ΕΡΓΑΣΙΑ

Τεχνολογίες αισθητήρων για το Διαδίκτυο των πραγμάτων

Φοιτητές:

Εμβολιάδης Αλέξανδρος

Κύρου Θεολόγος

Θέμα: Μικρογραφία Αντιπυρικού Συστήματος

Σκοπός

Η εργασία θα αποτελεί μια μικρογραφία ενός αντιπυρικού συστήματος. Το σύστημα θα αποτελείται από 2 Arduino, ένα από τα οποία θα τοποθετείται σε δασώδη περιοχή σε απόσταση d<=15Km από έναν οικισμό, και έναν στον οικισμό. Ο σκοπός του συγκεκριμένου συστήματος είναι να ειδοποιεί τον οικισμό σε περίπτωση φωτιάς στο συγκεκριμένο σημείο ή στο σπίτι και να προβαίνει αντίστοιχα σε κατάλληλες ενέργειες. Το σύστημα είναι προφανές ότι για να είναι αποτελεσματικό χρειάζεται περισσότερους αισθητήρες σε περισσότερα από ένα σημείο και για αυτό η εργασία χαρακτηρίζεται μικρογραφία.

Λειτουργία

Το ένα Arduino τοποθετημένο σε δασική έκταση, θα περιλαμβάνει αισθητήρες θερμοκρασίας, υγρασίας, και έντασης του ανέμου. Αν τα στοιχεία που δέχονται δείχνουν ενδεχόμενο πυρκαγιάς (πχ χαμηλή υγρασία ,υψηλή θερμοκρασία και μεγάλη ταχύτητα του ανέμου για τα κωνοφόρα δάση), θα στέλνει το κατάλληλο σήμα επικινδυνότητας στο άλλο Arduino χρησιμοποιώντας το δίκτυο Wifi(θεωρώντας ότι ο οικισμός και το δάσος απέχουν μεγάλη απόσταση , η χρήση του συστήματος LoRa είναι ενδεδειγμένη, ωστόσο λόγω κόστους προτιμήθηκε ένα Wifi module).

Το δεύτερο Arduino, που βρίσκεται στον οικισμό, θα αποτελείται από έναν αισθητήρα καπνού, έναν δέκτη σήματος Wifi, δύο Led, και ένα Buzer. Αναλόγως το σήμα που θα δέχεται και την επικινδυνότητα θα προβαίνει στις παρακάτω ενέργειες:

1. Ενεργοποίηση του Buzer, σε περίπτωση καπνού
2. Σήμα στο κινητό μέσω Wifi, που θα αναφέρει την επικινδυνότητα για πυρκαγιά ή την ύπαρξη πυρκαγιάς.
3. Ενεργοποίηση του αντίστοιχου Led (συναγερμού), που σημαίνει ότι υπάρχει φωτιά στον οικισμό ή έρχεται από το δάσος.

Ανάλυση Λειτουργίας του κάθε Arduino

FOREST ARDUINO:

Το Arduino, θα δέχεται δεδομένα από τον αισθητήρα Θερμοκρασίας,Υγρασίας και ανέμου σε loop, και δεδομένων κάποιον Benchmarks, θα ανιχνεύει αν υπάρχει πυρκαγιά. Το Wifi του Arduino θα βρίσκεται κατά βάση σε sleep, και θα στέλνει σήμα, μόνο εάν πληρούνται τα στοιχεία για πυρκαγιά, αλλά και μια φορά το πρωί , που θα στέλνει για να δείξει πιθανότητα πυρκαγιάς. Επίσης θα υπάρχει η δυνατότητα test, για να πάρεις τα αντίστοιχα στοιχεία των αισθητήρων εκείνη τη στιγμή,και να ελέγξεις την ορθή λειτουργία του συστήματος.

Ο λόγος που επιλέγεται αυτή η low energy consumption τακτική , είναι ότι το Arduino αυτό θα βρίσκεται σε απομακρυσμένη-δύσβατη περιοχή, επομένως θα πρέπει να ξοδεύει όσο λιγότερη ενέργεια γίνεται. Το Arduino , θα τροφοδοτείται απο 2 μπαταρίες LiPO, οι οποίες θα επαναφορτίζονται από μια φωτοβολταϊκή κυψέλη.

RESIDENT ARDUINO:

Το Arduino αυτό, καταρχάς θα δέχεται τα δεδομένα του άλλου Arduino, κάθε πρωί για πιθανότητα πυρκαγιάς (υψηλή θερμοκρασία χαμηλή υγρασία = αυξημένος κίνδυνος, κρύο και υγρασία = χαμηλός κίνδυνος), και εννοείται αν υπάρχει πυρκαγιά στην περιοχή του Forest Arduino. Σε περίπτωση δυνατού ανέμου, θα δηλώνει την μεγαλύτερη ταχύτητα επέκτασης της φωτιάς. Στην περίπτωση αυτή , θα ενεργοποιεί το Κόκκινο Led προσομοιώνοντας ένα μήνυμα στο κινητό .

Στη συνέχεια, αν πλησιάσει καπνός , ο αισθητήρας καπνού, θα το καταλάβει , δείχνοντας ότι η φωτιά έχει πλησιάσει κι άλλο, και το Arduino θα αντιδράσει δείχνοντας άλλο σήμα μέσω των Led αλλά ενεργοποιώντας και το Buzzer.

Σε περίπτωση που ανιχνευθεί καπνός, χωρίς να έχει ανιχνευθεί φωτιά στην δασώδη περιοχή, το Arduino μέσω των LED , θα στέλνει άλλο σήμα, υπονοώντας μια φωτιά από άλλη κατεύθυνση, ή ακόμη και από την ίδια την οικία.

Φυσικά , σε ένα ολοκληρωμένο τέτοιο σύστημα, με πολλά Forest Arduino, η ανίχνευση καπνού στο Resident Arduino, και η μη ύπαρξη στα υπόλοιπα, σημαίνει εσωτερική φωτιά.

Αυτό το Arduino, θα βρίσκεται στο σπίτι, επομένως δεν χρειάζεται κάποια μπαταρία ή φωτοβολταϊκό , και θα τροφοδοτείται από το ρεύμα.

Introduction

In recent years, environmental crisis shakes the whole planet. It is thought,by the scientific community, that the key driver is climate change. They also understand that climate adaptation in fire-prone countries is inadequate to deal with wildfires that are set to worsen. They also consider that, an adaptation plan to overcome climate change will take many years, because there are many factors affecting this adaptation(states, organizations, individuals etc.). In order to combine engineering with social aspects of human life, in this work, we present a micro-scale solution, for monitoring forests in Greece and alert when it is possible for a fire to break out.

The idea is simple: we will use a device with a variety of measuring instruments such as humidity,temperature and windspeed sensors. We will set some specific values in those quantities, that implies a great probability for a wildfire to occur. Also, we will establish communication between this device in the forest, and another device in an inhabited area. This establishment will alert individuals in this area in order to prevent losses in living beings and will activate some actuators to dampen the properties in this area, such as buildings, farmlands etc.

We use the characterization micro-scale for some very specific reasons: at first, the whole system can be considered as a low-cost system, in terms of components cost. Because of this,we may have a significant error in our measurements, which will probably lead to a great percentage of False Positives and True Negatives. With this in mind, an application in real-world may not be practical. Also, because of our budget, some of the effectors are abstract: we could not afford a bunch actuators that prevent properties from catching fire. Instead, we use some mechanical actuators in order to present the “activation” when values from the forest imply a great probability of a wildfire break out.

Mediterranean climate

1. *Climate in General*

Mediterranean climate describes a bunch of climate circumstances predominating in regions like California, Argentina and Mexico, Australia, South Africa and , of course, coastal regions of the Mediterranean Sea . Most common circumstances are hot and dry summers, cool and wet winters. Under the Koppen Climate Classification, "hot dry-summer" climates (classified as *Csa*) and "cool dry-summer" climates (classified as *Csb*) are often referred to as "Mediterranean". Under the Köppen climate system, the first letter indicates the climate group (in this case temperate climates). Temperate climates or "*C*" zones have an average temperature above 0 °C (32 °F) (or −3 °C (27 °F)), but below 18 °C (64 °F), in their coolest months. The second letter indicates the precipitation pattern ("*s*" represents dry summers). Köppen has defined a dry summer month as a month with less than 30 mm (1.2 in) of precipitation and as a month within the high-sun months of April to September, in the case of the Northern Hemisphere and October to March, in the case of the Southern Hemisphere, and it also must contain exactly or less than one-third that of the wettest winter month. Some, however, use a 40 mm (1.6 in) level. The third letter indicates the degree of summer heat: "*a*" represents an average temperature in the warmest month above 22 °C (72 °F), while "*b*" indicates the average temperature in the warmest month below 22 °C (72 °F).

During summer, regions of Mediterranean climate are strongly influenced by the subtropical ridge which keeps atmospheric conditions very dry with minimal cloud coverage. In some areas, such as coastal California, the cold current has a stabilizing effect on the surrounding air, further reducing the chances for rain, but often causing thick layers of marine fog that usually evaporates by mid-day. Similar to desert climates, in many Mediterranean climates there is a strong diurnal character to daily temperatures in the warm summer months due to strong heating during the day from sunlight and rapid cooling at night.

In winter, the subtropical ridge migrates towards the equator, making rainfall much more likely. As a result, areas with this climate receive almost all of their precipitation during their winter and spring seasons, and may go anywhere from 4 to 6 months during the summer and early fall without having any significant precipitation . In the lower latitudes, precipitation usually decreases in both the winter and summer. Toward the polar latitudes, total moisture usually increases; for instance, the Mediterranean climate in Southern Europe has more rain. The rainfall also tends to be more evenly distributed throughout the year in Southern Europe, while in places such as the Eastern Mediterranean, or in Southern California, the summer is nearly or completely dry. In places where evapotranspiration is higher, steppe climates tend to prevail, but still follow the basic pattern of the Mediterranean climate.

The majority of the regions with Mediterranean climates have relatively mild winters and very warm summers. However, winter and summer temperatures can vary greatly between different regions with a Mediterranean climate. For instance, in the case of winters, Los Angeles experiences mild to warm temperatures in the winter, with frost and snowfall almost unknown, whereas Tashkent has cold winters with annual frosts and snowfall; or, to consider summer, Seville experiences rather high temperatures in that season. In contrast, San Francisco has cool summers with daily highs around 21 °C (70 °F) due to the continuous upwelling of cold subsurface waters along the coast.

Because most regions with a Mediterranean climate are near large bodies of water, temperatures are generally moderate, with a comparatively small range of temperatures between the winter low and summer high (although the daily range of temperature during the summer is large due to dry and clear conditions, except along the immediate coasts). Temperatures during winter only occasionally fall below the freezing point and snow is generally seldom seen. Summer temperatures can be cool to very hot, depending on distance from a large body of water, elevation, and latitude, among other factors. Strong winds from inland desert regions can sometimes boost summer temperatures, quickly increasing the risk of wildfires. Notable exceptions to the usual proximity from bodies of water, thus featuring extremely high summer temperatures, include south-eastern Turkey and northern Iraq, surrounded by hot deserts to the south and mountains to the north. Those places routinely experience summer daily means of over 30 °C (86 °F), while receiving enough rainfall in winter not to fall into arid classifications.

As in every climatologic domain, the highland locations of the Mediterranean domain can present cooler temperatures in winter than the lowland areas, temperatures which can sometimes prohibit the growth of typical Mediterranean plants. Some Spanish authors opt to use the term Continential Mediterranenan Climate for some regions with lower temperature in winter than the coastal areas (direct translation from *Clima Mediterráneo Continentalizado*), but most climate classifications (including Köppen's *Cs* zones) show no distinction.

Additionally, the temperature and rainfall pattern for a *Csa* or even a *Csb* climate can exist as a microclimate in some high-altitude locations adjacent to a rare tropical *As* (tropical savanna climate with dry summers, typically in a rainshadow region, as in Hawaii). These have a favourable climate, with mild wet winters and fairly warm, dry summers.

2. *The case of Greece*

Greece has a predominantly Mediterranean type of climate (Köppen climate classification Csa, Csb) with warm, dry summers and mild, wet winters. Several microclimates, including oceanic (Köppen Cfb) in the coastal regions and alpine in the high mountains (Köppen ET), exist in the country. The southern islands experience a milder climate than the northern interior lands. Greece lies in southern Europe, at the southern end of the Balkan Peninsula. Turkey lies to the east, along with the Aegean and Cretan Seas. The Mediterranean Sea is in the south, the Ionian Sea in the west, and Albania, North Macedonia, and Bulgaria share the northern borders of Greece. The Pindus mountain range and the proximity to large bodies of water, including the Mediterranean Sea, chiefly influence the climate.

Greece has a mainly mountainous topography, with 80% hilly terrain on the mainland. The Pindus mountain range stretches from northwest to southeast and is an extension of the Dinaric Alps. Canyons and karsts dissect the steep mountain peaks. The Rhodope range in the far northeast is a region of thick, ancient forests. Plains with arable land mostly lie in the central region. Greece has an extensive coastline of 8498 miles and consists of more than 225 small islands. Aliakmonas, Achelous, Maritsa, and Haliacmon are among the major rivers that drain into the adjacent seas. Many Greek rivers are often shallow and challenging for navigation. Wetlands form a major part of the ecosystems near the rivers and estuaries. Crete is the largest island, followed by Euboea, Lesbos, and Rhodes. Volcanic islands are home to thermal springs. The highest point on the mainland measures 2918 meters above sea level, on Mount Olympus.

Summers are hot in the interior regions of Greece, such as Thessaly and Macedonia, but mild in the mountains. The Mediterranean coast is hot and sunny during the summer, but sea breezes moderate the temperatures. An annoying north wind occasionally blows in the Aegean Sea in the warmest months, which is dangerous for seafaring. Winters are mild and rainy along the coast, the southern islands, and the western portion of the Pindus range. The northern regions are colder than the south, with plenty of snowfall in the cold season. Cold air masses from Siberia penetrate even the central regions of Greece. The average low temperatures drop below -2.8°C (27°F) in January in Florina in the extreme north. The southern islands receive significant rainfall during winter nights and morning and are prone to high wind speeds. Snowfall on the islands occurs only on the northernmost ones, such as Thasos and Samothrace. Spring is a pleasant period, with relatively less rainfall. The skies are clear, and there is abundant sunshine everywhere. Autumn arrives late in Greece and is mild in the second half of the season. Rainfall increases by October and peaks in December in most regions.

The annual precipitation in Greece varies from 457.2mm (18") in the east to over 914.4mm (36") in the west and south. A significant portion of the rainfall occurs in the winter. The eastern side of the Pindus range falls in the rain shadow region and is drier than the western part. The annual sunshine ranges from 2300 hours in the north, to 2500 hours in the central region, and 2800 hours in the southern islands. The average humidity ranges from 62% to 74% annually and is at the highest during the summer. The longest days are in June and the shortest in December. The average sea temperatures are warm between 21.1°C (70°F) to 25°C (77°F) in the summer. The Mediterranean Sea is warmer than the Ionian and Aegean Seas.

The highest temperature on record for Greece is 48°C (118.4°F), set in Athens, on July 10, 1977. The lowest temperature on record is -27.8°C (-18°F), set in Ptolemaida, on January 27, 1963.

Vegetation Zones in Greece

The forests of Greece mostly have fir trees, pine trees, and bushes. These trees exist in the alpine parts of Greece, which are found in the northern part of the country and in high altitudes in Sterea and Peloponnese. Forests in lower altitudes mostly have poplars, plane trees, oaks, and cypress trees. At the foot of the trees, various bushes and flowers grow.

Greece is mostly a mountainous country and almost 25% of the territory is covered with forests, making it the fourth largest country in Europe with respect to forest resources. The majority of the forests in Greece are natural and not technical. The most usual trees to see in the forests in Greece are firs, Aleppo pines, black pines and other coniferous trees, while there are also large quantities of beeches, chestnuts, oaks and plane trees.

Though there are not much data for specific values of temperature, humidity and wind-speed, the elements necessary for starting a fire are oxygen,heat and fuel. We can see that there is abundance of these elements in the forests,with the trees themselves playing the role of the fuel.

However, there are some hints for specific types of forests that implying ,weather or not, a wildfire is about to break out. For example, conifer forests are known for the dryness of their trees. In major heat exposure, there is nothing stopping a wildfire from breaking out, because the trees themselves, and their nuts, are the perfect fuels, with nothing wet inside them or nearby. In addition to that, windspeed plays a major role in the spread of a wildfire along many regions. In summer 2021, at South Evoia, where great wildfires flared, windspeed played a dominant role in this major destruction. Having in mind that most forests in Greece consist of this tree-type, we need to get precise measurements of these variables in order to secure living beings nearby or , even better, prevent a wildfire from occurring.

The Human Factor

We have discussed how nature itself is compatible with wildfire breakouts, though human factor plays a significant role also. In fact, most of breakouts are occured because of human activity in the forest region(arsons, branch burning, arbitary building etc). As mentioned by Michael Mann, climate change affects the severity of the fire season and the amount and type of vegetation on the land, which are major variables in predicting wildfires.However, humans contribute another set of factors that influence wildfires, including where structures are built and the frequency and location of ignitions from a variety of sources—everything from cigarettes on the highway to electrical poles that get blown down in Santa Ana winds. As a result of the near-saturation of the landscape, humans currently are responsible for igniting more than 90 percent of the wildfires in California.The researchers found that omitting the human influence on California wildfires overstates the influence of climate change on the fires. The authors recommend considering climate change and human variables at the same time for future models. In Greece, it is mentioned that a wildfire by natural causes, breakouts every 100-150 years.Apparently, only 5% of forest wildfires occur by natural causes and the rest 95% occur due to human activity. Also, the greek state has lost count in arbitary buildings near/inside forests and only in recent years effort is being made in order to minimize these violations. To prevent loss of property and/or human lives and other living beings, we suggest this micro-scale system, in a more supported-way to the drastic measures that must be taken to somehow control the climate change, and not replace them.

1.Arduino Uno Rev3

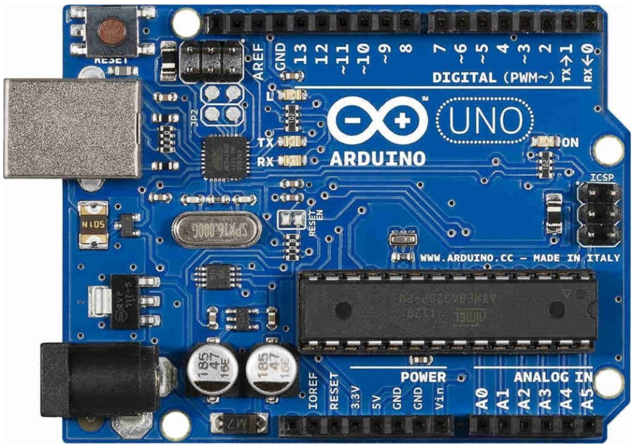


Figure 1. Arduino Uno Rev3

The board used for the development of this project is the Arduino Uno Rev3(Figure 1). The Arduino UNO is the best board to get started with electronics and the most used and documented board of the whole Arduino family. Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller.

1.1 Power

The 2 Arduino boards used in this project, have their own peculiarities regarding the power supply.

There are three ways to power an Arduino depending the projects needs:

- USB-jack
- Power jack
- Vin pin

The power supply tree, can be found on the official datasheet and is depicted in the picture below.(Figure 2)

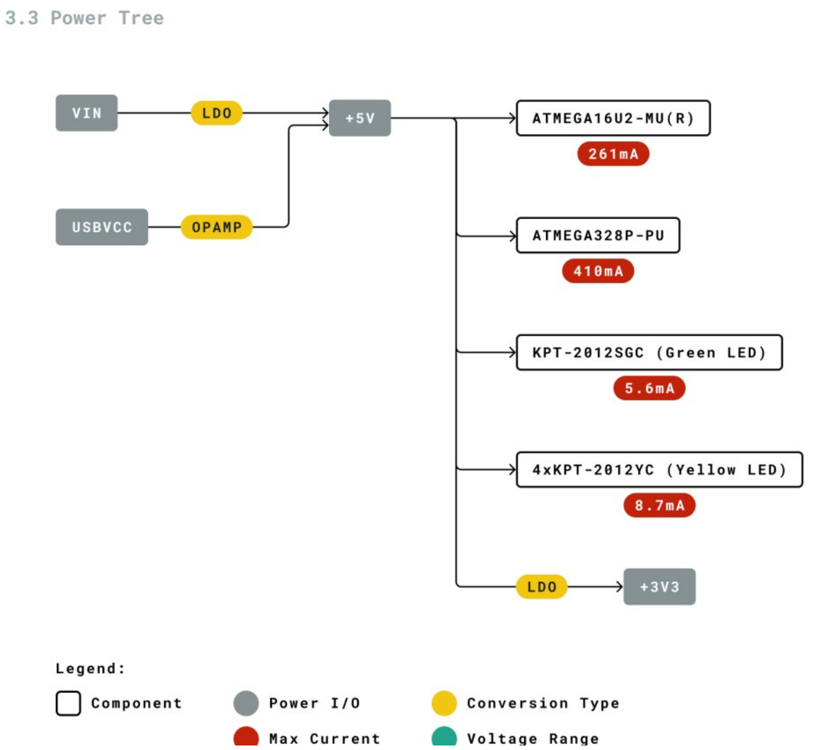


Figure 2. Arduino Uno R3 Power Tree

The **USB** connection, is mostly used in indoors systems, as it needs a computer to connect to. The computer provides the Arduino with 5V and creates somewhere around 100mA of current. This is the most convenient method to program the board.

The **Power Jack** option, needs a 7-12V DC voltage, which can be drawn from an adapter or a battery. The Arduino then, after a 1-1.8V voltage drop, seeing there is a power supply bigger than the 5V of the usb, cuts off the usb power, and through a regulator transforms the voltage to 5V. Then in parallel a regulator transforms it to 3.3V, giving to default options to the board. An advantage of this method, is that the user can draw the original Voltage and current provided by the power jack, through the Vin pin. That is convenient for sensors or actuators that need more current or voltage than the Arduino board can endure (>100mA or >5V).

The **Vin** pin, if not used as output when the the power jack is plugged in, is used as per name, input. It can be connected to a 7-12V DC voltage as well, and is regulater by the internal voltage regulator. An important note, is that there is possible to power the 3.3V or 5V pin, but more Voltage than that may destroy the pin as it bypasses the voltage regulator.

In general the Arduino board can operate at maximum current of 200mA and 7-12V. More than that the board reaches its maximum temperatures because of the regulator heat, and the board can malfunction or burn. Each pin can provide a current of 3 mA or 15 mA, depending on the pin, or receive a current of 6 mA or 9 mA, depending on the pin. They also have an internal pull-up resistor (disconnected by default) of 100 KOhm for protection.

For this project, the **Resident Arduino**, is powered through the power jack with a 9V adapter, while the **Forest Arduino** due to its remote location, it is powered through two 3.3V Lipo batteries in series, which are rechargeable with a 1W solar panel.

The pins the Arduino Uno R3 provides, can be seen on the picture below(Figure 3).

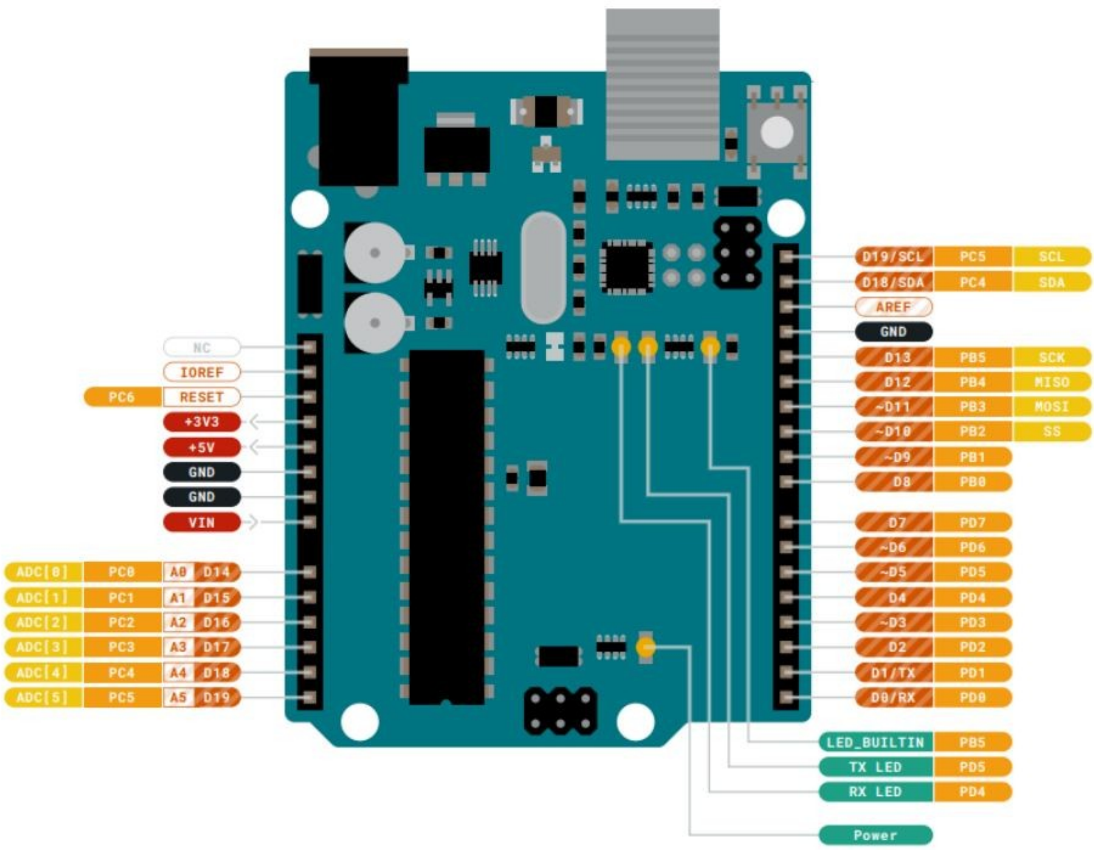


Figure 3.Arduino Uno R3 pins

1.2 Arduino Proto Shield

Arduino official Proto Shields were used with the board. Uno Proto Shield gives possibility to carry your Arduino projects from breadboard and turn them into a proper circuit and also use your specially designed circuits with Arduino. (Figure 4)

You can use proto shield without soldering. By a Mini Breadboard you placed on middle section you can do prototyping and tests without dealing with soldering. Proto Shield provides easy useage by carrying all Arduino pins to upper layer. Also there places on proto shield for surface mount devices (SMD). Therefore you can use that type of materials as well.

It also provides some extras like a reset button, 1 general use button and 2 LED circuits that can be used directly, while it has a mini breadboard attached.

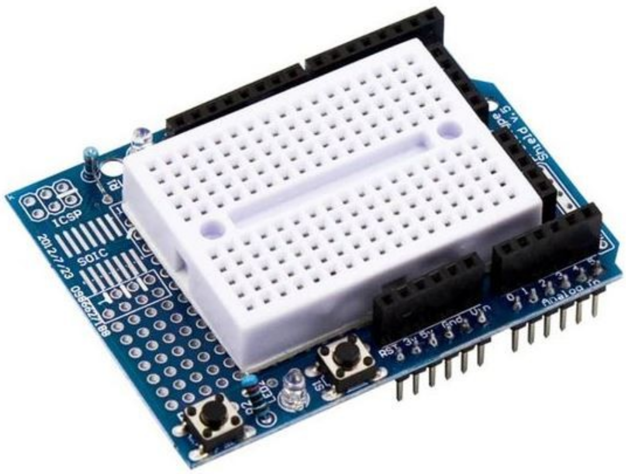


Figure 4 .Arduino Proto Shield Uno

2. Sensors

The Sensors used in the system are listed as:

- DHT11 Temperature-Humidity sensor (F)
- MQ-135 Smoke sensor (R)
- Hobby Motor 1.5V DC 18000rpm + Propeller 60mm Nylon (Hole Diameter 2mm), DIY Wind Speed Sensor (R)

,for F=Forest Arduino and R=Resident Arduino.

2.1 DHT11 Temperature-Humidity sensor

For the measurement of the Temperature and Humidity a single sensor module was used, the Waveshare DHT11 (Figure 5).

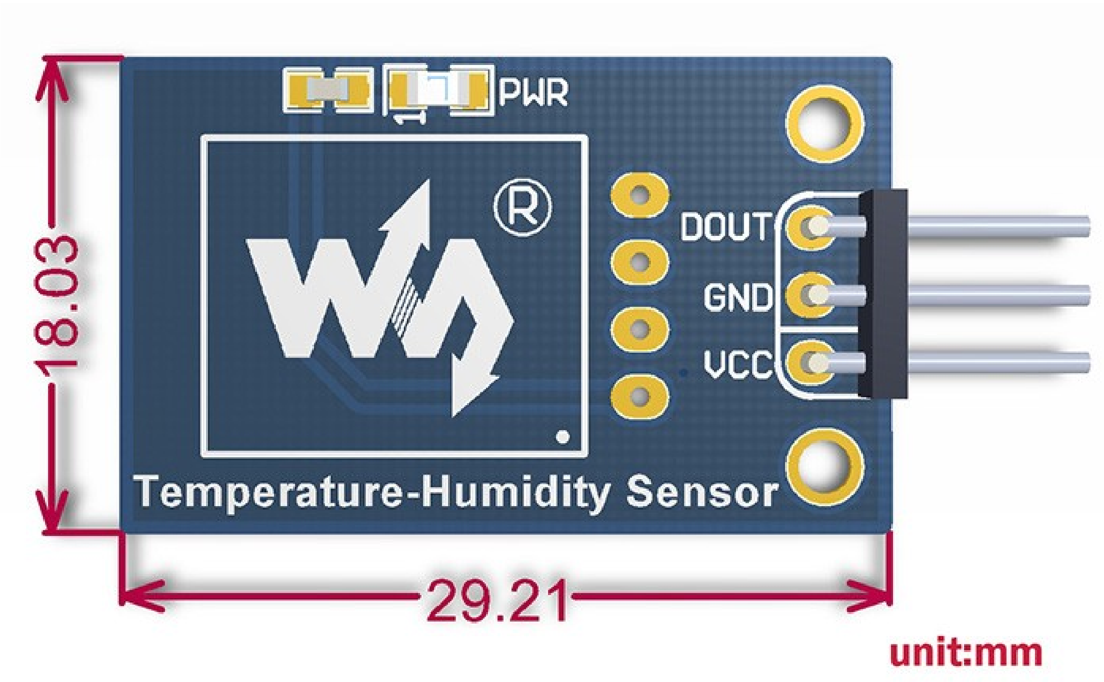


Figure 5. DHT11 Temperature Humidity sensor

The DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature. The humidity sensing capacitor has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz .i.e. it gives one reading for every second. DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

The sensor has three pins and is mounted to a small PCB. The PCB mounted version is nice because it includes a surface mounted 10K Ohm pull up resistor for the signal line. Here are the pinouts for both versions:

1.VCC - red wire Connect to 3.3 - 5V power.

2.Data out - white or yellow wire

3.Ground - black wire

The code we used with this sensor is simple as the “DHT.h” library makes it quite easy.

Taking in mind that this sensor will be attached to the Forest Arduino, it will be powered through the 5V Arduino pin, while the 0.3mA it draws won’t cause any damage to the board.

Below there is the Sketch with the connections(Figure 6).

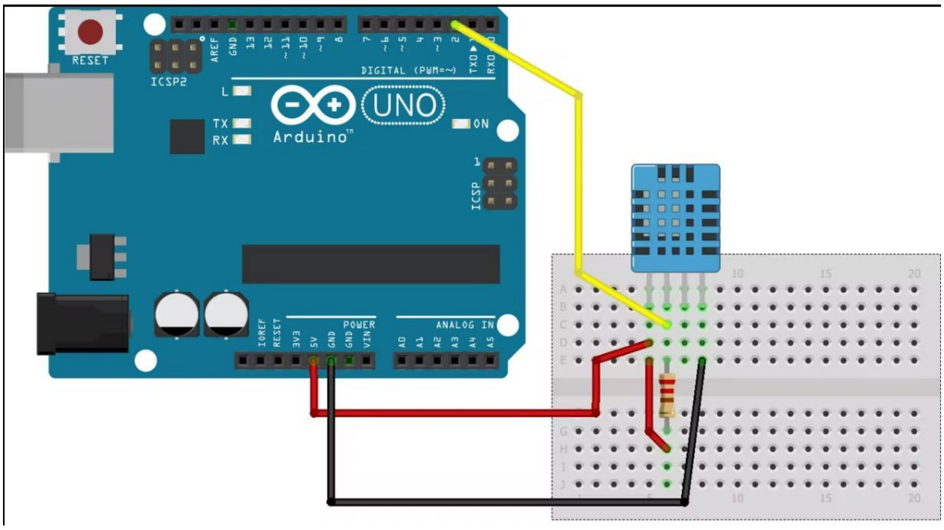


Figure 6. DHT11 - Sketch

2.2 Waveshare MQ-135 Gas Sensor

The MQ-135, just one of the MQ gas sensor series, is used to detect smoke, alcohol and benzene. In our system it will be used to detect fire smoke. (Figure 7)

Figure 7. MQ-135 Gas Sensor

The MQ series of gas sensors utilize a small heater inside with an electrochemical sensor these sensors are sensitive to a range of gasses are used at room temperature. MQ135 sensor is a Sno2 with a lower conductivity of clean air. When the target explosive gas exists, then the sensor’s conductivity increases more increasing more along with the gas concentration rising levels. By using simple electronic circuits, it converts the charge of conductivity to correspond output signal of gas concentration.

The MQ-135 gas sensor senses gases like ammonia nitrogen, oxygen, alcohols, aromatic compounds, sulfide, and smoke. The boost converter of the chip MQ-3 gas sensor is PT1301. The operating voltage of this gas sensor is from 2.5V to 5.0V. The MQ-3 gas sensor has a lower conductivity to clean the air as a gas sensing material. In the atmosphere, we can find polluting gases, but the conductivity of the gas sensor increases as the concentration of polluting gas increases.

The gas sensor layer of the sensor unit is made up of tin dioxide (SnO_2); it has lower conductivity compare to clean hair and due to air pollution the conductivity is increased. The air quality sensor detects ammonia, nitrogen oxide, smoke, CO_2, and other harmful gases. The air quality sensor has a small potentiometer that permits the adjustment of the load resistance of the sensor circuit. The 5V power supply is used for air quality sensor.(Figure 8).

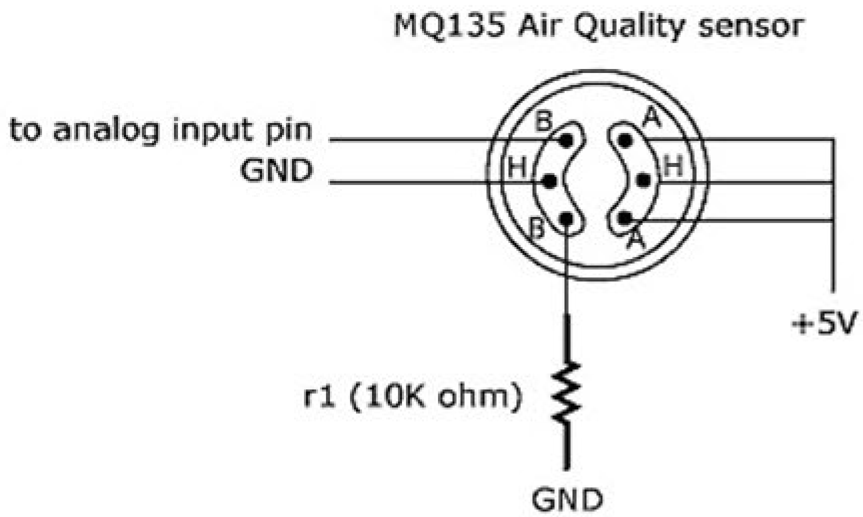


Figure 8. MQ-135 Pins

The air quality sensor is a signal output indicator instruction. It has two outputs: analog output and TTL output. The TTL output is low signal light which can be accessed through the IO ports on the Microcontroller. The analog output is a concentration, i.e. increasing voltage is directly proportional to increasing concentration. This sensor has a long life and reliable stability as well.

As the module needs 5V and around 200mA of current, it could the Arduino board, so it will be powered by the Vin pin, and through a Voltage Regulator L7805CV - 5V 1.5A with two 100μF capacitors.

The connections made with the Arduino can be seen on the Sketch below(Figure 9).

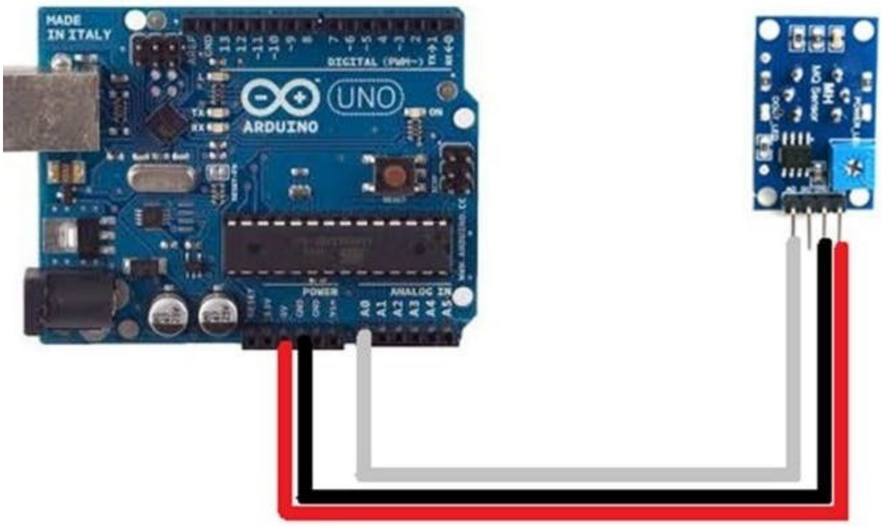


Figure 9. MQ-135 Sketch

This sensor, requires by default a 24-hour runtime, before it starts to function normal, as the heater needs to prepare. After that, on every use it needs 20 seconds of heating before it functions properly.

After experimenting and researching for other experiments, we found out that below 700 ppm there is no harm, >700 ppm there is some gas, and over 1200 there is immediate danger. The ppm is the analog read value of the sensor.

In our system, we decided to hit the alarm when the value of 700 ppm is exceeded, as the sensor will be at an outside area where the smoke cannot be concentrated easily and it depends on the air flow, when it's not of tremendous size.

2.3 DIY Wind Speed sensor

In search for the need of a Wind Speed sensor or Anemometer, that could, due to the exuberant prices of the market, we decided to make one. As the project's demands, the wind speed accuracy is not a priority, and we decided that 3 values are enough : No wind, Wind and Strong Wind.

Thus wise, we decided to use a DC motor. A DC motor converts electrical energy into mechanical energy whereas DC generator converts mechanical energy into an electrical energy. Thus, if electrical energy can turn a DC motor, mechanical energy should generate electricity.

By plugging a fan on the motor, the wind will turn it,generating voltage, which will lead to an analog read pin on the arduino, where it will be translated to measurement.

The DC motor chosen is a 1.5V voltage and has 18000rpm. We need low voltage to protect the arduino board and the pin, and the many rpm is for enabling in to turn easier with the wind force. On top of it , we put a Nylon fan.

We then had to perform experiments to map the analog voltage readings of the arduino, to actual wind speed. The experiment was, driving with a car on different speeds, with the Anemometer out of the window, while taking measurements.

Alert	Wind Speed (km/h)	Analog Read (0-1023)
-	0-10	0-40
Caution Approaching Fire	10.1 - 30	41-120
DANGER Rapid Moving Fire	>30.1	>121

The DIY anemometer can be seen in the picture below,

3. Actuators

The Actuators used in the system are listed as:

- Buzzer 5V Passive (R)
- 1 LED , simulating watering system (R)
- 2 LEDs (R)

,for F=Forest Arduino and R=Resident Arduino.

3.1 Buzzer

As an alarm on the Resident Arduino, it was decided to use a Buzzer, and a specifically a Passive Piezo 5V Buzzer (Figure 10) . This buzzer is a passive buzzer, which means that it requires an oscillating signal to produce sound. Thankfully Arduino provides libraries for that use.

Figure 10. Passive Piezo Buzzer

A piezo buzzer contains a piezo element, which is a thin disk of piezoelectric ceramic adhered to a metal plate. Both sides of the piezoelectric element have a silver electrode applied to allow for electrical contact. The whole thing can be packaged in a plastic case with or without a drive circuit.

Piezo buzzers work by leveraging the reverse piezoelectric effect — the effect where the material deforms in the presence of an electric charge. In short, a piezo buzzer works by applying an alternating voltage to the piezoelectric ceramic material. The introduction of such an input signal causes the piezoceramic to vibrate rapidly, resulting in the generation of sound waves.

The connections used can be seen on the Sketch below (Figure 11) .

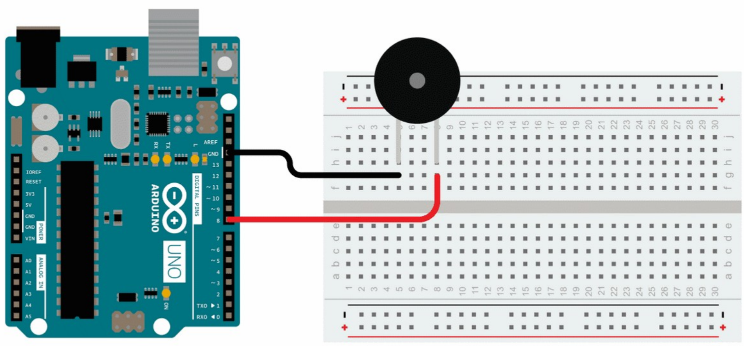


Figure 11. Buzzer Sketch

For programming the buzzer, although we do not need any external libraries, the tone() Arduino function will do it. How tone() function works?

This function generates a square wave of the specified frequency (and 50% duty cycle) on a pin by taking over one of the Atmega's internal timers. A duration can be specified, otherwise the wave continues until a call to noTone(). The pin can be connected to a piezo buzzer or other speaker to play tones.

Only one tone can be generated at a time. If a tone is already playing on a different pin, the call to tone() will have no effect. If the tone is playing on the same pin, the call will set its frequency.

Use of the tone() function will interfere with PWM output on pins 3 and 11 (on boards other than the Mega) while it is not possible to generate tones lower than 31Hz.

3.2 LEDs

In the Resident 3 LEDs will be used. One of them simulates a watering fire protect system, while the other 2 are for signaling different messages (Figure 12).

Figure 11. LEDs

The LEDs will be programmed to output useful signals to the user, to inform him of any situation. Specifically, the LED for watering, will be enabled when a fire is detected. The other 2 LEDs, will light up depending there is smoke or fire.

The LEDs, require a 3.3V voltage, and 220ohm resistors.

Below is the connections Sketch for the LEDs (Figure 13).

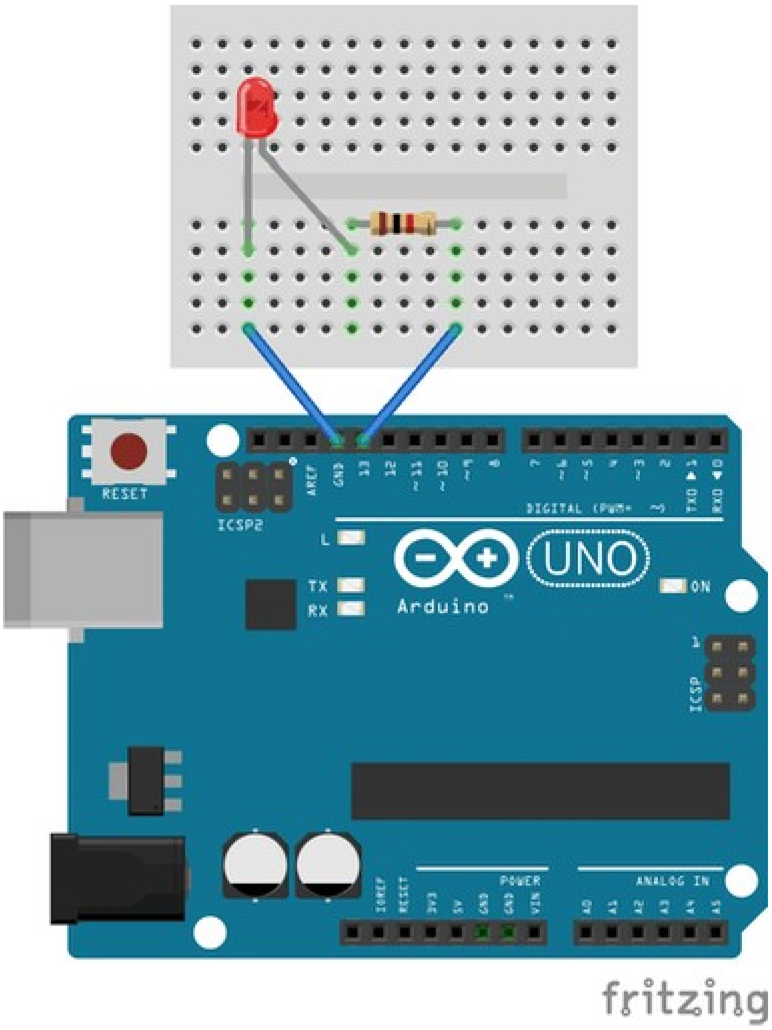


Figure 13. LED Sketch

The code used for LEDs, is pretty simple, and involves setting LOWs and HIGHs on the digital input pin of the LED.

4. Connectivity

The module used for the communication between the two Arduino Uno R3, is the ESP8266 WiFI Module.

4.1 ESP8266

The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network or act as local network itself (Figure 14).

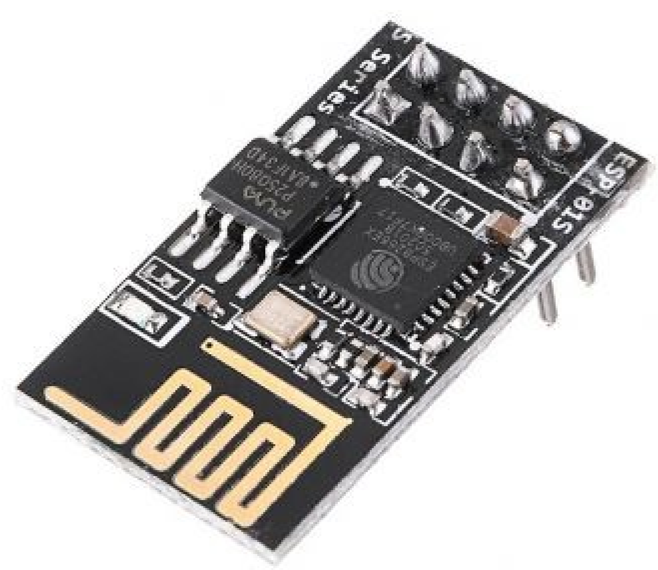


Figure 14. ESP8266 Wifi Module

The features of the ESP8266 are listed below:

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz
- Memory:
  - 32 KiB instruction RAM
  - 32 KiB instruction cache RAM
  - 80 KiB user-data RAM
  - 16 KiB ETS system-data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
  - Integrated TR switch, balun, LNA, power amplifier and matching network
  - WEP or WPA/WPA2 authentication, or open networks
- 17 GPIO pins
- Serial Peripheral Interface Bus (SPI)
- I²C (software implementation)
- I²S interfaces with DMA (sharing pins with GPIO)
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)

The pinout is as follows for the common ESP-01 module(Figure 15):

1. GND, Ground (0 V)
2. GPIO 2, General-purpose input/output No. 2
3. GPIO 0, General-purpose input/output No. 0

4. RX, Receive data in, also GPIO3
5. VCC, Voltage (+3.3 V; can handle up to 3.6 V)
6. RST, Reset
7. CH_PD, Chip power-down
8. TX, Transmit data out, also GPIO1

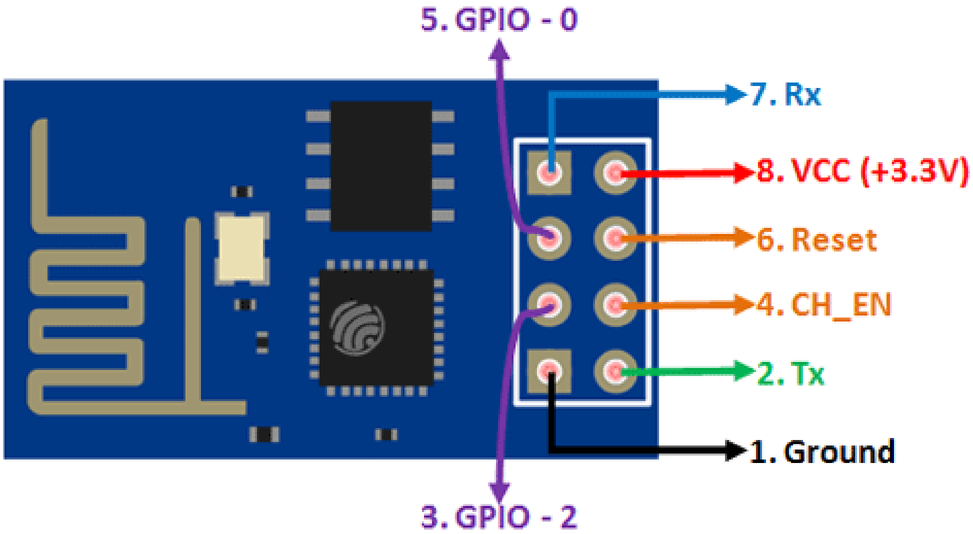


Figure 15. ESP8266 PinOut

4.1.1 Power ESP8266

The ESP8266 ,needs a 3.3V voltage and can draw up to 150mA-200mA of current. That means that the Arduino can't provide the current to enable it's adequate performance, and it might get damaged as well.

For this reason we will power them with the Vin method. On both the Forest and the Resident Arduino, we will draw voltage from the Vin pin, which with a linear regulator **Voltage regulator LD1117 - 3.3V 800mA,** will be reduced to 3.3V.

It is important to note that the ESP8266 does not have an integrated voltage regulator, so it needs exactly 3.3V to function properly, and a 5V voltage will damage it.

The LD1117 will be used with two electrolytic capacitors of 100μF for the input voltage and 10μF for the output. LD117

The wiring can be seen in the diagram below (Figure 16).

Figure 16. LD1117 Wiring

While this setup is adequate for testing this module, in real use the LD1117 will be overheated and a heat sink would be need. To keep the cost of the project low we didn't use one, as it won't affect some demo presentations.

The connections used to connect the Arduino board to the ESP8266 module can be seen on the sketch below (Figure 17).

Figure 17. ESP8266 Sketch

4.1.2 Usage

The ESP8266 has two different ways of usage. One is by sending the code from the Serial of the Arduino to the ESP8266 serial, and the other one is by programming the ESP8266 on its own. The second option, requires a different connection than the one shown above and will erase the firmware that is pre-installed on the module memory (Figure 18). It is also important to remove the ATmega328P or even better just disable it by short circuiting the Arduino RESET pin to the GND.

Figure 18. ESP8266 - Flashing Programming Sketch

What actually happens, is we disable the microcontroller of the Arduino, and we connect the Tx and Rx pins of the Arduino to the ones of the ESP8266, enabling code to pass by.

There are various ESP8266 libraries, that help programmers build different kind of servers or requests.

However, in our system we won't be using any of those, as we will use the default AT firmware with the "SoftwareSerial.h" library. Although the AT firmware is installed, we updated it to the latest version, using the connection shown above and with the software found here https://www.espressif.com/en/support/download/at

With AT commands passed through the Arduino Serial, we turned the Resident Arduino into an access point and the Forest Arduino to the client sending the request.

The ESP8266 Commands mainly include the following.

- AT+RST command is used to restart the module
- AT+CWQAP command is used to quit the AP
- AT+ CIPSTATUS command is used to obtain the connection status
- AT+CWJAP command is used to join the AP
- AT+CWMODE command is used to wifi mode
- AT+CWLAP command is used to list the AP
- AT+CIPSTART command is used to set up the connection for TCP otherwise UDP
- AT+CIPCLOSE command is used to close the connection for TCP otherwise UDP
- AT+ CIPMUX command is used to fix multiple connections
- AT+ CWSAP command is used to fix the parameters of AP
- AT+ CIPSERVER command is used to set as serve
- AT+CIPSEND command is used to send data
- AT+CIFSR command is used to obtain an IP address
- IPD command is used to receive data

The ESP8266 can draw much power when sending signals, so its use has to be careful especially to the Forest Arduino which hasn't unlimited power. There is also an available pin16 of ESP8266 which puts the arduino into Deep Sleep mode, but the microscale of the pins requires professional soldering equipment and skills, so it was not used on this project.

4.1.3 TCP/IP Protocol

The ESP8266, is using the TCP/IP protocol to communicate with other devices. This protocol has several layers of stacked technologies, which one should go under to really understand how this module works.

The Link layer

The link layer contains the physical link between two devices, an Ethernet cable, for example, or a Wi-Fi connection. This is the layer that is closest to the hardware.

To connect an ESP8266 to the network, you have to create a Wi-Fi link. This can happen in two different ways:

1. The ESP8266 connects to a wireless access point (WAP or simply AP). The AP can be built-in to your modem or router, for example.
   In this configuration, the ESP acts like a wireless **station.**
2. The ESP8266 acts as an **access point** and wireless stations can connect to it. These stations could be your laptop, a smartphone, or even another ESP in station mode.

Once the Wi-Fi link is established, the ESP8266 is part of a **local area network** (LAN). All devices on a LAN can communicate with each other.

Most of the time, the AP is connected to a physical Ethernet network as well, this means that the ESP8266 can also communicate with devices that are connected to the AP (modem/router) via a wired Ethernet connection (desktop computers, gaming consoles and set-top boxes, for instance).

If the ESP8266 is in access point mode, it can communicate with any station that is connected to it, and two stations (e.g. a laptop and a smartphone) can also communicate with each other.

The ESP can be used in AP-only, station-only, or AP+station mode. The later is the one used in this project.

TL;DR

The link layer is the physical link between devices: in the case of the ESP8266, this is a WiFi connection. The ESP can act as a station and connect to an access point, or act as an access point and let other devices connect to it.

The Internet or Network layer

Although the devices are now physically connected (either through actual wires (Ethernet) or through radio waves (Wi-Fi)), they can't actually talk to each other yet, because they have no way of knowing where to send the message to.

That's where the **Internet Protocol** (IP) comes in. Every device on the network has a personal IP address. The DHCP server (Dynamic Host Configuration Protocol Server) makes sure that these addresses are unique.

This means that you can now send a message to a specific address.

There are two versions of the Internet Protocol: IPv4 and IPv6. IPv6 is an improved version of IPv4 and has much more addresses than IPv4 (because there are much more devices than available IPv4 addresses).

The IP address consists of 4 numbers, for example *192.168.1.5* is a valid IPv4 address. It actually consists of two parts: the first part is *192.168.1*, this is the address of the local network. The last digit, 5 in this case, is specific to the device.

By using IP addresses, we can find the ESP8266 on the network, and send messages to it. The ESP can also find our computer or our phone or another ESP8266, if it knows their respective IP addresses.

What about the Internet?

As you might have noticed, I only talked about the *local* area network, these are the computers in your own house. So how can the ESP8266 communicate with the Internet, you may ask? Well, there's a lot of network infrastructure involved in 'The Internet', and they all obey the IP rules, to make sure most of your packets arrive at there destination. It's not that simple of course, there's a lot of things going on, like routing and Network Address Translation (NAT).

TL;DR

The Internet layer uses IP addresses in order to know where it should send the data. This means that two devices can now send packets of data to each other, even over the Internet.

The Transport layer

The different devices in the network do their best to deliver these IP packets to the addresses, however, it's not uncommon for a packet to get lost, so it will never arrive. Or the packet might get corrupted on the way: the data is no longer correct. IP also can't guarantee that the packets arrive in the same order they were sent in.

This means that we can't *reliably* send messages yet by only using the link and the Internet layer, since we can never know when and whether a packet will arrive, or know for certain that a received packet is correct.
We need a third layer on top of the Internet layer: the Transport layer.

There are mainly two protocols that make up this third layer: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

- **TCP** makes sure that all packets are received, that the packets are **in order**, and that corrupted packets are re-sent. This means that it can be used for communication between multiple applications, without having to worry about **data integrity** or packet loss. This is why it's used for things like downloading webpages, sending email, uploading files etc.

- **UDP** on the other hand, doesn't guarantee that every packet reaches its destination, it does **check for errors** however, but when it finds one, it just destroys the packet, without re-sending it. This means that it's **not as reliable as TCP**, but it's faster, and has a much **lower latency**, because it doesn't require an open connection to send messages, like TCP does. That's why it's used in voice and video chats, and for example in online games. We will use as said before , the **TCP protocol**.

TL;DR

The IP protocol is not reliable, and has no error checking. TCP solves this by re-sending lost or corrupt packages, and orders packets that are received in the wrong order. UDP also checks for corrupt packages, but doesn't re-send them, so it has less latency than TCP.

The Application layer

We now have reliable communication using TCP, but there's still one problem. Think of it this way: you are sending a letter, and TCP guarantees that it will arrive at its destination, but if the receiver doesn't understand the language it's written in, he won't know what to do with it.

In other words, we need a fourth layer of protocols, for two programs to be able to communicate with each other.

There's lots of different protocols out there, but we'll mostly focus on the protocols for web servers and browsers.

HyperText Transfer Protocol

The HyperText Transfer Protocol, or HTTP, is the protocol (cfr. language) that is used by both web servers and web clients in order to communicate. It uses text to perform send requests and responses from the client to the server and back again.

For example, when you type http://www.google.com into the address bar of a web browser (client), it will send an HTTP GET request to the Google web server. The server understands this HTTP request, and will send the Google webpage as a response. Or when you upload an image to Instagram, your browser sends an HTTP POST request with your selfie attached to the Instagram server. The server understands the request, saves the image and adds it into the database, sends the URL of the new image back to your browser, and the browser will add the image on the webpage.

As you can see, neither the client nor the server has to worry about the integrity of the messages they send, and they know that the recipient understands their language, and that it will know what to do with a certain HTTP request.

Most modern sites use a secure version of HTTP, called HTTPS. This secure connection encrypts the data, for security reasons. (You don't want anyone reading the packets from your mail server, or the packets you sent to your bank, for instance.)

WebSocket

HTTP is great for things like downloading webpages, uploading photos etc. but it's quite slow: every time you send an HTTP request, you have to start a new TCP connection to the server, then send your request, wait for the server to respond, and download the response. Wouldn't it be great if we didn't have to open a new connection every time we want to send some data, and if we could send and receive data at the same time at any moment we'd like? That's where WebSocket comes to the rescue: you can keep the TCP connection with the server open at all times, you get perfect TCP reliability, and it's pretty fast.

Open Sound Control

HTTP and WebSocket both use TCP connections. What if you want lower latency? Well, Open Sound Control, or OSC, uses UDP to send small pieces of data, like ints, floats, short text etc ... with very low latency. It was originally designed for controlling low latency audio applications, but it's a very flexible protocol, so it's often used for low-latency tasks other than audio control.

Briefly explained, this is the stack ESP8266 uses to ensure communication with other devices.

5. Power Consumption

One of the most important factor of a successful IoT system, is its sustainability, and that often means in terms of energy. In our system the most vulnerable part is obvisously the Forest Arduino, as it has no energy source through cable and is on a harsh environment.

5.1 System Consumption

Below are listed the Voltage and the Current its component of the system needs.

	3.3V	5V	Current
ESP8266	☑		40mA-200mA
DHT11		☑	<2,5mA
MQ135		☑	<150mA
Hobby Motor	☑		~
Buzzer		☑	<32mA
LEDs	☑		220Ω

If we divide them to the Resident and Forest Arduino accordingly, and calculate the amount of time used then we can calculate the total amount of current need for each.

Although the Resident Arduino has unlimited power supply, from the house, we will calculate it as well.

According to our plan ,the Forest Arduino will take measurement and send measurement every hour. On the rest of the time it will be on DeepSleep mode, with the build it Arduino function DeepSleep(), consuming 40μA.

Every hour, it will use wake, to use the DHT11, consuming 2,5mA, reading the wind with no cost, and around 40mA to make the calculations through the micro controller. Then the information will be send through the ESP8266 which can draw up to 200mA and averaging at 170mA and 20μA in DeepSleep mode. The estimated usage time is around 20 seconds for the ESP8266 to make sure there is connection and send the message until sleeping again. So in total the Forest Arduino uses = (170mA + 2,5mA +40mA) / (60 * 3) + 20μA + 40μA = 1,20 mA/hour.

Thereby, the 2-LiPo's of 1800mAh in series proposed for the system, are enough for it to last 900 hours or 37,5 days until the 50% undervoltage limit. Considering they will be recharged by the 1W solar panel, we have a well sustainable system.

The Resident Arduino current consumption is = (170mA + 150mA) = 320mA/hour, but it won't have a power sufficiency problem since it will be plugged on the home circuit.

5.2 Energy Sufficiency

As previously mentioned, the Forest Arduino will be powered with a rechargeable self-sustainable system, consisted of 2x3.7V LiPo batteries, and a 1W solar panel. The charging and the output to the Arduino, is calibrated by a Solar lithium battery charger CN3065.

Batteries

The two alternatives for the batteries, are the traditional well tested lead acid batteries and the more advanced but more dangerous if not used properly Lithium-ion batteries. As their names imply, lithium-ion batteries are made with the metal lithium, while lead-acid batteries are made with lead.

With these differences in chemistry come differences in performance and cost. While both lithium-ion and lead acid battery options can be effective storage solutions, here’s how they stack up when compared head to head in key categories:

	LITHIUM-ION	LEAD ACID
Cost		X
Capacity	X	
Depth of discharge	X	
Efficiency	X	
Lifespan	X	

In most cases, lithium-ion battery technology is superior to lead-acid due to its reliability and efficiency, among other attributes. However, in cases of small off-grid storage systems that aren’t used regularly, less expensive lead-acid battery options can be preferable.

In detail: how do lithium-ion and lead acid batteries compare?

Lithium-ion and lead acid batteries can both store energy effectively, but each has unique advantages and drawbacks. Here are some important comparison points to consider when deciding on a battery type:

Cost

The one category in which lead acid batteries seemingly outperform lithium-ion options is in their cost. A lead acid battery system may cost hundreds or thousands of dollars less than a similarly-sized large scale lithium-ion setup – lithium-ion batteries currently cost anywhere from $5,000 to $15,000 including installation, and this range can go higher or lower depending on the size of system you need.

While lead acid batteries typically have lower purchase and installation costs compared to lithium-ion options, the lifetime value of a lithium-ion battery evens the scales. Below, we'll outline other important features of each battery type to consider, and explain why these factors contribute to an overall higher value for lithium-ion battery systems.

Capacity:

The Tesla Powerwall 2 – one of the most popular lithium-ion solar batteries

A battery's capacity is a measure of how much energy can be stored (and eventually discharged) by the battery. While capacity numbers vary between battery models and manufacturers, lithium-ion battery technology has been well-proven to have a significantly **higher energy density** than lead acid batteries. This means that more energy can be stored in a lithium-ion battery using the same physical space. Because you can store more energy with lithium-ion technology, you can discharge more energy, thus power more appliances for longer periods of time, which is of great significance on our IoT system.

Depth of discharge

A battery's depth of discharge is the percentage of the battery that can be safely drained of energy without damaging the battery. While it is normal to use 85 percent or more of a lithium-ion battery's total capacity in a single cycle, lead acid batteries should not be discharged past roughly 50 percent, as doing so negatively impacts the lifetime of the battery. The superior depth of discharge possible with lithium-ion technology means that lithium-ion batteries have an even higher *effective capacity* than lead acid options, especially considering the higher energy density in lithium-ion technology mentioned above.

Efficiency

Just like solar panel efficiency, battery efficiency is an important metric to consider when comparing different options. Most lithium-ion batteries are 95 percent efficient or more, meaning that 95 percent or more of the energy stored in a lithium-ion battery is actually able to be used. Conversely, lead acid batteries see efficiencies closer to 80 to 85 percent. Higher efficiency batteries charge faster, and similarly to the depth of discharge, improved efficiency means a higher effective battery capacity.

Lifespan

Batteries are also similar to solar panels in that they degrade over time and become less effective as they age. Discharging a battery to power your home or appliances and then recharging it with solar energy or the grid counts as one “cycle”. The numbers vary from study to study, but lithium-ion batteries generally last for several times the number of cycles as lead acid batteries, leading to a longer effective lifespan for lithium-ion products.

When should you install a lead acid battery vs. a lithium-ion battery?

If you need a battery backup system, both lead acid and lithium-ion batteries can be effective options. However, it’s usually the right decision to install a lithium-ion battery given the many advantages of the technology – longer lifetime, higher efficiencies, and higher energy density. Despite having higher upfront costs, lithium-ion batteries are usually more valuable than lead-acid options.

One case where lead-acid batteries may be the better decision is in a scenario with an off-grid solar installation that isn’t used very frequently. For example, keeping a lead-acid battery on a boat or RV as a backup power source that is only used every month or so is a less expensive option than lithium-ion, and due to the lower usage rate, you’ll avoid many of the drawbacks of lead-acid technology, such as their shorter lifespan.

Temperature

Another advantage of the LiPo batteries is their survivability on high temperatures. Extreme heat speeds up the chemical reaction inside a lead acid battery and causes an increase in the self-discharge and plate corrosion. This leads to sulfation which can cause irreparable damage to the battery. For each 10°F rise in temperature, the life of a sealed lead acid battery is cut in half. On the other hand, lithium’s performance is far superior in high temperature applications, In fact LiPos at 55’ C, still has twice the cycle life as lead acids. Lithium will outperform lead under most condition but is especially strong at elevated temperatures.

Summary

Considering that in our full scale system, there will be need for several Forest Arduino, the LiPo battery is the best choice in terms of capacitiy, efficiency and of course of lifespan, as some of the locations the Arduinos will be spread won't be as easily accessible as others. Another important factor, are the high temperatures this outside system may experience, and the better performance of LiPos under those temperatures. LiPo batteries of course carry an additional cost, but it will be less than the cost of the usual repairing or damage lead acid batteries will cause.

Thus wise, we chose LiPo batteries(Figure 19). To keep the cost of this demo low, we purchased 2x 3.3V LiPo batteries, which are connected in series to reach a voltage of 7.4V, capable of powering both the arduino and the ESP8266. Altough we made the rechargeable system, it needs around 1800mAh to work properly and sufficiently, which adds up the cost, so we purchased 240mAh batteries for making a demo Solar charging system which will power a LED. Accordingly the arduino will be powered with a 9V Alkaline battery for the demo, plugged on the arduino power jack with the equivalent adapter(Figure 20).

Figure 19. LiPO 3.3V batteries

Figure 20. Power Jack Battery Adapter

Solar Panel

The solar panel chosen for the Forest Arduino system was chosen to be compatible with the batteries and the solar charger. So it was decided upon a 1W 125x63mm Solar panel with 6.9 open circuit voltage. On the large scale we could go with a higher Watt panel but to keep the demo cost low we went for the 1Watt (Figure 21).

Figure 21. Solar Panel 1W 125x63mm

Solar Charger

The chosen Solar Charger to regulate the whole process of the panel charging the battery and the battery powering the system, will happen with a cheap bought module the Solar Lithium Battery Charger - CN3065.

This is a super mini Solar Lipo charger based on the CN3065 - a single lithium battery charge management chip.

This Solar charger provide you with the ability to get the most possible power out of your solar panel or other photovoltaic device and into a rechargable LiPo battery. Set-up is easy as well, just plug your solar panel into one side of the Solar charger and your battery into the other and you are good to start charging!

The output of the Solar Charger is intended to charge a single polymer lithium ion cell. The load should be connected in parallel with the battery. By default, the Solar charge comes set to a maximum charge current of 500mA with a maximum recommended input of 6V (minimum 4.4V). It's recommended that batteries not be charged at greater than their capacity rating.

6. Cost and Market

6.1 Cost

The cost of this project can be seen on the table below.

	Price	Quantity	Cost
Arduino Uno Rev3	17,74 €	2,00 €	35,48 €
ESP8266 WiFi Module	4,68 €	2,00 €	9,35 €
LiPo battery 3.3V 270mAh	4,19 €	2,00 €	8,38 €
Proto shield Arduino	3,06 €	2,00 €	6,12 €
Solar Charger CN3065	3,15 €	1,00 €	3,15 €
Solar Panel 1W 125x63mm	3,95 €	1,00 €	3,95 €
DHT11 Waveshare Sensor	3,15 €	1,00 €	3,15 €
LED	0,16 €	4,00 €	0,64 €
Hobby Motor DC 1.5V	0,65 €	1,00 €	0,65 €
Wiring	8,00 €	1,00 €	8,00 €
MQ135 Gas Sensor Waveshare	5,32 €	1,00 €	5,32 €
Buzzer 5V Passive	0,48 €	1,00 €	0,48 €
Electrolytic Capacitors	0,04 €	4,00 €	0,16 €
Voltage Regulators	0,40 €	2,00 €	0,80 €
Breadboards	2,50 €	2,00 €	5,00 €
Power Jack Battery	1,08 €	1,00 €	1,08 €

Holder			
9V Battery	5,00 €	1,00 €	5,00 €
Resistors	0,50 €	1,00 €	0,50 €
Casualties Costs			20,00 €
Shipping Costs			15,00 €
Equipment & Tools			40,00 €
Total Cost			**171,71 €**

The Total Project cost ,with casualties, shipping and equipment was calculated at 171,71€. If moving into production then the cost for the Resident Arduino would be around 40€ , while the Resident Arduino at 48€ which can be reduced to 34€ if we use the atmega328p instead of the Arduino for production purposes. Also, in production mode, LoRa modules will be used instead of the Wifi, adding the cost to 32€ (R) and 40€ (F), and covering a signal range of 5km.

Therefore, we can see on the table below the cost of this system, depending the kilometers we need to cover, using 4 points and 8 points(Figure 23).

Table 1: 4-points Range - Cost table

km	Cost
5	192.00 €
10	312.00 €
15	432.00 €
20	552.00 €
25	672.00 €
50	1,272.00 €

Table 2: 8-points Range - Cost table

km	Cost
5	352.00 €
10	672.00 €
15	992.00 €
20	1,312.00 €
25	1,632.00 €
50	3,232.00 €

Figure 23: 4-point 1km System (R = Resident Arduino, F = Forest Arduino

6.2 Market Competition

As common as indoors fire detection systems may be, market lacks of outdoors fire detection systems. This is due to several factors, one of which is that the best method to monitor for wildfires is through satellite.

Although the best method, live-time satellite surveillance costs a huge amount of money which are available to nations and huge companies, so a more traditional sensor system such as ours, comes handy for low budget residents or companies with high wildfire risk.

After market research the main market distributor of outdoors fire detection systems Is Solo – NoClimb, and their prices vary between 1.500 – 2.500 euro, for a range of 5km and is consisted with a single unit, which using a 9m pole can detect heat and smoke.

As we can see our IoT system offers, a more safe approach by spreading sensors along a forest with a much lower price.

7. Future Development

The most important factor of a fire detection system, is reaching the information or warning to the people the fastest possible. That is why, the buzzer and the LEDs are not enough, and a Mobile application is essential.

The information received from forest Arduino,and to the Resident Arduino, can be sent to a server if the Resident Arduino is connected to the Resident Wifi. Then it can alert the user to a mobile app wherever he might be.

We actually managed to sent the sensors reading to a server using the online tool from https://thingspeak.com/ (Figure 24). Of course those data can be compared to open weather API data, and with the use of neural networks techniques, a model can be constructed for detecting wildfires faster or trying to predict them.

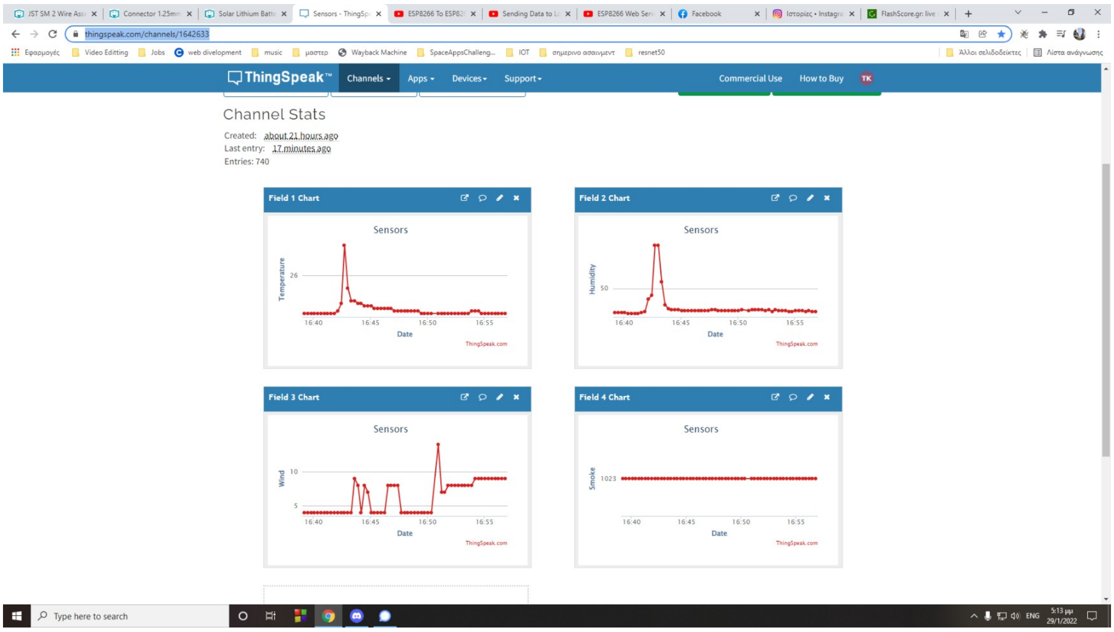


Figure 24: ThingsSpeak Reading for Sensors

8. Code

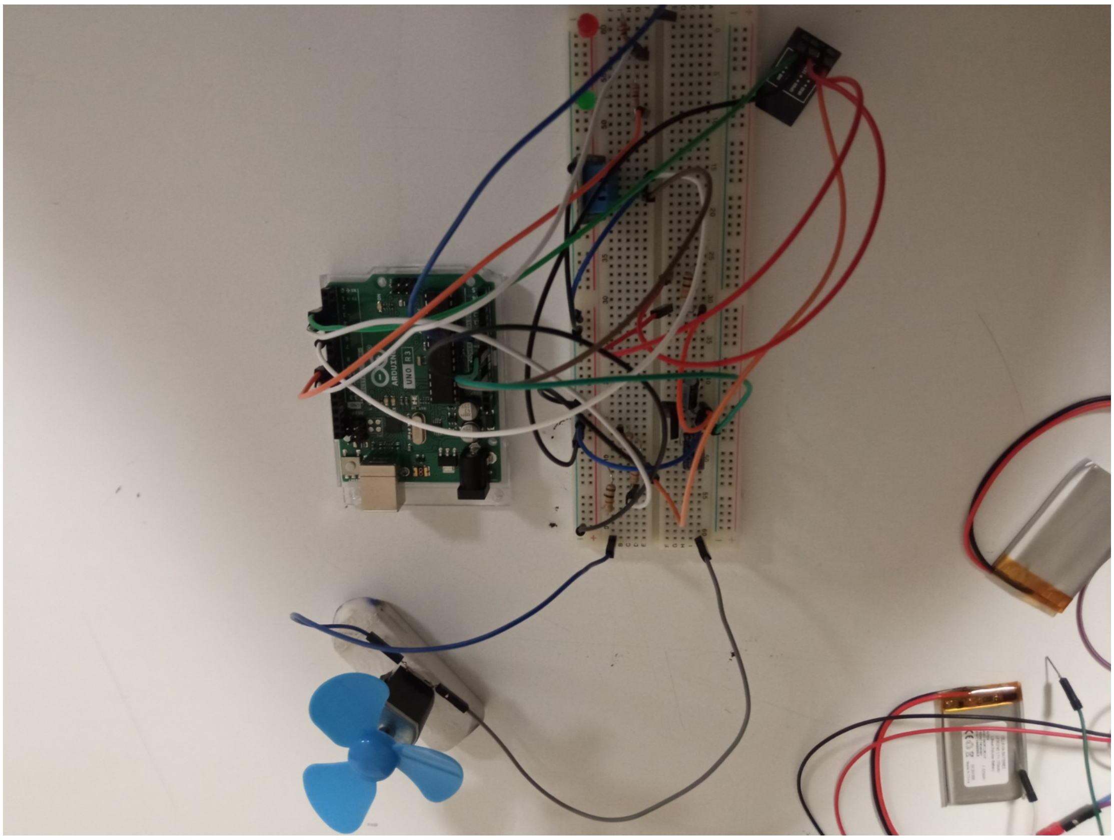
The system was programmed using the Arduino IDE, and the all the programming is on the Arduino.

The ESP8266 modules, although can be programmed, we used the AT default firmware they use. This was done by sending commands from the Arduino Serial, to the espSerial to run.

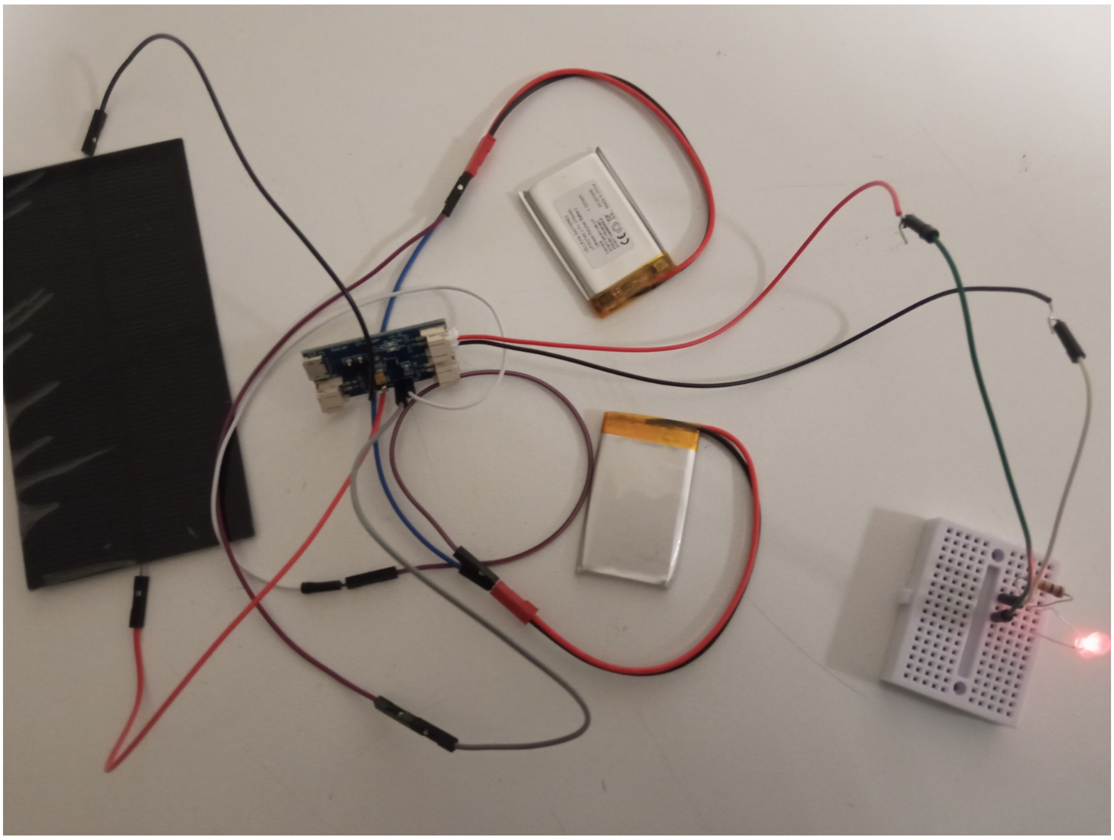
The whole code is deposited on https://github.com/kyrouT/Arduino-Outdoor-Forest-Detection-System .

9. Project Testing

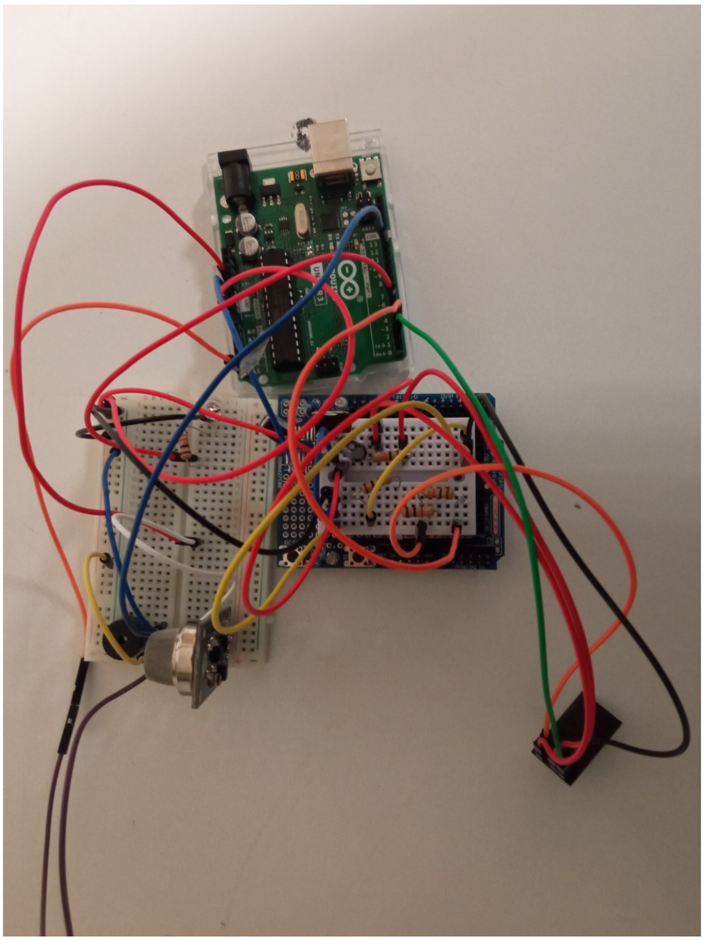
The system was successfully assembled and tested. You can see it on the pictures below.



Forest Arduino



Solar Recharging System



Resident Arduino