

# A Smart System for Garden Watering using Wireless Sensor Networks\*

Constantinos Marios  
Angelopoulos  
Research Academic Computer  
Technology Institute (CTI) and  
University of Patras  
aggeloko@ceid.upatras.gr

Sotiris Nikolettseas  
Research Academic Computer  
Technology Institute (CTI) and  
University of Patras  
nikole@cti.gr

Georgios Constantinos  
Theofanopoulos  
University of Patras  
theofan@ceid.upatras.gr

## ABSTRACT

As water supplies become scarce and polluted, there is an urgent need to irrigate more efficiently in order to optimize water use. In this paper, we present a WSN based, smart home-irrigation system that consists of heterogeneous motes, special sensors and actuators. The system is fully adaptive not only to environmental conditions but also to the specific water needs that different plants may have. This way, it manages to perform efficient home irrigation, while it provides an IPv6-capable managing system.

## Categories and Subject Descriptors

C.2.4 [Distributed Systems]: Distributed Applications

## General Terms

Experimentation, Design

## Keywords

Wireless Sensor Networks, irrigation control, smart watering

## 1. INTRODUCTION

Wireless Sensor Networks consist a crucial part of the Future Internet. Thus, they will play an important role in our everyday life in years to come. The applications of WSNs range from distributed monitoring systems to smart embedded managing systems. As water supplies become scarce and polluted, there is a dire need to irrigate more efficiently in order to optimize water use. Recent advances in soil water monitoring combined with the growing popularity of Wireless Sensor Networks make the commercial use of such systems applicable not only to agriculture and industry [5, 6], but to homes as well.

\*This work was partially supported by EU/FIRE HOBNET project - STREP ICT-257466

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*MobiWac'11*, October 31–November 4, 2011, Miami, Florida, USA.  
Copyright 2011 ACM 978-1-4503-0901-1/11/10 ...\$10.00.

To date, typical irrigation automations include electromechanical programmers that can control the watering procedure. These systems are programmed to irrigate at regular time intervals for predefined periods of time; e.g. once a day for half an hour. The programming of these automated systems is heuristically based on experience and is poorly adaptable to changes in weather conditions, as well as the existence of different water needs by different kind of plants. As a result, water resources are poorly used, plantation is over- or under-irrigated and increased costs of garden maintenance are introduced.

Furthermore, the use of Wireless Sensor Networks gives watering systems monitoring as well as remote management capabilities. With sensor motes being IPv6 capable, they can be represented as resources in a RESTful architecture, thus allowing remote access and control to the system (e.g. via Android devices).

## 2. SYSTEM DESCRIPTION

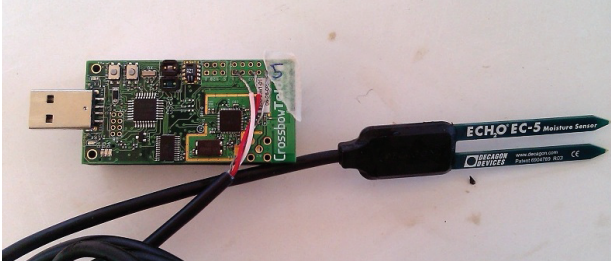
Our system architecture includes sensor motes, soil humidity sensors, mote driven electro-valves that control the water flow towards the plants, and a Java application running on a PC, that collects data from the sensor network and stores them in a MySQL database. For the purposes of this study, we abstract a regular home garden with three pots containing different plants with highly diverse watering needs; a geranium that has very limited watering needs (once a week), a lavender that under normal weather conditions has medium watering needs (three times a week) and a mint that requires regular watering (in high temperatures during summertime, even twice a day).

The soil humidity of each pot is monitored by a mote equipped with soil humidity sensor. The watering of each pot is controlled by a corresponding mote-driven electro-valve independently from the other pots. When the soil of the pot is too dry, then the corresponding mote, monitoring soil humidity, informs the mote that drives the corresponding electro-valve to start watering the pot. When the soil humidity returns to normal levels, the soil monitoring mote signals the electro-valve mote to cut off water supply for that pot.

Throughout the operation of the system, the levels of the soil humidity of each pot are forwarded to the Sink by the corresponding motes. The Sink is a mote connected to a PC on a USB port that acts as a gateway for the rest of the motes. When it receives a soil humidity measurement, it forwards it to the PC where a Java application receives data

and stores them in a MySQL database for post-processing. Apart from soil humidity, measurements concerning temperature are also forwarded to the Sink and logged at the database.

## 2.1 Hardware Description



**Figure 1:** The EC-5 soil sensor shouldered on a TelosB mote

### 2.1.1 Sensor Motes

For our implementation we used two mote platforms, TelosB [3] and IRIS motes [2]. Both of them are ZigBee compliant, small, light weight and when using energy saving protocols can be powered with two AA batteries for several weeks, even months. These characteristics makes them ideal for our smart garden watering system as they can easily be deployed everywhere while being independent of power installations. Furthermore, IRIS was combined with the MDA100CB sensor and data acquisition board which has a precision thermistor, a light sensor/photocell and general prototyping area. This prototyping area was used in order to connect a relay, through which IRIS motes were able to control the electro-valves.

### 2.1.2 Soil Sensors

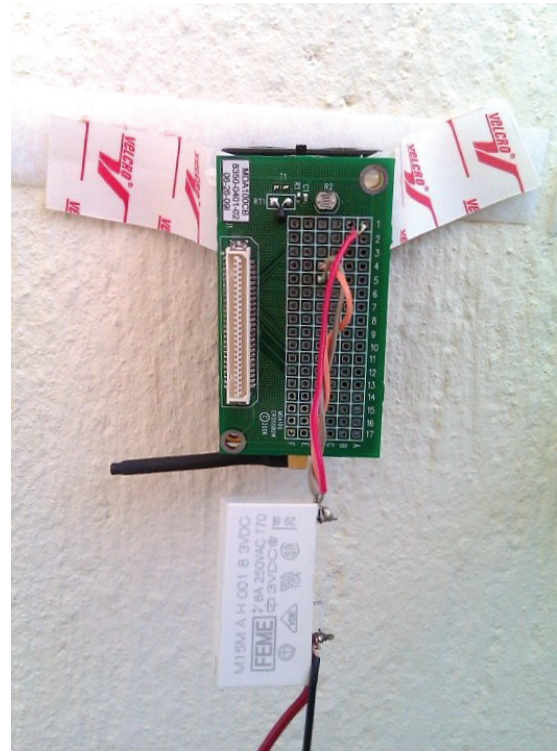
The EC-5 soil humidity sensor by Decagon [1] was used for soil monitoring. It consists of a cable, which on one end has two prongs and on the other end has 3 wires. The prongs are pushed inside the potting soil and the three wires of the other end are connected to the 10-pin expansion connector of TelosB motes. The bare wire is connected to the ground pin, the red one is connected to the ADC channel pin (programmed as input) and the white wire is connected to the VCC pin. This sensor had to be used along with TelosB motes as it provides 12-bit data, while the IRIS mote has a 10-bit ADC.

For successful communication between TelosB and the soil humidity sensors for collecting soil humidity data, we used the component `Msp430Adc12ClientC` and the corresponding interface. In the source code we set the ADC1 (i.e. pin 5 of the 10-pin TelosB expansion connector) in which we had previously shouldered the red wire (analogue out) of the soil humidity sensor. We, also, used the component `Sensirion SHT11` of TelosB to monitor the air temperature.

### 2.1.3 Electro-valves

In order to control the irrigation process we use solenoid valves provided by Irritol. A solenoid valve is an electromechanical valve that is controlled by an electric current. The electric current runs through a solenoid, which is a wired

coil wrapped around a metallic core. The solenoid creates a controlled magnetic field when electrical current is passed through it. This magnetic field affects the state of the solenoid valve, causing the valve to open or close. The electric valves operate with a 9V-32V battery.



**Figure 2:** A relay shouldered on the MDA100CB data acquisition board of an IRIS mote

In order for the IRIS motes to be able to drive the valves we used relays that take as input a 3 VDC current and can control circuits of up to 250 VAC. Due to the low current IRIS motes can provide as output, the relay is driven by using two digital outputs of the prototyping area of the MDA100CB board. Then, the relay is connected in series with the electro-valve and an external 9V battery. This way, when the IRIS mote triggers the relay, the circuit is closes, the electro-valve opens and the irrigation process begins.

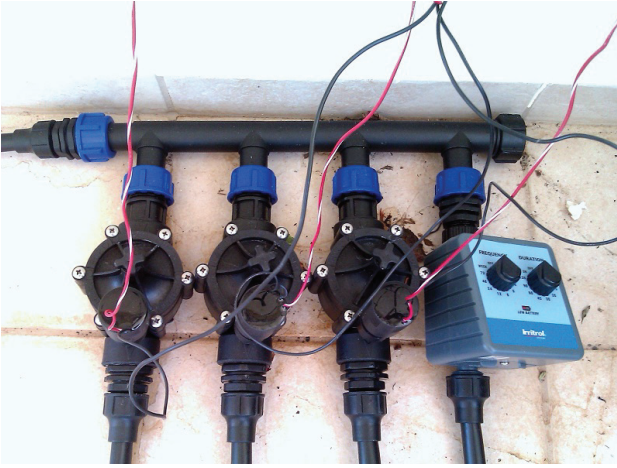
Finally, in order to have a base of reference, we also installed a common irrigation programmer. This programmer can be set to irrigate at regular time intervals for fixed periods of time; i.e. every second day for half an hour.

## 2.2 Software Description

### 2.2.1 Implementation in TinyOS

The sensor motes were programmed in TinyOS [4]. TinyOS has a component based architecture and forms an event-driven operating system. The motes were assigned unique mote IDs and were programmed so that each TelosB informs about soil humidity the IRIS controlling the corresponding to the pot electro-valve. In order to achieve TelosB - IRIS communication we had to configure them to use the same ZigBee channel. We have chosen to use channel 26 in order to have minimal radio interferences from other wireless networks.





**Figure 3:** The installed electro-valves and the common irrigation programmer. The red and black cables of the solenoids are connected in series with the relay and the external power source (batteries or AC-DC converter)

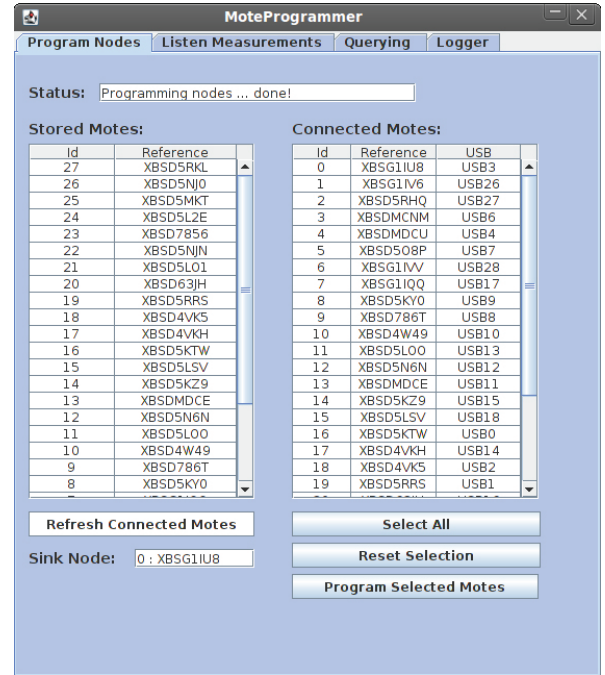
Our system comprises of three TinyOS applications. The *SoilSensorApp* (runs on TelosB motes) uses the *humidity\_msg* and the *valve\_msg* message types defined in the *SoilSensor.h* header. The first type is used in order to send humidity values to the Sink to be logged at the MySQL database. The second type is used in the TelosB - IRIS communication. If the value of soil humidity gets below a lower bound, defined by variable *low\_lim*, or above an upper bound, defined by variable *upper\_lim*, then a *valve\_msg* message with an appropriate value is sent to the IRIS mote controlling the corresponding electro-valve to set its state to be on or off.

The second TinyOS application is the *ValveApp* (runs on IRIS motes). With this application running, if the mote receives a *valve\_msg* message, it reads its payload and accordingly sets or clears the output ADC pins. This way it can trigger the relay connected to these pins on/off; thus driving the electro-valve.



**Figure 4:** The entire smart gardening system deployed

The sink node runs a modified version of the Basestation-App that comes along with the 2.1.0 TinyOS distribution. This application receives the radio messages sent to the sink by the portable nodes and forwards them to the PC using UART. Using MIG we construct a Java class that allows the Java application running on the PC to receive and handle these data. The Java application is called *Moteprogrammer* and is developed by our research team. It automatically detects any motes that are connected to the desktop PC via USB ports and allows the administrator of the network to interact with the motes e.g. to flush (i.e program) the motes with the nesC application binary code, to reset the motes and communicate with them in order to change the event generation rate, as well as access data stored in the MySQL database.

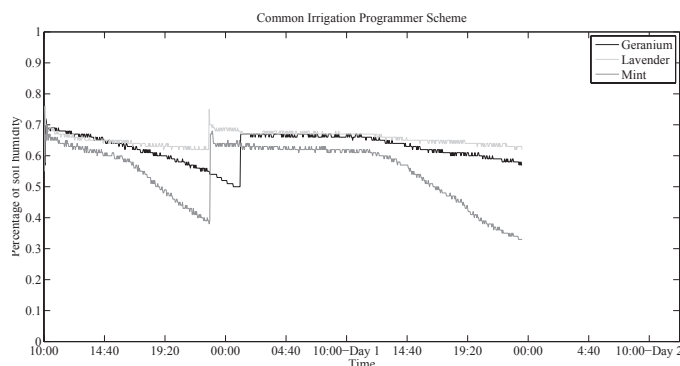


**Figure 5:** Screenshot of the MoteProgrammer application

### 3. PERFORMANCE EVALUATION

We evaluate the performance of the traditional irrigation scheme, using a common irrigation programmer, and the smart irrigation scheme, using a wireless sensor network. We let each system to water for two days three pots; each one containing a different plant. The plants were chosen in order to have diversity in their water needs. Sorted by descending water needs, each pot contains a geranium, a lavender and a mint. The season the experiments were conducted was summertime and the temperature was around 36°C during the day and 30°C during the night.

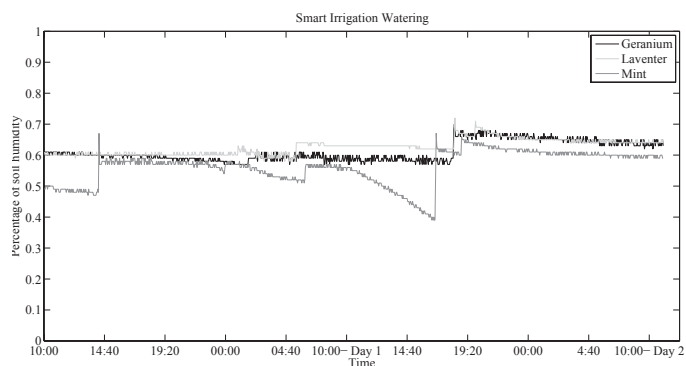
Figures 6 and 7 show the soil humidity of each pot over time for the two irrigation schemes respectively. We note that the pot containing the mint, that has the highest water needs from all three plants we used, depletes humidity from its containing soil much faster. On the contrary, the geranium absorbs humidity much slower, therefore the soil dries out at a lower rate.



**Figure 6: Soil humidity over time when using common irrigation programmer**

When using a common irrigation programmer there exist great variations in the concentration of soil humidity. The soil tends to dry out and then is flooded with water causing an almost vertical increase of the soil humidity values. These extreme variations are not to the benefit of the plants, as they require soil humidity to remain at a given level. Furthermore, dried out soil causes great amount of water to pour away as it cannot withhold water to the same degree as even lightly moist soil.

On the other hand, the smart home-irrigation system manages to maintain soil humidity at the same level. It dissipates less water and it provides an irrigation scheme that is adaptable to the specific watering needs of each plant. The most important feature is the fact that by constantly monitoring the humidity levels, it basically adapts to current environmental conditions. Whether there are high temperature or sunlight variations or not, the system will adjust the irrigation process so as to maintain the same level of soil humidity.



**Figure 7: Soil humidity over time when using smart home irrigation system with WSN**

## 4. CONCLUSION AND FUTURE WORK

In this paper we presented the architecture and the implementation of a smart home irrigation system. The system consists of two types of sensors motes (TelosB and IRIS), special soil humidity sensors, electro-valves that are mote-driven with the use of relays and a Java application that is used for data collection. Performance evaluation showed that our system manages to maintain soil humidity levels regardless of external factors (i.e. variations at temperature and sunlight). It also proved that the system is aware of the different watering needs each

In future work, we plan to use solar panels along with rechargeable batteries in order to make our system self sustainable in terms of energy consumption. We also plan to incorporate to our system the ability to be managed remotely. This will be done by representing sensor motes as resources in a RESTful architecture, thus allowing to access and control the system with the use of web-services (e.g. via Android smart-phones).

## 5. REFERENCES

- [1] Decagon ec-5:  
<http://www.decagon.com/products/sensors/soil-moisture-sensors/ec-5-soil-moisture-small-area-of-influence/>.
- [2] Iris datasheet,  
<http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html?download=135>
- [3] Telosb datasheet:  
[www2.ece.ohio-state.edu/~bibiye/ee582/telosmote.pdf](http://www2.ece.ohio-state.edu/~bibiye/ee582/telosmote.pdf).
- [4] Tinyos 2.1.0 documentation.
- [5] Balendonck and al. Sensors and wireless sensor networks for irrigation management under deficit conditions (flow-aid). In *International Conference on Agricultural Engineering (AgEng2008)*, 2008.
- [6] M. Dursun and S. Ozden. A wireless application of drip irrigation automation supported by soil moisture sensors. *Scientific Research and Essays*, Vol. 6(7):1573 – 1582, 2011.