

不讲理论的STM32教程

不讲理论的STM32教学



STM32
F103C8T6

ARMCortex-M3
48PIN

BUGXIONG

粉丝群：
622921667

用最简洁的话，
用最情简的视频，
用最完美的课程体系，
教会你**STM32**的使用

逐行敲代码！
配套教材以及配套课件！

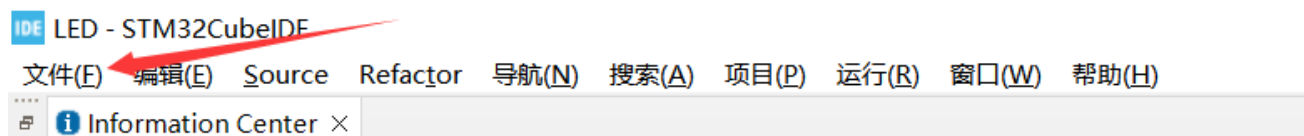
讲述人：阿熊学长

基础部分

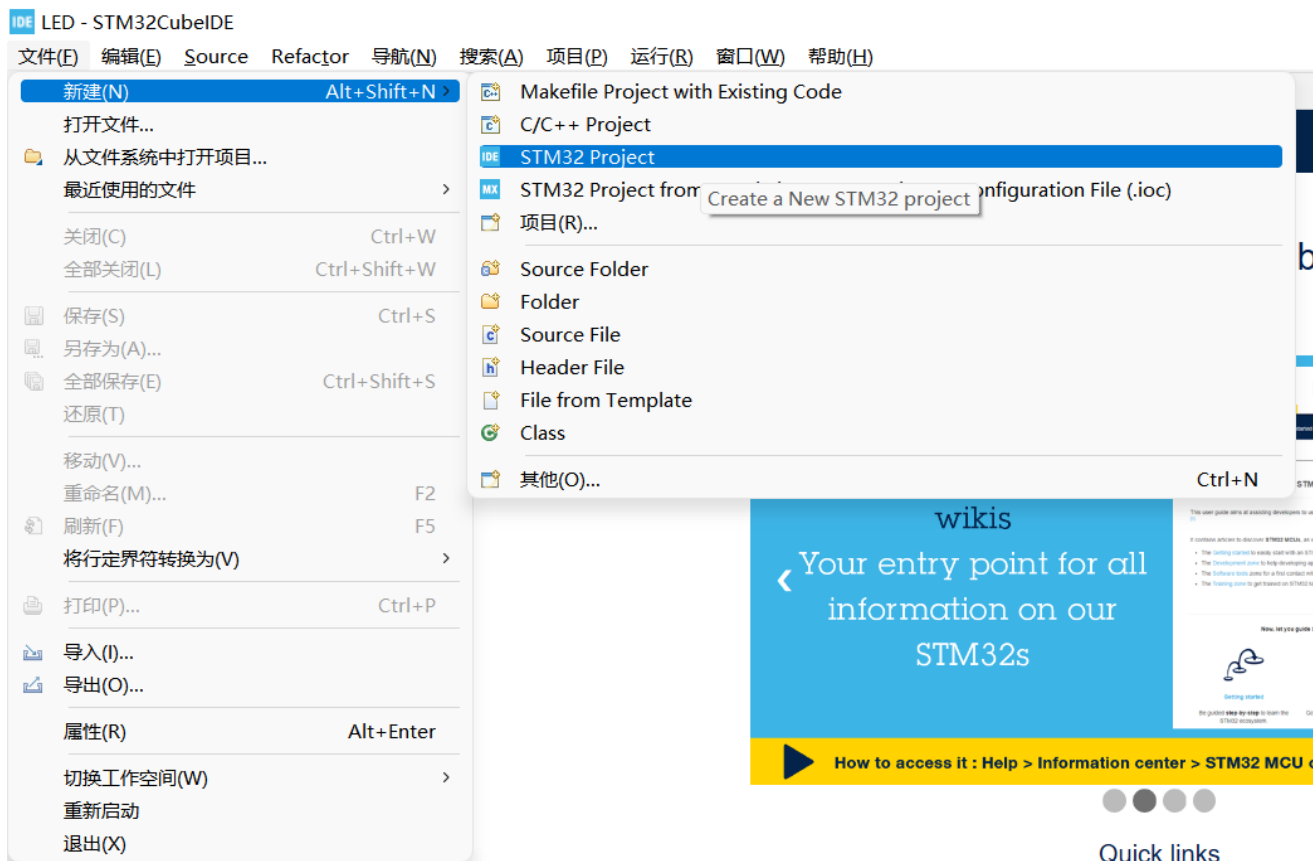
第四章：LED的点亮

1.新建项目

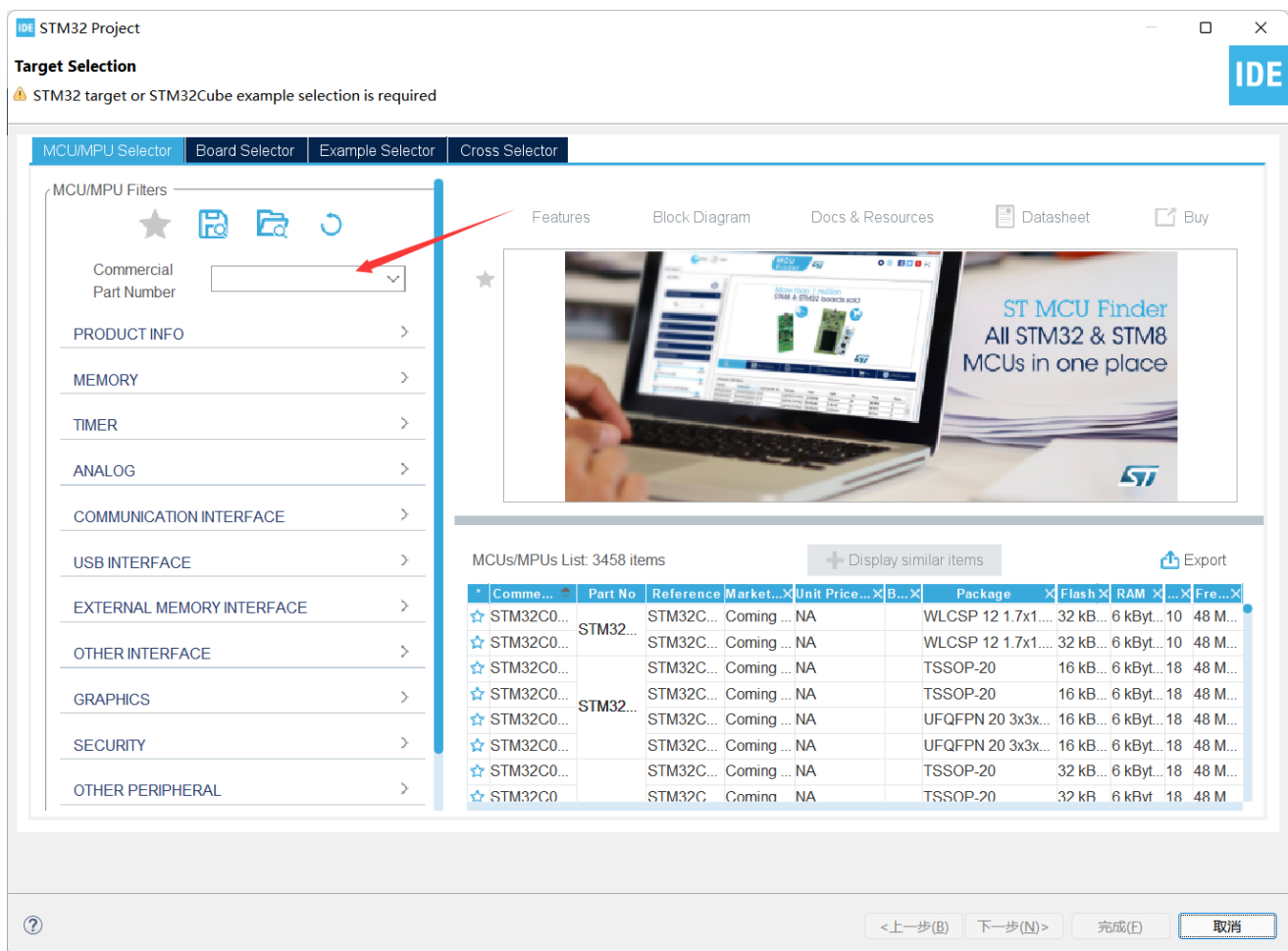
1.点击“文件”



2.点击“新建”，然后点击“STMProject”



3.填写芯片型号



4.选择芯片，然后点击“下一步>”

STM32F1 Series

STM32F103C8T6

Mainstream Performance line, Arm Cortex-M3
MCU with 64 Kbytes of Flash memory, 72 MHz
CPU, motor control, USB and CAN

ACTIVE


Product is in mass
production

Unit Price for 10kU (US\$) : **2.4195**

LQFP 48 7x7x1.4 mm

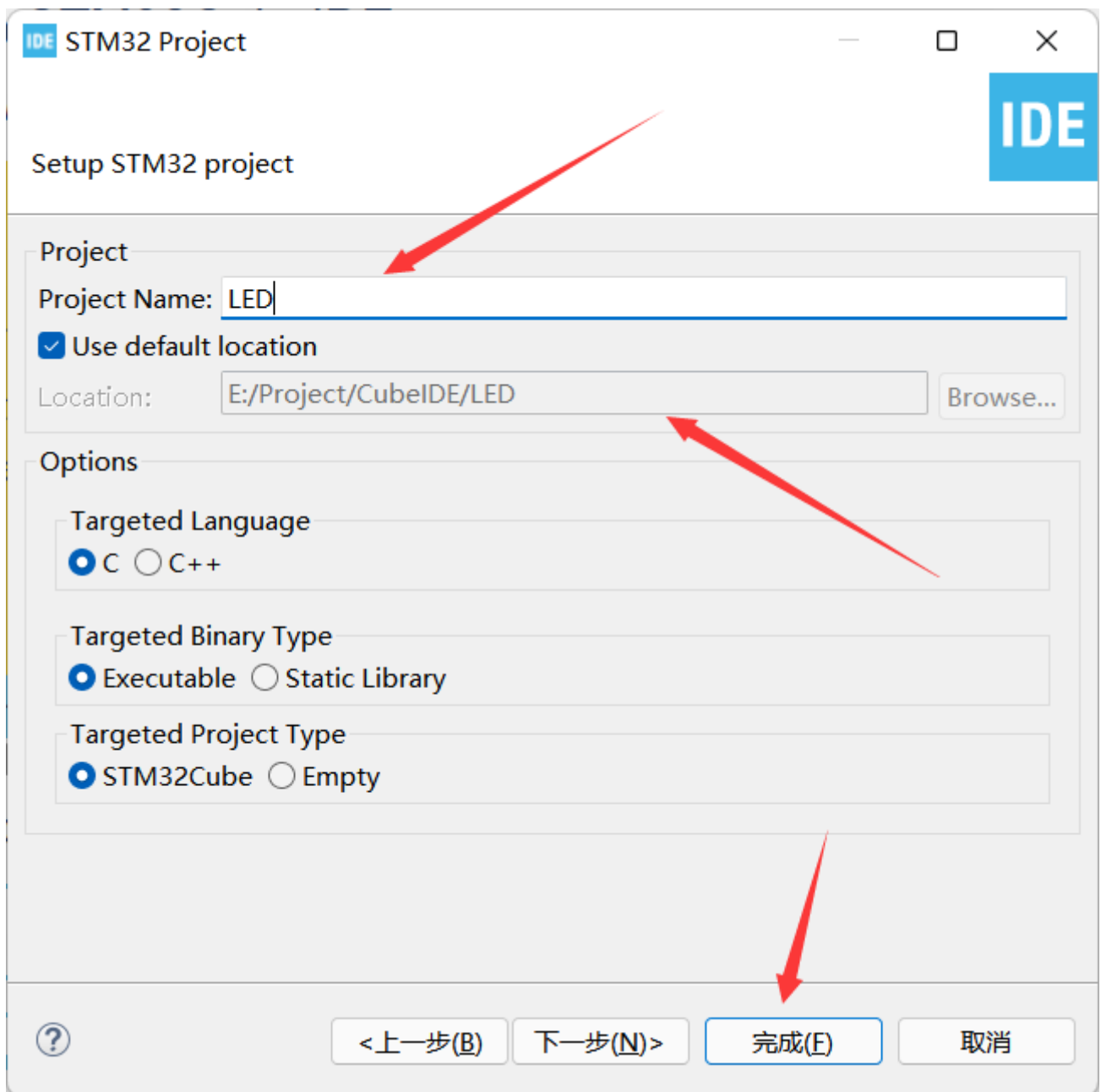
The STM32F103xx medium-density performance line family incorporates the high-

MCUs/MPUs List: 2 items

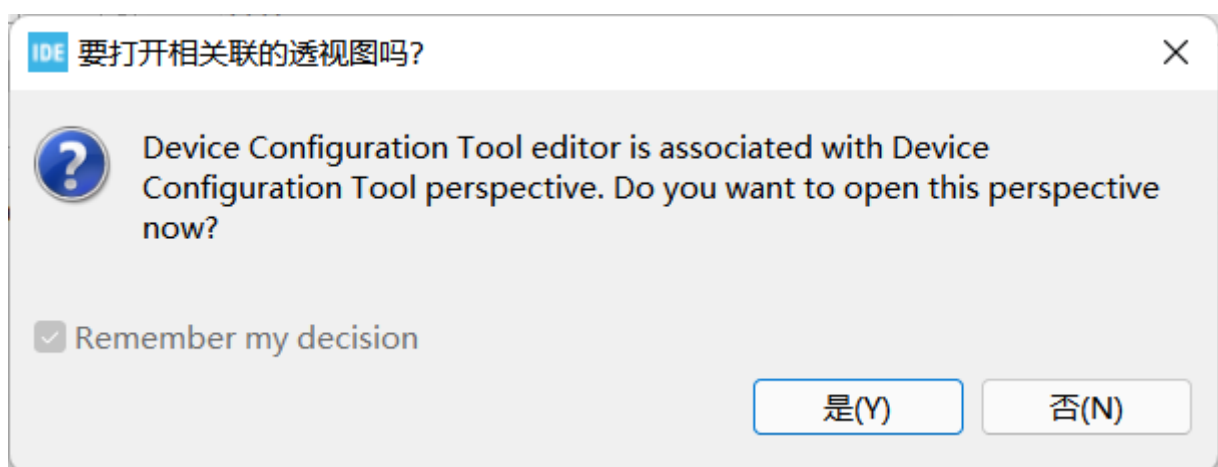
 Display similar items Export

| * | Comme... | Part No | Reference | Market... | Unit Price ... | B... | Package | Flash | RAM | ... | Fre... |
|---|-------------|-----------|------------|-----------|----------------|------|---------------------|----------|----------|-----|--------|
| ☆ | STM32F10... | STM32F... | STM32F1... | Active | 2.4195 | | LQFP 48 7x7x1.4 ... | 64 kB... | 20 kB... | 37 | 72 MHz |
| ☆ | STM32F10... | STM32F... | STM32F1... | Active | 2.4195 | | LQFP 48 7x7x1.4 ... | 64 kB... | 20 kB... | 37 | 72 MHz |

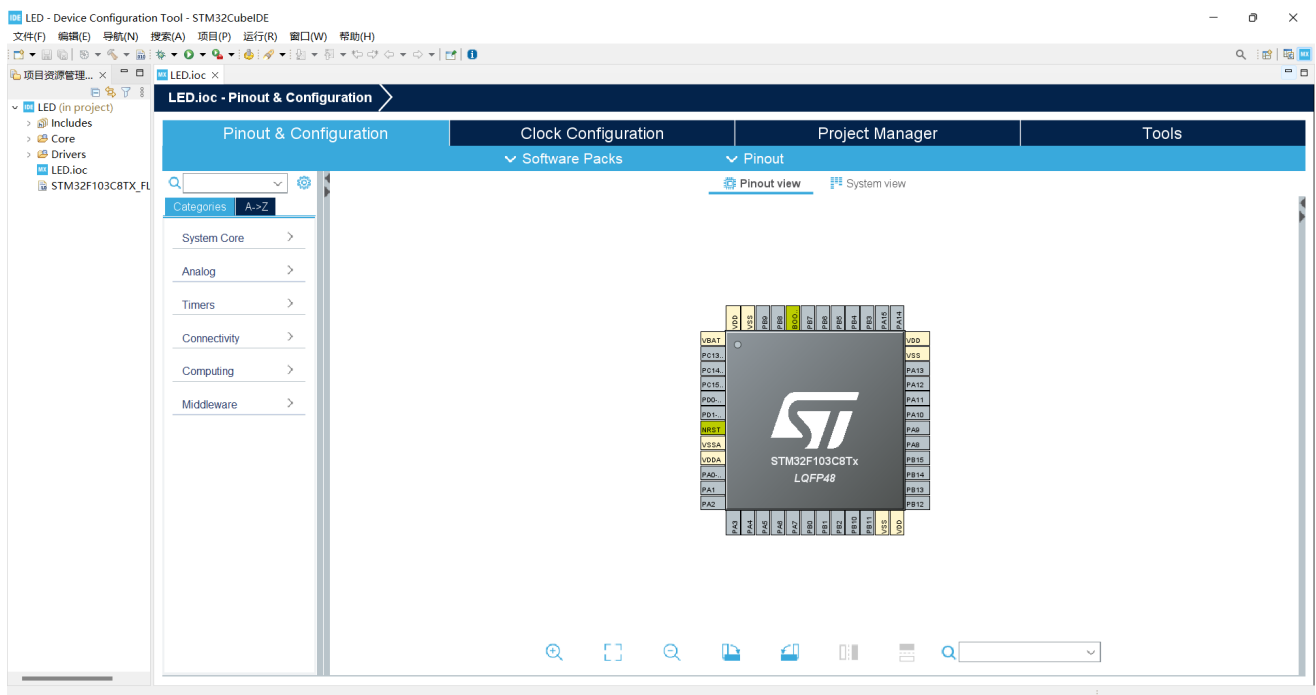
5.填写项目名称，路径，然后点击“完成”



6.打开透视图



弹出此对话框，选择“是”



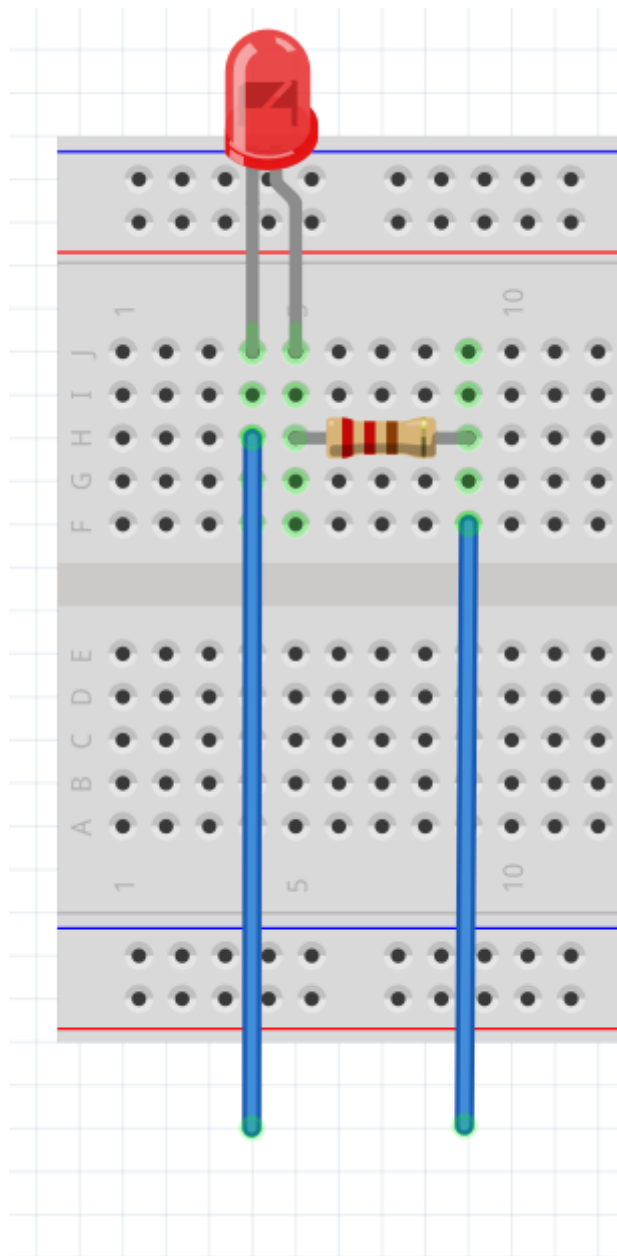
项目创建完成！

我们可以看到芯片示意图

2.项目实现

1.原理图配置

由于给位小伙伴手里的开发板可能是各种型号的在这里阿熊将为大家使用面包板进行连接，希望小伙伴们可以跟着阿熊一起做，这样可以减少发生错误的概率，也方便解决问题

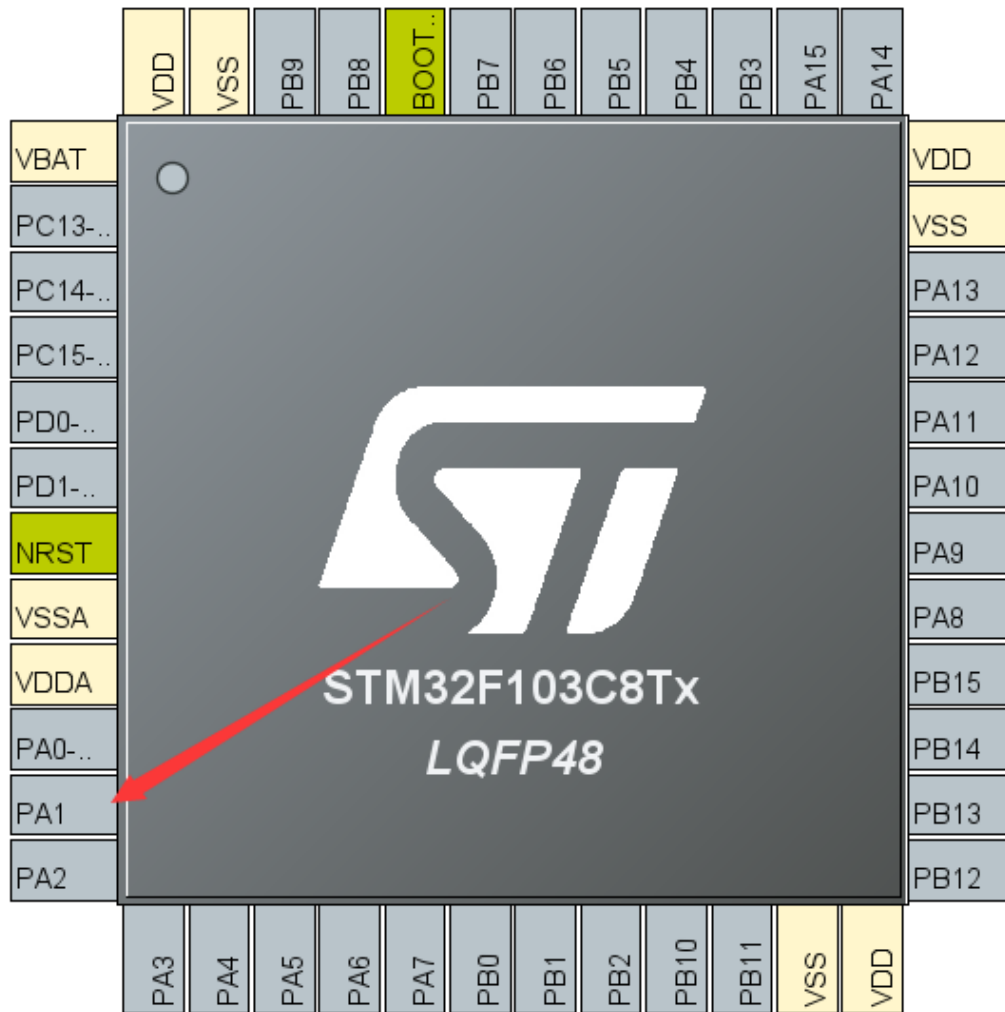


我们使用了面包板，LED小灯泡一个，1K电阻一个，导线若干，如图连接小灯泡和1K电阻，左侧导线连接我的STM32的PA1(PIN11)引脚,右侧导线连接VCC，这样就完成了最基本的电路连接

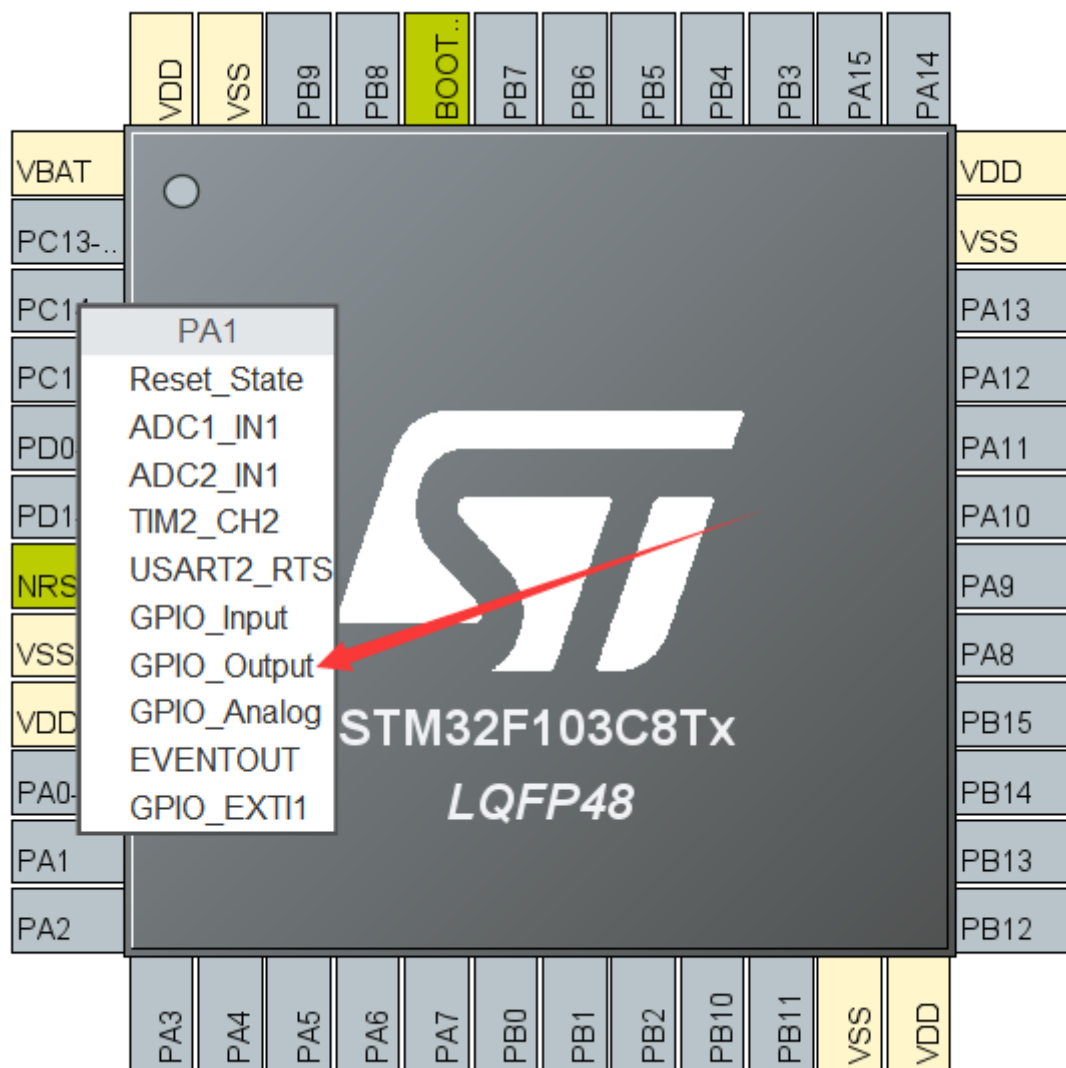
根据最基本的电路原理，当我们PA1(PIN11)引脚为低电频的话，LED就会点亮，现在开始写代码

2.代码书写

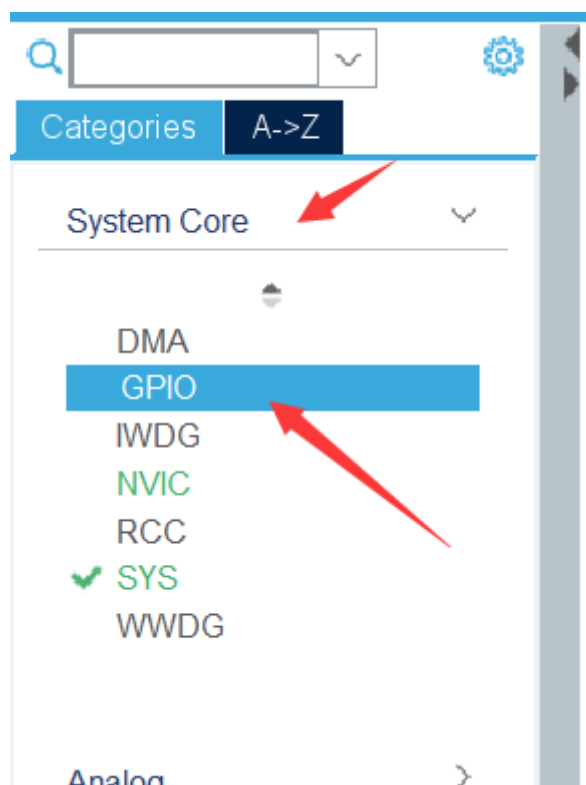
1.点击“PA1”



2.点击“GPIO_Output”



3.依次点击图中所指向的地方



4.点击PA1，以及将电平设置为低(默认为LOW)

CategoriesA->Z

System Core

DMA

GPIO

IWDG

NVIC

RCC

SYS

WWDG

Analog

Timers

Connectivity

Computing

Middleware

Configuration

Group By Peripherals

GPIO

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

| Pi... | Signal... | GPIO ... | GPIO ... | GPIO ... | Maxim... | User L... | Modifi... |
|-------|-----------|----------|-----------|-----------|----------|-----------|--------------------------|
| PA1 | n/a | Low | Output... | No pul... | Low | | <input type="checkbox"/> |

PA1 Configuration :

GPIO output level

Low

GPIO mode

Output Push Pull

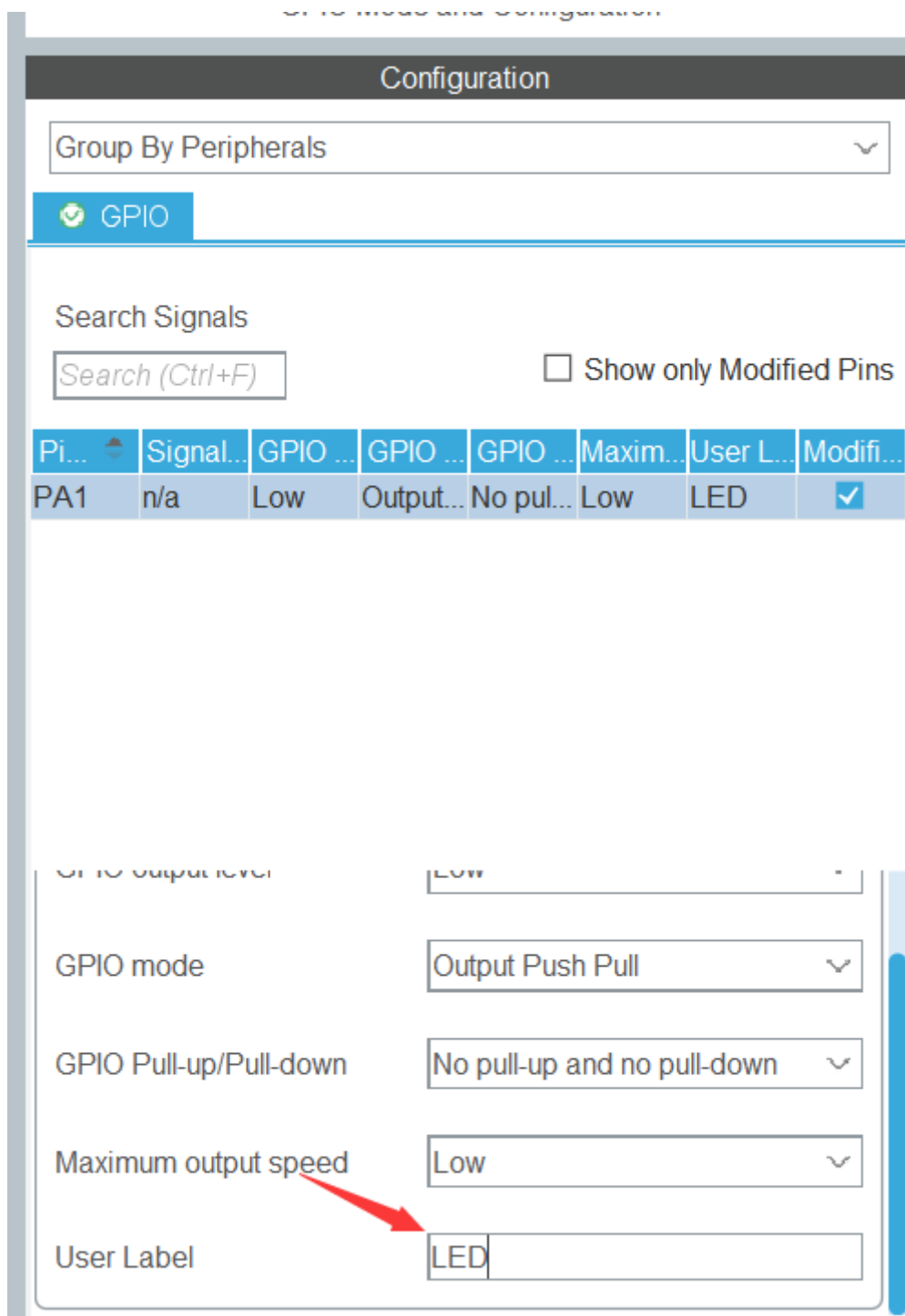
GPIO Pull-up/Pull-down

No pull-up and no pull-down

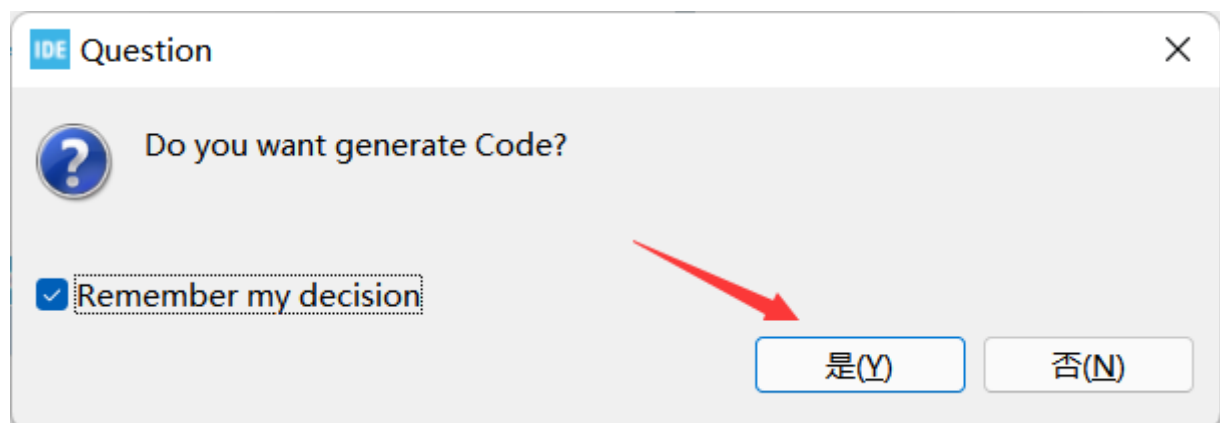
Maximum output speed

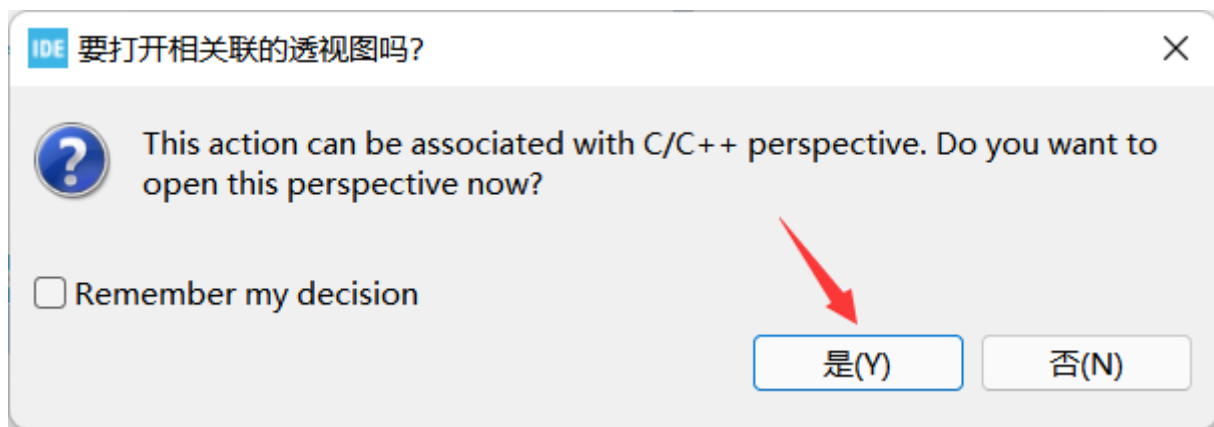
Low

5.更改标签为LED

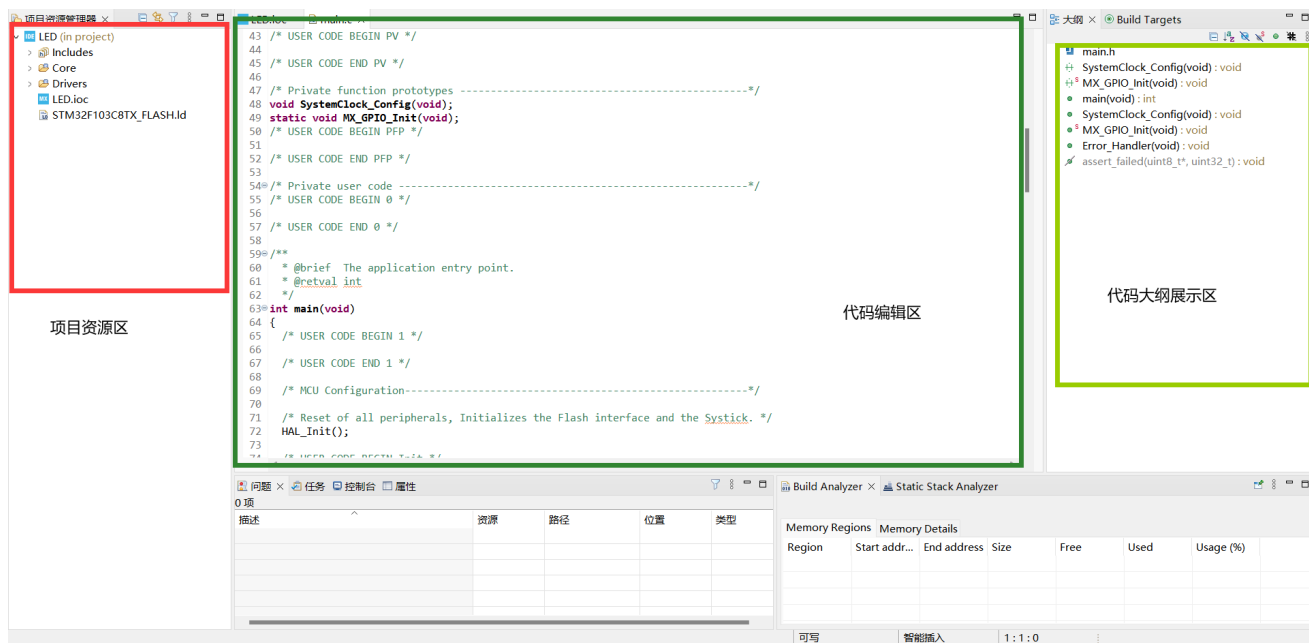


6. 键盘“Ctrl+S”保存，弹出两个对话框，都点击是，代码框架就自动生成了





7.然后就可以看到这样的3个分区



项目资源区：项目的所有文件，展示在这里

代码编辑区：用来书写代码，可以切换编辑不同文件

代码大纲区：可以看到左侧代码编辑区的代码大纲，点击可以直接专挑过去

8.(可不做)删掉无意义的注释

```

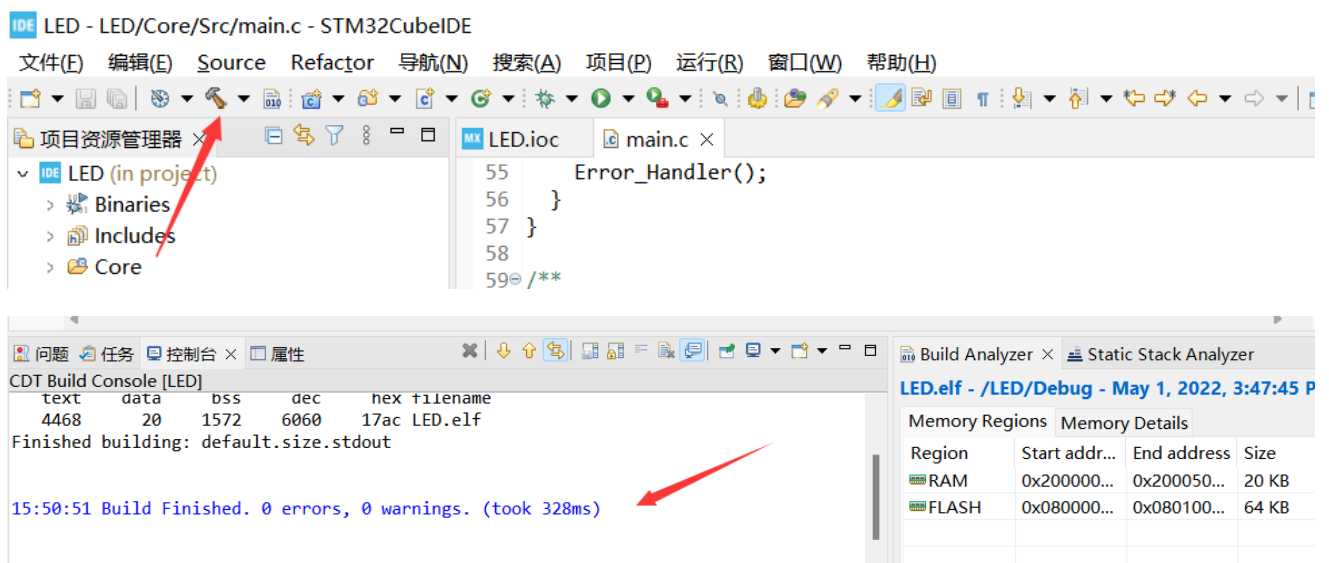
1 #include "main.h"
2
3 void SystemClock_Config(void);
4
5 static void MX_GPIO_Init(void);
6
7 int main(void)
8 {
9
10     HAL_Init();
11
12     SystemClock_Config();
13
14     MX_GPIO_Init();
15
16     while (1)
17     {
18
19     }
20 }
21
22
23 /**
24  * @brief System Clock Configuration
25  * @retval None
26  */
27 void SystemClock_Config(void)
28 {
29     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
30     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
31
32     /*Configure the main clock divider, this divides HSI by 2
33     */
34     RCC_OscInitStruct.RCC_OscInitStruct = {0};
35     RCC_OscInitStruct.PLL.PLLState = RCC_OscInitStruct.None;
36     HAL_RCC_OscConfig(&RCC_OscInitStruct);
37
38     /*Configure the system clock frequency, HSI divided by 2
39     */
40     RCC_ClkInitStruct.RCC_ClkInitStruct = {0};
41     RCC_ClkInitStruct.PLL.PLLState = RCC_ClkInitStruct.None;
42     HAL_RCC_ClockConfig(&RCC_ClkInitStruct, HAL_RCC_GetPCLK2Freq());
43
44     /*Configure the systick clock, HSI divided by 2
45     */
46     SysTick_Config(HAL_RCC_GetHCLKFreq()/1000);
47
48     /*Configure the GPIO pins
49     */
50     MX_GPIO_Init();
51
52     /*Configure the UART pins
53     */
54     MX_USART1_UART_Init();
55
56     /*Configure the I2C pins
57     */
58     MX_I2C1_Init();
59
60     /*Configure the SPI pins
61     */
62     MX_SPI1_Init();
63
64     /*Configure the ADC pins
65     */
66     MX_ADC1_Init();
67
68     /*Configure the DAC pins
69     */
70     MX_DAC1_Init();
71
72     /*Configure the TIMers
73     */
74     MX_TIM1_Init();
75
76     /*Configure the TIMers
77     */
78     MX_TIM2_Init();
79
80     /*Configure the TIMers
81     */
82     MX_TIM3_Init();
83
84     /*Configure the TIMers
85     */
86     MX_TIM4_Init();
87
88     /*Configure the TIMers
89     */
90     MX_TIM5_Init();
91
92     /*Configure the TIMers
93     */
94     MX_TIM6_Init();
95
96     /*Configure the TIMers
97     */
98     MX_TIM7_Init();
99
100    /*Configure the TIMers
101    */
102    MX_TIM8_Init();
103
104    /*Configure the TIMers
105    */
106    MX_TIM9_Init();
107
108    /*Configure the TIMers
109    */
110    MX_TIM10_Init();
111
112    /*Configure the TIMers
113    */
114    MX_TIM11_Init();
115
116    /*Configure the TIMers
117    */
118    MX_TIM12_Init();
119
120    /*Configure the TIMers
121    */
122    MX_TIM13_Init();
123
124    /*Configure the TIMers
125    */
126    MX_TIM14_Init();
127
128    /*Configure the TIMers
129    */
130    MX_TIM15_Init();
131
132    /*Configure the TIMers
133    */
134    MX_TIM16_Init();
135
136    /*Configure the TIMers
137    */
138    MX_TIM17_Init();
139
140    /*Configure the TIMers
141    */
142    MX_TIM18_Init();
143
144    /*Configure the TIMers
145    */
146    MX_TIM19_Init();
147
148    /*Configure the TIMers
149    */
150    MX_TIM20_Init();
151
152    /*Configure the TIMers
153    */
154    MX_TIM21_Init();
155
156    /*Configure the TIMers
157    */
158    MX_TIM22_Init();
159
160    /*Configure the TIMers
161    */
162    MX_TIM23_Init();
163
164    /*Configure the TIMers
165    */
166    MX_TIM24_Init();
167
168    /*Configure the TIMers
169    */
170    MX_TIM25_Init();
171
172    /*Configure the TIMers
173    */
174    MX_TIM26_Init();
175
176    /*Configure the TIMers
177    */
178    MX_TIM27_Init();
179
180    /*Configure the TIMers
181    */
182    MX_TIM28_Init();
183
184    /*Configure the TIMers
185    */
186    MX_TIM29_Init();
187
188    /*Configure the TIMers
189    */
190    MX_TIM30_Init();
191
192    /*Configure the TIMers
193    */
194    MX_TIM31_Init();
195
196    /*Configure the TIMers
197    */
198    MX_TIM32_Init();
199
200    /*Configure the TIMers
201    */
202    MX_TIM33_Init();
203
204    /*Configure the TIMers
205    */
206    MX_TIM34_Init();
207
208    /*Configure the TIMers
209    */
210    MX_TIM35_Init();
211
212    /*Configure the TIMers
213    */
214    MX_TIM36_Init();
215
216    /*Configure the TIMers
217    */
218    MX_TIM37_Init();
219
220    /*Configure the TIMers
221    */
222    MX_TIM38_Init();
223
224    /*Configure the TIMers
225    */
226    MX_TIM39_Init();
227
228    /*Configure the TIMers
229    */
230    MX_TIM40_Init();
231
232    /*Configure the TIMers
233    */
234    MX_TIM41_Init();
235
236    /*Configure the TIMers
237    */
238    MX_TIM42_Init();
239
240    /*Configure the TIMers
241    */
242    MX_TIM43_Init();
243
244    /*Configure the TIMers
245    */
246    MX_TIM44_Init();
247
248    /*Configure the TIMers
249    */
250    MX_TIM45_Init();
251
252    /*Configure the TIMers
253    */
254    MX_TIM46_Init();
255
256    /*Configure the TIMers
257    */
258    MX_TIM47_Init();
259
260    /*Configure the TIMers
261    */
262    MX_TIM48_Init();
263
264    /*Configure the TIMers
265    */
266    MX_TIM49_Init();
267
268    /*Configure the TIMers
269    */
270    MX_TIM50_Init();
271
272    /*Configure the TIMers
273    */
274    MX_TIM51_Init();
275
276    /*Configure the TIMers
277    */
278    MX_TIM52_Init();
279
280    /*Configure the TIMers
281    */
282    MX_TIM53_Init();
283
284    /*Configure the TIMers
285    */
286    MX_TIM54_Init();
287
288    /*Configure the TIMers
289    */
290    MX_TIM55_Init();
291
292    /*Configure the TIMers
293    */
294    MX_TIM56_Init();
295
296    /*Configure the TIMers
297    */
298    MX_TIM57_Init();
299
300    /*Configure the TIMers
301    */
302    MX_TIM58_Init();
303
304    /*Configure the TIMers
305    */
306    MX_TIM59_Init();
307
308    /*Configure the TIMers
309    */
310    MX_TIM60_Init();
311
312    /*Configure the TIMers
313    */
314    MX_TIM61_Init();
315
316    /*Configure the TIMers
317    */
318    MX_TIM62_Init();
319
320    /*Configure the TIMers
321    */
322    MX_TIM63_Init();
323
324    /*Configure the TIMers
325    */
326    MX_TIM64_Init();
327
328    /*Configure the TIMers
329    */
330    MX_TIM65_Init();
331
332    /*Configure the TIMers
333    */
334    MX_TIM66_Init();
335
336    /*Configure the TIMers
337    */
338    MX_TIM67_Init();
339
340    /*Configure the TIMers
341    */
342    MX_TIM68_Init();
343
344    /*Configure the TIMers
345    */
346    MX_TIM69_Init();
347
348    /*Configure the TIMers
349    */
350    MX_TIM70_Init();
351
352    /*Configure the TIMers
353    */
354    MX_TIM71_Init();
355
356    /*Configure the TIMers
357    */
358    MX_TIM72_Init();
359
360    /*Configure the TIMers
361    */
362    MX_TIM73_Init();
363
364    /*Configure the TIMers
365    */
366    MX_TIM74_Init();
367
368    /*Configure the TIMers
369    */
370    MX_TIM75_Init();
371
372    /*Configure the TIMers
373    */
374    MX_TIM76_Init();
375
376    /*Configure the TIMers
377    */
378    MX_TIM77_Init();
379
380    /*Configure the TIMers
381    */
382    MX_TIM78_Init();
383
384    /*Configure the TIMers
385    */
386    MX_TIM79_Init();
387
388    /*Configure the TIMers
389    */
390    MX_TIM80_Init();
391
392    /*Configure the TIMers
393    */
394    MX_TIM81_Init();
395
396    /*Configure the TIMers
397    */
398    MX_TIM82_Init();
399
400    /*Configure the TIMers
401    */
402    MX_TIM83_Init();
403
404    /*Configure the TIMers
405    */
406    MX_TIM84_Init();
407
408    /*Configure the TIMers
409    */
410    MX_TIM85_Init();
411
412    /*Configure the TIMers
413    */
414    MX_TIM86_Init();
415
416    /*Configure the TIMers
417    */
418    MX_TIM87_Init();
419
420    /*Configure the TIMers
421    */
422    MX_TIM88_Init();
423
424    /*Configure the TIMers
425    */
426    MX_TIM89_Init();
427
428    /*Configure the TIMers
429    */
430    MX_TIM90_Init();
431
432    /*Configure the TIMers
433    */
434    MX_TIM91_Init();
435
436    /*Configure the TIMers
437    */
438    MX_TIM92_Init();
439
440    /*Configure the TIMers
441    */
442    MX_TIM93_Init();
443
444    /*Configure the TIMers
445    */
446    MX_TIM94_Init();
447
448    /*Configure the TIMers
449    */
450    MX_TIM95_Init();
451
452    /*Configure the TIMers
453    */
454    MX_TIM96_Init();
455
456    /*Configure the TIMers
457    */
458    MX_TIM97_Init();
459
460    /*Configure the TIMers
461    */
462    MX_TIM98_Init();
463
464    /*Configure the TIMers
465    */
466    MX_TIM99_Init();
467
468    /*Configure the TIMers
469    */
470    MX_TIM100_Init();
471
472    /*Configure the TIMers
473    */
474    MX_TIM101_Init();
475
476    /*Configure the TIMers
477    */
478    MX_TIM102_Init();
479
480    /*Configure the TIMers
481    */
482    MX_TIM103_Init();
483
484    /*Configure the TIMers
485    */
486    MX_TIM104_Init();
487
488    /*Configure the TIMers
489    */
490    MX_TIM105_Init();
491
492    /*Configure the TIMers
493    */
494    MX_TIM106_Init();
495
496    /*Configure the TIMers
497    */
498    MX_TIM107_Init();
499
500    /*Configure the TIMers
501    */
502    MX_TIM108_Init();
503
504    /*Configure the TIMers
505    */
506    MX_TIM109_Init();
507
508    /*Configure the TIMers
509    */
510    MX_TIM110_Init();
511
512    /*Configure the TIMers
513    */
514    MX_TIM111_Init();
515
516    /*Configure the TIMers
517    */
518    MX_TIM112_Init();
519
520    /*Configure the TIMers
521    */
522    MX_TIM113_Init();
523
524    /*Configure the TIMers
525    */
526    MX_TIM114_Init();
527
528    /*Configure the TIMers
529    */
530    MX_TIM115_Init();
531
532    /*Configure the TIMers
533    */
534    MX_TIM116_Init();
535
536    /*Configure the TIMers
537    */
538    MX_TIM117_Init();
539
540    /*Configure the TIMers
541    */
542    MX_TIM118_Init();
543
544    /*Configure the TIMers
545    */
546    MX_TIM119_Init();
547
548    /*Configure the TIMers
549    */
550    MX_TIM120_Init();
551
552    /*Configure the TIMers
553    */
554    MX_TIM121_Init();
555
556    /*Configure the TIMers
557    */
558    MX_TIM122_Init();
559
560    /*Configure the TIMers
561    */
562    MX_TIM123_Init();
563
564    /*Configure the TIMers
565    */
566    MX_TIM124_Init();
567
568    /*Configure the TIMers
569    */
570    MX_TIM125_Init();
571
572    /*Configure the TIMers
573    */
574    MX_TIM126_Init();
575
576    /*Configure the TIMers
577    */
578    MX_TIM127_Init();
579
580    /*Configure the TIMers
581    */
582    MX_TIM128_Init();
583
584    /*Configure the TIMers
585    */
586    MX_TIM129_Init();
587
588    /*Configure the TIMers
589    */
590    MX_TIM130_Init();
591
592    /*Configure the TIMers
593    */
594    MX_TIM131_Init();
595
596    /*Configure the TIMers
597    */
598    MX_TIM132_Init();
599
600    /*Configure the TIMers
601    */
602    MX_TIM133_Init();
603
604    /*Configure the TIMers
605    */
606    MX_TIM134_Init();
607
608    /*Configure the TIMers
609    */
610    MX_TIM135_Init();
611
612    /*Configure the TIMers
613    */
614    MX_TIM136_Init();
615
616    /*Configure the TIMers
617    */
618    MX_TIM137_Init();
619
620    /*Configure the TIMers
621    */
622    MX_TIM138_Init();
623
624    /*Configure the TIMers
625    */
626    MX_TIM139_Init();
627
628    /*Configure the TIMers
629    */
630    MX_TIM140_Init();
631
632    /*Configure the TIMers
633    */
634    MX_TIM141_Init();
635
636    /*Configure the TIMers
637    */
638    MX_TIM142_Init();
639
640    /*Configure the TIMers
641    */
642    MX_TIM143_Init();
643
644    /*Configure the TIMers
645    */
646    MX_TIM144_Init();
647
648    /*Configure the TIMers
649    */
650    MX_TIM145_Init();
651
652    /*Configure the TIMers
653    */
654    MX_TIM146_Init();
655
656    /*Configure the TIMers
657    */
658    MX_TIM147_Init();
659
660    /*Configure the TIMers
661    */
662    MX_TIM148_Init();
663
664    /*Configure the TIMers
665    */
666    MX_TIM149_Init();
667
668    /*Configure the TIMers
669    */
670    MX_TIM150_Init();
671
672    /*Configure the TIMers
673    */
674    MX_TIM151_Init();
675
676    /*Configure the TIMers
677    */
678    MX_TIM152_Init();
679
680    /*Configure the TIMers
681    */
682    MX_TIM153_Init();
683
684    /*Configure the TIMers
685    */
686    MX_TIM154_Init();
687
688    /*Configure the TIMers
689    */
690    MX_TIM155_Init();
691
692    /*Configure the TIMers
693    */
694    MX_TIM156_Init();
695
696    /*Configure the TIMers
697    */
698    MX_TIM157_Init();
699
700    /*Configure the TIMers
701    */
702    MX_TIM158_Init();
703
704    /*Configure the TIMers
705    */
706    MX_TIM159_Init();
707
708    /*Configure the TIMers
709    */
710    MX_TIM160_Init();
711
712    /*Configure the TIMers
713    */
714    MX_TIM161_Init();
715
716    /*Configure the TIMers
717    */
718    MX_TIM162_Init();
719
720    /*Configure the TIMers
721    */
722    MX_TIM163_Init();
723
724    /*Configure the TIMers
725    */
726    MX_TIM164_Init();
727
728    /*Configure the TIMers
729    */
730    MX_TIM165_Init();
731
732    /*Configure the TIMers
733    */
734    MX_TIM166_Init();
735
736    /*Configure the TIMers
737    */
738    MX_TIM167_Init();
739
740    /*Configure the TIMers
741    */
742    MX_TIM168_Init();
743
744    /*Configure the TIMers
745    */
746    MX_TIM169_Init();
747
748    /*Configure the TIMers
749    */
750    MX_TIM170_Init();
751
752    /*Configure the TIMers
753    */
754    MX_TIM171_Init();
755
756    /*Configure the TIMers
757    */
758    MX_TIM172_Init();
759
760    /*Configure the TIMers
761    */
762    MX_TIM173_Init();
763
764    /*Configure the TIMers
765    */
766    MX_TIM174_Init();
767
768    /*Configure the TIMers
769    */
770    MX_TIM175_Init();
771
772    /*Configure the TIMers
773    */
774    MX_TIM176_Init();
775
776    /*Configure the TIMers
777    */
778    MX_TIM177_Init();
779
780    /*Configure the TIMers
781    */
782    MX_TIM178_Init();
783
784    /*Configure the TIMers
785    */
786    MX_TIM179_Init();
787
788    /*Configure the TIMers
789    */
790    MX_TIM180_Init();
791
792    /*Configure the TIMers
793    */
794    MX_TIM181_Init();
795
796    /*Configure the TIMers
797    */
798    MX_TIM182_Init();
799
800    /*Configure the TIMers
801    */
802    MX_TIM183_Init();
803
804    /*Configure the TIMers
805    */
806    MX_TIM184_Init();
807
808    /*Configure the TIMers
809    */
810    MX_TIM185_Init();
811
812    /*Configure the TIMers
813    */
814    MX_TIM186_Init();
815
816    /*Configure the TIMers
817    */
818    MX_TIM187_Init();
819
820    /*Configure the TIMers
821    */
822    MX_TIM188_Init();
823
824    /*Configure the TIMers
825    */
826    MX_TIM189_Init();
827
828    /*Configure the TIMers
829    */
830    MX_TIM190_Init();
831
832    /*Configure the TIMers
833    */
834    MX_TIM191_Init();
835
836    /*Configure the TIMers
837    */
838    MX_TIM192_Init();
839
840    /*Configure the TIMers
841    */
842    MX_TIM193_Init();
843
844    /*Configure the TIMers
845    */
846    MX_TIM194_Init();
847
848    /*Configure the TIMers
849    */
850    MX_TIM195_Init();
851
852    /*Configure the TIMers
853    */
854    MX_TIM196_Init();
855
856    /*Configure the TIMers
857    */
858    MX_TIM197_Init();
859
860    /*Configure the TIMers
861    */
862    MX_TIM198_Init();
863
864    /*Configure the TIMers
865    */
866    MX_TIM199_Init();
867
868    /*Configure the TIMers
869    */
870    MX_TIM200_Init();
871
872    /*Configure the TIMers
873    */
874    MX_TIM201_Init();
875
876    /*Configure the TIMers
877    */
878    MX_TIM202_Init();
879
880    /*Configure the TIMers
881    */
882    MX_TIM203_Init();
883
884    /*Configure the TIMers
885    */
886    MX_TIM204_Init();
887
888    /*Configure the TIMers
889    */
890    MX_TIM205_Init();
891
892    /*Configure the TIMers
893    */
894    MX_TIM206_Init();
895
896    /*Configure the TIMers
897    */
898    MX_TIM207_Init();
899
900    /*Configure the TIMers
901    */
902    MX_TIM208_Init();
903
904    /*Configure the TIMers
905    */
906    MX_TIM209_Init();
907
908    /*Configure the TIMers
909    */
909

```

然后就是这样！代码清爽了很多

可以看到，系统给我们做好了一些初始化，我们暂时先不用去理解，后面的课程会慢慢为大家讲述

9.我们点击“锤子”的图标，也就是“编译”，或者键盘快捷键敲击“Ctrl+B”进行编译

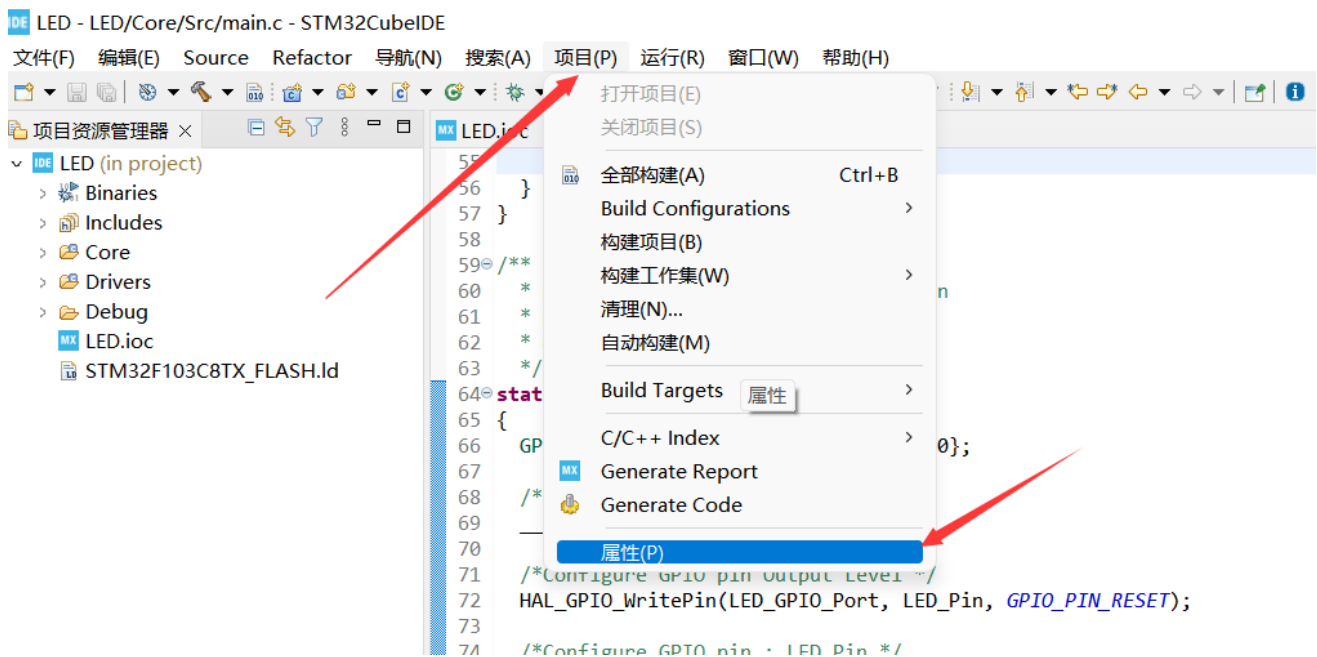


0错误，0警告

OK，编译完成

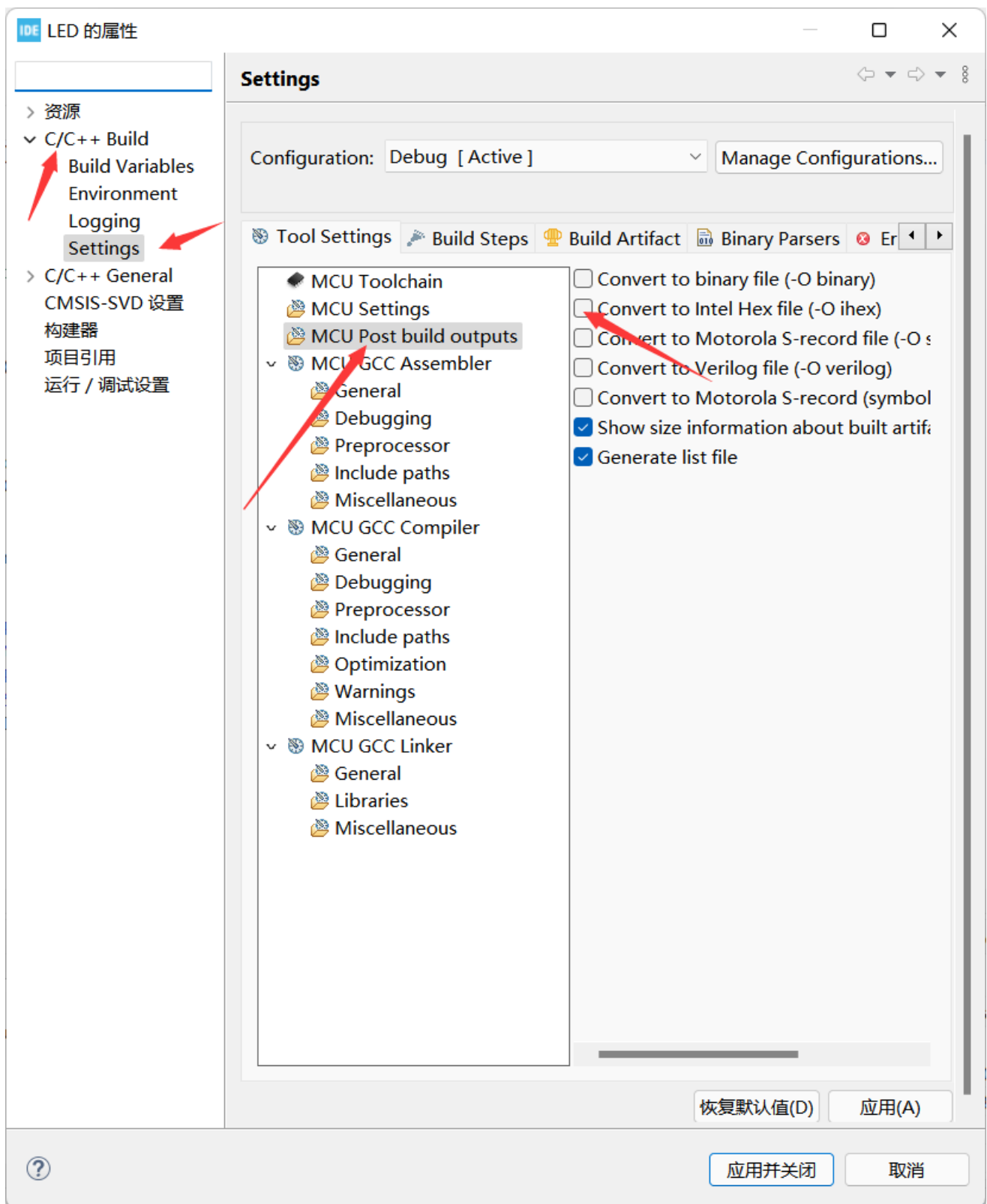
3.程序烧录

1.依次点击“项目”，然后是“属性”



2.从左往右一次点击箭头所指向的地方

依次分别为“C/C++Build”，“Setting”，“MCU Post build outputs”，“Convert to intel Hex file(-O ihex)”



勾选上以后，这样才会生成HEX的烧录文件

3.再次点击编译，就出现了LED.HEX

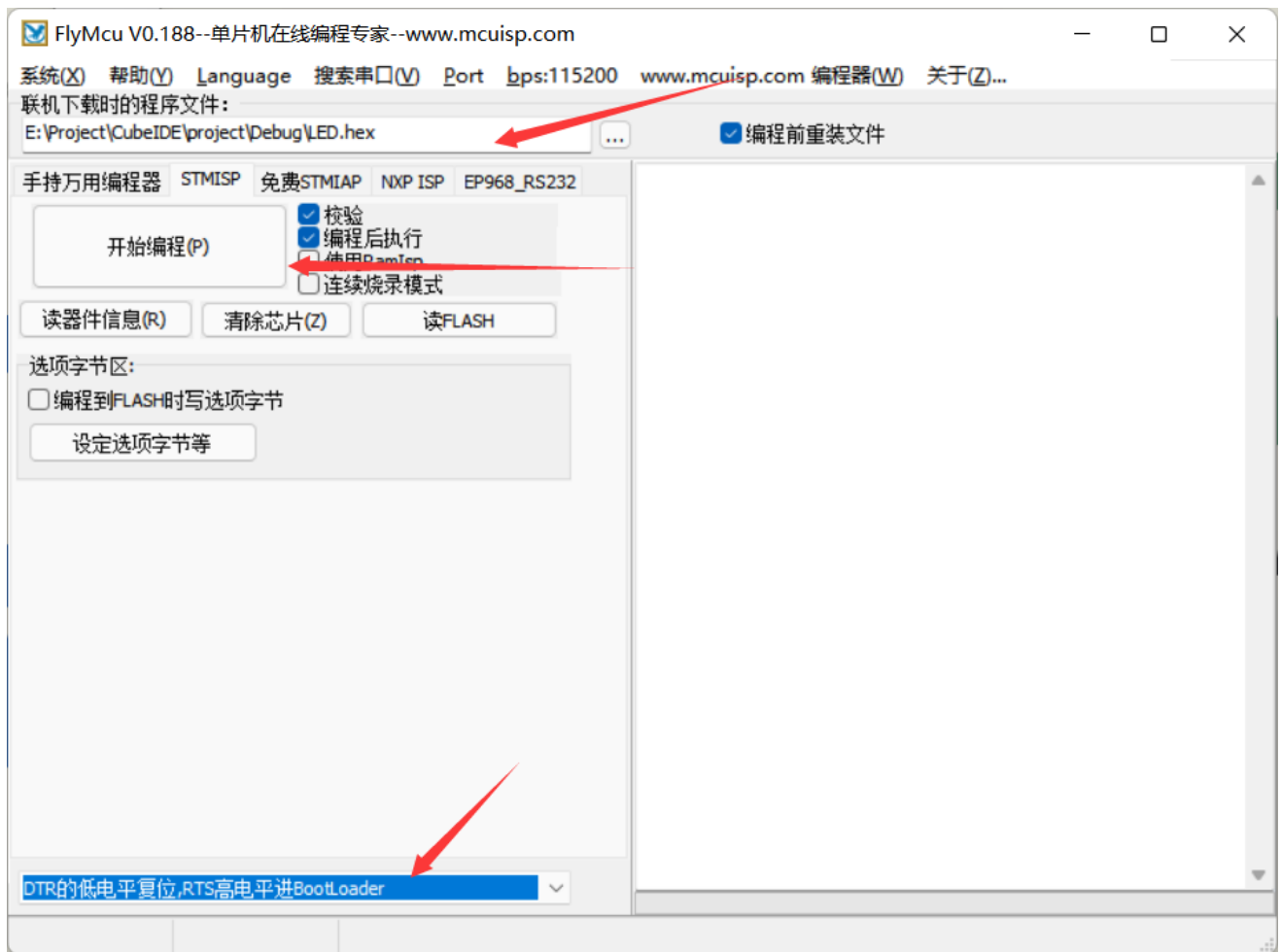
| 名称 | 修改日期 | 类型 | 大小 |
|------------------------|----------------|--------------|----|
| .settings | 2022/5/1 11:03 | 文件夹 | |
| Core | 2022/5/1 11:03 | 文件夹 | |
| Debug | 2022/5/1 15:47 | 文件夹 | |
| Drivers | 2022/5/1 11:02 | 文件夹 | |
| .cproject | 2022/5/1 15:32 | CPROJECT 文件 | |
| .mxproject | 2022/5/1 15:32 | MXPROJECT 文件 | |
| .project | 2022/5/1 11:03 | PROJECT 文件 | |
| LED.ioc | 2022/5/1 15:32 | IOC 文件 | |
| STM32F103C8TX_FLASH.ld | 2022/5/1 15:32 | LD 文件 | |

| 名称 | 修改日期 | 类型 | 大小 |
|--------------|----------------|--------------|----|
| Core | 2022/5/1 15:47 | 文件夹 | |
| Drivers | 2022/5/1 15:47 | 文件夹 | |
| LED.elf | 2022/5/1 18:04 | ELF 文件 | |
| LED.hex | 2022/5/1 18:04 | HEX 文件 | |
| LED.list | 2022/5/1 18:04 | LIST 文件 | |
| LED.map | 2022/5/1 18:04 | MAP 文件 | |
| makefile | 2022/5/1 18:04 | 文件 | |
| objects.list | 2022/5/1 15:47 | LIST 文件 | |
| objects.mk | 2022/5/1 15:47 | Makefile 源文件 | |
| sources.mk | 2022/5/1 18:04 | Makefile 源文件 | |

4.打开FlyMcu.exe

小伙伴可以在阿熊的配套资料里自行下载哦！

5.选择对应的文件，以及烧录方式，最后点击开始编程，等待烧录完成就算结束了！



4.观察现象，以及简单的介绍实现原理

1.现象展示

我们按照前面的设计，弄好了我们的电路，烧录程序，然后RST一下，就可以看到，一下的实验现象

我们而可以看到小灯泡成功亮起来了，任务就算大功告成了，接下来是我们的原理部分

2.原理简介

```
#include "main.h"

void SystemClock_Config(void);

static void MX_GPIO_Init(void);

int main(void)
{
    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();
```



```

while (1)
{

}

}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Initializes the RCC Oscillators according to the specified
parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /** Initializes the CPU, AHB and APB buses clocks
    */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                   |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0) !=
HAL_OK)
    {
        Error_Handler();
    }
}

/**
 * @brief GPIO Initialization Function
 * @param None

```

```

    * @retval None
    */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOA_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : LED_Pin */
    GPIO_InitStruct.Pin = LED_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LED_GPIO_Port, &GPIO_InitStruct);

}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return
state */
    __disable_irq();
    while (1)
    {

    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line
number
 *
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */

```

```

    /* User can add his own implementation to report the file name and
    line number,
        ex: printf("Wrong parameters value: file %s on line %d\r\n", file,
    line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

以上是main.c的所有代码！

由于是刚入门，就说一下大概实现的功能

首先头文件的引入，以及两个函数声明

```

#include "main.h"//头文件的引入

void SystemClock_Config(void); //时钟初始化的函数声明

static void MX_GPIO_Init(void); //GPIO的初始化函数声明

```

时钟初始化函数是一些有关于时钟的配置，就像我们人类规定时间单位

GPIO初始化函数就是我们之前在图形化界面，进行的配置，他自动生成的代码

```

static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0}; //结构体定义

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOA_CLK_ENABLE(); //时钟初始化

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_RESET); //设置LED_Pin
    为RESET(0)

    /*Configure GPIO pin : LED_Pin */
    GPIO_InitStructure.Pin = LED_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LED_GPIO_Port, &GPIO_InitStructure); //引脚初始化
}

```

可以看到，首先是定义了结构体，他有很多属性，这里设置了Pin(第几引脚)，Mode(模式),PULL(上下拉状态),Speed(速度)

这里的LED_Pin就是我们前面设置好的PA1，LED是我们命的名，后面的_Pin是系统自动添加的

模式选择的是“推挽输出”，这里先不解释他的意思，后续讲到GPIO是会细讲

最后是HAL_GPIO_Init(LED_GPIO_Port, &GPIO_InitStruct)的意思就是讲PA1进行初始化

最后是我们的主函数

```
int main(void)
{
    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();

    while (1)
    {

    }

}
```

主函数内容就比较简单了，初始化我们的库文件，系统时钟，以及GPIO，其他的暂时都不需要

由于在GPIO的函数中设置了，PA1初始状态为低电平，所以只要我们的单片机一通电，就可以让我们的LED点亮了

以上我们就完成了STM32的基础部分了，我们介绍了STM32一些基本的东西，以及CBUEIDE的安装汉化，还成功的点亮了LED，接下跟着阿熊继续往下学吧！