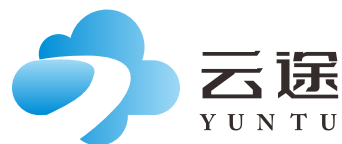


# 应用笔记

AN0058

---

云途 MCU Keil 工程搭建指南



# 目录

1	Keil 版本 .....	3
2	工程搭建 .....	3
2.1	新建工程 .....	3
2.2	编译选项及配置 .....	7
3	工程修改 .....	13
4	__main 去除 .....	15
4.1	linker 配置修改 .....	15
4.2	汇编文件修改 .....	15
4.3	链接文件修改 .....	15

## 历史版本

版本号	日期	修改
1.0	2023-03-03	初始版本

# 1 Keil 版本

云途推荐使用 Keil  $\mu$ Vision V5.30 及以上版本，使用 arm compiler 6 编译器

## 2 工程搭建

### 2.1 新建工程

1. Project->New  $\mu$ Vision Project

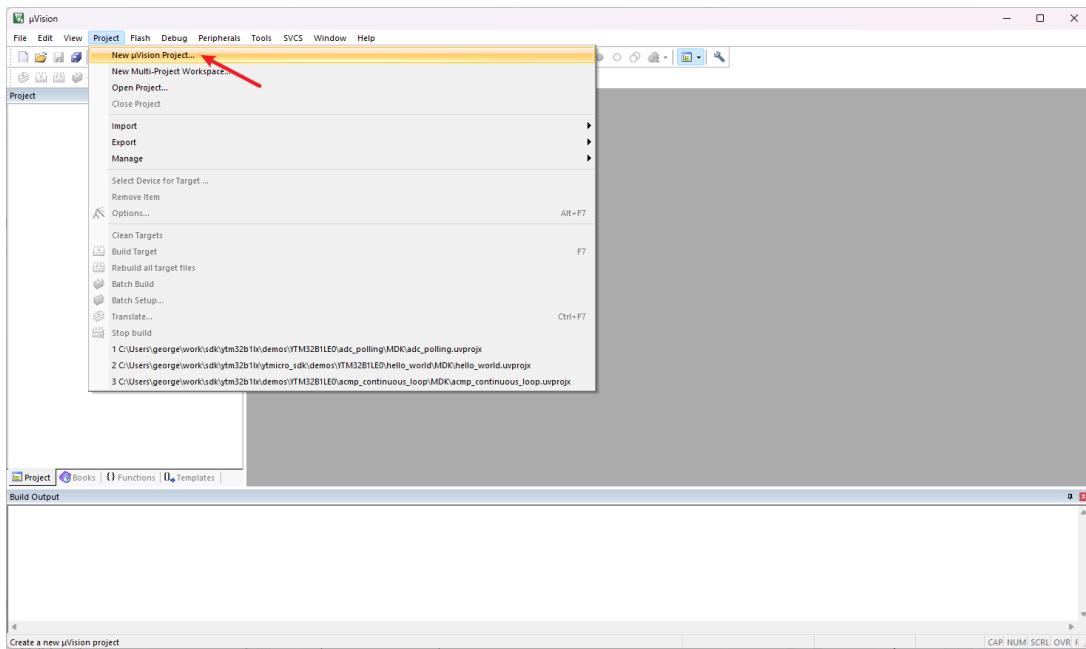


图 1: 新建工程

2. 选择相应的目录，并给工程命名

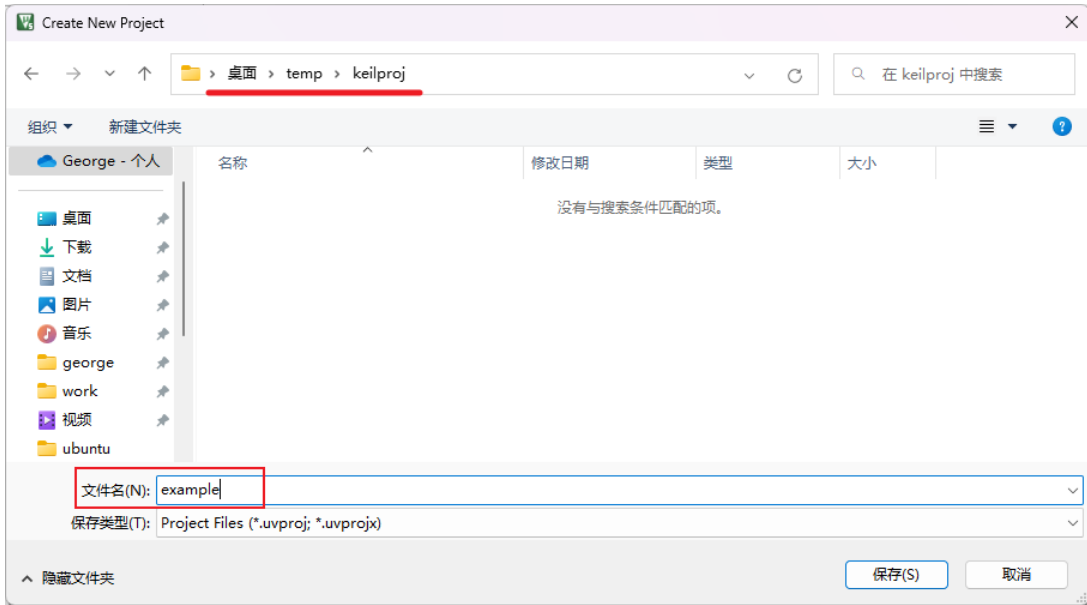


图 2: 命名工程

3. 选择所使用的云途 MCU 型号 (例:YTM32B1LE05H0MLHT), 然后点击 ok

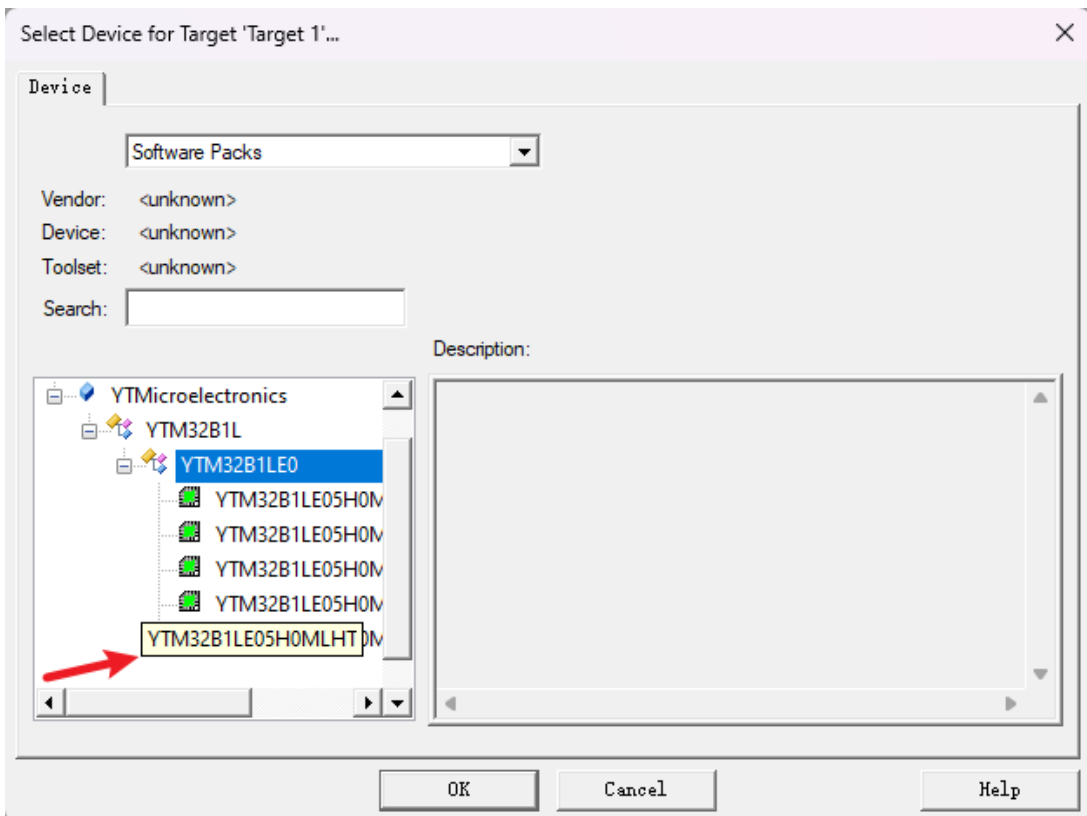


图 3: 选择器件

4. 之后会出现如下对话框，目前暂时不支持 Keil 的这套方式选取模块代码，因此略过，点击 OK

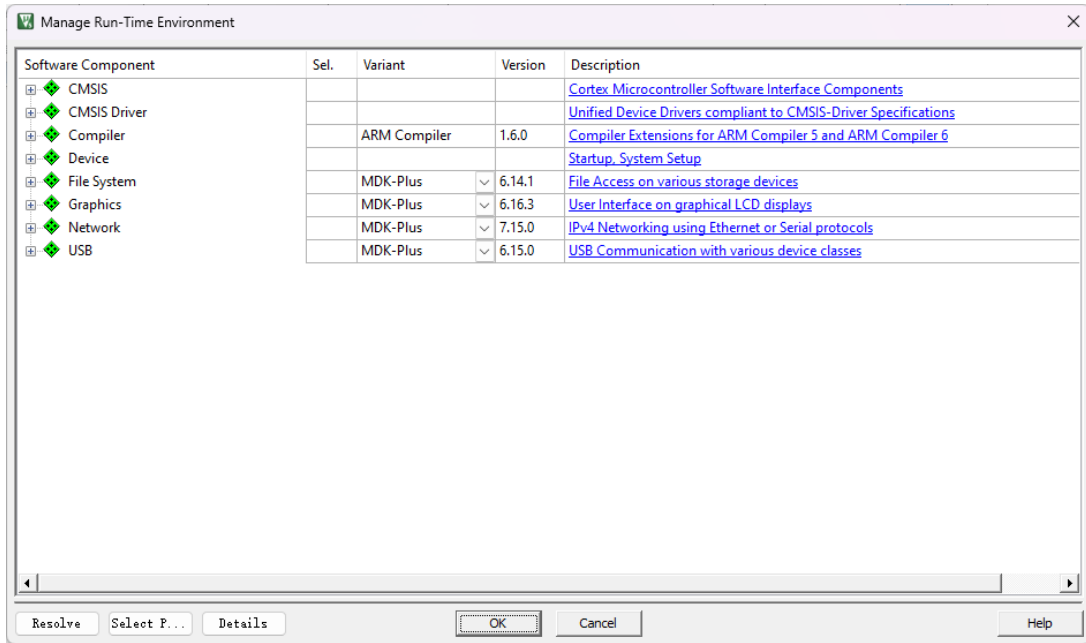


图 4: 忽略 Run-time Env

点击 ok 以后出现如下界面

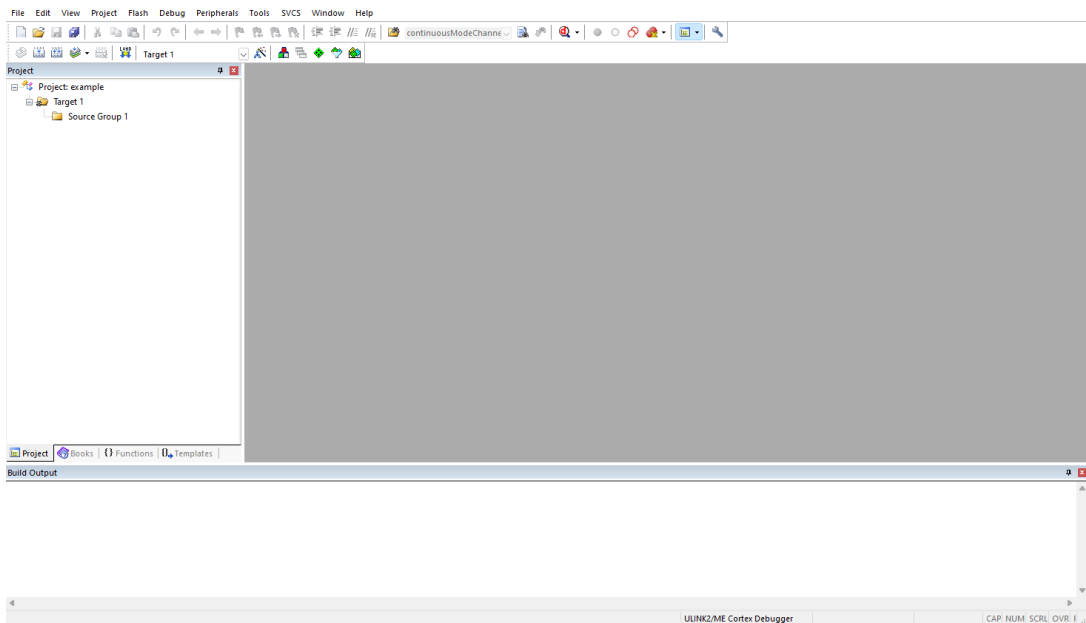


图 5: 初步建立

5. 可以给 Target 1 重命名为 Debug，然后添加 group 和 file。添加 group 如图，在 Debug 上单击右键

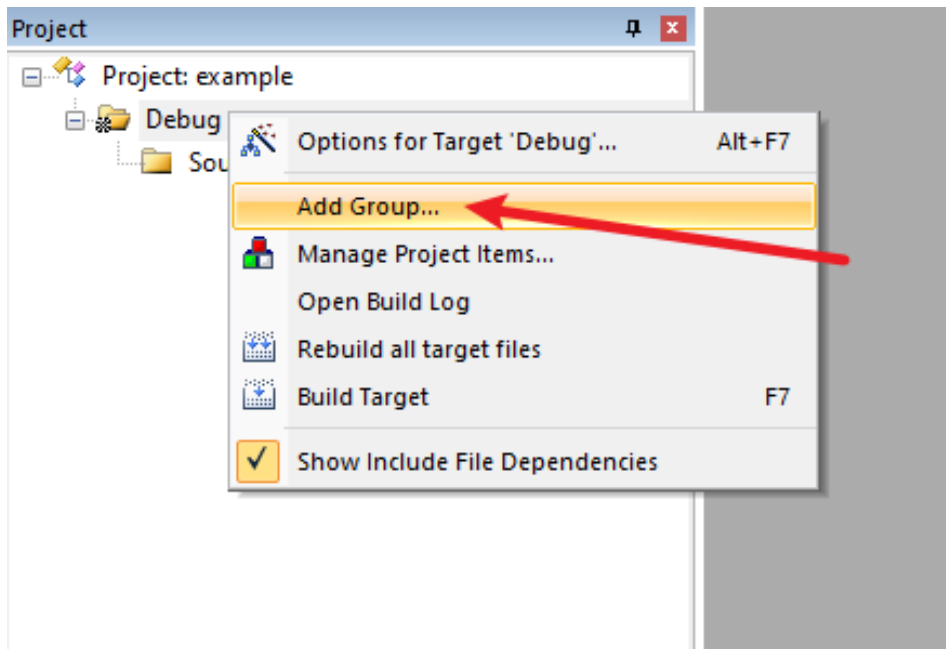


图 6: 命名目标并添加组

这里我们添加 board, app, drivers 和 startup 4 个 group, 然后分别往里添加所需的 c 文件, 方式如下: 在 group 上单击右键

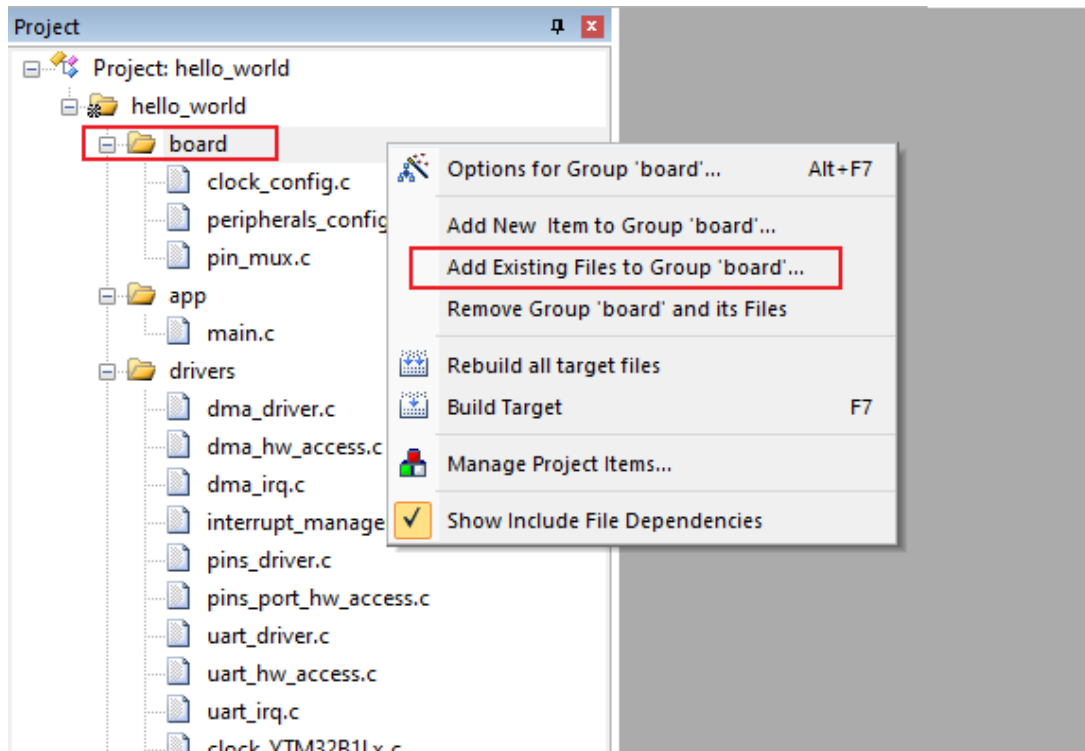


图 7: 添加文件

如果是添加新的文件就点击'Add New Item to Group 'board'。具体文件可以参考云途 SDK 包里的相应的 demo。如果有需要自己的代码, 那么 startup group 下的 3 个文件 startup.c, system\_<chip>.c 以及 <chip>\_startup\_armc.S 是必需的, 当然用户可参考其内容自己编写, 也可以直接使用。链接文件使用 SDK 包下面提供的 scf 文件, 位置在 platform/devices/YTM32B1LE0/linker/armc/YTM32B1LE0\_flash.scf。如图是添加好的工程目录结构

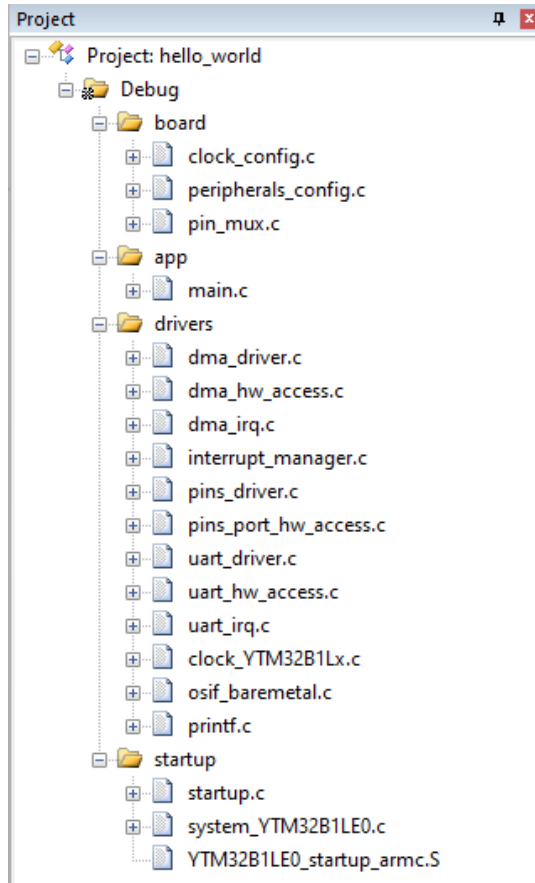


图 8: 工程目录结构

## 2.2 编译选项及配置

c 文件和汇编文件添加好后，需要指定头文件的位置，配置如下。先点击类似魔术棒的按钮



图 9: 选项配置

点击之后如下图，选择 C/C++ (AC6)，然后点击 Include Paths 的 browse 按钮，如图

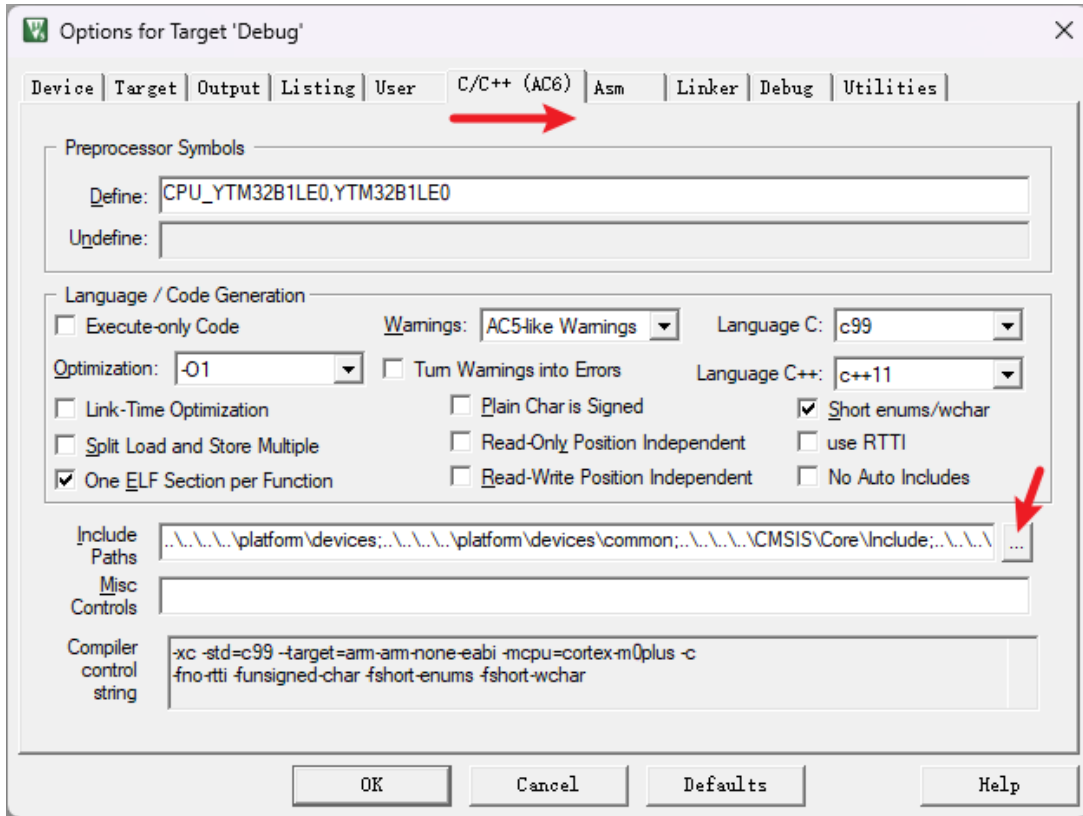


图 10: c 参数配置

然后点击 Include Paths 按钮之后如图把所用到的头文件的目录加入进去

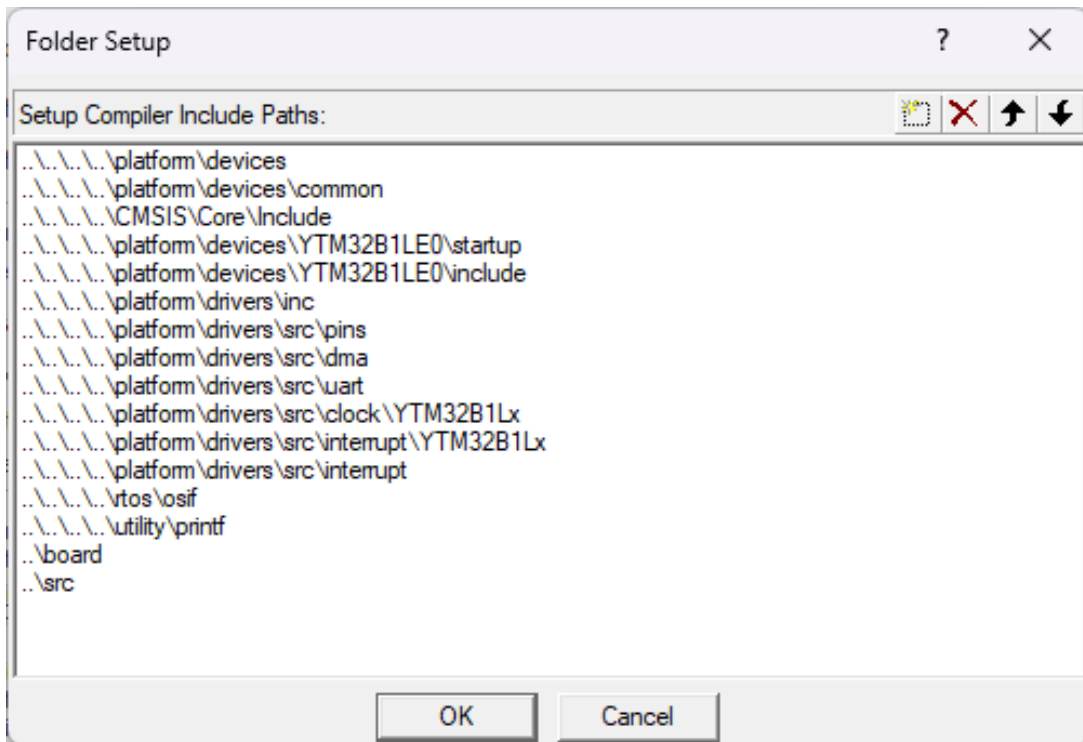


图 11: 添加 include 目录

Preprocessor 和 Optimization 配置如图

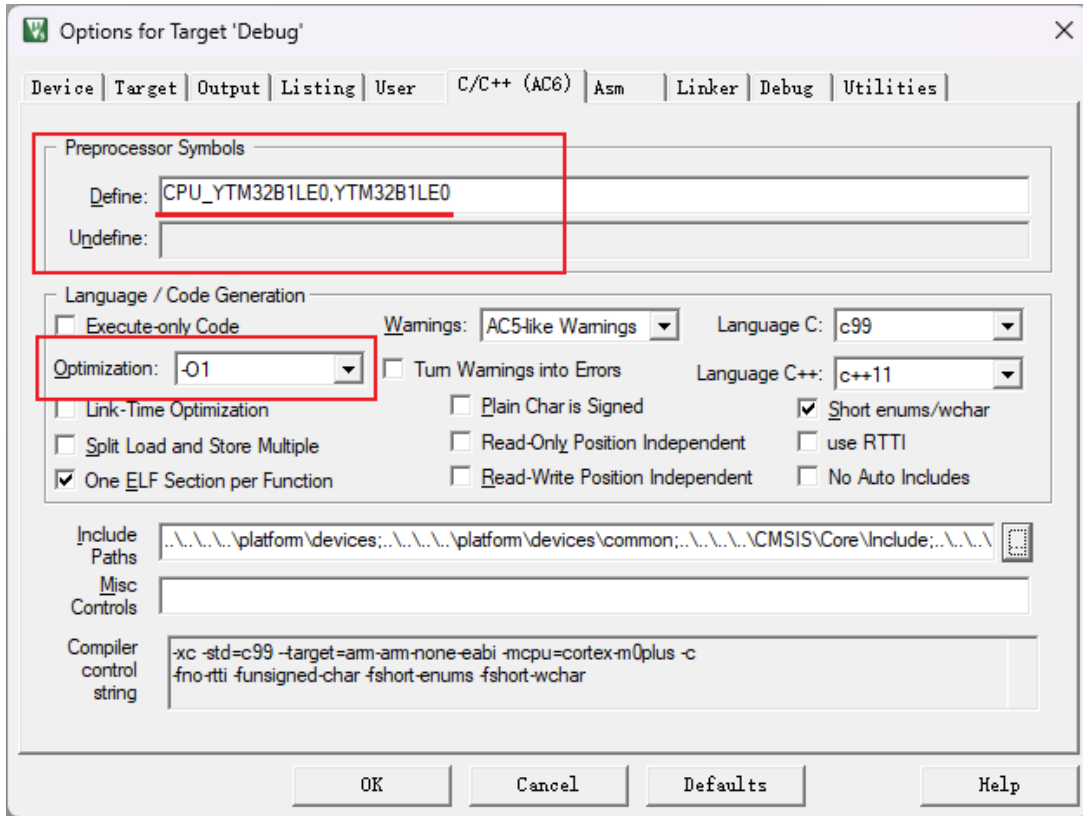


图 12: 预处理和优化等级

Asm 配置如图，选择 armclang(Auto Select)

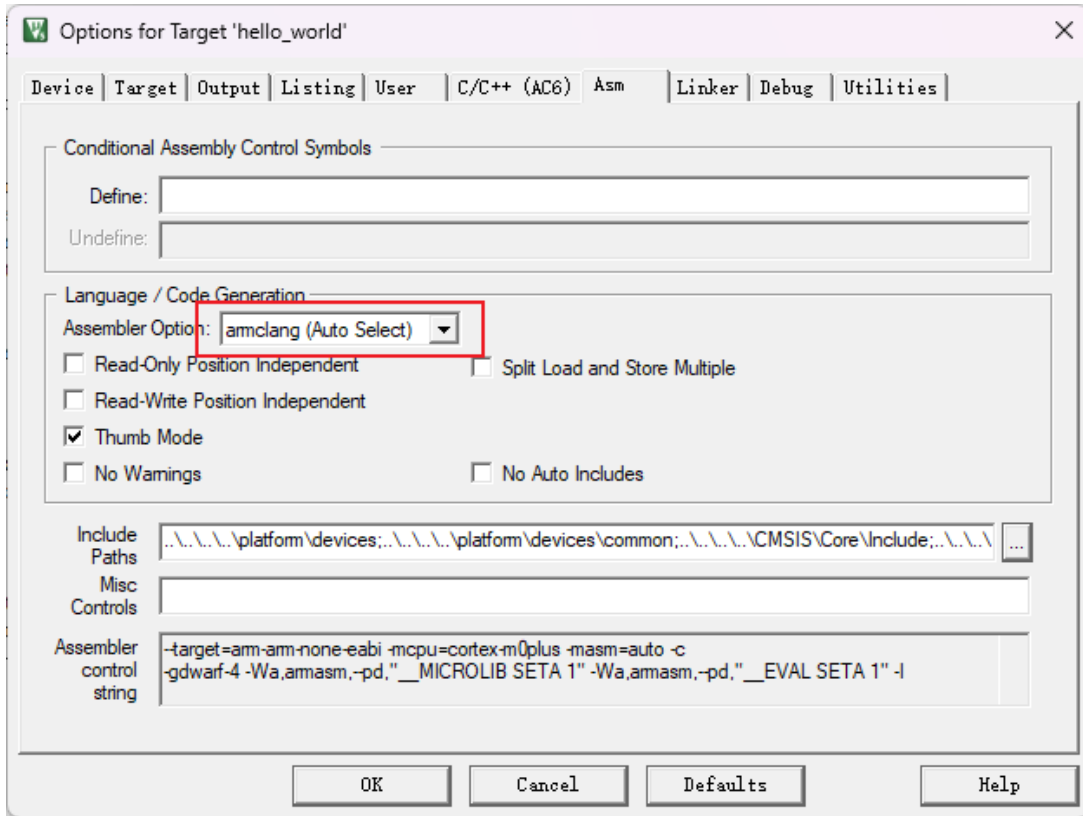


图 13: 汇编器选择

linker 选择以上提到的路径中的 scatter file

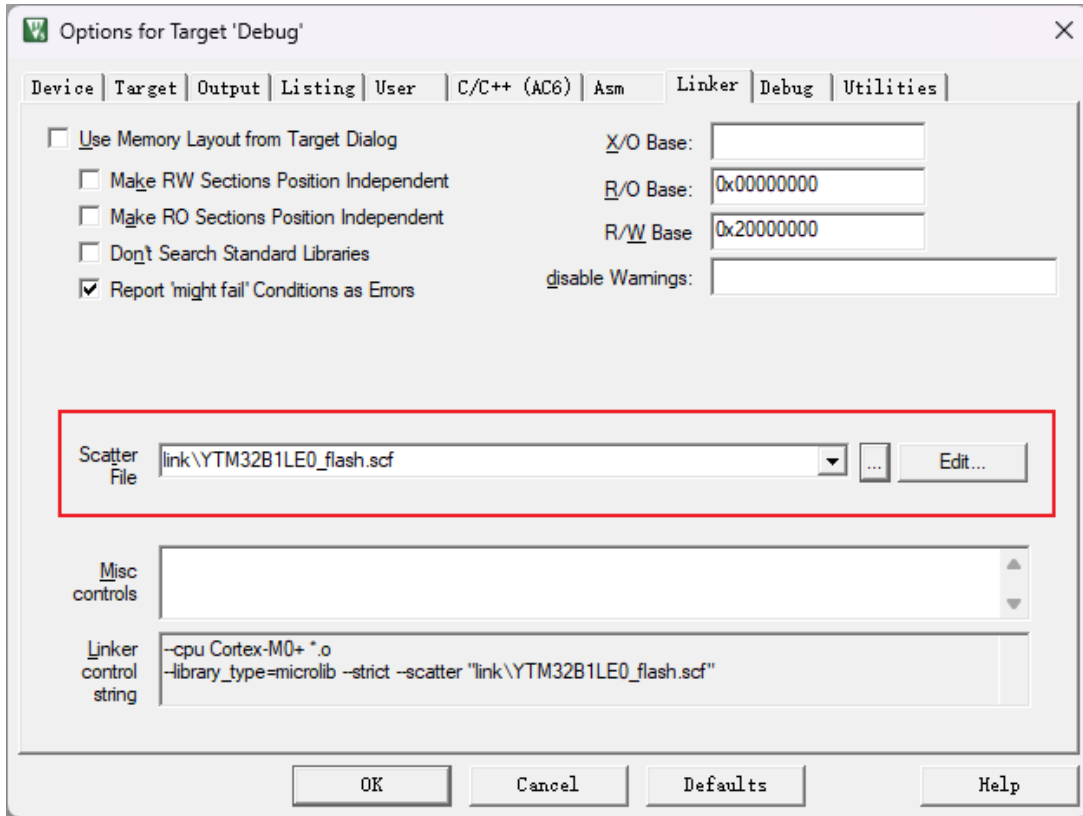


图 14: 链接文件

debugger 的选择如下，前提是安装好云途提供的 pack 包，包里提供了下载算法，用户可以直接下载调试

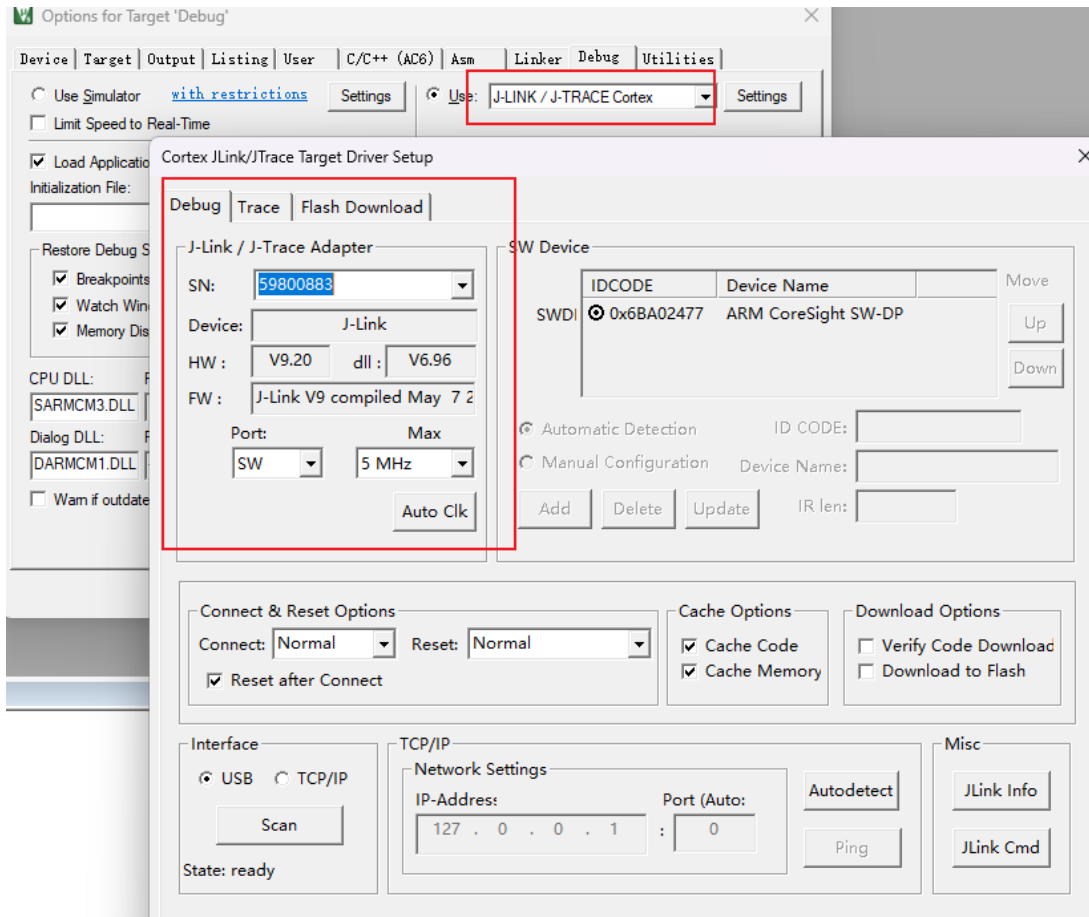


图 15: 调试配置

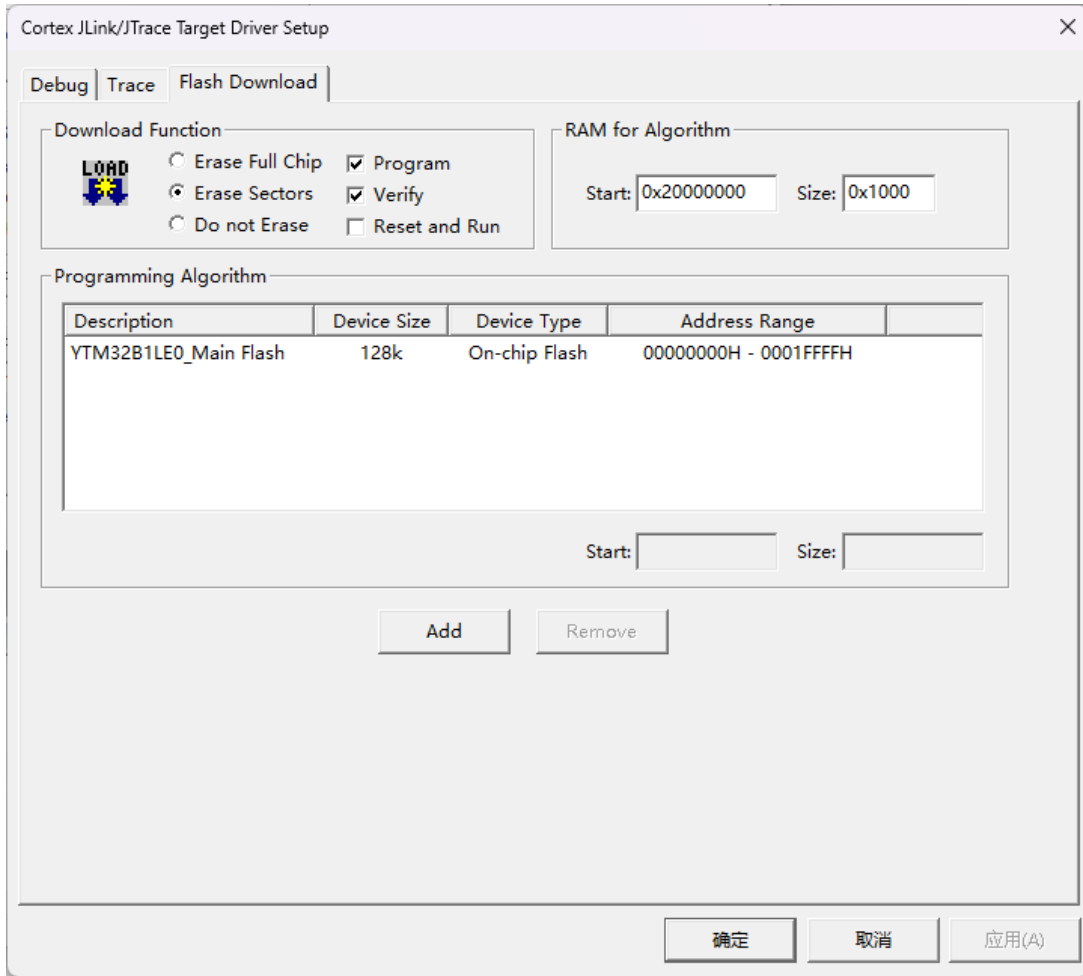


图 16: 下载算法

### 3 工程修改

如有移植的需求，那么 startup 里面 3 个文件是必需的，另外加上 main.c 和链接文件就可以正常编译了；如果需要修改 data 段的大小，可以修改 m\_data\_end 的值，如果需要修改栈的大小，可以修改 Stack\_Size 的大小

```

27 #define __ram_vector_table_size__      0x000000c0
28
29 #define m_interrupts_start              0x00000000
30 #define m_interrupts_size              0x000000c0
31
32 #define m_text_start                    0x000000c0
33 #define m_text_end                      0x0001ffff
34 #define m_text_size                     m_text_end - m_text_start + 1
35
36 #define m_interrupts_ram_start          0x20000000
37 #define m_interrupts_ram_size          __ram_vector_table_size__
38
39 #define m_data_start                    0x200000c0
40 #define m_data_end                      0x20003fff
41 #define m_data_size                     m_data_end - m_data_start + 1
42
43 /* Sizes */
44 #if (defined(__stack_size__))
45     #define Stack_Size                    __stack_size__
46 #else
47     #define Stack_Size                    0x0400
48 #endif
49
50 #if (defined(__heap_size__))
51     #define Heap_Size                     __heap_size__
52 #else
53     #define Heap_Size                     0x0200
54 #endif

```

图 17: 链接文件修改

其中栈顶地址就是 0x20004000, 栈的初始化工作是在 \_\_main 函数里面完成, 这部分是 keil 自己的库完成的, 然后再跳到用户的 main 函数中

```

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

图 18: \_\_main

下面这段汇编代码是将 stack 的 end address 作为栈顶地址

```

LDR    R0, =__VECTOR_TABLE
LDR    R1, [R0]
MSR    MSP, R1

```

图 19: 栈顶地址设置

## 4 \_\_main 去除

如果想不使用 keil 自带的 \_\_main, 那么需要修改工程的配置, 而且另外需要自己实现 data 段, bss 段和堆栈的初始化

### 4.1 linker 配置修改

在 Misc\_controls 里添加 -no\_startup 配置选项让 keil 不链接 \_\_main

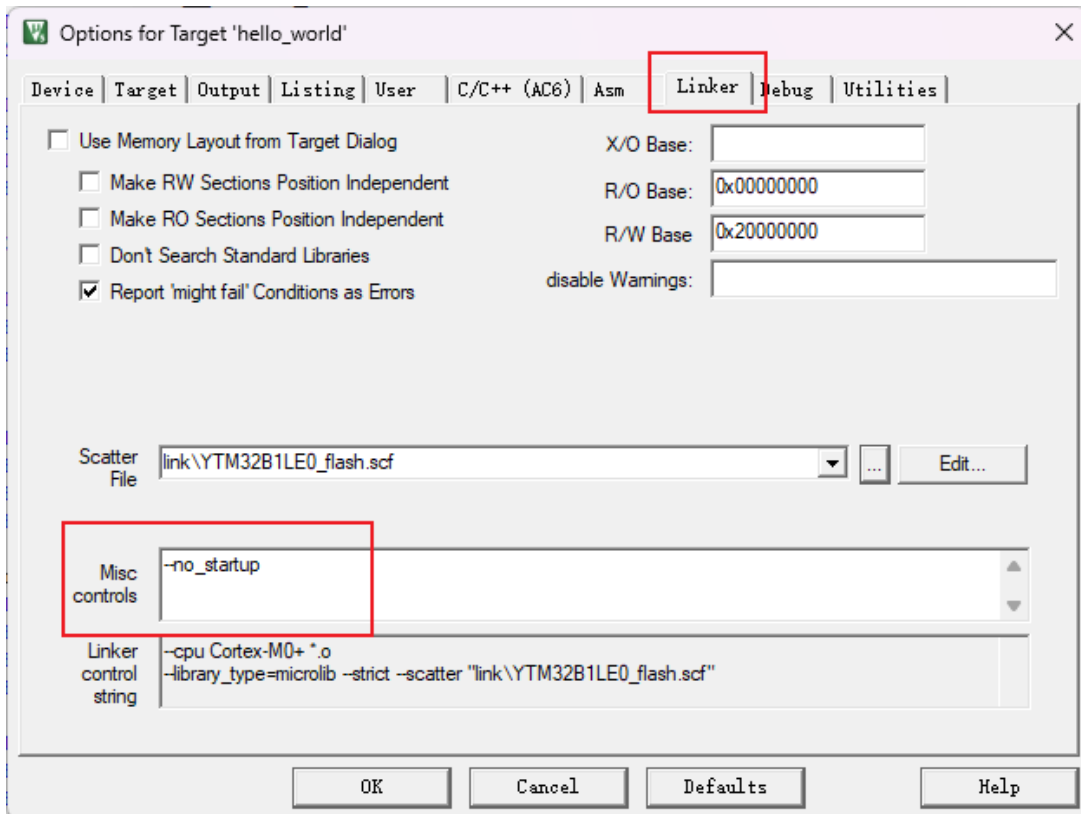


图 20: no\_startup 设置

### 4.2 汇编文件修改

在 startup 汇编代码中将 \_\_main 改成 main

```
IMPORT main
```

```
LDR R0, =main
```

```
BX R0
```

### 4.3 链接文件修改

把链接文件中 \*(InRoot\$\$Section) 去掉

```
55  -----
56  LR_m_text m_interrupts_start
57  { ; load region size_region
58  VECTOR_ROM m_interrupts_start m_interrupts_size {
59  YTM32B1LE0_startup_armc.o (RESET +First)
60  }
61
62  VECTOR_RAM m_interrupts_ram_start m_interrupts_ram_size {
63  .ANY (RESET)
64  }
65
66  ER_m_text m_text_start FIXED m_text_size { ; load address = execution address
67  *(InRoot$$Sections)
68  .ANY (+RO)
69  }
70
71  RW_m_data m_data_start m_data_size-Stack_Size-Heap_Size { ; RW data
72  .ANY (.code_ram)
73  .ANY (+RW,+ZI)
74  }
75
76  ARM_LIB_HEAP ((ImageLimit(RW_m_data) == m_data_start) ? m_data_start : +0) EMPTY Heap_Si
77  }
78
79  ARM_LIB_STACK m_data_start+m_data_size EMPTY -Stack_Size { ; Stack region growing down
```

图 21: 链接文件修改

## 版权与联系方式

云途半导体及其子公司保留随时对云途半导体产品或者本文档进行更改，更正，增强，修改和改进的权利，恕不另行通知。购买者在下单前应了解产品的最新相关信息。云途的产品根据云途的团队和订单确认时的销售条件进行销售。购买者对云途产品的选择权，挑选以及使用负全部责任，云途不对购买者的产品设计和应用协助承担任何责任。

云途在此未授予任何知识产权的任何明示或暗示许可。

转售与本文所述信息不同的云途产品将使云途对此类产品的任何保证失效。

云途半导体和云途半导体标志使云途半导体的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息将取代并替换之前在本文档的任何先前版本中提供的信息。

©2020 - 2023 云途半导体版权所有

**联系我们:**

**主页:** [www.ytmicro.com](http://www.ytmicro.com)