Internet of Things (IoT)

Intelligent and Digital Manufacturing

Derek Badman

Patrick Maynard

Linda George

Noor Pola

2024

Contents

Executive Summary 2

1. Introduction 2

2. What is Internet of Things (IoT) 2

3. Experiment 2

3.1. Premise 2

3.2. Breakdown 2

3.2.1. Physical Setup 2

3.2.2. Flashing and Verification 4

3.2.3. Troubleshooting 4

3.3. Outcome 5

4. Our Findings/Experience 6

5. Applications in the industry 8

6. Security Concerns and Implementation 8

7. Next Steps for IoT 9

7.1 Security and Regulation 9

7.2 Trends in Technology 9

7.3 Market Growth 9

8. Conclusion 10

References 11

Appendix 13

1. Arduino MKRWfFi1010V2.0 Schematic 13

2. Raw Experiment Data 14

3. Arduino Boilerplate Code 16

4. GitHub Repository 19

# Executive Summary

# Introduction

# Objectives

The objective of this report is to explore the internet of things as used in todays industry. We will attempt to do this by conducting a small scale expeirment to show the basic functionality of an IoT device and then expand upon that with a look at how it is roled out on a larger scale.

# What is Internet of Things (IoT)

# Experiment

## Premise

To get a better working understanding of the Internet of Things concept we felt it best to setup an IoT project from start to finish on a small scale. We found a project that would monitor temperature and humidity of an area using a small, embedded board and a multi-function sensor. Then using the data gathered from the device populate a web-based dashboard provided by Arduino Cloud [https://app.arduino.cc/].

## Breakdown

## 3.2.1. Physical Setup

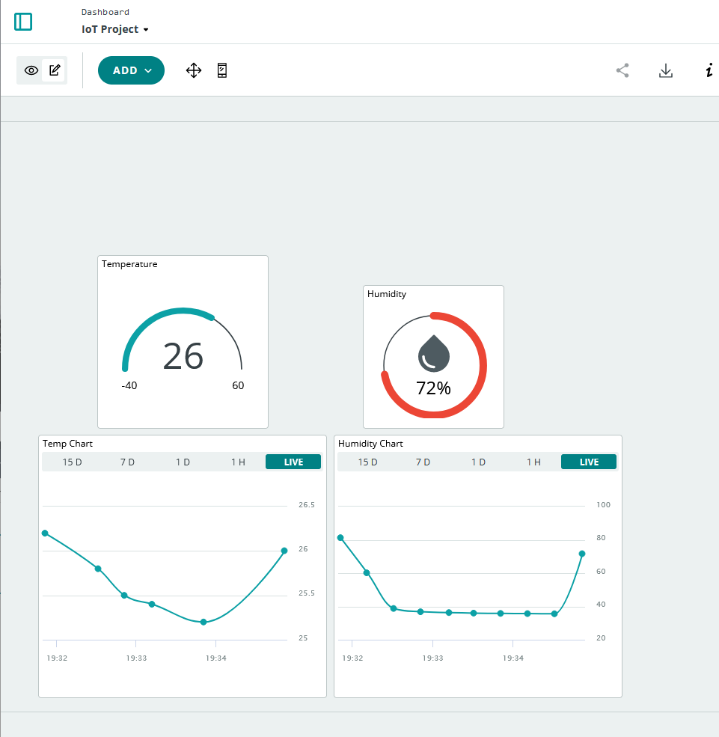
We began with setting up and wiring our breadboard with our Arduino MKR 1010 Wi-Fi board and a DHT multi-function sensor (Fig. 1). After the board, now named Zaria, had come online it came time to flash it with the boilerplate code and wi-fi credentials to enable it to communicate via wireless on its own.

A close-up of a machine

Description automatically generated

Fig 1: MRK Board and DHT Sensor Setup

## 3.2.2. Flashing and Verification

A screenshot of a computer

Description automatically generatedOnce Zaria was online uploading the code was the next step (Fig. 2). Using the Arduino Cloud (<https://app.arduino.cc>) we flashed the code and set up a dashboard to monitor the output of the sensor (Fig. 3).

Fig 2: Code Upload

Fig 3: Live Dashboard

## 3.2.3. Troubleshooting

After getting out initial feedback from our sensor we noticed that it was reading incredibly high. After re-wiring and re-flashing, the conclusion was that the sensor itself was intermittent. After replacing our DHT11(Fig. 4) with a DHT22 (Fig. 5) sensor the project was back on track.

A close up of a device

Description automatically generated

Fig 5: DHT 22 Sensor

Fig 4: DHT 11 Sensor

## Outcome

The final product of our small scale IoT experiment was a fully functional temperature and humidity monitor for a home server room. Thanks to Arduino Cloud we were able to make a multi-platform IoT dashboard (Fig. 6 and Fig. 7) using minimal hardware.

# A screenshot of a device Description automatically generatedOur Findings/Experience

Fig 7: Desktop Dashboard

Fig 6: Android Dashboard

The most notable lesson learned from this experiment has been, the more companies make IoT devices modular and accessible the more everyday users can innovate with them. With just a few simple boards and a web browser, we were able to set up a basic IoT monitoring system. After the system had run for a few days, we were easily able to export some data from it to present (Fig. 8). If we wanted to take it further, the boilerplate code could be updated to send the data directly to our database and polled from there for any number of custom tasks. The IoT possibilities are endless. This small-scale experiment is a single use example of what we would see in a factory monitoring system. But instead of one board and one sensor, we would see hundreds or thousands of sensors fed back to a controller and used to populate several dashboards and end-user tools (Fig. 7).

A hand holding a tablet with a graph on it

Description automatically generated

Fig. 7

A person holding a clipboard

Description automatically generated

Fig 8: Data Export

# Applications in the industry

# Security Concerns and implementation

# Next Steps for IoT

Within the next five to ten years, the Internet of Things is expected to undergo rapid growth to support an increased user interface. To succeed in developing a technology that will continue to make societal impacts and leave a lasting footprint in technological history. Internet of Things will need to focus on the following areas of development:

## 7.1 Security and Regulation

As internet crime increasingly continues to rise (125% in 2021 compared to 2020), cybersecurity and the regulation of the same is an important focus for the developers of Internet of Things technology.

Internet of Things developers are expected to incorporate practical cybersecurity measures such as encryption, certificate-based authentication, and security standardization across platforms.

Developers of the Internet of Things will be challenged by ensuring the technology and security measures conform to privacy legislation across global jurisdictions. In Canada, developers must ensure they comply with the federal Personal Information Protection and Electronic Documents Act.

## 7.2 Trends in Technology

A large focus of developers of the Internet of Things throughout the development of the technology will be to keep up with ever-changing trends and societal progression. As society evolves, technology is expected to become ever more integrated into the consumer’s daily life.

Internet of Things is expected to expand to a futuristic society that incorporates technology such as smart cities, wearables, and transportation. As such, developers are expected to focus much of their attention on ensuring the Internet of Things technology is advanced in a way that can support such a change in global societies.

## 7.3 Market Growth

It is important for developers of Internet of Things technology to be aware of the differing market needs and industries of the many users of the technology. For example, the needs and wants of a user of wearable technology will differ greatly from a user of medical technology.

Developers will have to implement practical approaches and apply revolutionary as well as precedent techniques to ensure that different industries are accommodated. This will ensure a smooth transition to Internet of Things technology.

Despite the many challenges that the developers of Internet of Things technology will face in the coming years, there is no doubt that the Internet of Things will have a significant impact on society as a whole.

# Summary and Conclusion

9.1 Summary

9.2 Conclusion

# References

Articles :

Bhattacharayya, R. et al. (2010). Low-cost, ubiquitous RFID-tag-antenna-based sensing. Proceedings of the IEEE. 98 (10). 1593-1600.

L. D. Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," in IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233-2243, Nov. 2014, doi: 10.1109/TII.2014.2300753.

Li, S., Xu, L.D. & Zhao, S. The internet of things: a survey. Inf Syst Front 17, 243–259 (2015). https://doi.org/10.1007/s10796-014-9492-7

Madakam, S. , Ramaswamy, R. and Tripathi, S. (2015) Internet of Things (IoT): A Literature Review. Journal of Computer and Communications, 3, 164-173. doi: 10.4236/jcc.2015.35021.

LaFlamme, R., & LaFlamme, R. (2023, October 30). *What’s next for the Internet of Things* Auvik. https://www.auvik.com/franklyit/blog/future-of-it/

Imber, D. (2024, February 21). The Latest Cyber Crime Statistics (updated February 2024) AAG IT Support. *AAG IT Services*. https://aag-it.com/the-latest-cyber-crime-statistics#:~:text=Cyber%20Crime%20Overview,businesses%20and%20individuals%20in%2 2022.

Legislative Services Branch. (2019, June 21). *Consolidated federal laws of Canada, Personal Information Protection and Electronic Documents Act*. https://laws-lois.justice.gc.ca/eng/acts p-8.6FullText.html#:~:text=An%20Act%20to%20support%20and,the%20Statutory%20Instrums20Act%20and

Websites:

*Temperature Monitoring with Arduino IoT Cloud using DHT22*. (n.d.). projecthub.arduino.cc. https://projecthub.arduino.cc/attari/temperature-monitoring-with-arduino-iot-cloud-using-dht22-cd8e34

# Appendix

## Screenshot 2024-03-03 at 11.10.02 AM.pngScreenshot 2024-03-03 at 11.06.37 AM.png1. Arduino MKRWfFi1010V2.0 Schematic

## 2. Raw Experiment Data

| **Time** | **Humidity** | **Temperature** |
| --- | --- | --- |
| 2024-02-23T00:15:07.154216356Z | 870.5 | 640.2999877929690 |
| 2024-02-23T00:15:27.16232095Z | 870.5999755859380 | 640.2000122070310 |
| 2024-02-23T00:16:07.189374697Z | 870.7000122070310 | 640.0999755859380 |
| 2024-02-23T00:16:27.197187536Z | 870.5999755859380 | 640.2000122070310 |
| 2024-02-23T00:17:27.372441351Z | 870.5 | 640.0999755859380 |
| 2024-02-23T00:17:47.223373163Z | 870.7000122070310 | 640.2000122070310 |
| 2024-02-23T00:18:27.23820977Z | 870.5999755859380 | 640 |
| 2024-02-23T00:18:47.65518405Z | 870.7000122070310 | 640.0999755859380 |
| 2024-02-23T00:19:07.293200722Z | 870.5999755859380 | 640 |
| 2024-02-23T00:19:27.298825208Z | 870.7000122070310 | 640.0999755859380 |
| 2024-02-23T00:19:47.306921705Z | 870.5999755859380 | 640 |
| 2024-02-23T00:20:07.315180638Z | 870.7000122070310 | 640.0999755859380 |
| 2024-02-23T00:21:07.741060404Z | 871.0999755859380 | 640.5999755859380 |
| 2024-02-23T00:21:27.371427129Z | 845.5 | 640.5 |
| 2024-02-23T00:21:47.449481926Z | 0 | 0 |
| 2024-02-23T00:23:11.921398119Z | 0 | 0 |
| 2024-02-23T00:25:34.866884658Z | 0 | 0 |
| 2024-02-23T00:25:49.308228086Z | 819.7999877929690 | 666.2000122070310 |
| 2024-02-23T00:27:03.422177734Z | 819.7999877929690 | 665.7999877929690 |
| 2024-02-23T00:27:19.556141981Z | 33.29999923706060 | 665.7999877929690 |
| 2024-02-23T00:27:39.562747806Z | 33.5 | 25.5 |
| 2024-02-23T00:27:59.575034211Z | 34 | 25.399999618530300 |
| 2024-02-23T00:28:19.583631639Z | 33.5 | 25.5 |
| 2024-02-23T00:29:39.617700076Z | 33.599998474121100 | 25.399999618530300 |
| 2024-02-23T00:29:59.623955856Z | 33.70000076293950 | 25.299999237060500 |
| 2024-02-23T00:30:39.960013182Z | 33.79999923706060 | 25.200000762939500 |
| 2024-02-23T00:30:59.661877726Z | 33.900001525878900 | 25.100000381469700 |
| 2024-02-23T00:31:39.685076624Z | 34 | 25.200000762939500 |
| 2024-02-23T00:31:59.701111112Z | 86.5999984741211 | 25.100000381469700 |
| 2024-02-23T00:32:19.903859517Z | 81.30000305175780 | 26 |
| 2024-02-23T00:32:39.715065425Z | 60.20000076293950 | 26.200000762939500 |
| 2024-02-23T00:32:59.728030558Z | 38.79999923706060 | 25.799999237060500 |
| 2024-02-23T00:33:19.75402825Z | 36.79999923706060 | 25.5 |
| 2024-02-23T00:33:40.191138091Z | 36.29999923706060 | 25.399999618530300 |
| 2024-02-23T00:33:59.768802624Z | 36 | 25.200000762939500 |
| 2024-02-23T00:34:19.785873899Z | 35.79999923706060 | 26 |
| 2024-02-23T00:34:39.792169472Z | 35.70000076293950 | 27.799999237060500 |
| 2024-02-23T00:34:59.801103872Z | 35.599998474121100 | 25.799999237060500 |
| 2024-02-23T00:35:20.131639357Z | 71.80000305175780 | 25.700000762939500 |
| 2024-02-23T00:35:39.81817694Z | 95 | 25.5 |
| 2024-02-23T00:35:59.847205574Z | 58.5 | 25.299999237060500 |
| 2024-02-23T00:36:19.842703549Z | 39.5 | 25.200000762939500 |
| 2024-02-23T00:36:40.004774028Z | 37.099998474121100 | 25.100000381469700 |
| 2024-02-23T00:36:59.862867658Z | 36.400001525878900 | 25.200000762939500 |
| 2024-02-23T00:37:19.872705943Z | 36.099998474121100 | 25.100000381469700 |
| 2024-02-23T00:37:39.883085759Z | 36 | 25 |
| 2024-02-23T00:37:59.908612733Z | 35.79999923706060 | 25.100000381469700 |
| 2024-02-23T00:38:20.357643565Z | 35.599998474121100 | 24.899999618530300 |
| 2024-02-23T00:38:39.913532292Z | 35.5 | 25.200000762939500 |

## 3. Arduino Boilerplate Code

/\*

Sketch generated by the Arduino IoT Cloud Thing "MKR WiFi 1010 and DHT22"

https://create.arduino.cc/cloud/things/e75efe13-eb5e-432a-86d3-0bf1cd34aaac

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

CloudTemperatureSensor temperature;

CloudRelativeHumidity humidity;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions

which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

\*/

#include "thingProperties.h"

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#define DHTPIN 7 // Digital pin connected to the DHT sensor

#define DHTTYPE DHT22 // Write DHT11 or DHT22 According to your Sensor

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

unsigned long previousMillis = 0;

const long interval = 20000; //milliseconds total time for 20 Seconds

void setup() {

// Initialize serial and wait for port to open:

Serial.begin(9600);

// This delay gives the chance to wait for a Serial Monitor without blocking if none is found

delay(1500);

// Defined in thingProperties.h

initProperties();

// Connect to Arduino IoT Cloud

ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/\*

The following function allows you to obtain more information

related to the state of network and IoT Cloud connection and errors

the higher number the more granular information you’ll get.

The default is 0 (only errors).

Maximum is 4

\*/

setDebugMessageLevel(2);

ArduinoCloud.printDebugInfo();

dht.begin(); //Init DHT

Serial.println(F("DHTxx Unified Sensor Example"));

// Print temperature sensor details.

sensor\_t sensor;

dht.temperature().getSensor(&sensor);

Serial.println(F("------------------------------------"));

Serial.println(F("Temperature Sensor"));

Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);

Serial.print (F("Driver Ver: ")); Serial.println(sensor.version);

Serial.print (F("Unique ID: ")); Serial.println(sensor.sensor\_id);

Serial.print (F("Max Value: ")); Serial.print(sensor.max\_value); Serial.println(F("°C"));

Serial.print (F("Min Value: ")); Serial.print(sensor.min\_value); Serial.println(F("°C"));

Serial.print (F("Resolution: ")); Serial.print(sensor.resolution); Serial.println(F("°C"));

Serial.println(F("------------------------------------"));

// Print humidity sensor details.

dht.humidity().getSensor(&sensor);

Serial.println(F("Humidity Sensor"));

Serial.print (F("Sensor Type: ")); Serial.println(sensor.name);

Serial.print (F("Driver Ver: ")); Serial.println(sensor.version);

Serial.print (F("Unique ID: ")); Serial.println(sensor.sensor\_id);

Serial.print (F("Max Value: ")); Serial.print(sensor.max\_value); Serial.println(F("%"));

Serial.print (F("Min Value: ")); Serial.print(sensor.min\_value); Serial.println(F("%"));

Serial.print (F("Resolution: ")); Serial.print(sensor.resolution); Serial.println(F("%"));

Serial.println(F("------------------------------------"));

// Set delay between sensor readings based on sensor details.

delayMS = sensor.min\_delay / 1000;

STHAM();

}

void loop() {

ArduinoCloud.update();

// Your code here

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= interval) {

STHAM();

previousMillis = currentMillis;

}

}

void STHAM(){

// Get temperature event and print its value.

sensors\_event\_t event;

dht.temperature().getEvent(&event);

if (isnan(event.temperature)) {

Serial.println(F("Error reading temperature!"));

//Assign temperature value 0 to Cloud Variable

temperature=0;

}

else {

Serial.print(F("Temperature: "));

Serial.print(event.temperature);

Serial.println(F("°C"));

//Assign temperature value to Cloud Variable

temperature=event.temperature;

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

if (isnan(event.relative\_humidity)) {

Serial.println(F("Error reading humidity!"));

//Assign humidity value 0 to Cloud Variable

humidity=0;

}

else {

Serial.print(F("Humidity: "));

Serial.print(event.relative\_humidity);

Serial.println(F("%"));

//Assign humidity value to Cloud Variable

humidity=event.relative\_humidity;

}

}

/\*

Since Temperature is READ\_WRITE variable, onTemperatureChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onTemperatureChange() {

// Add your code here to act upon Temperature change

}

/\*

Since Humidity is READ\_WRITE variable, onHumidityChange() is

executed every time a new value is received from IoT Cloud.

\*/

void onHumidityChange() {

// Add your code here to act upon Humidity change

}

## 4. GitHub Repository

<https://github.com/PMcTwist/IoT_Project>

A screenshot of a computer

Description automatically generated