Knowledge Based Article

Smart Watering System

*Kacper Popis*

CMP 4267 – Computer Systems

Birmingham City University

Birmingham, United Kingdom

kacper.popis@mail.bcu.ac.uk

*Abstract* — In agriculture, irrigation systems are installed into plots of farmland to water crops in a much more time-efficient manner rather than the farmer watering crops themselves resulting in less strain on the farmer. However typical irrigation systems are not as efficient as they can be as most tend to run at specific times without checking water presence in the soil and usually overwatering the crops. In this article, a prototype of a Smart Watering System will be explained, along with a logic gate diagram to give a greater understanding of the prototype. Additionally, all hardware components will be justified as to why they have been included and how each component interacts with one another to create a complete system. Potential changes and improvements will be discussed along with any security concerns that may arise.

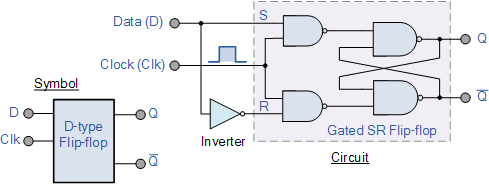
# Introduction

IoT, also known as Internet of Things, is a constantly evolving network of physical devices (such as sensors and vehicles) connected to the internet with the ability to interact with each other. Device communication over the internet is done via protocols, the most common being TCP / IP (Transmission Control Protocol / Internet Protocol). With the introduction of more physical objects into IoT over the years (like washing machines and fridges), the hardware itself has gotten more compact and simpler which allows for more possibilities of digitalisation in industries, a major one being agriculture. It is estimated that there are “over 14 billion connected IoT devices” [1] and “it’s expected there will be 25.44 billion IoT devices by 2030” [1].

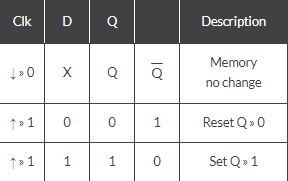
As an example, one way of implementing IoT would be introducing a camera surveillance system inside a building and connecting all camera devices into one network to provide a live view of all the cameras which can be accessed from anywhere – either on-site via a connected desktop or through a mobile application that connects to the devices and displays the camera’s feed. Another example would be the implementation of RFID (Radio-Frequency Identification) tags and scanners for data collection and security purposes. With this technology, you can get more accurate readings of when certain actions are taken, such as an employee clocking in or out of work. This data can then be relayed directly to a database allowing anyone with access right to view all the data.

With these technical developments of IoT, there have been multiple benefits to digitalisation in the agriculture industry as it can help optimise the current methods of farming and crop production while keeping waste (of both water and crops) and costs as minimal as possible. Research shows that “agriculture uses up approximately 65% of freshwater but almost half of it is currently wasted” [2] and through developing and implementing this product the amount of water waste can be reduced drastically. Through responsible consumption of precious resources, this product assists with one of the UN (United Nations) Goals – goal 12, responsible consumption and production. By managing how much water is supplied to the crops and when it is applied, better quality crops can be produced as they will not be too watered down and the soil itself will sustain less damage than it would if a regular irrigation system were used. This method will also cut down costs for the farmer as less money will have to be spent on the water as water from the irrigation system will only be used when necessary, compared to being used on a time-based cycle. Another UN Goal that this product can help to attain is goal 2 – zero hunger. Through this product, greater yields of healthy crops will be produced as fewer crops will die from both underwatering and overwatering as the sensors implemented into the prototype will monitor the real-time conditions of the crops and the system will react corresponding to the data it records.

The Arduino Uno R3 that is used in this prototype consists of a flash memory with a size of 32KB (kilobytes) and a processor with a clock speed of 16MHz (megahertz). Although this microcontroller works similarly to a desktop computer it lacks the processing power of a computer and an operating system itself. The Arduino board is a circuit manufactured onto a circuit board with chips (such as memory) and ports (like USB ports) installed into it. The CPU (Central Processing Unit) works hand-in-hand with memory to execute processes as the two components are linked together via buses like a data bus and an address bus. An address bus carries memory addresses while a data bus carries from and to the CPU and memory – buses are made up of a group of wires. Inside the CPU multiple parts have different allocated tasks. For example, inside a CPU you would find a Program Counter. The job of this component is to hold the address of the command to be executed next (the address is found in the memory), this is to make the execution of tasks much faster as the CPU can fetch specific instructions in the correct order [3] based on the program counter. Another part of the CPU is an ALU (Arithmetic-Logic Unit), which is in charge of completing mathematical operations using adders (to do addition and subtraction) and transistors “that can convey signals in 0s and 1s” [4]. The control unit inside the CPU communicates with the ALU to tell it what operation needs to be completed, meanwhile, the register stores the necessary data that will need to be processed by the ALU. Once the ALU receives the data from the register and the control unit it will process the calculation and store the output. The ALU also utilises flip-flops (more specifically D type flip-flops) to “hold the status flags, temporary input values and information on the operation” [5]. Flip-flops are a method of storing binary data as they can only hold 1 bit of data (that is either 1 or 0). As an example, inside an ALU 4 flip-flops are used for status flags; overflow, negative result, zero result and carry 1 [5]. Figure 1 displays a logic gate circuit behind a ‘D Type Flip-flop’ and below is Figure 2 which displays the truth table for the flip-flop.



1. D Type Flip-flop [6]



1. D Type Flip-flop truth table [6]

# Methodology

The prototype utilises an Arduino Uno microcontroller for this system to execute all the functions and calculations that are required for the system to work as intended. Arduino is an open-source electronics platform [7] which allows users to make use of a combination of sensors and devices to create and develop a circuit similar to ones used in industry. For the Arduino board to have functionality, the user must develop a program using the Arduino IDE (integrated development environment – software for developing code) to be able to interact with the components of the circuit. This IDE utilizes a programming language which is a variant of C++.

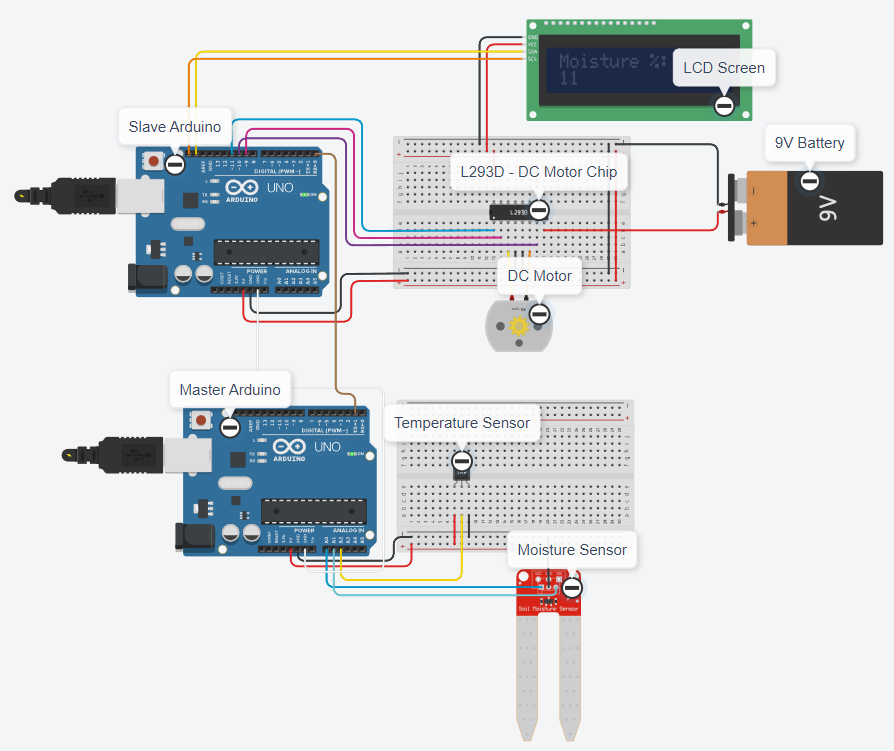
For components to communicate with the Arduino, and vice versa, the components need to be connected to the board via the DGI/O (Digital Input / Output) or analog pins – in this prototype both types of pins are utilised. The soil moisture sensor and temperature sensor are connected to analog pins because the board’s analog pins accept values from 0 – 1023 which can provide an accurate reading of the variable resistor inside the sensors to the microcontroller, whereas the DGI/O pins can only accept values 0 and 1 (also represented as Low and High). The driver chip for the DC motor connects 2 input pins and an enabler to the DGI/O pins on the Arduino board. The chip has an enabler pin on both sides of the chip, in the prototype only 1 enabler pin is used as only half of the chip is being used – if the enabler pin receives a high signal, then all the pins on the corresponding half of the chip will be interactable. The 2 input pins on the driver chip are in charge of receiving a high or low signal from the microcontroller to then direct the signal to output pins in the chip which connect to the 2 terminals on the DC motor. The rotation of the DC motor is decided by which output signal is high.

This smart watering system will provide the user with the necessary features to maintain efficient crop yields while staying the most cost-efficient as possible to increase the profitability of the user. The proposed water system can utilize industry-used moisture sensors and temperature sensors to collect data that will be manipulated by the master microcontroller in ways such as; sending a message to the slave microcontroller to activate a sprinkler if the value from the moisture sensor drops below a specified threshold, displaying moisture and temperature values on an LCD (liquid crystal display) screen for the consumer to see accurate measurements of the crop’s conditions. These measurements will be collected at regular set intervals (for example every 20 minutes) as this will prevent the microcontrollers from overflowing with too much data and causing delays in the system. Also, this will allow for scalability for the system itself as the consumer will be allowed to connect more data collection nodes to cover more area without causing issues for the existing system.

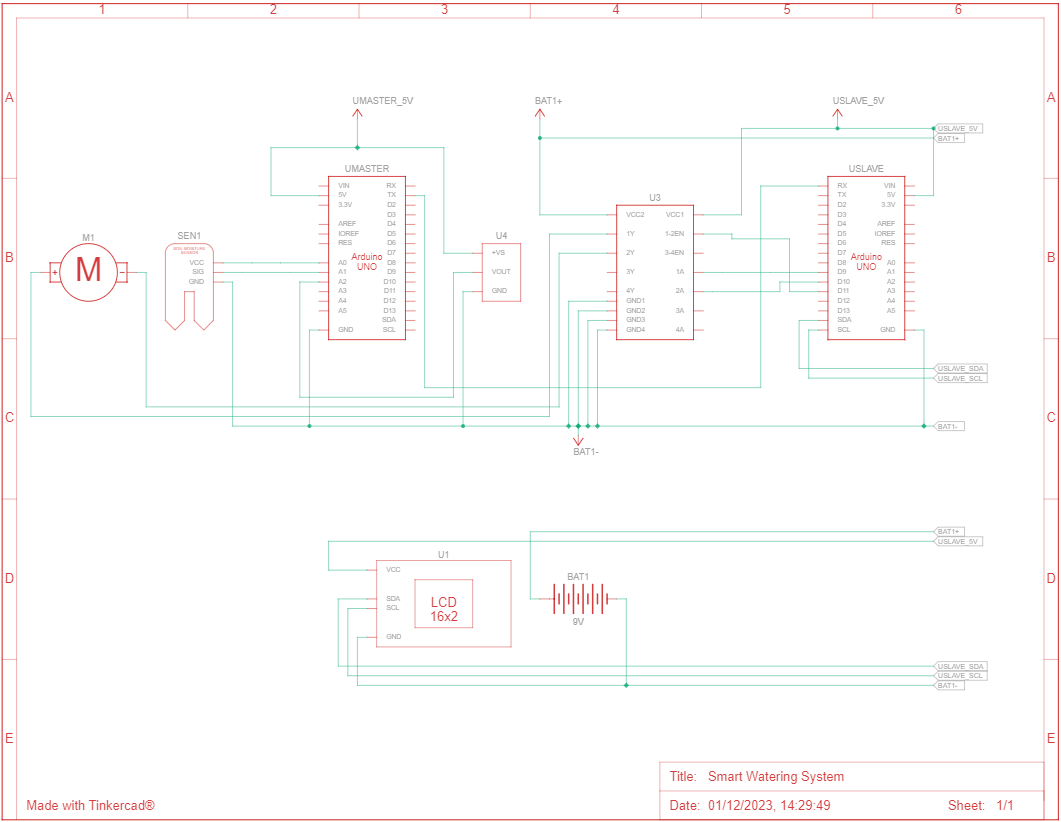
The communication protocol that is used between the microcontrollers is the I2C (Inter-Integrated Circuit) protocol, the reason why this is the chosen protocol for this prototype is because the amount of data being transferred is small which will not have a major impact on the processing speeds and will not disrupt the communication of the two controllers. Also, with future scalability in mind, this communication protocol will be a valid choice as the same methodology can be used when upgrading to wireless by keeping multiple slave and master Arduinos functioning the same way at the cost of a potentially slower data transfer. However, if the product was to include wireless communication, then there would have to be extra security measures put into place as the data transmission would be exposed to noise that could disrupt the message or other potential ways of data being captured.

Figure 3 below, is a circuit depicting how each component of the system will communicate with each other to demonstrate a prototype of the final product. Alongside it is the schematic view of the prototype circuit. The circuit consists of:

* 2x Arduino Uno
* 1x LCD screen
* 1x 9v Battery
* 1x DC Motor
* 1x L239D Chip
* 1x Soil Moisture Sensor
* 1x Temperature Sensor



1. Tinkercad Prototype Circuit



1. Schematic View of Tinkercad Circuit

## Implementation of a Moisture Sensor

Moisture Sensors use dielectric-permittivity to calculate the volumetric content of water in soil [8] to change the resistance value of the sensor, this resistance value is then constantly communicated to the connected microcontroller. The resistance of the Moisture Sensor decreases as more water is present in the soil and increases when less water is present. These sensors will be connected in groups of other moisture sensors and transmit all their data towards their corresponding master node. This connection can be changed from wired to wireless as running cables through large plots will end up harming the environment itself because cables could break down due to the conditions outside resulting in cable replacements or possibly endangering the crops.

## Implementation of a Temperature Sensor

Temperature sensors are working alongside moisture sensors. These sensors will be feeding information towards the master node at similar intervals to the moisture sensors and the data collected from these sensors will be displayed to the user on an LCD screen. When placed in plots of land, these temperature sensors will be connected near the moisture sensors (not in the soil itself) to gather more accurate data on crop conditions.

## Implementation of an LCD Screen

LCD Screens will display data gathered from the moisture and temperature sensors to the user, the information that is to be displayed will be sent from the master Arduino directly to the slave Arduino, which will then display the data on the LCD screen. The data on the LCD screen updates every time the slave Arduino receives data from the master Arduino. This LCD screen will display moisture and temperature measurements simultaneously of each group of sensors and allow the user to cycle through different groups to see their conditions. Also, the user will have a message displayed to them when crops will soon be watered.

Using the data collected from both moisture sensors and temperature sensors, the microcontrollers will be able to control an irrigation system (for example, a network of sprinklers) to determine when water should be spraying crops to keep them at their most optimal growth conditions. The microcontroller in charge of the irrigation system will receive a command when to water plants and when to stop from the microcontroller handling the sensors’ data. The threshold for activation of sprinklers will have a default value set in the slave Arduino and when the moisture reaches that specified threshold the irrigation system that the controller is connected to will activate (in Figure 1, the DC motor is in charge of the irrigation system to run when crops will need to be watered, and stop when watering needs to stop) and water the crops for a fixed amount of time - 20 / 30 minutes. An alternative solution to the irrigation system is instead of watering the crops for a fixed amount of time, the crops could be watered until a specific amount of moisture is in the soil, for example watering starts when moisture reaches 40% and finishes when moisture reaches 80%.

However, there are multiple aspects when it comes to farming that will need to be accounted for to make this product more efficient and adaptive to different farmer’s needs, such as different soil types and different crops. Through this, the product will not be limited to only a specific group of farmers, but rather inclusive to all which will bring in more customers for the company. To make this product more flexible, there will be different modes developed that the user will be able to cycle through using buttons, for example, vegetation mode would provide a moisture range of 21% - 60% meanwhile trees/shrub mode would have a moisture range of 21% - 40%. These moisture ranges have been set to these specific numbers as research shows that “the optimal range of soil moisture for crops depends on the specified plant species, but the range for most crops is between 20% - 60%” [9]. The user will also be able to have a custom mode where they will be able to set the moisture range themselves, this is to comply with different soil conditions and different crop types.

The current prototype uses a 9V battery as a way to power the DC motor, however, a more suitable option would be to implement a solar panel into the product to power the circuit compared to a battery as the farmer would have to replace the battery every time it runs out. Another issue with using a battery is that there is no way of knowing if a battery needs to be replaced, or when it will need to be replaced until the system fully stops working. This battery could also cause problems to the soil and crops themselves as the battery could potentially leak or wear down and harm the soil as a result of it. This solar panel could also connect to a rechargeable battery to store excess power that it generates, this power could then be utilised at night as the solar panel will not be able to generate power. In Figure 3 you can see a logic circuit diagram of the prototype with additional features that have been mentioned, also a truth table of this circuit can be found on the last page of the article.

A black line drawing of a diagram

Description automatically generated

1. Logic Gate Diagram of system activation

The Boolean expression for this circuit:

Key:

* A = Rechargeable Battery Power
* B = Solar Power
* C = Low Moisture
* D = Vegetation Mode
* E = Shrub / Tree Mode
* Q = Activate Irrigation System

# Conclusion

The current prototype of a smart watering system has plenty of possibilities to be improved to make it more efficient at its tasks and be more prepared for the conditions that it will be used under. These improvements will also make the product more beneficial towards its consumers, as more features will be introduced, the adaptability of the product will rise making it a valuable asset to more consumers. Also, these improvements will benefit the business as more customers will be interested in using this product as it reduces the amount of tasks necessary to farm crops, while also decreasing the amount of wasted resources and money used on maintain farms while upkeeping quality. This product will also give the customers easier control over their farmland as they will be able to set what conditions have to be met for the plants to be watered, alongside receiving live data of the conditions the plants are in – which without the use of technology, such as soil moisture sensors, would have took multiple minutes, if not hours based on the size of farms, to collect the same data that is collected within seconds. Additionally, there will have to be security measures put into place to ensure that wireless transmission will be working as intended while reducing the risks of the communication failing or ultimately being stopped due to cyber attacks. The proposed product itself has clear goals that it needs to achieve, that being UN Sustainability Goals, and it will help achieve these goals with the proposed improvements to the current prototype.

##### References

[1] - Howarth, J. (2021). *80+ Amazing IoT Statistics (2022-2030)*. [online] Exploding Topics. Available at: https://explodingtopics.com/blog/iot-stats [Accessed 1 Dec. 2023].

[2] - www.schroders.com. (n.d.). *Why agriculture holds the key to tackling water waste*. [online] Available at: https://www.schroders.com/en-us/us/individual/insights/why-agriculture-holds-key-to-tackling-water-waste [Accessed 1 Dec. 2023].

[3] - lenovo. (n.d.). *What is a Program Counter? How Does it Work? | Lenovo US*. [online] Available at: https://www.lenovo.com/us/en/glossary/program-counter/?orgRef=https%253A%252F%252Fwww.google.com%252F [Accessed 1 Dec. 2023].

[4] - Spiceworks. (n.d.). *What Is an Arithmetic Logic Unit (ALU)? 7 Key Components*. [online] Available at: https://www.spiceworks.com/tech/hardware/articles/what-is-alu [Accessed 1 Dec. 2023].

‌[5] - Anon, (n.d.). *How flip-flops are implemented in the Intel 8086 processor*. [online] Available at: https://www.righto.com/2023/09/8086-flip-flops.html [Accessed 1 Dec. 2023].

‌[6] - Basic Electronics Tutorials. (2018). *D-type Flip Flop Counter or Delay Flip-flop*. [online] Available at: https://www.electronics-tutorials.ws/sequential/seq\_4.html [Accessed 1 Dec. 2023].

[7] - Arduino (2023). *What is Arduino? | Arduino Documentation*. [online] docs.arduino.cc. Available at: https://docs.arduino.cc/learn/starting-guide/whats-arduino [Accessed 1 Dec. 2023].

[8] - Nagothu, S.K., 2016, February. Weather based smart watering system using soil sensor and GSM. In *2016 world conference on futuristic trends in research and innovation for social welfare (startup conclave)* (pp. 1-3). IEEE.

[9] - eos. (2022). *Soil Moisture: Methods Of Measuring & Tools For Monitoring*. [online] Available at: https://eos.com/blog/soil-moisture [Accessed 1 Dec. 2023].

‌

Truth Table for Figure 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | Q |
| 0 | 0 | 0 | 0 | 0 | **0** |
| 0 | 0 | 0 | 0 | 1 | **0** |
| 0 | 0 | 0 | 1 | 0 | **0** |
| 0 | 0 | 0 | 1 | 1 | **0** |
| 0 | 0 | 1 | 0 | 0 | **0** |
| 0 | 0 | 1 | 0 | 1 | **0** |
| 0 | 0 | 1 | 1 | 0 | **0** |
| 0 | 0 | 1 | 1 | 1 | **0** |
| 0 | 1 | 0 | 0 | 0 | **0** |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | **0** |
| 0 | 1 | 1 | 0 | 0 | **0** |
| 0 | 1 | 1 | 0 | 1 | **1** |
| 0 | 1 | 1 | 1 | 0 | **1** |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | **0** |
| 1 | 0 | 0 | 0 | 1 | **0** |
| 1 | 0 | 0 | 1 | 0 | **0** |
| 1 | 0 | 0 | 1 | 1 | **0** |
| 1 | 0 | 1 | 0 | 0 | **0** |
| 1 | 0 | 1 | 0 | 1 | **1** |
| 1 | 0 | 1 | 1 | 0 | **1** |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | **0** |
| 1 | 1 | 0 | 0 | 1 | **0** |
| 1 | 1 | 0 | 1 | 0 | **0** |
| 1 | 1 | 0 | 1 | 1 | **0** |
| 1 | 1 | 1 | 0 | 0 | **0** |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 |