WCRC CubeSat Telemetry Helper

1. Introduction

Satellites are widely used for telecommunication, in particular in a remote area which is not covered by terrestrial communication. Satellites have several missions e.g., earth monitoring, weather monitoring, remote sensing, and navigation. Nano-satellites is a small satellite between 1 to 10 kg. It can be used for a limited function such as transmit data sensory. Ground station is a part of satellite systems. It connects the earth side with the satellite and contains of several components. Those components are antenna, tracking rotator, transceiver module, and others. Based on ground station main function, it has two functions. There are master station and Telemetry, Tracking, and Command (TT&C) station. To build ground station needs a lot of cost and can be reduced by using Software Defined Radio (SDR).

SDR is a programmable radio module that can be reconfigured in physical layer level using a software. SDR is flexible to designed, low cost and has efficient power. A design of loop back system has been proposed for TT&C in GNU Radio. Yulianti et. al. designed and simulate an SDR for nano-satellite using 2.4 GHz USRP module. Another research also use USRP for ground station with CCSDS frame.

GNU Radio is a free license software to simulate and build radio applications. This paper design a TT&C subsystem in nano-satellite ground station. It use GNU radio for design and HackRF for simulation. It design framing data for packet encoder blocks based on the nano-satellite parameters and tested with two scenarios.

2. System Design

TT&C is used to get information from nano-satellite. It can contains of satellite condition and satellite location for monitoring the nanosatellite from ground station. The condition of the satellite is measured using sensors which will define the spacecraft healthy state and transmitted using telemetry system, which commonly uses VHF/ UHF band. TT&C also sends the position of the satellite by transmitting the location data based on calculation or space based navigation system which will helps the user to track the spacecraft location. Another important function of TT&C is process the commands from the earth. These contain of various commands such as attitude command, data command, even payload shutdown command[7]. One of the nano- satellite communication modules that supports TT&C is AX100, it uses GMSK as modulation method, and works at UHF band[8].

Specification based on nano satellite transceiver module named AX100 by GOMSPACE shown in Table 1. Then its specification turned into flowgraph in GNU Radio that tested with two scenarios.

Table 1. System Specification.

Items	Parameter
Frequency	437.5MHz
Modulation	GMSK
Method Framing	CCSDS
BER	1.00E-05
FEC	R=1/2, K=7 & RS=(255,223)

2.1. First Scenario

The first scenario is the simple one where the characteristic of TT&C communication is implemented in a computer using GNU Radio and SDR as the radio front end module. Design of first scenario shown in Figure 1 and Figure 2.

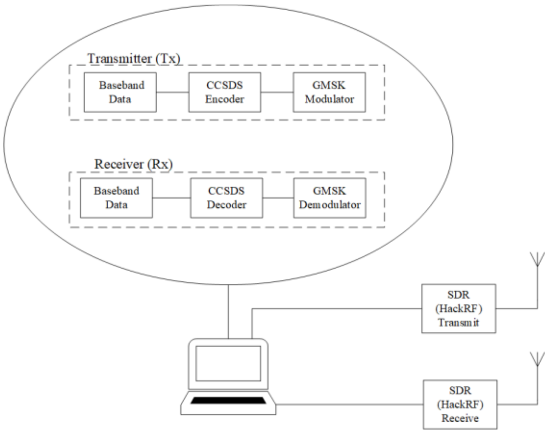
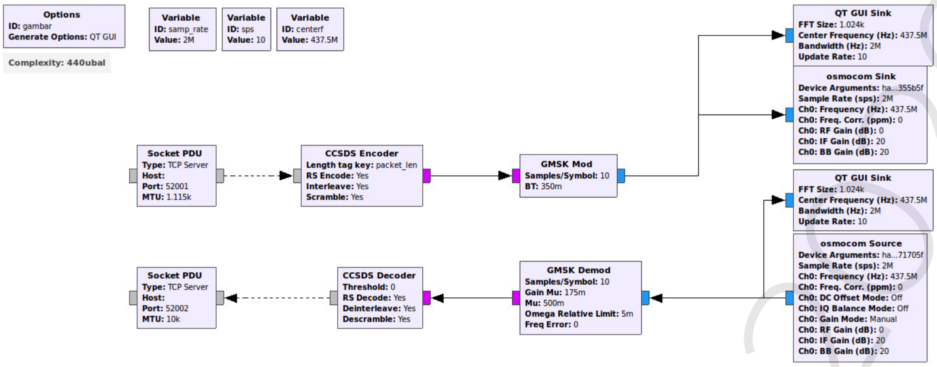


Figure 1. First Scenario Design.

Figure 2. First Scenario Setup.

The simulation block system have been designed and matched with how the AX100 specification. Baseband data used were the image with size 11506 bytes, and text file with size



Options
ID: gambar
Generate Options: QT GUI

Complexity: 440ubal

Variable
ID: samp_rate
Value: 2M

Variable
ID: sps
Value: 10

Variable
ID: centerf
Value: 437.5M

Socket PDU
Type: TCP Server
Host:
Port: 52001
MTU: 1.115k

CCSDS Encoder
Length tag key: packet_len
RS Encode: Yes
Interleave: Yes
Scramble: Yes

GMSK Mod
Samples/Symbol: 10
BT: 350m

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 437.5M
Bandwidth (Hz): 2M
Update Rate: 10

osmocom Sink
Device Arguments: ha...355b5f
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 437.5M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 437.5M
Bandwidth (Hz): 2M
Update Rate: 10

osmocom Source
Device Arguments: ha...71705f
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 437.5M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

GMSK Demod
Samples/Symbol: 10
Gain Mu: 175m
Mu: 500m
Omega Relative Limit: 5m
Freq Error: 0

CCSDS Decoder
Threshold: 0
RS Decode: Yes
Deinterleave: Yes
Descramble: Yes

Socket PDU
Type: TCP Server
Host:
Port: 52002
MTU: 10k

Figure 3. First Scenario GNU Radio Flowgraph.

2729 bytes. Framing was based on standard from the Consultative Committee for Space Data Systems (CCSDS). The parameter of its frame is defined in Blue Book CCSDS Recommended Standard for TM Synchronization and Channel Coding. The length of a single frame is 1279 Bytes, where consists of 4 Bytes for both attached sync marker (ASM) and forward error correction, 1115 Bytes for data, the rest of 160 Bytes for parity. In each 1115 Bytes for data divided into 5 subframes where length of each is 223 Bytes.

The modulation is GMSK which is widely used in small satellite communication module. GMSK also used in AX100. Gaussian minimum shift keying (GMSK) is one of variants modulation method from minimum shift keying (MSK). GMSK pulse form from MSK square pulse replaced by sinusoidal pulse with gaussian filter. Afterward flowgraph executed in one laptop then transmit and receive through SDR hardware, in this case is HackRF One. HackRF is half duplex SDR hardware. Hence need two of HackRF to run this scenario. The designed of first scenario above implemented in GNU Radio Companion software like the Figure 3.

The flowgraph in Figure 3 has been adapted from the design in Figure 1. It is one file GNU Radio that include transmit and receive flowgraph. At the top of the file there is flowgraph for transmit, whilst at the bottom of the file there is flowgraph for receive. Baseband data represent by Socket PDU block at the first and the end of file. To meet specification that have been designed, for framing CCSDS used CCSDS Encoder and CCSDS Decoder blocks. Those block are OOT Module in GNU Radio for processing data which is encoded according to the CCSDS Standard, also those block handles Reed Solomon, interleaving, and scrambling/ randomization [12]. Based on specification as well modulation method represent by GMSK Mod and GMSK Demod both in transmit side and receive side. Parameter for modulation method adjusted to SDR Hardware specification, in this case is HackRF One.

At last, SDR Hardware represent by osmocom sink and osmocom source. Those block are OOT Module in GNU Radio as well as CCSDS Encoder, and works for supporting SDR Hardware except USRP. Declared Device Arguments in osmocom blocks is important, because of its use two devices. Setup for first scenario shown in Figure 2. The red circle act as transmitter and the blue one act as receiver. Both HackRF connect into one laptop, and run on one file.

2.2. Second Scenario

Second scenario used two laptops instead of one, and its designed flowgraph same as first scenario however divided into two that will run on each laptop. Second scenario represent communication between nanosatellite and its ground station. Design of second scenario is shown in Figure 4. The second scenario setup is shown in Figure 5.

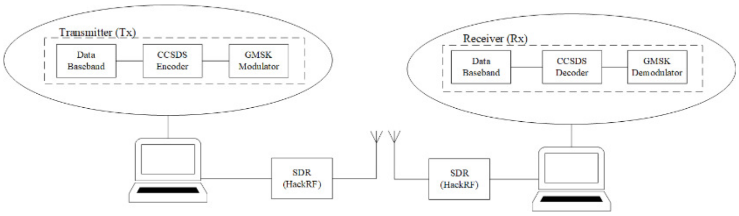


Figure 4. Second Scenario Design.

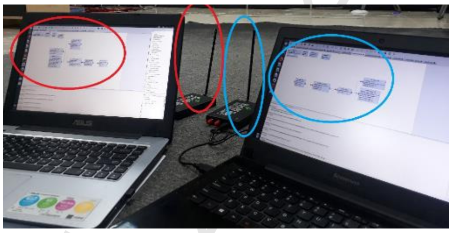
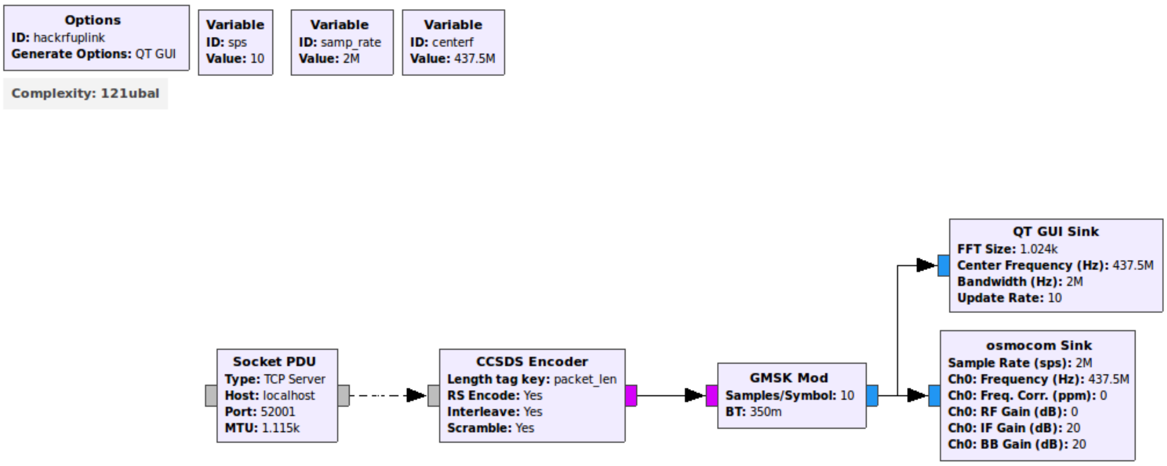


Figure 5. Second Scenario Setup.



Options
ID: hackrfuplink
Generate Options: QT GUI

Complexity: 121ubal

Variable
ID: sps
Value: 10

Variable
ID: samp_rate
Value: 2M

Variable
ID: centerf
Value: 437.5M

Socket PDU
Type: TCP Server
Host: localhost
Port: 52001
MTU: 1.115k

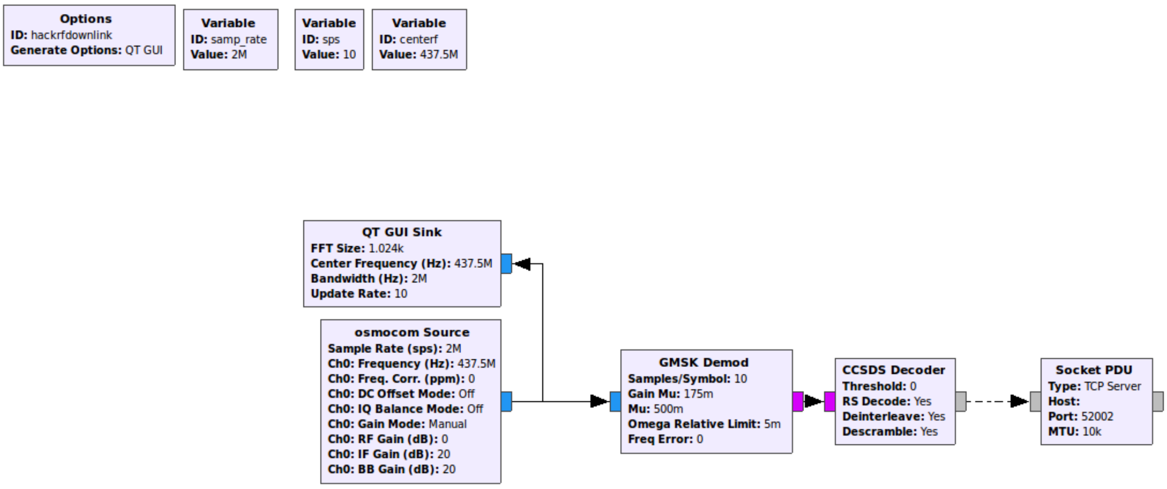
CCSDS Encoder
Length tag key: packet_len
RS Encode: Yes
Interleave: Yes
Scramble: Yes

GMSK Mod
Samples/Symbol: 10
BT: 350m

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 437.5M
Bandwidth (Hz): 2M
Update Rate: 10

osmocom Sink
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 437.5M
Ch0: Freq. Corr. (ppm): 0
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

Figure 6. Transmitter for Second Scenario GNU Radio Flowgraph.



Options
ID: hackrfdownlink
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 2M

Variable
ID: sps
Value: 10

Variable
ID: centerf
Value: 437.5M

QT GUI Sink
FFT Size: 1.024k
Center Frequency (Hz): 437.5M
Bandwidth (Hz): 2M
Update Rate: 10

osmocom Source
Sample Rate (sps): 2M
Ch0: Frequency (Hz): 437.5M
Ch0: Freq. Corr. (ppm): 0
Ch0: DC Offset Mode: Off
Ch0: IQ Balance Mode: Off
Ch0: Gain Mode: Manual
Ch0: RF Gain (dB): 0
Ch0: IF Gain (dB): 20
Ch0: BB Gain (dB): 20

GMSK Demod
Samples/Symbol: 10
Gain Mu: 175m
Mu: 500m
Omega Relative Limit: 5m
Freq Error: 0

CCSDS Decoder
Threshold: 0
RS Decode: Yes
Deinterleave: Yes
Descramble: Yes

Socket PDU
Type: TCP Server
Host:
Port: 52002
MTU: 10k

Figure 7. Receiver for Second Scenario GNU Radio Flowgraph.

Second scenario uses similar design as the first scenario but using two computer that describe transmitter and receiver. Baseband data used in this scheme are image and file text with the same size as first scenario and so do the framing and modulation. The designed of second scenario implemented in GNU Radio Companion software shown in Figure 6 and Figure 7. The flowgraph divide into two files which one for transmit side and one for receive side. On the transmitter side like Figure 6 used same blocks as first scenario before but separated into two files. Baseband, framing, modulation method, and SDR Hardware use the same blocks. Also in receiver side shown in Figure 7 used same blocks as first scenario. Parameter Device Arguments does not declare because only one osmocom block used in one file, different with first scenario which use two osmocom blocks in one file. Setup for second scenario shown in Figure 5. As the image shown in Figure 5, the red circle act as transmitter and the blue one act as receiver. Both HackRF connect into different laptop which run each flowgraph that have been designed in Figure 6 and Figure 7.

3. Result and Analysis

The block systems from both scenario have been designed in GNU Radio and integrated with HackRF module. The experiment of each system performed 10 times, for both scenarios, with two baseband data each scenario. The experiment result is observed to know how the received baseband data and to compare the original baseband data with it. It is included counting the number errors while the transmission is running. To ensure that the transmit signal is using GMSK modulation then the QT GUI Sink block is used on all scenarios to monitor the signal. The output in time domain is shown in Figure 8.

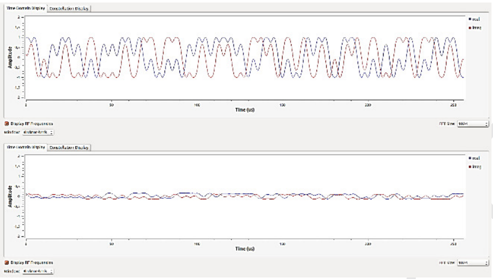


Figure 8. First Scenario Time Domain Output.

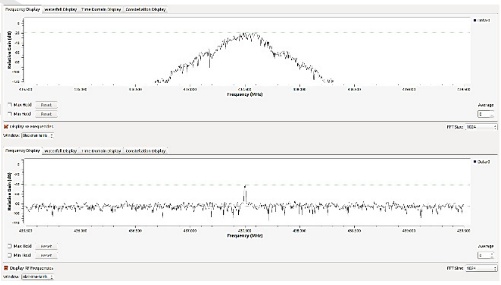


Figure 9. First Scenario Frequency Domain Output.

The image shown at Figure 8 shows that the baseband signal has been succesfully modulated by the GMSK block system. At the top side of the Figure 8 shows transmit signal whilst at the bottom side shows receive signal. There were difference at amplitude and signal form in both transmit signal and receive signal. These difference occured because the signal experienced some noise and attenuation transmission. While at the time second scenario were executed, it also transmitted and received GMSK modulation, where it can be proved by seeing output in time domain by QT GUI Sink block.

Image that shown on Figure 10 represent signal form in time domain which have been modulated by GMSK modulation at transmit side. Also the image that shown on Figure 11 represent signal form in time domain which have been modulated by GMSK modulation at receive side. Whilst to ensure that signal transmitted on 437.5MHz in all scenarios, the same block is used and observed in frequency domain. From top side of the image that shown in Figure 9 represent signal in frequency domain that have been transmitted in 437.5MHz, where at bottom side of the image shown signal that have been received in 437.5MHz. As well as on second scenario, it use same frequency as first scenario, that can be prove in Figure 12 and Figure 13. Image that shown in Figure

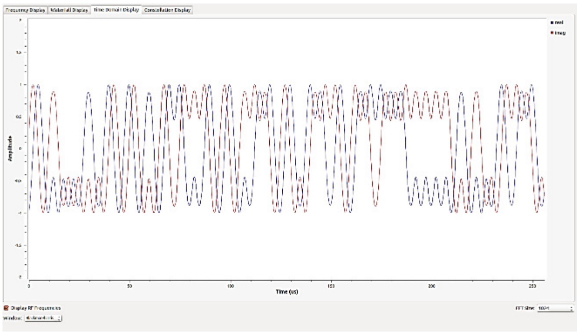


Figure 10. Second Scenario Transmitter Time Domain Output

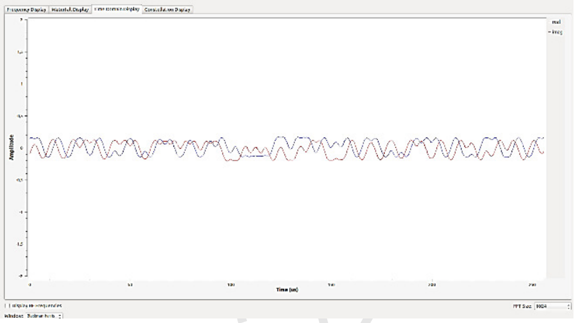


Figure 11. Second Scenario Receiver Time Domain Output

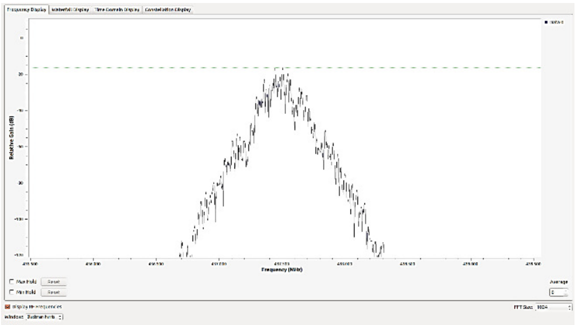


Figure 12. Second Scenario Transmitter Domain Frequency Output

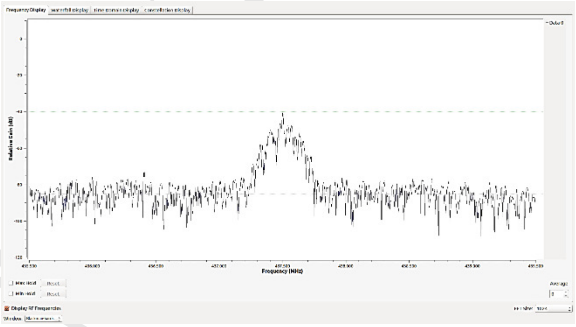


Figure 13. Second Scenario Re- ceiver Frequency Domain Output

12 represent signal in frequency domain that have been transmit at 437.5MHz. Also in receiver side as shown in Figure 13, it receive at frequency 437.5MHz.

The result from transmitting image in first scenario have no difference with the original and the received file, as shown in Figure 14. But on its size there is difference. The original image have size 11506 Bytes whilst received image have size 11150 Bytes. Which is divide into ten frames causes by CCSDS Encoder.

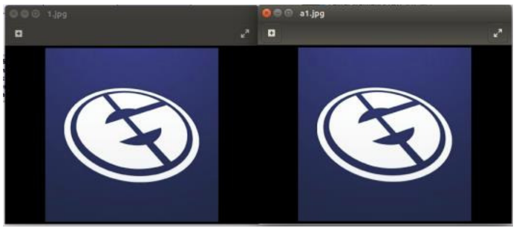


Figure 14. Comparison between Original Image and Received Image in First Scenario.

First scenario performed ten times to transfer the image and file text. The transmit errors occured in every trial. Number of errors are obtained from notification cli in GNU Radio by activate Verbose in CCSDS Encoder block. Error in every trials were caused by nature noise and attenuation of transmission. The average error on image transfer is 2.3 error. This error caused the size and the information int the file suffered decreasing size and losing the information. As well as the image transmit before, the size of file text also decreasing. That caused by CCSDS Encoder that divided it into two frames. The result file text content from first scenario shown

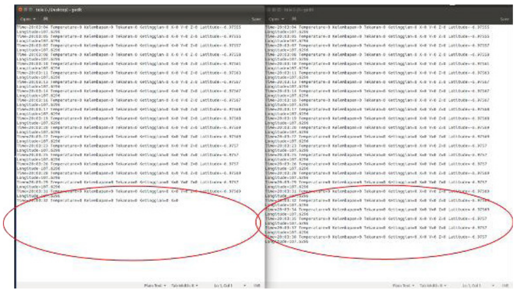


Figure 15. Comparison between Received File Text and Original File Text in First Scenario.

On the left side in Figure 15, is received file. Compared to the right one which is the original file. Received file have been changed, some part of its content gone missing. This cause by CCSDS Encoder block which implemented on system that divide file into several frames. From difference of its content, therefore the size also have difference, where the original file text have size 2729 Bytes and the received file text have size 2230 Bytes.

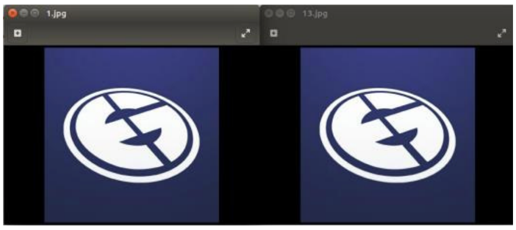


Figure 16. Comparison between Original Image and Received Image in Second Scenario.

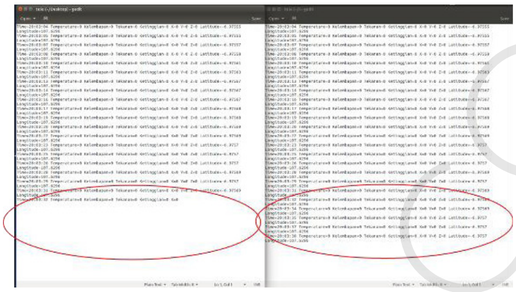


Figure 17. Comparison between Received File Text and Original File Text in Second Scenario.

The second scenario also has been done in ten times of trials, both image transfer and file text transfer. For the result of transmitting image in second scenario have same result as first scenario. The comparison between original and received image look similar, yet the size have difference as the first scenario where divide by ten frames, causes by CCSDS Encoder block, where the original image have size 11506 Bytes and the received image have size 11150 Bytes. The result of transmitting image in second scenario shown in Figure 16.

On right side in Figure 16 is received image whilst in left side is original image. There is no difference between both images. Second scenario run in ten times as well, both image and file text transfer. Unlike first scenario, second scenario have more errors. Where average error on image transfer is 25.9 error. This error causes by nature noise and attenuation of transmission. Also on file text transfer in second scenario have same result as first scenario. The content and size between the original file and received file have difference, same as first scenario where the original file text have size 2729 Bytes and the received file text have size 2230 Bytes. The result file text content from second scenario shown in Figure 17.

WCRC CONTACT 2020 GSC