Abdullah Gül University

Department of Computer Engineering

Embedded Systems

Smart Watch Project Progress Report

Instructor

Abdulkadir Gülşen

09.05.2025

Tolgahan Keleş

Muhammed Çağrı Akkuş

Umut Kaya

Introduction

This report aims to provide a comprehensive view of the objectives and ongoing development of the Smart Watch Project. The main goal of the project is to design and develop a wearable device that integrates multiple functions in a simulated environment. Specifically, it is intended to monitor and display key health indicators such as heart rate, keep track of time, and assess various environmental conditions such as temperature. Recently, wearable devices have gained significant attention for their potential to assist in real-time well-being management. This project supports that vision by integrating essential sensors and enabling timely delivery of necessary information to users via their wrist. The report outlines key development stages including simulation design, component selection, implementation progress, encountered challenges, and future goals.

Design

Features:

- **Temperature Monitoring:** Uses the SI7021 sensor to read ambient temperature and displays it on an LCD. A buzzer alerts the user if the temperature is below 15°C or above 40°C.

- **Humidity Monitoring:** Reads humidity data from the same sensor and displays it on the LCD.

- **Time Tracking (Clock):** Uses an RTC module configured via STM32 to continuously show the current time on the LCD.

- **Alarm Function:** Activates buzzer when the set time is reached. Future work may allow reset or snooze functions.

- **Visual Display:** Temperature, humidity, pulse, and time are shown on a 16x2 LCD. Messages such as "ALARM!!!" indicate alerts.

- **Audible Alerts:** Buzzer sounds when thresholds are exceeded or alarms are triggered.

- **Simulation Environment:** Entire system is simulated in Proteus. STM32CubeMX is used for microcontroller configuration and Keil uVision for coding.

- **Chronometer with Touch Button:** A touch button allows users to start, stop, and reset a stopwatch (chronometer) function displayed on the LCD.

Implementation

The implementation phase includes hardware simulation, microcontroller setup, sensor integration, software development, and real-time data visualization.

1. Hardware Simulation Setup

The system was designed and simulated in Proteus. Virtual components include:
- STM32F103C6 microcontroller
- SI7021 sensor for temperature and humidity
- 16x2 LCD display
- 5V Active Buzzer
- Pull-up resistors for I2C stability
- LCD configured in 4-bit mode using GPIO
- Power and ground connected appropriately

2. STM32CubeMX Configuration

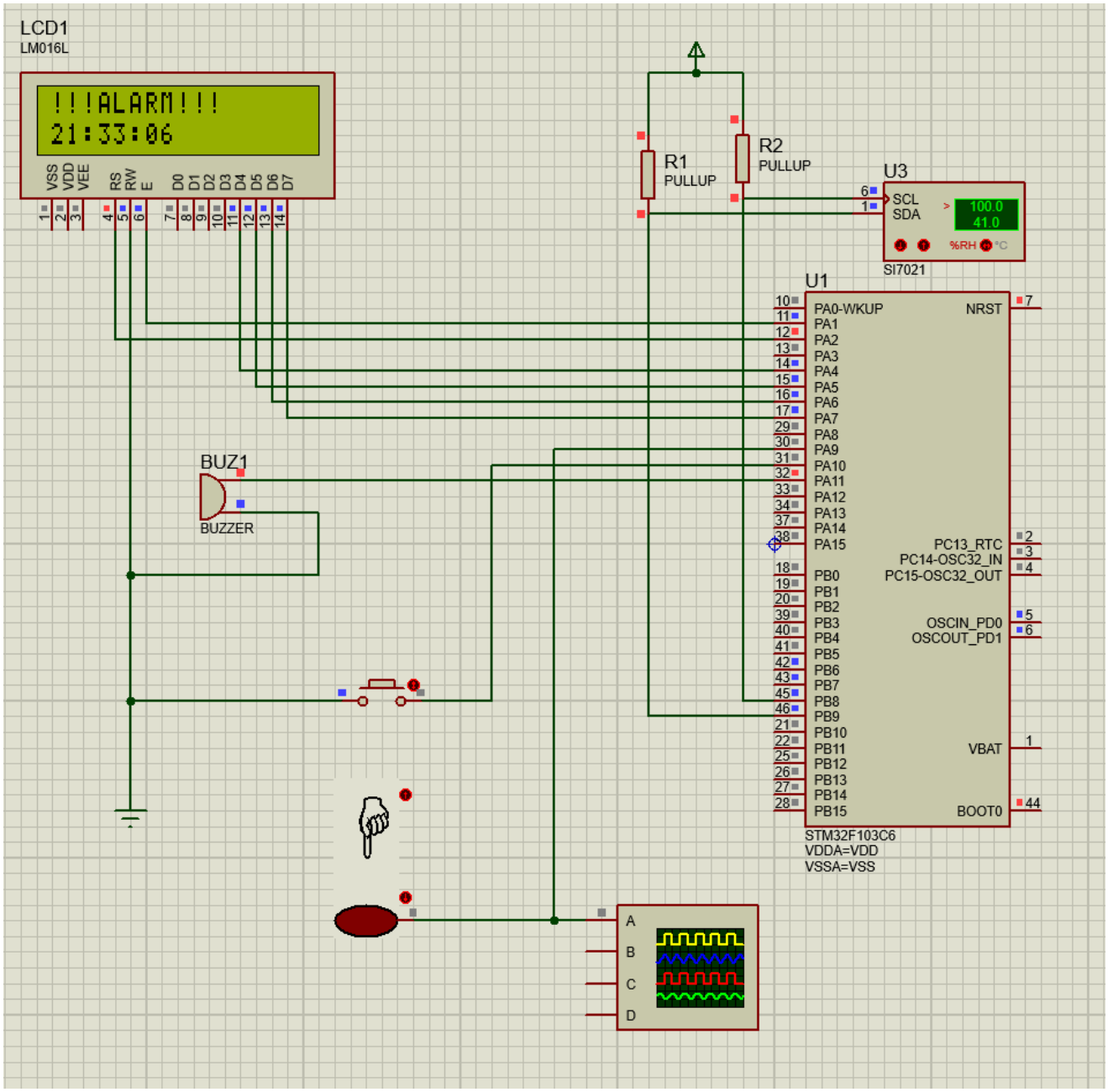
- GPIOs configured for LCD and buzzer
- I2C1 initialized for SI7021 sensor
- RTC enabled via LSE oscillator
- Clock settings and pin remapping completed

3. Firmware Development in Keil uVision

- SI7021 data polled periodically and displayed
- RTC time read and displayed
- Flag-based buzzer logic implemented
- LCD updated every second with data and alarm messages

4. Proteus Simulation and Testing

- Simulated values tested
- LCD output verified
- Buzzer tested under threshold conditions
- I2C validated with waveform and logic probes



Price List of Components

Component	Description	Approximate Price (TL)
STM32F103C8T6 Development Board	Microcontroller unit	₺242,33
DHT11 Sensor	Temperature and humidity sensor	₺295,19
DS3231 RTC Module	Real-time clock module	₺71,66
16x2 LCD Display	Visual output	₺98,33
Active Buzzer 5V	Sound alert module	₺7,49

TOTAL COST: ₺913,10

Progress Made So Far

Hardware Setup

- STM32 board, SI7021, and LCD tested
- Temperature and humidity simulation verified

Software Development

- STM32 peripherals configured
- Initial code developed and RTC integrated

Simulation

- Built simulation in Proteus
- Tested alarm features and data visualization

Challenges Encountered

I2C Communication with the SI7021 Sensor

Integrating the SI7021 temperature and humidity sensor using the I2C protocol proved challenging. The sensor did not always respond reliably to initialization and data read commands, particularly in the early development stages. We had to fine-tune the timing and sequencing of I2C commands and ensure proper pull-up resistor configuration.

FreeRTOS and Proteus Compatibility Issues

We faced issues when simulating FreeRTOS-based code in the Proteus environment. Proteus does not fully support real-time operating systems, leading to unexpected behavior and

instability in task scheduling. This limited our ability to test multitasking features virtually and forced us to test on physical hardware.

ADC Interrupt Handling Problems

We encountered a persistent problem where the ADC (Analog-to-Digital Converter) interrupt could not be properly stopped once started. This resulted in continuous triggering of the interrupt routine, leading to resource conflicts and system instability. The issue was resolved by explicitly disabling the interrupt in the appropriate ISR and control registers after the desired number of conversions.

Future Work

-Allow user-defined alarms

-Add stopwatch, step counter, and sleep monitoring features

-Implement hydration reminders based on activity and temperature

-Enable melody playback on buzzer using PWM

-Add FreeRTOS for improved task management and modular structure

-Solve I2C communication issues with the SI7021 temperature sensor in Proteus simulation

-Integrate pulse sensor and fully implement heart rate display and alerts

Conclusion

The Smart Watch Project has demonstrated substantial progress in both hardware integration and embedded software development. Core functionalities—including temperature and humidity monitoring using the SI7021 sensor, real-time clock (RTC) display, and a fully functional alarm system—have been successfully implemented and verified within a Proteus simulation environment. The system offers both visual and audible feedback through an LCD screen and buzzer, providing a responsive and user-friendly interface.

Throughout development, several technical challenges were encountered and addressed. These included issues with I2C communication for the SI7021 sensor, limitations in FreeRTOS compatibility with Proteus, and difficulties managing ADC interrupts for pulse sensor readings. Despite these obstacles, stable performance was achieved for the core modules.

The project is now positioned for further enhancement. Future work includes adding more advanced health monitoring features such as pulse rate tracking, stopwatch functionality, and hydration reminders. PWM-based buzzer melodies and FreeRTOS integration are also planned to improve responsiveness and multitasking capabilities.

In conclusion, the prototype provides a solid foundation for a compact, simulation-verified smartwatch system, and offers a clear roadmap for transitioning to a physical hardware implementation.