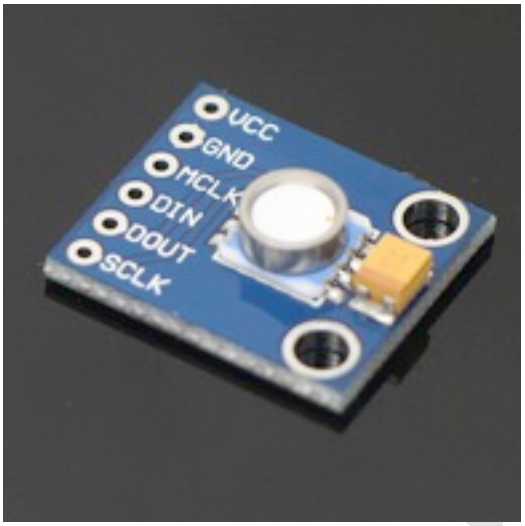
Water Depth (Level) Sensor



Description:

This is an amazing digital pressure sensor from Intersema has wide range applications.The MS5540C pressure sensor measures Absolute water or air pressure. It also measure temperature at the same time. So you can use it for measuring water depth in sea or water level in tank. It can be also be used with different fluids measurements such as oil. You might use it also with air for measuring altitude (barometer) and temperature.

The MS5540C carries a metal protection cap filled with silicone gel for enhanced protection against water and humidity. The properties of this gel ensure function of the sensor even when in direct water contact. The MS5540C is qualified referring to the ISO Standard 2281 and can withstand a pressure of 100 m in salt water. Nevertheless the user should avoid drying of hard materials like for example salt particles on the silicone gel surface. In this case it is better to rinse with clean water afterwards.

The sensor measure water depth with very good accuracy of 0.1 mbar (1 cm water level). Please read note 3 below for linear range and accuracy. The pressure module have wide range applications ranging from liquid level measurements in tanks, water depth and temperature in diving, snorkeling and ROV.

The pressure sensor is factory calibrated and future electronics Egypt has extensively tested the sensor and prepared a tutorial (check below) showing how to read factory calibration coefficients and use it to measure both pressure and temperature in an accurate manner.

Features:

- Pressure Linear range (air Applications): 10 to 1100 mbar
- Under water depth: 10m (Please read note 3 below)
- Resolution 0.1 mbar (1 cm water)
- 16 Bit ADC
- SPI serial interface
- 1 system clock line (32.768 kHz)
- Temperature range -40..+85°C
- Low voltage (2.2 to 3.6V)
- Low power (Standby current:0.1uA)
- Small size 6.2 x 6.4 mm

Important Notes:

1- When using the pressure sensor for underwater applications, user MUST isolate all the metal contacts (For example using melted wax), the only part that is left exposed to water is the sensor surface (the white gel surface should not be covered or touched).

2- Special care has to be taken to not mechanically damage the gel. Damaged gel could lead to air entrapment and consequently to unstable sensor signal. Do not test the sensor by pressing the surface with your finger or any hard metal

3- The sensor can withstand up to 100 meter under water, however the linear range for factory calibrated coefficients stored on the sensor ROM adjusted for linear range of about 7m. If you want more linear range you may calibrate your own factors and use each group of factors for each range.

Water Depth Sensor connection to Arduino and Code

Sensor **Arduino**

VCC ---------- >3.3v

GND ---------- >GND

DIN (MOSI) ----------> pin (11)

DOUT (MISO) ---------> pin (12)

SCLK ----------> pin (13)

MCLK ----------> pin (9)

Arduino code:

```
#include <SPI.h>
 int clock = 9;


void resetsensor() //this function keeps the sketch a little shorter
{
  SPI.setDataMode(SPI_MODE0);
  SPI.transfer(0x15);
  SPI.transfer(0x55);
  SPI.transfer(0x40);
}

void setup() {
  Serial.begin(9600);
  SPI.begin(); //see SPI library details on arduino.cc for details
```

```
  SPI.setBitOrder(MSBFIRST);
  SPI.setClockDivider(SPI_CLOCK_DIV32); //divide 16 MHz to communicate on 500 kHz
  pinMode(clock, OUTPUT);
  delay(100);
}

void loop()
{
  TCCR1B = (TCCR1B & 0xF8) | 1 ; //generates the MCKL signal
  analogWrite (clock, 128) ;

  resetsensor();//resets the sensor - caution: afterwards mode = SPI_MODE0!

  //Calibration word 1
  unsigned int word1 = 0;
  unsigned int word11 = 0;
  SPI.transfer(0x1D); //send first byte of command to get calibration word 1
  SPI.transfer(0x50); //send second byte of command to get calibration word 1
  SPI.setDataMode(SPI_MODE1); //change mode in order to listen
  word1 = SPI.transfer(0x00); //send dummy byte to read first byte of word
  word1 = word1 << 8; //shift returned byte
  word11 = SPI.transfer(0x00); //send dummy byte to read second byte of word
  word1 = word1 | word11; //combine first and second byte of word

  resetsensor();//resets the sensor

  //Calibration word 2; see comments on calibration word 1
```

```
unsigned int word2 = 0;
byte word22 = 0;
SPI.transfer(0x1D);
SPI.transfer(0x60);
SPI.setDataMode(SPI_MODE1);
word2 = SPI.transfer(0x00);
word2 = word2 <<8;
word22 = SPI.transfer(0x00);
word2 = word2 | word22;

resetsensor();//resets the sensor

//Calibration word 3; see comments on calibration word 1
unsigned int word3 = 0;
byte word33 = 0;
SPI.transfer(0x1D);
SPI.transfer(0x90);
SPI.setDataMode(SPI_MODE1);
word3 = SPI.transfer(0x00);
word3 = word3 <<8;
word33 = SPI.transfer(0x00);
word3 = word3 | word33;

resetsensor();//resets the sensor

//Calibration word 4; see comments on calibration word 1
unsigned int word4 = 0;
```

```
byte word44 = 0;
SPI.transfer(0x1D);
SPI.transfer(0xA0);
SPI.setDataMode(SPI_MODE1);
word4 = SPI.transfer(0x00);
word4 = word4 <<8;
word44 = SPI.transfer(0x00);
word4 = word4 | word44;


long c1 = word1 << 1;
long c2 = ((word3 & 0x3F) >> 6) | ((word4 & 0x3F));
long c3 = (word4 << 6) ;
long c4 = (word3 << 6);
long c5 = (word2 << 6) | ((word1 & 0x1) >> 10);
long c6 = word2 & 0x3F;


resetsensor();//resets the sensor


//Temperature:
unsigned int tempMSB = 0; //first byte of value
unsigned int tempLSB = 0; //last byte of value
unsigned int D2 = 0;
SPI.transfer(0x0F); //send first byte of command to get temperature value
SPI.transfer(0x20); //send second byte of command to get temperature value
delay(35); //wait for conversion end
SPI.setDataMode(SPI_MODE1); //change mode in order to listen
tempMSB = SPI.transfer(0x00); //send dummy byte to read first byte of value
```

```
tempMSB = tempMSB << 8; //shift first byte
tempLSB = SPI.transfer(0x00); //send dummy byte to read second byte of value
D2 = tempMSB | tempLSB; //combine first and second byte of value


resetsensor();//resets the sensor


//Pressure:
unsigned int presMSB = 0; //first byte of value
unsigned int presLSB =0; //last byte of value
unsigned int D1 = 0;
SPI.transfer(0x0F); //send first byte of command to get pressure value
SPI.transfer(0x40); //send second byte of command to get pressure value
delay(35); //wait for conversion end
SPI.setDataMode(SPI_MODE1); //change mode in order to listen
presMSB = SPI.transfer(0x00); //send dummy byte to read first byte of value
presMSB = presMSB << 8; //shift first byte
presLSB = SPI.transfer(0x00); //send dummy byte to read second byte of value
D1 = presMSB | presLSB;


const long UT1 = (c5 * 8) + 20224;
const long dT =(D2 - UT1);
const long TEMP = 200 + ((dT * (c6 + 50))/1024);
const long OFF  = (c2*4) + (((c4 - 512) * dT)/4096);
const long SENS = c1 + ((c3 * dT)/1024) + 24576;


long PCOMP = ((((SENS * (D1 - 7168))/16384)- OFF)/32)+250;
float TEMPREAL = TEMP/10;
```

```
  Serial.print("pressure =    ");
  Serial.print(PCOMP);
  Serial.println(" mbar");


  const long dT2 = dT - ((dT >> 7 * dT >> 7) >> 3);
  const float TEMPCOMP = (200 + (dT2*(c6+100) >>11))/10;
  Serial.print("temperature = ");
  Serial.print(TEMPCOMP);
  Serial.println(" °C");
  Serial.println("**********************************");


  delay(1000);
}
```