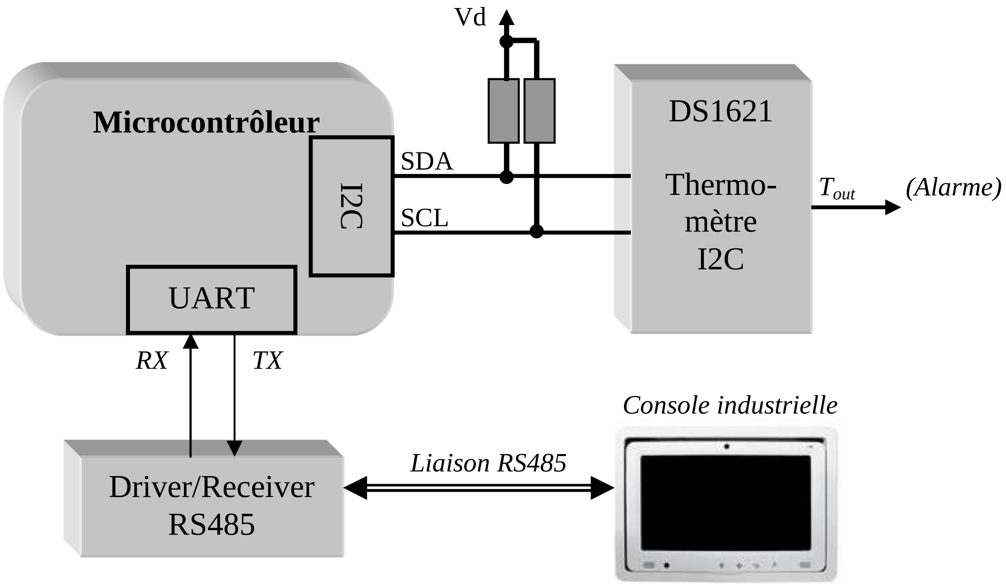
UTBM **Département Energie**
Printemps 2015 **Durée : 2 heures. Sans documents.**

Médian MC43

Thermomètre thermostat numérique I2C DS1621

Le thermomètre numérique DS1621 dont la documentation est donnée en annexe permet la mesure de la température dans un appareillage industriel et le déclenchement d'une alarme. Une console industrielle RS232 permet d'afficher certains paramètres liés au processus industriel. Le microcontrôleur dsPIC30F4011 communique avec le DS1621 par l'intermédiaire d'un bus I2C piloté par son contrôleur I2C interne. Il effectue les communications avec la console par l'intermédiaire de son UART interne associé à une liaison physique RS485. Le schéma fonctionnel du montage étudié est donné ci-dessous.



Description rapide du DS1621 :
Le thermomètre/thermostat DS1621 permet la mesure de températures comprise entre -55°C et 125°C (registre ***Read Temperature***) avec une précision de 0.5°C (valeurs sur 9 bits transmises sur 2 octets). La fonction thermostat consiste en l'activation de la sortie ***Tout*** lorsque la température mesurée dépasse la valeur haute (registre ***TH***). La sortie ***Tout*** se désactive lorsque la température redescend en dessous de la valeur basse (registre ***TL***) comme expliqué sur le diagramme d'hystérésis de la Figure3. La Table2 et la Figure2 donnent la correspondance entre la température et la valeur numérique des registres de température. La précision peut optionnellement être améliorée en appliquant la relation de la page 4 qui exploite les 2 registres complémentaires ***Read Counter*** (valeur ***Count_Remain***) et ***Read Slope*** (valeur ***Count_Per_C***).
La conversion A/N interne au DS1621 de la température peut s'effectuer de manière continue (en permanence) après avoir transmis une commande de conversion (***Start Convert T***), ou en mode mono coup c'est-à-dire uniquement à chaque envoi de la commande de conversion ***Start Convert T*** (configuration par le bit ***1SHOT*** du mot de contrôle). La polarité de ***Tout*** et le mode de conversion sont à définir par écriture dans le mot de contrôle (***Configuration register***) décrit pages 5 et 6 de la datasheet (bits ***POL*** et ***1SHOT***). Les autres bits de ce registre sont des bits d'état accessibles en lecture uniquement.

Principe de communication I2C du DS1621 : un mot de contrôle (***Command Byte***) permet de préciser une commande ou le registre auquel on désire accéder. Le ***Command Byte*** doit être transmis pour chaque trame d'écriture après envoi de l'adresse du DS1621. Les différentes significations du ***Command byte*** sont données page 10 et Table3 de la datasheet. Des exemples sont fournis Figure 5 et page 12.

I Etude du thermomètre I2C DS1621 (12 points)

I.1 On désire attribuer l'adresse 0x90 au DS1621. Indiquer comment procéder.

I.2 Donner la condition nécessaire pour que l'échantillonnage de la température à Fe=1 Hz soit réalisable sans filtrage anti-repliement.

I.3 Indiquer si un conflit électrique est possible ou non sur un bus I2C et pourquoi.

I.4 On désire configurer le DS1621 en mode « conversion continue » et « sortie ***Tout*** active à l'état haut ». Donner le ***Configuration register*** correspondant.

I.5 Donner la trame I2C permettant de configurer le ***Configuration register*** du DS1621 en précisant le contenu des octets en hexadécimal et le sens des acquittements.

I.6 Donner la trame I2C permettant de configurer la température haute ***TH*** à 90°C.

I.7 Donner la trame I2C permettant de configurer la température basse ***TL*** à 70°C.

I.8 Donner la trame I2C permettant de lancer les conversions A/N de la température.

I.9 On désire mesurer la température en précision étendue (exploitant ***Count_Per_C*** et ***Count_Remain***). Tous les registres nécessaires au calcul de la température sont lus **en une seule trame**. Donner la trame I2C correspondante.

I.10 Déterminer la durée totale de transmission de cette trame dans le cas où la vitesse de transmission est maximale.

II Etude des communications avec la console industrielle (8 points)

II.1 Justifier le choix d'une liaison RS485 plutôt que RS232.

II.2 Rappeler la vitesse de transmission maximale et la longueur maximale des liaisons RS232 et RS485.

II.3 Le format de transmission est le suivant : 115200 bauds, données de 8 bits, parité paire, 1 bit de stop. Représenter la trame émise sur la broche TX en sortie de l'UART du microcontrôleur dans le cas de la transmission de l'octet de donné 0x34.

II.4 Déterminer la vitesse de transmission utile maximale en octets/s.

II.5 La réception est gérée par interruption et stockage en buffer circulaire. Donner l'organigramme du principe fonctionnel de la fonction d'interruption en charge du stockage dans le buffer circulaire des données reçues.

II.6 Compléter la fonction d'interruption correspondante ci-dessous écrite en langage C pour le microcontrôleur dsPIC30F4011.

```
#define     taillebuffer1     200         /* taille buffer de reception UART */
#define     SCI_noerror       0x0000
#define     SCI_buf_error     0x0001      /* indicateur buffer de réception plein */
#define     SCI_RX_error      0x0002      /* indicateur erreur de réception UART */
int   buffer1[taillebuffer1];             /* buffer de réception */
int   RD1;                                /* index de lecture dans buffer1 */
int   WR1;                                /* index d'écriture dans buffer1 */
int   nbval1;                             /* nbre d'octets stockés dans buffer1 */
int   SCI_error;                          /* indicateur de mémorisation des erreurs*/

/* Interruption de réception par l'UART */
void __attribute__((__interrupt__)) _U2RXInterrupt(void)
      {
      IFS1bits.U2RXIF = 0;    /* Remise à 0 du flag d'interruption */
      if(..................) SCI_error|=SCI_buf_error;
      else
            {
            if(U2STAbits.OERR==1 && U2STAbits.FERR==1 && U2STAbits.PERR==1)
               SCI_error|=SCI_RX_error;
            ............ = U2RXREG; /* stockage donnée reçue en buffer circulaire */
            if(...................) WR1=0;
            nbval1++;
            }
      }
```

DS1621
Digital Thermometer and

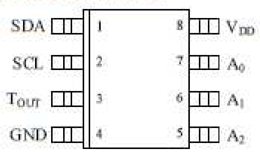
BENEFITS AND FEATURES

- Simply Adds Temperature Monitoring and Control to Any System
  - Measures Temperatures From -55°C to +125°C in 0.5°C Increments. Fahrenheit Equivalent is -67°F to 257°F in 0.9°F Increments
  - Temperature is Read as a 9-Bit Value (2-Byte Transfer)
  - Converts Temperature to Digital Word in Less than 1s
  - Thermostatic Settings are User Definable And Nonvolatile
- Can Be Used in a Wide Variety of Applications
  - Power Supply Range (2.7V to 5.5V)
  - Data is Read From/Written Via a 2-Wire Serial Interface (Open Drain I/O Lines)
- Saves Space
  - Temperature Measurements Require No External Components
  - 8-pin DIP or SO package (208-mil) Packages

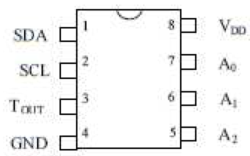
APPLICATIONS

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers

PIN ASSIGNMENT



DS1621S 8-PIN SO (150mil)
DS1621V 8-PIN SO (208mil)



DS1621 8-PIN DIP (300mil)

PIN DESCRIPTION

SDA	- 2-Wire Serial Data Input/Output
SCL	- 2-Wire Serial Clock
GND	- Ground
T_{OUT}	- Thermostat Output Signal
A0	- Chip Address Input
A1	- Chip Address Input
A2	- Chip Address Input
V_{DD}	- Power Supply Voltage

DESCRIPTION

The DS1621 Digital Thermometer and Thermostat provides 9-bit temperature readings, which indicate the temperature of the device. The thermal alarm output, T_{OUT}, is active when the temperature of the device exceeds a user-defined temperature TH. The output remains active until the temperature drops below user defined temperature TL, allowing for any hysteresis necessary.

User-defined temperature settings are stored in nonvolatile memory so parts may be programmed prior to insertion in a system. Temperature settings and temperature readings are all communicated to/from the DS1621 over a simple 2-wire serial interface.

ORDERING INFORMATION

ORDERING NUMBER	PACKAGE MARKING	DESCRIPTION
DS1621	DS1621	DS1621 in 300 mil DIP
DS1621+	DS1621 (See Note)	DS1621 in Lead-Free 300 mil DIP
DS1621S	DS1621	DS1621 in 150 mil SOIC
DS1621S+	DS1621 (See Note)	DS1621 in Lead-Free 150 mil SOIC
DS1621S/T&R	DS1621	DS1621 in 150 mil SO, 2500 Piece Tape-and-Reel
DS1621S+T&R	DS1621 (See Note)	DS1621 in Lead-Free 150 mil SO, 2500 Piece Tape-and-Reel
DS1621V	DS1621V	DS1621 in 208 mil SOIC
DS1621V+	DS1621V (See Note)	DS1621 in Lead-Free 208 mil SOIC
DS1621V/T&R	DS1621V	DS1621 in 208 mil SO, 2500 Piece Tape-and-Reel
DS1621V+T&R	DS1621V (See Note)	DS1621 in Lead-Free 208 mil SO, 2500 Piece Tape-and-Reel

Note: A "+" symbol will also be marked on the package near the Pin 1 indicator.

Table 1. DETAILED PIN DESCRIPTION

PIN	SYMBOL	DESCRIPTION
1	SDA	Data input/output pin for 2-wire serial communication port.
2	SCL	Clock input/output pin for 2-wire serial communication port.
3	T_{OUT}	Thermostat output. Active when temperature exceeds TH; will reset when temperature falls below TL.
4	GND	Ground pin.
5	A2	Address input pin.
6	A1	Address input pin.
7	A0	Address input pin.
8	V_{DD}	Supply voltage input power pin. (2.7V to 5.5V)

OPERATION

Measuring Temperature
A block diagram of the DS1621 is shown in Figure 1.

The DS1621 measures temperature using a bandgap-based temperature sensor. A delta-sigma analog-to-digital converter (ADC) converts the measured temperature to a digital value that is calibrated in °C; for °F applications, a lookup table or conversion routine must be used.

The temperature reading is provided in a 9-bit, two's complement reading by issuing the READ TEMPERATURE command. Table 2 describes the exact relationship of output data to measured temperature. The data is transmitted through the 2-wire serial interface, MSB first. The DS1621 can measure temperature over the range of -55°C to +125°C in 0.5°C increments.

Figure 1. DS1621 FUNCTIONAL BLOCK DIAGRAM

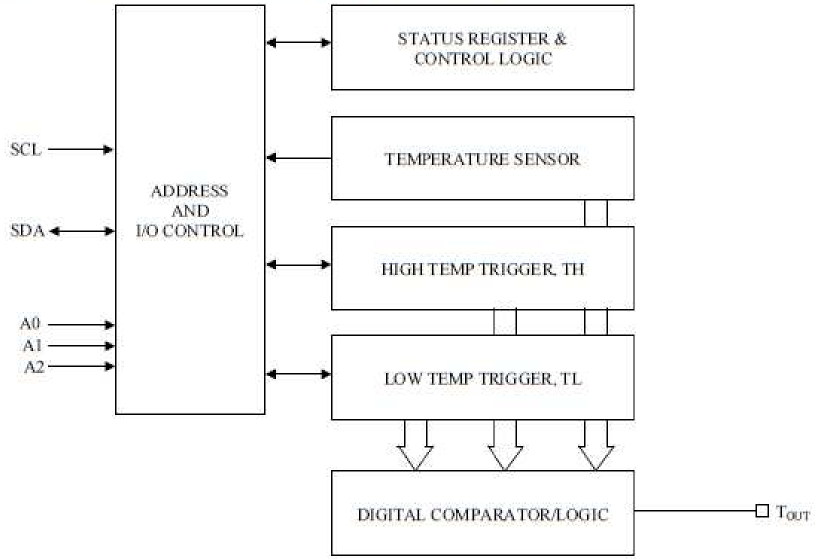


Table 2. TEMPERATURE/DATA RELATIONSHIPS

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	01111101 00000000	7D00h
+25°C	00011001 00000000	1900h
+½°C	00000000 10000000	0080h
+0°C	00000000 00000000	0000h
-½°C	11111111 10000000	FF80h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

Since data is transmitted over the 2-wire bus MSB first, temperature data may be written to/read from the DS1621 as either a single byte (with temperature resolution of 1°C) or as two bytes. The second byte would contain the value of the least significant (0.5°C) bit of the temperature reading as shown in Table 1. Note that the remaining 7 bits of this byte are set to all "0"s.

Temperature is represented in the DS1621 in terms of a ½°C LSB, yielding the following 9-bit format:

Figure 2. TEMPERATURE, TH, and TL FORMAT

MSB								LSB							
1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0

T = -25°C

Higher resolutions may be obtained by reading the temperature and truncating the 0.5°C bit (the LSB) from the read value. This value is TEMP_READ. A Read Counter command should be issued to yield the COUNT_REMAIN value. The Read Slope command should then be issued to obtain the COUNT_PER_C value. The higher resolution temperature may be then be calculated by the user using the following:

$$TEMPERATURE = TEMP\_READ - 0.25 + \frac{(COUNT\_PER\_C - COUNT\_REMAIN)}{COUNT\_PER\_C}$$

The DS1621 always powers up in a low power idle state, and the Start Convert T command must be used to initiate conversions.

The DS1621 can be programmed to perform continuous consecutive conversions (continuous-conversion mode) or to perform single conversions on command (one-shot mode). The conversion mode is programmed through the 1SHOT bit in the configuration register as explained in the *Operation and Control* section of this datasheet. In continuous conversion mode, the DS1621 begins continuous conversions after a Start Convert T command is issued. Consecutive conversions continue to be performed until a Stop Convert T command is issued, at which time the device goes into a low-power idle state. Continuous conversions can be restarted at any time using the Start Convert T command.

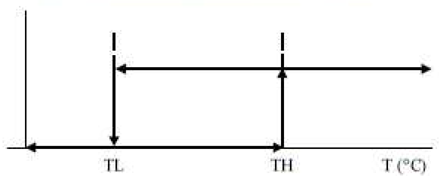
In one-shot mode, the DS1621 performs a single temperature conversion when a Start Convert T command is issued. When the conversion is complete, the device enters a low-power idle state and remains in that state until a single temperature conversion is again initiated by a Start Convert T command.

Thermostat Control
In its operating mode, the DS1621 functions as a thermostat with programmable hysteresis as shown in Figure 3. The thermostat output updates as soon as a temperature conversion is complete.

When the DS1621's temperature meets or exceeds the value stored in the high temperature trip register (TH), the output becomes active and will stay active until the temperature falls below the temperature stored in the low temperature trigger register (TL). In this way, any amount of hysteresis may be obtained.

The active state for the output is programmable by the user so that an active state may either be a logic "1" (V_{DD}) or a logic "0" (0V). This is done using the POL bit in the configuration reagister as explained in the *Operation and Control* section of this datasheet.

Figure 3. THERMOSTAT OUTPUT OPERATION



OPERATION AND CONTROL
The DS1621 must have temperature settings resident in the TH and TL registers for thermostatic operation. A configuration/status register also determines the method of operation that the DS1621 will use in a particular application, as well as indicating the status of the temperature conversion operation.

The configuration register is defined as follows:

MSb	Bit 6	Bit5	Bit 4	Bit 3	Bit 2	Bit 1	LSb
DONE	THF	TLF	NVB	X	X	POL	1SHOT

where

DONE = Conversion Done bit. "1" = Conversion complete, "0" = Conversion in progress.

THF = Temperature High Flag. This bit will be set to "1" when the temperature is greater than or equal to the value of TH. It will remain "1" until reset by writing "0" into this location or removing power from the device. This feature provides a method of determining if the DS1621 has ever been subjected to temperatures above TH while power has been applied.

TLF = Temperature Low Flag. This bit will be set to "1" when the temperature is less than or equal to the value of TL. It will remain "1" until reset by writing "0" into this location or removing power from the device. This feature provides a method of determining if the DS1621 has ever been subjected to temperatures below TL while power has been applied.

NVB = Nonvolatile Memory Busy flag. "1" = Write to an E^2 memory cell in progress, "0" = nonvolatile memory is not busy. A copy to E^2 may take up to 10 ms.

POL = Output Polarity Bit. "1" = active high, "0" = active low. This bit is nonvolatile.

1SHOT = One Shot Mode. If 1SHOT is "1", the DS1621 will perform one temperature conversion upon receipt of the Start Convert T protocol. If 1SHOT is "0", the DS1621 will continuously perform temperature conversions. This bit is nonvolatile.

X = Reserved.

For typical thermostat operation the DS1621 will operate in continuous mode. However, for applications where only one reading is needed at certain times or to conserve power, the one-shot mode may be used. Note that the thermostat output (T_{OUT}) will remain in the state it was in after the last valid temperature conversion cycle when operating in one-shot mode.

2-WIRE SERIAL DATA BUS
The DS1621 supports a bidirectional 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter, and a device receiving data as a receiver. The device that controls the message is called a "master." The devices that are controlled by the master are "slaves." The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1621 operates as a slave on the 2-wire bus. Connections to the bus are made via the open-drain I/O lines SDA and SCL.

The following bus protocol has been defined (See Figure 4):

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

Start data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

Stop data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth-bit.

Within the bus specifications a regular mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1621 works in both modes.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 4. DATA TRANSFER ON 2-WIRE SERIAL BUS

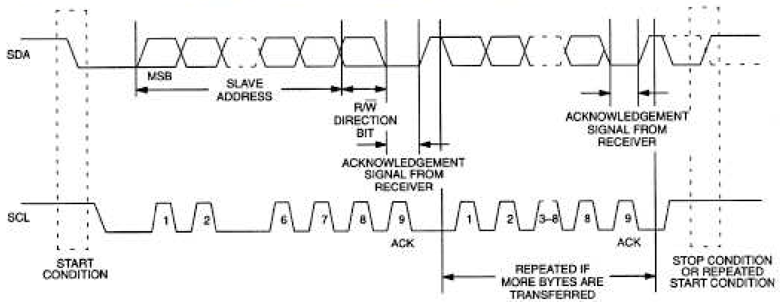


Figure 4 details how data transfer is accomplished on the 2-wire bus. Depending upon the state of the R/W bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.

2. **Data transfer from a slave transmitter to a master receiver.** The first byte, the slave address, is transmitted by the master. The slave then returns an acknowledge bit. Next follows a number of data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released.

The DS1621 may operate in the following two modes:

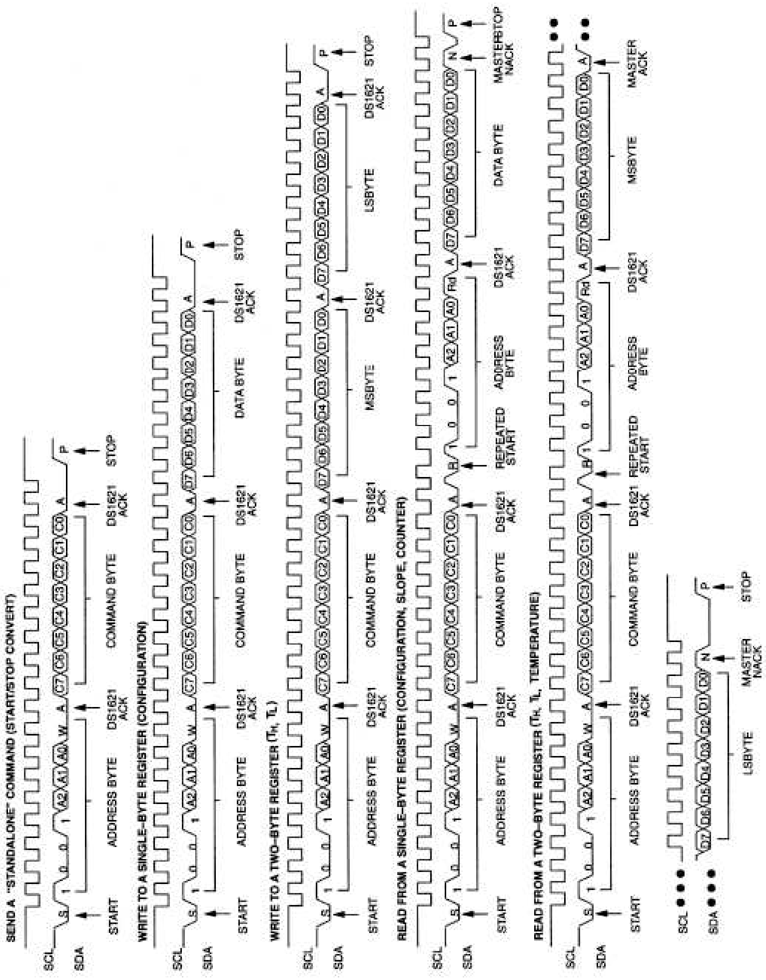
1. **Slave receiver mode:** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

2. **Slave transmitter mode:** The first byte is received and handled as in the slave receiver mode. However, in this mode the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1621 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

SLAVE ADDRESS

A control byte is the first byte received following the START condition from the master device. The control byte consists of a 4-bit control code; for the DS1621, this is set as 1001 binary for read and write operations. The next 3 bits of the control byte are the device select bits (A2, A1, A0). They are used by the master device to select which of eight devices are to be accessed. These bits are in effect the 3 least significant bits of the slave address. The last bit of the control byte ($R/\overline{W}$) defines the operation to be performed. When set to a "1" a read operation is selected, when set to a "0" a write operation is selected. Following the START condition the DS1621 monitors the SDA bus checking the device type identifier being transmitted. Upon receiving the 1001 code and appropriate device select bits, the slave device outputs an acknowledge signal on the SDA line.

Figure 5. 2-WIRE SERIAL COMMUNICATION WITH DS1621



COMMAND SET

Data and control information is read from and written to the DS1621 in the format shown in Figure 5. To write to the DS1621, the master will issue the slave address of the DS1621 and the R/$\overline{W}$ bit will be set to "0". After receiving an acknowledge, the bus master provides a command protocol. After receiving this protocol, the DS1621 will issue an acknowledge and then the master may send data to the DS1621. If the DS1621 is to be read, the master must send the command protocol as before and then issue a repeated START condition and the control byte again, this time with the R/$\overline{W}$ bit set to "1" to allow reading of the data from the DS1621. The command set for the DS1621 as shown in Table 3 is as follows:

Read Temperature [AAh]

This command reads the last temperature conversion result. The DS1621 will send 2 bytes, in the format described earlier, which are the contents of this register.

Access TH [A1h]

If R/$\overline{W}$ is "0" this command writes to the TH (HIGH TEMPERATURE) register. After issuing this command, the next 2 bytes written to the DS1621, in the same format as described for reading temperature, will set the high temperature threshold for operation of the T_{OUT} output. If R/$\overline{W}$ is "1" the value stored in this register is read back.

Access TL [A2h]

If R/$\overline{W}$ is "0" this command writes to the TL (LOW TEMPERATURE) register. After issuing this command, the next 2 bytes written to the DS1621, in the same format as described for reading temperature, will set the high temperature threshold for operation of the T_{OUT} output. If R/$\overline{W}$ is "1" the value stored in this register is read back.

Access Config [ACh]

If R/$\overline{W}$ is "0" this command writes to the configuration register. After issuing this command, the next data byte is the value to be written into the configuration register. If R/$\overline{W}$ is "1" the next data byte read is the value stored in the configuration register.

Read Counter [A8h]

This command reads the value Count_Remain. This command is valid only if R/$\overline{W}$ is "1".

Read Slope [A9h]

This command reads the value Count_Per_C. This command is valid only if R/$\overline{W}$ is "1".

Start Convert T [EEh]

This command begins a temperature conversion. No further data is required. In one-shot mode the temperature conversion will be performed and then the DS1621 will remain idle. In continuous mode this command will initiate continuous conversions.

Stop Convert T [22h]

This command stops temperature conversion. No further data is required. This command may be used to halt a DS1621 in continuous conversion mode. After issuing this command, the current temperature measurement will be completed and the DS1621 will remain idle until a Start Convert T is issued to resume continuous operation.

Table 3. DS1621 COMMAND SET

INSTRUCTION	DESCRIPTION	PROTOCOL	2-WIRE BUS DATA AFTER ISSUING PROTOCOL	NOTES
TEMPERATURE CONVERSION COMMANDS				
Read Temperature	Read last converted temperature value from temperature register.	AAh	<read 2 bytes data>	
Read Counter	Reads value of Count_Remain	A8h	<read data>	
Read Slope	Reads value of the Count_Per_C	A9h	<read data>	
Start Convert T	Initiates temperature conversion.	EEh	idle	1
Stop Convert T	Halts temperature conversion.	22h	idle	1
THERMOSTAT COMMANDS				
Access TH	Reads or writes high temperature limit value into TH register.	A1h	<write data>	2
Access TL	Reads or writes low temperature limit value into TL register.	A2h	<write data>	2
Access Config	Reads or writes configuration data to configuration register.	ACh	<write data>	2

NOTES:

1. In continuous conversion mode a Stop Convert T command will halt continuous conversion. To restart the Start Convert T command must be issued. In one-shot mode a Start Convert T command must be issued for every temperature reading desired.

2. Writing to the E^2 requires a maximum of 10ms at room temperature. After issuing a write command, no further writes should be requested for at least 10ms.

MEMORY FUNCTION EXAMPLE

Example: Bus master sets up DS1621 for continuous conversion and thermostatic function.

BUS MASTER MODE	DS1621 MODE	DATA (MSB FIRST)	COMMENTS
TX	RX	START	Bus Master initiates a START condition.
TX	RX	<address,0>	Bus Master sends DS1621 address; $R/\overline{W}$ = 0.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	ACh	Bus Master sends Access Config command protocol.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	02h	Bus Master sets up DS1621 for output polarity active high, continuous conversion.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	START	Bus Master generates a repeated START condition.
TX	RX	<address,0>	Bus Master sends DS1621 address; $R/\overline{W}$ = 0.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	A1h	Bus Master sends Access TH command.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	28h	Bus Master sends first byte of data for TH limit of +40°C.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	00h	Bus Master sends second byte of data for TH limit of +40°C.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	START	Bus Master generates a repeated START condition.
TX	RX	<address,0>	Bus Master sends DS1621 address; $R/\overline{W}$ = 0.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	A2h	Bus Master sends Access TL command.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	0Ah	Bus Master sends first byte of data for TL limit of +10°C.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	00h	Bus Master sends second byte of data for TL limit of +10°C.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	START	Bus Master generates a repeated START condition.
TX	RX	<address,0>	Bus Master sends DS1621 address; $R/\overline{W}$ = 0.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	EEh	Bus Master sends Start Convert T command protocol.
RX	TX	ACK	DS1621 generates acknowledge bit.
TX	RX	STOP	Bus Master initiates STOP condition.