# CoAP based IoT data transfer from a Raspberry Pi to Cloud

Thomas Lee Scott
*School of Computing, Mathematics & Digital Technology*
*Manchester Metropolitan University*
Manchester, United Kingdom
tomscott292@gmail.com

***Abstract*—This research investigates the use of The Constrained Application Protocol (CoAP) in transmitting sensor data to the cloud. It aims to explore how CoAP fits into the Internet of Things (IoT) ecosystem and what advantages, if any, it offers over other IoT protocols. A framework is proposed using a Raspberry Pi (RPi) and sensor acting as an IoT endpoint. This endpoint will poll the sensor and using CoAP will send the latest data formatted as JavaScript Object Notation (JSON) to a CoAP cloud endpoint at regular intervals. The cloud service will receive, format and display the data from the RPi to the user.**

***Index Terms*—Internet of Things, CoAP, M2M, constrained devices, Raspberry Pi board.**

## I. Introduction

The reduced cost of low powered small devices, such as the RPi, has made it more accessible to create bespoke systems. This combined with the increasing popularity of home automation allows for these devices to be used in the IoT.

The IoT can be viewed as a large distributed network comprising of highly dynamic devices [1]. Small low powered "smart" devices can connect and communicate with one another. Some of these devices can contain or communicate with sensors that record real world data. This data can then be transmitted to other devices allowing them to trigger actions. In this way groups of smart devices can be used to improve day to day situations such as automated houses (thermostats and heating etc.), security and improved monitoring.

The Raspberry Pi [2] is a credit card sized computer developed by the Raspberry Pi Foundation. The RPi's ability to act as a GNU/Linux server and the interfacing services provided by its general purpose I/O pins make it a popular choice of hardware for IoT applications. [3]

With 48% of the UK market considering their smartphone as the most important device for Internet access [4], allowing users to use their handheld devices to view and manage their data has become increasingly necessary. Cloud platforms that allow access from any device go a long way to solving this problem. Storing sensor data in the cloud allows for easy access to users from any device as well as allowing for scalable storage.

As these devices are limited in computing power, it is important that the devices communicate efficiently. This paper explores the use of CoAP as a protocol to transmit sensor data from a small, low powered device (RPi) to send sensor data to the cloud.

In this system, a sensor will be attached to the RPi; the RPi will be responsible for taking the data from the sensor and then using the CoAP protocol to transmit this data to the cloud platform.

### A. Motivation

With the increasing adoption of IoT systems and technology in day to day life, it becomes increasingly important for the devices to be able to operate at peak efficiency. As there are numerous technologies and standards that allow different IoT devices to communicate, this paper investigates the use of CoAP in order to determine: 1) how CoAP transfers data from a IoT device to another CoAP node. 2) how a RPi can be used as a flexible platform to provide sensor data. 3) how this data can be sent to a cloud platform.

### B. Related Work and Contribution

[5] carries out a similar investigation, using a RPi device as a IoT node connected to sensors. The RPi collects the data from the sensors and then transmits the data to a cloud platform. The cloud platform in this instance is a HyperText Transfer Protocol (HTTP) server which will receive the sensor data and display it to the user. [5] proposes using the Message Queuing Telemetry Transport (MQTT) protocol to transmit the sensor data from the RPi to the HTTP server. MQTT is a popular IoT protocol developed to specialise in the transfer of data from Wireless Sensor Networks (WSNs) [6]. MQTT works on a publish/subscribe model; in [5] the RPi acts a *publisher*, publishing the sensor data to the broker and the HTTP server acts as a *subscriber*. The MQTT broker is responsible for coordinating subscribers to the data and subscribers will usually have to contact the broker explicitly in order to subscribe [6]. This contrasts to the approach taken in this proposal using CoAP, where each node in the CoAP network acts as both a server and a client in a more traditional HTTP model and nodes within the infrastructure will communicate with one another directly.

[7] used a RPi connected to sensors to measure patients' body temperature and transmit this data wirelessly to the cloud. In that paper, the data was transmitted to an Amazon Web Services (AWS) cloud computing platform. There the

data was stored, mined in order to make decisions, and displayed to the user allowing the data to be updated and reviewed. The data was transmitted from the RPi to the AWS server using Secure Socket Layer (SSL). The development of specialised protocols for constrained devices, such as CoAP and MQTT could allow these health monitoring RPis to save power, save network bandwidth and potentially receive more readings to process.

[8] used a RPi combined with a DHT22 sensor to measure the indoor temperature in real time. This data was transmitted using HTTP to a Representational State Transfer (REST) Application programming interface (API), where the temperature was stored in a database. These temperatures were then used to inform an application replicating the actions of an air conditioner. The use of a RESTful API in this paper would allow the project to easily be adapted to using CoAP to replace HTTP.

## II. Background

### A. Internet of Things

The Internet of Things (IoT) is an umbrella term used to describe physical 'smart' devices equipped with telecommunication interfaces, connected to one another via the Internet [9]. Whereas the Internet traditionally connected computers, the embedding of electronics into physical objects has allowed the Internet to expand [1]. These devices can contain sensors which will produce data and in some cases these devices can be controlled remotely. The combination of these devices in a network, especially when the actions of one devices are informed by the data from another device, is the foundation of IoT [10]. IoT systems can impact in many areas such as home automation, where devices can work together to automate heating and security aspects of the home [8], and medicine where devices can be used as monitors to provide real time information about patient health [3].

### B. The Constrained Application Protocol

The Constrained Application Protocol (CoAP) is a transfer protocol specialised for use with the web, constrained nodes and constrained networks [11]. The protocol is designed for Machine-to-Machine (M2M) applications and is ideally suited for use within the IoT ecosystem. CoAPs features of observable resources, multicasting, M2M discovery make it a better fit for IoT applications than HTTP [12].

CoAP recognises that web services have become dependant in REST architecture and works to implement a subset of REST common with HTTP while optimising for M2M applications [11]. It achieves this by offering built-in discovery, multicast support and asynchronous message exchanges [11].

CoAP uses a compact binary format with a fixed header size of 4 bytes, exchanging messages over User Datagram Protocol (UDP) or Datagram Transport Layer Security (DTLS) to send messages securely. CoAP resources are addressable by Uniform Resource Identifiers (URIs) and can be interacted with through the same methods as HTTP: GET, PUT, POST and DELETE.

With regards to reliability, CoAP offers four types of messages: Confirmable, Non-Confirmable, Acknowledgement and Reset [13]. After a Confirmable request is sent to a CoAP endpoint, the endpoint will respond with an Acknowledgement message. This message can contain the requested data in a 'piggybacked' response. Otherwise, an empty Acknowledgement message is sent and a Confirmable message will be sent once the data is ready. The original requester will then respond with an empty Acknowledgement message to confirm receipt of the data [11].

## III. Designed CoAP-based IoT Architecture

The system shall consist of four main elements: the sensor, the RPi, CoAP and the cloud platform. The sensor will collect the data and pass this to the RPi. The RPi will then be responsible for manipulating the data into a suitable format for transmission via CoAP. The implementation of CoAP will communicate with the cloud platform. The cloud platform will store the data, allowing access to users.

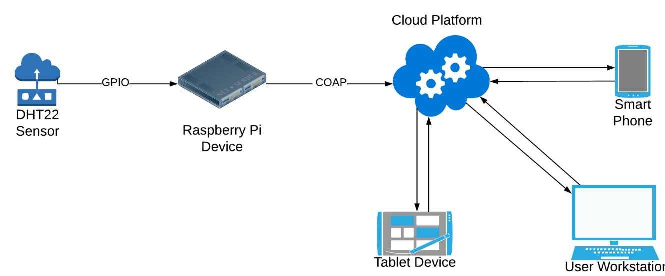


Fig. 1. The project infrastructure.

The DHT22 sensor will connect to the RPi using the RPi's on board General Purpose Input Output (GPIO) ports as shown in Fig. 2. Using the AdaFruit Python DHT library and the CoAPthon library a Python script will create a CoAP endpoint on the RPi. The AdaFruit Python DHT library is a library that provides methods to interact with DHT sensors connected to the RPi's GPIO pins, and the CoAPthon library is a Python implementation of the CoAP protocol.

This endpoint will act as an interface for the DHT22 sensor. The script will use the AdaFruit DHT methods to get the DHT22 sensors temperature and humidity data. This data will then be formatted into JSON in order to be transmitted. Then using the CoAPthon library a CoAP message will be created with the sensors JSON data as the payload. This message will then be sent over UDP to a CoAP URI hosted by the cloud platform. The cloud endpoint will receive the JSON data, format it and display it to the user, who will be accessing the cloud platform via HTTP. This process is shown in Algorithm 1.

**Data:** Temperature and humidity data from sensor
**Result:** Sensor data sent to Cloud CoAP endpoint
initialisation;
define interval in seconds (DHT22 sensor interval is 2 seconds);
**while** *running* **do**
  get latest sensor data from DHT22 sensor;
  **if** *sensor data is returned* **then**
    format data into json object;
    create CoAP Post message with cloud uri as destination;
    send CoAP message;
  **else**
    wait interval time and continue;
  **end**
**end**

**Algorithm 1:** How to get data from sensor and send to cloud.

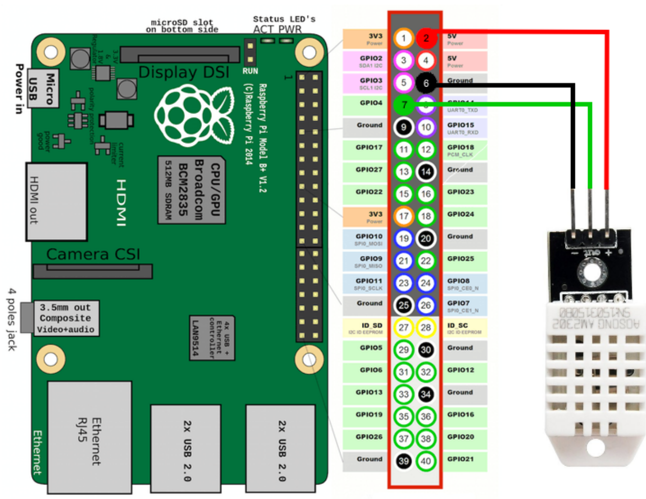


Fig. 2. Wiring diagram for connecting the sensor to the RPi.

## IV. CoAP-based IoT Architecture Components

The aim of the proposed system is to investigate the implementation of CoAP on a RPi and how CoAP can be used to transmit data to the cloud.

To achieve this, a CoAP endpoint will need to be created on the RPi. The RPi will collect sensor data at intervals and store them locally on the device. The RPi will act as a CoAP endpoint that will send to REST POST requests with the sensor data and the time the reading was taken in a JSON format.

The clouds responsibility will be to receive the POST requests from the CoAP endpoint hosted on the RPi and to store and format the data. The cloud should send an Acknowledgement message to the CoAP endpoint, containing a 2.01 (Created) Response Code or a 2.04 (Changed) Response Code and the URI of the created / updated resource [11].

With CoAP endpoints acting as both a client that sends requests and a server that responds to the requests, implementation of CoAP will be needed in each the RPi and the cloud platform. The RPi will then regularly retrieve readings from the sensor and send a POST request at intervals to the cloud CoAP endpoint.

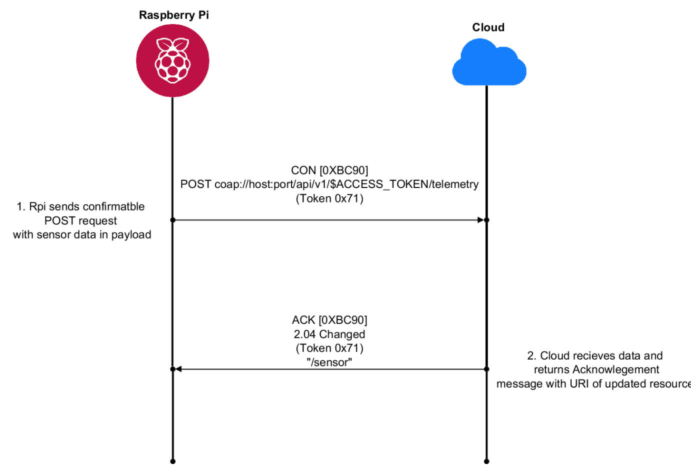


Fig. 3. Diagram showing the communication between RPi and the Cloud.

### A. Hardware

1) Raspberry Pi 3
Raspberry Pi 3 Model B, includes built in WiFi, GPIO ports and a 1.2GHz Quad-Core processor.
2) Micro SD card
Used to load operating system onto the RPi.
3) Power cord
Supplies power to the RPi.
4) DHT22 temperature and humidity sensor
Connects to the RPi using the RPi's GPIO ports. Will be used to provide data.

### B. Software

1) Python 3
The Python programming language will be used to create the scripts and software needed on the RPi. This is due to the language's popularity when creating projects on the RPi and the language's wide selection of networking packages.
2) CoAPthon
Python implementation of CoAP. Licensed under the MIT license. Github.
3) AdaFruit Python DHT
Python library to retrieve sensor data from the DHT22. Github.
4) Git Version Control
Source code version control system to allow for adjustments to the code.
5) ThingsBoard Cloud Platform
Will act as an end point to the RPi where the sensor data will be stored. ThingsBoard Homepage

## V. Conclusion and Future Work

This research investigates the use of CoAP in transmitting sensor data to the cloud. It aims to explore how CoAP fits into the IoT ecosystem and what advantages it offers over other IoT protocols. It also shows how a RPi can be used with Python to create an IoT device capable of transferring data to the cloud.

The RPi has been connected to a DHT22 temperature and humidity sensor using the RPis GPIO pins. A Python script running on the RPi uses an external library provided by AdaFruit to poll the sensor at intervals and obtain the current temperature and humidity. This data is then formatted and displayed to the console. Further work can be done in connecting the RPi to the cloud. The Python script running on the RPi can be improved by having a CoAP endpoint created which can then send the sensor data to the cloud endpoint. This can be achieved using the CoAPthon library identified in this paper. The cloud platform provided by ThingWorks allows for IoT devices to send data directly to the cloud by providing a CoAP REST API to which data can be sent.

## References


[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870512000674

[2] R. Pi, "model B," *Raspberrypi. org. Saatavissa: https://www. raspberrypi. org/products/raspberry-pi-3-model-b/. Hakupäivä*, vol. 6, p. 3, 2018.

[3] R. Kumar and M. P. Rajasekaran, "An IoT based patient monitoring system using raspberry Pi," in *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, Jan. 2016, pp. 1–4.

[4] Ofcom, "The Communications Market 2018: Narrative report," Aug. 2018. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/cmr/cmr-2018/report

[5] S. D. Rode, S. Sagrolikar, and M. S. Kulkarni, "IOT based Raspberry PI home Automation Using Cloud," vol. 3, no. 2, p. 5, 2017.

[6] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, Jan. 2008, pp. 791–798.

[7] M. S. Jassas, A. A. Qasem, and Q. H. Mahmoud, "A smart system connecting e-health sensors and the cloud," in *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, May 2015, pp. 712–716.

[8] C. Lee, S. Park, Y. Jung, Y. Lee, and M. Mathews, "Internet of Things: Technology to Enable the Elderly," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, Jan. 2018, pp. 358–362.

[9] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016.

[10] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," *IEEE Internet Initiative*, vol. 1, pp. 1–86, 2015.

[11] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC Editor, Tech. Rep. RFC7252, Jun. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7252

[12] M. Kovatsch, M. Lanter, and Z. Shelby, "Californium: Scalable cloud services for the Internet of Things with CoAP," in *2014 International Conference on the Internet of Things (IOT)*, Oct. 2014, pp. 1–6.

[13] P. Bellavista and A. Zanni, "Towards better scalability for IoT-cloud interactions via combined exploitation of MQTT and CoAP," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. Bologna, Italy: IEEE, Sep. 2016, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7740614/