STM32F4 Development Board Familiarization

Instructions of setting up the Blinky app process :-

Step 1: Create a New Project

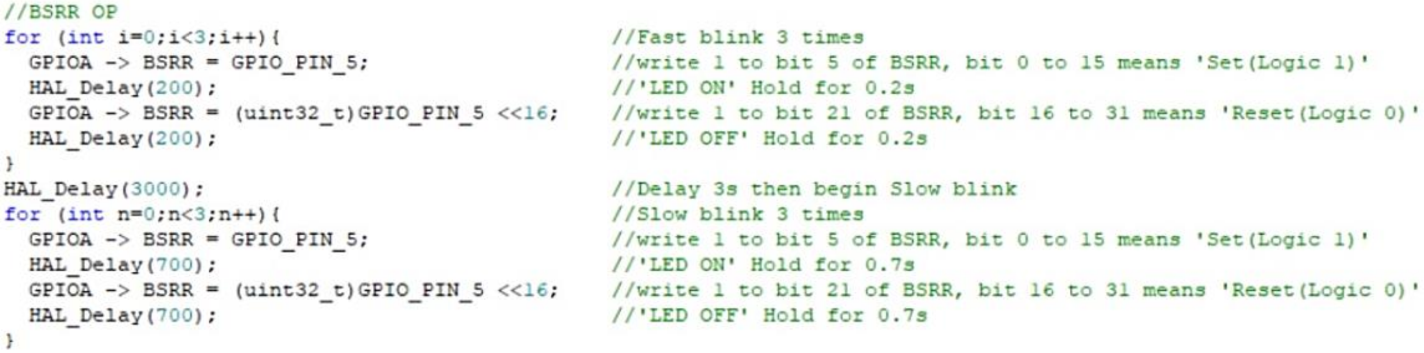
1. Open STM32CubeMX → New Project
2. Select STM32F446RETx
3. Enable GPIOA PA5 (GPIO_Pin_5) as an output. Output level is set to low (logic ‘0’).
4. Check clock status at configuration setup.
5. Generate code for MDK-ARM (Keil)

Step 2: Configure LED Pin

1. Initialize HAL_Init()
2. Check HAL_Init() & SystemClock_Config().
3. Verifying HAL_GPIO_WritePin is functioning.
4. In main.c, ensure GPIOA clock is enabled.

Step 3: Write the Blink Logic

1. Paste the below inside While(1){}



```
//BSRR OP
for (int i=0;i<3;i++){                        //Fast blink 3 times
  GPIOA -> BSRR = GPIO_PIN_5;                 //write 1 to bit 5 of BSRR, bit 0 to 15 means 'Set(Logic 1)'
  HAL_Delay(200);                             //'LED ON' Hold for 0.2s
  GPIOA -> BSRR = (uint32_t)GPIO_PIN_5 <<16;  //write 1 to bit 21 of BSRR, bit 16 to 31 means 'Reset(Logic 0)'
  HAL_Delay(200);                             //'LED OFF' Hold for 0.2s
}
HAL_Delay(3000);                              //Delay 3s then begin Slow blink
for (int n=0;n<3;n++){                        //Slow blink 3 times
  GPIOA -> BSRR = GPIO_PIN_5;                 //write 1 to bit 5 of BSRR, bit 0 to 15 means 'Set(Logic 1)'
  HAL_Delay(700);                             //'LED ON' Hold for 0.7s
  GPIOA -> BSRR = (uint32_t)GPIO_PIN_5 <<16;  //write 1 to bit 21 of BSRR, bit 16 to 31 means 'Reset(Logic 0)'
  HAL_Delay(700);                             //'LED OFF' Hold for 0.7s
}
```

Step 4: Build and Flash

1. Click **Build** or press Ctrl+B
2. Translate (Ctrl+F7) and confirm no Error and no Warning.
3. Build (F7) to creating hex file.
4. Connect the board via USB.
5. Download ~code to flash memory.

Step 5: Demonstration

The LED on **PA5** will blink 3 times quickly, wait 3 seconds, then blink 3 times slowly.