Utilization of New Technologies in the Production of Pharmaceutical Olive Oil

Ioannis Vlachos, Romanos Kalamatianos, Ioannis Karydis, Antonia Spiridonidou, and Markos Avlonitis

1 Introduction

Olive trees are one of the most dominant crops in the EU in terms of occupied area (Eurostat. Agri-Environmental Indicator—Cropping Patterns 2017), with over 1500 cultivars (Fogher et al. 2010) in the Mediterranean. Moreover, olive trees and their products have been proven to offer great health benefits, such as reducing the risk of heart diseases. They are also known for their anticancer properties (Sedef and Sibel 2014).

Olive trees are plagued by a variety of diseases such as the peacock spot which causes dark spotting on the leaves with a halo around each spot. Another olive tree disease caused by the fungus *Cercospora cladosporioides* results in yellowing and leaf drop. Both aforementioned diseases occur when temperature is approximately between 1.67 °C and 26.67 °C in combination with rainfall (Paul). Aside from diseases, olive trees are affected by the pest insect *Bactrocera oleae* also known as olive fruit fly, which infests the fruit of the tree. Population dynamics of the olive fruit fly are greatly affected by environmental factors, most notably temperature and humidity (Fletcher 1989; Genc and Nation 2008; Yokoyama et al. 2006; Broufas et al. 2009; Tsitsipis 1977, 1980; Pappas et al. 2011).

Since most of the diseases and pests that infest olive orchards are dependable on environmental factors, being able to monitor them and use the collected data as input to models and simulation algorithms could give us the opportunity to predict the outbreak of diseases and pests. Although environmental data for a specific location can be obtained via national or European institutions, collecting data on-site is of greater value. Through the use of ICT, the remote monitoring of olive groves could be achieved and thus have an overall overview of the olive trees' health.

I. Vlachos · R. Kalamatianos (✉) · I. Karydis · A. Spiridonidou · M. Avlonitis
Department of Informatics, Ionian University, Corfu, Greece
e-mail: gvlachos@ionio.gr; rkalam@ionio.gr; karydis@ionio.gr; avlon@ionio.gr

P. Vlamos (ed.), *GeNeDis 2018*, Advances in Experimental Medicine and Biology 1194, https://doi.org/10.1007/978-3-030-32622-7_22

A number of indoor and outdoor systems exist that are suitable for the purposes of collecting environmental data. Barrenetxea et al. (2008) proposed an outdoor wireless monitoring system for environmental monitoring. The aforementioned system uses solar panels, while MAC is used for data synchronization. Recently, Deshmukh and Shinde (2016) designed a low-cost WSN (wireless sensor network) monitoring system using open hardware with the major advantage that it has gateway node of WSN. Gaddam (2014) designed an outdoor wireless monitoring system for monitoring and predicting droughts in the early stages using open-source hardware.

Additionally, Mesas-Carrascosa and co-authors (2015) suggested a low-cost system for precision agriculture which is using a device that records data and a smartphone application that links said device to a data server for additional processing and analysis. In this paper we present a low-cost sensor system that could be installed on the field in order to collect environmental data. In the following section, we present the system and the individual parts it is comprised of.

2 Low-Cost Sensor System

We developed a low-cost system that logs environmental data such as temperature, humidity, and sunlight with a configurable sampling rate.

The system is comprised from the following parts:

- A low-cost microcontroller board
- A RTC (real-time clock) module with EEPROM (electrically erasable programmable read-only memory) microSD card slot adapter
- Low-cost sensors
- Power supply system
- Housing

2.1 Microcontroller Board

During the design of the system, we had to choose between an Arduino board, a Raspberry Pi board, and a BeagleBone board. Arduino was chosen because it is the cheapest and more robust for the application of data logging.

The Arduino hardware and software open-source platform gives us the opportunity to easily build the system we want for monitoring environmental data. The amount of extra shields (microSD card shield with RTC); breakout boards such as sensors, in combination with the IDE (integrated development environment); and the huge amount of libraries that exist for the Arduino platform were some of the reasons we chose it as the backbone of our system.

The Arduino Uno R3 (Fig. 1) microcontroller board is based on the atmel (ATmega328) microcontroller. The board has 6 analog input pins and 14 digital

Fig. 1 Microcontroller unit based on ATmega328 microcontroller – Arduino Uno R3

input/output pins. It is important to note that the 6 out of the 14 digital inputs/outputs can be used as PWM (pulse-width modulation) outputs. A 16Mhz crystal quartz is the responsible component for the stable clock of the microcontroller, and a power jack is available on board to supply the microcontroller with the appropriate power. In addition, an onboard USB female connector exists in order to connect the Arduino Uno board with any computer. Finally, an ICSP (In Circuit Serial Programming) header and a reset button exist on the Arduino Uno R3 board.

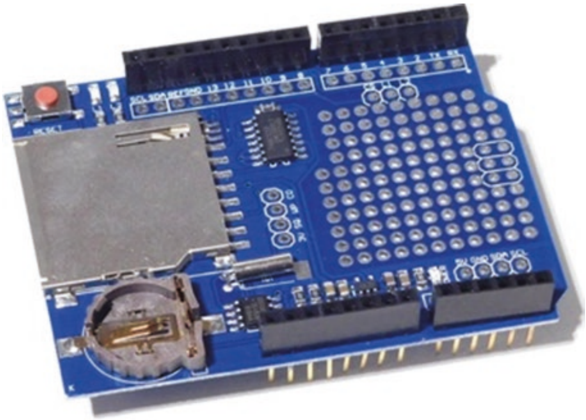
The programming of the microcontroller board ATmega328 can be done with the Arduino IDE interface by using the USB port of the microcontroller board. With the appropriate programming, we can easily program the Arduino Uno board to log values from its analog and digital inputs in a rate of 100 times per second (100 Hz).

2.2 RTC Module

To accomplish our objective, we need a real-time clock in order to add time stamps in our data. A possibility to store our data in a file on an SD card is also needed. Our microcontroller board Arduino Uno R3 supports breakout boards such as RTC and SD card and also shields for the same purpose. We decided to use the data logger shield (Fig. 2) because it can be fitted exactly to the Arduino Uno R3 microcontroller board without losing the flexibility and connectivity of the microcontroller board with the sensors. The data logger shield fits to the Arduino Uno R3 board one-to-one pin. The microcontroller cooperates easily and without any kind of compatibility problems with the data logger shield by using the appropriate library (that accompanies the shield) at the programming time.

The time stamp from the RTC has the following format yy/mm/dd – hh/mm/ss/ms. If the need arises, we can modify the way that the date is presented.

Fig. 2 Data logger shield for Arduino Uno R3 board



2.3 Card Slot Adapter

The SD card slot part of the data logger shield is also programmed from the microcontroller board and stores the data from the sensors. The SD card can work with FAT-16 and FAT-32 formats without problems.

The data logger shield is using the SPI (Serial Peripheral Interface) bus directly connected with the microcontroller, so the SPI library is attached at the programming sketch. Also, the SD and the RTClib libraries are included in the programming sketch for the cooperation of the shield with the microcontroller. Finally, the data logger shield is powered directly from the pins (Vcc,Gnd) of the microcontroller.

Generally the data logger shield is attached to the microcontroller board Arduino Uno R3 like this (MOSI, pin 11; MISO, pin 12; CLK, pin 13; CS, pin 10).

2.4 Low-Cost Sensors

Temperature/Humidity Sensor (Fig. 3) Using a low-cost sensor to measure temperature and humidity with the microcontroller board, Arduino Uno R3 is simple and easy. Using the digital sensor DHT33 (*RHT04*) and a three-pin (power ground, data) temperature/humidity sensor, we can measure indoor and outdoor temperature and humidity with the specifications as listed below:

- Operating range: humidity 0–100%RH, temperature − 40 to 100 °C
- Range of measuring humidity: 0–100%RH (0–50° temperature compensation)
- Range of measuring temperature: −40 to +100 °C
- Resolution or sensitivity: humidity, 0.1%RH; temperature, 0.1 °C
- Accuracy of measuring humidity: ±2.0%RH
- Accuracy of measuring temperature: ±0.5 °C
- Repeatability: humidity, + − 0.5%RH; temperature, + − 0.2 °C

Fig. 3 *Temperature/humidity sensor DHT33 – RHT04*

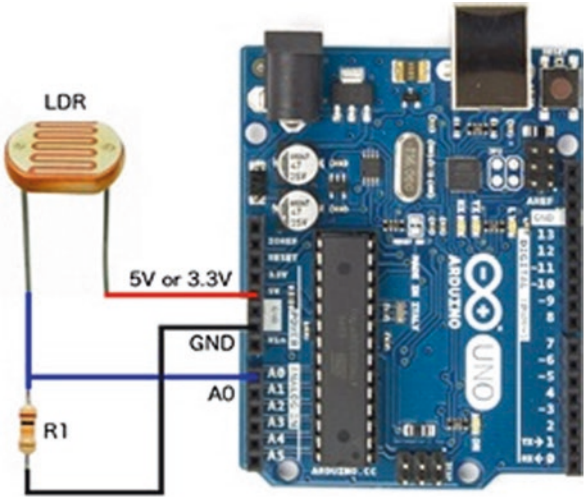


Fig. 4 Sunlight sensor – LDR

The sensor is powered directly from the microcontroller board due to its operational power supply range (3.3 V or 5 V) and its power dissipation which is very low.

The microcontroller communicates with the sensor with the digital pin 5, and it interconnects using the library wire. Additionally, the sensor DHT33 has its own library[1] support for the microcontroller board Arduino Uno R3.

Sunlight Sensor As a low-cost sensor, we use an LDR (light-dependent resistor). The LDR (Fig. 4) change its resistance according to the light that hits its surface. The resistance at the pins increases in total darkness (normally takes values on the M-ohms range) and decreases when light covers its surface (it normally reaches values in the range of K-ohms). Using an external resistor (R1), connected with the LDR, we measure the voltage value at the connection point network (LDR-R1) at the analog input 0 of the microcontrollers. Actually, by connecting the free pin of the LDR to the Vcc 5volt power from the microcontroller pin +5v, we create a voltage divider consisted from two resistors (LDR + R1). By calculating the voltage at the input of the analog input pin 0 of the microcontroller and by extra calculations

[1] http://playground.arduino.cc/Main/DHTLib

made inside the microcontroller according to the LDR characteristic specifications, we can determine the sunlight measurement in (lux).

The external resistor R1 can vary in value depending on the LDR resistor in full light. In our system we use a 10kohm external resistor of ¼ watt power.

2.5 SD Card

All the data from the sensors are recorded into a CSV (comma-separated value) file at the predefined intervals along with a time stamp of when the recording occurred. The time of the data collection can be very easily modified according to the requirements.

The CSV file is stored in a class 4 SD card which is more than fast enough with a capacity of 4GB. It can be used to store data for years without the need to be replaced.

2.6 Power Supply

The Arduino Uno R3 microcontroller board can be powered through the following four different ways, from which we chose the first one:

- Through a power jack (7–12 volt dc)
- Through a USB port (5 volt)
- Through a Vin pin (6–12 volt dc)
- Through a + 5v pin (5 volts only)

We use the 18,650 LiPo batteries (Fig. 5b) that give 4.2 volts when they are fully charged and 3.7 volt when they are almost empty. By serially connecting two 18,650 batteries, we have an amount of 8.4 volts, and we are on the range or operation voltage while we "feed" the microcontroller system from the power jack. Each battery

Fig. 5 The power supply of the system: (**a**) the battery pack, (**b**) 18,650 batteries, and (**c**) solar panel

Fig. 6 IP66 housing box

has a capacity of 3400mah, so when we serially connect the two batteries, we get 8.4 volt and a capacity of 3400mah. In case we want to double the capacity, we place another pair of serially connected batteries, wired in parallel with the other two, which will give us a pack of four 18,650 batteries with a voltage of 8.4 volt and a capacity of 6800mah. For a 15-minute recording interval, the system can operate for 8 days without changing the batteries.

The four batteries are fitted in a battery pack (Fig. 5a). The two pairs of serially connected batteries are connected in parallel with each other in order to create a battery bank with 8.4 volt and 6400mah capacity. A male power jack connects the power bank with the Arduino Uno R3 microcontroller data logger system. An additional fuse of (100ma) is used to protect the system from power malfunctions.

The system can also be modified by adding a small solar panel (Fig. 5c) to charge the batteries and make the system autonomous in locations that are not easily accessible.

2.7 Housing

All the components that comprise our low-cost data logging system are hosted in an IP66 box (Fig. 6) and can be easily transported due to its lightweight. With the hosted box, the system is robust against heavy weather conditions like rain.

3 Conclusions

In this article, we present our low-cost sensor system which is based on the Arduino microcontroller board that aims to collect environmental data. The main objective was to be able to monitor the climate factors inside olive groves; however it can be used for other applications as well. Due to the low manufacturing cost, we will be able to install a great number of sensor systems inside olive groves. In return, we will be able to collect a large amount of data and use them to identify, among others, the microclimate of an olive grove and feed the data to computational models in order to predict outbreaks of olive fruit fly and olive tree-related diseases.

Future work includes extending the life expectancy of the batteries by decreasing the power consumption between two data logs. Furthermore, the system could be made autonomous by integrating a solar panel to recharge the batteries for a period of 6 months. Finally, we will work toward an upgrade of the whole system by sending the recorded data to a remote server and by equipping the board with more sensors (soil moisture sensor, rain detection sensor, barometric pressure sensor, etc.).

Acknowledgments The financial support of the European Union and Greece (Partnership Agreement for the Development Framework 2014–2020) under the Regional Operational Programme Ionian Islands 2014–2020 for the project "Olive Observer" is gratefully acknowledged.

References

Barrenetxea G, Ingelrest F, Schaefer G, Vetterli M, Couach O, Parlange M (2008) SensorScope: out-of-the-box environmental monitoring. In: Proceeding of ACM/IEEE IPSN, St. Louis, MO, USA, p 332–343

Broufas GD, Pappas ML, Koveos DS (2009) Effect of relative humidity on longevity, ovarian maturation, and egg production in the olive fruit Fly (Diptera: Tephritidae). Ann Entomol Soc Am 102(1):70–75. https://doi.org/10.1603/008.102.0107

Deshmukh AD, Shinde UB (2016) A low cost environment monitoring system using raspberry Pi and arduino with Zigbee. In: Inventive Computation Technologies (ICICT), international conference, Coimbatore, India on. vol 3, IEEE

Eurostat. Agri-Environmental Indicator—Cropping Patterns (2017) Available online: https://ec.europa.eu/eurostat/statistics-explained/index.php/Agri-environmental_indicator_-_cropping_patterns. Accessed 25 Sept 2018

Fletcher BS (1989) Temperature–development rate relationships of the immature stages and adults of tephritid fruit flies, World Crop Pests: fruit flies—their biology, natural enemies and control, ed. AS Robinson, G Hooper, Amsterdam, Elsevier 3A:273–289

Fogher C, Busconi M, Sebastiani L, Bracci T (2010) Chapter 2—Olive genomics. In: Preedy VR, Watson RR (eds) Olives and olive oil in health and disease prevention. Academic Press, Cambridge, pp 17–24

Gaddam A (2014) Designing a wireless sensors network for monitoring and predicting droughts. ICST 20140: 8th International Conference on Sensing Technology, Liverpool, UK, 2-4 September 2014

Genc H, Nation JL (2008) Survival and development of Bactrocera oleae Gmelin (Diptera: Tephritidae) immature stages at four temperatures in the laboratory. Afr J Biotechnol 7:2495–2500

Mesas-Carrascosa FJ, Santano DV, Meroño JE, de la Orden MS, García-Ferrer A (2015) Open source hardware to monitor environmental parameters in precision agriculture. Biosyst Eng 137:73–83

Pappas ML, Broufas GD, Koufali N, Pieri P, Koveos DS (2011) Effect of heat stress on survival and reproduction of the olive fruit fly Bactrocera (Dacus) oleae. J Appl Entomol 135:359–366

Paul M. Vossen: peacock spot and cercospora foliar disease on olive. http://cesonoma.ucanr.edu/files/27173.pdf
Sedef NE, Sibel K (2014) Olive tree (Olea europaea) leaves: potential beneficial effects on human health. Nutr Rev 67(11):632–638. https://doi.org/10.1111/j.1753-4887.2009.00248.x
Tsitsipis JA (1977) Effect of constant temperatures on eggs of olive fruit fly, Dacus oleae (Diptera: Tephritidae). Ann Zool Ecol Anim 9:133–139
Tsitsipis JA (1980) Effect of constant temperatures on larval and pupal development of olive fruit flies reared on artificial diet. Environ Entomol 9:764–768
Yokoyama VY, Rendon RA, Sivinski J (2006) Biological control of olive fruit fly (Diptera: Tephritidae) by Reseases of Psyttalia cf. concolor (Hymenoptera: Braconidae) in California, Parasitoid longevity in presence of the host, and host status of walnut husk fly. In: Proceedings of the 7th international symposium on fruit flies of economic importance, Salvador, p 157–164