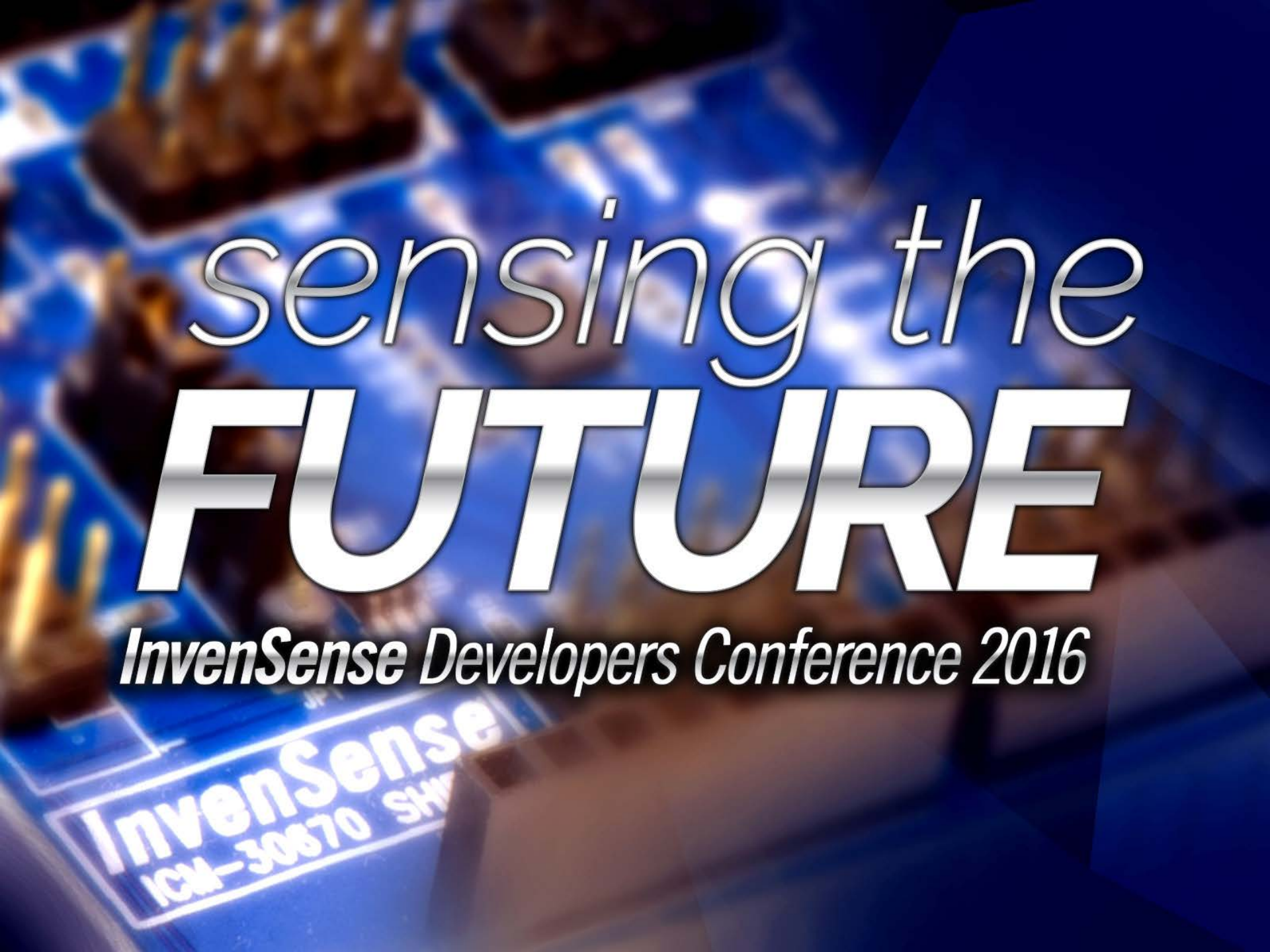

sensing the
FUTURE
InvenSense Developers Conference 2016
InvenSense
ICM-30670

ICM3063X Introduction

Sebastien Riccardi *Director System Team*

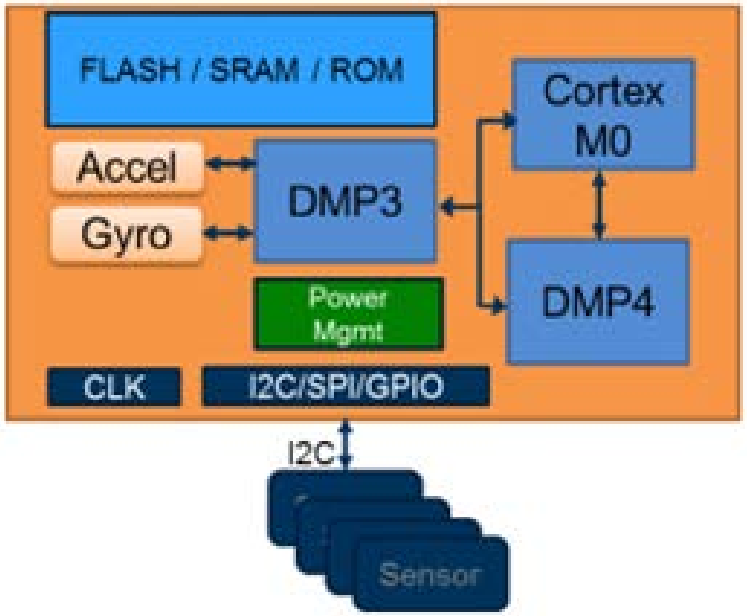
ICM3063X Market

sensing the FUTURE

ICM3063X SOC Architecture

sensing the FUTURE

Block Diagram:

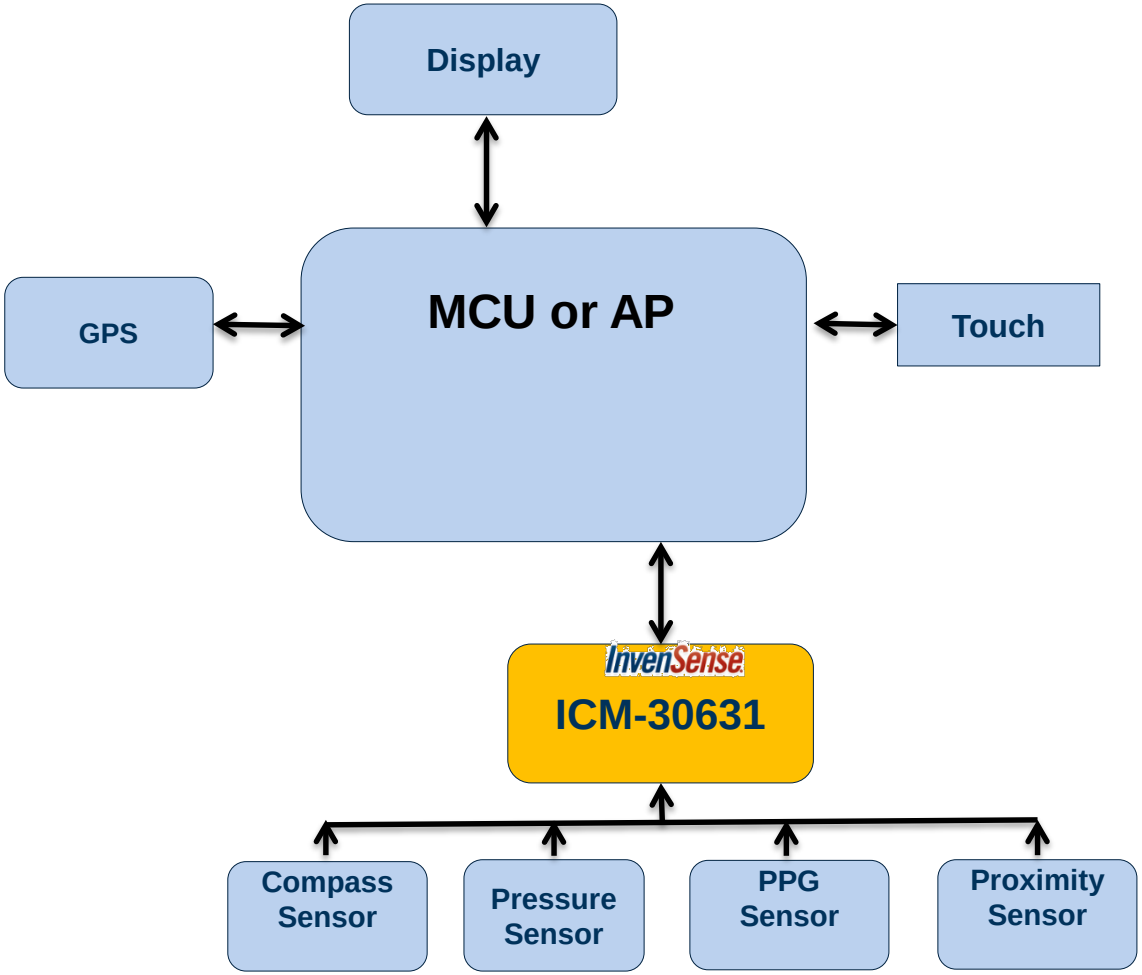


Features:

- Tri-Core Sensor-Hub
  - Cortex M0
  - DMP4 Motion Engine
  - DMP3 Motion Engine
- 6 Axis sensors integrated
- 64KB of Flash
- 64KB of RAM
- SPI/I2C Host Interface
- I2C Master for additional sensors
- 3 GPIO pin configurable
- 3 on-chip oscillators

ICM3063X in your wearable product

sensing the FUTURE

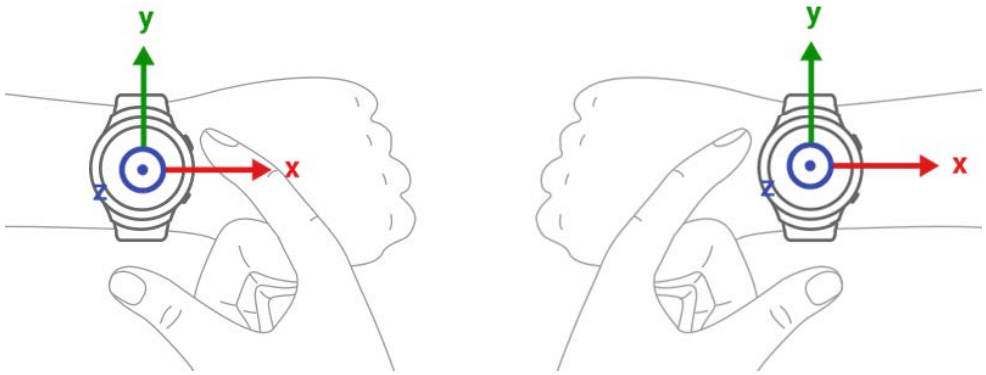


Sensors Available Description

Accelerometer and Gyroscope

sensing the FUTURE

- **Description :** Accelerometer and Gyroscope sensors data with realtime calibration
- **Parameters :**
  - ODR (max 200Hz)
  - Orientation matrix
  - Offset



- **Dependencies :** No
- **Output Data :**
  - Data calibrated (in G and DPS) with calibration accuracy flag
  - Data uncalibrated (in G and DPS)
- **ODR :** Every sample or batching mode

Consumption :	Acc	500µA@100Hz		
	Gyr	1600µA@100Hz	✓ ICM3061	✓ ICM3062

Magnetometer

- **Description :** Magnetometer AK09911 driver with live calibration
- **Parameters:**
  - ODR (max 100Hz)
  - Orientation matrix
  - Offset
- **Dependencies :** Need an external AK09911
- **Output Data :**
  - Data calibrated in µT with calibration accuracy flag
  - Data uncalibrated in µT
- **ODR :** Every sample or batching mode

Consumption : 330µA@100Hz ✗ ICM3061 ✓ ICM3062

Pressure and Ambiant Temperature

sensing the FUTURE

- **Description :** Pressure and Ambiant temperature BMP280 Driver
- **Parameters:**
  - ODR (max 16Hz)
- **Dependencies :** Need an external BMP280 Sensor
- **Output Data :**
  - Pressure in Pa
  - Ambiant Temperature in °C
- **ODR :** Every sample or batching mode

Consumption : Pres	59µA@5Hz			
Temp	55µA@5Hz		ICM3061	 ICM3062

Proximity Sensor

- **Description :** Proximity Sensor Driver
- **Parameters:**
  - ODR (max 16Hz)
- **Dependencies :** Need an external CM36671 Sensor
- **Output Data :**
  - Distance in millimeters
- **ODR :** Every sample or batching mode

Consumption : 1950 µA

 ICM3061

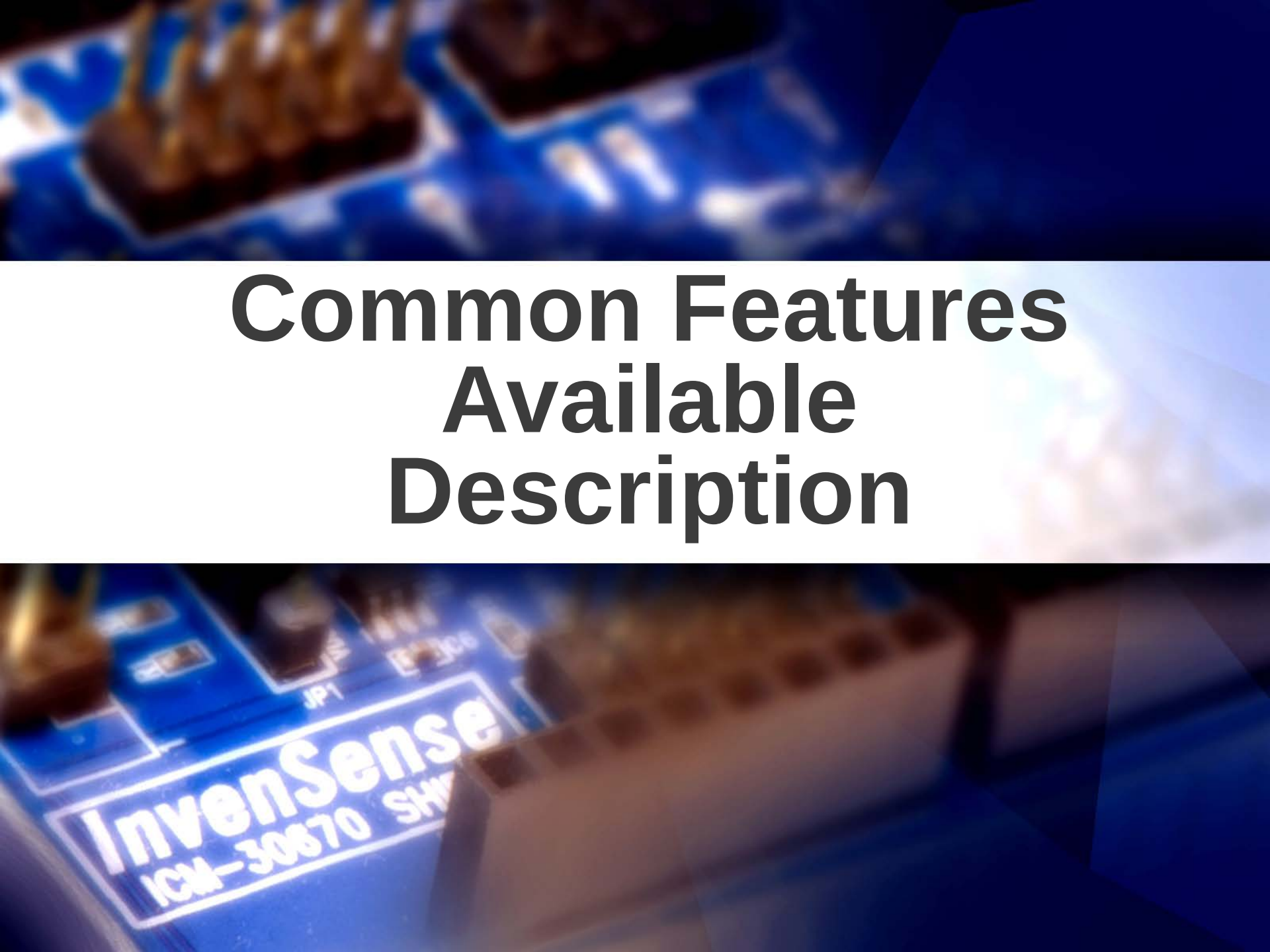
 ICM3062

PPG Sensor

- **Description :** PPG Driver
- **Parameters :** No
- **Dependencies :** Need an external PAH8001/PAH8002/ADPD174/PPS960 Sensor
- **Output Data :**
  - PPG Raw Data
  - Touch Status
- **ODR :** Every sample or batching mode

Consumption : 2000µA to 5000µA

 ICM3061  ICM3062



Common Features Available Description

Gravity and Linear Acceleration

sensing the FUTURE

- **Description :** This algorithm calculates the direction and the magnitude of gravity and the linear acceleration.
- **Parameters :**
  - ODR (max 200Hz)
- **Dependencies :**
  - Accelerometer
  - Gyroscope
- **Output Data :**
  - Gravity Vector in G
  - Linear Acceleration Vector in G
- **ODR :** Every sample or batching mode

Consumption : 1700µA@100Hz

 ICM3061

 ICM3062

GRV : 6 Axis Orientation

sensing the FUTURE

- **Description :** This algorithm calculates the 6 axis orientation of the device
- **Parameters :**
  - ODR (max 200Hz)
- **Dependencies :**
  - Accelerometer
  - Gyroscope
- **Output Data:**
  - Rotation Vector Quaternions
- **ODR :** Every sample or batching mode

Consumption : 1700µA@100Hz

✓ ICM3061 ✓ ICM3062

Geomag : 6 Axis Orientation

sensing the FUTURE

- **Description :** This algorithm calculates the 6 axis orientation of the device
- **Parameters :**
  - ODR (max 100Hz)
- **Dependencies :**
  - Accelerometer
  - Magnetometer Sensor (AK09911)
- **Output Data:**
  - Rotation Vector Quaternions
- **ODR :** Every sample or batching mode

Consumption : 800µA@100Hz

 ICM3061

 ICM3062

RV : 9 Axis Orientation

sensing the FUTURE

- **Description :** This algorithm calculates the 9 axis orientation of the device
- **Parameters:**
  - ODR(max 200Hz)
- **Dependencies :**
  - Accelerometer
  - Gyroscope
  - Magnetometer Sensor (AK09911)
- **Output Data:**
  - Rotation Vector Quaternions
- **ODR :** Every sample or batching mode

Consumption : 1900µA@100Hz

 ICM3061

 ICM3062

Orientation depredicated : 9 Axis Orientation

sensing the FUTURE

- **Description :** This algorithm calculates the 9 axis orientation of the device
- **Parameters :**
  - ODR (max 200Hz)
- **Dependencies :**
  - Accelerometer
  - Gyroscope
  - Magnetometer Sensor (AK09911)
- **Output Data :**
  - Euler Angles (yaw, pitch, roll) in degrees
- **ODR :** Every sample or batching mode

Consumption : 1900µA@100Hz

 ICM3061

 ICM3062

Wrist Features Available Description

AAR

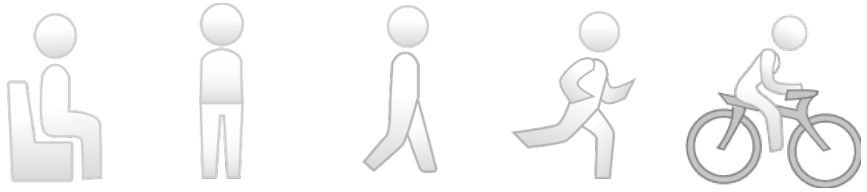
- **Description :** The AAR is an advanced activity recognition algorithm
- **Parameters :**
  - Enable Notification
- **Dependencies :**
  - Accelerometer
- **Output Data :** Activity (Walk, Run, Bike, Still)
- **ODR :** On Activity change or on request

Consumption : 300µA

✓ ICM3061 ✓ ICM3062

AAR Extended

- **Description :** The AAR Extended is an algorithm which add Sit and Stand Activity recognition to standard AAR algorithm
- **Parameters :** No
- **Dependencies :**
  - AAR
  - Accelerometer
- **Output Data :** Activity (Walk, Run, Bike, Sit and Stand)
- **ODR:** On Activity change

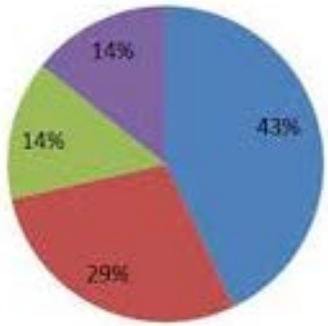


Consumption : 300µA

✓ ICM3061 ✓ ICM3062

AAR Statistics

- **Description :** The AAR Statistics summarizes all activities statistics
- **Parameters :** No
- **Dependencies :**
  - AAR Extended
  - Step Counter
- **Output Data :**
  - Walk/Run/Bike/Sit/Stand Duration
  - Walk and Run Step Counters
- **ODR :** On Activity change

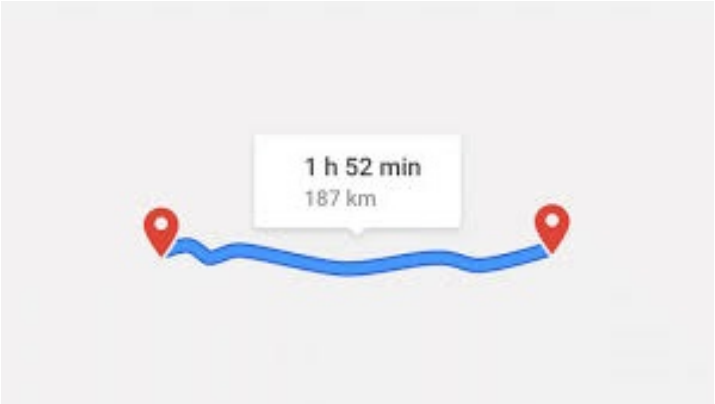


Consumption : 300µA

✓ ICM3061 ✓ ICM3062

Distance Walk/Run

- **Description :** This algorithm calculates Distance during walking and running activity.
- **Parameters :**
  - User Height
  - Enable Notification
- **Dependencies :**
  - AAR
  - Step Counter
- **Output Data :**
  - Walk Distance in meters
  - Run Distance in meters
- **ODR :** Each step or on request

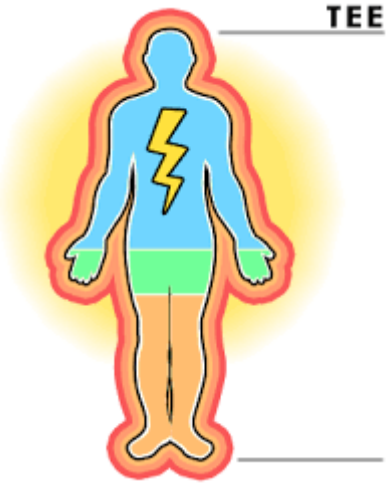


Consumption : 300µA

✓ ICM3061 ✓ ICM3062

Energy Expenditure

- **Description :** This algorithm computes the user Energy expenditure.
- **Parameters :**
  - User Height, Weight, Age and Gender
  - Enable Notification
- **Dependencies :**
  - AAR
- **Output Data :**
  - Energy Expenditure in Kcal Instant and cumulative
  - Energy Expenditure in METS Instant and cumulative
- **ODR :** Each minutes or on request



Consumption : 300µA

✓ ICM3061 ✓ ICM3062

Floor Climbed Counter

sensing the FUTURE

- **Description :** This algorithm counts the number of floor climbed
- **Parameters :**
  - User Height, Weight, Age and Gender
  - Enable Notification
- **Dependencies :**
  - AAR
  - Pressure Sensor (BMP280)
- **Output Data :**
  - Number of Floor Up
  - Number of Floor Down
- **ODR :** On Change

Consumption : 350µA

✓ ICM3061 ✓ ICM3062

Shake Wrist

- **Description :** This Feature detects Shake Wrist gesture
- **Parameters :**
  - Max Period for a half oscillation
- **Dependencies :**
  - Accelerometer
- **Output Data :** Event
- **ODR :** On Event

Consumption : 250µA

✓ ICM3061 ✓ ICM3062

Double Tap

- **Description :** This Feature detects DoubleTap gesture
- **Parameters :**
  - Minimum threshold
  - Maximum duration between Single Tap
- **Dependencies :**
  - Accelerometer
- **Output Data :** Event
- **ODR :** On Event

Consumption : 300µA/1100µA

✓ ICM3061 ✓ ICM3062

Bring To See

- **Description :** This Feature detects Bring To See gesture
- **Parameters :** No
- **Dependencies :**
  - Accelerometer
  - AAR
- **Output Data :** Event
- **ODR :** On Event

Consumption : 300µA

✓ ICM3061 ✓ ICM3062

Sedentary Reminder

sensing the FUTURE

- **Description :** The Sedentary reminder is an alarm which occurs if the user stays in Sedentary activity too much time.
- **Parameters :**
  - AlarmParameterInSecond:

    Maximum Sit Duration Condition before Alarm occurs.
  - HysteresisParameterInSecond:

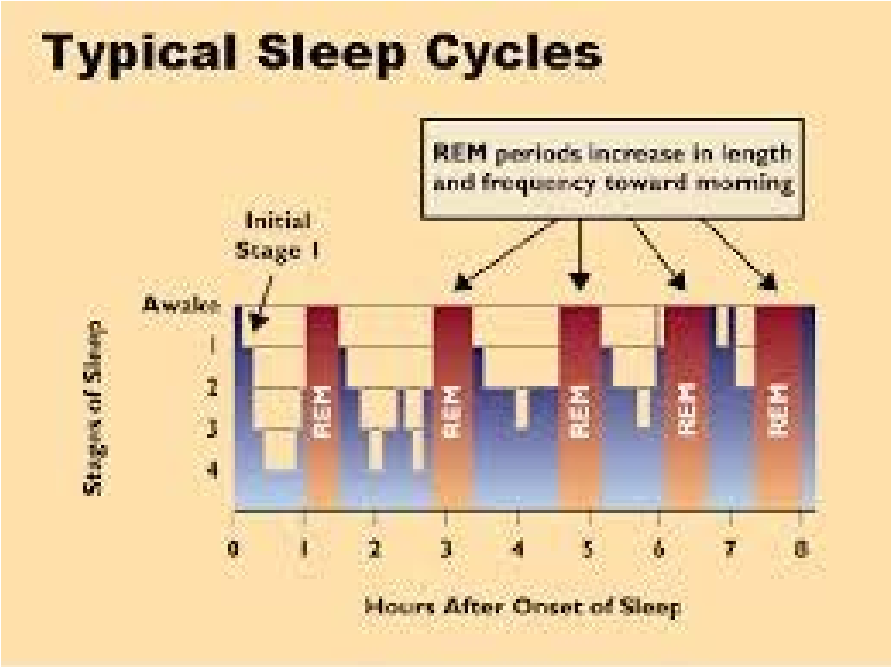
    Minimum Duration of non-sit condition to reset Sedentary Time Counter.
- **Dependencies :**
  - AAR Extended
- **Output Data :** Event
- **ODR :** On Event

Consumption : 300µA

✓ ICM3061 ✓ ICM3062

Sleep Analysis

- **Description :** This algorithm provides user sleep phases and summarizes your night (On trigger).
- **Parameters :** No
- **Dependencies :**
  - Accelerometer
- **Output Data :** Event
- **ODR :** On Event



Consumption : 250µA

✓ ICM3061 ✓ ICM3062

HRM/HRV

- **Description :** This algorithm calculate your heart rate monitor and variability (with motion cancellation)
- **Parameters :** No
- **Dependencies :**
  - PPG Sensor (PAH8001/PAH8002/PPS960/ADPD174)
  - Accelerometer
  - Gyroscope
- **Output Data :**
  - HRM
  - HRV
- **ODR :** Each 1,28s

Consumption : 1000µA+PPG

✓ ICM3061 ✗ ICM3062

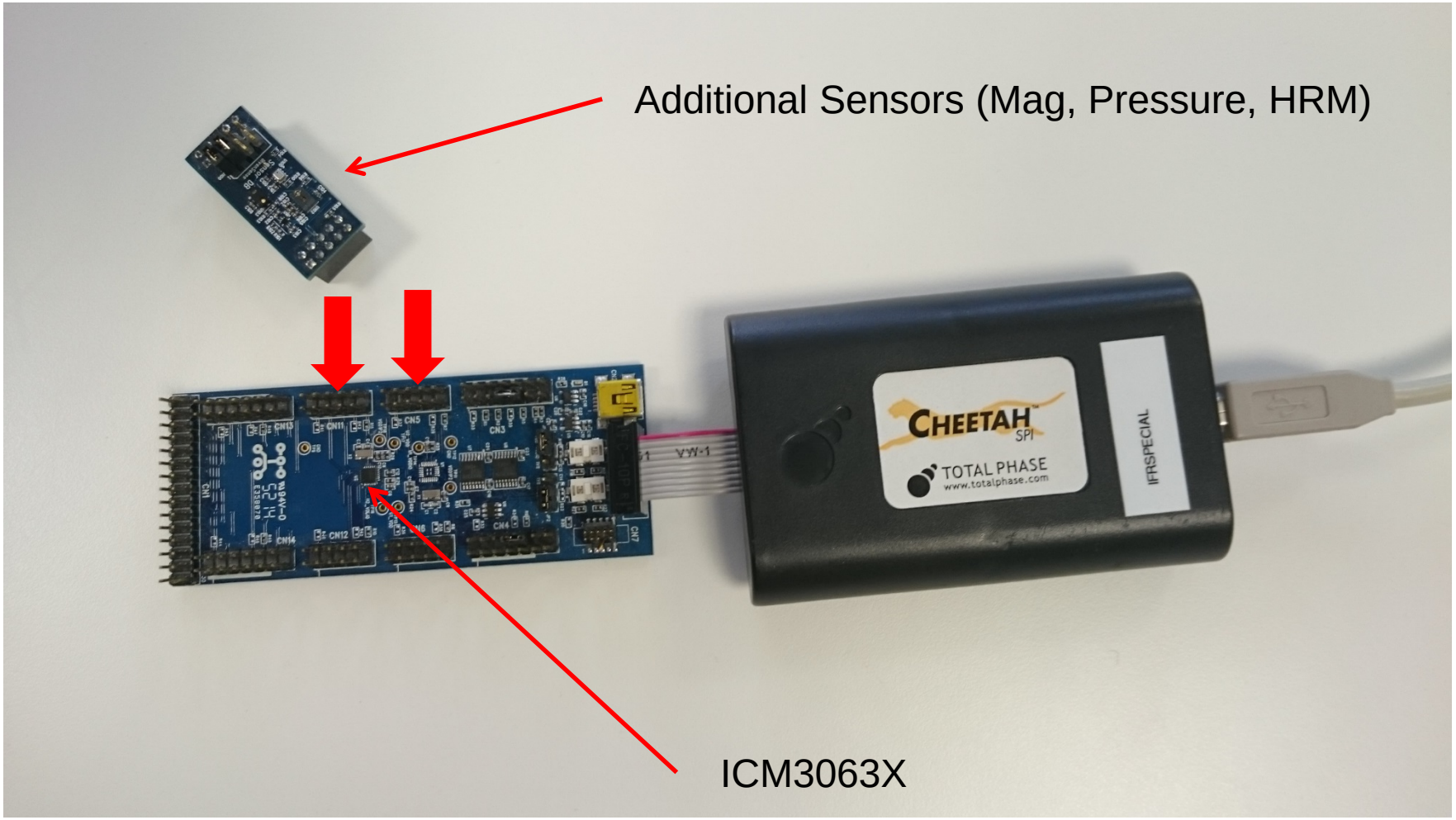
ICM30631 Vs ICM30632

sensing the FUTURE

Features List	ICM30631	ICM30632
Accelerometer	✓	✓
Gyroscope	✓	✓
Pressure (BMP280)	✓	✗
Temperature (BMP280)	✓	✗
Magnetometer (AK09911)	✗	✓
Proximity (CM36671)	✗	✓
PPG PAH8001	✓	✗
PPG PAH8002	✓	✗
PPG PPS960	✓	✗
PPG ADPD174	✓	✗
Gravity	✗	✓
Linear Acc	✗	✓
Orientation Depredicated (*)	✗	✓
RV (*)	✗	✓
GRV	✓	✓
GEOMAG (*)	✗	✓
HRM/HRV/HRT	✓	✗
Sleep Analysis	✓	✓
B2S	✓	✓
Double Tap	✓	✓
Shake Wrist	✓	✓
Floor Climbed	✓	✗
Energy Expenditure	✓	✓
Distance Walk/Run	✓	✓
AAR (Still, Walk, Run, Bike)	✓	✓
AAR Extended (+Still Stand/Sit)	✓	✓
AAR Statistics (Time Accrual, Step Counter)	✓	✓
Sedentary Reminder	✓	✓

Hardware Development Kit Overview

ICM30631 Evaluation Board



Device Driver and Tools Overview

Device Driver and Tools Deliverable

sensing the FUTURE

PC Windows

Sensor CLI

LibIDD
(Invensense Device Driver)

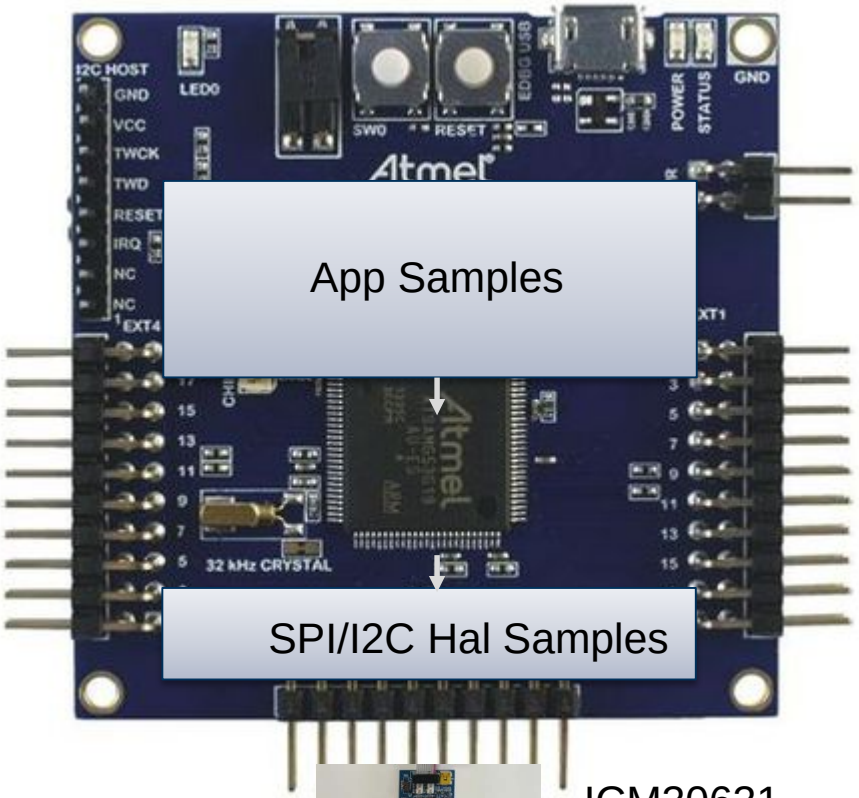
Adapter
Cheetah

USB/SPI
Bridge

ICM30631
SDK

Launch Sensorcli.bat

Embedded



ICM30631
SDK

Device Driver Integration 1/4

sensing the FUTURE

My Host SPI HAL:

```
/**     @example  ExampleSerifHal.c
*       @brief    Basic template for \ref SerifHal implementation.
*/

const inv_serif_hal_t my_serif_instance = {
        my_serif_open_read_reg,  /* callback to read_reg low level method */
        my_serif_open_write_reg, /* callback to read_reg low level method */
        128,                     /* maximum number of bytes allowed per read transaction,
                                    (limitation can come from internal buffer in the system driver) */
        128,                     /* maximum number of bytes allowed per write transaction,
                                    (limitation can come from internal buffer in the system driver) */
        INV_SERIF_HAL_TYPE_SPI,  /* type of the serial interface (between SPI or I2C) */
        (void *)0xDEAD           /* some context pointer passed to read_reg/write_reg callbacks */
};

int my_serif_open_read_reg(void * context, uint8_t reg, uint8_t * rbuffer, uint32_t rlen)
{
        (void)context, (void)reg, (void)rbuffer, (void)rlen;
        // MyTarget_SPI_do_read_reg(&reg, 1, rbuffer, rlen);
        return 0; // shall return a negative value on error
}


int my_serif_open_write_reg(void * context, uint8_t reg, const uint8_t * wbuffer, uint32_t wlen)
{
        (void)context, (void)reg, (void)wbuffer, (void)wlen;
        // MyTarget_SPI_do_write_reg(&reg, 1, wbuffer, wlen);
        return 0; // shall return a negative value on error
}
```

Device Driver Integration 2/4

sensing the FUTURE

ICM3063X Initialization:

```
void ICM3063X_Init(inv_device_icm30xxx_t  * device_icm30xxx, inv_device_t * device)
{

    inv_device_icm3063x_init2(&device_icm30xxx, &serif_instance, &sensor_listener,
    device_buffer, sizeof(device_buffer), false /* disable debug */);

    device = inv_device_icm30xxx_get_base(&device_icm30xxx);

    inv_device_whoami(device, &whoami);

    inv_device_load(device, INV_DEVICE_ICM30XXX_IMAGE_TYPE_M0_FLASH, flash_image,
                        sizeof(flash_image), true /* verify */, false /* force */);

    inv_device_load(device, INV_DEVICE_ICM30XXX_IMAGE_TYPE_M0_SRAM, ram_image,
                        sizeof(ram_image), true /* verify */, false /* force */);

    inv_device_setup(device);

    inv_device_load(device, INV_DEVICE_ICM30XXX_IMAGE_TYPE_DMP3, dmp3_image,
                        sizeof(dmp3_image), true /* verify */, false /* force */);
    inv_device_load(device, INV_DEVICE_ICM30XXX_IMAGE_TYPE_DMP4, dmp4_image,
                        sizeof(dmp4_image), true /* verify */, false /* force */);
}
```

Device Driver Integration 3/4

sensing the FUTURE

ICM3063X Setup:

```
void ICM3063X_Setup(inv_device_t * device)
{

      /*
       * Start accelerometer sensor at 20 Hz with a batch timeout of 1s
       */
        inv_device_set_sensor_period(device, INV_SENSOR_TYPE_ACCELEROMETER, 50);
        inv_device_set_sensor_timeout(device, INV_SENSOR_TYPE_ACCELEROMETER, 1000);
        inv_device_start_sensor(device, INV_SENSOR_TYPE_ACCELEROMETER);

      /*
       * Start Double Tap Detection and Heart Rate Monitor
       */
        inv_device_start_sensor(device, INV_SENSOR_TYPE_DOUBLE_TAP);
        inv_device_start_sensor(device, INV_SENSOR_TYPE_HEART_RATE);

}
```

Device Driver Integration 4/4

sensing the FUTURE

ICM3063X Manage Data and Event:

```
void MyInterruptHandler(void)
{
        int rc = 0;
        rc += inv_device_poll(device);
}

/*
 * Callback called upon sensor event reception
 * This function is called in the same function than inv_device_poll()
 */
static void sensor_event_cb(const inv_sensor_event_t * event, void * arg)
{

    if (event->status == INV_SENSOR_STATUS_DATA_UPDATED)
    {
                switch(event->sensor)
                {
          case INV_SENSOR_TYPE_HEART_RATE :
                MyGUI_PrintScreen ("BPM=%d", event->data.hrm.bpm)
                break;
          case INV_SENSOR_TYPE_DOUBLE_TAP :
          // Add all Sensors ID
                }
     }
}
```

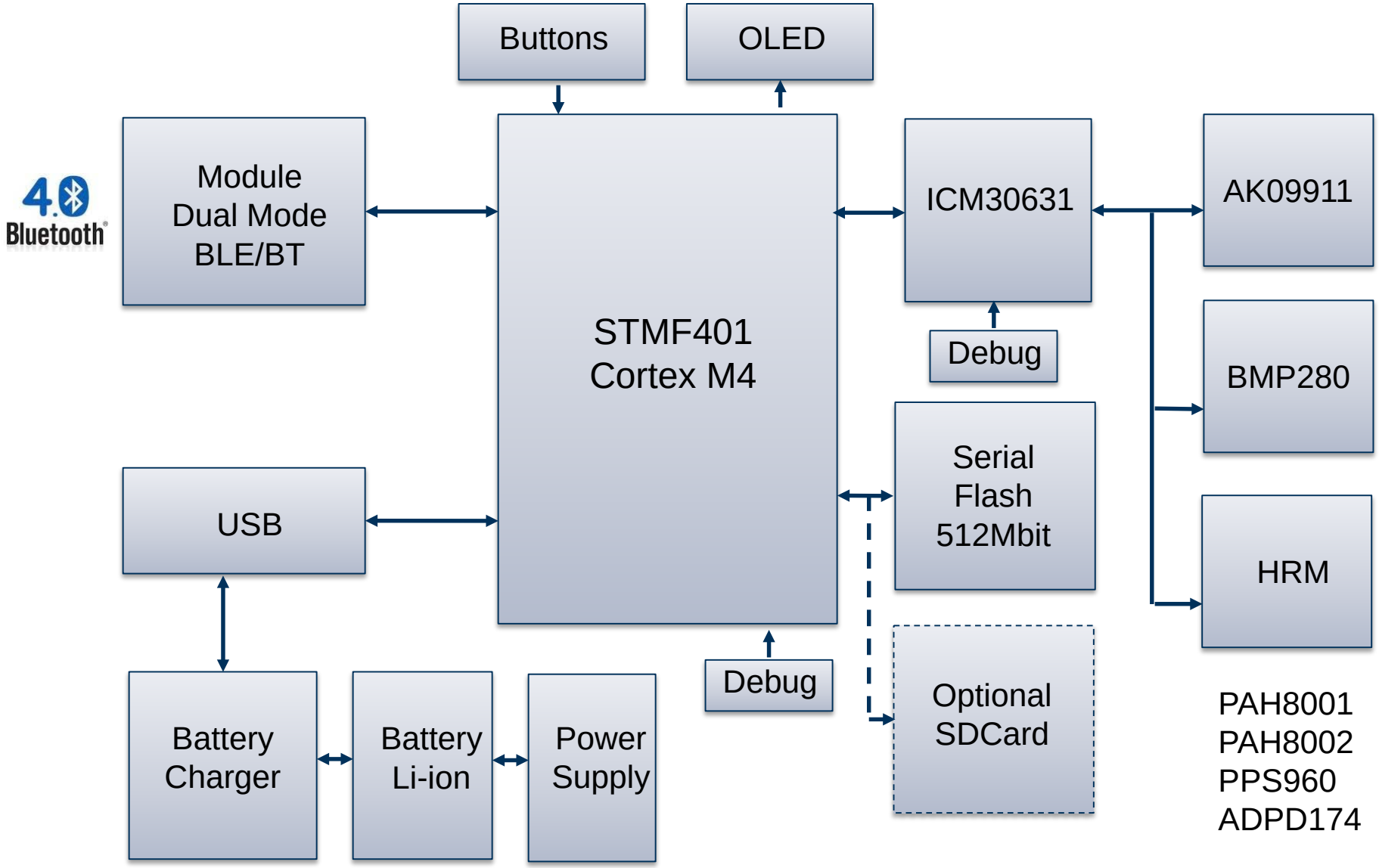
Hardware Development Platform Overview

ICM-3063X : Wrist Platform

sensing the FUTURE

ICM-3063X : HW ARCHITECTURE

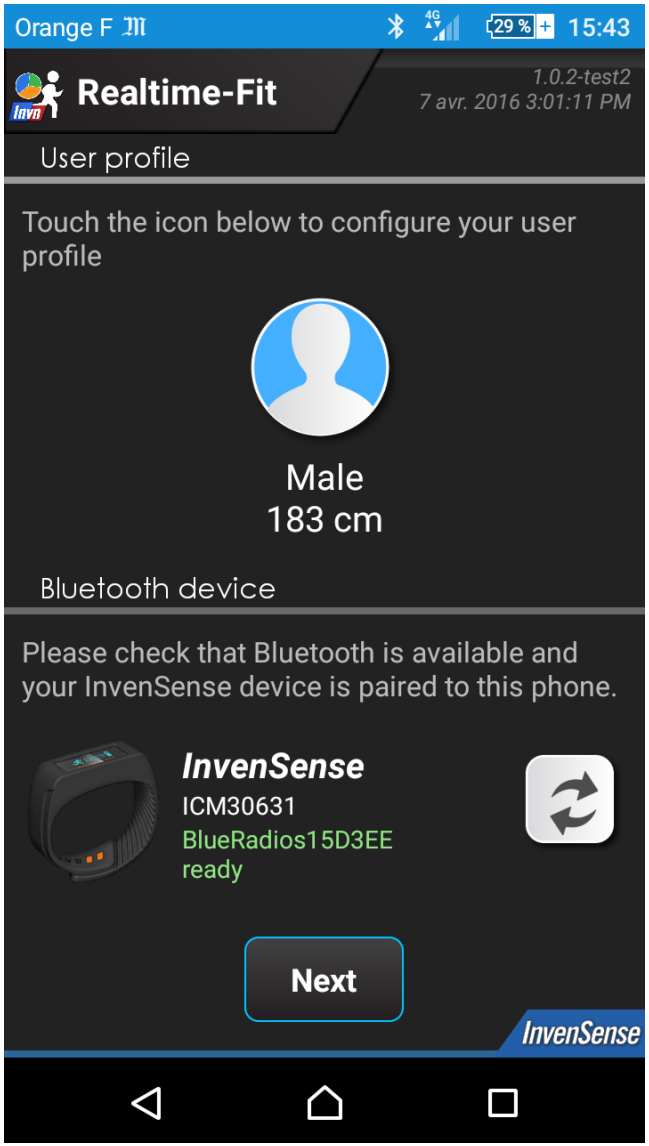
sensing the FUTURE

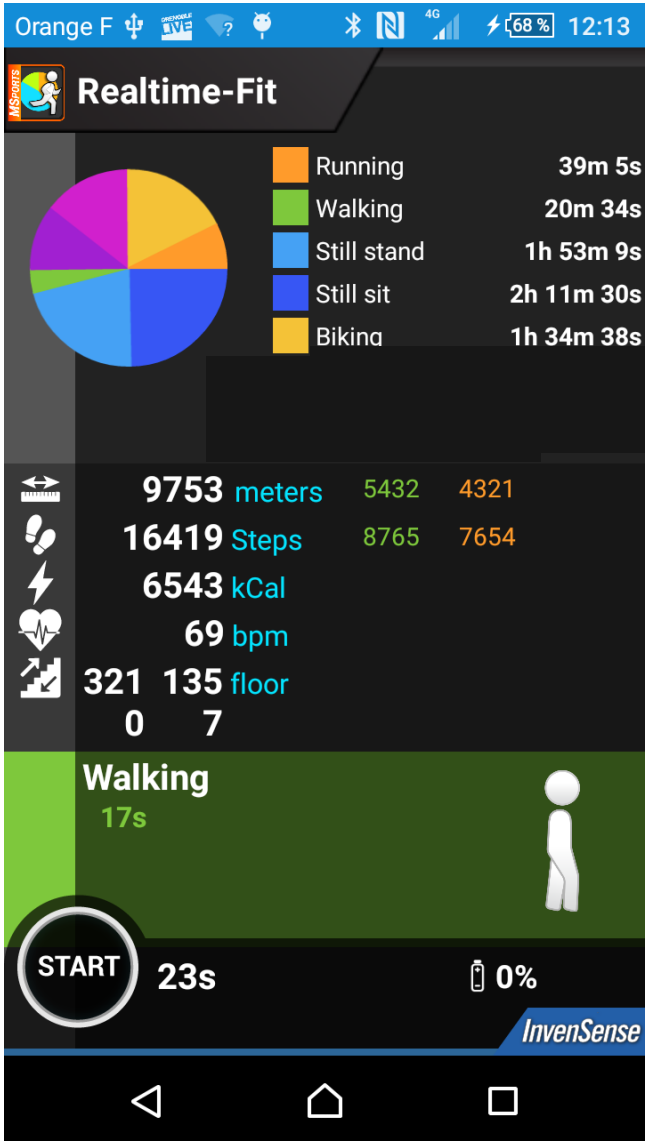


Software Evaluation Overview

ICM-30631 : Android RealTime APK

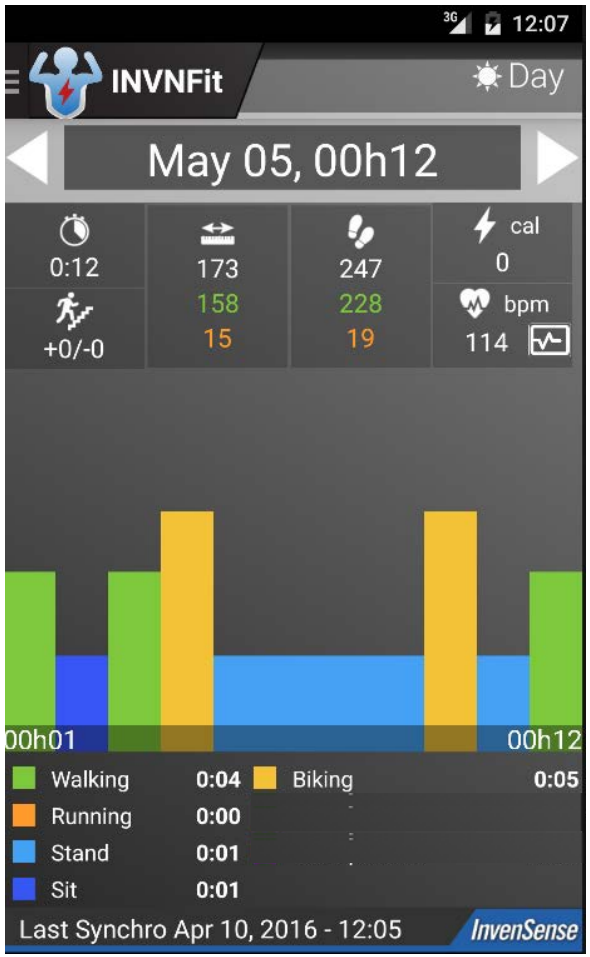
sensing the FUTURE

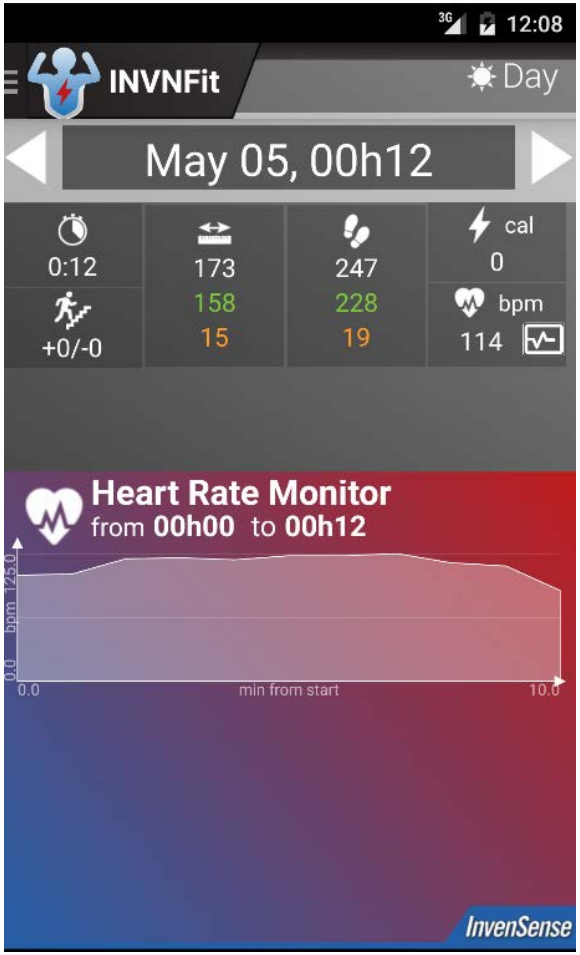


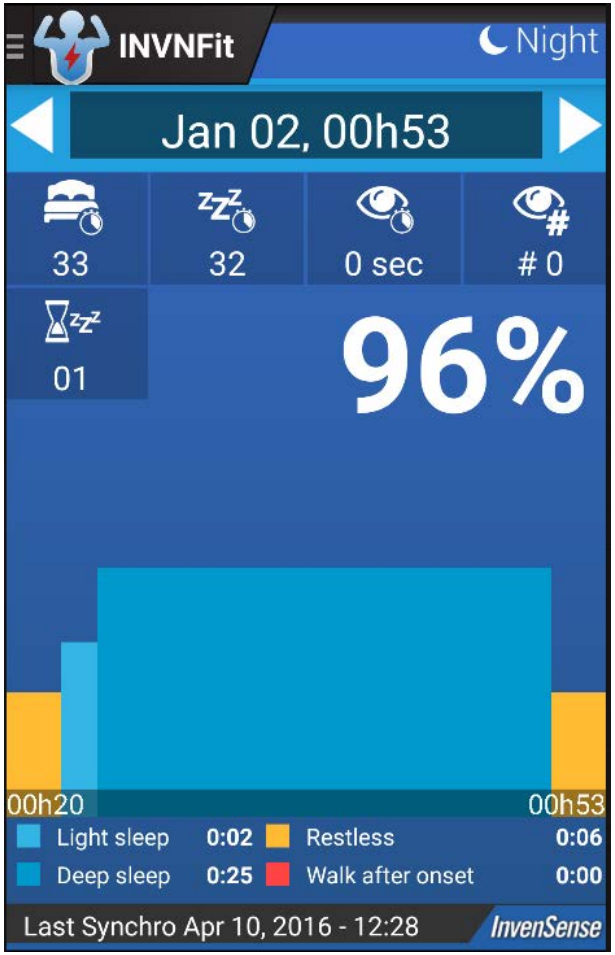


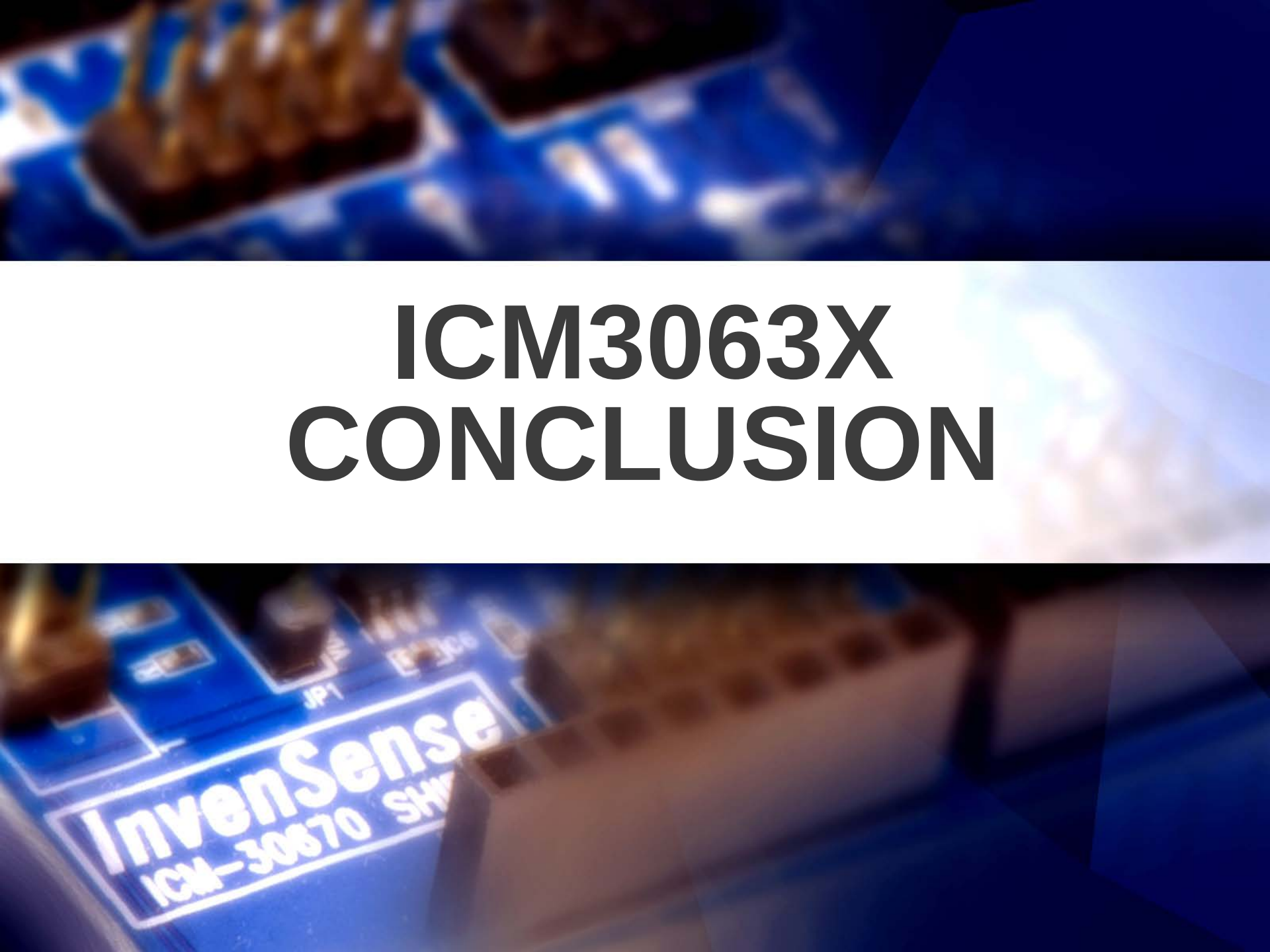
ICM-30631 : Android DayTime APK

sensing the FUTURE









ICM3063X CONCLUSION

Why use ICM3063X in your product ?

sensing the FUTURE

Accelerate your time to market:

- No Algorithm development
- No Data to collect for algorithm development and tuning
- No algorithm integration
- No external driver to develop (magnetometer, PPG, pressure, proximity)
- Accelerate power consumption optimization

Questions ?

?

Thank You