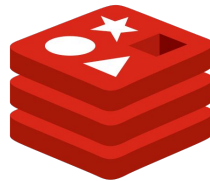
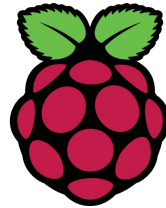


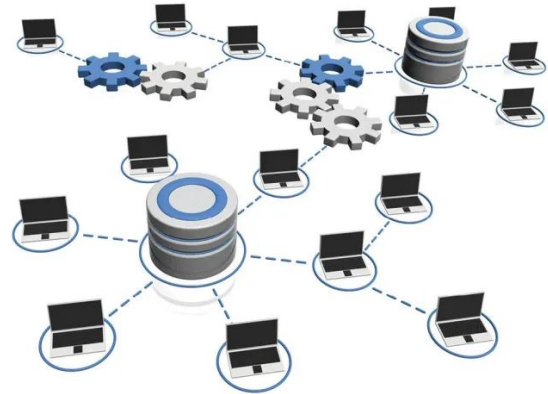
Skill Mapping IoT Workshop

Create an IoT Device Worker from RPi and Connect to THOpenSCADA via REDIS

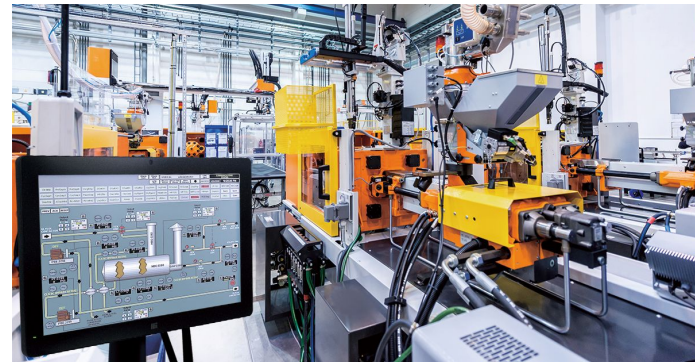
Nattapong Wattanasiri



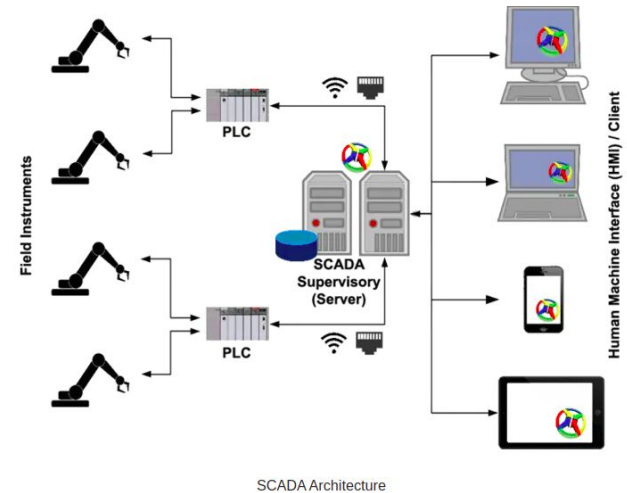
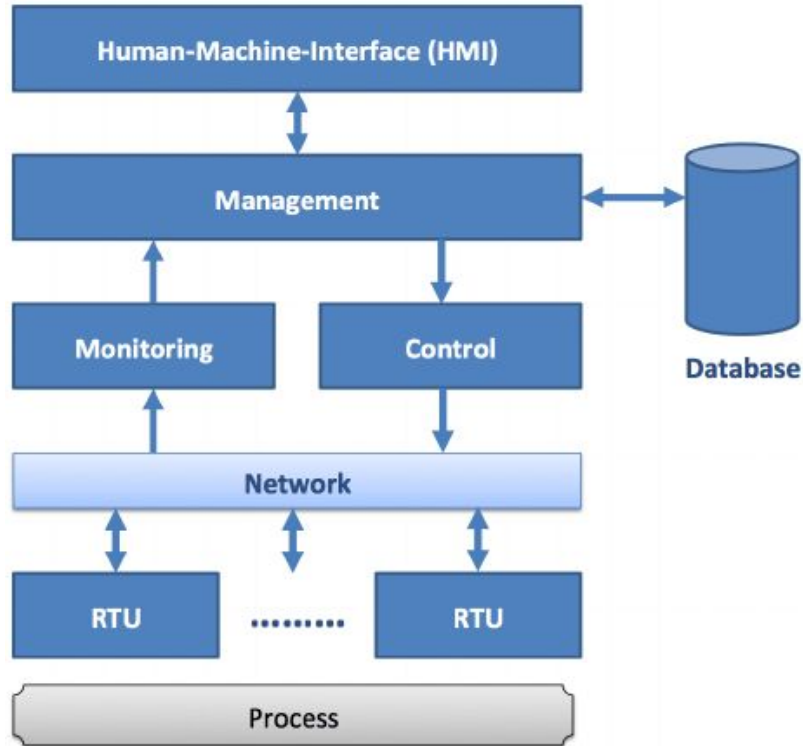
redis



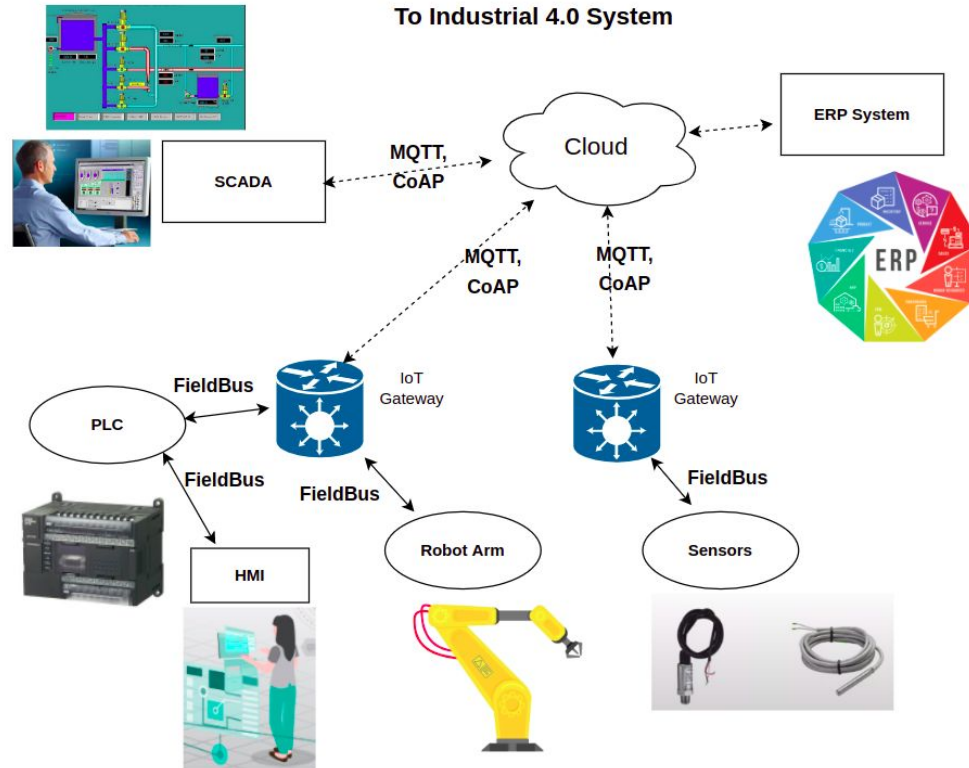
What is SCADA?



SCADA Architecture



IoT Device in SCADA



Device worker?

- Software for running the device in specific task.
- Data Acquisition
- Writing them to the database

Database for IoT Device?

- Sensor data: time series data in multiple dimensions
- Logging

Column-Family



influxdb

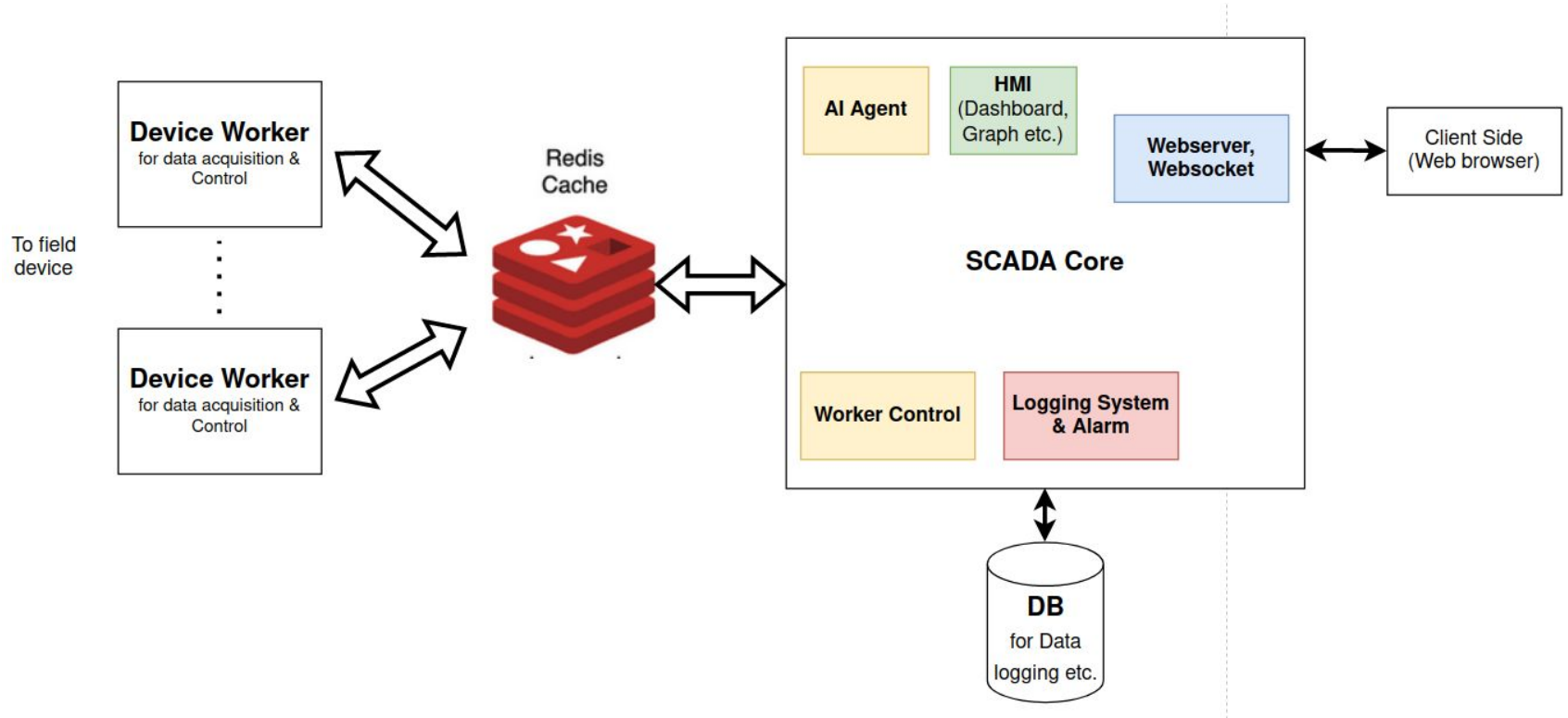


CrateDB



mongoDB

THOpenSCADA



What is Redis? And Why?

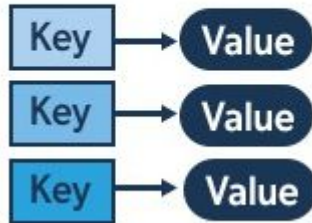
- **In-memory** data structures (**On RAM**)
- key-value DB
- Very fast

The stream data type enables high-rate data ingestion, messaging, event sourcing, and notifications in SCADA



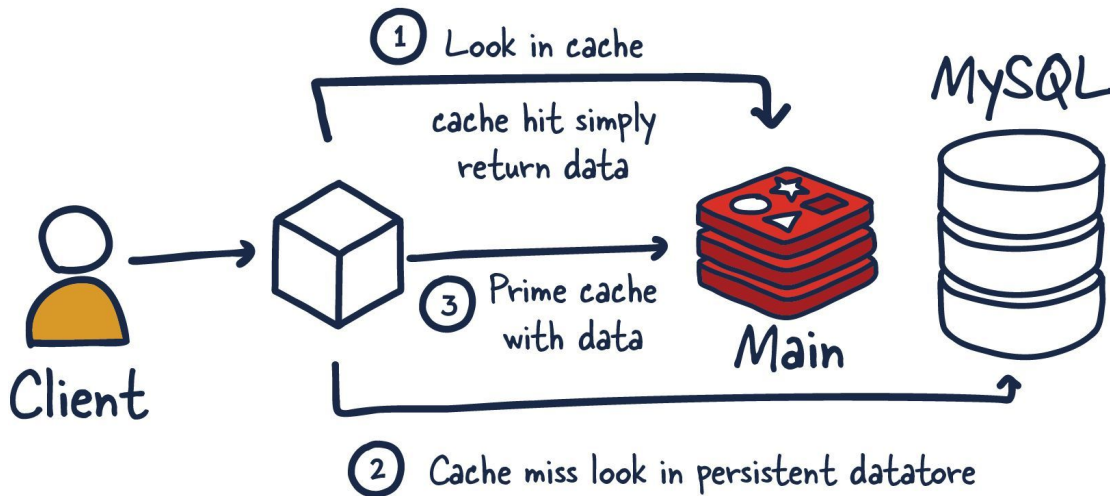
redis

Key-Value



Redis use case

How is redis traditionally used

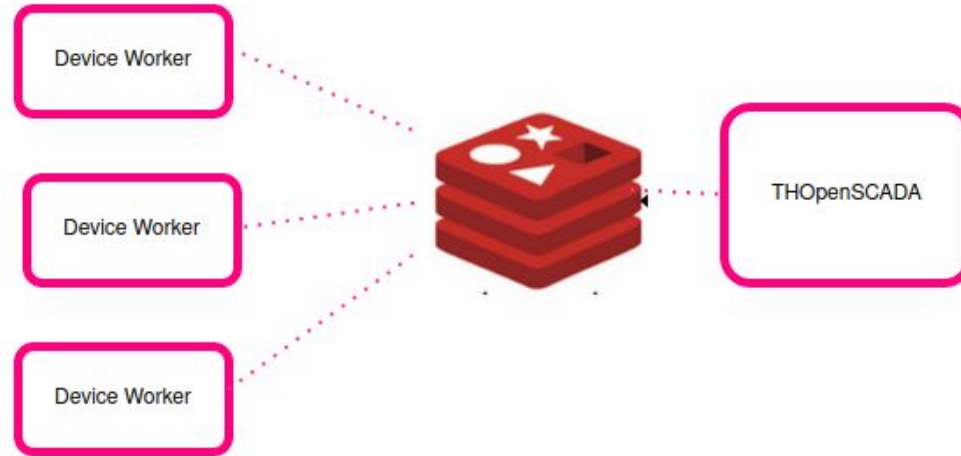


Redis use case



Objective

- Create joystick device worker on RPi (From previous workshop)
- Connect to THOpenSCADA via Redis

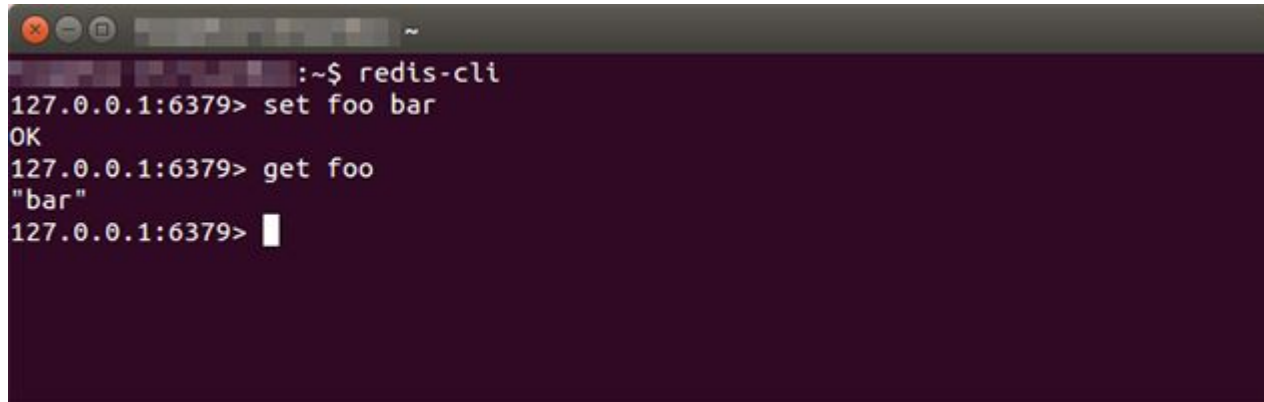


Install Redis-cli

\$ sudo apt update

\$ sudo apt install redis-tools

\$ sudo apt install libevent-dev

A terminal window with a dark purple background and white text. The window title bar shows standard Linux window controls (close, minimize, maximize) and a tilde symbol (~). The terminal content shows a user at a shell prompt (~\$) running 'redis-cli'. The prompt changes to '127.0.0.1:6379>'. The user enters 'set foo bar', and the terminal outputs 'OK'. Then the user enters 'get foo', and the terminal outputs '"bar"'. Finally, the user enters another prompt '127.0.0.1:6379>' with a cursor at the end.

```
~$ redis-cli
127.0.0.1:6379> set foo bar
OK
127.0.0.1:6379> get foo
"bar"
127.0.0.1:6379> 
```

Install C Redis Library

```
$ git clone https://github.com/redis/hiredis.git
```

```
$ cd path/to/hiredis
```

```
$ make
```

```
$ make install
```

Install C Redis Library

Add this line to file /etc/ld.so.conf

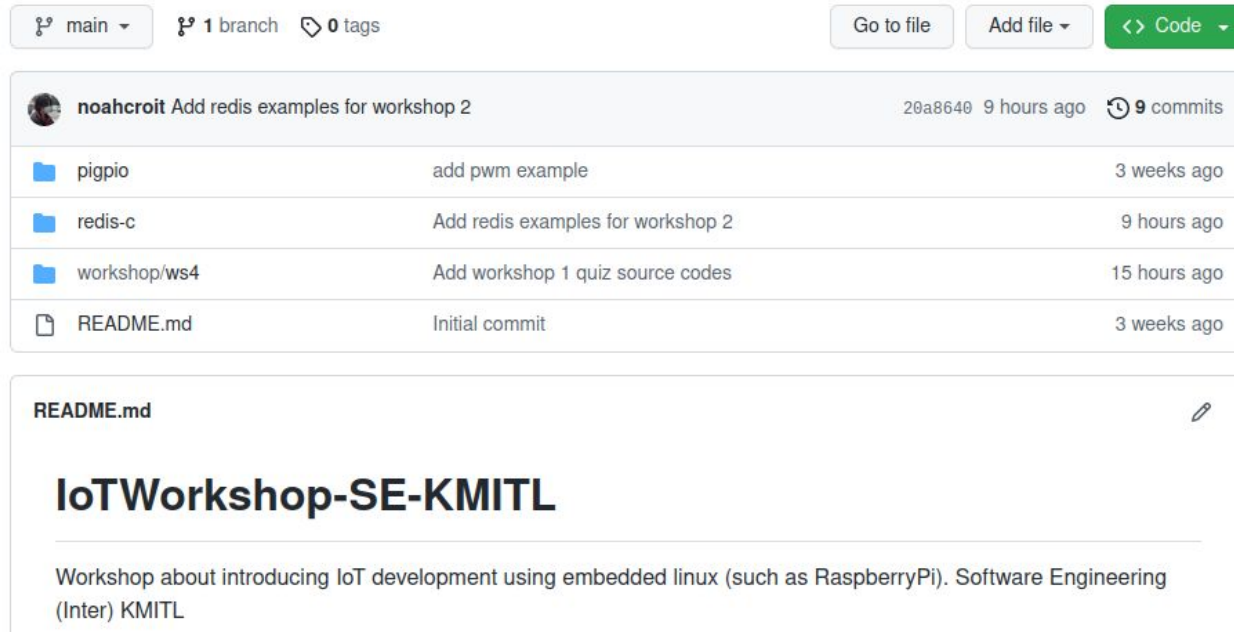
```
noah-croit@noahcroit-desktop:~/Workspace/ictlab/smartmeter$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf
include /usr/local/lib/
include /usr/include
include /usr/lib
```

Then, run

```
$ sudo /sbin/ldconfig
```

Download source files for workshop

```
$ git clone https://github.com/noahcroit/loTWorkshop-SE-KMITL.git
```



The screenshot shows the GitHub interface for the repository 'noahcroit/loTWorkshop-SE-KMITL'. At the top, there are navigation buttons: 'main' (selected), '1 branch', and '0 tags'. To the right are buttons for 'Go to file', 'Add file', and a green 'Code' button. Below this is a commit history table with the following entries:

Commit Hash	Author	Message	Time
20a8640	noahcroit	Add redis examples for workshop 2	9 hours ago
		add pwm example	3 weeks ago
		Add redis examples for workshop 2	9 hours ago
		Add workshop 1 quiz source codes	15 hours ago
		Initial commit	3 weeks ago

Below the commit history is the 'README.md' file content. It features the title 'IoTWorkshop-SE-KMITL' in a large, bold font, followed by a horizontal line and the text: 'Workshop about introducing IoT development using embedded linux (such as RaspberryPi). Software Engineering (Inter) KMITL'.

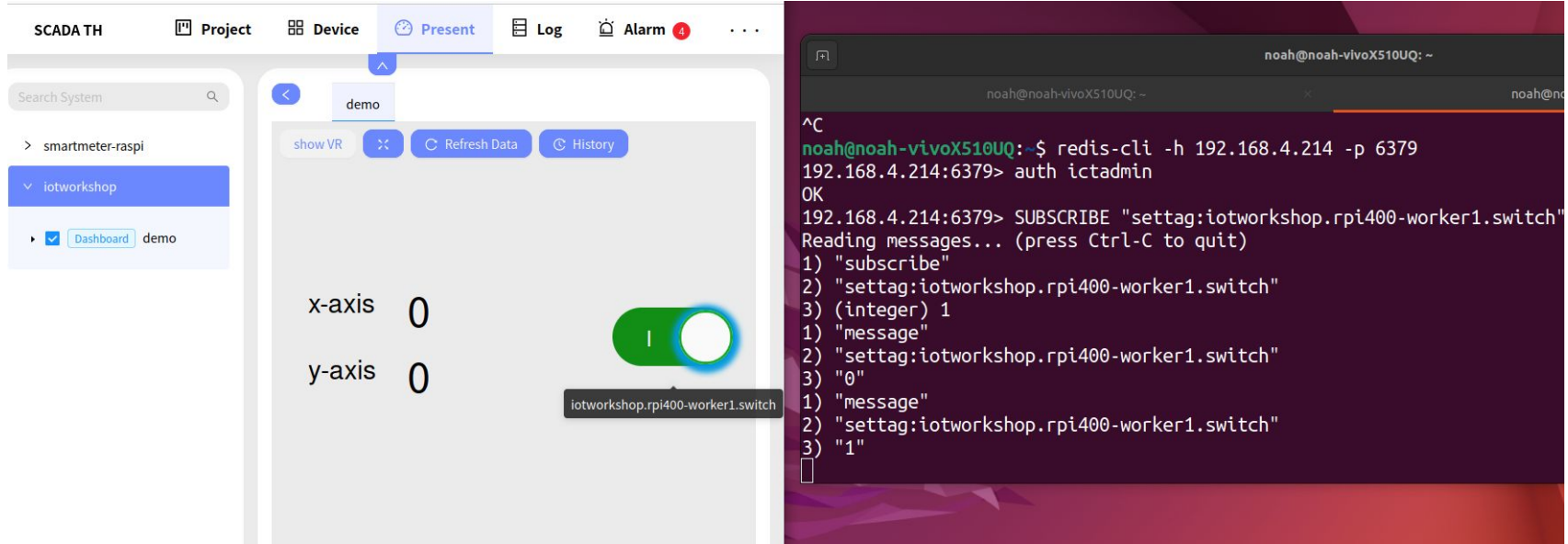
Create system in THOpenSCADA

Test Redis-cli with THOpenSCADA

The screenshot displays the THOpenSCADA web interface. The top navigation bar includes 'SCADA TH', 'Project', 'Device', 'Present' (active), 'Log', 'Alarm', 'AI', 'Workflow', and 'Admin'. On the left, a sidebar shows a search bar and a tree view with 'martmeter-raspi' and 'iotworkshop' (expanded). Under 'iotworkshop', 'Dashboard' is selected, showing a 'demo' view. The main content area features a 'demo' tab and buttons for 'show VR', 'Refresh Data', and 'History'. A terminal window is open, showing the following commands and output:

```
noah@noah-vivoX510UQ: ~  
noah@noah-vivoX510UQ:~$ redis-cli -h 192.168.4.214 -p 6379  
192.168.4.214:6379> auth ictadmin  
OK  
192.168.4.214:6379> set "tag:iotworkshop.rpi400-worker1.joystick-x" 0  
OK  
192.168.4.214:6379> set "tag:iotworkshop.rpi400-worker1.joystick-y" 1  
OK  
192.168.4.214:6379> 
```

Test Redis-cli with THOpenSCADA



The image displays the THOpenSCADA web interface on the left and a terminal window on the right, illustrating the integration of Redis with the SCADA system.

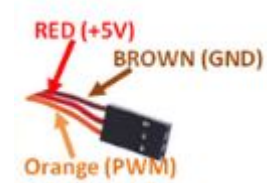
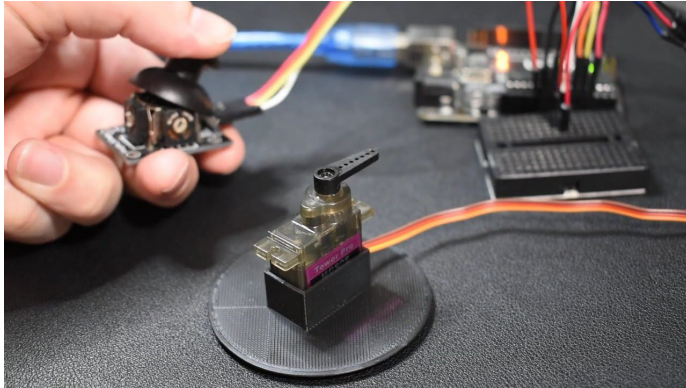
THOpenSCADA Interface:

- Navigation bar: SCADA TH, Project, Device, Present, Log, Alarm (4).
- Search System: Search System (magnifying glass icon).
- Tree view: smartmeter-raspi, iotworkshop (expanded), Dashboard (checked), demo.
- demo panel: show VR, Refresh Data, History buttons. x-axis 0, y-axis 0. A green toggle switch labeled iotworkshop.rpi400-worker1.switch is shown in the 'on' position.

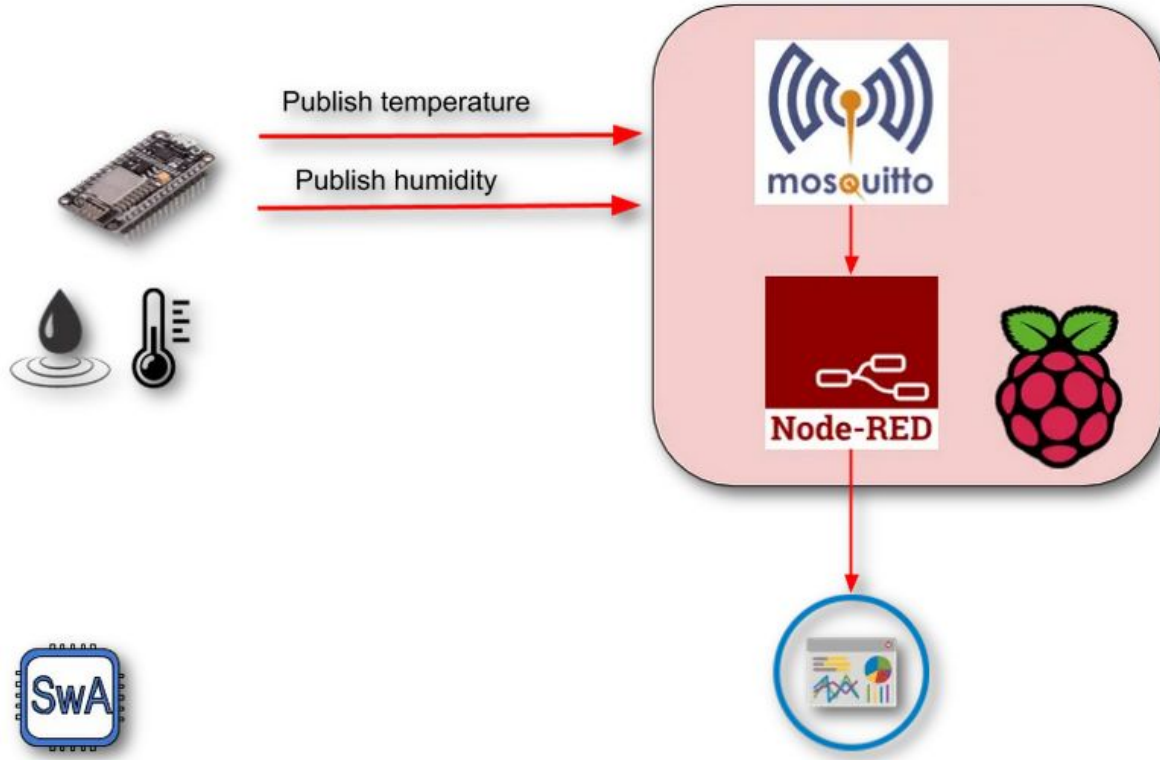
Terminal Output:

```
noah@noah-vivoX510UQ: ~  
noah@noah-vivoX510UQ: ~  
^C  
noah@noah-vivoX510UQ: ~$ redis-cli -h 192.168.4.214 -p 6379  
192.168.4.214:6379> auth ictadmin  
OK  
192.168.4.214:6379> SUBSCRIBE "settag:iotworkshop.rpi400-worker1.switch"  
Reading messages... (press Ctrl-C to quit)  
1) "subscribe"  
2) "settag:iotworkshop.rpi400-worker1.switch"  
3) (integer) 1  
1) "message"  
2) "settag:iotworkshop.rpi400-worker1.switch"  
3) "0"  
1) "message"  
2) "settag:iotworkshop.rpi400-worker1.switch"  
3) "1"  
^C
```

Control servo with joystick program



Create your own SCADA with NodeRED & MQTT



Q&A

