CogWatch – Cognitive Rehabilitation of Apraxia and Action Disorganisation Syndrome

D2.2.1 Report on devices I

Deliverable No.		D2.2.1	
Workpackage No.	**WP2**	Workpackage Title	**System Devices and Networks**
Task No.	**T2.2**	Activity Title	**Monitoring and Feedback Devices**
Authors (per company, if more than one company provide it together)		**José M. Cogollor, Javier Rojo and Manuel Ferre (UPM-ROMIN), Laura Pastor, Matteo Pastorino and Maria Teresa Arredondo (UPM-LST), Chris Baber, Manish Parekh and Martin Russell (UOB-EECE)**	
Status (F: final; D: draft; RD: revised draft):		**F**	
File Name:		**Cogwatch_D2.2.1_Devices_I_Final**	
Project start date and duration		**01 November 2011, 36 Months**	

EXECUTIVE SUMMARY

This deliverable presents the devices to be used in the first prototype of CogWatch project. They are derived from the previous deliverable D2.1 Report on system specification whose objectives were the analysis of system specifications and the definition of the architecture.

All the devices involved in the first prototype of CogWatch are defined in terms of hardware, software, firmware and communications. The purpose of the report is to provide a technical overview of these devices, which are classified depending on the function they provide, so there are devices for monitoring and for feedback.

The final CogWatch architecture for the first prototype is detailed through block diagrams and an explanation of the functionalities to be finally implemented.

Taking into account the goal of this document, the first prototype will comprise a wearable component for feedback (watch), sensorized objects for the execution of the task (milk container, kettle and cup), an all-in-one (host) computer for feedback and Kinect™ sensor for monitoring and visual information capturing but the principal functionality will be the patient hand tracking as the patient recording would be done alternatively with other devices. Although in this prototype, the sensorized shirt from RGB Medical Devices will not be implemented, a summary of the state of this future component will be introduced.

Finally, the first prototype composed by the devices described in this document will be installed for technical evaluation in the Living Lab of UPM (Spain), where it will be trialled by a non-patient group.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

REVISION HISTORY

Revision no.	Date of Issue	Author(s)	Brief Description of Change
v0	20/09/2012	UPM	First draft of the contents.
V1	02/10/2012	UPM	Executive summary and Introduction.
V2	05/10/2012	UPM	Section 2 and contribution to section 3 (Kinect™ and Watch).
V3	09/10/2012	UoB EECE	General reviews, section 2 and contribution to section 3 (Sensorized objects and tools).
V4	15/10/2012	UPM	Contribution to section 3 (Kinect™ and VTE Monitor).
V5	15/10/2012	UPM	Conclusions, sections renumbered, text editing, formatting, Table/Figures numbering revisions. Header (public to private), review revisions.
V6	26/10/2012	UPM	Final version after review.
Final	30/10/2012	Maria Teresa Arredondo and Matteo Pastorino	Minor changes and Revision of the modifications after the peer review.

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviation	Abbreviation
AADS	Apraxia and Action Disorganization Syndrome
ADC	Analogue Digital Converter
ADL	Activities of Daily Living
API	Application Programming Interface
CAT	Computed Axial Tomography
CMOS	Complementary Metal-Oxide-Semiconductor
COLLADA	COLLAborative Design Activity
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CTS	Clear To Send
DC	Direct Current
DRP	Digital Radio Processing
ECG	Electrocardiogram
FPS	Frames Per Second
FSR	Force Sensitive Resistor
GUI	Guide User Interface
HCI	Host Controller Interface
LCD	Liquid Cristal Display
MRI	Magnetic Resonance Imaging
NIBP	Non Invasive Blood Pressure
OLED	Organic Light-Emitting Diode
PHS	Personal Healthcare System

RF	Radio Frequency
RFID	Radio Frequency Identification
RTOS	Real Time Operating System
RISC	Reduced Instruction Set Computer
SDK	Software Development Kid
SPO2	Blood Oxygen Saturation
TFT	Thin Film Transistor
TI	Texas Instruments
VTE	Virtual Task Execution

1. INTRODUCTION

This section introduces the deliverable D2.2.1 Report on devices I for the first prototype of CogWatch project. It provides the main purpose and scope of the system.

1.1 Purpose

This Report on devices I describes the components and functions of the devices to be installed and used in the first prototype for this CogWatch project in terms of their hardware components, software and firmware architectures.

CogWatch requires the development of novel tools and objects, portable and wearable devices as well as ambient systems to personalise the cognitive rehabilitation carried out at home for stroke patients with AADS symptoms. The specifications of these devices were presented in the previous deliverable D2.1 Report on system specification.

1.2 Scope

Traditional approaches to rehabilitation for AADS include the therapist providing verbal or visual cues as the patients perform selected tasks (such as making a cup of tea). Repeated practice with encouragement from the therapist, leads to improving performance and avoiding mistakes that can occur in a trial–and–error approach and at the same time avoiding the danger on the patient. It also leads to the perseveration of the mistakes.

This approach is labour intensive for therapy time and also limits the opportunity for the patient to practice the skill. For these reasons, the CogWatch project proposes a novel solution to AADS cognitive rehabilitation by providing an instrumented environment in the kitchen, including motion tracking with action recognition software, which allows cueing of the patient together with monitoring for errors.

Moreover, CogWatch can lead to be a Personal Healthcare System (PHS) that delivers personalised, long–term and continuous cognitive rehabilitation of activities of daily living (ADL) for stroke AADS patients at home using portable, wearable and ubiquitous interfaces and virtual reality modules. It is customized to suit the needs of individual patients at the same time as being practical and affordable for home installation so that rehabilitation takes place in familiar environments performing familiar tasks.

The CogWatch project requires activity to be captured in everyday settings to enable analysis which can support decisions about how well the activity is being performed and whether it might require support or guidance.

In broad terms, one can approach the challenge from three directions:

- Instrumenting the environment, primarily through vision-based systems.
- Instrumenting the person, through sensors on the person's hands or arms.
- Instrumenting the objects themselves.

The first of these approaches is being explored through other work in CogWatch, using marker-based systems such as the Qualisys motion capture system and marker-less tracking using the Microsoft Kinect™. These are discussed in section 3.1.1 of this report.

The second approach involves placing sensors on the person. These sensors could include the electromagnetic trackers, such as the Zebris (ultrasonic) or Polhemus (electromagnetic) device used by CogWatch partners in TUM to give coordinates of the hand in 3D space (see D3.2.1 Report on data analysis for action recognition 1), or accelerometers on the hand to capture motion [1]. While this approach offers useful data to characterise a wide range of human activity, it suffers from the problem of requiring attachment of sensors to the person. This is potentially problematic for the following reasons:

- It requires the individual to have sensors fitted to them, or to wear bespoke clothing with the sensors attached which could affect comfort and performance.
- It requires the sensors to be positioned correctly on the person.
- It requires actions to be performed which best utilise the sensors in their fixed positions.

For these reasons in section 3.1.2 a solution to this problem focussed in sensorized objects is widely described.

The CogWatch system is being developed in relation to a set of scenarios involving ADL tasks. Activity of Daily Living comprises tasks of basic self-care such as preparing food and drinks in the kitchen, using the toilet, or washing and grooming in the bathroom. Their performance involves a sequence of component actions on environmental objects directed at some desired end goal. It is thought that successful performance depends on specifying object actions in spatiotemporal terms at a higher cognitive level and then elaborating these into specific movements of limbs, which are monitored as they progress against expected sensory consequences.

In this first prototype, the first task to be executed in order to evaluate the system and at the same time the future patients, is that focused on preparing tea.

2. GENERAL ARCHITECTURE

Figure 1 presents a description of the architecture designed for the first prototype in terms of devices. The final functionalities to be provided by each component are also detailed.

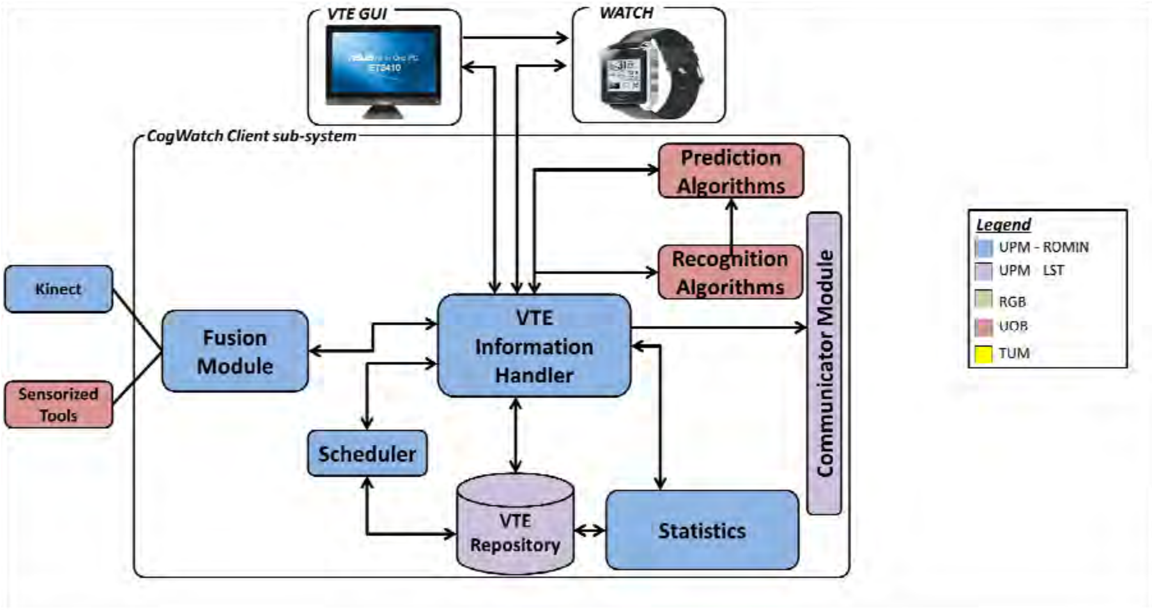


Figure 1 - General architecture of devices for first prototype in CogWatch.

As shown in the figure, the architecture presented describes the devices to be implemented in the prototype. This representation is complemented by that one in the corresponding deliverable D2.3.1 Report on networks I. This document will be focused on the specifications of each device that composes the global architecture. Deliverable D2.3.1. comprises the architecture from the point of view of the integration of all the modules.

The devices to be considered are divided into two categories as explained in the next section: monitoring devices and feedback devices. Regarding the monitoring devices, these are in charge of monitoring the execution of the task and movements of the patient in order to be able to detect possible errors and malfunctioning. Meanwhile, feedback devices are in charge of providing the corresponding cues and feedback to make the patient aware of the errors committed and possible risks.

First of all, the monitoring devices considered are: Kinect™ and sensorized objects and tools. Kinect™, which will be connected by wire to the All-in-one computer (VTE GUI monitor), will be in charge of obtaining:

- Position of the patient's hands.
- Colour, depth and 3D images from the whole scenario and patient behavior.

The information provided by Kinect™ will be used for analysis, movement recognition and evaluation of the patient behaviors.

Meanwhile, the sensorized objects and tools will be manipulated by the patient in order to execute the corresponding task. These objects will be fitted with accelerometers and force sensitive resistors to capture interaction by the user. Data from these sensors will be collated by a micro-controller and then communicated, via Bluetooth, to the host computer.

Considering the feedback devices, these are composed by: watch and VTE monitor. The watch is in charge of:

- Vibration when the patient is grasping an incorrect object or executing an incorrect task or in a bad way.

So, the watch will be paired with the host computer via Bluetooth and will be the only component of the first prototype in providing vibrotactile feedback when necessary. It is important to mention that the functions of this device will be extended for the second and final prototype during the CogWatch project.

The VTE GUI monitor (host computer) will be in charge of providing:

- Virtual Task Execution (VTE) and/or figures of the action which patient has to execute after an error is detected or in case of ignorance.
- Verbal cues for complementing the simulation/figure.
- Text messages and sounds for warning.
- Tactile user interface (GUI) for starting/stop the application/simulation and for selecting the task.
- Collecting all the corresponding data in the processor.

The first four functions mentioned above regarding the VTE monitor are focused on the interaction of the patient with the system. The last function is composed by all the blocks that comprise the CogWatch Client Sub-System, as shown in Figure 1.

Finally, although sensorized shirts will not be used in the first prototype a summary of the current state of the work from RGB Medical Devices is included.

RGB has been working on the development of an embedded textile solution to integrate the measurement of several Vital Signs, such as ECG, SPO2 and NIBP (Figure 2). The following issues have been considered in the case of ECG:

- Validation of wearable system.
- Acquisition and improved electrodes data analysis: It is being studied the development of electrodes by means of "printed electronics" with different formulations (Ag, Ag/AgCl inks) as well as the effect of deposition of hydrogel membranes over textiles / printed electrodes.
- Improvement on the ergonomy of the wearable device. Substitution where possible of tunneled cables by means of textile conductive printed /textile links over textile.
- Wireless technology provides BT connection so that the user gets feedback messages on the process.

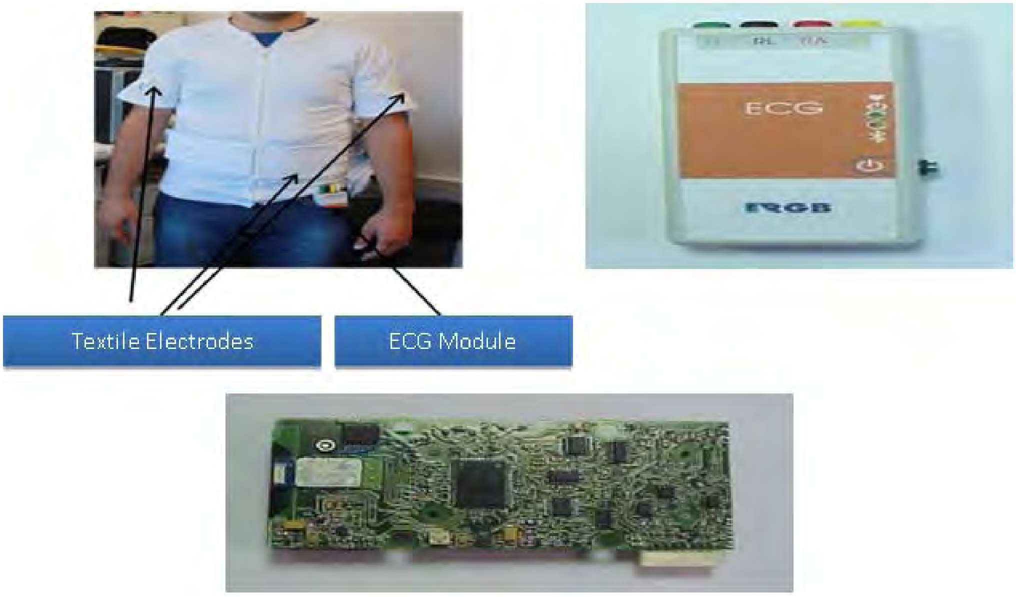


Figure 2 - RGB hardware: module and electrodes.

The integrated electrodes have been made with conductive tissue using nylon tissue covered with silver "nylon knit fabric silver coated", since it provides the best contact impedance value and being also more comfortable to wear.

In order to minimize artefacts to movement, the design of the shirt must guarantee a good and tight adjustment of the electrodes to the body. Therefore, in the arm electrodes, a Velcro® ribbon has been placed (Figure 3). The ribbon material absorbs the contour variations due to arm movement, which would otherwise affect the skin-electrode contact.

The Waist electrodes can be tightly fixed to the body by means of a belt that can also be used to allocated vital signs modules.

Signal conduction from electrodes to the acquisition modules is made with ECG cable tunnelled though hidden tubes, using the same tissue as the shirt, and connected by means of snaps. The correct operation of the device was validated in healthy volunteers.

(a) **(b)**

Figure 3 - RGB Shirt: (a) Inside view; (b) outside view of adjustments in arms and waist.

3. COGWATCH FIRST PROTOTYPE DEVICES

This new section aims to describe in more detail all the devices defined above in terms of hardware and software architecture, taking into account the classification proposed.

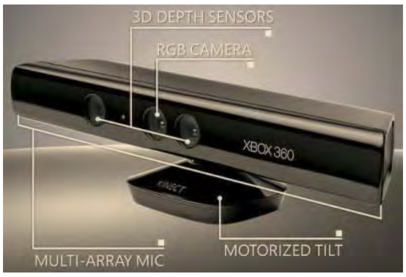
3.1 Monitoring Devices

3.1.1 Kinect™

First, it is relevant to mention that Kinect™ has already been used before in rehabilitation applications, especially, to assist in the recovery from physical and cognitive deficits after stroke [2], [3] and [4]. Moreover, GestSure Technologies has developed an application to help doctors navigate MRI's and CAT scans during surgery; Jintronix has developed a software application that allows patients with recovering from stroke to perform physical therapy exercises from within their own home; and the Johns Hopkins University uses Kinect™ and gesture based tele-surgery to help in fine and precise manipulation of surgical tools while conducting surgeries.

3.1.1.1 Hardware description

Kinect™ (Figure 4) sensor is composed by an IR projector and receiver for depth measuring, a simple RGB camera, and a multi-channel microphone.



(a)

(b)

Figure 4 - (a) A photo showing all the different parts of the sensor, external view of the 3D depth sensors, RGB camera, multi – array microphone, and a motorized tilt taken at Microsoft's E3 2010. (b) Internal view of the device from [5].

The working principle is well described by the functionality of their components. First a simple RGB camera manages the capture of images at 30 fps. Second, the infrared provides the information about the depth of the different elements involved in the task in the workspace. The method of determining 3D position for a given object or hand in the scene is described by the inventors as a triangulation process [6]. As described in [7] a single infrared beam is split by refraction after exiting a carefully developed lens. This refraction creates a point cloud on an object that is then transmitted back to a receiver on the assembly.

Table 1 provides the relevant information about Kinect™ features:

SENSOR ITEM	SPECIFICATION RANGE
Viewing angle	43º vertical by 57º horizontal field of view
Mechanized vertical tilt	+/- 28º
Frame rate (depth and colour stream)	30 frames per second (FPS)
Resolution, depth stream	QVGA (320 x 240)
Resolution, colour stream	VGA (640 x 480)

Table 1 - Kinect™ principal features for capturing image video and motion.

On the one hand, the colour camera provides a colour image of the environment. Colour data is available in the two following formats:

- RGB colour which provides 32-bit, linear X8R8G8B8-formatted colour bitmaps, in sRGB colour space.
- YUV colour which provides 16-bit, gamma-corrected linear UYVY-formatted colour bitmaps, where the gamma correction in YUV space is equivalent to sRGB gamma in RGB space. This format uses less memory when holding bitmap data and allocates less buffer memory when the stream is opened due to that the YUV stream uses only 16 bits per pixel.

Both colour formats are computed from the same camera data, in order to represent the same image by using both the YUV data and RGB data at the same time.

The sensor array uses a USB connection to pass data to the VTE monitor, and that connection provides a given amount of bandwidth. The Bayer colour image data that the sensor returns at 1280x1024 is compressed and converted to RGB before transmission to the runtime. The runtime then decompresses the data before it is passed to the application. The use of compression makes it possible to return colour data at frame rates as high as 30 FPS, but the algorithm that is used leads to some loss of image fidelity.

On the other hand, the depth data stream provides frames in which each pixel represents the Cartesian distance, in millimeters, from the camera plane to the nearest object at that particular x and y coordinate in the depth sensor's field of view. The following depth data streams are available:

- 640×480 pixels.
- 320×240 pixels.
- 80×60 pixels.

A depth data value of 0 indicates that no depth data is available at that position, because all the objects were either too close to the camera or too far away from it. This and other important aspects of programming with Kinect™ can be reviewed in [8].

The following figure, Figure 5, shows a representation of the point cloud process and how the depth view works and is useful for distinguish between the kettle, the mug and the hand in the right part.

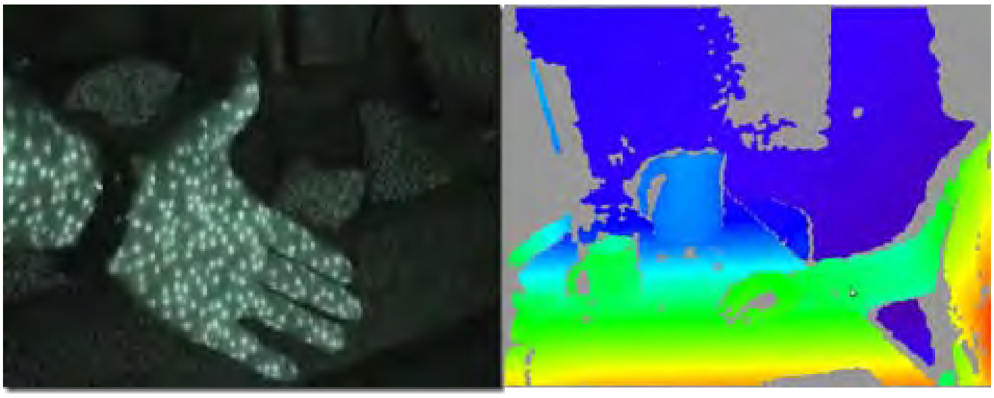


Figure 5. Kinect™ hand tracking and infrared view.

In the following section the functionality of how the Kinect™ is used for getting the hand positions to complement information for the prediction and action recognition algorithms is described.

3.1.1.2 Firmware, drivers and programming design

In 2011, Microsoft released the Software Development Kit (SDK) focused on Kinect™, which has allowed developers and authors to develop and create different applications depending on the objective pursued with Kinect™**.** Developers have designed non-gaming applications for areas such as interior design, tracking consumer behaviour, and for educational purposes. Regarding CogWatch project, the SDK has supported the corresponding functions in order to develop the correct software and interface to ensure the communication with the device and the performance of the whole device.

Using complex built-in firmware, Kinect™ can determine the three – dimensional position of objects and hands in its line-of-sight by this process. The main advantage of this assembly is that it allows 3D registration without a complex setup of multiple cameras and at a much lower cost than traditional motion labs and robotic vision apparatuses.

The feasibility of the data for action recognition and the comparative with a better class motion capture system where realised with positive conclusions and published in [9].

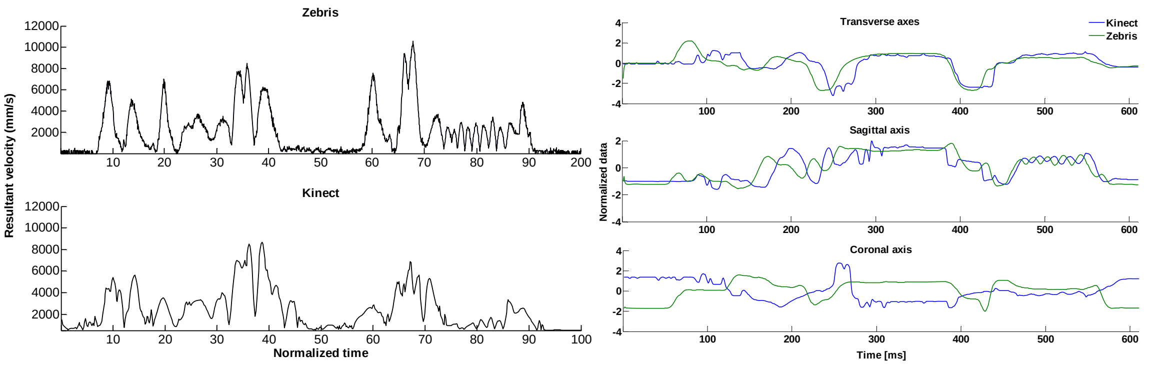


Figure 6. Comparative of velocity(left) and hand position across the 3 Cartesian axes(right) [9].

Figure 6, shows the data comparison between the Kinect™ and a better class motion caption system, Zebris. This data are filtered and processed for providing the hand position.

The Data acquisition from Kinect™ leads in a 30 fps video and a pure data file with the hand positions coordinates at each frame. All these data are properly dated and packed.

Figure 7 summarized the Kinect™ data acquisition process:

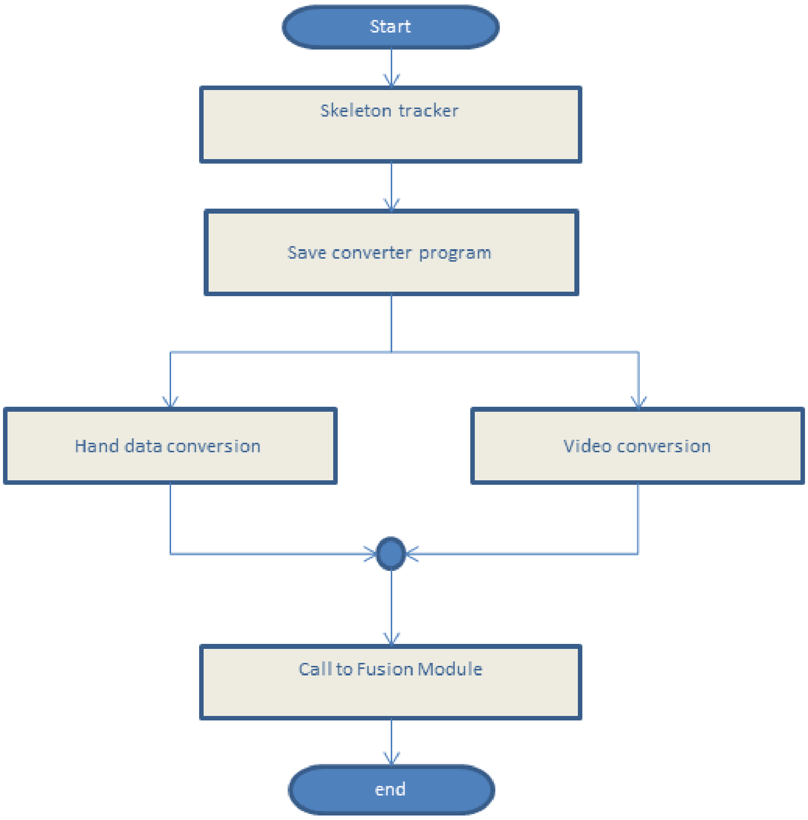


Figure 7 - Kinect™´s acquisition process diagram.

Firstly the skeleton tracker program is launched for acquiring the patient skeleton and the position of the hands then the data is prepare to be saved in common formats and the data is dated with a header and a time stamp for each frame. Then the information is ready to be sent to the fusion module where it will be combined with information from other sensors.

Internally the Kinect™ data acquisition process is described in the following diagram. First, the data are acquired so as to throw the camera sensor using Kinect SDK© and the interface developed, second, the data is filtered and synchronized. Finally, the information is stored and sent to the fusion module.

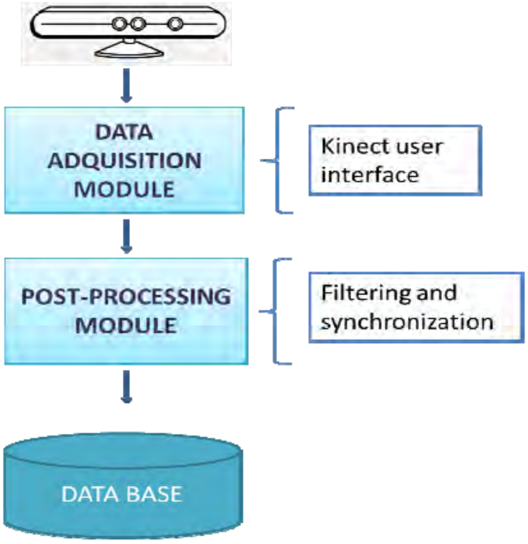


Figure 8. Data acquisition process.

3.1.2 Sensorized objects and tools

In this section, the current range of sensor units is considered. While problems involving monitor everyday activities described in the introduction section are typically dealt with through the recruitment of compliant participants in controlled settings, the range of capabilities of participants in CogWatch trials could mean that sensor sets would need to be designed, fitted and calibrated for each individual. Thus, the third approach could provide useful data to characterise activity without the need to attach sensors to the person. Previous work has shown that sensors on objects and tools can be used to capture activity in maintenance tasks [10] [11]. CogWatch is developing novel implementations of these concepts to capture domestic activity. For this report, the focus will be on the use of instrumented coasters which sit underneath mugs, jugs and kettles and which can be used to capture aspects of user activity.

3.1.2.1 Hardware description

The instrumented coaster is designed to provide data that can be used to recognise or analyse interactions with everyday objects such as mugs, jugs and kettles (Figure 9). The aim of the design is to integrate sensors onto the objects themselves without interfering with their use or significantly modifying their appearance, and without requiring any device to be fitted to the person using them. To achieve this aim, a wireless device that fits discreetly underneath the object was created. We call this device the CogWatch Coaster. The sensors contained in the CogWatch Coaster include a 3-axis accelerometer and 3 force sensitive resistors, together with a Bluetooth transmitter to send the data to the host computer. This choice of sensors was made on the basis of three criteria:

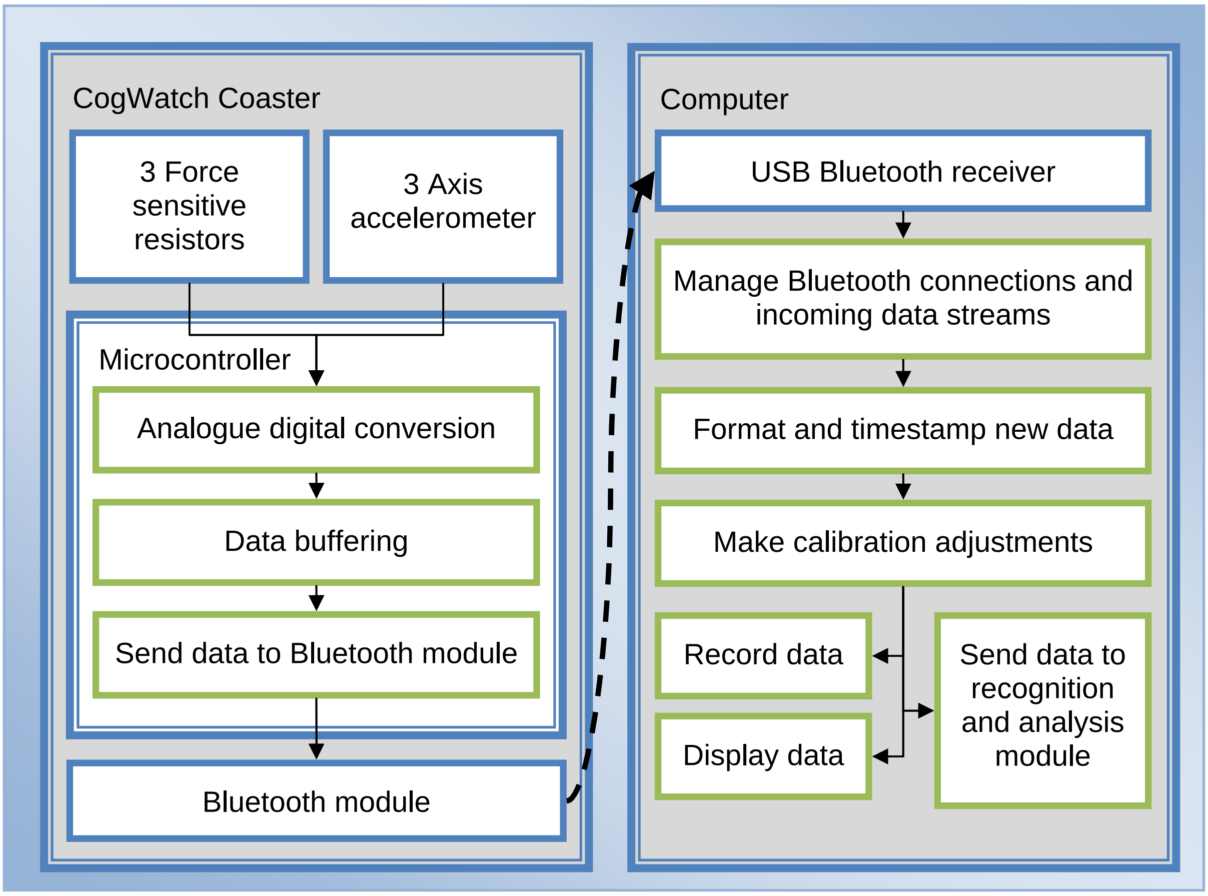
- Capture of data relevant to the modelling of ADL: the current models of ADL (both in terms of predictive models and in terms of experimental design) involve the movement of objects and their change in state. Changes in state are currently defined in terms of the addition of materials, such as water and milk. Thus, the minimal set of sensors required for these models would record motion (hence the accelerometers) and change in weight (hence the force sensitive resistors). Later versions could include additional sensing capability as required by the modelling work.

- Size of the CogWatch Coaster: Although there are many other sensors that could be added, doing so would increase the size of the device making it obtrusive to use. The form factor for the original CogWatch Coaster was defined by the circumference of a coffee mug: the design fits exactly under the base of a mug. A further design was made to fit under the kettle.
- Power and Battery Life Considerations: The selection of sensors was based on reasonable power efficiency relative to cost of the sensors and their processing capability.

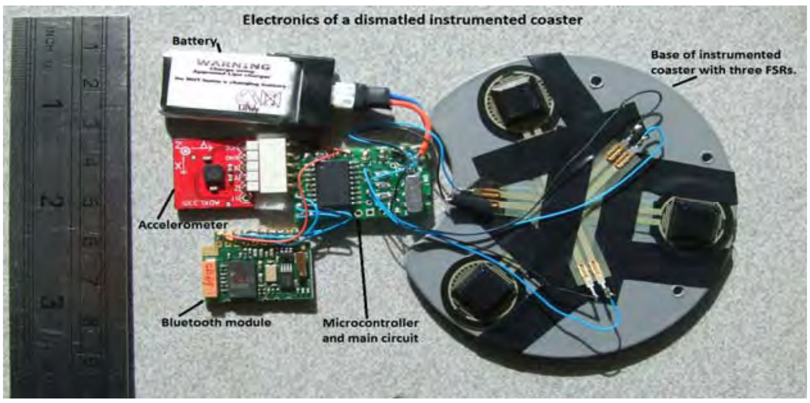
Figure 9 - Kettle, mug and milk container with instrumented coaster in a tea making task trial.

The instrumented coaster (Figure 10) is controlled by a Microchip dsPIC30F3012 microcontroller. This microcontroller has an integrated 12 bit analogue digital converter (ADC) that is used for digitising the sensor readings. The ARF7044 Bluetooth module is used to transmit the sensor data to a computer via a Bluetooth wireless connection. The microcontroller is programmed to digitise, compress and prepare the sensor data and manage the transmission of the data via the Bluetooth module. The data are buffered on the microcontroller so that they can be re-transmitted (avoiding data loss) if the wireless connection is interrupted for a short period.

The entire circuit is powered by a 260mAh 3.7V rechargeable LiPo battery, which provides a minimum of 2 hours operation (future revisions will be more efficient). The battery is charged using a separate circuit that can be USB or mains powered.



(a)



(b)

Figure 10 - (a) CogWatch Coaster system design; (b) The base of the CogWatch Coaster with FSRs fitted and the circuit.

ACCELEROMETER

The accelerometer used in the coaster is the Analog Devices ADXL335. It provides acceleration measurements on 3 axes in a range of ±3g. This accelerometer provides an analogue output signal that is read by the ADC of the microcontroller.

There are multiple kinds of interaction with instrumented devices that the accelerometer is well suited to detect:

- When an instrumented object is moved, the accelerometer measures the changes in motion.
- The accelerometer can be used to detect the orientation of an object; accelerometers interpret gravity as continuous upwards acceleration, the distribution of this force between the three axis of the accelerometer can be used to discern the orientation of the device. This effect is useful in detecting actions such as pouring with a kettle, or detecting whether an object is being held in an incorrect orientation.
- Accelerometers are well suited to detecting the shock of an object being dropped.

FORCE SENSITIVE RESISTORS

Force sensitive resistors (FSRs) are used to measure changes in the weight of the instrumented object. As force is applied to an FSR the electrical resistance decreases. Using a voltage divider circuit the resistance is changed into a voltage that can be read by the microcontroller ADC.

The FSRs are less than 20mm wide and less than 1mm thick so they are well suited the size of the instrumented coaster. Three sensors is the minimum number that can be used to support the entire support the full weight of an object with stability. All the objects weight must go through the sensors otherwise inconsistent weight distribution would create unreliable readings.

The measures of changes of weight of an object can be used by the computer to detect when liquids are added or removed from a mug or a jug for example. When the sensors are not in contact with any surface their weight reading becomes zero, allowing them to be used to help detect when an item is picked up.

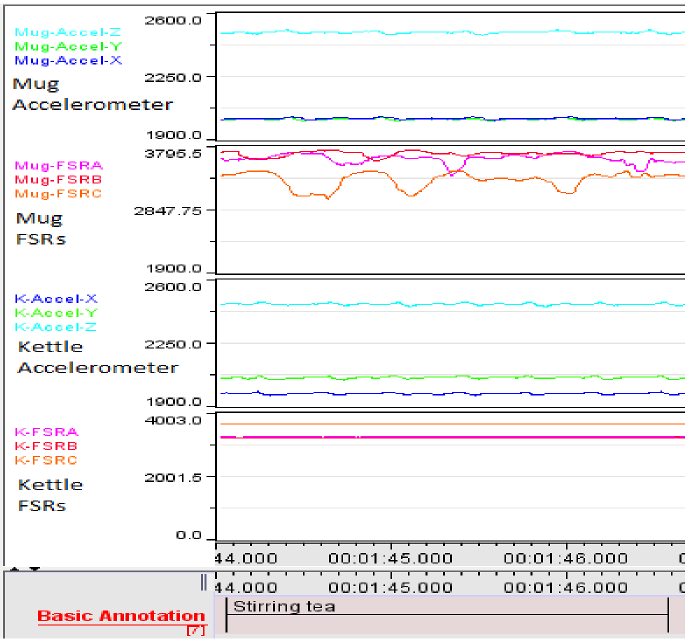


Figure 11 - The effect of stirring a mug of tea on the mug's FSR sensors.

Initial tests have shown that, the FSRs also pick up subtle changes in weight distribution across the base of the coaster. For example, it might be possible to detect when a mug of tea is being stirred (Figure 11); the motion in this case being far too small to be picked up by the accelerometer. However sometimes this effect creates noise such as detecting the wobbling of a table when irrelevant actions are performed (Figure 12). The utility of some of these non-obvious outputs from the FSRs need to be evaluated automatically by the recognition algorithms once there is sufficient data.

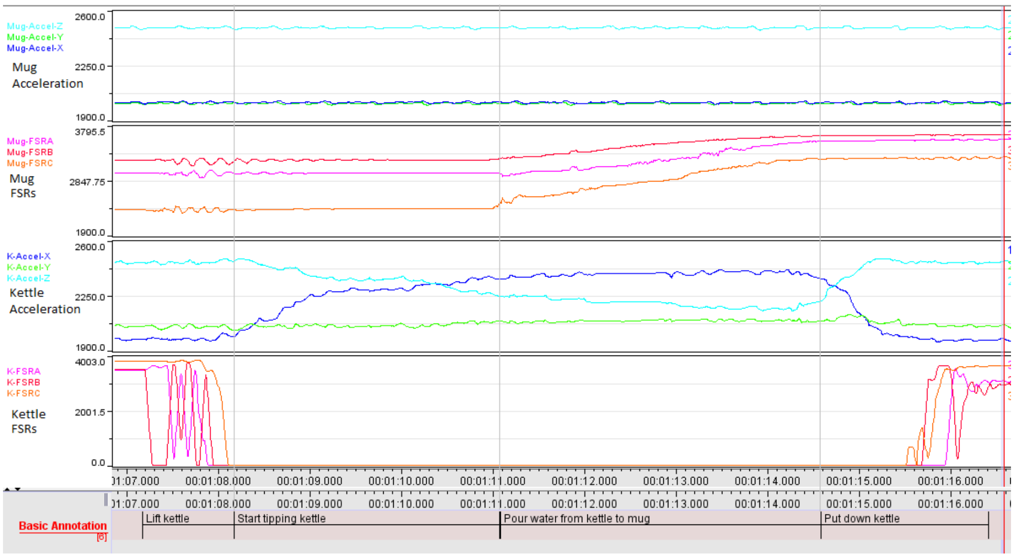


Figure 12 - The sensor data from a mug and kettle when water is poured from the mug to the kettle.

CALIBRATION

There are a many factors that affect the accuracy and repeatability of measurements from the FSR sensors. The contents of the instrumented object, the slope and the evenness of the surface the object is on, the distribution of weight between the 3 force sensors can all affect the readings. This can cause large changes in the three sensor readings when an object is moved, even without any changes in the actual weight of the object. The raw sensor readings cannot directly be summed to obtain a single unified reading from the three sensors due to the non-linear nature of the voltage divider and the resistance output of the sensors. The non-linear effect of the voltage divider can be removed using the voltage divider equation $Rfsr = \frac{Rfixed}{\left(\frac{Vin}{Vout}\right)-1}$, which results in the resistance of the FSR.

The datasheet for the FSRs claim that the response to force of the FSR is roughly 1/R. Figure 11 shows that the 1/R approximation results in a strong linear relationship:

The output of the FSRs is affected by the size shape and configuration of the contact to the active area of the sensor. For this reason the FSRs were individually tested to ascertain their accuracy and output relationship to weight.

The FSR sensors were tested as part of a complete CogWatch Coaster attached to a mug prototype. This was done to replicate real usage as any changes can greatly affect results. Separate tests were done from the individual sensors and the whole device. For each test, the mug + coaster set was placed on a digital scales. Water was poured into the mug to increase the weight to set values. When the scale reading indicated a value, the outputs from the FSRs were recorded. This was repeated for a series of weights ranging from an empty to a full cup of water. The measurements include all of the weight supported through the sensors including the coaster and mug. The total weight of the mug and CogWatch coaster is 275g, which makes this the minimum weight that can be measured when all three FSRs are used.

The results for individual sensors (Figure 13) show that sensor B has quite different readings to the other 2. The results are within the datasheet's part-to-part tolerance of ± 25%. The difference in gradient of the best fit line is of interest because it means its values cannot be summed with the other sensors without introducing error when weight distribution changes. Measurements like these will be used as calibration to adjust the data before the recognition/analysis algoritms are applied. This is especially important for detecting small changes in weight due to actions such as adding milk to tea, which has the potential to be obscured by error. Figure 14 shows the response of all three sensors together as weater is added to the cup, from empty to full. It shows that the FSRs respond with a readable increase to changes as small as 10g (which is equal to a small portion, i.e., around 2 teaspoons, of milk). However this may not always result in reliable detection because of noise (such as the table shaking the mug as shown at the start of Figure 12) could be enough to drown out changes this small. Further reduction of error though calibration may help when attempting to filter out such noise.

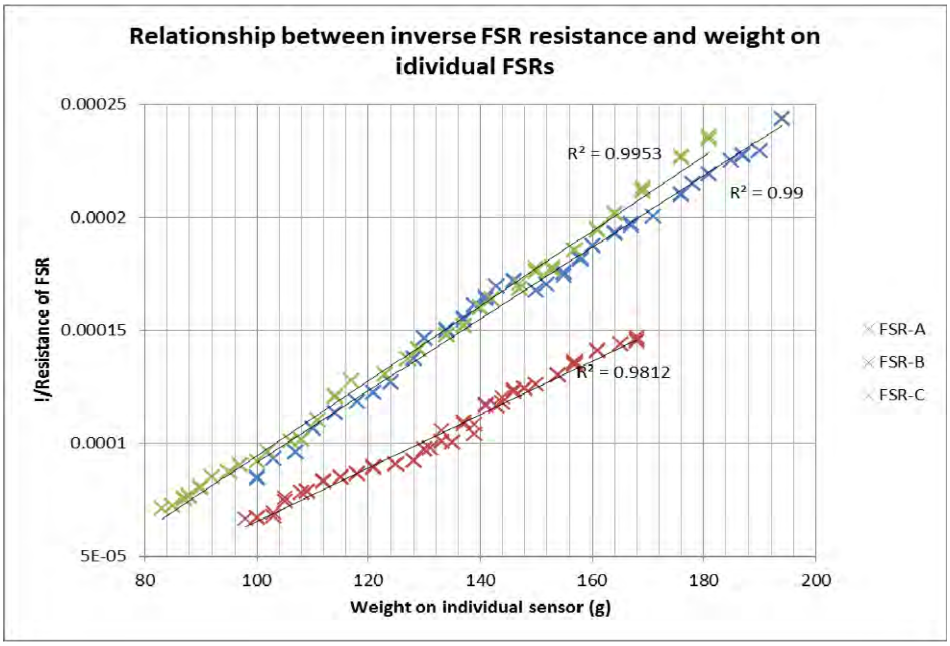


Figure 13 - Graph of 1/R against weight for individual sensors measured separately.

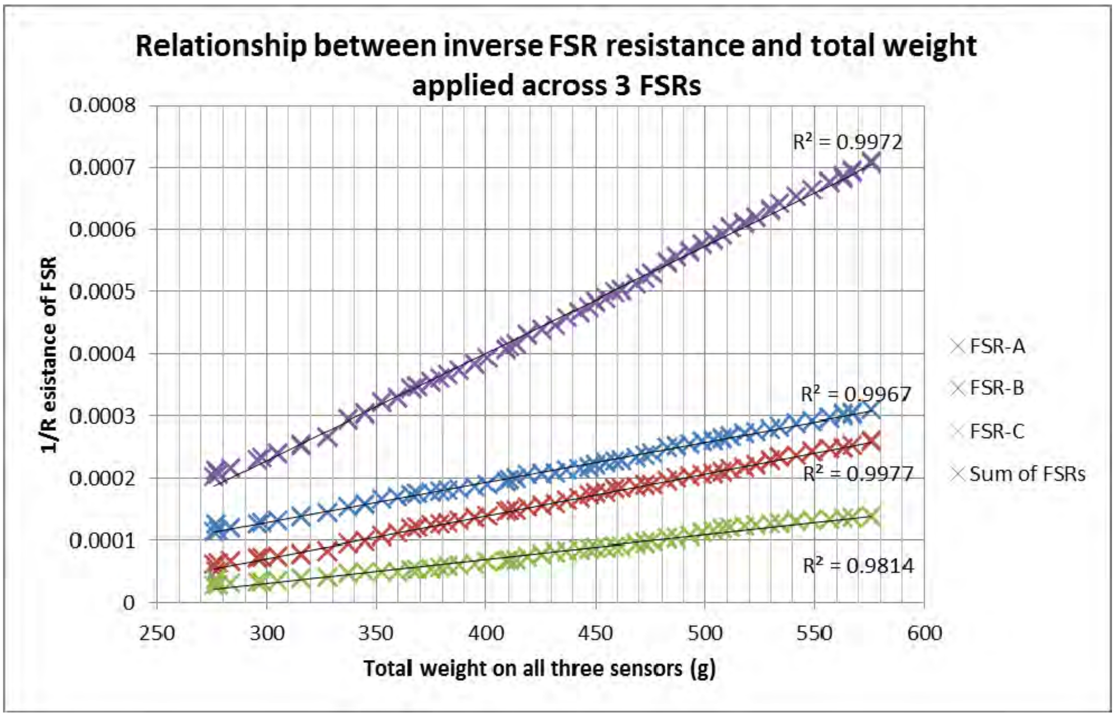


Figure 14 - Graph of 1/R against the weight applied across all three FSRs.

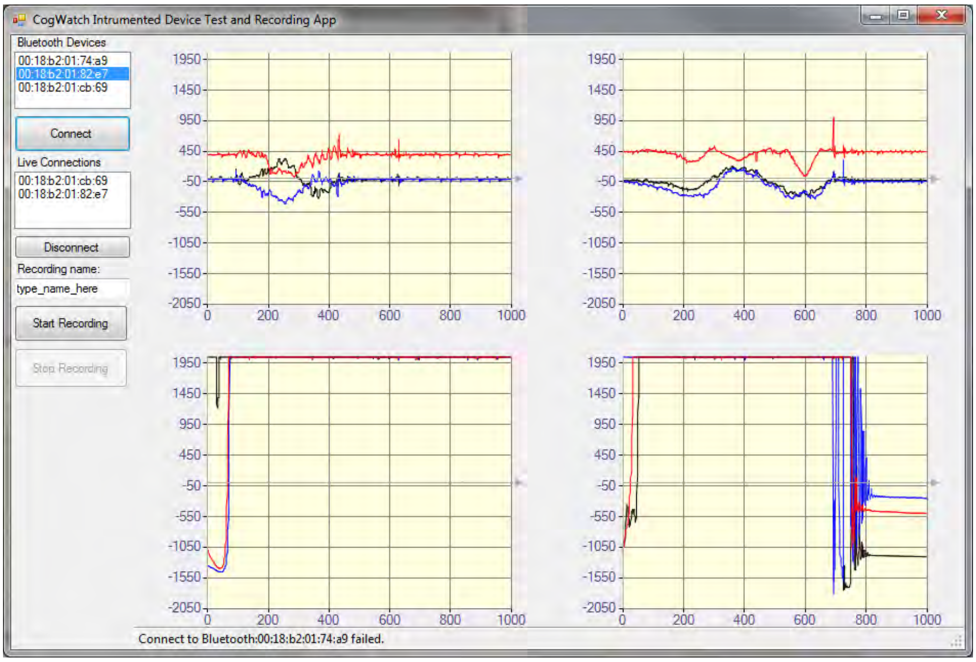


Figure 15 - Coaster test software with two connected coasters and their live data graphs.

3.1.2.2 Firmware and drivers description

Computer software has been written to accommodate all the needs in evaluating and testing the CogWatch coaster prototype. The software has multiple functions and features:

- The software manages the wireless connection. Bluetooth module on the coaster is configured to continuously listen for incoming connections; this allows the software to connect to and disconnect from the coaster devices without the need for the user to interact with the coasters.
- The software is able to connect to multiple coaster devices simultaneously, managing the data streams from each device with multi-threaded code.
- The software displays the sensor output of each connected device live on graphs, using separate graphs for accelerometer and force data for clarity. This allows the user to check that the sensors are operating correctly and perform visual analysis on live data.
- The software can record the data from each device to files on the computer. The sensor data is extracted from the byte data stream and then written to plain text comma separated value files. This data can be opened in excel for simple analysis as well as used by more complex analysis tools such as MathWorks MATLAB. Every sensor reading is time stamped so that the data from multiple devices can be synchronised. The time stamping makes sure that gaps in the data due to poor wireless connection are handled correctly.
- The software was written so that it is simple enough to use by non-technical users for collecting data for the evaluation of the prototype coaster. Making connections is as simple as selecting a device and clicking connect, making recordings involves

typing in a name and clicking start and stop record buttons. The errors and connection problems that can occur with wireless devices are handled automatically. Bluetooth pairing, which is required to make a Bluetooth device function with a computer is also handled automatically.

The software is written in the C# programming language with all of the threaded management of data streams and connections wrapped in simple functions and classes. This should make the code easy to adapt for integration into the rest of the system.

3.2 Feedback Devices

3.2.1 <u>Watch</u>

Figure 16 shows the digital watch finally selected for the first prototype from Meta Watch [12]:

Figure 16 - Watch for first prototype (Meta Watch) [12].

The watch, which is paired with the VTE monitor, can be charged, programmed, and debugged using the 4 pin Spy-Bi-Wire interface and clip (Figure 17). Simplicity of the methodology and the large capacity of the battery make this device suitable for the problematic that involves this case.

Figure 17 - Programming clip [12].

3.2.1.1 Hardware description

For this first prototype Meta Watch was selected to provide only a vibro-active feedback to warn the patient that an event is taking place and he or she must provide an answer. The principal reasons for choosing this device are related with the low consumption and could be explained by the hardware architecture.

First, the low power consumption of the device is obtained due to the internal architecture of it. The Meta Watch core a Processor Tracker™ that is a 16 bits RISC based processor designed for the low power microcontroller series of Texas Instruments™. The MSP430F5438A is composed by several modules for providing the different ports and features resumes in the following table.(Table 2).

Figure 18 describes the main components which compose the selected watch.

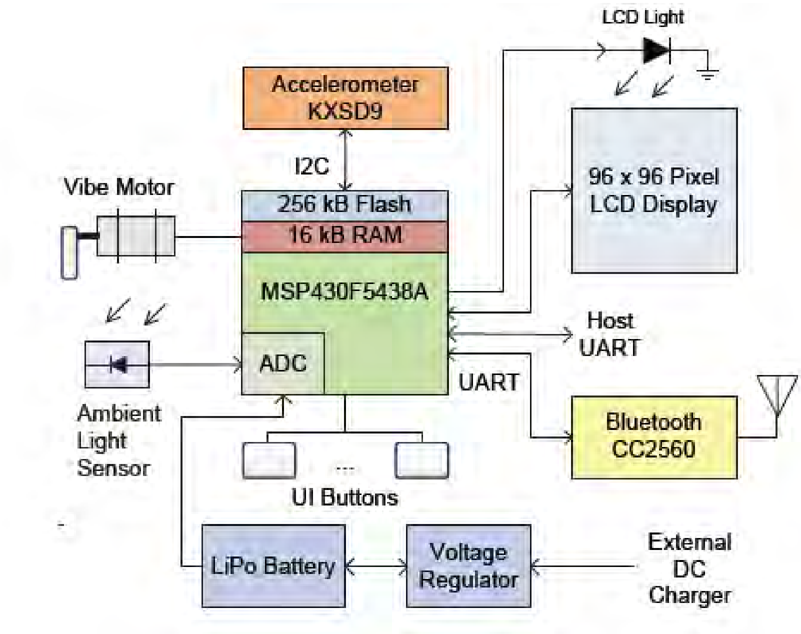


Figure 18 - Block diagram for Meta Watch components.

It has an internal clock that allows process from 8 to 24 MIPS. Furthermore, the digitally controlled oscillator allows wake-up from low-power modes to active mode in less than 5 µs. That means that the main delay in the activation of the clock for giving a vibroactive feedback to the patient will be located in the Bluetooth™ module.

For this and more information please refer to [13]

	MSP430F5438A
Frequency (MHz)	25
Flash (KB)	256
SRAM (B)	16384
GPIO	87
Timers - 16-bit	3
Watchdog	Yes
Real-Time Clock	Yes
Brown Out Reset	Yes
SVS	Yes
USCI_A (UART/LIN/IrDA/SPI)	4
USCI_B (I2C & SPI)	4
DMA	3
Multiplier	32x32
Temp Sensor	Yes
ADC	12-bit SAR
ADC Channels	16
Pin/Package	0DIESALE, 0WAFERSALE, 100LQFP, 113BGA MICROSTAR JUNIOR

Table 2 - Parametric features of MSP430F5438A [13].

Secondly, the digital display is a 96x96 pixel TFT LCD. This is the current type of display implemented in the watch. The LCD display is pixel monochrome. Each LCD buffer is arranged as 96 lines of 12 bytes. A block diagram of the pixel layout is shown in Figure 19. This device was selected for this prototype to ensure a long time between charges of the battery.

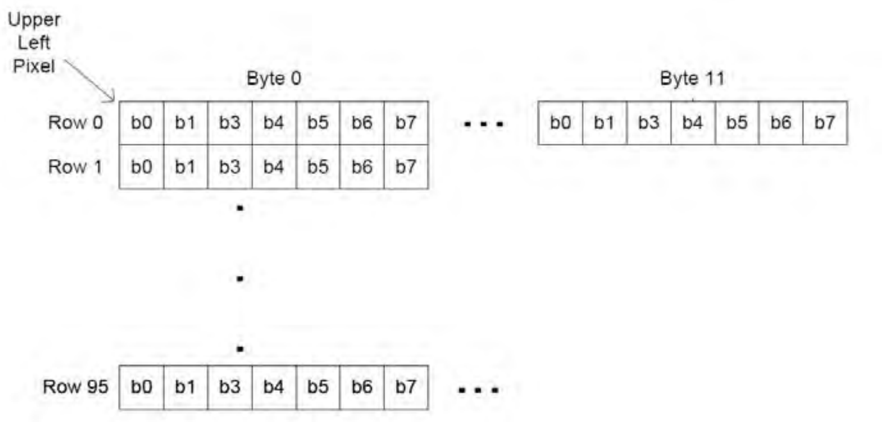


Figure 19 - LCD display pixel layout.[12]

Thirdly, an accelerometer could be embedded in the watch hardware. Motion events can be detected by setting it to wake up on orientation change, tap detection, or programming the accelerometer's motion thresholds. The usage of the accelerometer is highly application dependent. The amount of power used by the accelerometer would exceed the rest of the watch if it were left on continuously. For that reason, accelerometer is not included in this first prototype.

Fourthly, the battery state can be handled automatically. For Meta Watch, the full state of the battery is achieved in, approximately, 4 hours and it lasts between 4 and 5 days. A micro USB socket with a kind of clip is usually implemented which allow to plug the watch into a USB hub or any 5V USB power source to recharge.

Common batteries of 3.7V 75mAH lithium ion coin cell can be used.

Fifthly, the watch contains a vibration motor as its function is the main goal of the device. Its default use is to notify the user different kind of events, low battery, and to indicate that the Bluetooth link is lost. This will be used to achieve the goal of CogWatch focused on haptic feedback.

Finally, the communication between the watch and the VTE monitor is over the Bluetooth radio CC2560 also from Texas Instruments (Figure 20). CC2560 turns out to be a Bluetooth Host Controller Interface (HCI) solution which enables easy design as well as short time to market for Bluetooth applications in medical, industrial and consumer electronics applications. In addition, it supports Bluetooth 2.1. For this and more information please refer to [14]

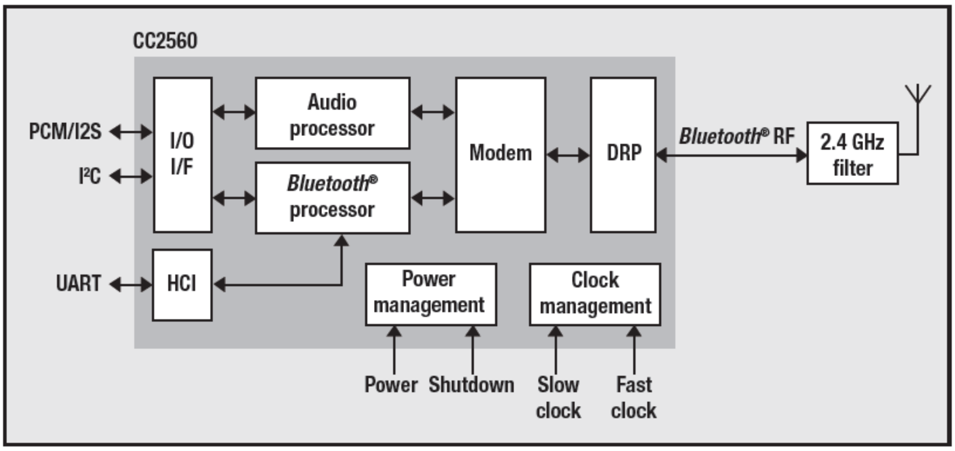


Figure 20 - Functional diagram of CC2560 [14].

.

Table 3 shows the main features of this module:

Parameter	Value	Condition/notes
Power supply voltage	1.7 to 4.8 V	Battery or DC to DC
Operating ambient temperature range	-40 to 85C	Industrial temperature range
Output power	+12 dBm	GFSK, typical
Receiver sensitivity	-95 dBm	GFSK, typical, dirty Tx on
Shut-down current	1 µA	Typical
Deep sleep current	40 µA	Typical
Ultra-low-power scan	135 µA	1.28-second interval
EDR full throughput	39.2 mA	Tx = 3-DH1, Rx = 3-DH5
eSCO	8.3 mA	2-EV3 64 Kbps, no retransmission

Table 3 - Technical specifications of CC2560 [14].

Apart from an advanced power management for extended battery life and ease of design, the following advantages can be highlighted:

- On-chip power management, including direct connection to battery or DC to DC.
- Low power consumption for active, standby and scan Bluetooth modes.
- Proprietary low-power scan algorithm achieves page and inquiry scans at one-third the normal power.
- Shut-down and sleep modes to minimize power consumption when Bluetooth is not used.

In addition. the module features a flexible clock management interface with support for:

- Automatic fast-clock detection mechanism.
- Frequency adjustment to offset and drift.

3.2.1.2 Operating system, firmware and programming design

This point has been decisive in the selection of the watch because Meta Watch offers an open-source solution that let the user programs and communicates with the watch by using different languages.

The operating system FreeRTOS [15] is a small preemptive RTOS commonly used in embedded applications. Communication between tasks is via message queues. There is a system buffer pool to support alloc / free of messages. Events are also sent as messages. Each task has an input message queue.

The FreeRTOS structure is composed by different tasks to be considered. Firstly, an Application Task provides a way to customize functionality of the watch while executing a task during the rehabilitation but it also provides functionality when disconnected from the VTE monitor. It will be possible for the watch functions to be controlled completely through messages from the VTE monitor.

Secondly, a Display Task is responsible for managing updates to the LCD display. Updates may be in response to a message from the Application Task or an internal event such as time update.

Thirdly, a Background Task handles the system control and status functions, such as ambient light measurement, battery charging, button status and monitoring/control of the vibration motor.

Fourthly, events and inter-task communication are performed using messages. A buffer pool is supplied to assist in managing message flow. Tasks are responsible for allocating a buffer when sending a message and freeing (or forwarding) a message placed on their queue.

Fifthly, the Idle Task is responsible for power management of the microprocessor. When no other tasks need to run, the processor is placed in a low power sleep mode.

Finally, Bluetooth communication is managed with a pair of tasks. Most of the communication does not require the involvement of the other tasks. Messages are sent to the VTE monitor by placing them on the Bluetooth transmit queue. Messages received from the VTE monitor are placed on the queue of the task(s) that has registered for them.

The Application Task is responsible for handling some Bluetooth events, such as pairing and (re)connection. Bluetooth API calls are provided to assist in managing the link.

DEFAULT WATCH MODES

The watch supports three basic running modes to add some structure to the interaction with the VTE monitor. The modes are Idle, Notification and Application Mode, each mode has its own state machine, options, and display buffer. Idle Mode is the most basic mode and any timeouts will ultimately return here. Notification mode can interrupt either Idle mode or Application mode, but when Notification Mode is exited, the watch will return to the last active mode (Figure 21).

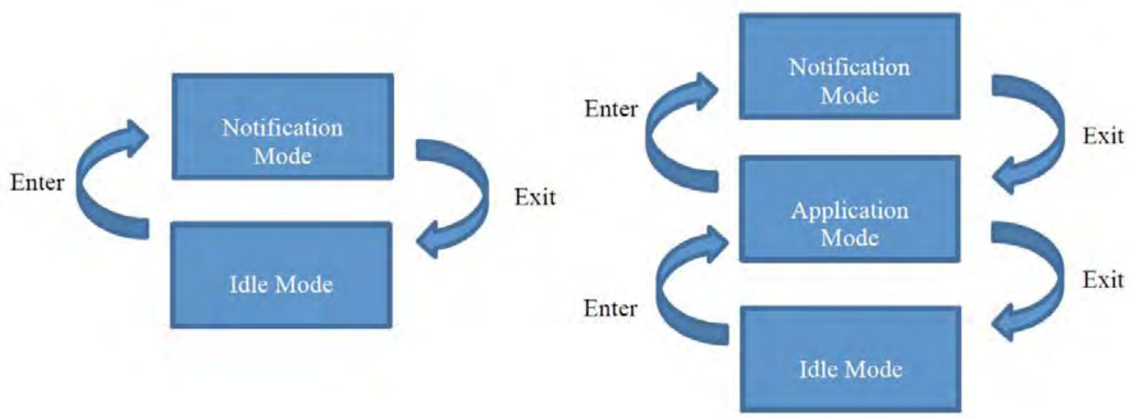


Figure 21 - Watch modes.

First of all, Idle mode is the default mode of the watch. The digital watch displays the Idle mode buffer constantly. The analog watch displays are OFF during Idle mode. To conserve power the watch should strive to be in Idle mode as often as possible and the phone should update Idle mode buffers only the minimum needed to keep the display information up to date.
By default, the digital display for Idle mode is shared with a watch controlled system area that keeps time and date. This operation can be changed so the entire display is available for the remote application. By default the watch reserves all button input for built-in user interface functions.

Secondly, Notification mode is for important transient notifications. You can think of notification mode as interrupting the current mode for a short time and then dropping back again, like an SMS alert, or a Tweet. Notification mode can interrupt both Idle mode and Application mode. Notification mode is explicitly entered and exited using a protocol message. Notification Mode will timeout after a period of inactivity. The mode that was active before the interruption is the mode the watch will return to. The entire screen is available during this mode.

Finally, Application mode is a mode used when control over the screen and buttons is required for an extended period of time. This “dumb terminal” mode can be used for an application to tailor a specific type of interface. Application mode is explicitly entered and exited using a protocol message. During Application mode, the watch will periodically send a keep-alive message that must be responded to or Application modes will time out.

The next list resumes the situations where the watch vibration is required see Figure 22. This means that the Meta Watch system changes from idle to notification mode.

- New session is coming and the system is calling the patient.
- Semantic feedback information, informing about an error.
- Semantic feedback information, informing about a danger.
- A message is displayed in the screen and waits for interactions.

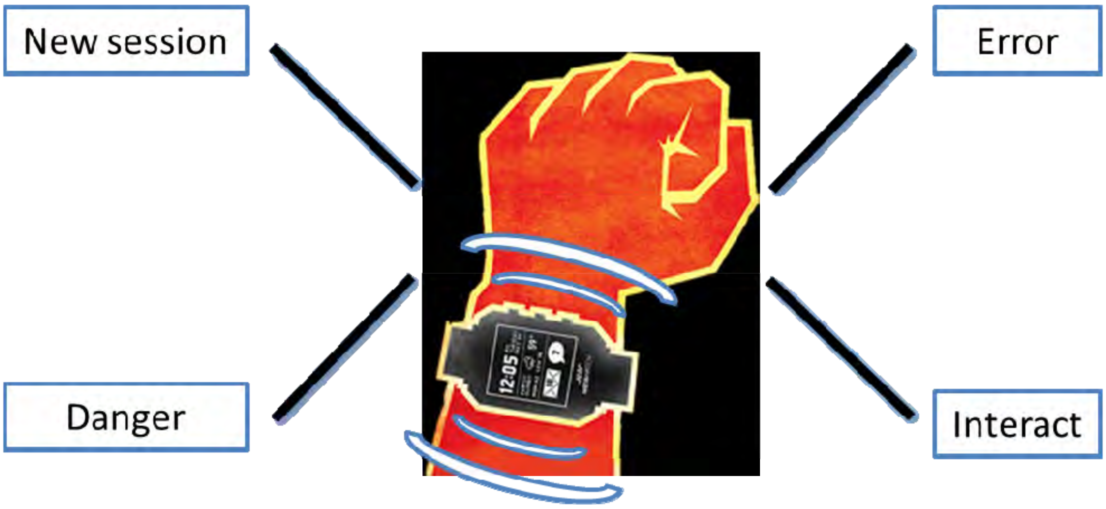


Figure 22. Meta Watch- interactive cases.

REMOTE PROTOCOL MESSAGE

The Meta Watch Remote Protocol is a lightweight, bi-directional message based command protocol that can be used to remotely configure and control the watch. These protocol messages are sent via the Bluetooth SPP wireless link.

The protocol has a standard packet format. Packets are based on a 32 byte maximum message length (Figure 23):

Start	Length	MsgType	Options	Data	CRC
B_0	$B_1 = N$	B_2	B_3	$B_{3+1} \ldots B_{N-2}$	B_{N-1}, B_N

Figure 23 - Packet format for remote messages.

- Start: ASCII <SOF> = 0x01.
- Length: Total number of bytes in the packet including start and CRC.
- Type: Packet type, typically a command.
- Options: Additional information about the type or command.
- Data: Packet data, 0 to 26 bytes.
- Check: CRC-CCITT All packet bytes (up to the CRC).

- The CRC matches the settings in the MSP430 it is CRC-CCITT (0xFFFF) with reverse input bit order.
- CRC({0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39}) = 0x89F6.

FIRMWARE

The watch firmware is a program that was compiled for the MSP430 microprocessor in the watch and saved as a file. Typically this file contains data in a binary format, but it could also be encoded in ASCII format. Common extensions for these files are .d43, .txt .or .hex.
At the moment, the latest version of the Meta Watch digital watch firmware is DigitalWatch_SPP_V3_1_0.d43.

SOFTWARE DESCRIPTION FOR COGWATCH

As mentioned several times along this document, the software designed is used in order to make the watch vibrates.

The programming environment chosen is Microsoft Visual Studio and the program is composed by the following main files:

- Meta Watch1.vcxproj:

  This is the main project file for VC++ projects generated. It contains information about the version of Visual C++ that generated the file, and information about the platforms, configurations, and project features selected.

- Meta Watch1.vcxproj.filters:

  This is the filters file for VC++ projects generated. It contains information about the association between the files in the project and the filters. This association is used in the IDE to show grouping of files with similar extensions under a specific node (for e.g. ".cpp" files are associated with the "Source Files" filter).

- Meta Watch1.h:

  This is the main header file for the application. It includes other project specific headers (including Resource.h) and declares the CMeta Watch1App application class.

- Meta Watch1.cpp:

  This is the main application source file that contains the application class CMeta Watch1App.

As other features, the application includes support to use ActiveX controls.

Finally, other standard files are:

- StdAfx.h, StdAfx.cpp:

  These files are used to build a precompiled header (PCH) file named Meta Watch1.pch and a precompiled types file named StdAfx.obj.

- Resource.h:

  This is the standard header file, which defines new resource IDs. Microsoft Visual C++ reads and updates this file.

3.2.2 VTE Monitor

VTE monitor is in charge of:

- Displaying a simple simulation and/or figures of the action which patient has to execute after an error is detected or in case of ignorance.
- Verbal cues for complementing the simulation/figure.
- Text messages and sounds for warning.
- User interface for starting/stop the application/simulation and for selecting the task.
- Collecting all the corresponding data in the processor.

3.2.2.1 Hardware description

The system description includes a home processor that can assess different displays.

The selected resource is an All-in-one computer, because it is composed by a screen and a CPU. It can be an advantage because it doesn't need too many cables, so it is easier to be installed at home.

VTE monitors are good devices and easy to be installed. They have tactile screen which is quite good and useful because the patient may have an interface and he/she might choose the task that he/she wants to do in order to start and stop the application.

This VTE will act as the central unit but it is open for future versions additional screens and Kinect™ sensors.

Table 4 resumes the hardware specification of the ASUS All-in-one PC used for the development of the first prototype:

VTE specifications

Display	23.6"(59.9cm), 16:9, Wide Screen, Full HD 1920x1080, LED-backlight
Touch Screen	Multi Touch
CPU	Intel® Core™ i5-2400S (6M Cache, 2.50 GHz, Turbo boost up to 3.3GHz)
Chipset	Intel® H61
Graphic	NVIDIA GT 540M(1GB)
Memory	2 GB Up to 8 GB at 1333MHz 2 x SO-DIMM
Storage	Up to 2TB 6Gb/s (7200rpmRPM)
Wireless Data Network	802.11 a/b/g/n

LAN	10/100/1000Mbps
Camera	1.3 M Pixel
Audio	Sonic Master
	DTS Surround Sensation UltraPC™
Speaker	2 x 3 W
Dimensions	590 x 461 x 60 ~230 mm (WxHxD)

Table 4 - All-in-one PC ASUS ET2210ENTS specifications [16].

3.2.2.2 Software description

For covering all the features previously described effectiveness it has been designed an architecture according with the necessities of the patient.

The application will react to the patient demands and will interact with suggestions and cues that will be related with what situation the system assumes with tha current interpretation of the data provided by sensors.

All the internal processes like the prediction and action recognition, data acquisition and storage must be in some way show in the graphical interface. The patients must be capable of free moving and the interpretation and association with the virtual environment must be clear on close to real time.

Usability and motivational factors seem to be an important reason for the success of computer-based rehabilitation [17].

It is important to emphasize the global complexity of the system add mixed an expert system for manipulating all the specific information of the patient that the clinician gives to the system with all the new information learned by the algorithms.

The following figure, Figure 24, explains the main functionalities of the VTE:

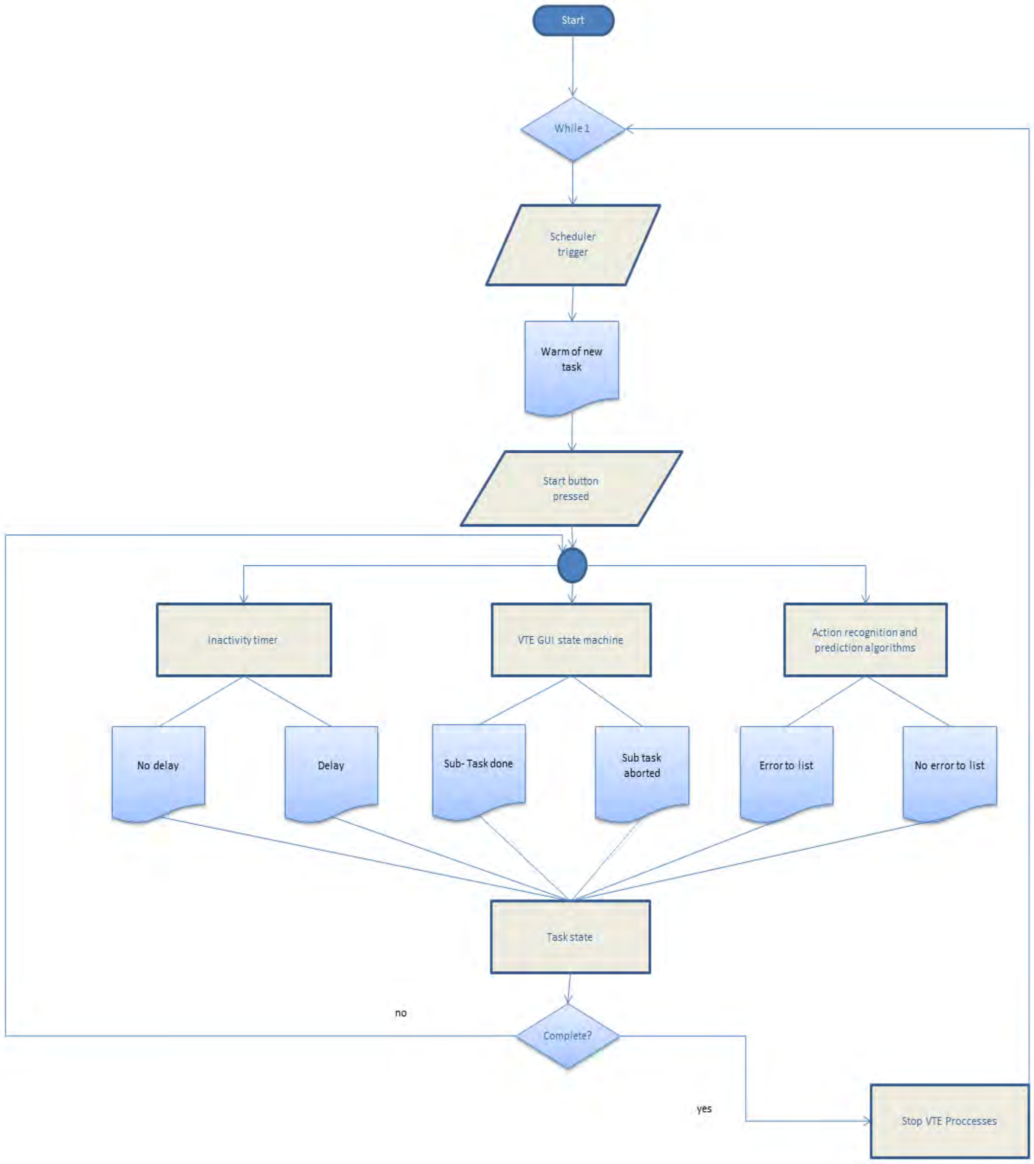


Figure 24 - VTE General functions diagram.

The first thing to consider is that the unique way to start the application is to press the start button in the user interface. The System can warm the patient that a new session is coming.

That is the main function of the scheduler and a warning will be displayed in the VTE monitor to that purpose but it will never start unless the patient touches the start button. By this way is ensured that the patient agrees to start a new session.

Then some activities will be launched in parallel as different modules and classes. Three processes will be activated in this case.

First, the Inactivity timer will calculate the time that no interaction of the patient with the objects is taking into account. This information is useful for the system to know when is required to ask any help to the patient.

Second, the VTE GUI module is in charge of displaying images, voice messages and simulations depending of the ability of the patient to advance along the different steps of the task that he is doing. This state machine is explained in the following section.

Third, algorithms for prediction and action recognition are called and fill an error table where the last input corresponds to the current error that the patient could be committing.

All this modules interacts and share information through the Information Handle module.

Once every sub-task is complete the modules are reset and launched again until the task is finished and in that case or when a fatal error is committed by the patient or when the patient subsequently fails to perform the task and is asked to take a break, the VTE GUI will show a stop button to confirm that the session is over. Then, all the video recorded and data are stored in the local data base and the Information Handle is called for closing the session.

VTE GUI

This is a module of the general architecture that interacts with the patient and the clinician if present.

This Module (Figure 25) is a C# graphic user interface based in Windows forms that is always running to enable the clinician to program new sessions or delete them.

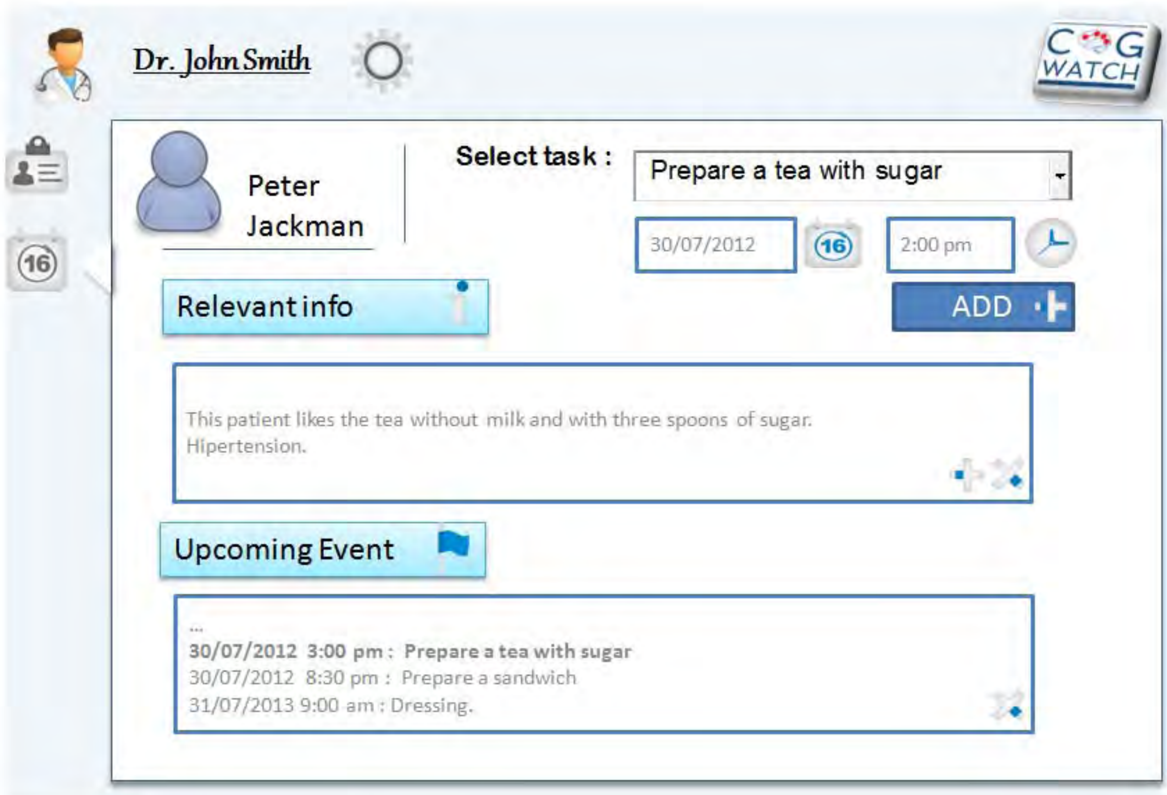


Figure 25 - GUI interface for task scheduling.

Nevertheless, the main function of this module is the assistance of the patient while the session is running. There is a protocol to follow in the assistance of the patients.

Figure 26 explains using a state machine representation the procedure during the beginning of session of the patient.

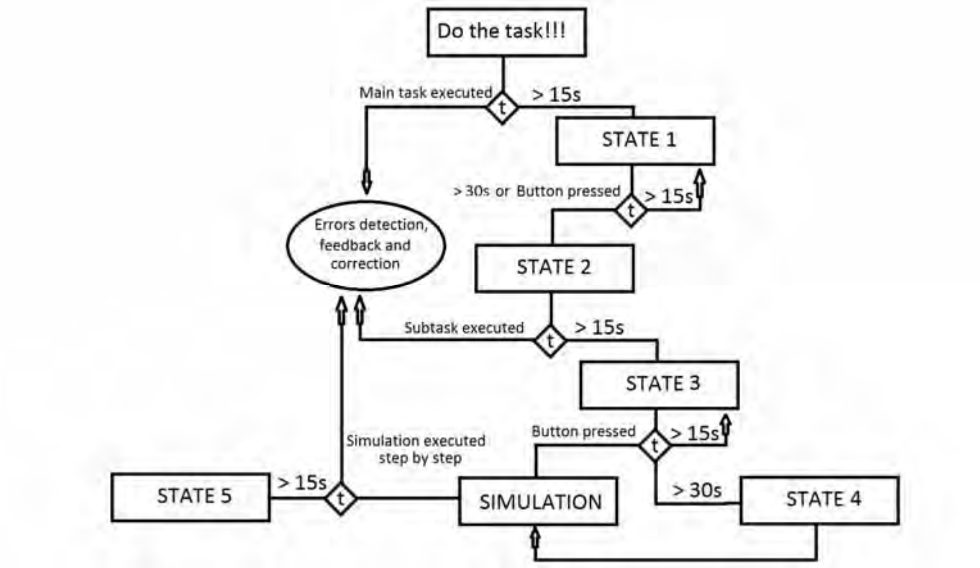


Figure 26 - The procedure during a session.

Firstly, the patient is asked by the clinician to execute the main task of preparing tea. In case patient is exhibiting difficulty with proceeding with the task, for example, after 15 seconds of inactivity, it will be necessary to go down along the second level of the tree (subtasks) and STATE 1 will show a help button to be pressed and will talk to the patient by verbal message: "Please, touch the help button if you need help".

If the patient does not touch the help button on the screen after 15 seconds, the system will repeat the prompt. After the repetition, if the patient doesn't touch the screen within other 15 seconds then the system will move to STATE 2. If patient touches the screen, the system will directly move to STATE 2.

When the state of the system is set STATE 2, this interface will display an image of the corresponding subtask in the current level of the tree (e.g., heat water) and the verbal message: "Heat water".

If the patient shows inactivity after 15 seconds, then the system will move to STATE 3 and the screen will play an auditory message to the patient: "Please, touch the image if you need a help with the task". If the participant touches the image on the screen, the VTE will display a simulation of the relevant subtask, i.e. heat water. If the patient does not touch the image within 15 seconds, the system will repeat again the prompt.

If the participant fails to touch the image within 15 seconds more, the system will be in STATE 4 and will display the verbal message: "Please, follow the cue on the monitor", followed by the simulation (75% of the original simulation speed) of the actions that comprise a subtask.

Thirdly, after watching the simulation, the system will wait 15 seconds for an action to be initiated. If the patient fails to initiate any movement in that time period then the system should prompt via VTE verbal and text message: "Let's take a break and try again in 5 minutes" STATE 5 and the session will start again from STATE 4 when the START button is pressed.

Each time the patient remains blocked, without executing any action/movement, and the system ask for help is called patient assistance request.

If the patient manages to execute the action instead of taking a break the system will supervise the performance as usual.

GRAPHICAL CONTENTS

- Images: the first cue provided to the patient is an image. These images are predefined and will be stored in a local directory in ".bmp" format. All this images will be combined with an auditory message.
- Auditory messages: the auditory messages will be stored in a local directory with a ".wav" extension. Initially the messages will be available in three languages English, German and Spanish but the system will be configurable to allow the inclusion of other languages to adjust to patients´ mother tongue.

SIMULATION

The simulation allows the patient to interact in a non-immersive virtual reality that differs from a simple video in the way that the patient can more easily relate the real world with the elements that appears in the screen.

Some of the advantages of a non-immersive virtual reality for post stroke rehabilitation are mentioned in [17] and are listed below:

- Engaging/motivation
- Repetitive intensive
- Adaptable to patient education
- Usable in chronic phase
- Activities of daily living

The strongest connection between the real world and the virtual world will be given by the emplacement in the simulation of the objects through the use of the information from all the available data to emulate in the screen the real set up that the patient is actually facing.

The different objects that take part in the selected task for the first prototype must be modelled to provide the homologous set up in a non-immersive virtual reality environment.

The required models are:

Objects:

- Kettle (to boil water).
- Cup (in which to make the tea).
- Tea bags (initially 3 tea bags).
- Brick of milk.
- Bowl (to contain sugar cubes and the additional teaspoon).
- Jug of water.
- Spoon (to stir the tea).
- Boxes/Containers (one to contain new tea bags and one to contain used tea bags).

Manipulators:

- Right and left Hand (Use for first perspective vision).

Ingredients:

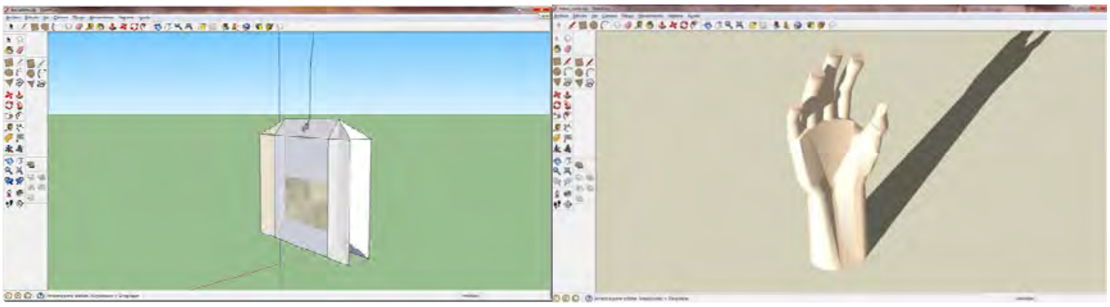
- Tea bags.
- Sugar cubes.
- Water.
- Milk.

The models could be classified as active or passive as some of the objects don't change their state and, so that, their correspondent model doesn´t has any animation, and some others have complete changes in their morphology and consequently increases its complexity.

According with the kind of model different formats where chosen in relation with their complexity grade.

For simple models it was chosen COLLADA (COLLAborative Design Activity) and is a universal format designed to prevent incompatibility accepted by several modelling software. It was created by Sony Computer Entertainment in collaboration with Khronos Group. These models have ".dae" file extension. In this project the models were created under Google© Sketchup and special textures were added with Blender.

Figure 27 shows the COLLADA model of a hand designed in Google© Sketchup. This kind of models has no skeleton and no animation. They can only be translated, rotated or resized in order to animate them:



(a) (b)

Figure 27 - COLLADA model of the (a) tea bag (b) right hand.

For complex models it was chosen the X file format. This models have a ".x" file extension and were introduced firstly with Direct X 2.0. It is an efficient 3D model mesh file format because of how easy it is to load into programs and games. It allows the inclusion of skeletons in modells and predefined segmented animations. It can be readed with the Direct X viewer from the version 6.0 and it is suitable for supplying realistic animations. The program used for the inclusion of skeletons and animations to the active models was fragMOTION that is distribute curently under version 1.2.0., it is a powerful 3D modeler used for the creation and animation of characters.

Next Figure 28 shows the process of associting vertexof the mesh to the different bones of the squeleton assigned to the right hand model:

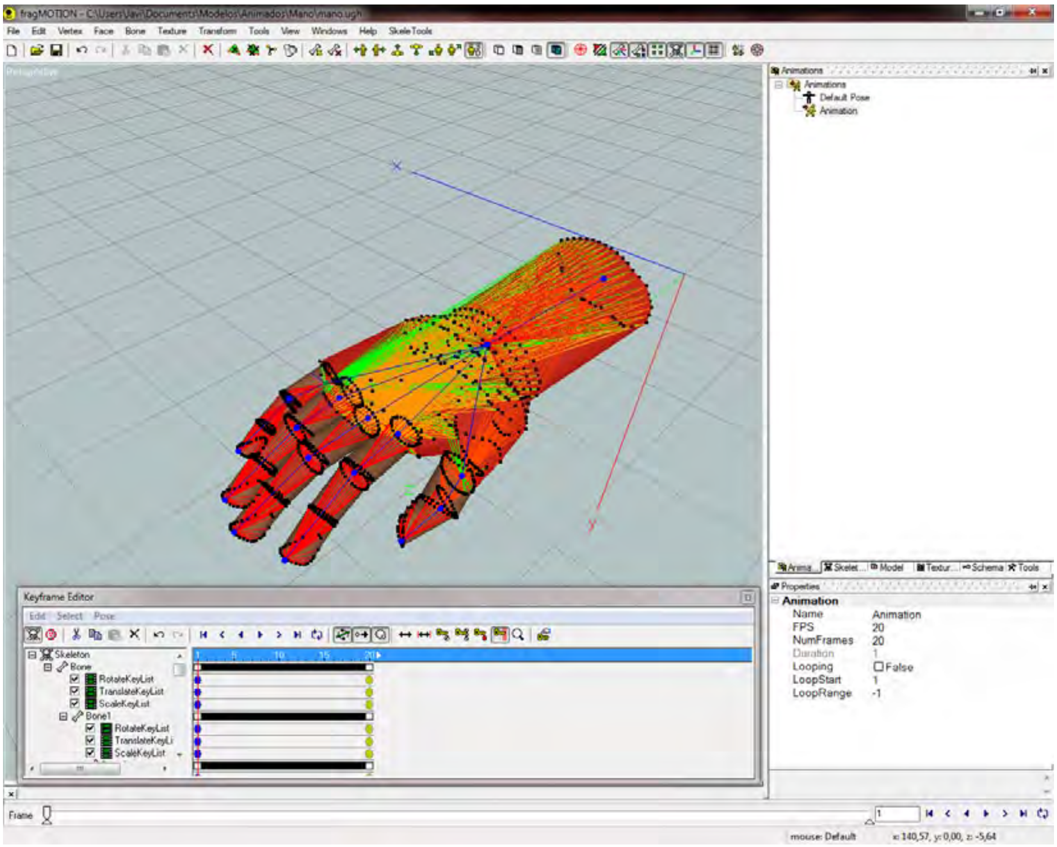


Figure 28 - X file animated model of the right hand.

Regarding the render, for 3D features a program based on direct x rendering has been selected.

The direct x API was selected due to their following characteristics:

- Performance:
  - Less CPU cycles, stalls and bandwidth cost.
  - Reduce validation overhead, in-GPU multipass.
  - Render to an entire cubemap or volume
- Consistency:
  - Strictly defined behaviour across chipsets.
  - Visual Quality and material management.
- Visual effects
  - Multisample antialiasing.
- More expressive programmability, geometry shader, texture:
  - Compression formats, resource views.

The use of ActiveX technology allows to create C Sharp applications and drop the DX Studio Player in as a component. Figure 29 shows this control in their design view:

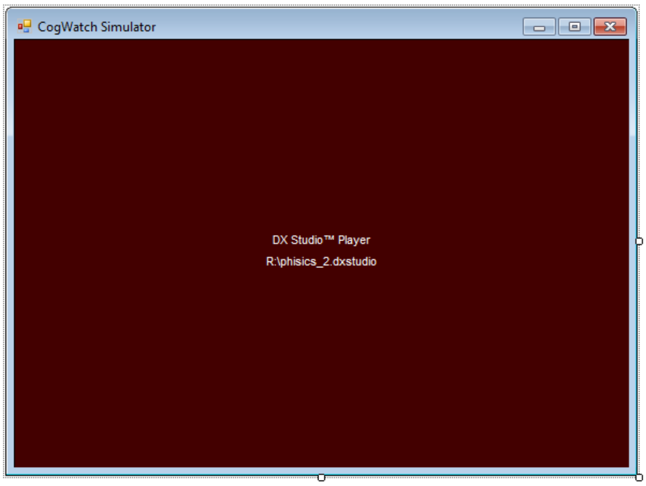


Figure 29 - Simulator player control.

A complete interactive document can be compiled into a single redistributable executable file. DX Studio Player is a free tool that could be incorporate to the main installation easily as the developers facilitate tools to integrate it and check for updates of the most current version of direct X controls.

This Player allows the interaction of the Information Handle Module and VTE GUI with the current simulation and the data for objects location.

DX Studio projects use Javascript code for driving the simulation and easily interact with the code.

The files produced by DX Studio use standard XML to describe the entire scenes. The files also contain all the resources needed to display the 3D world, compressed into the same file using standard ZIP compatible algorithms. A security option allows this data to be encrypted if necessary.

4. CONCLUSIONS

CogWatch system will provide a personalized, long-term and continuous cognitive rehabilitation for stroke patients with Apraxia and Action Disorganisation Syndrome (AADS).

This deliverable has described the features of the devices involved and to be considered in the first prototype of CogWatch in terms of design, components, software and implementation. Deep and technical descriptions have been provided for each component.

The delivery of the first prototype is coming soon and this new step has been focused on settling down the final architecture of the prototype in terms of the components the patient will interact with. The devices patients will use for rehabilitation basically will consist of two groups: for monitoring and for feedback. Monitoring devices are in charge of monitoring the execution of the task and movements of the patient in order to be able to detect possible errors and malfunctioning. Meanwhile, feedback devices are in charge of providing the corresponding cues and feedback to make the patient aware of the errors committed and possible risks.

With the architecture and components propose, it is expected that the system will work properly and effective in order to provide rehabilitation in the task considered and focused on preparation of tea.

The components described in the document comprise the subsystem called CogWatch client, which corresponds to the patient side and installed at his/her house. Considering the subsystem focused on the server, it is described in the deliverable D2.3.1 Report on networks I and complements to this current deliverable D2.2.1 Report on devices I.

Although sensorized shirt is out of the range of this first prototype, a summary of the current state of its development has been mentioned in order to show the continuous work from RGB Medical Devices.

Finally, it is essential to mention that the functionalities of both, the watch and sensorized objects, will be extended for future prototypes. For example, a new version of the watch will be analyzed in order to try to incorporate tactile interface like a touchable big button. Meanwhile, sensorized objects will be complemented by RFID tags located in the bracelet of the watch in order to improve the accuracy and effectiveness in detecting errors while grasping and manipulating objects. For this first prototype, the simplest version of the devices has been considered in order to move step by step to more advanced solutions.

REFERENCES

1. Amft, O. and Tröster, G., 2009, On-body sensing solutions for automatic dietary monitoring, *Pervasive Computing, April-June,* 62-70.
2. G. Alankus, R. Proffitt, C. Kelleher, J. Engsberg. Stroke therapy through motion-based games: A case study, Proc. of ACM ASSETS'10.
3. Yao-Jen Chang, Li-Der Chou, Tsen-Yung Wang, Shu-Fang Chen. A Kinect™-based Vocational Task Prompting System for Individuals with Cognitive Impairments, Personal and Ubiquitous Computing, DOI: 10.1007/s00779-011-0498-6.
4. Yao-Jen Chang, Shu-Fang Chen, Jun-Da Huang. A Kinect™-based system for physical rehabilitation: a pilot study for young adults with motor disabilities, Research in Developmental Disabilities, Vol. 32, Iss. 6, 2011. pp. 2566-2570.
5. Technical (imagen JPEG, 600 × 283 píxeles) - Scaled (0%) [Internet]. [cited 2012 sep 20]. http://www.ros.org/wiki/Kinect™_calibration/technical?action=AttachFile&do=get&targe =ros_Kinect™.jpg
6. Freedman B., Shpunt A., Machline M., Arieli Y. Depth Mapping Using Projected Patterns. May 13, 2010. U.S. Patent 2010/0118123.
7. R. M. Wham, «Three-Dimensional Kinematic Analysis Using the Xbox Kinect™», University of Tennessee Honors Thesis Projects, may 2012.
8. <<ProgrammingGuide_Kinect™SDK>>[Online].Available:http://202.120.53.119/wordpress/wp-content/uploads/2011/06/ProgrammingGuide_Kinect™SDK.pdf .[Accessed: 23-oct-2012].
9. Cogollor José M., Hughes Charmayne, Ferre Manuel, Rojo Javier, Hermsdörfer Joachim, Wing Alan, Campo Sandra, "Handmade task tracking applied to cognitive rehabilitation", Special Issue "New Trends towards Automatic Vehicle Control and Perception Systems", MPDI Sensors magazine, 2012.
10. Ogris, G., Lukowicz, P., Stiefmeier, T. and Tröster, G., 2012, Continuous activity recognition in a maintenance scenario: combining motion sensors and ultrasonic hands tracking,
11. Parekh, M. and Baber, C., 2010, Tool use as gesture: new challenges for maintenance and rehabilitation, *British Computer Society Human-Computer Interaction,* http://ewic.bcs.org/upload/pdf/ewic_hci10_paper27.pdf
12. «Meta Watch - Home». [Online]. Available: http://www.Meta Watch.org/. [Accessed: 5-oct-2012].
13. «Texas Instruments - Products». [Online]. Available: http://www.ti.com/product/msp430f5438A. [Accessed: 5-oct-2012].

14. «Texas Instruments - Products». [Online]. Available: http://www.ti.com/product/cc2560-pan1325. [Accessed: 5-oct-2012].
15. «FreeRTOS - Home». [Online]. Available: http://www.freertos.org/. [Accessed: 5-oct-2012].
16. ASUS - PC Todo en uno- ASUS ET2210ENTS [Internet]. [cited 2012 oct 15]. Available from: http://www.asus.es/Allinone_PCs/22_inch/ET2210ENTS/#specifications
17. Man DWK. Common Issues of Virtual Reality in Neuro-Rehabilitation. [cited 2012 oct 15]; Available from: http://cdn.intechopen.com/pdfs/13660/InTech-Common_issues_of_virtual_reality_in_neuro_rehabilitation.pdf