

LYF-CH347 高速编程器 V1.0 使用手册

软件名称：LYF-CH347 高速编程器

版本号：V1.0

手册首次发布日期：2025 年 7 月 4 日

公司/团队名称：冷月枫电子工作室

版权声明：©2025 冷月枫电子工作室，保留所有权利

日期	版本	描述	作者
2025.7.4	V1.0	初始发布	Lengyuefeng

目录

一、 简介	3
1.1 软件概述	3
1.2 系统要求	3
1.3 安装与卸载指南	3
二、 快速入门	3
三、 功能模块详解	4
3.1 主界面布局	4
3.2 设备连接	5
3.3 文件管理	6
3.4 辅助操作	8
3.5 编程操作	15
3.6 编程器设置	20
3.7 语言	21
3.8 关于	21
四、 常见问题（FAQ）	22
五、 安全警告	22
六、 附录	22

一、简介

1.1 软件概述

用途：此软件适用于冷月枫工作室出品的 USB 高速编程器，用于读写 EEPROM 和 FLASH 等存储芯片

支持设备：支持 I2C_24_EEPROM、SPI_25_NOR_FLASH、SPI_45_DATA_FLASH 和 SPI_95_EEPROM

新增亮点功能：增加电压自动切换、增加时钟频率自动切换

1.2 系统要求

操作系统：Windows 10/11（64 位）

USB 接口：USB 2.0 或者 USB 3.0

设备驱动：需安装 CH347 驱动程序（可官网下载）

1.3 安装与卸载指南

软件免安装，直接双击打开即可使用

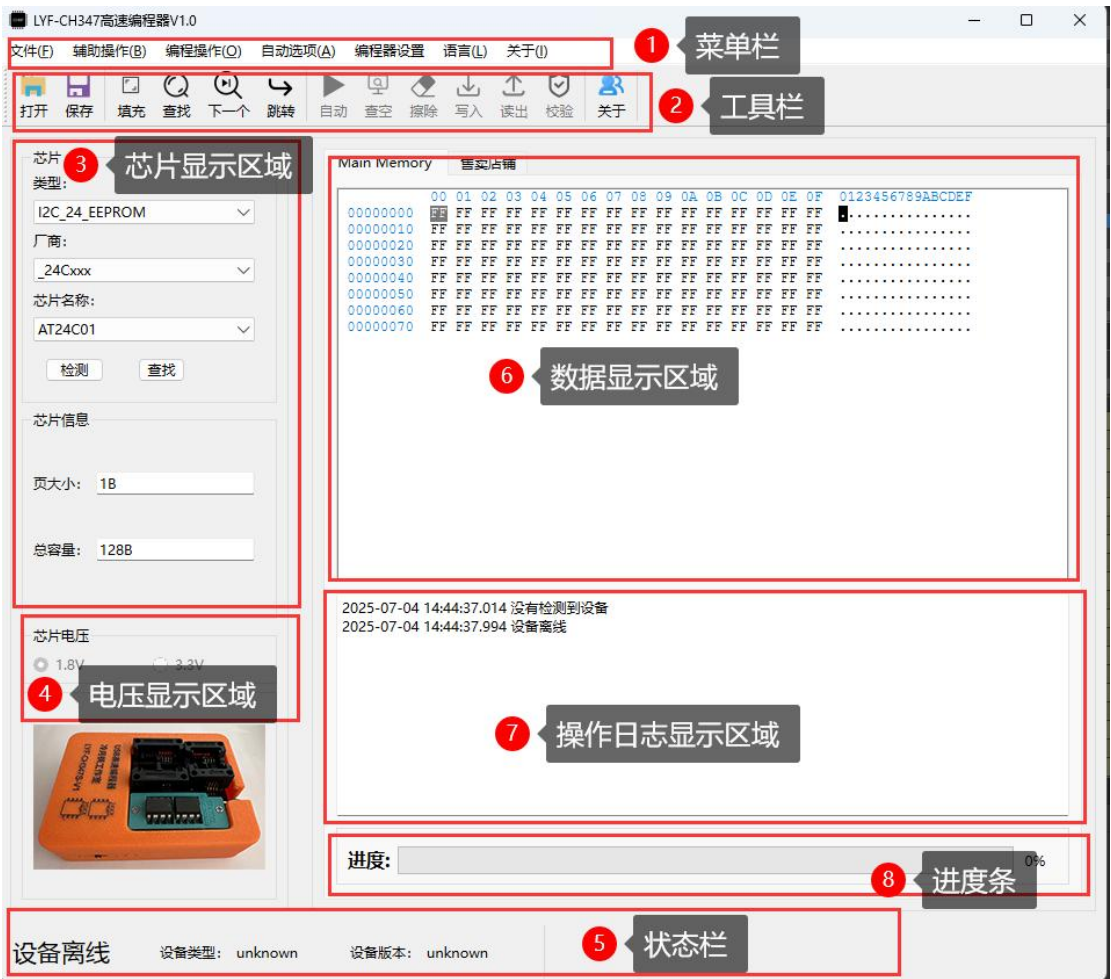
二、快速入门

- 1、放置芯片：将芯片放置在烧录座上
- 2、连接硬件：将编程器通过 USB 连接到电脑
- 3、启动软件：双击打开软件，识别成功后软件底部状态栏显示设备在线
- 4、烧录示例：
 - 1、选择芯片型号（直接搜索或者点击检测按钮（芯片需要有 ID））
 - 2、打开要烧录的文件
 - 3、点击自动按钮完成烧录（如果勾选校验会自动校验）

三、功能模块详解

3.1 主界面布局

软件界面功能区域划分如下：



①菜单栏：

所有功能选项

②工具栏：

功能按钮快速点击

③芯片显示区域：

显示当前操作的芯片型号、厂商、芯片名称、芯片大小和芯片容量等

④电压显示区域

显示当前设置的电压，也可点击对应按钮设置输出电压

⑤状态栏:

显示当前设备连接状态，在线或者离线以及对应设备类型和设备版本等

⑥数据显示区域:

显示文件打开内容或者读取的芯片的数据以及要操作的数据

⑦操作日志显示区域:

显示当前操作状态，设备状态、读取状态、擦除状态、写入状态和设置状态

⑧进度条:

大概表现当前操作的进度

3.2 设备连接

启动软件后，如果无设备插入，显示状态为离线



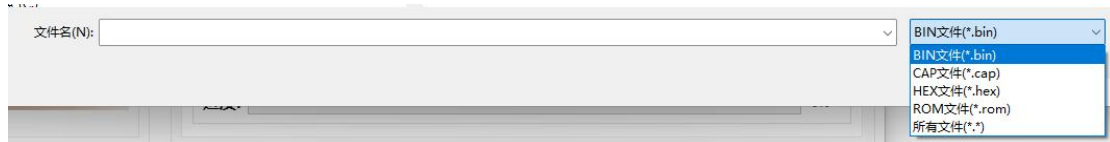
此时如果插入设备，设备显示在线



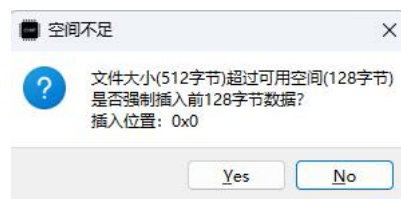
3.3 文件管理

3.3.1 文件打开

软件支持 bin/cap/hex/rom 文件格式

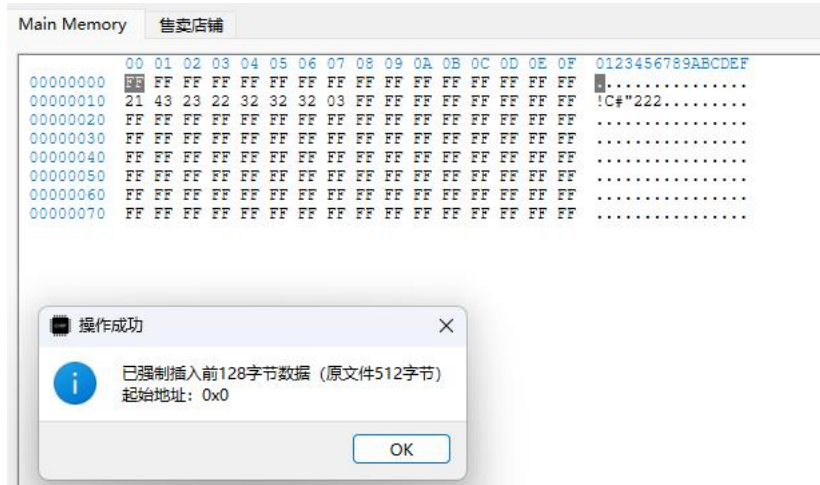


当打开文件数据长度超出当前选中芯片总容量大小，会进行提示空间不足

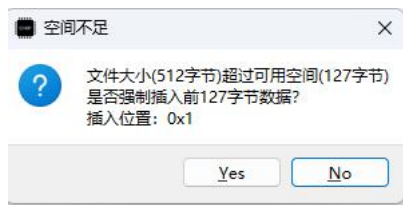
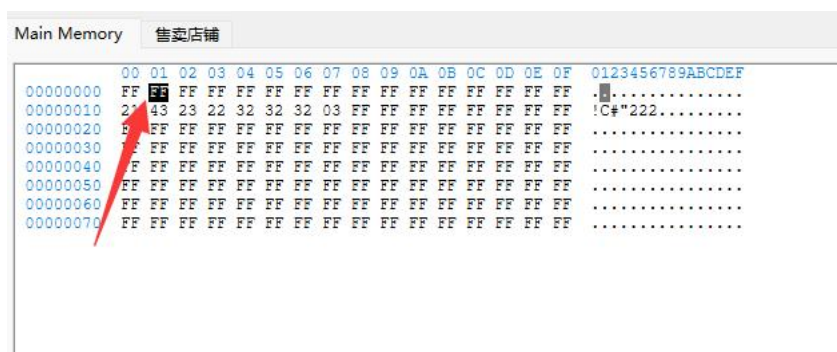


点击 No 忽略此次操作，不打开文件

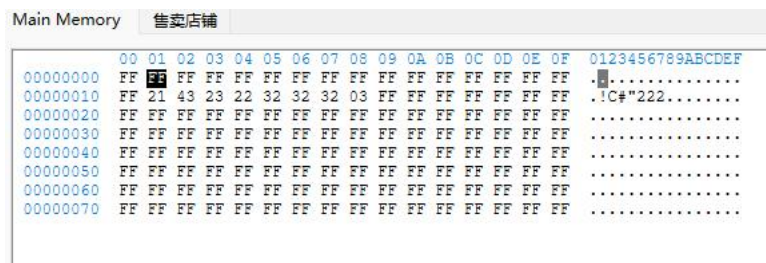
点击 Yes 将会打开文件的前 128（这个数据根据芯片容量变化）字节填充到
数据显示区域



此次需要注意，在数据显示区域的光标是第一个字节，这里默认插入的首地址为光标所在位置，比如将光标放在第二个字节，此时再打开文件



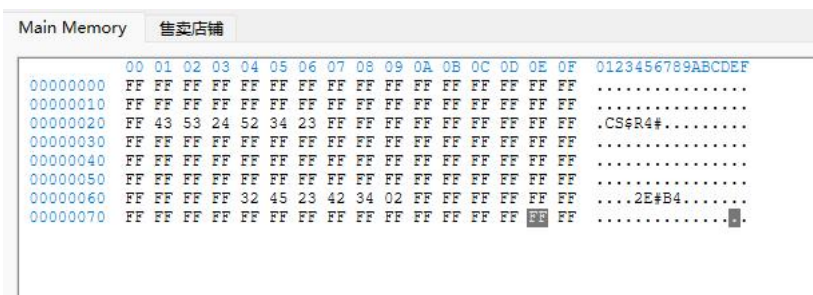
此次将从第 2 个字节开始插入，插入数据为剩下的 127 个字节



3.3.2 文件保存

点击保存按钮可将数据显示区域的数据保存为文件。

此时数据显示区域数据为：



点击保存按钮，保存文件为 testSaveFile.bin



提示文件保存路径



用 winhex 工具打开 testSaveFile.bin 文件，查看是否和数据显示区域数据一致。

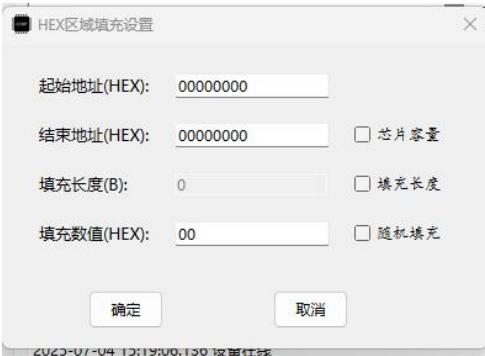
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
00000010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
00000020	FF	43	53	24	52	34	23	FF	FF	FF	FF	FF	FF	FF	FF	FF	yCS\$R4#yyyyyyyy
00000030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
00000040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
00000050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy
00000060	FF	FF	FF	FF	32	45	23	42	34	02	FF	FF	FF	FF	FF	FF	yyyy2E#B4 yyyyyy
00000070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyy

通过对比可得数据保存文件和之前数据显示区域数据一致。

3.4 辅助操作

3.4.1 填充

通过此按钮可以在数据显示区域填充指定的数据

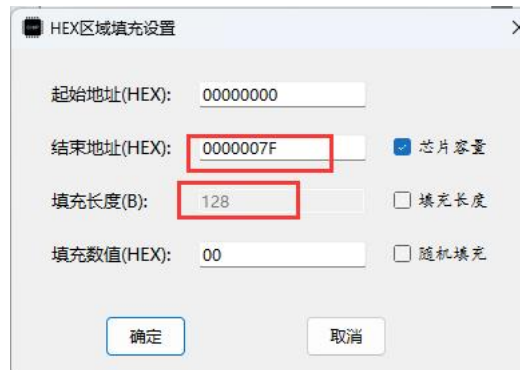


默认都是从起始地址开始填充的，填充数据长度有两种方式，一种是输入结束地址，另一种是输入填充的长度，默认是使用结束地址，勾选填充长度后使用填充长度（可输入填充长度）

指定填充内容：直接在填充数值区域输入要填充的数据（一个字节）或者勾选后面的随机填充，将会随机填充指定长度的数据。

这里有三个可选项

芯片容量：勾选后自动将结束地址和填充长度设置为芯片的容量大小，可以通过此命令直接填充整个芯片。



填充长度：勾选填充长度后使用填充长度，可手动输入填充数据长度

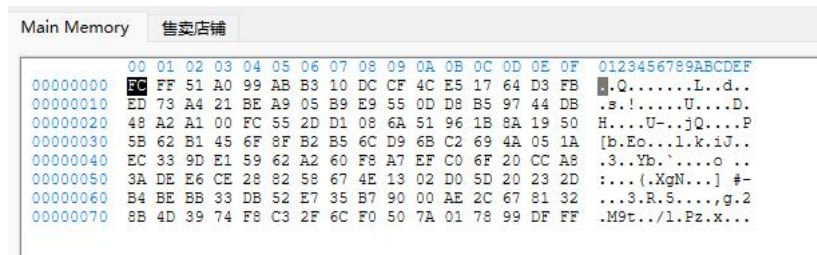


随机填充：勾选堆积填充后，点击确定后将在数据显示区域填充随机的数据

比如，这里勾选芯片容量和随机填充，将会在数据显示区域随机填充所有数据



随机填充的数据



3.4.2 查找

查找功能可以查找数据显示区域指定的内容



在上面查找内容框中输入要查找的内容，可以是一个字节、两个字节或者多个字节（以查找 0x74 为例）

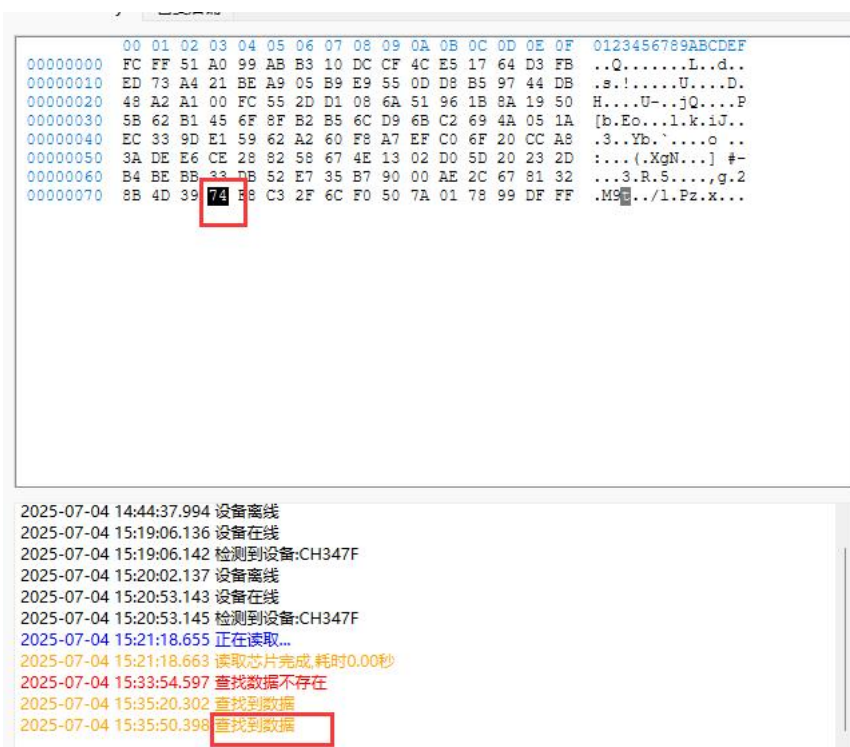
查找数据不存在：

2025-07-04 15:33:54.597 查找数据不存在

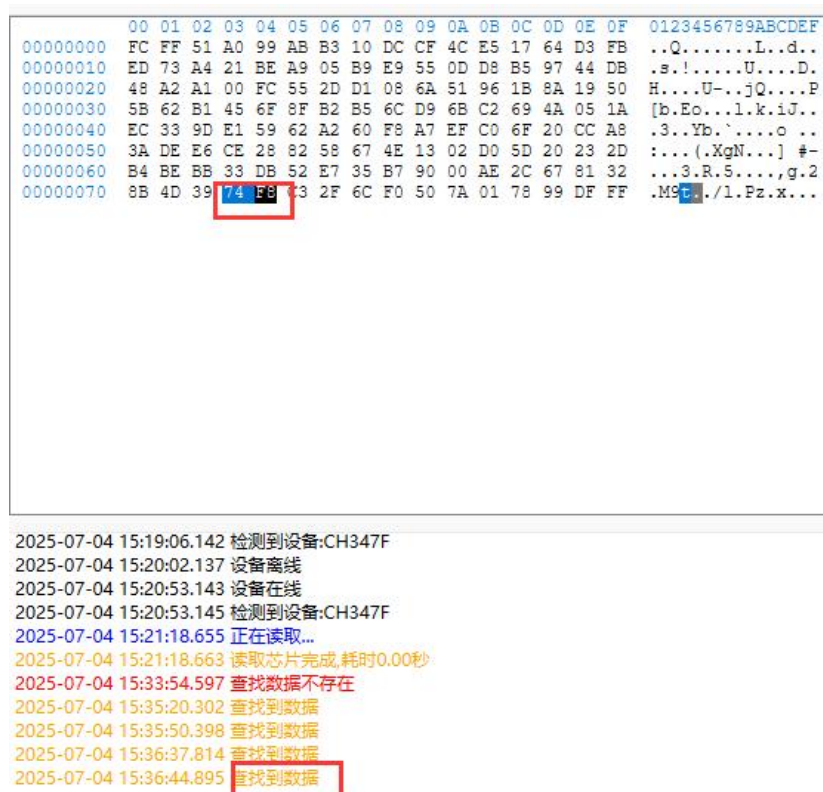
查找一个字节：



提示查找到数据，并且查找到的数据高亮显示



查找两个字节:



查找四个字节:

Main Memory

售卖店铺

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	FC	FF	51	A0	99	AB	B3	10	DC	CF	4C	E5	17	64	D3	FB	..Q.....L..d..
00000010	ED	73	A4	21	BE	A9	05	B9	E9	55	0D	D8	B5	97	44	DB	.s!.....U....D.
00000020	48	A2	A1	00	FC	55	2D	D1	08	6A	51	96	1B	8A	19	50	H....U-..jQ....P
00000030	5B	62	B1	45	6F	8F	B2	B5	6C	D9	6B	C2	69	4A	05	1A	[b.Eo...l.k.iJ..
00000040	EC	33	9D	E1	59	62	A2	60	F8	A7	EF	C0	6F	20	CC	A8	.3..Yb.....o..
00000050	3A	DE	E6	CE	28	82	58	67	4E	13	02	D0	5D	20	23	2D	:...(.XgN...) #-
00000060	B4	BE	BB	33	DB	52	E7	35	B7	90	00	AE	2C	67	81	32	...3.R.5.....g.2
00000070	8B	4D	39	74	F8	C3	2F	6C	F0	50	7A	01	78	99	DF	FF	.M9t.../1.Pz.x...

2025-07-04 15:20:02.137 设备离线

2025-07-04 15:20:53.143 设备在线

2025-07-04 15:20:53.145 检测到设备:CH347F

2025-07-04 15:21:18.655 正在读取...

2025-07-04 15:21:18.663 读取芯片完成,耗时0.00秒

2025-07-04 15:33:54.597 查找数据不存在

2025-07-04 15:35:20.302 查找数据

2025-07-04 15:35:50.398 查找数据

2025-07-04 15:36:37.814 查找数据

2025-07-04 15:36:44.895 查找数据

2025-07-04 15:37:23.223 查找数据

3.4.3 下一个

查找下一个功能是在查找功能基础上实现的,执行前必须执行一次查找功能

比如这里查找 0x62 (数据显示区域 0x62 有两个)

第一次查找:

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	FC	FF	51	A0	99	AB	B3	10	DC	CF	4C	E5	17	64	D3	FB	..Q.....L..d..
00000010	ED	73	A4	21	BE	A9	05	B9	E9	55	0D	D8	B5	97	44	DB	.s!.....U....D.
00000020	48	A2	A1	00	FC	55	2D	D1	08	6A	51	96	1B	8A	19	50	H....U-..jQ....P
00000030	5B	62	B1	45	6F	8F	B2	B5	6C	D9	6B	C2	69	4A	05	1A	[b.Eo...l.k.iJ..
00000040	EC	33	9D	E1	59	62	A2	60	F8	A7	EF	C0	6F	20	CC	A8	.3..Yb.....o..
00000050	3A	DE	E6	CE	28	82	58	67	4E	13	02	D0	5D	20	23	2D	:...(.XgN...) #-
00000060	B4	BE	BB	33	DB	52	E7	35	B7	90	00	AE	2C	67	81	32	...3.R.5.....g.2
00000070	8B	4D	39	74	F8	C3	2F	6C	F0	50	7A	01	78	99	DF	FF	.M9t.../1.Pz.x...

2025-07-04 15:21:18.663 读取芯片完成,耗时0.00秒

2025-07-04 15:33:54.597 查找数据不存在

2025-07-04 15:35:20.302 查找数据

2025-07-04 15:35:50.398 查找数据

2025-07-04 15:36:37.814 查找数据

2025-07-04 15:36:44.895 查找数据

2025-07-04 15:37:23.223 查找数据

2025-07-04 15:39:02.406 查找数据

2025-07-04 15:39:05.528 查找数据

2025-07-04 15:39:07.598 已查询到数据底部

2025-07-04 15:39:09.318 查找数据

点击下一个

此次查找到跳转到第二个 0x62

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	FC	FF	51	A0	99	AB	B3	10	DC	CF	4C	E5	17	64	D3	FB	..Q.....L..d..
00000010	ED	73	A4	21	BE	A9	05	B9	E9	55	0D	D8	B5	97	44	DB	.s!.....U....D.
00000020	48	A2	A1	00	FC	55	2D	D1	08	6A	51	96	1B	8A	19	50	H....U-..jQ....P
00000030	5B	62	B1	45	6F	8F	B2	B5	6C	D9	6B	C2	69	4A	05	1A	[b.Eo...l.k.iJ..
00000040	EC	33	9D	E1	59	62	A2	60	F8	A7	EF	C0	6F	20	CC	A8	.3..Y...`.....o..
00000050	3A	DE	E6	CE	28	82	58	67	4E	13	02	D0	5D	20	23	2D	...(.XgN...) #-
00000060	B4	BE	BB	33	DB	52	E7	35	B7	90	00	AE	2C	67	81	32	...3.R.5.....g.2
00000070	8B	4D	39	74	F8	C3	2F	6C	F0	50	7A	01	78	99	DF	FF	.M9t../l.Pz.x...

2025-07-04 15:33:54.597 查找数据不存在

2025-07-04 15:35:20.302 查找数据

2025-07-04 15:35:50.398 查找数据

2025-07-04 15:36:37.814 查找数据

2025-07-04 15:36:44.895 查找数据

2025-07-04 15:37:23.223 查找数据

2025-07-04 15:39:02.406 查找数据

2025-07-04 15:39:05.528 查找数据

2025-07-04 15:39:07.598 已查询到数据底部

2025-07-04 15:39:09.318 查找数据

2025-07-04 15:39:37.534 查找数据

再点击下一个

此时没有 0x62 了，提示数据已经全部查询完了，没有找到 0x62

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	FC	FF	51	A0	99	AB	B3	10	DC	CF	4C	E5	17	64	D3	FB	..Q.....L..d..
00000010	ED	73	A4	21	BE	A9	05	B9	E9	55	0D	D8	B5	97	44	DB	.s!.....U....D.
00000020	48	A2	A1	00	FC	55	2D	D1	08	6A	51	96	1B	8A	19	50	H....U-..jQ....P
00000030	5B	62	B1	45	6F	8F	B2	B5	6C	D9	6B	C2	69	4A	05	1A	[b.Eo...l.k.iJ..
00000040	EC	33	9D	E1	59	62	A2	60	F8	A7	EF	C0	6F	20	CC	A8	.3..Y...`.....o..
00000050	3A	DE	E6	CE	28	82	58	67	4E	13	02	D0	5D	20	23	2D	...(.XgN...) #-
00000060	B4	BE	BB	33	DB	52	E7	35	B7	90	00	AE	2C	67	81	32	...3.R.5.....g.2
00000070	8B	4D	39	74	F8	C3	2F	6C	F0	50	7A	01	78	99	DF	FF	.M9t../l.Pz.x...

2025-07-04 15:35:20.302 查找数据

2025-07-04 15:35:50.398 查找数据

2025-07-04 15:36:37.814 查找数据

2025-07-04 15:36:44.895 查找数据

2025-07-04 15:37:23.223 查找数据

2025-07-04 15:39:02.406 查找数据

2025-07-04 15:39:05.528 查找数据

2025-07-04 15:39:07.598 已查询到数据底部

2025-07-04 15:39:09.318 查找数据

2025-07-04 15:39:37.534 查找数据

2025-07-04 15:40:17.453 已查询到数据底部

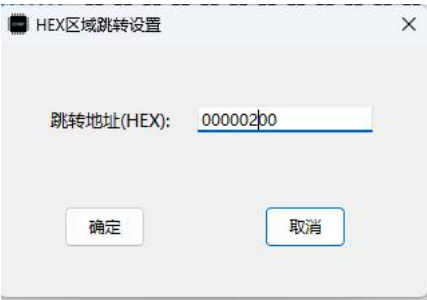
3.4.4 跳转

跳转功能可以跳转到指定的地址，比如想要查看某个地址后面的数据，就可以通过跳转指令定位到指定的地址并且在首行进行显示

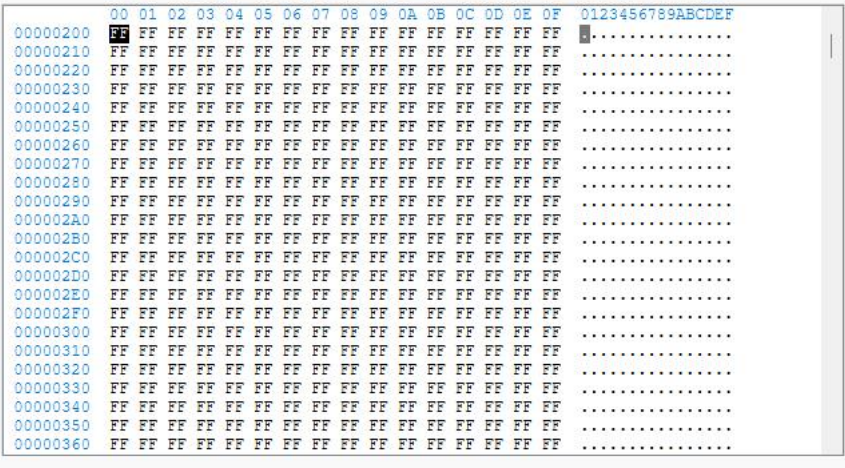
此时数据显示区域如下（首行是 0x0 地址）：



点击跳转指令，跳转到 0x200 地址处



此时数据显示区域首行显示地址为 0x200,后面显示的是 0x200 后的数据



3.5 编程操作

3.5.1 自动

自动功能是集成 ID 校验、擦除、写入、校验和读取为一体的一键快速操作按钮。其中 ID 校验、擦除、写入、校验和读取都可单独设置是否执行，在菜单栏自动选项中进行配置



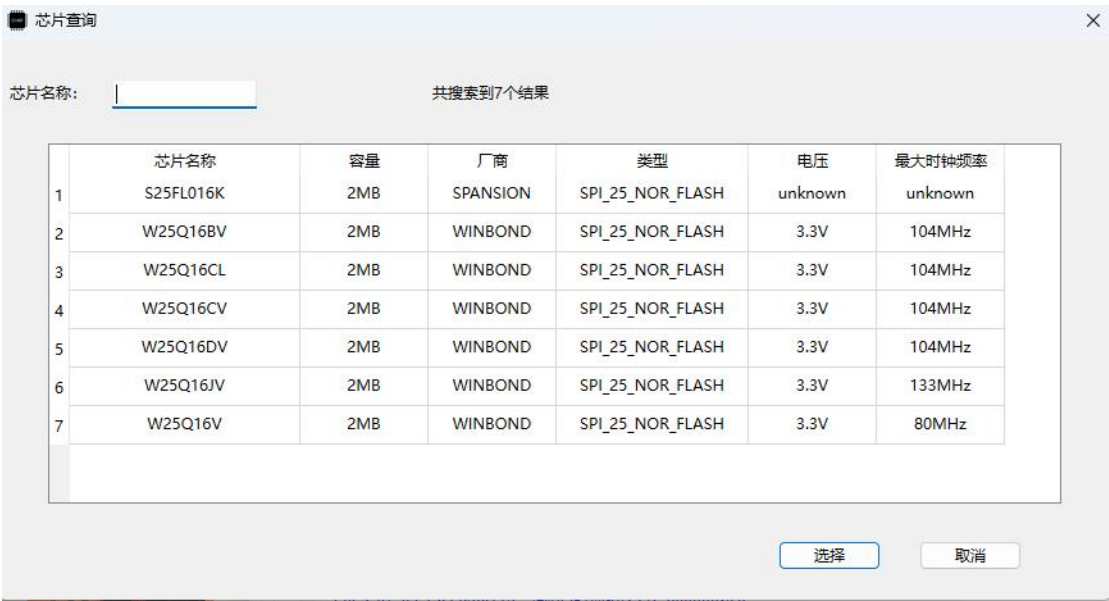
在对应选中前打钩，此项将会生效，此数据参数保存，下次重启后不用再重新设置。

需要说明的是 24xx 系列芯片不执行此功能（写入和读取直接点对应按钮即可，更简单直接）

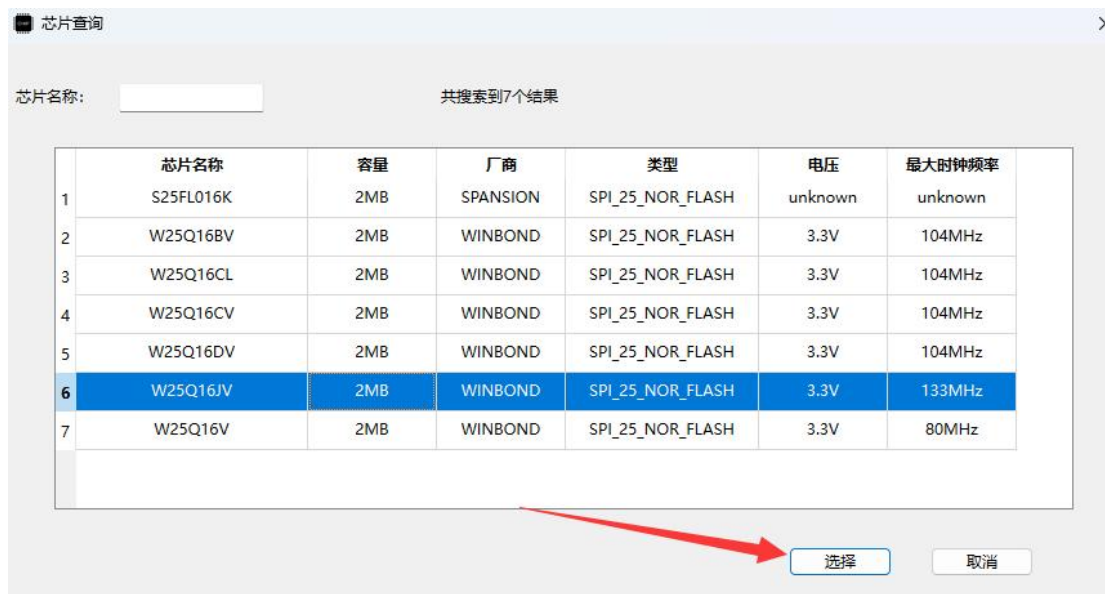
这里以 **W25Q16** 为例演示：

首先选中芯片，点击检测按钮

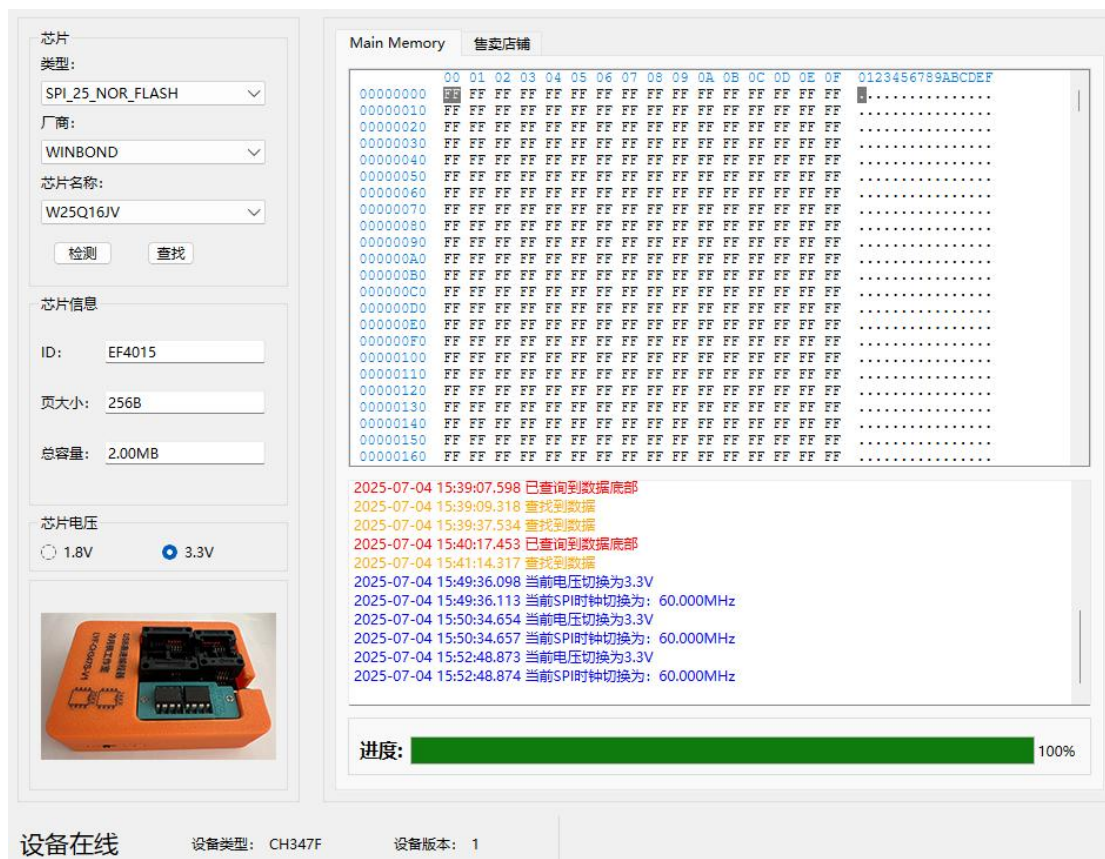
系统会根据 ID 自动查找对应芯片，然后弹出对话框让我们选择，



这里选择 W25Q16JV

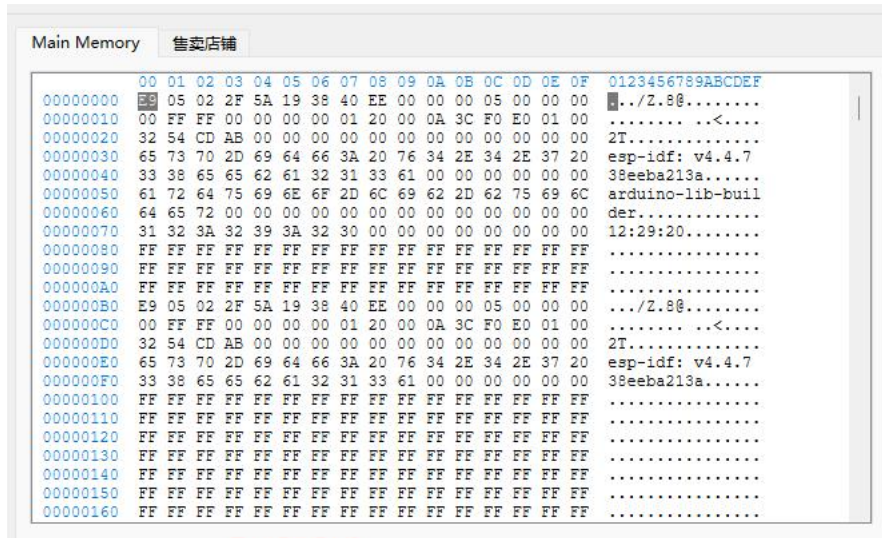


是否切换到 3.3V 电压，这里选择是，如果选择否不改变当前电压

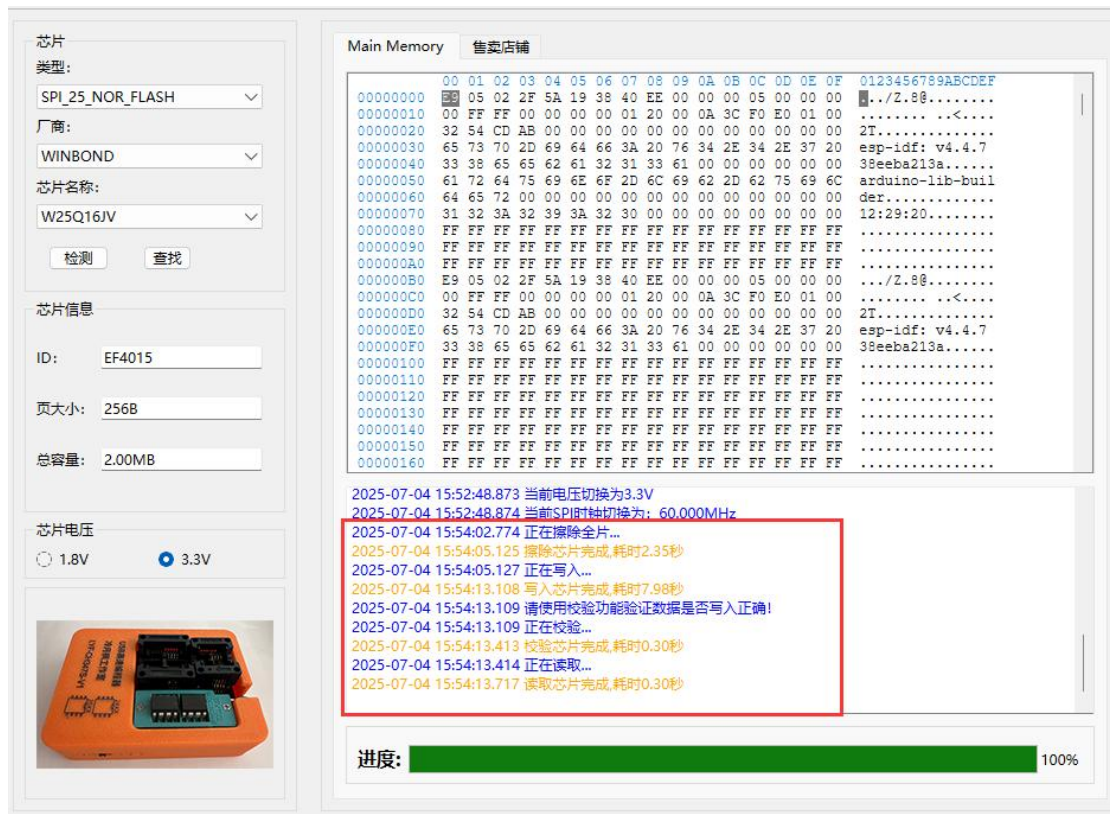


冷月枫官方网址: <https://www.lengyuefeng.com/>

然后打开要烧录的固件



点击自动按钮一键烧录



从操作日志显示区域可以看到执行操作的步骤，先擦除，然后写入，校验，最后读取。

3.5.2 查空

查空功能可以知道芯片内部是不是有数据

点击读取功能，

Main Memory

售卖店铺

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	E9	05	02	2F	5A	19	38	40	EE	00	00	00	05	00	00	00	.../Z.8@.....
00000010	00	FF	FF	00	00	00	00	01	20	00	0A	3C	F0	E0	01	00	<.....
00000020	32	54	CD	AB	00	00	00	00	00	00	00	00	00	00	00	00	2T.....<.....
00000030	65	73	70	2D	69	64	66	3A	20	76	34	2E	34	2E	37	20	esp-idf: v4.4.7
00000040	33	38	65	65	62	61	32	31	33	61	00	00	00	00	00	00	38eeba213a.....
00000050	61	72	64	75	69	6E	6F	2D	6C	69	62	2D	62	75	69	6C	arduino-lib-buil
00000060	64	65	72	00	00	00	00	00	00	00	00	00	00	00	00	00	der.....
00000070	31	32	3A	32	39	3A	32	30	00	00	00	00	00	00	00	00	12:29:20.....
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000000A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000000B0	E9	05	02	2F	5A	19	38	40	EE	00	00	00	05	00	00	00	.../Z.8@.....
000000C0	00	FF	FF	00	00	00	00	01	20	00	0A	3C	F0	E0	01	00	<.....
000000D0	32	54	CD	AB	00	00	00	00	00	00	00	00	00	00	00	00	2T.....<.....
000000E0	65	73	70	2D	69	64	66	3A	20	76	34	2E	34	2E	37	20	esp-idf: v4.4.7
000000F0	33	38	65	65	62	61	32	31	33	61	00	00	00	00	00	00	38eeba213a.....
00000100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

2025-07-04 15:54:13.413 校验芯片完成,耗时0.30秒

2025-07-04 15:54:13.414 正在读取...

2025-07-04 15:54:13.717 读取芯片完成,耗时0.30秒

2025-07-04 15:55:20.958 正在进行判空...

2025-07-04 15:55:21.260 芯片不是空片,耗时0.30秒

2025-07-04 15:55:46.470 正在读取...

2025-07-04 15:55:46.773 读取芯片完成,耗时0.30秒

2025-07-04 15:55:57.062 正在读取...

2025-07-04 15:55:57.364 读取芯片完成,耗时0.30秒

2025-07-04 15:56:14.453 正在读取...

2025-07-04 15:56:14.756 读取芯片完成,耗时0.30秒

这个是读取的数据，明显可以看到当前芯片有数据，不为空，点击查空按钮提示芯片不是空片

Main Memory

售卖店铺

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	E9	05	02	2F	5A	19	38	40	EE	00	00	00	05	00	00	00	.../Z.8@.....
00000010	00	FF	FF	00	00	00	00	01	20	00	0A	3C	F0	E0	01	00	<.....
00000020	32	54	CD	AB	00	00	00	00	00	00	00	00	00	00	00	00	2T.....<.....
00000030	65	73	70	2D	69	64	66	3A	20	76	34	2E	34	2E	37	20	esp-idf: v4.4.7
00000040	33	38	65	65	62	61	32	31	33	61	00	00	00	00	00	00	38eeba213a.....
00000050	61	72	64	75	69	6E	6F	2D	6C	69	62	2D	62	75	69	6C	arduino-lib-buil
00000060	64	65	72	00	00	00	00	00	00	00	00	00	00	00	00	00	der.....
00000070	31	32	3A	32	39	3A	32	30	00	00	00	00	00	00	00	00	12:29:20.....
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000090	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000000A0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
000000B0	E9	05	02	2F	5A	19	38	40	EE	00	00	00	05	00	00	00	.../Z.8@.....
000000C0	00	FF	FF	00	00	00	00	01	20	00	0A	3C	F0	E0	01	00	<.....
000000D0	32	54	CD	AB	00	00	00	00	00	00	00	00	00	00	00	00	2T.....<.....
000000E0	65	73	70	2D	69	64	66	3A	20	76	34	2E	34	2E	37	20	esp-idf: v4.4.7
000000F0	33	38	65	65	62	61	32	31	33	61	00	00	00	00	00	00	38eeba213a.....
00000100	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000110	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000120	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000130	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000140	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000150	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
00000160	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	

2025-07-04 15:54:13.717 读取芯片完成,耗时0.30秒

2025-07-04 15:55:20.958 正在进行判空...

2025-07-04 15:55:21.260 芯片不是空片,耗时0.30秒

2025-07-04 15:55:46.470 正在读取...

2025-07-04 15:55:46.773 读取芯片完成,耗时0.30秒

2025-07-04 15:55:57.062 正在读取...

2025-07-04 15:55:57.364 读取芯片完成,耗时0.30秒

2025-07-04 15:56:14.453 正在读取...

2025-07-04 15:56:14.756 读取芯片完成,耗时0.30秒

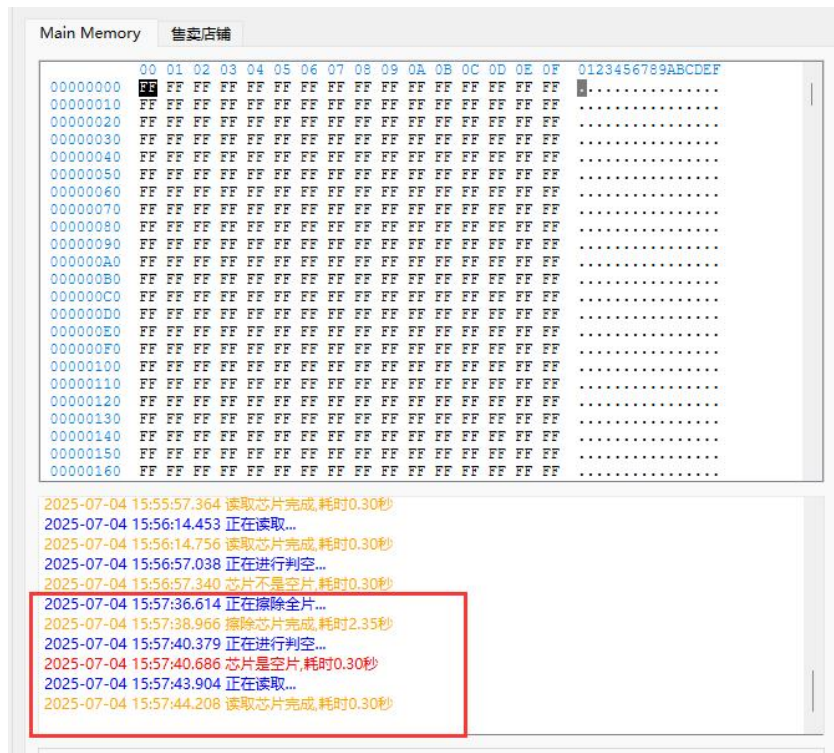
2025-07-04 15:56:57.038 正在进行判空...

2025-07-04 15:56:57.340 芯片不是空片,耗时0.30秒

进度:

100%

此时点击擦除按钮后再点击查空按钮，显示芯片是空片



3.5.3 擦除

擦除指令用来擦除芯片的全部内容，执行擦除指令后可以执行查空指令，查看芯片是否是空片，如果芯片是空片，说明擦除成功，如果芯片不为空片，说明擦除失败。

```
2025-07-04 16:01:56.358 正在擦除全片...
2025-07-04 16:01:58.708 擦除芯片完成,耗时2.35秒
```

3.5.4 写入

写入功能是将数据显示区域的数据全部写入到芯片中，写入成功后可以通过校验按钮校验数据是否真正写入完成，如果提示校验完成，表明数据确实写入成功，如果提示校验芯片失败，请重试，表明数据写入有误。

写入成功：

```
2025-07-04 16:00:24.774 正在写入...
2025-07-04 16:00:35.534 写入芯片完成,耗时10.76秒
2025-07-04 16:00:35.535 请使用校验功能验证数据是否写入正确!
2025-07-04 16:00:41.798 正在校验...
2025-07-04 16:00:42.102 校验芯片完成,耗时0.30秒
```

冷月枫官方网址: <https://www.lengyuefeng.com/>

写入失败:

```
2025-07-04 16:01:27.344 正在校验...
2025-07-04 16:01:27.649 校验芯片失败,请重试,耗时0.30秒
```

3.5.5 读出

读出功能是将芯片的数据全部读出并显示到数据显示区域,读取完成后提示读取芯片完成

```
2025-07-04 16:03:41.073 正在读取...
2025-07-04 16:03:41.378 读取芯片完成,耗时0.30秒
```

3.5.6 校验

校验功能是将芯片的数据全部读出然后和数据显示区域的数据做对比,全部一致校验成功,有一个字节不一致校验失败。

校验成功:

```
2025-07-04 16:04:39.843 正在校验...
2025-07-04 16:04:40.156 校验芯片完成,耗时0.30秒
```

校验失败:

```
2025-07-04 16:04:49.096 正在校验...
2025-07-04 16:04:49.404 校验芯片失败,请重试,耗时0.30秒
```

3.6 编程器设置

3.6.1 SPI 时钟设置

SPI 时钟设置, CH347 可以支持多种 SPI 的时钟频率,可以根据不同芯片进行选择, SPI 时钟频率最高可设置为 60MHZ,一般的芯片都支持,个别芯片不支持太高的频率,可通过此按钮进行设置。



设置成功后会在对应频率的前面显示一个小圆点。

同时也会在操作显示区域显示当前设置

```
2025-07-04 16:07:35.535 当前SPI时钟切换为: 30.000MHz
2025-07-04 16:07:39.980 当前SPI时钟切换为: 15.000MHz
2025-07-04 16:07:41.856 当前SPI时钟切换为: 3.750MHz
2025-07-04 16:07:45.858 当前SPI时钟切换为: 3.750MHz
```

3.6.2 IIC 时钟设置

CH347 可以支持多种 IIC 的时钟频率，可以根据不同芯片进行选择，IIC 时钟频率最高可设置为 1MHZ，默认设置为 400KHZ，如果个别芯片不支持，可以设置更低的时钟频率

```
2025-07-04 16:09:14.516 当前I2c时钟切换为: 1.00MHz
2025-07-04 16:09:16.984 当前I2c时钟切换为: 400.00KHz
2025-07-04 16:09:19.832 当前I2c时钟切换为: 100.00KHz
```

3.7 语言

目前只支持中文。

3.8 关于

此页面显示资料下载网址，可通过此网址下载对应资料。



冷月枫官方网址: <https://www.lengyuefeng.com/>

四、常见问题（FAQ）

Q1:编程器无法识别设备？

检查 USB 连接、驱动安装、设备管理器是否显示 CH347

Q2:烧录之后校验不成功？

可以尝试降低时钟频率后再次烧录

五、安全警告

警告：“烧录前务必备份原数据，错误操作可能导致芯片损坏！”

六、附录

官网：www.lengyuefeng.com