



Event Messaging System

Document History

Version	Date	Description
1.1		Initial Draft.
1.2		Worn status now sent as % value in worn status change event. Added Fall Detect Event
1.3		Added vertical jump event
1.4		Added dash event
1.5		Added Self Test Result Event
1.6	15-12-2011	Reformat; Initial Release

References

Ref #	ID	Description
[1]		

Document Notes

All numbers in this document are written in decimal, except hexadecimal numbers which are prefixed by '0x'. For example 5436 is decimal, while 0x5436 is hexadecimal.

This document is confidential and does not constitute a public document.

Possession should only be under NDA or other relevant confidentiality agreement.

This document has been prepared by Zephyr Technology Ltd and is not to be distributed, copied or reproduced without permission.

Date: Friday, 16 December 2011

Author:

Page

File Name: Event Messaging System.docx

SFe

Page 2 of 9

Contents

1. SYSTEM OVERVIEW	4
1.1. List of Events	5
1.1.1. Event Types:	5
1.1.2. System Events	6
1.1.3. Physiology Events	7

This document is confidential and does not constitute a public document.

Possession should only be under NDA or other relevant confidentiality agreement.

This document has been prepared by Zephyr Technology Ltd and is not to be distributed, copied or reproduced without permission.

Date: Friday, 16 December 2011

Author:

Page

File Name: Event Messaging System.docx

SFe

Page 3 of 9

1. System Overview

The event-based system is designed to transmit and/or log physiological or system events that may be of interest to users or to Zephyr.

Events of interest to a user will be logged or sent over the air as required. Some system events may never be seen by a user and only used for system monitoring and debugging by Zephyr.

The event-based system will allow a BioHarness to operate remotely, without an active Bluetooth connection, and only connect to a physiology data monitoring device when certain events happen, such as a change of ROG status indicating that there is a change in the subject's health. Such a system would result in longer battery life resulting from the reduced power consumption achieved by only enabling a Bluetooth connection when necessary.

The initial implementation does not contain the "Bluetooth on demand" system and relies on a Bluetooth connection to be constantly active to be able to send event messages. If there is no active Bluetooth connection, it will not attempt to initiate a connection. However all events are always logged regardless of a Bluetooth connection.

Future implementations will exploit the possibility of having a Bluetooth connection only when an event needs to be sent.

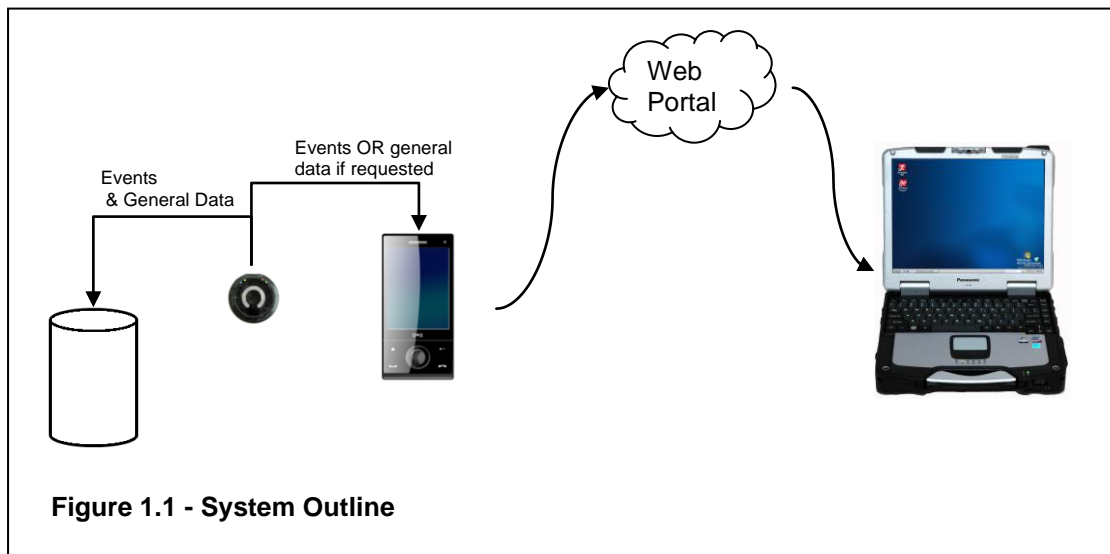


Figure 1.1 - System Outline

This document is confidential and does not constitute a public document.

Possession should only be under NDA or other relevant confidentiality agreement.

This document has been prepared by Zephyr Technology Ltd and is not to be distributed, copied or reproduced without permission.

Date: Friday, 16 December 2011

File Name: Event Messaging System.docx

Author:

SFe

Page

Page 4 of 9

1.1. List of Events

Events are defined into different categories such as System Events, Physiology events, Error events, etc. This allows for easier filtering by the application which analyses events.

Every event has a 16-bit event code which is divided into three fields 'Event Type', 'Event Source' and 'Event ID' as shown in the table below. This allows events to be easily filtered based on their type and source.

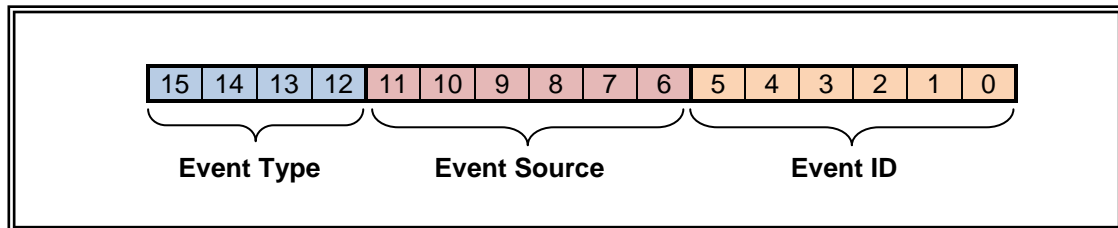


Figure 1.2: Event Code Description

1.1.1. Event Types:

- 0 = System Events
- 1 = Physiological Events
- 2 = Error Events (None currently implemented)
- 3 = Debug Events (none currently implemented)

1.1.2. System Events

System Events are events which describe changes to the system's state. The following event sources are defined for system events:

0 = Bluetooth
 1 = Button
 2 = Battery
 3 = Diagnostics

Event					Description	Event-Specific Data		
Name	Type	Source	ID	Code		Offset	Size	Description
Button Press	0	1	0	0x0040	When button gets pressed	-	-	No event-specific data
Emergency Button Press	0	1	1	0x0041	Emergency button press (3 presses within 2 seconds)	-	-	No event-specific data
Battery Level Low	0	2	0	0x0080	Battery level reaches a low level.	0	1	Current battery level (%)
Self Test Result	0	3	0	0x00C0	Result of diagnostic self-test	0	1	Breathing Sensor Test Result
						1	1	Accelerometer Test Result
						2	1	Logging Memory Test Result
						3	1	Bluetooth Module Test Result
						4	1	Apple Authentication Test Result
						5	1	Reserved for Test 6 (set to 0)
						6	1	Reserved for Test 7 (set to 0)
						7	1	Reserved for Test 8 (set to 0)
						8	1	Reserved for Test 9 (set to 0)
						9	1	Reserved for Test 10 (set to 0)

Self Test Result (8-bit signed value):

0 Test Not Performed
 1..127 Test Passed
 -1..-128 Test Failed



1.1.3. Physiology Events

Physiology Events are events which related to the physiology information reported by the device, normally change of state of certain physiology parameters. The following Event Sources are defined for physiology events:

- 0 = ROG Algorithm
- 1 = Worn Detection
- 2 = Heart Rate Calculation
- 3 = Fall Detection Algorithm
- 4 = Jump Test Algorithm
- 5 = Dash Test Algorithm

The table below contains a description of the currently defined physiology events.

Event					Description	Event-Specific Data		
Name	Type	Source	ID	Code		Offset	Size	Description
ROG change	1	0	0	0x1000	Change in ROG status	0	1	ROG status before change
						1	1	ROG status after change
Worn Status Change	1	1	0	0x1040	Status of worn confidence changes	0	1	Worn status before change (0-100%)
						1	1	Worn status after change (0-100%)
HR reliability change	1	2	0	0x1080	Change in heart rate reliability	0	1	Heart rate reliability before change (0-100%)
						1	1	Heart rate reliability after change (0-100%)
Fall Detected	1	3	0	0x10C0	A fall was detected	0	0	No Event-Specific Data
Jump Detected	1	4	0	0x1100	A jump was detected	0	2	Flight Time (milliseconds, unsigned 16-bit, LSB first)
						2	2	Peak Acceleration (10mg units, signed 16-bit, LSB first)
Dash Detected	1	5	0	0x1140	A dash was detected	0	2	Peak activity (10mg units, signed 16-bit, LSB first)

ROG Status	Bits	ROG Status
	0-2	000 Invalid
		001 Green
		010 Orange
		011 Red
		1xx Reserved

Note: Fall detection is implemented through use of a proprietary 3 rd party algorithm. Use of this by integrators will require payment of a license fee.

This document is confidential and does not constitute a public document.
Possession should only be under NDA or other relevant confidentiality agreement.

This document has been prepared by Zephyr Technology Ltd and is not to be distributed, copied or reproduced without permission.

Date: Friday, 16 December 2011
File Name: Event Messaging System.docx

Author: Page
SFe Page 8 of 9

