

# An Automated Plant Pot Controlled via the Internet based on Arduino Applications

A. Chatzopoulos  
Dept. of Industrial Design  
and Production  
Engineering,  
University of West Attica,  
Athens, Greece

M. Papoutsidakis  
Dept. of Industrial  
Design and Production  
Engineering,  
University of West Attica  
Athens, Greece

Giannis Sofianos  
Dept. of Industrial  
Design and Production  
Engineering,  
University of West Attica  
Athens, Greece

Dimitrios Piromalis  
Dept. of Industrial Design  
and Production  
Engineering,  
University of West Attica  
Athens, Greece

## ABSTRACT

The aim of this paper is to develop a low-cost Arduino based automatic pot that performs the following functions: (i) monitor plant's living conditions, such as temperature, humidity, environment, and soil moisture, (ii) send plant's collected data via the Internet to a properly configured webpage so the user can be able to remotely monitor them, (iii) automatic control plant soil's humidity to activate automatic watering when humidity levels reach minimum set values, (iv) remote and/or manual control of the plant's lighting and watering. Project's main purpose is to absolve the user from his daily employment for the plant's maintenance and strengthen plant living conditions for its optimal growth while keeping equipment cost at a minimum. An Arduino Mega 2560 is selected to operate as a plant pot's data logger, an automation control unit and a web server [1], [2]. An Ethernet shield attached on it provides the necessary hardware to connect the microcontroller (MCU) to the Internet. Arduino webpage's code and data acquisition are stored in a micro SD card attached to shields' card slot. Appropriate temperature and humidity sensors connected to Arduino used for data acquisition to later forwarded as XML files to the webpage.

## Keywords

Arduino, Automated pot, IoT, Internet of Things, Internet

## 1. INTRODUCTION

Shaping and flowering of pot plants involve manual work in order to obtain healthy plants [3]. On the other hand, automation is all about the use of machines, control systems and information technologies to optimize productivity in the production of goods and delivery of services [4], [5]. This paper's fully automated pot integrates automation into a regular pot in order to obtain healthy plants without the effort of user manual work. It cares about the plant's watering, lighting, temperature, while at the same time monitoring plant's living conditions data and presenting them remotely to the user via the Internet. As a bonus, user can have remote access to his plant and to its automated pot. The automated plant pot provides the following specifications:

1. An automated (or manual) plant's watering depending on soil moisture.
2. An automated (or manual) plant's lighting.
3. Monitoring of plant's living condition such as temperature, air humidity, soil moisture.
4. A simple IoT allowing the user to monitor and control automated pot in real-time and get flexible analytic reports. The plant's living condition data

presenting on a clean and simple user interface webpage with a dashboard.

5. Real-time user's alarms when something is going wrong.
6. User can access the IoT platform by any device (smartphone, tablet, personal computer) equipped with Internet connectivity. No need for a specialized application or software installation. Works with any browser.

Automated plant pot's main objective is to provide optimum plant growth while at the same time minimize user's care, so his only supervision will be to identify any plant disease or insects.

## 2. MAIN CONCEPT

The development of this automated plant pot is based on the use of a three-level pot (Fig. 1: Automated pot plant's schematic). At the upper level, there are the pot's plant, the temperature, air's humidity, and soil's humidity sensors. At the top of the pot, there is a lamp powered by light emitting diodes (LEDs) which provides lighting to the plant. LEDs could provide specific quality and quantity of light overcoming existing limitations for normal plant growth. A combination of a percentage of red, blue and white LED light was chosen for optimal plant growth [6]. At pot's middle level there is a water tank and a water pump that provides plant's watering. And finally, at the lowest level of the automated pot there are the electronic parts of the device, namely Arduino MCU, power supply and, wirings.

More specifically: The third level is watertight from the upper levels to prevent damage from water leakage and inside it hosts the electronic hardware (Arduino Mega, power supply, wirings). On its sides, there are ventilation gaps for cooling electronic parts from overheating through the natural, fresh, air flow. On its left side, there are the USB, Ethernet (RJ-45) ports, and power cable that powers the automated plant's pot. USB port is useful for firmware and software future upgrade. At the present time, the plant's pot Internet connection is implemented through the connection between Ethernet port (RJ-45 connector) and home's router. In the distant future, a wireless interface will be used to achieve WiFi connectivity. Power cord is connected to home's electrical supply (220V – 240V AC @ 50Hz).

The middle level of the pot stores plant's water into small tank water and a submersible water pump is used to transfer water to the upper level when the automated pot starts plant's watering. It communicates with the upper level via a gap

equipped with a three-stage filtering system to drain plant's excess water. The three-stage filtering system consists of a:

1. wire mesh (to support upper level's drainage materials),
2. fiberglass wool (for mechanical filtration),
3. carbon filter (for chemical cleaning)

Any excess water that is not absorbed by the plant or its soil it will be returned filtered to the water tank, so as to optimize the water economy.

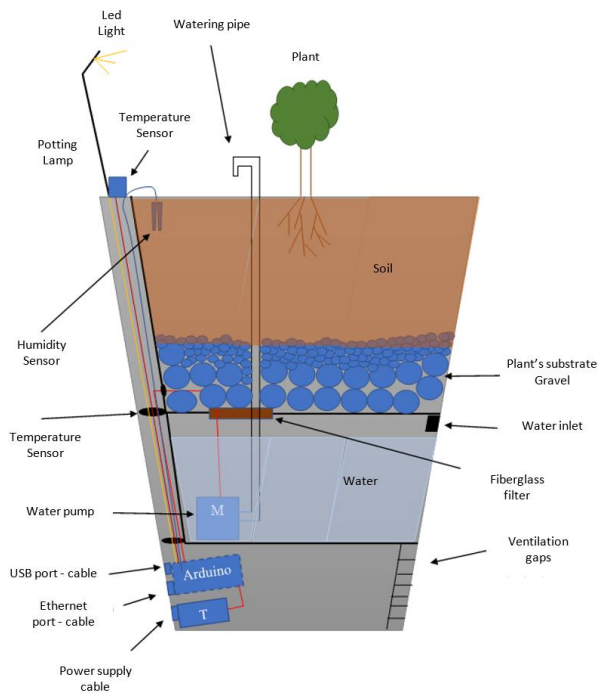


Fig. 1: Automated pot plant's schematic

The upper level of the pot contains water drainage materials such as small and medium size gravel and the necessary soil to support the plant. At top right there is the water inlet for plant's watering. Particular care was given during the construction of the automated pot so that the upper layer materials do not penetrate into the middle level. Hence, small size gravel was used to support the plant's soil so as not to pass through the filtering system and medium size gravel was used to support the small size one for the same reason. Finally, medium size gravel is supported into its place through the wire mesh. Furthermore, at the top of the pot, humidity's, temperature's, moisture's sensors, and the potting lamp were placed. Sensor's wirings and water pipe are coming through a watertight part of the pot and connected to the third and middle level respectively. Sensor's wirings are connected through interfaces to the MCU (Arduino Mega).

Automatic watering is achieved by real-time monitoring of moisture's and humidity's sensors. When they reached the user's specified lower values, then the MCU powers the water pump to watering the plant. The amount of water is also specified by the user. A future automated plant pot's upgrade will be to recall plant's living condition data from the Internet so to bypass user settings or even co-count them.

### 3. HARDWARE

The following figure (Fig. 2) presents the automated pot's electronic circuit. The main part of the circuit is the MCU (Arduino Mega) which is connected to the Ethernet shield, plant's sensors, and water pump.

Arduino Mega was selected because of its low price, good specifications and its ability to connect to the Internet, and operate as a web server in order the user to be able to remotely monitor plant's living conditions data. Furthermore, Arduino is the world's leading open-source hardware and software ecosystem. Maybe a future automated pot's implementation could use the cheaper Arduino Uno [7].

The selected Ethernet shield was Arduino Ethernet Shield 2 which allows an Arduino to connect to the Internet. It is based on the Wiznet W5500 Ethernet chip (IEEE802.3af compliant). The Wiznet W5500 provides a network (IP) stack capable of both TCP and UDP and it supports up to eight simultaneous socket connections [8]. The Ethernet shield is connected to Arduino Mega using its long wire-wrap headers extending through it. It has a standard RJ-45 connection, with an integrated line transformer and Power over Ethernet enabled. There is also an onboard micro-SD card slot, which it was used to serve webpages' files (HTML, CSS, JS code) for serving over the network, and to store plant sensors' data [9].

The original Arduino's Ethernet [10] and SD libraries [11] were used in order to operate smoothly the Ethernet shield. The use of Arduino Mega as a web server was achieved by appropriate code in C language [12], while plant's website UI (user interface) was created using HTML, CSS, and JavaScript code, stored on the micro-SD card inside the onboard's Ethernet Shield's card slot. In this way plant's user may access plant's living conditions data using any terminal or device connected to the Internet e.g. mobile or smartphone, tablet, laptop, smartwatch, personal computer, etc.

A DHT 11 sensor [13] and a soil moisture sensor [14] was used to read temperature, humidity and moisture respectively. DHT-11 is a basic low-cost temperature & humidity sensor features a temperature & humidity sensor complex with a calibrated digital signal output [15]. Inside, it integrates a humidity sensor and a thermistor (a variable resistance whose value varies with temperature). Its four-pin connections are really simple: the two pins (positive pin and ground) are for the 3-5V power supply, and the other one is for data output. The fourth pin is not connected.

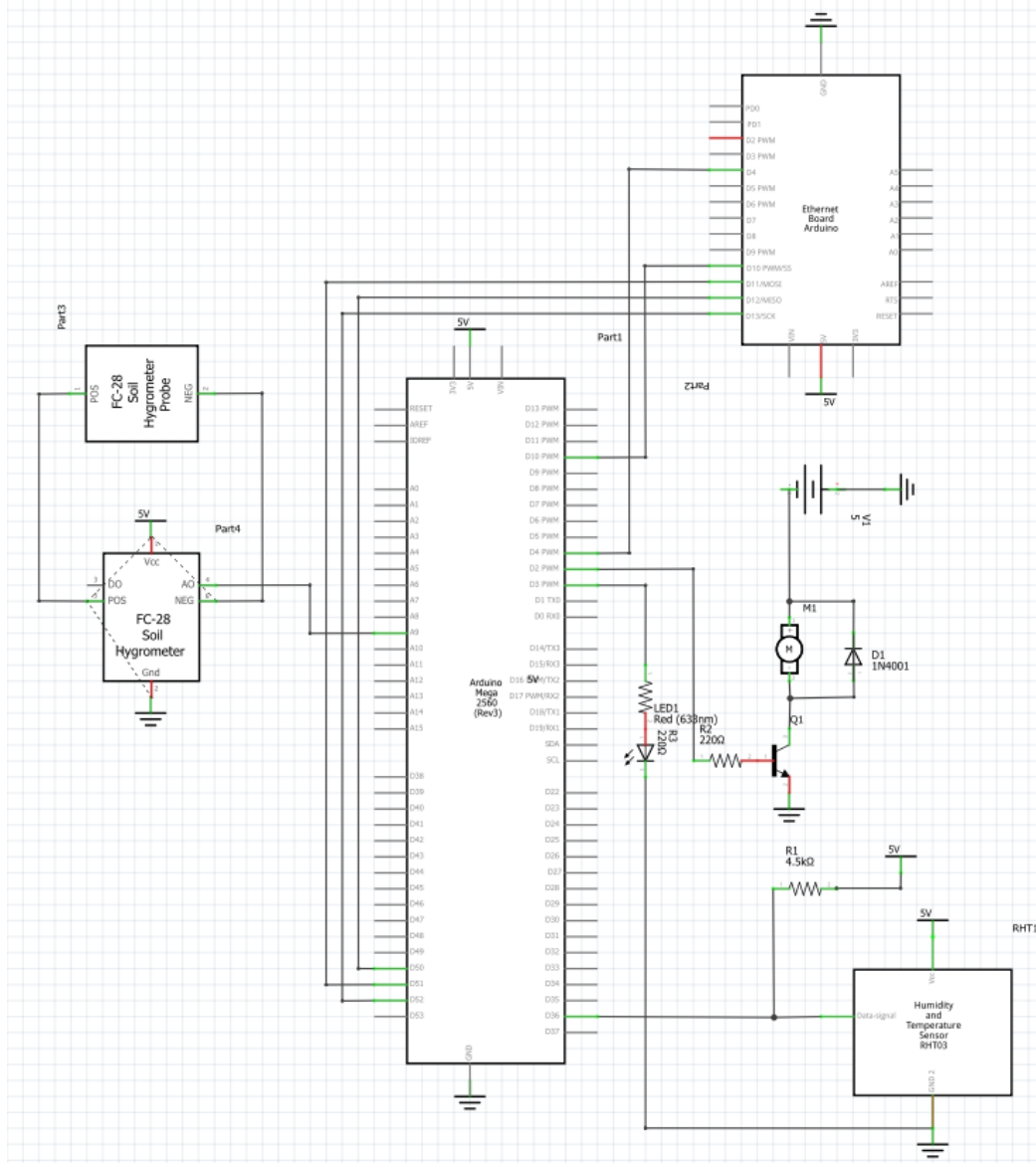
DHT-11's technical specifications:

1. Humidity: 20-90%, Accuracy 2-5%
2. Temperature: 0 to 50 ° C, accuracy  $\pm 0.5$  ° C
3. Power supply: 3-5.5V DC
4. Maximum current: 2.5mA

When power is supplied to the sensor, it will need one second in order to pass the unstable status. One capacitor valued 100nF was added between VDD and GND for power filtering. The communication process based on a serial Interface (single-wire two-way). Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms. Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first. Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal Tdata + 8bit check sum. If the data transmission is right, the check-sum should be

the last 8bit of "8bit integral RH data + 8bit decimal RH data

+ 8bit integral T data + 8bit decimal T data" [15].



**Fig. 2: Automated plant pot's electronic circuit**

The soil moisture sensor uses capacitance to measure dielectric permittivity of the surrounding medium. In soil, dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil. The sensor averages the water content over the entire length of the sensor. There is a 2 cm zone of influence with respect to the flat surface of the sensor, but it has little or no sensitivity at the extreme edges. This sensor was used to measure the loss of moisture over time due to evaporation and plant uptake [16]. Its three-pin connections are really simple: the two pins (positive pin and ground) are for the 5V power supply, and the other one is for analog data output.

Soil moisture sensor's technical specifications:

1. Working Voltage: 5V
2. Working Current: <20mA
3. Interface type: Analog
4. Working Temperature: 10°C~30°C

The plant pot's water pump is based on a DC motor working at 5V DC. It is connected to the Arduino MCU via a current amplifier circuit implemented by a 2N2222 NPN transistor (Q1) and a 220Ω resistor (R2). Diode 1N40001 is used to protect the transistor from the induction current generated by the DC motor during abrupt power supply interruption [17]. A 220Ω resistor (R3) is used for plant's LED protection. The full electronic circuit is powered by a power supply 12V DC/1A. However, any power supply between 6-20V DC/1A may be used.

In the following there is the automated plant pot's materials list:

1. 3 level plant pot
2. 3 stage filtering system (wire mesh, fiberglass wool, carbon filter)
3. Small and medium size gravel
4. Soil for the plant

5. Arduino Mega 2560 Rev3
6. Arduino Ethernet Shield 2
7. Transistor 2N2222
8. Diode 1N4001
9. 32 GB micro SD Card
10. DHT 11 sensor
11. Soil moisture sensor
12. LED for plant growth
13. Resistors: 1x10K $\Omega$ , 2x220 $\Omega$
14. Capacitor 100nF
15. Water pump 5-6V
16. Power supply 220-240V AC 50Hz / 12V DC, 1A
17. Other hardware: wires, water tank, water pipe, bolts, sealing gaskets, etc.

#### 4. SOFTWARE

The whole software of the automated plant pot includes:

1. Arduino's main software which reads and stores (SD card) the sensors data, and drives the water pump and lighting LEDs.
2. The web-server software implemented in C/C++ and transforms Arduino MCU to a web server [18], [19].
3. The webpage software (stored on the SD card), implemented in HTML, CSS, JavaScript code, and is responsible for webpage's UI, sensors' data acquisition, and user's sending commands to the automated plant pot. Special care was paid to making webpage's UI as simple, clean and functional as possible (Fig. 3).

In the following figure (Fig. 3) the flow chart of the automated pot's software is presented.

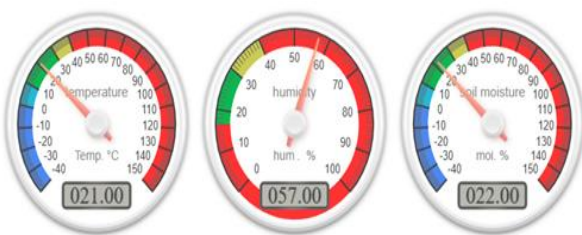


Fig. 3: Webpage's User Interface (UI)

In the beginning, the program initializes and defines SD, SPI, DHT, Adafruit, Ethernet libraries, IP and Mac Address of the Ethernet Shield.

Then, it checks for the existence of the SD card. If an SD card is found the program continues, otherwise it infinitely searches for it. The program requires the existence of the SD card to store sensors data and to read webpage's code. Once recognized, it proceeds to initialize it.

Subsequently, it reads the Ethernet shield and searches for the user's connected device, e.g. tablet, smartphone, PC (user's device is the web-client from the server-client model being implemented). If a connected device is found, then a client-server connection is implemented and the web server

(Arduino) sends the appropriate website (HTML, CSS, JS, code) to user's device browser in order to decode it and display the webpage and the UI of the automated pot.

Sensors data are sending through a properly formatted XML file. Then, the web server is reading if there are any received data (user commands) from the user's device browser. If the user press webpage's Plant's Lighting button, then automated pot's lighting LED is getting ON. If the user press webpage's Plant's Watering button, then pot's water pump is getting ON and the plant is watering.

If the user doesn't press any button, then the program continues and reads soil's moisture sensor's data, and it starts the watering process if this data reaches the user's desired minimum.

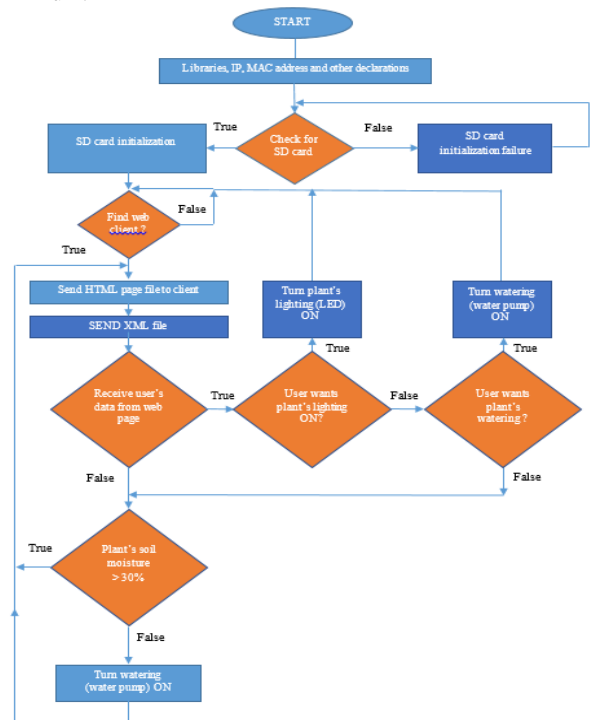


Fig. 4: Software's flow chart of the automated plant pot

The automated pot's software make use of the following open source libraries [10]: SD library [11], Serial Peripheral Interface (SPI) library [20], Ethernet library [21], SD library [11], Adafruit\_Sensor library and DHT.h library.

#### 5. CONCLUSION

The advantages of using the automatic pot are obvious to both user and plant. By remotely monitoring plant's living condition the user doesn't need to be present at the site and take care of it. With regard to the plant, better living conditions are created for its development, because of the automated process of plant's watering and lighting.

Some future improvements include:

1. A camera addition on the pot's top so visually inspect plant's health as to discern symptoms of any illness that occurs on the trunk or leaves of the plant.
2. A remotely (via the Internet) sprinkle pesticide mechanism installation, controlled by the user when identifies plant's disease.
3. A wireless interface to achieve Wi-Fi connectivity.

4. An automated mechanism to recall plant's living condition data from the Internet so to bypass user settings or even co-count them [22].
5. A Bluetooth integration and the development of a specialized Android app [2], [23].
6. A more cost effective implementation based on the cheaper Arduino Uno or Nano [24].

## 6. ACKNOWLEDGMENTS

All authors would like to thank the University of West Attica and specifically the Post Graduate Program of Studies (MSc) "New Technologies in Shipping and Transport", for the financial support provided to them to undertake this research project.

## 7. REFERENCES

- [1] M. Papoutsidakis, A. Chatzopoulos, K. Kalovrektis, and C. Drosos, "A Brief Guide for the Continuously Evolving  $\mu$  Controller Raspberry PI Mod . B," *Int. J. Comput. Appl.*, vol. 176, no. 8, pp. 30–33, 2017.
- [2] G. Vordos, A. Chatzopoulos, M. Papoutsidakis, and D. Piromalis, "Balance Control of a Small Scale Sphere with an Innovative Android Application," *J. Multidiscip. Eng. Sci. Technol.*, vol. 5, no. 10, pp. 8957–8963, 2018.
- [3] N.-L. Ada et al., "Light-scattering shade net increases branching and flowering in ornamental pot plants," *J. Hortic. Sci. Biotechnol.*, vol. 83, no. 1, pp. 9–14, Jan. 2008.
- [4] D. Kanjilal, D. Singh, R. Reddy, and J. Mathew, "Smart Farm: Extending Automation To The Farm Level," *Int. J. Sci. Technol. Res.*, vol. 3, no. 7, pp. 109–113, 2014.
- [5] M. Papoutsidakis, A. Chatzopoulos, D. Piromalis, and E. Symeonaki, "A Summary of Future Trends in Automation Industrial Processes," *Int. J. Eng. Sci. Invent.*, vol. 7, no. 10, pp. 48–55, 2018.
- [6] M. R. Sabzalian et al., "High performance of vegetables, flowers, and medicinal plants in a red-blue LED incubator for indoor plant production," *Agron. Sustain. Dev.*, vol. 34, no. 4, pp. 879–886, 2014.
- [7] M. Papoutsidakis, A. Chatzopoulos, C. Drosos, and K. Kalovrektis, "An Arduino Family Controller and its Interactions via an Intelligent Interface," *Int. J. Comput. Appl.*, vol. 179, no. 30, pp. 5–8, 2018.
- [8] Arduino, "Arduino Ethernet Shield," *Arduinio Comunicación*, 2018. [Online]. Available: <https://www.arduino.cc/en/Main/ArduinoEthernetShieldV1>. [Accessed: 01-Mar-2019].
- [9] A. Xatzopoulos, M. Papoutsidakis, and G. Chamilothis, "Mobile Robotic Platforms as Educational Tools in Mechatronics Engineering," in *International Scientific Conference eRA – 8*, 2013, pp. 41–51.
- [10] Arduino, "Arduino Libraries," 2015. [Online]. Available: <https://www.arduino.cc/en/reference/libraries>. [Accessed: 01-Mar-2019].
- [11] Arduino, "Arduino - SD," 2016. [Online]. Available: <https://www.arduino.cc/en/Reference/SD>. [Accessed: 01-Mar-2019].
- [12] S. Electronics, "Basic Arduino Web Server," 2013. [Online]. Available: <https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/basic-web-server/>. [Accessed: 01-Mar-2019].
- [13] G. Vasilakis, "How to use DHT-11 sensor," 2013. [Online]. Available: <http://www.ardumotive.com/how-to-use-dht-11-sensor-gr.html>. [Accessed: 01-Mar-2019].
- [14] CircuitsToday, "Arduino & Soil Moisture Sensor-Interfacing Tutorial," 2018. [Online]. Available: <http://www.circuitstoday.com/arduino-soil-moisture-sensor>. [Accessed: 01-Mar-2019].
- [15] M. Electronics, "DHT11 Humidity & Temperature Sensor Datasheet," 2018. [Online]. Available: <https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. [Accessed: 01-Apr-2019].
- [16] Eprolabs, "Moisture Sensor - ePro Labs WiKi," 2016. [Online]. Available: [https://wiki.eprolabs.com/index.php?title=Moisture\\_Sensor](https://wiki.eprolabs.com/index.php?title=Moisture_Sensor). [Accessed: 01-Mar-2019].
- [17] V. Mavrovounioti, A. Chatzopoulos, M. Papoutsidakis, and D. Piromalis, "Implementation of an 2-wheel Educational Platform for STEM Applications," *J. Multidiscip. Eng. Sci. Technol.*, vol. 5, no. 10, pp. 8944–8948, 2018.
- [18] M. Papoutsidakis, E. Symeonaki, A. Chatzopoulos, and D. Piromalis, "Iot Design To Support Wireless Sensor Networks And Data Transmission," *Int. J. Eng. Sci. Invent.*, vol. 7, no. 10, pp. 54–62, 2018.
- [19] M. Papoutsidakis, R. Tanwar, A. Chatzopoulos, and D. Tseles, "Custom Made Embedded Automation Systems For Smart Homes - Part 2 : The Implementation," *Int. J. Eng. Appl. Sci. Technol.*, vol. 2, no. 5, pp. 16–19, 2017.
- [20] Arduino, "Arduino - SPI," Arduino,[Online]. Available: <https://www.arduino.cc/en/reference/SPI>, 2015. [Online]. Available: <https://www.arduino.cc/en/Reference/SPI>. [Accessed: 01-Mar-2019].
- [21] Arduino, "Arduino Ethernet," 2015. [Online]. Available: <https://www.arduino.cc/en/Reference/Ethernet>. [Accessed: 01-Mar-2019].
- [22] M. Papoutsidakis, A. Chatzopoulos, C. Drosos, and D. Piromalis, "A System for Distance Control and GPS Mapping for Automotive Applications," *J. Multidiscip. Eng. Sci. Technol.*, vol. 5, no. 10, pp. 8949–8952, 2018.
- [23] S. Melkonian, A. Chatzopoulos, M. Papoutsidakis, and D. Piromalis, "Remote Control via Android for a Small Vehicle ' s 2-Wheels Balancing," *J. Multidiscip. Eng. Sci. Technol.*, vol. 5, no. 10, pp. 8964–8967, 2018.
- [24] M. Papoutsidakis, A. Chatzopoulos, D. Piromalis, and D. Tseles, "A 4-DOF Robotic Arm - Kinematics and Implementation as Case Study in Laboratory Environment," *Int. J. Comput. Appl.*, vol. 176, no. 8, pp. 34–38, 2017.