Rapid Development of Software Defined Radio: FMCW Radar on Zynq SDR

Kenneth Barlee, Professor Robert Stewart, Dr Louise Crockett

Institute of Sensors Signals & Communications, Electronic and Electrical Engineering, University of Strathclyde, Glasgow

{ kenneth.barlee, r.stewart, louise.crockett } @strath.ac.uk

University of Strathclyde Engineering

Background and Motivation

Software Defined Radio (SDR) aims to have greater flexibility than traditional radio systems by shifting parts of the radio architecture from physical analogue circuitry to the digital domain. While the digital processing could be carried out solely using a General Purpose Processor (GPP), the latest development tools provide the facility to partition designs and target Field Programmable Gate Array (FPGA) hardware. FPGAs are inherently suited to the kinds of operations required in SDR transmitters and receivers; such as mod/demodulation, filtering and synchronisation. Combining programmable hardware with software permits complex radio systems to be rapidly prototyped and deployed from the desktop.

FMCW Radar is a relatively simple radar technology. Here, an FMCW chirp is transmitted, bounces off a surface and reflects back to the receive antenna. The received signal is out of phase with the transmitted signal, due to the additional propagation time. The time difference between the Transmit (Tx) and Receive (Rx) chirps is directly proportional to the distance travelled (distance-speed-time), and by calculating what the time difference is, the propagation distance can be estimated. A standard use case for FMCW radar is Adaptive Cruise Control.

The **Coffee Can Radar** project was originally developed by academics at MIT [1]. As part of a radar course, it aims to have students build FMCW radars from $100 worth of analogue components that are capable of estimating range. These radars do not work in real time, as the received signals need to be processed offline in MATLAB.

Using this as a starting point, work was carried out to develop a similar system that could operate in real time using only SDR equipment. A Zynq ZC706 development board was chosen for this task, along with an FMCOMMs 3 radio front end. Details about these components and the development cycle are given in the following sections.

Figure 1: Adaptive Cruise Control based on FMCW Radar technology, operating in the 77GHz band

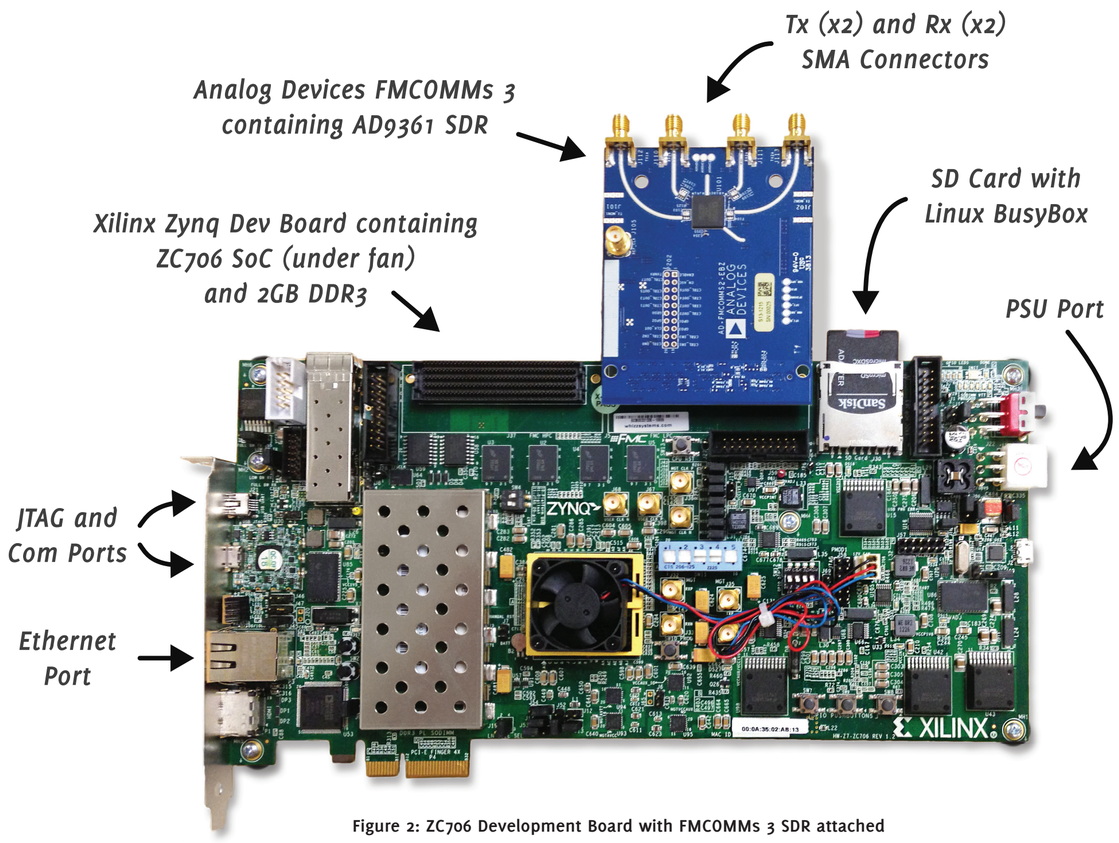


Figure 2: ZC706 Development Board with FMCOMMs 3 SDR attached

Zynq & FMCOMMs SDR MathWorks Support

The development board shown here contains a ZC706 System on Chip (SoC) [2]. This is from a family of devices called Zynq, and was produced by Xilinx. The SoC contains both a dual core ARM processor and large FPGA. The combination of these two units on one piece of silicon permits hardware acceleration to be performed in a stand-alone system.

Attached to the dev board via an FMC connector is an FMCOMMs 3 [3]. Produced by Analog Devices, this high end 2x2 SDR contains an AD9361 transceiver, which operates between 70MHz – 6GHz, has a maximum bandwidth of 56MHz and sampling rate of 61.44MHz.

MathWorks have produced a *Zynq SDR* Hardware Support Package [4], which provides the tools to allow users to interface with Zynq based dev boards and FMCOMMs Radios from within MATLAB and Simulink. From within their model based environment, their tools allow you to partition designs, and target parts to the FPGA fabric while generating the remainder into software for the ARM. This means you can develop deployable Software Defined Radio systems from the desktop.

FMCW ZC706/ FMCOMMs 3 CANtenna Radar

Initially, a simulation model of the radar was constructed that 'transmitted' through a Free Space channel model that was available in the MATLAB Phased Array Toolbox library. Satisfied it was working correctly, the design was partitioned into *hardware* and *software* subsystems. The hardware subsystem utilised Simulink blocks from the HDL Coder library (all of which were hardware compatible), while the software subsystem used Embedded Coder blocks.

Next, HDL code was generated using the SDR Workflow Advisor. This was run through Xilinx's Vivado to create a bitstream for the FPGA part of the Zynq SoC. Once this was targeted onto the Zynq, the remainder of the design underwent a C code generation process to produce software for the BusyBox operating system on the ARM.

Inkeeping with the original project, CANtennae were used to transmit and receive the signals. These work best at 2.4GHz, so 2.4GHz was chosen as the centre frequency. The chirp was configured to be 56MHz wide, and have a period of 50µs. With the sampling rate of 61.44MHz, the radar system had a range resolution of 2.67m. While this was not ideal, it was the minimum resolution possible with the hardware max-ed out! The completed design was capable of estimating the distance between its CANtennae and the surface they were pointed at, while updating a spectrogram on the host computer with this information in real time.

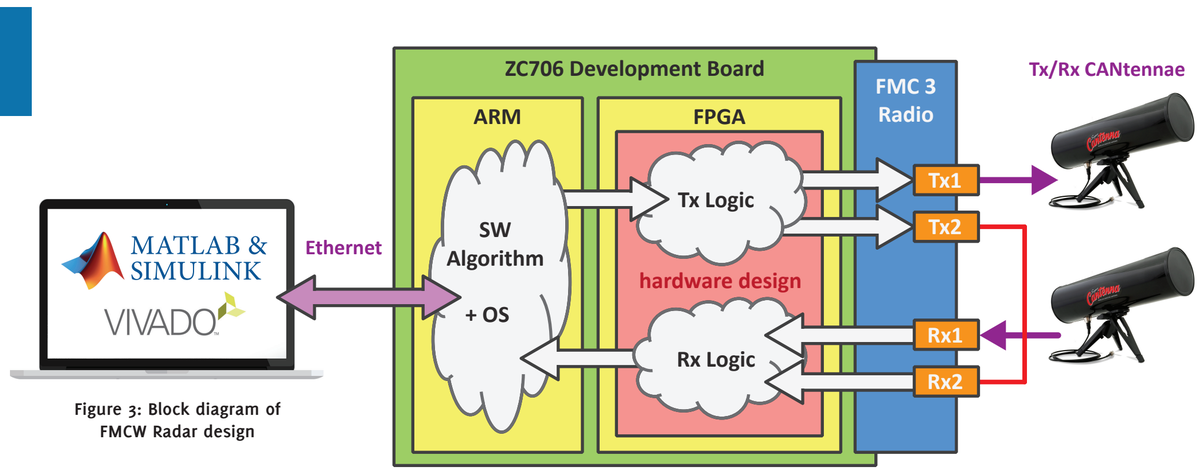
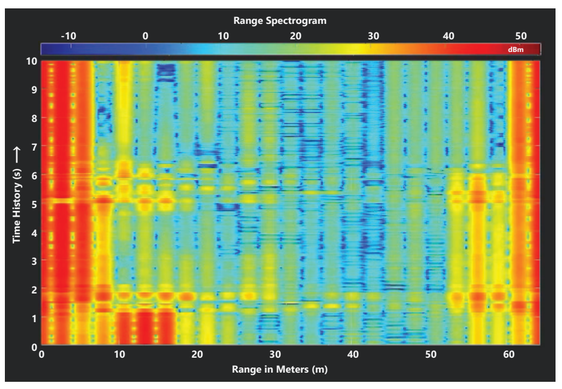


Figure 3: Block diagram of FMCW Radar design



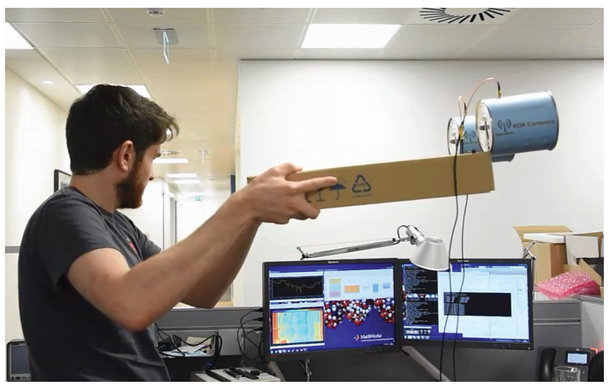


Figure 4: Screenshots of the radar outputs when pointing the CANtennas around the office

(Darker red colours indicate where most of the spectral energy is. In this screenshot, the chirps were being transmitted a distance of around 13m, which is why there is a high powered band of energy around this point in the spectrogram)

MathWorks® XILINX® ZYNQ ANALOG DEVICES cidcom Centre for Intelligent Dynamic Communications

References:

[1] - http://bit.ly/1eSSpwJ

[2] - http://bit.ly/1XSiRcH

[3] - http://bit.ly/26oGAHe

[4] - http://bit.ly/24fDqU9