LANKA NIPPON BIZTECH INSTITUTE

2024/3/4-2024/3/5

IoT and Embedded Systems Basics

Kazuaki Tanaka

Kyushu Institute of Technology

About Me

- Kazuaki TANAKA（たなか かずあき）
- Kyushu Institute of Technology, JAPAN Associate Professor
- Ph. D in Information Science

- Research topics: embedded systems, IoT, mruby, wireless communication

What is “Embedded Systems”

- Computer system “Embedded” in equipment (machines)
- Provides “specific” function
- Example:
  - Home appliance
  - Medical equipment
  - Industrial machinery
  - Mobile phone

Need for embedded systems

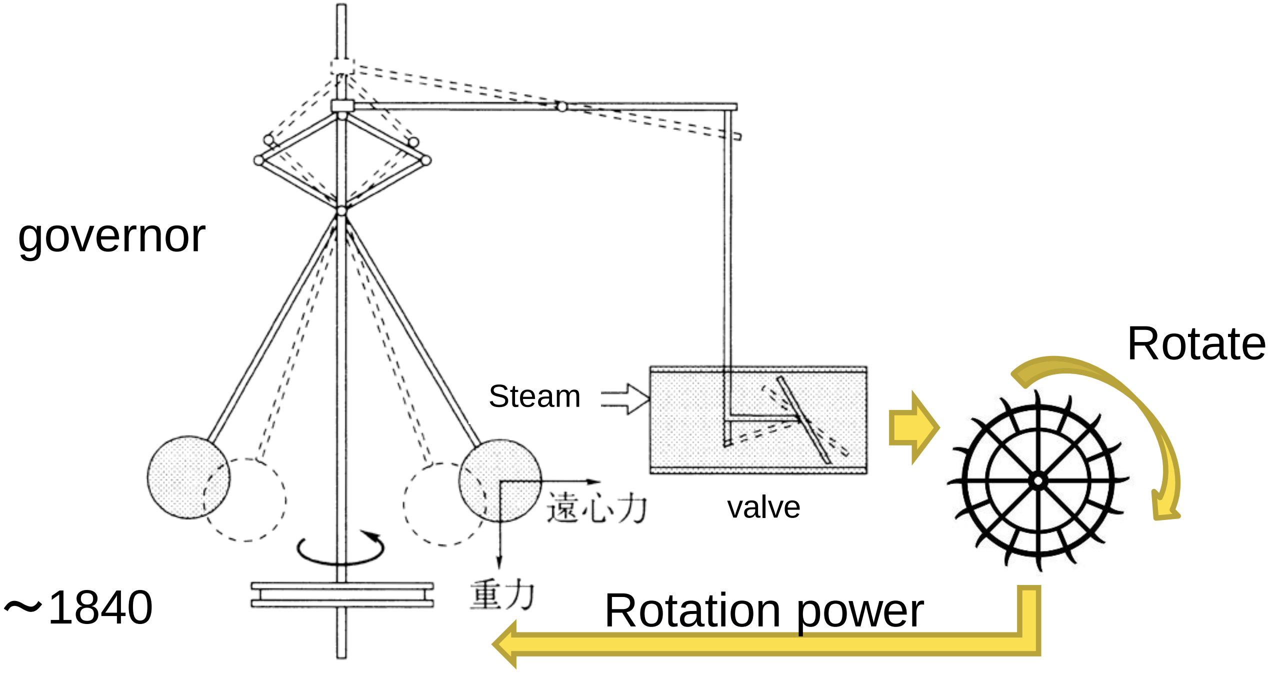
- Historically, embedded systems are related control systems
  - start from the age of the steam engine

- Systems involving multiple devices and functions
- Technology to control the final output (=goal)

- In the case of steam engines
  - Steam is used to rotate a turbine for power
  - Keep at the constant rotating speed of machine

(Example) Simple control system

- Steam engine (or internal fuel combustion engine) control
  - steam engine governor (keep rotational speed)



Year 1750～1840

Revolution

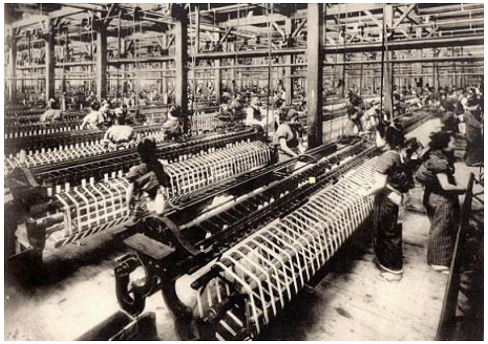
- This mechanism is REVOLUTION
- Revolution: Technology to change the society
- After governor invention,
  factories was built.

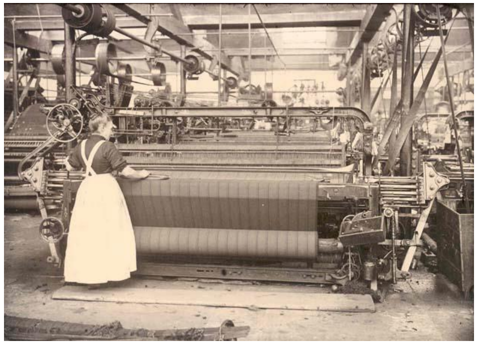
Society changes

Cotton

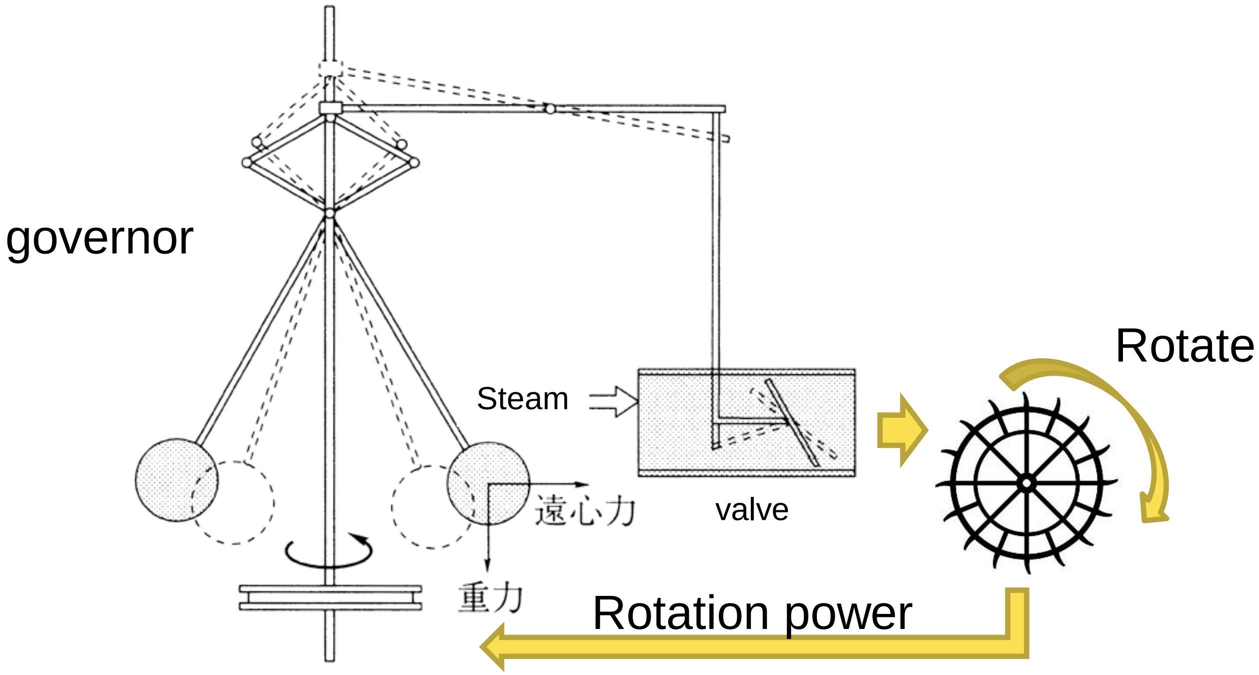
Thread

Cloth





Control system

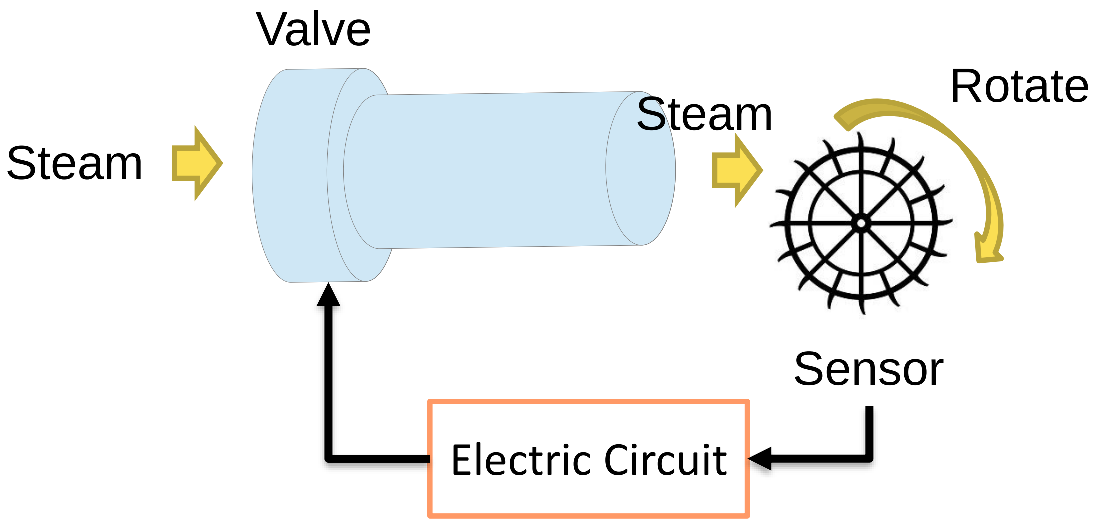


- Rotation speed increased, valve is closed
- Rotation speed decreased, valve is opened

Next revolution

- Electric circuit

Automatic control (Negative feedback loop)

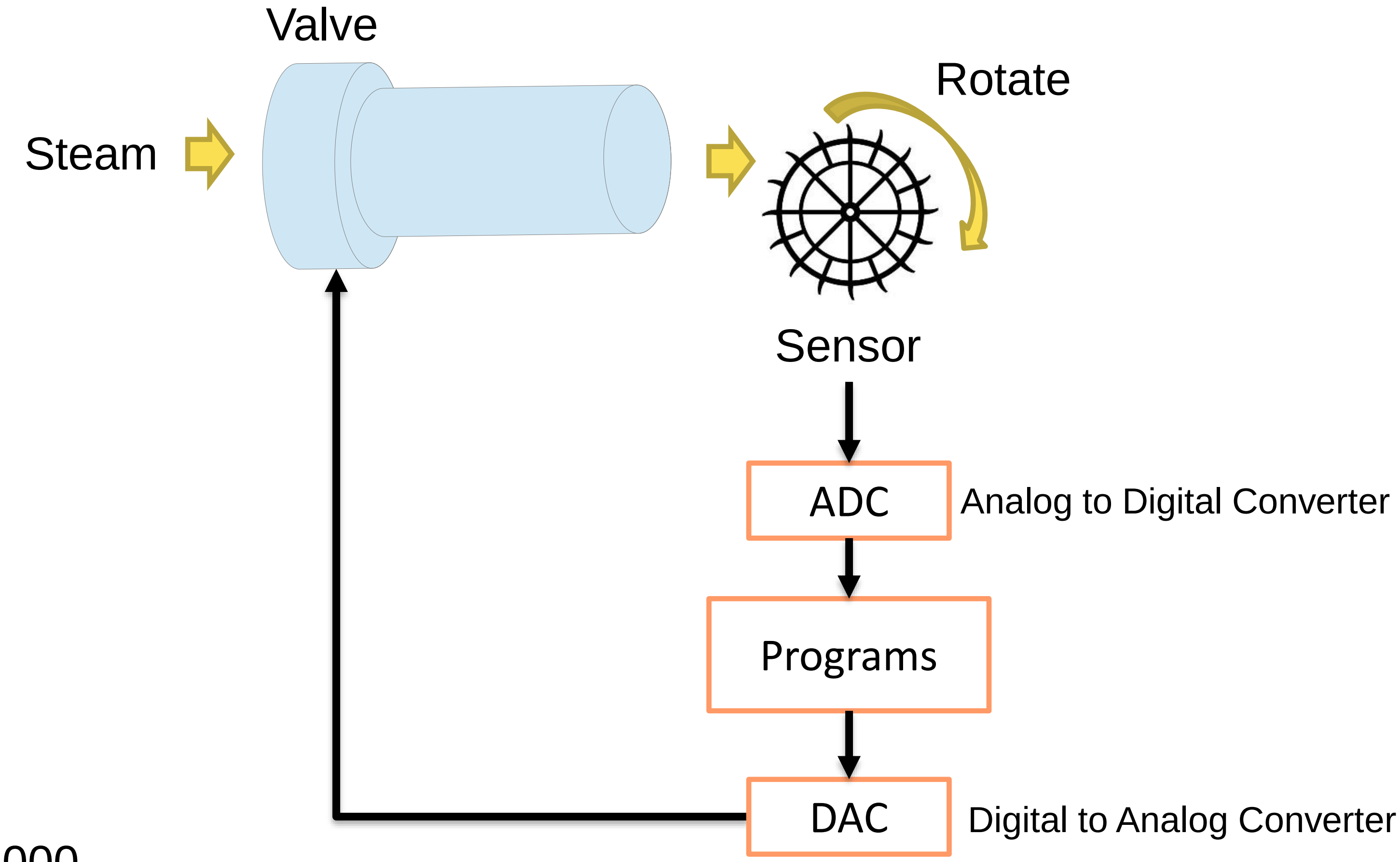


Year 1870～1915

Next revolution

- Computer

Computer control



Valve

Steam

Rotate

Sensor

ADC

Analog to Digital Converter

Programs

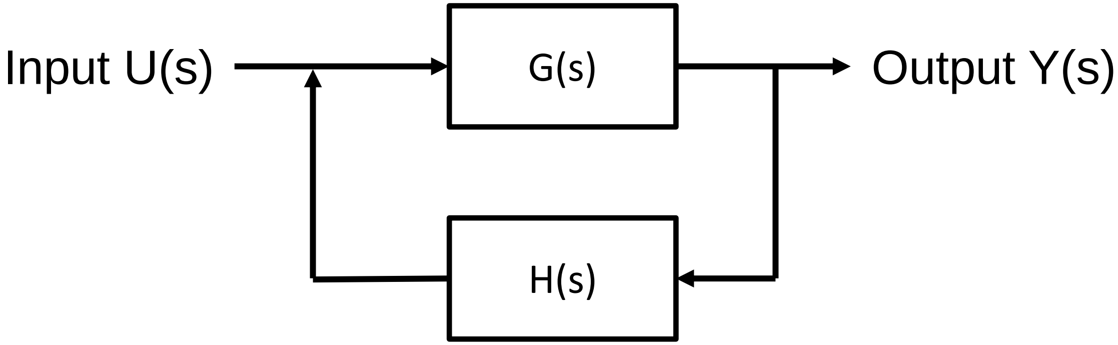
DAC

Digital to Analog Converter

Year 1950～2000

Control system basics: Feedback Control

- Transfer function of feedback control



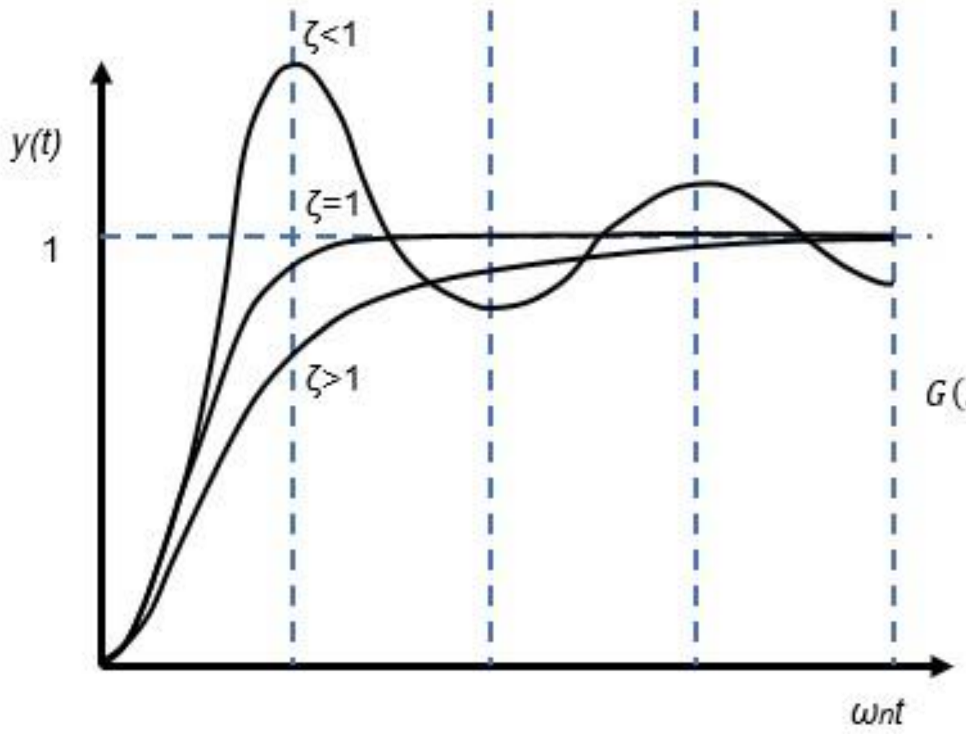
$$Y(s) = G(s)(U(s) - H(s)Y(s))$$

$$Y(s) = \frac{G(s)}{1 + G(s)H(s)} U(s)$$

Just solve this equation!

That’s all?

- Everything done by using feedback loop?



$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Industrial Revolution

- No control … before industry

⇩ 1750～1840

- Governor

⇩ 1870～1915

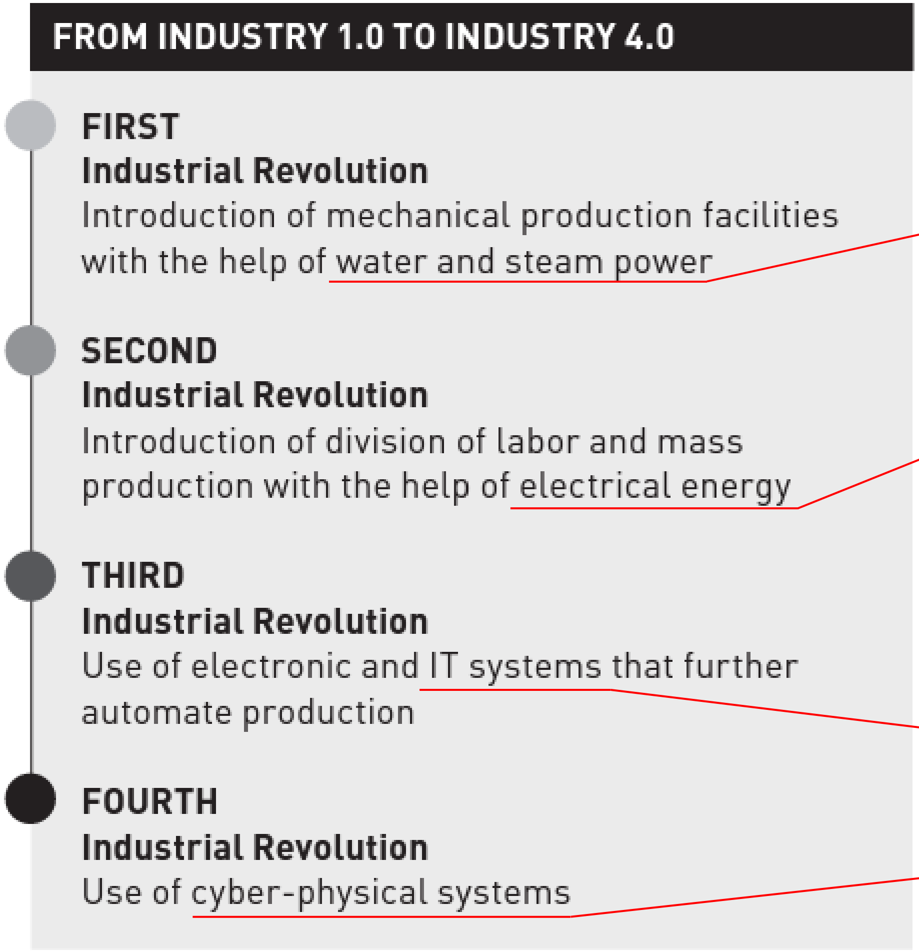
- Electronic control

⇩ 1950～2000

- Computer control

⇩ 2011～

Industrial Revolution



FROM INDUSTRY 1.0 TO INDUSTRY 4.0

FIRST
Industrial Revolution
Introduction of mechanical production facilities with the help of water and steam power

SECOND
Industrial Revolution
Introduction of division of labor and mass production with the help of electrical energy

THIRD
Industrial Revolution
Use of electronic and IT systems that further automate production

FOURTH
Industrial Revolution
Use of cyber-physical systems

Source: DFKI 2011

1750～1840 Steam power

1870～1915 Electric circuit

1950～2000 Computer, IT

2011～ Smart system, IoT

German Research Centre for Artificial Intelligence

You are standing in 4th industrial revolution

KEY technology: Smart system, IoT, AI

Requirements in embedded systems

- Sensor Input
  - Timing of sensor acquisition
- Execute control algorithm
- Control output
  - User interface
  - Communication
  - Logging, Analyzing, Diagnostic

What is important in embedded systems?

- Real-time performance
- Execution speed
- Cost
- Quality
- Delivery time

What's the difference?

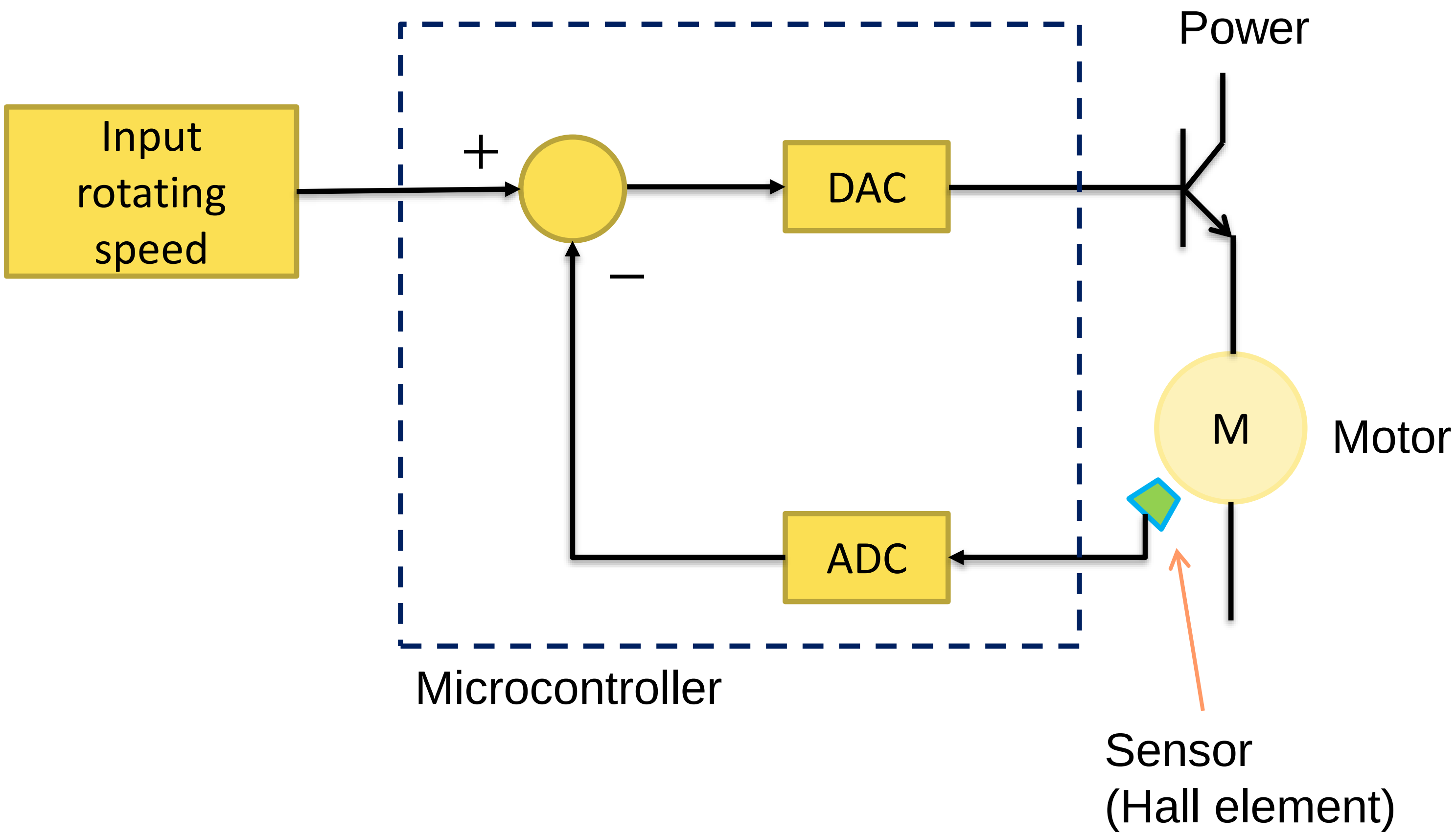
Real-time process

- NOT “the execution speed is fast”.

- Execution time must be accurate.
- Execution time must be estimated.

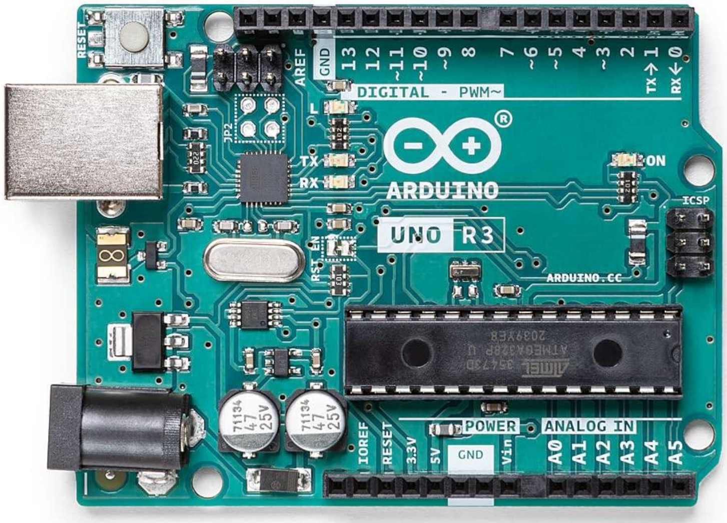
Real-time system

- Control the motor speed

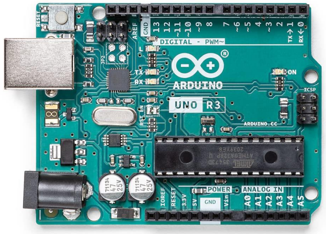


（Processor example） ATmega328

- Used in Arduino
- It has all the features you need in a computer.
  - CPU (8Bit Bus, 16MHz)
  - 32KB ROM(Flash)
  - 2KB SRAM
  - I/O
  - ADC, DAC
  - Hardware interruption



Computer / Micro-controller

		
Processing Power	64bit CPU 3.2 GHz 8GB Memory	8bit CPU 16 MHz 2KB Memory
Realtime processing (Processing Cycle)	100Hz	10 KHz

DEMO

```
led1 = GPIO.new(0)

while true do
   led1.write 1
   sleep 0.5
   led1.write 0
   sleep 0.5
end
```

DEMO

```
led1 = GPIO.new(0)

while true do
  led1.write 1
  sleep 0.5
  led1.write 0
  sleep 0.5
end
```

```
led1 = GPIO.new(5)

while true do
  led1.write 1
  sleep 0.6
  led1.write 0
  sleep 0.6
end
```

Embedded system development

- Requires development of both hardware and software
  - Both Knowledge
- Hardware development
  - A small computer called a microcontroller
  - Processor, memory, etc.
- Software development
  - Requires “cross development”

Software development (for general-purpose systems)

- Write programs using a PC keyboard and display.
- Run the developed program on the PC.
- Easy to develop.

Software development (for embedded systems)

- Write a program on PC.
- Transfer the developed program to the hardware (microcontroller).
- Run the developed program on the microcontroller.

embedded system development

- Technology for both hardware and software is necessary
  - Both hardware and software knowledge

  - Hardware: has some limitation, physical law
  - Software: no limitation, anything allowed

To solve the issues,

- Reduce the lines of code
- Reuse previously developed software, libraries
- Increase software readability

- Reduce development time
- Increase reliability
- Rapid software development

Embedded system development

- KEY is the development of software part
  - Development time (productivity, cost)
  - Product safety and reliability

- Consider the software product development
  - It is rare to create whole new code
  - Combine existing functions

Embedded development methodology and approach

- Traditional waterfall cannot handle increasing complexity and diversity of software

- Model Based Development (MBD)
  - Use modeling methods - improve quality
- Formal Method
  - Define rigorous specifications - improve quality
- Functional Safety
  - Operate safely in principle - Quality Improvement

Model Based Development (MBD)

- Design software using models (mainly diagrams and tables)
  - Development workload is reduced because it is possible to concentrate on the scope of the model
- Popular modeling methods
  - UML diagrams
  - Data flow diagrams (DFD)
  - State transition diagrams

How to eliminate program bugs?

- The easiest and most reliable way
  - Do not write programs (code)

- Diagrams and tables
  - Easy to find errors
  - Everyone will have the same understanding (Prevent misunderstanding)

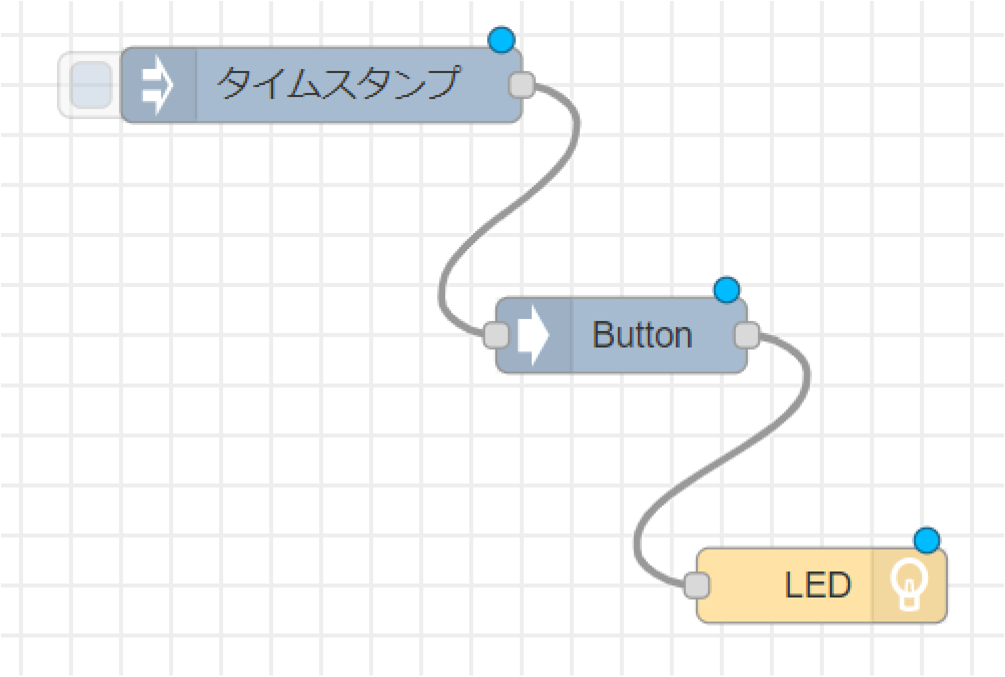
Dataflow diagrams (DFD)

- Represent the flow of data (sequence of processing) as a combination of diagrams

- Create blocks of functions (frequently used functions)
- Connect functions each other by wire (directed graphs)

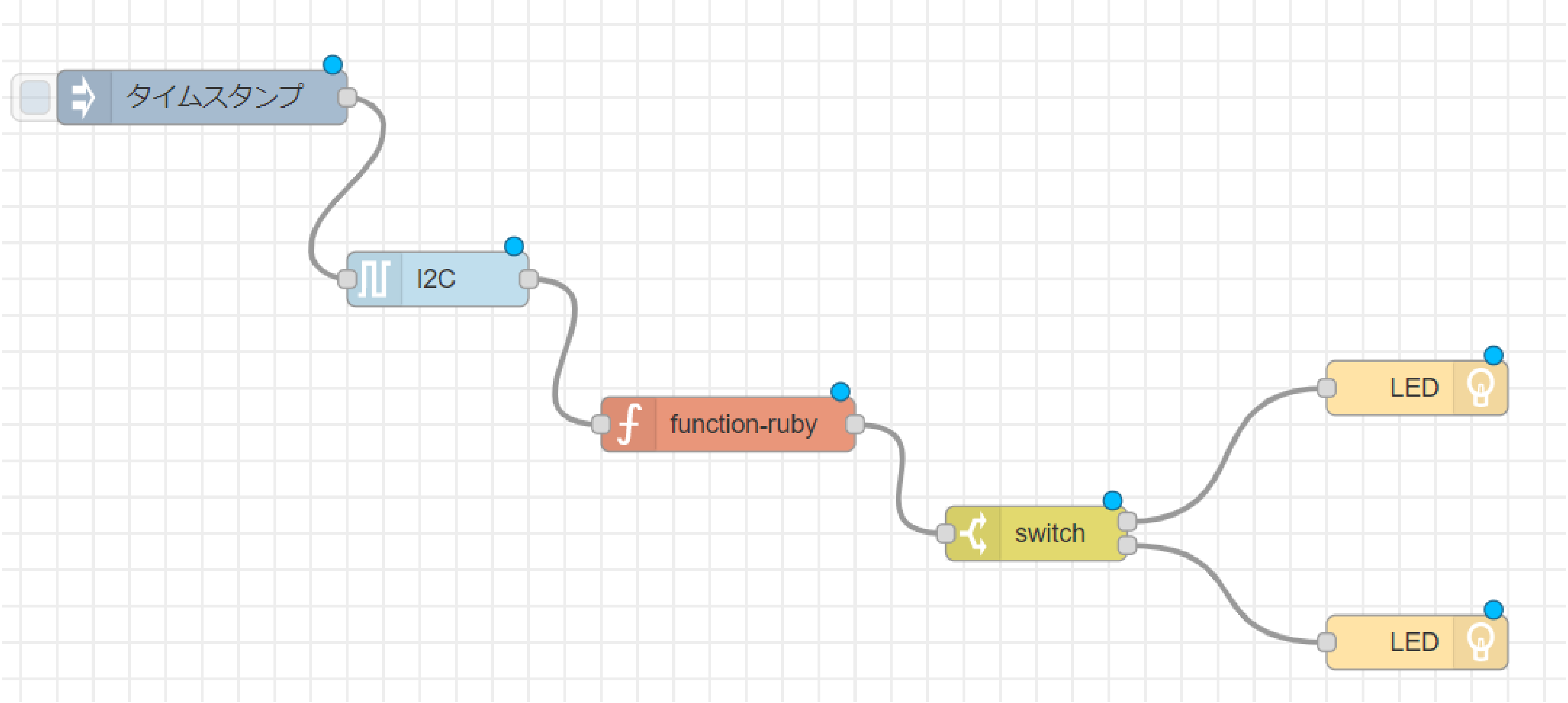
Example of data flow diagram

- LED lights up when button is pressed



Example of data flow diagram 2

- Get values from sensor and convert value
- Turns on LEDs according to the values



functional safety

- Intrinsic Safety and Functional Safety

- Intrinsic Safety
  - Removing Hazards (remove whole risk)
- Functional safety
  - Use functions that ensure safety

Examples of Intrinsic and Functional Safety

Risk of train accidents
(e.g., train-vehicle collisions)

Intrinsic Safety

Elevated railroad

Functional safety

Railroad crossing

Conclusion

- Embedded Systems Development
  - Hardware and software development
  - Large proportion of software
  - Software development with less hardware dependence is required
- Real-time performance
- Using models (MBD)
- Using Graphs (DFD)

Society in the future

- Industrial Revolution
  - The 4th Industrial Revolution
  - Major changes in social structure due to technological innovation
  - IoT and AI creates a better society, changing society

- Engineering (hardware and software) becomes more important
  - Programming skill is must

About Kyutech

- The Spirit of Kyushu Institute of Technology

技術に堪能なる士君子の育成

Gentlemen well versed in technological skills

- Education that emphasizes the acquisition of technology as well as human development