## **1. General Description**

The BoSL depth probe combines a number of CotS sensors to provide a low-cost low-power avenue to depth, temperature and EC measurement of water in a small package.

By combining the MS5803, DS18B20, and EC measurement it inherits the excellent characteristics of these sensors to deliver high quality measurements.

The sensor itself requires little external circuitry to run and has been specifically designed for quick installation with the BoSL Board datalogger. It is compatible Arduino based boards or other microcontrollers.

The predominantly digital data interface reduces the impact of noise interfering with signal quality over longer cables.

Robust waterproofing enables the sensor to last in challenging environments for extended periods of time, while still being easy to install and adaptable for a wide range of installation scenarios.

The probes quick conversion time on all three measurement types allow it to quickly return to a low power sleep mode minimising battery use and enabling logging times of over a year when combined with appropriate dataloggers.

## **2. Applications**

* Storm water drains
* Water flows

## **3. Features**

* Depth measurement
  + cm resolution
  + differential pressure
* Temperature measurement
  + 0.5°C accuracy
* EC measurement
* -- μA sleep current
* 3.3V operation

## **4. Device Drawing**

Add Drawing Here

## **5. Pin Description**

The BoSL depth probe is terminated either with an RJ-45 T-586B wiring or a free CAT5 cable.

|  |  |  |
| --- | --- | --- |
| Pin | | Description |
| Name | Colour |
| ECP | white/orange | Power for EC measurement |
| VCC | orange | Input supply voltage |
| NC | white/green | No connect |
| SCL | blue | MS5803 I2C serial clock line |
| DQ | white/blue | DS18B20 1-wire data input/output |
| GND | green | Ground |
| ECD | white/brown | EC reading data output |
| SDA | brown | MS5803 I2C serial data |

## **6.1 Absolute Maximum Ratings**

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Min | Max | Unit |
| Voltage on any pin relative to ground | -0.3 | 4.0 | V |
| Water submersion depth |  | 100 (from ms5803, waterproofing of case undetermined) | m |

## **6.2 Recommended Operating Conditions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Min | Typ | Max | Unit |
| VCC |  | 3.3 |  | V |
| Water submersion depth |  |  | 10 | m |
| EC voltage divider resistor resistance (RD) |  | 100?? |  | Ω |
| Voltage on ECP |  | 3.3 |  | V |
| ECD output ADC bit depth |  | 10 |  | bits |
| I2C clock frequency |  |  | 400 | kHz |
| Exterior cable length |  | 10 | 30 | m |

## **6.3 Electrical Characteristics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Min | Typ | Max | Unit |
| Active current (IActive) |  |  |  | mA |
| Sleep current (ISleep) |  |  |  | μΑ |
| I2C address |  | 0x76 |  |  |

## **7. Operation**

### **Temperature** **Measurement**

Temperature measurement is facilitated by the DS18B20 over the 1-wire interface. The DS8B20 is wired directly to the pins VCC, GND, DQ. A 4.7 kΩ resistor connected between DQ and VCC.

Communicating with the sensor requires a one-wire interface compatible device. A list of commands for the sensor can be found on the DS18B20 datasheet [1]. For connecting the device to an Arduino or Arduino compatible datalogger such as the BoSL board it is recommended that the Dallas Temperature Arduino Library is used [2].

### **Depth** **Measurement**

The depth measurement is calculated from the pressure differential between an external known atmospheric pressure value, and the reading from the MS-5803. It is recommended that the BoSL Water Probe is located as close as possible to the bottom of the water flow for accurate depth measurements. The MS5803 is wired directly to the pins VCC, GND, SCL, SDA. It is set to I2C communication mode with device address 0x76. Pull up resistors with resistance 4.7 kΩ are connected to the SCL and SDA lines, eliminating the need for these to be added externally. The MS-5803 contains its own temperature sensor which can be read from however it is strongly recommended that the DS18B20 is used as it has much less thermal resistance to the surrounding water.

Communicating with the sensor requires a I2C interface compatible device. A list of commands for the sensor can be found on the MS5803 datasheet [3]. For connecting the device to an Arduino or Arduino compatible datalogger such as the BoSL board it is recommended that the SparkFun MS5803-14BA Breakout Arduino Library is used [4].

### **EC** **Measurement**

The principal of measurement for delivering EC readings is measuring the resistance across a pair of submerged pins. The two pins are connected directly to ECP and ECD. This allows for end user flexibility on the resistance measurement technique.

A calibration curve is required to convert the resistance reading between the EC pins into a conductance measurement for the water, however if this is not required in the application then that the resistance is inversely proportional to the conductivity can be used.

An implementation of EC measurement is shown in [10. Typical Application](#_10._Typical_Application)

## **8. Sleep Mode**

Sleep mode on the BoSL depth probe greatly reduce the current consumption and allow the sensor to when paired with a suitable datalogger to log for periods of up to years without needing batteries to be replaced.

To enable sleep mode no commands should be sent to the DS18B20 or the MS5803. The ECP and ECD pins should be put into a high impedance state. If this is not possible then ECP should be pulled to ground.

To wake from sleep ECP and EDC should be returned to their normal state, commands can be sent to the DS18B20 or the MS5803 to wake them.

Each of the EC, DS18B20, and MS5803 can be put to sleep independently.

## **9. Physical Overview**

Dimensioned Labelled Drawing

## **10. Typical Application**

RD = 100 Ω

**BoSL Board Integration**

The wiring diagram and code snippets below can be used to achieve full functionality of the BoSL depth probe. Measurements of the water pressure, temperature, and EC can be called from functions in software, with their values returned for local logging or remote transmission.

Depth Probe

BoSL Board

A1

A0

RJ45 Link

In this design a MS-5803 located on the BoSL board is used to capture the current air pressure so water depth can be calculated via: D = (Pprobe – Pair)/(ρwaterg). A voltage divider is used to measure the resistance between the EC pins. RD is chosen to be 100 Ω for this as it makes the range of EC values commonly expected in water systems to be easily readable. The resistance between the EC pins is given by REC = RD (VECD/VCC - 1).

Code for BoSL board integration. Provides methods for reading temperature, pressure, and EC which can be implemented into further logging code. The BoSL depth probe will return to sleep automatically after a measurement is taken. Please ensure that you have the latest versions of recommended libraries [2] [4].

//include libraries for DS18B20 and MS5803  
#include <OneWire.h>  
#include <DallasTemperature.h>  
#include <SparkFun\_MS5803\_I2C.h>  
  
MS5803 ProbeMS5803(ADDRESS\_HIGH); //define depth probe MS5803 object  
MS5803 AirMS5803(ADDRESS\_LOW); //define bosl board MS5803 object

OneWire oneWire(A2); //define 1-wire object on A2 for DS18B20  
DallasTemperature ProbeDS18B20(&oneWire); //define DS18B20 object

void setup() {  
 pinMode(A0,INPUT);//configure pin to read from ECD  
 pinMode(A1,OUTPUT);  
 digitalWrite(A1,LOW);//ground one side of EC voltage divider  
}  
//function to return raw EC reading  
int readEC(){  
 digitalWrite(A3,HIGH);//power ECP to take out of sleep mode  
 analogRead(A0);  
 int ec = analogRead(A0); //read EDC twice to charge ADC capacitor  
 digitalWrite(A3,LOW);//return EC to sleep mode  
 **return** ec  
}  
//function to read water pressure from probe  
float readProbePressure(){  
 ProbeMS5803.reset();  
 ProbeMS5803.begin(); //initialise MS5803  
 **return** ProbeMS5803.getPressure(ADC\_4096);//get and return pressure  
}  
//function to read air pressure from BoSL board  
float readAirPressure(){  
 AirMS5803.reset();  
 AirMS5803.begin(); //initialise MS5803  
 **return** AirMS5803.getPressure(ADC\_4096);//get and return pressure  
}  
//function to read temperature from DS18B20  
float readTemprature(){  
 ProbeDS18B20.begin();  
 ProbeDS18B20.requestTemperatures();// initialise DS18B20  
 **return** ProbeDS18B20.getTempCByIndex(0);//get and return temperature  
}

## **11. Known Issues**

* Low resistance between the EC pins may cause the temperature reading from the MS-5803 to go negative.

## **12. Ordering Information**

# **13.** **References**

|  |  |
| --- | --- |
| [1] | MAXIM Intergrated, “DS18B20,” June 2019. [Online]. Available: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf. |
| [2] | M. Burton, “Arduino Library for Maxim Temperature Integrated Circuits,” Git Hub, [Online]. Available: https://github.com/milesburton/Arduino-Temperature-Control-Library. [Accessed 16 August 2020]. |
| [3] | TE Connectivity, “MS5803-14BA,” August 2017. [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=MS5803-14BA&DocType=Data+Sheet&DocLang=English. |
| [4] | SparkFun, “SparkFun\_MS5803-14BA\_Breakout\_Arduino\_Library,” SparkFun, [Online]. Available: https://github.com/sparkfun/SparkFun\_MS5803-14BA\_Breakout\_Arduino\_Library. [Accessed 16 August 2020]. |

## **14. Revision Information**

|  |  |  |
| --- | --- | --- |
| Revision Date | Description | Pages Changed |
| 16/08/2020 | Initial release |  |