# mnIntroduction:

The aim of the project is to automate the water system at home. The system is implemented using micro-controller board and IOT (internet of things) dashboard to monitor the data on a mobile console app.

# Main System (down)

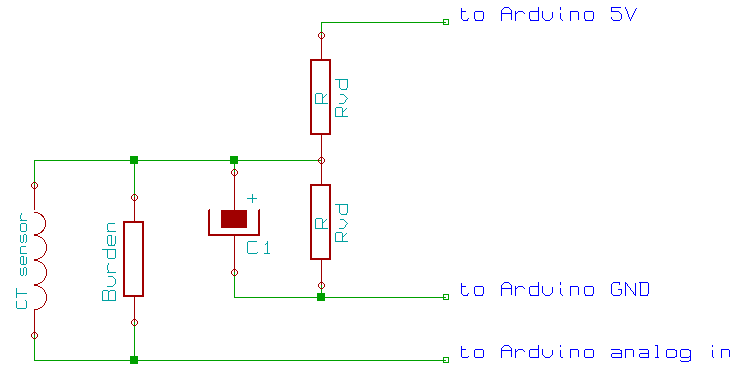
## Setup:

### Hardware:

* Node MCU v3
* Flow meter (YFB1) half inch in diameter
* Clamp current meter (SCT-013 invasive clamp meter 30A/1V)
* Ultrasonic sensor for water level detection
* Solenoid water valve
* 5V relay, for the overload control
* Power source ; 12 for the Arduino and solenoid valve, 5V to power up the sensors
* Two cooling fans
* 5V valve to turn on the water pump

### On the Arduino Pins:

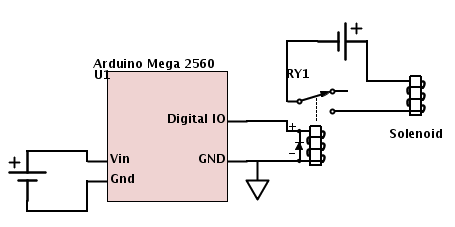
1. Water leveling: (JSN-SR04T)
   1. D1 For RX as GPIO 5
   2. D2 for the ultrasonic TX as GPIO 4
   3. 5V VCC and GND
2. Flow meter:
   1. Gray wire 5v
   2. Black wire, signal to D6 pin as GPIO 12
   3. Black- white wire to GND
3. Clamp Current meter: The Analog pin is A0



1. Relay and overload Valve: The relay controls the overload valve, since it is 12 V

* Relay to Pin D3 as GPIO 0
* Valve circuit: use the below sketch for demo, pay attention to the Diode

1. Water pump relay: DD4 as GPIO 2 pinout to the contactor that controls the water pump systems that dad installed it previously, so that the water pump turns on automatically.



### Installed Schematic:

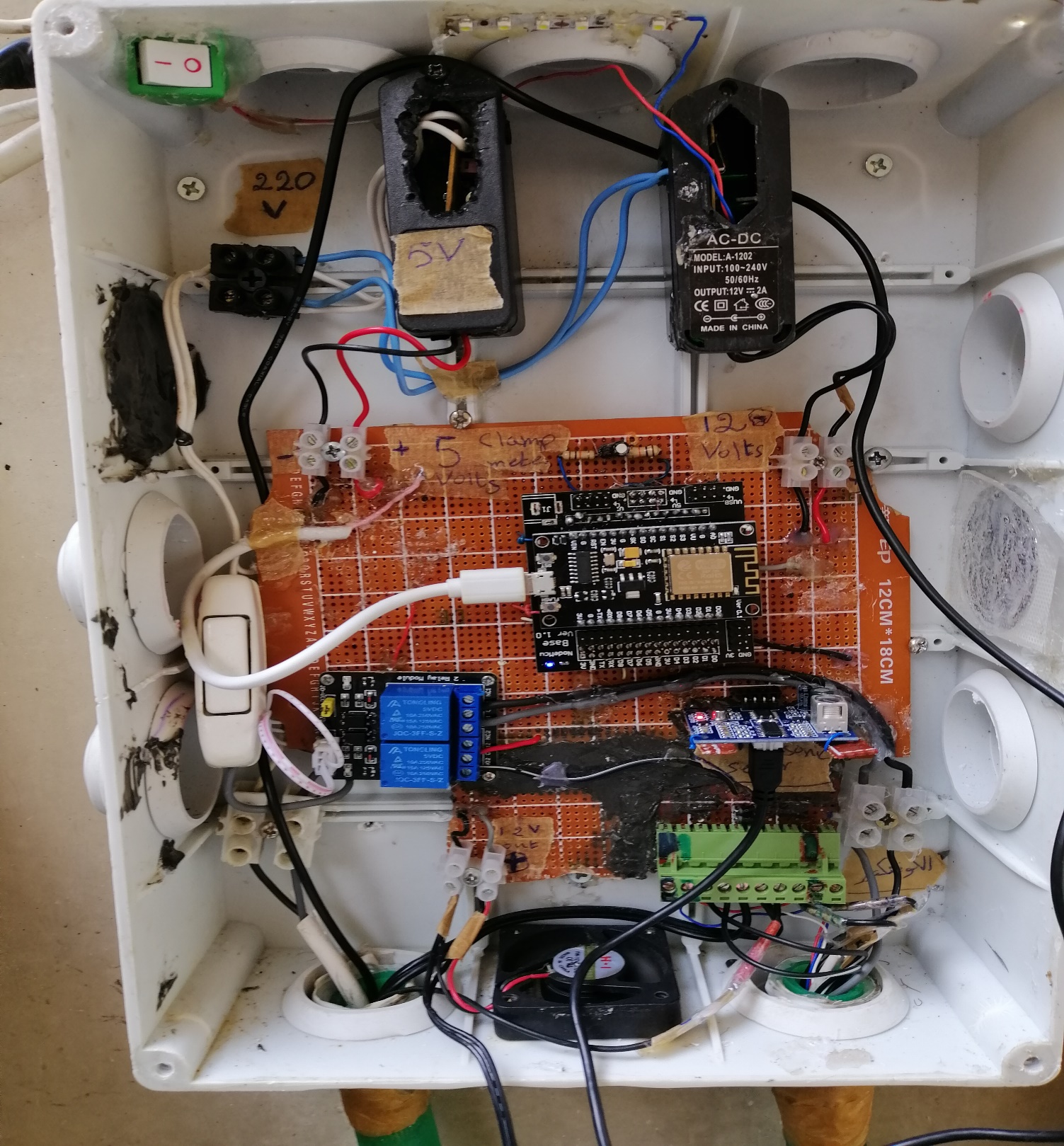


Figure 1: The main water management system

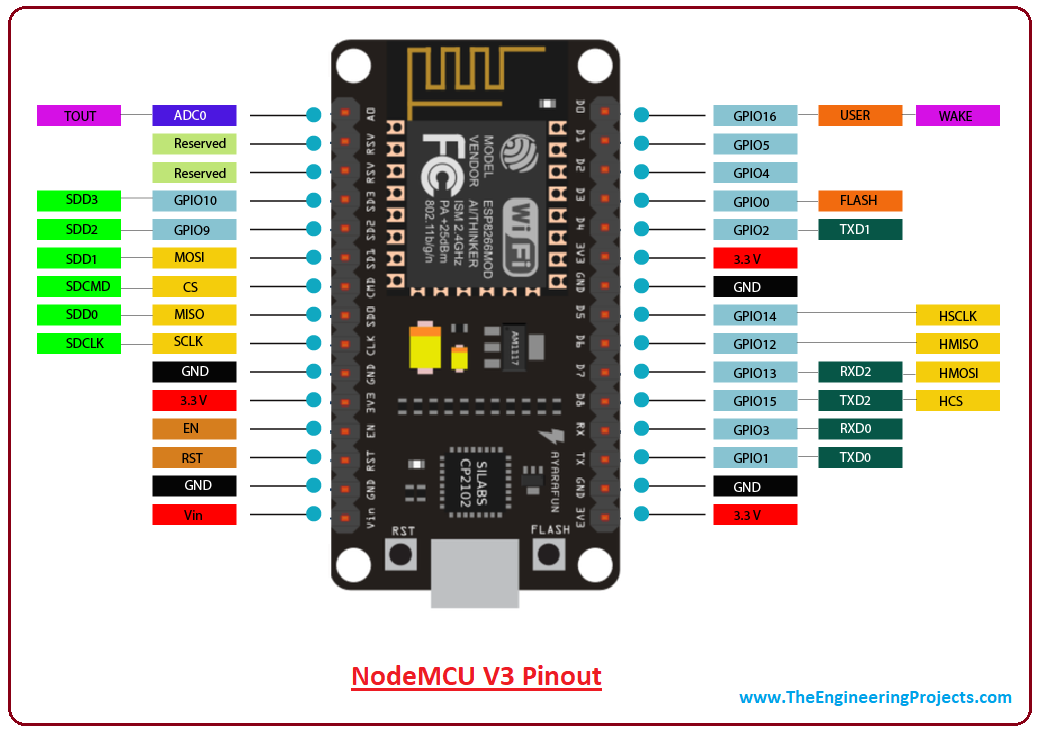


Figure : Node MCU v 3 pinouts

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RESET | GND | Clamp Meter | Clamp Meter | Overload Valve 12 V + | Overload Valve 12 V - | Flow meter 5V + | Flow meter signal wire | Flow meter 5V- | Indicator Led TX GPIO 1 |

### The Green screws pinout extension as follows:

* GPIO 1: to show the system is on, the LED is in the grey box at my room
* For the flow meter, the wires are marked. However:
* The 5 V is the black wire without any writing on it (red marking)
* The signal wire is dotted with white lines (green marking)
* The GND wire has some writings on it (blue marked)
* In the grey box at my room:
* Red wire: reset pinout
* White wire: D41
* Blue wire: GND
* The Arduino file is “Water Management NodeMCU” in the (water management file)

## Field Schematic:

Water Pump

**RIGHT CONTACTOR BOX**

* On the contactor level: connect R2, R4, 2, & 4 to water pump hot wire pin
* Free 220 V hot wire to pins 1 and 3.

**RIGHT CONTACTOR BOX**

R1 and R3 from Dad’s old water management system output

The Right contactor NC (Normal Closed) pins are connected directly to dad’s old water pumping system, and NO (Normal Open) pins are connected to a free 220 V hot wire activated by the Arduino relay.

# The Up-Roof System:

The aim of the system installed to lively monitor the water level in the tank (on the distance between the sensor and the water)

## Hardware:

* Arduino Uno
* Ultrasonic Sensor (JSN-SR04T) using digital pins 2 & 3 (2: trigger, 3:echo)
* 220 V to 12 V adaptor 2A
* Thermal resistor for temperature reading ( pin A0)
* And a push button incase the box was opened (digital pin 4)

## The Circuit

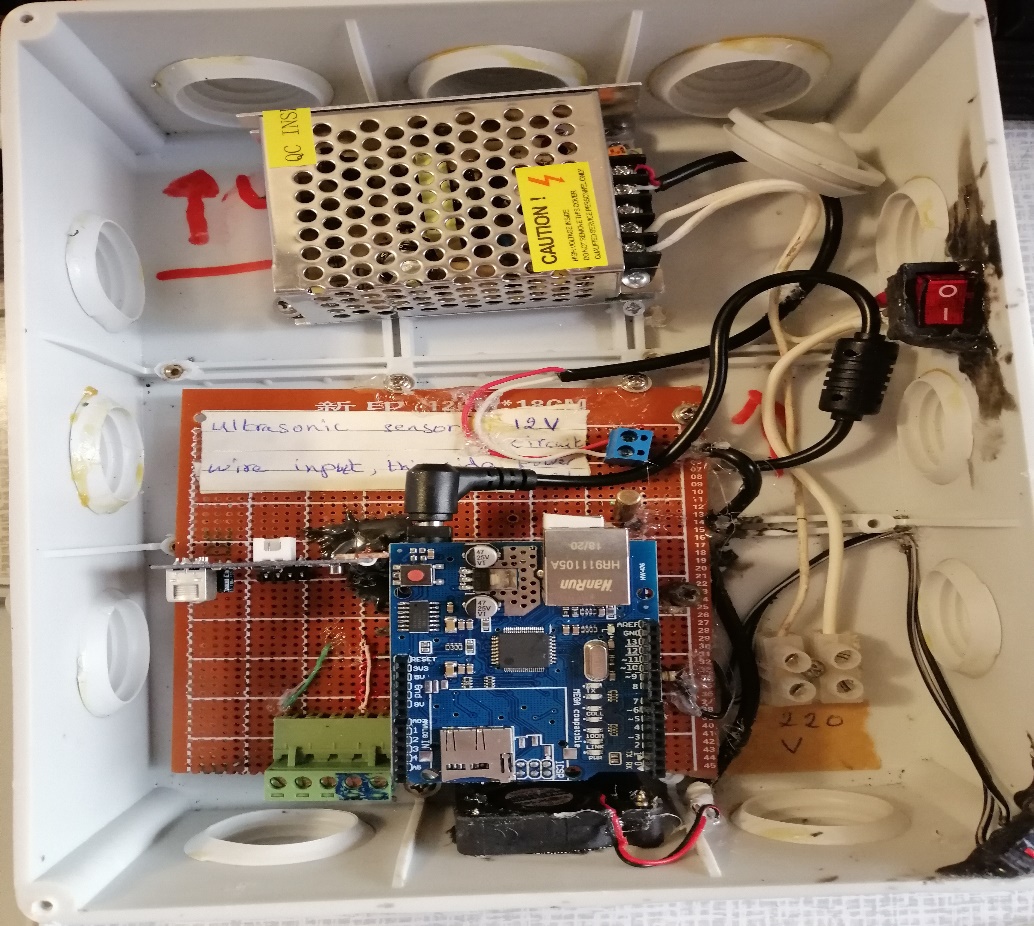


Figure 3: the installed circuit on the Roof

### The Green screws pinout extension as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D6 | D5 | EMPTY | Reset PIN | GND |

* D5: A LED that shows the box is open
* D6: A LED that shows the Arduino is active
* The software file called (upper\_tanks) stored in the same project’s file

# Internet Part:

I am using Internet of things dashboard called Thinger (<https://thinger.io/>) and a mobile console application.

For the mega 2560 (that of the tanks below) program use the thinger.io latest library version 2.13.0 and for the Arduino Uno (that of the tanks on the roof) use the Thinger.io library version 2.9.2

For the Future, using the APIs and Thinger configurations, I will use 3rd party servers to send sms notifications or IFTT web hook trigger based notifications