Integration of Multiple Sensor-based Wellness Systems for Supportive Housing Using Bluetooth Low Energy

Suseela Poomdla
Computer Science Department
California State University, Fullerton
Fullerton, The USA
suseelapoomdla@csu.fullerton.edu

Joel Anil John
Computer Science Department
California State University, Fullerton
Fullerton, The USA
joel.aniljohn@csu.fullerton.edu

Pavan Kumar Kandregula
Electrical and Computer Engineering Department
California State University, Fullerton
Fullerton, The USA
kandregula123@csu.fullerton.edu

Rayleene Murrieta
Electrical and Computer Engineering Department
California State University, Fullerton
Fullerton, The USA
rayleene.murrieta@csu.fullerton.edu

Kiran George
Electrical and Computer Engineering Department
California State University, Fullerton
Fullerton, The USA
kgeorge@fullerton.edu

Anand Panangadan
Computer Science Department
California State University, Fullerton
Fullerton, The USA
apanangadan@fullerton.edu

***Abstract*—Residents in permanent supportive housing, defined as long-term, community-based housing combined with supportive services, often have health conditions requiring daily medication and monitoring for disabling incidents inside their home. Sensor-based solutions for checking for immobile residents, and automatic pill dispensers to ensure the timely delivery of medications are being developed to support living with health conditions. This work describes how multiple such health monitoring devices can be coordinated from a single user interface with communication over Bluetooth Low Energy (BLE). Individual subsystems, including an automatic pill dispenser and a smart wellness check, are designed for use within a supportive housing apartment unit. An Arduino RP2040 microcontroller is used as each subsystem's central processing unit. The foundation of the communication between the different subsystems is the BLE protocol. BLE was chosen for its adaptability to heterogeneous devices and low power consumption. The BLE communication architecture links these devices, facilitating reliable data transfer and device-to-device coordination. An Android application running on a tablet PC acts as the central coordinator of all subsystems and provides a unified user interface. The application scans for BLE devices, connects to discovered subsystems, and displays real-time sensor information in a user-friendly format. Nearby peripheral devices are identified using BLE advertising and scanning. Bidirectional communication channels are created once linked, enabling the central device to communicate with the peripherals and receive data from them. This BLE architecture supports simultaneous communication with multiple peripherals. Different parameters of the BLE protocol are systematically varied; the parameters are the MTU size, time interval for BLE device scans, and the resulting number of GATT connections and disconnections under various time intervals. Fine tuning these parameters enables the identification of the optimal set of parameter values for reliable data transfer between the central user interface and individual subsystems with minimum latency and maximum energy conservation.**

***Keywords*—Healthcare, IoT, Arduino, Android, Pill dispenser, BLE, mobile application**

I. INTRODUCTION

Housing that is provided to persons who have experienced long-term homelessness and who have debilitating conditions require wraparound services for long-term viability. For instance, such housing often provide medical support so that residents can improve their health. Ideally, such support should monitor the intake of medicines and ensure that the resident is not experiencing a sudden health condition while in their house. Adherence to proper medication is vital for proper rehabilitation of patients. Bedell et al. [1] have found that the prescribed medication and the ones consumed have a discrepancy of nearly 76%. This further delays the health improvement of residents. Incorrect dosage and missed timings also increase the mortality rate [2].

With the increased flexibility of IoT sensors, its usage in the medical field has also increased. Digital health is the result of the latest advancements. Sensor-based monitoring has been used for patients who have mobility issues. In certain scenarios, the sensors, such as wearables, can be intrusive to the patient. In our work on using IoT technology in supportive housing, the sensors used are non-intrusive. The housing unit contains sensors, and mobility is detected based on movement, and does not require the resident to wear any sensor.

As part of digital health, our supportive housing unit also has automatic pill dispensers to support medication adherence. Automatic pill dispensers automatically dispense the pills on time and notify the users to consume the pill. The difference between our “smart” pill dispenser and existing systems in the market is the monitoring of pill intake, i.e., the system monitors whether the user has consumed the pill using sensor data.

In this work, we describe the design of an Android tablet-based user interface that connects all the IoT devices in the supportive housing unit. The Android tablet is the central device that connects all the sensors and provides a single user interface via a GUI-based application. Android Application connects to these sensors via Bluetooth. For low energy consumption, we are using Bluetooth low-energy sensors. Hence, Bluetooth Low Energy (BLE) API facilitates communication. The communication established is bi-directional. The Android application acts as the Central device in BLE terminology and communicates directly with end users. We also present details of how the performance of the system depends on BLE-related parameters such as the time to scan, MTU size, and the number of peripherals.

The contributions of this work are as follows. (1) we present the design of a Tablet PC-based interface that links different types of IoT devices specific to supportive housing, and (2) we describe how communication performance (specifically reliability of establishing a communication link) varies depending on device parameters.

The rest of the paper is arranged as follows. Section II reviews related work for medication adherence, Section III describes system architecture of the hardware, and Section IV describes the software design. We conclude in Section V.

II. Related Work

A "Smart Pill Box" using Arduino and GSM interfaces was introduced by Rajan *et al.* [3], prioritizing medication adherence. Al-Mahmud *et al.*[4] has developed a system that sends mail notifications regarding medications. A Med Tracker system was developed by Hayes *et al.* [5], giving an in-depth insight into medication adherence, but it is limited as it needs a Bluetooth computer. Schueller *et al.* [6] have shown that digital mental health for mobile apps has helped a specific segment. Research by Morita *et al.* [7] has concluded an increase in need and support for sensing technologies adapted to smart homes and mobile applications for digital health. Tsai *et al.*[8] has proposed and developed a prototype smart pillbox with neural recognition to recognize pills and connection to smartphone via Bluetooth. Othman and Ek [9] proposed and developed a smart pill dispenser with an infrared sensor, LCD display system, and connection to iPhone or Apple smartphone. Antoun *et al.*[10] made a prototype with rotating motors, and connection to phone via Bluetooth. The pillbox has containers that are controlled separately with its own LED.

III. System Architecture

A. Pill dispenser

The Arduino Nano RP (2040) microcontroller is the core of the built Pill Dispenser design (Fig.1). It is equipped with a built-in Bluetooth Low Energy module, making establishing a smooth connection to an Android application easier. We used a rechargeable battery as a power source to run the pill dispenser. Adherence to medication regimens is ensured by exact time monitoring using an external Adafruit DS3231 Precision RTC module. In order to allow users to create and maintain their prescription regimens, the dispenser is built with various user interface components, as illustrated in Fig. 1. These elements include push buttons and an LCD with an I2C serial interface. A pill container with seven slots is powered by a 28BYJ-48 stepper motor and the ULN2003 driver board, enabling automated pill dispensing at preset intervals. The Android app, the Smart Cup, and the Pill Dispenser are all able to communicate with each other over BLE.

An RFID RC522 Module receiver is added to improve the system's accuracy and safety by guaranteeing that the Smart Cup is positioned correctly before pills are delivered. It uses the SPI protocol to connect with the microcontroller. This innovation reduces the possibility of tablets spilling outside the cup by acting as a fail-safe.

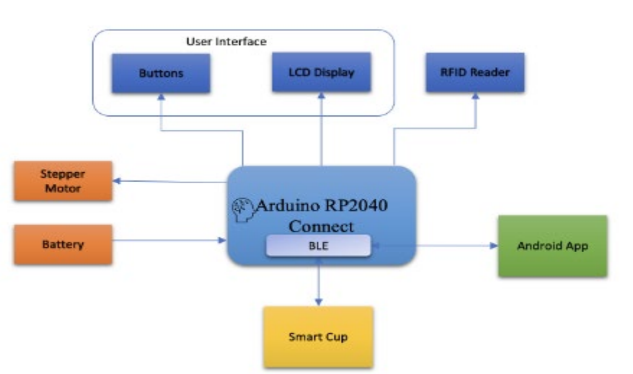

Fig. 1. Block diagram of Pill Dispenser.[11]

B. Smart Cup

The Smart Cup, another integral system component, utilizes the same Arduino Nano RP 2040 Connect microcontroller used for the Pill Dispenser (Fig.2). It leverages BLE technology to connect and synchronize with the Pill Dispenser, ensuring that pills are dispensed into the cup at the scheduled time. Moreover, it is powered by wireless charging since older adults may face difficulty changing the battery. To monitor the consumption of dispensed pills, the Smart Cup is equipped with an LSM6DSOX Inertial Measurement Unit (IMU), encompassing an accelerometer and gyroscope to detect the motion and angular momentum of the cup. Additionally, an ultrasonic sensor (HC-SR04) is employed to find the user's proximity after dispensing pills, verifying that the user has consumed the pills. This promotes medication adherence and enhances data accuracy for the Android Application.

This core system architecture is described more fully in [11].

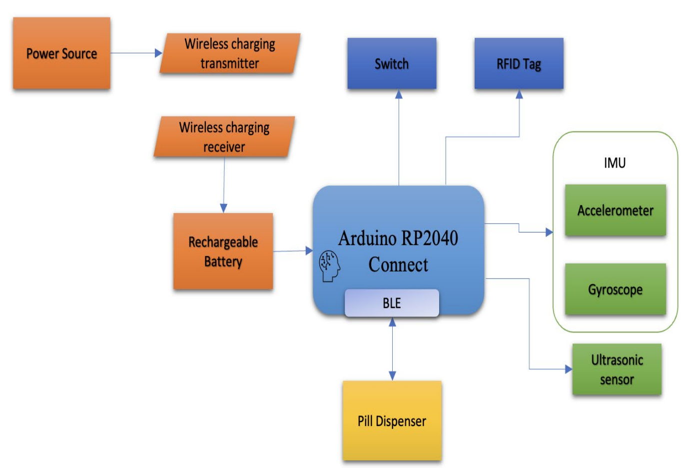

Fig. 2. Block diagram of Smart Cup [11]

IV. SOFTWARE DESIGN

The software part of the project consists of an Android application. The Android application connects to multiple peripherals. In this application, the peripherals are the sensors that are used to monitor the supportive housing units. In BLE terminology, the *client* is the device that searches for the sensors and initiates the connection. The *peripherals* are the devices that advertise that they have information but do not initiate connection. Generally, peripherals are the sensors. The App is central in BLE terminology, and the IoT devices/sensors are peripherals.

The Android app we developed has a tabbed structure. Each tab can connect and read data from one peripheral. So we have the Home Tab, Dashboard Tab, and Notification Tab. These tabs are the defaults from Android Framework. In this work, we describe using two tabs to connect to two peripherals. The Home tab connects to the pill dispenser and the Dashboard tab connects to the intelligent wellness and mobility sensors.

A. Android Permissions

The latest versions of Bluetooth have become more security-centric and user-centric. Multiple permissions are needed to access notifications and Bluetooth in newer API versions. The user permissions such as Location, Bluetooth, and Display notifications are requested and stored (Fig.3, Fig.4, and Fig.5). Permissions given once are stored permanently and are not requested again, even during app crashes. Permissions, once stored, are present till the app gets re-installed.

Fig. 3. Bluetooth Permission.

Fig. 4. Notification Permission

Fig. 5. Location Permission

Fig. 6. Turn ON Bluetooth

B. Android Bluetooth

As Bluetooth is the primary connection between apps and peripherals, the Android tablet's Bluetooth must be ON to establish connections. If the user does not turn ON Bluetooth manually, we turn the Bluetooth ON programmatically (Fig.6.). As we turn the Bluetooth ON, we programmatically request permission from the user to do that.

1) Home Tab:

The app opens and functions normally only after acquiring the required permissions. Once the app opens, the user can see three tabs: Home, Dashboard, and Notification, each one of them corresponding to one peripheral.

The Home Dashboard connects to the smart pill dispenser. The Android App and Smart Pill identify each other based on UUID. UUIDs are unique and are used to identify every service peripherals provided. The app uses two unique UUIDs for connection: characteristic UUID and service UUID. This set of UUIDs are used by Android App to connect to peripherals. Once the client knows UUID, the system can establish the connection.

The connection between the client and the peripheral is a two-step process. Once it is ready to dispense the pill, Smart Pill Dispenser starts advertising its connection details. The client (which is the Android app) is constantly scanning for new connections. Once the peripheral advertisement starts, the client picks up the peripheral in the following scan and establishes bi-directional communication.

As the connection is bi-directional, the system can send and receive data. For the smart pill dispenser, the app read the notifications sent from the cup and sends the days and times to the pill dispenser. Smart Pill dispenser can take input from the Android app for the days and the times the pill has to be dispensed. The message consists of days from Monday to Sunday with time in a 24-hour format. The user can select multiple days at different times (Fig.7). This data is collected cumulatively for all days and converted into a JSON file. The smart pill dispenser receives the data in JSON format.

"Launch smart scan" for smart pill dispenser and "stop scan" are the two buttons used to connect to the peripheral manually. Once the launch scan for the smart pill dispenser is selected, the scan starts to search and connect to the peripheral using its known UUID. We use a GATT connection to read the Smart Pill Dispenser data.

GATT connections with peripherals are a one-time read for every GATT connection. After the Smart Pill Dispenser starts advertising, the Android app uses its UUID to connect to it. We connect to the GATT server on the BLE device. Clients read the data advertised by the device and then disconnect the GATT connection through Bluetooth. The app reads data once the GATT connection is established. Hence, we re-establish the connection for a periodic interval, read the data, and disconnect it from the GATT server. Hence, once the launch scan is selected, the Android app scan connects and reads the data from Peripheral.

In the Home Tab (shown in Fig.8), the notification fragment section displays the data in the read message. The font of the notification fragment is large compared to others (Fig.9), so the information is much more visible for patients. Stop Scan for Smart Pill Dispenser stops the scan to search for the peripherals manually.

Fig. 7. Select Day and Time

Fig. 8. Home Tab without update

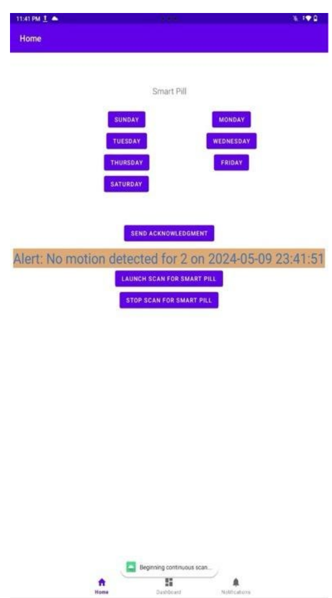

Fig. 9. Home Tab with update

Fig. 10. Dashboard with Data

2) Dashboard Tab:

The Dashboard Tab displays the connection between the Android App (Central Device) and Smart Wellness device (Peripheral). We first describe the Smart Wellness Scan and the use of its user interface. We need non-intrusive monitoring to check if a resident has been immobile for a long time.

Smart wellness is a nonintrusive solution where we monitor the supportive housing room, not the individual. If there is no movement, then an interface on the tablet PC will display a message that there is no movement for a certain period. The Android app uses Characteristic and Service UUID to connect to Smart Wellness Peripheral. The Dashboard Tab has two options: a one-time scan Button and a Periodic Scan (Start Background Scan and Stop Background Scan button). The difference between the two scans is that the one-time scan starts the Scan, connects to device advertising, reads the data, and stops the Scan. It is completely a one-time occurrence.

Initiating a Periodic Scan scans for peripherals for a certain duration at fixed regular intervals. In addition to this, we scan for a certain amount of time to connect the GATT server to the device. Once we select the start background scan button, a notification pops up to the user, notifying them that a background scan will continuously run during the background check for the Smart Wellness update.

Once the scan starts, it waits for a certain period to see if Smart Wellness is advertising; if it finds this device, the app reads and displays the data as a notification text and disconnects. If no peripheral is found, the scan automatically stops after a certain time. This scan repeat after every a set period of time(From results we kept to 500 ms)(Fig.11); the one-time scan button is disabled as a scan is already running in the background. The background scan stops when the stop background scan button is selected.

Data from the peripheral is displayed in view between the start and stop background scan buttons (Fig.10).

C. *Multiple Peripheral Connections*

Connecting to more than one peripheral is challenging using a single central Android tablet. The connections are hard coded based on the UUID to ensure each tab connects only to a single device. In Android terminology, a tab is called a fragment. Hence, the user interface contains three fragments (Home, Dashboard, and Notification). Notification Fragment is for future use when connecting to any other peripheral. The following callback methods are updated for every GATT connection. They are:

1. onConnectionStateChange
2. onMtuChanged
3. onCharacteristicChanged
4. onServicesDiscovered
5. onCharacteristicRead
6. onCharacteristicWrite

As these methods are unique to every GATT connection that is established, these callback methods are invoked for every peripheral. Hence, to connect to multiple connections, we go sequentially. We first connect to one peripheral, display the data, and connect to another one. The Android application starts scanning, connects to the smart pill dispenser using its unique GATT connection, and disconnects from this device. Repeating this process, we can continue scanning, reading, and disconnecting sequentially between the two peripherals as shown below.

Peripheral	Seq	BLE Data Read
Smart Pill Dispenser (One time Connection)	1	Was able to read the latest data
Smart Wellness (One Time Connection)	2	Was able to read the latest data
Smart Pill Dispenser (One time Connection)	3	Was able to read the latest data

One of the parameters that can be modified is MTU (Maximum Transmission Unit). We set the maximum transmission unit by default to 512 bytes. However, the latest Android versions set it to 517 bytes by default for the first call. Hence, no further modifications are needed on this. We keep the call to 512 bytes in the onServicesChanged method, which is called when we discover the service UUID. This is when we set the MTU, call the onMtuChanged method, and set the MTU. Even though the MTU parameter can be set directly, the underlying packet transmission is still dependent on the BLE device and network.

D. *Scan time*

As mentioned previously, the Central device, which is the Android app, scans for a certain amount of time before it can pick up the advertised peripheral. For ease of testing, we set the interval to 10 seconds. However, we checked the minimum time to scan and read the data from the peripheral. The bar graph (Fig.11) below shows the time taken to scan and connect and the probability of establishing a connection. We note that at low time intervals (below 250ms), the system is unable to establish a connection while at time intervals above 280ms a connection is reliably established. At time intervals between 250ms and 280ms, connections are established less reliably (i.e., the scan has to be reattempted for a GATT connection).

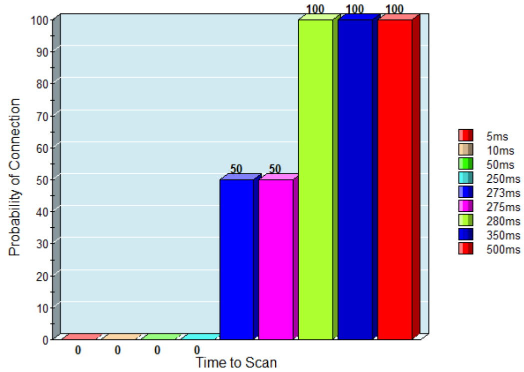

Fig. 11. Scan time vs. Reliability of establishing a GATT connection

V. Conclusion

The paper explains how using BLE IoT sensors in supportive housing incorporates digital health. Supportive housing has patients with health conditions that require timely pill consumption. As elderly patients sometimes forget or are confused about their medication, automatic pill dispensers can provide a reminder. The automatic pill dispenser described here dispenses the pill according to the time set and also monitors whether the pill was consumed or not based on sensor data. This aids in proper pill medication handling for the patients. Smart wellness checks monitor the house non-intrusively to ensure movement in the house. Sensor alerts get triggered if there is no movement for the threshold time set. The sensors are connected to an Android tablet's central UI device. The Android tablet maintains the connections with sensors via Bluetooth. The Android application uses BLE (Bluetooth Low Energy) API to communicate with the sensors. We varied the time it takes to scan and connect to each peripheral. Experimental results show that a scan interval greater than 280ms is required to establish a connection reliably.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 2125654.

References

[1] S. E. Bedell, S. Jabbour, R. Goldberg, H. Glaser, S. Gobble, Y. Young-Xu, T. B. Graboys, and S. Ravid, "Discrepancies in the use of medications: their extent and predictors in an outpatient practice," Arch Intern Med, vol. 160, pp. 2129-34, 2000.

[2] I. Dhawan, A. Tewari, S. Sehgal, and A. C. Sinha, “Medication errors in anesthesia: unacceptable or unavoidable?” *Revista brasileira de anestesiologia*, vol. 67, pp. 184–192, 2017.

[3] B. P. T. Rajan et al., "Smart Pill Box With Reminder To Consume And Auto-Filling Process Using IOT," 2021 Fifth International Conference on I- SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 40-44, doi: 10.1109/I-SMAC52330.2021.9641043.

[4] O. Al-Mahmud, K. Khan, R. Roy and F. Mashuque Alamgir, "Internet of Things (IoT) Based Smart Health Care Medical Box for Elderly People," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-6, doi: 10.1109/INCET49848.2020.9153994.

[5] T. L. Hayes, J. M. Hunt, A. Adami and J. A. Kaye, "An Electronic Pillbox for Continuous Monitoring of Medication Adherence," 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, USA, 2006, pp. 6400-6403, doi: 10.1109/IEMBS.2006.260367.

[6] S.M. Schueller, J. F. Hunter, C. Figueroa, and A. Aguilera, “Use of digital mental health for marginalized and underserved populations,” Current Treatment Options in Psychiatry, vol. 6, pp. 243–255, 2019.

[7] P. P. Morita, K. S. Sahu, and A. Oetomo, “Health monitoring using smart home technologies: Scoping review,” JMIR mHealth and uHealth, vol. 11, p. e37347, 2023.

[8] K.-L. Tsai, B.-Y. Liau, Y.-M. Hung, G.-J. Yu, and Y.-C. Wang, “Development of smart pillbox using 3d printing technology and convolutional neural network image recognition,” Sens Mater, vol. 32, no. 5, pp. 1907–1912, 2020

[9] N. B. Othman and O. P. Ek, "Pill dispenser with alarm via smart phone notification," 2016 IEEE 5th Global Conference on Consumer Electronics, Kyoto, Japan, 2016, pp. 1-2, doi: 10.1109/GCCE.2016.7800399.

[10] W. Antoun, A. Abdo, S. Al-Yaman, A. Kassem, M. Hamad and C. El-Moucary, "Smart Medicine Dispenser (SMD)," *2018 IEEE 4th Middle East Conference on Biomedical Engineering (MECBME)*, Tunis, Tunisia, 2018, pp. 20-23, doi: 10.1109/MECBME.2018.8402399.

[11] V. Peddisetti, P. K. Kandregula, J. A. John, S. Poomdla, K. George and A. Panangadan, "Smart Medication Management: Enhancing Medication Adherence with an IoT-Based Pill Dispenser and Smart Cup," 2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC), Laguna Hills, CA, USA, 2024, pp. 137-144, doi: 10.1109/AIMHC59811.2024.00032