README- App 2 – Radar Sample App (Ethernet)

The functions in this folder provide samples of MATLAB code for implementing an interface to the PulsON 400 (P400) Monostatic Radar Module (MRM). The MRM Application Programming Interface (API) Specification describes the complete set of messages supported by MRM.

The interface uses Ethernet User Datagram Protocol (UDP) packets implemented by instantiating Java classes in MATLAB. It does not require any special MATLAB toolboxes.

These basic examples include using Java classes to create a UDP socket, create UDP packets, and send and receive UDP packets. The code also contains examples of integer data representation, handling byte order, error handling, and other concepts.

This is a brief description of the basic MRM communication functions included:

- sckt_mgr.m - hides java open/close UDP socket code. Notice the persistent SCKT variable.
- str2dat.m - converts a structure to data.
- get_cfg_rqst.m - sends a MRM_GET_CONFIG_REQUEST message (see MRM API.)
- set_cfg_rqst.m - sends a MRM_SET_CONFIG_REQUEST message (see MRM API.)
- parse_msg.m - parses MRM responses into a structure. Currently only operates on
- MRM_GET_CONFIG_CONFIRM, MRM_CONTROL_CONFIRM, and MRM_SCAN_INFO structures.
- ctl_rqst.m - sends a MRM_CONTROL_REQUEST message to the MRM (see MRM API.)
- read_pckt.m - waits TIMEOUT ms for a response from the MRM. Returns response.
- The chng_cfg_xmpl is a good starting example for understanding the code. This script gets the configuration, changes a parameter value, and sets the configuration.

The reader is referred to MATLAB documentation regarding the use of Java classes within MATLAB. Other than the use of Java classes, most of the code should be familiar to a typical MATLAB programmer.

MRMDemo.m was added as a simple radar application script that makes use of the MATLAB functions listed above.   Type 'help MRMDemo' or view MRMDemo.m for more information on this script.

```
% MRM Demo
% Hand-held Radar
%
% This demo illustrates direct access from MATLAB to the MR through
% the MRM API and UDP sockets.
%
% This code was initially developed by senior design students at the University
% of Alabama in Huntsville (UAHuntsville) and targeted at a
% hand-held radar.
%
% This code should work on any PC running MATLAB.  It has been tested
% against Windows7 and Mac OSX running MATLAB.  We used a UWB horn
% antenna to improve handheld directionality.
%
% INSTRUCTIONS:
% - Connect MRM using an ethernet cable.
% - Power up and test the connection using 'ping' in a cmd window.
% - Adjust the mrmIpAddr below to match that of your MRM.
% - Execute this script from the MATLAB command prompt.
% - Any the change in reflectivity relative to the first scan will be
%   displayed.
%
% This script illustrates:
% 1. Connecting to the MRM and querying configuration using the
% MRM_GET_CONFIG_REQUEST command (see MRM API.)
% 2. Parsing the MRM_GET_CONFIG_CONFIRM structure with the big-endian
% byte swap required on INTEL processors.
% 3. Adjusting the CONFIG and setting start, stop, and integration based on
% hand-held radar preferences using the MRM_SET_CONFIG_REQUEST message.
% 4. Loop, commanding, receiving, processing, and plotting one radar scan
% at a time.
% 5. In the included algorithm the INITIAL SCAN is used as a template and
% subtracted from all following scans.  Each "residual" scan is rectified
% and low-passed to produce an envelope.  This envelope is thresholded
% to produce a detection list.
%
% NOTE: The algorithm demonstrated is targeted at an Android tablet
% with modest communication and processing capability.  Therefore we
% rather than commanding the MRM to continuous scan mode (which could
% overburden our host) our main loop iteratively requests a single scan
% from the MRM after which we process and display before requesting
% another.
% This adds ~2ms of jitter to our scan timing but since we aren't using
% a motion filter (we use the first scan as a reference template) scan
% timing isn't so important.
```