

## LCT200 I2C & 串口互转芯片

### 介绍

LCT200 是一款可以通过 I2C 接口通讯，拓展 2 路独立串口的通讯芯片，同时也支持通过 2 路串口读写 I2C 接口的数据。LCT200 的封装为 TSSOP-20。

### 主要功能：

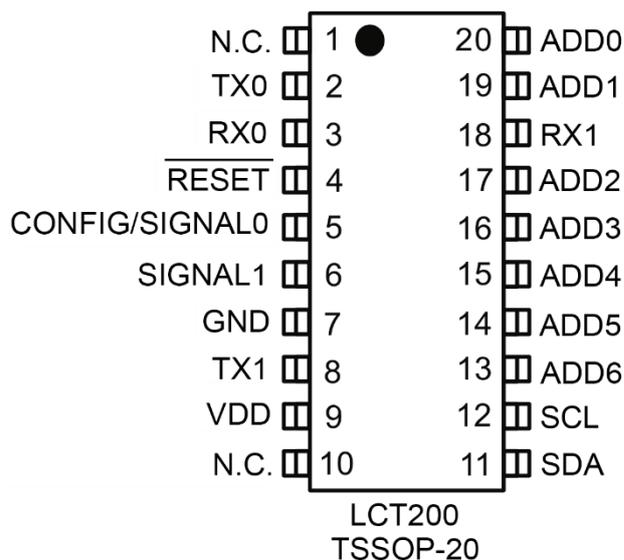
- 通过对 I2C 接口读写实现拓展 2 路独立串口功能
- 通过读写 2 路串口实现读写 I2C 端口连接芯片的数据
- 将串口接收到的数据缓存到芯片内部，等待 I2C 接口读取
- 可以配置 LCT200 的 I2C 通讯地址
- 可以在一个系统中使用多个 LCT200，配置为不同 I2C 地址即可
- 可以使转发芯片进入低功耗模式

### 应用范围：

- 需要进行串口拓展的应用
- 需要与多个串口设备通讯的应用

### 工作特性：

- LCT200 的工作电压范围是 2.4V~5.5V
- 通讯引脚耐压 5.5V
- 芯片封装为 TSSOP-20
- 低功耗状态功耗仅为 5uA
- 工作状态最大功耗约为 3mA



### 使用方法：

**场景一：主控 MCU 的串口资源已经用尽，还需要其他的串口进行收发**

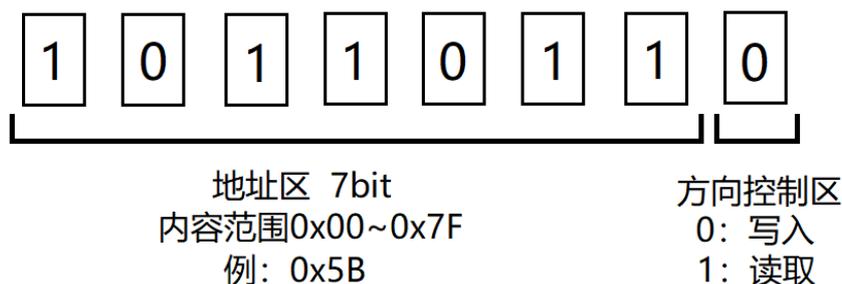
#### 硬件连接：

主控 MCU 的 I2C 引脚接 LCT200 的 I2C 引脚（11 脚、12 脚）；  
主控 MCU 的 2 个 IO 脚接接 LCT200 的 SIGNAL0 和 SIGNAL1 脚（5 脚、6 脚），用于接收信号中断和参数配置；

#### 芯片配置：

I2C 地址配置：LCT200 的 ADD0~ADD6 是 I2C 地址配置脚，是配置 LCT200 在 I2C 通讯时的 I2C 地址，ADD0 代表最低位，ADD6 代表第六位。悬空引脚代表高电平 (1)，接地代表低电平 (0)。因此，如果全部 ADD 地址脚全部悬空，代表当前 LCT200 的 I2C 地址是 0x7F；如果 ADD6 接地，代表当前 LCT200 的 I2C 地址是 0x3F；

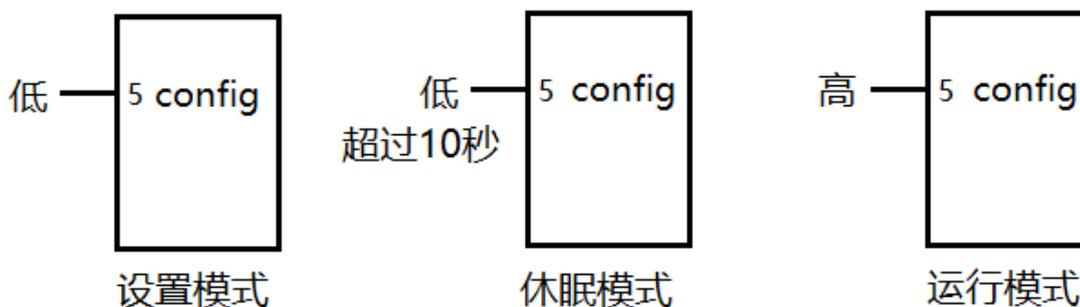
PS：这里要着重说明一下 I2C 总线地址的定义方式，I2C 总线在通讯开始信号发送后，主机在传递的第一个字节发送想要通讯的从机地址，这个地址共 8 个位(bit)组成，前 7 位是从机的 I2C 地址，最后一位为“读取”或“写入”的方向控制位。那么实际上地址的范围只有 7 位组合的 128 种结果，即 0x00~0x7F。在主机实际通讯过程中，会在地址位的最后一位追加方向控制位，组成 8 位的地址字节。本文中所有提到的有关 I2C 地址的定义，均指代前 7 位的地址位 (即 0x00~0x7F)，特此声明。



## 运行参数配置：

当 LCT200 第一次上电初始化时，需要主控 MCU 配置运行参数，配置会自动保存：

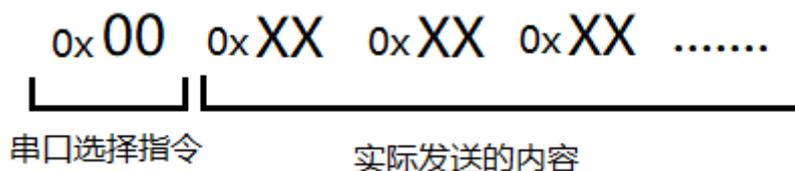
将 LCT200 的 SIGNAL0 (5 脚) 拉低 100ms 后，LCT200 进入设置模式 (最长不得超过 10s，否则 LCT200 会进入休眠模式)，通过 I2C 向 LCT200 的 I2C 地址写入配置指令 (指令内容见下文)，发送完毕后等待 100ms，使运行参数生效，拉高 SIGNAL0 (5 脚)，LCT200 进入运行模式；



## 发送数据：

当主控芯片需要通过串口 0 发送数据，则需要 LCT200 的 I2C 地址写入数据：0x00+0xXX+……+0xXX；

如果需要通过串口 1 发送数据，则需要 LCT200 的 I2C 地址写入数据：0x01+0xXX+……+0xXX；



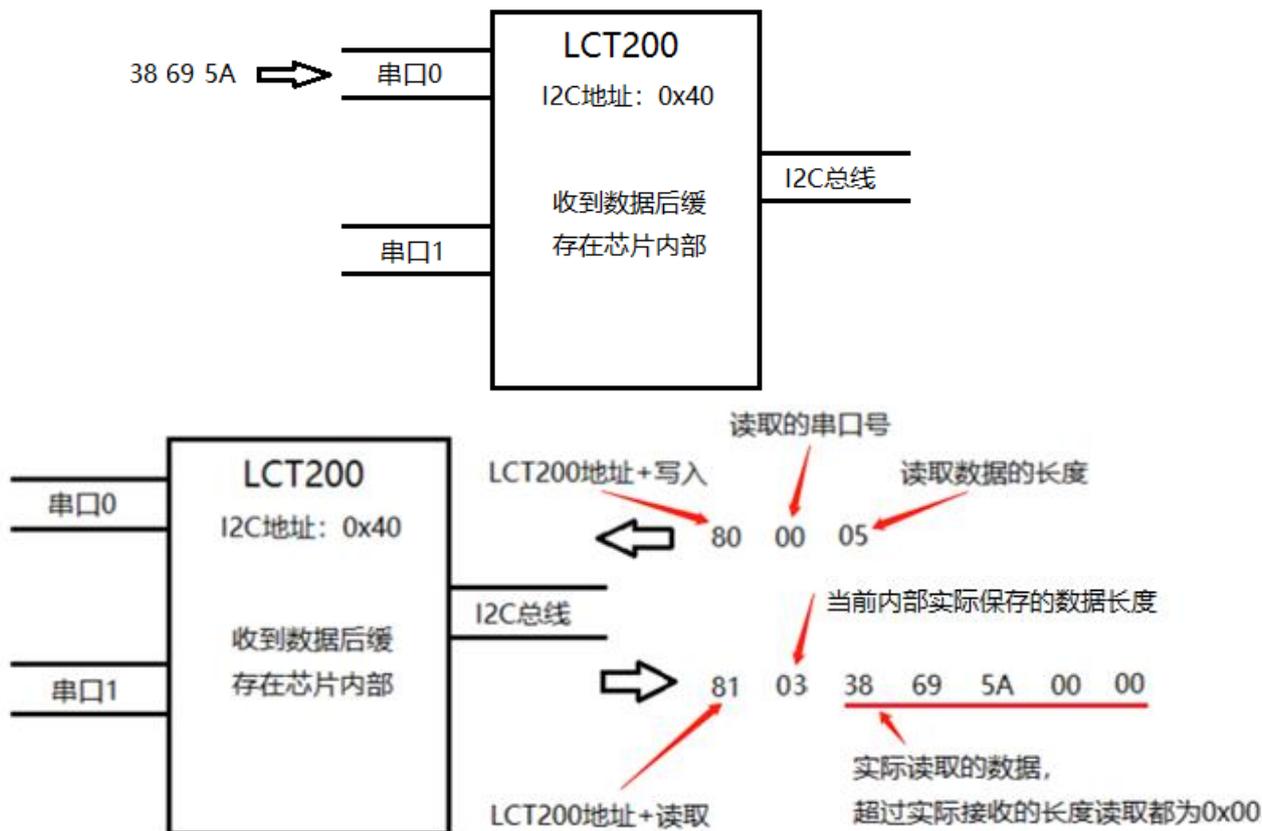
## 接收数据：

**通讯模式为交互模式：** 需要通过初始化的配置指令写入主控 MCU 的 I2C 地址，同时主控 MCU 也需要配置为 I2C 从机模式，并开启 I2C 中断用来接收数据；当 LCT200 的串口接收到一个数据包后，会通过 I2C 接口访问主控 MCU，此时 LCT200 位 I2C 主机，主控 MCU 位 I2C 从机，数据内容为：0x7F(LCT200 的 I2C 地址)，0x00(00 代表串口 0，01 代表串口 1)，0xXX……0xXX (实际收到的串口数据)。通讯结束后，LCT200 恢复为 I2C 从机模式，等待主控 MCU 控制发送数据，或者等待下一个串口数据接收完毕。



**通讯模式为读写模式：** 读写模式下，无需配置主控 MCU 的 I2C 地址，LCT200 收到串口数据后，会把数据缓存在内部的内存中，等待主控 MCU 以 I2C 读取的方式取出数据。每一路串口最多可以保存 192 字节数据，并采用先进先出的逻辑进行传递，如果没有及时取出保存在内部的数据，当 LCT200 收到的数据达到保存的最大值后，新的数据会把最先保存的数据替换掉，所以，如果串口数据量较大，用户需要及时通过 I2C 将数据读取出来，避免丢包、残包的情况。具体操作如下：当串口 0 接收完一组数据后，SIGNAL0(5 脚)会主动下拉 100ms，当主控 MCU 接收到引脚下拉后，用 I2C 接口写入一个字节 0x00，代表接下来读取串口 0 的缓存数据，然后按照用户预计的数据长度读取数据，**读取到的第一个字节代表此时该串口 0 内部已经保存的数据长度**，用户读取的长度如果小于内部保存的长度，那么没有读取的数据依然保存在 LCT200 内部，等待下一次读取。如果读取长度超过实际接收到的部分，读取到的数据为 0x00。

当串口 1 接收完一组数据后，SIGNAL1(6 脚)会主动下拉 100ms，当主控 MCU 接收到引脚下拉后，用 I2C 接口写入一个字节 0x01 后，代表接下来读取串口 1 的缓存数据，然后按照用户预计的数据长度读取数据，**读取到的第一个字节代表此时该串口 1 内部已经保存的数据长度**，用户读取的长度如果小于内部保存的长度，那么没有读取的数据依然保存在 LCT200 内部，等待下一次读取。如果读取长度超过实际接收到的部分，读取到的数据为 0x00。如果主控 MCU 的 IO 引脚资源非常紧张，也可以不接 SIGNAL1(6 脚)，主控 MCU 每隔固定时间就读取一次缓存来获取数据。这种方式适合一个数据包超过 192 字节的情况，而且这种方法可以提高一定的实时性。



## 场景二：主控需 MCU 要访问 I2C 设备，但是主控 MCU 的 I2C 接口无法使用，选择使用串口访问 I2C 设备

### 硬件连接：

主控 MCU 的串口连接 LCT200 的任意一路串口，把需要访问的 I2C 设备连接到 LCT200 的 I2C 总线上

.主控 MCU 的 1 个 IO 脚接接 LCT200 的 SIGNAL0 (5 脚)，用于参数配置；

### 芯片配置：

I2C 地址配置：如果被通讯的 I2C 设备始终处于从机模式等待被访问（例如 EEPROM），则无需配置 LCT200 的 I2C 地址

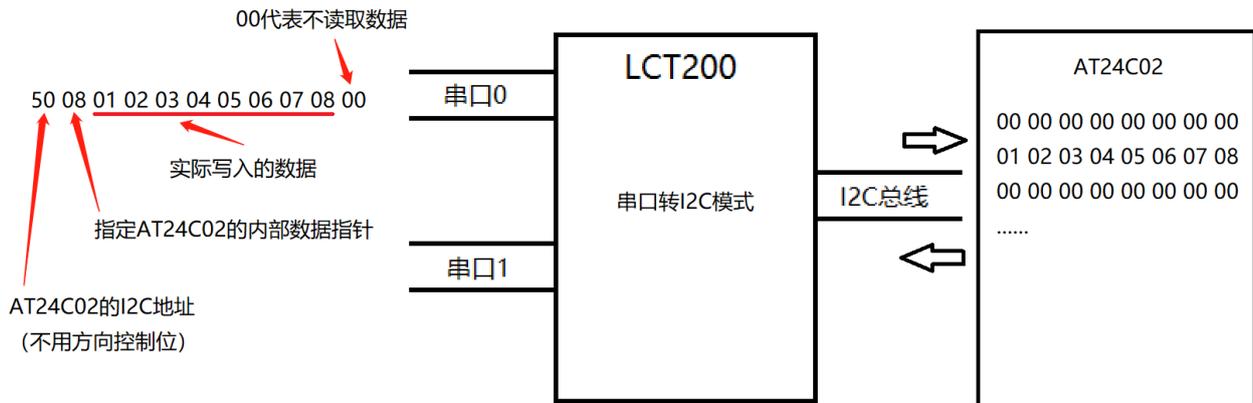
### 运行参数配置：

当主控 MCU 上电初始化时，将 LCT200 的 SIGNAL0 (5 脚) 拉低 100ms 后，通过主控 MCU 的串口向 LCT200 的任意串口发送配置指令（指令内容见下文），发送完毕后等待 100ms，拉高 SIGNAL0 (5 脚)，LCT 进入工作状态；

### 向 I2C 设备写入数据：

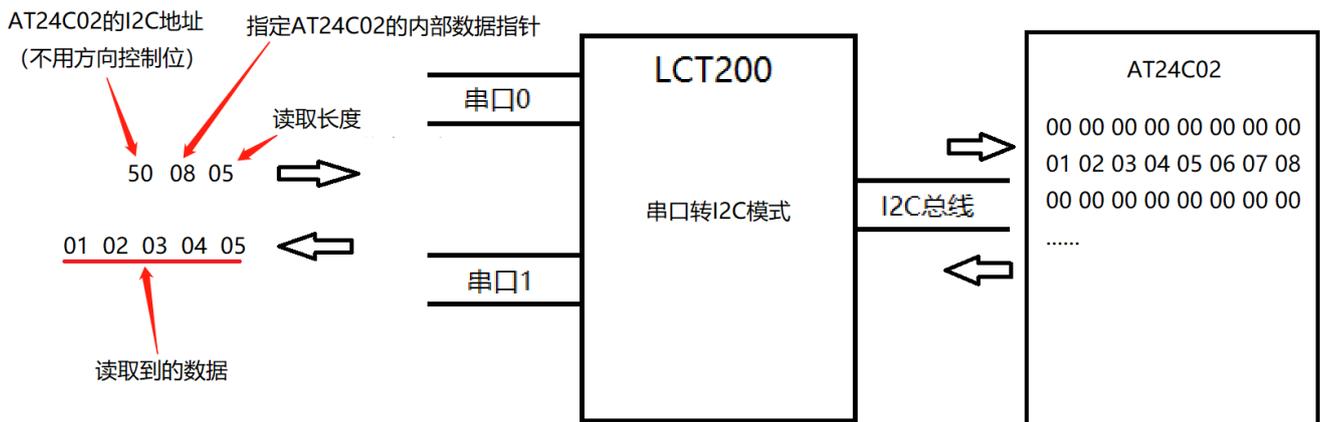
#### 1、仅写入模式：

当主控 MCU 需要向 I2C 设备仅写入数据时，需要向任意一个串口发送：  
 0xXX(目标 I2C 设备地址) + 0xXX(写入数据)+……+0xXX(最后一个数据)+0x00(读取长度 0)



## 2、写入指令后读取模式：

当主控 MCU 需要写入指令后读取指定长度的数据，需要向任意一个串口发送：  
 0xXX(目标 I2C 设备地址) + 0xXX(写入指令)+……+0xXX(最后一个指令)+0xXX(读取长度 XX)



## 3、直接读取模式：

当主控 MCU 需要仅读取指定长度的数据，需要向任意一个串口发送：  
 0xXX(目标 I2C 设备地址) + 0xXX(读取长度 XX)



## I2C 总线繁忙判断：

在使用 LCT200 的 I2C 转串口的功能时，您可能会遇到这样的情况：

当前 LCT200 运行模式为“交互模式”，此时主控 MCU 需要占用 I2C 总线，但此时 LCT200 正好同时收到外部串口数据，此时总线会出现两个“主机”设备抢占总线控制权的问题，如果处理不好，容易造成丢失数据包的问题。所以，如果在您的实际开发案例中遇到的类似的问题，LCT200 提供了 I2C 总线是否繁忙的判断功能。

在交互模式下，当 LCT200 的任意串口收到数据并接收完毕后，会在 I2C 总线空闲时下拉 CONFIG (5 脚)，直到 LCT200 向主机传递完整的数据后，松开下拉的 CONFIG (5 脚)，此时总线恢复空闲。因此在主控 MCU 需要使用 I2C 总线时，可以先判断 CONFIG (5 脚) 是否被拉低，如果没有就可以使用 I2C 总线。同时也建议主控 MCU 在使用 I2C 总线期间下拉 CONFIG (5 脚)，使用完毕后恢复总线浮空上拉，这样可以避免 LCT200 抢占总线控制权。

## 配置指令：

### 指令格式

类型	数据头	数据长度	指令 ID	指令内容	数据尾
长度	1	1	1	N	1
内容	0xFB	N	0xXX	0xXX...0xXX	0xBF

实例：0xFB 0x08 0x00 0x03 0x03 0x02 0x00 0x00 0x15 0x00 0xBF

内容：

0xFB	数据头
0x08	数据长度
0x00	指令 ID：设置运行参数
0x03	设置串口 0 波特率 9600bps
0x03	设置串口 1 波特率 9600bps
0x02	设置 I2C 发送频率 100kHz
0x00	设置工作模式为 I2C 转串口模式
0x00	设置通过 I2C 与主控 MCU 通讯模式为交互模式
0x15	设置 I2C 预设通讯的主机 MCU 地址为 0x15
0x00	设置 I2C 强制通讯模式为关闭
0xBF	数据尾

## 设置运行参数(指令 ID 0x00)

指令长度：8

指令内容：

位置	内容	注释	默认
	串口 0 波特率		
	0x00	串口 0 波特率 1200	
	0x01	串口 0 波特率 2400	
	0x02	串口 0 波特率 4800	

1	0x03	串口 0 波特率 9600	0x03
	0x04	串口 0 波特率 19200	
	0x05	串口 0 波特率 38400	
	0x06	串口 0 波特率 57600	
	0x07	串口 0 波特率 115200	
2	<b>串口 1 波特率</b>		0x03
	0x00	串口 1 波特率 1200	
	0x01	串口 1 波特率 2400	
	0x02	串口 1 波特率 4800	
	0x03	串口 1 波特率 9600	
	0x04	串口 1 波特率 19200	
	0x05	串口 1 波特率 38400	
	0x06	串口 1 波特率 57600	
	0x07	串口 1 波特率 115200	
3	0x00	I2C 发送频率: 400kbps	0x02
	0x01	I2C 发送频率: 200kbps	
	0x02	I2C 发送频率: 100kbps	
	0x03	I2C 发送频率: 50kbps	
	0x04	I2C 发送频率: 25kbps	
4	<b>I2C 与串口通讯的工作模式</b>		0x00
	0x00	I2C 转串口模式	
	0x01	串口转 I2C 模式	
5	<b>通过 I2C 与主控 MCU 通讯模式 (I2C 转串口模式有效)</b>		0x00
	0x00	<b>交互模式</b> 在此模式下, 主控芯片与转发芯片通讯时, 在总线空闲状态均为从机模式, 在通讯时, 发起通讯的一方作为主机, 另一方作为从机	
	0x01	<b>查询模式</b> 在此模式下, 转发芯片始终处于从机状态, 串口接收到的数据均会保存在芯片内部, 等待主控芯片使用查询命令读取数据; 在此模式下, 通过 I2C 转发串口数据, 也要使用查询模式的发送格式	
6	<b>I2C 预设通讯的主机 MCU 地址 (I2C 转串口模式有效)</b>		0x7F
	--	串口收到完整数据包后, 在转发给 I2C 总线上的接收设备时, 默认寻找发送的从机地址; 设置好此位后, 总线上负责接收数据的单片机也务必设置为此地址, 并开启 I2C 接收功能; 如果在 I2C 发送地址位后没有收到从机响应, 则会在串口回复错误指令, 并丢弃此次串口收到的数据包 <b>此位功能仅在通讯方式为“透明传输”时生效</b>	
	I2C 强制发送模式开关		
	0x00	<b>I2C 强制发送模式关闭</b>	

7	0x01	<b>I2C 强制发送模式开启</b> 在此模式下，当 LCT200 通过 I2C 作为主机与从机通信时，在发送地址字节或者数据字节结束后，无论从机是否响应，转发芯片都会继续把剩余数据发送完毕。在强制模式下，I2C 的通讯频率固定为 16kbps	0x00
---	------	--	------

## 设置运行参数

指令 ID 0x00

指令长度：8

### 实例 1：设置为 I2C 转串口，交互模式

主控芯片→转发芯片：0xFB 0x08 0x00 0x07 0x07 0x02 0x00 0x00 0x7F 0x00 0xBF

内容：

0xFB	数据头
0x08	数据长度
0x00	指令 ID：设置运行参数
0x07	设置串口 0 波特率 115200bps
0x07	设置串口 1 波特率 115200bps
0x02	设置 I2C 发送频率 100kHz
0x00	设置工作模式为 I2C 转串口模式
0x00	设置通过 I2C 与主控 MCU 通讯模式为交互模式
0x15	设置 I2C 预设通讯的主机 MCU 地址为 0x15
0x00	设置 I2C 强制通讯模式为关闭
0xBF	数据尾

### 实例 2：设置为 I2C 转串口，查询模式

主控芯片→转发芯片：0xFB 0x08 0x00 0x07 0x07 0x02 0x00 0x01 0x7F 0x00 0xBF

内容：

0xFB	数据头
0x08	数据长度
0x00	指令 ID：设置运行参数
0x07	设置串口 0 波特率 115200bps
0x07	设置串口 1 波特率 115200bps
0x02	设置 I2C 发送频率 100kHz
0x00	设置工作模式为 I2C 转串口模式
0x01	设置通过 I2C 与主控 MCU 通讯模式为查询模式
0x15	(无效设置)
0x00	(无效设置)
0xBF	数据尾

### 实例 3：设置为串口转 I2C

主控芯片→转发芯片： 0xFB 0x08 0x00 0x07 0x07 0x02 0x00 0x01 0x7F 0x00 0xBF

内容：

0xFB	数据头
0x08	数据长度
0x00	指令 ID：设置运行参数
0x07	设置串口 0 波特率 115200bps
0x07	设置串口 1 波特率 115200bps
0x02	设置 I2C 发送频率 100kHz
0x01	设置工作模式为串口转 I2C 模式
0x01	(无效设置)
0x15	(无效设置)
0x00	设置 I2C 强制通讯模式为关闭
0xBF	数据尾

## 读取运行参数

指令 ID 0x10

指令长度： 1

### 1、使用 I2C 接口查询

- 将 LCT200 的 SIGNAL0（5 脚）拉低 100ms
- 对当前 LCT200 的 I2C 地址读取长度为 11 字节的数据，数据结构符合设置运行参数的数据结构

### 2、使用任意串口查询

- 将 LCT200 的 SIGNAL0（5 脚）拉低 100ms
- 主控 MCU 对任意串口使用波特率 9600bps 发送数据 0xFB + 0x01 + 0x10 + 0xBF
- 该串口回复 11 字节数据包，数据结构符合设置运行参数的数据结构

## 其他功能：

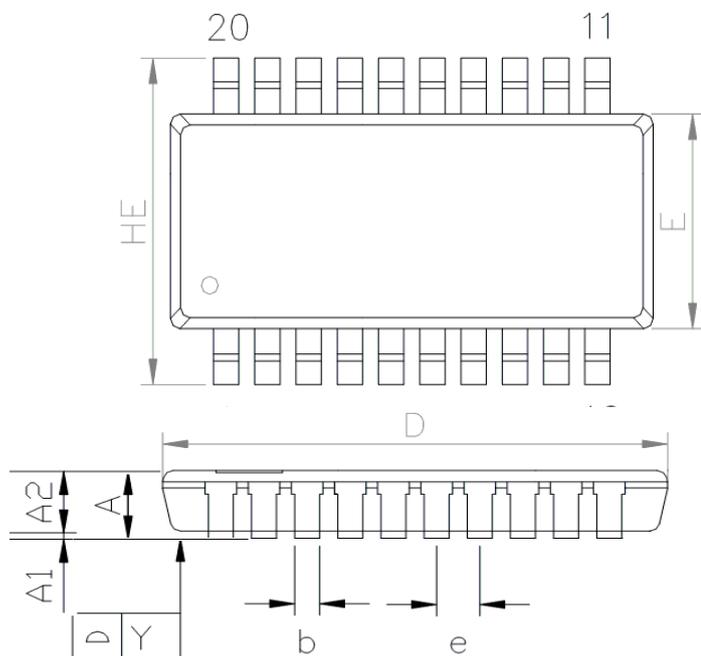
### LCT200 低功耗状态与唤醒：

LCT200 拥有低功耗状态，适合在用户需要低功耗的场景中使用；

进入休眠模式方法：LCT200 的 SIGNAL0（5 脚）拉低 10 秒，并且在这 10 秒内不要通过 I2C 或串口向 LCT200 发送任意数据，10 秒后 LCT200 自动进入低功耗状态，芯片功耗降至 5uA。

唤醒方法：将 LCT200 的 SIGNAL0（5 脚）拉高或者悬空后 100mS 就进入正常工作模式。在低功耗模式下的 LCT200 无法接收 I2C 指令或者串口数据。

## 封装信息:



SYMBOL	DIMENSION (MM)			DIMENSION (INCH)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	-	-	1,20	-	-	0,047
A1	0,05	-	0,15	0,002	-	0,006
A2	0,80	0,90	1,05	0,031	0,035	0,041
E	4,30	4,40	4,50	0,169	0,173	0,177
HE	6,40 BSC			0,252 BSC		
D	6,40	6,50	6,60	0,252	0,256	0,260
L	0,50	0,60	0,75	0,020	0,024	0,030
L1	1,00 REF			0,039 REF		
b	0,19	-	0,30	0,007	-	0,012
e	0,65 BSC			0,026 BSC		
c	0,09	-	0,20	0,004	-	0,008
$\theta$	0°	-	8°	0°	-	8°
Y	0,10 BASIC			0,004 BASIC		