element14
/ AN AVNET COMMUNITY

April 2023

INTRODUCTION TO SMART AGRICULTURE

Table of Contents

04 CHAPTER - 1 Introduction

04 CHAPTER - 2 Using Arduino for Control

06 CHAPTER - 3 Wireless Communications for IoT Applications in Smart Agriculture

08 CHAPTER - 4 Arduino IoT Cloud Platform

11 CHAPTER - 5 Powering your Smart Agriculture Project

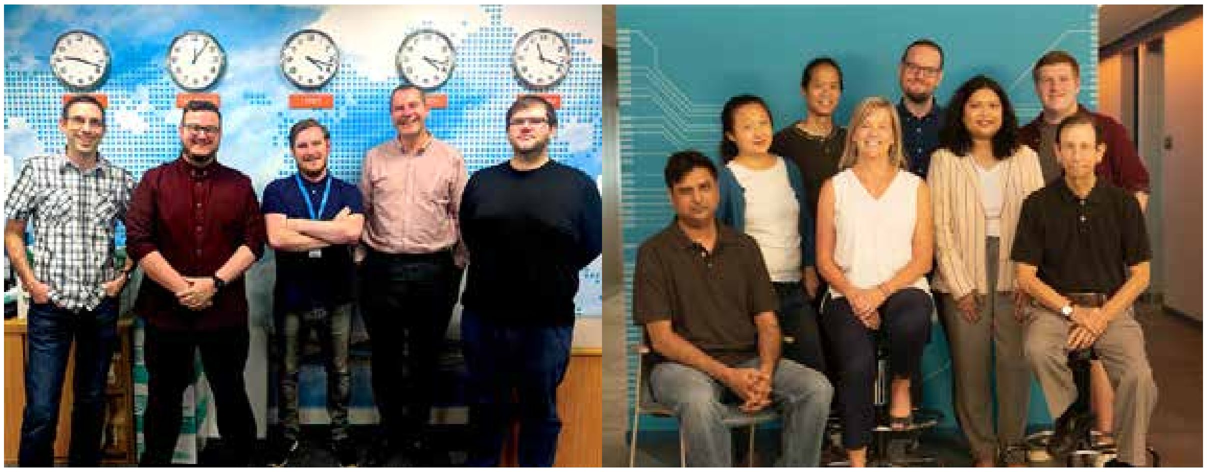
12 CHAPTER - 6 Conclusion

Introduction to Smart Agriculture

element14 is a Community of over 800,000 makers, professional engineers, electronics enthusiasts, and everyone in between. Since our beginnings in 2009, we have provided a place to discuss electronics, get help with your designs and projects, show off your skills by building a new prototype, and much more. We also offer online learning courses such as our Essentials series, video tutorials from element14 Presents, and electronics competitions with our Design Challenges.

With the population of the world increasing every year, efficiency and sustainability in farming is becoming more and more important. All of the world's industries are working to make the shift to Green Technology. Smart agriculture leverages modern advancements in technology to reduce costs and increase productivity. Arduino boards are versatile devices that can be used to communicate with and control the sensors and machinery that are used in farming. This eBook introduces you to the Arduino family of products, including the Arduino Cloud platform, and how they can be used to build applications for smart agriculture.

element14 Community Team



CHAPTER - 1 Introduction

Technology has been playing an important role in agriculture for many years now. It has given farmers the tools necessary to help increase productivity, improve crop yields, and reduce costs. One of the more exciting technologies currently influencing agriculture are Arduino microcontrollers. These small, low cost devices can be programmed to control a wide variety of sensors, motors, and relays. For example, soil moisture levels, temperature, humidity, and light levels can all be measured using the right sensors with an Arduino. In addition, they can be used to control irrigation systems, monitor crop growth, and drive autonomous vehicles.

A closely related technology that is also becoming increasingly important in the sustainable transformation of agriculture is the Internet of Things (IoT). IoT generally describes a network of physical objects (the "Things") which include embedded sensors, software, and technology used to collect data. These devices also connect to the Internet and transfer the accumulated data to a central location, where it can be processed, monitored, and used to take action. IoT devices are already becoming quite common in our everyday lives. Some well-known examples includes smart thermostats, smart doorbells, and robot vacuum cleaners.

This eBook covers the use of Arduino and IoT in smart agriculture applications. We will begin with a closer look at how Arduino control boards are used to control sensors and motors. Furthermore, we will cover options to communicate with devices remotely. Once we have an understanding of what can be collected and how, we will look into how this information can be organized and interpreted to make informed decisions. As we move through some of the topics, we will cover some examples and use cases where appropriate. The complete list of topics covered is shown below:

- Using Arduino for controlling sensors, motors, and relays
- Options for connecting devices wirelessly for remote communication
- Options for monitoring information in an organized manner
- Powering your devices

CHAPTER - 2 Using Arduino for Control

An Arduino board is a great tool for controlling sensors and motors. It is an open source platform, inexpensive, easy-to-use, and has a large community for support and reference material. As such, there are many tutorials available for getting started with connecting and programming the device. Furthermore, a wide range of boards has also become available since the creation of the original Arduino Uno.

There are currently three families of Arduino boards, plus a pro series. The classic devices, as the name

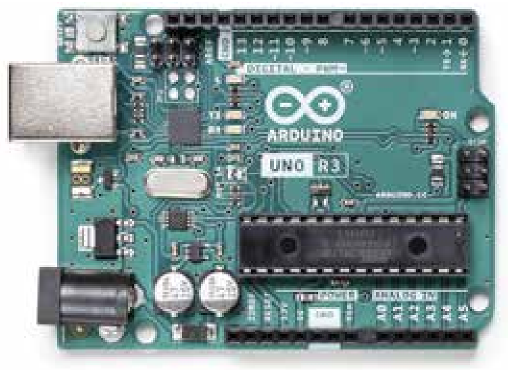


Figure 1. Arduino Uno

suggests, includes all the original devices, such as the Uno and Mega boards. These boards are great starting points for communicating with devices such as sensors and motors. They differ mainly in the amount of GPIO and peripheral interfaces available. The next available family is the Nano line. These boards come in a smaller form factor and vary on the features they include. Wi-Fi and BLE modules can be included on board for wireless connectivity, in addition to sensors such as humidity, temperature, pressure, and microphone. The last family of boards is the MKR family. The MKR series was designed to be combined with shields and carrier boards to create projects without the need for any additional circuitry. Every board features a Cortex-M0 32-bit SAMD21 low power processor and a radio module that enables Wi-Fi, Bluetooth, LoRa, and NB-IoT wireless communication.

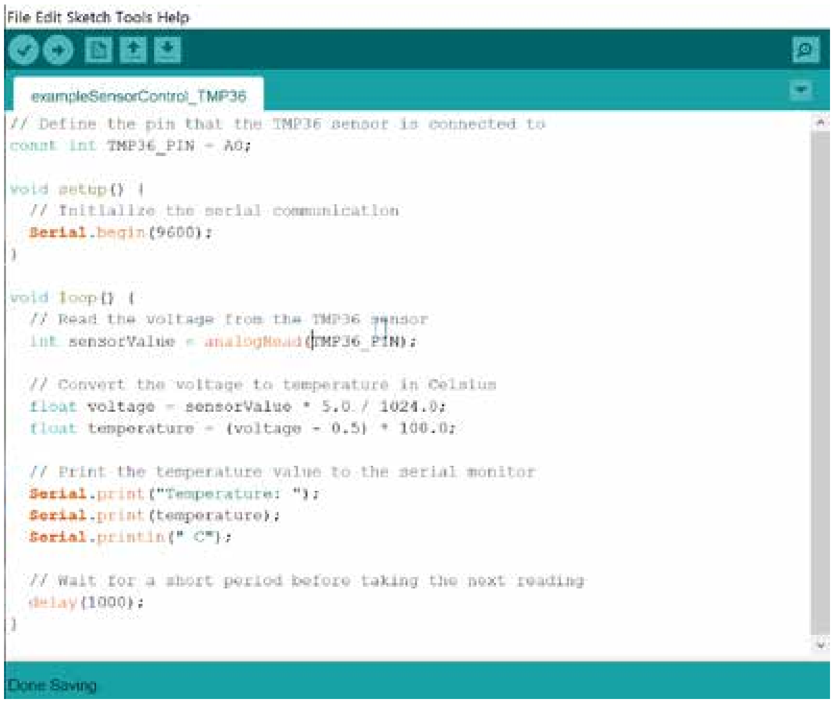
Once a specific board is chosen, getting it to communicate with various sensors or motors involves just a few steps. Gather the required components for the project; this includes the Arduino, sensors, jumper wires, power supplies, and breadboard or protoboard. The following steps outline the general workflow for creating working prototype projects.

1. Connect the sensors to the Arduino. This involves identifying the proper pins to use on the Arduino. For devices that require SPI or I2C for communication, the Arduino has dedicated pins for these functions. Specific pins are also required to read analog inputs, create pulse width modulated outputs (PWM), or use GPIO. In addition, it is important to make sure that the sensors and Arduino voltages are compliant with one another. Many sensors or integrated circuits operate with a 3.3V supply voltage, while some Arduino boards operate with 5V voltages. Applying 5V to a 3.3V device can cause permanent damage to the lower operating voltage device. To interface with these lower voltages, a variety of level shifting devices are available.

2. Write the code to control the sensors. In general, code is written in the Arduino IDE and uploaded to the microcontroller. While other options are possible, this is the most straightforward and common method.

3. Upload the program to the Arduino device and test the functionality. Uploading code to the Arduino is done through the IDE and involves the click of just a few buttons. After the code has been loaded onto the Arduino, it begins to run, allowing the user to observe or test the functionality of the prototype.

Below is an example program written to read in temperature data from a general purpose [TMP36 temperature sensor](https://www.analog.com/en/products/tmp36.html).



```
// Define the pin that the TMP36 sensor is connected to
const int TMP36_PIN = A0;

void setup() {
  // Initialize the serial communication
  Serial.begin(9600);
}

void loop() {
  // Read the voltage from the TMP36 sensor
  int sensorValue = analogRead(TMP36_PIN);

  // Convert the voltage to temperature in Celsius
  float voltage = sensorValue * 5.0 / 1024.0;
  float temperature = (voltage - 0.5) * 100.0;

  // Print the temperature value to the serial monitor
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" C");

  // Wait for a short period before taking the next reading
  delay(1000);
}
```

Figure 2. Example Arduino program for reading in data from a temperature sensor

The TMP36 is a low-cost and easy-to-use temperature sensor. The device has three pins, one of which gets connected to a supply voltage between 2.7V and 5.5V, another gets connected to ground, and the last outputs a voltage that is linearly proportional to Celsius temperature and can be read using an analog input pin on the Arduino. Without any external calibration, the sensor can typically provide an accuracy of +/-2° Celsius over a temperature range from -40°C to 125°C.

In Figure 2, the pin used for reading the sensor output is first declared. Next, the serial port is initialized in order to display information to the computer screen while testing. Inside the main loop, the voltage is read from the Arduino pin that is connected to the sensor output. This voltage is then converted to a temperature value and printed to the screen. The delay() command at the bottom of the loop tells the program to wait for one-second before proceeding, ensuring the output is readable and not changing too fast. Although this is a simple example program for a temperature sensor, it demonstrates the simplicity the Arduino offers in interfacing to devices. The general idea for connecting to different types of sensors is the same. First, read in the sensor data, convert it to a meaningful quantity, and then "do something" with that data. In the above case, we printed the temperature to a screen, however the data can be used to trigger a motor, turn on an LED, or upload the data point to a database for storing.

An Arduino board is well suited to working with electronics such as sensors and motors. It provides a simple platform for controlling and relaying information between devices, in addition to having a very large community for support and reference. Generally, initial prototypes are created using an Arduino board, along with jumper wires, breadboards, and sensor modules. Once the prototype is tested and proven to be working, the next step usually taken is to create a PCB design to hold all the electronic components and circuits. This allows for a cleaner overall solution that is custom tailored to one's specific needs. For instance, the Arduino circuitry can be combined onto a single board with a temperature sensor, humidity sensor, and photodiode for detecting light levels. This eliminates the need for any jumper wires, allows for a custom size and shape that is more optimal for specific applications, and creates a more professional final product.

CHAPTER - 3 Wireless Communications for IoT Applications in Smart Agriculture

After choosing the design and testing sensors or motor connections, the next step might be adding wireless communication to the device. This allows a network of remote devices to relay information back to a central hub for monitoring. For example, an array of soil moisture sensors can be used to monitor conditions throughout a field, allowing the user or system to target dryer areas for irrigation. Wireless connectivity is actually an integral feature in IoT solutions for agriculture applications. It may be the only usable communication option for monitoring the vast amounts of devices used in smart agriculture. On the other hand, wireless communication is also a convenient feature for the urban farmer or DIY enthusiast. With that said, there are a few different wireless technologies that can be used to bring remote connectivity to farming applications, whether large or small scale.

The first option we will cover is Long Range Radio (LoRa). LoRa was designed with IoT networks and machine-to-machine interaction in mind. It also allows long-range communication and extended battery life, making it an ideal choice for remote monitoring applications such as smart agriculture. LoRa is a proprietary technology developed by Semtech and communicates wirelessly in unlicensed spectrum bands from 137MHz to 1020MHz. It uses a spread spectrum modulation that operates with low data rates and can reliably communicate over a range of multiple kilometers in rural environments. In urban environments, this range is reduced to hundreds of meters. LoRa can also support multiple network topologies such as point-to-point, mesh, and star. For users who are just starting to work with wireless communications, the most important part may be its ease of use. Using devices that utilize the LoRa protocol requires minimal setup and configuration. It also features end-to-end encryption, which keeps sensitive data secure.

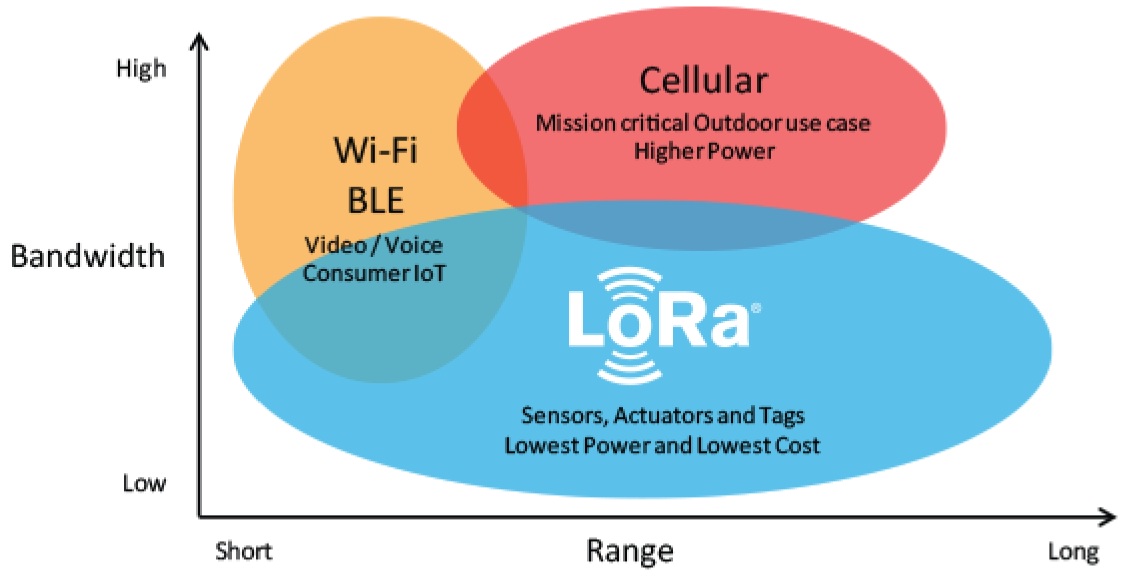

High
Bandwidth
Low
Wi-Fi
BLE
Video / Voice
Consumer IoT
Cellular
Mission critical Outdoor use case
Higher Power
LoRa®
Sensors, Actuators and Tags
Lowest Power and Lowest Cost
Short
Range
Long

Figure 3. Plot showing the bandwidth vs range of different wireless technologies (https://www.semtech.com/lora)

The second option available is SigFox. SigFox is a global wireless communications provider that offers a low power wide area network (LPWAN) for IoT devices. Similar to LoRa, it allows for long-range communication and enables long battery life through low power operation. It also operates wirelessly in the unlicensed spectrum, utilizing a narrowband modulation. Each message is 100Hz wide and transfers data at 100 or 600 bits per second depending on the region of operation. SigFox uses a star network architecture where devices are not physically attached to a specific base station. Instead, the broadcasted communication can be received by any base station within range, of which there may be three or more on average. These base stations are then used to transmit data to the cloud, where the data is processed and managed. Like LoRa, SigFox also offers end-to-end encryption for secure wireless communication links.

A third available option is known as the Narrowband Internet of Things (NB-IoT). Like the two aforementioned technologies, NB-IoT is designed for IoT wireless communication and to meet the requirements of low power, extended battery life, and long-range coverage. However, unlike the other technologies discussed, it operates in the spectrum of licensed cellular networks. This can be within existing GSM carrier links, unused guard bands between LTE channels, or independently. NB-IoT boosts communication range by using narrowband communication, utilizing transmission repetitions, and deploying different transmission bandwidths to improve efficiency. Since the technology utilizes cellular networks, it does have the advantage of being able to leverage existing cellular infrastructure, for example, cell towers and base stations. To gain access to the networks, the devices typically require a SIM card, which allows the network to authenticate its connection. In addition, some sort of paid subscription plan is required for using the network's resources; however, with that investment also comes high levels of security. For NB-IoT, advanced encryption and authentication mechanisms are used to keep data transmitted over the network secure.

It is also worth mentioning Wi-Fi and Bluetooth here. While they are not long-range options for wireless communications on their own, there have been promising developments for use in smart agriculture applications. Bluetooth mesh IoT is a mesh networking system that, along with the newer BLE technology, offers a low power method for many devices to connect with one another. As a result, one can find that a large network of Bluetooth devices can offer an option of coverage for a large area. This also provides an ideal method of wireless communication for urban farming applications.

One example application using LoRa technology is a **smart irrigation system**. A smart irrigation system uses soil moisture sensors and a valve controller to improve agricultural efficiency. In addition, light sensors, carbon dioxide sensors, and humidity sensors were included for additional data analytics capabilities. The overall goal of the smart irrigation system is to control the opening and closing of a water valve based on the soil moisture levels. As a result, water consumption and labor costs can be reduced and crop irrigation efficiency can improve.

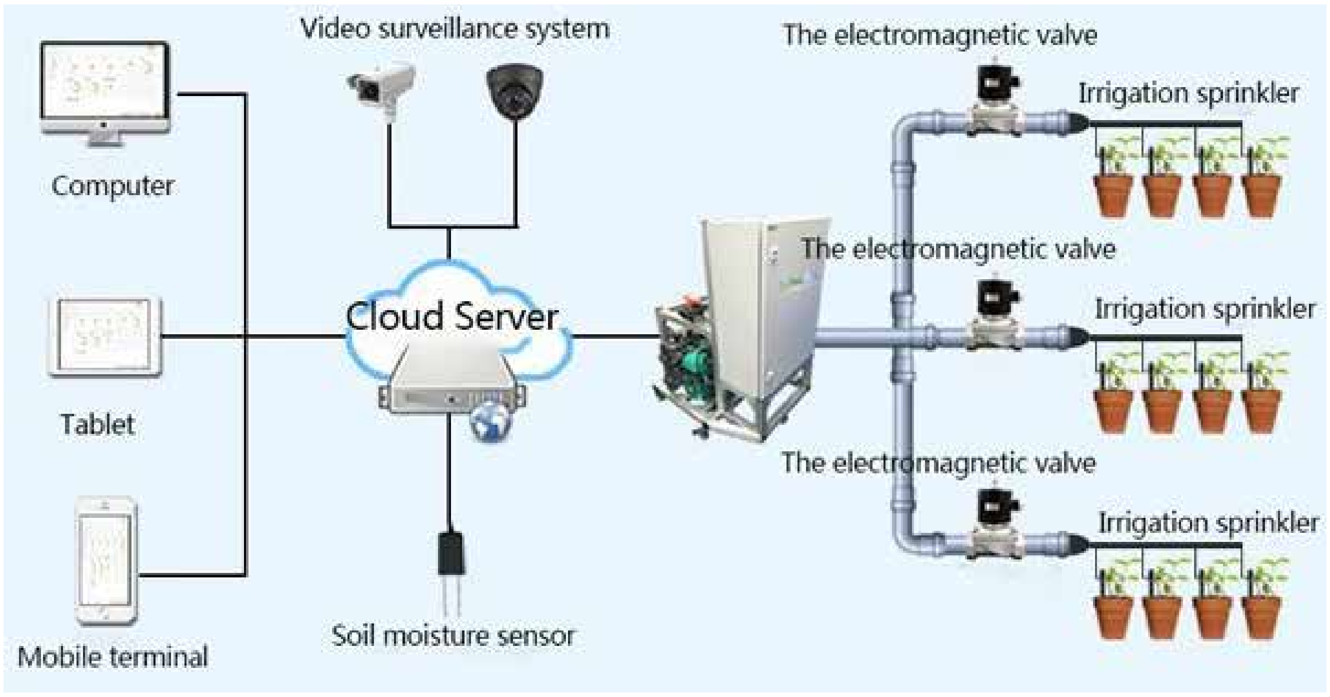


Figure 4. Example diagram showing the architecture of a smart irrigation system (https://www.renkeer.com/soil-moisture-sensor-for-irrigation/)

CHAPTER - 4 Arduino IoT Cloud Platform

The Arduino IoT cloud platform is a cloud-based service designed for managing and monitoring IoT devices. It allows developers to connect their Arduino powered projects to the Internet, collect data, build a visual dashboard, and monitor devices remotely. The platform includes a range of tools for building and deploying IoT devices, such as an online IDE, device manager, data visualization tool, and an app builder. Like other Arduino products, a key benefit to using the platform is its ease of use. Minimal knowledge of networking and cloud computing is required, and developing applications is intuitive and simple. Key features include data monitoring, variable synchronization across devices, a jobs scheduler, over-the-air updates, Amazon Alexa support, and dashboard sharing.

To begin using the Arduino IoT cloud, a supported board must first be chosen. This will depend on the application needs, as well as the type of wireless communication that one wishes to use. For example, if we are working on a project that uses LoRa for wireless communication, there are two supported boards available from Arduino. These are the MKR WAN 1300 and the MKR WAN 1310.

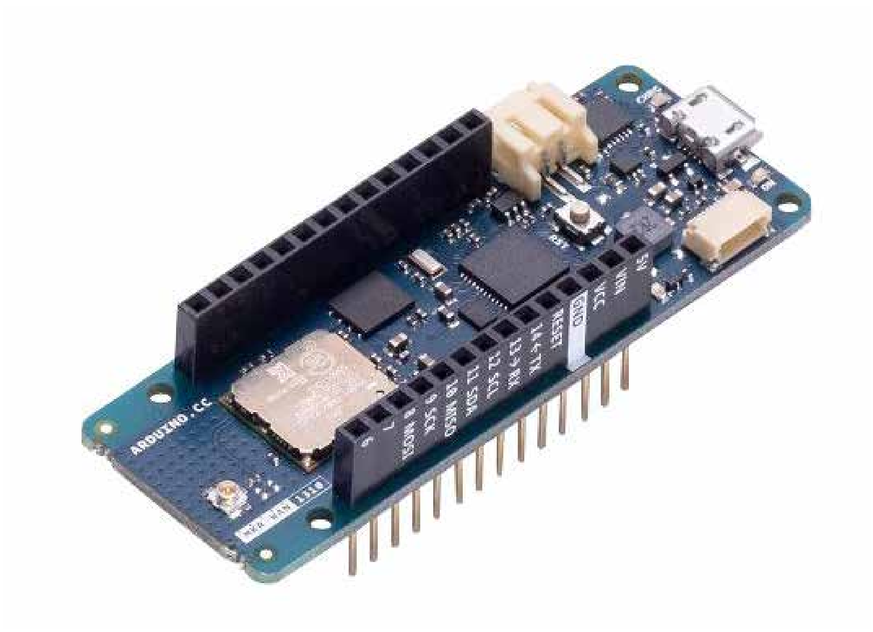
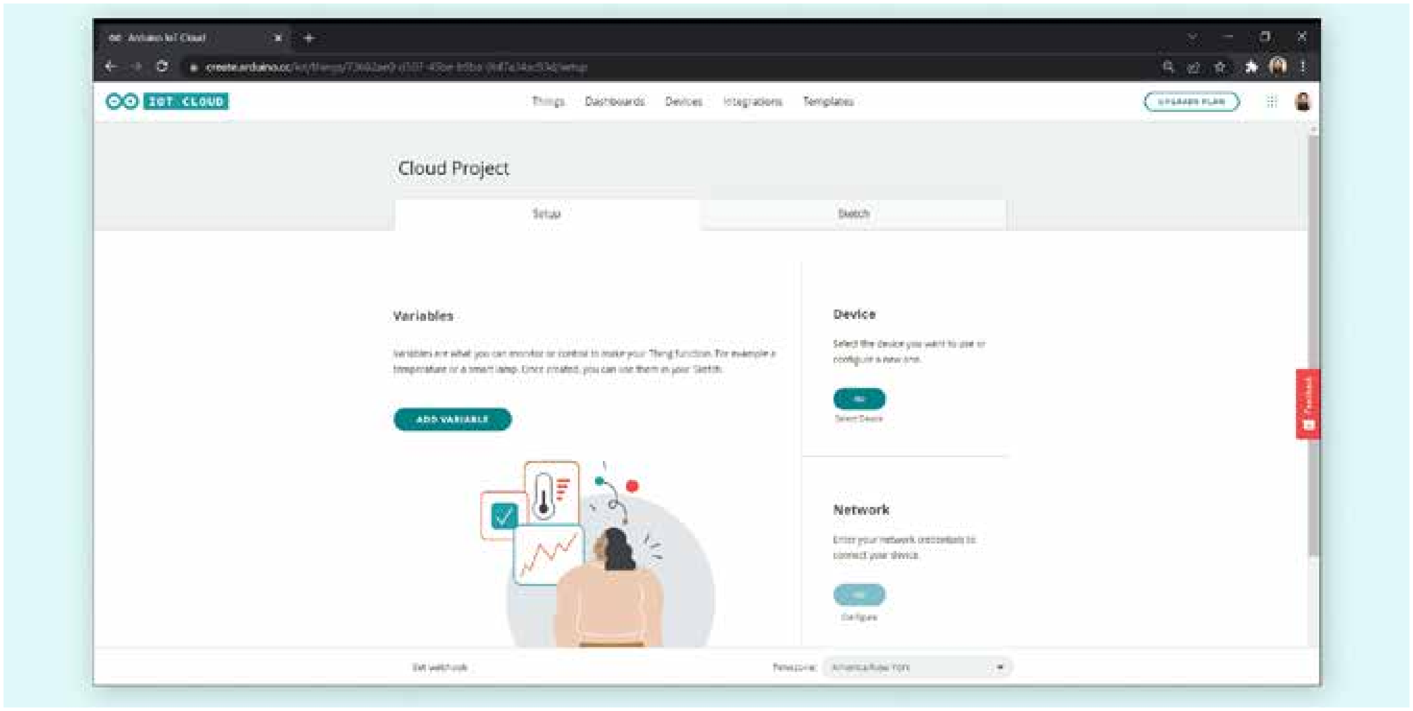


Figure 5. **Arduino MKR WAN 1310**

The next step involves creating an account on the Arduino IoT Cloud website. After gaining access to the website, we can begin a new project by creating a “Thing”. Upon clicking on the “Things“ tab at the top of the screen, we are presented with a page that will allow us to choose a device, connect to a network, and create new variables. The variables created will automatically generate in a sketch that we can use later when writing code for the device. All the expected variable types are available such as ***int*** and ***float*** types; however, there are also special variable types available such as ***temperature*** and ***luminance***. There is a large amount of additional specialized variables available for use. They can be used as normal variables, but provide specialized wrappers that make working with third party devices and dashboards more fluent.



Cloud Project

Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

Device

Select the device you want to use or configure a new one.

Network

Enter your network credentials to connect your device.

Set webhook

Timezone: America/New York

Figure 6. Arduino IoT “Thing” page for setting up devices in a network (arduino.cc)

Devices are added and managed using the "Devices" tab at the top of the screen. In addition, connecting a Wi-Fi network can be done through the "Network" section of the "Setup" page (shown in Figure 6). After configuring the project, we can begin writing code for the devices. The IDE and sketches work in a similar fashion to the original Arduino IDE. When opening up a sketch there will be some additional code included for connecting to the network and cloud. The web IDE also allows users to verify and upload code to their devices. One of the convenient features of the IoT platform is the ability to wirelessly update devices. There is no need to manually connect to and program each device; code can be pushed onto the devices through the web application.

The last step in getting a complete platform running is to create a dashboard. The dashboard can be created through the "Dashboards" tab at the top center of the screen; this brings you to a page that allows you to see and edit existing dashboards, as well as create new ones. The dashboards are generally created using widgets, which are linked to variables that have been created and used. It is as simple as choosing a widget, selecting a Thing, and choosing the variable to link with the widget. Upon completion, the widget will automatically start updating with real time information, assuming the IoT device is powered on and wirelessly connected. Figure 7 shows an example dashboard with various available widgets.

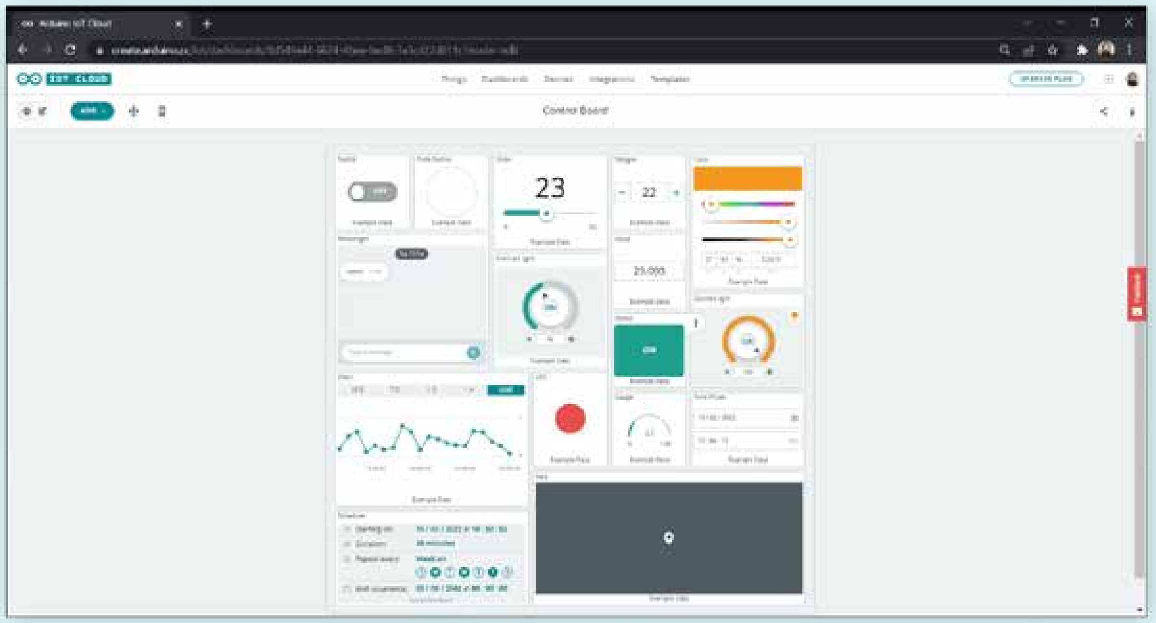


Figure 7. Arduino IoT Dashboard (arduino.cc)

One example use case of the Arduino IoT Cloud is monitoring sensor data from various boards in the Arduino MKR family. These include the MKR Wi-Fi 1010 and MKR ENV Shield. The shield integrates sensors such as temperature, humidity, pressure, and illuminance. Following the steps above, the board must first be configured as a new device and a Thing must be created. Next, new variables are created which correspond to the data that will be collected (i.e. temperature) and a network connection must be made. After writing code that allows the sensor values to be read, the dashboard can be constructed. Figure 8 shows the dashboard created to display the data from various sensors.

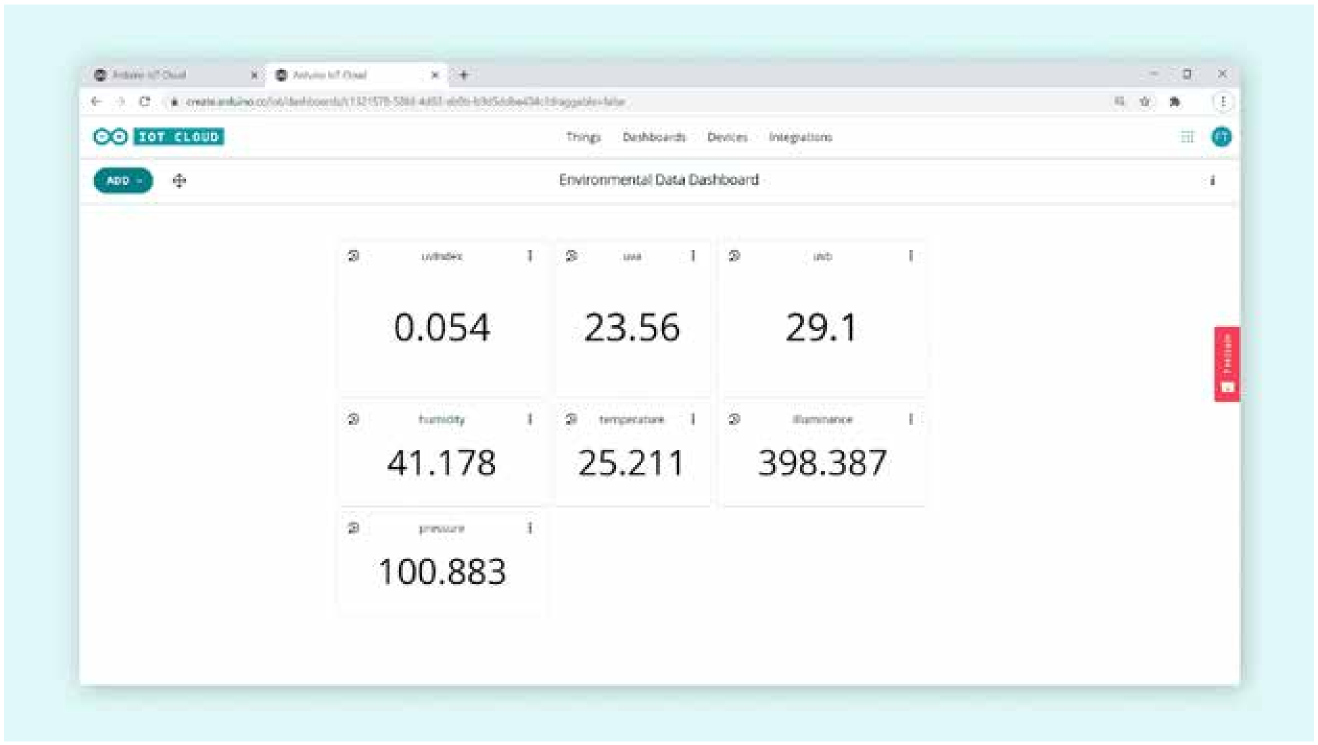


Figure 8. Example Arduino IoT dashboard displaying sensor data from MKR shield board (arduino.cc)

CHAPTER - 5 Powering your Smart Agriculture Project

Powering your smart devices is an essential part of building a successful IoT network. In urban agriculture environments, a simple solution such as power from a wall outlet or battery may suffice. In larger agricultural environments, however, many devices can be located in remote areas where running power to them or replacing batteries may be impractical. The first step in powering the design comes from choosing the right components. In these types of environments, choosing devices with low power in mind is critical; saving even a few microwatts can make the difference in a battery lasting either a year or five years. Furthermore, the wireless communication used should be optimized for the lowest possible power consumption. Most of the wireless communication protocols previously discussed already make efficient use of low power communications. However, the most critical power consumption applications require care when choosing the best communication scheme.

Regardless, IoT devices will need some sort of battery to hold charge, providing power when needed. Battery choices may be constrained by the size of the device or available budget. Generally, smaller, inexpensive batteries offer less power capacity. The three most widely available battery options are alkaline, lithium-ion, and lithium-polymer batteries. Alkaline is the most common, but also the poorest choice due to their output voltage degrading relatively quickly and their limited temperature range of operation. Lithium-ion batteries are the next best option and offer a considerably better voltage discharge characteristic curve. Lastly, lithium-polymer batteries offer wide temperature ranges, high energy densities, and low self-discharge rates. Nickel-metal hydride (NiMH) is an additional battery technology that offers a middle ground between alkaline and lithium. The plot below compares the different battery technologies voltage output over time under a continuous discharge of 1W.

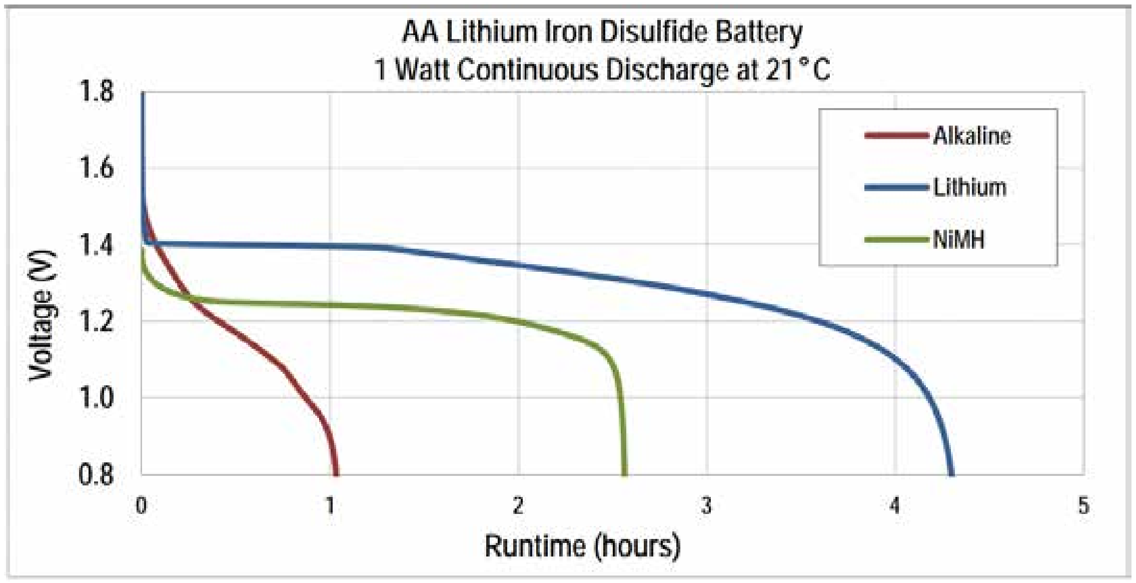


Figure 9. Comparison of different battery technologies (https://makermax.ca/articles/alkaline-vs-lithium-ion/)

For devices in remote locations, it may make sense to add some type of energy harvesting to the device to provide additional charge to the battery when possible. The most common of these approaches, especially in agriculture applications, is through solar energy. A variety of different solar panels currently exist, with some designed specifically for IoT applications. Some of these include urethane panels, ETFE panels, and glass panels. They differ in cost, size, weight, and longevity. Modern solar panel technology has vastly improved since first generation panels became available. Today it is possible to find solar panels with efficiencies approaching 40%, whereas in the past they were about 10%.

Although not practical for many agricultural environments, additional energy-harvesting methods include vibration energy harvesting, fluid flow energy harvesting, and direct energy harvesting where energy is converted from a wireless signal to electrical power.

CHAPTER - 6 Conclusion

We have covered some of the basics of how an Arduino board is used to communicate with sensors and how this information can be transmitted wirelessly and displayed in an organized and professional interface. The Arduino IoT Cloud serves as a versatile platform for users to begin building their IoT solutions. Additionally, we reviewed various wireless communication protocols associated with IoT technology. Finally, a brief review of technology to power remote devices was covered.

In conclusion, Arduino and IoT are already influencing agriculture. The technology allows for increased efficiency and productivity while reducing costs. As the industry moves towards Green Technology, farmers will be able to optimize their operations and increase their yields, making it possible to feed the growing population of the world while also reducing the environmental impact of agriculture.

Find out more about Arduino here.

element14 / AN AVNET COMMUNITY

300 S. Riverside Plaza, Suite 2200
Chicago, IL 60606

https://community.element14.com/

Facebook.com/e14Community
Twitter.com/e14Community