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.

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

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.

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

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

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

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

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

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

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

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.

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

<Canvas Lab group Number> - Final Project Proposal - CSE/EE 474

Description

Provide a brief overview of the project. Explain the motivation behind the project, the problem it aims to solve, or the opportunity it seeks to capitalize on. You can use this document as a template for your proposal submission.

Example:
The proposed project is a "Smart Plant Monitoring and Watering System" designed to automate the process of monitoring and watering plants, ensuring they receive optimal care without constant human intervention. Motivated by the common challenge of maintaining healthy plants amidst busy schedules or limited horticultural knowledge, this project addresses the frequent issues of overwatering and underwatering. By integrating sensors and actuators with an ESP-32 ESP32, the system will monitor soil moisture levels and ambient light conditions to automate watering schedules, providing consistent and precise care. This innovative solution not only enhances convenience but also promotes sustainable water usage, leveraging the growing interest in smart home devices and the Internet of Things (IoT) to create an automated system that ensures optimal plant health with minimal human involvement."

Components/Diagram

Draft a diagram that outlines the components and hardware planned for your project at a high level. You should itemize each component separately, with a datasheet and/or purchase links. You are expected to also show different project requirements are satisfied here. Your project needs to make use of either the esp-32's multi-core capabilities, or integrate multiple esp-32 microcontrollers.

Please be aware that adjustments to your initial plan are acceptable, provided that you maintain open communication with the teaching assistants as your project progresses.

NOTE: We have a list of available components listed in the appendix. We have limited quantities of these sensors. You can collect the sensors as needed from the ECE store in ECE 137. You must return these sensors at the end of the quarter. You are allowed to use sensors other than the ones listed, but you are responsible for acquiring the sensors.

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

Example:
Smart Plant Monitoring and Watering System

SENSORS		
Component	Description	Link(s)
Soil Moisture Sensor	Detects the moisture level in the soil.	[Link to Documentation](#) [Link to Purchase](#)(if any)
Light Sensor	Measures the ambient light level to assess if the plant is receiving adequate sunlight.	[Link to Documentation](#) [Link to Purchase](#)(if any)

MICROCONTROLLER		
Component	Description	Link(s)
Arduino ESP-32	Acts as the central processing unit. Core one is dedicated to reading and monitoring the sensor readings. The second core will control the actuator for the pump and control the light. This task separation ensures that the sensor readings are not affected by the controller actions and maximizes system response time.	[Link to Documentation](#) [Link to Purchase](#)(if any)

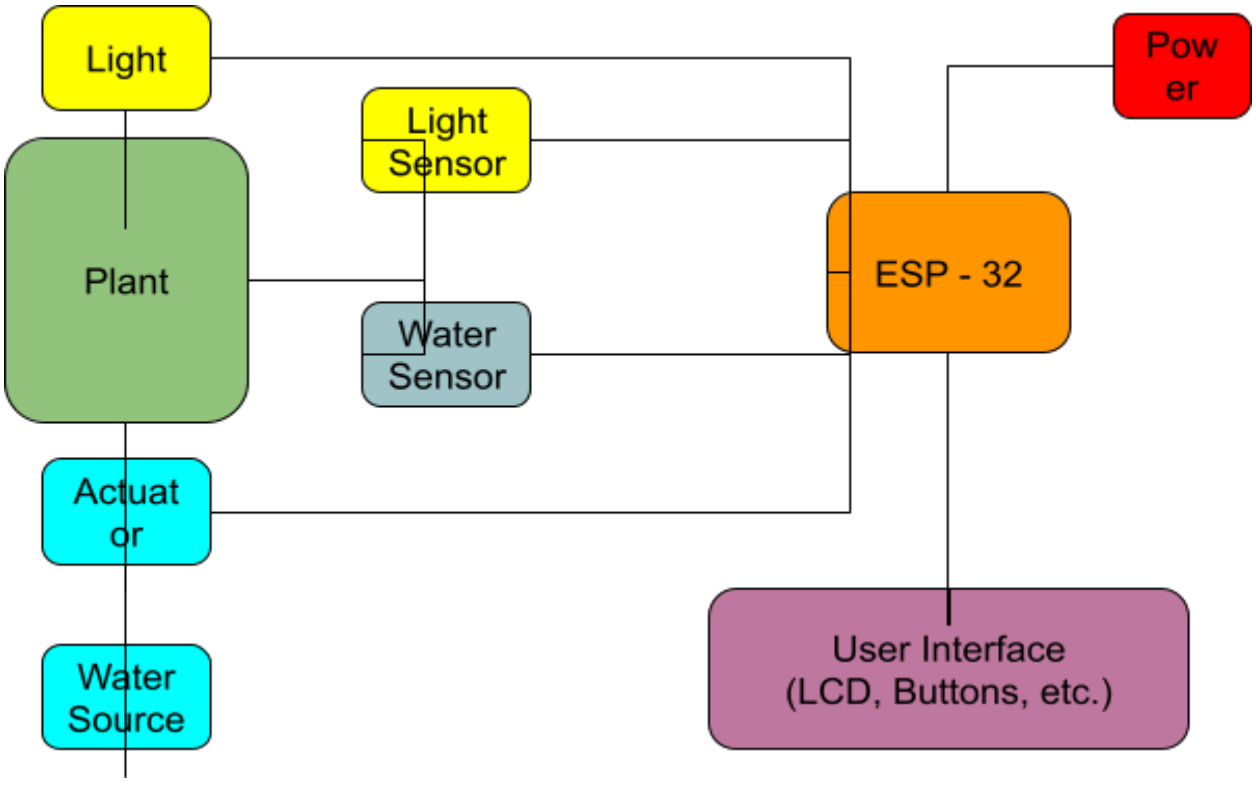
ACUTATTOR		
Component	Description	Link(s)
Water Pump/Solenoid Valve	Controls the flow of water to the plant based on commands from the Arduino ESP-32.	[Link to Documentation](#) [Link to Purchase](#)(if any)

USER INTERFACE		
Component	Description	Link(s)
LCD Display	Shows real-time data, including soil moisture levels, light levels, and watering status.	[Link to Documentation](#) [Link to Purchase](#)(if any)
Buttons/Keypad	Allows the user to manually override automatic watering, set moisture thresholds, or configure watering schedules.	[Link to Documentation](#) [Link to Purchase](#)(if any)

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

COMMUNICATION		
Component	Description	Link(s)
Serial Communication	Enables data logging and monitoring on a PC/laptop for more detailed analysis or long-term tracking.	[Link to Documentation](#) [Link to Purchase](#)(if any)
I2C Communication	Enables communication between sensors and microcontroller	[Link to Documentation](#) [Link to Purchase](#)(if any)

POWER SUPPLY		
Component	Description	Link(s)
Power Supply	Provides power to the Arduino ESP-32, sensors, water pump, and LCD display.	[Link to Documentation](#) [Link to Purchase](#)(if any)



References:

Include any references or citations for the research and sources you used in preparing your proposal.

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

Project Criteria Checklist

(Fill out the last column of this table, explaining how your proposal meets the criteria. You can find detailed criteria in final project document:)

Criteria	How your project meets this criteria
Ensure Reliable Timing and I/O Operations	
Achieve High Performance	
Incorporate Novel Hardware	
Feature Measurement or Control	
User Interface	
Multi-core/ Multi-controller	
Queue for data handling	

<Group member name 1, Group member student ID 1>
<Group member name 2, Group member student ID 2>

Appendix: List of available components

TBD