**static** int32\_t **SRR\_CfarFovCfg** (int32\_t argc, **char**\* argv[])

{

CFARCAProc\_FovCfg\_t fovCfg;

uint32\_t procDirection;

int32\_t retVal;

/\* Initialize configuration: \*/

**memset** ((**void** \*)&fovCfg, 0, **sizeof**(fovCfg));

/\* Populate configuration: \*/

procDirection = (uint32\_t) **atoi** (argv[2]);

fovCfg.min = (**float**) **atof** (argv[3]);

fovCfg.max = (**float**) **atof** (argv[4]);

retVal = **rlSetcfarFov**(RL\_DEVICE\_MAP\_INTERNAL\_BSS, 1U, &fovCfg);

**if** (retVal != RL\_RET\_CODE\_OK)

{

System\_printf ("Error: Unable to configure the cfarFov [Error %d]\n", retVal);

**return** -1;

}

**return** 0;

}

rlReturnVal\_t **rlSetcfarFov**(rlUInt8\_t deviceMap, rlUInt16\_t cnt, CFARCAProc\_FovCfg\_t\* data)

{

rlReturnVal\_t retVal;

/\* check if deviceIndex is out of defined value \*/

**if** ((**rlDriverIsDeviceMapValid**(deviceMap) != RL\_RET\_CODE\_OK) || \

(RL\_NULL\_PTR == data) || (cnt == 0U))

{

/\* set return error code \*/

retVal = RL\_RET\_CODE\_INVALID\_INPUT;

}

**else**

{

/\* Initialize Command and Response Sub Blocks \*/

rlDriverMsg\_t inMsg;

rlDriverMsg\_t outMsg = {0};

/\* Initialize Command and Response Sub Blocks \*/

rlPayloadSb\_t inPayloadSb[RL\_CMD\_PL\_LEN\_MAX/(**sizeof**(CFARCAProc\_FovCfg\_t) + \

RL\_SBC\_ID\_SIZE + RL\_SBC\_LEN\_SIZE)];

retVal = RL\_RET\_CODE\_OK;

/\* Fill in-message packet \*/

**rlDriverConstructInMsg**(RL\_RF\_DYNAMIC\_CONF\_SET\_MSG, &inMsg, &inPayloadSb[0]);

retval = **rlDriverExecuteSetApi**(deviceMap, RL\_RF\_STATIC\_CONF\_SET\_MSG,

RL\_RF\_CHAN\_CONF\_SB, (rlUInt8\_t\*)data,

(rlUInt16\_t)**sizeof**(rlChanCfg\_t));

**return** retVal;

}

rlReturnVal\_t **rlDriverExecuteSetApi**(rlUInt8\_t deviceMap, rlUInt16\_t msgId,

rlUInt16\_t sbcID, rlUInt8\_t \*msgData, rlUInt16\_t inLen)

{

/\*LDRA waiver 8 D - DD data flow anomalies found- \*/

rlReturnVal\_t retVal;

/\* Initialize in-message structure to zero \*/

rlDriverMsg\_t inMsg = {0};

/\* Initialize out-message structure to zero \*/

rlDriverMsg\_t outMsg = {0};

/\* Initialize in-payload sub-block structure to zero \*/

rlPayloadSb\_t inPayloadSb = {0};

/\* Construct command packet \*/

/\* AR\_CODE\_REVIEW MR:R.2.2 <INSPECTED> "Values are updated in other function.

LDRA Tool Issue" \*/

/\*LDRA\_INSPECTED 8 D \*/

rlDriverConstructInMsg(msgId, &inMsg, &inPayloadSb);

/\* Fill in-message Payload \*/

/\* AR\_CODE\_REVIEW MR:R.2.2 <INSPECTED> "Values are used by called function.

LDRA Tool Issue" \*/

/\*LDRA\_INSPECTED 105 D \*/

rlDriverFillPayload(msgId, sbcID, &inPayloadSb, msgData, inLen);

/\* Send Command to mmWave Radar Device \*/

/\* AR\_CODE\_REVIEW MR:D.4.1,D.4.14 <INSPECTED> "Can't be NULL" \*/

/\*LDRA\_INSPECTED 45 D \*/

retVal = rlDriverCmdInvoke(deviceMap, inMsg, &outMsg);

**return** retVal;

}