V0.1 11/20/20

- Performs single shot GPADC read. User needs to modify for their desired channel
- GPADC measurement not yet verified. Runs without error.

Installation & Build

- Install mmWave SDK 3.5.0.4
- Unzip mss_gpadc and replace mss folder in C:\ti\mmwave_sdk_03_05_00_04\packages\ti\demo\xwr68xx\mmw\mss
- Follow build instructions as specified in SDK User's Guide

Requirements

- GPADC reads from MSS cannot occur while run time calibrations are enabled
  - Standard disable of runtime calibrations(does not include APLL & SYNTH calibrations)

```
3599    calibrationCfg.dfeDataOutputMode = gMmwMssMCB.cfg.ctrlCfg.dfeDataOutputMode;
3600    calibrationCfg.u.chirpCalibrationCfg.enableCalibration    = false;/*For MSS GPADC: Disable runtime calib*/
3601    calibrationCfg.u.chirpCalibrationCfg.enablePeriodicity    = false;/*For MSS GPADC: Disable runtime calib*/
```

  - Use additional mmWaveLink API to disable APLL & SYNTH calibrations (this is not normally exposed, as it is recommended to allow BSS to trigger these calibrations every 1sec). User assumes risk of disabling these calibrations and needs to determine if/when calibrations need to be resumed in order to maintain necessary RF quality

```
3450        /*For MSS GPADC: Must disable APLL & SYNTH runtime calib.*/
3451 #if 1
3452        rfCalDisabledata.funcCfg.bits.b1CalDisGpadc = 1;
3453        rfCalDisabledata.funcCfg.bits.b1ApllCalDisRuntime = 1;
3454        rfCalDisabledata.funcCfg.bits.b1Synth1CalDis = 1;
3455        rfCalDisabledata.funcCfg.bits.b1Synth2CalDis = 1;
3456        rfCalDisabledata.funcCfg.bits.b1PDTrimEfuseDis = 1;
3457        retVal = rlRfSetCalibDisableConfig(RL_DEVICE_MAP_INTERNAL_BSS, &rfCalDisabledata);
3458        Task_sleep(5); //Delay to enable any calib that was triggered prior to disable to complete
3459
3460        if(retVal != 0)
3461        {
3462            System_printf("Error: rlRfSetCalibDisableConfig retVal=%d\n", retVal);
3463            return -1;
3464        }
3465 #endif
```

Usage

- To perform read: `call BSS_gpadc_read(uint32_t paramLUTentry);`

```
3467        /*For MSS GPADC: Perform gpadc read*/
3468        uint32_t    gpadcStartTime, gpadcElapsedTime;
3469        gpadcStartTime = Cycleprofiler_getTimeStamp();
3470        gpadcValue = BSS_gpadc_read(0); //pass in the index to z_GpAdcParamLut[] based on desired signal/mux
3471        gpadcElapsedTime = (Cycleprofiler_getTimeStamp() - gpadcStartTime)/R4F_CLOCK_MHZ;
3472        System_printf ("GPADC result: avg %d min %d max %d sum %d time[us] %d\n",
3473                gpadcValue.h_Avg, gpadcValue.h_Min, gpadcValue.h_Max, gpadcValue.w_Sum, gpadcElapsedTime);
3474
```

  paramLUTentry index corresponds to below and is used to specify which channel:

```
T_GPADC_BIST_PARAM z_GpAdcParamLut[11] = {
    /*! ConfigVal, {ParamVal, CollectSamples, SkipSamples, ChirpBreak, Reserved}, time, min, max */
    {   0x400U,  { 124U, 4U,  8U, 0U, 0U }, 74U, 0U, 0U }, /* #13 M_GPADC_MEAS_EXT_ANATEST1 */
    {   0x800U,  { 124U, 4U,  8U, 0U, 0U }, 74U, 0U, 0U }, /* #14 M_GPADC_MEAS_EXT_ANATEST2 */
    {  0x1000U,  { 124U, 4U,  8U, 0U, 0U }, 74U, 0U, 0U }, /* #15 M_GPADC_MEAS_EXT_ANATEST3 */
    {  0x2000U,  { 124U, 4U,  8U, 0U, 0U }, 74U, 0U, 0U }, /* #16 M_GPADC_MEAS_EXT_ANATEST4 */
    {  0x4000U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #17 M_GPADC_MEAS_EXT_ANAMUX */
    {  0x8000U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #18 M_GPADC_MEAS_EXT_VSENSE */
    {   0x002U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #19 M_GPADC_MEAS_EXT_ANATEST1BUF */
    {   0x004U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #20 M_GPADC_MEAS_EXT_ANATEST2BUF */
    {   0x008U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #21 M_GPADC_MEAS_EXT_ANATEST3BUF */
    {   0x010U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }, /* #22 M_GPADC_MEAS_EXT_ANATEST4BUF */
    { 0x10000U,  { 124U, 4U, 10U, 0U, 0U }, 74U, 0U, 0U }};/* #23 M_GPADC_MEAS_EXT_ANAMUXBUF */
```

Output

- To view GPADC print statement connect to CCS debug and load debug binaries. Time is elapsed time for call to BSS_gpadc_read

```
[C674X_0] Debug: DPM Module Sync is done
[Cortex_R4_0] ******************************************
Debug: Launching the MMW Demo on MSS
******************************************
Debug: Launched the Initialization Task
Debug: mmWave Control Initialization was successful
Debug: mmWave Control Synchronization was successful
Debug: CLI is operational
Debug: Sending rlRfSetLdoBypassConfig with 0 0 0
GPADC result: avg 568 min 568 max 569 sum 2274 time[us] 15
```