1 Introduction

○ Objective:

★ Bonus:

2 Requirements

water proof ruler

3 Water Level Sensor

https://circuitdigest.com/microcontroller-projects/interfacing-water-level-sensor-with-arduino

Much of the information and the images for this sensor can be found on this Last Minute Engineer page[**LME˙water˙sensor**].

1. Introduce water level sensor
2. Discuss how they work using electrical conductivity
3. Build circuit
4. Program **

   Use serial comm to return and plot values
5. Experiment

   Create a chart for sensor outputs based on water level

   Is the relationship linear?
6. Introduce if-statements

   Can you use if-statements to make LEDs turn on/off as the water level changes?
7. Use different salinity of water to mess with their systems
8. Experiment

3.1 How it works

Water is electrically conductive

This sensor uses an electrical signal to indicate a water level. As the water in the jar rises, more and more of the sensor is going to be covered. As the sensor gets covered, it is able to output a high voltage signal. We can measure this analog voltage signal to calculate the amount of water in the jar.

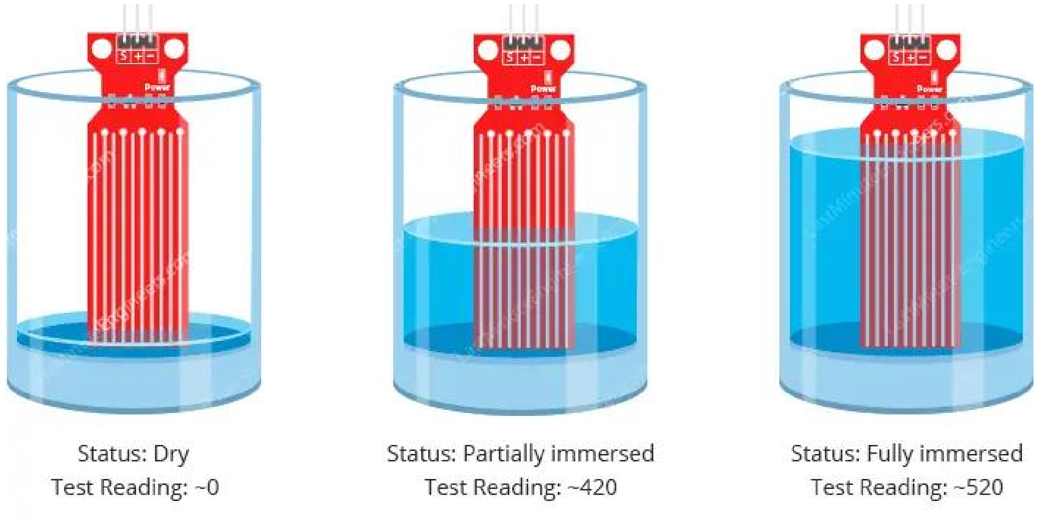
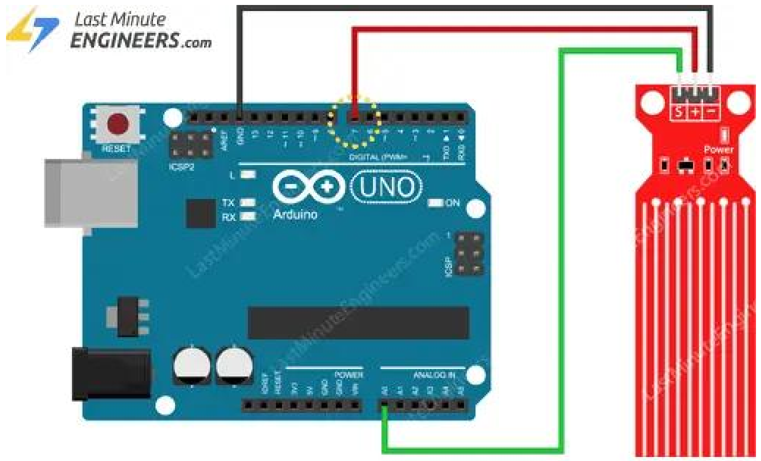


Figure 1: Sensor response to different water levels.

3.2 Circuit

The circuit for this sensor is fairly simple. All we need to do is provide a ground and power source to make it work, and then a connect a wire from the sense pin on the sensor to an analog pin on the Arduino.

Metal, like the contact strips on our sensor, will start to rust or corrode when it is exposed to water. The corrosion of a metal can occur much faster if an electrical current is flowing while it is exposed to water. As a result, we will be powering our sensor using a digital pin. We will turn on the power when we want to take a measurement and then switch it off when we are not using it.



Use a digital pin to power the sensor only when taking a reading to reduce corrosion.

Figure 2: Sensor response to different water levels.

3.3 Program

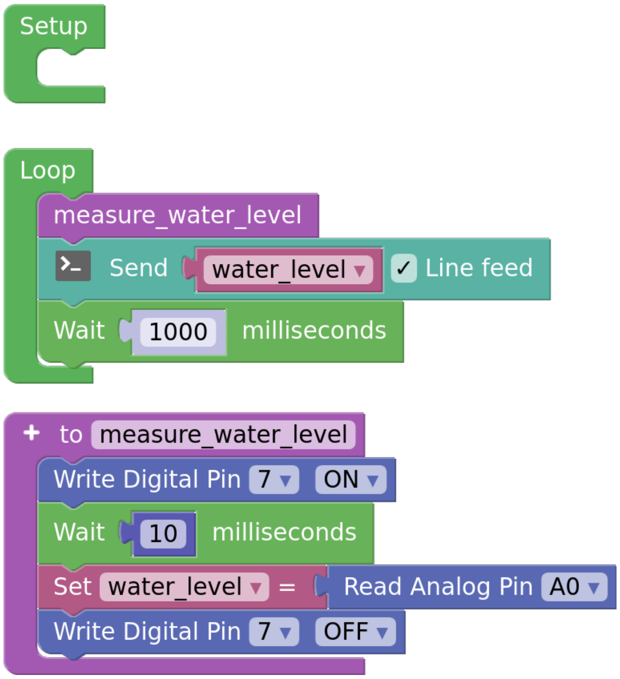
Here is the program that we will be using to collect measurements. The main loop is quite simple.

1. Collect a measurement for the level of the water.
2. Send the measurement to the computer.
3. Delay until it time to take the next measurement.

The more complicated part is the process to obtain a measurement. This is done using the function ***measure_water_level***.

1. Turn digital pin 7 on, this provides power to the sensor.
2. Wait for 10 milliseconds, allowing the sensor time to settle after the power has been turned on.
3. Make a measurement for the level of the water.
4. Turn off pin 7, switching off the power to the sensor to protect it.
5. Return the measurement.

update caption



```
Setup

Loop
    measure_water_level
    >_ Send  water_level ▾  ✓ Line feed
    Wait  1000  milliseconds

+ to measure_water_level
    Write Digital Pin 7 ▾  ON ▾
    Wait  10  milliseconds
    Set water_level ▾ =  Read Analog Pin A0 ▾
    Write Digital Pin 7 ▾  OFF ▾
```

Figure 3: Sensor response to different water levels.

3.4 Experiment

For some of the following tasks that we will be doing, we need to know what the sensor's output is for a range of different water levels. Start with an empty cup and record the sensor's output. Using a ruler, add water to raise the water level 5mm. Rec

1. Start with an empty cup and record the sensor's output.
2. Using a ruler, add water to raise the water level 5mm.
3. Record the new measurement from the sensor.
4. Repeat steps 2 & 3 until the sensor is fully covered. Try not to get water on the electronics at the top of the sensor if possible.

○ Objective: Create a chart of the sensor's output for a range of different water levels in the cup. You can use Google Spreadsheets to record your results.

★ Bonus:

3.5 If Statements

3.5.1 Objective

For this part, we will be adding LEDs to your circuit. The goal of the LEDs is to have then turn on when the water reaches specific levels. You can use the results from the previous experiment to decide which at what measurement value each LED should turn on/off.

Connect a green LED to pin 9, yellow to 10, and red to pin 11. Remember, you always have to use a resistor with LEDs. You can reference last weeks notes if you forget how to connect LEDs in a circuit.

3.5.2 If/Else Flow Control

To make the LED turn on/off depending on the level of the water we will require a new type of programming logic flow control with ***if/else statements***. They work using the following flow: ***if*** some condition is true then ***do*** some action, ***else*** do something else. Look at figure 4 and look for the bold & italics terms in the program.

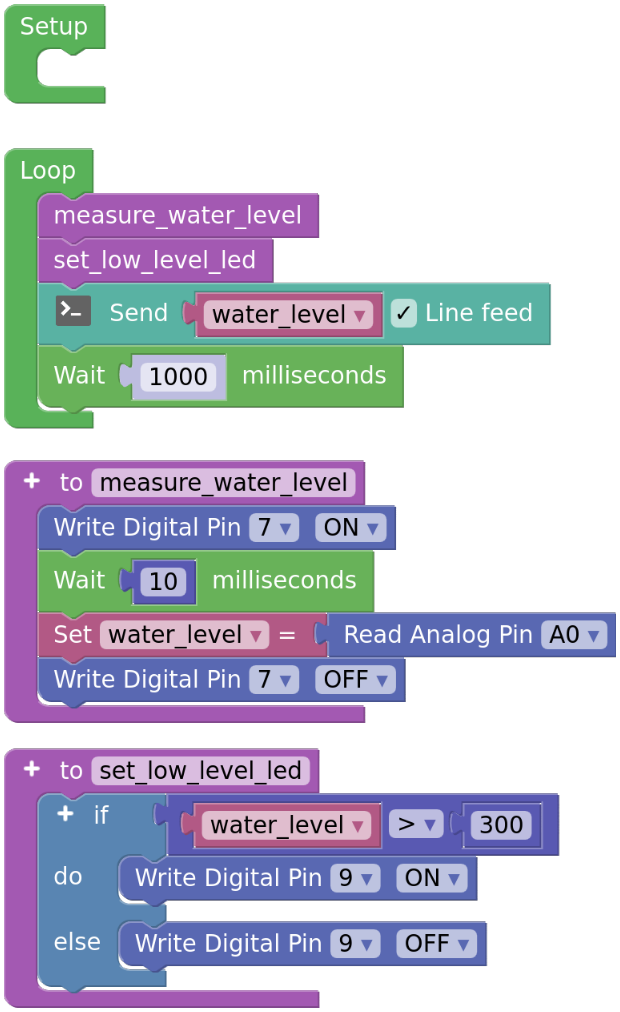
3.5.3 Program

We will need to add a new function for each LED. An example is provided below for the first LED. You will need to make a new function for each of the other LEDs. The new function uses an if/else statement. This function, ***set_low_level_led***, with have the following flow:

1. ***if*** the water level measurement exceeds the value 300, then
   - ***do***, turn on pin 9 to light up the LED
2. ***else***, turn off pin 9 to switch off the LED

Note: the value used here is 300. It was chosen as a random example and you will need to change it based on the experimental results you obtained in the previous section.

update caption



```
Setup

Loop
  measure_water_level
  set_low_level_led
  Send water_level ✓ Line feed
  Wait 1000 milliseconds

to measure_water_level
  Write Digital Pin 7 ON
  Wait 10 milliseconds
  Set water_level = Read Analog Pin A0
  Write Digital Pin 7 OFF

to set_low_level_led
  if water_level > 300
  do Write Digital Pin 9 ON
  else Write Digital Pin 9 OFF
```

Figure 4: Sensor response to different water levels.

3.5.4 Your Turn

◯ Objective: Add 3 LEDs to your circuit: a green, yellow and red one. Make the LEDs turn on only when:

1. Green - the cup is 1/4 full or more
2. Yellow - the cup is 1/2 full or more
3. Red - the cup is 3/4 full or more

★ Bonus (A big challenge!): Can you make the LEDs progressively light over their ranges? You will need to use the map function and PWM output to make this work. For example, the yellow LED should remain fully off until the green LED is at full brightness when the cup is 1/4 full. Then, as the cup fills up from 1/4 → 1/2, the yellow LED should progressively get brighter and brighter.

3.5.5 Testing the System

We will be testing your results to see how it performs!

4 Line Detector

https://circuitdigest.com/microcontroller-projects/interfacing-ir-sensor-module-with-arduino

4.1 How it works

4.2 Circuit

4.3 Program

4.4 Experiment

5 Ultra-Sonic Range Finder

https://circuitdigest.com/microcontroller-projects/interfacing-hcsr04-ultrasonic-sensor-module-with-esp32

5.1 How it works

The ultrasonic sensor is capable of measuring distances between 2 cm and 400 cm (4 meters). It does this by emitting a short sound wave and listening for an echo. The duration of time between the initial sound pulse and the echo can be used to determine how far away the object is.

5.2 Circuit

5.3 Program

5.4 Experiment