Sprint 1

Summary

User	Change:	**acquisition without a wire.**
	Features:	device works on battery,
		it is possible to identify a period of the recorded event.
Dev	Main:	**synchronized acquisition on battery.**
	Implied:	MCU and motion sensor selection,
		internal timer configuration in order to trigger the acquisition.

Table B.1. Sprint 1 - summary.

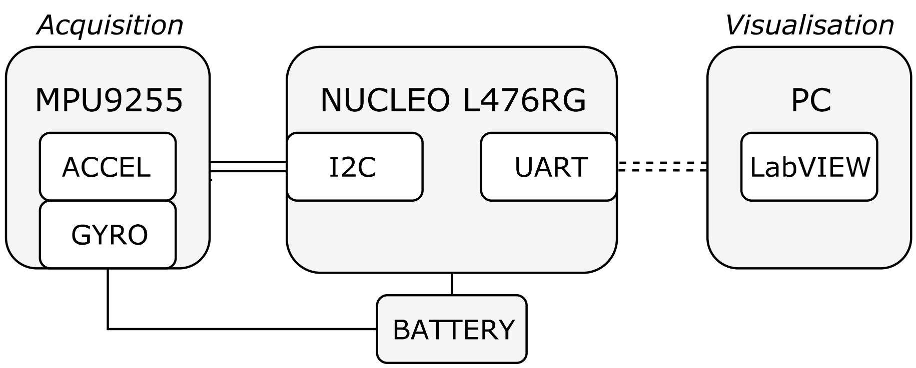

Figure B.2. Sprint 1 - system overview.

During this week the author was thinking about hardware architecture and finally decided, that the device should consist of development boards and dedicated shield. 4xAA battery holder for supply was used and everything was packed in a gray plastic box. Such a decision was made because of limited budget, PCBs' design skills and long production cycle of them.

The Detector was configured for running from the external power supply and IMU data acquisition at fixed frequency. Single-packet data transfer was replaced by a multi-packet one.

Principle of operation

The device waits for a user button press, then it starts data packet[4] acquisition with frequency 1 kHz, giving an opportunity to register eight[5] seconds of measurement. If it is ready, it is possible observe led state changing. This means that device can transmit the data to the User Application on the PC. After successful wiring all the data can be sent (starting with user button press) and observed on graphs.

[4] *Packet* - sequence of samples which represents a full single measurement.
[5] Packet length is limited by MCU resources.

Sprint 2

Summary

User	Change:	**event-oriented** acquisition.
	Features:	record always last 8 seconds of acquisition.
		it is possible to identify a period of the recorded event.
Dev	Main:	**circularly buffered** acquisition.
	Implied:	circular buffer implementation,
		serial transceiver placement.

Table B.2. Sprint 2 - summary.

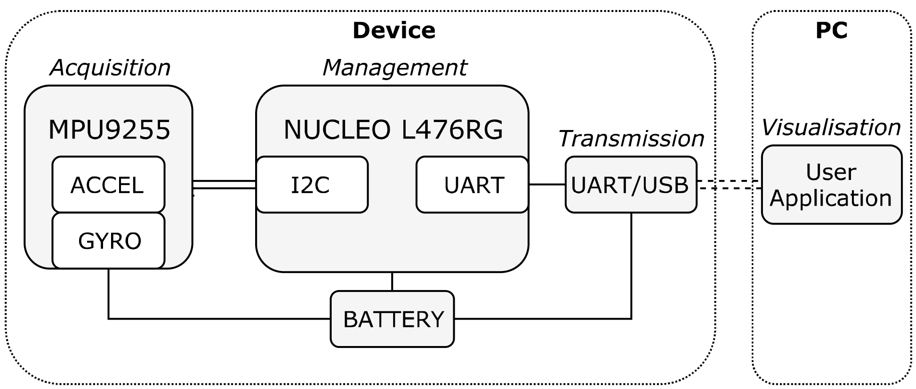

Figure B.3. Sprint 2 - system overview.

Type	Size	Max
Flash	34.4 kB	1,024.0 kB
RAM	95.9 kB	128.0 kB

Figure B.4. Memory usage, Mbed.

Basic tests showed that about 10 s of measurement were necessary to register a typical fall accident. Having a frequency equal to 1 kHz and assuming 8000 samples per packet, it gave about 96 kB of data[6] required in RAM section of the MCU.

In the last version, acquisition always lasted 8 s - it pressured a user into haste. As a result, the author decided to implement circular data buffering. It allowed the user to stop the ongoing measurement in the appropriate time, preserving valuable data.

Another important fact was noticed: the device should support measurements shortest that 8 s (as short as user decides). The requirement was added to the backlog.

Principle of operation

The device operates as described in Section B. The only difference is that the user button stops recording. After pressing it, there are last 8 seconds of measurement registered and ready for transfer.

[6]Double (acceleration, rotation) 3-axis 16b measurement gives 12 B per sample.

Sprint 3

Summary

Dev	Main:	**file system test**.
	Implied:	SD card file save/read possibilities checked.

Table B.3. Sprint 3 - summary.

A lot of time the author spent on debugging and testing the SDFileSystem library in order to provide a quality of measurement packets. The main target was to prepare the system for the incoming feature (allowing saving IMU packet to a file on SD card). Even though user has not received a new feature, an important project step was taken.

A problem encountered while running examples available on Mbed OS. As it turned out, latest versions of libraries had some bugs resulted in SD card reading errors. Fortunately, the library used in the project worked well with written sample code, giving an opportunity to apply the library solution into the source code.

Principle of operation

The device connects to a SD card using SPI bus, creates a folder using file system and finally generates a file with specific content. Next, assuming successful reading, the read content is being compared with the saved one. Equality means that the test is passed.

Sprint 4

Summary

User	Change:	**half-automatic measurement system**.
	Features:	user application is responsible for data transfer triggering,
		enhanced user-error-resistant interface.
Dev	Main:	circularly buffered acquisition with **command decoding system**.
	Implied:	automatic device detection,
		frame decoding,
		establishing the serial interface (1 Mb).

Table B.4. Sprint 4 - summary.

The system overview has not changed since Section B.

Data transfer between the device and the user application has been constricted so far. User had to configure a serial port manually, sending data was being triggered by a button on the detector. What is more, the fact, that every measurement was being displayed on a separate graph, caused a great deal of user confusion. Features implemented this week resolved major problems which occurred.

It is worth mentioning, that there was a need to change a LabVIEW version in order to meet set requirements. The lack of property node and FTDI libraries[7] overcame a willingness to explore the new generation of acquisition software.

Principle of operation

The device starts data packet acquisition **automatically** with frequency 1 kHz, giving an opportunity to register eight seconds of measurement. **Every 8 seconds** (because of buffer overwriting) **user led switches its state**. **User button press stops recording and switches the operating mode of the device in order to decode user application commands**. After successful wiring we can send all data and observe results on graphs, starting with a **reaction on the user application prompt**.

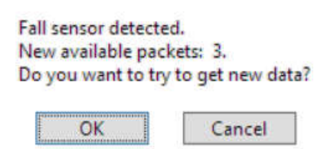

Figure B.5. User prompt which appears when the new data is available to transfer.

[7]LabVIEW NXG 1.0 does not provide that kind of features.

Sprint 5

Summary

User	Change:	automation **cont.**
	Features:	the device can save the measurement data to a file,
		user has easy access to the device UI (Button, Buzzer).
Dev	Main:	buffered acquisition with **measurement memory** and command decoding system.
	Implied:	file system (from Section B) implementation,
		circular buffer editing.

Table B.5. Sprint 5 - summary.

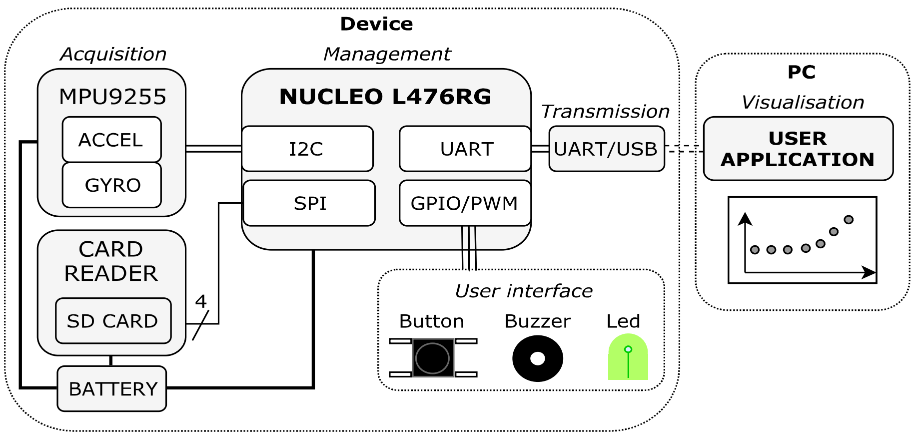

Figure B.6. Sprint 5 - system overview.

Three issues occurred. Firstly, the author noticed strange behavior during data acquisition: displayed waveforms seemed to be noncontinuous. Secondly, user application required reset if the device had not been connected to the PC earlier. Apart from that, system has not been prepared for transferring saved packets yet.

Principle of operation

The device starts data packet acquisition automatically with frequency 1 kHz, giving an opportunity to register eight seconds of measurement, responded by user led state switching. **Long** user button press stops recording and enables the command decoding. After successful wiring all the data can be sent and observed on graphs. **Short press results in saving a packet into a raw file on SD card**.

Sprint 6

Summary

Dev	Main:	buffered, **externally-triggered** writable acquisition with a command decoder.
	Implied:	triggering from the sensor interrupt pin, data continuity test.

Table B.6. Sprint 6 - summary.

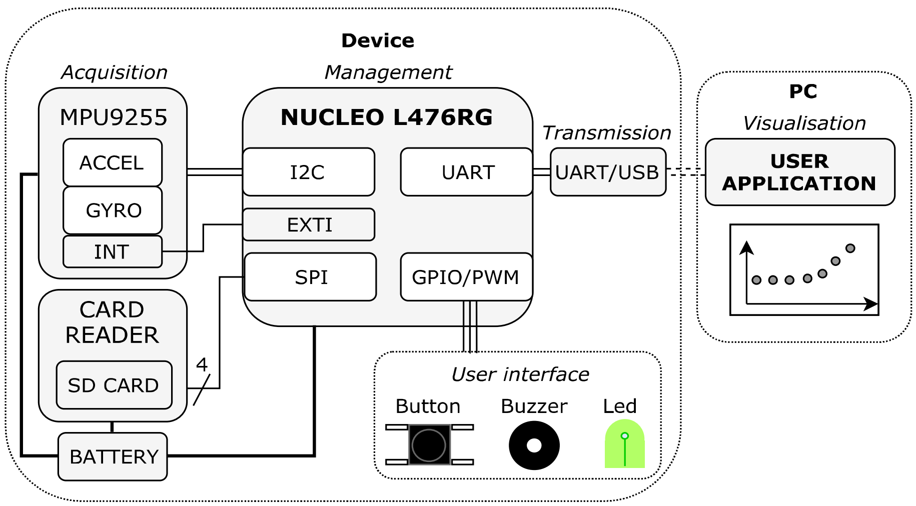

Figure B.7. System overview - final version.

As it turned out, measurement was continuous (Section 6.2). Large peaks were caused by the impact during rotating device on the table.

Principle of operation

The device operates as described in Section B.

Sprint 7

Summary

User	Change:	**packet browser**.
	Features:	user application allows to browse available packets, transfer all packets saved on the device.
Dev	Main:	circularly buffered acquisition with **data viewing** system.
	Implied:	list of packets, data frame restructuring, periodic device detection.

Table B.7. Sprint 7 - summary.

This sprint started a cycle of operations towards a stable product release which should provide the distributed measurement system described in Section 1.3.1. The author concentrated on adding features and bugs resolving.

Detector	**User Application**
- get back to acquisition after data transfer	- save/load measurement packets
- delete stored measurements after data transfer	- separate normal and service mode
- unify programming language	- process UI events and other operations simultaneously
- test various sensor configurations	- do not receive empty packets
- estimate a time device works on battery	- delete single packet from the list
	- append (if necessary) downloaded packets to the list

Table B.8. Sprint 7 - Backlog.

Principle of operation

The device operates as described in Section B. The only difference is that sampling rate equals to 100 Hz and 5 s is the maximal length of measurement packet (for that frequency).

Sprint 8

Summary

User	Change:	**multi packet storage**.
	Features:	save/load browsed packets,
		delete selected packet.
Dev	Main:	circularly buffered acquisition with **data management** and viewing system.
	Implied:	save/load measurement packets,
		delete single packet from the list,
		delete stored measurements after data transfer,
		do not receive empty packets,
		append (if necessary) downloaded packets to the list,
		process UI events and other operations simultaneously,
		separate normal and service mode.

Table B.9. Sprint 8 - summary.

Detector	**User Application**
- **implement fall detection algorithm**	- **only packet list and graphs should be on the front panel**
- unify programming language	- **add deployment configuration**
- test various sensor configurations	- **add menu**
- estimate a time device works on battery	- **enhance UI by system controls and indicators**
- **debug mode with logging to a file**	- **add sensor configuration wizard**
	- **display proper units on graphs**
	- **store packets in a binary file**
	- **export single packet to CSV/JSON**

Table B.10. Sprint 8 - Backlog.

Principle of operation

The device operates as described in Section B.

Sprint 9

Summary

User	Change:	**modern interface**.
	Features:	clear front panel,
		append data option within a dialog box,
		graph palette is enabled.
Dev	Main:	**distributed measurement system**.
	Implied:	unify programming language,
		only packet list and graphs should be on the front panel,
		add menu,
		enhance UI by system controls and indicators,
		display proper units on graphs.

Table B.11. Sprint 9 - summary.

Detector	**User Application**
- implement fall detection algorithm	- add sensor configuration wizard
- test various sensor configurations	- **list packet parameters (id, comment, duration)**
- estimate a time device works on battery	- store packets in a binary file
- debug mode with logging to a file	- export single packet to CSV/JSON/**PNG**
	- add deployment configuration

Table B.12. Sprint 9 - Backlog.

Principle of operation

The device operates as described in Section B.

Sprint 10

Summary

User	Change:	**released stable data acquisition system**.
	Features:	export single packet to CSV/JSON/PNG,
		configure sensor using wizard.
Dev	Main:	**fall sensor - data acquisition mode ready**.
	Implied:	add deployment configuration,
		list packet parameters (id, comment, duration),
		store packets in a binary file.

Table B.13. Sprint 10 - summary.

Detector	User Application
- implement fall detection algorithm	- **refactoring and documentation**
- test various sensor configurations	
- estimate a time device works on battery	
- debug mode with logging to a file	
- **refactoring and documentation**	

Table B.14. Sprint 10 - Backlog.

Principle of operation

The device operates as described in Section B.

Sprint 11

Summary

User	Change:	**released fall detection mode**.
	Features:	implement fall detection algorithm,
		select between fall detection and data acquisition,
		alarm when fall occurs.
Dev	Main:	fall sensor - **edited main module**.
	Implied:	test various sensor configurations,
		estimate a time device works on battery.

Table B.15. Sprint 11 - summary.

Detector	**User Application**
- refactoring and documentation	- refactoring and documentation
- debug mode with logging to a file	

Table B.16. Sprint 11 - Backlog.

Principle of operation

Two working modes are available.

The fall detection mode is default and runs automatically after device startup. Data is being acquired with frequency dependent on the configuration file (or default 100 Hz if not exist). The algorithm process acquired data and if it detects potential fall, the device waits a couple of seconds giving a user an opportunity to cancel incoming alarm (in case of invalid detection). After that, acquisition data is saved, the alarm is called and lasts as long as the device is turned on.

The acquisition mode could be turned on by pressing and holding a user button while device startup. It works as described in Section B.

Sprint 12

Summary

User	Change:	**minor fixes**.
	Features:	edit comment on double-click,
		recording instead of *packet*,
		remove unnecessary buttons,
		write User Manual.
Dev	Main:	**refactored application**.
	Implied:	-

Table B.17. Sprint 12 - summary.

Detector	**User Application**
- debug mode with logging to a file	-

Table B.18. Final Backlog.

Principle of operation

The device operates as described in Section B.