**一、Pin函数：**

FE CA 08 42 10 03 A1

第1个字节： F：数据包起始校验 E：IO口配置校验

第2个字节： IO函数选择 CA: void pPinInit(); DA: void pPinOutPut(); AA:void pPinDeInit();

第3个字节： IO口端口选择 eg: 08--GPIOA 详见BYTE说明

第4个字节： Pin选择 eg: 42--GPIO\_Pin\_5

第5个字节： Pin\_Mode选项 eg: 10--GPIO\_Mode\_Out\_PP

第6个字节： GPIO\_Speed 选择 eg: 03--GPIO\_Speed\_50MHz

第7个字节： 校验结束

void pPinInit(uint32\_t IO,uint32\_t Pin,uint32\_t GPIO\_Mode , uint32\_t GPIO\_Speed)；

FE AA 08 A1

第1个字节： F：数据包起始校验 E：IO口配置校验

第2个字节： IO函数选择

第3个字节： IO口端口选择 eg: 08--GPIOA 详见BYTE说明

第4个字节： 校验结束

void pPinDeInit(uint32\_t IO);

FE DA 08 42 00 A1

第1个字节： F：数据包起始校验 E：IO口配置校验

第2个字节： IO函数选择

第3个字节： IO口端口选择 eg: 08--GPIOA 详见BYTE说明

第4个字节： Pin选择 eg: 42--GPIO\_Pin\_5 详见BYTE说明

第5个字节: Pin 0x00设置为低电位 0x01设置为高电位

第6个字节： 校验结束

void pPinOutPut(uint32\_t IO,uint32\_t Pin, uint32\_t Level)；

FE DB 08 42 A1

第1个字节： F：数据包起始校验 E：IO口配置校验

第2个字节： IO函数选择

第3个字节： IO口端口选择 eg: 08--GPIOA 详见BYTE说明

第4个字节： Pin选择 eg: 20--GPIO\_Pin\_5 详见BYTE说明

第5个字节： 校验结束

void pPinReadBit(uint32\_t IO,uint32\_t Pin);

Information:这里是指浮空输入读取

校验的第一个字节是FE 第二个字节为 00 或 01（01表示高电位）

返回 FE 00 A1

1. **USART函数：**

FA CA 44 03 00 00 00 00 0C 28 A1

第1个字节： F：数据包起始校验 A： USART配置校验

第2个字节： USART函数选择 0xCA--void pUsartInit();

第3个字节: USART 端口选择 eg: 0x44-USART2 详见BYTE说明

第4个字节: USART RAUD选择 eg: 0x03--115200 详见BYTE说明

第5个字节: USART WordLength选择 eg:0x00--8位数据位 详见BYTE说明

第6个字节： USART StopBits 选择 eg: 0x00--1位停止位 详见BYTE说明

第7个字节： USART Parity 选择 eg: 0x00--无校验位 详见BYTE说明

第8个字节： USART HardwareFlowControl 选择 eg: 0x00--无硬件流控制 详见BYTE说明

第9个字节： USART Mode 选择 eg: 0x0C--发送接收都工作 详见BYTE说明

第10个字节： USART MsgMdoe 选择 eg: 0x28--自动读取传感器字节流为40个字节 详见BYTE

第11个字节： 校验结束

void pUsartInit(uint32\_t Channel,uint32\_t BaudRate ,uint32\_t WordLength,uint32\_t StopBits,uint32\_t Parity,uint32\_t HardwareFlowControl ,uint32\_t Mode , uint32\_t msgMode );

Information: 这里的第10个字节，是指USART接受数据放入缓存是否是自动的还是响应式的，0x00表示的是响应式的缓存为500个字节，0x0f则表示自动获取数据量为16个字节。

FA AA 44 A1

第1个字节： F：数据包起始校验 A： USART配置校验

第2个字节： USART函数选择 0xAA--void pUsartDeInit();

第3个字节: USART 端口选择 eg: 0x44-USART2 详见BYTE说明

第4个字节： 校验结束

void pUsartDeInit(uint32\_t Channel);

FA DA 44 FF ...... A1

第1个字节： F：数据包起始校验 A： USART配置校验

第2个字节： USART函数选择 0xDA--void pUsartWrite();

第3个字节: USART 端口选择 eg: 0x44-USART2 详见BYTE说明

第4个字节: USART 通讯字节长度 eg: 0xFF 为传输256 个字节

.......： USART 传输256字节

第n个字节: 校验结束

void pUsartWrite(uint32\_t Channel,uint8\_t \*word,short length);

FA DB 44 A1

第1个字节： F：数据包起始校验 A： USART配置校验

第2个字节： USART函数选择 0xDB--void pUsartRead();

第3个字节: USART 端口选择 eg: 0x44-USART2 详见BYTE说明

第4个字节: 校验结束

void pUsartRead(uint32\_t Channel);

Information: 这里直接读取缓存的字节信息有两种模式，第一种是响应式得到缓存字节，第二种是传感器模块连续不断发送字节，进行缓存。这里的配置需要在USART初始化的时候进行设置。USART2 返回数据包开头校验位为FA D2 + 00 01(short类型数量) +数据+结尾校验位为A2，UART4 返回数据包开头校验位为FA C3+ 00 01(short类型数量) +数据+结尾校验位为A2

FA DC A1 (此函数是通过串口反馈数据首发函数 \*\*至关重要)

第1个字节： F：数据包起始校验 A： USART配置校验

第2个字节： USART函数选择 0xDC--void pUsartOnLine(void);

第3个字节: 校验结束

1. ADC函数：

FC CA 0C ...... A1

第1个字节: F：数据包起始校验 C： ADC配置校验

第2个字节： ADC函数选择 0xCA--void pADCInit();

第3个字节: ADC 采样分频 eg:0x0C-- 9MHz

...... : ADC 多通道选择 eg: {0x00 0x01 }-- 采用通道一和通道 二 详见BYTE说明

第4个字节: 校验结束

void pADCInit(uint32\_t clkdiv,uint8\_t \*channel)；

Information: 这里采样是基于ADC1 使用DMA的方式进行采样存储

FC AA A1

第1个字节: F：数据包起始校验 C： ADC配置校验

第2个字节： ADC函数选择 0xAA--void pADCDeInit();

第3个字节: 校验结束

Void pADCDeInit();

Information：这里默认注释ADC1

FC DA 01 A1

第1个字节: F：数据包起始校验 C： ADC配置校验

第2个字节： ADC函数选择 0xDA--void pADCRead();

第3个字节: ADC 读取通道选择 eg: 0x01--读取通道一的ADC值 详见BYTE说明

第4个字节: 校验结束

void pADCRead(uint8\_t channel);

Information: 这里读取通道，依据电路实际通道设置

返回初始校验为FC 结束校验位为A3

1. PWM函数：

说明：

PWM输出和采用TIM1 TIM8 TIM2 TIM3 TIM4 TIM5 定时器均可

PWM捕获采用 TIM2 TIM3 TIM4 TIM5 定时器均可

脉冲读取采用TIM1 TIM8 TIM2 TIM3 TIM4 TIM5定时器均可并且利用基本定时器TIM6 进行时间段内自动采样。

FD CA 04 01 02 00 00 20 03 C8 00 90 01 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xCA--void pPWMInit();

第3个字节: PWM 选择 eg: 0x04 -- TIM3

第4~7个字节：PWM输出通道选择 eg: 0x01 0x02 0x00 0x00-- 使用PWM通道1和通道2

第8~9个字节：PWM周期为800

第10~11个字节：PWM预分频为 200

第11~12个字节：PWM初始高电平周期为400

第13个字节： 校验结束

void pPWMInit(uint32\_t TIM,uint8\_t \*Channel ,uint16\_t Period, uint16\_t Prescaler, uint16\_t Pulse)；

FD CB 04 04 00 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xCB--void pPWMICInit();

第3个字节: PWM 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4~5个字节：两通道检验 0X04 表示使用通道一 ，00表示不适用通道2 通道依据本协议

第6个字节： 校验结束

void pPWMICInit(uint32\_t TIM,uint32\_t \*useMode)；

FD CC 04 03 E8 03 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xCC--void pPWMPluseInit();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节: TIM\_Encoder\_Mode eg: 03--TIM\_EncoderMode\_TI12 详见BYTE说明

第5个字节： 设定记录脉冲时间 eg: 1000ms

第6个字节 校验结束

void pPWMPluseInit(uint32\_t TIM,uint32\_t mMode,uint32\_t Cir\_Tim)；

FD AA 04 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xAA--void pPWMDeInit();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节： 校验结束

void pPWMDeInit(uint32\_t TIM);

FD DA 04 01 90 01 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xDA--void pPWMSetPulseWid();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节： 选择通道 eg:0x01-- 通道1 详见BYTE说明

第5~6个字节： PWM设置脉宽 eg:此时为400

第7个字节： 校验结束

void pPWMSetPulseWid (uint32\_t TIM,uint16\_t Channel,uint16\_t pulsewid)；

Information:这里设置都是指有效电平宽度

FD DB 04 11 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xDB--void pPWMICGetPulseWid();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节： 选择读取通道 eg:0x11-- 读取通道1高电平 0x01读取通道1低电平 详见BYTE说明

第5个字节： 校验结束

void pPWMICGetPulseWid(uint32\_t TIM,uint32\_t Channel)；

初始校验为0xFD 0XDB

FD DC 04 01 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xDC--void pPWMICGetFrequency();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节： 选择读取通道 eg:0x01-- 通道1读取 详见BYTE说明

第5个字节： 校验结束

void pPWMICGetFrequency(uint32\_t TIM,uint32\_t Channel);

初始校验为0xFD 0XDC

FD DD 04 01 A1

第1个字节: F：数据包起始校验 D： PWM配置校验

第2个字节: 0xDD--void pPWMGetPulse();

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4个字节： 选择读取通道 eg:0x01-- 通道1 详见BYTE说明

第5个字节： 校验结束

void pPWMGetPulse(uint32\_t TIM,u8 channel);

初始校验为0xFD 0xDD

FD DE 04 00 48 A1

第1个字节： F：数据包起始校验 D： PWM配置校验

第2个字节: 0xDE--void pPWMSetFrequency(uint32\_t TIM, uint16\_t frequancy);

第3个字节: 选择 eg: 0x04 -- TIM3 详见BYTE说明

第4~5个字节： PWM设置预分频 eg:此时为72

第6个字节： 校验结束

void pPWMSetFrequency(uint32\_t TIM, uint16\_t frequancy);

1. IIC函数：

FB CA 08 44 08 48 A1

第1个字节: F：数据包起始校验 B： IIC配置校验

第2个字节: 0xCA--void pIICInit();

第3个字节： IO口端口选择 eg: 08--GPIOA

第4个字节： Pin选择 eg: 44--GPIO\_Pin\_6 此为SCL

第5个字节： IO口端口选择 eg: 08--GPIOA 这里为了方便使用不同端口

第6个字节： Pin选择 eg: 48--GPIO\_Pin\_7 此为SDA

第7个字节： 校验结束

void pIICInit(uint32\_t IO1,uint32\_t SCL\_Pin,uint32\_t IO2, uint32\_t SDA\_Pin)；

Information:这里依次设置多个IIC，最多设置两个

FB DA 00 01 06 a0 03 06 01 03 06 FF 03 02 A1

第1个字节: F：数据包起始校验 B： IIC配置校验

第2个字节: DA-表示执行iic自配函数，

第3个字节: 00-表示使用第一个iic, 01-表示使用第二个iic

以后的字节：

I2CStart(); //01

I2CSendByte(0xa0); //06 a0

I2CWaitAck(); //03

I2CSendByte(address); //06 01

I2CWaitAck(); //03

I2CSendByte(info); //06 FF

I2CWaitAck(); //03

I2CStop(); //02

以上的自配函数是指向AT24C02指定地址写入一个字节数据

最后一个字节：校验结束

Information : 这里的IIC自配函数编号为

void IIC\_Start() 0x01

void IIC\_Stop() 0x02

void IIC\_WaitAck() 0x03

void IIC\_SendAck() 0x04

void IIC\_SendNoAck() 0x05

void IIC\_SendByte() 0x06

void IIC\_RecByte() 0x07

void IIC\_SendByte()

void IIC\_WaitAck() 0x63

void IIC\_RecByte()

void IIC\_WaitAck() 0x73

如以上的向AT24C02写数据可以为

FB DA 01 63 a0 63 01 63 FF 02 A1

当有读取指令的时候，会自动返回读取数值。

开头校验为是FB 选择IIC 00 或01 数据量 0x0F

例如： FB 01 0F 表示二号IIC, 数据的量为16个

六：Wifi函数（Manibus-ZE）

F9 CA 68 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xCA --void pWifiInit(void); 这里就默认使用USART3

第3个字节: 特殊校验位

第4个字节： 校验结束

F9 CB 69 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xCA --void pWifiReset(void); 这里就默认使用USART3

第3个字节: 特殊校验位

第4个字节： 校验结束

F9 CC 6A A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xCC--void pWifi\_AT\_Test(void);

第3个字节: 特殊校验位

第4个字节： 校验结束

Info: 有状态返回值 返回为 F9 STATE A6（详见BYTE说明）

F9 CD 01 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xCD--void pWifi\_Use\_MultipleId(FunctionalState enumEnUnvarnishTx);

第3个字节： 0x01 表示ENABLE 0x00 表示DISABLE

第4个字节： 校验结束

F9 CE 00 FF \*\*\*\*\*\*\* FF \*\*\*\*\*\*\* FF 00 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xCE--void pWifi\_LinkServer(ENUM\_NetPro\_TypeDef enumE, char \* ip, char \* ComNum, ENUM\_ID\_NO\_TypeDef id);

第3个字节： 选择网络通讯模式 00 表示TCP

第4个字节: FF 隔离校验位

第5~5+n个字节: \*\*\*\*\*\*\*表示链接服务器的IP

第5+n+1个字节: FF 隔离校验位

第5+n+2 ~ 5+n+2+m个字节: \*\*\*\*\*\*\*表示服务器端口

第5+n+2+m+1个字节: FF 隔离校验位

第5+n+2+m+2个字节: 00表示 ID 为0 详见BYTE说明

第5+n+2+m+3个字节: 校验结束

Info: 返回状态字节串 F9 00或 07 A6 详见BYTE说明

F9 DA 00 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xDA--void pWifi\_NetModeChoose(ENUM\_Net\_ModeTypeDef enumMode);

第3个字节: wifi 工作模式选择 00-- STA 模式 详见BYTE说明

第4个字节： 校验结束

Info: 返回状态字节串 F9(开头校验) 00（00表示正确，03表示该函数出现错误） A6（结尾校验）

F9 DB FF \*\*\*\*\*\*\* FF \*\*\*\*\*\*\*\*\* FF A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xDB--void pWifi\_JoinAP(char \* pSSID, char \* pPassWord);

第3个字节： FF 隔离校验位

第4~4+n个字节: \*\*\*\*\*\*\*表示 链接WIFI 的SSID

第4+n+1个字节： FF 隔离校验位

第4+n+2~4+n+2+m个字节: \*\*\*\*\*\*\*表示 链接WIFI 的密码

第4+n+2+m+1个字节: FF 隔离校验位

第4+n+2+m+2个字节: 校验结束

Info: 返回状态字节串 F9 00或 04 A6 详见BYTE说明

F9 DC FF \*\*\*\*\*\* FF \*\*\*\*\*\* FF 00 A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xDC--void pWifi\_BuildAP(char \* pSSID, char \* pPassWord, ENUM\_AP\_PsdMode\_TypeDef enunPsdMode);

第3个字节： FF 隔离校验位

第4~4+n个字节: \*\*\*\*\*\*\*表示 建立WIFI 的SSID

第4+n+1个字节： FF 隔离校验位

第4+n+2~4+n+2+m个字节: \*\*\*\*\*\*\*表示 建立WIFI 的密码

第4+n+2+m+1个字节: FF 隔离校验位

第4+n+2+m+2个字节: 安全模式

第4+n+2+m+3个字节: 校验结束

Info: 返回状态字节串 F9 00或 05 A6 详见BYTE说明

F9 DE 01 FF \*\*\*\*\*\*\* FF \*\*\*\*\*\*\* FF A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xDE--void pWifi\_ifServerMode(FunctionalState enumMode, char \* pPortNum, char \* pTimeOver);

第3个字节： 01表示ENABLE

第4个字节： FF 隔离校验位

第5~5+n个字节： \*\*\*\*\*\* 表示端口

第5+n+1个字节： FF 隔离校验位

第5+n+2~5+n+2+m个字节： \*\*\*\*\*\* 表示超时时间

第5+n+2+m+1个字节： FF 隔离校验位

第5+n+2+m+2个字节： 校验结束

Info: 返回状态字节串 F9 00或 08 A6 详见BYTE说明

F9 DF 5D A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xDE--void pWifi\_GetLinkStatus(void);

第3个字节： 特殊校验位

第4个字节： 校验结束

Info: 返回状态字节串 F9 00（无）或 02（STATUS: 2） A6   
有 STATUS: 2 STATUS: 3 STATUS: 4

F9 D9 9D A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xD9 --void pWifi\_GetIdLinkStatus(void);

第3个字节： 特殊校验位

第4个字节： 校验结束

Info: 返回状态字节串 F9 00（无） A6

返回的是连接状态

F9 D8 8A A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xD8--void pWifi\_Inquire\_ApIp ( );

第3个字节： 特殊校验位

第4个字节： 校验结束

Info: 返回状态字节串 F9 00或 09 A6 详见BYTE说明

F9 D7 7C A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xD7--void pWifi\_UnvarnishSend ( void );

第3个字节： 特殊校验位

第4个字节： 校验结束

Info: 返回状态字节串 F9 00或 10 A6 详见BYTE说明

F9 D6 6B A1

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xD6--void pWifi\_ExitUnvarnishSend (void);

第3个字节: 特殊校验位

第4个字节： 校验结束

F9 D5 01 00 A1 （此函数是通过WIFI反馈数据首发函数 \*\*至关重要

第1个字节: F：数据包起始校验 9： WIFI配置校验

第2个字节: 0xD5--void pWifiOnLine ( FunctionalState enumEnUnvarnishTx, ENUM\_ID\_NO\_TypeDef ucId );

第3个字节: 01 表示ENABLE

第4个字节： 表示ID 详见BYTE说明

第5个字节： 校验结束

//F9 D4 01 A1 (这里不使用)

//第1个字节: F：数据包起始校验 9： WIFI配置校验

//第2个字节: 0xD4--void pWifi\_ReceiveString(FunctionalState enumEnUnvarnishTx);

//第3个字节: 01 表示 ENABLE

//第4个字节： 校验结束