Temperature Sensors Test Report

# Test Components:

|  |  |  |
| --- | --- | --- |
| **Sensor Description** | **Part Number** | **Output Interface** |
| TI ±0.1°C Precision Analog Temperature Sensor | LMT70 | Analog (Voltage) |
| Microchip MCP9808 digital temperature sensor with ±0.5°C (max.) accuracy | MCP9808 | I2C upto 400KHz |
| TI digital humidity sensor with integrated temperature sensor  Relative Humidity Accuracy ±2% (typical)  Temperature Accuracy ±0.2°C (typical) | HDC1080 | I2C upto 400KHz |
| Silicon Labs digital humidity sensor with integrated temperature sensor  Relative Humidity Accuracy ±3% (typical)  Temperature Accuracy ±0.4°C (typical) | Si7021 | I2C upto 400KHz |

# Test on 5/11

## Task1:

Measure room temperature, which is set to 78**°**F (25.5**°**C). The LMT70 evaluation board and HDC1080 evaluation board are placed side by side on a desk facing up. The measurement screenshots are taken 15 min after the first measurement started.

Conclusion: The result from LMT70 is very close to the room temperature the set by the thermostat while the result from HDC1080 is about 1**°**C higher than expected room temperature.

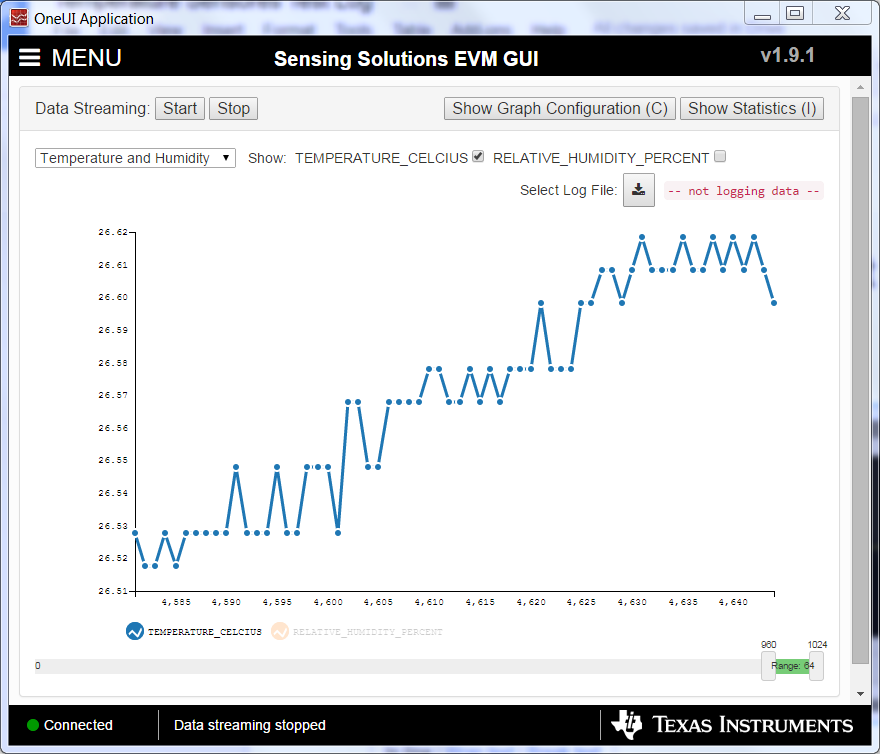


Figure 1: Test results of HDC1080

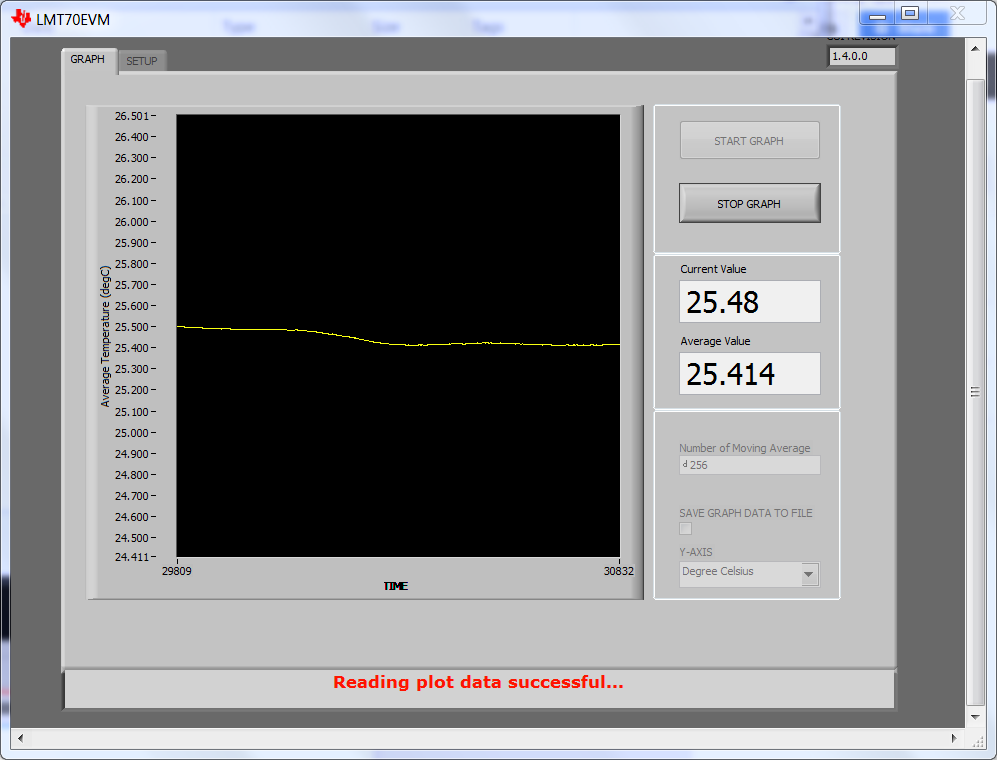


Figure 2: Test results of LMT70

# Test on 5/30

Measure room temperature, which is set to 78**°**F (25.5**°**C). The LMT70 evaluation board and HDC1080 evaluation board are placed side by side on a desk facing up. The MCP9808 sensor board and the Si7201 sensor board is each connected to a MSP430F5529 launch board, which is communicated to them via I2S and the results are displayed on a terminal window via USB using virtual Comm connection. The measurement screenshots are taken 15 min after the first measurement started.

The following are a snapshot of the measurement results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Thermostat | LMT70 | HDC1080 | MCP9808 | Si7201 |
| Temperature (**°**C) | 25.5 | 25.90 | 27.17 | 25.7696 | 25.625 |
| Humidity (RH %) |  | N/A | 51.05 | N/A | 51.3845 |

Conclusion: once again, reading from HDC1080 is more than 1**°**C higher than the reading from other 3 sensors. I will use another HDC1080 board to do the measurement again to rule out any hardware failure.

# Test on 6/1

Measure room temperature, which is set to 78**°**F (25.5**°**C). The LMT70 evaluation board and HDC1080 evaluation board are placed side by side on a desk facing up. The MCP9808 sensor board and the Si7201 sensor board is each connected to a MSP430F5529 launch board, which is communicated to them via I2S and the results are displayed on a terminal window via USB using virtual Comm connection. The measurement screenshots are taken 15 min after the first measurement started. I also used a precision thermometer to measure the room temperature and use its reading as a reference

The following are a snapshot of the measurement results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Thermometer | LMT70 | HDC1080 | MCP9808 | Si7201 |
| Temperature (**°**C) | 25.44 | 26.18 | 27.86 | 26.00 | 25.97 |

Conclusion: All 4 sensors readings are at least 0.5**°**C higher than the reading from a precision thermometer, which is used as a referencing point. At the same time, the reading from the HDC1080 continues to demonstrate it is way off compares to the other 3 sensors. I think we should remove it from our candidate list.

# Comm Port Test of Huawie’s eM300 module on 6/6

## Test setup:

-Used RealTerm on a PC to capture and display any message received from the RS232 port.

-Used a RS232 to USB convertor to connect PC and eM300-8b’s UART0 port

-According to eM300-8b’s spec, RealTerm is configured to with baud rate of 9600, 8 data bit, no parity, 1 stop bit and there is no hardware control.

-Use a scope to monitor eM300-8b UART0’s tx pin.

## Test Results:

Upon the eM300-8b test board is powered up, it sends out a set of message periodically on its UART0 port.

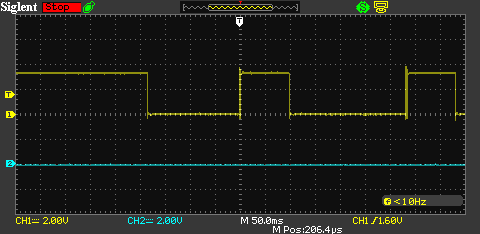
According to Huawei’s spec, the expectation of the test is to see a string of “\r\nHuawei eLTE-IoT\r\nOK\r\n" shown in the RealTerm terminal. Instead, only junk characters are captured on the screen. Furthermore, the screenshots captured by the oscilloscope suggests 2 sets of messages are sent out with 2 different baud rates. The following are some of the screen shots:

Figure : zoom out capturing

There are 2 sets of messages spitted out periodically from the eM300 module; these 2 sets of messages are approximately 760ms apart of each other.

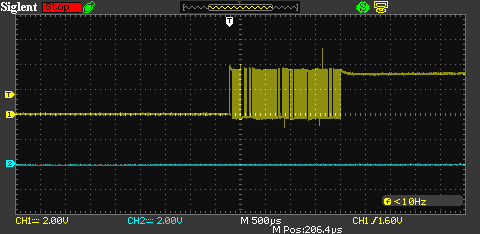


Figure : zoom in view of first message

Both figure 4 and 5 contains the 1 of the 2 messages shows in figure 3. Figure 5 clearly shows that the baud rate of this message is about 115000, much higher than the 9600 baud rate listed in the spec.

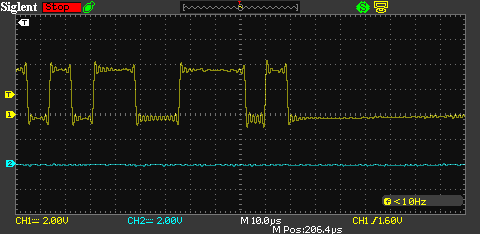


Figure : Close up view of 1st message

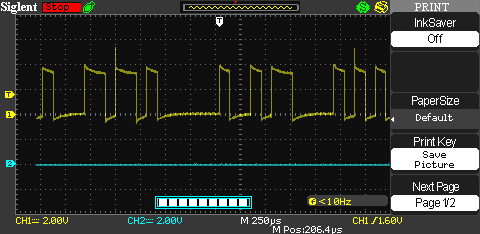


Figure : Close up of the second message

Figure 6 shows the close up of the second message in Figure 3. The time scale suggests that this message is sent at the baud rate of 9600, which matches the spec but it is different than the first message in Figure 3.

Based on the results captured from oscilloscope, I capture data using both 115200 and 9600 as the baud rate speed to see whether at least 1 of the 2 messages are valid.

Raw data that are captured at 9600 baud rate:

E6001000000079BDABD48AAA6AFAAB3A5DD5956B855D95EB6169E5EBE5EB63351527BF796D2157E5EB6169E5EB00

Raw data that are captured at 115200 baud rate:

000079BDABD48AAA6AFAAB3A5DD5956D2157E5EB6169E5EBE5EB63351527BF796D2157E5EB6169E5EB0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

In either settings, the captured data do not match the expected string specified in the eM300 spec (“\r\nHuawei eLTE-IoT\r\nOK\r\n").