

```
Core/Source/main.c - STM32CubeIDE
Navigate Search Project Run Window Help
sinksio.c main.c startup_stm32f401ccfx.s stm32f4xx_it.c STM32F401CCFP_FLASH.ld
46 /* USER CODE END PV */
47
48 /* Private function prototypes -----*/
49 void SystemClock_Config(void);
50 static void MX_GPIO_Init(void);
51 static void I2C1_Init(void);
52 /* USER CODE BEGIN PFP */
53
54 /* USER CODE END PFP */
55
56 /* Private user code -----*/
57
58 int Intp[16] = {0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x81, 0x42, 0x24, 0x18, 0x24, 0x42, 0x81};
59
60 int i;
61
62 unsigned char buff[] = {0x03, 0x00};
63 /* USER CODE BEGIN 0 */
64
65 /* USER CODE END 0 */
66
67 /**
68  * @brief The application entry point.
69  * @retval int
70  */
71 int main(void)
72 {
73     /* USER CODE BEGIN 1 */
74
75     /* USER CODE END 1 */
76
77     /* MCU Configuration -----*/
78
79     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
80     HAL_Init();
81
82     /* USER CODE BEGIN Init */
83
84     /* USER CODE END Init */
85
86     /* Configure the system clock */
87     SystemClock_Config();
88
89     /* USER CODE BEGIN SysInit */
90
91     /* USER CODE END SysInit */
92
93     /* Initialize all configured peripherals */
94     MX_GPIO_Init();
95     I2C1_Init();
96     /* USER CODE BEGIN 2 */
97
98     /* USER CODE END 2 */
99
100     /* Infinite loop */
101     /* USER CODE BEGIN WHILE */
102     HAL_I2C1_Master_Transmit(&hi2c1, 0x82, buff, 2, 10);
103     buff[0] = 0x01;
104     while (1)
105     {
106         for(i=0;i<16;i++){
107             GPIO->ODR = Intp[i] ^ 0x0f;
108             buff[i] = i;
109             HAL_I2C1_Master_Transmit(&hi2c1, 0x82, buff, 2, 10);
110             HAL_Delay(500);
111         }
112         /* USER CODE END WHILE */
113
114         /* USER CODE BEGIN 3 */
115
116         /* USER CODE END 3 */
117     }
118
119     /**
120     * @brief System Clock Configuration
121     * @retval None
122     */
123 void SystemClock_Config(void)
124 {
125     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
126     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
127
128     /* Configure the main internal regulator output voltage
129     */
130     __HAL_RCC_PWR_CLK_ENABLE();
131     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
132     /* Initializes the CPU, AHB and APB buses clocks
133     */
134     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
135     RCC_OscInitStruct.HSISource = RCC_HSI_ON;
136     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
137     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
138     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
139     {
140         Error_Handler();
141     }
142     /* Initializes the CPU, AHB and APB buses clocks
143     */
144     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
145     |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
146     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
147 }
```

```
Core/Source/main.c - STM32CubeIDE
tor Navigate Search Project Run Window Help
sinksio.c main.c startup_stm32f401ccfx.s stm32f4xx_it.c STM32F401CCFP_FLASH.ld
94 MX_GPIO_Init();
95 MX_I2C1_Init();
96 /* USER CODE BEGIN 2 */
97
98 /* USER CODE END 2 */
99
100 /* Infinite loop */
101 /* USER CODE BEGIN WHILE */
102 HAL_I2C1_Master_Transmit(&hi2c1, 0x82, buff, 2, 10);
103 buff[0] = 0x01;
104 while (1)
105 {
106     for(i=0;i<16;i++){
107         GPIO->ODR = Intp[i] ^ 0x0f;
108         buff[i] = i;
109         HAL_I2C1_Master_Transmit(&hi2c1, 0x82, buff, 2, 10);
110         HAL_Delay(500);
111     }
112     /* USER CODE END WHILE */
113
114     /* USER CODE BEGIN 3 */
115
116     /* USER CODE END 3 */
117 }
118
119 /**
120 * @brief System Clock Configuration
121 * @retval None
122 */
123 void SystemClock_Config(void)
124 {
125     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
126     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
127
128     /* Configure the main internal regulator output voltage
129     */
130     __HAL_RCC_PWR_CLK_ENABLE();
131     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
132     /* Initializes the CPU, AHB and APB buses clocks
133     */
134     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
135     RCC_OscInitStruct.HSISource = RCC_HSI_ON;
136     RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
137     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
138     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
139     {
140         Error_Handler();
141     }
142     /* Initializes the CPU, AHB and APB buses clocks
143     */
144     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
145     |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
146     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
147 }
```