Medication Dispenser

Nuno Santos, student at Instituto Superior Técnico

Abstract — The average life expectancy of human beings is increasing, resulting in an aging population. Associated with aging, there is often loss of cognitive capabilities and memory problems, which can have a major impact on quality of life and autonomy of people required to comply with a therapeutic plan.

This paper describes the development of a prototype of a medication dispenser, which aims to combat the lack of adherence to medication regimens, especially for the elderly and people with cognitive limitations.

The features of several products on the market that address this issue were analyzed, leading to the identification of several restrictions and limitations. Based on this analysis, a list of required features and the overall architecture of the solution to be developed were defined.

One of the main requirements was to create an open solution that focuses on communication with other devices (smartphones, computers and tablets) and the connection to the Internet, allowing it to be adapted to different contexts of use.

Given the emphasis in the openness and remote interaction capacity, a programming API has also been created, allowing third parties to develop and optimize applications to interact with the dispenser in specific contexts, and allowing the automation of configuration procedures (e.g. defining the therapy plan) and the collection of events (e.g., failed doses and access attempts in hours not allowed).

The prototype developed fulfilled the proposed requirements, culminating in a modular, versatile and affordable solution, which implements the planned features.

Keywords — Medication dispenser, Therapeutic management, Assistive technologies, Embedded systems.

I. INTRODUCTION

The average life expectancy of human beings is increasing, resulting in an aging population. Associated with aging, there is often a loss of cognitive capabilities and memory problems. The quantity of medication that each person takes daily is also increasing, making therapeutic plans more complicated.

As a consequence of these problems, older people tend to forget to take their medication or take it wrongly. This can lead to the inefficacy of the therapeutics, undesired drug interaction and, in some cases, even death.

In order to avoid those scenarios, the typical solution is to have someone (either a familiar or a contracted professional) controlling the person’s compliance to the medication. But, as result of the actual socio-economic context, this is not always possible. Besides that, some people don’t want to lose their independence and have someone constantly monitoring them.

The main goal of this work is to develop a prototype of a smart medication dispenser that helps older people to manage their medication, in order to avoid errors in following their prescribed treatment.

This device, previously configured by a caregiver, should warn the user when it’s time to take his medication and dispense only the required pills.

The solution should be an open system, which can be used by other parties to create their own applications that interact with the dispenser. This allows it to be used in multiple contexts, with optimized configuration interfaces. These applications should be able to run in different platforms, such as smartphones, computers and tablets, allowing the automation of configuration and information gathering processes.

In the next section, we describe and analyze a product existing in the market that already tries to solve the problem presented. In section III, we present our solution, with some details about the possible contexts of use, its functionalities and the main electronic modules integrated in the prototype. In section IV, we describe the most relevant implementation details and in section V we present some results about the prototype created. Finally, in section VI, we present the conclusions of this work.

II. STATE OF THE ART

To solve the problem presented, there are already some products in the market, which were analyzed.

One interesting solution is the MedSmart MD2, from e-pill [1]. This dispenser, shown in figure 1, has a circular structure, divided in 29 compartments, which is rotated in order to make the pills of a given dose available. The pills can then be accessed in an opening located in the external plastic structure. In order for the dispenser to detect that a dose was retrieved, the user should tilt the device to get the medication [1].

![Fig. 1. MedSmart MD2 equipment. From [2].](image)

This dispenser allows for a maximum of six alarms of...
medication per day, which will repeat through the days of operation [1]. This can be a limitation if the person needs to take more than six doses per day or if the therapeutic plan is not regular.

At a dose time, the dispenser moves the circular structure to the corresponding compartment and signals the event with a flashing light and a sequence of beeps. The compartment remains accessible until the user takes the medication or the next alarm fires. In the last case, that dose is considered failed [1]. This means that there is no proper control of the maximum allowed time for the person to take a dose. In the limit, a dose can be taken right before the next one, which may lead to medication interactions.

This dispenser has a feature called “early dose” that allows the user to anticipate one dose, which is very pertinent if the person needs to go somewhere. The early dose is ordered in a button that can be mechanically protected to disable that function [1].

The interface used to configure the dispenser is composed by five buttons and a display [1]. Although the functionalities of MedSmart are not very complex, this interface is not the most practical to configure the device, especially when it comes to input numeric fields.

In order for the caregiver to monitor the compliance of the user to the medication, the dispense time of the doses and the failures are recorded. The history can be accessed through the same interface used for configuration [1].

To complement the described functionalities, there is a plus version of MedSmart that allows the remote monitoring of the device. These functionalities include remote notifications of failed doses, the crossing of a pre-defined threshold of remaining pills and the download of the compliance history. The notifications can be sent to the caregiver by email, SMS or phone call [3].

The plus version of MedSmart costs 789.95 USD and the standard version costs 489.95 USD [4].

Although the MedSmart dispenser was the most relevant solution found, other interesting products were analyzed, namely the Phillips Medication Dispenser and the MedFolio Electronic Pillbox. To complement the research, medication management applications were also analyzed, with emphasis on the app MedHelper.

III. PROPOSED SOLUTION

1) Functionalities

All commercial products analyzed are closed systems, optimized for a specific context, with low versatility and limited configuration options.

To solve these problems, we intend to develop an open platform that can communicate with applications running on devices such as tablets or smartphones, allowing to create better configuration interfaces and to automate procedures.

To allow the dispenser to fit the necessities of each user, some of the limitations observed in commercial products will be eliminated, allowing for a more flexible configuration of dispense times.

The dispenser will also give the caregiver the possibility of remotely monitor relevant events, such as knowing if the user takes the medication on time.

Some of the proposed functionalities are listed next.

Medication dispensing

- Possibility of defining 28 alarms for medication dispenses. For each alarm, it is possible to define the firing date and time, without limiting the number of alarms per day.
- Possibility of defining a maximum tolerance time for the user to retrieve the medication. After that time, if not dispensed, the dose is considered failed.
- Possibility of dispensing a dose earlier. The period before the dispense time can be configured independently for each alarm.
- Possibility of configuring the alarms using a simplified mode (called daily mode) where the caregiver just needs to specify the firing time of the alarms for a day. The configured alarms will repeat every day, making it adequate for regular regimens.
- Possibility of managing PRN medication (medication taken only when needed), by defining a minimum safety interval between doses.

User interface

- Medication dispensing is done just by pressing a button.
- Alarms are signaled by a flashing light and a beeping sound. The intensity and the “on/off” pattern of both visual and audio signals can be configured accordingly to the user’s preferences.

Textual warnings

- Possibility of defining a maximum of 30 textual warnings, shown in the dispenser’s display. This allows remembering the user of medical procedures (e.g. measuring blood pressure), medical appointments or other relevant subjects.

Information recording

- Recording pertinent information of operation, such as dispense time of medications or attempts of getting dispenses outside the configured time.

Interaction with the dispenser

- Bluetooth support, allowing communication with a smartphone, tablet or computer.
- Wi-Fi support, allowing communication through the Internet. This allows the remote configuration of the dispenser and sending notifications to the caregiver.
- GSM support, allowing the sending of remote notifications through SMS and the possibility to make voice calls to/from the dispenser.
- Possibility of direct interaction with the dispenser, using its keypad and display, to configure it and to access registered information.
Application development support

- Creation of a generic API that simplifies the interaction of external devices with the dispenser, allowing its configuration and download of recorded information.

2) Usage scenarios

One of the main objectives was to develop an open solution that could be integrated in a broader system and capable of operating in multiple contexts.

In a typical scenario the dispenser manages the medication of a person, whose caregiver may be a family member. In this case, since it’s not expected that the caregiver will have specific formation, it is possible to develop an application for a smartphone or tablet to guide him in the configuration process. If the caregiver doesn’t feel comfortable with this type of technology, the dispenser also offers the possibility of configuring it directly, using its keypad and display.

Another application scenario is the use of the dispenser integrated in a caregiving network. In this case, it is expected that one person (or a small team) will manage multiple units. So, it is possible to develop more sophisticated applications to automate the process of configuring and getting information from the dispenser. Figure 2 illustrates a possible architecture.

![Fig. 2. Architecture that allows the automation of the configuration procedure of the dispenser](image)

Besides this, it is possible for the caregivers to remotely monitor some events, such as if the user has taken or not the medication on time. The remote notifications can be sent by the dispenser through the Internet (as exemplified in figure 3) and/or by SMS, using GSM.

![Fig. 3. Architecture that illustrates the exchange of data through the Internet.](image)

These are only some examples of different architectures that can be implemented using the dispenser. This is possible because the proposed solution is open, modular, and allows external applications, running in various devices, to communicate with the dispenser using a simple protocol.

3) Dispenser’s Architecture

In order to simplify the development process and to facilitate changes in the code, a modular architecture for the dispenser was developed, as shown in figure 4.

![Fig. 4. Dispenser’s architecture.](image)

The GSM, Bluetooth and Wi-Fi modules support the communication with the dispenser and implement the associated protocols.

The display and the keypad are the devices that the caregiver can use to directly interact with the dispenser. The pressure buttons are the way the user interacts with the system and the signalization of alarms is done using a generic buzzer and a LED.

The dispensing mechanism will be similar to the MedSmart approach. It will use a motor to move a circular structure, divided in compartments. Since the creation of the mechanical structure of the dispenser is outside the scope of this project, only a rudimentary prototype of this feature will be created, in order to validate the implementation.

The recording of information will be implemented using non-volatile memory (EEPROM). Since a correct timekeeping is fundamental for the application, the dispenser will use a Real-Time Clock.

All these modules are controlled by a central entity, called “processing module”, which will be implemented using an adequate microcontroller.

4) Selection of modules

In order to be able to develop the proposed solution in the time available, it was decided to use, whenever possible, functional modules that have all the required electronics to work, needing only to be integrated in the solution.

It’s important to refer that the choices made were limited by the available equipment (for example, it was not possible to use SMD components due to the lack of specialized tools).

Although significant care was taken in choosing modules with a low price, the main selection priority was their
functionality.

**GSM module**

Since there are many GSM modules in the market that allow communication with a microcontroller, the decision was made based on the price.

The module chosen consists of a board based on the SIM900 device, from SIMCom [5], which can be bought for 20 euros.

The SIM900 can operate in the main GSM frequencies (850/900/1800/1900 MHz). It supports I2C, SPI and serial communication and has an analog audio interface [5].

This device can operate with a power supply between 3.2 V and 4.8 V and consumes 22 mA while in idle mode (registered in the cellular network and ready to communicate) [5].

**Wi-Fi module**

To implement the Wi-Fi communication, we decided to use a module called ESP-01, which is based on the device ESP8266 from Espressif Systems [6]. This is one of the cheapest modules available on the market, costing only about 2.5 USD.

The ESP8266 integrates a 32 bits RISC CPU and supports I2C, SPI and serial communication. This device implements 802.11 b, g and n standards, supports Wi-Fi direct and integrates the full TCP/IP stack [6].

The ESP-01 module operates at 3.3 V and consumes about 70 mA during normal operation. During transmission, it can have a peak consumption of 300 mA [7].

**Bluetooth module**

To implement the Bluetooth communication, one of the requirements was the support for SPP (Serial Port Profile), which allows the exchange of data through an emulated serial connection [8].

From the multiple modules available that fulfill the requirement, the cheapest and simplest one found was HC-06, from Guangzhou HC Information Technology [9].

This module implements Bluetooth V2.0 and works only as a slave. It can be directly connected to the UART of a microcontroller, allowing for data rates up to 1382400 Baud.

HC-06 operates at 3.3 V and has a consumption of 30 to 40 mA during the pairing process and of 8 mA during communication [9]. It already has a built-in antenna and has a maximum range of 10 m [10], which is enough since it is expected that the caregiver will be in the same room as the dispenser when using this communication mode.

**RTC module**

To implement the RTC module, the preferred solution was the DS3231 from Maxim Integrated [11]. There are multiple modules based on this device costing only around 1 euro.

We chose this device because it uses an internal oscillator with temperature compensation, which offers a precision of ±2 minutes per year [11]. Besides that, it can be accessed through I2C [11], which is supported by the majority of microcontrollers.

This device operates internally at 3.3 V but the power source can be between 2.3 and 5.5 V, with an average current consumption of 300 µA. This device can also operate using a backup battery, which allows it to maintain the timekeeping functions if the main power supply fails [11].

The RTC will be used integrated in a module that also contains an AT24C32 device, which is a 4 KB EEPROM from Atmel, also accessible through I2C [12].

**Motor**

In order to implement the dispensing mechanism, a step motor was selected. This type of device can be controlled with great precision using digital pulses, allowing it to work without a feedback mechanism [13].

Since the structure that holds the pills is divided in 29 compartments (28 for medication and one empty starting compartment), it was required that the motor had a precision below 0.1º.

From the solutions found, we decided to use the 28BYJ-48, which is a very cheap motor that operates at 5 V and has a precision of about 0.088º per step [14].

Although steppers can operate in open loop, we decided to include a feedback mechanism in order to avoid the accumulation of errors and to allow the synchronization with a known position, for example, after a power failure occurs. The guarantee that the correct compartment is dispensed is mandatory due to the nature of this application.

We decided to implement the feedback mechanism with an extrusion in the circular structure, which marks a reference compartment. This extrusion will be detected by an optocoupler (model ITR9608 from Everlight [15]).

**Display and keypad**

In order to implement the direct interface with the caregiver, a display and a keypad will be used.

To facilitate the navigation through the menus and the input of numeric fields, we decided to use a generic 4x4 membrane keypad. This allows the use of ten digit keys, four navigation buttons and “OK” and “CANCEL” buttons.

The display chosen is an eighty characters LCD, divided in four rows. This was considered enough to display the messages for textual warnings and alarms, and also to implement the configuration menus.

The model used (QC2004A) has retro-illumination, operates at 5 V and has a current consumption of 1.6 mA (with the retro-illumination disabled) [16]. Since this LCD needs eight data lines for control, a module based on the integrated circuit PCF8574 was used, in order to convert this interface to I2C [17] and allow controlling the display with only two data lines.

**Buzzer**

To produce the sound alarm, a generic passive buzzer was
selected to allow controlling the frequency of the sound. To control the volume, a digital potentiometer was used. The model chosen was the MCP41010, a 256 positions 10 KΩ potentiometer, controlled through SPI [18].

**Processing module**

Since there is a great variety of microcontrollers with different specifications, an estimation of the resources needed was made based on the peripheral modules described early and on the initial planning done for the software application to be developed.

In order to communicate with all the modules, the need for supporting I2C, SPI and Serial communication (three UARTS) was identified.

In order to implement all the functions proposed, we estimated that not less than 4 KB of RAM, about 4 KB of EEPROM and around 100 KB of program memory were needed.

Since the communication protocols (GSM, Bluetooth and Wi-Fi) are implemented by the electronic modules themselves, the most critical functions to implement are the ones related to the serial communication with the modules and the interaction with the user or caregiver, which should be perceived as instantaneous.

So, a microcontroller capable of executing around ten millions of instructions per second (MIPS) was considered enough. Since many RISC microcontrollers can achieve almost one MIPS per MHz, an execution speed of about 10 MHz was considered adequate.

In order to allow a faster development, we decided to use a microcontroller development board, with all the main electronic components needed to work built in (power supply, programming interface and pins connected to accessible plugs). Besides that, the microcontroller should have stable development tools and good documentation and support from the community.

From the solutions analyzed regarding the main microcontroller families currently used (AVR and PIC), we considered that the Arduino platform offered the best choice, due to the quantity of documentation available, the simple development environment and the support from the community [19], which offers many libraries that make the interaction with peripherals very simple [20].

From all the boards available in the Arduino family, the Arduino Mega 2560 is the one that fulfills all the requirements defined and offers some tolerance margin in the resources available.

This board is based on Atmel’s ATmega2560 microcontroller. It has an 8 bit RISC architecture, operates at 16 MHz and has 4 KB of EEPROM, 8 KB of SRAM and 256 KB of FLASH program memory. It has support for SPI, I2C and has 4 UARTs. It also has 54 digital IO pins (of which 15 provide Pulse Width Modulation) and 16 analog input pins [21].

5) **Other components**

In order to offer a robust solution, the dispenser will have two power sources, one from a 9 V transformer connected to the power grid, and a 6 V backup battery (with a capacity of 1.2 Ah). The dispenser automatically changes to the battery if the main power supply fails.

Since the circuit has modules operating at 5 V and 3.3 V, the voltage from the power supplies is converted to those levels by two distinct voltage regulators, as shown in figure 5.

![Dispenser’s power supply circuit.](image)

To allow turning off the LCD, Bluetooth and Wi-Fi modules, to save power if they are not in use, there is a MOSFET in their supply circuit. The MOSFET used can be controlled by an IO pin of the microcontroller, allowing to control if current is or is not flowing to the modules. This method can’t be applied to the GSM module used because its power supply state is controlled by a pressure button. Nevertheless, this could be solved by a custom electronic design, instead of using a pre-assembled GSM module.

6) **Mechanical structure**

In order to demonstrate the working principle of the dispenser and the feasibility of the approach taken, a prototype of the mechanical structure of the equipment was created using a 3D printer. The result is shown in figure 6.

![Mechanical structure of the dispenser.](image)
generic and scalable solution, with modular functionalities, to simplify the reutilization and modification of the code in a possible future transition to a product.

2) Date and Hour

Keeping a precise track of the elapsed time is very important for the dispenser. An initial approach used an internal timer of the microcontroller, which was programmed to fire at every second, to execute the time updating algorithm. But, in consequence of the inaccuracy of the quartz crystal circuit, this method showed a high degree of imprecision, with the clock getting late approximately half a minute per day.

To solve this problem, the real-time clock described early, which has a high precision, was used to synchronize the internal representation of the time and date in the microcontroller. To avoid a constant access to the RTC, this procedure is executed once in every thirty minutes intervals, allowing keeping the error below an unnoticeable threshold.

3) Medication Dispensing

There are two distinct operation modes for the dispenser. In the first mode, called “configuration mode”, the caregiver can configure all the dispensing alarms and PRN medication (medication taken only when needed). The second mode, called “operation mode”, is where the alarms fire and the medication can be dispensed.

Although all major configurations should be done in the configuration mode, there is the possibility to perform small corrections in the operation mode, such as changing the firing hour or date of an alarm.

Dispensing alarms

Although the implementation of the dispensing alarms may seem simple, there are numerous details and conflicting situations that need to be taken into account, making the process relatively complex.

Every alarm has a firing time and date, which is compared with the timestamp of the dispenser at every minute. If they match, the alarm will fire.

In order to avoid an excessive consumption of RAM, all the alarms are stored in the EEPROM (ordered by their timestamp). But, to avoid accessing this memory every minute to check the firing time, the next alarm scheduled to fire is maintained in RAM.

In order to give more flexibility to the dispenser, each alarm can have multiple compartments. So, when it fires, the user should press the dispense button as many times as there are compartments. In EEPROM, the multiple compartments of an alarm are represented as adjacent entries with the same timestamp. As shown in figure 7, the number of compartments for an alarm is calculated when it is loaded from EEPROM to RAM.

When an alarm is loaded to RAM, if it has an early dose time defined, the timestamp for the early dose is calculated and, for each minute, it is verified against the dispenser’s timestamp. When they match, if there is no alarm active, it is signaled in the display and can be dispensed if the user presses the dispense button. This feature is compatible with the definition of multiple compartments for the alarm, although it makes the management more complex.

After an alarm fires, its tolerance time starts being monitored and if it elapses without the dispensing being made, that dose is considered failed.

PRN medication

To manage PRN medication, the dispenser allows the definition of a minimum safety interval between doses. In order to distinguish an early dose from a PRN dispensing, a dedicated button for PRN medication had to be added, which also makes sense since it is a special request.

Monitoring of unauthorized dispensations

The unauthorized attempts to access regular and PRN medication, pressing the correspondent buttons, are monitored and recorded. This allows inferring if the user has experienced some state of confusion and is trying to take more medication than he should.

4) Dispensing mechanism

As stated early, the dispenser has a feedback mechanism, implemented with an optocoupler. This allows to, unequivocally, align with a reference compartment, from any arbitrary position.

In order to allow more complex dispensing functions, without imposing that the dispensed compartments are adjacent, it is possible to go from any compartment to any other. This allows managing both regular and PRN medication with the same mechanism.

Since the motor can move in both directions (clockwise and counter-clockwise), this feature is used to optimize the movement, which is always done using the shortest path. Whenever this movement implies passing through the reference compartment, a synchronization process is executed.

5) Communication with the dispenser

The possibility of multiple devices being able to communicate with the dispenser (using GSM, Bluetooth and Wi-Fi) is a major factor of innovation and distinction from existent solutions.
Since the communication with Wi-Fi and Bluetooth modules is done using the UARTs of the microcontroller, a simple and compact protocol was developed, which allows to send configuration commands to the dispenser and to retrieve recorded information.

**Bluetooth communication**

Bluetooth is used to establish communication with nearby devices, such as tablets and smartphones. The Bluetooth module emulates a serial communication and allows those devices to send commands to the dispenser, using the structure defined in the mentioned protocol.

**Wi-Fi communication**

Although the ESP-01 comes with a firmware that allows it to be controlled by sending AT commands to its serial port, the preliminary tests performed showed that it was not enough robust. In order to solve this problem, a personalized application was developed using the ESP8266 Arduino based IDE [22].

In this application, the module works simultaneously as a client, in order to send remote notification of events, and as a server, in order to be able to receive configuration commands, using the protocol described next.

The communication is performed using sockets.

**GSM communication**

The GSM module has a built-in firmware that allows its control using AT commands [23], sent to its serial port.

This interface is used by the dispenser to send textual messages (SMS) to the caregiver about events that occurred during operation. In this case, the data is sent to the module in a human readable format, using the AT commands’ structure. This allows the reception of the notifications in any cellphone, even if it is not a smartphone.

Although the reception of SMS was successfully tested in the dispenser, we considered that this mode was not suitable to allow the configuration of the dispenser since the caregiver would have to type all the text with the correct structure.

**Commands**

As stated before, the communication with the dispenser through Wi-Fi and Bluetooth is done using structured messages, called commands.

As shown in figure 8, those commands are composed of two delimiting bytes, an operation code, a checksum to verify if errors have occurred and the data.

To allow sending the delimiting bytes as part of the data, it is possible to use the byte 253 as an escape byte.

Every command sent to the dispenser has a response message, not only to return data (when it applies), but also to guarantee that the execution was successful or to inform that a failure occurred.

This protocol is generic and very modular, with a separation between the algorithm that handles the structure of the message and the handlers that use the data, making it very easy to port to different applications.

**Remote notifications**

In order for the caregiver to monitor the operation of the dispenser and to avoid a constant polling for events, the device generates remote notifications about important events, with emphasis to the ones related with the regime compliance.

These notifications, which are optional, can be configured to be sent through the internet (Wi-Fi module) and/or by SMS (GSM module).

In order to avoid the loss of information, there is a circular buffer to accumulate notifications whose delivery failed, allowing a later re-transmission.

**Application programming interface**

In order to facilitate the development of applications that interact with the dispenser, three different APIs were developed: one for Bluetooth communication for Android (in Java), one for Serial and Bluetooth communication from a PC (also in Java), and one for communication over the Internet (in Python).

These APIs hide the protocol and low level communication details from the programmer, giving him simple methods (such as “read alarm” or “set clock”) that only receive the data needed for configuration as arguments and return the data read from the dispenser. Using the Android API, we developed a testing application that allows configuring and reading data from the dispenser, as shown in figure 9.

**Voice calls**

Besides allowing sending textual notifications, the GSM module also allows making calls from and to the dispenser, giving a direct way for the caregiver to interact with the user.

The dispenser can be configured do automatically answer a call, or answer only when a button is pressed (GSM button). To start a call, the user just has to press this same button and
the dispenser will automatically start a call to a pre-defined destination number. This is a very easy way of having voice calls, even if the user is not familiarized with a phone or has difficulties related with cognitive limitations.

6) Security

Although the implementation of sophisticated security mechanisms was outside the scope of this project, some simple solutions were implemented in order to mitigate some security issues detected.

To prevent unauthorized people to change the dispenser configurations and accidental changes by the user, the direct interaction with the dispenser (through the keypad and the display) can be protected with a numeric PIN. In order to access the configuration menus, a PIN code needs to be inserted first. If the code is wrong more than a pre-defined number of times, input will be blocked for a certain amount of time.

To avoid the same problem using Bluetooth, this communication channel can also be protected with a safety code (which can be configured to a maximum of 32 bytes in size). If this feature is active, the communicating application needs to send the correct code before sending any other command. Once the right code is sent, the channel remains open for a pre-defined amount of time and the application can send the commands. In order to avoid brute force attacks, a maximum amount of tries can be configured and, if that threshold is crossed, the possibility of sending the code is blocked for a configurable amount of time.

Finally, to prevent unauthorized people from calling to the dispenser through the GSM interface, a list of three allowed phone numbers can be defined. If this feature is active, calls from any other number are automatically rejected.

7) Energy management

Since it is possible to turn on and off some electronic modules (Bluetooth, Wi-Fi and LCD), this option can be used to reduce the energy consumption of the dispenser.

Besides that, a more sophisticated approach was taken, which allows to configure the dispenser to automatically turn off those modules (and also the LED and Buzzer alarm signaling) when a failure of the main power source is detected, in order to extend the battery autonomy. The caregiver can change which of them continue operating and which of them are turned off. When the main power source is restored, the disconnected modules are automatically reconnected.

V. ANALYSIS OF RESULTS

1) Tests

In order to validate the developed solution, all the functionalities implemented were tested.

To test the firing of alarms and textual warnings, multiple utilization scenarios were simulated, showing that the solution operates correctly.

The synchronization scheme with the RTC applied to the timekeeping algorithm demonstrated a high degree of precision, allowing all the events to occur when configured by the caregiver.

The dispensing mechanism has also shown to work with a very high degree of precision, always moving to the correct compartment of medication. After multiple dispenses, the synchronization mechanism guaranteed that no error accumulated.

To test the robustness of the communication using Bluetooth, a total of 20000 “read time” commands were sent to the dispenser, in order to check how many would fail. All the messages were sent successfully and the responses were correctly returned without any data corruption. Next, a total of 2000 “read configurations” messages were sent, interleaved with unstructured messages of random size (between 1 and 60 bytes) with randomly generated content. Again, all the messages were delivered successfully and the random data sent did not interfere.

The same tests were performed for Wi-Fi communication. In this case, for the first test, ten connections couldn’t be made and six returned no data in the response. Although the actual cause of those failures could not be found, they are probably due to the fact that the dispenser was considerably far from the router, which could cause some interferences or problems in the signal propagation. Nevertheless, for 20000 messages, and taking into account the price of the module, the results are very good. For the second test, no problems occurred during the transmission and all the response messages were returned successfully.

Those results show that the Wi-Fi and Bluetooth modules operate with a high degree of confidence, and the serial protocol created to structure the data (and the corresponding algorithm) is very robust, even in the presence of meaningless bytes.

In order to test the communication time, some measurements were made for some commands, as shown in table I. The values presented are the average values from all the commands sent. The values obtained are very reasonable and show acceptable latency. Naturally, more complex commands take more time to execute than simpler commands. For example, configuring the dispenser using the diary mode takes some time because all the dates for the effective alarms need to be generated and then loaded to the EEPROM. Nevertheless, the time is still acceptable, being smaller than 2 seconds.
It is important to refer that a considerable amount of the power consumption results from the electronic modules used. Those modules include some components that could be removed (such as signaling LEDs) or that could be optimized. For example, the Arduino board has a USB controller (ATmega8U2) which draws about 13.5 mA [24] and that isn’t needed since this interface is not used in the prototype.

VI. CONCLUSION

The goal of this work was to develop a prototype of a smart medication dispenser to help older people manage their medication. Although there are some products in the market that try to solve this problem, they have some limitations, which we tried to overcome in the proposed solution.

The implementation of this solution involved an extensive work in areas such as embedded systems, electronics and programming. There was also an extensive research work to find adequate electronic modules to use, in order to develop a solution that can, in the future, be adapted to a product.

The development process was not easy, since the implementation of the proposed functionalities required a lot of effort to ensure proper working, to avoid undesirable interactions, to adapt the programming style to fit the limitations of the microcontroller resources and to interface with the electronic modules used, because some of them were poorly documented.

Nevertheless, the dispenser prototype was successfully developed, culminating in an open and versatile solution that fulfills the proposed objectives and can be used by other parties to develop compatible applications. Besides that, the main limitations found in the products analyzed were solved.

We hope that this work can give a positive contribute in helping people taking their medication correctly, improving their quality of life and giving more tranquility to those who care for them.

REFERENCES


