

I2C soil moisture sensor

The sensor can be read via I2C protocol and provides these features:

- Soil moisture sensing
- Light sensing
- Temperature sensing
- Reset chip
- I2C address change
- Deep sleep

Please checkout FAQ below before asking questions, if you still need help, please don't hesitate to contact me.

Technical data

- Version 2.7.5
- Supply voltage 3.3V - 5V
- Current consumption: 1.1mA @ 5V, 0.7mA @ 3.3V when idle, 14mA @ 5V, 7.8mA @ 3.3V when taking a measurement. When constantly polling sensor at full speed, current consumption averages to 4.5mA @ 5V, 2.8mA @ 3.3V
- Operating temperature 0°C - 85°C
- Moisture reading drift with temperature - <10% over full temp range
- Don't forget to provide pullups for SCL and SDA lines
- Default I2C address is 0x20 (hex)
- To read soil moisture, read 2 bytes from register 0
- To read light level, start measurement by writing 3 to the device I2C address, wait for 3 seconds, read 2 bytes from register 4
- To read temperature, read 2 bytes from register 5
- To change the I2C address of the sensor, write a new address (one byte [1..127]) to register 1; the new address will take effect after reset
- To reset the sensor, write 6 to the device I2C address.
- **Do not hotplug** the sensor into the active I2C bus - address change command has no protection and this might result in a random number set as an address of the sensor. Use I2C scan sketch to find out the address if the sensor stops responding with proper values.

More documentation and example code is available on [github page](#). Also check out the [FAQ](#)

The easiest way to get the sensor working on Arduino compatible environment is to use the [library by Ingo Fischer](#) it is available thru Arduino environment, make sure to use version 1.1.4 or later as it contains some fixes for address change and bus hang bugs fixed in the firmware version 2.6.

How to interpret the readings

Both light and moisture sensors give relative values. Meaning, more moisture will give you higher reading, more light, lower reading.

Moisture is somewhat linear. I test all sensors before shipping and they give about 290 - 310 in free air at 5V supply.

I didn't measure linearity of the light sensor, it gives 65535 in a dark room away from desk lamp. When it's dark, it takes longer to measure light, reading the light register while measurement is in progress will return the previous reading. Be aware, light sensor is pretty noisy.

Temperature is measured by the thermistor on the body of the sensor. Calculated absolute measurement accuracy is better than 2%. The returned value is in tenths of degrees Celsius. I.e. value 252 would mean 25.2°C.

Note Upon reading the moisture or temperature value, a value from the previous read command is returned and the new measurement is started. If you do rare measurements and want to act instantly, do two consecutive readings to get the most up to date data. Also you can read GET_BUSY register via i2c - it will indicate when the measurement is done. Basically the process goes like this: read from GET_CAPACITANCE, discard results, then read from GET_BUSY until you get '0' as an answer, then read from GET_CAPACITANCE again - the returned value is the soil moisture NOW.

Controller support

The sensor works fine with Arduino and RaspberryPi. Examples are available on [github](#) page.

- Arduino library: <https://github.com/Apollon77/I2CSoilMoistureSensor>
- Raspberry Pi library: <https://github.com/ageir/chirp-rpi>
- Particle proton library: <https://github.com/VintageGeek/I2CSoilMoistureSensor>
- Micropython library: <https://github.com/scopelemanuele/pyChirpLib>

Thanks guys for writing this software, open source yay!

Note for ESP8266 based systems

In some cases the default ESP8266 Arduino I2C library has the clock stretching timeout set too low. If you experience intermittent communication, add this to your code: `Wire.setClockStretchLimit(4000)`

Waterproofing

The sensor comes coated with PRF202 - a moisture resistant varnish for electronics. It's ok for play around in a flower pot but not enough for outdoor use. **You must add an additional protection to the whole sensor after soldering cable to it!** Some suggestions on making the sensor more robust after attaching the cable:

- Polyester or epoxy resin - this method is the most bullet proof as the resin is totally resistant to the water. On the par side, note that sensitivity of the sensor will decrease depending on how thick the layer you are going to apply. Also applying the resin in uniform manner presents some challenge.
- PlastiDip - some of my customers have tried this method - easy to apply by spraying and not a lot of loss of sensitivity.
- Rubber balloon - yes, just roll a long balloon over the sensor :)

Be sure to coat the whole thing - the sensor part, the electronics and the cable connection itself so no bare copper or solder is accessible to the water.

Rugged version

Pre-coated version is available since December 2016. The sensor is coated in epoxy resin, cured and additionally protected by adhesive-lined heat shrink. The 1m (3 feet) long cable is pre-soldered. **Light sensor is covered by heatshrink**, so the light reading will always report total darkness.

The pinout of the rugged version (with white cable):

- RED - VCC
- BLACK - GND
- BLUE or GREEN- SDA
- YELLOW - SCK

Shield is not grounded on the sensor end and must be grounded on the master end.

Stuff people did with it

- A review of several cheap soil moisture sensors available on the market: [Testing Capacitive soil moisture sensors](#)
- A [video from Growing Robot](#) about using the sensors with Raspberry Pi
- A [complete Open Source logging solution](#) employing a batch of 40 sensors in Crete * [Smart and Safe Outdoor Plant Watering System](#)
- [Smart Watering Hardware](#)
- [Waterproof holder for I2C Soil Moisture Sensor](#)
- [Weather / Watering Cover for Chirp! Plant Alarms & Sensors](#)

Drop me a message if you want your project listed here.

Frequently asked questions

Do you have the exact dimensions, or even better a CAD drawing of the sensor in 3D?

Sensors are 18mm x 149mm. Zipped step file is available in github repository.

How to set up multiple sensors connected to one Arduino Uno?

You can connect several sensors to a single I2C bus. Exact maximum amount depends on many factors, mostly on the wiring length. To have several sensors working on a bus, you have to assign an individual address to each sensor. To do that, connect each sensor one by one and run an address change routine. Code is available as [an example](#) of Arduino library. Raspberry Pi example is also [available](#).

Can I use a X meter long cable without affecting the sensor measurements?

It depends. Note that I2C bus is not ment to run on cables, so it's a hack already. Nevertheless, a lot of products do this. I suggest reading online resources about running I2C on long cables to get some background information, but generally, to support long cables and ensure robust communication you can do the following: * lower pullup resistor value, calculation is a bit complicated, but it seems you can go as low as 1k. More info [here](#). * slow down I2C speed. Consult your controller board documentation on how to do that * Use shielded cable

How to solder wire to the sensor?

Do not solder wires to the 6 pad connector - this is a programming header. Solder to 4 pads marked VCC, SCL, SDA, GND at the edge of the board. I suggest using a shielded 4 wire cable designed for security signalization installations. Do not solder the shield to the sensor, but ground it at the controller board side. Fix cable to the board using zip tie. Cover the wire connection and SMD parts with **glue-lined** heatshrink. Do not use a simple heatshrink as it will eventually let water in and will not allow it to evaporate resulting in short sensor lifetime.

Do you offer the source code from the i2c chip project? Can I reflash firmware of the sensor?

Yes, sure, the sensor is open source hardware, all the code and hardware files are available on the [github page](#). You can use 6 pad header to program it using any AVR programming hardware like TinyISP, UsbAsp and similar. Also, you can use

your [Arduino board as an AVR ISP programmer](#).

Communication with ESP8266-based systems is unreliable, sensor sends data then stops working.

In depth technical analysis is [available here](#).

Add this call after you initialize I2C bus: `Wire.setClockStretchLimit(4000);`

How durable are non-rugged sensors?

Standard solder mask seems to last about 3 months before flaking off. It really depends on the type of your soil and how often do you remove and insert the sensor as scratches will damage the surface and shorten the lifespan. I have had sensors working flawlessly for over a year in the flower pot at home when not disturbed. Note that you **must** protect sensor electronics with **glue-lined** heatshrink if you are going to install it outdoors. I use [this heatshrink](#) but you can use any similar one.

Can you set different address for each sensor before shipping?

Yes I can for orders of 50pcs or more, please drop me an email - mic at wemakethings net

Rugged sensors

How do I connect the sensor to my board?

Wire colors go like this: * RED - VCC * BLACK - GND * BLUE or GREEN- SDA * YELLOW - SCK

You can leave the metal sheet and metal wire unconnected for a quick test, but connect it to GND for proper installation.

Please check if your controller board has onboard pullup resistors for I2C. You need to add pull up resistors to both SCL and SDA lines if they are not there. Raspberry Pi has pullups onboard, Arduino UNO does not.

What is the expected life of the rugged sensor for continuous use outdoor?

I still don't have enough data to answer this question. Rugged sensors should be in practice indestructable, but failures sometimes happen. If you are using rugged sensor please let me know about the failures you experience.

Light sensor of rugged variant?

Light sensing does not work for rugged sensors as light sensor is covered by heatshrink.