**Software/Signal Processing**

This document outlines the basic behavior of the software components of the radar project. The software is responsible for the management of a collection of inputs and outputs: the VCO control waveform, the VCO enable signal, and the mixed-down receive waveform. It uses those inputs and outputs to calculate the range and speed of an object. To accomplish this, the software has been divided into four basic modules: ADC, ADC processing, DAC, and GUI.

The ADC and ADC processing modules are closely interrelated. To ensure a consistent sample rate, the ADC module uses a hardware peripheral of the MKL26Z4 microcontroller on the teensy board called the periodic interrupt timer (PIT) to generate a constant clock signal at the desired sampling frequency. The ADC can then be configured to start a conversion on every rising clock edge of the PIT. This allows consistent conversion with no processor intervention. To close the loop, the direct memory access (DMA) module is configured to automatically transfer data from the ADC to a 512 byte buffer whenever a new reading is available. This allows the firmware to trigger a reading of 512 12-bit words from the ADC at a constant sample rate with only one function call to trigger the required peripherals.

The ADC processing module uses the ADC module’s API to trigger readings and then process the returned data. To determine the frequency of the input waveform, some basic processes are used. First, the module loops over the collected data and determines the average value. Then, the module subtracts that average from every sample. The module then loops over the data to find zero crossings, recording the index of each crossing found. Finally, the difference between each recorded index is averaged together and doubled to calculate the average period (in samples) of the collected waveform. The received frequency can then be determined by dividing the sampling rate by the average period in samples.

The DAC module facilitates control of the transmit stage of the radar. It outputs a constant voltage or a variable voltage waveform to control the output frequency of the VCO. Currently, the DAC is configured to output a set voltage to keep the VCO at a continuous frequency. This is important for use as a doppler radar. The user can configure the voltage output during runtime to change the base frequency of the radar, but sweeps cannot yet be configured. The DAC module is planned to allow the user to select the range and period of the sawtooth wave used to sweep the VCO for FMCW operation.

Finally, the GUI module glues together all the underlying modules and handles all user interaction. It displays a splash screen on an LCD with the company logo upon boot and then switches to the main menu page after a timeout or button press. The user can then use the hardware buttons to enable or disable the radar, vary the frequency of operation, or force the radar to operate in one-shot mode. In one-shot mode, the radar will only capture and process one 512-byte block of data. This would allow the user to point the radar at an object of interest and capture a single speed or distance measurement.

The firmware is maintained on Github: <https://github.com/GenericRadarOrganization/firmware>. It requires that the user has the arm-none-eabi flavor of GCC installed locally as well as the teensy loader CLI for loading the firmware. To build and flash the firmware, the command “make burn” must be run in the root directory of the project.