

จาก Embedded System สู่ AI Robot



เมื่ออุปกรณ์ Embedded มี "สิ่งรับรู้... ตวงตา
สมอง และการสื่อสาร"

บทนำ - ทำไม Embedded System ต้องพัฒนาไปสู่ AI

Embedded System ในอดีตควบคุมมอเตอร์และอ่านค่า Sensor ได้ แต่ไม่สามารถ "เข้าใจสถานการณ์" ได้ แนวคิดใหม่: Embedded + AI + Sensor ทำให้อุปกรณ์มีความสามารถใหม่ 3 ด้าน:

PERCEPTION

รับรู้ (Perception): อุปกรณ์สามารถรับข้อมูลจากสภาพแวดล้อมผ่าน Sensor และกลั่นกรอง เพื่อสร้างความเข้าใจเชิงพื้นที่และบริบทรอบข้าง

UNDERSTANDING

วิเคราะห์ (Understanding): AI ประมวลผลข้อมูลที่รับมา วิเคราะห์รูปแบบจำแนกประเภท และเข้าใจความหมายของสถานการณ์แบบเรียลไทม์

DECISION & ACTION

ตัดสินใจ & ตอบสนอง (Decision & Action): ระบบสามารถเลือกการกระทำที่เหมาะสมและสั่งการ Actuator หรือ Motor ได้โดยอัตโนมัติ ไม่ต้องพึ่ง

Embedded AI คืออะไร

นิยาม

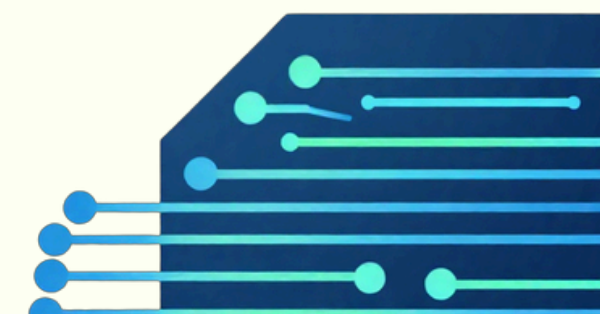
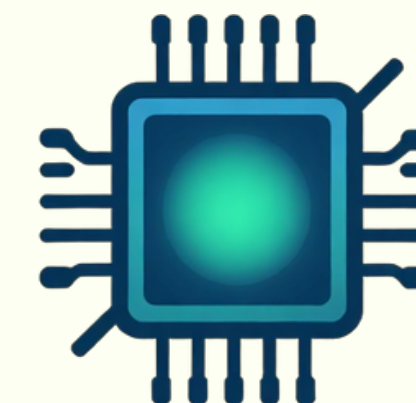
การนำโมเดล AI ไปประมวลผลบนอุปกรณ์ Embedded โดยตรง เช่น Microcontroller, SoC, Edge AI Device และ NPU Accelerator โดยไม่ต้องพึ่งพา Cloud

ข้อดี

ตัดสินใจแบบ Real-time, ลด Latency, ประหยัด Bandwidth, ทำงานแบบ Offline ได้, ความเป็นส่วนตัวสูง (Privacy) และลดการส่งข้อมูลขึ้น Server

HARDWARE

ตัวอย่าง: ESP32-S3 AI Camera, Raspberry Pi + AI Accelerator, NVIDIA Jetson Nano, Xilinx Kria KR260 สำหรับ Edge AI Processing

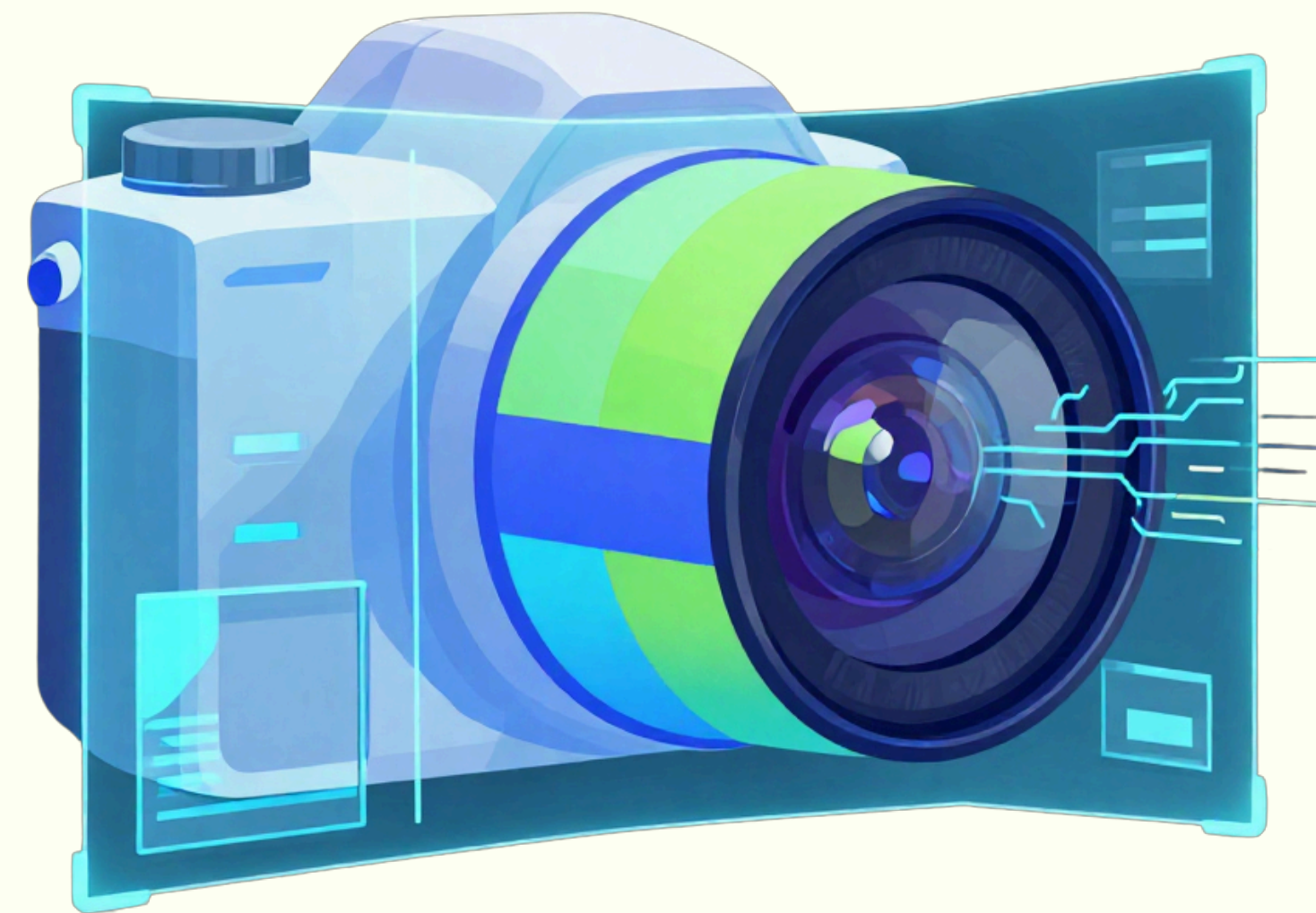


กล้องคือ "ดวงตา" ของระบบ AI

Sensor ทัวไปให้ค่าเดียว แต่กล้องให้ข้อมูลเชิงพื้นที่และบริบทที่ละเอียด AI ทำให้กล้องสามารถ:

- **Object Detection:** ตรวจจับวัตถุและระบุตำแหน่ง
- **Classification:** จำแนกประเภทของวัตถุ
- **Segmentation:** วิเคราะห์ระดับ Pixel เหมาะกับควัน ไฟ น้ำ
- **Pose Estimation:** วิเคราะห์ท่าทางและการเคลื่อนไหว

ตัวอย่างงานวิจัย: FireSpot-Seg สำหรับตรวจจับควันไฟป่าระยะเริ่มต้น



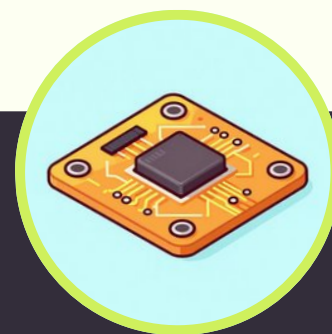
Sensor Fusion กล่องไม่ได้ทำงาน!! ด้วย ตัวคนเดียว

Camera + Vision AI



กล่องตรวจจับและวิเคราะห์พื้นที่ภาพ ให้ข้อมูลเชิงพื้นที่และบริบทที่ละเอียด ใช้ AI ประมวลผล Object Detection และ Segmentation

IMU + GPS Sensor



IMU ควบคุมสมดุลและทิศทาง GPS ระบุตำแหน่งพิกัดแม่นยำ ทั้งสองทำงานร่วมกันเพื่อการนำทางที่เสถียร

AI Fusion & Decision

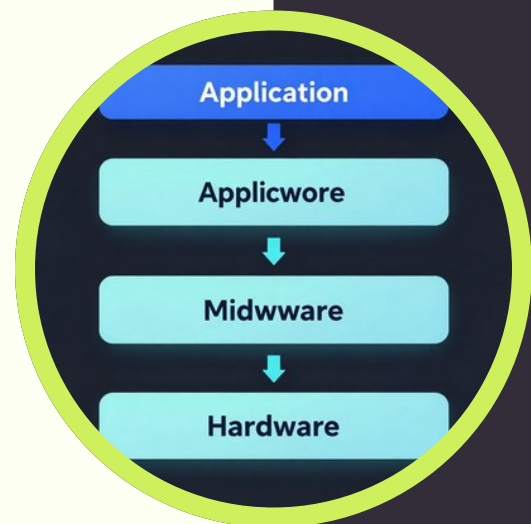


AI รวมข้อมูลจากทุก Sensor เข้าด้วยกัน วิเคราะห์สภาพแวดล้อม ลด Error และตัดสินใจสั่งการ Motor ได้แม่นยำในสภาพแวดล้อมจริง

จาก Embedded สู่ Robot Middleware (ROS: Robot Operating System)

ปัญหา

เมื่อ Robot ซับซ้อนขึ้น: มี Sensor
จำนวนมาก, Software หลายส่วน,
Hardware หลายชนิด ทำให้การพัฒนา
และดูแลระบบทำได้ยากขึ้นมาก



MIDDLEWARE

Middleware คือชั้นกลางระหว่าง
Hardware กับ Application. ROS ไม่ใช่
OS แต่เป็น Framework สำหรับพัฒนา
Software Robot รองรับ Hardware →
Middleware (ROS) → Application



Robot Operating System {ROS}

ส่วนประกอบหลักของ ROS และการทำงานร่วมกัน
ระหว่าง Node, Topic และ Message ในระบบ Robot

NODE

หน่วยประมวลผลอิสระใน ROS เช่น Camera Node รับภาพจากกล้อง, AI Detection Node ประมวลผล Vision, Motor Control Node สั่งการขับเคลื่อน แต่ละ Node ทำงานแยกกันและสื่อสารผ่าน Topic

TOPIC

ช่องทางการสื่อสารแบบ Publish/Subscribe ระหว่าง Node เช่น /camera/image สำหรับส่งภาพ, /detection/result สำหรับผลลัพธ์ AI และ /cmd_vel สำหรับควบคุมมอเตอร์

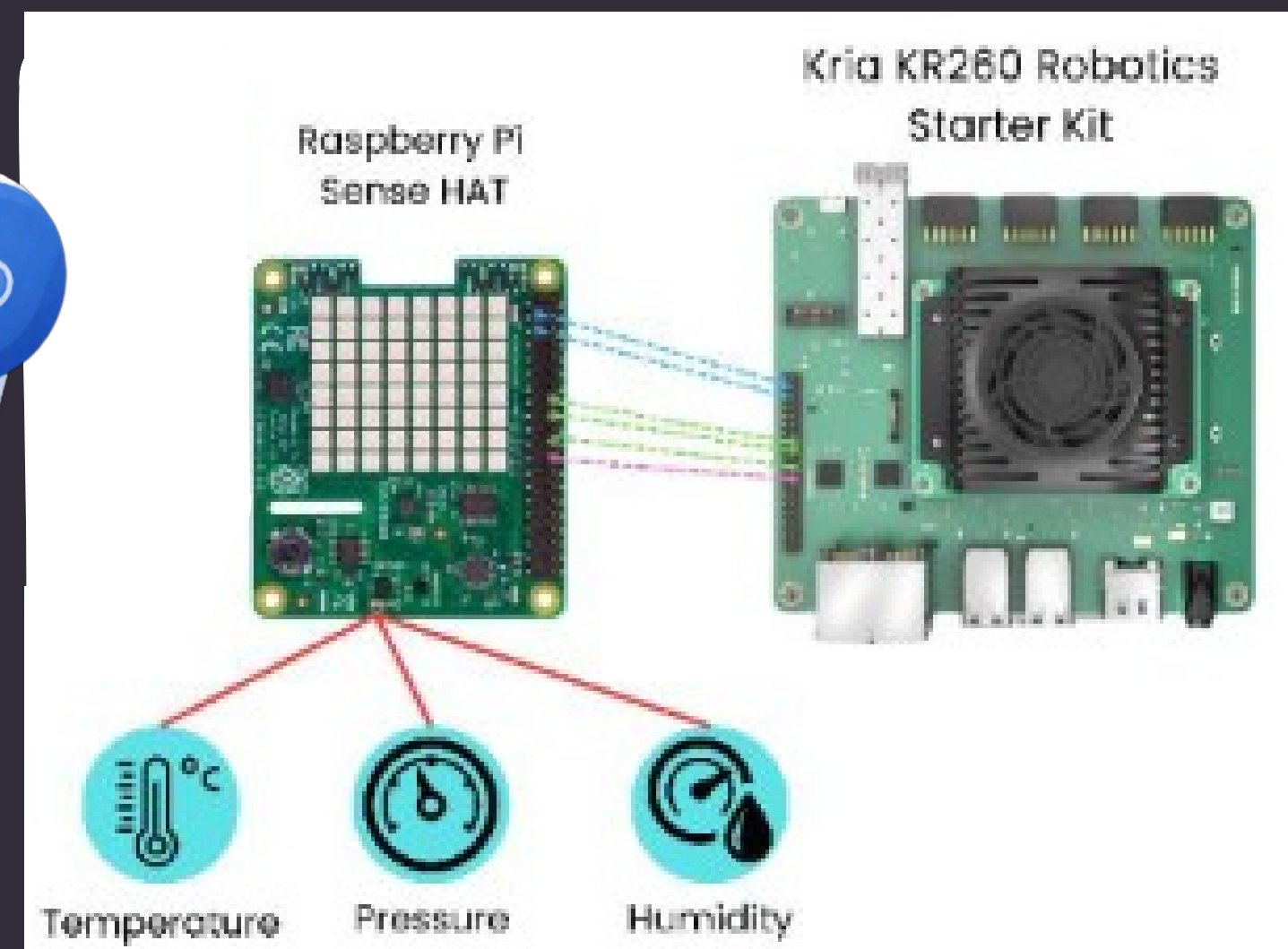
MESSAGE

รูปแบบข้อมูลที่ส่งผ่าน Topic เช่น sensor_msgs/Image สำหรับภาพกล้อง, std_msgs/String สำหรับข้อความ และ geometry_msgs/Twist สำหรับคำสั่งเคลื่อนที่ของหุ่นยนต์

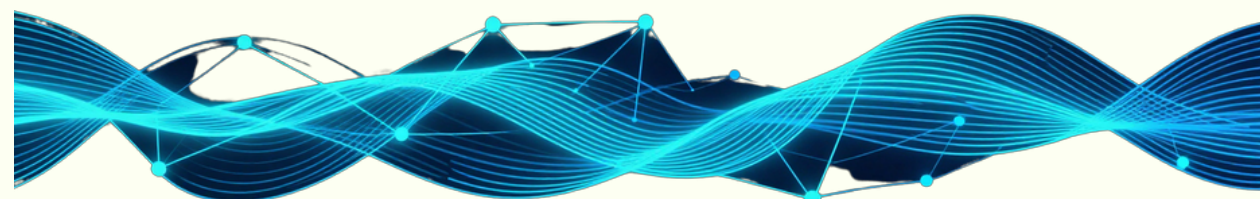
ตัวอย่างงาน วิจัย ROS - Kria KR260

แพลตฟอร์ม KR260 เป็น Edge AI Device ที่รวม FPGA และ ROS
เหมาะกับ Real-time Vision Processing งานวิจัยได้ทดสอบการเชื่อมต่อ
ต่อ Microcontroller, Sensor Interface และ ROS บนฮาร์ดแวร์จริง

ผลการประเมิน Interface ของ Kria KR260 Robotics
Starter Kit:



AI + ROS + Camera สถาปัตยกรรมแบบครบวงจร



ดวงตา

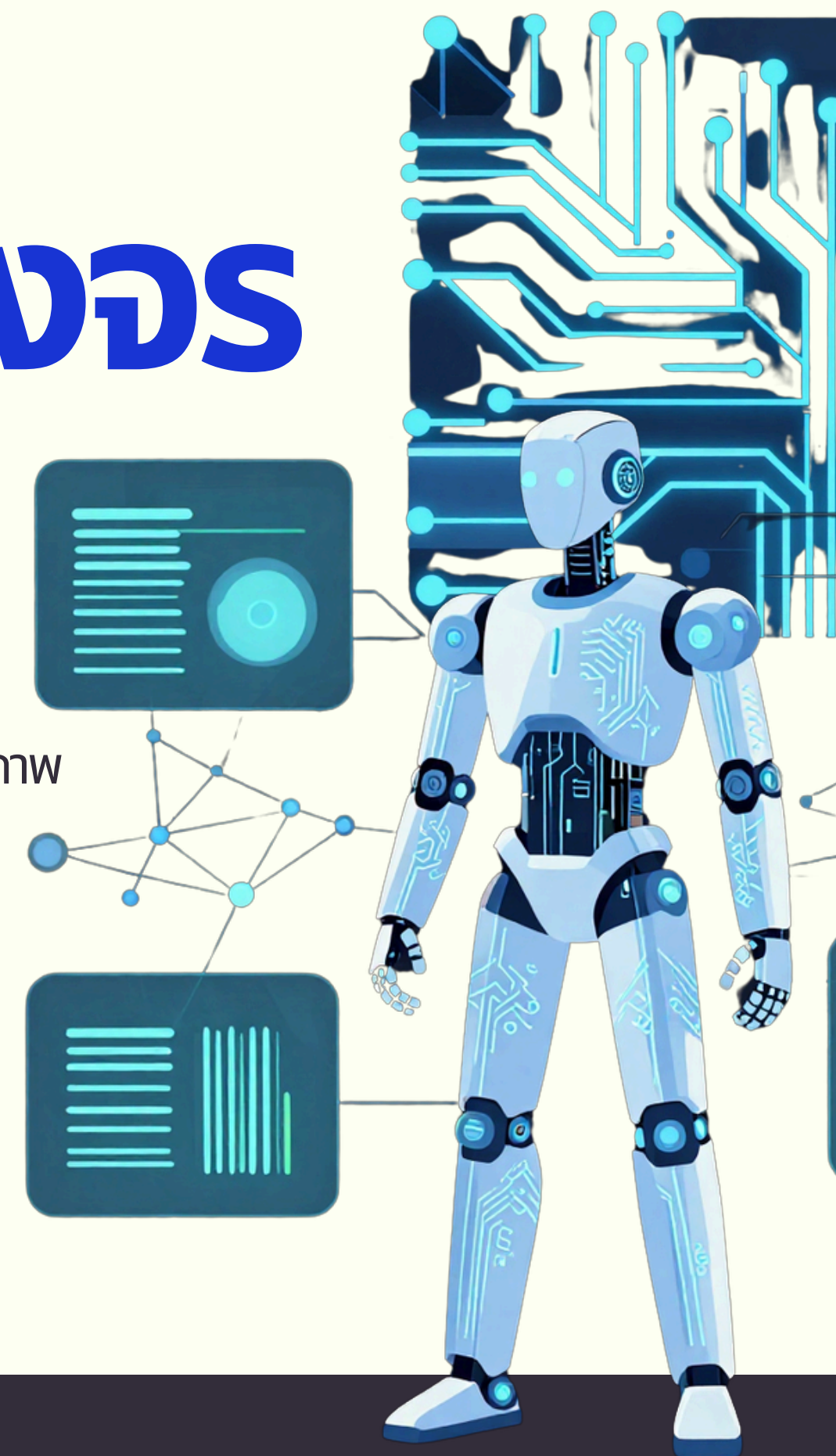
Camera คือ "ดวงตา" ของระบบ — รับภาพผ่าน ROS Camera Node ส่งข้อมูลภาพเข้าสู่โมเดล Vision AI เช่น YOLO หรือ CNN เพื่อตรวจจับวัตถุแบบเรียลไทม์

สมอง

AI Model คือ "สมอง" — ประมวลผลภาพจาก Camera วิเคราะห์และตัดสินใจผ่าน Decision Node แล้วส่งคำสั่งไปยัง Robot Controller และ Motor

ระบบประสาท

ROS คือ "ระบบประสาท" — เชื่อมต่อทุก Node ผ่าน Topic และ Message ส่งข้อมูลผ่าน MQTT ไปยัง Dashboard หรือศูนย์ควบคุม



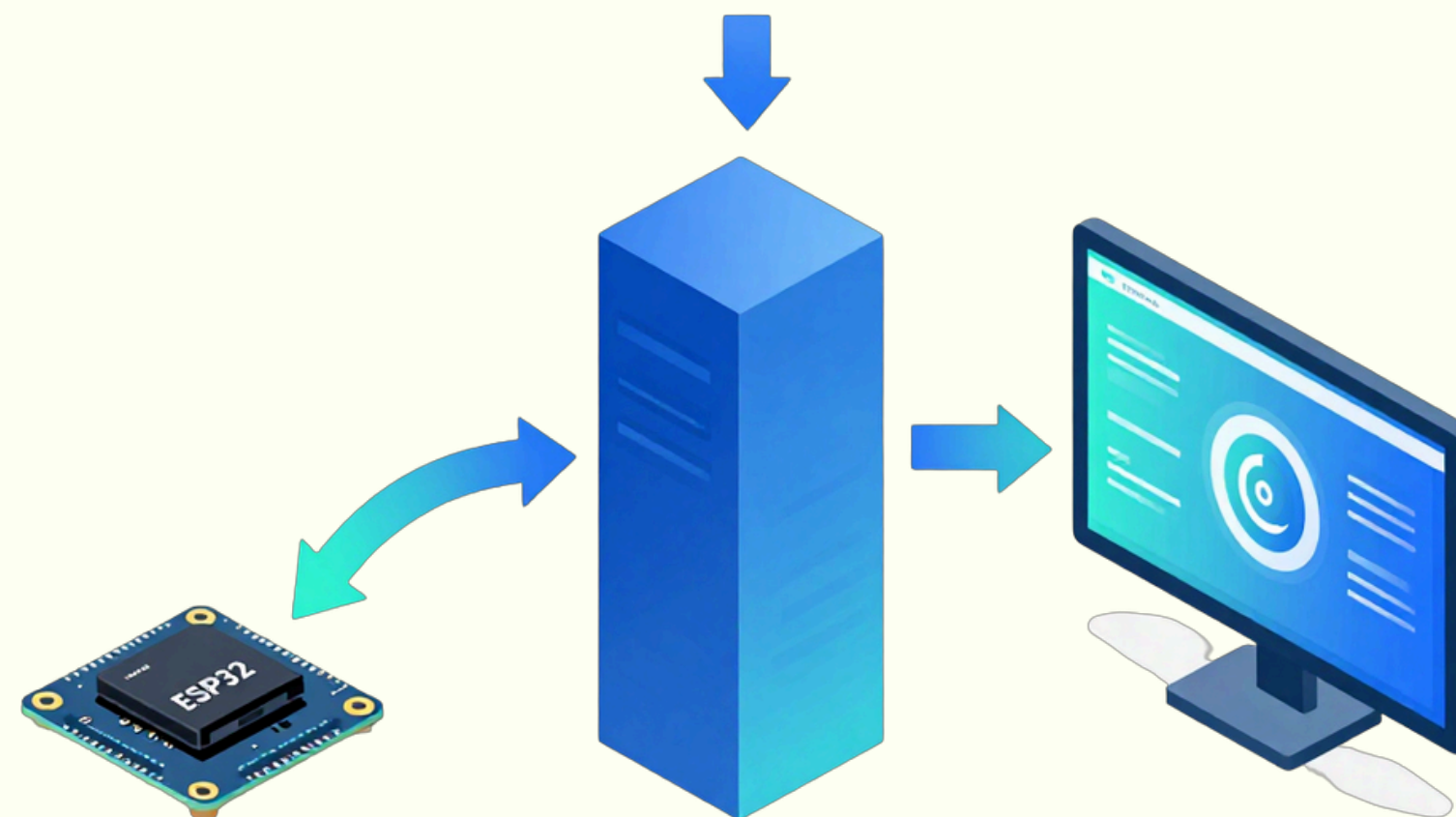
MQTT กับ Embedded AI

MQTT เหมาะกับ IoT Robot เพราะเป็นโปรโตคอลเบา ประหยัด bandwidth เหมาะกับอุปกรณ์ทรัพยากรจำกัด

Architecture: Sensor Device (ESP32, Camera, IMU) → MQTT Publish → MQTT Broker → Dashboard/AI Server

ตัวอย่าง Topic: robot/camera/status

PAYLOAD : {
"OBJECT" : "PERSON",
"CONFIDENCE" : 0.95,
"LOCATION" : "FRONT"}



สรุป

วิวัฒนาการ: Embedded System →
Embedded AI → Vision AI → ROS Robot →
Autonomous Intelligent System
กล้อง = ดวงตา | AI = สมอง | ROS = ระบบประสาท |
Hardware = ร่างกาย
"อุปกรณ์ Embedded ที่มี 'ดวงตา สมอง และการสื่อสาร' คืออนาคตของเทคโนโลยีอัจฉริยะ"



Profile Name

Profile Type

MQTT Broker Profile Settings

Broker Address

Broker Port

Client ID

General **User Credentials** **SSL/TLS** **Proxy** **LWT**

User Name

Password

KANTIT.COM/DF609