MakerHub · MakerFlow

คู่มือครู: MakerFlow กับบอร์ด POP-32i

> เขียนโปรแกรมหุ่นยนต์ POP-32i (INEX) แบบลากวางบล็อกบนเว็บ **https://maker-hub.net/makerflow** — ไม่ต้องติดตั้ง Arduino IDE ไม่ต้องเขียนโค้ด ไม่ต้องคอมไพล์ แก้แล้วส่งลงหุ่นได้ในไม่ถึงวินาที
>
> ฉบับนี้ตรงกับ **เฟิร์มแวร์ 0.4.0-pop32** · บล็อกของบอร์ดนี้ **27 ชนิด** · สร้างจากรายการบล็อกชุดเดียวกับที่โปรแกรมใช้ — ตัวเลขและชื่อบล็อกในเล่มนี้จึงตรงกับหน้าจอเสมอ

สารบัญ

1. ภาพรวม — ทำงานยังไง
2. รู้จักบอร์ด
3. เปิดใช้ MakerFlow (ทุกคาบ)
4. ติดตั้งเฟิร์มแวร์ (ครั้งเดียวต่อบอร์ดหนึ่งใบ — ครูทำก่อนคาบ)
5. บทเรียนไล่ระดับ (ใช้สอนได้ทันที)
6. ข้อจำกัดของ POP-32i (สำคัญเวลาออกแบบใบงาน)
7. บล็อกทั้งหมดของ POP-32i (27 ชนิด)
8. ตัวอย่างสำเร็จรูปของ POP-32i (3 ชิ้น)
9. ข้อความเตือนที่เจอบ่อย (ข้อความจริงจากแผงตรวจกราฟ)
10. แก้ปัญหาหน้างาน

---

1. ภาพรวม — ทำงานยังไง

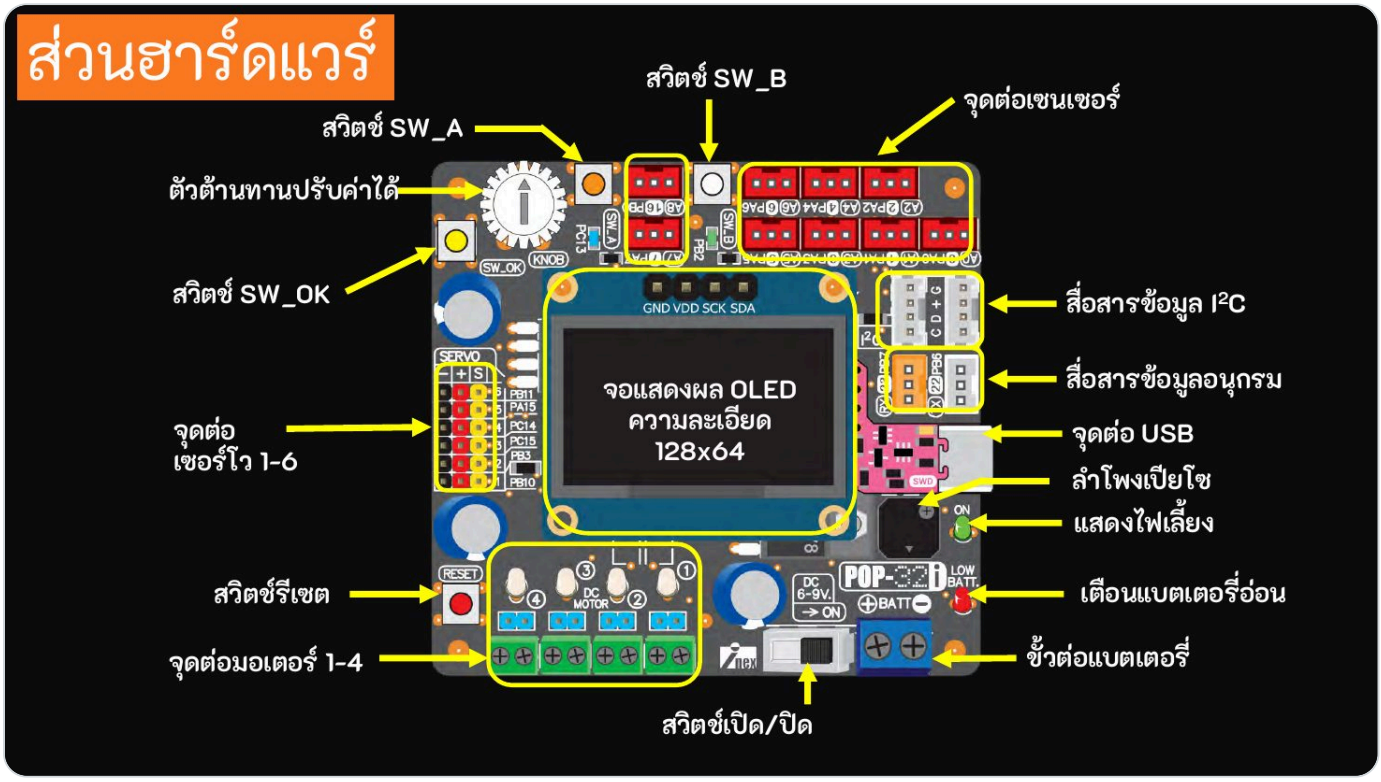
- **ติดตั้งเฟิร์มแวร์ MakerFlow ลงบอร์ดครั้งเดียว** (ครูทำก่อนคาบ ใช้เวลา ~1 นาทีต่อบอร์ด — หัวข้อ 4)
- จากนั้น "การเขียนโปรแกรม" คือการต่อบล็อกบนเว็บแล้วกด **ส่งลงบอร์ด** — กราฟถูกส่งผ่านสาย USB ไม่ใช้อินเทอร์เน็ตตอนส่ง และ **บันทึกลงบอร์ด** ⇒ ถอดสาย ปิด-เปิดใหม่ หุ่นก็ยังทำงานตามกราฟเดิม
- ระหว่างต่อสายอยู่ เปิด **ดูค่าสด** เพื่อเห็นค่าเซนเซอร์และค่าที่วิ่งบนเส้นแบบเรียลไทม์ — ใช้คาลิเบรตและหาจุดผิด

สิ่งที่ต้องมี

สิ่งที่ต้องมี	รายละเอียด
บอร์ด POP-32i + **แบตเตอรี่ 2 เซลล์ (7.4V) ที่ชาร์จแล้ว**	**สาย USB อย่างเดียวจ่ายไฟไม่พอ** — ไม่เปิดสวิตช์แบต คอมจะมองไม่เห็นบอร์ดเลย

สิ่งที่ต้องมี	รายละเอียด
สาย USB-C **แบบรับส่งข้อมูล**	สายชาร์จอย่างเดียวใช้ไม่ได้ (สาเหตุอันดับหนึ่งของ "ต่อไม่ติด")
คอมพิวเตอร์ + **Chrome หรือ Edge**	Windows · macOS · Linux · Chromebook
❌ ใช้ไม่ได้	**iPad · iPhone · Safari · Firefox** — ไม่มีช่องทางคุยกับบอร์ดผ่าน USB (Web Serial)
ครั้งแรกเท่านั้น	ไฟล์เฟิร์มแวร์ + โปรแกรม **STM32CubeProgrammer** (หัวข้อ 4)

2. รู้จักบอร์ด



ส่วนประกอบของบอร์ด POP-32i

MakerFlow เรียกอุปกรณ์ตาม **ชื่อช่องที่พิมพ์บนบอร์ด** ไม่ใช่เลขขาของชิป — นักเรียนเลือก "มอเตอร์ช่อง 2" ไม่ต้องรู้ว่าเป็นขาอะไร และต่อผิดขาไม่ได้

บนบอร์ด	มีกี่ช่อง	ใช้บล็อก
จุดต่อมอเตอร์ 1–4	4	⚙️ **มอเตอร์ (ช่อง 1–4)** · 🚗 **ขับหุ่น (สองล้อ)** · 🛞 **ขับหุ่น (ซ้าย/ขวาแยกกัน)**
จุดต่อเซอร์โว 1–6	6	🦾 **เซอร์โว (ช่อง 1–6)**

บนบอร์ด	มีกี่ช่อง	ใช้บล็อก
จุดต่อเซนเซอร์ A0–A8	9	**🎚 อ่านพอร์ต A (ค่าดิบ)** · **〰 เซนเซอร์เส้น (0–100)** · **🔲 อ่านพอร์ต A แบบดิจิทัล** · **💡 สั่งพอร์ต A เปิด/ปิด**
สวิตช์ SW_A / SW_B / SW_OK	3	**🔘 ปุ่มบนบอร์ด**
ตัวต้านทานปรับค่าได้ (ปุ่มหมุน)	1	**🎛 ปุ่มหมุนบนบอร์ด**
จอ OLED 128x64	1	**📟 แสดงบนจอบอร์ด**
ลำโพงเปียโซ	1	**🔔 เสียงบี๊บ**

⚠️ USB-C มีสองช่องซ้อนกัน — ใช้คนละงาน

ช่อง	ใช้ทำอะไร
ช่องบน (SWD)	**ติดตั้งเฟิร์มแวร์** ด้วย STM32CubeProgrammer — ใช้ครั้งเดียวตอนเตรียมบอร์ด
ช่องล่าง	**ใช้กับ MakerFlow ทุกวัน** — เชื่อมต่อบอร์ด · ส่งกราฟ · ดูค่าสด

เสียบผิดช่องคือปัญหาที่เจอบ่อยที่สุดในห้อง — แปะสติกเกอร์ "MakerFlow ↓ ช่องล่าง" ไว้บนบอร์ดช่วยได้มาก

ข้อควรรู้ของบอร์ดนี้

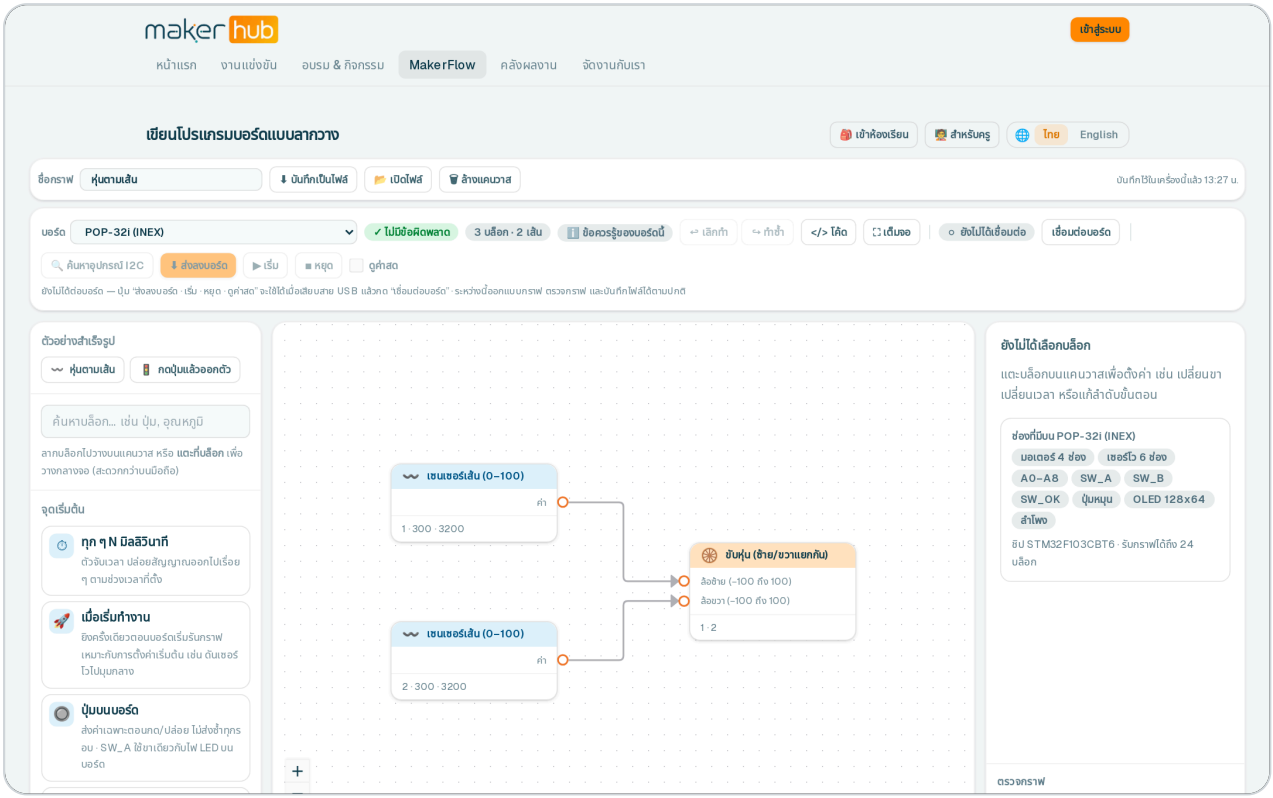
- ปุ่ม SW_A ใช้ขาเดียวกับไฟ LED บนบอร์ด (PC13)
- ปุ่มหมุนกับปุ่ม SW_OK ใช้ ADC ช่องเดียวกัน (PB1)
- ต้องต่อแบตเตอรี่ 2 เซลล์ (7.4V) แล้วเปิดสวิตช์ก่อน — สาย USB จ่ายไฟให้บอร์ดไม่พอ บอร์ดจะไม่โผล่ในคอมเลย
- ค่าที่วิ่งบนเส้นซ้อนกันได้ไม่เกิน 12 ชั้น (กราฟยาวเป็นสายเดี่ยวเกินกว่านั้นบอร์ดจะฟ้อง ไม่ค้าง)

---

3. เปิดใช้ MakerFlow (ทุกคาบ)

1. เปิด **https://maker-hub.net/makerflow?board=pop32i** ใน Chrome หรือ Edge — ลิงก์นี้ตั้งบอร์ดเป็น POP-32i ไว้ให้แล้ว (ถ้าเปิดเฉย ๆ ให้เลือกบอร์ด **POP-32i (INEX)** ที่ช่อง "บอร์ด")
2. ต่อแบต → **เปิดสวิตช์บอร์ด** → เสียบสาย USB-C ที่ **ช่องล่าง** เข้ากับคอม

3. กด **เชื่อมต่อบอร์ด** → หน้าต่างของเบราว์เซอร์ขึ้นรายการพอร์ต → เลือกพอร์ตของบอร์ด → กด "เชื่อมต่อ"
4. ต่อบล็อก (หรือกดตัวอย่างสำเร็จรูป) → ดูแผง **ตรวจกราฟ** ว่าไม่มีข้อความสีแดง → กด **ส่งลงบอร์ด**
5. ติ๊ก **ดูค่าสด** เพื่อดูค่าบนเส้น · กด **หยุด** เพื่อหยุดกราฟ



หน้าจอ MakerFlow ที่ตั้งบอร์ด POP-32i พร้อมตัวอย่าง “หุ่นตามเส้น”

> 💡 **ห้องเรียน MakerFlow** — ครูสร้างห้องที่ **https://maker-hub.net/classroom** ได้รหัสห้อง 6 ตัว + QR นักเรียนเข้า **https://maker-hub.net/class** ด้วยชื่อกับเลขที่ (ไม่ต้องมีอีเมล) งานบันทึกขึ้นระบบ และครูเห็นทั้งห้องว่าใครต่อบอร์ดแล้ว ใครส่งงานแล้ว — เลือกบอร์ด POP-32i ตอนสร้างห้องได้เลย

🛑 ความปลอดภัยของหุ่น (อ่านก่อนสอนเรื่องมอเตอร์)

- **ยกล้อให้ลอยก่อนกดส่งลงบอร์ดทุกครั้ง** — กราฟเริ่มทำงานทันทีที่ส่งเสร็จ หุ่นจะวิ่งตกโต๊ะได้
- **ถอดสาย USB แล้วหุ่นยังวิ่งต่อ** (ตั้งใจ — หุ่นแข่งต้องวิ่งเองได้) ⇒ จะหยุดให้กด **หยุด** บนเว็บ หรือ **ปิดสวิตช์บอร์ด**
- ส่งกราฟใหม่ที่ไม่มีบล็อกมอเตอร์ = มอเตอร์ทุกช่องหยุด
- ชาร์จแบตเต็มก่อนคาบ — แบตอ่อนแล้วไฟ **LOW BATT** สีแดงติด มอเตอร์จะแรงไม่เท่าเดิมและค่าเซนเซอร์เพี้ยน

4. ติดตั้งเฟิร์มแวร์ (ครั้งเดียวต่อบอร์ดหนึ่งใบ — ครูทำก่อนคาบ)

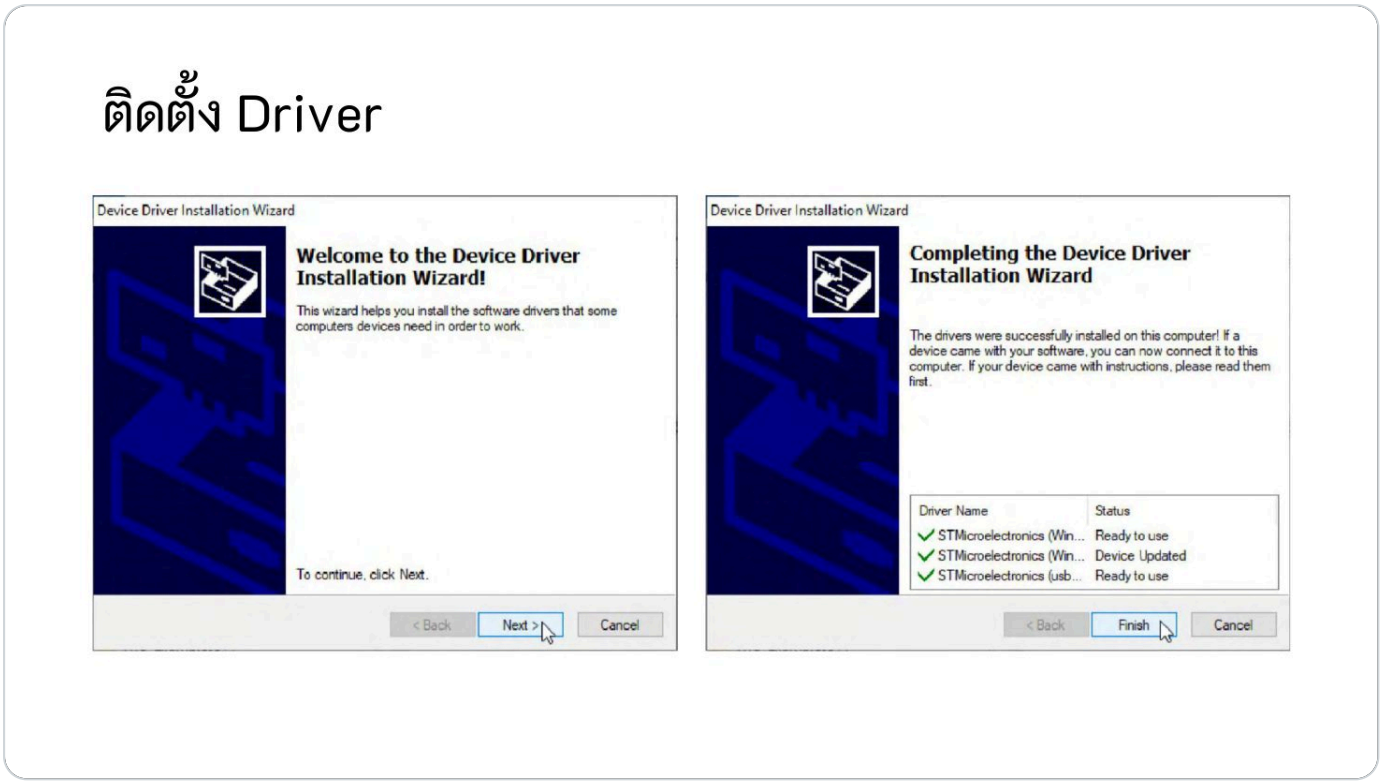
บอร์ดใหม่จากกล่องยังไม่มีเฟิร์มแวร์ MakerFlow ต้องติดตั้งก่อนหนึ่งครั้ง **POP-32i ติดตั้งจากเบราว์เซอร์ไม่ได้** (เป็นชิป STM32 คนละตระกูลกับ ESP32) ต้องใช้โปรแกรม STM32CubeProgrammer ผ่านช่อง USB-C **ด้านบน**

4.1 ไฟล์เฟิร์มแวร์

ดาวน์โหลด	**https://maker-hub.net/firmware/pop32i/makerflow-pop32i-0.4.0.bin**
รุ่น	0.4.0-pop32
ขนาด	123,356 ไบต์
แฟลชที่ที่อยู่	—
SHA-256	—

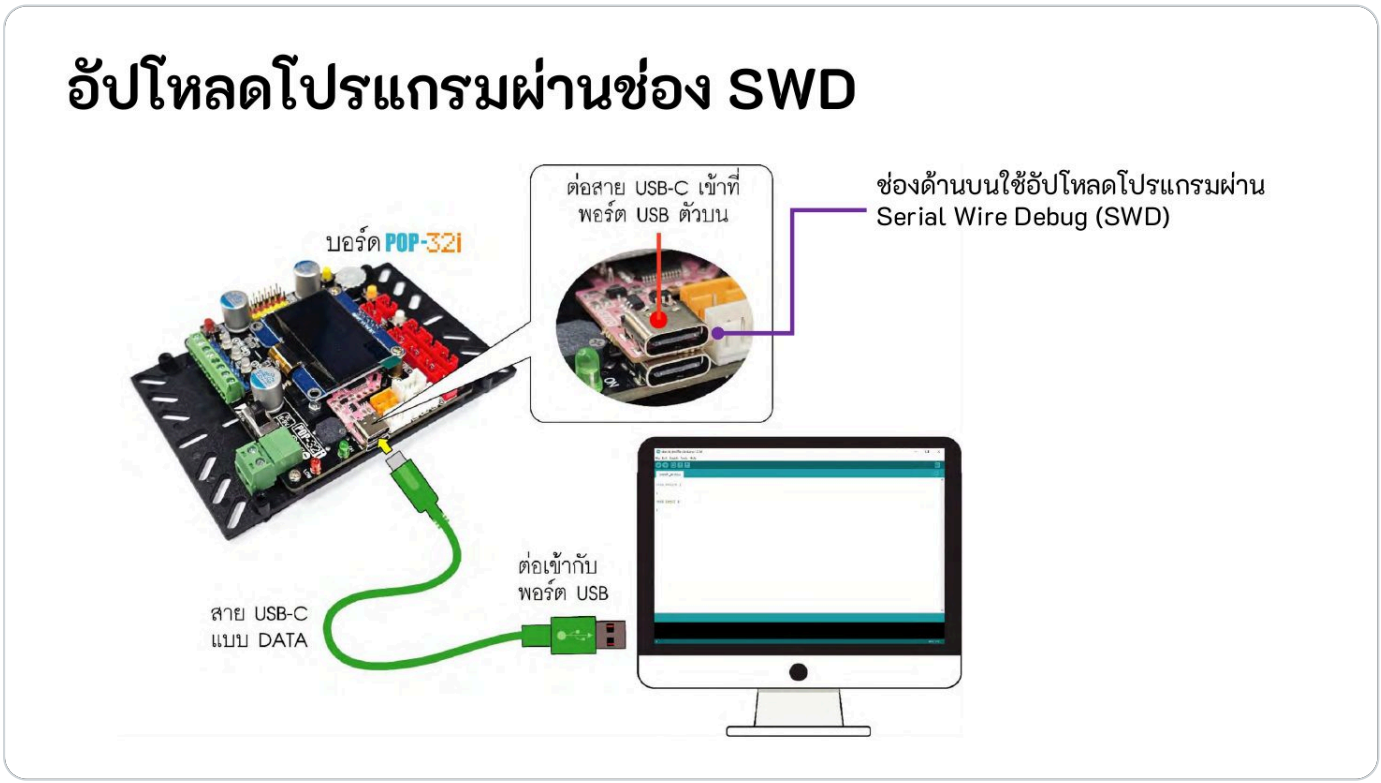
4.2 ติดตั้ง STM32CubeProgrammer (Windows — ครั้งเดียวต่อเครื่อง)

1. ดาวน์โหลดจาก **https://www.st.com/en/development-tools/stm32cubeprog.html** (ต้องสมัครบัญชี ST ฟรี — ครูที่สอนทั้งห้อง ให้สมัครครั้งเดียวแล้วเก็บตัวติดตั้งไว้ในแฟลชไดรฟ์)
2. แตกไฟล์ ZIP ก่อน แล้วดับเบิลคลิกตัวติดตั้ง · กดถัดไปตามค่าเริ่มต้น (ใช้พื้นที่ราว 640 MB)
3. ระหว่างติดตั้งจะมีหน้าต่าง **Device Driver Installation Wizard** แยกขึ้นมา — **ห้ามกดยกเลิก** กด Next แล้ว Finish ไดรเวอร์ของ STMicroelectronics ต้องขึ้นว่า *Ready to use* ทั้งสามตัว (ถ้าล้มเหลว ให้คลิกขวาตัวติดตั้ง → Run as administrator แล้วทำใหม่)



หน้าต่างติดตั้งไดรเวอร์ — ต้องขึ้น Ready to use ครบ

4.3 แฟลชเฟิร์มแวร์ลงบอร์ด



ช่อง USB-C ด้านบนคือช่อง SWD สำหรับติดตั้งเฟิร์มแวร์

1. ต่อแบต → **เปิดสวิตช์บอร์ด** → เสียบสาย USB-C ที่ **ช่องบน** เข้ากับคอม

2. เปิด **STM32CubeProgrammer** → มุมขวาบนเลือกการเชื่อมต่อเป็น **ST-LINK** → กด **Connect** (ถ้าเชื่อมต่อไม่ได้: ตรวจว่าเปิดสวิตช์บอร์ดแล้ว และเสียบช่องบนไม่ใช่ช่องล่าง)
3. ไปหน้า **Erasing & Programming** (ไอคอนรูปดาวน์โหลดทางซ้าย)
4. **File path** → Browse → เลือกไฟล์ `.bin` ที่ดาวน์โหลดมา
5. **Start address** ใส่ `0x08000000` · ติ๊ก **Verify programming** และ **Run after programming**
6. กด **Start Programming** → รอจนขึ้น *File download complete*
7. กด **Disconnect** → ย้ายสายไป **ช่องล่าง** → เปิด MakerFlow แล้วกด **เชื่อมต่อบอร์ด** ถ้าต่อติด แถบสถานะจะแสดงรุ่นเฟิร์มแวร์ 0.4.0-pop32 = ติดตั้งสำเร็จ

macOS / Linux (บรรทัดคำสั่ง) — ติดตั้ง `.bin` แล้วสั่ง `-w`

> ⚠️ **ติดตั้ง MakerFlow แล้ว โปรแกรม Arduino เดิมบนบอร์ดจะถูกทับ** และการอัปโหลดแบบ HID bootloader ผ่านช่องล่างของ Arduino IDE จะใช้ไม่ได้อีก · ถ้าจะกลับไปสอนเขียนโค้ดด้วย Arduino IDE ให้ตั้ง **Upload method = STM32CubeProgrammer (SWD)** แล้วอัปโหลดผ่านช่องบนตามปกติ — ใช้ได้ทุกเมื่อ และจะกลับมาใช้ MakerFlow ก็แค่แฟลชไฟล์นี้ซ้ำ
>
> เคยติดตั้งแล้ว **ไม่ต้องทำซ้ำ** ยกเว้นหน้าเว็บขึ้นว่า "ไม่รู้จักบล็อกชนิด ..." (แปลว่าเฟิร์มแวร์บนบอร์ดเก่ากว่าหน้าเว็บ ให้แฟลชไฟล์รุ่นล่าสุด)

---

5. บทเรียนไล่ระดับ (ใช้สอนได้ทันที)

แต่ละบทใช้เวลาประมาณ 10–20 นาที · ทุกบทเริ่มจากหน้าจอเปล่า (กด **ล้างแคนวาส**) · ลากบล็อกจากคลังด้านซ้าย แล้วลากเส้นจากจุด **ขาออก** (ขวาของบล็อก) ไปจุด **ขาเข้า** (ซ้ายของบล็อก)

บทที่ 1 — สวัสดีจอ (รู้จักต้นทาง → ปลายทาง)

1. วาง 🚀 **เมื่อเริ่มทำงาน** และ 🖥 **แสดงบนจอบอร์ด** แล้วลากเส้นเชื่อมกัน
2. คลิกบล็อกจอ → ช่อง **ข้อความนำหน้า** พิมพ์ `HELLO` · ติ๊ก **ล้างจอก่อนเขียน**
3. ส่งลงบอร์ด — จอขึ้น HELLO ตามด้วยค่าที่ส่งเข้ามา
4. **ประเด็นสอน:** ทุกกราฟต้องมี *ต้นทาง* ที่ปล่อยสัญญาณ ไม่งั้นไม่มีอะไรเกิดขึ้น · จอแสดงได้เฉพาะอักษรอังกฤษ ตัวเลข และสัญลักษณ์ (ภาษาไทยจะเป็น ?)

บทที่ 2 — กดปุ่มแล้วมีเสียง

1. วาง 🔘 **ปุ่มบนบอร์ด** (เลือกปุ่ม **SW_B**) → ต่อเข้า 🔔 **เสียงบี๊บ**

2. ส่งลงบอร์ดแล้วกด SW_B · ลองเปลี่ยน **ความถี่** เป็น 1000 แล้วส่งใหม่
3. **ประเด็นสอน:** ปุ่มส่งค่า จริง ตอนกด และ *เท็จ* ตอนปล่อย (ส่งเฉพาะตอนเปลี่ยน ไม่ส่งซ้ำ) · เสียงดับเมื่อได้รับ *เท็จ*
4. ⚠️ SW_A ใช้ขาเดียวกับไฟ LED บนบอร์ด — บทแรก ๆ ใช้ SW_B จะไม่สับสน

บทที่ 3 — อ่านปุ่มหมุนขึ้นจอ (ดูค่าสด)

1. วาง 🎛️ **ปุ่มหมุนบนบอร์ด** → ต่อเข้า 🖥️ **แสดงบนจอบอร์ด** (ข้อความนำหน้า _)
2. ส่งลงบอร์ด · ติ๊ก **ดูค่าสด** แล้วหมุนปุ่ม — ตัวเลขบนเส้นและบนจอเปลี่ยนตาม (0–4000)
3. ลองตั้ง **ปรับช่วงเป็น 0 ถึง** เป็น _ — ค่ากลายเป็น 0–100
4. **ประเด็นสอน:** "ค่าดิบ" กับ "ค่าที่แปลงแล้ว" · ช่อง *ไม่ส่งถ้าเปลี่ยนน้อยกว่า* ช่วยตัดสัญญาณรบกวน

บทที่ 4 — คุมความเร็วมอเตอร์ด้วยปุ่มหมุน

1. 🎛️ **ปุ่มหมุนบนบอร์ด** → 📐 **แปลงช่วงค่า** → ⚙️ **มอเตอร์ (ช่อง 1–4)** (ช่อง 1)
2. ตั้งบล็อกแปลงช่วงค่า: จาก 0–4000 เป็น -100 ถึง 100
3. **ยกล้อให้ลอย** แล้วส่งลงบอร์ด · หมุนปุ่ม: ตรงกลาง = หยุด · หมุนไปทางหนึ่ง = เดินหน้า · อีกทาง = ถอยหลัง
4. **ประเด็นสอน:** มอเตอร์รับ -100 ถึง 100 (เครื่องหมายคือทิศ ขนาดคือแรง) · ถ้าล้อหมุนกลับทาง ติ๊ก *กลับทางหมุน* แทนการสลับสาย

บทที่ 5 — กดปุ่มแล้วออกตัว (ตัวอย่างสำเร็จรูป)

1. คลังบล็อก → **ตัวอย่างสำเร็จรูป** → **กดปุ่มแล้วออกตัว**
2. ตรวจช่อง **ช่องล้อซ้าย / ช่องล้อขวา** ในบล็อก 🚗 **ขับหุ่น (สองล้อ)** ให้ตรงกับหุ่นจริง (หุ่นแต่ละตัวต่อไม่เหมือนกัน)
3. ลองเปลี่ยน **ท่า** เป็นหมุนซ้าย/เลี้ยวขวา แล้วสังเกตความต่างระหว่าง "หมุนอยู่กับที่" กับ "เลี้ยวล้อเดียว"

บทที่ 6 — คาลิเบรตเซนเซอร์เส้น แล้วตามเส้น

1. วาง 🎚️ **อ่านพอร์ต A (ค่าดิบ)** (พอร์ต A1) ส่งลงบอร์ด ติ๊ก **ดูค่าสด**
2. วางเซนเซอร์บน **พื้นดำ** จดค่า · วางบน **พื้นขาว** จดค่า · ทำซ้ำกับ A2
3. เปิด **ตัวอย่างสำเร็จรูป** → **หุ่นตามเส้น** → ใส่ค่าที่จดลงช่อง **ค่าที่วัดได้บนพื้นดำ** / **ค่าที่วัดได้บนพื้นขาว** ของบล็อกเซนเซอร์เส้นทั้งสองตัว
4. ส่งลงบอร์ด วางหุ่นคร่อมเส้น

5. **ประเด็นสอน:** ค่าคาลิเบรตเป็นของ *หุ่นตัวนั้นบนสนามนั้น* ใช้ข้ามตัวไม่ได้ · ถ้าหุ่นวิ่งหนีเส้น ติ๊ก *กลับค่า* หรือสลับเส้นที่ต่อเข้าล้อซ้าย/ขวา — นี่คือจุดเริ่มของการคุมแบบสัดส่วน (proportional control)

บทที่ 7 — ท่าต่อเนื่องด้วยลำดับขั้นตอน

📋 **ลำดับขั้นตอน** ทำงานทีละขั้นจากบนลงล่าง (เช่น เดินหน้า → รอ 1 วินาที → หมุน → รอ → หยุด) โดย "รอ" ไม่หยุดกราฟส่วนอื่น เหมาะกับท่าออกตัวและภารกิจที่เป็นขั้นตอน · บอร์ดนี้ใส่ได้ไม่เกิน 24 ขั้นต่อบล็อก

---

6. ข้อจำกัดของ POP-32i (สำคัญเวลาออกแบบใบงาน)

เพดาน	ค่า
จำนวนบล็อกต่อกราฟ	**24**
จำนวนเส้นต่อกราฟ	**32**
ขนาดกราฟที่ส่งได้ต่อครั้ง	**2,000 ไบต์**

- **ตัวที่เต็มก่อนมักเป็นขนาดกราฟ ไม่ใช่จำนวนบล็อก** — ข้อความภาษาไทยในช่องข้อความใช้ 3 ไบต์ต่อตัวอักษร ใบงานที่ใช้เกิน ~18 บล็อกให้ลองส่งจริงก่อนแจก
- บอร์ดนี้ **ไม่มี** บล็อกขาดิจิทัลอิสระ · PWM · NeoPixel · เซนเซอร์ I2C · RS485 ของบอร์ด ESP32 (ไม่มีขาว่างให้ต่อเอง และหน่วยความจำไม่พอ) — ใช้ช่องที่มีบนบอร์ดแทน
- เปิดไฟล์กราฟที่ทำไว้กับบอร์ด ESP32 จะมีบล็อกที่ใช้ไม่ได้ แผงตรวจกราฟจะบอกว่าตัวไหน

---

7. บล็อกทั้งหมดของ POP-32i (27 ชนิด)

สัญลักษณ์: **ขาเข้า** = จุดด้านซ้ายของบล็อก · **ขาออก** = จุดด้านขวา

ต้นทาง — จุดเริ่มของสัญญาณ (7)

⏱ ทุก ๆ N มิลลิวินาที

ตัวจับเวลา ปล่อยสัญญาณออกไปเรื่อย ๆ ตามช่วงเวลาที่ตั้ง

ขา	รับ/ส่งค่า
ขาออก: ครั้งที่	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ทุก ๆ	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 3600000 · มิลลิวินาที	

🚀 เมื่อเริ่มทำงาน

ยิงครั้งเดียวตอนบอร์ดเริ่มรันกราฟ เหมาะกับการตั้งค่าเริ่มต้น เช่น ดันเซอร์โวไปมุมกลาง

ขา	รับ/ส่งค่า
ขาออก: เริ่ม	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
หน่วงก่อนยิง	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 3600000 · มิลลิวินาที	0 = ยิงทันที · ใส่ค่าไว้เมื่ออุปกรณ์ต้องการเวลาตั้งตัวก่อน

🔘 ปุ่มบนบอร์ด

ส่งค่าเฉพาะตอนกด/ปล่อย ไม่ส่งซ้ำทุกรอบ · SW_A ใช้ขาเดียวกับไฟ LED บนบอร์ด

ขา	รับ/ส่งค่า
ขาออก: สถานะ	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ปุ่ม	ช่องบนบอร์ด	_	—	
กันเด้ง	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 500 · มิลลิวินาที	
อ่านทุก	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 1000 · มิลลิวินาที	

🎛️ ปุ่มหมุนบนบอร์ด

ใช้ตั้งค่าหน้าสนามโดยไม่ต้องแก้โปรแกรม เช่น ปรับความเร็วหุ่นก่อนออกตัว

ขา	รับ/ส่งค่า
ขาออก: ค่า	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
อ่านทุก	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 3600000 · มิลลิวินาที	
ไม่ส่งถ้าเปลี่ยนน้อยกว่า	ตัวเลข	_	ต่ำสุด 0 · หน่วย	
ปรับช่วงเป็น 0 ถึง	ตัวเลข	_	ต่ำสุด 0	0 = คืนค่าดิบ 0–4000

🎚 อ่านพอร์ต A (ค่าดิบ)

คืนค่า 0–4095 ตามที่วัดได้จริง เหมาะกับการดูค่าดิบตอนคาลิเบรต

ขา	รับ/ส่งค่า
ขาออก: ค่า	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
พอร์ต	ช่องบนบอร์ด	_	—	
อ่านทุก	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 3600000 · มิลลิวินาที	
ไม่ส่งถ้าเปลี่ยนน้อยกว่า	ตัวเลข	_	ต่ำสุด 0 · หน่วย	

〰 เซนเซอร์เส้น (0–100)

แปลงค่าดิบเป็น 0–100 ด้วยค่าที่คาลิเบรตไว้ · ค่าคาลิเบรตเป็นของหุ่นแต่ละตัวและพื้นสนามแต่ละที่ ใช้ข้ามตัวไม่ได้

ขา	รับ/ส่งค่า
ขาออก: ค่า	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
พอร์ต	ช่องบนบอร์ด	_	—	
ค่าที่วัดได้บนพื้นดำ	ตัวเลข	_	—	
ค่าที่วัดได้บนพื้นขาว	ตัวเลข	_	—	
กลับค่า	ติ๊กถูก	ไม่ติ๊ก	—	
อ่านทุก	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 3600000 · มิลลิวินาที	

🔲 อ่านพอร์ต A แบบดิจิทัล

ใช้กับสวิตช์ชนหรือเซนเซอร์ที่ให้ผลเป็นเปิด/ปิด เสียบที่พอร์ต A ช่องไหนก็ได้

ขา	รับ/ส่งค่า
ขาออก: สถานะ	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
พอร์ต	ช่องบนบอร์ด	_	—	
กดแล้วเป็นจริง	ติ๊กถูก	ติ๊ก	—	บอร์ดดึงไฟขึ้นไว้ ⇒ ตอนกดจะอ่านได้ 0
กันเด้ง	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 500 · มิลลิวินาที	
อ่านทุก	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 1000 · มิลลิวินาที	

ตรรกะ / แปลงค่า (11)

📌 แทนที่ด้วยค่าคงที่

ได้รับอะไรก็ตาม ส่งออกเป็นค่าที่ตั้งไว้เสมอ · ใช้ป้อนสีให้ไฟ NeoPixel (#rrggbb) หรือข้อความขึ้นจอได้

ขา	รับ/ส่งค่า
ขาเข้า: เข้า	ค่าใดก็ได้
ขาออก: ออก	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ค่า	ค่าใดก็ได้	_	—	เลือกชนิดให้ตรงกับปลายทาง · ตัวเลข 0 = ปิด/เท็จ แต่ข้อความ “0” บอร์ดนับเป็น **เปิด/จริง**

🔀 สลับเปิด / ปิด

ทุกครั้งที่ได้รับสัญญาณจะสลับค่าไปมาระหว่างเปิดกับปิด

ขา	รับ/ส่งค่า
ขาเข้า: กระตุ้น	ค่าใดก็ได้

ขา	รับ/ส่งค่า
ขาออก: สถานะ	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
เริ่มต้นเป็นเปิด	ติ๊กถูก	ไม่ติ๊ก	—	

⚖️ เปรียบเทียบ

เช่น อุณหภูมิ > 30 แล้วส่งค่าจริงไปเปิดพัดลม

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ตัวเลข
ขาออก: จริง/เท็จ	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
เงื่อนไข	เลือกจากรายการ	_	มากกว่า > · มากกว่าหรือเท่ากับ ≥ · น้อยกว่า < · น้อยกว่าหรือเท่ากับ ≤ · เท่ากับ = · ไม่เท่ากับ ≠	
เทียบกับ	ตัวเลข	_	—	

📐 แปลงช่วงค่า

เช่น แปลงค่าดิบ 0–4095 จากโพเทนชิโอมิเตอร์ ให้เป็น 0–100 เปอร์เซ็นต์

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ตัวเลข
ขาออก: ค่าใหม่	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
จากช่วง ต่ำสุด	ตัวเลข	_	—	
จากช่วง สูงสุด	ตัวเลข	_	—	
เป็นช่วง ต่ำสุด	ตัวเลข	_	—	
เป็นช่วง สูงสุด	ตัวเลข	_	—	

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ตัดค่าไม่ให้เกินช่วง	ติ๊กถูก	ติ๊ก	—	

🧮 คำนวณ

คำนวณกับค่าที่วิ่งเข้ามา เช่น หารด้วย 10 เพื่อเปลี่ยนหน่วย

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ตัวเลข
ขาออก: ผลลัพธ์	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
การกระทำ	เลือกจากรายการ	_	บวก + · ลบ – · คูณ × · หาร ÷ · เศษเหลือ mod · เลือกค่าน้อยกว่า · เลือกค่ามากกว่า · ค่าสัมบูรณ์ (ไม่ใช้ตัวเลขข้างล่าง) · ปัดเศษ (ตัวเลขข้างล่าง = จำนวนทศนิยม · 0 = จำนวนเต็ม)	
กับตัวเลข	ตัวเลข	_	—	

➗ คำนวณสองค่า

คิดเลขระหว่างสองขาเข้า · แต่ละขาจำค่าล่าสุดไว้ (เริ่มที่ 0) และคำนวณใหม่ทุกครั้งที่ขาใดขาหนึ่งได้ค่าใหม่ · ใช้ผสมความเร็วล้อ (ความเร็ว + เลี้ยว) หรือต่อ “ตาแมลง” หนึ่งคู่ด้วยมือ: ตา A → หน่วงเวลา → คูณกับตา B

ขา	รับ/ส่งค่า
ขาเข้า: A	ตัวเลข
ขาเข้า: B	ตัวเลข
ขาออก: ผลลัพธ์	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
การกระทำ	เลือกจากรายการ	_	A + B · A – B · A × B · เลือกค่าที่น้อยกว่า · เลือกค่าที่มากกว่า	

🔗 ตรรกะ และ/หรือ

จำค่าล่าสุดของแต่ละขาไว้ แล้วคำนวณใหม่ทุกครั้งที่มีค่าเข้ามา

ขา	รับ/ส่งค่า
ขาเข้า: A	จริง/เท็จ
ขาเข้า: B	จริง/เท็จ
ขาออก: ผล	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
การกระทำ	เลือกจากรายการ	—	และ (ทั้งคู่ต้องจริง) · หรือ (อย่างน้อยหนึ่ง) · ต่างกันเท่านั้น · กลับค่า (ใช้เฉพาะขา A)	

🔢 ตัวนับ

นับจำนวนครั้ง เช่น นับคนเดินผ่านเซนเซอร์

ขา	รับ/ส่งค่า
ขาเข้า: นับ	ค่าใดก็ได้
ขาเข้า: รีเซ็ต	ค่าใดก็ได้
ขาออก: จำนวน	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
เพิ่มครั้งละ	ตัวเลข	—	—	
วนกลับเป็น 0 เมื่อถึง	ตัวเลข	—	ต่ำสุด 0	ใส่ 0 = ไม่วนกลับ

⏳ หน่วงเวลา

ส่งค่าเดิมต่อไปหลังผ่านไปตามเวลาที่ตั้ง — ไม่หยุดการทำงานของกราฟส่วนอื่น

ขา	รับ/ส่งค่า
ขาเข้า: เข้า	ค่าใดก็ได้
ขาออก: ออก (ช้ากว่า)	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
หน่วง	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 3600000 · มิลลิวินาที	

🚪 ประตูเปิด / ปิด

ปล่อยค่าผ่านเฉพาะตอนที่ขา “อนุญาต” เป็นจริง เช่น ให้ทำงานเฉพาะตอนเปิดสวิตช์

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ค่าใดก็ได้
ขาเข้า: อนุญาต	จริง/เท็จ
ขาออก: ผ่าน	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
เริ่มต้นเปิดอยู่	ติ๊กถูก	ไม่ติ๊ก	—	ปิดไว้ = ค่าจะยังไม่ผ่านจนกว่าขา “อนุญาต” จะเป็นจริง

📋 ลำดับขั้นตอน

รวมหลายขั้นตอนไว้ใน node เดียว ทำให้กราฟไม่รกด้วย node จิ๋ว ๆ

ขา	รับ/ส่งค่า
ขาเข้า: เริ่ม	ค่าใดก็ได้
ขาออก: จบแล้ว	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ขั้นตอน	ลำดับขั้นตอน	(1 รายการ)	—	ทำจากบนลงล่างทีละขั้น — “รอ” ไม่หยุดกราฟส่วนอื่น
ทำวนซ้ำไม่รู้จบ	ติ๊กถูก	ไม่ติ๊ก	—	ถ้าเปิด ต้องมีขั้นตอน “รอ” อย่างน้อยหนึ่งขั้น ไม่งั้นลำดับจะวนกินซีพียูจนกราฟส่วนอื่นไม่เดิน

ปลายทาง — สั่งอุปกรณ์ (8)

🖥 แสดงค่าบนหน้าจอ

ส่งค่าขึ้นมาแสดงในหน้าต่างผลลัพธ์ของเว็บ ใช้ตรวจว่ากราฟทำงานถูกไหม

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ชื่อกำกับ	ข้อความ	_	—	

⚙️ มอเตอร์ (ช่อง 1–4)

ค่าบวก = หมุนไปข้างหน้า · ค่าลบ = หมุนกลับทาง · 0 = หยุด

ขา	รับ/ส่งค่า
ขาเข้า: กำลัง (–100 ถึง 100)	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ช่องมอเตอร์	ช่องบนบอร์ด	_	—	
กลับทางหมุน	ติ๊กถูก	ไม่ติ๊ก	—	

🚗 ขับหุ่น (สองล้อ)

รับ true/false = เดิน/หยุด ตามความเร็วที่ตั้งไว้ · รับตัวเลข = ใช้เป็นความเร็วแทน (0 = หยุด) · ตั้งช่องล้อได้ เพราะหุ่นแต่ละตัวต่อไม่เหมือนกัน

ขา	รับ/ส่งค่า
ขาเข้า: เดิน/หยุด หรือความเร็ว	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ท่า	เลือกจากรายการ	_	เดินหน้า · ถอยหลัง · หมุนซ้ายอยู่กับที่ · หมุนขวาอยู่กับที่ · เลี้ยวซ้าย (ล้อเดียว) · เลี้ยวขวา (ล้อเดียว) · หยุด	
ความเร็ว	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 100	
ช่องล้อซ้าย	ช่องบนบอร์ด	_	—	
ช่องล้อขวา	ช่องบนบอร์ด	_	—	

🛞 ขับหุ่น (ซ้าย/ขวาแยกกัน)

หัวใจของการตามเส้น: ป้อนค่าจากเซนเซอร์ซ้ายเข้าล้อซ้าย เซนเซอร์ขวาเข้าล้อขวา หุ่นจะเลี้ยวเข้าหาเส้นเอง

ขา	รับ/ส่งค่า
ขาเข้า: ล้อซ้าย (−100 ถึง 100)	ตัวเลข
ขาเข้า: ล้อขวา (−100 ถึง 100)	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ช่องล้อซ้าย	ช่องบนบอร์ด	_	—	
ช่องล้อขวา	ช่องบนบอร์ด	_	—	

🦾 เซอร์โว (ช่อง 1–6)

ส่ง −1 เพื่อปล่อยแกน (หยุดออกแรงค้าง) · จบกราฟแล้วบล็อกนี้ปล่อยแกนให้เอง

ขา	รับ/ส่งค่า
ขาเข้า: องศา 0–180	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ช่องเซอร์โว	ช่องบนบอร์ด	_	—	
องศาต่ำสุด	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 180	
องศาสูงสุด	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 180	

💡 สั่งพอร์ต A เปิด/ปิด

ใช้กับไฟหรืออุปกรณ์เปิด/ปิดที่ต่อพอร์ต A

ขา	รับ/ส่งค่า
ขาเข้า: เปิด/ปิด	จริง/เท็จ

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
พอร์ต	ช่องบนบอร์ด	_	—	

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
กลับค่า	ติ๊กถูก	ไม่ติ๊ก	—	

🖥 แสดงบนจอบอร์ด

จอ 8 บรรทัด บรรทัดละ 21 ตัวอักษร · บล็อกนี้สั่งแสดงผลให้เองทุกครั้ง ไม่ต้องกลัวลืมเหมือนตอนเขียนโค้ดเอง

ขา	รับ/ส่งค่า
ขาเข้า: ค่า	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
บรรทัด (0–7)	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 7	
ตำแหน่งตัวอักษร (0–20)	ตัวเลข	_	ต่ำสุด 0 · สูงสุด 20	
ข้อความนำหน้า	ข้อความ	(ว่าง)	เว้นว่างได้	
ล้างจอก่อนเขียน	ติ๊กถูก	ไม่ติ๊ก	—	

🔔 เสียงบี๊บ

ใช้บอกสถานะโดยไม่ต้องมองจอ เช่น บี๊บตอนเจอเส้น · เสียงไม่หยุดการทำงานของบล็อกอื่น

ขา	รับ/ส่งค่า
ขาเข้า: สั่งให้ดัง	ค่าใดก็ได้

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ความถี่	ตัวเลข	_	ต่ำสุด 40 · สูงสุด 5000 · เฮิรตซ์	
นานเท่าไร	ตัวเลข	_	ต่ำสุด 1 · สูงสุด 5000 · มิลลิวินาที	

เซนเซอร์ (1)

🪰 ตาแมลง (จับการเคลื่อนไหว) ⚠️ (ยังไม่ได้ทดสอบกับอุปกรณ์จริง)

สมองแมลงวันจิ๋ว (Reichardt EMD): ตาแต่ละคู่จับว่าภาพเลื่อนไปทางไหน แล้วเทียบ **ฝั่งซ้าย** กับ **ฝั่งขวา** — ผนังฝั่งไหนใกล้กว่าจะดูเลื่อนเร็วกว่า บล็อกจึงสั่ง **เลี้ยวหนี** ฝั่งนั้น (บวก = เลี้ยวขวา) · แสงทั้ง

ห้องที่สว่าง/มืดลงพร้อมกันไม่ทำให้เลี้ยว · ส่งค่าออก 20 ครั้งต่อวินาที

ขา	รับ/ส่งค่า
ขาออก: เลี้ยว (−100 ถึง 100)	ตัวเลข
ขาออก: การเคลื่อนไหวฝั่งซ้าย	ตัวเลข
ขาออก: การเคลื่อนไหวฝั่งขวา	ตัวเลข

ช่องตั้งค่า	ชนิด	ค่าเริ่มต้น	ช่วง / ตัวเลือก	คำอธิบาย
ขาของตา (เรียงจากซ้ายไปขวา)	รายการขา/ช่อง (เรียงซ้าย→ขวา)	เลือกให้ตามบอร์ด (POP-32i: A0–A7)	2–8 ขา	ตาดวงแรก = ซ้ายสุด · ต้องเป็นขาที่อ่านแอนะล็อกได้ · ตา 4 ดวงขึ้นไปจึงแบ่งซ้าย/ขวาได้
การหน่วงของตา (τ)	ตัวเลข	—	ต่ำสุด 5 · สูงสุด 2000 · มิลลิวินาที	ตาไวต่อภาพที่เลื่อนเร็วประมาณ ระยะห่างของตา ÷ τ · ค่าน้อย = ชอบภาพเร็ว · ค่ามาก = ชอบภาพช้า
แรงเลี้ยว	ตัวเลข	—	—	เลี้ยว = แรงเลี้ยว × (ซ้าย − ขวา) แล้วตัดที่ ±100
ไม่สนใจถ้าต่างกันน้อยกว่า	ตัวเลข	—	ต่ำสุด 0 · หน่วย	กันรถส่ายเพราะสัญญาณรบกวนเล็ก ๆ
อ่านตาทุกดวง	ตัวเลข	—	ต่ำสุด 20 · สูงสุด 1000 · ครั้ง/วินาที	
กลับเครื่องหมาย (เลี้ยวเข้าหาแทนเลี้ยวหนี)	ติ๊กถูก	ไม่ติ๊ก	—	

8. ตัวอย่างสำเร็จรูปของ POP-32i (3 ชิ้น)

กดจากคลังบล็อก หัวข้อ **ตัวอย่างสำเร็จรูป** แล้วขึ้นแคนวาสทันที

ตัวอย่าง	สอนอะไร	ใช้บล็อก
〰 หุ่นตามเส้น	เซนเซอร์เส้นซ้ายคุมล้อซ้าย เซนเซอร์ขวาคุมล้อขวา หุ่นจะเลี้ยวกลับเข้าหาเส้นเอง — ต้องคาลิเบรตค่าพื้นดำ/พื้นขาวก่อนใช้จริง	เซนเซอร์เส้น (0–100) · ขับหุ่น (ซ้าย/ขวาแยกกัน)
🚦 กดปุ่มแล้วออกตัว	กด SW_OK แล้วหุ่นเดินหน้า ปล่อยแล้วหยุด พร้อมบี๊บและขึ้นสถานะบนจอ — ท่ามาตรฐานก่อนออกตัวในสนามแข่ง	ปุ่มบนบอร์ด · ขับหุ่น (สองล้อ) · เสียงบี๊บ · แสดงบนจอบอร์ด

ตัวอย่าง	สอนอะไร	ใช้บล็อก
🪰 **รถสมองแมลงวัน (FLYCAR)**	ตาแมลง 8 ดวงที่พอร์ต A0–A7 เทียบว่าผนังฝั่งไหนเลื่อนผ่านเร็วกว่า แล้วสั่งเลี้ยวหนี (ล้อซ้าย M1 = ความเร็ว + เลี้ยว · ล้อขวา M2 = ความเร็ว – เลี้ยว) · ล้อแต่ละข้างจำกัดไว้ ±70 กันมอเตอร์ได้ไฟเกิน · จอบอร์ดแสดงค่าเลี้ยว · ถ้ารถต่อล้อคนละช่อง แก้ที่บล็อกขับหุ่น	ตาแมลง (จับการเคลื่อนไหว) · เมื่อเริ่มทำงาน · แทนที่ด้วยค่าคงที่ · คำนวณสองค่า · ขับหุ่น (ซ้าย/ขวาแยกกัน) · แสดงบนจอบอร์ด

9. ข้อความเตือนที่เจอบ่อย (ข้อความจริงจากแผงตรวจกราฟ)

สีแดง = บอร์ดจะไม่รับกราฟจนกว่าจะแก้ · สีเหลือง = ส่งได้แต่ควรรู้ไว้

เกิดเมื่อ	หน้าจอขึ้นว่า
ใส่บล็อกเกิน 24 ตัว	กราฟมี 26 บล็อก เกินที่ “POP-32i (INEX)” รับไหว (สูงสุด 24) — บอร์ดนี้มีหน่วยความจำ 20 KB
กราฟยาวเกินที่บอร์ดรับได้ต่อครั้ง (ตัวที่เจอบ่อยกว่าจำนวนบล็อก)	กราฟยาว 2210 ไบต์ เกินที่ “POP-32i (INEX)” รับได้ต่อครั้ง (สูงสุด 24) — ลดจำนวนบล็อก หรือย่อข้อความ/ชื่อบล็อกให้สั้นลง (ตัวอักษรไทยใช้ 3 ไบต์)
สั่งมอเตอร์/เซอร์โวช่องเดียวกันจากสองบล็อก	ช่อง M1 ถูกสั่งจากสองที่ (บล็อก “ขับหุ่น (สองล้อ)” จองไว้แล้ว) — มอเตอร์หรือเซอร์โวหนึ่งช่องควรถูกสั่งจากที่เดียว ไม่งั้นจะกระตุก
ใช้พอร์ต A ช่องเดียวเป็นทั้งขาเข้าและขาออก	พอร์ต M1 ถูกใช้เป็นทั้งขาออก (บล็อก “ขับหุ่น (สองล้อ)”) และขาเข้า/ขาออกซ้ำอีกที่ — ขาเดียวสลับโหมดไปมาจะอ่านค่าผิดและสั่งอุปกรณ์ไม่ติด ใช้คนละพอร์ตกัน
เลือกช่องที่บอร์ดไม่มี	ไม่มีช่อง “M1” บน “POP-32i (INEX)” — เลือกได้ 1 ถึง 4
เปิดไฟล์ที่ทำไว้สำหรับบอร์ด ESP32 (มีบล็อกที่ POP-32i ไม่มี)	บล็อก “PWM หรี่ไฟ / ปรับความเร็ว” ใช้กับบอร์ด “POP-32i (INEX)” ไม่ได้ — ลบออก หรือสลับบอร์ดกลับไปเป็นรุ่นที่รองรับ
พิมพ์ภาษาไทยลงจอบนบอร์ด	จอบนบอร์ดแสดงได้เฉพาะตัวอักษรอังกฤษ ตัวเลข และเครื่องหมาย — ตัวอักษรไทยจะขึ้นเป็น ?
บล็อกต้นทางที่ไม่ได้ต่อไปไหน	บล็อก “ปุ่มบนบอร์ด” ยังไม่ได้ต่อไปไหน จะไม่มีผลอะไร
ต่อเส้นวนกลับเป็นวงกลม	กราฟต่อเป็นวงกลม (A → B → A) บอร์ดจะตัดการทำงานเมื่อวนลึกเกินไป

10. แก้ปัญหาหน้างาน

(ก) กดเชื่อมต่อบอร์ดแล้วไม่เห็นพอร์ตเลย

1. **เปิดสวิตช์บอร์ดหรือยัง / แบตหมดไหม** — POP-32i ไม่ติดจากไฟ USB อย่างเดียว ไม่เปิดสวิตช์ = คอมมองไม่เห็นเลย

2. เสียบ **ช่องล่าง** หรือยัง (ช่องบนใช้ติดตั้งเฟิร์มแวร์เท่านั้น)
3. เปลี่ยนสาย USB — สายชาร์จอย่างเดียวจ่ายไฟได้แต่ไม่มีเส้นข้อมูล
4. ใช้ Chrome/Edge บนคอมพิวเตอร์ ไม่ใช่ iPad/Safari/Firefox
5. Linux: ผู้ใช้ต้องอยู่กลุ่ม `dialout` แล้วออก-เข้าระบบใหม่)

(ข) เลือกพอร์ตแล้วแต่ต่อไม่ติด / หมดเวลา

- บอร์ดยังไม่มีเฟิร์มแวร์ MakerFlow (บอร์ดใหม่ หรือเคยอัปโหลดโปรแกรม Arduino ทับ) → ติดตั้งตามหัวข้อ 4
- ปิดแท็บ MakerFlow อื่นหรือโปรแกรมอื่นที่เปิดพอร์ตค้างอยู่ (Arduino IDE Serial Monitor) — พอร์ตหนึ่งเปิดได้ทีละโปรแกรม
- ถอดสาย กด RESET บนบอร์ด แล้วเสียบใหม่

(ค) ส่งลงบอร์ดไม่ได้

- มีข้อความ **สีแดง** ในแผงตรวจกราฟ — แก้ตามข้อความก่อน (หัวข้อ 9)
- ขึ้นว่า "ไม่รู้จักบล็อกชนิด ..." = เฟิร์มแวร์บนบอร์ดเก่ากว่าหน้าเว็บ → แฟลชรุ่น 0.4.0-pop32

(ง) ส่งสำเร็จแต่หุ่นไม่ทำอะไร

1. กราฟมีบล็อก **ต้นทาง** ไหม (ตัวจับเวลา · เมื่อเริ่มทำงาน · ปุ่ม · เซนเซอร์)
2. เปิด **ดูค่าสด** ไล่ดูว่าสัญญาณไปหยุดที่บล็อกไหน
3. มอเตอร์ไม่หมุน: ตรวจช่องมอเตอร์ในบล็อกตรงกับช่องที่เสียบจริง · แบตอ่อน (ไฟ LOW BATT แดง)

(จ) จอบนบอร์ดขึ้นเครื่องหมาย ? — จอแสดงภาษาไทยไม่ได้ ใช้อักษรอังกฤษ

(ฉ) หุ่นวิ่งต่อทั้งที่ถอดสายแล้ว — ตั้งใจให้เป็นแบบนั้น กด **หยุด** บนเว็บ หรือปิดสวิตช์บอร์ด

(ช) STM32CubeProgrammer กด Connect แล้วขึ้น error

- เปิดสวิตช์บอร์ดแล้วหรือยัง · เสียบ **ช่องบน** หรือยัง · ไดรเวอร์ติดตั้งครบหรือไม่ (หัวข้อ 4.2 ข้อ 3)

ภาพส่วนประกอบบอร์ดและภาพช่อง SWD จากสไลด์หลักสูตร POP-BOT32i ของ INEX · MakerFlow โดย MakerHub Co., Ltd. · ฉบับออนไลน์ล่าสุด: https://maker-hub.net/manual/makerflow-pop32i.html