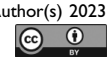
Check for updates

Original article

IJMEE

International Journal of Mechanical Engineering Education
2025, Vol. 53(2) 391–410
© The Author(s) 2023

Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/03064190231190052
journals.sagepub.com/home/ijj


Teaching embedded control system design of electromechanical devices using a lab-scale smart farming system

Himali Mistry[1], Dina Shona Laila[2,3], and Mathias Foo[1]

Abstract

This article presents the design and monitoring of a lab-scale smart farming system through the integration of control and app designs that can be used for teaching embedded control application to electromechanical systems. A combination of sensors and actuators is used to develop an Arduino-based embedded feedback control system that could be implemented in a smart farming environment. Specifically, we look at controlling electromechanical devices to actuate the fan and water pump to provide the optimal temperature and moisture, respectively, to enhance plant growth in a smart farming setting. The effectiveness of the feedback control is tested by conducting a plant growth experiment. Using garden cress (*Lepidium sativum*) as a case study, the plant grown in the controlled temperature and moisture settings shows substantially healthier growth compared to the one grown in the non-controlled environment. In addition, an app is designed and developed to transform the Arduino data stream from the sensors into valuable insights that could help the users to monitor and improve the overall crop health. The developed system in this paper enables students to learn integral skills from interdisciplinary engineering fields (e.g. systems, control, mechanical and computer) to solve an agricultural problem.

[1]School of Engineering, University of Warwick, Coventry, West Midlands, UK
[2]Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Lulea, Norrbottens, Sweden
[3]Department of Electrical and Electronic Engineering, Universiti Teknologi Brunei, Gadong, Brunei-Muara, Brunei Darussalam

Corresponding author:
Mathias Foo, School of Engineering, University of Warwick, Library Road, Coventry, West Midlands, CV4 7AL, UK.
Email: M.Foo@warwick.ac.uk

Keywords
Embedded control systems, engineering education, smart farming, electromechanical devices, Arduino, sensors, interdisciplinary engineering

Introduction

It is vital to equip the next generation of engineers with a high-level of critical thinking and creativity[1] to enable them to contribute to global societal development,[2] which is why this article is centred around bridging the gap between current engineering education and practical applications in industry. This article leverages the fundamental understanding of interdisciplinary engineering concepts and provides an opportunity to apply these concepts through design and analysis to address a problem in the agricultural context.

According to the United Nation's projection, the world population will reach 9.7 billion by 2050. This is expected to create demands for global agricultural production to rise 69% between 2010 and 2050.[3] In order to meet these demands, it is crucial for farmers to leverage data analytic models and predictive technologies to achieve greater production capabilities, simultaneously improving both yield increase and cost reduction aspects.[4] At the same time, the drive to produce locally to reduce the carbon footprint contribution from transportation also requires the use of technology in farming to create a suitable climate for the vegetation to grow off-season.

Smart farming, a farming approach that employs information and communication technologies, is deemed as the viable solution to improve farming practices to increase production sustainably[5–7] It is a data-driven approach that leverages advanced technologies such as big data analytics, artificial intelligence (AI), machine learning, and cloud computing to improve the efficiency, productivity, and sustainability of farming practices.[8] Smart farming uses a network of sensors and Internet of Things (IoT) devices that collect data about various aspects of the farming environment, such as ground mapping robots, weather monitoring, and even livestock behaviour monitoring.[9] It can even be adapted to urban farming settings by providing real-time monitoring and analysis of critical plant health parameters. A recent study shows how AI has crucially contributed to sustainable solutions as fresh produce can now be harvested in urban settings when traditional agriculture is not possible.[10] Furthermore, it reduces the carbon footprint of food production by reducing transportation distances and associated emissions, as well as providing opportunities for social and economic development in urban areas.[11] All these data from smart farming can be utilised to generate insights and recommendations for optimising crop yields, reducing waste, and improving resource management using computational algorithms.[12] The backbone architecture to implement these resource managements is the feedback control system.[13,14]

The aim of this article is thus to propose practical teaching material that could devise a smart farming methodology that is accessible, adaptable, and sustainable. The teaching material will be driven by the embedded control system design of electromechanical devices to introduce students to emerging sensing and embedded systems technologies.

This practical teaching material can be adopted for laboratory activities for engineering students, including systems engineering, electronic engineering, mechanical engineering, mechatronic engineering, and computer engineering. While the application addressed in this practical teaching material is directly related to agriculture, the sensors, control and software technologies that are involved are easily transferable to all the engineering fields mentioned above, therefore it is suitable as part of the curriculum of any of those engineering courses.

The central objective of the proposed teaching material on engineering education is to give students an opportunity to translate their existing interdisciplinary engineering knowledge and technical skills into a practical context. The laboratory will outline a low-cost and adaptable smart farming method that can be integrated into a controlled-farming environment, where it offers a cost-effective solution using readily available electromechanical components that outperform current manual farming methods. Thus, through reproducing this solution, students can understand how significant embedded control design and electromechanical devices can be applied and utilised in an industrial context beyond the scope of their theory-based engineering lectures. Students should also ensure that their solutions are logically structured around the project costs, time frames and other performance factors.

This teaching material is developed based on a project done for the first author's third (final) year individual project for the Bachelor of Systems Engineering degree programme, and this design exercise is to be used as one of the assignments for the module called 'Advanced Systems and Software Engineering' offered in the third year of the degree programme.

System overview

The smart farming system consists of the following component groups as illustrated in Figure 1:

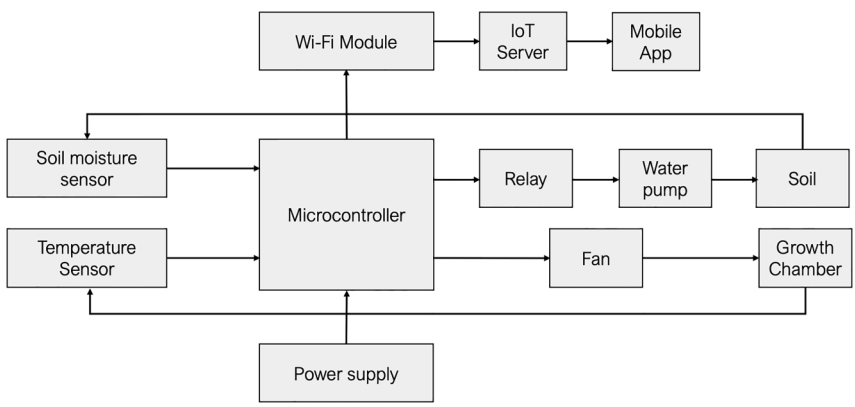


Figure 1. Smart farming system block diagram.

1. **Microcontrollers:** Arduino Uno R3 including internal communication network
2. **Power supply:** 9V battery for the Arduino and subsequent actuator needs
3. **IT and OT architecture:** ESP8266-01 Wi-Fi module, IoT server, and Blynk app
4. **Measurements:** Soil (moisture), growth chamber (ambient temperature)
5. **Sensors:** Temperature sensor, soil moisture sensor, and relay
6. **Actuators:** Fan and water pump.

The smart farming system is designed to utilise a set of sensors, to measure some environmental variables that affect the growth of the plants. An embedded control approach is applied, utilising a simple Arduino Uno as the controller. The objective is to monitor the ambient temperature and the soil moisture, and the controller is used to ensure these variables are maintained at the level for optimal growth of the plant. From a control engineering perspective, this is equivalent set points (reference) tracking problem. The control algorithm is used to activate the actuators, that is, the fan and water pump, to regulate the temperature and the soil moisture to reach their respective set points. The details of the system components and the development are discussed in the next section.

System development

The system development was expected to provide students with the knowledge and skill to do a step-by-step engineering system design. This includes the selection of the suitable components as listed in the previous section, the calculation of the power requirement, the design of the electronic circuit and the software for the control function and the user interface, respectively, and carrying out all the testing and integration.

Sensors

Sensors are required to detect and measure signals from certain physical variables on the conditions of the system environment. An electronic sensor is connected to a data acquisition device where the measurement data are sent for further processing or use. Here, the two sensors of interest are the temperature and the moisture sensors. The implementation of these sensors involves familiarisation with the basic software aspects of Arduino IDE, exercising foundational circuit knowledge including breadboard electronic rules and calibration techniques. One important thing to note when implementing several sensors together is that data from each one sensor should not interfere with the other sensors, and should be logged and stored separately. This allows individual testing, which also facilitates the troubleshooting and debugging processes. The I/O (input/output) testing approach is widely recognised in the industry for its ability to help engineers efficiently identify issues, instil confidence in clean and maintainable code, and enhance the system's performance, reliability, and user satisfaction.[15,16]

Microcontroller and embedded control

An Arduino Uno is used as the controller. This microcontroller is chosen due to its affordability, wide compatibility with various other components and is simple to program without the need for high programming skills. The Arduino Integrated Development Environment, also known as Arduino Software (IDE)[17] that is quite easy to learn and has also included the text editor, is sufficient to implement the program to control this smart farming system. This makes it an ideal choice for any engineering students wanting to learn skills such as electronics design and embedded control. If more input and output ports are needed, for instance, if more sensors are included, using an Arduino Mega would be an alternative.

An embedded feedback control system is essential for a smart farm to ensure that a controlled environment can be created and maintained. The feedback control system should be able to access the information acquired by each of the sensors to detect any deviation from the ideal operating conditions. Any deviations detected can then trigger responses to the electromechanical devices to output corrective actions, in this case changing the temperature and moisture via the fan and water pump, respectively.

Wi-Fi module

Although the engineering of feedback control has been successfully implemented, it is constrained by a wire that is connected to a laptop if data would be streamed from the Arduino. This is not a practical design aspect, thus adding wireless compatibility would ensure practicality for real-world use. Unfortunately, the Arduino Uno R3 does not have an inbuilt Wi-Fi functionality, so it is paired with a low cost ESP8266-01S Wi-Fi module to send and receive data over Wi-Fi. This will allow the system to log data and output control signals to the electromechanical devices without any manual or wired interaction.

To allow a seamless exchange of information between microcontrollers, computers and electronic components, specific protocols are in place to facilitate their communication and there are multiple methods to achieve this. The protocol UART (Universal Asynchronous Receiver/Transmitter) is a hardware communication interface that allows serial communication between devices. Additionally, Wi-Fi is a wireless communication technology that enables these devices to connect to a local area network (LAN) or the internet wirelessly.

While connection via USB was the most common approach, this entailed a dedicated wire connection from the computer (host) to the Arduino (client). The UART serial communication method was used instead as it has two wires TX (for transmitting data) and RX (for receiving data) directly to the ESP8266-01 Wi-Fi module without the need to wire Arduino to the laptop. Although UART has the advantage of receiving serial data in parallel form, there are more chances of losing the data. To minimise this risk, it is recommended to promptly transfer the data to remote cloud storage. This can be achieved by establishing a process where data from IoT devices are transferred immediately and securely to remote cloud storage methods. By doing so, the data are stored in a centralised

```
void loop()
{
  Blynk.run();       //Continously runs Blynk connection
  timer.run();       //Continously runs Blynk timer
}
```

Figure 2. Blynk connection code.

and more secure environment, reducing the likelihood of data loss, unauthorised access, or device malfunctions.

Once the Wi-Fi connection is set up, it is important to consider how the system should mitigate any external connection issues. For instance, there is a momentary Wi-Fi signal loss in the initial setup that has caused the system to crash completely and an error message is displayed when the Wi-Fi module is plugged into the laptop. To overcome this, a '`Blynk.run()`' script as shown in Figure 2 is added at the start of the `void loop()` in Arduino IDE code to ensure that the system continuously attempts to connect. The system outputs '`NaN`', which stands for 'Not A Number', for any readings missed and proceeds with the code execution without the need to manually reconfigure the Wi-Fi connection.

IoT-based mobile app (Blynk)

A user-friendly software application (both for the web and app interfaces) is developed, where plant health can be monitored with real-time updates. The app displays relevant plant data from the controlled environment and appropriately stores the data in a cloud-based IoT device. Utilising data transformation tools, the app presents interactive ways for users to explore their data, providing valuable insights instead of burdening them with raw data and manual analysis. Additionally, the app allows seamless data streaming from the Arduino device, either through direct readings from the serial monitor or by exporting data to .csv format. This showcases that the purpose of this app is not solely to stream data; its forefront functionality is to provide insightful and informative metrics to the farmers.

The inclusion of real-time charts in data visualisation simplifies pattern identification and outlier detection within large data sets, facilitating faster decision-making by leveraging this pre-processed data. Another notable feature of the app is its notification system, which alerts users exclusively when an actuator is activated. Additionally, the app offers an action log history alongside real-time updates, enabling users to review any missed activities.

Growth chamber and hardware configurations

The sensors and the electromechanical devices must be installed appropriately such that they can be operated within the smart farming context. A growth chamber suitable for an indoor farming environment can be constructed to directly compare growth rates. Here,

the construction of the suitable plant growth chamber and the configuration of the hardware used are discussed.

Growth chamber assembly

A cardboard box is used as the growth chamber, with the dimension of $L = 43$ cm; $W = 56$ cm; $H = 32$ cm as shown in Figure 3. A separator is used to divide the chamber into two, side-by-side, such that two experiments, that is, plant growing with and without feedback control can be conducted simultaneously and in quite an identical environment. A standard plant grow LED (light emitting diode) light is used to provide the required sunlight for the plant to grow. The light is programmed using a standard timer switch to switch-on for 16 h and switch-off for 8 h mimicking a typical 'long-day' lighting exposure for garden cress.[18,19]

Temperature sensor

The TMP36 temperature sensor is used in this development. This sensor is a standard sensor that usually is provided with the Arduino Starter Kit. The documentation[20]

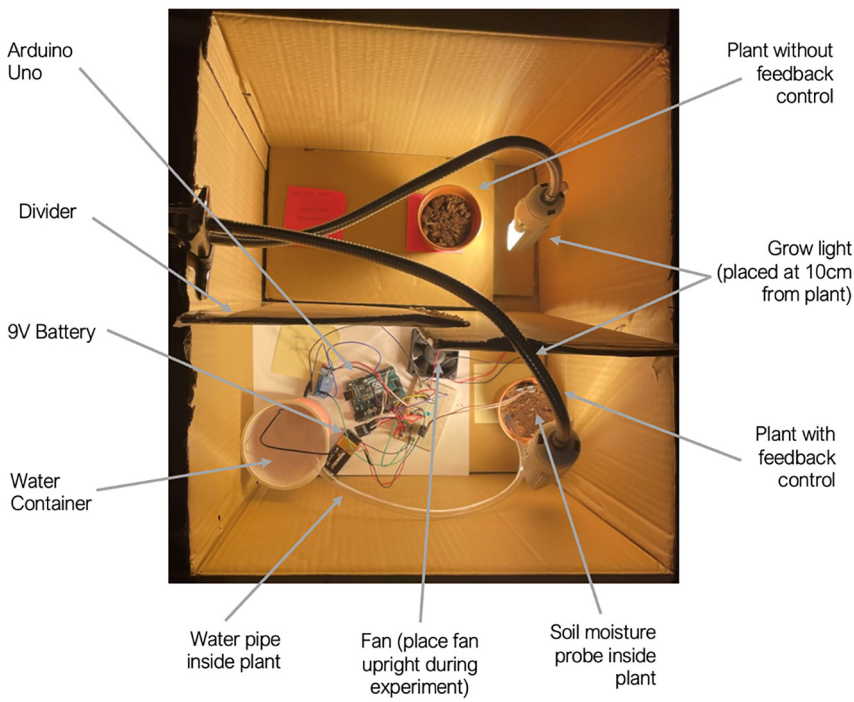


Figure 3. Growth chamber and embedded control setup for smart farming.

suggests that this sensor is not weatherproof, so this component would be compromised if it is directly exposed to weather elements. Nevertheless, this sensor is suitable for an indoor experiment. The pinout configuration of the temperature sensor is shown in Table 1.

The output voltage measurement on the pin A0 from the sensor needs to be converted into the temperature value in Celsius (°C) to ensure any meaningful interpretation of the data. The steps required to perform the conversion are summarised as follows:

1. **Record the sensor voltage output level** `A0_Vout` **on pin A0.**

$$\texttt{A0_Vout} = \texttt{analogRead}(\text{A0});$$

2. **Find the sensor output as a fraction of 5 V.**
   Arduino Uno yields resolution of 5/1024 (V/level)

$$\texttt{Voltage_Per_UnitLevel} = (\texttt{A0_Vout} * 5)/1024;$$

3. **Correct the sensor range by subtracting the offset.**
   The datasheet[20] indicates that the output configuration of 0–1.75 V corresponds to the temperature −50 °C to 125 °C. Therefore, every 0.01 V corresponds to 1 °C as shown in the calculation below:

$$\text{Celsius toV volt resolution} = \frac{(1.75\ \text{V} - 0\ \text{V})}{(125\,^{\circ}\text{C} - (-50)\,^{\circ}\text{C})} = 0.01\text{V}/^{\circ}\text{C}$$

   As 0 V corresponds to −50 °C, an offset of $-50\,^{\circ}\text{C} \times 0.01\text{V}/^{\circ}\text{C} = -0.5$ V is introduced.
4. **Convert from millivolts (mV) to degree Celsius** (°C)**.**
   The datasheet[20] states that the TMP36 has 10 mV resolution, therefore, multiply the calibrated `A0_Vout` by $10^3/10 = 100$ °C/V.

Table 1. TMP36 pinout

Pin name	Arduino connection	Description
PIN 1 (+VS)	+5 V	+VS pin powers the module with operating voltage 2.7–5.5 V
PIN 2 (VOUT)	A0	Output voltage mapped between level 0 and 1023
PIN 3 (GND)	GND	Ground pin connected to the power supply ground (0 V)

The steps above can be described using the following equation:

$$\text{Temperature } (^{\circ}\text{C}) = 100\left[\texttt{A0_Vout}\left(\frac{5}{1024}\right) - 0.5\right]$$

Soil moisture sensor

The moisture sensor used here is the FC-28 Soil Moisture Resistor with LM393 Comparator Chip. An electrical current of 35 mA is sent between each of the electrode probes to measure the soil resistance between them. For the soil with a high water content, there is a higher electrical conductivity and thus a lower resistance to the current flow because water is a good conductor of electricity.

The Ohm's law, that is, $V = IR$, was used to deduce the relationship between resistance R and voltage V, with I the current. It means the lower the resistance, the lower the output voltage sensor reading will be. Additionally, it can be deduced that the higher the moisture content is in the soil, the lower the output voltage sensor reading will be, and vice versa.

The pinout configuration of soil moisture sensor FC-28 is given in Table 2. The soil moisture sensor is calibrated (0% to 100%) according to its contextual conditions. The calibration procedure can be summarised as follows:

1. Take the output voltage reading level for completely 'dry' (no water in the soil).

```
M_Dry =analogRead(A1);
```

2. Take the output voltage reading level for completely 'wet' (fully submerged in wet soil).

```
M_Wet =analogRead(A1);
```

3. Map the dry and wet readings to 0% and 100%, respectively.

```
Soil_Moisture_Pin =analogRead(A1);
Soil_Moisture_Value = map(Soil_Moisture_Pin, M_Dry, M_Wet, 0, 100);
```

Table 2. LM393 pinout.

Pin name	Arduino connection	Description
Vcc	+5 V	Pin powers the module with operating voltage 3.3–5 V
GND	GND	Ground pin connected to power supply ground (0 V)
A0	A1	Analog out pin for analog output between level 0 and 1023

4. Repeat steps 1 to 3 for at least five times and find the respective averages.

In this experiment, the soil used was characterised by a dry soil variable `M_Dry` value of 813 and a wet soil variable `M_Wet` value of 454. Through several calibration exercises, it was found that the relationship between these values is linear, which is then mapped to a range of 0 to 100. This allows for a uniform distribution to be established between the minimum and maximum values, ensuring consistent and proportional representation throughout the range.

Electromechanical axial fan

A 12 V, 0.16 A axial fan is used to directly regulate the temperature of the growth chamber. To control the temperature, using a fan is sufficient and no need for a heater, because the temperature of the growth chamber does not go below the room temperature (i.e. between 18 °C and 20 °C). This range of minimum temperature is not detrimental to the plant health as opposed to an excessively high or low temperature. A reference value or set point of 22 °C is chosen[21] with a tolerance of ±1 °C acting as a dead-zone from this reference temperature to ensure the fan would not continuously turn on and off excessively when the temperature gets close to the set point. To enable the variation of the fan speed, a pulse width modulation (PWM) is introduced which depends on how significant the temperature deviation is from the reference value.

Electromechanical water pump

A submersible 3 V DC water pump is connected to a water pipe tubing (diameter 0.7 cm) to alter the soil moisture content. An external power source (9 V) is also introduced at this stage to remove the dependency on the laptop battery as the actuator primary source of power. Unlike the fan that could be switched on for a full iteration duration (15 min), additional care needs to be taken when switching on the water pump as a mere 3 s of switching on the pump could cause flooding and drowning the plant. The watering increment (the duration for which the pump is switched on), in this case, the '0.5 seconds' could be altered in the future depending on the specific plant or number of plants to be watered at the same time. In addition, a 3-hour iteration is deemed reasonable as one tends to water plants 1 to 2 times a day so collating data for 4 to 8 times per day is reasonable.

In this small-scale project, a 9 V battery was used and it was sufficient to run the system for 16 days to verify the functionalities of the system. If we were to run the system for more than 16 days, we would need to connect the wall wart to the main power supply. However, if the system is to run continuously for a longer time, where no power supply is available around the smart farming area, ideally a solar panel with a battery will be used.

Wiring configuration and Arduino flowchart

The wiring diagram of the connection from the Arduino Uno to the other components, that is, the sensors, actuators (the two electromechanical devices), the Wi-Fi module, and the power source, is shown in Figure 4. The Arduino flowcharts describing the algorithms to turn on the fan and water pump are shown in Figure 5.

Blynk web and mobile app interfaces design

To set up the Blynk networking for a smart farming app, the Blynk library needs to be configured to connect to a local access point using the user's specific SSID and password for their Wi-Fi. Furthermore, in order to ensure secure communication between the app and the connected devices within the smart farming system, it is necessary to generate Blynk security credentials such as the 'Auth Token' from the account area.

Blynk IoT is a commercial app that offers a comprehensive suite of tools, protocols, and integrations, enabling farmers to leverage its platform for efficient device management, real-time data monitoring, and seamless connectivity, thus driving smart farming innovation within the IoT ecosystem.

Blynk primarily uses the Internet Protocol (IP) for networking, allowing communication over the Internet. It relies on the Transmission Control Protocol (TCP) to establish connections between the Blynk app and the Arduino or other hardware devices. TCP provides a reliable and ordered data transmission between the Blynk app and the hardware device, ensuring that data packets are delivered correctly and in the correct order.

Transmitting the measured data from the smart farming system to a mobile app interface will help users make more informed decisions and respond more quickly to system changes as all parameters are logged in a useful form. The two key functions of the mobile app are to monitor and control the smart farm measured variables. The control function is done by sending a command to activate the two actuators in the system. These two functions can be found on the `Dashboard` and `Control` tabs as illustrated in Figures 6 and 7, respectively.

Interface design

Students could introduce different user interfaces and visualisation elements such as dials and graphs from the Blynk Widget Box to help users interpret the data generated by the system by converting it into a graphical and summarised form. Once the design has been completed, each element can be individually configured by adding script so the graphic can achieve real-time interaction and analytics generated by the Arduino.

A 'virtual pin' can be used to direct relevant variables outputs from Arduino to Blynk. This virtual pin will be used as a datastream storage that multiple elements can utilise; for example, the temperature sensor will be fed V1 (virtual pin) datastream corresponding to `Temperature_Value`, following the script shown below:

```
Blynk.virtualWrite (V1, Temperature_Value);
```

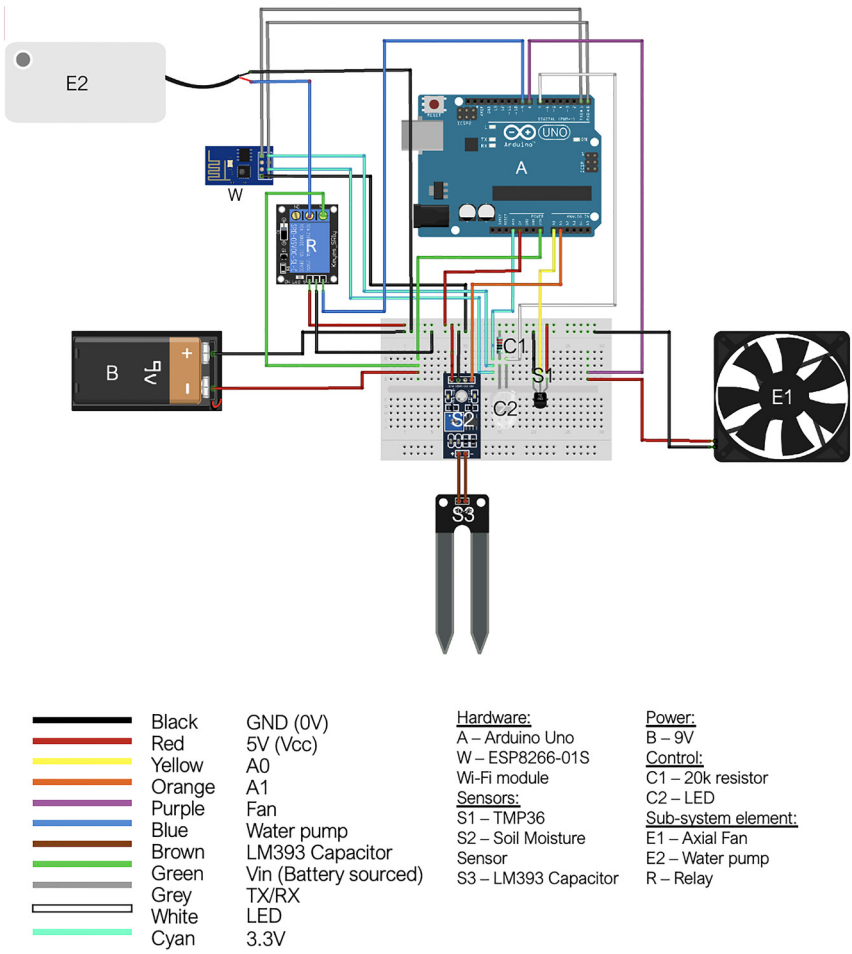


Figure 4. Hardware circuit diagram for the embedded control of two electromechanical devices, i.e., fan and water pump.

It is important that there should be a consistent and synchronous design between the web dashboard and the app dashboard from an aesthetic and functional points of view. For example, if a switch on one dashboard is toggled, the other dashboard will also be updated.

Dashboard

The `Dashboard` tab displays real-time analytics of the two factors that affect plant growth. The plots could provide better visualisation of the data making it easier to

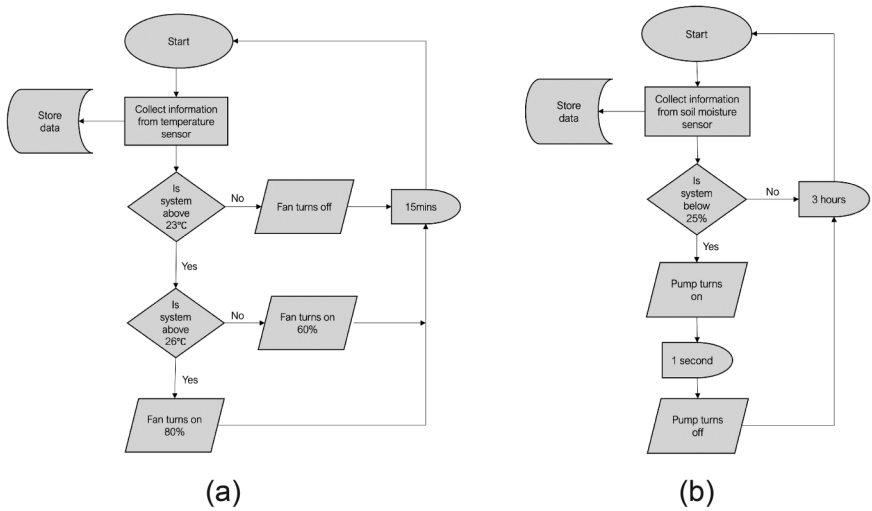


Figure 5. Flowchart for the control and operation of (a) the fan and (b) the water pump.

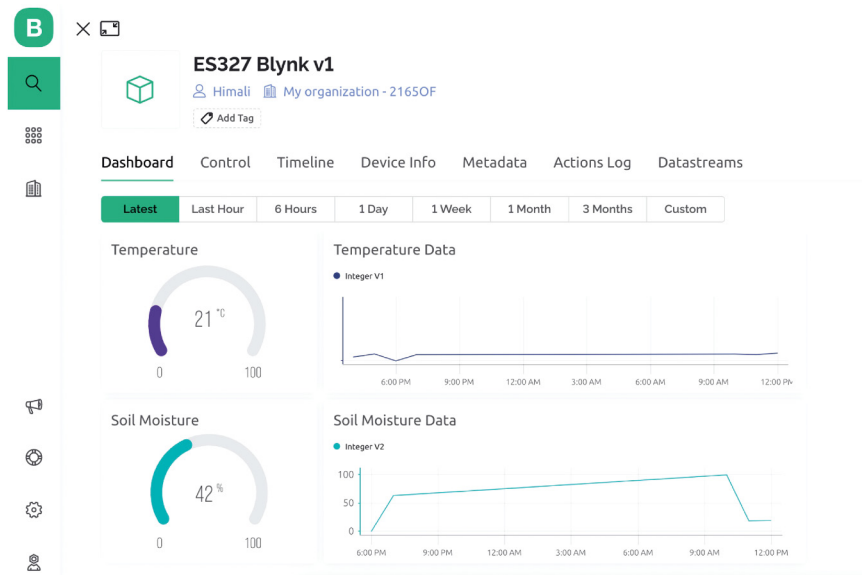


Figure 6. A screenshot of web interface `Dashboard` tab designed using Blynk that displays the measured data from the temperature and soil moisture sensors.

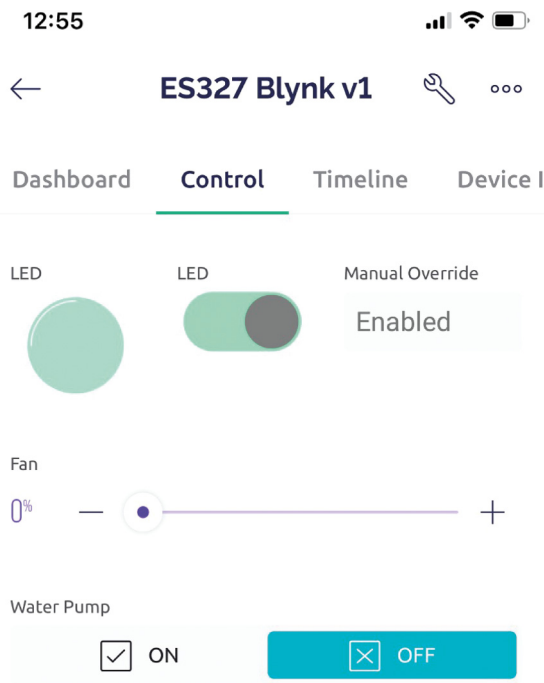


Figure 7. A screenshot of App interface `Control` tab designed using Blynk that displays the functions to perform manual control to override the automated system.

highlight the trends and outliers quickly. Data insights can be generated over varying time frames to allow adaptability for different plants and their respective life cycle lengths.

Control

The `Control` tab allows users to provide manual control to override the automated system. This manual control functionality allows users to perform fine tuning or to make temporary changes based on their judgement without having to alter or disrupt the script.

Students can actively consider the system's practical aspects and implement them into the app design as well. For example, the user could add elements to the app to check for continuous wireless connection between itself and the Arduino through a simple toggle to the LED on/off switch. In our development, the LED C2 in Figure 4 serves this purpose.

Results and discussion

Experimental setup

To test the functionality of the smart farming setup and also the app, an experiment was conducted. For the experiment, two pots of garden cress (*Lepidium sativum*) and the hardware components were placed in the growth chamber following Figure 3. The descriptions of the two plants under consideration are given below:

- Plant 1: The temperature and soil moisture were regulated to optimal conditions as previously outlined in the growth chamber design section above.
- Plant 2: The temperature was regulated at room temperature, which lied within the healthy range. This setting was to ensure that there was no risk of a preceding high temperature, which may cause the plant enzymes to denature, and no risk of photosynthesis was prevented from taking place. Plant 2 was manually watered twice a day to the extent deemed necessary.

Measurements of the temperature were acquired every 15 min, while the measurements for the soil moisture were acquired every 3 h as discussed above. A simple on/off control was used to switch on and off the two electromechanical devices. The fan would turn on when the chamber temperature was higher than the upper limit ($> 23\,^{\circ}C$) and the axial pump would turn on when the soil moisture was lower than the lower limit ($< 25\%$) As mentioned in the design section, rather than using the full 3 h measurement cycle, the pump timer was set for 0.5 s so that enough water was released to increase the soil moisture without overwatering it.

Hypothesis

The experiment was conducted to demonstrate the effectiveness of the deployment of the embedded control of the electromechanical devices. From this experiment, students can be expected to form the following hypothesis.

> The plant grown with the feedback system (Plant 1) will develop faster and produce a healthier yield in the same amount of time than the plant grown without the feedback system (Plant 2).

Students can address the hypothesis using both qualitative and quantitative perspectives to ensure a conclusive justification that could enhance the reliability of the results.

Qualitative results

An expected outcome of this experiment was shown in Figure 8. The experiment was conducted over a 16-day lifecycle (i.e. when the cress should be ready for harvesting according to the seedling package instructions). Plant 1 clearly showed better qualitative growth in terms of the number of germinated (successful sprouting of seeds[22]) seedlings

and healthier leaf characteristics compared to Plant 2. It was important to note that Plant 2 could still reach further growth however, this was beyond the considered lifecycle range. For students to relate these results to actual practice, in the context of conventional farming practices, Plant 2 represents a typical growth rate using traditional farming methods, while Plant 1 represents the accelerated growth rate with higher yield in terms of the lifecycle when employing feedback control. The major difference in yield production between the plants at the end of the defined lifecycle can provide justification for the significance of embedded control of electromechanical devices for improved farming.

Quantitative results

While qualitative results are a good indicator of plant productivity, students should also learn to analyse quantitative results. Students can learn to decide on which important metrics that can be used to quantify the results. Here, the number of successful germination, stem length, leaf width and number of leaves bloomed measured at day 16 can be used as the quantitative metrics.

Plant 1 has a 92% germination success (23/25), whereas Plant 2 had a 64% germination success (16/25). The averages of the other quantitative metrics are calculated based on the number of germinated seeds and they are summarised in Table 3.

Students can be encouraged to present the results in Table 3 in a graphical manner. Given the measurements were done based on the average of successfully germinated seeds with an error bar, the most suitable graphic representation would be the bar graph. An example of the bar graphs depiction is shown in Figure 9. These bar graphs enable clear visual comparison that Plant 1 appeared to outperform Plant 2 in all three metrics indicating that better growth is achieved when the important growth factors such as temperature and soil moisture[23] are regulated by the embedded controlled system but a statistical test was needed to accept or reject the hypothesis.

Another aspect of analysing quantitative results that students can be aware of was the measurement error. This error analysis was of particular importance in biological measurements.[24] Although the two stem length averages had a small difference (of $45.4 - 28.4 = 17$ mm), taking the ratio of stem length of Plant 1 to Plant 2 showed that Plant 1 notably grew 154% times higher than Plant 2. Students can also note that the stem

Table 3. Results summary of the quantitative metrics used to analyse plant growth.

Plant	Average stem length (mm)	Average number of leaves per stem	Average largest leaf width (mm)
1	45.43 ± 5.73	5.48 ± 0.79	3.52 ± 0.59
2	28.38 ± 4.60	4.63 ± 1.15	3.13 ± 0.72

Note that these results are calculated based on the number of successful germinated seeds for both plants. The numbers after the $\pm$ sign denote standard deviation.

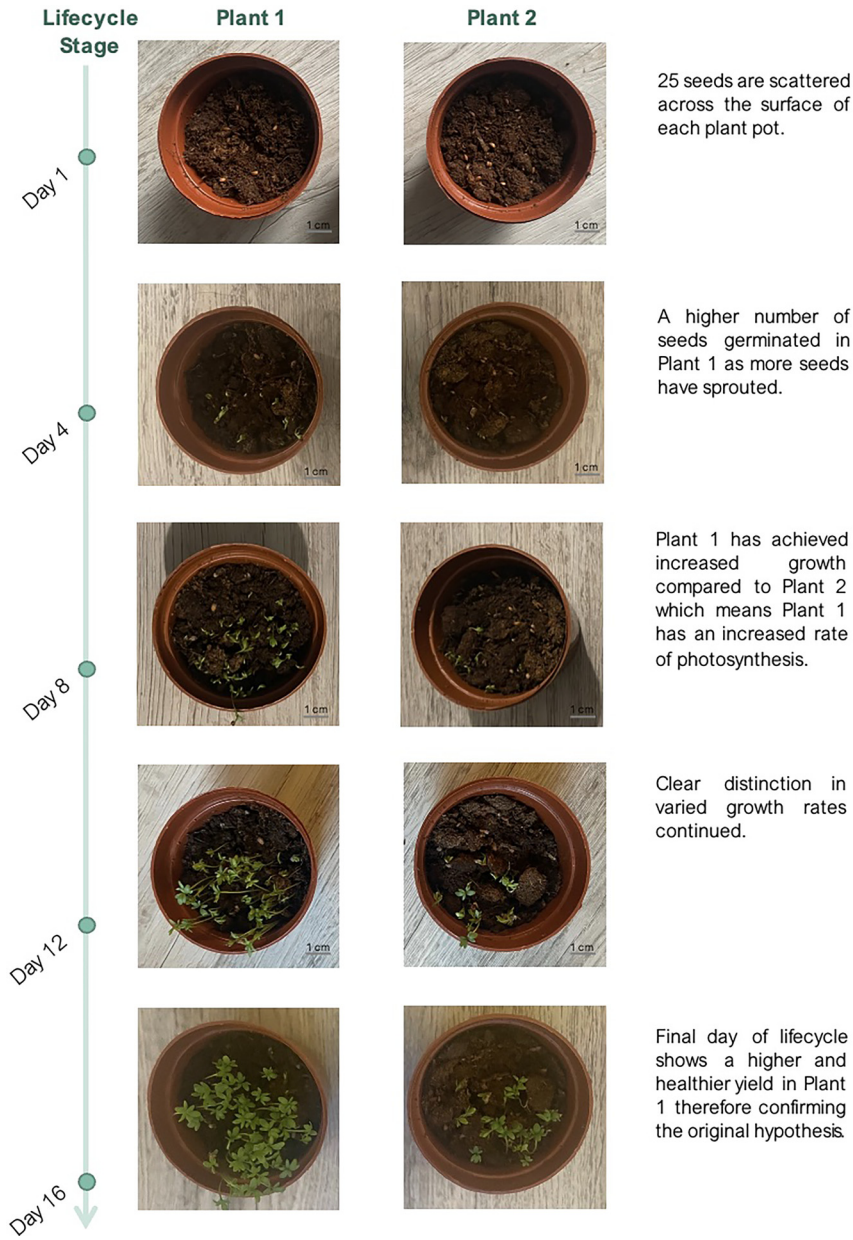


Figure 8. Lifecycle stage of plant grown under two different conditions. Plant 1 and Plant 2 are grown with and without feedback control deployment, respectively.

length and the leaf width measurements had an uncertainty of ±0.5 mm due to the 1 mm resolution of the ruler that was used to collect the results.

Conclusion

An embedded feedback control system using Arduino employing electromechanical devices has been built for application to a lab-scale smart farming system. An experiment using garden cress has been conducted to test the hypothesis and give students an opportunity to translate their engineering skillset within the practical context of agriculture. At the end of the defined lifecycle, the plant grown in the growth chamber equipped with feedback control demonstrates better and healthier growth compared to the plant grown without feedback control. A web and app interfaces are also designed to provide real time and historical measurements of ambient temperature and soil moisture useful for plant growth. We expect that instructors and particularly students who are interested in the practical application of embedded control of electromechanical devices could benefit from the design of this article.

Modification and extension are very possible, depending on the needs. For instance, if the effect of the light cycle on plant growth is to be considered, then a light sensor and artificial light can be added to the system. The smart farming system's inherent flexibility and expandability make it an invaluable learning tool for students, as it can be customised to cater to their specific needs in developing various technical or interdisciplinary application skills.

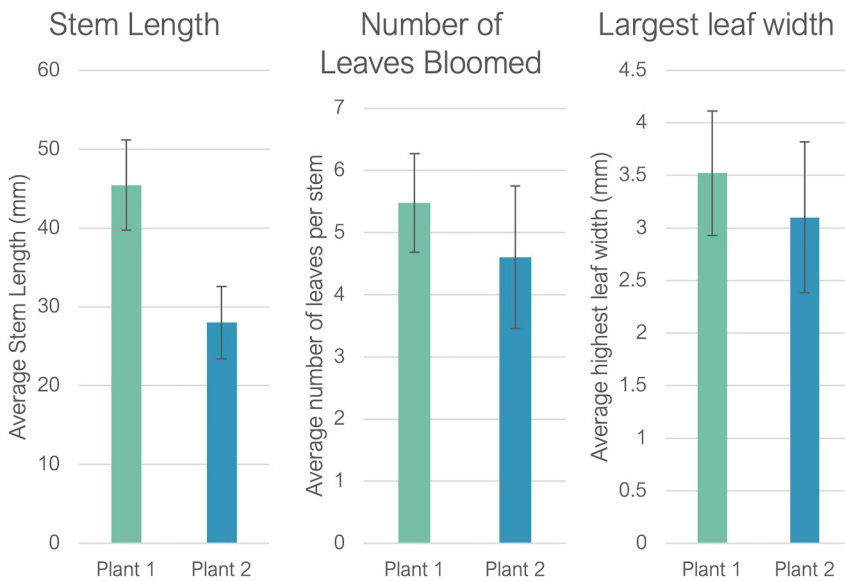


Figure 9. Graphical representation of the experimental data for better visualisation.

Acknowledgements

The authors would like to thank the students and teaching staff from the School of Engineering at University of Warwick and the Faculty of Electronic Engineering at Universiti Teknologi Brunei for all the fruitful discussion time.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research and/or authorship of this article: This work was supported by British Council through UK-Brunei Higher Education Teaching and Learning Partnership grant.

ORCID iDs

Himali Mistry https://orcid.org/0009-0000-8272-1044
Mathias Foo https://orcid.org/0000-0003-1400-2659

Data Availability Statement

All the IDE codes and script for App Design are available and can be downloaded from https://github.com/himalimistry/IJMEE-Smart-Farming-System

Supplemental material

Supplemental material for this article is available online.

References

1. Ahern A, Dominguez C, McNally C, et al. A literature review of critical thinking in engineering education. *Stud Higher Educ* 2019; 44: 816–828.
2. Burleson G and Austin-Breneman J. Engineering for global development: Characterizing the discipline through a systematic literature review. In*International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers, Vol. 84010, pp. 1–11.
3. Meola A,. Smart farming in 2020: How IoT sensors are creating a more efficient precision agriculture industry. *Bus Insider* 2020.
4. Neethirajan S and Kemp B,. Digital livestock farming. *Sens Biosensing Res* 2021; 32: 100408.
5. Walter A, Finger R, Huber R, et al. Smart farming is key to developing sustainable agriculture. *Proc Natl Acad Sci USA* 2017; 114: 6148–6150.
6. Mohamed ES, Belal A, Abd-Elmabod SK, et al. Smart farming for improving agricultural management. *Egypt J Remote Sens Space Sci* 2021; 24: 971–981.
7. O'Shaughnessy SA, Kim M, Lee S, et al. Towards smart farming solutions in the US and South Korea: a comparison of the current status. *Geogr Sustain* 2021; 73: 312–327.
8. Dineva K and Atanasova T,. Cloud data-driven intelligent monitoring system for interactive smart farming. *Sensors* 2022; 22: 6566.

9. Dagar R, Som S and Khatri SK,. Smart farming—IoT in agriculture. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, pp. 1052–1056.
10. Podder AK, Bukhari AA, Islam S, et al. IoT based smart agrotech system for verification of urban farming parameters. *Microprocess Microsyst* 2021; 82: 104025.
11. Colson-Fearon B and Versey HS,. Urban agriculture as a means to food sovereignty? A case study of Baltimore City residents. *Int J Environ Res Public Health* 2022; 19: 12752.
12. Tantalaki N, Souravlas S and Roumeliotis M,. Data-driven decision making in precision agriculture: the rise of big data in agricultural systems. *J Agr Food Inform* 2019; 20: 344–380.
13. Chetty V, Woodbury N and Warnick S,. Farming as feedback control. In *2014 American Control Conference*. IEEE, pp. 2688–2693.
14. Castañeda-Miranda A and Castaño-Meneses VM,. Internet of things for smart farming and frost intelligent control in greenhouses. *Comput Electron Agr* 2020; 176: 105614.
15. Tillmann N, De Halleux J and Xie T,. Parameterized unit testing: Theory and practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*. pp. 483–484.
16. Ellims M, Bridges J and Ince DC,. Unit testing in practice. In *15th International Symposium on Software Reliability Engineering*. IEEE, pp. 3–13.
17. Arduino. Overview of the Arduino IDE 1. [Accessed: March 21, 2023]. https://docs.arduino.cc/software/ide-v1/tutorials/Environment.
18. Sattari Vayghan H, Tavalaei S, Grillon A, et al. Growth temperature influence on lipids and photosynthesis in *Lepidium sativum*. *Front Plant Sci* 2020; 11: 745.
19. Tarkowská D, Krampolová E and Strnad M,. Plant triterpenoid crosstalk: the interaction of brassinosteroids and phytoecdysteroids in *Lepidium sativum*. *Plants* 2020; 9: 1325.
20. Rev H,. *Low Voltage Temperature Sensors*. Analog Devices, 2015. https://www.analog.com/media/en/technical-documentation/data-sheets/3600fd.pd.
21. Hatfield JL and Prueger JH,. Temperature extremes: effect on plant growth and development. *Weather Clim Extremes* 2015; 10: 4–10.
22. Bewley JD, Bradford KJ, Hilhorst HW, et al. Germination. *Seeds: Physiology of Development, Germination and Dormancy, 3rd Edition* 2013; 133–181.
23. Al-Kayssi A, Al-Karaghouli A, Hasson A, et al. Influence of soil moisture content on soil temperature and heat storage under greenhouse conditions. *J Agr Eng Res* 1990; 45: 241–252.
24. Houle D, Pélabon C, Wagner GP, et al. Measurement and meaning in biology. *Q Rev Biol* 2011; 86: 3–34.