Index

Abstract

The work presented in this thesis is dedicated to the development of a remote sensor for HUMSAT which is a development of a nano-satellite constellation .
The HUMSAT system architecture is composed of three segments ; space segment , ground segment , and user segment . the user segment formed by the sensors freely deployed and developed by users and by the facilities that users shall design and construct by their own in order to retrieve and send the data.
Data transmission between sensors and the satellite could be Non-bi-directional , bi-directional and full-bi-directional , this thesis describes the non-bi-directional data transmission so a single sensor transmits continuously the frames that it has generated previously until a spacecraft collects them .
A temperature sensor (NTC thermistor) has been created in order to take part of the user segment , for this goal it has been designed an electronic circuit and its' PCB , using mentor circuit designer , the circuit is built of a temperature sensor which is connected with a microcontroller (MSP430F5438) and a transceiver (Si4464) , the sensor will relieve the temperature every a defined period and send it to the microcontroller , the microcontroller will recognize if this temperature is in a certain defined range , if the temperature is out of the range the microcontroller will communicate the transceiver to send the data packet .
Some specifics has to be taken into consideration for the data transmission , frequency band is 401-402 MHz , GMSK modulation , EIR is about +27 dBm , for these specifics the Si4464 transceiver has been chosen , the output power of the transceiver was not sufficient , in order to increase the output power of the transceiver a power amplifier is implemented .
Two RF switches is implemented to isolate the transmission and reception chains .
For low power consumption , it is thought to deactivate the transceiver and the PA when it is not necessary to be active , like while the microcontroller is still reading temperature from the sensor , and if the temperature is in the range defined .
In order to configure the microcontroller , the transceiver , and generally reading and transmitting data , C programming codes is written in IAR Embedded workbench .

CHAPTER I

Introduction

In the current day and age, connectivity is one of the key elements in both day-to-day life and in business. Instant information is not only demanded by users in a myriad of applications, it is also expected. New applications, made possible by huge investments in optical fiber networks and 3G communication networks, are now becoming available in urbanized areas all around the world. However, the great challenge for the coming decade is to remain connected in remote areas where no ground based communications infrastructure is available. In order to track and monitor cargo, equipment or other assets on a global scale in a similar way as is now possible in industrialized nations a new solution for fulfilling the information needs of modern businesses is required.

many new projects are developing global tracking and monitoring systems particularly aimed at remote areas of the world such as the open ocean, sparsely inhabited areas, developing nations without extensive ground infrastructure. Using low cost space infrastructure, a network of small satellites and strategically placed ground stations, it is possible to offer global tracking, monitoring and tracing services at an unprecedented price/performance level.

The HUMSAT project is an initiative originated from the international cooperation of three universities: the University of Vigo (**www.uvigo.es**), the California Polytechnic State University (**www.calpoly.edu**) and the National Autonomous University of Mexico (UNAM - **www.unam.mx**).

This project has been supported, from the beginning, both by the Education Office of ESA (through the GEOID initiative) and by the United Nations Office of Outter Space Affairs (OOSAUN).

One of the main objectives of the HUMSAT project is to promote the international cooperation among different universities from all over the world, several entities from different countries are expected to join the HUMSAT project; from universities both from developed and developing countries, to aerospace companies which will provide technical support and expertise.To achieve these objectives, a community of developers will be established through a dedicated website; and a board composed of members from several different universities, will be the responsible for several management tasks like the registry of new sensors to be added to the network, the improvement of the interface definition documents for the HUMSAT system... etc. The universities from several ESA member states involved in the development and design of the project will act as lower-level suppliers, designing and constructing several systems for the mission, within the context of the GEOID initiative and under the coordination of the prime contractor of the system. Furthermore, other universities from non-ESA member states will also be allowed to create sensors and satellites for the HUMSAT system, even though they were not part of the GEOID initiative.

The main purpose of the HUMSAT system is the development of a satellite-based system for connecting a set of users with a network of worldwide distributed sensors which they have previously deployed.

Sensors will be responsible for acquiring user data and for transmitting it to the satellites through an standard radio interface (SSI). Users will be able to define their own sensors, for monitoring different types of parameters; for example, water temperature or wind speed. The spacecraft of the GEOID initiative will form the first backbone for the communications services provided by the system.

This way, several applications can be constructed based on a system that provides low data rate communication services; such as remote monitoring of climate change parameters (Earth's surface temperature, atmospheric pressure...).

This thesis provides a description for the SSI interface of the HUMSAT system. This description takes into consideration both sides of the interface (sensor and satellites) and establishes, therefore, a set of requirements that shall be met by both sides of this interface. Due to the functional requirements of the mission, two different communications links shall be defined: uplink communications (from sensors to the spacecraft, mandatory for all sensors and spacecraft) and downlink communications (from spacecraft to sensors, optional both for sensors and spacecraft) , in this thesis I describe the uplink communications and I try to develop a temperature sensor capable to interface with the spacecraft defined by HUMSAT .

Chapter II

HUMSAT project

The purpose of the HUMSAT project is the development of a nano-satellite constellation and its corresponding ground segments to provide support for humanitarian initiatives, especially in developing areas. Furthermore, the HUMSAT project will have strong educational objectives boosting cooperation between universities from different countries. The HUMSAT project is aiming to provide a wide range of applications such as climate change monitoring, remote disaster tracking or public health communications.

The HUMSAT project has been presented at several symposiums under the framework of the United Nations Program on Space Applications. In addition, the HUMSAT project has been **endorsed** by:

- European Space Agency.
- United Nations through the Office for Outer Space Affairs. (UN-OOSA)
- University of Vigo (Spain)
- California Polytechnic University (USA)
- Autonomous National University of Mexico and CRECTEALC (Mexico)

Among others, the HUMSAT project will have strong educational **objectives** such as:

- Provide hands-on-project experience on a space project to engineering/science students.
- Promote international cooperation between universities about space technology.
- Transfer technology from developed to developing areas.

In terms of main **functions**, the HUMSAT project will be capable of:

- Managing the worldwide sensor network.
- Communicating with the GENSO network.
- Accessing the data obtained from the satellite.
- Defining new experiments for the proposed payloads.

With regard to **configuration** the HUMSAT system is composed of:

- An space segment based on a constellation of CubeSat spacecraft.
- A ground segment composed of:

1. The ground stations included in the GENSO network.
2. Non-mandatory specific ground stations, additionally constructed by each university.
3. Additional data distribution facilities.

- A user segment composed of:

1. The sensors deployed all around the world.
2. User facilities for accessing the data gathered by the space segment.

2.1 GEOID Initiative

The GEOID (Genso Experimental Orbital Initial Demonstration) initiatives expected to be the **communications backbone** for the initial version of the HUMSAT system. The GEOID initiative is considered as the ESA contribution to the HUMSAT project, in fact it will be used with educational purposes and as a test-bed for the HUMSAT system.

Following the requirements extracted from the applicable documents from the HUMSAT system, the following statements are representing the GEOID initiative:

- Spacecraft will be based in the use of the CubeSat standard and will conform a constellation, initially composed of the **9 CubeSats.**
- Additional spacecraft (not only CubeSats) could be added to the constellation for improving or for maintaining the level of service achieved by this first constellation.
- Each CubeSat shall be operated through the **GENSO network** or through standalone ground stations, by an operations team designed by each university.

- All over the world widespread sensors will conform a network of autonomous sensors deployed, designed and maintained by users.
- Sensors will be gathering data and sending it back to the satellites.
- Communications between sensors and satellites will be established by sensors, which will periodically send short frames to the channel - S2U (Sensor-TO-User) service.
- The communications interface is named SSI (Space-Sensor Interface) interface.
- Users shall use the UHI interface for sending messages to the sensors through the U2S (User-TO-Sensor) service.
- Satellites will receive a set of appropriate telecommands from the ground segment for sending the message back to the sensors.

2.2 System architecture

The main global function of the HUMSAT system is to provide communication between users and a set of generic sensor platforms. The architecture needed in order to accomplish this purpose has been defined as a system composed of three segments and the corresponding interfaces between them. The three main segments are:

- **Space segment**, which is composed of several constellations and satellites which may act as a communications backbone for the system, offering coverage for the users/sensors to send data to the users/sensors.
- **Ground segment**, that will be used mainly for operating the satellites and for sending data to and from the user segment. This segment will be based in the extensive use of the services provided by the GENSO network (for communicating with several satellites)

and by the services of standalone ground stations developed, on their own, by additional users.

- **User segment**, formed by the sensors freely deployed and developed by users and by the facilities that users shall design and construct by their own in order to retrieve the data.

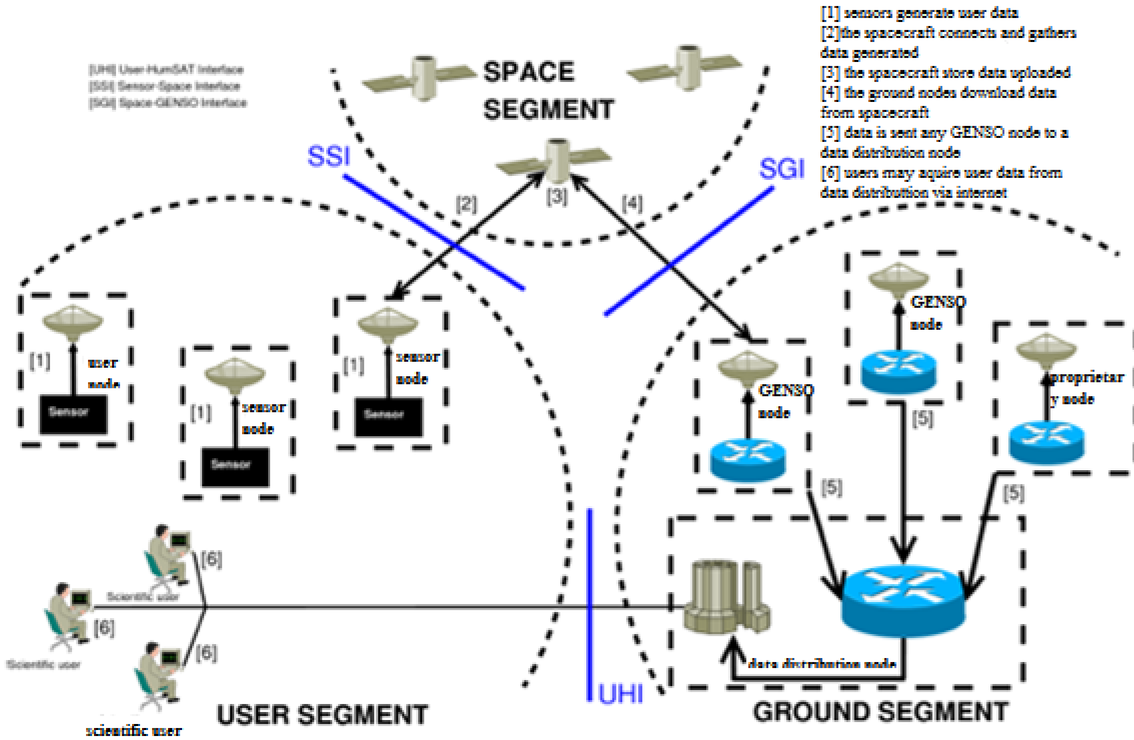


figure 2.1- from the ground segment of HUMSAT through the Internet-based interface.

2.2.1 Interfaces

Among these segments, the next main interfaces have been identified:

- **SSI** (Sensor-Space Interface): an RF interface between the user-deployed sensors an spacecraft that will serve as the communications link between the space segment and the user segment, both for uploading data from the sensors to the spacecraft and for downloading data from the satellites back to the sensors.
- **SGI** (Space-Ground Interface): this is another RF interface that will serve as the communications link between the space and the ground segment. Its main purposes will be to command and operate the satellite and to permit spacecraft operators to retrieve the data gathered by the spacecraft form the sensors.
- **UHI** (User-HUMSAT Interface): this interface is based on the use of the Internet and will allow users to access the services offered by the HUMSAT system. Internet services will be provided so that users will be able to download the data that the HUMSAT system has transported from its sensors and will also be able to add new data to be sent

back to the sensors like, for example, a new schedule for the sensors to make measurements with a different configuration, or with a different timing. Security access restrictions shall be included in this interface so that the system guarantees both authenticity and privacy of the data gathered.

2.2.2 SSI interface definition

The SSI interface shall manage the communications between the spacecraft of the HUMSAT system and the sensors freely deployed by the users of the system. This is a communications scheme based in an aerial interface in which a radio on-board at each spacecraft shall receive data from the radio of the sensors deployed around the globe. Therefore, not all deployed sensors will be visible for the spacecraft at a time: depending on the beamwidth of the on-board antenna, each spacecraft will be capable of receiving data from a different amount of sensors.

Most of the deployed sensors are expected to be low-cost low-processing-capacity devices. These two conditions involve that the definition of the SSI interface is expected to be easy and cheap to implement, providing only a short-messages exchange service between spacecraft and sensors.

Besides, spacecraft are also expected to send data to the certain types of sensors (**bidirectional sensors**). Since these kind of sensors might be switched off while spacecraft are passing by, spacecraft will only be able to send the data that they have stored for them after they receive data from the sensors.

In case users require to develop sensors which will require spacecraft to contact them without sensors send a DF frame, an optional sensors discovery mechanism can be implemented. Sensors that implement also this discovery mechanism will be known as **full bi-directional sensors**.

This be discovery mechanism shall triggered by sensors, which shall transmit a frame to permit spacecraft to discover them. Once spacecraft discover them, they will be able of sending them data that users had previously sent to the system.

The typical communications scenario that has been described above in this section is depicted in the figure below. In that figure, three different scenarios are described: a single-direction communication between sensors and spacecraft, a collision in the channel and a bi-directional

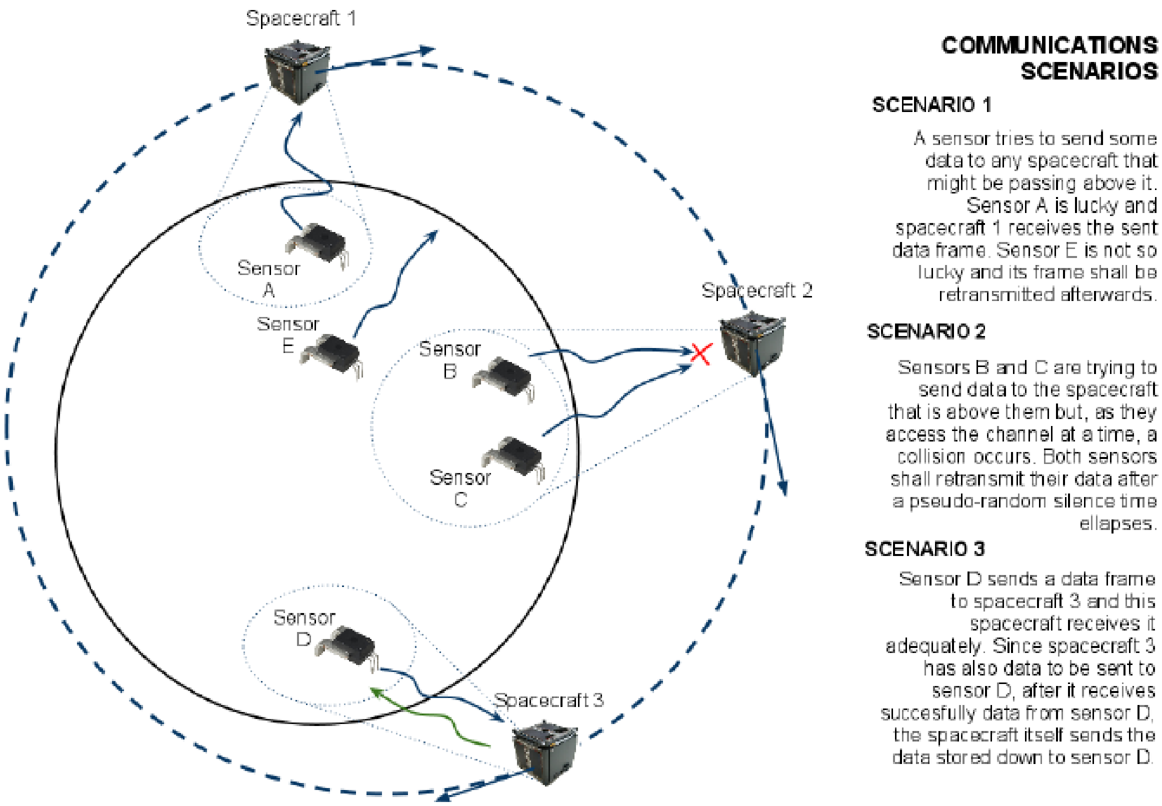


Figura 2.2 – communications scenario , spacecraft-sensors .

communication with a single sensor. Any other communications scenario can be described further as a combination of these three basic.

According to the previous definition of the SSI Interface, this one shall be implemented as a protocol stack that provides the next set of core functionalities:

1. Transfer data between sensors and spacecraft.
2. Detect any errors that may occur during the transmission of data between sensors and spacecraft.
3. Manage the access to the communications channel sensors-spacecraft.
4. Identify which data packet each sensor owns.

The transport of the data gathered from the sensors or the data to be sent to the sensors from the users, is not a direct responsibility of this protocol. Nevertheless, in order to use other protocols to reliably transport this data to the final user and to subsequently send the data to the owner of each sensor, each data set shall be uniquely identified by the sensor identifier that either generated it or that it shall be sent to.

In accordance with the previous description, the SSI interface shall be implemented as a protocol stack formed by the following layers:

- **Physical Layer**, a component of the system which shall implement part of functionality 1.
- **Link Layer**, a component of the system which shall implement functionalities 1, 2, 3 and 4.

2.3 HUMSAT protocol stack

Sections below provide a more detailed description for each of the layers of the HUMSAT protocol stack. It shall be taken into account that the definition of the SSI interface involves the transfer of data from sensors to spacecraft and from spacecraft back to sensors; but no data transportation between other entities has to be considered.

2.3.1 Frequency Band

HUMSAT system is expected to be exploited in the band of 401-402 MHz - a request of a frequency for that band has already been placed. However, initial testing spacecraft and sensors will be operated in the amateur frequency band for satellites close to 437 MHz. Therefore, radio amateur identifiers shall be included in all frames whenever this band shall be used.

2.3.2 Physical layer

Common Characteristics For Both Uplink/Downlink between sensor and spacecraft :

- The HUMSAT project shall operate in the **UHF satellite** bands between **401-402 MH**z.
- Exact frequency is still TBD.
- Doppler shift shall be compensated **in the spacecraft**.
- The maximum **Doppler shift** to be compensated in the spacecraft shall be **+- 8 KHz** .
- Bitrate shall be **1200 bps**.
- Signal modulation shall be **GMSK (TBC)**.
- Bitstream shall be **differentially encoded**.

Specific Uplink Characteristics (Sensor-To-Spacecraft) :

- The sensor shall transmit with an **EIRP of 0.5W**.
- The sensor shall transmit with a **linear polarization**.
- Due to the polarization mismatch a maximum signal loss of 3 dB is expected to occur.
- Users may use different polarizations for the sensor antennas; however, no link is guaranteed for different polarizations and, thus, in case users decide to change the polarization to be used in their sensors, they shall bear in mind that HUMSAT-compatible spacecraft may not remain compatible with those sensors.

Specific Downlink Characteristics (Spacecraft-To-Sensor) :

- The spacecraft receiver sensitivity shall be **-122 dBm** .
- In case of bidirectional services HUMSAT compatible communications system on-board spacecraft shall transmit with an **EIRP** of **0.5 W**.
- The bigger the beamwidth of the antenna for which the EIRP condition is met, the bigger the footprint and, thus, the more the sensors that a spacecraft can communicate with at a time. The spacecraft shall transmit with a **circular polarization.**

- Due to the polarization mismatch a maximum signal loss of 3 dB is expected to occur.

Out Of Band Emissions :

the out of band emissions shall be attenuated by 43+10*log(P) dBc in a 4KHz reference adjacent bandwidth, hence, for a transmitted power of 1W, the attenuation is: **43+10*log(1) 40 dBc** after a 4 KHz adjacent band, and the resultant mask of Out of Band emissions is obtained as:

1. The maximum attenuation in dBsd: **A(dBsd) = A(dBc) - P(dBW) + P4KHz(dB(W/4KHz)) 43 - 3 - 18 = 22 dBsd**.
2. The spurious limit is **200%** of the total assigned bandwidth.
3. The mask follows the equation: **40log(F/50 + 1) dBsd (TBC)**. Hence, the following mask:

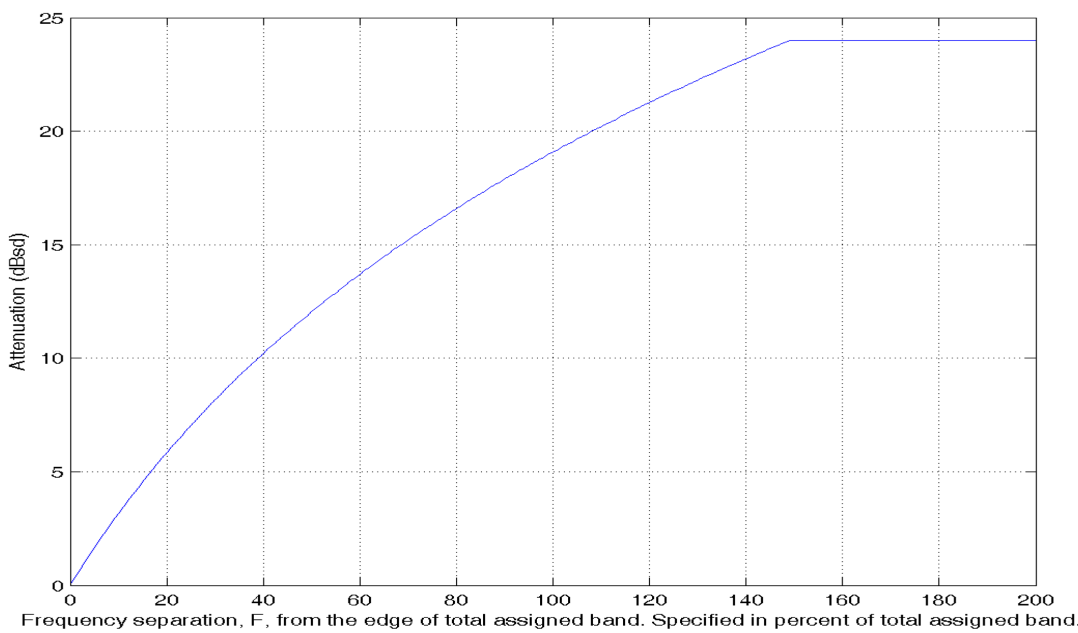


Figure 2.3 - OoB mask with a purious limit of 24 dBsd

2.3.3 Link Layer

MAC Policy And ARQ Strategy Definitions :

Since the RF channel shall be shared among spacecraft and sensors, a MAC policy shall be defined. This MAC policy shall meet the following requirements:

- All types of sensors shall send data to spacecraft.
- Spacecraft shall send data back to bi-directional sensors whenever required.

1. Bi-directional sensors shall be identified by external means and not with the data exchanged by this protocol.
2. Spacecraft shall acknowledge data received correctly to bi-directional sensors.
3. Spacecraft shall contain an updated list with the identifiers of all bi-directional deployed sensors.

- Bi-directional sensors may implement additional discovery mechanisms for allowing spacecraft to contact them even if they do not have data to send to the spacecraft.
- Sensors shall not implement any collision detection mechanism.

1. Spacecraft are not expected to implement any of these mechanisms either.

- Channel occupation time shall be such that collision probability is minimized.

ARQ strategy shall be very simple since most of sensors are not expected to have a very big computational capacity. Thus, the next ARQ strategy shall be implemented:

- Spacecraft shall be listening without transmitting.
- Sensors shall transmit the frame with the data they generated (**data frame**) periodically.
- Sensors shall repeat the transmission of the data frame after a pseudo-random time longer than 90 s has ellapsed **(T rtx):**

1. **T rtx (seconds) = 90 + random(0, 5)**
2. The addition of a random amount of time is required for avoid continuous collisions in the channel between sensors.

- After spacecraft receive the **data frame**, they shall check whether the sensor is bi-directional or not.
- For bi-directional sensors, spacecraft shall transmit, as a response to the **data frame** received, either (depending on whether the spacecraft has data for that sensor or not):
- Another **data frame** containing data stored in the spacecraft for that sensor.

1. This data frame sent by spacecraft shall be interpreted by sensors as an **implicit acknowlegdement** of the previous data frame.
2. Sensors shall reply to this data frame with an explicit acknowledgement.
   In case the acknowledgement generated by the sensor is lost, spacecraft may not confirm the sending of the data to the sensor and, thus, spacecraft shall wait for another transmission attempt.

- An **explicit acknowledgement frame** without data.
- Non-bi-directional sensors shall repeat their data frames continuously a number of times to be defined by users for each specific application.
- Since non-bi-directional sensor cannot receive data, spacecraft do not have to acknowledge their frames.

Frames Definition :

Frame size shall be short enough to minimize collisions in the RF channel, hence, a fixed length of 79 B (close to half a second at 1200 bps) is established for data transfer frames , For the implementation of the MAC and ARQ strategy defined in section above, two different types of frames have to be defined:

- **Data Frame (DF):** frame that shall be used for sending data both from sensors to spacecraft and from spacecraft to sensors.
- **Signaling Frame (SF):** frame that contains no data and that shall be used for different .

Both frames are formed up by a common header, a data field (only for the DF frames) and a CRC field at the end of the frame. Figure below shows the definition of the frames to be used by this protocol stack and how they shall be serialized for being sent to lower-level layers (from link to physical) and de-serialized for upper-level layers (from physical to link):

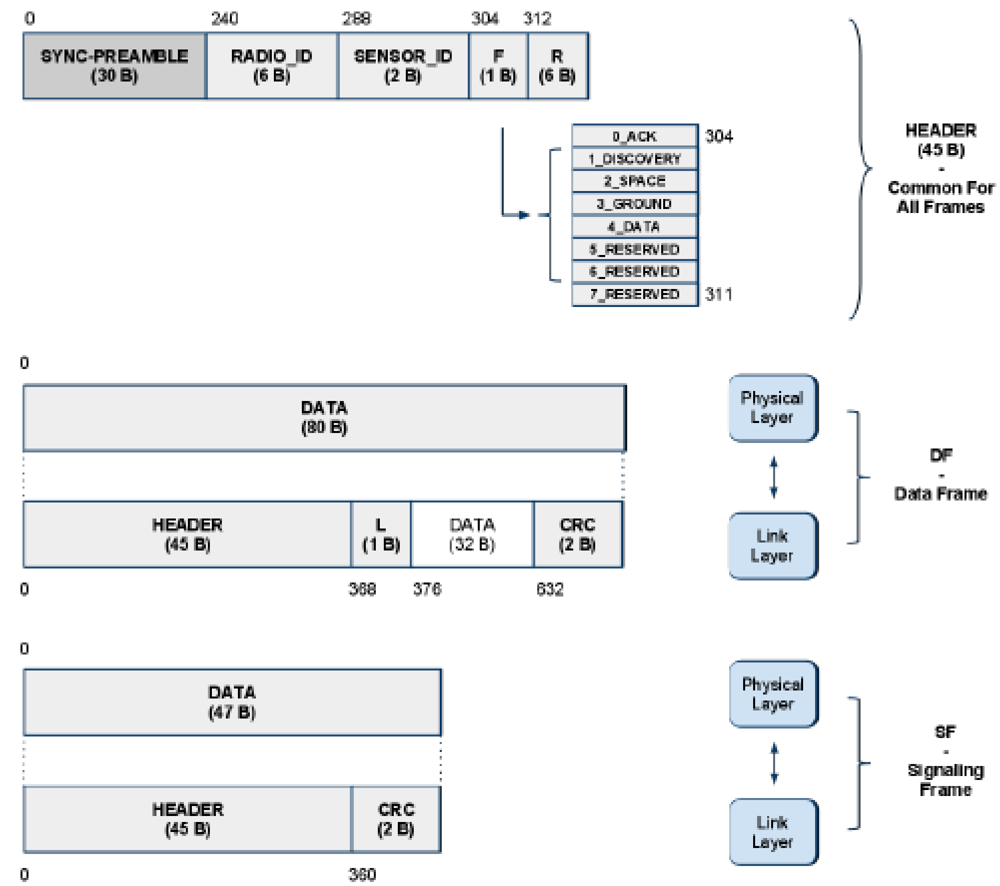


Figure 2.4 - frames definition .

Common Header Definition :

- SYNC PREAMBLE (30 B): synchronization preamble.
- RADIO ID (6 B): mandatory identifier for using UHF amateur radio bands.
  – This code shall be included in all frames whenever amatuer UHF radio bands are used, both in uplink and in downlink.
  – In case non-amateur bands are used, this field shall be filled up with zeros.

- SENSOR ID (2 B): per-device identifier assigned by the HUMSAT community.
- F (1 B): byte containing flags for signaling purposes.

  – 0 ACK: shall be set to ’1’ if this frame acknowledges a data frame.
  – 1 DISCOVERY: shall be set to ’1’ if this frame is used for permitting spacecraft to localize bi-directional sensors.
  – 2_SPACE: shall be set to ’1’ in all frames send from any spacecraft.
  – 3_GROUND: shall be set to ’1’ in all frames send from any sensor.
  – 4_DATA: shall be set to ’1’ in all frames that contain data.
  – Bits from 309 to 311 (5 RESERVED, 6 RESERVED, 7 RESERVED) are reserved for future usage and, by this time, they shall be filled up with zeros.

- R (6 B): bytes reserved for future uses.

DF Frame Definition :

- HEADER (48 B): common header for all frames.
- L (1 B): length of the data contained within this data frame.
- DATA (32 B): data to be transferred

  – Since the length of the frame is fixed, in case not all 32 B are filled up with data, transmitters shall add a proper **padding with zeros** for filling up the space left in this field.

- CRC (2 B): 16 bits for error detection within the frame.
  – It shall be calculated for all the previous fields of the frame, without including the SYNC PREAMBLE field: RADIO ID, SENSOR ID, F, R, L, DATA.

SF Frame Definition :

- HEADER (48 B): common header for all frames.
- CRC (2 B): 16 bits for error detection within the frame.
  – It shall be calculated for all the previous fields of the frame, without including the SYNC PREAMBLE field: RADIO ID, SENSOR ID, F, R.

SYNC PREAMBLE Word :

- The synchronization word is 30 B long (240 bits) and occupies the channel for about 200 ms (at 1200 bps).

- The synchronization word shall not be included in the CRC calculations.
- The synchronization word shall be composed of:
  – 27 bytes of 0xFF for receiver synchronization
  – 3 bytes that indicate the start of the frame through changes in the phase of the GSMK differentially encoded signal
  – The synchronization word remains as follows:

SYNC PREAMBLE =

0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0xFF, 0xFF, 0xFF,

0xFF, 0xFF, 0x7E, 0x7E, 0x7E

CRC Code Definition :

- The CRC code shall be CRC-16-CCITT .
  – **Polynomial: 0x1021**

Sensor Addressing :

Sensor addressing scheme shall support two different addressing modes:

- **Radio Amateur UHF Bands**: the use of these bands requires devices to be identified with a 6 B length identifier assigned. It is expected to begin the exploitation of demonstrators by using this band wh
  – Spacecraft will already have one, since they will be operated through GENSO and, thus, they will be required to be operated through radio amateur bands.
  – Sensors could be exploited in the radio amateur band although a satellite communication band will be requested to ITU for exploiting HUMSAT sensors. In this case, sensors shall also include an identifier for distinguishing each one of them among others with the same identifier assigned by local regulations for the use of the radio amateur bands.

- **Specific HUMSAT UHF band**: which will require not to include the previous radio amateur identifier. However, sensors shall still be distinguishes among them and, therefore, an identifier provided by the HUMSAT community shall be included in every frame.

The RADIO ID field of every frame is maintained even though amateur bands are not used for comunications, so that compatibility between amateur band and non-amateur band based equipment is preserved.

2.3.4 Non-Bi-Directional Sensors-Spacecraft

Diagram below describes the scenario expected to be the most common among all communications scenarios between spacecraft and non-bi-directional sensors: a single sensor transmits continuously the frames that it has generated previously until a spacecraft collects them.

Other situations (like possible collisions with frames from other sensors) are not represented in this diagram since they can be extrapolated from this diagram together with the definition of the protocol included in sections above.

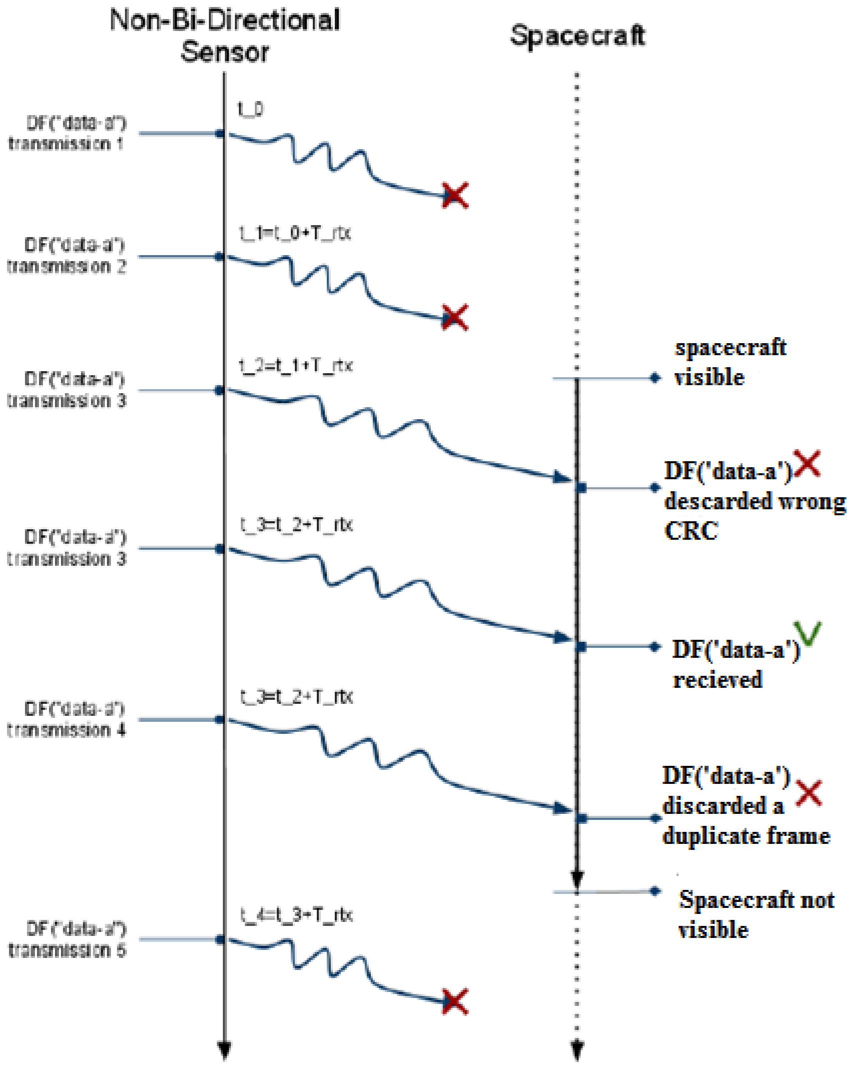


Figure 2.5 - messages exchanges between non-bi-directional sensors and spacecraft .

Protocol Functioning :

The way in which sensors and spacecraft shall behave for remaining compatible within this protocol, shall also be defined. This way, the following flowcharts for the definition of those components are presented in the figure below:

- **Link Layer for non-bi-directional sensors**: non-bidirectional sensors shall implement a very easy protocol which permits them only to upload data to spacecraft. No downlink channel is available.
- **Link Layer both for full-bi-directional sensors and for bi-directional sensors**: bi-directional sensors shall have a downlink channel for receiving data from spacecraft. This allows the implementation of a simple ack-based ARQ strategy. (not mentioned)
- **Link Layer for spacecraft**: spacecraft shall implement a protocol that allows them to support both types of sensors. (not mentioned) .

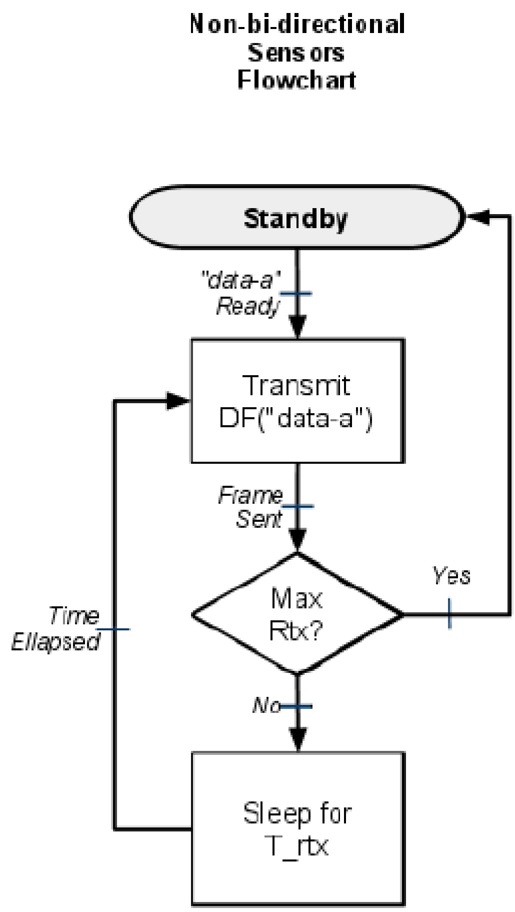


Figure 2.6 - Non-Bi-Directional sensors behaviour - HUMSAT Protocol.

CHAPTER III

Temperature Sensors

Temperature is playing a vital role in many industrial processes and therefore an accurate measurement is a must. Non-accurate temperatures can cause big fatal consequences such as reducing the lifetime of an equipment if overheated with just a few degrees.

3.1 Some basic concepts regard sensors

• **Transducer** : is a device that converts a primary form of energy into a corresponding signal with a different energy forms : mechanical, thermal, electromagnetic, optical, chemical, etc.
– take form of a **sensor** or an **actuator**
e.g Conventional Transducers :
– thermocouple: temperature difference
– compass (magnetic): direction
• **Sensor** : is a device that detects/measures a signal or stimulus acquiring information from the "real world" .
e.g Microelectronic Sensors are millimeter sized, highly sensitive :
– photodiode/phototransistor: photon energy (light)
• **Actuator** : is a device that generates a signal or stimulus Typically interested in **electronic** sensor , convert desired parameter into electrically measurable signal.
e.g infrared detectors, proximity/intrusion alarms :
– piezoresisitve pressure sensor: air/fluid pressure
– microaccelerometers: vibration, Δ-velocity (car crash)
– chemical sensors: O_2, CO_2, Cl, Nitrates (explosives)
– DNA arrays: match DNA sequences

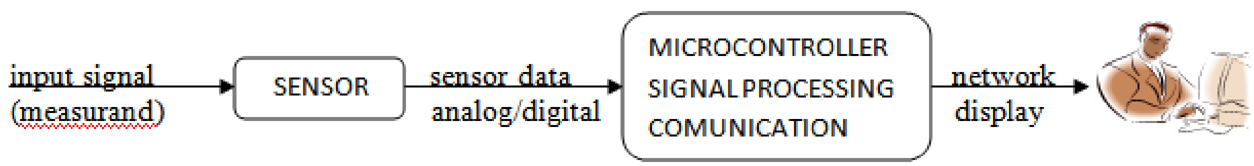


Figure 3.1 - typical electronic sensor system .

3.2 Basics of Temperature sensors

There are a wide variety of temperature sensors on the market today, including Thermocouples, Resistance Temperature Detectors (RTDs), Thermistors, Infrared, and Semiconductor Sensors. In This thesis I will discuss three of these alternatives: the RTD, thermistor, and semiconductor sensors since my project will be based on a temperature sensor .

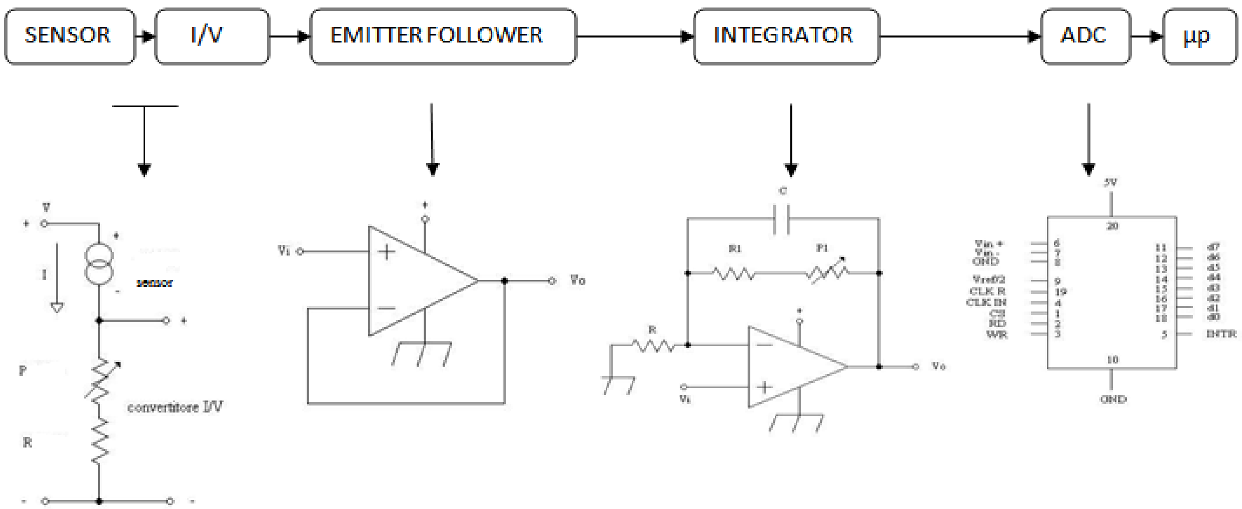


figure 3.2 - sensing chain

in this generic scheme we can notice the functionality of every component :

Sensor : the temperatur sensor trasform the temperatur in input (°K) , in current in output , the sensor has a sensibility of e.g . 1 uA/°K .

I/V : the output will be connected to an I/V convertor to convert the input current in voltage one , it is a resistance and a potentiometer in series .

Emmiter follower : the emmiter follower has the function of impedance adapter because it is composed of an OpAmp that has a low resistance in input and a high one in output .

Integrator : has two functions , Eliminate disturbances of the network frequency (Low-Pass Filter) , and amplify the output voltage because otherwise it could be very small to the ADC input .

ADC : convert the analog signal coming from the integrator to digital signal and send it to the microprocessor which would elaborate it .

3.2.1 Resistance Temperature Detector (RTD)

The RTD is a temperature sensing device whose resistance changes with temperature. Typically built from platinum, though devices made from nickel or copper are not uncommon, RTDs can take many different shapes like in figure 3.3 . To measure the resistance across an RTD, apply a constant current, measure the resulting voltage, and determine the RTD resistance. We then use a resistance vs. temperature plot to determine the temperature of the surrounding medium . RTDs exhibit fairly linear resistance to temperature curves over their operating regions, and any nonlinearities are highly predictable and repeatable.

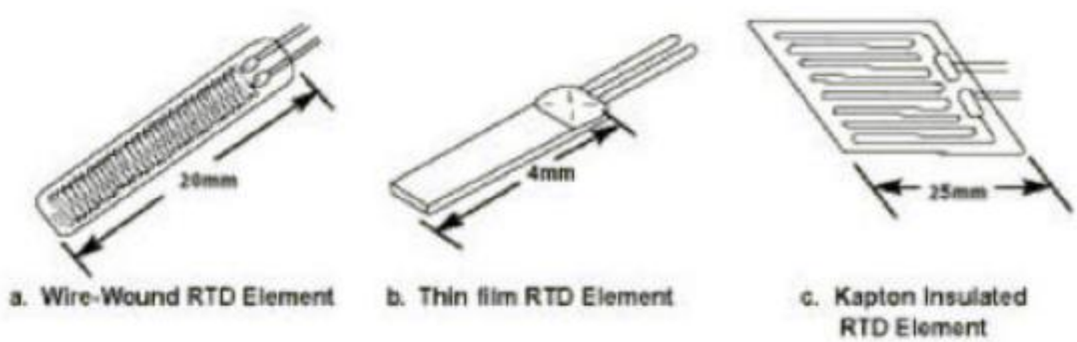


Figure 3.3 - RTD element styles .

The RTD requires external current excitation, as well as signal conditioning to account for lead wire effects and self-heating. Analog Devices supplies a device (like ADT70) , which provides both excitation and signal conditioning for a platinum RTD . The output of this device (5 mV/°C) is be fed through an analog to digital converter, to be converted by the DSP to temperature readings.

3.2.2 Thermistor

Similar to the RTD, the thermistor is a temperature sensing device whose resistance changes with temperature. Thermistors, however, are made from semiconductor materials , Resistance is determined in the same manner as the RTD, but thermistors exhibit a highly nonlinear resistance vs. temperature curve . Thus, in the thermistor's operating range we can see a large resistance change for a very small temperature change. This makes for a highly sensitive device, ideal for set-point applications.

Like the RTD, thermistors require external current excitation and significant signal conditioning. Chips do exist that will convert a thermistor value directly to digital data Simple circuits can be built to read in the thermistor voltage, which can be fed to an ADC, and the temperature determined digitally.

3.2.3 IC Temperature Sensors

RTDs and thermistors may be simple devices, but they are likely not suited to any mechatronics application. We need to buy the sensor, purchase a chip or create our own circuitry to apply a constant current and measure the resulting voltage, and run this output through and ADC. All these components need to be matched, (for example the ADC needs to have high enough resolution to take advantage of the 5mv/°C change from the ADT70). This can end up being quite complicated and costly.
Chips with temperature sensors built into the integrated circuit may be a better alternative. IC temperature sensors employ the principle that a bipolar junction transistor's (BJT) base-emitter voltage to collector current varies with temperature:

$$V_{BE} = \frac{kT}{q}\, ln\left(\frac{Ic}{Is}\right)$$

The Bandgap temperature sensor is an example of circuit utilizing this principle , Tuning the resistor values R1 & R2 and the number of BJTs N results in a direct relationship between V_{PTAT} and temperature:

$$Vptat = \frac{2R_1(V_{BE} - V_N\,)}{R_2} = 2\frac{R_1}{R_2}\frac{kT}{q}\ln(N)$$

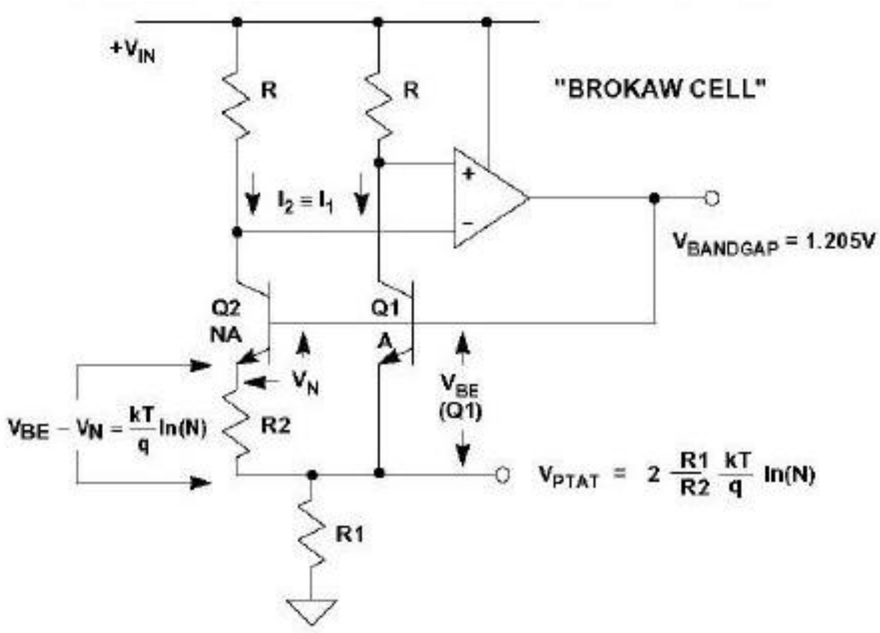


Figure 3.4 – classic bandgap temperature sensor

As the temperature sensor is built into the integrated circuit, manufacturers can do all the design for us. Signal conditioning, linearization, and analog to digital conversion can all be done on chip.

Chapter IV

Wireless data transmission

Data transmission or data-communications, the branch of telecommunications concerned with the transmission of information represented, on the basis of predetermined rules, in a formalized form by symbols or analog signals; the information either is intended for machine processing (for example, by computers) or has already undergone machine processing. The term "data transmission" is also applied to the actual process of transmitting the information. Such information is called data.

The principal difference between data transmission and telegraph, telephone, and other types of communication is that the information, or data, is sent or received by a machine rather than a human being; in data transmission from computer to computer there is no human being on either end of the communication line. Data transmission frequently requires greater reliability, rate, and accuracy of transmission because of the greater importance of the information being transmitted and the impossibility of logical monitoring by human beings during the transmitting and receiving processes. Together with computer technology, data transmission serves as the technical base for information and computing systems, including automatic control systems of various levels of complexity. The use of data transmission facilities speeds up the collection and dissemination of information and permits subscribers with inexpensive terminal equipment to enjoy the services of large computer centers.
Sending and receiving data via cables (e.g., telephone lines or fibre optics) or **wireless** relay systems. Because ordinary telephone circuits pass signals that fall within the frequency range of voice communication (about 300–3,500 hertz), the high frequencies associated with data transmission suffer a loss of amplitude and transmission speed. Data signals must therefore be translated into a format compatible with the signals used in telephone lines. Digital computers use a modem to transform outgoing digital electronic data; a similar system at the receiving end translates the incoming signal back to the original electronic data. Specialized data-transmission links carry signals at frequencies higher than those used by the public telephone network.

The most common wireless technologies use electromagnetic wireless telecommunications, such as radio. With radio waves distances can be short, such as a few metres for televisionremote control, or as far as thousands or even millions of kilometres for deep-space radio communications.
It encompasses various types of fixed, mobile, and portable applications, including two-way radios , cellular telephones, personal digital assistants (PDAs), and wireless networking. Other examples of applications of radio *wireless technology* include GPS units, garage door openers, wireless computer mice, keyboards and headsets , headphones, radio receivers, satellite television, broadcast television and cordless telephones.
Wireless operations permit services, such as long-range communications, that are impossible or impractical to implement with the use of wires. The term is commonly used in the

telecommunications industry to refer to telecommunications systems (e.g. radio transmitters and receivers, remote controls etc.) which use some form of energy (e.g. radio waves, acoustic energy, etc.) to transfer information without the use of wires. Information is transferred in this manner over both short and long distances.

Wireless networking (*e.g.*, the various types of unlicensed 2.4 GHz WiFi devices) is used to meet many needs. Perhaps the most common use is to connect laptop users who travel from location to location. Another common use is for mobile networks that connect via satellite. A wireless transmission method is a logical choice to network a LAN segment that must frequently change locations. The following situations justify the use of wireless technology:

- To span a distance beyond the capabilities of typical cabling,
- To provide a backup communications link in case of normal network failure,
- To link portable or temporary workstations,
- To overcome situations where normal cabling is difficult or financially impractical, or
- To remotely connect mobile users or networks.

For wireless data transmission we need three things :

1. Transmitter : is an electronic device which, with the aid of an antenna, produces radio waves. The transmitter itself generates a radio frequency alternating current, which is applied to the antenna. When excited by this alternating current, the antenna radiates radio waves.
2. Receiver : is an electronic device that receives radio waves and converts the information carried by them to a usable form. It is used with an antenna. The antenna intercepts radio waves (electromagnetic waves) and converts them to tiny alternating currents which are applied to the receiver, and the receiver extracts the desired information.
3. Wirless channel : the higher the frequency of the channel the more information it can transfer .

There are two types of wireless data transmission : short range data transmission , and long range data transmission .

4.1 Short range data transmission

Wireless short range data transmission in the unlicensed frequency bands is becoming more and more important in many industrial, home and office applications. Particularly devices for the transmission in the 434 MHz band and around 900 MHz are penetrating new application areas, the next step is the extensive use of the 2.4 MHz band by the Bluetooth standard, wireless LANs and proprietary applications.

Therefore, engineers coming from other branches of electronic engineering are faced with the need to become familiar with basics of RF and microwave technology, to design RF transmission systems and to build their own transmitters and receivers.

4.1.1 the Bluetooth

Bluetooth is a low cost, low power, radio frequency technology for short-range communications. It can be used to replace the cables connecting portable/fixed electronic devices, build ad-hoc networks or provide data/voice access points.

Frequency	2.4GHz ISM band, Frequency hopping
Modulation	Gaussian shaped BFSK
Data rate	723Kbps
Operating range	10m~100m
Size	28mm x 15mm x 2mm (Mitsumi WML-C05)
Cost	Long term: $5/endpoint ($20 currently)
Power	0.1W (Active)
Security	Good. FHSS. Link layer authentication and encryption
Acceptance	SIG have about 2500 member companies

Figure 4.1 – Bluetooth summary

Figure 4.2 is typical hardware architecture of one Bluetooth module. Although the original goal is single chip implementation, due to difficulties of integrating RF part into CMOS chip, many vendors now use one baseband chip and one RF chip. Cambridge Silicon Radio is working on one chip solution, and already has some exciting products available. Its BlueCore has almost all the functions in the chip, and only needs about 10 discrete components to construct a module. This makes people believe that the $5 goal may be not very far away.

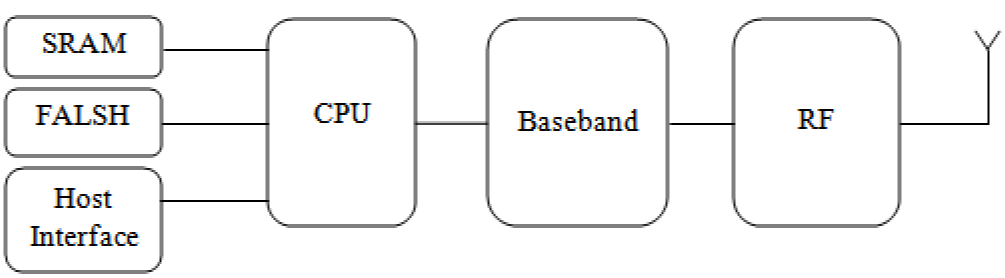


Figure 4.2 – Hardware architecture of Bluetooth .

4.1.2 the WiFi

WiFi was created specifically to operate as a wireless Ethernet , it an open-standard technology that enables wireless connectivity between equipments and local area networks . public access WiFi services are designed to deliver line services over short distances , typically 50 – 150 meters . in this cases , WiFis are connected to a local database , and give the end user access through a kiosk , or portable device .

Internet access through public WiFis is a new and very hot trend , providing many benefits and conveniences over other types of mobile internet access .

First, performance is 50-200 times faster than dial-up internet connections or cellular data access . second , users don’t have to worry about cords , wires or sharing an access point , such as a phone jack .

A global directory that would provide users with a search engine to locate the closest access point . even without the directory , wireless devices make it very easy to connect . most wireless enabled devices have a Firmware utility that indicate a user proximity to a wireless access point .

Service provider place an antenna , or an access point , at designated hot spot . the antenna transmits a wireless signal to adapter card in a user’s computer or device . users connect to the wireless through a page in their internet browser .

Coverage extends over a 50 – 150 meter radius of the access point . connection speeds range from 1.6 Mbps , which is comparable to fixed DSL transmission speed , to 11 Mbps . new standards promise to increase speeds to 54 Mbps .

Today’s WiFis run in the unlicensed 2.4 GHz to 5 GHz radio spectrums . the 2.4 GHz frequency is already crowded , it has been allocated for several purposes besides wireless services . the 5 GHz spectrum is much larger bandwidth , providing high speeds , greater reliability and better throughput .

4.2 long range data transmission

4.2.1 WiMAX

Worldwide Interoperability for Microwave Access (WiMAX) is currently one of the hottest technologies in wireless. The Institute of Electrical and Electronics Engineers (IEEE) 802 committee, which sets networking standards such as Ethernet (802.3) and WiFi (802.11), has published a set of standards that define WiMAX. IEEE 802.16-2004 (also known as Revision

D) was published in 2004 for fixed applications; 802.16 Revision E (which adds mobility) is publicated in July 2005. The WiMAX Forum is an industry body formed to promote the IEEE 802.16 standard and perform interoperability testing. The WiMAX Forum has adopted certain profiles based on the 802.16 standards for interoperability testing and "WiMAX certification".

These operate in the 2.5GHz, 3.5GHz and 5.8GHz frequency bands, which typically are licensed by various government authorities. WiMAX, is based on an RF technology called Orthogonal Frequency Division Multiplexing (OFDM), which is a very effective means of transferring data when carriers of width of 5MHz or greater can be used. Below 5MHz carrier width, current CDMA based 3G systems are comparable to OFDM in terms of performance.

WiMAX is a standard-based wireless technology that provides high throughput broadband connections over long distance. WiMAX can be used for a number of applications, including "last mile" broadband connections, hotspots and high-speed connectivity for business customers. It provides wireless metropolitan area network (MAN) connectivity at speeds up to 70 Mbps and the WiMAX base station on the average can cover 30 miles (50 Km) .

4.2.2 Cell phone 3G & 4G services

3G is currently the world's best connection method when it comes to mobile phones, and especially for mobile Internet. 3G stands for 3rd generation as it just that in terms of the evolutionary path of the mobile phone industry. 4G means 4th generation. This is a set of standard that is being developed as a future successor of 3G in the very near future.

The biggest difference between the two is in the existence of compliant technologies. There are a bunch of technologies that fall under 3G, including WCDMA, EV-DO, and HSPA among others. Although a lot of mobile phone companies are quick to dub their technologies as 4G, such as LTE, WiMax, and UMB, none of these are actually compliant to the specifications set forth by the 4G standard. These technologies are often referred to as Pre-4G or 3.9G.

4G speeds are meant to exceed that of 3G. Current 3G speeds are topped out at 14Mbps downlink and 5.8Mbps uplink. To be able to qualify as a 4G technology, speeds of up to 100Mbps must be reached for a moving user and 1Gbps for a stationary user. So far, these speeds are only reachable with wired LANs.

Another key change in 4G is the abandonment of circuit switching. 3G technologies use a hybrid of circuit switching and packet switching. Circuit switching is a very old technology that has been used in telephone systems for a very long time. The downside to this technology is that it ties up the resource for as long as the connection is kept up. Packet switching is a technology that is very prevalent in computer networks but has since appeared in mobile phones as well. With packet switching, resources are only used when there is information to be sent across. The efficiency of packet switching allows the mobile phone company to squeeze more conversations into the same bandwidth. 4G technologies would no longer utilize circuit switching even for voice calls and video calls. All information that is passed around would be packet switched to enhance efficiency.

4.2.3 Ultra-wideband (UWB) communication :

Ultra-wideband (UWB) transmission is a widely used technology in radar and remote sensing application sand has recently received great attention in both academia and industry for applications in wireless communications . A UWB system is defined as any radio system that has a10-dB bandwidth larger than 25 percent of its center frequency, or has a 10-dB bandwidth equal to or larger than 1.5 GHz if the center frequency is greater than 6 GHz. UWB usually refers to impulse based waveforms that can be used with different modulation schemes. The transmitted signal consists of a train of very narrow pulses at baseband, normally on the order of a nanosecond. Each transmitted pulse is referred to as a monocycle. The information can be carried by the position or amplitude of the pulses. In general, narrower pulses in the time domain correspond to electromagnetic radiation of wider spectrum in the frequency domain. Thus, the baseband train of nanosecond impulses can have a frequency spectrum spanning from zero to several GHz, resulting in the so called UWB transmission.

4.2.4 Satellite

The satellite itself is also known as the space segment, and is composed of three separate units, namely the fuel system, the satellite and telemetry controls, and the transponder. The transponder includes the receiving antenna to pick-up signals from the ground station, a broad band receiver, an input multiplexer, and a frequency converter which is used to reroute the received signals through a high powered amplifier for downlink. The primary role of a satellite is to reflect electronic signals. In the case of a telecom satellite, the primary task is to receive signals from a ground station and send them down to another ground station located a considerable distance away from the first. This relay action can be two-way, as in the case of a long distance phone call. Another use of the satellite is when, as is the case with television broadcasts, the ground station's uplink is then downlinked over a wide region, so that it may be received by many different customers possessing compatible equipment. Still another use for satellites is observation, wherein the satellite is equipped with cameras or various sensors, and it merely downlinks any information it picks up from its vantagepoint. **The Ground Station** is the earth segment. The ground station's job is two-fold. In the case of an uplink, or transmitting station, terrestrial data in the form of baseband signals, is passed through a baseband processor, an up converter, a high powered amplifier, and through a parabolic dish antenna up to an orbiting satellite. In the case of a downlink, or receiving station, works in the reverse fashion as the uplink, ultimately converting signals received through the parabolic antenna to base band signal , **receive** (download) at a speed of about 1 Mbps and **send** (upload) at a speed of about 200 kbps

4.3 Modulation formats

The academic field of "Data Transmission" is loaded with modulation schemes. Most involve translation of data bits or patterns into a unique combination of phase, frequency or amplitude. Some of the more notable techniques are listed in the table .

MODULATION TECHNIQUE	COMMON ACRONYM
Frequency Shift Keying	***FSK***
Multi-level Frequency Shift Keying	***MFSK***
Continuous Phase Frequency Shift Keying	***CPFSK***
Minimum Shift Keying	***MSK***
Gaussian Minimum Shift Keying	***GMSK***
Tamed Frequency Modulation	***TFM***
Phase Shift Keying	***PSK***
Quadrature Phase Shift Keying	***QPSK***
Differential Quadrature Phase Shift Keying	***DQPSK***
$\pi/4$ Differential Quadrature Phase Shift Keying	***$\pi/4$ DQPSK***
Quadrature Amplitude Modulation	***QAM***

Table 4.1

Each of the modulation formats listed in Table 4.1 is suited to a specific application. In general, schemes where two or more bits are represented by a symbol (e.g. QAM, QPSK) require better signal to noise ratios (SNR) than two-level (binary) schemes for similar bit error rate (BER) performance. Additionally, in a wireless system, schemes that have more than two levels (m-ary) generally require greater power amplifier linearity. Most implementations of the modulation formats listed in Table 4.1 are synchronous. When data rates exceed 1200 bits/second or when the transmission medium is subject to non-ideal affects (e.g. fading or SNR < 25 dB) synchronous data transmission is preferred over asynchronous. Synchronous data transmission is characterized by the presence of a clock which is synchronous to the data. The clock is embedded in, and therefore recoverable from, the modulated signal. MSK is a synchronous modulation format.

Another important consideration in data transmission is bandwidth. Digitally modulated data, composed of sharp "one to zero" and "zero to one" transitions, results in a spectrum rich in harmonic content that is not well suited to RF transmission. Hence, digital modulation formats that minimize bandwidth (BW) consumption are in vogue. As implied earlier, digital modulation involves the mapping of changes in data states to changes in amplitude, frequency, phase, or some combination of the three. After smoothing the transitions (discontinuities) in phase, frequency or amplitude, we can see, through Fourier analysis, BW consumption is reduced. An entire family of modulation formats, categorized as continuous phase modulation (CPM) minimize BW consumption by eliminating phase discontinuities. CPM state changes are represented by non-abrupt changes in phase and frequency while the amplitude of the carrier envelope remains constant (i.e. phase modulation or frequency modulation).

4.3.1 MSK Basics

MSK is a continuous phase modulation scheme where the modulated carrier contains no phase discontinuities and frequency changes occur at the carrier zero crossings. MSK is unique due to the relationship between the frequency of a logical zero and one: the difference between the frequency of a logical zero and a logical one is always equal to half the data rate. In other words, the modulation index is 0.5 for MSK, and is defined as

$m = \Delta f \times T$

where,

$\Delta f = |f_{logic\ 1} - f_{logic\ 0}|$

$T = 1/\text{bit rate}$

For example, a 1200 bit per second baseband MSK data signal could be composed of 1200 Hz and 1800 Hz frequencies for a logical one and zero respectively (see Figure).

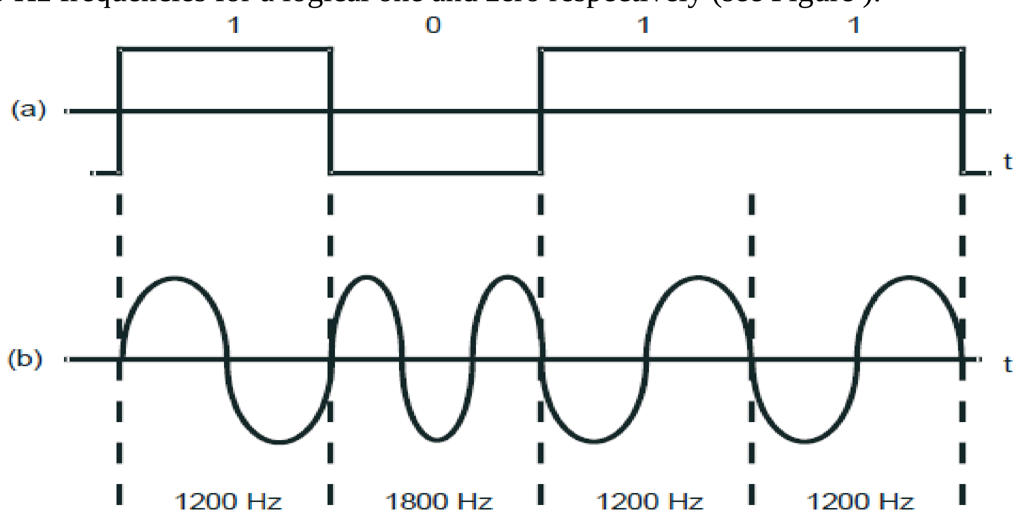


Figure 4.3 - 1200 baud MSK data signal , a) NRZ data , b) MSK signal .

Baseband MSK, as shown in Figure 4.3, is a robust means of transmitting data in wireless systems where the data rate is relatively low compared to the channel BW. MX-COM devices

such as the MX429 and MX469 are single chip solutions for baseband MSK systems, incorporating modulation and demodulation circuitry on a single chip.
An alternative method for generating MSK modulation can be realized by directly injecting NRZ data into a frequency modulator with its modulation index set for 0.5 (see Figure 4.4). This approach is essentially equivalent to baseband MSK. However, in the direct approach the VCO is part of the RF/IF section, whereas in baseband MSK the voltage to frequency conversion takes place at baseband.

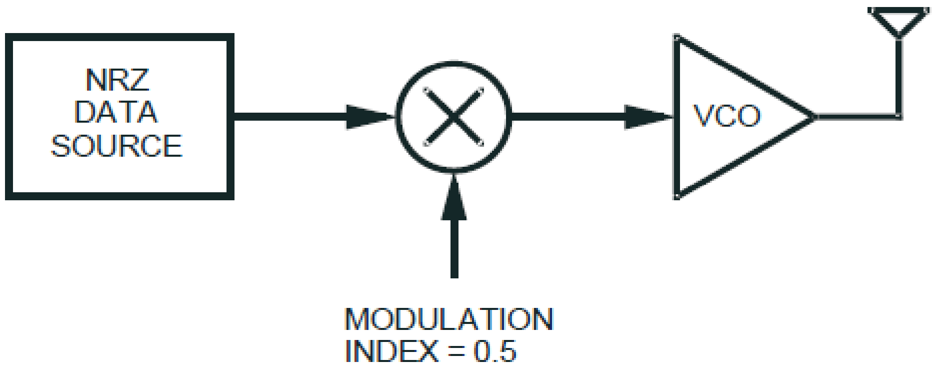


Figure 4.4 - direct MSK modulation .

The fundamental problem with MSK is that the spectrum is not compact enough to realize data rates approaching the RF channel BW. A plot of the spectrum for MSK reveals sidelobes extending well above the data rate . For wireless data transmission systems which require more efficient use of the RF channel BW, it is necessary to reduce the energy of the MSK upper sidelobes.

4.3.2 GMSK Modulation

we stated that a straightforward means of reducing this energy is lowpass filtering the data stream prior to presenting it to the modulator (pre-modulation filtering). The pre-modulation lowpass filter must have a narrow BW with a sharp cutoff frequency and very little overshoot in its impulse response. This is where the Gaussian filter characteristic comes in. It has an impulse response characterized by a classical Gaussian distribution (bell shaped curve), as shown in Figure 4.5. Notice the absence of overshoot or ringing.

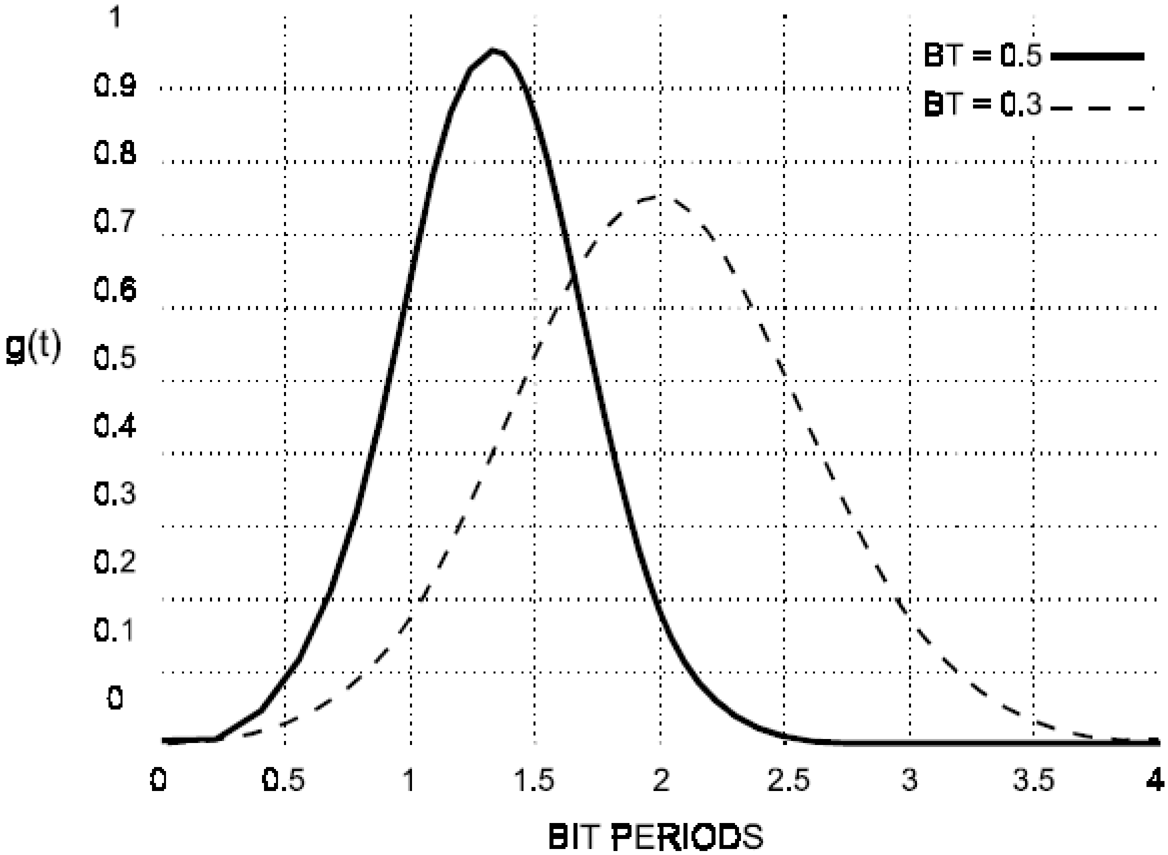


Figure 4.5 - Gausssian filter impluse response for BT = 0.3 and BT = 0.5

Figure 4.5 depicts the impulse response of a Gaussian filter for BT = 0.3 and 0.5. BT is related to the filter's -3dB BW and data rate by

$$BT = \frac{f_{-3dB}}{bit\ rate}$$

Hence, for a data rate of 9.6 kbps and a BT of 0.3, the filter's -3dB cutoff frequency is 2880Hz .
Still referring to Figure 4.5, notice that a bit is spread over approximately 3 bit periods for BT=0.3 and two bit periods for BT=0.5. This gives rise to a phenomena called inter-symbol interference (ISI). For BT=0.3 adjacent symbols or bits will interfere with each other more than for BT=0.5.
GMSK with BT= ∞ is equivalent to MSK. In other words, MSK does not intentionally introduce ISI. Greater ISI allows the spectrum to be more compact, making demodulation more difficult. Hence, spectral compactness is the primary trade-off in going from MSK to Gaussian pre-modulation filtered MSK. Figure 4.6 displays the normalized spectral densities for MSK and GMSK. Notice the reduced sidelobe energy for GMSK. Utlimately, this means channel spacing can be tighter for GMSK when compared to MSK for the same adjacent channel interference.

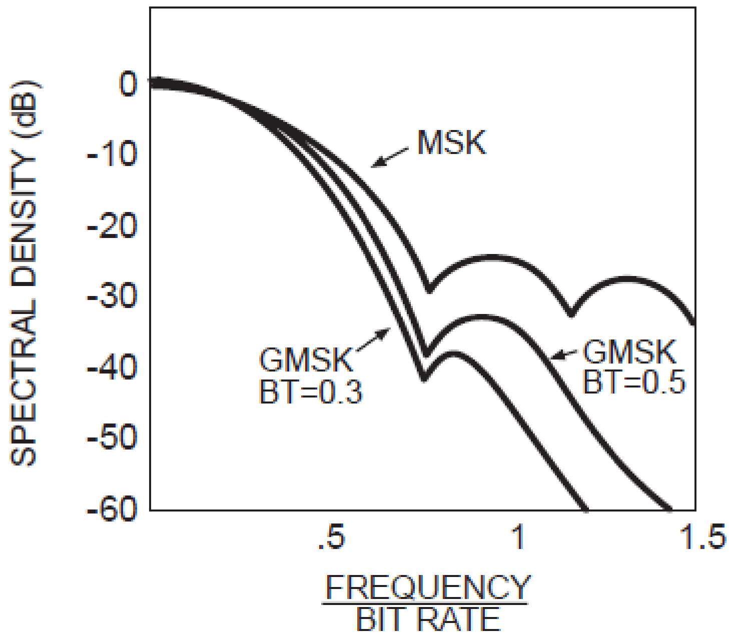


Figure 4.6 - Spectral density for MSK and GMSK .

The performance of a GMSK modem is generally quantified by measurement of the signal-to-noise ratio (SNR) versus BER. SNR is related to Eb/N0 by

$$\frac{E_b}{N_0} = \frac{S}{RN_0} = \frac{S}{N}\left(\frac{B_n}{R}\right)$$

Where ,
S = signal power .
R= data rate in bits per second
N_0 = noise power in spectral density (watts /Hz)
E_b = energy per bit
E_b*N_0 = N = noise power
B_n = noise BW of IF filter

4.3.3 I and Q modulation

Quadrature (I and Q) modulation can also be effective in eliminating synthesizer shortcomings. In I and Q modulation, the Gaussian filtered data signal is separated into in-phase (I) and quadrature phase (Q) components. The modulated RF signal is created by mixing the I and Q components up to the frequency of the RF carrier, where they are summed together. The role of the synthesizer has now been reduced to merely changing carrier frequency for channel selection. The key to optimum performance with quadrature modulation is accurate creation of the I and Q components.

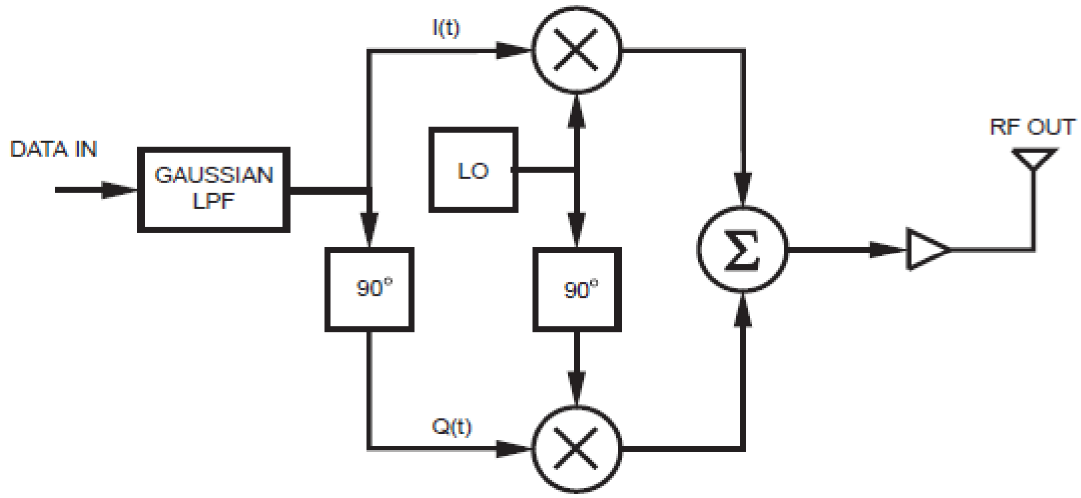


Figure 4.7 - I and Q radio block diagram .

Baseband I and Q signals can be created by using an all-pass phase shifting network. This network must maintain a 90 degree phase relationship between the I and Q signals for all frequencies in the band of interest.

4.3.4 Demodulation GMSK

Demodulation of the GMSK signal requires as much attention to the preservation of an unadulterated wave form as does modulation of the signal. The choice of a Gaussian shaped pre-modulation filter was made for three main reasons:
1) narrow bandwidth and sharp cutoff
2) lower overshoot impulse response
3) preservation of the filter output pulse area.
The first condition gives GMSK modulation its spectral efficiency. It also improves its noise immunity when demodulating. The second condition affords GMSK low phase distortion. This is a major concern when the receiver is demodulating the signal down to baseband, and care must be taken in the design of the IF filtering to protect this characteristic. The third condition ensures the coherence of the signal. While this is quite strict and not realizable with a physical Gaussian filter, the phase response can be kept linear and therefore sufficient for coherent demodulation.

CHAPTER V

Detailed description of the developed circuit and Firmware

In order to develop the remote sensor and the SSI interface requested by HUMSAT project , I used a special program called MENTOR GRAFICS DESIGN , by using MENTOR it is possible to design a circuit using components from available library , and to print the circuit using expedition PCB , some components were not available in the library so I found their data sheets by searching it in the website of electronic seller like Digi-Key and RS-components , then I designed these components with respect to their data sheets to insert them in the library [21] .
For the Firmware development I used IAR embedded workbench IDE writing codes for the microcontroller and the transceiver using C language codes , here are a short description of the circuit and Firmware developed .

5.1 temperature sensor (NTC)

My project doesn't request a special temperature sensor , any temperature sensor could be good for this mission if doesn't consume a lot of power .

The NTC Thermistor is a glass protected temperature sensor, which has a temperature span from -40°C to 150°C , it has very little power dissipation about 125 mW @ 25°C . Its' resistance value range is 2.2kΩ to 100kΩ highly accurate with 1% of tolerance . The sensor has a simple 3 wire interface: ground, power, and output.[22]

3 V dedicated for power supply the sensor , that has an analog output connected to the microprocessor with the pin P6.0A0 which is connected to the Bus A (A_D6_A0), this is connected to an internal ADC , so in the Firmware some codes have been written to convert the analog signal to digital one .

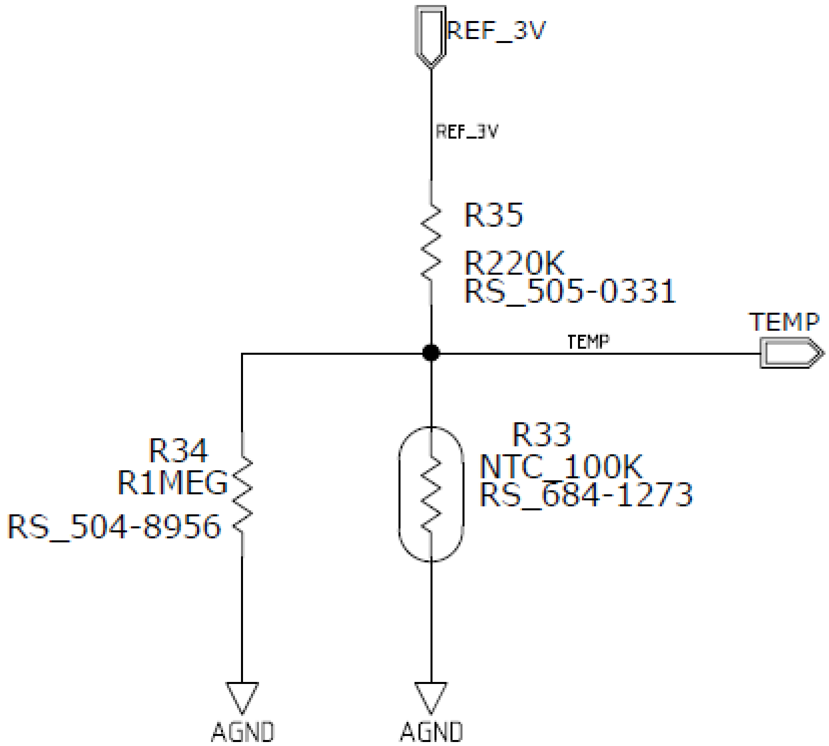


Figure 5.1 – NTC thermistor temperature sensor .

5.2 Si4464 Transceiver

as I mentioned before , HUMSAT project request some specifications to be respected for data transmission, these specification could be realized with Si4464 transceiver of lab silicon laboratory ,features :

- Frequency range 119-1050MHz so my desiderated frequency band is covered 401-402 MHz
- Si4464 supports different modulation options among them GMSK that I need .
- Si4464 has a Max Output power about +20dBm , but this is not sufficient , the transmission power needed (+27dBm) this problem could be solved inserting in the transmission path a power amplifier that could operate at the same frequency and has about +27 dBm of output power .

one of our aims in IC design is to reduce the power consumption of the circuit , Si4464 has low active power consumption of about 70 mA at +20dBm , and power supplied with 1.8 to 3.6 V .

The Si4464 communicates with the (MSP430F5438) over a standard 4-wire serial peripheral interface (SPI): SCLK, SDI, SDO, and nSEL. The SPI interface is designed to operate at a maximum of 10 MHz.

Transceiver PIN	Microcontroller PIN	BUS
SCLK	UCA0CLK	A_D4_CLK
nSEL	UCA0STE	B_D4_CLK
SDI	UCASIMO	A_D1_TX_SIMO
SDO	UCASOMI	A_D0_RX_SOMI

Table 5.1 - transceiver connections with the microcontroller

The host microcontroller writes data over the SDI pin and can read data from the device on the SDO output pin , The nSEL pin should go low to initiate the SPI command. The first byte of SDI data will be one of the firmware commands followed by n bytes of parameter data which will be variable depending on the specific command. The rising edges of SCLK should be aligned with the center of the SDI data.

The Si4464 contains an internal MCU which controls all the internal functions of the radio. For SPI read commands a typical MCU flow of checking clear-to-send (CTS) is used to make sure the internal MCU has executed the command and prepared the data to be output over the SDO pin , Once the CTS value reads FFh then the read data is ready to be clocked out to the host MCU. The typical time for a valid FFh CTS reading is 20 µs.

The internal MCU will clock out the SDO data on the negative edge so the host MCU should process the SDO data on the rising edge of SCLK.

An application programming interface (API), which the host MCU will communicate with, is embedded inside the device. The API is divided into two sections, commands and properties. The commands are used to control the chip and retrieve its status. The properties are general configurations which will change infrequently.

The complete command and property descriptions are provided in “AN625: Si446x API Descriptions”.

Four general purpose IO pins are available to utilize in the application. The GPIO are configured by the GPIO_PIN_CFG command in address 13h , we utilize two GPIO pins to switch the RF switch .

5.3 (HMC453ST89) power amplifier PA

The output power of the Si4464 transceiver is not enough , we need +27dBm to transmit the data came from the transceiver to the satellite , for this reason I inserted an external power amplifier

HMC453ST89 in the transmission line of the transceiver , the HMC453ST89 could be adapted to operate in various frequency range among them 400 to 410 MHz to give 32dBm of output power.

a negative aspect of the HMC453ST89 is to be a consumer component its' current rating is about 725 mA supplied with 5 V , that should be considered to think about a technique to reduce the power consumption , by deactivate the component when it is not necessary to be active , the following figure show the circuit of the PA in 400 to 410 MHz .

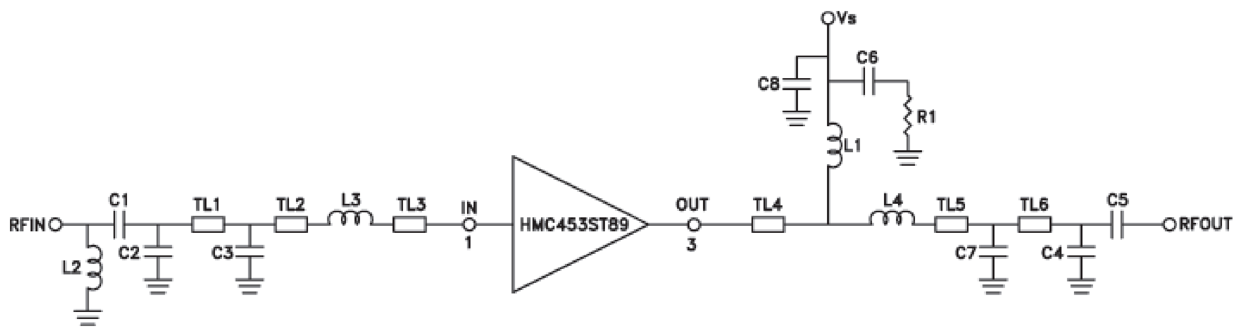


Figure 5.2 - HMC453ST89 PA for 400 to 410 MHz .

Recommended component values	
C1,C2	10 pF
C3,C4	8.2 pF
C5	39 pF
C6	100 pF
C7	12 pF
C8	2.2 uF
L1	47 nH
L2	40 nH
L3	4.3 nH
L4	5.1 nH
R1	5.1 ohm

	TL1	*TL2*	*TL3*	*TL4*	*TL5*	*TL6*
Impedance	*50 ohm*	*50 ohm*	*50 ohm*	*50 ohm*	*50 ohm*	*50 ohm*
Physical Length	*0.16''*	*0.04''*	*0.06''*	*0.21''*	*0.04''*	*0.10''*
Electrical Length	*4°*	*1°*	*1°*	*5°*	*1°*	*2°*
Physical Width	*0.64 mm*	*0.64 mm*	*0.64 mm*	*0.64 mm*	*0.64 mm*	*0.64 mm*
Thichness	*18 um*	*18 um*	*18 um*	*18 um*	*18 um*	*18 um*
Height	*0.36 mm*	*0.36 mm*	*0.36 mm*	*0.36 mm*	*0.36 mm*	*0.36 mm*
PCB material :FR-4 , Er = 4.7						

Table 5.2,5.3 - component values and transmission lines dimensions for 400 to 410 MHz of HMC453ST89.

5.4 load Switch

As to deactivate the HMC453ST89 we inserted a load switch between the dedicated power supply 5 V and the power amplifier controlled by the microcontroller ,connected with the pin P2.5 ,and high Enabled by the Bus signal B_D9_EN_PWM2 , while the microcontroller still reading values from the sensor this load switch will be active and will interrupt the supply power from the power amplifier , it is very simple technique to reduce power consumption of HMC453ST89 , in this figure you can note the schematic of the load switch , designed using MENTOR .

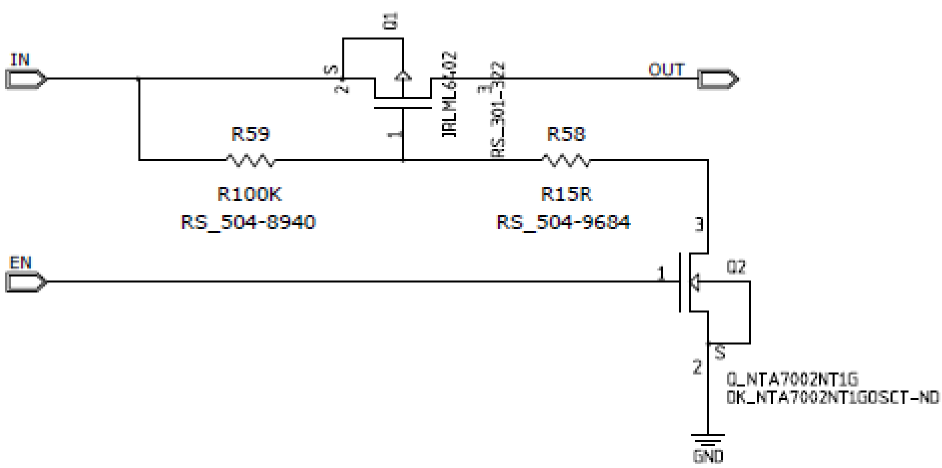


Figure 5.3 – load switch

5.5 SKY13290-313LF , RF switch

I used SKY13290-313LF , RF switch , for receive/transmit switching , although this is a high isolation switch I use two switches , one for each path to increase the isolation between the transmission and receiving paths .

The features of the SKY13290-313LF could satisfy the specifications , its’ frequency range is 20MHz to 2.5 GHz and has High input power compression: 0.1 dB @ > +40 dBm , The device is controlled with positive, negative, or a combination of both voltages, two GPIO signals coming from the transceiver control this switch to chose within transmit/receive function , since my project request to transmit I configured the transceiver to transmit always . GPIO_0 connected to

SW1 pin is high and GPIO_1 connected to SW2 pin is low , see the schematic . The RF signal paths within the device are fully bilateral and it has Low current consumption: <100 µA @ 3 V.

5.6 MSP430F5438 microcontroller

The Texas Instruments MSP430 family of ultralow-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 5 µs.

The MSP430F5438 is a microcontroller configuration with three 16-bit timers, a high performance 12-bit analog-to-digital (A/D) converter, up to four universal serial communication interfaces (USCI), hardware multiplier, DMA, real-time clock module with alarm capabilities, and up to 87 I/O pins.

Typical applications for this device include analog and digital sensor systems, digital motor control, remote controls, thermostats, digital timers, hand-held meters.

5.7 Solar power-supplied circuit

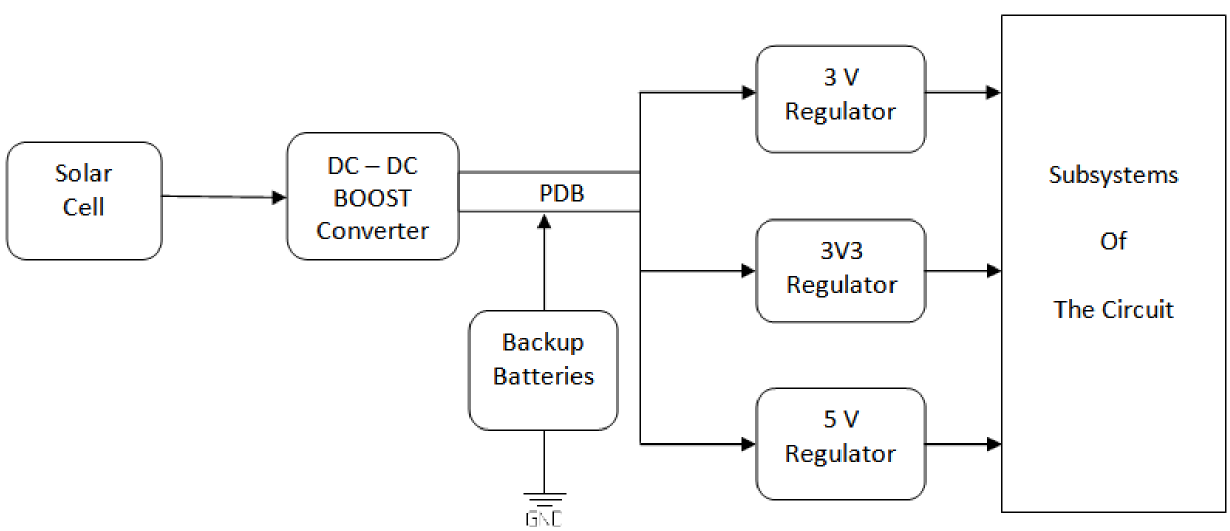


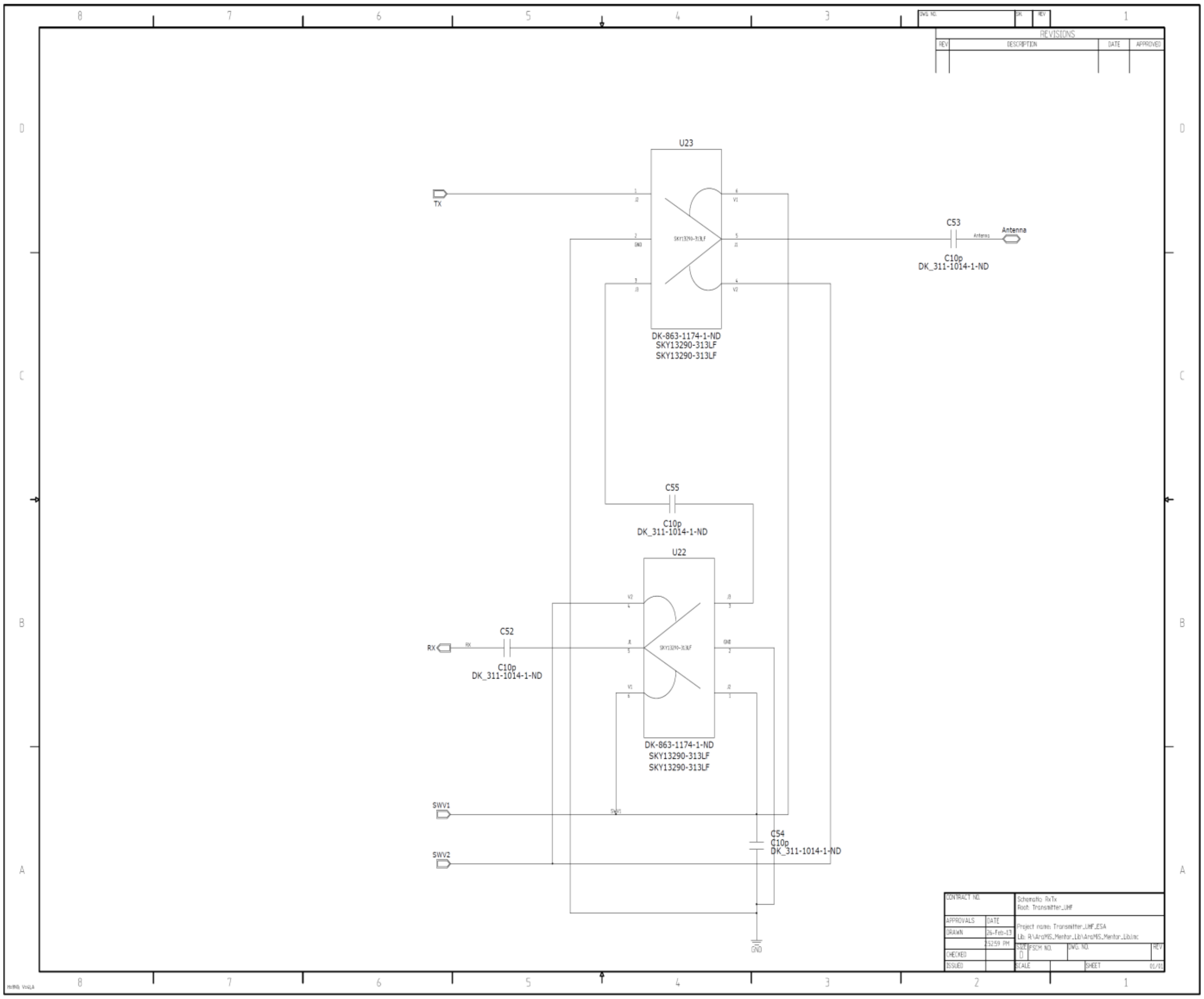
Figure 5.4 – block diagram of the solar power-supplied circuit

We use two triple Junction GaAs solar cells , each generate 2.2 V , they are connected in series and provide an output voltage of 4.4 V .

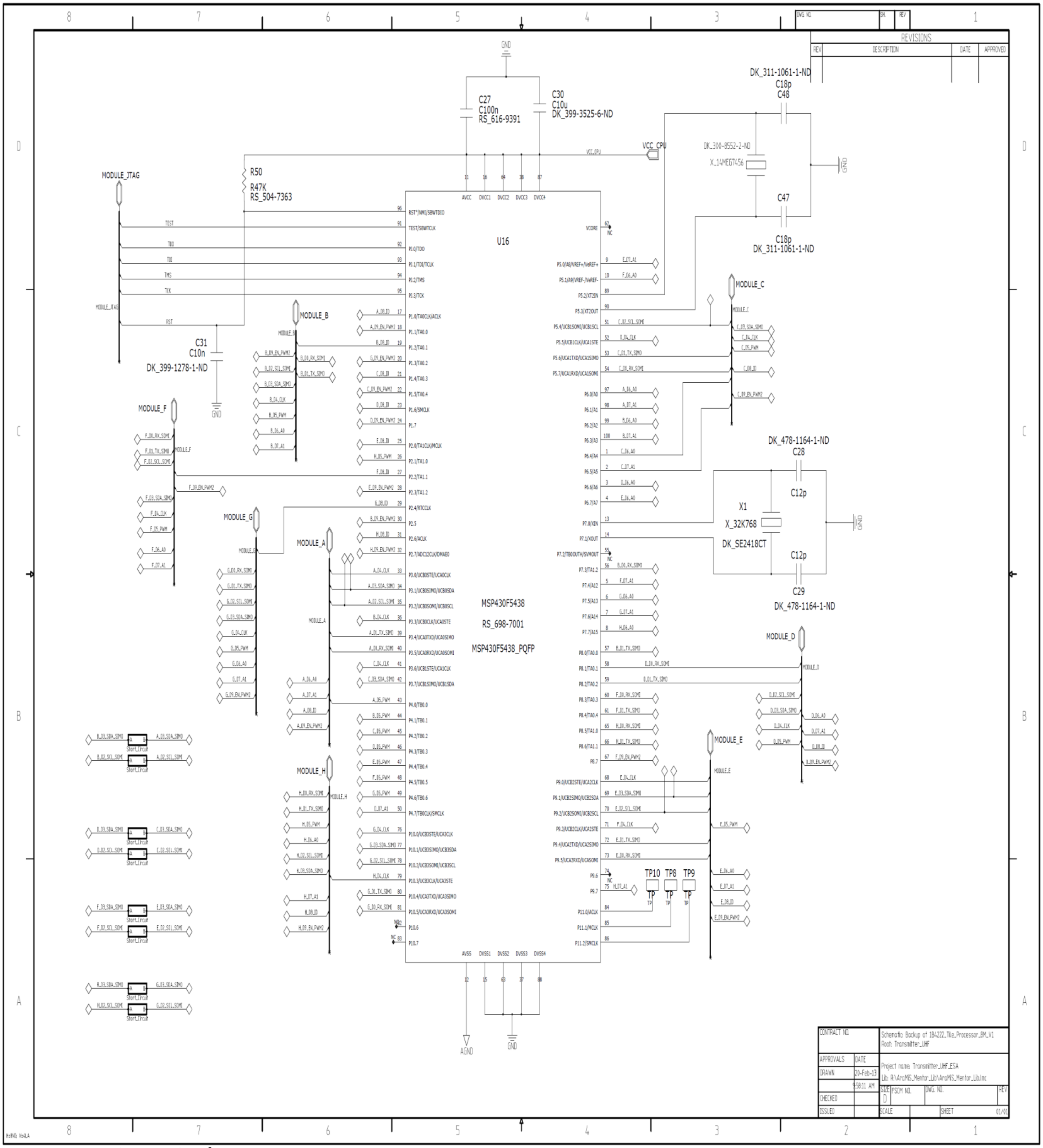
The Boost converter converts the solar cells voltage to PDB (power distribution Bus) level (14 V $\pm$ 2 V) . In systems that utilize solar cells as the source of energy it is recommended to employ some sort of storage device. A storage device can prove very useful as it can store an unused energy generated by the solar cells throughout the day and, in turn, this store energy can be used to power a system when no sunlight is available to the solar cells, thus making the system more practical.

To power supply the circuit from DC-DC converter built, three voltage regulators would need to be used to supply a constant voltage to the circuit itself , down converting the PDB voltage to different voltage levels , 5V , 3V3 , and 3V .

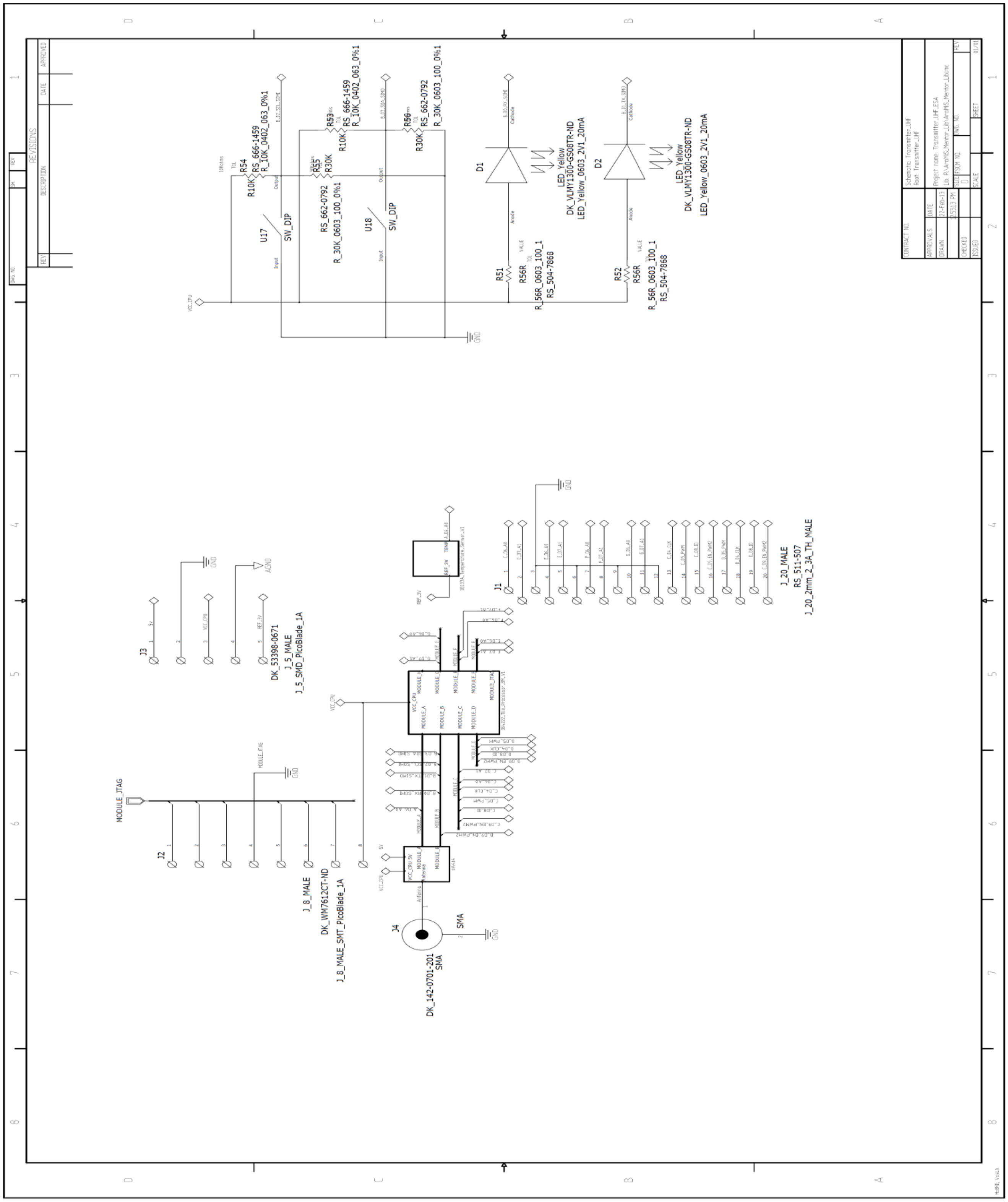
5.8 schematics & PCB Layout



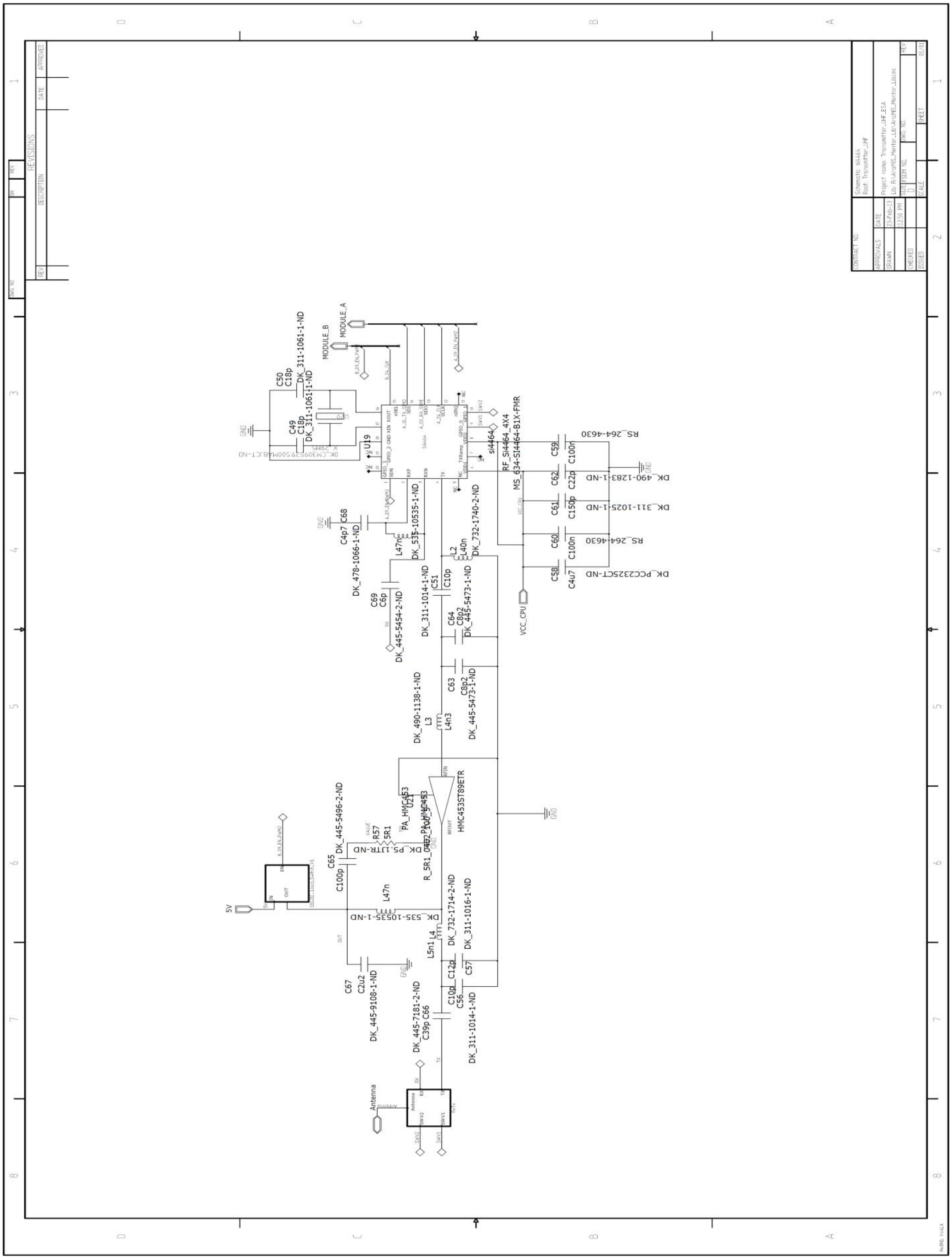
RF switches



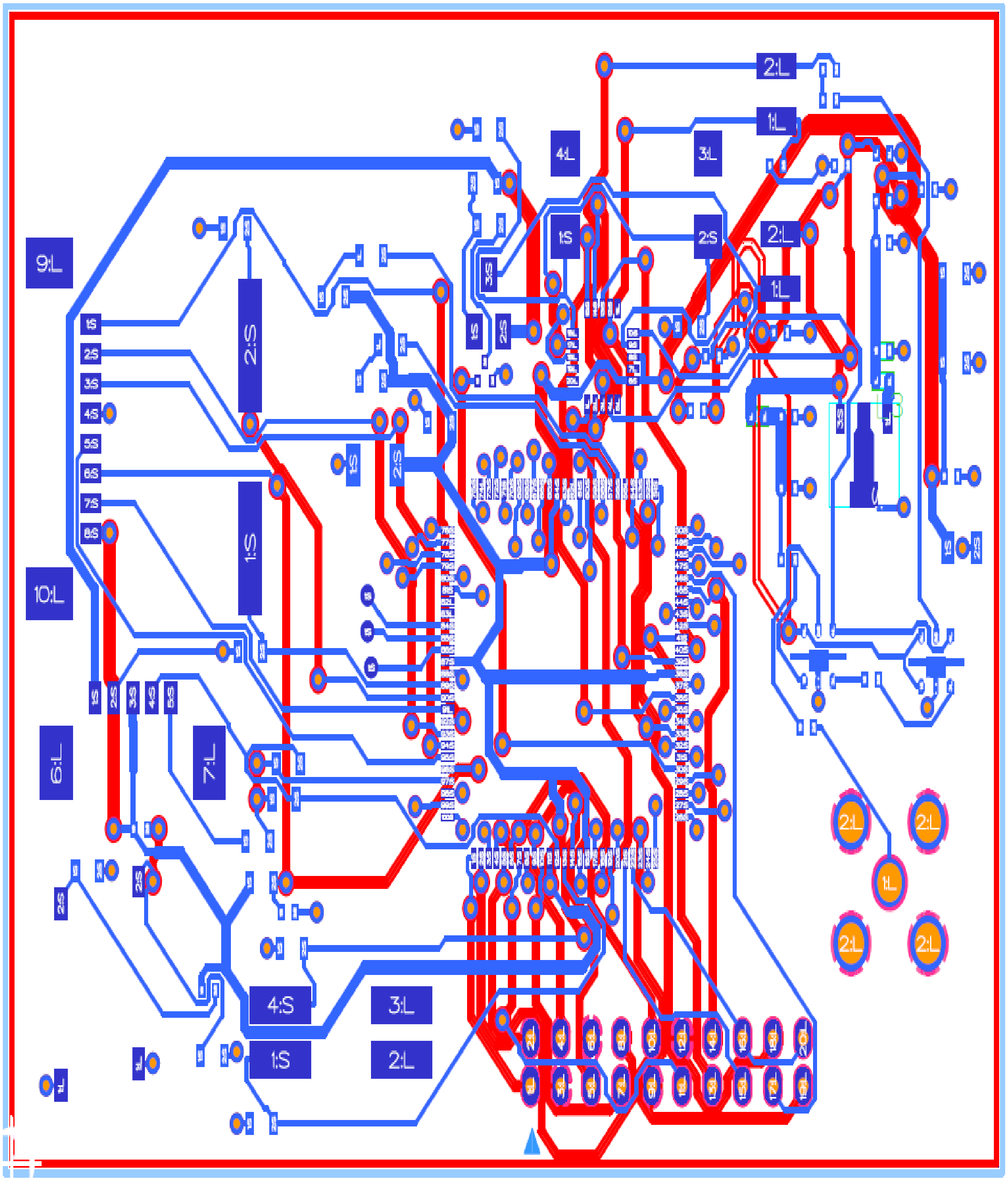
MSP4305F5438 Schematic Design using Mentor



Block diagram of the circuit using Mentor

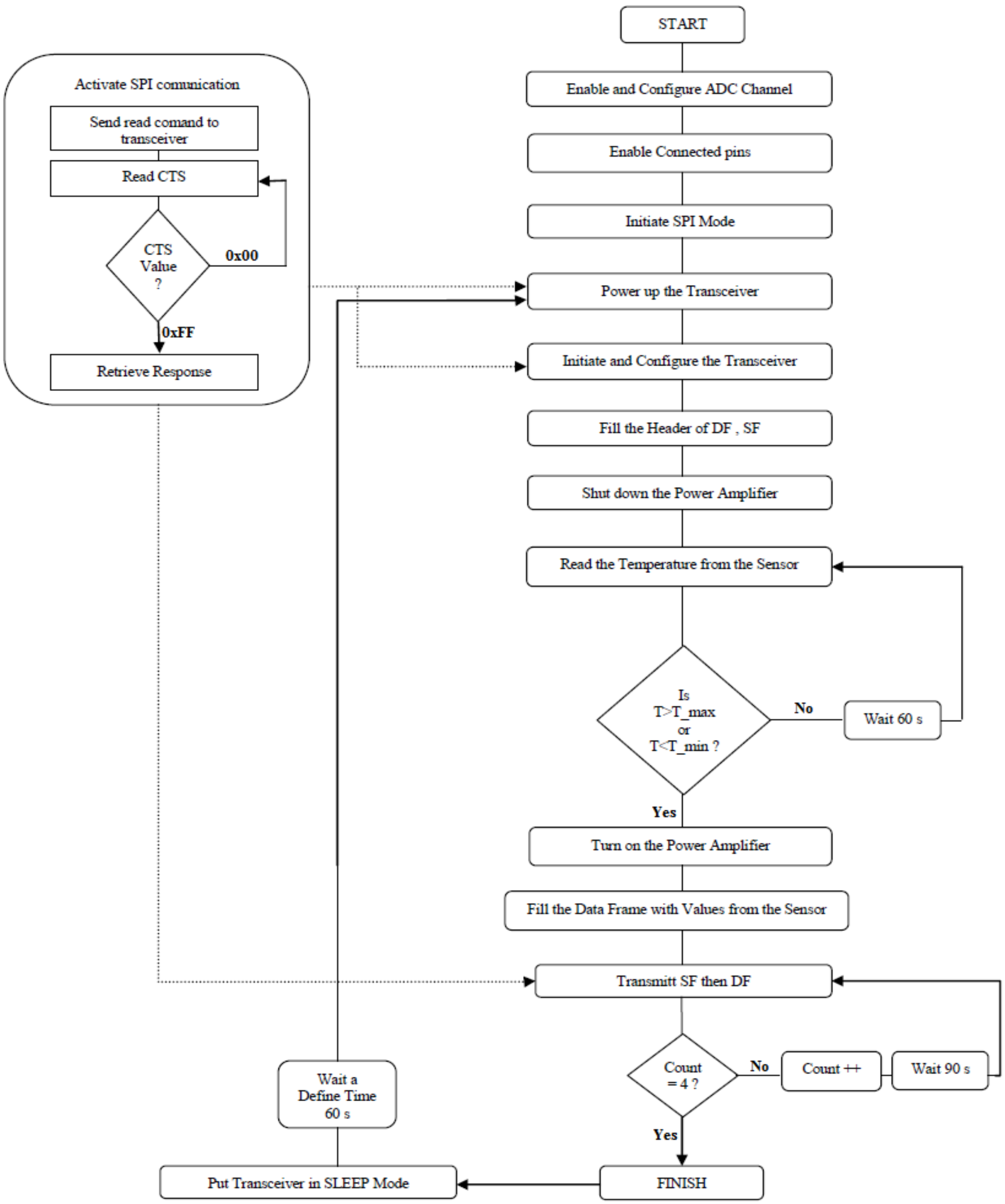


Transceiver and PA Schematic Design using Mentor



PCB Layout

5.9 Firmware FlowChart



5.10 C programming code

```
/* C Program Code for MSP430F5438 */

#include <math.h>
#include "msp430f5438.h"

/* #define section
* define the index of the content in the data packet
* and registers etc.
*/

#define PCK_SIZE                        80 /* data packet size */
#define PCK_HEADER_SIZE                 45 /* data packet header size */
#define PCK_HEADER_SYNCP_SIZE           30
#define PCK_HEADER_SYNCP_B0             0
#define PCK_HEADER_RADIOID_SIZE         6
#define PCK_HEADER_RADIOID_B0           30
#define PCK_HEADER_SENSORID_SIZE        2
#define PCK_HEADER_SENSORID_B0          36
#define PCK_HEADER_F_SIZE               1
#define PCK_HEADER_F_B0                 38
#define PCK_HEADER_R_SIZE               6
#define PCK_HEADER_R_B0                 39

#define PCK_L_SIZE                      1
#define PCK_L_B0                        45
#define PCK_DATA_SIZE                   32
#define PCK_DATA_B0                     46
#define PCK_CRC_SIZE                    2
#define PCK_CRC_B0                      78
#define PCK_CRC_B1                      79
#define PCK_PAYLOAD_SIZE                78
#define PROP_PKT_GROUP                  12

#define TRANS_Q_BUF_SIZE                64    /* Transceiver API Command+Parameters Buffer Size*/
#define TRANS_A_BUF_SIZE                16    /* Transceiver API Reply Buffer Size*/
#define TRANS_RDCMD_BUF                 0x44
#define TRANS_CMD_POWER_UP              0x02
#define TRANS_CMD_GET_INT_STATUS        0x20
#define TRANS_CMD_SET_PROPERTY          0x11
#define TRANS_CMD_CHANGE_STATE          0x34
#define TRANS_CMD_FILL_TXFIFO           0x66
#define TRANS_CMD_START_TX              0x31

#define REPEAT_PACK_COUNT       4
#define TEMP_MAX                100        //  check the polarity, gain and offset of the output value
#define TEMP_MIN                10         //                of the ADC12 and temperature sensor
#define EXPT_TMP_DATA_COUNT     1          // expected number of data entries from temperature sensor // number of data
to put into package

#define CRC16_SEED                      0xFFFF


#define CLK_FREQ_MHZ            400 // this is the transmission frequency
#define CLK_PERIOD_NS           2.5
#define MAX_CTS_RETRY               2500
#define CC_330US                    330000/CLK_PERIOD_NS    //  Tune with actual clock freq (clock cycles for 330 us)
#define CC_5MS                      70422      //  Tune with actual clock freq (... for 5 ms)

char packet[PCK_SIZE];
unsigned char adc12_data_index;             // 0~32

char trans_q[TRANS_Q_BUF_SIZE];       // buffer to store the command and parameters you want to send to transceiver
char trans_a[TRANS_A_BUF_SIZE];       // buffer for holding the answer from the transceiver
signed char trans_q_length;
```

```
signed char trans_a_length;      // length of expected answer from the transceiver
signed char trans_r_length;      // real length of the answer

int askTrans(void){
  int i;

  P3OUT &= ~BIT3;                                      // nSEL active low
  for (i = 0 ; i < trans_q_length; i ++){
    while (!(UCA0IFG&UCTXIFG)) ;    // SPI TX Ready for Next Char?
    UCA0TXBUF = trans_q[i];
  }

  i = 0;
  trans_r_length = 0;

  while (!(UCA0IFG&UCTXIFG)) ;       // SPI TX Ready for Next Char? (Finished sending command?)
                                    // The interrupt flag will be cleared automatically when new char is written to the UCA0TXBUF

  P3OUT |= BIT3;                                       // Command Sent!

  while(1){
    P3OUT &= ~BIT3;                                    // Try to Read Response
    UCA0TXBUF = TRANS_RDCMD_BUF;                       // Send the ReadCmdBuffer Byte
    while (!(UCA0IFG&UCTXIFG));
    while (!(UCA0IFG&UCRXIFG));

    if (UCA0RXBUF != 0xFF){
      P3OUT |= BIT3;
      __delay_cycles(16);
      if (i ++ < MAX_CTS_RETRY)
        return 1;
    }else
      break;
  }

  while (trans_a_length != trans_r_length){
    while (!(UCA0IFG&UCRXIFG)) ;                                     // Get A Byte back?

    trans_a[trans_r_length] = UCA0RXBUF;               // Store the responses from Transceiver
    trans_r_length ++;
  }

  P3OUT |= BIT3;
  return 0;
}

/* Get the Interrupt Bytes from Transceiver (Clear Pending Interrupts)*/
void transGetIntStatus(void){
  trans_q[0] = TRANS_CMD_GET_INT_STATUS;                // API Commmand read INT Status
  trans_q[1] = 0x00;
  trans_q[2] = 0x00;
  trans_q[3] = 0x00;
  trans_q_length = 4;                                   // Command Length : 4 Bytes
  trans_a_length = 8;                                   // Requires 8 bytes reponse
  askTrans();
}

/* Power up the Transceiver (Power Up Reset)*/
void transPowerUp(void){
  P1OUT |= BIT1;                                        // SDN = 1
  __delay_cycles(CC_330US);                             // Delay for 330us
  P1OUT &= ~BIT1;                                       // SDN = 0
  __delay_cycles(CC_5MS);

  trans_q[0] = TRANS_CMD_POWER_UP;                      // API Commmand power up
  trans_q[1] = 0x01;                                    // Write global control registers
  trans_q[2] = 0x00;                                    // Write global control registers
  trans_q_length = 3;                                   // Command Length : 3 Bytes
  trans_a_length = 0;                                   // Requires no reponse
  askTrans();
```

```
  transGetIntStatus();                                   // Cleared the int Status
}

/*  Init/Config the Transceiver*/
void transInit(void){
  trans_q[0] = TRANS_CMD_SET_PROPERTY;                  // Use property command
  trans_q[1] = PROP_PKT_GROUP;                          // Select property group
  trans_q[2] = 4;                                        // Number of properties to be written
  trans_q[3] = 0E;                              //PROP_PKT_FIELD_1_LENGTH_12_8      // Specify first property
  trans_q[4] = 0x00;
  trans_q[5] = PCK_SIZE;                                 // PCK_SIZE byte long packet field
  trans_q[6] = 0x00;                                     // modulation
  trans_q[7] = 0xA2;                                     // Start CRC calc. from this field, check CRC at the end
  trans_q_length = 0x08;
  trans_a_length = 0x00;
  askTrans();
}

/* Put Transceiver in SLEEP Mode*/
void transSleep(void){
  trans_q[0] = TRANS_CMD_CHANGE_STATE;                   // Change state command
  trans_q[1] = 0x01;                                     // SLEEP state
  trans_q_length = 2;
  trans_a_length = 0;
  askTrans();
}

void transSendpacket(int bSF){
  int j = 0;
  int i = 0;

  if (bSF == 1){
    trans_q[0] = TRANS_CMD_FILL_TXFIFO;
    for (i = 0; i < PCK_HEADER_SIZE; i ++){
      trans_q[1+i] = packet[i];
    }

    trans_q[1+i] = packet[PCK_CRC_B0];
    trans_q[2+i] = packet[PCK_CRC_B1];

    trans_q_length = 48;
    trans_a_length = 0;
    askTrans();
  }
  else{
    for (i = 0; i < PCK_SIZE; i += 40){
      trans_q[0] = TRANS_CMD_FILL_TXFIFO;
      for (j = 0; j < 20; j ++){
        trans_q[1+j] = packet[i+j];
      }

      trans_q_length = 41;
      trans_a_length = 0;
      askTrans();

      while(1){
        transGetIntStatus();
        if (trans_a[2]&BIT2) break;                      // TXFIFO Almost empty?
        __delay_cycles(16);
      }
    }
  }

  trans_q[0] = TRANS_CMD_START_TX;                       // Use Tx Start command
  trans_q[1] = 0;                                        // Set channel number
  trans_q[2] = 0x10;                                     // sleep state after Tx, start Tx immediately
  trans_q[3] = 0x00;
  trans_q[4] = 80;                                       // Packet Length is 80 (payload)
  trans_q_length = 0x05;
```

```
  trans_a_length = 0;
  askTrans();

  while(1){
    transGetIntStatus();
    if (trans_a[3]&BIT5) break;                                   // packet Sent?
    __delay_cycles(16);
  }
}

void transSendPrevpacket(void){
  trans_q[0] = TRANS_CMD_START_TX;                                // Use Tx Start command
  trans_q[1] = 0;                                                 // Set channel number
  trans_q[2] = 0x14;                                              // sleep state after Tx, send previous packet again
  trans_q[3] = 0x00;
  trans_q[4] = 80;                                                // Packet Length is 80 (payload)
  trans_q_length = 0x05;
  trans_a_length = 0;
  askTrans();
}

/*  internal CRC module in MSP430F5438
CRC16 Poly : 0x1021
Endian: Normal (not Rervesed)
*/
void fill_pck_crc16(char *data, unsigned int size)
{
  int i;
  CRCINIRES = CRC16_SEED;

  for(i = PCK_HEADER_RADIOID_B0; i < size; ++i)
    CRCDI_L = data[i];

  data[size] = CRCINIRES_H;
  data[size+1] = CRCINIRES_L;

  return ;
}

void main(void){
  // declarations of the variables
  int i;
  unsigned int j;

  WDTCTL = WDTPW+WDTHOLD;                                         // Stop watchdog timer

  P6SEL |= 0x01;                                                  // Enable A/D channel A0
  /* Initialize ADC12_A for temperature sensor*/
  ADC12CTL0 = ADC12ON+ADC12SHT0_8+ADC12MSC;                       // Turn on ADC12, set sampling time,Check with the temp sensor
                                                                  //specification about the sampling time.
                                                                  // set multiple sample conversion
  ADC12CTL1 = ADC12SHP+ADC12CONSEQ_2;                             // Use sampling timer, set mode
  ADC12CTL0 |= ADC12ENC;                                          // Enable conversions
  ADC12CTL0 |= ADC12SC;                                           // Start conversion

  P1DIR |= BIT1;                                                  // set P1.1 to output (SDN)
  P2DIR |= BIT5;                                                  // set P2.5 to output

  P3DIR |= BIT3;                                                  // P3.3 as Output (slave SPI nSEL)

  /* Init eUSCI to SPI mode*/
  P3SEL |= BIT0 + BIT4 + BIT5;                                    // P3.4, P3.5, P3.0
  UCA0CTL0 = UCSWRST;                                             // eUSCI RESET
  // UCCKPL = 0b
  UCA0CTL1 =  UCSYNC +                                            // SPI mode
    UCCKPH +
      UCMSB +                                                     // MSB First (This is not indicated in the Transceiver DS)
        UCMST                                                     // Master Mode
          ;                                                       // 3-Pin SPI
  UCA0CTL0 |=  UCSSEL_2;                                          // SMCLK
```

```
  UCA0BRW = 50;                                    // Bit rate scaler = 50, SPI Bit Clock = (SMCLK/50) (8MHz < 10MHz)
  UCA0CTL0 &= ~UCSWRST;                 // Clear the SW Reset

  UCA0IE = UCTXIE + UCRXIE;             // Enable both SPI RX and TX interrupts

  transPowerUp();
  transInit();

  // fill the header section
  for (i = 0; i < PCK_HEADER_SYNCP_SIZE; i ++){ // SYNO_P, ALL ZERO
    packet[PCK_HEADER_SYNCP_B0+i] = 0xFF;
  }

  packet[PCK_HEADER_SYNCP_B0+PCK_HEADER_SYNCP_SIZE-3] = 0x7E;
  packet[PCK_HEADER_SYNCP_B0+PCK_HEADER_SYNCP_SIZE-2] = 0x7E;
  packet[PCK_HEADER_SYNCP_B0+PCK_HEADER_SYNCP_SIZE-1] = 0x7E;

  for (i = 0 ; i < PCK_HEADER_SENSORID_SIZE; i ++){ // SENSOR ID, ALL ZERO
    packet[PCK_HEADER_SENSORID_B0+i] = 0x00;
  }

  packet[PCK_HEADER_F_B0] =0x14; // 0_ACK(0), 1_DISCOREY(0), 2_SPACE(0), 3_GROUND(1), 4_DATA(1), 5,6,7_RESERVED(0)

  for (i = 0; i < PCK_HEADER_R_SIZE; i++){ // R, RESEVERED, ALL ZERO
    packet[PCK_HEADER_R_B0+i] = 0x00;
  }

  packet[PCK_L_B0] = 32; // length of the data section

  __bis_SR_register(GIE);                                          // Enable interrupts

  while (1){          // every minut, read temperatute and verify if it is in the range , initialize, turn on, send packet, else do nothing; go in shutdown and wait one minut(in low-power mode)
    adc12_data_index = 0;
    ADC12IE = BIT0;                                                // Enable ADC12IFG.0

    P2OUT &= ~BIT5;                                                // Set P2.5 to 0 to shutdown the amplifier

    __bis_SR_register(LPM4_bits);                      // Enable LPM4
    while(adc12_data_index != EXPT_TMP_DATA_COUNT) ;  // wait till the packet is filled with data from temp sensor

        if (packet[PCK_DATA_B0+0] < TEMP_MAX && packet[PCK_DATA_B0+0] > TEMP_MIN){ // check if the temperature from sensor is in the range
                __delay_cycles(10000);                             // tune with hardware to wait for 60s
                continue;                                        // within normal range then skip following operationss
        } // else continue to send the package

    /* calculate the two CRC bytes */
    fill_pck_crc16(packet, PCK_PAYLOAD_SIZE);

    // trasmit the packet over the tranceiver
    P2OUT |= BIT5;                                                 // Amplifier ON!
    __delay_cycles(1000);                                  //  wait if neccessary to ensure the amplifier stablize

    transSendpacket(1);                                // Load and send Signaling Frame
    __delay_cycles(1000);

    transSendpacket(0);                                // Load the payload packet to Transceiver and send

    for (i = 0; i < REPEAT_PACK_COUNT; i ++){          // transmit same packet for another 4 times
                                                 // if T(amplifier_stable) <<< 19s, shutdown amplifier here also
      __delay_cycles(1000);
      transSendPrevpacket();
    }
  }
}
```

```
#pragma vector=ADC12_VECTOR
__interrupt void ADC12ISR (void)
{
 switch(__even_in_range(ADC12IV,34))
 {
 case  0: break;                    // Vector  0:  No interrupt
 case  2: break;                    // Vector  2:  ADC overflow
 case  4: break;                    // Vector  4:  ADC timing overflow
 case  6:                           // Vector  6:  ADC12IFG0
   if (adc12_data_index != EXPT_TMP_DATA_COUNT){
     packet[PCK_DATA_B0+adc12_data_index] = ((signed char) ADC12MEM0) >> 4;    // Move results
              adc12_data_index++;
   } else {
     ADC12IE = 0x00;                // Disable the interrupt
   }
 case  8: break;                    // Vector  8:  ADC12IFG1
 case 10: break;                    // Vector 10:  ADC12IFG2
 case 12: break;                    // Vector 12:  ADC12IFG3
 case 14: break;                    // Vector 14:  ADC12IFG4
 case 16: break;                    // Vector 16:  ADC12IFG5
 case 18: break;                    // Vector 18:  ADC12IFG6
 case 20: break;                    // Vector 20:  ADC12IFG7
 case 22: break;                    // Vector 22:  ADC12IFG8
 case 24: break;                    // Vector 24:  ADC12IFG9
 case 26: break;                    // Vector 26:  ADC12IFG10
 case 28: break;                    // Vector 28:  ADC12IFG11
 case 30: break;                    // Vector 30:  ADC12IFG12
 case 32: break;                    // Vector 32:  ADC12IFG13
 case 34: break;                    // Vector 34:  ADC12IFG14
 default: break;
 }
}
```

Conclusion

It is very interesting to take part of HUMSAT project in order to be able to Free access to the data through internet , especially that HUMSAT has an Active support and participation by Space agencies and highly recognized International Organizations , that offer Worldwide coverage.

I believe that this project could be a useful idea for temperature monitoring in different environments and areas without infrastructure like the desert mountains and the sea surface with very low cost , it could be used to monitor temperature in industrial environment like a nuclear power plant where the temperature changes frequently .

A future purposes of this project is to realize the receiving part of the communication to be able to exchange data with the spacecraft itself that could be the temperature of the space .

Another idea is to involve this project in military purposes to give the possibility to monitor certain areas from far stations , especially that this project is compatible with other types of sensors .

Greetings

ho desiderio di ringraziare con affetto i miei genitori per il sostegno ed il grande aiuto che mi hanno dato , Desidero anche ringraziare il Professor REYNERI ed I suoi assistenti per le numerose ore dedicate alla mia tesi , ringrazio anche la mia linea di comando nella scuola di applicazione dell'esercito per la loro collaborazione , in fine ringrazio con affetto i miei amici Ingegneri compagni di corso Anisa Giovanni Elio Alessio Vito Andrea pietro , in particolare gli Elettronici Claudio ed Andrea, per il loro grande sostegno.

References

[1] Humsat general description document , universidade de vigo , by Fernando aguado .

[2] **National Semiconductor**
National Temperature Sensor Handbook
http://luna.et-inf.fho-emden.de/datenblaetter/sensor/temphb.pdf

[3] **Watlow Electric Manufacturing Company**
Temperature Sensors, Educational Series Book 4, 1995
http://www.watlow.com/literature/prodtechinfo/files/sensors/we4e_16.pdf

[4] **Analog Devices, Walt Kester**
Temperature Sensors
http://www.analog.com/UploadedFiles/Associated_Docs/245380809Power_sect6.pdf

[5] **National Instruments**
Temperature Sensor Vendors and Tutorials
http://www.ni.com/devzone/advisors/temperature.htm

[6] Murota, K. and Hirade, K., "GMSK Modulation for Digital Mobile Radio Telephony," IEEE Transactions on Communications, vol COM-29, No. 7. pp. 1044-1050, July 1981.

[7] Pebbles, P. Jr., "Digital Communication Systems," Prentice-Hall, Englewood Cliffs, NJ, 1987

[8] Sklar, B., "Digital Communications: Fundametals and Applications," Prentice-Hall, Englewood Cliffs, NJ,1988.

[9] "Cellular Digital Packet Data System Specification, Release 1.0," 1993.

[10] "Radio/Modem Reference Design Guide, Issue 1, Revision 0," RAM Mobile Data, Woodbridge, NJ, 1991.

[11] Practical GMSK Data Transmission, Fred Kostedt, Engineer for MX-COM. James C Kemerling, Engineer for MX-COM.

[12] Temperature Sensors , Drew Gilliam .

[13] telecom regularity authority, technical affairs and technology sector,WiFi technology , july 2013.

[14] http://www.differencebetween.net/technology/difference-between-3g-and 4g/#ixzz1EOXLlyom

[15] Characterization of Ultra Wideband Communication Channels, by Ali Hussein Muqaibel

[16] Satellite Communications , David Hart, dhart@cis.ohio-state.edu .

[17] WiMax Overview Sanida Omerovic, Faculty of Electrical Engineering, University of Ljubljana, Slovenia.

[18] data transmission , Encyclopedia

[19] Bluetooth , Encyclopedia

[20] wireless data transmission , Wikipedia.

[21] DocuMentor , maria caldaralo , Maurizio tranchero .

[22] http://docs-europe.electrocomponents.com/webdocs/0d9a/0900766b80d9a0b4.pdf