SMART PATIENT MONITORING GLOVE – PROPOSAL REPORT

Group Members: Fatma Sena KALDIRIMCI, Zeynep Sude ŞAHİN, Hakan SOLAK

1.Project Overview

The objective of the project is the design of a wearable glove STM32 based embedded system system that allows patients who have lost the ability to speak or move to easily communicate their basic needs and condition to their caregivers. The patients can activate pre-programmed messages using simple buttons affixed to their fingers. The pulse sensor is integrated with the system to read the real-time health information of the patient and display it on an OLED screen. One of the system's most important features is its ability to provide WiFi capability to the STM32, thanks to ESP32S, and to send all data collected by the system (patient requests and heart rate data) directly to the ThingSpeak cloud, enabling remote patient monitoring. In the event of an emergency, the caregiver can receive notifications via the ESP32S module for rapid response. The possibility of remote patient monitoring will enhance the value of the patient's quality of life and providing safety. Additionally, the embedded system continues to monitor the patient's pulse with aid of the integrated pulse sensor and will send an audible warning with a buzzer in case of danger. A button design on the little finger can enable the room light to be turned on and off from afar, allowing patients to have better interaction with their environment. This system is powered by a portable power bank and in terms of energy efficiency, has low power consumption due to its sleep mode. This project is an innovative embedded system solution that improves patient safety through the use of wearable technology, wireless data transmission, and health monitoring.

2.Project Objectives

Wearable and Ergonomic Input Unit Development: Design a button interface that is easy for the patient to use, controlled by finger capable of distinguishing between short and long presses.
Instant Feedback: Provide visual feedback on the OLED screen (e.g., "Water requested") and auditory feedback (different tones) via a buzzer for each patient action.
Vital Signs Monitoring: Continuously measure the patient's heart rate using an integrated pulse sensor and display this data instantly on the OLED screen.
Remote Monitoring and Data Recording: Periodically send pulse data and patient requests to the ThingSpeak cloud platform via the ESP32S Wi-Fi module and visualize the data.
Notification System: When the patient pushes the button, push notifications (SMS,e-mail or mobile app notifications) will be sent to the caregiver's phone, via service-like ThingSpeak.
Low Power Consumption: Using power management practices to allow the system to run for extended times off a portable power source.

3. System Architecture

The system utilizes the STM32 microcontroller as the core processing unit. This microcontroller gathers and processes data from peripheral devices (sensors, buttons) and sends information to local output devices (display, buzzer) and to the cloud platform through the ESP32S module.

3.1. Hardware Architecture

Central Processing Unit (MCU): STM32

- **Function:** As the brain of the system, the MCU performs all logical functions: reads data from the sensors, processes button press inputs, controls the output units, and communicates with the Wi-Fi module. The

MCU is appropriate for this project and possesses a sufficient number of GPIO, ADC, PWM, I²C, and UART protocols.

Input Units

- **Buttons (4 pieces):** Used to communicate patient requests. Each button is wired to a GPIO pin on the STM32, with the pin status being read through an interrupt-based structure, allowing immediate response time.
- **Pulse Sensor**: This is an analog sensor that measures the heart rate of the patient. The sensor's analog output is connected to the ADC (Analog-to-Digital Converter) pin of the STM32.

Output Units

- **OLED Display:** Displays patient requests, pulse rate, and system status. Communicates with STM32 via I²C communication protocol. This enables efficient communication using a small number of pins.
- **Buzzer:** Provides auditory feedback. Connects to a PWM (Pulse Width Modulation) output of the STM32. By changing the frequency of the PWM signal, different tones are produced when different buttons are pressed.

Communication and Network Unit

- **ESP32S Wi-Fi Module:** It is the unit that allows the system to connect to the Internet.
  Connection: Connection: It communicates with STM32 via UART serial communication protocol (TX-RX pins). STM32 uses ESP32 as a "communication bridge."
  Function: Receives data from the STM32 (e.g., "pulse=75", "request=water"). After connecting to the Wi-Fi network, it converts this data into an HTTP POST request in a format compatible with the ThingSpeak API and sends it to the ThingSpeak servers over the Internet. All Wi-Fi connection and TCP/IP operations are managed by the ESP32. The STM32 triggers these operations by sending AT commands.

Power Unit

- The system's power needs will be met using a standard power bank. The power bank will provide 5V to the system via the USB connection. This approach simplifies the power circuit design and allows for easy charging.

3.2. Software Architecture and Data Flow

The software will be developed in a modular, interrupt-based structure that complies with CMSIS standards. This structure will enable the system to efficiently manage multiple tasks simultaneously.

Data Flow Scenario:

1. Patient Interaction: The patient presses the button indicating that he/she wants water.

2.Interrupt Triggering: A state change occurs on the GPIO pin to which the button is connected, triggering a GPIO interrupt on the STM32. The main program flow is paused, and the relevant interrupt service routine (ISR) runs.

3. Data Processing (STM32): The ISR identifies the button and the duration of the press (short/long). The "Water Requested" message is assigned to a variable.

4. Local Feedback (STM32):

The STM32 sends the text "Water Requested" to the OLED display via the I²C protocol.

The STM32 activates the buzzer by adjusting the PWM signal to generate a specific tone assigned to the relevant button.

5.Preparing to Send Data to the Cloud (STM32):

- The STM32 formats the data to be sent to ThingSpeak (for example, a text like field1=2).
- It sends this text to the ESP32 module via the UART protocol.

6. Communication over the Internet (ESP32S):

○ The ESP32S receives data from the UART.
○ It connects to the preconfigured Wi-Fi network.
○ It creates an HTTP POST request with the ThingSpeak API key and the incoming data.
○ It sends this request to the ThingSpeak server. The data is saved successfully.

7.Simultaneous Heart Rate Measurement:

○ Independent of the above operations, a timer interrupt periodically triggers the ADC module.

○ The ADC reads the analog value from the pulse sensor.

○ The raw data read is processed by an algorithm and converted into a BPM (beats per minute) value.

○ This calculated BPM value is both sent to the OLED display and periodically uploaded to the appropriate field in ThingSpeak via ESP32.

GitHub Link: https://github.com/SenaKaldirimci/SmartPatientGlove