UP ELECTRICAL
AND ELECTRONICS
ENGINEERING
INSTITUTE

EEE 192 Electrical and Electronics Engineering Laboratory VI
2nd Semester AY 2020-2021

CanSat Project Documentation

Date	02-July-2021
Group	MCDE1-4
Members	Belen, Alarcon Ace Fresnoza, Philippe Raphael Vicencio, Rico Antonio

Summary

Commonly used as an introduction to space science and technology applications, a Can Satellite (CanSat) is a miniature satellite primarily deployed to measure atmospheric parameters.

In this project, CanSat operation was modeled to last 150 minutes: 120 minutes for sensor measurement during ascent up to descent, and 30 minutes for retrieval. Due to the limitations of Infineon Designer, sensors were modelled as linear piecewise voltage generators and data points are already provided. Sensor output was interfaced to the analog-to-digital converter (ADC) using the appropriate analog signal processing (ASP) blocks determined by the output swing. Measured data was also stored in the microcontroller's built-in ROM and transmitted via a Binary Phase Shift Keying (BPSK) modulation scheme. To power the CanSat subsystems, a single-ended primary-inductor (SEPIC) converter and a Ćuk converter was used to provide a dual-rail voltage supply. Power analysis on the subsystems proved that a single 3.7 V battery with a capacity of 1500mAh can provide sustained voltage levels during the CanSat operation. The memory and radio frequency (RF) link requirements were also analyzed such that the design is within operating margins.

Table of Contents

I. Introduction

A satellite is an object that moves on a path around a relatively larger object. Satellites can be either natural or man-made. For example, the Moon is a natural satellite since it moves around Earth [1]. On the other hand, man-made satellites are commonly built instrumentation devices designed to orbit around Earth, as well as other heavenly bodies.

The uses of man-made satellites vary from weather forecasting, space exploration, communications, imaging, among many others. These play an important role in the advancement of human life and research.

Satellites may vary in size depending on its purpose. An increasing trend of smaller satellites built is observed, mainly due to greater flexibility in terms of development, launching, and post-launching operations [2].

A Can Satellite (CanSat) is an emulation of a real satellite, which is built in a size soda can [3]. With its scaled-down design and operations, schools and universities utilize CanSats as an introduction to space science and technology applications to their students. In this context, the main challenge of a CanSat is to fit all necessary parts of a satellite and perform the missions it is designed to accomplish.

A typical CanSat can be divided into four subsystems, namely: the power, communications, sensors, and bus subsystems. The power subsystem is concerned with the power requirements of the satellite, accounting the power consumed by the other subsystems to operate properly. Next, the communications subsystem deals with the interaction of the satellite with the ground station by transmitting data and receiving commands, if needed. Then, the sensors subsystem reads the data gathered by the sensors integrated to the satellite and converts them to digital information. Finally, the bus subsystem supports the components of the whole CanSat system. All subsystems must be designed and implemented correctly and properly to ensure a successful CanSat mission [4] - [6].

CanSat missions are commonly classified into two primary and secondary missions. For most CanSat projects, atmospheric instrumentation and measurement of air parameters are set as the primary mission while the secondary mission is usually chosen by the teams themselves. Some examples of secondary missions are GPS location, testing of radiation levels, and camera integration for 3D imaging [7].

In this project, the team must include pressure, temperature, and humidity as the primary parameters need to be measured by the CanSat, whereas GPS and carbon dioxide data are measured to determine land areas fit for vegetation.

II. Project Objectives, Scope and Limitations

A. Project Objectives

The aim of this project is to design a CanSat that will measure atmospheric parameters for weather monitoring. The specific objectives are listed as follows:

- To maintain sufficient power supply during a 150-minute operation period
- To establish reliable transmission of atmospheric data to the ground station with a maximum distance of 50 km from the CanSat
- To generate sensor output (temperature, relative humidity, and pressure) based on a given profile and encapsulate sensor data into packets
- To provide a secondary mission and discuss its implications on CanSat operations
- To perform simulations on the CanSat subsystems interface using Infineon Designer
- To assess the developed CanSat system given the design requirements

B. Scope and Limitations

This project involved designing a CanSat primarily to collect atmospheric data. The payload consists of a power module, a communications board that transmits data to the ground station, and sensors to measure temperature, humidity, and pressure.

This project was limited to simulations using Infineon Designer. In particular, the XMC1100-T038 microcontroller was used to run the three modules. Infineon DAVE was also used to create C project files that will drive the microcontroller. CanSat development, up to performance analysis of the system was purely done using software. Consequently, there will be no flight testing involved in this project.

Simulations will not account for risks involving physical contact with the CanSat. That is, while unpredictable occurrences such as weather conditions may affect the CanSat mission, such events will be disregarded on the tests.

Lastly, discussion on the secondary mission will be limited to a qualitative approach. No simulation will be performed on the modified system, but its hypothetical effects on resource allocation will be examined.

III. CanSat Operations

A. System Block Diagram

In this project, the Can Satellite was used as an instrument for atmospheric data observation. Sensor modules that record pressure, humidity, and temperature are integrated into the system and transmitted through the communications module. The overall system is powered by a voltage source that is controlled by the power module. Furthermore, the CanSat modules will be interfaced by implementing an Inter-Integrated Circuit (I^2C) Protocol. Figure 3.2 illustrates the system block diagram representing the Can Satellite modules:

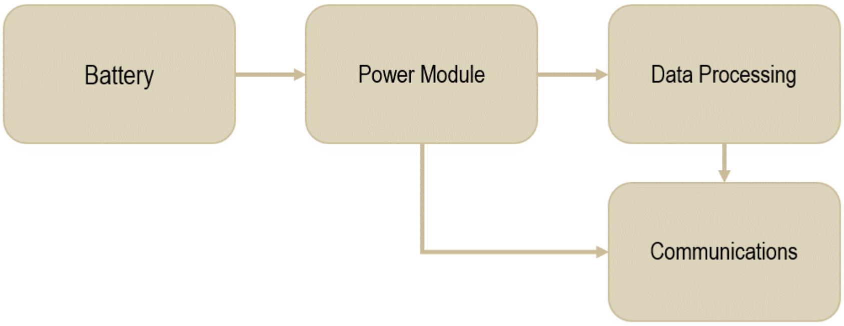


Figure 3.1. Can Satellite System Block Diagram

B. Operational Stages

With respect to the standard implementation of the CanSat mission, the operations of the CanSat can be classified into three main stages: the Launch stage, the Descent stage, and the Landing stage:

Launch Stage

During the Launch stage, the CanSat is deployed to a specified elevation by various means such as Unmanned Aerial Vehicles (UAVs) or weather balloons. Over the ascent, sensor data is measured and a one-way communications link to a ground station is established to transmit the measured data. Once this elevation level is reached, the CanSat is released into freefall. Subsequently, the parachute mechanism is activated and the operation transitions to the next stage.

Descent Stage

In the Descent stage, the CanSat continues to transmit sensor data to the ground station throughout its flight time. Moreover, the payload can be accomplished in this period. All sensor data must be stored into the memory module for recovery during the Landing stage.

Landing Stage

After the flight time of the CanSat, it is retrieved by the team during the Landing stage. Here, the stored data is retrieved, and system checks are made for further analysis.

C. Modified Operational Stages

However, since physical implementation of the mission was restricted due to an ongoing pandemic, actual operations were focused on simulating CanSat operations. The following modifications were made to the operational stages:

Launch Stage

No physical launching will be done. Instead, data points during the ascent are provided. For system modeling, the CanSat is expected to achieve a desired elevation of 50km over a 60-minute period. Sensor data is also encapsulated into packets and transmitted at a rate of one packet per minute, for a total of 60 packets during the launch stage.

Descent Stage

Similarly, packet transmissions will continue during the descent stage. Data points are also provided for this stage. As the simulated flight time is set to 60 minutes, an additional 60 packets will be received by the ground station for further analysis.

Landing Stage

The retrieval time for the CanSat is modelled to a 30-minute idle period. This will be accounted for in the power budget of the CanSat.

D. Flowchart

In summary, Figure 3.2 illustrates the operational stages of the CanSat.

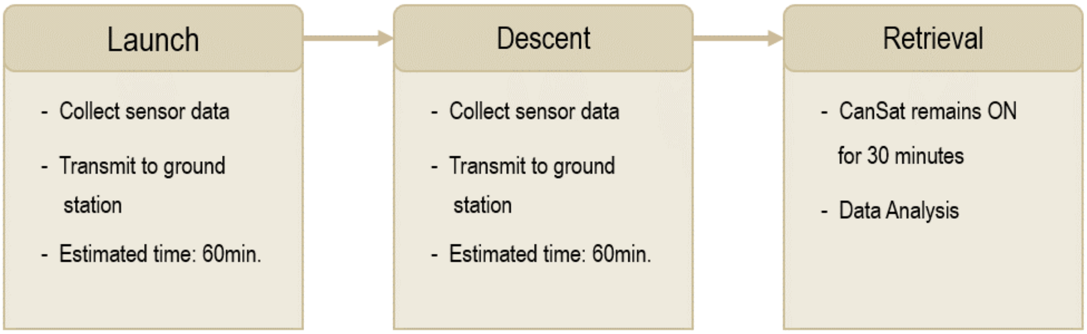


Figure 3.2. Can Satellite Operational Flowchart

IV. Secondary Mission

The Can Satellite is also expected to perform an auxiliary task aside from the primary mission. Remote sensing of suitable lands for vegetation is achievable by measuring environmental factors such as temperature, humidity, and carbon dioxide levels. By allowing set standards for the sensor readings, land areas can be classified according to appropriate crop conditions. This, in turn, gives insight to agriculturists and farmers on agricultural planning.

A. Additional Peripherals

To allow the CanSat to track its location during data measurement, a GPS module must be attached to the main system. In this project, a MediaTek MT3329 GPS module is proposed. Moreover, a XENSIV PAS CO_2 sensor is also included in the sensor subsystem. This measures carbon dioxide levels at the present location of the CanSat. The existing sensor readings from the primary mission will also be utilized for data analysis and classification of land areas.

B. Power Budget

It is assumed that most of the power is consumed by the ASP block of each peripheral. From the MediaTek MT3329 datasheet [24], the GPS module will typically have an output swing of 0.4-2.4V. These values are close to the output of the pressure sensor, so an ASP block with similar parameters will be used. The estimated power consumption of the GPS module is then 7.5mW [See Section VI.C.]

For the carbon dioxide sensor [25], the average power consumption is provided in the datasheet with a value of 11mW. Adding that to the previous estimation, 18.5mW would be needed to power the extra peripherals. With the remaining 3331.5mWh, the required 46.25mWh for the implementation of the carbon dioxide sensor is possible since there is enough power based on the power budget.

C. Memory Budget & Packet Information

For simplicity, it is assumed that data from the secondary mission peripherals will have the same resolution (12 bits) as the sensors used in the primary mission. It is also assumed that data is read from the peripherals every minute during the two-hour operation. All data is stored in the built-in ROM. The GPS module will have an output in the decimal degrees format which contain two floats.

$$120 \times 12 \text{ bytes} = 1440 \text{ bytes} = 1.44 \text{ kB}$$

Considering the peripherals from the primary and secondary mission, a total of 6 values (78 bits including a parity bit for each value) will be stored. Since the XMC1100-T038 features a 32-bit architecture, all data would be stored in 3 words where 18 bits are unused and set to 0. The memory budget would then be increased to 1.44 kB [See Appendix B].

Regarding packet transmissions, additional peripheral sensors will correspond to 36 more bits (12 bits × 3 sensor values) within the encapsulated data bits. Hence, the total bit length of each packet will be increased to 88 bits.

D. System Integration

The CO2 sensor is interfaced to the microcontroller's ADC using the same protocol as the other sensors (I2C). Note that the CO2 sensor has an average output swing of 0-3.3V, so a non-inverting amplifier of gain 1.2 can be used as an ASP block.

The GPS module will communicate with the microcontroller through the UART protocol. Required (latitude and longitude) data would be retrieved using the NMEA 183 standard [26]. Only the latitude and longitude are retrieved from the Global Positioning System Fix Data (GGA) Sentence.

V. Sensors and Sampling

Due to the limitations of Infineon Designer, the sensors were modeled after voltage generators. The team used a linear piecewise function to obtain the desired output for each sensor. Sensors were assessed based on its capability to convert the sensor readings to the correct voltage level. Computations regarding sensors and sampling are discussed in detail in Appendix C.

A. Pressure Sensor

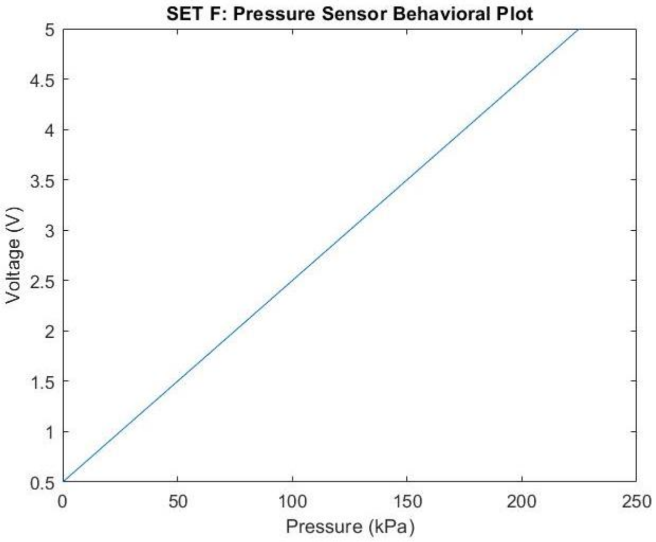


Figure 5.1. Pressure Sensor Behavioral Plot

Shown in Figure 4.1 is the voltage-pressure characteristic curve of the pressure sensor used in this project. It is assumed that the model is a linear function. Using two points in the graph (0, 0.5) and (225, 5), the relationship between voltage and pressure is then computed as: $V = 0.02P + 0.5$, where V is the voltage in volts and P is the pressure in kPa.

It is also assumed that the permissible pressure values range from 0 kPa to 101.325 kPa [8]. Under this assumption, the output of the pressure sensor will be between 0.5 V and 2.5265 V. Comparison between the pressure readings of the CanSat and the equivalent voltage output is then seen on Figure 4.2. Note that MATLAB was used in plotting for presentation purposes only [See Appendix D].

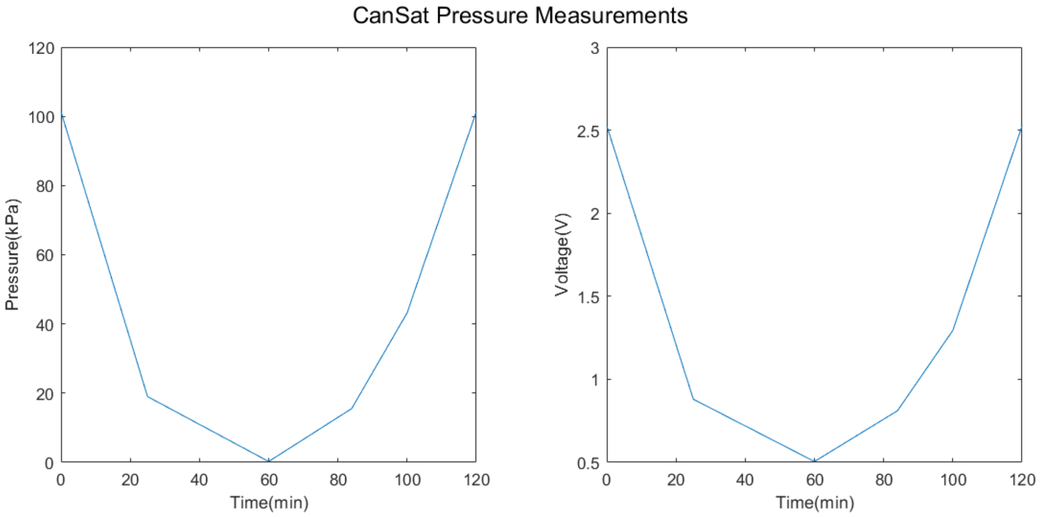


Figure 5.2. Comparison of Pressure Sensor Reading and the Voltage-Equivalent Plot

B. Humidity Sensor

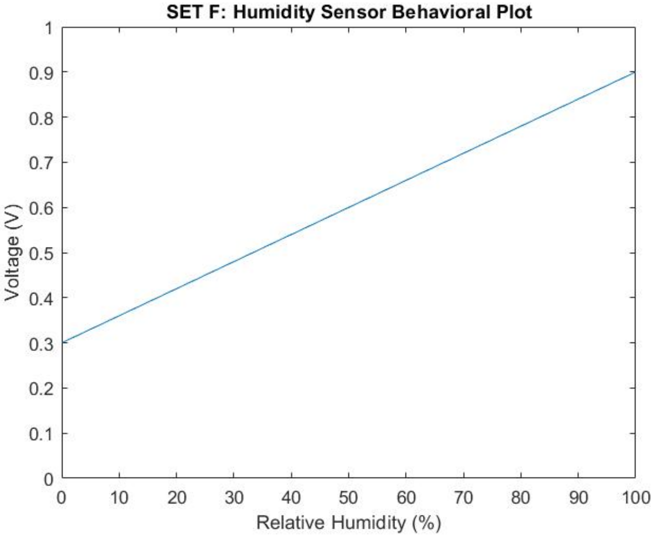


Figure 5.3. Humidity Sensor Behavioral Plot

The humidity sensor in this project exhibits a voltage-humidity behavioral plot as seen on the above graph. It is assumed that the relationship between the two parameters can be modeled as a linear function. The points (0, 0.3) and (100, 0.9) in the graph were used in solving the linear relationship between the two parameters. The resulting equation was determined to be: $V = 0.006H + 0.3$, where V is the voltage in volts and H is the relative humidity in %.

Measured relative humidity will have values ranging from 0% to 100%. Consequently, the output voltage will be between 0.3 V and 0.9 V. Shown in Figure 5.4 (right) is the generated voltage output based on the relative humidity readings seen on the graph on the left.

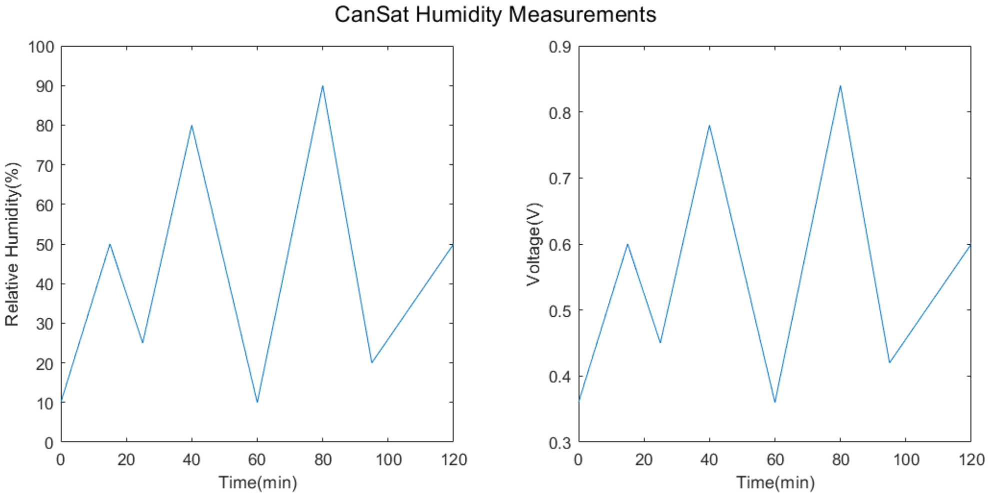


Figure 5.4. Comparison of the Humidity Sensor Reading and the Voltage-Equivalent Plot

C. Temperature Sensor

The project also used a thermistor with the following characteristic function:

$$R_{thermistor} = 10^4 \times e^{3574\left(\frac{1}{T}-\frac{1}{298.15}\right)}$$

$R_{thermistor}$ is the resistance in Ohms and T is the temperature in Kelvin. The resistance was then converted to a voltage level using a voltage divider shown below, with V_{DD} set to 5 V. It was specified that at 0°C, $V_{thermistor}$ has a value of approximately 1.36 V [9]. Thus, R_2 was set as an 80 kΩ resistor. The equivalent voltage was then shown as: $V = 5(\frac{R_t}{R_t + 80000})$.

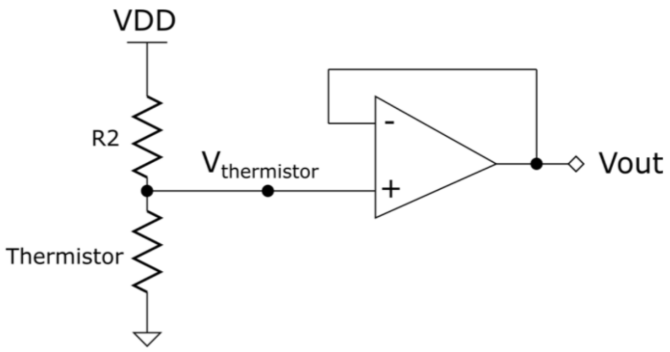


Figure 5.5. Temperature Sensor ASP Block

Temperature in the stratosphere (6 to 20 km above the Earth's surface with an upper boundary of 50km) ranges from -51°C to -15°C [10]. Meanwhile, the average maximum temperature in the Philippines is about 29°C [11]. Using these values, the range of the voltage output can then be set from 0.4819 V to 4.4152 V. Then the difference between the plot of the original temperature readings and the equivalent voltage output can be observed on the figure below.

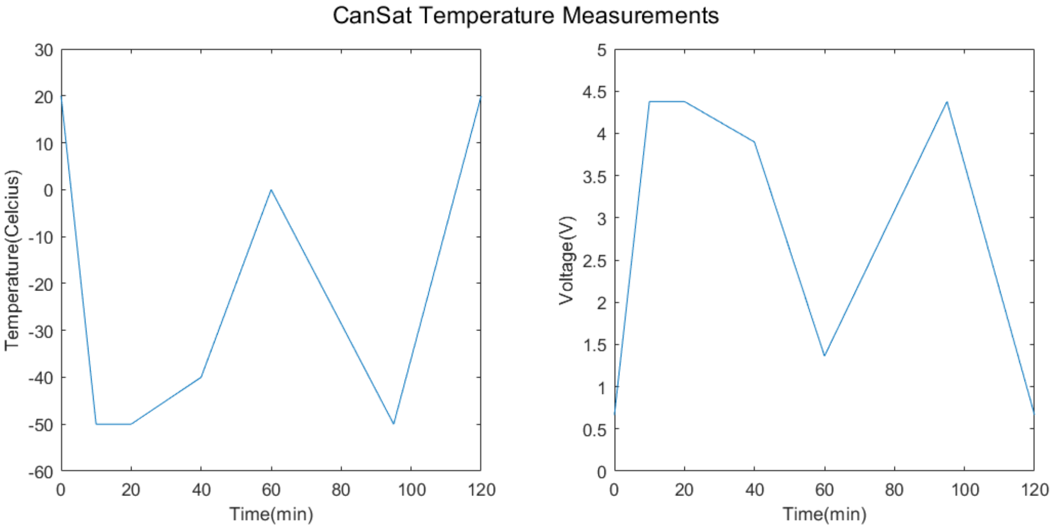


Figure 5.6. Comparison of the Temperature Sensor Reading and the Voltage-Equivalent Plot

D. Analysis

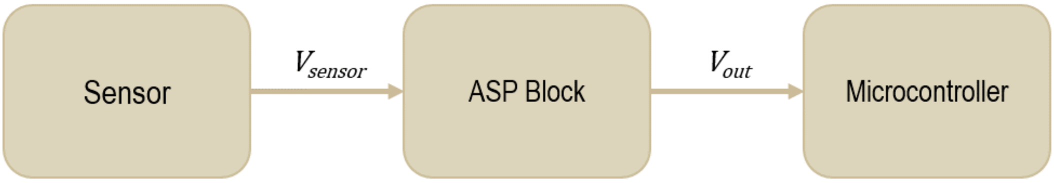


Figure 5.7. Data Processing Module Block Diagram

For any of the analog signal processing (ASP) blocks, voltage output should be within 0 V to 5 V (V_{ref}). Note that for all the sensors, using a voltage follower to interface sensor output to the microcontroller unit (MCU) would satisfy this criterion. However, this results to a significant value of quantization error for a small voltage swing. In these cases (i.e., humidity sensor and pressure sensor), a non-inverting amplifier with the appropriate gain was used. A voltage follower was also used as an ASP block for the temperature sensor since it produces a well-defined mapping of values.

In determining the resolution of the ADC, the approximated ranges of the sensor output (0-101.33 kPa, 0-100%, and -51-29°C) was used. Considering only integer values, then at most 102 units need to be represented. Note that 7 bits are enough to represent any integer data of sensor readings. 10 bits are used to account for at most one decimal place. To solve for the ADC resolution, the formula below is used where; N is the resolution, U is the number of units to represent, and D is the number of decimal places.

$$2^N = U \cdot 10^D$$

$$N = \log_2 U \cdot 10^D$$

$$\log_2 102 = 6.67 \approx 7 \qquad\qquad \log_2 1020 = 9.99 \approx 10$$

It is assumed that the 5 V voltage swing is not maximized for each ASP block, so two additional bits are used such that the resolution is 12 bits. Lastly, a check bit set to even parity is added for a total of 13 bits per sensor data. The two equations below are used in converting the 12-bit code to an analog voltage level and vice-versa:

$$ADC_{code} = 2^N \left(\frac{V_{out}}{V_{ref}}\right)$$

$$V_{out} = V_{ref} \left(\frac{DAC_{code}}{2^N}\right)$$

To determine the sampling frequency of the ADC, the equation below was used; where $r = \frac{\Delta A}{\Delta t}$ is the maximum rate of change of the sensor output, E is the maximum allowable error, and N is the resolution [12]. The maximum allowable error is set to 5%. As solved earlier, the ADC will have a 12-bit resolution.

$$\frac{r}{E \cdot 2^N} \leq f_s \leq r$$

To solve for r, the given profile observed by the CanSat is used (refer to Figures 5.2, 5.4, and 5.6). Calculations are simplified by assuming that at ADC_{code} 0x0000 the sensor measurement is at minimum and maximum at ADC_{code} 0x1000. From the given plots, the maximum rate of change r was calculated as 134, 110, and 359 ADC units for the pressure, humidity, and temperature sensor, respectively [See Appendix D]. The least of these three slopes was then set as the upper bound of f_s. From the computed range of frequencies, a sampling rate value of 1 Hz (or 1 sample per second) was selected.

$$r = 110\ \frac{\text{ADC units}}{\text{min}} \approx 1.8333\ \frac{\text{ADC units}}{\text{s}}$$

$$\frac{1.8333}{0.05 \cdot 2^{12}} \leq f_s \leq 1.8333$$

$$0.0089 \leq f_s \leq 1.8333$$

$$f_s = 1\ \text{Hz}$$

VI. Budget Studies

A. Power Budget

The voltage source used in this project was a single 3.7 V battery with a capacity of 1500mAh, translating to 5500mWh of energy. It was assumed that most of the power consumption will come from the XMC1100 microcontroller, the communications module, and the ASP block in the data processing module. Estimates were provided as a guide to circuit design. The *Remaining Power Budget* would then imply that the battery capacity can sustain the whole system for the whole duration of the CanSat operation.

The table below summarizes the estimates made for the Power Budget:

Parameters	Units	Value
Battery Capacity	mWh	5550
PMU Power Consumption	mW	-650
Comms Power Consumption	mW	-300
ASP Block Power Consumption	mW	-100
Sensors Power Consumption	mW	-50
CanSat System Power Consumption	mWh	-2750
MCU Power Consumption (Comms & Data Processing)	mWh	-100
MCU Power Consumption (PMU)	mWh	-100
Remaining Power Budget	**mWh**	**2600**

Table 6.1. Power Budget Parameters

B. RF Link Budget

The RF Link Budget was computed considering the maximum descent height from deployment of the CanSat, as well as path losses present throughout packet transmissions. Within a set *Link Margin*, the minimum transmitted power was obtained and used as a design goal for assembly of the transmitter circuit.

The table below summarizes the Link Budget parameters with their corresponding values. Detailed analysis can be found under Appendix A.

Parameters	Units	Value
Operating Frequency	MHz	0.5
Maximum Distance from Ground Station	km	50
Minimum Transmitted Power	dBW	-12.59
Tx Antenna Gain	dB	0
Maximum Path Loss	dB	60.41
Rx Gain	dB	0
Rx Sensitivity	dB	-80
Received Power	dBW	-73
Link Margin	**dB**	**7**

Table 6.2. Link Budget Parameters

C. Memory Budget

In Section V-D, 12 bits were allocated for each sensor reading. Considering all three sensors, then a total of 36 bits are needed to convert the equivalent voltage levels. For each sensor reading a parity bit is appended for error checking for a total of 39 bits.

The 12 bits (excluding the parity bit) added in the communications module are predefined and constant. To save space, these 12 bits are not stored in memory. Since the XMC1100-T038 features a 32-bit architecture, data is stored in two words (64 bits). Note that 25 of these bits are unused, so it is set to 0 by default [See Appendix B].

All sensor readings during the two-hour flight are recorded. In total, there will be 120 readings from each sensor. Thus, 0.96 kB of memory is needed for storage as seen on the computation below.

$$120 \times 8 \text{ bytes} = 960 \text{ bytes} = 0.96 \text{ kB}$$

VII. Circuit Analysis

This section discusses the functionality of the circuit used in each module. The motivations behind the choice of circuit topology are also provided. Derivations are also included to explain how the circuit works for each subsystem.

A. Power Module

Since the system demands varying input voltage values, two switching converters were used as the power management unit. Using a combination of a Single-Ended Primary-Inductor Converter (SEPIC) and a Ćuk converter, the required power and supply voltages from the other modules were achieved.

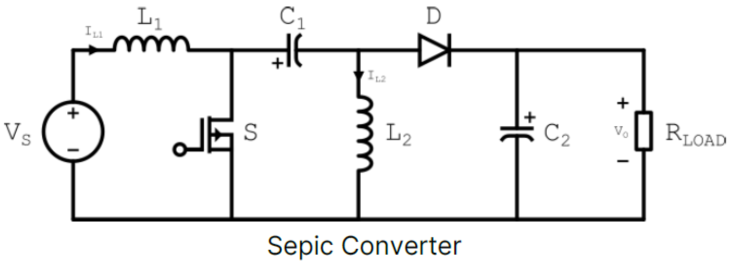


Figure 7.1. SEPIC Converter Circuit Diagram

The SEPIC converter is a type of switching DC-DC converter that can produce both step-up and step-down voltages. Unlike the buck-boost and Ćuk converter, SEPIC converters produce a non-inverted output. This is important because if the output is inverted, there is a need to pass through an inverting amplifier with a gain of 1 to achieve desired voltage value including its polarity. This additional circuit can decrease the power management's efficiency [13]. Through this converter, the positive source voltage can be implemented.

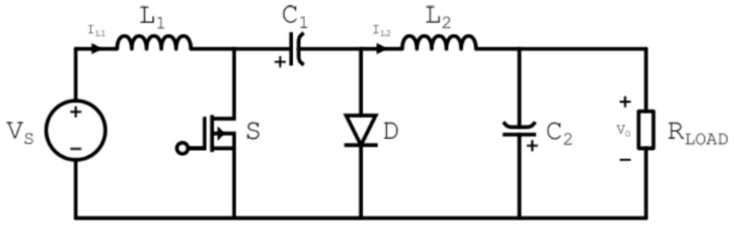


Figure 7.2. Ćuk Converter Circuit Diagram

To produce the negative voltage supply for the two subsystems, a Ćuk converter was also used. Similar to the buck-boost converter, it produces a magnitude voltage either higher or lower than the input voltage. The Ćuk converter was chosen over the Buck-Boost converter as it has less ripple effect, which can affect circuit performance [14].

The power module also needs a feedback system, which regulates the output voltages of the two converters. To achieve this, the ADC is used.

B. Communications Module

The communications module solely consists of the transmitter circuit, with main stages consisting of an Oscillator (RC Multivibrator-Chebyshev LPF) Cascade, a Binary Phase-Shift Keying (BPSK) Modulator, and a Class AB Transistor Power Amplifier. Derivation of component values used in this module can be found under Appendix F-1.

For the Oscillator block, the following configuration was used:

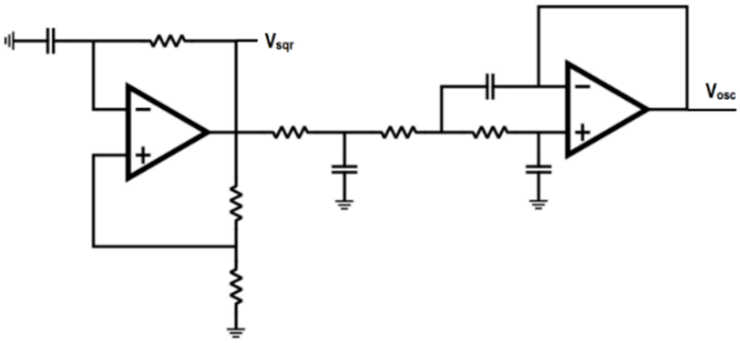


Figure 7.3. Op-amp Cascade Oscillator Circuit Diagram

The first op-amp block functions as an astable multivibrator, which produces a square wave V_{sqr} (of defined frequency f) toggling between the input supply voltages $\pm V_{CC}$. Then, the second op-amp block serves as a low-pass filter. This removes odd harmonics of the square wave input, to produce a sinusoidal wave output. The output V_{osc} is then used as the carrier signal for the phase-shift modulator.

For the BPSK Modulator block, the following configuration was used:

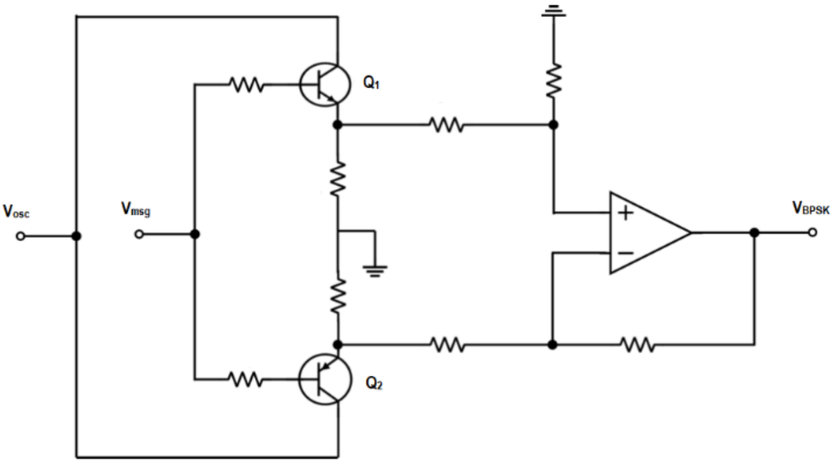


Figure 7.4. BPSK Modulator Circuit Diagram

Transistors Q_1 and Q_2 are both configured as emitter followers, and their respective outputs are fed into a differential op-amp configuration. The message signal from the MCU is fed into the base resistances of both transistors, effectively serving as a switch alternating the cut-off operation of each transistor.

Moreover, it can be shown [See Appendix F-1] that the output of the BPSK Modulator is approximated by:

$$V_{BPSK} = V_{o1} - V_{o2}$$

Where V_{o1} and V_{o2} are the emitter follower outputs of Q_1 and Q_2, respectively. This implies an inverted output when only Q_2 is conducting. Since Q_2 operates at the negative cycle of V_{msg}, it can be concluded that the BPSK modulator introduces a phase shift of 180° when there is a negative input detected.

Finally, for the Power Amplifier block, a Class AB Power Amplifier was used. This will give unity voltage gain and drive the antenna for the required transmitted power. This configuration also provides high efficiency and low distortion, as compared to Class A and Class B amplifiers [15]. Biasing diodes were used to provide the necessary threshold voltages on transistors T_1 and T_2 to ensure Class AB operation.

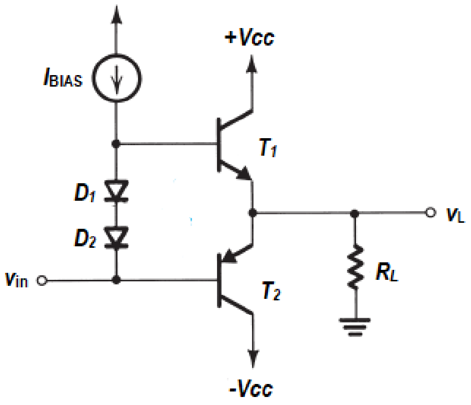


Figure 7.5. Transistor Power Amplifier Circuit Diagram

C. Data Processing Module

In Section V, it was discussed that the pressure sensor and the humidity sensor will use a non-inverting amplifier as an ASP block. Meanwhile, a voltage follower was used to interface the temperature sensor output to the microcontroller.

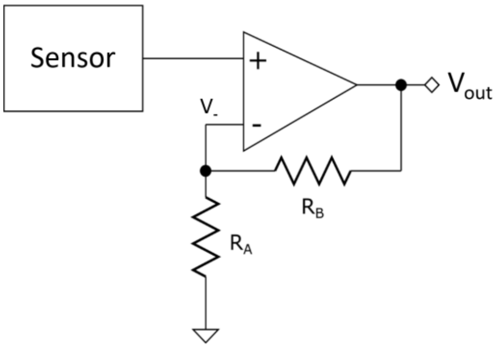


Figure 7.6. Humidity and Pressure Sensor ASP Block
a. Non-Inverting Amplifier

A non-inverting amplifier is an operational amplifier configuration where a positive gain is applied on the input signal. The derivation of the voltage gain is shown below. For the pressure sensor, R_B is set to 27 kΩ and R_A to 62 kΩ resulting to a voltage gain of 1.4354. The humidity sensor will use a 68kΩ resistor and 22 kΩ as R_B and R_A, respectively. [See Appendix E-1] Consequently, the ASP block of the humidity sensor will have a gain of 4.0909.

$$\begin{aligned} V_{sensor} &= V_{-} \\ I_{R_B} = I_{R_A} &= \frac{V_{sensor}}{R_A} \\ V_{out} &= I_{R_B}(R_B + R_A) \\ &= \frac{V_{sensor}}{R_A}(R_B + R_A) \\ &= V_{sensor}\left(1 + \frac{R_B}{R_A}\right) \end{aligned}$$

The temperature sensor will use a voltage follower as the ASP block since the 0 V to 5 V voltage swing is nearly maximized by the sensor output. However, there is a need to decrease the input to the ASP such that the output will not reach saturation. The voltage input is lowered using a voltage divider as in Figure 8.8.b, where R_A and R_B, are set to 24kΩ and 110kΩ, respectively. This implies that the ASP input is multiplied by a factor of 0.8209.

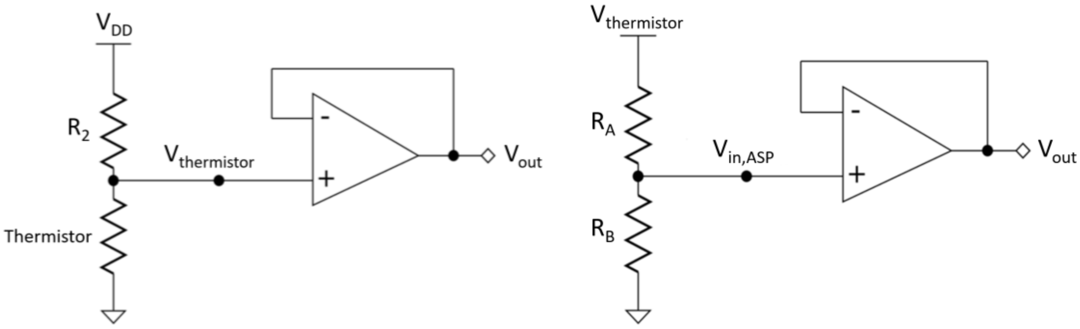


Figure 7.7. Temperature Sensor ASP Block
a. (left) Unity Gain Buffer b. (right) Divider with Voltage Follower

In the voltage follower configuration, a voltage gain of 1 is applied to the input signal. This is a special case of the non-inverting amplifier where R_B is a short circuit and R_A is an open circuit. In this section, it is assumed that the amplifier is ideal such that $A_o \gg 1$. As proof that the configuration has unity gain, computations are shown below.

$$V_{thermistor} = V_+$$
$$V_{out} = V_-$$
$$V_+ = V_-$$
$$\therefore V_{thermistor} = V_{out}$$

VIII. Methodology

A. Preparation Phase

An operational flowchart was first created to provide a descriptive overview of the project specifications and outcomes. A detailed discussion on CanSat operations can be found under Section III.

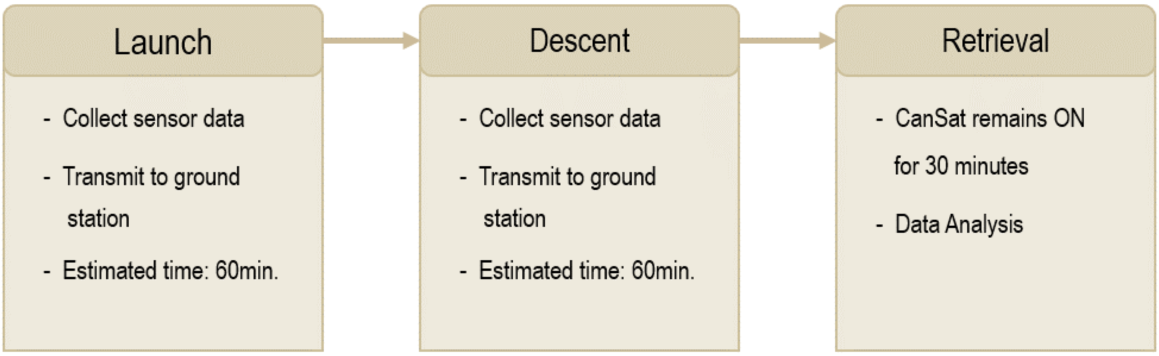


Figure 3.2. Can Satellite Operational Flowchart

Additionally, preliminary design assessments were performed to determine the components required by the project. For each subsystem, the following budget studies were made in preparation for the assembly phase:

Power Module

From Section VI-A, we refer to Table 6.1 for the estimated power budget:

Parameters	Units	Value
Battery Capacity	mWh	5550
PMU Power Consumption	mW	-650
Comms Power Consumption	mW	-300
ASP Block Power Consumption	mW	-100
Sensors Power Consumption	mW	-50
CanSat System Power Consumption	mWh	-2750
MCU Power Consumption (Comms & Data Processing)	mWh	-100
MCU Power Consumption (PMU)	mWh	-100
Remaining Power Budget	**mWh**	**2600**

Table 6.1. Power Budget Parameters

Communications Module

From Section VI-B, we then refer to Table 6.2 for the computed link margin:

Parameters	Units	Value
Operating Frequency	MHz	0.5
Maximum Distance from Ground Station	km	50
Minimum Transmitted Power	dBW	-12.59
Tx Antenna Gain	dB	0
Maximum Path Loss	dB	60.41
Rx Gain	dB	0
Rx Sensitivity	dB	-80
Received Power	dBW	-73
Link Margin	**dB**	**7**

Table 6.2. Link Budget Parameters

Additionally, the minimum power transmitted to the load should be at least $13.59\ \mathrm{dBW}$, translating to $55.07\ \mathrm{mW}$ of power.

Data Processing Module

The memory budget is calculated using the ADC resolution computed in Section V-D and the fact that the XMC1100 features a 32-bit architecture. To store all sensor data, 0.96 kB of memory is needed which will make use of the device’s built-in ROM. This is discussed in detail in Section VI-C.

B. Assembly Phase

Then, circuit topologies were determined and designed to meet the project specifications. While Section VII discusses the functional blocks involved in this project, this Section summarizes the selection of component values for the CanSat modules. In this phase, the team used Infineon Designer to construct schematic diagrams representing the CanSat system.

Power Module

The schematic diagram for the power module was constructed as shown in Figure 8.1. V_{in} models the battery pack used by the PMU, while V_{SEPIC} and V_{Cuk} represent the supply voltages provided by the PMU to the other subsystems. The computation for the component values can be seen in Appendix G-1.

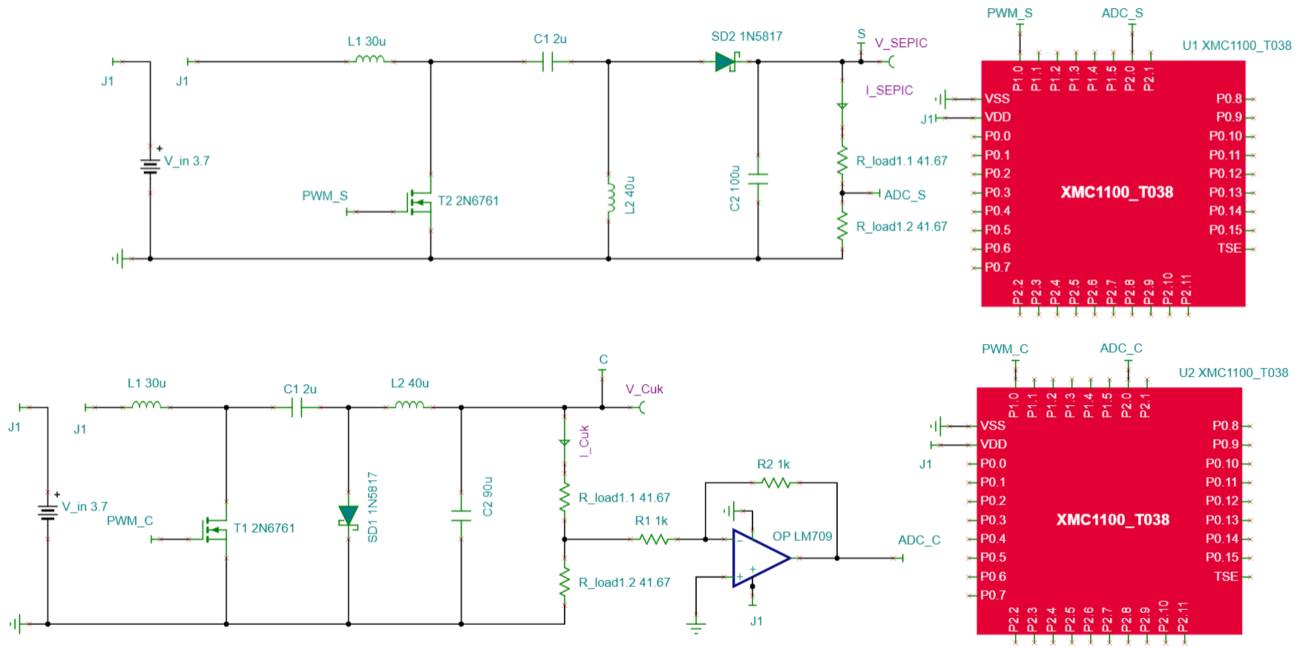


Figure 8.1. Power Module Schematic Diagram

The initial duty cycles of the converters are set to be 67.7% and 67.9% for the SEPIC and Ćuk converters, respectively.

Communications Module

The schematic diagram for the communications module was constructed as shown in Figure 8.2. V_{msg} represents the packet bitstream to be transmitted, while V_{load} represents the transmitted signal. A detailed discussion on the design process can be found under Appendix F. It is also assumed that the message bitstream will be transmitted at a rate of 50 kpbs, translating to a bit period of $20\ \mu\text{s}$. For this diagram, V_{msg} was modeled by a voltage generator producing a 50 kHz square wave.

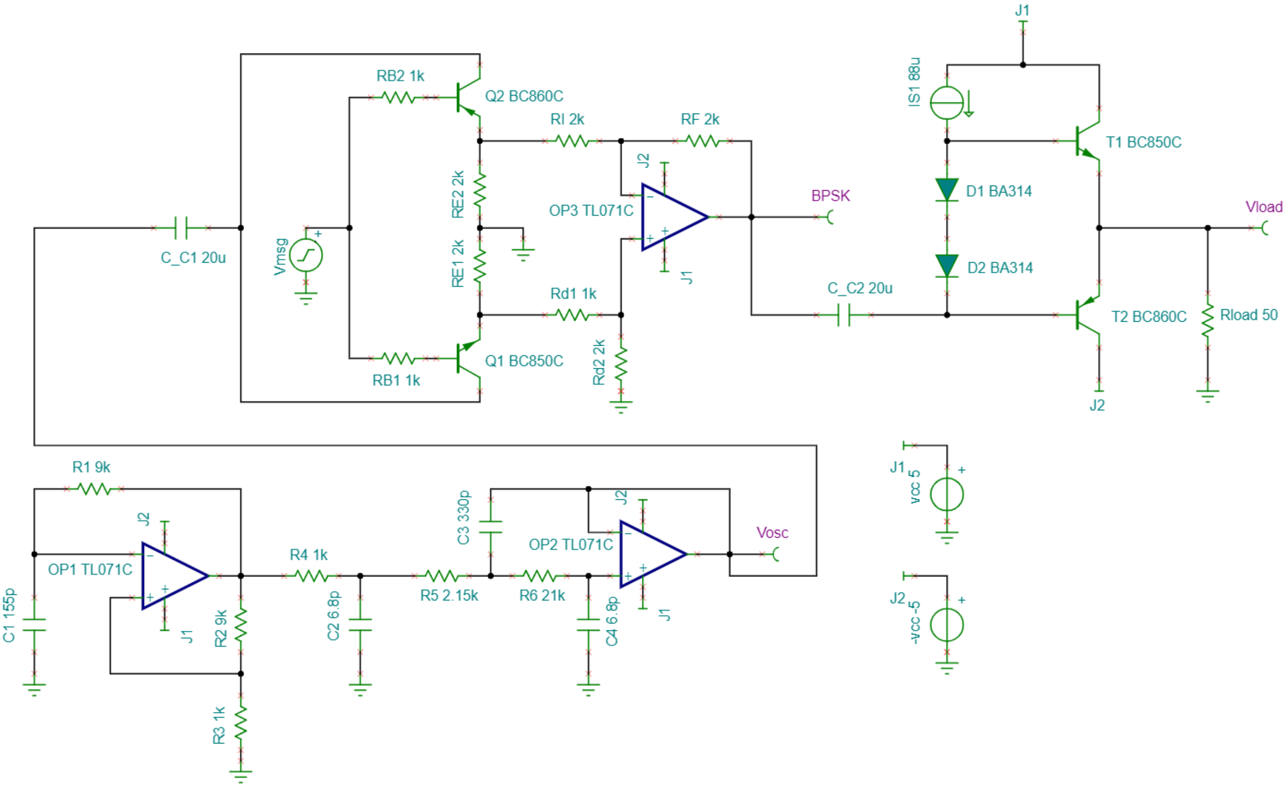


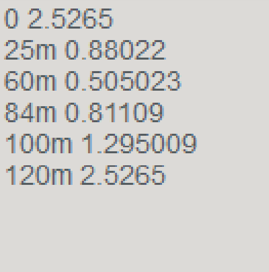
Figure 8.2. Communications Module Schematic Diagram

Data Processing Module

Due to the limitations of the simulation software, data read per minute during the two-hour mission is provided. For each sensor, there are 120 predetermined data points. These given values are mapped to voltage levels in Microsoft Excel using the formulas derived in Section V. We take note of the extremes of the sensor data to determine the appropriate ASP block for each sensor.

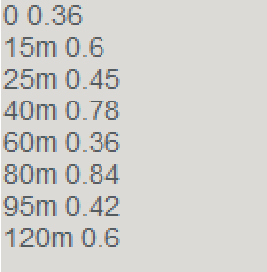
A linear piecewise voltage generator in Infineon Designer is then used to unify the voltage levels into a plot. The configuration is shown below.

Pressure Sensor:



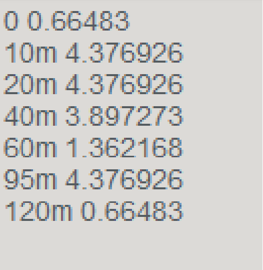
```
0 2.5265
25m 0.88022
60m 0.505023
84m 0.81109
100m 1.295009
120m 2.5265
```

Humidity Sensor:



```
0 0.36
15m 0.6
25m 0.45
40m 0.78
60m 0.36
80m 0.84
95m 0.42
120m 0.6
```

Temperature Sensor:



```
0 0.66483
10m 4.376926
20m 4.376926
40m 3.897273
60m 1.362168
95m 4.376926
120m 0.66483
```

Using the maximum voltage level for each sensor, the appropriate ASP block is determined such that the 5 V voltage swing is optimal. However, using a real operational amplifier (LM709), the output swing is reduced to 3.7 V.

For the humidity and pressure sensors, the ASP block is selected as a non-inverting amplifier since the maximum voltage level of the sensor is less than the optimal voltage swing. On the other hand, the temperature sensor will use a unity gain amplifier with a voltage divider on the input side. Computation of the component values are discussed in detail in Section VII.

The ASP block output is connected to the available ADC pins of the XMC1100. Since there are no restrictions on which sensor data should be read first, any of the ASP block outputs are connected to any appropriate pin. The temperature, humidity, and pressure ASP block outputs are fed to pins 2.2, 2.3, and 2.4, respectively.

Lastly, the schematic diagram for the data processing module was constructed as shown in Figure 8.3. $V_{Xsensor}$ represents the piecewise voltage generators for the analog sensor readings *(P-pressure, T-temperature, H-humidity)*, while V_{X-ASP} represents the gain-adjusted voltage readings to be fed into the ADC.

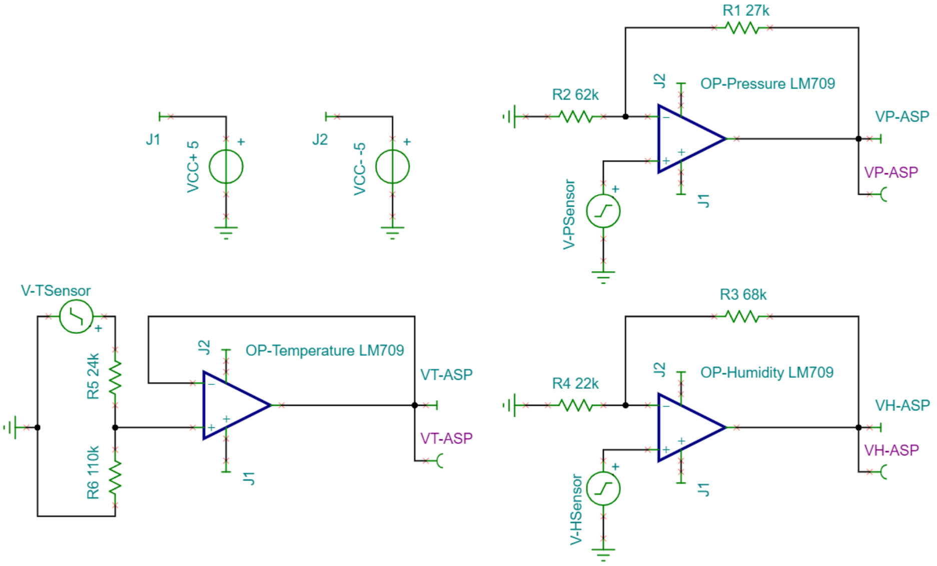


Figure 8.3. Data Processing Module Schematic Diagram

MCU Integration

The XMC 1100-T038 microcontroller unit was also configured to connect the ASP blocks with the transmitter circuit. Using Infineon DAVE, the main clock was set to $32\ \mathrm{MHz}$. For Analog-to-Digital conversion of the analog sensor data, pins 1, 37, and 38 were configured as ADC measurement pins. Then, pins 20 and 21 were configured as digital output pins for the packet bitstream. Figure 8.4 shows the virtual pin configurations of the MCU:

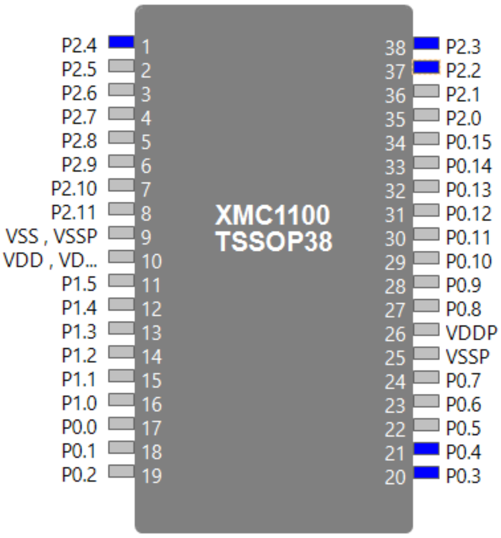


Figure 8.4. XMC 1100-T038 Virtual Pin Configuration

A C program file was also programmed into the MCU to achieve ADC and packet formatting. A detailed discussion of this program can be found under Appendix I. Then, the schematic diagram shown in Figure 8.5 was constructed to integrate the MCU into the data processing and communications module:

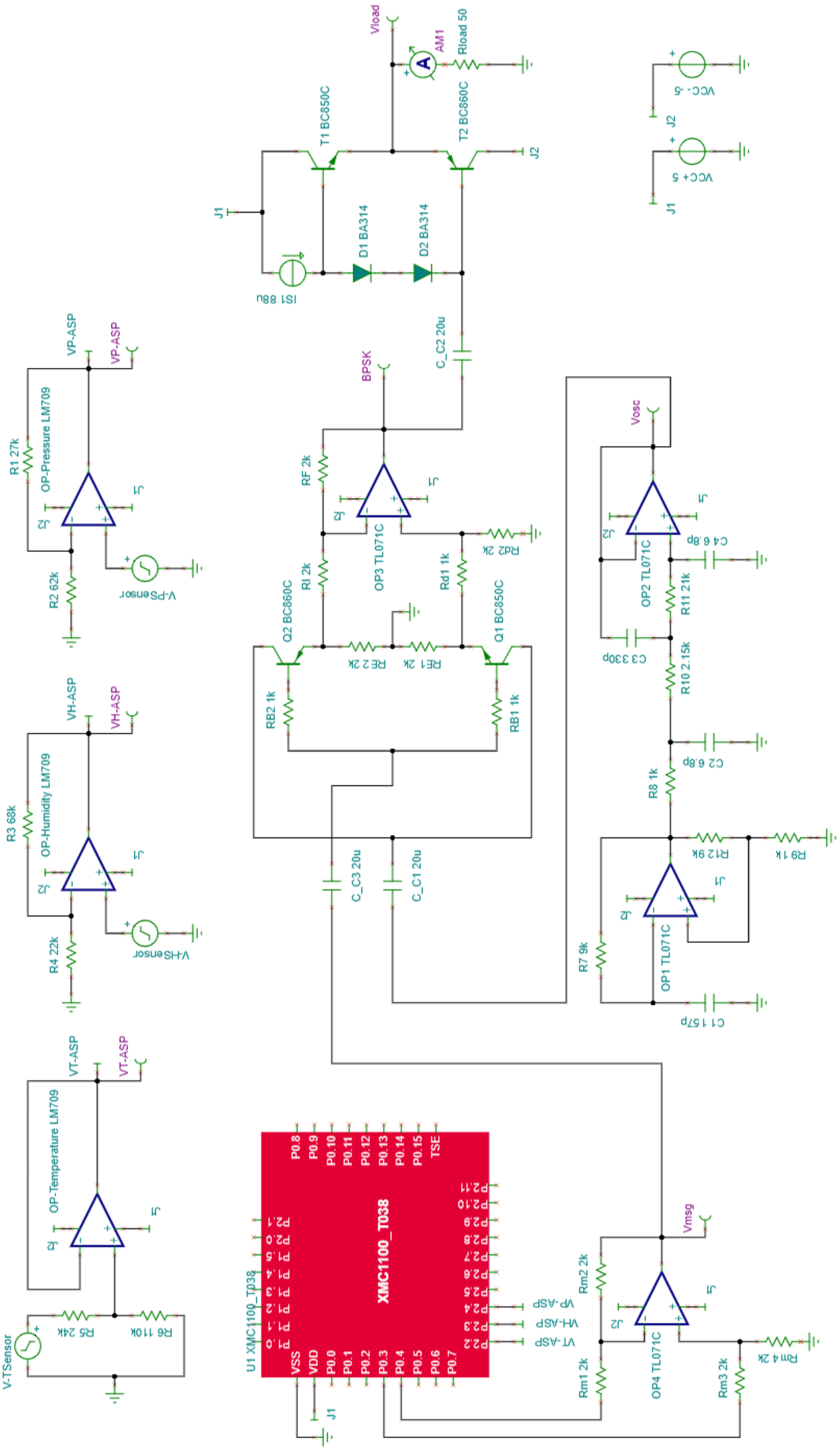


Figure 8.5. Integrated ASP-Transmitter Schematic Diagram

C. Simulation Phase

After circuit analysis and design, the corresponding MCU simulation-related software were configured using Infineon Designer:

Power Module

The load shown in Figure 8.1 was designed as a voltage divider since the microcontroller has a limit for input voltages. This was taken into consideration for the program. Looking at the Ćuk converter, a unity gain inverting amplifier was added in the circuit. This satisfies the positive voltage input required by the ADC. For the microcontroller, the pins used are: P1.0 for the PWM, P2.0 for the ADC input, and VSS and VDD for the sources. A C program file was programmed to create the voltage regulation scheme using `ADC_Measurement`. The C program file can be seen at Appendix I.

Communications Module

The message signal used in Figure 8.2 was a voltage generator to model the bitstream output. The practical implementation of the message signal was achieved using the MCU Integrated circuit as seen in Figure 8.5.

The C program shown in Appendix I takes the binary ADC measurements and formats it into packets with pre-defined start/stop bits and bit parity check bits. Then, two digital output pins were used to output logic high and low values, with pin P0.3 dedicated to logic high outputs and pin P0.4 dedicated to logic low outputs. Moreover, a delay function was implemented to hold the logic outputs to match the required bit rate of $50\ \text{kbps}$, translating to a 50kHz simulated signal.

It is also observed from Figure 8.5 that a differential amplifier was used to connect the XMC 1100 with the transmitter circuit. Since digital output voltages are limited to non-negative values, the differential amplifier configures the digital outputs to produce a square wave with both positive and negative cycles.

Data Processing Module

It is first noted that in XMC1100, ADC readings are limited to one global register. This means that when a value is read from one pin, the previous reading from another pin is deleted. Hence, an array is used to store the results and process it when all pins have been read. The configuration and the program that enables this behavior can be seen on Appendix I.

To verify if the ADC is correctly reading the voltage levels, the output bitstream and the read voltage level may be compared. However, this proved to be a tedious task since the bitstream is not readable and there are 120 data points per sensor to be compared.

Thus, a program that functions similarly as a comparator circuit is used. The threshold voltage is set at 2.5 V such that ADC readings higher than the set value will produce a high; otherwise, the output is low. The results of this verification test are shown on the next chapter.

IX. Results and Data

Using the simulation outputs obtained from Section VIII, the system budget parameters were assessed to verify if the CanSat is operating within the estimated margins. First, the power budget was computed accounting for all subsystems:

Power Budget

The table below summarizes the power budget analysis for the whole CanSat system:

Component	Operating Voltage (V)	Operating Current (mA)	Power (mW)	Duration (h)	Energy Consumption (mWh)
XMC1100	3.7	6.1	22.57	2.5	56.425
XMC1100	5	6.1	30.5	2.5	76.25
PMU	5	60.09	602	2.5	1505
Comms	5	41.96	209.8	2.5	524.5
Pressure	2.5	0.005	0.0125	2.5	0.03125
Humidity	3.6	0.0163	0.5400	2.5	1.35
Temperature	5	0.1500	0.0647	2.5	0.16175
ASP	5	2	21.9064	2.5	54.766
Total	-	-	891.3936	2.5	2218.484

Table 9.1. CanSat System Power Consumption

The total power consumption is 2218.484 mWh. This is the expected power consumption for the whole flight of the CanSat including the recording of data. The 3.7 V battery is rated at 1500 mAh. This means that there is 5550 mWh energy available. There is still 3331.5 mWh remaining after the whole duration of the flight. This translates to a *Remaining Power Budget* of approximately 60% of the battery capacity, implying that the power consumption is still on budget. This is expounded in Appendix H.

Power Module

Figures 9.2 and 9.3 show the transient responses of the converters designed for the power management system.

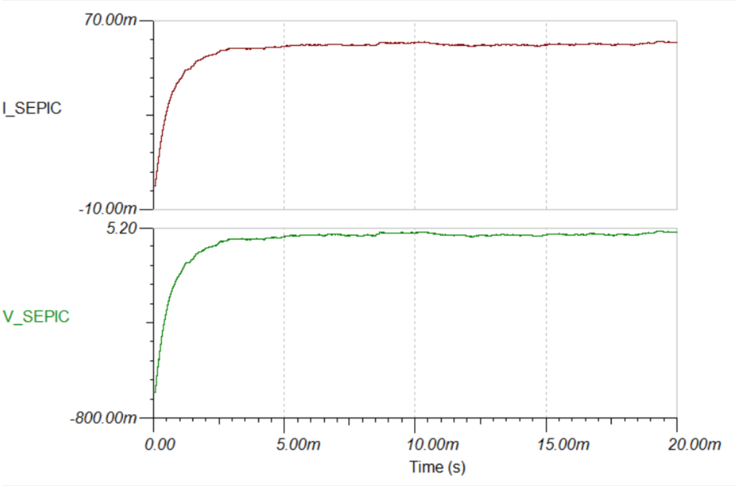


Figure 9.2. SEPIC Converter Transient Response (0-20 ms)

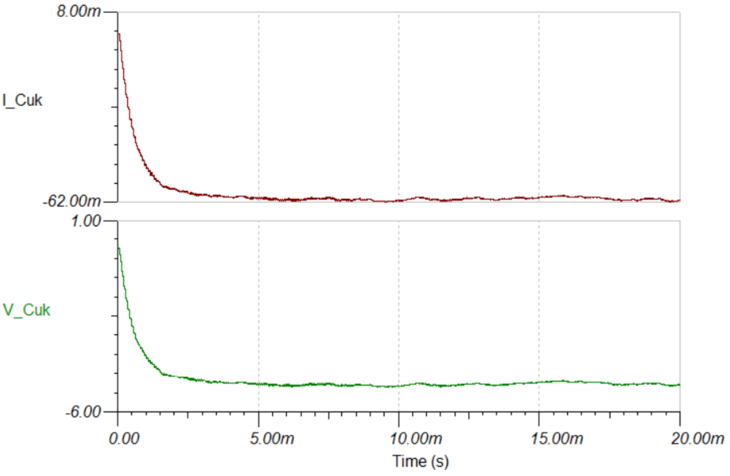


Figure 9.3. Ćuk Converter Transient Response (0-20 ms)

As seen on the figures above, the responses are stable with minimal output voltage ripple. The verification of the converters is done on the next chapter.

Communications Module

Next, the transient plot curves for the communications module were obtained as shown in Figure 9.4. Note that the message signal used was the modeled voltage generator.

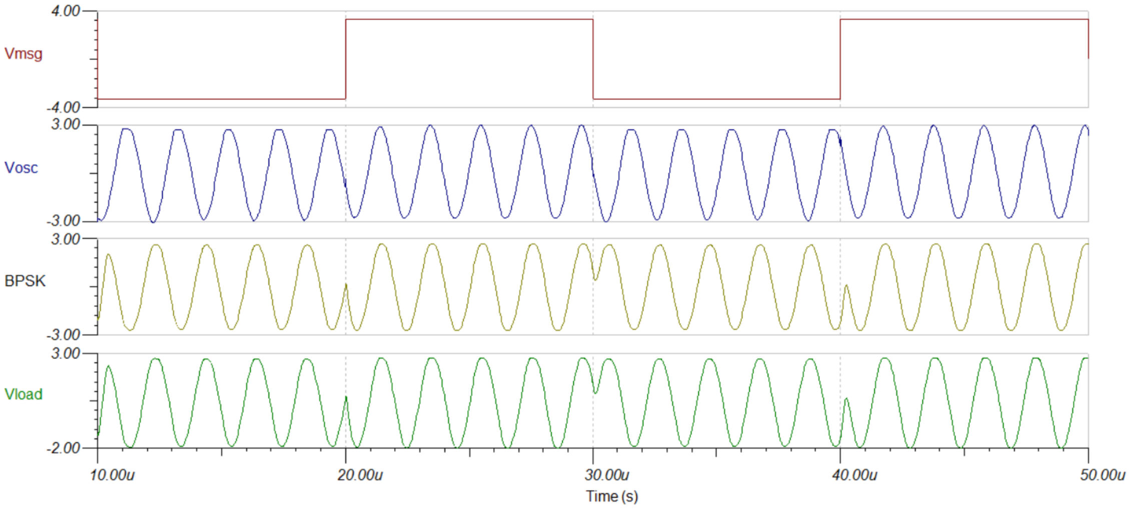


Figure 9.4. Communications Module Transient Response

Data Processing Module

Shown in Figure 9.5 are the output plot of the linear voltage generator for the corresponding sensor. Note that these converted voltage levels were obtained from the given atmospheric measurements.

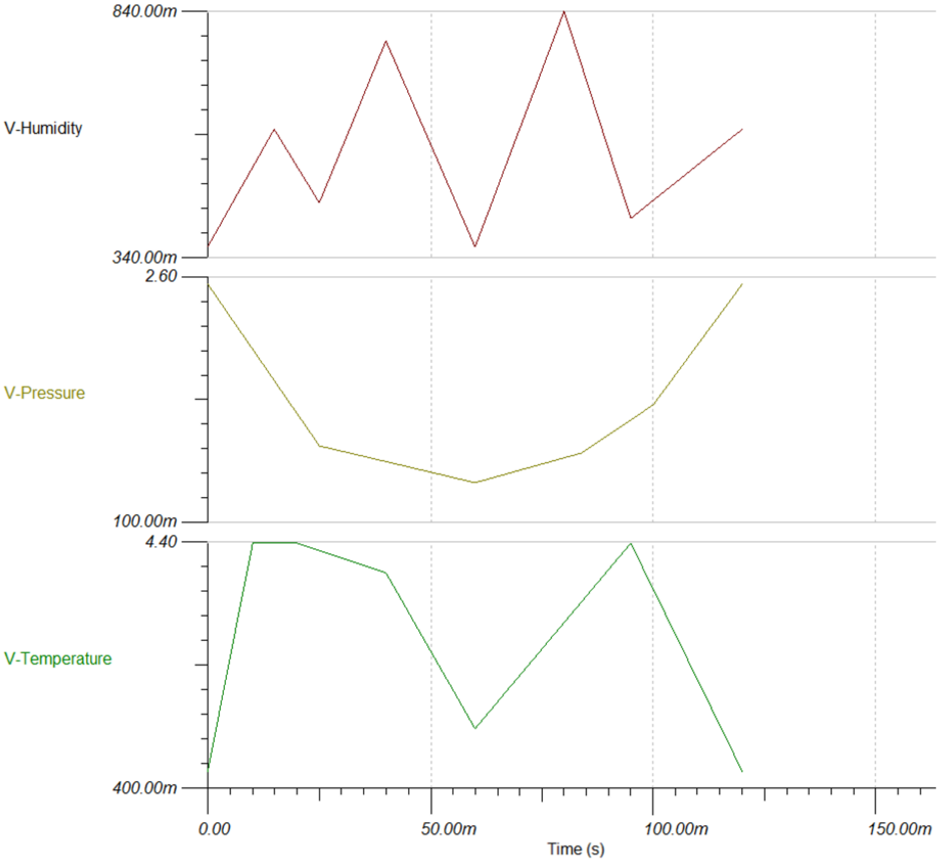


Figure 9.5. Linear Voltage Generator Output

The figures shown below are the output of the ASP block of each sensor. These voltage values are fed to the respective ADC pins of the microcontroller. Note that MATLAB is used in plotting for presentation purposes. For graphs produced using Infineon Designer simulations, see Appendix E-1.

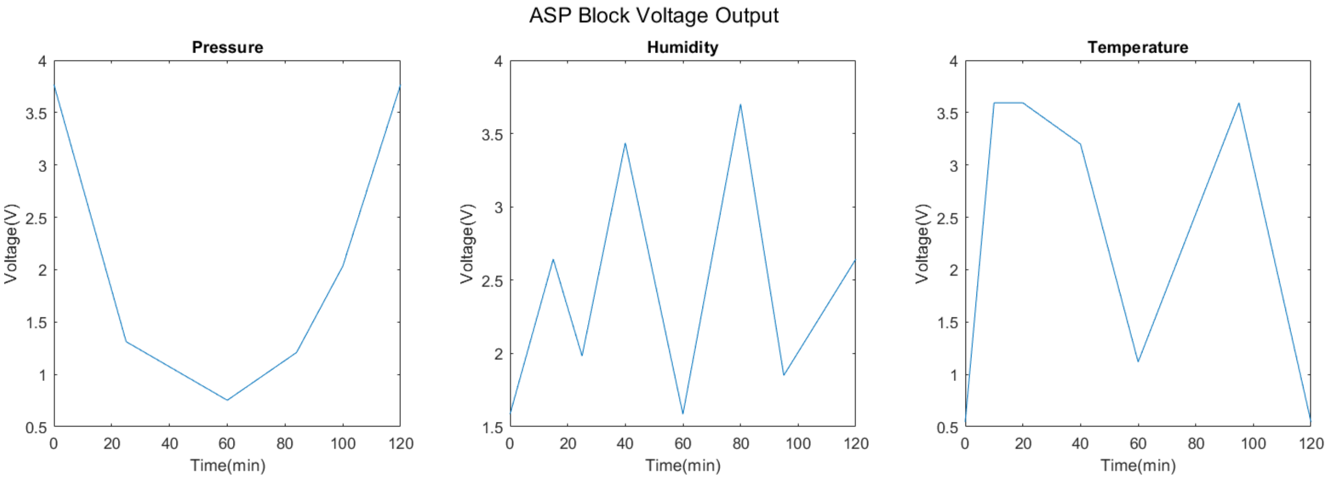
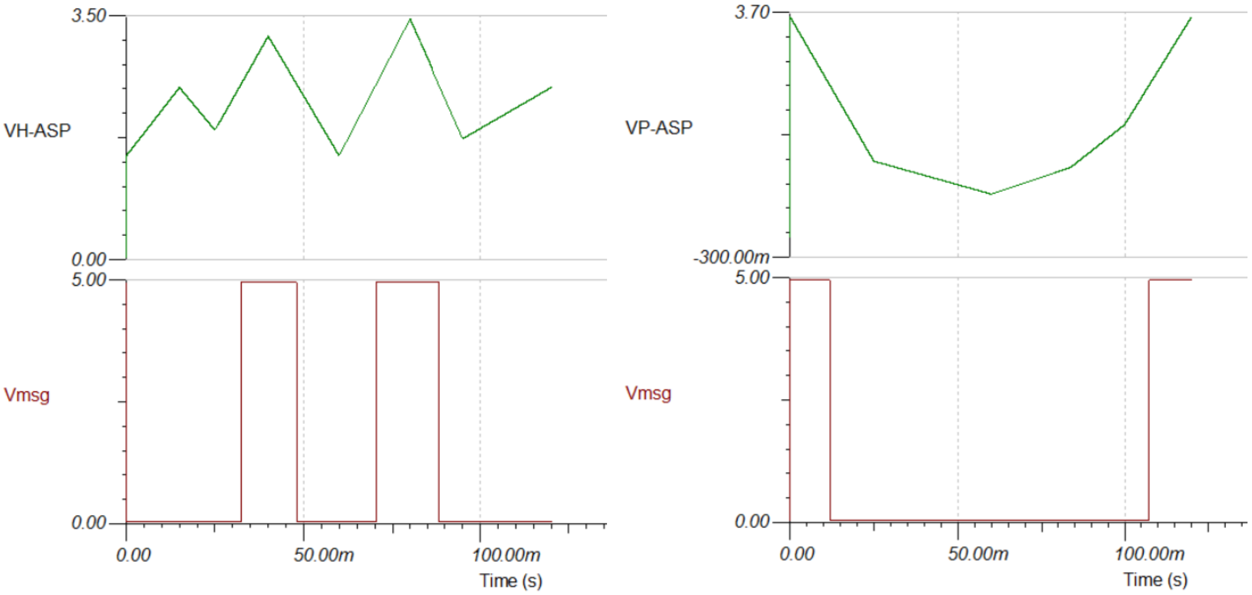


Figure 9.6. ASP Block Voltage Output

It is shown in Figure 9.7 that the ADC configuration is functioning properly using a program which operates similarly as a comparator.



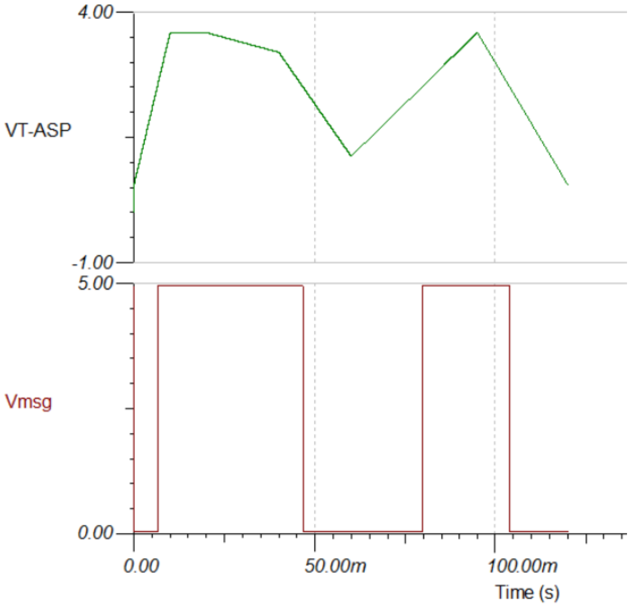
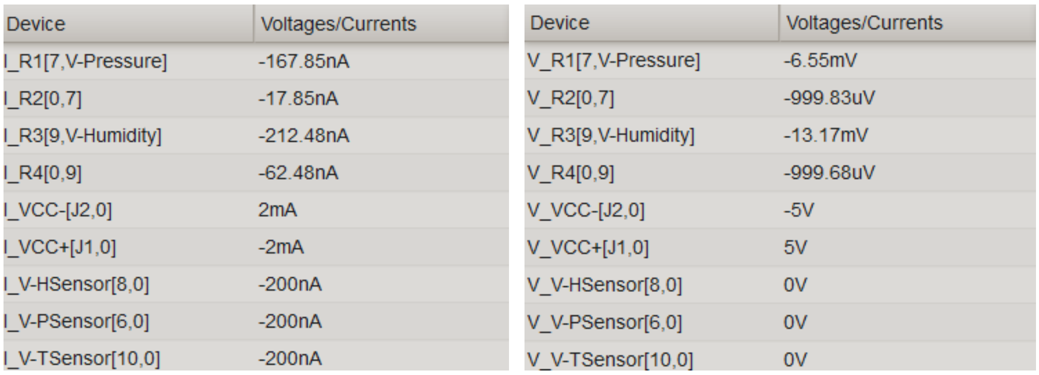


Figure 9.7. ADC Verification Test Plots

Lastly, the table below lists the results of the DC analysis on the supply and output nodes. These values are used in determining the power consumption of the data processing module.



Device	Voltages/Currents	Device	Voltages/Currents
I_R1[7,V-Pressure]	-167.85nA	V_R1[7,V-Pressure]	-6.55mV
I_R2[0,7]	-17.85nA	V_R2[0,7]	-999.83uV
I_R3[9,V-Humidity]	-212.48nA	V_R3[9,V-Humidity]	-13.17mV
I_R4[0,9]	-62.48nA	V_R4[0,9]	-999.68uV
I_VCC-[J2,0]	2mA	V_VCC-[J2,0]	-5V
I_VCC+[J1,0]	-2mA	V_VCC+[J1,0]	5V
I_V-HSensor[8,0]	-200nA	V_V-HSensor[8,0]	0V
I_V-PSensor[6,0]	-200nA	V_V-PSensor[6,0]	0V
I_V-TSensor[10,0]	-200nA	V_V-TSensor[10,0]	0V

Figure 9.8. DC Analysis Parameters

MCU Integration

The combined simulation of the schematic diagram seen in Figure 8.5 was also obtained, and the output transient plots are observed in the succeeding plots shown in Figures 9.9.a-c:

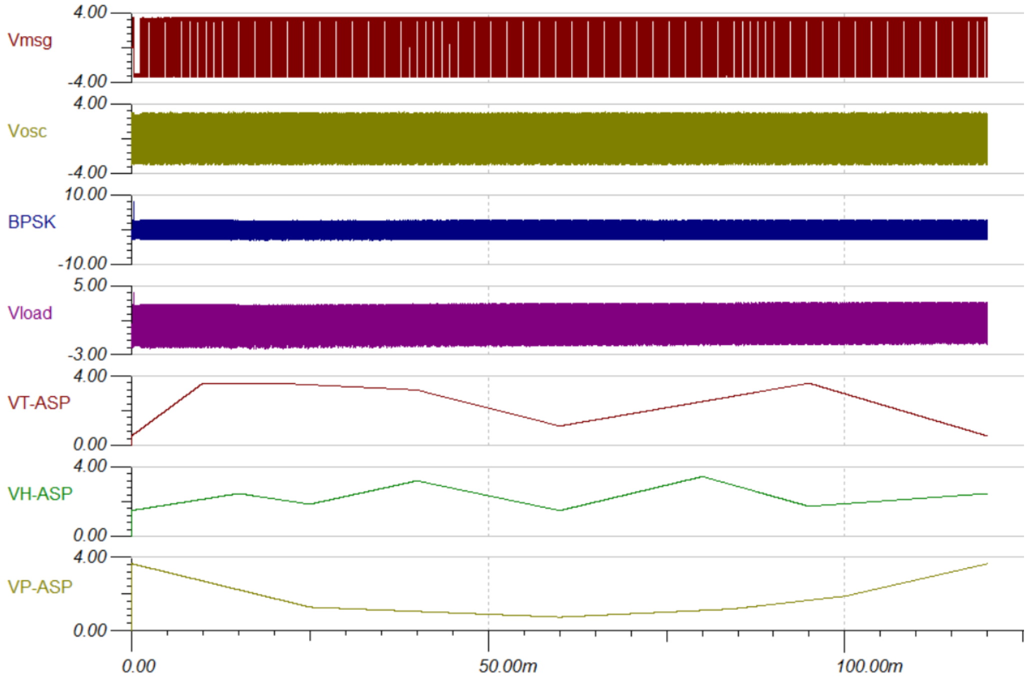


Figure 9.9.a. MCU-Integrated Transient Plots (0-120ms)

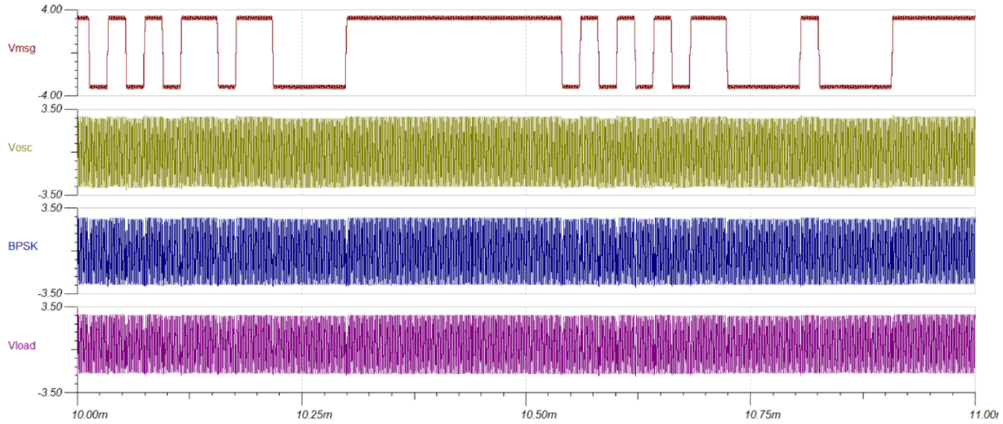
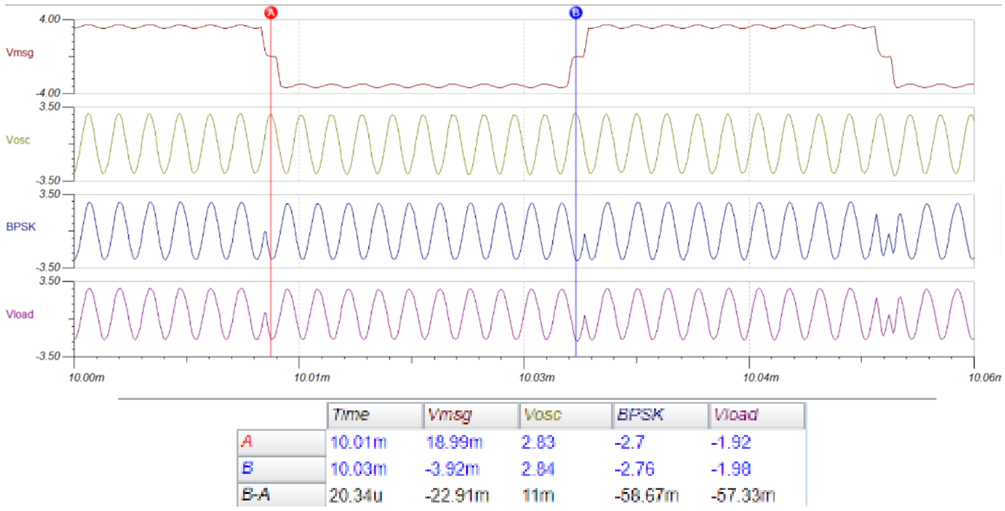


Figure 9.9.b. MCU-Integrated Transient Plots (10-11ms)



	Time	Vmsg	Vosc	BPSK	Vload
A	10.01m	18.99m	2.83	-2.7	-1.92
B	10.03m	-3.92m	2.84	-2.76	-1.98
B-A	20.34u	-22.91m	11m	-58.67m	-57.33m

Figure 9.9.c. MCU-Integrated Transient Plots (10.00-10.06ms)

X. Analysis

Power Module

The power module is composed of two switched-mode power supply (SMPS), the SEPIC and the Ćuk converter. A feedback system is also implemented to control the output voltage of the converters. The past chapters show that the designed converters managed to output the desired values for the system.

One limitation of the Infineon Designer is that it cannot output two PWM voltages at the same time. Due to this limitation, the PWM module is simulated separately with their respective microcontrollers.

To verify that the converters are working properly, the initial part of the data processing module is used to see if it can output the desired response. Note that in this simulation, since the converters are separated, the other voltage is set using a general voltage source.

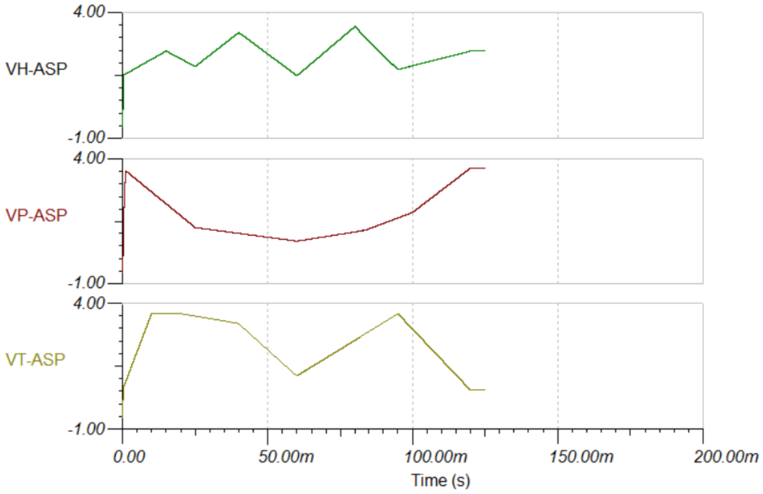


Figure 10.1. ASP Block Outputs using the Designed SEPIC Converter

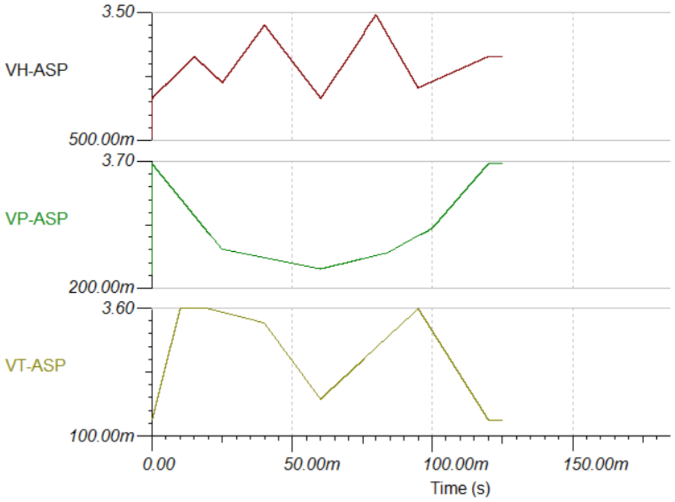


Figure 10.2. ASP Block Outputs using the Designed Ćuk Converter

As seen in Figures 10.1 and 10.2, the obtained graphs and values match the desired output for this stage of the data processing module. With this, it is concluded that the designed power module system was able to output the correct voltages and currents needed for the system.

Due to limitations of Infineon Designer, the whole system cannot be tested. Therefore, the verification of the power module is based on the output of the graphs above.

Communications Module

Upon inspection of Figure 9.9.b, it was verified that the bitstream output resembles the expected form of the message signal. Although there were small oscillations observed from V_{msg}, the modulation scheme was still proven to be operational. Moreover, upon measuring the message signal as seen in Figure 9.9.c, the waveform attained an approximate bit rate of $20.3\ \mu\mathrm{s}$. This translates to a message frequency of $49\ \mathrm{kHz}$, which is close to the design specification of $50\ \mathrm{kHz}$.

Also, the power transmitted to the load was determined to be approximately $55.8\ \mathrm{mW}$. Detailed computations can be found under Appendix F-2C. This exceeds the minimum transmitted power required as stated by the Link Margin.

Data Processing Module

Using the provided atmospheric measurements, the team was able to produce equivalent voltage plots using linear piecewise voltage generators shown in Figure 9.5. For the pressure and humidity sensors, a proportional relationship was seen between the true value and the converted voltage level since the conversion formula is linear. Meanwhile, an inversely proportional relationship is seen on the thermistor because the conversion involves a decaying exponential.

Sensor	Minimum Value	Maximum Value	Vmin (V)	Vmax (V)
Pressure	0 kPa	101.33 kPa	0.5000	3.6265
Humidity	0%	100%	0.3000	0.9000
Temperature	-15 °C	29 °C	0.4819	4.4152

Table 10.1. Linear Piecewise Voltage Generator Output Extremes

For the ASP block output seen in Figure 9.6, all sensors nearly reached the maximum output of 3.7 V. Note that the gain is not optimal since the reference voltage is still 5 V. However, this is justifiable since op amp models are limited in Infineon Designer. It is recommended to use an op amp that can produce a higher voltage level.

ASP Block	Max Sensor Output (V)	Optimal Gain (Vref / Vmax)	Gain	Max ASP Block Output (V)
Pressure	2.5265	1.9790	1.4354	3.6265
Humidity	0.9	5.5556	4.0909	3.6818
Temperature	4.4152	1.1423	0.8209	3.6244

Table 10.2. Summary of Computations on ASP Block Gain

Tests done were then verified to verify if the correct value is read by the ADC. Originally, a program that converts ADC readings to binary and produces an output bitstream was used. A demo of this process is seen below. However, this proved to be a tedious task since adjacent 1s or 0s are difficult to distinguish from the plot and there are 360 data points to compare.

A simpler program with a similar function to a comparator is then used as shown in Figure 9.7. The plot behaves as expected such that a high output is seen when the read value is greater than 2.5V and a low output is seen otherwise. The response is also fast – that is, the output immediately switches after encountering the 2.5 V threshold which is significant since the simulation is operating in milliseconds of time.

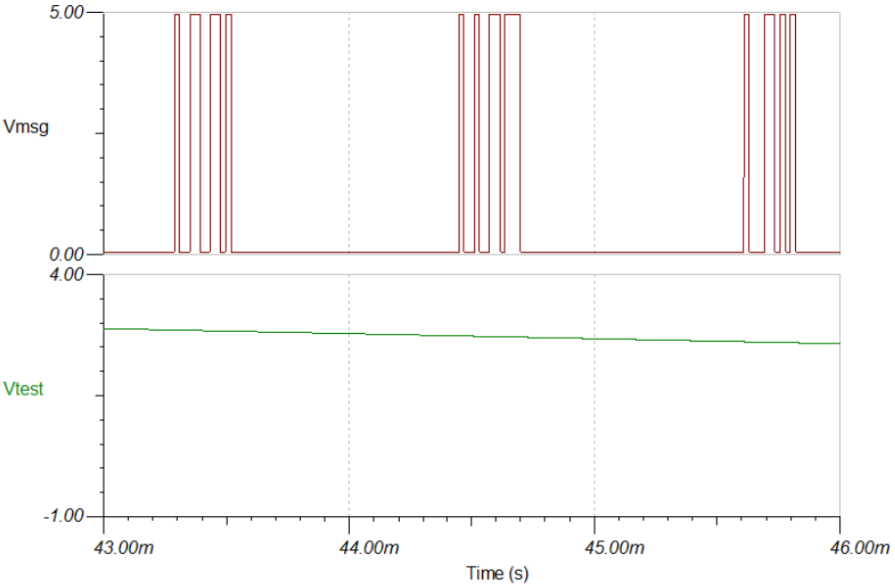


Figure 10.3. Comparison Test of Output Bitstream and Voltage Level

XI. Conclusion

The primary mission of the Can Satellite was simulated using Infineon Designer. With the data analysis performed on the different CanSat subsystems, the team has formulated the following conclusions:

With the computed Remaining Power Budget being 60 percent of the battery capacity, the power supply is then sufficient for the 150-minute operation of the CanSat. Moreover, the Power Management Unit sufficiently provides the required supply voltages and currents for the Transmitter and the Data Processing Module, as shown in the transient response of the switching converters.

With regards to the communications link, the simulated bitstream adheres to the expected format with a bit rate approximately close to the target frequency. The transmitter circuit also implements the Binary Phase Shift Keying scheme with the chosen circuit components used in simulations. Power computations also verify that the minimum transmitted power is attained such that reliable transmission is attainable despite path loss.

Sensors were modelled as linear piecewise voltage generators with the provided data points. After applying op-amp gain blocks, the desired sensor profile was achieved. The XMC1100 configurations were also able to perform A-D conversion on the analog sensor data obtained from the gain-adjusted sensor models.

The secondary mission was discussed along with its implied modifications on resource allocations. With the adjusted computations on the original system, CanSat operation was ensured according to the memory and power budget.

The project objectives have been met upon completion of the simulated CanSat system. Therefore, we conclude that the design and project specifications have been achieved.

Bibliography

[1] B. Dunbar, “What Is a Satellite?”, NASA, 16-Jun-2015. [Online]. Available: https://www.nasa.gov/audience/forstudents/5-8/features/nasa-knows/what-is-a-satellite-58.html. [Accessed: June 28, 2021]

[2] “Satellites are Getting Smaller, Cheaper, and More Modular,” Tech Times, 30-Dec-2019. [Online]. Available: https://www.techtimes.com/articles/246704/20191230/satellites-are-getting-smaller-cheaper-and-more-modular.htm. [Accessed: June 28, 2021]

[3] “What is a CanSat?”, The European Space Agency. [Online]. Available: http://www.esa.int/Education/CanSat/What_is_a_CanSat. [Accessed: June 28, 2021]

[4] F. Bagheroskouei, S. Karbasian, M. Baghban and R. Amjadifard, "Design and implementation of the electrical power subsystem for a small satellite," *2017 8th International Conference on Recent Advances in Space Technologies (RAST)*, Istanbul, Turkey, 2017, pp. 209-214, doi: 10.1109/RAST.2017.8003000.

[5] R. Linhart, A. Voborník and I. Veŕtát, "Communication Subsystem for PilsenCUBE Nanosatellite," 2018 International Conference on Applied Electronics (AE), Pilsen, Czech Republic, 2018, pp. 1-4, doi: 10.23919/AE.2018.8501441.

[6] R. K. D. Aranas et al., "Design of a Free and Open Source Data Processing, Archiving, and Distribution Subsystem for the Ground Receiving Station of the Philippine Scientific Earth Observation Micro-satellite". ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLI-B7, 909–911, https://doi.org/10.5194/isprs-archives-XLI-B7-909-2016, 2016.

[7] “Choose your mission,” *The European Space Agency*. [Online]. Available: https://www.esa.int/Education/CanSat/Choose_your_mission. [Accessed: June 28, 2021]

[8] IUPAC. Compendium of Chemical Terminology, 2nd ed. (the "Gold Book"). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford (1997). Online version (2019-) created by S. J. Chalk. ISBN 0-9678550-9-8. https://doi.org/10.1351/goldbook. [Accessed: April 18, 2021]

[9] Electrical and Electronics Engineering Institute. University of the Philippines, Diliman. (2021) Challenge Lab Specifications

[10] U.S. National Weather Service. (n.d.) Layers of the Atmosphere. [Online]. Available: https://www.weather.gov/jetstream/layers [Accessed: April 17, 2021]

[11] Department of Science and Technology, Philippines. (2021) Climate of the Philippines. [Online]. Available: http://bagong.pagasa.dost.gov.ph/information/climate-philippines [Accessed: April 17, 2021]

[12] Wright, Mark (2003). “Sampling rates for analog sensors”, Embedded Website [Online]. Available: https://www.embedded.com/sampling-rates-for-analog-sensors/ [Accessed: April 20, 2021]

[13] G. Sharp, "Sepic Converter Design and Operation", May 1, 2014. p. 6. [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-050114-131841/unrestricted/SEPIC_MQP_Final_Report.pdf

[14] Ramanath, A. (2020, August 3). What is a Ćuk Converter? EEPower. https://eepower.com/technical-articles/intro-to-cuk-converters-part-1/#.

[15] Sedra, A. S., & Smith, K. C. (2015). Class AB Output Stage. In *Microelectronic circuits* (pp. 935–937). Oxford University Press.

[16] “LM709 Datasheet,” Texas Instruments. Dallas, Texas, United States.

[17] “XMC1000 Family Datasheet,” Infineon Technologies AG, Neubiberg, Germany.

[18] “BMP180 Datasheet,” Bosch Sensortec. Kusterdingen, Germany.

[19] “Si7020 Datasheet,” Silicon Labs. Austin, Texas, United States.

[20] Sedra, A. S., & Smith, K. C. (2015). Operation of the Astable Multivibrator. In *Microelectronic circuits* (pp. 1413–1415). Oxford University Press.

[21] Albaugh, N. P. (2006). “500 Khz Sine and Square Wave Oscillator”. Youspice [Online]. Available: https://www.youspice.com/spiceprojects/spice-simulation-projects/signal-generation-circuits-spice-simulation-projects/500-khz-sine-and-square-wave-oscillator/ [Accessed: May 19, 2021]

[22] “BC850C Datasheet,” Siemens Semiconductor Group. Munich, Germany.

[23] “BC860C Datasheet,” Siemens Semiconductor Group. Munich, Germany.

[24] “MT3329 Datasheet,” MediaTek. Hsinchu, Taiwan

[25] “XENSIV PAS CO2 Datasheet,” Infineon Technologies AG, Neubiberg, Germany.

[26] National Marine Electronics Association (2018). “NMEA Sentences for GPS Receivers” [Online]. Available: https://www.nmea.org/content/STANDARDS/NMEA_0183_Standard [Accessed: May 24, 2021]

[27] D. W. Hart, “DC-DC Converters,” in Power electronics, Sidney: McGraw Hill, 2010. [Accessed: May 30, 2021]

Appendices

Appendix A

Link Budget Analysis

For computations of the Link Budget, the Received Power P_{Rx} can be related to the Transmitted Power P_{Tx} with corresponding gains g_{Rx}, g_{Tx} and the path loss using the following equation:

$$P_{Rx} = P_{Tx} + g_{Tx} - FSPL + g_{Rx}\,, \qquad (\mathrm{A.1})$$

where the Free-Space Path Loss is given by

$$FSPL_{dB} = 32.45 + 20\log_{10} f_{MHz} + 20\log_{10} D_{km}.$$

Provided that the carrier frequency $f_c = 500$ kHz and the maximum distance from the ground station $D = 50$ km, the maximum Free-Space Path Loss would be:

$$FSPL_{dB} = 32.45 + 20\log_{10}(0.5) + 20\log_{10}(50) = \; 60.4088 \;\; dB$$

Furthermore, assuming transceiver gains to be $g_{Tx} = g_{Rx} = 0\ \mathrm{dB}$, equation $(\mathrm{A.1})$ can be simplified to

$$P_{Rx} = P_{Tx} - 60.4088\ \mathrm{dB} \qquad (\mathrm{A.2})$$

We then define the Link Margin LM as the difference between the Received Power and the Receiver Sensitivity $R_{X,S}$:

$$LM = P_{Rx} - R_{X,S}$$

For reliable data transmission, *LM* must be at least 7 dB. Given from the design specifications that $R_{X,S} = -50\ \mathrm{dBm} = \; -80\ \mathrm{dB}$, the resulting inequality follows:

$$\begin{gathered} 7\ \mathrm{dB} \leq P_{Rx} + 80\ \mathrm{dB} \\ P_{Rx} \geq -73\ \mathrm{dB} \qquad (\mathrm{A.3}) \end{gathered}$$

Hence, combining equations $(\mathrm{A.2})$ and $(\mathrm{A.3})$ the minimum P_{Tx} can be computed:

$$P_{Tx} - 60.4088\ \mathrm{dB} \geq \; -73\ \mathrm{dB}$$

$$P_{Tx,min} = -12.5912\ \mathrm{dB}$$

$$\therefore P_{Tx,min} = \mathbf{55.0656\ mW}$$

Therefore, the communications module must deliver at least 55.07 mW of power to the antenna to establish a reliable communication link.

Appendix B

Data Transfer and Storage

A. Packet Information

[PACKET START] 1111 1110 – dataH – dataP – dataT – P – 0001 [PACKET END]

Each packet sent by the communications module will apply the above format. The last 4 bits and the first 8 bits are predefined and constant. Data bits will contain sensor data and one parity bit. Lastly, the P bit is a check bit set at even parity. In total, each packet will contain 52 bits.

B. Stored Data in Memory

[WORD1 START] dataH – dataP – 00 0000 [WORD1 END]
[WORD2 START] dataT – 000 0000 0000 0000 0000 [WORD2 END]

The two bytes stored each minute will have the format above. The humidity and pressure measurements and its corresponding parity bits are stored in the first word. Six more bits are appended to complete the word. Meanwhile, the temperature data including the parity bit is stored in the second word. The next 19 bits are set to 0 since it is unused memory.

C. Secondary Mission Memory Budget

[WORD1 START] dataH – dataP – 00 0000 [WORD1 END]
[WORD2 START] dataT – dataC – 00 0000 [WORD2 END]
[WORD2 START] dataG1 – dataG2 – 00 0000 [WORD2 END]

The three bytes stored each minute will have the format above. The humidity and pressure measurements and its corresponding parity bits are stored in the first word. Six more bits are appended to complete the word. The same procedure is repeated for the temperature and CO2 sensor data, and the two values from the GPS module.

D. Secondary Mission Packet Information

[PACKET START] 1111 1110 –
dataH – dataP – dataT – dataC – dataG1 – dataG2 –
P – 0001 [PACKET END]

With the inclusion of secondary mission peripherals, sensor readings are appended to the data bits, preserving the format of the predefined bits. Assuming the same bit length for each sensor (with two readings from the GPS module), each packet will contain 88 bits.

Appendix C

Computations on Sensors and Sampling

A. Sensor Reading to Voltage Conversion

Two-Point Formula:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Pressure Sensor:

$$V - 0.5 = \frac{5 - 0.5}{225 - 0}(P - 0)$$

$$V - 0.5 = \frac{4.5}{225}P$$

$$V = 0.02P + 0.5$$

Humidity Sensor:

$$V - 0.3 = \frac{0.9 - 0.3}{100 - 0}(H - 0)$$

$$V - 0.3 = \frac{0.6}{100}H$$

$$V = 0.006H + 0.3$$

B. Derivation of Resistor Value in the Temperature Sensor [9]

Model #	Zero Power Resistance at 25°C	Beta (K)	Voltage Reading at 0°C
55031	10000	3574	1.362167942

$$R_{thermistor} = R_0 e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

$$R_{thermistor} = 10^4 \times e^{3574\left(\frac{1}{T} - \frac{1}{298.15}\right)}$$

$$R_{T=0\mathrm{C}} = 10^4 \times e^{3574\left(\frac{1}{273.15} - \frac{1}{298.15}\right)}$$

$$= 29955.59816$$

$$V_{T=0\mathrm{C}} = V_{DD}\frac{R_{T=0\mathrm{C}}}{R_{T=0\mathrm{C}} + R_2}$$

$$1.362167942 = 5 \times \frac{29955.59816}{29955.59816 + R_2}$$

$$\therefore R_2 = 80\ \mathrm{k\Omega}$$

C. Conversion from Parameter Ranges to Voltage Ranges

Pressure Sensor:

$$P_{min} = 0$$

$$V_{min} = 0.02P_{min} + 0.5$$

$$= 0.5$$

$$P_{max} = 101.325$$

$$V_{max} = 0.02(101.325) + 0.5$$

$$= 2.5265$$

Humidity Sensor:

$$H_{min} = 0$$

$$V_{min} = 0.006H_{min} + 0.3$$

$$= 0.3$$

$$H_{max} = 100$$

$$V_{max} = 0.006(100) + 0.3$$

$$= 0.9$$

Temperature Sensor:

$$T_{min} = -51$$

$$R_{min} = 10^4 \times e^{3574\left(\frac{1}{222.15} - \frac{1}{298.15}\right)}$$

$$V_{min} = \frac{R_{min}}{R_{min} + 80\text{k}} = 0.4818$$

$$T_{max} = 29$$

$$R_{max} = 10^4 \times e^{3574\left(\frac{1}{302.15} - \frac{1}{298.15}\right)}$$

$$V_{max} = \frac{R_{max}}{R_{max} + 80\text{k}} = 4.4152$$

D. Sensor Data

The table below shows the given data measurements for each sensor. For the pressure and humidity sensors, CanSat measurements are directly translated to voltage levels. The temperature parameter is first converted to resistance before it is converted to voltage.

Time	Pressure (kPa)	Pressure (V)	Humidity (%)	Humidity (V)	Temperature (°C)	Temperature (Ω)	Temperature (V)
0	101.3250	2.5265	10.0000	0.3600	20.0000	12268.5746	0.6648
1	98.3809	2.4676	12.6667	0.3760	13.0000	16531.7567	0.8563
2	95.4369	2.4087	15.3333	0.3920	6.0000	22612.0462	1.1018
3	92.4928	2.3499	18.0000	0.4080	-1.0000	31430.9812	1.4103
4	89.5488	2.2910	20.6667	0.4240	-8.0000	44455.6958	1.7860
5	86.6047	2.2321	23.3333	0.4400	-15.0000	64071.1581	2.2236
6	83.6607	2.1732	26.0000	0.4560	-22.0000	94242.3705	2.7043
7	80.7166	2.1143	28.6667	0.4720	-29.0000	141722.6864	3.1959
8	77.7726	2.0555	31.3333	0.4880	-36.0000	218319.7559	3.6592
9	74.3160	1.9863	34.0000	0.5040	-43.0000	345272.2305	4.0594
10	70.8595	1.9172	36.6667	0.5200	-50.0000	561978.1677	4.3769
11	67.4029	1.8481	39.3333	0.5360	-50.0000	561978.1677	4.3769
12	63.9464	1.7789	42.0000	0.5520	-50.0000	561978.1677	4.3769
13	60.4898	1.7098	44.6667	0.5680	-50.0000	561978.1677	4.3769
14	57.0332	1.6407	47.3333	0.5840	-50.0000	561978.1677	4.3769
15	53.5767	1.5715	50.0000	0.6000	-50.0000	561978.1677	4.3769
16	50.1201	1.5024	47.5000	0.5850	-50.0000	561978.1677	4.3769
17	46.6636	1.4333	45.0000	0.5700	-50.0000	561978.1677	4.3769
18	43.2070	1.3641	42.5000	0.5550	-50.0000	561978.1677	4.3769
19	39.7504	1.2950	40.0000	0.5400	-50.0000	561978.1677	4.3769
20	36.2939	1.2259	37.5000	0.5250	-50.0000	561978.1677	4.3769
21	32.8373	1.1567	35.0000	0.5100	-49.5000	542211.8316	4.3571
22	29.3808	1.0876	32.5000	0.4950	-49.0000	523224.3069	4.3369
23	25.9242	1.0185	30.0000	0.4800	-48.5000	504981.8217	4.3162
24	22.4676	0.9494	27.5000	0.4650	-48.0000	487452.1956	4.2951
25	19.0111	0.8802	25.0000	0.4500	-47.5000	470604.7593	4.2735
26	15.5545	0.8111	28.6668	0.4720	-47.0000	454410.2780	4.2515
27	15.1044	0.8021	32.3334	0.4940	-46.5000	438840.8792	4.2291
28	14.6543	0.7931	36.0001	0.5160	-46.0000	423869.9843	4.2061
29	14.2042	0.7841	39.6668	0.5380	-45.5000	409472.2441	4.1828
30	13.7541	0.7751	43.3334	0.5600	-45.0000	395623.4771	4.1590
31	13.3040	0.7661	47.0001	0.5820	-44.5000	382300.6120	4.1348
32	12.8539	0.7571	50.6668	0.6040	-44.0000	369481.6323	4.1101
33	12.4038	0.7481	54.3334	0.6260	-43.5000	357145.5250	4.0850
34	11.9537	0.7391	58.0001	0.6480	-43.0000	345272.2305	4.0594
35	11.5036	0.7301	61.6668	0.6700	-42.5000	333842.5965	4.0334
36	11.0535	0.7211	65.3335	0.6920	-42.0000	322838.3339	4.0070
37	10.6034	0.7121	69.0001	0.7140	-41.5000	312241.9741	3.9802
38	10.1533	0.7031	72.6668	0.7360	-41.0000	302036.8302	3.9530
39	9.7032	0.6941	76.3335	0.7580	-40.5000	292206.9585	3.9253
40	9.2531	0.6851	80.0000	0.7800	-40.0000	282737.1236	3.8973
41	8.8030	0.6761	76.5000	0.7590	-38.0000	248176.3609	3.7811
42	8.3529	0.6671	73.0000	0.7380	-36.0000	218319.7559	3.6592

Time	Pressure (kPa)	Pressure (V)	Humidity (%)	Humidity (V)	Temperature (°C)	Temperature (Ω)	Temperature (V)
43	7.9028	0.6581	69.5000	0.7170	-34.0000	192467.2092	3.5319
44	7.4527	0.6491	66.0000	0.6960	-32.0000	170031.1070	3.4002
45	7.0026	0.6401	62.5000	0.6750	-30.0000	150516.9926	3.2648
46	6.5525	0.6311	59.0000	0.6540	-28.0000	133507.7654	3.1265
47	6.1024	0.6220	55.5000	0.6330	-26.0000	118650.7244	2.9864
48	5.6523	0.6130	52.0000	0.6120	-24.0000	105646.9190	2.8454
49	5.2022	0.6040	48.5000	0.5910	-22.0000	94242.3705	2.7043
50	4.7521	0.5950	45.0000	0.5700	-20.0000	84220.8190	2.5643
51	4.3020	0.5860	41.5000	0.5490	-18.0000	75397.7139	2.4260
52	3.8519	0.5770	38.0000	0.5280	-16.0000	67615.2247	2.2903
53	3.4019	0.5680	34.5000	0.5070	-14.0000	60738.0861	2.1578
54	2.9518	0.5590	31.0000	0.4860	-12.0000	54650.1329	2.0293
55	2.5017	0.5500	27.5000	0.4650	-10.0000	49251.4005	1.9053
56	2.0516	0.5410	24.0000	0.4440	-8.0000	44455.6958	1.7860
57	1.6015	0.5320	20.5000	0.4230	-6.0000	40188.5558	1.6719
58	1.1514	0.5230	17.0000	0.4020	-4.0000	36385.5294	1.5631
59	0.7013	0.5140	13.5000	0.3810	-2.0000	32990.7275	1.4599
60	0.2512	0.5050	10.0000	0.3600	0.0000	29955.5982	1.3622
61	0.8888	0.5178	14.0000	0.3840	-1.5000	32199.9768	1.4349
62	1.5264	0.5305	18.0000	0.4080	-3.0000	34640.2938	1.5108
63	2.1641	0.5433	22.0000	0.4320	-4.5000	37295.9658	1.5898
64	2.8017	0.5560	26.0000	0.4560	-6.0000	40188.5558	1.6719
65	3.4394	0.5688	30.0000	0.4800	-7.5000	43342.0346	1.7570
66	4.0770	0.5815	34.0000	0.5040	-9.0000	46783.0770	1.8450
67	4.7146	0.5943	38.0000	0.5280	-10.5000	50541.3977	1.9358
68	5.3523	0.6070	42.0000	0.5520	-12.0000	54650.1329	2.0293
69	5.9899	0.6198	46.0000	0.5760	-13.5000	59146.2732	2.1253
70	6.6276	0.6326	50.0000	0.6000	-15.0000	64071.1581	2.2236
71	7.2652	0.6453	54.0000	0.6240	-16.5000	69471.0375	2.3239
72	7.9028	0.6581	58.0000	0.6480	-18.0000	75397.7139	2.4260
73	8.5405	0.6708	62.0000	0.6720	-19.5000	81909.2756	2.5295
74	9.1781	0.6836	66.0000	0.6960	-21.0000	89070.9348	2.6341
75	9.8158	0.6963	70.0000	0.7200	-22.5000	96955.9882	2.7396
76	10.4534	0.7091	74.0000	0.7440	-24.0000	105646.9190	2.8454
77	11.0910	0.7218	78.0000	0.7680	-25.5000	115236.6611	2.9512
78	11.7287	0.7346	82.0000	0.7920	-27.0000	125830.0547	3.0566
79	12.3663	0.7473	86.0000	0.8160	-28.5000	137545.5206	3.1613
80	13.0040	0.7601	90.0000	0.8400	-30.0000	150516.9926	3.2648
81	13.6416	0.7728	85.3331	0.8120	-31.3306	163196.5251	3.3552
82	14.2792	0.7856	80.6664	0.7840	-32.6639	177132.1668	3.4444
83	14.9169	0.7983	75.9997	0.7560	-33.9972	192433.5360	3.5318
84	15.5545	0.8111	71.3331	0.7280	-35.3305	209251.0064	3.6171
85	17.2828	0.8457	66.6664	0.7000	-36.6638	227753.2826	3.7003
86	19.0111	0.8802	61.9997	0.6720	-37.9971	248129.8476	3.7810
87	20.7394	0.9148	57.3330	0.6440	-39.3304	270593.7644	3.8591
88	22.4676	0.9494	52.6664	0.6160	-40.6637	295384.8886	3.9344
89	24.1959	0.9839	47.9997	0.5880	-41.9970	322773.5564	4.0069
90	25.9242	1.0185	43.3330	0.5600	-43.3303	353064.8227	4.0764
91	27.6525	1.0530	38.6664	0.5320	-44.6636	386603.3396	4.1427
92	29.3808	1.0876	33.9997	0.5040	-45.9969	423778.9782	4.2060

Time	Pressure (kPa)	Pressure (V)	Humidity (%)	Humidity (V)	Temperature (°C)	Temperature (Ω)	Temperature (V)
93	31.1090	1.1222	29.3330	0.4760	-47.3302	465033.3170	4.2661
94	32.8373	1.1567	24.6664	0.4480	-48.6635	510867.1398	4.3230
95	34.5656	1.1913	20.0000	0.4200	-50.0000	561978.1677	4.3769
96	36.2939	1.2259	21.2000	0.4272	-47.2000	460811.5294	4.2604
97	38.0222	1.2604	22.4000	0.4344	-44.4000	379697.1918	4.1299
98	39.7504	1.2950	23.6000	0.4416	-41.6000	314329.4257	3.9856
99	41.4787	1.3296	24.8000	0.4488	-38.8000	261392.6709	3.8283
100	43.2070	1.3641	26.0000	0.4560	-36.0000	218319.7559	3.6592
101	44.9353	1.3987	27.2000	0.4632	-33.2000	183112.3867	3.4797
102	46.6636	1.4333	28.4000	0.4704	-30.4000	154207.0793	3.2921
103	49.7003	1.4940	29.6000	0.4776	-27.6000	130374.4620	3.0986
104	52.7370	1.5547	30.8000	0.4848	-24.8000	110643.2240	2.9018
105	55.7738	1.6155	32.0000	0.4920	-22.0000	94242.3705	2.7043
106	58.8105	1.6762	33.2000	0.4992	-19.2000	80557.1522	2.5087
107	61.8473	1.7369	34.4000	0.5064	-16.4000	69095.2612	2.3172
108	64.8840	1.7977	35.6000	0.5136	-13.6000	59460.7761	2.1318
109	67.9208	1.8584	36.8000	0.5208	-10.8000	51333.9864	1.9543
110	70.9575	1.9192	38.0000	0.5280	-8.0000	44455.6958	1.7860
111	73.9943	1.9799	39.2000	0.5352	-5.2000	38614.9590	1.6277
112	77.0310	2.0406	40.4000	0.5424	-2.4000	33639.4572	1.4801
113	80.0678	2.1014	41.6000	0.5496	0.4000	29387.9142	1.3433
114	83.1045	2.1621	42.8000	0.5568	3.2000	25744.0963	1.2173
115	86.1413	2.2228	44.0000	0.5640	6.0000	22612.0462	1.1018
116	89.1780	2.2836	45.2000	0.5712	8.8000	19912.2822	0.9965
117	92.2148	2.3443	46.4000	0.5784	11.6000	17578.7572	0.9007
118	95.2515	2.4050	47.6000	0.5856	14.4000	15556.4153	0.8140
119	98.2883	2.4658	48.8000	0.5928	17.2000	13799.2226	0.7356
120	101.3250	2.5265	50.0000	0.6000	20.0000	12268.5746	0.6648

Appendix D

Sensors and Sampling Scripts

Python code used to determine the maximum slope on given CanSat measurement plots:

```
def max_slope(x, y):
    m =0
    for (r,i) in enumerate(x):
        try:
            temp = ((y[i]-y[i+1])/(r-x[i+1]))
        except IndexError as e:
            pass
        if (m < abs(temp)):
            m = abs(temp)
    return m

time_p = [0,25,60,84,100,120]
data_p = [101.325,19.01108,0.25116,15.55452,43.207,101.325]

time_h = [0,15,25,40,60,80,95,120]
data_h = [10,50,25,80,10,90,20,50]

time_t = [0,10,20,40,60,95,120]
data_t = [20,-50,-50,-40,0,-50,20]

r_p = 4096*(max_slope(time_p,data_p)/101.325)
r_h = 4096*(max_slope(time_h, data_h)/100)
r_t = 4096*(max_slope(time_t,data_t)/80)
print(min(r_p, r_h, r_t))
```

Python code used to determine data points used in Matlab plotting:

```
time = [] # insert sensor time here
data = [] # insert sensor data here

def find_turning_points(x, y):
    t_points_x = []
    t_points_y = []
    slope = float('inf')
    for (i,r) in enumerate(x):
        try:
            temp = ((y[i]-y[i+1])/(r-x[i+1]))
        except IndexError as e:
            temp = float('inf')
            pass
        if (slope != round(temp,2)):
            slope = round(temp,2)
            t_points_x.append(r)
            t_points_y.append(y[i])
    return (t_points_x, t_points_y)

time_x, data_x = find_turning_points(time,data)
print(time_x, data_x)
```

Matlab code used in plotting CanSat sensor measurements:

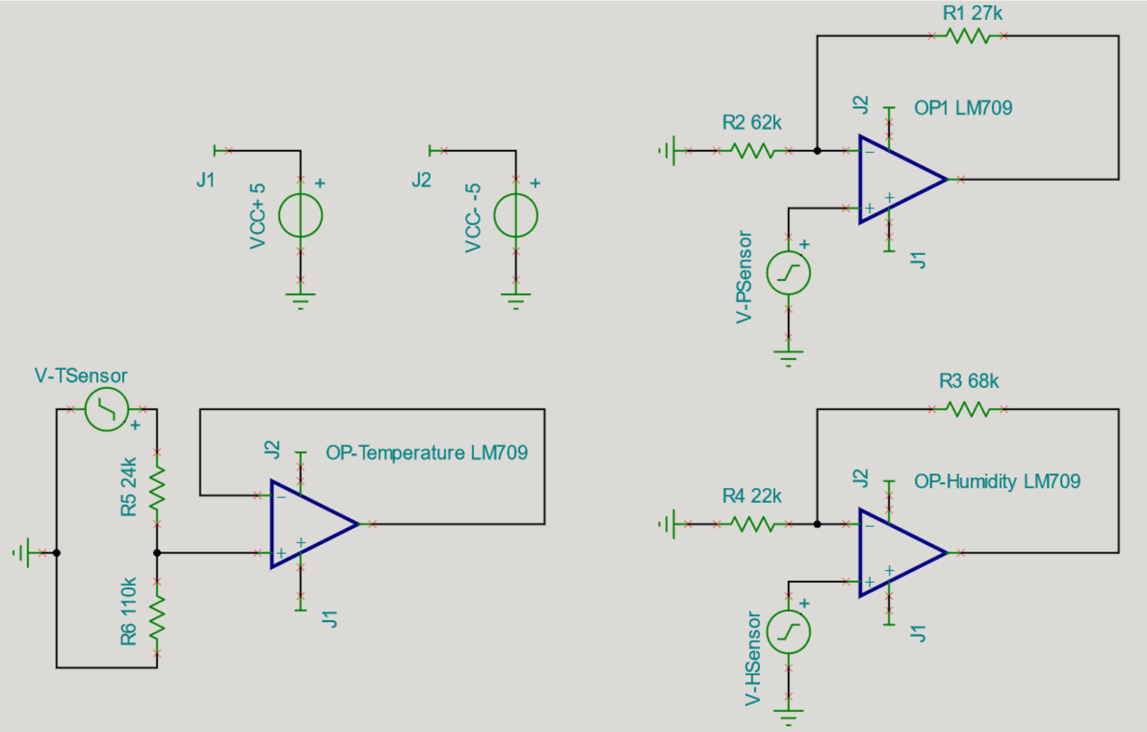
```
% Pressure

figure;
sgtitle("CanSat Pressure Measurements");
time_p=[0,25,60,84,100,120];
data_p=[101.325,19.01108,0.25116,15.55452,43.207,101.325];
subplot(1,2,1)
plot(time_p,data_p,'-');
xlabel("Time(min)");
ylabel("Pressure(kPa)");
subplot(1,2,2)
volt_p = [2.5265,0.88022,0.505023,0.81109,1.295009,2.5265];
plot(time_p,volt_p,'-');
xlabel("Time(min)");
ylabel("Voltage(V)");

% Humidity

figure;
sgtitle("Humidity Sensor Reading");
time_h= [0,15,25,40,60,80,95,120];
data_h=[10,50,25,80,10,90,20,50];
subplot(1,2,1)
plot(time_h,data_h,'-');
xlabel("Time(min)");
ylabel("Relative Humidity(%)");
ylim([0 100]);
subplot(1,2,2)
volt_h = [0.36,0.6,0.45,0.78,0.36,0.84,0.42,0.6];
plot(time_h,volt_h,'-');
xlabel("Time(min)");
ylabel("Voltage(V)");
ylim([0.3,0.9]);

% Temperature

figure;
sgtitle("CanSat Temperature Measurements");
time_t= [0,10,20,40,60,95,120];
data_t=[20,-50,-50,-40,0,-50,20];
subplot(1,2,1)
plot(time_t,data_t,'-');
xlabel("Time(min)");
ylabel("Temperature(Celcius)");
ylim([-60 30]);
subplot(1,2,2)
volt_t = [0.66483,4.376926,4.376926,3.897273,1.362168,4.376926,0.66483];
plot(time_t,volt_t,'-');
xlabel("Time(min)");
ylabel("Voltage(V)");
ylim([0 5]);
```

Appendix E-1

Sensor Simulations

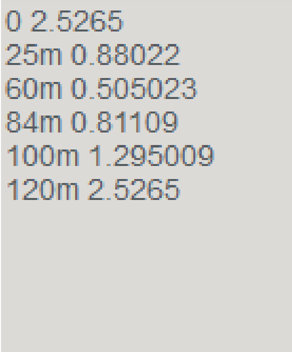
A. Schematic

Shown below is the Infineon Designer schematic of the data processing module. Amplifiers (LM709) use a ±5V dual voltage supply. Sensor output is fed into the non-inverting input.



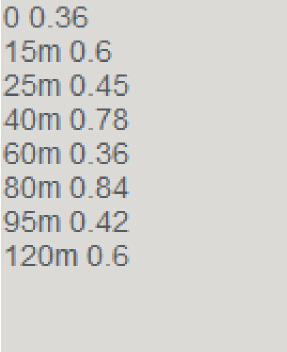
B. Linear Voltage Generator Configuration

Pressure Sensor:



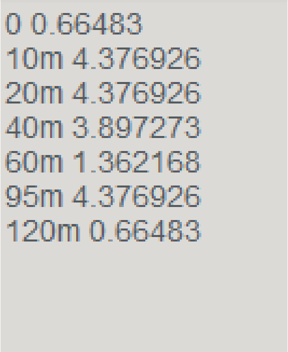
```
0 2.5265
25m 0.88022
60m 0.505023
84m 0.81109
100m 1.295009
120m 2.5265
```

Humidity Sensor:



```
0 0.36
15m 0.6
25m 0.45
40m 0.78
60m 0.36
80m 0.84
95m 0.42
120m 0.6
```

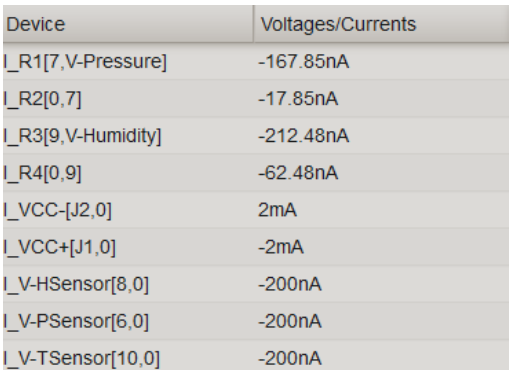
Temperature Sensor:



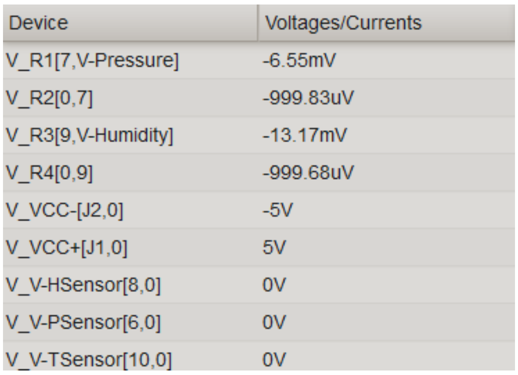
```
0 0.66483
10m 4.376926
20m 4.376926
40m 3.897273
60m 1.362168
95m 4.376926
120m 0.66483
```

C. DC Analysis Output

The table below are the results of the DC analysis on the supply and output nodes. These values are used in determining the power consumption of the data processing module.



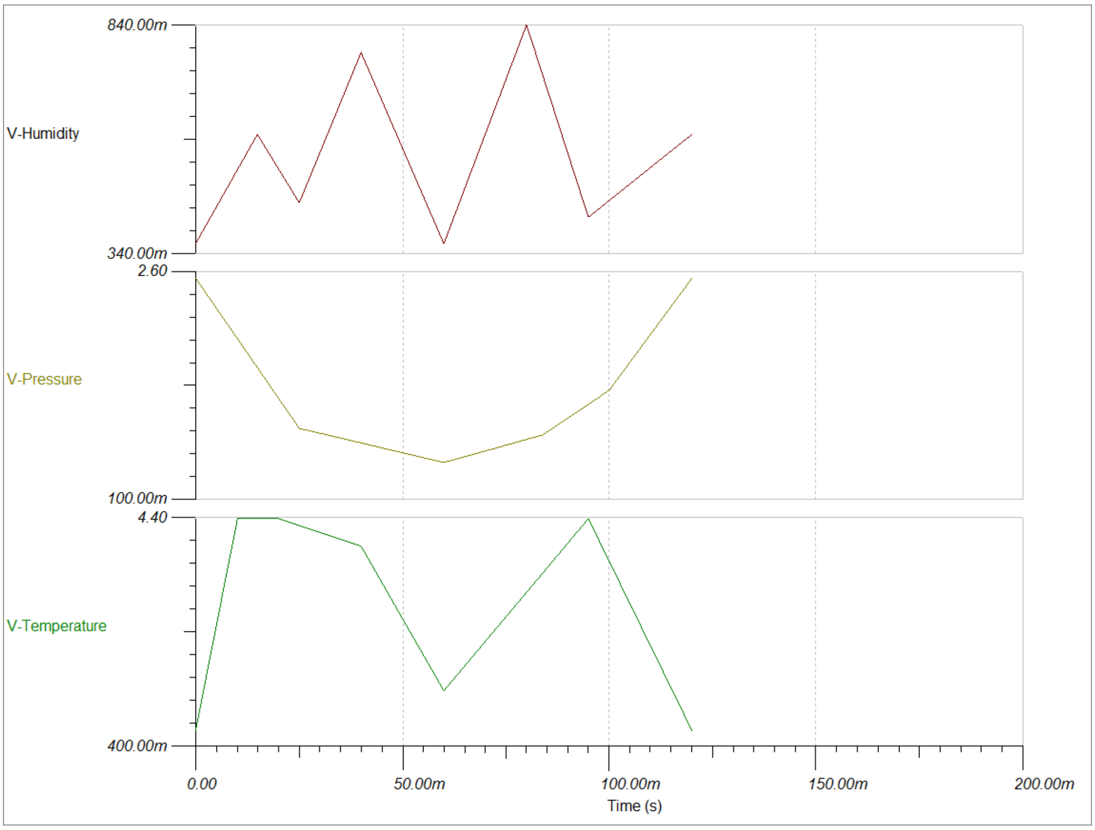
Device	Voltages/Currents
I_R1[7,V-Pressure]	-167.85nA
I_R2[0,7]	-17.85nA
I_R3[9,V-Humidity]	-212.48nA
I_R4[0,9]	-62.48nA
I_VCC-[J2,0]	2mA
I_VCC+[J1,0]	-2mA
I_V-HSensor[8,0]	-200nA
I_V-PSensor[6,0]	-200nA
I_V-TSensor[10,0]	-200nA



Device	Voltages/Currents
V_R1[7,V-Pressure]	-6.55mV
V_R2[0,7]	-999.83uV
V_R3[9,V-Humidity]	-13.17mV
V_R4[0,9]	-999.68uV
V_VCC-[J2,0]	-5V
V_VCC+[J1,0]	5V
V_V-HSensor[8,0]	0V
V_V-PSensor[6,0]	0V
V_V-TSensor[10,0]	0V

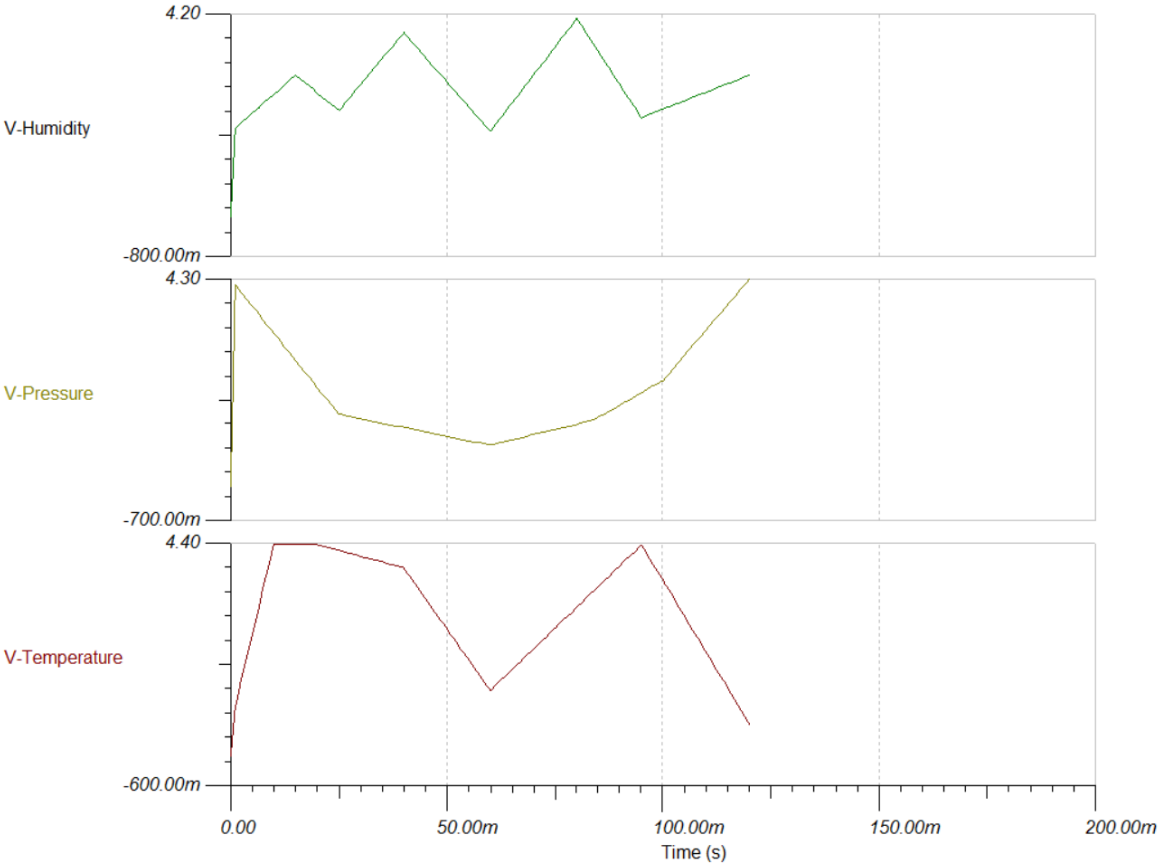
D. Sensor (Voltage-Equivalent) Plot

Shown in the figure below is the converted voltage output of the sensors. Note that due to Infineon Designer’s limitations, time is set in milliseconds instead of minutes.



E. ASP Block Voltage Output

Shown in the figure below is the observed output of each ASP block. Note that due to Infineon Designer's limitations, time is set in milliseconds instead of minutes.



F. Selection of Non-Inverting Amplifier Resistor Values

Resistor values are selected such that the gain obtained is near optimal and the resistance is large enough to lower the power consumption.

Sensor	Maximum Output	Optimal Gain (Vref / Vmax)	Rb (Ω)	Ra (Ω)	Gain
Pressure	2.5265	1.9790	27 k	62 k	1.4354
Humidity	0.9	5.5556	68 k	22 k	4.0909

Appendix E-2

Data Processing Module Power Computations

We refer to the simulations performed in Appendix E-1. For the non-inverting amplifiers, we have the following values:

$$\begin{aligned}
I_{out,p} &= -0.1679\text{ mA}\\
I_{out,h} &= -0.2125\text{ mA}\\
I_{supply} &= \frac{2\text{ mA}}{3}\\
&= 0.6667\text{ mA}\\
V_{out,p} &= -6.55\text{ mV} - 0.9998\text{ mV}\\
&= -7.5498\text{ mV}\\
V_{out,h} &= -13.17\text{ mV} - 0.9997\text{ mV}\\
&= -14.1697\text{ mV}
\end{aligned}$$

Note that each ASP block uses a dual power supply of ±5V. It is assumed that the output current flows outwards. The computation of the power consumption of the non-inverting amplifier is then shown as:

$$\begin{aligned}
P_{ASP} &= I_{supply}\left(V_{supply+} - V_{supply-}\right) + |I_{out}|\left(V_{supply} - V_{out}\right)\\
P_{ASP,p} &= 0.6667\text{ mA}\left(5\text{V} - (-5\text{ V})\right) + 0.1679\text{ mA}\left(5\text{ V} - (-7.5498\text{ mV})\right)\\
&= 7.5075\text{ mW}\\
P_{ASP,h} &= 0.6667\text{ mA}\left(5\text{V} - (-5\text{ V})\right) + 0.2125\text{ mA}\left(5\text{ V} - (-14.1697\text{ mV})\right)\\
&= 7.7322\text{ mW}
\end{aligned}$$

In the case of the voltage follower, we assume that there is no current flowing at the output. This is due to the very large internal resistance in the MCU. Therefore, most of the power consumed will come from the supply. Finally, the power consumption of the data processing module can be computed as the total power consumed by the three ASP blocks.

$$\begin{aligned}
P_{ASP,t} &= I_{supply} \times \left(V_{supply+} - V_{supply-}\right)\\
&= 6.6667\text{ mW}\\
P_{ASP,total} &= 7.5075 + 7.7322\ + 6.6667\\
&= 21.9064\text{ mW}
\end{aligned}$$

The power consumption of the modelled sensors is calculated using the typical supply parameters of a real sensor. If the typical values are not given in the datasheet, we use the maximum ratings instead.

We use the BMP180 [18] model for the pressure sensor and Si7020 [19] model for the humidity sensor. Note that the thermistor model (55031) is already given so the average resistance value is computed. Calculations are shown below:

Pressure Sensor:

$$V_{typ,p} = 2.5\text{V}$$

$$I_{max,p} = 5\text{uA}$$

$$P_{sensor,p} = 0.0125 \text{ mW}$$

Temperature Sensor:

$$V_{supply,t} = 5 \text{ V}$$

$$R_{ave,t} = \frac{603990.8 + 8532.6}{2}$$

$$P_{sensor,t} = \frac{V_{supply,t}^2}{R_{ave,t} + R_2}$$

$$P_{sensor,t} = 0.0647 \text{ mW}$$

Humidity Sensor:

$$V_{max,h} = 3.6 \text{ V}$$

$$I_{max,h} = 150\text{uA}$$

$$P_{sensor,h} = 0.5400 \text{ mW}$$

Total Power:

$$P_{sensor,total} = 0.0125\text{mW} + 0.0647\text{mW} + 0.5400\text{mW}$$

$$P_{sensor,total} = 0.6172 \text{ mW}$$

Appendix F-1

Derivations on Communications Block Schematic Diagram

A. Oscillator Block

For the oscillator block, we first design the astable multivibrator. In this circuit, we used a TL071C op-amp. Based on the parameter values in Infineon Designer, its input resistance is $10\ \mathrm{T\Omega}$ so we can assume that the input terminals serve as a virtual ground. We also note that the maximum slew rate of the TL071C is at $13\ \mathrm{MV}/s$ which is significantly large such that we can assume output linearity.

From Sedra and Smith [20], the period of the symmetrical square wave generated by the astable multivibrator can be determined using the equation:

$$T = 2\tau \ln \frac{1+\beta}{1-\beta} \qquad (\mathrm{F1.1})$$

Where $\tau = R_1 C_1$ is the charging constant and $\beta = R_3/(R_2 + R_3)$ is the feedback fraction from voltage divider network $R_2||R_3$. Selecting $R_1 = R_2 = 9\ \mathrm{k\Omega}$ and $R_3 = 1\ \mathrm{k\Omega}$, we solve for the value of C_1 using equation $(\mathrm{F1.1})$:

$$\beta = \frac{1\ \mathrm{k\Omega}}{1\ \mathrm{k\Omega} + 9\ \mathrm{k\Omega}} = 0.1$$

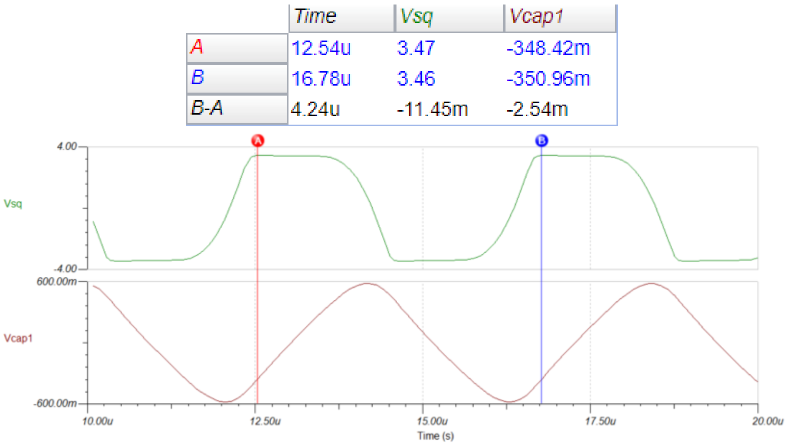
$$T = \frac{1}{f_C} = 2\ \mathrm{\mu s}$$

$$2\ \mathrm{\mu s} = (2)(9000C_1) \ln \frac{1+0.1}{1-0.1}$$

$$\therefore C_1 = \frac{2\ \mathrm{\mu s}}{(2)(9000)\ln(1.1/0.9)}$$

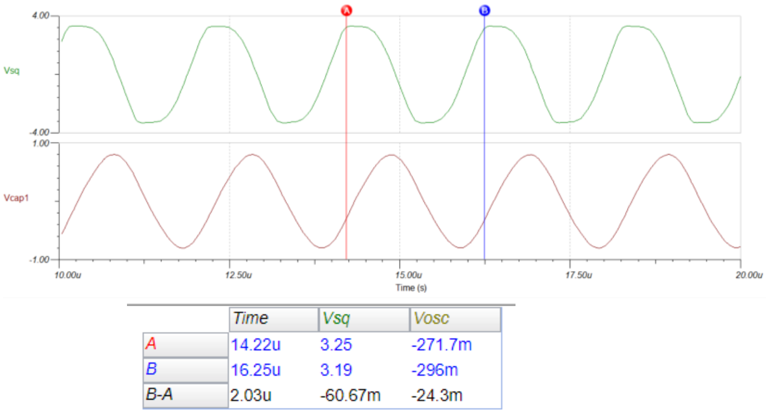
$$= 553.6987\ \mathrm{pF}$$

However, simulations confirm that the measured period is greater than twice the computed period, as shown below:



	Time	Vsq	Vcap1
A	12.54u	3.47	-348.42m
B	16.78u	3.46	-350.96m
B-A	4.24u	-11.45m	-2.54m

As such, the value of C_1 was compensated to produce the required frequency of $500\ \mathrm{kHz}$. At $C_1 = 157\ \mathrm{pF}$, a period of $2.03\ \mathrm{\mu s}$ was achieved:



	Time	Vsq	Vosc
A	14.22u	3.25	-271.7m
B	16.25u	3.19	-296m
B-A	2.03u	-60.67m	-24.3m

Then, for the low pass filter we used an existing model from Albaugh [21]. In this setup, R_4 was increased to $R_4 = 1\ \mathrm{k\Omega}$ to achieve the functionality of the original circuit.

B. Modulator Block

For the BPSK modulator, complementary bipolar junction transistors BC850C/860C were used. From the datasheets provided, the typical BJT threshold voltages are approximately $V_{BE,on} = 0.66\ \mathrm{V}$ [22] and $V_{EB,on} = -0.65\ \mathrm{V}$ [23], respectively.

We assume that the emitter followers are symmetrical, i.e., component values on one side are equal on its opposing side, hence only the design process for the half-circuit involving Q_1, R_{B1}, and R_{E1} will be discussed.

We also note that the message signal is a square wave with two output levels $V_{msg} = \pm 5\ \mathrm{V}$. As such, there will be two cases for the base voltages of Q_1:

When $V_{msg} = 5\ \mathrm{V}$, we expect $V_{BE} > V_{BE,on}$ and thus Q_1 is conducting. Setting $R_{B1} = 1\ \mathrm{k\Omega}$ and $R_{E1} = 2\ \mathrm{k\Omega}$, the simulated voltage gain was determined to be approximately unity. Hence, $V_{o1} = V_c$ for $V_{msg} = 5\ \mathrm{V}$.

When $V_{msg} = -5\ \mathrm{V}$, we expect $V_{BE} < V_{BE,on}$ and thus Q_1 will be in cut-off.

Similarly, we can conclude that Q_2 conducts when $V_{msg} = -5\ \mathrm{V}$ since $V_{EB} < V_{EB,on}$, and that the output is given by $V_{o2} = V_C$ for $V_{msg} = -5\ \mathrm{V}$.

Then, V_{o1} and V_{o2} are the node voltage inputs for the differential amplifier. Here, a TL071C op-amp is used. Again, assuming virtual ground, we can apply nodal analysis at the inverting terminal to get:

$$\frac{V_- - V_{o2}}{R_I} + \frac{V_- - V_{BPSK}}{R_F} = 0$$

$$\left(\frac{R_I + R_F}{R_I R_F}\right)V_- - \left(\frac{1}{R_I}\right)V_{o2} = \left(\frac{1}{R_F}\right)V_{BPSK}$$

$$\therefore V_{BPSK} = \left(\frac{R_I + R_F}{R_I}\right)V_- - \left(\frac{R_F}{R_I}\right)V_{o2} \qquad (\mathrm{F1.2})$$

At the non-inverting terminal, we have the nodal equation:

$$\frac{V_+ - V_{o1}}{R_{d1}} + \frac{V_+}{R_{d2}} = 0$$

$$\left(\frac{R_{d1} + R_{d2}}{R_{d1} R_{d2}}\right)V_+ = \left(\frac{1}{R_{d1}}\right)V_{o1}$$

$$\therefore V_+ = \left(\frac{R_{d2}}{R_{d1} + R_{d2}}\right)V_{o1} \qquad (\mathrm{F1.3})$$

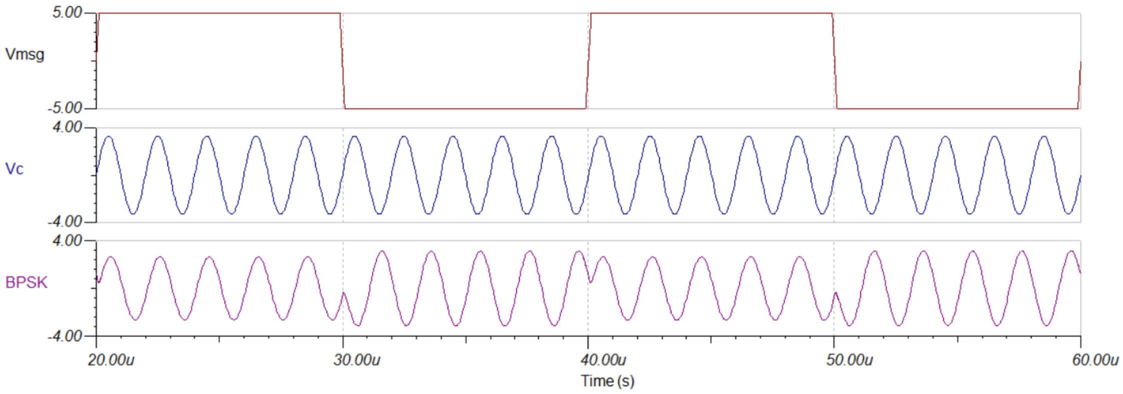
Thus, combining equations $(\mathrm{F1.2})$ and $(\mathrm{F1.3})$, with $V_+ = V_-$:

$$V_{BPSK} = \left(\frac{R_I + R_F}{R_I}\right)\left(\frac{R_{d2}}{R_{d1} + R_{d2}}\right)V_{o1} - \left(\frac{R_F}{R_I}\right)V_{o2} \qquad (\mathrm{F1.4})$$

Setting all resistance values to be $R = 2\ \mathrm{k}\Omega$, equation $(\mathrm{F1.4})$ becomes simplified to:

$$V_{BPSK} = V_{o1} - V_{o2}$$

However, simulations indicate output attenuation at the positive cycle of V_{msg} as shown below. The expected modulator output was achieved when the resistance value of R_{d1} was lowered to $1\ \mathrm{k}\Omega$.



C. Power Amplifier Block

For the transistor power amplifier, complementary bipolar junction transistors BC850C/860C were used. BA314 biasing diodes were also used for both D_1 and D_2. From the datasheets provided, the typical BJT threshold voltages are approximately $V_{BE,on} = 0.66\text{ V}$ [22] and $V_{EB,on} = -0.65\text{ V}$ [23], respectively. The saturation current of both biasing diodes was determined from Infineon Designer's parameter values to be $I_{SD} = 1.221\text{ fA}$.

We define the base-to-base voltage V_{BB} as the measured voltage between the base terminals of the complementary pair:

$$\begin{aligned} V_{BB} &= V_{BE,on} + (-V_{EB,on}) \\ &= 1.31\text{ V} \end{aligned}$$

We note that the base-to-base voltage V_{BB} is in parallel to the biasing diodes. Thus, for both transistors to be conducting the required bias voltage $V_{D12} = 2V_D$ can be related to V_{BB}:

$$\begin{aligned} 2V_D &= V_{BB} \\ \therefore V_D &= 0.65\text{ V} \end{aligned}$$

But for the biasing diodes to conduct, a bias current I_{bias} must be supplied. Assuming negligible BJT base currents, $I_D = I_{bias}$. Moreover, using Shockley's diode equation, we can solve for the value of the required bias current. We assume the thermal voltage to be 26 mV at 25 °C.

$$\begin{aligned} I_{bias} &= I_{SD}(e^{V_D/V_T} - 1) \approx I_{SD}e^{V_D/V_T} \\ &= (1.221\text{ fA})e^{0.65\text{ V}/26\text{ mV}} \\ &= 87.9180\ \mu\text{A} \end{aligned}$$

Rounding up, we select $I_{bias} = 88\ \mu\text{A}$.

D. Coupling Capacitors

For AC coupling of the different stages, we select coupling capacitor values based on the reactance X_C at the operating frequency f_C:

$$X_C = \frac{1}{(2\pi f_C)C}$$

At DC, X_C exhibits infinite impedance and thus effectively prevents loading in between stages. Then, at 500 kHz, the reactance must be negligible such that it acts as a short circuit. Selecting $C_{C1} = C_{C2} = 20\ \mu\text{F}$, we show that the coupling reactances are relatively small compared to the resistances used in the individual stages:

$$\begin{aligned} X_C &= \frac{1}{(1000\pi)(20\ \mu\text{F})} \\ &= 15.92\ \Omega \ll 1\text{ k}\Omega \end{aligned}$$

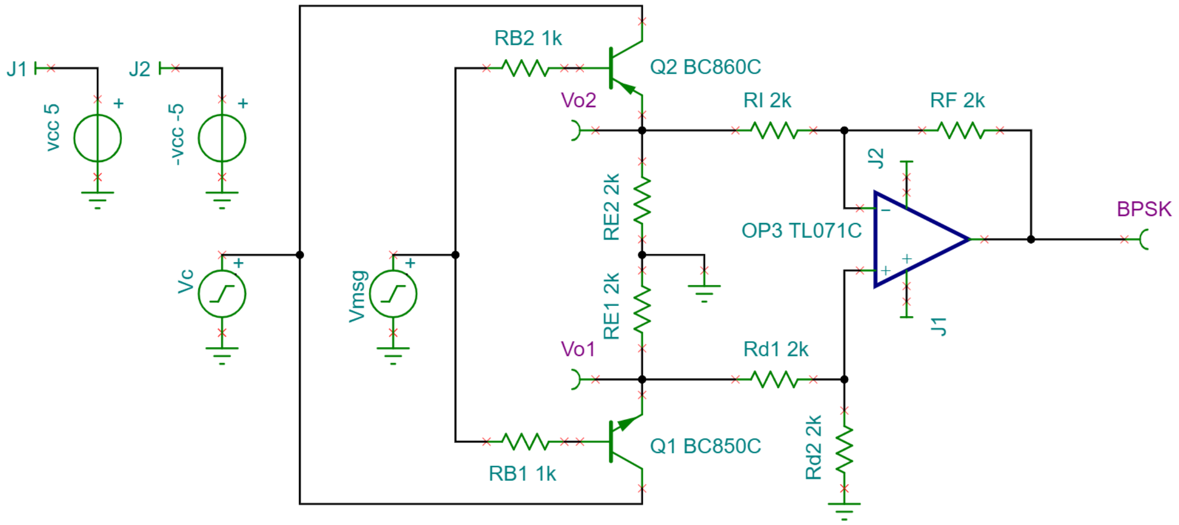
Appendix F-2

Communications Block Simulations

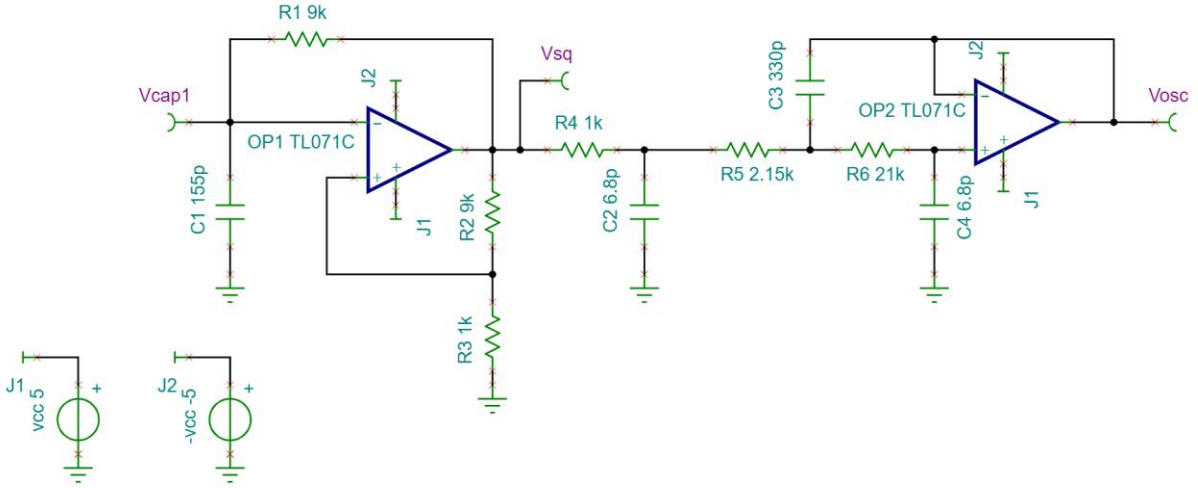
A. Schematic

Shown below is the Infineon Designer schematic diagram of the communications module. Operational Amplifiers (TL071C) use a ±5V dual voltage supply. Design process is broken down into stages (connected by coupling capacitors) to verify the integrity of each output.

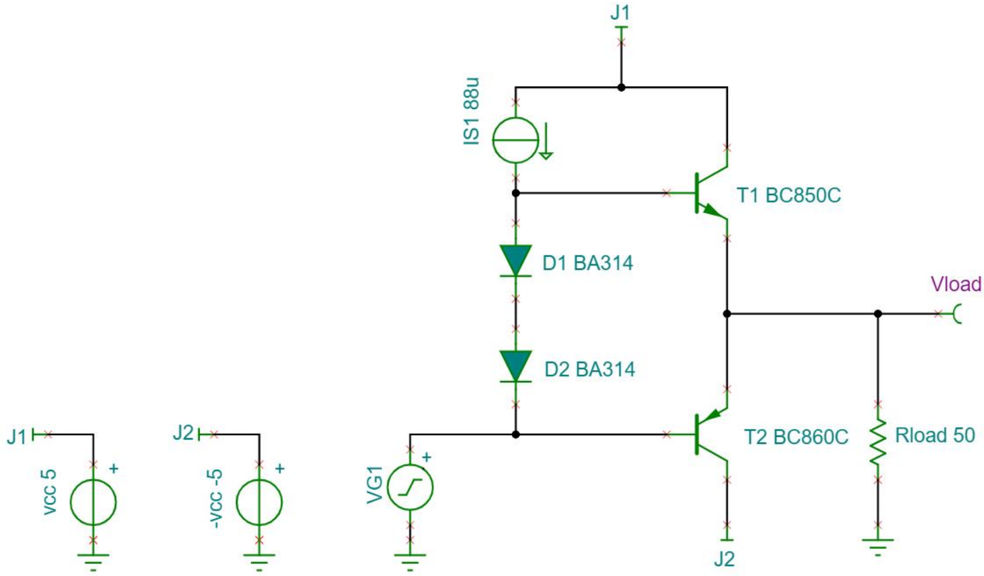
BPSK Modulator Schematic Diagram:



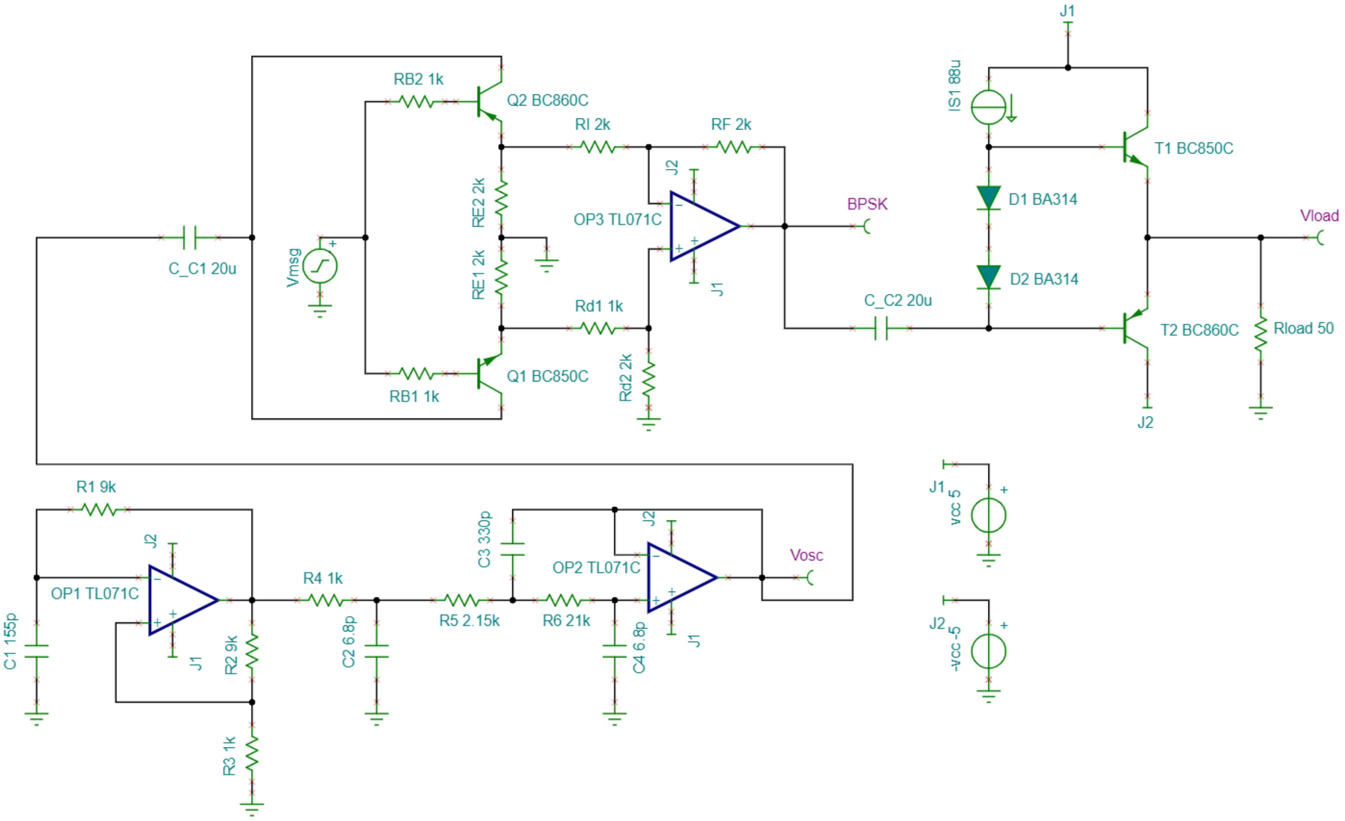
Oscillator Block Schematic Diagram:



Power Amplifier Schematic Diagram:

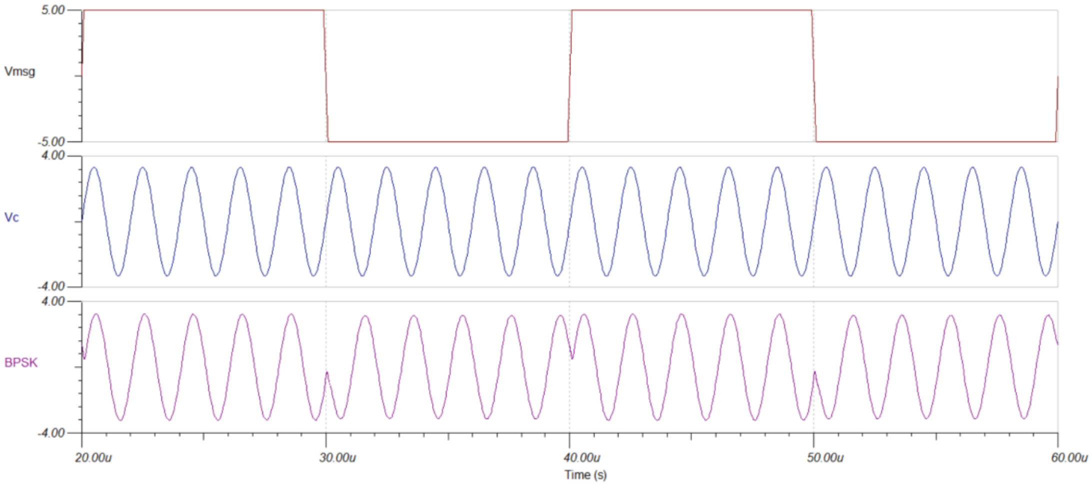


Complete Communications Block Schematic Diagram:

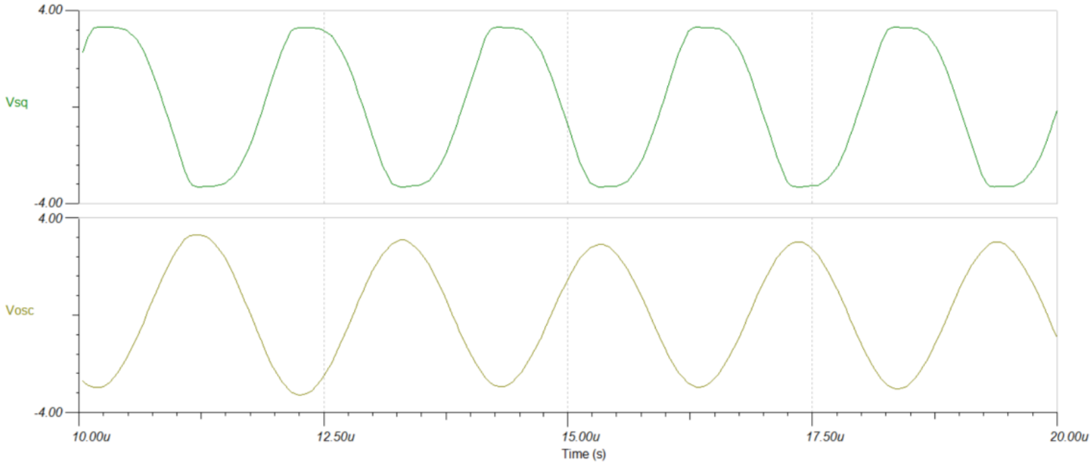


B. Initial Simulation Outputs

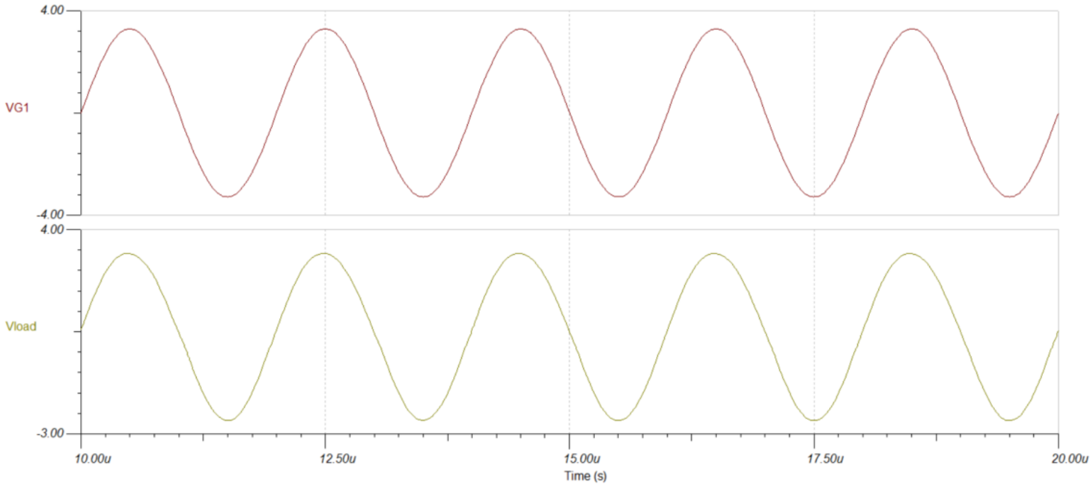
BPSK Modulator Output: (Inputs are modeled by voltage generators)



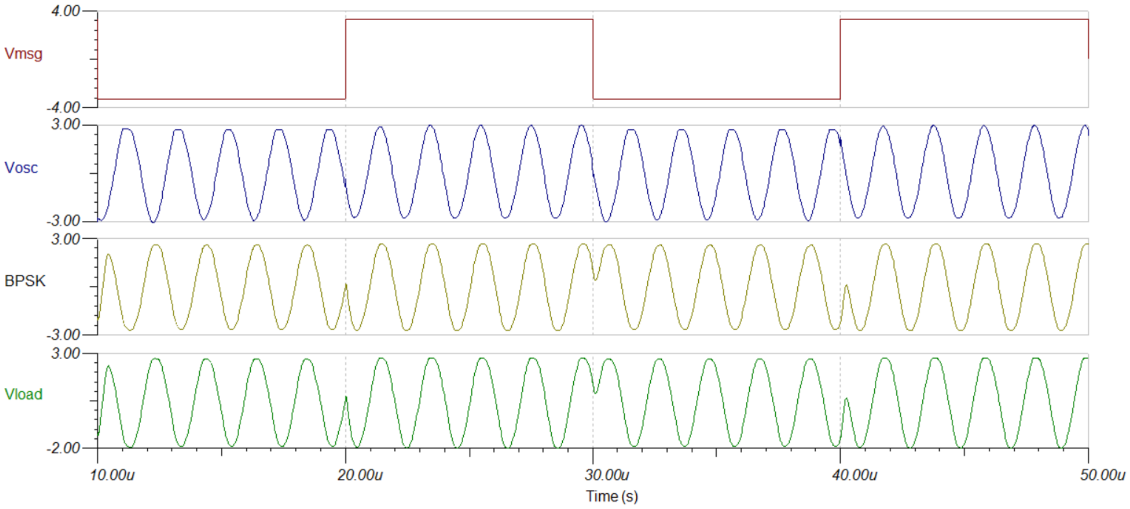
Oscillator Output:



Power Amplifier Output: (Input is modeled by a voltage generator)

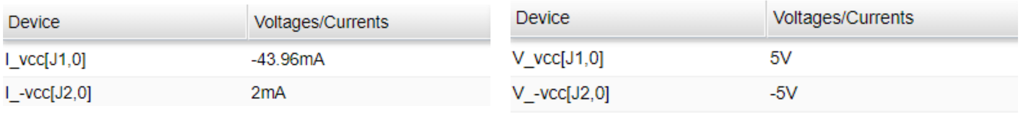


Complete Communications Block Output (message now modelled as $\pm 3.3\ \mathrm{V}$):



C. Large Signal Analysis Output

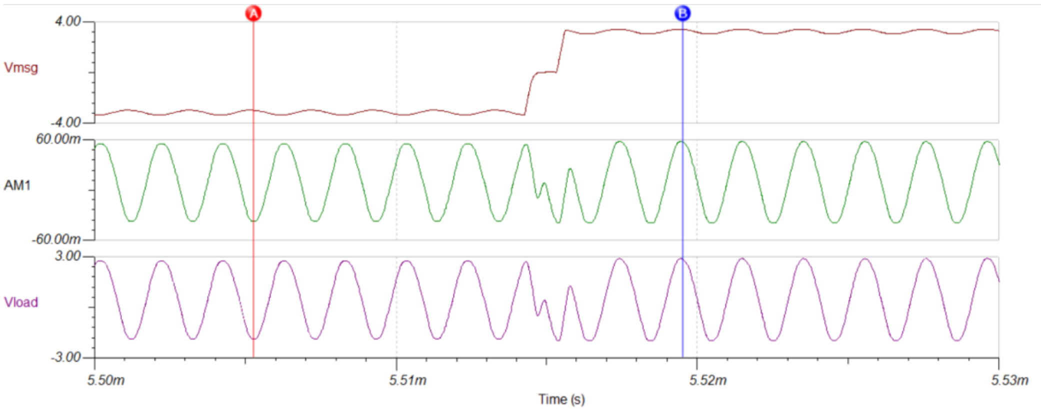
The table below lists the measured DC supply voltages (with their respective currents) of the communications block. There were obtained using Infineon's DC Analysis Tool. We note that the supplied current of $+V_{cc}$ is opposite the measured current since measurement was made with from the positive terminal of the supply to ground.

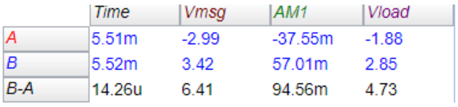


Device	Voltages/Currents
I_vcc[J1,0]	-43.96mA
I_-vcc[J2,0]	2mA

Device	Voltages/Currents
V_vcc[J1,0]	5V
V_-vcc[J2,0]	-5V

Moreover, to measure the RMS load current, ammeter $\mathrm{AM1}$ was connected in series to the load resistor in the schematic diagram. Shown below is the measured peak-to-peak values of the load voltage and load current:





	Time	Vmsg	AM1	Vload
A	5.51m	-2.99	-37.55m	-1.88
B	5.52m	3.42	57.01m	2.85
B-A	14.26u	6.41	94.56m	4.73

Appendix F-3

Communications Module Power Computations

We refer to the simulations performed in Appendix F-2. From the load antenna, we have the following peak values (defined as half the peak-to-peak value):

$$V_{load,peak} = 2.365\text{ V}$$
$$I_{load,peak} = 47.22\text{ mA}$$
$$P_{load} = \frac{1}{2}V_{load,peak}I_{load,peak}$$
$$= 55.8377\text{ mW}$$

We therefore verify that the power delivered to the antenna is sufficient to establish a reliable transmission as stated in the link budget analysis at Appendix A.

Then, solving for the power drawn by the voltage sources, we have the following values:

$$I_{out,vcc+} = 43.96\text{ mA} \qquad I_{out,vcc-} = 2\text{ mA}$$

$$P_{vcc+} = I_{out,vcc+} \cdot V_{cc+} \qquad P_{vcc-} = I_{out,vcc-} \cdot V_{cc-}$$

$$= 219.8\text{ mW} \qquad = -10\text{ mW}$$

$$\therefore P_{supply} = P_{vcc+} + P_{vcc-}$$

$$= 209.8\text{ mW}$$

Therefore, the communications block draws 209.8 mW of power.

Appendix G-1

Power Management Unit Component Values Computations and Simulations

The following equations are obtained from the sixth chapter of Hart's Power Electronics book [27].

A. SEPIC Converter

For the SEPIC converter, the goal is to have a voltage output of 5 V and an output current of 60 mA. In addition to these, the switching frequency is set to 500 kHz. The output voltage ripple is set to 5 mV.

Using the target output voltage and the input voltage, which is 3.7 V, the duty cycle is found to be:

$$\frac{V_o}{V_i} = \frac{D}{1-D}$$

$$D = 0.5747$$

For the inductors L1 and L2, the following equations are used to compute the values.

$$L1 \geq \frac{V_O(1-D)^2}{2f_{sw}I_O D} \qquad L2 \geq \frac{V_O(1-D)}{2f_{sw}I_o}$$

The obtained values are 26.226 µH and 35.441 µH for L1 and L2, respectively. For simplicity, the component values to be used for L1 and L2 are 30 µH and 40 µH, respectively.

For the capacitors C1 and C2, the following equations are used to compute the values. The used voltage ripple for C1 is 1% of the set input voltage (37 mV).

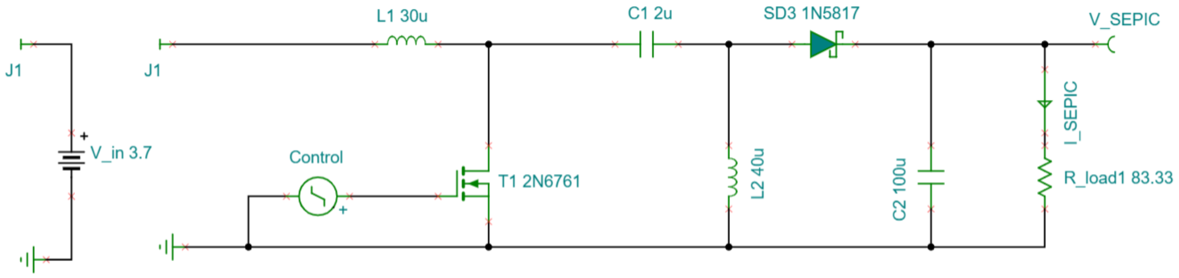
$$C1 \geq \frac{I_O D}{f_{sw}\Delta V_{C1}} \qquad C2 \geq \frac{I_O D}{f_{sw}\Delta V_{C2}}$$

The obtained values are 1.864 µF and 13.793 µH for C1 and C2, respectively. For simplicity, the component values to be used for C1 and C2 are 2 µF and 100 µF, respectively. The output capacitance value is higher to improve the ripples of the system.

The MOSFET chosen for the SEPIC converter is N-MOS Enhancement-type 2N6761 and the diode used is a Schottky diode 1N5817. Both are chosen since they satisfy the requirements of the converter.

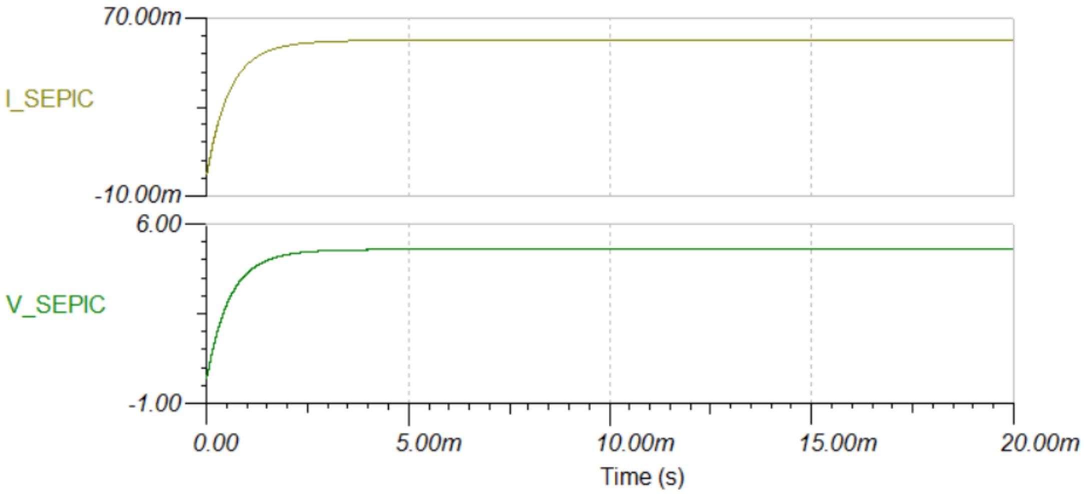
After the calculations and choosing of components, an Infineon Design schematic diagram is built to determine if the required specifications are obtained.

SEPIC Converter Schematic Diagram:



Initially, the desired values of the output voltage and current are not met. Therefore, the duty cycle is adjusted to 0.603.

SEPIC Converter Output:



As seen in the result of the transient analysis, the obtained output current voltage and current match the target values for the SEPIC converter. The output voltage steady state is approximately 5.01 V and the output current steady state is 60.09 mA.

B. Ćuk Converter

For the Ćuk converter, the goal is to have a voltage output of -5 V and an output current of 60 mA. In addition to these, the switching frequency is set to 500 kHz. The output voltage ripple is set to 5 mV.

Using the target output voltage and the input voltage, which is 3.7 V, the duty cycle is found to be:

$$\frac{V_o}{V_i} = \frac{-D}{1-D}$$

$$D = 0.5747$$

For the inductors L1 and L2, the following equations are used to compute the values.

$$L1 \geq \frac{V_O(1-D)^2}{2f_{sw}I_OD} \qquad L2 \geq \frac{V_O(1-D)}{2f_{sw}I_o}$$

The obtained values are 26.226 µH and 35.441 µH for L1 and L2, respectively. For simplicity, the component values to be used for L1 and L2 are 30 µH and 40 µH, respectively.

For the capacitors C1 and C2, the following equations are used to compute the values. The used voltage ripple for C1 is 1% of the set input voltage (37 mV).

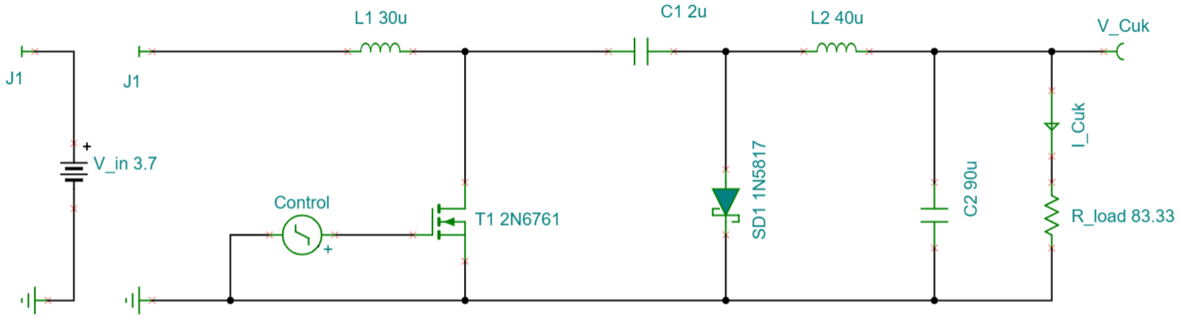
$$C1 \geq \frac{I_OD}{f_{sw}\Delta V_{C1}} \qquad C2 \geq \frac{1-D}{8L_2f^2{}_{sw}\Delta V_{C2}}$$

The obtained values are 1.864 µF and 1.2 µF for C1 and C2, respectively. For simplicity, the component values to be used for C1 and C2 are 2 µF and 90 µF, respectively. The output capacitance value is higher to improve the ripples of the system.

The MOSFET chosen for the SEPIC converter is N-MOS Enhancement-type 2N6761 and the diode used is a Schottky diode 1N5817. Both are chosen since they satisfy the requirements of the converter.

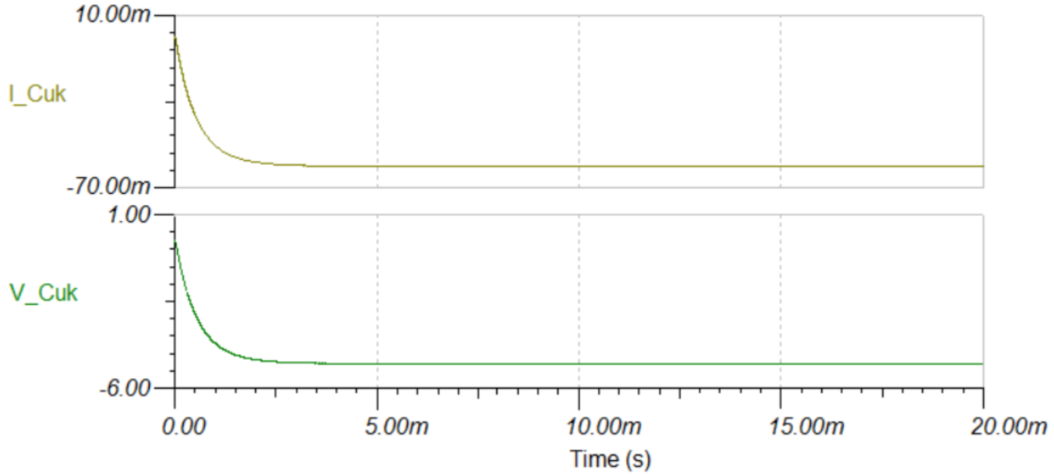
After the calculations and choosing of components, an Infineon Design schematic diagram is built to determine if the required specifications are obtained.

Ćuk Converter Schematic Diagram:



Initially, the desired values of the output voltage and current are not met. Therefore, the duty cycle is adjusted to 0.603.

Ćuk Converter Output:



As seen in the result of the transient analysis, the obtained output current voltage and current match the target values for the Ćuk converter. The output voltage steady state is approximately -5.01 V and the output current steady state is 60.09 mA.

Appendix G-2

Power Management Unit Power Computations

To get the power consumption of the converters used in the power management unit, the output voltage and current are multiplied.

For the SEPIC converter, the average output current is 60 mA and the average output voltage is 5.02 V. Therefore,

$$P_{SEPIC} = (60.09\ mA)(5.01\ V)$$
$$P_{SEPIC} = 301.5\ mW$$

For the Ćuk converter, the average output current is -60 mA and the average output voltage is -5.08 V. Therefore,

$$P_{CUK} = (-60.09\ mA)(-5.01\ V)$$
$$P_{SEPIC} = 301.5\ mW$$

To get the total power consumed by the PMU, the two powers are added:

$$P_{PMU} = 301.5\ mW + \ 301.5\ mW$$
$$P_{PMU} = 602.0\ mW$$

Therefore, the communications block draws 606.0 mW of power.

Appendix H

Power Budget Analysis

The table below shows the power budget ananlysis for the whole CanSat system:

Component	Operating Voltage (V)	Operating Current (mA)	Power (mW)	Duration (h)	Energy Consumption (mWh)
XMC1100	3.7	6.1	22.57	2.5	56.425
XMC1100	5	6.1	30.5	2.5	76.25
PMU			602	2.5	1505
Comms	5	41.96	209.8	2.5	524.5
Pressure	2.5	0.005	0.0125	2.5	0.03125
Humidity	3.6	0.0163	0.5400	2.5	1.35
Temperature	5	0.1500	0.0647	2.5	0.16175
ASP	5	2	21.9064	2.5	54.766
Total			891.3936		2218.484

Using the power budget computed from different modules as seen in Appendices E-2, F-3, and G-2, the power budget table is made. The total power consumption is 2218.484 mWh. This is the expected power consumption for the whole flight of the CanSat including the recording of data. The 3.7 V battery is rated at 1500 mAh. This means that there is 5550 mWh energy available. There is still 3331.5 mWh remaining after the whole duration of the flight. This means that the power is still on budget.

Appendix I

Microcontroller Programming

Communications and Data Processing Module

C code used to perform A-D conversion and bitstream digital output formatting:

```c
//Declarations from DAVE Code Generation (includes SFR declaration)
#include <DAVE.h>

// MAIN_CLOCK is set to 32 MHz
// Delay is approximately: (1/MAIN_CLOCK)*i
void delay(unsigned long int i) { while(i--) {} }

int is_even_parity(uint32_t sensor_data){
    int count = 0;
    for (int i = 0; sensor_data > 0; i++){
        if (sensor_data % 2) { count++; }
        sensor_data >>= 1;
    }
    if (count % 2) { return 0; }
    else { return 1; }
}

void get_bitstream(uint32_t data_1, uint32_t data_2){
    int bitstream_1[21];
        for (int i = 0; i < 21; i++) {
            if (data_1 == 0) {
                bitstream_1[21-i-1] = 0;
            }
            else {
                bitstream_1[21-i-1] = data_1 % 2;
                data_1 >>= 1;
            }
        }
    int bitstream_2[31];
        for (int i = 0; i < 31; i++) {
            if (data_2 == 0) {
                bitstream_2[31-i-1] = 0;
            }
            else {
                bitstream_2[31-i-1] = data_2 % 2;
                data_2 >>= 1;
            }
        }
    for (int i = 0; i < 21; i++) {
        if (bitstream_1[i] == 1) {
            DIGITAL_IO_SetOutputHigh(&DIGITAL_IO_0);
            DIGITAL_IO_SetOutputLow(&DIGITAL_IO_1);
         }
        else {
            DIGITAL_IO_SetOutputLow(&DIGITAL_IO_0);
            DIGITAL_IO_SetOutputHigh(&DIGITAL_IO_1);
        }
        delay(110);
    }
    for (int i = 0; i < 31; i++) {
        if (bitstream_2[i] == 1) {
            DIGITAL_IO_SetOutputHigh(&DIGITAL_IO_0);
            DIGITAL_IO_SetOutputLow(&DIGITAL_IO_1);
        }
        else {
            DIGITAL_IO_SetOutputLow(&DIGITAL_IO_0);
            DIGITAL_IO_SetOutputHigh(&DIGITAL_IO_1);
        }
        delay(110);
    }
}

// results[0] = pin 2.2, results[1] = pin 2.4, results[2] = pin 2.3
XMC_VADC_RESULT_SIZE_t results[3];
int n = 3;

void ADC_Measurement_Handler() {
// When n is 3, data from all 3 sensors have been read.
// We can then get the bitstream.
    if (n==3) {
        n = 0;
        // Packet to be transmitted.
        // Packet is separated into two sections, one 21 bit data and one
31 bit data.
        // The value here is still in decimal form and needs to be
converted to 1s and 0s.
        uint32_t data_1;
        uint32_t data_2;

        // Logic to compute the check bit set to even parity, such that
        // parity bit is set to 0 when the number of 1s is even.
        if (is_even_parity(results[0])==1) {
            data_1 = 0x1FC000 + (results[0] << 1);
        }
        else {
            data_1 = 0x1FC001 + (results[0] << 1);
        }
        if (is_even_parity(results[1])==1) {
            data_2 = (results[1] << 20) + 0x0001;
        }
        else {
            data_2 = (results[1] << 20) + 0x40001;
        }
        if (is_even_parity(results[2])==1) {
            data_2 = data_2 + (results[2] << 6);
        }
        else {
            data_2 = data_2 + (results[2] << 6) + 0x0020;
        }
        get_bitstream(data_1, data_2);
        delay(110);
        ADC_MEASUREMENT_StartConversion(&ADC_MEASUREMENT_0);
    }
    results[n] = ADC_MEASUREMENT_GetGlobalResult();
    n++;
}
```

```
int main(void) {
    DAVE_STATUS_t status;
    status = DAVE_Init();           /* Initialization of DAVE APPs  */
        if(status != DAVE_STATUS_SUCCESS) {
            XMC_DEBUG("DAVE APPs initialization failed\n");
            while(1U);
        }
    ADC_MEASUREMENT_StartConversion(&ADC_MEASUREMENT_0);
    while(1U) {
    }
}
```

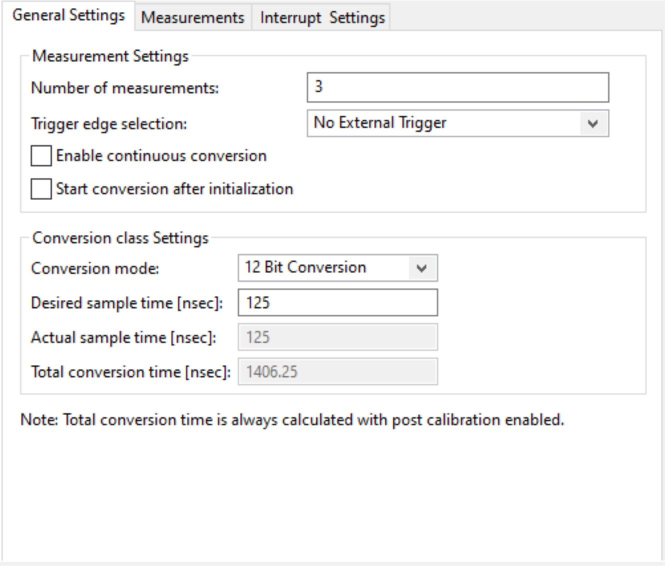
Power Module

C code used to perform duty cycle regulation for output voltages using ADC:

```
//Declarations from DAVE Code Generation (includes SFR declaration)
#include <DAVE.h>

//initialize
XMC_VADC_RESULT_SIZE_t result;
double v_out;
double v_in;
double duty;

void ADC_Measurement_Handler() {

       result = ADC_MEASUREMENT_GetGlobalResult(); //get digital value
       result = result >> 2;              //bit shift to right
       v_out = 2*result*5 / 1024;  //get actual output voltage
       v_in = v_out * (1 - duty) / duty;        //compute input voltage

       duty = 5 / (5 + v_in);                   //adjust duty cycle

       PWM_SetDutyCycle(&myPWM, 10000*duty);    //*10000 since duty is
in decimal
}

int main(void)
{
  DAVE_STATUS_t status;
  //PWM_STATUS_t status1;
  status = DAVE_Init();           /* Initialization of DAVE APPs  */
  PWM_Start(&myPWM);

  if (status != DAVE_STATUS_SUCCESS)
  {
    XMC_DEBUG("DAVE APPs initialization failed\n");
   while(1U);
  }

  ADC_MEASUREMENT_StartConversion(&ADC_MEASUREMENT_0);

  while(1U) {
  }
}
```

ADC Configuration

Configuration used to read and convert voltage levels:



Measurement Settings

Number of measurements: 3

Trigger edge selection: No External Trigger

☐ Enable continuous conversion

☐ Start conversion after initialization

Conversion class Settings

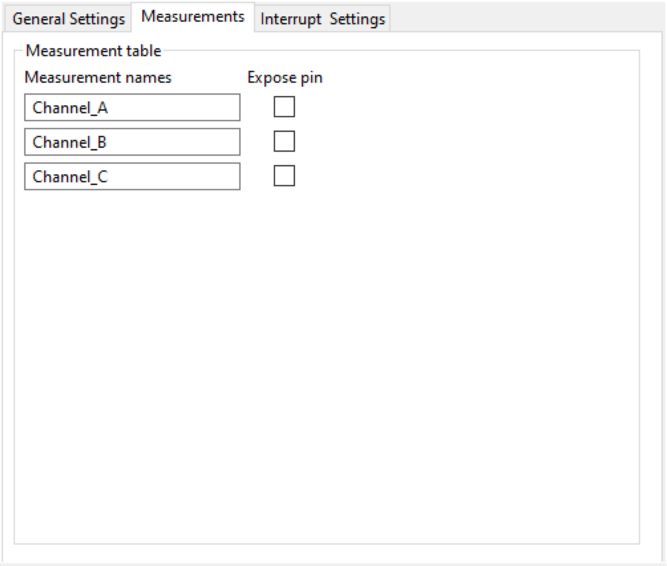
Conversion mode: 12 Bit Conversion

Desired sample time [nsec]: 125

Actual sample time [nsec]: 125

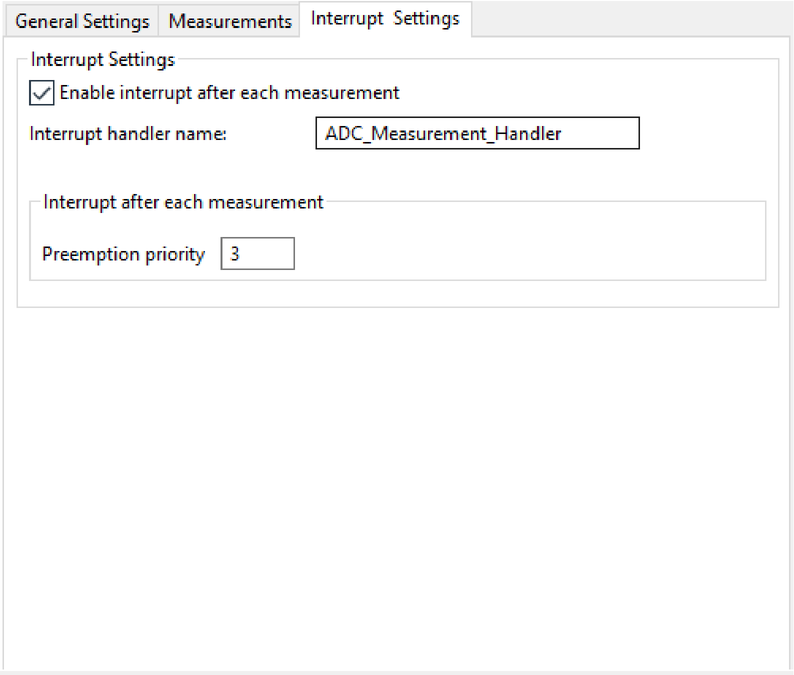
Total conversion time [nsec]: 1406.25

Note: Total conversion time is always calculated with post calibration enabled.



Measurement table

Measurement names	Expose pin
Channel_A	☐
Channel_B	☐
Channel_C	☐



Interrupt Settings

☑ Enable interrupt after each measurement

Interrupt handler name: ADC_Measurement_Handler

Interrupt after each measurement

Preemption priority 3