EMBEDDED AI & ROBOTICS

สำหรับนักศึกษาและการอบรม

จาก Embedded System สู่ AI Robot





Aj.กานต์



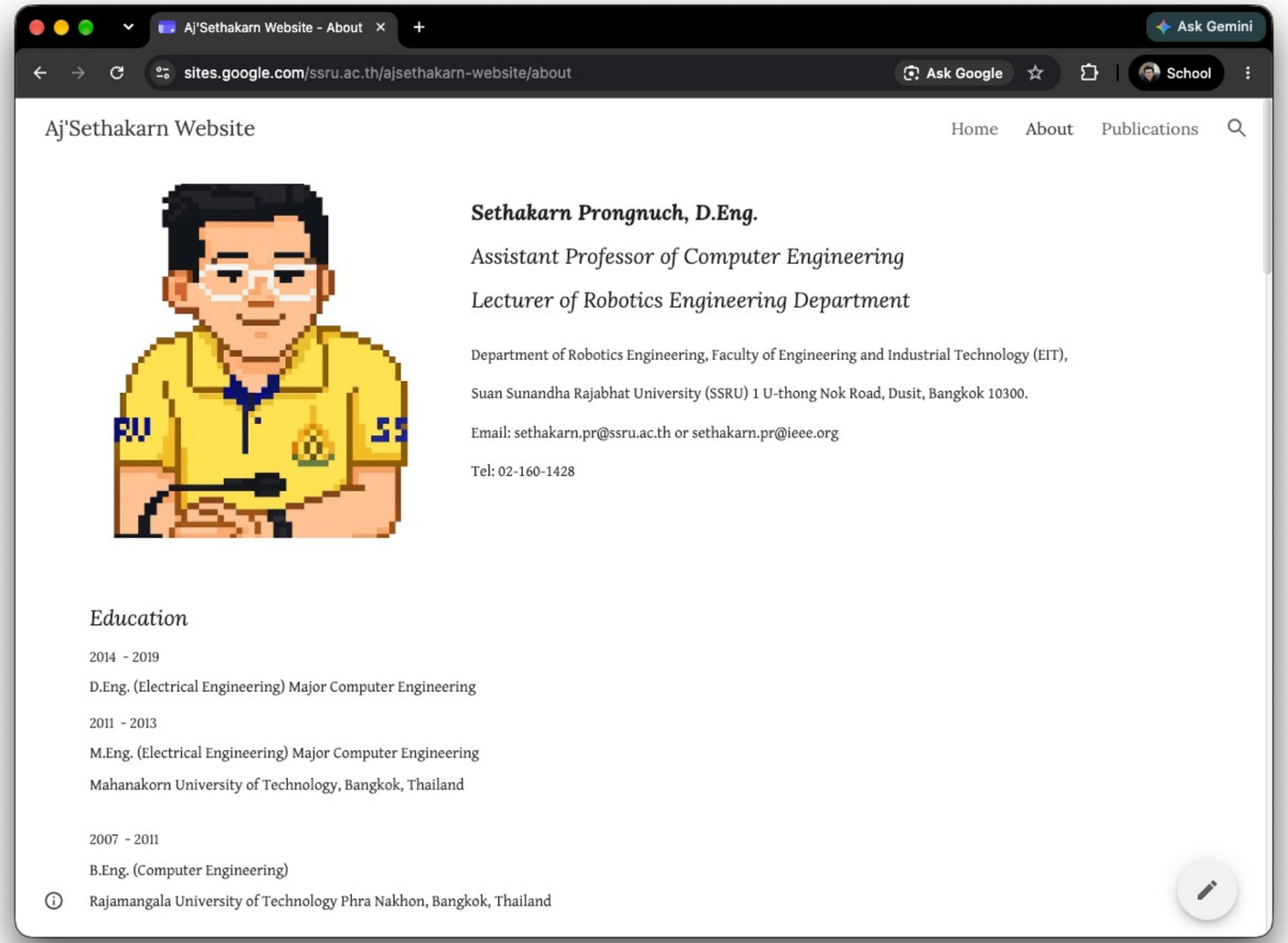
Aj.บิวตี้



Aj.Zax

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

แนะนำวิทยาการ

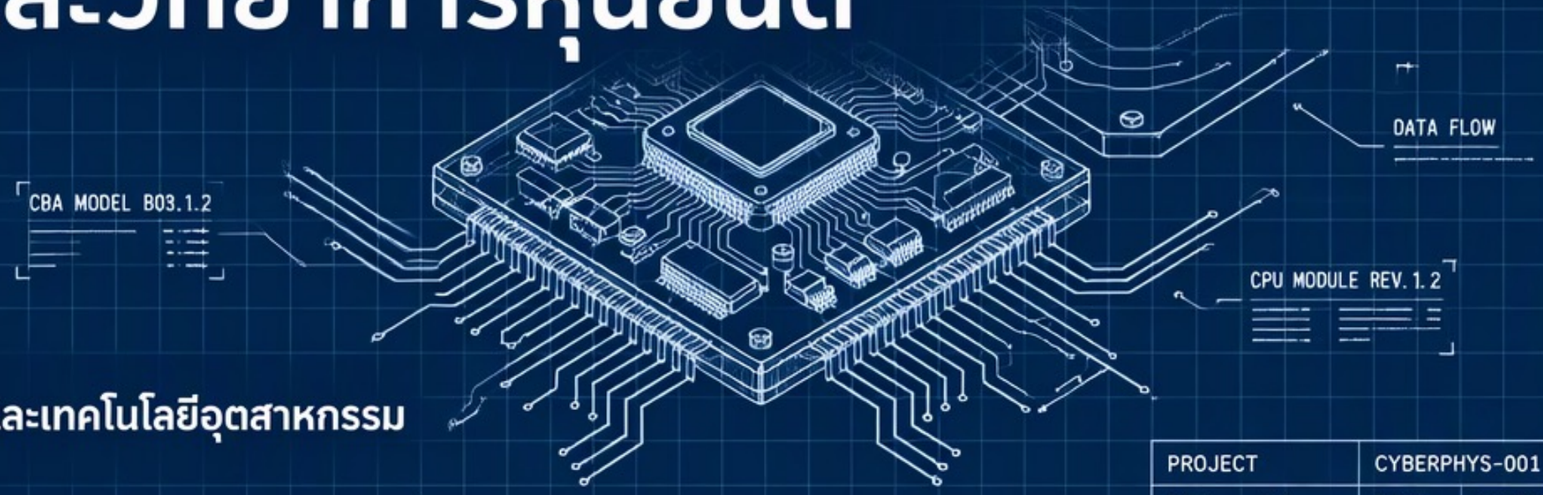


bitly



The Cyber-Physical Blueprint

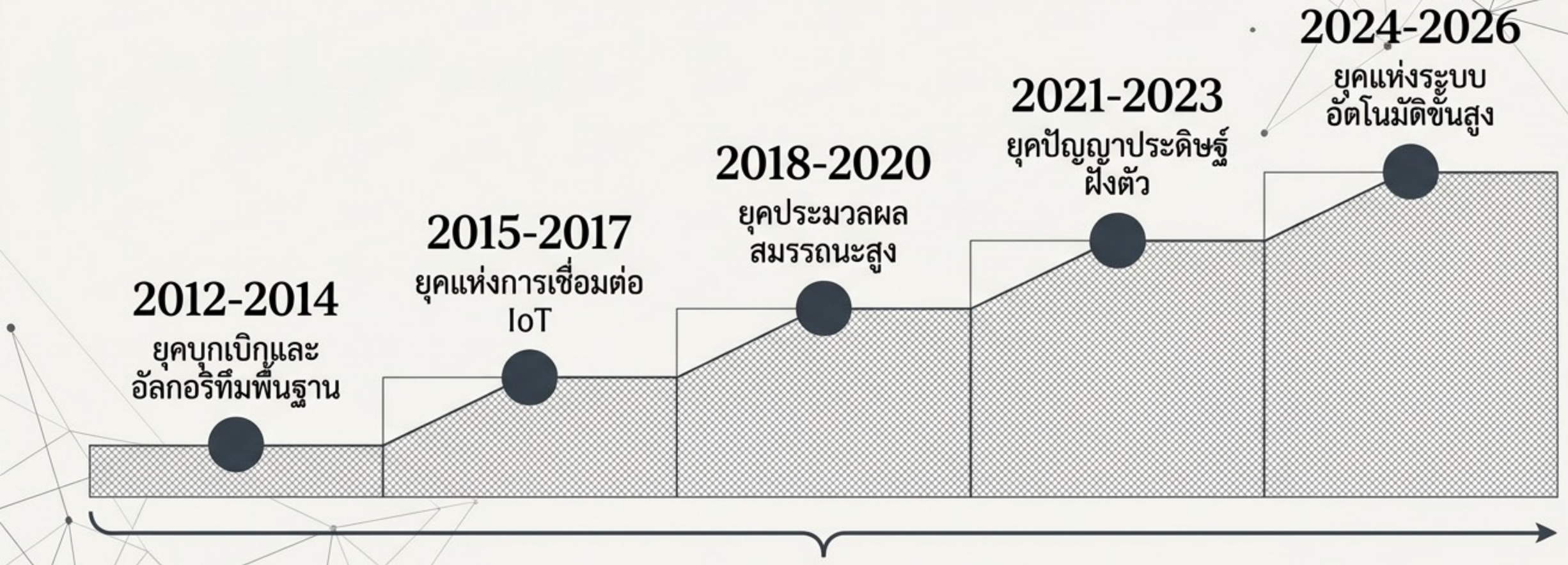
สถาปัตยกรรมการบูรณาการระบบสมองกลฝังตัว ปัญญาประดิษฐ์ และวิทยาการหุ่นยนต์



นำเสนอโดย
ผศ. ดร.เศรษฐกาล โปร่งนุช
สาขาวิชาวิศวกรรมหุ่นยนต์ คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏสวนสุนันทา

PROJECT		CYBERPHYS-001	
REV.	1.0	DATE	2026-01-01

วิวัฒนาการ 5 ยุค: ขับเคลื่อนอนาคตด้วยเทคโนโลยี



การเปลี่ยนผ่านจากระบบควบคุมพื้นฐาน สู่ ปัญญาประดิษฐ์แบบกระจายศูนย์ (Edge AI)

Gemini Notebook

การจัดกลุ่มหมวดหมู่งานวิจัย 72 บทความ (Research Categorization)

งานวิจัยด้าน Embedded AI, Robotics, Computer Vision, IoT & STEM Education (2012 - 2026)

1. ระบบหุ่นยนต์ ยานยนต์อัตโนมัติ & โดรน (Robotics, Autonomous Vehicles & Drones) — 16 บทความ

- หุ่นยนต์บริการ & ประชาสัมพันธ์: SSRU PR Robot (CE-R1) ควบคุมด้วย Raspberry Pi & Smartphone (3 บทความ)
- หุ่นยนต์ร่วมปฏิบัติงาน & อุตสาหกรรม: Mobile Collaborative Robot (Youibot P200 + Dobot CR5) & ROS2 Arm Control (3 บทความ)
- หุ่นยนต์เคลื่อนที่สำรวจ: หุ่นยนต์ 3D Printed สองล้อ, FPV Tank Rover ตีนตะขาบ, Spherical Magnetic Robot (4 บทความ)
- ยานโร็คนขับ & โดรน: Mini-UAV / Selfie Drone สำรวจในอาคาร & รถหุ่นยนต์ AI โรณนับ PiCar-X (3 บทความ)
- ระบบประมวลผลสำหรับหุ่นยนต์: การประเมินประสิทธิภาพบอร์ด Kria KR266 & สิ่งการแชนกลร่วมกับ Kinect V2 (3 บทความ)

2. สถาปัตยกรรมสมองกลฝังตัว & คลัสเตอร์ประมวลผล (Embedded Architecture & HPC Clusters) — 15 บทความ

- สถาปัตยกรรมออกแบบร่วม (Co-design): Reconfigurable System บน FPGA, System-on-Chip (SoC) & Ethernet (4 บทความ)
- สถาปัตยกรรมคำนวณแบบผสมผสาน (HSA): Heterogeneous Computing, HSA Accelerator & OpenCL/OpenMP (3 บทความ)
- คลัสเตอร์สมองกลฝังตัว (Embedded HPC): High-Performance SBC Cluster 18 Cores / Parallella (3 บทความ)
- อัลกอริทึม & การจัดสรรงาน: Task Scheduling Heuristics ในระบบกระจายตัว IoT & Parallel Processing Optimization (3 บทความ)
- การประมวลผลสัญญาณขอมาน: Adaptive LMS-FIR Filter & Sobel Edge Detection Acceleration (2 บทความ)

3. คอมพิวเตอร์วิชัณน์ & AI ประยุกต์ (Computer Vision, Deep Learning & Embedded AI) — 15 บทความ

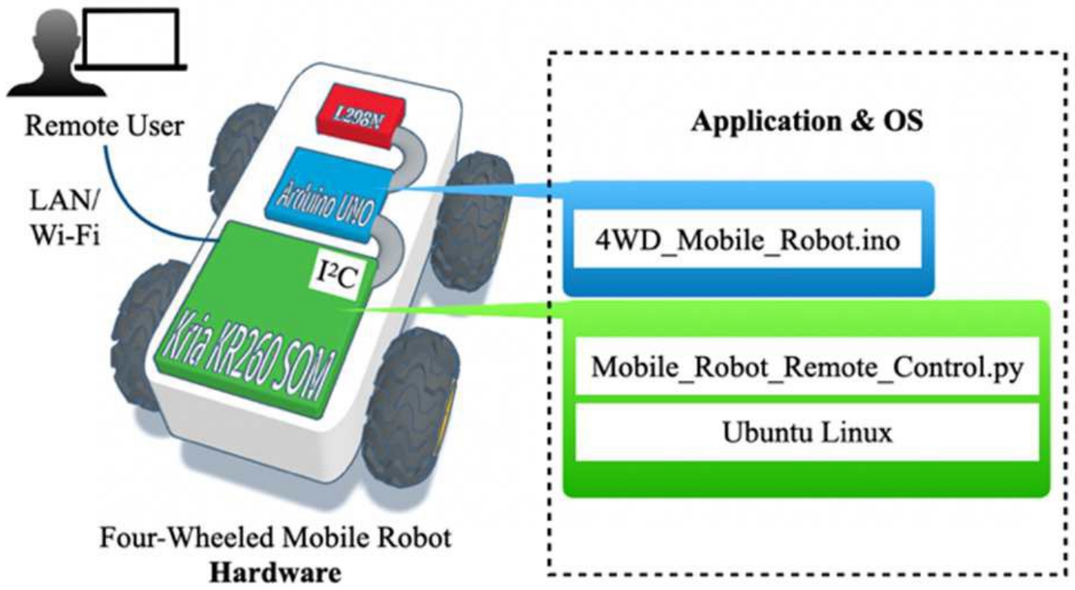
- การตรวจจับป้ายจราจร & ยานพาหนะ: Traffic Sign Recognition บนทางหลวงด้วย YOLOv11 & Deep Learning จำแนกรถ 4 ล้อ (4 บทความ)
- ระบบติดตามใบหน้า & ท่าทางมือ: MediaPipe Gesture Tracking, Hand Detection & Real-time Object Tracking (4 บทความ)
- AI ประยุกต์เพื่อการเกษตร & ฟาร์ม: ผู้ฟังไม้อัตโนมัติคัดแยกเชื้อด้วย CNN & เครื่องผสมอาหารลุกโค่อัจฉริยะ (3 บทความ)
- ระบบความปลอดภัย & สังบดกรกร่ม: Face Spoofing Detection (YSAF Framework) & Drowsiness Safety Detection (4 บทความ)

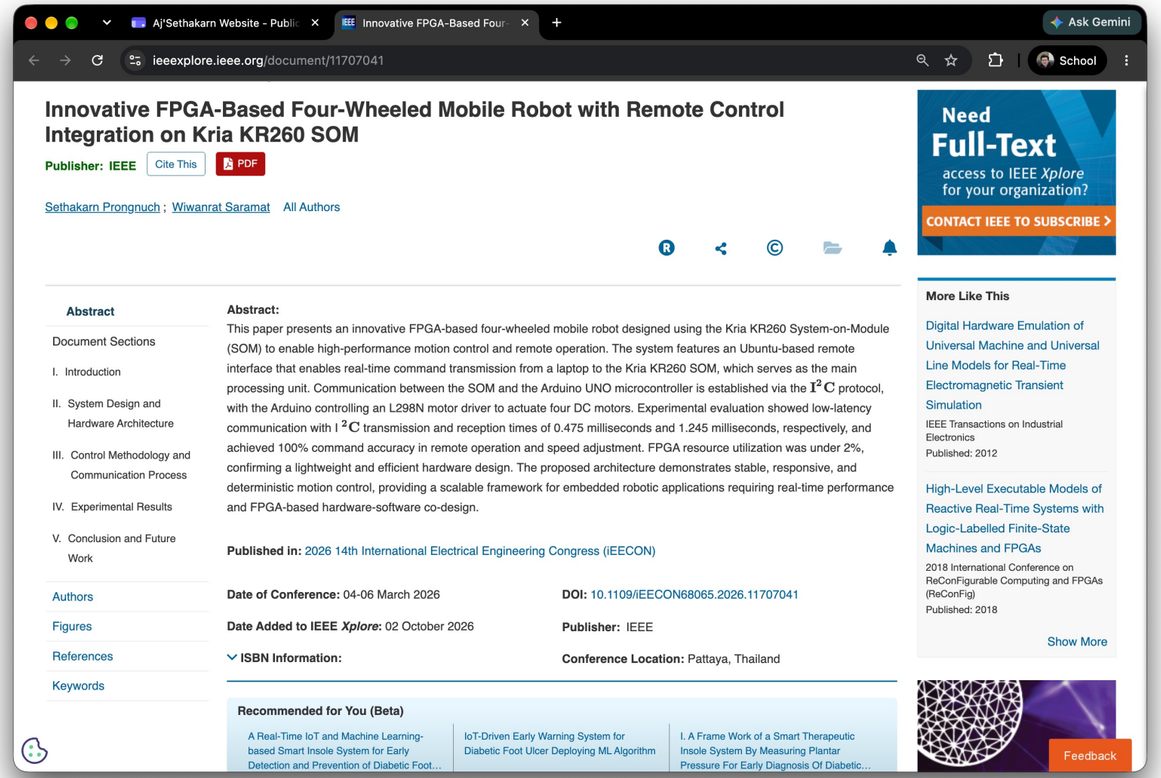
4. IoT เกษตรอัจฉริยะ & ระบบควบคุมอัจฉริยะ (IoT, Smart Agriculture & Voice Control) — 14 บทความ

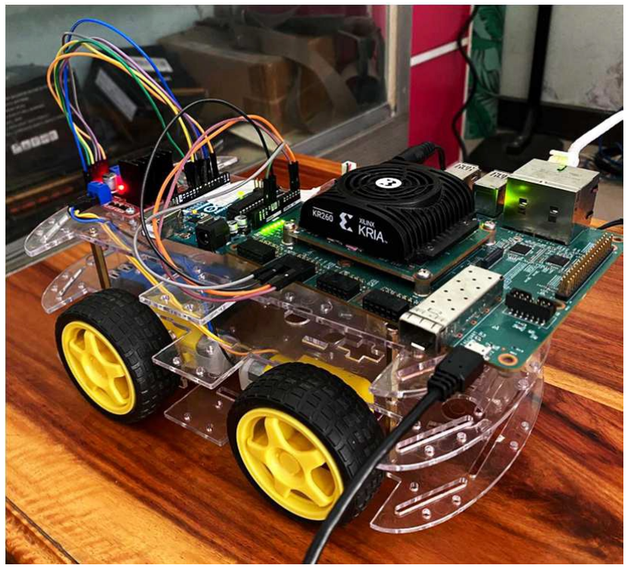
- ระบบควบคุมด้วยเสียง (Voice UI): Voice-Controlled ROS2 Kilted Kaiju, Active Park Assist & Car Parking (4 บทความ)
- IoT เกษตรอัจฉริยะ: ระบบลดความชื้นข้าวเปลือกควบคุมด้วย NodeMCU, ควบคุมความชื้นโรงเรือนกล้วยไม้ & ผู้พิค่มูโร่ Web App (4 บทความ)
- อัลกอริทึมหาตำแหน่งดัดดั้ง: Genetic Algorithm (GA) สำหรับค้นหาตำแหน่งดัดดั้ง WiFI Access Point (2 บทความ)
- อุปกรณ์อัจฉริยะ & สภาพแวดล้อม: Smart Basket ระบบตระกร้าอัจฉริยะ & ระบบลายโปรงจกเตอร์อัตโนมัติด้วย IR (4 บทความ)

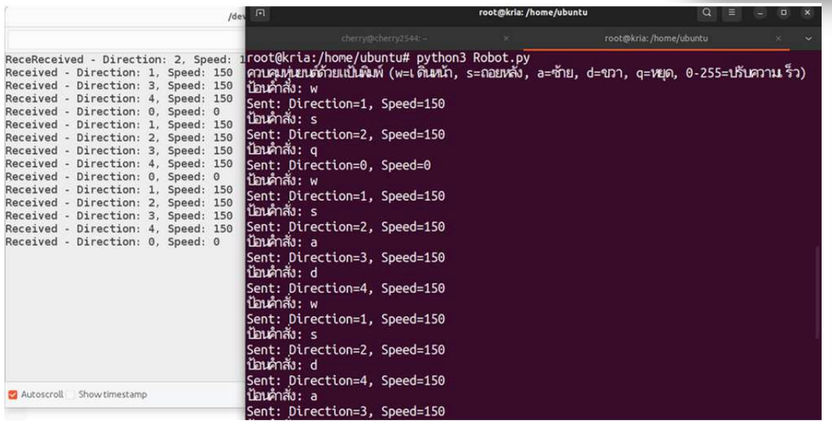
5. การศึกษาวิทยาการหุ่นยนต์, STEM & นวัตกรรมสื่อเรียนรู้ (Robotics Education, STEM & Learning Platforms) — 12 บทความ

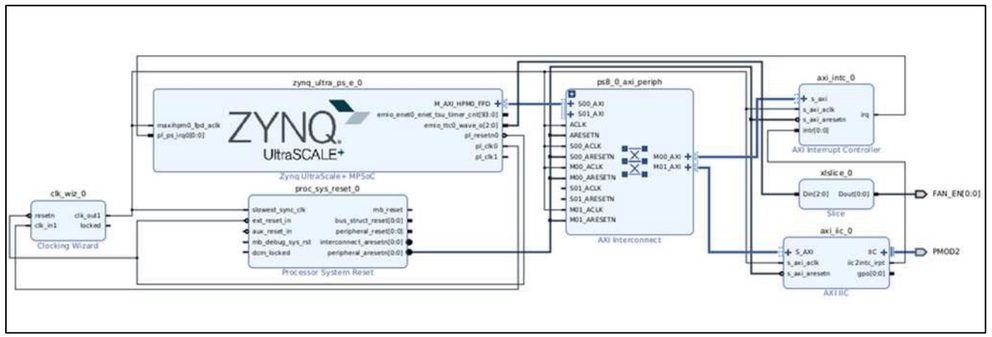
- ค่ายการเรียนรู้ STEM ออนไลน์: Robot & Packaging Design Camp & Robot & Real Estate Management Camp (4 บทความ)
- ความร่วมมือข้ามศาสตร์ (Bilateral Learning): Bilateral Cooperative Training วิศวกรรมศาสตร์ & สถาปัตยกรรมศาสตร์ (3 บทความ)
- แพลตฟอร์มการสอน & Visual Programming: BlockMicc Programming Visual Platform & MSStack IoT Platform (3 บทความ)
- นวัตกรรมสื่อการสอนสถาปัตยกรรมคอมพิวเตอร์: Technology-Assisted STEM Learning Space & SoC Kits (2 บทความ)











S. Prongnuch and W. Saramat, "Innovative FPGA-Based Four-Wheeled Mobile Robot with Remote Control Integration on Kria KR260 SOM," *2026 14th International Electrical Engineering Congress (iEECON)*, Pattaya, Thailand, 2026, pp. 1-5, doi: 10.1109/iEECON68065.2026.11707041.

Layer 4: หุ่นยนต์เชิงกายภาพ (Physical AI & Robotics)

The Body: AMR, UAV, Actuators

Layer 3: ปัญญาประดิษฐ์บนอุปกรณ์ปลายทาง (Edge AI)

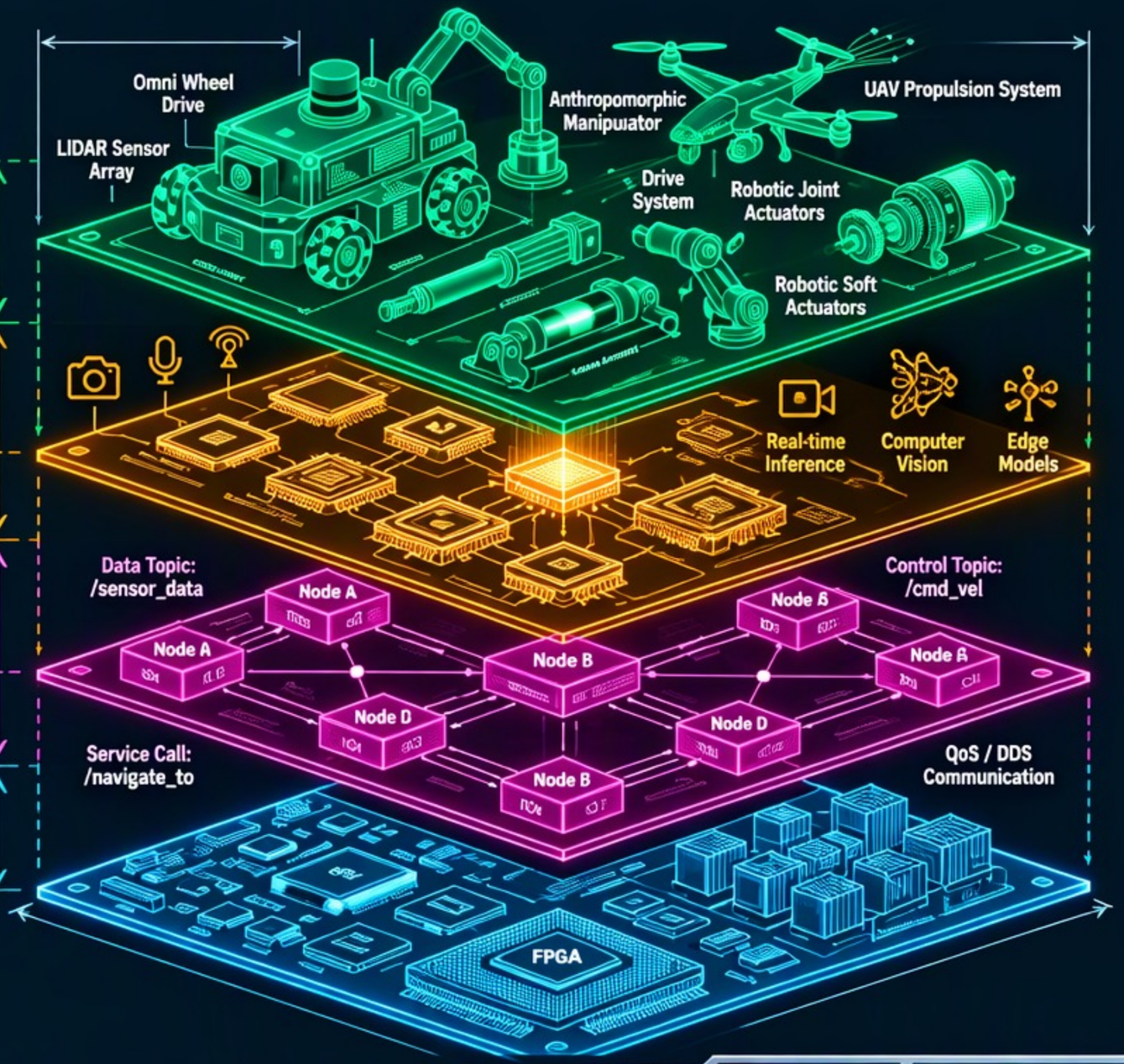
The Brain: MediaPipe, YOLO, Deep Learning

Layer 2: ROS / ROS2

The Nervous System: Middleware, Communication, Nodes, Topics

Layer 1: ระบบสมองกลฝังตัว (Embedded Systems)

The Foundation: Hardware logic, FPGAs, Clusters

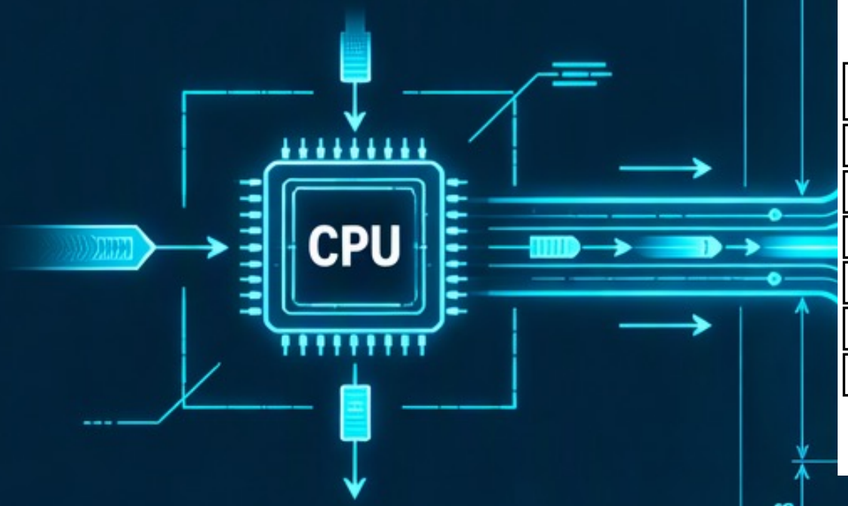


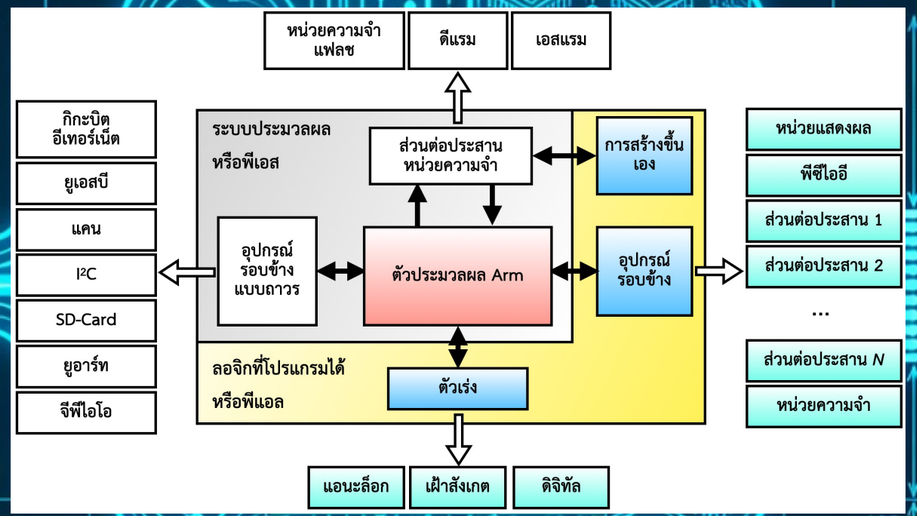
จากฮาร์ดแวร์สู่การเคลื่อนไหว: การผสาน 4 เทคโนโลยีเพื่อสร้างระบบอัตโนมัติที่สมบูรณ์แบบ

Layer 1: ระบบสมองกลฝังตัว (The Hardware Foundation)

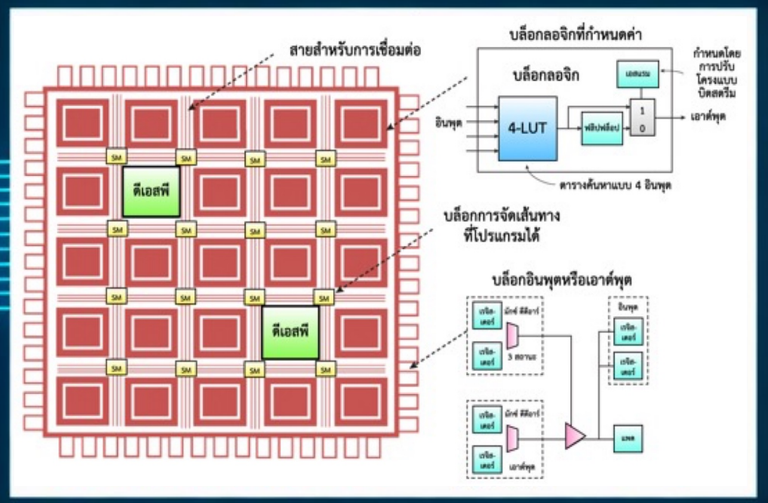
The FPGA Split-Brain

ยืดหยุ่นแต่อนุกรม (Flexible but Sequential)
CPU / ARM

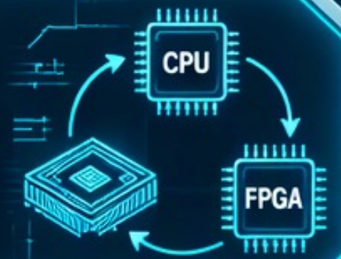




ตรรกะคู่ขนาน (Parallel Logic Gates)
FPGA



Heterogeneous Computing
การรวมหน่วยประมวลผลต่างชนิด (CPU, Coprocessor, FPGA) ในระบบเดียว เพื่อประสิทธิภาพสูงสุดและประหยัดพลังงาน (อ้างอิง: Parallella Board & ZYNQ SoC)

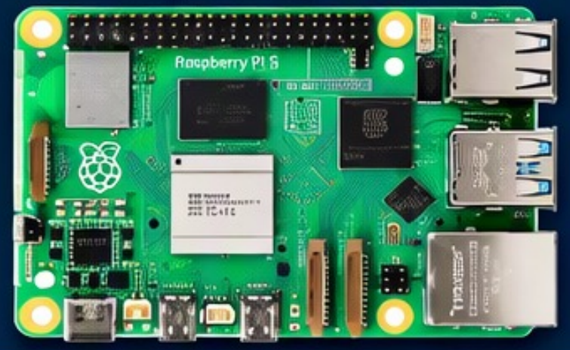


Hardware-Software Co-design
การออกแบบฮาร์ดแวร์ (Vivado) และซอฟต์แวร์ (Vitis) ควบคู่กันบน Kria KR260 เพื่อลดความซ้ำซ้อนของระบบ



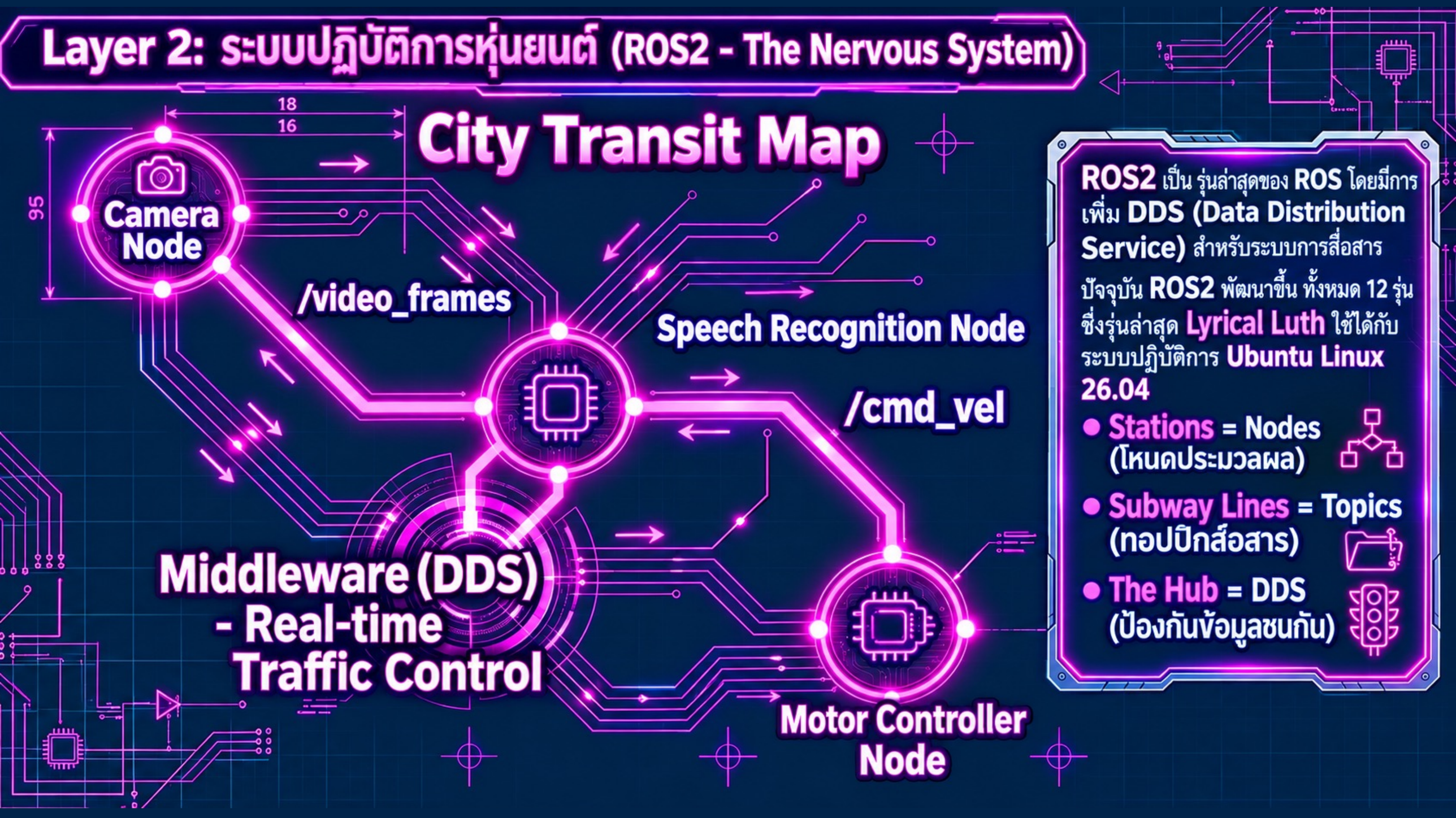


เมทริกซ์สถาปัตยกรรมการประมวลผล (The Compute Architecture Matrix)

	Homogeneous (CPU-based)	Heterogeneous (SoC/FPGA)	Embedded Cluster
ตัวอย่างบอร์ด	Raspberry Pi 4 / 5 	Kria KR260 	Parallella (16-core Epiphany) 
จุดเด่น	ใช้งานง่าย, ราคาถูก (Accessible, Low Cost)	ความเร็วสูง, ปรับแต่งตรรกะได้ (High Speed, Configurable Logic)	ประมวลผลขนานขั้นสูง (Massive Parallelism)
กรณีศึกษา	ตรวจจับป้ายจราจร (Traffic Sign Detection)	ระบบควบคุมมอเตอร์แบบ Real-time	การคำนวณทางวิทยาศาสตร์ (Scientific Computing)

Insight: การเลือกสถาปัตยกรรมไม่ได้อยู่ที่ความแรงสูงสุด แต่อยู่ที่ความเหมาะสมของพลังงาน-ความหน่วง-ต้นทุน (Power-Latency-Cost)

Layer 2: ระบบปฏิบัติการหุ่นยนต์ (ROS2 - The Nervous System)



ROS2 เป็น รุ่นล่าสุดของ **ROS** โดยมีการเพิ่ม **DDS (Data Distribution Service)** สำหรับระบบการสื่อสาร

ปัจจุบัน **ROS2** พัฒนาขึ้น ทั้งหมด 12 รุ่น ซึ่งรุ่นล่าสุด **Lyrical Luth** ใช้ได้กับระบบปฏิบัติการ **Ubuntu Linux 26.04**

- **Stations** = Nodes (โหนดประมวลผล)
- **Subway Lines** = Topics (ทอปปิกส์อสาร)
- **The Hub** = DDS (ป้องกันข้อมูลชนกัน)

กรณีศึกษา: การควบคุมหุ่นยนต์ด้วยเสียงผ่าน ROS2 (Voice-Controlled Kilted Kaiju)

1 Speech Node (ROS2)
รับเสียงมนุษย์และแปลงเป็นข้อความ (เช่น ‘เดินหน้า” -> FWD)

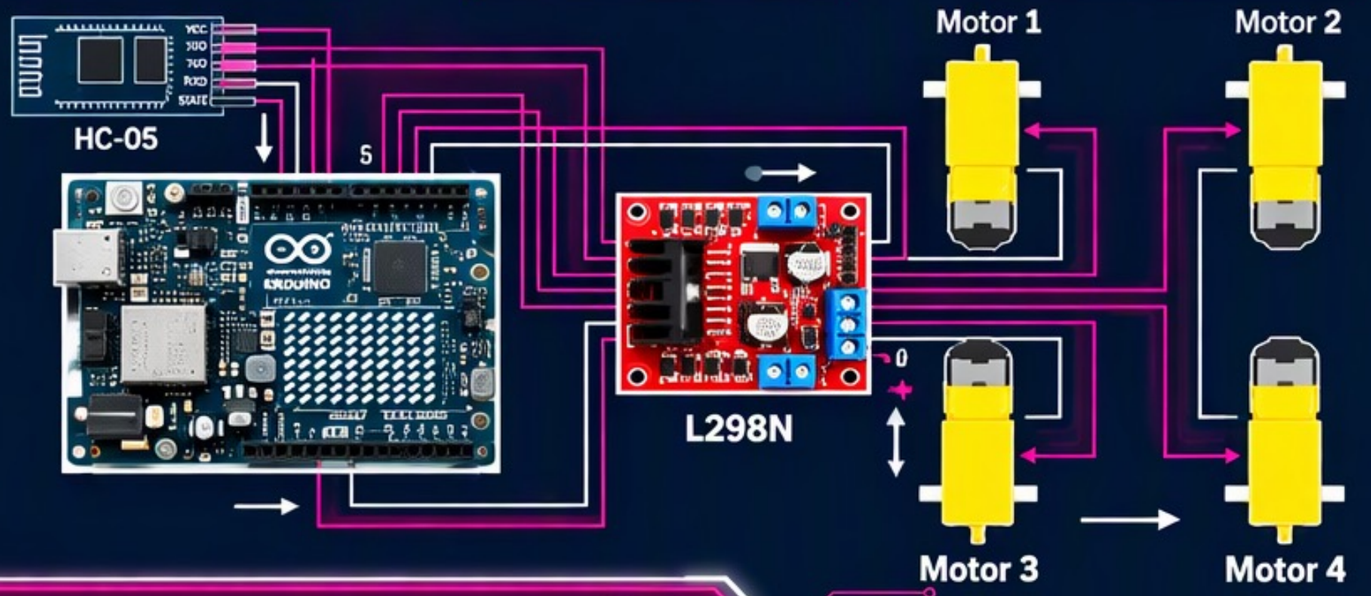
2 Bridge Node (ROS2)
แปลงคำสั่งเป็น Serial Data ส่งผ่าน Topic /bt/outgoing

3 Wireless Link
ส่งผ่าน Bluetooth (HC-05)

4 Microcontroller
Arduino รับข้อมูล แปลงเป็น PWM

5 Actuation
L298N Driver สั่งการมอเตอร์ 4 ล้อ



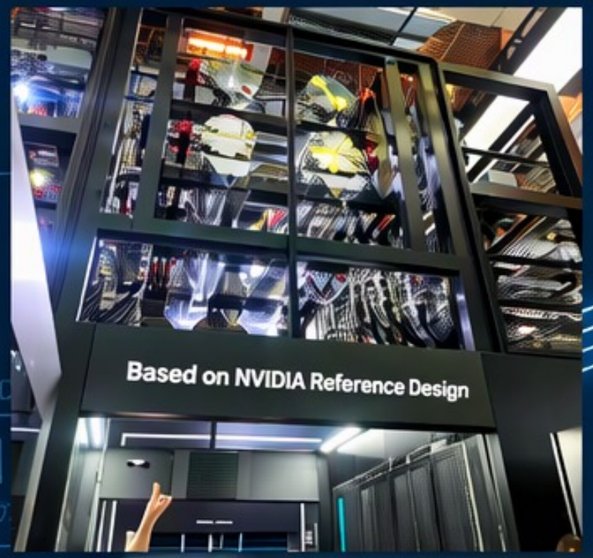


Insight: ROS2 จัดการความซับซ้อนระดับสูง ปล่อยให้ Arduino จัดการฮาร์ดแวร์ระดับต่ำ

Layer 3: ปัญญาประดิษฐ์บนอุปกรณ์ปลายทาง (The Shift to Edge AI)

Cloud AI

- **ความหน่วง (Latency):** สูง (ต้องรอเครือข่าย)
- **แบนด์วิดท์ (Bandwidth):** พึ่งพาอินเทอร์เน็ตสูง
- **ความเป็นส่วนตัว (Privacy):** ข้อมูลถูกส่งออกนอกระบบ



Edge AI

- **ความหน่วง (Latency):** ต่ำมาก (Real-time บนตัวอุปกรณ์)
- **แบนด์วิดท์ (Bandwidth):** ทำงานได้แบบออฟไลน์ (Offline-capable)
- **ความเป็นส่วนตัว (Privacy):** ปลอดภัย ข้อมูลไม่รั่วไหล (Data stays on-device)



หลักฐานอ้างอิง: งานวิจัย Traffic Sign Recognition บน Raspberry Pi 5 พิสูจน์ว่า Edge AI สามารถตรวจจับป้ายจราจรได้แบบ Real-time โดยไม่ต้องพึ่งพาระบบคลาวด์

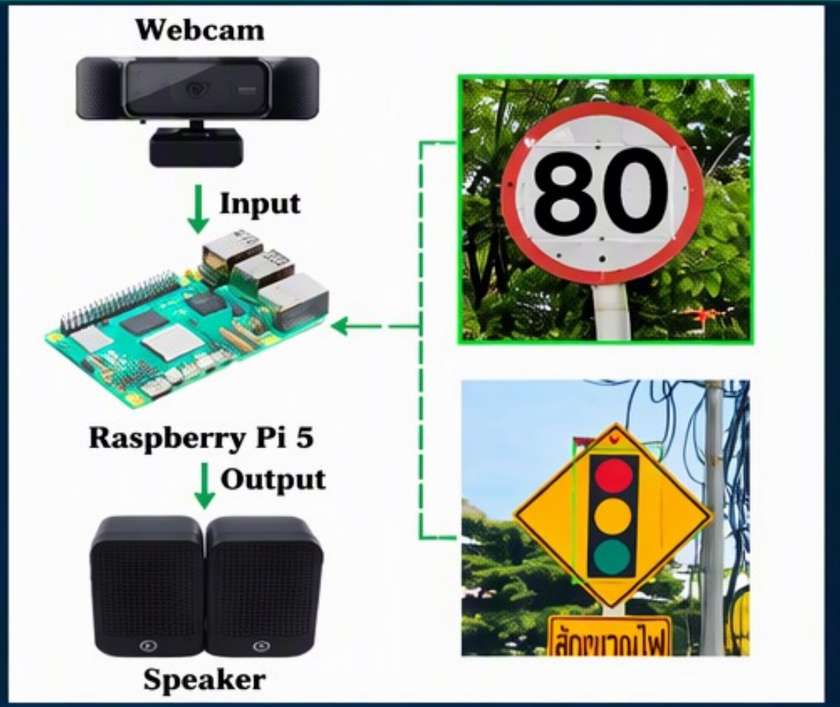
กายวิภาคของ Edge AI: MediaPipe vs. YOLOv11 (The Anatomy of Computer Vision)

MediaPipe (โครงสร้างจุดอ้างอิง - Landmarks)



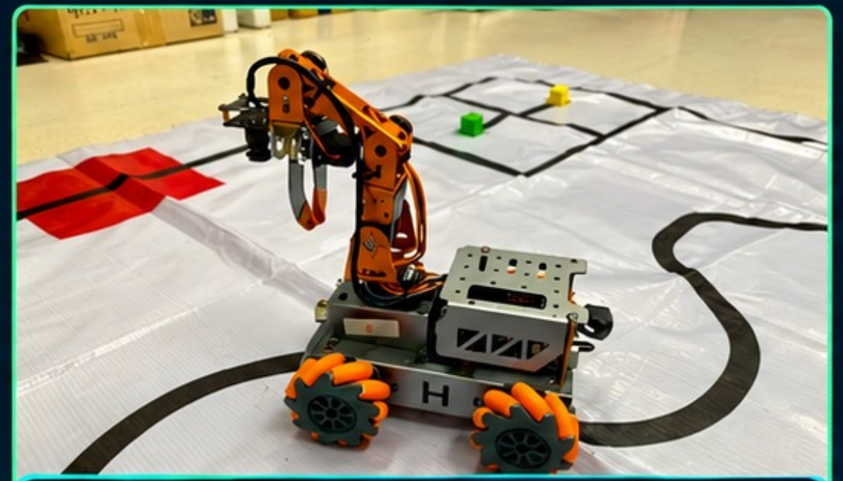
- **Method:** วิเคราะห์เวกเตอร์ด้วยความคล้ายคลึงโคไซน์ (Cosine Similarity)
- **Application:** ระบบตรวจสอบสิทธิ์แบบคู่ (Dual Authentication) ความแม่นยำ **95%**

YOLOv11 (กรอบล้อมรอบ - Bounding Boxes)

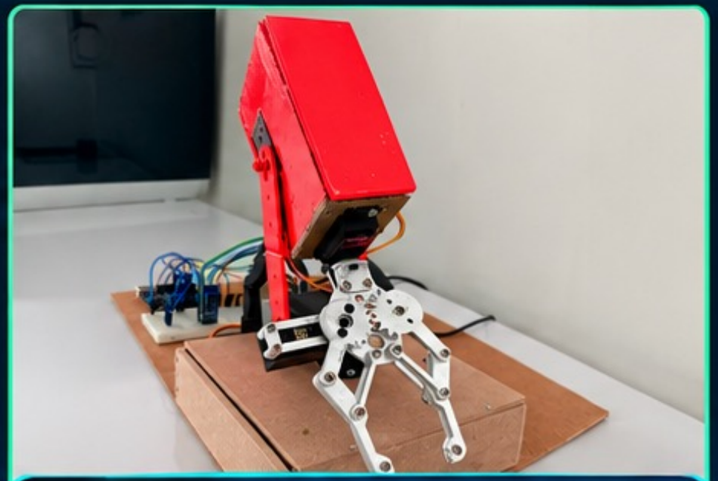


- **Method:** ตรวจจับวัตถุแบบองค์รวมในภาพเดียว (Single-shot Detection)
- **Application:** แยกแยะป้ายจราจรความเร็วสูงบน Raspberry Pi 5 เพื่อแจ้งเตือนผู้ขับขี่

Layer 4: ปัญญาประดิษฐ์เชิงกายภาพ (Physical AI & Real-World Interaction)



หุ่นยนต์เคลื่อนที่อิสระ (AMR)
ขับเคลื่อนด้วยล้อ Mecanum เคลื่อนที่ได้ทุกทิศทาง



แขนกลหุ่นยนต์ (Robotic Arms / Cobots)
5 แกนพร้อมทรีปปอร์ สำหรับหยิบจับหรือคัดแยก



อากาศยานไร้คนขับ (UAV / Drones)
ต้องการระบบควบคุมน้ำหนักเบาและตอบสนองฉับไว

ข้อควรระวังด้านความปลอดภัย เช่น

e-stop
หยุดการทำงานทันทีเมื่อเกิดอันตราย

การจำกัดแรง
ควบคุมแรงบิด/แรงดัน/แรงดึง ไม่ให้เกินค่าปลอดภัย

การทดสอบในพื้นที่ปลอดภัย
ก่อนใช้งานจริง เพื่อป้องกันอุบัติเหตุ และลดความเสี่ยง

Core Concept: Physical AI คือจุดตัดที่ โค้ด กลายเป็น แรงกระทำ (Kinetic Force)
การออกแบบต้องคำนึงถึงแรงเสียทาน แบตเตอรี่ และความปลอดภัย

วงจรป้อนกลับเซ็นเซอร์-มอเตอร์
(The Sensor-Motor Infinity Loop)

Data

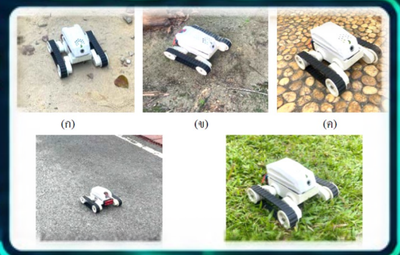


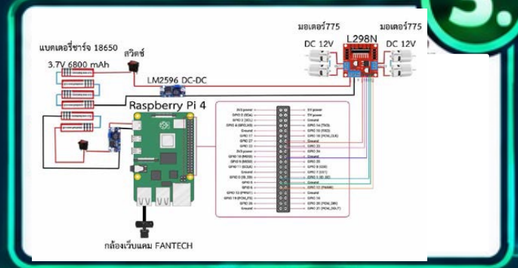
2. การรับรู้ (Sense)
เซ็นเซอร์และกล้องรับข้อมูล

Data

Data

1. สภาพแวดล้อม (Environment)
โลกกายภาพที่เปลี่ยนแปลงตลอดเวลา

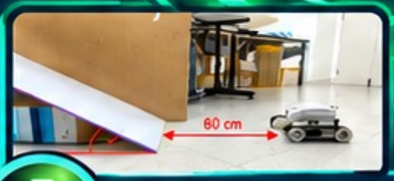




3. การประมวลผล (Think)
Edge AI & ROS2
วิเคราะห์และตัดสินใจ

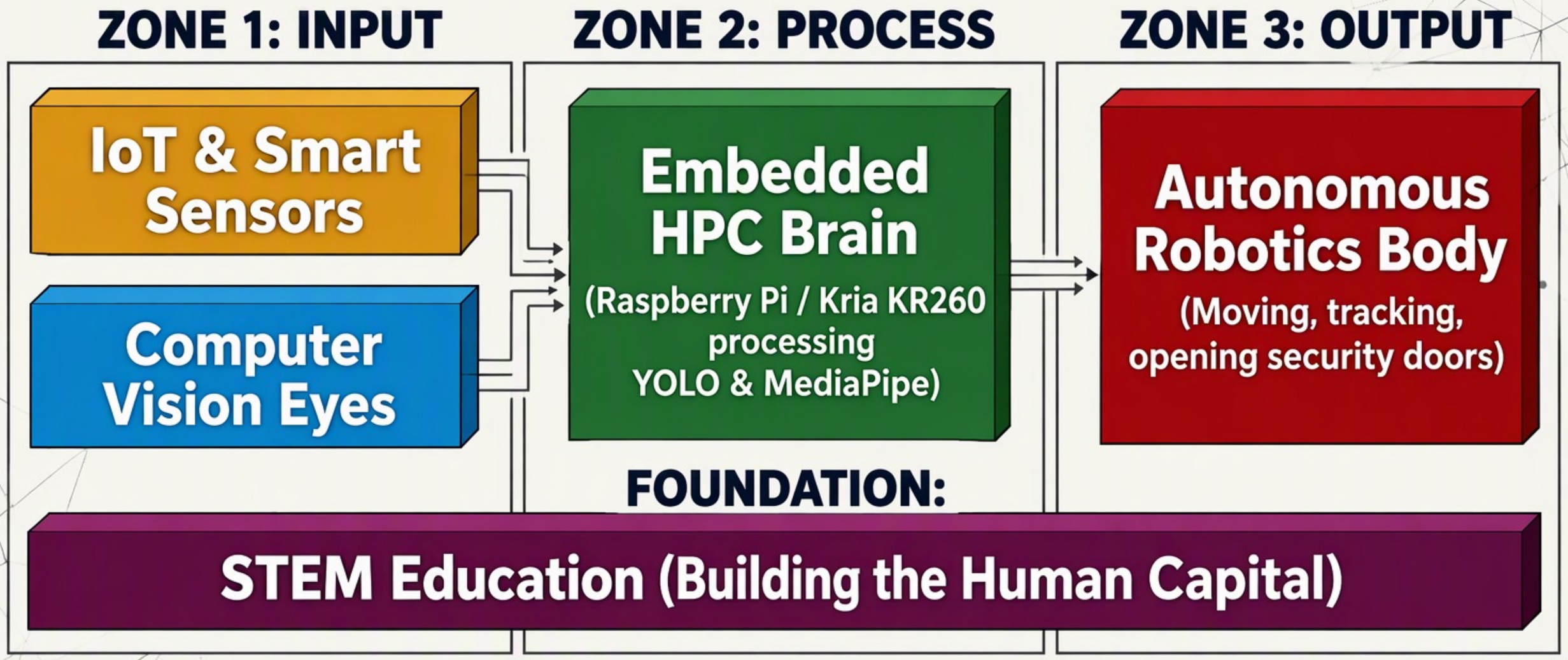
Data

Data



4. การตอบสนอง (Act)
มอเตอร์และเซอร์โวทำงาน

วิกฤตความหน่วง: วงจรนี้ต้องเกิดขึ้นในระดับมิลลิวินาที (Milliseconds)
หากเกิดความหน่วงในขั้นตอน Think หุ่นยนต์จะล้มเหลวในขั้นตอน Act


ZONE 1: INPUT
ZONE 2: PROCESS
ZONE 3: OUTPUT
IoT & Smart Sensors
Computer Vision Eyes
Embedded HPC Brain
(Raspberry Pi / Kria KR260 processing YOLO & MediaPipe)
Autonomous Robotics Body
(Moving, tracking, opening security doors)
FOUNDATION:
STEM Education (Building the Human Capital)

Embedded System

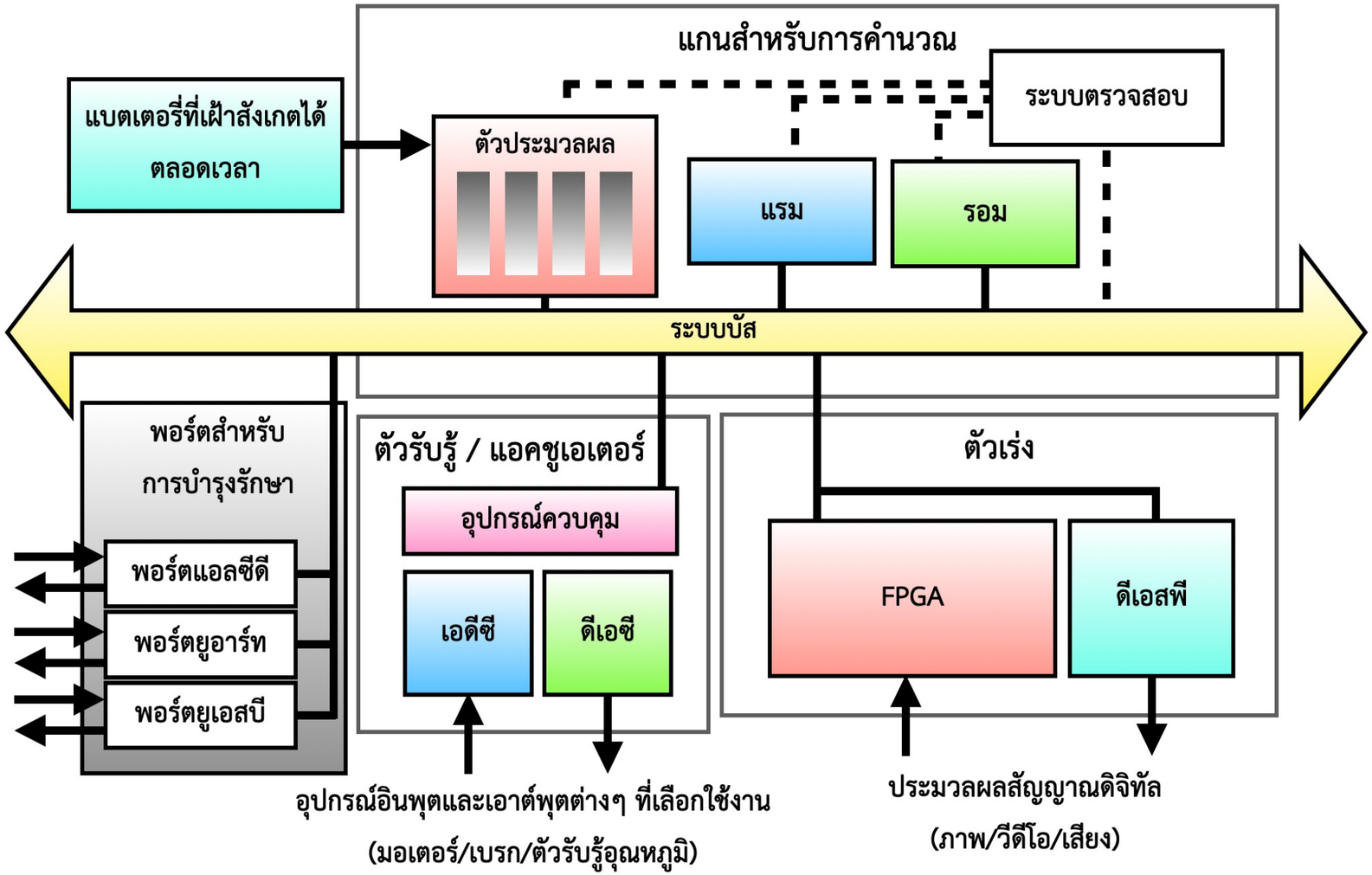
ระบบคอมพิวเตอร์ที่ฝังตัวอยู่ในอุปกรณ์ต่างๆ เพื่อทำงานเฉพาะด้าน

1. **ระบบฝังตัว** ตามศัพท์บัญญัติ สำนักงานราชบัณฑิตยสภา นำมาใช้แทนคำว่า **ระบบสมองกลฝังตัว**

2. **ระบบฝังตัว** จะมีคุณลักษณะที่แตกต่างกัน ขึ้นอยู่กับความหลากหลายของอุปกรณ์อินพุต และเอาต์พุต ที่เลือกใช้งานกับระบบฝังตัว

3. โดยทั่วไปเรียกว่า **ตัวรับรู้ (Sensor)** หรือตัว**ชับเคลื่อน (Actuator)** ซึ่งเป็นส่วนสำคัญที่มีความจำเป็นในการปรับแต่งระบบฝังตัวให้มีความเหมาะสมต่อการใช้งานสำหรับงานนั้นๆ

แผนภาพองค์ประกอบของระบบฝังตัว



ROS (Robot Operating System)

ระบบปฏิบัติการสำหรับหุ่นยนต์ที่ช่วยให้การพัฒนาเป็นเรื่องง่ายขึ้น

ROS



- **การสร้างหุ่นยนต์สำหรับทำภารกิจหรือแก้ไขปัญหาในชีวิตประจำวัน** จะต้องมี**โครงสร้างพื้นฐาน**
- **โครงสร้างพื้นฐาน** ได้แก่ **มิดเดิลแวร์ (Middloware)** ที่ประกอบด้วย **คลังโปรแกรม (Library)** สำหรับช่วยในการจัดการระบบหุ่นยนต์ และเครื่องมือช่วยในการพัฒนาหุ่นยนต์จำนวนมาก
- **ROS เป็นมิดเดิลแวร์สำหรับพัฒนาหุ่นยนต์** ที่ประสบความสำเร็จมากที่สุด และเป็นมาตรฐานสำหรับใช้เขียนโปรแกรมหุ่นยนต์อย่างแพร่หลาย
- **ภาษาสำหรับการเขียนโปรแกรมร่วมกับ ROS** ที่นิยม ได้แก่ **Python** และ **C/C++**

ตัวอย่าง ROS Distributions ยอดนิยม

Distro	Release date	Logo	EOL date
Lyrical Luth	May 22, 2026		May 2031
Kilted Kaiju	May 23, 2025		December 2026
Jazzy Jalisco	May 23, 2024		May 2029

รุ่น **Jazzy (LTS)** และ **Lyrical (LTS)** คือรุ่นที่ผู้เริ่มต้นควรใช้

ทำไมต้องใช้ ROS ?

ช่วยให้การพัฒนาหุ่นยนต์
ง่ายขึ้น มีประสิทธิภาพมากขึ้น

- สมมุติว่าหากต้องการสร้างหุ่นยนต์ ที่มีรูปแบบการทำงาน ดังนี้

1. มีกล้องเชื่อมต่ออยู่ที่หุ่นยนต์



2. หุ่นยนต์อ่านค่าข้อมูลที่ส่งมาจากกล้อง
แล้วนำไปทำการประมวลผลภาพ
(Image Processing)



3. ควบคุมหุ่นยนต์ระยะไกลผ่านคอมพิวเตอร์
แบบพกพา (Laptop) เพื่อดูวิดีโอ
ที่ส่งมาจากตัวหุ่นยนต์



เมื่อนำ ROS มาใช้งาน



/image_raw

QoS: Best Effort



/detections

QoS: Reliable



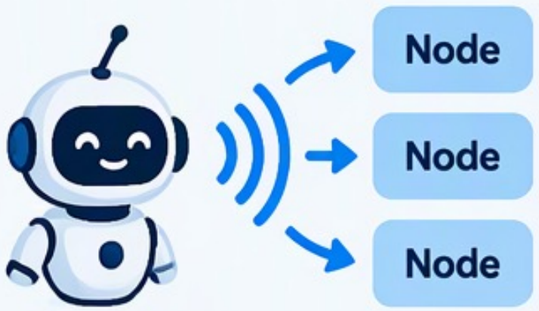
ลำดับการทำงาน ดังนี้

1. สั่ง ros2 run เปิด camera_node ได้ทันที ไม่ต้องเปิดเซิร์ฟเวอร์กลางก่อน
2. โหนดประกาศตัวเองผ่าน DDS discovery ทุกโหนดที่อยู่ ROS_DOMAIN_ID เดียวกัน และอยู่วงแลนเดียวกันจะเห็นกันเอง
3. camera_node เผยแพร่ (publish) ภาพไปที่หัวข้อ /image_raw ด้วย QoS แบบ Best Effort
4. image_proc_node และ rviz2 บอกรับ (subscribe) หัวข้อเดียวกันด้วย QoS ที่เข้ากันได้
5. หากเปลี่ยนกล้องรุ่นใหม่ ให้แก้ไขเฉพาะ camera_node ตราบใดที่ชื่อหัวข้อ ชนิดข้อความ และ QoS ยังเหมือนเดิม

รูปแบบการสื่อสารใน ROS 2 มี 4 แบบ

Topic

เผยแพร่–บอกรับ ทางเดียว ต่อเนื่อง หลายต่อหลาย



ตัวอย่าง

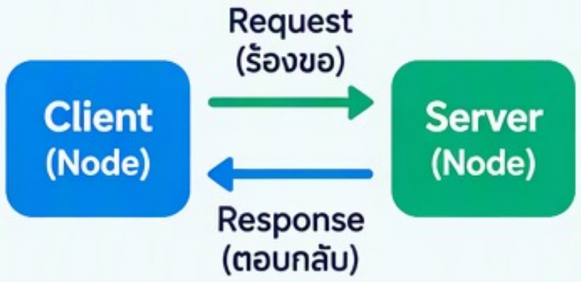
/scan, /image_raw, /cmd_vel

คำสั่งตัวอย่าง

```
ros2 topic echo /scan
```

Service

ร้องขอ–ตอบกลับ ใช้งาน งานสั้น ๆ ที่จบทันที



ตัวอย่าง

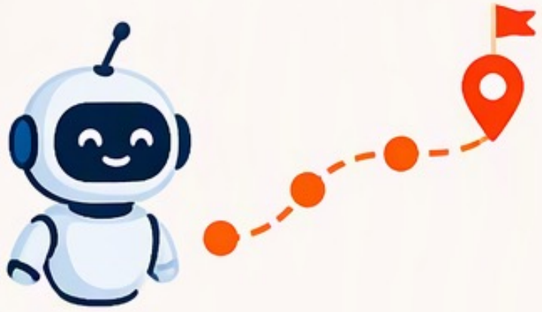
เปิด/ปิดมอเตอร์, อ่านค่าตั้งต้น

คำสั่งตัวอย่าง

```
ros2 service call /reset ...
```

Action

งานที่ใช้เวลานาน มีความคืบหน้า และยกเลิกได้



ตัวอย่าง

นำทางไปยังจุดหมาย, หยิบวัตถุ

คำสั่งตัวอย่าง

```
ros2 action send_goal ...
```

Parameter

ค่าปรับแต่งของโหนด แก้ไขได้ขณะทำงาน



ตัวอย่าง

ความเร็วสูงสุด, ค่าคาลิเบรต

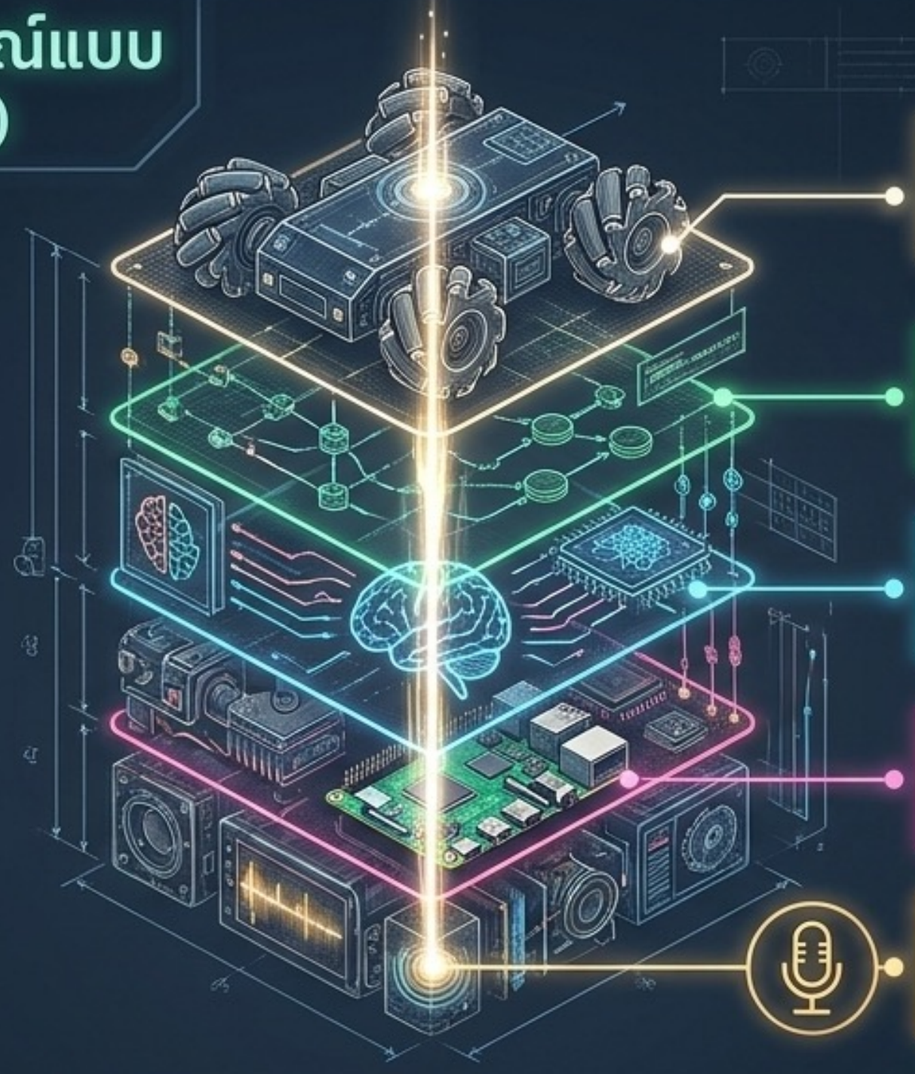
คำสั่งตัวอย่าง

```
ros2 param set /node p 1.0
```

เลือกใช้รูปแบบการสื่อสารที่เหมาะสม เพื่อพัฒนาโค้ด ROS 2 ได้อย่างมีประสิทธิภาพ

ROS 2

การบูรณาการระบบสมบูรณ์แบบ (The Grand Unification)



5. [Physical]
ล้อ Mecanum หมุนตอบสนองทันที
(Kinetic Actuation)

4. [ROS2]
ส่งข้อความผ่าน Topic แจ้งโหนดควบคุม
(Middleware Routing)

3. [Edge AI]
โมเดล AI แปลงเสียงเป็น Text Command
(Speech Recognition)

2. [Embedded]
ประมวลผลคลื่นเสียงบน Raspberry Pi
(Hardware Compute)

1. [Physical]
หูฟังรับเสียงพูด “เลี้ยวซ้าย”
(Human Voice Input)

ความเชี่ยวชาญในยุคหน้า ไม่ใช่การเก่งเทคโนโลยีใดเทคโนโลยีหนึ่ง แต่คือการเข้าใจ รอยต่อ (The Seams) ระหว่างระบบ

Gemini Notebook

แนวโน้มสำคัญของเทคโนโลยีหุ่นยนต์และ AI สำหรับอุปกรณ์ระบบฝังตัว
(Key Trends in Robotics & AI for Embedded Devices)

Humanoid Robots (หุ่นยนต์คล้ายมนุษย์)



- ความสามารถในการทรงตัวและการเคลื่อนไหวคล้ายมนุษย์ (Human-like Balance & Locomotion)
- การหยิบจับที่ซับซ้อนและการปฏิสัมพันธ์กับสภาพแวดล้อม (Complex Manipulation & Interaction)
- มีศักยภาพในการผลิต การบริการ และการดูแลสุขภาพ (Potential in Manufacturing, Service, and Healthcare)
- ตัวอย่าง: Unitree G1, Tesla Optimus, Agility Digit

การควบคุมหุ่นยนต์ด้วย LLM/VLM
(LLM/VLM in Robot Control)



- ทำความเข้าใจคำสั่งภาษาธรรมชาติและบริบท (Natural Language Command Understanding)
- การวางแผนงานและการแก้ปัญหาอย่างชาญฉลาด (Intelligent Task Planning & Problem Solving)
- การผสานข้อมูลหลายรูปแบบ ภาพและข้อความ (Multimodal Integration: Vision & Text)
- ตัวอย่าง: OpenAI GPT-6, Google Gemini, Anthropic Claude

NPU/AI Accelerator บนบอร์ดขนาดเล็ก
(Embedded NPU/AI Accelerators)



- เร่งความเร็วการประมวลผล AI และ NPU ในตัว (High-Speed AI & NPU Acceleration)
- ประสิทธิภาพพลังงานสูง เหมาะสำหรับอุปกรณ์ Edge (Low Power, High Efficiency for Edge Devices)
- ช่วยให้นำ AI ไปใช้ในหุ่นยนต์ขนาดเล็กและ IoT ได้จริง (Enablement of Small Robots & IoT AI)
- ตัวอย่าง: Raspberry Pi AI Kit (with Hailo), NVIDIA Jetson (Nano/Orin), Edge TPUs

หุ่นยนต์อัจฉริยะ AI ที่เข้าถึงได้ เพื่ออนาคตที่ดีกว่า

Smarter Robotics • Accessible AI • A Brighter Tomorrow

SANDO: Safe Autonomous Trajectory Planning for Dynamic Unknown Environments

Kota Kondo, Jesús Tordesillas, Juan Rached, Lili Sun, Yixuan Jia, and Jonathan P. How

3 [cs.RO] 25 Sep 2026

***Abstract*—This paper presents SANDO, a safe trajectory planner for 3D dynamic unknown environments. In such environments, the locations and motions of obstacles are not known in advance, and a collision-free plan can become unsafe at any moment as obstacles move unpredictably, requiring fast replanning. Existing soft-constraint planners are fast but do not guarantee collision-free paths, while hard-constraint methods typically ensure safety at the cost of longer computation times. SANDO addresses this trade-off through three main contributions. First, to avoid overly conservative or infeasible corridors near dynamic obstacles, a heat map-based A* global planner steers the path away from high-risk regions using soft costs, and a spatiotemporal safe flight corridor (STSFC) generator produces time-layered polytopes that inflate obstacles only by their worst-case reachable set at each time layer, rather than by the worst case over the entire horizon. Second, trajectory optimization is formulated as a mixed-integer quadratic program with hard collision-avoidance constraints, and a variable elimination technique reduces the number of decision variables, enabling fast computation. Third, a formal safety analysis establishes collision-free guarantees under explicit velocity-bound, size-bound, and estimation-error assumptions. Ablation studies confirm that variable elimination provides up to 7.4× speedup in optimization time, and that STSFCs are critical for maintaining feasibility in dense dynamic environments. Benchmark simulations against state-of-the-art**

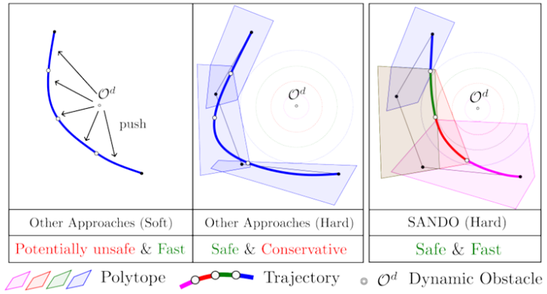


Fig. 1. Comparison of trajectory planning approaches near a dynamic obstacle $\mathcal{O}^d$. **Left (Other Approaches – Soft):** Soft-constraint methods use penalty-based collision avoidance, producing fast but potentially unsafe trajectories. **Center (Other Approaches – Hard):** Other hard-constraint methods inflate the obstacle by its full worst-case reachable set and generate purely spatial corridors, producing safe but overly conservative trajectories that detour far from the obstacle. **Right (SANDO):** SANDO generates spatiotemporal safe flight corridors (STSFCs) that account for when the trajectory passes each region, inflating the obstacle only by its reachable set at the corresponding time layer, yielding a safe and fast trajectory.

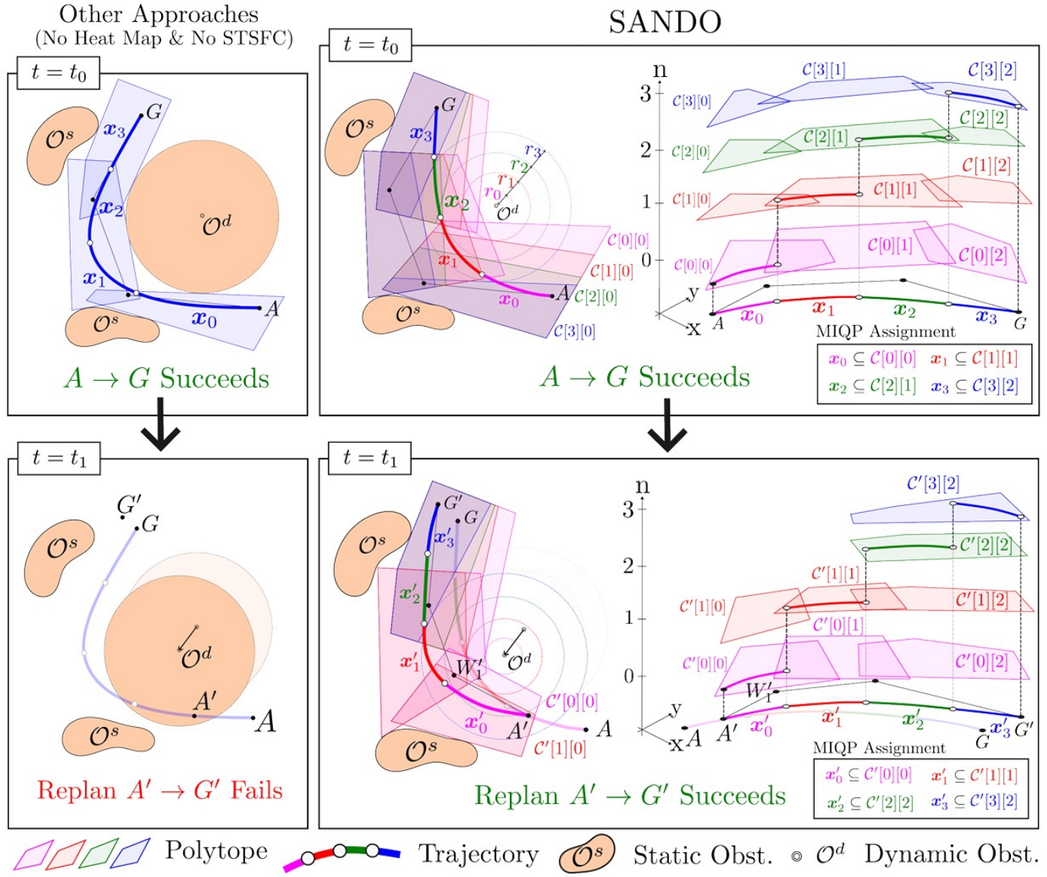
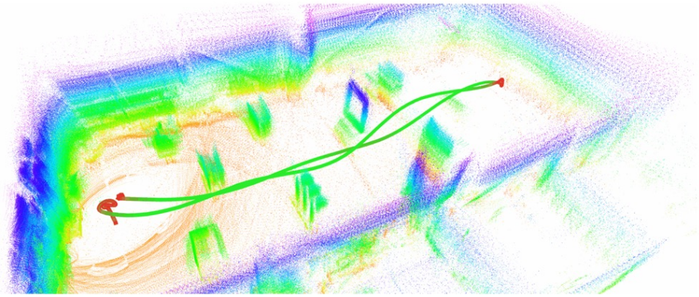




Fig. 10. Holybro PX4 Development Kit X500 is equipped with a Pixhawk flight controller and protective propeller guards. A Livox Mid-360 LiDAR is mounted on top for perception and localization.





K. Kondo, J. Tordesillas, J. Rached, L. Sun, Y. Jia, and J. P. How, "SANDO: Safe autonomous trajectory planning for dynamic unknown environments," arXiv:2604.07599 [cs.RO], 2026.

IDEAS
TECHNOLOGY
A BRIGHTER
TOMORROW

PEOPLE
SOLUTIONS
A BETTER
TOMORROW

Q&A

Thank you for listening



Together for a Smarter Tomorrow