



AHEAD OF WHAT'S POSSIBLE™

Analog Devices, Inc.

[www.analog.com](http://www.analog.com)

# ADI VSM WATCH FIRMWARE

**Revision List****Table 1: Revision List**

<b>Revision</b>	<b>Date</b>	<b>Description</b>
0.1	31-Aug-2020	Initial Draft
0.2	9-Sep-2020	Extending document sections
0.3	06-Oct-2020	Added DCB section
0.4	22-Nov-2020	Added Low touch
0.5	27-Nov-2020	Added ADPD/ADXL sections
0.6	22-Jan-2021	Added LT tuning

**Copyright, Disclaimer Statements****Copyright Information**

Copyright (c) 2021 Analog Devices, Inc. All Rights Reserved. This software is proprietary and confidential to Analog Devices, Inc. and its licensors. This document may not be reproduced in any form without prior, express written consent from Analog Devices, Inc.

**Disclaimer**

Analog Devices, Inc. reserves the right to change this product without prior notice. Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

## Table of Contents

<b>Revision List.....</b>	<b>2</b>
<b>Copyright, Disclaimer Statements .....</b>	<b>3</b>
<b>Table of Contents .....</b>	<b>4</b>
<b>List of Figures .....</b>	<b>5</b>
<b>List of Tables .....</b>	<b>5</b>
<b>List of Code Snippets .....</b>	<b>6</b>
<b>1 Introduction .....</b>	<b>7</b>
1.1 Purpose .....	7
1.2 Scope .....	7
1.3 References .....	7
1.4 Acronyms.....	7
<b>2 Overview of VSM Watch.....</b>	<b>9</b>
2.1 Functional Roles of Watch .....	9
2.2 Firmware Architecture .....	10
2.2.1 Firmware Development .....	10
2.2.2 Memory Layout .....	10
2.2.3 Bootloader.....	11
2.2.4 Watch Application .....	11
2.2.5 M2M2 Protocol in VSM Watch.....	12
2.2.5.1 M2M2 Message Passing – Header Structure.....	13
2.2.5.2 M2M2 Message Passing Implementation .....	13
2.2.6 BLE Protocol in VSM Watch.....	14
2.2.7 Timestamps .....	15
2.2.8 Watch UX.....	16
2.2.9 Lightweight File System (LFS).....	16
<b>3 Configurations and Settings.....</b>	<b>18</b>
3.1.1 Device Configuration Block (DCB).....	18
<b>4 Firmware Applications and Packet format .....</b>	<b>20</b>
4.1 Applications .....	20
4.1.1 ADPD.....	20
4.1.1.1 AUTOMATIC GAIN CONTROL (AGC) .....	20
4.1.2 ADXL .....	21
4.1.3 Electrocardiogram (ECG) .....	21
4.1.4 Electro Dermal Activity (EDA).....	21
4.1.5 Bio Impedance Activity (BIA) .....	22

4.1.6 Low Touch (LT) Application.....	22
4.1.6.1 LT Config file creation.....	22
4.1.6.2 Wrist on/off detection.....	23
4.1.6.3 LT app in bootup sequence .....	23
4.1.6.4 LT Application Tuning.....	24
4.1.7 System application .....	25
4.2 Packet format.....	25
<b>5 Embedded Algorithms on Watch .....</b>	<b>26</b>
5.1 Heart Rate Monitoring (HRM) algorithm .....	26
5.2 Pedometer algorithm.....	26
5.3 Signal Quality Index (SQI) algorithm .....	26
5.4 ECG Heart Rate algorithm .....	26
<b>6 FAQs .....</b>	<b>27</b>
6.1 Watch Usage .....	28
6.2 Battery .....	28
6.3 NAND Flash.....	28
6.4 Sensor/Bio-Medical Applications.....	29
6.4.1 Accelerometer (ADXL) .....	29
6.4.2 Photo Plethysmography (PPG) .....	30
<b>List of Figures</b>	
Figure 1: Functional roles of the watch .....	9
Figure 2: NOR Flash Memory Layout .....	10
Figure 3: Firmware architecture .....	11
Figure 4: Tools and SW Interaction .....	12
Figure 5: M2M2 header structure.....	13
Figure 6: Point to point M2M2 Messaging using Post Office .....	13
Figure 7: Publish-Subscribe M2M2 Messaging using Mailbox .....	14
Figure 8: Interaction of Watch applications with AWT/SDK using M2M2 messaging.....	14
Figure 9: Watch UX Source code directory structure .....	16
<b>List of Tables</b>	
Table 1: Revision List .....	2
Table 2: DCB block information .....	19
Table 3: Packet Format .....	25
Table 4: Supported Sampling Rates (Hz) .....	25
Table 5: NAND Flash Usage .....	29
Table 6: Accelerometer ADXL362 range .....	30

**List of Code Snippets**

**No table of figures entries found.**

# 1 Introduction

ADI VSM watch is a product designed to showcase the capabilities of the sensors for measuring the vital signs monitoring.

## 1.1 Purpose

This is a live document is to give an insight into the firmware configurations and settings of the ADI Watch.

## 1.2 Scope

The different sections covered in this document are:

- [Overview of VSM watch](#)
- [Configurations and Settings](#)
- [Firmware Applications and packet format](#)
- [FAQs](#)

## 1.3 References

Section	Document Name
BLE NUS	<a href="#">NordicNUS</a>

## 1.4 Acronyms

ADI	Analog Devices Inc
API	Application Processing Interface
BLE	Blue Tooth Low Energy
BSP	Board Support Package
DFU	Device Firmware Upgrade
DCB	Device Configuration Block
DCFG	Device Configuration
ECG	ElectroCardioGram

EDA	Electro Dermal Activity
FW	Firmware
LCD	Liquid Crystal Display
LCFG	Library Configuration
M2M2	Machine to Machine Version2
NUS	Nordic UART Service
OTA	Over The Air
PPG	Photo Plethysmography
USB	Universal Synchronous Bus
UX	User Experience
AWT	Applications Wavetool



## 2 Overview of VSM Watch

### 2.1 Functional Roles of Watch

Figure 1 illustrates the main functional roles of the watch.

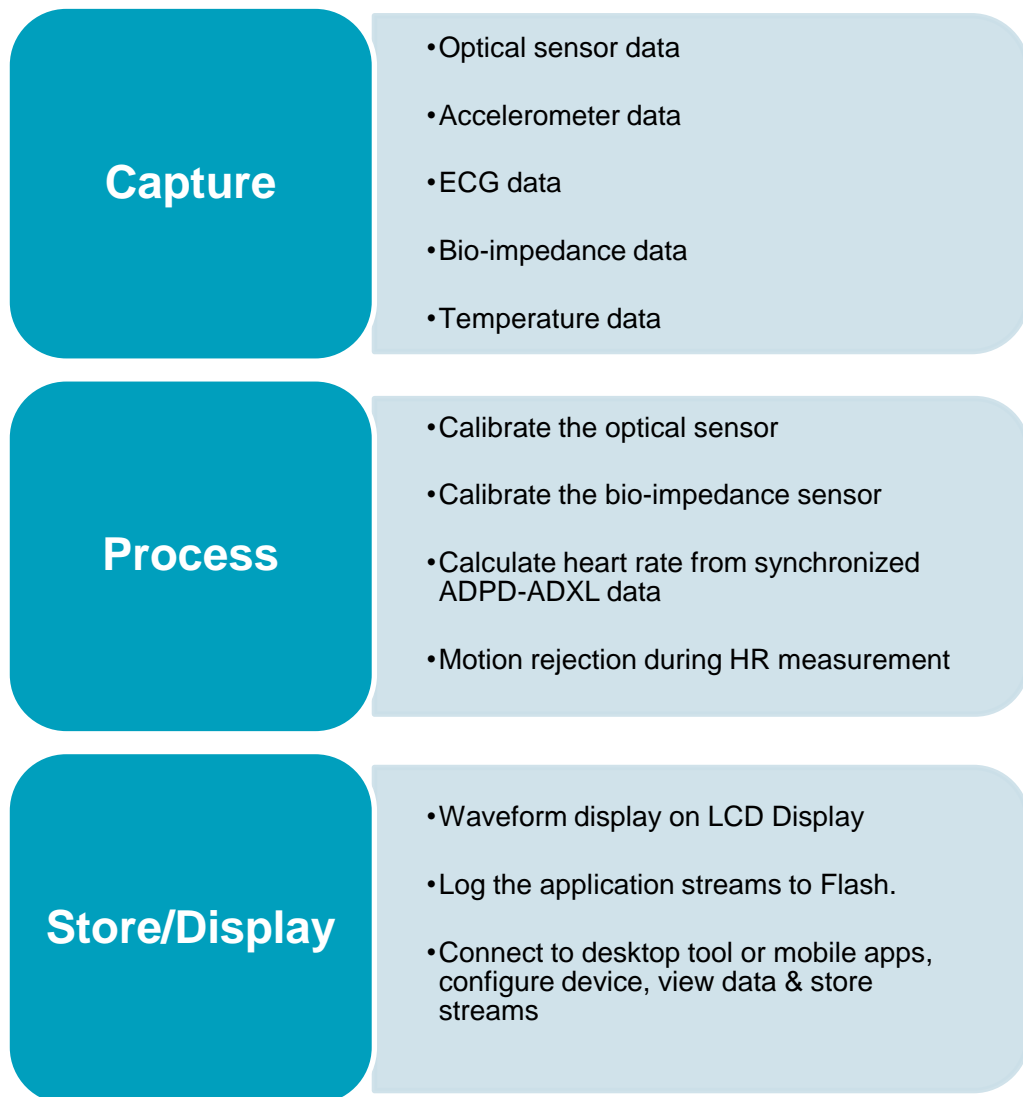


Figure 1: Functional roles of the watch

## 2.2 Firmware Architecture

### 2.2.1 Firmware Development

nRF5\_SDK\_15.2.0 and FreeRTOS V10.0.0 source code is being used for the firmware development.

"SEGGER Embedded Studio for ARM Release 4.12 Build 2018112601.37855 Windows x64" is used for developing the Watch application binary.

For developer testing of the commands provided to the tools from the Watch, a command line interface using Python 2.7 version is used. (the 32-bit is tested, 64-bit version might have some dependency problems so it is highly recommended to use 32-bit version)

The Watch Firmware is comprised of the bootloader, watch application and BLE Softdevice.

### 2.2.2 Memory Layout

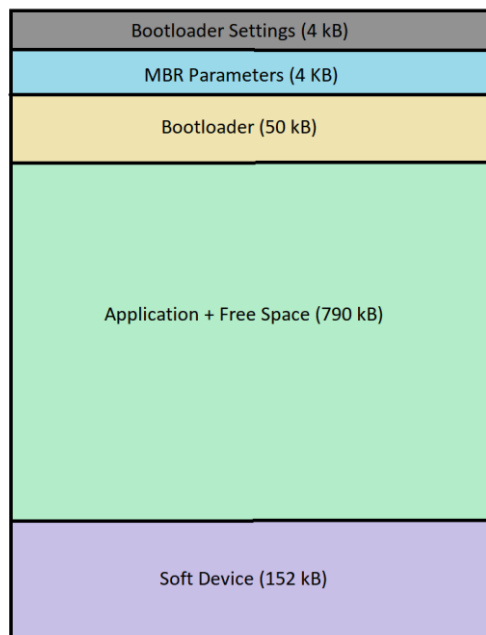


Figure 2: NOR Flash Memory Layout

The internal NOR Flash of the nRF52840 is divided as shown in Figure 2 to host the Bootloader, Watch application and the SoftDevice for BLE protocol stack.

## 2.2.3 Bootloader

The bootloader starts the Watch application that is located at a specific place in memory and initializes the device before the application is started. The most important feature provided by the bootloader module is the Device Firmware Update (DFU) functionality. Major features of DFU are:

- Updating the application, SoftDevice and bootloader
- Updates from authenticated source (signed updates)
- Various transports: (BLE, USB)

## 2.2.4 Watch Application

The Watch application is developed using the nRF52 SDK framework, having FreeRTOS and using the Softdevice BLE stack. The watch firmware architecture is shown in Figure 3.

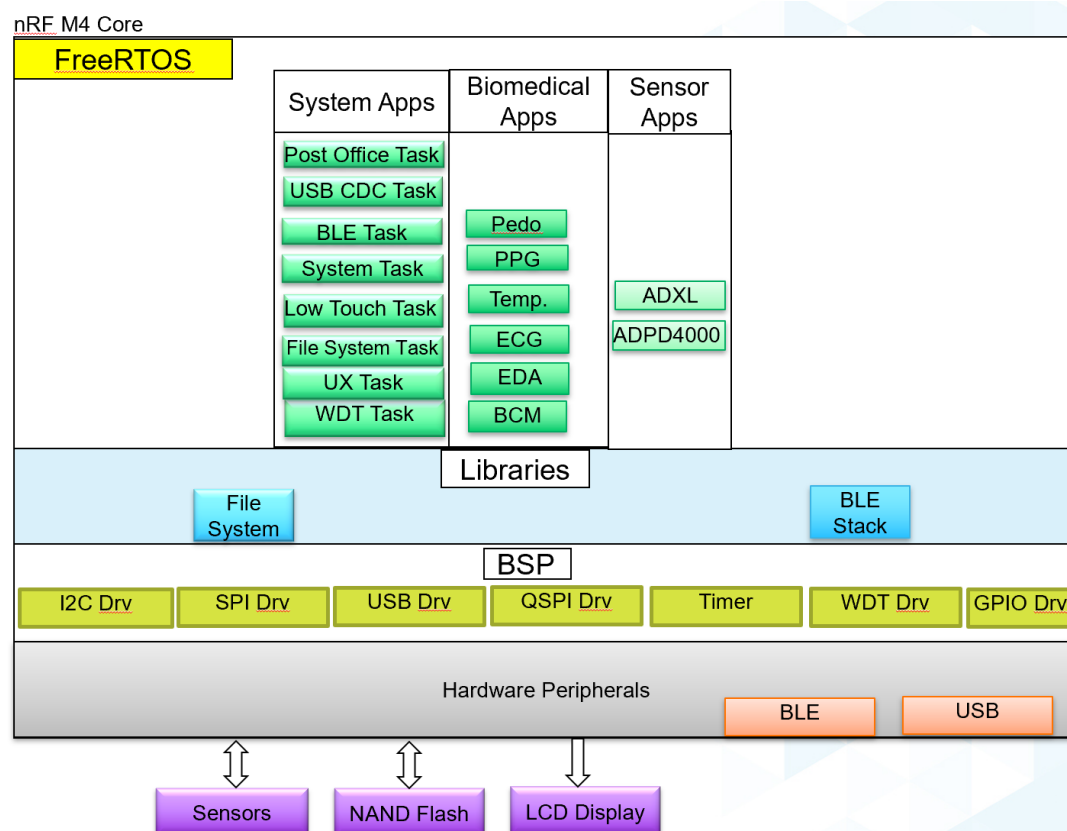


Figure 3: Firmware architecture

The watch application is based on FreeRTOS operating system with BSP and libraries provided by nRF SDK 15.2.0 version. The nRF52840 ARM Cortex M4 core interacts with the sensors, external NAND Flash and has BLE & USB interfaces for data communication with the PC and mobile applications. The watch application comprises of system, biomedical and raw sensor applications. It has a lightweight filesystem (LFS) to manage the log file collection to the NAND Flash. The watch

has the option to store application specific Device Configuration Block (DCB) within internal NOR Flash of nRF52840.

The interaction of the Watch application internals with the external Tools is as shown in Figure 4.

The Watch interacts with ADI's evaluation tool "Application Wave Tool" which supports connecting the Watch to the tool either via USB or BLE. The Watch can also be evaluated using iOS application provided by ADI. The CLI utility is used by the developers for developer level testing of the Watch. Using the external tools, the supported applications may be started to view the graphs or log collection maybe started/stopped/retrieved.

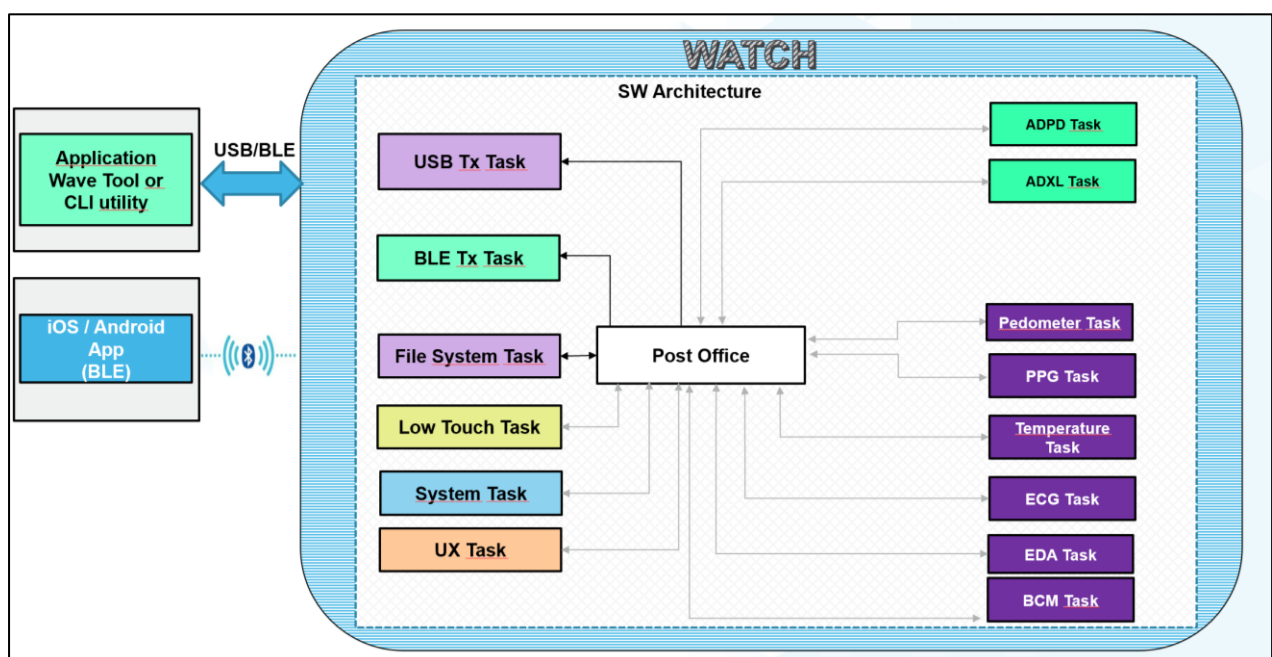


Figure 4: Tools and SW Interaction

Watch application firmware uses FreeRTOS Framework with ADXL, ADPD4x00, PPG, ECG, EDA, BCM, USB CDC ACM, BLE and Post Office as FreeRTOS tasks. Post Office task is a routing task, which routes the M2M2 packets coming from external tools correctly to the intended internal task. The external M2M2 packets are handled by the USB or BLE Tasks according to the medium used for communication.

## 2.2.5 M2M2 Protocol in VSM Watch

M2M2 stands for Machine to Machine Messaging Version2. This is a protocol from ADI designed to enable embedded systems to use the "micro-kernel" and "micro-services" system design architectures with minimal overhead and footprint. With this the system is broken up into well-defined applications, which are designed and implemented to be completely independent of each other. Applications interact by passing messages:

→ Point-to-point messages → Publish-subscribe messages

### 2.2.5.1 M2M2 Message Passing – Header Structure

M2M2 protocol uses an 8-byte UDP-style header (does NOT use a UDP stack). This has a variable length data field and the applications define their own payloads.

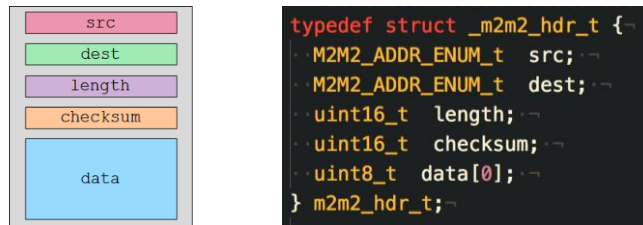


Figure 5: M2M2 header structure

All applications have an M2M2 address and packets are dynamically routed between applications - and across machines - based on addresses. Each application stream also has an address.

### 2.2.5.2 M2M2 Message Passing Implementation

M2M2 provides two entities used to perform message passing.

1. Post Office - Provides message routing within a processor
  - a. Messages are routed directly from one application to another
  - b. The message itself is not copied, a pointer is passed instead

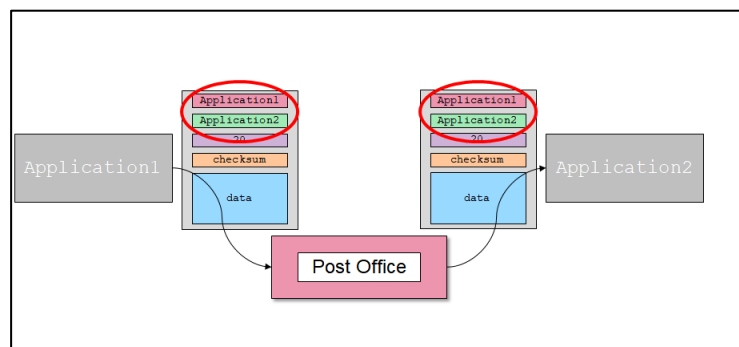
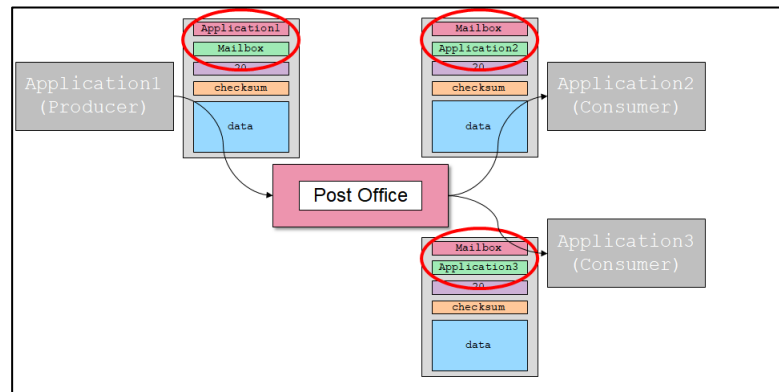


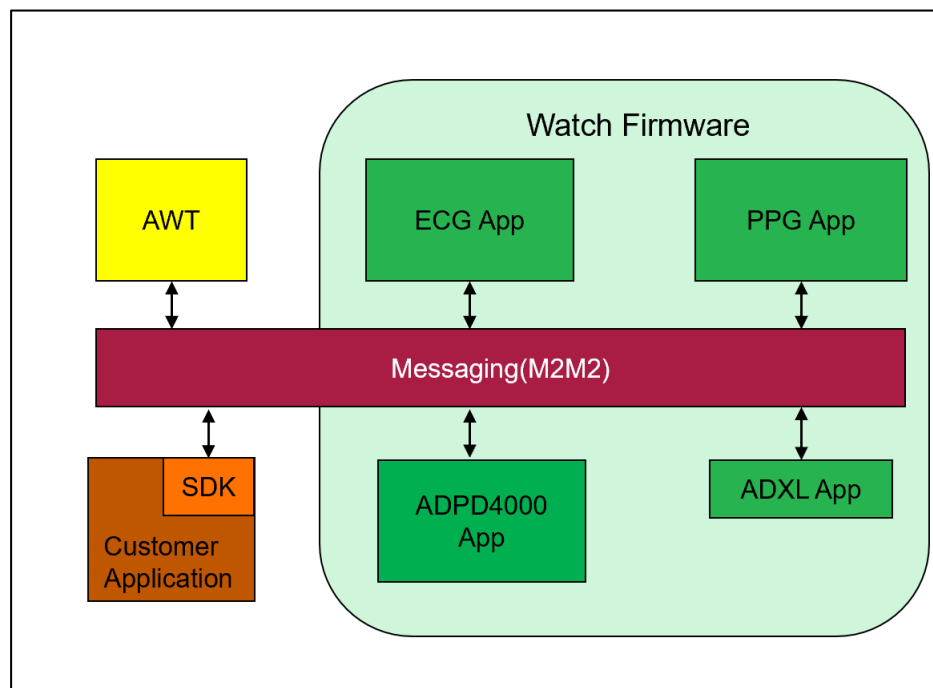
Figure 6: Point to point M2M2 Messaging using Post Office

2. Mailbox - Provides many-to-many publish-subscribe messaging
  - a. Messages are sent by one or more “producer” applications to one or more “consumer” application
  - b. Message flow is one-way
  - c. “Producer” applications send messages to a mailbox
  - d. Consumer applications send a message to the Post Office requesting that they be subscribed to a mailbox
  - e. “Consumer” applications receive asynchronous messages from the mailbox



**Figure 7: Publish-Subscribe M2M2 Messaging using Mailbox**

Figure 8 illustrates how Watch application interacts with AWT or a customer application developed using the Watch SDK and M2M2 messaging protocol.



**Figure 8: Interaction of Watch applications with AWT/SDK using M2M2 messaging**

## 2.2.6 BLE Protocol in VSM Watch

The study watch uses the SoftDevice – s140 from Nordic SDK, which is a precompiled and linked binary image implementing Bluetooth 5 Low Energy protocol stack for nRF52 series of SoCs. The Watch application uses the serial port emulation over BLE and includes the Nordic UART Service (NUS).

The Maximum BLE data throughput achieved from the Watch is 12 KB/sec at 10 meters.

To improve the performance throughput of the BLE connection and characterize the stable data rate, following parameters were changed in **Study Watch** firmware:

- BLE GAP data length: 251 (earlier it was 27) The SoftDevice handler will configure the stack with these parameters
- BLE Tx and Rx PHY: BLE\_GAP\_PHY\_2MBPS (earlier it was BLE\_GAP\_PHY\_AUTO)
- BLE Connection Interval: 7.5ms (earlier min and max connection interval was 10 and 40 ms). Now both min and max are set to 7.5ms and 35 respectively.
- BLE\_COMMON\_OPT\_CONN\_EVT\_EXT option set to one, which is the parameter for enabling extended connection events

Following changes were made to the **BLE Dongle**:

- BLE scan max connection interval: 7.5ms
- BLE scan min connection interval: 7.5ms
- BLE GAP data length: 251 (earlier it was 27) The SoftDevice handler will configure the stack with these parameters
- BLE\_COMMON\_OPT\_CONN\_EVT\_EXT option set to one, which is the parameter for enabling extended connection events

In BLE Tx path in firmware, packet combining is enabled upon entering the high data rate mode. High data rate mode is decided when the first STREAM\_SUBSCRIBE response is detected within packet to be sent out. High data rate mode exit happens when last STREAM\_UNSUBSCRIBE response from packet to be sent out is received. As of now, MAX\_TX\_PKT\_COMB\_CNT is 4 in high data rate mode and MIN\_TX\_PKT\_COMB\_CNT is 1 otherwise. The idea of packet combining is to make sure that the buffer submitted to ble\_nus\_data\_send() api is used maximally, otherwise NRF\_ERROR\_RESOURCES was seen.

## 2.2.7 Timestamps

Timestamps used in firmware serve the following purposes:

- Synchronization of data from multiple streams
- Synchronization of data with external sources
- Extraction of wall clock and calendar for samples
- Comparing the HR and other parameters with the reference devices

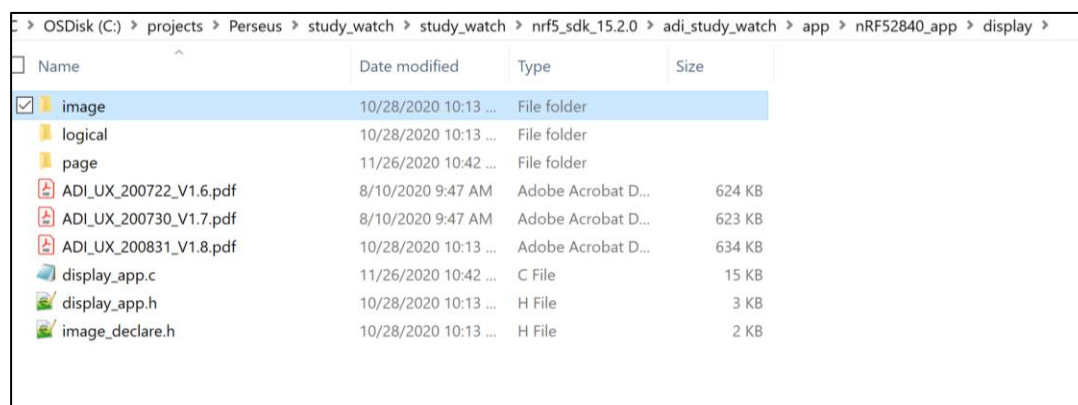
The resolution of timestamp is 0.03125ms but this value is divided by 32 to view it in 1ms resolution. This doesn't lose the precision as the timestamp is in double precision value. The timestamp tick starts from 1st January 2000. All the reference devices' ticks also correspond to this date or can be made to correspond to this date as it is standard practice.

**Note:** The tool has been modified in such a way that the timestamp information is in UTC time zone irrespective of whether the date and time is in UTC or local time zone. The UTC offset is used for this purpose.

## 2.2.8 Watch UX

The Watch UX can be used to view current status of the system such as battery status, NAND Flash used space, logging status and view waveforms.

Customisation of existing pages maybe done from the Watch application's source code structure, which is organised as shown below:



**Figure 9: Watch UX Source code directory structure**

Latest Watch UX design document would be captured as ADI\_UX\_XXXXXX\_xx.x.pdf document available in the above directory path. To customize any page in Watch Display, one can go to the page/ folder and edit the page\_XXX.c file.

## 2.2.9 Lightweight File System (LFS)

Light weight file system (LFS) is implemented for logging purposes, maintain records of files created in Table of Contents (TOC) block, and it also enables the download of logs. LFS uses 1MB of flash to store file version, footprint, configuration files for low touch and stored file information. The blocks on the NAND flash is organized as

Block0	Block1	Block2	Block3	Block4	Block5	-----	-----	Block2047
RESERVED BLOCK	TOC BLOCK	CFG BLOCK	TMP BLOCK	DATA BLOCK	DATA BLOCK	DATA BLOCK	DATA BLOCK	DATA BLOCK
Stores the LFS version and revision	Stores information on data files and bad blocks	Configuration Files	Used as temporary block during block erasure and update	Log data	Log data	Log data	Log data	Log data

Current version of LFS implements

- Circular buffer to write/erase data.



- This helps to evenly spread writing/erasing throughout flash, thus minimizing write/erase cycles for area of flash.
  - Circular buffer pointers hold previous written page/erased block information which helps in minimizing traversal times in finding good area to write resulting in improved writing efficiency.
- Bad block management
  - Holds list of bad blocks
  - Used to skip writing/erasing area whenever bad block is found from list.
  - List is dynamically updated when write/erase to area fails.
  - This implementation avoids accessing of drivers for each area before writing to check bad block, thus minimizing read time, and improving writing/erasing efficiency.

**Note:** If the contents in TOC block gets corrupted (which is a remote chance), then users can request for a watch recovery firmware from ADI through [healthcare-support@analog.com](mailto:healthcare-support@analog.com). The contents of the NAND flash will be erased by this recovery firmware and a new block will be assigned as TOC block.

## 3 Configurations and Settings

After a reset, the VSM Watch comes up with default configurations for all the sensors. There are default dcfg files (with Register settings) which are loaded on to the sensors when an application is started. This may be changed from Wavetool.

When a DCB is present for an application, it will take precedence over the default firmware configuration. DCBs for each application can be manipulated using Wavetool.

### 3.1.1 Device Configuration Block (DCB)

Each deployment will likely require a unique configuration that is selected by the researchers such that the end users will receive the watches ready-to-go when turned on.

For example, a deployment may require PPG and EDA only, while another deployment may require PPG, while a third may require PPG, ECG and Pedometer. In addition, the sensors may be tuned for each specific use case. It is important that for a specific deployment, the researchers be able to customize a group of watches. The watch has been designed to have a customizable start-up configuration.

The watch APIs are defined by M2M2 commands. Hence, the configuration is defined by a series of M2M2 commands that will place the device into a desired state.

The term device configuration block (DCB) is used to refer to the device configuration of sensors as well as application configuration of bio-medical applications.

The general configuration block contains a series of M2M2 commands – this is used by the LT application.

Other DCB blocks contain

- a. the register address-value pairs
  - ADI\_DCB\_ADPD4x00\_BLOCK\_IDX
  - ADI\_DCB\_ADXL362\_BLOCK\_IDX
  - ADI\_DCB\_AD7156\_BLOCK\_IDX
- library configurations
  - ADI\_DCB\_PPG\_BLOCK\_IDX
  - ADI\_DCB\_EDA\_BLOCK\_IDX
  - ADI\_DCB\_EDA\_BLOCK\_IDX
  - ADI\_DCB\_WRIST\_DETECT\_BLOCK\_IDX

The various DCB available in the firmware are:

DCB blocks used now	DCB blocks that will be used in future
ADI_DCB_ADPD4x00_BLOCK_IDX	ADI_DCB_AD5940_BLOCK_IDX
ADI_DCB_ADXL362_BLOCK_IDX	ADI_DCB_PEDOMETER_BLOCK_IDX
ADI_DCB_PPG_BLOCK_IDX	ADI_DCB_TEMPERATURE_BLOCK_IDX
ADI_DCB_ECG_BLOCK_IDX	ADI_DCB_UI_CONFIG_BLOCK_IDX
ADI_DCB_EDA_BLOCK_IDX	ADI_DCB_USER0_BLOCK_IDX

ADI_DCB_AD7156_BLOCK_IDX	ADI_DCB_USER1_BLOCK_IDX
ADI_DCB_GENERAL_BLOCK_IDX	ADI_DCB_USER2_BLOCK_IDX
ADI_DCB_WRIST_DETECT_BLOCK_IDX	ADI_DCB_USER3_BLOCK_IDX

The DCB is based on the FDS library provided by the nRF5 SDK.

For more information, refer: [Nordic Infocenter](#)

The data content in each of the DCB blocks is different and dependent on the DCB block index. Each DCB block, has a DCB entry with WORD size of 4 bytes in size. Allowable maximum number of words in each DCB block is also different between the DCB block indices. DCB read, write, delete access is given to the user through AWT. The template files for the DCB blocks maybe located in AWT installation directory, “*cfg/DCB\_Config*” path.

DCB block information is captured in the table below:

DCB blocks used now	Data content	Maximum entries	Template DCB file name
ADI_DCB_AD7156_BLOCK_IDX	register address-value pair	58	adpd4000_dcb.dcfg
ADI_DCB_AD7156_BLOCK_IDX	register address-value pair	25	adxl_dcb.dcfg
ADI_DCB_PPG_BLOCK_IDX	PPG lcfg	53	ppg_dcb.lcfg
ADI_DCB_ECG_BLOCK_IDX	ECG lcfg	2	ecg_dcb.lcfg
ADI_DCB_EDA_BLOCK_IDX	EDA lcfg	2	eda_dcb.lcfg
ADI_DCB_AD7156_BLOCK_IDX	register address-value pair	20	Not Available yet
ADI_DCB_WRIST_DETECT_BLOCK_IDX	LT App lcfg	4	wrist_detect_dcb.lcfg

**Table 2: DCB block information**

Once a DCB block is written, it will be persistent between watch resets, until it is explicitly deleted. The application would work with DCB configuration, if available (default firmware configuration is overridden)

DCB Settings option in AWT provides the facility to Write/Read/Erase of DCB blocks.

- Write DCB option lets to choose the DCB file to be written to the DCB block
- Read DCB option reads the DCB block and saves it as a file
- Erase option clears the DCB block, such that firmware default dcfg could be used thereafter.

The DCB option from the Multiview of AWT does the Write/Read/Erase of multiple blocks. Alert messages for DCB block present are provided by the AWT for each of the applications.

## 4 Firmware Applications and Packet format

### 4.1 Applications

The study watch comes with several applications integrated. These can be sensor applications which produce raw sensor data, bio-medical applications which processes the raw sensor data and generate vital signs/activity parameters such as heart rate, step count, low-touch application or system control applications.

#### 4.1.1 ADPD

This is the optical sensor application using ADI's ADPD4000/ADPD4100 sensor. It is configured to run in external sampling mode. Hence, based on the output data rate (ODR) set, it will be externally triggered by the MCU.

1. Output Data Rate (ODR): Default - 50 Hz  
ODR tested – 12.5Hz, 25 Hz, 50 Hz, 100 Hz, 250 Hz, 300Hz, 400 Hz, 500Hz
2. Slots: Out of 12 slots, first 6 slots (A to F) are enabled by default.  
Slot - D & Slot - E is configured to be used for temperature application and only channel-1 is enabled for it.  
Slot - F is configured to be used for PPG application with both channel-1 and channel-2 enabled.
3. Following slots are used for each LED -  
Slot – F: Green LED  
Slot – G: Red LED  
Slot – H: IR LED  
Slot – I: Blue LED  
For each of these slots, both the channels are enabled.
4. LED Pulse Count: Default – 64 pulses set for each of the four LEDs.

##### 4.1.1.1 AUTOMATIC GAIN CONTROL (AGC)

ADPD application includes the feature of automatic gain control (AGC) which helps in optimizing the ADPD signal. This feature is available for all the four LEDs - green, red, IR and blue. It involves collecting first 10 samples from both channel-1 and channel-2 of the configured slot(s) and optimizing the channel TIA gain and LED current for the slot(s). There is also an option for AGC recalibration, whenever user requires, which will recalculate and adjust the TIA gain and LED current of the LED slot(s), using a new set of 10 samples.

### 4.1.2 ADXL

This is the motion detection application using ADI's accelerometer ADXL362. It is configured to run in external sampling mode. Hence, based on the output data rate (ODR) set, it will be externally triggered by the MCU.

1. Measurement Range – Default: +/-8g  
Other supported range: +/-2g, +/-4g
2. Output Data Rate (ODR) – Default: 50 Hz  
ODR tested – 12.5Hz, 25Hz, 50Hz, 100Hz, 200Hz, 400Hz

### 4.1.3 Electrocardiogram (ECG)

This is the ECG sensor application and it is responsible for configuring the AD5940 + AD8233 with certain configuration options.

1. SAMPLING FREQUENCY - This is specific to AD5940 application.  
Sampling frequencies tested - 100Hz, 200Hz, 300Hz, 400Hz, 500Hz, 1000Hz.
2. PGA Gain – This is specific to the AD5940 configuration  
PGA gain tested - 1, 1.5, 2
3. AFE Power Modes - Power Modes for AD5940 + AD8233 is tested - Low power mode.

### 4.1.4 Electro Dermal Activity (EDA)

This is EDA skin impedance measurement application and it is responsible for configuring the AD5940 with certain configurable options. The default excitation frequency used is 100Hz while the sampling frequency is 400Hz.

1. Output Data Rate (ODR) - ODR of EDA to be set.  
ODR tested – 4Hz,8Hz,16Hz,25Hz,30Hz
2. DFT Number - Number of frequency points/resolution of measurement.  
DFT points tested - 8,16,32  
If EDA Sampling frequency is set > 16, default DFT Number is 8.
3. Decimation Factor - Number of packets to be sent or effective sample rate of packets.  
Decimation factors tested - Decimation factor of 1/2.
4. Dynamic Scaling - Auto scale Enable -> RTIA Auto Calibration Enable / Disable  
If RTIA Auto scale Enable
  - Min scale → Minimum scale of resistance to be calibrated.
  - Max scale → Maximum scale of resistance to be calibrated.
  - Lpriasel → Resistance mid vale between Min and Max scales.  
Measurements are normally calibrated from Lpriasel to max scale

### 4.1.5 Bio Impedance Activity (BIA)

This is BIA body impedance measurement application and it is responsible for configuring the AD5940 with certain configurable options

1. Output Data Rate (ODR) - ODR of BIA to be set.  
ODR tested – 4Hz,8Hz,16Hz,20Hz,25Hz
2. PGA Gain - This is specific to the AD5940 configuration.  
PGA gain tested - 1, 1.5, 2
3. AFE Power Modes - Power Modes for AD5940 is tested - Low power mode and High-power mode.
4. DFT Number - Number of frequency points/resolution of measurement.  
Frequency tested points - 8192, 16384
5. Sine Frequency setting - Frequency Tested – 50KHz

### 4.1.6 Low Touch (LT) Application

Low touch (LT) allows for a few unique features when using the watch. It essentially allows the watch to be preconfigured to automatically log specific data streams if the watch is on the user's wrist – when it is off the wrist, the device will stop logging and then go to a lower power state. This mode is used to start/stop logging without the use of button/tool.

#### 4.1.6.1 LT Config file creation

The LT application uses the capacitive sensor AD7156 for wrist detection. The configuration of the AD7156 will be available either through a default configuration within the firmware or a customer configuration to be put within ADI\_DCB\_AD7156\_BLOCK\_IDX – such that the device can be tuned to specific customer requirements.

The LT application makes use of either the DCB block, ADI\_DCB\_GENERAL\_BLOCK\_IDX in the NOR flash or the USER\_CONFIG.LOG present in the NAND Flash to run the preconfigured M2M2 commands upon wrist detect ON and OFF. When both the DCB config as well as USER\_CONFIG are present, the former config takes precedence.

AWT is to be used to write the LT configuration file – either the DCB file or the NAND config file onto the Watch. Deletion of the config file is also done with the AWT. When Watch is turned ON, the Low Touch application also gets launched, if the config files are present. This means that with Wrist ON or Wrist OFF detected, LT logging also happens for the combination as put in the config files. LT application can be configured from the LT application page on the AWT.

Currently the config files supported from the AWT for LT application is for the various use-cases UC1,2,3,4,5 supported from the MultiView. AWT allows the user to select any of the five multi view use-cases with different sensor streams and their configuration settings such as mode of operation,

sampling frequency etc. AWT converts the user selected options into corresponding m2m2 commands and loads into the NAND flash/DCB of the watch.

Once LT config file is written into the Watch from LT Logging View of the MultiView, the user can disconnect from the WT, to do further logging tests. On connecting a Watch to AWT, it detects if LT application is active. User should accept to switch Watch to normal mode to continue using the AWT. The LT config files (both ADI\_DCB\_GENERAL\_BLOCK\_IDX and USER\_CONFIG.LOG) get deleted when the Watch switches to the normal mode. Switching to normal mode, from LT mode fails, if LT logging is in progress. Watch needs to be taken off the wrist, to stop LT logging. If even after removing the Watch, LT logging continues, this can happen if LT tuning is not proper – user can disable LT app from the Watch display. From LOW\_TOUCH\_LOGGING page in Display and within that the LT APP sub-page, press select button to disable the LT Application. After this connect Watch to AWT and accept to switch to Normal mode. If there is ADI\_DCB\_WRIST\_DETECT\_BLOCK\_IDX present, user option will be taken whether to delete this DCB or not. More details on ADI\_DCB\_WRIST\_DETECT\_BLOCK\_IDX is given in section 4.1.6.4

#### **4.1.6.2 Wrist on/off detection**

The on or off wrist event detection on watch works on the principle of capacitive sensing. When the watch is placed on the skin, the capacitance value goes low, and goes high when it is taken off from the skin. For objects like table, books, mouse pad, metal, wood (except for objects like static shield bags/covers) etc., there will be slight incremental change in the capacitance value, but it is considerably less compared to that of change coming from human skin. Based on this change, firmware will detect whether the watch is on wrist or not. To avoid interpreting the spurious capacitive signals during the watch on/off wrist event, time limits have been put to filter out such signals. The watch must be placed on wrist for at least 5 seconds to consider it as valid watch-on-wrist event. Similarly watch must be placed off the wrist for at least 3 seconds to consider it as valid watch-off-wrist event.

Once the valid watch-on-wrist event is detected, firmware will send the command to start the low touch logging. Config file from the DCB/NAND flash is read and then it checks if the memory is full or maximum file count is reached (Logging will not start if memory is full or the maximum data file count of 62 is reached). If all prerequisites are met, then it starts sending and handling the m2m2 commands from the config file read.

Once the valid watch-off-wrist event is detected, firmware will send the commands already read from the config file, to stop the low touch logging.

Error conditions during the start or stop of the low touch logging will be indicated on the Watch display. The error response will be sent to the tool if it is connected.

#### **4.1.6.3 LT app in bootup sequence**

While booting up, the firmware checks whether the config file is stored on the DCB/NAND flash. If the config file is found, then the watch will configure itself in low touch mode, else it continues with normal mode.

Once watch configures in low touch mode, the firmware will start logging the sensor streams mentioned in the config file once the watch-on-wrist event is detected.

#### 4.1.6.4 LT Application Tuning

The wrist detection is based on capacitance change that is detected by the touch sensor, at watch bottom and connected to AD7156. There would a change in capacitance, when watch is placed on wrist from where it was kept initially – another surface like air or a table. This difference in capacitance is not a fixed value and varies between person to person and even from one watch to another. To improve the accuracy for wrist detection and thus to improve the LT application, the latter works with a default firmware lcfg, which is reconfigurable.

Current lcfg parameters for LT Application is captured in table below:

Index	Field	Default value in firmware	Remark
0	LT_APP_LCFG_ONWR_TIME	5 sec	Time for the watch to be kept On Wrist to start the LT logging
1	LT_APP_LCFG_OFFWR_TIME	3 sec	Time for the watch to be taken Off Wrist to stop the LT logging
2	LT_APP_LCFG_AIR_CAP_VAL	1380 uF	Capacitance value sensed by bottom touch sensor when watch is placed on Air
3	LT_APP_LCFG_SKIN_CAP_VAL	1340 uF	Capacitance value sensed by bottom touch sensor when watch is placed on Skin

Firmware provides support to change the default firmware lcfg or write this to DCB as ADI\_DCB\_WRIST\_DETECT\_BLOCK\_IDX. Following the instructions from AWT, LT Application can be tuned to make it work consistently and accurately for that person from that Watch. If tuning has been done to modify the default lcfg, it would be lost after a Watch reset. Adding the LT application lcfg into the DCB, would retain it even after a Watch reset. Having a DCB configuration would override the default firmware lcfg.

To use the LT Tuning View from the AWT, the user needs to know what is the air capacitance and skin capacitance which needs to be configured. For this, the user needs to connect Watch with AWT via BLE, go to MultiView, select the Low touch logging icon, enter the Tuning View and follow the instructions there. From Tuning view, when Read button against the Current capacitance box is pressed, it reads the LT capacitance. Note down the air capacitance by placing the Watch face in air, without touching bottom touch sensors and then press the read button next to Current Capacitance box, to get the reading for the air capacitance. And similarly, skin capacitance by placing the Watch bottom touch sensor on the wrist with straps tightened. It is suggested to enter the minimum value seen, of some 10 trials for the air capacitance and maximum value seen, of 10 trials for the skin capacitance. These values can be entered from the LT Tuning view of AWT, to fine tune the LT application to make on wrist and off wrist reliable.



### 4.1.7 System application

System application is a house-keeping application in the watch firmware. It sends out periodic battery information streams, has the system information along with BLE MAC address, is a place holder for low touch application – wherein it gets current status of all other sensor application and initiates the start or stop command sequence of Low Touch application. It also includes handling the control commands for setting power state of the watch, resetting the flash, resetting the watch, getting chip ID of various chips in the watch, entering bootloader mode, etc.

## 4.2 Packet format

Each application/stream in the study watch has a different packet structure and it depends on the kind of data that is to be send out. The details of the packet structure are available in the interface files for each application/stream. The table below details the packet size of each application/stream.

Application	M2M2 Packet Size(bytes)	Sample Rate(Hz)	Samples/Packet	Bytes/second
ADPD4K	44	50	2	1100*
		Above 50	6	733**
ADXL	55	50	5	550
Syncppg	72	50	4	900
PPG	46	1	1	46
Temperature	20	1	1	20
Pedometer	23	1	1	23
ECG	61	100	11	555
EDA	61	4	6	41
BCM	65	1	1	65

**Table 3: Packet Format**

\* → The rate given for ADPD4K is for one channel of one slot. The rate will increase linearly with increase in slots and channels. Eg:- At 50Hz, for 2 channels of one slot, the rate will be 2200bytes/sec, while for two slots with both channels, it will be 4400bytes/sec.

\*\* → calculated for 100Hz

ADPD	ADXL	Syncppg	PPG	Temperature	Pedometer	ECG	EDA	BCM
50	50	50	50	1	25	100	4	20
100	100					200	8	
200	200					250	16	
250	400					300	25	
300						400	30	
400						500		
500						1000		
1000								

**Table 4: Supported Sampling Rates (Hz)**

## **5 Embedded Algorithms on Watch**

The study watch comes embedded with couple of algorithms, which process the raw data obtained from sensor applications to output certain vital sign parameter(s).

### **5.1 Heart Rate Monitoring (HRM) algorithm**

This algorithm measures heart rate using PPG/ADPD signal while removing the motion-based interference. It operates on a single channel PPG/ADPD signal, together with 3-axis accelerometer data to produce the heart rate. The algorithm is provided as a pre-built Cortex-M4 library along with header file. It is designed to work with PPG and accelerometer data at 50Hz.

### **5.2 Pedometer algorithm**

The pedometer algorithm processes the data from the accelerometer and determines the step count. Since the algorithm requires data at 25Hz, the incoming accelerometer data is decimated to get this rate.

### **5.3 Signal Quality Index (SQI) algorithm**

PPG signals collected via wearable devices are prone to noise sources and other artefacts that negatively impact the measurement accuracy of the sensor. The Signal Quality Index (SQI) algorithm gives a score (index) for each time window/segment of PPG data in order to determine if it is of a sufficiently high quality to be useful for other vital sign extraction or clinical diagnostic algorithms estimate of the heart rate. The SQI feature is supported for PPG signal frequencies ranging from 25-100Hz. The SQI score is a floating-point value between 0 (poor signal quality) and 1 (excellent quality). The WT has option to display the SQI in all use-case views. The SQI can be calculated on any of the Green, Red, IR or Blue LEDs present on watch.

### **5.4 ECG Heart Rate algorithm**

This algorithm measures heart rate from the ECG signal by detecting the QRS peak of the ECG signal. The algorithm is provided as a pre-built Cortex-M4 library along with header file. It is designed to work with ECG signal with ODR up to 200Hz.

## **6 FAQs**

This is divided into various sections:

[Watch Usage](#)

[Battery](#)

[NAND Flash](#)

[Sensor/Bio-Medical Applications](#)

[Accelerometer](#)

[Photo Plethysmography](#)

## 6.1 Watch Usage

### *What are the steps to be followed when a connection failure to the watch is observed?*

The watch can be connected to a tool using either BLE or USB cable connection.

#### **BLE connection:**

Once the watch is reset, the tool tries to scan for BLE devices and gives a list of the MAC ID of various watches in the vicinity. The user must select the right MAC ID of their watch to complete the connection. If the user is not sure of the MAC ID, then the watch can be reset when one of the MAC IDs from the list will be removed. Once the watch comes of reset, its MAC ID will be displayed. The user can see the MAC ID of the Watch from LCD Display, by going to **Settings page → MAC Address** page. The user can then connect to this MAC ID.

#### **USB connection:**

The user must connect the USB micro cable to establish an USB connection to the watch. Check the COM port that is identified with this connection using a HyperTerminal such as Tera Term or by observing the new COM port displayed in Device Manager-->Ports (COM & LPT) of the host machine. The user must select this COM port from the list of COM ports displayed by the Application Wavetool.

### *If the log file is corrupted while streaming, then what should be done?*

Try to retrieve the file again but ensure that the host machine is not doing any other task during streaming.

### *How can the watch be shut down using the buttons?*

From LCD Display, navigate to **Settings page -> Power Off** page, press **Select** button and then navigate to **Yes** or **No** for confirming. Press **Select** button once more.

## 6.2 Battery

### *How long the watch must be charged?*

Before using the watch for the first time, it must be charged for approximately 1 hour for the battery to get fully charged.

## 6.3 NAND Flash

### *What is the maximum size allowed to store in NAND flash?*

The NAND Flash has a storage capacity of 512MB. The maximum effective size for data files storage is 511MB. The initial 1MB is used for storing certain system level parameters and information.

### *How long can we log the data into flash?*

The amount of time that data can be logged to flash depends on the application(s). The table below shows the time for few of the applications (assuming the battery is being charged).

Application(s)/User-Case	Time (hrs)
Use-case1 (ADPD@500Hz, ADXL@50Hz, Temp@1Hz)	18.83
Use-case2 (ADPD@100Hz, ADXL@50Hz, EDA@30Hz, Temp@1Hz)	63.55
Use-case3 (ADPD@100Hz, ADXL@50Hz, ECG@250Hz, Temp@1Hz)	43.48
Use-case4 (PPG@50Hz, ADXL@50Hz, ECG@1000Hz, Temp@1Hz)	22.86
Use-case5 (ADPD4k @100Hz from Slot F,G,H,I - 2 Channel Data)	25.37
ADPD@50Hz – one slot and single channel	135.31
ADPD@100Hz – one slot and single channel	202.96
ECG@100Hz	268.40
EDA@30Hz	488

Table 5: NAND Flash Usage

***What is the maximum number of files that can be stored on the NAND Flash?***

The maximum number of files that can be stored on the NAND Flash is 63. This is irrespective of the space left on the NAND Flash. In case of Low touch, this limit includes the configuration file also, i.e 62 data files + 1 configuration file.

***How are the filenames on the NAND flash created?***

The current file system uses 8.3 naming. This limits the length of the possible filename. File naming convention for NAND logging is: 31-12-2012 23:59:59 - MMDDTIME <2B2B4B>, we get new file named every 10 seconds. Unit value for seconds was neglected to get unique time values in hex within 4byte.

***Can a log file be removed from a list of log files on the NAND Flash?***

A single log file on the NAND Flash cannot be removed. Instead the NAND Flash can be formatted to remove all log files on it.

***How can the user inject a subject ID in the log?***

There is a provision in tools in the logging panel for the user to enter a 16-character subject ID which will be prefixed with a fixed key ID. This can be found at the start of the CSV log. The user can differentiate a log among a set of logs using this unique ID.

## 6.4 Sensor/Bio-Medical Applications

### 6.4.1 Accelerometer (ADXL)

***What is the measurement unit for accelerometer raw data (seen in Tools View) and the converter/multiplier to convert accelerometer raw data into unit m/s<sup>2</sup> (9.8g)?***

The ADXL 362 gives results directly in 'g'. It has options to be configured in +/- 2g, +/- 4g or +/- 8g. And with the resolution and the value obtained from acceleration (say A) it can be converted to m/s<sup>2</sup> as shown in table below, where value in m/s<sup>2</sup> is obtained as  $(A * \text{Resolution} * 9.8)$

Range	Resolution	Value in m/s <sup>2</sup>
+/-8g (Default on ADXL View)	4mg/LSB	$A * 0.004 * 9.8$
+/-4g	2mg/LSB	$A * 0.002 * 9.8$
+/-2g	1mg/LSB	$A * 0.001 * 9.8$

Table 6: Accelerometer ADXL362 range

## 6.4.2 Photo Plethysmography (PPG)

### *How is the PPG and acceleration data synchronized?*

The PPG sensor (ADPD) and the acceleration sensor (ADXL) are synchronized by hardware on the PPG application. The PPG algorithm requires the same sampling frequency on both devices to perform properly under motion conditions.

The ADPD and ADXL devices are configured to run in external sampling mode. The MCU triggers both the devices based on the ODR configured. Since the trigger pulses are generated from a single clock source, both are synchronized. ADPD and ADXL are also tested for synchronization when they are integer multiples of the source clock.

ADPD has been tested at ODR values of 12.5,25,50,100,250,300,400,500 and 1000Hz.

ADXL has been tested at ODR values of 12.5,25,50,100,200,400Hz