# Instructions:



## PLANT MONITOR



Instructions version 1b.

# TABLE OF CONTENTS

# WARNING

Only the prong of the Plant Monitor below the white line should be allowed to get wet. If the top of the board gets wet, disconnect it from everything, dry it using a paper towel and then leave it the throughly dry out before trying to use it again.

# INTRODUCTION

The MonkMakes Plant Monitor measures soil moisture, temperature and relative humidity. This board is compatible with the BBC micro:bit, Raspberry Pi and most microcontroller boards.

- Superior capacitative sensor (no electrical contact with soil)
- Alligator / crocodile clip rings (for use with BBC micro:bit and Adafruit Clue etc.
- Ready soldered header pins for Arduino and other microcontroller boards.
- Easy to use UART serial interface
- Additional analog output for moisture only
- Built-in RGB LED (switchable)

In the following sections you will find instructions for using this board with:

- The BBC micro:bit (Makecode)
- The Raspberry Pi 4 (Python)
- The Raspberry Pi Pico (MicroPython)
- Arduino (Arduino C - SoftSerial)

# USING THE PLANT MONITOR

The plant monitor should be placed as shown below.



The front side of the prong should be as close to the edge of the pot as possible. The sensing all takes place from the far side of the prong.

The electronics should be facing out of the pot and the prong of the Plant Monitor pushed into the dirt as far as the white line (but no deeper).

It's a good idea to attach the wires you are going to use to connect to the Plant Monitor before positioning it in the plant pot.

Once powered up, the plant monitor will immediately start displaying the level of wetness using the builtin LED. Red means dry, green means wet. Before you put the Plant Monitor in the pot, try gripping the prong in your hand and the moisture of your body should be enough to alter the LED's color.

To make full use of wetness, temperature and humidity readings form the Plant Monitor, connect it up as shown below.
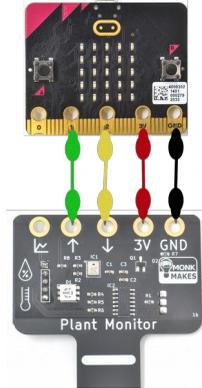


The red and black leads provide power from the micro:bit to the Plant Monitor and the yellow and green leads send serial data to and from the Plant Monitor.

There is a Makecode extension for the Plant Monitor that makes it easy to write code and the best way to get started is to load the example project from here:

https://makecode.microbit.org/_XK0cdj3u49Yg

When you download this project onto your micro:bit it will show the wetness using the bar graph block. Pressing button A will show the temperature in degrees C and pressing button B, the relative humidity as a percentage.

Here's the code:



The on start block must contain the block Start Plant Monitor, that initiates

communication between the micro:bit and the Plant Monitor. The variable show_wetness is used to switch the displaying of the bar graph on and off, otherwise it will try and draw the bar graph over the top of the temperature and humidity readings.

The forever block uses the block Plant Wetness and displays it as a bar graph as long as show_wetness is true.

When button A is pressed, the screen is cleared and show_wetness set to false and then the temperature is displayed. After a pause for you to read the temperature, show_wetness is set to true again so that the bar graph continues refreshing.

The button B code is much the same, except that the relative humidity is displayed rather than the temperature.

You can see the Plant Monitor extension in the Palette – it has an icon of a tree.
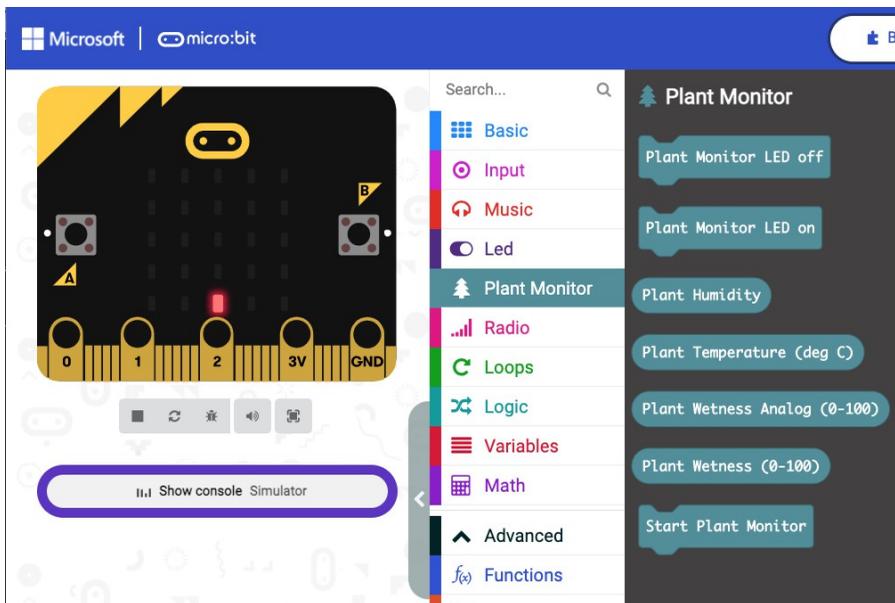


As well as the Start Plant Monitor, Plant Humidity and Plant Temperature blocks, there are also blocks that turn the LED on and off.

The Plant Wetness Analog block allows you to measure the wetness using the Plant Monitor's analog output pin. This is useful if you want to use the Makecode plotting facility, as this is not possible using the serial interface.

To use the analog interface, connect your micro:bit like this:

As before, power is supplied by the micto:bit, but this time, micro:Bit pin 0 is used to connect to the analog output of the Plant Monitor.

Connecting the Plant Monitor in this way only measures wetness, but it does mean that because the serial interface of the micro:bit is not used, it can be used to send serial data back to Makecode's plotting feature.

Try out the project here:
https://makecode.microbit.org/_YEL61KA0cKDd

The code for this is really quite minimal.



If you have your micro:bit paired to the Makecode editor in your browser, then as soon as the program has uploaded, you should see an option Show Console Device appear. Clicking on this will show a real-time plot of the wetness reading.

# RASPBERRY PI

Connect your Raspberry Pi to the Plant Monitor using female to female jumper wires as shown below:
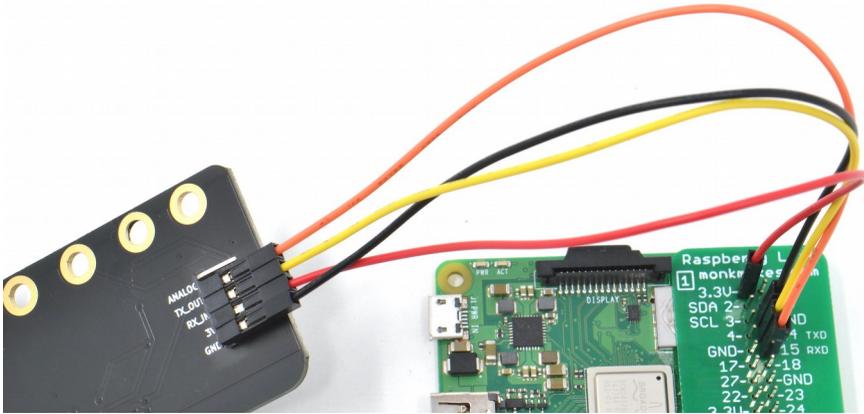


The connections are:

- GND to GND
- 3.3V on the Raspberry Pi to 3V on the Plant Monitor
- 14 TXD on the Raspberry Pi to RX_IN on the Plant Monitor
- 15 RXD on the Raspberry Pi to TX_OUT on the Plant Monitor

**WARNING: Do not connect this board to the 5V pin of the Raspberry Pi. The board is designed to operate at 3.3V and a 5V supply is likely to destroy it.**

Since the Raspberry Pi does not have analog inputs, then the only interface option is to use the serial UART interface. This interface has to be enabled on the Raspberry Pi from the Raspberry Pi Configuration tool that you will find in the Preferences section of the Start menu.
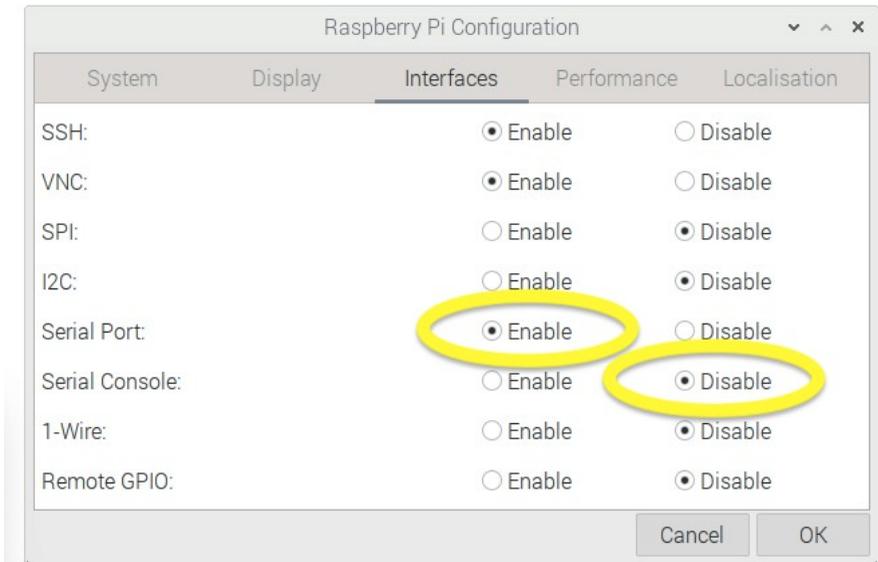
Enable the Serial Port and Disable the Serial Console. You may be prompted to restart your Pi for these changes to take effect.

To download the example programs for the Plant Monitor open a terminal window and enter the command:

```
$ git clone https://github.com/monkmakes/pmon.git
```

This will download all the example programs for various different platforms into a folder called pmon, so change to the right directory for the Raspberry Pi examples by entering the command:

```
$ cd pmon/raspberry_pi
```

Before you can run the examples, you will need to install the GUIZero library with the command:

```
$ pip3 install guizero
```

You can now run the examples. The first example 01_meter.py displays the wetness level, the temperature and humidity level. Run it using the command below.

```
$ python3 01_meter.py
```

.. and the following window should appear.
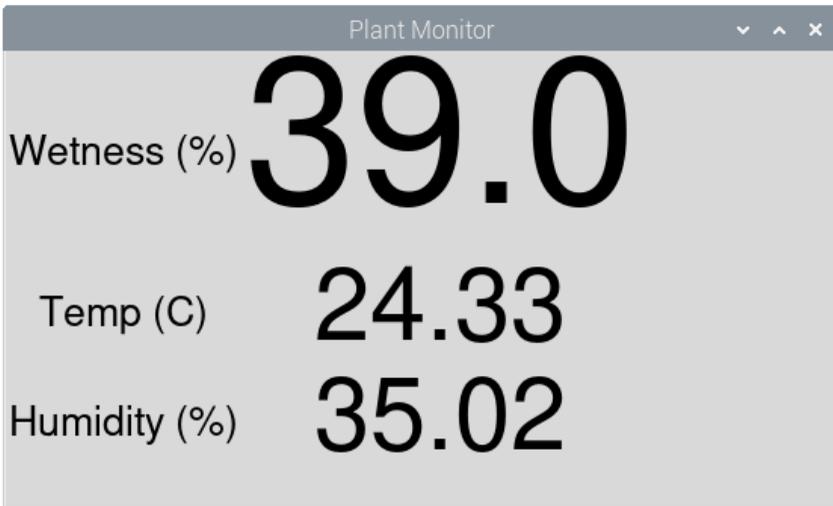


Try holding the prong of the moisture meter and you should see the wetness % rise. Similarly putting your finger over the metal box on the Plant Monitor that is the temperature and humidity sensor will change both readings.

The second example (02_data_logger.py) is a data logger that records all three values periodically and put them into a file that you can then import into a spreadheet.

Run the program as shown below and collect some data. You might like to put the Plant Monitor into a plant pot (see page 4) and record readings every minute for 24 hours (but perhaps start with a smaller set of data).

```
$ python3 02_data_logger.py
Enter interval between readings (seconds):60
Enter filename:test.txt
Logging started at: 2022-05-05 14:54:33
Press CTRL-c to end logging
time(s)        wetness        temp(C)        humidity(%)

0        0.0        23.9        36
5        0.0        23.88        35
10        52.0        23.87        35
15        37.0        23.82        36
20        36.0        24.58        45
25        93.0        25.43        36
^C
Logging to file test.txt complete
```

When you have got enough data, CTRL-c the program. If you don't have a spreadheet installed on your Raspberry Pi, install LibreOffice using the Recommended Software tool in the Preferences section of the Start menu. Run LibreCalc and then import the file of data.
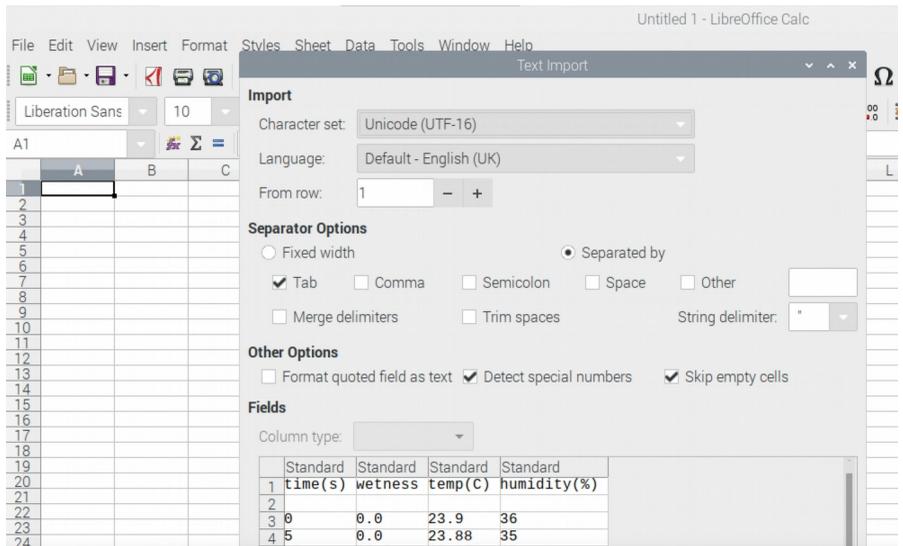


You can draw a chart of the data by selecting it (including the headings) and then inserting a Chart. Pick a chart type of XY.
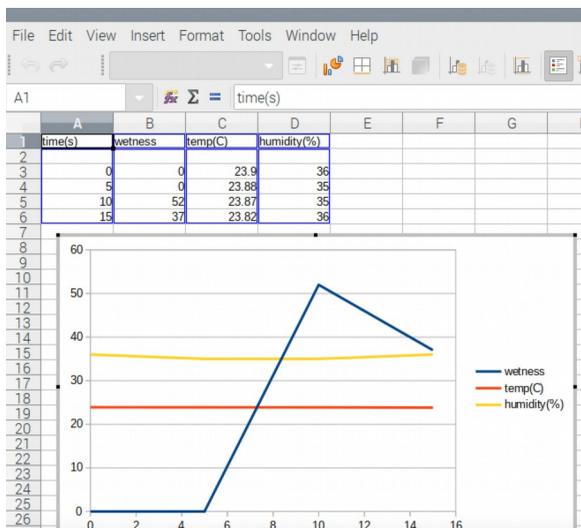
# RASPBERRY PI PICO

Connect the Plant Monitor to your Raspberry Pi Pico using solderless breadboard and female to male jumper wires as shown below, or if you prefer female to female jumper wires, directly from board to board..
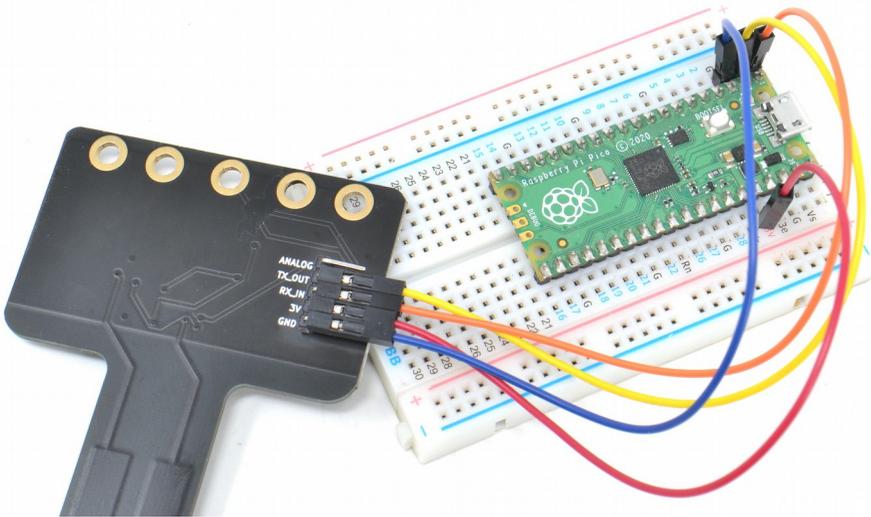


The connections are as follows:

- GND to GND

- 3V on the Pico to 3V on the Plant Monitor

- TX on the Pico to RX_IN on the Plant Monitor

- RX on the Pico to TX_OUT on the Plant Monitor

Using the MonkMakes Breadboard for Pico (https://www.monkmakes.com/pico_bb) will make it very much easier to identify which pin of the Pico is which.

To get you started, you will find a MicroPython library and test program in the examples for this board on its github page here: https://github.com/monkmakes/pmon

If you are unfamiliar with git, the easiest way to download it is to go to the github page above and then use the download ZIP feature (see below).
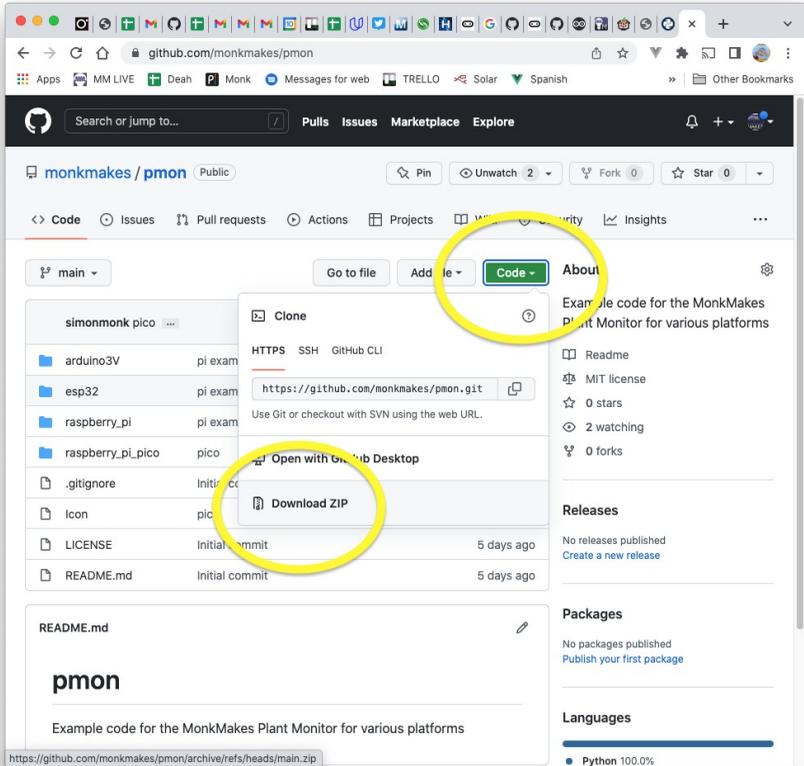
This download contains the example files for the Plant Monitor for lots of platforms, not just the Pico. So having downloaded the ZIP archive and extracted it, you will find it contains a folder called raspberry_pi_pico. Inside this folder you will find two files, that you should open using Thonny (https://thonny.org/).

- pmon.py – a MicroPython library for the Plant Monitor
- test.py – a test program using the library

Before you can run test.py you must copy the file pmon.py onto your Pico, by using the SaveAs menu option in Thonny and then selecting Raspberry Pi Pico for the destination. When you run test.py you should see the wetness, temperature and humidity being printed out.

Use the program test.py as a template for your own programs for the Pico.

Thonny - /Volumes/GoogleDrive/My Drive/Products/_GENERAL_ELECTRONICS/Plant_Monitor/code/pmon/raspberry_pi_pico/test.py @ 12 : 45

**pmon.py** | **test.py**

```python
1   from pmon import PlantMonitor
2   import time
3
4   time.sleep(2) # PlantMonitor startup time
5
6   pm = PlantMonitor()
7
8   while True:
9       w = pm.get_wetness()
10      t = pm.get_temp()
11      h = pm.get_humidity()
12      print("Wetness: {0} Temp: {1} Humidity: {2}".format(w, t, h))
13      time.sleep(1)
```

Object inspector

⇦ ⇨ | Data | Attributes

**Shell**

```
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
  Wetness: 1 Temp: 25.46 Humidity: 25.48
  Wetness: 0 Temp: 25.46 Humidity: 28.85
  Wetness: 0 Temp: 25.44 Humidity: 28.84
  Wetness: 0 Temp: 25.31 Humidity: 28.89
  Wetness: 0 Temp: 25.26 Humidity: 29.32
  Wetness: 0 Temp: 25.23 Humidity: 29.85
  Wetness: 55 Temp: 25.21 Humidity: 29.91
  Wetness: 55 Temp: 25.19 Humidity: 30.46
  Wetness: 61 Temp: 25.19 Humidity: 30.6
  Wetness: 61 Temp: 25.17 Humidity: 30.58
  Wetness: 60 Temp: 25.16 Humidity: 30.39
  Wetness: 60 Temp: 25.16 Humidity: 30.27
  Wetness: 59 Temp: 25.13 Humidity: 30.32
```

MicroPython (Raspberry Pi Pico)

# ARDUINO

Warning: The Plant Monitor is designed to operate at 3.3V, not the 5V that some Arduinos such as the Arduino Uno operate at. So, never power the Plant Monitor with 5V and make sure that none of it's input pins receive more than 3.3V.

To connect a 5V Arduino, such as the Arduino Uno or Leonardo you will need to use a level converter or a (as we have here) a 1kΩ resistor to limit the current flowing from the 5V Soft Serial transmit pin of the Arduino (pin 11) to the 3.3V RX_IN pin of the Plant Monitor.

Here's what this looks like, solderless breadboard is used to hold the resistor (in the middle of the breadboard), male to male jumper wires to connect the the Arduino to the breadboard and female to male jumper wires to connect the Plant Monitor to the breadboard. The connections are as follows:

- GND on the Arduino to GND on the Plant Monitor

- 3V on the Arduino to 3V on the Plant Monitor

- Pin 10 on the Arduino to TX_OUT on the Plant Monitor

- Pin 11 on the Arduino to RX_IN on the Plant Monitor **via a 1kΩ resistor**. Note that the resistor is not needed for a 3V Arduino.
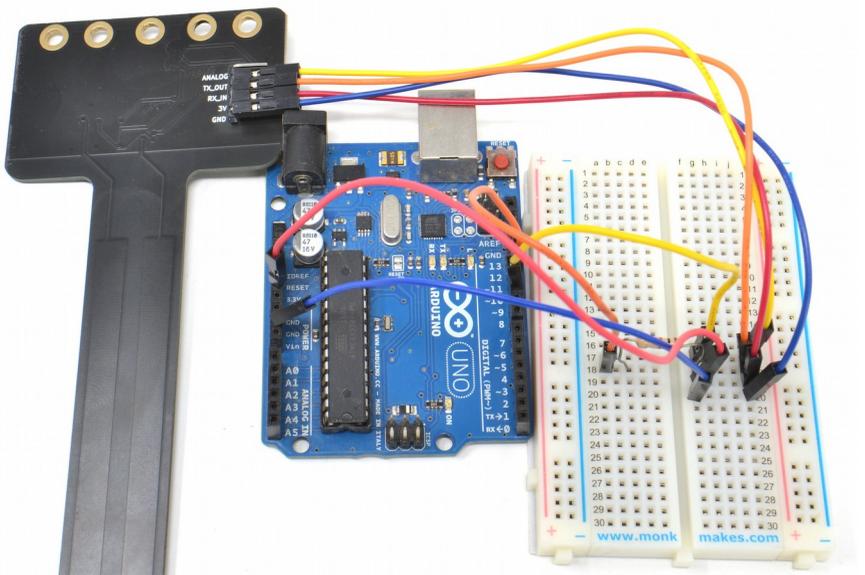
Once its all connected, you can install the Arduino library for the PlantMonitor by going to https://github.com/monkmakes/mm_plant_monitor and the from the Code menu, select Download ZIP.
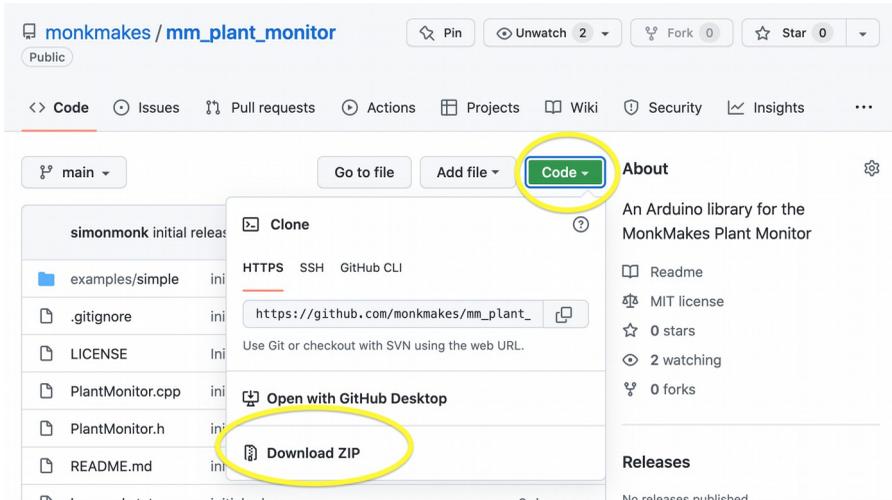


Now open the Arduino IDE and from the Sketch menu select the option to Add .ZIP Library and navigate to the ZIP file you just downloaded.
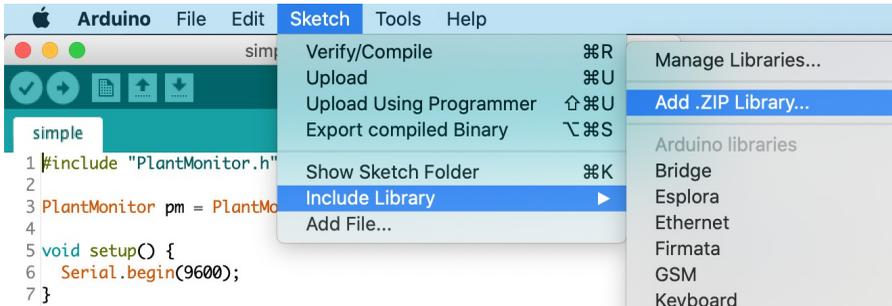


As well as installing the library, this will also fetch an example program that you will find in the Examples sub-menu of the File menu, under the category Examples from Custom Libraries.

Upload the example called *Simple* to your Arduino and then open the Serial Monitor. Here, you will see a series of readings. You can also turn the Plant Monitor's LED on and off from the Serial Monitor by sending serial commands. Type L in the send area of the Serial Monitor and then press the Send button to turn the LED on, and l (lower-case L) to turn the LED off.
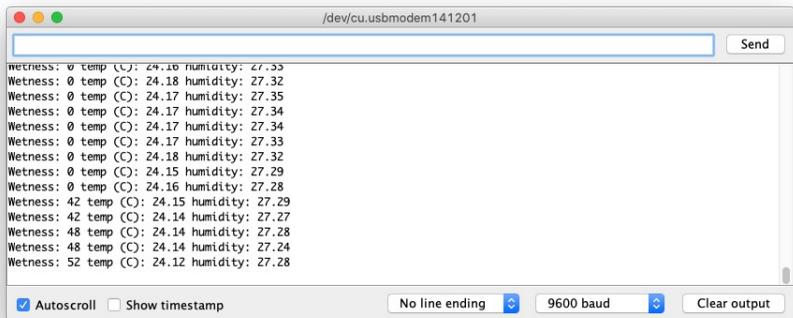


Here is the code for this example:

```
#include "PlantMonitor.h"

PlantMonitor pm = PlantMonitor(10, 11); // RX, TX

void setup() {
  Serial.begin(9600);
}

void loop() {
  if (Serial.available()) {
    char cmd = Serial.read();
    if (cmd == 'l') {
      pm.ledOff();
    }
```

Page 17

```
    else if (cmd == 'L') {
      pm.ledOn();
    }
  }
  report();
  delay(1000);
}

void report() {
  Serial.print("Wetness: ");
  Serial.print(pm.getWater());
  Serial.print(" temp (C): ");
  Serial.print(pm.getTemp());
  Serial.print(" humidity: ");
  Serial.println(pm.getHumidity());
}
```

The library uses another Arduino library called `SoftSerial` to communicate with the Plant Monitor. This can carry out serial communication on any of the Arduino pins. So, when an instance of `PlantMonitor` called `pm` is created, the pins to be used to communicate to the Plant Monitor hardware are specified (in this case, 10 and 11).  If you like, you can change 10 and 11 for other pins.

The main `loop` checks for incoming messages of *L* or *l* from you to turn the LED on or off respectively, using the `pm.ledOn` or `pm.ledOff` commands.

Getting readings from the PlantMonitor all takes place in the report function that writes out all the readings to the Arduino IDE's Serial Monitor.

# TROUBLESHOOTING

**Problem:** When I first connect power to the PlantMonitor, the LED cycles through colors. Is this normal?

**Solution:** Yes, this is the Plant Monitor doing a self-test as it starts up.


**Problem:** The LED on the Plant Monitor does not light at all.

**Solution:** Check the power connections to the Plant Monitor. Alligator leads and jumper wires can become faulty. Try changing the leads.


**Problem:** I am connecting using the serial interface, and I get wetness readings, but the humidity and temperature readings are wrong and not changing.

**Solution:** You may have inadvertently powered your Plant Monitor from 5V rather than 3V. This may have destroyed the temperature and humidity sensor.


# SUPPORT

You can find the Product's information page here: https://monkmakes.com/pmon including a datasheet for the product.

If you need further support, please email support@monkmakes.com.

# MONKMAKES

As well as this kit, MonkMakes makes all sorts of kits and gadgets to help with your electronics projects. Find out more, as well as where to buy here: https://monkmakes.com you can also follow MonkMakes on Twitter @monkmakes.