



Chương 3: Cách sử dụng Timer và ngắt Timer

```
using System;  
using System.Data;
```

```
namespace MyProject
```

```
{  
    /// <summary>
```

```
    /// Customer Master
```

```
    /// </summary>
```

```
    /// <remarks></remarks>
```

```
    public class Customer
```

```
    {  
        private string pCustomerCode;
```

```
        private string pCustomerName;
```

```
        private string pAddress;
```

```
        private string pMobileNo;
```

```
        private string pEmail;
```

```
        public string CustomerCode
```

```
        {  
            get { return pCustomerCode;
```

```
            set { pCustomerCode = value; }
```

```
        }  
        public string CustomerName
```

```
        {  
            get { return pCustomerName;
```

```
            set { pCustomerName = value; }
```

```
        }  
        public string Add
```

```
        {  
            get { }
```

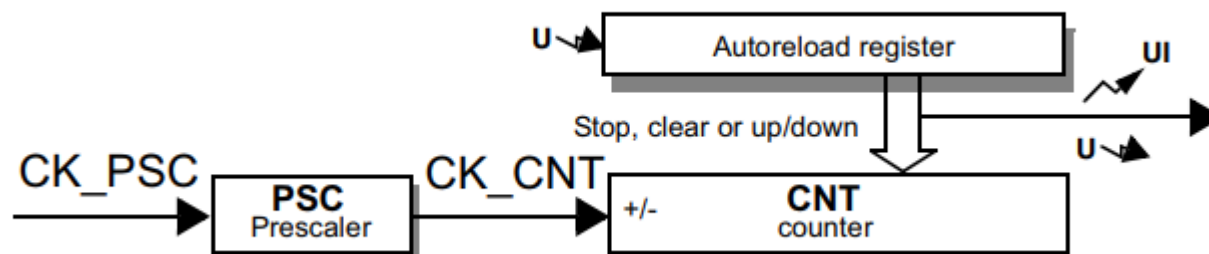
```
        }  
    }  
}
```

Timer là gì?

Timer là bộ hẹn giờ bên trong của STM32. Timer có 2 công dụng chính. Đo đặc thời gian và định thời gian cho một sự kiện nào đó

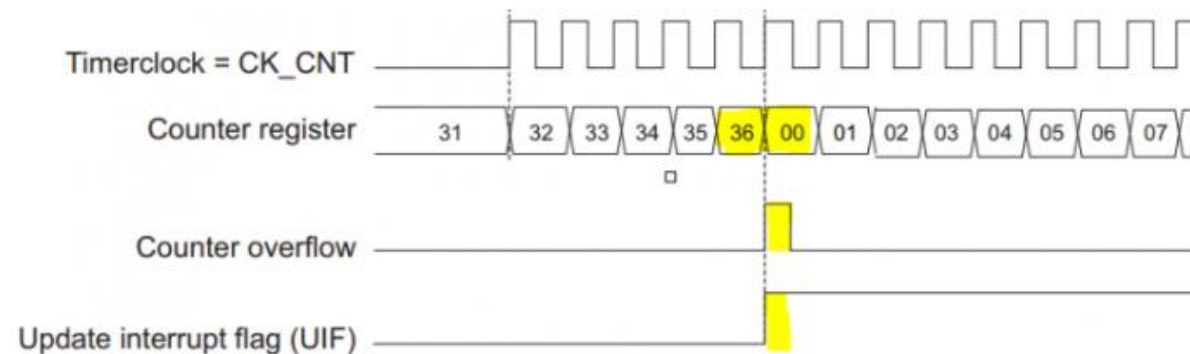
Cấu tạo của Timer bao gồm:

- + Counter Register (TIMx_CNT): lưu giá trị đếm
- + Prescaler Register (TIMx_PSC): lưu giá trị chia từ tần số cơ sở cấp cho Timer để tạo ra tần số thích hợp
- + Auto-Reload Register (TIMx_ARR): lưu giá trị đích đếm lên hoặc đếm xuống

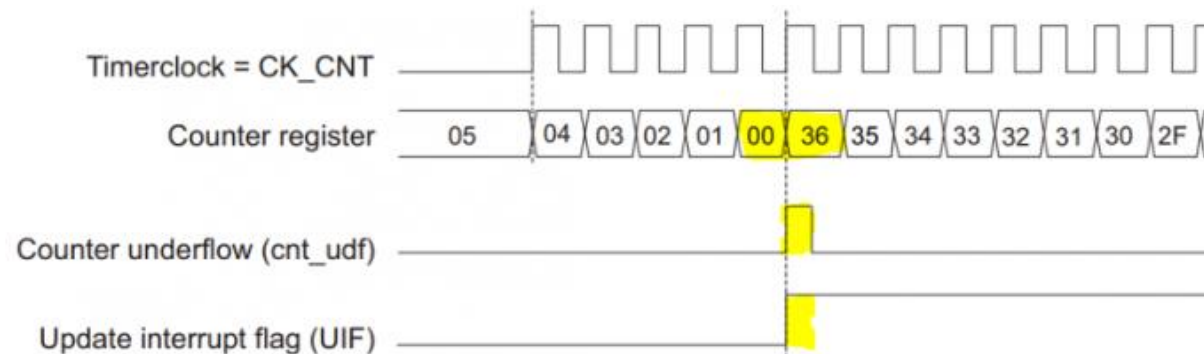


Các chế độ đếm của Timer

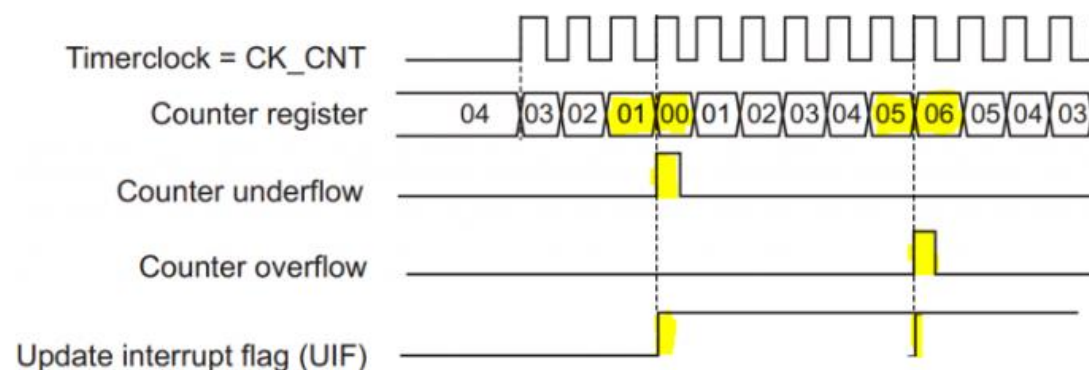
Đếm lên – Sự kiện tràn: overflow



Đếm Xuống – Sự kiện tràn: underflow

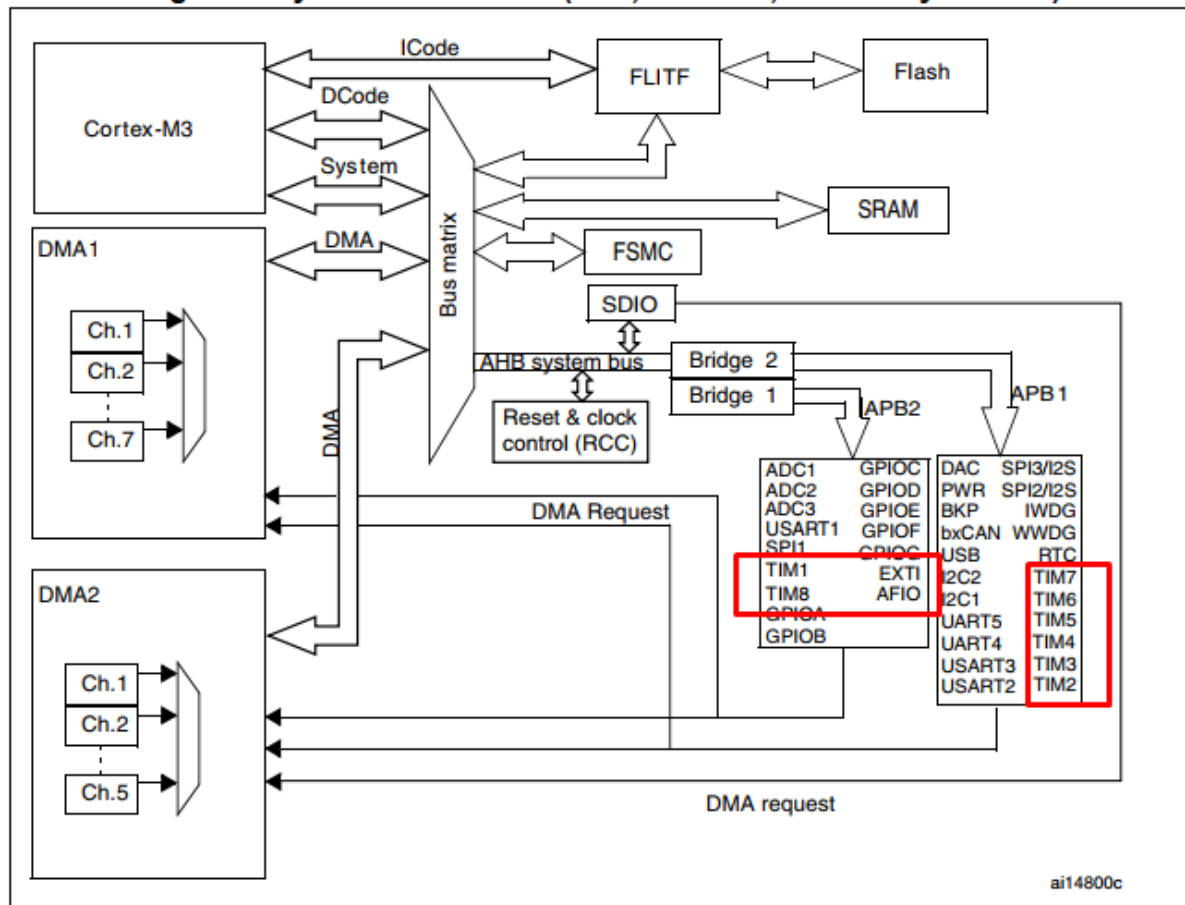


Đếm Xuống – Sự kiện tràn: underflow + overflow

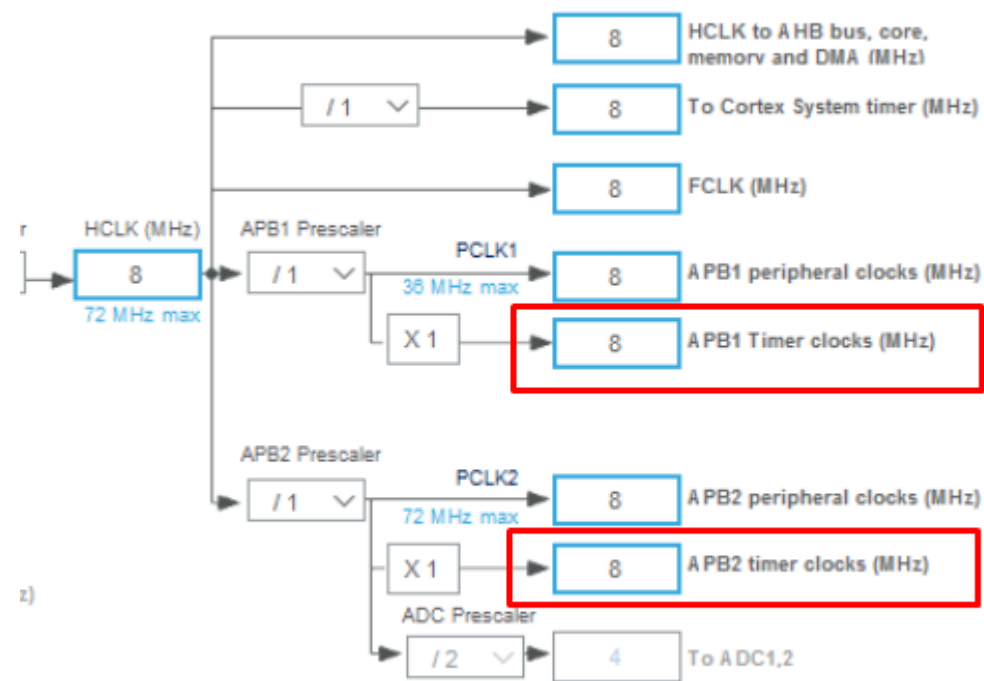


Clock cho Timer

Figure 1. System architecture (low-, medium-, XL-density devices)



Tim 1 và Tim 8 dùng bộ APB2
Tim 2 -> Tim 7 dùng bộ APB1



Tương ứng khi khai báo trên CubeMX

Công thức tính thời gian tràn của Timer



KHUÊ NGUYỄN CREATOR
Chia sẻ đam mê

$$UpdateEvent = \frac{Timer_{clock}}{(Prescaler + 1)(Period + 1)}$$

- UpdateEvent là thời gian mong muốn xảy ra sự kiện tràn
- Timer Clock = 1/F clock
- Prescaler là số chia để tạo ra tần số mỗi lần count của Timer
- Period là mục tiêu khi counter đếm tới sẽ có sự kiện tràn

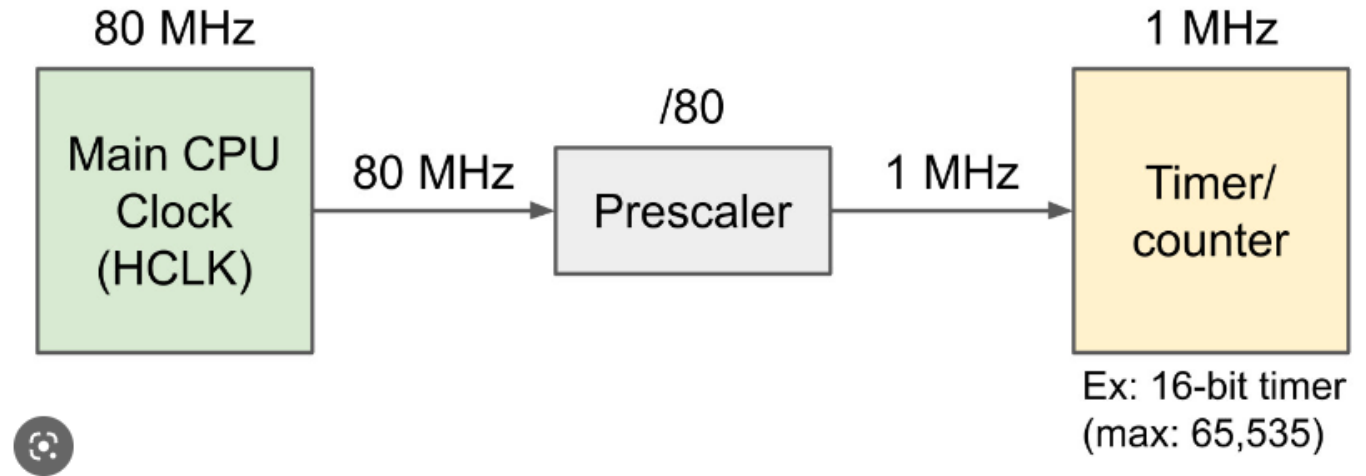
Bài tập:

Tính Prescaler và Period để có Thời gian tràn là 1s, với Clock là 8Mhz

Các bước làm Timer hoạt động

Các bước để Timer hoạt động như sau:

- Khởi tạo bộ Timer 1 với xung Clock thích hợp
- Ghi giá trị cho bộ chia tần Prescaler
- Ghi giá trị cho thanh ghi Auto-reload
- Bật Timer cho hoạt động
- Xử lý các sự kiện xảy ra như tràn



Tạo xung với Timer

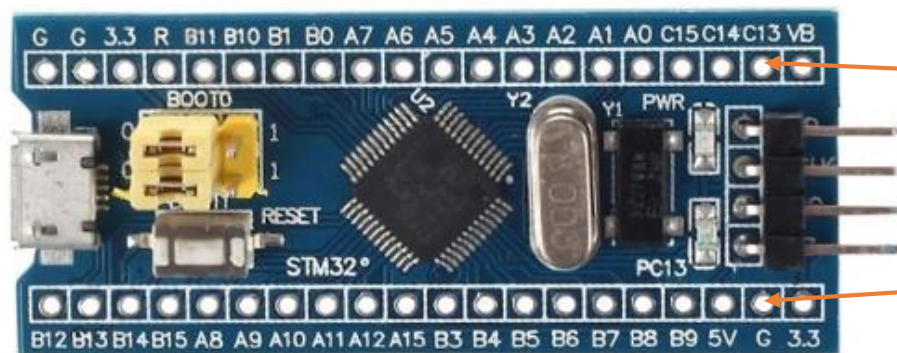
Chuẩn bị

- KIT STM32F103
- Logic Analyzer

Nối chân PC13 vào CH1 của Logic Analyzer
Chung chân GND

Bài tập 1:

Tính toán thông số nạp vào Timer 2 để nháy với $T = 500 \text{ ms}$



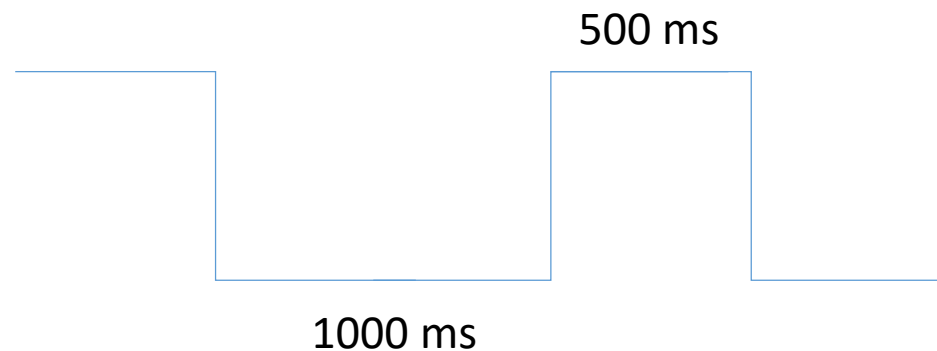
Tạo xung với Timer



KHUÊ NGUYỄN CREATOR
Chia sẻ đam mê

Bài 2:

Tính toán nạp vào Timer 2 để tạo xung có xung dương 500ms, xung âm 1000 ms



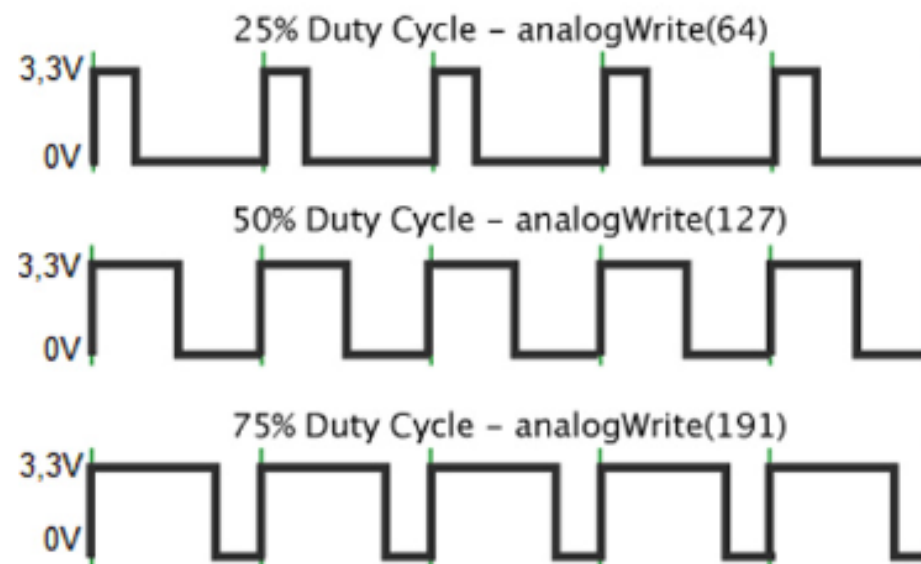
Tạo xung với Timer



KHUÊ NGUYỄN CREATOR
Chia sẻ đam mê

Bài 3:

Tính toán nạp vào Timer 2 để tạo xung có xung dương xung dương tăng dần từ 0 – 999ms



Đọc thời gian xung

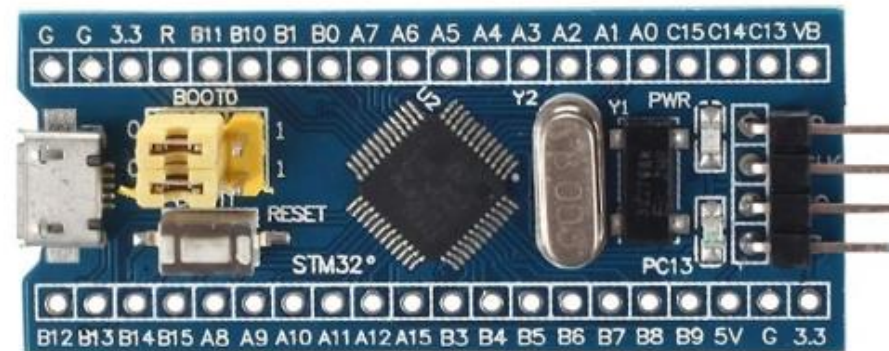
Chuẩn bị:

- KIT STM32F103
- Nút nhấn
- Logic Analyzer

Mắc nút nhấn vào chân PA0, Logic Analyer vào chân PA0

Bài 1:

Đếm thời gian bắt đầu nhấn nút đến khi nhả ra.





Bài 1:

Nhấn giữ giữ nút N giây, sau đó sáng led theo N giây đó.

Bài 2:

Nhấn nút và led sáng dần.



Chuẩn bị:

- KIT STM32F103
- Nút nhấn
- Logic Analyzer

Mắc nút nhấn vào chân PA0, Led vào chân PC13

Lập trình nút nhấn 3 chế độ

Nhấn nút 1 lần: nháy led 1s

Nhấn giữ nút 3s: nháy led 3s

Nhấn đúp nút: nháy led 3 lần