Hardware Integration Documentation

Module: Digital Buzzer Control Using GPIO Output

1.1 Objective

To generate an audible alert using an active buzzer connected to the STM32 GPIO pin.

The buzzer is controlled using a simple GPIO HIGH/LOW signal and is used to provide an audio indication of system operation.

1.2 Components Used

Component	Specification
Microcontroller	Sourya Board (3.3V Logic GPIO)
Buzzer	Active Buzzer (3.3V Compatible)
Power Supply	3.3V Regulated Output

1.3 Electrical Specifications

1.3.1 Voltage and Current

• Operating Voltage: 3.3V

• GPIO Output Voltage: 3.3V

• GPIO Output Current: Within STM32 Operating Limits

• Buzzer Type: Active Buzzer (Internal Oscillator)

⚠️ Important Design Constraint

Digital GPIO drive is suitable only for active buzzers.

Passive buzzers require PWM or frequency generation circuitry to produce sound.

1.4 Pin Assignment Summary

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

1.5 Detailed Connection Description

1.5.1 Buzzer Connection

Purpose

Provides an audible indication that the microcontroller is running and executing the program correctly.

Connections

STM32 PA6 (D0) → Buzzer Positive (+)

Buzzer Negative (–) → GND

Operating Principle

• PA6 HIGH → Buzzer ON

• PA6 LOW → Buzzer OFF

Configuration

• PA6 configured as GPIO Output

• Output Type: Push-Pull

• Pull-up/Pull-down: None

1.6 Peripheral Configuration Summary

Peripheral	Setting
PA6	GPIO Output

1.7 Functional Operation Description

1. The microcontroller continuously executes the main loop.
2. PA6 output state is toggled periodically.
3. When PA6 becomes HIGH:
    - Buzzer turns ON and produces sound.
4. When PA6 becomes LOW:
    - Buzzer turns OFF and becomes silent.
5. The process repeats continuously, creating an intermittent beep pattern.

Code

In main.c file inside the while(1):

```
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);
}
```

1.8 Known Limitation (Documented Clearly)

• Digital GPIO provides limited drive strength.

• Sound intensity depends on:

- Buzzer sensitivity
- Supply voltage (3.3V)
- GPIO output capability

• Output volume will be lower than transistor-driven or PWM-based designs.

✅ This behavior is expected and acceptable for basic alert and indication systems.

1.9 Safety and Reliability Considerations

• GPIO current remains within STM32 limits.

• No floating signals are present.

• Common ground reference is maintained.

• Simple GPIO operation results in low failure risk.

1.10 Readiness for Future Enhancement

This design can be upgraded to:

- Transistor-driven buzzer control
- PWM-based tone generation
- Variable frequency alarm patterns
- Interrupt-driven alert generation
- Event-based notification systems
- Multi-level warning and alarm indications

Conclusion

The Digital Buzzer Control application successfully demonstrates GPIO output control on the Sourya Board using PA6 (D0). The periodic activation and deactivation of the buzzer verifies proper hardware configuration, successful firmware execution, and correct operation of the STM32 GPIO subsystem.