



# Raspberry Pi Meets IBM i

Andy Youens  
FormaServe Systems Ltd  
IBM Champion 2021, 2022 & 2023



# Presenter - Andy Youens



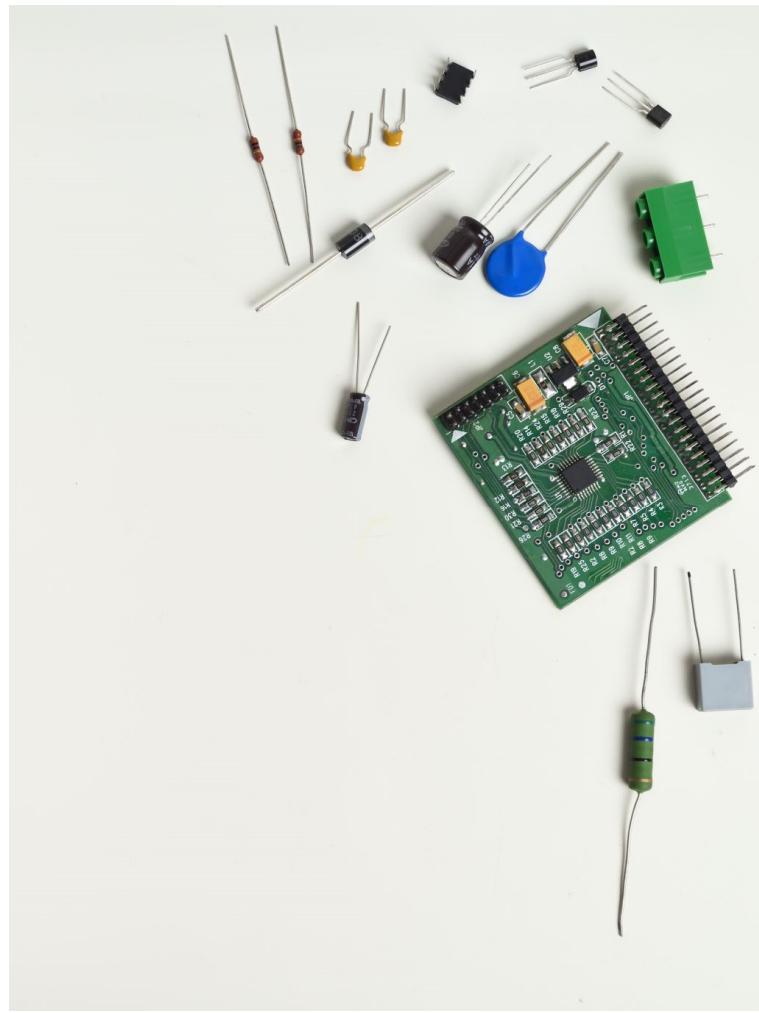


- We are proud to be a leading IBM i software vendor with a remarkable track record of over 33 years
- We offer comprehensive IBM i training courses, both remotely & on-site, to equip individuals & teams with the necessary skills & knowledge
- As open-source on IBM i specialists, we possess deep expertise in leveraging open-source technologies to enhance & optimise IBM i environments
- Our team comprises of modernisation experts who can guide you in transforming your IBM i to embrace modern practices & technologies
- Part of IBM ISV Advisory Council
- **Provide Open-Source support to i-UG education initiative**



- FormaServe & NC Communications have announced a strategic partnership to publish PowerWire
- Do you want to write a blog for us & share your insights & experiences with the IBM i community?
- Do you want to advertise with us & reach thousands of potential customers & partners in the IBM i market?
- Give me a shout!

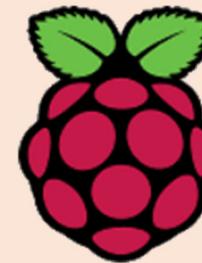




 FormaServe

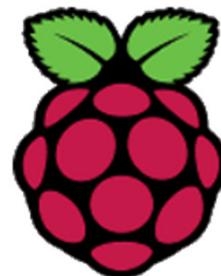
# Agenda

- Intro to Pico W & Raspberry Pi
- How to code on the Pico & Pi
- Communications between all 3
- Storing images on IBM i
- Demo of all 3 bits of kit working together



**IBM i**

# Raspberry Pi & Pico



Raspberry Pi®



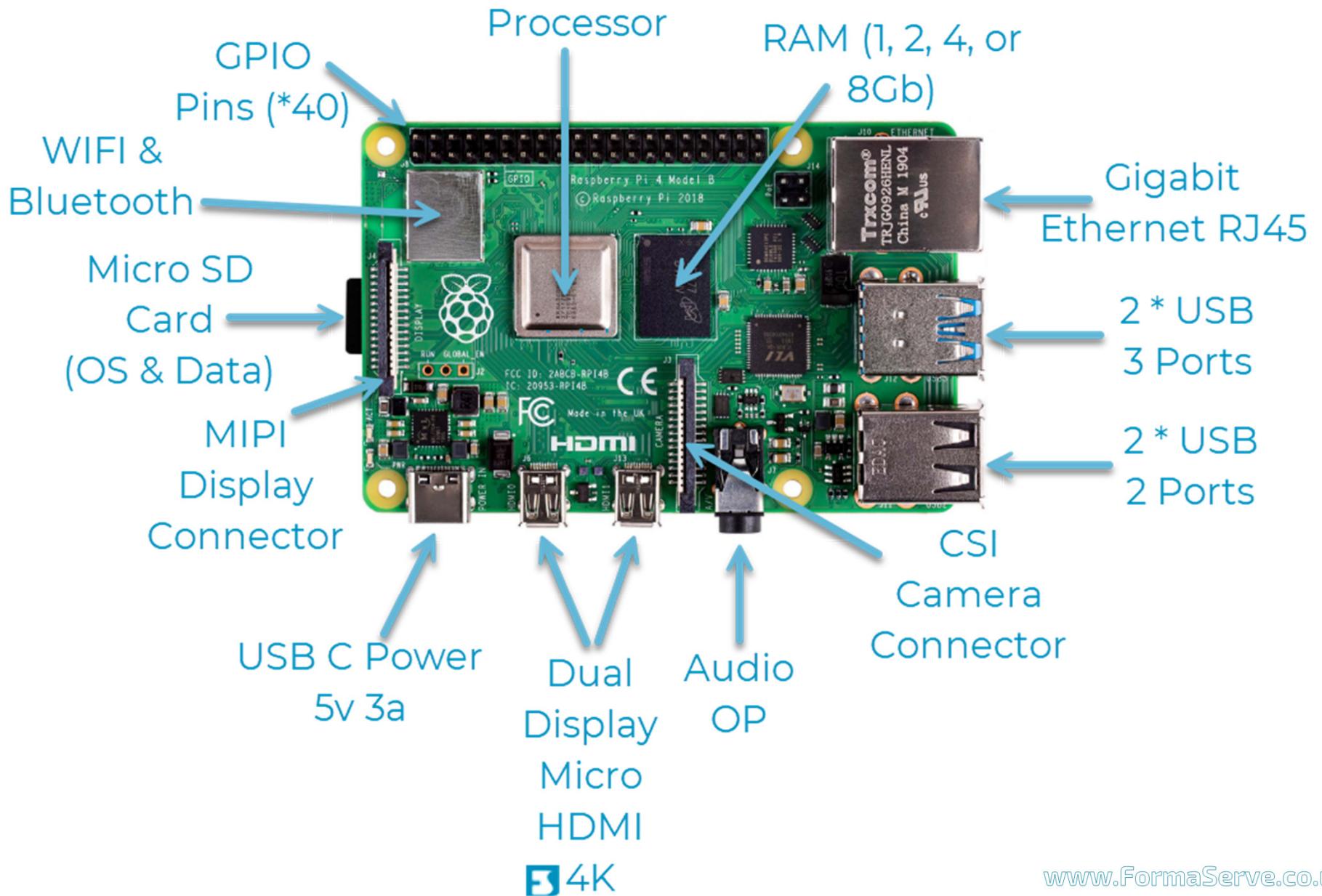
***Don't let its size fool you!***

# What is the Raspberry Pi?



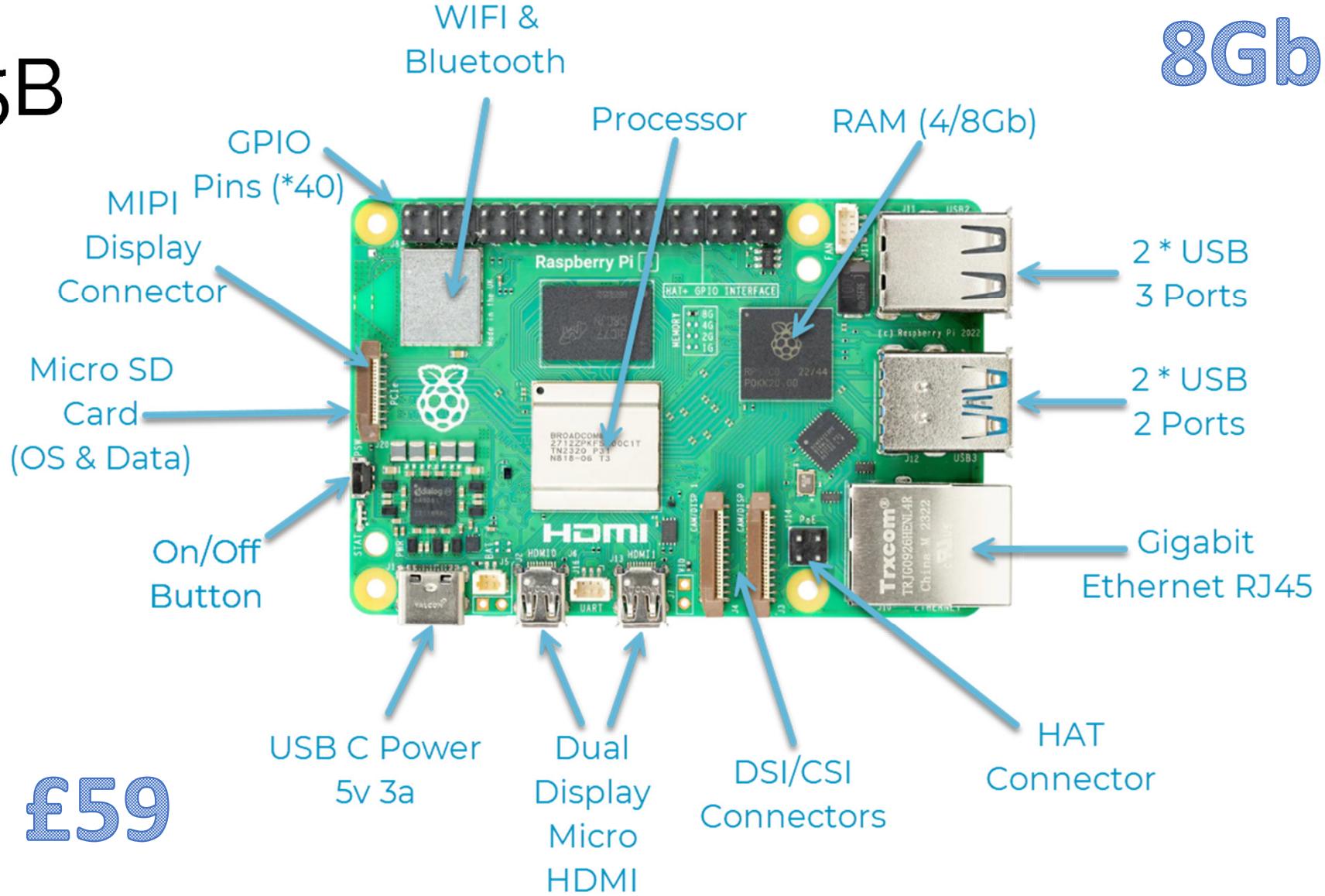
- ➊ Raspberry Pi is the worlds most popular SBC (Single Board Computer)
- ➋ Does pretty much everything a desktop can do
- ➌ Suitable for all ages
- ➍ Comes with all the software you require
- ➎ Great for developers

# Pi 4B



# Pi 5B

**8Gb £79**



**4Gb £59**

# Raspberry Pi Pins

- GPIO (General Purpose IO)
- SPI (Serial Peripheral Interface)
- I<sup>2</sup>C (Inter-integrated Circuit)
- UART (Universal Asynchronous Receiver/Transmitter)
- PCM (Pulse Code Modulation)
- Ground
- 5V (Power)
- 3.3V (Power)

3v3 Power	1	■	●	2	5v Power
GPIO 2 (I <sup>2</sup> C1 SDA)	3	●	●	4	5v Power
GPIO 3 (I <sup>2</sup> C1 SCL)	5	●	●	6	Ground
GPIO 4 (GPCLK0)	7	●	●	8	GPIO 14 (UART TX)
Ground	9	●	●	10	GPIO 15 (UART RX)
GPIO 17	11	●	●	12	GPIO 18 (PCM CLK)
GPIO 27	13	●	●	14	Ground
GPIO 22	15	●	●	16	GPIO 23
3v3 Power	17	●	●	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	●	●	20	Ground
GPIO 9 (SPI0 MISO)	21	●	●	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	●	●	24	GPIO 8 (SPI0 CE0)
Ground	25	●	●	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	●	●	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	●	●	30	Ground
GPIO 6	31	●	●	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	●	●	34	Ground
GPIO 19 (PCM FS)	35	●	●	36	GPIO 16
GPIO 26	37	●	●	38	GPIO 20 (PCM DIN)
Ground	39	●	●	40	GPIO 21 (PCM DOUT)

<https://pinout.xyz/pinout/ground>

# Raspberry Pi Operating Systems

## Raspberry Pi OS

- a Linux distro
- Was Raspbian

## Ubuntu

## Windows 10 IoT Core

## Other Linux distro's can be installed

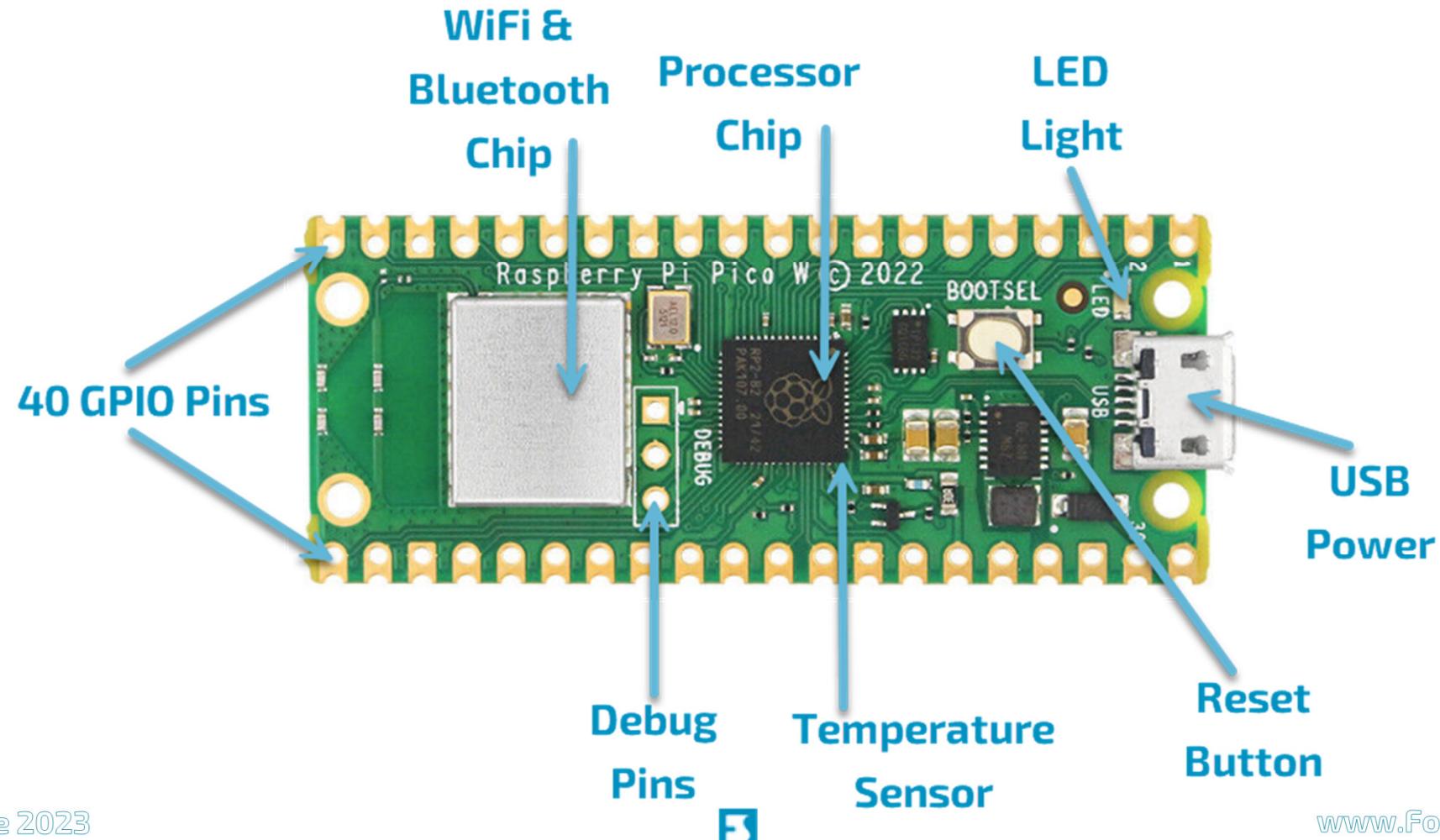
- Ubuntu Mate
- Arch Linux ARM
- Fedora

# What is the PICO W?

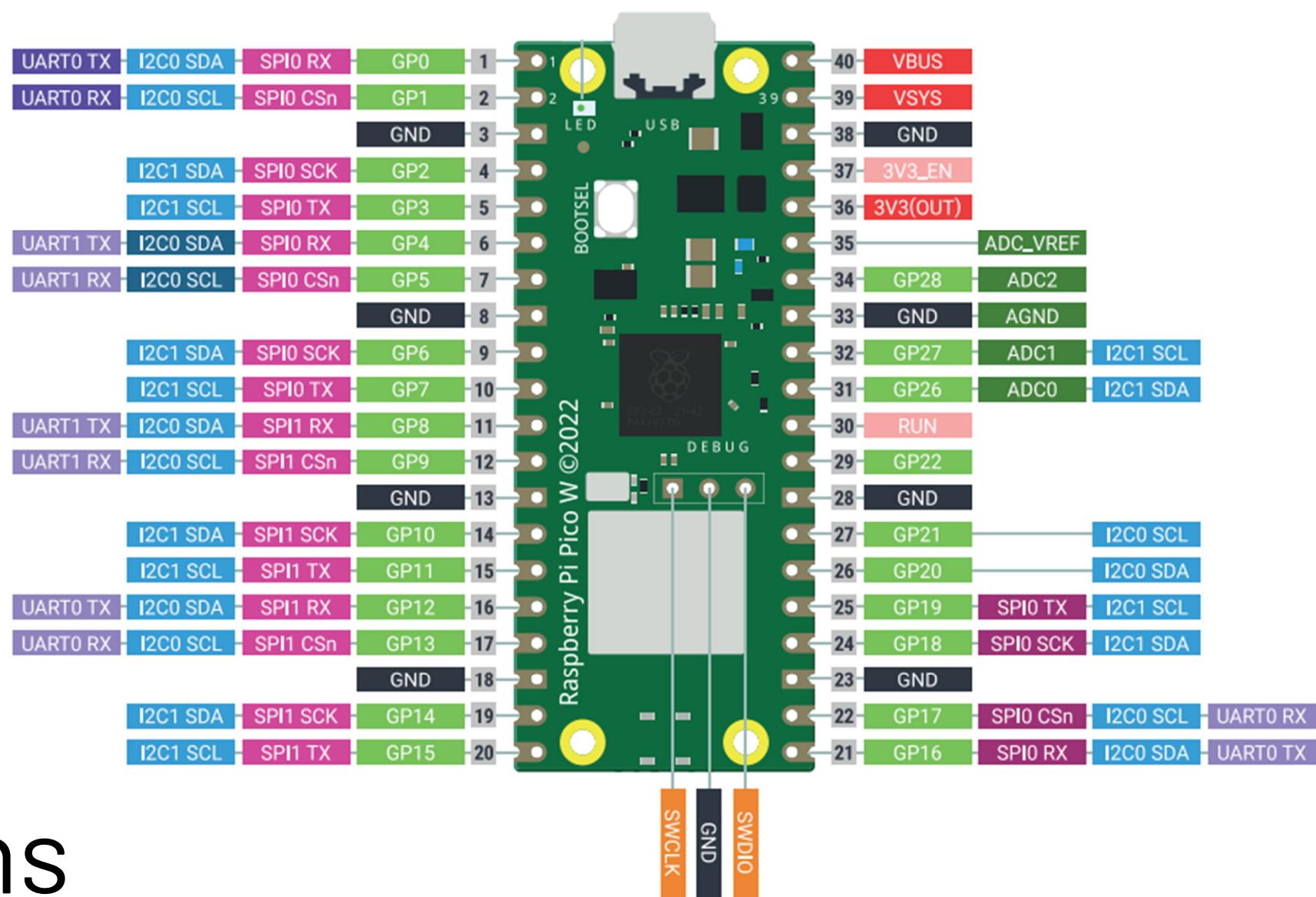


- Aka Pi Cow 😎
- A microcontroller board
- Released in 2022
- Has a WIFI module, allows it to communicate with Wi-Fi & Bluetooth
- Pico W is built around the RP2040 microcontroller chip
- Has 40 GPIO pins for sensors & IOT devices
- Onboard LED & temperature sensor
- Runs Micro-Python

# Pico W Components



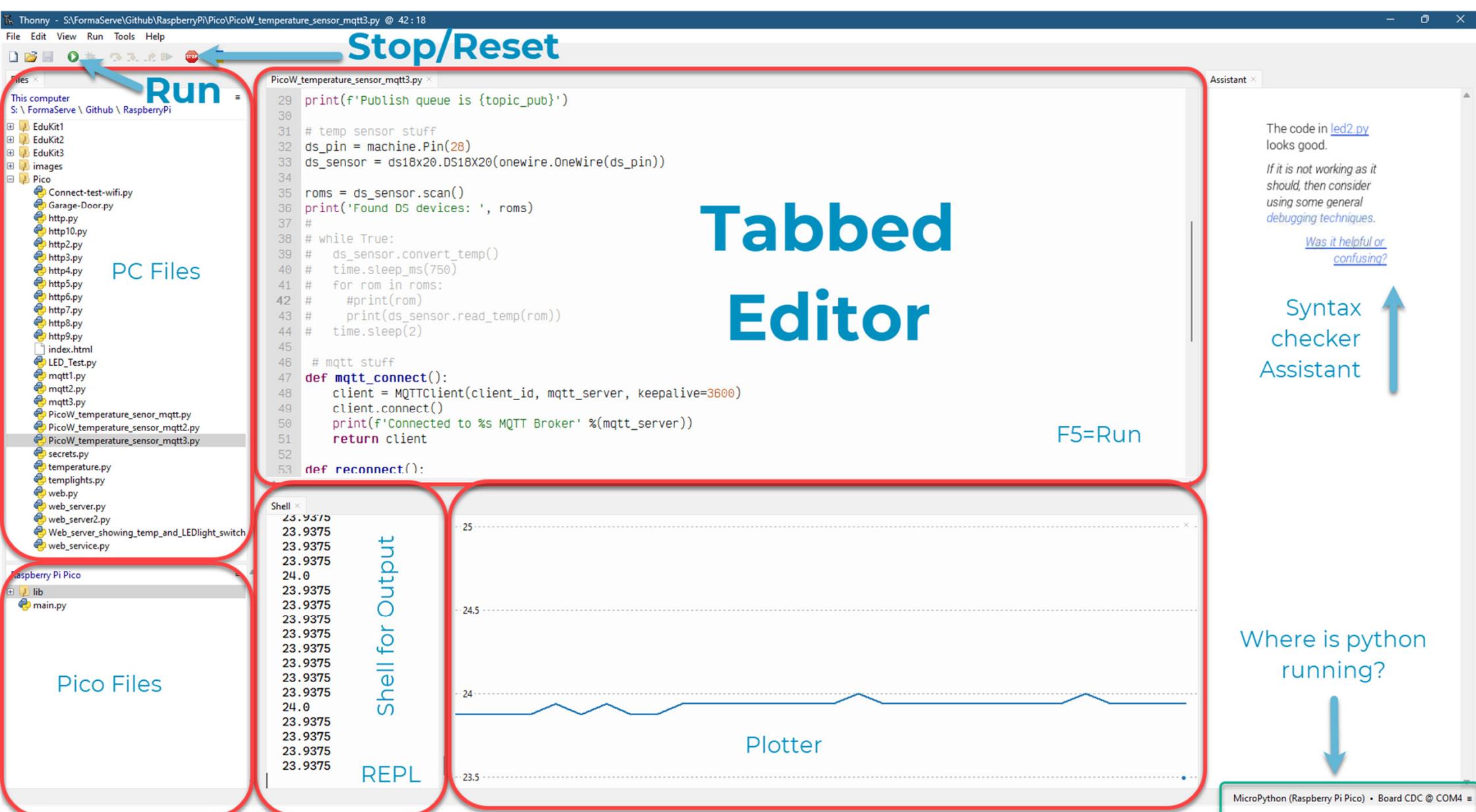
RP2040



# Pico Pins

# Thonny IDE

- The easiest way to program your pi devices
- Installed already on Raspberry Pi
- Latest version 4.1.2 August 2023
- Can be installed on Windows/Mac
- Download from
  - <https://thonny.org/>
- Comes with Python installed
- Is Open Source!



# Python Programming Language

---

- A high-level, interpreted programming language known for its simplicity
- Created in the late 1980s, it was designed to be easy to learn & use
- Python supports many programming paradigms, including procedural, object-oriented & functional programming
- One of Python's key strengths is its extensive standard library, providing a wide range of modules & packages
- Variables can be assigned without explicitly specifying their data type
- Indentation is crucial in Python, it is used to define code blocks
- Is platform-independent, making it a portable language
- Its open-source
- Vast ecosystem of third-party libraries & frameworks



# Pico Projects



- **Garage Doors**

- Is my garage door open?
- Opening/closing a garage door from mobile

- **Soil Moisture Monitor**

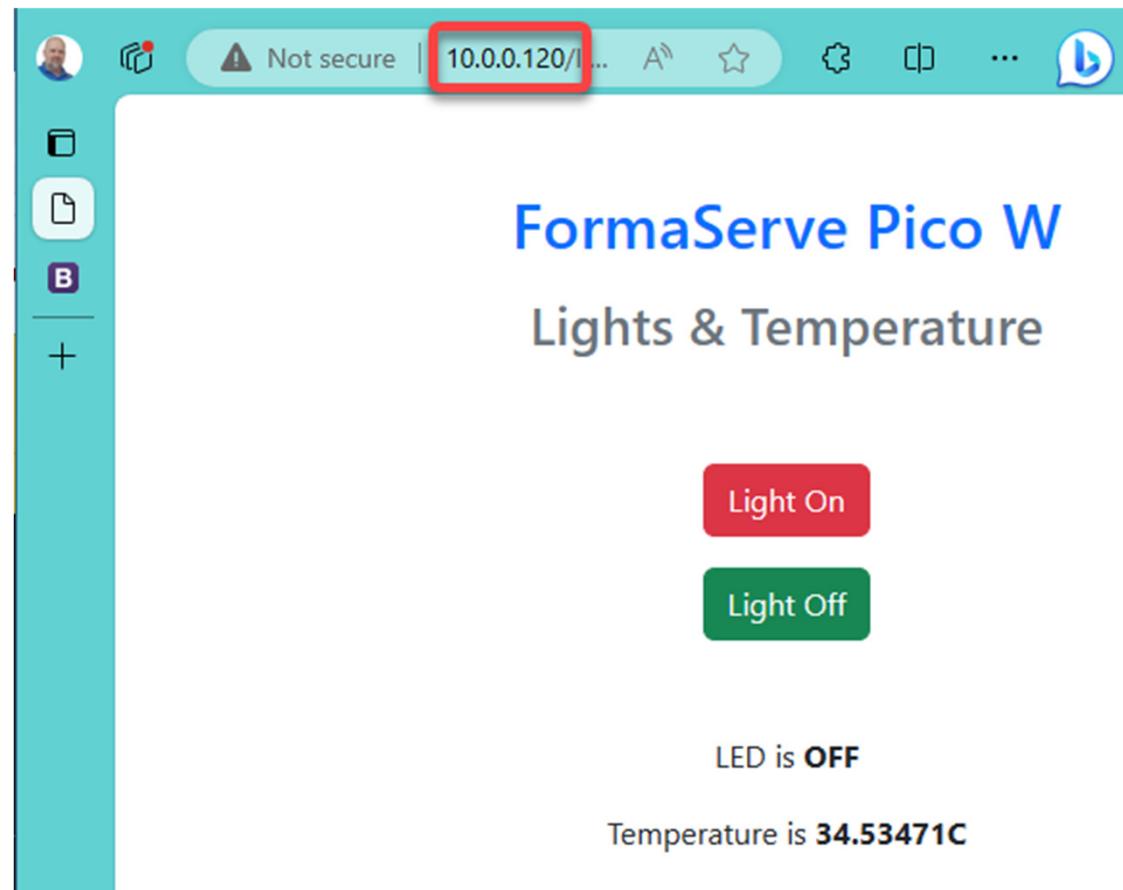
- **Weather Station Monitor**

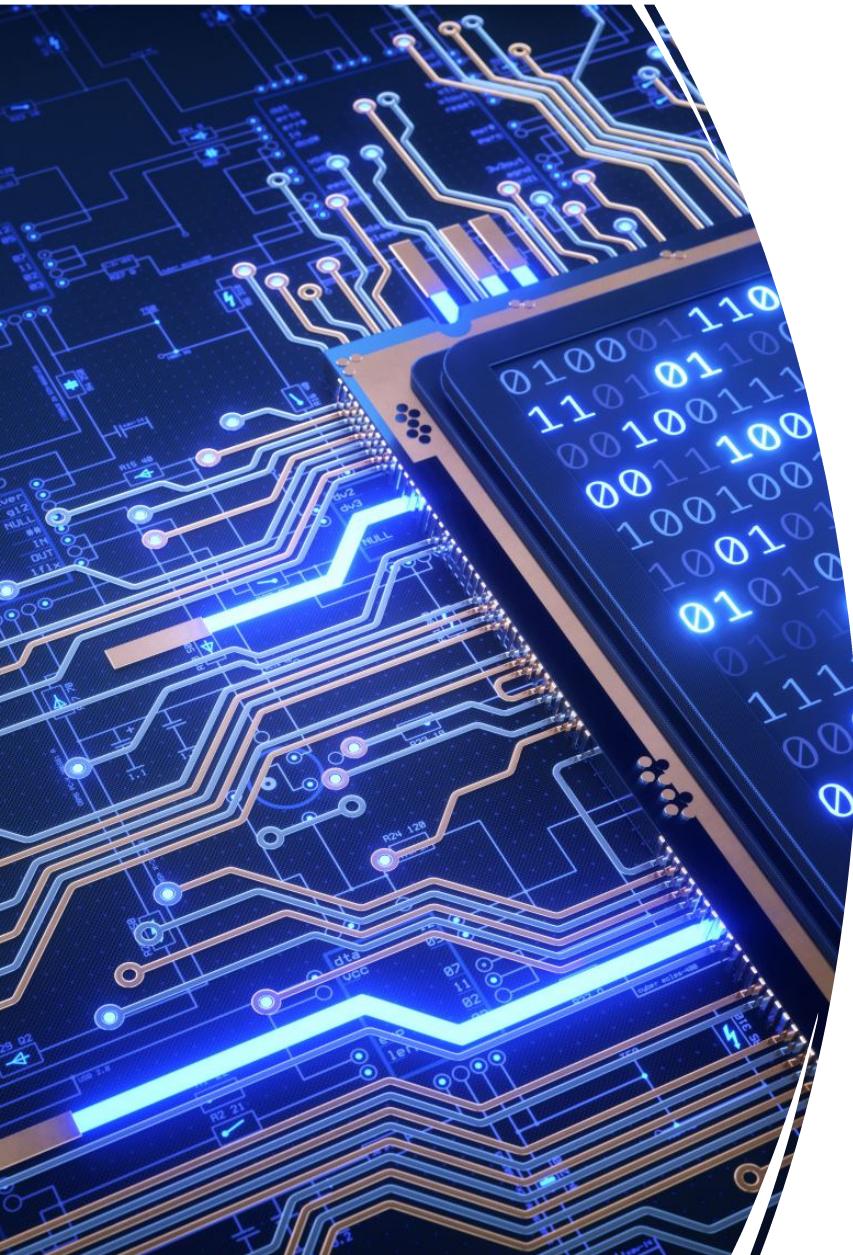
- **Provide Web Services & Web Pages**

# Pico W Demo

---

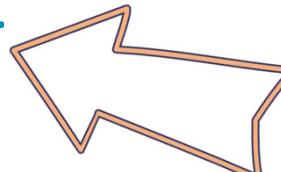
- We will
  - Get the onboard temperature & light status
  - Display it in a browser
  - Allow the onboard LED to be switched on/off





# Communicating with the IBM i - How?

- 
- **Remote Shells**
    - SSH/SFTP/Telnet/FTP/RSync
  - **Database Connectivity**
    - ODBC/JDBC
  - **Web Services**
    - REST/SOAP interfaces
  - **IBM MQ**
  - **Node-RED**
  - **MQTT**



# What is MQTT?

---

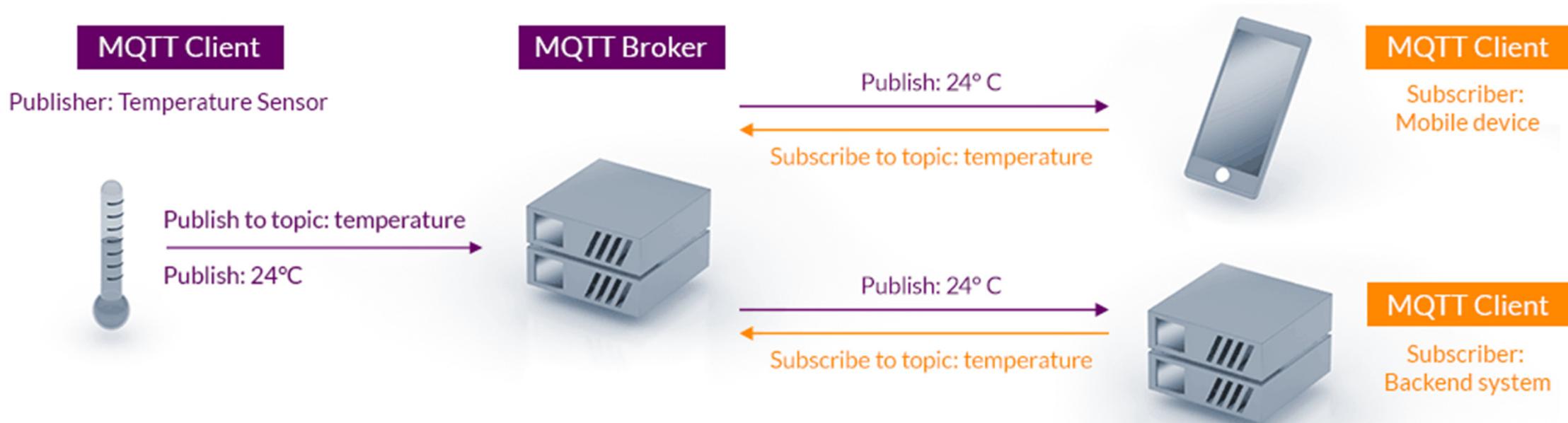
-  A messaging protocol designed for efficient communication between devices
-  Clients publish messages to topics & clients subscribe to topics to receive messages
-  Known for its simplicity, low bandwidth usage & support for unreliable networks
-  Widely adopted for IoT applications due to its efficiency & scalability
-  Developed by IBM in 1999



# Who Uses MQTT?

- **Automotive** - Used by car manufacturers like Ford to manage their car-sharing service that operates a fleet of over 14,000 vehicles across 18 cities. Also used for predictive maintenance, driver assistance & autonomous driving features.
  - **Logistics** - Used by companies like DHL to enable real-time monitoring of their autonomous drones that deliver medical supplies
  - **Manufacturing** - Used by companies like Celikler Holding to monitor their power plant operations. Enables easy data transfer from the factory floor to the cloud.
  - **Smart Home** - Used by many consumer IoT devices like thermostats, appliances, security systems, smart watches, fitness trackers, etc.
- Everyone!**

# MQTT Architecture



# Pub & Sub



From: <https://mqtt.org/>

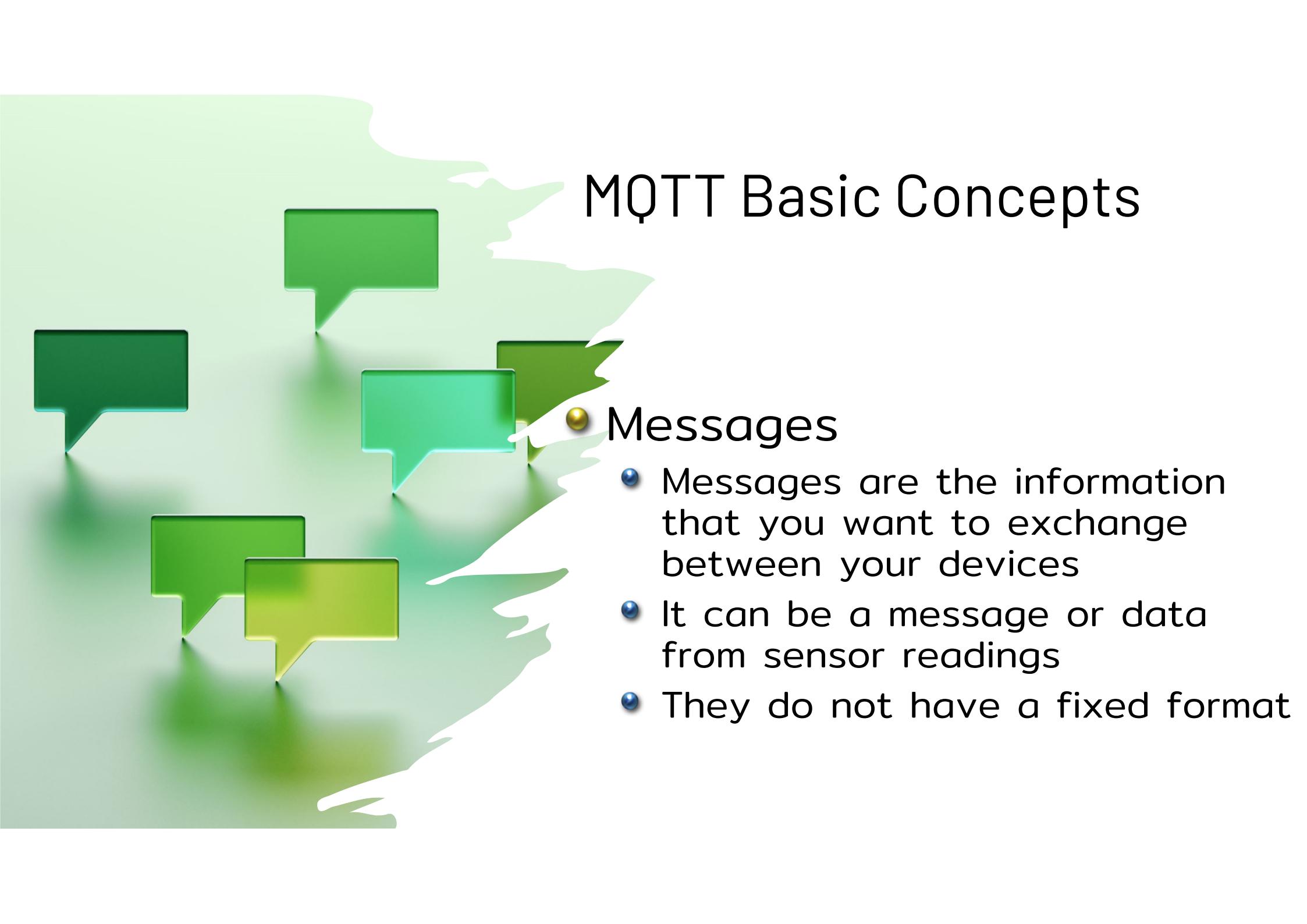
[www.FormaServe.co.uk](http://www.FormaServe.co.uk)

## MQTT Terms

- Pub & Sub
- Messages
- Topics
- Broker

# MQTT Basic Concepts

- Pub & Sub
    - MQTT works as a publish & subscribe system
    - Devices publish messages on a specific topic
    - All devices that are subscribed to that topic, receive the message
- 



# MQTT Basic Concepts

## Messages

- Messages are the information that you want to exchange between your devices
- It can be a message or data from sensor readings
- They do not have a fixed format



# MQTT Basic Concepts

## Topics

- The way you register interest for incoming messages or how you specify where you want to publish the message
- Topics are represented with strings separated by a forward slash
  - company/office1/temperature
- Case sensitive!



# MQTT Basic Concepts

---

## Broker

- A central hub, allows clients to talk to each other
- Can be managed or self-hosted
- Receives all messages
- Publishes the message to all clients
- There are many Internet MQTT brokers you can use
  - Mosquitto - Most popular
  - Adafruit.io - Great dashboard

# IBM i as MQTT Broker?



A screenshot of the Formaserve Open Source Package Management interface. It shows a list of installed packages. The package "activemq" is highlighted with a red border. The interface has a blue header bar with icons for Open Source Package Management, File, View, Connection, Utilities, and a search bar. Below the header is a navigation bar with tabs for Installed packages, Updates available, and Available packages. The main area displays a table of packages with columns for Package, Version, and Release.

A screenshot of an SSH session titled "SSH Galatea". The session shows the command "activemq start" being run, followed by its output: "INFO: Loading '/QOpenSys/pkg/lib/activemq//bin/env'" and "INFO: Using java '/QOpenSys/QIBM/ProdData/JavaVM/jdk80/64bit/bin/java'". A blue arrow points from the text "Process with pid '271199' is already running" to the status command "activemq status" which is then run, showing "ActiveMQ is running (pid '271199')". The session ends with "andy@GALATEA:~\$".

# IBM i as MQTT Broker?

The screenshot shows the ActiveMQ Administration Console interface. The URL in the browser bar is `galatea:8161/admin/topics.jsp;jsessionid=node0wqyhc9tnz3847ajnfxlta1j5.node0`, with the port number 8161 highlighted by a red box.

The main content area displays the "Topics" section. A large blue watermark reading "powerwire.eu/activemq" is overlaid across the middle of the page. The table lists several topics:

Name	Number Of Consumers	Messages Enqueued	Messages Dequeued	Operations
ActiveMQ.Advisory.Connection	0	42	0	<a href="#">Send To Active Subscribers</a> <a href="#">Active Producers</a> <a href="#">Delete</a>
ActiveMQ.Advisory.Consumer.Topic	0	4	0	<a href="#">Send To Active Subscribers</a> <a href="#">Active Producers</a> <a href="#">Delete</a>
ActiveMQ.Advisory.MasterBroker	0	1	0	<a href="#">Send To Active Subscribers</a> <a href="#">Active Producers</a> <a href="#">Delete</a>
Hackaday.nodered.test	0	8	0	<a href="#">Send To Active Subscribers</a> <a href="#">Producers</a>
iot	0	563	527	<a href="#">Send To Active Subscribers</a> <a href="#">Producers</a>
test	0	95	0	<a href="#">Send To Active Subscribers</a> <a href="#">Producers</a>

At the bottom of the page, the copyright notice reads "Copyright 2005-2015 The Apache Software Foundation."

A dark blue background featuring a complex, glowing network graph composed of numerous small, semi-transparent blue spheres connected by thin white lines, creating a sense of depth and connectivity.

# Storing Images on IBM i



# Storing Images on IBM i

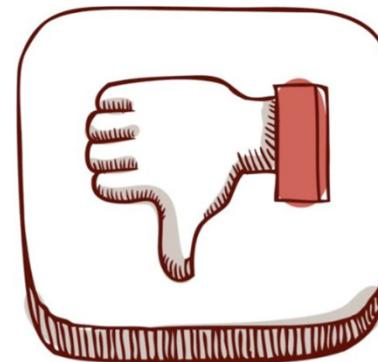
---

- Storing images on the IBM i can be achieved in 2 ways
  - Store the image on the IFS & save a location link in the database
  - Store the image directly in table/file on DB2 for i
- Let us investigate both methods
- One way is no better than the other, we have used both methods in our applications we have developed at FormaServe

# Pro's & Con's

## • **Pro's for storing an IFS Folder**

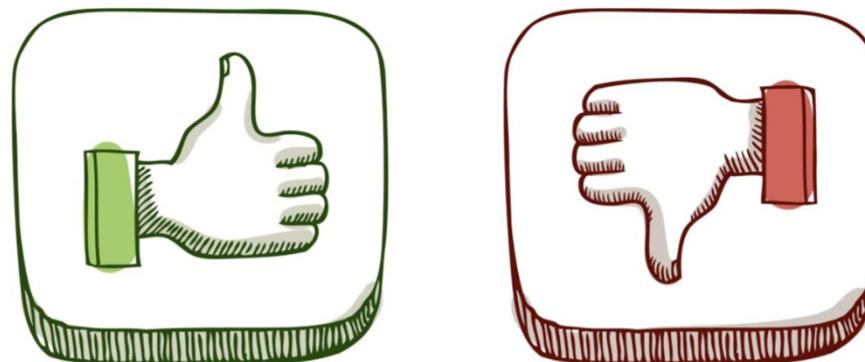
- Separate from database
- Can access without SQL
- Can be manipulated, compressing, etc, without any interference from the database
- Files can be located on any server in your org



# Pro's & Con's

## • **Pro's for storing in the database**

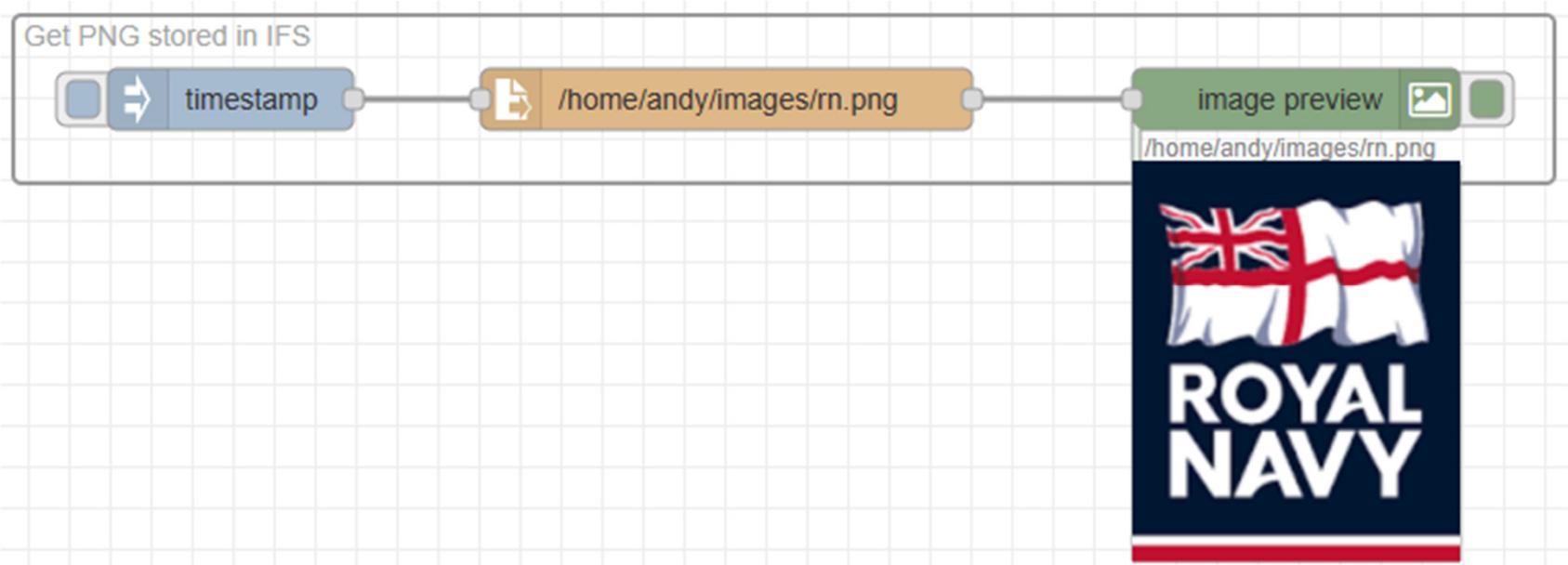
- All in one location
- Easy for save/restore procedures
- One SQL statement retrieves everything
- Makes DR easier if mirroring to another server



# Files in IFS Demo

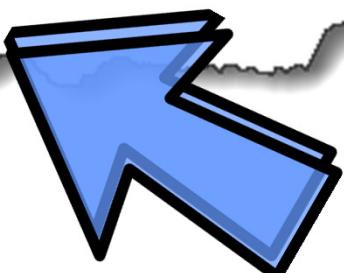
---

- Preview an image stored on the IFS
- A Node-RED flow would look like this



# Storing Images on the IFS

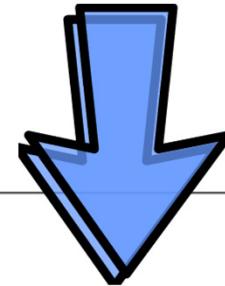
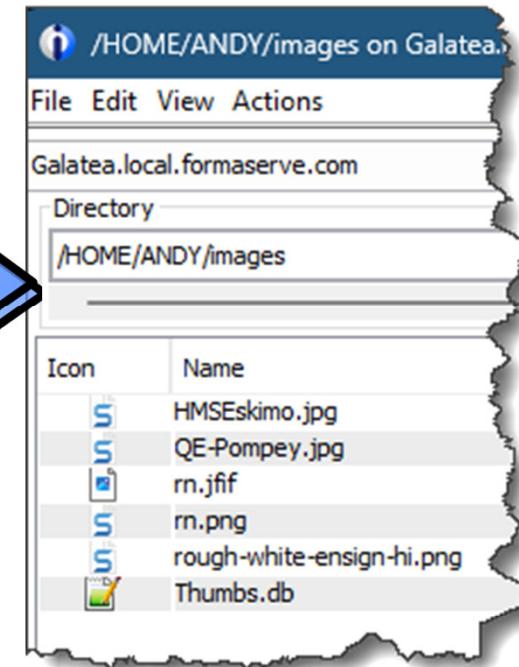
```
Create Table Imagesifs (
    Photoname Varchar(100),
    Photolocation Varchar(100)
);
```



- Images are stored in any file location on the IFS
- A simple table would then hold the location of that photograph image
- The image would not reside within the database
- Our demo coming up will use this method
- A table to hold the location of an image would look something like this

# Storing images on the IFS

- Our IFS directory contains images
- The following table is used to store the file location of any images that are found in the IFS



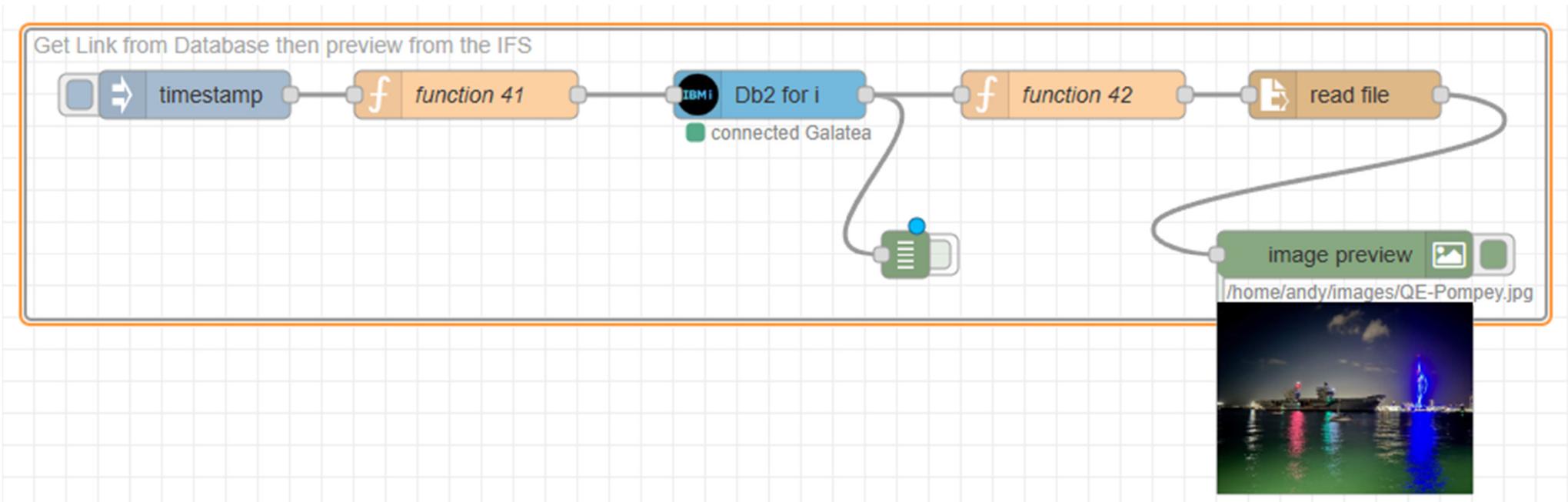
```
-- what images have we got?  
select * from imageLink ;
```

LINK	OBJECT_TYPE	DATA_SIZE	OWNER	CHANGED_ON	LINKID
/home/andy/images/rn.png	*STMF	57241	ANDY	2020-05-17 16:53:37	1
/home/andy/images/rough-white-ensign-hi.png	*STMF	41742	ANDY	2020-05-05 15:39:54	2
/home/andy/images/HMSEskimo.jpg	*STMF	204157	ANDY	2023-02-22 17:45:49	3
/home/andy/images/QE-Pompey.jpg	*STMF	59136	ANDY	2022-12-13 18:50:17	4

# Getting Link for Image

---

- In this Node-RED flow, we use DB2 to get the link for the image & then read the file before previewing it



# Storing Images within DB2 for i

DB2 for i allows for large objects to be stored directly into a file

There are SQL 3 column types we can use for storing images

**BLOB**

Binary Large Object

**CLOB**

Character Large Object

**DBCLOB**

Double-Byte Character Large Object



# Storing Images on DB2 for i - LOBs

- BLOB data is not associated with any character set & it is treated as a stream of bytes
- CLOB data is stored in the database character set, & it supports single-byte
- All large object types have a maximum size of 2 Gigabytes
- You define your maximum size when creating the table



# Storing Images on DB2 for i

- To create a table that has a column that can contain photos, or images, it must be created with a column as a blob
- Its size can be up to 2 Gigabytes
- A simple table to hold photographs would be defined as below

```
Create Table Images (
    Photoname Varchar(100),
    Image Blob(2G)
);
```

# Demo of Using Blobs

---

- In IBMs demo database there is a table called Emp\_photo which holds a photo of an employee
- This photo is held in the column called Picture
- Picture is defined as a BLOB with a size limit of 100k
- The layout of this table is shown here



The screenshot shows a SQL editor window titled "Run SQL Scripts - Galatea.local.formaserve.com(Galatea)". The code pane contains the following SQL:

```
1 set schema iot ;
2
3 select * from emp_photo;
```

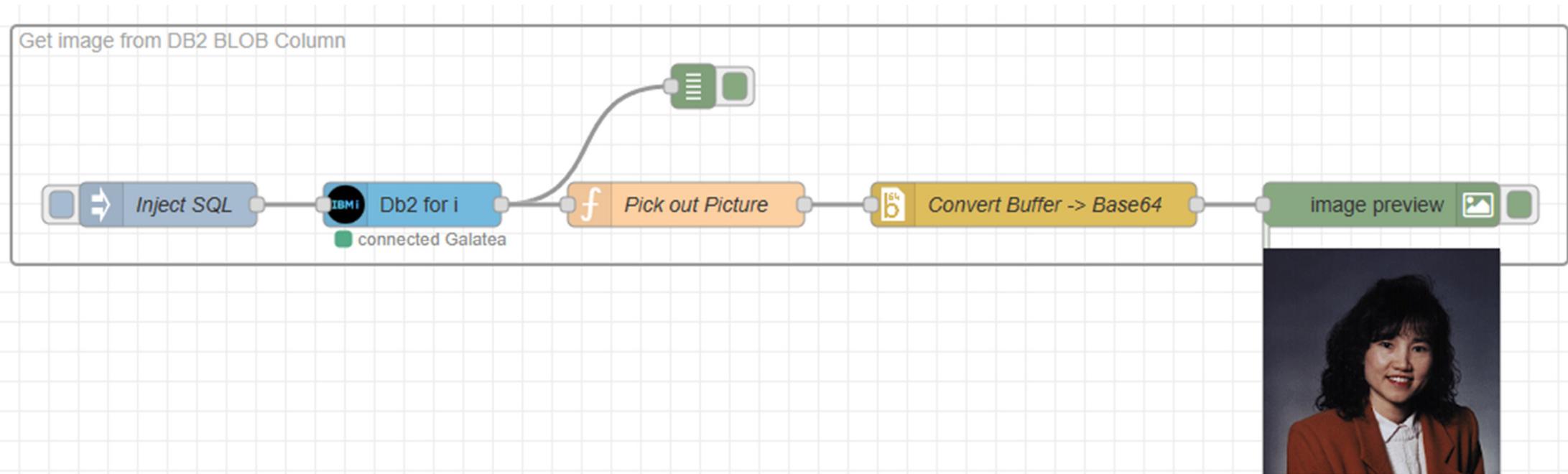
The results pane shows the table structure:

All Columns	
Empno	CHAR(6)
Photo_Format	VARCHAR(10)
<b>Picture</b>	<b>BLOB(100 KB)</b>
Emp_Rowid	CHAR(40)
Dl_Picture	DATALINK(1000)

A large blue arrow points from the text "The layout of this table is shown here" towards the screenshot.

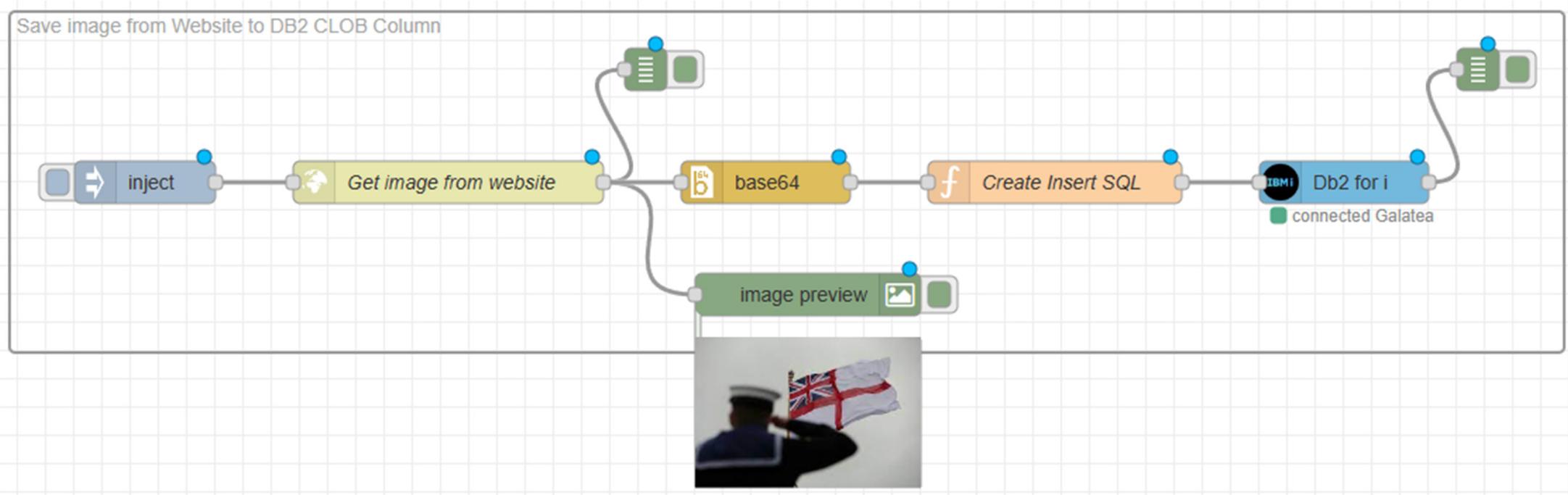
# Demo of Using DB Blobs

- Using Node-RED we retrieve a BLOB column from DB2 for i & preview the data



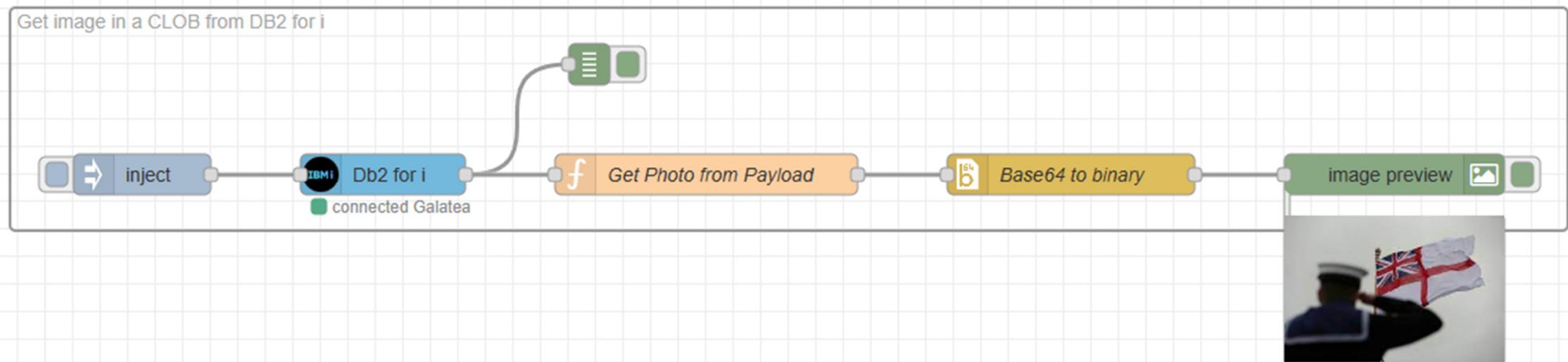
# Demo of using DB CLOBs

- Storing images in DB2 for i CLOB column



# Demo of using DB CLOBs

- Retrieving images in DB2 for i CLOB column





Pico W



Raspberry Pi



IBM i





# IOT to IBM i Example - Intruders!

- We have a Pico W with a PIR sensor
- If it detects any movement,
  - It publishes a message to a MQTT broker (Our IBM i)
  - Red LED gives visual indication of movement
- The Raspberry Pi is subscribed to the Pico topic
- If it receives a message from that topic, it will
  - Take a photo
  - Publish the photo to a different topic
  - Send an alert email
  - Red LED gives visual indication of movement
- IBM i is subscribed to the Raspberry Pi topic
- If a message is received, it
  - Stores the photo on the IFS
  - Writes an entry to an audit log of the photo location - DB2 for i
- IBM i has a node.js website showing all intruder events
  - Clicking on an event shows the photo that was taken by the Pi

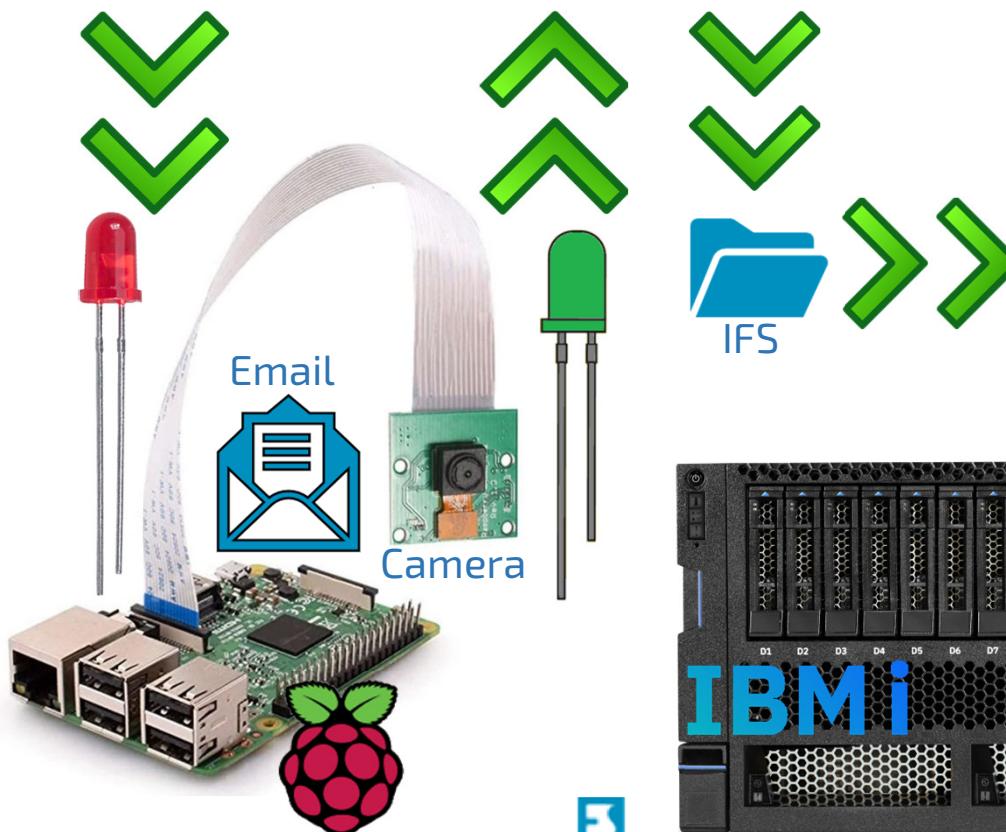
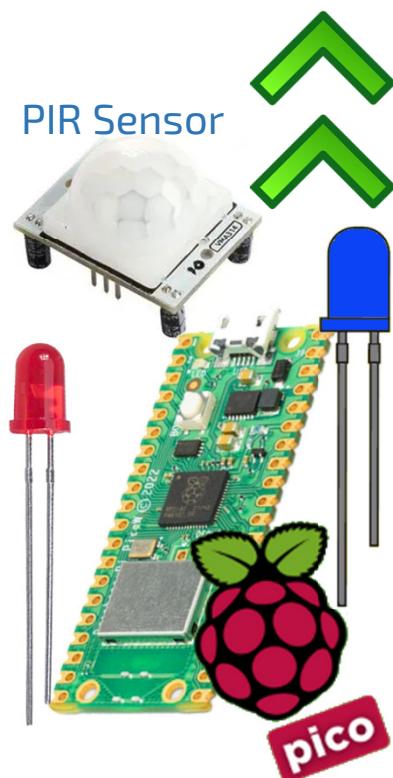


# From Pico to Pi to IBM i

IBM i



IBM i

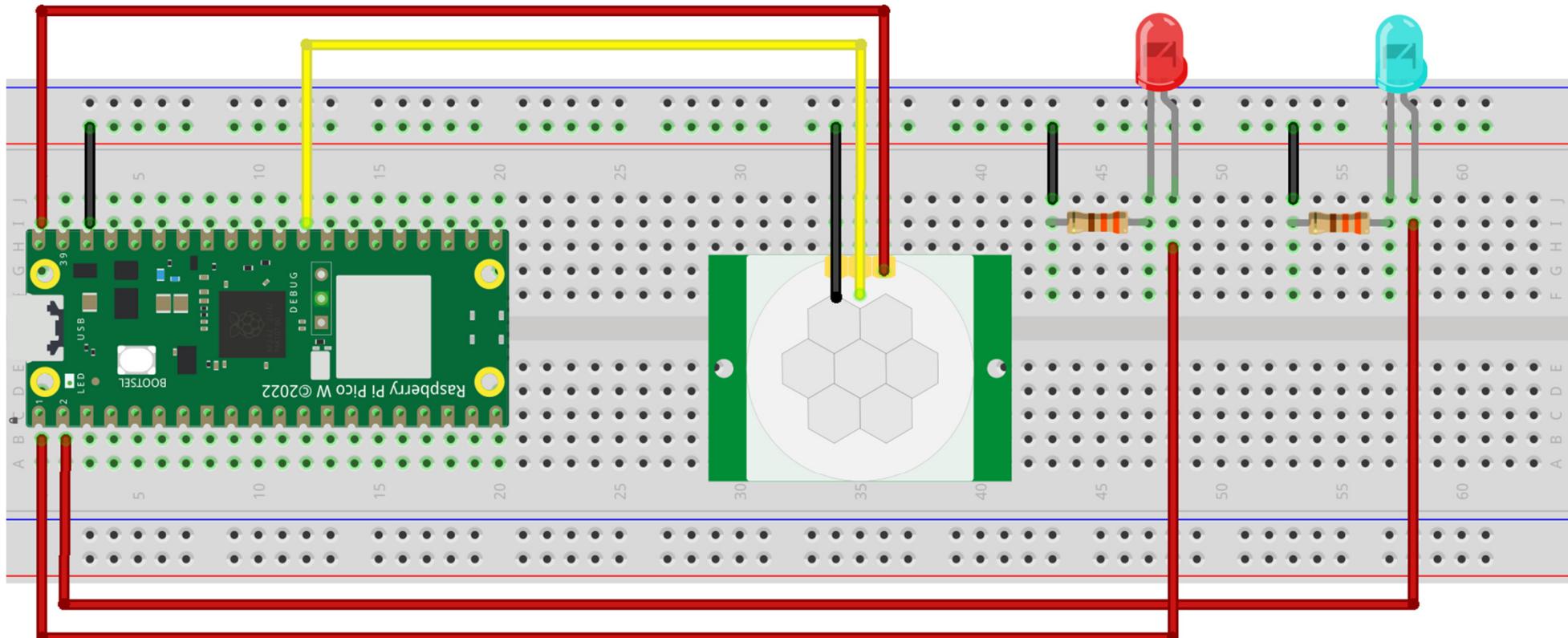


A screenshot of an IBM i application interface titled "f\_Movement - Intruder Access". It shows a table with company information and a list of access logs. The table includes columns for ID, Photo Name, Photo Location, Access Date, and Access Time. The logs show four entries from September 1st, 2023.

ID	Photo Name	Photo Location	Access Date	Access Time
47	20230901T120945658Z.png	/home/andy/movement/	1st Sep 2023	13.09
46	20230901T120916049Z.png	/home/andy/movement/	1st Sep 2023	13.09
45	20230901T120913364Z.png	/home/andy/movement/	1st Sep 2023	13.09
44	20230901T120909215Z.png	/home/andy/movement/	1st Sep 2023	13.09



# Pico W Wiring

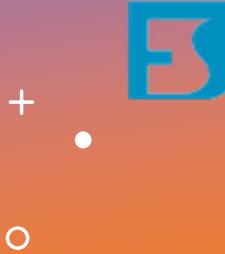


fritzing

[www.FormaServe.co.uk](http://www.FormaServe.co.uk)



# Pico W Code - Micro Python



```
# connect to WIFI & switch BLUE LED on
def wifi_connect():

    ledBLUE.value(False)
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.connect(ssid, password)

    while wlan.isconnected() == False:
        print('Waiting for connection...')
        sleep(1)

    ip = wlan.ifconfig()[0]
    print(f'Connected to WiFi, IP address is {ip}')
    ledBLUE.value(True)

    return ip
```

```
# mqtt stuff
def mqtt_connect():

    client = MQTTClient(client_id, mqtt_server, keepalive=3600)
    client.connect()
    print(f'Connected to {mqtt_server}, our MQTT Broker running on IBM i')
    print(f'MQTT Publishing on queue {topic_pub}')

    return client

# continous loop to detect movement
while True:

    # we got an intruder?
    if pir.value() == 1:
        # increment counter
        n +=1
        print(f"Movement detected {(n)} - Red LED On")
        ledRED.value(True)
        # send movement message to MQTT server
        client.publish(topic_pub, topic_msg)
        print('Movement published to MQTT Broker')
```

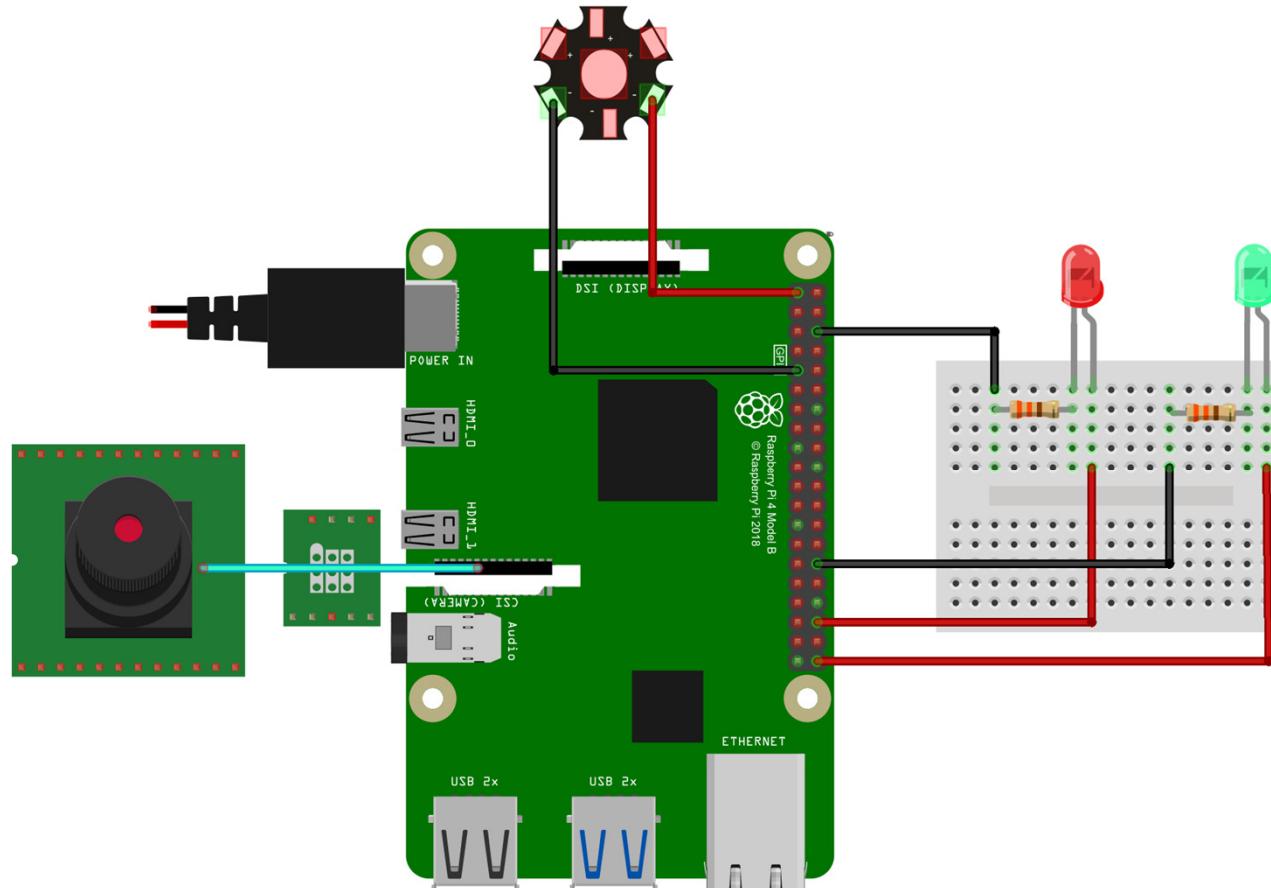
# Raspberry Pi Wiring



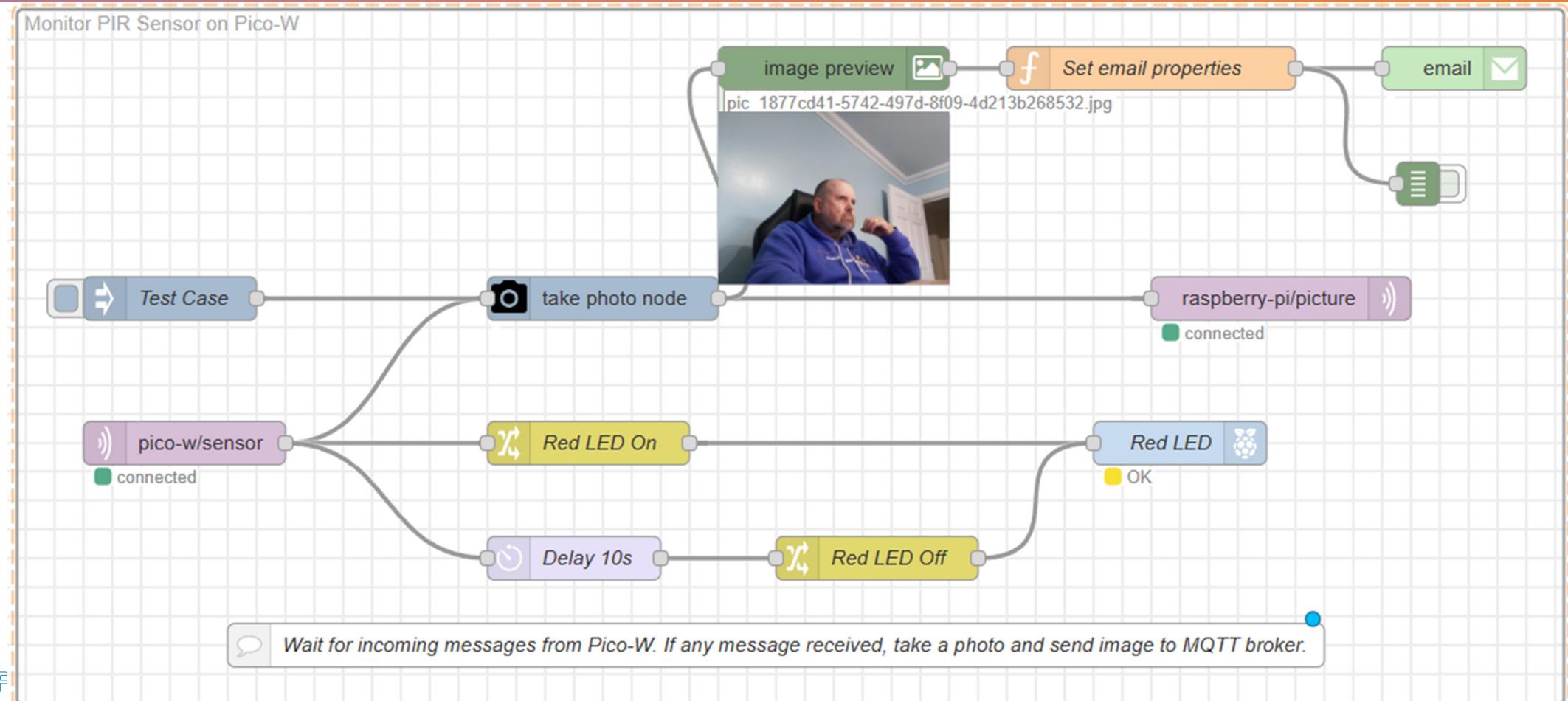
+

.

o



# Raspberry Pi Code

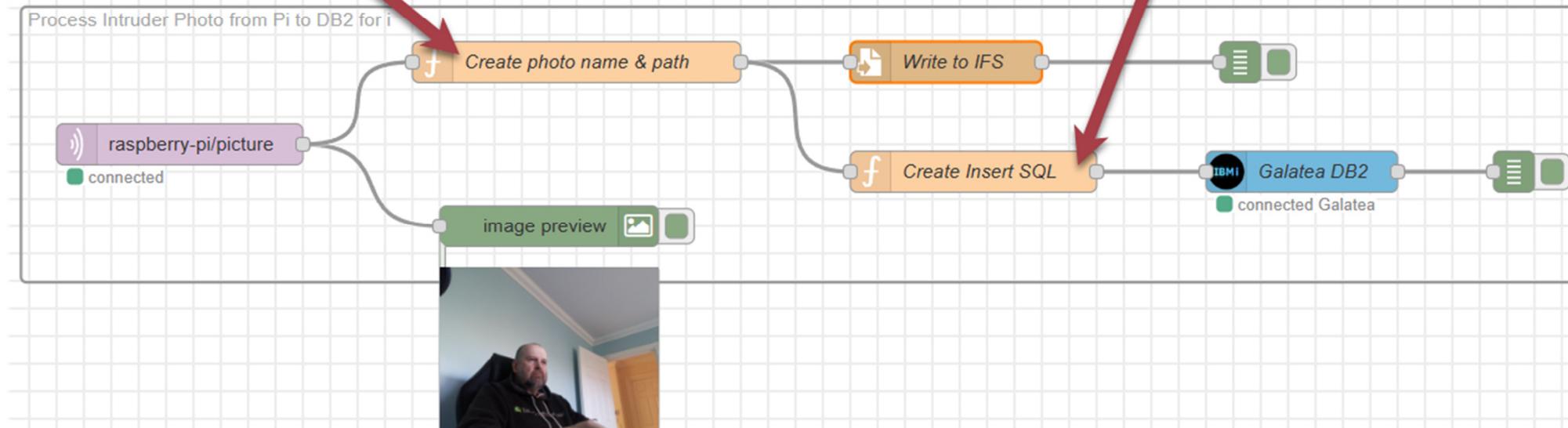


# IBM i - Part 1

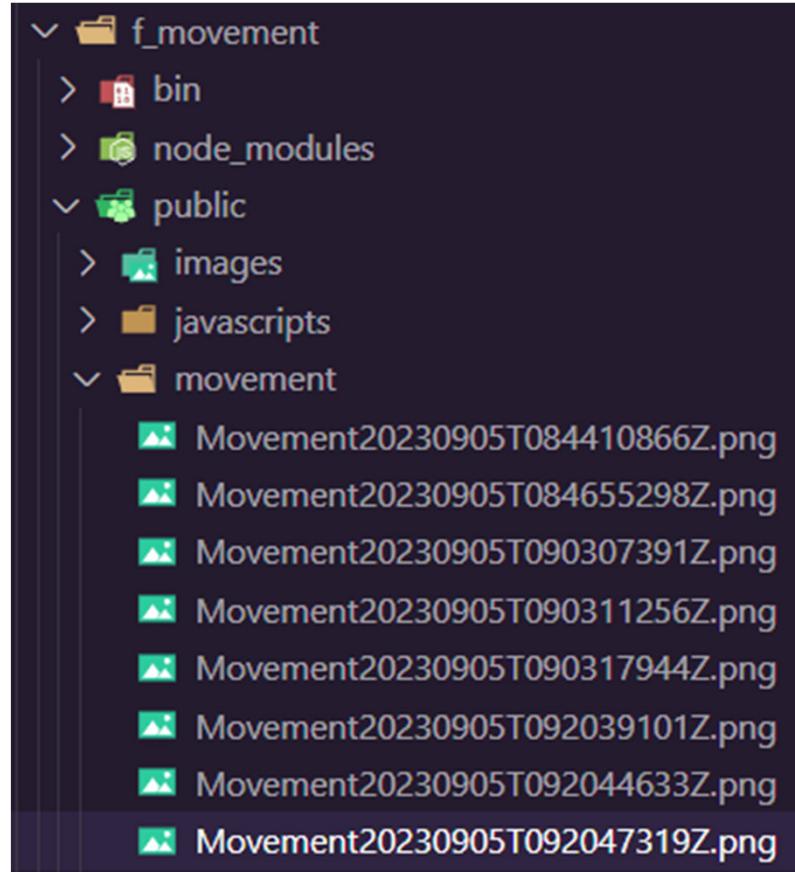
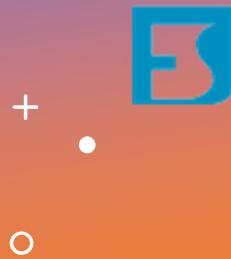


```
msg.file = '/home/andy/f_movement/public/movement/' + Movement + '.png'
```

```
// build SQL statement  
const sql = `insert into iot.MoveIFS(photoname, photolocation)  
values('${msg.file}', '${msg.path}')`
```



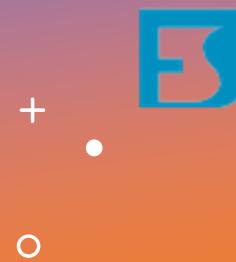
# IBM i - Part 1



## Storing images on the IFS

- Our IFS directory contains images

# IBM i - Part 1



Contents of IOT.MOVEIFS - Galatea.local.formaserve.com(Galatea)

File View

ID	PHOTONAME	PHOTOLOCATION	ADD_TS
170	20230912T201339679Z.png	/home/andy/f_moveme...	2023-09-12 21:13:39.687975
171	20230912T201348800Z.png	/home/andy/f_moveme...	2023-09-12 21:13:48.803589
172	20230912T201352370Z.png	/home/andy/f_moveme...	2023-09-12 21:13:52.379223
173	20230912T201418442Z.png	/home/andy/f_moveme...	2023-09-12 21:14:18.450044
174	20230912T201434406Z.png	/home/andy/f_moveme...	2023-09-12 21:14:34.416365
175	20230912T201438552Z.png	/home/andy/f_moveme...	2023-09-12 21:14:38.556440
176	20230912T202624916Z.png	/home/andy/f_moveme...	2023-09-12 21:26:24.924670
177	20230912T202818840Z.png	/home/andy/f_moveme...	2023-09-12 21:28:18.844874
178	20230912T202823898Z.png	/home/andy/f_moveme...	2023-09-12 21:28:23.903800
179	20230912T202913615Z.png	/home/andy/f_moveme...	2023-09-12 21:29:13.620342
180	20230913T002951095Z.png	/home/andy/f_moveme...	2023-09-13 01:29:51.123579
181	20230913T003019198Z.png	/home/andy/f_moveme...	2023-09-13 01:30:19.283912
182	20230913T003218573Z.png	/home/andy/f_moveme...	2023-09-13 01:32:18.591066
183	20230913T003221608Z.png	/home/andy/f_moveme...	2023-09-13 01:32:21.623120
184	20231103T165455583Z.png	/home/andy/movement/	2023-11-03 16:54:56.138522
185	20231106T092231170Z.png	/home/andy/movement/	2023-11-06 09:22:31.370380
186	20231106T145635115Z.png	/home/andy/movement/	2023-11-06 14:56:35.161861
187	20231106T150218575Z.png	/home/andy/f_moveme...	2023-11-06 15:02:18.617275

Done: 187 rows retrieved.

Columns... Save Results... Refresh

## Storing images on the IFS

- The following table is used to store the file location of any images that are found in the IFS



## f\_Movement - Intruder Access

Company: FormaServe

Date: 10th Sep 2023

Show 10 entries

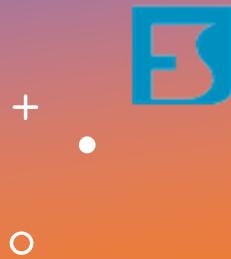
Search:

Open	ID	Photo Name	Photo Location	Access Date	Access Time
<a href="#">Click Me</a>	187	20231106T150218575Z.png	/home/andy/f_movement/public/movement/	6th Nov 2023	15.02
<a href="#">Click Me</a>	186	20231106T145635115Z.png	/home/andy/movement/	6th Nov 2023	14.56
<a href="#">Click Me</a>	185	20231106T092231170Z.png	/home/andy/movement/	6th Nov 2023	09.22
<a href="#">Click Me</a>	184	20231103T165455583Z.png	/home/andy/movement/	3rd Nov 2023	16.54
<a href="#">Click Me</a>	183	20230913T003221608Z.png	/home/andy/f_movement/public/movement/	13th Sep 2023	01.32
<a href="#">Click Me</a>	182	20230913T003218573Z.png	/home/andy/f_movement/public/movement/	13th Sep 2023	01.32
<a href="#">Click Me</a>	181	20230913T003019198Z.png	/home/andy/f_movement/public/movement/	13th Sep 2023	01.30
<a href="#">Click Me</a>	180	20230913T002951095Z.png	/home/andy/f_movement/public/movement/	13th Sep 2023	01.29
<a href="#">Click Me</a>	179	20230912T202913615Z.png	/home/andy/f_movement/public/movement/	12th Sep 2023	21.29
<a href="#">Click Me</a>	178	20230912T202823898Z.png	/home/andy/f_movement/public/movement/	12th Sep 2023	21.28

Showing 1 to 10 of 187 entries

Previous [1](#) [2](#) [3](#) [4](#) [5](#) ... [19](#) Next

# IBM i - Part 2

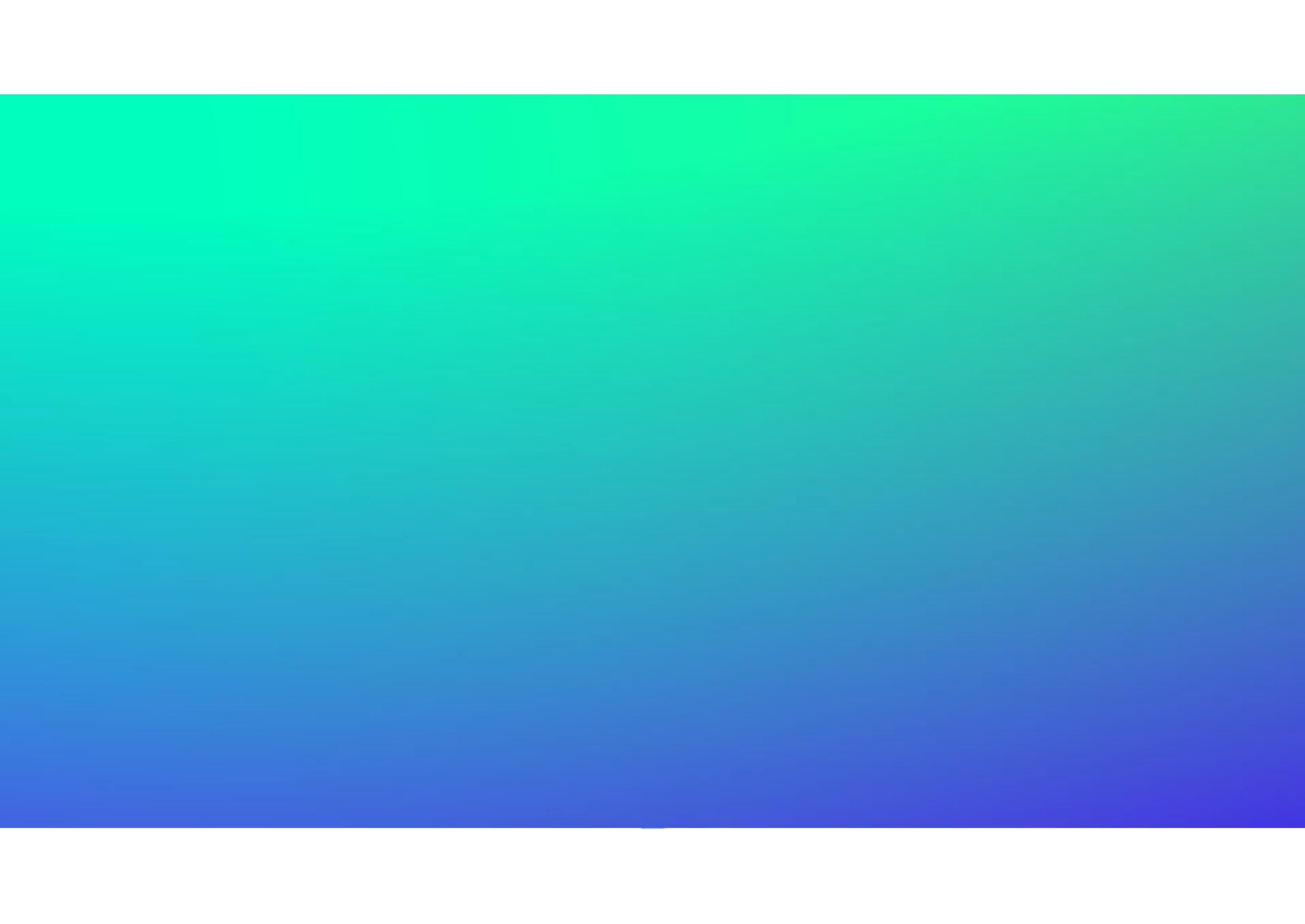


```
async function execMovement() {
  debug(' → Into execMovement')
  let schema = 'IOT'
  let sql = `Select * from ${schema}.MoveIFS order by ID desc;`
  debug(`SQL: ${sql}`)
  const connection = new Connection({ url: '*LOCAL' })
  const statement = new Statement(connection)
  const movement = await statement.exec(sql)
  debug(`SQL Results: ${JSON.stringify(movement, null, 2)}`)
  res.render('index', {
    company: 'FormaServe',
    moment,
    formatCurrency,
    opts,
    title,
    today,
    output: movement,
  })
}
```

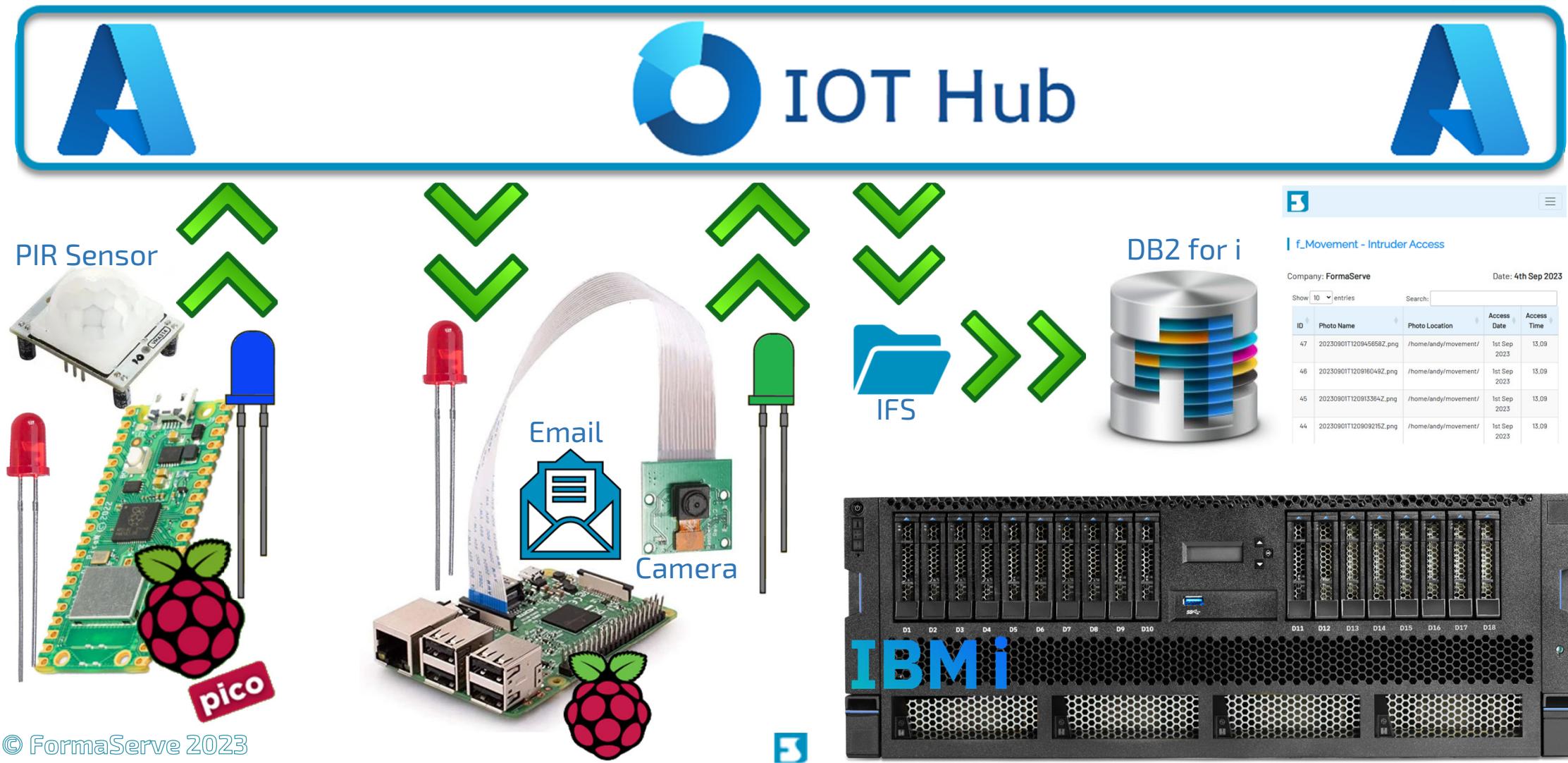
## Display Intruder Results

- The following table is used to store the file location of any images that are found in the IFS





# From Pico to Pi to IBM i



© FormaServe 2023



# Any Questions?





# Thank You!

## Andy Youens

Andy@FormaServe.co.uk

[www.formaserve.co.uk](http://www.formaserve.co.uk)

[learning.formaserve.co.uk](http://learning.formaserve.co.uk)

01908 109978

© FormaServe Systems Ltd

