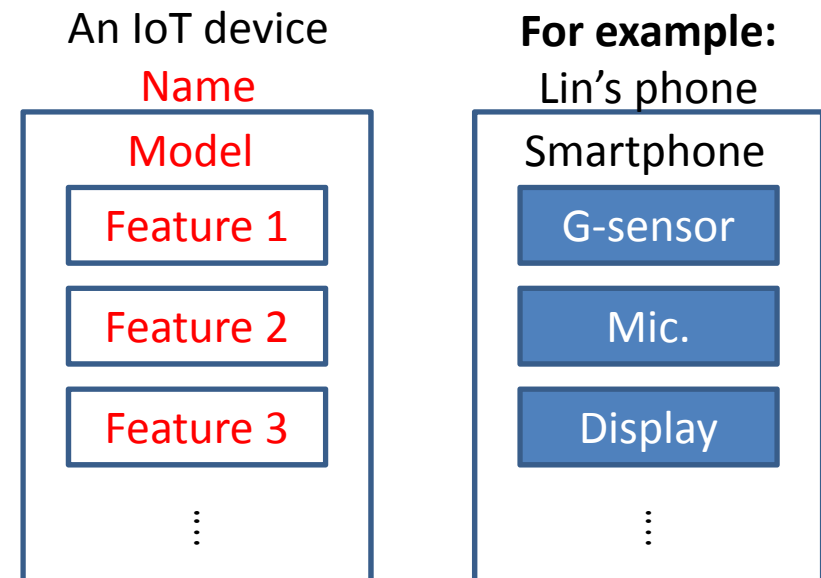
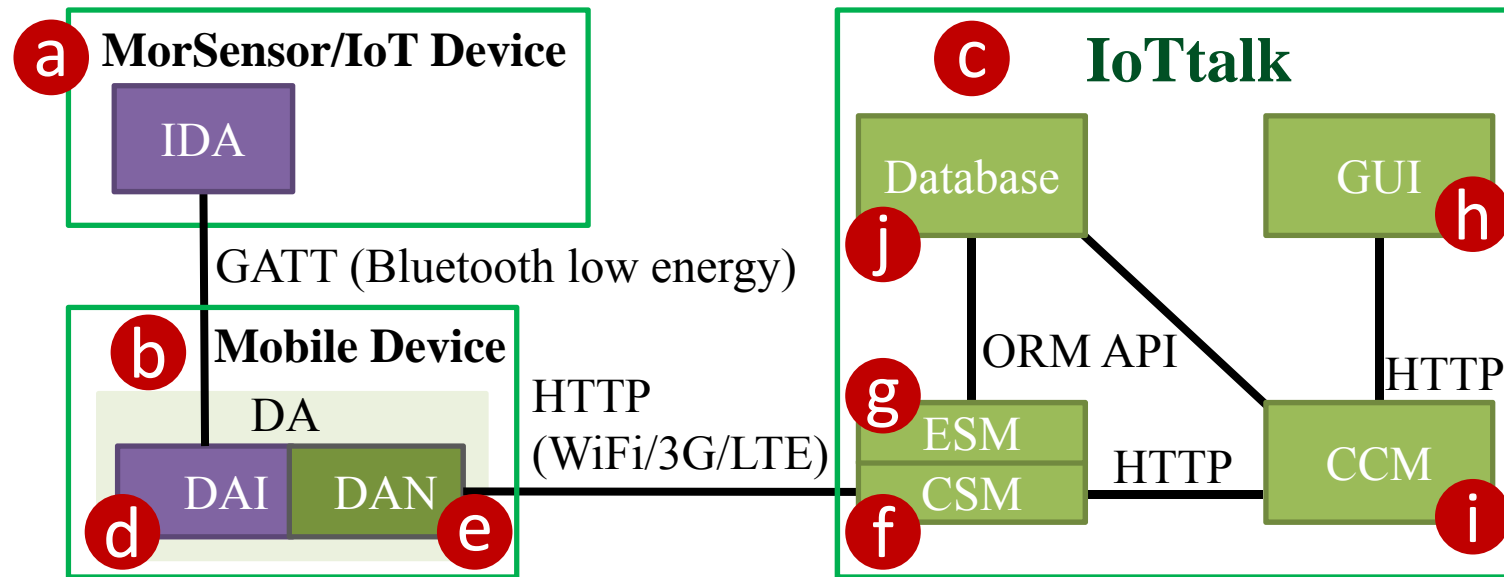


What is IoTtalk?

- **IoTtalk** is an IoT device management tool
- **IoT management concept**
 - **Device Feature**
 - The function or capability which an IoT can provide
 - **Device Model**
 - A set of device features
 - A device model refers to a specific product
 - **Device name**
 - The name of a specific product



System Architecture



IDA: IoT device application

DAI: Device application to IoT device

DAN: Device application to Network

ESM: Execution SubModule system

CSM: Communication SubModule system

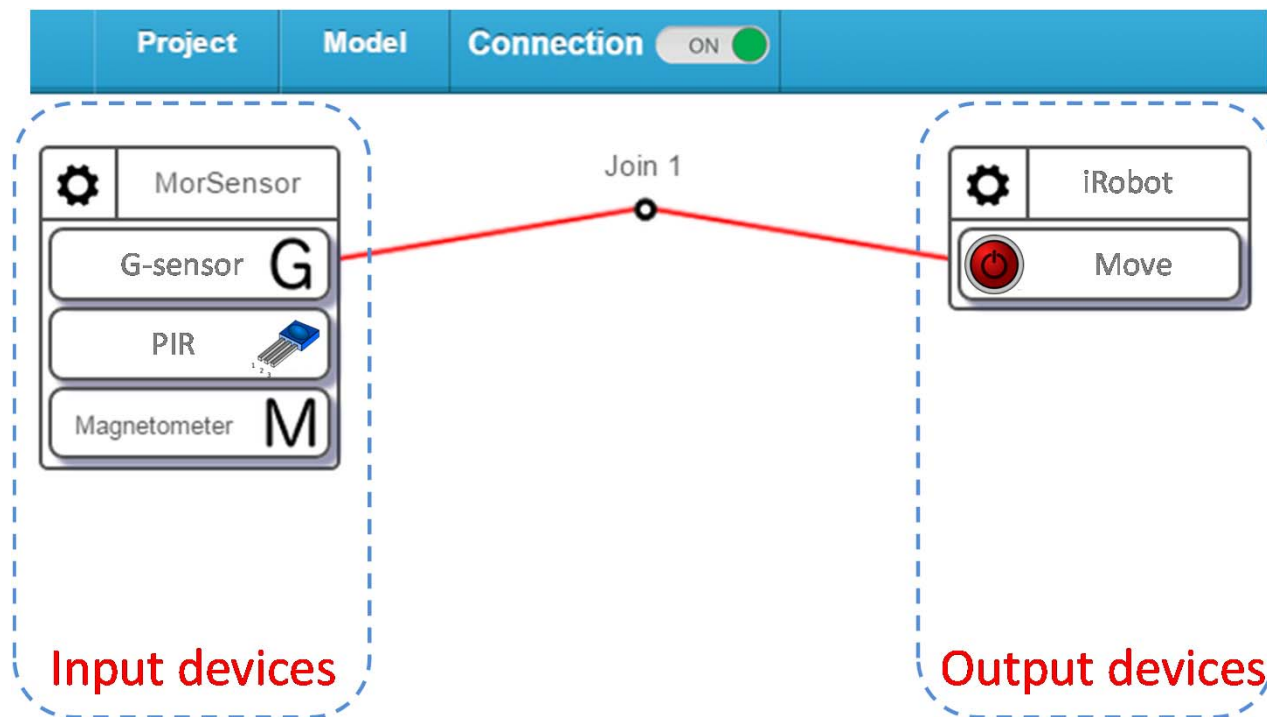
CCM: Creation, Configuration and Management system

Benefits of IoTtalk

- **Applications can simply develop with lower efforts**
- **Simple and intuitive GUI**
- **Application development without real devices is feasible**

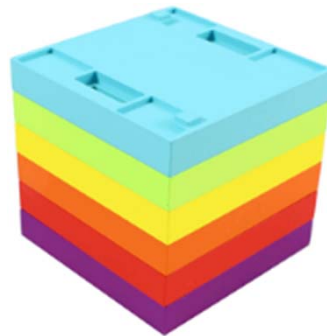
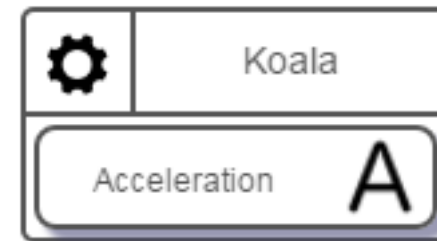
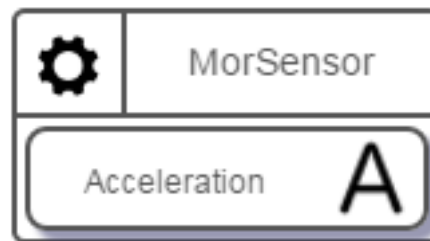
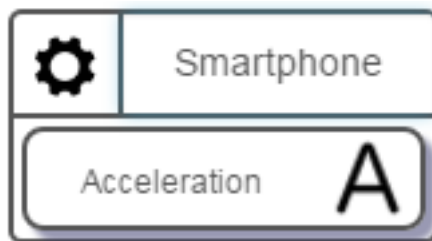
Simple and Intuitive GUI

- Connections by intuitional drawing links between IoT devices
- Transparently observe the connections between IoT devices
- Monitor the transmitting values between IoT devices
- Application debugging is more easier



Applications can simply develop with lower efforts

- Reusable DF modules
 - Even they are different IoT devices



Application development without real devices is feasible

- Do not need the real devices first to develop applications
- The simulator provides the numerical values as inputs

The screenshot displays the IoTalk web interface in a browser window. The address bar shows the URL `140.113.199.200:7788/connection/21#`. The interface includes a top navigation bar with tabs for **Project21**, **Model**, **Connection** (with an ON toggle), and **Delete**. A **Simulation** toggle is also present and turned ON.

The main workspace shows a visual programming diagram. On the left, a block labeled **Koala** (with a gear icon) contains an **Acceleration** input and a large letter **A**. A red line connects this block to a central **Join 1** node. Another red line connects the **Join 1** node to a block labeled **Dandelion** (with a gear icon) on the right. The **Dandelion** block contains a **Color-O** input and a large letter **C**.

On the right side of the interface, there are two data monitors:

- IDF Monitor**: The **Sub-stage** is set to **Input**. It displays a table of numerical data with columns for **Timestamp**, **X₁**, **X₂**, and **X₃**. The data is as follows:

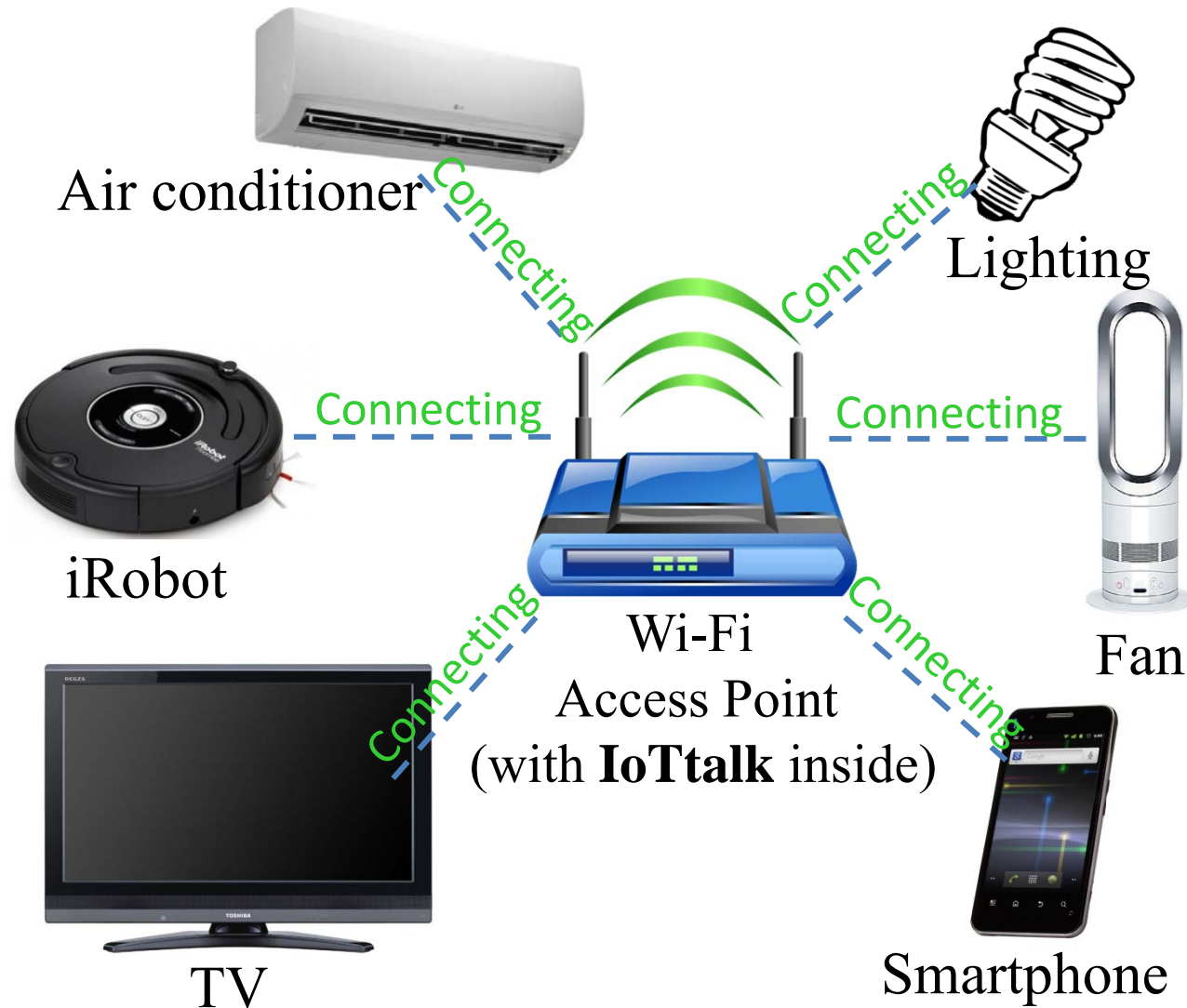
Timestamp	X ₁	X ₂	X ₃
04:10:11	0.56	0.30	0.16
04:10:12	0.69	0.78	0.79
04:10:13	0.14	0.81	0.61
04:10:15	0.30	0.79	1.00

Below the table is an **Input Data** text field and a **Send** button.

- ODF Monitor**: The **Sub-stage** is set to **Function**. It displays a table of numerical data with columns for **Timestamp**, **Y_{1,F}**, **Y_{2,F}**, and **Y_{3,F}**. The data is identical to the IDF Monitor:

Timestamp	Y _{1,F}	Y _{2,F}	Y _{3,F}
04:10:11	0.56	0.30	0.16
04:10:12	0.69	0.78	0.79
04:10:13	0.14	0.81	0.61
04:10:15	0.30	0.79	1.00

Appliances/IoT Devices Connection



Easy to Deploy and Operate



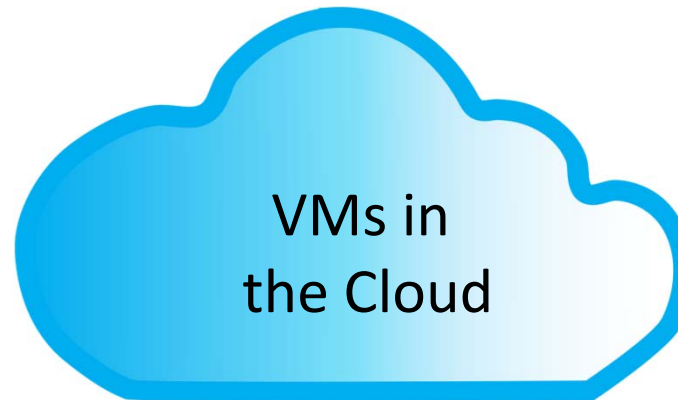
Intel Edison



Raspberry Pi3



PC server



For example, you can try

<http://140.113.199.200:7788/connection>

Application Example

