

HOT SDK Capability Document

Table of Contents

1. Overview	2
2. Hardware Control:.....	2
3.Commands:.....	2
3.1 Display	2
3.1.1 Display Organization	2
3.1.2 Graphics Functions.....	6
3.1.3 Text Display functions	6
3.2 Vibration Motor	7
3.3 LED Light:	7
4. Notifications.....	8
4.1 Touch:	8
4.2 Accelerometer	8
4.3 Gyro-meter	8
4.4 Bluetooth State:	8
4. Phone Apps:.....	8
5. Watch Faces	9
5. 1 Digital Watch Faces	9
5.2 Analog Watch faces.....	10
6. Watch Canvas Applications	10
7. Watch Native Apps.....	10
8. App Examples.....	10
8.1 Watch Face Example: Flight Clock Watch-face	10

8.2 Fitness Apps.....	10
-----------------------	----

The Hot Smart Watch SDK allows application developers to create watch native apps as well as connected smartphone apps to utilise the power of HOT Smart Watch and create watchfaces and applications.

1. Overview

The SDK is divided into Phone apps SDK and Native watch apps SDK. The phone apps provide less hardware interactive applications that are developed and that are run in the connected phone. The Native watch apps run in the watch and provide more efficient control and communication with the hardware.

The information from the app to watch are called **Commands** while the information from the watch to app is called **Notifications**. The Phone SDK wraps these communications details and provides an easy interface for the application developer. In case of native watch apps these **Commands** and **Notifications** are exchanged directly between watch and OS.

2. Hardware Control:

The apps can control Display, Vibration Motor and LED light. The apps can get input from Touch, Accelerometer/Gyro and Bluetooth state changes.

3. Commands:

3.1 Display

The apps can display information on the screen either at pixel level or higher level graphical objects or with native font text display. The watch screen display is monochrome display with resolution of 144x168 pixels with top left of the screen as the 0,0 coordinate. Each pixel is a single bit. A full screen data takes about 3024 bytes of data (144x168/8). The watchfaces are allowed to use the full screen display of 144x168, while the other apps can utilize non-title area of 144x150pixels

3.1.1 Display Organization

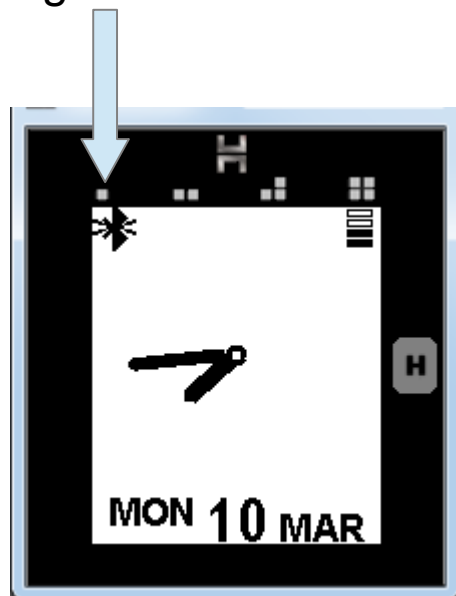
The main Display screens of the watch are organized in a 5*5 matrix as shown below. Each column is dedicated for a specific type of application.

Note: In addition to these screens, the watch provides advanced menu for phone, messaging, utility etc. These screens are not available for SDK.

Analog Watch Faces (AWF)	Digital Watch Faces (DWF)	Info Screens (IS)	Custom Apps (CA)	Custom Apps (CA)
1	1	1	1	6
2	2	2	2	7
3	3	3	3	8
4	4	4	4	9
5	5	5	5	10

- First column is dedicated for Analog watch faces. This can be accessed by clicking on the top left icon. Once entered in this category, user can browse up-down for different watchfaces by swiping up and down. User can load custom watchface in non-default locations (currently 3 locations)

Analog Watch Faces



- Second column is dedicated for Digital watchfaces. This can be accessed by clicking on the second icon. Once entered in to the category user can browse up-down for different digital watchfaces by swiping up and down. User can load custom watchface in non-default locations (currently 3 locations)

Digital Watch Faces



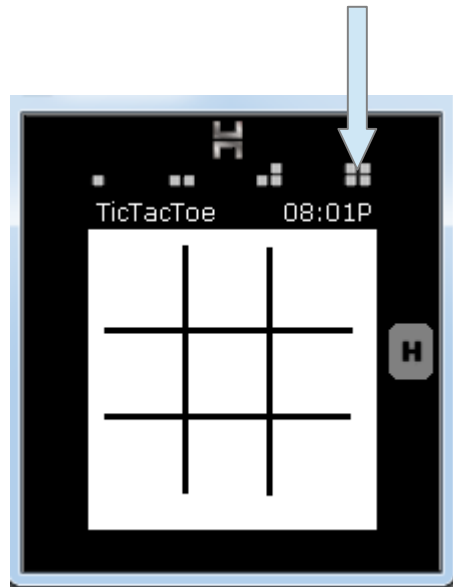
- Third column is reserved for info screens. This can be accessed by clicking on the third icon. Once entered in to the category user can browse up and down between different info screens like stock quote, News feed, weather etc. by swiping up and down.

Info Screens



- Fourth (and Fifth) columns are reserved for Custom applications. This can be accessed by clicking fourth icon (sliding right for 5th slot). Once entered into this category user can browse up and down, right and left to access different custom applications.

Custom app screens



3.1.2 Graphics Functions

The default screen color is white (1). ClearScreen function set full screen in to white. Native SDK Text handling functions by default draw black letters on the white background. Graphics functions are available both in Phone Apps SDK and Native Watch SDK.

Native watch SDK Graphics Function Examples

Bool PutPixel(int instID,int x,int y,Color col): sets pixel at x, y to the given color. Color is either BLACK or WHITE

void ClearScreen() : clears the full screen to white 1.

Rect(short xs, short ys, short width, short height,color col) : Sets the area to the given color

Boolean DrawRect(DWORD instID ,Rect rect,Color boderCol, int thickness,Color fillCol): Draws a line from xs,ys to xe, ye in the given color

3.1.3 Text Display functions

These functions are used to display text on the watch screen. Native SDK functions accept simple parameters to determine the type of the supported texts. Whereas certain Phone SDK functions may

accept font objects to obtain more control over the text fonts. Native SDK support 3 different sizes for fonts they are Small (9 point), Medium(13 point) and Large(14 point). Each type support both short and tall variation. Below given table shows the variants. Text Display function are available in both Phone SDK and in Native Watch SDK.

FONT_SMALL_NORMAL	9 Point Font Verdana
FONT_SMALL_TALL	12 Point Amplitude
FONT_MEDIUM_NORMAL	13 Point Font Verdana
FONT_MEDIUM_TALL	18 Point Amplitude
FONT_LARGE_NORMAL	14 Point Font Verdana
FONT_LARGE_TALL	20 Point Amplitude

Native SDK Text Function Examples

TextStyle DispTextLine(DWORD instID, char mesg,short xS, short yS, short width, short height, FontType type, Align alignStyle)

Draws a single line text. The height and width have to be larger than font size.

FontType: Enum Value to select the font type. Refer above table

AlignType: Enum Value to select alignment. LEFT- Left align, CENTER- Center align, RIGHT- Right align

3.2 Vibration Motor

The watch vibration motor is used for alerting the user. About 5 patterns of alerts are used for different indications. Other than these per-assigned types, there are three user assignable patterns reserved to implement application specific notification. Vibration Motor Functions are available in both Phone SDK and Native Watch SDK.

Native Vibration Function Examples

Boolean VibratePattern(VibrationType type, int Duration)

VibeType: Enum value to select the vibration type. See below list

Duration: 0: pattern is run once, pattern is repeated this many times (max 5 times)

VIBE_INCOMINGCALL	Medium pattern (used for incoming call)
VIBE_MESGALERT	Medium pattern (used for message alerts)
VIBE_PHONELOST	Long spin (used for Phone lost etc)
VIBE_BTCONNECT	Small spin (used for BT connect etc)
VIBE_TOUCH	Very Small spin (used for single tactile touch)
VIBE_USER_SHORT	Short Pattern reserved for user application
VIBE_USER_MEDIUM	Medium Pattern reserved for user application
VIBE_USER_LONG	Long Pattern reserved for user application

3.3 LED Light

The watch comes with LED backlight. The curve model also has an additional flash light. SDK functions allows user to control Back Light and Flash Light. LED Light Functions are available in both Phone SDK and Native Watch SDK.

Boolean LEDBackLight(int onSeconds, Brightness bright)

The backlight is switched on for onSeconds (max 20 seconds) with brightness (BRIGHT_DIM: 1/4th, BRIGHT_HALF: 1/2, BRIGHT_FULL: Full brightness)

Bool LEDFlashLED(int onSeconds, Brightness bright)

The flashlight is switched on for onSeconds (max 10 seconds) with brightness (BRIGHT_DIM: 1/4th, BRIGHT_HALF: 1/2, BRIGHT_FULL: Full brightness)

4. Notifications

4.1 Touch:

Every finger touch generates an interrupt in the watch hardware. Every touch includes Touch down, Touch move and touch up. Single touch will have touch down and touch up from same point. The Pattern touch will have touch down and touch up at different spots with in-between pattern points. X, Y locations of the touch correspond to X, Y of the display pixels. The icon area touch is not exposed to SDK.

User applications can subscribe to touch notification using SDK functions. SDK delivers the touch notification and corresponding co-ordinates to the call back functions registered through SDK calls.

Call back notifications delivers up to 16 touch points. For long touch and drag it sends maximum 16 points where 0th element represent first touch-down and 15th element the touch up, 1-14 in-between drag points. Touch Notifications are available in both Phone SDK and Native Watch SDK.

4.2 Accelerometer

User application can subscribe to accelerometer events and taps. SDK deliver events to the call back functions at the defined sampling frequencies. Taps detect taps in two directions. Acceleration is detected in all X,Y,Z axis. Accelerometer Notifications are available in both Phone SDK and Native Watch SDK.

4.3 Gyroscope

Gyroscope provides angular position in X,Y,Z axis. User applications can subscribe to Gyro events by registering a call back. User can select a sampling rate (10Hz,25Hz,50Hz,100Hz) and number of samples expected in each batch. Gyro Notifications are available in both Phone SDK and Native Watch SDK.

4.4 Bluetooth State:

Knowing the status of connection is important for most communication applications that exchange data with phone. SDK provides functions to poll the Bluetooth status as well as to register call backs to receive Bluetooth Status. Bluetooth Notifications are available in both Phone SDK and Native Watch SDK.

4. Phone Apps:

These apps are developed in Android (Java) and iPhone (objective C) latforms and run on the smart phone. These apps send and receive messages to/from watch to communicate with the hardware. The data speed between the app and watch is typically about 3-5KBytes/sec. Because of the data speed limitation and higher power consumption in the watch due to the data communication, these apps are typically less interactive and lower data driven.

5. Watch Faces

The watch face app tells the watch on how to layout the time information on the screen. Once information format is sent to watch, the watch manages the time update on the watch screen. The time information can have any of the below data

Time: HH(12/24), MM, SS, AM-PM

Date: YY(2 or 4 digits), MM (number or 3 digit Month), DD

SDK provides a set of functions to create a new watch face. As shown in the screen layout figure above, user can select a free slot available in the first two columns depending on the type of watchface they want to create. Calling watchface SDK functions creates meta-data which is then downloaded to watch. Watch OS interprets this meta-data and create different watch faces.

The Watchface SDK is available only in phone apps SDK and not in native SDK

5. 1 Digital Watch Faces

The digital watch faces display all the information in numbers and alphabets using pre-existing fonts supported on the watch. They can have background image, plane (default) or inverted backgrounds.

Icons Supported for Watch faces

Type	Code	Size pixels (WxH)	Font	Image
Number Digits				
0-9 digits	1	8 x 14	Arial Reverse	0
0-9 digits	2	16 x 24	Arial	0
0-9 digits	3	8 x 14	Arial	0
0-9 digits	4	20 x 35	Arial Reverse	0
0-9 digits	5	34 x 64	Arial	0
0-9 digits	11	16 x 16	Digital Reverse	1
0-9 digits	12	24 x 24	Digital	1
0-9 digits	21	32 x 48	7Segment	0
0-9 digits	22	24 x 32	7 segment	0
Weekday				
3 digit Sun to Sat	1	42 x 16	Arial	SUN
3 digit Sun to Sat	11	48 x 18	Digital	SUN
3 digit Sun to Sat	12	72 x 25	Digital	SUN
Weekday full	13	144 x 18	Digital	SUNDAY
Weekday Full	21	144 x 20	7 segment	MONDAY
Month				
3 digit Jan to Dec	1	42 x 16	Arial	JAN
3 digit Jan to Dec	11	64 x 32	7 segment	JAN

5.2 Analog Watch faces

Analog watch faces include hour and minute needles running on top of a background image. It can also have window areas for day, weekday, month or any other information display.

6. Watch Canvas Applications

These apps run on the phone and use the watch screen as a canvas for display information and as a touch input device. The powerful SDK function makes it easy to create complex forms and display it on the watch screen. Every touch on the watch will generate a call back for this app for further action by the app. They can display raw pixel data or display higher objects such as icons, buttons, text etc.

7. Watch Native Apps

These are apps that are written in C and run in the controller of the watch. They are typically graphically intensive apps that require quick user interaction. They can be games or utility apps that interact with the user and present data to the user.

8. App Examples

8.1 Watch Face Example: *Flight Clock Watch-face*

The flight clock app will display Origin time, Destination time, Time to flight, Flight Time, Destination weather etc. The app runs in the phone and displays this information on watch

- The app gets the flight details from the airline ticket number and extracts the origin, destination and flight time details
- Display origin city string “DFW:”
- Display origin time in line 1
- Display destination city string “NYC”
- Display destination time in line 2
- Display time to flight (or flight time once the flight starts)
- Display destination weather string “Temp:42F Rainy”

8.2 Fitness Apps

The fitness apps collect accelerometer/gyro information from the watch and process the data for useful functions. A phone app can collect this information and use this data to present fitness statistics (calories per day) or sport improvements (golf swing analysis).

9. SDK Deployment Schedule

The HOT watch SDK will be deployed in following phases with timeline announced shortly.

SDK Phase	Timeline
Essential Functionality Android Phone App SDK	April 30 th 2014
Android SDK with Windows socket based simulator support	May 15 th , 2014
Essential Functionality Watch Native App SDK	May 30 th , 2014
Watch firmware version with SDK essential SDK support	June 15 th , 2014
Full Functionality Android Phone App SDK	June 30 th , 2014
Full Functionality Watch Native App SDK	July 30 th , 2014
Full Functionality iPhone Phone App SDK	Aug 30, 2014

