Infrared Temperature Sensor
用户手册

产品特点

本模块是一款非接触式红外温度传感器，能根据被测物体的红外辐射能量大小和波长分布来检测物体的表面温度。

通讯接口是 SMBus，支持 PWM 输出。传感器具体型号是 MLX90614ESF-BCC，带有温度梯度补偿。其金属封装里同时集成了红外感应热电堆探测器芯片和信号处理专用集成芯片。由于集成了低噪声放大器、17 位模数转换器和强大的数字信号处理单元，使得高精度和高分辨度的温度计得以实现。温度计具备出厂校准化，有数字 PWM 和 SMBus（系统管理总线）输出模式。作为标准，配置为 10 位的 PWM 输出格式用于连续传送温度范围为-20~120 ℃的物体温度，其分辨率为 0.14 ℃。上电默认模式是 SMBus 输出格式。

- 非接触式，高精度，高分辨率，响应时间快
- 出厂自带校准，带温度梯度补偿
- 内置电平转换电路，可直接接入 3.3V 或 5V 的 MCU 系统

版本：V1.2，日期：2017 年 9 月 4 日

参数

- 工作电压：3.3V ~ 5V
- 环境温度测量范围：-40°C ~ 85 °C
- 物体温度测量范围：-70°C ~ 380 °C
- 分辨率：0.02°C
- 精度：±0.5°C (0~50°C)
- 视场角(FOV)：35°
- 产品尺寸：28mm x 16 mm
- 固定孔尺寸：2.0mm

主要用途：

- 高精度非接触温度测量、工业温度控制、带温度控制的家用电器等

接口说明：（以接入 MCU 为例）

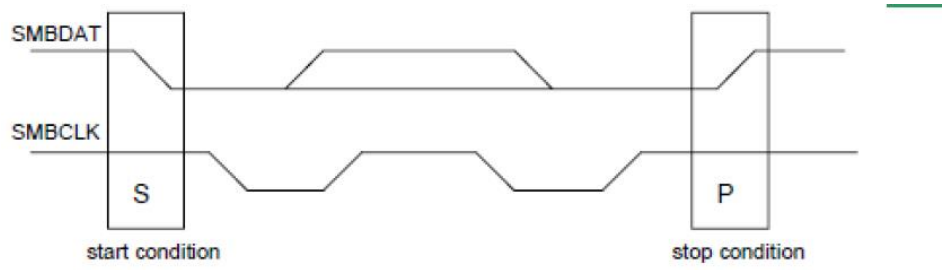
- VCC：接 3.3V ~ 5V
- GND：接 GND
- SDA：接 MCU.I2C 数据线
- SCL：接 MCU.I2C 时钟线

通讯方式

模块的通讯方式有 PWM 和 SMBus（系统管理总线），这里只介绍 SMBus 接口通讯。SMBus 是一种两线制接口。它基于 I2C 总线原理演变而来，可以认为是简化版的 I2C 总线。

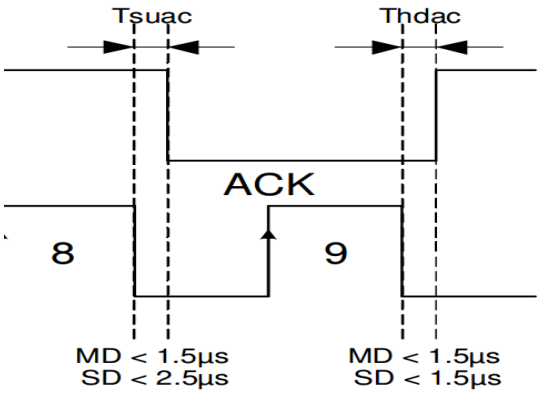
SMBus 和 I2C 总线一样，在传送数据过程中共有三种类型信号：开始信号、结束信号和应答信号。

开始信号：SCL 为高电平时，SDA 由高电平向低电平跳变，开始传送数据。

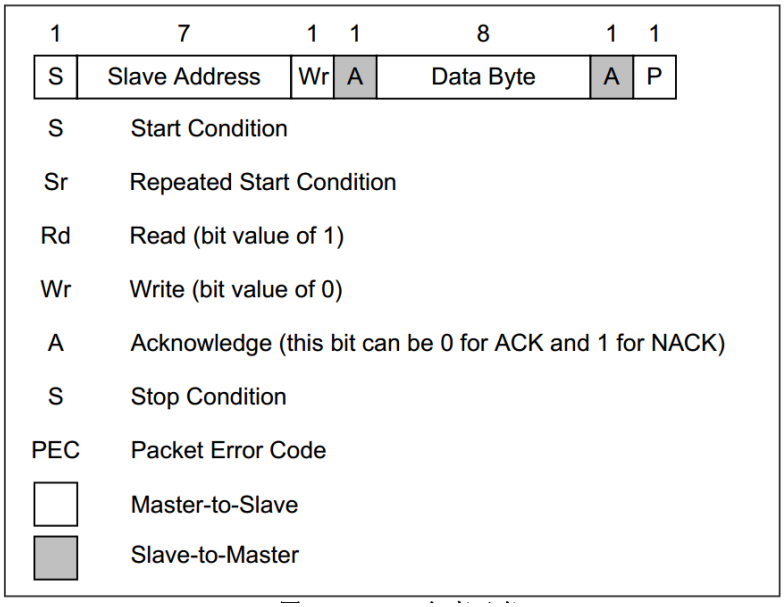


结束信号：SCL 为高电平时，SDA 由低电平向高电平跳变，结束传送数据。

应答信号：主机（从机）接收到从机（主机）的 8bit 数据后，需要向对方发送应答信号，使对方知道数据发送是否成功，即把 SDA 拉低一个完整的 SCL 周期。

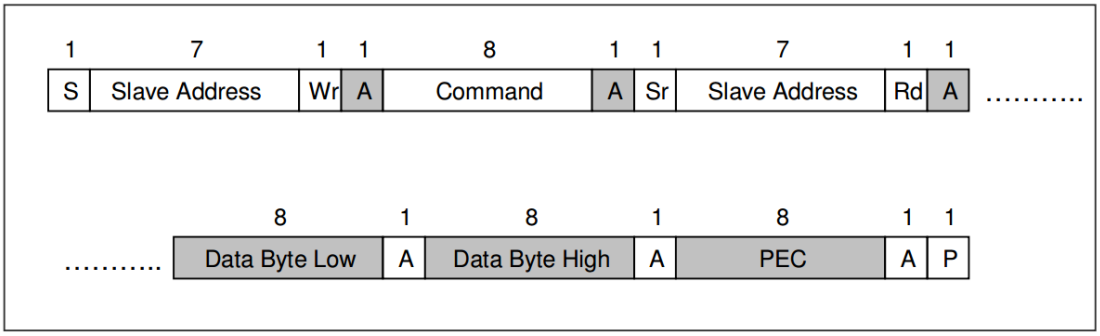


SMBus 总线协议



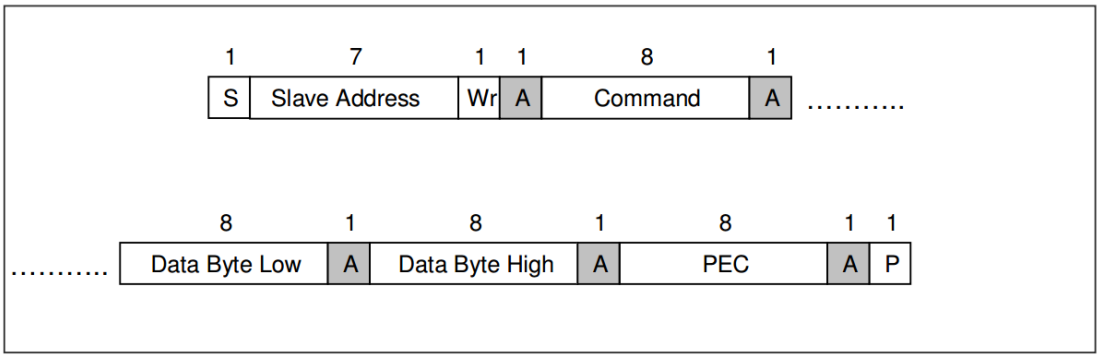
在从机（SD）接收到每 8 位数据后，会回复 ACK/NACK 信息。当主机（MD）初始化通信，将首先发送 SD 的地址，只有能识别该地址的 SD 会确认，其它的会保持沉默。如果 SD 未确认其中的任意字节， MD 应停止通信并重新发送信息 PEC 的计算结果是基于除 START、REPEATED START、STOP、ACK 和 NACK 位外的所有位 。PEC 是 CRC-8 的多项式 X8+X2+X1+1。每个字节的最高有效位首先传送。

SMBus 读数据时序



首先主机会发送一个 Start 信号，然后将从机的 7 位地址与“读”操作位组合成 8 位的数据发送给从机，从机接收到后会响应一个 ACK，主机再发送 8 位格式的命令给从机，从机接收到后发送 ACK，此时主机重新发送一个 Start 信号，然后将从机的 7 位地址与读操作位组合成 8 位的数据发送给从机，从机接收到数据后发送 ACK，然后将其寄存器中的值发送给主机，主机每接收到一个字节数据后都要回应一个 ACK，最后收到 PEC 后回应 ACK，并且发送 Stop 信号结束通信。

SMBus 写数据时序



首先主机会发送一个 Start 信号，然后将从机的 7 位地址与“写”操作位组合成 8 位的数据发送给从机，从机接收到后会响应一个 ACK 信号，主机再发送 8 位格式的命令给从机，从机接收到后发送 ACK 信号，然后主机发送低字节数据，收到从机的 ACK 后，再发送高字节数据，收到从机的 ACK 后，再发送 PEC 字节数据，收到从机的 ACK 后，主机发送 Stop 信号以结束通信。

MLX90614 的 RAM 寄存器

RAM (32x17)		
Name	Address	Read access
Melexis reserved	**0x00**	**Yes**
…	**…**	**…**
Melexis reserved	**0x03**	**Yes**
Raw data IR channel 1	**0x04**	
Raw data IR channel 2	**0x05**	
T_A	**0x06**	**Yes**
T_{OBJ1}	**0x07**	**Yes**
T_{OBJ2}	**0x08**	**Yes**
Melexis reserved	**0x09**	**Yes**
…	**…**	**…**
Melexis reserved	**0x1F**	**Yes**

MLX90614 的 EEPROM 寄存器：

EEPROM (32X16)		
Name	Address	Write access
To_{max}	0x00	Yes
To_{min}	0x01	Yes
PWMCTRL	0x02	Yes
Ta range	0x03	Yes
Emissivity correction coefficient	0x04	Yes
Config Register1	0x05	Yes
Melexis reserved	0x06	No
…	…	…
Melexis reserved	0x0D	No
SMBus address (LSByte only)	0x0E	Yes
Melexis reserved	0x0F	Yes
Melexis reserved	0x10	No
…	…	…
Melexis reserved	0x18	No
Melexis reserved	0x19	Yes
Melexis reserved	0x1A	No
Melexis reserved	0x1B	No
ID number	0x1C	No
ID number	0x1D	No
ID number	0x1E	No
ID number	0x1F	No

MLX90614 的命令格式

Opcode	Command
000x xxxx*	RAM Access
001x xxxx*	EEPROM Access
1111_0000**	Read Flags
1111_1111	Enter SLEEP mode

注*： xxxxx 代表要读取/写入的上面的 RAM、EEPROM 寄存器地址的低 5 位。

注**： 行为类似读命令。 MLX90614 在传送外 16 位数据后会反馈 PEC，其中只有 4 位是主机（MD）需要的，它会在传送完第一个字节后停止通信，读取和读取标示符的区别在于后者没有重复起始位。

读取标示符：

- Data[7] - EEBUSY – 先前对 EEPROM 的读/写操作正在进行，高有效。
- Data[6] – 未使用
- Data[5] - EE_DEAD - EEPROM 发生双重错误，高有效。
- Data[4] - INIT – POR 初始化程序正在进行，低有效。
- Data[3] – 未执行。
- Data[2..0] 和 Data[8..15] – 都为 0。

如何使用

程序分析

从上面的 SMBus 总线协议和通讯时序可以知道，开始通讯时，首先主机会发送一个开始信号，然后将从机的 7 位地址（SA）与读/写操作位组合成 8 位的数据发送给从机，当我们的 SMBus 总线上只有一只 MLX90614 传感器时，7 位的从机地址（SA）默认是 0x00，当需要用同一总线操作多只 MLX90614 时，可以修改 EEPROM 中的从机地址。那么主机发送完开始信号后，如果我们要读 MLX90614，那么就接着发送 (SA<<1) + 0 = 0x00，如果我们要写，那么就接着发送 (SA<<1) + 1 = 0x01。

根据上面的 RAM 寄存器地址表格，可以知道在 RAM 寄存器中，环境温度寄存器地址为 0x06，物体温度寄存器地址为 0x07,从命令表格知道，访问 RAM 的操作码是 0x00，访问 EEPROM 的操作码是 0x20，一般我们只需要从 RAM 中读取温度，不需要访问 EEPROM。那么我们要访问 RAM 中的环境温度寄存器，命令为：0x00 | 0x06 = 0x06，访问 RAM 中的物体温度寄存器，命令为：0x00 | 0x07 = 0x07。

以 STM32 的程序为例，该驱动程序是通过 PB8、PB9 来模拟 SMBus 时序，驱动代码在 smbus.c 文件中。我们按照上面介绍的时序读取到环境温度、物体温度数据后，可以根据数据手册来计算环境温度、物体温度：

摄氏度值（°C）： ((TempData_H <<8) + TempData_L)*0.02 - 273.15

测量原理

对于非接触式红外测温模块，很重要的一个概念是“视场 (FOV)”。 视场是由温差电堆接收到 50%的辐射信号来确定的，并且和传感器的主轴线相关。测得的温度是视场内被测物体的温度加权平均值，所以当被测物体完全覆盖 FOV 视场时的准确度是最高的。

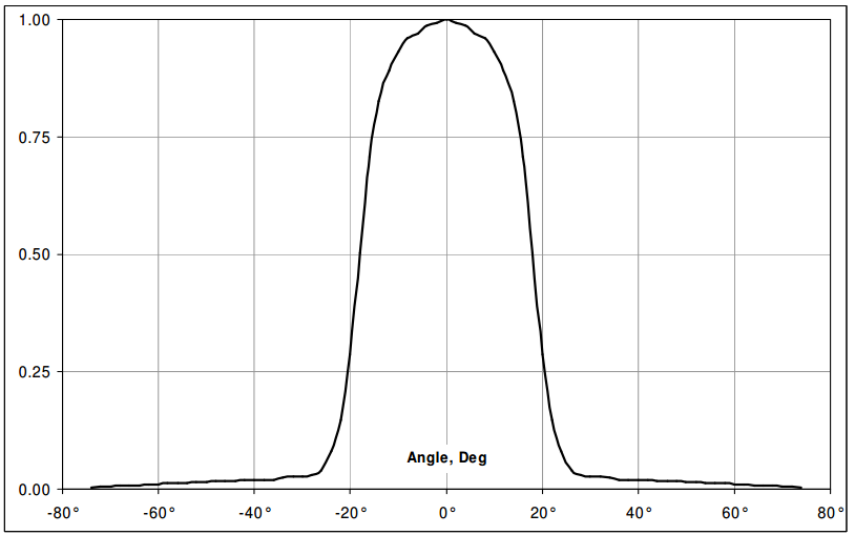


Figure 31: Typical FOV of MLX90614xCC

本模块的传感器型号是 MLX90614ESF-BCC，上图为 BCC 的 FOV 图，FOV = 35°，即：

被测物体半径÷与传感器探头的距离 = tan35°，即当被测物体的半径为 5CM 时，最大的测量距离为 7CM（指的是保证该温度值是准确的最大测量距离）。

而市面上常用的 BAA 的 FOV 图如下：

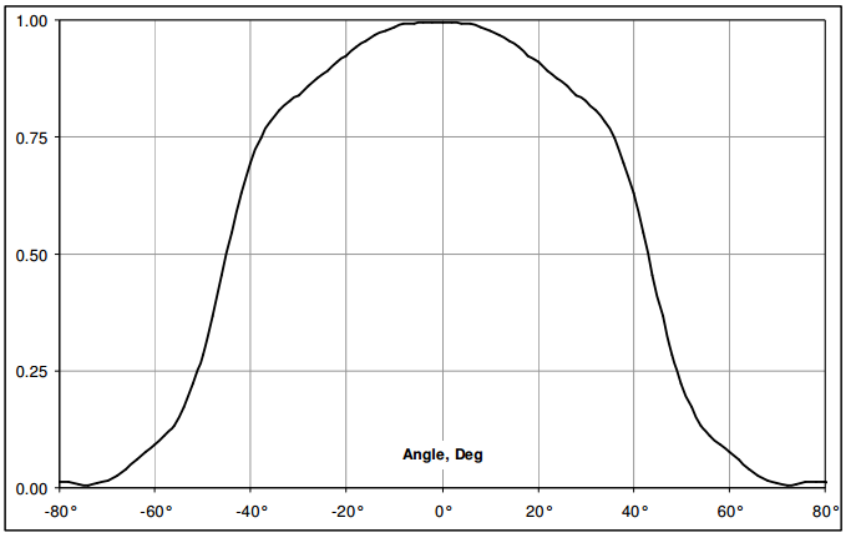


Figure 28: Typical FOV of MLX90614xAA

对比了一下两个型号的传感器，实际测试的结果是：

拿手掌作为温度参考物，当 BCC 距离手掌 7CM 以内时，温度基本不变，当距离扩大至 15CM 时，温度值才下降了 1°C。而对于 BAA，当距离手掌 2CM 以外时，温度就已经开始变化得很快，当距离扩大至 4CM 时，温度值就已经下降 1°C。

区别很明显，BCC 比 BAA 性能要好不少，可见测温的距离和准确度和 FOV 有关，FOV 越小，性能就越好。

操作现象

下面以 NUCLEO-F103RB 和 Arduino UNO 开发板为例。

NUCLEO-F103RB

1) 用杜邦线将传感器模块的 VCC 接到 NUCLEO-F103RB 的 3V3 或者 5V 引脚，GND 接到 GND，SCL 接到 PB8，SDA 接到 PB9。
2) 用 keil 软件打开程序 .\ MDK-ARM \ Infrared-Temperature-Sensor-Code.uvprojx，编译、下载。
3) 打开串口监视软件，选择正确的串口号，并设置如下：波特率：115200；数据位：8；停止位：1；校验位：None；控制流：None。

Arduino UNO

1) 用杜邦线将传感器模块的 VCC 接到 Arduino UNO 的 3V3 或者 5V 引脚，GND 接到 GND，SCL 接到 SCL，SDA 接到 SDA。
2) 将最外层的 WaveShare_MLX90614 文件夹复制到 Arduino 软件的安装目录 Arduino\libraries 下面。点击 File --> Examples --> WaveShare_MLX90614--> WaveShare_MLX90614 打开程序，并编译、下载。
3) 点击 Tools -> Port 选择 Arduino 开发板的串口号，用打开串口监视器，设置 No line ending, 115200baud。

预期结果

把传感器的探头正对着热源，例如手掌，串口便会打印对应的读数，如：

```
Ambient Temp: 29.11 ℃          Object Temp: 34.47 ℃
Ambient Temp: 29.13 ℃          Object Temp: 34.43 ℃
Ambient Temp: 29.17 ℃          Object Temp: 34.47 ℃
Ambient Temp: 29.13 ℃          Object Temp: 34.59 ℃
Ambient Temp: 29.17 ℃          Object Temp: 34.53 ℃
```