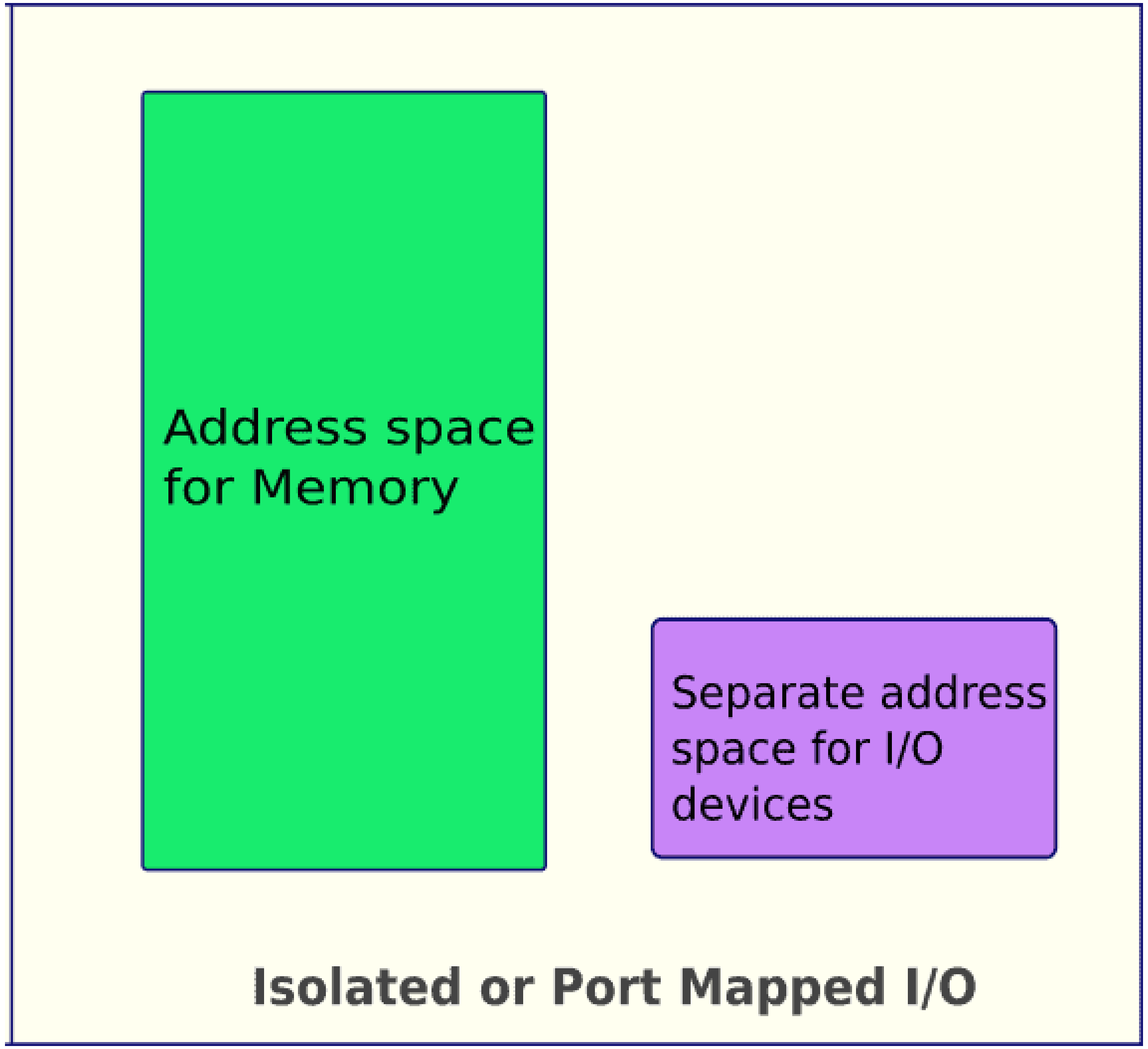
Thynk Loop

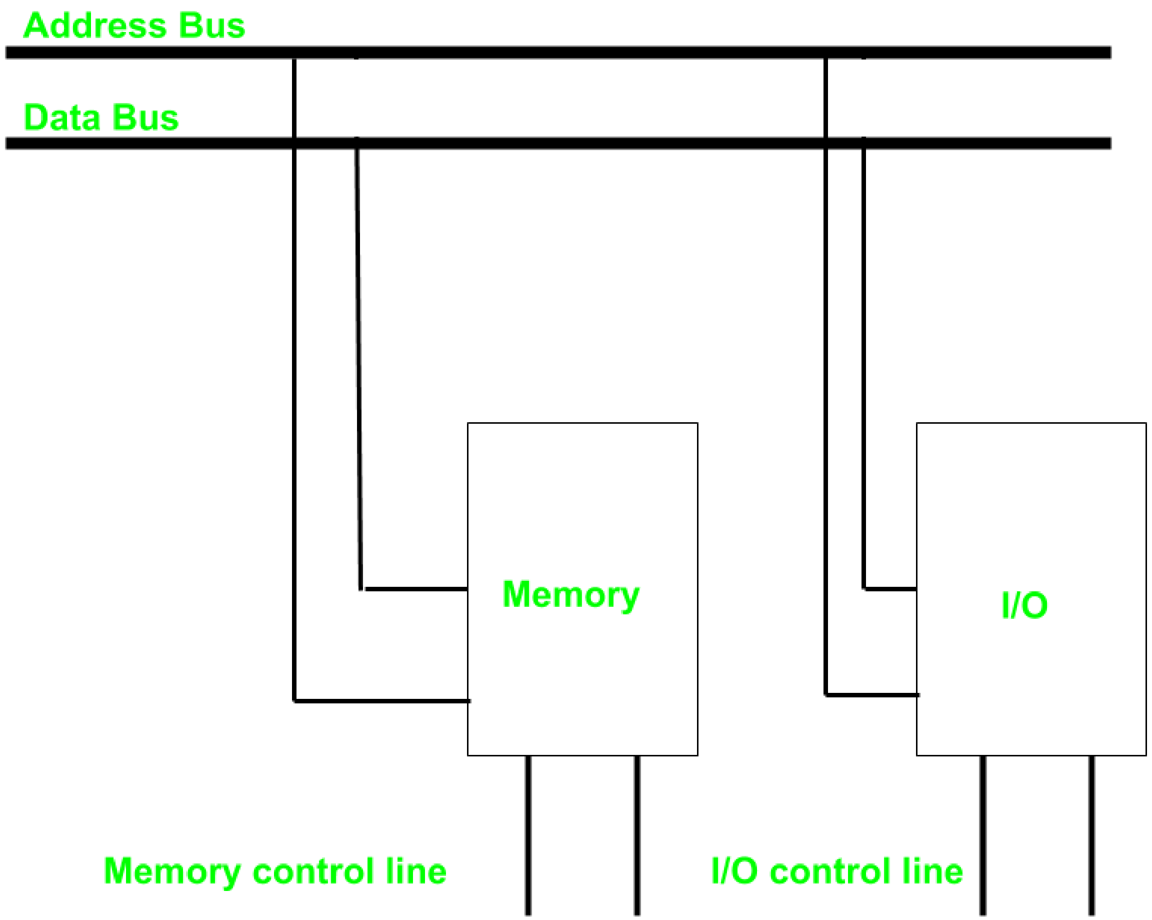
Module 7

Memory Mapping

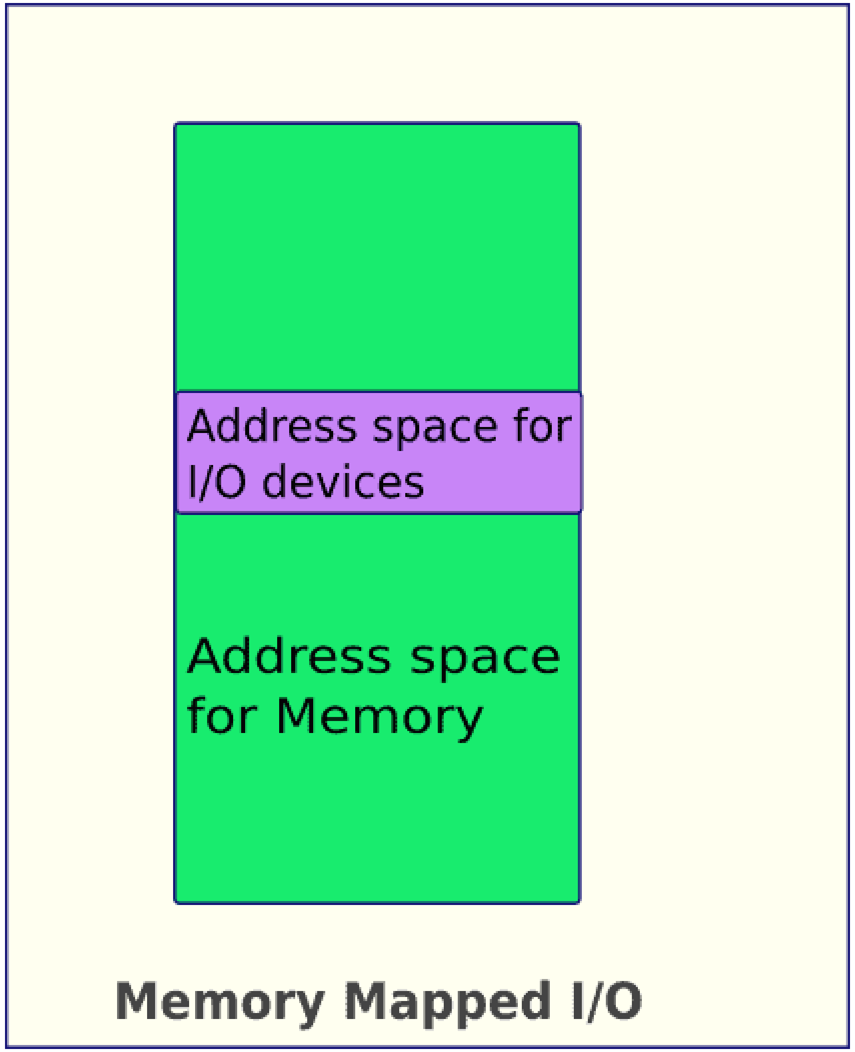
Port Mapped I/O


Address space for Memory
Separate address space for I/O devices
Isolated or Port Mapped I/O

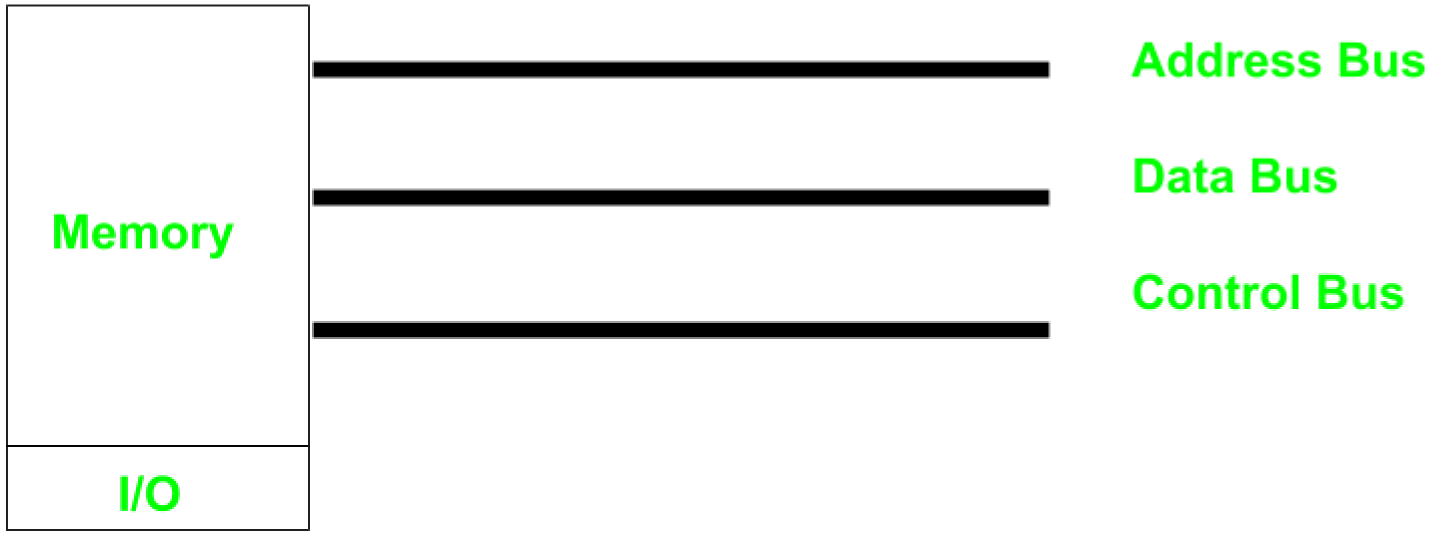
How its accessed



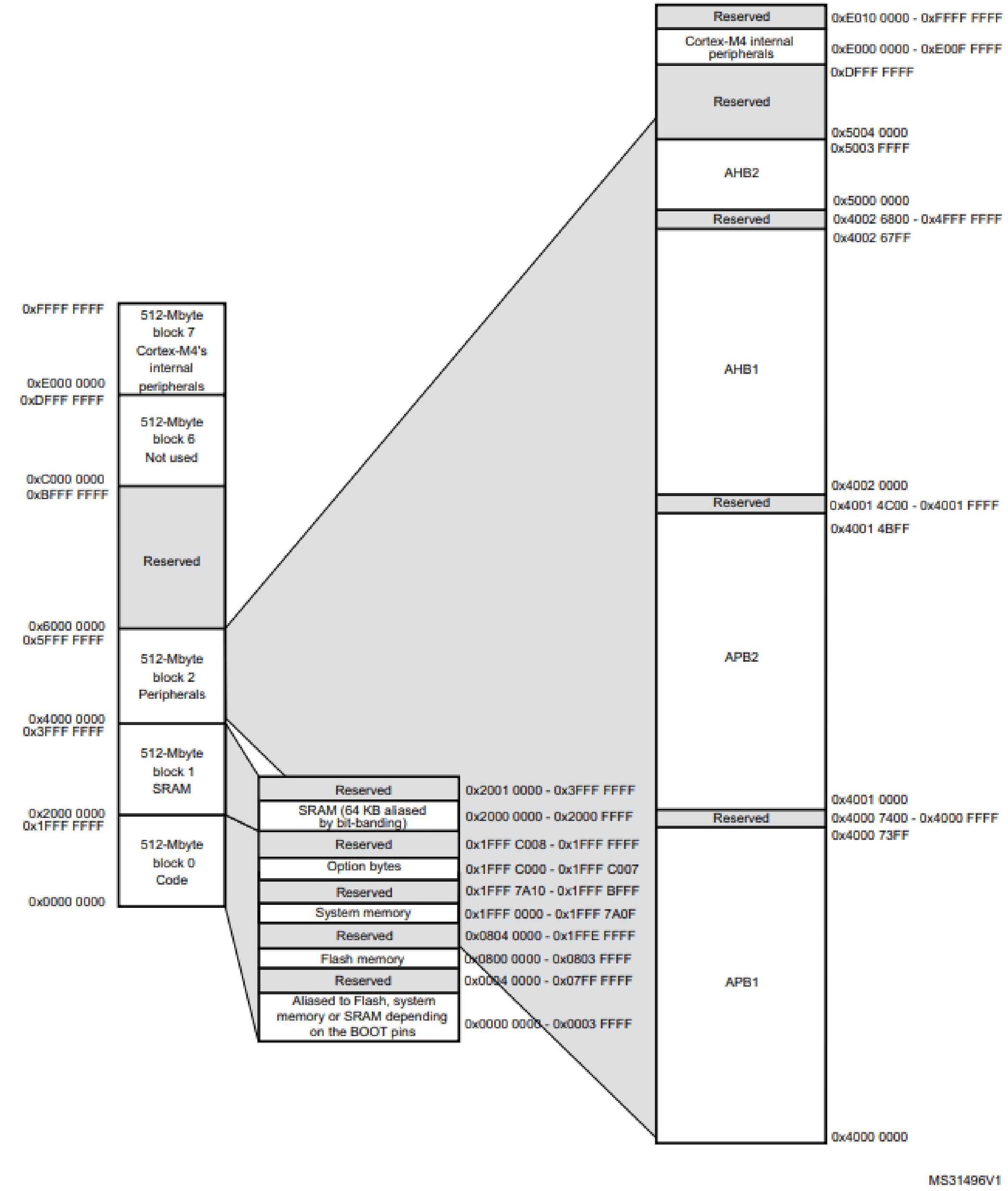
Memory Mapped I/O



How its accessed



Memory Mapping in STM32



Address	Block
0xFFFF FFFF 0xE000 0000	512-Mbyte block 7 Cortex-M4's internal peripherals
0xDFFF FFFF 0xC000 0000	512-Mbyte block 6 Not used
0xBFFF FFFF 0x6000 0000	Reserved
0x5FFF FFFF 0x4000 0000	512-Mbyte block 2 Peripherals
0x3FFF FFFF 0x2000 0000	512-Mbyte block 1 SRAM
0x1FFF FFFF 0x0000 0000	512-Mbyte block 0 Code

Region	Address
Reserved	0x2001 0000 - 0x3FFF FFFF
SRAM (64 KB aliased by bit-banding)	0x2000 0000 - 0x2000 FFFF
Reserved	0x1FFF C008 - 0x1FFF FFFF
Option bytes	0x1FFF C000 - 0x1FFF C007
Reserved	0x1FFF 7A10 - 0x1FFF BFFF
System memory	0x1FFF 0000 - 0x1FFF 7A0F
Reserved	0x0804 0000 - 0x1FFE FFFF
Flash memory	0x0800 0000 - 0x0803 FFFF
Reserved	0x0004 0000 - 0x07FF FFFF
Aliased to Flash, system memory or SRAM depending on the BOOT pins	0x0000 0000 - 0x0003 FFFF

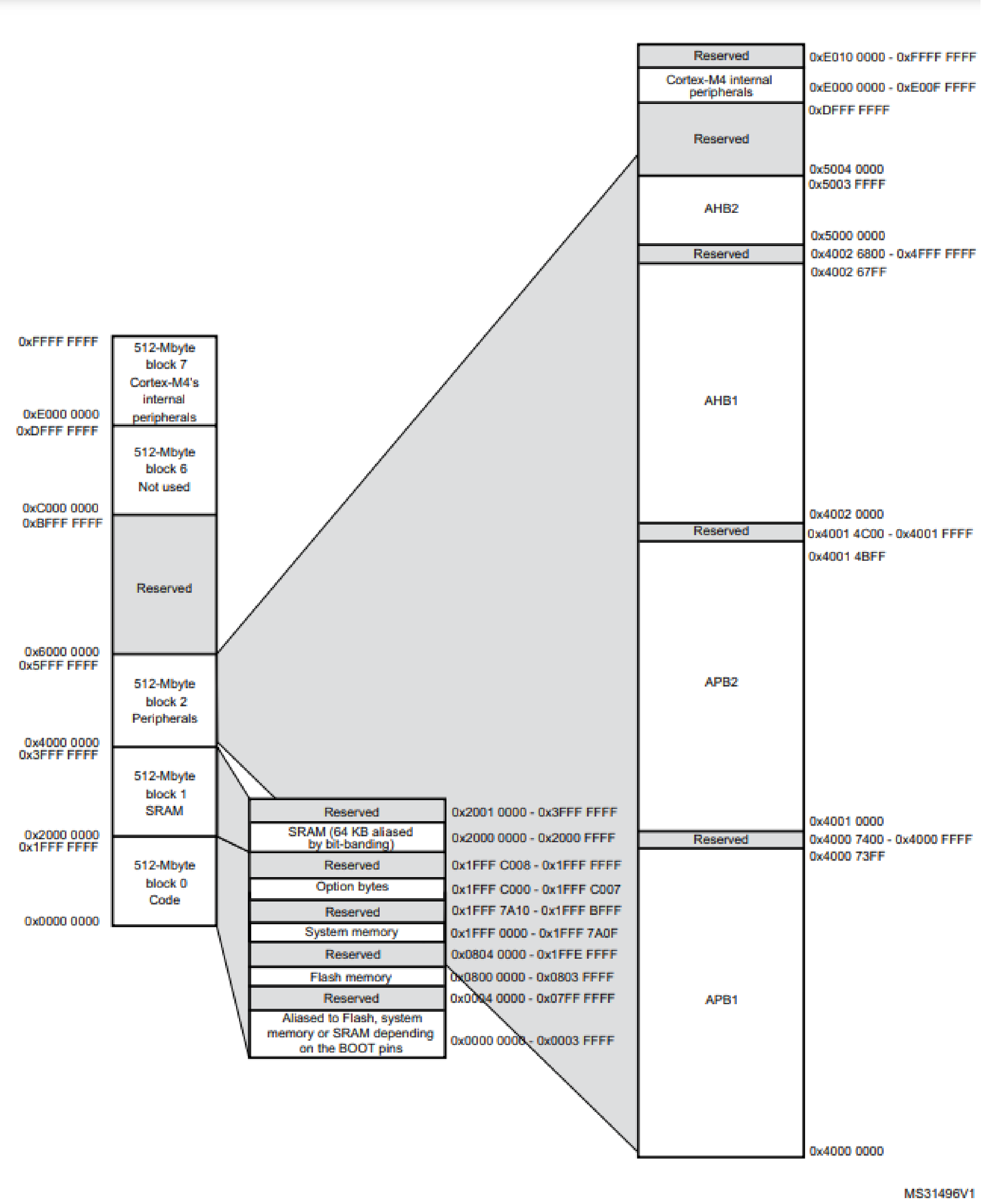
Region	Address
Reserved	0xE010 0000 - 0xFFFF FFFF
Cortex-M4 internal peripherals	0xE000 0000 - 0xE00F FFFF
Reserved	0xDFFF FFFF 0x5004 0000
AHB2	0x5003 FFFF 0x5000 0000
Reserved	0x4002 6800 - 0x4FFF FFFF
AHB1	0x4002 67FF 0x4002 0000
Reserved	0x4001 4C00 - 0x4001 FFFF
APB2	0x4001 4BFF 0x4001 0000
Reserved	0x4000 7400 - 0x4000 FFFF
APB1	0x4000 73FF 0x4000 0000

MS31496V1

GPIO Register Mapping

Datasheet

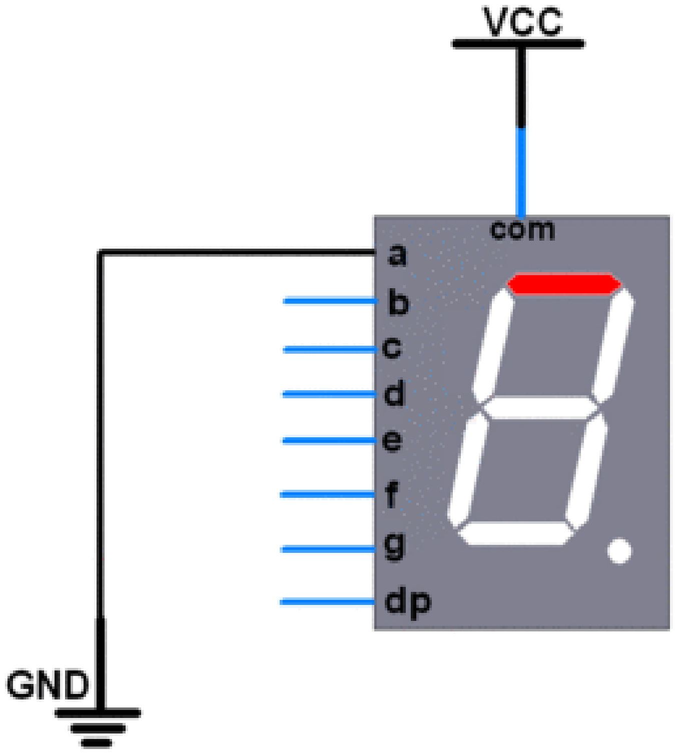
- PG - 50



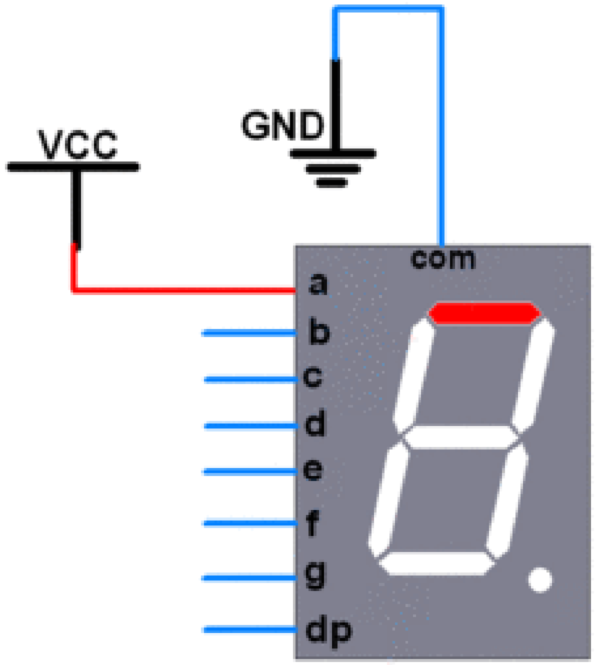
0xFFFF FFFF
512-Mbyte block 7 Cortex-M4's internal peripherals
0xE000 0000
0xDFFF FFFF
512-Mbyte block 6 Not used
0xC000 0000
0xBFFF FFFF
Reserved
0x6000 0000
0x5FFF FFFF
512-Mbyte block 2 Peripherals
0x4000 0000
0x3FFF FFFF
512-Mbyte block 1 SRAM
0x2000 0000
0x1FFF FFFF
512-Mbyte block 0 Code
0x0000 0000
Reserved
0x2001 0000 - 0x3FFF FFFF
SRAM (64 KB aliased by bit-banding)
0x2000 0000 - 0x2000 FFFF
Reserved
0x1FFF C008 - 0x1FFF FFFF
Option bytes
0x1FFF C000 - 0x1FFF C007
Reserved
0x1FFF 7A10 - 0x1FFF BFFF
System memory
0x1FFF 0000 - 0x1FFF 7A0F
Reserved
0x0804 0000 - 0x1FFE FFFF
Flash memory
0x0800 0000 - 0x0803 FFFF
Reserved
0x0004 0000 - 0x07FF FFFF
Aliased to Flash, system memory or SRAM depending on the BOOT pins
0x0000 0000 - 0x0003 FFFF
Reserved
0xE010 0000 - 0xFFFF FFFF
Cortex-M4 internal peripherals
0xE000 0000 - 0xE00F FFFF
0xDFFF FFFF
Reserved
0x5004 0000
0x5003 FFFF
AHB2
0x5000 0000
Reserved
0x4002 6800 - 0x4FFF FFFF
0x4002 67FF
AHB1
0x4002 0000
Reserved
0x4001 4C00 - 0x4001 FFFF
0x4001 4BFF
APB2
0x4001 0000
Reserved
0x4000 7400 - 0x4000 FFFF
0x4000 73FF
APB1
0x4000 0000
MS31496V1

7-Segment Display

Types of 7-Segment Display

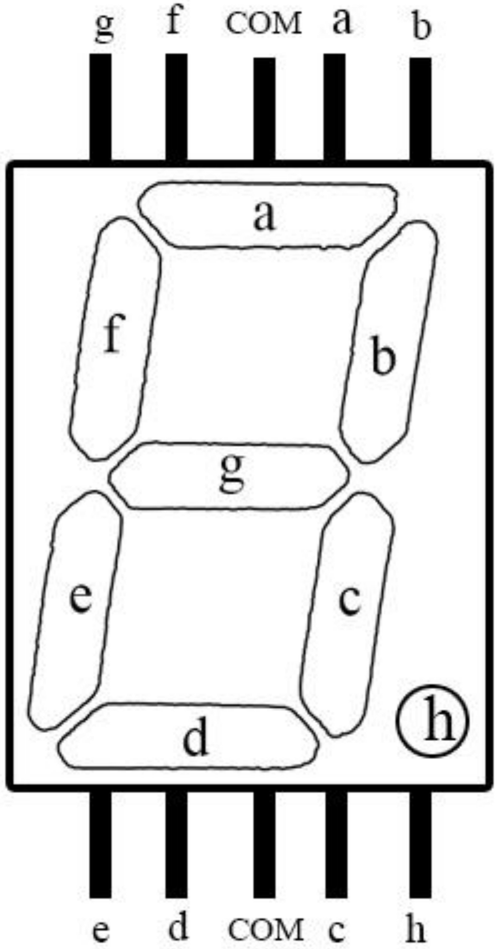


Common Anode



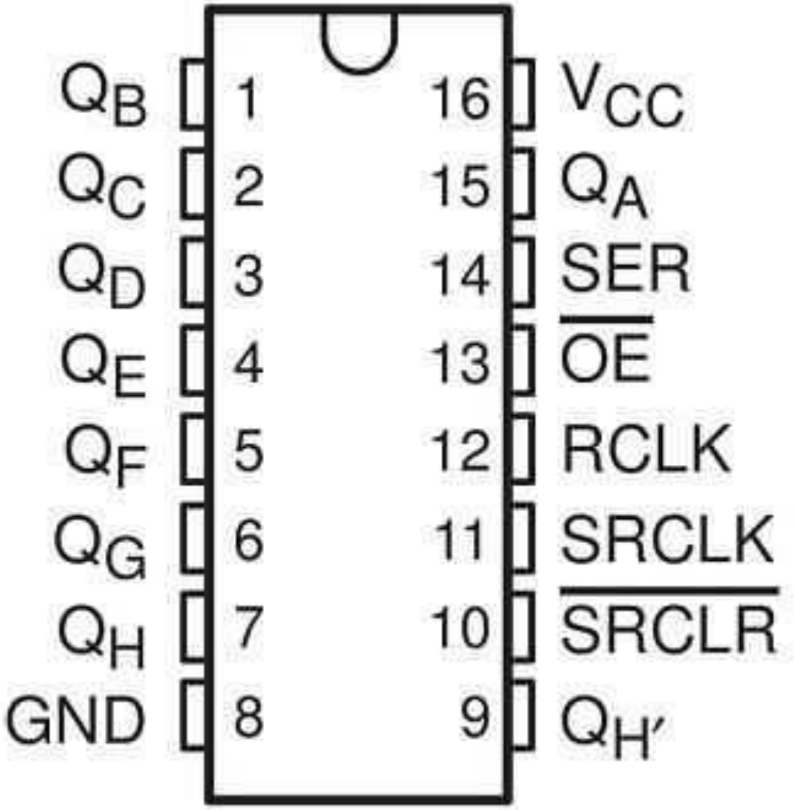
Common Cathode

Pin Outs



Shift Register – 74HC595

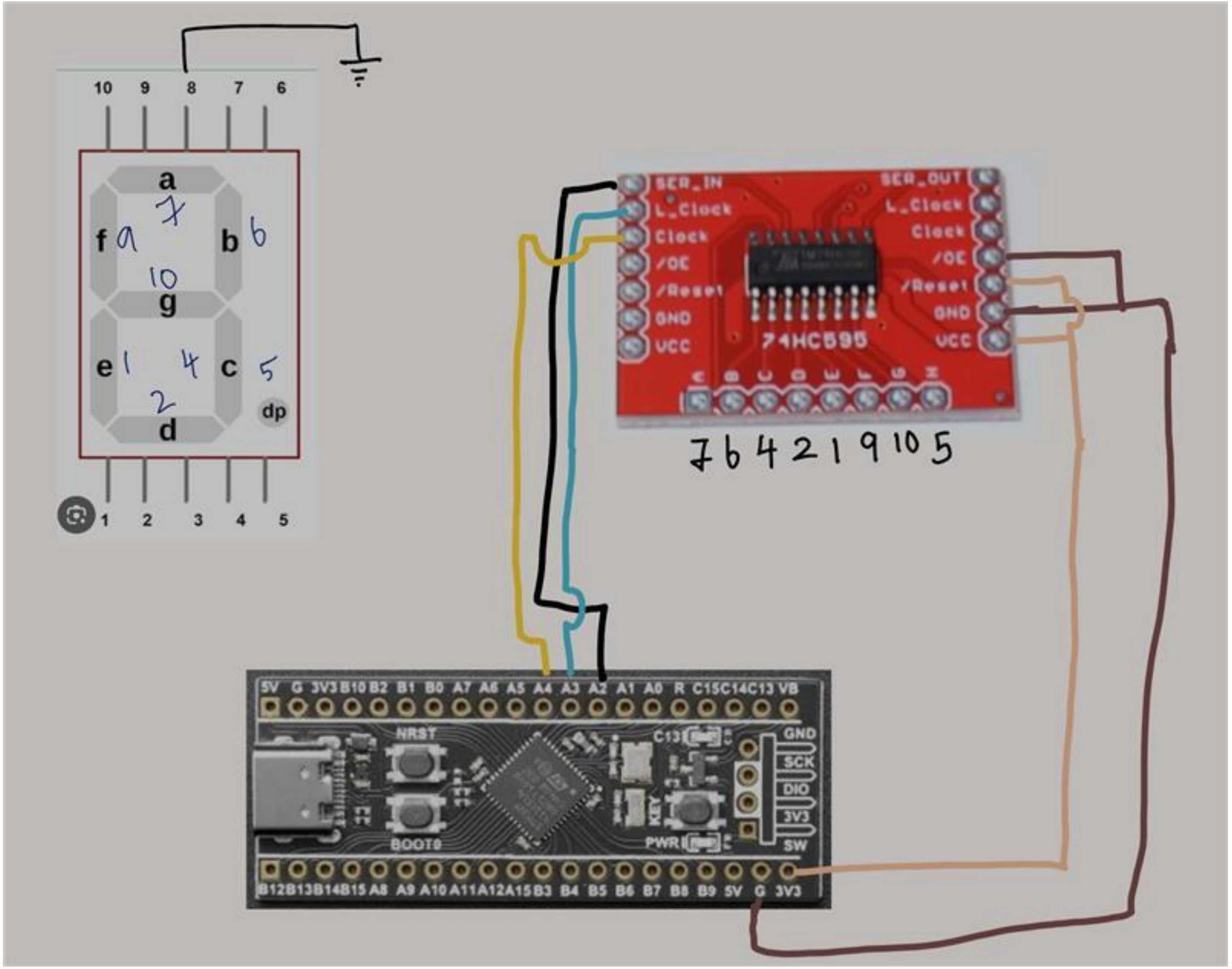
Types of 7-Segment Display



LAB - 1

Drive 7-Segment display using STM32 through shift register

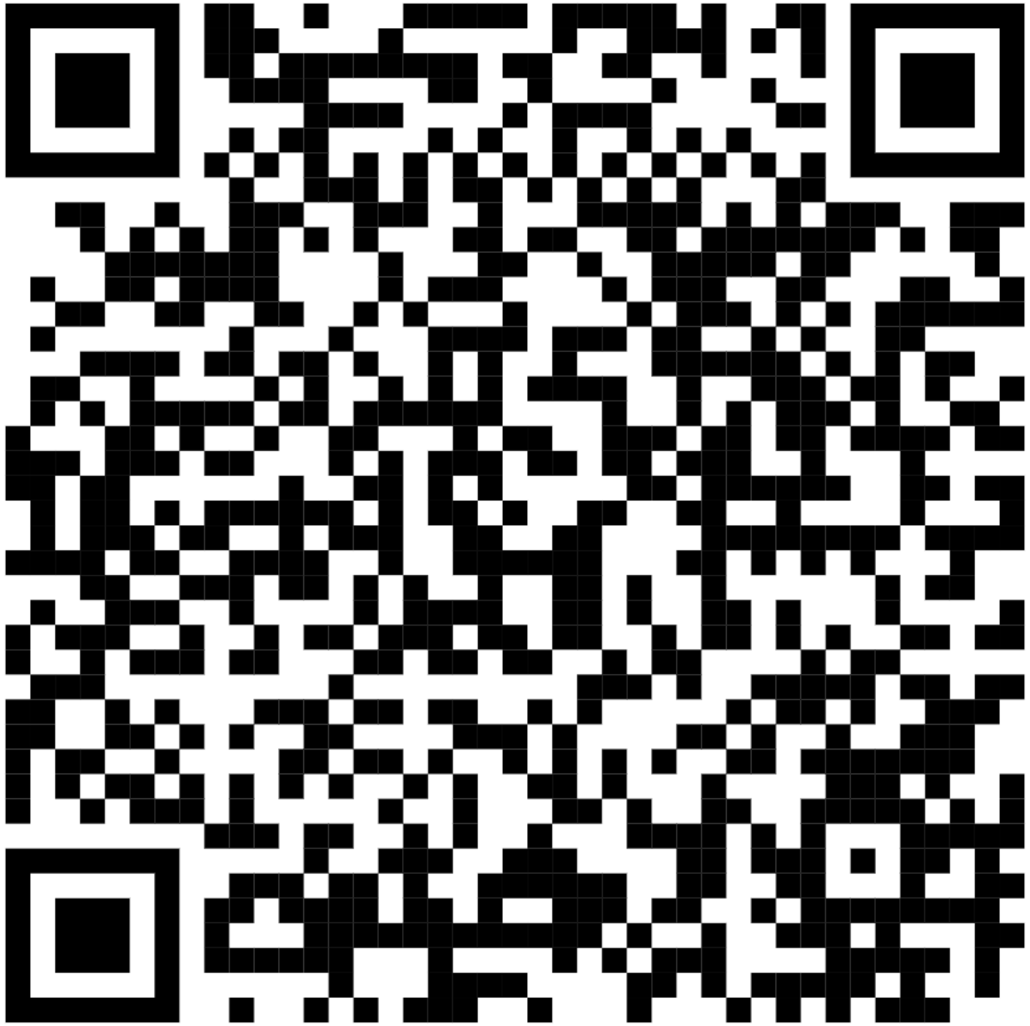
Pin Connections



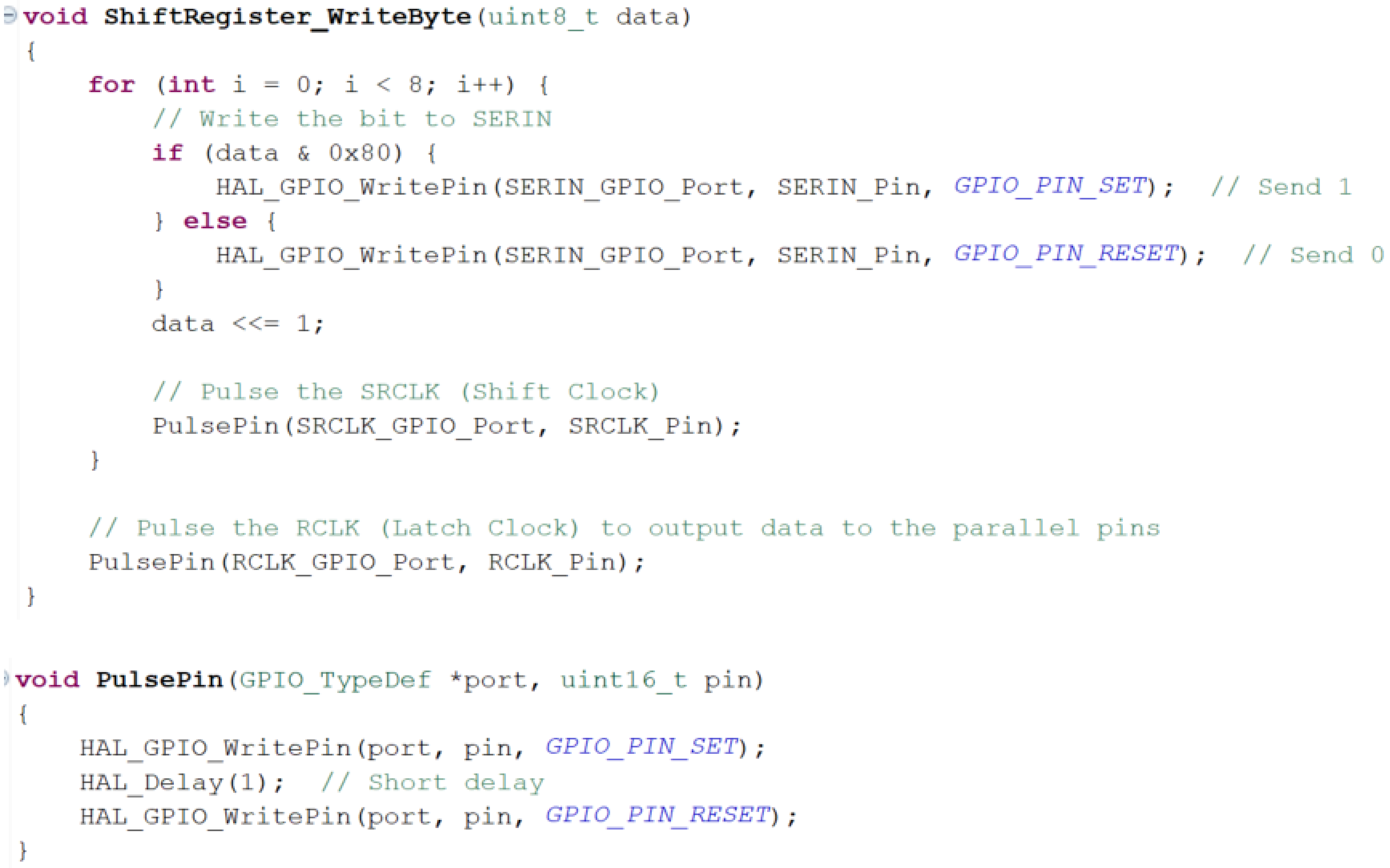
Pseudo Code

```
1. Init
    SERIN  = 0
    SRCLK = 0
    RCCLK = 0

2. Setting the data pin
    SERIN = 1

3 Creating the clock
    SRCLK = 0
    Delay(1)  in ms
    SRCLK = 1
    Delay(1)  in ms
    SRCLK = 0
    SERIN = 0

4 Setting the output latch
    RCLK = 0
    Delay(1)  in ms
    RCLK = 1
    Delay(1)  in ms
    RCLK = 0
```

GitHub Repo



Code Snippets – Shift Register Write



```c
void ShiftRegister_WriteByte(uint8_t data)
{
    for (int i = 0; i < 8; i++) {
        // Write the bit to SERIN
        if (data & 0x80) {
            HAL_GPIO_WritePin(SERIN_GPIO_Port, SERIN_Pin, GPIO_PIN_SET);  // Send 1
        } else {
            HAL_GPIO_WritePin(SERIN_GPIO_Port, SERIN_Pin, GPIO_PIN_RESET);  // Send 0
        }
        data <<= 1;

        // Pulse the SRCLK (Shift Clock)
        PulsePin(SRCLK_GPIO_Port, SRCLK_Pin);
    }

    // Pulse the RCLK (Latch Clock) to output data to the parallel pins
    PulsePin(RCLK_GPIO_Port, RCLK_Pin);
}

void PulsePin(GPIO_TypeDef *port, uint16_t pin)
{
    HAL_GPIO_WritePin(port, pin, GPIO_PIN_SET);
    HAL_Delay(1);  // Short delay
    HAL_GPIO_WritePin(port, pin, GPIO_PIN_RESET);
}
```