WIND TURBINE GENERATOR SUPERVISION PROJECT

Agapito Gallart and Edgar Felipe Rojas Cala

UdL | state and ubiquitous systems

Index

1. Project proposal

The development of an embedded system is required that allows the monitoring of wind turbine towers, acquiring information from multiple towers, and sending it to multiple users in monitoring stations.

the applicant requires a series of hardware requirements and a minimum number of components such as a Supervision station, Data providers, and an MQTT Broker; Likewise, the communication protocols between the components and their devices are proposed.

In this documents, we show the requirements, the hardware propose, the design process, the conclusions, and finally a technical section where you should see the final implementation of the project with all schematics and communication protocols that were required

2. Project requirements

next show the requirements of the project owners:

- Thereis just only one Supervision Station for the whole WTGF.
- All sensors are placed far from the Supervision Station.
- Temperature and humidity sensors are placed near the Rotor Hub.
- The Tower movement sensor is placed in the middle of the tower.
- Sensor measurements must be obtained in Real-Time.
- Sensors are controlled by the Data providers, and they will show the current sensor readings.
- Whole WTGF data is transferred to the Supervision Station through an MQTT broker.
- The MQTT broker will show graphically the current number of subscribers.
- The maintenance manager requires access to the last 24h data that should be available graphically on the Supervision Station

3. User stories

the user's stories are shown below, across the project implementation these stories were used for the PBI and process management

1. It is necessary to obtain the temperature and humidity at the top of the WTG
2. It is necessary to determine the WTG tower movement
3. Sensor data must be sent to a Supervision Station
4. MQTT broker must show the current number of connections (percentage)
5. Maintenance manager requires graphical representation of the obtained sensor data, within the last 24h, in the Supervision Station.

4. Engineering book

Product backlog items

the process of implementation is based on the SCRUM method, for that reason, next show the product backlog items to build the project

Task	Score	related user story
download Arduino	100	1;2;3;4;5
Arduino "Hello World"	6	1;2;3;4;5
LCD_ArduinoTest	7	1;5
DHT11 ArduinoTest	7	1
ESP8266 on Arduino IDE	25	1;2;3;4;5
ESP01 Wifi	25	1;2;3;4;5
ESP01 Wifi As Client	30	1;2;3
Html_Client	5	1;2
LCD_visual_out-wifi	8	1;5
LCD on ESP01	22	1
DHT11 on ESP01	20	1
DP1_final.info	50	1;4
Drawing_on_LCD_screen_DP1	2	1
DP1 MQTT Pub	30	1;3;4
I2C Arduino	25	2;5
accelerometer_I2C_arduino	5	2
LCD I2C Arduino	5	2
I2C ESP01	15	2
accelerometer_I2C_ESP01	10	2
LCD I2C ESP01	10	2
DP2 MQTT Pub	10	2;3
DP2_final.ino	35	2;3
MCP23017 acquire information	5	4
MCP23017 Arduino	5	4
MCP23017_I2C_MCU8266	30	4
MCU8266_wifi_as_AccesPoint	25	4
MCU2866_MQTT_Broker	2	3;4
MQTT Mosquitto	10	3
MQTT_Broker_final.ino	30	3;4
Arduino connect whit ESP01	30	5
SupervisionStation MQTT Sub	10	3;5
SupervisionStation_wifi_as_Client	10	3;5

chibi_HelloWorld	30	5
Chibi Uart	50	5
Chibi I2C master	50	5
LCD_UART_Raspberry	25	5
Arduino_I2C_slave	25	5
improve project manamegent	100	1;2;3;4;5
SupervisionStation_final.ino	20	3;5
final delivery	100	1;2;3;4;5

the project requires the implementation of 40 tasks in total, whit a score based on 300 points per sprint and a total of 1000 points for the project; These tasks were distributed considering two important parameters:

- the project progress presentation sprints (time references).
- the different stages of development and components required by the project.

Sprint diary

As mentioned in the previous section, part of the tasks was separated into the different sprints; a timeline for each task of the different sprints are on the annexes of this

The development of each sprint is shown below, both in the training process and in the tasks used for each sprint.

Sprint 1

The beginning of this sprint was highly complex due to problems for the organization of the group; also the main objective of this delivery was to introduce the members of the group to the world of hardware programming. Finally, the last objective of this sprint was to start with the part that according to the workgroup was the easiest to do.

Task	duration (estimated)	task percentage
download Arduino	20 mins	100%
Arduino "Hello World"	15 mins	100%
LCD_ArduinoTest	15 mins	100%
DHT11 ArduinoTest	20 mins	100%
ESP8266 on Arduino IDE	30 mins	100%
ESP01 Wifi	20 mins	100%
ESP01 Wifi As Client	35 mins	100%
Html_Client	25 mins	100%
LCD_visual_out-wifi	15 mins	100%
LCD on ESP01	40 mins	100%
DHT11 on ESP01	20 mins	100%
DP1_final.ino	30 mins	70%

the estimated time for all the tasks proposed at the beginning of the sprint was 285 minutes but due to a long introduction with the Arduino program, the total time spent was more than 450 minutes. The time used to develop this sprint let complete 91.6% of the tasks; DP1_final.ino was the only task that was not completed since the task requires specific arrangements.

In this sprint, in addition to looking for the first user story, it was also sought to make an introduction to hardware programming through the Arduino component; a Task that took a large part of the proposed time and generated a change in the time of the other tasks.

Finally, this sprint proposes to build a data producer 1 according to the user story number 1 and the schematic propose; to complete that the Data producer 1 (DP01) requires the use of an Arduino UNO Board, a Dht11 sensor, and an LCD Screen (from Sparkfun company); all of this components was connected over a protoboard according to the electronics' standard protocol.

After the sprint the group understood that it was late on the delivery of components and needed to progress for the next sprints; also it need to get better the Scrum management

Sprint 2

For this new Sprint, we evolve to get better on the delivery and it proposes to send the Data producer 2 an MQTT Broker with their MQTT protocol tries white MQTT Mosquitto on pc

Task	duration (estimated)	task percentage
Drawing_on_LCD_screen_DP1	45 mins	100%
DP1 MQTT Pub	25 mins	100%
I2C Arduino	25 mins	100%
accelerometer_I2C_arduino	30 mins	100%
LCD I2C Arduino	15 mins	100%
I2C ESP01	45 mins	100%
accelerometer_I2C_ESP01	20 mins	100%
LCD I2C ESP01	20 mins	100%
DP2 MQTT Pub	25 mins	100%
DP2_final.ino	10 mins	100%
MCP23017 acquire information	40 mins	90%
MCP23017 Arduino	30 mins	100%
MCP23017_I2C_MCU8266	25 mins	90%
MCU8266_wifi_as_AccesPoint	20 mins	100%
MCU2866_MQTT_Broker	35 mins	100%
MQTT Mosquitto	40 mins	100%
MQTT_Broker_final.ino	20 mins	90%
Arduino connect whit ESP01	35 mins	50%
SupervisionStation MQTT Sub	15 mins	100%
SupervisionStation_wifi_as_Client	15 mins	100%

Due to the goals for this sprint that wanted to complete the user, Stories 2, 3, and 4 this sprint count with twenty task respect the advice Data producer 2 (DP01), MQTT_Broker and Supervision Station. Also, some of this task was complete; the main reason was that some bugs according to the hardware implementation; for example for DP02 a problem to connect with Wifi was for the voltage supply on ESP01 (that requires 5 volts and not the 3.3 volts on the datasheet); so that at the end of this sprint some task as DP02 final.

ino was incomplete. even so for this sprint, the group got a 96% of sprint and an 80% of the final project.

In this sprint the group already has the knowledge of hardware programming and could advance on the project; also we use components as the MCU 8266, the ESP01, an AXL, an LCD display, and an MCP23017 integrated.

Finally, the project for this sprint was most advanced but still, it needed a better Scrum presentation; also the final part (supervision Station- Chibi) was the most important, the tasks uncompleted would wait until the final sprint or more.

Sprint 3

The final Sprint was focused on the Supervision Station module that was composed of a big amount by the Chibi IPS a real-time operating system save on a raspberry PI board; to complete the tasks for this sprint (UART and I2C communication on Chibi) was required the control on the stop-time on Chibi's tasks.

The other important part of the sprint was the Arduino communication; especially the UART communication with the ESP01 device because it was an incomplete task from sprint 1. Finally, the team could complete the Sprint's task successfully and in record time; these tasks are shown below

Task	duration (estimated)	task percentage
chibi_HelloWorld	30 mins	100%
Chibi Uart	30 mins	100%
Chibi I2C master	30 mins	100%
LCD_UART_Raspberry	25 mins	100%
Arduino_I2C_slave	20 mins	100%
improve project manamegent	4 días	100%
SupervisionStation_final.ino	30 mins	100%

Even though the tasks of this sprint were completed in their entirety, the project group, at the end of the sprint, still had some tasks regarding presentation arrangements and code that would require that a final task be generated, which included the completion of the four modules and its end date is the day of delivery

Final project Conclusions

Throughout the project, the group faced certain difficulties; From the beginning, logistical problems with the formation of teams prevented it from focusing on developing one of the most important tasks of the project (the introduction to Arduino); even so, the workgroup achieved the first installment, which despite not being so large, was compensated in the second where in the end the team had managed to complete 80% of the project.

On the second Sprint, the team lose a member, although that his part was done some days ago and wasn't a problem on the sprint goals, it was a strong hit for all next sprints. for the final Sprint, the rest of the teamwork together and was easy to complete the final tasks.

Finally, it's recommended a change on the raspberry use and implementation; maybe make more focus on IOS programming through other operative systems based on Linux that use Arduino has sensor harvesting sensors and emitters.

5. Technical documentation

This section shows what corresponds to the development of the project, emphasizing the components, the construction of the requirements, and the communication structure that was used to implement the requirements.

For the realization of this project, hardware programming is required, the team uses the Arduino IDE will be used, which allows you to program Arduino boards and boards with integrated esp 8266 (which are the boards that will be used the most in this project); On the other hand, for the programming of the Chibi operating system, the visual studio code program was used to edit the files.

Each of the modules from the main schema was built over protoboard whit wire according to the electronical standard design of prototypes; the modules have different power requirements according to the final use and the ubication on the wind turbine. next show you each module, the components, the power necessities, and the final schematic.

Components

the module Data Producer 1 is composed by:

- Power Supply
  In: 12V DC Out: 3.3V or 5V DC whit switch on/off
- ESP01 (whit esp8266 module and protoboard support)
  In: 3.3v DC (Datasheet) , wifi module, UART, 2 gpio Pins
- DHT 11 module
  In: 3.3v – 5V DC module to control DHT11 sensor (3 pins GND, VCC, Data) connected by GPIO pins
- LCD screen 128X64
  In: 5V – 6.5V DC from SparkFun. UART communication using adapter component (include)

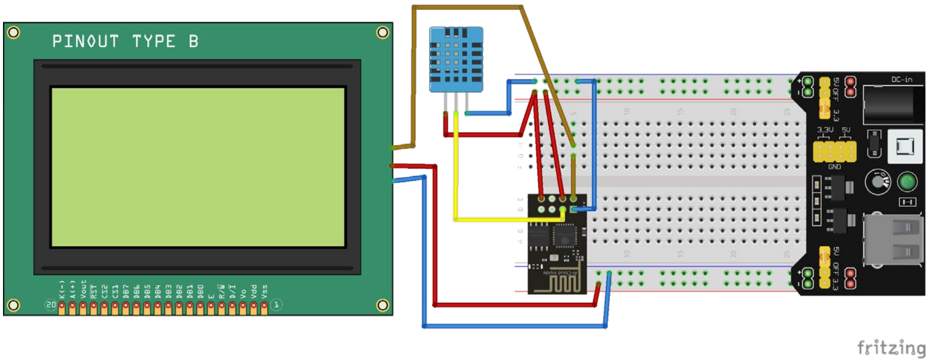


Figure 1: Data Producer 1 schematic

The module Data Producer 2 are composed by:

- Power Supply
  In: 12V DC Out: 3.3V or 5V DC whit switch on/off
- ESP01 (whit esp8266 module and protoboard support)
  In: 3.3V DC (Datasheet), wifi module, UART, 2 gpio Pins
- Adlx module
  In 2V -3.6V DC; SPI, I2C communication, 13 bit of resolution
- 16X2 LCD Display
  3.3V – 6V DC, I2C adapter component (include)

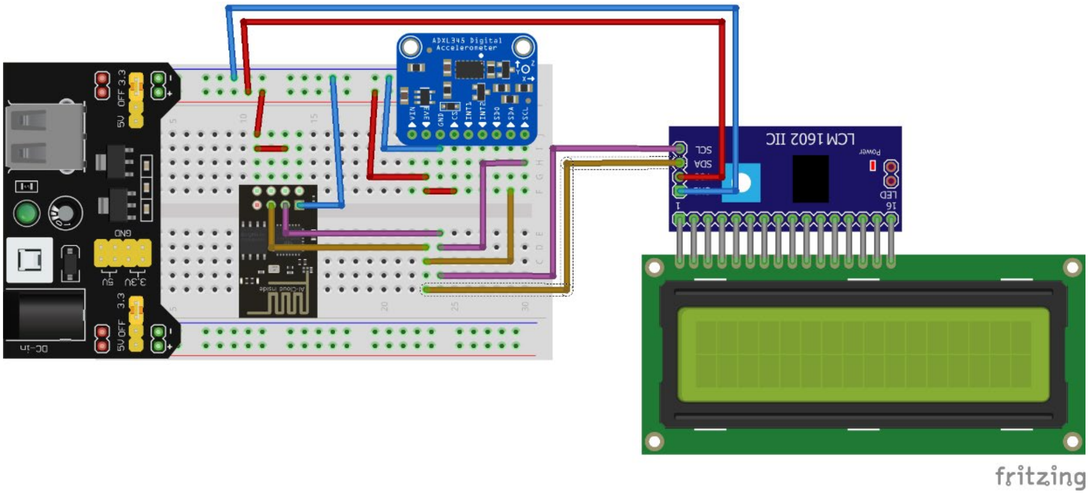
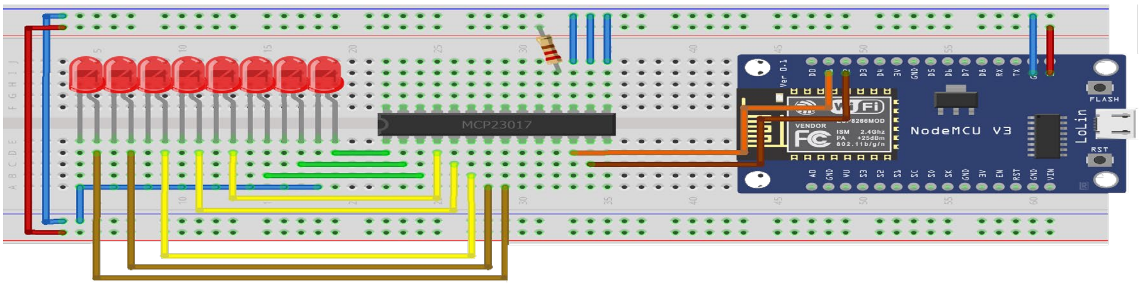


Figure 2 Data Producer 2 schematic

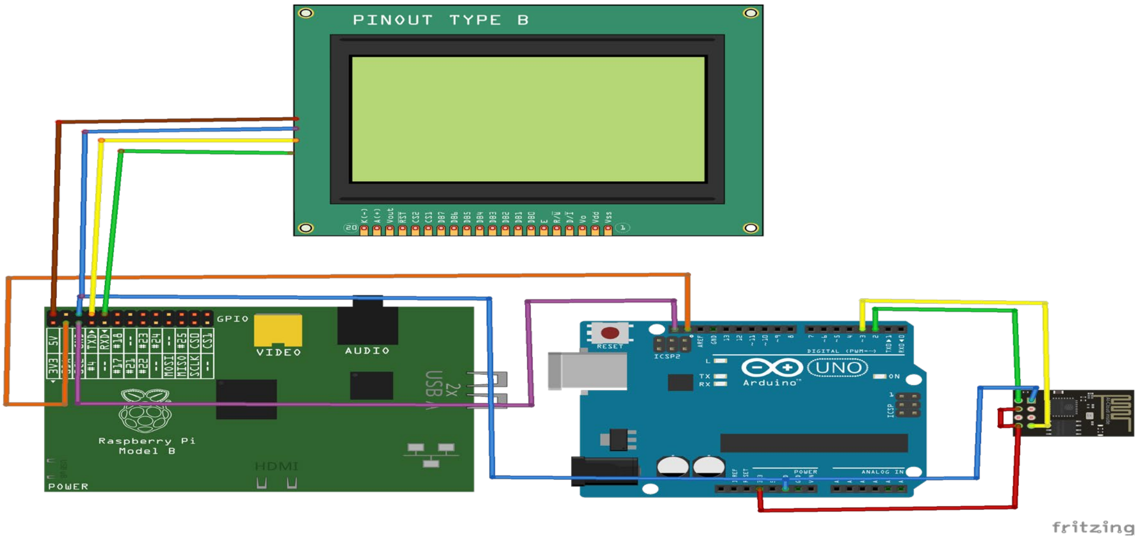
The module MQTT-Broker is composed by:

- Node MCU (whit esp8266)
  In: 5V DC, wifi module (esp8266), multiple communication protocols available multiple gpio pin
- integrated mcp2017
  multiple gpio(16 pins) controller by I2C communication, 1.8V -5.5V DC 1.7Mhz clock, cascade connection
- led´s



The module Supervision Station is composed by:

- ESP01 (whit esp8266 module and protoboard support)
  In: 3.3V DC (Datasheet), wifi module, UART Communication
- Arduino UNO
  Hardware board with ATMEGA328 microcontroller, Digital pin in/out, 9V DC input analog pin, UART I2C(master and slave) SPI communication
- Raspberry pi b series
  development target with 512Mb memory multiple ports and digital inputs, USB, multiple communication protocols requires an operating system to function
- LCD screen 128X64
  In: 5V – 6.5V DC from SparkFun. UART communication using adapter component (include)



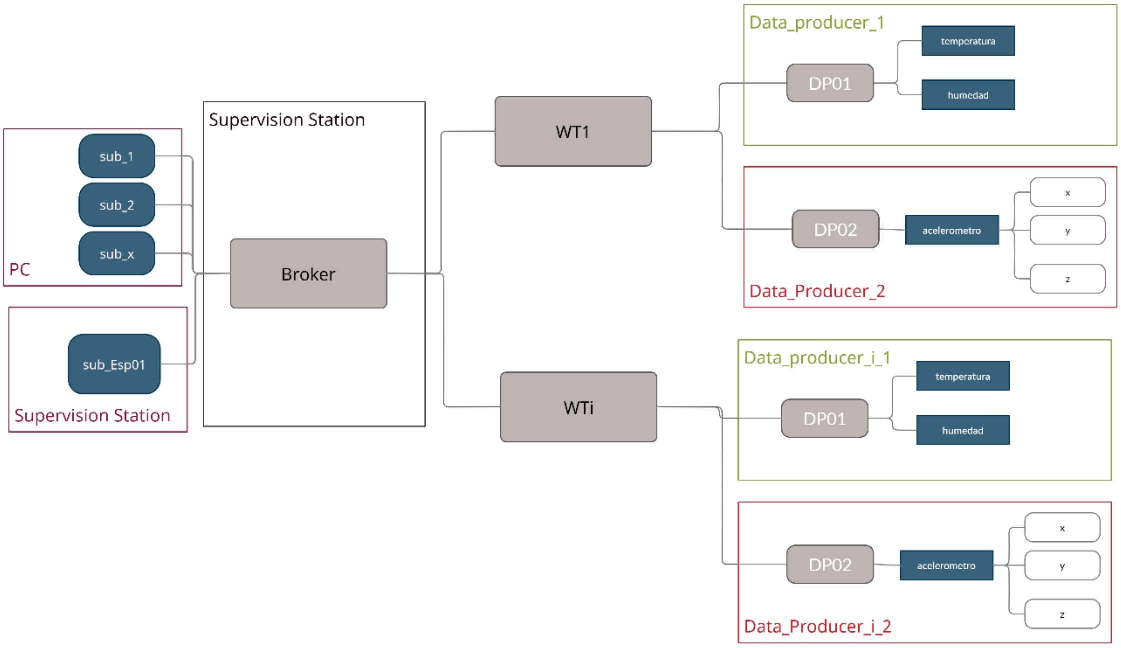
communication protocols

The communication protocols used in the project are exposed below, in addition, emphasis is placed on the method to send data via the internet known as MQTT.

MQTT method

Figure 3MQTT structure

The following is the MQTT communication structure used by the project to send information through the internet from the information receivers to the supervision station; all the data is chained to the root called "Broker"; from here they are separated in each turbine (WT_i) that contain the different Data producers for temperature and position. Finally, the receivers can subscribe to the different topics and acquire the information.



Wifi Protocol

This communication method is used to connect all the modules of the project, and it is from which the MQTT method is based to transmit information safely at the transport level.

I2C Protocol

This communication protocol was used especially on:

- Data Producer 2
  Connect the sensors and the ESP01
- MQTT Broker
  Connect the MCP23017 and the NodeMCU
- Supervision Station
  The connection between raspberry and Arduino

UART protocol

This communication protocol was used especially on:

- Data Producer 1
  Connect the 128*64 LCD Screen and the ESP01
- Supervision Station
  Connection Arduino and ESP01