IoT Project:
Height / capacity monitor

Grupo K:
Tim Reiprich
Tegshigzugder Otgonbayar
Luca Maltagliati

January 15, 2020

This documentation will be updated until the final delivery.

Contents

Context and general summary of the operation of the system

1 Introduction

2 Context and general description of the device

Our project focuses on the measurement of containers holding some kind of liquid. Based on the volume of these containers alarms can be sent to the users to notify them about the amount. To accomplish this, sensors will be used, that can determine if the height of the volume is above or below a certain threshold.

2.1 Hardware scheme

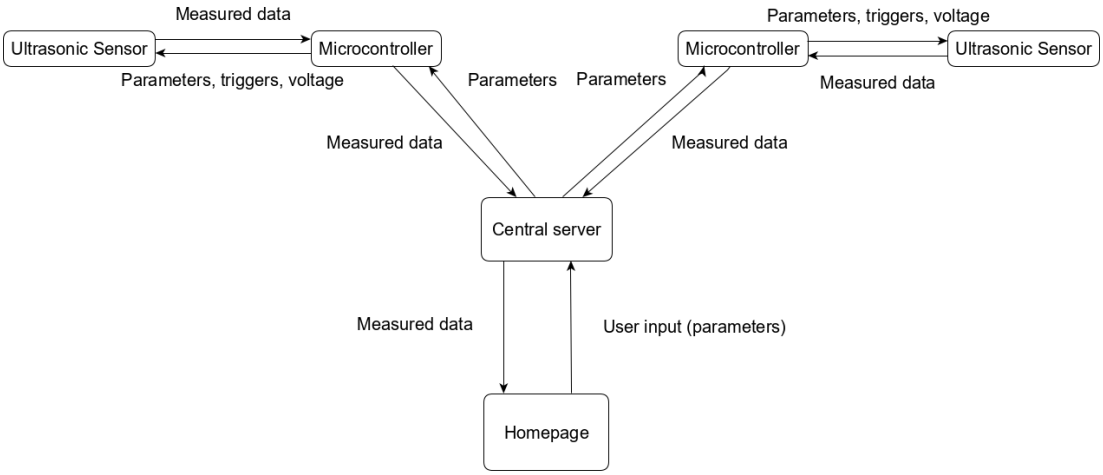

Figure 1: Hardware scheme

The basic scheme of system hardware is described as follows: the main components are the micro-controller as central server, the sensors and a web-page. The sensors must be able to connect to the servers by a WiFi-connection and for this we assume that the connection is robust and will be available in the given environment.

2.2 Sensor

We will use an ultrasonic sensor to measure the level of the liquid. An example of the sensor is Ultrasonic Sensor Module HC-SR-04 by Robokart. We want to choose this approach because it is one of the best ways to sense proximity and detect levels with high reliability. Every sensor is connected to a microcontroller that collects the data and sends them to the central server via WiFi.

Every container will have a microcontroller that is connected to at least one sensor (see figure 4). Certain specified heights of minimal and maximal volumes will be set, depending on various factors, like the position of the container, the probability of it being used, the liquid type and as such.

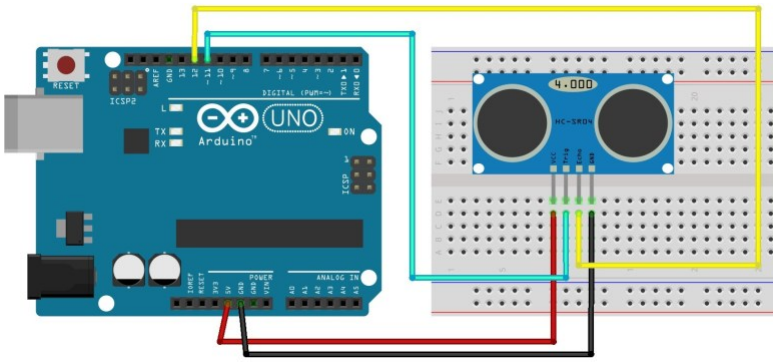

Figure 2: Example of a sensor connected to a microcontroller

2.3 Central server

We assume a LAN network to which the controllers and central server are connected. Through this the microcontrollers will be able to communicate with the central server by sending continuously the measurements of the liquid heights. The central server will perform checks on the thresholds and if needed send an alert to the user.

As seen in Figure 1, the central server can also host a web page (e.g. with the Node-RED framework) in which the data are visualized in diagrams. This will be in an user-friendly interface, for it to be usable by non-technical people also.

3 Steps of the project

The first steps of the project will be to research about similar projects and identifying platforms, programming languages and frameworks, choosing the most suitable ones for our case. Furthermore, the connection between devices plays a crucial role, so we need to know what kind of protocols are needed.

Possible extensions are to also give specific numbers of the height of the containers, how much volume they contain. It is also possible to set up LEDs; their light will warn about specific situations, e.g. when a container is empty a green light is shown, a red one when is full.

Complete list of components that will be used, with a short description and why

List of ordered components

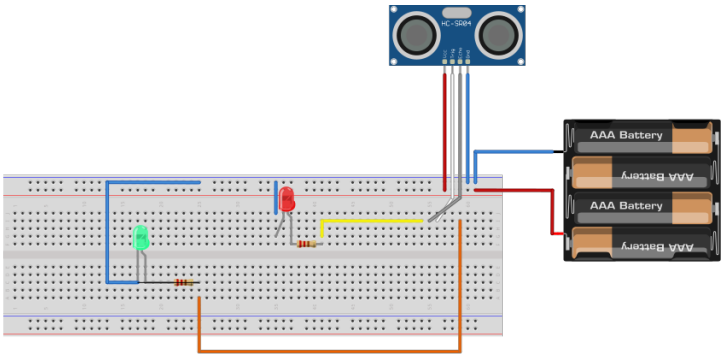

Figure 3: Hardware scheme

- NodeMCU DevKit v1.0 development board (ESP8266) - we will use the microcontroller connected to sensor. It collects and sends data to broker
- Infrared proximity sensor (digital / analog output module) - the main sensor to measure the height of the object
- Rechargeable battery - power source for the microcontroller and sensor
- Battery charge controller - for the battery
- Small materials:
  - cables - to connect sensor and microcontroller
  - breadboard - to connect sensor and microcontroller
  - LEDs - to signal certain thresholds

Complete wiring scheme - will be documented during the implementation

- Flowchart of operation
- Another entry in the list
- List of functions used to communicate with the sensors, including input / output parameters and operation.
- List of functions that will be used to communicate with the base station, including input / output parameters and operation.
- Other functions necessary for device operation
- Other technical considerations of the devices not described above.

Base station documentation: - will be documented during the implementation

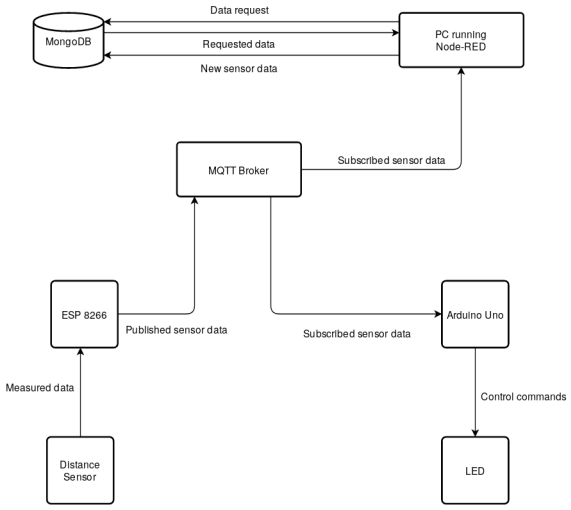

Figure 4: first draft of the flow chart

- Scheme of the software components that run on the base station and how they are related to each other (Broker MQTT, Node-RED, Mongodb?).
- List of functions (flows) themselves to be used to communicate with devices, including input parameters / output (messages) and operation.
- List of functions (flows) themselves to be used to communicate with the database and the SCADA, including parameters input / output (messages) and operation.
- Other own functions necessary for the operation of the base station
- Other technical considerations not described above.

Device-base station interface. (Message structure)

User manual:

- System installation, specifications and calibration
- System use and maintenance manual
- Detection and solution of possible errors

3.1 Node-RED

x

3.2 Arduino

x

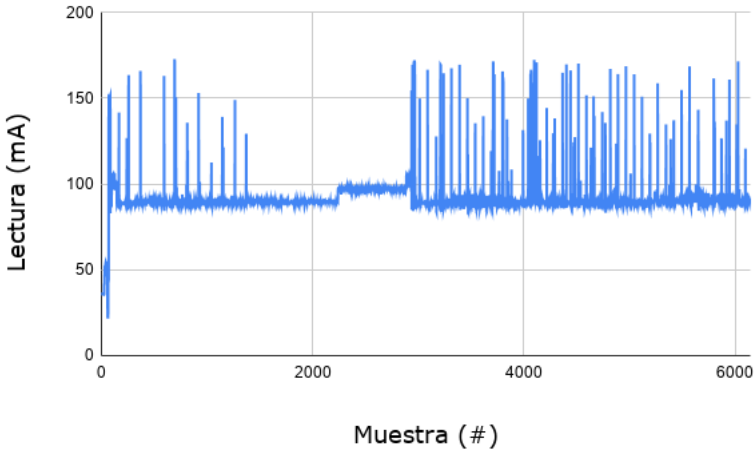

Figure 5: Hardware scheme

3.3 Experiments and Results

4 Power Consumption

- graph of power consumption
- low consumption → esp starts, connects to Wifi, duration of startup variies
- high consumption → data is sent and received, spikes of constumption whenever data exchange happens
- para ahorrar: lower frequency of data exchange, if LEDs on basket aren't needed (because esp is in basket or whatever), then only the virtual LEDs in Node-red should be enough and a little bit of power can be saved

5 Summary