ARM Microcontroller Internship Programme
Trainer: *Kiran Jaybhave*
Sunbeam – Industrial Training Programme
Sunbeam Pune (Hinjawadi)

STM32F407VG – GPIO
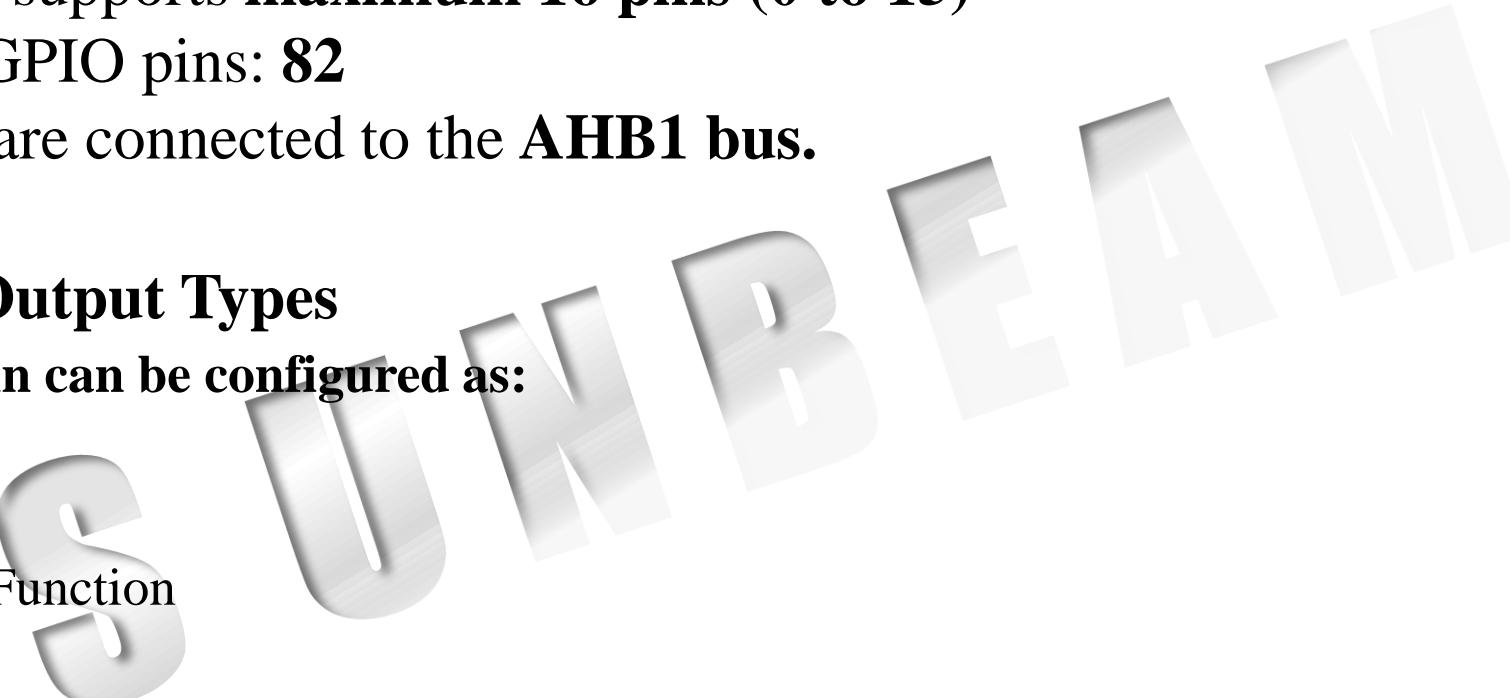
❑GPIO Basics

- **GPIO (General Purpose Input Output)** allows the microcontroller to interact with external hardware such as LEDs, switches, buzzers, etc.
  GPIO programming is the **first step in embedded systems** because it directly connects software to hardware.

- **Multiplexed GPIO Pins**

- GPIO pins in STM32 are **multiplexed**:

  - One pin can act as GPIO

  - Or be used for alternate functions like UART, SPI, I2C, etc.

  - The required function is selected using configuration registers.

GPIO Ports in STM32F407VG

- STM32F407VG has **GPIO ports from GPIOA to GPIOI**
- Each GPIO port supports **maximum 16 pins (0 to 15)**
- Total available GPIO pins: **82**
- All GPIO ports are connected to the **AHB1 bus.**

- **GPIO Input / Output Types**
  - **Each GPIO pin can be configured as:**
    - Input
    - Output
    - Alternate Function
    - Analog
  - **GPIO pins also support**:
    - External interrupt capability
    - Configurable speed
    - Internal pull-up or pull-down resistors

- **Output Driver Types**
  - GPIO output pins support two types:
  - **Push-Pull**: Drives both HIGH and LOW (used for LEDs)
  - **Open-Drain**: Drives LOW only, needs external pull-up (used in I²C)
  - The output type is selected using the **OTYPER register**.

- **Pull-up / Pull-down Configuration**
  - GPIO pins can be configured with:
  - No pull-up / pull-down
  - Pull-up resistor
  - Pull-down resistor

- **GPIO Speed Configuration**
  - GPIO output speed controls how fast the pin switches:
    - Low
    - Medium
    - High
    - Very High
  - Higher speed gives faster transitions but increases power consumption and EMI.
- **GPIO Registers Overview**
  - GPIO functionality is controlled using registers:
    - MODER – selects pin mode
    - OTYPER – selects output type
    - OSPEEDR – selects output speed
    - PUPDR – selects pull-up / pull-down
    - IDR – reads input
    - ODR – writes output
    - BSRR – atomic bit set/reset

❖MODER – GPIO Mode Register

- Each GPIO pin uses **2 bits** in the MODER register
- Mode selection:
- Example:
  Pin 12 → MODER bits **24 and 25**

Bits	Mode
00	Input
01	Output
10	Alternate Function
11	Analog

❖OTYPER – Output Type Register

- One bit per GPIO pin
- Controls electrical behavior of output pin
- Example:
  Pins 12–15 → bits 12–15

Value	Type
0	Push-Pull
1	Open-Drain

❖ OSPEEDR – Output Speed Register

- Two bits per pin control speed
- Example:
  Pins 12–15 → bits 24–31

Bits	Speed
00	Low
01	Medium
10	High
11	Very High

❖ PUPDR – Pull-up / Pull-down Register

- Two bits per pin configure internal resistors
- Used mainly for input pins.

Bits	Configuration
00	No pull
01	Pull-up
10	Pull-down

❖ IDR – Input Data Register

- Used to **read the current state** of GPIO input pins
- Each bit corresponds to one pin
- Example:
  If bit = 1 → pin is HIGH
  If bit = 0 → pin is LOW

❖ ODR – Output Data Register

- Used to **set or clear GPIO output pins**
  - Writing 1 sets pin HIGH
  - Writing 0 sets pin LOW
- For STM32F407 Discovery LEDs:
  - Modify **bits 12 to 15**

❖ BSRR – Bit Set / Reset Register

- Allows **atomic pin control**
- Lower 16 bits → Set pin
- Upper 16 bits → Reset pin
- Prevents read-modify-write problems.

❖ Enabling GPIO Clock (RCC)
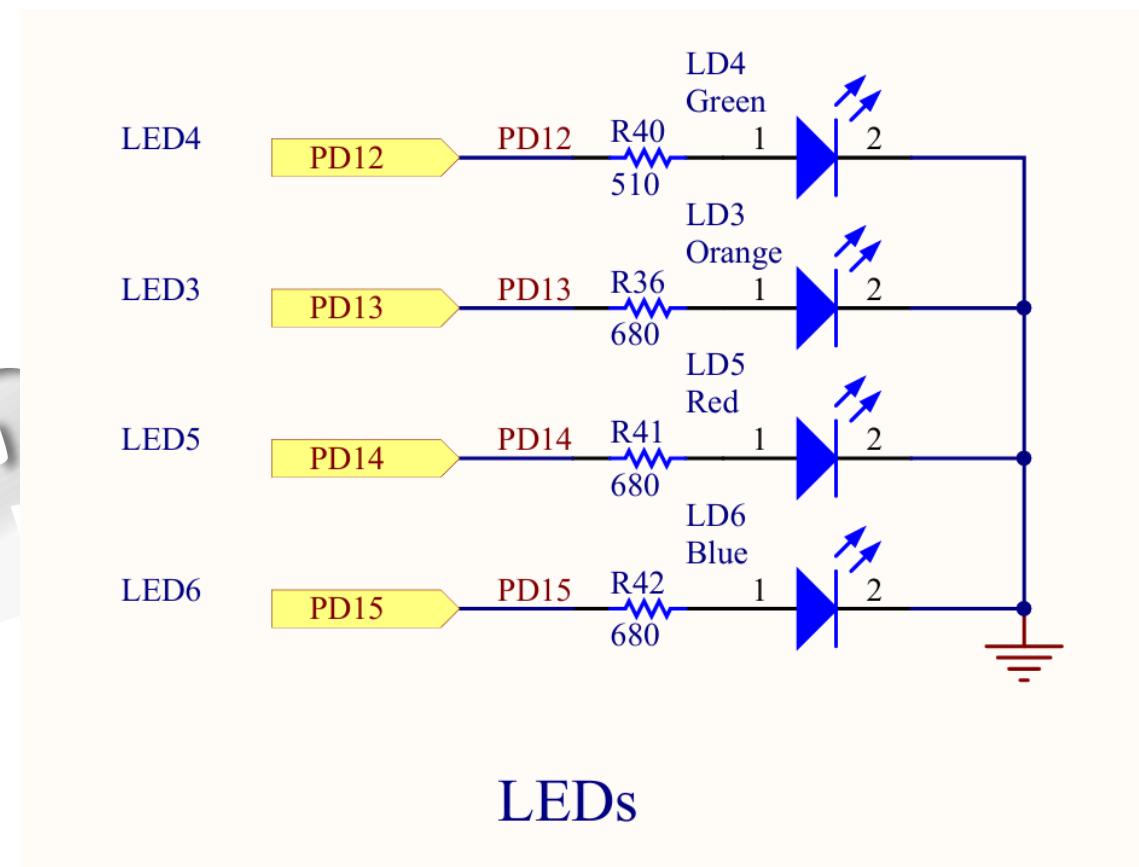
- GPIO peripherals need clock to operate
- Clock is enabled using **AHB1ENR register**
- Example:
- RCC->AHB1ENR |= (1 << 3); // Enable GPIOD
- Without clock → GPIO will not work.

- **GPIO Programming Steps (Output)**
  - Enable GPIO clock
  - Configure pin mode as output
  - Configure output type
  - Configure output speed
  - Configure pull-up / pull-down
  - Set or clear GPIO pin

LEDs

Thank you!

Kiran Jaybhave

email – kiran.jaybhave@sunbeaminfo.com