5

Sensor Sub-System

Carlos Pérez and Daniel Rodriguez-Martin

Universitat Politècnica de Catalunya – UPC, CETpD – Technical Research Centre for Dependency Care and Autonomous Living,
Vilanova i la Geltrú (Barcelona), Spain

5.1 Introduction

The sensor is an important part of REMPARK since it is in charge of capturing relevant inertial data from the patient's movement pattern. The location of the sensor is the left side of the waist and its operation is autonomous and done in real-time. The sensor, described along the present chapter, has an embedded set of algorithms, already introduced in precedent chapters, and able to determine indicators for specific movement disorders related to PD, mainly Dyskinesia, Freezing of Gait (FOG) and Bradykinesia.

Following the requirements derived for this sub-system (discussed in Chapter 3), the main design principles, among many others, are:

- the achievement of a device with a reasonable operating autonomy (adjusted power consumption), but keeping a small physical size;
- a device capable of being worn comfortably, permitting the regular activities along the day; and
- the local execution, in real-time, of the embedded algorithmic set for symptoms' detection.

The system includes the classical elements of an Inertial Measurement Unit (IMU), together with a dedicated part for battery control and energy consumption optimisation. The battery level and the status of the main application process are indicated to the user using a LED.

A microcontroller (μC) is in the nucleus of the sensor, being responsible for the real-time and local execution of the developed algorithms on the acquired data. This microcontroller is also responsible for handling and controlling the rest of the components.

5.2 Sensor's Data Processing Flow

Before entering into the details of the sensor sub-system, it is necessary to establish the Processing Data flow implemented within the sensor device for executing the symptoms' detection algorithms in real-time.

Developed algorithms operate on acquired samples on inertial sensors, and Figure 5.1 shows a data flow compatible with the previously presented processing specifications. Since many of the implemented algorithms are based on a windowed analysis of the related signals, the figure distinguishes three main parts:

- A first one is a set of calculations performed after acquiring a sample, which mainly consist in the filtering of the signals to condition them, followed by the execution of the fall detection algorithm.
- A second one comprising the calculations performed at the end of a window time, which comprises the main computations for the monitoring of the motor symptoms (detection of Bradykinesia, Dyskinesia and FoG).
- The third part comprises the computations done once per minute, in which the final output of the algorithms is obtained and shared through the mobile phone.

5.3 Hardware Requirements

According to the introductory section, the internal organization of the sensor sub-system used in REMPARK appears in Figure 5.2. This section will introduce the requirements of some critical hardware parts, together with a timing analysis before the specification of the concrete components chosen for the physical implementation.

It must be noted that during the REMPARK database collection phase, the indicated three triaxial sensors (accelerometer, gyroscope and magnetometer) were used to collect a complete set of signals from as many physical magnitudes related to movement as possible. However, in the algorithm development phase, acceleration signals were identified to be enough to monitor motor

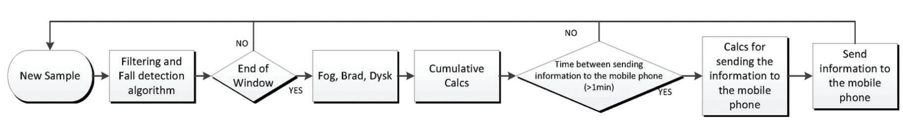

Figure 5.1 Data flow for the algorithms implementation.

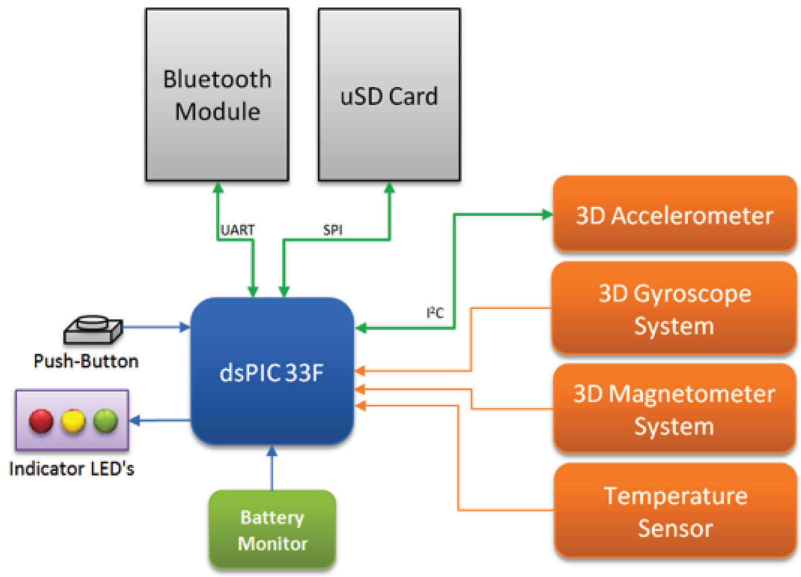

Figure 5.2 Internal organisation of the sensor unit.

symptoms. In this way, it must be indicated that the final sub-system only includes the 3D accelerometer as a sensor.

The most relevant requirements are:

- The concrete component chosen as the correct accelerometer must fulfil the needed range for the acceleration measurements (i.e. full scale values) and to be able to operate at the necessary sampling frequency.
- A main constraint for the concrete microcontroller to be used is the online implementation of the algorithms, so the most important characteristics are memory capacity and the exhibiting computational capacity (throughput).

Memory needs were determined by estimating how many resources should be used by the algorithmic set, in terms of program and data memory. An additional space managed by the firmware, necessary for the storage of the acquired data was also considered.

The theoretical approximation of the required microcontroller computation time is difficult to be done, and for this reason a measurement strategy was organized at the laboratory. Measurements were done around a microprocessor based platform allowing the equivalent simulation of the algorithms.

5.3.1 Memory Requirements

The calculations of the memory size required by the execution of the algorithms and the firmware operation were performed by estimating the online deployment of the algorithms, according the different blocks indicated in Figure 5.1. In concrete, it was considered:

- A combination of algorithms executed on each acquired sample (detection of a fall, conditioning and filtering of the signal).
- A set of algorithms executed per window.
- An estimated need of the window management and the communication system.
- An estimation of the local storage system, using a local μSD card.

Table 5.1 shows the results of these estimations, in terms of necessary minimum size and type of memory (program or data memory).

5.3.2 Sampling Frequency and Full-Scale Values

Sampling frequency and full scale values are key characteristics for choosing the correct accelerometer. Frequency also imposes specific time restrictions on the online implementation of the algorithms, which finally creates important constraints on the specific microcontroller to be used for the implementation.

Table 5.1 Estimated memory usage

Description	Memory Type	Memory Usage (bytes)
Basic System with window Management and communication	Program Memory	6,5 KB
	Data Memory	5 KB
Bradykinesia Algorithm Memory Usage	Program Memory	3,5 KB
	Data Memory	0,5 KB
FoG Algorithm Memory Usage	Program Memory	3,5 KB
	Data Memory	0,5 KB
Dyskinesia Algorithm Memory Usage (FFT included)	Program Memory	3,5 KB
	Data Memory	0,7 KB
Filters+FallDetection Algorithm Memory Usage	Program Memory	1,2 KB
	Data Memory	0,5 KB
μSD for Debug purposes Memory Usage	Program Memory	2,8 KB
	Data Memory	4,2 KB
TOTAL	Program Memory	20 KB
	Data Memory	11 KB

Going along the different developed algorithms, it is necessary to evaluate the necessary sampling frequency to be applied. For this purpose, and also for considering the main frequency characteristics related to human movement the following points are considered:

- In the case of Dyskinesia, it was observed that this symptom increases the power spectra in the lower frequencies up to 4 Hz.
- In the detection of FOG episodes, it was observed that normal walking has a principal frequency around 2 Hz, while FOG episodes are characterised by a principal frequency in the range of 3–8 Hz.
- The developed method for detecting Bradykinesia uses the accelerometer signals associated with the patients' gait. The frequency content of gait is known to be below 20 Hz.

According the Nyquist theorem, the minimum sampling frequency should be 40 Hz. Therefore, the waist sensor must incorporate an accelerometer with a sampling frequency of at least 40 Hz. In order to keep the computing resources as low as possible, this minimum frequency is set as the real-time acquisition frequency.

Moreover, regarding the full-scale range of the sensor, according to the measurements obtained in the signals collected along the project, it is determined that a Full Scale of 6 g (where 1 g = 9,8 m/s^2) for the accelerometer is enough for the analysis of human movement with a sensor worn in the waist.

5.3.3 Time Restrictions on the On-Line Implementation

As previously mentioned, the estimate for the processing time requirement was made on a set of measurements of processing time spent by a microprocessor, configured to work at 40 MIPS, and running specific algorithms that have an equal or greater burden compared with the algorithms developed in the project for symptoms' detection. This is an indirect way to fix minimum processing requirements for the microcontroller to be included in the sub-system.

Considering that some operations must be done after acquiring a sample or when a time window has been completed, Figure 5.3 shows the temporal organization of the different computations included in the online implementation of the algorithms, according the data flow presented in Figure 5.1.

The timing of the accelerometer data collection is represented in orange (related to a sampling frequency of 40 Hz, with a time between samples of 25 ms). In the sample zoom (top section of the figure), it is shown the time necessary to both acquire the accelerometer measurements and to perform the algorithm's calculations that are done at every sample.

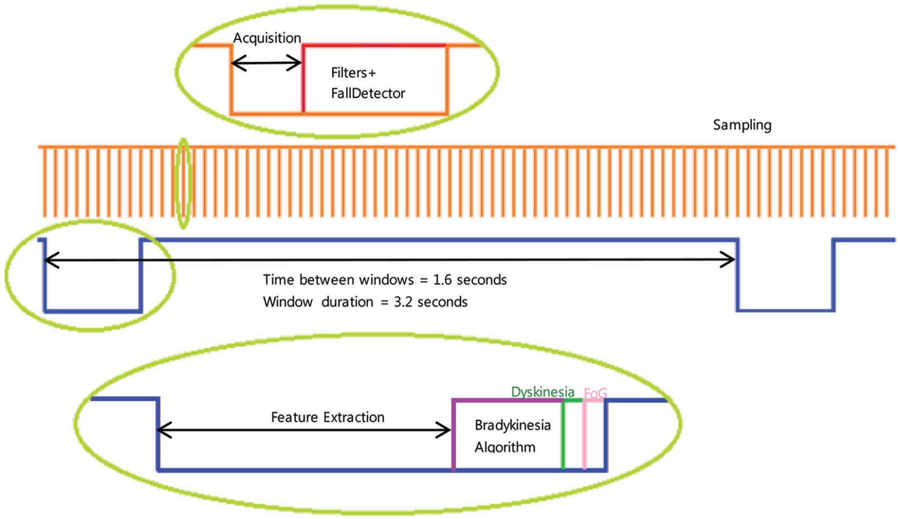

Figure 5.3 Timing for the online algorithms implementation.

The duration of a window, which is 3.2 seconds corresponding to 128 samples, is represented in blue at the bottom section. Note that a new window starts every 64 samples (every half a window), as described in Chapter 4. In the zoomed window, the computations done at the end of an acquired window are shown. These computations start from those provided by the processing done after each sample. Most of the time spent in the window computations correspond to the feature extraction. Once obtained, the bradykinesia, dyskinesia and FoG algorithms are applied. In order to ensure a correct processing, their results are obtained before the end of the next window is reached.

Table 5.2 presents an estimation of the processing time for each part in which the algorithms are divided. This estimation is based on the necessary

Table 5.2 Estimated processing time

Description	Timing	Time (ms)
Sampling Frequency	Between Samples	25
Adquisition	Every Sample	0,014
Filters + Fall detection algorithm	Every Sample	0,0318
Windowing time	Between Windows	1600
Feature Extraction	Every Window (max)	151
Bradykinesia Algorithm	Every Window (max)	55
FoG Algorithm	Every Window (max)	3
Dyskinesia Algorithm	Every Window (max)	3

operations to accomplish each computation in terms of memory usage and the inherent complexity of the calculus.

5.4 Sensor Device Components

This section describes the selection of the components for the sensor subsystem according to the presented requirements.

5.4.1 Microcontroller

The microcontroller (labelled as dsPIC33F in Figure 5.2) is responsible for the management of the data acquisition with a fixed sampling frequency, the analysis of the raw data applying the corresponding online algorithm and sending the processed data (or, alternatively, the creation of a local log in the µSD card).

The selected microcontroller in the project, for these purposes, was the Microchip® dsPIC33FJ64MC804. One of the main advantages of this device is the availability of an integrated DSP engine, which enables advanced computations in short time, when compared with regular microcontrollers (e.g., 32F, 24F, 18F and 16F). The microcontroller memory includes 44KB for program memory while the memory dedicated to RAM reaches 16KB (both are according the requirements indicated in Table 5.1).

This dsPIC architecture is known as "Modified Harvard" which uses 16 bits-long data and 24 bits-long instructions and it processes 40 Mega-instructions per second (MIPS). The DSP engine enhances the operational capacities of the µC. It allows 16-bit data multiplications for both fractional and integers, and it also allows inverse multiplication among 32 and 16 bits of data. The DSP, despite its relative small size, becomes useful when computation power is required, saving time to the main µC threat.

The dsPIC operational voltage range is 3.0 to 3.6 V (the whole system voltage supply is 3.3 V). Working at 40 MIPS and supplied with 3.3 V, the microprocessor consumption is 60 mA in "run mode" (normal operation mode), while in "idle mode" (special waiting state) its power consumption drops to 20 mA, and in energy safe mode ("sleep") maximum current peaks reach 28 µA.

An internal DMA (Direct Memory Access) module allows the communication between the CPU, the memory and the peripherals independently from the process being executed in the main thread of the program, enabling the execution of this process in parallel. The dsPIC has 8 DMA channels that may be associated to any peripheral I/O port. The DMA allows the parallel execution of various processes interrelated among them through asynchronous events (interrupts).

The whole operation of the system is effectively managed by the CPU using some internal specific peripherals, allowing:

- Data acquisition.
- Communication with the wireless module.
- Communication with the digital sensor.

Each peripheral has two associated DMA channels (transmission and reception) that provide access to a shared memory block.

5.4.2 Accelerometer Details

The selected accelerometer is the LIS3LV02DQ (LIS) manufactured by ST Microelectronics [1] which was, at the moment of the design, the only available Microelectromechanical System device (MEMS) using a digital interface within the device. The inclusion of the sensors, the signal conditioning and the converters within the same packaging ensures a very good performance against perturbations. Besides, the inclusion of the three axis (3D) in the same package also provides a better functional and geometrical symmetry. Finally, it is important to highlight that the accelerometer includes a signal compensation based on internal calibration curves, which compensates the measured signal based on the temperature sensor also included in the accelerometer integrated circuit.

The bandwidth used by the LIS is 640 Hz for each axis. Given the particularities of human motion, the sample frequency required for it is 40 Hz, therefore, an excessively small sample period is avoided. One of the main advantages offered by LIS is that the full scale may be chosen between two values (± 2 g or ± 6 g). The higher full scale allows a sensitivity of 340 LSB/g, which is the selected one for the device. The precision of the accelerometer is 2.9 mg for every bit change.

5.4.3 Bluetooth Module

According to the REMPARK system requirements, the movement sensor device has to communicate wirelessly with, at least, a mobile phone to share the output of the algorithms. The system has been developed including a wireless communication Bluetooth v2.1 + EDR (IEEE 802.15.1) chip to implement the communication channel. For this purpose, the sub-system includes a WT12 Bluegiga® communication module, with an on-chip integrated antenna.

The communication is organised through the UART port of the system. When the system boots the connection and communication parameters are configured; once communication is established, and because the SPP profile is used, the module works as a bridge between the UART port and the Bluetooth unit. This module works at 115200 bps and its consumption is 31.5 mA according to its technical specifications [2].

5.4.4 Power Management

The sensor includes a battery management system that tries to save as much energy as possible. As shown in Figure 5.4, the operation is based on 3 low-dropout regulators and each of them powers selected parts of the sensor: first regulator supplies energy to the microcontroller, which manages the operation of the remaining regulators. The additional two regulators supply the analogue and the wireless circuitry.

The system is powered by an 1130 mAh Lithium-Ion battery. According to the functional requirements for REMPARK system, sensor usage is suggested to be similar to a mobile phone. The system has a peak of 100 mA instantaneous current consumption; therefore, the worst case estimated life of the battery is a minimum of 20 hours. Then, the device should be charged a maximum of once per day, and normally during night.

The system incorporates a battery charger and a battery monitor. The battery monitor indicates to the user its current state through a RGB-LED.

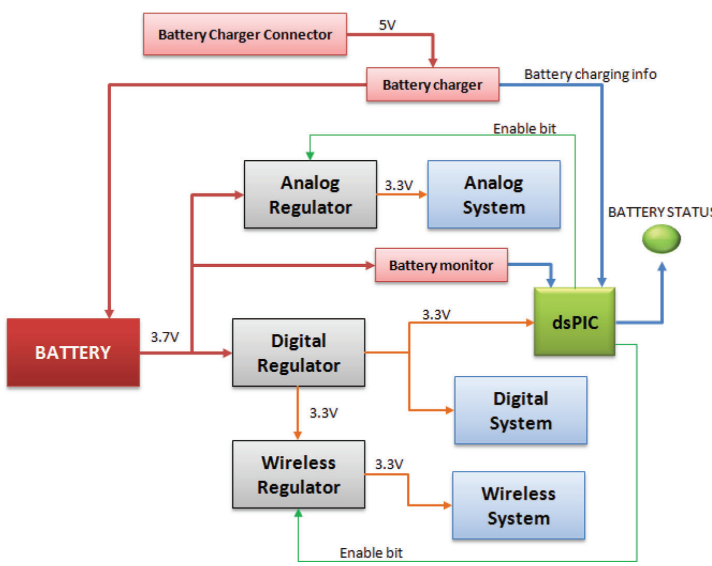

Figure 5.4 Power Management block diagram.

When the battery is very low, an interrupt is produced in the microcontroller and the system automatically closes all its peripherals and communications and enters in the "sleep" mode.

5.4.5 External Memory Unit

The sensor system contains an external memory unit in the form of a μSD card, as it has been introduced, managed through one of the available microcontroller SPI channels. This memory unit is intended to store, if desired, the raw data captured by the sensors. After extracting the μSD card from the sensor unit, it is possible to get the generated log file in order to analyse the inertial data. This functionality was especially useful during the REMPARK database construction process.

5.5 Sensor Casing and Operation

All the described electronic components of the sensor unit plus the Li-ion battery are encapsulated in a 99 × 53 × 19 mm plastic case. The total weight is 125 g (including the battery). The prototype also includes a wall battery charger. Figure 5.5 shows the casing view of the sensor unit.

As parts of the user interface, four elements can be externally identified in the unit: the main switch, an action button, the indicator LED and a charger connector. Figure 5.6 indicates the location of these components.

The behaviour of the sensor unit is determined by two states: Off and On. These states are mutually exclusive; a third state, related to the battery supervision, is compatible with the previous ones and may be understood as

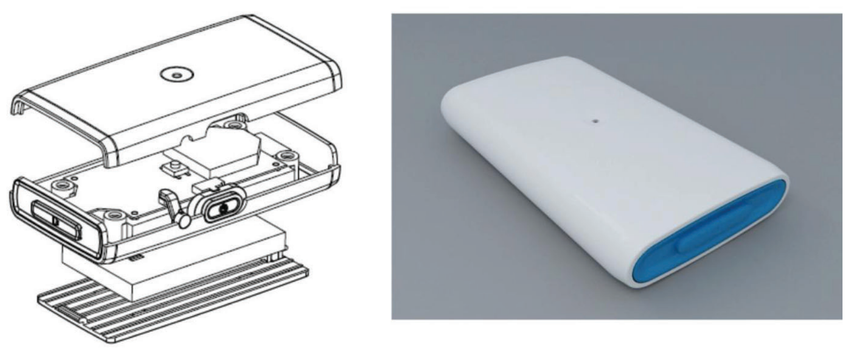

Figure 5.5 Sensor unit casing.

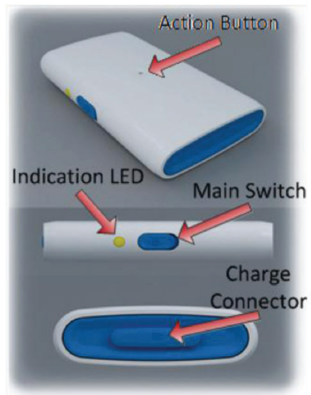

Figure 5.6 External components of the sensor unit.

an independent state. Figure 5.7 represents graphically the relations between the states of the sensor unit.

In the OFF state the sensor cannot work, no battery consumption exists and no battery level may be presented. Besides, the device does not react when the action button is pressed. In the ON state the sensor initializes the microcontroller. When the sensor has already initialized all the internal devices it is ready to start the sensing process. When ready, the sensor reads inertial signals and processes them. Data are sent every minute to the mobile gateway as described above.

The state of the sensor is indicated using a unique multicolour LED by taking advantage of its blinking light. Different states can be distinguished due to the diversity of the colour code of the LED. For example, a blinking green/yellow light means the sensor is in a sense and analysis process. A blinking green/blue light means the sensor is sending data. If the sensor is

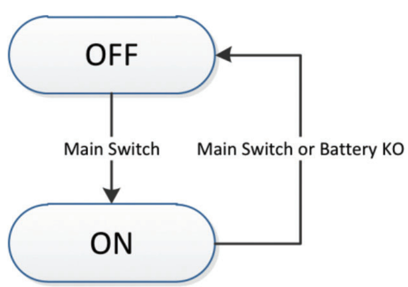

Figure 5.7 Status of the sensor unit.

Figure 5.8 Special belt and REMPARK final sensor.

charging the battery, it will be indicated with a fixed orange light. Otherwise, a fixed green light will notify battery if fully charged. Finally, error state will be indicated by means of a fixed red light.

The sensor unit case is surrounded by a retention mechanism (belt) able to fix the sensor unit on the patient's waist, as depicted in Figure 5.8. The part of the belt that is in direct contact with the patient's skin has been manufactured using a biocompatible neoprene material.

5.6 Conclusion

The specification of the concrete requirements has been done along the chapter, allowing the selection of the most convenient components integrating the sensor unit of the REMPARK system. The microcontroller is capable to embed all the developed algorithms and to execute them in real-time and considering all the restrictions related with the data capturing timing.

The system is light and small, being suitable to be worn in a belt without being intrusive for the patient. Furthermore, the system is capable to store raw data and to send messages to an external device. In this way, the developed sensor complies, thus, largely with all the established requirements, becoming the heart of the REMPARK system.

References

[1] STMicroelectronics, Inc. LIS3LV02DQ datasheet. MEMS Inertial Sensor., (2005).
[2] Bluegiga. WT12 data sheet versión 2.5., (2008).