Final Project: Applying FreeRTOS and the embedded systems approach to real-life applications

Overview

This final project aims to consolidate and apply the knowledge and skills acquired throughout the course by designing and implementing a complex, real-time embedded system using FreeRTOS on the ESP32 platform. Students will explore advanced scheduling techniques, interface novel hardware components, and develop user-friendly interfaces. The project will emphasize effective time management, high performance under CPU load, and the integration of dual-core operations to achieve parallel task execution. Through hands-on experience, students will demonstrate their ability to manage tasks, synchronization, and timing, culminating in a functional system that addresses a real-world problem or provides an innovative entertainment solution.

Learning Outcomes

- Apply Embedded System Design Principles: Demonstrate the ability to design and implement a comprehensive embedded system that meets specific project requirements, leveraging hardware and software integration.

- Optimize System Performance: Develop skills in optimizing system performance through effective task management, parallel processing, and real-time scheduling using FreeRTOS on the ESP32 platform.

- Problem-Solving and Innovation: Enhance problem-solving abilities by addressing real-world challenges or creating innovative entertainment solutions through the application of embedded systems technologies.

Yehoshua Luna, 2322458
Carter Lee, 2478429

Instructions

In this final project, you are encouraged to apply the knowledge and skills acquired from Labs 1 to 4, using the Real-Time Operating System (RTOS) to design and implement a complex, embedded system project. The goal is to create a system that not only challenges your engineering capabilities but also addresses a real-world problem or provides entertainment value. Your project should incorporate tasks that demonstrate proficiency in managing time, digital I/O operations, and CPU load optimization, pushing the boundaries of what you have previously achieved.

You will be expected to follow the tasks:

1. Ideate and generate a single proposal document that describes your final project idea
    a. We strongly encourage you to propose your own projects.
    b. The project must satisfy the criteria stated in the project assessment criteria table

2. Work on the final project and deliver a completed and working prototype/solution/product
    a. You will have approximately 2 weeks to complete the project. You are encouraged to develop a timeline early and adhere to it.
    b. You are expected to use the TA office hours for consulting with the TAs. It's not the only time you work on your project.
    c. You are expected to give verbal progress reports at the end of each week to the teaching staff
3. This is an independent, self-guided project. As such no step-by-step lab/final project manual is provided. Your final deliverables will be included as a separate header in this document.

Yehoshua Luna, 2322458
Carter Lee, 2478429

Project assessment criteria

Criteria	Explanation of criteria	Examples	Minimum Requirement
Ensure Reliable Timing and I/O Operations	Your system must manage time effectively and utilize digital input/output with precision.	A real-time clock to manage scheduling, and debounced buttons for user input.	Use at least two ESP-32 timers and implement debouncing for all physical inputs.
Achieve High Performance	The project should be designed to operate under high speed and CPU load. It must include at least one task that operates at or above 50 Hz (20ms period), showcasing efficiency and real-time capabilities.	A motor control system operating at 50 Hz to ensure smooth and precise motor movements.	Implement at least one task running at 50 Hz, and one task running at either 32 or 64 or 128 Hz. Ensure task scheduling and load balancing using FreeRTOS.
Incorporate Novel Hardware	Interface at least two new devices to the ESP-32 ESP-32 that were not part of the earlier lab exercises. This encourages the exploration and integration of unfamiliar components. [Click Here]	Integrating an IMU (MPU-6050) and a Capacitive Soil Sensor to expand sensing capabilities.	Successfully interface and communicate with at least two new peripherals not used in earlier labs.
Feature Measurement or Control	Whether through sensing or actuation, your project should manipulate physical parameters in a meaningful way, addressing a clearly defined problem.	A temperature control system that regulates a heating element based on sensor input from the Temp/Humidity Sensor or locking a door if an unauthorized user is present	Implement a system that measures or controls at least two physical real-world parameters with clear problem-solving.
User Interface	Develop a physical user-friendly interface using components such as switches, keypads, buttons, LEDs, or displays. The project should be operable without the need for direct connections to the development environment or manual code adjustments.	An LCD display (1602 I2C LCD) showing system status and buttons to control system modes.	Develop an interface with at least three components (e.g., display, button, keypad) providing clear user interaction opportunities.
Utilize a	Incorporate a second ESP-32	Two ESP-32	Use a secondary ESP-32 board

Yehoshua Luna, 2322458
Carter Lee, 2478429

Criteria	Explanation of criteria	Examples	Minimum Requirement
Second ESP-32 Board/ Utilize dual core operations	board into your system, utilizing serial communication for interaction between the two boards. Ensure that the secondary ESP-32 is seamlessly integrated with the primary ESP-32, working in unison to accomplish the overall tasks. The functionality of the secondary ESP-32 should not be isolated; instead, it must complement and enhance the capabilities of the primary ESP-32 to achieve a cohesive system operation. Alternatively, you can use both cores of one ESP-32. The requirements (less serial communication) are the same as above.	boards, where one handles sensor data collection and the other manages user interface and control logic.	or both cores of ESP-32. Implement serial communication (or inter-core communication) and ensure seamless task distribution.
Use of Queues for Managing Task and Scheduling	The project must use queues to manage data between tasks and ensure efficient scheduling.	Use queues to handle sensor data and communicate between tasks, such as updating a display based on sensor input.	Implement at least two queues to manage data flow between tasks

Yehoshua Luna, 2322458
Carter Lee, 2478429

Project deliverables and submission requirements

You have to submit the following items:

1. Project proposal
    a. Get verbal Approval from John before you submit
    b. Use template in this document. Example text is provided under each of the headers

Final Project Important dates

2. Final project, in-person demo.
3. Final project report (By deadline)
    a. Use the provided lab template
    b. For Bonus submit a video demonstrating your project working. Upload to YouTube and embed a link in the report.
4. All code. (By deadline)
    a. This includes all your working code, including any external libraries you used, any .cpp files you used, and your .ino file
    b. Submit all your code files as .zip file

Project Examples:

The examples below are to inspire you and guide the thought process on what the project is about and what it should include. If you decide to work on one of these examples instead of creating your own project, make sure to change or add something unique. The “**Smart Plant Monitoring and Watering System**” used in the proposal is also included in this example list

1. **Enhanced Home Security System:** Design a sophisticated security solution using ultrasonic sensors for perimeter detection and PIR sensors for motion detection inside. Integrate a PIN keypad and an RFID keyfob for secure entry. The system should allow for a configurable delay (e.g., 10 seconds) to disarm the system after an alert, with real-time feedback on system status through an LED display or an LCD screen.
2. **Interactive Arcade Console:** Develop an arcade-style console that plays classic games like Space Invaders or Tetris. Use a joystick or thumbstick for navigation and an 8x8 LED matrix for gameplay display. Implement a feature to display the player's score or the number of aliens destroyed on a 7-segment display. Enhance the game experience by adding levels of difficulty and saving high scores.
3. **Personal Weather Station:** Create a weather monitoring system that collects data from various sensors (temperature, humidity, atmospheric pressure) and presents the information on a user-friendly interface, such as an LCD display. Integrate features like weather forecasting, trend analysis, and alerts for extreme conditions. Optionally, add web connectivity for remote monitoring.
4. **Digital Multimeter (DMM) Toolkit:** Enhance the DMM design to support a wider range of measurements, including frequency and temperature, alongside standard parameters like voltage, current, and resistance. Ensure high accuracy across all ranges and

implement a clear, intuitive interface using a 4x7 segment display or an LCD for results. Provide user guidance for measurement selection and potential safety warnings.

Yehoshua Luna, 2322458
Carter Lee, 2478429

Carter & Yehoshua - Final Project Proposal - CSE/EE 474

Description

The proposed project is a "Smart Alarm Clock" designed to provide a customizable and interactive alarm clock. This project aims to improve upon traditional alarm clocks by having remote control functionality and user-friendly interaction. The system will be built around an ESP32-S3 microcontroller, using its dual-core capabilities to efficiently manage timing and user-interface tasks simultaneously. The cores will focus on real-time clock updates, user input, display updates, and alarm control logic. Users can set and control the alarm time using a potentiometer and pushbutton interface or conveniently manage it from a distance via an IR remote. The built-in RTC module ensures precise timekeeping, while a photoresistor automatically adjusts the LCD backlight based on ambient light levels for energy efficiency. When the alarm triggers, a piezo buzzer and dual LEDs will alert the user both audibly and visually.

Components/Diagram

SENSORS		
Component	Description	Link(s)
RTC Module	Provides accurate date/time and temperature information.	ds3231.pdf
Photoresistor	Measures ambient light levels to determine whether the LCD backlight should be enabled or not.	GL5528.xls

MICROCONTROLLER		
Component	Description	Link(s)
ESP32-S3	Acts as the central processing unit. Core one is dedicated to reading and monitoring the sensor. The second core will handle the user interface.	Document Html | Espressif Documentation Document Html | Espressif Documentation

USER INTERFACE		
Component	Description	Link(s)
LCD Display	Shows time, alarm status, alarm time, and temperature.	CN0295D DATASHEET.pdf
Potentiometer	Provides user alarm time selection using voltage divider with ADC pin.	Suntan
IR Remote/Module	Allows the user to turn the alarm on/off from a distance.	KTS017.pdf

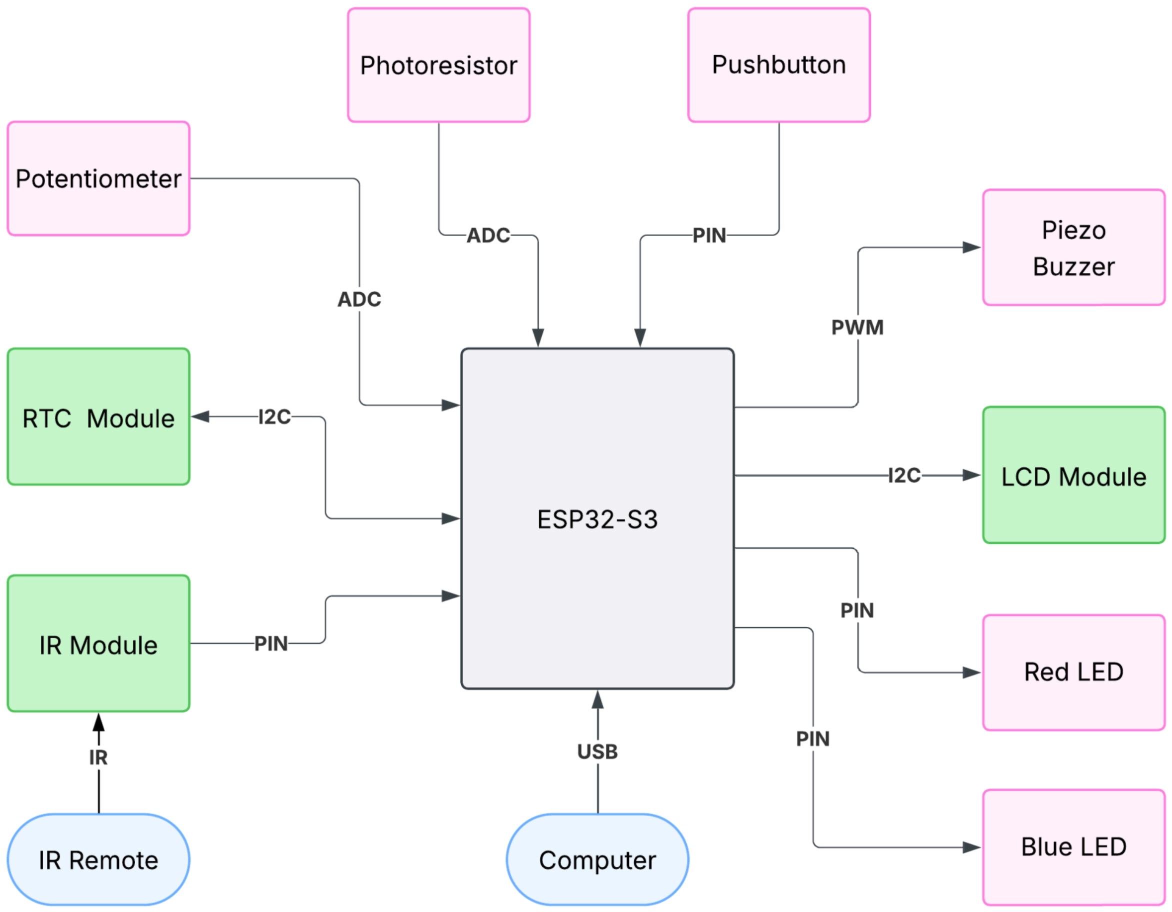
Yehoshua Luna, 2322458
Carter Lee, 2478429

Pushbutton	Used to toggle the alarm on/off and confirm the alarm time selected from the potentiometer.	MS-100630 (Page 1)
Piezo Buzzer	Alerts the user that the alarm time has been reached by playing a melody of notes.	piezoelectronic_buzzer_ps_en.pdf
Two LEDs	Alerts the user that the alarm time has been reached by flashing repeatedly in an annoying fashion.	1498852.pdf

COMMUNICATION		
Component	Description	Link(s)
Serial Communication	Enables data logging and monitoring using a computer for more detailed analysis or long-term tracking.	Serial | Arduino Documentation
I2C Communication	Enables communication between sensors and microcontroller	I2C-bus specification and user manual
GPIO Pins	Enables digital communication between IR Module, pushbutton, piezo buzzer, and LEDs.	GPIO & RTC GPIO - ESP32-S3 - — ESP-IDF Programming Guide v5.5.1 documentation

POWER SUPPLY		
Component	Description	Link(s)
Computer	Provides power to the ESP32-S3, which then powers everything else.	N/A

Yehoshua Luna, 2322458
Carter Lee, 2478429



Project Criteria Checklist

Criteria	How your project meets this criteria
Ensure Reliable Timing and I/O Operations	One ESP32 timer will be used for collecting IR module measurements at about 38kHz. Another can be used for periodic potentiometer, photoresistor and LED functions. Finally, a third timer will be used for PWM sent to the piezo buzzer. Timer interrupts will be used for pushbutton presses.
Achieve High Performance	Photoresistor and potentiometer data will be measured at a rate of 32Hz or more, RTC Module data will be read at a rate of 1Hz, and IR Module data will be read at approximately 38kHz.

Yehoshua Luna, 2322458
Carter Lee, 2478429

Incorporate Novel Hardware	We will be using the IR Receiver/Module and the RTC Module, both of which have not been used previously in this course.
Feature Measurement or Control	Our system measures time, temperature, and ambient light intensity.
User Interface	The LCD screen will be used as our primary way to display information to the user. A pushbutton and potentiometer will be used to set the alarm time and status. The IR Module offers a secondary layer of user input from a distance. (The other option is to just use the remote for inputs.) A piezo and two LEDs will be used to alert the user that the alarm has gone off.
Multi-core/ Multi-controller	The alarm clock tasks will be split between the two ESP32-S3 cores. One will handle sensor information and communication, the other will monitor the user interface and alarm.
Queue for data handling	Photoresistor measurements will be stored in a queue that holds the past few seconds of data, which is then averaged to determine whether the ambient light level is high enough to turn off the backlight. The same will be done for temperature measurements received from the RTC Module.