Relay Home

Manual

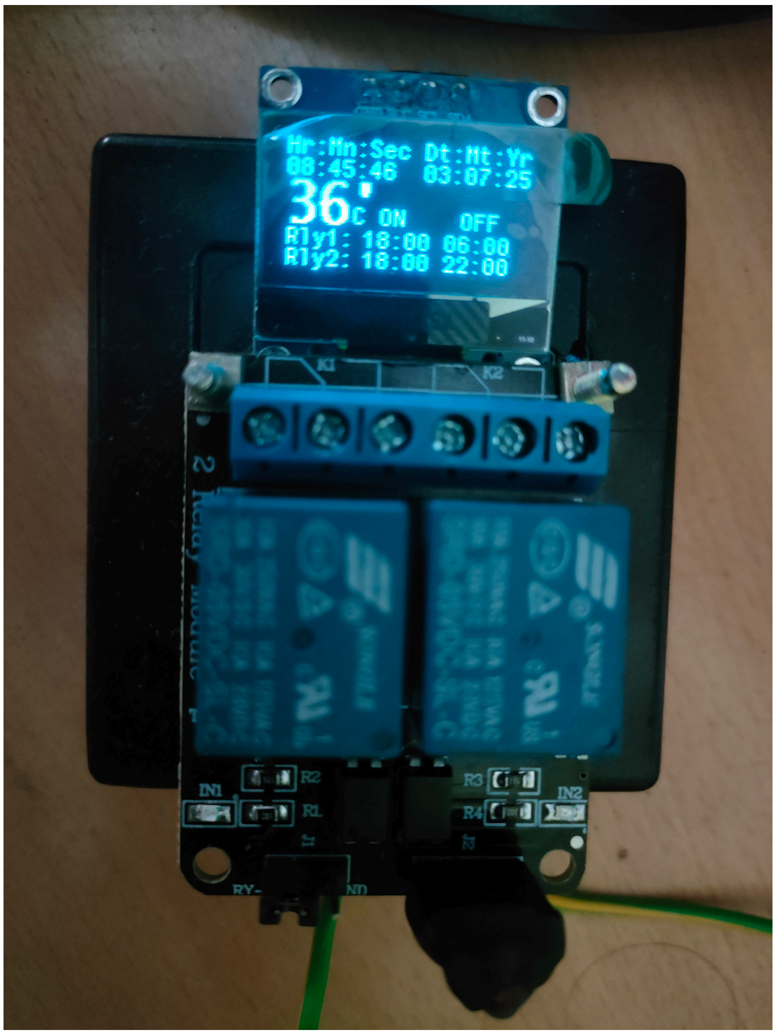
Ver 1, July 2025

Table of Contents

Chapter 1. Ordering Information

S.no	Components	Comment
1	STM32F103C8T6	`BluePill Board (150Rs)`
2	DS3231	`RTC with EEPROM Module (100Rs)`
3	SSD1306	`0.96Inch OLED Display (150Rs)`
4	Power Supply	`5V 1A (35Rs) or 5V 2A (80Rs)`
5	Relay Module	`5V, Duel Channel (70Rs)`
6	Enclosure	`PEM02, 60x55x24mm (Inner Dimension) (60Rs)`
7	Soldering Wire	`24AWG Heat Resistant Silicone (20Rs/Meter)`
8	Additional Information	• MCU uses I²C Communication with RTC, EEPROM & Display • DS3231 Runs the Clock also measures the Temperature • EEPROM Stores the Relay ON/OFF Timings • SSD1306 Displays the Required Information • MCU Uses USB Rx Communication to Receive the Inputs • MCU Uses USB Tx Communication to Transmit Information • Power Supply 5V 1A is bought from "ifuturetech.org" • Power Supply 5V 2A is bought from "Quartz Components" • Soldering Wire is Very Good quality from "ifuturetech.org" • Enclosure (PEM02)is bought from "www.probots.co.in" • Rubber Sleeve, JST, RMC, FRC wires from "Componentstree.com" • Other Items like STM, RTC etc can be ordered from "Robu.in" `"Electron Components","Evelta","Etstore","Ktron"`

Chapter 2. Product



- Sun Mon Tue Wed Thu Fri Sat

```
SetTime= 07:20:00 Sat 28:06:25
```

```
Relay1= On 18:00 Off 06:00
```

```
Relay2= On 18:00 Off 22:00
```

Chapter 3. Introduction

This product is developed as an Hobby for Home use. I have enquired about the similar poduct in the Market, Its cost is 2000Rs. The Product is without Display and it has Only One Output.

Advantages

- The cost of Making this product is around 700Rs.
- It has duel Relay and Display with Additional Features.
- The cost can be even reduced without Display.

3.1. Application

The One of the Use case is mentioned Below.

Purpose	Value
Automatic TurnOn and TurnOFF	The Required Time is set

3.2. Commands

```
The Below Command has to be sent serially, to set the Values
```

Sun Mon Tue Wed Thu Fri Sat

Set Time Command. Sets The RTC Time.

Command	`SetTime= 07:20:00 Sat 28:06:25`
Response	`USB_SetTime_Success `

Sun Mon Tue Wed Thu Fri Sat

Set Relay Time Command. Sets The Relay ON Time and OFF Time.

Command	`Relay1= On 18:00 Off 06:00`
Response	`USB_Relay1_SetTime_Success `
Command	`Relay2= On 18:00 Off 22:00`
Response	`USB_Relay2_SetTime_Success `

3.3. Recommended Operating Range

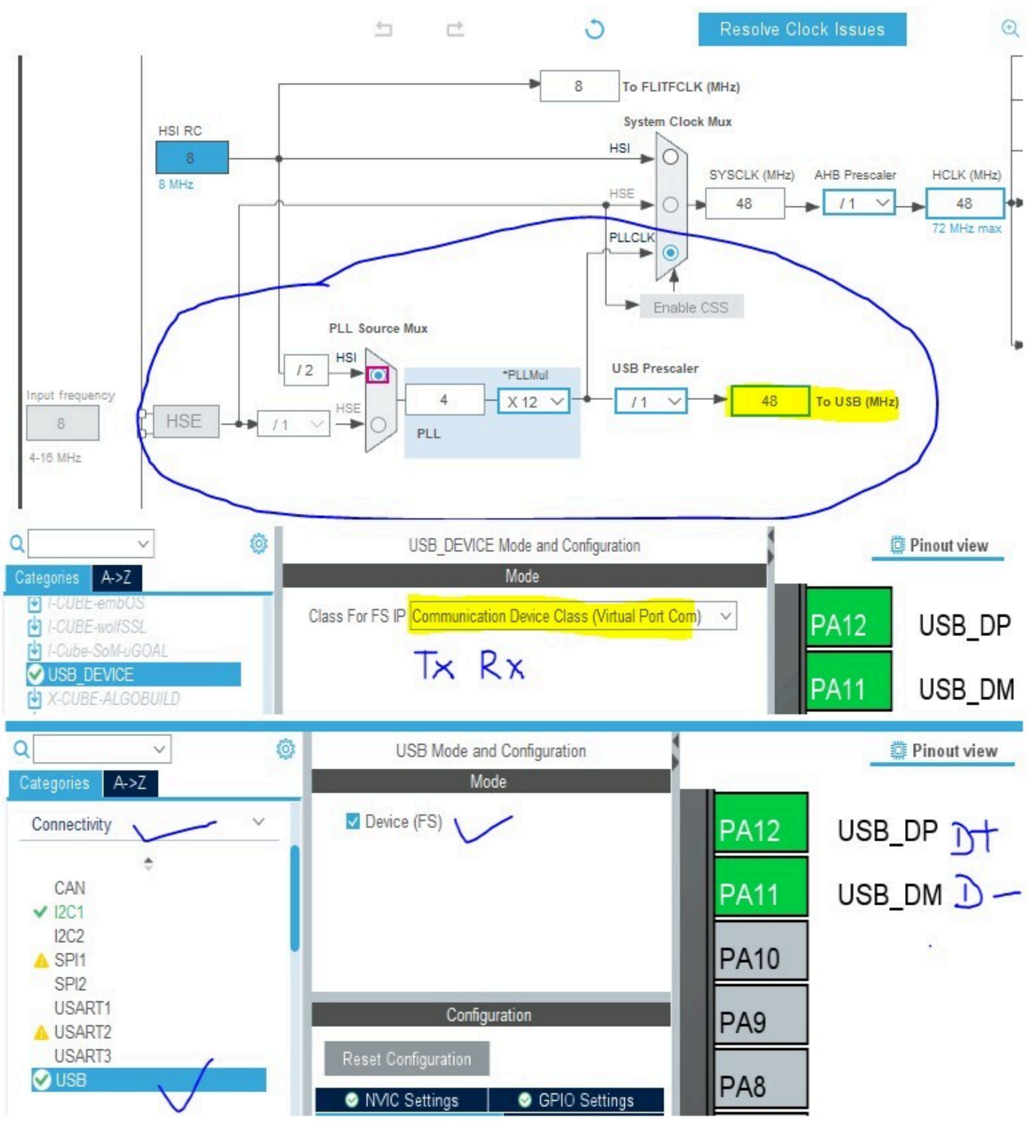
Parameter	Minimum	Typical	Maximum	Units
Voltage	4.5	5	5.5	V

Chapter 4. USB Protocol

Link to Merge Image https://image.pi7.org/join-images-online

Link to convert jpeg to jpg https://cloudconvert.com/jpeg-to-jpg

4.1. Cube IDE settings

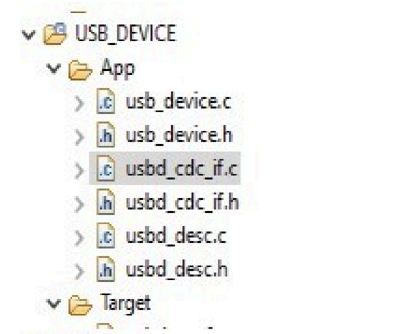


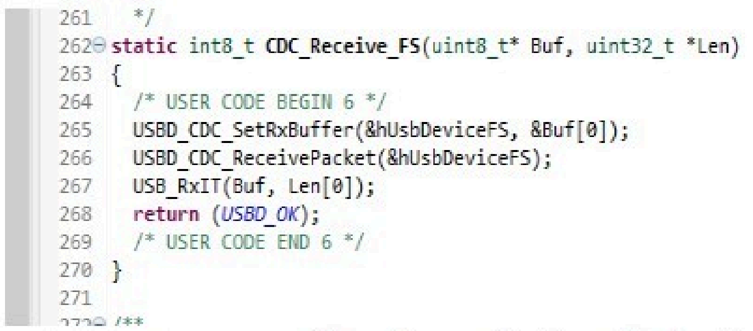
Class For FS IP Communication Device Class (Virtual Port Com)

Tx Rx

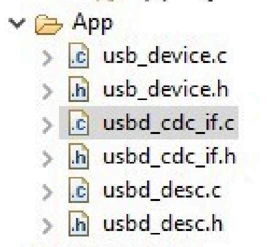
Device (FS) ✓

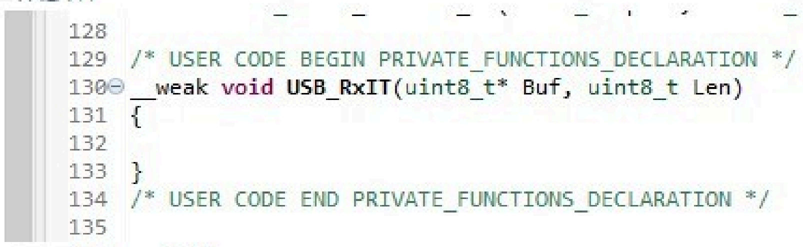
4.2. Code Changes


```
USB_DEVICE
  App
    usb_device.c
    usb_device.h
    usbd_cdc_if.c
    usbd_cdc_if.h
    usbd_desc.c
    usbd_desc.h
  Target
```


```
  */
static int8_t CDC_Receive_FS(uint8_t* Buf, uint32_t *Len)
{
  /* USER CODE BEGIN 6 */
  USBD_CDC_SetRxBuffer(&hUsbDeviceFS, &Buf[0]);
  USBD_CDC_ReceivePacket(&hUsbDeviceFS);
  USB_RxIT(Buf, Len[0]);
  return (USBD_OK);
  /* USER CODE END 6 */
}

/**
```

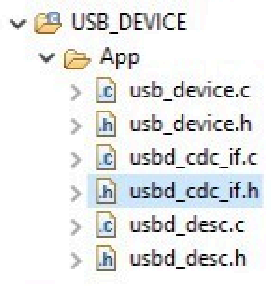

```
App
  usb_device.c
  usb_device.h
  usbd_cdc_if.c
  usbd_cdc_if.h
  usbd_desc.c
  usbd_desc.h
```

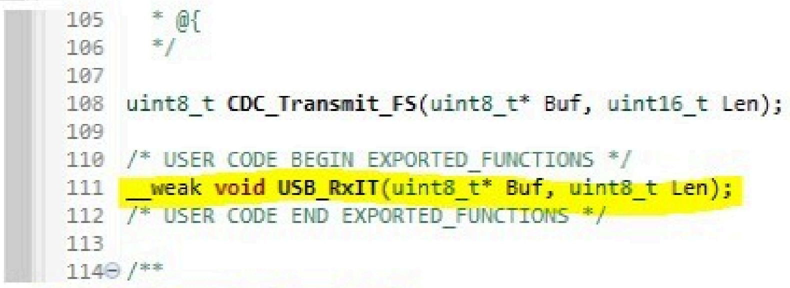

```

/* USER CODE BEGIN PRIVATE_FUNCTIONS_DECLARATION */
__weak void USB_RxIT(uint8_t* Buf, uint8_t Len)
{

}
/* USER CODE END PRIVATE_FUNCTIONS_DECLARATION */

```

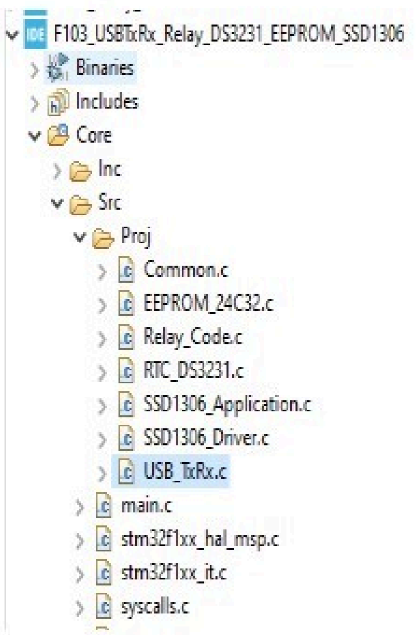

```
USB_DEVICE
  App
    usb_device.c
    usb_device.h
    usbd_cdc_if.c
    usbd_cdc_if.h
    usbd_desc.c
    usbd_desc.h
```

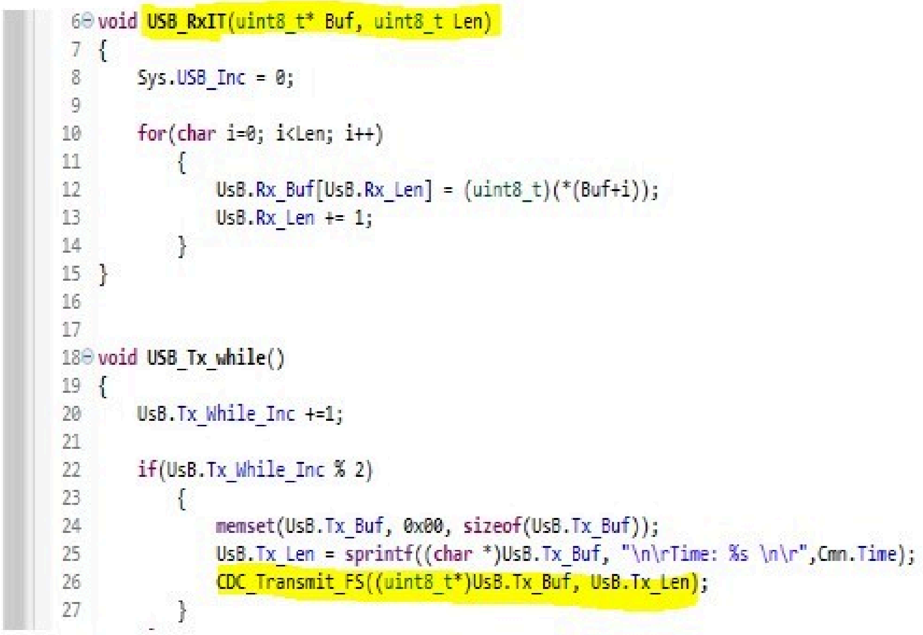

```
  * @{
  */

uint8_t CDC_Transmit_FS(uint8_t* Buf, uint16_t Len);

/* USER CODE BEGIN EXPORTED_FUNCTIONS */
__weak void USB_RxIT(uint8_t* Buf, uint8_t Len);
/* USER CODE END EXPORTED_FUNCTIONS */

/**
```


```
F103_USBTxRx_Relay_DS3231_EEPROM_SSD1306
  Binaries
  Includes
  Core
    Inc
    Src
      Proj
        Common.c
        EEPROM_24C32.c
        Relay_Code.c
        RTC_DS3231.c
        SSD1306_Application.c
        SSD1306_Driver.c
        USB_TxRx.c
      main.c
      stm32f1xx_hal_msp.c
      stm32f1xx_it.c
      syscalls.c
```


```
void USB_RxIT(uint8_t* Buf, uint8_t Len)
{
    Sys.USB_Inc = 0;

    for(char i=0; i<Len; i++)
        {
            UsB.Rx_Buf[UsB.Rx_Len] = (uint8_t)(*(Buf+i));
            UsB.Rx_Len += 1;
        }
}


void USB_Tx_while()
{
    UsB.Tx_While_Inc +=1;

    if(UsB.Tx_While_Inc % 2)
        {
            memset(UsB.Tx_Buf, 0x00, sizeof(UsB.Tx_Buf));
            UsB.Tx_Len = sprintf((char *)UsB.Tx_Buf, "\n\rTime: %s \n\r",Cmn.Time);
            CDC_Transmit_FS((uint8_t*)UsB.Tx_Buf, UsB.Tx_Len);
        }
```

Chapter 5. I²C Protocol

- Since SSD1306 Display is used, 400Khz is Must

```
/* USER CODE END I2C1_Init 1 */
hi2c1.Instance = I2C1;
hi2c1.Init.ClockSpeed = 400000;
hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
hi2c1.Init.OwnAddress1 = 0;
hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
hi2c1.Init.OwnAddress2 = 0;
hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
```

- GPIO Pull up must be set, check inside stm32f1xx_hal_msp.c

```
__HAL_RCC_GPIOB_CLK_ENABLE();
/**I2C1 GPIO Configuration
PB6     ------> I2C1_SCL
PB7     ------> I2C1_SDA
*/
GPIO_InitStruct.Pin = GPIO_PIN_6|GPIO_PIN_7;
GPIO_InitStruct.Mode = GPIO_MODE_AF_OD;
GPIO_InitStruct.Pull = GPIO_PULLUP;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
```

5.1. Full Working Code is Attached in the GitHub Link

Note: Give 3.3v to SSD1306 and 5V to Other Modules.

```
https://github.com/ferdinvivian/T1/tree/main/MCU_Coding/Project
```