**SENSOR TAG CC2650**

The Sensor Tag is an Interesting Innovative Product which is targeted for software devices for mobile application development made by Texas Instruments (TI). The Sensor Tag CC2650 is a Bluetooth Enabled latest product with all the advanced features which supports the Cloud-connection strengthening the IoT idea.

**Features of Sensor Tag CC2650:**

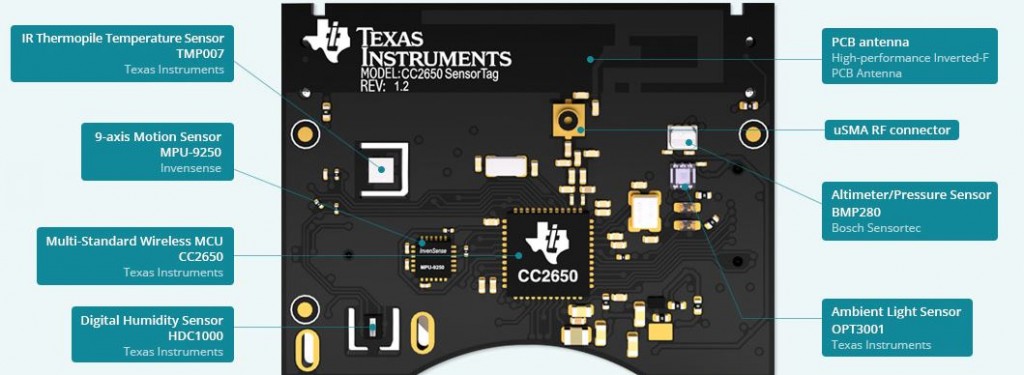
1. **Low Power:** Includes 10 low-power MEMS sensors. Longer Battery life than the coin cell and enabler of battery less applications.
2. **High Performance ARM Cortex M3 (CC2650).**
3. **Cloud Connectivity:** Can be accessed and controlled from anywhere and enables Seamless Integration with the mobile apps.
4. **More Sensors:** It adds Support for more Low power Sensors. Support for light, digital microphone, magnetic sensor, humidity, pressure, accelerometer, gyroscope, magnetometer, object temperature, and ambient temperature.
5. **Extreme Flexibility for IoT Applications:** Enable ZigBee or LowPAN through a firmware upgrade.
6. **DevPacks Allows to expand the SensorTag to fit our Designs:** Interchangeable Between all SensorTag types. [1]

**Components and Hardware Design:**

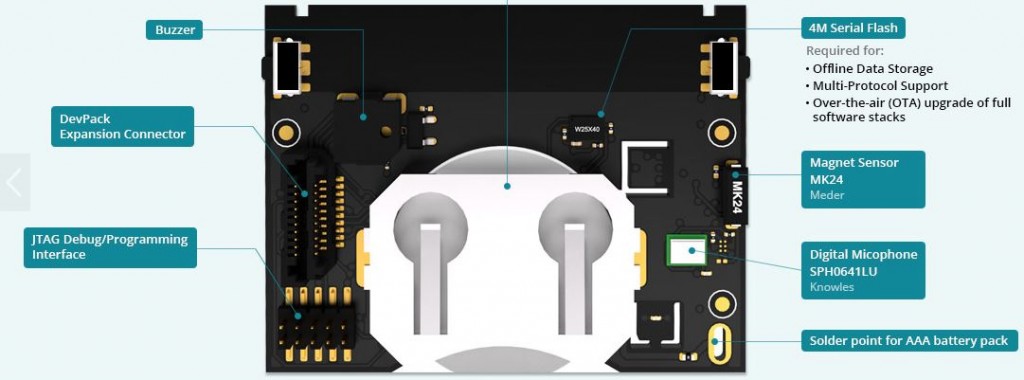
The CC2650 consists of a Cortex-M0 RF core and a Cortex-M3 user application core. The [Texas Instruments TI SensorTag](http://www.ti.com/ww/en/wireless_connectivity/sensortag/) is a battery powered, Bluetooth Low Energy (BLE) device, with a number of different sensors that can be used for various applications and projects. Available sensors are:

The SensorTag has support for the following sensors: IR Temperature, both object and ambient temperature Movement, 9 axis (accelerometer, gyroscope, magnetometer) Humidity, both relative humidity and temperature Barometer, both pressure and temperature Optical, light intensity Sensor Overview Sensor Name Data size Manufacturer IR Temperature TMP007 2 x 16bit Texas Instruments Movement MPU9250 9 x 16 bit InvenSense Humidity HDC1000 2 x 16bit Texas Instruments Pressure BMP280 2 x 24bit Bosch Optical OPT3001 1 x 16bit Texas Instruments [2]

The CC2650 SensorTag is a BLE peripheral device based on CC26xx hardware platform including five peripheral sensors with a complete software solution for sensor drivers interfaced to a GATT server running on TI BLEv2.1 stack. The GATT server contains a primary service for each sensor for configuration and data collection.



### Inside the Casing -Top Side [1]



### Inside the Casing – Bottom Side [2]

4 GATT Server Power on after inserting the battery the LEDs are used to indicate the status of the power on self-test. If all sensors AND the external flash pass the self-test the green LED will blink 5 times in rapid succession. If any of the tested units failed, the red LED will blink. After completing initialisation the green LED will blink every second as long as the device is advertising.

**MOTION DETECTION IN CC2650:**

The SensorTag can be configured to send notifications for every sensor by writing 0x0001 to the characteristic configuration < GATT\_CLIENT\_CHAR\_CFG\_UUID> for the corresponding sensor data, the data is then sent as soon as the data has been updated. The sensors are enabled by writing 0x01 (NB: The movement sensor uses a bitmap to control individual axes) to the corresponding configuration characteristic and then disabled by writing 0x00. For regular data collection the SensorTag does not use any interrupt features provided by the sensors, but the movement sensor (MPU9250) uses interrupt for the wake-on-movement (shake detection).

The SensorTag uses I2C to interface to the sensors. The MPU9250 is assigned a separate I2C-bus (bus #1), and also separately powered. Also note that the MPU9250 consists of two separate devices in the same package; the magnetometer has its own I2C address. The movement sensor has a WOM (Wake-On-Motion) feature that optionally may be used, so that movement data is only reported when the device is touched. When WOM is enabled the SensorTag reports movement data for 10 seconds after the device has detected a movement.

The movement sensor used on the SensorTag is [MPU9250](http://www.invensense.com/products/motion-tracking/9-axis/mpu-9250/) from InvenSense.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Type | UUID | Access | Size (bytes) | Description |
| Data | AA81\* | R/N | 18 | GyroX[0:7], GyroX[8:15], GyroY[0:7], GyroY[8:15], GyroZ[0:7], GyroZ[8:15], AccX[0:7], AccX[8:15], AccY[0:7], AccY[8:15], AccZ[0:7], AccZ[8:15], MagX[0:7], MagX[8:15], MagY[0:7], MagY[8:15], MagZ[0:7], MagZ[8:15] |
| Notification | 2902 | R/W | 2 | Write 0x0001 to enable notifications, 0x0000 to disable. |
| Configuration | AA82\* | R/W | 2 | One bit for each gyro and accelerometer axis (6), magnetometer (1), wake-on-motion enable (1), accelerometer range (2). Write any bit combination top enable the desired features. Writing 0x0000 powers the unit off. |
| Period | AA83\* | R/W | 1 | Resolution 10 ms. Range 100 ms (0x0A) to 2.55 sec (0xFF). Default 1 second (0x64). |

The **configuration** characteristic is used to decide which data (axis) should be activated, if wake-on-motion is to be enabled, and the range of the accelerometer (2, 4, 8 and 16 G).

|  |  |
| --- | --- |
| Bits | Usage |
| 0 | Gyroscope z axis enable |
| 1 | Gyroscope y axis enable |
| 2 | Gyroscope x axis enable |
| 3 | Accelerometer z axis enable |
| 4 | Accelerometer y axis enable |
| 5 | Accelerometer x axis enable |
| 6 | Magnetometer enable (all axes) |
| 7 | Wake-On-Motion Enable |
| 8:9 | Accelerometer range (0=2G, 1=4G, 2=8G, 3=16G) |
| 10:15 | Not used |

The Wake-On-Motion (WOM) feature allows the MPU to operate with only the accelerometer enabled, but will give an interrupt to the CC2650 when motion is detected. After a shake is detected, the SensorTag will provide movement data for 10 seconds before entering the MPU re-enters low power WOM state.

Note that this is raw data, the motion processing capabilities of the MPU9250 have not been invoked. The data consists of nine 16-bit signed values, one for each axis. The order in the data is Gyroscope, Accelerometer and Magnetomer.

***Gyroscope*** raw data make up 0-5 of the data from the movement service, in the order X, Y, Z axis. Data from each axis consists of two bytes, encoded as a signed integer. For conversion from gyroscope raw data to degrees/second, use the algorithm below on each of the first three 16-bit values in the incoming data, one for each axis. Note that the axis data from a disabled axis will be 0, so the size of the incoming data is always 18 bytes. When the WOM feature is enabled, the latest measured data will be continuously transmitted.

float sensorMpu9250GyroConvert(int16\_t **data**)

{

//-- calculate rotation, **unit** deg/s, range -250, +250

**return** (**data** \* 1.0) / (65536 / 500);

}

***Accelerometer*** raw data make up bytes 6-11 of the data from the movement service, in the order X, Y, Z axis. Data from each axis consists of two bytes, encoded as a signed integer. For conversion from accelerometer raw data to Gravity (G), use the algorithm below on each the three 16-bit values in the incoming data, one for each axis.

*// Accelerometer ranges*

#define ACC\_RANGE\_2G 0

#define ACC\_RANGE\_4G 1

#define ACC\_RANGE\_8G 2

#define ACC\_RANGE\_16G 3

**float** **sensorMpu9250AccConvert**(**int16\_t** rawData)

{ **float** v;

**switch** (accRange)

{

**case** ACC\_RANGE\_2G:

*//-- calculate acceleration, unit G, range -2, +2*

v = (rawData \* 1.0) / (32768/2);

**break**;

**case** ACC\_RANGE\_4G:

*//-- calculate acceleration, unit G, range -4, +4*

v = (rawData \* 1.0) / (32768/4);

**break**;

**case** ACC\_RANGE\_8G:

*//-- calculate acceleration, unit G, range -8, +8*

v = (rawData \* 1.0) / (32768/8);

**break**;

**case** ACC\_RANGE\_16G:

*//-- calculate acceleration, unit G, range -16, +16*

v = (rawData \* 1.0) / (32768/16);

**break**;

}

**return** v;

}

***Magnetometer*** raw data make up bytes 7-12 of the data from the movement service, in the order X, Y, Z axis. Data from each axis consists of two bytes, encoded as a signed integer. The conversion is done in the SensorTag firmware so there is no calculation involved apart from changing the integer to a float if required. The measurement unit is uT (micro Tesla).

float sensorMpu9250MagConvert(int16\_t **data**)

{

//-- calculate magnetism, **unit** uT, range +-4900

**return** 1.0 \* **data**;

}