Continuous Wave Accelerometer Feasibility Study

September 2009

Chris Jones, Martin-Jones Technology Ltd

Document history	
Version 0.1	16 October 2009
Version 0.2	5 November 2009

1 Project Overview and System Architecture 2
2 Requirements 2
3 Sensor choice 2
4 CPU choice 3
4.1 CPU resources 3
4.2 Power management 5
4.3 Development 5
4.4 Physical size 5
4.5 Choices 5
5 User interface 5
6 Data storage 6
7 Communications 7
7.1 Bandwidth 7
7.2 Candidate interfaces 7
8 Power 8
8.1 Power budget 8
8.2 Battery choice 11
9 Mechanical assembly 12
10 Conclusion 12

1 Project Overview and System Architecture

The Continuous Wave Accelerometer (CWA) is intended to be used to collect data on the activity of a human subject as part of a large-scale medical research initiative. The activity data will be collected in the form of frequent three-axis accelerometer readings over a period of several days.

The CWA will be prepared for use by research staff and typically posted to the subject. The subject will, unsupervised, attach it to their body as instructed and leave it in place for the period of data collection.

Once the data collection is complete, the subject will return the CWA to research staff who will extract the collected data and possibly prepare the CWA for use by another subject.

2 Requirements

The requirements for the CWA are fully described in another document, ‘Needs and wishes for open source waveform accelerometer’. However, the key points are summarised here.

The CWA must:

- be able to collect three-axis accelerometer data of up to +/- 6g for at least 7 days at a sampling rate of 40Hz and resolution of 12 bits.
- be small enough to be posted and attached unobtrusively to the human body, typically on the wrist.
- be able to ‘sleep’ for several days before starting data collection
- be economical to manufacture in thousand-off quantities
- store data in such a way as to make it straightforward to extract by research staff
- store sufficient metadata to allow calibration factors to be checked and clock drift to be compensated for.

3 Sensor choice

There is a wide range of MEMS (Microelectromechanical Systems) accelerometer devices available, offering good performance in extremely small physical sizes. These sensors typically contain three axes of accelerometer, signal conditioning and analogue to digital conversion circuitry. Their packages are normally 5mm x 5mm x 1mm or smaller, and their power requirements small, just a few milliwatts, making them ideal for this sort of application.

A brief survey of available parts which may meet the CWA’s requirements found the following candidate devices.

Manufacturer	Part number	Size (mm)	Range	Interface	Resolution at +/-6g	Active power consumption V_{DD}=2.5V
Analog Devices	ADXL346	3x3x0.95	+/- 2/4/8/16g	SPI, I2C	11.6 bits	87.5µW
Freescale	MMA7455L	3x5x1	+/- 8g	SPI, I2C	9.6 bits	1000µW
ST	LIS3LV02DQ	7x7x1.8	+/- 2/6g	SPI, I2C	12 bits	1500µW
ST	LIS302DLH	3x5x0.9	+/- 2/4/8g	SPI, I2C	15.6 bits	625uW
Kyonix	KXSD9-1026	3x3x0.9	+/- 2/4/6/8g	SPI, I2C	12 bits	550µW

The resolution figures have been calculated as an equivalent number of bits if the part does not include a +/- 6g resolution mode.

It is clear that the Analog Devices and Freescale parts cannot satisfy the 12 bit resolution requirement, leaving only the ST and Kyonix parts as candidates.

The ST LIS302DLH and Kyonix KXSD9-1026 are directly competitive in size and power requirements, but the ST part offers improved resolution which makes it the best candidate as a choice in this application long-term. The equivalent 15.6 bit resolution at +/-6g easily exceeds the requirement and gives an opportunity to improve the overall resolution of the CWA in the future.

4 CPU choice

Some form of controller is required to manage the operation of the CWA. The controller must be physically small and consume little power. There is no requirement for particularly high-speed data processing, so an off-the-shelf integrated microcontroller is an appropriate choice.

4.1 CPU resources

4.1.1 I/O interfaces

The CPU needs to interface to all the subsystems of the CWA, including the accelerometer, Flash memory, power management, communications and LEDs. The following table summarises the I/O requirements.

	In	Out	I/O	Special
Accelerometer (SPI bus)				
MOSI		1		
MISO	1			
SCLK		1		
nCE		1		
INT	1			
NAND Flash (8 bit parallel)				
D7-D0			8	
CLE		1		
ALE		1		
NCE		1		
NRE		1		
New		1		
NWP		1		
RnB	1			
Power management				
Charge		1		
Charge complete	1			
LEDs				
R, G, B		3		
B charge pump		1		
USB				
D+/D-				2
VBUS	1			
Pullup		1		
Total	**5**	**15**	**8**	**2**

4.1.2 Memory

The CPU must contain sufficient RAM for buffering data read from the accelerometer. Data is written to the Flash memory in 2048-byte pages, so two 2048-byte buffers should be sufficient to cope with repeated write attempts on bad blocks.

Other RAM requirements should be relatively small, since the data logging code is unlikely to contain very much complex state.

The non-volatile program ROM (usually Flash ROM) in the CPU must be large enough to accommodate all the software required to run the CWA. It is usual for each range of microcontrollers to include a number of pin-compatible models with varying amounts of memory, so scaling the memory size up or down should be relatively straightforward.

4.1.3 Peripherals

An SPI bus controller is required to interface to the accelerometer.

A selection of built-in timers is important, since most of the actions taken by the CPU will be on a timed schedule. This approach should also allow the CPU to spend most of its time asleep, only being woken by interrupts from timers. It is therefore of particular importance that at least one timer is available which can ‘wake up’ the CPU from an extremely low-power state, whilst itself drawing only a small amount of power.

In order to meet the communications requirement, a USB device controller must also be built in to the CPU.

4.2 Power management

The CPU must be low-power when running, but even more importantly be able to drop into a very-low-power sleep state when it is not immediately required to perform any action. In practice it will should spend most of its time in this state, in between reading samples from the accelerometer and storing them in the Flash memory.

4.3 Development

Since the CWA is intended to be an open-source project, an open-source development environment should be available for the CPU. Its architecture should also be reasonably familiar to developers in the academic community.

4.4 Physical size

The CPU should be as small as possible. An increasing number of parts are now available in leadless QFN packages which saves a noticeable amount of space when compared with the previous generation of leaded QFP packages.

4.5 Choices

A selection of microcontrollers meeting these requirements is presented below.

	Part	I/O pins	RAM (KB)	ROM (KB)	Package	Size (mm)	Architecture
ST	STM32F103T6U	26	10	32	QFN36	6 x 6 x 1.0	ARM Cortex-M3
ST	STM32F103R6H	51	10	32	BGA64	5 x 5 x 1.2	ARM Cortex-M3
NXP	LPC1343FHN33	28	8	32	QFN33	7 x 7 x 0.85	ARM Cortex-M3
Atmel	AT32UC3B164	28	16	64	QFN48	7 x 7 x 1.0	AVR32
Atmel	AT91SAM7S64	32	16	64	QFN64	9 x 9 x 0.9	ARM 7
TI	MSP430F5529	63	10	128	LQFP80	12 x 12 x 1.4	MSP430

The ARM, AVR32 and MSP430 architectures are well supported by open-source tools (the gcc toolchain), with the ARM being most familiar to the existing project team.

All these microcontrollers appear to meet the requirements for I/O, storage and peripherals. The decision of which one to use will most likely depend on physical size and power requirements to be evaluated later in this document.

5 User interface

The CWA is deliberately designed with almost no user interface of its own. Once configured by research staff, it will run autonomously, starting and stopping data collection at predefined times.

In order to provide a simple mechanism to indicate correct operation and error conditions to both research staff and subjects, a simple tricolour LED will be included which can shine through the CWA's housing.

An example of a tricolour LED available in a very small (1mm x 2.5mm x 1mm deep) package at low cost is the Avago HSMF-C113. It costs less than £0.39 in volumes of 500 or more.

It is worth noting for design purposes that the forward voltage of Gallium Nitride-based blue LEDs is typically 3.5-3.9V, which is higher than the terminal voltage of a partly-discharged Lithium Ion cell, which is representative of a typical power source for this application. A simple charge pump arrangement will therefore be required to run the blue LED at full brightness.

6 Data storage

It is possible to estimate the volume of data to be stored using the requirements for the sampling rate, resolution and data collection period. Assuming:

Number of axes = 3

Sampling rate = 40 samples/second per axis

Sample resolution = 12 bits

Data collection period = 7 days = 604800 seconds

Total data volume = 3 x 40 x 12 x 604800 = 871×10^6 bits

If the data is efficiently packed into 8-bit bytes, this is 109×10^6 bytes.

There will be a certain amount of overhead for timestamps, calibration data, subject identification, error checking and so on. It is reasonable to assume 10% overhead, which leads to a total of 120×10^6 bytes.

This data must be stored in a non-volatile memory, so that it is retained even if the battery powering the CWA becomes exhausted during or after data collection. There is only one realistic choice for such storage in the volume required in this application: NAND Flash memory. This type of memory is commonly used as storage in solid-state laptop hard drives, music players, mobile phones and other portable devices, and is thus cheap and readily available. It consumes no power when not being accessed, making it ideal for battery-powered use.

A range of devices are presented here for comparison.

Maker	**Part number**	**Size Mbyte**	**Interface**	**Package**	**Size (mm)**	**Price (100 off)**
Hynix	H27UF081G1M-TPCB	128	8/16 bit	TSOP48	12 x 20 x 1.2	£1.91
Hynix	H27UF082G2B-TPCB	256	8/16 bit	TSOP48	12 x 20 x 1.2	£2.37
Hynix	H27UF084G2B-TPCB	512	8/16 bit	TSOP48	12 x 20 x 1.2	£3.40
Hynix	H27UF084G2M-TPCB	512	8 bit	TSOP48	12 x 20 x 1.2	<£3.40 (est)
Hynix	H27UF084G2M-UPCB	512	8 bit	ULGA52	12 x 17 x 0.6	<£3.40 (est)
Samsung	K9F2G08U0B	256	8 bit	TSOP48	12 x 20 x 1.2	
Samsung	K9F4G08U0B	512	8 bit	TSOP48	12 x 20 x 1.2	
Micron	MT29F4G08AACWC	512	8 bit	TSOP48	12 x 20 x 1.2	
Micron	MT29F4G08AACHC	512	8 bit	VFBGA63	9.5 x 12 x 0.6	
Micron	MT29F1G01ZACHC	128	SPI	VFBGA63	9.5 x 12 x 0.6	

This comparison demonstrates that there is a wide range of devices available with suitable storage capacity. Most of them are the industry-standard TSOP48 package which, though relatively large at 12mm x 20mm, still fits within a space appropriate for this application. The ULGA52 package is only marginally smaller, but uses a different soldering method. The choice between the two packages would seem to be mostly down to manufacturing convenience.

A small number of NAND Flash devices are available in a smaller VFBGA63 package, which may be convenient if space becomes the most important issue. In particular, Micron offers a range of 128Mbyte devices with a serial (SPI) interface which could both simplify and speed the interface from the microcontroller to the Flash memory. However, these devices are only available from one supplier and presently only in the 128Mbyte capacity, so could represent a limit on future expansion of the CWA's capabilities.

7 Communications

As the CWA is a data-collecting device, there is a clear need to be able to get configuration data into it and collected data out of it. There is, however, no requirement for communication during the data collection period. This means that any communication mechanism used does not have to work when the CWA is environmentally sealed.

Whatever communication mechanism is chosen it should be directly electrically compatible with a standard desktop PC. A mechanical adapter may be required, since a standard connector might not fit on the CWA or meet its environmental sealing requirements. It is acceptable to require special software on the PC to access the CWA.

7.1 Bandwidth

The CWA will be collecting a significant amount of data, of the order of 100 megabytes (see 'Data storage' section above). It is important that downloading this volume of data from the device should not take an unreasonable amount of time, because the downloading process occupies both equipment and personnel. Downloading the data must remain manageable even with very large numbers (tens of thousands) of subjects.

7.2 Candidate interfaces

There are several interfaces commonly encountered on desktop and laptop PCs.

7.2.1 Serial interface

The serial interface is very simple to implement, generally being supported directly by most microcontrollers though requiring external line drivers and receivers. It was once the most common means of interfacing with PCs. However, it is rarely present on new machines, having been supplanted by more modern interfaces. It is also limited in speed, normally to 115200bits/second (approximately 11kbytes/second).

7.2.2 Firewire or IEEE1394

This interface supports very high speeds (400 or 800Mbits/second) and is commonly found on modern PCs. It is aimed at high-quality video streaming and high performance mass storage applications. Hardware to support it is not found on microcontrollers, and so implementing it in this application would be expensive and space-consuming.

7.2.3 USB

The USB interface has effectively replaced the serial port as the most common means of connecting external hardware to PCs. It has been ubiquitous for at least a decade and is commonly supported on high-performance microcontrollers without requiring any external hardware. Although it exists in a ‘high speed’ (480Mbits/second) variant, microcontrollers usually only support ‘full speed’ (12Mbits/second) operation.

7.2.4 Wireless interfaces

There are a number of wireless interface standards in common use for connecting peripherals to PCs. These include Wi-Fi (IEEE802.11), Bluetooth and Zigbee. In general, these interfaces are not supported directly by commonly-available microcontrollers, requiring additional hardware. In addition, they are generally too power-hungry to be enabled all the time in a tiny device like the CWA.

Wireless communication could also introduce ambiguity in a research environment in which multiple CWAs are being configured and having their data downloaded simultaneously. Some form of user interface would be required on the device to ensure that a neighbouring CWA was not connected by mistake.

The benefits of wireless connection do not, in this application, outweigh the additional complexity it introduces, since there is no requirement for ‘live’ data transfer.

7.2.5 Choice

For this application, the USB interface is a clear choice. It is readily available, requires no extra hardware in the CWA and is directly compatible with PCs. Its data transfer rate is theoretically fast enough to download the expected 120Mbyte of data in a few minutes.

8 Power

The CWA will require a power source of some sort. This power source must provide sufficient energy to last through a complete sleep and data collection cycle.

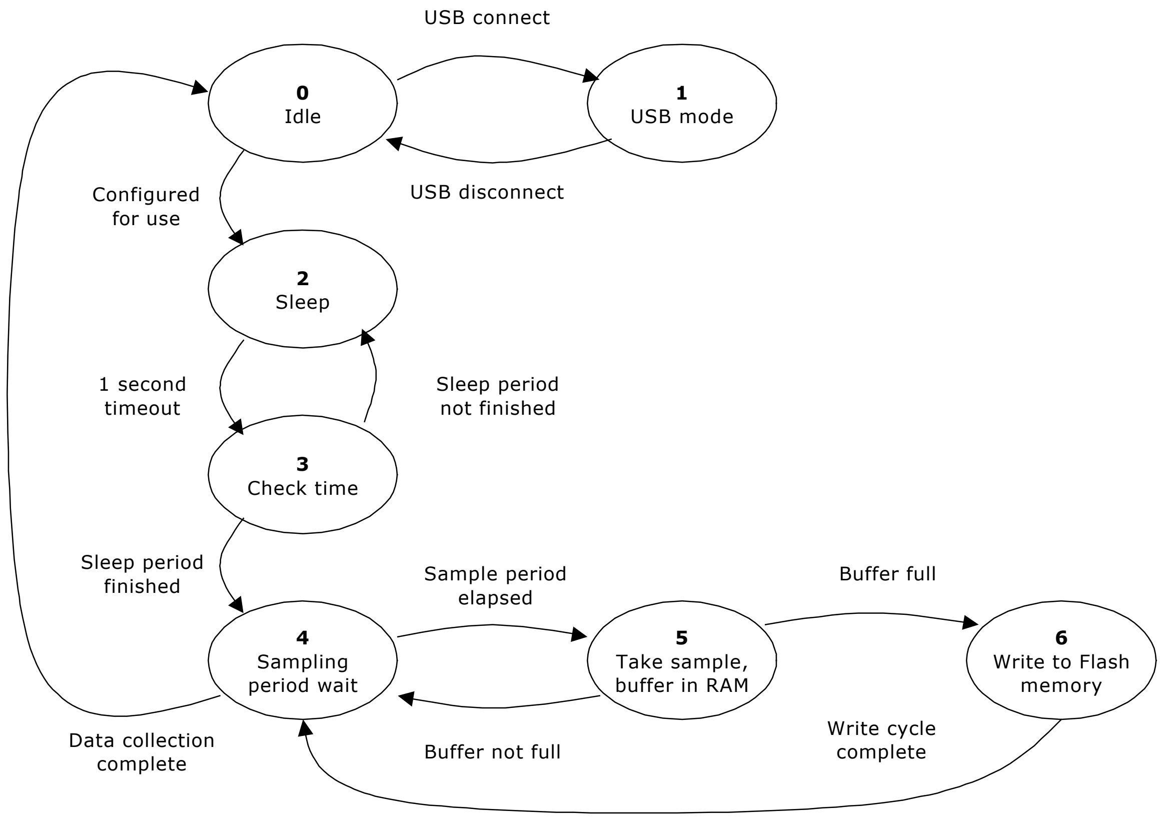
Techniques are available to harvest small amounts of power from the environment, from radio waves, movement or light. However, in the physical space available in this application, these are generally only capable of harvesting less than 100uW of power which is insufficient for active data collection as will be seen from the power budget calculations in this section.

A battery is therefore going to be required. Given the requirement for environmental sealing of the CWA, a rechargeable battery is likely to be a convenient choice, since no access to it need be provided. It is important, however, to consider the deterioration in performance of a rechargeable battery after a large number of charge/discharge cycles.

8.1 Power budget

In order to calculate a power budget for the CWA, it is useful to consider the complete cycle of its operation from configuration to data downloading. During this time the various major components of the system will consume varying amounts of power as the system enters the different states involved in its operation.

A simplified state diagram for the system is shown below.



	State	Description
0	Idle	The CWA is effectively switched off. Only the bare minimum of power supply is running. The accelerometer and Flash memory are disabled, and the CPU is in the minimum power state consistent with being woken up by a USB connection being made.
1	USB mode	The CPU and its USB peripheral are active while the USB connection is made. The accelerometer is disabled but the Flash memory is likely to be active. In some respects the power consumption in this mode is unimportant, since power is available from the USB port.
2	Sleep	This state represents most of the ‘wait time’ before data collection starts. The CPU, accelerometer and Flash memory are all inactive. The only active component is a timer.
3	Check time	During the ‘wait time’ the CPU must wake up periodically and check whether data collection is due to begin. Only the CPU is active during these brief periods.
4	Sampling period wait	Once data collection has started, the accelerometer is switched on and the timer is used to time the sample period. The accelerometer must be enabled continuously during data collection to avoid aliasing effects. In between samples, the CPU and Flash memory are inactive.
5	Take sample, buffer in RAM	At the time of each sample, the CPU becomes active for a brief period to read three axes of data from the accelerometer. The readings are buffered temporarily in the CPU’s RAM.
6	Write to Flash memory	After taking a number of samples, the CPU’s RAM buffer will contain sufficient data to fill one block of Flash memory. At this point the Flash memory will be activated for long enough to store the buffered data.

Given knowledge of the requirements for the pre-collection sleep time, data collection period length, sample rate and resolution, and the properties of the microcontroller and Flash memory, it is possible to estimate the duration and frequency of each of the states.

	State	Duration	Frequency	Notes
0	Idle	Undefined	Undefined	Unimportant for this calculation
1	USB mode	Undefined	Undefined	CWA is powered from USB, so does not contribute to battery drain
2	Sleep	1s	1 per second during sleep time	
3	Check time	100us	1 per second during sleep time	
4	Sampling period wait	1/40s	40 per second during data collection	Sampling rate of 40Hz
5	Take sample, buffer in RAM	100us	40 per second during data collection	Assuming CPU can wake up in a small number of microseconds
6	Write to Flash memory	1ms	Every 11.3 seconds during data collection	Sample size of 12 bits, Flash block size 2048 bytes, no write errors

Using the manufacturers' data sheets for each of the components (accelerometer, microcontroller and Flash memory) it is possible to estimate the power consumption of the complete system in each of the states.

For the purposes of calculation, the accelerometer is assumed to be the ST LIS302DLH. The various manufacturers of Flash memory quote almost identical power consumption figures, so that part is assumed to be generic. The primary variable is therefore the microcontroller. The following table shows the estimated power consumption of the key components in each state. The power consumption is expressed in microamps, assuming a power supply voltage of 2.5V.

From this point onwards, the states 'Idle' and 'USB mode' are ignored because they are not used during the battery-powered data collection cycle.

	State	Sensor	Flash	CPU	CPU	CPU	CPU	CPU
		LIS302DLH		STM32F103	LPC1343	AT32UC3B	AT91SAM7S	MSP430
2	Sleep	1	10	20	500	8	560	2.5
3	Check time	1	10	4000	4000	3700	6000	1600
4	Sampling period wait	250	10	20	500	8	560	2.5
5	Take sample, buffer in RAM	250	10	4000	4000	3700	6000	1600
6	Write to Flash memory	250	15000	4000	4000	3700	6000	1600

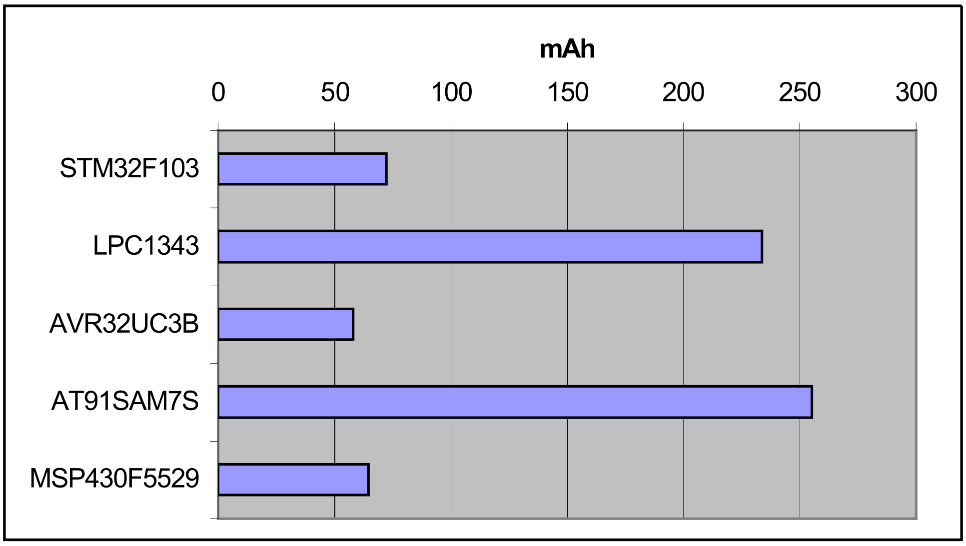
In addition to the variable power consumption of each of the components shown above, it is expected that there would be a quiescent current drain of approximately 50uA due to a voltage regulator (the Zetex ZXCL250 series is recommended for low-quiescent-current applications and small physical size) and other power management systems.

Integrating over the complete sleep and data collection cycle, it is possible to calculate the total energy consumption of the CWA. It is expressed here in milliamp-hours, since this is the unit in which the capacity of batteries is most commonly expressed.

The following table assumes the use of a LIS302DLH accelerometer, generic Flash memory, and 50uA quiescent current drain. It shows the total energy consumption of a CWA based around each of the microcontrollers considered, over a one week sleep period and one week of data collection.

STM32F103	72.3mAh
LPC1343	233.6mAh
AVR32UC3B	58.0mAh
AT91SAM7S	255.1mAh
MSP430F5529	64.7mAh

The results are shown graphically below.



The LPC1343 and AT91SAM7S show dramatically greater power consumption because they do not have built-in 'real time clock' timers. The timing functions of the CWA therefore depend on one of the microcontroller's general purpose timers and its associated high-frequency crystal oscillator. These components are relatively power-hungry. A 'real time clock' timer, on the other hand, runs from a low-frequency (typically 32.768kHz) crystal and consumes very little power. The microcontrollers with this feature (STM32F103, AVR32UC3B and MSP430F5520) show very much lower overall energy consumption as a result.

The STM32F103, AVR32UC3B and MSP430F5520 show very similar overall energy consumption figures and are comparable in most other ways. The STM32F103, however, is available in a usefully smaller physical package than the others, so it is the leading choice of microcontroller for this application.

8.2 Battery choice

Rechargeable Lithium Polymer batteries are available in conveniently small sizes. Some examples are shown here.

Make	Type	Voltage	Capacity	Dimensions
Varta	LPP402025CE	3.7V	150mAh	25 x 20 x 4mm
Yuntong	YT452025HH	3.7V	130mAh	25 x 20 x 4.5
Yuntong	YT531124	3.7V	85mAh	25 x 11 x 5.3

Charging of Lithium Polymer batteries from a USB source can conveniently be managed by a single-chip charger such as the Maxim MAX1551.

9 Mechanical assembly

The CWA will be worn at all times during the data collection period, so it will be subjected to all sorts of environmental conditions. Baths, showers and swimming, for example, will require the CWA to be waterproofed.

Achieving this whilst allowing sufficient electrical access to charge the battery, configure the device and download data is conveniently achieved by performing all these functions through the USB port.

If the main circuit board of the CWA is permanently potted in resin in order to seal it, the USB wiring can be conveniently brought out of the sealed area of the CWA on a thin, flexible PCB. The USB connection to this PCB could be made using either a custom 'bed-of-nails' onto gold-plated contact areas, which has the advantage of very low cost and zero physical size in the CWA, or a conventional micro USB connector could be fitted, which has the advantage of not requiring any special hardware to interface with.

10 Conclusion

Implementing a Continuous Wave Accelerometer which meets the requirements in section 2 is feasible. The following key components are recommended for the design.

Sensor	STMicroelectronics LIS302DLH
Microcontroller	STMicroelectronics STM32F103R6H
Flash memory	Generic TSOP48-packaged NAND Flash
Interface	USB
Indicator	Avago HSMF-C113
Battery	Varta LPP402025CE
Battery management	Maxim MAX1551
Voltage regulator	Zetex ZXCL250 series