SOFTWARE-DEFINED RADIOS IN SATELLITE COMMUNICATIONS

2 authors:

Gara Quintana
Norwegian University of Science and Technology

Roger Birkeland
Norwegian University of Science and Technology

SOFTWARE-DEFINED RADIOS IN SATELLITE COMMUNICATIONS

Gara Quintana-Díaz(1)**, Roger Birkeland**(1)

(1)*Norwegian University of Technology and Science (NTNU), O.S. Bragstad plass 2A, 7491 Trondheim, Norway. gara.quintana@ntnu.no, roger.birkeland@ntnu.no*

ABSTRACT

A Software Defined Radio (SDR) is a flexible technology that enables the design of adaptive communications systems. A generic hardware design can be used to address different communication needs, such as changing frequencies, modulation schemes and data rates. Applied to small satellites, some of the implications are increased data throughput when down-linking or up-linking by varying communications parameters and making use of one hardware design and implementation for communicating for many missions, just by updating the software. Therefore, development time for small satellite communication systems can be reduced in the future. This one of the reason why many universities and other organisations around the world are investing in this type of space technology. The technology can support different kinds of applications, such as Earth observation and communication services. This paper analyses various hardware and software platforms and includes a survey on SDRs that have been designed and developed for satellite communications in the last years. In the survey both ground stations and satellites using SDR have been included. Furthermore, a short discussion on SDR designs have been included.

1 INTRODUCTION

The interest in small satellites (or SmallSats) is continuously growing, both in CubeSats and other customised platforms. Many universities and other organisations around the world are investing in this type of space technology for various applications, such as space exploration and Earth observation. When observing our planet there are two especially relevant areas to focus on: oceans, as 71 % of the Earth is water [1], and Arctic monitoring, because of the dramatic effect of global warming. In-situ monitoring of these extremely harsh areas is difficult, expensive and they are not fully covered by communication systems [2]. This is one reason why it is important to research new solutions in order to improve ocean and Arctic monitoring. One possibility is to deploy a coordinated infrastructure composed of different types of vehicles and platforms, such as AUVs, UAVs and small satellites [2].

The Norwegian University of Science and Technology (NTNU) together with the Center for Autonomous Marine Operations and Systems (NTNU-AMOS) have recently launched a new research programme. It has a concerted and unified cross-disciplinary focus on designing, building and operating small satellites (or SmallSats) as parts of a system of autonomous robots and agents for maritime sensing, surveillance and communication. These activities should contribute to fundamental and interdisciplinary research on autonomous systems in marine applications. The programme is associated with the Faculty for Information Technology and Electrical Engineering's strategic research area Coastal and Arctic Maritime Operations and Surveillance (CAMOS) and has planned two missions. The first is to acquire high quality images for oceanographic studies using a Hyperspectral Imager (HSI) and the second one to provide Arctic researchers with easier and faster access to scientific data by using a flexible communications system.

One important aspect to consider when building any type of satellites is communications. Communication systems enable data transfer between sensor systems, satellites and end users. Most kinds of communications systems are designed for worst-case scenarios, and satellite channel characteristics are highly variable due to atmospheric and ionospheric effects, especially in Low Earth Orbits (LEO). Designing for worst-case leaves an expensive and overly designed system that does not maximise channel capacity. To compensate for this, there is a need to develop enhanced communications systems that can adapt to variable characteristics, for instance changing modulation, power levels or carrier frequency on-the-fly.

Software Defined Radio (SDR) is a flexible technology which enables the design of an adaptive communications system. This means that a generic hardware design can be used to address different communication needs, with varying frequencies, modulation schemes and data rates [3]. Applying this concept to small satellites can increase data throughput, add the possibility to perform software updates over-the-air and make it possible to reuse the hardware platform for multiple missions with different requirements [4]. Therefore, development time for future small satellite communication systems can be reduced, even though the development time of the first implementation might be longer than for a traditional radio system.

However, this idea of launching SDR into space is not new. There are many universities, agencies and companies that are currently addressing this issue and some have already launched their own designs. Various SDR platforms and designs are analysed for use in small satellites in challenging scenarios, data retrieval from diverse Arctic sensors or multi-agent communications, for instance. This paper also studies the state-of-the-art of SDR both for spacecraft and ground stations developed by different universities and organisations.

2 AVAILABLE HARDWARE PLATFORMS

In addition to requirements for frequency, bandwidth and regulations found in every communication system, Software Defined Radios (SDR) are highly dependent on the hardware platform used to run the software. In small satellites, the main design drivers are size, mass, cost and power consumption.

In Figure 1 an overview of some SDR platforms is shown. The vertical axis is *cost* whereas the horizontal axis is *mass*. These are two important aspects to consider when choosing a radio suitable for a small satellite mission. Ideally, the best platform would be the one on the lower left corner of the graph: an inexpensive and light solution. In our comparison, GomSpace SDR is the most expensive, and it has an average mass. However, it is also the only space-qualified hardware platform. While decreasing the cost, the next platforms is the different EPIQ and USRP models. The inexpensive platforms, with a cost of less than 300 €, are the ones from FunCube, Lime and RTL. All with average to low mass.

Each of the platforms are described briefly below.

2.1 GomSpace SDR

The SDR system is built by combining three standalone components: GomSpace NanoCom TR-600 (transceiver), NanoMind Z-7000 (processor/FPGA-unit) and the NanoDock SDR [5]. There is shielding added to the components and the size of the system is 22 x 16 x 5 mm. The RF capabilities are provided by the AD9361 transceiver that deals with the phase (I) and quadrature

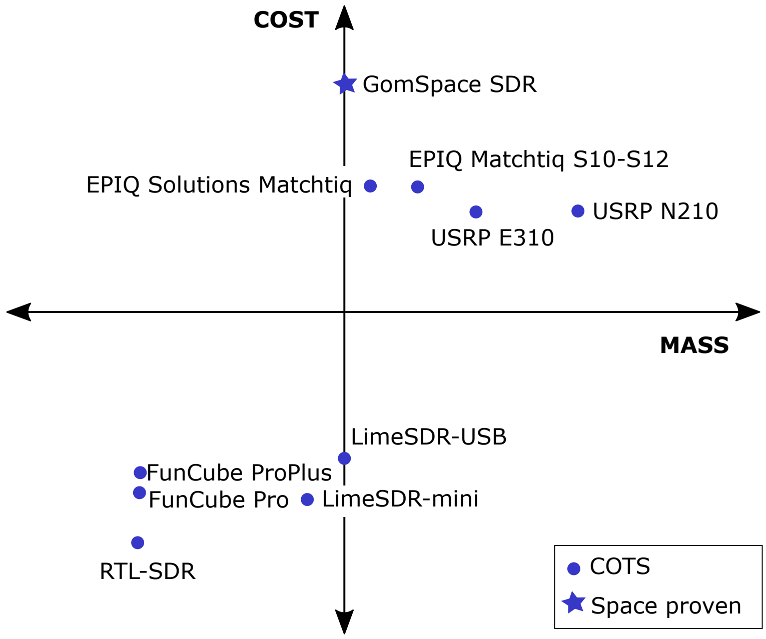

Figure 1: SDR platform overview (cost vs mass).

(Q) samples. Xilinx Zync 7030 SoC (System on Chip), which includes dual core ARM Cortex A9 processors and FPGA logic, performs the processing. However, there is no possibility of using GNURadio to program the transceiver at this moment. The noise figure of the NanoCom TR-600 receiver ranges from 5.1-7.8 dB and the power consumption is 3 W (idle).

Technical characteristics of AD9361 [6] are shown in table 1. The frequency range of this component is what limits the GomSpace SDR to 70 MHz - 6 GHz. In addition, AD9361 has two channels for MIMO (Multiple Input Multiple Output) and supports TDD (Time Division Duplex) and FDD (Frequency Division Duplex).

Features	**AD9361**
Transmitter frequency band	47-70 MHz
Receiver frequency band	70 MHz - 6 GHz
Channel bandwidth	200 KHz- 56 MHz
Noise figure	2 dB at 800 MHz LO
Operation modes	TDD and FDD

Table 1: AD9361 technical characteristics

2.2 USRP from Ettus Research

USRP E310 is part of the Embedded Series platform, which uses an OpenEmbedded framework and can be programmed with GNURadio [7]. The transceiver is also AD9361 and the processing unit is the Xilinx Zynq 7020 SoC (including dual core ARM Cortex A9 processors and a FPGA). The size of this SDR is 133 x 68 x 26.4 mm. The noise figure of the overall receiver is 8 dB and power consumption ranges from 2-6 W.

USRP N210 from Networked Series has a higher performance, as the Analogue to Digital Converters (ADCs) and Digital to Analogue Converters (DAC) have higher resolution and sample frequency, at the expense of increasing its mass and size (220 x 160 x 50 mm) [8]. It can also be programmed using GNURadio. The RF frontend consists of a daughterboard, and the processing unit is based on a Xilinx Spartan3-DSP. The frequency range is from DC to 6 GHz and the receiver has a noise figure of typically 5 dB.

2.3 EPIQ

EPIQ Solutions Matchstiq has a few SDR models, namely the S10-S12, with similar cost and characteristics as the USRPs [9]. It uses Xilinx Zynq 7020 SoC and the same transceiver as most of the SDRs, the AD9361, so it has same RF capabilities. The noise figure of the receiver is also 8 dB. Moreover, the size is 112 x 50.8 x 36.3 mm and the power consumption 2-6 W.

2.4 Lime microsystems

Lime microsystems offers two transceiver chips similar to Analog Devices one, the LMS6002D and LMS7002M. In Table 2 both transceivers are compared.

Features	LMS6002D	LMS7002M
Transmitter frequency band	47-70 MHz	30 MHz-3.8 GHz
Receiver frequency band	70 MHz-6 GHz	30 MHz-3.8 GHz
Channel bandwidth	0-28 MHz	Up to 48[1], 96 [2], 160 MHz[3]
Noise figure	3.5-10 dB	2-3.5 dB
Operation modes	TDD and FDD	TDD and FDD

Table 2: Lime microsystems transceivers technical characteristics

LimeSDR-USB has a smaller frequency band than the previous mentioned SDRs, from 0.1 MHz to 3.8 GHz, and a maximum bandwidth of 61.44 MHz [10]. It uses an Altera Cyclone IV EP4CE40F23C8N and LMS7002M transceiver chip (noise figure of 2-3.5 dB). The size is 60 x 100 mm. In addition, it can be programmed using GNURadio framework. Compared to the USRP embedded series and the Matchstiqs, this radio does not come with an integrated processor.

LimeSDR-mini is similar, also programmable with GNURadio, but has fewer features [11]. The frequency range is from 0.01 MHz to 3.5 GHz and the maximum bandwidth is 30.72 MHz. The RF transceiver is the same chip but the FPGA is Altera MAX 10 (10M16SAU169C8G). The main advantage is that is smaller, 69 x 31.4 mm and inexpensive.

2.5 FunCube

FunCube Pro is a very small SDR which uses a Silicon tuner as RF frontend and a PIC24FJ32 GB002 as microprocessor. The transmitter frequency range is from 0.64 MHz-1.1 GHz, whereas from receiving is from 1.27 - 1.7 GHz.

FunCube ProPlus is a similar SDR. It covers from 150 KHz-240 MHz and from 420 MHz to 1.9 GHz and has a maximum bandwidth of 56 MHz. Both FunCube models where made to support HAM radio satellite missions.

2.6 RTL SDR

RTL SDR is also a very small SDR, limited to receiving only. The frequency band covered is from 0.5 MHz to 1.766 GHz, with a maximum channel bandwidth of 2.4 MHz. It consists of a Rafael Micro R820T chip, a transceiver chip with 3.5 dB noise figure, and a digital modulator.

2.7 Others

There are many other SDRs available. BladeRF uses an Altera Cyclone IV and LMS6002D transceiver [12] and HackRF One uses an NXP microcontroller MAX2837 transceiver [13], both programmable using GNURadio. SWIFT has several SDR models with ongoing small satellites designs [14] such as SWIFT-UTX, SWIFT-SLX, SWIFT-WRX. Finally, AstroSDR [15] is a Space Plug-and-Play CubeSat SDR which uses Xilinx Zynq Z-7045 and the AD9362 transceiver, and therefore covers similar bands, and has a power consumption of 4-40 W.

In Figure 2 maximum channel bandwidth of the platforms mentioned above is plotted against frequency in a qualitative way. As it can be seen most of the SDRs work in many bands.

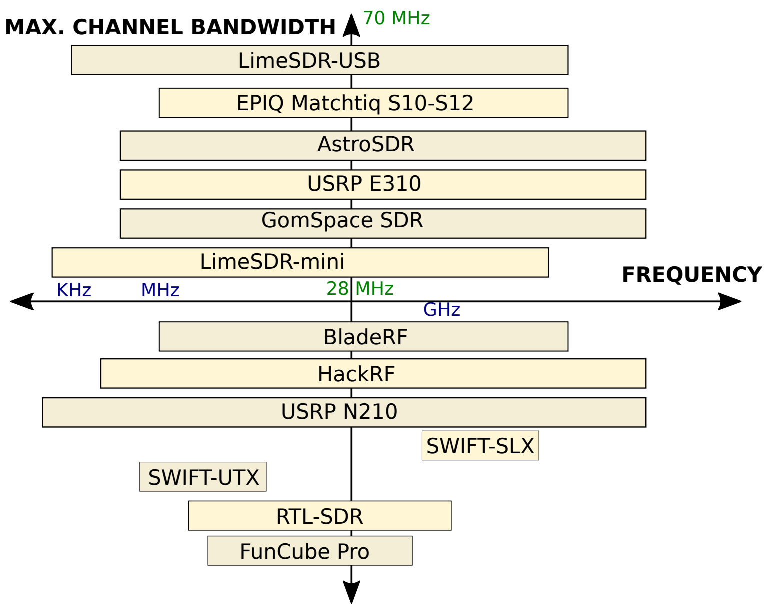

Figure 2: Maximum channel bandwidth vs frequencies covered in SDR platforms.

Another aspect to be considered when choosing an SDR is the software development tools. Some platforms are GNURadio compatible, like the USRP, Lime SDR, FUNCube, HackRF and others. This gives the users and developer access to a well-know open source ecosystem to base the software development on. This can be used both for ground stations and for the space segment. The USRPs also support the National Instruments LabView, thus a LabVIEW interface could be used to program them. However, due to the need for extra hardware and software to run LabView, this interface can reduce the development time but it can only be used in the ground station. One alternative would be to develop programs from scratch, for example in C for the microcontroller, or VHDL/Verilog for the FPGA. The programming language required for the SDR platform is definitely a factor to take into account.

3 SDR SURVEY

This section showcases a survey of SDRs that all have been designed and developed for satellite communications over the last years. We present an analysis of various hardware and software platforms. In the survey both ground stations and satellites using SDR are included.

3.1 Space Segment

The Aerospace Corporation and the University of Michigan-Flint have described a SDR design for their pipeline of small satellites, the Aerocubes [16]. The purpose of using this flexible technology was to increase the data throughput by using Adaptive Coding Modulation (ACM) technique and changing the error encoding characteristics and modulation depending on the channel variation. Simulations were carried out for a typical AeroCube pass over a ground station. By changing modulation (BPSK, QPSK, 16APSK, 32APSK) and code rates (from 1/4 to no encoding at all) without modifying the symbol rate, the throughput was increased by a factor of two if compared to QPSK at 1/2-rate coding. Looking at the SDR design, they adapted the firmware from an earlier SDR implementation and used a Zynq7020 board as the processing unit. The LMS6002D transceiver was used as RF frontend. The carrier frequency of the first generation design is 914 MHz (1 MHz of bandwidth), whilst the second generation transceiver will work on 26.1 GHz. The power consumption is 1.2 W when receiving and 2.5 W when transmitting 30 dBm.

Istanbul Technical University has also contributed to the development of space SDRs using Components Off-The-Shelf (COTS). In [17] two SDRs are described: one for the ground station and one for a CubeSat. The SDR for space is half-duplex and it is implemented in three boards: the transmitter, the receiver and a FPGA board, containing an Altera EP3C25E144I7N. It uses UHF Industrial Scientific and Medical (ISM) band, 433.92 MHz, and a 2FSK modulation. In addition, the power consumption is quite low; 2 W when transmitting and 0.7 W when receiving. The ground station SDR used two USRPs, a computer, one Low Noise Amplifier (LNA) and a power amplifier. Even though their project was carried out by undergraduate students, components for the CubeSat SDR were tested under space conditions. This small satellite is called HavelSat [18] and was launched in April 2017 [19].

University of Vigo in Spain and University of Porto [20] have been working for several years in the HumSat project, supported by the United Nations office for outer space affairs (UNOOSA), the European Space Agency (ESA) and the International Astronautical Federation (IAF). This project is a collaboration between multiple universities and centres, and its objectives are: to develop a data communications system for areas where there is not enough infrastructure for humanitarian purposes and to have sensors in remote areas. The SDR is a transmitter built on a board with a RF stage, a control stage and a power stage. The frequency ranges used from 440-470 MHz, with GMSK modulation and the power consumption when transmitting 30 dBm is 3.2 W. In stand-by mode the power consumption is 0.14 W. The first version of this SDR was launched in 2013 in a 1 unit CubeSat, called Xatcobeo. The next version is planned to be launched in December 2018.

Applied Physics Laboratory from Johns Hopkins University has built a TRL-9 SDR which has flown in the Van Allen Probes mission from NASA (National Aeronautics and Space Administration) with an S-band configuration (an X-band and Ka-band link can be possible too) [21], [22]. This SDR, called Frontier Radio, enables the possibility of changing to multiple modulation schemes, such as, BPSK, QPSK, PM/subcarrier for reception and up to 64PSK and 16QAM for transmission. Frontier radio is and FPGA-based design that uses RTAX4000 for the processing part and has different exciter slices depending on the frequency band used.

NASA has a huge interest in pushing SDR technology forward. Their objective is that an SDR may provide a flexible transceiver platform that can be tailored to several missions, just by changing software or hardware logic [3]. This is one of the reasons that can explain why there are several student satellites in the Educational Launch of Nanosatellites (ELaNa) programme planning to launch SDRs. For instance, LinkSat from Buffalo University; Space Hauc from University of Massachusetts; STF1 from West Virginia University and other member of a consortium; VCC A, B, C from Old Dominion University, Virginia Tech and University of Virginia; and OPEN from University of North Dakota. In addition, in 2012 NASA launched a Space Communications and Navigation (SCAN) testbed to provide with an on-orbit SDR facility. Earlier than same year, NASA published a paper which describes three different SDR developments for CoNNeCT (Communications, Navigation, and Networking reConfigurable Testbed) project [23]. In the first two cases the waveform and platform provider were General Dynamics and Harris. In the last case, JPL (Jet Propulsion Laboratory) and Cinnati Electronics developed the platform. Regarding frequency bands, Harris SDR was Ka-band and GD and JPL developed an S-band SDR. All of them using at least one Xilinx Virtex FPGA for the processing section and some radiofrequency (RF) converters and power amplifier for the RF frontend.

3.2 Ground Segment

SDRs are not only being utilised in the space segment, but also as part of ground stations.

University of Bologna built an SDR-based ground station suitable for ESA's European Student Earth Orbiter (ESEO) project [24]. As previously mentioned, SDRs enable the possibility of adding new waveforms by updating the software. Therefore, it is easier to update all ground stations in a network, just by sharing the updated software. In this development, the USRP N210 with an RF daughterboard is used as the SDR platform and the RF frontend. The ground station uses the UHF band, particularly radio-amateur frequencies (437 MHz for downlink and 435.2 MHz for uplink).

University of Surrey has focused on SDRs for concurrent multi-satellite communications. In [25] it has been developed a flexible system that can receive different types of signals of different satellites on a ground station using SDR technology. The transceiver board used is AD-FMCOMMS3-EBZ and for the processing part a Xilinx Zynq 7020 FPGA to achieve parallel architectures. The frequency band covered is limited by the transceiver, being 70 MHz - 6 GHz.

National Cheng Kung University is another university that has developed an SDR-based ground station to track small satellites [26]. The hardware used includes a ADLink PXI-3710 system controller and receiver blocks are implemented on Matlab/Simulink. Frequency bands considered are amateur VHF (140-150 MHz), UHF (430-440 MHz) and ISM band (2.4 GHz). Several bands can be received at the same time due to the implementation of an interference cancellation approach.

The Norwegian University of Science and Technology is also working on a GENSO-compatible station [27]. In addition to having developed an SDR-based ground station using a USRP2 and NGHam [28].

4 DISCUSSION

This study of SDR state-of-the-art comes from the need to use this technology to support several missions at the NTNU Small Satellite programme. In addition to set up a versatile SDR ground station, the main aim is to support science data collecting missions where there is poor communications infrastructure, like in the Arctic. To provide Arctic researchers with easier and faster access to scientific data harvested by sensor nodes, the payload should be flexible, so that physical retrieval of the data from sensors can be less frequent. In order to make better use of the resources available (bit rate, power, link properties, timing and delay, and the amount of data), the payload should be re-configurable and adaptable in-flight. This is where SDR technology comes into play.

The SDR payload is meant to be an experimental system with two purposes: The first is demonstrating re-programming of the SDR in-flight, the second is to demonstrate simple Adaptive Coding and Modulation (ACM) capabilities. Employing ACM, the bit rate and modulation can change within one pass, or at least between passes, based on the predicted "quality" of the pass. The re-programmable features can comprise a selection of frequency bands, channel bandwidths, bit rates, modulation and power levels. Depending on available frequency bands (for uplink and/or downlink), the payload should support at least two frequency bands; for example VHF or UHF, and L-band or S-band. The SDR alternatives available support two separate RX/TX paths, so each antenna system can be individually mated to one RXTX interface. If more than a frequency bands are required, then filter banks/diplexer must be used.

Three main options considered for the payload design are:

1. Buy and integrate space proven hardware platforms, such as GomSpace SDR.
2. Buy and integrate no-space proven COTS hardware (URSPs, LimeSDR, EPIQ Solution SDRs,...)
3. Make an in-house design and integration of a custom SDR. Based on the AD9361 transceiver chip and an FPGA, for example.

On the one hand, the first option is safe for the mission. However, it is very expensive. Also, buying a complete SDR implies less control of the mission. On the other hand, making a custom design would increase the team's knowledge of SDR and enable full control of the SDR. Most universities in this study have done that, but it is less reliable as the components are not space-qualified and the system has to be developed from scratch. It seems like the second option is the best compromise. A trade-off study will be carried out to help decide which design approach is going to be followed.

Another important aspect is how to design the RF front-end. In order to be able to communicate using multiple bands, both the SDR hardware platform and also the RF front-end must support the bands. More than one antenna will be needed to receive both UHF and S-band. This means that a diplexer is needed between the SDR and the antennas. The SDR platforms usually have an internal LNA, but an additional one may be needed. In this case, there are two possibilities: to use a broadband LNA (designed for both bands) or two different LNAs, one for each band. Using multiple bands can add complexity and cost to the system but enhances communication (enabling different data rates and providing redundancy, for instance), therefore a trade-off analysis must be made.

Most of the radios presented in Section 2 can be used as part of a ground station design, as long as it fits the frequency bands of the mission. Since one usually has access to computers at the ground station, the fully embedded solutions (USRP E-series, Matchstiq and HackRF) might not be desirable, as it will be easier to work on a regular computer both during development and operations.

For the space segment, the opposite is true. In this case, both size and power are major concerns. Therefore, highly integrated embedded solutions are preferable. This can point in the direction for the USRP E-series or the Matchstiqs. The Lime SDR could also be used, however it must be integrated with a processor running Linux. These radios can be used in a *hybrid* COTS solution. The processor is included in the USRP E and Matchstiq. It is important that the radio chosen has a good quality, is frequency stable and have good RFI (radio frequency interference) and EMC (ElectroMagnetic Compatibility) properties.

5 CONCLUSION

This survey attempts to give insight into SDRs for small satellites and ground stations. It became clear that it is not an easy task to compare the platforms, because not all of them provide the information needed for a coherent analysis. It is also challenging to find information about university projects and figure out they have launched the satellites described in their papers. Nevertheless, there is no doubt that a lot of research groups have worked on developing space SDRs. There are already a lot of small satellites using this technology for science applications, radio measurements, navigation, communications and technology demonstrators.

Most of the university projects studied seem to use a custom SDR solution based on FPGAs. The Zynq board and the AD9361 transceiver chip from Analog Devices are very commonly used in these implementations. The transceiver chips from Lime microsystems has also been utilised a few cases. This suggests to conclude that the AD9361 can be used to reduce the risk. The component has been flown in space several times in different SDRs implementations.

Choosing an SDR platform depends on many factors and the risk of component failure is very important to consider. GomSpace SDR seems to be the safest choice. Nevertheless, it is the most expensive one and the team working with would not have so much control over the hardware nor software. The URSPs can be a reasonable choice, especially for the ground station segment. There are no requirements for size or weight, and experience with these platforms are available.

EPIQ Solutions Matchtiq and LimeSDR could also be considered both for the space and ground segments. However the LimeSDR must be integrated with an external processor capable of running Linux. FunCube and RTL-SDRs can be used for the first time because they are inexpensive, but may be limiting the performance of a production system. HackRF and BladeRF are interesting platforms but have not been used in space so far. SWIFT SDRs seem very attractive, but there is no enough information about them available. Finally, AstroSDR is an SDR designed for CubeSat but its power consumption may be too much for a 3 unit (3U) or 6U CubeSat.

The need to develop flexible satellite communications systems, particularly for small satellites, has been described. Different hardware implementations of this technology have been highlighted and their technical characteristics have been explained. In addition, a survey on SDR technology developments by several universities was carried out. Their mission or goal was described, as well as the main components used and the radio parameters of their design. Finally, how to approach an SDR development for NTNU's communications mission was briefly discussed.

6 REFERENCES

[1] How much water is there on earth, from the usgs water science school. [Online]. Available: https://water.usgs.gov/edu/earthhowmuch.html

[2] R. Birkeland, "Freely drifting cubesat constellations for improving coverage for arctic sensor networks," in *Communications (ICC), 2017 IEEE International Conference on.* IEEE, 2017, pp. 1–6.

[3] NASA, "Small Spacecraft Technology State of the Art," no. February, pp. 1–197, 2014. [Online]. Available: http://www.nasa.gov/sites/default/files/files/Small{_}Spacecraft{_}Technology{_}State{_}of{_}the{_}Art{_}2014.pdf

[4] I. Simms, William Herbert, K. Varnavas, and E. Eberly, "High speed, low cost telemetry access from space development update on programmable ultra lightweight system adaptable radio (pulsar)," August 2014. [Online]. Available: http://hdl.handle.net/2060/20140012879

[5] "Gomspace — software defined radio," https://gomspace.com/Shop/payloads/software-defined-radio.aspx, (Accessed on 04/13/2018).

[6] "Ad9361 datasheet and product info — analog devices," http://www.analog.com/en/products/rf-microwave/integrated-transceivers-transmitters-receivers/wideband-transceivers-ic/ad9361.html, (Accessed on 04/13/2018).

[7] Ettus Research, “Usrp e310 embedded sdr,” Online, 2015, accessed 16.12.2015. [Online]. Available: http://www.ettus.com/content/files/USRP_E310_Product_Sheet.pdf

[8] “Usrp n210 software defined radio (sdr) - ettus research,” https://www.ettus.com/product/details/UN210-KIT, (Accessed on 04/13/2018).

[9] “Matchstiq sdr platform,” https://www.epiqsolutions.com/matchstiq/, (Accessed on 04/13/2018).

[10] “Software defined radio - lime micro,” http://www.limemicro.com/products/software-defined-radio/, (Accessed on 04/13/2018).

[11] L. Microsystems, “Lime sdr,” online, 2017. [Online]. Available: https://www.crowdsupply.com/lime-micro/limesdr

[12] “Nuand — bladerf software defined radio,” https://www.nuand.com/, (Accessed on 04/13/2018).

[13] “Hackrf one · mossmann/hackrf wiki · github,” https://github.com/mossmann/hackrf/wiki/HackRF-One, (Accessed on 04/13/2018).

[14] “Swift software defined radios for cubesats,” http://www.tethers.com/SWIFT.html, (Accessed on 04/13/2018).

[15] “Astrosdr – rincon research corporation,” https://www.rincon.com/shop/space-systems/astrosdr-family/, (Accessed on 04/13/2018).

[16] E. Grayver, A. Chin, J. Hsu, S. Stanev, D. Kun, and A. Parower, “Software defined radio for small satellites,” in *2015 IEEE Aerospace Conference*, March 2015, pp. 1–9.

[17] O. Ceylan, A. Caglar, H. B. Tugrel, O. Cakar, A. O. Kislal, K. Kula, and H. B. Yagci, “Satellites,” *IEEE microwave magazine*, no. March, pp. 26–33, 2016.

[18] E. Baceski, S. Gökçebağ, A. Erdem, C. G. Erbay, M. Akyol, K. Arslankoz, . Arslan, M. A. Ağca, Y. B. Aydın, A. R. Aslan, and O. Ceylan, “Havelsat: A software defined radio experimentation cubesat,” in *2015 7th International Conference on Recent Advances in Space Technologies (RAST)*, June 2015, pp. 831–834.

[19] “Gunter’s space page - information on spaceflight, launch vehicles and satellites,” http://space.skyrocket.de/, (Accessed on 04/13/2018).

[20] A. G. C. Guerra, A. S. Ferreira, M. Costa, D. Nodar-López, and F. A. Agelet, “Integrating small satellite communication in an autonomous vehicle network: A case on oceanography,” https://arxiv.org/pdf/1710.10977.pdf, (Accessed on 04/13/2018).

[21] M. P. Angert, B. M. Bubnash, R. J. Hearty, M. B. O’Neill, S. X. Ling, D. E. Matlin, and S. Cheng, “Advancements in hardware design for the frontier radio used for the solar probe plus mission,” in *Aerospace Conference, 2017 IEEE*. IEEE, 2017, pp. 1–11.

[22] C. B. Haskins, M. P. Angert, E. J. Sheehi, W. P. Millard, N. Adams, and J. R. Hennawy, “The Frontier Software-Defined Radio for the Solar Probe Plus Mission,” pp. 1–11, 2016.

[23] S. K. Johnson, R. C. Reinhart, and T. J. Kacpura, “CoNNeCT’s approach for the development of three Software Defined Radios for space application,” *2012 IEEE Aerospace Conference*, pp. 1–13, 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6187147

[24] M. Bosco, P. Tortora, and D. Cinarelli, “Alma Mater Ground Station transceiver: A software defined radio for satellite communications,” *2014 IEEE Metrology for Aerospace (MetroAeroSpace)*, pp. 549–554, 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6865986/

[25] M. R. Maheshwarappa, “Software defined radio (sdr) architecture for concurrent multi-satellite communications,” Ph.D. dissertation, University of Surrey, 2016.

[26] J. C. Juang, C. T. Tsai, and J. J. Miau, “A software-defined radio approach for the implementation of ground station receivers,” *Small Satellites for Earth Observation*, pp. 293–298, 2008.

[27] Øyvind Karlsen, “Ground station considerations for the amos satellite programme,” master thesis, Norwegian Univesity of Science and Technology (NTNU), 2017. [Online]. Available: https://brage.bibsys.no/xmlui/handle/11250/2462967

[28] A. Løfaldli and R. Birkeland, “Implementation of a software defined radio prototype ground station for cubesats,” in *Proceedings of the ESA Small Satellites and Services Symposium*, 2016.