A31L123

Non-contact IR Temperature Sensor Board Design Guide

Application Note

Version 1.02

Contents

List of Figures

List of Tables

1 Introduction

This document explains how to design a Non-contact Infrared Red (IR) Temperature Sensor Board using A31L123.

ABOV's A31L123 is a 32-bit low power MCU, and can reduce operating current by using DEEP SLEEP mode.

During DEEP SLEEP mode, A31L123 consumes only about 1uA of the current and is capable of operating the Non-contact IR Temperature Sensor. (For specification of A31L123, please refer to **23.11 Supply current characteristics** of A31L123 User's manual.)

After configuring the board with A31L123 MCU according to the instructions in this document, users can measure the target temperature and ambient temperature, as well as the operating current of A31L123 that is used on the board during DEEP SLEEP mode.

In addition, this document provides hardware and software materials and solutions for implementing the Non-contact IR Temperature Sensor Board.

2 Electric Circuits and Components

In chapter 2, components and circuit diagram for the Non-contact IR Temperature Sensor board is introduced.

2.1 MLX90614

The MLX90614 is an IR thermometer for non-contact temperature measurements. The temperature value measured by the MLX90614 is stored in a form of 16-bit hex data at the specified address in RAM. The data stored in RAM or EEPROM can be read by using a corresponding command and can be changed as necessary.

The 1st data can be obtained after 0.25sec once it starts operating, and each address contains 16-bit data respectively. For Read and Write operations, LSByte is first processed (the lower 8bits of the 16-bit data are processed) and then MSByte is processed (the upper 8bits of the 16-bit data are processed).

Figure 1 shows a basic circuit diagram of the MLX90614.

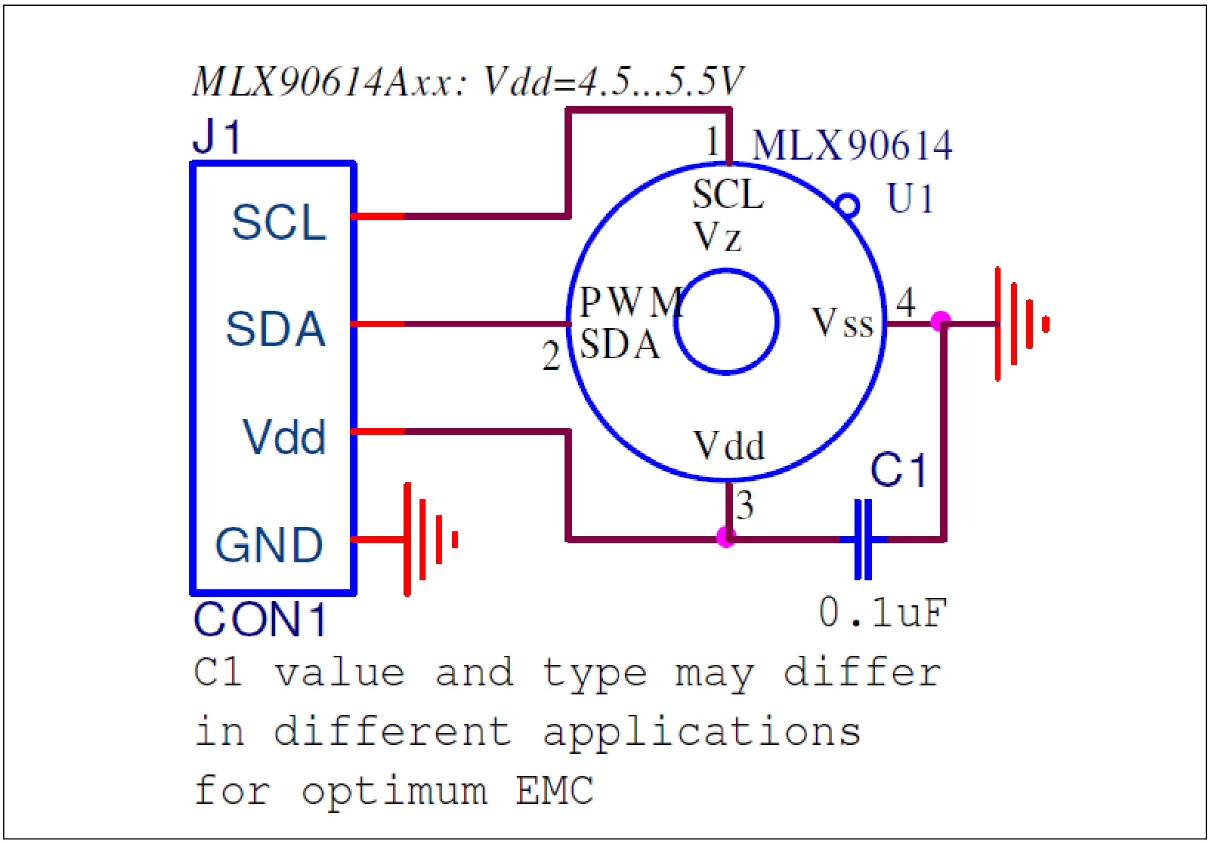


Figure 1. Basic Circuit Diagram of MLX90614

The MLX90614 features the followings:

- Operation voltage: 2.6V to 3.6V
- Communication method: PWM/ I2C (10kHz to 100kHz)
- Measured temperature:
  - Ambient temperature -40°C to +125°C
  - Object temperature -70°C to +380°C
- Measurement accuracy/ resolution: 0.5°C/ 0.02°C (PWM 0.14°C resolution -20°C to 120°C measurement)
- Factory calibrated output
- Measured value is an average temperature value of all objects in FOV.
- Factors that affect the sensor's temperature cannot be placed in the peripheral area where the sensor is mounted. The sensor cannot come too closed to the target.
- For the xCx version (ex:MLX90614BCC), the internally measured thermal gradient and the measured temperature can be used for compensation.
  - The xCx version becomes less sensitive to the thermal gradient but not completely free from its effect. Therefore, it is important to avoid the causes of the thermal gradients or protect the sensor from the causes of the thermal gradients.
- It is already calibrated for object radiation 1, but it can be adjusted to between 0.1 and 1.0 without re-calibration.

2.1.1 EEPROM

Table 1. EEPROM Table

EEPROM (32 × 16)		
Name	**Address**	**Write access**
To_{max}	0x00	Yes
To_{min}	0x01	Yes
PWMCTRL	0x02	Yes
Ta range	0x03	Yes
Emissivity	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

EEPROM features the followings:

- Only parts of EEPROM can be modified, but data at whole addresses can be read.
- Before starting Write operation on the EEPROM, the current data must be erased by writing 0 at the corresponding address.
- The emissivity (0x04) can be adjusted to the value between 0.1 and 1.0.
- To maintain factory calibration, if users do not use any special method, values of the following bits and registers must not be changed:
  - Emissivity
  - Config Register1
  - Addresses 0x0F and 0x19

2.1.2 RAM

Table 2. RAM Addresses

RAM (32 × 16)		
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

RAM addresses features the followings:

- Writing is not possible, but some registers of RAM can be read.
  - 0x06: Ambient temperature
  - 0x07: Object temperature 1
  - 0x08 : Object temperature 2

2.1.3 Command

Table 3. SMBus Commands

Opcode	Command
000x xxxx*	RAM access
001x xxxx*	EEPROM access
1111_0000**	Read flags.
1111_1111	Enter SLEEP mode.

SMBus command features the followings:

- If users want to access RAM/ EEPROM, enter the address that users want to Read or Write to in the “x xxxx” field.
  - RAM Obj1 temperature: 0x07
  - EEPROM emissivity: 0x24

2.1.4 Read

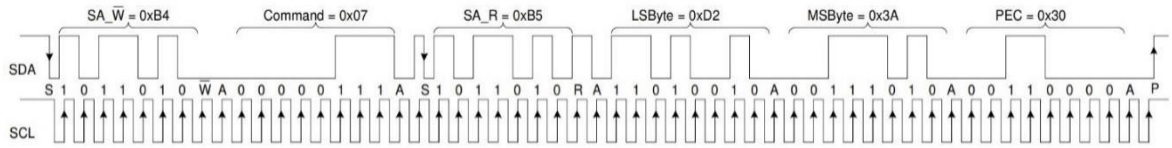


Figure 8: Read word format (SA=0x5A, read RAM=0x07, result=0x3AD2, PEC=0x30)

Figure 2. Example Timing Diagram of Read

When performing Read, please remember that the actual address of the sensor is not 0x5A but 0xB4. This actual address is the value shifted by 1bit (MSB first). In addition, it is important to remember that the last byte is the result of CRC-8 calculation.

2.1.5 Write

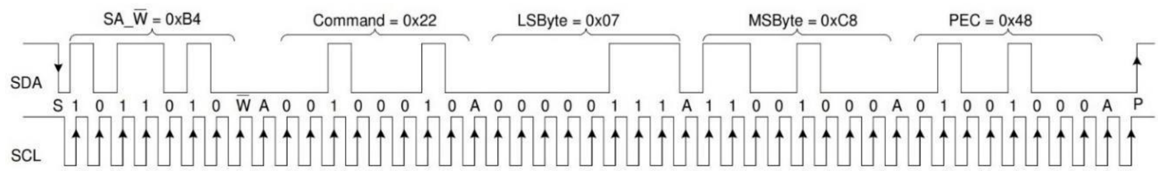


Figure 9: Write word format (SA=0x5A, write EEPROM=0x02, data=0xC807, PEC=0x48)

Figure 3. Example Timing Diagram of Write

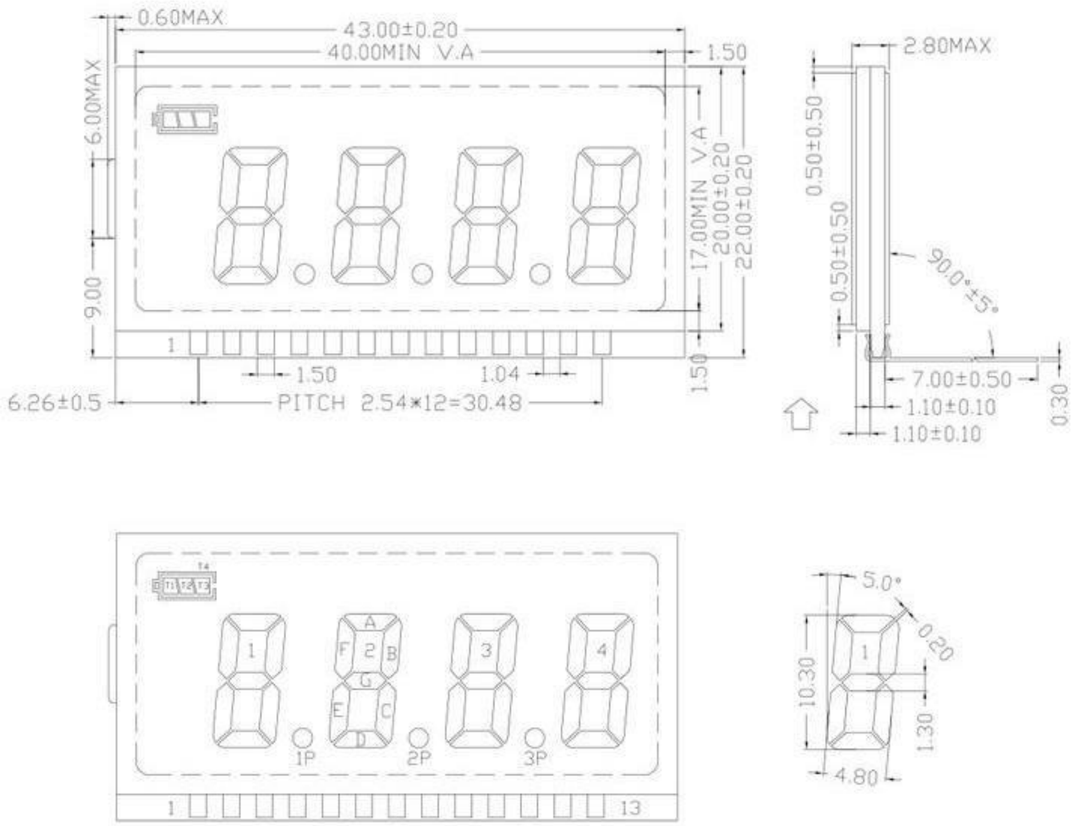
When performing Write, please be careful of the followings:

- Data can only be written to EEPROM.
- Before writing data, writing 0 must be executed.
  - Minimum wait time of 5ms after writing 0 is required (10ms is recommended to wait).
- The last byte is the result of CRC-8 calculation.

2.2 LCD Module and Interfaces

LCD module displays the measured temperature using 4-digit 7-segment LCDs. For detailed information or specification of the LCD components, please refer to the GDC8310 technical documents.

This section provides examples and references that explain a method to utilize and apply the LCD components.



PIN	1	2	3	4	5	6	7	8	9	10	11	12	13
COM0				COM0	1D	1P	2D	2P	3D	3P	4D		T2
COM1			COM1		1E	1C	2E	2C	3E	3C	4E	4C	T3
COM2		COM2			1F	1G	2F	2G	3F	3G	4F	4G	T1
COM3	COM3				1A	1B	2A	2B	3A	3B	4A	4B	T4

Figure 4. 4-Digit LCD Panel (GDC8310)

Figure 4 describes the information of basic circuit connections. To control the LCD module, users need to be aware of the following facts:

- Users can control the LCD module shown in the Figure 4 by using the LCD driver peripheral of A31L123.
- GDC8310 features the 3.3V of operating voltage and 1/4 Duty 1/3 Bias driving method.

A31L123 MCU and GDC8310 can be connected through the COM and SEG ports as shown in Figure 5.

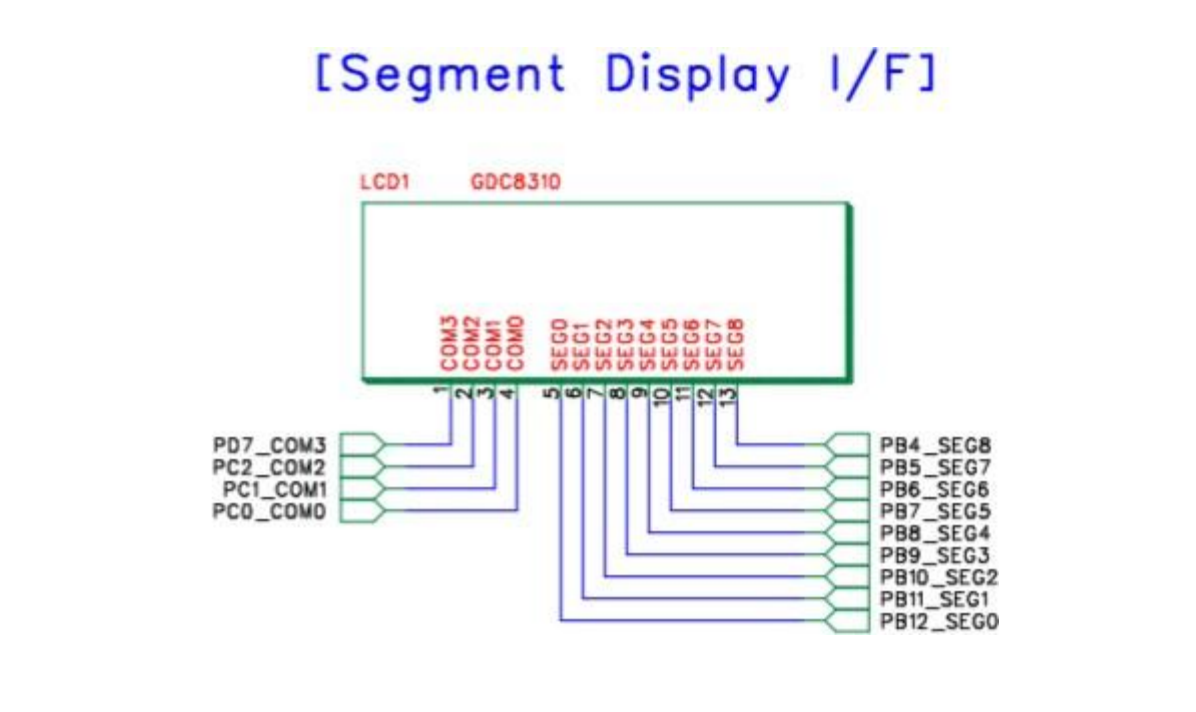


Figure 5. Circuit Diagram of the Connection between A31L123 and 4-Digit LCD Panel

3 Software Structure

Chapter 3 introduces the requirements for the software.

3.1 State Machine

A state machine generally consists of a finite number of states. Based on the current state and the given input values, the state machine makes a change of the current state and generates outputs.

Example software provided in this section is implemented based on the FreeRTOS.

3.2 Example Software Code for a State Machine

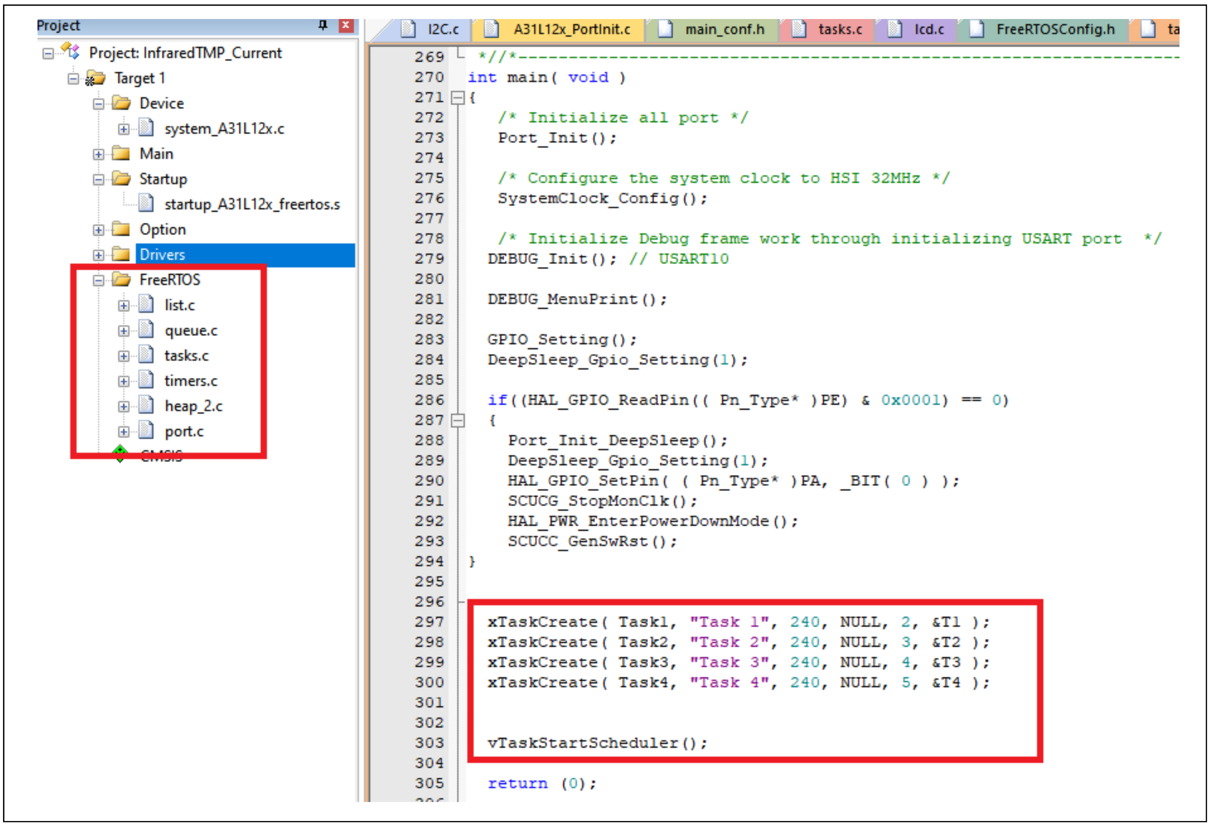


Figure 6. Example FW (main) of Non-contact IR Temperature Sensor

This example code consists of 4 Tasks, and can be modified according to user usage. Each Task's purpose is introduced in the followings:

- Task1 operates for the UART communication. It uses the user input or Non-contact IR Temperature Sensor's protocols as its input. (20ms period)
- Task2 executes the command that is input by users. (100ms period)

- Task3 operates for the battery animation, status LED, DEEP SLEEP mode, and LCD display. (500ms period)
- Task4 adds the function that users want.

```
void i2c_start(void)
{
  //start
  I2C_M_INIT;//Set SCL, SDA to output
  I2C_M_SDA_H;
  I2C_M_SCL_H; // Set SCL, SDA High
  Delay_us(I2C_DELAY);
  I2C_M_SDA_L; // Clear SDA
  Delay_us(I2C_DELAY);
  I2C_M_SCL_L; // Clear SCL
}

void i2c_stop(void)
{
  I2C_M_SDA_O; // Set SDA to output
  I2C_M_SDA_L; // Clear SDA Low
  Delay_us(I2C_DELAY);
  I2C_M_SCL_H; // Set SCL High
  Delay_us(I2C_DELAY);
  I2C_M_SDA_H; // Set SDA High
  I2C_M_SDA_I; // Set SDA to Input
}

void write_i2c_byte(unsigned char byte)
{
  unsigned char i = 0;
  I2C_M_SDA_O; // Set SDA to output
  Delay_us(I2C_DELAY);
  for (i = 0; i < 8 ; i++)
  {
    if((byte & 0x80)==0x80)
      I2C_M_SDA_H; // Set SDA High
    else
      I2C_M_SDA_L; // Clear SDA Low
    Delay_us(I2C_DELAY);
    I2C_M_SCL_H; // Set SCL High, Clock data
    __nop();
    Delay_us(I2C_DELAY);
```

Figure 7. Example Code implementing I2C Communication using GPIO

Figure 7 shows an example code of I2C that is communicating with Non-contact IR Temperature Sensor. More code is introduced in the attached source code file.

3.3 User Software

This section introduces GUI software for user convenience.

Figure 8 shows an icon of the GUI software.

Figure 8. GUI Software Icon

Figures below describe the sequence in which GUI operations are set up:

Figure 9. User GUI Operation 1

1. Select a COM Port and the corresponding device in Figure 9.

2. Click on the “COM OPEN” button.

Figure 10. User GUI Operation 2

1. When the COM Port is connected correctly, the “Start” button becomes active as shown in Figure 10.
2. Click on the “Start” button to receive Data inputs from the Non-contact IR Temperature Sensor.
3. The Data inputs has the protocols below:
   - ***! tx00230 00229 00223 Ex9993 62E3 0200 F71C C005 9FF4 BE5A 3008 E294 628B B20A !****

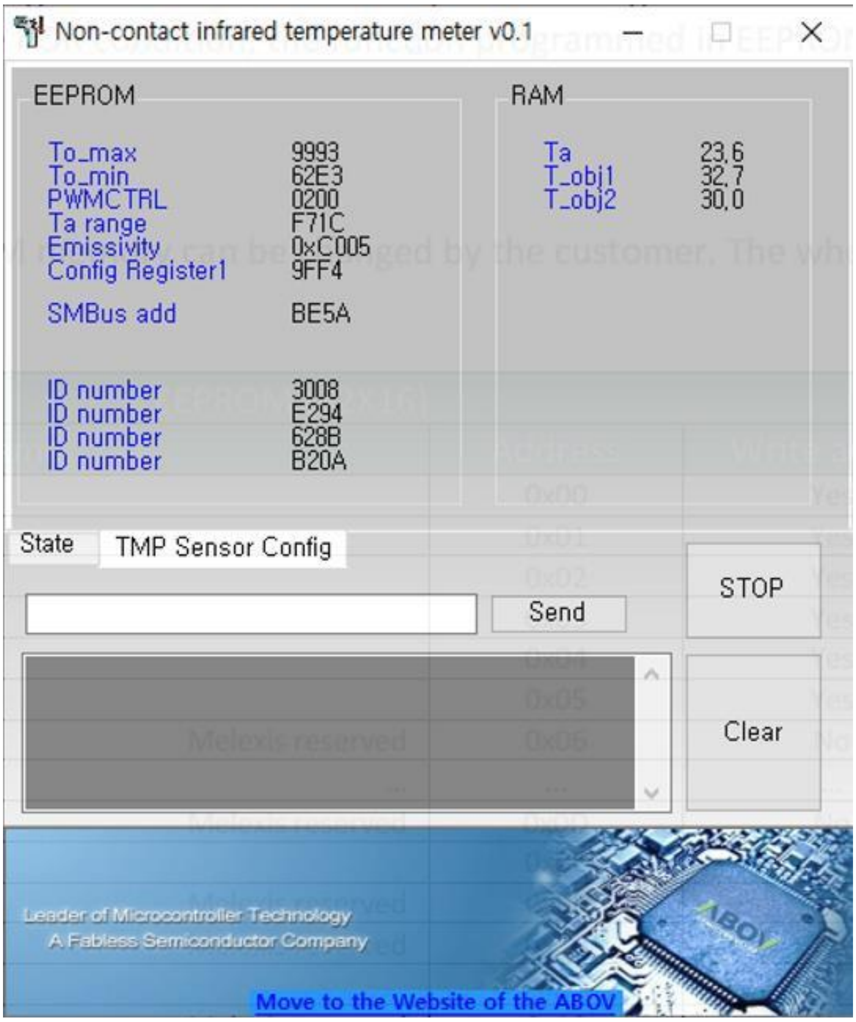


Figure 11. User GUI Operation 3

Select the "TMP Sensor Config" in Figure 11 to check the status of EEPROM and RAM.

4 Quick Setup and Use

Chapter 4 guides users to set up and use the Non-contact IR Temperature Sensor board.

4.1 Structure of Non-contact IR Temperature Sensor board

Figure 12 shows an actual image of the current sensor evaluation board.

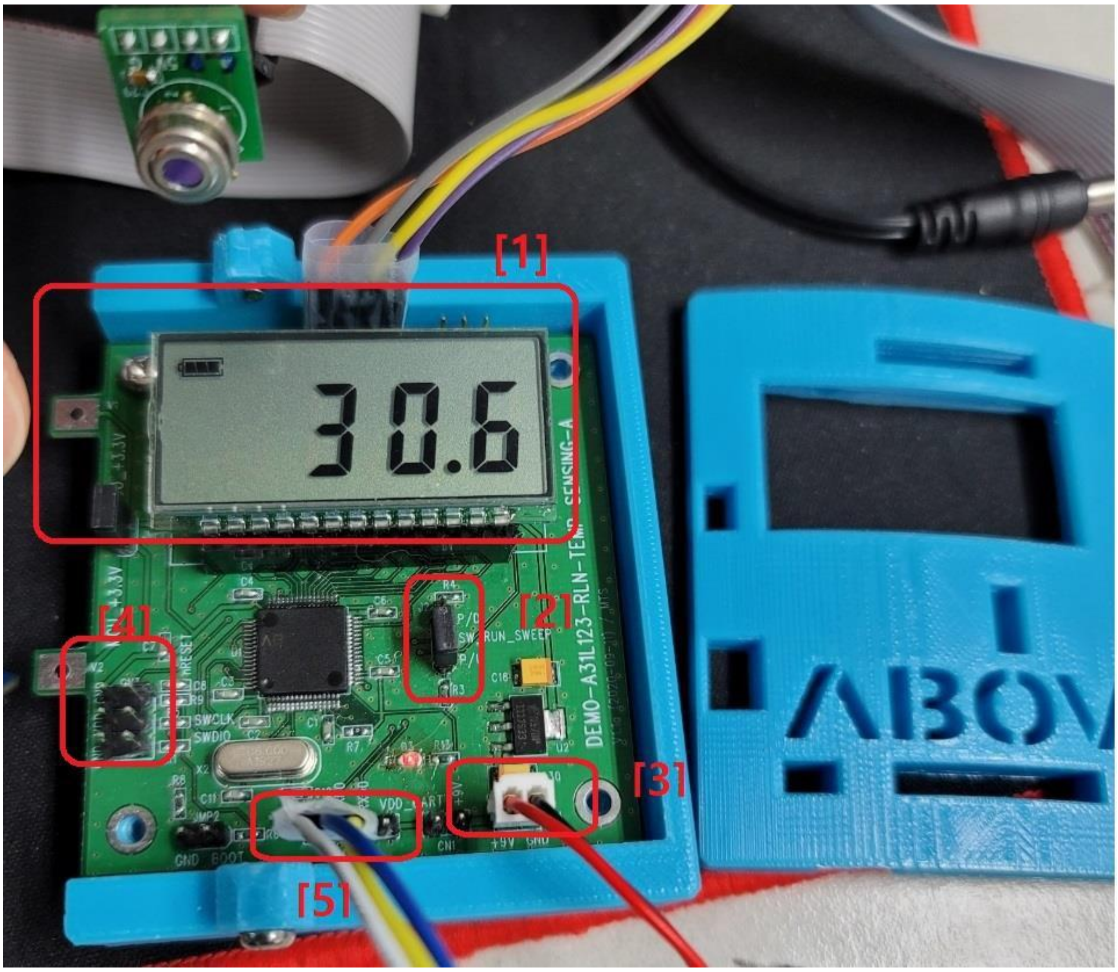


Figure 12. Actual Image of Non-contact IR Temperature Sensor Board

Each part corresponding to the number in Figure 12 has the following features:

[1] An LCD module with the current sensing input ports N and P on its left side.

[2] A switch selecting RUN mode or DEEP SLEEP mode.

[3] A battery input terminal that is dedicated for 9V voltage.

[4] Terminals for an MCU (A31L123) download and debugging (SWD).

[5] Terminals for an MCU (A31L123) UART communication that are used for debugging and test.

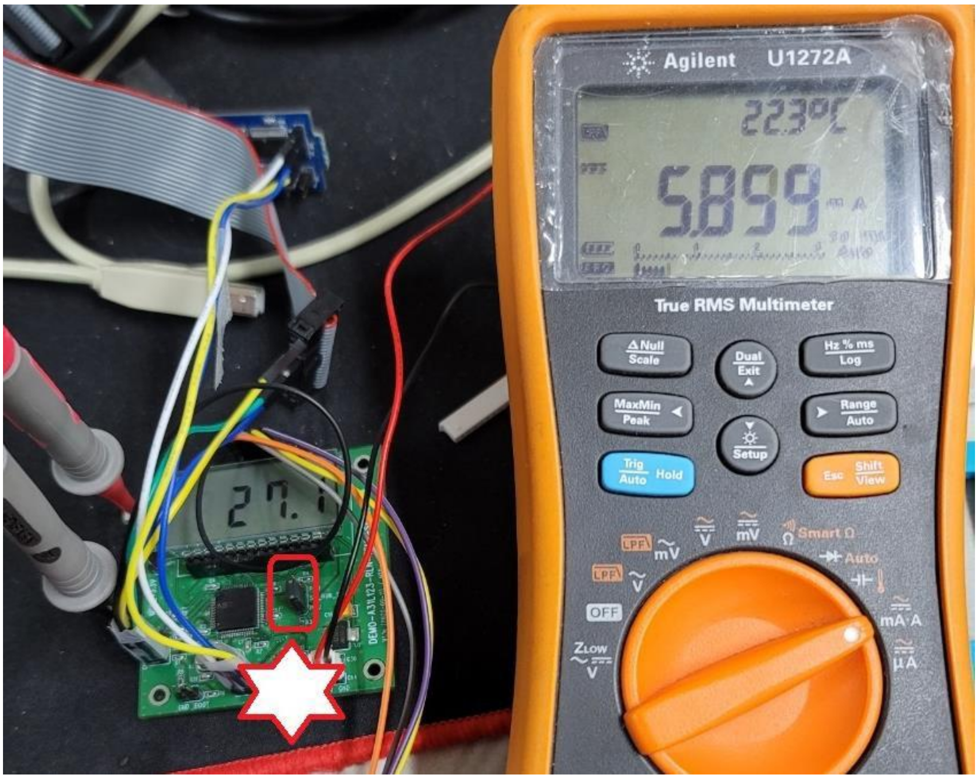


Figure 13. Current Consumption in RUN Mode

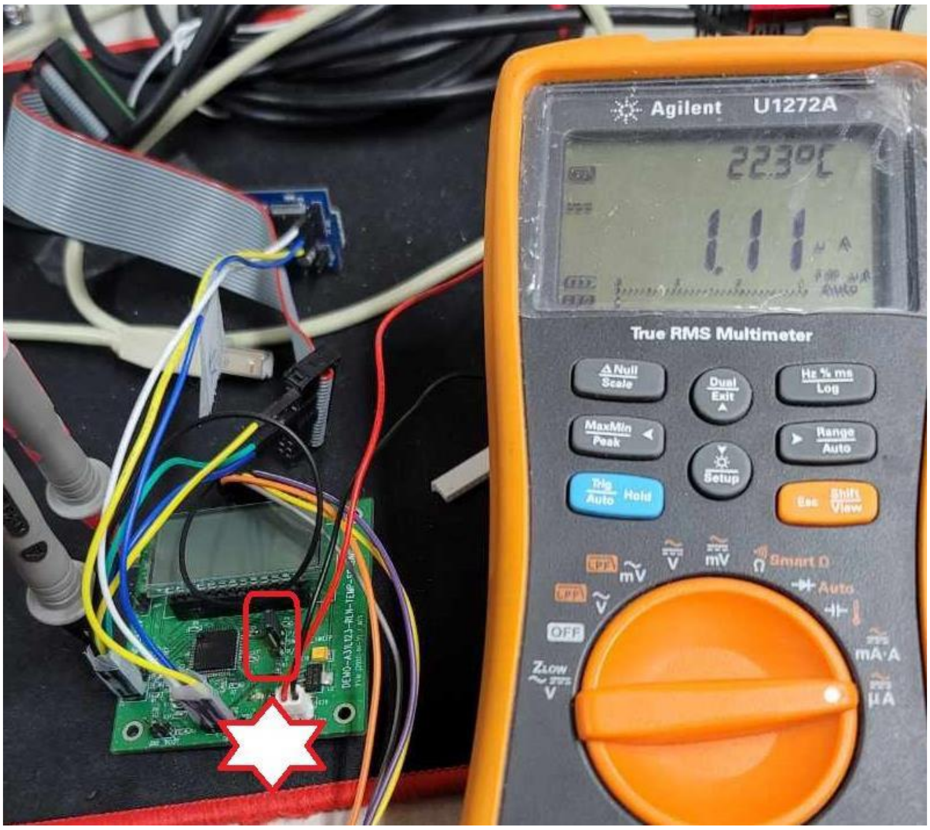


Figure 14. Current Consumption in DEEP SLEEP Mode

4.2 Example Circuit Diagram of Non-contact Temperature Sensor Board

Figure 15 shows an example circuit diagram of Non-contact IR Temperature Sensor board.

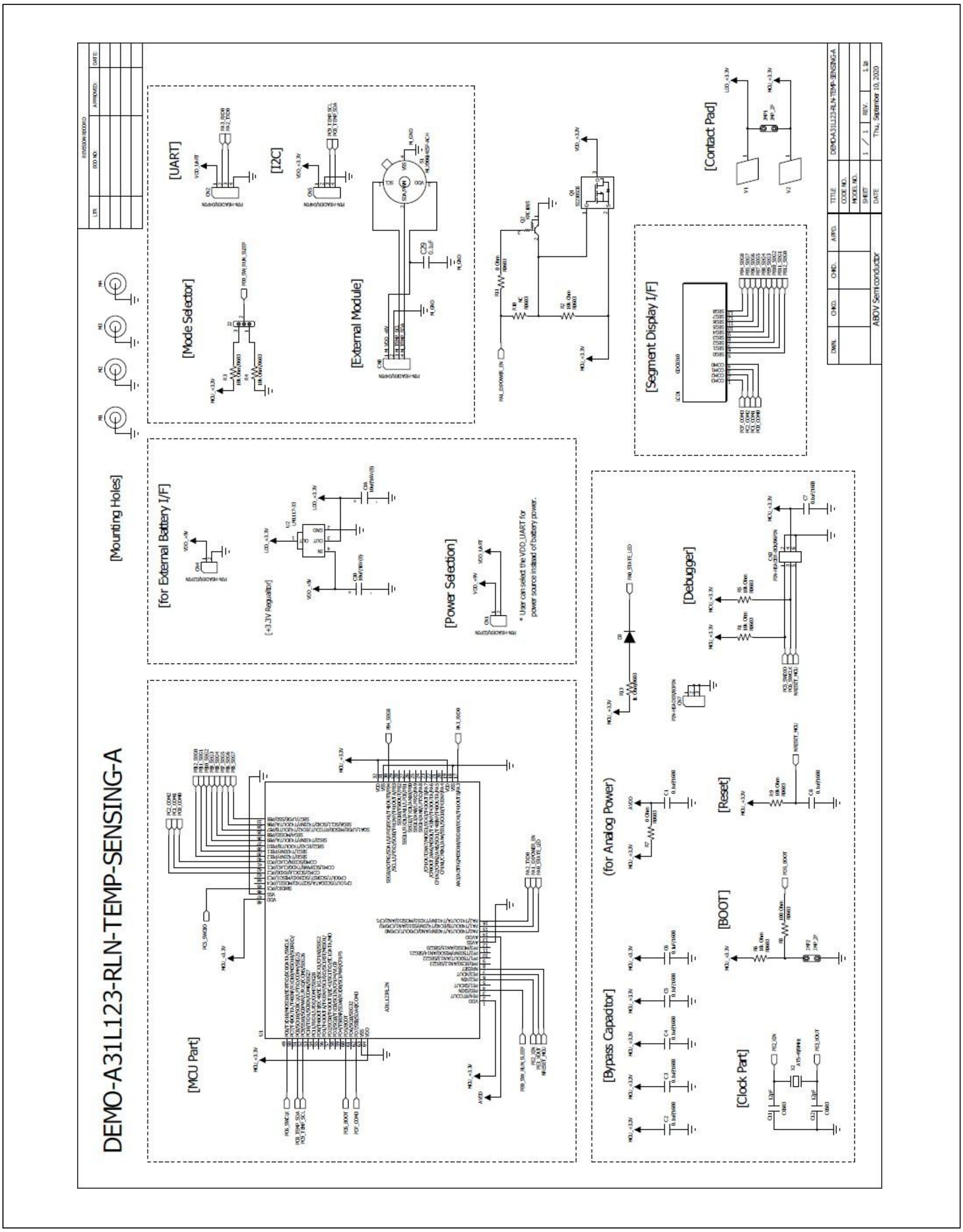


Figure 15. Example Circuit Diagram of Non-contact IR Temperature Sensor Board

5 Conclusion

This “A31L123 Non-contact IR Temperature Sensor Board Design Guide” provides a guide to hardware and software design for Non-contact IR Temperature Sensor board, and shows an actual evaluation board to introduce a method to use it.

6 References

- TMLX90614 family (Infra-Red Thermometer)
- ABOV Semi – A31L123 (32bit-MCU)

Revision History

Version	Date	Description
1.00	20.12.28	Initial preliminary version created
1.01	22.11.14	Revised the font of this document
1.02	24.12.02	Updated the disclaimer.

Korea

Regional Office, Seoul
R&D, Marketing & Sales
8th Fl., 330, Yeongdong-daero,
Gangnam-gu, Seoul,
06177, Korea

Tel: +82-2-2193-2200
Fax: +82-2-508-6903
www.abovsemi.com

HQ, Ochang
R&D, QA, and Test Center
37, Gangni 1-gil, Ochang-eup,
Cheongwon-gun,
Chungcheongbuk-do,
28126, Korea

Tel: +82-43-219-5200
Fax: +82-43-217-3534
https://www.abovsemi.com/en/main.php

Domestic Sales Manager
Tel: +82-2-2193-2206
Fax: +82-2-508-6903
Email: sales_kr@abov.co.kr

Global Sales Manager
Tel: +82-2-2193-2281
Fax: +82-2-508-6903
Email: sales_gl@abov.co.kr

China Sales Manager
Tel: +86-755-8287-2205
Fax: +86-755-8287-2204
Email: sales_cn@abov.co.kr