Thermal Canyon

Product Architecture Document (PAD)

May 2015

Revision 0.9

Intel Confidential

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products in the design phase of development.

All Thermal Canyons featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use Thermal Canyons in advertising, promotion or marketing of any product or services and any such use of Intel's internal Thermal Canyons is at the sole risk of the user.

Intel, Intel Atom and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

\*Other names and brands may be claimed as the property of others.

Copyright © 2012, Intel Corporation. All rights reserved.

Contents

[1 Introduction 6](#_Toc419362918)

[1.1 Terminology 6](#_Toc419362919)

[1.2 Background 6](#_Toc419362920)

[1.3 Thermal Canyon Requirements 7](#_Toc419362921)

[2 Hardware Implementation 13](#_Toc419362922)

[2.1 Temperature Sensor 14](#_Toc419362923)

[2.1.1 Sensor design 14](#_Toc419362924)

[2.2 Microcontroller 15](#_Toc419362925)

[2.3 GSM/GPRS Modem 15](#_Toc419362926)

[2.3.1 Modem – Microcontroller interface 16](#_Toc419362927)

[2.3.2 Dual SIM 17](#_Toc419362928)

[2.3.3 Antenna 17](#_Toc419362929)

[2.4 Battery and Battery Charging 18](#_Toc419362930)

[2.5 Power supply implementation 19](#_Toc419362931)

[2.6 SD CARD 19](#_Toc419362932)

[2.7 16 x 2 Character LCD 20](#_Toc419362933)

[2.8 Miscellaneous 21](#_Toc419362934)

[2.8.1 Buttons 21](#_Toc419362935)

[2.8.2 Power On/Off Switch 21](#_Toc419362936)

[2.8.3 Buzzer 21](#_Toc419362937)

[2.8.4 Mechanical 22](#_Toc419362938)

[2.8.5 Wall adaptors 22](#_Toc419362939)

[2.9 Debug Options 23](#_Toc419362940)

[2.9.1 Modem Debug 23](#_Toc419362941)

[2.9.2 Microcontroller Debug 23](#_Toc419362942)

[3 Software Implementation 24](#_Toc419362943)

[3.1 Drivers 24](#_Toc419362944)

[3.2 Middleware 25](#_Toc419362945)

[3.3 Application 25](#_Toc419362946)

Figures

[Figure 1‑1 - Thermal Canyon high level block diagram 7](#_Toc419362947)

[Figure 2‑1 - Thermal Canyon overview 13](#_Toc419362948)

[Figure 2‑2 – Thermistor with the housing 14](#_Toc419362949)

[Figure 2‑3 - Temperature sensor and signal conditioning 15](#_Toc419362950)

[Figure 2‑4 - Modem - Microcontroller interface 17](#_Toc419362951)

[Figure 2‑5 – External Antenna 18](#_Toc419362952)

[Figure 2‑6 – Battery Charging Block Diagram 18](#_Toc419362953)

[Figure 2‑7 - Power supply block diagram 19](#_Toc419362954)

[Figure 2‑8 - SD Card SPI Pin Out 20](#_Toc419362955)

[Figure 2‑9 - 16 x 2 Character LCD with White Backlight 21](#_Toc419362956)

[Figure 2‑10 – Thermal Canyon mechanical backside 22](#_Toc419362957)

[Figure 2‑11 – Adapter Option 1 23](#_Toc419362958)

[Figure 2‑12 – Adapter Option 2 23](#_Toc419362959)

[Figure 3‑1 – Software high level architecture 24](#_Toc419362960)

Tables

[Table 1‑1 - Thermal Canyon requirements 8](#_Toc419362961)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Document Number | Revision Number | Description | Revision Date |
| XXXXX | 0.1 | Initial Release | Nov 2014 |
| XXXXX | 0.3 | Update after gerber out of prototype (Fab 1) | January 2015 |
| XXXXX | 0.31 | Updated with the platform name – Thermal Canyon | February 2015 |
| XXXXX | 0.5 | Updated s/w based on chages for Demo 1 | March 2015 |
| XXXXX | 0.9 | Updated H/W and S/W based on changes for Demo 2 | May 2015 |

§

# Introduction

Thermal Canyon is the code name of the temperature sensing device which is inteneted to measure temperature inside a refrigerator. In addition to measuring temperature, it has the capability to store the temperate data, send the temperature data log over a cellular network and alert the user by sending a text in cases where the temperature of the refrigerator crosses the pre-defined thresholds

1. The requirement document is, by its nature, iteratively developed. Refer to the latest version for the final implementation details.

## Terminology

| Term | Description |
| --- | --- |
| TS | Temperature sensor |
| M2M | Machine to Machine |
| NTC | Negative temperature coefficient |
| AT | Attention |
| uC | Micro controller |
| ADC | Analog to Digital Converter |
| SD CARD | NVRAM for storing temperature data |

## Background

Thermal Canyon has three major components – Temperature Sensor, microcontroller with ADC for converting, storing and displaying the temperature data in digital form and a modem to transmit data over a cellular network.



Figure 1‑1 - Thermal Canyon high level block diagram

This a high level block diagram of Thermal Canyon. As shown in the block diagram above, the thermistors (min 1, max 5) are housed inside the refrigerator. The signal conditioning circuit /amplifier amplifies voltage which represents the temperature and feeds it to the ADC of the microctroller (uC). The uC samples the thermistor after a fixed duration and stores the temperature data in SD CARD. The uC also transmits the temperature data over the cellular network through the GSM/GPRS M2M modem module. A 16x 2 Character LCD displays the temperature for each channel, battery level, signal strength, data/time, GPRS, SIM in use and Alerts. This unit is normally powered by the wall adaptor. A battery back-up option is provided in case of power outage. Battery capacity is monitored by a fuel gauge. LCD / LEDs and buttons are used to indicate error conditions and provide input.

Refer the subsequent chapters of this document for detailed description

## Thermal Canyon Requirements

The table below lists all the requirements for Thermal Canyon

Table 1‑1 - Thermal Canyon requirements

| Req# | Title | Description | Comments |
| --- | --- | --- | --- |
| R01 | Band Support | Quad Band | Supported  Telit GL865 – Quad supports all four GSM bands |
| R02 | Buttons | On/Off | A separate on/off button is provided for turning on/off the unit mainly during instation or maintanence |
| Stop Alarm | Button is provided to acknowledge that alarm by the user. This can be used to snoozer / turn of the Alarm as implemented in the code (At present code implents this as a snooze button) |
| Change Display / Screen Toggle | Button toggle the display on the screen to show different information like temperature of each channels, alerts, battery, sim, etc. |
| Extra Button | TBD |
| R03 | Battery | Built-in rechargable Li-Poly | Supported. Prototype will have a 750mAh battery |
| R04 | SIM | Dual Micro SIM | Supporting a push push type micro SIM connector  SIM will switch based on selection provided from the server or based on network status |
| R05 | Temperature sensor Accuracy | ±0.5°C or better within the range -30°C to +20°C | Supported |
| R06 | Resolution | ±0.1°C or better within the range -30°C to +20°C | Supported |
| R07 | Temperature Range | – 30 °C to 50 °C | Supported |
| R08 | Operating temperature Range of device | 0 °C to 50 °C | Supported |
| R09 | Battery Life | 12 hours or more | supported |
| R10 | Storage Capacity | 7 days of data at 1 sample per 10 minutes  Update: Data should be stored for 3 months for all the 5 temperature sensors, including power availability, when sampling once every 15 minutes | 2GB FAT 16 formted SD card supported. Can store much more data then required. |
| R11 | Number of sensors | 5 sensors | Supported |
| R12 | Sensor Connector | 3-pin headset plug | Supported |
| R13 | Sensor type | Analog NTC thermistor | Using Vishay NTCASRFE3C90406 |
| R14 | Sensor cable | For 3m – 2 condutor flat flex  For 20m- any cable | Supported |
| R15 | Sensor Calibration | Allow linear calibration parameters for ADC based of fixed thermistor inputs. Optionally allow calibration parameters for probe. | Supported |
| R16 | Certification | CE, FCC | WIP |
| R17 | Mode of operation | GPRS mode: where data is sent and configured via GPRS  SMS mode: where data is sent and configured via SMS | Supported |
| R20 | Sensor Data | • Start time (UTC timestamp)  • Sampling interval. (integer, mins)  • Temperature reading (float, degrees C)  • Battery level (percentage)  • Power status: Not plugged in, Charging, Plugged in but not charging (0, 1, 2)  • Include IMEI and phone number | Supported |
| R21 | Upload process: Normal transimission | • The device should upload data in batches. It should upload data if any of the following conditions are met:  • If current temperature is > HITEMP or temperature is < LOTEMP (excursion)  • Upload at ALERT\_UPLOAD\_INTERVAL  • Ohterwise upload every UPLOAD\_INTERVAL  • If there is a break in sampling (for example battery draining). (Then a new POST or SMS should be made.) | Supported |
| R22 | Upload Process: Failed Transmission | • If a transmission fails, retry up to 3 times with GPRS  • SMS retry based on hardware indicating failed transmission.  • Delete successfully transmitted data older than 7 days from storage.  • Keep all non-successfully transmitted data. | Supported |
| R23 | Upload Process: SMS mode | • Suggested SMS Formats/Compressions can be provided (involves byte shifting and addition).  • Data should be uploaded whenever there is a full SMS message or when time since upload is > UPLOAD\_INTERVAL (whichever comes first) or ALERT\_UPLOAD\_INTERVAL if there is currently an excursion.  • Retransmission on failure up to 3 times. The wait until the next UPLOAD\_INTERVAL to try again. | Supported |
| R24 | Data retention | • If uploads keep failing and the device storage is full delete the oldest data sample to make room for the newest data sample.  • Update: Data should be stored for 3 months for all the 5 temperature sensors, including power availability, when sampling once every 15 minutes | WIP |
| R26 | Time synchronization | Sync RTC with GSM | Supported |
| R27 | Configuration Parameters | Refer requirements document for more details | Supported |
| R28 | GPRS mode data upload protocol | Refer requirements document for more details | Supported |
| R29 | SMS mode data upload protocol | Refer requirements document for more details | Supported |
| R30 | SMS commands | The device should accept SMS commands to perform the following task:  • Update all configuration parameters  • Update APN  When an SMS command is issued, the device should attempt and HTTP POST with all the configuration settings and APN settings to the Upload URL.  Protocols *TBD*  Refer requirements document | WIP |
| R31 | ADC Calibration | Refer requirements document for more details | Supported |
| R32 | Full System Calibration | Refer requirements document for more details | Supported |
| R33 | Antenna performance | Antenna should have the best possible coverage | Prototype has both PCB trace antenna (for lower cost) and an external antenna. Based on the feedback from field trail, it will be decided which antenna will be used |
| R34 | Power supply | Use a simple 5V DC supply. More details [here](http://www.current-logic.com/shop/index.php?main_page=product_info&cPath=38&products_id=311) | Supporting a micro-B USB receptacle |
| R35 | Field replacable battery | Device should support a field replacable battery | WIP for Volume Manufacturing |
| R36 | Sensor data to temperature conversion | Recommendation is to implement the Steinhart algorithm for accurate temperature sensing | Supported |
| R37 | Reset/Reboot | A provison to reset/reboot the device | Power On/Off button is provided. Additionaly a separate RESET button is provided for debug on the board (not accessbile on the chassis) |
| R38 | External display | LCD 16x2 Character Display | Supported |
| R39 | Audible alarm | Alarm must be loud and must have provision to acknowledge. | Supported with Buzzer |
| R40 | Remaning battery capacity | Device when running on battery should monitor and transmit the remaining battery capacity over cellular netwrk | Fuel Gauge supported on the prototype to monitor the remaining battery capacity. Supported |
| R41 | Field update capability | Device should support firmware update on the field through UART BSL. JTAG update is not preferred. | Both JTAG and BSL are provided. Onlyy BSL is avaiable with the chassis. |
| R42 | Operation in case of dead battery or battery absent condition | The devices should be capable of powering up in case of dead battery/battery absent condition when the wall adapter is plugged in | Supported. |

§

# Hardware Implementation

This chapter outlines the hardware implementation of Thermal Canyon. The three major components are

1. Temperature Sensor: A NTC thermistor with a value of 10k is used for this design. The number of sensors supported in a system can vary from 1 to 4.
2. Microcontroller: MSP430 is used for this application. MSP430’s ADC is used to
   1. Convert the temperature data to digital form.
   2. Store the temperature data on a SD Card
   3. Send the temperature data to the modem to be transmitted over the cellular network.
   4. Perform other functionality like – battery charging, fuel gauging, handling button inputs, LCD and LED indication.
   5. Display the temperature of each channel, date/time, battery capacity, connectivity status and error conditions
3. Modem: A Telit GSM/GPRS Modem (Part# GL865-QUAD V2) provides the cellular connectivity.

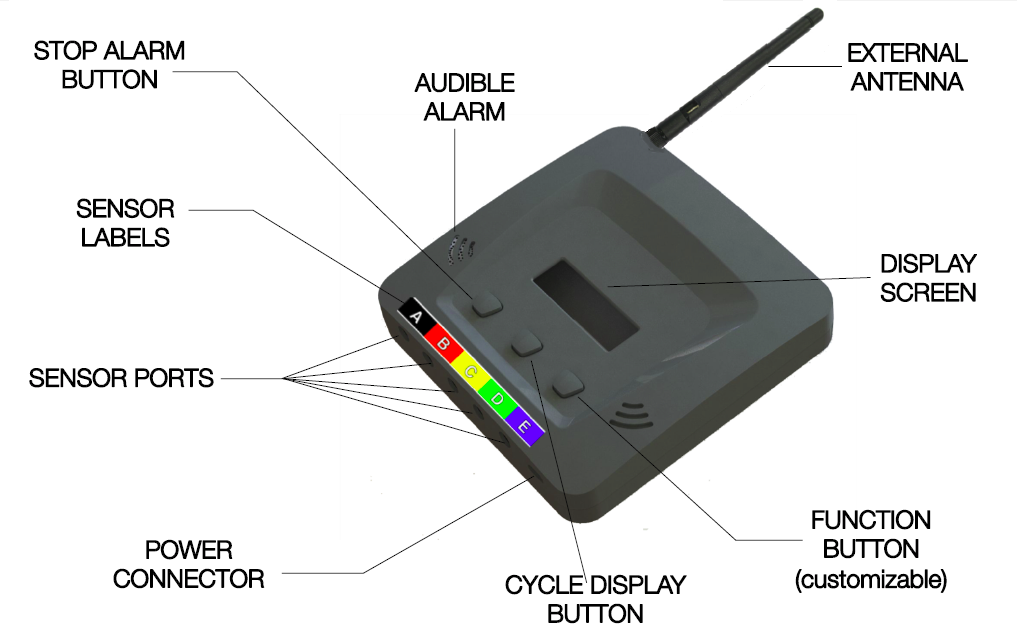


Figure 2‑1 - Thermal Canyon overview

## Temperature Sensor

Temperature sensor is the key element of Thermal Canyon. NTC thermistor module from Vishay is used for this design. Part # NTCASRFE3C90406.

This thermistor is available in an insulated package Key features of this thermistor are

* Sensor design following class II insulation (principal + supplementary insulation for the sensor head)
* High adhesive strength between silicone cable and encapsulating lacquer
* Specifically developed design allows for a very good water, moisture and ice resistance (min. 1000 h water immersion)
* Suitable for evaporator temperature measurement. Very high number of thermal cycles resistant (min.100 000 cycles)
* The cables jackets are suitable for back-panel polyurethane foaming process (max. 100 °C, 5 min)
* Compliant to RoHS Directive 2002/95/EC and in accordance to WEEE 2002/96/EC
* Surface temperature sensors
* The housing and cable are cold flexible at - 60 °C

• The housing plastic is FDA grade

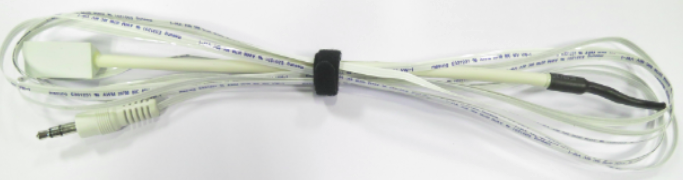


Figure 2‑2 – Thermistor with the housing

### Sensor design

ISL60002 from Intersil with voltage set at 2.5V is used as a reference for the ADC on the microcontroller. It also power the voltage follower opamp and the thermistor voltage divider. The opamp Part# is OPA348AIDBVR from Texas Insturments. This is a quad package and can be used for all the 4 temperature sensors.

The input to ISL60002 is from the 2.8V LDO on the platform. This device is capable of sourcing 7 mA.

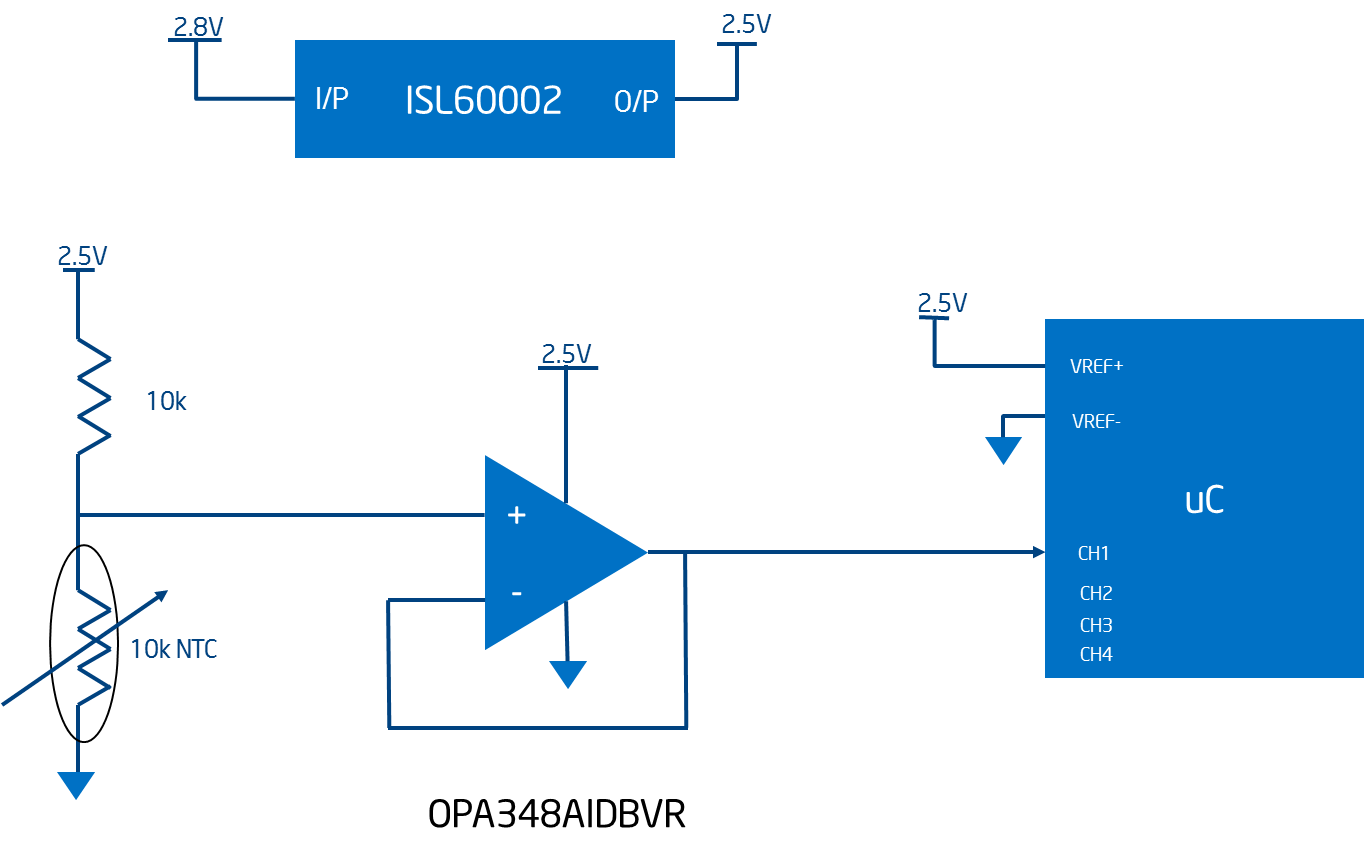


Figure 2‑3 - Temperature sensor and signal conditioning

## Microcontroller

A MSP430 series micro-controller from TI will be used for this design. The microcontroller is responsible for

* Reading the temprerature data from the thermistor through the ADC interface
* Storing the temperature data in the SD CARD
* Configuring the modem via UART through AT commands
* Sending the temperature data via UART to the modem
* Controlling the battery charger and fuel gauge
* Displaying temperature and other paramters on a 16x2 Character LCD

## GSM/GPRS Modem

A Machine to machine (M2M) module from Telit is used. Part# GL865-Quad V2. Key features of this module are

* Quad Band GSM/GPRS 850/900/1800/1900 MHz
* GSM/GPRS protocol stack 3GPP Release 4 compliant
* Control via AT commands according to 3GPP TS 27.007 and Telit custom AT commands
* SIM access profile
* Dual SIM (Fail Over Implementation)
* TC/IP stakc access via AT commands
* Real time clock
* Network Strength indication
* Embedded TCP/IP stack, including TCP, IP, UDP, SMTP, ICMP and FTP protocols
* PFM over the air update service
* Event monitor
* Asynchronous non-transparent CSD up to 9.6 kbps
* V.110
* GPRS Class 10
* Dimensions: 24.4x24.4x2.6 mm
* Extended temperature range from -40 deg to +85 deg C
* ITU-T V.24 serial link through CMOS UART. Baud rate from 300 to 115,200 bps

### Modem – Microcontroller interface

The modem interfaces the microcontroller of the UART interface. The microcontroller sends the AT commands to the modem over the UART interface. Modem acts as a DCE (Data circuit-terminating Equipment) and the microcontroller acts as a DTE (Data Terminal Equipment). All the I/O on the microcontroller and the modem side are powered at 2.8V (except for reset which is at 1.8V).

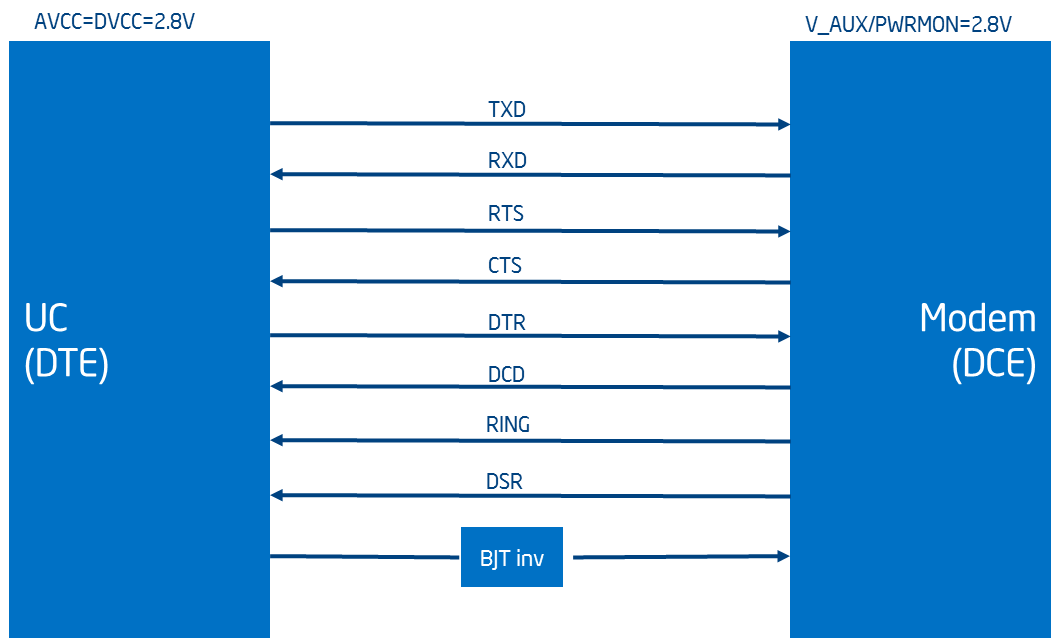


Figure 2‑4 - Modem - Microcontroller interface

TXD and RXD are the UART siganls. There are other side band signals which are required for communication be microcontroller and the modem. RTS, CTS, DTR, DCD, Ring, DSR and Reset are GPIO on the microcontroller. Except for reset, all are powered at 2.8V. Reset is powered as 1.8V. Reset signal is level shifted using a BJT inverted. Modem has an internal PU on Reset. Reset is used only during catastrophic condition. For normal power-up the modem has an internal POR circuit.

NOTE: Enusure that the GPIO on the microcontroller which is used for modem reset, drives a 0 on the pin during normal operation.

The modem is powered off the battery (3.1V to 4.2V). And LDO inside the modem module generates 2.8V for the I/O interface.

### Dual SIM

Two micro-SIM (push-push type) will be used for this application. Fail Safe mode of SIM switching is implemented. SIM can switch based on configuration information from the Server asking to switch SIM or based on network conditions. SIM slots must have restricted access such that only technician can access this with the right tools.

On the platform, the SIMs can be accessed by opening a lid which is screwed using hex/align key screws for restricted access.

### Antenna

By default a quad band patch antenna would be used Thermal Canyon.



Figure 2‑5 – External Antenna

Prototype will also have an option for PCB trace antenna.

The housing (mechanical chassis) will be made up of plastic

## Battery and Battery Charging

Thermal Canyon will have a 750 mAh Li-ION battery.This will provided wel over 12hours of backup power (requirement is for 12 hours backup only). The device should also inhibit very low battery operation for protecting the battery and extended battery life.

The remaining battery capacity (in %) and the status of AC mains will to be transmitted over cellular network along with the temperature data. The battery capacity is also display on the LCD along with the charging status. This requirement calls out for the need of a battery charger and fuel gauge for monitoring the battery capacity.

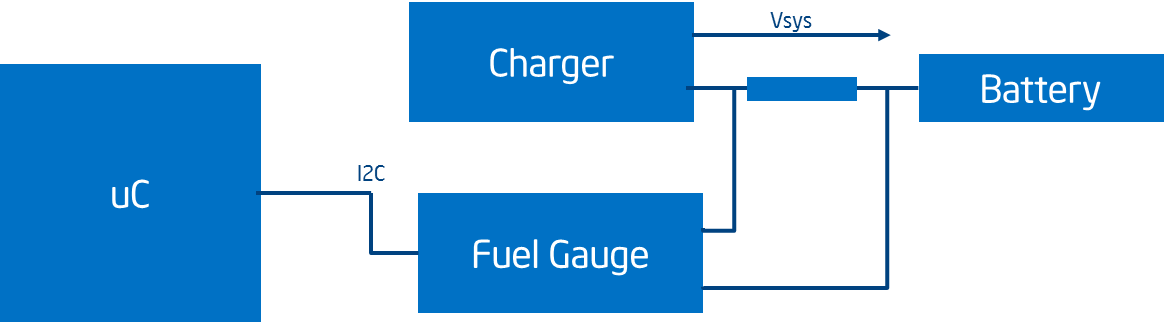


Figure 2‑6 – Battery Charging Block Diagram

Battery Part# *TBD*

Charger Part# BQ24074RGTR from Texas Instruments. This is a standalone LI-ION charger and doesn’t need any configuration from the host. Its adaptive learining algorithm learns the battery characteristics and capacity over time giving a very accurate estimate of the battery capacity.

The system can operate without the battery or in a dead-battery condition when external power is available. Ensure the power supply is capable of providing the peak power (2W) required during data trasnsmission over cellular network.

Fuel Gauge Part# BQ27441DRZT-G1A

Battery must have restricted access such that only technician can access this with the right tools. On the platform, the battery can be accessed by opening a lid which is screwed using hex/align key screws for restricted access.

Battery should be field replacable LI-ION battery. Prototype will not have a field replacable battery.

## Power supply implementation

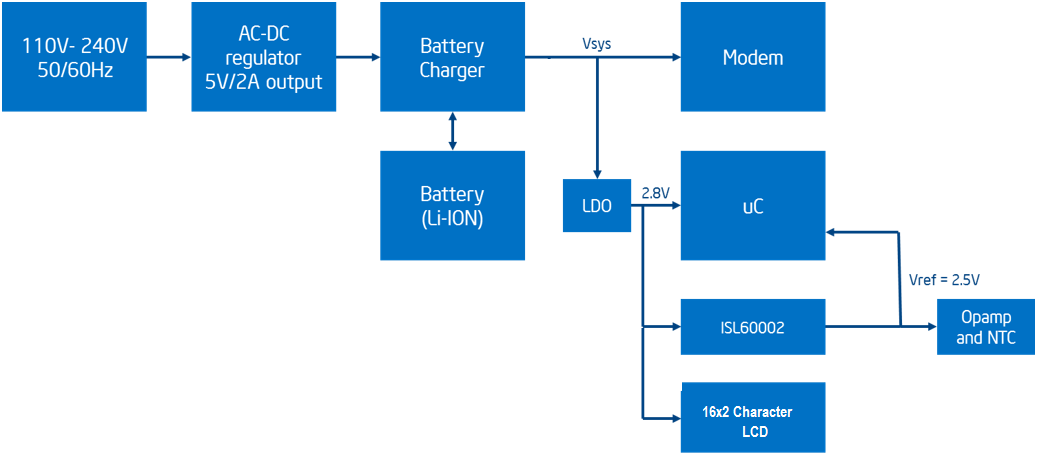


Figure 2‑7 - Power supply block diagram

## SD CARD

The SD CARD is used to store the temperature samples for each channel, time, battery levels, charging status, etc on this device. As the MSP430 does not support SD card interface by default, SD card will be used in the SPI mode (This is not a requirement but a limitation for MSP430). Due to RAM constraints on MSP430, we will use FAT16 format and support only 2GB SD Card.

The SD Card data must be stored in csv format and must be structured for easy viewing of data. The SD card must be able to store 3 months of data backlog. With 2GB SD card provided, the device would be able to store many many months of data (very much more than required).

SD Card must have restricted access such that only technician can access this with the right tools. On the platform, the SD Card can be accessed by opening a lid which is screwed using hex/align key screws for restricted access.

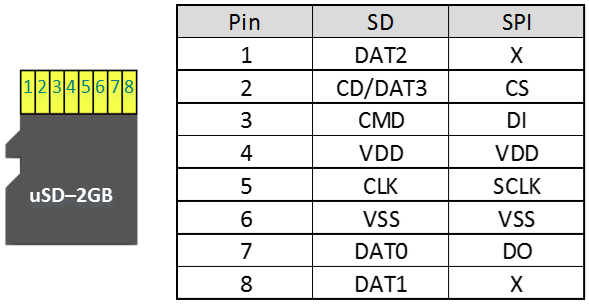


Figure 2‑8 - SD Card SPI Pin Out

## 16 x 2 Character LCD

The prototype will support a 16 x 2 Character LCD with white Backlight. It will support I2C interface and the characters will be displayed by sending the ASCII code. Requirment is that the display should be readable from a distance of 1.5 meters.

Display requirements:

1. The display page can be toggled to show multiple information by pressing a button.
2. Each display page must show the data and time on the top.
3. Display must be readable at 1.5 meters.
4. The default main screen must show the Sensor B Temprature, Battery level, Antenna Strength and GPRS/no GPRS.
5. All sensor temperature and alert conditions should be avaible by toggling through the screens
6. Should show battery level and charging status.
7. Should show which SIM is connected and its signal strength and GPRS avaiable or not.
8. Backlight must be always on. This will also indicate device is functional.

**Display Panel Selected:** NHD-C0216CiZ-FSW-FBW-3V3 (Newhaven Display)

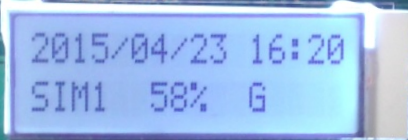


Figure 2‑9 - 16 x 2 Character LCD with White Backlight

## Miscellaneous

### Buttons

System will have four push buttons.

1. 1st button is to stop/snooze the alarm
2. 2nd button is to toggle throught the many display windows.
3. 3rd button is TBD and can be repurposed to do a soft reset.
4. 4th button is avaiable for doing a hard reset. However, this button is not avaible or user as it is not accessible outside the enclosure.

### Power On/Off Switch

Toggle Switch for Power On/Off is provided within enclosure. This will allow the device to be turned off for installation or maintenance.

The On/Off switch must have restricted access such that only technician can access this with the right tools. On the platform, the switch can be accessed by opening a lid which is screwed using hex/align key screws for restricted access.

### Buzzer

Buzzer is provided to generate audible alaram in case of some breach in temperature or power condition. The buzzer must be loud and clearly audible.

### Mechanical

The enslosure will be made of ABS plastic (not supported for the protos)

Access to the electronics will be restricted as the screws will be hex/allen keyed screws.

Restricted access to the the SD card, SIMs, power button, battery will be possible by opening a lid which will again be screwed using hex/allen keyed screws.

Multiple wall mounting mechanisams are provided. Option to directly nail the unit to wall will be provided with a metal mounting plate. Another option to hang the unit on two pre fixed nails/screws is also provided.

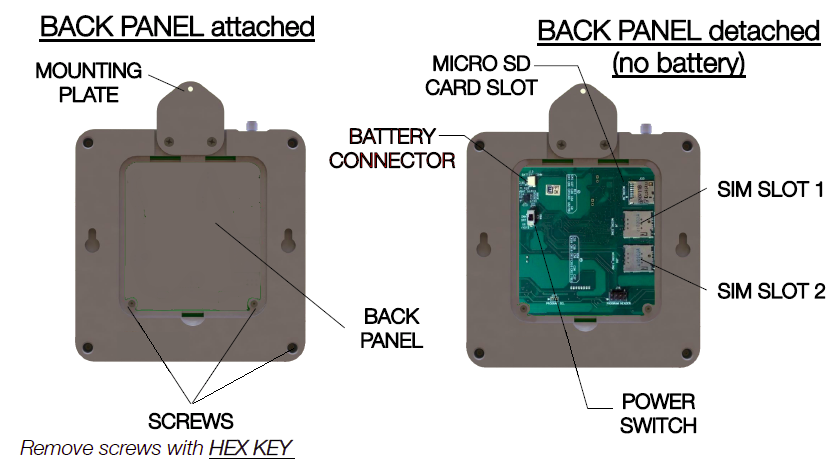


Figure 2‑10 – Thermal Canyon mechanical backside

### Wall adaptors

There are three option for the wall adaptors. Option 3 will be supported on the prototype and all the future iterations.

Option 1:

Use a standard 5V 1.5A wall adapter with a barrel jack. The adapter which will be chosen will be certified



Figure 2‑11 – Adapter Option 1

Option 2:

The unit will have 3-pin IEC male socket. An AC to DC supply will be mounted on the PCB. This supply will take 100V-240V AC input at 50/60 Hz and supply and generate 5V output required for the operation of the modem and microcontroller.



Figure 2‑12 – Adapter Option 2

Option 3:

Details of the power supply are given [here](http://www.current-logic.com/shop/index.php?main_page=product_info&cPath=38&products_id=311). Thermal Canyon will support a micro-B USB receptacle to support this power supply

Option 3 will be used for Thermal Canyon

## Debug Options

### Modem Debug

Test Points for UART debug will be provided on the PCB

### Microcontroller Debug

JTAG connector will be provided on the PCB for microcontroller debug

BSL header is provided for programming of the microcontroller.

# Software Implementation

Main loop

Application

Middleware

FAT16

LCD

I2C

UART

HTTP

ADC

Timer

SMS

MMC

Driver

RTC

Hardware (TI MSP430FR5969, Telit GL 866 QUAD)

Figure 3‑1 – Software high level architecture

The above figure depicts high level software architecture.

The software for the data logger is an embedded main loop that interacts with peripherals to

* periodically sample temperature values,
* logs the samples to a file,
* periodically uploads to server through GPRS and SMS
* periodically checks for configuration updates from server
* periodically checks for SMS message for custom operation (e.g STATUS query, RESET)

The software comprises of following components

## Drivers

MMC

Provides functionatily to initialize, read and write (in sectors) to the SD-card by using SPI interface (*eUSCI\_A1*).

Note: Sector size is limited to 512 bytes.

I2C

Provides functions to initialize I2C bus (*eUSC1 B0*), transmit and receive from multiple slaves (e.g Fuel gauge, LCD).

UART

Provides functions to initialize serial interface (*eUSC1 A0*), transmit and receive from GPRS modem.

Provides functions to get response to AT commands (e.g CCLK, CSQ, CMGR)

ADC

Provides functions to sample ADC channels (maximum 5) in a sequence.

Timer

Provides delay functions in resolution of 10 ms and 250 us.

RTC

Provides functions to initialize RTC, to generate minute change events and to return UTC and local time.

## Middleware

FAT16

This layer is reused from <http://elm-chan.org/fsw/ff/00index_e.html>.

Note: It is observed that this file system allows creating duplicate files at root directory. Workaround is to use sub-directory and stores files only in sub-directory.

SMS

Provide functions to send SMS to gateway, to a specific phone number and to process the received SMS messages.

Note: SMS messages received should follow pattern such as start tag $ST and end tag $EN. Due to RAM limitation, SMS message should not exceed 75 characters.

HTTP

Provide functions to perform HTTP POST and HTTP GET.

Note: HTTP POST is retried for one more time in case of POST failure. The maximum timeout for POST response from server is 20 secs.

## Application

Main loop

**Step1**: “WDTCTL = WDTPW | WDTHOLD;” This command Disables the watchdog Timer.

**Step2**: Initializing the port for SPI CS, MISO, MOSI, SCK gpio pins.

**Step3**: Configures SPI at 400 KHz, Master mode, MSB.

**Step4**: Configures UART at 115200, one Start, one Stop, No Parity.

**Step5**: Configures ADC channel sequences, pulse mode, 12 bit resolution.

First sample by trigger and then reset automatic trigger by

prior conversion.

**Step6**: Initializes the Ports: LCD reset, Backlight enable.

**Step7**: Initializing the battery via I2C protocol.

**Step8**: Configures internal DCO clock at 8MHz.

**Step9**: Initializing the buzzer gpio pin.

**Step10**: Initializes modem for the configured SIM slot.

**Step11: Sampletemp();** This function is used for taking average of five sensor sample value from 100 consecutive samples of each sensor.

**Step12: uart\_tx("AT+CSQ\r\n");**  This command returns the signal strength of the device which indicates the signal condition is marginal or not.

**Step13: dopost\_gprs\_connection\_status(GPRS); :** Command returns the GPRS network registration status.

**Step14: uart\_tx("AT+CGSN\r\n");** Command returns the  the IMEI (International Mobile station Equipment Identity) of the mobile terminal.

**Step15: uart\_tx("AT+CCLK?\r\n");** command is used for getting the internal RTC of the modem.

**Step16: sendhb();** : Sends heartbeat to server in following format

10,IMEI,sim slot, gateway, s0,s1,s2,s3,s4,batt%,power plugged.

**Step17: f\_mount(&FatFs, "", 0);** : It mount the fatfile system to the default drive.

**Step18: dwLastseek:** This variable is used for getting the last read offset value from FRAM.

**In a while loop**

Step19: Checking whether the adc conversion is completed or not by using **isConversionDone** variable.

Step20: Triggers ADC sampling data at the configured sample period (default one minute).

**ConvertADCToTemperature(ADCvar[iIdx],&Temperature[iIdx], iIdx);**

This function is converting the current sensor adc value to temperature.

**if**((iMinuteTick - iSampleTimeElapsed) >= g\_iSamplePeriod) is used for taking average of five sensor sample value from 100 consecutive samples of each sensor.

**Step21: logsampletofile(&filw,&iBytesLogged);** This function using for samples are log to a file with rtc time.

The samples are logged to a file in the following format

$TS=20150406:05:37:33,R01,F100,P1,A--.-,B-05.2,C16.5,D26.5,E-15.1,F100,P1,A--.-,B-05.1,C16.2,D26.0,E15.5, (where TS = time stamp, R = Sample period, F = Fuel gauge or battery level, P = Power plugged status, A = sensor A temperature, B = sensor B temperature, C = sensor C temperature, D = sensor D temperature, E = sensor E temperature.Note the temperature value is --.- in case the sensor is not plugged in.).

**Step22: monitoralarm();** This function is using for Monitors the

temperature thresholds and triggers buzzer in case of violation

for the configured alarm period.

**Step23**:

**if**((((iMinuteTick - iUploadTimeElapsed) >=

g\_iUploadPeriod)||(iStatus & TEST\_FLAG) ||

(iStatus & BACKLOG\_UPLOAD\_ON) ||

(iStatus & ALERT\_UPLOAD\_ON)) &&!(iStatus & NETWORK\_DOWN))

This condition is checking for the upload configured time

interval and network signal strength status.

If the condition is satisfied then opening the file

and reading from sd card by using f\_open and f\_read function.

It formats the logged data into HTTP POST message in following format IMEI=353173063204364&ph=8455523642&v=1.20140817.1&sid=0|1|2|3|4&sdt=2015/03/03:08:22:08&i=1&t=25.1,26.1,25.9,24.3,23.7,25.1,26.1,25.9,24.3,23.7|25.1,26.1,25.9,24.3,23.7,25.1,26.1,25.9,24.3,23.7|25.1,26.1,25.9,24.3,23.7,25.1,26.1,25.9,24.3,23.7|25.1,26.1,25.9,24.3,23.7,25.1,26.1,25.9,24.3,23.7|25.1,26.1,25.9,24.3,23.7,25.1,26.1,25.9,24.3,23.7&b=100,100,100,100,100,100,100,100,100,100&p=1,1,1,1,1,1,1,1,1,1 (where IMEI is device unqiue number, ph is phone number, v is verison, sid is list of sensor ids, i is sample period, t is temperature values for each sensor per sample period, b for battery levels per sample period, p for power plugged status per sample period)

**Step24:** At the configured upload period (default 10 minutes), it encodes the sampled data for SMS in the following format 11,20150303:082208,interval,sensorid,DATADATADATAT,sensorid,DATADATADATA,sensorid,dATADATADA,sensorID,DATADATADATADATAT, sensorID,DATADATADATADATAT,batt level,battplugged.

**Step25: dopost\_sms\_status(); :** Check whether the sms has been successfully sent or not by using this variable **file\_pointer\_enabled\_sms\_status**

**if**((file\_pointer\_enabled\_sms\_status)||(file\_pointer\_enabled\_gprs\_status))

This condition is used for checking either GPRS or SMS is successfully sent or not.

**Step26**: If previous condition is true (GPRS OR SMS is sent) then the file pointer will move to the next packet.

FRAMCtl\_write8(SampleData,INFOB\_ADDR,**sizeof**(CONFIG\_INFOB)); This function indicates saving the last read offset by using this function.

Maintains file pointer in FRAM to indicate the postion of next read location.

In case of backlog, it performs upload of backlog data at every configured sample period.

**Step27**: At the configured cfg update period (default 5 minutes), or

**if**((iMinuteTick - iSMSRxPollElapsed) >= SMS\_RX\_POLL\_INTERVAL)

It pulls the configuration information from server and updates the parameters in FRAM. if there is any change and send heartbeat on successful processing of SMS message by using

sendhb(); function.

**Step28**: Updating the signal strength by using the **uart\_tx("AT+CSQ\r\n");** function.

**Step29: dopost\_gprs\_connection\_status(GSM); :** Command returns the GSM registration status.

**Step30: if**((gprs\_network\_indication == 0) ||((iSignalLevel <= NETWORK\_DOWN\_SS) || (iSignalLevel >= NETWORK\_MAX\_SS)))

This condition is used for checking the signal strength whether it is either less than 14db or greater than 31db then sim switching will happen and modem get reinitiate for avoiding modem error state.

modem\_init(pstCfgInfoA->cfgSIMSlot); **:** Reiniting the modem by using this function and updating the new sim registration status to FRAM by using this below function.

FRAMCtl\_write8(pstCfgInfoA,INFOA\_ADDR,**sizeof**(CONFIG\_INFOA));

**Step31**: This condition is using for lcd update :

**if**(((iMinuteTick - iLCDShowElapsed) >= LCD\_REFRESH\_INTERVAL))

In the idle period, it updates display on every minute.

**lcd\_show(); :** This function is using for data updates on every minute.

It updates displays in the following modes on the channel selection button event

mode 1 - Displays date time, battery percentage, sensor B temperature, 'network signal strength, GPRS network registration status'

Mode 2- Display all 5 temperature sensor readings

Line1:   2015/03/10 13:44

Line2:  Sensor A -31.5C  ← this is for temperature of channel A with no alert

Line2:  Alert A -31.5C ← this is for a temperature alert on channel A

Line2: 57% POWER OUT ← this is for when power is out

Line2: 14% BATT ALERT ← this is for when battery low

Line2: 56% CHARGING ← this is for no power and battery alert.

Line2: SIM1, Signal strength percentage.

Line2: SIM2, Signal strength percentage.

**Step32**: At the configured SMS message check period (default 2 minutes and only if there is gap of 2 minutes before the next cfg processing or upload processing cycle can happen), it process the received SMS text for commands such as STATUS and RESET. For STATUS, it responds with the current temperature readings of all sensors. For RESET, it trigger software *BOR* request.

**Step33: if**(iStatus & BUZZER\_ON) : This condition is using for enable the buzzer for 5 to 10 seconds by setting and resetting one configured gpio pin.

**Step34**: In the event of low power (such as battery level less than 10%), it displays “Low Battery”, disables LCD, sampling and posting of data. It waits till power is plugged in, on which it enables LCD, re-initalize the modem and continues to sample and post data.

§

§