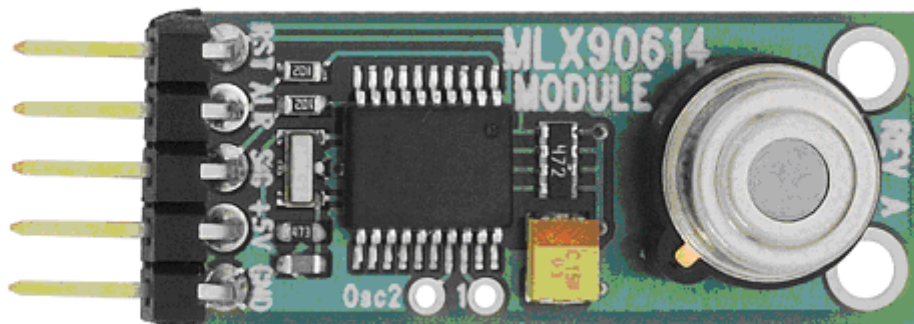


MLX90614 Infrared Thermometer Module (#28040) - PRELIMINARY

The MLX90614 Infrared Thermometer Module is an intelligent non-contact temperature sensor with a single pin serial interface for connection to most microcontrollers. The MLX90614 sensor is designed for non-contact temperature measurements of objects placed within the sensor's cone of detection. The sensor is comprised of an integrated ASIC and infrared sensitive thermopile detector. The sensor communicates with an SX20AC/SS-G coprocessor over a digital SMBus, which Parallax has programmed to simplify an otherwise fairly complex communication protocol.

Users of this sensor should also review the MLX90614 datasheet from www.melexis.com. The datasheet includes technical information pertaining to the sensor's cone of detection, precision and applications. When reviewing the datasheet please note that Parallax is using Melexis P/N MLX90614ESF-AAA, the single zone infrared thermometer sensor.



Features

- Outputs continuous data flow with an active alarm running in background
- 16-bit digital temperature output data, ranging from -70 to +380 °C
- Auto-baud detection (2400, 4800, 9600, 19.2K, 38.4K) for microcontroller-to-MLX90614 communications
- SIP module format fits easily in breadboards or through-hole prototype areas
- Multiple modules can be connected to a single I/O pin for serial networking
- Module can act as a stand alone sensor for alarming control
- Sleep setting for low power consumption
- Starts up active without pre-programming using preset writeable defaults

Example Applications

- Sensing and measuring surface temperatures without surface contact
- Alarming alert for over temperature conditions
- Temperature sensing element for residential, commercial and industrial building air conditioning
- Industrial temperature control of moving parts such as die protection
- Human or animal presence
- Movement detection
- Multiple zone temperature control – up to 100 sensors can be read via common 2 wires
- Thermal relay and alert
- Body temperature measurement

Key Specifications

- Supply voltage: +5 VDC
- Supply current: 20 mA typical; 5 mA in sleep mode
- Package: 5-pin SIP, 0.1" spacing
- Operating temperature: 0 to 70 °C.
- Dimensions: 6.5 mm H x 13.8 mm W x 40.4 mm T (0.26 in x 0.54 in x 1.59 in)
- Weight: 7 g (0.25 oz)

Connections and Pin Definitions

The BASIC Stamp 2 is used as the example processor in this documentation. The MLX90614 module mounts into a socket terminal or breadboard. Connection to the BASIC Stamp module is made via five connections using 3 I/O pins. Before installing the MLX90614 module, you should perform the following steps:

1. Clear the current BASIC Stamp program to ensure that all pins are set to inputs. The simplest program to do this contains just one statement:

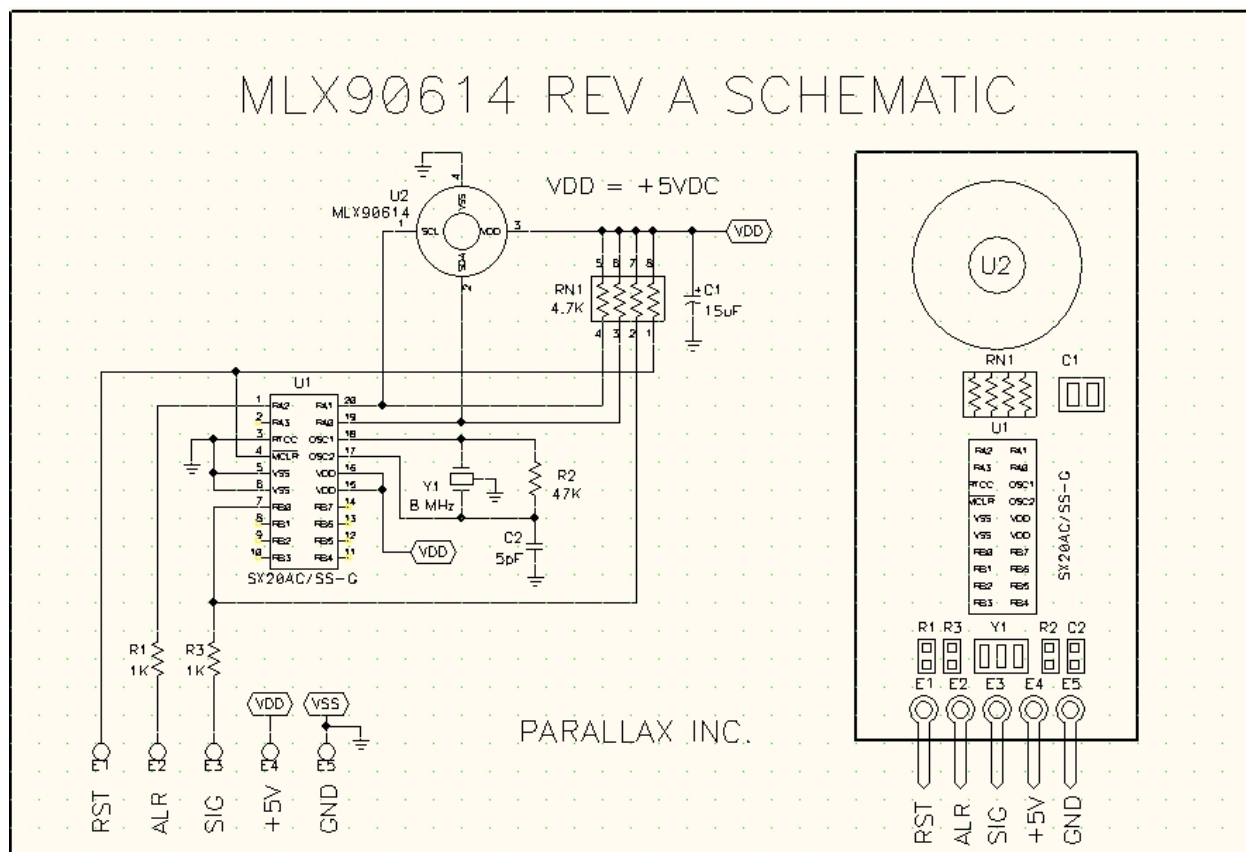
```
END
```

2. Disconnect power from the BASIC Stamp circuit.
3. Connect five separate wires between the MLX90614 module and the BASIC Stamp module. Additional modules maybe connected to the serial bus provided each has a different slave address.
4. Reconnect power to the BASIC Stamp circuit and program as desired.

The table below lists the MLX90614 module / BASIC Stamp pin connections:

MLX90614 Module	Function	BASIC Stamp
GND	Ground link between microcontroller / MLX90614 Module	Gnd
5V	Voltage link between microcontroller / MLX90614 Module	5V (V _{DD})
SIG	Serial I/O link between microcontroller / MLX90614 Module	Any I/O pin
ALR	Alarm control output, high on alarm	optional
RST	Reset link to coprocessor, pull low to high for reset	optional

Schematic



How it Works

Using a Parallax SX20AC/SS-G coprocessor, the MLX90614 module receives serial commands from the BASIC Stamp or other microcontroller, then configures and controls the MLX90614 sensor. Resulting output data is directed back through the same I/O pin. The READ and WRITE commands conform to those indicated in the MLX90614 datasheet for writing to the SMBus, an I²C style format

The MLX90614 sensor has an internal EEPROM, where sensor addresses (for using multiple sensors on a single bus), alarm values and data transmission intervals are set.

The SMBus protocol between the thermopile sensor and the SX20AC/SS-G coprocessor includes a byte for the sensor's packet error code (PEC) calculation. The PEC value isn't a linear value that shows data integrity, but a number calculated with a CRC-8 procedure to show whether or not the data is valid which results in a "pass" or "fail". Once the correct communication is established between a microcontroller and MLX90614 module it is not likely to have a failed PEC calculation unless the module is removed from the circuit during operation.

Upon power up the MLX90614 module will be reading the temperature data held in RAM address \$07. There is a minimum 1 ms delay between each temperature reading. In the non-sleep operation mode the timing is adjustable from 1 to 2550 ms and can be set in the low data byte of the reading time interval command address (\$21). In the sleep operation mode the timing is adjustable from 1 to 255 seconds and can be set in the high data byte of the reading time interval command address (\$21). It takes approximately 1 ms to perform a temperature reading. Interval timing for non-sleep and sleep operation modes have been factory set at 50 ms and 1.15 seconds, respectively.

The MLX90614 module's alarm can be set between -70 °C and +380 °C in the low and high data bytes of the alarm command address (\$20). The factory default setting is 100 °C.

References

1. Melexis MLX90614 datasheet, No. 390109614 Rev. 003, 03 Oct 2007
2. Melexis SMBus communication with MLX90614, No. 390119061402 Rev 001, Oct 2006
3. Melexis Understanding MLX90614 on-chip digital filters, No. 390119061406 Rev 001, Apr 2007

Communication Protocol

All MLX90614 commands begin with the string:

"!ITEM"

The "!" is used for synchronization and baud rate detection; the "TEM" identifies the device type – a thermopile sensor. What follows will generally be a command letter and unit identifier. To alert a new command the SEROUT statement must hold the BASIC Stamp pin low for a short period by sending a 0 byte preceding the command.

The commands may be put in RAM (while running) or EEPROM (permanently, requires a power cycle). Use "!ITEMW" to write any of these parameters to ROM.

Command	Function	Writes to / Command	MLX90614 Module Replies With
"!ITEMR"	Read Sensor Data	N/A (reads any command)	Replies with three bytes (temperature low byte, high byte and PEC value)
"!ITEMW"	Write Sensor Data	EEPROM / \$20, \$21, or \$2E	No reply
"!ITEMC"	Set Sensor for Continuous Data Reading	RAM / \$07	No reply
"!ITEMc"	Set Sensor for Continuous Data Reading and Output	RAM / \$07	Replies with "TEM" and five more bytes (slave address, low temperature byte, high temperature byte, PEC).
"!ITEMZ"	Set Sensor for Continuous Data Reading at Low Power	RAM / \$07	No reply.
"!ITEMz"	Set Sensor for Continuous Data Reading and Output at Low Power	RAM / \$07	Replies with "TEM" and five more bytes (slave address, low temperature byte, high temperature byte, PEC).
"!ITEMI"	Set Sensor Reading Interval Time	RAM / \$07	No reply
"!ITEMA"	Set Sensor Alarm	RAM / \$07	No reply
"!ITEMS"	Stop Sensor Continuous Data	RAM (no command)	No reply
"!IID"	Get the Version Number	N/A	Replies with three bytes of string

"R" Read Sensor Data

```
SEROUT pin, baudrate, [0,"!ITEMR",slaveaddr,commandaddr]
```

The baud rate is always a variable or constant value for 2400, 4800, 9600, 19.2K or 38.4K baud. The "R" command will read data in the sensor at the specified slaveaddr location. The \$07 holds the location of the sensor's temperature data. After sending this command the BASIC Stamp must use SERIN to retrieve the three data bytes.

Example:

```
SEROUT pin, baudrate,[0,"!TEMR",$5A,$07]
SERIN pin, baudrate,[lowbyte,highbyte,pec]
```

The example sets a READ request at sensor address \$5A and serially transmits object temperature data stored in RAM (\$07). The lowbyte and highbyte represent the 0.02 counts/K temperature. The PEC byte can be used in the PEC calculation should the user desire.

"W" Write Sensor Data

```
SEROUT pin, baudrate, [0,"!TEMW",slaveaddr,commandaddr,lowdatabyte,highdatabyte]
```

The "W" command will write the data into the module's specified EEPROM address. This is permanent until rewritten and requires a power cycle on the module. See available command addresses in the table below:

EEPROM Address	Purpose	Low Byte	High Byte	Default
\$20	Temperature Alarm	count word*		100 °C (186575 counts)
\$21	Read Time Intervals	count X 10 ms (non sleep)	count x 1.15 seconds (sleep)	50 ms non sleep and 1.15 seconds sleep
\$2E	Slave Address and Setup State	\$7F>= Slave address =>\$01	"C", "c", "Z", "z", or "S"	\$5A slave address and "C"

*count word x 0.02 = K

Example:

```
SEROUT pin, baudrate,[0,"!TEMW",$5A,$2E,$1A,"z"]
PAUSE 50 'minimum time allowed to write to sensor EEPROM
```

This command always writes four bytes. The example will write a new slave address (\$1A) in an EEPROM location (\$2E) and set a sleep mode ("z") for continuous data output. A 50 ms delay lets the module erase and write to sensor EEPROM before advancing to the next PBASIC statement.

"C" Set Sensor for Continuous Data Reading

"c" Set Sensor for Continuous Data Reading and Output

```
SEROUT pin, baudrate, [0,"!TEMc",slaveaddr, commandaddr]
```

The "C" command will cause the module to read object temperature data continuously. No data will be serially transmitted back. The "c" command will cause the module to serially transmit sensor data continuously. It is necessary to read data continuously to check for over alarm setting. "C" is the default command on power up. To set the default command a "C" or "c" can be written into the high byte of the slave address EEPROM (\$2E).

Example:

```
SEROUT pin, baudrate,[0,"!TEMc",$5A,$07]
Again:
SERIN pin, baudrate,[Wait{"TEM"),slave,lowbyte,highbyte,pec]
GOTO AGAIN
```

The example reads and transmits the slave address (\$5A), object temperature low and high data bytes, and PEC byte stored in the sensor RAM. Note that the "TEM" string indicates the start of the data output and must be included in data readings. This allows you to differentiate multiple sensors on a single bus.

Note: Care must be taken in setting multiple sensors up for continuous output. Be sure your interval settings are such that multiple sensors are not transmitting at the same time or you will experience data collision making the values unreadable.

```
SEROUT pin, baudrate,["!TEMC",$5A,$07]
```

This example reads the slave address (\$5A), object temperature low and high data bytes, and PEC byte stored in the sensor RAM (\$07) but does not transmit any data.

"Z" Set Sensor for Continuous Data Reading at Low Power

"z" Set Sensor for Continuous Data Reading and Output at Low Power

```
SEROUT pin, baudrate, [0,"!TEMZ","",slaveaddr,commandaddr]
```

The "Z" command will cause the module to read object temperature data continuously. No data will be serially transmitted. The "z" command will cause the module to serially transmit sensor data continuously. It is necessary to read data continuously so check for over alarm setting. The temperature reading is performed internally every 1152 ms and the data transmitted based on specified interval time. The "Z" and "z" commands will reduce the module power from a normal 20 ma to about 5 mA. Power to the MLX90614 sensor is maintained at 5 volts. Only the coprocessor is placed in a pulse sleep mode for low power effect. To change the startup default a "Z" or "z" can be written into the high byte of the slave address EEPROM (\$2E).

Example:

```
SEROUT pin, baudrate,[0,"!TEMz",$5A,$07]
Again:
  SERIN pin, baudrate,[Wait{"TEM"},slave,lowbyte,highbyte,pec]
  GOTO AGAIN
```

The example reads and transmits the slave address (\$5A), object temperature low and high data bytes, and PEC byte stored in the sensor RAM (\$07). Note that the "TEM" string indicates the start of the data output and is included in data readings. To exit from this command the reset pin on the module must be held in a low state for 5 ms and then put in a high input state for 50 ms. (i.e, low pin, pause 5, input pin, pause 50)

Note: Care must be taken in setting multiple sensors up for continuous output. Be sure your interval settings are such that multiple sensors are not transmitting at the same time or you will experience data collision making the values unreadable.

```
SEROUT pin, baudrate,["!TEMZ",$5A,$07]
```

If followed by a SERIN statement, this example reads the slave address (\$5A), object temperature low and high data bytes, and PEC byte stored in the sensor RAM (\$07) but does not transmit any data.

"I" Set Sensor Reading Interval Time

```
SEROUT pin, baudrate, [0,"!TEMI",slaveaddr,intervalnonsleep,intervalsleep]
```

The "I" command sets the frequency at which data is sent out in the non-sleep ("c", continuous) and sleep ("z") modes. No data will be serially transmitted back by this command. Using this command will not change interval time setting written in the EEPROM (\$21) and will make those settings inactive unless the sensor module is power cycled. The interval time byte in non-sleep operation can be set between 1 to 255 and is equal 10 ms/count. A zero in this byte will result in a 1 ms time interval. The interval time byte in sleep operation can be set between 1 to 255 to equal 1150 ms/count. Upon powering up or performing a reset the EEPROM (\$21) interval time settings will be restored.

Example:

```
SEROUT pin, baudrate,[0,"!TEMI",$5A,10,4]
```

The example will set the module non-sleep and sleep reading interval time bytes at 100 ms and 4.6 seconds (4 x 1.15 seconds), respectively.

"A" Set Sensor Alarm

```
SEROUT pin, baudrate, [0, "!TEMA", slaveaddr, alarmlowbyte, alarmhighbyte]
```

The MLX90614 module has an alarm pin shown by the silkscreen ALR. This pin is high when the alarm value temperature is exceeded. Use the "A" command to send the module new sensor alarm settings. No data will be serially transmitted back from the module when the alarm value is set. The new alarm value takes effect immediately, however power must be cycled on the MLX90614 module in order for the new alarm settings to take effect in the EEPROM (\$20).

Example:

```
Alarm VAR WORD
Alarm = 18650
SEROUT pin, baudrate,[0,"!TEMA",$5A,alarm.lowbyte,alarm.highbyte]
```

Suppose you want to set the alarm for 373 K (100 °C). The sensor functions in units of 0.02 counts/K. For example, $373 / 0.02 = 18,650$ counts. This value is sent to the module in two pieces - alarmlowbyte and alarmhighbyte. When 100 °C is exceeded the ALR pin will be high.

"S" Stop Sensor Continuous Data

```
SEROUT pin, baudrate, [0,"!TEMS",slaveaddress]
```

The "S" command will stop continuous temperature data from the module. To change the startup default a "S" can be written into the high byte of the slave address EEPROM (\$2E).

Example:

```
SEROUT pin, baudrate,[0,"!TEMS",$5A]
```

The example above sends a stop request at the slave address (\$5A) to quit reading data.

"!ID" Get the Version Number

```
SEROUT pin, baudrate, [0,"!ID", slaveaddress]
```

The "ID" command will cause the module to return a 3 byte version number.

Example:

```
LOW 1:PAUSE 5:INPUT 1:PAUSE 50          'reset module
SEROUT pin, baudrate, [0,"!ID", $5A]    'read ID
SERIN pin, baudrate, 500, OVER, [STR VERSION\3]
GOTO START
OVER: DEBUG "no slave found"
START: DEBUG "version is ", STR VERSION\3
```

By using "ID" the module will return 3 bytes. Byte 1 is the Major number, Byte 2 is the decimal point and Byte 3 is the Minor number. These PBASIC statements may be placed at the start of any PBASIC program to detect module response.

Programming the MLX90614 module

The examples that follow will demonstrate how easy the MLX90614 module is to program. Please note that writing to the EEPROM is not complete until the 5V power to the module is cycled.

Example 1 – MLX90614_Demo-Simple.BS2

This example demonstrates basic temperature reading from the MLX90614 Module, as well as a PEC byte CRC check. Before you run this program you should load a slave address using Example 2.

Example 2 – MLX90614_Write-EEPROM.BS2

Use this program to write a slave address, alarm level and data intervals to the MLX90614 Module, particularly useful if you use Example 3 and place multiple modules on the same serial bus. Note that after you write the slave address it is necessary to switch the power off and on again. A cycle of power with the BASIC Stamp Editor's Debug Terminal open will display the new address.

Example 3 – MLX90614_Demo-Networked.BS2

Once a module has been given a slave address, you can put many of them on a single bus. All ground, power and serial I/O lines are shared. Reset lines can be shared if required. No addressing with pull-up or pull-down resistors is required as used in I²C communications. To use this program, enter each of the slave addresses in the constants section.