

水位数字传感器

BM62S3202-1

BMduino Library V1.0.1 说明

版本: V0.02

日期: 2024-09-09

www.bestmodulescorp.com

目录

简介	3
BMduino Lib 函数	3
BMduino 范例	4
范例 1: Read_WaterSensor_Value	4

简介

BM62S3202-1 是倍创推出的水位数字传感器，使用 UART 通信方式。本文档对 BM62S3202-1 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例使用 BM53A367A 模块，演示了读取水位值的功能。

适用型号：

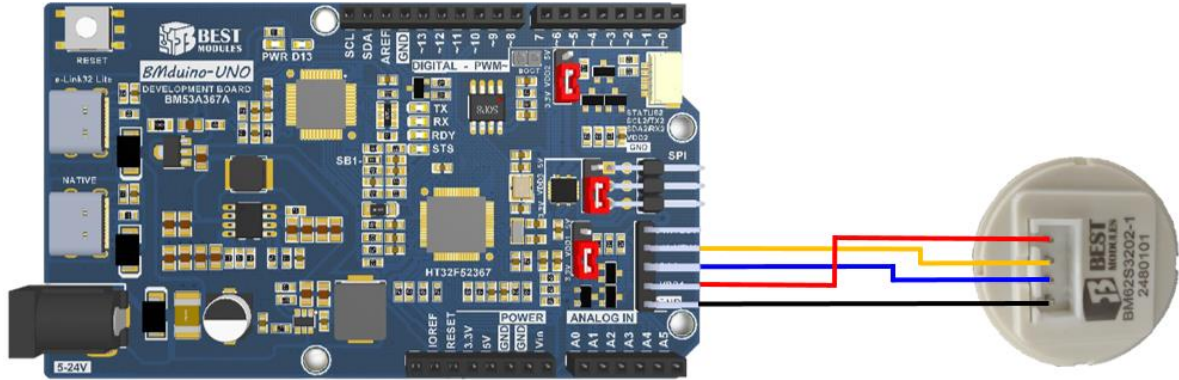
型号	说明
BM62S3202-1	水位数字传感器
BM53A367A	BMduino-UNO

BMduino Lib 函数

Arduino Lib 名称：		Lib 版本： V1.0.1	
读取函数			
1	void Read_WaterSensor_ID(uint8_t* ID)		
	描述	获取设备 ID	
	参数	存放设备 ID 的数组	
	返回值	—	
	备注	—	
2	uint16_t Read_WaterSensor_Value()		
	描述	获取水位数据	
	参数	—	
	返回值	水位值，单位 mmH2O	
	备注	—	
3	float Read_WaterSensor_Temperature()		
	描述	获取温度数据	
	参数	—	
	返回值	温度值，单位℃	
	备注	—	

BMduino 范例

范例 1: Read_WaterSensor_Value



实物连接示意图

范例实现功能：使用 UART 通信方式，每 1s 读取水位数据，并且显示在串口监视器上。

1. 范例使用方式：在 Arduino IDE 中编写 Read_WaterSensor_Value 函数，并进行调用
2. 示例说明：
 - a. 构建对象&初始化模块

```
#define Head 0xFA //水位传感器头码
#define Get_WaterLevel_CMD 0x01//获取水位数据命令
#define Get_Temperature_CMD 0x02//获取温度数据命令
#define Get_ID_CMD 0x05//获取设备 ID 命令

void setup()
{
  Serial.begin(9600); //初始化串口监视器
  Serial1.begin(38400); //初始化读取传感器串口
  while(!Serial); //等待串口初始化完成
  while(!Serial1); //等待串口初始化完成
  Serial.println("Water level digital sensor testing");//串口打印
  Serial.println("Get ID...");//串口打印
  uint8_t WaterSensor_ID[8]; //定义存放设备 ID 数组
  Read_WaterSensor_ID(WaterSensor_ID); //获取设备 ID
  Serial.print("ID:");//串口打印
  for(int i = 0; i < 8; i++) //串口打印设备 ID
  {
    Serial.print(WaterSensor_ID[i]);
    Serial.print(" ");
  }
}
```

```
Serial.println();//串口打印换行
}
void loop() {
    int WaterSensor_value = 0;//定义水位数据变量
    Serial.println("Get WaterLevel...");//串口打印
    WaterSensor_value = Read_WaterSensor_Value();//获取水位数据
    Serial.print("WaterLevel: ");//串口打印
    Serial.print(WaterSensor_value);//串口打印水位数据
    Serial.println("mmH2O");//串口打印水位数据单位
    delay(1000);//延时 1000ms
}
void Read_WaterSensor_ID(uint8_t* ID)
{
    uint8_t length = 0;//定义长度变量
    uint16_t checksum = 0;//定义校验和变量
    uint8_t TXdata[4];//定义发送数组
    TXdata[0] = Head;//水位数字传感器头码
    TXdata[1] = Get_ID_CMD;//水位数字传感器读设备 ID 命令
    TXdata[2] = length;//发送的数据长度
    checksum = TXdata[1] + TXdata[2];//计算校验和
    for(int i = 0; i < length; i++)
    {
        checksum = checksum + TXdata[3 + i];
    }
    TXdata[length + 3] = checksum;
    Serial1.write(TXdata, length + 4);//发送数据给水位数字传感器
    delay(10);//延时 10ms
    uint8_t RXdata[12];//定义接收变量
    if(Serial1.available())//判断串口是否有接收到数据
    {
        Serial1.readBytes(RXdata,12);//获取串口缓存数据
        checksum = 0;//校验和为 0
        for(int i = 0; i < 10; i++)//计算校验和
        {
            checksum += RXdata[1 + i];
        }
        if((checksum%256) == RXdata[11])//判断校验是否正确
        {
            for(int i = 0; i < 8; i++)//将接收到的 ID 数据进行返回
            {
                ID[i] = RXdata[i + 3];
            }
        }
    }
}
```

```
    }  
  }  
}  
  
uint16_t Read_WaterSensor_Value()  
{  
    uint8_t length = 0; //定义长度变量  
    uint16_t checksum = 0; //定义校验和变量  
    uint8_t TXdata[4]; //定义发送数据数组  
    TXdata[0] = Head; //水位数字传感器头码  
    TXdata[1] = Get_WaterLevel_CMD; //水位数字传感器读水位数据命令  
    TXdata[2] = length; //发送的数据长度  
    checksum = TXdata[1] + TXdata[2]; //计算校验和  
    for(int i = 0; i < length; i++)  
    {  
        checksum = checksum + TXdata[3 + i];  
    }  
    TXdata[length + 3] = checksum;  
    Serial1.write(TXdata, length + 4); //发送数据到水位数字传感器  
    delay(10); //延时 10ms  
    uint8_t RXdata[6]; //定义接收数据的变量  
    uint16_t value = 0;  
    if(Serial1.available()) //判断串口是否接收到数据  
    {  
        Serial1.readBytes(RXdata, 6); //读取串口的缓存数据  
        value = RXdata[4] * 256 + RXdata[3]; //将数据转换成水位数据  
        uint8_t checksum = RXdata[1] + RXdata[2] + RXdata[3] + RXdata[4]; //计算校验和  
        if(value > 32767){ value = 0; }  
        if(checksum != RXdata[5]) //判断校验是否正确  
        {  
            value = 0;  
        }  
    }  
    return value; //返回水位数据  
}  
  
float Read_WaterSensor_Temperature()  
{  
    uint8_t length = 0; //定义长度变量  
    uint16_t checksum = 0; //定义校验和变量  
    uint8_t TXdata[4]; //定义发送数据数组  
    TXdata[0] = Head; //水位数字传感器头码  
    TXdata[1] = Get_Temperature_CMD; //水位数字传感器读温度数据命令
```

```

TXdata[2] = length; //发送的数据长度
checksum = TXdata[1] + TXdata[2]; //计算校验和
for(int i = 0; i < length; i++)
{
    checksum = checksum + TXdata[3 + i];
}
TXdata[length + 3] = checksum;
Serial1.write(TXdata, length + 4); //发送数据到水位数字传感器
delay(10); //延时 10ms
uint8_t RXdata[6]; //定义接收数据的变量
float value = 0;
if(Serial1.available())//判断串口是否接收到数据
{
    Serial1.readBytes(RXdata,6); //读取串口的缓存数据
    value = RXdata[4] * 256 + RXdata[3]; //将数据转换成温度数据
    uint8_t checksum = RXdata[1] + RXdata[2] + RXdata[3] + RXdata[4]; //计算校验和
    if(value > 32767){ value = value - 65536; }
    value = value / 10.0;
    if(checksum != RXdata[5]) //判断校验是否正确
    {
        value = 0;
    }
}
return value;//返回温度数据
}

```

3. 打开串口监视器，波特率选择 9600，串口监视器显示如下：

