Adaptive LED Brightness Based on Ambient Light

Student Name(s): Aryan Basnet,Pushkar Lokhande,Sanket Chavan.

College: AISSMS IOIT

Date: 07/11/2025

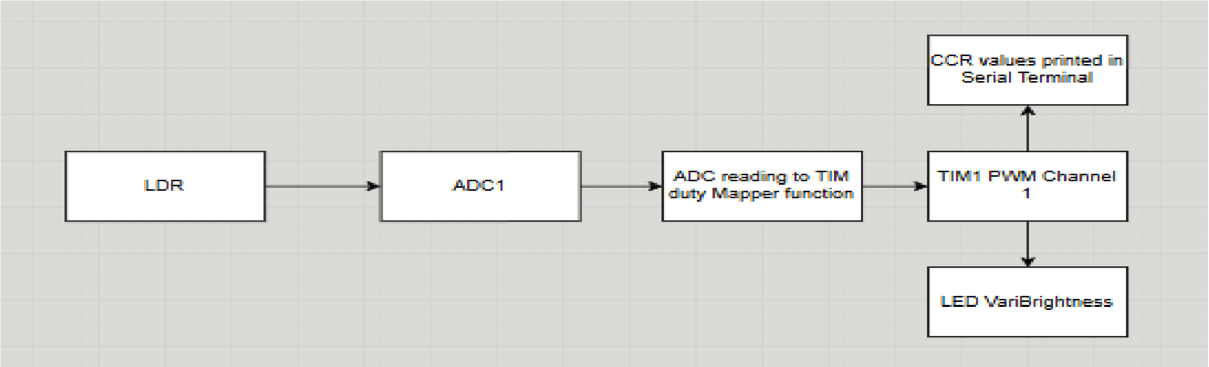
I. Project Summary

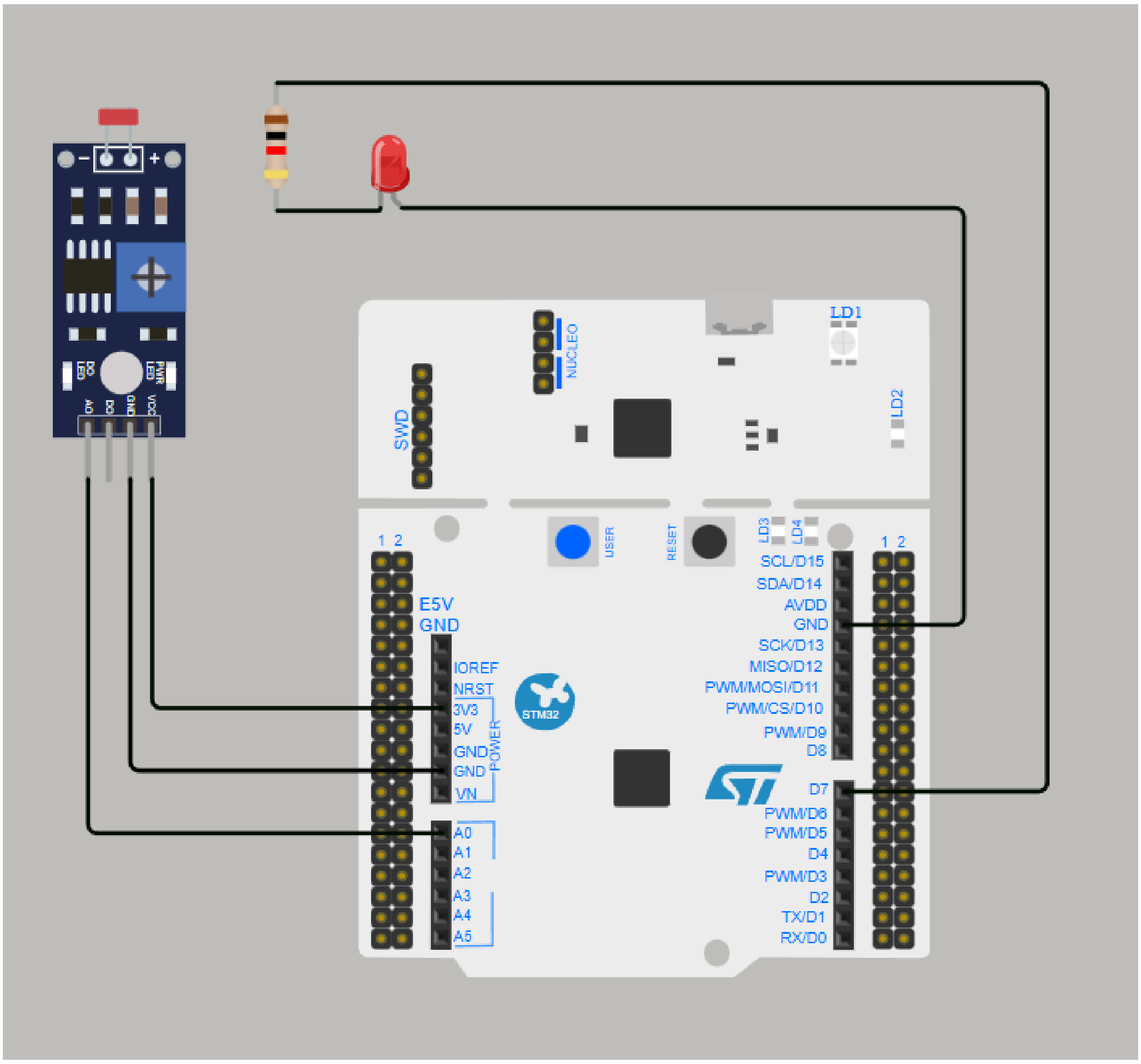
This project implements an energy-efficient streetlight system using an STM32F446RE microcontroller. The LED brightness adapts to ambient light levels measured by an LDR. The system uses ADC for sensing, PWM for brightness control, and UART for logging. It follows an Agile development process with iterative sprints and documented user stories.

II. Agile Process Documentation

User Story ID	User Story	Action Items	Acceptance criteria
US1	As a developer, I want to measure ambient light using an LDR so that I can adjust LED brightness accordingly. .	Design voltage divider circuit with LDR Configure ADC input pin Sample every 500ms.	ADC reads light level every 500 ms, Values vary with light intensity.
US2	As a developer, I want to control LED brightness using PWM so that it responds smoothly to ambient light changes.	Configure timer for PWM Map ADC values to duty cycle Update PWM output	LED brightness changes smoothly- No flickering or abrupt changes
US3	As a developer, I want to log light level and duty cycle over UART so that I can monitor system behavior.	Initialize UART- Format and transmit logs every 500 ms	UART prints: LDR Value, Duty: ADC value or MAP Value Logs match LED behavior

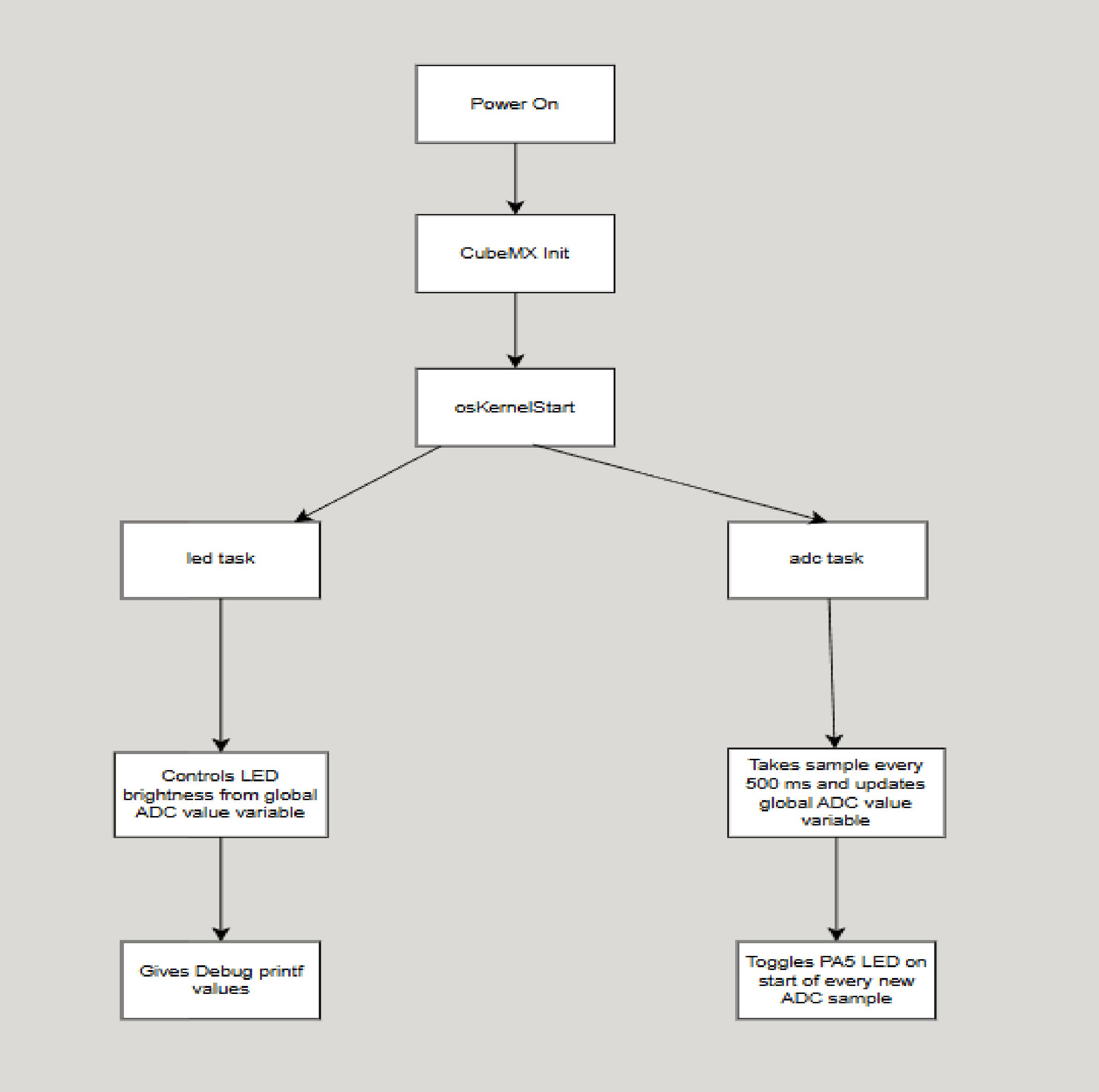
III. System Design


LDR
ADC1
ADC reading to TIM duty Mapper function
TIM1 PWM Channel 1
CCR values printed in Serial Terminal
LED VariBrightness



IV. Implementation

- **ADC**: 12-bit resolution, sampling every 500 ms
- **PWM**: TIM1, 2kHz frequency, Variable duty
- **UART**: Baud rate 115200bps


Power On
CubeMX Init
osKernelStart
led task
adc task
Controls LED brightness from global ADC value variable
Takes sample every 500 ms and updates global ADC value variable
Gives Debug printf values
Toggles PA5 LED on start of every new ADC sample

V. Calculations

- ADC range: 0–4095
- Duty cycle = (ADC value / 4095) × 100%
- Range scaling value = 10.25

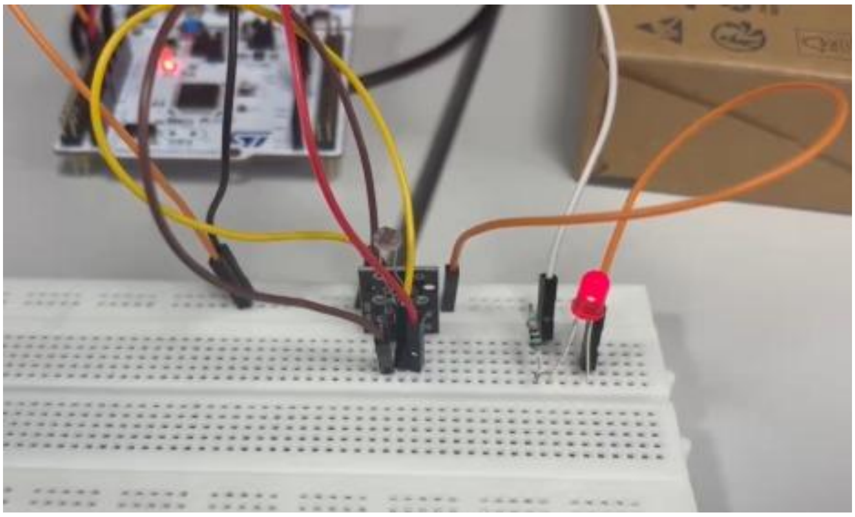
VI. Test Cases

Test Case ID	Test Description	Expected Result	Validation Method	Test case working or not
TC1	Verify LED turns ON Or OFF using GPIO Toggle	LED turns ON and OFF	Visual confirmation	Working
TC2	UART check basic UART functionality using “Hello World’ at 115200 baud rate	UART displays “Hello World” on Putty	Visual confirmation	Working
TC3	Verify LDR Readings of ADC	Displays ADC value in UART	Visual confirmation UART log consistency	Working
TC4	Verify and check PWM CCR on UART	Displays Duty cycle	Visual confirmation	Working
TC5	Verify ADC sampling after every 500ms.	ADC value should update correctly every 500 ms with stable readings reflecting the input signal.	PA5 LED toggle every 500 ms in adc_sampling task	Working

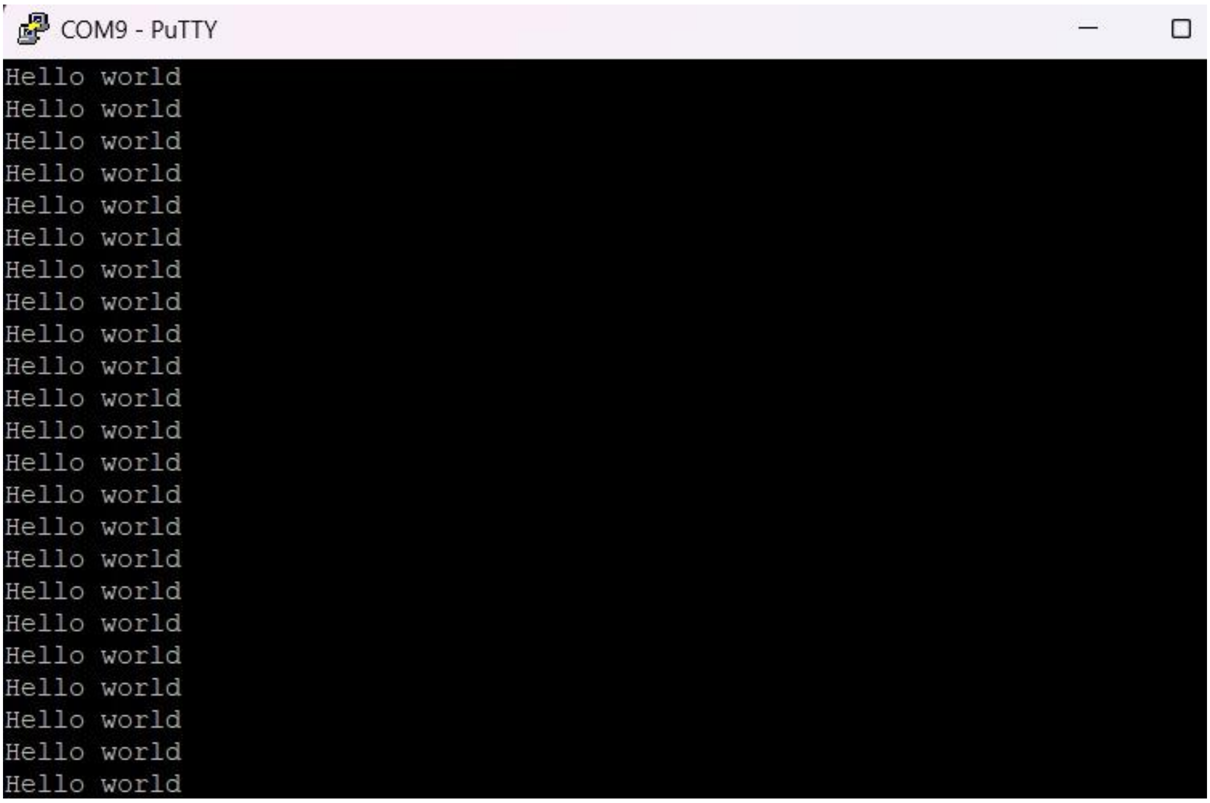
TC6	Verify LED turns ON when there is no incident light on the LDR sensor.	Light - LED OFF or No Light- LED ON	Visual confirmation	Working
TC7	Verify system monitors light intensity and changes brightness with every new ADC sample.	System reads and updates light intensity values every 500 ms , accurately reflecting any changes in ambient light within each sampling period.	LED Toggle for every 500 ms ADC sample and LED variable brightness works as expected	Working

Test Case Validation Media:

TC1 –

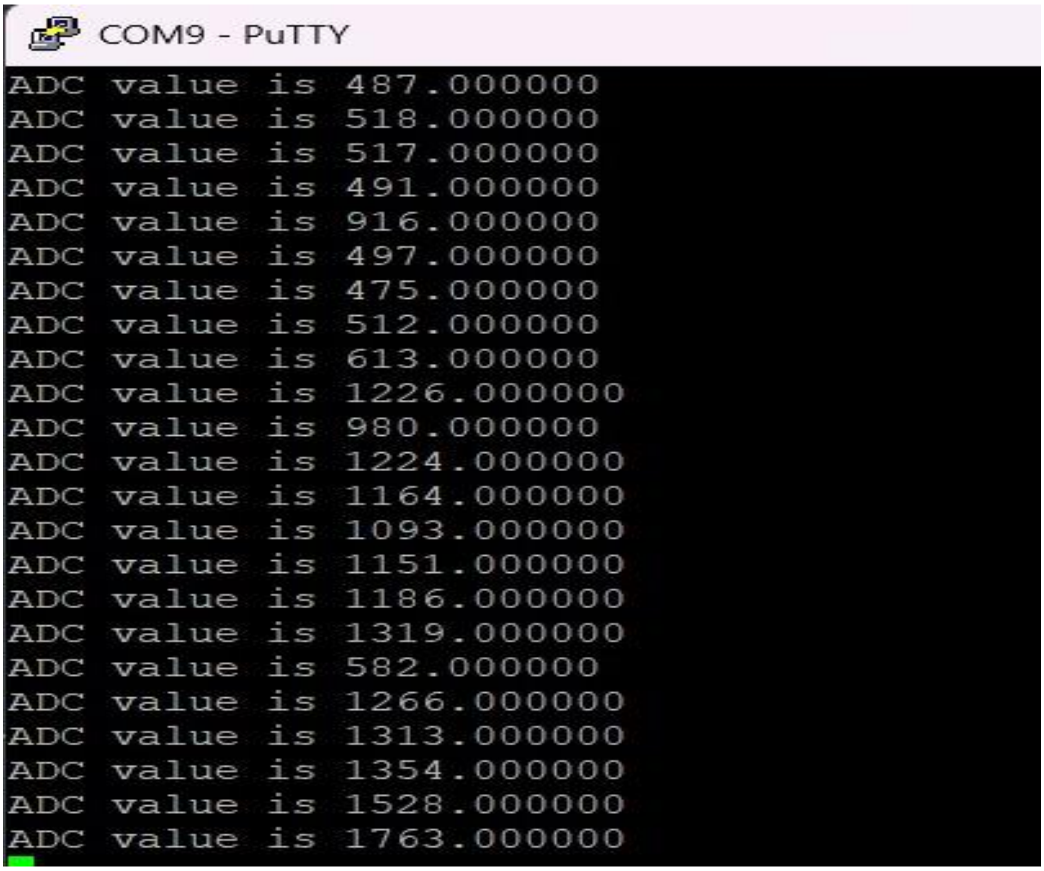


TC2 –



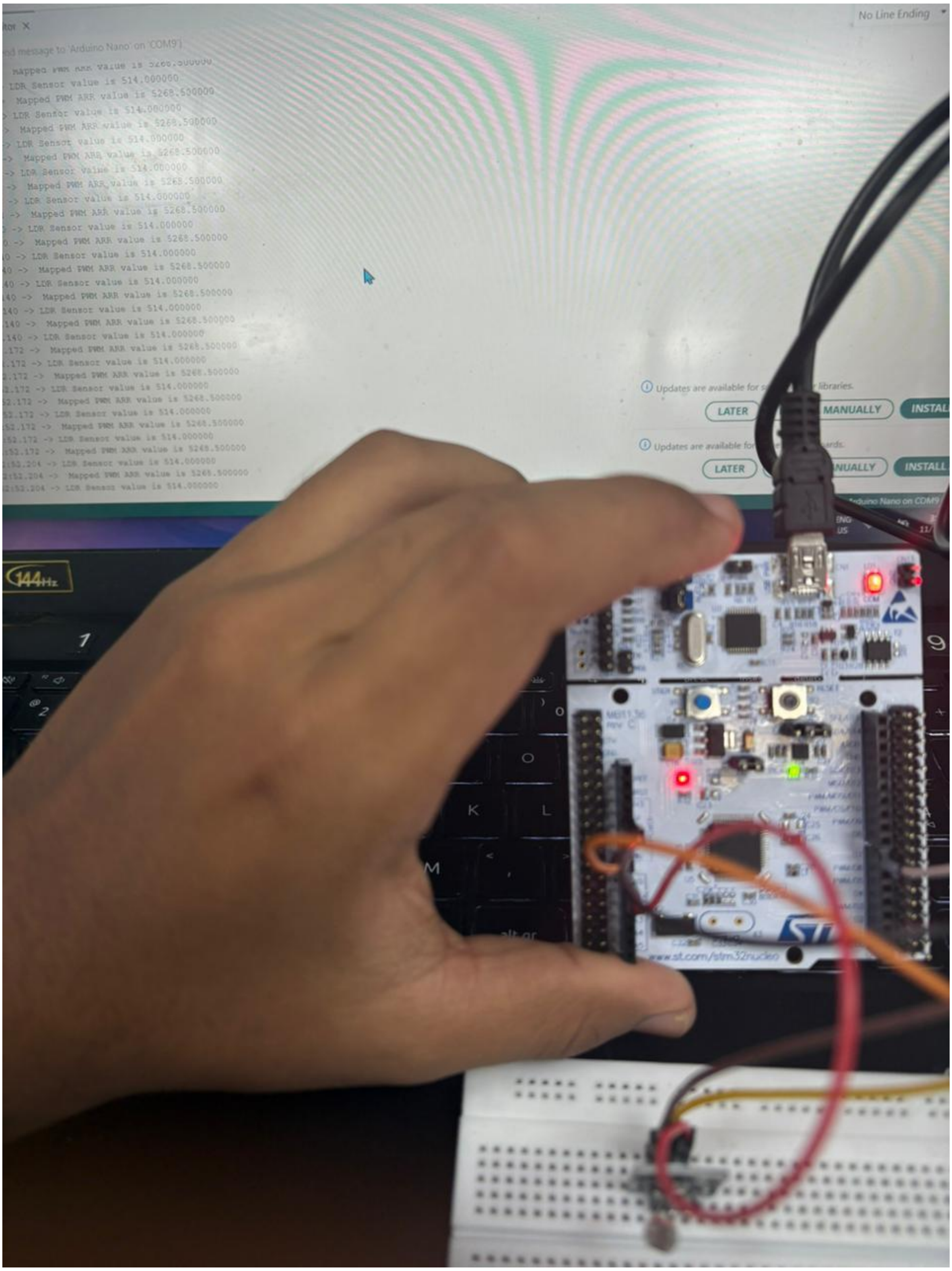
```
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
Hello world
```

TC3 & TC4 –



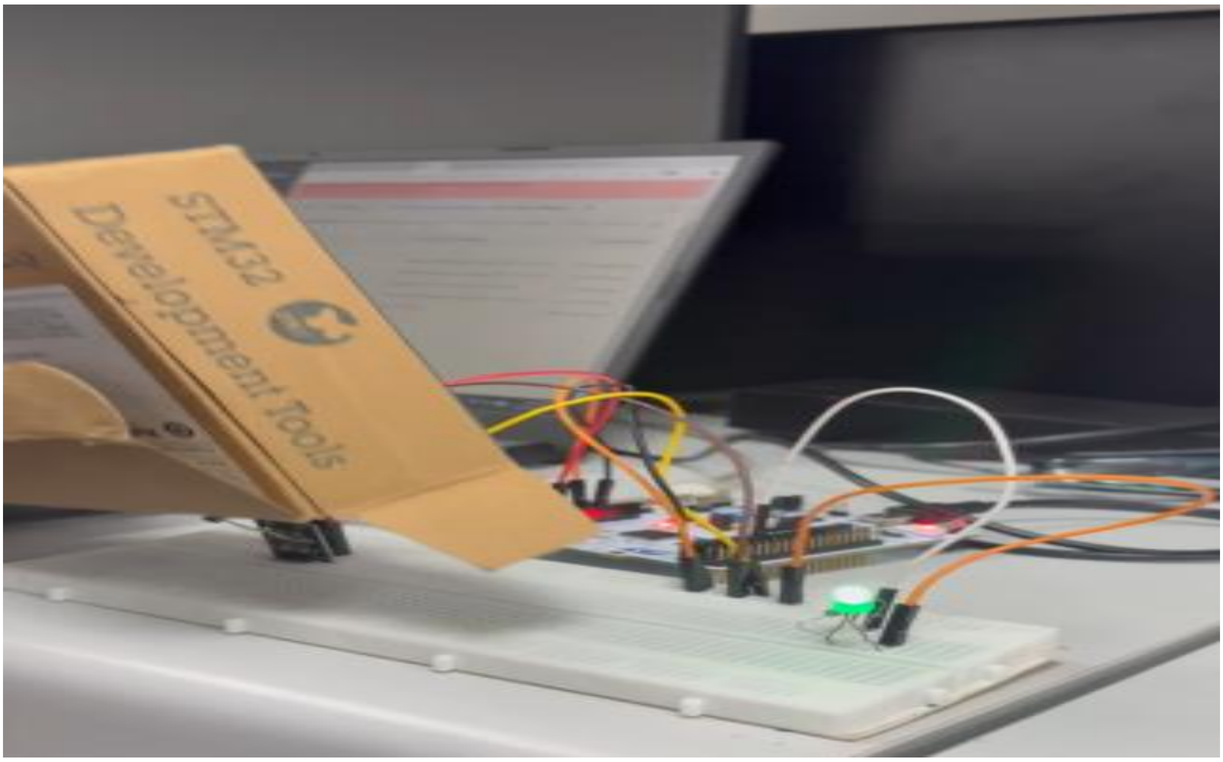
```
ADC value is 487.000000
ADC value is 518.000000
ADC value is 517.000000
ADC value is 491.000000
ADC value is 916.000000
ADC value is 497.000000
ADC value is 475.000000
ADC value is 512.000000
ADC value is 613.000000
ADC value is 1226.000000
ADC value is 980.000000
ADC value is 1224.000000
ADC value is 1164.000000
ADC value is 1093.000000
ADC value is 1151.000000
ADC value is 1186.000000
ADC value is 1319.000000
ADC value is 582.000000
ADC value is 1266.000000
ADC value is 1313.000000
ADC value is 1354.000000
ADC value is 1528.000000
ADC value is 1763.000000
```

TC5 –

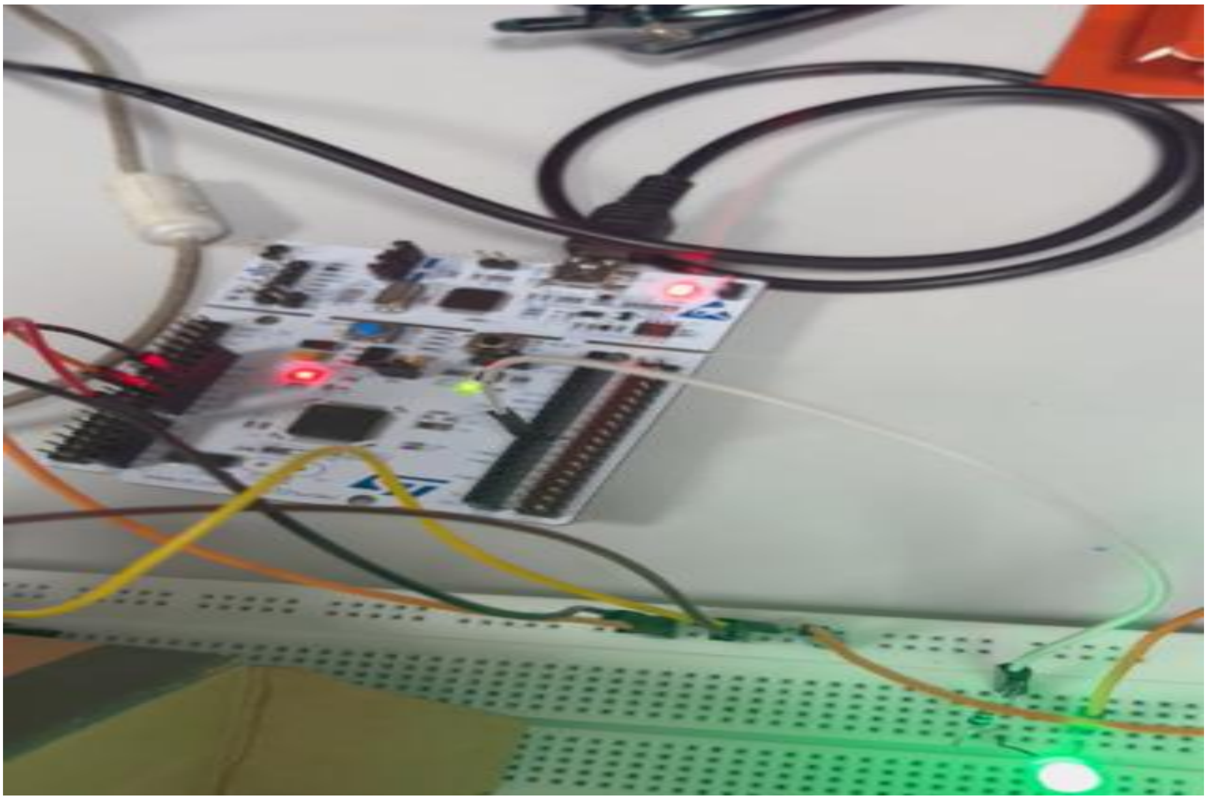


```
tor ×
nd message to 'Arduino Nano' on 'COM9')
 Mapped PWM ARR value is 5268.500000
 LDR Sensor value is 514.000000
  Mapped PWM ARR value is 5268.500000
> LDR Sensor value is 514.000000
>  Mapped PWM ARR value is 5268.500000
-> LDR Sensor value is 514.000000
->  Mapped PWM ARR value is 5268.500000
 -> LDR Sensor value is 514.000000
 ->  Mapped PWM ARR value is 5268.500000
 -> LDR Sensor value is 514.000000
 ->  Mapped PWM ARR value is 5268.500000
 -> LDR Sensor value is 514.000000
0 ->  Mapped PWM ARR value is 5268.500000
0 -> LDR Sensor value is 514.000000
40 ->  Mapped PWM ARR value is 5268.500000
40 -> LDR Sensor value is 514.000000
140 ->  Mapped PWM ARR value is 5268.500000
140 -> LDR Sensor value is 514.000000
.140 ->  Mapped PWM ARR value is 5268.500000
.140 -> LDR Sensor value is 514.000000
.172 ->  Mapped PWM ARR value is 5268.500000
.172 -> LDR Sensor value is 514.000000
2.172 ->  Mapped PWM ARR value is 5268.500000
2.172 -> LDR Sensor value is 514.000000
52.172 ->  Mapped PWM ARR value is 5268.500000
52.172 -> LDR Sensor value is 514.000000
:52.172 ->  Mapped PWM ARR value is 5268.500000
:52.172 -> LDR Sensor value is 514.000000
:52.172 ->  Mapped PWM ARR value is 5268.500000
:52.204 -> LDR Sensor value is 514.000000
2:52.204 ->  Mapped PWM ARR value is 5268.500000
2:52.204 -> LDR Sensor value is 514.000000
```

TC6 –



TC7 –



VII. Sprint Review & Retrospective

- **Completed Stories**: US1, US2, US3
- **Unfinished Stories**: None
- **What Went Well**: Modular code, accurate sensing
- **What to Improve**: Add smoothing filter for ADC noise

VIII. Discussion

Challenges Faced:

- ADC noise
- PWM flicker at low duty cycles

Solutions:

- Range mapper function compensated for minor ADC noise.
- Adjusted PWM frequency to get better PWM resoloution.

VIII. Conclusion

The project successfully met all success criterion: smooth LED brightness control, accurate ADC sampling, and reliable UART logging.

Not a single test case was left invalidated and all testing criterion were met perfectly.

IX. References

- STM32F446RE Datasheet
- STM32CubeIDE Documentation (UM1725 HAL API description
- UART and PWM tutorials from STMicroelectronics and ControllersTECH