

引言

自举程序存储在STM32器件的内部自举ROM（系统存储器）中，由ST在生产期间进行编程。其主要任务是通过一种可用的串行外设（如USART、CAN、USB、I²C）将应用程序下载到内部Flash中。每种串行接口都定义了相应的通信协议，其中包含兼容的命令集和序列。本文档适用于表 1 中列出的产品，这些产品在文档中统称为STM32。该文档描述了使用STM32器件的自举程序时支持的外设以及需要考虑的硬件要求。

表1. 适用产品

类型	料号或产品系列
微控制器	STM32C0系列: STM32C011xx、STM32C031xx
	STM32F0系列: STM32F03xxx、STM32F04xxx、STM32F05xxx、STM32F07xxx、STM32F09xxx
	STM32F1系列
	STM32F2系列
	STM32F3系列: STM32F301xx、STM32F302xx、STM32F303xx、STM32F318xx、STM32F328xx、STM32F334xx、STM32F358xx、STM32F373xx、STM32F378xx、STM32F398xx
	STM32F4系列: STM32F401xx、STM32F405xx、STM32F407xx、STM32F410xx、STM32F411xx、STM32F412xx、STM32F413xx、STM32F415xx、STM32F417xx、STM32F423xx、STM32F427xx、STM32F429xx、STM32F437xx、STM32F439xx、STM32F446xx、STM32F469xx、STM32F479xx
	STM32F7系列: STM32F722xx、STM32F723xx、STM32F732xx、STM32F733xx、STM32F745xx、STM32F746xx、STM32F756xx、STM32F765xx、STM32F767xx、STM32F769xx、STM32F777xx、STM32F779xx
	STM32G0系列: STM32G030xx、STM32G031xx、STM32G041xx、STM32G07xxx、STM32G08xxx、STM32G0B0xx、STM32G0B1xx、STM32G0C1xx、STM32G050xx、STM32G051xx、STM32G061xx
	STM32G4系列: STM32G431xx、STM32G441xx、STM32G47xxx、STM32G48xxx、STM32G491xx、STM32G4A1xx
	STM32H7系列: STM32H72xxx、STM32H73xxx、STM32H74xxx、STM32H75xxx、STM32H7A3xx、STM32H7B3xx
	STM32L0系列
	STM32L1系列: STM32L100xx、STM32L151xx、STM32L152xx、STM32L162xx
	STM32L4系列: STM32L431xx、STM32L432xx、STM32L433xx、STM32L442xx、STM32L443xx、STM32L451xx、STM32L452xx、STM32L462xx、STM32L471xx、STM32L475xx、STM32L476xx、STM32L486xx、STM32L496xx、STM32L4A6xx、STM32L4R5xx、STM32L4R7xx、STM32L4R9xx、STM32L4S5xx、STM32L4S7xx、STM32L4S9xx、STM32L412xx、STM32L422xx、STM32L4P5xx、STM32L4Q5xx、STM32L431xx、STM32L432xx、STM32L433xx、STM32L442xx、STM32L443xx、STM32L451xx、STM32L452xx、STM32L462xx、STM32L471xx、STM32L475xx、STM32L476xx、STM32L486xx、STM32L496xx、STM32L4A6xx、STM32L4R5xx、STM32L4R7xx、STM32L4R9xx、STM32L4S5xx、STM32L4S7xx、STM32L4S9xx、STM32L412xx、STM32L422xx、STM32L4P5xx、STM32L4Q5xx
	STM32L5系列: STM32L552xx、STM32L562xx
	STM32WB系列: STM32WB10xx、STM32WB15xx、STM32WB30xx、STM32WB35xx、STM32WB50xx、STM32WB55xx
	STM32WL系列: STM32WLE5xx、STM32WL55xx
	STM32U5系列: STM32U575xx、STM32U585xx

目录

1	概述	22
2	相关文档	22
3	词汇表	23
4	通用自举程序描述	27
4.1	自举程序激活	27
4.2	自举程序标识	30
4.3	硬件连接要求	38
4.4	自举程序存储器管理	40
4.5	自举程序UART波特率检测	42
4.6	编程限制	43
4.7	ExitSecureMemory特性	44
5	STM32C011xx器件自举程序	47
5.1	自举程序配置	47
5.2	自举程序选择	48
5.3	自举程序版本	48
6	STM32C031xx器件自举程序	49
6.1	自举程序配置	49
6.2	自举程序选择	50
6.3	自举程序版本	50
7	STM32F03xx4/6器件自举程序	51
7.1	自举程序配置	51
7.2	自举程序选择	52
7.3	自举程序版本	52
8	STM32F030xC器件自举程序	53
8.1	自举程序配置	53
8.2	自举程序选择	55

8.3	自举程序版本	55
9	STM32F05xxx和STM32F030x8器件自举程序	56
9.1	自举程序配置	56
9.2	自举程序选择	57
9.3	自举程序版本	58
10	STM32F04xxx器件自举程序	59
10.1	自举程序配置	59
10.2	自举程序选择	61
10.3	自举程序版本	62
11	STM32F070x6器件自举程序	63
11.1	自举程序配置	63
11.2	自举程序选择	65
11.3	自举程序版本	66
12	STM32F070xB器件自举程序	67
12.1	自举程序配置	67
12.2	自举程序选择	69
12.3	自举程序版本	70
13	STM32F071xx/072xx器件自举程序	71
13.1	自举程序配置	71
13.2	自举程序选择	73
13.3	自举程序版本	74
14	STM32F09xxx器件自举程序	75
14.1	自举程序配置	75
14.2	自举程序选择	77
14.3	自举程序版本	77
15	STM32F10xxx器件自举程序	78
15.1	自举程序配置	78
15.2	自举程序选择	79

15.3	自举程序版本	79
16	STM32F105xx/107xx器件自举程序	81
16.1	自举程序配置	81
16.2	自举程序选择	83
16.3	自举程序版本	84
16.3.1	如何识别STM32F105xx/107xx自举程序版本	84
16.3.2	当日期代码小于937时，STM32F105xx/STM32F107xx器件的自举程序不可用	85
16.3.3	USART自举程序Get-Version命令返回0x20	而不是0x22 86
16.3.4	插入USB线缆时，PA9功耗过大	(在自举程序V2.0中) 86
17	STM32F10xxx超大容量器件自举程序	87
17.1	自举程序配置	87
17.2	自举程序选择	88
17.3	自举程序版本	88
18	STM32F2xxxx器件自举程序	90
18.1	自举程序V2.x	90
18.1.1	自举程序配置	90
18.1.2	自举程序选择	91
18.1.3	自举程序版本	92
18.2	自举程序V3.x	93
18.2.1	自举程序配置	93
18.2.2	自举程序选择	95
18.2.3	自举程序版本	96
19	STM32F301xx/302x4 (6/8) 器件自举程序	97
19.1	自举程序配置	97
19.2	自举程序选择	99
19.3	自举程序版本	99
20	STM32F302xB (C) /303xB (C) 器件自举程序	100
20.1	自举程序配置	100
20.2	自举程序选择	102
20.3	自举程序版本	102

21	STM32F302xD (E) /303xD (E) 器件自举程序	103
21.1	自举程序配置	103
21.2	自举程序选择	104
21.3	自举程序版本	105
22	STM32F303x4 (6/8) /334xx/328xx器件自举程序	106
22.1	自举程序配置	106
22.2	自举程序选择	107
22.3	自举程序版本	107
23	STM32F318xx器件自举程序	108
23.1	自举程序配置	108
23.2	自举程序选择	109
23.3	自举程序版本	110
24	STM32F358xx器件自举程序	111
24.1	自举程序配置	111
24.2	自举程序选择	112
24.3	自举程序版本	113
25	STM32F373xx器件自举程序	114
25.1	自举程序配置	114
25.2	自举程序选择	116
25.3	自举程序版本	116
26	STM32F378xx器件自举程序	117
26.1	自举程序配置	117
26.2	自举程序选择	118
26.3	自举程序版本	119
27	STM32F398xx器件自举程序	120
27.1	自举程序配置	120
27.2	自举程序选择	121
27.3	自举程序版本	122

28	STM32F40xxx/41xxx器件自举程序	123
28.1	自举程序V3.x	123
28.1.1	自举程序配置	123
28.1.2	自举程序选择	125
28.1.3	自举程序版本	126
28.2	自举程序V9.x	127
28.2.1	自举程序配置	127
28.2.2	自举程序选择	130
28.2.3	自举程序版本	131
29	STM32F401xB (C) 器件自举程序	132
29.1	自举程序配置	132
29.2	自举程序选择	135
29.3	自举程序版本	136
30	STM32F401xD (E) 器件自举程序	137
30.1	自举程序配置	137
30.2	自举程序选择	140
30.3	自举程序版本	141
31	STM32F410xx器件自举程序	142
31.1	自举程序配置	142
31.2	自举程序选择	145
31.3	自举程序版本	146
32	STM32F411xx器件自举程序	147
32.1	自举程序配置	147
32.2	自举程序选择	150
32.3	自举程序版本	151
33	STM32F412xx器件自举程序	152
33.1	自举程序配置	152
33.2	自举程序选择	156
33.3	自举程序版本	157
34	STM32F413xx/423xx器件自举程序	158

34.1	自举程序配置	158
34.2	自举程序选择	162
34.3	自举程序版本	163
35	STM32F42xxx/43xxx器件自举程序	164
35.1	自举程序V7.x	164
35.1.1	自举程序配置	164
35.1.2	自举程序选择	166
35.1.3	自举程序版本	168
35.2	自举程序V9.x	169
35.2.1	自举程序配置	169
35.2.2	自举程序选择	172
35.2.3	自举程序版本	174
36	STM32F446xx器件自举程序	175
36.1	自举程序配置	175
36.2	自举程序选择	179
36.3	自举程序版本	180
37	STM32F469xx/479xx器件自举程序	181
37.1	自举程序配置	181
37.2	自举程序选择	185
37.3	自举程序版本	187
38	STM32F72xxx/73xxx器件自举程序	188
38.1	自举程序配置	188
38.2	自举程序选择	192
38.3	自举程序版本	193
39	STM32F74xxx/75xxx器件自举程序	194
39.1	自举程序V7.x	194
39.1.1	自举程序配置	194
39.1.2	自举程序选择	197
39.1.3	自举程序版本	198
39.2	自举程序V9.x	199
39.2.1	自举程序配置	199

	39.2.2	自举程序选择	203
	39.2.3	自举程序版本	204
40		STM32F76xxx/77xxx器件自举程序	205
	40.1	自举程序配置	205
	40.2	自举程序选择	209
	40.3	自举程序版本	211
41		STM32G03xxx/STM32G04xxx器件自举程序	212
	41.1	自举程序配置	212
	41.2	自举程序选择	214
	41.3	自举程序版本	215
42		STM32G07xxx/08xxx器件自举程序	216
	42.1	自举程序配置	216
	42.2	自举程序选择	218
	42.3	自举程序版本	219
43		STM32G0B0xx器件自举程序	220
	43.1	自举程序配置	220
	43.2	自举程序选择	223
	43.3	自举程序版本	224
44		STM32G0B1xx/0C1xx器件自举程序	225
	44.1	自举程序配置	225
	44.2	自举程序选择	228
	44.3	自举程序版本	229
45		STM32G05xxx/061xx器件自举程序	230
	45.1	自举程序配置	230
	45.2	自举程序选择	232
	45.3	自举程序版本	233
46		STM32G431xx/441xx器件自举程序	234
	46.1	自举程序配置	234

46.2	自举程序选择	237
46.3	自举程序版本	238
47	STM32G47xxx/48xxx器件自举程序	239
47.1	自举程序配置	239
47.2	自举程序选择	242
47.3	自举程序版本	244
48	STM32G491xx/4A1xx器件自举程序	245
48.1	自举程序配置	245
48.2	自举程序选择	248
48.3	自举程序版本	249
49	STM32H72xxx/73xxx器件自举程序	250
49.1	自举程序配置	250
49.2	自举程序选择	254
49.3	自举程序版本	255
50	STM32H74xxx/75xxx器件自举程序	256
50.1	自举程序配置	256
50.2	自举程序选择	260
50.3	自举程序版本	261
51	STM32H7A3xx/B3xx器件自举程序	262
51.1	自举程序配置	262
51.2	自举程序选择	266
51.3	自举程序版本	267
52	STM32L01xxx/02xxx器件自举程序	268
52.1	自举程序配置	268
52.2	自举程序选择	270
52.3	自举程序版本	271
53	STM32L031xx/041xx器件自举程序	272
53.1	自举程序配置	272

53.2	自举程序选择	273
53.3	自举程序版本	273
54	STM32L05xxx/06xxx器件自举程序	274
54.1	自举程序配置	274
54.2	自举程序选择	276
54.3	自举程序版本	276
55	STM32L07xxx/08xxx器件自举程序	277
55.1	自举程序V4.x	277
55.1.1	自举程序配置	277
55.1.2	自举程序选择	279
55.1.3	自举程序版本	280
55.2	自举程序V11.x	281
55.2.1	自举程序配置	281
55.2.2	自举程序选择	283
55.2.3	自举程序版本	285
56	STM32L1xxx6 (8/B) A器件自举程序	286
56.1	自举程序配置	286
56.2	自举程序选择	287
56.3	自举程序版本	287
57	STM32L1xxx6 (8/B) 器件自举程序	288
57.1	自举程序配置	288
57.2	自举程序选择	289
57.3	自举程序版本	289
58	STM32L1xxxC器件自举程序	290
58.1	自举程序配置	290
58.2	自举程序选择	292
58.3	自举程序版本	293
59	STM32L1xxxD器件自举程序	294
59.1	自举程序配置	294
59.2	自举程序选择	296

59.3	自举程序版本	297
60	STM32L1xxxE器件自举程序	298
60.1	自举程序配置	298
60.2	自举程序选择	300
60.3	自举程序版本	301
61	STM32L412xx/422xx器件自举程序	302
61.1	自举程序配置	302
61.2	自举程序选择	305
61.3	自举程序版本	307
62	STM32L43xxx/44xxx器件自举程序	308
62.1	自举程序配置	308
62.2	自举程序选择	311
62.3	自举程序版本	313
63	STM32L45xxx/46xxx器件自举程序	314
63.1	自举程序配置	314
63.2	自举程序选择	318
63.3	自举程序版本	320
64	STM32L47xxx/48xxx器件自举程序	321
64.1	自举程序V10.x	321
64.1.1	自举程序配置	321
64.1.2	自举程序选择	323
64.1.3	自举程序版本	325
64.2	自举程序V9.x	326
64.2.1	自举程序配置	326
64.2.2	自举程序选择	329
64.2.3	自举程序版本	331
65	STM32L496xx/4A6xx器件自举程序	332
65.1	自举程序配置	332
65.2	自举程序选择	336
65.3	自举程序版本	338

66	STM32L4P5xx/4Q5xx器件自举程序	339
66.1	自举程序配置	339
66.2	自举程序选择	343
66.3	自举程序版本	345
67	STM32L4Rxxx/4Sxxx器件自举程序	346
67.1	自举程序配置	346
67.2	自举程序选择	350
67.3	自举程序版本	352
68	STM32L552xx/STM32L562xx器件自举程序	353
68.1	自举程序配置	353
68.2	自举程序选择	357
68.3	自举程序版本	358
69	STM32WB10xx/15xx器件自举程序	359
69.1	自举程序配置	359
69.2	自举程序选择	361
69.3	自举程序版本	362
70	STM32WB30xx/35xx/50xx/55xx器件自举程序	363
70.1	自举程序配置	363
70.2	自举程序选择	365
70.3	自举程序版本	366
71	STM32WLE5xx/55xx器件自举程序	367
71.1	自举程序配置	367
71.2	自举程序选择	369
71.3	自举程序版本	369
72	STM32U575xx/85xx器件自举程序	370
72.1	自举程序配置	370
72.2	自举程序选择	374
72.3	自举程序版本	375

73	器件相关的自举程序参数	376
74	自举程序时序	381
	74.1 自举程序启动时序	381
	74.2 USART连接时序	384
	74.3 USB连接时序	387
	74.4 I2C连接时序	389
	74.5 SPI连接时序	391
附录A	使用“ExitSecureMemory”函数的函数示例.....	392
75	版本历史	394

表格索引

表1.	适用产品	2
表2.	自举程序激活模式	27
表3.	嵌入式自举程序	31
表4.	利用自举程序进行STM32 F2、F4和F7电压范围配置	41
表5.	Write、Read、Erase和Go命令所支持的存储区	41
表6.	自举程序USART检测的软件抖动计算	42
表7.	STM32产品的Flash存储器对齐限制	43
表8.	ExitSecureMemory入口地址	46
表9.	系统存储器自举模式下STM32C011xx器件的配置	47
表10.	STM32C011xx自举程序版本	48
表11.	系统存储器自举模式下STM32C031xx器件的配置	49
表12.	STM32C031xx自举程序版本	50
表13.	系统存储器自举模式下STM32F03xx4/6器件的配置	51
表14.	STM32F03xx4/6自举程序版本	52
表15.	系统存储器自举模式下STM32F030xC器件的配置	53
表16.	STM32F030xC自举程序版本	55
表17.	系统存储器自举模式下STM32F05xxx和STM32F030x8器件的配置	56
表18.	STM32F05xxx和STM32F030x8器件自举程序版本	58
表19.	系统存储器自举模式下STM32F04xxx器件的配置	59
表20.	STM32F04xxx自举程序版本	62
表21.	系统存储器自举模式下STM32F070x6器件的配置	63
表22.	STM32F070x6自举程序版本	66
表23.	系统存储器自举模式下STM32F070xB器件的配置	67
表24.	STM32F070xB自举程序版本	70
表25.	系统存储器自举模式下STM32F071xx/072xx器件的配置	71
表26.	STM32F071xx/072xx自举程序版本	74
表27.	系统存储器自举模式下STM32F09xxx器件的配置	75
表28.	STM32F09xxx自举程序版本	77
表29.	系统存储器自举模式下STM32F10xxx器件的配置	78
表30.	STM32F10xxx自举程序版本	79
表31.	系统存储器自举模式下STM32F105xx/107xx器件的配置	81
表32.	STM32F105xx/107xx自举程序版本	84
表33.	系统存储器自举模式下STM32F10xxx超大容量器件的配置	87
表34.	STM32F10xxx超大容量自举程序版本	88
表35.	系统存储器自举模式下STM32F2xxxx器件的配置	90
表36.	STM32F2xxxx自举程序V2.x版本	92
表37.	系统存储器自举模式下STM32F2xxxx器件的配置	93
表38.	STM32F2xxxx自举程序V3.x版本	96
表39.	系统存储器自举模式下STM32F301xx/302x4 (6/8) 器件的配置	97
表40.	STM32F301xx/302x4 (6/8) 自举程序版本	99
表41.	系统存储器自举模式下STM32F302xB (C) /303xB (C) 器件的配置	100
表42.	STM32F302xB (C) /303xB (C) 自举程序版本	102
表43.	系统存储器自举模式下STM32F302xD (E) /303xD (E) 器件的配置	103
表44.	STM32F302xD (E) /303xD (E) 自举程序版本	105
表45.	系统存储器自举模式下STM32F303x4 (6/8) /334xx/328xx器件的配置	106
表46.	STM32F303x4 (6/8) /334xx/328xx自举程序版本	107
表47.	系统存储器自举模式下STM32F318xx器件的配置	108
表48.	STM32F318xx自举程序版本	110

表49.	系统存储器自举模式下STM32F358xx器件的配置.....	111
表50.	STM32F358xx自举程序版本.....	113
表51.	系统存储器自举模式下STM32F373xx器件的配置.....	114
表52.	STM32F373xx自举程序版本.....	116
表53.	系统存储器自举模式下STM32F378xx器件的配置.....	117
表54.	STM32F378xx自举程序版本.....	119
表55.	系统存储器自举模式下STM32F398xx器件的配置.....	120
表56.	STM32F398xx自举程序版本.....	122
表57.	系统存储器自举模式下STM32F40xxx/41xxx器件的配置.....	123
表58.	STM32F40xxx/41xxx自举程序V3.x版本.....	126
表59.	系统存储器自举模式下STM32F40xxx/41xxx器件的配置.....	127
表60.	STM32F40xxx/41xxx自举程序V9.x版本.....	131
表61.	系统存储器自举模式下STM32F401xB (C) 器件的配置.....	132
表62.	STM32F401xB (C) 自举程序版本.....	136
表63.	系统存储器自举模式下STM32F401xD (E) 器件的配置.....	137
表64.	STM32F401xD (E) 自举程序版本.....	141
表65.	系统存储器自举模式下STM32F410xx器件的配置.....	142
表66.	STM32F410xx自举程序V11.x版本.....	146
表67.	系统存储器自举模式下STM32F411xx器件的配置.....	147
表68.	STM32F411xx自举程序版本.....	151
表69.	系统存储器自举模式下STM32F412xx器件的配置.....	152
表70.	STM32F412xx自举程序V9.x版本.....	157
表71.	系统存储器自举模式下STM32F413xx/423xx器件的配置.....	158
表72.	STM32F413xx/423xx自举程序V9.x版本.....	163
表73.	系统存储器自举模式下STM32F42xxx/43xxx器件的配置.....	164
表74.	STM32F42xxx/43xxx自举程序V7.x版本.....	168
表75.	系统存储器自举模式下STM32F42xxx/43xxx器件的配置.....	169
表76.	STM32F42xxx/43xxx自举程序V9.x版本.....	174
表77.	系统存储器自举模式下STM32F446xx器件的配置.....	175
表78.	STM32F446xx自举程序V9.x版本.....	180
表79.	系统存储器自举模式下STM32F469xx/479xx器件的配置.....	181
表80.	STM32F469xx/479xx自举程序V9.x版本.....	187
表81.	系统存储器自举模式下STM32F72xxx/73xxx器件的配置.....	188
表82.	STM32F72xxx/73xxx自举程序V9.x版本.....	193
表83.	系统存储器自举模式下STM32F74xxx/75xxx器件的配置.....	194
表84.	STM32F74xxx/75xxx自举程序V7.x版本.....	198
表85.	系统存储器自举模式下STM32F74xxx/75xxx器件的配置.....	199
表86.	STM32F74xxx/75xxx自举程序V9.x版本.....	204
表87.	系统存储器自举模式下STM32F76xxx/77xxx器件的配置.....	205
表88.	STM32F76xxx/77xxx自举程序V9.x版本.....	211
表89.	系统存储器自举模式下STM32G03xxx/G04xxx器件的配置.....	212
表90.	STM32G03xxx/04xxx自举程序版本.....	215
表91.	系统存储器自举模式下STM32G07xxx/8xxx器件的配置.....	216
表92.	STM32G07xxx/08xxx自举程序版本.....	219
表93.	系统存储器自举模式下STM32G0B0xx器件的配置.....	220
表94.	STM32G0B0xx自举程序版本.....	224
表95.	系统存储器自举模式下STM32G0B1xx/0C1xx器件的配置.....	225
表96.	STM32G0B1xx/0C1xx自举程序版本.....	229
表97.	系统存储器自举模式下STM32G05xxx/061xx器件的配置.....	230
表98.	STM32G05xxx/061xx自举程序版本.....	233
表99.	系统存储器自举模式下STM32G431xx/441xx器件的配置.....	234
表100.	STM32G431xx/441xx自举程序版本.....	238

表101.	系统存储器自举模式下STM32G47xxx/48xxx器件的配置	239
表102.	STM32G47xxx/48xxx自举程序版本	244
表103.	系统存储器自举模式下STM32G491xx/4A1xx器件的配置	245
表104.	STM32G491xx/4A1xx自举程序版本	249
表105.	系统存储器自举模式下STM32H72xxx/73xxx器件的配置	250
表106.	STM32H72xxx/73xxx自举程序版本	255
表107.	系统存储器自举模式下STM32H74xxx/75xxx器件的配置	256
表108.	STM32H74xxx/75xxx自举程序版本	261
表109.	系统存储器自举模式下STM32H7A3xx/7B3xx器件的配置	262
表110.	STM32H7A3xx/7B3xx自举程序版本	267
表111.	系统存储器自举模式下STM32L01xxx/02xxx器件的配置	268
表112.	STM32L01xxx/02xxx自举程序版本	271
表113.	系统存储器自举模式下STM32L031xx/041xx器件的配置	272
表114.	STM32L031xx/041xx自举程序版本	273
表115.	系统存储器自举模式下STM32L05xxx/06xxx器件的配置	274
表116.	STM32L05xxx/06xxx自举程序版本	276
表117.	系统存储器自举模式下STM32L07xxx/08xxx器件的配置	277
表118.	STM32L07xxx/08xxx自举程序版本	280
表119.	系统存储器自举模式下STM32L07xxx/08xxx器件的配置	281
表120.	STM32L07xxx/08xxx自举程序V11.x版本	285
表121.	系统存储器自举模式下STM32L1xxx6 (8/B) A器件的配置	286
表122.	STM32L1xxx6 (8/B) A自举程序版本	287
表123.	系统存储器自举模式下STM32L1xxx6 (8/B) 器件的配置	288
表124.	STM32L1xxx6 (8/B) 自举程序版本	289
表125.	系统存储器自举模式下STM32L1xxxC器件的配置	290
表126.	STM32L1xxxC自举程序版本	293
表127.	系统存储器自举模式下STM32L1xxxD器件的配置	294
表128.	STM32L1xxxD自举程序版本	297
表129.	系统存储器自举模式下STM32L1xxxE器件的配置	298
表130.	STM32L1xxxE自举程序版本	301
表131.	系统存储器自举模式下STM32L412xx/422xx器件的配置	302
表132.	STM32L412xx/422xx自举程序版本	307
表133.	系统存储器自举模式下STM32L43xxx/44xxx器件的配置	308
表134.	STM32L43xxx/44xxx自举程序版本	313
表135.	系统存储器自举模式下STM32L45xxx/46xxx器件的配置	314
表136.	STM32L45xxx/46xxx自举程序版本	320
表137.	系统存储器自举模式下STM32L47xxx/48xxx器件的配置	321
表138.	STM32L47xxx/48xxx自举程序V10.x版本	325
表139.	系统存储器自举模式下STM32L47xxx/48xxx器件的配置	326
表140.	STM32L47xxx/48xxx自举程序V9.x版本	331
表141.	系统存储器自举模式下STM32L496xx/4A6xx器件的配置	332
表142.	STM32L496xx/4A6xx自举程序版本	338
表143.	系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置	339
表144.	STM32L4P5xx/4Q5xx自举程序版本	345
表145.	系统存储器自举模式下STM32L4Rxxx/4Sxxx器件的配置	346
表146.	STM32L4Rxx/4Sxx自举程序版本	352
表147.	系统存储器自举模式下STM32L552xx/562xx器件的配置	353
表148.	STM32L552cc/562xx特殊命令	356
表149.	STM32L552xx/562xx自举程序版本	358
表150.	系统存储器自举模式下STM32WB10xx/15xx的配置	359
表151.	STM32WB10xx/15xx自举程序版本	362
表152.	系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx的配置	363

表153.	STM32WB30xx/35xx/50xx/55xx自举程序版本	366
表154.	系统存储器自举模式下STM32WLE5xx/55xx器件的配置	367
表155.	STM32WLE5xx/55xx自举程序版本	369
表156.	系统存储器自举模式下STM32U575xx/85xx的配置	370
表157.	STM32U575xx/585xx特殊命令	373
表158.	STM32U575xx/85xx自举程序版本	375
表159.	器件相关的自举程序参数	376
表160.	STM32器件的自举程序启动时序 (ms)	381
表161.	STM32器件的USART自举程序最小时序 (ms)	384
表162.	STM32器件的USB自举程序最小时序 (ms)	387
表163.	STM32器件的I2C自举程序最小时序 (ms)	389
表164.	STM32器件的SPI自举程序最小时序 (ms)	391
表165.	文档版本历史	394
表166.	中文文档版本历史	408

图片目录

图1.	USART连接	38
图2.	USB连接	38
图3.	I2C连接	39
图4.	SPI连接	39
图5.	CAN连接	39
图6.	ExitSecureMemory函数的使用	45
图7.	从自举程序访问安全存储区	46
图8.	STM32C011xx器件的自举程序V5.x选择	48
图9.	STM32C031xx器件的自举程序V5.x选择	50
图10.	STM32F03xx4/6器件的自举程序选择	52
图11.	STM32F030xC器件的自举程序选择	55
图12.	STM32F05xxx和STM32F030x8器件的自举程序选择	57
图13.	STM32F04xxx器件的自举程序选择	61
图14.	STM32F070x6器件的自举程序选择	65
图15.	STM32F070xB器件的自举程序选择	69
图16.	STM32F071xx/072xx器件的自举程序选择	73
图17.	STM32F09xxx器件的自举程序选择	77
图18.	STM32F10xxx器件的自举程序选择	79
图19.	STM32F105xx/107xx器件的自举程序选择	83
图20.	STM32F10xxx超大容量器件的自举程序选择	88
图21.	STM32F2xxxx器件的自举程序V2.x选择	91
图22.	STM32F2xxxx器件的自举程序V3.x选择	95
图23.	STM32F301xx/302x4 (6/8)的自举程序选择	99
图24.	STM32F302xB (C) /303xB (C) 器件的自举程序选择	102
图25.	STM32F302xD (E) /303xD (E) 器件的自举程序选择	104
图26.	STM32F303x4 (6/8) /334xx/328xx器件的自举程序选择	107
图27.	STM32F318xx器件的自举程序选择	109
图28.	STM32F358xx器件的自举程序选择	112
图29.	STM32F373xx器件的自举程序选择	116
图30.	STM32F378xx器件的自举程序选择	118
图31.	STM32F398xx器件的自举程序选择	121
图32.	STM32F40xxx/41xxx器件的自举程序V3.x选择	125
图33.	STM32F40xxx/41xxx器件的自举程序V9.x选择	130
图34.	STM32F401xB (C) 器件的自举程序选择	135
图35.	STM32F401xD (E) 器件的自举程序选择	140
图36.	STM32F410xx器件的自举程序V11.x选择	145
图37.	STM32F411xx器件的自举程序选择	150
图38.	STM32F412xx器件的自举程序V9.x选择	156
图39.	STM32F413xx/423xx器件的自举程序V9.x选择	162
图40.	STM32F42xxx/43xxx自举程序V7.x的双存储区自举实现	166
图41.	STM32F42xxx/43xxx器件的自举程序V7.x选择	167
图42.	STM32F42xxx/43xxx自举程序V9.x的双存储区自举实现	172
图43.	STM32F42xxx/43xxx器件的自举程序V9.x选择	173
图44.	STM32F446xx器件的自举程序V9.x选择	179
图45.	STM32F469xx/479xx自举程序V9.x的双存储区自举实现	185
图46.	STM32F469xx/479xx器件的自举程序V9.x选择	186
图47.	STM32F72xxx/73xxx器件的自举程序V9.x选择	192
图48.	STM32F74xxx/75xxx器件的自举程序V7.x选择	197

图49.	STM32F74xxx/75xxx器件的自举程序V9.x选择	203
图50.	STM32F76xxx/77xxx自举程序V9.x的双存储区自举实现	209
图51.	STM32F76xxx/77xxx器件的自举程序V9.x选择	210
图52.	STM32G03xxx/G04xxx的自举程序V5.x选择	214
图53.	STM32G07xxx/G08xxx的自举程序V11.0选择	218
图54.	STM32G0B0xx器件的自举程序选择	223
图55.	STM32G0B1xx/0C1xx的自举程序选择	228
图56.	STM32G05xxx/061xx的自举程序选择	232
图57.	STM32G431xx/441xx的自举程序选择	237
图58.	STM32G47xxx/48xxx的自举程序选择	242
图59.	STM32G47xxx/48xxx自举程序V13.x的双存储区自举实现	243
图60.	STM32G491xx/4A1xx的自举程序选择	248
图61.	STM32H72xxx/73xxx器件的自举程序V9.0选择	254
图62.	STM32H74xxx/75xxx器件的自举程序V9.x选择	260
图63.	STM32H7A3xx/7B3xx的自举程序V9.x选择	266
图64.	STM32L01xxx/02xxx器件的自举程序选择	270
图65.	STM32L031xx/041xx器件的自举程序选择	273
图66.	STM32L05xxx/06xxx器件的自举程序选择	276
图67.	STM32L07xxx/08xxx自举程序V4.x的双存储区自举实现	279
图68.	STM32L07xxx/08xxx器件的自举程序V4.x选择	280
图69.	STM32L07xxx/08xxx自举程序V11.x的双存储区自举实现	283
图70.	STM32L07xxx/08xxx器件的自举程序V11.x选择	284
图71.	STM32L1xxx6 (8/B) A器件的自举程序选择	287
图72.	STM32L1xxx6 (8/B) 器件的自举程序选择	289
图73.	STM32L1xxxC器件的自举程序选择	292
图74.	STM32L1xxxD器件的自举程序选择	296
图75.	STM32L1xxxE器件的自举程序选择	300
图76.	STM32L412xx/422xx自举程序V9.x的双存储区自举实现	305
图77.	STM32L412xx/422xx的自举程序V13.x选择	306
图78.	STM32L3x2xx/44xxx自举程序V9.x的双存储区自举实现	311
图79.	STM32L43xxx/44xxx器件的自举程序V9.x选择	312
图80.	STM32L45xxx/46xxx自举程序V9.x的双存储区自举实现	318
图81.	STM32L45xxx/46xxx器件的自举程序V9.x选择	319
图82.	STM32L47xxx/48xxx自举程序V10.x的双存储区自举实现	323
图83.	STM32L47xxx/48xxx器件的自举程序V10.x选择	324
图84.	STM32L47xxx/48xxx自举程序V9.x的双存储区自举实现	329
图85.	STM32L47xxx/48xxx器件的自举程序V9.x选择	330
图86.	STM32L496xx/4A6xx自举程序V9.x的双存储区自举实现	336
图87.	STM32L496xx/4A6xx器件的自举程序V9.x选择	337
图88.	STM32L4P5xx/4Q5xx自举程序V9.x的双存储区自举实现	343
图89.	STM32L4P5xx/4Q5xx的自举程序V9.x选择	344
图90.	STM32L4Rxxx/STM32L4Sxxx自举程序V9.x的双存储区自举实现	350
图91.	STM32L4Rxx/4Sxx器件的自举程序V9.x选择	351
图92.	STM32L552xx/562xx的自举程序V9.x选择	357
图93.	STM32WB10xx/15xx的自举程序V11.x选择	361
图94.	STM32WB30xx/35xx/50xx/55xx的自举程序V13.0选择	365
图95.	STM32WLE5xx/55xx的自举程序V12.x选择	369
图96.	STM32U575xx/85xx的自举程序V9.x选择	374
图97.	自举程序启动时序描述	381
图98.	USART连接时序描述	384
图99.	USB连接时序描述	387
图100.	I2C连接时序描述	389

图101. SPI连接时序描述 391

1 概述

本文档适用于基于Arm^{®(a)}的器件。

2 相关文档

有关所支持的每个产品（如表 1 中所列）的信息，请参见www.st.com提供的以下文档：

- 数据手册或产品简介
- 参考手册
- 应用笔记
 - AN3154: *STM32自举程序中使用的CAN协议*
 - AN3155: *STM32自举程序中使用的USART协议*
 - AN3156: *STM32自举程序中使用的USB DFU协议*
 - AN4221: *STM32自举程序中使用的I2C协议*
 - AN4286: *STM32自举程序中使用的SPI协议*
 - AN5405: *STM32自举程序中使用的FDCAN协议*

arm

a. Arm是Arm Limited（或其子公司）在美国和/或其他地区的注册商标。

3 词汇表

C0系列:

STM32C011xx指代STM32C011xx器件。

STM32C031xx指代STM32C031xx器件。

F0系列:

STM32F03xxx指代STM32F030x4、STM32F030x6、STM32F038x6、STM32F030xC、STM32F031x4和STM32F031x6器件。

STM32F04xxx指代STM32F042x4和STM32F042x6器件。

STM32F05xxx和**STM32F030x8**器件指代STM32F051x4、STM32F051x6、STM32F051x8、STM32F058x8和STM32F030x8器件。

STM32F07xxx指代STM32F070x6、STM32F070xB、STM32F071xB STM32F072x8和STM32F072xB器件。

STM32F09xxx指代STM32F091xx和STM32F098xx器件。

F1系列:

STM32F10xxx指代小容量、中容量、大容量、小容量超值型、中容量超值型和大容量超值型器件:

小容量器件包括Flash容量介于16KB到32KB之间的STM32F101xx、STM32F102xx和STM32F103xx微控制器。

中容量器件包括Flash容量介于64 KB到128 KB之间的STM32F101xx、STM32F102xx和STM32F103xx微控制器。

大容量器件包括Flash容量介于256 KB到512 KB之间的STM32F101xx和STM32F103xx微控制器。

小容量超值型器件包括Flash容量介于16 KB到32 KB之间的STM32F100xx微控制器。

中容量超值型器件包括Flash容量介于64 KB到128 KB之间的STM32F100xx微控制器。

大容量超值型器件包括Flash容量介于256 KB到512 KB之间的STM32F100xx微控制器。

STM32F105xx/107xx指代STM32F105xx和STM32F107xx器件。

STM32F10xxx超大容量指代Flash容量介于768 KB到1 MB之间的STM32F101xx和STM32F103xx器件。

F2系列:

STM32F2xxxx指代STM32F215xx、STM32F205xx、STM32F207xx和SMT32F217xx器件。

F3系列:

STM32F301xx/302x4 (6/8) 指代STM32F301x4、STM32F301x6、STM32F301x8、STM32F302x4、STM32F302x6和STM32F302x8器件。

STM32F302xB (C) /303xB (C) 指代STM32F302xB、STM32F302xC、STM32F303xB和STM32F303xC器件。

STM32F302xD (E) /303xD (E) 指代STM32F302xD、STM32F302xE、STM32F303xD和STM32F303xE器件。

STM32F303x4 (6/8) /334xx/328xx指代STM32F303x4、STM32F303x6、STM32F303x8、STM32F334x4、STM32F334x6、STM32F334x8和STM32F328x8器件。

STM32F318xx指代STM32F318x8器件。

STM32F358xx指代STM32F358xC器件。

STM32F373xx指代STM32F373x8、STM32F373xB和STM32F373xC器件。

STM32F378xx指代STM32F378xC器件。

STM32F398xx指代STM32F398xE器件。

F4系列:

STM32F40xxx/41xxx指代STM32F405xx、STM32F407xx、STM32F415xx和STM32F417xx器件。

STM32F401xB (C) 指代STM32F401xB和STM32F401xC器件。

STM32F401xD (E) 指代STM32F401xD和STM32F401xE器件。

STM32F410xx指代STM32F410x8和STM32F410xB器件。

STM32F411xx指代STM32F411xD和STM32F411xE器件。

STM32F412xx指代STM32F412Cx、STM32F412Rx、STM32F412Vx和STM32F412Zx器件。

STM32F413xx/423xx指代STM32F413xG、STM32F413xH和STM32F423xH器件。

STM32F42xxx/43xxx指代STM32F427xx、STM32F429xx、STM32F437xx和STM32F439xx器件。

STM32F446xx指代STM32F446xE和STM32F446xC器件。

STM32F469xx/479xx指代STM32F469xE、STM32F469xG、STM32F469xI、STM32F479xG和STM32F479xI器件。

F7系列:

STM32F72xxx/73xxx指代STM32F723xx、STM32F732xx和STM32F733xx器件。

STM32F74xxx/75xxx指代STM32F745xx、STM32F746xx和STM32F756xx器件。

STM32F76xxx/77xxx指代STM32F765xx、STM32F767xx、STM32F769xx、STM32F777xx和STM32F779xx器件。

G0系列:

STM32G03xxx/04xxx指代STM32G03xxx和STM32G04xxx器件。

STM32G07xxx/08xxx指代STM32G07xxx和STM32G08xxx器件。

STM32G0B1xx/C1xx指代STM32GB1xx和STM32G0C1xxx器件。

STM32G0B0xx指代STM32G0B0xx器件。

STM32G05xxx/61xx指代STM32G050xx、STM32G051xx和STM32G061xx器件。

G4系列:

STM32G431xx指代STM32G431xx器件。

STM32G441xx指代STM32G441xx器件。

STM32G47xxx指代STM32G471xx、STM32G473xx和STM32G474xx器件。

STM32G48xxx指代STM32G483xx和STM32G484xx器件。

STM32G491xx指代STM32G491xx器件。

STM32G4A1xx指代STM32G4A1xx器件。

H7系列:

STM32H72xxx/73xxx指代STM32H72xxx和STM32H73xxx器件。

STM32H74xxx/75xxx指代STM32H74xxx和STM32H75xxx器件。

STM32H7A3xx/7B3xx指代STM32H7A3xx/和STM32H7B3xx器件。

L0系列:

STM32L01xxx/02xxx指代STM32L011xx和STM32L021xx器件。

STM32L031xx/041xx指代STM32L031xx和STM32L041xx器件。

STM32L05xxx/06xxx指代STM32L051xx、STM32L052xx、STM32L053xx、STM32L062xx和STM32L063xx超低功耗器件。

STM32L07xxx/08xxx指代STM32L071xx、STM32L072xx、STM32L073xx、STM32L081xx、STM32L082xx和STM32L083xx器件。

L1系列:

STM32L1xxx6 (8/B) 指代STM32L1xxV6T6、STM32L1xxV6H6、STM32L1xxR6T6、STM32L1xxR6H6、STM32L1xxC6T6、STM32L1xxC6H6、STM32L1xxV8T6、STM32L1xxV8H6、STM32L1xxR8T6、STM32L1xxR8H6、STM32L1xxC8T6、STM32L1xxC8H6、STM32L1xxVBT6、STM32L1xxVBH6、STM32L1xxRBT6、STM32L1xxRBH6、STM32L1xxCBT6和STM32L1xxCBH6超低功耗器件。

STM32L1xxx6(8/B)A指代STM32L1xxV6T6-A、STM32L1xxV6H6-A、STM32L1xxR6T6-A、STM32L1xxR6H6-A、STM32L1xxC6T6-A、STM32L1xxC6H6-A、STM32L1xxV8T6-A、STM32L1xxV8H6-A、STM32L1xxR8T6-A、STM32L1xxR8H6-A、STM32L1xxC8T6-A、STM32L1xxC8H6-A、STM32L1xxVBT6-A、STM32L1xxVBH6-A、STM32L1xxRBT6-A、STM32L1xxRBH6-A、STM32L1xxCBT6-A和STM32L1xxCBH6-A超低功耗器件。

STM32L1xxxC指代STM32L1xxVCT6、STM32L1xxVCH6、STM32L1xxRCT6、STM32L1xxUCY6、STM32L1xxCCT6和STM32L1xxCCU6超低功耗器件。

STM32L1xxxD指代STM32L1xxZDT6、STM32L1xxQDH6、STM32L1xxVDT6、STM32L1xxRDY6、STM32L1xxRDT6、STM32L1xxZCT6、STM32L1xxQCH6、STM32L1xxRCY6、STM32L1xxVCT6-A和STM32L1xxRCT6-A超低功耗器件。

STM32L1xxxE指代STM32L1xxZET6、STM32L1xxQEH6、STM32L1xxVET6、STM32L1xxVEY6和STM32L1xxRET6超低功耗器件。

L4系列:

STM32L412xx/422xx指代STM32L412xB、STM32L412x8和STM32L422xB器件。

STM32L43xxx/44xxx指代STM32L431xx、STM32L432xx、STM32L433xx以及STM32L442xx和STM32L443xx器件。

STM32L45xxx/46xxx指代STM32L451xx、STM32L452xx和STM32L462xx器件。

STM32L47xxx/48xxx指代STM32L471xx、STM32L475xx、STM32L476xx和STM32L486xx器件。

STM32L496xx/4A6xx指代STM32L496xE、STM32L496xG和STM32L4A6xG器件。

STM32L4Rxxx/4Sxxx指代STM32L4R5xx、STM32L4R7xx、STM32L4R9xx、STM32L4S5xx、STM32L4S7xx和STM32L4S9xx器件。

STM32L4P5xx/4Q5xx指代STM32L4P5xx/STM32L4Q5xx器件。

L5系列:

STM32L552xx指代STM32L552xx器件。

STM32L562xx指代STM32L562xx器件。

WB系列:

STM32WB10xx指代STM32WB10xx器件。

STM32WB15xx指代STM32WB15xx器件。

STM32WB30xx指代STM32WB30xx器件。

STM32WB35xx指代STM32WB35xx器件。

STM32WB50xx指代STM32WB50xx器件。

STM32WB55xx指代STM32WB55Cx、STM32WB55Rx和STM32WB55Vx器件。

WL系列:

STM32WLE5xx指代STM32WLE5xx器件。

STM32WL55xx指代STM32WL55xx器件。

U5系列:

STM32U575xx指代STM32U575xx器件。

STM32U585xx指代STM32U585xx器件。

注: **BL_USART_Loop**指USART自举程序执行循环。
BL_CAN_Loop指CAN自举程序执行循环。
BL_I2C_Loop指I2C自举程序执行循环。
BL_SPI_Loop指SPI自举程序执行循环。

4 通用自举程序描述

4.1 自举程序激活

自举程序通过应用表 2 所述模式之一激活。

如果Boot From Bank2选项激活（就支持此功能的产品而言），自举程序执行Dual Boot机制，如图“STM32xxx的双存储区自举实现”中所示，其中STM32xxx是相关的STM32产品（示例：图 40）；否则，执行自举程序选择方案，如图“STM32xxx的自举程序VY.x选择”所示，其中STM32xxx是相关的STM32产品（示例：图 21）。

当读出保护Level2激活时，任何情况下STM32都不在系统存储器上自举，自举程序不能执行（除非从Flash用户代码跳转到它，否则除Get、GetID和GetVersion之外的所有指令都无法访问）。

表2.自举程序激活模式

模式	激活条件
模式1	Boot0(pin) = 1和Boot1(pin) = 0
模式2	Boot0(pin) = 1和nBoot1(bit) = 1
模式3	Boot0(pin) = 1, Boot1(pin) = 0和BFB2(bit) = 1
	Boot0(pin) = 0, BFB2(bit) = 0且两个存储区均不包含有效代码
模式4	Boot0(pin) = 1, Boot1(pin) = 0和BFB2(bit) = 1
	Boot0(pin) = 0, BFB2(bit) = 0且两个存储区均不包含有效代码
	Boot0(pin) = 1, Boot1(pin) = 0和BFB2(bit) = 0两个存储区均不包含有效代码
模式5	Boot0(pin) = 1, Boot1(pin) = 0和BFB2(bit) = 0
	Boot0(pin) = 0, BFB2(bit) = 1且两个存储区均不包含有效代码
	Boot0(pin) = 1, Boot1(pin) = 0和BFB2 (bit) = 1
模式6	Boot0(pin) = 1, nBoot1(bit) = 1和nBoot0_SW(bit) = 1
	nBoot0 (bit) = 0, nBoot1(bit) = 1和nBoot0_SW(bit) = 0
	Boot0(pin) = 0, nBoot0_SW(bit) = 1且主Flash存储器为空
	nBoot0(bit) = 1, nBoot0_SW(bit)=0且主Flash存储器为空
模式7	Boot0(pin) = 1, nBoot1(bit) = 1和BFB2(bit) = 0
	Boot0(pin) = 0, BFB2(bit) = 1且两个存储区均不包含有效代码
	Boot0(pin) = 1, nBoot1(bit) = 1和BFB2(bit) = 1
模式8	Boot(pin) = 0和BOOT_ADD0(optionbyte) = 0x0040
	Boot(pin) = 1和BOOT_ADD1(optionbyte) = 0x0040

表2.自举程序激活模式（续）

模式	激活条件
模式9	nDBANK(bit) = 1, Boot(pin) = 0和BOOT_ADD0(optionbyte) = 0x0040
	nDBANK(bit) = 1, Boot(pin) = 1和BOOT_ADD1(optionbyte) = 0x0040
	nDBANK(bit) = 0, nDBOOT(bit) = 1, Boot(pin) = 0和BOOT_ADD0(optionbyte) = 0x0040
	nDBANK(bit) = 0, nDBOOT(bit) = 1, Boot(pin) = 1和BOOT_ADD1(optionbyte) = 0x0040
	nDBANK(bit)=0, nDBOOT(bit)=0, BOOT_ADDx(optionbyte)不在存储器范围内, 或者位于ICP存储器范围内
模式10	nDBANK(bit)=0, nDBOOT(bit)=0, BOOT_ADDx(optionbyte)在Flash存储器范围内, 且两个存储区均不包含有效代码
	Boot(pin) = 0和BOOT_ADD0(optionbyte) = 0x1FF0
模式11	Boot(pin) = 1和BOOT_ADD1(optionbyte) = 0x1FF0
	nBoot0(bit) = 0, nBoot1(bit) = 1, nBOOT0_SEL(bit) = 1且BOOT_LOCK(bit) = 0
	Boot0(pin) = 1, nBoot1(bit) = 1且nBOOT0_SEL (bit) = 0
	nBoot0(bit) = 1, nBOOT0_SEL(bit) = 1, BOOT_LOCK(bit) = 0且主Flash存储器为空
模式12	Boot0(pin) = 0, nBOOT0_SEL(bit) = 0, BOOT_LOCK(bit) = 0且主Flash存储器为空
	TZen = 0, Boot0(pin) = 0, nSWBoot0(bit) = 1且NSBOOTADD0 [24:0] = 0x017F200
	TZen = 0, Boot0(pin) = 1, nSWBoot0(bit) = 1且NSBOOTADD1 [24:0] = 0x017F200
	TZen = 0, nBoot0(bit) = 0, nSWBoot0(bit) = 0且NSBOOTADD1 [24:0] = 0x017F200
	TZen = 0, nBoot0(bit) = 1, nSWBoot0(bit) = 0且NSBOOTADD0 [24:0] = 0x017F200
	TZen = 1, Boot0(pin) = 0, nSWBoot0(bit) = 1且SECBOOTADD0 [24:0] = 0x01FF000 & RSSCMD = 0
	TZen = 1, Boot0(pin) = 1, nSWBoot0(bit) = 1 & RSSCMD = 0, BOOT_LOCK=0或 (BOOT_LOCK = 1且SECBOOTADD0 [24:0] = 0x01FF000)
	TZen = 1, nBoot0(bit) = 1, nSWBoot0(bit) = 0且SECBOOTADD0 [24:0] = 0x01FF000 & RSSCMD=0, BOOT_LOCK=0或 (BOOT_LOCK=1且SECBOOTADD0 [24:0] = 0x01FF000)
模式13	TZen=1, nBoot0(bit)=0, nSWBoot0(bit)=0&RSSCMD=0, BOOT_LOCK=0或BOOT_LOCK = 1且SECBOOTADD1 [24:0] = 0x01FF000
	TZen=1, RSSCMD=0x1C0, BOOT_LOCK=0或 (BOOT_LOCK=1且SECBOOTADD1 [24:0] = 0x01FF000)
	nBoot0(bit) = 0, nBoot1(bit) = 1且nSWBoot0(bit) = 0
	nBoot0(bit) = 1, nBoot1(bit) = 1, nSWBoot0(bit) = 0且用户Flash为空
模式14	nBoot1(bit) = 1, nSWBoot0(bit) = 1且Boot0(pin) = 1
	nBoot1(bit) = 1, nSWBoot0(bit) = 1, Boot0(pin) = 0且用户Flash为空
	BOOT_LOCK(bit) = 0, nBoot1(bit) = 1, Boot0(pin) = 1且nSWBoot0(bit) = 1
	BOOT_LOCK(bit) = 0, nBoot1(bit) = 1, nBoot0(bit) 0且nSWBoot0(bit) = 0
模式14	BOOT_LOCK(bit) = 0, Boot0(pin) = 0, nSWBoot0(bit) = 1, BFB2(bit)=1且两个存储区均不包含有效代码
	BOOT_LOCK(bit) = 0, nBoot0(bit), nSWBoot0(bit) = 0, BFB2(bit)=1且两个存储区均不包含有效代码



表2. 自举程序激活模式（续）

模式	激活条件
模式15	BOOT_LOCK(bit)=0, Boot0(pin) = 1, nBoot1(bit) = 1且nBoot0_SW(bit) = 1
	BOOT_LOCK(bit)=0, nBoot0(bit) = 0, nBoot1(bit) = 1且nBoot0_SW(bit) = 0
模式16	Boot0(pin) = 1, nBoot1(bit) = 1和nBoot0_SW(bit) = 1
	nBoot0 (bit) = 0, nBoot1(bit) = 1和nBoot0_SW(bit) = 0
	Boot0(pin) = 0, nBoot0_SW(bit) = 1且主Flash存储器为空

注: *BOOT_LOCK实现因产品而异, 请参阅参考手册了解详细信息。*

除了上述模式之外, 用户可通过从用户代码跳转到系统存储器来执行自举程序。跳转到自举程序之前, 用户必须:

- 禁用所有外设时钟
- 禁用所用的PLL
- 禁用中断
- 清空挂起的中断

通过离开自举程序激活条件并生成硬件复位, 或使用Go命令来执行用户代码, 可退出系统存储器自举模式。

注: *执行Go命令时, 在跳转到用户应用程序之前, 自举程序使用的外设寄存器不会初始化为默认复位值。如果要使用这些寄存器, 须在用户应用程序中对其重新配置。因此, 如果应用程序正在使用IWDG, 则必须调整IWDG预分频器值来满足应用程序的要求 (因为已将预分频器设置为最大值)。某些产品并未设置所有复位值。了解更多信息, 请参考每个产品自举程序版本的已知限制详情。*

注: *对于具有双存储区自举特性的STM32器件, 为了从用户代码跳转到系统存储器, 用户必须首先使用SYSCFG寄存器将系统存储器自举程序重新映射到地址0x00000000 (除了STM32F7系列), 然后跳转到自举程序。对于STM32F7系列, 用户必须禁用nDBOOT和/或nDBANK特性 (在选项字节中), 然后跳转到自举程序。*

注: *对于内置有自举程序、使用DFU/CAN接口、需要外部时钟源 (HSE) 进行DFU/CAN操作的STM32器件, 自举程序固件会基于内部振荡器时钟 (HSI、MSI) 动态检测HSE的值。当由于温度或其它条件导致内部振荡器精度发生变化并超出容差范围时 (理论值约为1%), 自举程序计算所得的HSE频率值可能不正确。在这种情况下, 自举程序DFU/CAN接口可能会出现故障, 甚至完全无法工作。*

4.2 自举程序标识

根据所用STM32器件的不同，自举程序可支持一个或多个嵌入式串行外设，用来将代码下载到内部Flash中。自举程序标识符（ID）可提供有关支持的串行外设的信息。

对于给定的STM32器件，自举程序通过以下各项标识：

1. **自举程序（协议）版本**：自举程序中使用的串行外设（例如USART、CAN、USB）通信协议的版本。可使用自举程序Get Version命令检索此版本。
2. **自举程序标识符（ID）**：STM32器件自举程序的版本，以**0xXY**形式的单字节代码表示，其中：
 - **X**指定器件自举程序所用的嵌入式串行外设：
 - X = 1：使用一个USART
 - X = 2：使用两个USART
 - X = 3：使用USART、CAN和DFU
 - X = 4：使用USART和DFU
 - X = 5：使用USART和I²C
 - X = 6：使用I²C
 - X = 7：使用USART、CAN、DFU和I²C
 - X = 8：使用I²C和SPI
 - X = 9：使用USART、CAN（或FDCAN）、DFU、I²C和SPI
 - X = 10：使用USART、DFU和I²C
 - X = 11：使用USART、I²C和SPI
 - X = 12：使用USART和SPI
 - X = 13：使用USART、DFU、I²C和SPI

- **Y**指定器件的自举程序版本

下面以自举程序ID 0x10为例。这表示仅使用一个USART的器件自举程序的第一个版本。

自举程序ID编程在器件系统存储器最后一个字节地址减1所对应的空间中，可通过自举程序“Read memory”命令来读取，或者通过使用JTAG/SWD直接访问系统存储器来读取。

注： *自举程序ID格式适用于除STM32F1xx系列之外的所有STM32器件系列。STM32F1xx的自举程序版本只适用于嵌入式器件的自举程序版本，不适用于其支持的协议。*

表 3提供STM32器件中内置的自举程序的识别信息。

表3.嵌入式自举程序

STM32 系列	设备		支持的串行外设	自举程序ID		自举程序 (协议) 版本
				ID	存储单元	
C0	STM32C011xx		USART1/ I2C1	0x51	0x1FFF17FE	USART (V3.1) I2C1 (V1.1)
	STM32C031xx		USART1/ I2C1	0x52	0x1FFF17FE	USART (V3.1) I2C1 (V1.1)
F0	STM32F05xxx/STM32F030x8		USART1/USART2	0x21	0x1FFFF7A6	USART (V3.1)
	STM32F03xx4/6		USART1	0x10	0x1FFFF7A6	USART (V3.1)
	STM32F030xC		USART1/ I2C1	0x52	0x1FFFF796	USART (V3.1) I2C1 (V1.0)
	STM32F04xxx		USART1/USART2/ DFU (USB设备FS) / I2C1	0xA1	0x1FFFF6A6	USART (V3.1) DFU (V2.2) I2C (V1.0)
	STM32F071xx/072xx		USART1/USART2/ I2C1/ DFU (USB设备FS)	0xA1	0x1FFFF6A6	USART (V3.1) DFU (V2.2) I2C (V1.0)
F0	STM32F070x6		USART1/USART2 / DFU (USB设备FS) /I2C1	0xA2	0x1FFFF6A6	USART (V3.1) DFU (V2.2) I2C (V1.0)
	STM32F070xB		USART1/USART2/ DFU (USB设备FS) /I2C1	0xA3	0x1FFFF6A6	USART (V3.1) DFU (V2.2) I2C (V1.0)
	STM32F09xxx		USART1/USART2/ I2C1	0x50	0x1FFFF796	USART (V3.1) I2C (V1.0)
F1	STM32F10xxx	小容量	USART1	NA	NA	USART (V2.2)
		中容量	USART1	NA	NA	USART (V2.2)
		大容量	USART1	NA	NA	USART (V2.2)
		中容量超值型	USART1	0x10	0x1FFFF7D6	USART (V2.2)
		大容量超值型	USART1	0x10	0x1FFFF7D6	USART (V2.2)
	STM32F105xx/107xx		USART1/USART2 (重映射) / CAN2 (重映射) / DFU (USB设备)	NA	NA	USART (V2.2 ⁽¹⁾) CAN (V2.0) DFU (V2.2)
STM32F10xxx超大容量		USART1/USART2 (重映射)	0x21	0x1FFFF7D6	USART (V3.0)	
F2	STM32F2xxxx		USART1/USART3	0x20	0x1FFF77DE	USART (V3.0)
			USART1/USART3/ CAN2/ DFU (USB设备FS)	0x33	0x1FFF77DE	USART (V3.1) CAN (V2.0) DFU (V2.2)

表3.嵌入式自举程序（续）

STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
F3	STM32F373xx	USART1/USART2/ DFU (USB设备FS)	0x41	0x1FFFF7A6	USART (V3.1) DFU (V2.2)
	STM32F378xx	USART1/USART2/ I2C1	0x50	0x1FFFF7A6	USART (V3.1) I2C (V1.0)
	STM32F302xB (C) /303xB (C)	USART1/USART2/ DFU (USB设备FS)	0x41	0x1FFFF796	USART (V3.1) DFU (V2.2)
	STM32F358xx	USART1/USART2/ I2C1	0x50	0x1FFFF796	USART (V3.1) I2C (V1.0)
	STM32F301xx/302x4 (6/8)	USART1/USART2/ DFU (USB设备FS)	0x40	0x1FFFF796	USART (V3.1) DFU (V2.2)
	STM32F318xx	USART1/USART2/ I2C1/ I2C3	0x50	0x1FFFF796	USART (V3.1) I2C (V1.0)
	STM32F302xD (E) /303xD (E)	USART1/USART2/ DFU (USB设备FS)	0x40	0x1FFFF796	USART (V3.1) DFU (V2.2)
	STM32F303x4 (6/8) /334xx/328xx	USART1/USART2/ I2C1	0x50	0x1FFFF796	USART (V3.1) I2C (V1.0)
	STM32F398xx	USART1/USART2/ I2C1/I2C3	0x50	0x1FFFF796	USART (V3.1) I2C (V1.0)

表3.嵌入式自举程序（续）

STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
F4	STM32F40xxx/41xxx	USART1/USART3/ CAN2/ DFU (USB设备FS)	0x31	0x1FFF77DE	USART (V3.1) CAN (V2.0) DFU (V2.2)
		USART1/USART3/ CAN2 / DFU (USB设备FS) /I2C1/I2C2/I2C3/ SPI1/SPI2	0x91	0x1FFF77DE	USART (V3.1) CAN (V2.0) DFU (V2.2) SPI (V1.1) I2C (V1.0)
	STM32F42xxx/43xxx	USART1/USART3/ CAN2 / DFU (USB设备FS) / I2C1	0x70	0x1FFF76DE	USART (V3.1) CAN (V2.0) DFU (V2.2) I2C (V1.0)
		USART1/USART3/ CAN2 / DFU (USB设备FS) / SPI1/ SPI2/ SPI4 I2C1/I2C2/I2C3/	0x91	0x1FFF76DE	USART (V3.1) CAN (V2.0) DFU (V2.2) SPI (V1.1) I2C (V1.0)
	STM32F401xB (C)	USART1/USART2/ DFU (USB设备FS) / I2C1/I2C2/I2C3/ SPI1/SPI2/ SPI3	0xD1	0x1FFF76DE	USART (V3.1) DFU (V2.2) SPI (V1.1) I2C (V1.0)
	STM32F401xD (E)	USART1/USART2/ DFU (USB设备FS) / I2C1/I2C2/I2C3/ SPI1/SPI2/ SPI3	0xD1	0x1FFF76DE	USART (V3.1) DFU (V2.2) SPI (V1.1) I2C (V1.1)
	STM32F410xx	USART1/USART2/ I2C1/I2C2/I2C4 SPI1/SPI2	0xB1	0x1FFF76DE	USART (V3.1) I2C (V1.2) SPI (V1.1)
	STM32F411xx	USART1/USART2/ DFU (USB设备FS) / SPI1/SPI2/ SPI3 I2C1/I2C2/I2C3	0xD0	0x1FFF76DE	USART (V3.1) DFU (V2.2) SPI (V1.1) I2C (V1.1)
	STM32F412xx	USART1/USART2/ USART3/CAN2/ DFU (USB设备FS) / I2C1/I2C2/I2C3/I2C4/ SPI1/SPI3/SPI4	0x91	0x1FFF76DE	USART (V3.1) CAN (V2.0) DFU (V2.2) SPI (V1.1) I2C (V1.2)
	STM32F413xx/423xx	USART1/USART2/ USART3/CAN2/ DFU (USB设备FS) / I2C1/I2C2/I2C3/I2C4/ SPI1/SPI3/SPI4	0x90	0x1FFF76DE	USART (V3.1) CAN (V2.0) DFU (V2.2) I2C (V1.2) SPI (V1.1)
	STM32F446xx	USART1/USART3/CAN2/DFU (USB设备FS) / I2C1/I2C2/I2C3/SPI1/ SPI2/ SPI4	0x90	0x1FFF76DE	USART (V3.1) CAN (V2.0) DFU (V2.2) SPI (V1.1) I2C (V1.2)
	STM32F469xx/479xx	USART1/USART3/ I2C1/I2C2/I2C3/ CAN2/ DFU (USB设备FS) / SPI1/ SPI2/ SPI4	0x90	0x1FFF76DE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)

表3.嵌入式自举程序（续）

STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
F7	STM32F72xxx/73xxx	USART1/USART3/ CAN1/ DFU (USB设备FS) / I2C1/I2C2/I2C3/ SPI1/SPI2/SPI4	0x90	0x1FF0EDBE	USART (V3.1) CAN (V2.0) DFU (V2.2) I2C (V1.2) SPI (V1.2)
	STM32F74xxx/75xxx	USART1/USART3/ I2C1/I2C2/I2C3/ CAN2/ DFU (USB设备FS)	0x70	0x1FF0EDBE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2)
		USART1/USART3/ I2C1/I2C2/I2C3/ CAN2/ DFU (USB设备FS) / SPI1/SPI2/SPI4	0x90	0x1FF0EDBE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.2)
	STM32F76xxx/77xxx	USART1/USART3/ CAN2/ DFU (USB设备FS) / I2C1/I2C2/I2C3/ SPI1/SPI2/SPI4	0x93	0x1FF0EDBE	USART (V3.1) CAN (V2.0) DFU (V2.2) I2C (V1.2) SPI (V1.2)
G0	STM32G07xxx/08xxx	USART1/USART2/ USART3/I2C1/I2C2/ SPI1/SPI2	0xB2	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1)
	STM32G03xxx/04xxx	USART1/USART2/ I2C1/I2C2	0x53	0x1FFF1FFE	USART (V3.1) I2C (V1.2)
	STM32G0B0xx	USART1/USART2/USART3 I2C1/I2C2 SPI1/SPI2 DFU (USB设备FS)	0xD0	0x1FFF9FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2)
	STM32G0B1xx/0C1xx	USART1/USART2/USART3 I2C1/I2C2 SPI1/SPI2 DFU (USB设备FS) FDCAN	0x92	0x1FFF9FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2) FDCAN (V1.0)
	STM32G05xxx/061xx	USART1/USART2 I2C1/I2C2	0x51	0x1FFF1FFE	USART (V3.1) I2C (V1.2)
G4	STM32G431xx/441xx	USART1/USART2/USART3 I2C2/I2C3 SPI1/SPI2 DFU (USB设备FS)	0xD4	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2)
	STM32G47xxx/48xxx	USART1/USART2/USART3 I2C2/I2C3/I2C4 SPI1/SPI2 DFU (USB设备FS)	0xD5	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2)
	STM32G491xx/4A1xx	USART1/USART2/USART3 I2C2/I2C3 SPI1/SPI2 DFU (USB设备FS)	0xD2	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2)

表3.嵌入式自举程序（续）

STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
H7	STM32H72xxx/73xxx	USART1/USART2/USART3 I2C1/I2C2/I2C3/ DFU (USB设备FS) / SPI1/SPI2/SPI3/SPI4/ FDCAN1	0x93	0x1FF1E7FE	USART (V3.1) I2C (V1.2) DFU (V2.2) SPI (V1.1) FDCAN (V1.0)
	STM32H74xxx/75xxx	USART1/USART2/USART3 I2C1/I2C2/I2C3/ DFU (USB设备FS) / SPI1/SPI2/SPI3/SPI4/ FDCAN1	0x90	0x1FF1E7FE	USART (V3.1) I2C (V1.1) DFU (V2.2) SPI (V1.2) FDCAN (V1.0)
	STM32H7A3xx/B3xx	USART1/USART2/USART3 I2C1/I2C2/I2C3/ DFU (USB设备FS) / SPI1/SPI2/SPI3/SPI4/ FDCAN1	0x91	0x1FF13FFE	USART (V3.1) I2C (V1.2) DFU (V2.2) SPI (V1.2) FDCAN (V1.0)
L0	STM32L01xxx/02xxx	USART2/SPI1	0xC3	0x1FF00FFE	USART (V3.1) SPI (V1.1)
	STM32L031xx/041xx	USART2/SPI1	0xC0	0x1FF00FFE	USART (V3.1) SPI (V1.1)
	STM32L05xxx/06xxx	USART1/USART2/ SPI1/ SPI2	0xC0	0x1FF00FFE	USART (V3.1) SPI (V1.1)
	STM32L07xxx/08xxx	USART1/USART2/ DFU (USB设备FS)	0x41	0x1FF01FFE	USART (V3.1) DFU (V2.2)
USART1/USART2/ SPI1/SPI2/ I2C1/I2C2		0xB2	0x1FF01FFE	USART (V3.1) SPI (V1.1) I2C (V1.2)	
L1	STM32L1xxx6 (8/B)	USART1/USART2	0x20	0x1FF00FFE	USART (V3.0)
	STM32L1xxx6 (8/B) A	USART1/USART2	0x20	0x1FF00FFE	USART (V3.1)
	STM32L1xxxC	USART1/USART2/ DFU (USB设备FS)	0x40	0x1FF01FFE	USART (V3.1) DFU (V2.2)
	STM32L1xxxD	USART1/USART2/ DFU (USB设备FS)	0x45	0x1FF01FFE	USART (V3.1) DFU (V2.2)
	STM32L1xxxE	USART1/USART2/ DFU (USB设备FS)	0x40	0x1FF01FFE	USART (V3.1) DFU (V2.2)

表3.嵌入式自举程序（续）

STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
L4	STM32L412xx/422xx	USART1/USART2/USART3 I2C1/I2C2/I2C3/ DFU (USB设备FS) / SPI1/SPI2	0xD1	0x1FFF6FFE	USART (V3.1) I2C (V1.2) DFU (V2.2) SPI (V1.1)
	STM32L43xxx/44xxx	USART1/USART2/USART3/ I2C1/I2C2/I2C3/ CAN1/ DFU (USB设备FS) / SPI1/SPI2	0x91	0x1FFF6FFE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)
	STM32L45xxx/46xxx	USART1/USART2/USART3/ I2C1/I2C2/I2C3/ CAN1/ DFU (USB设备FS) / SPI1/SPI2	0x92	0x1FFF6FFE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)
	STM32L47xxx/48xxx	USART1/USART2/ USART3/ I2C1/I2C2/I2C3/ DFU (USB设备FS)	0xA3	0x1FFF6FFE	USART (V3.1) I2C (V1.2) DFU (V2.2)
		USART1/USART2/ USART3/ I2C/I2C2/I2C3/ SPI1/SPI2/ CAN1/ DFU (USB设备FS)	0x92	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) CAN (V2.0) DFU (V2.2)
	STM32L496xx/4A6xx	USART1/USART2/USART3/ I2C1/I2C2/I2C3/ CAN1/ DFU (USB设备FS) / SPI1/SPI2	0x93	0x1FFF6FFE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)
	STM32L4Rxxx/STM32L4Sxxx	USART1/USART2/USART3/ I2C1/I2C2/I2C3/ CAN1/ DFU (USB设备FS) / SPI1/SPI2	0x95	0x1FFF6FFE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)
STM32L4P5xx /Q5xx	USART1/USART2/USART3 I2C1/I2C2/I2C3/ CAN1/ DFU (USB设备FS) / SPI1/SPI2	0x90	0x1FFF6FFE	USART (V3.1) I2C (V1.2) CAN (V2.0) DFU (V2.2) SPI (V1.1)	
L5	STM32L552xx/562xx	USART1/USART2/USART3 I2C1/I2C2/I2C3 SPI1/SPI2/SPI3 DFU (USB设备FS) FDCAN1	0x92	0x0BF97FFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2) FDCAN (V1.0)

表3.嵌入式自举程序（续）

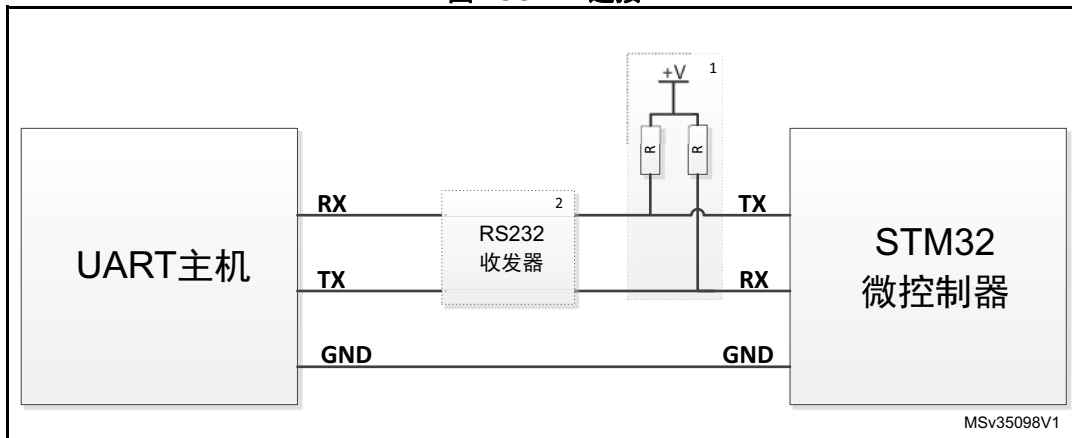
STM32 系列	设备	支持的串行外设	自举程序ID		自举程序 (协议) 版本
			ID	存储单元	
WB	STM32WB10xx/15xx	USART1/ I2C1/ SPI1	0xB1	0x1FFF6FFE	USART (V3.1) I2C (V1.2) SPI (V1.1)
	STM32WB30xx/35xx/50xx/55xx	USART1/ I2C1/I2C3 SPI1/SPI2 DFU (USB设备FS)	0xD5	0x1FFF6FFE	USART (V3.2) I2C (V1.2) SPI (V1.1) DFU (V2.2)
WL	STM32WLE5xx/55xx	USART1/USART2 SPI1/SPI2	0xC4	0x1FFF3EFE	USART (V3.1) SPI (V1.1)
U5	STM32U575xx/ STM32U585xx	USART1/USART2/USART3 I2C1/I2C2/I2C3 SPI1/SPI2/SPI3 DFU (USB设备FS) FDCAN1	0x92	0x0BF99EFE	USART (V3.1) I2C (V1.2) SPI (V1.1) DFU (V2.2) FDCAN (V1.1)

1. 对于互连型器件，USART自举程序将返回V2.0协议版本而不是V2.2协议版本。有关详细信息，请参见www.st.com提供的“STM32F105xx和STM32F107xx版本Z”勘误表。

4.3 硬件连接要求

要使用USART自举程序，主机必须通过串口电缆连接到所需USARTx接口的RX和TX引脚。

图1.USART连接

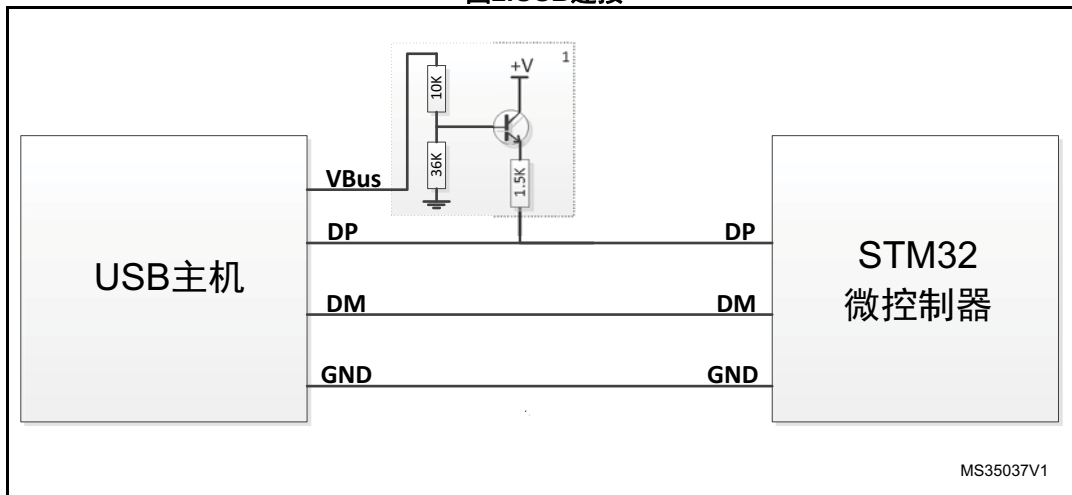


1. 如果上拉电阻未连接到主机端，则须添加一个上拉电阻。
2. 必须连接RS232收发器，来适配STM32器件和主机之间的电平（3.3 - 12 V）。

注：+V典型值为3.3 V，R典型值为100 KΩ。这些值取决于应用和所用硬件。

要使用DFU，应将微控制器的USB接口和USB主机（即PC）连接起来。

图2.USB连接

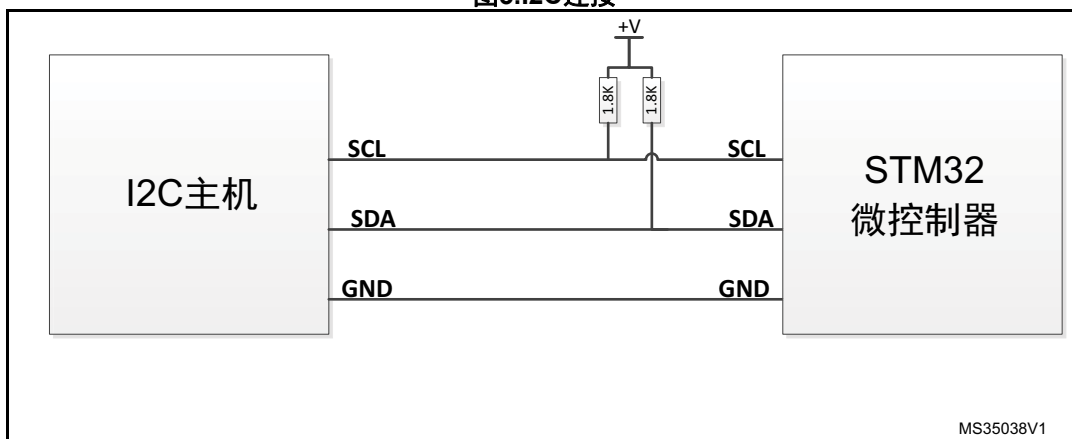


1. 需要时，此附加电路可利用VBus将上拉电阻连接到（DP）引脚。请参考产品章节（描述了系统存储器自举模式中STM32配置的表格）来了解是否需要将外部上拉电阻连接到（DP）引脚。

注：+V典型值为3.3 V。该值取决于应用和所用硬件。

要使用I2C自举程序，可利用数据（SDA）和时钟（SCL）引脚，将主机（主设备）和所需I2Cx接口（从设备）连接起来。SDA线和SCL线都须连接一个1.8 KΩ的上拉电阻。

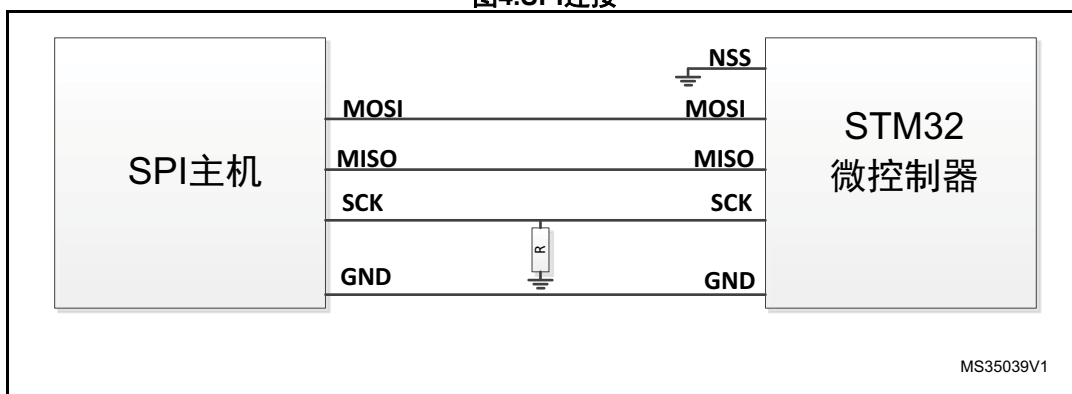
图3.I2C连接



注: +V典型值为3.3 V。该值取决于应用和所用硬件。

要使用SPI自举程序,可利用MOSI、MISO和SCK引脚,将主机(主设备)和所需SPIx接口(从设备)连接起来。NSS引脚必须连接到GND。须连接一个下拉电阻到SCK线。

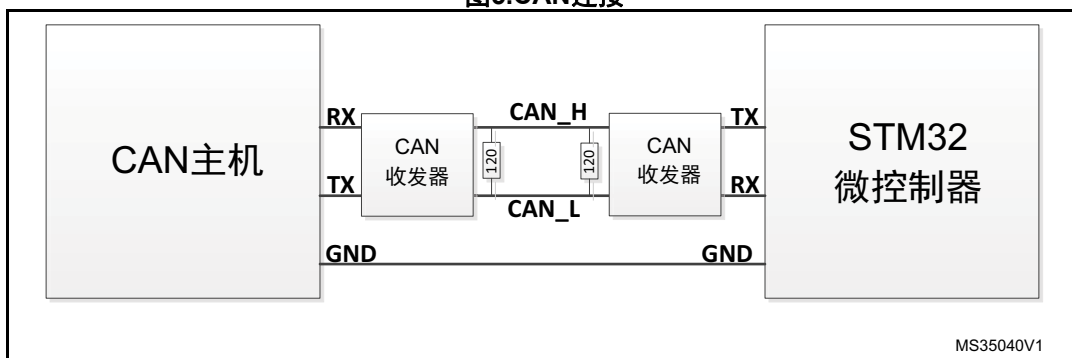
图4.SPI连接



注: R的典型值为10 K Ω 。该值取决于应用和所用硬件。

要使用CAN接口,主机必须通过CAN收发器和串口电缆连接到所需CANx接口的RX和TX引脚。须添加一个120 Ω 的电阻作为终端电阻。

图5.CAN连接



注： 当自举程序固件支持DFU时，强制要求在其他接口选择阶段中，无USB主机连接到USB外设。选择阶段之后，用户可插入一个USB线缆而不影响所选自举程序执行，产生系统复位的命令除外。

在自举程序启动时（检测阶段），建议将不用的自举程序接口（USART_RX、SPI_MOSI、CAN_RX和USB D+/D-线，如果存在的话）的RX引脚保持为已知的（低或高）电平。在检测阶段将这些引脚浮空，可能会激活不用的接口。

4.4 自举程序存储器管理

因此，所有使用自举程序命令的写操作只能采用字对齐形式（地址应为4的倍数）。要写入的数据量也必须为4的倍数

（允许非对齐半页写操作地址）。

一些产品嵌入了具有特殊功能的自举程序：

- 一些产品不支持批量擦除操作。要使用自举程序执行批量擦除操作，可采用以下两种方法：
 - 使用Erase命令逐个擦除所有扇区
 - 将保护级别设为级别1。然后再设为级别0（使用Read protect命令，然后使用Read Unprotect命令）。执行此操作后将批量擦除内部Flash内容。
- 除了标准存储器（内部Flash、内部SRAM、选择字节和系统存储器）之外，STM32L1和L0系列产品的自举程序固件还支持数据存储器。起始地址和该区域的大小取决于产品，请参考产品参考手册来获取更多信息。数据存储器允许执行读和写操作，但不能使用Erase命令进行擦除。向数据存储单元执行写操作时，自举程序固件会在任何写操作之前对此单元执行擦除操作。数据存储器的写操作必须采用字对齐形式（要写入的地址应为4的倍数），并且要写入的数据量也必须为4的倍数。要擦除数据存储单元，可在此单元中写入零。
- 除了标准存储器（内部Flash、内部SRAM、选择字节和系统存储器）之外，STM32F2、F4、F7和L4系列产品的自举程序固件还支持OTP存储器。起始地址和该区域的大小取决于产品，请参考产品参考手册来获取更多信息。OTP存储器允许执行读和写操作，但不能使用Erase命令进行擦除。向OTP存储单元执行写操作时，需确保相关保护位为非零值。
- 对于STM32F2、F4和F7系列产品，内部Flash存储器写操作格式取决于电压范围。默认支持按照字节格式（不支持半字、字和双字操作）执行写操作。要提高写操作的速度，用户必须使用适当的电压范围，可支持以半字、字或双字执行写操作，并利用自举程序软件通过虚拟内存单元来实时更新该配置。该存储单元并非物理单元，但可根据所使用的协议通过常见的自举程序读/写操作进行读写。该存储单元包含4个字节，如表4所述。支持1、2、3或4字节访问。但是，保留字节必须保持默认值（0xFF），否则请求将收到NACK应答。

表4.利用自举程序进行STM32 F2、F4和F7电压范围配置

地址	大小	说明
0xFFFF0000	1字节	该字节控制电压范围的当前值。 0x00: 电压范围 [1.8 V, 2.1 V] 0x01: 电压范围 [2.1 V, 2.4 V] 0x02: 电压范围 [2.4 V, 2.7 V] 0x03: 电压范围 [2.7 V, 3.6 V] 0x04: 电压范围 [2.7 V, 3.6 V] 并且使用双字写/擦除操作。 这种情况下, 必须通过VPP引脚提供9 V电压 (有关双字写操作步骤的详细信息, 请参见产品参考手册)。 其它: 所有其它值都不适用, 否则会收到NACK应答。
0xFFFF0001	1字节	Reserved 0xFF: 默认值。 其它: 所有其它值都不适用, 否则会收到NACK应答。
0xFFFF0002	1字节	Reserved 0xFF: 默认值。 其它: 所有其它值都不适用, 否则会收到NACK应答。
0xFFFF0003	1字节	Reserved 0xFF: 默认值。 其它: 所有其它值都不适用, 否则会收到NACK应答。

下表中根据自举程序命令列出了其有效的存储区。

表5.Write、Read、Erase和Go命令所支持的存储区

存储区	Write指令	Read指令	Erase指令	Go指令
Flash	支持	支持	支持	支持
RAM	支持	支持	不支持	支持
系统存储器	不支持	支持	不支持	不支持
数据存储器	支持	支持	不支持	不支持
OTP存储器	支持	支持	不支持	不支持

4.5 自举程序UART波特率检测

对于UART接口波特率检测，在不同STM32器件上实现了两种机制：

- 使用内部HSI和定时器（使用GPIO作为输入，如AN3155所述检测下降沿和上升沿）进行软件波特率检测。
使用此机制的器件可能受软件抖动（波特率计算的变量错误）的影响，抖动量可能高达±5%。
因此，在这种情况下，连接到STM32自举程序UART接口的主机应支持±5%的波特率偏差。
软件抖动值是变量，每次重试时都不一样，因此可使用多次连接重来克服软件抖动（连接并检查自举程序回答是否正确，如果回答不正确，复位器件并重新尝试连接，直至收到正确的回答为止。在收到正确回答后，其余通信不受软件抖动的影响）。此外，还可通过降低波特率值（即使用56000 bps而非115200 bps）来减少软件抖动。
[表 6](#)提供了波特率为115200 bps时的最大软件抖动值。
波特率越小，软件抖动越小。
- 使用UART自动波特率功能进行波特率检测。使用此机制的器件不会产生任何软件抖动。

表6.自举程序USART检测的软件抖动计算

系列	波特率检测方法	最大软件抖动 对于115200 bps
STM32F0	软件波特率检测	-1%
STM32F1	软件波特率检测	-3%
STM32F2	软件波特率检测	-5%
STM32F3	软件波特率检测	-2%
STM32F4	软件波特率检测	-6%
STM32F7	软件波特率检测	-6%
STM32L0	软件波特率检测	-2%
STM32L1	软件波特率检测	-3%
STM32L4	软件波特率检测	-5%
STM32G07x/8x UART3 STM32G03x/4x UART2	软件波特率检测	-4%
STM32G07x/8x UART1/UART2 STM32G03x/4x UART1	自动波特率	N/A
STM32G4	自动波特率	N/A
STM32H7	自动波特率	N/A
STM32WB	自动波特率	N/A
STM32WL	自动波特率	N/A

4.6 编程限制

在使用自举程序接口对Flash存储器执行写访问时，必须根据表 7维持设定地址的对齐。如果执行写操作的地址未对齐，写操作将失败且所有后续的程序操作也将失败。

表7.STM32产品的Flash存储器对齐限制

系列	对齐
STM32F0	4字节
STM32F1	4字节
STM32F2	4字节
STM32F3	4字节
STM32F4	4字节
STM32F7	8字节
STM32G0	4字节
STM32G4	4字节
STM32H7	8字节
STM32L0	8字节
STM32L1	8字节
STM32L4	8字节
STM32L5	16字节
STM32WB	8字节
STM32WL	8字节
STM32U5	16字节

对齐的示例：

- 4字节：0x08000014对齐且成功，0x08000012未对齐且失败
- 8字节：0x08000010对齐且成功，0x08000014未对齐且失败

注：在某些产品（STM32F4和STM32F7）上，可通过对器件特性空间的写操作修改对齐限制。

4.7 ExitSecureMemory特性

安全存储区用于将处理敏感信息（秘密信息）的安全自举代码/数据与应用代码隔离开来：

- 通过FLASH_CR寄存器中的SEC_PROT安全存储位（写入一次）控制访问
- 在自举时执行一次，然后通过写入安全存储位进行锁定
 - 受保护代码：隐藏在安全存储区中，直至下一次复位解锁SEC_PROT位
- 通过选项字节（Flash存储器FLASH_SEC_R寄存器中的SEC_SIZE）定义宽度（Flash页数）

ExitSecureMemory是一款在系统存储器上开发并驻留的软件。当用户自举代码跳转到它时，软件允许SEC_PROT位置为“1”，然后跳转到应用代码。在跳转到ExitSecureMemory函数之前，必须将SEC_SIZE置为需用的值。

如图 6所示，有两种跳转方法可供客户使用：

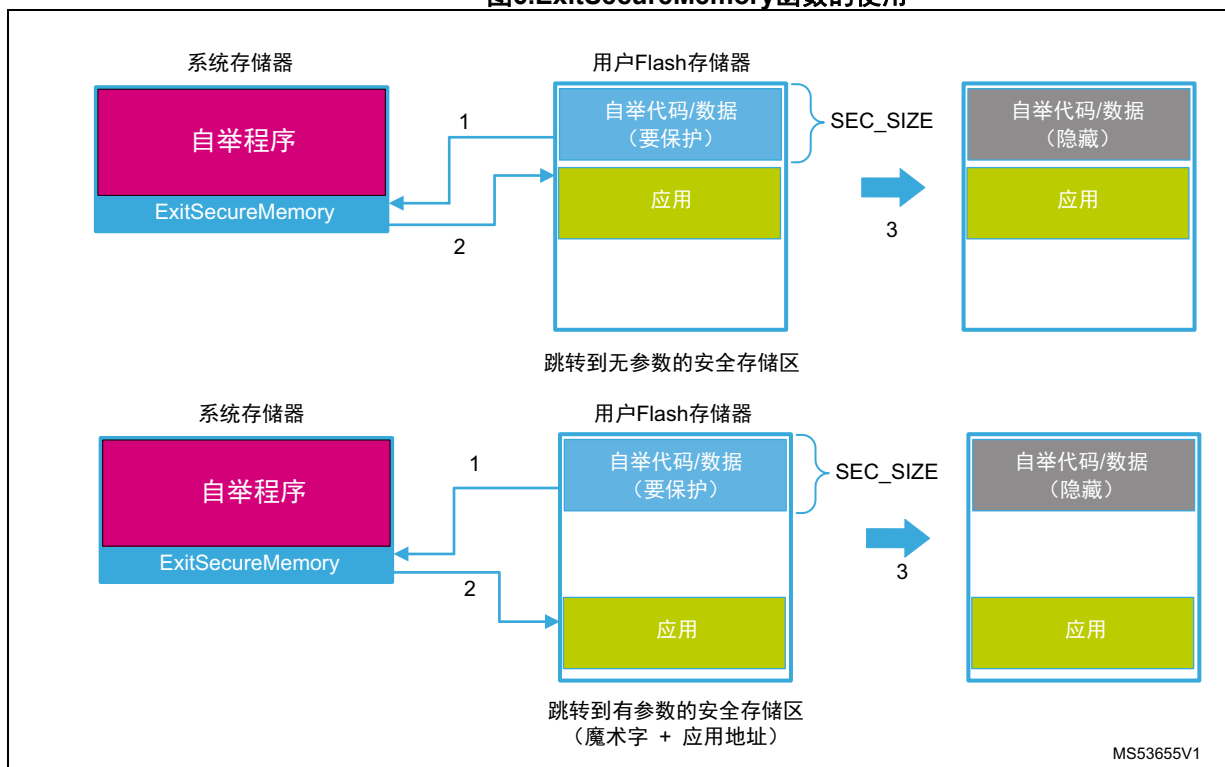
跳转到无参数的安全存储器函数

在这种情况下，必须在定义的安全存储器之后立即加载应用。

跳转到使用两个参数的安全存储器函数

1. 魔术字
 - 0x08192A3C
 - 对于单存储区产品，用于Flash中的安全自举代码/数据和跳转
 - 对于双存储区产品，用于Bank1中的安全自举代码/数据和跳转
 - 0x08192A3D
 - 对于双存储区产品，用于安全自举代码/数据并跳转到Bank2中的应用
2. 用户地址 = 应用地址
 - 在这种情况下，可将应用加载到任何需要的地址（根据定义的用户地址）

图6.ExitSecureMemory函数的使用



注： 请参阅参考手册了解关于选项字节配置的更多信息。

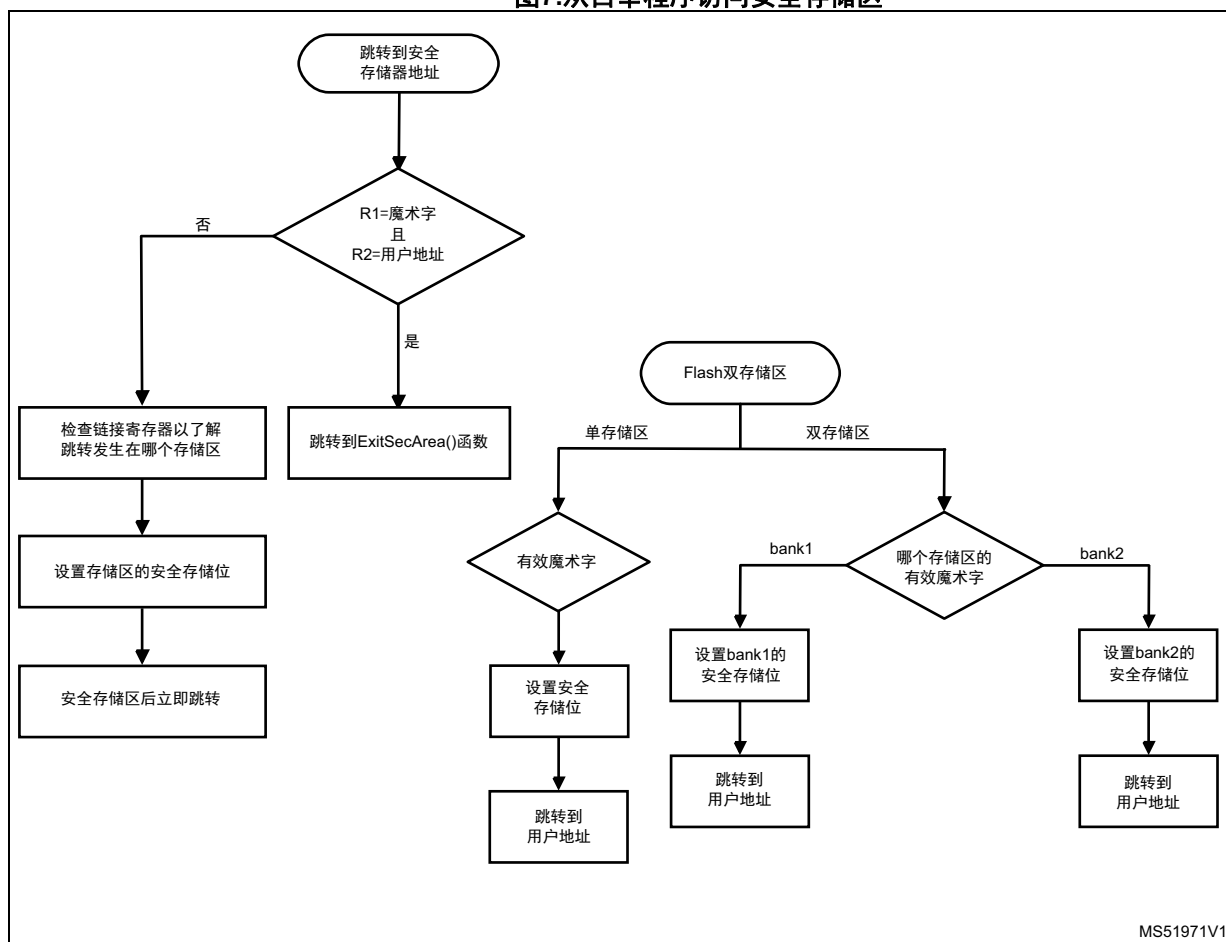
注： 附录A中提供了可用于调用“ExitSecureMemory”的函数示例。

表8.ExitSecureMemory入口地址

MCU		ExitSecureMemory??
STM32G0	STM32G07xxx/08xxx	0x1FFF6800
	STM32G03xxx/04xxx	0x1FFF1E00
	STM32G0Bxxx/0Cxxx	0x1FFF6800
	STM32G05xxx/061xx	0x1FFF6800
STM32G4	STM32G47xxx/48xxx	0x1FFF6800
	STM32G431xx/441xx	0x1FFF6800
	STM32G491xx/4A1xx	0x1FFF6800

有关详细信息，请参见。图 7

图7.从自举程序访问安全存储区



MS51971V1

1. 自举程序不检查用户地址的完整性，由用户负责确保要跳转到的地址的有效性。

5 STM32C011xx器件自举程序

5.1 自举程序配置

STM32C011xx自举程序通过应用模式11激活（参见表 2：自举程序激活模式）。表 13显示了该自举程序使用的硬件资源。

表9. 系统存储器自举模式下STM32C011xx器件的配置

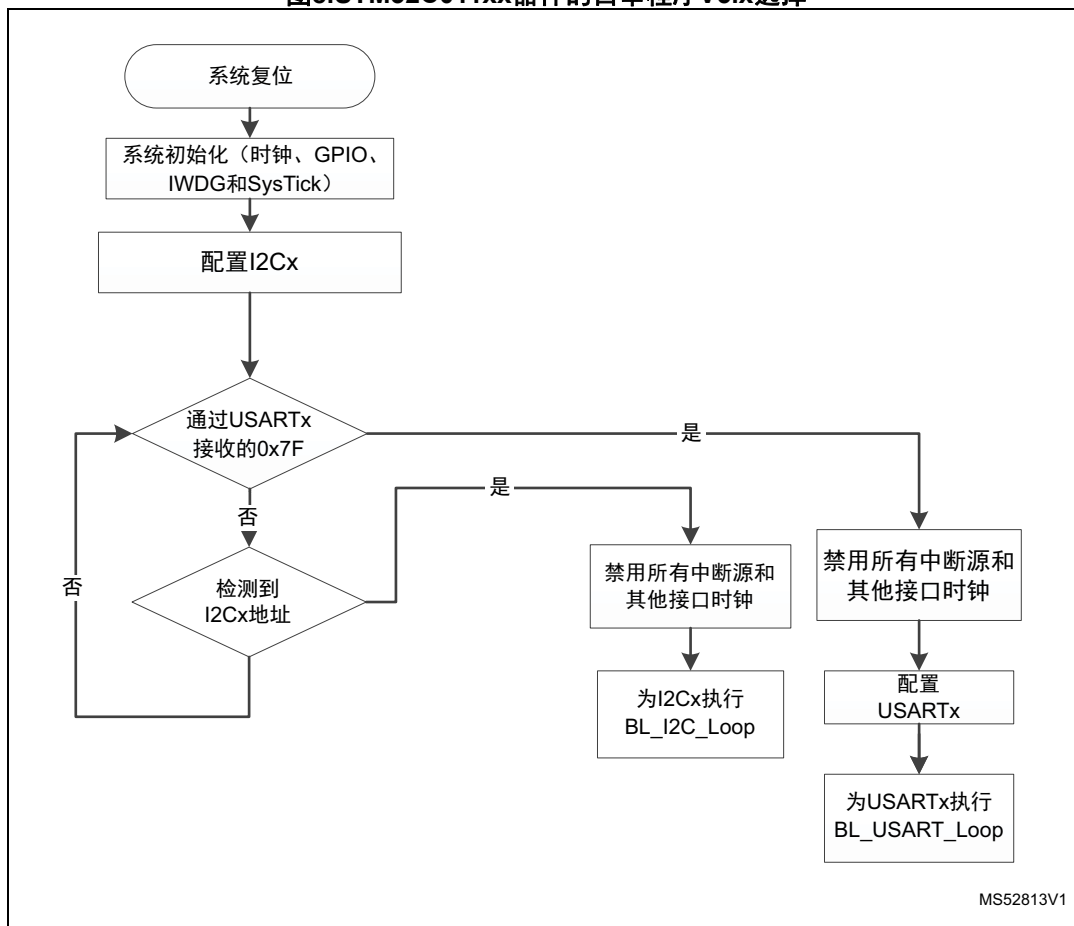
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为24 MHz（无PLL）
	RAM	-	自地址0x20000000起的3.5KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFF0000起的6 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址： @0x1FFF1600
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1100100x （其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

注：在WLCSP12、SO8N、TSSOP20和UFQFN20封装中，USART1 PA9/PA10 IO在PA11/PA12上重映射。

5.2 自举程序选择

图 10显示了自举程序选择机制。

图8.STM32C011xx器件的自举程序V5.x选择



5.3 自举程序版本

下表列出了STM32C011xx器件自举程序版本。

表10.STM32C011xx自举程序版本

自举程序版本号	说明	已知限制
V5.1	初始自举程序版本	无

6 STM32C031xx器件自举程序

6.1 自举程序配置

STM32C031xx自举程序通过应用模式11激活（参见表 2：自举程序激活模式）。表 13显示了该自举程序使用的硬件资源。

表11. 系统存储器自举模式下STM32C031xx器件的配置

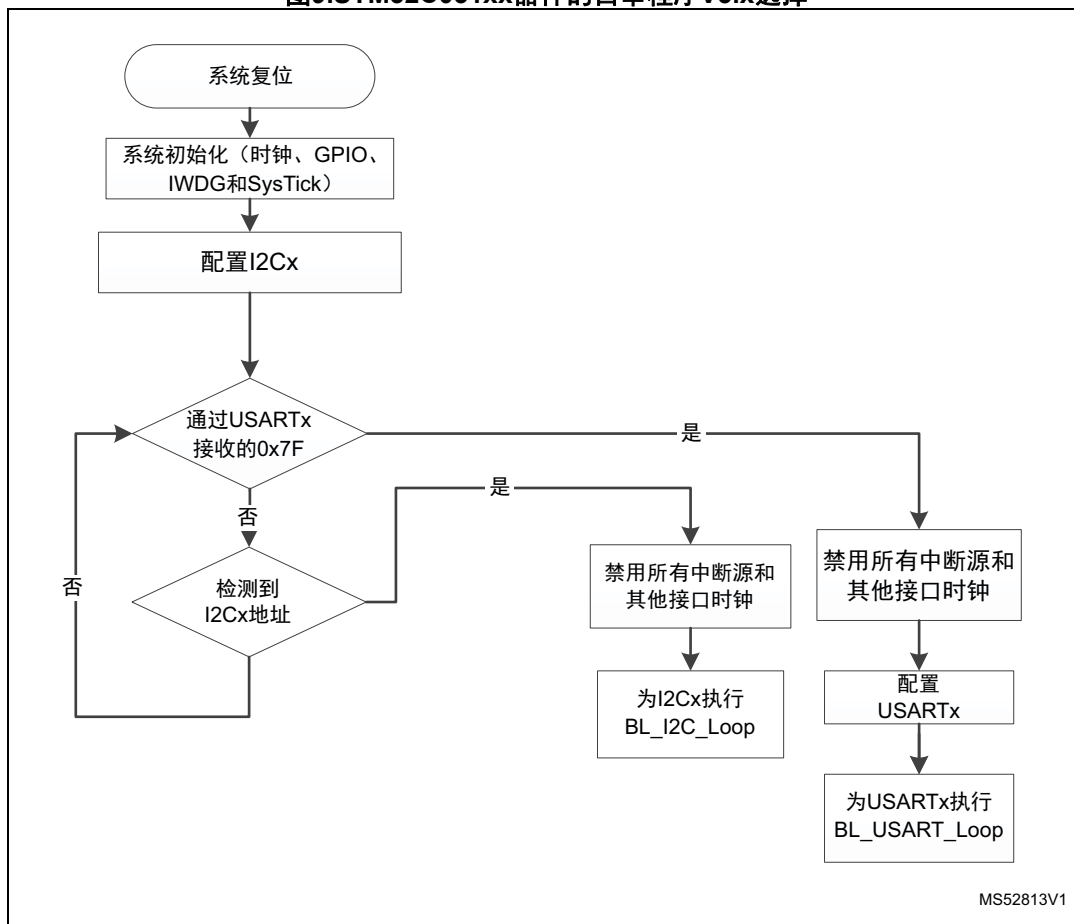
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为24 MHz（无PLL）
	RAM	-	自地址0x20000000起的3.5KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFF0000起的6 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址： @0x1FFF1600
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1100011x （其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

注：在TSSOP20和UFQFN28封装中，USART1 PA9/PA10 IO在PA11/PA12上重映射。

6.2 自举程序选择

图 10显示了自举程序选择机制。

图9.STM32C031xx器件的自举程序V5.x选择



6.3 自举程序版本

下表列出了STM32C031xx器件自举程序版本。

表12.STM32C031xx自举程序版本

自举程序版本号	说明	已知限制
V5.2	初始自举程序版本	无

7 STM32F03xx4/6器件自举程序

7.1 自举程序配置

STM32F03xx4/6自举程序通过应用模式2激活（参见表 2：自举程序激活模式）。表 13显示了该自举程序使用的硬件资源。

表13.系统存储器自举模式下STM32F03xx4/6器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于24 MHz（使用由HSI驱动 的PLL）。 1个Flash等待周期。
	RAM	-	自地址0x20000000起的2 KB空间供自举程 序固件使用。
	系统存储器	-	自地址0x1FFFE000起的3 KB空间包含自举 程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大 值，并定期进行刷新以防止看门狗复位 （如果用户事先使能硬件IWDG选项）。
USART1自举程序 (PA10/PA9)	USART1	启用	初始化后，USART1的配置为：8位，偶校验 位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输 入上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用 推挽，上拉模式。
USART1自举程序 (PA14/PA15)	USART1	启用	初始化后，USART1的配置为：8位，偶校验 位和1个停止位。
	USART1_RX引脚	输入	PA15引脚：USART1处于接收模式。用于输 入上拉模式。
	USART1_TX引脚	输出	PA14引脚：USART1处于发送模式。用于复 用推挽，上拉模式。
USART1自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

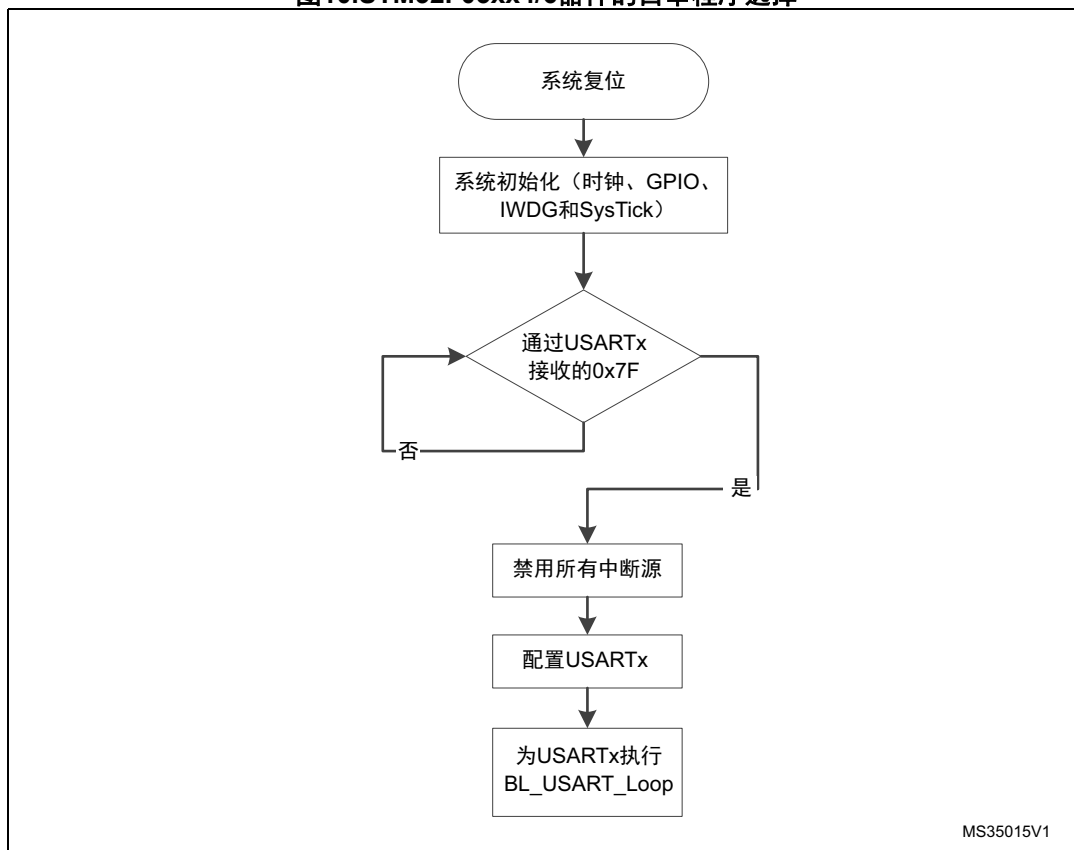
系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

注：STM32F03xx4/6器件在自举程序模式下自举之后，系统发生复位之前，无法再实现串行线调试（SWD）通信。这是因为SWD使用的PA14引脚（SWCLK）已由自举程序使用。

7.2 自举程序选择

图 10显示了自举程序选择机制。

图10.STM32F03xx4/6器件的自举程序选择



7.3 自举程序版本

下表列出了STM32F03xx4/6器件自举程序版本。

表14.STM32F03xx4/6自举程序版本

自举程序版本号	说明	已知限制
V1.0	初始自举程序版本	对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号，而不是1个NACK信号。

8 STM32F030xC器件自举程序

8.1 自举程序配置

STM32F030xC自举程序通过应用模式2激活（参见表 2：自举程序激活模式）。表 15显示了该自举程序使用的硬件资源。

表15.系统存储器自举模式下STM32F030xC器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48 MHz，使用HSI 8 MHz为时钟源。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序 (PA2/PA3)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。
USART2自举程序 (PA14/PA15)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000001x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

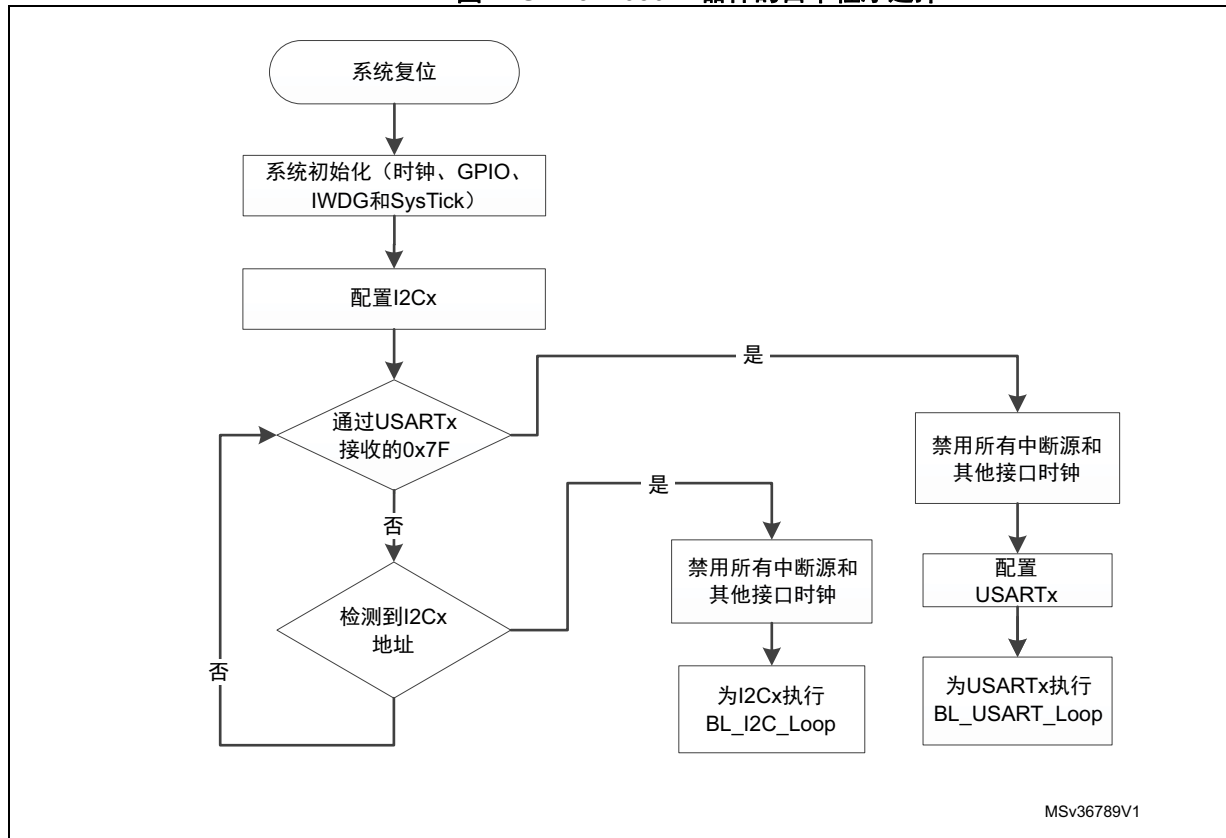
注: STM32F030xC器件在自举程序模式下使用USART2自举之后,系统发生复位之前,无法再实现串行线调试(SWD)通信,因为SWD使用的PA14引脚(SWCLK)已由自举程序(USART2_RX)使用。

系统时钟由嵌入式内部高速RC提供,自举程序执行不需要外部石英时钟。

8.2 自举程序选择

图 11显示了自举程序选择机制。

图11.STM32F030xC器件的自举程序选择



8.3 自举程序版本

表 16列出了STM32F030xC器件自举程序版本。

表16.STM32F030xC自举程序版本

自举程序版本号	说明	已知限制
V5.2	初始自举程序版本	PA13设置为输入上拉模式，即使不被自举程序使用

9 STM32F05xxx和STM32F030x8器件自举程序

9.1 自举程序配置

STM32F05xxx和STM32F030x8自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。表 17显示了该自举程序使用的硬件资源。

表17.系统存储器自举模式下STM32F05xxx和STM32F030x8器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于24 MHz（使用由HSI驱动 的PLL）。 1个Flash等待周期。
	RAM	-	自地址0x20000000起的2 KB空间供自举程 序固件使用。
	系统存储器	-	自地址0x1FFFE000起的3 KB空间包含自举 程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大 值，并定期进行刷新以防止看门狗复位（如 果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验 位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复 用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复 用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验 位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式。用于复 用推挽，上拉模式。
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式。用于复 用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

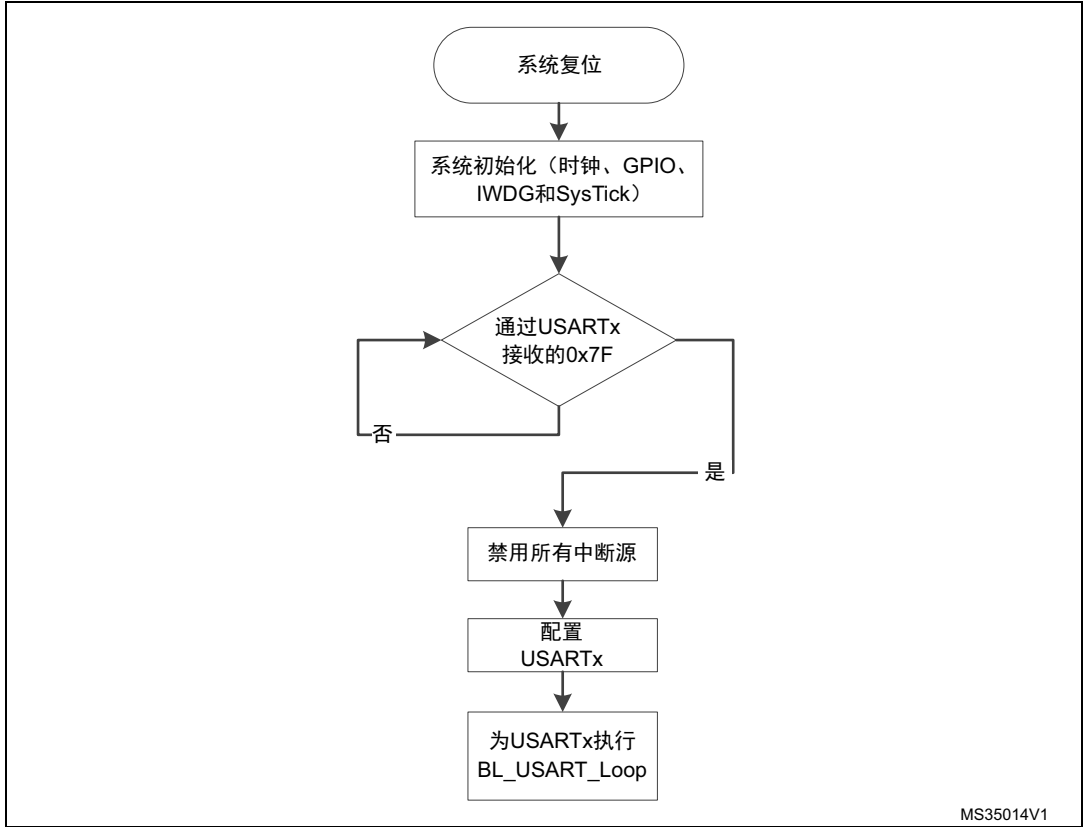
系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

注： STM32F05xxx和STM32F030x8器件在自举程序模式下自举之后，系统发生复位之前，无法再实现串行线调试（SWD）通信，因为SWD使用的PA14引脚（SWCLK）已由自举程序（USART2_TX）使用。

9.2 自举程序选择

图 12显示了自举程序选择机制。

图12.STM32F05xxx和STM32F030x8器件的自举程序选择



9.3 自举程序版本

表 18列出了STM32F05xxx和STM32F030x8器件自举程序版本。

表18.STM32F05xxx和STM32F030x8器件自举程序版本

自举程序版本号	说明	已知限制
V2.1	初始自举程序版本	<ul style="list-style-type: none"> - 在自举程序启动时，HSITRIM的值被置为0（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。为了获得更好结果，使用支持的最小晶振值（即4 MHz）。 - 对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号，而不是1个NACK信号。 - PA13设置为输入上拉模式，即使不被自举程序使用。

10 STM32F04xxx器件自举程序

10.1 自举程序配置

STM32F04xxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 19显示了该自举程序使用的硬件资源。

表19.系统存储器自举模式下STM32F04xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48MHz，使用HSI4848MHz为时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 48 MHz作为时钟。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFC400起的13KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111110x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

表19.系统存储器自举模式下STM32F04xxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线 无需外部上拉电阻。用于复用推挽，无上拉/下拉模式。

注: STM32F04xxx器件在自举程序模式下使用USART2自举之后，系统发生复位之前，无法再实现串行线调试（SWD）通信，因为SWD使用的PA14引脚（SWCLK）已由自举程序（USART2_RX）使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

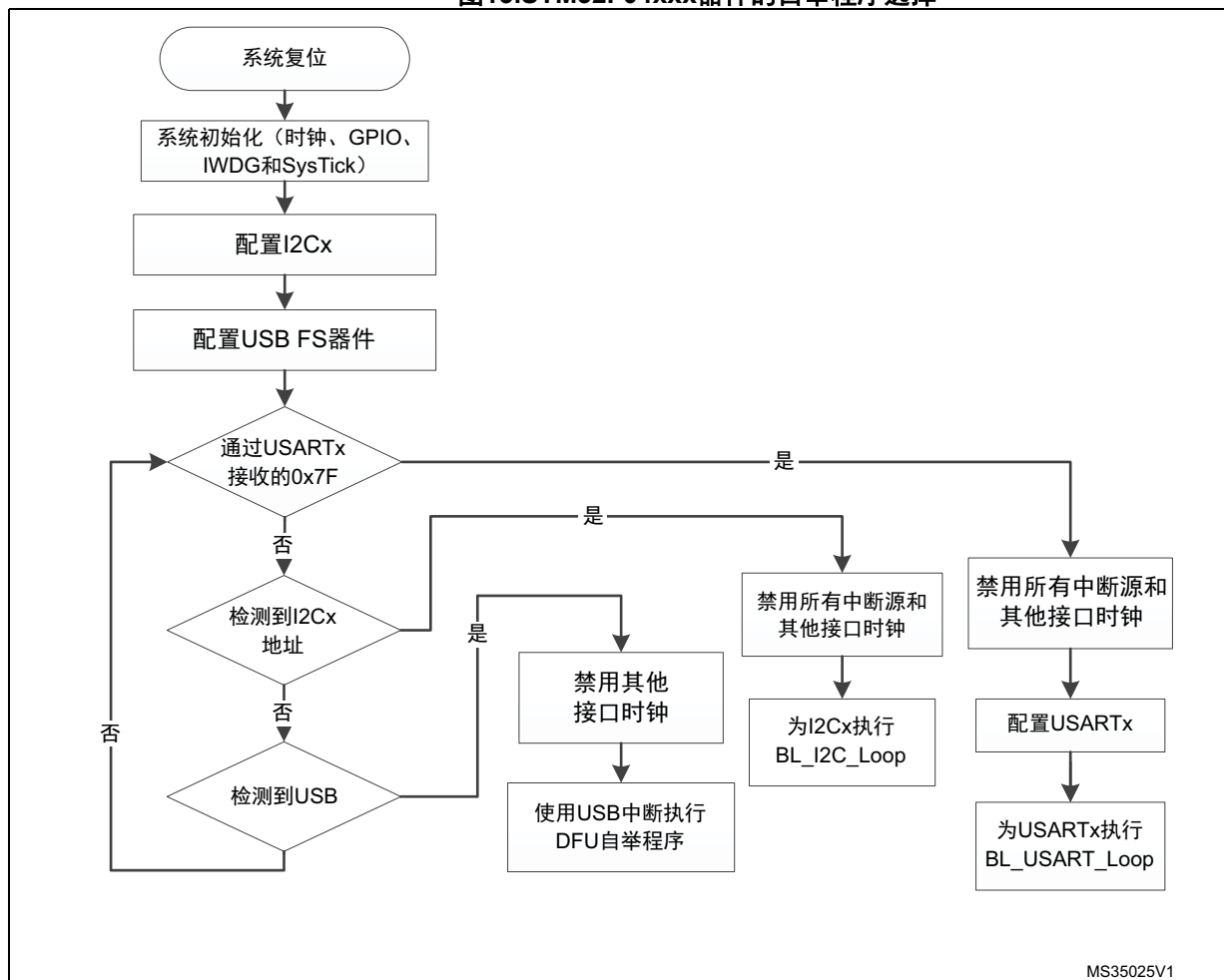
注: 由于该产品有空数据检查机制，不可能从用户代码跳转到系统自举程序。此类跳转导致跳回用户Flash存储器空间。

但是，如果跳转时用户Flash的前4个字节（地址0x0800 0000）为空（即在跳转前擦除第一个扇区或在Flash为空时执行SRAM中的代码），则在跳转后执行系统自举程序。

10.2 自举程序选择

图 13显示了自举程序选择机制。

图13.STM32F04xxx器件的自举程序选择



10.3 自举程序版本

下表列出了STM32F04xxx器件自举程序版本：

表20.STM32F04xxx自举程序版本

自举程序版本号	说明	已知限制
V10.0	初始自举程序版本	在自举程序启动时，HSITRIM的值被置为0（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。 为了获得更好结果，使用支持的最小晶振值（即4 MHz）。 PA13设置为输入上拉模式，即使不被自举程序使用。
V10.1	对于小封装，请在PA11/12 IO上加入对USART/USB接口的动态支持。	

11 STM32F070x6器件自举程序

11.1 自举程序配置

STM32F070x6自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 21显示了该自举程序使用的硬件资源。

表21.系统存储器自举模式下STM32F070x6器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	启动时，使用HSI将系统时钟频率配置为48 MHz。如果没有外部时钟（HSE），系统时钟将由HSI提供。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、8、6、4] MHz。PLL用来为USB和系统时钟产生48 MHz频率。
		-	为HSE使能时钟安全系统（Clock Security System, CSS）中断。外部时钟的任何故障（或移除）都会产生系统复位。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFC400起的13 KB空间包含自举程序固件。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0111110x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏模式下使用。
DFU自举程序	USB:	启用	USBFS配置为强制器件模式。使能USBFS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11引脚：USB FS DM线
	USB_DP引脚		PA12引脚：USB FS DP线 无需外部上拉电阻。

注：如果HSI误差超过了1%，则自举程序不能正常工作。

注: STM32F070x6器件在自举程序模式下使用USART2自举之后,系统发生复位之前,无法再实现串行线调试(SWD)通信,因为SWD使用的PA14引脚(SWCLK)已由自举程序(USART2_RX)使用。

自举程序具有两种操作情况,取决于自举程序启动时是否存在外部时钟(HSE):

- 如果存在HSE且其值为24、18、16、12、8、6或4MHz,则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1、USART2和I2C1功能正常,可用于与器件自举程序进行通信。
- 如果没有HSE,则使用HSI默认时钟源且只有USART1、USART2和I2C1功能正常。

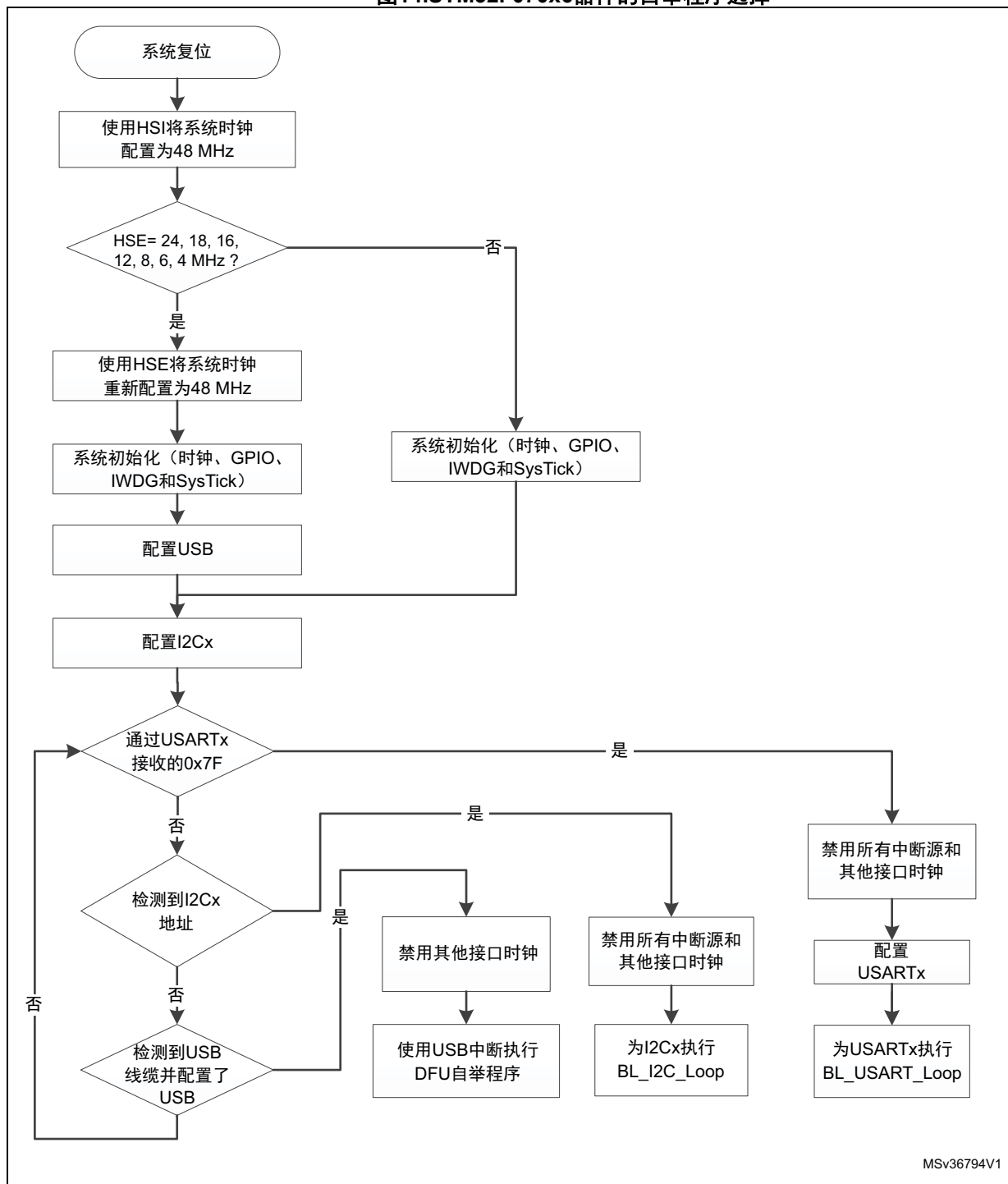
如果自举程序启动时连接了外部时钟(HSE),则它必须保持,因为它会用作系统时钟源。

注: 由于该产品有空数据检查机制,不可能从用户代码跳转到系统自举程序。此类跳转导致跳回用户Flash空间。但是,如果跳转时用户Flash的前4个字节(地址0x0800 0000)为空(即在跳转前擦除第一个扇区或在Flash为空时执行SRAM中的代码),则在跳转后执行系统自举程序。

11.2 自举程序选择

图 14显示了自举程序选择机制。

图14.STM32F070x6器件的自举程序选择



11.3 自举程序版本

表 22列出了STM32F070x6器件自举程序版本。

表22.STM32F070x6自举程序版本

自举程序版本号	说明	已知限制
V10.2	初始自举程序版本	在自举程序启动时，HSITRIM的值被置为0（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。为了获得更好结果，使用支持的最小晶振值（即4 MHz）。
V10.3	时钟配置固定为HSI 8 MHz	

12 STM32F070xB器件自举程序

12.1 自举程序配置

STM32F070xB自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。表 23显示了该自举程序使用的硬件资源。

表23.系统存储器自举模式下STM32F070xB器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	启动时，使用HSI将系统时钟频率配置为48MHz。如果没有外部时钟（HSE），系统时钟将由HSI提供。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、8、6、4]MHz。PLL用来为USB和系统时钟产生48 MHz频率。
		-	为HSE使能时钟安全系统（Clock Security System, CSS）中断。外部时钟的任何故障（或移除）都会产生系统复位。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFC800起的12 KB空间包含自举程序固件。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入上拉模式
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0111011x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

表23.系统存储器自举模式下STM32F070xB器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。使能USB FS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11引脚：USBFS DM线用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USBFS DP线用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。

注： 如果HSI误差超过了1%，则自举程序不能正常工作。

注： STM32F070xB器件在自举程序模式下使用USART2自举之后，系统发生复位之前，无法再实现串行线调试（SWD）通信，因为SWD使用的PA14引脚（SWCLK）已由自举程序（USART2_RX）使用。

自举程序具有两种操作情况，取决于自举程序启动时是否存在外部时钟（HSE）：

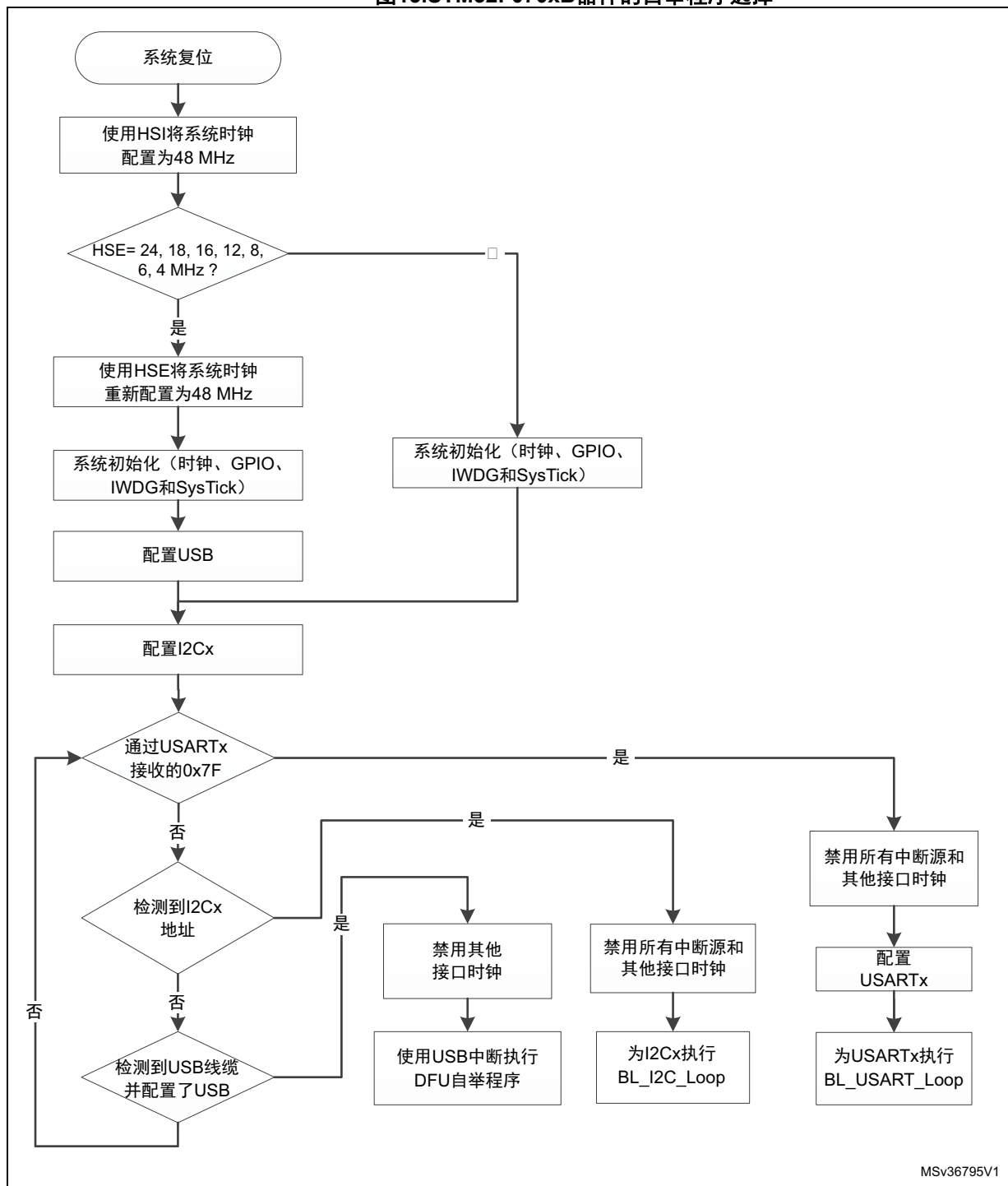
- 如果存在HSE且其值为24、18、16、12、8、6或4MHz，则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1、USART2和I2C1功能正常，可用于与器件自举程序进行通信。
- 如果没有HSE，则使用HSI默认时钟源且只有USART1、USART2和I2C1功能正常。

如果自举程序启动时连接了外部时钟（HSE），则它必须保持，因为它会用作系统时钟源。

12.2 自举程序选择

图 15显示了自举程序选择机制。

图15.STM32F070xB器件的自举程序选择



12.3 自举程序版本

表 24列出了STM32F070xB器件自举程序版本。

表24.STM32F070xB自举程序版本

自举程序版本号	说明	已知限制
V10.2	初始自举程序版本	在自举程序启动时，HSITRIM的值被置为（0）（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。为了获得更好结果，使用支持的最小晶振值（即4 MHz）。PA13设置为复用推挽模式，即使不被自举程序使用。
V10.3	时钟配置固定为HSI 8 MHz	

13 STM32F071xx/072xx器件自举程序

13.1 自举程序配置

STM32F071xx/072xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。表 25显示了该自举程序使用的硬件资源。

表25.系统存储器自举模式下STM32F071xx/072xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48MHz，使用HSI4848MHz为时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 48 MHz作为时钟。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFC800起的12KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA15引脚：USART2处于接收模式。用于输入上拉模式
	USART2_TX引脚	输出	PA14引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111011x（其中x = 0对应写，x = 1对应读）。
		I2C1_SCL引脚	输入/输出
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。

表25.系统存储器自举模式下STM32F071xx/072xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线 无需外部上拉电阻。用于复用推挽，无上拉/下拉模式。

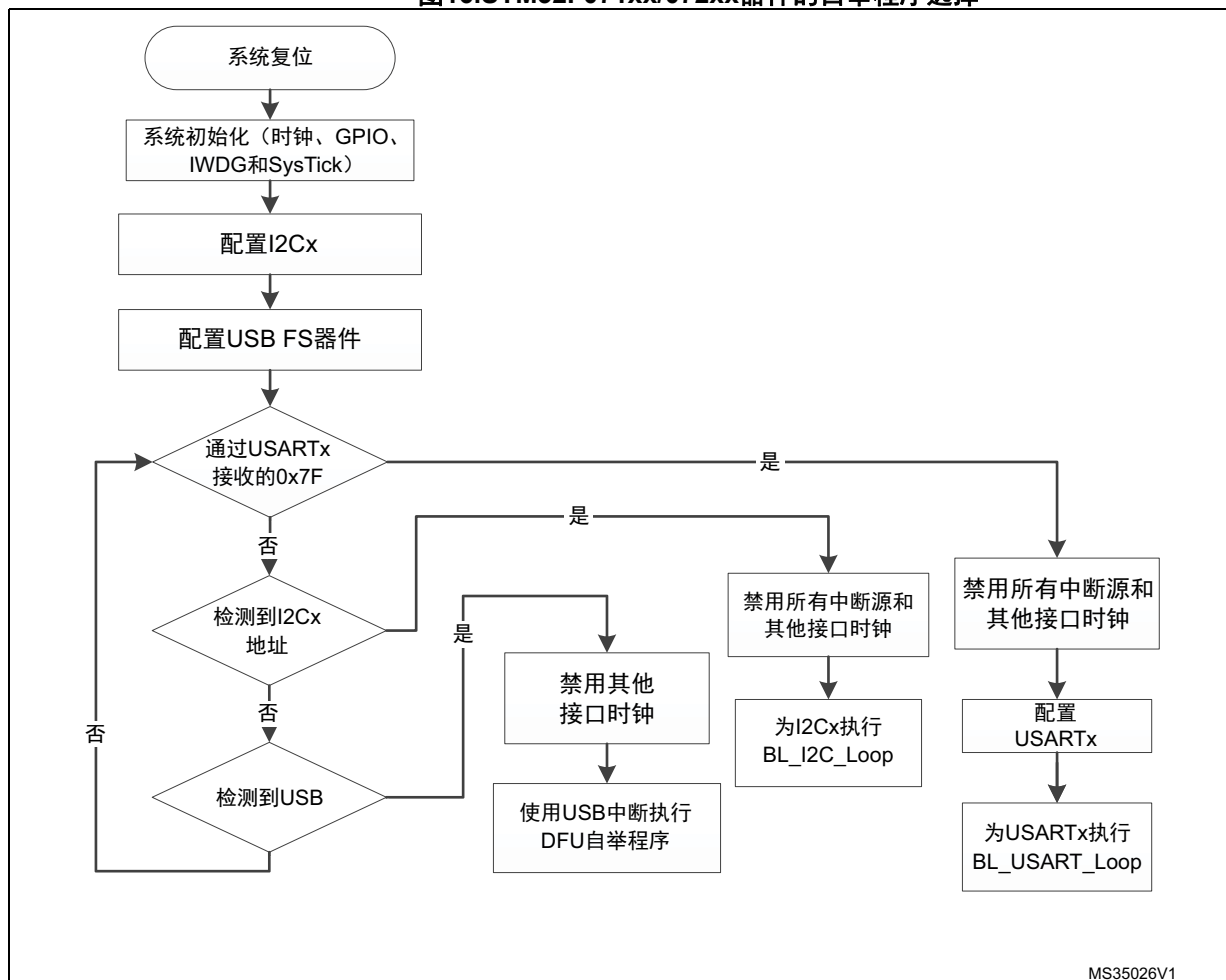
注: STM32F071xx/072xx 器件在自举程序模式下使用USART2自举之后，系统发生复位之前，无法再实现串行线调试（SWD）通信，因为SWD使用的PA14引脚（SWCLK）已由自举程序（USART2_RX）使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

13.2 自举程序选择

图 16显示了自举程序选择机制。

图16.STM32F071xx/072xx器件的自举程序选择



13.3 自举程序版本

表 26列出了STM32F071xx/072xx器件自举程序版本。

表26.STM32F071xx/072xx自举程序版本

自举程序版本号	说明	已知限制
V10.1	初始自举程序版本	在自举程序启动时，HSITRIM的值被置为（0）（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。为了获得更好结果，使用支持的最小晶振值（即4 MHz）。PA13置为复用推挽，上拉模式，即使不被自举程序使用。

14 STM32F09xxx器件自举程序

14.1 自举程序配置

STM32F09xxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表 27.系统存储器自举模式下STM32F09xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48MHz，使用HSI4848MHz为时钟源。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
			PA15引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
PA14引脚：USART2处于发送模式。用于复用推挽，上拉模式。			
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000001x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

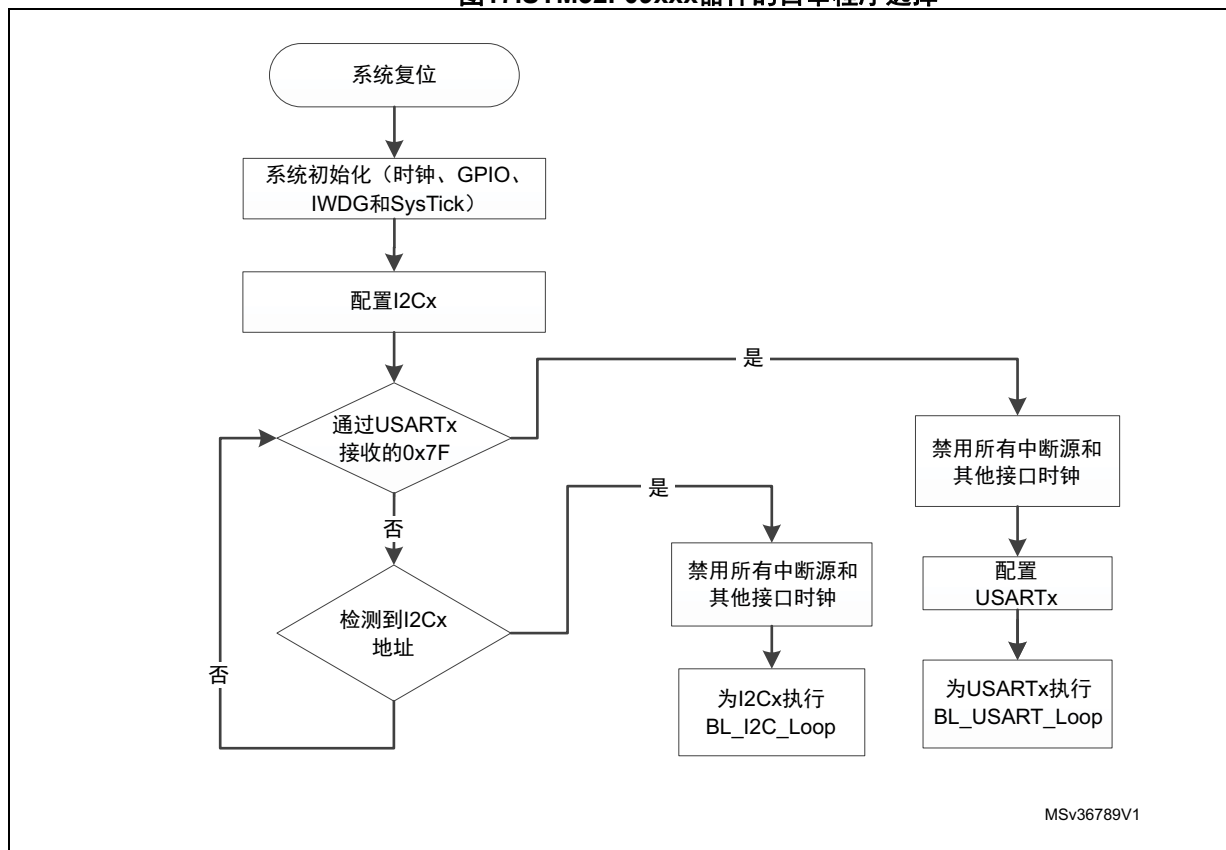
注: *STM32F09xxx* 器件在自举程序模式下使用USART2自举之后, 系统发生复位之前, 无法再实现串行线调试 (SWD) 通信, 因为SWD使用的PA14引脚 (SWCLK) 已由自举程序 (USART2_RX) 使用。

系统时钟由嵌入式内部高速RC提供, 自举程序执行不需要外部石英时钟。

14.2 自举程序选择

下图显示了自举程序选择机制。

图17.STM32F09xxx器件的自举程序选择



14.3 自举程序版本

下表列出了STM32F09xxx器件自举程序版本。

表28.STM32F09xxx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	在自举程序启动时，HSITRIM的值被置为（0）（在RCC_CR寄存器上的HSITRIM位），而不是默认值（16），因此，晶振测量会产生偏差。为了获得更好结果，使用支持的最小晶振值（即4 MHz）。PA13设置为输入上拉模式，即使不被自举程序使用。

15 STM32F10xxx器件自举程序

15.1 自举程序配置

STM32F10xxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。表 29显示了该自举程序使用的硬件资源。

表29.系统存储器自举模式下STM32F10xxx器件的配置

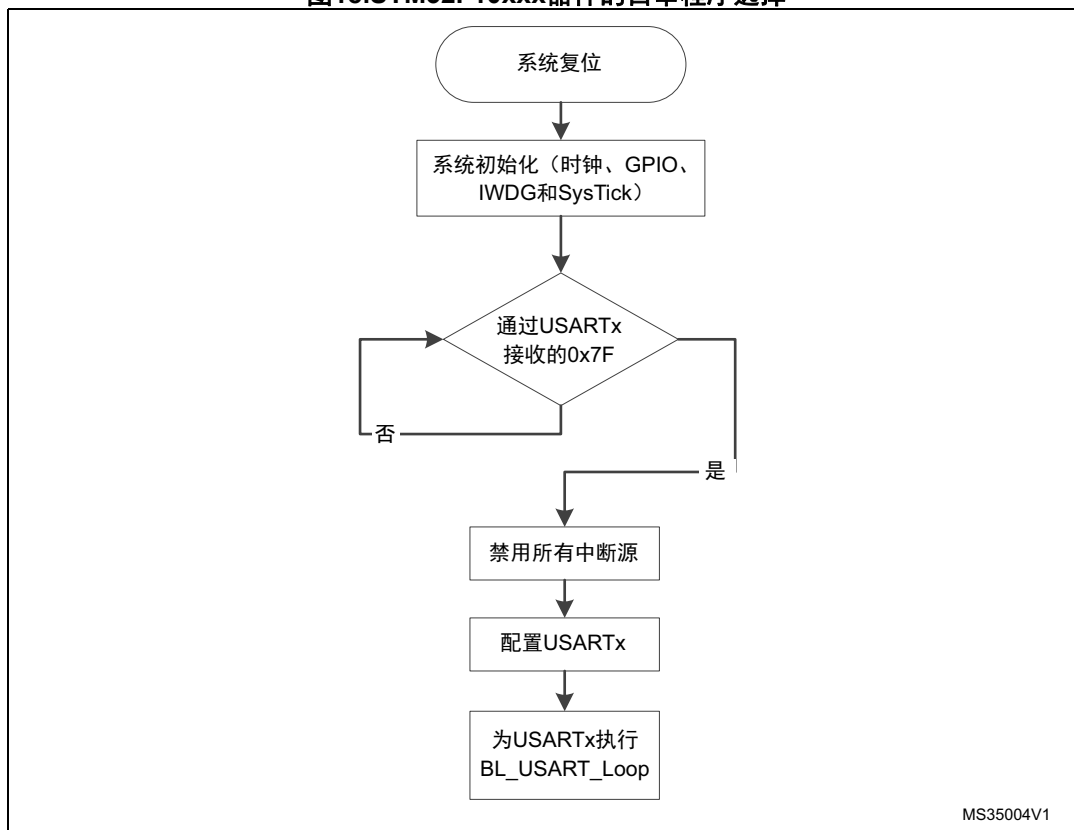
自举程序	功能/外设	状态	备注
USART1自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。
	RAM	-	自地址0x20000000起的512 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFF000起的2 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	推挽输出	PA9引脚：USART1处于发送模式。用于复用推挽上拉模式。
	SysTick定时器	启用	用于自动检测主机串口波特率。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

15.2 自举程序选择

下图显示了自举程序选择机制。

图18.STM32F10xxx器件的自举程序选择



15.3 自举程序版本

表 30列出了STM32F10xxx器件自举程序版本。

表30.STM32F10xxx自举程序版本

自举程序版本号	说明
V2.0	初始自举程序版本
V2.1	<ul style="list-style-type: none"> - 更新Go命令以初始化主栈指针 - 更新Go命令，以在跳转地址位于选项字节区域或系统存储区时返回NACK - 更新Get ID命令以返回两个字节的器件ID - 将自举程序版本更新为V2.1
V2.2	<ul style="list-style-type: none"> - 更新Read Memory、Write Memory和Go命令，以通过NACK响应拒绝对自举程序所用的RAM存储器前0x200字节的访问 - 更新Readout Unprotect命令，以在禁止ROP之前将整个RAM内容初始化为0x0

注: *自举程序ID格式适用于除STM32F1xx系列之外的所有STM32器件系列。STM32F1xx的自举程序版本只适用于嵌入式器件的自举程序版本, 不适用于其支持的协议。*

16 STM32F105xx/107xx器件自举程序

16.1 自举程序配置

STM32F105xx/107xx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表31.系统存储器自举模式下STM32F105xx/107xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。仅用于USARTx自举程序，以及在CAN2、USB检测期间用于CAN和DFU自举程序（选择CAN或DFU自举程序后，时钟源由外部石英时钟提供）。
		HSE使能	只有DFU和CAN自举程序必须使用外部时钟，外部时钟必须提供以下其中一种频率：8 MHz、14.7456 MHz或25 MHz。对于CAN自举程序，使用14.7456 MHz作为HSE时，PLL仅用于生成48 MHz的频率。对于DFU自举程序，PLL则用于从所有支持的外部时钟频率生成48 MHz的系统时钟。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFB000起的18 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	推挽输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2用于接收（重映射引脚）
	USART2_TX引脚	推挽输出	PD5引脚：USART2用于发送（重映射引脚）
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表31.系统存储器自举模式下STM32F105xx/107xx器件的配置（续）

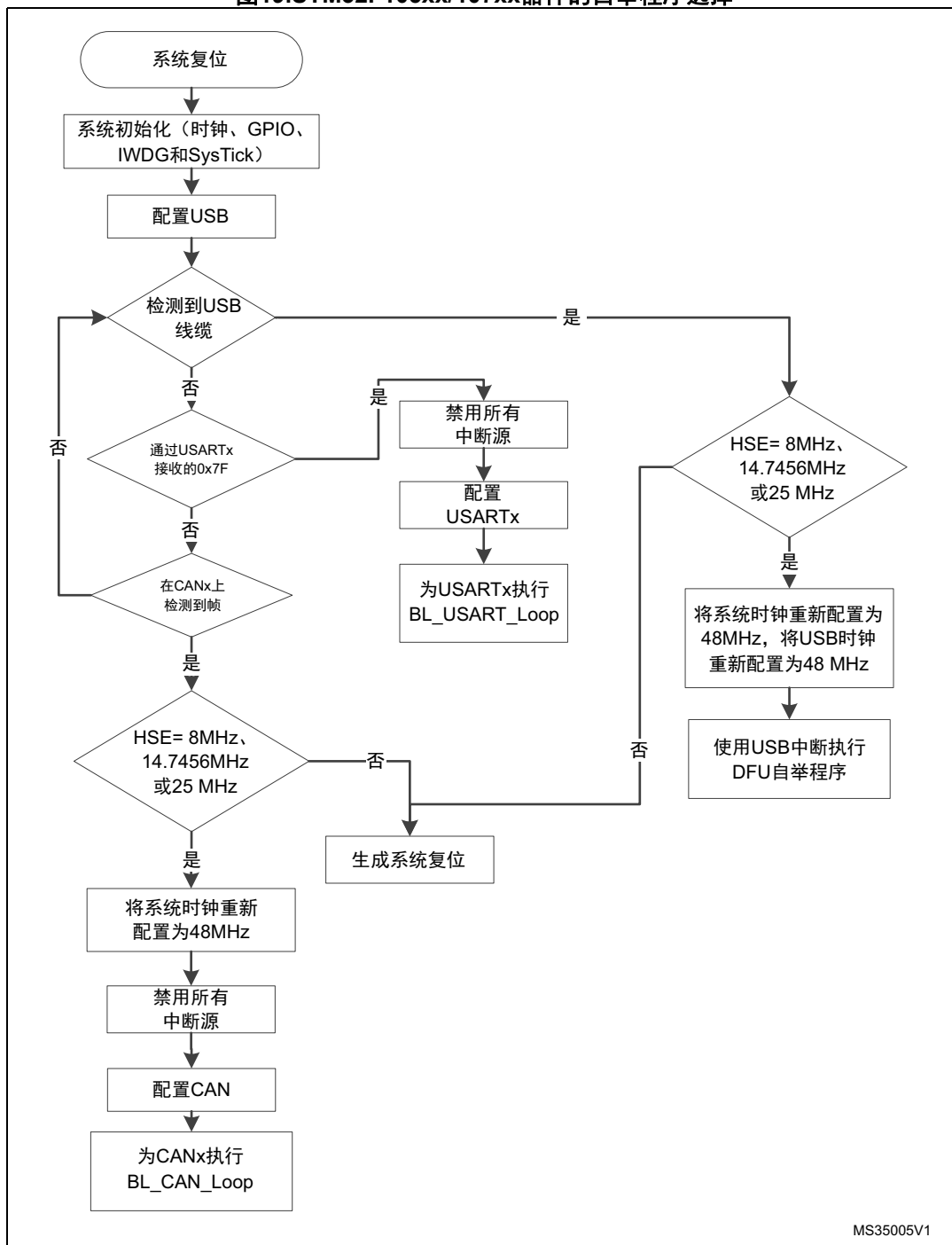
自举程序	功能/外设	状态	备注
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2用于接收（重映射引脚）。用于复用推挽上拉模式。
	CAN2_TX引脚	推挽输出	PB6引脚：CAN2用于发送（重映射引脚）。用于输入无上拉/下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_VBUS引脚	输入	PA9：电源电压线
	USB_DM引脚	输入/输出	PA11引脚：USB_DM线
	USB_DP引脚		PA12引脚：USB_DP线。 无需外部上拉电阻

对于USARTx自举程序，系统时钟由内部高速RC提供。该内部时钟同样用于DFU和CAN自举程序，但仅限选择阶段。选择阶段结束后，执行DFU和CAN自举程序需要使用外部时钟（8 MHz、14.7456 MHz或25 MHz）。

16.2 自举程序选择

图 19显示了自举程序选择机制。

图19.STM32F105xx/107xx器件的自举程序选择



16.3 自举程序版本

下表列出了STM32F105xx/107xx器件自举程序版本。

表32.STM32F105xx/107xx自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本
V2.0	<ul style="list-style-type: none"> - 更新自举程序检测机制，以修正检测阶段期间此自举程序未用外设的GPIO连接到低电平或悬空的问题。 有关详细信息，请参见 第 16.3.2 节 - 向量表设置为0x1FFFB000而不是0x00000000 - 更新Go命令（针对所有自举程序）：USART1、USART2、CAN2、GPIOA、GPIOB、GPIOD和SysTick外设寄存器设置为默认复位值 - DFU自举程序：执行Leave DFU命令前将USB中断挂起清零 - DFU子协议版本由V1.0变为V1.2 - 自举程序版本更新为V2.0
V2.1	<ul style="list-style-type: none"> - 修正第 第 16.3.4 节 中所述的PA9过大功耗问题。 - 更正Get-Version命令（如AN3155中定义）。该命令将返回0x22，而不是自举程序V2.0中的0x20。更多详细信息，请参见 第 16.3.3 节。 - 自举程序版本更新为V2.1
V2.2	<ul style="list-style-type: none"> - 修正DFU选项字节描述符（由于可读/写但不可擦除，因此设置为“e”而不是“g”）。 - 修正用于Flash读/写/擦除操作的DFU轮询时序。 - 提高DFU自举程序接口的耐用性。 - 自举程序版本更新为V2.2。

注： 自举程序ID格式适用于除STM32F1xx系列之外的所有STM32器件系列。STM32F1xx的自举程序版本只适用于嵌入式器件的自举程序版本，不适用于其支持的协议。

16.3.1 如何识别STM32F105xx/107xx自举程序版本

自举程序V1.0用于日期代码小于937的器件（有关如何查找器件标记上的日期代码，请参见STM32F105xx和STM32F107xx数据手册）。

自举程序V2.0和V2.1用于日期代码大于或等于937的器件。

自举程序V2.2用于日期代码大于或等于227的器件。

自举程序版本有两种区分方法：

- 使用USART自举程序时，V2.1版本中更正了AN2606和AN3155中定义的Get-Version命令。该命令将返回0x22，而不是自举程序V2.0中的0x20。

- 自举程序代码开始位置的向量表中的值有所不同。对于自举程序V2.0、V2.1和V2.2，用户软件（或通过JTAG/SWD）在地址0x1FFFB004处读取的分别是0x1FFFE945、0x1FFFE9A1和0x1FFFE9C1。
- DFU版本如下：
 - 自举程序V2.1中为V2.1版
 - 自举程序V2.2中为V2.2版可从DFU器件描述符的bcdDevice字段中读取该版本。

16.3.2 当日期代码小于937时，STM32F105xx/STM32F107xx器件的自举程序不可用

说明

激活自举程序阶段，如果USART1_RX（PA10）、USART2_RX（PD6，重映射）、CAN2_Rx（PB5，重映射）、OTG_FS_DM（PA11）和/或OTG_FS_DP（PA12）引脚保持低电平或悬空，则自举程序不可用。

自举程序无法通过CAN2（重映射）、DFU（器件模式）或STG（重映射）进行连接。

在64引脚封装中，USART2_RX信号重映射引脚PD6不可用，而是在内部接地。这种情况下，自举程序完全不可用。

解决方案

- 对于64引脚封装
无。自举程序不可用。
- 对于100引脚封装
根据所用外设的不同，必须按照下述要求在激活自举程序阶段使未使用的外设引脚保持高电平：
 - 如果使用USART1连接自举程序，则PD6和PB5必须保持高电平。
 - 如果使用USART2连接自举程序，则PA10、PB5、PA11和PA12必须保持高电平。
 - 如果使用CAN2连接自举程序，则PA10、PD6、PA11和PA12必须保持高电平。
 - 如果使用DFU连接自举程序，则PA10、PB5和PD6必须保持高电平。

注： 此限制仅适用于日期代码小于937的STM32F105xx和STM32F107xx器件。日期代码大于或等于937的STM32F105xx和STM32F107xx器件并不受此影响。有关如何查找器件标记上的日期代码，请参见STM32F105xx和STM32F107xx数据手册。

16.3.3 USART自举程序Get-Version命令返回0x20而不是0x22

说明

在USART模式下，Get-Version命令（如AN3155中定义）返回0x20而不是0x22。此限制存在于自举程序版本V1.0和V2.0中，自举程序版本2.1对此进行了修正。

解决方案

无。

16.3.4 插入USB线缆时，PA9功耗过大（在自举程序V2.0中）

说明

从系统存储器模式自举后连接USB电缆时，PA9引脚（连接 $V_{BUS}=5\text{ V}$ ）同时与配置为复用推挽的USARTTX引脚共用，由于尚未对USART外设提供时钟，因此该引脚强制设置为0。因而流经PA9 I/O的漏电流超出25 mA，可能影响I/O板的可靠性。

自举程序版本2.1对此限制进行了修正，实现方法为：在RX引脚接收到正确的0x7F且对USART提供时钟时，将PA9配置为复用功能推挽。否则将PA9配置为复用输入悬空。

解决方案

无。

17 STM32F10xxx超大容量器件自举程序

17.1 自举程序配置

STM32F10xxx超大容量自举程序通过应用模式3激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表33.系统存储器自举模式下STM32F10xxx超大容量器件的配置

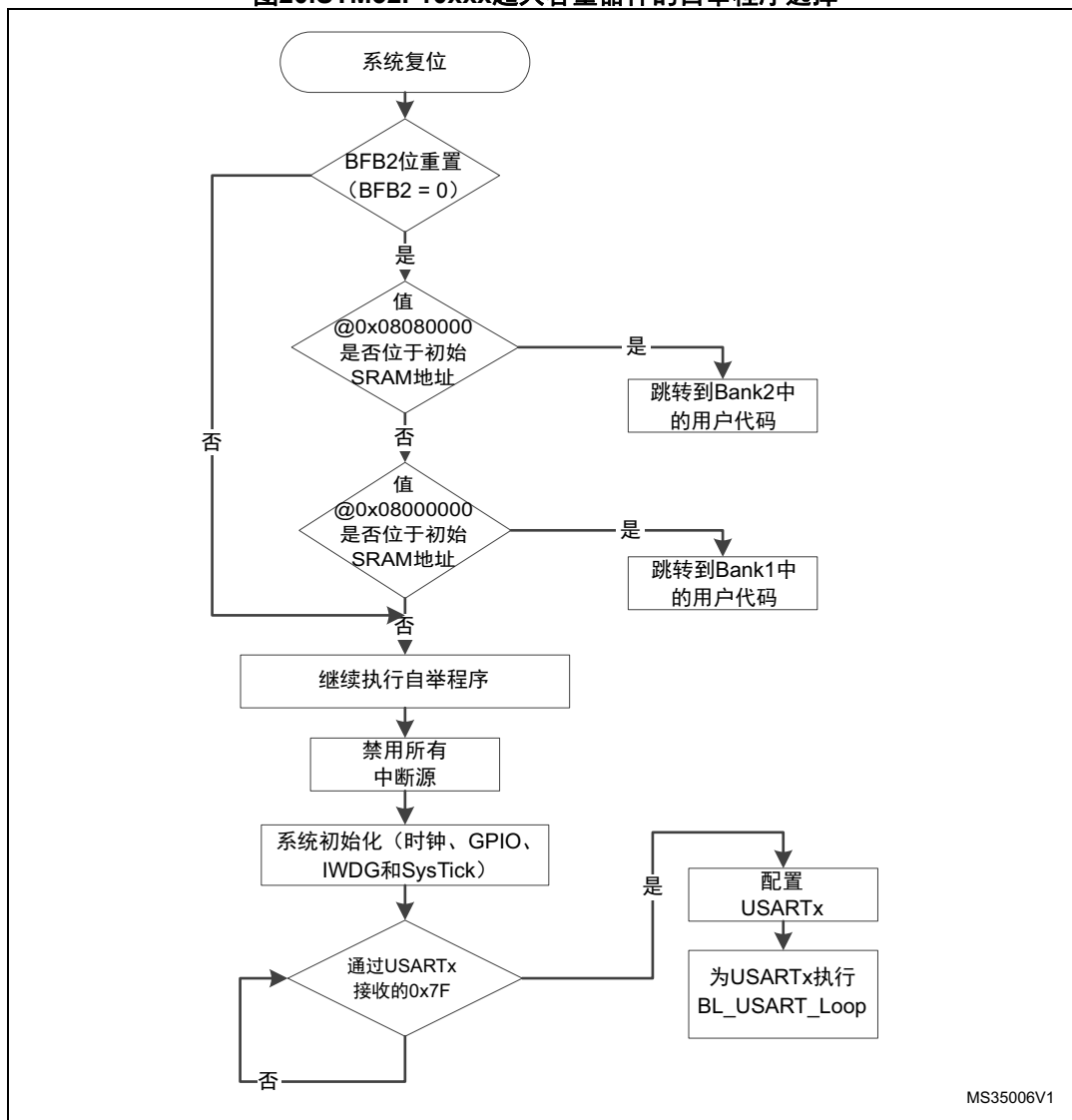
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。
	RAM	-	自地址0x20000000起的2 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFE000起的6 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	推挽输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2用于接收（重映射引脚）。用于输入上拉模式。
	USART2_TX引脚	推挽输出	PD5引脚：USART2用于发送（重映射引脚）。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

17.2 自举程序选择

图 20显示了自举程序选择机制。

图20.STM32F10xxx超大容量器件的自举程序选择



MS35006V1

17.3 自举程序版本

表 34列出了STM32F10xxx超大容量器件自举程序版本。

表34.STM32F10xxx超大容量自举程序版本

自举程序版本号	说明
V2.1	初始自举程序版本

注: *自举程序ID格式适用于除STM32F1xx系列之外的所有STM32器件系列。STM32F1xx的自举程序版本只适用于嵌入式器件的自举程序版本, 不适用于其支持的协议。*

18 STM32F2xxx器件自举程序

STM32F2xxx器件可使用两种自举程序版本：

- V2.x支持USART1和USART3
修订版本A、Z和B中嵌入了此版本
- V3.x支持USART1、USART3、CAN2和DFU（USB FS设备）
所有其他修订版本（Y、X、W、1、V、2、3和4）中均嵌入了此版本。

18.1 自举程序V2.x

18.1.1 自举程序配置

STM32F2xxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表35.系统存储器自举模式下STM32F2xxx器件的配置

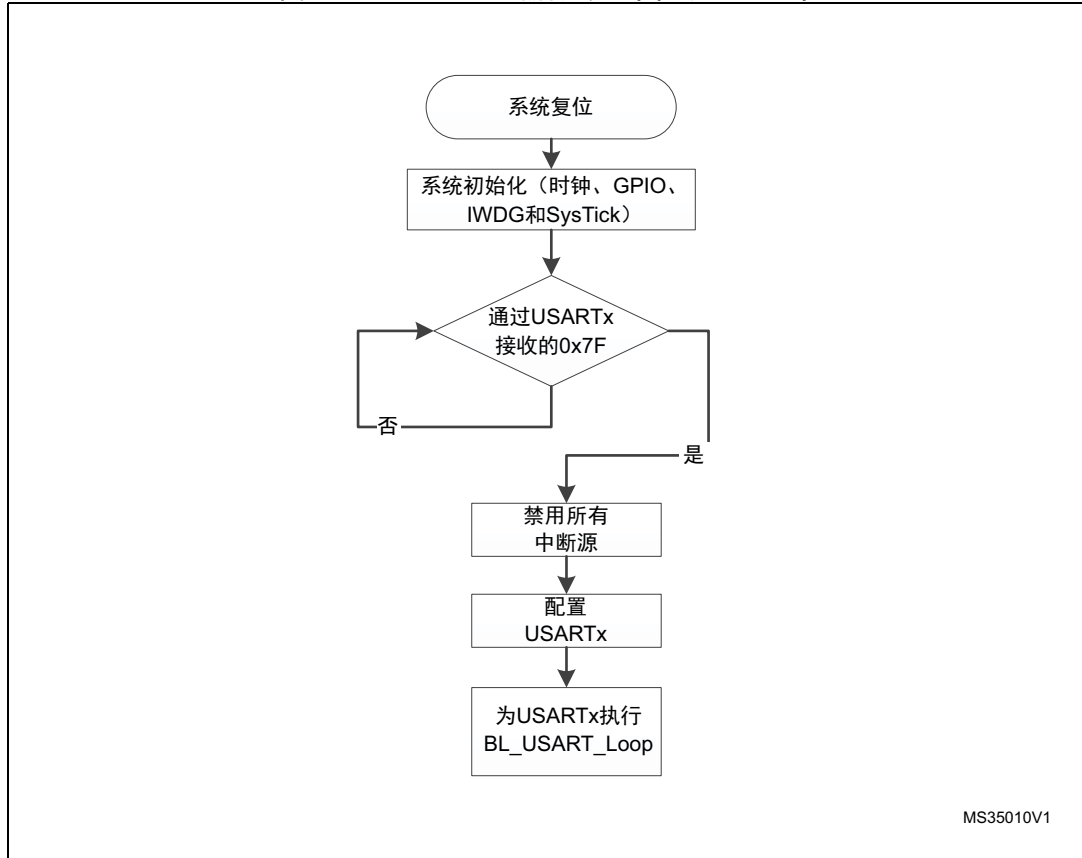
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为24 MHz。
	RAM	-	自地址0x20000000起的8 KB空间。
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式
USARTx自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

系统时钟由嵌入式内部高速RC提供，自举程序代码不需要外部石英时钟。

18.1.2 自举程序选择

图 21显示了自举程序选择机制。

图21.STM32F2xxx器件的自举程序V2.x选择



18.1.3 自举程序版本

下表列出了STM32F10xxx器件V2.x自举程序版本：

表36.STM32F2xxx自举程序V2.x版本

自举程序版本号	说明	已知限制
V2.0	初始自举程序版本	当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址0x6000 0000）发出Read Memory命令或Write Memory命令时，自举程序器件会中止这一命令，但不会向主机发送NACK (0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和。对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。 ⁽¹⁾

1. 如果待读/写的“数据数 - 1” (N-1) 不等于有效命令代码 (0x00、0x01、0x02、0x11、0x21、0x31、0x43、0x44、0x63、0x73、0x82或0x92)，则无法从主机发现缺陷，因为该命令始终会收到NACK应答（当作不受支持的新命令）。

18.2 自举程序V3.x

18.2.1 自举程序配置

STM32F2xxxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表37.系统存储器自举模式下STM32F2xxxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USARTx接口期间（选择CAN或DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择CAN或DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的8 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的29 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于上拉模式
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于上拉模式

表37.系统存储器自举模式下STM32F2xxx器件的配置（续）

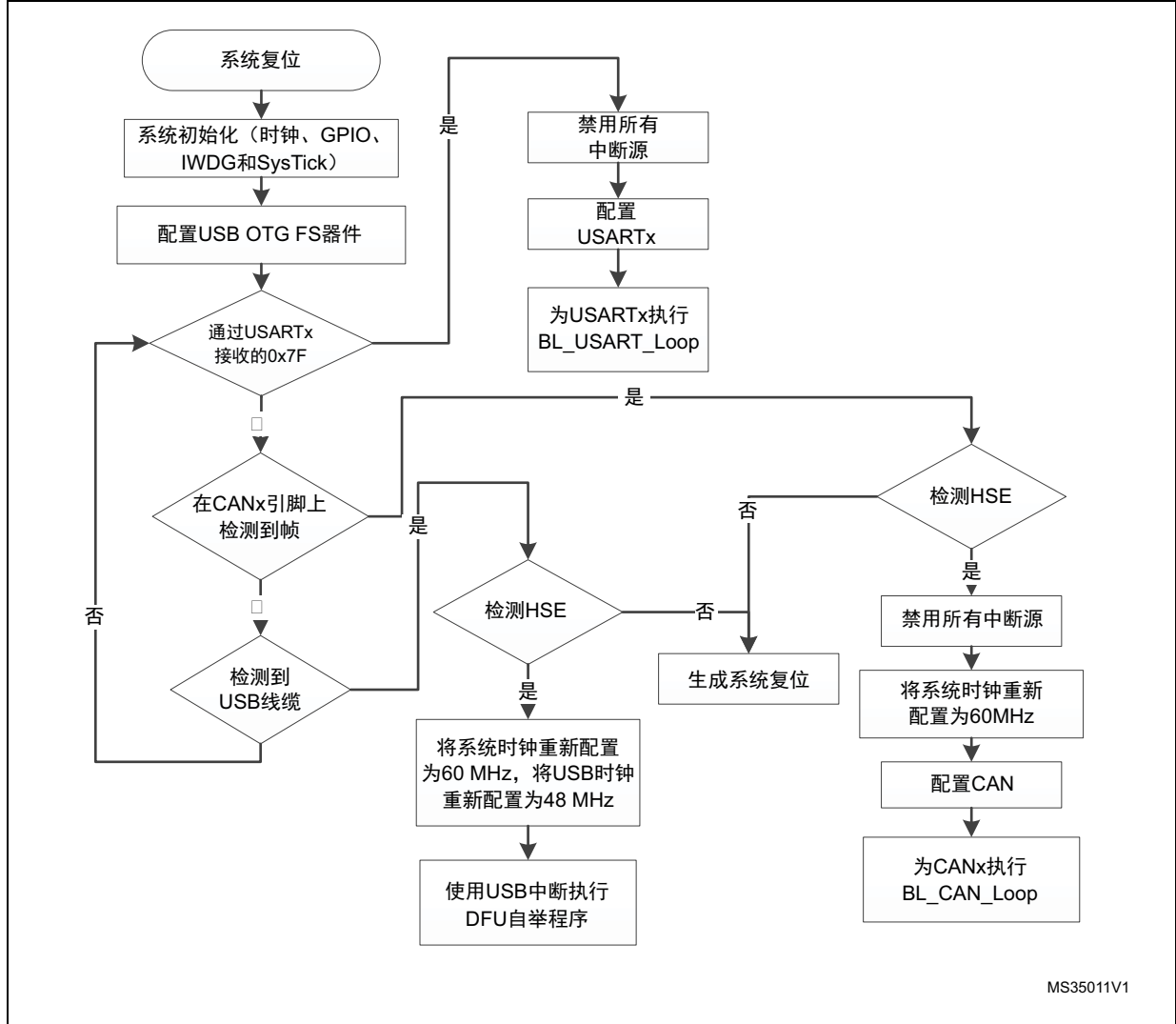
自举程序	功能/外设	状态	备注
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USBDM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

对于USARTx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

18.2.2 自举程序选择

下图显示了自举程序选择机制。

图22.STM32F2xxxx器件的自举程序V3.x选择



18.2.3 自举程序版本

下表列出了STM32F2xxx器件V3.x自举程序版本：

表38.STM32F2xxx自举程序V3.x版本

自举程序版本号	说明	已知限制
V3.2	初始自举程序版本。	<ul style="list-style-type: none"> – 当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址0x6000 0000）发出Read Memory命令或Write Memory命令时，自举程序器件会中止这一命令，但不会向主机发送NACK (0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和⁽¹⁾。 – 选项字节、OTP和设备功能描述符（位于DFU接口中）设置为“g”而不是“e”（不是可擦除存储区）。
V3.3	修正V3.2限制。提高DFU接口耐用性。	<ul style="list-style-type: none"> – 对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号（而不是1个NACK信号）。 – 对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。

1. 如果待读/写的“数据数 - 1” (N-1) 不等于有效命令代码 (0x00、0x01、0x02、0x11、0x21、0x31、0x43、0x44、0x63、0x73、0x82或0x92)，则无法从主机发现缺陷，因为该命令始终会收到NACK应答（当作不受支持的新命令）。

19 STM32F301xx/302x4 (6/8) 器件自举程序

19.1 自举程序配置

STM32F301xx/302x4 (6/8) 自举程序通过应用模式2激活（详述参见 [表 2: 自举程序激活模式](#)）。下表介绍了该自举程序使用的硬件资源。

表39.系统存储器自举模式下STM32F301xx/302x4 (6/8) 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48MHz，使用HSI4848MHz为时钟源。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、9、8、6、4、3] MHz。 PLL用于生成USB48MHz时钟和48MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表39.系统存储器自举模式下STM32F301xx/302x4 (6/8) 器件的配置 (续)

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽, 无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽, 无上拉/下拉模式。 必须连接一个1.5 KΩ的外部上拉电阻到USB_DP引脚。

自举程序具有两种操作情况, 取决于自举程序启动时是否存在外部时钟 (HSE) :

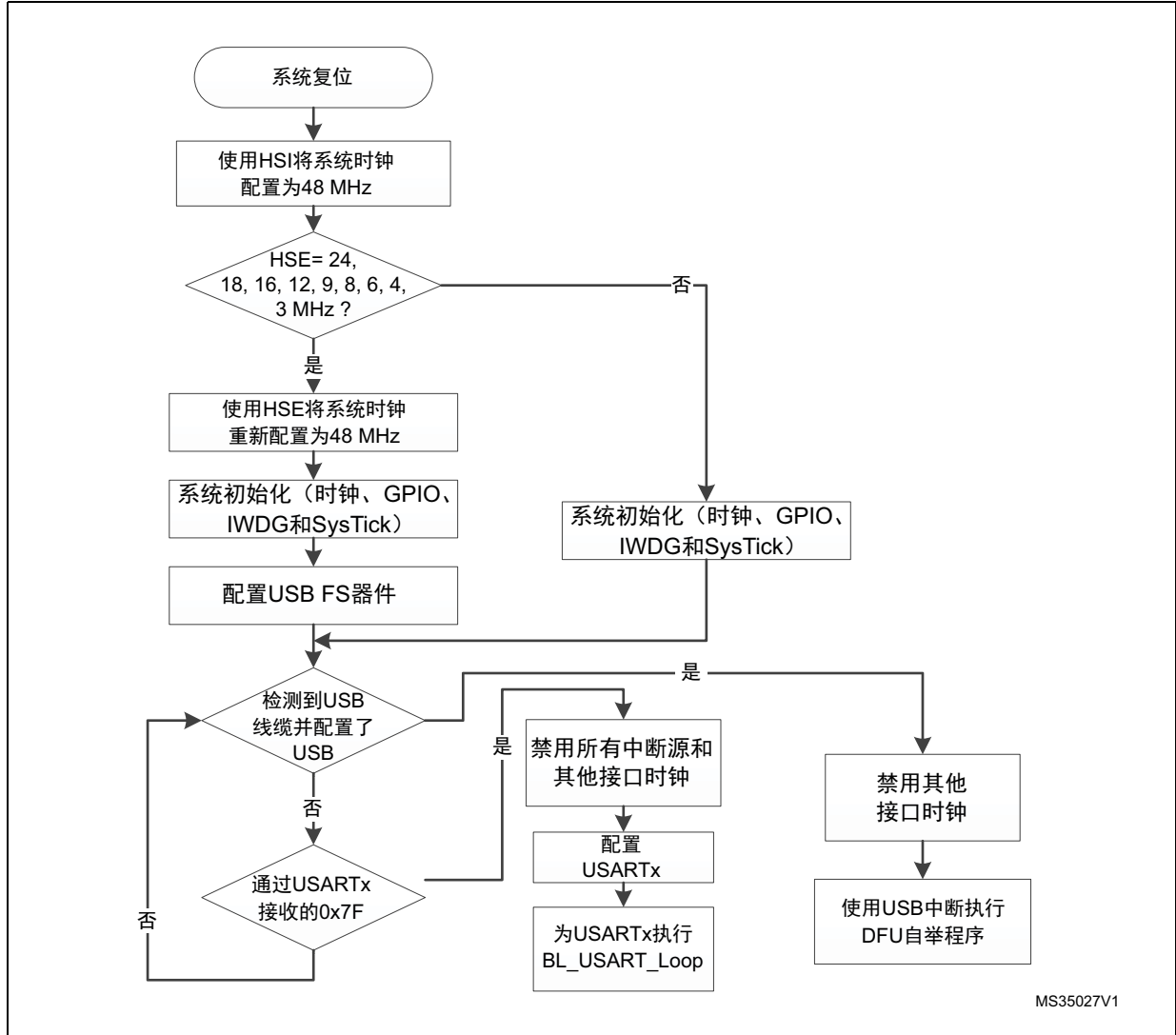
- 如果存在HSE且其值为24、18、16、12、9、8、6、4或3 MHz, 则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1和USART2功能正常, 可用于与器件自举程序进行通信。
- 如果没有HSE, 则使用HSI默认时钟源且只有USART1和USART2功能正常。

如果自举程序启动时连接了外部时钟 (HSE), 则它必须保持, 因为它会用作系统时钟源。

19.2 自举程序选择

图 23显示了自举程序选择机制。

图23.STM32F301xx/302x4 (6/8) 的自举程序选择



MS35027V1

19.3 自举程序版本

下表列出了STM32F301xx/302x4 (6/8) 器件自举程序版本：

表40.STM32F301xx/302x4 (6/8) 自举程序版本

自举程序版本号	说明	已知限制
V4.0	初始自举程序版本	无

20 STM32F302xB (C) /303xB (C) 器件自举程序

20.1 自举程序配置

STM32F302xB (C) /303xB (C) 自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表41.系统存储器自举模式下STM32F302xB (C) /303xB (C) 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	启动时，使用HSI将系统时钟频率配置为48MHz。如果没有外部时钟（HSE），系统时钟将由HSI提供。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、9、8、6、4、3] MHz。 PLL用于生成48MHz USB时钟和48MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的5KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表41.系统存储器自举模式下STM32F302xB (C) /303xB (C) 器件的配置 (续)

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽, 无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽, 无上拉/下拉模式。 必须连接一个1.5 KΩ的外部上拉电阻到USB_DP引脚。

自举程序具有两种操作情况, 取决于自举程序启动时是否存在外部时钟 (HSE) :

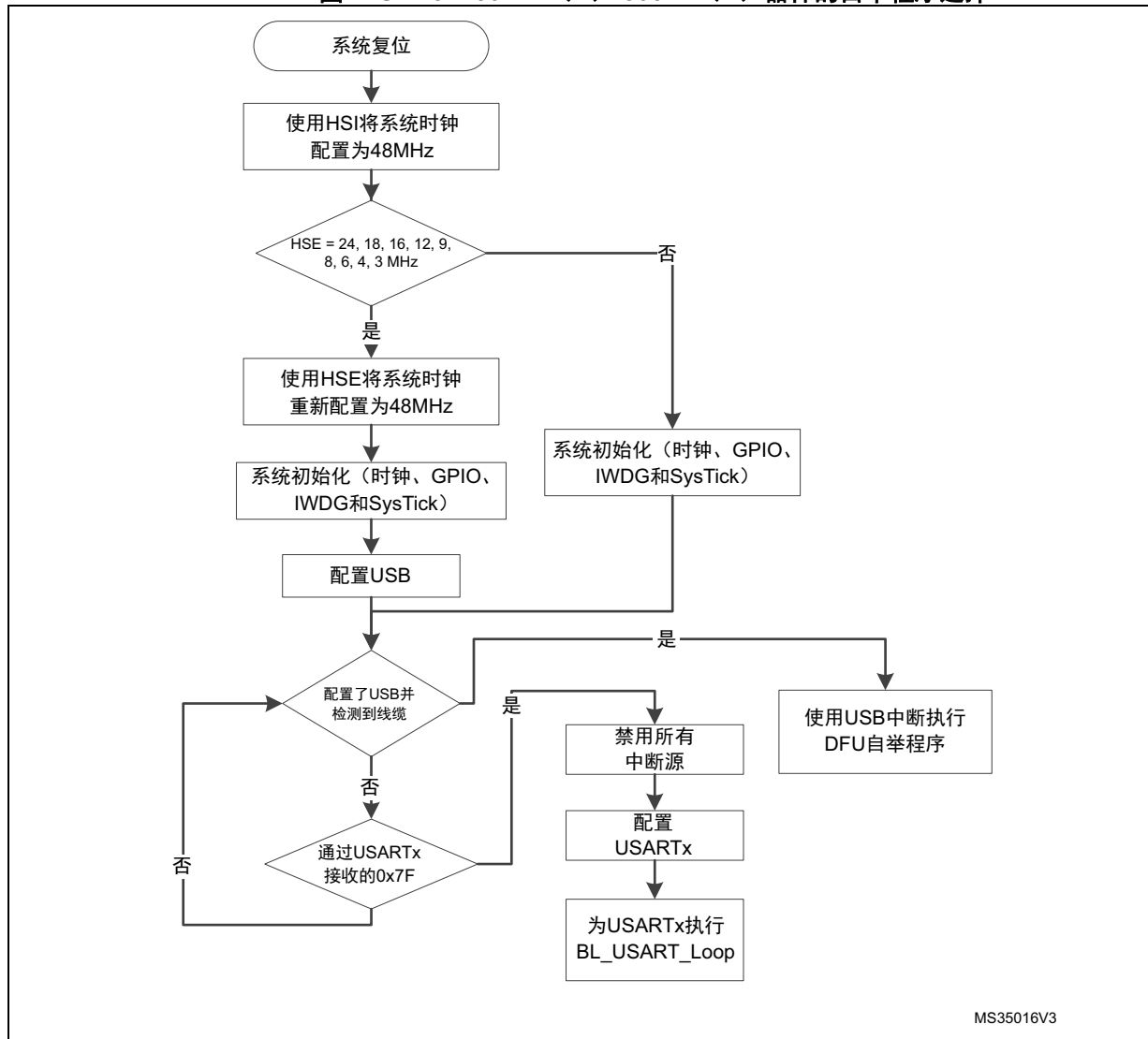
- 如果存在HSE且其值为24、18、16、12、9、8、6、4或3 MHz, 则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1和USART2功能正常, 可用于与器件自举程序进行通信。
- 如果没有HSE, 则使用HSI默认时钟源且只有USART1和USART2功能正常。

如果自举程序启动时连接了外部时钟 (HSE), 则它必须保持, 因为它会用作系统时钟源。

20.2 自举程序选择

下图显示了自举程序选择机制。

图24.STM32F302xB (C) /303xB (C) 器件的自举程序选择



MS35016V3

20.3 自举程序版本

下表列出了STM32F302xB (C) /303xB (C) 器件自举程序版本。

表42.STM32F302xB (C) /303xB (C) 自举程序版本

自举程序版本号	说明	已知限制
V4.1	初始自举程序版本	无

21 STM32F302xD (E) /303xD (E) 器件自举程序

21.1 自举程序配置

STM32F302xD (E) /303xD (E) 自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表43.系统存储器自举模式下STM32F302xD (E) /303xD (E) 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为48 MHz，使用HSI48 48 MHz为时钟源。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、9、8、6、4、3] MHz。PLL用于生成48 MHz USB时钟和48 MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
DFU自举程序	USB:	启用	USBFS配置为强制器件模式。使能USBFS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11引脚：USB FS DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USB FS DP线 用于复用推挽，无上拉/下拉模式。必须连接一个1.5 KΩ的外部上拉电阻到USB_DP引脚。

自举程序具有两种操作情况，取决于自举程序启动时是否存在外部时钟（HSE）：

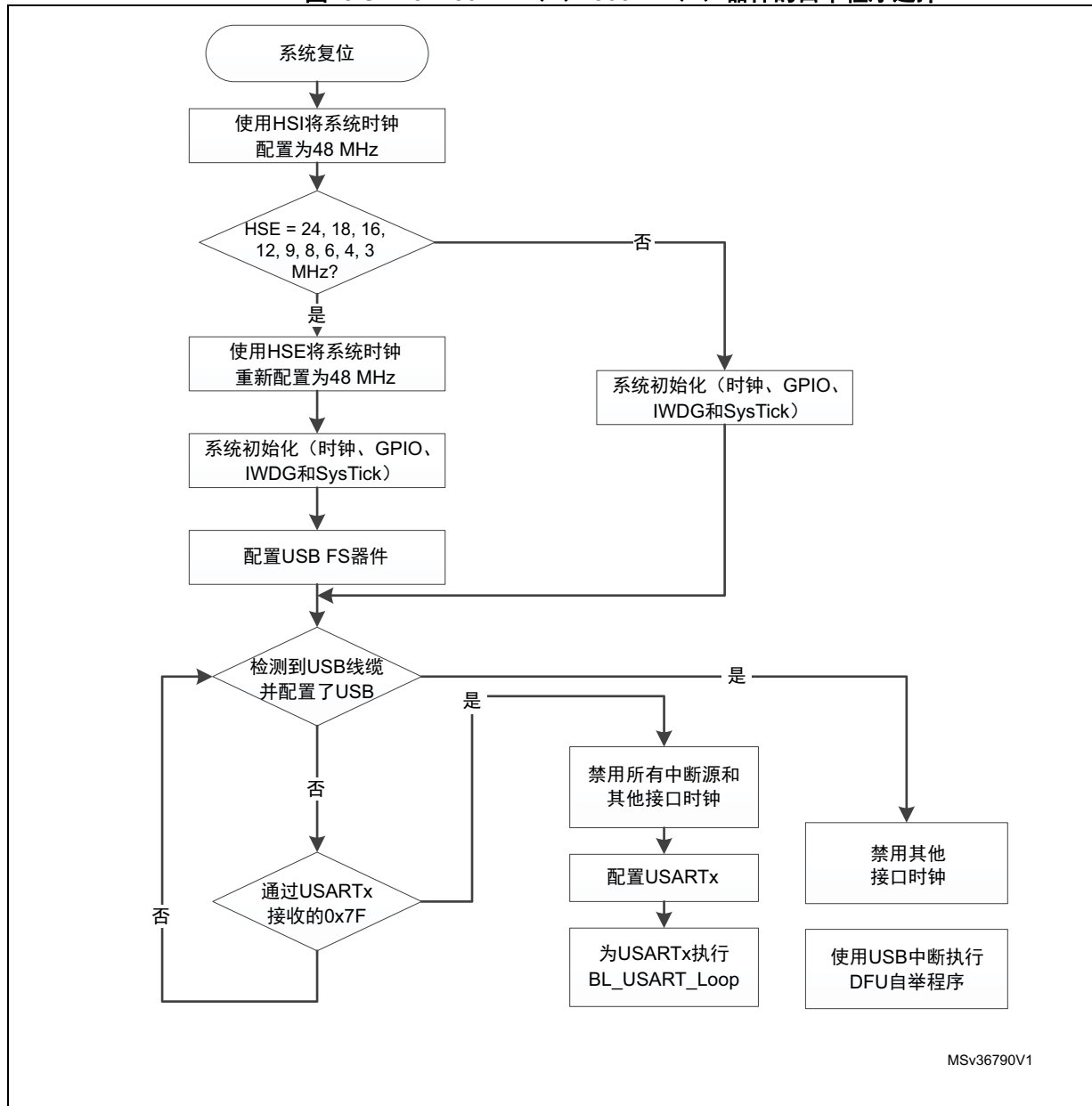
- 如果存在HSE且其值为24、18、16、12、9、8、6、4或3 MHz，则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1和USART2功能正常，可用于与器件自举程序进行通信。
- 如果没有HSE，则使用HSI默认时钟源且只有USART1和USART2功能正常。

如果自举程序启动时连接了外部时钟（HSE），则它必须保持，因为它会用作系统时钟源。

21.2 自举程序选择

下图显示了自举程序选择机制。

图25.STM32F302xD (E) /303xD (E) 器件的自举程序选择



MSv36790V1

21.3 自举程序版本

下表列出了STM32F302xD (E) /303xD (E) 器件自举程序版本。

表44.STM32F302xD (E) /303xD (E) 自举程序版本

自举程序版本号	说明	已知限制
V4.0	初始自举程序版本	无

22 STM32F303x4 (6/8) /334xx/328xx器件自举程序

22.1 自举程序配置

STM32F303x4 (6/8) /334xx/328xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表45.系统存储器自举模式下STM32F303x4 (6/8) /334xx/328xx器件的配置

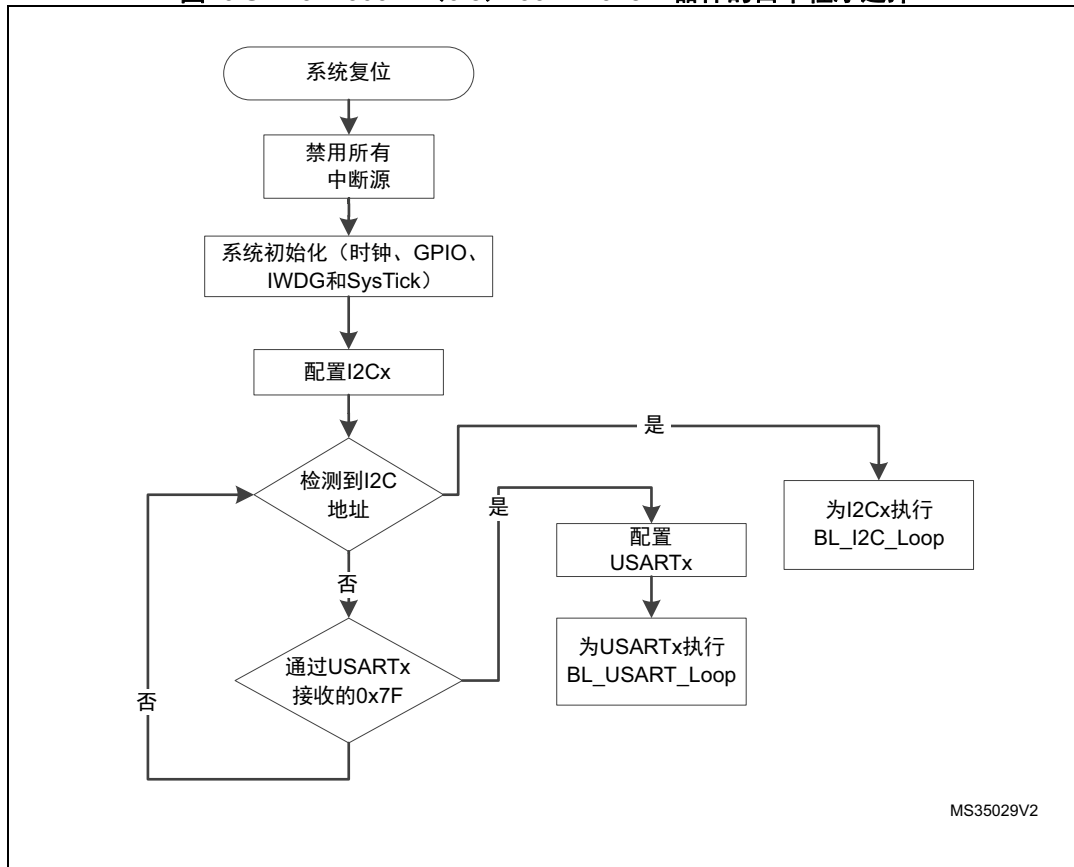
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为60 MHz，使用HSI 8 MHz为时钟源。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111111x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

22.2 自举程序选择

下图显示了自举程序选择机制。

图26.STM32F303x4 (6/8) /334xx/328xx器件的自举程序选择



22.3 自举程序版本

下表列出了STM32F303x4 (6/8) /334xx/328xx器件自举程序版本：

表46.STM32F303x4 (6/8) /334xx/328xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	无

23 STM32F318xx器件自举程序

23.1 自举程序配置

STM32F318xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表47.系统存储器自举模式下STM32F318xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为60 MHz，使用HSI 8 MHz为时钟源。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111101x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

表47.系统存储器自举模式下STM32F318xx器件的配置（续）

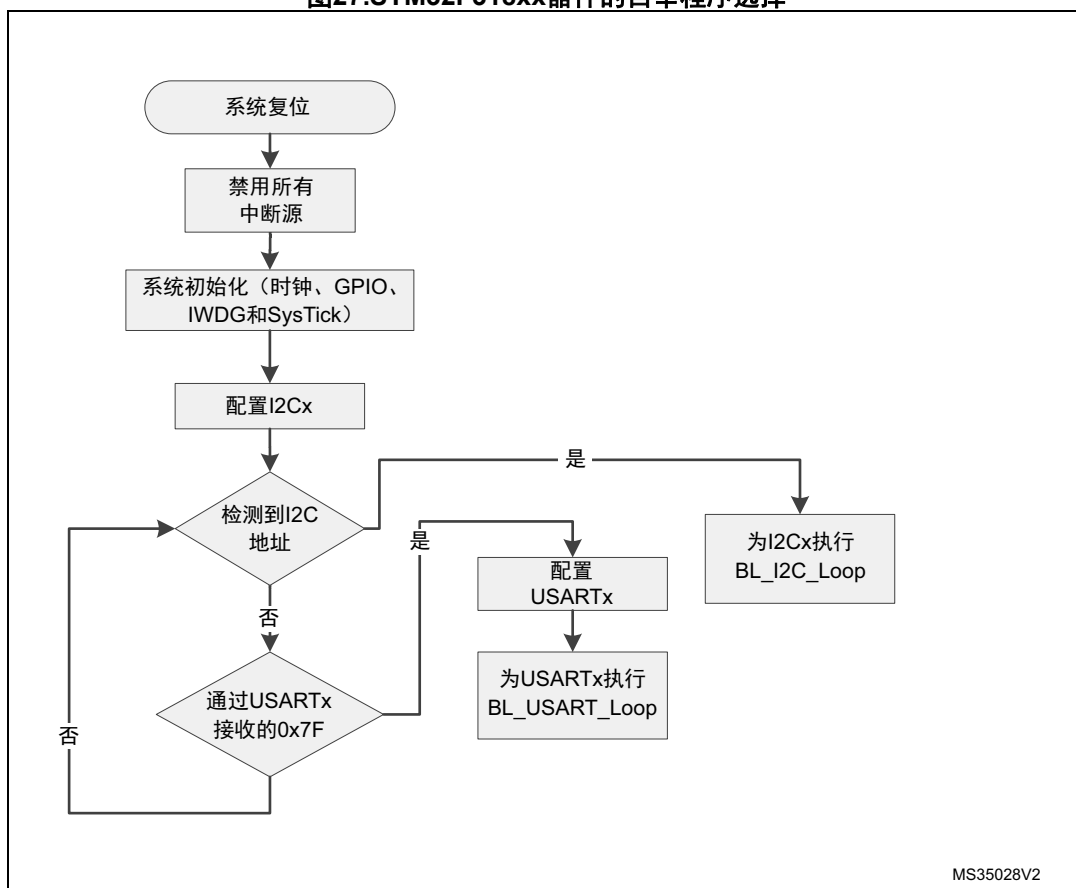
自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111101x（其中x = 0对应写，x = 1对应读）且数字滤波器禁用。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB5引脚：数据线在开漏无上拉/下拉模式下使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

23.2 自举程序选择

下图显示了自举程序选择机制。

图27.STM32F318xx器件的自举程序选择



23.3 自举程序版本

下表列出了STM32F318xx器件自举程序版本：

表48.STM32F318xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	无

24 STM32F358xx器件自举程序

24.1 自举程序配置

STM32F358xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表49.系统存储器自举模式下STM32F358xx器件的配置

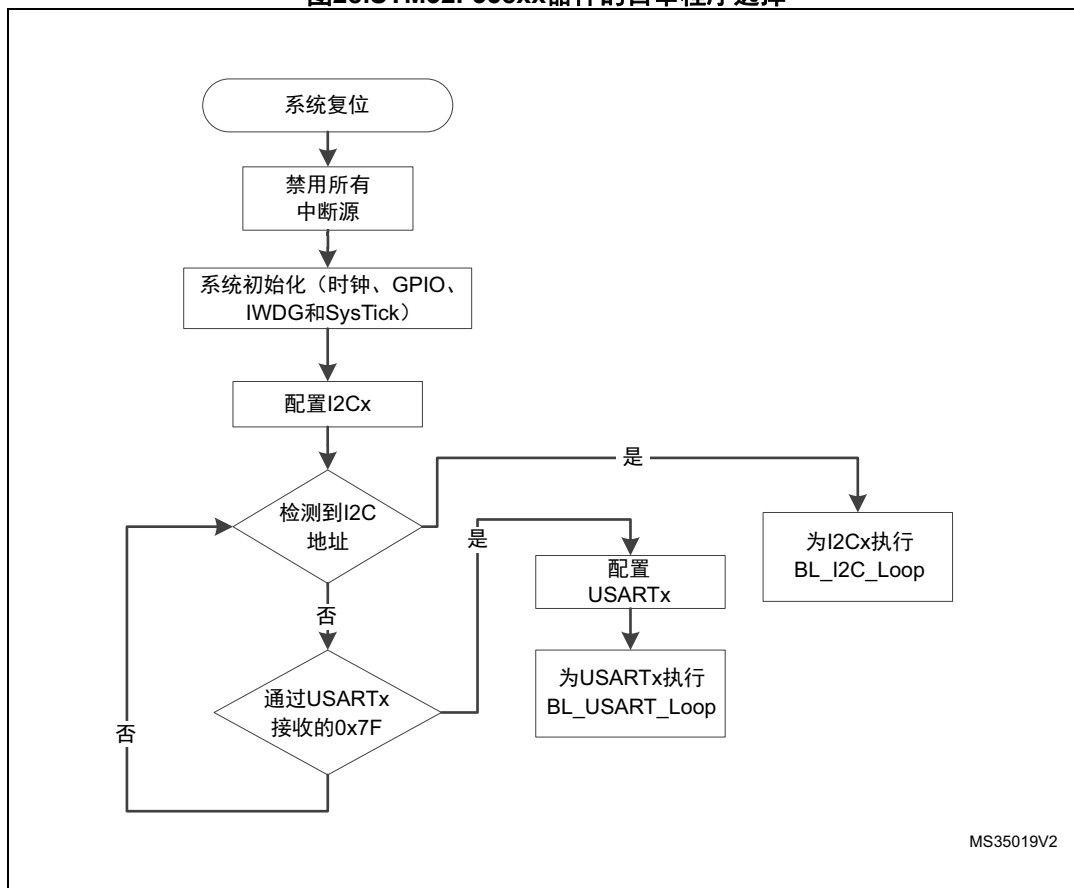
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为8 MHz。
	RAM	-	自地址0x20000000起的5KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。禁止窗口功能。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0110111x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

24.2 自举程序选择

下图显示了自举程序选择机制。

图28.STM32F358xx器件的自举程序选择



MS35019V2

24.3 自举程序版本

下表列出了STM32F358xx器件自举程序版本。

表50. STM32F358xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	对于USART1和USART2接口，自举程序支持的最大波特率为57600波特。

25 STM32F373xx器件自举程序

25.1 自举程序配置

STM32F373xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表51.系统存储器自举模式下STM32F373xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	启动时，使用HSI将系统时钟频率配置为48MHz。如果没有外部时钟（HSE），系统时钟将由HSI提供。
		HSE使能	外部时钟可用于所有自举程序接口，其频率值须为如下之一：[24、18、16、12、9、8、6、4、3] MHz。 PLL用于生成48MHz USB时钟和48MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的5KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表51.系统存储器自举模式下STM32F373xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 必须连接一个1.5 KΩ的外部上拉电阻到USB_DP引脚。

自举程序具有两种操作情况，取决于自举程序启动时是否存在外部时钟（HSE）：

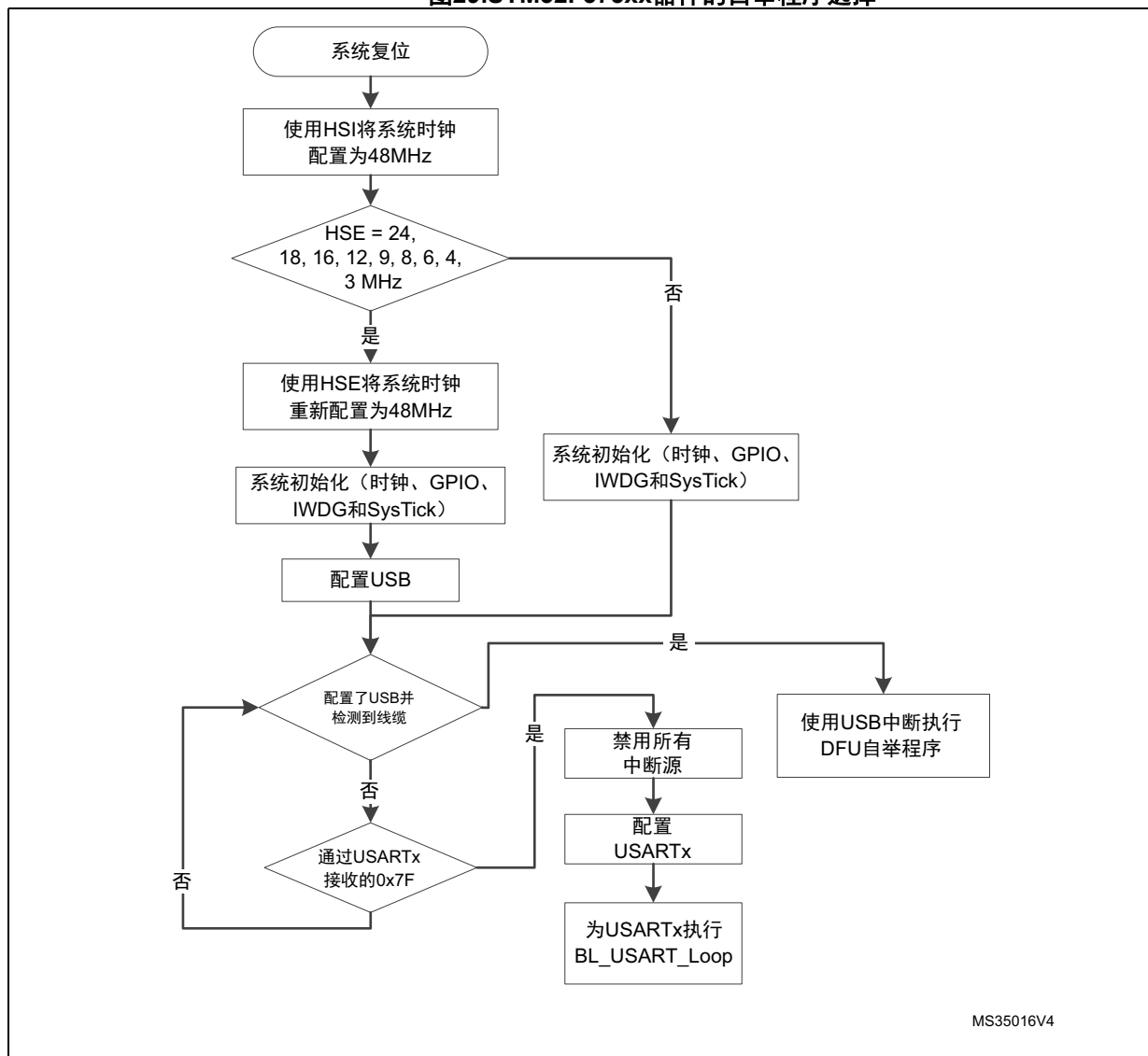
- 如果存在HSE且其值为24、18、16、12、9、8、6、4或3 MHz，则使用HSE作为时钟源并将系统时钟配置为48 MHz。DFU接口、USART1和USART2功能正常，可用于与器件自举程序进行通信。
- 如果没有HSE，则使用HSI默认时钟源且只有USART1和USART2功能正常。

注： 如果自举程序启动时连接了外部时钟（HSE），则它必须保持，因为它会用作系统时钟源。

25.2 自举程序选择

下图显示了自举程序选择机制。

图29.STM32F373xx器件的自举程序选择



MS35016V4

25.3 自举程序版本

下表列出了STM32F373xx器件自举程序版本。

表52.STM32F373xx自举程序版本

自举程序版本号	说明	已知限制
V4.1	初始自举程序版本	无

26 STM32F378xx器件自举程序

26.1 自举程序配置

STM32F378xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表53.系统存储器自举模式下STM32F378xx器件的配置

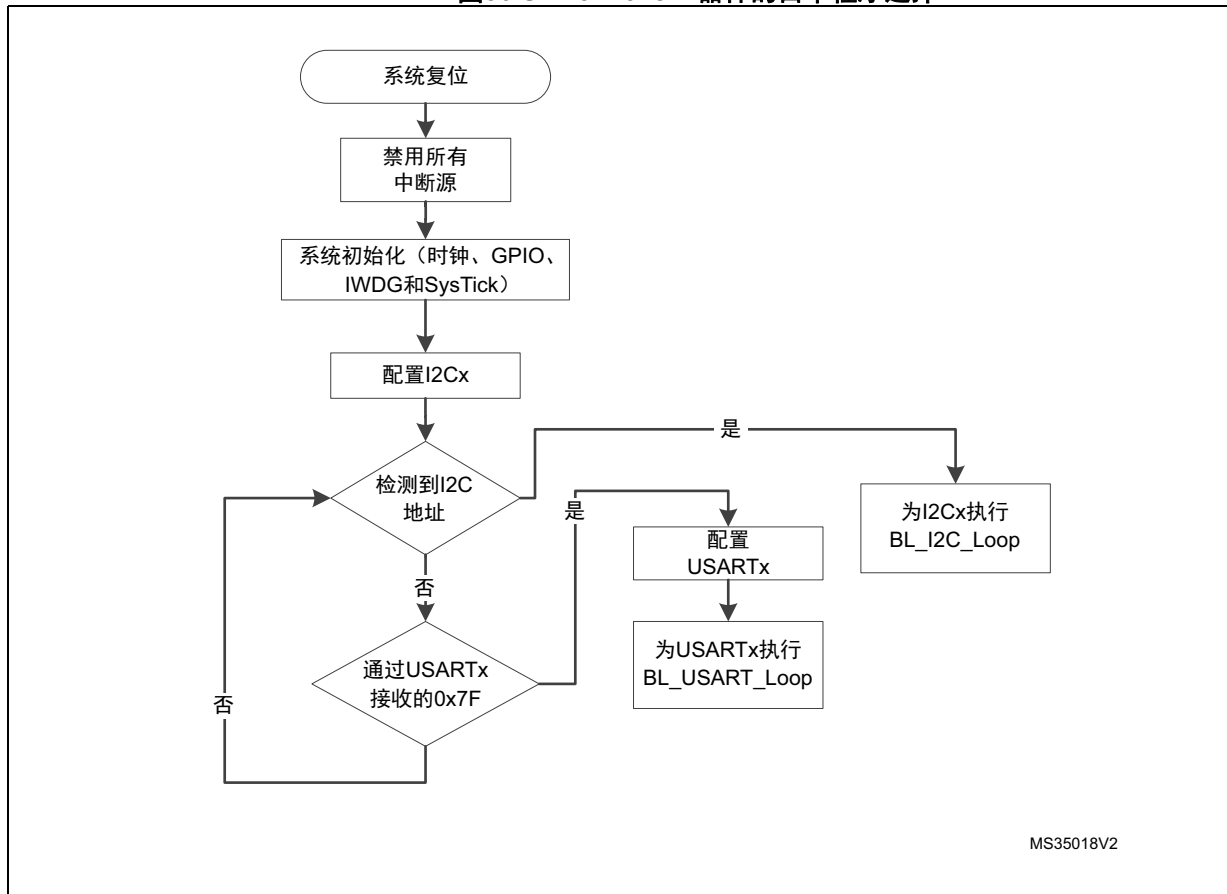
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为8 MHz。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FFFD800起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。禁止窗口功能。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0110111x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

26.2 自举程序选择

下图显示了自举程序选择机制。

图30.STM32F378xx器件的自举程序选择



26.3 自举程序版本

下表列出了STM32F378xx器件自举程序版本。

表54. STM32F378xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	对于USART1和USART2接口，自举程序支持的最大波特率为57600波特。

27 STM32F398xx器件自举程序

27.1 自举程序配置

STM32F398xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表55.系统存储器自举模式下STM32F398xx器件的配置

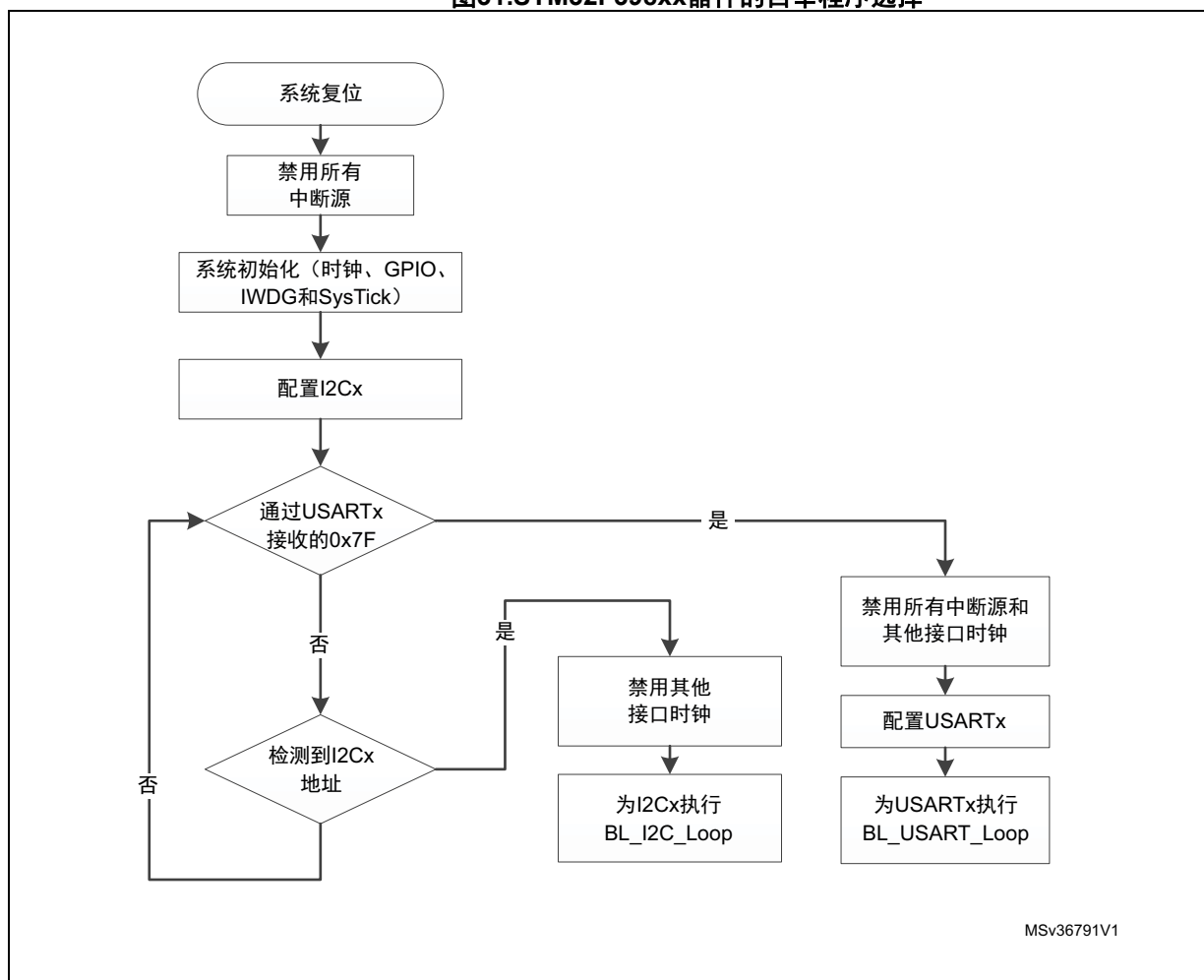
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为60 MHz，使用HSI 8 MHz为时钟源。
	RAM	-	自地址0x20000000起的6 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFD800起的7 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000000x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000000x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB5引脚：数据线在开漏无上拉/下拉模式下使用。

对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

27.2 自举程序选择

下图显示了自举程序选择机制。

图31.STM32F398xx器件的自举程序选择



27.3 自举程序版本

下表列出了STM32F398xx器件自举程序版本。

表56.STM32F398xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	无

28 STM32F40xxx/41xxx器件自举程序

28.1 自举程序V3.x

28.1.1 自举程序配置

STM32F40xxx/41xxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表57.系统存储器自举模式下STM32F40xxx/41xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USARTx接口期间（选择CAN或DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择CAN或DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的8 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF 0000起的29 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表57.系统存储器自举模式下STM32F40xxx/41xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
DFU自举程序	USB：	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

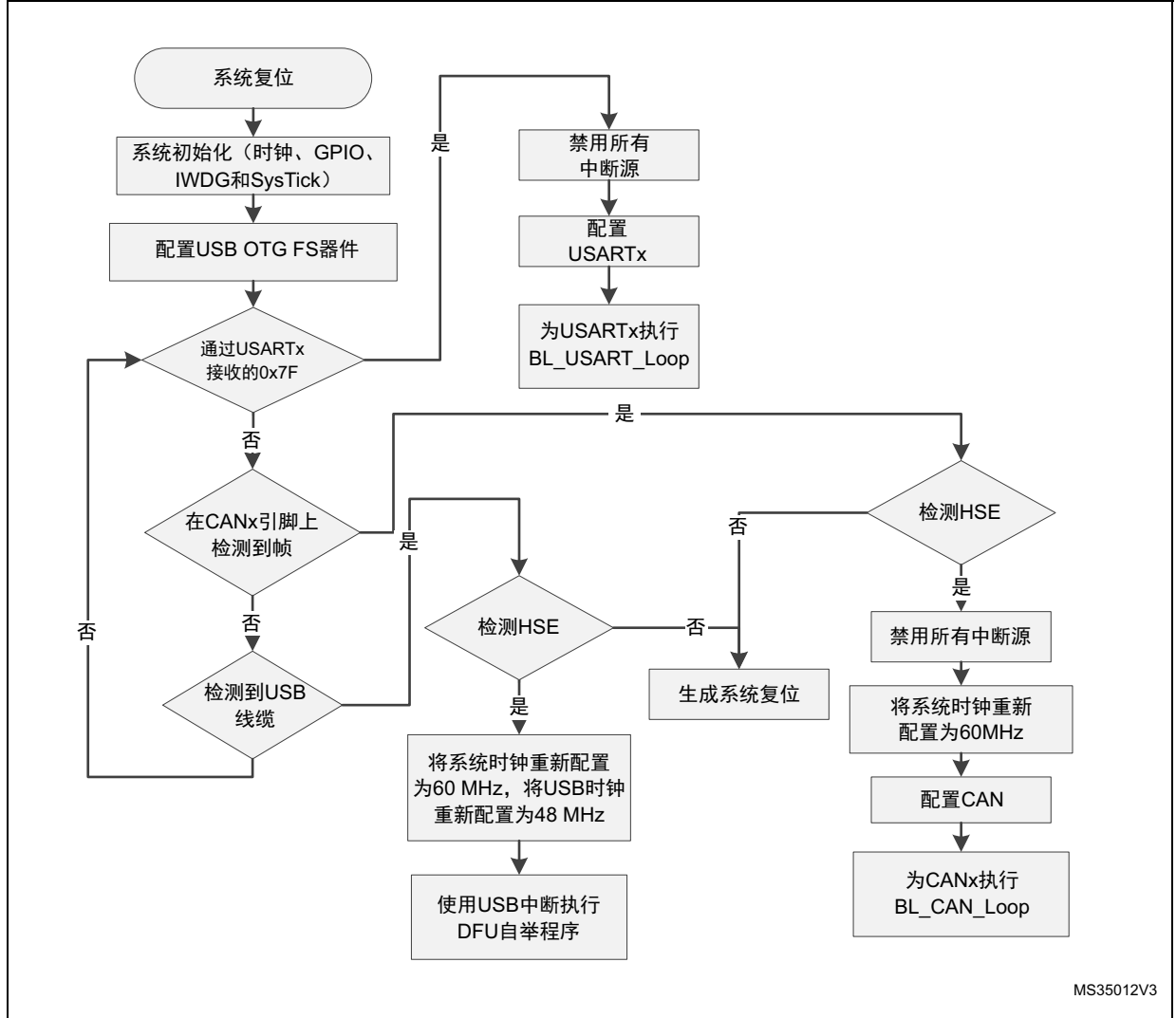
对于USARTx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

28.1.2 自举程序选择

下图显示了自举程序选择机制。

图32.STM32F40xxx/41xxx器件的自举程序V3.x选择



MS35012V3

28.1.3 自举程序版本

下表列出了STM32F40xxx/41xxx器件V3.x自举程序版本：

表58. STM32F40xxx/41xxx自举程序V3.x版本

自举程序版本号	说明	已知限制
V3.0	初始自举程序版本	<ul style="list-style-type: none"> – 当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址0x6000 0000）发出Read Memory命令或WriteMemory命令时，自举程序器件会中止这一命令，但不会向主机发送NACK (0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和⁽¹⁾。 – 选项字节、OTP和设备功能描述符（位于DFU接口中）设置为“g”而不是“e”（不是可擦除存储区）。 <p>在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。</p>
V3.1	修正V3.0限制。提高DFU接口耐用性。	<ul style="list-style-type: none"> – 对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号（而不是1个NACK信号）。 – 对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。 <p>在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。</p>

1. 如果待读/写的“数据数 - 1” (N-1) 不等于有效命令代码 (0x00、0x01、0x02、0x11、0x21、0x31、0x43、0x44、0x63、0x73、0x82或0x92)，则无法从主机发现缺陷，因为该命令始终会收到NACK应答（当作不受支持的新命令）。

28.2 自举程序V9.x

28.2.1 自举程序配置

STM32F40xxx/41xxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。表 59显示了该自举程序使用的硬件资源。

注：只有采用WLCSP90封装的STM32F405xx/415xx器件才内置自举程序版本V9.0。
版本V9.1被迁移到产品的所有封装中。

表59.系统存储器自举模式下STM32F40xxx/41xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择CAN或DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。仅当选择CAN或DFU（USBFS设备）接口时才使用HSE时钟源。外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式

表59.系统存储器自举模式下STM32F40xxx/41xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111010x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111010x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111010x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。

表59.系统存储器自举模式下STM32F40xxx/41xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB：	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

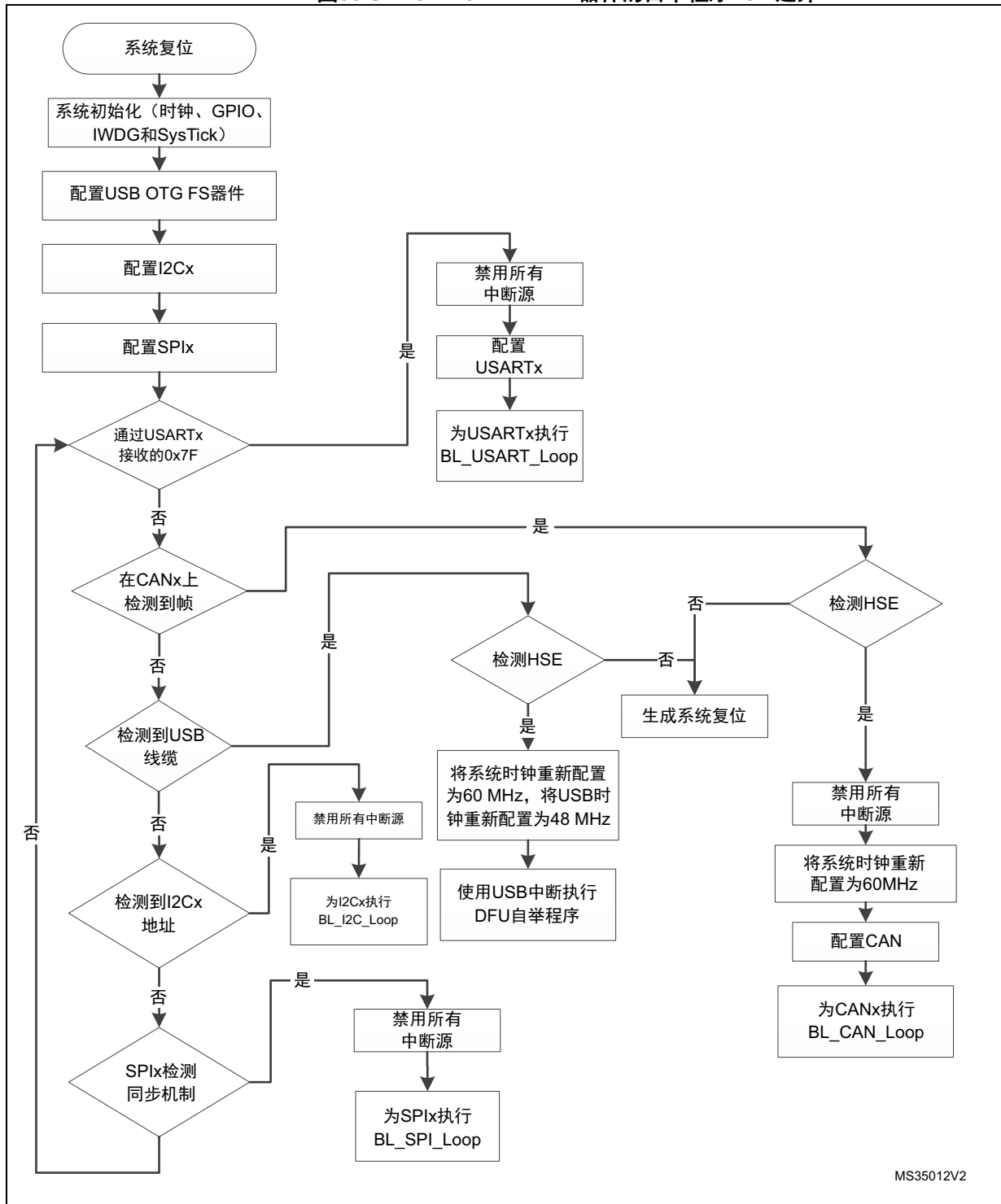
对于USARTx、I2Cx和SPIx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USB FS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

28.2.2 自举程序选择

下图显示了自举程序选择机制。

图33.STM32F40xxx/41xxx器件的自举程序V9.x选择



28.2.3 自举程序版本

下表列出了STM32F40xxx/41xxx器件V9.x自举程序版本。

表60.STM32F40xxx/41xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	<p>此自举程序是自举程序v3.1的更新版本。</p> <p>该新版本自举程序支持I2C1、I2C2、I2C3、SPI1和SPI2接口。</p> <p>此自举程序所用的RAM从8KB增加到了12KB。</p> <p>此自举程序的ID为0x90。</p> <p>连接时间增加了。</p>	<ul style="list-style-type: none"> – 对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号（而不是1个NACK信号）。 – 对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。 <p>在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。</p>
V9.1	<p>此自举程序版本是自举程序v9.0的更新版本，将被迁移到所有封装中，即使是内置自举程序版本V3.1的封装。</p> <p>它包含对V9.0的已知限制的修复。</p>	无

29 STM32F401xB (C) 器件自举程序

29.1 自举程序配置

STM32F401xB (C) 自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表61.系统存储器自举模式下STM32F401xB (C) 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表61.系统存储器自举模式下STM32F401xB (C) 器件的配置 (续)

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB3引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB4引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表61.系统存储器自举模式下STM32F401xB (C) 器件的配置 (续)

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

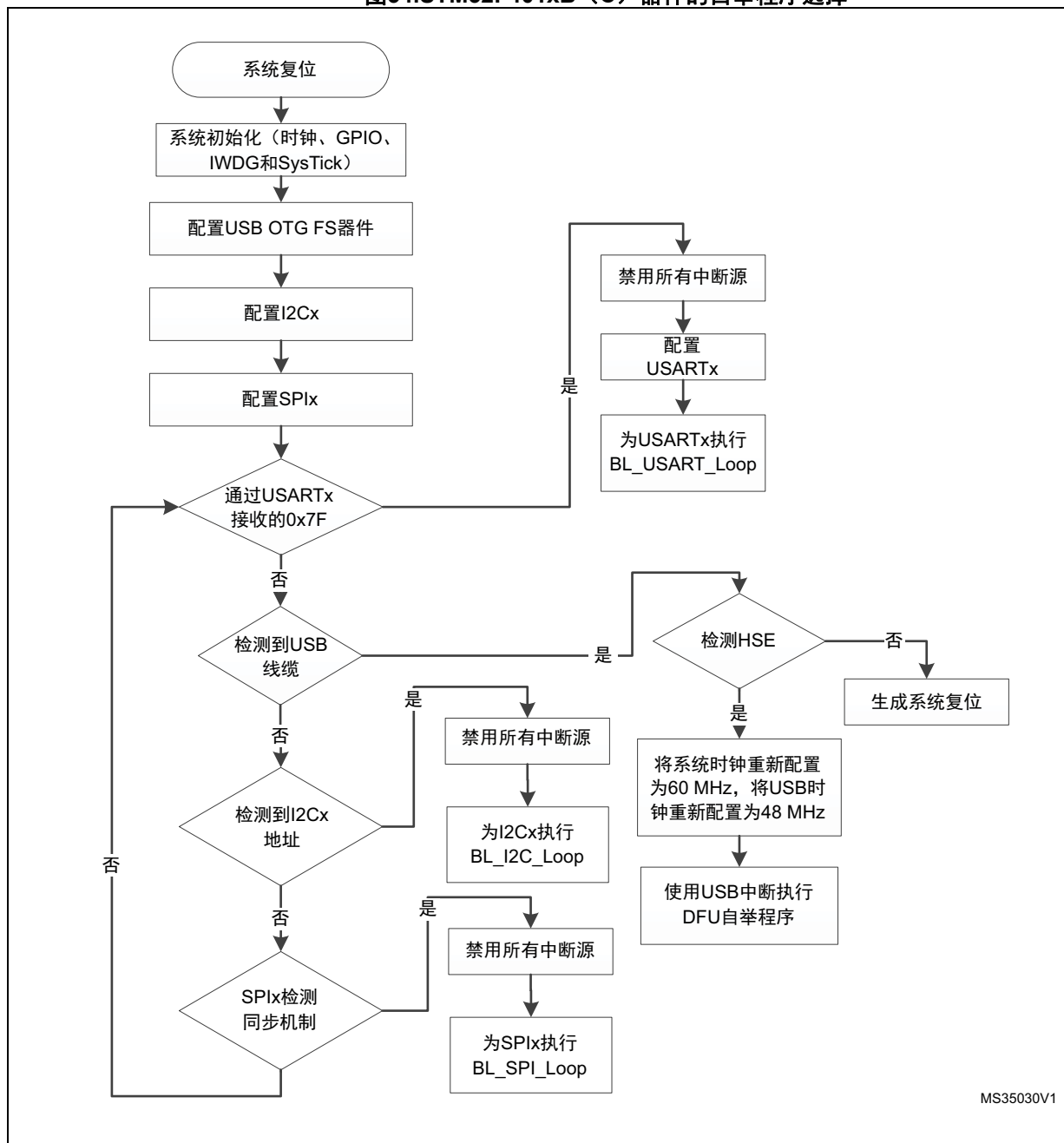
对于USARTx、I2Cx和SPIx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USB FS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

29.2 自举程序选择

下图显示了自举程序选择机制。

图34.STM32F401xB (C) 器件的自举程序选择



29.3 自举程序版本

下表列出了STM32F401xB (C) 器件自举程序版本。

表62.STM32F401xB (C) 自举程序版本

自举程序版本号	说明	已知限制
V13.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

30 STM32F401xD (E) 器件自举程序

30.1 自举程序配置

STM32F401xD (E) 自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表63.系统存储器自举模式下STM32F401xD (E) 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表63.系统存储器自举模式下STM32F401xD (E) 器件的配置 (续)

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB3引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB4引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表63.系统存储器自举模式下STM32F401xD (E) 器件的配置 (续)

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11: USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

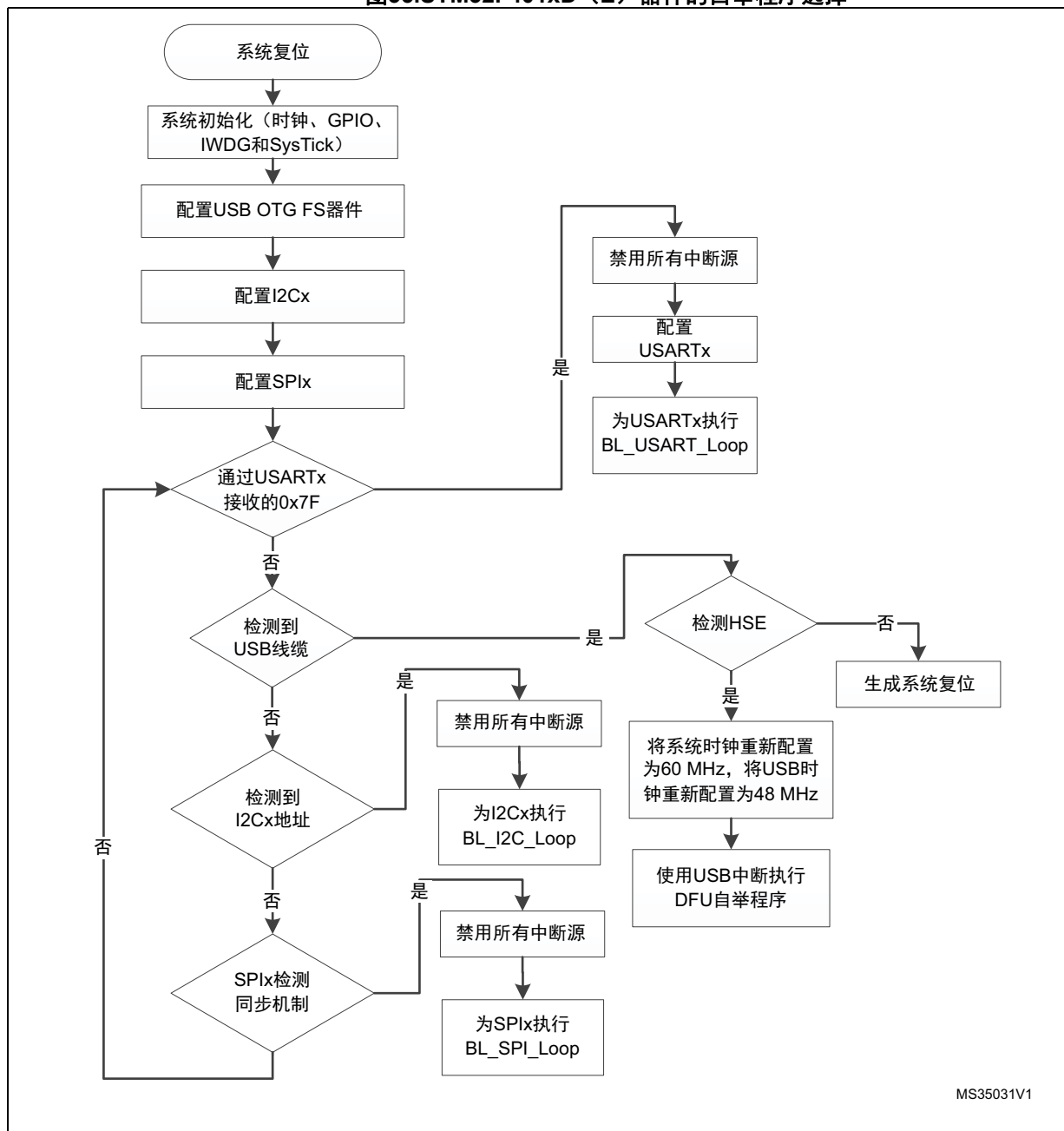
对于USARTx、I2Cx和SPIx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

30.2 自举程序选择

下图显示了自举程序选择机制。

图35.STM32F401xD (E) 器件的自举程序选择



30.3 自举程序版本

下表列出了STM32F401xD (E) 器件自举程序版本。

表64.STM32F401xD (E) 自举程序版本

自举程序版本号	说明	已知限制
V13.1	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

31 STM32F410xx器件自举程序

31.1 自举程序配置

STM32F410xx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表65.系统存储器自举模式下STM32F410xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
	RAM	-	自地址0x20000000起的5 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表65.系统存储器自举模式下STM32F410xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000111x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000111x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C4自举程序	I2C4	启用	I2C4配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000111x（其中x = 0对应写，x = 1对应读）。
	I2C4_SCL引脚	输入/输出	PB15引脚：对于STM32F410Cx/Rx器件，时钟线在开漏无上拉/下拉模式下使用。 PB10引脚：对于STM32F410Tx器件，时钟线在开漏无上拉/下拉模式下使用。
	I2C4_SDA引脚	输入/输出	PB14引脚：对于STM32F410Cx/Rx器件，数据线在开漏无上拉/下拉模式下使用。 PB3引脚：对于STM32F410Tx器件，数据线在开漏无上拉/下拉模式下使用。

表65.系统存储器自举模式下STM32F410xx器件的配置（续）

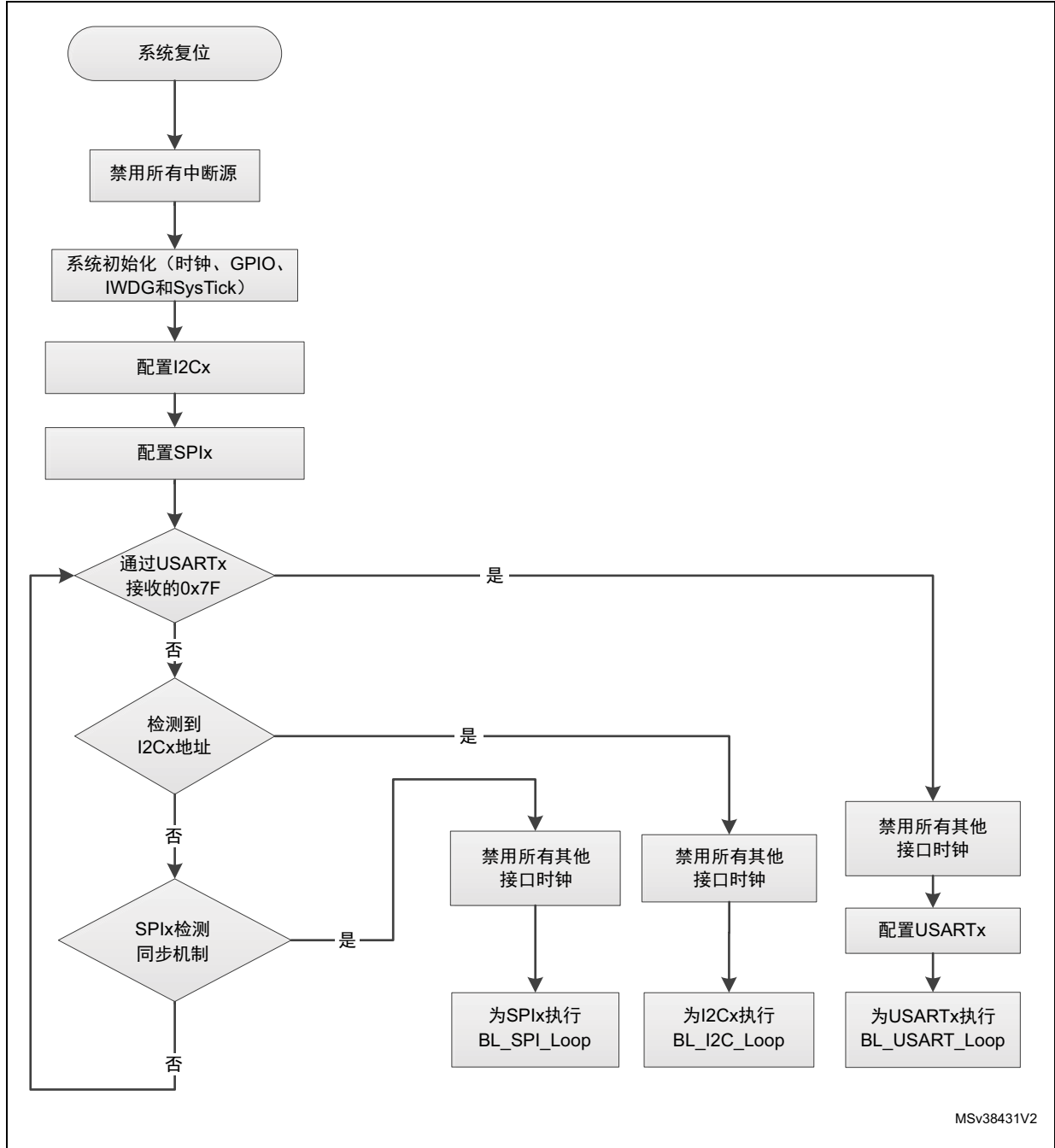
自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，对于STM32F410Cx/Rx器件，用于推挽下拉模式。 PB5引脚：从数据输入线对于STM32F410Tx器件，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，对于STM32F410Cx/Rx器件，用于推挽下拉模式。 PB4引脚：从数据输出线，对于STM32F410Tx器件，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，对于STM32F410Cx/Rx器件，用于推挽上拉模式。 PA15引脚：从芯片选择引脚，对于STM32F410Tx器件，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PC3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PC2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。

对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

31.2 自举程序选择

图 36显示了自举程序选择机制。

图36.STM32F410xx器件的自举程序V11.x选择



31.3 自举程序版本

下表列出了STM32F410xx器件自举程序V11.x版本。

表66.STM32F410xx自举程序V11.x版本

自举程序版本号	说明	已知限制
V11.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。
V11.1	对于STM32F410Tx器件，支持I2C4和SPI1。	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

32 STM32F411xx器件自举程序

32.1 自举程序配置

STM32F411xx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表67.系统存储器自举模式下STM32F411xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表67.系统存储器自举模式下STM32F411xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB3引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111001x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB4引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表67.系统存储器自举模式下STM32F411xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB：	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

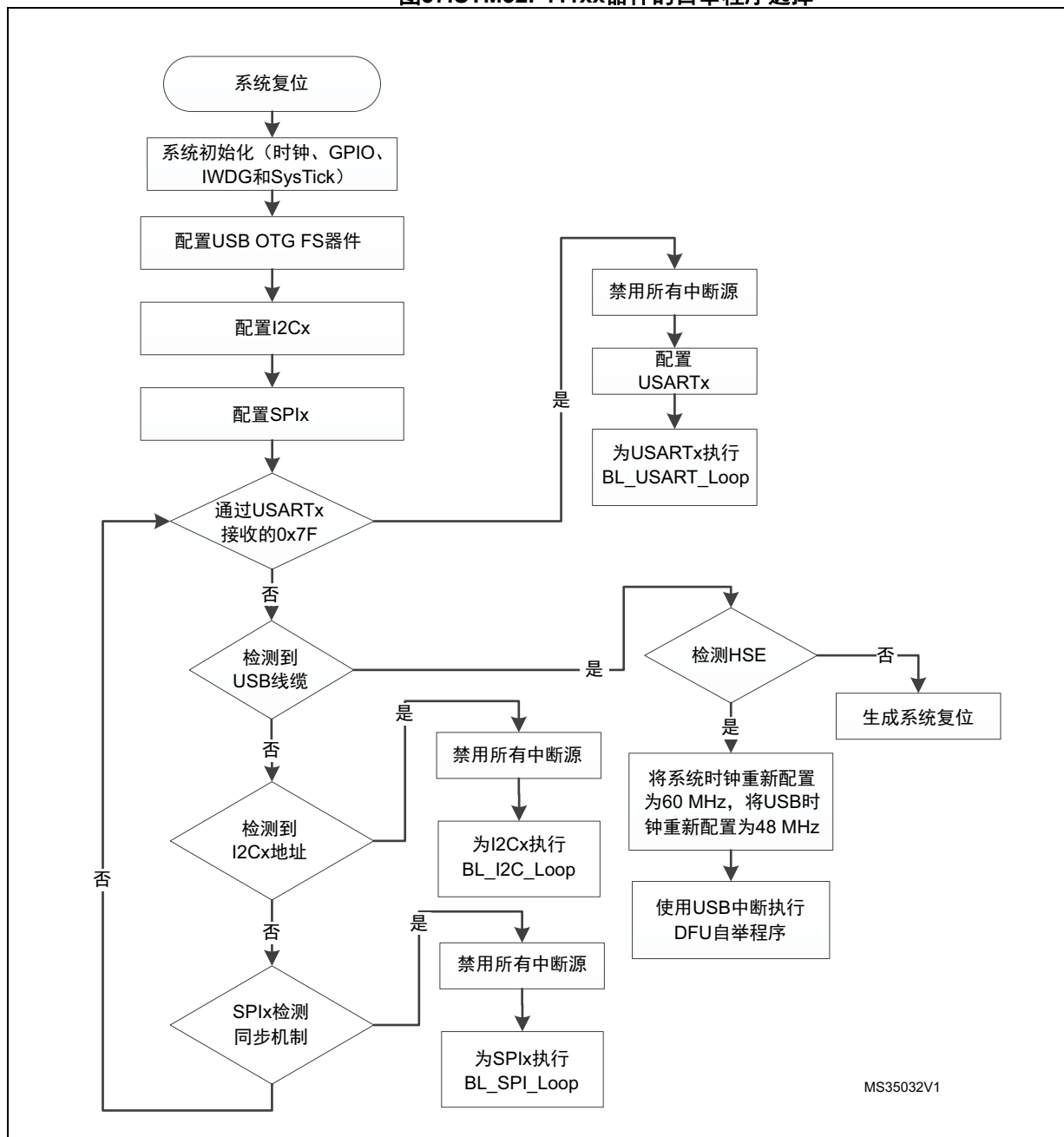
对于USARTx、I2Cx和SPIx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

32.2 自举程序选择

下图显示了自举程序选择机制。

图37.STM32F411xx器件的自举程序选择



MS35032V1

32.3 自举程序版本

下表列出了STM32F411xx器件自举程序版本。

表68.STM32F411xx自举程序版本

自举程序版本号	说明	已知限制
V13.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

33 STM32F412xx器件自举程序

33.1 自举程序配置

STM32F412xx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。表中介绍了该自举程序使用的硬件资源。

表69.系统存储器自举模式下STM32F412xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于输入上拉模式。

表69.系统存储器自举模式下STM32F412xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000110x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000110x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000110x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB4引脚：数据线在开漏无上拉/下拉模式下使用。

表69.系统存储器自举模式下STM32F412xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C4自举程序	I2C4	启用	I2C4配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000110x（其中x = 0对应写，x = 1对应读）
	I2C4_SCL引脚	输入/输出	PB15引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C4_SDA引脚	输入/输出	PB14引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽上拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽上拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽上拉模式。

表69.系统存储器自举模式下STM32F412xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11引脚: USB _{DM} 线。用于复用推挽, 无上拉/下拉模式。
	USB_DP引脚		PA12引脚: USB _{DP} 线。用于复用推挽, 无上拉/下拉模式。 无需外部上拉电阻。
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率, 可利用PLL和HSE将系统时钟配置为60 MHz。

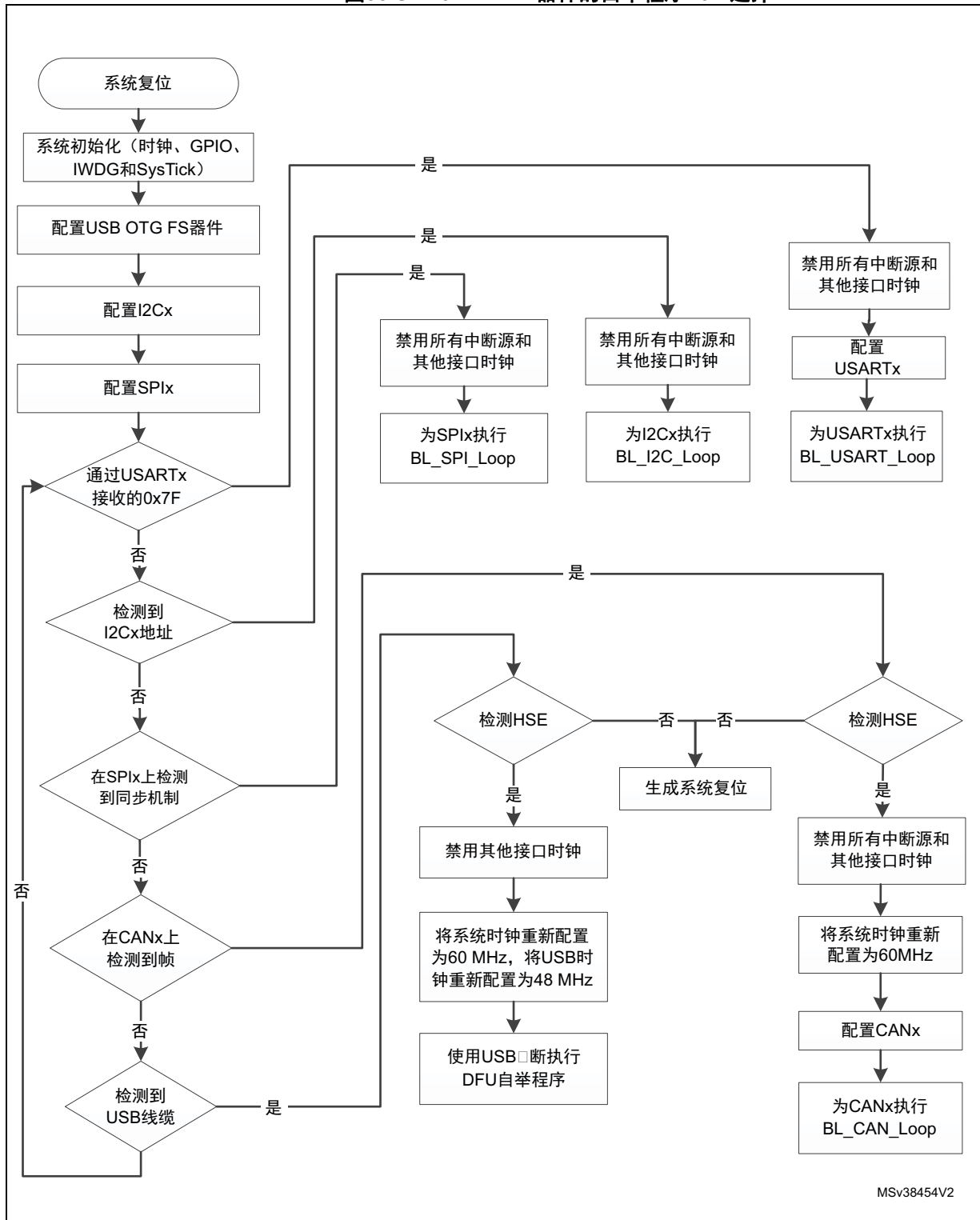
对于USARTx和I2Cx自举程序, 系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU (USBFS设备), 但仅限选择阶段。选择阶段结束后, 执行CAN和DFU自举程序需要使用频率为1 MHz的倍数 (介于4 MHz到26 MHz之间) 的外部时钟。

注: 由于存在HSI偏差且HSI被用于检测HSE值, 用户必须使用低频而不是高频HSE晶振值 (低频值的误差容限更大, 因此更好检测)。例如, 最好使用8 MHz, 而不是25 MHz。

33.2 自举程序选择

图 38显示了自举程序选择机制。

图38.STM32F412xx器件的自举程序V9.x选择



MSv38454V2



33.3 自举程序版本

下表列出了STM32F412xx器件自举程序V9.x版本。

表70.STM32F412xx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。
V9.1	修复了USART3接口引脚排列	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

34 STM32F413xx/423xx器件自举程序

34.1 自举程序配置

STM32F413xx/423xx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表71.系统存储器自举模式下STM32F413xx/423xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的60 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 4个Flash等待周期。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考 自举程序存储器管理 来获取更多信息）。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表71.系统存储器自举模式下STM32F413xx/423xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于输入上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001011x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001011x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表71.系统存储器自举模式下STM32F413xx/423xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001011x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB4引脚：数据线在开漏无上拉/下拉模式下使用。
I2C4自举程序	I2C4	启用	I2C4配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001011x（其中x = 0对应写，x = 1对应读）
	I2C4_SCL引脚	输入/输出	PB15引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C4_SDA引脚	输入/输出	PB14引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： – 从模式 – 全双工 – 8位MSB，速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： – 从模式 – 全双工 – 8位MSB，速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽下拉模式。

表71.系统存储器自举模式下STM32F413xx/423xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI4自举程序	SPI4	启用	SPI4配置为： – 从模式 – 全双工 – 8位MSB，速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SP4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11引脚：USBM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USBP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

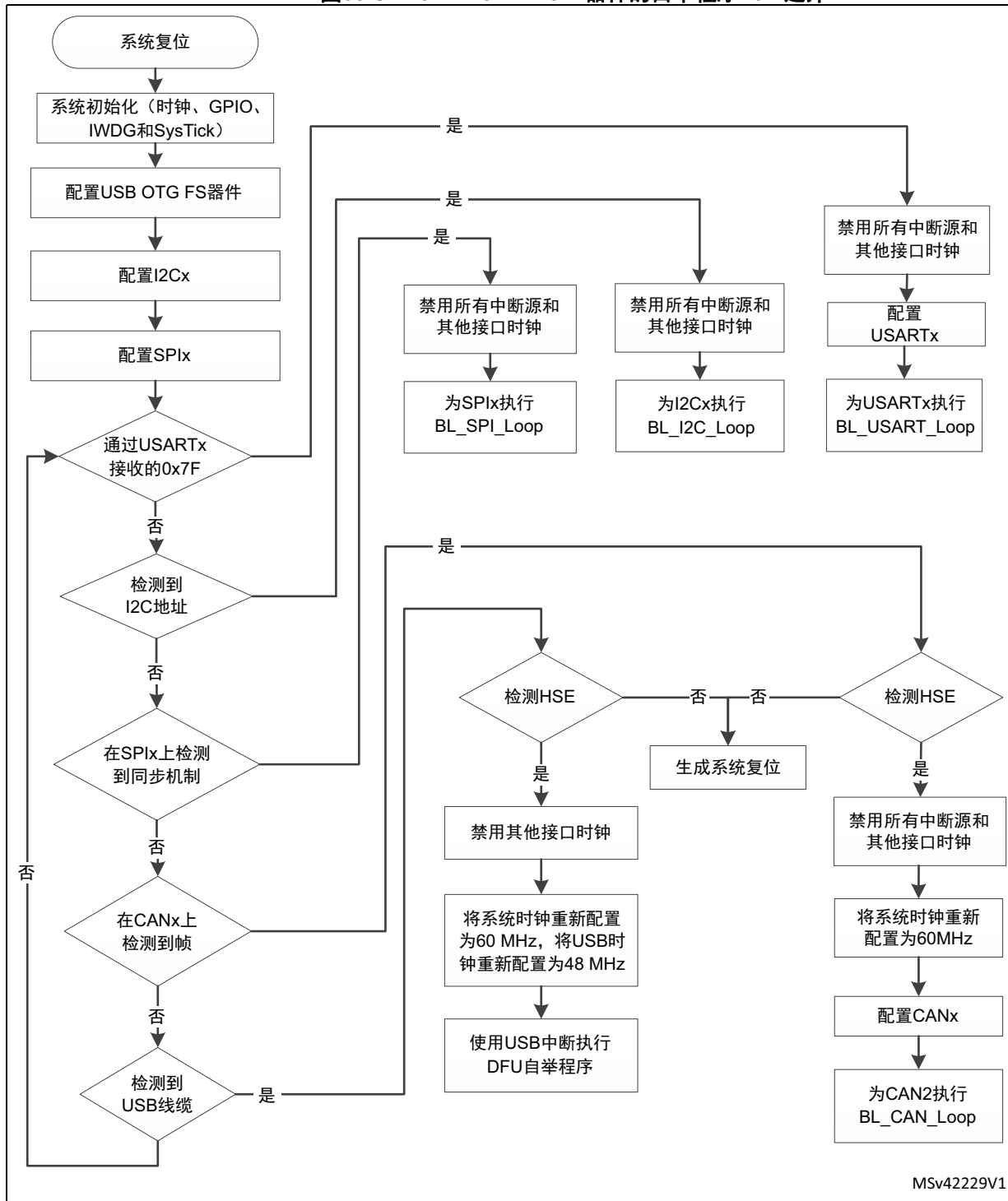
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注：由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

34.2 自举程序选择

图 39显示了自举程序选择机制。

图39.STM32F413xx/423xx器件的自举程序V9.x选择



34.3 自举程序版本

下表列出了STM32F413xx/423xx器件自举程序V9.x版本。

表72.STM32F413xx/423xx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

35 STM32F42xxx/43xxx器件自举程序

35.1 自举程序V7.x

35.1.1 自举程序配置

STM32F42xxx/43xxx自举程序通过应用模式5激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表73.系统存储器自举模式下STM32F42xxx/43xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为24 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USART或I2C接口期间（选择CAN或DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择CAN或DFU（USBFS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的8 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。

表73.系统存储器自举模式下STM32F42xxx/43xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111000x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

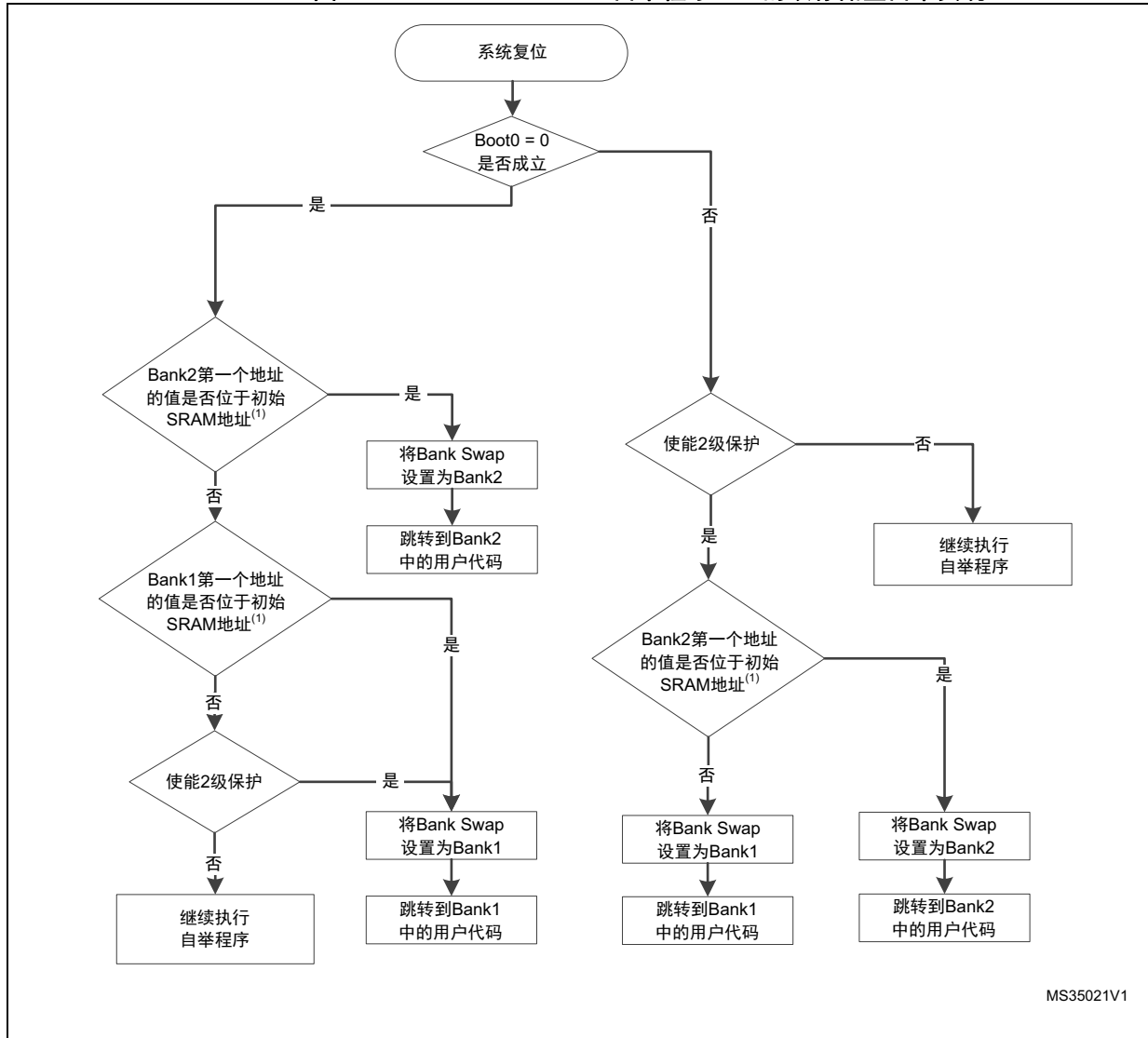
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

35.1.2 自举程序选择

图 40和图 41显示了自举程序选择机制。

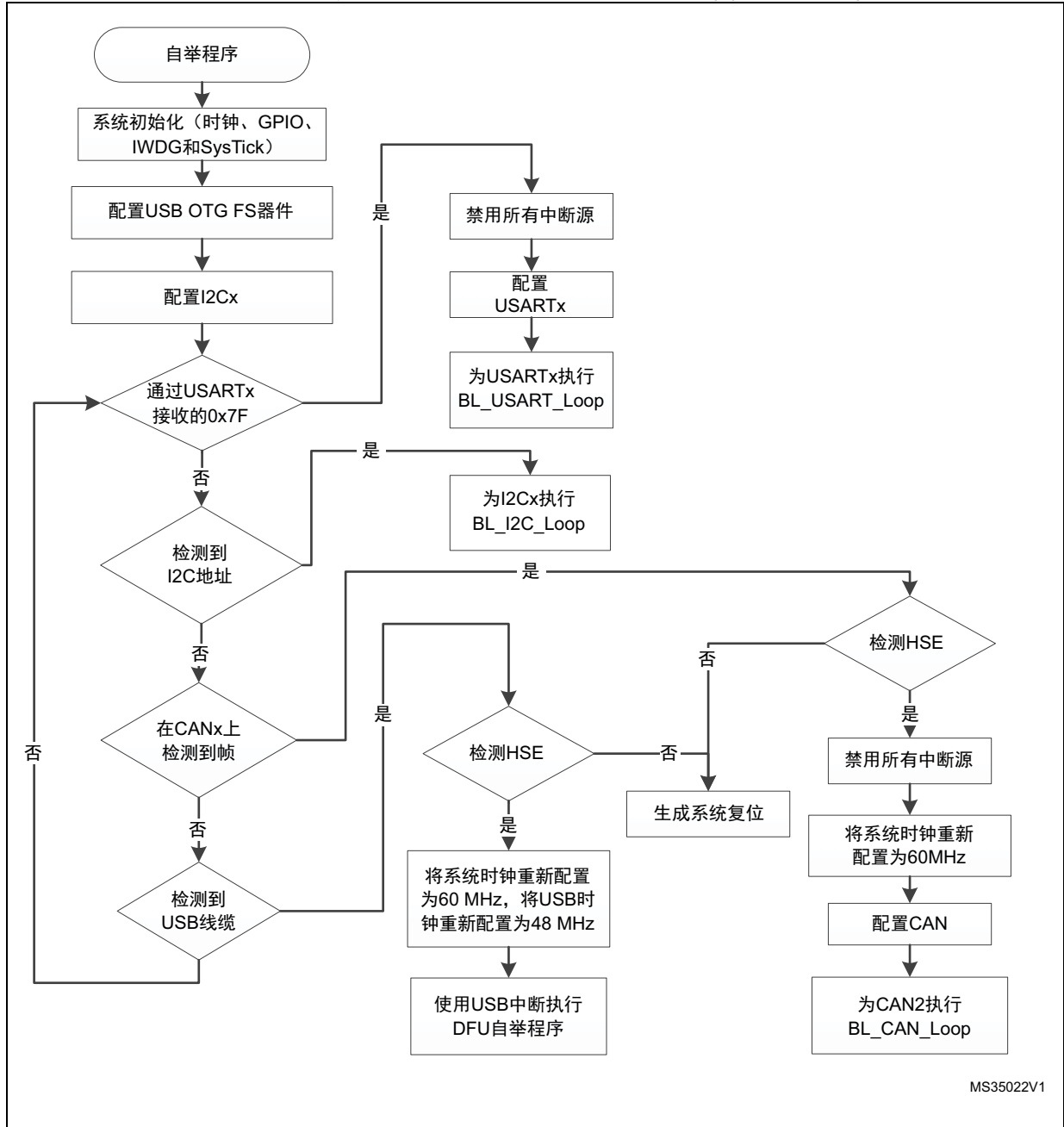
图40.STM32F42xxx/43xxx自举程序V7.x的双存储区自举实现



MS35021V1

1. 对于双存储区自举机制，CCM RAM不是有效的栈指针地址。

图41.STM32F42xxx/43xxx器件的自举程序V7.x选择



35.1.3 自举程序版本

下表列出了STM32F42xxx/43xxx器件自举程序V7.x版本。

表74.STM32F42xxx/43xxx自举程序V7.x版本

自举程序版本号	说明	已知限制
V7.0	初始自举程序版本	<p>对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。</p> <p>对于USBDFU接口，在双存储区模式中，擦除操作对于第二个存储区不起作用。返回至单存储区模式，擦除目标扇区，然后重新激活双存储区模式。</p> <p>在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。</p>

35.2 自举程序V9.x

35.2.1 自举程序配置

STM32F42xxx/43xxx自举程序通过应用模式5激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表75.系统存储器自举模式下STM32F42xxx/43xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。 HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择CAN或DFU自举程序后，时钟源由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。 仅当选择CAN或DFU（USB FS设备）接口时才使用HSE时钟源。 外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。

表75.系统存储器自举模式下STM32F42xxx/43xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125kbps，11位标识符。 注意：由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111000x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111000x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b0111000x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表75.系统存储器自举模式下STM32F42xxx/43xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

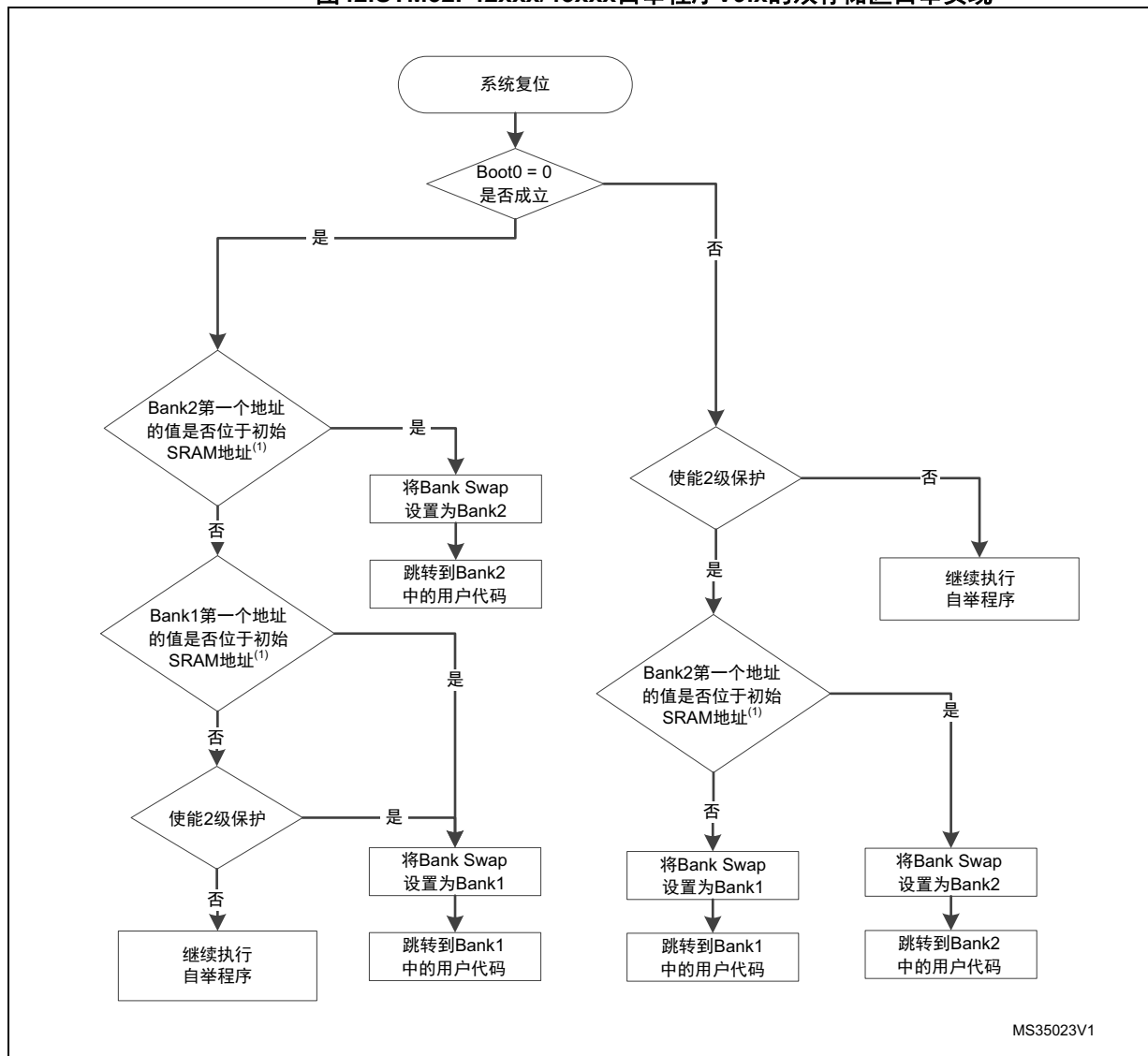
对于USARTx、I2Cx和SPIx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

35.2.2 自举程序选择

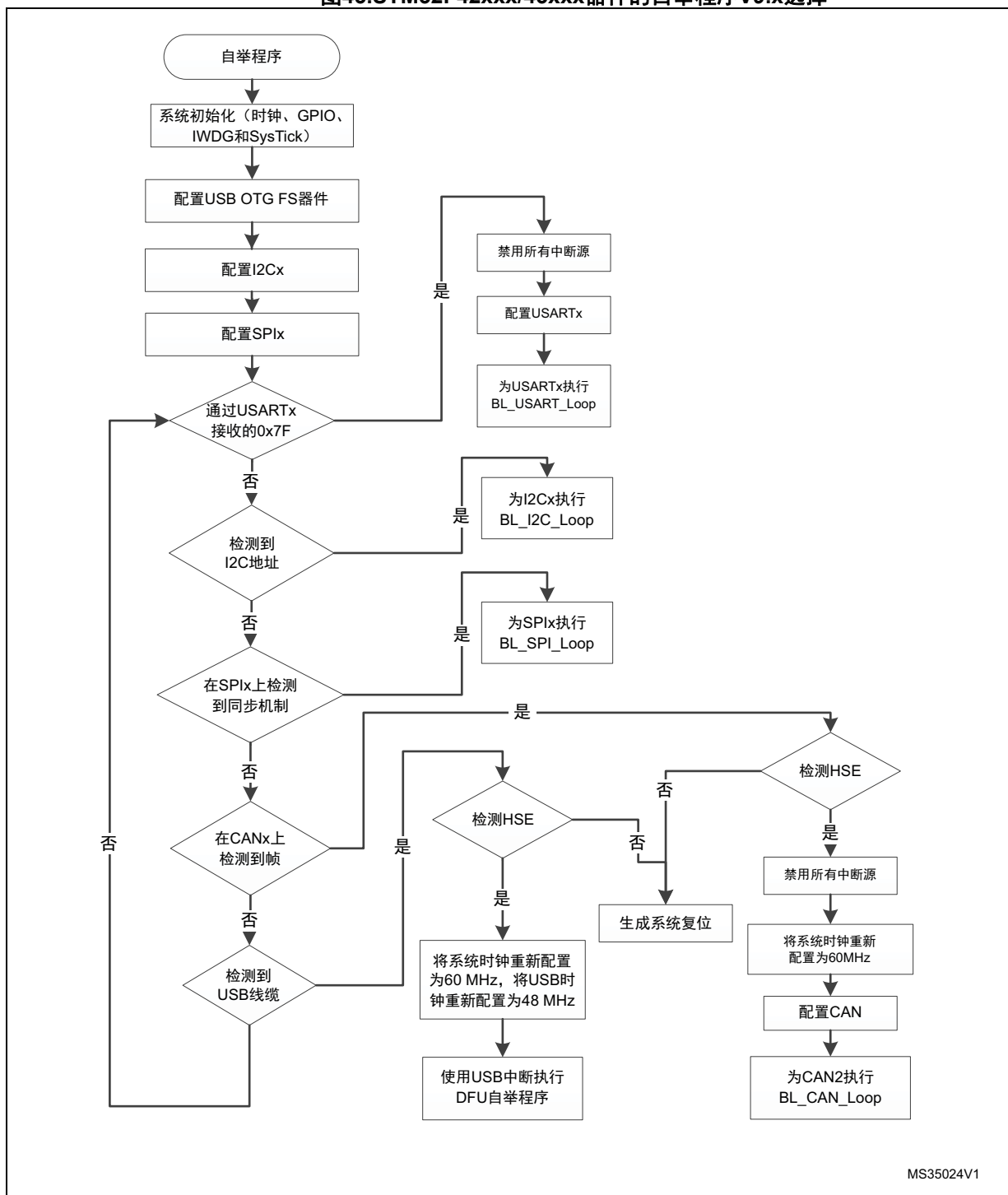
图 42和图 43显示了自举程序选择机制。

图42.STM32F42xxx/43xxx自举程序V9.x的双存储区自举实现



1. 对于双存储区自举机制，CCM RAM不是有效的栈指针地址。

图43.STM32F42xxx/43xxx器件的自举程序V9.x选择



MS35024V1

35.2.3 自举程序版本

表 76列出了STM32F42xxx/43xxx器件自举程序V9.x版本。

表76.STM32F42xxx/43xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	此自举程序是自举程序v7.0的更新版本。 该新版本自举程序支持I2C2、I2C3、SPI1、SPI2和SPI4接口。此自举程序所用的RAM从8 Kb增加到了12 KB。 此自举程序的ID为0x90 连接时间增加了。	对于USBDFU接口，在双存储区模式中，擦除操作对于第二个存储区不起作用。返回至单存储区模式，擦除目标扇区，然后重新激活双存储区模式。 在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。
V9.1	此自举程序是自举程序v9.0的更新版本。此新版本实现了新的I2C No-stretch命令（I2C协议v1.1），并且对于所有协议（USB、USART、CAN、I2C和SPI），能够在RDP1使能时利用ReadOutUnprotect命令禁用PcROP。此自举程序的ID为0x91	对于CAN接口，Write Unprotect命令不起作用。使用Write Memory命令，并直接写入选项字节以禁用写保护。 对于USBDFU接口，在双存储区模式中，擦除操作对于第二个存储区不起作用。返回至单存储区模式，擦除目标扇区，然后重新激活双存储区模式。 在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

36 STM32F446xx器件自举程序

36.1 自举程序配置

STM32F446xx自举程序通过应用模式1激活（详述参见表2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表77.系统存储器自举模式下STM32F446xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART、I2C和SPI自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围为 [1.71 V, 3.6 V]。 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - 禁用预读取。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表77.系统存储器自举模式下STM32F446xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0111100x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0111100x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表77.系统存储器自举模式下STM32F446xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0111100x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PC7引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。

表77.系统存储器自举模式下STM32F446xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
CAN2和DFU自举程序	TIM17	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

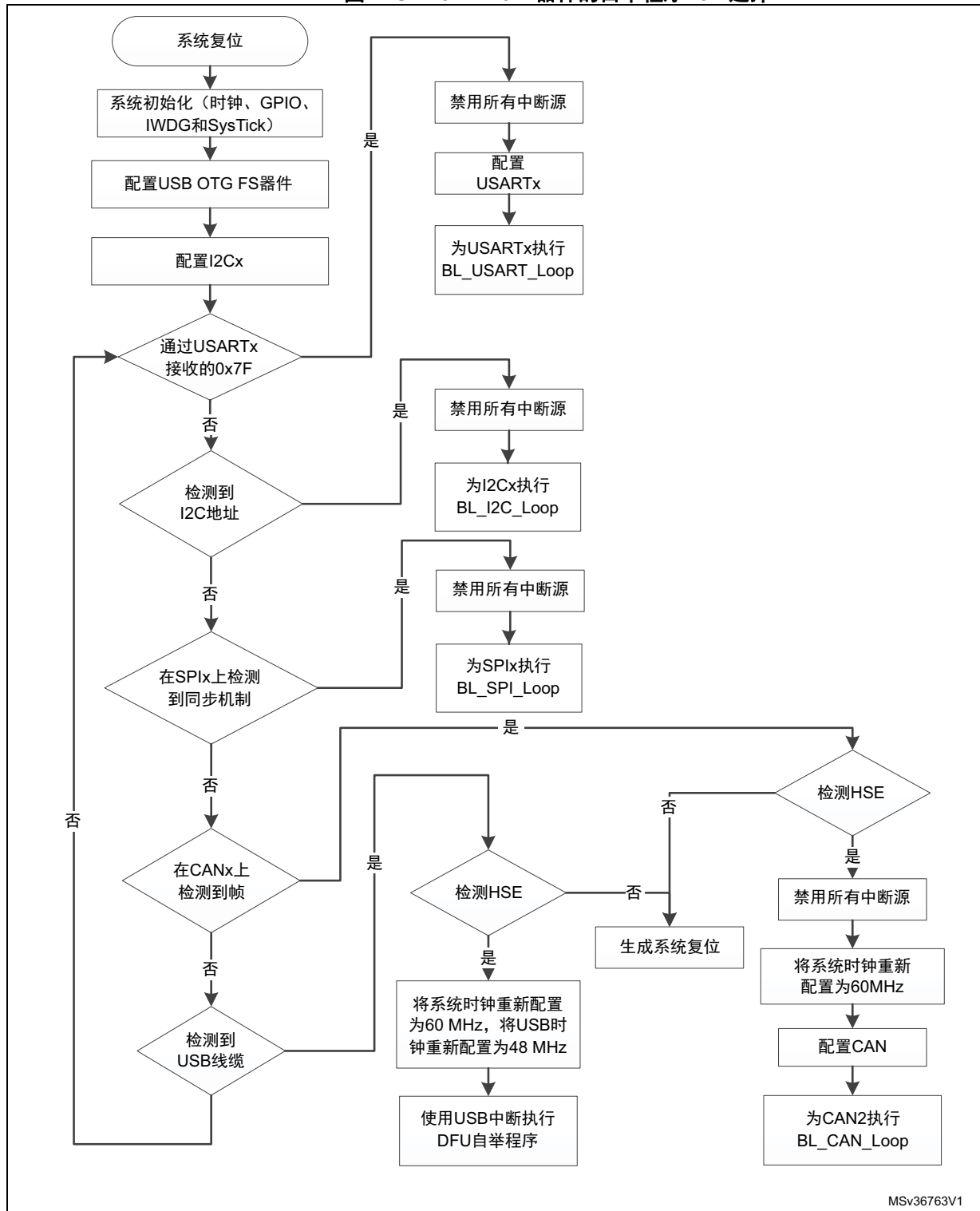
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

36.2 自举程序选择

下图显示了自举程序选择机制。

图44.STM32F446xx器件的自举程序V9.x选择



36.3 自举程序版本

下表列出了STM32F446xx器件自举程序V9.x版本：

表78.STM32F446xx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x00000000，因此CCMRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

37 STM32F469xx/479xx器件自举程序

37.1 自举程序配置

STM32F469xx/479xx自举程序通过应用模式5激活（详述参见表 2：自举程序激活模式）。表 79显示了该自举程序使用的硬件资源。

表79.系统存储器自举模式下STM32F469xx/479xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用PLL作为系统时钟，频率为60 MHz。HSI时钟源用于启动阶段（接口检测阶段）和选择USART、SPI或I2C接口期间（选择CAN或DFU自举程序后，时钟源将由外部晶振提供）。
		HSE使能	系统时钟频率为60 MHz。仅当选择CAN或DFU（USB FS设备）接口时才使用HSE时钟源。外部时钟提供的频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的29 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为 [1.62 V, 2.1 V]。此范围内仅支持按照字节格式（不支持半字、字和双字操作）对内部Flash执行写操作。可使用自举程序命令在运行期间配置电压范围。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。

表79.系统存储器自举模式下STM32F469xx/479xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB05引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000100x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000100x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PF0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF1引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000100x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。

表79.系统存储器自举模式下STM32F469xx/479xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SP4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式。使能USB_OTG_FS中断向量以用于USBDFU通信。
	USB_DM引脚	输入/输出	PA11引脚：USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。

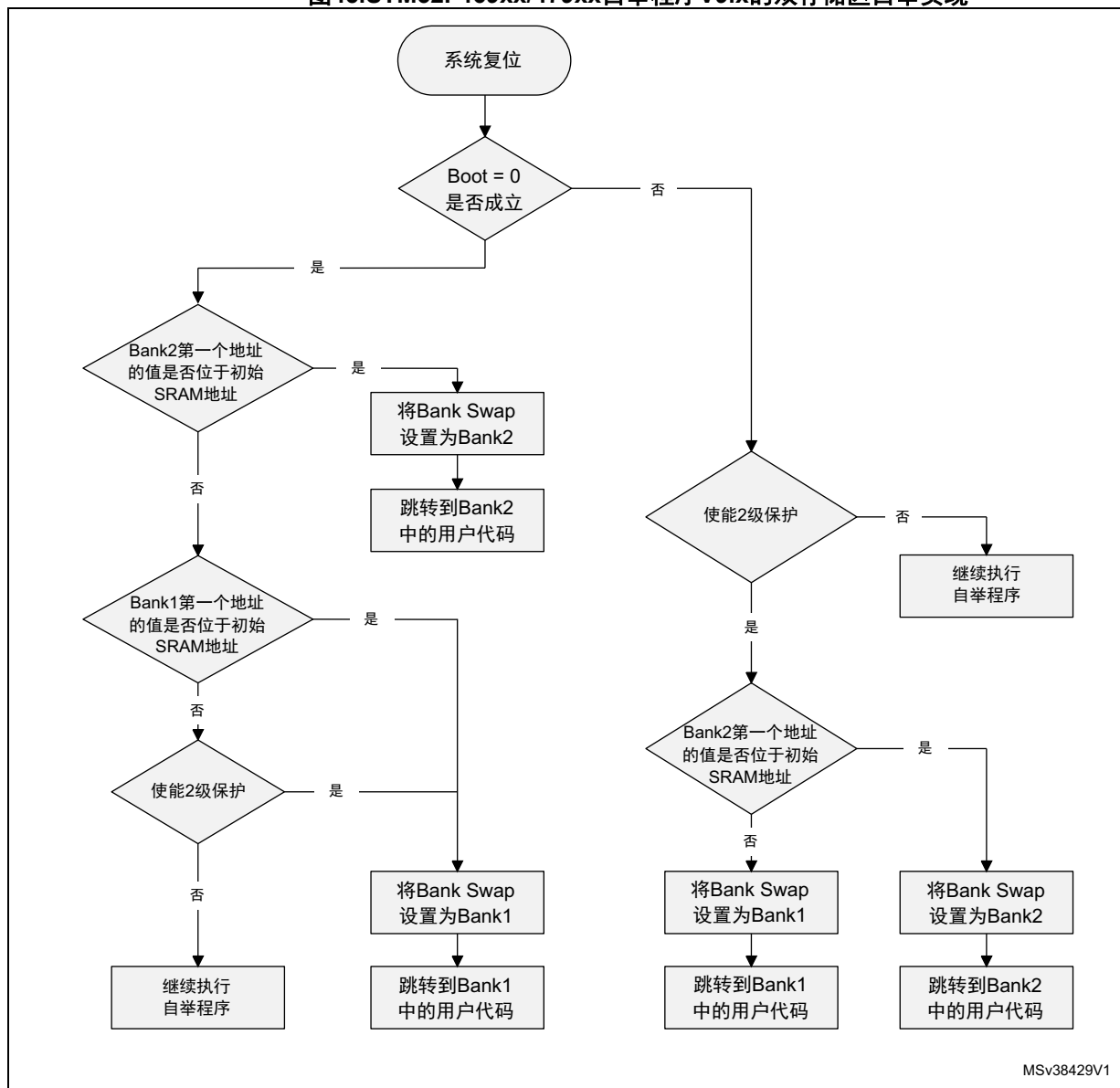
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USB FS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到48 MHz之间）的外部时钟。

注：由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

37.2 自举程序选择

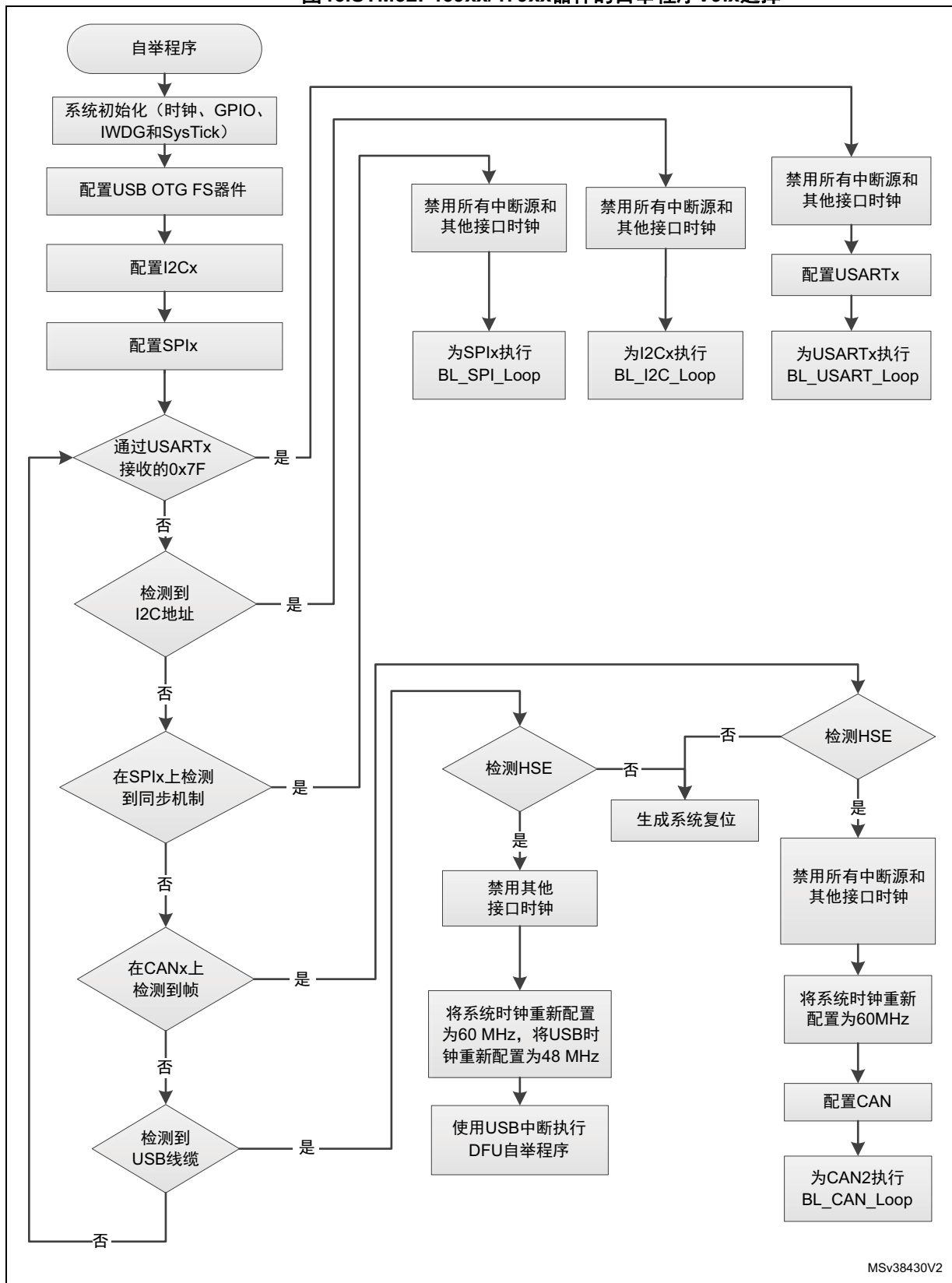
图 45和图 46显示了自举程序选择机制。

图45.STM32F469xx/479xx自举程序V9.x的双存储区自举实现



MSv38429V1

图46.STM32F469xx/479xx器件的自举程序V9.x选择



MSv38430V2



37.3 自举程序版本

表 80列出了STM32F469xx/479xx器件V9.x自举程序版本：

表80.STM32F469xx/479xx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	在执行Go命令（跳转到用户代码）后，自举程序将AHB1ENR值重置为0x0000000，因此CCRAM（如果有）未激活（必须在启动时通过用户代码重新启用）。

38 STM32F72xxx/73xxx器件自举程序

38.1 自举程序配置

STM32F72xxx/73xxx自举程序通过应用模式8激活（详述参见表 2：自举程序激活模式）。表 81显示了该自举程序使用的硬件资源。

表81.系统存储器自举模式下STM32F72xxx/73xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	使能时钟安全系统（CSS）中断以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的59 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表81.系统存储器自举模式下STM32F72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB11/PB10)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC11/PC10)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为：波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PD0引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PD1引脚：CAN1处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001001x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001101x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表81.系统存储器自举模式下STM32F72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001001x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。

表81.系统存储器自举模式下STM32F72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SP4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。
DFU自举程序	USB：	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11引脚：USBM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USBP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。
CAN1和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

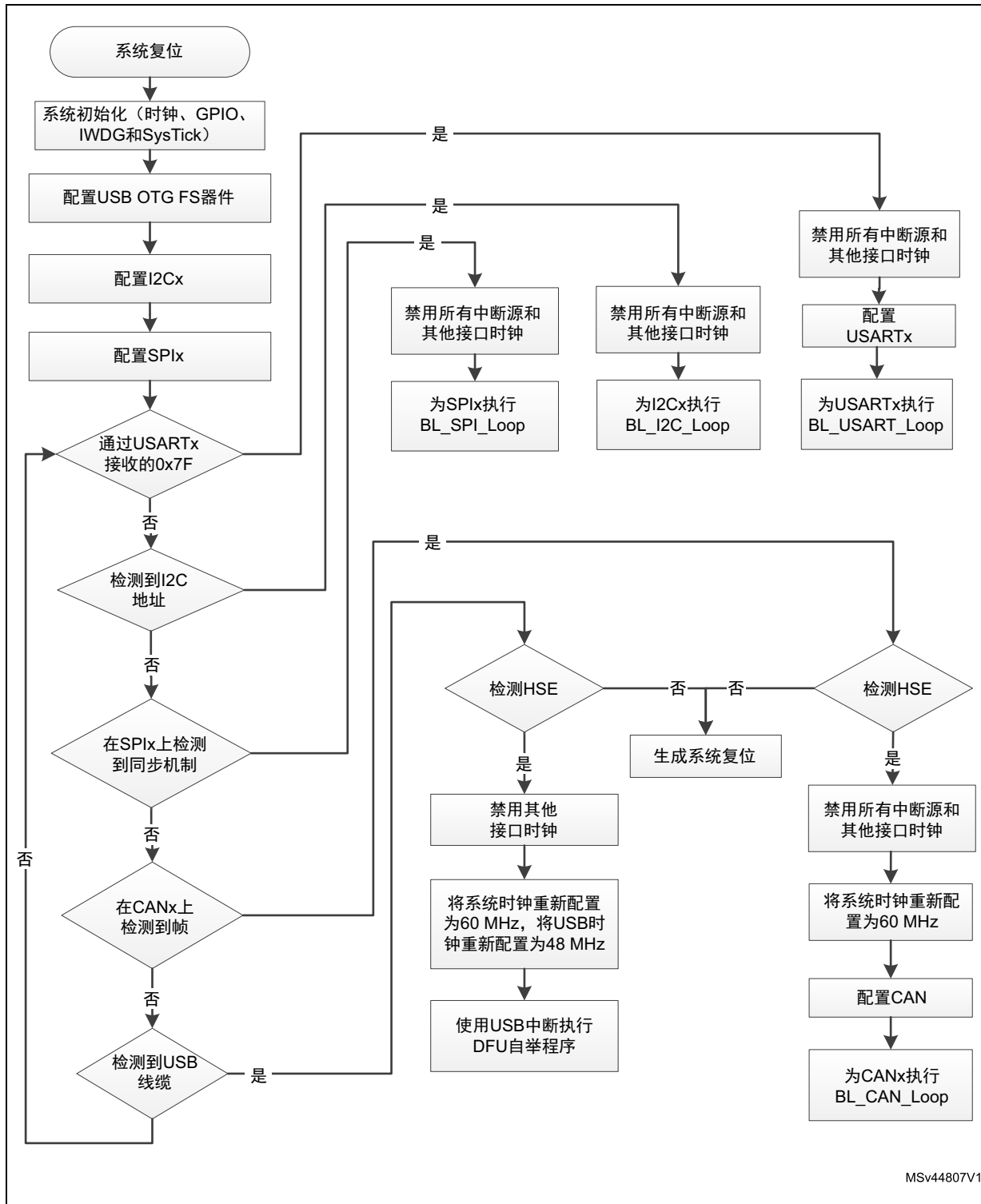
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USB FS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

38.2 自举程序选择

图 47显示了自举程序选择机制。

图47.STM32F72xxx/73xxx器件的自举程序V9.x选择



MSv44807V1



38.3 自举程序版本

表 82列出了STM32F72xxx/73xxx器件自举程序V9.x版本。

表82.STM32F72xxx/73xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	<p>由于软件抖动导致波特率计算错误，高UART波特率（115200 bps）连接可能失败。</p> <p>在这种情况下，自举程序可能以与主机波特率相差至多$\pm 5\%$的波特率进行响应。</p> <p>解决方案：如果主机的波特率误差容限低于$\pm 5\%$，则使用低于57600 bps的波特率。</p>

39 STM32F74xxx/75xxx器件自举程序

STM32F74xxx/75xxx器件可使用两种自举程序版本：

- V7.x支持USART1, USART3, CAN2, I2C1, I2C2, I2C3和DFU (USB FS设备)。该版本嵌入STM32F74xxx/75xxx rev.A器件。
- V9.x支持USART1, USART3, CAN2, I2C1, I2C2, I2C3, SPI1, SPI2, SPI4和DFU (USB FS设备)。该版本嵌入STM32F74xxx/75xxx rev.Z和rev.1类器件。

注：当读出保护Level2激活时，STM32F74xxx/75xxx器件也可在系统存储器上自举，除Get、GetID和GetVersion之外的所有指令都无法访问。

39.1 自举程序V7.x

39.1.1 自举程序配置

STM32F74xxx/75xxx自举程序通过应用模式8激活（详述参见表 2：自举程序激活模式）。表 83显示了该自举程序使用的硬件资源。

表83.系统存储器自举模式下STM32F74xxx/75xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU (USB FS设备) 接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	时钟安全系统 (CSS) 中断使能以用于CAN和DFU自举程序。外部时钟发生任何故障 (或被移除) 都将产生系统复位。 外部时钟生成系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的60 KB空间包含自举程序固件
	IWDG	-	独立看门狗 (IWDG) 预分频器配置为最大值，并定期进行刷新以防止看门狗复位 (如果用户事先使能硬件IWDG选项)。
	电源	-	电压范围为 [1.8V, 3.6V]。在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行 (请参考自举程序存储器管理章节来获取更多信息)。

表83.系统存储器自举模式下STM32F74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表83.系统存储器自举模式下STM32F74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式。
	USB_DM引脚	输入/输出	PA11引脚：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

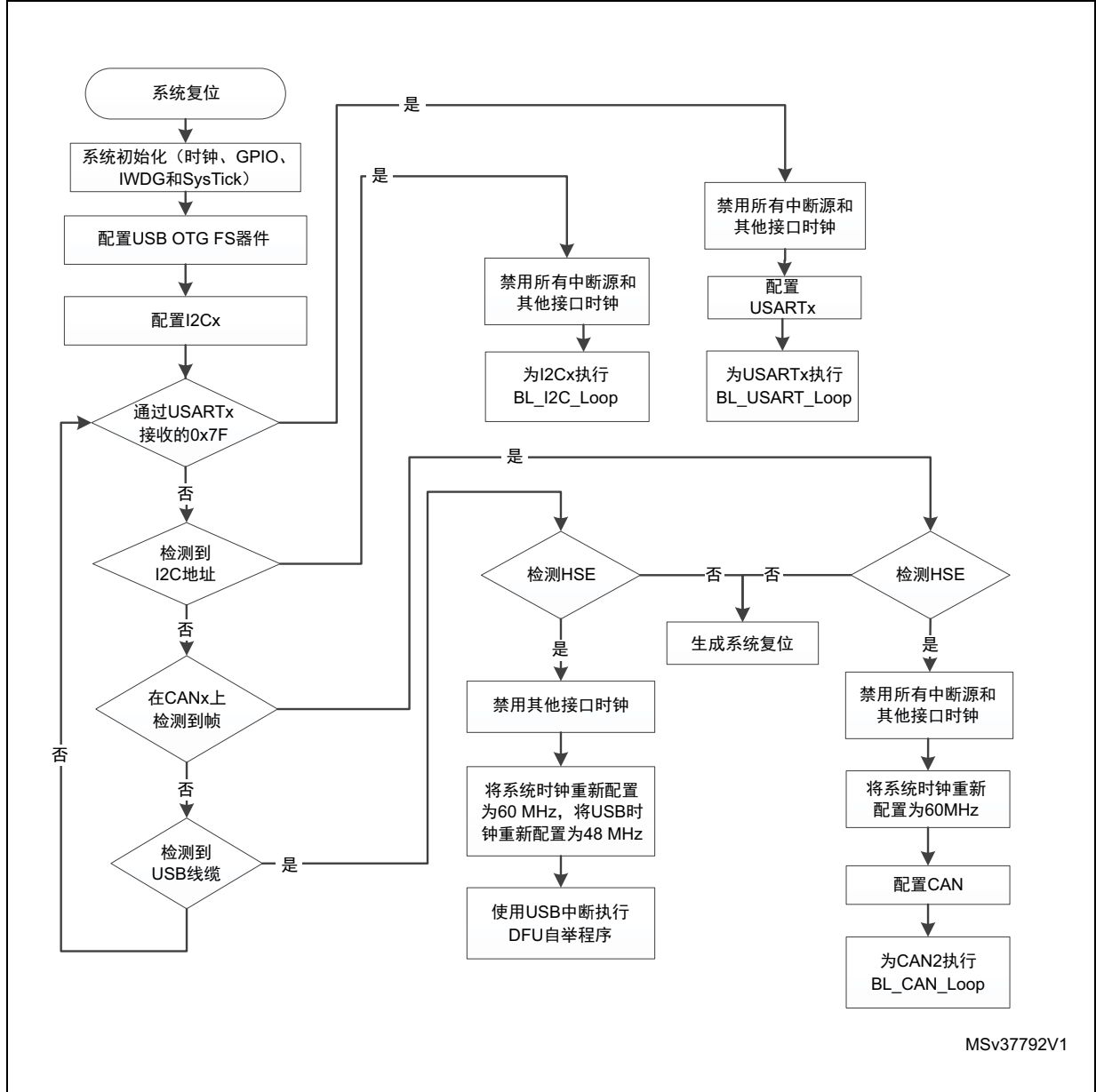
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

39.1.2 自举程序选择

下图显示了自举程序选择机制。

图48.STM32F74xxx/75xxx器件的自举程序V7.x选择



39.1.3 自举程序版本

下表列出了STM32F74xxx/75xxx器件自举程序V7.x版本：

表84.STM32F74xxx/75xxx自举程序V7.x版本

自举程序版本号	说明	已知限制
V7.0	初始自举程序版本	由于软件抖动导致波特率计算错误，高UART波特率（115200 bps）连接可能失败。 在这种情况下，自举程序可能以与主机波特率相差至多± 5%的波特率进行响应。 解决方案：如果主机的波特率误差容限低于± 5%，则使用低于57600 bps的波特率。

39.2 自举程序V9.x

39.2.1 自举程序配置

STM32F74xxx/75xxx自举程序通过应用模式8激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表85.系统存储器自举模式下STM32F74xxx/75xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART、I2C和SPI自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	时钟安全系统（CSS）中断使能以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。 外部时钟生成系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的60 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表85.系统存储器自举模式下STM32F74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC10/PC11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表85.系统存储器自举模式下STM32F74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000101x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。

表85.系统存储器自举模式下STM32F74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式。
	USB_DM引脚	输入/输出	PA11引脚: USB DM线。用于复用推挽, 无上拉/下拉模式。
	USB_DP引脚		PA12引脚: USB DP线。用于复用推挽, 无上拉/下拉模式。无需外部上拉电阻。
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率, 可利用PLL和HSE将系统时钟配置为60 MHz。

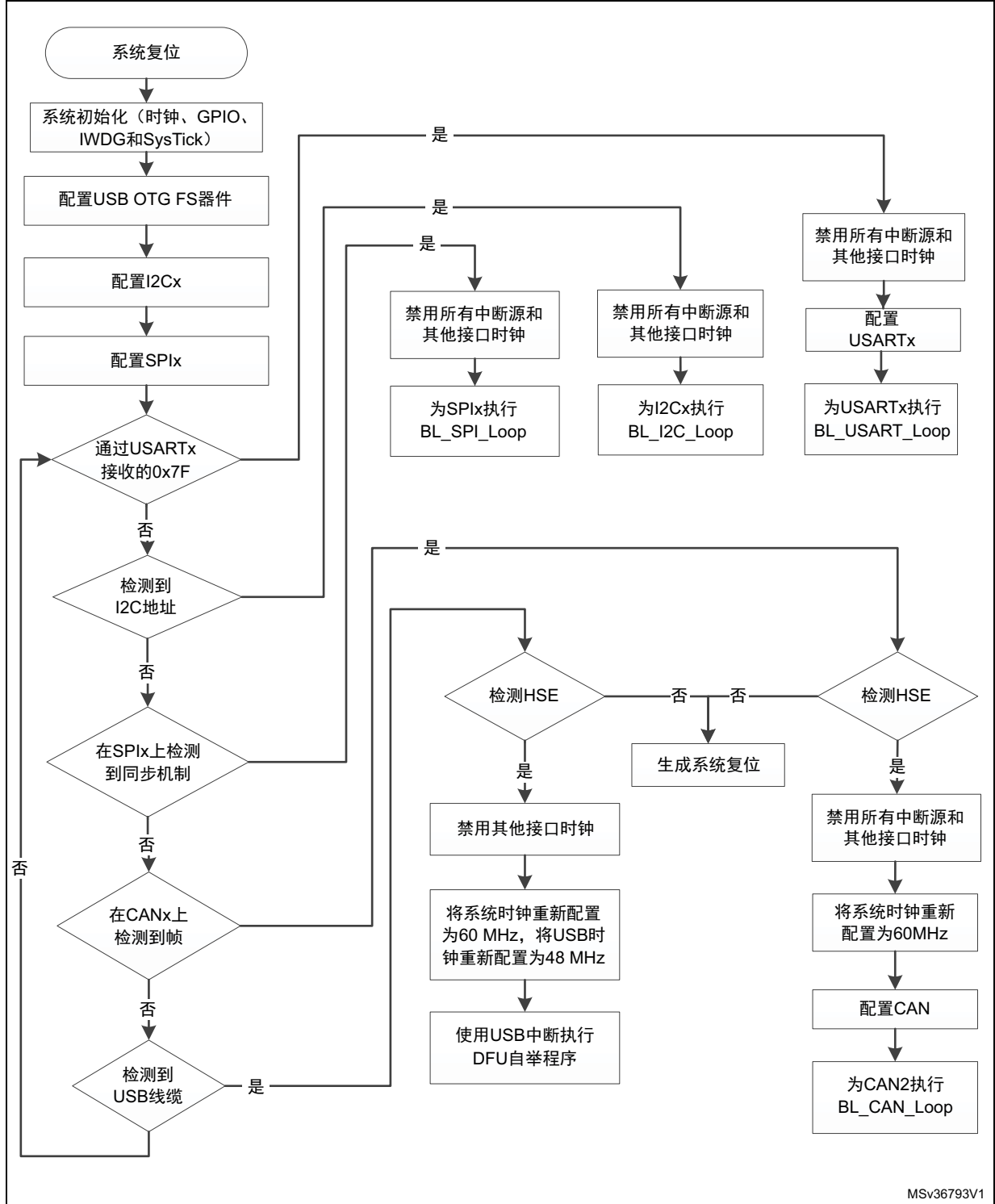
对于USARTx、I2Cx和SPIx自举程序, 系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU (USBFS设备), 但仅限选择阶段。选择阶段结束后, 执行CAN和DFU自举程序需要使用频率为1 MHz的倍数 (介于4 MHz到26 MHz之间) 的外部时钟。

注: 由于存在HSI偏差且HSI被用于检测HSE值, 用户必须使用低频而不是高频HSE晶振值 (低频值的误差容限更大, 因此更好检测)。例如, 最好使用8 MHz, 而不是25 MHz。

39.2.2 自举程序选择

图 49显示了自举程序选择机制。

图49.STM32F74xxx/75xxx器件的自举程序V9.x选择



MSv36793V1



39.2.3 自举程序版本

下表列出了STM32F74xxx/75xxx器件自举程序V9.x版本：

表86.STM32F74xxx/75xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	<p>由于软件抖动导致波特率计算错误，高UART波特率（115200 bps）连接可能失败。</p> <p>在这种情况下，自举程序可能与主机波特率相差至多$\pm 5\%$的波特率进行响应。</p> <p>解决方案：如果主机的波特率误差容限低于$\pm 5\%$，则使用低于57600 bps的波特率。</p>

40 STM32F76xxx/77xxx器件自举程序

40.1 自举程序配置

STM32F76xxx/77xxx自举程序通过应用模式9激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表87.系统存储器自举模式下STM32F76xxx/77xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	仅当选择CAN或DFU（USBFS设备）接口时才使用HSE。在此情况下，使用HSE作为时钟源并将系统时钟配置为60 MHz。 HSE频率必须为1 MHz的倍数，且范围介于4 MHz到26 MHz之间。
		-	时钟安全系统（CSS）中断使能以用于CAN和DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。 外部时钟生成系统复位。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的59 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围为 [1.8V, 3.6V] 在此范围内： - 3种Flash等待状态。 - 系统时钟60 MHz。 - ART加速器使能。 - Flash写操作按字节执行（请参考自举程序存储器管理章节来获取更多信息）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表87.系统存储器自举模式下STM32F76xxx/77xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB11/PB10)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于输入上拉模式。
USART3自举程序 (PC11/PC10)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
CAN2自举程序	CAN2	启用	初始化后，CAN2的配置为：波特率125 kbps，11位标识符。 注意： 由于CAN1管理着CAN2和SRAM之间的通信，因此执行CAN2自举程序期间要为CAN1提供时钟。
	CAN2_RX引脚	输入	PB5引脚：CAN2处于接收模式。用于复用推挽，上拉模式。
	CAN2_TX引脚	输出	PB13引脚：CAN2处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001001x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001001x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。

表87.系统存储器自举模式下STM32F76xxx/77xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001001x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式
	SP4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽下拉模式。

表87.系统存储器自举模式下STM32F76xxx/77xxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11引脚：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。
CAN2和DFU自举程序	TIM11	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

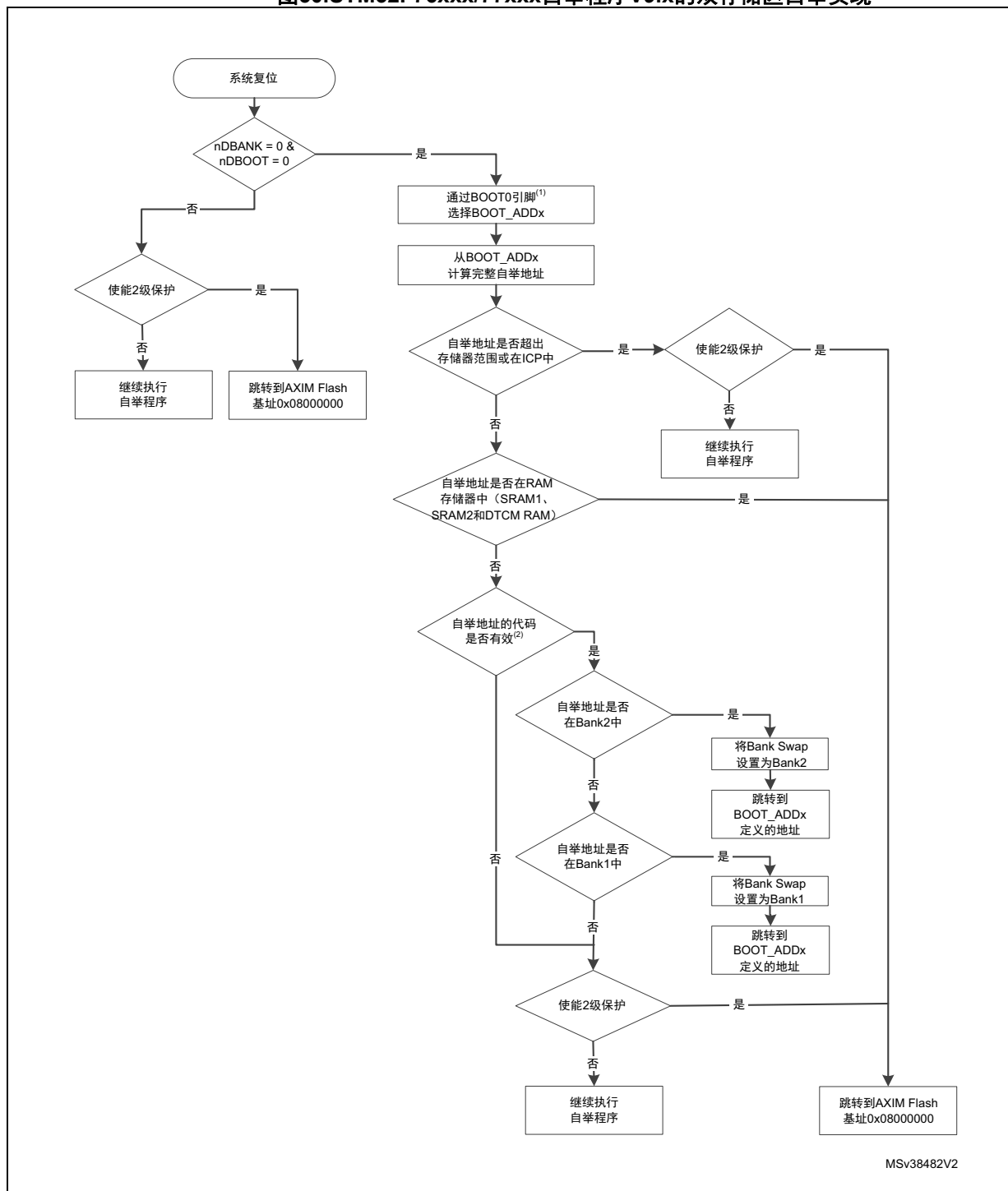
对于USARTx和I2Cx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于CAN和DFU（USBFS设备），但仅限选择阶段。选择阶段结束后，执行CAN和DFU自举程序需要使用频率为1 MHz的倍数（介于4 MHz到26 MHz之间）的外部时钟。

注： 由于存在HSI偏差且HSI被用于检测HSE值，用户必须使用低频而不是高频HSE晶振值（低频值的误差容限更大，因此更好检测）。例如，最好使用8 MHz，而不是25 MHz。

40.2 自举程序选择

图 50和图 51显示了自举程序选择机制。

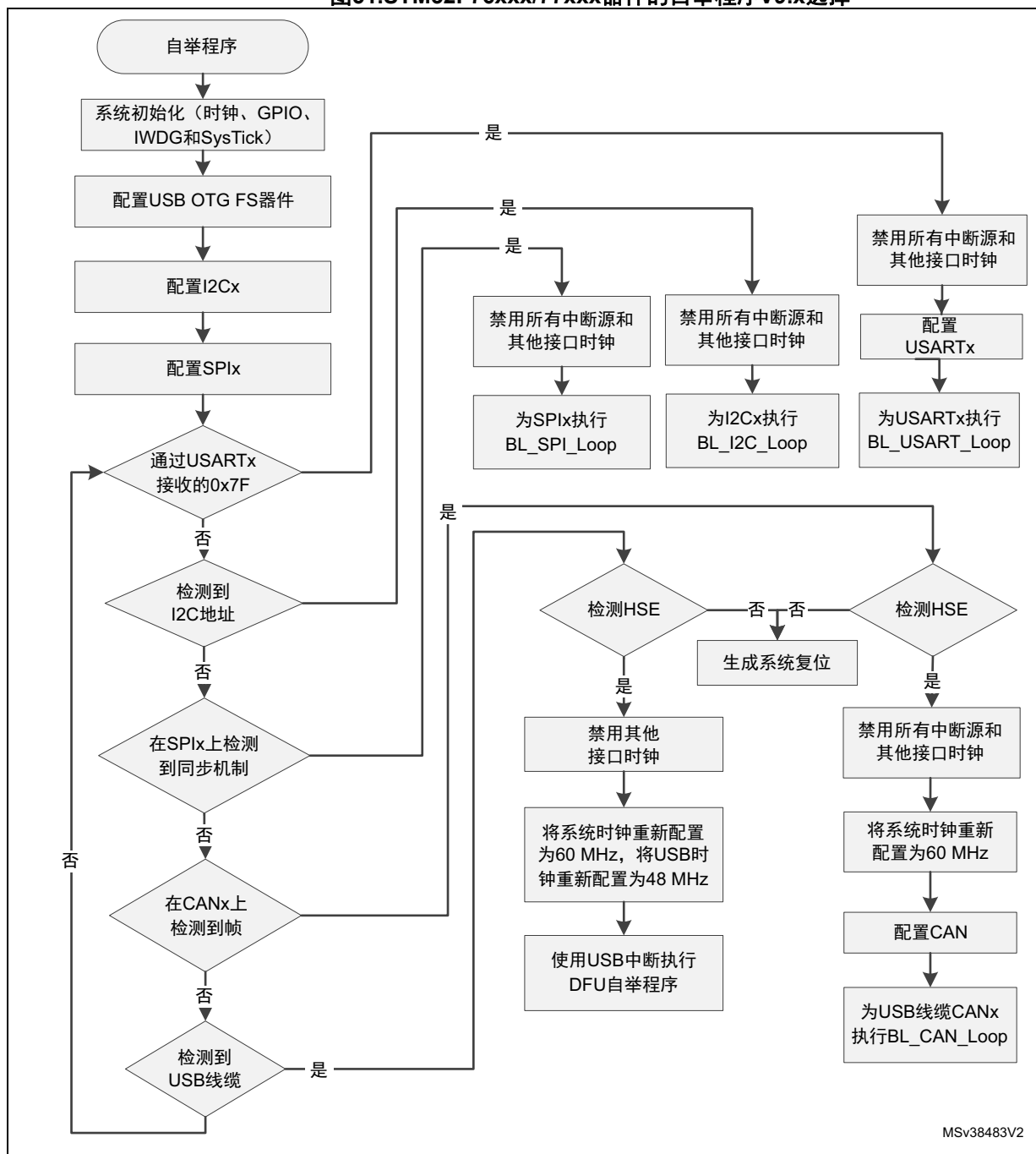
图50.STM32F76xxx/77xxx自举程序V9.x的双存储区自举实现



1. 如表 88下的已知限制所述，不管BOOT0引脚状态如何，都只会考虑BOOT_ADD0的值。
2. 对于双存储区自举机制，ITCM RAM不是有效的栈指针地址。



图51.STM32F76xxx/77xxx器件的自举程序V9.x选择



MSv38483V2

40.3 自举程序版本

下表列出了STM32F76xxx/77xxx器件自举程序V9.x版本。

表88.STM32F76xxx/77xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.3	初始自举程序版本	<p>当Flash存储器配置为双存储区自举模式（nDBANK=nDBOOT=0）时，不管BOOT0引脚状态如何，都仅会考虑BOOT_ADD0的值（当BOOT0引脚=1时，BOOT_ADD0的值不是BOOT_ADD1）。</p> <p>解决方案：为管理仅有BOOT_ADD0的双存储区自举，请参考AN4826：“STM32F7系列Flash存储器双存储区模式”</p> <p>由于软件抖动导致波特率计算错误，高UART波特率（115200 bps）连接可能失败。</p> <p>在这种情况下，自举程序可能以与主机波特率相差至多± 5%的波特率进行响应。</p> <p>解决方案：如果主机的波特率误差容限低于± 5%，则使用低于57600 bps的波特率。</p> <p>使用USB接口时的Bank2扇区擦除问题。擦除Bank2中具有索引 (i) 的扇区将导致扇区 (i+4) 被擦除。</p>

41 STM32G03xxx/STM32G04xxx器件自举程序

41.1 自举程序配置

STM32G03xxx/G04xxx自举程序通过应用模式11激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。注意，STM32G030x没有BOOT_LOCK(bit)，在使用模式11时应考虑这一点。

表89.系统存储器自举模式下STM32G03xxx/G04xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于24 MHz（使用由HSI驱动的PLL）。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的8 KB空间
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址：@0x1FFF1D00
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010110x （其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。

表89.系统存储器自举模式下STM32G03xxx/G04xxx器件的配置（续）

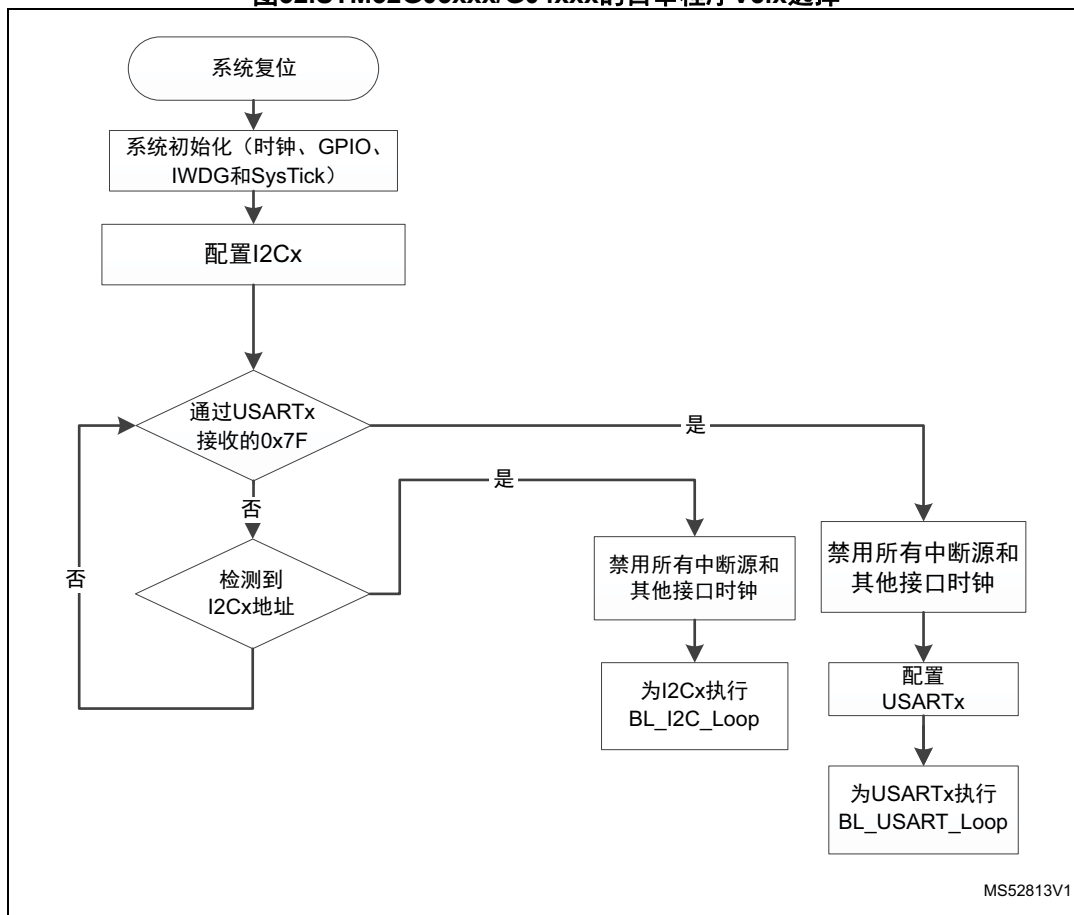
自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010110x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。

注：在SO8、WLCSP18、TSSOP20和UFQFN28封装中，USARTPA9/PA10在PA11/PA12上重映射。

41.2 自举程序选择

图 52显示了自举程序选择机制。

图52.STM32G03xxx/G04xxx的自举程序V5.x选择



41.3 自举程序版本

表 90列出了STM32G03xxx/G04xxx器件自举程序版本。

表90.STM32G03xx/04xxx自举程序版本

自举程序版本号	说明	已知限制
V5.1	初始自举程序版本	<ul style="list-style-type: none"> - 只支持48和32引脚封装 - 问题同时见于两种封装，如果PA3保持低电平，系统将卡在USART2检测流程且不会检测到其他接口。
V5.2	增加了对小型封装8/20和28引脚的支持	问题见于所有封装（SO8除外，无PA3引脚），如果PA3保持低电平，系统将卡在USART2检测流程且不会检测到其他接口。
V5.3	修正V5.2限制	无

42 STM32G07xxx/08xxx器件自举程序

42.1 自举程序配置

STM32G07xxx/G08xxx自举程序通过应用模式11激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。注意，STM32G070x没有BOOT_LOCK(bit)，在使用模式11时应考虑这一点。

表91.系统存储器自举模式下STM32G07xxx/8xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于24 MHz（使用由HSI驱动的PLL）。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址： @0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

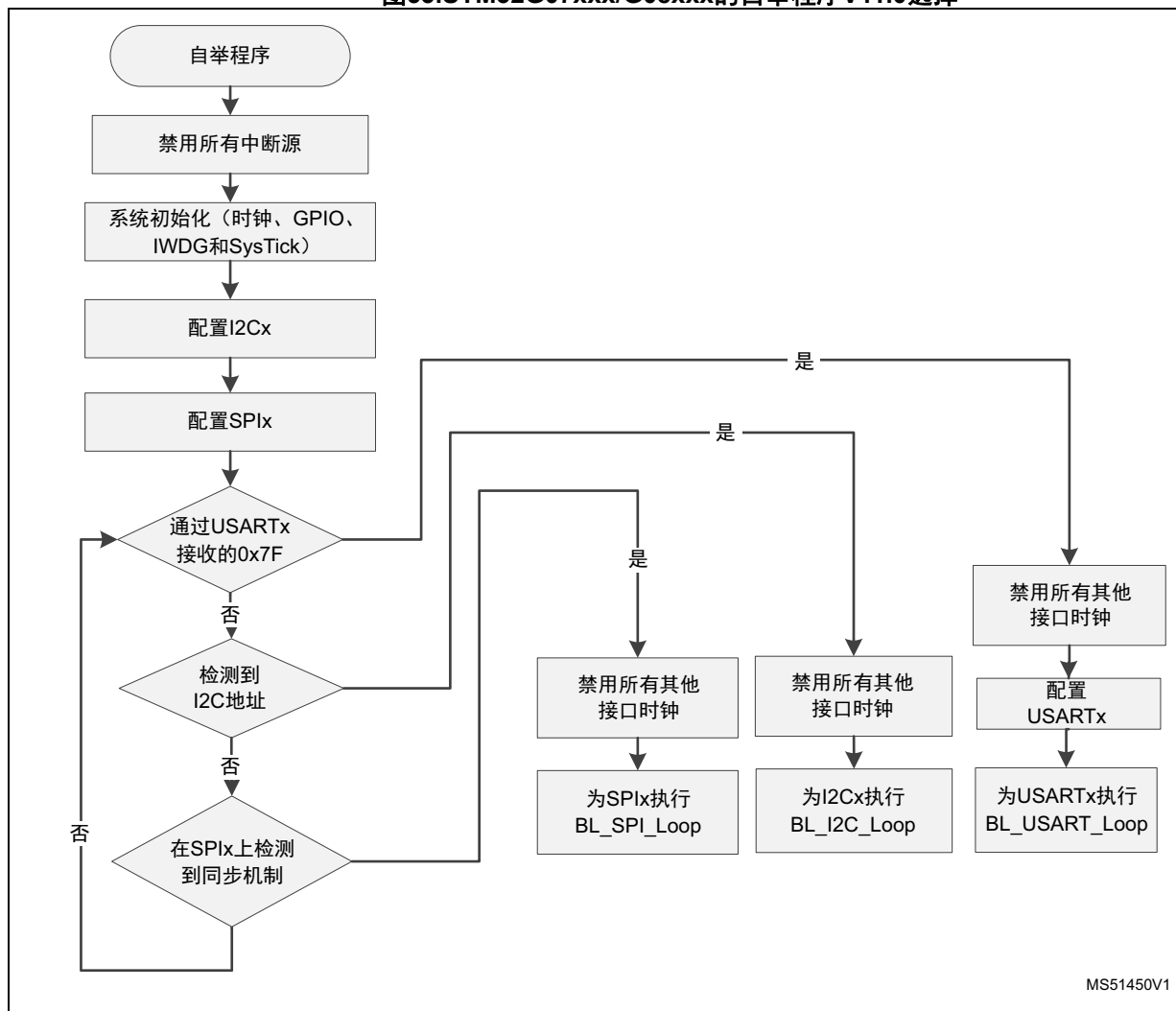
表91.系统存储器自举模式下STM32G07xxx/8xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010001x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010001x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到Gnd。

42.2 自举程序选择

图 53显示了自举程序选择机制。

图53.STM32G07xxx/G08xxx的自举程序V11.0选择



42.3 自举程序版本

表 92列出了STM32G07xxx/8xxx器件自举程序版本。

表92.STM32G07xx/08xxx自举程序版本

自举程序版本号	说明	已知限制
V11.0	初始自举程序版本	不支持小于LQFP64的封装
V11.1	支持所有封装	无
V11.2	增加了安全存储区特性	<ul style="list-style-type: none"> - 使用USART协议时选项字节启动缺失 - 在使用Go命令时，RCC寄存器RCC_ICSCR不置为默认值。HSITRIM值被置为默认值之外的另一个值。 - 在使用“Go”命令时，RCC寄存器不置为默认值（HSITRIM未正确复位）。 - 如果在使能SRAM奇偶校验选项字节前未初始化SRAM，使能此特性会导致自举程序崩溃。

43 STM32G0B0xx器件自举程序

43.1 自举程序配置

STM32G0B0xx自举程序通过应用模式11激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。注意，STM32G0B0x没有BOOT_LOCK(bit)，在使用模式11时应考虑这一点。

表93.系统存储器自举模式下STM32G0B0xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于60 MHz（使用由HSI驱动的PLL）。如果没有外部时钟（HSE），系统时钟将由HSI提供
		HSE使能	外部时钟可用于所有自举程序接口，其频率值应为如下之一：[48、32、16、12、8] MHz。PLL用来为USB和系统时钟产生48 MHz频率。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	以下两个存储区共享自举程序固件： - 自地址0x1FFF0000至0x1FFF6FFF的28 KB空间 - 28 KB（0x1FFF8000 – 0x1FFFEFFF）的一部分
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的出口安全存储区的地址：@0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表93.系统存储器自举模式下STM32G0B0xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011101x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011101x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到Gnd。

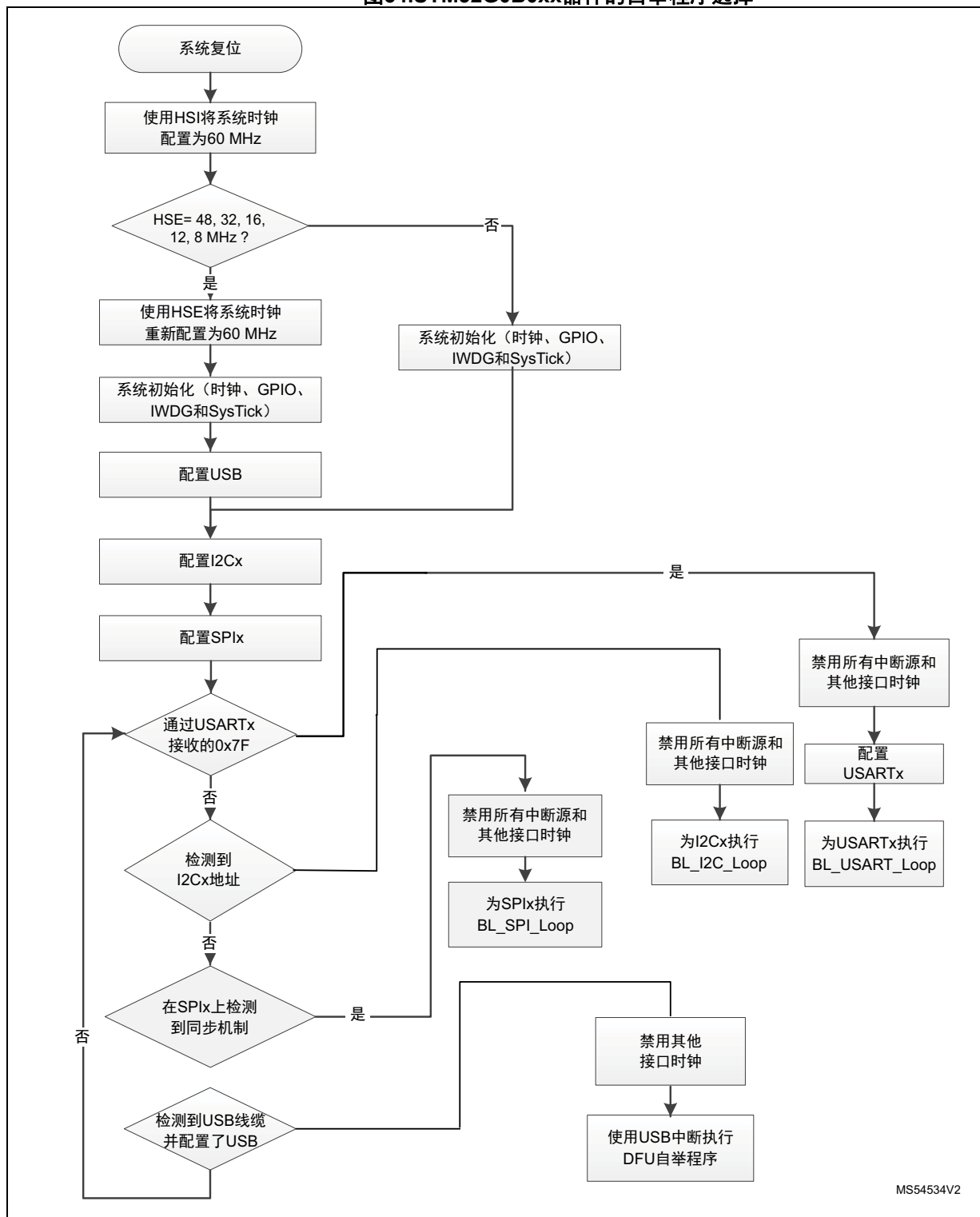
表93.系统存储器自举模式下STM32G0B0xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。使能USB FS中断向量以用于USBDFU通信。注意：由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。无需外部上拉电阻

43.2 自举程序选择

下图显示了自举程序选择机制。

图54.STM32G0B0xx器件的自举程序选择



43.3 自举程序版本

表 94列出了STM32G0B0xx器件自举程序版本。

表94.STM32G0B0xx自举程序版本

自举程序版本号	说明	已知限制
V13.0	初始自举程序版本	无

44 STM32G0B1xx/0C1xx器件自举程序

44.1 自举程序配置

STM32G0B1xx/0C1xx自举程序通过应用模式11激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表95.系统存储器自举模式下STM32G0B1xx/0C1xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于60 MHz（使用由HSI驱动的PLL）。
		-	对于DFU自举程序，使能时钟恢复系统（Clock RecoverySystem, CRS），支持USB以HSI48MHz作为时钟。
		-	对FDCAN使用PLLQ提供的20 Mhz时钟频率
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	以下两个存储区共享自举程序固件： - 自地址0x1FFF0000至0x1FFF6FFF的28 KB空间 - 28 KB（0x1FFF8000 – 0x1FFFEFFF）的一部分
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址： @0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表95.系统存储器自举模式下STM32G0B1xx/0C1xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011101x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011101x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到Gnd。

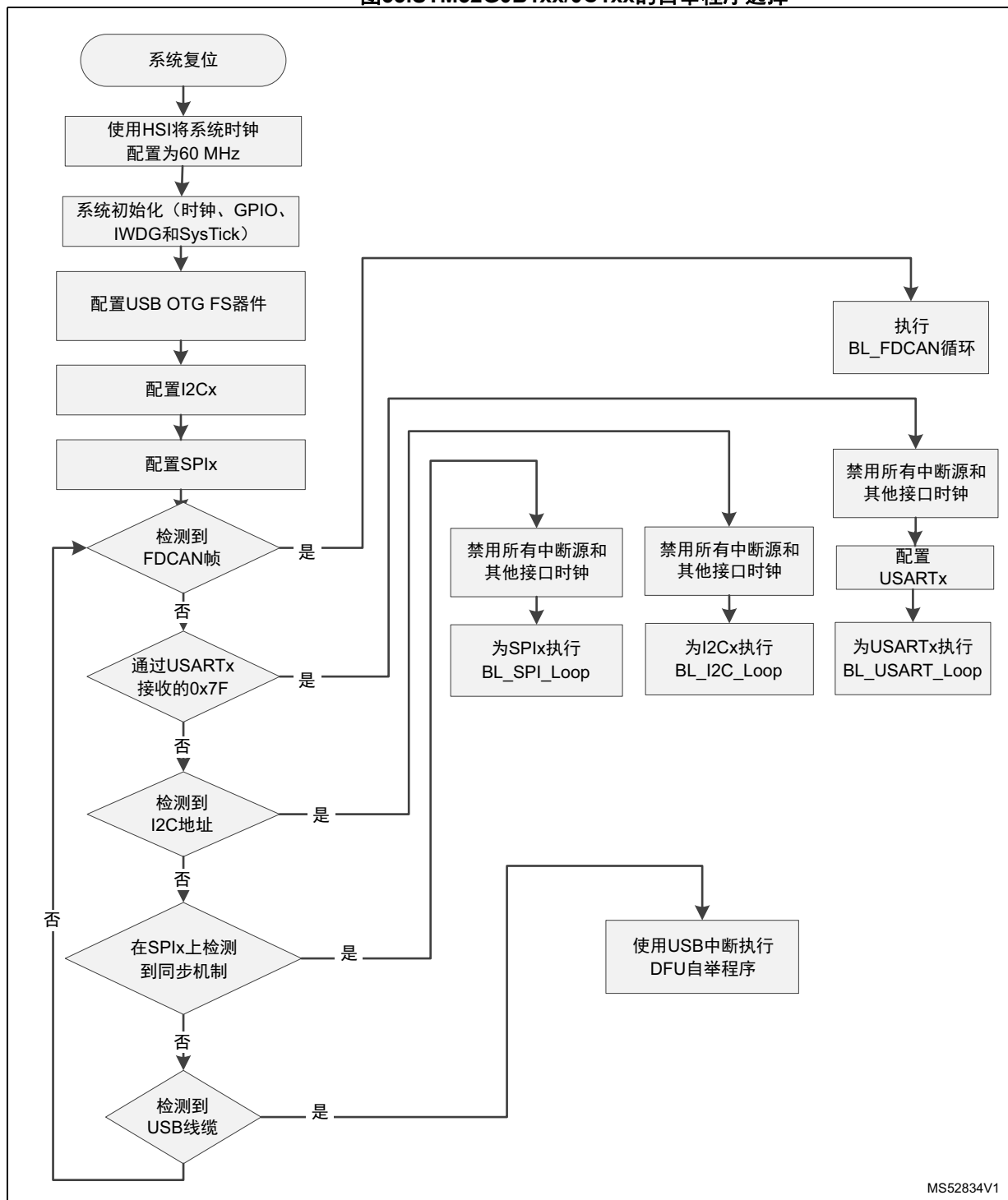
表95.系统存储器自举模式下STM32G0B1xx/0C1xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。使能USB FS中断向量以用于USBDFU通信。注意：由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于无上拉/下拉模式。 无需外部上拉电阻
FDCAN	FDCAN1	启用	初始化后，FDCAN1的配置为： -- 比特率为0.5 Mbps -- FrameFormat = FDCAN_FRAME_FD_BRS 模式 = FDCAN_MODE_NORMAL -- AutoRetransmission = 使能 -- TransmitPause = 禁用 -- ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PD0引脚：FDCAN1处于接收模式。用于复用推挽，上拉模式。
	FDCAN1_Tx引脚	输出	PD1引脚：FDCAN1处于发送模式。用于复用推挽，上拉模式。

44.2 自举程序选择

下图显示了自举程序选择机制。

图55.STM32G0B1xx/0C1xx的自举程序选择



MS52834V1



44.3 自举程序版本

表 96列出了STM32G0B1xx/0C1xx器件自举程序版本。

表96.STM32G0B1xx/0C1xx自举程序版本

自举程序版本号	说明	已知限制
V9.2	初始自举程序版本	无

45 STM32G05xxx/061xx器件自举程序

45.1 自举程序配置

STM32G05xxx/061xx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。注意，STM32G050x没有BOOT_LOCK(bit)，在使用模式6时应考虑这一点。

表97.系统存储器自举模式下STM32G05xxx/061xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于24MHz（使用由HSI驱动的PLL）。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的8 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的安全存储区的地址： @0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USART2自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1100010x （其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。

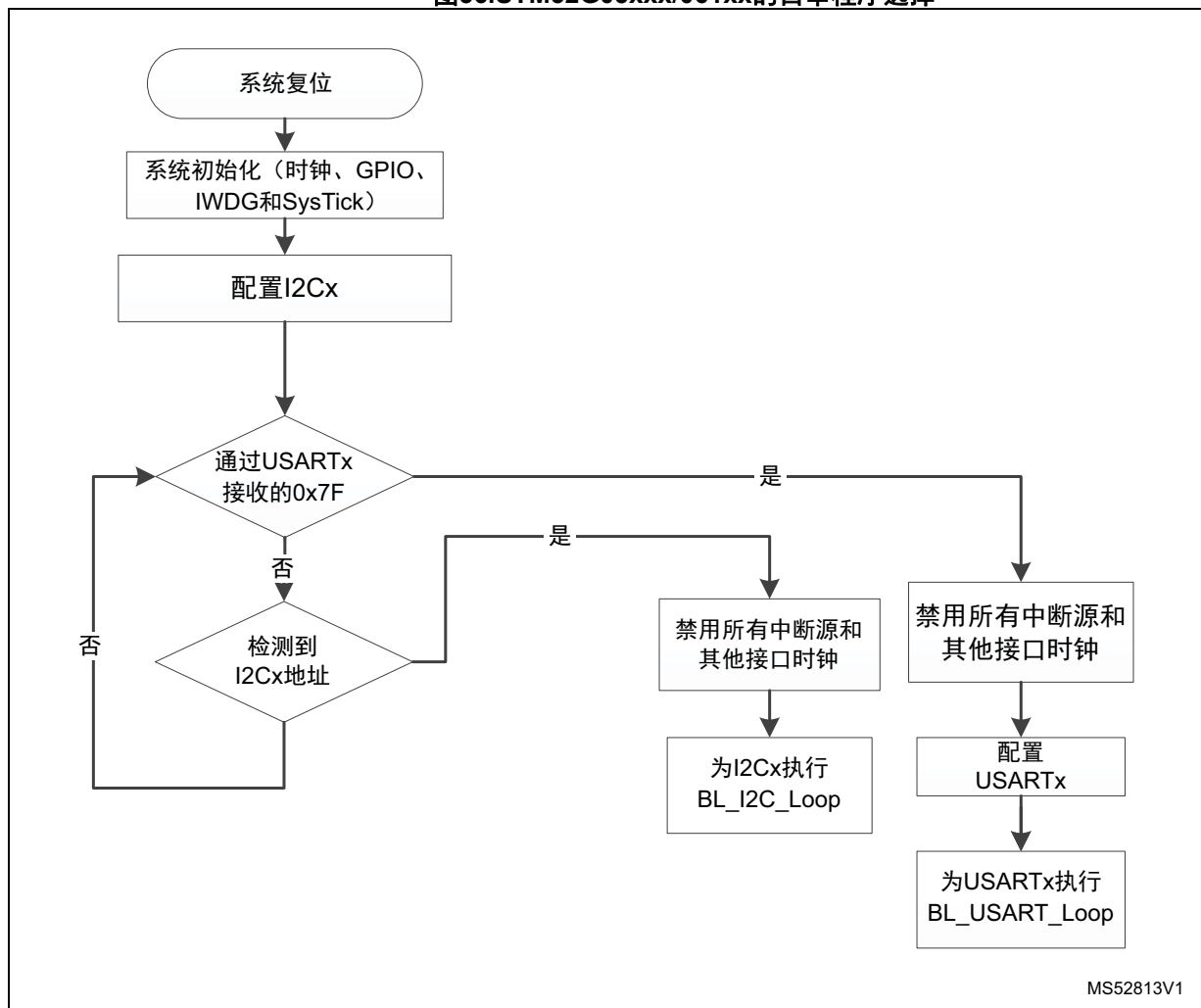
表97.系统存储器自举模式下STM32G05xxx/061xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1100010x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。

45.2 自举程序选择

下图显示了自举程序选择机制。

图56.STM32G05xxx/061xx的自举程序选择



45.3 自举程序版本

表 98列出了STM32G05xxx/061xx器件自举程序版本。

表98.STM32G05xxx/061xx自举程序版本

自举程序版本号	说明	已知限制
V5.0	初始自举程序版本	检测阶段的USART2软件抖动问题
V5.1	修正V5.0限制	无

46 STM32G431xx/441xx器件自举程序

46.1 自举程序配置

STM32G431xx/441xx自举程序通过应用模式15激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表99.系统存储器自举模式下STM32G431xx/441xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于72 MHz（使用由HSI驱动的PLL）
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的出口安全存储区的地址：@0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表99.系统存储器自举模式下STM32G431xx/441xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010100x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PC4引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PA8引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010100x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC8引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽，下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

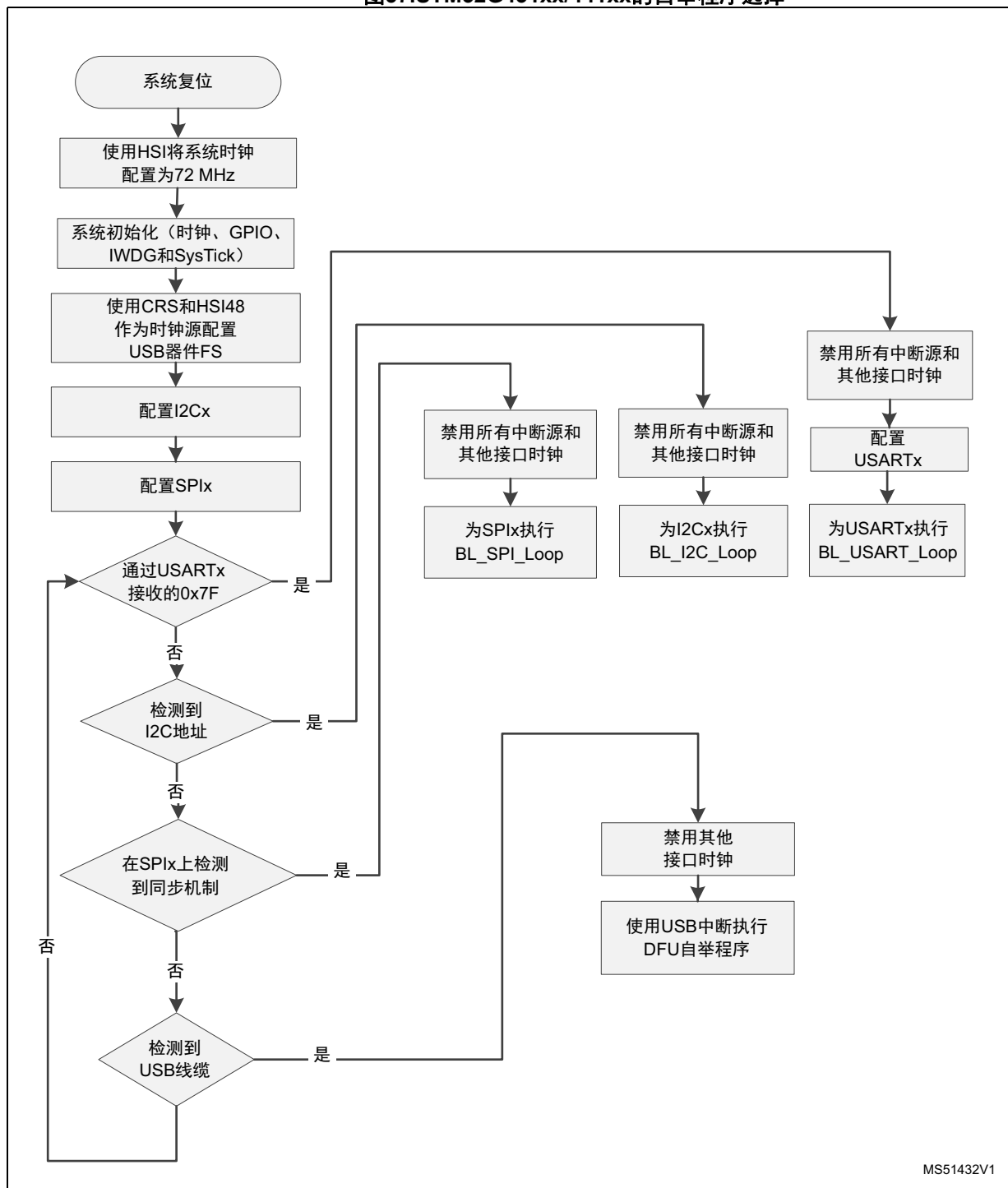
表99.系统存储器自举模式下STM32G431xx/441xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

46.2 自举程序选择

下图显示了自举程序选择机制。

图57.STM32G431xx/441xx的自举程序选择



MS51432V1

46.3 自举程序版本

表100.STM32G431xx/441xx自举程序版本

自举程序版本号	说明	已知限制
V13.3 (0xD3)	初始自举程序版本	不支持CCSRAM
V13.4 (0xD4)	修正V13.3限制	增加了CCSRAM支持信息

47 STM32G47xxx/48xxx器件自举程序

47.1 自举程序配置

STM32G47xxx/48xxx自举程序通过应用模式14激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表101.系统存储器自举模式下STM32G47xxx/48xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于72 MHz（使用由HSI驱动的PLL）
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的出口安全存储区的地址：@0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表101.系统存储器自举模式下STM32G47xxx/48xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010011x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PC4引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PA8引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010011x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC8引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏上拉模式下使用。
I2C4自举程序	I2C4	启用	I2C4配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010011x (其中x = 0对应写，x = 1对应读)
	I2C4_SCL引脚	输入/输出	PC6引脚：时钟线在开漏上拉模式下使用。
	I2C4_SDA引脚	输入/输出	PC7引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽，无上拉无下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

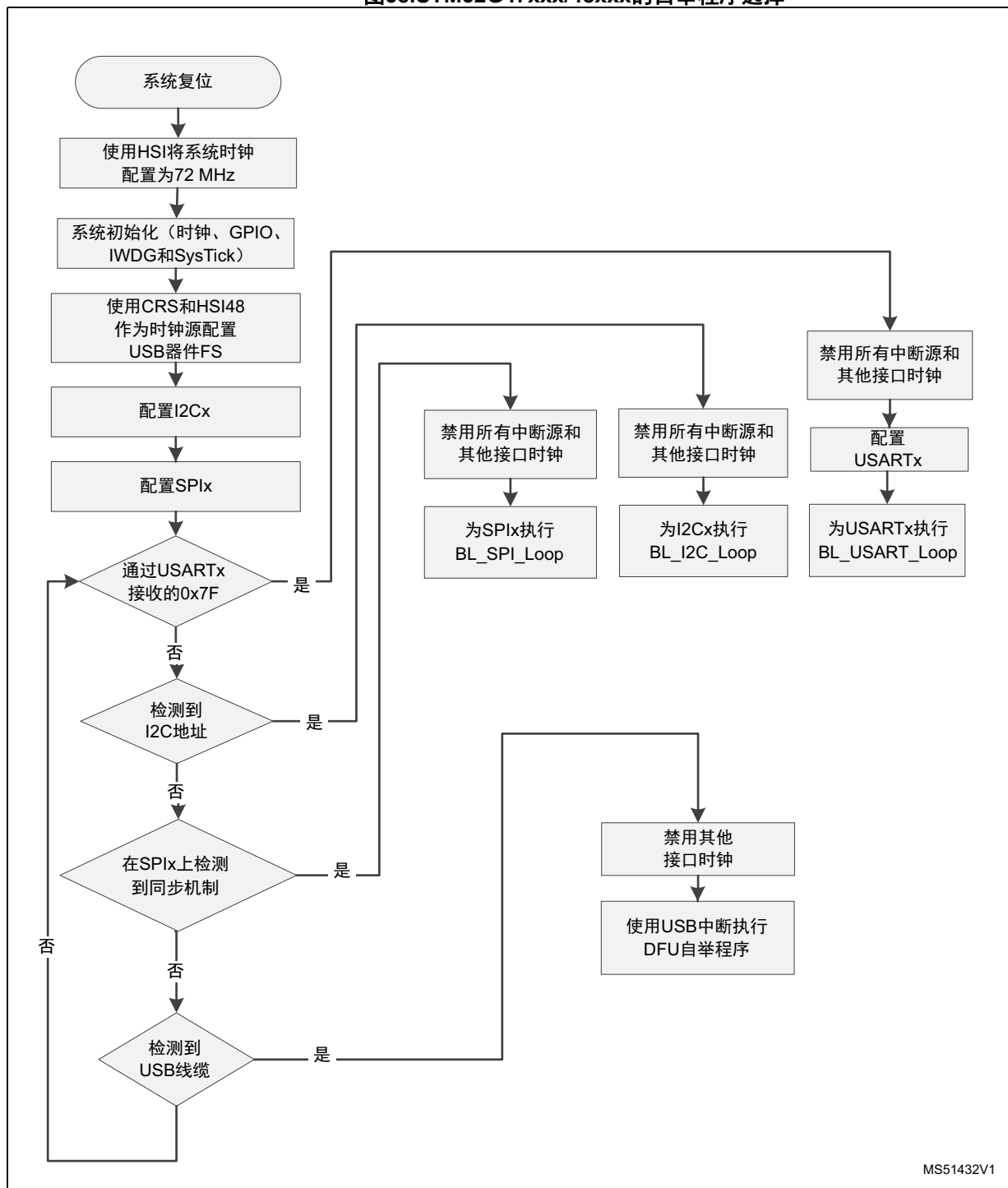
表101.系统存储器自举模式下STM32G47xxx/48xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽，下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意： 由于USB外设被自举程序使用，VDDUSBO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11：USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

47.2 自举程序选择

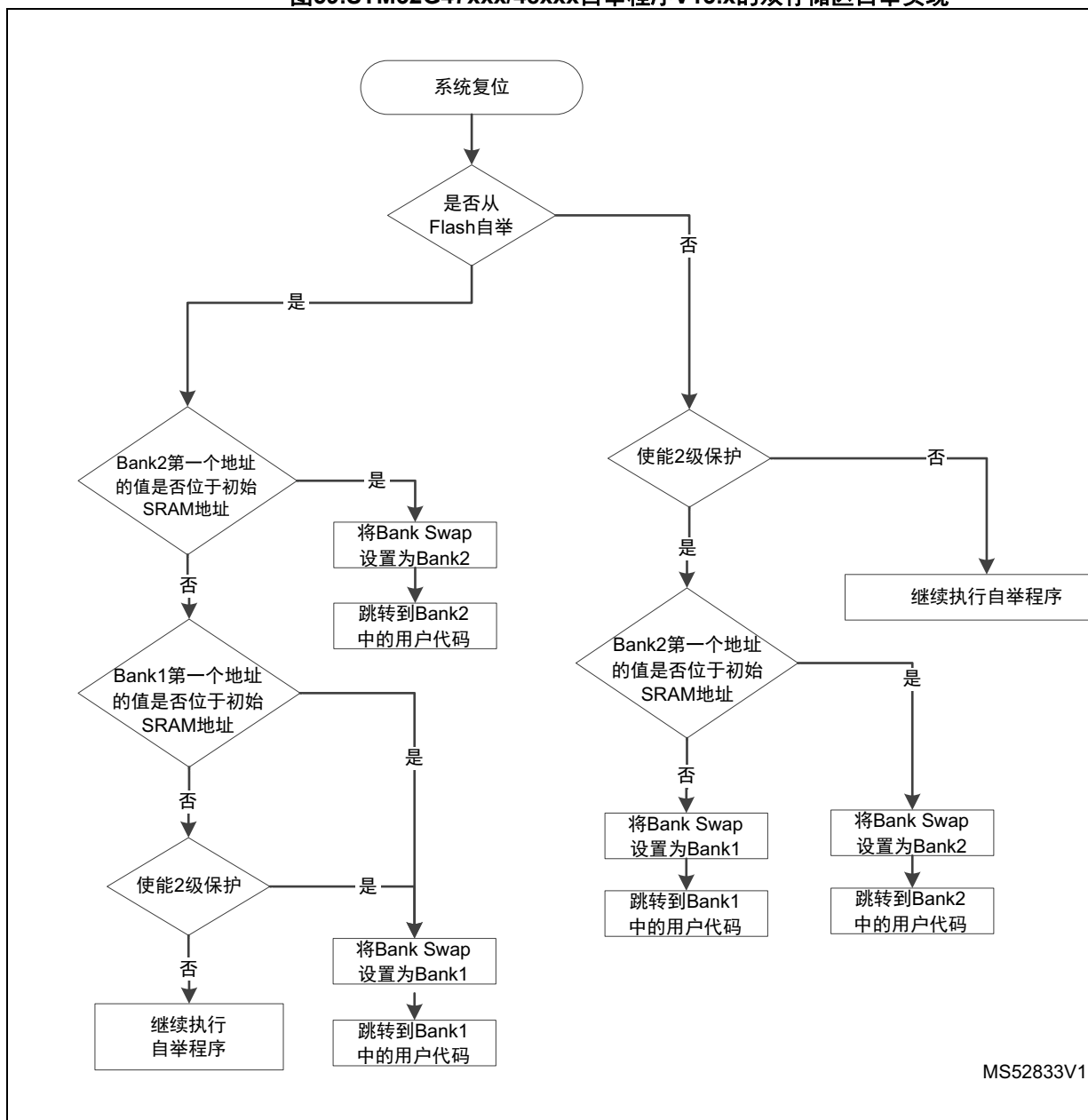
下图显示了自举程序选择机制。

图58.STM32G47xxx/48xxx的自举程序选择



MS51432V1

图59.STM32G47xxx/48xxx自举程序V13.x的双存储区自举实现



MS52833V1

47.3 自举程序版本

表102.STM32G47xxx/48xxx自举程序版本

自举程序版本号	说明	已知限制
V13.3 (0xD3)	初始自举程序版本	从bank2自举不工作
V13.4 (0xD4)	修正V13.3限制	不支持CCSRAM/ENGI
V13.5 (0xD5)	– 修正V13.4限制 – 增加了CCSRAM/ENGI支持	无

48 STM32G491xx/4A1xx器件自举程序

48.1 自举程序配置

STM32G491xx/4A1xx自举程序通过应用模式15激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表103.系统存储器自举模式下STM32G491xx/4A1xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于72 MHz（使用由HSI驱动的PLL）
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
安全的存储区	-	-	要跳转到的出口安全存储区的地址：@0x1FFF6800
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表103.系统存储器自举模式下STM32G491xx/4A1xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011111x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PC4引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PA8引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011111x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC8引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽，无上拉无下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽，下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

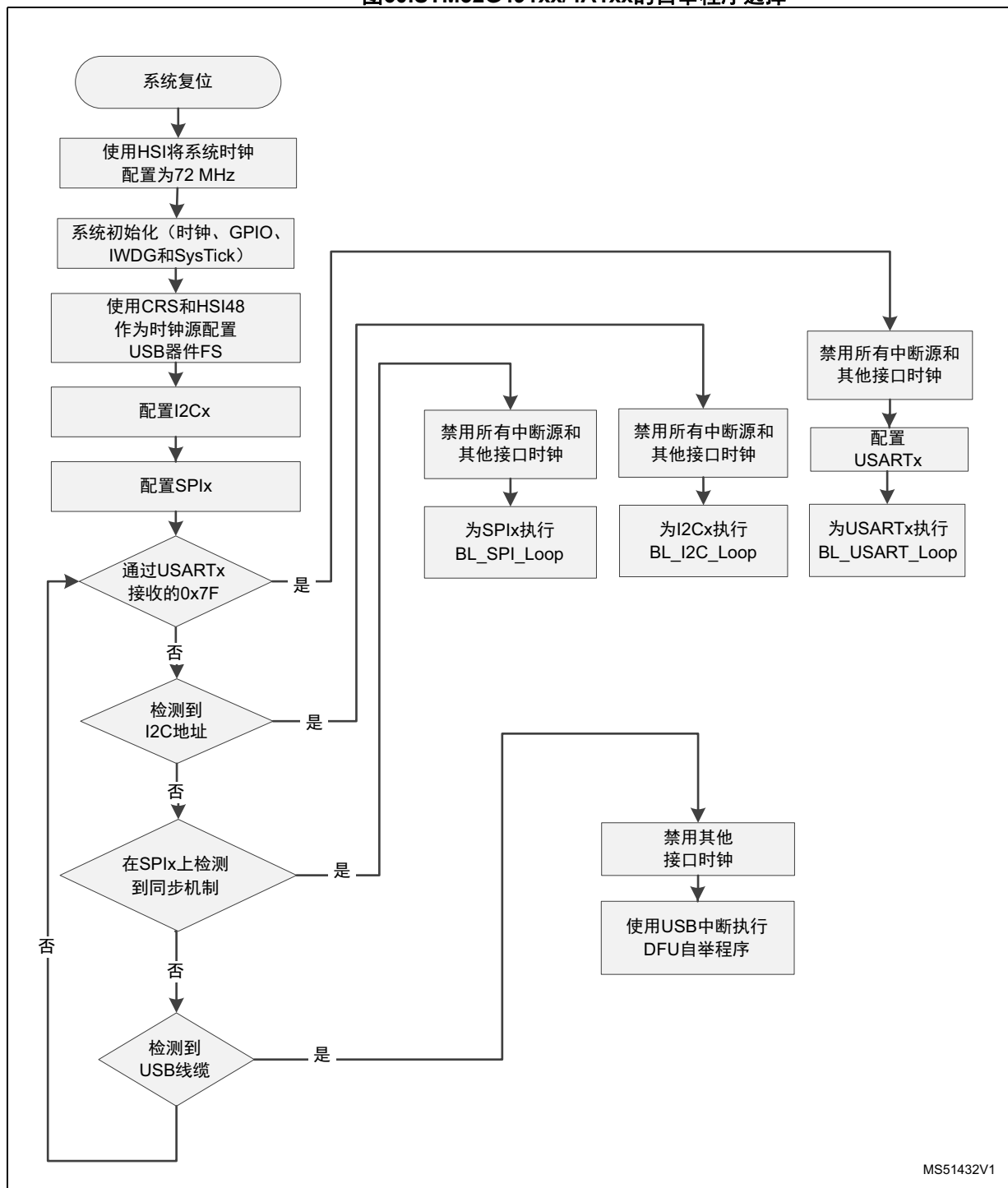
表103.系统存储器自举模式下STM32G491xx/4A1xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意: 由于USB外设被自举程序使用, VDDUSBO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

48.2 自举程序选择

下图显示了自举程序选择机制。

图60.STM32G491xx/4A1xx的自举程序选择



MS51432V1

48.3 自举程序版本

表104.STM32G491xx/4A1xx自举程序版本

自举程序版本号	说明	已知限制
V13.2	初始自举程序版本	无

49 STM32H72xxx/73xxx器件自举程序

49.1 自举程序配置

STM32H72xxx/73xxx自举程序通过应用模式10激活（详述参见表 2：自举程序激活模式）。表 105显示了该自举程序使用的硬件资源。

表105.系统存储器自举模式下STM32H72xxx/73xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于66 MHz（使用由HSI驱动的PLL）
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
		-	对FDCAN使用PLLQ提供的20 Mhz时钟频率
	RAM	-	自地址0x24000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF09800起的84 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围设置为电压范围3。 自举程序软件使用4个字节写入PWR_CR3寄存器，此举将锁定该寄存器。只有通过重启才能解锁。BL版本0x93修复了这个问题。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。设置为输入，直至在BL版本0x93上检测到USART1。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。设置为输入，直至在BL版本0x93上检测到USART2。

表105.系统存储器自举模式下STM32H72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于复用推挽，下拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于复用推挽，下拉模式。设置为输入，直至在BL版本0x93上检测到USART3。
USART3自举程序 (PD8/PD9)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PD9引脚：USART3处于接收模式。用于复用推挽，下拉模式。
	USART3_TX引脚	输出	PD8引脚：USART3处于发送模式。用于复用推挽，下拉模式。设置为输入，直至在BL版本0x93上检测到USART3。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010111x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010111x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010111x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。

表105.系统存储器自举模式下STM32H72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽下拉模式。
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽下拉模式。
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽下拉模式。
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽上拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND

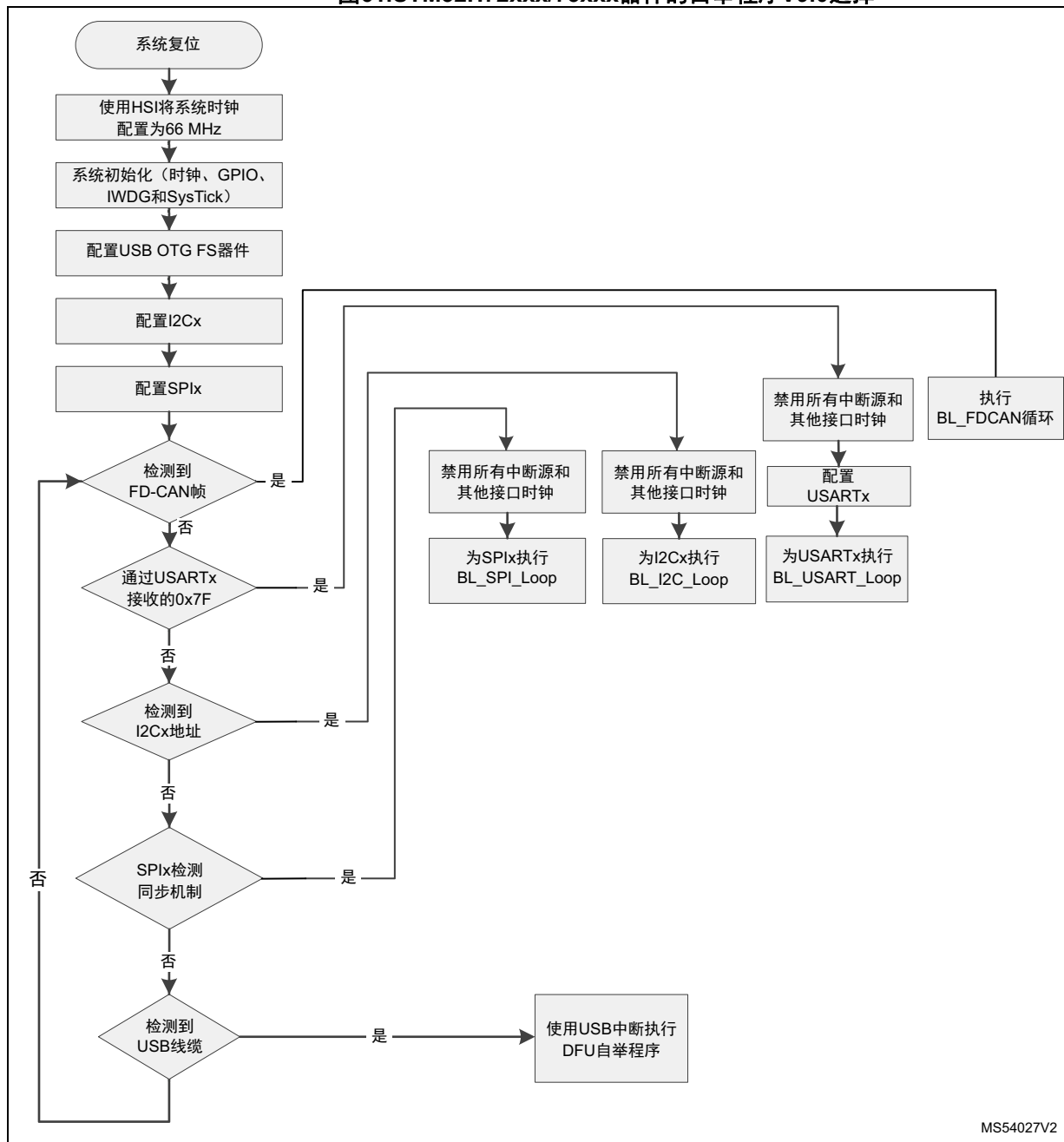
表105.系统存储器自举模式下STM32H72xxx/73xxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
FDCAN自举程序 (PH13/PH14)	FDCAN1	启用	初始化后，FDCAN1的配置为： 比特率为0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式FDCAN_MODE_NORMA AutoRetransmission使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PH14引脚: FDCAN1处于接收模式。用于复用推挽，下拉模式。
	FDCAN1_Tx引脚	输出	PH13引脚: FDCAN1处于发送模式。用于复用推挽，下拉模式。
FDCAN自举程序 (PD1/PD0)	FDCAN1	启用	初始化后，FDCAN1的配置为： 比特率为0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式FDCAN_MODE_NORMA AutoRetransmission使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PD0引脚: FDCAN1处于接收模式。用于复用推挽，下拉模式。
	FDCAN1_Tx引脚	输出	PD1引脚: FDCAN1处于发送模式。用于复用推挽，下拉模式。

49.2 自举程序选择

图 61显示了自举程序选择机制。

图61.STM32H72xxx/73xxx器件的自举程序V9.0选择



49.3 自举程序版本

表 108列出了STM32H72xxx/73xxx器件自举程序版本。

表106.STM32H72xxx/73xxx自举程序版本

自举程序版本号	说明	已知限制
V9.1	初始自举程序版本	<ul style="list-style-type: none"> - 并非使用任何BL接口都能修改TCM_AXI OB - 使用USB时返回的描述存储器大小的字符串不正确
V9.2	修复上一个版本的所有问题	<ul style="list-style-type: none"> - 在BL上自举，将RDP设置为1级，执行复位或重启和插入USB线缆时发生崩溃循环。 - 当TCM_AXI_SHARED选项字节不是“0”时，BL在RDP为1级时不工作。在进入RDP L1之前，此OB的值必须置为“0”。 - 自举程序软件使用4个字节写入PWR_CR3寄存器，此举将锁定该寄存器。只有通过重启才能解锁。
V9.3	<ul style="list-style-type: none"> - 修复上一个版本的所有问题。 - 将USART TX从先前版本的推挽模式修改为输入。 	无

50 STM32H74xxx/75xxx器件自举程序

50.1 自举程序配置

STM32H74xxx/75xxx自举程序通过应用模式10激活（详述参见表 2：自举程序激活模式）。表 107显示了该自举程序使用的硬件资源。

表107.系统存储器自举模式下STM32H74xxx/75xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为64 MHz。 如果选择了USART或SPI或I2C接口，在启动时（接口检测阶段）使用HSI时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
		-	对FDCAN使用的时钟频率为固定的20 MHz且由PLLQ提供
	RAM	-	自地址0x20000000起的16 KB空间和自地址0x24000000起的208 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的122 KB包含自举程序固件 自举程序起始地址为0x1FF09800
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	电压范围设置为电压范围3。	
USART1自举程序 (PA9/PA10)	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART1自举程序 (PB14/PB15)	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PB15引脚：USART1处于接收模式。用于输入上拉模式。
	USART1_TX引脚	输出	PB14引脚：USART1处于发送模式。用于输入上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，无上拉/下拉模式。用于复用推挽，无上拉/下拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，无上拉/下拉模式。

表107.系统存储器自举模式下STM32H74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于复用推挽，无上拉/下拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于复用推挽，无上拉/下拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001110x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001110x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001110x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。

表107.系统存储器自举模式下STM32H74xxx/75xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PI3引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI2_MISO引脚	输出	PI2引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI2_SCK引脚	输入	PI1引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI2_NSS引脚	输入	PI0引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
SPI4自举程序	SPI4	启用	SPI4配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz – 极性：CPOL低、CPHA低、NSS硬件。
	SPI4_MOSI引脚	输入	PE14引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI4_MISO引脚	输出	PE13引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI4_SCK引脚	输入	PE12引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI4_NSS引脚	输入	PE11引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。

表107.系统存储器自举模式下STM32H74xxx/75xxx器件的配置（续）

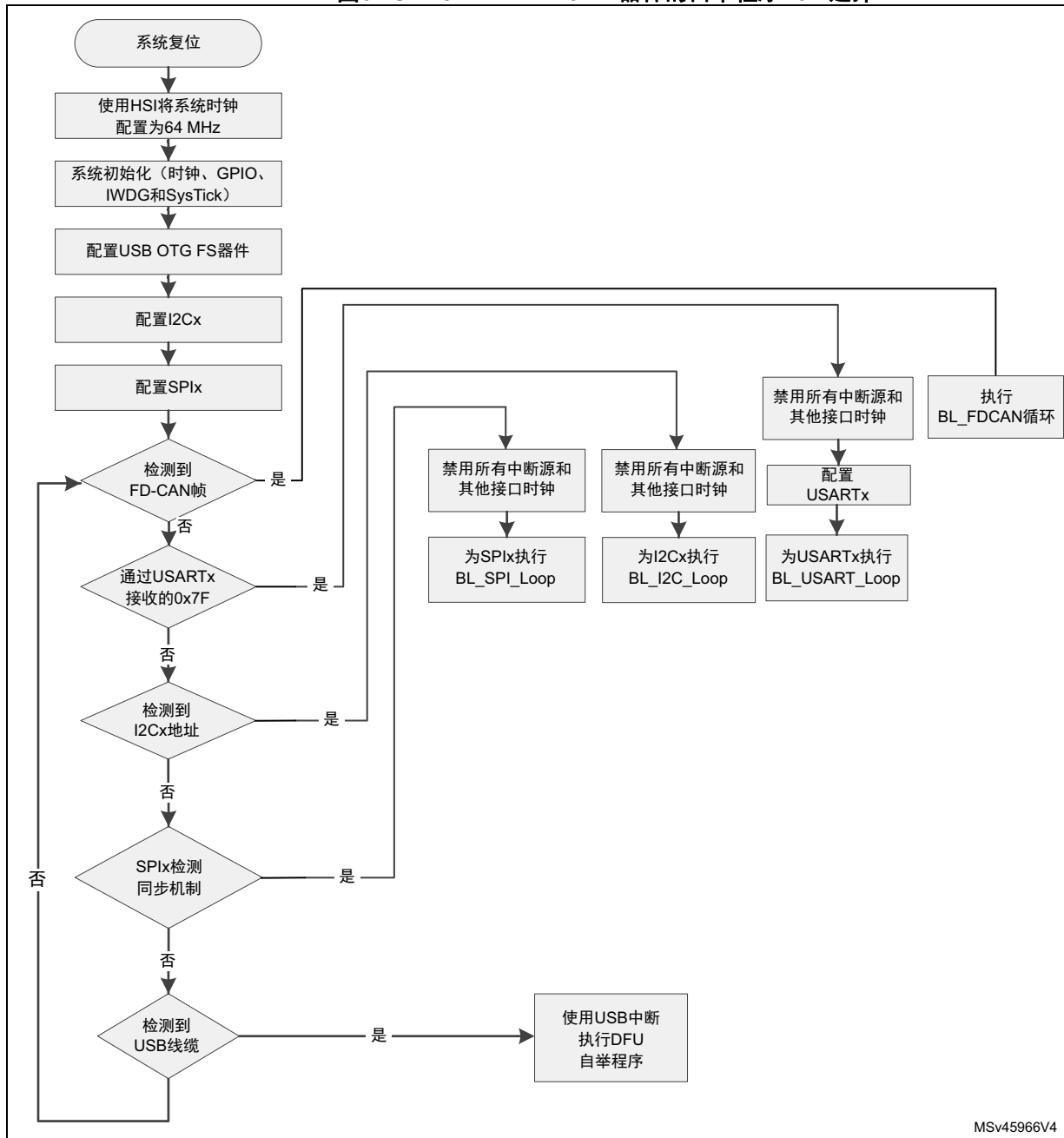
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
FDCAN自举程序	FDCAN1	启用	初始化后，FDCAN1的配置为： 比特率为0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式FDCAN_MODE_NORMA AutoRetransmission使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PH14引脚：FDCAN1处于接收模式。用于复用推挽，上拉模式。
	FDCAN1_Tx引脚	输出	PH13引脚：FDCAN1处于发送模式。用于复用推挽，上拉模式。

注： 为了使用PB14/PB15引脚连接自举程序USART1，用户必须发送两个同步字节。
DFU模式不支持USBREGEN模式。如果使用1.8V电源为STM32供电，将不可能使用BDFU，除非使用3.3 V电源。

50.2 自举程序选择

图 62显示了自举程序选择机制。

图62.STM32H74xxx/75xxx器件的自举程序V9.x选择



MSv45966V4



50.3 自举程序版本

表 108列出了STM32H74xxx/75xxx器件自举程序版本。

表108.STM32H74xxx/75xxx自举程序版本

自举程序版本号	说明	已知限制
V13.2 (0xD2)	初始自举程序版本	<ul style="list-style-type: none"> - “Go”命令不工作 - USART2连接不工作 - SPI1连接不工作 - I2C上的批量擦除不正常工作（在该命令中只擦除Bank2）
V13.3 (0xD3)	<ul style="list-style-type: none"> - 将USB时钟输入从HSE切换为具有CRS的HSI48 - V13.2修复了已知限制 	<ul style="list-style-type: none"> - USART/SPI和I2C上的存储区擦除不工作 - DFU自举程序批量擦除不工作
V9.0 (0x90)	<ul style="list-style-type: none"> - 增加了FDCAN接口支持 - 修正V13.3限制 - V9.0是生产中的最新版本并取代V13.2和V13.3 	<ul style="list-style-type: none"> - 使用USART或SPI时没有收到“Go”命令的第一个ACK - FDCAN写入存储器的限制，长度>63字节的数据写入失败 - 如果PB15置为GND，用户将不能连接到BL接口。只有USB能够连接，原因在于它使用中断进行检测。如果PB14/PB15上的USART1未使用，则PB15不得下拉。 - 一些应用上的跳转问题。为确保能够工作，应用栈指针必须小于(RAM结束 @ - 16字节) - 重启后需要再次复位才能连接到BL接口 - 双核配置下无法使用BL进行“CM4_BOOT_ADDx”选项字节编程 - FDCAN Getversion命令给出不良FDCAN协议版本(0x11)。必须是0 x10 (V1.0)。

51 STM32H7A3xx/B3xx器件自举程序

51.1 自举程序配置

STM32H7A3xx/7B3xx自举程序通过应用模式10激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表109.系统存储器自举模式下STM32H7A3xx/7B3xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为64 MHz。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI48 MHz作为时钟
		-	对FDCAN使用的时钟频率为固定的20 MHz且由PLLQ提供
	RAM	-	自地址0x24000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFFA000起的40 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序 (PB10/PB11)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PB11引脚：USART3处于接收模式。用于复用推挽，下拉模式。
	USART3_TX引脚	输出	PB10引脚：USART3处于发送模式。用于复用推挽，下拉模式。

表109.系统存储器自举模式下STM32H7A3xx/7B3xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序 (PD8/PD9)	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PD9引脚：USART3处于接收模式。用于复用推挽，下拉模式。
	USART3_TX引脚	输出	PD8引脚：USART3处于发送模式。用于复用推挽，下拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b10101111x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB9引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b10101111x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PF1引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PF0引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b10101111x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PA8引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC9引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽，无上拉无下拉模式。
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽，无上拉无下拉模式。
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽无上拉，无上拉无下拉模式。
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表109.系统存储器自举模式下STM32H7A3xx/7B3xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PC12引脚：从数据输入线，用于推挽，无上拉/下拉模式。
	SPI3_MISO引脚	输出	PC11引脚：从数据输出线，用于推挽，无上拉/下拉模式。
	SPI3_SCK引脚	输入	PC10引脚：从时钟线，用于推挽，无上拉/下拉模式。
	SPI3_NSS引脚	输入	PA15引脚：从芯片选择引脚，用于推挽，无上拉/下拉模式。
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻

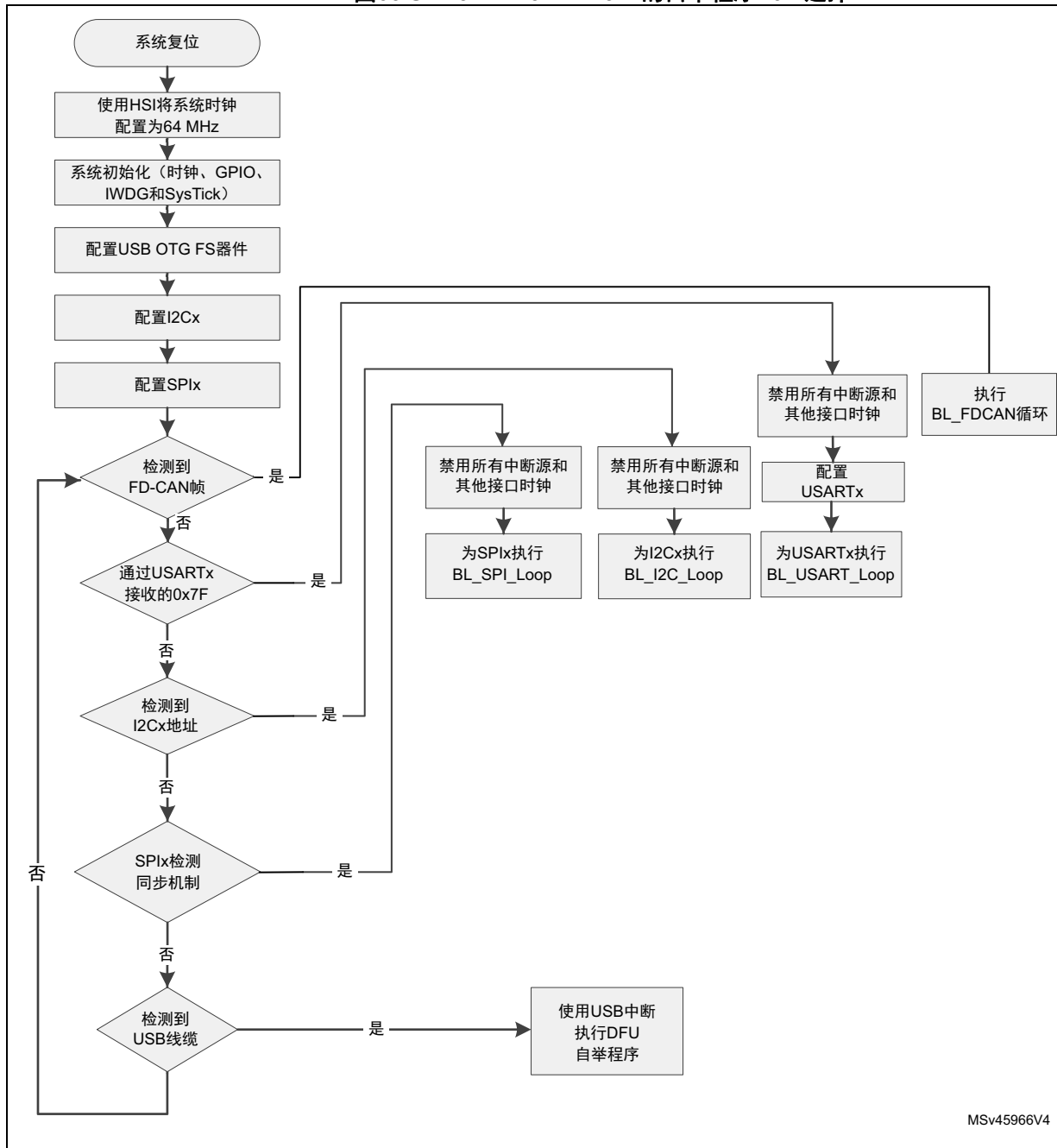
表109.系统存储器自举模式下STM32H7A3xx/7B3xx器件的配置（续）

自举程序	功能/外设	状态	备注
(PH13/PH14) 上的 FDCAN自举程序	FDCAN1	启用	初始化后的FDCAN1配置为：比特率0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式FDCAN_MODE_NORMA AutoRetransmission使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PH14引脚：FDCAN1处于接收模式。用于复用推挽，下拉模式。
	FDCAN1_Tx引脚	输出	PH13引脚：FDCAN1处于发送模式。用于复用推挽，下拉模式。
(PD1/PD0) 上的 FDCAN自举程序	FDCAN1	启用	初始化后的FDCAN1配置为：比特率0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式FDCAN_MODE_NORMA AutoRetransmission使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入	PD0引脚：FDCAN1处于接收模式。用于复用推挽，下拉模式。
	FDCAN1_Tx引脚	输出	PD1引脚：FDCAN1处于发送模式。用于复用推挽，下拉模式。

51.2 自举程序选择

图 62显示了自举程序选择机制。

图63.STM32H7A3xx/7B3xx的自举程序V9.x选择



MSv45966V4



51.3 自举程序版本

表 108列出了STM32H7A3xx/7B3xx器件自举程序版本。

表110.STM32H7A3xx/7B3xx自举程序版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	<ul style="list-style-type: none">– 使用USB时返回的描述Flash存储器大小的字符串不正确（预期值为256 x 8 Kb，但返回了256 x 2 KB）– 自举程序不支持OTP存储器
V9.1	修复上一个版本的所有问题。	<ul style="list-style-type: none">– 在BL上自举，将RDP设置为1级，执行复位或重启和插入USB线缆时发生崩溃循环。

52 STM32L01xxx/02xxx器件自举程序

52.1 自举程序配置

STM32L01xxx/02xxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 111显示了该自举程序使用的硬件资源。

表111.系统存储器自举模式下STM32L01xxx/02xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为32MHz，使用HSI16MHz为时钟源。
	RAM	-	自地址0x20000000起的2 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的4 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART2自举程序 (PA9/PA10)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA10引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA9引脚：USART2处于发送模式。用于输入上拉模式。
USART2自举程序 (PA2/PA3)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。
USART2自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
SPI1自举程序（适用于除TSSOP14之外的所有器件封装）	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。

表111.系统存储器自举模式下STM32L01xxx/02xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序（仅适用于采用TSSOP14封装的器件）	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA14引脚：从数据输出线，用于推挽下拉模式。 注： 此IO还作为SWCLK被用于调试接口，因此，当自举程序正在运行时，调试者无法在“动态”模式中连至器件。
	SPI1_SCK引脚	输入	PA13引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注： 对于TSSOP14封装的器件，具有SPI1接口的自举程序上需要NSS引脚同步。

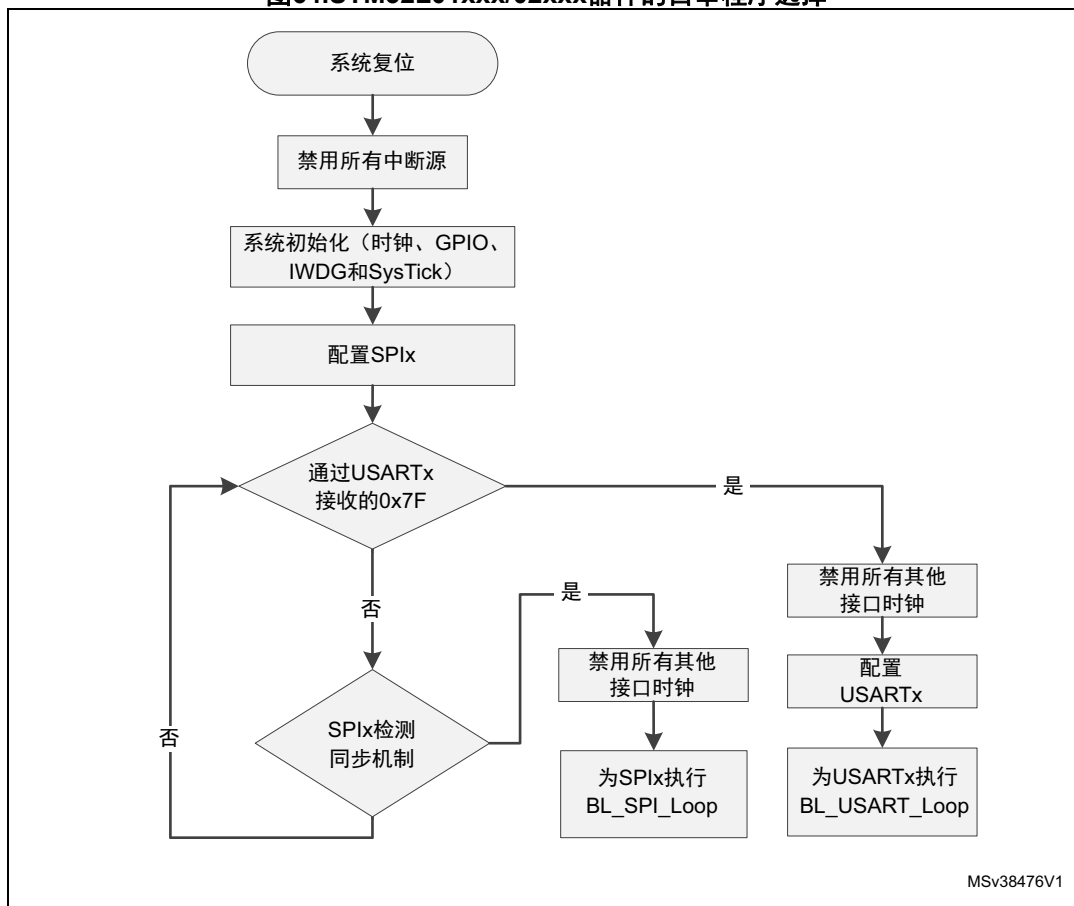
对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

注： 由于该产品有空数据检查机制，不可能从用户代码跳转到系统自举程序。此类跳转导致跳回用户Flash存储器空间。但是，如果跳转时用户Flash存储器的前4个字节（地址0x0800 0000）为空（即在跳转前擦除第一个扇区或在Flash为空时执行SRAM中的代码），则在跳转后执行系统自举程序。

52.2 自举程序选择

表 64显示了自举程序选择机制。

图64.STM32L01xxx/02xxx器件的自举程序选择



52.3 自举程序版本

下表列出了STM32L01xxx/02xxx器件自举程序版本。

表112.STM32L01xxx/02xxx自举程序版本

自举程序版本号	说明	已知限制
V12.2	初始自举程序版本	在TSSOP14封装的器件上，具有SPI1接口的自举程序不工作。
V12.3	此自举程序是自举程序V12.2的更新版本。该新版本针对TSSOP14封装的器件增加了对SPI接口的支持。	对于TSSOP14器件的SPI1接口，在开始通信前需要NSS引脚上的下降沿来正确同步SPI接口。若NSS引脚接地（在器件复位后的所有时间），则SPI通信不同步，自举程序无法与SPI接口正常工作。

53 STM32L031xx/041xx器件自举程序

53.1 自举程序配置

STM32L031xx/041xx自举程序通过应用模式2激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表113.系统存储器自举模式下STM32L031xx/041xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为32MHz，使用HSI16MHz为时钟源。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的4 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART2自举程序 (PA9/PA10)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA10引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA9引脚：USART2处于发送模式。用于输入上拉模式。
USART2自举程序 (PA2/PA3)	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。
USART2自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。

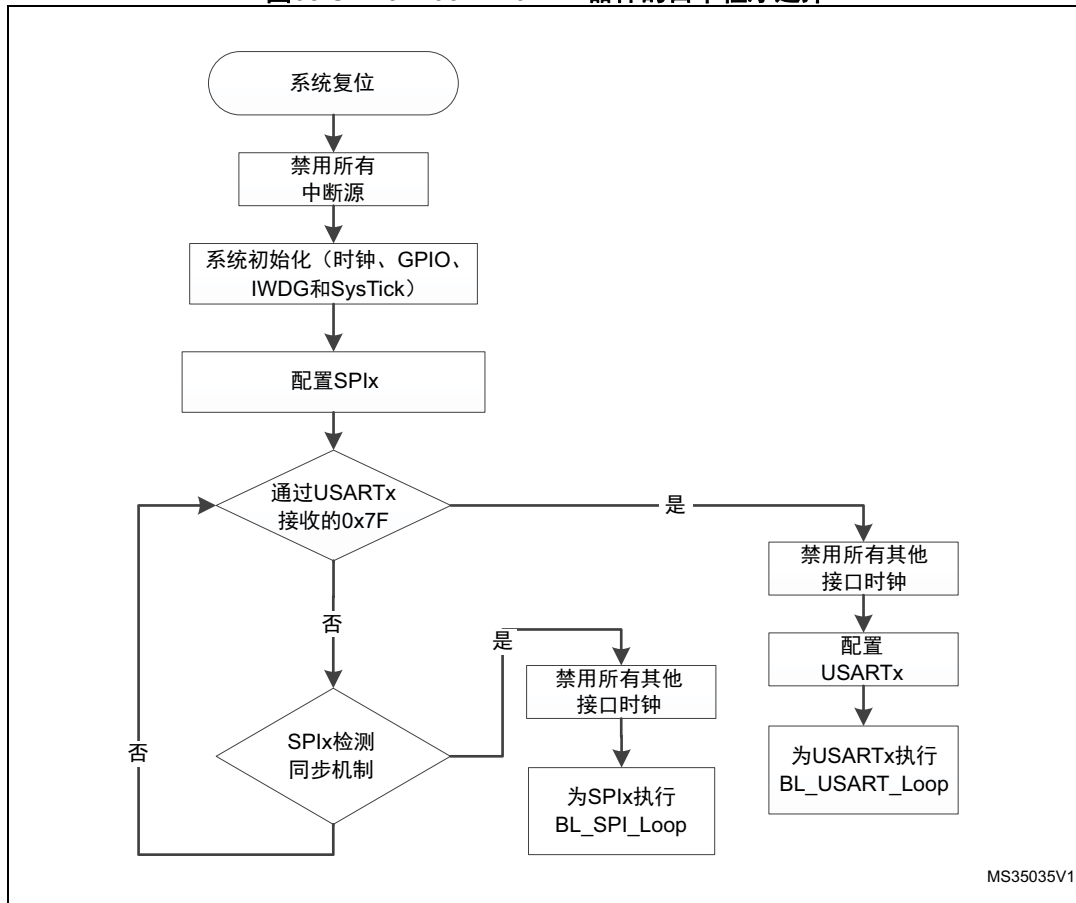
对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

对于该产品，自举程序Read/Write命令不支持SRAM空间。

53.2 自举程序选择

图 65显示了自举程序选择机制。

图65.STM32L031xx/041xx器件的自举程序选择



53.3 自举程序版本

表 114列出了STM32L031xx/041xx器件自举程序版本。

表114.STM32L031xx/041xx自举程序版本

自举程序版本号	说明	已知限制
V12.0	初始自举程序版本	无

54 STM32L05xxx/06xxx器件自举程序

54.1 自举程序配置

STM32L05xxx/06xxx自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表115.系统存储器自举模式下STM32L05xxx/06xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为32MHz，使用HSI16MHz为时钟源。
	电源	-	电压范围设置为电压范围1。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的4 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表115.系统存储器自举模式下STM32L05xxx/06xxx器件的配置（续）

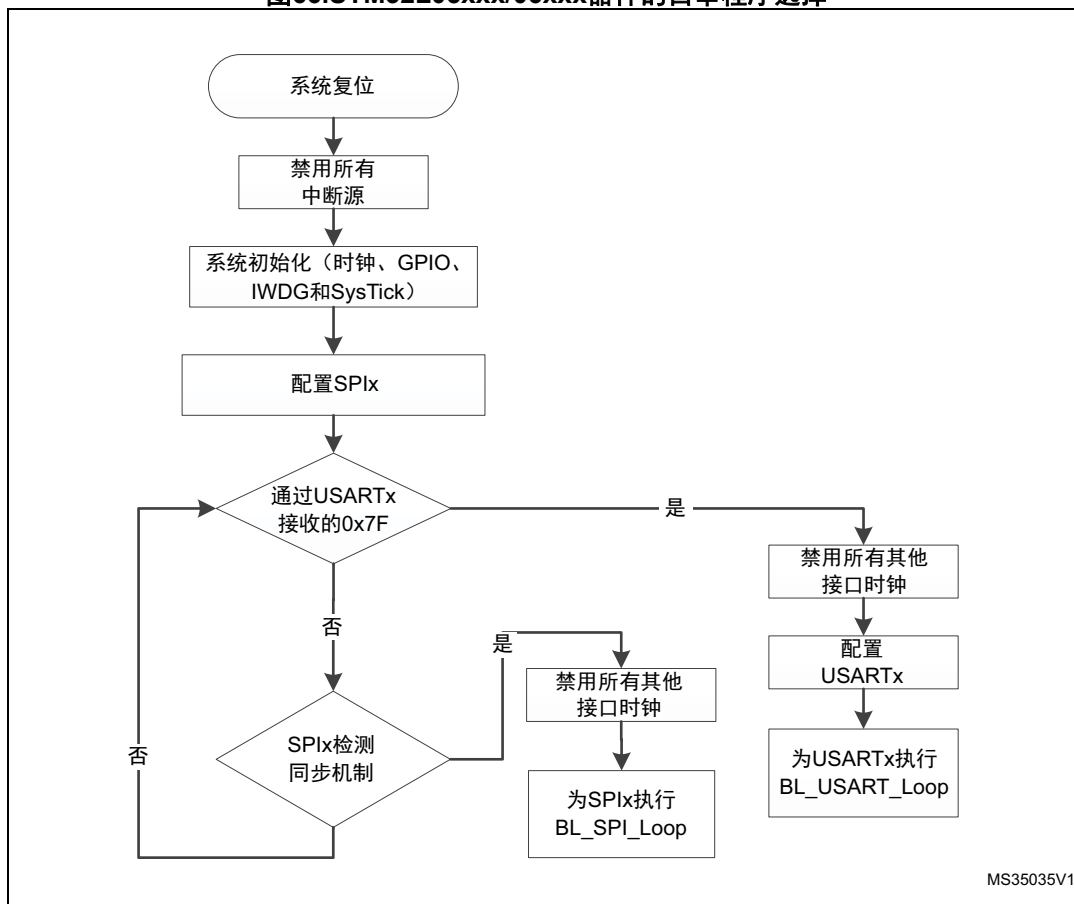
自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。

对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

54.2 自举程序选择

下图显示了自举程序选择机制。

图66.STM32L05xxx/06xxx器件的自举程序选择



54.3 自举程序版本

下表列出了STM32L05xxx/06xxx器件自举程序版本：

表116.STM32L05xxx/06xxx自举程序版本

自举程序版本号	说明	已知限制
V12.0	初始自举程序版本	PA13置为复用推挽上拉模式，PA14置为复用推挽下拉模式，即使不使用。

55 STM32L07xxx/08xxx器件自举程序

STM32L07xxx/08xxx器件可使用两种自举程序版本：

- V4.x支持USART1， USART2和DFU（USB FS设备）。
该版本嵌入STM32L072xx/73xx和STM32L082xx/83xx器件。
- V11.x支持USART1， USART2， I2C1， I2C2， SPI1和SPI2。
此版本内置在其他STM32L071xx/081xx器件。

55.1 自举程序V4.x

55.1.1 自举程序配置

当双存储区自举特性可用时，通过应用模式2或模式7激活STM32L07xxx/08xxx自举程序（如表 2：自举程序激活模式中所述）。表 117显示了该自举程序使用的硬件资源。

表117.系统存储器自举模式下STM32L07xxx/08xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为32 MHz，使用HSI 16 MHz为时钟源。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表117.系统存储器自举模式下STM32L07xxx/08xxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USBFS配置为强制器件模式。使能USBFS中断向量以用于USB DFU通信。
	USB_DM引脚	输入/输出	PA11引脚：USBFSDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12引脚：USBFS DP线用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻。

对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

55.1.2 自举程序选择

图 67和图 68显示了自举程序选择机制。

图67.STM32L07xxx/08xxx自举程序V4.x的双存储区自举实现

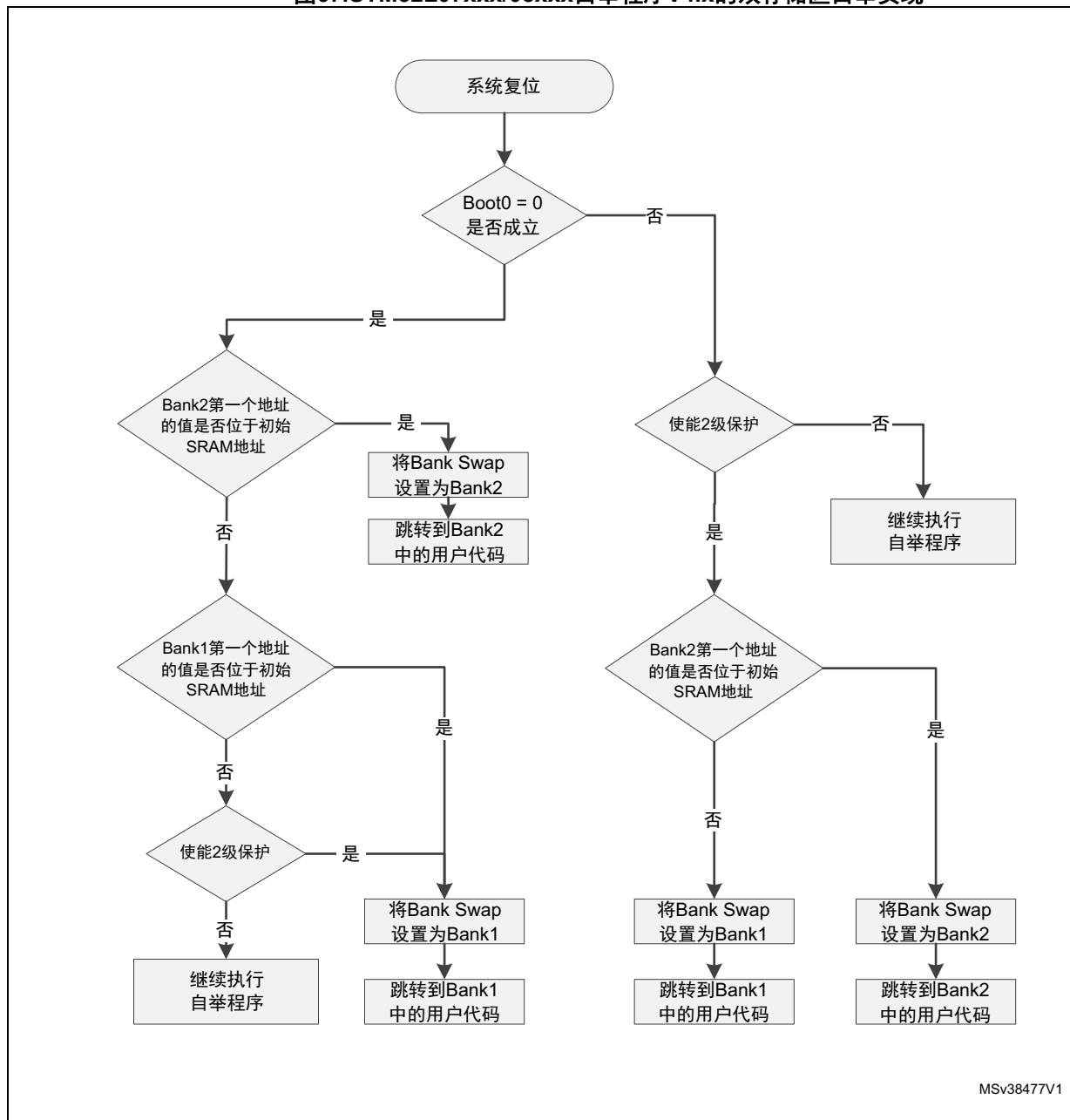
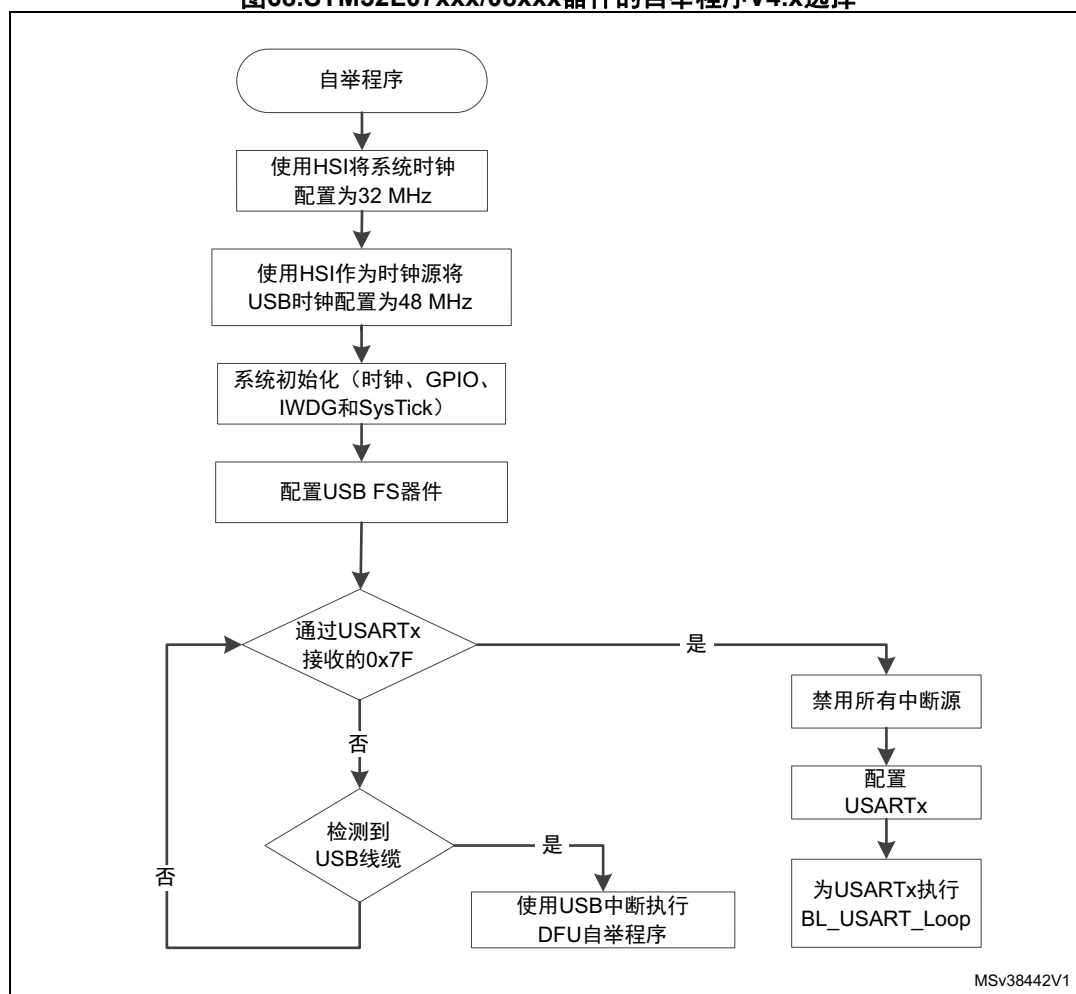


图68.STM32L07xxx/08xxx器件的自举程序V4.x选择



55.1.3 自举程序版本

表 118列出了STM32L07xxx/08xxx器件自举程序版本。

表118.STM32L07xxx/08xxx自举程序版本

自举程序版本号	说明	已知限制
V4.0	初始自举程序版本	PA4、PA5、PA6和PA7IO配置为下拉模式，即使不被自举程序使用。
V4.1	此自举程序是自举程序V4.0的更新版本。该新版本实现了双存储区自举特性。	PA4、PA5、PA6和PA7IO配置为下拉模式，即使不被自举程序使用。

55.2 自举程序V11.x

55.2.1 自举程序配置

当双存储区自举特性可用时，通过应用模式2或模式7激活STM32L07xxx/08xxx自举程序（参见表 2：自举程序激活模式）。表 119显示了该自举程序使用的硬件资源。

表119.系统存储器自举模式下STM32L07xxx/08xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为32MHz，使用HSI16MHz为时钟源。
	RAM	-	自地址0x20000000起的5 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FF00000起的8 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000010x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：I2C1时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：I2C1数据线在开漏无上拉/下拉模式下使用。

表119.系统存储器自举模式下STM32L07xxx/08xxx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000010x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：I2C2时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：I2C2数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8 MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不适用IO，则此IO可连接到GND。

对于所有自举程序接口，系统时钟由内部高速RC提供。因此，执行自举程序时无需外部石英时钟。

55.2.2 自举程序选择

图 69和图 70显示了自举程序选择机制。

图69.STM32L07xxx/08xxx自举程序V11.x的双存储区自举实现

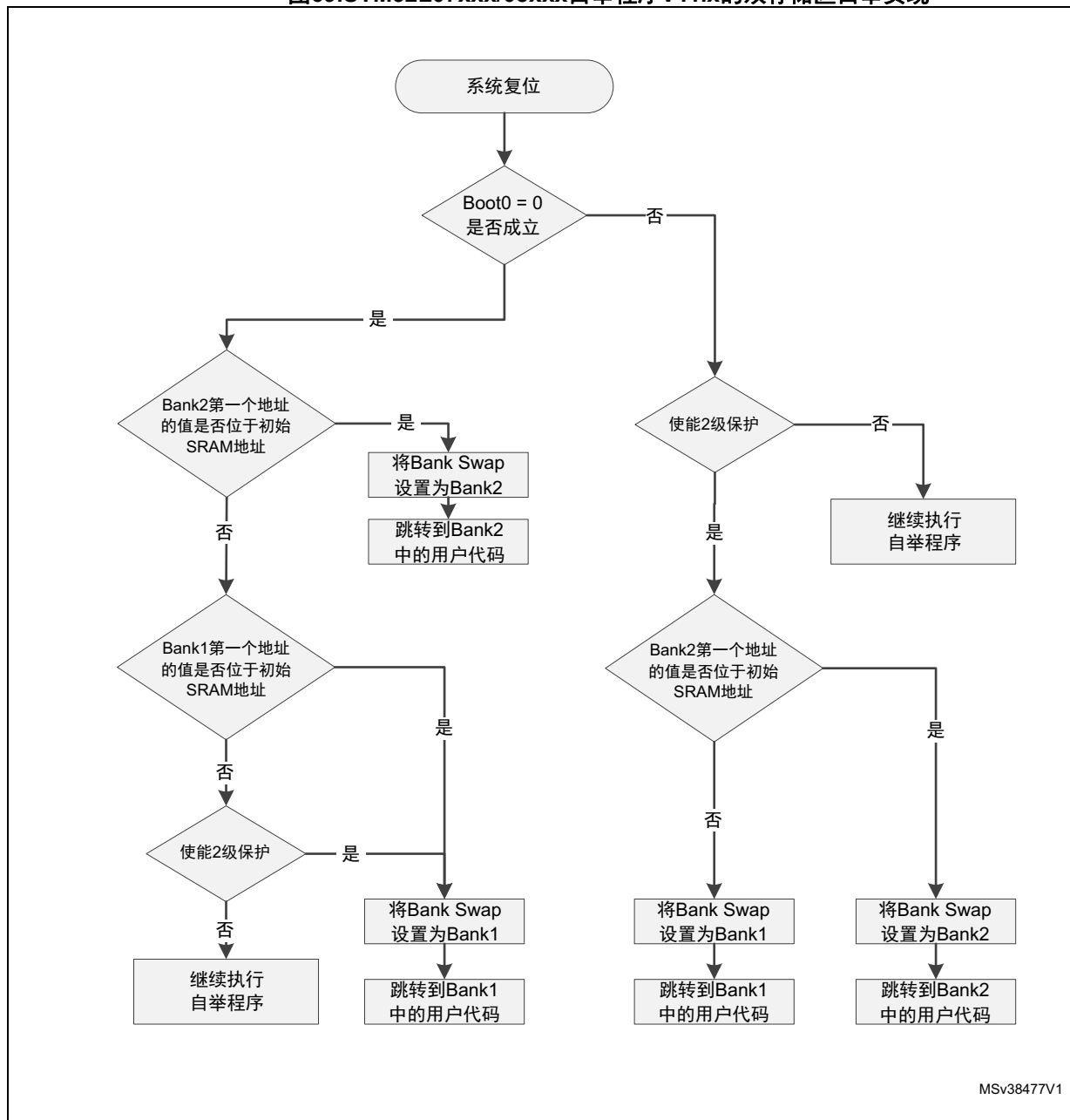
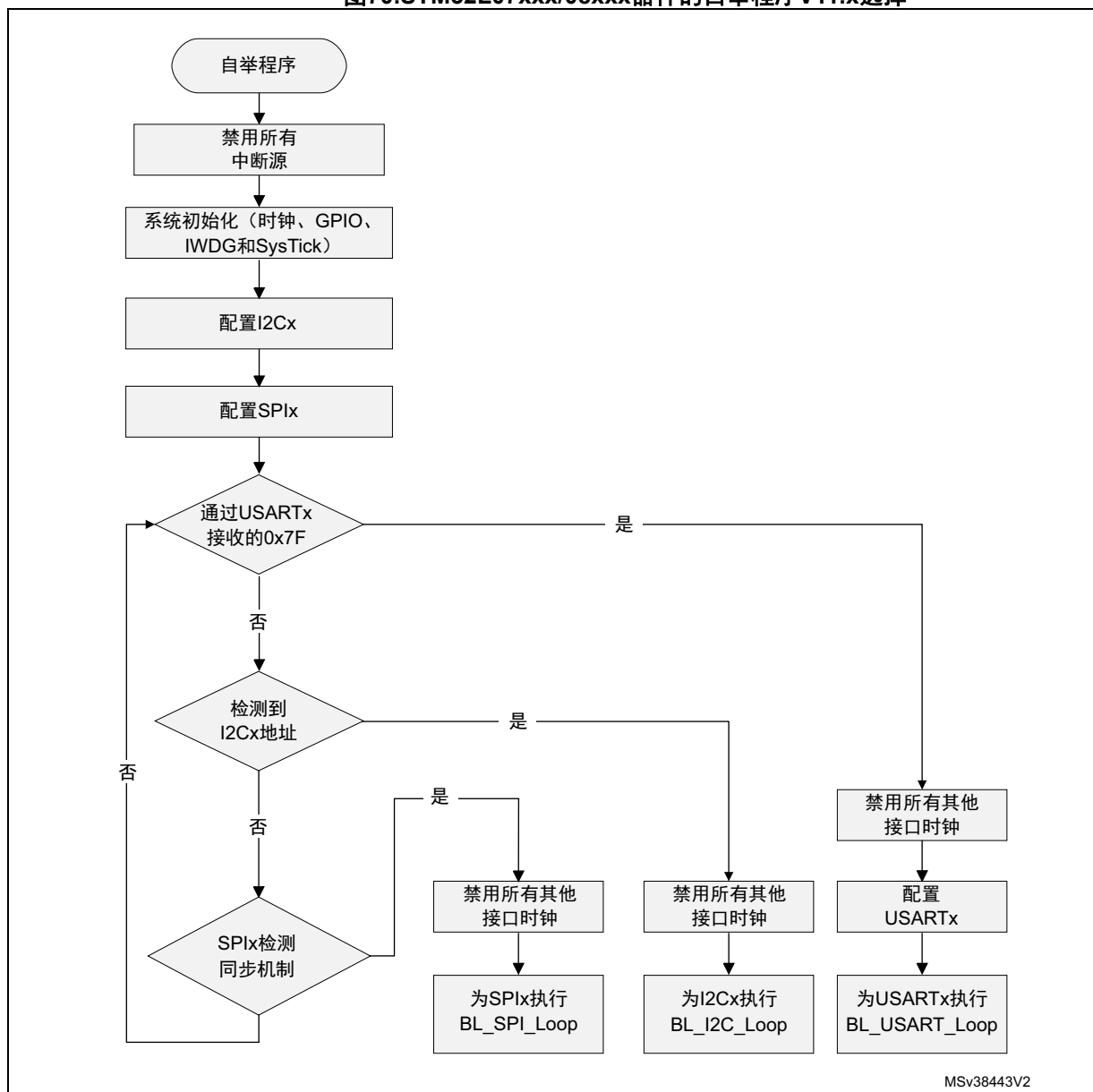


图70.STM32L07xxx/08xxx器件的自举程序V11.x选择



55.2.3 自举程序版本

下表列出了STM32L07xxx/08xxx器件自举程序版本：

表120.STM32L07xxx/08xxx自举程序V11.x版本

自举程序版本号	说明	已知限制
V11.1	初始自举程序版本	无
V11.2	此自举程序是自举程序V11.1的更新版本。该新版本实现了双存储区自举特性。	无

56 STM32L1xxx6 (8/B) A器件自举程序

56.1 自举程序配置

STM32L1xxx6 (8/B) A自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表121.系统存储器自举模式下STM32L1xxx6 (8/B) A器件的配置

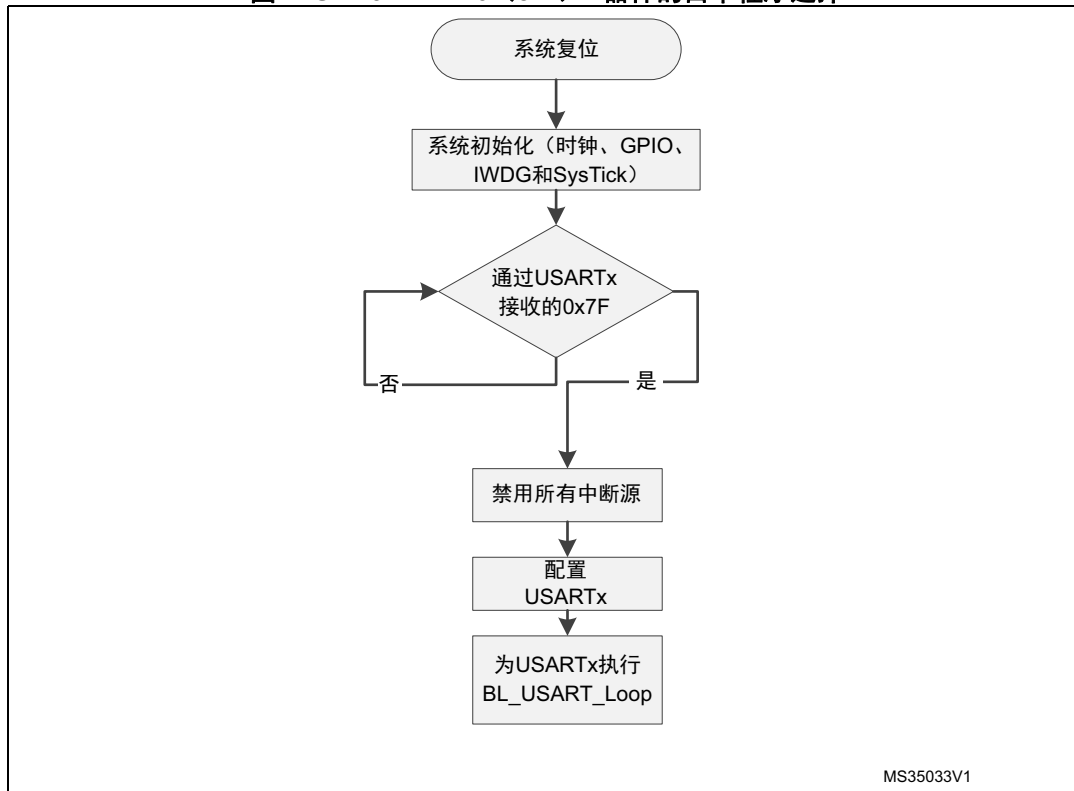
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为16 MHz。
	RAM	-	自地址0x20000000起的2 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的4 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为电压范围1。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

56.2 自举程序选择

下图显示了自举程序选择机制。

图71.STM32L1xxx6 (8/B) A器件的自举程序选择



56.3 自举程序版本

下表列出了STM32L1xxx6 (8/B) A器件自举程序版本：

表122.STM32L1xxx6 (8/B) A自举程序版本

自举程序版本号	说明	已知限制
V2.0	初始自举程序版本	当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址0x60000000）发出ReadMemory命令或WriteMemory命令时，自举程序器件会中止这一命令，但不会向主机发送NACK(0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和。 ⁽¹⁾

1. 如果待读/写的“数据数 - 1” (N-1) 不等于有效命令代码，则无法从主机发现缺陷，因为该命令始终会收到NACK应答（当作不受支持的新命令）。

57 STM32L1xxx6 (8/B) 器件自举程序

57.1 自举程序配置

STM32L1xxx6 (8/B) 自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表123.系统存储器自举模式下STM32L1xxx6 (8/B) 器件的配置

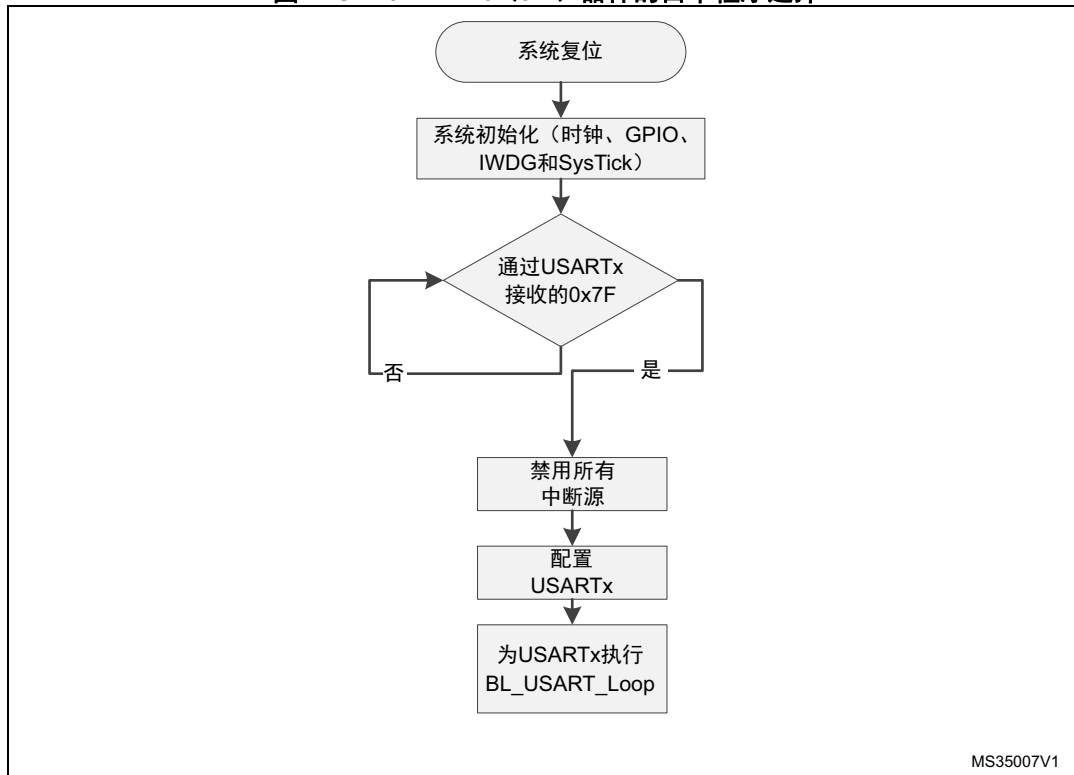
自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率为16 MHz。
	RAM	-	自地址0x20000000起的2 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的4 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为电压范围1。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测主机串口波特率。

系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟。

57.2 自举程序选择

下图显示了自举程序选择机制。

图72.STM32L1xxx6 (8/B) 器件的自举程序选择



57.3 自举程序版本

下表列出了STM32L1xxx6 (8/B) 器件自举程序版本：

表124.STM32L1xxx6 (8/B) 自举程序版本

自举程序版本号	说明	已知限制
V2.0	初始自举程序版本	<p>当通过一个不受支持的存储器地址和一个正确的地址检验和（即，地址0x6000 0000）发出Read Memory命令或Write Memory命令时，自举程序器件会中止这一命令，但不会向主机发送NACK (0x1F)。因此，接下来的两个字节（即，待读/写的字节数及其校验和）会被看作一条新命令及其校验和。⁽¹⁾</p> <p>PA13/14/15配置为复用推挽上拉模式（PA14为下拉），即使不使用。</p>

1. 如果待读/写的“数据数 - 1” (N-1) 不等于有效命令代码，则无法从主机发现缺陷，因为该命令始终会收到NACK应答（当作不受支持的新命令）。

58 STM32L1xxxC器件自举程序

58.1 自举程序配置

STM32L1xxxC自举程序通过应用模式1激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表125.系统存储器自举模式下STM32L1xxxC器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为16 MHz。仅用于USARTx自举程序，以及在USB检测期间用于DFU自举程序（选择DFU自举程序后，时钟源由外部石英时钟提供）。
		HSE使能	只有DFU自举程序必须使用外部时钟，外部时钟必须介于以下频率范围： [24、16、12、8、6、4、3、2] MHz。 PLL用于生成48MHzUSB时钟和32MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为电压范围1。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表125.系统存储器自举模式下STM32L1xxxC器件的配置（续）

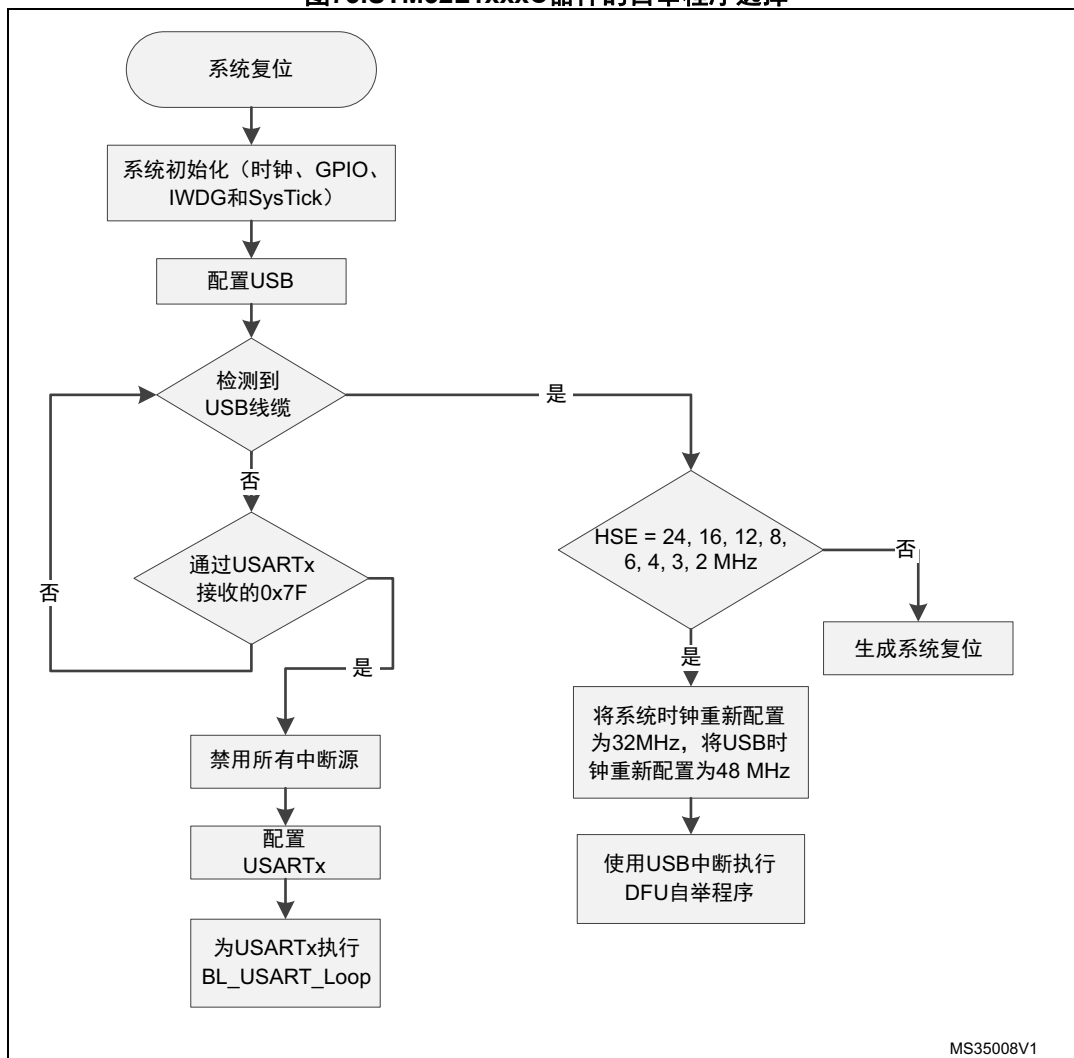
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。

对于USARTx自举程序，系统时钟由嵌入式内部高速RC提供。该内部时钟同样用于DFU自举程序，但仅限选择阶段。选择阶段结束后，执行DFU自举程序时需要频率介于范围 [24、16、12、8、6、4、3、2] MHz的外部时钟。

58.2 自举程序选择

下图显示了自举程序选择机制。

图73.STM32L1xxxC器件的自举程序选择



MS35008V1

58.3 自举程序版本

下表列出了STM32L1xxxC器件自举程序版本。

表126.STM32L1xxxC自举程序版本

自举程序版本号	说明	已知限制
V4.0	初始自举程序版本	对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号，而不是1个NACK信号。 PA13/14/15配置为复用推挽上拉模式（PA14为下拉），即使不使用。

59 STM32L1xxxD器件自举程序

59.1 自举程序配置

STM32L1xxxD自举程序通过应用模式4激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表127.系统存储器自举模式下STM32L1xxxD器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为16 MHz。仅用于USARTx自举程序，以及在USB检测期间用于DFU自举程序（选择DFU自举程序后，时钟源由外部石英时钟提供）。
		HSE使能	只有DFU自举程序必须使用外部时钟，外部时钟必须介于以下频率范围：[24、16、12、8、6、4、3、2] MHz。 PLL用于生成48MHz USB时钟和32MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为电压范围1。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表127.系统存储器自举模式下STM32L1xxxD器件的配置（续）

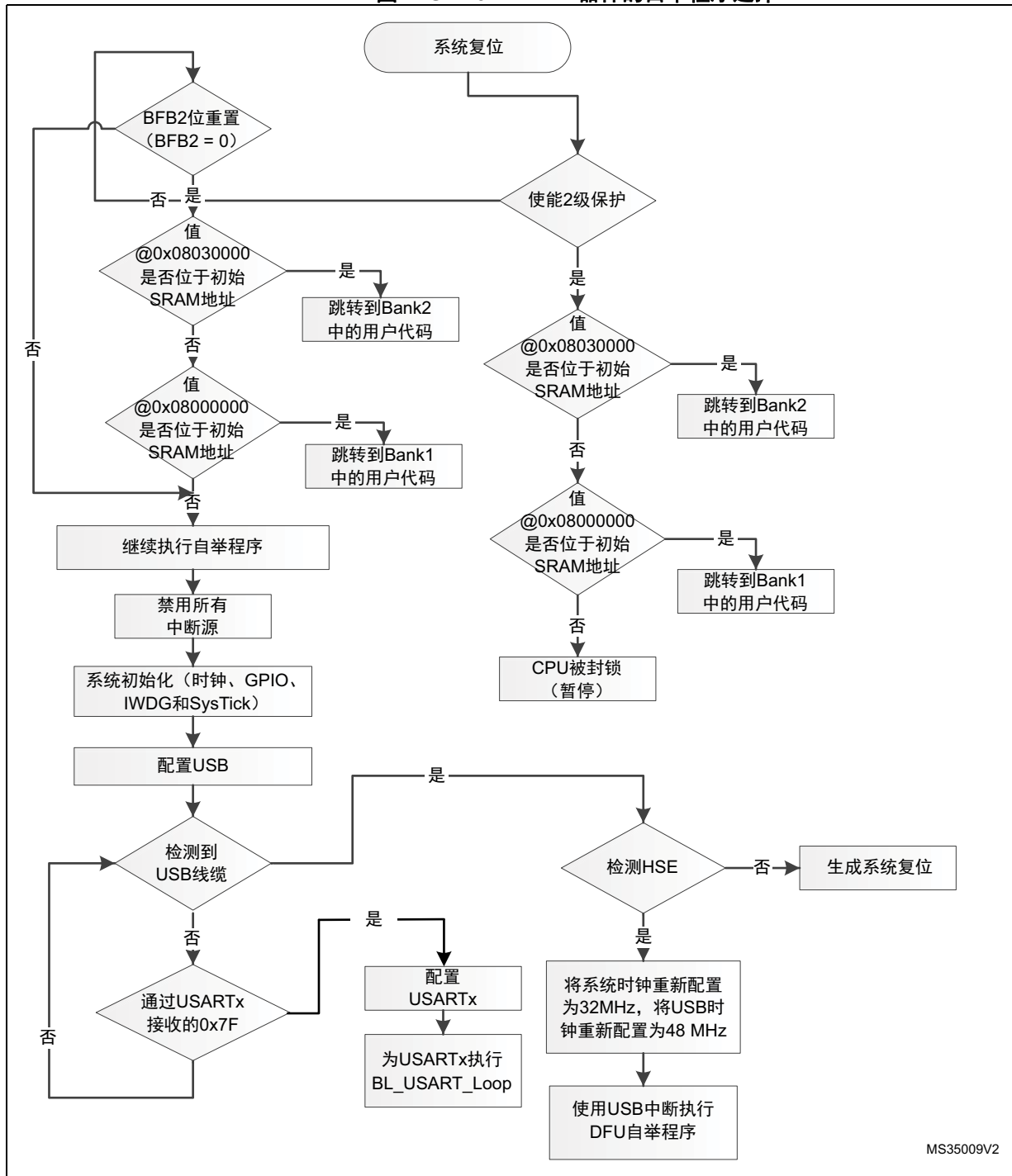
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。

对于USARTx自举程序，系统时钟由内部高速RC提供。该内部时钟同样用于DFU自举程序，但仅限选择阶段。选择阶段结束后，执行DFU自举程序时需要频率介于范围 [24、16、12、8、6、4、3、2] MHz的外部时钟。

59.2 自举程序选择

下图显示了自举程序选择机制。

图74.STM32L1xxD器件的自举程序选择



MS35009V2

59.3 自举程序版本

下表列出了STM32L1xxxD器件自举程序版本：

表128.STM32L1xxxD自举程序版本

自举程序版本号	说明	已知限制
V4.1	初始自举程序版本	<ul style="list-style-type: none"> - 自举程序代码中PA13 (JTMS/SWDIO) I/O输出速度配置为400 kHz, 因此, 在串行线模式下, 一些调试器将无法在自举程序运行期间与器件相连。 - 选择DFU自举程序时, RTC将复位, 进而丢失所有RTC信息 (如日历、闹钟), 包括备份寄存器内容。注意: 选择USART自举程序时, RTC配置 (包括备份寄存器) 保持不变。
V4.2	修正V4.1限制 (仅适用于第Z版器件)	<ul style="list-style-type: none"> - 当BFB2=0或读保护级别设为2时, 跳转到存储区1/存储区2后, 堆栈上溢8字节。 解决方案: 跳转到主程序之前, 用户代码须强制放入启动文件栈顶地址。这可以在“Reset_Handler”程序中实现。 - 当用户代码的栈位于SRAM (即@0x2000C000) 以外时, 自举程序无法跳转到被视为无效的用户代码。这种情况可能在使用编译器时发生, 因为此时会将堆栈置于SRAM栈顶的非物理地址 (即地址0x2000C000) 处。 解决方案: 将堆栈手动置于物理地址处。
V4.5	修正V4.2限制。 提高DFU接口耐用性 (仅适用于第Y版器件)。	<ul style="list-style-type: none"> - 对于USART接口, 当发送ReadMemory或Write Memory命令且RDP电平有效时, 将发送两个连续的NACK信号 (而不是1个NACK信号)。

60 STM32L1xxxE器件自举程序

60.1 自举程序配置

STM32L1xxxE自举程序通过应用模式4激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表129.系统存储器自举模式下STM32L1xxxE器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	使用HSI作为系统时钟，频率为16 MHz。仅用于USARTx自举程序，以及在USB检测期间用于DFU自举程序（选择DFU自举程序后，时钟源由外部石英时钟提供）。
		HSE使能	只有DFU自举程序必须使用外部时钟，外部时钟必须介于以下频率范围：[24、16、12、8、6、4、3、2] MHz。 PLL用于生成48MHz USB时钟和32MHz系统时钟。
		-	使能时钟安全系统（CSS）中断以用于DFU自举程序。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的4 KB空间供自举程序固件使用。
	系统存储器	-	自地址0x1FF00000起的8 KB空间包含自举程序固件。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	电压范围设置为电压范围1。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。USART2使用其重映射引脚。
	USART2_RX引脚	输入	PD6引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PD5引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表129.系统存储器自举模式下STM32L1xxxE器件的配置（续）

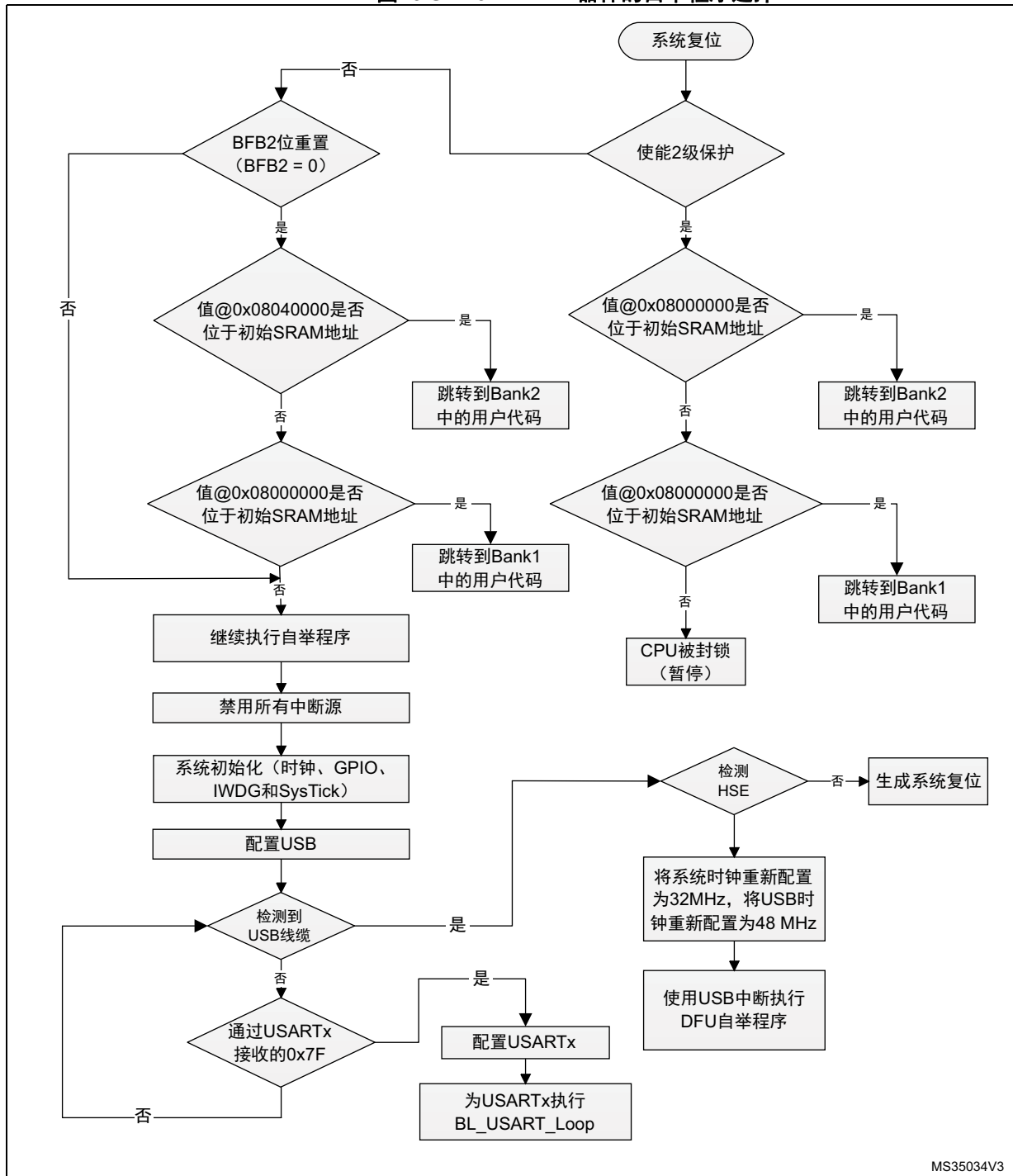
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	使用USB全速模式
	USB_DM引脚	输入/输出	PA11: USB DM线。
	USB_DP引脚		PA12: USB DP线。

对于USARTx自举程序，系统时钟由内部高速RC提供。该内部时钟同样用于DFU自举程序，但仅限选择阶段。选择阶段结束后，执行DFU自举程序时需要频率介于范围 [24、16、12、8、6、4、3、2] MHz的外部时钟。

60.2 自举程序选择

下图显示了自举程序选择机制。

图75.STM32L1xxxE器件的自举程序选择



MS35034V3



60.3 自举程序版本

下表列出了STM32L1xxxE器件自举程序版本：

表130.STM32L1xxxE自举程序版本

自举程序版本号	说明	已知限制
V4.0	初始自举程序版本	对于USART接口，当发送Read Memory或Write Memory命令且RDP电平有效时，将发送两个连续的NACK信号（而不是1个NACK信号）。 PA13/14/15配置为复用推挽上拉模式（PA14为下拉），即使不使用。

61 STM32L412xx/422xx器件自举程序

61.1 自举程序配置

STM32L412xx/422xx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表131.系统存储器自举模式下STM32L412xx/422xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为72 MHz的系统时钟，以及USART、I2C、SPI和USB自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock RecoverySystem, CRS），支持USB以HSI48MHz作为时钟。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。

表131.系统存储器自举模式下STM32L412xx/422xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010010x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010010x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010010x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不使用IO，则此IO可连接到GND。

表131.系统存储器自举模式下STM32L412xx/422xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz 极性：CPOL低、CPHA低、NSS硬件
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不使用IO，则此IO可连接到GND。
DFU自举程序	USB：	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意： 由于USB外设被自举程序使用，VDDUSBO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻

注： 如果VDDUSB引脚没有连接到VDD，则SPIFlash存储器写操作可能因电压问题而发生错误。请参考产品的数据手册和勘误表获取更多信息。

61.2 自举程序选择

下图显示了自举程序选择机制。

图76.STM32L412xx/422xx自举程序V9.x的双存储区自举实现

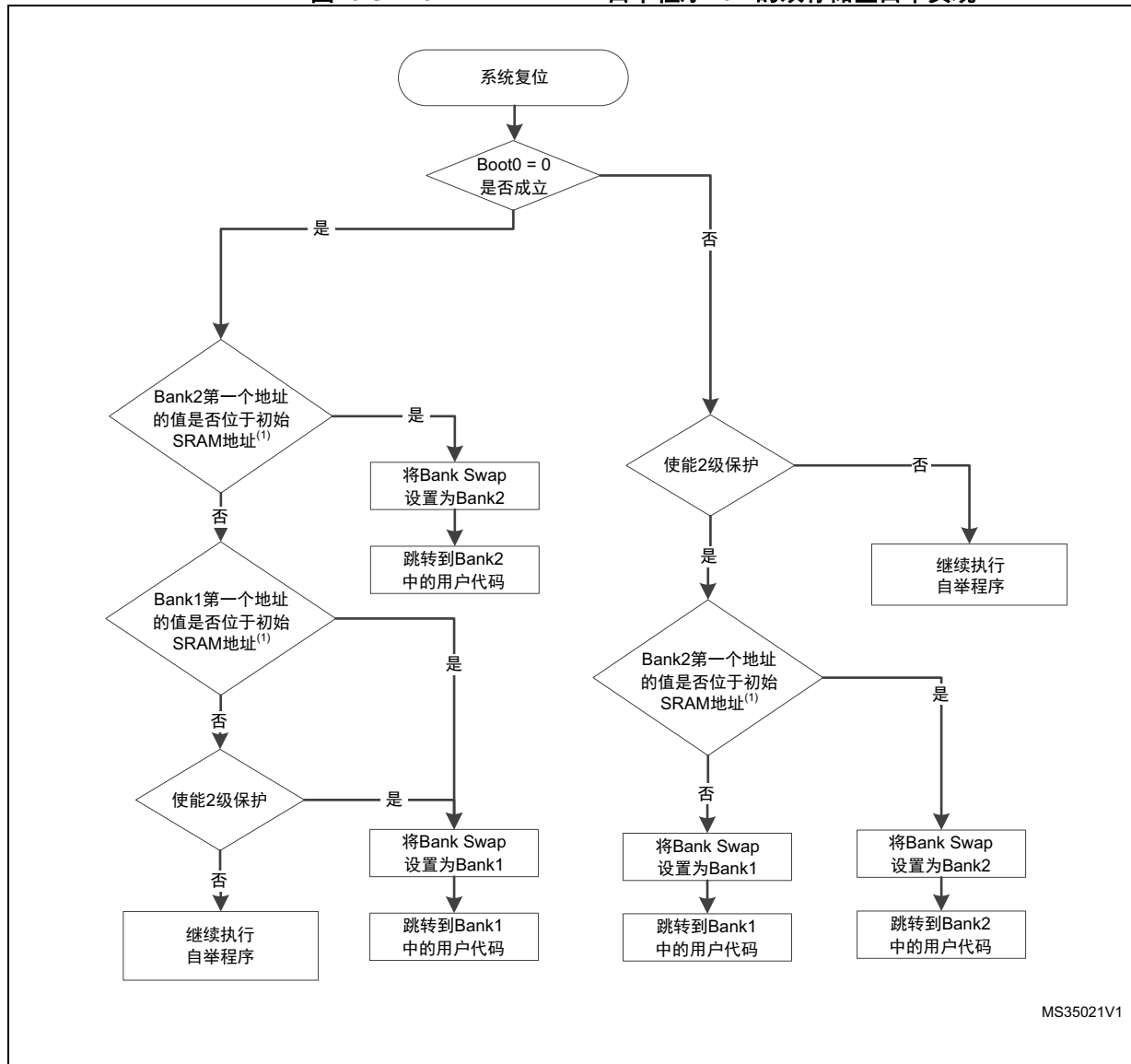
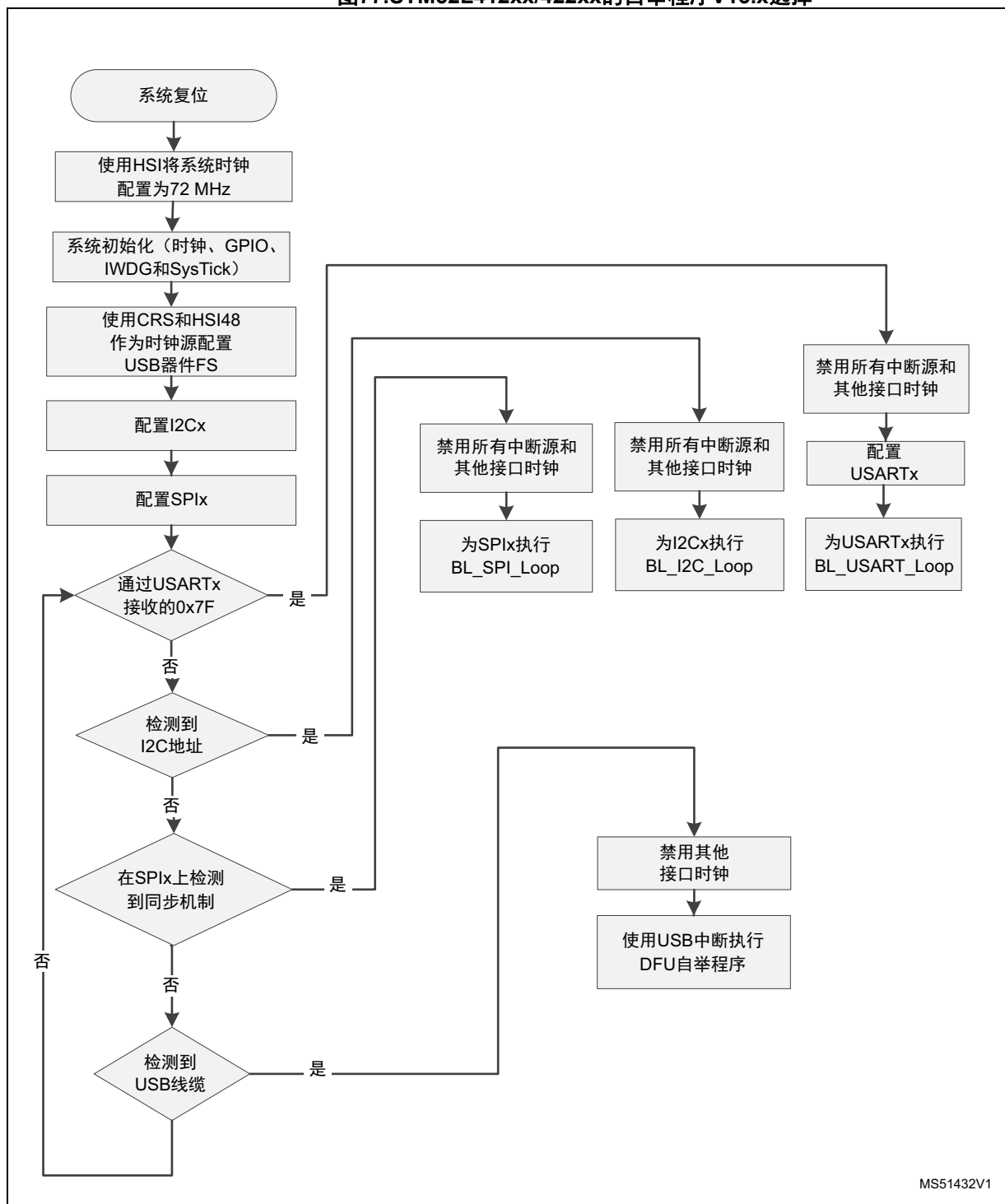


图77.STM32L412xx/422xx的自举程序V13.x选择



MS51432V1

61.3 自举程序版本

表 132列出了STM32L412xx/422xx器件自举程序版本。

表132.STM32L412xx/422xx自举程序版本

自举程序版本号	说明	已知限制
V13.1	初始自举程序版本	<ul style="list-style-type: none"> - 在连接阶段，USART以两个（而非一个）ACK字节 (0x79) 进行响应。 - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 <p>解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。</p>

62 STM32L43xxx/44xxx器件自举程序

62.1 自举程序配置

自举程序V9.1版本更新修复了USB-DFU接口的相关已知限制，在版本信息ID等于0x10的器件上实现（若需更详细信息，请参考表 134）。

STM32L43xxx/44xxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表133.系统存储器自举模式下STM32L43xxx/44xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART、I2C、SPI和USB自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock RecoverySystem, CRS），支持USB以HSI4848MHz作为时钟。
		HSE使能	HSE仅在选择了CAN接口时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		-	当HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。

表133.系统存储器自举模式下STM32L43xxx/44xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001000x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001000x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001000x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

表133.系统存储器自举模式下STM32L43xxx/44xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： – 从模式 – 全双工 – 8位MSB – 速度高达8 MHz 极性：CPOL低、CPHA低、NSS硬件
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不使用IO，则此IO可连接到GND。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为： 波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推挽，上拉模式。
	TIM16	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。
DFU自举程序	USB：	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意： 由于USB外设被自举程序使用，VDDUSBIO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻

注： 如果VDDUSB引脚没有连接到VDD，则SPIFlash存储器写操作可能因电压问题而发生错误。请参考产品的数据手册和勘误表获取更多信息。

62.2 自举程序选择

下图显示了自举程序选择机制。

图78.STM32L3x2xx/44xxx自举程序V9.x的双存储区自举实现

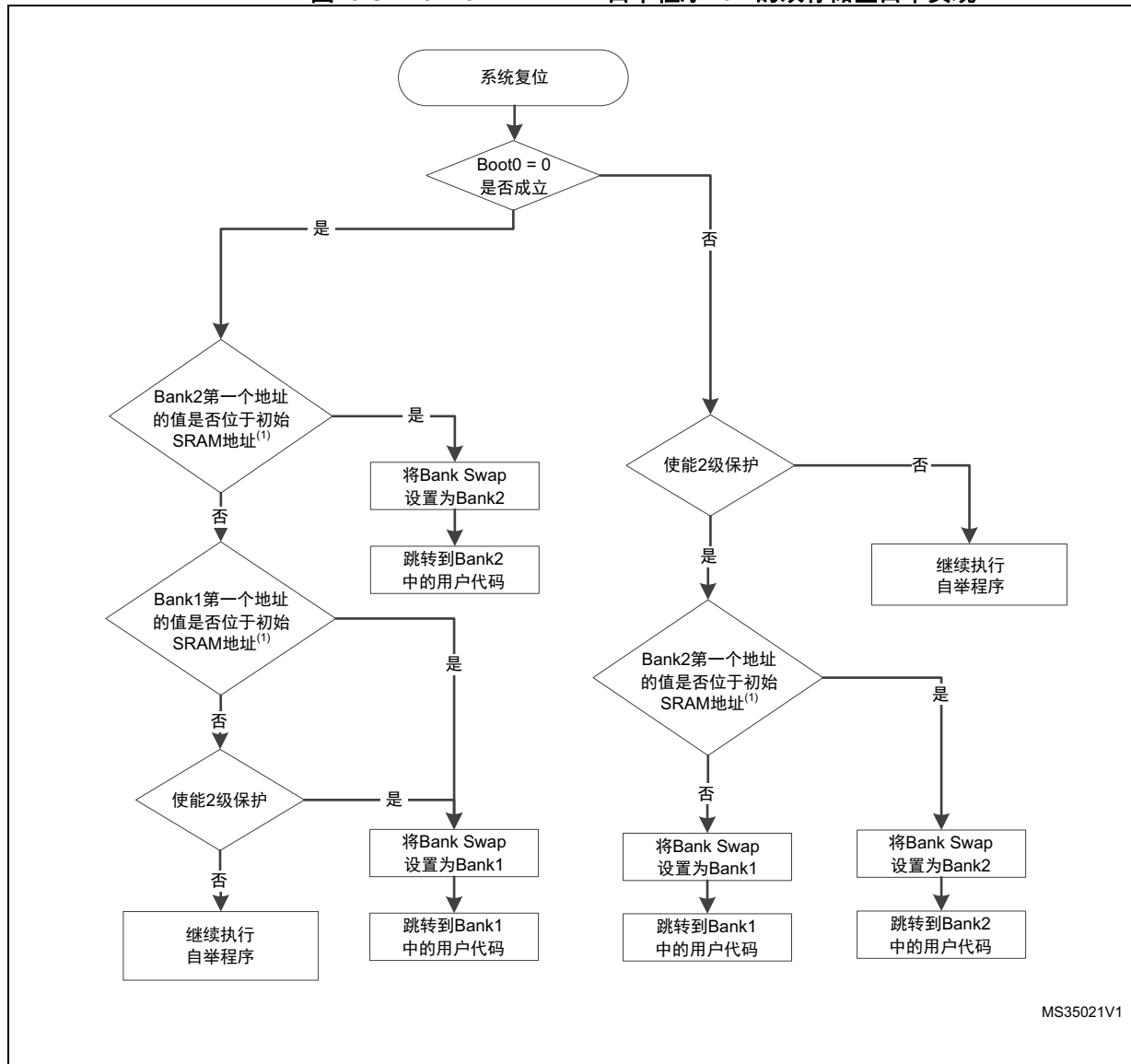
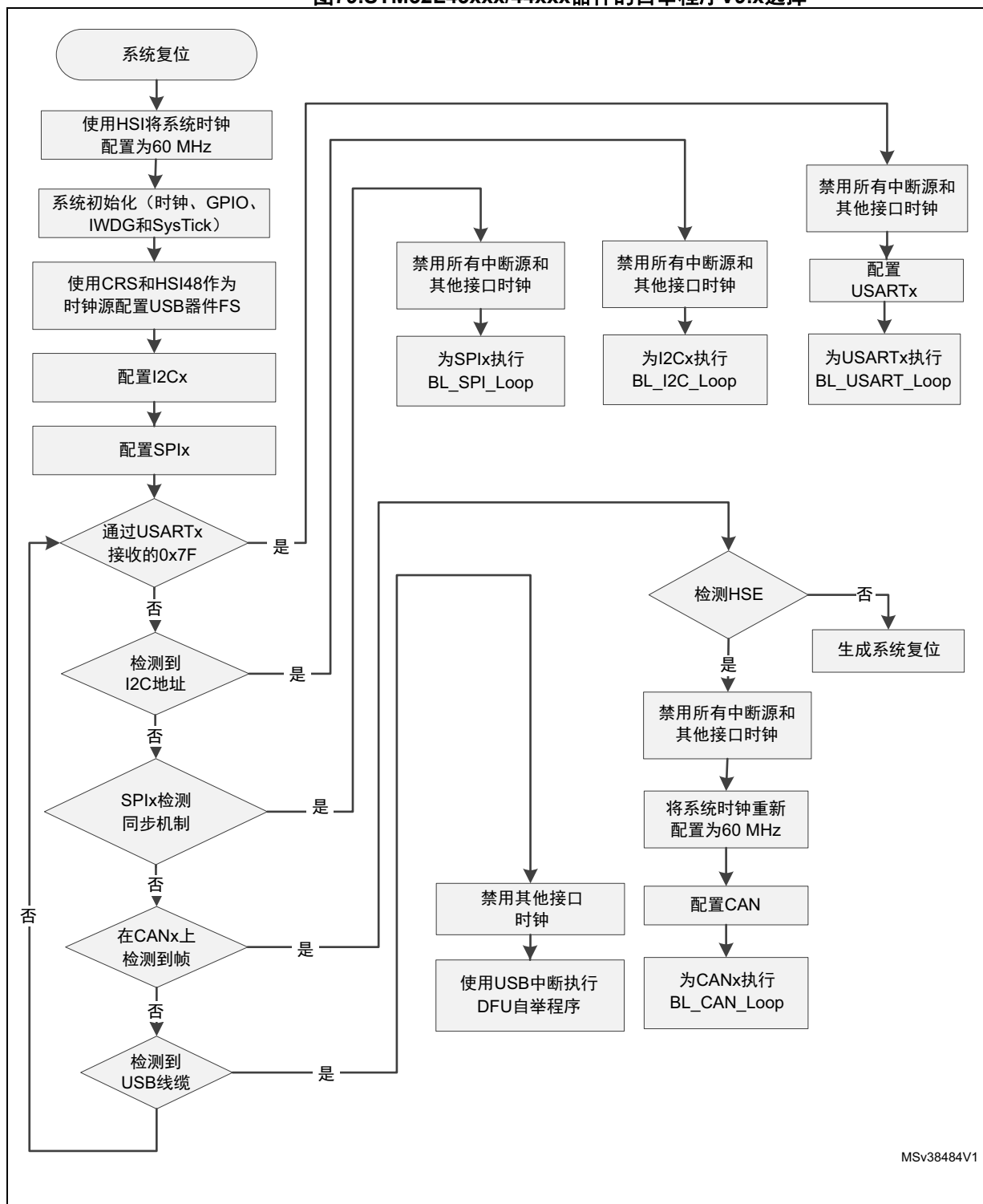


图79.STM32L43xxx/44xxx器件的自举程序V9.x选择



MSv38484V1

62.3 自举程序版本

表 134列出了STM32L43xxx/44xxx器件自举程序版本。

表134.STM32L43xxx/44xxx自举程序版本

自举程序版本号	说明	已知限制
V9.1	初始自举程序版本	<p>请检查您STM32L43xxx/44xxx器件的版本信息ID，可以在0x1FFF6FF2地址读取。</p> <p>版本信息ID等于0xFF：</p> <ul style="list-style-type: none"> - 对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。 - 对于USB-DFU接口，CRS（时钟恢复系统）未正确配置，这可能导致随机的USB通信错误（取决于温度和电压）。在大多数情况下，通信错误会表现为建立包时的“阻塞”响应。 - 使用“Go”命令时，系统自举程序去初始化会将RCC_APB1ENR寄存器中的RTCAPBEN位清零 <p>解决方案：在软件中手动调用__HAL_RCC_RTC_CLK_ENABLE()使RTCAPBEN位置位。</p> <p>版本信息ID等于0x10：无</p> <ul style="list-style-type: none"> - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 <p>解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。</p>
V9.1（续）	初始自举程序版本（续）	<ul style="list-style-type: none"> - SPI写操作失败限制： <ul style="list-style-type: none"> a.在自举程序SPI Flash存储器写操作中，一些随机64位（2个双字）可能保留空白（0xFF）。 <p>根本原因：</p> <ul style="list-style-type: none"> a.自举程序使用64位强制转换写操作（由SPI DMA中断），导致对相同Flash存储器地址的双重访问且64位未写入。 <p>解决方案：</p> <ul style="list-style-type: none"> a.WA1：在发送write命令与其ACK请求之间添加延迟。延迟的持续时间必须是256字节Flash存储器写入时间的持续时间。 b.WA2：在写入后回读，如果发生错误，再次开始写入。 c.WA3：修补RAM以便写入Flash存储器，实现无64位强制转换的存储器写入。在总编程时间方面，WA1和WA3比WA2更高效。 <p>限制的严重性：</p> <ul style="list-style-type: none"> a.限制导致客户SPI主机软件中的修改，为每次写操作增加了3-4 ms延迟。 b.由于主机无论如何都必须等待Flash存储器写入时间段结束，因此延迟并非浪费（主机将通过延迟等待，而不是通过发送ACK请求）。 c.只有SPI上不存在限制且限制无法影响USART/I2C/CAN - 如果在系统自举程序自举（通过系统复位）前应用使用RTC，则CAN接口可能不会正常工作（无法建立连接），除非执行重启或应用在系统自举程序自举前复位RTC。

63 STM32L45xxx/46xxx器件自举程序

63.1 自举程序配置

STM32L45xxx/46xxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表135.系统存储器自举模式下STM32L45xxx/46xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为72 MHz的系统时钟，以及USART、I2C、SPI和USB自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock RecoverySystem, CRS），支持USB以HSI4848 MHz作为时钟。
		HSE使能	系统时钟频率为60 MHz。 HSE仅在选择了CAN接口时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		-	当HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
	电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。

表135.系统存储器自举模式下STM32L45xxx/46xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001010x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001010x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001010x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。

表135.系统存储器自举模式下STM32L45xxx/46xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不适用IO，则此IO可连接到GND。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为： 波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推挽，上拉模式。
	TIM16	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

表135.系统存储器自举模式下STM32L45xxx/46xxx器件的配置（续）

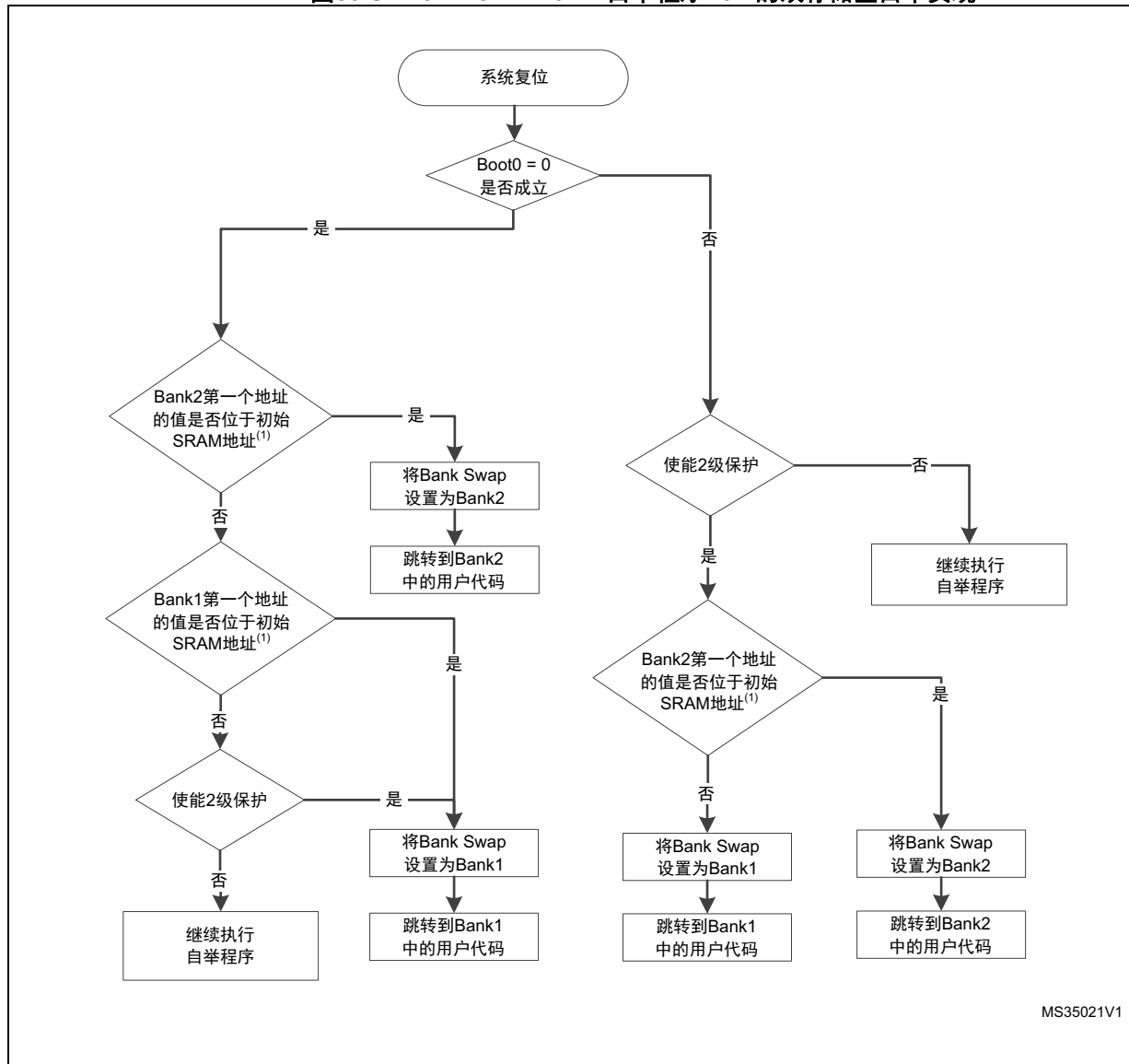
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意： 由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3V电源。
	USB_DM引脚	输入/输出	PA11: USBDM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻

注： 如果VDDUSB引脚没有连接到VDD，则SPIFlash存储器写操作可能因电压问题而发生错误。请参考产品的数据手册和勘误表获取更多信息。

63.2 自举程序选择

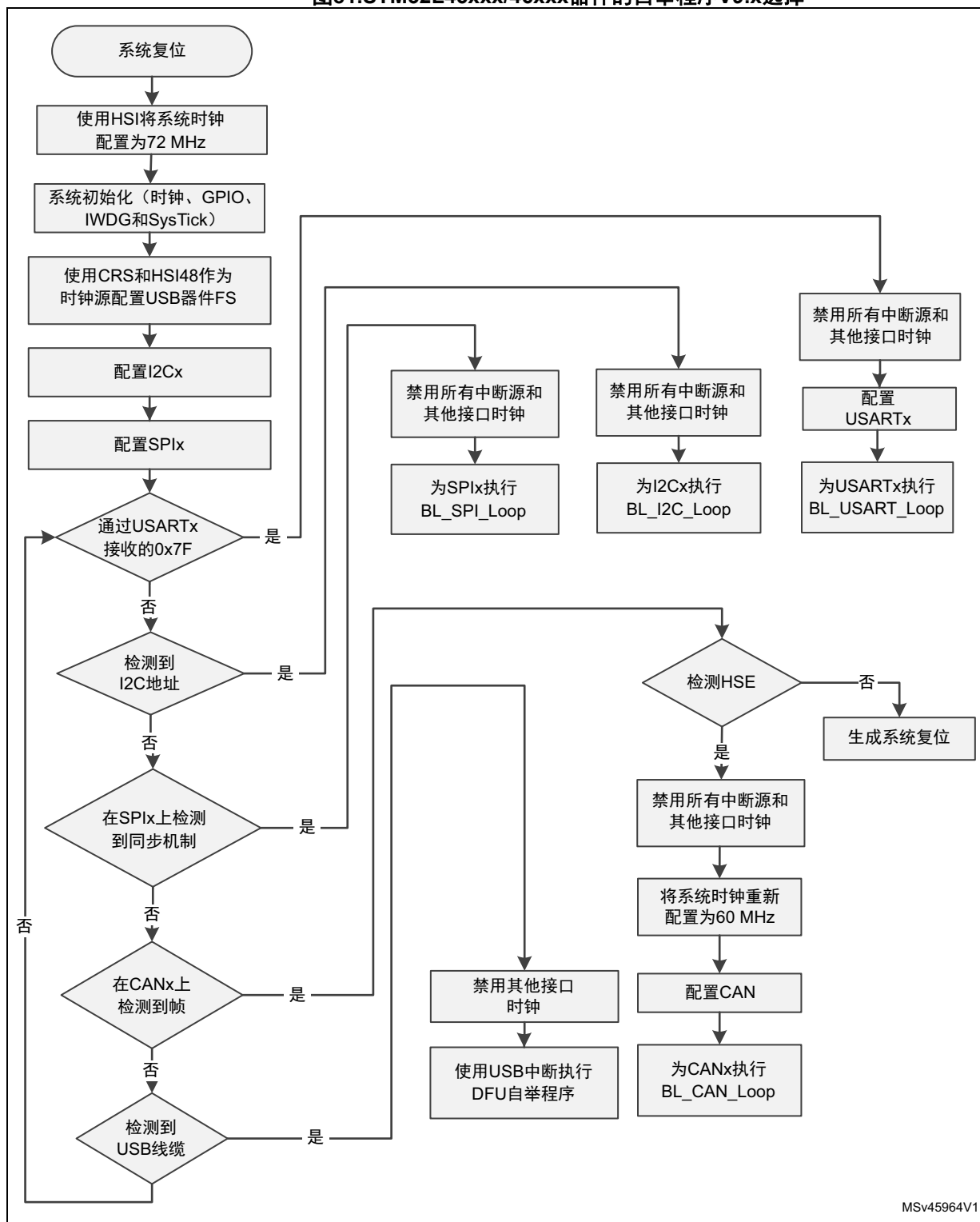
下图显示了自举程序选择机制。

图80.STM32L45xxx/46xxx自举程序V9.x的双存储区自举实现



MS35021V1

图81.STM32L45xxx/46xxx器件的自举程序V9.x选择



MSv45964V1

63.3 自举程序版本

表 136列出了STM32L45xxx/46xxx器件自举程序版本。

表136.STM32L45xxx/46xxx自举程序版本

自举程序版本号	说明	已知限制
V9.2	初始自举程序版本	<ul style="list-style-type: none"> - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。 - SPI写操作失败限制： <ul style="list-style-type: none"> a.在自举程序SPI Flash存储器写操作中，一些随机64位（2个双字）可能保留空白（0xFF）。 根本原因： <ul style="list-style-type: none"> a.自举程序使用64位强制转换写操作（由SPI DMA中断），导致对相同Flash存储器地址的双重访问且64位未写入。 解决方案： <ul style="list-style-type: none"> a.WA1：在发送write命令与其ACK请求之间添加延迟。延迟的持续时间必须是256字节Flash存储器写入时间的持续时间。 b.WA2：在写入后回读，如果发生错误，再次开始写入。 c.WA3：修补RAM以便写入Flash存储器，实现无64位强制转换的存储器写入。在总编程时间方面，WA1和WA3比WA2更高效。 限制的严重性： <ul style="list-style-type: none"> a.限制导致客户SPI主机软件中的修改，为每次写操作增加了3-4 ms延迟。 b.由于主机无论如何都必须等待Flash存储器写入时间段结束，因此延迟并非浪费（主机将通过延迟等待，而不是通过发送ACK请求）。 c.只有SPI上不存在限制且限制无法影响USART/I2C/CAN。

64 STM32L47xxx/48xxx器件自举程序

STM32L47xxx/48xxx可使用两种自举程序版本：

- V10.x支持USART、I2C和DFU（USB FS设备）。
该版本嵌入STM32L47xxx/48xxx rev.2和rev.3类器件。
- V9.x支持USART、I2C、SPI、CAN和DFU（USB FS设备）。
该版本嵌入STM32L47xxx/48xxx rev.4器件。

64.1 自举程序V10.x

64.1.1 自举程序配置

STM32L47xxx/48xxx自举程序通过应用模式7激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表137.系统存储器自举模式下STM32L47xxx/48xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为24 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	HSE仅在选择了USB接口且无LSE时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		LSE已使能	LSE用来微调MSI，MSI配置为48MHz，为USB时钟源。LSE必须等于32,768 KHz。如果未检测到LSE，则若连接了USB，可使用HSE来代替。
		MSI已使能	MSI配置为48 MHz，且用作USB时钟源。仅当检测到LSE时才使用MSI，否则（未检测到LSE），若连接了USB则使用HSE。
		-	当LSE或HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。

表137.系统存储器自举模式下STM32L47xxx/48xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000011x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1000011x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。 7位从地址为0b1000011x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式
	USB_DM引脚	输入/输出	PA11：USB DM线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻
	TIM17	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为24 MHz。

对于USARTx和I2Cx自举程序，无需外部时钟。

USB自举程序（DFU）需要LSE（低速外部时钟）或HSE（高速外部时钟）：

- 例如，LSE存在时，无论HSE是否存在，MSI都会由LSE进行配置和调节，可提供一个等于48 MHz的精确时钟，该频率是USB的时钟源。系统时钟以24 MHz的HSI为时钟。
- 假如存在HSE，则系统时钟和USB时钟会分别配置为24MHz和48MHz，以HSE作为时钟源。

64.1.2 自举程序选择

图 82和图 83显示了自举程序选择机制。

图82.STM32L47xxx/48xxx自举程序V10.x的双存储区自举实现

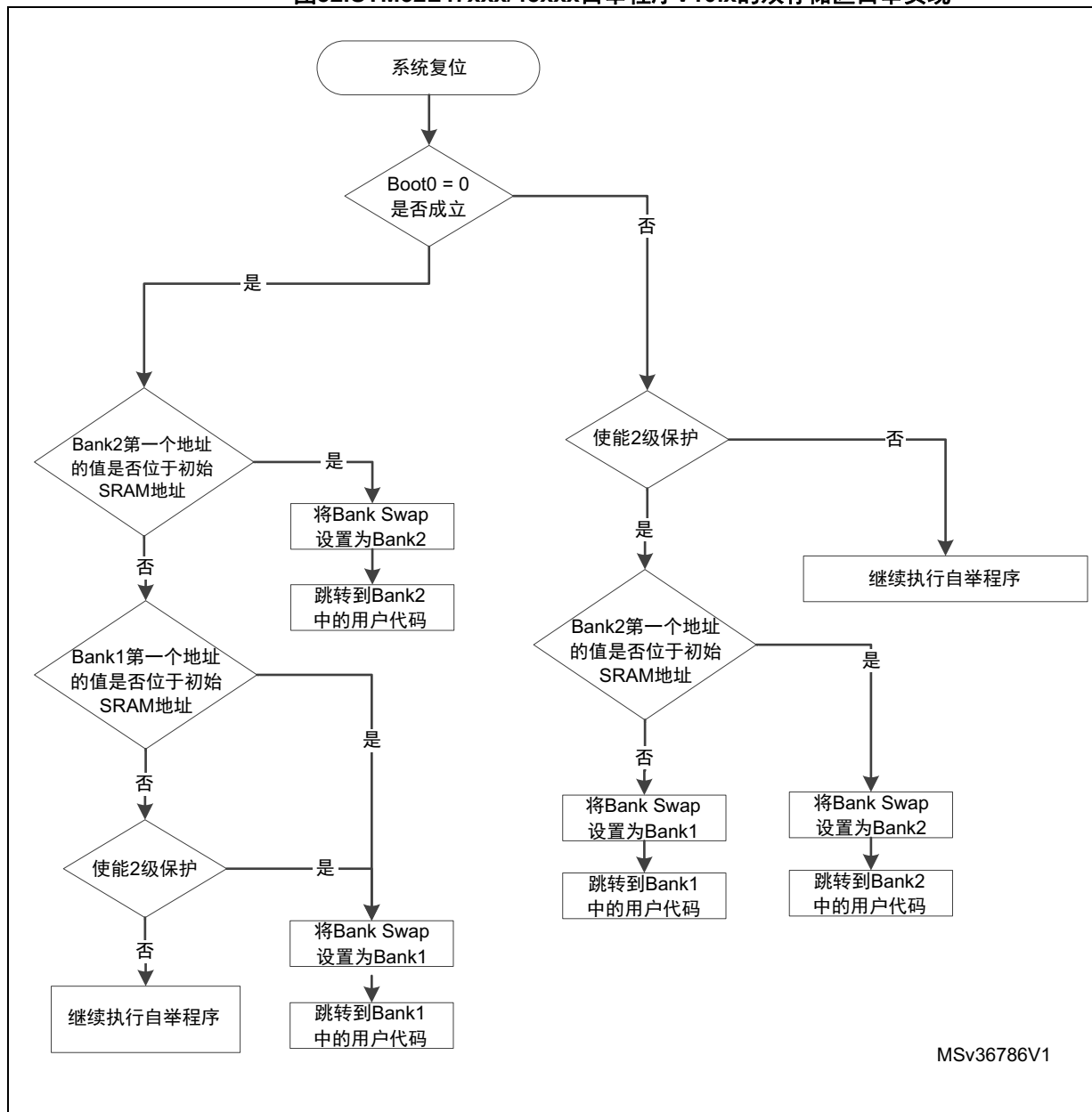
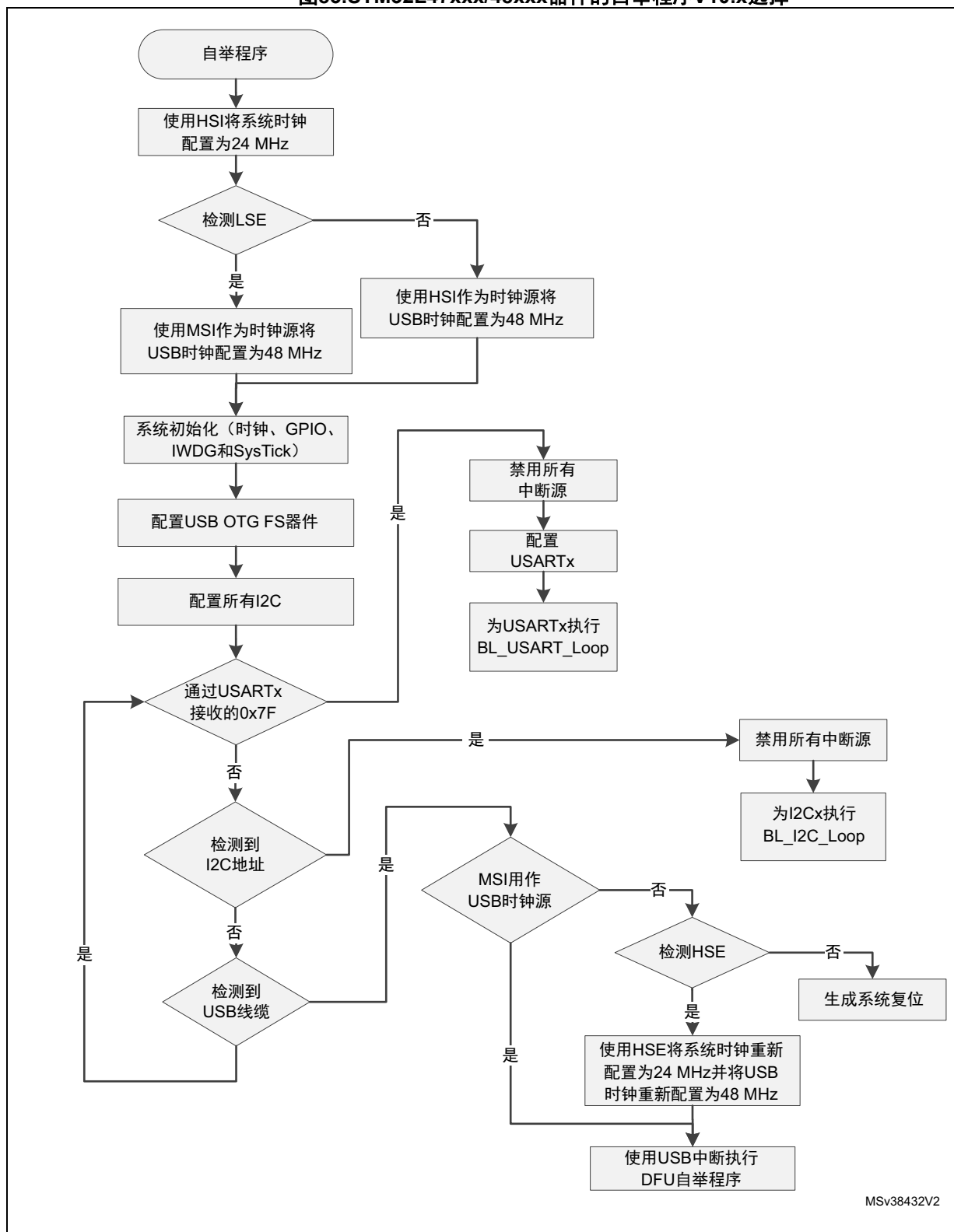


图83.STM32L47xxx/48xxx器件的自举程序V10.x选择



MSv38432V2

64.1.3 自举程序版本

下表列出了STM32L47xxx/48xxx器件自举程序V10.x版本：

表138.STM32L47xxx/48xxx自举程序V10.x版本

自举程序版本号	说明	已知限制
V10.1	初始自举程序版本	对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。 解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。 向SRAM中写入是损坏的。
V10.2	Fix write in SRAM issue	对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。 解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。
V10.3	增加了支持以MSI作为USB时钟源（MSI由LSE调节）。 更新了双存储区自举功能，以支持用户堆栈映射到SRAM2中的情况。	<ul style="list-style-type: none"> - 对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。 解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。 - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。

64.2 自举程序V9.x

64.2.1 自举程序配置

STM32L47xxx/48xxx自举程序通过应用模式7激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表139.系统存储器自举模式下STM32L47xxx/48xxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为72 MHz的系统时钟，以及USART和I2C自举程序操作，启动时HSI用作其时钟源。
		HSE使能	HSE仅在选择了USB接口且无LSE时使用。HSE必须是下列值其中之一：[24、20、18、16、12、8、6、4] MHz。 如果使用了USB，则系统时钟为72 MHz；如果使用了CAN，则系统时钟为60 MHz。
		LSE已使能	LSE用来微调MSI，MSI配置为48MHz，为USB时钟源。LSE必须等于32,768 KHz。如果未检测到LSE，则若连接了USB，可使用HSE来代替。
		MSI已使能	MSI配置为48 MHz，且用作USB时钟源。仅当检测到LSE时才使用MSI，否则（未检测到LSE），若连接了USB则使用HSE。
		CSS	当LSE或HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位。
	RAM	-	自地址0x20000000起的13KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART2处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART2处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。

表139.系统存储器自举模式下STM32L47xxx/48xxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000011x（其中x = 0对应写，x = 1对应读）。
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000011x（其中x = 0对应写，x = 1对应读）。
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达400 kHz，7位地址，从模式，模拟滤波器开启。7位从地址：0b1000011x（其中x = 0对应写，x = 1对应读）。
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： 从模式，全双工（Full Duplex），8位MSB，速度达8MHz，极性：CPOL低，CPHA低，NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表139.系统存储器自举模式下STM32L47xxx/48xxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： 从模式，全双工（Full Duplex），8位MSB， 速度达8MHz，极性：CPOL低，CPHA低，NSS 硬件
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模 式。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为：波特率125 kbps， 11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推 挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推 挽，上拉模式。
DFU自举程序	USB:	启用	USBFS配置为强制器件模式。使能USBFS中断 向量以用于USB DFU通信。 注意： 由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11引脚：USBFS DM线。用于复用推挽，无上 拉/下拉模式。
	USB_DP引脚		PA12引脚：USBFS DP线。用于复用推挽，无上 拉/下拉模式。 无需外部上拉电阻。

假如存在HSE，则系统时钟和USB时钟会分别配置为72 MHz和48 MHz，以PLL（由HSE计时）作为时钟源。

注： 如果VDDUSB引脚没有连接到VDD，则SPI Flash存储器写操作可能因电压问题而发生错误。请参考产品的数据手册和勘误表获取更多信息。

64.2.2 自举程序选择

图 84和图 85显示了自举程序选择机制。

图84.STM32L47xxx/48xxx自举程序V9.x的双存储区自举实现

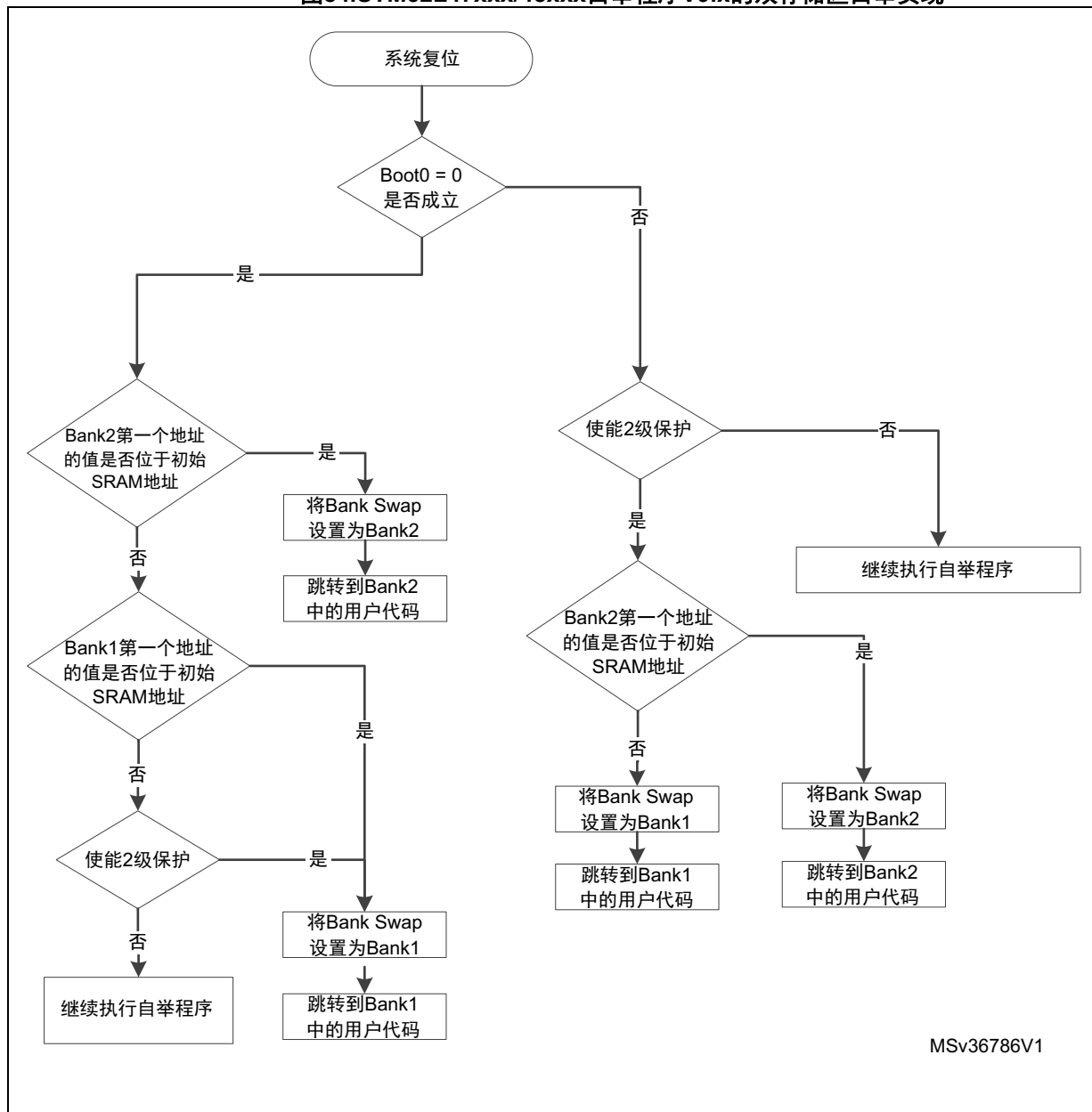
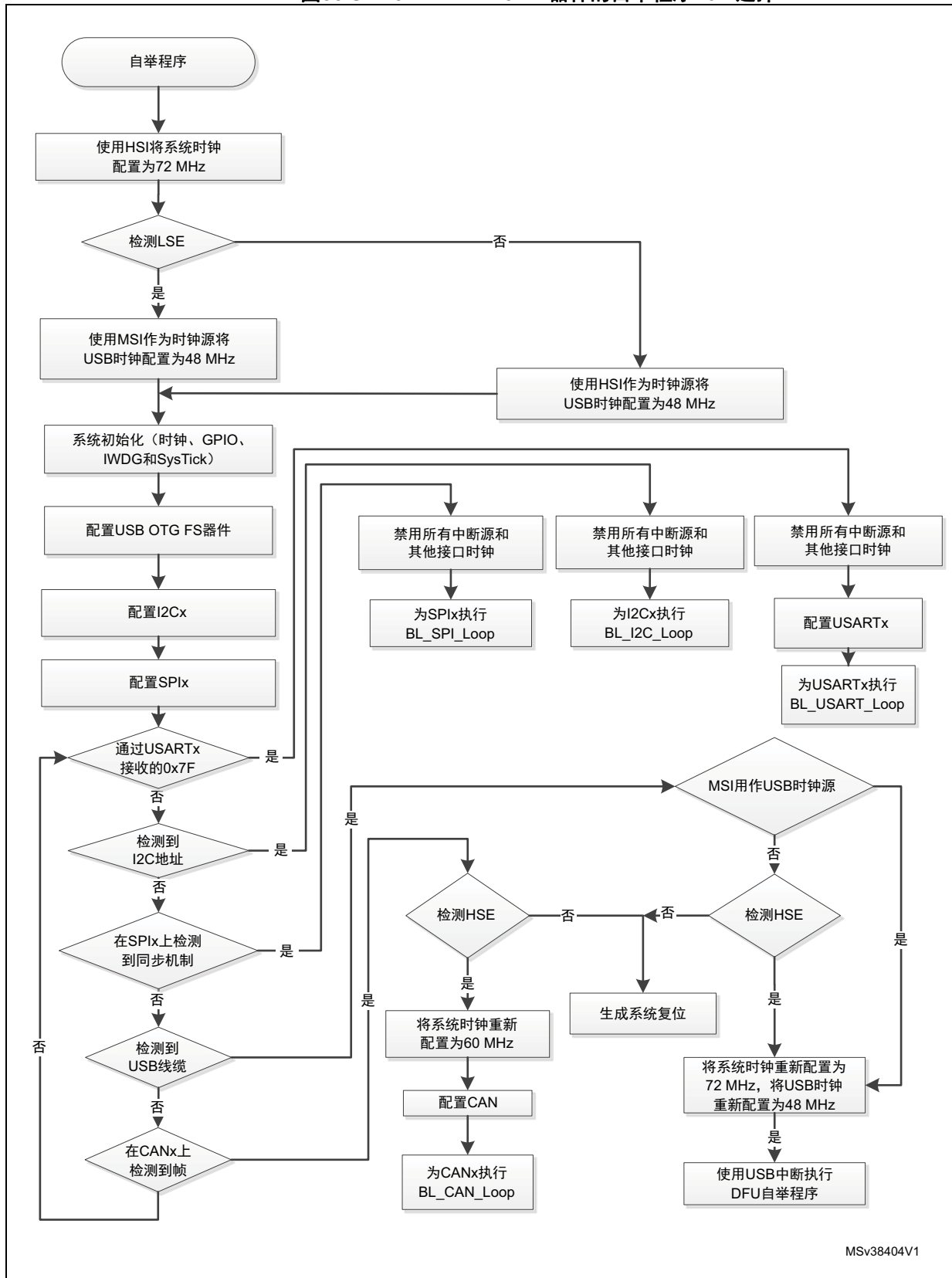


图85.STM32L47xxx/48xxx器件的自举程序V9.x选择



MSV38404V1

64.2.3 自举程序版本

下表列出了STM32L47xxx/48xxx器件自举程序V9.x版本：

表140.STM32L47xxx/48xxx自举程序V9.x版本

自举程序版本号	说明	已知限制
V9.0	初始自举程序版本	对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。 解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。 向SRAM中写入是损坏的
V9.1	废弃版本（不用）	无
V9.2	Fix write in SRAM issue	<ul style="list-style-type: none"> - 对于使用DFU接口的存储器写操作：若缓冲器大于256字节且不是8字节的倍数，则写存储器操作结果会损坏。 解决方案：若文件大于256字节，则增加填充字节，使其与8字节的整数大小对齐。 - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。

65 STM32L496xx/4A6xx器件自举程序

65.1 自举程序配置

STM32L496xx/4A6xx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 141显示了该自举程序使用的硬件资源。

表141.系统存储器自举模式下STM32L496xx/4A6xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为72 MHz的系统时钟，以及USART、I2C和SPI自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI 48 MHz作为时钟。
		HSE使能	HSE仅在选择了CAN接口时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		-	当HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入无上拉/下拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入无上拉/下拉模式。

表141.系统存储器自举模式下STM32L496xx/4A6xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入无上拉/下拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入无上拉/下拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001100x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001100x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001100x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。

表141.系统存储器自举模式下STM32L496xx/4A6xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不使用IO，则此IO可连接到GND。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为：波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推挽，上拉模式。
	TIM16	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

表141.系统存储器自举模式下STM32L496xx/4A6xx器件的配置（续）

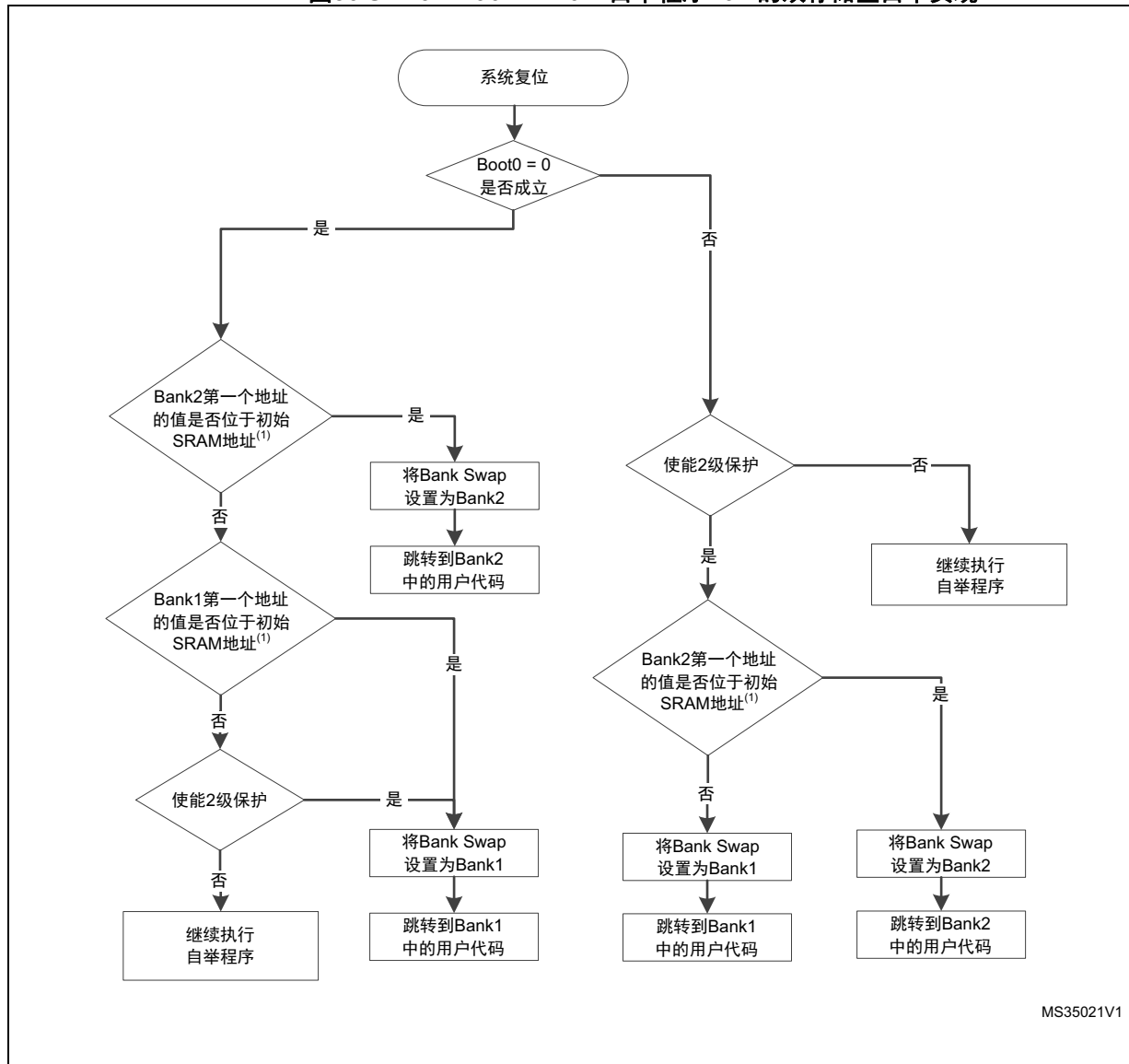
自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB OTG FS配置为强制器件模式。 使能USBOTGFS中断向量以用于USBDFU通信。 注意：由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB _{DM} 线。用于复用推挽，无上拉/下拉模式。
	USB_DP引脚		PA12: USB _{DP} 线。用于复用推挽，无上拉/下拉模式。 无需外部上拉电阻

注：如果VDDUSB引脚没有连接到VDD，则SPIFlash存储器写操作可能因电压问题而发生错误。请参考产品的数据手册和勘误表获取更多信息。

65.2 自举程序选择

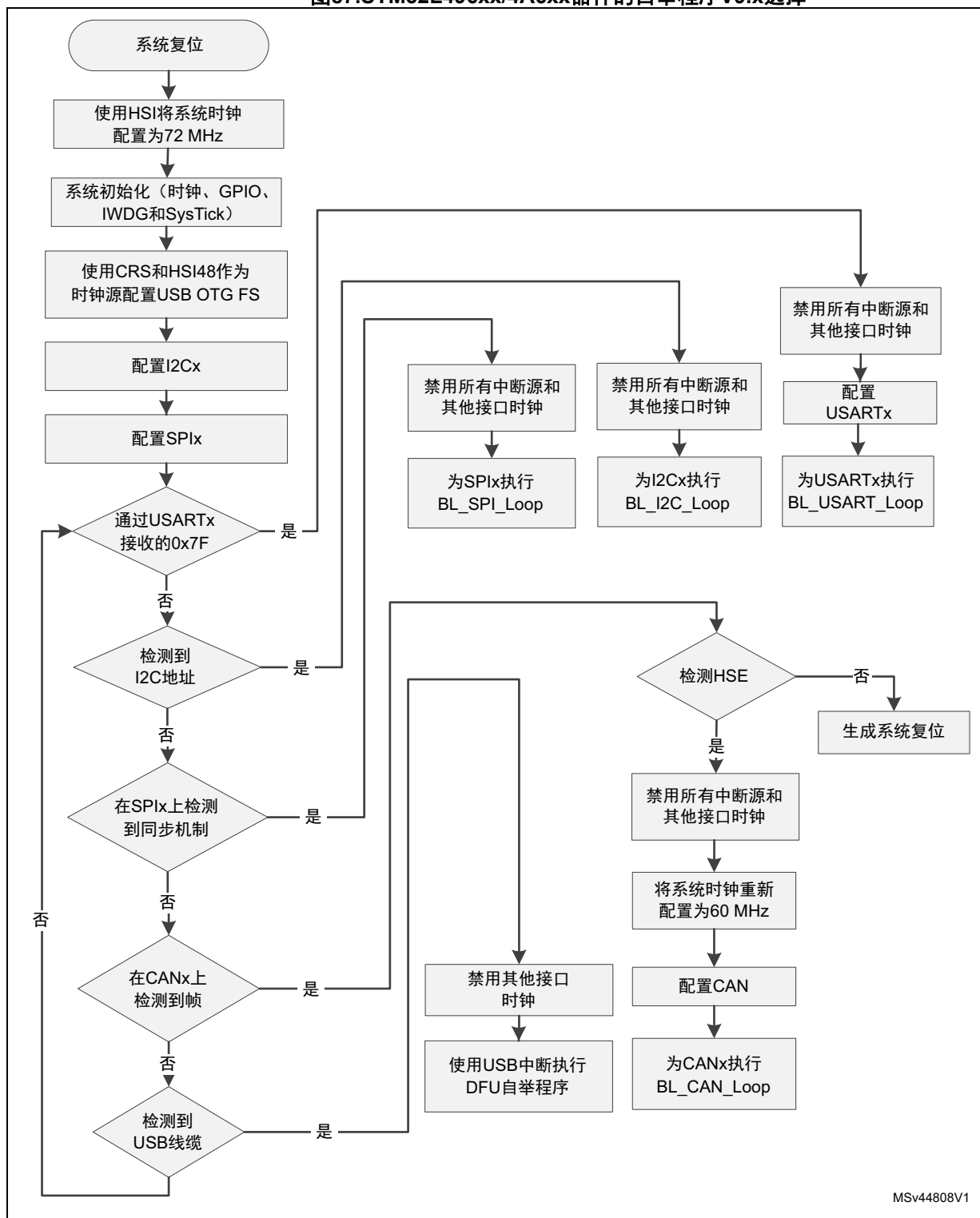
下图显示了自举程序选择机制。

图86.STM32L496xx/4A6xx自举程序V9.x的双存储区自举实现



MS35021V1

图87.STM32L496xx/4A6xx器件的自举程序V9.x选择



MSv44808V1

65.3 自举程序版本

表 142列出了STM32L496xx/4A6xx器件自举程序版本。

表142.STM32L496xx/4A6xx自举程序版本

自举程序版本号	说明	已知限制
V9.3	初始自举程序版本	<ul style="list-style-type: none"> - 自举程序器件将中止存储区擦除命令，并向主机发送NACK (0x1F)。解决方案：使用Erase指令 (0x44)，通过页面擦除执行存储区擦除操作。 - SPI写操作失败 <ul style="list-style-type: none"> 限制： a. 在自举程序SPI Flash存储器写操作中，一些随机64位（2个双字）可能保留空白 (0xFF)。 根本原因： a. 自举程序使用64位强制转换写操作（由SPI DMA中断），导致对相同Flash存储器地址的双重访问且64位未写入。 解决方案： a. WA1：在发送write命令与其ACK请求之间添加延迟。延迟的持续时间必须是256字节Flash存储器写入时间的持续时间。 b. WA2：在写入后回读，如果发生错误，再次开始写入。 c. WA3：修补RAM以便写入Flash存储器，实现无64位强制转换的存储器写入。 在总编程时间方面，WA1和WA3比WA2更高效。 限制的严重性： a. 限制导致客户SPI主机软件中的修改，为每次写操作增加了3-4 ms延迟。 b. 由于主机无论如何都必须等待Flash存储器写入时间段结束，因此延迟并非浪费（主机将通过延迟等待，而不是通过发送ACK请求）。 c. 只有SPI上不存在限制且限制无法影响USART/I2C/CAN。 - 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。 解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。

66 STM32L4P5xx/4Q5xx器件自举程序

66.1 自举程序配置

STM32L4P5xx/4Q5xx自举程序通过应用模式7激活（详述参见表 2：自举程序激活模式）。表 145显示了该自举程序使用的硬件资源。

表143.系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART、I2C、SPI和USB自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI 48 MHz作为时钟。
		HSE使能	HSE仅在选择了CAN接口时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		-	当HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式

表143.系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011011x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011011x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011011x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。

表143.系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为： 波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推挽，上拉模式。

表143.系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意: 由于USB外设被自举程序使用, VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB _{DM} 线。用于复用推挽, 无上拉/下拉模式。
	USB_DP引脚		PA12: USB _{DP} 线。用于复用推挽, 无上拉/下拉模式。 无需外部上拉电阻

66.2 自举程序选择

图 90和图 91显示了自举程序选择机制。

图88.STM32L4P5xx/4Q5xx自举程序V9.x的双存储区自举实现

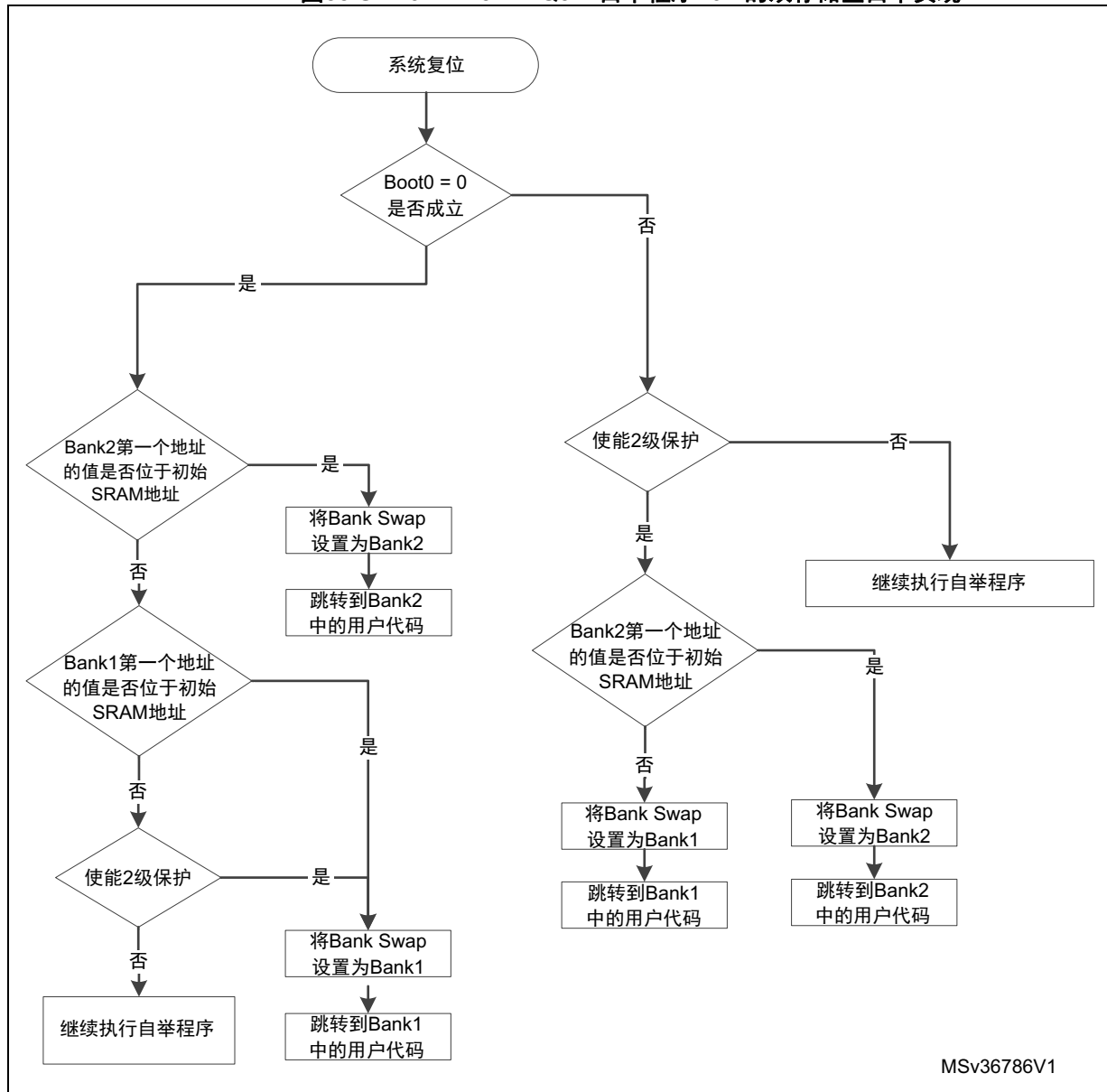
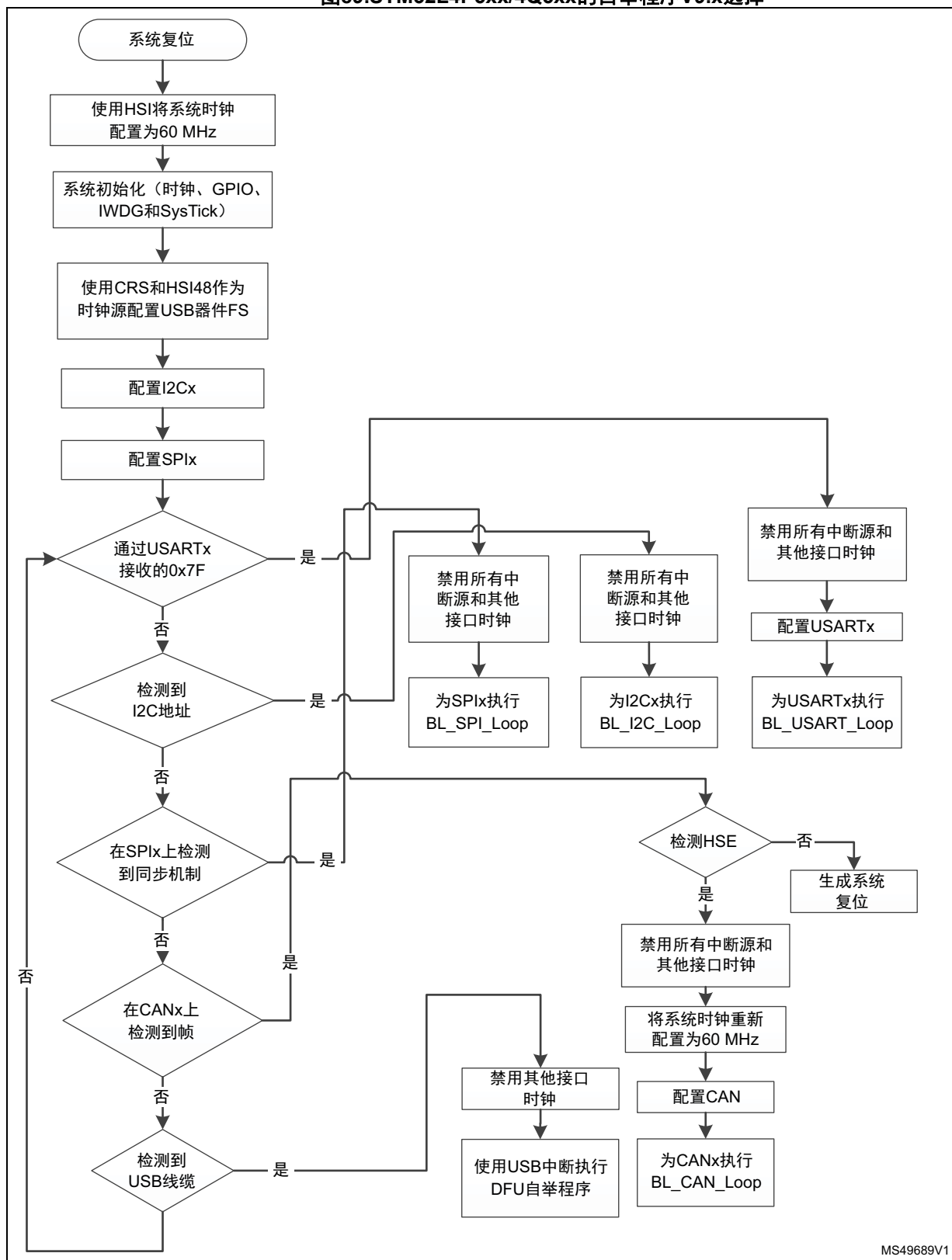


图89.STM32L4P5xx/4Q5xx的自举程序V9.x选择



MS49689V1

66.3 自举程序版本

表 144列出了STM32L4P5xx/4Q5xx器件自举程序版本。

表144.STM32L4P5xx/4Q5xx自举程序版本

自举程序版本号	说明	已知限制
V9.0	cut1.0样品上的初始自举程序版本	<p>– 由于自举程序在必须使用半字访问执行PcROP访问时使用字节访问，因此不能写入PcROP选项字节。</p> <p>解决方案：使用加载程序接口加载SRAM中的代码段然后跳转到这里，该代码将写入PcROP值。</p>

67 STM32L4Rxxx/4Sxxx器件自举程序

67.1 自举程序配置

STM32L4Rxx/4Sxx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 145显示了该自举程序使用的硬件资源。

表145.系统存储器自举模式下STM32L4Rxxx/4Sxxx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	对于配置为60 MHz的系统时钟，以及USART、I2C、SPI和USB自举程序操作，启动时HSI用作其时钟源。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI 48 MHz作为时钟。
		HSE使能	HSE仅在选择了CAN接口时使用。HSE必须是下列值其中之一：[24、20、18、16、12、9、8、6、4] MHz。
		-	当HSE使能时，使能时钟安全系统（CSS）中断。外部时钟发生任何故障（或被移除）都将产生系统复位
	RAM	-	自地址0x20000000起的12 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28672字节包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
电源	-	如果选择了电压调节范围2，则DFU不能用于与自举程序之间的通信。自举程序固件不能配置PWR_CR1寄存器中的电压调节范围值。	
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于输入无上拉/下拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于输入无上拉/下拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于输入上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于输入上拉模式。

表145.系统存储器自举模式下STM32L4Rxxx/4Sxxx器件的配置（续）

自举程序	功能/外设	状态	备注
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于输入上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于输入上拉模式。
USARTx自举程序	SysTick定时器	启用	用于自动检测用于USARTx自举程序的主机串口波特率。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010000x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010000x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1010000x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。

表145.系统存储器自举模式下STM32L4Rxxx/4Sxxx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。
CAN1自举程序	CAN1	启用	初始化后，CAN1的配置为： 波特率125 kbps，11位标识符。
	CAN1_RX引脚	输入	PB8引脚：CAN1处于接收模式。用于复用推挽，上拉模式。
	CAN1_TX引脚	输出	PB9引脚：CAN1处于发送模式。用于复用推挽，上拉模式。
	TIM16	启用	该定时器用来确定HSE的值。一旦确定了HSE频率，可利用PLL和HSE将系统时钟配置为60 MHz。

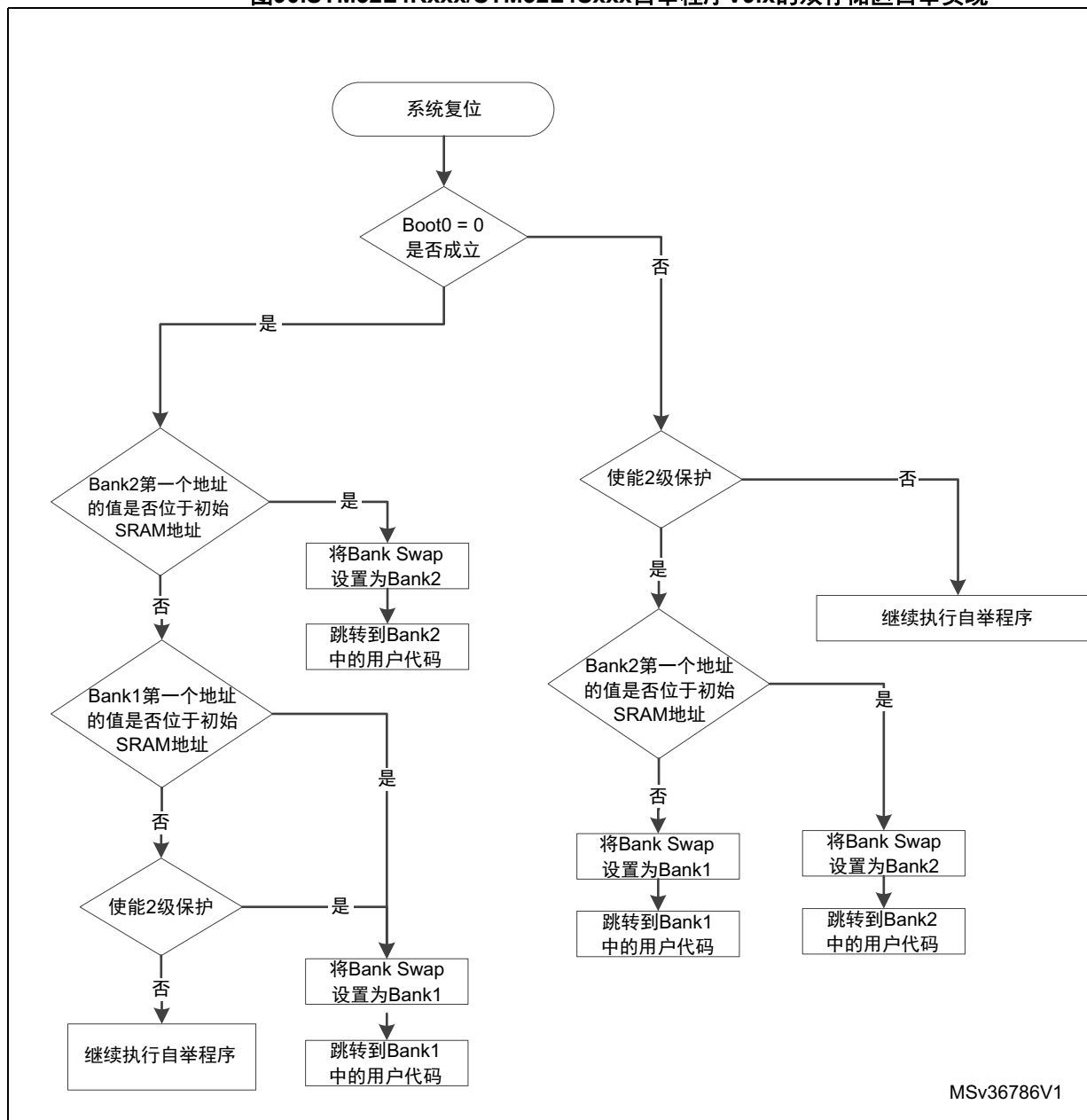
表145.系统存储器自举模式下STM32L4Rxxx/4Sxxx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意: 由于USB外设被自举程序使用, VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。
	USB_DP引脚		PA12: USB DP线 无需外部上拉电阻

67.2 自举程序选择

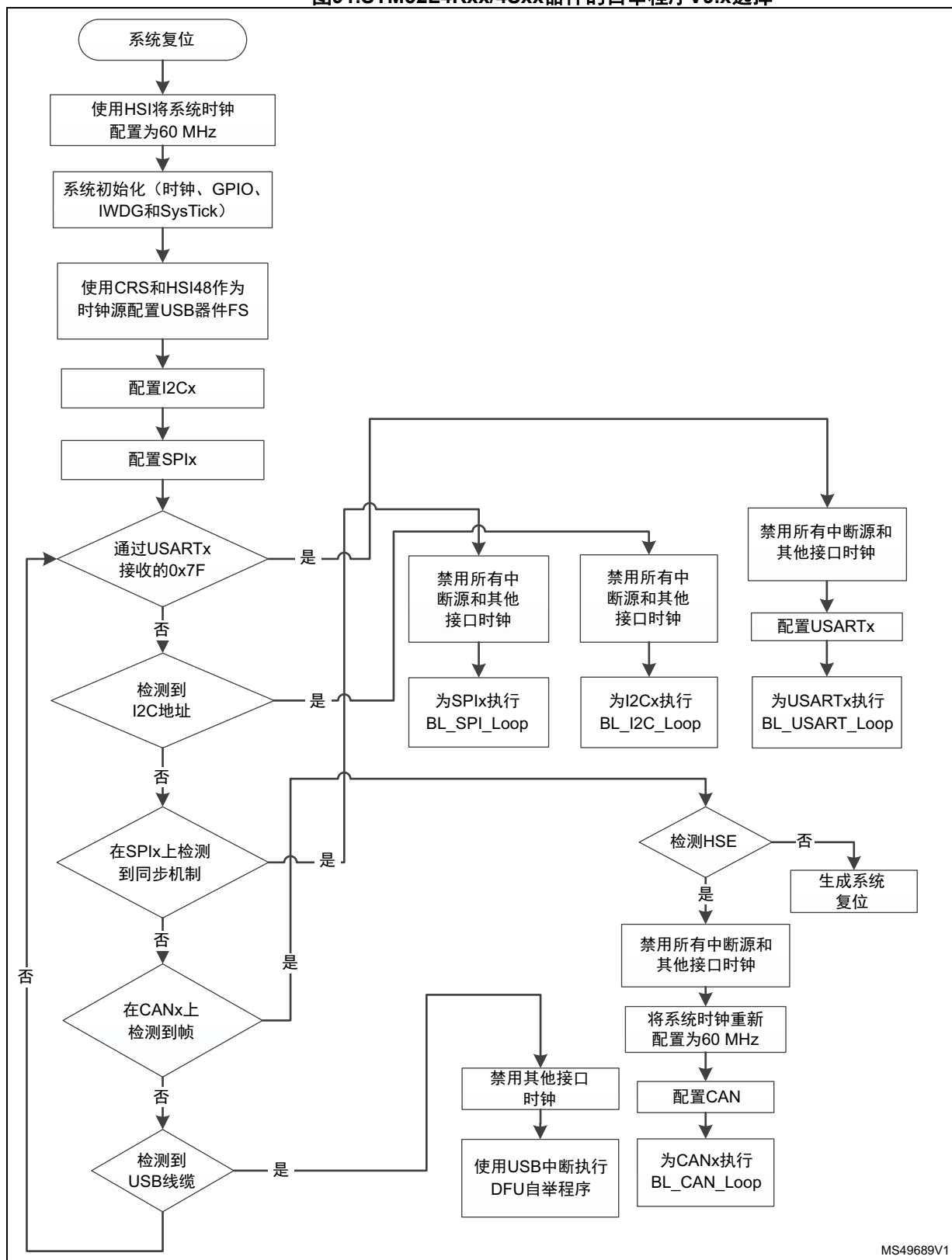
图 90和图 91显示了自举程序选择机制。

图90.STM32L4Rxxx/STM32L4Sxxx自举程序V9.x的双存储区自举实现



MSv36786V1

图91.STM32L4Rxx/4Sxx器件的自举程序V9.x选择



MS49689V1



67.3 自举程序版本

表 146列出了STM32L4Rxx/4Sxx器件自举程序版本。

表146.STM32L4Rxx/4Sxx自举程序版本

自举程序版本号	说明	已知限制
V9.0	cut1.0示例上的初始自举程序版本	– 无

68 STM32L552xx/STM32L562xx器件自举程序

68.1 自举程序配置

STM32L552xx/562xx自举程序通过应用模式12激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表147.系统存储器自举模式下STM32L552xx/562xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于60 MHz（使用由HSI驱动的PLL）。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI 48 MHz作为时钟。
		-	对FDCAN使用PLLQ提供的20 MHz时钟频率
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x0BF90000起的32 KB空间。
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。

表147.系统存储器自举模式下STM32L552xx/562xx器件的配置（续）

自举程序	功能/外设	状态	备注
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0101100x (其中x = 0对应写，x = 1对应读)
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0101100x (其中x = 0对应写，x = 1对应读)
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b0101100x (其中x = 0对应写，x = 1对应读)
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

表147.系统存储器自举模式下STM32L552xx/562xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。
SPI3自举程序	SPI3	启用	SPI配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PB5引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PG10引脚：从数据输出线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PG9引脚：从时钟线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PG12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。
FDCAN自举程序	FDCAN1	启用	初始化后，FDCAN1的配置为： 比特率为0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式 = FDCAN_MODE_NORMAL AutoRetransmission = 使能 TransmitPause = 禁用 ProtocolException = 使能
	FDCAN1_Rx引脚	输入/	PB9引脚：FDCAN1处于接收模式。用于复用推挽，上拉模式。
	FDCAN1_Tx引脚	输出	PB8引脚：FDCAN1处于发送模式。用于复用推挽，上拉模式。

表147.系统存储器自举模式下STM32L552xx/562xx器件的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意: 由于USB外设被自举程序使用, VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

表148.STM32L552cc/562xx特殊命令

支持的特殊命令（USART/I2C/SPI/FDCAN） 操作码 - 0x50							
功能	子操作码 (2字节)	发送的数据 量 (2字节)	发送的数据	接收的数据 量	接收的数据	接收的状态 数据量 (2字节)	接收的状态数据
Trust Zone禁用 当TZEN=1且RDP = 1时必须运行	0x82	0x4	0x0	0x0	NA	0x1	0x0
从RDP L1降级为 RDP 0.5 当TZEN=1且RDP = 1时必须运行	0x82	0x4	0x1	0x0	NA	0x1	0x0

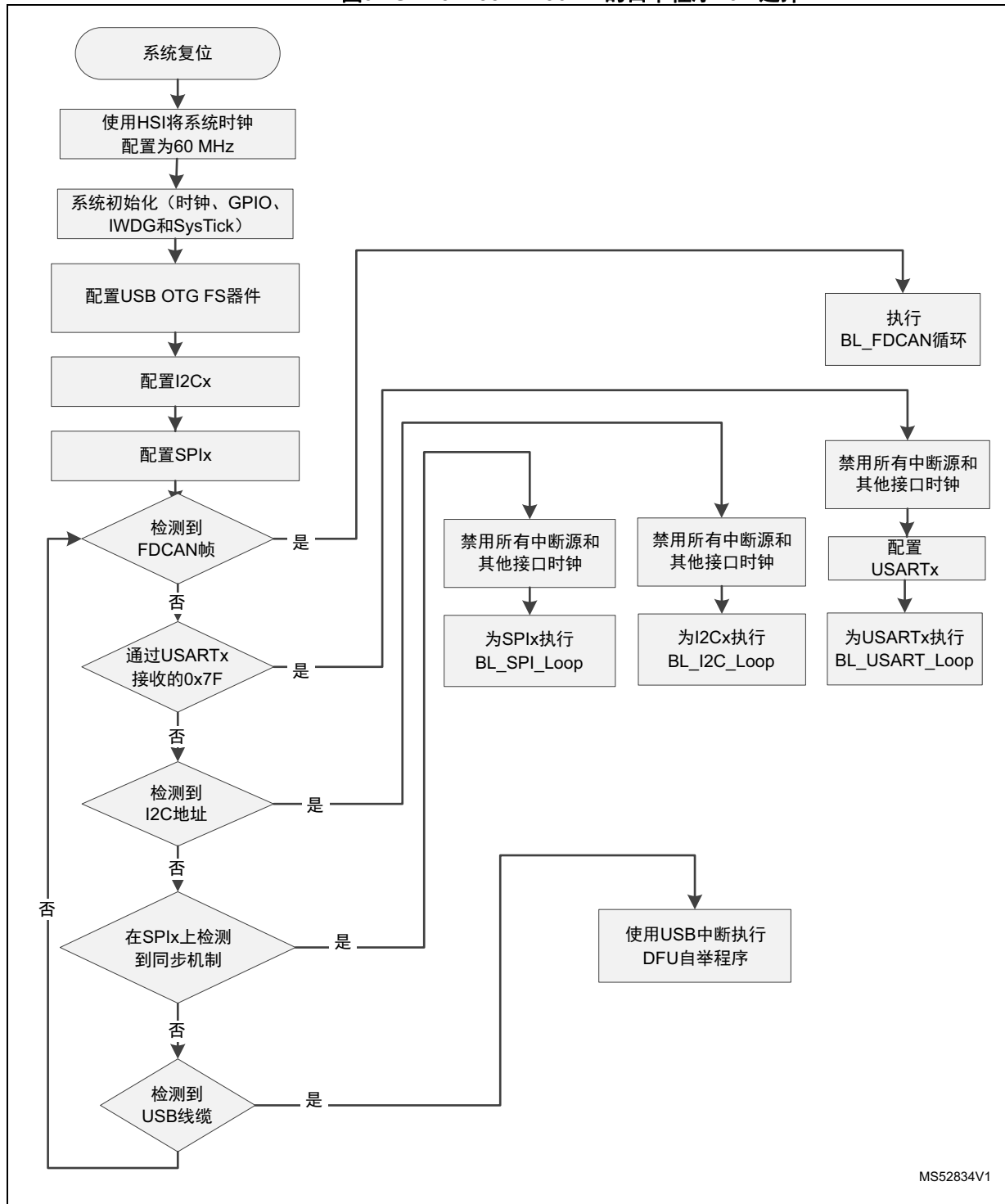
注: 根据USB协议规范, USB特殊命令与其他协议相比稍有不同:

- 不使用操作码, 直接使用子操作码
- 在一个字节而非两个字节中处理子操作码
- 通过USB帧按字节发送数据。无需增加要发送的数据量
- 基于USB原生协议格式化返回的数据和状态

68.2 自举程序选择

下图显示了自举程序选择机制。

图92.STM32L552xx/562xx的自举程序V9.x选择



68.3 自举程序版本

表 149列出了STM32L552xx/562xx器件自举程序版本。

表149.STM32L552xx/562xx自举程序版本

自举程序版本号	说明	已知限制
V13.0	cut1.0样品上的初始自举程序版本	<ul style="list-style-type: none"> - USART3不工作 - SPI3不工作 - USB-DFU上的OB启动不工作 - 所有协议中无SRAM2读取/写入 - 只在USART/I2C上实现了安全选项字节读取 - 在RDP降级时自动执行从TZen = 1到TZen = 0的降级
V9.0	只有cut2.0支持的版本 <ul style="list-style-type: none"> - 修复上一个版本的所有问题 - 增加了FDCAN支持 - 为TZen禁用增加了新命令 - 支持销售型号256KB 	<ul style="list-style-type: none"> - BL的所有接口并非都能用来将TZen选项字节置为1 无WA可用 - 使用BL接口时不能设置RDP 0.5级和RDP 0.5级中的选项字节 无WA可用 - 在TZen = 1中使能HWIWDG选项字节时发生多次复位 无WA可用 - 当TZen = 1且RDP级别为0时，不能将安全选项字节置位 无WA可用 - USB上的“Go”命令不工作
V9.1	<ul style="list-style-type: none"> - 修复上一个版本的所有已知限制 - 增加使能BOOT_LOCK BL命令 - 增加RDP L1至0.5降级支持 	使用FDCAN接口时选项字节编程不正常工作 这使得选项字节的更改在重启后才生效。
V9.2	<ul style="list-style-type: none"> - 修复上一个版本的所有已知限制 - Silicon Revision Z的版本 	无

注：在跳转到BL时，必须禁用缓存。

69 STM32WB10xx/15xx器件自举程序

69.1 自举程序配置

STM32WB10xx/15xx自举程序通过应用模式6激活（详述参见表 2：自举程序激活模式）。表 152显示了该自举程序使用的硬件资源。

表150.系统存储器自举模式下STM32WB10xx/15xx的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	MSI已使能	系统时钟频率为64 MHz（使用由MSI计时的PLL）。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001111x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。

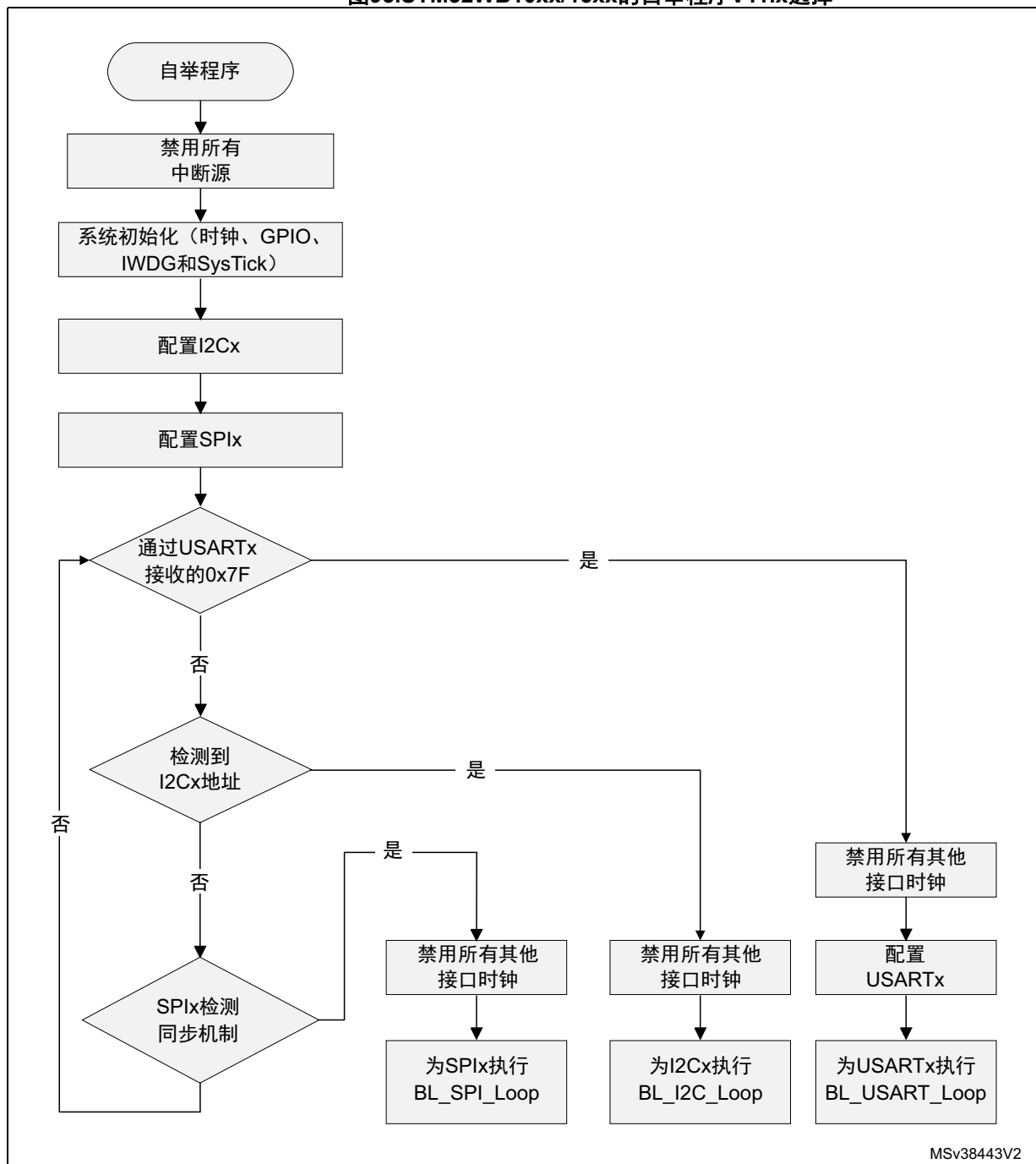
表150.系统存储器自举模式下STM32WB10xx/15xx的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

69.2 自举程序选择

下图显示了自举程序选择机制。

图93.STM32WB10xx/15xx的自举程序V11.x选择



69.3 自举程序版本

表151.STM32WB10xx/15xx自举程序版本

自举程序版本号	说明	已知限制
V11.1	初始自举程序版本	<ul style="list-style-type: none">- I2C Write Protect命令（0x73）执行Read Unprotect而不是禁用写保护。- 解决方案：使用No-Stretch Write Unprotect命令（0x74）正确执行解除写保护操作

70 STM32WB30xx/35xx/50xx/55xx器件自举程序

70.1 自举程序配置

STM32WBxxx自举程序通过应用模式16激活（详述参见表 2：自举程序激活模式）。表 152 显示了该自举程序使用的硬件资源。

表152.系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	MSI已使能	系统时钟频率为64 MHz（使用由MSI计时的PLL）。
		-	对于DFU自举程序，使能时钟恢复系统（Clock Recovery System, CRS），支持USB以HSI 48 MHz作为时钟。
	RAM	-	自地址0x20000000起的20 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的28 KB包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001111x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏无上拉/下拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1001111x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏无上拉/下拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏无上拉/下拉模式下使用。

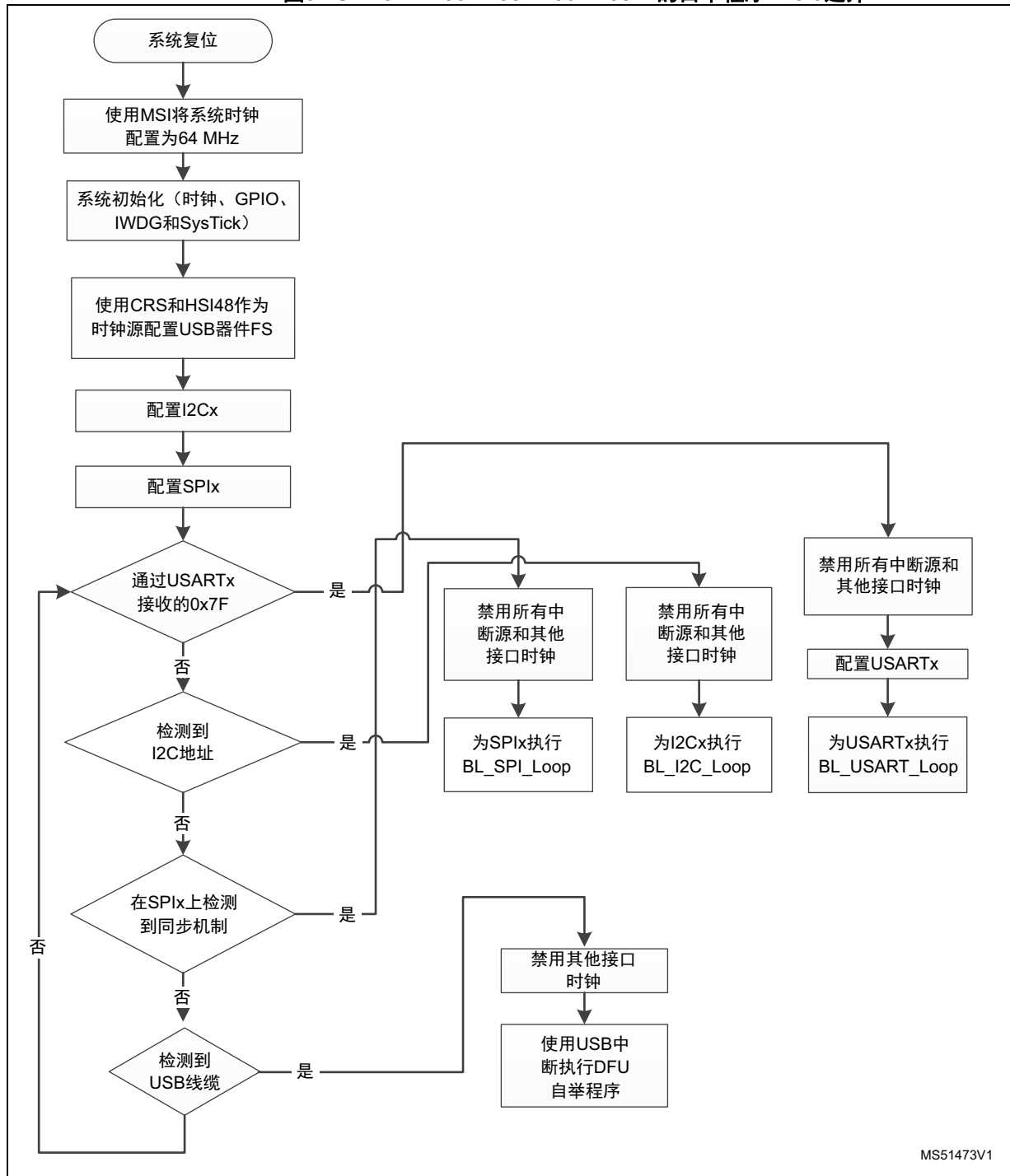
表152.系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx的配置（续）

自举程序	功能/外设	状态	备注
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不使用IO，则此IO可连接到GND。
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意：由于USB外设被自举程序使用，VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11：USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12：USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

70.2 自举程序选择

下图显示了自举程序选择机制。

图94.STM32WB30xx/35xx/50xx/55xx的自举程序V13.0选择



70.3 自举程序版本

表153.STM32WB30xx/35xx/50xx/55xx自举程序版本

自举程序版本号	说明	已知限制
V13.5	初始自举程序版本	<ul style="list-style-type: none"> - 由于在命令结束时执行了NVIC_SystemReset而非Flash选项字节重载，Readout Unprotect命令不正常工作。这使得RDP级别的更改在重启后才生效。 - I2C Write Protect命令（0x73）执行Read Unprotect而不是禁用写保护。 解决方案：使用No-Stretch Write Unprotect命令（0x74）正确执行解除写保护操作

注： 在操作进行期间执行多次复位时的不稳定性导致USART自举程序发生上溢或FrameError错误且不可恢复，除非执行硬件复位。已通过FUS V1.0.1和V1.0.2中的解决方案修复。

71 STM32WLE5xx/55xx器件自举程序

71.1 自举程序配置

STM32WLE5xx/55xx自举程序通过应用模式13激活（详述参见表 2：自举程序激活模式）。表 154显示了该自举程序使用的硬件资源。

表154.系统存储器自举模式下STM32WLE5xx/55xx器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于48 MHz（使用由HSI驱动的PLL）。
	RAM	-	自地址0x20000000起的8 KB空间供自举程序固件使用
	系统存储器	-	自地址0x1FFF0000起的16 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。 注：如果SPI主设备不适用IO，则此IO可连接到GND。

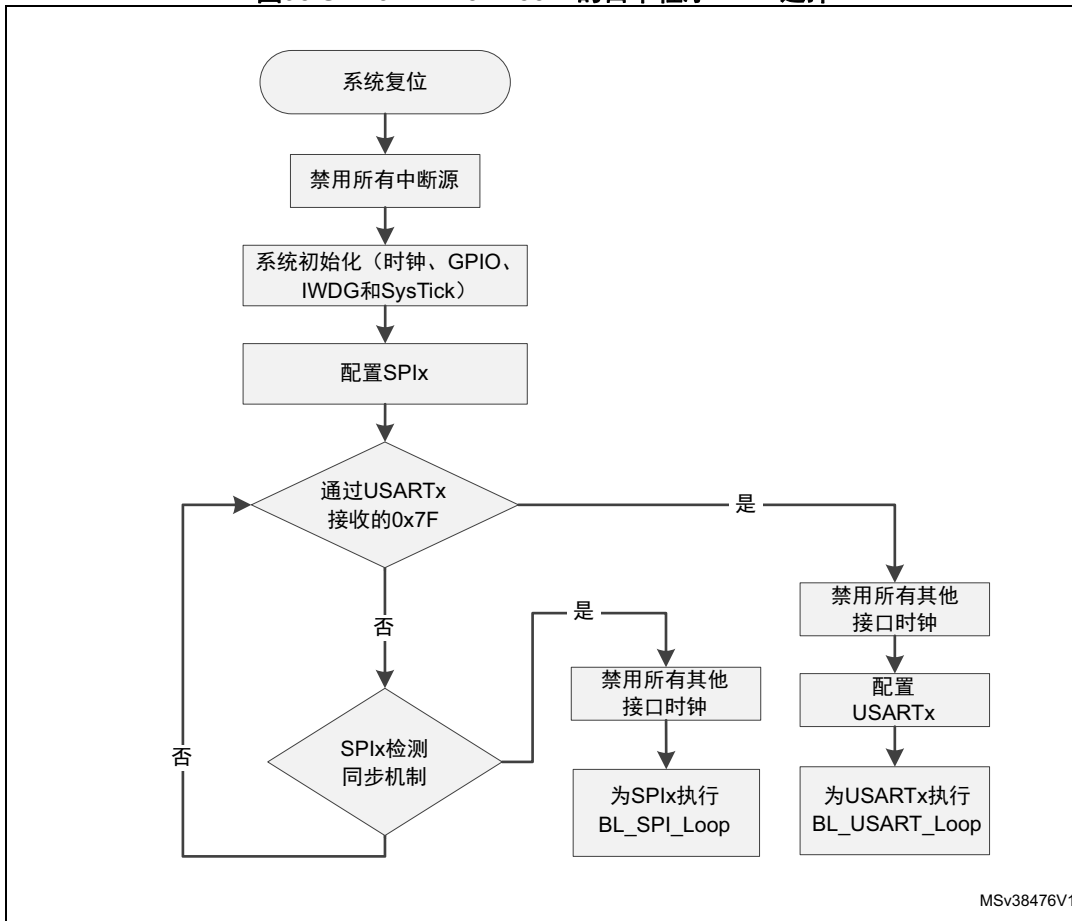
表154.系统存储器自举模式下STM32WLE5xx/55xx器件的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。 注： 如果SPI主设备不使用IO，则此IO可连接到GND。

71.2 自举程序选择

图 95显示了自举程序选择机制。

图95.STM32WLE5xx/55xx的自举程序V12.x选择



71.3 自举程序版本

表155.STM32WLE5xx/55xx自举程序版本

自举程序版本号	说明	已知限制
V12.2	修订Z样品上的 初始自举程序版本	无
V12.3	修订Z样品上的最终自举程序版本	无
V12.4	修订Y样品上的最终自举程序版本	无

72 STM32U575xx/85xx器件自举程序

72.1 自举程序配置

STM32U575xx/85xx自举程序通过应用模式12激活（详述参见表 2：自举程序激活模式）。下表介绍了该自举程序使用的硬件资源。

表156.系统存储器自举模式下STM32U575xx/85xx的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	RCC	HSI使能	系统时钟频率等于60 MHz（使用由HSI驱动PLL）。
	RAM	-	自地址0x20000000起的16 KB空间供自举程序固件使用
	系统存储器	-	自地址0x0BF90000起的64 KB空间包含自举程序固件
	IWDG	-	独立看门狗（IWDG）预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件IWDG选项）。
USART1自举程序	USART1	启用	初始化后，USART1的配置为：8位，偶校验位和1个停止位。
	USART1_RX引脚	输入	PA10引脚：USART1处于接收模式。用于复用推挽，上拉模式。
	USART1_TX引脚	输出	PA9引脚：USART1处于发送模式。用于复用推挽，上拉模式。
USART2自举程序	USART2	启用	初始化后，USART2的配置为：8位，偶校验位和1个停止位。
	USART2_RX引脚	输入	PA3引脚：USART2处于接收模式。用于复用推挽，上拉模式。
	USART2_TX引脚	输出	PA2引脚：USART2处于发送模式。用于复用推挽，上拉模式。
USART3自举程序	USART3	启用	初始化后，USART3的配置为：8位，偶校验位和1个停止位。
	USART3_RX引脚	输入	PC11引脚：USART3处于接收模式。用于复用推挽，上拉模式。
	USART3_TX引脚	输出	PC10引脚：USART3处于发送模式。用于复用推挽，上拉模式。
I2C1自举程序	I2C1	启用	I2C1配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011010x（其中x = 0对应写，x = 1对应读）
	I2C1_SCL引脚	输入/输出	PB6引脚：时钟线在开漏上拉模式下使用。
	I2C1_SDA引脚	输入/输出	PB7引脚：数据线在开漏上拉模式下使用。

表156.系统存储器自举模式下STM32U575xx/85xx的配置（续）

自举程序	功能/外设	状态	备注
I2C2自举程序	I2C2	启用	I2C2配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011010x（其中x = 0对应写，x = 1对应读）
	I2C2_SCL引脚	输入/输出	PB10引脚：时钟线在开漏上拉模式下使用。
	I2C2_SDA引脚	输入/输出	PB11引脚：数据线在开漏上拉模式下使用。
I2C3自举程序	I2C3	启用	I2C3配置为： I2C速度：高达1 MHz，7位地址，从模式，模拟滤波器开启。 7位从地址：0b1011010x（其中x = 0对应写，x = 1对应读）
	I2C3_SCL引脚	输入/输出	PC0引脚：时钟线在开漏上拉模式下使用。
	I2C3_SDA引脚	输入/输出	PC1引脚：数据线在开漏上拉模式下使用。
SPI1自举程序	SPI1	启用	SPI1配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI1_MOSI引脚	输入	PA7引脚：从数据输入线，用于推挽下拉模式
	SPI1_MISO引脚	输出	PA6引脚：从数据输出线，用于推挽下拉模式
	SPI1_SCK引脚	输入	PA5引脚：从时钟线，用于推挽下拉模式
	SPI1_NSS引脚	输入	PA4引脚：从芯片选择引脚，用于推挽下拉模式。

表156.系统存储器自举模式下STM32U575xx/85xx的配置（续）

自举程序	功能/外设	状态	备注
SPI2自举程序	SPI2	启用	SPI2配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI2_MOSI引脚	输入	PB15引脚：从数据输入线，用于推挽下拉模式
	SPI2_MISO引脚	输出	PB14引脚：从数据输出线，用于推挽下拉模式
	SPI2_SCK引脚	输入	PB13引脚：从时钟线，用于推挽下拉模式
	SPI2_NSS引脚	输入	PB12引脚：从芯片选择引脚，用于推挽下拉模式。
SPI3自举程序	SPI3	启用	SPI3配置为： - 从模式 - 全双工 - 8位MSB - 速度高达8 MHz - 极性：CPOL低、CPHA低、NSS硬件。
	SPI3_MOSI引脚	输入	PB5引脚：从数据输入线，用于推挽下拉模式
	SPI3_MISO引脚	输出	PG10引脚：从数据输入线，用于推挽下拉模式
	SPI3_SCK引脚	输入	PG9引脚：从数据输出线，用于推挽下拉模式
	SPI3_NSS引脚	输入	PG12引脚：从芯片选择引脚，用于推挽下拉模式。
FDCAN自举程序	FDCAN1	启用	初始化后，FDCAN1的配置为： 比特率为0.5 Mbps FrameFormat = FDCAN_FRAME_FD_BRS 模式 = FDCAN_MODE_NORMAL AutoRetransmission = 启用 TransmitPause = 禁用 ProtocolException = 启用
	FDCAN1_Rx引脚	输入/	PB9引脚：FDCAN1处于接收模式。用于复用推挽，上拉模式。
	FDCAN1_Tx引脚	输出	PB8引脚：FDCAN1处于发送模式。用于复用推挽，上拉模式。

表156.系统存储器自举模式下STM32U575xx/85xx的配置（续）

自举程序	功能/外设	状态	备注
DFU自举程序	USB:	启用	USB FS配置为强制器件模式。 使能USB FS中断向量以用于USB DFU通信。 注意: 由于USB外设被自举程序使用, VDDUSB IO必须连接到3.3 V电源。
	USB_DM引脚	输入/输出	PA11: USB DM线。用于输入无上拉/下拉模式。
	USB_DP引脚		PA12: USB DP线。用于输入无上拉/下拉模式。 无需外部上拉电阻

表157.STM32U575xx/585xx特殊命令

支持的特殊命令（USART/I2C/SPI/FDCAN） 操作码 - 0x50							
功能	子操作码 (2字节)	发送的数据 量 (2字节)	发送的数据	接收的数据 量	接收的数据	接收的状态 数据量 (2字节)	接收的状态数据
Trust Zone禁用 当TZEN=1且RDP = 1时必须运行	0x82	0x4	0x0	0x0	NA	0x1	0x0
从RDP L1降级为 RDP 0.5 当TZEN=1且RDP = 1时必须运行	0x82	0x4	0x1	0x0	NA	0x1	0x0
解锁写保护 当RDP = 1时必须 运行	0x82	0x4	0xYY02 ⁽¹⁾	0x0	NA	0x1	0x0

1. 0xYY可以有4个值（0: WRP区；1: WRP1A；2: WRP2A；3: WRP1B；4: WRP2B）

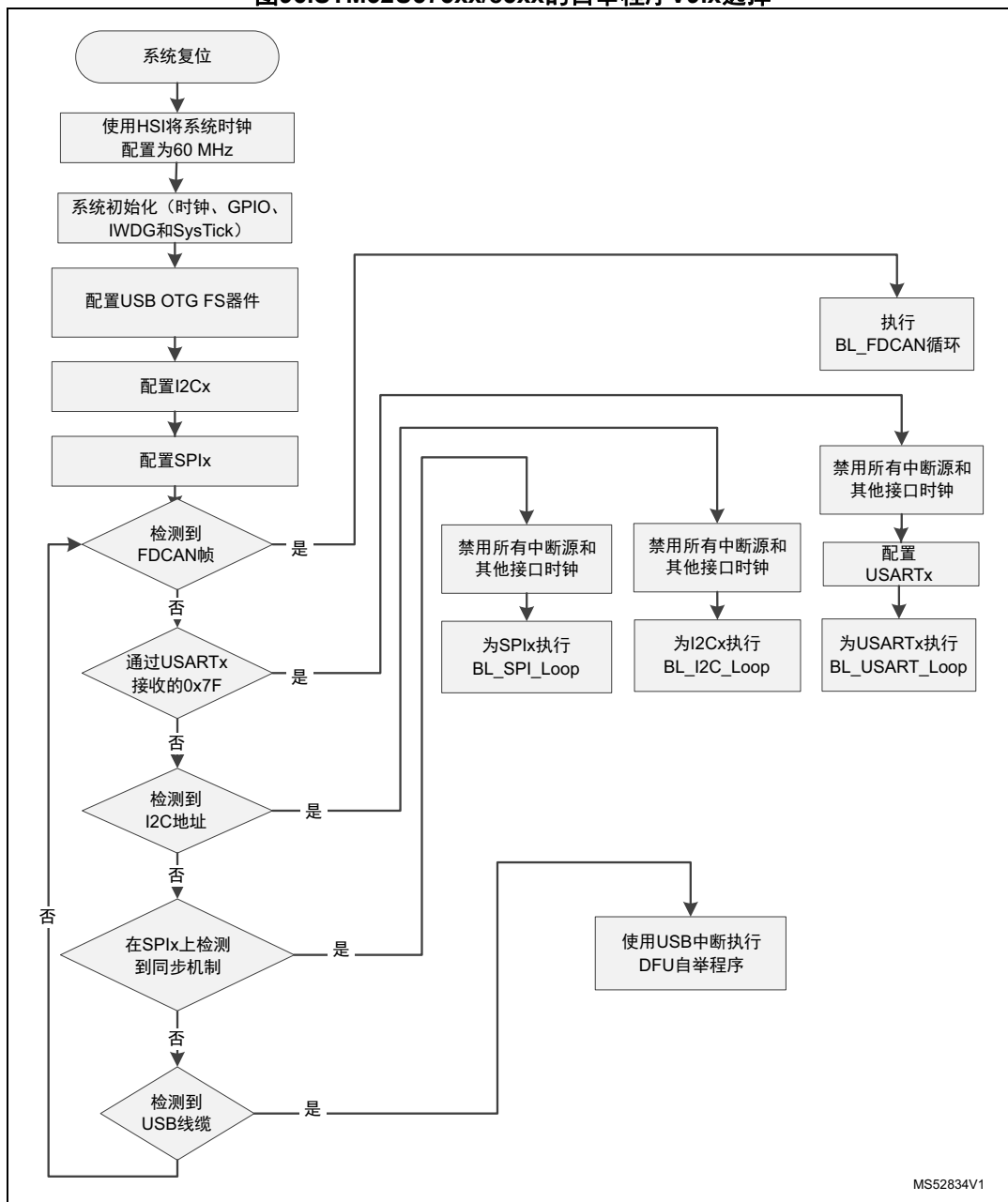
注: 根据USB协议规范，USB特殊命令与其他协议相比稍有不同：

- 不使用操作码，直接使用子操作码
- 在一个字节而非两个字节中处理子操作码
- 通过USB帧按字节发送数据。无需增加要发送的数据量
- 基于USB原生协议格式化返回的数据和状态

72.2 自举程序选择

图 96显示了自举程序选择机制。

图96.STM32U575xx/85xx的自举程序V9.x选择



MS52834V1

72.3 自举程序版本

表158.STM32U575xx/85xx自举程序版本

自举程序版本号	说明	已知限制
V9.2	初始自举程序版本	无

73 器件相关的自举程序参数

对于所有STM32器件，每种串行外设的自举程序协议命令集和序列都相同。但是，一些参数取决于器件和自举程序版本：

- PID（产品ID）
- 当请求ReadMemory、Go和WriteMemory命令时，自举程序接收有效RAM存储地址（自举程序执行过程中所用RAM区域不可访问）。
- 系统存储区。

表 159显示了每种STM32器件的这些参数值。

表159. 器件相关的自举程序参数

STM32系列	设备	PID	BL ID	RAM	系统存储器
C0	STM32C011xx	0x443	0x51	0x20000000 - 0x20002FFF	0x1FFF0000 - 0x1FFF17FF
	STM32C031xx	0x453	0x52	0x20002000 - 0x200017FF	
F0	STM32F05xxx和STM32F030x8	0x440	0x21	0x20000800 - 0x20001FFF	0x1FFFE000 - 0x1FFFF7FF
	STM32F03xx4/6	0x444	0x10	0x20000800 - 0x20000FFF	
	STM32F030xC	0x442	0x52	0x20001800 - 0x20007FFF	0x1FFFD800 - 0x1FFFF7FF
	STM32F04xxx	0x445	0xA1	NA	0x1FFFC400 - 0x1FFFF7FF
	STM32F070x6	0x445	0xA2	NA	0x1FFFC400 - 0x1FFFF7FF
	STM32F070xB	0x448	0xA2	NA	0x1FFFC800 - 0x1FFFF7FF
	STM32F071xx/072xx	0x448	0xA1	0x20001800 - 0x20003FFF	0x1FFFC800 - 0x1FFFF7FF
	STM32F09xxx	0x442	0x50	NA	0x1FFFD800 - 0x1FFFF7FF

表159. 器件相关的自举程序参数 (续)

STM32系列	设备		PID	BL ID	RAM	系统存储器
F1	STM32F10xxx	小容量	0x412	NA	0x20000200 - 0x200027FF	0x1FFFF000 - 0x1FFFF7FF
		中容量	0x410	NA	0x20000200 - 0x20004FFF	
		大容量	0x414	NA	0x20000200 - 0x2000FFFF	
		中容量超值型	0x420	0x10	0x20000200 - 0x20001FFF	
		大容量超值型	0x428	0x10	0x20000200 - 0x20007FFF	
	STM32F105xx/107xx	0x418	NA	0x20001000 - 0x2000FFFF	0x1FFFB000 - 0x1FFFF7FF	
	STM32F10xxx超大容量	0x430	0x21	0x20000800 - 0x20017FFF	0x1FFFE000 - 0x1FFFF7FF	
F2	STM32F2xxxx		0x411	0x20	0x20002000 - 0x2001FFFF	0x1FFF0000 - 0x1FFF77FF
				0x33		
F3	STM32F373xx	0x432	0x41	0x41	0x20001400 - 0x20007FFF	0x1FFFD800 - 0x1FFFF7FF
	STM32F378xx			0x50	0x20001000 - 0x20007FFF	
	STM32F302xB (C) /303xB (C)	0x422	0x41	0x41	0x20001400 - 0x20009FFF	
	STM32F358xx			0x50		
	STM32F301xx/302x4 (6/8)	0x439	0x40	0x40	0x20001800 - 0x20003FFF	
	STM32F318xx			0x50		
	STM32F303x4 (6/8) / 334xx/328xx	0x438	0x50	0x20001800 - 0x20002FFF		
	STM32F302xD (E) /303xD (E)	0x446	0x40	0x20001800 - 0x2000FFFF		
	STM32F398xx	0x446	0x50	0x20001800 - 0x2000FFFF		

表159.器件相关的自举程序参数（续）

STM32系列	设备	PID	BL ID	RAM	系统存储器
F4	STM32F40xxx/41xxx	0x413	0x31	0x20002000 - 0x2001FFFF	0x1FFF0000 - 0x1FFF77FF
			0x91	0x20003000 - 0x2001FFFF	
	STM32F42xxx/43xxx	0x419	0x70	0x20003000 - 0x2002FFFF	
			0x91	0x20003000 - 0x2002FFFF	
	STM32F401xB (C)	0x423	0xD1	0x20003000 - 0x2000FFFF	
	STM32F401xD (E)	0x433	0xD1	0x20003000 - 0x20017FFF	
	STM32F410xx	0x458	0xB1	0x20003000 - 0x20007FFF	
	STM32F411xx	0x431	0xD0	0x20003000 - 0x2001FFFF	
	STM32F412xx	0x441	0x90	0x20003000 - 0x2003FFFF	
	STM32F446xx	0x421	0x90	0x20003000 - 0x2001FFFF	
	STM32F469xx/479xx	0x434	0x90	0x20003000 - 0x2005FFFF	
STM32F413xx/423xx	0x463	0x90	0x20003000 - 0x2004FFFF		
F7	STM32F72xxx/73xxx	0x452	0x90	0x20004000 - 0x2003FFFF	0x1FF00000 - 0x1FF0EDBF
	STM32F74xxx/75xxx	0x449	0x70	0x20004000 - 0x2004FFFF	0x1FF00000 - 0x1FF0EDBF
			0x90	0x20004000 - 0x2004FFFF	0x1FF00000 - 0x1FF0EDBF
	STM32F76xxx/77xxx	0x451	0x93	0x20004000 - 0x2007FFFF	0x1FF00000 - 0x1FF0EDBF

表159. 器件相关的自举程序参数 (续)

STM32系列	设备	PID	BL ID	RAM	系统存储器
G0	STM32G03xxx/04xxx	0x466	0x52	0x20001000 - 0x20001FFF	0x1FFF0000 - 0x1FFF1FFF
	STM32G07xxx/08xxx	0x460	0xB2	0x20002700 - 0x20009000	0x1FFF0000 - 0x1FFF6FFF
	STM32G0B0xx	0x467	0xD0	0x20004000 - 0x20020000	0x1FFF0000 - 0x1FFF6FFF 0x1FFF8000 - 0x1FFFEFFF
	STM32G0B1xx/0C1xx	0x467	0x92	0x20004000 - 0x20020000	0x1FFF0000 - 0x1FFF6FFF 0x1FFF8000 - 0x1FFFEFFF
	STM32G05xxx/061xx	0x456	0x51	0x20001000 - 0x20002000	0x1FFF0000 - 0x1FFF1FFF
G4	STM32G431xx/441xx	0x468	0xD4	0x20004000 - 0x20005800	0x1FFF0000 - 0x1FFF7000
	STM32G47xxx/48xxx	0x469	0xD5	0x20004000 - 0x20018000	0x1FFF0000 - 0x1FFF7000
	STM32G491xx/A1xx	0x479	0xD2	0x20004000 - 0x2001C000	0x1FFF0000 - 0x1FFF7000
H7	STM32H72xxx/73xxx	0x483	0x93	0x20004100 - 0x2001FFFF 0x24004000 - 0x2404FFFF	0x1FF00000 - 0x1FF1E7FF
	STM32H74xxx/75xxx	0x450	0x90	0x20004100 - 0x2001FFFF 0x24034000 - 0x2407FFFF	0x1FF00000 - 0x1FF1E7FF
	STM32H7A3xx/B3xx	0x480	0x90	0x20004100 - 0x2001FFFF 0x24034000 - 0x2407FFFF	0x1FF00000 - 0x1FF13FFF
L0	STM32L01xxx/02xxx	0x457	0xC3	NA	0x1FF00000 - 0x1FF00FFF
	STM32L031xx/041xx	0x425	0xC0	0x20001000 - 0x20001FFF	0x1FF00000 - 0x1FF00FFF
	STM32L05xxx/06xxx	0x417	0xC0	0x20001000 - 0x20001FFF	0x1FF00000 - 0x1FF00FFF
	STM32L07xxx/08xxx	0x447	0x41	0x20001000 - 0x20004FFF	0x1FF00000 - 0x1FF01FFF
0xB2			0x20001400 - 0x20004FFF		

表159. 器件相关的自举程序参数 (续)

STM32系列	设备	PID	BL ID	RAM	系统存储器
L1	STM32L1xxx6 (8/B)	0x416	0x20	0x20000800 - 0x20003FFF	0x1FFF0000 - 0x1FFF01FFF
	STM32L1xxx6 (8/B) A	0x429	0x20	0x20001000 - 0x20007FFF	
	STM32L1xxxC	0x427	0x40		
	STM32L1xxxD	0x436	0x45	0x20001000 - 0x2000BFFF	
	STM32L1xxxE	0x437	0x40	0x20001000 - 0x20013FFF	
L4	STM32L412xx/422xx	0x464	0xD1	0x20002100 - 0x20008000	0x1FFF0000 - 0x1FFF6FFF
	STM32L43xxx/44xxx	0x435	0x91	0x20003100 - 0x2000BFFF	0x1FFF0000 - 0x1FFF6FFF
	STM32L45xxx/46xxx	0x462	0x92	0x20003100 - 0x2001FFFF	0x1FFF0000 - 0x1FFF6FFF
	STM32L47xxx/48xxx	0x415	0xA3	0x20003000 - 0x20017FFF	0x1FFF0000 - 0x1FFF6FFF
			0x92	0x20003100 - 0x20017FFF	
	STM32L496xx/4A6xx	0x461	0x93	0x20003100 - 0x2003FFFF	0x1FFF0000 - 0x1FFF6FFF
	STM32L4Rxx/4Sxx	0x470	0x95	0x20003200 - 0x2009FFFF	0x1FFF0000 - 0x1FFF6FFF
STM32L4P5xx/Q5xx	0x471	0x90	0x20004000 - 0x2004FFFF	0x1FFF0000 - 0x1FFF6FFF	
L5	STM32L552xx/562xx	0x472	0x92	0x20004000 - 0x2003FFFF	0x0BF90000 - 0x0BF97FFF
WB	STM32WB10xx/15xx	0x494	0xB1	0x20005000 - 0x20040000	0x1FFF0000 - 0x1FFF7000
	STM32WB30xx/35xx/50xx/WB55xx	0x495	0xD5	0x20004000 - 0x2000BFFF	0x1FFF0000 - 0x1FFF7000
WL	STM32WLE5xx/WL55xx	0x497	0xC4	0x20002000 - 0x2000FFFF	0x1FFF0000 - 0x1FFF3FFF
U5	STM32U575xx/ STM32U585xx	0x482	0x92	0x20004000 - 0x200BFFFF	0x0BF90000 - 0x0BF9FFFF

74 自举程序时序

本章给出了自举程序固件的典型时序，用来确保主机和设备之间的正确同步。

描述了两种时序：

- STM32器件自举程序资源初始化时间。
- 通信接口选择时间。

这些时序之后，自举程序准备好接收和执行主机命令。

74.1 自举程序启动时序

自举程序复位后，主机必须等待，直至STM32自举程序准备好利用专门接口通信启动检测阶段。这个时间对应于自举程序启动时序，期间初始化自举程序所用资源。

图97.自举程序启动时序描述

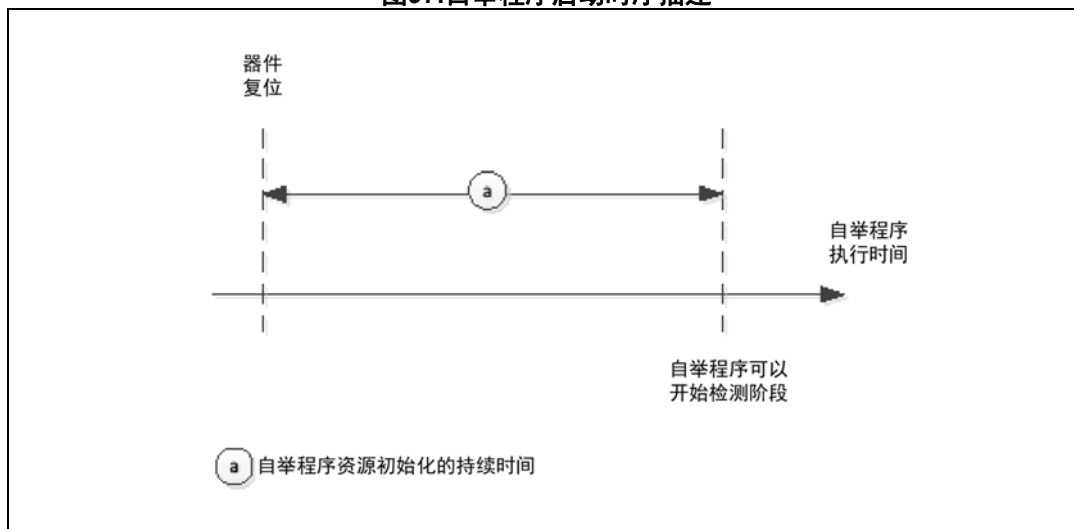


表160.STM32器件的自举程序启动时序 (ms)

设备	最小自举程序启动	HSE超时
STM32F03xx4/6	1.612	NA
STM32F05xxx和STM32F030x8器件	1.612	NA
STM32F04xxx	0.058	NA
STM32F071xx/072xx	0.058	NA
STM32F070x6	HSE已连接	3
	HSE未连接	230
STM32F070xB	HSE已连接	6
	HSE未连接	230
STM32F09xxx	2	NA
STM32F030xC	2	NA

表160.STM32器件的自举程序启动时序 (ms) (续)

设备		最小自举程序启动	HSE超时
STM32F10xxx		1.227	NA
STM32F105xx/107xx	PA9引脚低电平	1.396	NA
	PA9引脚高电平	524.376	
STM32F10xxx超大容量		1.227	NA
STM32F2xxxx	V2.x	134	NA
	V3.x	84.59	0.790
STM32F301xx/302x4 (6/8)	HSE已连接	45	560.5
	HSE未连接	560.8	
STM32F302xB (C) /303xB (C)	HSE已连接	43.4	2.236
	HSE未连接	2.36	
STM32F302xD (E) /303xD	HSE已连接	7.53	NA
	HSE未连接	146.71	NA
STM32F303x4 (6/8) /334xx/328xx		0.155	NA
STM32F318xx		0.182	NA
STM32F358xx		1.542	NA
STM32F373xx	HSE已连接	43.4	2.236
	HSE未连接	2.36	
STM32F378xx		1.542	NA
STM32F398xx		1.72	NA
STM32F40xxx/41xxx	V3.x	84.59	0.790
	V9.x	74	96
STM32F401xB (C)		74.5	85
STM32F401xD (E)		74.5	85
STM32F410xx		0.614	NA
STM32F411xx		74.5	85
STM32F412xx		0.614	180
STM32F413xx/423xx		0.642	165
STM32F429xx/439xx	V7.x	82	97
	V9.x	74	97
STM32F446xx		73.61	96
STM32F469xx/479xx		73.68	230
STM32F72xxx/73xxx		17.93	50
STM32F74xxx/75xxx		16.63	50
STM32G03xxx/04xxx		0.390	NA
STM32G07xxx/08xxx		0.390	NA

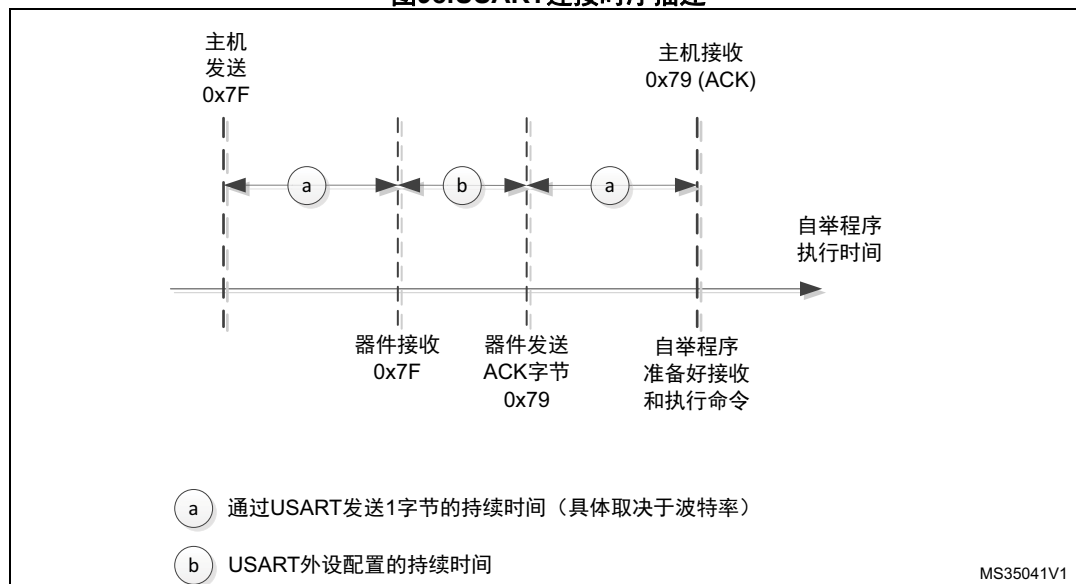
表160.STM32器件的自举程序启动时序 (ms) (续)

设备		最小自举程序启动	HSE超时
STM32G0Bxxx/Cxxx		0.390	NA
STM32G05xxx/061xx		0.390	NA
STM32G4xxxx		0.390	NA
STM32H72xxx/73xxx		53.975	NA
STM32H74xxx/75xxx		53.975	2
STM32H7A3xx/B3xx		53.975	NA
STM32L01xxx/02xxx		0.63	NA
STM32L031xx/041xx		0.62	NA
STM32L05xxx/06xxx		0.22	NA
STM32L07xxx/08xxx	V4.x	0.61	NA
	V11.x	0.71	NA
STM32L1xxx6 (8/B) A		0.542	NA
STM32L1xxx6 (8/B)		0.542	NA
STM32L1xxxC		0.708	80
STM32L1xxxD		0.708	80
STM32L1xxxE		0.708	200
STM32L43xxx/44xxx		0.86	100
STM32L45xxx/46xxx		0.86	NA
STM32L47xxx/48xxx	V10.x	LSE已连接	55
		LSE未连接	2560
	V9.x	LSE已连接	55.40
		LSE未连接	2560.51
STM32L412xx/422xx		0.86	NA
STM32L496xx/4A6xx		76.93	100
STM32L4P5xx /Q5xx		NA	NA
STM32L4Rxx/4Sxx		NA	NA
STM32L552xx/562xx		0.390	NA
STM32WB10xx/15xx/30xx/35xx/50xx/55xx		0.390	NA
STM32WLE5xx/WL55xx		0.390	NA
STM32U575xx/85xx		0.390	NA

74.2 USART连接时序

USART连接时序是主机在发送同步数据（0x7F）和接收第一个确认响应（0x79）之间所需等待的时间。

图98.USART连接时序描述



1. 接收到任何不同于0x7F（或线路故障）的其他字符会导致自举程序以错误的波特率开始通信。自举程序测量0x7F中第一个比特的上升沿到0x7F中最后一个比特的下降沿之间的信号长度，来推导出波特率值
2. 自举程序不会将所计算的波特率重新调整为标准波特率值（即1200、9600、115200.....）。

注：对于STM32F105xx/107xx系列器件，PA9引脚（USB_VBUS）用于检测USB主机连接。仅当PA9引脚在检测阶段具有高电平（即主机与该端口相连并在USB总线上提供5 V电压）时，才会对USB外设执行初始化。此时初始化和关闭USB外设需要耗用更长的时间。要在未使用PA9引脚时将自举程序检测时间缩至最短，可在USART检测阶段器件复位之后，使PA9引脚保持低电平，直到发送器件ACK信号。

表161.STM32器件的USART自举程序最小时序（ms）

设备	一个USART发送一个USART字节	USART配置	USART连接
STM32F03xx4/6	0.078125	0.0064	0.16265
STM32F05xxx和STM32F030x8器件	0.078125	0.0095	0.16575
STM32F04xxx	0.078125	0.007	0.16325
STM32F071xx/072xx	0.078125	0.007	0.16325
STM32F070x6	0.078125	0.014	0.17
STM32F070xB	0.078125	0.08	0.23
STM32F09xxx	0.078125	0.07	0.22
STM32F030xC	0.078125	0.07	0.22
STM32F10xxx	0.078125	0.002	0.15825
STM32F105xx/107xx	PA9引脚低电平	0.078125	0.16325
	PA9引脚高电平	105	105.15625

表161.STM32器件的USART自举程序最小时序 (ms) (续)

设备	一个USART 发送一个USART字节	USART配置	USART连接	
STM32F10xxx超大容量	0.078125	0.006	0.16225	
STM32F2xxxx	V2.x	0.078125	0.009	0.16525
	V3.x			
STM32F301xx/302x4 (6/8)	HSE已连接	0.078125	0.002	0.15825
	HSE未连接			
STM32F302xB (C) /303xB (C)	HSE已连接	0.078125	0.002	0.15825
	HSE未连接			
STM32F302xD (E) /303xD	0.078125	0.002	0.15885	
STM32F303x4 (6/8) /334xx/328xx	0.078125	0.002	0.15825	
STM32F318xx	0.078125	0.002	0.15825	
STM32F358xx	0.15625	0.001	0.3135	
STM32F373xx	HSE已连接	0.078125	0.002	0.15825
	HSE未连接			
STM32F378xx	0.15625	0.001	0.3135	
STM32F398xx	0.078125	0.002	0.15885	
STM32F40xxx/41xxx	V3.x	0.078125	0.009	0.16525
	V9.x		0.0035	0.15975
STM32F401xB (C)	0.078125	0.00326	0.15951	
STM32F401xD (E)	0.078125	0.00326	0.15951	
STM32F410xx	0.078125	0.002	0.158	
STM32F411xx	0.078125	0.00326	0.15951	
STM32F412xx	0.078125	0.002	0.158	
STM32F413xx/423xx	0.078125	0.002	0.158	
STM32F429xx/439xx	V7.x	0.078125	0.007	0.16325
	V9.x		0.00326	0.15951
STM32F446xx	0.078125	0.004	0.16	
STM32F469xx/479xx	0.078125	0.003	0.159	
STM32F72xxx/73xxx	0.078125	0.070	0.22	
STM32F74xxx/75xxx	0.078125	0.065	0.22	
STM32G03xxx/04xxx	0.078125	0.01	0.11	
STM32G07xxx/08xxx	0.078125	0.01	0.11	
STM32G0Bxxx/Cxxx	0.078125	0.01	0.11	
STM32G05xxx/061xx	0.078125	0.01	0.11	
STM32G4xxxx	0.078125	0.003	0.159	

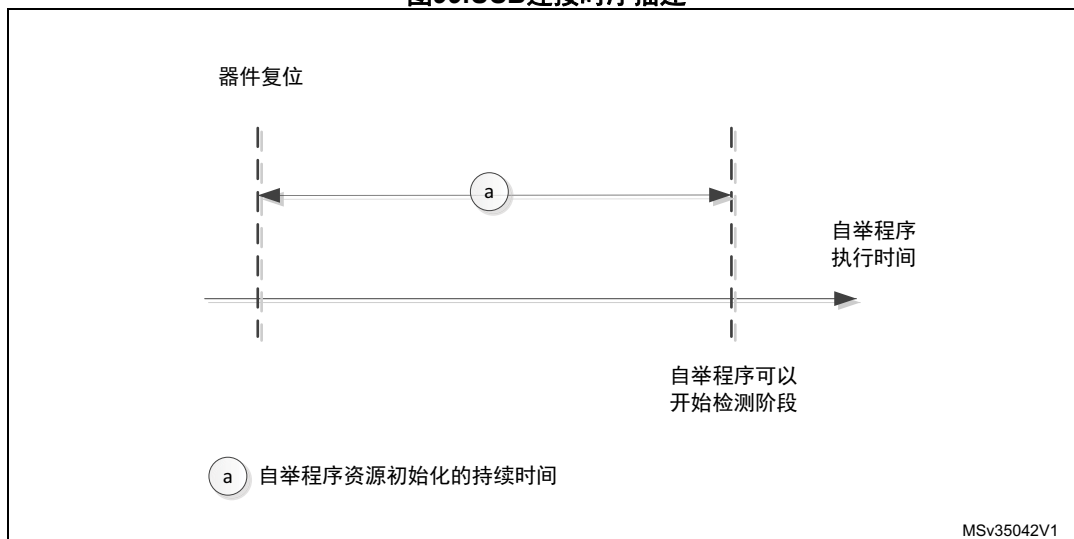
表161.STM32器件的USART自举程序最小时序 (ms) (续)

设备	一个USART 发送一个USART字节	USART配置	USART连接
STM32H72xxx/73xxx	0.078125	0.072	0.22825
STM32H74xxx/75xxx	0.078125	0.072	0.22825
STM32H7A3xx/B3xx	0.078125	0.072	0.22825
STM32L01xxx/02xxx	0.078125	0.016	0.17
STM32L031xx/041xx	0.078125	0.018	0.174
STM32L05xxx/06xxx	0.078125	0.018	0.17425
STM32L07xxx/08xxx	V4.x	0.078125	0.173
	V11.x	0.078125	0.158
STM32L1xxx6 (8/B) A	0.078125	0.008	0.16425
STM32L1xxx6 (8/B)	0.078125	0.008	0.16425
STM32L1xxxC	0.078125	0.008	0.16425
STM32L1xxxD	0.078125	0.008	0.16425
STM32L1xxxE	0.078125	0.008	0.16425
STM32L412xx/422xx	0.078125	0.005	0.2
STM32L43xxx/44xxx	0.078125	0.003	0.159
STM32L45xxx/46xxx	0.078125	0.07	0.22
STM32L47xxx/48xxx	V10.x	0.078125	0.159
	V9.x	0.078125	0.159
STM32L496xx/4A6xx	0.078125	0.003	0.159
STM32L4Rxx/4Sxx	NA	NA	NA
STM32L4P5xx/4Q5xx	NA	NA	NA
STM32L552xx/562xx	0.078125	0.01	0.11
STM32WB10xx/15xx/30xx/35xx/50xx/55xx	0.078125	0.003	0.159
STM32WLE5xx/WL55xx	0.078125	0.001	0.110
STM32U575xx/85xx	0.078125	0.001	NA

74.3 USB连接时序

USB连接时序是插入USB线缆到与设备建立正确连接之间主机所需等待的时间。该时序包括了枚举和DFU元件配置。USB连接取决于主机。

图99.USB连接时序描述



注:

对于STM32F105xx/107xx器件，如果外部HSE晶振频率不等于25 MHz（14.7456 MHz或8 MHz），则器件在与主机建立正确连接之前，将执行多个失败的枚举（包含连接/断开连接序列）。这是由基于帧起始（SOF）检测的HSE自动检测机制引起的。

表162.STM32器件的USB自举程序最小时序（ms）

设备		USB连接
STM32F04xxx		350
STM32F070x6		待定
STM32F070xB		320
STM32F105xx/107xx	HSE = 25 MHz	460
	HSE = 14.7465 MHz	4500
	HSE = 8 MHz	13700
STM32F2xxxx		270
STM32F301xx/302x4 (6/8)		300
STM32F302xB (C) /303xB (C)		300
STM32F302xD (E) /303xD		100
STM32F373xx		300
STM32F40xxx/41xxx	V3.x	270
	V9.x	250
STM32F401xB (C)		250
STM32F401xD (E)		250
STM32F411xx		250

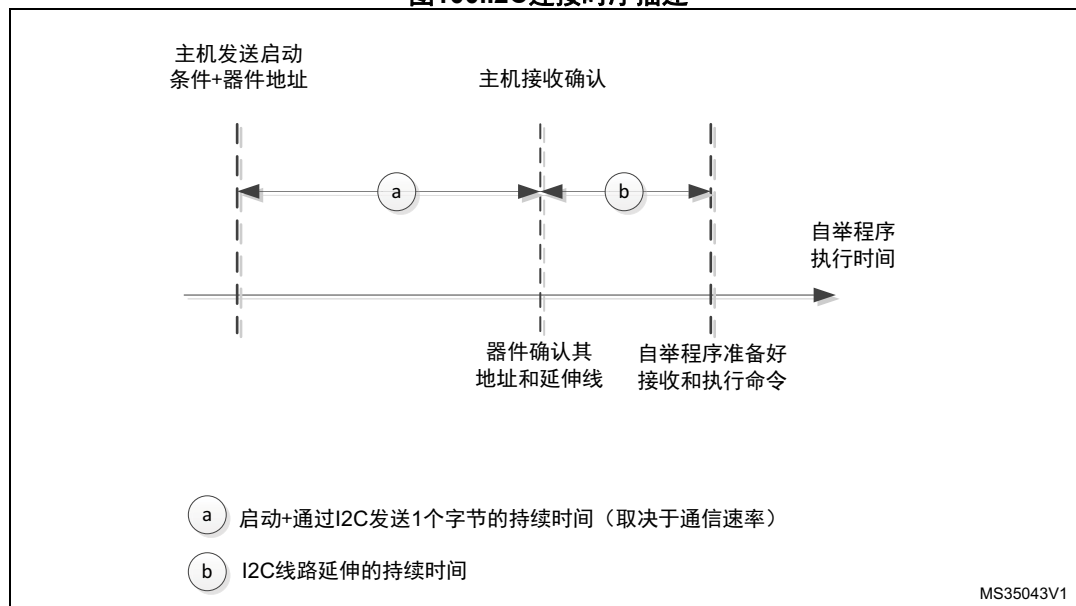
表162.STM32器件的USB自举程序最小时序 (ms) (续)

设备		USB连接
STM32F412xx		380
STM32F413xx/423xx		350
STM32F429xx/439xx	V7.x	250
	V9.x	
STM32F446xx		200
STM32F469xx/479xx		270
STM32F72xxx/73xxx		320
STM32F74xxx/75xxx		230
STM32G0B1xx/C1xx		300
STM32G4xxxx		300
STM32H72xxx/73xxx		53.9764
STM32H74xxx/75xxx		53.9764
STM32H7A3xx/B3xx		53.9764
STM32L07xxx/08xxx		140
STM32L1xxxC		849
STM32L1xxxD		849
STM32L412xx/422xx		820
STM32L43xxx/44xxx		820
STM32L45xxx/46xxx		330
STM32L47xxx/48xxx	V10.x	300
	V9.x	
STM32L496xx/4A6xx		430
STM32L4P5xx/4Q5xx		NA
STM32L4Rxx/4Sxx		NA
STM32L552xx/L562xx		300
STM32WB30xx/35xx/50xx/55xx		300
STM32U575xx/85xx		300

74.4 I2C连接时序

I2C连接时序是发送I2C设备地址到发送命令代码之间主机所需等待的时间。该时序包括了I2C线路延伸持续时间。

图100.I2C连接时序描述



注：I2C通信中采用了超时机制，只有遵守超时机制才能正常执行自举程序命令。此超时在同一命令中的两个I2C帧之间实施（例如，对于Write memory命令，将在命令发送帧与地址存储发送帧之间插入超时）。此外也将在同一I2C帧中的两个连续数据接收或发送操作之间插入同一超时周期。如果超时周期已过，则生成系统复位以避免自举程序崩溃。

在Erase Memory命令和Readout Unprotect命令中，在主机端实现时必须考虑操作持续时间。发送要擦除的页的代码后，主机须等待，直到自举程序器件执行页擦除并完成擦除命令的剩余步骤。

表163.STM32器件的I2C自举程序最小时序（ms）

设备	启动条件 + 发送一个I2C字节	I2C线路延伸	I2C连接	I2C超时
STM32F04xxx	0.0225	0.0025	0.0250	1000
STM32F070x6	0.0225	0.0025	0.0245	1000
STM32F070xB	0.0225	0.0025	0.0245	1000
STM32F071xx/072xx	0.0225	0.0025	0.0250	1000
STM32F09xxx	0.0225	0.0025	0.0245	1000
STM32F030xC	0.0225	0.0025	0.0250	1000
STM32F303x4 (6/8) /334xx/328xx	0.0225	0.0027	0.0252	1000
STM32F318xx	0.0225	0.0027	0.0252	1000
STM32F358xx	0.0225	0.0055	0.0280	10
STM32F378xx	0.0225	0.0055	0.0280	10

表163.STM32器件的I2C自举程序最小时序 (ms) (续)

设备	启动条件 + 发送一个I2C字节	I2C线路延伸	I2C连接	I2C超时	
STM32F398xx	0.0225	0.0020	0.0245	1500	
STM32F40xxx/41xxx	0.0225	0.0022	0.0247	1000	
STM32F401xB (C)	0.0225	0.0022	0.0247	1000	
STM32F401xD (E)	0.0225	0.0022	0.0247	1000	
STM32F410xx	0.0225	0.0020	0.0245	1000	
STM32F411xx	0.0225	0.0022	0.0247	1000	
STM32F412xx	0.0225	0.0020	0.0245	1000	
STM32F413xx/423xx	0.0225	0.0020	0.0245	1000	
STM32F42xxx/43xxx	V7.x	0.0225	0.0033	0.0258	1000
	V9.x	0.0225	0.0022	0.0247	1000
STM32F446xx	0.0225	0.0020	0.0245	1000	
STM32F469xx/479xx	0.0225	0.0020	0.0245	1000	
STM32F72xxx/73xxx	0.0225	0.0020	0.0245	1000	
STM32F74xxx/75xxx	0.0225	0.0020	0.0245	500	
STM32G03xxx/04xxx	0.0225	0.0020	0.0245	1000	
STM32G07xxx/08xxx	0.0225	0.0020	0.0245	1000	
STM32G0Bxx/Cxx	0.0225	0.0020	0.0245	1000	
STM32G05xxx/061xx	0.0225	0.0020	0.0245	1000	
STM32G4xxxx	0.0225	0.0020	0.0245	1000	
STM32H72xxx/73xxx	0.0225	0.05	0.0745	1000	
STM32H74xxx/75xxx	0.0225	0.05	0.0725	1000	
STM32H7A3xx/7B3xx	0.0225	0.05	0.0745	1000	
STM32L07xxx/08xxx	0.0225	0.0020	0.0245	1000	
STM32L412xx/422xx	0.0225	0.0020	0.0245	1000	
STM32L43xxx/44xxx	0.0225	0.0020	0.0245	1000	
STM32L45xxx/46xxx	0.0225	0.0020	0.0245	1000	
STM32L47xxx/48xxx	V10.x	0.0225	0.0020	0.0245	1000
	V9.x	0.0225	0.0020	0.0245	1000
STM32L496xx/4A6xx	0.0225	0.0020	0.0245	1000	
STM32L4P5xx/4Q5xx	NA	NA	NA	NA	
STM32L4Rxx/4Sxx	NA	NA	NA	NA	
STM32L552xx/L562xx	0.0225	0.0020	0.0245	1000	
STM32WB10xx/15xx/30xx/35xx/50xx/55xx	0.0225	0.0020	0.0245	1000	
STM32U575xx/85xx	0.0225	0.0020	0.0245	1000	

74.5 SPI连接时序

SPI连接时序是主机在发送同步数据（0xA5）和接收第一个确认响应（0x79）之间所需等待的时间。

图101.SPI连接时序描述

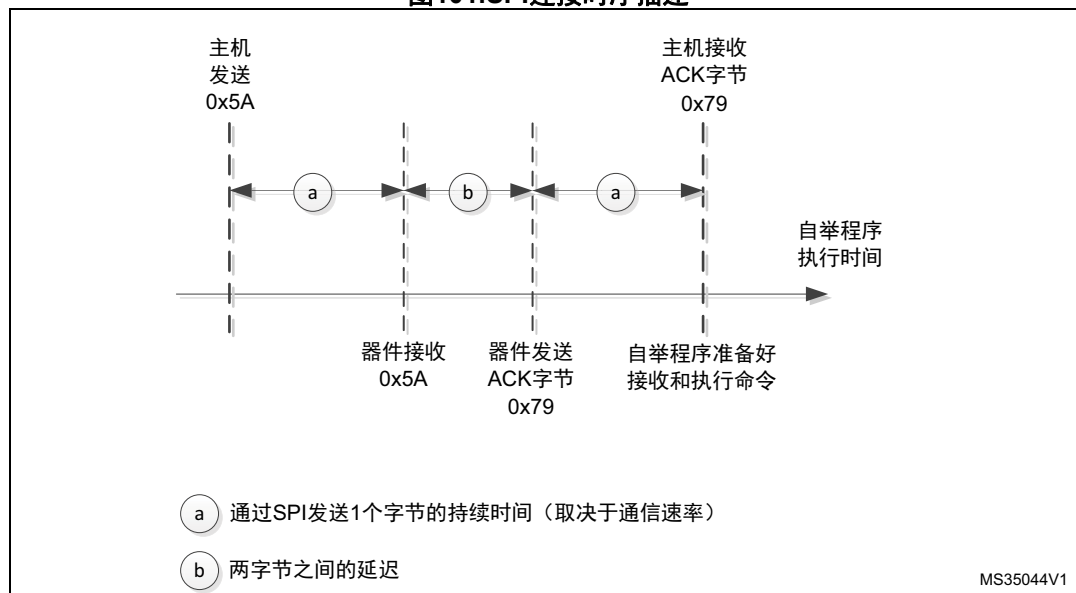


表164.STM32器件的SPI自举程序最小时序（ms）

设备	发送一个SPI字节	两字节之间的延迟	SPI连接
所有产品	0.001	0.008	0.01

附录A 使用“ExitSecureMemory”函数的函数示例

```
/**
*****
* @file main.c
*****
*/

/* 包含 -----*/
#include "main.h"

/* 私有函数原型 -----*/
static void ConfigClock(void);

void JUMP_WITHOUT_PARAM(uint32_t jump_address);
void JUMP_WITH_PARAM(uint32_t jump_address, uint32_t magic, uint32_t
applicationVectorAddress);

/* 私有函数 -----*/

/**
* @brief 主程序
* @参数 无
* @返回值: 无
*/
int main(void)
{
    ConfigClock();

    uint32_t application_address          = 0x08000800;
    uint32_t exit_secure_memory_address  = 0x1FFF1E00;
    uint32_t magic_number                = 0x08192A3C;
    uint32_t exit_with_magic_number      = 0x0;

    if (exit_with_magic_number)
    {
        JUMP_WITH_PARAM(exit_secure_memory_address, magic_number, application_address);
    }
    else
    {
        JUMP_WITHOUT_PARAM(exit_secure_memory_address);
    }
}
```



```
/**
 * @brief ConfigClock
 * @参数 无
 * @返回值: 无
 */
static void ConfigClock(void)
{
    /* 将按照所需项目的模板开发 */
}

/**
 * @brief JUMP_WITHOUT_PARAM
 * @param jump_address
 * @返回值: 无
 */
void JUMP_WITHOUT_PARAM(uint32_t jump_address)
{
    asm ("LDR R1, [R0]"); // jump_address
    asm ("LDR R2, [R0,#4]");
    asm ("MOV SP, R1");
    asm ("BX R2");
}

/**
 * @brief JUMP_WITH_PARAM
 * @param jump_address, magic, applicationVectorAddress
 * @返回值: 无
 */
void JUMP_WITH_PARAM(uint32_t jump_address, uint32_t magic, uint32_t
applicationVectorAddress)
{
    asm ("MOV R3, R0"); // jump_address
    asm ("LDR R0, [R3]");
    asm ("MOV SP, R0");
    asm ("LDR R0, [R3,#4]");
    asm ("BX R0");
}

/***** (C) STMicroelectronics版权所有 *****/
```

75 版本历史

表165.文档版本历史

日期	版本	变更
2007年10月 22日	1	初始版本。
2008年1月 22日	2	<p>在产的所有STM32 (rev.B和rev.Z) 都含有本应用笔记中所述的自举程序。</p> <p>修改: 第3.1节: 自举程序激活和 第1.4节: 自举程序代码序列。</p> <p>增加: 第1.3节: 硬件要求, 第1.5节: 选择USART波特率, 第1.6节: 使用自举程序和章节;</p> <p>注2链接到 表3: 自举程序命令中的Get, Get Version & Read Protection Status和Get ID命令, 增加了 注3。</p> <p>从文档中删除“permanent”一词 (Permanent Write Unprotect/Readout Protect/Unprotect)。少量文本更改。</p> <p>将自举程序版本升级为2.0。</p>
2008年5月 26日	3	<p>少量文本更改。在 表: 系统时钟由嵌入式内部高速RC提供, 自举程序执行不需要外部石英时钟中增加了RAM和系统存储器。</p> <p>删除了第8页的 第1.6节: 使用自举程序。</p> <p>删除修改, 修改了 注3, 在第9页的 表3: 自举程序命令中增加了 注1。</p> <p>修改了第11页的 字节3。</p> <p>修改了第13页的 字节2。</p> <p>修改了第15页的 字节2, 字节3-4和 字节5, 修改了 注3。</p> <p>修改了第18页的 字节8。</p> <p>第18页的 第2.5节: Go命令增加了注释。</p> <p>修改了 第20页的图11: Go命令: 器件端。</p> <p>第21页的 第2.6节: Write Memory命令增加了注释。</p> <p>修改了第24页的 字节8。</p> <p>修改了 图14: Erase Memory命令: 主机端和 图15: Erase Memory命令: 器件端。</p> <p>修改了第26页的 字节3。</p> <p>第9页的 表3: 字节程序命令。</p> <p>修改了注释并在第27页 第2.8节: Write Protect命令中增加了注释。</p> <p>修改了 图16: Write Protect命令: 主机端, 图17: Write Protect命令: 器件端, 图19: Write Unprotect命令: 器件端, 图21: Readout Protect命令: 器件端和 图23: Readout Unprotect命令: 器件端。</p>
2009年1月 29日	4	<p>此应用笔记同样适用于STM32F102xx微控制器。</p> <p>将自举程序版本更新为V2.2 (参见 表4: 自举程序版本)。</p>

表165.文档版本历史（续）

日期	版本	变更
2009年11月19日	5	<p>在表：系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟中增加了IWDG。增加了注释。</p> <p>在整个文档中将BL更改为自举程序。</p> <p>在表：系统时钟由嵌入式内部高速RC提供，自举程序执行不需要外部石英时钟中修改了Go命名说明。</p> <p>第2.4节：Read Memory命令中修正了自举程序等待的字节数。</p> <p>修改了图10：Go命令：主机端下面的注释。</p> <p>删除了第2.5节：Go命令中的注释并增加了注释。</p> <p>指定了起始RAM地址，并在第2.6节：Write Memory命令中增加了注释。将Write Memory命令发送到选项字节区域时擦除了所有选项。</p> <p>修改了图11：Go命令：器件端。</p> <p>修改了图13：Write Memory命令：器件端。</p> <p>在第2.7节：Erase Memory命令中增加了注释并修改了主机发送的字节3和4。</p> <p>第2.8节：Write Protect命令中增加了注释。</p>
2010年3月9日	6	<p>重新整理应用笔记。增加了超值型和互连型器件自举程序（替换AN2662）。更改了前言。增加了词汇表。</p>
2010年4月20日	7	<p>相关文档：增加了超大容量器件的数据手册和编程手册。</p> <p>术语表：增加了超大容量器件。</p> <p>表3：增加了超大容量器件的相关信息。</p> <p>第4.1节：自举程序配置：更新了第一个句子。</p> <p>第5.1节：自举程序配置：更新了第一个句子。</p> <p>增加了第6节：STM32F10xxx超大容量器件自举程序。</p> <p>表65：增加了超大容量器件的相关信息。</p>
2010年10月8日	8	<p>在表3和表65中增加了大容量超值型器件的信息。</p>
2010年10月14日	9	<p>删除了淘汰器件的参考资料。</p>
2010年11月26日	10	<p>增加了超低功耗器件的信息。</p>
2011年4月13日	11	<p>增加了STM32F205/215xx和STM32F207/217xx器件的相关信息。</p> <p>增加了第32节：自举程序时序</p>
2011年6月6日	12	<p>更新了：</p> <ul style="list-style-type: none"> - 表12：TM32L1xxx6 (8/B) 自举程序版本 - 表17：系统存储器自举模式下STM32F2xxxx器件的配置 - 表18：STM32F2xxxx自举程序V2.x版本 - 表20：STM32F2xxxx自举程序V3.x版本
2011年11月28日	13	<p>增加了STM32F405/415xx和STM32F407/417xx自举程序以及STM32F105xx/107xx自举程序V2.1的相关信息。</p> <p>在第4节：STM32F10xxx器件自举程序标题和概述中增加了超值型器件。</p>

表165.文档版本历史（续）

日期	版本	变更
2012年7月 30日	14	<p>增加了从STM32F051x6/STM32F051x8器件到大容量超低功耗STM32L151xx、STM32L152xx器件的自举程序的相关信息。</p> <p>在 第3.1节：自举程序激活 中增加了BOOT1位的情况。</p> <p>在 表3：嵌入式自举程序 中更新了互连型、大容量超低功耗STM32F2xx和STM32F4xx器件。</p> <p>在 表8：STM32F105xx/107xx自举程序版本 中增加了自举程序版本V2.2。</p> <p>在 第5.3.1节：如何识别STM32F105xx/107xx自举程序版本 中增加了自举程序V2.2。</p> <p>在 表15：系统存储器自举模式下STM32L1xxx大容量器件的配置 下增加了DFU接口的相关注释。在 表16：STM32L1xxx大容量器件自举程序版本 中增加了V4.2自举程序已知限制，更新了说明并增加了V4.5自举程序。</p> <p>在 表19：系统存储器自举模式下STM32F2xxx器件的配置 下增加了DFU接口的相关注释。在 表20：STM32F2xxx自举程序V3.x版本 中增加了V3.2自举程序已知限制并增加了V3.3自举程序。在 表21：系统存储器自举模式下STM32F40xxx/41xxx器件的配置 中更新了STM32F2xx和STM32F4xx系统存储器结束地址。</p> <p>在 表21：系统存储器自举模式下STM32F40xxx/41xxx器件的配置 下增加了DFU接口的相关注释。在 表22：STM32F40xxx/41xxx自举程序V3.x版本 中增加了V3.0自举程序已知限制并增加了V3.1自举程序。</p> <p>在 表26：STM32F051xx自举程序版本 中增加了自举程序V2.1已知限制。</p> <p>在 表65：器件相关的自举程序参数 中更新了STM32F051x6/x8系统存储器结束地址。</p> <p>增加了 表75：面向大容量超低功耗器件的USART自举程序时序，和 表78：面向STM32F051xx器件的USART自举程序时序。</p> <p>增加了 表88：面向大容量超低功耗器件的USB最小时序。</p>
2013年1月 24日	15	<p>更新了整个文档的通用产品名称（见 词汇表）。</p> <p>新增了以下各节：</p> <ul style="list-style-type: none"> - 第8节：STM32L1xxxC器件自举程序。 - 第13节：STM32F031xx器件自举程序。 - 第14节：STM32F373xx器件自举程序。 - 第15节：STM32F302xB (C) /303xB (C) 器件自举程序。 - 第16节：STM32F378xx器件自举程序。 - 第17节：STM32F358xx器件自举程序。 - 第18节：STM32F427xx/437xx器件自举程序。 - 第34.3节：I2C自举程序时序特性。 <p>更新了 第1节：相关文献 和 第2节：词汇表。</p> <p>增加了 表79到表85（USART自举程序时序）。</p> <p>用 图16，以及 图18、19和42，替代了 图6。</p> <p>修改了 表3、5、9、11、17、20、21、22至13、27、29、31、33、35、37和65。</p> <p>删除了 第3.3节：硬件连接要求 中的“X = 6：使用了一个USART”。</p> <p>第12.1节：自举程序配置 中用地址0x1FFF F802替代了地址0x1FFFF 8002。</p> <p>修改了 注：第28页， 第6.2节：自举程序选择 和 第9.2节：自举程序选择 中自举程序代码的相关执行步骤。</p>

表165.文档版本历史（续）

日期	版本	变更
2013年2月6日	16	<p>在整个文档中增加了I²C相关的信息。 改进了表1：可应用产品和第1节：相关文献。 表3：嵌入式自举程序修改如下： – 将“V6.0”替换为“V1.0” – 在STM32F31xx行中将“0x1FFFF7A6”替换为“0x1FFFF796” – 在STM32F051xx行中将“0x1FFF7FA6”替换为“0x1FFFF7A6” 更新了图6、9和11。 增加了词汇表中的注：和第3.1节：自举程序激活中的注：。 替换内容： – 在表17、19、19、22、21、27、37和59中将“1.62 V”替换为“1.8 V” – 在表33的RAM行中将“5 KB”替换为“4 KB” – 在表65的行F3中将“127页（每页2 KB）”替换为“4 KB（2页，每页2 KB）” – 在第3.3节：硬件连接要求中将“将自举程序ID编程在器件存储器的最后两个字节中”替换为“将自举程序ID编程在器件存储器的最后一个字节地址 - 1对应的单元中”。 – 在第10节：STM32F2xxx器件自举程序中“将STM32F2xxx器件版本Y”替换为“STM32F2xxx器件版本X和Y” – 在表11、15和26中将“电压范围2”替换为“电压范围1”。</p>
2013年5月21日	17	<p>更新了： – 引言 – 第2节：词汇表 – 第3.3节：硬件连接要求 – 第7节：STM32L1xxx6 (8/B) 器件自举程序中包括了超值型STM32L100 – 第32.2节：USART连接时序 – 第34.2节：USB自举程序时序特性 – 第34.3节：I2C自举程序时序特性 – 表1：适用产品 – 表3：嵌入式自举程序 – 表25：系统存储器自举模式下STM32F051xx器件的配置 – 表27：系统存储器自举模式下STM32F031xx器件的配置 – 表65：器件相关的自举程序参数 – 图17：STM32F031xx器件的自举程序选择 增加了第19节：STM32F429xx/439xx器件自举程序。</p>

表165.文档版本历史（续）

日期	版本	变更
2014年5月 19日	18	<p>增加：</p> <ul style="list-style-type: none"> - 图 1到图 5, 图 75, 图 10, 图 29, 图 30, 图 28, 从图 42到图 97, 图 101 - 表 4, 表 129, 表 130, 从表 13到表 52, 从表 53到表 50, 从表 75到表 76, 从表 到表 164 - 第38.4节, 第 35.2节, 第 74.1节, 第 74.5节 - 第 7节, 第 25节, 第 26节, 第 24节, 从第 19节到第 64节 - 图 1, 图 2, 图 3和图 4下面的注释 <p>更新了：</p> <ul style="list-style-type: none"> - 更新了从第 4节到第 9节和第 20节, 第 35节和第 35节, 篇章结构分三个子章节来组织: 自举程序配置, 自举程序选择和自举程序版本。 - 更新了第 64节和第 74节 - 更新了图 29和图 24的框图。 - 固定了表 73 中STM32F429xx/439xx器件的I2C地址 - 表 1, 表 2, 表 3, 表 31, 表 123, 表 125, 表 127, 表 35, 表 37, 表 57, 表 159 - 从图 18, 到图 32, 图 12, 从图 97到图 101 - 表 124上的注释
2014年7月 29日	19	<p>更新了：</p> <ul style="list-style-type: none"> - 表 2下面的注释 - 图 74和图 75 - 第 3节: 词汇表 - 第 35节: STM32F42xxx/43xxx器件自举程序上用STM32F42xxx/43xxx替换了STM32F427xx/437xx的参考 - 用“STM32F07xxx”替换了“STM32F072xx” - 用“STM32F051xx和STM32F030x8器件”替换了“STM32F051xx”。 - 表 31, 表 37, 表 57, 表 129, 表 73, 表 75, 表 19, 表 25, 表 61, 表 63和表 67上 OTG_FS_DP和OTG_FS_DM相关的注释栏 - 表 129上USB_DM相关注释栏。 - 表 3上用“STM32F42xxx/43xxx”替换了“STM32F429xx/439xx”的参考 - 表 75上SPI2_MOSI、SPI2_MISO、SPI2_SCK和SPI2_NSS引脚的相关注释栏 <p>增加了：</p> <ul style="list-style-type: none"> - 表 2下面的注释 - 表 1, 第 3节: 词汇表, 表 160, 表 161, 表 162, 表 163上STM32F411的参考 - 第 32节: STM32F411xx器件自举程序 <p>删除了表 3, 第 3节: 词汇表, 表 159, 表 160, 表 161, 表 162上STM32F427xx/437xx的参考</p>
2014年11月 24日	20	<p>更新了：</p> <ul style="list-style-type: none"> - 表 129和表 115上“SPI1_NSS引脚”和“SPI2_NSS引脚”行的注释 - 表 61、表 63和表 67上“SPI1_NSS引脚”、“SPI2_NSS引脚”和“PI3_NSS引脚”行的注释 - 图 1

表165.文档版本历史（续）

日期	版本	变更
2015年3月 11日	21	<p>更新了：</p> <ul style="list-style-type: none"> - 表 1、表 3、表 29、表 33、表 123、表 35、表 37、表 38、表 57、表 129、表 17、表 18、表 13、表 41、表 73、表 75、表 19、表 20、表 25、表 26、表 39、表 121、表 137、表 159、表 160、表 161、表 162 和 表 163 - 图 83 - 第 3 节：词汇表 - 第 4.1 节和第 4.4 节 <p>增加了：</p> <ul style="list-style-type: none"> - 第 64 节：STM32L47xxx/48xxx 器件自举程序和 第 36 节：STM32F446xx 器件自举程序
2015年6月9 日	22	<p>增加了：</p> <ul style="list-style-type: none"> - 第 11 节：STM32F070x6 器件自举程序 - 第 12 节：STM32F070xB 器件自举程序 - 第 14 节：STM32F09xxx 器件自举程序 - 第 21 节：STM32F302xD (E) /303xD (E) 器件自举程序 第 27 节：STM32F398xx 器件自举程序 - 第 38 节：STM32F72xxx/73xxx 器件自举程序 - 第 64.2 节：自举程序V9.x - 图 98 上的注释 1 和 2 <p>更新了：</p> <ul style="list-style-type: none"> - 表 1 - 第 3 节：词汇表 - 表 2 - 表 3 - 第 4.4 节：自举程序存储器管理 - 表 159、表 160、表 161、表 162 和 表 163
2015年9月 29日	23	<p>增加了：</p> <ul style="list-style-type: none"> - 第 31 节：STM32F410xx 器件自举程序 - 第 37 节：STM32F469xx/479xx 器件自举程序 - 第 53 节：STM32L031xx/041xx 器件自举程序 - 第 55 节：STM32L07xxx/08xxx 器件自举程序 <p>更新了：</p> <ul style="list-style-type: none"> - 表 1 - 第 3 节：词汇表 - 表 3 - 图 83、表 139、表 160、表 161、表 162、表 163
2015年11月 2日	24	<p>更新了：</p> <ul style="list-style-type: none"> - 表 1，表 3，表 159，表 160，表 161，表 162，表 163 - 第 37 节 <p>增加了：</p> <ul style="list-style-type: none"> - 第 28.2.1 节上的注释 - 第 33 节
2015年12月 1日	25	<p>更新了：</p> <ul style="list-style-type: none"> - 第 4.1 节，第 55 节 - 表 159

表165.文档版本历史（续）

日期	版本	变更
2016年3月3日	26	更新了： - 表 1、表 3、表 70、表 118、表 120、表 159 - 第 3 节，第 55.1.1 节，第 55.2.1 节，第 64 节 增加了： - 第 52 节：STM32L01xxx/02xxx 器件自举程序 - 图 67，图 69
2016年4月21日	27	增加了： - 第 40 节：STM32F76xxx/77xxx 器件自举程序，第 62 节：STM32L43xxx/44xxx 器件自举程序。 - 第 4.1 节：自举程序激活、第 10.1 节：自举程序配置、第 11.1 节：自举程序配置、图 40：STM32F42xxx/43xxx 自举程序V7.x 的双存储区自举实现、图 42：STM32F42xxx/43xxx 自举程序V9.x 的双存储区自举实现上的注释 更新了： - 表 1：适用产品，表 2：自举程序激活模式，表 15：系统存储器自举模式下 STM32F030xC 器件的配置，表 21：系统存储器自举模式下 STM32F070x6 器件的配置，表 23：系统存储器自举模式下 STM32F070xB 器件的配置，表 27：系统存储器自举模式下 STM32F09xxx 器件的配置，表 39：系统存储器自举模式下 STM32F301xx/302x4 (6/8) 器件的配置，表 41：系统存储器自举模式下 STM32F302xB (C) /303xB (C) 器件的配置，表 43：系统存储器自举模式下 STM32F302xD (E) /303xD (E) 器件的配置，表 51：系统存储器自举模式下 STM32F373xx 器件的配置，表 61：系统存储器自举模式下 STM32F401xB (C) 器件的配置，表 63：系统存储器自举模式下 STM32F401xD (E) 器件的配置，表 67：系统存储器自举模式下 STM32F411xx 器件的配置，表 138：STM32L47xxx/48xxx 自举程序V10.x 版本，表 140：STM32L47xxx/48xxx 自举程序V9.x 版本，表 159：器件相关的自举程序参数 - 第 3 节：词汇表，

表165.文档版本历史（续）

日期	版本	变更
2016年9月5日	28	<p>更新了：</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 表 15: 系统存储器自举模式下STM32F030xC器件的配置, 表 17: 系统存储器自举模式下STM32F05xxx和STM32F030x8器件的配置, 表 19: 系统存储器自举模式下STM32F04xxx器件的配置, 表 21: 系统存储器自举模式下STM32F070x6器件的配置, 表 23: 系统存储器自举模式下STM32F070xB器件的配置, 表 25: 系统存储器自举模式下STM32F071xx/072xx器件的配置, 表 27: 系统存储器自举模式下STM32F09xxx器件的配置, 表 31: 系统存储器自举模式下STM32F105xx/107xx器件的配置, 表 33: 系统存储器自举模式下STM32F10xxx超大容量器件的配置, 表 35: 系统存储器自举模式下STM32F2xxxx器件的配置, 表 37: 系统存储器自举模式下STM32F2xxxx器件的配置, 表 39: 系统存储器自举模式下STM32F301xx/302x4 (6/8) 器件的配置, 表 41: 系统存储器自举模式下STM32F302xB (C) /303xB (C) 器件的配置, 表 43: 系统存储器自举模式下STM32F302xD (E) /303xD (E) 器件的配置, 表 45: 系统存储器自举模式下STM32F303x4 (6/8) /334xx/328xx器件的配置, 表 47: 系统存储器自举模式下STM32F318xx器件的配置, 表 49: 系统存储器自举模式下STM32F358xx器件的配置, 表 51: 系统存储器自举模式下STM32F373xx器件的配置, 表 53: 系统存储器自举模式下STM32F378xx器件的配置, 表 55: 系统存储器自举模式下STM32F398xx器件的配置, 表 57: 系统存储器自举模式下STM32F40xxx/41xxx器件的配置, 表 59: 系统存储器自举模式下STM32F40xxx/41xxx器件的配置, 表 61: 系统存储器自举模式下STM32F401xB (C) 器件的配置, 表 63: 系统存储器自举模式下STM32F401xD (E) 器件的配置, 表 67: 系统存储器自举模式下STM32F411xx器件的配置, 表 73: 系统存储器自举模式下STM32F42xxx/43xxx器件的配置, 表 75: 系统存储器自举模式下STM32F42xxx/43xxx器件的配置, 表 77: 系统存储器自举模式下STM32F446xx器件的配置, 表 79: 系统存储器自举模式下STM32F469xx/479xx器件的配置, 表 83: 系统存储器自举模式下STM32F74xxx/75xxx器件的配置, 表 85: 系统存储器自举模式下STM32F74xxx/75xxx器件的配置, 表 115: 系统存储器自举模式下STM32L05xxx/06xxx器件的配置, 表 121: 系统存储器自举模式下STM32L1xxx6 (8/B) A器件的配置, 表 123: 系统存储器自举模式下STM32L1xxx6 (8/B) 器件的配置, 表 125: 系统存储器自举模式下STM32L1xxxC器件的配置, 表 127: 系统存储器自举模式下STM32L1xxxD器件的配置, 表 129: 系统存储器自举模式下STM32L1xxxE器件的配置, 表 134: STM32L43xxx/44xxx自举程序版本, 表 137: 系统存储器自举模式下STM32L47xxx/48xxx器件的配置, 表 159: 器件相关的自举程序参数 - 第 62.1节: 自举程序配置 - 图 26: STM32F303x4 (6/8) /334xx/328xx器件的自举程序选择, 图 27: STM32F318xx器件的自举程序选择, 图 29: STM32F373xx器件的自举程序选择, 图 30: STM32F378xx器件的自举程序选择, 图 33: STM32F40xxx/41xxx器件的自举程序V9.x选择, 图 36: STM32F410xx器件的自举程序V11.x选择, 图 38: STM32F412xx器件的自举程序V9.x选择, 图 46: STM32F469xx/479xx器件的自举程序V9.x选择, 图 51: STM32F76xxx/77xxx器件的自举程序V9.x选择, 图 70: STM32L07xxx/08xxx器件的自举程序V11.x选择, 图 83: STM32L47xxx/48xxx器件的自举程序V10.x选择

表165.文档版本历史 (续)

日期	版本	变更
2016年12月7日	29	<p>更新了:</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 第 3 节: 词汇表, 第 4.1 节: 自举程序激活, 表 3: 嵌入式自举程序, 表 14: STM32F09xxx 器件自举程序, 表 16: STM32F105xx/107xx 器件自举程序, 表 17: STM32F10xxx 超大容量器件自举程序, 表 18: STM32F2xxxx 器件自举程序, 表 19: STM32F301xx/302x4 (6/8) 器件自举程序, 表 20: STM32F302xB (C) /303xB (C) 器件自举程序, 表 22: STM32F303x4 (6/8) /334xx/328xx 器件自举程序, 表 24: STM32F358xx 器件自举程序, 表 27: STM32F398xx 器件自举程序, 表 31: STM32F410xx 器件自举程序, 表 34: STM32F413xx/423xx 器件自举程序, 表 63: 系统存储器自举模式下 STM32F401xD (E) 器件的配置, 第 16.3.1 节: 如何识别 STM32F105xx/107xx 自举程序版本, 第 30.1 节: 自举程序配置, 表 65: 系统存储器自举模式下 STM32F410xx 器件的配置, 表 67: 系统存储器自举模式下 STM32F411xx 器件的配置, 表 69: 系统存储器自举模式下 STM32F412xx 器件的配置, 第 32.1 节: 自举程序配置, 表 74: STM32F42xxx/43xxx 自举程序 V7.x 版本, 表 76: STM32F42xxx/43xxx 自举程序 V9.x 版本, 表 87: 系统存储器自举模式下 STM32F76xxx/77xxx 器件的配置, 表 88: STM32F76xxx/77xxx 自举程序 V9.x 版本, 表 112: STM32L01xxx/02xxx 自举程序版本, 表 120: STM32L07xxx/08xxx 自举程序 V11.x 版本, 表 133: 系统存储器自举模式下 STM32L43xxx/44xxx 器件的配置, 表 134: STM32L43xxx/44xxx 自举程序版本, 表 138: STM32L47xxx/48xxx 自举程序 V10.x 版本, 表 159: 器件相关的自举程序参数, 表 160: STM32 器件的自举程序启动时序 (ms), 表 162: STM32 器件的 USB 自举程序最小时序 (ms), 表 162: STM32 器件的 USB 自举程序最小时序 (ms), 表 163: STM32 器件的 I2C 自举程序最小时序 (ms) <p>增加了:</p> <ul style="list-style-type: none"> - 第 34 节: STM32F413xx/423xx 器件自举程序
2017年3月13日	30	<p>更新了:</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 表 3: 嵌入式自举程序, 表 18: STM32F05xxx 和 STM32F030x8 器件自举程序版本, 表 19: 系统存储器自举模式下 STM32F04xxx 器件的配置, 表 20: STM32F04xxx 自举程序版本, 表 22: STM32F070x6 自举程序版本, 表 24: STM32F070xB 自举程序版本, 表 25: 系统存储器自举模式下 STM32F071xx/072xx 器件的配置, 表 26: STM32F071xx/072xx 自举程序版本, 表 27: 系统存储器自举模式下 STM32F09xxx 器件的配置, 表 28: STM32F09xxx 自举程序版本, 表 39: 系统存储器自举模式下 STM32F301xx/302x4 (6/8) 器件的配置, 表 42: STM32F302xB (C) /303xB (C) 自举程序版本, 表 88: STM32F76xxx/77xxx 自举程序 V9.x 版本, 表 111: 系统存储器自举模式下 STM32L01xxx/02xxx 器件的配置, 表 134: STM32L43xxx/44xxx 自举程序版本, 表 159: 器件相关的自举程序参数, 表 139: 系统存储器自举模式下 STM32L47xxx/48xxx 器件的配置, 表 160: STM32 器件的自举程序启动时序 (ms), 表 161: STM32 器件的 USART 自举程序最小时序 (ms), 表 162: STM32 器件的 USB 自举程序最小时序 (ms), 表 163: STM32 器件的 I2C 自举程序最小时序 (ms), 表 164: STM32 器件的 SPI 自举程序最小时序 (ms) - 第 3 节: 词汇表、第 8.1 节: 自举程序配置、第 16.3.3 节: USART 自举程序 Get-Version 命令返回 0x20 而不是 0x22、第 62 节: STM32L43xxx/44xxx 器件自举程序和第 64 节: STM32L47xxx/48xxx 器件自举程序中的 RPN 参考 <p>增加了第 38 节: STM32F72xxx/73xxx 器件自举程序和第 65 节: STM32L496xx/4A6xx 器件自举程序。</p>

表165.文档版本历史 (续)

日期	版本	变更
2017年7月4日	31	<p>更新了:</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 表 2: 自举程序激活模式, 表 3: 嵌入式自举程序, 表 32: STM32F105xx/107xx自举程序版本, 表 37: 系统存储器自举模式下STM32F2xxxx器件的配置, 表 41: 系统存储器自举模式下STM32F302xB (C) /303xB (C) 器件的配置, 表 49: 系统存储器自举模式下STM32F358xx器件的配置, 表 51: 系统存储器自举模式下STM32F373xx器件的配置, 表 53: 系统存储器自举模式下STM32F378xx器件的配置, 表 59: 系统存储器自举模式下STM32F40xxx/41xxx器件的配置, 表 61: 系统存储器自举模式下STM32F401xB (C) 器件的配置, 表 63: 系统存储器自举模式下STM32F401xD (E) 器件的配置, 表 67: 系统存储器自举模式下STM32F411xx器件的配置, 表 73: 系统存储器自举模式下STM32F42xxx/43xxx器件的配置, 表 77: 系统存储器自举模式下STM32F446xx器件的配置, 表 79: 系统存储器自举模式下STM32F469xx/479xx器件的配置, 表 81: 系统存储器自举模式下STM32F72xxx/73xxx器件的配置, 表 83: 系统存储器自举模式下STM32F74xxx/75xxx器件的配置, 表 85: 系统存储器自举模式下STM32F74xxx/75xxx器件的配置, 表 107: 系统存储器自举模式下STM32H74xxx/75xxx器件的配置, 表 125: 系统存储器自举模式下STM32L1xxxC器件的配置, 表 127: 系统存储器自举模式下STM32L1xxxD器件的配置, 表 129: 系统存储器自举模式下STM32L1xxxE器件的配置, 表 135: 系统存储器自举模式下STM32L45xxx/46xxx器件的配置, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 162: STM32器件的USB自举程序最小时序 (ms), 表 163: STM32器件的I2C自举程序最小时序 (ms) - 引言, 第 3节: 词汇表 - 图 79: STM32L43xxx/44xxx器件的自举程序V9.x选择 <p>增加了:</p> <ul style="list-style-type: none"> - 第 50节: STM32H74xxx/75xxx器件自举程序, 第 63节: STM32L45xxx/46xxx器件自举程序
2018年2月16日	32	<p>更新了表 3: 嵌入式自举程序、表 108: STM32H74xxx/75xxx自举程序版本、表 141: 系统存储器自举模式下STM32L496xx/4A6xx器件的配置、表 142: STM32L496xx/4A6xx自举程序版本、表 159: 器件相关的自举程序参数、表 160: STM32器件的自举程序启动时序 (ms)、表 161: STM32器件的USART自举程序最小时序 (ms)、表 162: STM32器件的USB自举程序最小时序 (ms) 和表 163: STM32器件的I2C自举程序最小时序 (ms)。</p> <p>增加了第 67节: STM32L4Rxxx/4Sxxx器件自举程序</p>
2018年8月7日	33	<p>更新了第 10.1节: 自举程序配置中的 注: 和第 11.1节: 自举程序配置中的 注:</p>
2018年11月5日	34	<p>更新了表 1: 适用产品, 表 58: STM32F40xxx/41xxx自举程序V3.x版本, 表 60: STM32F40xxx/41xxx自举程序V9.x版本, 表 62: STM32F401xB (C) 自举程序版本, 表 64: STM32F401xD (E) 自举程序版本, 表 66: STM32F410xx自举程序V11.x版本, 表 68: STM32F411xx自举程序版本, 表 70: STM32F412xx自举程序V9.x版本, 表 72: STM32F413xx/423xx自举程序V9.x版本, 表 74: STM32F42xxx/43xxx自举程序V7.x版本, 表 76: STM32F42xxx/43xxx自举程序V9.x版本, 表 78: STM32F446xx自举程序V9.x版本, 表 80: STM32F469xx/479xx自举程序V9.x版本, 表 82: STM32F72xxx/73xxx自举程序V9.x版本, 表 84: STM32F74xxx/75xxx自举程序V7.x版本, 表 86: STM32F74xxx/75xxx自举程序V9.x版本, 表 88: STM32F76xxx/77xxx自举程序V9.x版本, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 162: STM32器件的USB自举程序最小时序 (ms)。</p> <p>增加了第 61节: STM32L412xx/422xx器件自举程序</p>

表165.文档版本历史（续）

日期	版本	变更
2018年12月6日	35	更新了表 1: 适用产品、第 3 节: 词汇表、表 160: STM32器件的自举程序启动时序 (ms)、表 161: STM32器件的USART 自举程序最小时序 (ms) 和表 163: STM32器件的I2C 自举程序最小时序 (ms)。 增加了第 42 节: STM32G07xxx/08xxx 器件自举程序
2019年2月21日	36	更新了表 1: 适用产品, 第 3 节: 词汇表, 表 3: 嵌入式自举程序, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART 自举程序最小时序 (ms), 表 162: STM32器件的USB 自举程序最小时序 (ms), 表 163: STM32器件的I2C 自举程序最小时序 (ms)。 增加了第 70 节: STM32WB30xx/35xx/50xx/55xx 器件自举程序
2019年5月6日	37	更新了表 1: 适用产品, 第 3 节: 词汇表, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART 自举程序最小时序 (ms), 表 162: STM32器件的USB 自举程序最小时序 (ms), 表 163: STM32器件的I2C 自举程序最小时序 (ms)。 增加了第 46 节: STM32G431xx/441xx 器件自举程序, 第 47 节: STM32G47xxx/48xxx 器件自举程序
2019年7月8日	38	更新了: - 表 1: 适用产品, 表 2: 自举程序激活模式, 表 3: 嵌入式自举程序, 表 71: 系统存储器自举模式下STM32F413xx/423xx 器件的配置, 表 107: 系统存储器自举模式下STM32H74xxx/75xxx 器件的配置, 表 108: STM32H74xxx/75xxx 自举程序版本, 表 113: 系统存储器自举模式下STM32L031xx/041xx 器件的配置, 表 134: STM32L43xxx/44xxx 自举程序版本, 表 135: 系统存储器自举模式下STM32L45xxx/46xxx 器件的配置, 表 142: STM32L496xx/4A6xx 自举程序版本, 表 153: STM32WB30xx/35xx/50xx/55xx 自举程序版本, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART 自举程序最小时序 (ms), 表 162: STM32器件的USB 自举程序最小时序 (ms), 表 163: STM32器件的I2C 自举程序最小时序 (ms) - 第 3 节: 词汇表, 第 4.1 节: 自举程序激活, 第 41.1 节: 自举程序配置, 第 46.1 节: 自举程序配置 - 图 62: STM32H74xxx/75xxx 器件的自举程序V9.x选择, 图 90: STM32L4Rxxx/STM32L4Sxxx 自举程序V9.x的双存储区自举实现 增加了第 4.2 节中的注:, 第 15.3 节中的注:, 第 50.1 节中的注:, 第 52.1 节中的注:, 以及第 41 节: STM32G03xxx/STM32G04xxx 器件自举程序
2019年9月16日	39	更新了: - 表 1: 适用产品, 表 2: 自举程序激活模式, 表 3: 嵌入式自举程序, 表 90: STM32G03xxx/04xxx 自举程序版本, 表 132: STM32L412xx/422xx 自举程序版本, 表 134: STM32L43xxx/44xxx 自举程序版本, 表 136: STM32L45xxx/46xxx 自举程序版本, 表 138: STM32L47xxx/48xxx 自举程序V10.x版本, 表 140: STM32L47xxx/48xxx 自举程序V9.x版本, 表 142: STM32L496xx/4A6xx 自举程序版本, 表 144: STM32L4P5xx/4Q5xx 自举程序版本, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART 自举程序最小时序 (ms), 表 162: STM32器件的USB 自举程序最小时序 (ms), 表 163: STM32器件的I2C 自举程序最小时序 (ms) - 第 3 节: 词汇表, 第 4.2 节: 自举程序标识 增加了图 59: STM32G47xxx/48xxx 自举程序V13.x的双存储区自举实现, 第 68 节: STM32L552xx/STM32L562xx 器件自举程序, 以及第 70.3 节: 自举程序版本中的注释
2019年10月3日	40	更新了表 3: 嵌入式自举程序、表 149: STM32L552xx/562xx 自举程序版本、表 153: STM32WB30xx/35xx/50xx/55xx 自举程序版本

表165.文档版本历史（续）

日期	版本	变更
2019年10月25日	41	<p>更新了：</p> <ul style="list-style-type: none"> - 表 82: STM32F72xxx/73xxx自举程序V9.x版本, 表 84: STM32F74xxx/75xxx自举程序V7.x版本, 表 86: STM32F74xxx/75xxx自举程序V9.x版本, 表 88: STM32F76xxx/77xxx自举程序V9.x版本, 表 89: 系统存储器自举模式下STM32G03xxx/G04xxx器件的配置, 表 108: STM32H74xxx/75xxx自举程序版本, 表 144: STM32L4P5xx/4Q5xx自举程序版本, 表 147: 系统存储器自举模式下STM32L552xx/562xx器件的配置, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 163: STM32器件的I2C自举程序最小时序 (ms) - 第 18节: STM32F2xxxx器件自举程序
2019年12月5日	42	<p>更新了：</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 表 2: 自举程序激活模式, 表 3: 嵌入式自举程序, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 162: STM32器件的USB自举程序最小时序 (ms), 表 163: STM32器件的I2C自举程序最小时序 (ms) - 第 3节: 词汇表 <p>增加: 第 51节: STM32H7A3xx/B3xx器件自举程序, 第 66节: STM32L4P5xx/4Q5xx器件自举程序, 第 71节: STM32WLE5xx/55xx器件自举程序</p>
2020年6月4日	43	<p>更新了：</p> <ul style="list-style-type: none"> - 表 1: 适用产品, 表 2: 自举程序激活模式, 表 3: 嵌入式自举程序, 表 99: 系统存储器自举模式下STM32G431xx/441xx器件的配置, 表 101: 系统存储器自举模式下STM32G47xxx/48xxx器件的配置, 表 102: STM32G47xxx/48xxx自举程序版本, 表 108: STM32H74xxx/75xxx自举程序版本, 表 110: STM32H7A3xx/7B3xx自举程序版本, 表 144: STM32L4P5xx/4Q5xx自举程序版本, 表 147: 系统存储器自举模式下STM32L552xx/562xx器件的配置, 表 149: STM32L552xx/562xx自举程序版本, 表 152: 系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx的配置, 表 159: 器件相关的自举程序参数 - 第 3节: 词汇表, 第 39节: STM32F74xxx/75xxx器件自举程序, 第 41.1节: 自举程序配置, 第 42.1节: 自举程序配置, 第 46.1节: 自举程序配置, 第 47.1节: 自举程序配置, 第 50.1节: 自举程序配置 <p>增加了：</p> <ul style="list-style-type: none"> - 第 4.5节: 自举程序UART波特率检测, 第 4.6节: 编程限制, 第 4.7节: ExitSecureMemory特性 - 以下章节中的 注: : 第 28.1.1节: 自举程序配置, 第 28.2.1节: 自举程序配置, 第 29.1节: 自举程序配置, 第 30.1节: 自举程序配置, 第 32.1节: 自举程序配置, 第 33.1节: 自举程序配置, 第 34.1节: 自举程序配置, 第 35.1.1节: 自举程序配置, 第 35.2.1节: 自举程序配置, 第 36.1节: 自举程序配置, 第 37.1节: 自举程序配置, 第 38.1节: 自举程序配置, 第 39.1.1节: 自举程序配置, 第 39.2.1节: 自举程序配置, 第 40.1节: 自举程序配置 - 图 78: STM32L3x2xx/44xxx自举程序V9.x的双存储区自举实现, 图 80: STM32L45xxx/46xxx自举程序V9.x的双存储区自举实现, 图 86: STM32L496xx/4A6xx自举程序V9.x的双存储区自举实现 - 附录A: 使用“ExitSecureMemory”函数的函数示例 <p>删除了图48。从STM32G03xxx/G04xxx的自举程序访问安全存储区, 图50.从STM32G07xxx/G08xxx的自举程序访问安全存储区, 图52.访问安全存储区, 图54.访问安全存储区</p>

表165.文档版本历史（续）

日期	版本	变更
2020年7月 29日	44	<p>引入了STM32H72xxx/73xxx器件，因此增加了第 49 节：STM32H72xxx/73xxx 器件自举程序及其子小节。</p> <p>更新了第 3 节：词汇表，第 41.1 节：自举程序配置中的注释，以及第 70.1 节：自举程序配置。</p> <p>更新了表 1：适用产品、表 2：自举程序激活模式、表 3：嵌入式自举程序、表 8：ExitSecureMemory 入口地址、表 101：系统存储器自举模式下STM32G47xxx/48xxx 器件的配置、表 110：STM32H7A3xx/7B3xx 自举程序版本、表 125：系统存储器自举模式下STM32L1xxxC 器件的配置、表 127：系统存储器自举模式下STM32L1xxxD 器件的配置、表 129：系统存储器自举模式下STM32L1xxxE 器件的配置、表 147：系统存储器自举模式下STM32L552xx/562xx 器件的配置、表 159：器件相关的自举程序参数、表 160：STM32 器件的自举程序启动时序 (ms)、表 161：STM32 器件的USART 自举程序最小时序 (ms)、表 162：STM32 器件的USB 自举程序最小时序 (ms) 和表 163：STM32 器件的I2C 自举程序最小时序 (ms)。</p> <p>更新了图 62：STM32H74xxx/75xxx 器件的自举程序V9.x 选择。</p> <p>对整个文档进行少量文字修订。</p>
2020年11月 6日	45	<p>在表 1：适用产品、表 3：嵌入式自举程序和第 3 节：词汇表中引入了STM32WB30xx、STM32WB35xx和STM32WI55xx</p> <p>更新了：</p> <ul style="list-style-type: none"> - 表 65：系统存储器自举模式下STM32F410xx 器件的配置，表 71：系统存储器自举模式下STM32F413xx/423xx 器件的配置，表 77：系统存储器自举模式下STM32F446xx 器件的配置，表 79：系统存储器自举模式下STM32F469xx/479xx 器件的配置，表 81：系统存储器自举模式下STM32F72xxx/73xxx 器件的配置，表 85：系统存储器自举模式下STM32F74xxx/75xxx 器件的配置，表 87：系统存储器自举模式下STM32F76xxx/77xxx 器件的配置，表 91：系统存储器自举模式下STM32G07xxx/8xxx 器件的配置，表 99：系统存储器自举模式下STM32G431xx/441xx 器件的配置，表 100：STM32G431xx/441xx 自举程序版本，表 101：系统存储器自举模式下STM32G47xxx/48xxx 器件的配置，表 107：系统存储器自举模式下STM32H74xxx/75xxx 器件的配置，表 108：STM32H74xxx/75xxx 自举程序版本，表 109：系统存储器自举模式下STM32H7A3xx/7B3xx 器件的配置，表 110：STM32H7A3xx/7B3xx 自举程序版本，表 111：系统存储器自举模式下STM32L01xxx/02xxx 器件的配置，表 113：系统存储器自举模式下STM32L031xx/041xx 器件的配置，表 118：STM32L07xxx/08xxx 自举程序版本，表 119：系统存储器自举模式下STM32L07xxx/08xxx 器件的配置，表 131：系统存储器自举模式下STM32L412xx/422xx 器件的配置，表 143：系统存储器自举模式下STM32L4P5xx/4Q5xx 器件的配置，表 145：系统存储器自举模式下STM32L4Rxxx/4Sxxx 器件的配置，表 147：系统存储器自举模式下STM32L552xx/562xx 器件的配置，表 152：系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx 的配置，表 154：系统存储器自举模式下STM32WLE5xx/55xx 器件的配置，表 155：STM32WLE5xx/55xx 自举程序版本，表 159：器件相关的自举程序参数，表 160：STM32 器件的自举程序启动时序 (ms)，表 161：STM32 器件的USART 自举程序最小时序 (ms) - 表 70：STM32WB30xx/35xx/50xx/55xx 器件自举程序、表 152：系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx 的配置、表 71：STM32WLE5xx/55xx 器件自举程序、表 154：系统存储器自举模式下STM32WLE5xx/55xx 器件的配置、表 95：STM32WLE5xx/55xx 的自举程序V12.x 选择和表 155：STM32WLE5xx/55xx 自举程序版本的标题

表165.文档版本历史 (续)

日期	版本	变更
2020年12月2日	46	更新了： - 表 3: 嵌入式自举程序, 表 89: 系统存储器自举模式下STM32G03xxx/G04xxx器件的配置, 表 101: 系统存储器自举模式下STM32G47xxx/48xxx器件的配置, 表 134: STM32L43xxx/44xxx自举程序版本, 表 136: STM32L45xxx/46xxx自举程序版本 增加了以下注释： - 注: 第 304页, 注: 第 310页, 注: 第 317页, 注: 第 328页, 注: 第 335页
2021年2月16日	47	更新了： - 表 1: 适用产品, 表 3: 嵌入式自举程序, 表 8: ExitSecureMemory入口地址, 表 88: STM32F76xxx/77xxx自举程序V9.x版本, 表 101: 系统存储器自举模式下STM32G47xxx/48xxx器件的配置, 表 131: 系统存储器自举模式下STM32L412xx/422xx器件的配置, 表 133: 系统存储器自举模式下STM32L43xxx/44xxx器件的配置, 表 135: 系统存储器自举模式下STM32L45xxx/46xxx器件的配置, 表 139: 系统存储器自举模式下STM32L47xxx/48xxx器件的配置, 表 141: 系统存储器自举模式下STM32L496xx/4A6xx器件的配置, 表 143: 系统存储器自举模式下STM32L4P5xx/4Q5xx器件的配置, 表 145: 系统存储器自举模式下STM32L4Rxx/4Sxx器件的配置, 表 147: 系统存储器自举模式下STM32L552xx/562xx器件的配置, 表 152: 系统存储器自举模式下STM32WB30xx/35xx/50xx/55xx的配置, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 162: STM32器件的USB自举程序最小时序 (ms), 表 163: STM32器件的I2C自举程序最小时序 (ms) - 第 3节: 词汇表 增加了第 43节: STM32G0B0xx器件自举程序和第 44节: STM32G0B1xx/0C1xx器件自举程序。
2021年4月1日	48	更新了： - 表 1: 适用产品, 表 3: 嵌入式自举程序, 表 8: ExitSecureMemory入口地址, 表 159: 器件相关的自举程序参数, 表 160: STM32器件的自举程序启动时序 (ms), 表 161: STM32器件的USART自举程序最小时序 (ms), 表 163: STM32器件的I2C自举程序最小时序 (ms) 增加了第 45节: STM32G05xxx/061xx器件自举程序和第 48节: STM32G491xx/4A1xx器件自举程序。
2021年7月6日	49	更新了： - 第 3节: 词汇表, 第 28.2.1节: 自举程序配置 - 表 3、表 13、表 15至表 20、表 23至表 28、表 29、表 31、表 33、表 37、表 39、表 41、表 43、表 45、表 47、表 49、表 51、表 53、表 55、表 57、表 59、表 60、表 61、表 63、表 65、表 67、表 69、表 71、表 73、表 75、表 77、表 79、表 81、表 83、表 85、表 87、表 89、表 91、表 93、表 95、表 97、表 99、表 101、表 102、表 103、表 105、表 107、表 109、表 109、表 111、表 113、表 115、表 116、表 117、表 119、表 121、表 123、表 124、表 125、表 126、表 127、表 129、表 130、表 131、表 133、表 135、表 137、表 139、表 141、表 143、表 145、表 147、表 152、表 154和表 159 增加了表 148: STM32L552cc/562xx特殊命令和第 69节: STM32WB10xx/15xx器件自举程序。
2021年9月23日	50	更新了： - 第 3节: 词汇表, 第 43.1节: 自举程序配置, 第 44.1节: 自举程序配置, - 表 1, 表 2, 表 3, 表 92, 表 106, 表 108, 表 110, 表 134, 表 151, 表 153, 表 159, 表 160, 表 161, 表 162, 表 163 增加了第 72节: STM32U575xx/85xx器件自举程序

表165.文档版本历史 (续)

日期	版本	变更
2021年10月 20日	51	更新了: - 表 3, 表 60, 表 92, 表 106, 表 159 - 第 28.2.1节: 自举程序配置
2022年2月4 日	52	更新了: - 第 3节: 词汇表, 第 4.1节: 自举程序激活, - 表 1, 表 2, 表 3, 表 7, 表 92, 表 108, 表 159 - 图 54 增加了第 5节: STM32C011xx器件自举程序和 第 6节: STM32C031xx器件自举程序。
2022年3月1 日	53	更新了: - 表 3, 表 105, 表 106, 表 159。 - 第 4.1节: 自举程序激活, 第 41.1节: 自举程序配置, 第 42.1节: 自举程序配置, 第 43.1节: 自举程序配置, 第 45.1节: 自举程序配置

表166.中文文档版本历史

日期	版本	变更
2022年8月9 日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST产品和/或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于ST产品的最新信息。ST产品的销售依照订单确认时的相关ST销售条款。

买方自行负责对ST产品的选择和使用，ST不承担与应用协助或买方产品设计相关的任何责任。

ST不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST产品如有不同于此处提供的信息的规定，将导致ST针对该产品授予的任何保证失效。

ST和ST徽标是ST的商标。若需ST商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。

© 2022 STMicroelectronics - 保留所有权利