Summer 2018

Developing a Program for Calibrating and Streaming from a Software-Defined Radio

Amanda Kolbeck
Montana Tech

Kevin Negus
Montana Tech

Follow this and additional works at: https://digitalcommons.mtech.edu/urp_aug_2018

Recommended Citation
https://digitalcommons.mtech.edu/urp_aug_2018/1

This Book is brought to you for free and open access by the Other Undergraduate Research at Digital Commons @ Montana Tech. It has been accepted for inclusion in 2018 Undergraduate Research by an authorized administrator of Digital Commons @ Montana Tech. For more information, please contact sjuskiewicz@mtech.edu.
Developing a Program for Calibrating and Streaming from a Software-Defined Radio

Amanda Kolbeck (Montana Tech) with mentor Kevin Negus (Montana Tech)

Results with a Carrier Frequency of 1250 MHz

The board connects via USB 3.0 and has RF transceivers, a field-programmable gate array (FPGA), and microcontroller chipsets.

LimeSDR Hardware

Background and Significance

Radio transmission is a valuable tool for communicating in areas where common signals can’t reach, and software-defined radios (SDRs) are the next step in radio communication.

SDRs are remote access radio systems that utilize software rather than hardware to transmit and receive radio signals.

SDRs provide a new level of customizability in radio transmission, allowing faster and easier alterations to be made to transmitting and receiving channels.

Methods

- Obtained a LimeSDR, a software-defined radio developed by Lime Microsystems.
- Developed a Python code using SoapySDR, a vendor-neutral library with a Python API.
- Created lower sideband, upper sideband, and double sideband signals in MATLAB using I/Q data.
- Played the signals and observed them on a spectrum analyzer.
- Calibrated the signals from the LimeSDR with the Python code.

Conclusions

- Signals transmitted from the LimeSDR can be successfully calibrated and tuned with a user-developed code.
- Calibration changes can be made quickly and efficiently when the LimeSDR is interfaced with computer software.
- The developed code can be further expanded upon to calibrate received signals.
- A collaborative online environment provides countless possibilities and applications for this hardware.

Acknowledgements

This work was supported by Montana Tech’s Summer Undergraduate Research Fellowship (SURF).