Embedded Motion Driver 5.1.2
Release Note

Contents

1. Revision History

Revision Date	Revision	Description
2/28/2014	1.0	Initial Release

2. Removed the rev id check for the MPU-6050 and MPU-6500

3. Added a comment about a known bug with the DMP

```
/*
* Known Bug -
 * DMP when enabled will sample sensor data at 200Hz and output to FIFO at the rate
 * specified in the dmp_set_fifo_rate API. The DMP will then sent an interrupt once
 * a sample has been put into the FIFO. Therefore if the dmp_set_fifo_rate is at 25Hz
 * there will be a 25Hz interrupt from the MPU device.
*
* There is a known issue in which if you do not enable DMP_FEATURE_TAP
* then the interrupts will be at 200Hz even if fifo rate
* is set at a different rate. To avoid this issue include the DMP_FEATURE_TAP
```

4. New Feature - added the self test for MPU6500 and MPU9250. Previously self test always passed.

New API -

```
/**
*  @brief      Trigger gyro/accel/compass self-test for MPU6500/MPU9250
*  On success/error, the self-test returns a mask representing the sensor(s)
*  that failed. For each bit, a one (1) represents a "pass" case; conversely,
*  a zero (0) indicates a failure.
*
*  \n The mask is defined as follows:
*  \n Bit 0:   Gyro.
*  \n Bit 1:   Accel.
*  \n Bit 2:   Compass.
```

```
 *
 * @param[out] gyro        Gyro biases in q16 format.
 * @param[out] accel       Accel biases (if applicable) in q16 format.
 * @param[in]  debug       Debug flag used to print out more detailed logs. Must first set up
logging in Motion Driver.
 * @return     Result mask (see above).
 */
int mpu_run_6500_self_test(long *gyro, long *accel, unsigned char debug)
```

5. New Feature - APIs to use the hardware calibration cancellations registers in the MPU6050 and MPU6500 instead of using the DMP registers. This way users can get calibrated accel data instead of only raw accel data.

New APIs -

```
/**
 * @brief      Read biases to the accel bias 6500 registers.
 * This function reads from the MPU6500 accel offset cancellations registers.
 * The format are G in +-8G format. The register is initialized with OTP
 * factory trim values.
 * @param[in]  accel_bias  returned structure with the accel bias
 * @return     0 if successful.
 */
int mpu_read_6500_accel_bias(long *accel_bias)

/**
 * @brief      Read biases to the accel bias 6050 registers.
 * This function reads from the MPU6050 accel offset cancellations registers.
 * The format are G in +-8G format. The register is initialized with OTP
 * factory trim values.
 * @param[in]  accel_bias  returned structure with the accel bias
 * @return     0 if successful.
 */
int mpu_read_6050_accel_bias
```

```
/**
 *  @brief      Push biases to the gyro bias 6500/6050 registers.
 *  This function expects biases relative to the current sensor output, and
 *  these biases will be added to the factory-supplied values. Bias inputs are LSB
 *  in +-1000dps format.
 *  @param[in]  gyro_bias  New biases.
 *  @return     0 if successful.
 */
int mpu_set_gyro_bias_reg

/**
 *  @brief      Push biases to the accel bias 6050 registers.
 *  This function expects biases relative to the current sensor output, and
 *  these biases will be added to the factory-supplied values. Bias inputs are LSB
 *  in +-8G format.
 *  @param[in]  accel_bias  New biases.
 *  @return     0 if successful.
 */
int mpu_set_accel_bias_6050_reg


/**
 *  @brief      Push biases to the accel bias 6500 registers.
 *  This function expects biases relative to the current sensor output, and
 *  these biases will be added to the factory-supplied values. Bias inputs are LSB
 *  in +-8G format.
 *  @param[in]  accel_bias  New biases.
 *  @return     0 if successful.
 */
int mpu_set_accel_bias_6500_reg
```