جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

أكاديمية كاوست
KAUST ACADEMY

Internet of Things (IoT)

DAY 5 Build your own device, then see where IoT lives

August 2 to 6, 2026

KGSP Summer Enrichment Program 2026
KAUST Academy, KAUST

Salah Abdeljabar
salah.abdeljabar@kaust.edu.sa

🗺️ On Today's Agenda

1. 🎓 **The brief**: one sensor, one actuator, one service, starting from the template
2. 🔧 **Build**: your own device, in pairs
3. 🎤 **Demos**: three minutes per pair, in front of the room
4. **IoT is everywhere**: the sectors, from the home to the factory to the city
5. **Edge computing**: when the cloud is too far away
6. **Where to go next**: single-board computers, and choosing the right brain
7. **Benefits & concerns**: the promise and the price

A simple thing that fully works beats an ambitious thing that does not.

PROJECT 13 · CAPSTONE

Capstone: Build Your Own IoT Device

Combine the week into one device that senses, acts, and connects to the cloud, then demo it.

Builds on: everything so far. This is where the twelve separate projects stop being separate.

⏱️ ~60 min 📊 Advanced 🔖 Prerequisite: Projects 1-12

⬇ Download the starter sketch

The starter already connects to Wi-Fi and Ubidots. Follow the `TODO 1/2/3` markers to make it yours.

Project 13 · Capstone 🎓

Design and demo an IoT device of your own choosing, starter sketch and rubric included.

- Combine a sensor, an actuator, and a network service into one device
- Scope a build to the time available
- Start from a template and extend it, the way most embedded work begins
- Debug a multi-part system by testing one piece at a time

Key pins: your choice

Open the guide ↗

The Brief

In teams, combine the week into one small end-to-end device that does all three:

Sense
DHT11, PIR, LDR, pot

Act
LED, RGB, buzzer, relay, OLED

Connect
Ubidots and/or Telegram

Idea	Sense	Act	Cloud
Smart room monitor	DHT11 + LDR	OLED readings	Ubidots charts
Motion security node	PIR	Buzzer	Telegram alert
Comfort meter	DHT11	RGB green/yellow/red	Ubidots dashboard

One session to build, then a short demo. A simple thing that fully works beats an ambitious thing that does not.

Start From the Template: Three TODOs

The starter is Project 11 hollowed out. It compiles and publishes `0.0` as-is, so the **cloud half works before you wire a sensor**.

```
float readMySensor() {            // TODO 1: your sensor (ADC pin, DHT, ...)
  return 0.0;
}
ubidots.add(VAR_SENSOR, value);   // TODO 2: publish each value you care about
ubidots.publish(DEVICE_LABEL);

void callback(char *topic, byte *payload, unsigned int length) {
  digitalWrite(OUTPUT_PIN, v >= 0.5);   // TODO 3: decide what a command means
}
```

- The template already does Wi-Fi, the broker, reconnect, the non-blocking publish timer, and `ubidots.loop()`.
- A new sensor is **three edits**: the `#include`, the object, and the `begin()` in `setup()`. Forgetting the third is the usual bug.
- A command payload is a **number**, so it can be a brightness, a target, or an index, not just on/off.

Scope It, Then Demo It

Finish on time

- Wire and test **one part at a time**
- Confirm good values on Serial before adding the cloud
- `millis()`, never `delay()`, so Wi-Fi stays responsive
- Keep a known-working version before the next feature

The demo (about 3 min)

- What problem does it solve, and for whom?
- Show it live: trigger the sensor, watch it respond
- One thing that was harder than expected
- One thing you would add with another day

The Week in One Loop

Every project was one idea. The capstone is where they stop being separate:

- **Read the world, decide, act** (P1): everything since is a richer version of that loop
- **Digital or analog, counts or units** (P2, P3, P9): know which sensor you have
- `millis()`, **never** `delay()` (P4): what lets a device do two things at once
- **The board can serve a page** (P5 to 9): great locally, invisible from outside
- **Keep state honest** (P8), and **local output has its place** (P10)
- **Publish/subscribe reaches anywhere** (P11), and **an existing app can be the UI** (P12)

The habits matter as much as the APIs: check hardware started, guard bad readings, keep credentials private, and print enough on Serial to see what is happening. The next device is the same shape, with different parts.

REFERENCE · GPIO PINOUT

ESP32 DEVKIT V1: GPIO Reference

A lookup table for the 30-pin ESP-WROOM-32 board: pick a pin without breaking booting or Wi-Fi.

> **Rule of thumb:** for a plain input or output, reach for a ✅ **safe** pin first: **GPIO 4, 13, 16, 17, 18, 19, 21, 22, 23, 25, 26, 27, 32, 33**.

Reference · GPIO Pinout

📌

The full ESP32 DEVKIT V1 pin table: safe pins, strapping pins, input-only pins, plus ADC, touch, DAC, and the default bus pins.

- Which GPIOs are safe to use freely
- Which pins must be left alone at boot (strapping pins)
- Which pins are input-only, and which carry ADC/DAC/touch
- Default I²C and SPI bus pins

Open the guide ↗

REFERENCE · COMPONENTS

Components Glossary

Every part in the kit: what it is, how to recognize it, whether direction matters, and where it's used.

> **Polarity key:** 🔴 **polarized**: direction matters, can be damaged if reversed · ○ **not polarized**: either way is fine.

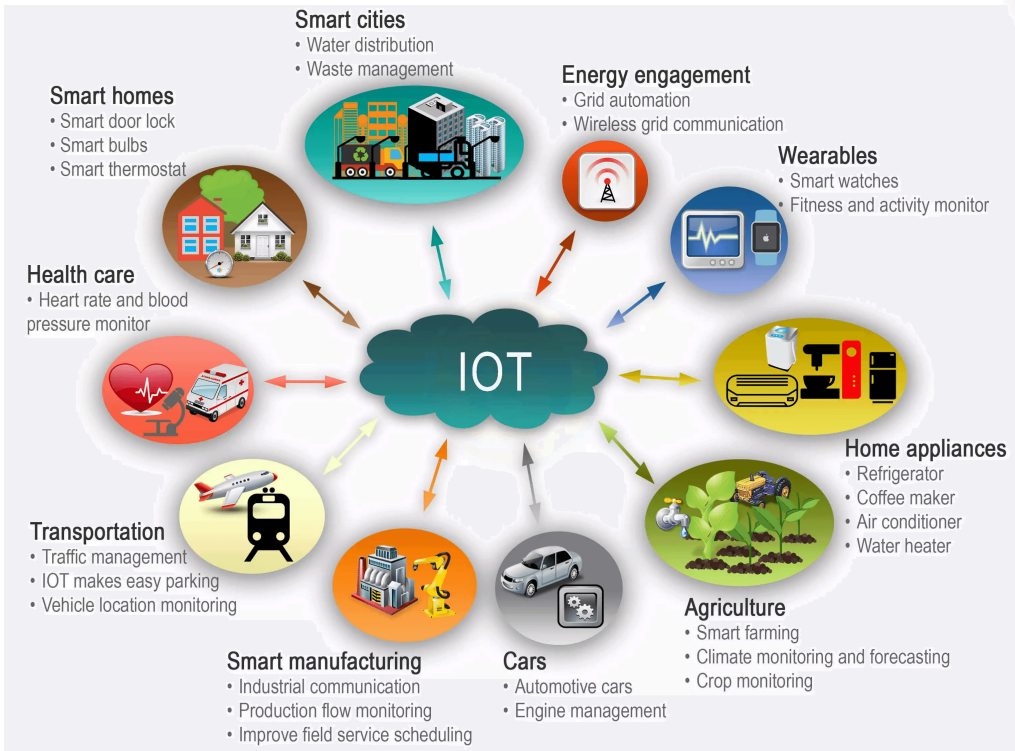
Reference · Components Glossary 🧩

Every part in the kit: how to identify it, whether it’s polarized, and which project uses it.

- A photo and description of each component in the kit
- Polarity called out where it matters
- A pointer to the project that first introduces each part

Open the guide ↗

IoT Is Everywhere 🌍



Source: Rejoicehub

Consumer IoT 

IoT devices you buy and use around the home

Consumer IoT

Devices consumers buy for everyday life: some incredibly useful, some questionable.

🛋️ Around the home

- Smart speakers & assistants
- Smart heating & thermostats
- Robotic vacuum cleaners
- Smart appliances & plugs

⌚ On the person

- Fitness & health trackers
- Wearables & smart watches
- Video doorbells & security cameras

> **Important**
>
> Consumer IoT empowers the **1 billion people with a disability**: voice-controlled ovens, robot vacuums, and self-monitored health conditions.

Commercial IoT 

IoT in the workplace, retail, and transport

Commercial IoT

Using IoT across offices, factories, retail, and logistics.

🏢 Buildings & retail

- Occupancy sensors for lighting/heating
- Factory safety: hard hats, noise levels
- Cold-storage temperature monitoring
- Smart shelves that signal restocking

🚚 Transport & logistics

- Vehicle location tracking
- Road-user mileage charging
- Driver hours & break compliance
- Depot arrival notifications

> ⓘ **Note**
>
> The goal: **cut cost and carbon**, improve **safety**, and keep operations running smoothly.

Infrastructure IoT 

Smarter cities, power, and environment

Infrastructure IoT

Monitoring and managing the infrastructure people use every day.

🏙️ Smart cities

- Pollution monitoring
- Traffic & smart parking
- Public transport management

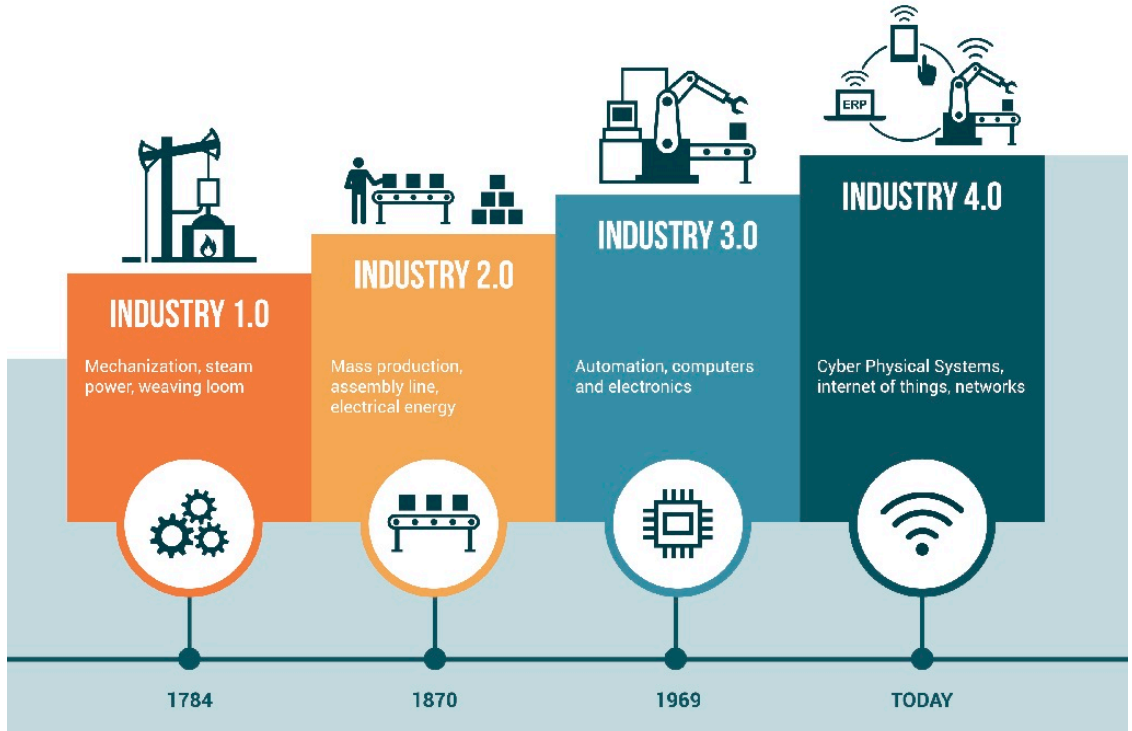
⚡ Utilities

- Smart power grids
- Home-level usage analytics
- Water & environmental monitoring

Real example: Copenhagen

Air pollution is measured across the city and used to suggest the **cleanest cycling and jogging routes**.

Industry 4.0 and Industrial IoT (IIoT)



Source: Aberdeen

From Factories to Farms

🏭 Factories

Monitor machinery with many sensors:

- Temperature, vibration, rotation speed
- Auto-stop when outside tolerances
- Predictive maintenance: AI predicts failures *before* they happen

🌱 Digital agriculture

Feeding a growing planet:

- Soil-moisture → automated watering
- Growing degree days to predict harvest
- Drones, satellites & AI for crop health

Why IIoT Matters

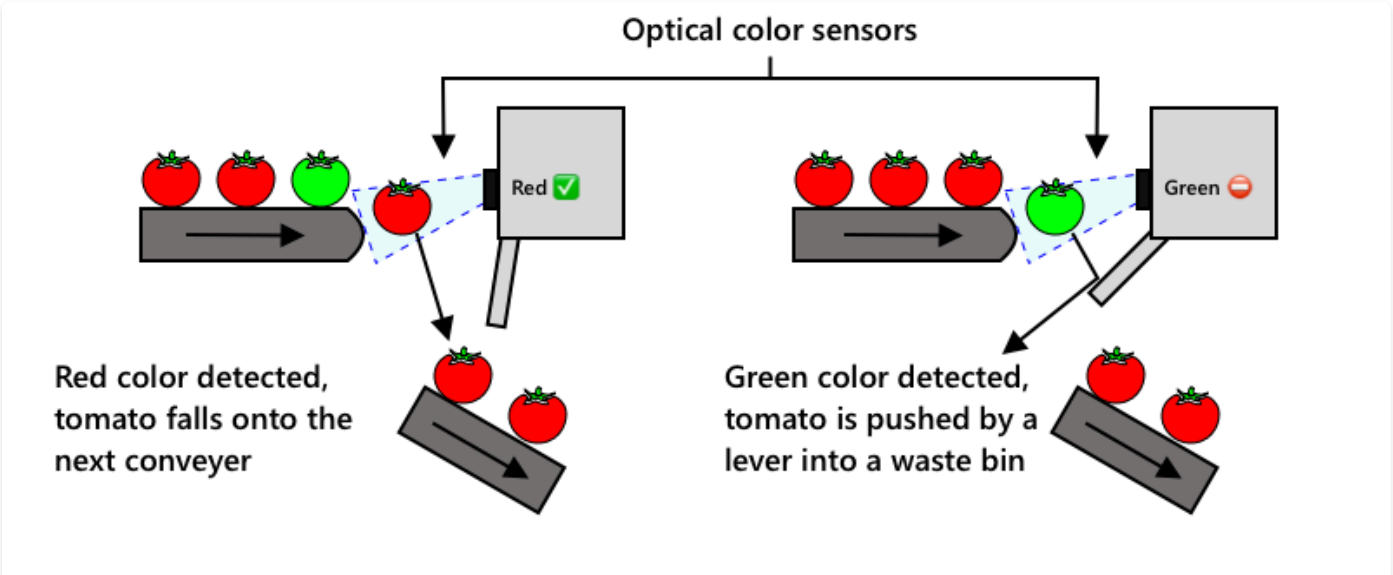
Important

Digital agriculture can help the **2 billion people** in 500 million households living on subsistence farming, starting with sensors that cost just a few dollars.

Think

What other IoT devices could help farmers grow more with less?

One Machine, End to End 🍅



Sense 👁️ An optical color sensor reads each tomato as it passes.

Decide 🧠 Red passes, green does not. The rule runs on the machine, in milliseconds.

Act 🦾 A lever pushes the rejects into a waste bin.

A whole IoT loop with no cloud in sight: at conveyor speed, the round trip would cost more than it's worth.

Edge Computing 

Process data locally instead of always calling the cloud

What is the Edge?

Even though the **I** in IoT is "Internet," devices don’t always have to use it. **Edge devices** are gateways on your **local network** that process data without a round-trip to the cloud.

⚡ Speed

Handle lots of data or a slow connection without waiting on the Internet.

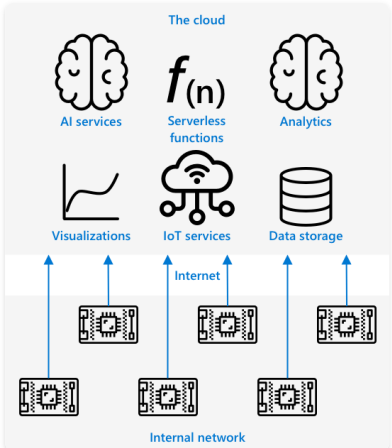
📴 Offline

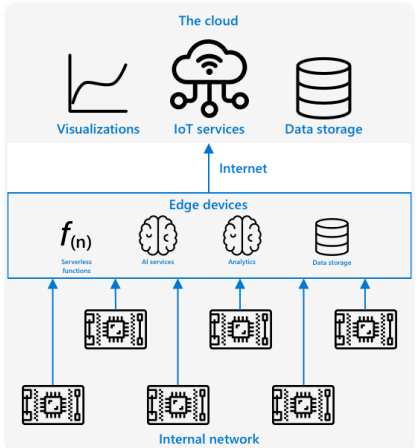
Keep working with no connectivity: on a ship, or in a disaster area.

🔒 Privacy

Sensitive data stays on the local device.

Without the Edge vs With the Edge





Without: every device crosses the Internet for every decision.

With: an edge device runs the same services on your network.

The cloud stack (functions, AI, storage) moves next to the devices. Only what matters goes up.

Edge in Action

A smart speaker listens for its **wake word** locally, only sending your speech to the cloud *after* it wakes.

Important

Everything said **before** the wake word (and **after** it stops listening) never leaves the device, keeping it private.

Single-Board Computers (SBCs) 💻

A full computer on a single board

MCU or SBC?

💻 Single-Board Computer

A complete computer on one board, running a **full operating system**.

- Runs **Linux**, coded in **Python**
- More power, memory & complexity
- Connect monitors, keyboards, USB
- e.g. **Raspberry Pi**

🧠 Microcontroller

A chip built to do **one specific task**.

- Runs **C/C++**, no full OS
- Tiny power draw, low cost
- Real-time, single-purpose
- e.g. **ESP32**

Side by Side

	🧠 Microcontroller	💻 Single-Board Computer
Cost	Cents to a few dollars	Tens of dollars
Power	Months on a battery	Usually mains-powered
Software	C/C++, no full OS	Linux + Python, full OS
Complexity	Single dedicated task	Many apps at once
Timing	Deterministic, real-time	OS-scheduled

Which Do I Pick?

Reach for an MCU

Battery-powered, **dedicated**, **real-time** tasks: a sensor node, a nightlight, a wearable.

Reach for an SBC

Complex logic, **multiple peripherals**, a **rich UI**, or **on-device AI**.

Tip

Many real products use **both**: an SBC coordinating several MCUs.

Benefits & Concerns ⚖️

The promise and the price of connecting everything

Weighing IoT

✅ Benefits

- Convenience & automation
- Efficiency: save cost & energy
- Safety & accessibility
- Data-driven insights

⚠️ Concerns

- **Privacy**: who sees your data?
- **Security** vulnerabilities
- Dependence on **connectivity**
- Interoperability between brands

Security Deserves Attention 🔒

> **Caution**
> There's an old joke: *"the **S** in IoT stands for Security."* There is no S. A device is only as secure as the cloud service behind it.

- Real attacks: the **Stuxnet** worm, hacked **baby monitors**
- Weak passwords and open connections are common entry points
- Good design treats privacy and security as first-class requirements

Thank you!

For spending the day building the Internet of Things with us.

Questions? Ask now, or write to me any time.

Slides, labs, and resources

KGSP Summer Enrichment Program 2026
KAUST Academy, KAUST

Salah Abdeljabar
salah.abdeljabar@kaust.edu.sa