02112 – Group Project

This project involves independently designing, building, and programming an embedded system that can interact with its environment through various input/output (I/O) methods.

The project is divided into two phases, which must be completed in sequence. You must first complete Phase 1, which involves building and programming a preset circuit. You **must** then have the teacher or TA approve it **before** proceeding to Phase 2. At the end of the course, you will be required to produce a report that describes the system you designed and explains its key features. On the last day of the course, each group will present their system at a final presentation.

Deadlines:

Group formation and registration:	6/1-26

Phase 1: No specific deadline, but must be finished and accepted by the teacher/TA before moving on to Phase 2

Phase 2 Report:	21/1-26
Final Presentations:	23/1-26

Phase 1: Environment Monitoring Station

You must build a system as shown in the circuit diagram below. The system is a light, temperature, and humidity monitor. It runs in two modes: 1) light monitoring and 2) temperature and humidity monitoring. In each mode, an RGB LED is used to display the light or temperature level using colors. For example, when monitoring light, red may signify the highest light intensity, while purple may signify complete darkness. A button is used to switch between the two modes. The exact functionality is described below.

Build

The system consists of the following components:

- ESP32-C3 as the main MCU
- Single-color LED (labelled 'MODE_LED')
- RGB LED (labelled 'RGB_LED')
- Button
- Photocell resistor as a light sensor
- AM2320 temperature and humidity sensor
- Various resistors, where needed

You must build the system so that it functions correctly. However, you do not need to use the exact pin placement or wiring. You must only ensure the system functions correctly. Also note that the diagram is simplified regarding the VCC and GND wiring. Ensure that VCC and GND are connected to the ESP32.

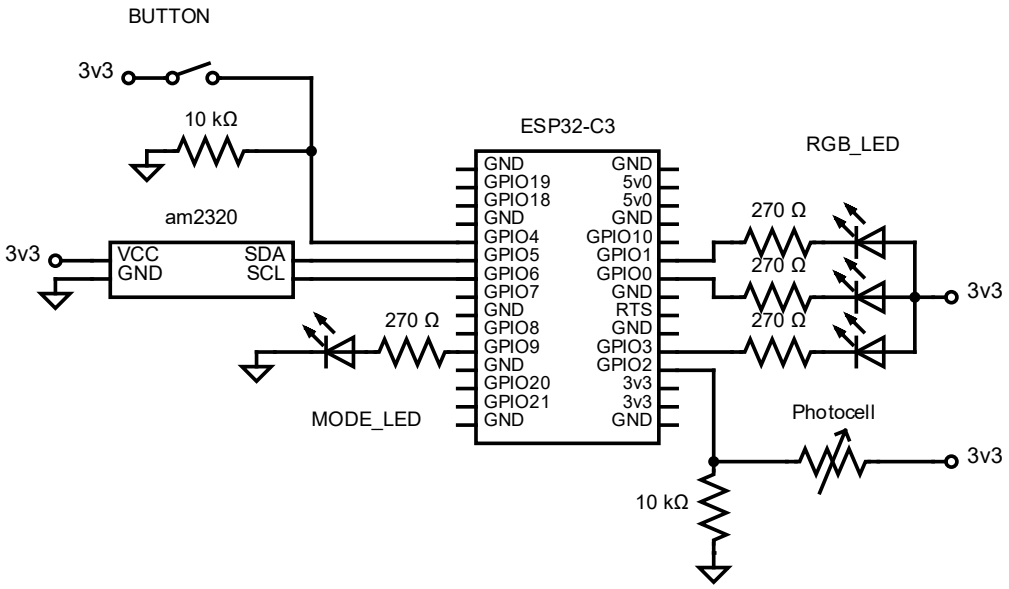

Figure 1- Phase 1 circuit diagram

Programming

You must program the built system to function as a monitoring station. When turned on or reset, it must first go through a startup sequence, which can be used to verify that the system is built correctly:

Startup sequence:

1. The MODE_LED is switched on and off 3 times, with a 1-second delay.
2. The RGB_LED must cycle through all its colors (each for 1 second), in the sequence showing the lowest to the highest measurement intensity.
3. The light sensor must be calibrated. The user must first be prompted to keep the sensor clear and then press the button. The user must then be prompted to cover the sensor completely (usually with the finger) and press the button again. The two button presses must be used to get a lower and upper bound to the voltage through the light sensor.

   All the above steps should include reasonable text output on the UART (using 'printf'), logging the startup sequence, and guiding the user through the process.

After the startup sequence is complete, the system should proceed to continuous monitoring of the sensors, with light monitoring as the initial mode of operation.

Monitoring:

The system should go through the following loop indefinitely:

1. Check if the button is being pressed; if so, switch mode and wait for the button to be released. While the button is pressed, nothing happens.
2. Take one measurement of either light, temperature, or humidity.
3. In light mode, display the light intensity as one of 6 different colors on the RGB LED. Also, print the light intensity over the UART.
   In temperature and humidity mode, display the temperature as one of the six colors on the RGB_LED. Assume a minimum temperature of 22 °C and a maximum of 28 °C. Also print both the temperature and humidity (%RH) over the UART.
4. Wait for 1 second, then repeat step 1.

Note: the AM2320 sensor will likely give erroneous responses over the I2C bus for the first few measurements. This must be reflected in the UART output as error messages, rather than temperature and humidity values.

Approval

When the system is built, call the teacher or TA. They will then test your system and verify its build and programming. They must allow you to proceed to the next phase. **Not receiving a Phase 1 approval will negatively impact your final grade.**

Phase 2: Build Your Own System

Do not start this phase until your Phase 1 system has been approved!

In this phase, you must design, build, and program a complete embedded system. You are free to either start from scratch or extend and improve your phase 1 system. The final product should be a coherent system that provides a set of functionalities. You may use components available as part of the course or your own components if you prefer.

The system will be evaluated based on its use of topics covered in the course. Below is a list of topics that may be used in your system. To achieve a passing grade in Phase 2, groups consisting of three people must have three of the following topics in their system. Groups of 4 must have at least 4 of the following topics:

- Use of volatile pointers. I.e., where removing ‘volatile’ from the code results in the system misbehaving.
- Use of memory management (malloc, free).
- Using PWM output.
- Using an analogue input.
- Using proper button debouncing.
- Using an interrupt. Each interrupt kind counts independently.
- Using either UART or SPI protocols.
- Using advanced components (each component counts independently). Examples include, but are not limited to, 7-segment displays, shift registers, servos, and LCD screens. Ask the teacher or TA whether other components may be classified as ‘advanced’. Components that use UART, I2C, or SPI protocols are considered advanced.

If you extend the Phase 1 system, none of the existing components will count towards the Phase 2 grading unless they are changed or improved.

Builds of higher quality (those that work without issues, integrate components well, etc.) and those with more points from the above list will be evaluated higher.

Report:

You must produce a report of no more than 3 pages, which is to be handed in by the deadline specified on the first page. The report must describe the system you have designed and what functionality it provides.

Here is a checklist of things that must be present in the report:

- A title.
- Names and student numbers of all group members.
- A very short section describing the contribution of each group member. If all members worked equally on all parts, this could also be stated.

- A description of the functionality of the system, potentially with pictures to illustrate.
- A description of the circuit built.
- A circuit diagram accurately describing the system. You may simplify the wiring of VCC and GND, as done in Figure 1 above. Pin assignments must be faithful to the actual build.
  You may use this tool to create your diagrams.
- If something doesn't work correctly, include a section on problems, their potential sources, and recommended remedies. Bugs and issues will not negatively affect the evaluation if described in this section.

In addition to the report, you must hand in your C source code in a single file and a picture of your system. This makes a total of 3 files that need to be handed in:

1. Report
2. C source code
3. Picture of your system

Final Presentation:

On the last day, each group will be examined on the system they built. At the presentation, you will have 5-10 minutes (do not exceed this time limit!) to present what you have built. You should bring slides and are expected to demonstrate the working system. After your presentation, we will have a 5-minute discussion with you on your project. The total presentation time for each group is 15 minutes.

You need to highlight the following elements in your presentation:

- The intended functionality.
- The circuit diagram with a description.
- The topics integrated into the system (see the above list).
- If you have made any improvements to the system between handing in the report and your final presentation, highlight what they are.

During the discussion, we will ask individual members of the group questions to verify that they contributed what the report states they contributed. If the report states that everyone contributed equally, we expect everyone to be able to answer questions about all aspects of the project. Additionally, please have your C code available so we can review it.