An Automated Irrigation System Using Arduino Microcontroller

Aslinda Hassan[1*], Siah Bing Sheng[1], Wahidah Md Shah[1], Nazrulazhar Bahaman[1]

[1]Centre for Advanced Computing (C-ACT), Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

[*]Corresponding e-mail: aslindahassan@utem.edu.my

Keywords: Arduino; soil moisture sensor, automated irrigation system

ABSTRACT – This paper proposes an automated irrigation using Arduino microcontroller system which is cost effective and can be used farm field or average home garden. The proposed system can automatically water the plants when the soil moisture sensor detects the soil is insufficient of water by using the Arduino microcontroller as the center core. When the soil moisture sensor sense the dry soil, it will show the moisture percentage on the LCD display, and the relay module will switch on the water pump automatically to start the watering process, or vice versa. Hardware testing is conducted to ensure the proposed system is fully functional.

1. INTRODUCTION

Irrigation is an artificial application of watering the land for agricultural production. The requirement of water to the soil depends on soil properties such as soil moisture and soil temperature. Effective irrigation can influence the entire growth process and automation in irrigation system using modern technology can be used to provide better irrigation management. In general, most of the irrigation systems are manually operated. These traditional techniques can be replaced with automated techniques of irrigation in order to use the water efficiently and effectively. Conventionally, farmers will present in their fields to do irrigation process. Nevertheless, nowadays farmers need to manage their agricultural activity along with other occupations. A sensor based automated irrigation system provides promising solution to farmers where the presence of a farmer in field is not compulsory during irrigation process.

Arduino is a flexible programmable hardware platform and designed to control the circuit logically. Central to the Arduino interface board is the main component of an integrated circuit chip that can be programmed using C++ language. This microcontroller is an AVR type, which produced by Atmel firm. The device can read the input, process the program, and produce many outputs based on project requirements.

This paper presents the implementation of an automated irrigation system using an Arduino microcontroller. In this system, a soil moisture sensor is used to detect and check the soil humidity of the plant. Based on the soil moisture level from the soil, the system will let the water pump to automatic water the plant when it is too dry and turn off the water pump when the soil of the plant is wet.

At the previous works, considering to the automated watering techniques, it can be found that the Arduino based sensors have been utilized for the plant watering system [1] and automated irrigation systems [2]–[4]. An Arduino Based Automatic Plant Watering System is proposed in [1] where the authors developed the Arduino microcontroller used to control two functional components which are the moisture sensors and the motor/water pump to automatically water the plant. The moisture sensor's function is to sense the level of moisture in the soil whereas the water pump supplies water to the plants.

In [2], a smart drip irrigation system using Raspberry Pi and Arduino is proposed for home automation system. The drip irrigation system makes the efficient use of water where the water is slowly dripped to the roots of the plants through narrow tubes and valves. The water flow from the system can be remotely controlled via email.

2. METHODOLOGY

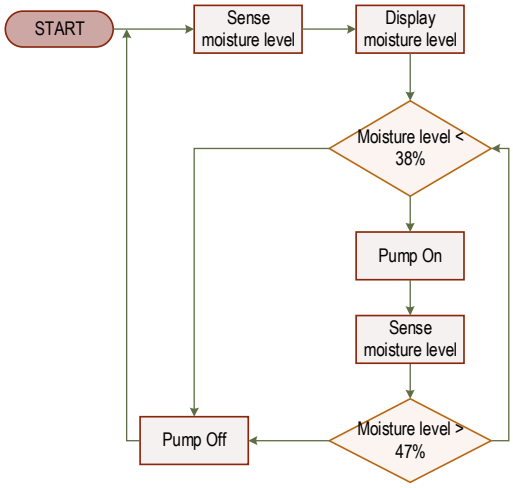

Figure 1 The project's flowchart

Figure 1 shows the flow chart of the automated irrigation system. From the flow chart, it is shown that when the moisture sensor detects the soil moisture level to be under 38%, the soil is considered dry and the water pump will be switched on. The water pump waters the soil until the level reaches 47% or more, then, it will automatically switch off the watering process. It has to be noticed that, when the sensor tests whether the moisture level in the soil is dropped to less than 47%, the system goes back to the first condition where it will sense the moisture level of the soil.

Figure 2 shows the complete hardware setup of the

proposed system which include the Arduino board and all the necessary attached hardware.

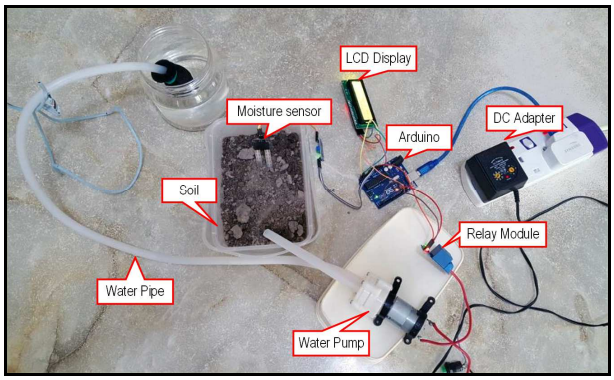

Figure 2 Hardware implementation

From Figure 2, it can be see that the Arduino is the center of this system which connects all the required hardware. The soil moisture sensor measures the level of moisture from the soil, and it is transfer the sensed values to the Arduino board to process and make decision.

The LCD display shows the value that the Arduino received from the moisture sensor. At the same time, the acquired data is sent to the relay module to determine whether to turn on or off the water pump. If the condition requires the water pump to be turned on, the water pipe attached to the pump will begin to draw water from the water source, and push the water to the other side of water pipe to complete the watering process for the soil.

3. RESULTS AND DISCUSSION

This section describes the test strategy that is used for this project. The test is used to determine whether the hardware and software will be test early to make sure that it is functioning according to the requirement. The main test activity is to test the sensor values and its functionality.

In the project, the soil moisture sensor acquires the voltage data and sends the data to the Arduino. The coding in the Arduino translates the analog reading to digital reading. The Arduino checks the upper and lower boundaries of the analog value before it is convert to digital value.

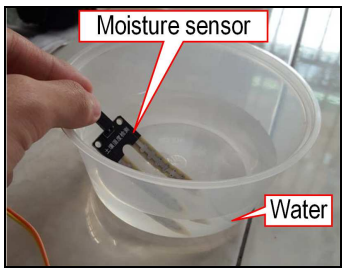

Figure 3 The functionality test for the moisture sensor

Figure 3 shows the functionality test for the moisture sensor. In this test, the soil moisture sensor captures high values for dry condition and low values for wet condition. The values of the upper boundary for dry condition are around 893 with 900 whereas the values of the lower boundary for wet condition are around 399 with 400. Figure 4 shows the moisture sensor functionality test for both dry and wet condition.

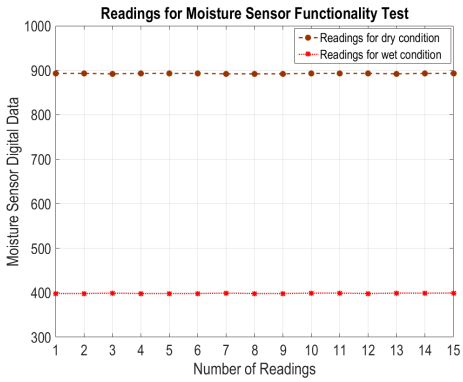

Figure 4 The result for the sensor functionality test.

4. CONCLUSIONS

The main purpose of this paper is to propose an automated irrigation system that water the plant without any human control. The automated irrigation system implemented is found to be feasible and cost effective for optimizing water resources for agricultural production. Aside from the automated irrigation system, the proposed system provides the monitoring function where users can check the soil moisture based on the reading on the LCD display. The proposed system has been designed and tested to function automatically. For future works, the automated irrigation system can be configured to measure the moisture level according to the moisture requirement of the different plants.

ACKNOWLEDGEMENT

Special thanks of gratitude to the Universiti Teknikal Malaysia Melaka for sponsoring the fund under the UTeM Short Term Grant with the reference number: PJP/2018/FTMK(1B)/S01628.

REFERENCES

[1] S. V Devika, S. Khamuruddeen, S. Khamurunnisa, J. Thota, and K. Shaik, "Arduino Based Automatic Plant Watering System," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 10, pp. 449–456, 2014.

[2] N. Agrawal and S. Singhal, "Smart drip irrigation system using raspberry pi and arduino," *Int. Conf. Comput. Commun. Autom.*, pp. 928–932, 2015.

[3] C. Kumar Sahu and P. Behera, "A low cost smart irrigation control system," in *2nd International Conference on Electronics and Communication Systems, ICECS 2015*, 2015, pp. 1146–1151.

[4] P. Singh and S. Saikia, "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module," in *IEEE Region 10 Humanitarian Technology Conference 2016, R10-HTC 2016 - Proceedings*, 2017