SENSYLINK Microelectronics

(CT1820)
Single-Wire Digital Temperature Sensor

CT1820 is a Digital Temperature Sensor with ±0.5°C Accuracy Compatible with 1-wire Interface.
It is ideally used in HVAC, Thermal management and Smart Farm etc.

Table of Contents

Figures and Tables

±0.5 °C Temperature Sensor with Single-Wire Digital Interface

Description

CT1820 is a digital temperature sensor with ±0.5°C accuracy. Temperature data can be read out directly via Single-Wire interface (compatible with 1-wire bus in protocol) by MCU.

It includes a high precision band-gap circuit, a 12-bit analog to digital converter that can offer 0.0625°C resolution, a calibration unit with non-volatile memory, 8-bit CRC generator and a digital interface block.

The chip is specially calibrated for ±0.5 °C(Max.) accuracy over -10°C to 85°C range in factory before shipment to customers.

Each chip has a unique 64-bit ROM ID, which allows multiple devices to connect the same Single-Wire bus. MCU can distinguish and access each device individually by different ROM ID.

It has programmable temperature Alarm function for upper and lower trigger temperature.

Available Package: TO-92, TO-92S, TO-92S-2, MSOP-8, SOT-23 package.

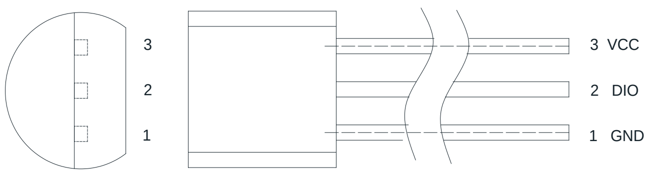
Features

- Operation Voltage: 2.5V to 5.5V
- Average Quiescent Current:
  30uA during Temperature conversion;
- Standby Current: 2.5uA (Max.)
- Temperature Conversion time: 30ms at 12-bit resolution
- Temperature Accuracy without calibration:
  ±0.5°C(Max.) from -10°C to 85°C
  ±1.0°C(Max.) from -10°C to 100°C
  ±2.0°C(Max.) from -55°C to 125°C
- 12 bit ADC for 0.0625°C resolution
- Compatible with 1-wire interface
- Programmable Upper/Lower trigger Temperature
- Compatible with DS18B20 and performance improved.
- Temperature Range: -55°C to 125°C

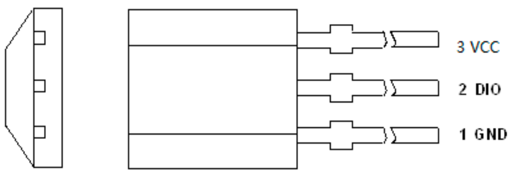
Applications

- Smart HVAC System
- Thermal Management
- Smart Farm

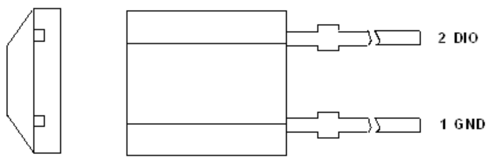
PIN Configurations (Top View)



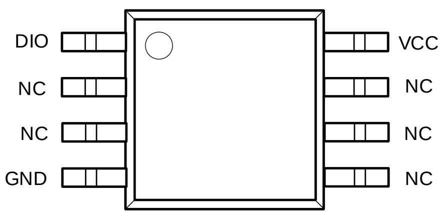
TO-92 (package code Z)



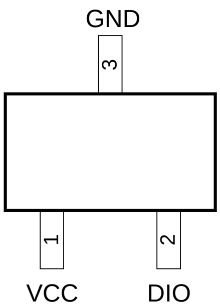
TO-92S (package code Z3)



TO-92S-2 (package code Z2)



MSOP-8(package code MM)



SOT-23(package code K)

Typical Application

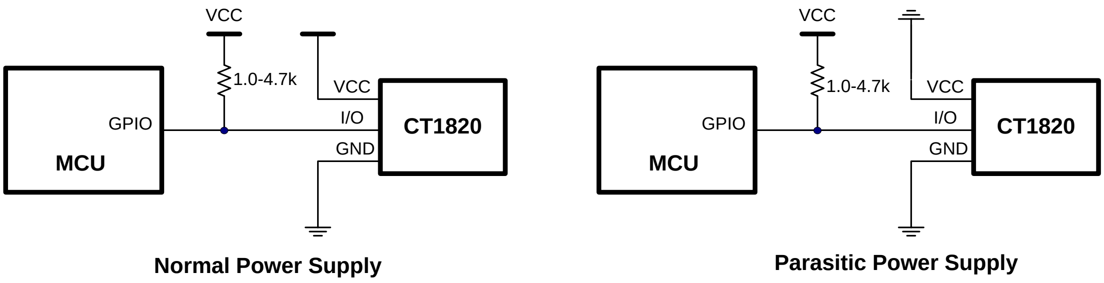


Figure 1. Typical Application of CT1820

Pin Description

PIN No.	PIN Name	Description
1	GND	Ground pin.
2	I/O	Digital interface data input and output pin, Generally there needs a pull-up resistor to VCC in most applications, between 1.0k and 4.7k. Also this pin can be used as parasitic power pin if there is no power from VCC pin.
3	VCC	Power supply input pin, if I/O pin is used as parasitic power pin, VCC pin should connect a 10nF ceramic cap at least to ground.

Function Block

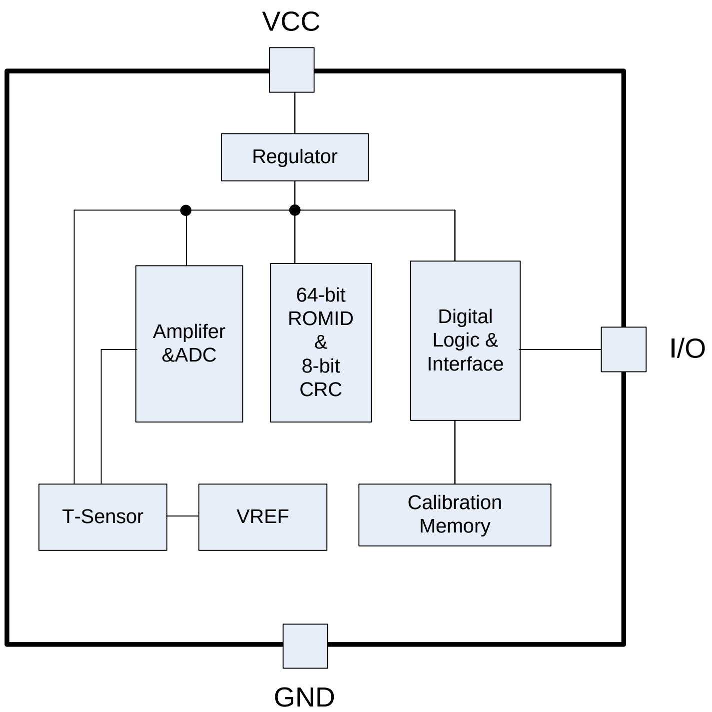
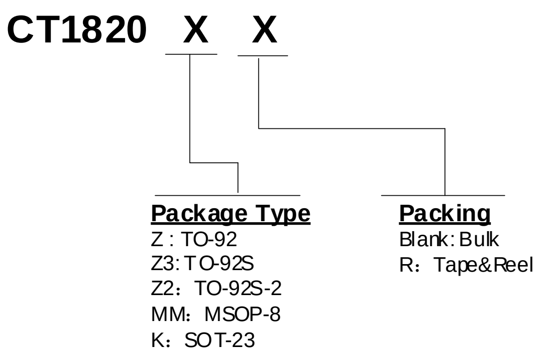


Figure 2. CT1820 function block

Ordering Information



Order PN	Accuracy	Green[1]	Package	Marking ID[2]	Packing	MPQ	Operation Temperature
CT1820Z	±0.5°C	Halogen free	TO-92	1820 YWWAXX	Bulk	1,000	-55°C~+125°C
CT1820Z3	±0.5°C	Halogen free	TO-92S	1820 YWWAXX	Bulk	1,000	-55°C~+125°C
CT1820Z2	±0.5°C	Halogen free	TO-92S-2	1820 YWWAXX	Bulk	1,000	-55°C~+125°C
CT1820MMR	±0.5°C	Halogen free	MSOP-8	1820 YWWAXX	Tape&Reel	3,000	-55°C~+125°C
CT1820KR	±0.5°C	Halogen free	SOT-23	ARWW	Tape&Reel	3,000	-55°C~+125°C

Notes

1. Based on ROHS Y2012 spec, Halogen free covers lead free. So most package types Sensylink offers only states halogen free, instead of lead free.

2. Marking ID includes 2 rows of characters. In general, the 1st row of characters are part number, and the 2nd row of characters are date code plus production information.

1) *Generally, date code is represented by 3 numbers. The number stands for year and work week information. e.g. 501stands for the first work week of year 2015;621 stands for the 21st work week of year 2016.*
2) *Right after the date code information, the next 2-3 numbers or letters are specified to stands for supplier or production location information.*

3. For very small outline package, there's 4 digits to stands for product information and date code, first 2 digits represent product code, and the other 2 digits stands for work week

Absolute Maximum Ratings (Note 3)

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC} to GND	-0.3 to 5.5	V
I/O pin Voltage	V_{IO} to GND	-0.3 to 5.5	V
Operation junction temperature	T_J	-50 to 150	ºC
Storage temperature Range	T_{STG}	-65 to 150	ºC
Lead Temperature (Soldering, 10 Seconds)	T_{LEAD}	260	ºC
ESD MM	ESD_{MM}	600	V
ESD HBM	ESD_{HBM}	6000	V
ESD CDM	ESD_{CDM}	1000	V

Note 3

1. *Stresses greater than those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at the "Absolute Maximum Ratings" conditions or any other conditions beyond those indicated under "Recommended Operating Conditions" is not recommended. Exposure to "Absolute Maximum Ratings" for extended periods may affect device reliability.*
2. *Using 2oz dual layer (Top, Bottom) FR4 PCB with 4x4 mm^2 cooper as thermal PAD*

Recommended Operating Conditions

Parameter	Symbol	Value	Unit
Supply Voltage	V_{CC}	2.5 ~ 5.0	V
Ambient Operation Temperature Range	T_A	-55 ~ +125	°C

Electrical Characteristics (Note 4)

Test Conditions: V_{CC} = 3.0V to 5.0V, T_A= -10 to 85°C unless otherwise specified. All limits are 100% tested at T_A=25°C.

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Supply Voltage	V_{CC}		2.5		5.0	V
Temperature Accuracy	T_{AC}	T_A = -10 to 85°C	-0.5		0.5	°C
		T_A = -10 to 100°C	-1.0		1.0	°C
		T_A = -55 to 125°C	-2.0		2.0	°C
Temperature Resolution				0.0625		°C
Average Operating Current	I_{AOC}	V_{IN} = 3.3V, during Temperature conversion		30		uA
Shutdown Current	$I_{SHUTDOWN}$	Idle, not temperature conversion		1.25	2.5	uA
Conversion time	t_{CON}	From active to finish completely		30		ms
Digital Interface						
Logic Input Capacitance	C_{IL}	I/O pin		20		pF
Logic Input High Voltage	V_{IH}	I/O pin	0.7*VCC		VCC	V
Logic Input Low Voltage	V_{IL}	I/O pin	0		0.2*VCC	V
Logic Input Current	I_{INL}	I/O pin	-2.0		2.0	uA
Communication Timing						
Single-Wire Communication Clock	T_{CLK}			12		us
Recovery time	t_{REC}		3.0			us
Time slot for "0" or "1"	t_{SLOT}		4*T+ t_{REC}			us
Device Reset Low Time	t_{RESET}			32*T		us
Device Reset High Response Time	t_{PDH}			2*T		us
Device Reset Low Response Time	t_{PDL}			8*T		us
Device Reset Response Sampling Time	t_{HSP}		2*T		10*T	us
Write '0' Low Time'	t_{W0L}		4*T		8*T	us
Write '1' Low Time'	t_{W1L}		2.0		1*T	us
Read bit Low Time	t_{RL}		2.5		1*T	us
Read bit sampling Time	t_{HSR}		t_{RL}		2*T	us
Memory Program Current	I_{PROG}			4.0		mA
Memory Program Time	t_{PROG}		200	300	400	us
Data Retention	t_{DR}		10			Years

Note 4:

1. *All devices are 100% production tested at Room temperature; All specifications over the automotive temperature range is guaranteed by design, not production tested.*

Part 1	Part 2	Part 3
Reset	ROM Function Command and/or Device Function Command	Data Tx/Rx
1. Device Reset	1. ROM Function Command, including: 1). Read ROM, 0x33 2). Match ROM, 0x55 3). Search ROM, 0xF0 4). Search Alarm, 0xEC 5). Skip ROM, 0xCC 2. Device Function Command, including: 6). Write Scratchpad, 0x4E 7). Read Scratchpad, 0xBE 8). Temperature Conversion, 0x44 9). Read Power Mode, 0xB4 10). Recall Memory, 0xB8 11). Copy Scratchpad, 0x48	Including, 1. Read Data from the chip, or 2. Write Data into register or memory.

Figure 3. Single-Wire Communication Protocol Operation Diagram

8 bits	48 bits	8 bits
CRC Code [$X^8 + X^5 + X^4 + 1$]	Serial Number [48-bit, factory trimmed]	Family Code [0x28]
MSB LSB	MSB LSB	MSB LSB

Figure 4. 64-bit ROM ID Definition

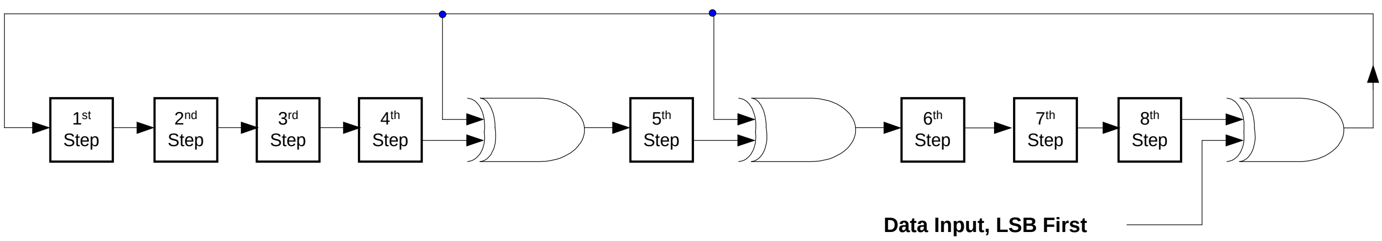


Figure 5. 8-bit CRC Code Generator Diagram

Byte Address	*Attribution	Scratchpad Register Definition
0x00	R	Temperature Data, LSB Byte, [0x50]
0x01	R	Temperature Data, MSB Byte, [0x05]
0x02	R/W	∗Upper Alarm trigger Temperature, TH
0x03	R/W	∗Lower Alarm trigger Temperature, TL
0x04	R	∗Configuration Register
0x05	R	∗RFU Byte0
0x06	R	∗RFU Byte1
0x07	R	∗RFU Byte2
0x08	R	8-bit CRC Code

Figure 6. Scratchpad Register map

**Note:*

1. These registers' data can be changed by writing scratchpad command, after power on reset, the data will reload from memory.

2. in Attribution column, R means read only; R/W means readable/writable.

3. RFU, means Reserved For User.

1 Function Descriptions

The chip can sense temperature and convert it into digital data by a 12-bit ADC. Also the chip supports user-programmable upper/lower trigger temperature settings. Single-Wire interface is compatible with 1-wire and the protocol shown in Figure-3. Generally, one complete communication with host, like MCU, should include Part1, Part2 and Part3. For Search ROM, Search Alarm command, it is an exception. The device supports 5 ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Conditional Search Alarm ROM, 5) Skip ROM, and 6 Device Function commands: 1) Read Scratchpad, 2) Write scratchpad, 3) Temperature Conversion, 4) Read power mode, 5) Recap Memory and 6) Copy Scratchpad. First, the host issues device reset, then above ROM function commands; then after successful completion, the chip can be accessed via above Device function commands by the host. CT1820 can be powered by the local power supply; it can also be powered from the communication line, which is called parasitic power supply.

1.1 Digital Temperature Data

The major function of the chip is to measure temperature. The A-to-D converter resolution of the sensor is 12 bit, corresponding to 0.0625°C. The CT1820 powers up in a low-power idle state. To initiate a temperature measurement and A-to-D conversion, the host has to issue a Temperature Conversion command [0x44h]. After the conversion, the temperature data is stored in the 2-byte temperature register in the scratchpad memory, and then the chip returns to idle state. The temperature data is stored in the temperature register as a 16-bit sign-extended two's complement format in degrees Celsius. The sign bits(S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 1 and Table 2 show examples of digital output data and the corresponding temperature (°C). The default temperature data is 85°C after power-on reset.

Table 1. 12-bit Temperature Data

Temperature (°C)	12-bit Digital Output (HEX)	12-bit Digital Output (BIN)
+125.0000	0x07D0	0 0 0 0, 0 1 1 1, 1 1 0 1, 0 0 0 0
+85.0000	0x0550	0 0 0 0, 0 1 0 1, 0 1 0 1, 0 0 0 0
+25.0625	0x0191	0 0 0 0, 0 0 0 1, 1 0 0 1, 0 0 0 1
+10.1250	0x00A2	0 0 0 0, 0 0 0 0, 1 0 1 0, 0 0 1 0
+0.5000	0x0008	0 0 0 0, 0 0 0 0, 0 0 0 0, 1 0 0 0
0.0000	0x0000	0 0 0 0, 0 0 0 0, 0 0 0 0, 0 0 0 0
-0.5000	0xFFF8	1 1 1 1, 1 1 1 1, 1 1 1 1, 1 0 0 0
-10.1250	0xFF5E	1 1 1 1, 1 1 1 1, 0 1 0 1, 1 1 1 0
-25.0625	0xFE6F	1 1 1 1, 1 1 1 0, 0 1 1 0, 1 1 1 1
-55.0000	0xFC90	1 1 1 1, 1 1 0 0, 1 0 0 1, 0 0 0 0

Table 2. Temperature Data in Register

	Byte Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSB	0x00	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
MSB	0x01	S	S	S	S	S	2^6	2^5	2^4

1.2 Temperature Alarm Setup, TH & TL

Temperature conversion results will be automatically compared with the alarm value programmed by user to determine whether there is an alarm condition. Alarm threshold also uses 2's complement format with 8-bit (7 bits data + 1 bit sign, S). Alarm temperature is set to 1°C increments. If the temperature conversion result is greater than or equal to TH value or less than the TL value (see Figure 6), it will generate an alarm flag on single-wire bus. After a temperature alarm, the device will respond with a Search Alarm command. If the result of the subsequent temperature conversion value is within the TH and TL defined range, the alarm condition is removed. The TH and TL registers data are stored with nonvolatile Memory, so they will retain data when the device is powered down. Also TH and TL can be accessed and changed by read or write scratchpad command. However TH and TL data are loaded from memory every time after power on reset.

1.3 ROM Function Command

After the host detects a presence pulse, it can issue a ROM command. These commands are related to each device's unique 64-Bit ROM ID code, allowing multiple Single-Wire devices connected on a single bus line and operated accordingly. These commands also allow the host to detect the number of Single-Wire devices and the types, and whether a device is in the alarm state. Each device supports basic five kinds ROM command (the actual situation relates to the specific model), each command code length is 8-Bit. Before Device Function commands are issued, the host must submit the appropriate ROM Function command. ROM Function command operation flowchart is shown in Figure 12. And there is a brief description for each ROM Function command and the usage.

1.3.1 Read ROM [0x33]

This ROM command only applies to one device on Single-Wire bus. It allows the host to read directly the device's 64-Bit ROM ID code without performing a search ROM. If the command is used for multi-node connections, data conflict is inevitable because each device will respond to this command at same time.

1.3.2 Match ROM [0x55]

This command is followed by 64-ROM ID allowing the host to access a specified Single-Wire device in multi-node connections. Only when the slave device completely matches 64-Bit ROM ID, it will respond to the function command that host issued; all other devices will wait for a reset pulse.

1.3.3 Search ROM [0xF0]

When the system powers up, the host must identify all Single-Wire devices' ROM ID codes on the bus. And therefore the host can determine the number and the type of devices. By repeatedly performing a Search ROM command (Search ROM command followed by bits of data exchange), the host identifies all Single-Wire devices on the bus. If the bus has only one device, you can use the read ROM command to replace Search ROM command. After completion of each ROM search, the host must return to the first step in the command sequence (initialization).

1.3.4 Search Alarm [0xEC]

Only those Single-Wire devices with alarm flag respond to this command. This works exactly the same as the search ROM command. This command allows the host to determine which device has generated the alarm (the temperature is higher than TH or lower than TL, etc.). In the same way as Search ROM command, after the completion of the search cycle, the host must return to the first step in the command sequence.

1.3.5 Skip ROM [0xCC]

In a single-node application, the host can use this command to quickly access the device on the bus without issuing identical ROM ID code information, which saves corresponding time instead of sending the 64-Bit ROM ID. However in a multi-node application, if the host wants all devices on the bus to perform the same subsequent function command, the host can also use the Skip ROM command. For example, the host issues a Skip ROM command before a Temperature Conversion [0x44] command; then all the CT1820 devices on the same bus will begin the temperature conversion simultaneously. In this way it saves time for performing the entire temperature measurements and gets the temperature conversion results simultaneously. This example is particularly useful for the analysis of temperature fields. Please note, if user issue a Skip ROM command followed by a Read Scratchpad [0xBE] command (including other read command), this command can only be applied to a single node system; otherwise multiple nodes will respond to the command and therefore cause conflicting communication data.

1.4 Device Function Command

Device Function command is similar as above ROM Function command and the flowchart (Figure 13) describes the implementation of several commands and function command protocol such as read/write scratchpad, start temperature conversion, read power and recall memory commands. And there are brief description for each Device Function command and the usage.

1.4.1 Write Scratchpad [0x4E]

This command allows the host to write 3-bytes of data into CT1820 scratchpad register. The first byte is TH register (byte address 0x02), the second byte the TL register (byte address 0x03), and the third byte the configuration register (byte address 0x04). Data is transmitted starting from the least significant bit. Before the host sends a reset signal, these three bytes must be written, otherwise it may lead to an error of incomplete data transfer.

1.4.2 Read Scratchpad [0xBE]

This command allows the host to read the contents of the scratchpad register. Data transmission always starts from byte address 0 (the least significant bit of the temperature register) and continues until finishing the remaining seven bytes of scratchpad register. If the host continues to read, it reads the 9th byte, the 8-Bit CRC. The CRC is generated by CT1820 using the same polynomial in ROM ID CRC generator. CRC is sent in the original format. If only part of the scratchpad register data is needed, the host can send a reset signal to end this reading operation.

1.4.3 Temperature Conversion [0x44]

This command starts the temperature conversion. After the conversion is complete, the measured temperature data will be stored into the scratchpad registers (byte address, 0x00, 0x01.). CT1820 then returns to a low-power idle state. If the device is in "parasitic power mode", the host pulls Single-Wire bus into the strong state (the time value and resolution independent) after sending the command. The host monitors the conversion process in each time slot. When the host reads the logic '1 'instead of '0', it indicates the temperature conversion is complete.

1.4.4 Read Power Mode [0xB4]

When this command is executed, the host always receives 0xFF either in VCC power supply mode or parasitic mode.

1.4.5 Recall Memory [0xB8]

This command reads the TH, TL alarm values and configuration data from backup memory, and copies them to the associated scratchpad register. After this command code is sent, the host sends a read time slot to monitor the recall process. When the host reads “1” instead of “0”, it indicates that the data read back is complete. When the chip powers up every time, the scratchpad register data is automatically loaded from the corresponding memory address as default data shown in above Figure 6.

1.4.6 Copy Scratchpad [0x48]

This command copies the contents of the scratchpad TH, TL and configuration registers Data written by issuing Write Scratchpad command [0x4E] to Memory. If the chip is used in parasite power mode, within 10µs (max) after this command is issued the host must switch to a strong pull-up condition on the Single-Wire bus for at least 10ms.

1.5 CRC generator

CRC (Cyclic Redundancy Check) bytes are provided as part of the chip’s 64-bit ROM ID code and in the 9^{th} byte of the scratchpad register. For the ROM ID code, CRC part is calculated based the first 56 bits and is contained in the most significant byte of the ROM ID code. The scratchpad register CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the host with a method of data validation when data is read from the chip. To verify that data has been read correctly, the host must do calculation the CRC based on the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the host. The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$CRC = X^8 + X^5 + X^4 + 1$

The host can re-calculate the CRC and compare it to the CRC values from the CT1820 using the polynomial generator shown in Figure 5. This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0x00. Starting with the least significant bit of the ROM ID code or the least significant bit of byte 0x00 in the scratchpad, one bit at a time should shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the CT1820 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. The summarization for the CRC generator is shown as below, which can be used for software reference:

a). Data input with LSB first;

b). Polynomial is $CRC = X^8 + X^5 + X^4 + 1$;

c). Data movement is left shift;

d). Initial data always is 0x00;

e). Data length is 8-bit;

When put CRC data as input, the result must be the initial data, 0x00.

1.6 64-bits ROM ID

Each CT1820 contains a unique ROM ID code as serial number. First 8 bits are Single-Wire family code, for this chip, it is 0x28, the next 48 bits are unique serial number, the last 8 bits is the CRC code generated based on previous 56 bits, shown in Figure 4. The shift register is first initialized to 0, then shifts the family code; each time shifted one bit, least significant bit first,. After the 8th bit of family code is shifted, it starts to shift 48 bits serial number. After the last bit of serial number is shifted, the shift register contains the CRC value. After shifting the 8-Bit CRC, all the bits of the shift register are same as initial data, 0x00.

2 Single-Wire Bus

Single-Wire bus system consists of a host and one or more salve devices. In any case, CT1820 are slave devices. The bus host could be a microcontroller or SoC. Discussion of Single-Wire bus system is divided into three parts: the hardware configuration, the operation sequence and Single-Wire timing.

2.1 Hardware Configuration

According to the definition of Single-Wire bus system, it has only one data line physically. In order to facilitate this, each device on the bus needs to have open-drain or tri-state output, and CT1820's Single-Wire port (I/O pin) uses an open-drain output. A typical circuit is shown in above Figure 1.

Multi-node system consists of a Single-Wire host and multiple slave devices. CT1820 supports 9.5kbps (default rate) of fixed and variable communication rate. Pull-up resistor depends primarily on the number of nodes, the communication distance and the line load. For example with the communication distance of less than 20cm, a single node and an independent power supply condition, CT1820 requires an external 4.7kΩ (typical) pull-up resistor. If the communication distance is greater than 30m, you need a 1.0k or smaller pull-up resistor even with single-node and an independent power supply, Single-Wire bus idle state is high. If for some reason the device needs suspend temporally and then return to work, it must be placed on the bus idle state.

2.2 Operation Sequence

To access CT1820 through Single-Wire port, the complete procedure is shown in previous Figure 3, it includes:

- Part 1, Device Reset, refer to below section for description in detail.
- Part 2, bus function command, including ROM Function command and Device Function command. In most cases, Device Function command is followed by ROM Function command. Sometimes, ROM Function command can be used independently without Device Function command, like Search ROM.
- Part3, Data Receiving/Transmitting, includes receive data from Single-Wire device or send data to Singe-Wire device.

2.3 Device Reset

When the chip is applied power first time, it will perform internal Power on Reset action automatically, reset all registers and configurations as initial state, and recall memory data into scratchpad as default.

All operations of Single-Wire bus always begin with a Device Reset. Device Reset consists of a reset pulse sent by the host and a device responses pulse shown in Figure 13. The presence pulse is used to notify the host that CT1820 is already connected on the bus. When a Single-Wire device sends a response pulse to the host, it tells the host that it is on the bus and ready to work. During the initialization process, host pulls the bus low for t_{RESET} period time, thus produces (Tx) Device Reset pulse. Then, the host releases the bus and goes into receive mode (Rx). When the bus is released, the bus is pulled up by an external pull-up resistor. When a Single-Wire device detects a rising edge, it will remain high for t_{PDH} (2T in typical), then the Single-Wire device generates a presence pulse by pulling the bus low for t_{PDL} (8T in typical). After that the bus is released and pulled back high by the external pull-up resistor, at least keeping the 6T time. Thus, the entire Single-Wire device response cycle is at least t_{PD} (16T in typical). After that, the host can begin to transfer the ROM command. If user need more precise communication time match, the host can measure the Single-Wire device response t_{PDL} (8T in typical) low pulse, and adjust the time of the original Device Reset pulse, t_{RESET}, and the read sampling timing. Once the device successfully captures the communicate reset pulse, it will use it to set the communication speed.

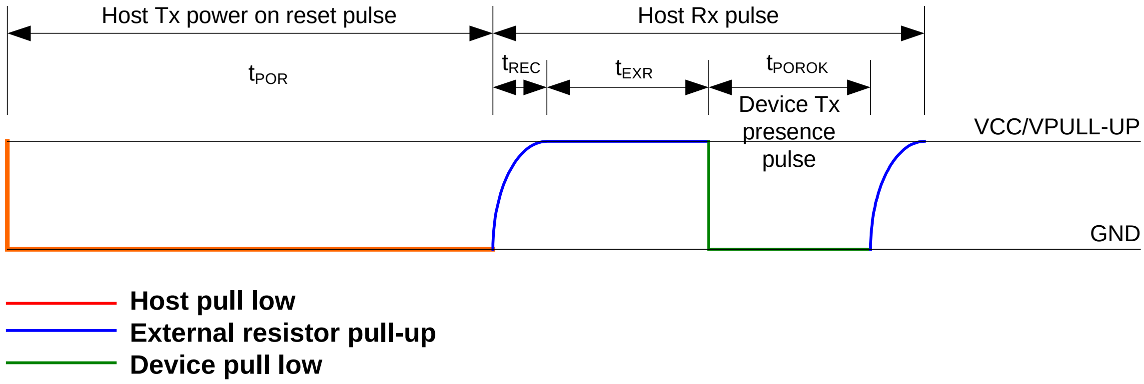


Figure 7 Power on Reset Timing Diagram

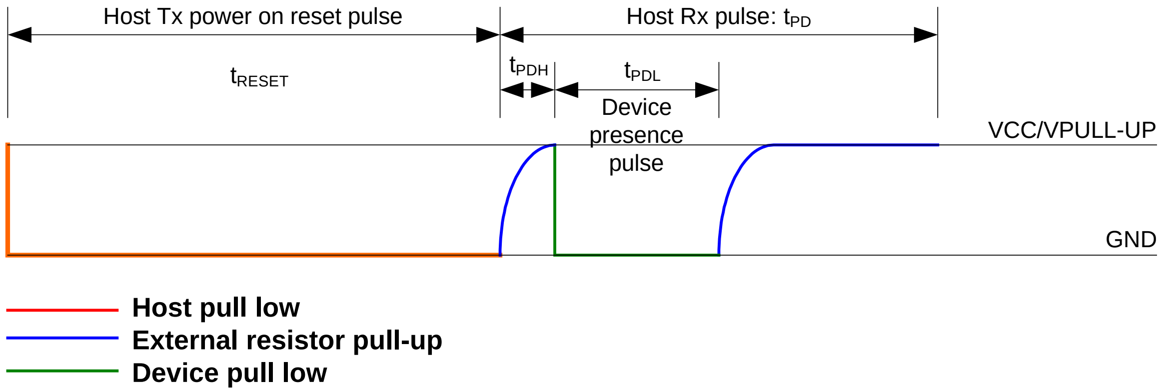


Figure 8 Device Reset Timing Diagram

2.4 Single-Wire Timing

After complete reset successfully on Single-Wire bus, the next step is to perform ROM Function command and/or Device Function command. The following section is to descript the bit transmission. All Single-Wire devices require to strictly comply with Single-Wire communication protocol to ensure data integrity. The protocol defines several signal types: power-on reset pulse, communications reset pulse, presence pulse, write “0”, write “1”, read “0”, and read “1”. All these signals except presence pulse are synchronous signals issued by the host. And all the commands and data are the low byte first which is different from other serial communication format (high byte first).

During Write Time Slot the host writes data to a Single-Wire device; and during the Read Time Slot, the host reads the data from the Single-Wire device. In each time slot, the bus can only transmit one bit data.

Write Time Slot

There are two write time slot modes: write "1" and write "0" slot. The host writes into Single-Wire device "1" by using a write "1" slot, and host write into Single-Wire device "0" by using write "0". All write time slots are at least t_{SLOT} (4*T + t_{REC} in typical), and need the recovery time at least 3μs between two separate time slots. Two kinds of write slots start with pull-down bus by the host shown in Figure 14. To produce a write "1" slot, the host must release the bus within t_{W1L} (<= 1*T) after pulling down for 1us, and then the bus is pulled-up by an external pull-up resistors on the bus. To produce a write "0" slot, after the host is pulling the bus low, it maintains a low level during the entire time slot, that is t_{W0L} (> = 4T). During the write time slots, Single-Wire device samples bus level status at t_{SSR} (2*T in typical) time. If sampling results at this time is high, then the logic "1" is written to the device; If "0", the write logic is "0".

Read Time Slot

Single-Wire device can only transmit data to the host after the host issues read time slots. After the host issues a read data command, a read time slot must be generated in order to read data from the Single-Wire device. A complete read time slot is at least t_{SLOT} (4*T + t_{REC}), and requires at least 3us recovery time between two separate time slots. Each time slot is generated by the host to initiate the read bit, a low level period is required to be at least 1μs shown in Figure 14. Once the device detects a Single-Wire bus low, the device immediately sent bit "0" or "1" on the bus. If Single-Wire device sends "1", the bus is pulled-up high by a pull-up resistor after the short low period; if sent "0", then the bus is keeping low for t_{DRV} (2*T in typical). After that the device releases the bus from pull-up resistors and back to idle high. Therefore, the data issued by Single-Wire device after read time slot at the beginning stay effective during time t_{DRV} (2*T in typical). During the read time slots the host must release the bus, and samples the bus states at 2T after the start of a slot (optimum sampling time point 1T).

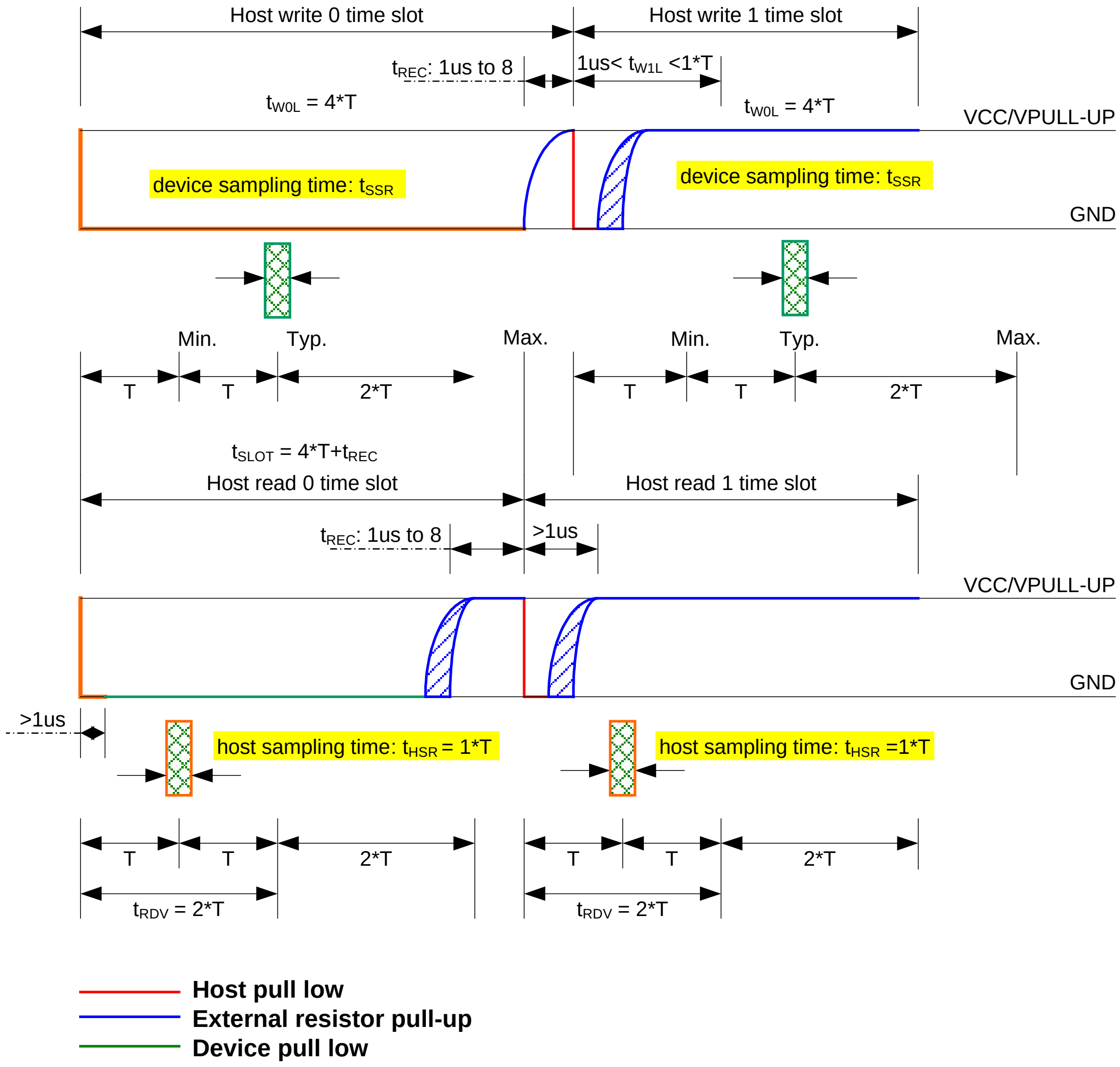
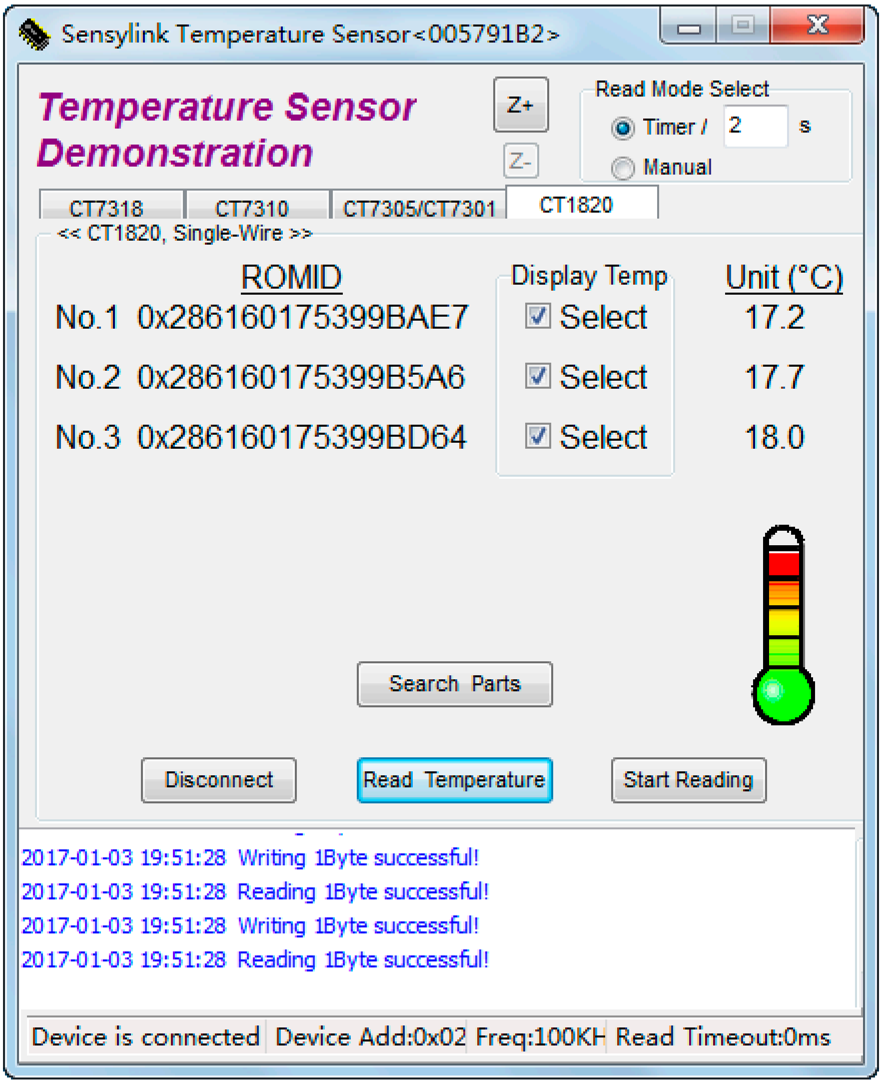
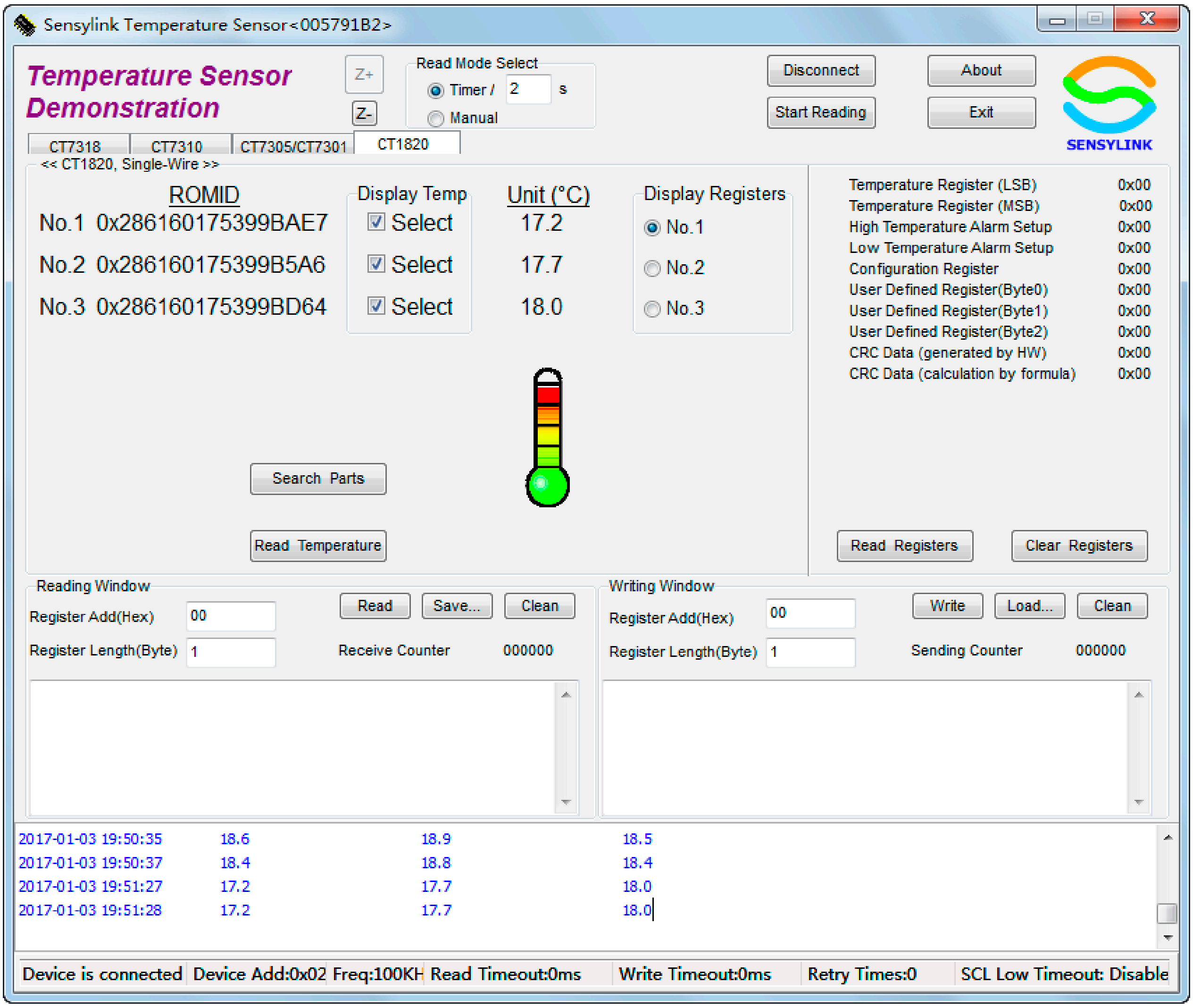


Figure 9 Read/Write Timing Slot Diagram

3 Software Reference Code

Below are windows based GUI of demo application for Sensylink temperature sensor, select CT1820 page, there are 4 Buttons. Press [search parts] button after pressing [connect] button, then, check all CT1820 devices found after search action, then press [Read Temperature] button, temperature data of all devices will display, as shown on the right side. Below lists important function software code based on C++ language, which is a reference for user write SW code.





For more information about software source code support, please contact our sales.

```c
/**********************************************/
uchar CT1820Init()
{
    uchar i;
    DIOPORT = 0;           //pull-low line
    delay600us();          //delay 450us to 650us
    DIOPORT = 1;           //pull-high line
    i = 0;  //
    while(DIOPORT)   //waiting for CT1820 pull-low line, once CT1820 give response.
    {
        delay500us();
        i++;
        if(i>1)     //if waiting time > 5ms
        {
            return 0; //Return 0, initialization fail
        }
    }
    return 1;       // Return 1, initialization success
}

void CT1820_Write_Byte(uchar dat)
{
    uint j;
    for(j=0; j<8; j++)
    {
        DIOPORT = 0;          //pull-low line with 1us
        i++;
        delay7us();
        DIOPORT = dat & 0x01;  //write one-bit data with LSB in first
        delay50us();
        DIOPORT = 1;   //release single-wire line， to be ready for next byte
        dat >>= 1;
    }
}

uchar CT1820_Read_Byte()
{
    uchar byte, bi;
    uint i, j;
    for(j=8; j>0; j--)
    {
        DIOPORT = 0;//pull-low line with 1us
        DIOPORT = 1;//then release line
        bi = DIOPORT;     //Read Data from line, LSB in first
```

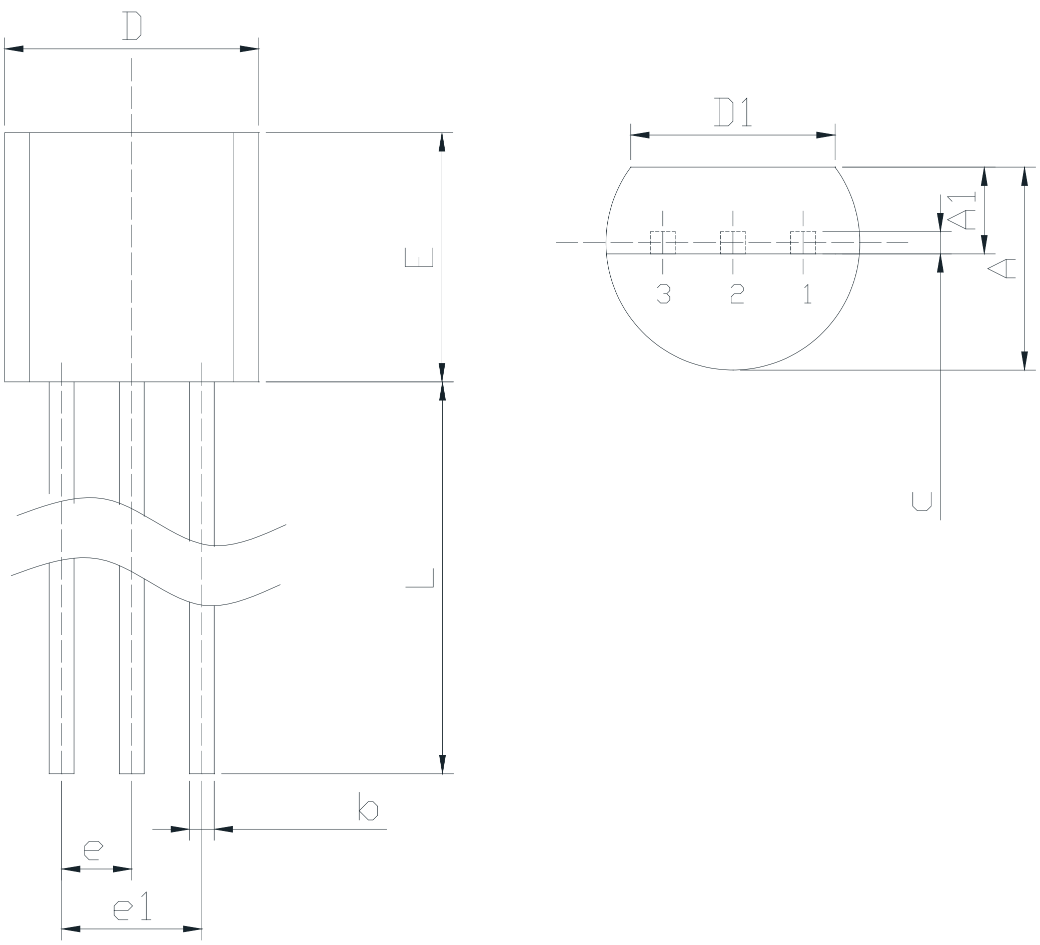
```
            /*move byte 1-bit to right， & move bi 7-bit to left*/
            byte = (byte >> 1) | (bi << 7);
            delay48us();
        }
        return byte;
}
void  CT1820_Temp_Conv()
{
        CT1820Init();
        CT1820_Write_Byte(0xcc);      // make the chip ready command
        delay50us();
        CT1820_Write_Byte(0x44);      // Temp converter command
        delay50ms();
        }

void  CT1820_Read_Temp_Com()
{
        CT1820Init();
        delay200us();
        CT1820_Write_Byte(0xcc);   // make the chip ready Command
        delay200us();
        CT1820_Write_Byte(0xbe);   // Read temperature command
}

int CT1820_Read_Temp_Degree()
{
        int temp = 0;
        uchar tmh, tml;
        CT1820_Temp_Conv();             //Send Temp converter command, 0x44
        CT1820_Read_Temp_Com();       //Read Temp
        delay200us();
        tml = CT1820_Read_Byte();      //Read LSB for Temperature in first
        delay200us();
        tmh = CT1820_Read_Byte();      //Then read MSB, Reg1_T_MSB
        temp = tmh;
        temp <<= 8;
        temp |= tml;
        return temp;
}
```

Package Outline Dimensions (TO-92)

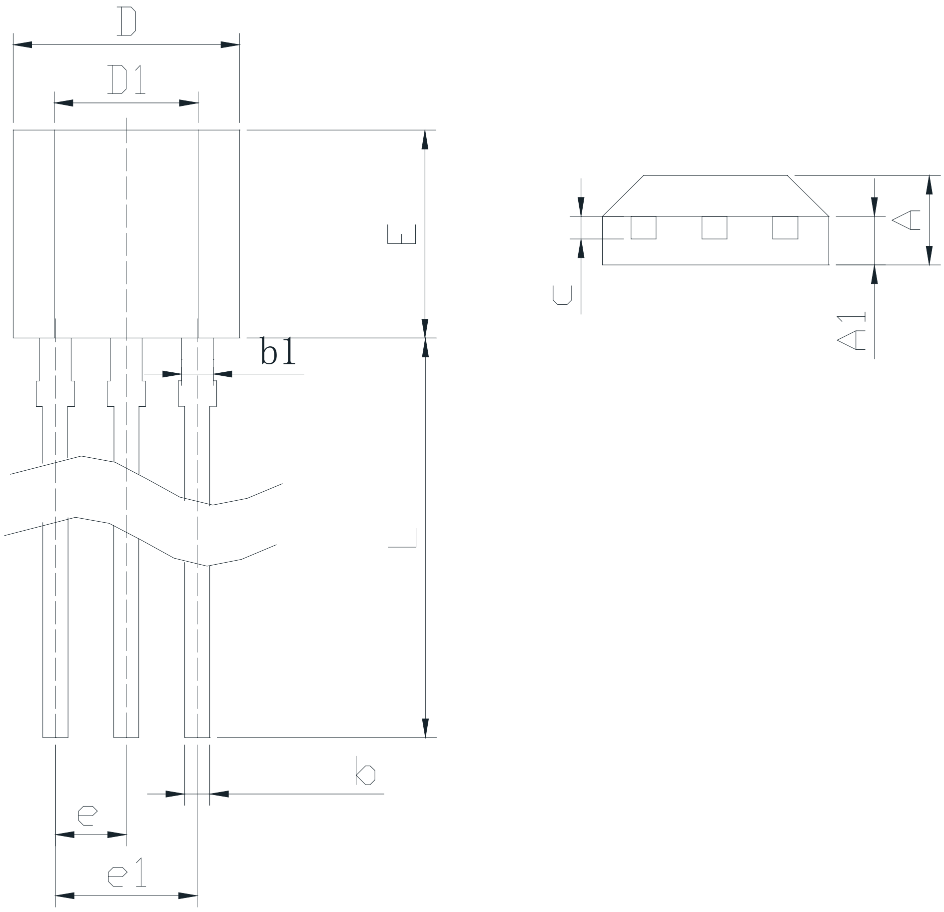
TO-92 Unit (mm)



Symbol	Dimensions in Millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	3.300	3.800	0.130	0.150
A1	1.100	1.400	0.043	0.055
b	0.380	0.550	0.015	0.022
c	0.300	0.510	0.012	0.020
D	4.300	4.700	0.169	0.185
D1	3.430		0.014	
E	4.300	4.700	0.169	0.185
e	1.270（TYP）		0.050（TYP）	
e1	2.540（TYP）		0.100（TYP）	
L	13.000	15.000	0.512	0.590

Package Outline Dimensions (TO-92S)

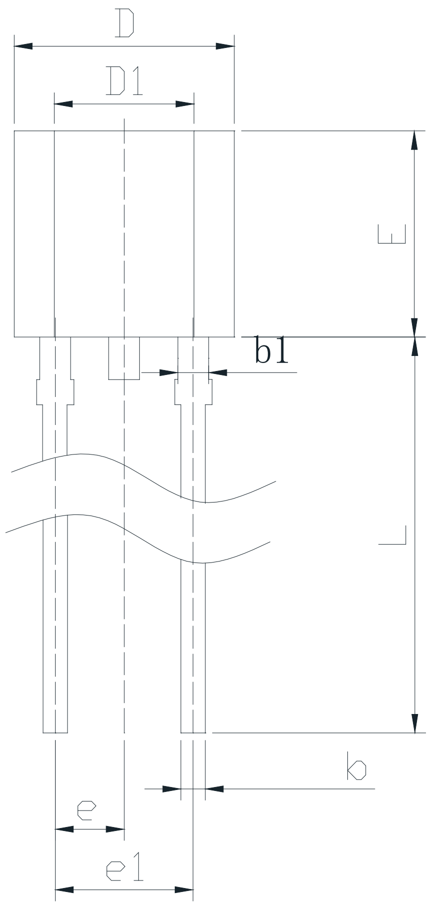
TO-92S Unit (mm)

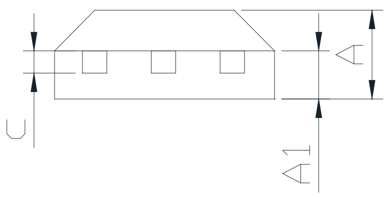


Symbol	Dimensions in Millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	1.420	1.620	0.056	0.064
A1	0.660	0.860	0.026	0.034
b	0.330	0.480	0.013	0.019
b1	0.400	0.510	0.016	0.020
c	0.330	0.510	0.013	0.020
D	3.900	4.100	0.154	0.161
D1	2.280	2.680	0.090	0.106
E	3.000	3.300	0.118	0.130
e	1.270（TYP）		0.050（TYP）	
e1	2.540（TYP）		0.100（TYP）	
L	15.100	15.500	0.594	0.610

Package Outline Dimensions (TO-92S-2)

TO-92S-2 Unit (mm)

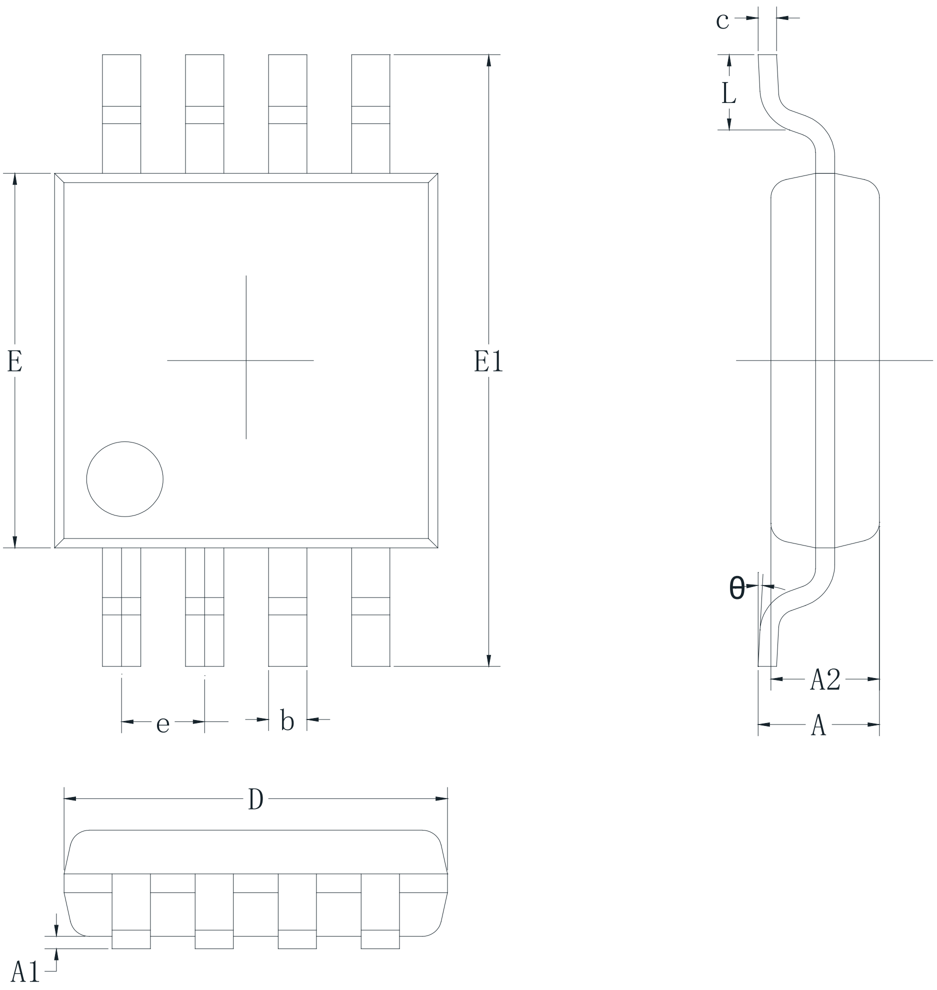




Symbol	Dimensions in Millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	1.420	1.620	0.056	0.064
A1	0.660	0.860	0.026	0.034
b	0.330	0.480	0.013	0.019
b1	0.400	0.510	0.016	0.020
c	0.330	0.510	0.013	0.020
D	3.900	4.100	0.154	0.161
D1	2.280	2.680	0.090	0.106
E	3.000	3.300	0.118	0.130
e	1.270（TYP）		0.050（TYP）	
e1	2.540（TYP）		0.100（TYP）	
L	15.100	15.500	0.594	0.610

Package Outline Dimensions (MSOP-8)

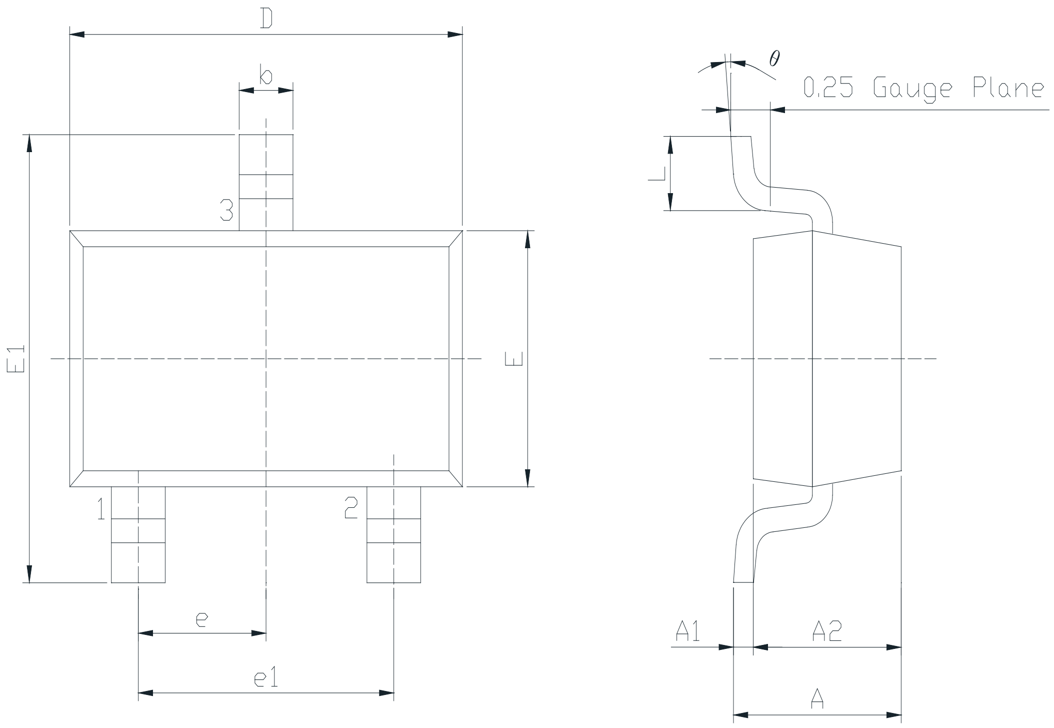
MSOP-8 **Unit (mm)**



Symbol	Dimensions in Millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	0.820	1.100	0.032	0.043
A1	0.020	0.150	0.001	0.006
A2	0.750	0.950	0.030	0.037
b	0.250	0.380	0.010	0.015
c	0.090	0.250	0.004	0.010
D	2.900	3.100	0.114	0.122
E	2.900	3.100	0.114	0.122
E1	4.700	5.100	0.185	0.201
e	0.650（BSC）		0.026（BSC）	
L	0.400	0.800	0.016	0.031
θ	0°	8°	0°	8°

Package Outline Dimensions (SOT-23)

SOT-23 **Unit (mm)**



Symbol	Dimensions in Millimeters		Dimensions in Inches	
	Min.	Max.	Min.	Max.
A	0.900	1.150	0.035	0.045
A1	0.000	0.100	0.000	0.004
A2	0.900	1.050	0.035	0.041
b	0.300	0.500	0.012	0.020
c	0.080	0.200	0.003	0.008
D	2.800	3.000	0.110	0.118
E	1.200	1.400	0.047	0.055
E1	2.200	2.600	0.087	0.102
e	0.950（BSC）		0.037（BSC）	
e1	1.800	2.000	0.071	0.079
L	0.300	0.600	0.012	0.024
θ	0°	8°	0°	8°

SENSYLINK

SENSYLINK Microelectronics Co.,Ltd

www.sensylink.com

IMPORTANT NOTICE