

电子技术实习

——树莓派实战

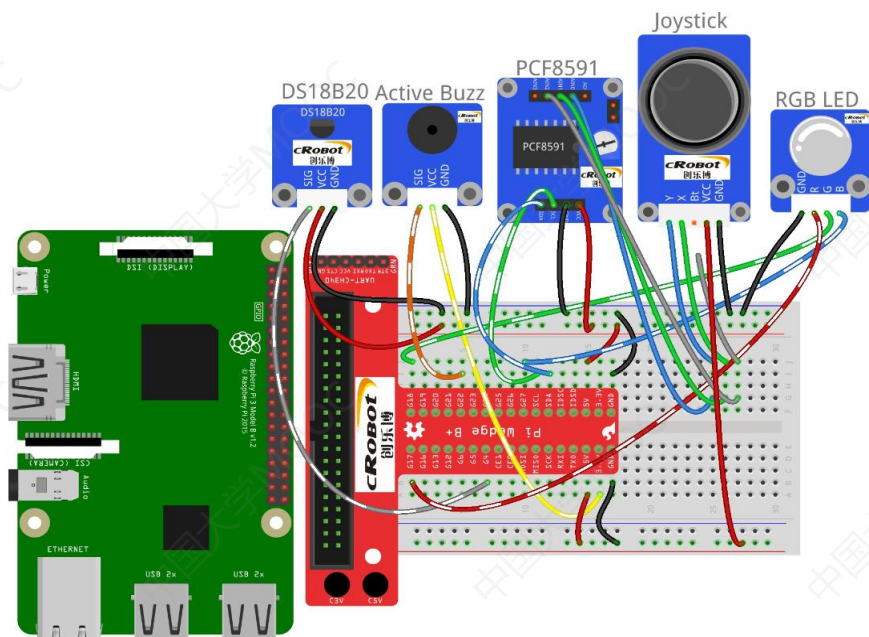
第9节 树莓派硬件



学习内容



北京科技大学
University of Science and Technology Beijing



树莓派4

嵌入式
硬件

Linux
操作命令

Python
语言编程

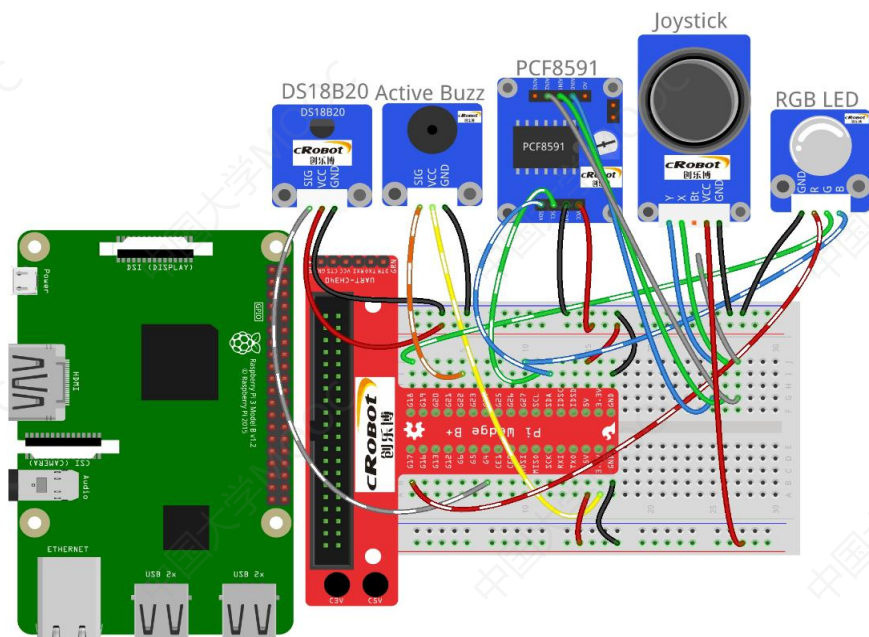
传感器
应用

人工
智能

学习内容



北京科技大学
University of Science and Technology Beijing



树莓派4

嵌入式
硬件

Linux
操作命令

Python
语言编程

传感器
应用

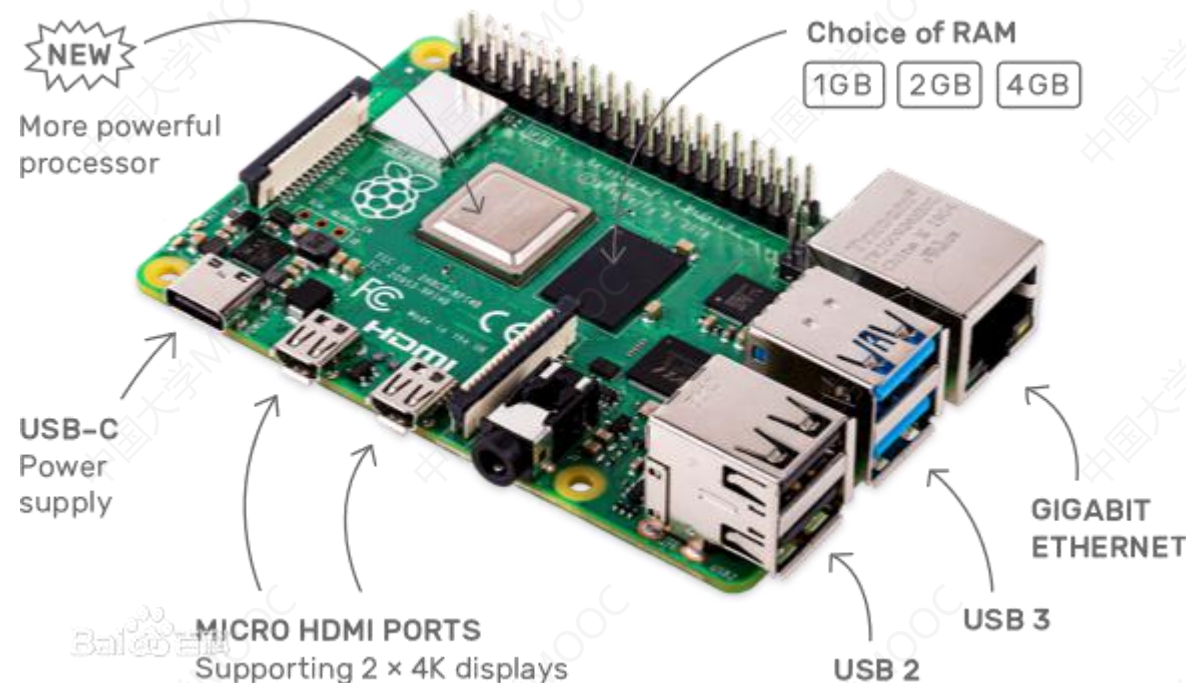
人工
智能

树莓派概述



北京科技大学
University of Science and Technology Beijing

它是一款基于ARM的微型电脑主板，以SD/MicroSD卡为内存硬盘，卡片主板周围有1/2/4个USB接口和一个10/100 以太网接口（A型没有网口），可连接键盘、鼠标和网线，同时拥有视频模拟信号的电视输出接口和HDMI高清视频输出接口，以上部件全部整合在一张仅比信用卡稍大的主板上，具备所有PC的基本功能只需接通电视机和键盘，就能执行如电子表格、文字处理、玩游戏、播放高清视频等诸多功能。Raspberry Pi B款只提供电脑板，无内存、电源、键盘、机箱或连线。



Raspberry Pi 4 Model B

树莓派历史



北京科技大学
University of Science and Technology Beijing

2012年2月29日——树莓派 (Raspberry Pi) 正式诞生！

树莓派由注册于英国的慈善组织“Raspberry Pi 基金会”开发，
Eben·Upton/埃·厄普顿为基金会创始人。

自问世以来，经历了A型、A+型、B型、B+型、2B型、3B型、3B+型、
4B型等型号的演进。

树莓派基金会提供了基于ARM的Debian和Arch Linux的发行版供大众下载。还提供支持Python作为主要编程语言，支持Java、BBC BASIC (通过 RISC OS 映像或者Linux的“Brandy Basic”克隆)、C 和Perl等编程语言。



树莓派之父
埃本·厄普顿
(Eben·Upton)



Raspberry Pi 1 Model A+



Raspberry Pi 2 Model B



Raspberry Pi 3 Model B+



Raspberry Pi 4 Model B

- | | |
|----|-------------------|
| 1 | 2012年2月树莓派B版 |
| 2 | 2013年2月红色中国版树莓派B |
| 3 | 2013年3月树莓派A |
| 4 | 2014年7月树莓派B+ |
| 5 | 2014年11月树莓派A+ |
| 6 | 2015年2月树莓派2B |
| 7 | 2015年11月树莓派zero |
| 8 | 2016年2月树莓派3B |
| 9 | 2017年11月树莓派zero w |
| 10 | 2018年3月树莓派3B+ |
| 11 | 2019年6月树莓派4B发布 |

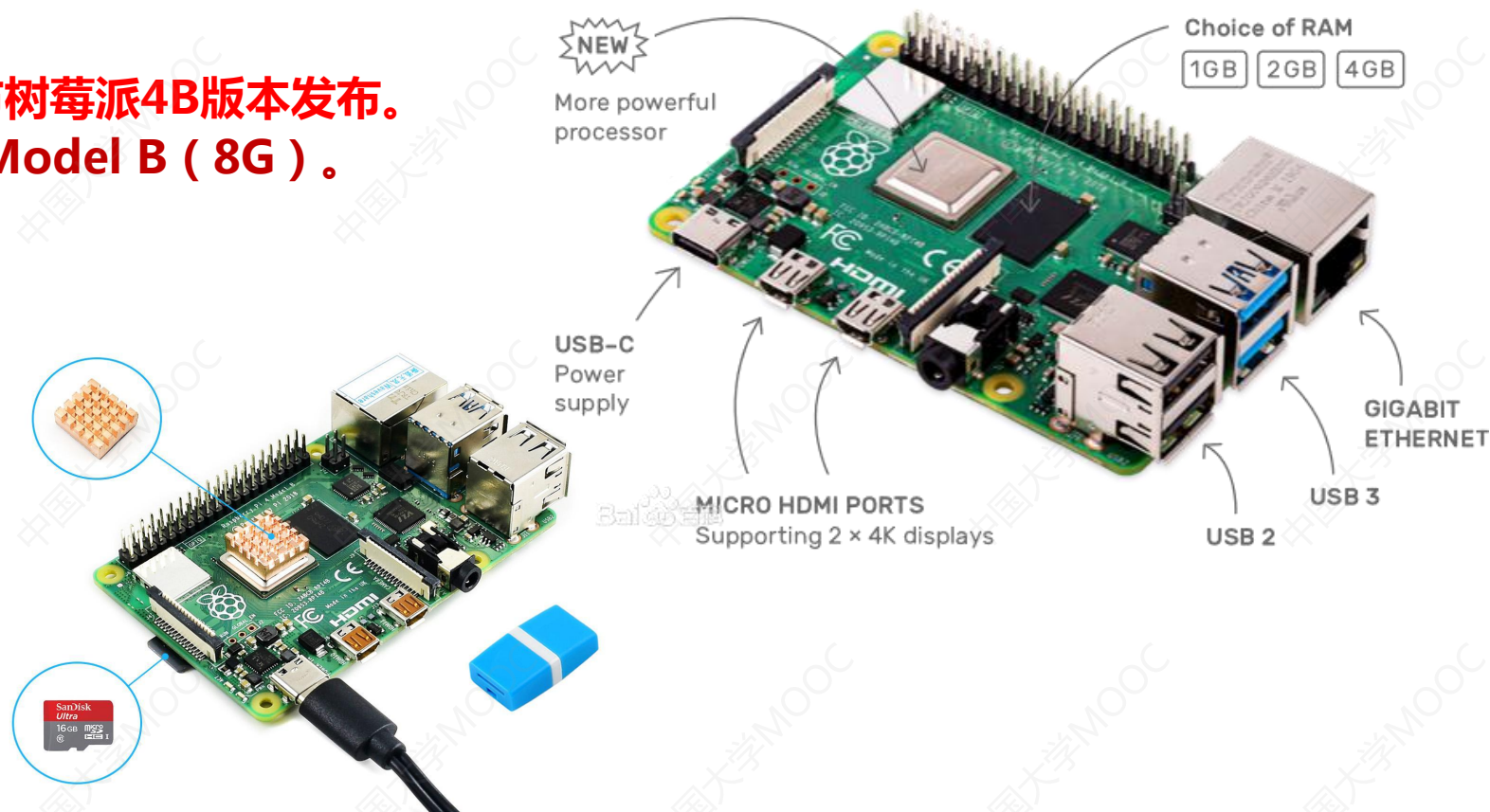
树莓派介绍 Pi 4 Model B



北京科技大学
University of Science and Technology Beijing

2019年6月25日，树莓派基金会宣布树莓派4B版本发布。
2020年5月，发布Raspberry Pi 4 Model B (8G)。

博通BCM2711处理器，四核Cortex A72架构，28nm工艺，主频1.5GHz，并提供千兆网口、蓝牙5.0、双MicroHDMI接口、耳机孔等。同时树莓派还专为其开发了基于Debian的64位专用操作系统Raspberry Pi OS。

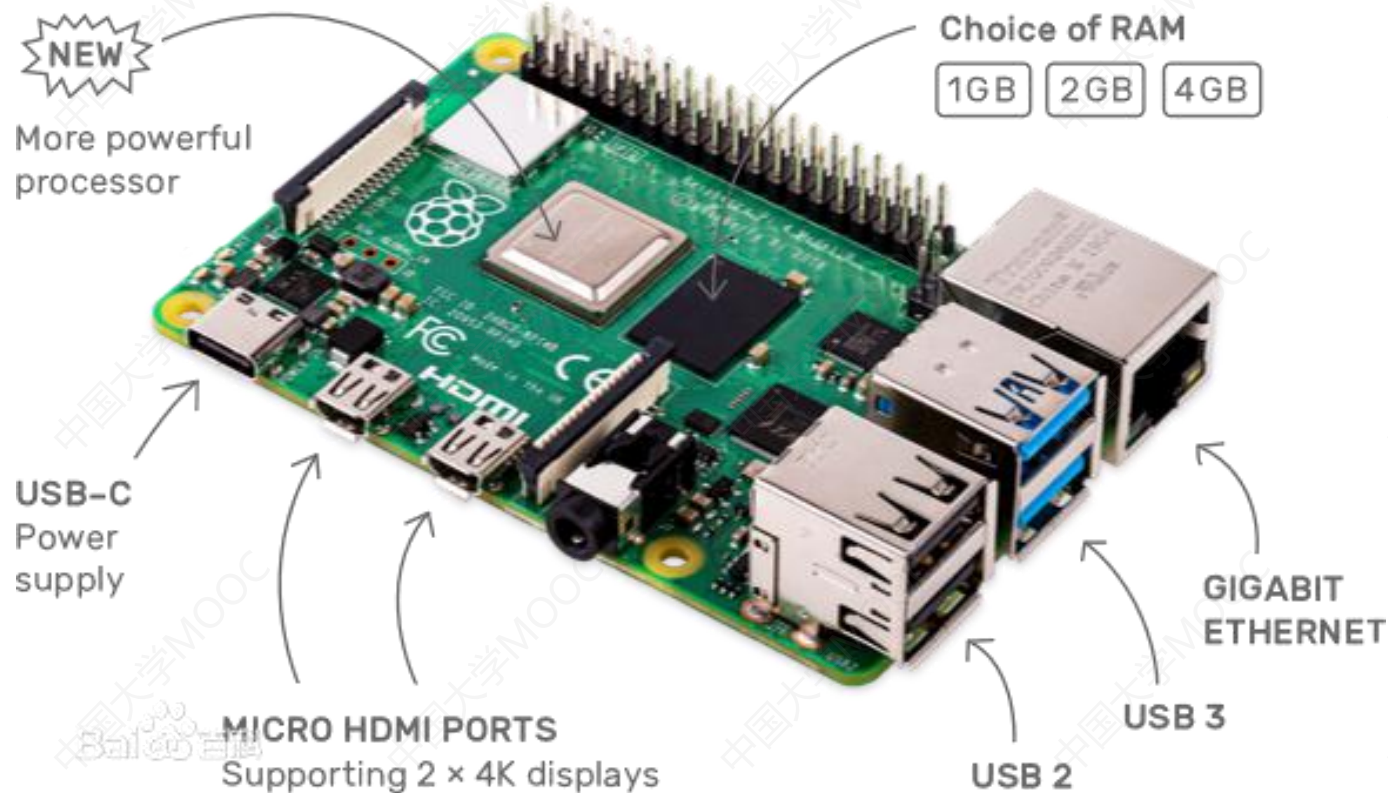
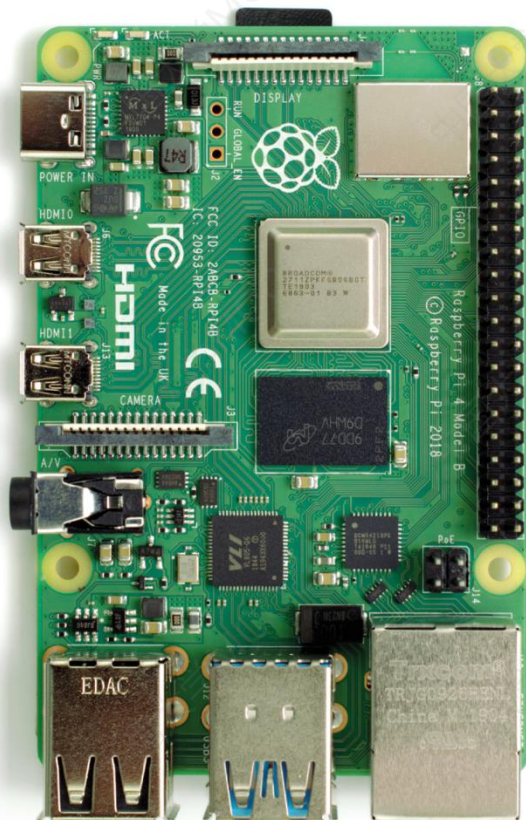


树莓派实质上是一台迷你的嵌入式计算机，就像其他任何一台台式机或者笔记本那样，利用树莓派可以做很多事情。当然，也会存在一些差别。普通的计算机主板都是依靠硬盘来存储数据，但是树莓派则使用TF卡作为“硬盘”。

树莓派 Pi 4B介绍



北京科技大学
University of Science and Technology Beijing



树莓派 Pi 4B介绍



北京科技大学
University of Science and Technology Beijing

博通BCM2711处理器

CPU : 四核Cortex A72架构

主频 : 1.5GHz

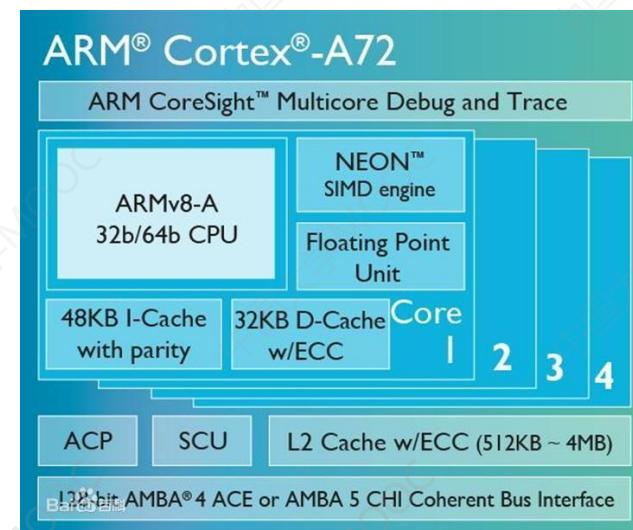
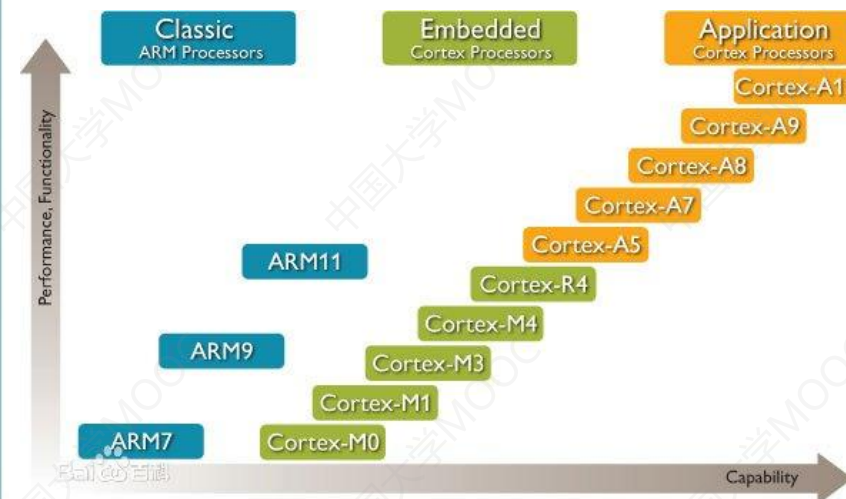
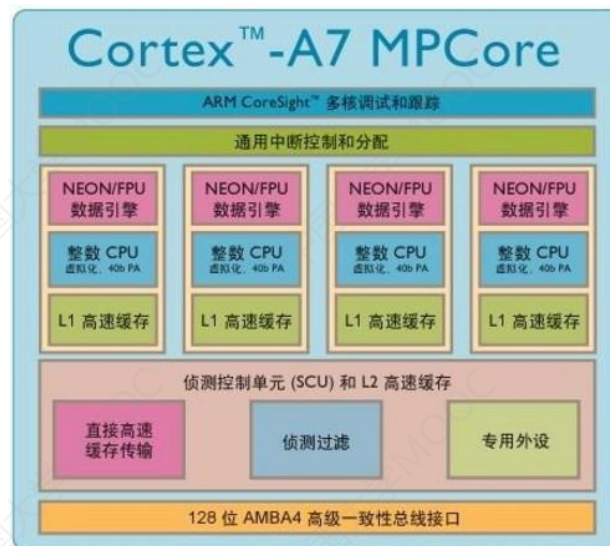
GPU : Broadcom VideoCore VI

主频 : 500MHz

工艺 : 28nm

Cortex-A72是ARM性能最出色的处理器，专为在ARM低功耗架构下要求高效能的各种设备所设计。

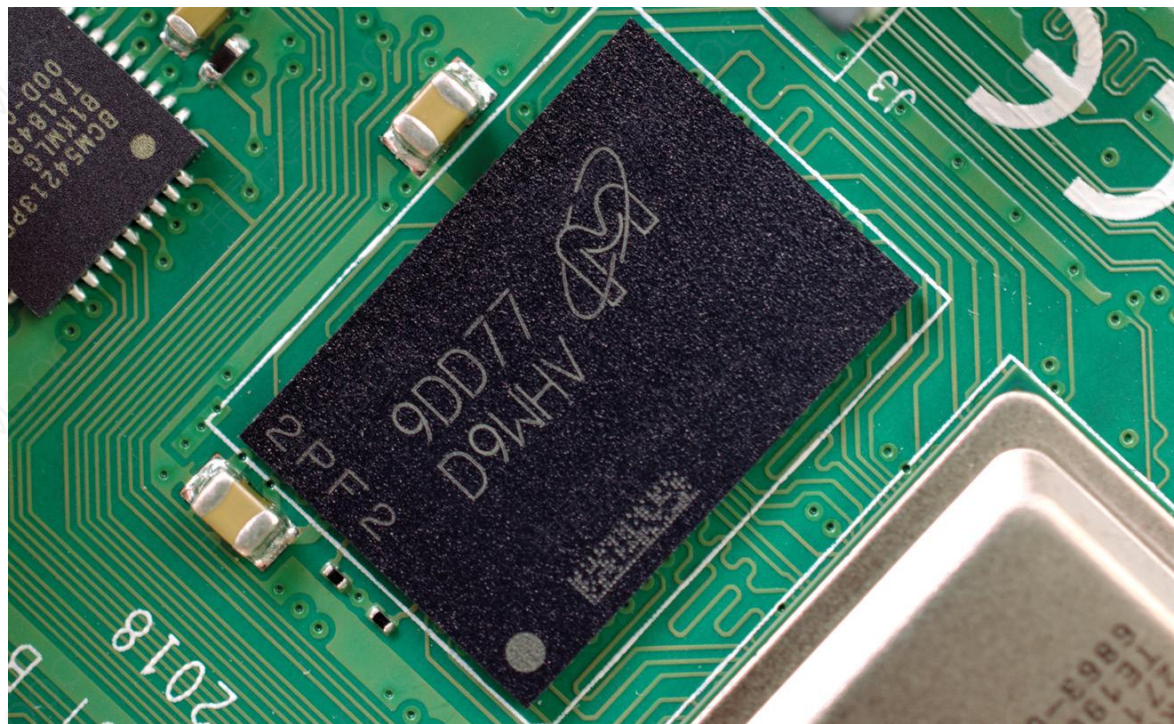
其目标应用市场包括：高端智能手机、大屏幕的移动设备、企业网络设备、服务器、无线基台、数字电视。



树莓派 Pi 4B介绍

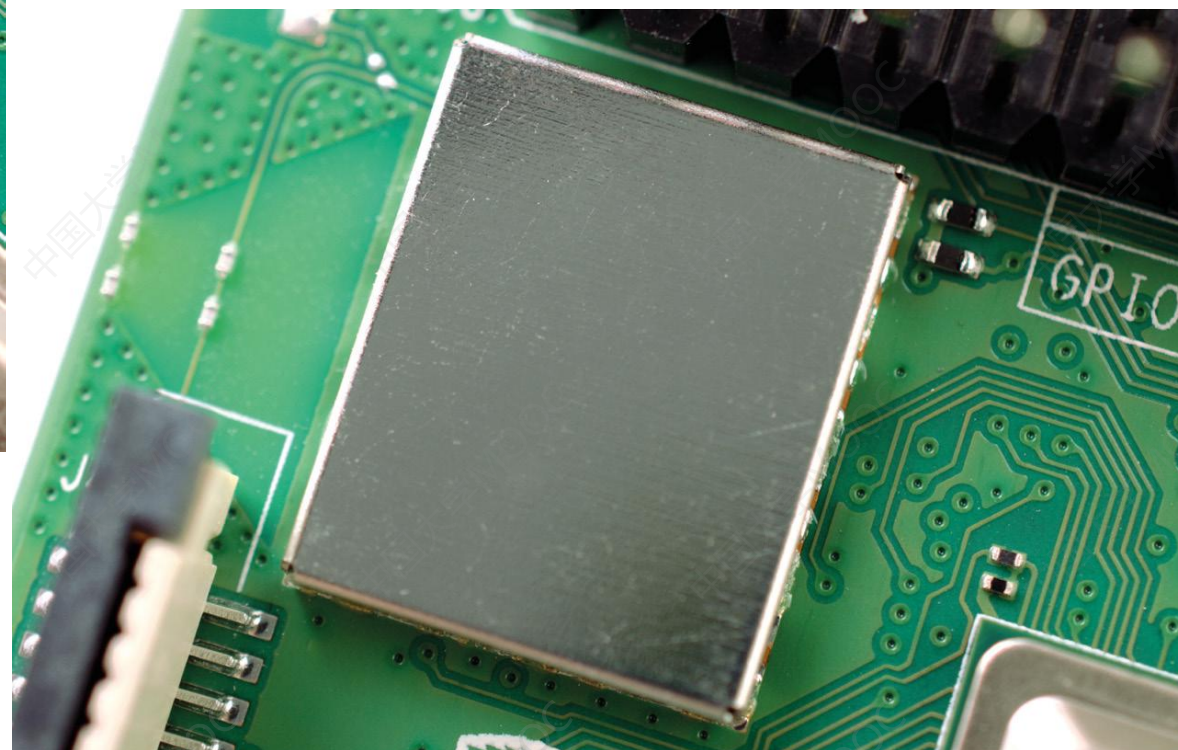


北京科技大学
University of Science and Technology Beijing



内存芯片 : 1G、2G、**4G** (DDR4)

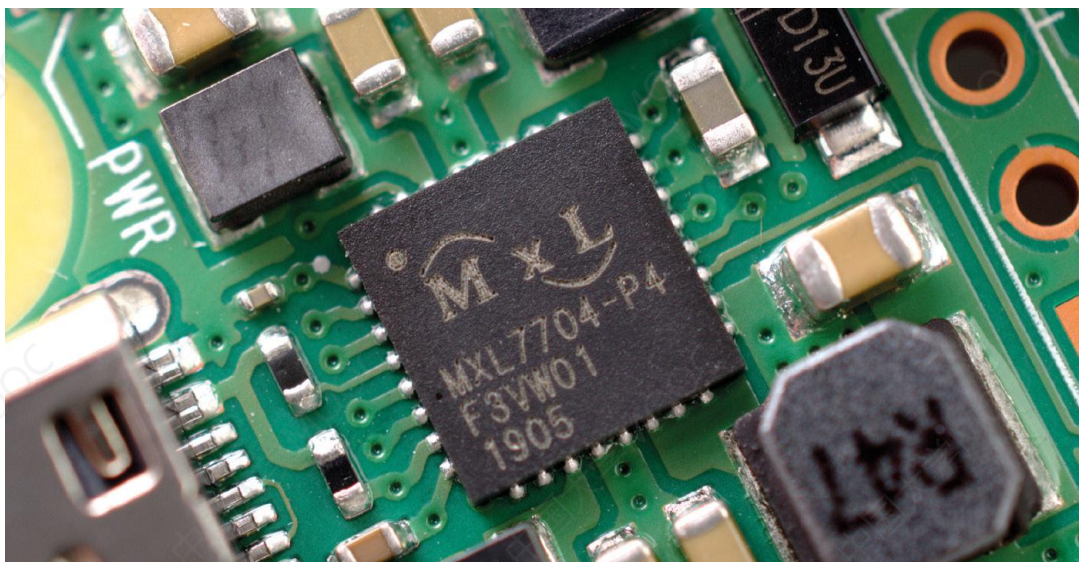
无线模块 : WiFi、蓝牙



树莓派 Pi 4B介绍

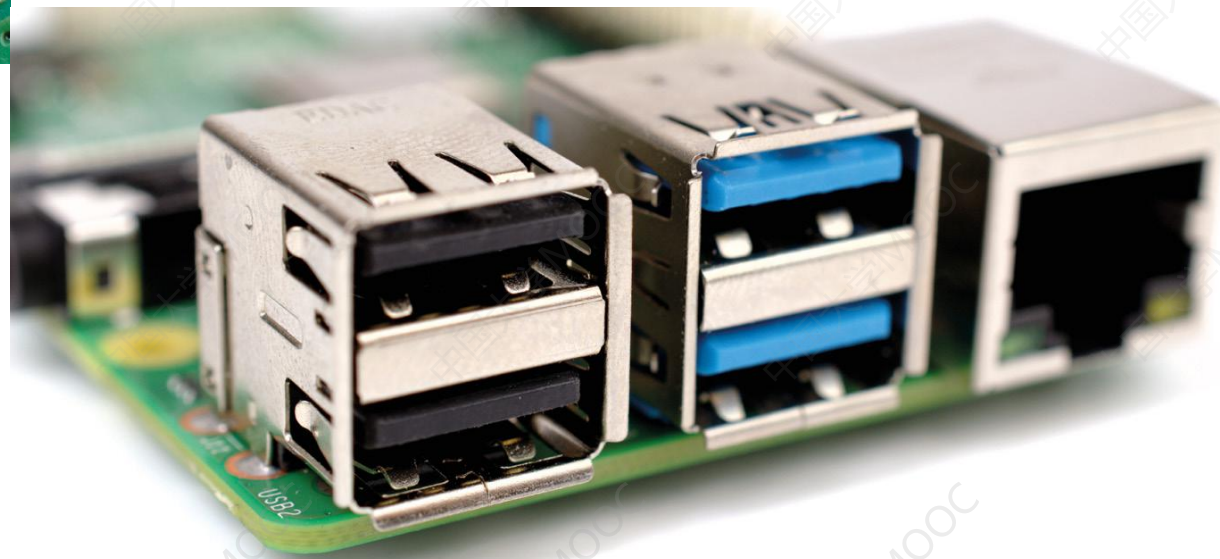


北京科技大学
University of Science and Technology Beijing



系统电源芯片、Type-C供电

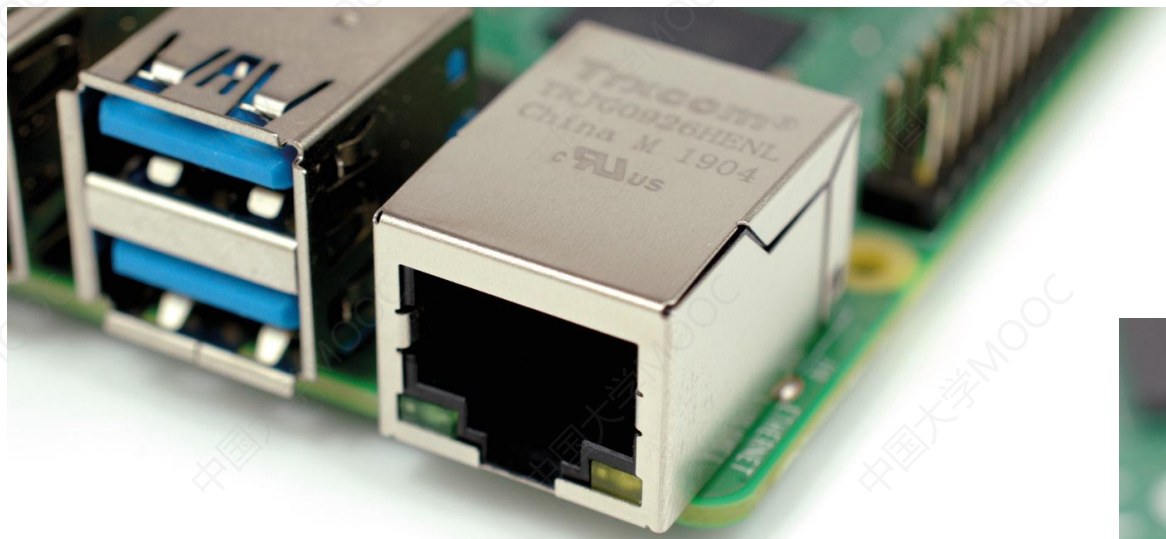
USB接口：USB2.0*2、USB3.0*2



树莓派 Pi 4B介绍

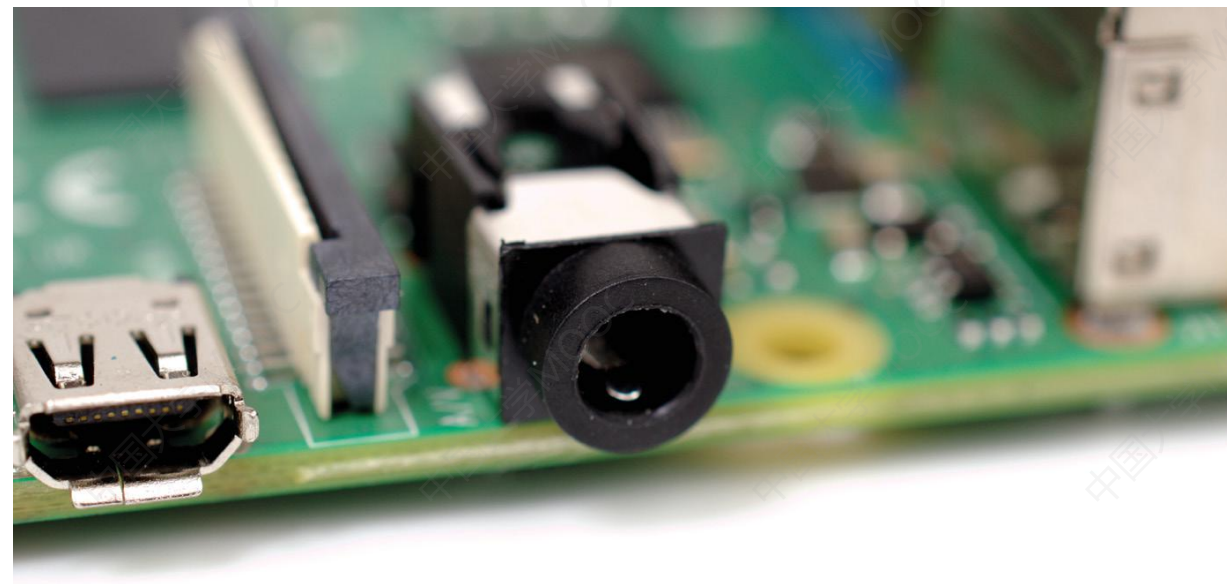


北京科技大学
University of Science and Technology Beijing



千兆以太网口

3.5mm音频接口



树莓派 Pi 4B介绍



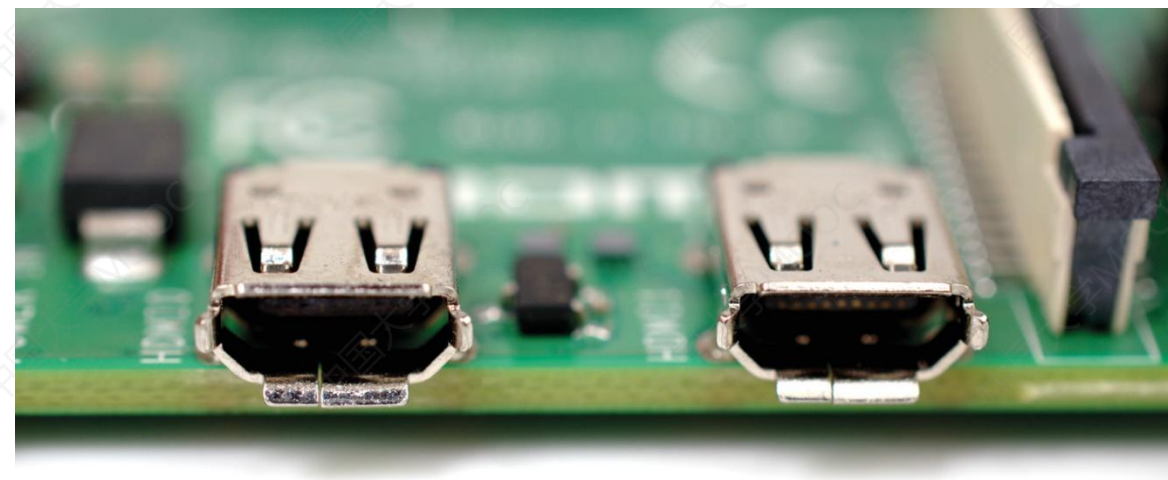
北京科技大学
University of Science and Technology Beijing



CSI摄像头接口



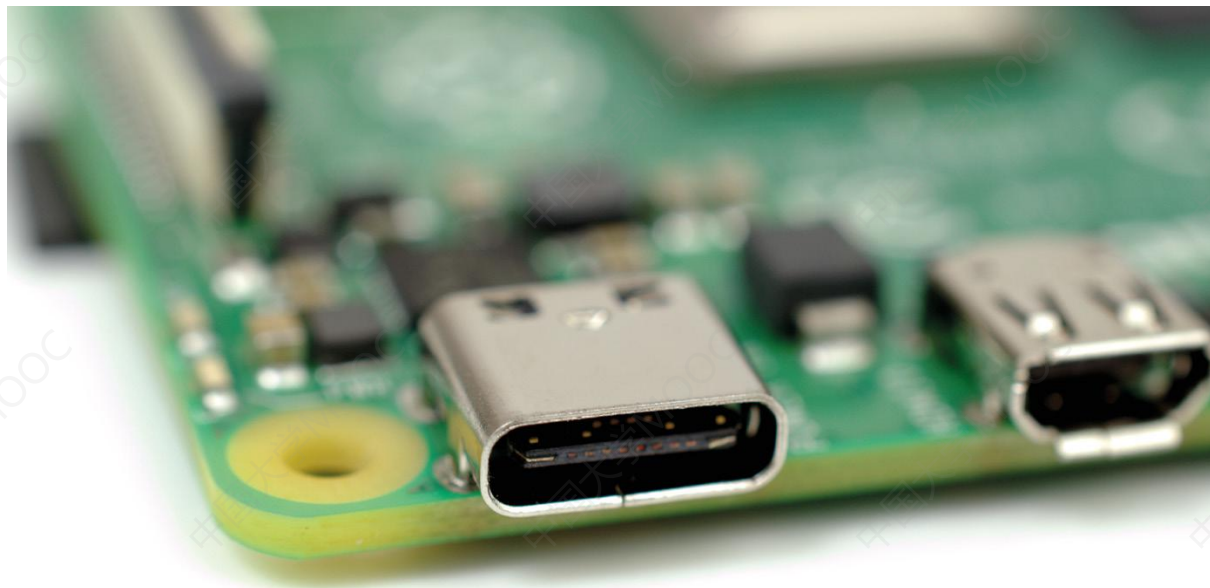
2*Micro HDMI 支持双频4K



树莓派 Pi 4B介绍

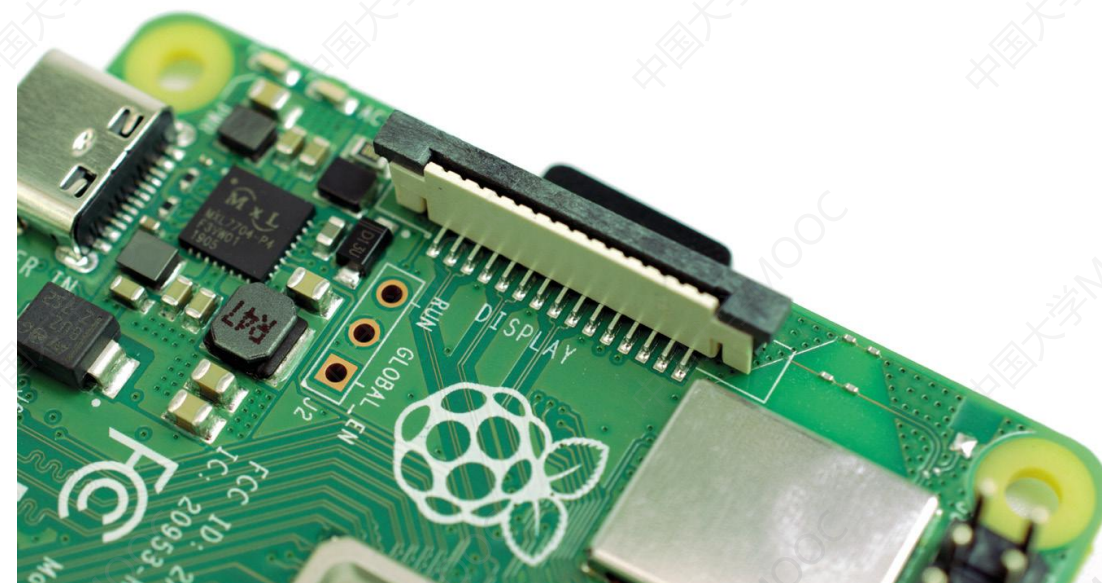


北京科技大学
University of Science and Technology Beijing



Type-C 电源接口

DSI显示屏接口



树莓派 Pi 4B介绍

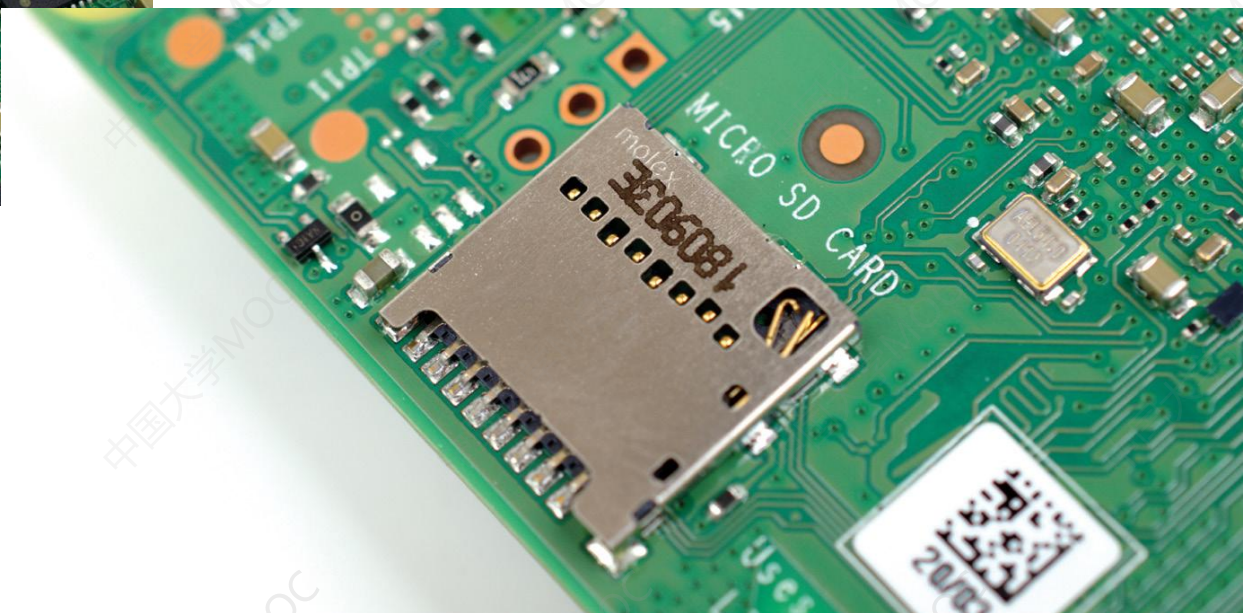


北京科技大学
University of Science and Technology Beijing



40针GPIO接口

Micro SD卡接口



树莓派 Pi 4B介绍



北京科技大学
University of Science and Technology Beijing

Pi 4B 运行需要：

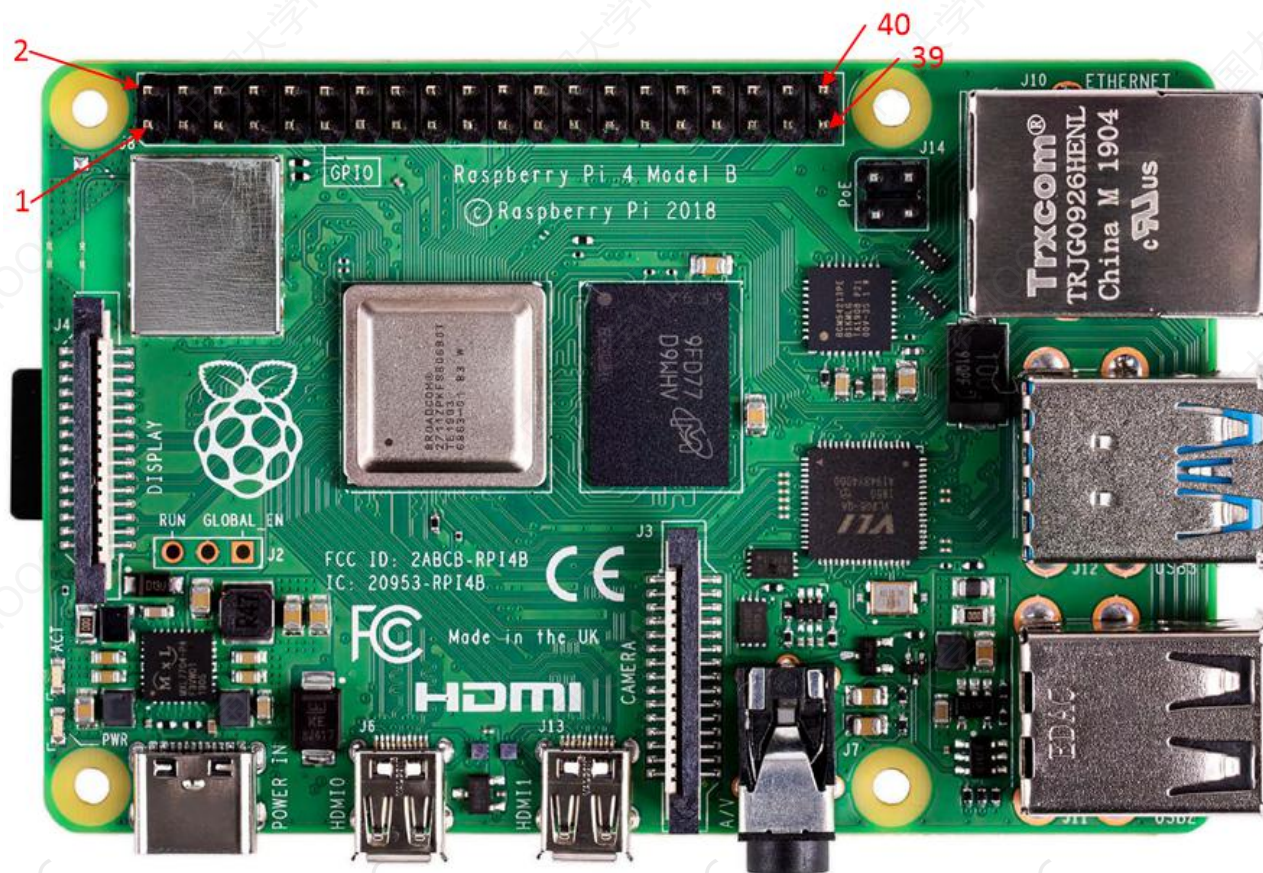
- 1、USB电源
- 2、Micro SD（带系统）
- 3、鼠标键盘（或者VNC）
- 4、Micro-HDMI线（或者VNC）



树莓派 Pi 4B介绍 - GPIO



北京科技大学
University of Science and Technology Beijing



树莓派 40Pin 引脚对照表

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

树莓派 Pi 4B介绍 - GPIO



北京科技大学
University of Science and Technology Beijing

GPIO基本和扩展功能 (BOARD编址)

板载接口编号 : GPIO

- 1、通用IO : 所有的IO都可以作为GPIO使用
- 2、串口 : 8、10
- 3、SPI口 : 19、21、23、24、26
- 4、I2C口 : 3、5、27、28
- 5、1-Wire接口 : 所有GPIO可以成单总线使用。

wiringPi编码中 7

树莓派 40Pin 引脚对照表

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

树莓派 Pi 4B介绍 - GPIO



GPIO基本功能：

通用IO：GPIO 通用型之输入输出的简称，其接脚可以供使用者由程控自由使用，PIN脚依现实考量可作为通用输入（GPI）或通用输出（GPO）或通用输入与输出（GPIO）。一个引脚可以用于输入、输出或其他特殊功能，有专用寄存器用来选择这些功能。

- 1、输入，可以通过读取某个寄存器来确定引脚电位的高低；
- 2、输出，可以通过写入某个寄存器来让这个引脚输出高电位或者低电位；
- 3、其他特殊功能，另外的寄存器来控制它们。

树莓派 40Pin 引脚对照表

wiringPi 编码	BCM 编码	功能名	物理引脚 BOARD编码		功能名	BCM 编码	wiringPi 编码
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

树莓派 Pi 4B介绍 - GPIO



北京科技大学
University of Science and Technology Beijing

串口：(BOARD编址)

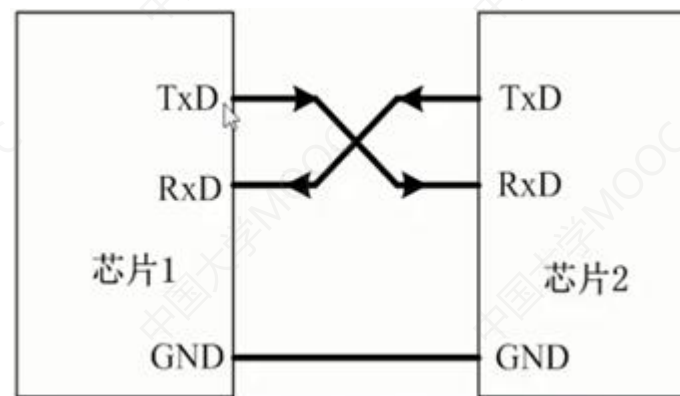
端口：8 (TXD)、10 (RXD)、6 (GND)

串行接口 (Serial Interface) 是指数据一位一位地顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信（可以直接利用电话线作为传输线），从而大大降低了成本，特别适用于远距离通信，但传送速度较慢。

串行通讯的距离可以从几米到几千米；根据信息的传送方向，串行通讯可以进一步分为单工、半双工和全双工三种。

同步串行接口和异步串行接口两大类

最常用的异步串行UART包含TTL电平的串口和232电平的串口。TTL电平是3.3V或者5V的，而232是负逻辑电平，它定义+5~+12V为0逻辑电平，而-12~-5V为1逻辑电平。

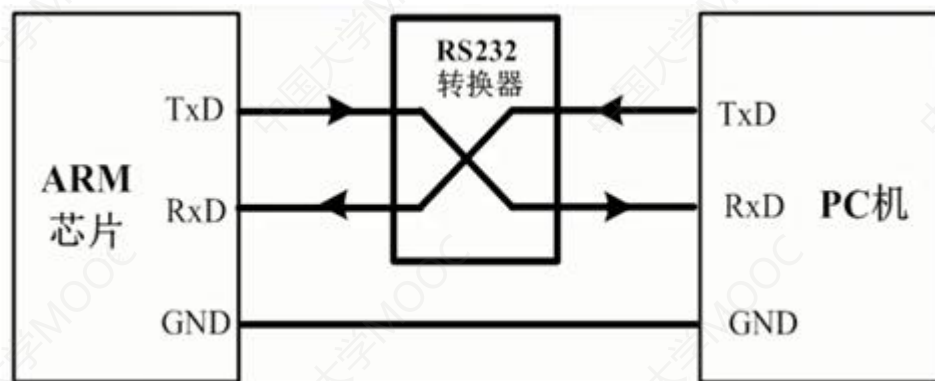


典型的TTL电平连接

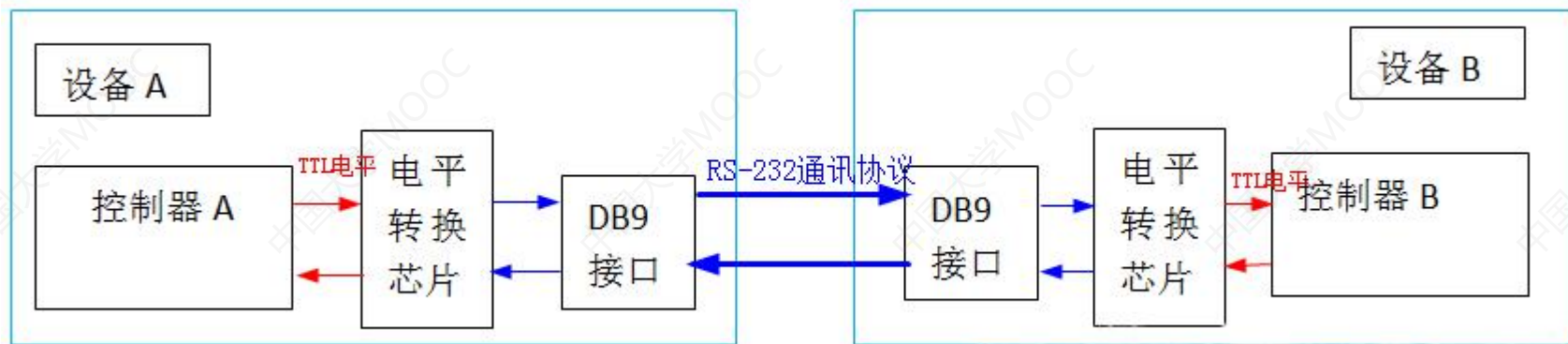
树莓派 Pi 4B介绍 - 串行通信



北京科技大学
University of Science and Technology Beijing



典型的嵌入式与PC串行通讯连接

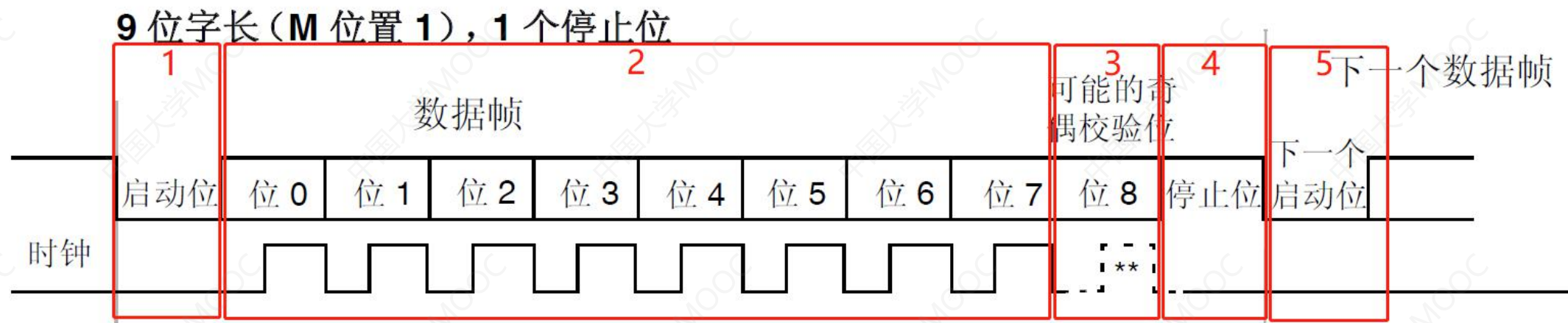


RS232电平的设备间串行通讯连接

树莓派 Pi 4B介绍 - 串行通信



北京科技大学
University of Science and Technology Beijing



1) 起始位

当未有数据发送时，数据线处于逻辑“1”状态；先发出一个逻辑“0”信号，表示开始传输字符。

2) 数据位

紧接着起始位之后。资料位的个数可以是4、5、6、7、8等，构成一个字符。通常采用ASCII码。从最低位开始传送，靠时钟定位。

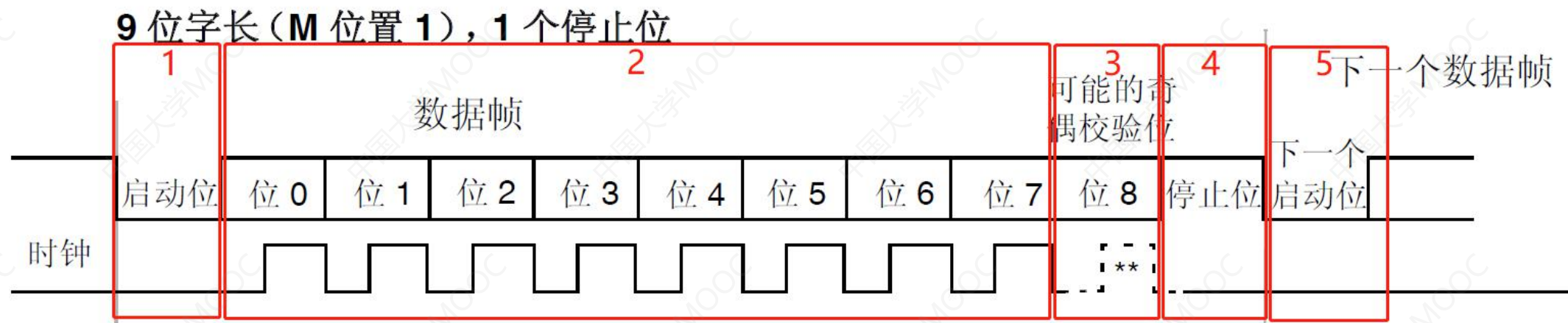
3) 奇偶校验位

资料为加上这一位后，使得“1”的位数应为偶数（偶校验）或奇数（奇校验），以此来校验资料传送的正确性。

树莓派 Pi 4B介绍 - 串行通信



北京科技大学
University of Science and Technology Beijing



4) 停止位

它是一个字符数据的结束标志。可以是1位、1.5位、2位的高电平。由于数据是在传输线上定时的，并且每一个设备有其自己的时钟，很可能在通信中两台设备间出现了小小的不同步。因此停止位不仅仅是表示传输的结束，并且提供计算机校正时钟同步的机会。适用于停止位的位数越多，不同时钟同步的容忍程度越大，但是数据传输率同时也越慢。

5) 空闲位或起始位

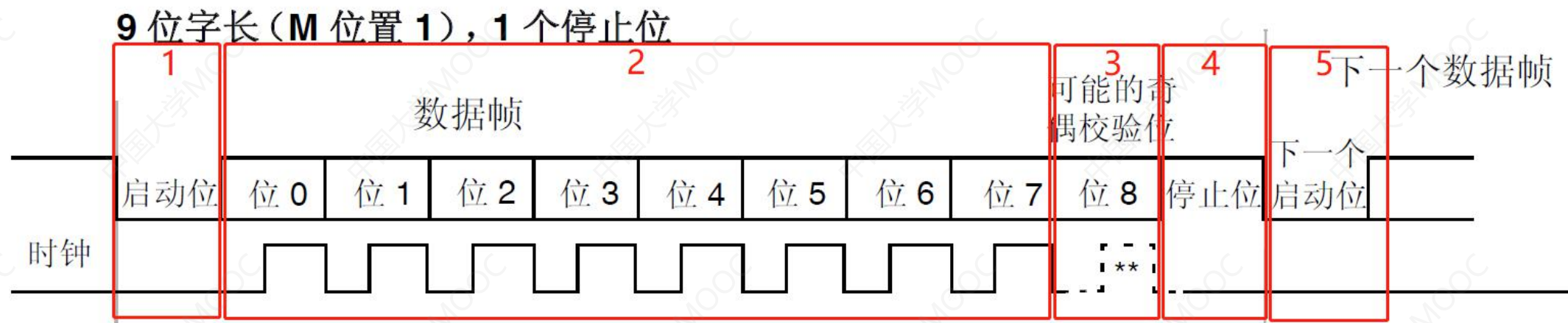
处于逻辑“1”状态，表示当前线路上没有资料传送，进入空闲状态。

处于逻辑“0”状态，表示开始传送下一数据段。

树莓派 Pi 4B介绍 - 串行通信



北京科技大学
University of Science and Technology Beijing



6) 波特率

表示每秒钟传送的码元符号的个数，是衡量数据传送速率的指标，它用单位时间内载波调制状态改变的次数来表示。

常用的波特率有：9600、115200.....

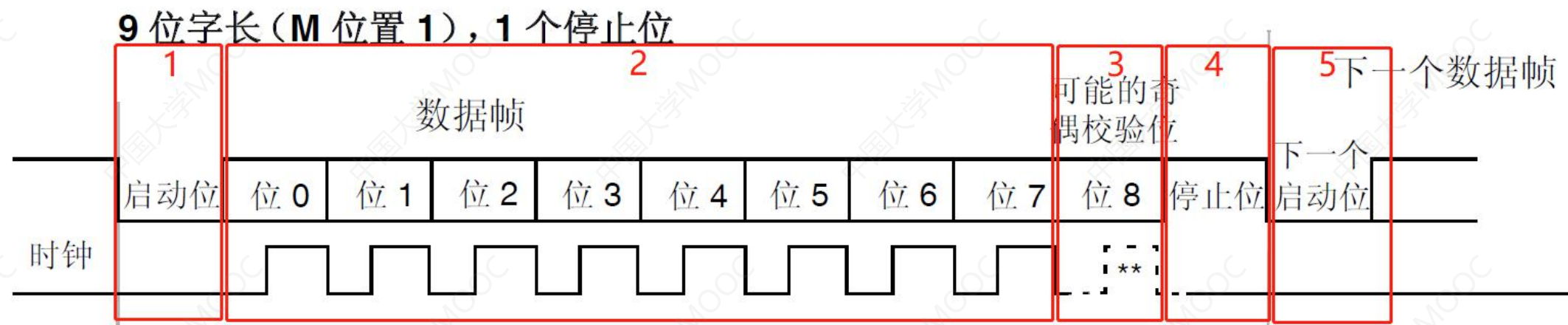
时间间隔计算：1秒除以波特率得出的时间

例如，波特率为9600的时间间隔为 $1s / 9600$ (波特率) = 104us。

树莓派 Pi 4B介绍 - 串行通信



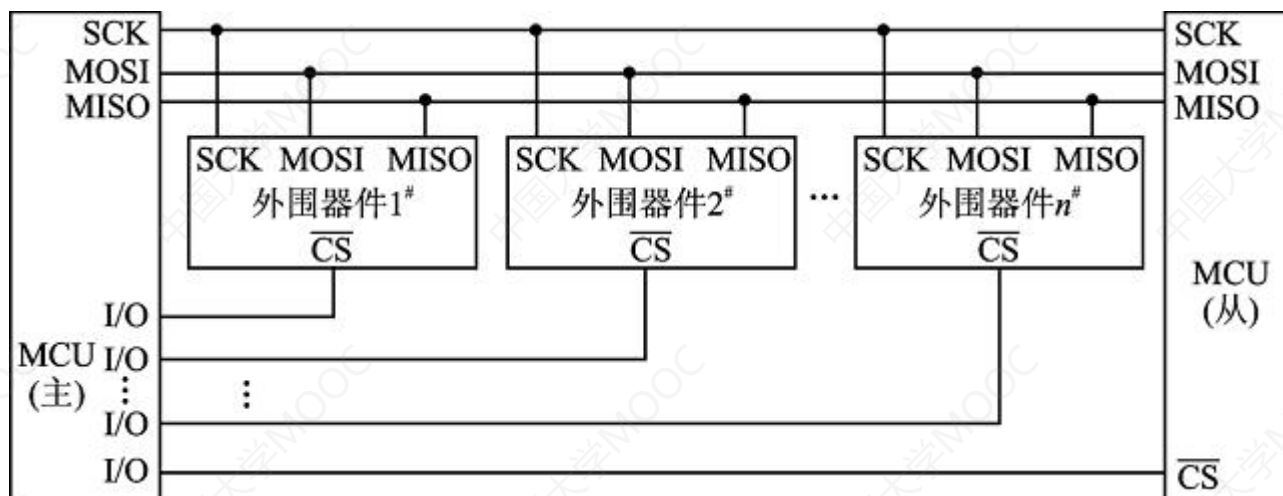
北京科技大学
University of Science and Technology Beijing



7) 通信设置参数

- 1、起始位：一般默认1位
- 2、数据位：可以设置为4、5、6、7、8。一般默认常用的是8位（1字节）
- 3、奇偶校验：可以设置为奇、偶、无校验位。常用无校验位。
- 4、停止位：可以设置为1、1.5、2位。一般常用1位。
- 5、波特率：可以设置为 1200、2400、4800、9600、...、115200。常用9600和115200。

树莓派 Pi 4B介绍 - SPI



SPI是串行外设接口(Serial Peripheral Interface)的缩写。是 Motorola 公司推出的一种同步串行接口技术，是一种高速的，全双工，同步的通信总线。

SPI通信特点：

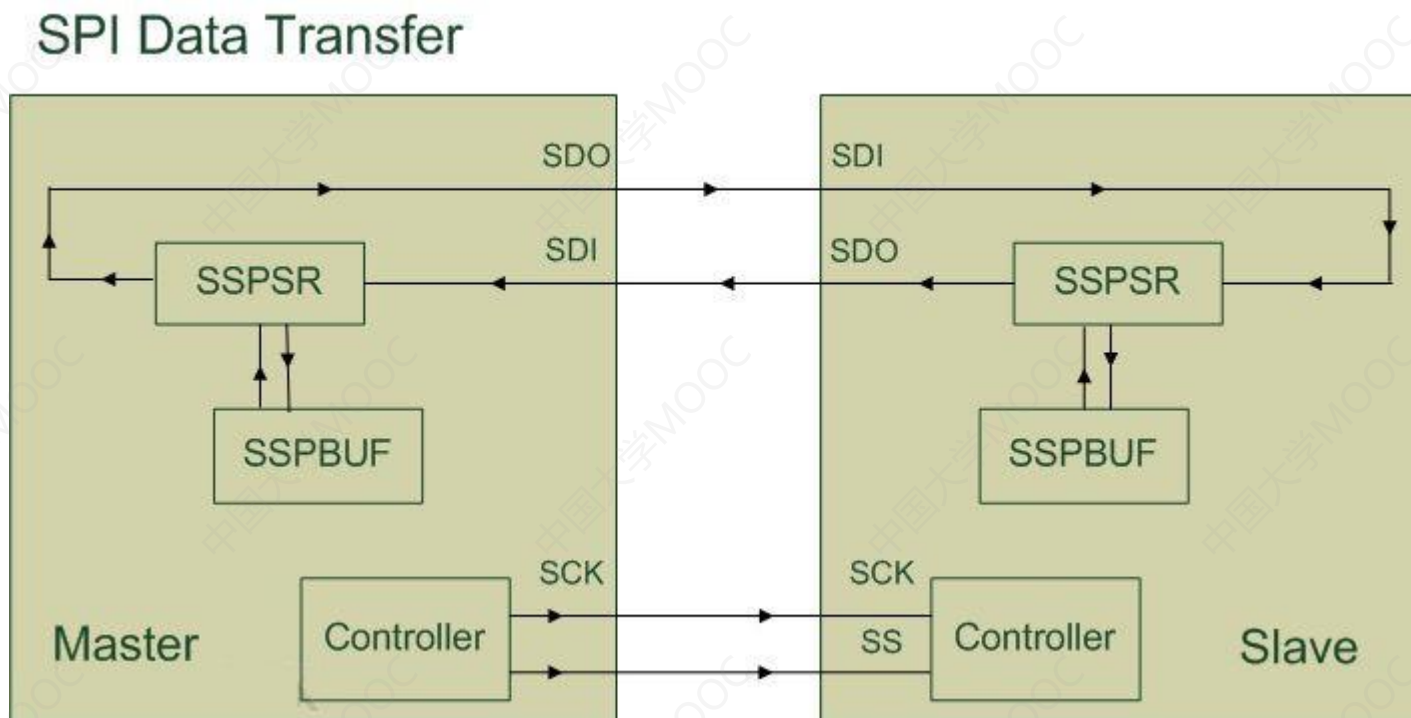
主从机通信模式

高速、同步、全双工、非差分、总线式

支持全双工通信、通信简单、数据传输速率快。

没有指定的流控制，没有应答机制确认是否接收到数据，所以在数据可靠性上有一定缺陷。

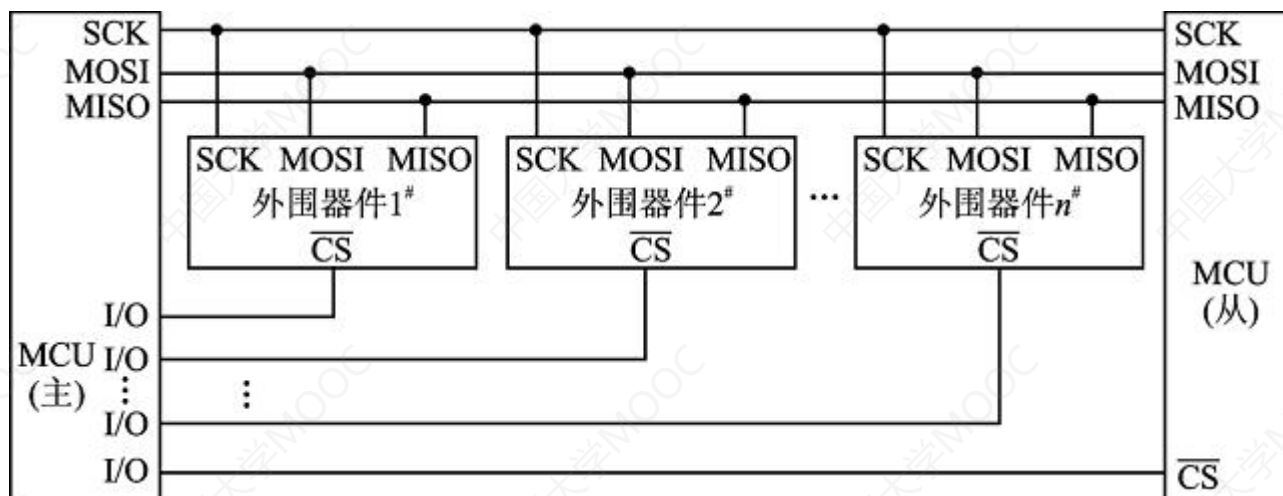
树莓派 Pi 4B介绍 - SPI



SPI的内部工作机制

SSPSR 是 SPI 设备内部的移位寄存器(Shift Register). 它的主要作用是根据 SPI时钟信号状态, 往 SSPBUF 里移入或者移出数据, 每次移动的数据大小由 Bus-Width 以及 Channel-Width 所决定。

树莓派 Pi 4B介绍 - SPI



SPI的通信原理很简单，它以主从方式工作，这种模式通常有一个主设备和一个或多个从设备，需要至少4根线，事实上3根也可以(单向传输时)。也是所有基于SPI的设备共有的，它们是SDI(数据输入)、SDO(数据输出)、SCLK(时钟)、CS(片选)。

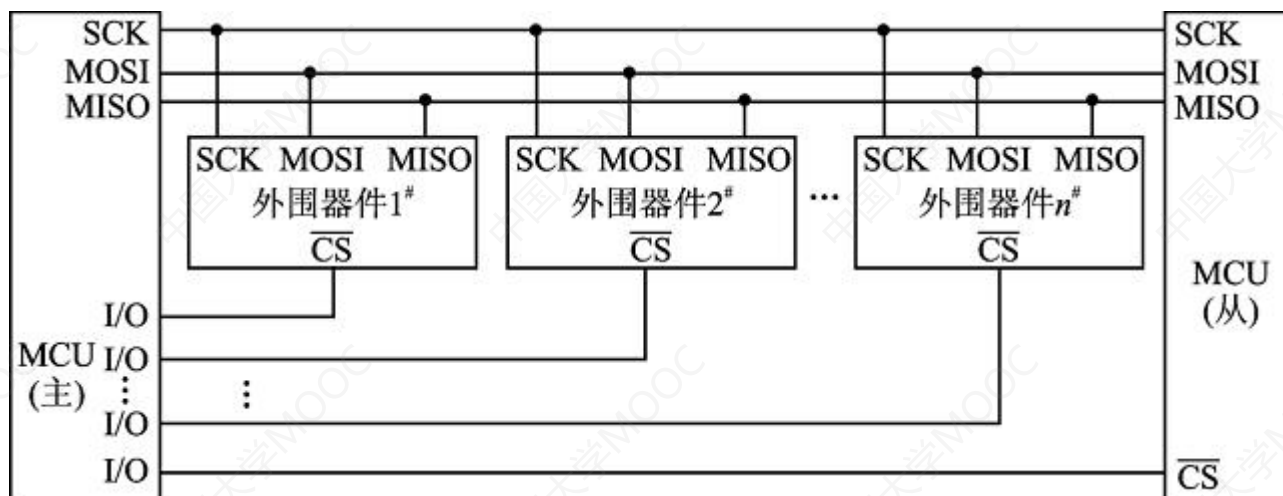
(1)SDO/MOSI – 主设备数据输出，从设备数据输入;

(2)SDI/MISO – 主设备数据输入，从设备数据输出;

(3)SCLK – 时钟信号，由主设备产生;

(4)CS/SS – 从设备使能信号，由主设备控制。

树莓派 Pi 4B介绍 - SPI



SPI通信有4种不同的模式

不同的从设备可能在出厂时就是配置为某种模式，这是不能改变的；但通信双方必须是工作在同一模式下，所以主设备的SPI模式进行配置，通过CPOL（时钟极性）和CPHA（时钟相位）来控制主设备的通信模式。

Mode0 : CPOL=0 , CPHA=0

Mode1 : CPOL=0 , CPHA=1

Mode2 : CPOL=1 , CPHA=0

Mode3 : CPOL=1 , CPHA=1

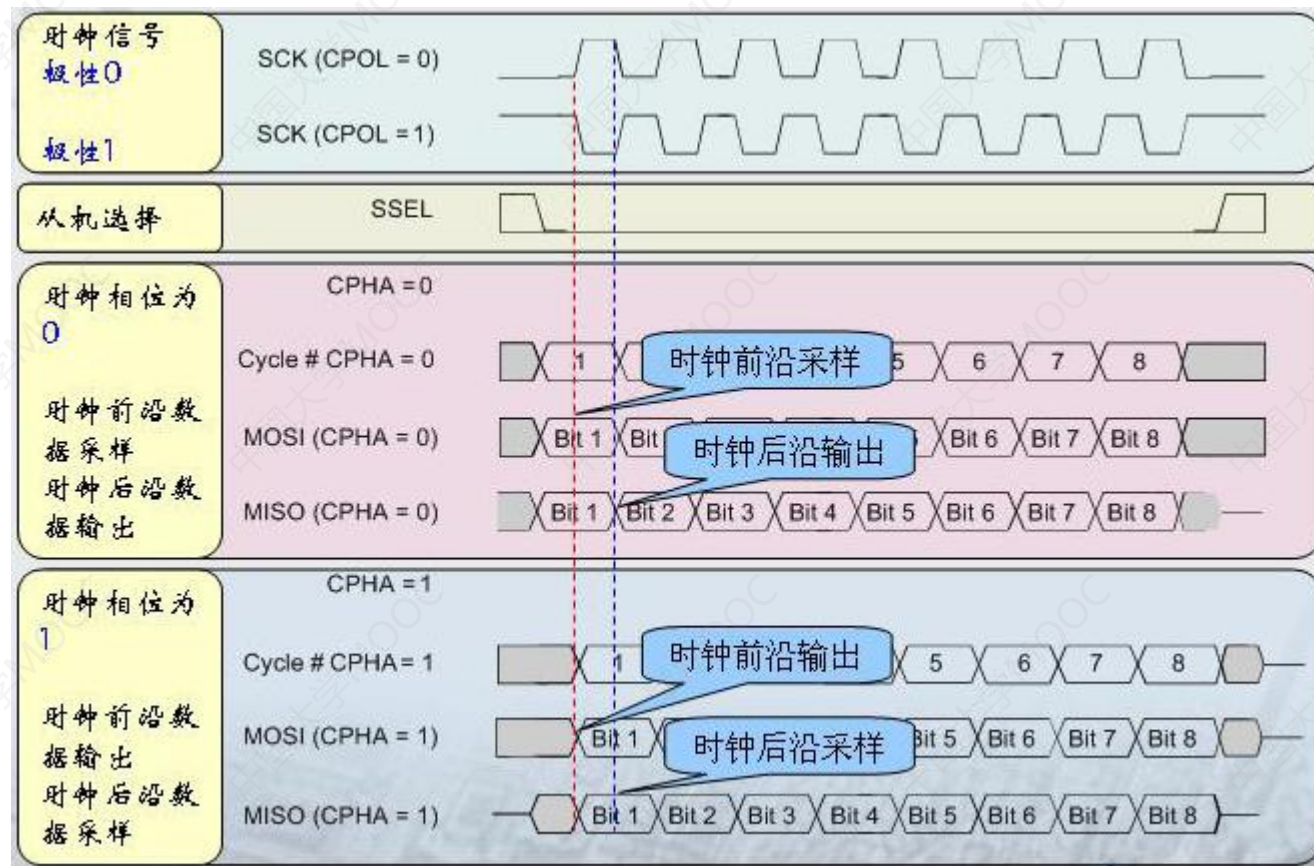
树莓派 Pi 4B介绍 - SPI



北京科技大学
University of Science and Technology Beijing

主设备能够控制时钟，因为SPI通信并不像UART或者IIC通信那样有专门的通信周期，有专门的通信起始信号，有专门的通信结束信号；

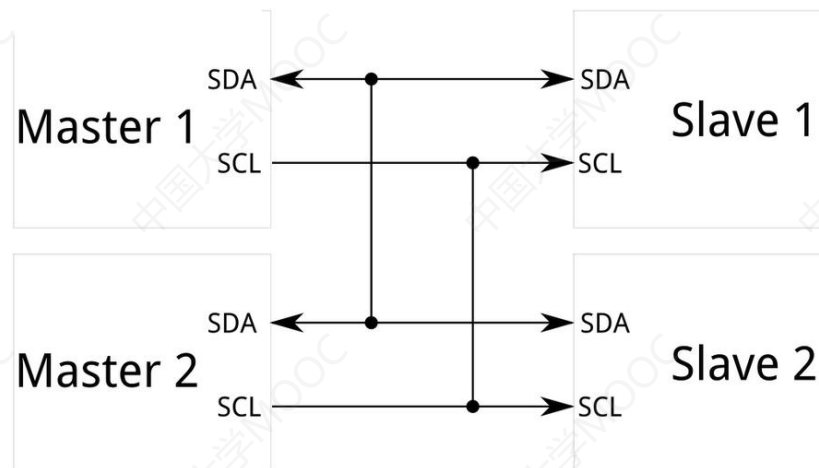
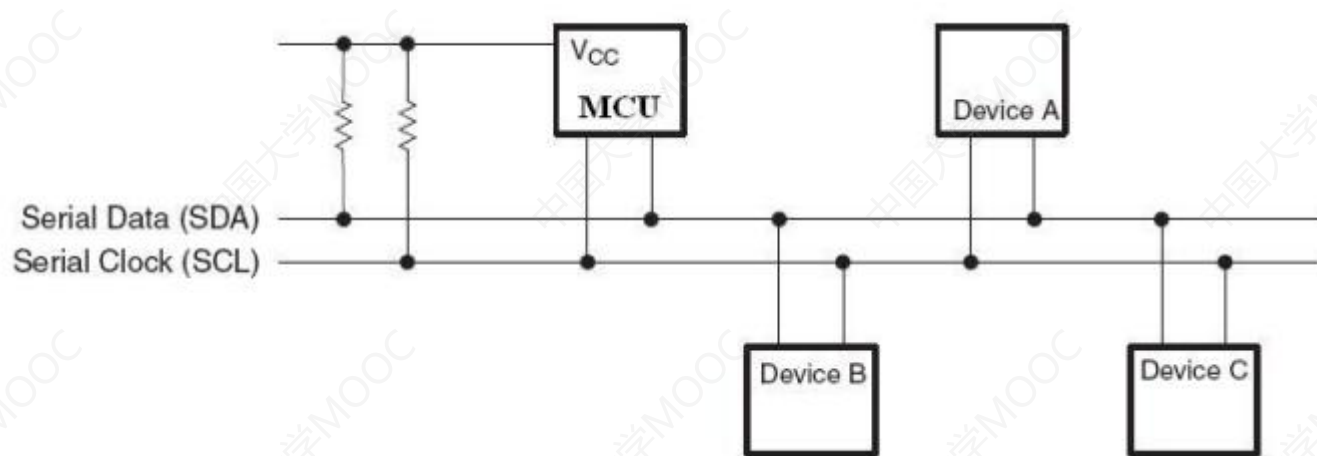
所以SPI协议能够通过控制时钟信号线，控制通信过程，当没有数据交互的时候时钟线保持高电平或保持低电平。



树莓派 Pi 4B介绍 - I2C (IIC)



北京科技大学
University of Science and Technology Beijing



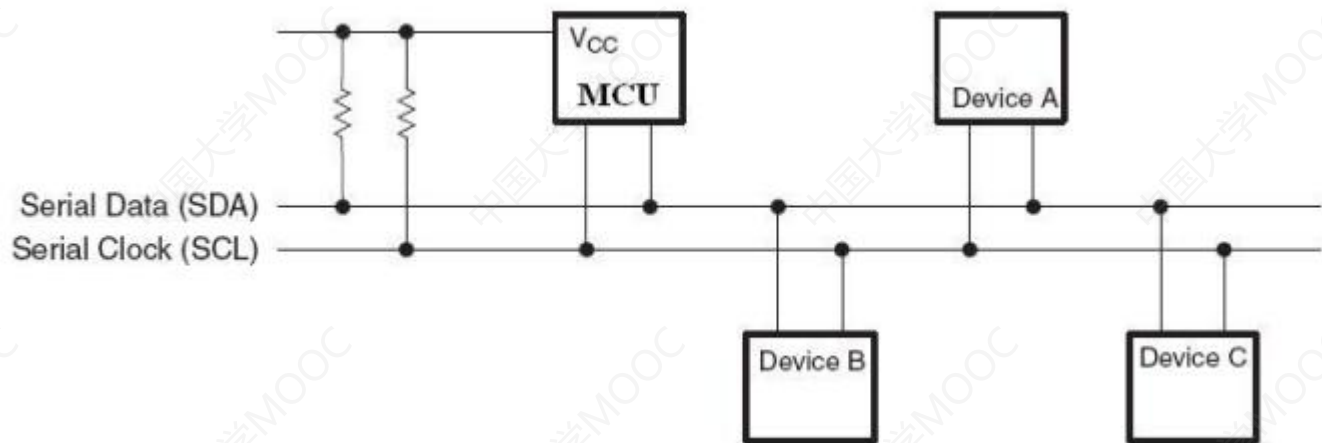
I2C通讯协议(Inter - Integrated Circuit)引脚少，硬件实现简单，可扩展性强现在被广泛地使用在系统内多个集成电路(IC)间的通讯。SCL和SDA都是OC开路输出，需要再总线上拉电阻。

实现I2C需要两根信号线完成信息交换，SCL时钟信号线，SDA数据输入/输出线。它属于同步通信，由于输入输出数据均使用一根线，因此通信方向为半双工。

I2C最少只需要两根线，和异步串口类似，但可以支持多个Slave设备。一个I2C理论上最多可挂载127个设备，但除去保留地址，最多可挂载112个设备。

大部分I2C设备支持100KHz和400KHz模式。

树莓派 Pi 4B介绍 - I2C (IIC)



使用I2C传输数据每发送8bits数据，就需要额外1bit的元数据（ACK或NACK）。

I2C支持双向数据交换，由于仅有一根数据线，故通信是半双工的。硬件复杂度也位于串口和SPI之间，而软件实现可以相当简单。

I2C设备都有设备地址，可以多对多（多个主设备，多个从设备）

一对多：这种比较常见，也不会有太多问题

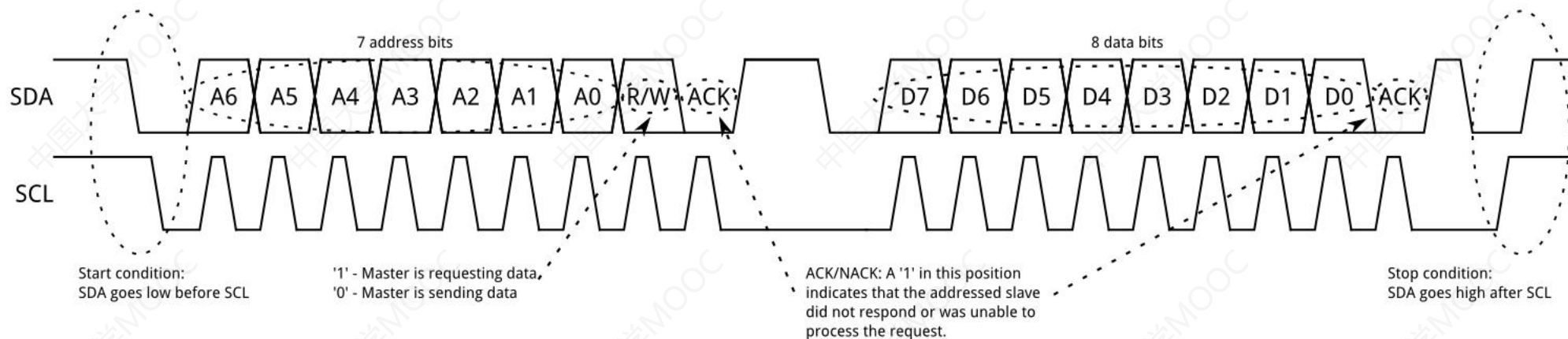
多对多：需要确保时钟（SCL）同步，同一时刻一台主设备工作。

SCL时钟速率不要高于从设备

树莓派 Pi 4B介绍 - I2C (IIC)



北京科技大学
University of Science and Technology Beijing

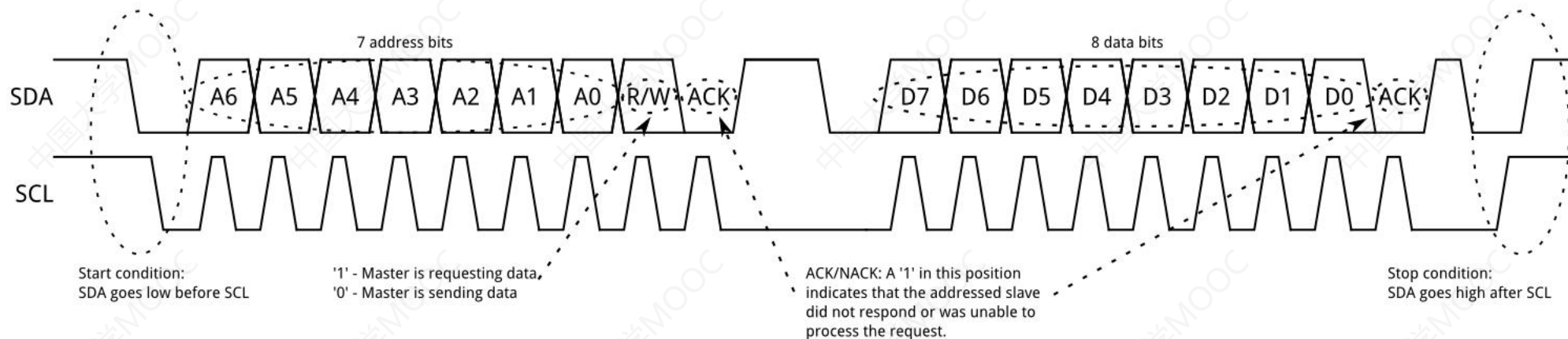


- 1、开始条件：SDA为高电平(1)，SCL为高电平时(1)，SDA由高向低跳变（下降沿）
- 2、发送数据：8个数据位，先传最高位（MSB），然后释放SDA线（1）（拉高电平）
SCL为低电平时(0)，SDA可进行电平转换；
SCL为高电平时(1)，SDA保持不变，接收设备（主设备或从设备）读取数据。
- 3、接收响应：ACK，在第九个周期（可以说是第九个数据位）
接收设备拉低SDA（0）；答复发送设备，接收完成；（I2C应答机制）
- 4、结束条件：SDA为低电平(0)，SCL为高电平时(1)，SDA由低向高跳变（上升沿）

树莓派 Pi 4B介绍 - I2C (IIC)



北京科技大学
University of Science and Technology Beijing



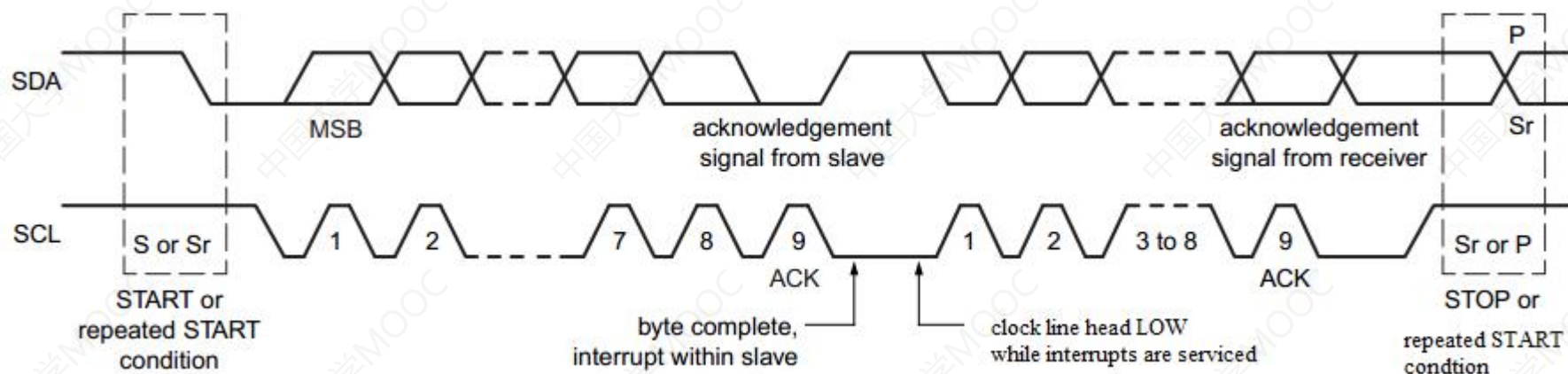
5、重复开始条件：

Master需要在一次通信中进行多次消息交换（例如与不同的Slave传输消息，或切换读写操作）

开始和结束不一一对应，可以发送多次开始条件，只有一个结束条件

- 主设备发送地址帧（地址+写），发送要读取的寄存器地址（如0xA0）
- 主设备发送地址帧（地址+读），接收（0xA0）寄存器数据

树莓派 Pi 4B介绍 - I2C (IIC)



6、时钟拉伸：

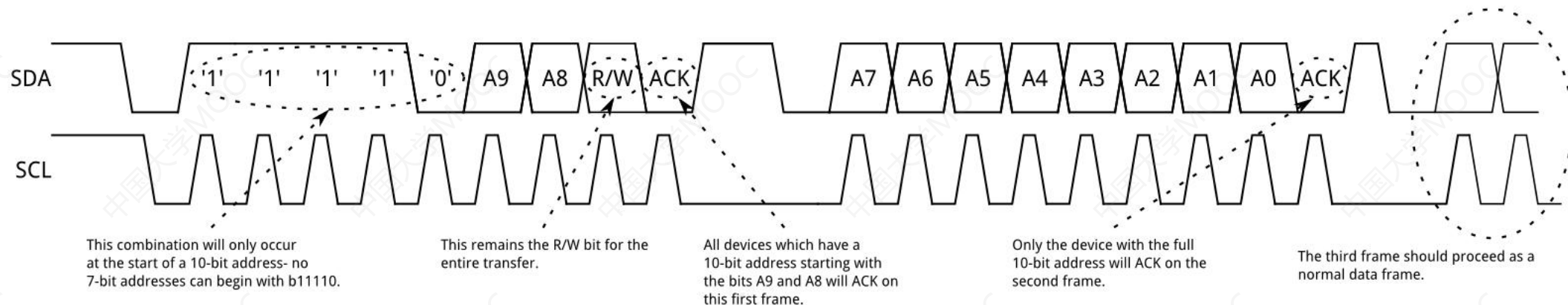
有时候，低速Slave可能由于上一个请求还没处理完，尚无法继续接收Master的后续请求，即Master的数据传输速率超过了Slave的处理能力。这种情况下，Slave可以进行时钟拉伸来要求Master暂停传输数据——通常时钟都是由Master提供的，Slave只是在SDA上放数据或读数据。

而时钟拉伸则是Slave在Master释放SCL后，将SCL主动拉低并保持，此时要求Master停止在SCL上产生脉冲以及在SDA上发送数据，直到Slave释放SCL（SCL为高电平）。之后，Master便可以继续正常的数据传输了。可见时钟拉伸实际上是利用了时钟同步的机制，只是时钟由Slave产生。

树莓派 Pi 4B介绍 - I2C (IIC)



北京科技大学
University of Science and Technology Beijing



7、传输详细内容

1) 地址帧：(用于指明消息是发送给哪个Slave)

地址帧：7个地址位+1个读写位(0写1读)

10bit地址帧：b1111 0XX(W/R)+bXXXXX XXXX；

2) 数据帧：由Master发往Slave的数据(或由Slave发往Master)，每一帧是8-bit的数据

3) 功能位：根据(1地址帧)的读写位，确定发送/接收设备

写：和发送地址帧一样，主设备(发送)，从设备(接收)；

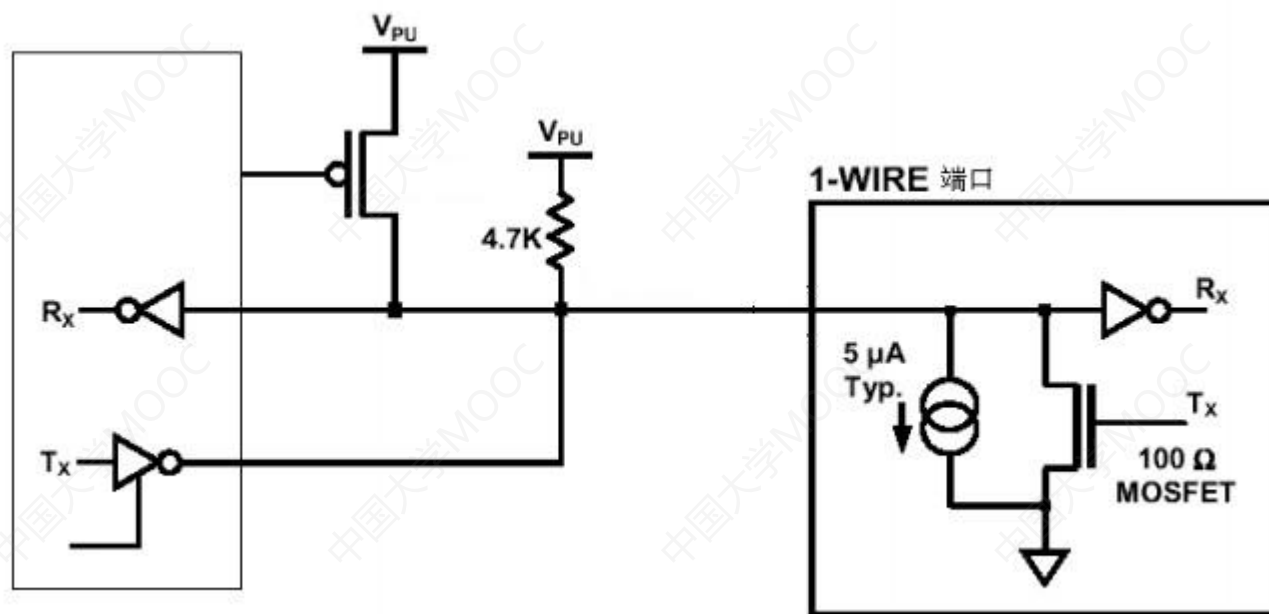
读：反过来，主设备(接收)，从设备(发送)；

后续的数据传输中，发送设备发送数据，接收设备需要发送ACK

树莓派 Pi 4B介绍 - 1-Wire



北京科技大学
University of Science and Technology Beijing



1-Wire (单总线技术)

单总线技术 (1-Wire Bus) 是美国的达拉斯半导体公司 (DALLAS SEMICONDUCTOR) 推出了一项特有的技术, 它采用单根信号线, 既可传输时钟, 又能传输数据, 而且数据传输是双向的, 因而这种单总线技术具有线路简单, 硬件开销少, 成本低廉, 便于总线扩展和维护等优点。

如果增加总线强上拉MOSFET, 可以使用寄生供电为从机供电, 从而实现GND和1-Wire信号线, 实现通讯和供电的功能。

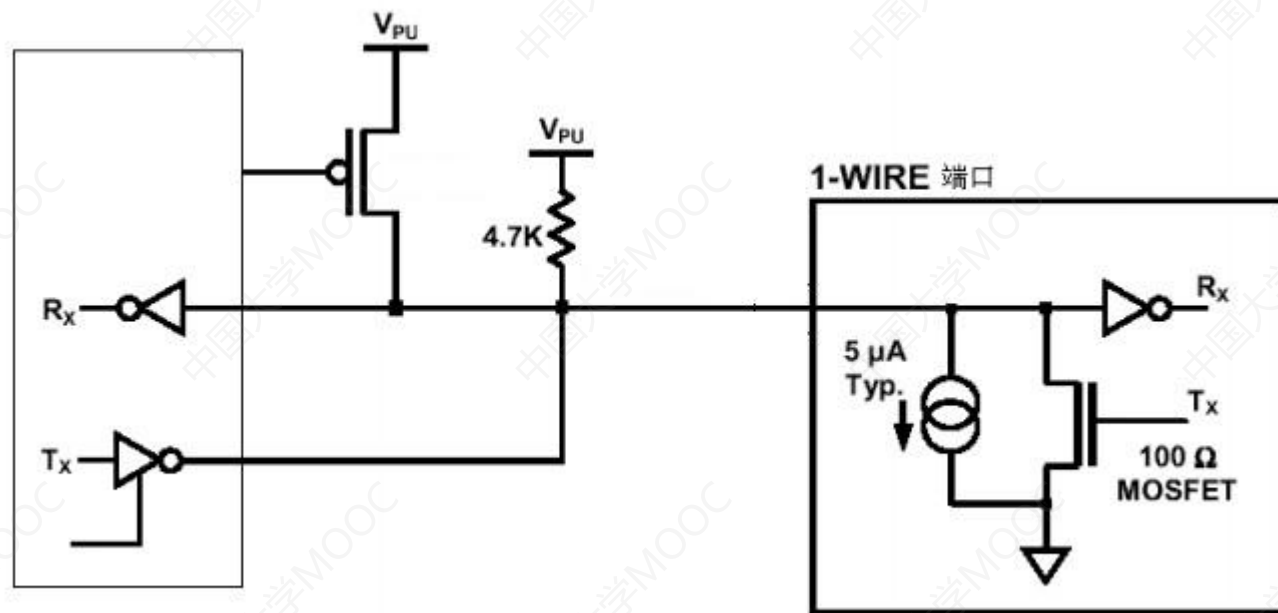
树莓派 Pi 4B介绍 - 1-Wire



北京科技大学
University of Science and Technology Beijing

1-Wire (单总线技术)

单总线要求外接一个约为5k的上拉电阻；单总线的闲置状态为高电平。不管什么原因，如果传输过程需要暂时挂起，且要求传输过程还能够继续的话，则总线必须处于空闲状态。位传输之间的恢复时间没有限制，只要总线在恢复期间处于空闲状态（高电平）。如果总线保持低电平超过480us，总线上的所有器件将复位。



树莓派 Pi 4B介绍 - 1-Wire



北京科技大学
University of Science and Technology Beijing

命令序列

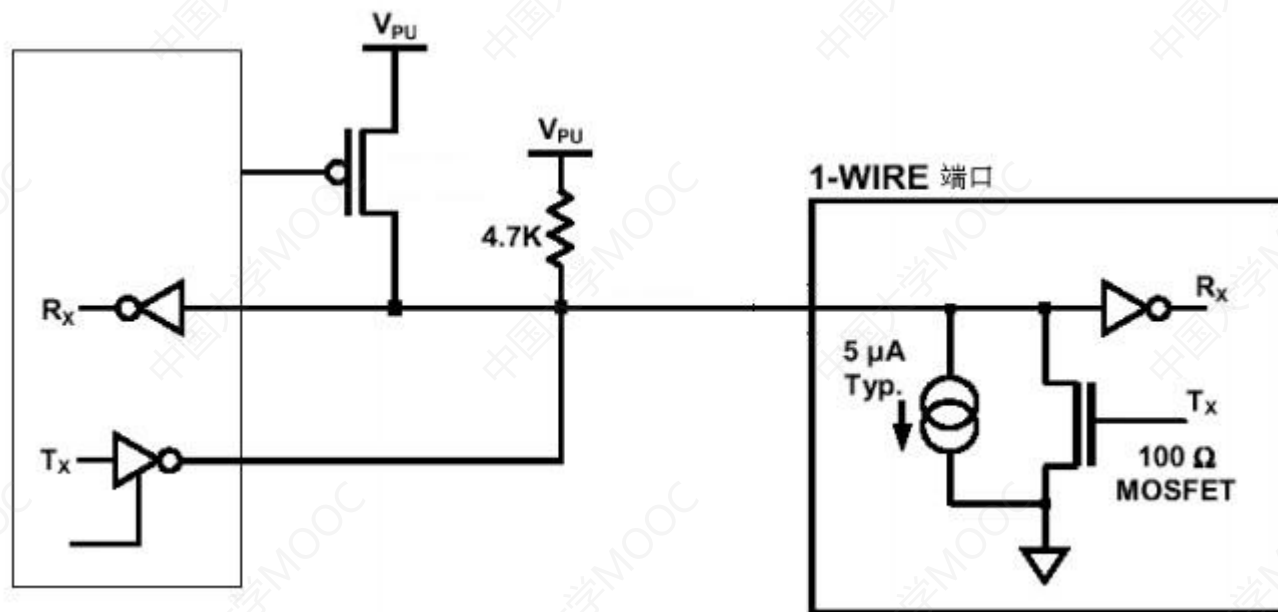
典型的单总线命令序列如下：

第一步：初始化

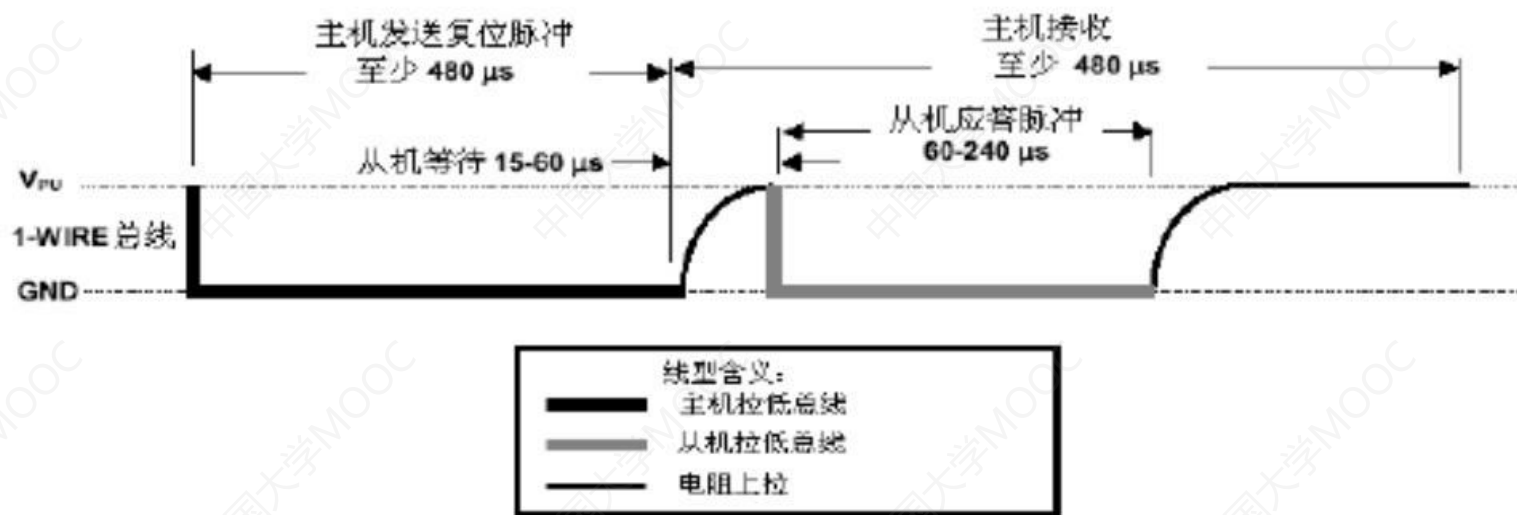
第二步：ROM命令（跟随需要交换的数据）

第三步：功能命令（跟随需要交换的数据）

每次访问单总线器件，必须严格遵守这个命令序列，如果出现序列混乱，则单总线器件不会响应主机。但是，这个准则对于搜索ROM命令和报警搜索命令例外，在执行两者中任何一条命令之后，主机不能执行其后的功能命令，必须返回至第一步。



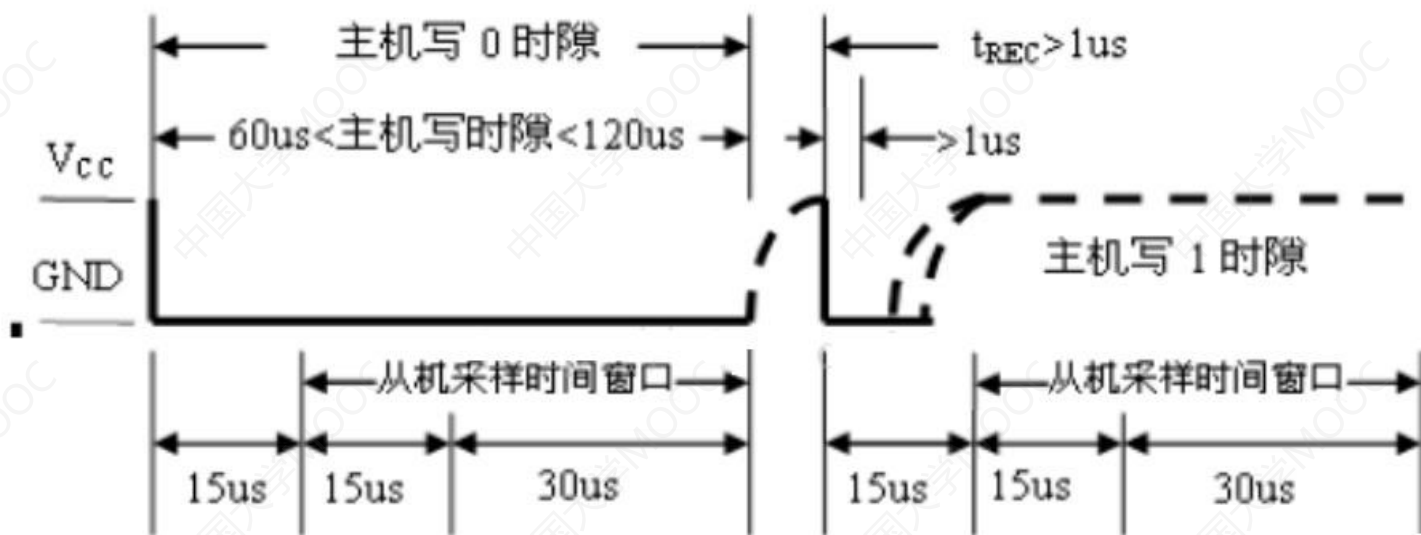
树莓派 Pi 4B介绍 - 1-Wire



初始化序列

单总线上的所有通信都是以初始化序列开始，由主机发出的复位脉冲及从机的应答脉冲组成，如图所示。当从机发出响应主机的应答脉冲时，即向主机表明它处于总线上，且工作准备就绪。在主机初始化过程，主机通过拉低单总线至少 $480\ \mu s$ ，以产生 (Tx) 复位脉冲。接着，主机释放总线，并进入接收模式 (Rx)。当总线被释放后， $5k$ 上拉电阻将单总线拉高。在单总线器件检测到上升沿后，延时 $15-60\ \mu s$ ，接着通过拉低总线 $60-240\ \mu s$ ，以产生应答脉冲。

树莓派 Pi 4B介绍 - 1-Wire



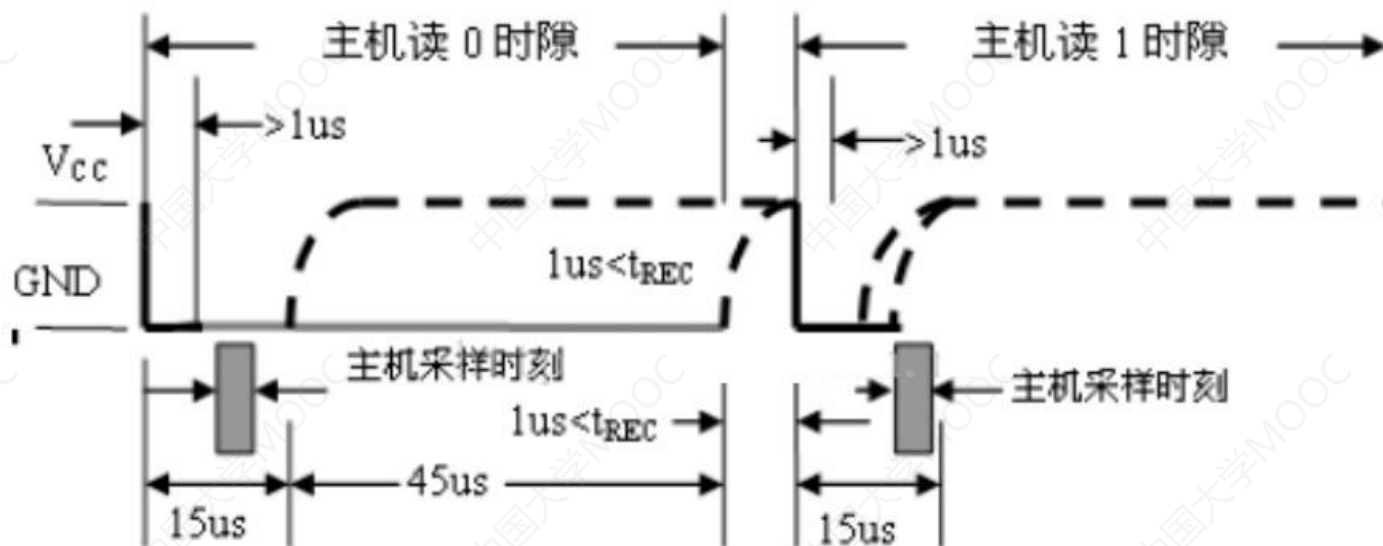
写时隙

在写时隙期间，主机向单总线器件写入数据；而在读时隙期间，主机读入来自从机的数据。在每一个时隙，总线只能传输一位数据。

产生写0时隙的方式：在主机拉低总线后，只需在整个时隙期间保持低电平即可（至少60μs）。

产生写1时隙的方式：主机在拉低总线后，接着必须在15μs之内释放总线，由5k上拉电阻将总线拉至高电平。

树莓派 Pi 4B介绍 - 1-Wire



读时隙

单总线器件仅在主机发出读时隙时，才向主机传输数据，所以在主机发出读数据命令后，必须马上产生读时隙，以便从机能够传输数据。所有读时隙至少需要 $60\mu s$ ，且在两次独立的读时隙之间至少需要 $1\mu s$ 的恢复时间。每个读时隙都由主机发起，至少拉低总线 $1\mu s$ ，如图所示。在主机发起读时隙之后，单总线器件才开始在总线上发送0或1。

若从机发送1，则保持总线为高电平；若发送0，则拉低总线。当发送0时，从机在该时隙结束后释放总线，由上拉电阻将总线拉回至空闲高电平状态。从机发出的数据在起始时隙之后，保持有效时间 $15\mu s$ ，因而，主机在读时隙期间必须释放总线，并且在时隙起始后的 $15\mu s$ 之内采样总线状态。

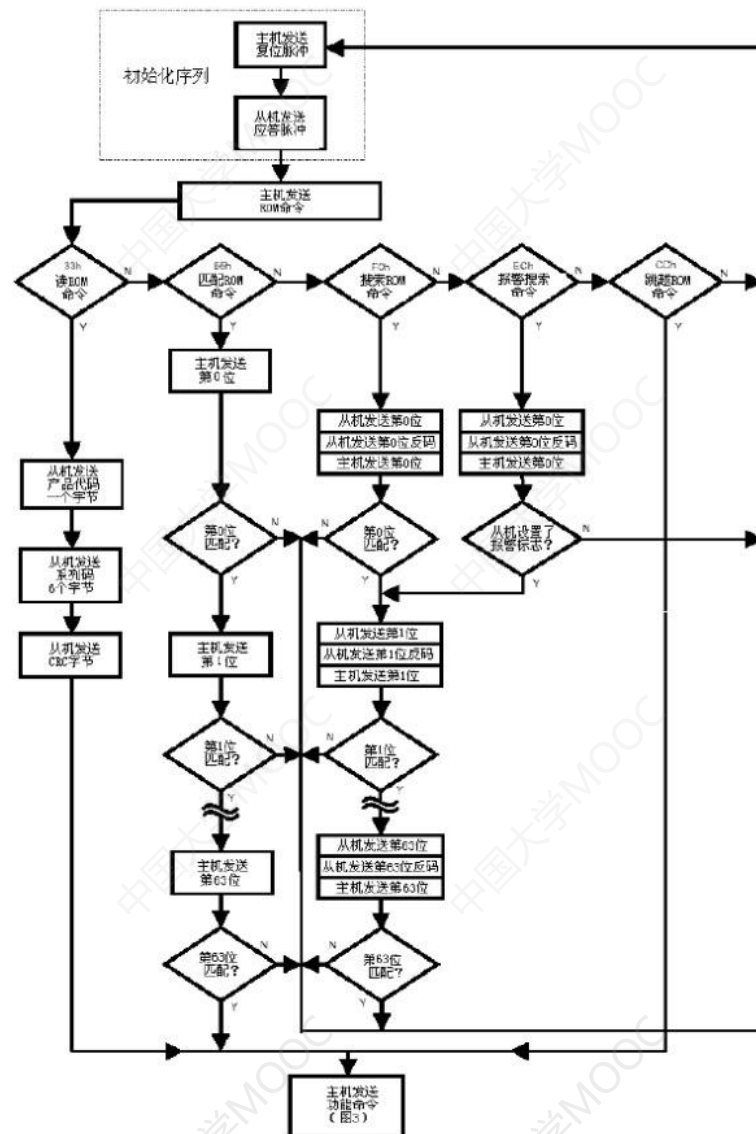
树莓派 Pi 4B介绍 - 1-Wire



ROM命令

在主机检测到应答脉冲后，就可以发出 ROM 命令。这些命令与各个从机设备的唯一64位 ROM代码相关，允许主机在单总线上连接多个从机设备时，指定操作某个从机设备。这些命令还允许主机能够检测到总线上有多少个从机设备以及其设备类型，或者有没有设备处于报警状态。从机设备可能支持5中 ROM 命令（实际情况与具体型号有关），每种命令长度为8位。主机在发出功能命令之前，必须送出合适的 ROM 命令。

- 1、搜索ROM[F0h]
- 2、ROM[33h](仅适用于单节点)
- 3、匹配ROM[55h]
- 4、跳跃ROM[CCh]（仅适用单节点）
- 5、报警搜索[Ech]（少数器件支持）



树莓派 Pi 4B介绍 - 1-Wire

功能命令（以DS18B20为例）

在主机发出 ROM 命令，以访问某个指定的 DS18B20，接着就可以发出 DS18B20 支持的某个功能命令。这些命令允许主机写入或读出DS18B20暂存器、启动温度转换以及判断从机的供电方式。DS18B20的功能命令总结下列表中

命令	描述	命令代码	发送命令后，单总线上的响应信息	注释
温度转换命令				
转换温度	启动温度转换	44h	无	1
存储器命令				
读暂存器	读全部的暂存器内容，包括 CRC 字节	BEh	DS18B20 传输多达 9 个字节至主机	2
写暂存器	写暂存器第 2、3 和 4 个字节的数据（即 T_H ， T_L 和配置寄存器）	4Eh	主机传输 3 个字节数据至 DS18B20	3
复制暂存器	将暂存器中的 T_H ， T_L 和配置字节复制到 EEPROM 中	48h	无	1
回读 EEPROM	将 T_H ， T_L 和配置字节从 EEPROM 回读至暂存器中	B8h	DS18B20 传送回读状态至主机	

▶▶▶ 树莓派 Pi 4B介绍 - 1-Wire



北京科技大学
University of Science and Technology Beijing

ROM搜索

ROM 搜索过程只是一个简单的三步循环程序：读一位、读该位的补码、写入一个期望的数据位。总线主机在 ROM 的每一位上都重复这样的三步循环程序。当完成某个器件后，主机就能够知晓该器件的 ROM 信息。剩下的设备数量及其 ROM 代码通过相同的过程即可获得。

下面的 ROM 搜索过程实例假设四个不同的器件被连接至同一条总线上，它们的 ROM 代码如下所示：

ROM1 00110101...

ROM2 10101010...

ROM3 11110101...

ROM4 00010001...