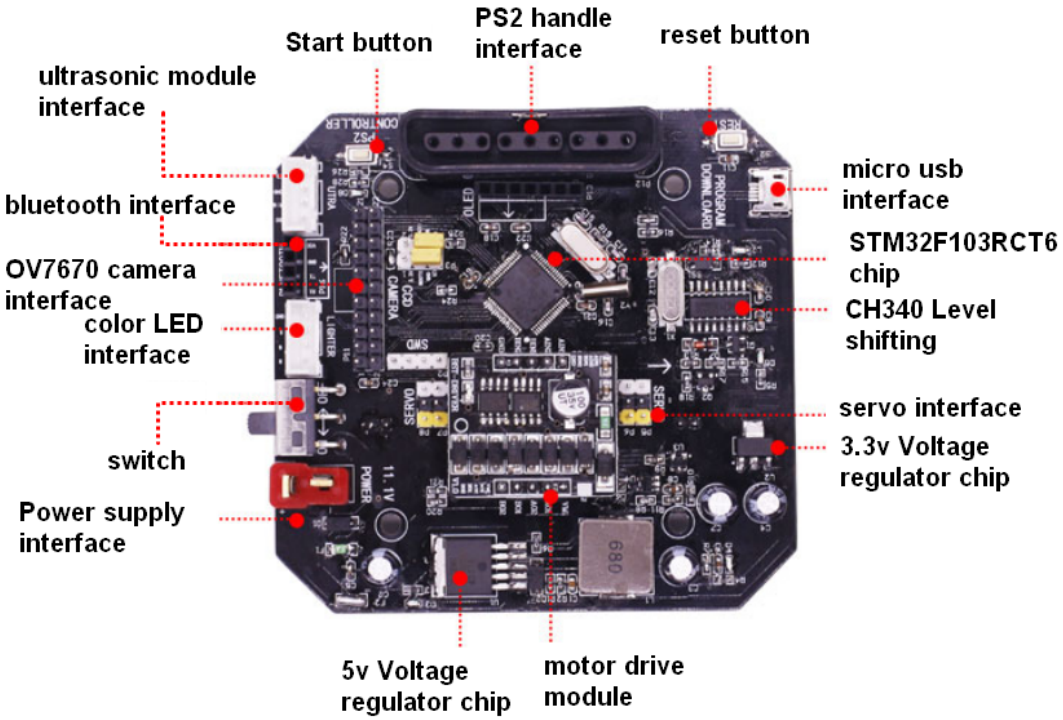
2.STM32 platform-------Advance

1) Preparation



1-1 STM32 expansion board

1-2 STM32 smart car

2)Purpose of Experimental

After the car is powered on, press the start button next to the PS2 logo on the expansion board, you will see that car advance 1s, stop 1s, back 1s, stop.

3)Principle of experimental

For the control of the 4-way DC geared motor, we use the 74HC244 driver chip and the H-bridge driver chip to drive the motor. In this module, M1A and M1B control one motor, M2A and M2B control one motor. When M1A is 0 and M1B is 1, the motor rotates forward. When M1A is 1, and M1B is 0, the motor reverses when both are When 0, the motor stops. When M2A is 0 and M2B is 1, the motor rotates forward. When M2A is 1, and M2B is 0, the motor reverses. When both are 0, the motor stops. This experiment mainly controls M1A to be low level, M2A is low level, and then the duty ratio of M1B and M2B high and low level is changed by timer to control the motor speed.

Let's talk about how to use a timer to control the motor.

Here we use timer 1 to drive the motor. First we first turn on the TIM1 clock as follows:

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE); //clock enable
```

Then initialize Timer 1 as follows:

```
TIM_TimeBaseStructure.TIM_Period = arr; //Set the value of the auto-reload register cycle of the next update event load activity
TIM_TimeBaseStructure.TIM_Prescaler = (psc-1); //Set the prescaler value used as the TIMx clock frequency divisor
TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1; //Set the clock split: TDTS = Tck_tim //36Mhz
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up; //TIM count up mode
TIM_TimeBaseStructure.TIM_RepetitionCounter = 0; // repeat count off
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure); //Initialize the time base unit of TIMx according to the specified parameters
```

Define timer update interrupts:

```
TIM_ITConfig(TIM1, TIM_IT_Update, ENABLE ); //Enable specified TIM1 interrupt, allow update interrupt
//Interrupt priority NVIC settings
NVIC_InitStructure.NVIC_IRQChannel = TIM1_UP_IRQn;    //TIM1 interrupt
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; //Preemptive priority level 0
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 3; //From priority level 3
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //IRQ channel is enabled
NVIC_Init(&NVIC_InitStructure); //Initialize the NVIC register
```

Enable timer 1:

```
TIM_Cmd(TIM1, ENABLE); //Enable TIMx
```

```
void TIM1_Int_Init(u16 arr,u16 psc)
```

The 2 parameters of this function are used to set the overflow time of TIM1. In the clock system part, we explained that when the system is initialized, the clock of the APB1 is

initialized by 2 in the default system initialization function SystemInit function, so the clock of APB1 is 36M, and the internal clock tree diagram of STM32 is: When the clock division number of APB1 is 1, the clock of TIM2~7 is the clock of APB1, and if the clock division number of APB1 is not 1, the clock frequency of TIM2~7 will be twice that of APB1 clock. Therefore, the clock of TIM3 is 72M, and based on the values of arr and psc we designed, we can calculate the interruption time. Calculated as follows:

Tout= ((arr+1)*(psc+1))/Tclk；

Tclk: Input clock frequency of TIM1 (in Mhz).

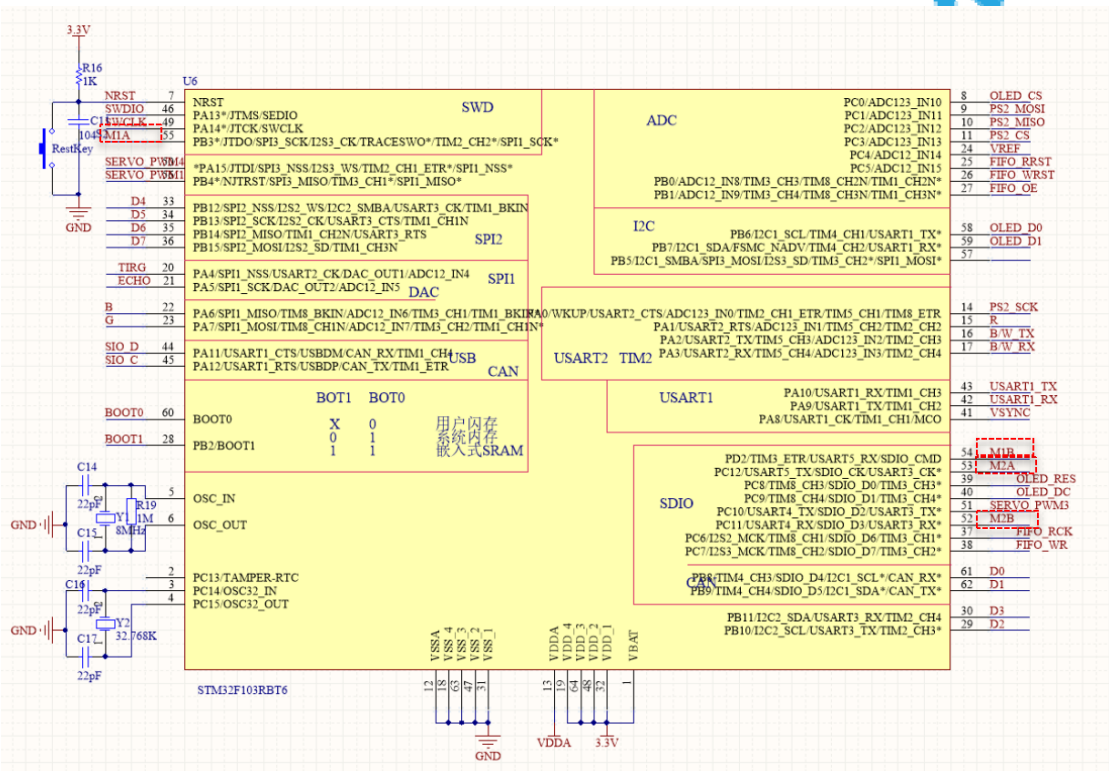
Tout: TIM1 overflow time (in us).

TIM1_Int_Init(9, 72);

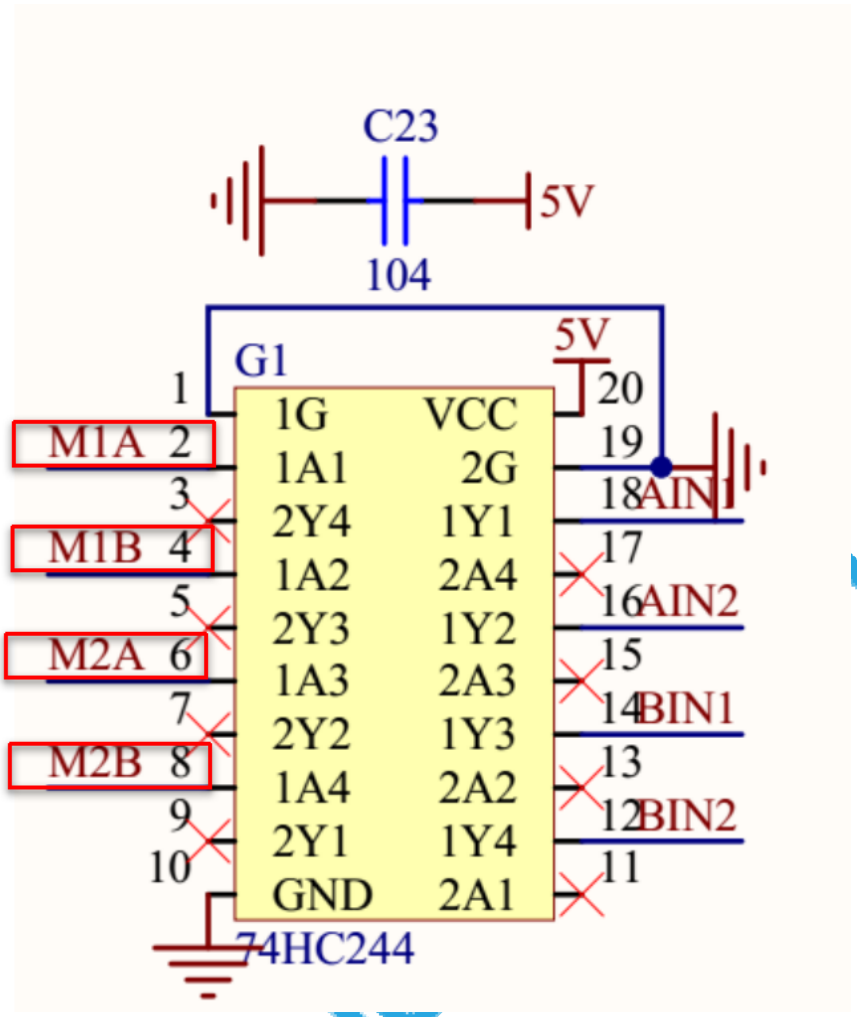
Here we set the overflow time as: (9+1)*(71+1)/72M = 10us

2) Experimental Steps

4-1 About the schematic



4-1 STM32 main control board circuit diagram



4-2 motor drive chip 74HC244

4-2 According to the circuit schematic:

M1A-----PB3
M1B-----PD2
M2A-----PC12
M2B----PC11

4-3 About the code

Please see the folder named Advance in the code folder.