Dipl.-Ing. Sören Schackert, B.Sc.

Matrikel-No.:

Course 448.045, LU, Embedded Systems

Project Description

Project description

Content:

1. General project description

1.1. Motivation

1.2. Base idea

2. Initial situation

2.1. Hardware

2.1.1. Sensor humidity

2.1.2. Water Pump

2.1.3 Relay

2.1.4. Micro-controller

2.2 Software

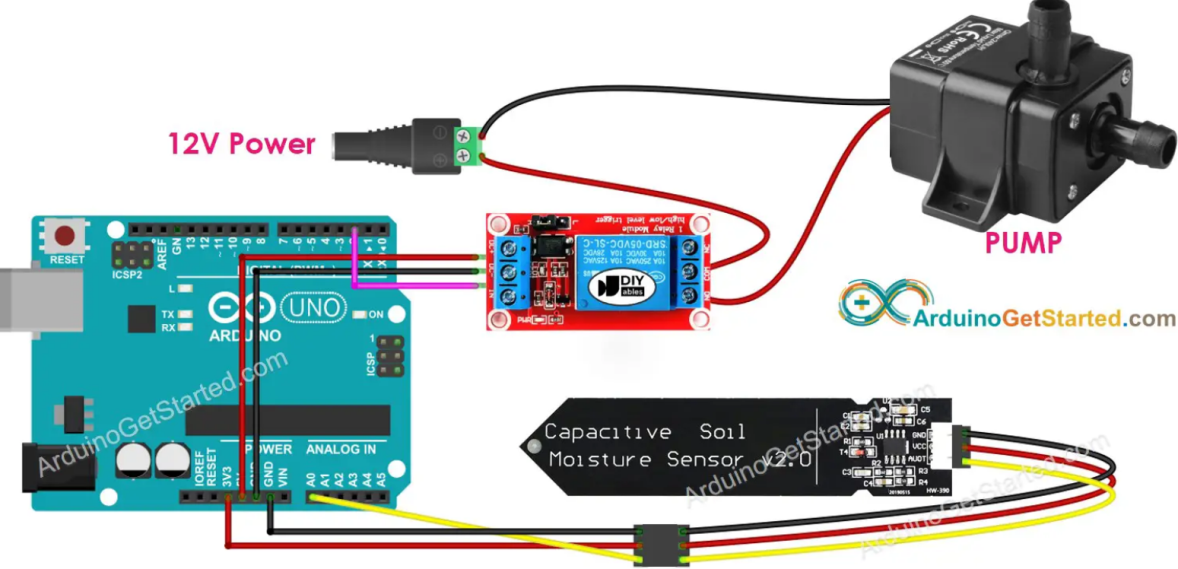
3. Project description

3.1. Propose

3.2. Cloud

1. General project description  
  
1.1 Motivation

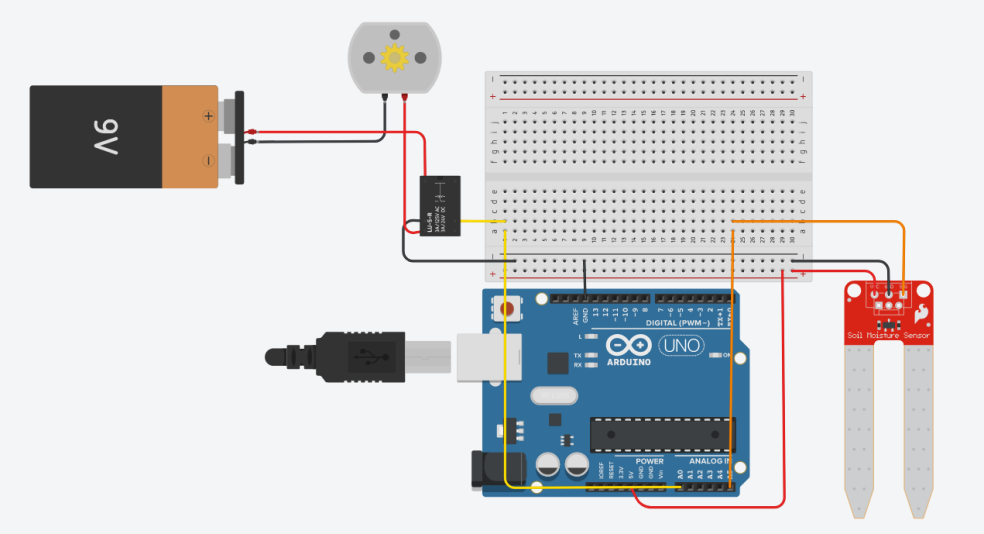
We want to solve the problem that many persons have in house. In many cases, you want to have hydrated your plants, but sometimes it´s not possible to do it. In the next project we want to have a system for solving this problem and also we will monitoring all the information about this system in Cloud.

Basically, this project we want to hydrate the plants when the plant needs to be hydrate using the combination of different technology and sent the relevant information to cloud about the system.  
  


In this project we will study the difference between using capacity sensor and resistors sensor and why we will use one and the not other.

1.2 Base idea

The idea is having a system that can proportionate water to the plant. Basically using a sensor (soil moisture), “Motor-Water” and a Microcontroller we can create a system that can do this target, the soil moisture will take samples that will sent to the microcontroller and say if the plant has enough water or not.



On the other hand, the system must be able to sent information of the current situation of the plant, or when was the last time that the system was activated.

Basically the user cans know the information about the system using a report in Cloud.

2. Initial situation – Hardware

2.1. Hardware  
  
2.1.1 Sensor

The sensor will take measures all the time of the status of the humidity of your plant. The sensor must send the information to the Micro-controller using a pin as IN-PUT. The pin is an IN-PUT because we want to read the information of the environment in this case and we want to know the value that the sensor.

Relevant information about the Capacitive soil Moisture Sensor:

* Analog output of moisture content that will be connected to analog input on a MCU
* More corrosion resistant than resistive type of sensor
* 3.3 or 5V operation.  Low power so may be driven from digital pin on MCU
* Dry soil needing water > 380

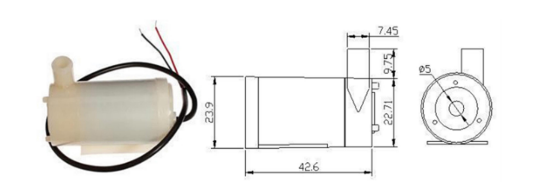
# https://i0.wp.com/makerselectronics.com/wp-content/uploads/2022/10/H9d5b250b20a84c918eeec6158fd47759B.png?fit=800%2C800&ssl=1 Capacitive Soil Moisture Sensor V2.0

2.1.2 Water Pump

The Water Pump will do something depending of the instructions of the sensor. In this case if the sensor will have samples > 380, the pump will be activating until the samples are under 380.

Relevant information about Water pump:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Voltage | Current | Power | Water-lift | Flow |
| JT-DC3L-4.5 | 4.5V | 0.18 | 0.91 | 0.55 | 100 |



**DC Motor little water pump**

2.1.3 Relay

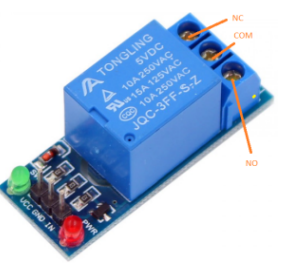
The relay will support the water Pump, this instrument is basically a switch that allow or not depending on the position the electricity for the circuit. The DC Motor will depend on the value of the relay basically and the relay will depend of the microcontroller, if the microcontroller is in High or Low level in the pin that the user is using.

Relevant information about Relay:

1. Module Interface Output Section

* NO - Normally Open
* COM - Common
* NC - Normally Closed

1. Module Interface Input section

* VCC: Connect 5V power supply positive (according to the relay voltage)
* GND: Connect 5V power supply negative
* IN: relay module signal trigger (low level active)  
  

**Relay JQC-3FF-S-Z**

2.1.4 Micro-controller

The Arduino UNO R3 was the selected option for working in this project, because it´s the best option for rookies and have a broad documentation. The micro-controller will read the information that we take with the sensor and then will send signals to the relay/Motor for providing water.

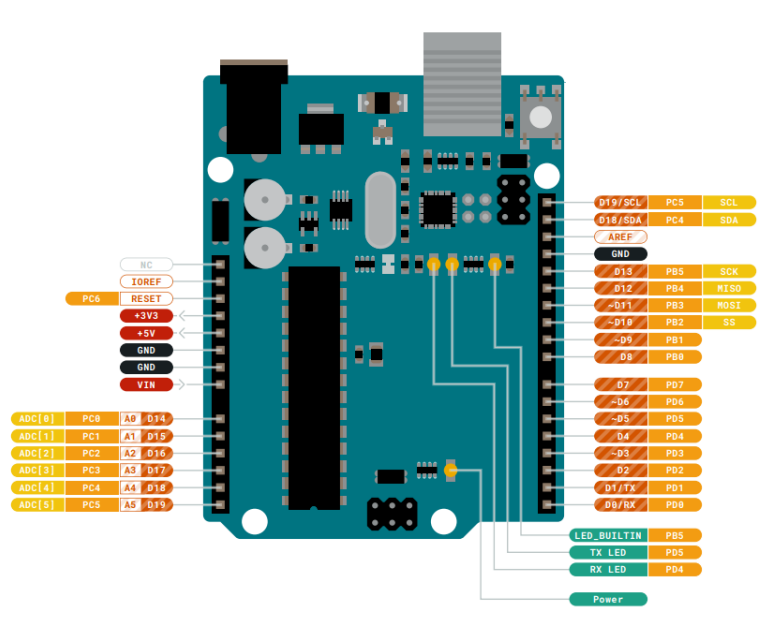
Relevant information about the Micro-controller:

* Analog part

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Function | Type | Description |
| 1 | NC | NC | Not Connected |
| 2 | IOREF | IOREF | Reference for digital logic V - connected to 5V |
| 3 | Reset | Reset | Reset |
| 4 | +3V3 | Power | +3V3 Power Rail |
| 5 | +5V | Power | +5V Power Rail |
| 6 | GND | Power | Ground |
| 7 | GND | Power | Ground |
| 8 | VIN | Power | Voltage Input |
| 9 | A0 | Analog/GPIO | Analog input 0 /GPIO |
| 10 | A1 | Analog/GPIO | Analog input 0 /GPIO |
| 11 | A2 | Analog/GPIO | Analog input 0 /GPIO |
| 12 | A3 | Analog/GPIO | Analog input 0 /GPIO |
| 13 | 4/SDA | Analog input/I2C | Analog input 4/I2C Data line |
| 14 | 5/SCL | Analog input/I2C | Analog input 5/I2C Clock line |

* Digital part

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Function | Type | Description |
| 1 | D0 | Digital/GPIO | Digital pin 0/GPIO |
| 2 | D1 | Digital/GPIO | Digital pin 1/GPIO |
| 3 | D2 | Digital/GPIO | Digital pin 2/GPIO |
| 4 | D3 | Digital/GPIO | Digital pin 3/GPIO |
| 5 | D4 | Digital/GPIO | Digital pin 4/GPIO |
| 6 | D5 | Digital/GPIO | Digital pin 5/GPIO |
| 7 | D6 | Digital/GPIO | Digital pin 6/GPIO |
| 8 | D7 | Digital/GPIO | Digital pin 7/GPIO |
| 9 | D8 | Digital/GPIO | Digital pin 8/GPIO |
| 10 | D9 | Digital/GPIO | Digital pin 9/GPIO |
| 11 | SS | Digital | SPI Chip Select |
| 12 | MOSI | Digital | SPI1 Main Out Secondary In |
| 13 | MISO | Digital | SPI Main In Secondary Out |
| 14 | SCK | Digital | SPI serial clock output |
| 15 | GND | Power | Ground |
| 16 | AREF | Digital | Analog reference voltage |
| 17 | A4/SD4 | Digital | Analog input 4/I2C Data line (duplicated) |
| 18 | A5/SD5 | Digital | Analog input 5/I2C Clock line (duplicated |



**Arduino UNO R3**

2.2 Software

Using the documentation that Arduino UNO has, we will write our solution, the solution will be written in C/C++.