STM32 CAN filter - four working modes of bxCAN

tags: CAN filter 4 working modes of bxCAN

1, the role of CAN filter

In interconnected products, bxCAN provides 28 bit-width-variable/configurable identifier filter sets, and the software programs them to select the message it needs in the message received by the pin, while other The message is discarded. There are 14 bit width variable/configurable identifier filter groups in other STM32F103xx series products. The CAN filter only needs to set the bxCAN message ID, then the MCU can only receive these CAN messages, which are filtered out from the hardware, and do not need software to participate at all, thus saving a lot of time for saving the MCU. Focus more on other matters, this is what the bxCAN filter means.

2, two filtering modes

2.1, list mode

If we only pay attention to a message ID, we need to write this ID to the list. If you need to pay attention to two, you need to write two message IDs. If you need to pay attention to 100, you need to write 100. If you need to 10,000, then you need to write 10,000, but the question is, is there a big list for us to use? Everyone knows that the resources on the MCU are limited, and it is impossible to provide 10.000 or

more, or even 100. Obviously, the way this list is limited by the size of the list. In fact, if a filter of bxCAN works in list mode and the scale is 32, only two message IDs can be written for each filter list. If the scale is 16, each filter list can write up to 4 CAN IDs (in the list mode, if the bit width is 32, two ID lists can be set. 16-bit mode can set 4 ID lists).

In the 16-bit wide list mode, the four 16-bit variables FilterIdHigh, FilterIdLow, FilterMaskIdHigh, and FilterMaskIdLow are used to store a standard CAN ID, so that four standard CAN IDs can be stored. Note that this is In this mode, the extended CANID cannot be processed. Any filter that needs to filter the extended CAN ID requires a 32-bit wide mode.

It can be seen that the resources of the MCU are very limited and cannot be left as we want. Therefore, we need to consider another alternative, which should not be subject to quantitative restrictions.

2.2, mask mode

If we prepare two pieces of paper, one writes the mask code and the other piece writes the verification code. When the corresponding bit on the mask code is 1, it means that this bit needs to be compared with the corresponding bit of the verification code. Otherwise, it means no. need. When the machine performs the task, it first performs an AND operation on the acquired ID code and the mask code, and then compares the result with the verification code, and determines whether to pass according to whether the judgment is the same.

The meaning of the mask code and the verification code: the number of results that can be passed depends entirely on the mask code. If it is set wide, it can pass more (all bits are 0, but any filtering operation, then anyone can pass), set narrow , then pass less (all bits set to 1, only one can pass). So what is the use of this? Because the mask mode of the bxCAN filter is in this way, in bxCAN, two registers (C

CAN_FiR2) are used to store the mask code and the verification code, thereby implementing the mask mode workflow. In this way, we know how the mask mode of the bxCAN filter works.

3、CAN ID

3.1, standard CAN ID

In 1986, the German electric company BOSCH developed a CAN communication protocol for automobiles. At the beginning, the CAN ID was defined as 11 bits. We call it the standard format. The standard CAN ID is stored in ID18~ID28, a total of 11 bits.

The standard ID is generally less than or equal to <=0x7FF (11 bits) and contains only the base ID.

For example, standard CAN ID 0x7E1, binary expansion is 0b 0[111 1110 0001], only 11 bits in brackets are valid, all of which are basic IDs.

3.2, extended CAN ID

With the development of the industry, it was later found that the 11-digit CAN ID was not enough, so 18 digits were added and the CAN ID was extended to 29 digits.

For example, expand CAN ID 0x1835f107, binary expansion is 0b 000 [1 1000 0011 10] [01 1111 0001 0000 0111], only the bits in parentheses are valid, a total of 29 bits, in the left parenthesis The 11 bits are the basic ID, and the 18 bits in the right parenthesis are the extended ID. After knowing this, we can easily split a CANID into a basic ID and an extended ID, which will be used multiple times in subsequent contents. Again, note that the extension ID is located to the right of the base ID. In the configuration of the extended CAN ID, the extension ID is located at the lower 18 bits, and the base ID is at the upper 11 bits, so to obtain the basic ID of the extended

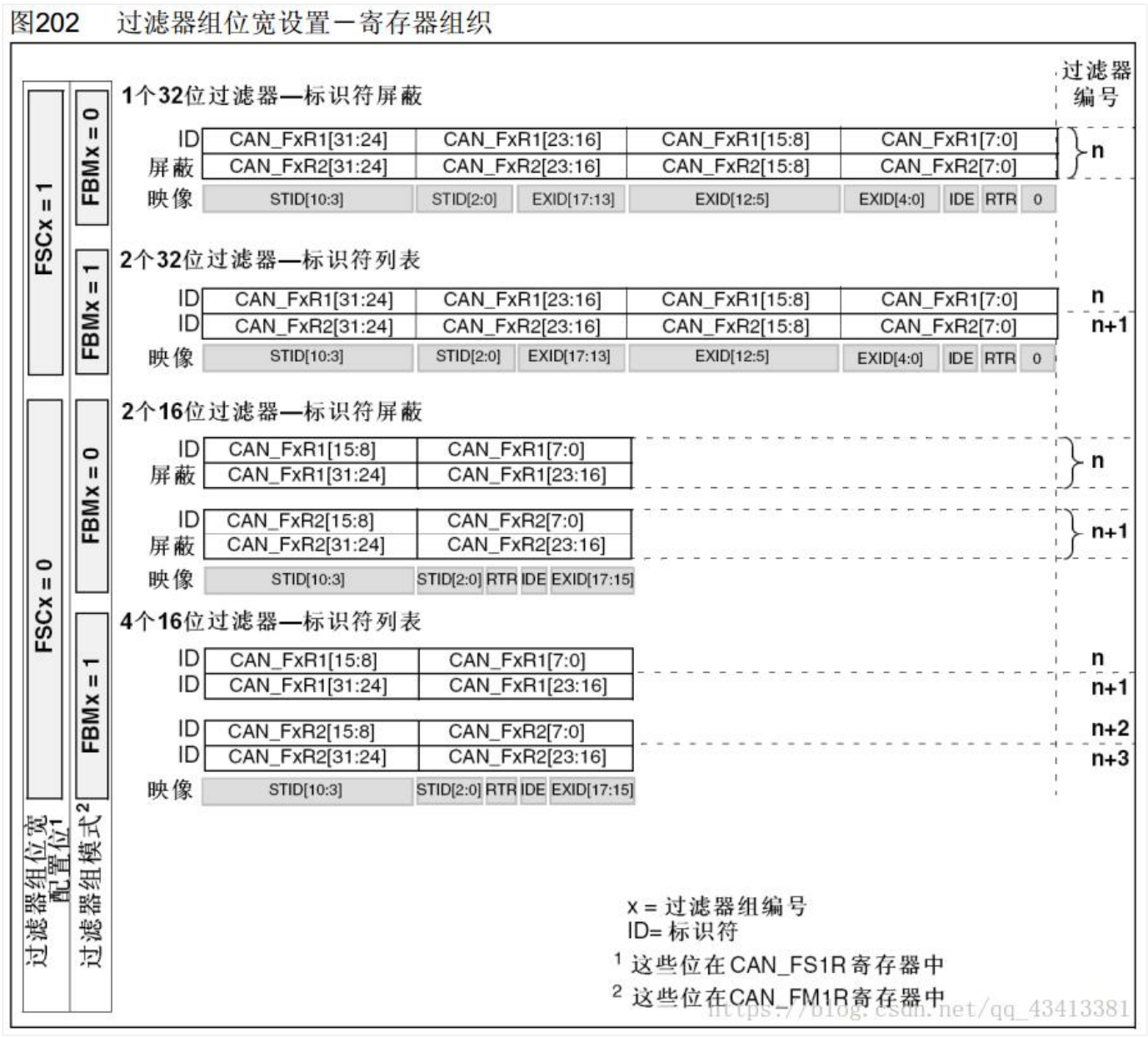
CANID, only You need to shift this CANID to the right by 18 bits (this algorithm will be used multiple times in the future, please remember!).

4 working modes

The filtering mode is divided into a list mode and a mask mode. Therefore, we need such a bit to mark. The user can set this bit to mark whether he wants the filter to work in list mode or mask mode. Therefore, this expression The bits of the filter mode are defined on the FBMx bit in the CAN_FM1R register.

According to the mode and bit width settings, we can get a total of 4 different combinations: 32-bit wide list mode, 16-bit wide list mode, 32-bit wide mask mode,

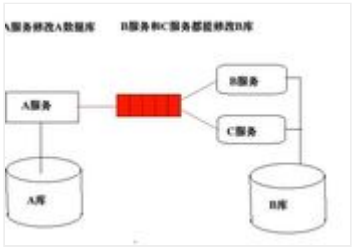
16-bit wide mask mode.



In bxCAN, there are two registers CAN_FxR1 and CAN_FxR2 in each filter. Both registers are 32-bit. His definition is not fixed. His definition is different for different working modes. In the list mode - 32-bit wide mode, the definitions of the two registers are the same, are used to store a specific expected CAN ID, so you can deposit 2 expected CAN ID (standard Both CAN ID and extended CAN ID can be used; if in mask mode - 32 bit wide mode, CAN_FxR1 is used as a 32-bit wide verification code, and CAN_FxR2 is used as a 32-bit wide mask. At 16-bit width, CAN_FxR1 and CAN_FxR2 are each split into two 16-bit wide registers. In the list mode - 16-bit wide mode, CAN_FxR1 and CAN_FxR2 are defined the same, and each is split into two, for a total of It can write 4 standard CAN IDs. If it is

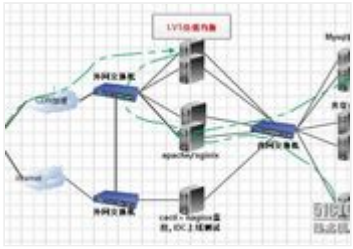
wide mask mode, it can be used as 2 pairs of verification code + mask combination, but it can only filter standard CAN ID. This is the solution for the bxCAN filter, which uses these four modes of operation.

Intelligent Recommendation



Four common working modes of RabbitMQ

1. Normal mode There is only one consumer in a queue, the producer sends the message to the queue, and the consumer takes the message from the queue 2. Working mode Multiple consumers listen to the sa...

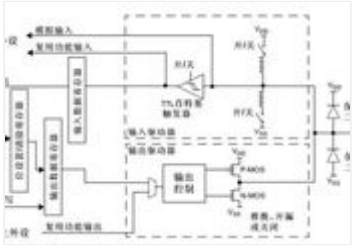


Four working modes of LVS cluster

Cluster What is a cluster Computer cluster is a kind of computer system, which is connected by a group of loosely integrated computer software and/or hardware to complete computing work in a highly ti...

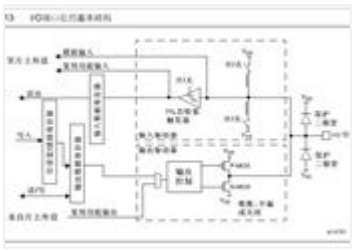
STM32 timer synchronous four modes

MasterOutputTrigger: This bit can select the mode to be transmitted to the master timer to achieve information from the synchronization (TRGO) . Combinations of these bits is as follows: 000 : Reset &...



8 working modes of STM32 GPIO and GPIO registers

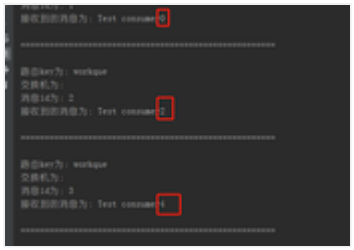
Basic knowledge STM32F103ZET6 has 7 groups of IO ports, namely GPIOA, GPIOB, GPIOC to GPIOG. Each group of IO ports has 16 IOs, so there are 112 IOs in total. In addition to being used as GPIO, most o...



Illustrate the principle of STM32 GPIO eigh t working modes

table of Contents 1.1 GPIO basic structure 1.2 Diagram of eight working modes Input GPIO input working mode 1—input floating mode GPIO input working mode 2—input pull-up mode GPIO input wo...

More Recommendation



Message queue (four)-rabbitMQ four worki ng modes

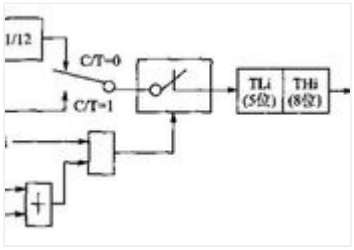
RabbitMQ working mode Mode summary: 1. Simple mode helloworld One producer, one consumer, no switch is needed (use the default switch) 2, Work Queue mode One producer, multiple consumers (competitive ...

STM32 Controller area network (bxCAN) Id entifier filtering

Identifier filtering In the CAN protocol the identifier of a message is not associated with the address of a node but related to the content of the message. Consequently a transmitter broadcasts ...

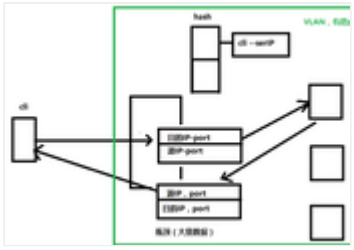
stm32 CAN filter group

Among interconnected products, CAN1 and CAN2 share 28 filter groups There are 14 filter groups in other STM32F103xx series products Bit width setting Four configuration methods: 1 32-bit mask bit patt...



Four working modes of the MCU timer/co unter

Mode 1 When M0=0 in TMOD and M1=0, it is a 13-bit counting or timing mode, in which TLi uses the lower 5 bits, and its structure is as shown in Fig. 4.2. Figure 4.2 Structure diagram of T0 and T1 in m...



Four working modes of lvs and their advan tages and disadvantages

1. VS/NAT (address translation to realize virtual server) After receiving the client's request, the dispatcher sends the client's request to the server according to the scheduling algorith...

Related Posts

- Reprinted-talk about the STM32 CAN filter-bxCAN filter 4 working modes and use method summary
- Four working modes of the switch
- RabbitMQ: Four working modes
- Several working modes of STM32 GPIO
- STM32 GPIO eight working modes
- Eight working modes of GPIO in stm32

- 8 working modes of STM32 GPIO
- stm32 GPIO eight working modes
- STM32 bxCAN bus configuration information
- Introduction to four working modes of lvs

Popular Posts

- DBUnit based manage (Dao) unit test
- Click on Bootstrap's popover automatically rolled back top issues
- Android NDK: Could not find application project directory !
- POJ 2070 Water Problem (Water Problem)
- PHP uses regular matching to find the src path in the image img tag
- JAVA Integer value comparison problem
- Eureka's workflow / working principle
- multi-threaded python crawling embarrassing one hundred
- MapReduce study notes (5) - Map to achieve the end join
- Basic usage of Bootstrap

Recommended Posts

- ARFoundation Quick Start-02 Environment Construction
- java.lang.NoClassDefFoundError: backtype/storm/spout/Scheme error
- 10. Python's object-oriented, self, class variables, instance variables, methods
- Database new field update new value
- Application of power 3D monitoring based on HTML5 technology (2)
- Four ways of array copy in java
- The apk file generated by android studio cannot be installed
- Save YUV to local BMP image
- Anticipation effect
- NGUI: HUD Text (head injury floating text)

Related Tags

RabbitMQ

MCU

C/C++

STM32

stm32

gpio

Internet of Things

CAN

Server cluster

Embedded