

07

Free Online Course

Web Application Programming 101

Basic



HTML



CSS



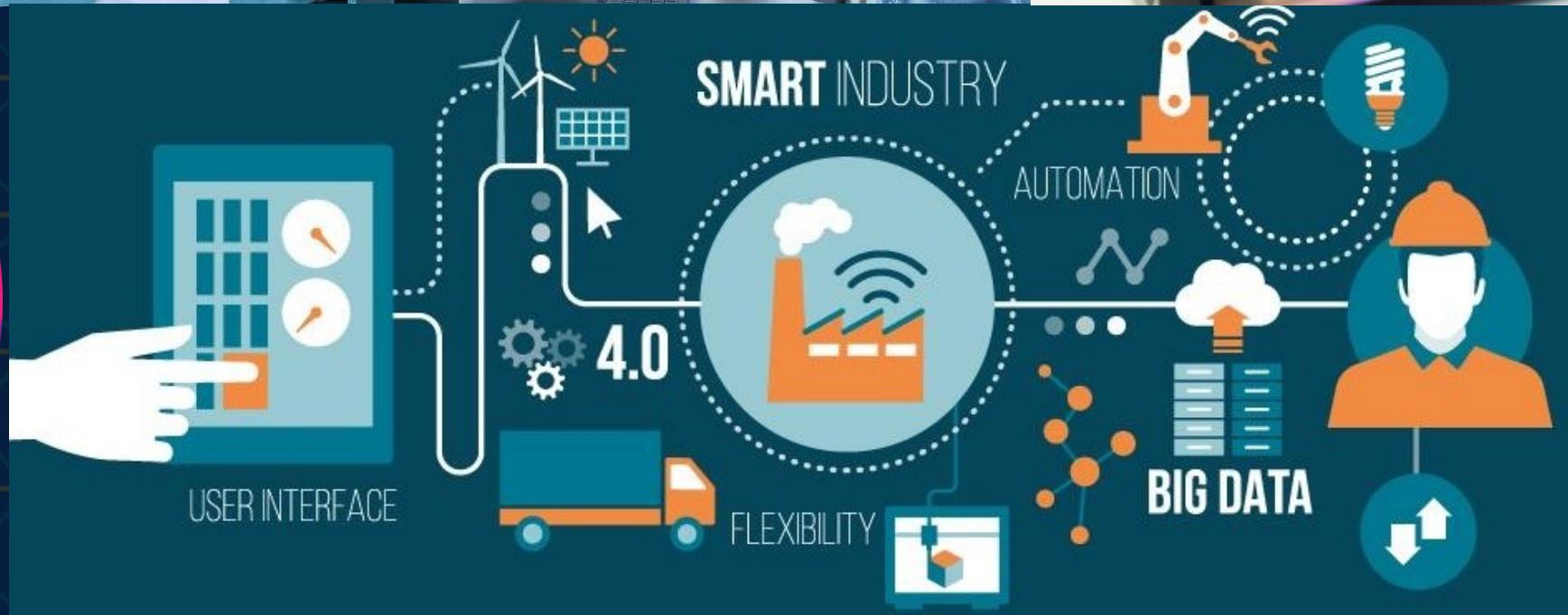
JS





- 1. Industry 4.0, Virtual Reality, Cyber-Physical System, and Digital Twin**
- 2. VR, CPS and DT DEMOs**
- 3. Virtual World/Machine Interfacing
Programming Practice**

Industry 4.0



Industry 4.0



Industrial Revolution

1784
Industry 1.0



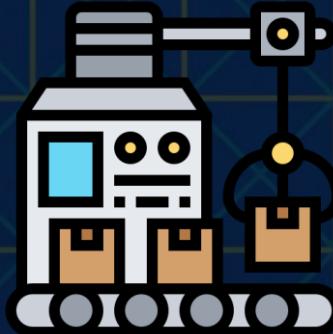
Mechanization
Stream power
Weaving loom

1870
Industry 2.0



Electrical energy
Assembly line
Mass production

1969
Industry 3.0



Robot & Automation
Digital & Electronic
Microprocessor and Computer

TODAY (YESTERDAY?)
Industry 4.0



Internet-of-Things
Cyber physical systems
Artificial intelligence
New technologies integration

5.0



Human creativity and smart technologies
will become a decisive force in 2035.

Virtual Reality (VR)



โลกเสมือนจริง



- Virtual Reality (VR) is an immersive experience also called a computer-simulated reality.
- It refers to computer technologies using reality headsets to generate the realistic sounds, images and other sensations that replicate a real environment or create an imaginary world.
- VR is a way to immerse users in an entirely virtual world.

Virtual Reality (VR)



Virtual Reality (VR)



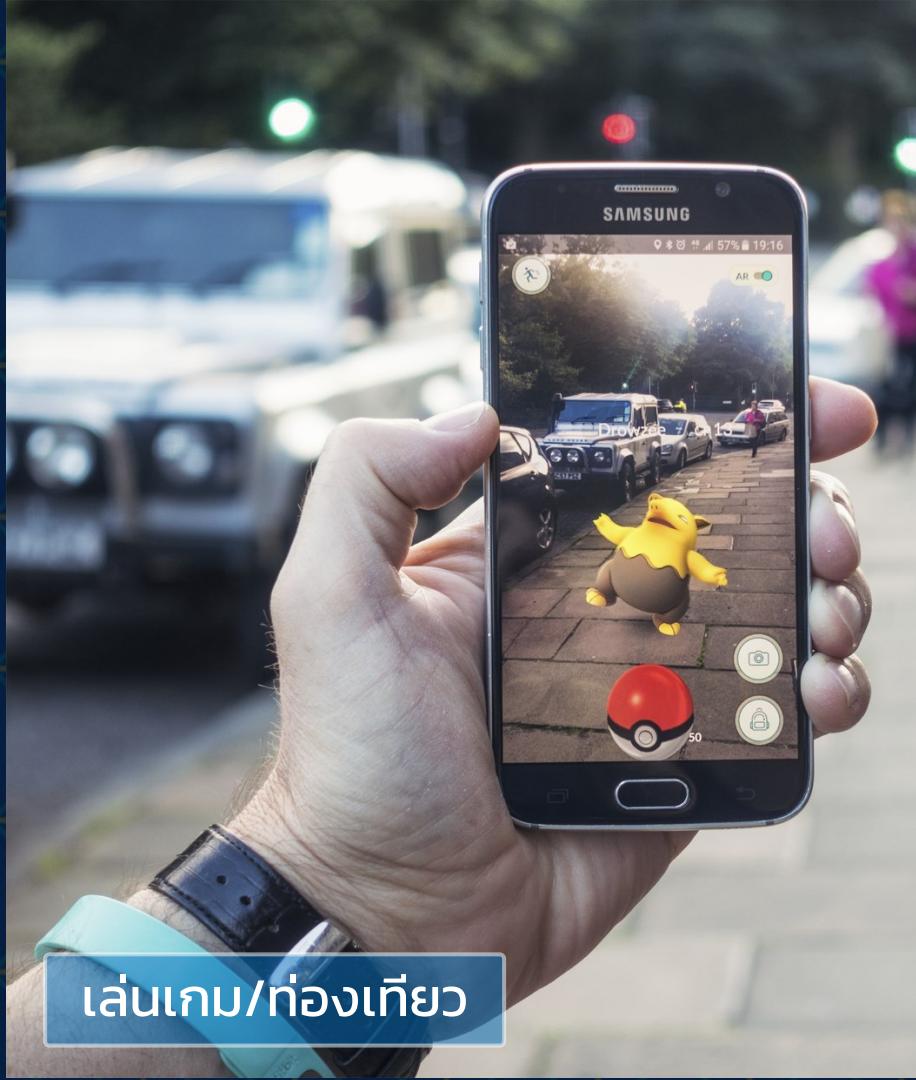
Furniture Showroom



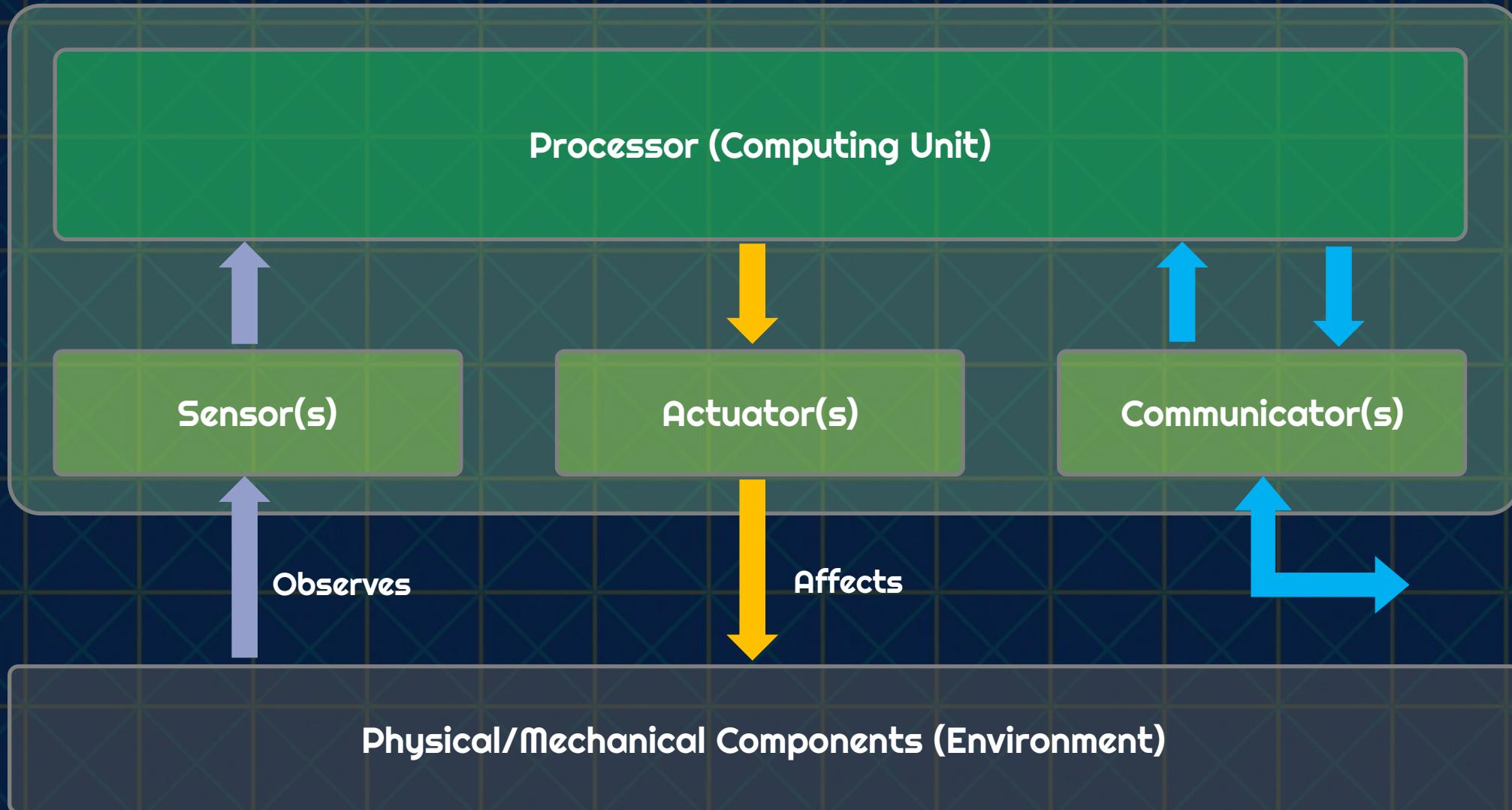
Car Showroom



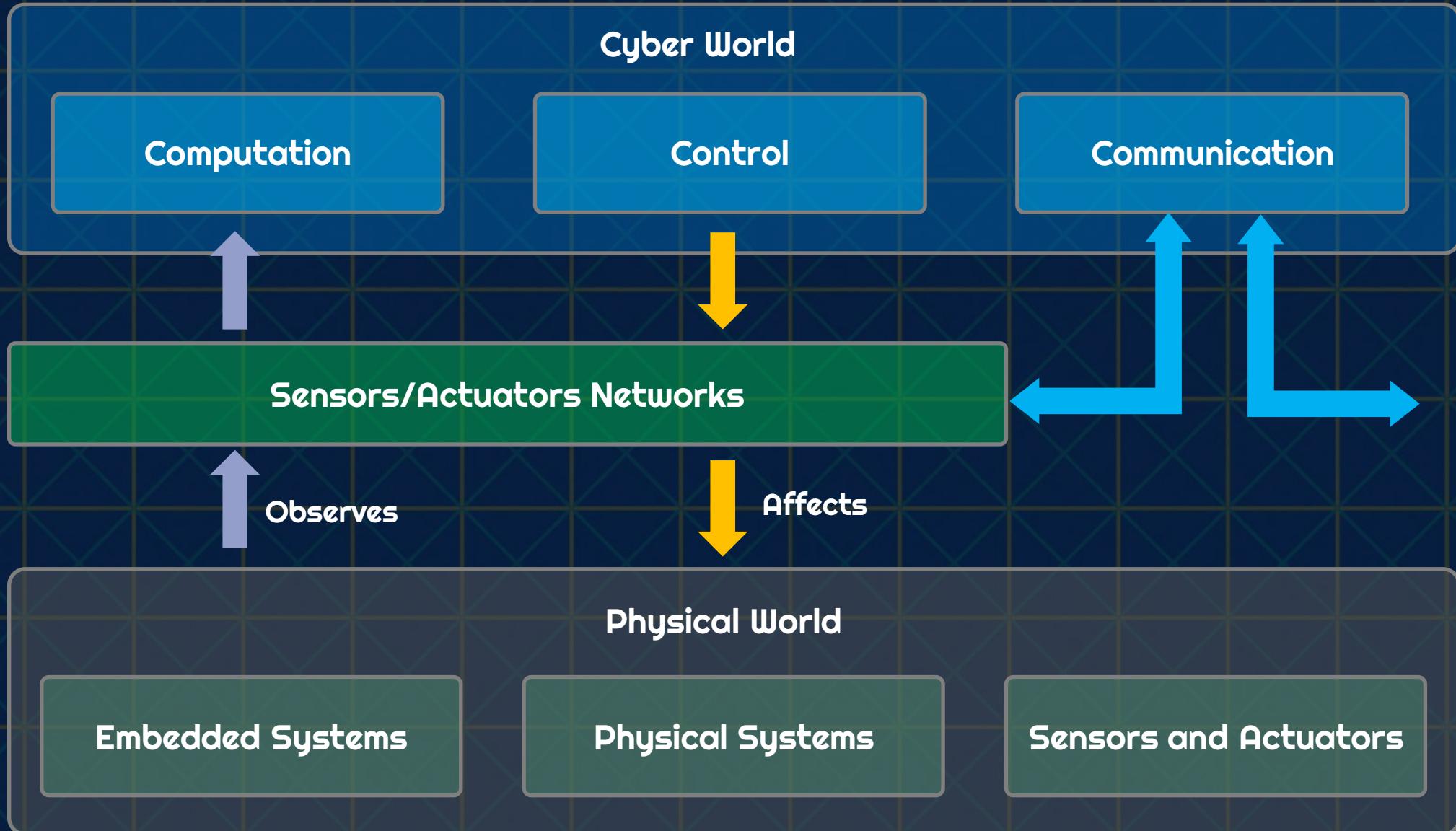
Augmented Reality (AR)



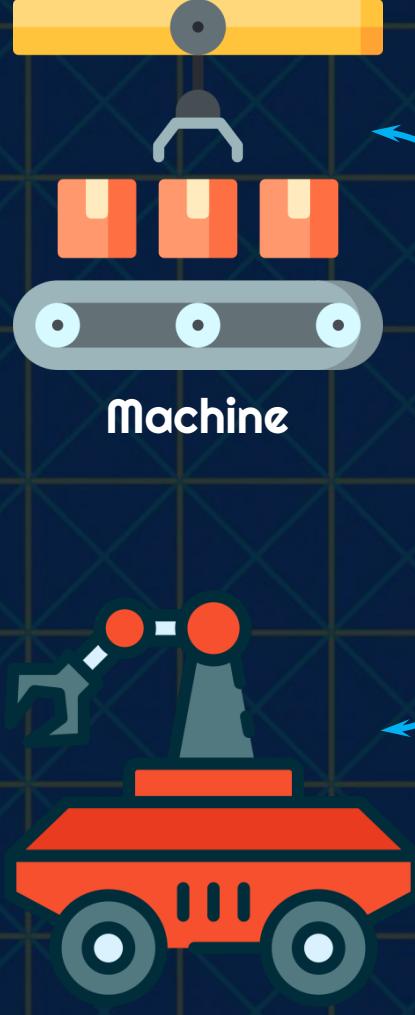
Cyber-Physical System (CPS)



Cyber-Physical System (CPS)

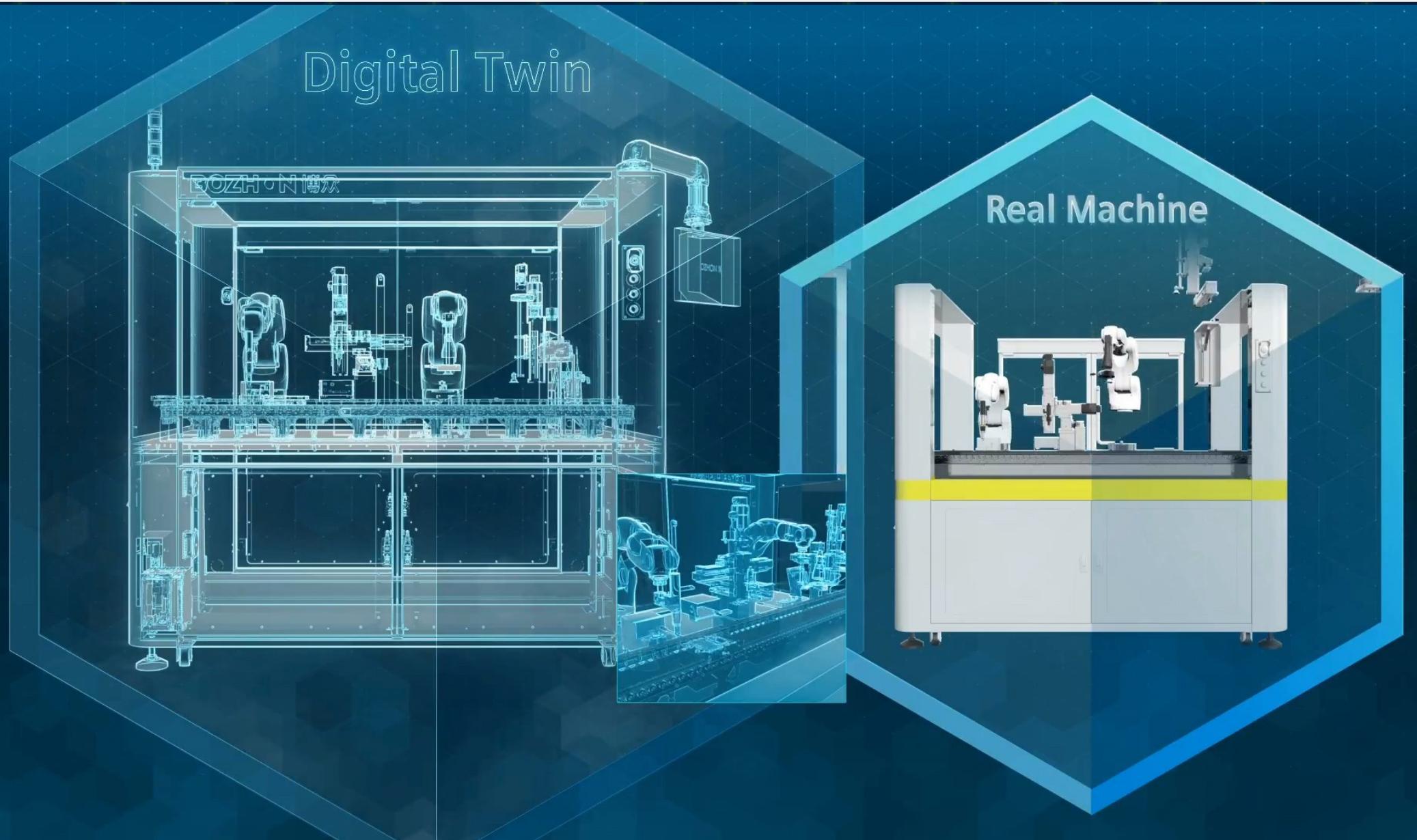


Cyber-Physical System (CPS)

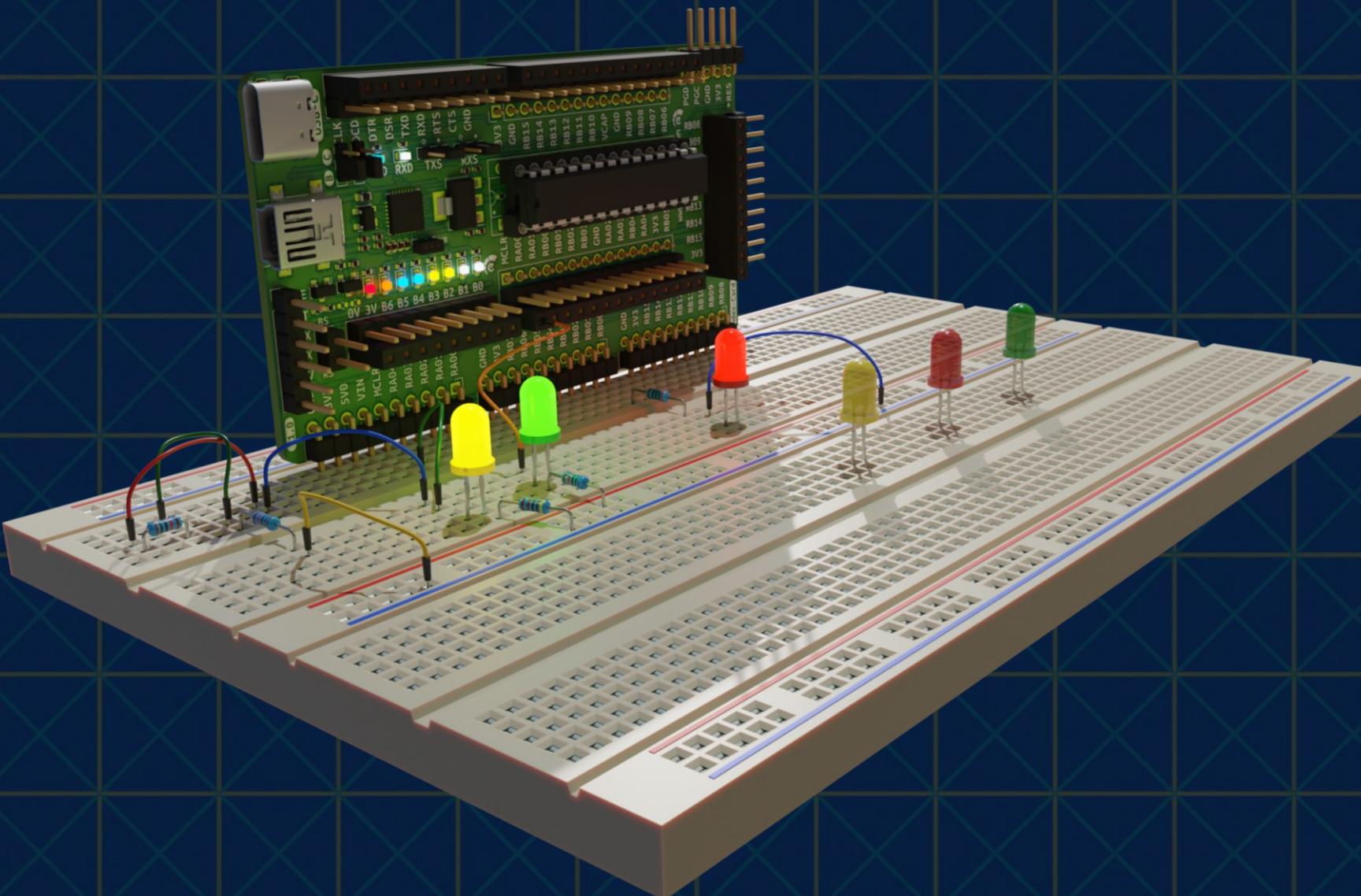


Real-time Control and Monitoring

Digital Twin (DT)



Digital Twin (DT) for Electronics and Embedded Systems



Node-RED and Digital Twin

The screenshot displays the ECC Flow-Based IDE v2.0.0 interface, featuring several windows:

- Node Editor**: Shows two flows, "Flow 1" and "Flow 2".
 - Flow 1** consists of four parallel "Beep" nodes (1kHz, 2kHz, 3kHz, 4kHz) connected to a "BEEP:50:1000:50" node, which then connects to a "data-bundle" node. This is followed by a "BEEP:2" node and a "data-bundle" node. Finally, it connects to four parallel "LED" nodes (0, 1, 2, 3) connected to "LED:0:INV", "LED:1:INV", "LED:2:INV", and "LED:3:INV" nodes respectively.
 - Flow 2** starts with a "WebSockets Read" node connected to a "Get ADC Value" node. This is followed by four parallel "Channel" nodes (A, B, C, D) connected to "Channel-A", "Channel-B", "Channel-C", and "Channel-D" nodes respectively. These then connect to a "data-bundle" node, which finally connects to a "BEEP:2" node.
- Dashboard**: A digital twin interface titled "LED" showing four LED controls labeled LED 0, LED 1, LED 2, and LED 3.
- MCU Monitor**: A 3D model of a microcontroller board with various components and status indicators. It includes a table of performance metrics:

Graphics	13.05
Physics	0.07
Engine	18.24
Frames/s	55.46
- Logs**: Four terminal-like panes showing logs from different servers:
 - HTTP Server**: Log entries for GET requests to /sim/public/assets/models/ and /mon/public/assets/fonts/.
 - Node-RED Server**: Log entries for websockets connections and receiver events.
 - MON WebSockets Server**: Log entries for ADC value updates.
 - MCU WebSockets Server**: Log entries for ADC value updates.

Web-based For IoT/IoT



The image shows a dual-screen setup illustrating a web-based IoT application development and deployment process.

Left Screen (Development Environment):

- Code Editor:** Visual Studio Code (Administrator) showing the file `main.js`. The code defines a function `main()` that creates a `Plotter` object with specific parameters and then clears it.
- Plotter Preview:** A plotter window showing two data series: a yellow line with circular markers and a green line with circular markers. The yellow line starts at (0, 0), goes up to (1, 100), drops to (2, 25), and then stays flat. The green line starts at (0, 0), goes up to (1, 12.5), drops to (2, 12.5), then to (3, 50), stays flat until (6, 50).

Right Screen (Web-based IoT Interface):

- Header:** ECC-Lab, IP: 127.0.0.1:5500/exampl...
- LED Control:** Four buttons labeled `LED3`, `LED2`, `LED1`, and `LEDO`. `LED1` is highlighted with a yellow cursor.
- PSW Status:** Four status indicators labeled `PSW #3` (green), `PSW #2` (yellow), `PSW #1` (pink), and `PSW #0` (blue).
- Meter and Counter:** Two circular meters. The left one shows a value of 74. The right one shows a value of 5. Below them is a text box: `%Duty Cycle: 74.000`, `Frequency: 5.000 Hz`, `T1: 0.200, T01: 0.148, T0f: 0.052`.
- Timing Diagram:** A waveform diagram showing digital signal levels over time.
- Propeller Logo:** A 3D rendering of a propeller with a blue and yellow logo on its hub.

Web Browser and Microcontroller Interfacing



The image shows a development environment with multiple windows:

- Visual Studio Code (Main Window):** Displays the file `main.js` from the path `examples > apps > web_apps > web06_knob_pwm > main.js`. The code uses WebSockets to interact with a microcontroller via UART. It defines a `Knob` class and a `main` function that initializes a `WSLink` object.
- Terminal:** Shows the output of the application, indicating successful PWM operations on pin 3 (pins 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41).
- Browser (ECC-Lab):** Shows four analog gauges with values 47, 72, 92, and 50. Each gauge has a red needle pointing to its value.
- Hardware Schematic:** A detailed schematic diagram of a PIC24FJ48GA002 microcontroller board. The board includes a PIC24FJ48GA002 microcontroller, various components like resistors, capacitors, and connectors, and several physical knobs and buttons labeled with letters A through S.
- Digital Oscilloscope:** A window titled "Digital Oscilloscope - OSC_PWM" showing waveforms for PWM signals. The left panel displays a schematic of a digital-to-analog converter (DAC) circuit, and the right panel shows the corresponding waveforms on a grid.



Let's Code

