Hardware Integration Documentation

Module: LED Blink Using GPIO Output

1.1 Objective

To implement a basic LED blinking application using the GPIO peripheral of the STM32 microcontroller available on the Sourya Board.

The objective of this task is to:

- Verify proper GPIO configuration.
- Confirm successful program execution.
- Demonstrate digital output control using software.
- Provide a visual indication of system operation.

1.2 Components Used

Component	Specification
Microcontroller Board	Sourya Board (STM32 Based)
LED	On-board / External LED
GPIO Pin	PA6 (D0 Pin)
Power Supply	3.3V Regulated Supply
Programming Tool	ST-LINK Programmer

1.3 Electrical Specifications

1.3.1 Operating Parameters

- Operating Voltage: 3.3V
- GPIO Logic Level: 3.3V
- GPIO Mode: Output Push-Pull
- Output State Control: Software Controlled
- Blink Method: GPIO Toggle

GPIO Characteristics

- Logic HIGH ≈ 3.3V
- Logic LOW ≈ 0V
- Maximum GPIO Drive Current: Within STM32 Specifications

1.4 Pin Assignment Summary

Function	STM32 Pin	Sourya Board Pin	Mode
LED Control	PA6	D0	GPIO Output

1.5 Detailed Connection Description

1.5.1 LED Connection

Purpose

The LED provides a visual indication of software execution by periodically switching between ON and OFF states.

Connection

PA6 (D0) → LED

Operating Principle

- PA6 HIGH → LED ON
- PA6 LOW → LED OFF

Configuration

- Pin: PA6
- Mode: GPIO Output Push-Pull
- Pull Configuration: No Pull
- Speed: Low Frequency

1.6 Peripheral Configuration Summary

Peripheral	Configuration
PA6	GPIO Output Push-Pull
Pull-up/Pull-down	None
Output Speed	Low
Initial State	LOW

1.7 Functional Operation Description

The microcontroller continuously executes a loop that toggles the state of the LED connected to PA6.

Operation Sequence:

1. LED is turned ON.

2. System waits for a fixed delay period.
3. LED is turned OFF.
4. System waits for the same delay period.
5. Steps repeat continuously.

This creates a visible blinking effect that confirms:

- Successful firmware execution.
- Proper GPIO functionality.
- Correct hardware connection.

1.8 Program Implementation

Source File

main.c

Code

```
while (1)
{
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
    HAL_Delay(500);
}
```

Alternative implementation:

```
while (1)
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
    HAL_Delay(500);

    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
    HAL_Delay(500);
}
```

1.9 Expected Output

Time Interval	LED State
0 – 500 ms	ON
500 – 1000 ms	OFF
1000 – 1500 ms	ON
1500 – 2000 ms	OFF

The LED continues blinking at approximately 1 Hz frequency.

1.10 Safety and Reliability Considerations

- GPIO current remains within STM32 operating limits.
- No floating pins are used.
- Simple software structure minimizes failure probability.
- Suitable for hardware verification and debugging purposes.

1.11 Applications

- System health indication
- Firmware execution verification
- GPIO testing
- Embedded systems laboratory experiments
- Initial hardware bring-up and debugging

1.12 Future Enhancements

This implementation can be extended to:

- Timer-based LED blinking
- Interrupt-driven status indication
- PWM-based brightness control
- Multi-LED pattern generation
- Error and fault indication systems
- RTOS task status monitoring

Conclusion

The LED Blink application successfully demonstrates GPIO output control on the Sourya Board using PA6 (D0). The periodic toggling of the LED verifies proper hardware configuration, successful firmware execution, and correct operation of the STM32 GPIO subsystem.