**A black background with a black square

Description automatically generated with medium confidence**

**TITLE**

**Light Sensor Interfacing with PWM Output LED**

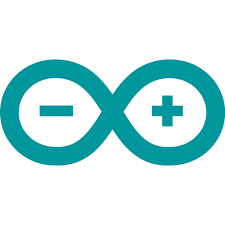
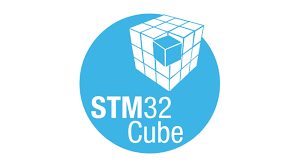
**GROUP 01:**

GAGANDEEP

HRISHIKESH

PRANAV R K

VINAYRAJ B

****

|  |  |
| --- | --- |
| INDEX | Pg No |
| Problem statement | 03 |
| Objectives | 03 |
| Flow chart | 04 |
| V-Model correlation | 04 |
| Project setup | 05 |
| Pin configuration | 06 |
| Result and Conclusion | 07 |

**Problem Statement:**

1. Estimate the ambient light level and set it as a base level.

2. Using the LDR (light dependent resistor) as a light sensor to sense the ambient light intensity.

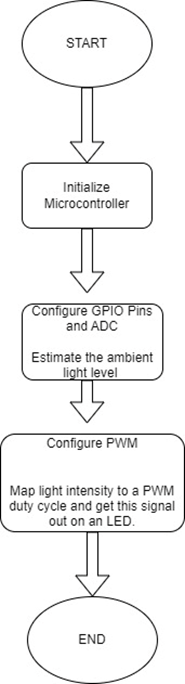
3. Map light intensity to a PWM duty cycle and get this signal out on an LED.

4. The LED brightness increases as the surrounding gets darker.

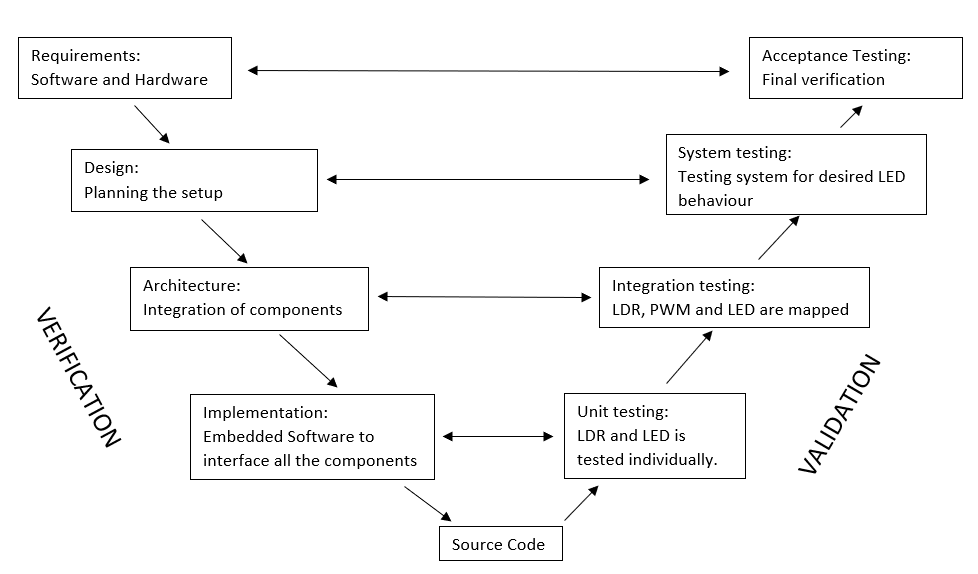
**Objectives:**

1. Estimating ambient light level
2. Sensing light intensity through LDR
3. Mapping the output of LED to generate PWM signals.
4. Achieving desired behavior of increasing LED brightness as surrounding gets darker.

**Flow Chart:**

****

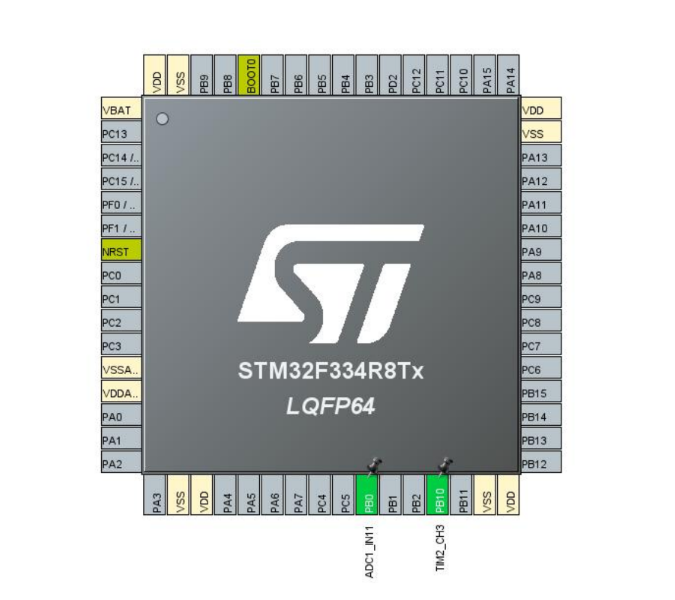
**V-Model Correlation:**

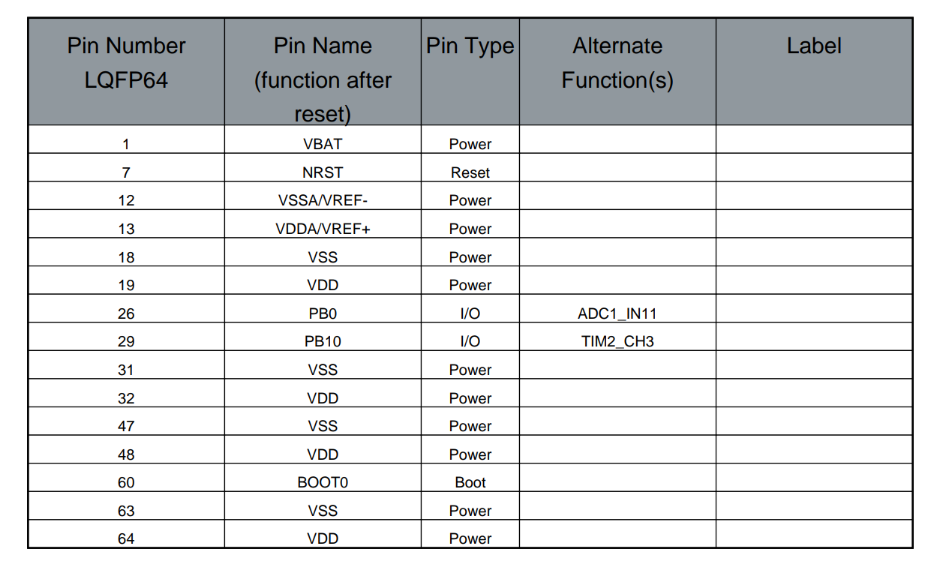
****

**Project Setup:**

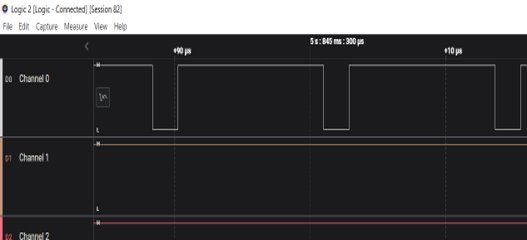
1. **Hardware Setup:**
   * Connect the LDR to an Analog pin on the STM32F33 microcontroller. Ensure that you have appropriate voltage dividers and resistors, if needed, to create a voltage range suitable for the ADC (Analog to Digital Converter) input.
   * Connect an LED to one of the PWM-capable pins on the microcontroller.
2. **Initialize the Microcontroller:**
   * Set up the STM32F33 microcontroller using the STM32CubeIDE or STM32 Arduino library. Ensure you have the necessary libraries and tools installed for your development environment.
3. **Configure the ADC:**
   * Configure the ADC to read the Analog voltage from the LDR. Set the resolution and reference voltage as needed.
4. **Configure the PWM:**
   * Configure the PWM module to control the LED's brightness. Set the PWM frequency and other parameters according to your requirements.
5. **Main Loop:**
   * In the main loop of your Arduino or STM32 code, continuously read the Analog value from the LDR using the ADC.
6. **Map Light Intensity to PWM Duty Cycle:**
   * Map the Analog reading from the LDR to the PWM duty cycle. You can use a simple linear mapping or more complex algorithms depending on your requirements. The goal is to increase the LED brightness as the surrounding light gets darker. You may need to experiment with the mapping values to achieve the desired effect.
7. **Output PWM Signal:**
   * Set the PWM duty cycle to control the LED brightness based on the mapped value.
8. **Delay:**
   * Introduce a small delay in the loop to avoid rapid fluctuations in LED brightness due to small changes in ambient light.

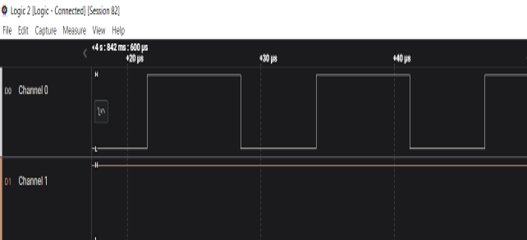
**Pin Configuration:**

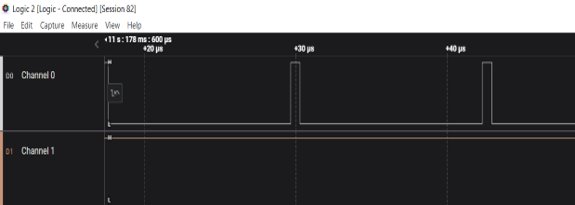
****

****

**Result:**

**** ****

**Conclusion:**

* **Analog to Digital Conversion**: Learning how to interface with Analog sensors, convert their output to digital values, and read those values using an ADC.
* **PWM Control**: Implementing PWM (Pulse Width Modulation) to control the brightness of an LED by mapping the light intensity measured by the LDR to a PWM duty cycle.
* **Project Lifecycle**: Understanding how the V-Model and testing methodology can be applied to manage and correlate project phases and testing phases effectively.