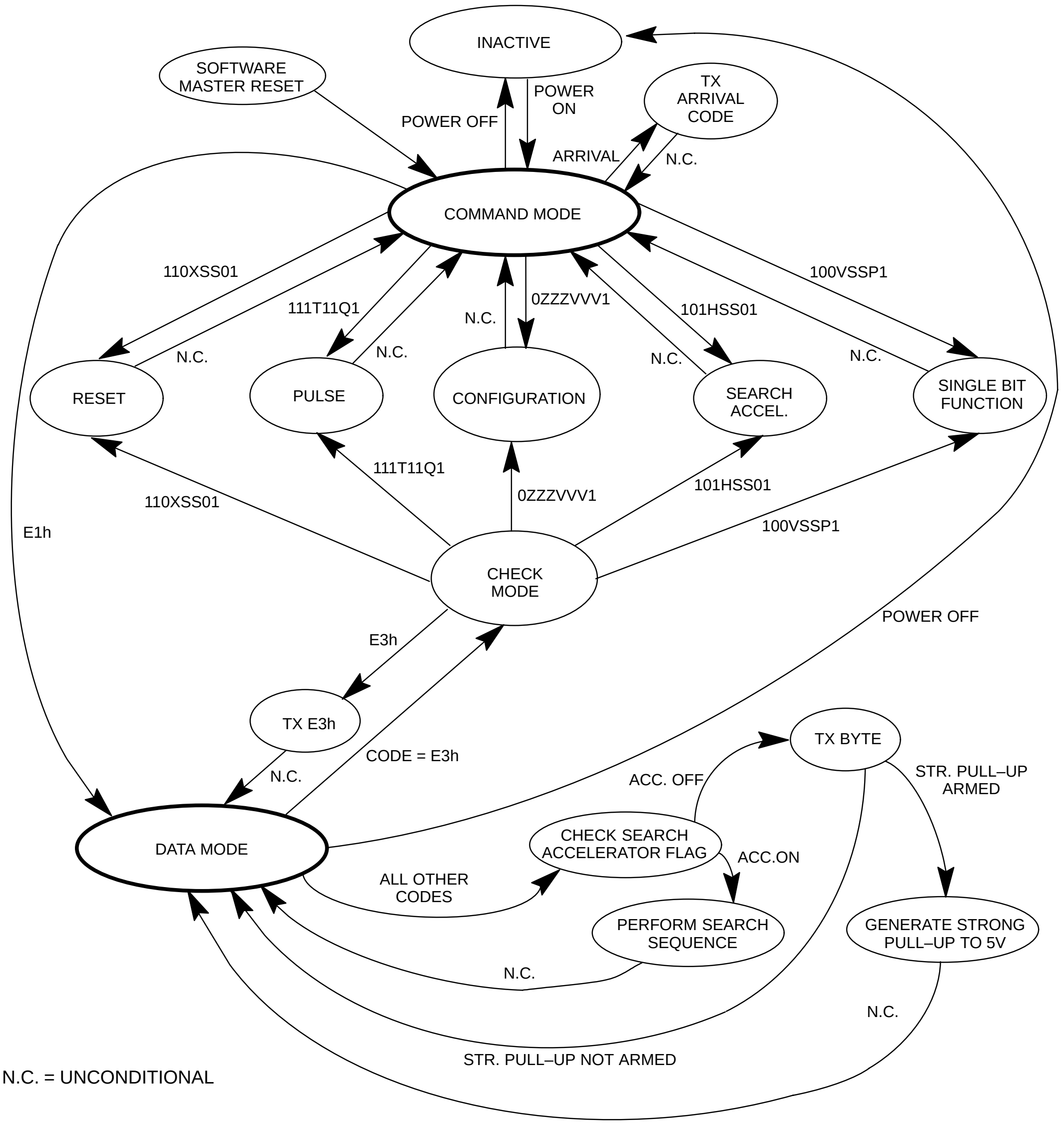
DEVICE OPERATION

The DS2480 can be described as a complex state machine with two static and several dynamic states. Two device–internal flags as well as functions assigned to certain bit positions in the command codes determine the behavior of the chip, as shown in the state transition diagram (Figure 2). The DS2480 requires and generates a communication protocol of 8 data bits per character, 1 stop bit and no parity. It is permissible to use two stop bits on the TXD line. However, the DS2480 will only assert a single stop bit on RXD.

When powering up, the DS2480 performs a master reset cycle and enters the **Command Mode**, which is one of the two static states. The device now expects to receive one 1–Wire reset command on the TXD line sent by the host at a data rate of 9600 bits per second (see section Communication Commands for details). This command byte is required solely for calibration of the timing generator the DS2480 and is not translated into any activity on the 1–Wire bus. After this first command byte the device is ready to receive and execute any command as described later in this document.

A master reset cycle can also be generated by means of software. This may be necessary if the host for any reason has lost synchronization with the device. The DS2480 will perform a master reset cycle equivalent to the power–on reset if it detects start polarity in place of the stop bit. The host has several options to generate this condition. These include making the UART generate a break signal, sending a NULL character at a data rate of 4800 bps and sending any character with parity enabled and selecting space polarity for the parity bit. As with the power–on reset, the DS2480 requires a 1–Wire reset command sent by the host at a data rate of 9600 bps for calibration.

STATE TRANSITION DIAGRAM Figure 2



N.C. = UNCONDITIONAL

LEGEND:		
	V	BINARY VALUE (TYPE OF WRITE TIME SLOT)
	SS	1–WIRE SPEED SELECTION CODE
	P	IF LOGIC 1, GENERATES STRONG PULL–UP TO 5V IMMEDIATELY FOLLOWING THE TIME SLOT
	T	TYPE OF PULSE; 0 = STRONG PULL–UP (5V), 1 = PROGRAMMING PULSE (12V)
	Q	1 = ARM STRONG PULL–UP AFTER EVERY BYTE; 0 = DISARM
	H	SEARCH ACCELERATOR CONTROL; 1 = ACCELERATOR ON, 0 = ACCELERATOR OFF
	ZZZ	CONFIGURATION PARAMETER CODE (WRITE), 000 = READ CONFIGURATION PARAMETER
	VVV	CONFIGURATION PARAMETER VALUE CODE (WRITE), CONFIGURATION PARAMETER CODE (READ)
	X	DON’T CARE

After the DS2480 has reached the command mode, the host can send commands such as 1–Wire Reset, Pulse, Configuration, Search Accelerator and Single Bit functions or switch over to the second static state called **Data Mode**. In data mode the DS2480 simply converts bytes it receives at the TXD pin into their equivalent 1–Wire wave forms and reports the results back to the host through the RXD pin. If the Search Accelerator is on, each byte seen at TXD will generate a 12–bit sequence on the 1–Wire bus (see section Search Accelerator for details). If the Strong Pull–up to 5V is enabled (see Pulse command) each byte on the 1–Wire bus will be followed by a pause of predefined duration where the bus is pulled to 5V via a low impedance transistor in the 1–Wire driver circuit.

While being in the Data Mode the DS2480 checks each byte received from the host for the reserved code that is used to switch back to Command Mode. To be able to write any possible code (including the reserved one) to the 1–Wire bus, the transition to the Command Mode is as follows: After having received the code for switching to Command Mode, the device temporarily enters the **Check Mode** where it waits for the next byte. If both bytes are the same, the byte is sent **once** to the 1–Wire bus and the device returns to the Data Mode. If the second byte is different from the reserved code, it will be executed as command and the device finally enters the Command Mode. As a consequence, if the reserved code that normally switches to Command Mode is to be written to the 1–Wire bus, this code byte must be sent twice (duplicated). This detail must be considered carefully when developing software drivers for the DS2480.

After having completed a memory function with a device on the 1–Wire bus it is recommended to issue a Reset Pulse. This means that the DS2480 has to be switched to Command mode. The host then sends the appropriate command code and continues performing other tasks. If during this time a device arrives at the 1–Wire bus it will generate a presence pulse. The DS2480 will recognize this unsolicited presence pulse and notify the host by sending a byte such as XXXXXX01b. The Xs represent undefined bit values. The fact that the host receives the byte unsolicited together with the pattern 01b in the least significant two bits marks the bus arrival. If the DS2480 is left in Data Mode after completing a memory function command it will not report any bus arrival to the host.

COMMAND CODE OVERVIEW

The DS2480 is controlled by a variety of commands. All command codes are 8 bits long. The most significant bit of each command code distinguishes between communication and configuration commands. Configuration commands access the configuration registers. They can write or read any of the configurable parameters. Communication commands use data of the configuration register in order to generate activity on the 1–Wire bus and/or (dis)arm the strong pull–up after every byte or (de)activate the Search Accelerator without generating activity on the 1–Wire bus. Details on the command codes are included in the State Transition diagram (Figure 2). A full explanation is given in the subsequent sections Communication Commands and Configuration Commands.

In addition to the command codes explained in the subsequent sections the DS2480 understands the following reserved command codes:

E1h	switch to Data Mode
E3h	switch to Command Mode
F1h	pulse termination

Except for these reserved commands, the Search Accelerator control and the first byte after power–on reset or master reset cycle, every legal command byte generates a response byte. The pulse termination code triggers the response byte of the terminated pulse command. Illegal command bytes do not generate a command response byte.

Once the device is switched back from Data Mode to Command Mode one must not repeat the E3h command while the Command Mode is still active.

COMMUNICATION COMMANDS

The DS2480 supports four communication function commands: Reset, Single Bit, Pulse, and Search Accelerator control. Details on the assignment of each bit of the command codes are shown in Table 1. The corresponding command response bytes are detailed in Table 2. The Reset, Search Accelerator Control and Single Bit commands include bits to select the 1–Wire communication speed (regular, flexible regular, Overdrive). Even if a command does not generate activity on the 1–Wire bus, these bits are latched inside the device and will take effect immediately.

COMMUNICATION COMMAND CODES Table 1

FUNCTION	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3, BIT 2	BIT 1	BIT 0
Single Bit	1	0	0	0 = write 0 1 = write 1	00 reg. speed 01 flex. speed 10 OD. speed 11 reg. speed	**See Text**	1
Search Accelerator Control	1	0	1	0 = accelerator off 1 = accelerator on **See Text**	00 reg. speed 01 flex. speed 10 OD. speed 11 reg. speed	0	1
Reset	1	1	0	(don’t care)	00 reg. speed 01 flex. speed 10 OD. speed 11 reg. speed	0	1
Pulse	1	1	1	0 = 5V strong pull–up 1 = 12V prog. pulse	11 pulse	**See Text**	1

COMMUNICATION COMMAND RESPONSE Table 2

FUNCTION	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Single Bit	1	0	0	same as sent			1–Wire read back, both bits same value	
Reset	1	1	**See Text**	0	1	0	00 = 1–Wire shorted 01 = presence pulse 10 = alarming presence pulse 11 = no presence pulse	
Pulse	1	1	1	same as sent			undefined	

(The Search Accelerator Control command does not generate a response byte.)

Reset

The Reset command must be used to begin all 1–Wire communication. The speed selection included in the command code immediately takes effect. The response byte includes a code for the reaction on the 1–Wire bus (bits 0 and 1) and a code for the chip revision (bits 2 to 4). If bit 5 of the response byte reads ‘1’, a programming voltage is present on the V$_{PP}$ pin, indicating that one may try programming EPROM devices.

Single Bit

The Single Bit command is used to generate a single time slot on the 1–Wire bus at the speed indicated by bits 2 and 3. The type of the time slot (write zero or write one) is determined by the logic value of bit 4. A read data time slot is identical to the write one time slot. Bits 0 and 1 of the response byte transmitted by the DS2480 at the end of the time slot reveal the value found on the 1–Wire bus when reading.

For a time slot without a subsequent strong pull–up, bit 1 of the command must be set to 0. For a time slot immediately followed by a strong pull–up bit 1 must be set to 1. As soon as the strong pull–up is over, the device will send a second response byte, code EFh (read 1) or ECh (read 0), depending on the value found on the 1–Wire bus when reading. The strong pull–up directly following the single bit is used in conjunction with the Crypto iButton.

Search Accelerator Control

The Search Accelerator Control command is used to set or reset the Search Accelerator control flag. Bit 4 of the command code contains the state to which the accelerator control flag is to be set. If the flag is set to a 1 (on) the device translates every byte received in data mode into a 12–bit sequence on the 1–Wire bus. For details on how the Search Accelerator works please refer to the section Search Accelerator Operation. Before activating the Search Accelerator, one must make sure that the strong pull–up after every byte is disarmed (see Pulse Command). The Search Accelerator command does not generate a command response byte.

Although the Search Accelerator Control command itself does not generate any 1–Wire activity, it can be used to select the communication speed on the 1–Wire bus. The speed selection (if different from the previous setting, e.g., from a Reset command) will take effect immediately.

Pulse

The Pulse command serves several functions that are selected by the contents of bit 1 and bit 4 of the command code. The main functions are generating a strong pull–up to 5V and generating 12V programming pulses for EPROM devices (if the 12V are available at the V_{PP} pin). The secondary function of the pulse command is arming and disarming a strong pull–up after every subsequent byte in data mode. The arm/disarm function is controlled by bit 1 of the command code. Bit 4 determines whether the device will generate a strong pull–up to 5V or a 12V programming pulse. The table below summarizes these options.

BIT 4	BIT 1	FUNCTION
0	0	strong pull–up to 5V and disarm
1	0	12V Programming Pulse and disarm
0	1	strong pull–up to 5V and arm
1	1	12V Programming Pulse and arm

The strong pull–up to 5V is required to program EEPROM devices or to operate special function devices that require a higher current for a limited time after having received a “go and convert” command. Therefore and because it significantly reduces the effective data throughput on the 1–Wire bus, the strong pull–up is disarmed most of the time. Although arming or disarming is simultaneously possible while generating a programming pulse, this is not recommended since it is likely to destroy the DS2480 if non–EPROM devices are connected to the 1–Wire bus.

The duration of the strong pull–up or programming pulse is determined by configuration parameters and ranges from a few microseconds up to unlimited (see section Configuration Commands). However, unlimited duration is not allowed in conjunction with arming the strong–pull–up after every byte. As long as the DS2480 is in Command Mode the host may terminate a strong pull–up or programming pulse prematurely at any time by sending the command code F1h.

The response byte is generated as soon as the strong pull–up or programming pulse is over (either because the predefined time has elapsed or due to termination). The response byte mainly returns the command code as sent by the host, but the two least significant bits are undefined.

If the strong pull–up is armed and the device is in Data Mode, the end of the strong pull–up will be signaled as code F6h if the most significant bit of the preceding data byte on the 1–Wire bus was a 1 and 76h otherwise. The host will see this response byte in addition to the response on the data byte sent (see also section Wave Forms later in this document).

SEARCH ACCELERATOR INTRODUCTION

The Search Accelerator is a logic block inside the DS2480 that allows using the Search ROM function very efficiently under modern operating systems such as Windows and Windows 95/NT. Without the DS2480 all 1–Wire port adapters have to involve the computer’s CPU for every single time slot or pulse to be generated on the 1–Wire bus.

Under DOS, accessing peripherals such as the UART or parallel port is very straight forward and therefore fast. Under Windows the situation is different and it may take several milliseconds or more to get the first time slot generated on the 1–Wire bus. Every subsequent time slot will be generated in much less time, since the computer simply sends out (“streams”) a long chain of bytes. This works reasonably well when reading or writing large blocks of data.

Searching the 1–Wire bus to identify all ROM IDs of the devices connected, however, requires reading two bits, making a decision and then writing a bit. This procedure is to be repeated 64 times to identify and address a single device. With the overhead of modern operating systems this fairly simple process takes a lot of time, reducing the discovery rate of devices on the 1–Wire bus from a typical value of 40 to 50 per second under DOS to less than 10 under Windows. To solve this problem the Search Accelerator was developed.

The Search Accelerator receives from the host information on the preferred path to chose during the execution of the Search ROM function as one contiguous chain of bytes and then translates it into the appropriate time slots on the 1–Wire bus. In addition, the Search Accelerator reports back to the host the ROM ID of the device actually addressed and the bit positions in which conflicts were found. (If the ROM ID of one device has a 0 in a bit position where another device has a 1, this is called a "conflict" on the electrical level and "discrepancy" on the logical level. See the Book of DS19xx iButton Standards for a more detailed discussion of the Search ROM). This helps the host to select the preferred path for the next Search ROM activity.

Since the ROM ID of all MicroLAN compatible devices is 64 bits long and a conflict may occur in any of these bits, the total length of data reported to the host is 128 bits or 16 bytes. To avoid data overrun (if the CPU sends data faster than it can be processed) the protocol for the Search Accelerator operation was defined so that one has to send as many bytes as one will receive. This way the CPU sends 16 bytes for each path and the UART guarantees the correct data timing and frees the CPU for other tasks while the DS2480 performs a Search ROM function.

SEARCH ACCELERATOR OPERATION

After the Search Accelerator is activated and the data mode is selected, the host must send 16 bytes to complete a single Search ROM pass on the 1–Wire bus. These bytes are constructed as follows:

first byte

7	6	5	4	3	2	1	0
r_3	x_3	r_2	x_2	r_1	x_1	r_0	x_0

et cetera

16th byte

7	6	5	4	3	2	1	0
r_{63}	x_{63}	r_{62}	x_{62}	r_{61}	x_{61}	r_{60}	x_{60}

In this scheme, the index (values from 0 to 63, "n") designates the position of the bit in the ROM ID of a MicroLAN compatible device. The character "x" marks bits that act as filler and do not require a specific value (don't care bits). The character "r" marks the path to go at that particular bit in case of a conflict during the execution of the ROM Search.

For each bit position n (values from 0 to 63) the DS2480 will generate three time slots on the 1–Wire bus. These are referenced as:

b0 for the first time slot (read data)
b1 for the second time slot (read data) and
b2 for the third time slot (write data).

The type of time slot b2 (write 1 or write 0) is determined by the DS2480 as follows:
b2 = r_n if conflict (as chosen by the host)
= b_0 if no conflict (there is no alternative)
= 1 if error (there is no response)

The response the host will receive during a complete pass through a Search ROM function using the Search Accelerator consists of 16 bytes as follows:

first byte

7	6	5	4	3	2	1	0
r'_3	d_3	r'_2	d_2	r'_1	d_1	r'_0	d_0

et cetera

16th byte

7	6	5	4	3	2	1	0
r'_{63}	d_{63}	r'_{62}	d_{62}	r'_{61}	d_{61}	r'_{60}	d_{60}

As before, the index (values from 0 to 63, "n") designates the position of the bit in the ROM ID of a MicroLAN compatible device. The character "d" marks the discrepancy flag in that particular bit position. The discrepancy flag will be 1 if there is a conflict or no response in that particular bit position and 0 otherwise. The character "r' " marks the actually chosen path at that particular bit position. The chosen path is identical to b2 for the particular bit position of the ROM ID.

To perform a Search ROM sequence one starts with all bits r_n being 0s. In case of a bus error, all subsequent response bits r'_n are 1's until the Search Accelerator is deactivated. Thus, if r'_{63} and d_{63} are both 1, an error has occurred during the search procedure and the last sequence has to be repeated. Otherwise r'_n (n = 0 ... 63) is the ROM code of the device that has been found and addressed.

For the next Search ROM sequence one re–uses the previous set r_n (n = 0 ... 63) but sets r_m to 1 with "m" being the index number of the highest discrepancy flag that is 1 and sets all r_i to 0 with i > m. This process is repeated until the highest discrepancy occurs in the same bit position for two consecutive passes.

The table below shows an example for the communication between host and DS2480 to perform one pass through the Search ROM function using the Search Accelerator. After a device has been identified and addressed, a (not specified here) memory function is executed and finally a reset pulse is generated. This example assumes that the DS2480 was in Command Mode and that regular 1–Wire speed is used.

Search Accelerator Usage Example

Action Sequence	Host TX	Host RX
Generate Reset Pulse	**C1**	**C9**
Set Data Mode	**E1**	**(nothing)**
Search ROM command	F0	(as sent)
Set Command Mode	E3	(nothing)
Search Accelerator On	**B1**	**(nothing)**
Set Data Mode	**E1**	**(nothing)**
Send 16 bytes	data	response
Set Command Mode	E3	(nothing)
Search Accelerator Off	**A1**	**(nothing)**
Set Data Mode	**E1**	**(nothing)**
Do Memory Function		
Set Command Mode	E3	(nothing)
Generate Reset Pulse	**C1**	**C9**

CONFIGURATION COMMANDS

The DS2480 is designed to be configurable for the varying requirements of its application. When the device powers up and/or performs a master reset cycle, the hard–wired default configuration settings take effect. These settings will work on a short 1–Wire bus and assume regular 1–Wire communication speed. To change these default settings and to verify the current settings, the logic of the DS2480 supports configuration commands. A summary of the available configuration parameters, their default settings at regular and Overdrive speed and their applicability is shown in Table 3.

Parameters not related to the communication speed on the 1–Wire bus specify the duration of the 12V programming pulse, the duration of the strong pull–up to 5V and the baud rate on the interface that connects the DS2480 to the host. The remaining three parameters are used to modify the 1–Wire communication wave forms if one selects "Flexible Speed" (see "Communication Commands" for speed selection).

Flexible speed is implemented to improve the performance of large MicroLAN Networks. This is accomplished by:

- limiting the slew rate on falling edges (e. g., at the beginning of time slots, to reduce ringing),
- extending the Write 1 low time (allows the current flow through the network to end slowly, to prevent voltage spikes from inductive kickback),
- delaying the time point when reading a bit from the 1–Wire bus (gives the network more time to stabilize, to get a higher voltage margin) and
- adding extra recovery time between Write 0 time slots (allows more energy transfer through the network, to replenish the parasite power supply of the devices on the bus).

The latter two functions are controlled by a single parameter. Taking advantage of flexible speed requires changing one or more of these parameters from their default values. Otherwise the waveforms will be identical to those at regular speed.

Each configuration parameter is identified by its 3–bit parameter code and can be programmed for one of a maximum 8 different values using a 3–bit value code. A matrix of parameter codes and value codes with the associated physical values in shown in Table 4.

CONFIGURATION COMMAND OVERVIEW Table 3

Parameter Description	Par. Code	Configurable at			Default	
		Regular	Flexible	Overdrive	Reg./Flex.	Overdrive
Pull–Down Slew Rate Control	001		√		15 V/µs	15 V/µs
Programming Pulse Duration	010	√	√	√	512 µs	512 µs
Strong Pull–up Duration	011	√	√	√	524 ms	524 ms
Write 1 low time	100		√		8 µs	1 µs
Data Sample Offset and Write 0 Recovery Time	101		√		3 µs 3 µs	1 µs 3 µs
RS232 Baud Rate	111	√	√	√	9.6k bits/s	9.6k bits/s

The numbers given for parameter 001 (Pull–Down Slew Rate Control) are nominal values. They may vary as specified in the Electrical Characteristics section and are almost independent of the load on the 1–Wire bus. Information on how to select the optimum value of this parameter is given in section “Controlled Edges”.

For the parameters 010 (Programming Pulse Duration) and 011 (Strong Pull–Up Duration) one may select indefinite duration. This value, however, should only be selected if one is not going to switch the device to Data Mode. As long as the device stays in Command Mode, any pulse function (programming or strong pull–up) that uses one of these parameters can be terminated by sending the command code F1h. Termination is not possible if the device is in Data Mode.

Parameter 111 (RS232 Baud Rate) has two functions. It selects the baud rate and allows inversion of the signal at the RXD pin. Using one of the value codes 100 to 111 will set the polarity at RXD to the opposite of what is defined by the logic level at the POL pin (asymmetry bit, see Figure 1). This may reduce the component count in some applications of the device. Note that when changing the baud rate, the DS2480 will send the command response byte at the new data rate.

A short explanation on the use of parameters 100 (Write 1 low time) and 101 (Data Sample Offset/Write 0 Recovery Time) is given in the section “Timing Diagrams” later in this document. The parameter code 110 is reserved for future extensions; one should not change the value code from its default setting.

CONFIGURATION PARAMETER VALUE CODES Table 4

Parameter Code	Value Codes								Unit
	000	001	010	011	100	101	110	111	
001 (PDSRC)	15	2.2	1.65	1.37	1.1	0.83	0.7	0.55	V/µs
010 (PPD)	32	64	128	256	512	1024	2048	∞	µs
011 (SPUD)	16.4	65.5	131	262	524	1048	2096	∞	ms
100 (W1LT)	8	9	10	11	12	13	14	15	µs
101 (DSO/W0RT)	3	4	5	6	7	8	9	10	µs
111 (RBR)	9.6	19.2	57.6	115.2	9.6	19.2	57.6	115.2	kbits/s

The syntax of configuration commands is very simple. Each 8–bit code word contains a 3–bit parameter code to specify the parameter and the 3–bit value code to be selected. Bit 7 of the command code is set to 0 and bit 0 is always a 1. To read the value code of a parameter, one writes all zeros for the parameter code and puts the parameter code in place of the parameter value code. Table 5 shows the details.

The configuration command response byte is similar to the command byte itself. Bit 0 of the response byte is always 0. When writing a parameter, the upper 7 bits are the echo of the command code. When reading a parameter, the current value code is returned in bit positions 1 to 3 with the upper 4 bits being the same as sent (see Table 6).

CONFIGURATION COMMAND CODES Table 5

FUNCTION	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Write Parameter	0	parameter code			parameter value code			1
Read Parameter	0	0	0	0	parameter code			1

CONFIGURATION COMMAND RESPONSE BYTE Table 6

FUNCTION	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
Write Parameter	0	same as sent			same as sent			0
Read Parameter	0	same as sent			parameter value code			0

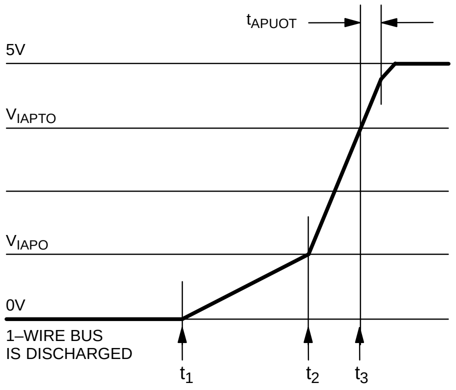
CONTROLLED EDGES

One of the tasks of the DS2480 is to actively shape the edges of the 1–Wire communication waveforms. This speeds up the recharging of the 1–Wire bus (rising edges) and reduces ringing of long lines (falling edges). The circuitry for shaping rising edges is always on. The slew rate of falling edges is actively controlled only at flexible speed and requires the parameter for slew rate control being different from its power–on default value.

All Rising Edges

The active pull–up of the rising edges reduces the rise time on the 1–Wire bus significantly compared to a simple resistive pull–up. Figure 4 shows how the DS2480 is involved in shaping a rising edge.

ACTIVE PULL–UP Figure 4

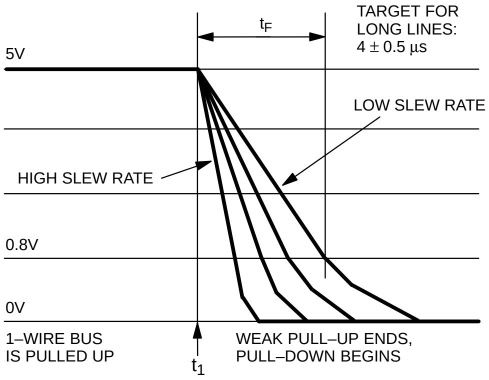


The circuit operates as follows: At t_1 the pull–down (induced by the DS2480 or a device on the bus) ends. From this point on the 1–Wire bus is pulled high by the weak pull–up current I_{WEAKPU} provided by the DS2480. The slope is determined by the load on the bus and the value of the pull–up current. At t_2 the voltage crosses the threshold voltage V_{IAPO}. Now the DS2480 switches over from the weak pull–up current I_{WEAKPU} to the higher current I_{ACTPU}. As a consequence, the voltage on the bus now rises faster. As the voltage on the bus crosses the threshold V_{IAPTO} at t_3, a timer is started. As long as this timer is on (t_{APUOT}), the I_{ACTPU} current will continue to flow. After the timer is expired, the DS2480 will switch back to the weak pull–up current.

Falling Edges (DS2480–initiated)

Whenever the DS2480 begins pulling the 1–Wire bus low to initiate a time slot, for example, it first turns off the weak pull–up current I_{WEAKPU}. Then, at regular and Overdrive speed it will generate a falling edge at a slew rate of typically 15V/µs. This value is acceptable for short 1–Wire busses and adequate for communication at Overdrive speed. For MicroLAN networks of more than roughly 30 meters length one should always use flexible speed. One of the parameters that is adjustable at flexible speed is the slew rate of DS2480–initiated falling edges. The effect of the slew rate control is shown in Figure 5.

SLEW RATE CONTROL Figure 5



As extensive tests have shown, MicroLAN networks at a length of up to 300 meters will perform best if the fall time t_F is in the range of 4 ± 0.5 µs. This translates into a slew rate of approximately 1V/µs. This slew rate is typically achieved by setting the configuration parameter 001 (Pull–Down Slew Rate Control) to a value of 100 (see Table 4). If the actual measured fall time is longer than the target value, one should use a value code of 011 or lower. If the fall time is shorter, one should use a value code of 101 or higher.

Once determined, the value code for the Pull–Down Slew Rate Control parameter should be stored in the host and always be loaded into the DS2480 after a power–on or master reset cycle.

TIMING DIAGRAMS

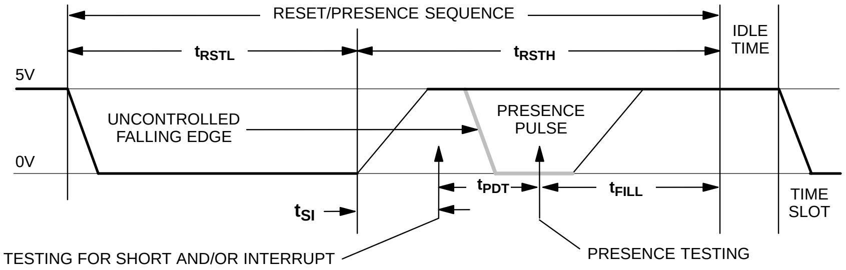
This section explains the wave forms generated by the DS2480 on the 1–Wire bus in detail. First the communication wave forms such as the Reset/Presence Detect Sequence and the time slots are discussed. After that follows a detailed description of the pulse function under various conditions. The wave forms as generated by the DS2480 may deviate slightly from specifications found in the “Book of DS19xx iButton Standards” or in data sheets of 1–Wire slave devices. However, at a closer look one will find that all of the timing requirements are met.

1–WIRE COMMUNICATION WAVE FORMS

One of the major features of the DS2480 is that it relieves the host from generating the timing of the 1–Wire signals and sampling the 1–Wire bus at the appropriate times. How this is done for the reset/presence detect sequence is shown in Figure 6a. This sequence is composed of four timing segments: the reset low time t_{RSTL}, the short/interrupt sampling offset t_{SI}, the presence detect sampling offset t_{PDT} and a delay time t_{FILL}. The timing segments t_{SI}, t_{PDT} and t_{FILL} comprise the reset high time t_{RSTH} where 1–Wire slave devices assert their presence or interrupt pulse. During this time the DS2480 pulls the 1–Wire bus high with its weak pull–up current.

The values of all timing segments for all 1–Wire speed options are shown in the table. Since the reset/presence sequence is slow compared to the time slots, the values for regular and flexible speed are the same. Except for the falling edge of the presence pulse all edges are controlled by the DS2480. The shape of the uncontrolled falling edge is determined by the capacitance of the 1–Wire bus and the number, speed and sink capability of the slave devices connected.

Reset/Presence Detect Figure 6a



Speed	t_{RSTL}	t_{SI}	t_{PDT}	t_{FILL}	t_{RSTH}
Regular	512 µs	8 µs	64 µs	512 µs	584 µs
Overdrive	64 µs	2 µs	8 µs	64 µs	74 µs
Flexible	512 µs	8 µs	64 µs	512 µs	584 µs

After having received the command code for generating a reset/presence sequence, the DS2480 pulls the 1–Wire bus low for t_{RSTL} and then lets it go back to 5V. The DS2480 will now wait for the short/interrupt sampling offset t_{SI} to expire and then test the voltage on the 1–Wire bus to determine if there is a short or an interrupt signal. If there is no short or interrupt (as shown in the picture), the DS2480 will wait for t_{PDT} and test the voltage on the 1–Wire bus for a presence pulse. Regardless of the result of the presence test, the DS2480 will then wait for t_{FILL} to expire and then send the command response byte to the host.

If the test for interrupt or short reveals a logic 0, the DS2480 will wait for 4096 µs and then test the 1–Wire bus again. If a logic 0 is detected, the 1–Wire bus is shorted and a command response byte with the code for SHORT will be sent immediately. If a logic 1 is detected, the device will wait for t_{FILL} to expire after which it will send the command response byte with the code for an alarming presence pulse. No additional testing for a presence pulse will be done. The DS2480 will perform the short/interrupt testing as described also at Overdrive speed, although interrupt signaling is only defined for regular speed.

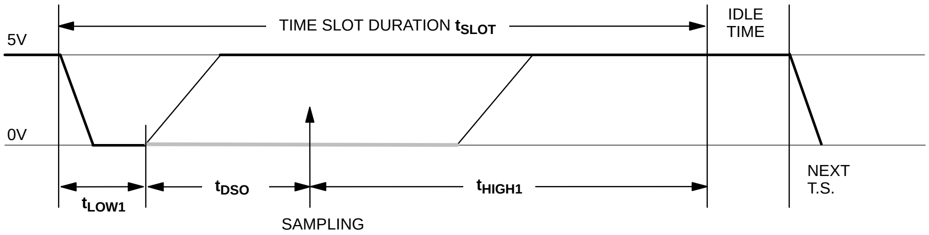
The idle time following the Reset/Presence Detect sequence depends on the serial communication speed and the host's response time.

A Write 1 and Read Data time slot is comprised of the segments t_{LOW1}, t_{DSO} and t_{HIGH}. During write 1 time slots, after the write 1 low time t_{LOW1} is over, the DS2480 waits for the duration of the data sample offset and then samples the voltage at the 1–Wire bus to read the response. After this, the waiting time t_{HIGH1} must expire before the time slot is complete. A Write 0 time slot only consists of the two segments t_{LOW0} and t_{REC0}.

If the network is large or heavily loaded, one should select flexible speed and extend t_{LOW1} to more than 8 µs to allow the 1–Wire bus to completely discharge. Since a large or heavily loaded network needs more time to recharge, it is also recommended to delay sampling the bus for reading. A higher value for t_{DSO} will increase the voltage margin and also provide extra energy to the slave devices when generating a long series of write 0 time slots. However, the total of t_{LOW1} + t_{DSO} should not exceed 22 µs*. Otherwise the slave device responding may have stopped pulling the bus low when transmitting a logic 0.

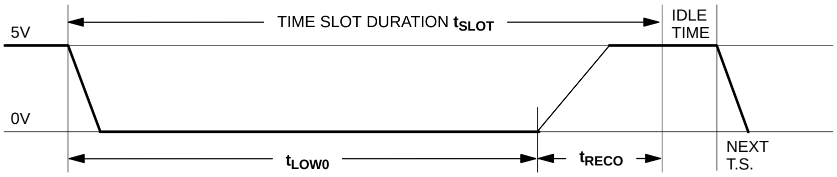
The idle time between time slots within a byte or during a 12–bit sequence while the Search Accelerator is on is zero. Between bytes, 12–bit search sequences and single bits the idle time depends on the RS232 data rate and the host's response time. The response byte is sent to the host as soon as the last time slot of a byte, 12–bit sequence or the command is completed.

WRITE 1 AND READ DATA TIME SLOT Figure 6b



Speed	t_{LOW1}	t_{DSO}	t_{HIGH1}	t_{SLOT}*
Regular	8 µs	3 µs	49 µs	60 µs
Overdrive	1 µs	1 µs	8 µs	10 µs
Flexible	8 to 15 µs	3 to 10 µs	49 µs	60 to 74 µs

WRITE 0 TIME SLOT Figure 6c



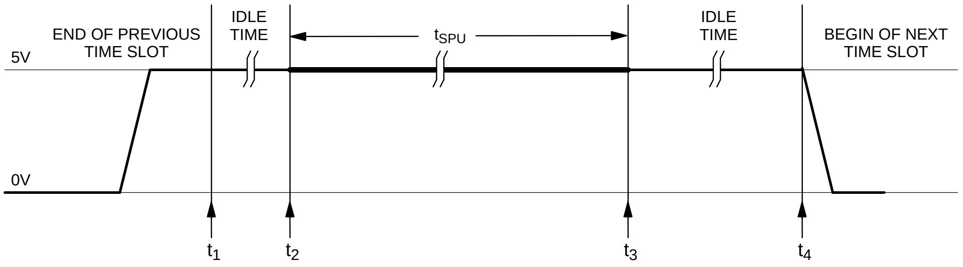
Speed	t_{LOW0}*	t_{REC0}	t_{SLOT}*
Regular	57 µs	3 µs	60 µs
Overdrive	7 µs	3 µs	10 µs
Flexible	57 µs	3 to 10 µs	60 to 67 µs

*In a 5V environment (± 1V, full temperature range) the tolerance of the internal time base of 1–Wire slave devices is much narrower than what it is when operated at the minimum voltage of 2.8V. Therefore, the timing generated by the DS2480 is in compliance with the requirements of all MicroLAN–compatible 1–Wire devices.

PULSE WAVE FORMS, DISARMED

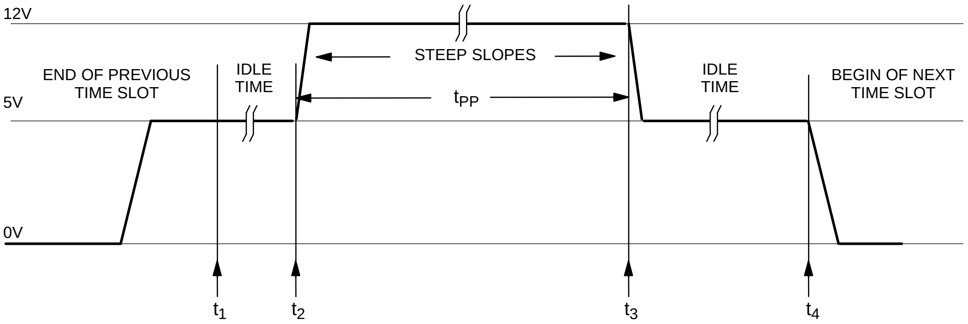
The pulse command can be used to generate a strong pull–up to 5V and a 12V programming pulse, respectively. The duration of the pulse is predefined if the parameter value code of parameter 010 (Programming Pulse Duration) and parameter 011 (Strong Pull–Up Duration), has a value from 000 to 110 (see Table 4). Figures 7a and 7b show the timing of a pulse with predefined duration, which should be considered the normal case. If an infinite duration is chosen (parameter value code 111), the host must terminate the pulse command, as shown in Figures 7c and 7d. All versions of Figure 7 assume that bit 1 of the pulse command is 0, i.e., disarmed mode. See section Communication Commands, Pulse, for more details on possibilities of the pulse command.

STRONG PULL–UP TO 5V, PREDEFINED DURATION Figure 7a



The processing of a pulse command is essentially the same, regardless if a strong pull–up or a programming pulse is requested. At t_1 the host starts sending the pulse command byte. At t_2 the DS2480 has received the command and immediately generates the pulse. The pulse ends at t_3 and the DS2480 sends out the command response byte to inform the host that the command is completed. The idle time between t_1 and t_2 is determined by the time to transmit the command byte at the selected baud rate. The idle time between t_3 and t_4 is comprised of the time to transmit the response byte, plus the response time of the host plus the time to transmit the command and/or data to generate the next time slot.

12V PROGRAMMING PULSE, PREDEFINED DURATION Figure 7b



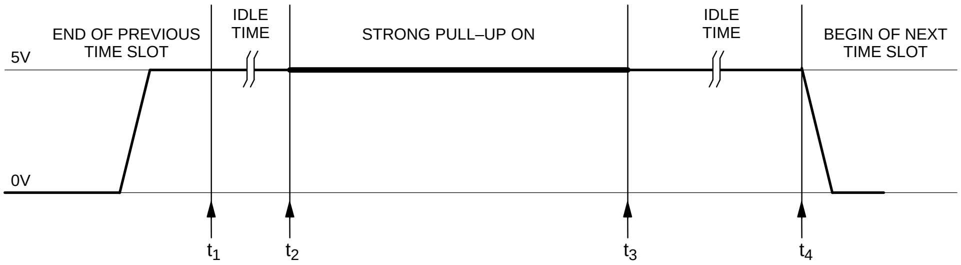
A correct programming pulse can only be generated if the 12V programming voltage is available at the Vpp pin of the DS2480. The slew rate of the rising and falling edge of the programming pulse ("steep slopes") is not actively controlled by DS2480.

For EPROM programming, only a single slave device should be connected to the 1–Wire bus and the cable must to be short, not to exceed a few meters. One should not attempt generating a programming pulse with a non–EPROM device on the bus; this may damage the device as well as the DS2480.

Certain applications may require a duration for a strong pull–up or programming pulse that cannot be realized using one of the predefined values. Selecting infinite duration allows the host to generate pulses of any length. As a consequence, however, the host becomes responsible to actively control the duration of the pulse. Failing to do so may require a power–on reset or master reset cycle of the DS2480. For this reason, infinite duration should only be used if absolutely necessary.

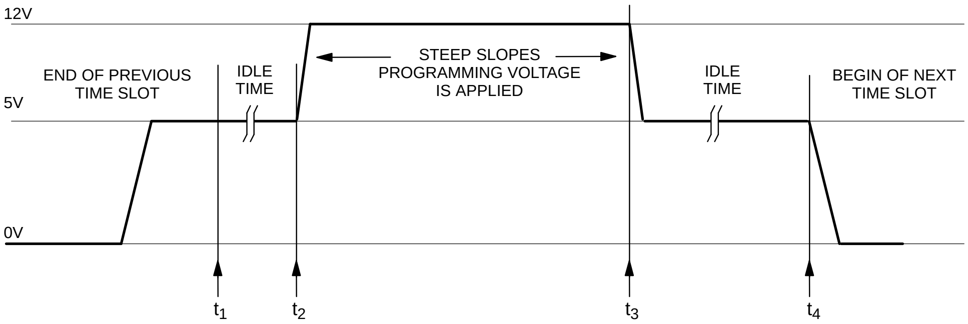
The time to end a pulse of infinite duration strongly depends on the baud rate of the communication between host and DS2480. Neglecting the response time of the host, the minimum pulse durations are: 86.8 µs at 115.2 kbps, 173.6 µs at 57.6 kbps, 520 µs at 19.2 kbps and 1.04 ms at 9.6 kbps.

STRONG PULL–UP TO 5V, INFINITE DURATION Figure 7c



As before, processing the command is essentially the same, regardless if it is for a strong pull–up or a programming pulse. At t_1 the host starts sending the pulse command byte. At t_2 the DS2480 has received the command and immediately activates the strong pull–up or switches in the 12V programming voltage. To end the pulse, the DS2480 must receive a termination command, code F1h, which occurs at t_3. The termination command does not generate a response byte. The DS2480 will immediately end the pulse and send out the response byte of the pulse command. The idle time between t_1 and t_2 is determined by the time to transmit the command byte at the selected baud rate. The idle time between t_3 and t_4 is comprised of the time to transmit the pulse response byte, plus the response time of the host plus the time to transmit the command and/or data to generate the next time slot.

12V PROGRAMMING PULSE, INFINITE DURATION Figure 7d

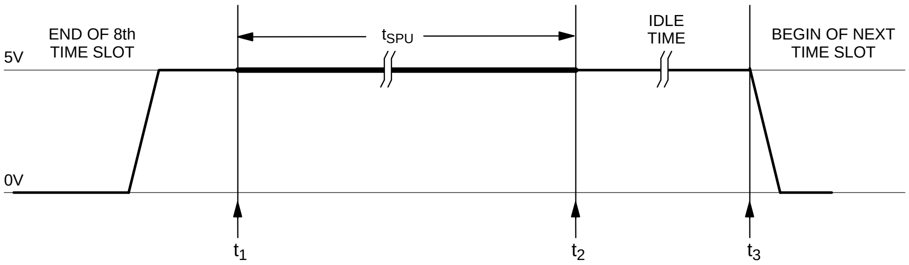


PULSE WAVE FORM, ARMED

As explained in section Communication Commands, bit 1 of the pulse command allows the arming of a strong pull–up to 5V if the bit is set to 1. If the strong pull–up is armed and the device is switched to data mode, there will be a strong pull–up immediately following every byte on the 1–Wire bus. This mode is implemented to provide extra energy when writing to EEPROM devices or to do a temperature conversion with the DS1920 Temperature iButton, for example. These devices need a strong pull–up immediately after the power–consuming activity has been initiated by a command code.

To arm the strong pull–up, one usually generates a "dummy" pulse with bit 1 being 1 while the device is in command mode. To save time, the dummy pulse may immediately be terminated by sending the termination command, code F1h. Then one switches to data mode and sends a command code that makes one or more slaves on the 1–Wire bus require extra energy. After the command execution is finished, one switches back to command mode and disarms the strong pull–up by generating another dummy pulse. A complete temperature conversion sequence that shows the use of the armed pulse is included in Section "Software Driver Examples".

STRONG PULL–UP TO 5V, ARMED, PREDEFINED DURATION Figure 8



Figures 8 shows the timing of the strong pull–up in data mode. At t_1 the 8th time slot of the byte sent to the 1–Wire bus is completed. Without any delay the DS2480 now activates the strong pull–up and simultaneously starts sending the data response byte to the host. At t_2 the strong pull–up ends and the DS2480 sends a pulse response byte to the host. The idle time between t_2 and t_3 is comprised of the time to transmit the pulse response byte, plus the response time of the host plus the time to transmit the command and/or data to generate the next time slot.

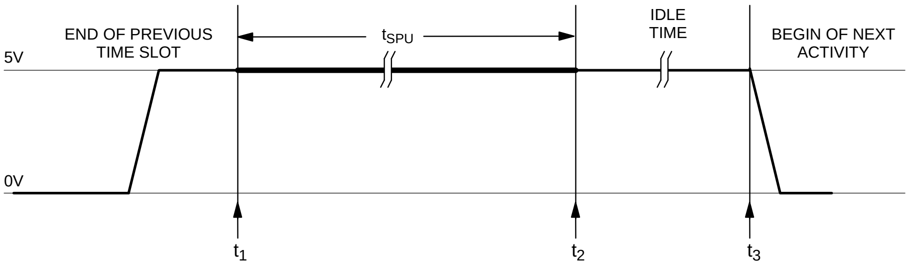
Since in data mode the pulse termination command is not applicable, the duration of the strong pull–up must be limited. See Table 4, parameter 011 (Strong Pull–Up Duration) for details. Selecting infinite duration will require a power–on or master reset cycle to get the DS2480 back to communicating with the host.

SINGLE BIT WITH STRONG PULL–UP

Similar to the pulse command the Single Bit command also allows generating a strong pull–up immediately following a time slot. The strong pull–up of the Single Bit command, however, is controlled directly by bit 1 of the command code and therefore needs not be armed. Since the DS2480 remains in command mode when using the Single Bit command, any duration of the strong pull–up including infinite may be selected.

Figure 9 shows the timing of the Single Bit command with the immediately following strong pull–up. Strong pull–up immediately following a single time slot is implemented to support the DS1954 Crypto iButton. After having received the command code to run a program, the Crypto iButton sends out an acknowledge bit and then starts the microcomputer to run the firmware program for a predefined time.

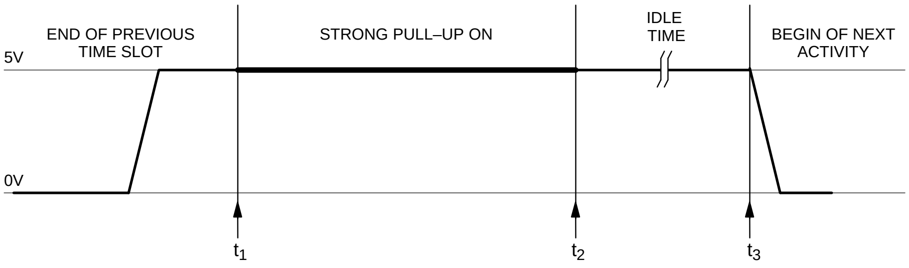
SINGLE BIT WITH STRONG PULL–UP, PREDEFINED DURATION Figure 9a



For predefined duration (Figure 9a) the timing is as follows: At t_1 the time slot is completed. Now the DS2480 activates the strong pull–up and simultaneously starts sending the response byte of the Single Bit command to the host. At t_2 the strong pull–up ends and the DS2480 sends out a pulse response byte. The idle time between t_2 and t_3 is comprised of the time to transmit the pulse response byte, plus the response time of the host plus the time to transmit the command and/or data to generate the next time slot.

For infinite duration (Figure 9b) the strong pull–up also begins immediately after the time slot is completed. To end the strong pull–up, the DS2480 must receive a termination command, code F1h, which occurs at t_2. The termination command does not generate a response byte. The DS2480 will then immediately end the strong pull–up and send out a pulse response byte. Everything else is the same as with predefined duration.

SINGLE BIT WITH STRONG PULL–UP, INFINITE DURATION Figure 9b



The minimum duration of a strong pull–up of infinite duration strongly depends on the baud rate of the communication between host and DS2480. The host must first receive the response byte of the single–bit command, react to it and then transmit the termination command. Neglecting the hosts response time, the shortest duration of an infinite strong pull–up therefore is 173.6 µs at 115.2 kbps, 347.2 µs at 57.6 kbps, 1.04 ms at 19.2 kbps and 2.08 ms at 9.6 kbps.

SOFTWARE DRIVER EXAMPLES

The DS2480 requires a software driver that translates the activities to be generated on the 1–Wire bus into the appropriate commands. The examples below cover typical situations, such as reading the ROM, writing to the scratchpad of a Memory iButton, reading the memory of a Memory iButton, programming an Add–Only iButton EPROM and performing a temperature conversion with the Temperature iButton DS1920. An example for the use of the Search Accelerator is included in the description of the Search Accelerator, earlier in this document.

The DS2480 command codes used in these examples are valid for regular speed and will work properly on short 1–Wire busses (< 10 meters). The response byte on the reset command assumes a normal presence pulse, no alarm or short.

The DS2480 includes a 1–byte buffer that stores a byte received from the host while the previous byte is being translated into activity on the 1–Wire bus. For this reason the host may send another byte even without having received the response byte. Sending bytes faster than they can be translated into 1–Wire activities may result in loss of data and/or synchronization and therefore should be avoided.

READ ROM SEQUENCE

Action Sequence	Host TX	Host RX
Generate Reset Pulse	**C1**	**C9**
Set data mode	**E1**	**(nothing)**
Read ROM Command	33	(as sent)
Read ROM ID (8 bytes)	FF (x8)	ROM ID
Set command mode	E3	(nothing)
Generate Reset Pulse	**C1**	**C9**

WRITE SCRATCHPAD SEQUENCE

Write 2 bytes to scratchpad at memory locations 16h and 17h

Action Sequence	Host TX	Host RX
Generate Reset Pulse	**C1**	**C9**
Set data mode	**E1**	**(nothing)**
Skip ROM Command	CC	CC
Write Scratchpad Cmd.	0F	(as sent)
Starting Address TA1	16	(as sent)
Starting Address TA2	00	(as sent)
Write to the Scratchpad	(2 bytes)	(as sent)
Set command mode	E3	(nothing)
Generate Reset Pulse	**C1**	**C9**

READ MEMORY SEQUENCE

Read 8 bytes from memory address 0040h

Action Sequence	Host TX	Host RX
Generate Reset Pulse	**C1**	**C9**
Set data mode	**E1**	**(nothing)**
Skip ROM Command	CC	(as sent)
Read Memory Command	F0	(as sent)
Starting Address TA1	40	(as sent)
Starting Address TA2	00	(as sent)
Read 8 bytes of data	FF (x8)	data
Set command mode	E3	(nothing)
Generate Reset Pulse	**C1**	**C9**

WRITE EPROM SEQUENCE (DS2505)

Write memory starting at address 40h

Action Sequence	Host TX	Host RX
Set Vpp dur. = 512 µs	**29**	**28**
Generate Reset Pulse	**C1**	**C9**
Set data mode	**E1**	**(nothing)**
Skip ROM Command	CC	(as sent)
Write Memory Command	0F	(as sent)
Starting Address TA1	40	(as sent)
Starting Address TA2	00	(as sent)
*** Send data byte	(data)	(as sent)
Receive CRC16	FF (x2)	CRC16
Set command mode	E3	(nothing)
Generate Program Pulse	**FD**	**response**
Set data mode	**E1**	**(nothing)**
Read written byte	FF	(data)
Go to ******* to write the next byte or end the sequence as shown below.		
Set command mode	E3	(nothing)
Generate Reset Pulse	**C1**	**C9**

TEMPERATURE CONVERSION SEQUENCE

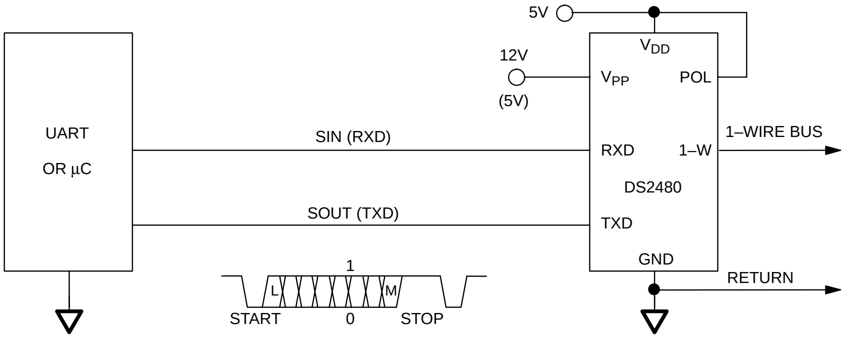
Action Sequence	Host TX	Host RX
Set pull–up dur. = 524 ms	**39**	**38**
Generate Reset Pulse	**C1**	**C9**
Set data mode	**E1**	**(nothing)**
Skip ROM Command	CC	(as sent)
Set command mode	E3	(nothing)
Arm strong pull–up	**EF**	**(nothing)**
Terminate pulse	**F1**	**response**
Set data mode	**E1**	**(nothing)**
Convert Temperature	44	(as sent)
Wait for pulse response	(nothing)	response
Set command mode	E3	(nothing)
Disarm strong pull–up	**ED**	**(nothing)**
Terminate pulse	**F1**	**response**
Generate Reset Pulse	**C1**	**C9**

HARDWARE APPLICATION EXAMPLES

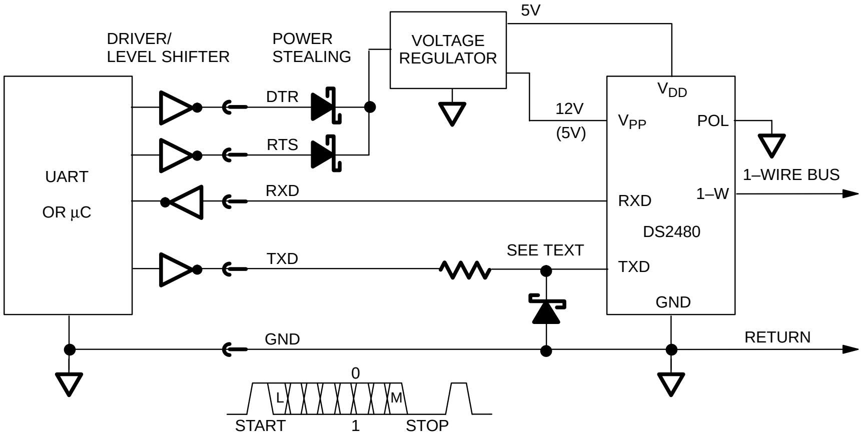
This section discusses five typical application scenarios of the DS2480. When directly designed into a new product, the device can be connected as shown in Figure 10a.

The circuit becomes more complex if a 1–Wire bus is to be interfaced to a port that provides and expects inverted signals, but does not necessarily meet the RS232C (± 12V) standard (Figure 10b).

UART DIRECT Figure 10a



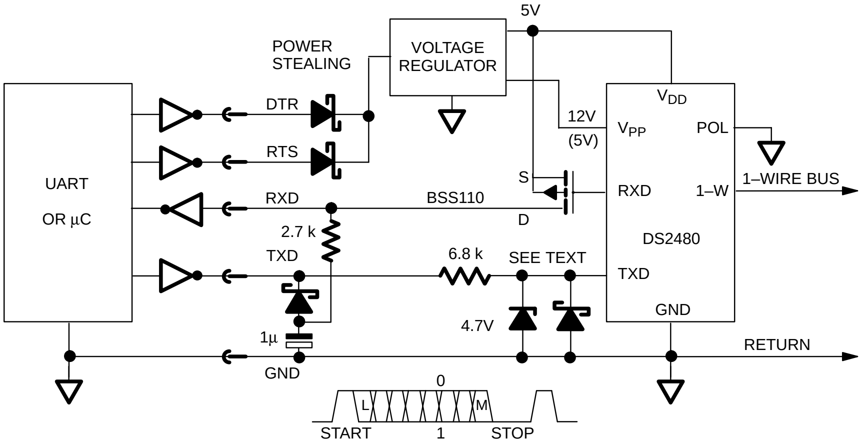
+/(–) 5V RS232 Figure 10b



The signals DTR and RTS provide the power to operate the DS2480. The resistor in the TXD line and the Schottky diode limit the negative voltage at the TXD pin of the DS2480 to 0.3V maximum. The resistor is typically 4.7 kΩ. If the inverting driver is current–limited to 1 mA the resistor is not required.

From the DS2480's perspective, this circuit will work with inverted signals of ± 5V as well as 0 to 5V. Depending on the voltage levels the host expects, it may be necessary to generate a negative voltage on the RXD line. Figure 10c shows how this can be accomplished for a true RS232C system.

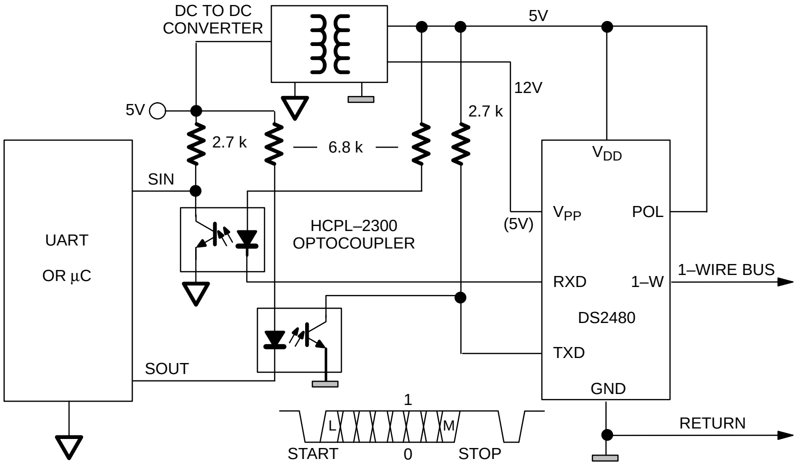
± 12V RS232 Figure 10c



In the interface to a true RS232C system (± 12V, Figure 10c) the power for the DS2480 is stolen from DTR and RTS. The software must make sure that at least one of these signals constantly provides the 12V operating voltage. The 6.8 kΩ resistor and the 4.7 V Zener diode in the TXD line limit the positive voltage at the TXD pin of the DS2480. The Schottky diode limits the negative voltage to 0.3V maximum. The Schottky diode in series with the capacitor forms a parasitic supply to generate the negative bias for the host's receive channel. The positive signal is switched in through the P–channel MOSFET that connects to the RXD output of the DS2480. In this circuit diagram the MOSFET switches the RXD line to +5V, which normally is sufficient for RS232C systems.

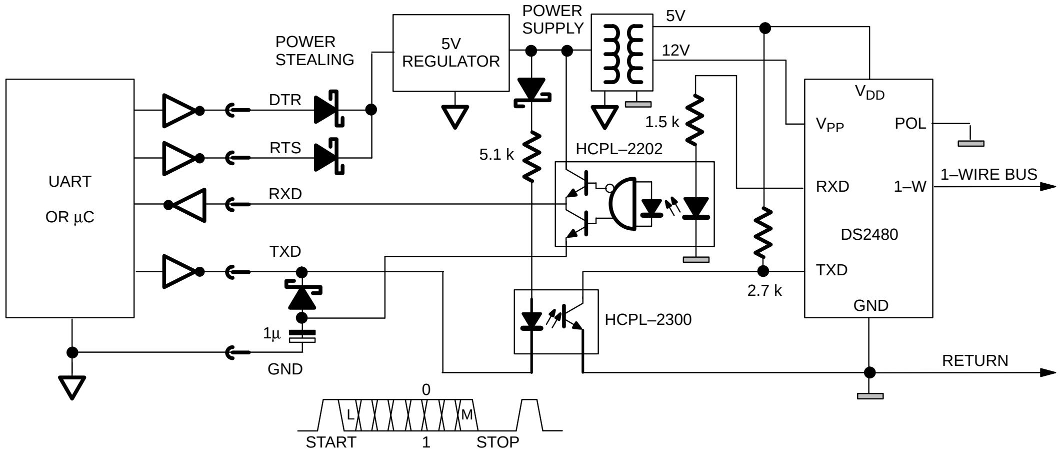
Switching to 12V is also possible, but requires a P–channel transistor with a different threshold voltage. The signal inversion caused by the transistor is compensated through the DS2480 by using a value code of 100, 101, 110 or 111 for the RS232 baud rate setting.

UART DIRECT OPTO ISOLATED Figure 10d



The circuit in Figure 10d is essentially the same as in Figure 10a. The main difference is the opto–isolation. The characteristics of the opto–isolators are not very critical. Using a different type, will affect the values of the resistors that limit the current through the LEDs and bias the photo transistors.

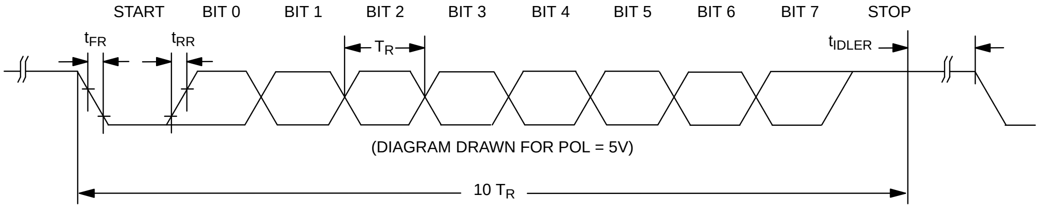
± 5 TO 12V RS232 OPTO ISOLATED Figure 10e



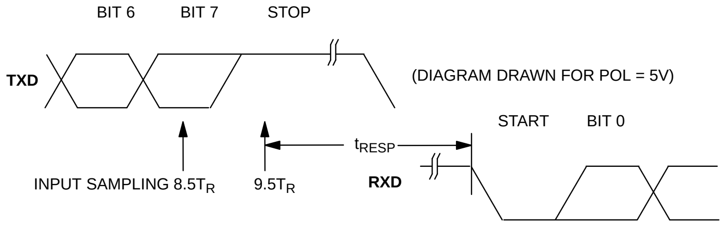
The circuit in Figure 10e combines the true RS232C interface with opto–isolation. The energy to power the LED in the TXD channel and to provide the positive voltage for the host's RXD input is stolen from DTR and RTS. The negative voltage for the RXD input is taken from the TXD line through a parasitic supply consisting of a Schottky diode in series with a capacitor. The HCPL–2202 opto–isolator has a totem pole output that allows switching in positive as well as negative voltage. The +5V are sufficient for most RS232C systems. Switching in 12V requires a opto–isolator with different voltage characteristics.

In the schematic the HCPL–2202 opto–isolator is sourced by the RXD pin of the DS2480. It can as well be connected the traditional way where the DS2480 sinks the current through the LED. This, however, causes a signal inversion that has to be compensated through the DS2480 by using a value code of 100, 101, 110 or 111 for the RS232 baud rate setting. Using other types of opto–isolators than shown in the schematic will at least require changing the values of the resistors.

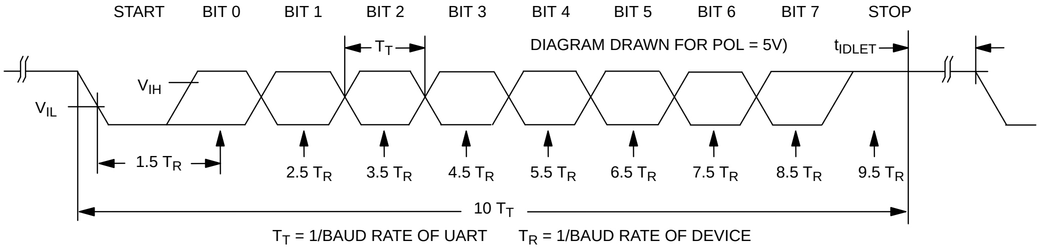
RS232 DATA TIMING RXD LINE Figure 11



RECEIVE DELAY TIMING Figure 12



RS232 DATA TIMING TXD LINE Figure 13

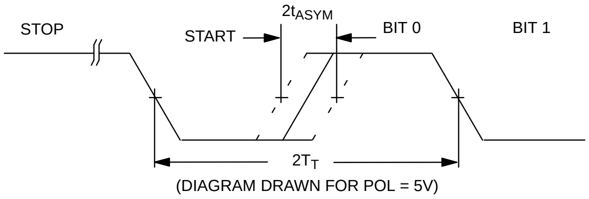


FUNCTION AND SPEED MATRIX Table 7

Baud Rate	1–Wire Speed			Function
	Regular	Flexible	Overdrive	
9600 bps	√	√	√	Search
	√	√	√	Command/Data
19200 bps	X	X	√	Search
	√	X	√	Command/Data
57600 bps	X	X	√	Search
	X	X	√	Command/Data
115200 bps	X	X	X	Search
	X	X	√	Command/Data

√	t_{IDLET} is guaranteed by the UART; no precautions necessary
X	not recommended unless t_{IDLET} is controlled by the host through a wait function

TXD LINE ASYMMETRY Figure 14



(DIAGRAM DRAWN FOR POL = 5V)

$t_{START} = T_T + (–)\ t_{ASYM}$ $t_{BIT0} = T_T – (+)\ t_{ASYM}$

ABSOLUTE MAXIMUM RATINGS*

Voltage on 1–W to Ground	–0.5V to +14.0V
Voltage on RXD, TXD, POL to Ground	–0.5V to +7.0V
Operating Temperature	–40°C to +85°C
Storage Temperature	–55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 4.5V to 5.5V;–40°C to 85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	
Programming Voltage	V_{PP}	11.75		12.25	V	1
Operating Current	I_{DD}		3.0	5.0	mA	
Idle Current on V_{PP}	I_{PP}		20		µA	2
Active Pull–up Timer Threshold	V_{IAPTO}	V_{DD}–1.4	V_{DD}–1.1		V	
Active Pull–up On Threshold	V_{IAPO}		0.95	1.2	V	
1–Wire Input High	V_{IH1}	3.4			V	
1–Wire Input Low	V_{IL1}			1.8	V	
TXD/POL input resistor	R_I	30			kΩ	3
TXD/POL input levels	V_{IH}	2.7			V	
TXD/POL input levels	V_{IL}			0.8	V	
1–Wire weak pull–up current	I_{WEAKPU}	1.5	3.0	5.0	mA	
1–Wire active pull–up current	I_{ACTPU}	9	15		mA	
Strong pull–up voltage drop @ 10 mA load on 1–W	ΔV_{STRPU}			0.6	V	4
Programming voltage drop @ 10 mA load on 1–W	ΔV_{PROG}			0.24	V	5
RXD sink current @ 0.4V	I_{OLR}	6			mA	
RXD source current @ V_{DD}–0.4V	I_{OHR}	–4			mA	
Power On Reset Trip Point	V_{POR}			3.3	V	
VPP Sensor Trip Point	V_{PPTRIP}		9.5		V	

CAPACITANCES

(t_A = 25°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
TXD/POL input capacitance	C_{IN}		5		pF	
1–Wire input capacitance	C_{IN1}		10		pF	

AC ELECTRICAL CHARACTERISTICS

(V_{DD} = 4.5V to 5.5V;–40°C to 85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
UART Bit Time	T_T	8.68		104	µs	6
Device Bit Time	T_R	8.68		104	µs	6, 7
Fall Time RXD	t_{FR}	8		20	ns	8
Rise Time RXD	t_{RR}	13		27	ns	8
Transmit Idle Time	t_{IDLET}	0			µs	9
Receive Idle Time	t_{IDLER}		t_{IDLET}		µs	
Asymmetry	t_{ASYM}			1	µs	10
Arrival Response Time	t_{ARR}	4.4		52	µs	11
Master Reset Time	t_{MR}	104			µs	12
Active Pull–Up on Time	t_{APUOT}	0.5		2.0	µs	13
Response Time	t_{RESP}	8.68 + Δ		104 + Δ	µs	11, 14

NOTES:

1. V_{PP} – ΔV_{PROG} must be within 11.5 to 12.0V.
2. Applies only if a 12.0V supply is connected. If V_{PP} and V_{DD} are tied together, current is less than 1 µA.
3. Input load is to GND.
4. Voltage difference between V_{DD} and 1–W.
5. Voltage difference between V_{PP} and 1–W.
6. 8.68 µs (115.2 kbps), 52 µs (19.2 kbps), 17.36 µs (57.6 kbps), 104 µs (9.6 kbps).
7. Nominal values; tolerance = ±5%.
8. At V_{CC} = 5.0V and 100 pF load to GND.
9. See Table 7, Function and Speed Matrix.
10. Independent of baud rate.
11. Minimum at 115.2 kbps, maximum at 9.6 kbps.
12. The master reset cycle is complete after t_{MR} is over.
13. Minimum value at Overdrive speed; maximum value at regular speed.
14. Δ is the time to complete the activity on the 1–Wire bus; values range from 0 (configuration command) up to 5130 µs (alarming presence pulse)