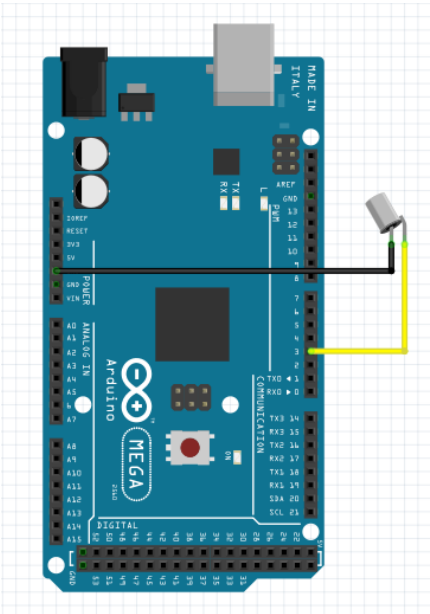
Water Level Switch

Introduction

This is a simple water level sensor. It has two wires and actually functions and behaves like a switch. You can use a water level sensor to detect other liquids as well. Using a water level sensor, you can detect if water passes certain threshold so you can control a water pump or close a valve.

Operation is simple. The sensor has two pins. There is a moving part, which is a floater. Inside this is a magnet that activates a switch inside the non-moveable part. When the switch is activated, the two wires are shorted.

Interfacing with Arduino



Normally, you use a pull-up resistor to connect switches to a logic. The pull-up resistor ensures that the input is at 5V or logic HIGH when the switch is not shorted. Luckily, Arduino pins has pull-up resistors that you can activate. Lets use the code (Water_level sketch)

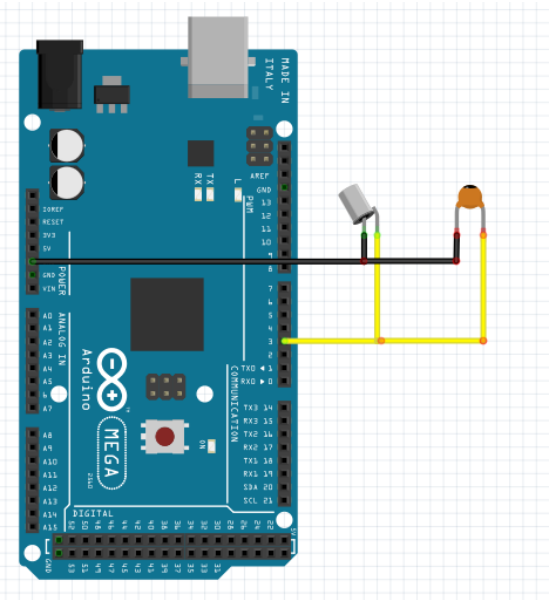
```
#define floatswitch 3

void setup() {
  pinMode(floatswitch, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(floatswitch), dofloat, FALLING);
  Serial.begin (9600);
}

void loop(){
}

void dofloat() {
  Serial.println("Switch is triggered");
  }
```

The program is very basic. It attaches pin 3 to trigger an interrupt whenever the input transitions from HIGH to LOW or “FALLING” edge of the logic (line 5).

It instructs the microcontroller to run “dofloat” whenever the falling edge is detected
Notice that the other wire of the sensor is connected to GND. This means, it will short pin 3 to ground when activated.

Note also how we enabled floatswitch as an input on line 4. INPUT_PULLUP instructs the microcontroller to enable its internal pullup resistor.

Open the serial monitor and see that Arduino is triggered whenever the water level sensor is straightened. However, notice that it is being triggered multiple times even if you only manually trigger it once. This is because the switch inside is mechanical and “bounces”. To correct this, we put a small value capacitor across/in parallel to the float sensor – like 0.1uF.



That’s better. The capacitor ensures that the circuit is stable by electrically removing the “bounce” effect and ensuring that the input to our Arduino only changes once.