ENV//MONITOR

เอกสารเทคนิครายอุปกรณ์ – Raspberry Pi 5

เวอร์ชัน 1.0
วันที่ 2026-08-01
ขอบเขต เอกสารรายอุปกรณ์เท่านั้น — สเปก, การต่อสาย, โปรโตคอล, การ calibrate, การแก้ปัญหา
ที่มาของข้อมูล datasheet ทางการของผู้ผลิตแต่ละราย

PMS7003	ฝุ่นละออง PM1.0 / PM2.5 / PM10	UART 9600
SCD40	CO_2 · อุณหภูมิ · ความชื้น	I2C 0x62
SGP30	TVOC · eCO_2	I2C 0x58
BH1750	ความสว่าง (lux)	I2C 0x23
USB Microphone	ระดับเสียง (dB)	USB Audio Class
กล้อง	ใบหน้า · EAR · การขยับ	CSI-2 / USB

สารบัญ

PMS7003 — ฝุ่นละออง PM1.0 / PM2.5 / PM10 5

1. สเปกจาก datasheet (V2.5, 2016-06-01) 5
2. ขาสัญญาณ 6
3. โปรโตคอล UART 6
4. การอ่านค่าใน Python 8
5. การ calibrate 8
6. ข้อจำกัดการใช้งานตาม datasheet 9
7. เกณฑ์ค่าที่แนะนำสำหรับแดชบอร์ด 9
8. การแก้ปัญหา 10

SCD40 — CO_2 · อุณหภูมิ · ความชื้น 11

1. สเปกจาก datasheet (v1.7, เมษายน 2025) 11
2. คำเตือนเรื่องแรงดัน 12
3. การต่อสาย 13
4. คำสั่ง I2C ที่ใช้ 13
5. ไลบรารี Python 14
6. Temperature offset — ขั้นตอนที่สำคัญที่สุด 14
7. การชดเชยความดันบรรยากาศ 15
8. การ calibrate CO_2 — ASC เทียบกับ FRC 15
9. เกณฑ์ค่า CO_2 ที่แนะนำสำหรับแดชบอร์ด 16
10. การแก้ปัญหา 17

SGP30 — TVOC · eCO_2 18

1. สเปกจาก datasheet (v0.9, สิงหาคม 2017) 18
2. คำเตือนเรื่องแรงดัน 19
3. การต่อสาย 19
4. คำสั่ง I2C 20
5. ข้อบังคับ: ต้องอ่านทุก 1 วินาทีเป๊ะ ๆ 20
6. Baseline — สิ่งที่ต้องทำให้ถูก ไม่งั้นค่าไร้ความหมาย 21
7. Humidity compensation 22
8. สารที่ทำให้เซนเซอร์เสื่อมถาวร 22
9. ไลบรารี Python 23
10. เกณฑ์ค่า TVOC ที่แนะนำ 23

11. การแก้ปัญหา 24

BH1750 — ความสว่าง (lux) **25**

1. สเปกจาก datasheet 25
2. ที่อยู่ I2C 25
3. การต่อสาย 26
4. คำสั่งและโหมดการวัด 26
5. ไลบรารี Python 27
6. การ calibrate 27
7. เกณฑ์ค่าความสว่างที่แนะนำ 28
8. การแก้ปัญหา 28

USB Microphone — ระดับเสียง (dB) **29**

1. ทำไมเลือก USB แทน I2S หรือ analog 29
2. การเลือกไมโครโฟน 29
3. การตั้งค่าบน Raspberry Pi OS 30
4. การคำนวณ dB 31
5. การ calibrate 32
6. เกณฑ์ระดับเสียงที่แนะนำ 33
7. ผลกระทบต่องบไฟและ USB 33
8. การแก้ปัญหา 34

กล้อง — ใบหน้า · EAR · การขยับ **35**

1. สรุปคำแนะนำ 35
2. ทำไม Camera Module 3 ถึงเหมาะกว่า USB webcam 35
3. สเปก Raspberry Pi Camera Module 3 36
4. การติดตั้ง Camera Module 3 37
5. ซอฟต์แวร์ 37
6. ความเป็นส่วนตัว — สำคัญมาก 38
7. ผลกระทบต่องบไฟและ CPU 39
8. การแก้ปัญหา 40

ทบทวนวรรณกรรม — งานวิจัยรองรับ · ช่องว่าง · กรอบแนวคิด · สมมติฐาน **41**

2.1 ที่มาและความสำคัญ 41

2.2 วิธีสืบค้นวรรณกรรม 41

2.3 ตารางสังเคราะห์งานวิจัย 42

2.4 คาร์บอนไดออกไซด์ (CO_2) 42

2.5 อุณหภูมิ 43
2.6 ฝุ่นละอองขนาดเล็ก (PM2.5) 44
2.7 เสียงรบกวน 44
2.8 แสงสว่าง 45
2.9 สารอินทรีย์ระเหยง่ายรวม (TVOC) 45
2.10 ช่องว่างของงานวิจัย (Research Gap) 46
2.11 กรอบแนวคิดการวิจัย (Conceptual Framework) 47
2.12 สมมติฐานการวิจัย 47
2.13 ข้อจำกัดของโครงงาน 48
2.14 รายการอ้างอิง 49

งบประมาณ — ราคาชิ้นส่วน · ชุดที่แนะนำ · ลำดับการซื้อ 52

1. ขอบเขตของงบนี้ 52
2. รายการที่ต้องซื้อ 53
3. สรุปงบตามชุดที่เลือก 54
4. สิ่งที่ตัดออกไปแล้วและประหยัดได้ 54
5. ค่าใช้จ่ายต่อเนื่อง 55
6. ร้านค้าที่ตรวจสอบราคา 55
7. ลำดับการซื้อที่แนะนำ 55

สรุปอุปกรณ์ทั้งหมด (1 หน้า) 57

PMS7003 — เซนเซอร์ฝุ่นละออง PM1.0 / PM2.5 / PM10

1. สเปกจาก datasheet (V2.5, 2016-06-01)

รายการ	ค่า
หลักการทำงาน	Laser scattering (MIE theory)
ขนาดอนุภาคเล็กสุดที่แยกแยะได้	0.3 µm
ช่วงวัด (PM2.5 มาตรฐาน)	0–500 µg/m³ (สูงสุด ≥1000 µg/m³)
ความละเอียด	1 µg/m³
ประสิทธิภาพการนับ	50% @ 0.3 µm · 98% @ ≥0.5 µm
ความคลาดเคลื่อนสูงสุด	±10 µg/m³ @ 0–100 · ±10% @ 100–500 µg/m³
ปริมาตรมาตรฐาน	0.1 ลิตร
เวลาตอบสนอง (ครั้งเดียว)	< 1 วินาที
เวลาตอบสนองรวม	≤ 10 วินาที
ไฟเลี้ยง	5.0V (ต่ำสุด 4.5V · สูงสุด 5.5V)
กระแสตอนทำงาน	≤ 100 mA
กระแสตอน standby	≤ 200 µA
ระดับลอจิก	**3.3V** (L < 0.8V · H > 2.7V)
อุณหภูมิใช้งาน	−10 ถึง +60 °C
ความชื้นใช้งาน	0–99 %RH
MTTF	≥ 3 ปี
ขนาด	48 × 37 × 12 mm

2. ขาสัญญาณ

ขา	ชื่อ	หน้าที่	ต่อไปที่ Pi 5
1	VCC	+5V	ขา 4 (5V)
2	VCC	+5V	ขา 4 (5V) — **ต่อทั้งสองขา**
3	GND	กราวด์	ขา 9 (GND)
4	GND	กราวด์	ขา 9 (GND) — **ต่อทั้งสองขา**
5	RESET	รีเซ็ตโมดูล (active low, 3.3V)	GPIO27 (ขา 13) — *ทางเลือก*
6	NC	—	✗ **ห้ามต่อ**
7	RX	รับข้อมูล (3.3V)	GPIO14 / TXD (ขา 8)
8	NC	—	✗ **ห้ามต่อ**
9	TX	ส่งข้อมูล (3.3V)	GPIO15 / RXD (ขา 10)
10	SET	HIGH/ลอย = ทำงาน · LOW = sleep	GPIO17 (ขา 11) — *ทางเลือก*

> ขา RESET และ SET มี pull-up ภายในอยู่แล้ว **ถ้าไม่ใช้ให้ปล่อยลอยไว้ ห้ามต่อลง GND**

ทำไมไม่ต้องใช้ level shifter

ลอจิกของ PMS7003 คือ 3.3V ตรงกับ Pi พอดี:

- Pi TXD ส่งออก 3.3V → PMS7003 ต้องการ H > 2.7V ✓
- PMS7003 TX ส่งออก 3.3V → Pi รับได้สูงสุด 3.3V ✓

datasheet ระบุว่า level shifter จำเป็น**เฉพาะเมื่อ MCU ฝั่งโฮสต์ใช้ไฟ 5V** ซึ่งไม่ใช่กรณีของ Pi 5

---

3. โปรโตคอล UART

พารามิเตอร์พอร์ต: 9600 bps · 8 data bits · **ไม่มี parity** · 1 stop bit · ไม่มี flow control

โหมดการทำงาน

โหมด	พฤติกรรม	ค่าเริ่มต้น
Active	เซนเซอร์ push เฟรมออกมาเอง ไม่ต้องร้องขอ	✓ ค่าเริ่มต้นหลังจ่ายไฟ
Passive	ต้องส่งคำสั่ง `0xE2` ถึงจะตอบกลับ	ต้องสั่งเปลี่ยนโหมดเอง

โหมด Active แบ่งย่อยอีกสองแบบ ซึ่งเซนเซอร์สลับให้เอง:

- **Stable** — ค่าเปลี่ยนแปลงน้อย → ส่งเฟรมทุก **2.3 วินาที**
- **Fast** — ค่าเปลี่ยนแปลงมาก → ส่งเฟรมทุก **200–800 ms** (ยิ่งฝุ่นเยอะยิ่งถี่)

> **แนะนำให้ใช้โหมด Active ตามค่าเริ่มต้น** อ่านแบบ event-driven จาก UART buffer ไม่ต้องเขียนโค้ดสลับโหมดเอง เรียบง่ายและตรงกับที่ผู้ผลิตออกแบบมา

โครงสร้างเฟรม 32 ไบต์

```
byte:  0    1    2  3    4  5    6  7    8  9   ...  28 29   30 31
      +----+----+-----+-----+-----+-----+-----+-----+-----+--------+
      |0x42|0x4D| len |Data1|Data2|Data3|Data4|Data5| ... |checksum|
      +----+----+-----+-----+-----+-----+-----+-----+-----+--------+
       header     28   PM1.0 PM2.5 PM10  PM1.0 PM2.5       sum of
                      (CF=1)(CF=1)(CF=1)(atmospheric)     byte 0-29
```

ตำแหน่ง	ฟิลด์	ความหมาย	ใช้ไหม
0–1	หัวเฟรม	`0x42 0x4D` คงที่	ใช้หาจุดเริ่มเฟรม
2–3	ความยาว	= 28 (`2×13 + 2`)	ตรวจความถูกต้อง
4–5	Data1	PM1.0 (CF=1, particle มาตรฐาน)	✗
6–7	Data2	PM2.5 (CF=1, particle มาตรฐาน)	✗
8–9	Data3	PM10 (CF=1, particle มาตรฐาน)	✗
10–11	**Data4**	**PM1.0 (สภาพบรรยากาศ)**	✓ **ใช้ตัวนี้**
12–13	**Data5**	**PM2.5 (สภาพบรรยากาศ)**	✓ **ใช้ตัวนี้**
14–15	**Data6**	**PM10 (สภาพบรรยากาศ)**	✓ **ใช้ตัวนี้**
16–17	Data7	จำนวนอนุภาค > 0.3 μm ต่อ 0.1 L	ทางเลือก
18–19	Data8	จำนวนอนุภาค > 0.5 μm ต่อ 0.1 L	ทางเลือก
20–21	Data9	จำนวนอนุภาค > 1.0 μm ต่อ 0.1 L	ทางเลือก
22–23	Data10	จำนวนอนุภาค > 2.5 μm ต่อ 0.1 L	ทางเลือก
24–25	Data11	จำนวนอนุภาค > 5.0 μm ต่อ 0.1 L	ทางเลือก
26–27	Data12	จำนวนอนุภาค > 10 μm ต่อ 0.1 L	ทางเลือก
28–29	Data13	สงวนไว้	✗
30–31	Checksum	ผลรวมของไบต์ 0–29	✓ **ต้องตรวจทุกเฟรม**

ทุกค่าเป็น **unsigned 16-bit big-endian** (ไบต์สูงมาก่อน)

KEY CF=1 กับ "สภาพบรรยากาศ" ต่างกันอย่างไร

- **CF=1 (Data1–3)** — ค่าที่คำนวณโดยใช้ค่าคงที่ปรับเทียบสำหรับ**ฝุ่นในโรงงาน** datasheet ระบุตรง ๆ ว่า *"CF=1 should be used in the factory environment"*
- **สภาพบรรยากาศ (Data4–6)** — ค่าที่ปรับเทียบสำหรับ**อากาศทั่วไป** ซึ่งเป็นกรณีใช้งานของโปรเจกต์นี้

> **ใช้ Data4/5/6 เท่านั้น** เป็นข้อผิดพลาดที่พบบ่อยมากที่ไลบรารีสำเร็จรูปหลายตัวอ่าน Data1–3 มาให้ ที่ความเข้มข้นต่ำสองชุดนี้มักให้ค่าใกล้เคียงกัน แต่จะแยกออกจากกันชัดเจนเมื่อฝุ่นสูง

การตรวจ checksum

```
checksum = ผลรวมของไบต์ที่ 0 ถึง 29 (รวมหัวเฟรม 0x42 และ 0x4D ด้วย)
```

เทียบกับค่า 16-bit ที่ไบต์ 30–31 **ถ้าไม่ตรงให้ทิ้งเฟรมนั้นทั้งเฟรม** แล้วหาหัวเฟรมใหม่ อย่าพยายามซ่อมข้อมูล — เฟรมถัดไปมาในไม่กี่วินาทีอยู่แล้ว

คำสั่งโหมด Passive (ถ้าจำเป็นต้องใช้)

โครงสร้าง 7 ไบต์: `0x42 0x4D <CMD> <DATAH> <DATAL> <LRCH> <LRCL>`

CMD	DATAL	ความหมาย
`0xE2`	—	อ่านค่าครั้งเดียว (ตอบกลับ 32 ไบต์เหมือนโหมด active)
`0xE1`	`0x00` / `0x01`	เปลี่ยนโหมด: passive / active
`0xE4`	`0x00` / `0x01`	sleep / wake

Verify bytes = ผลรวมของทุกไบต์ยกเว้นสองไบต์สุดท้าย

---

4. การอ่านค่าใน Python

ไลบรารีที่แนะนำ: **ไม่ต้องใช้เลย** — โปรโตคอลง่ายพอที่จะเขียนเองด้วย `pyserial` ซึ่งดีกว่าเพราะควบคุมการตรวจ checksum และการเลือก Data4–6 ได้เอง

```
pip install pyserial
```

โครงสร้าง parser ที่ถูกต้อง (pseudo-code — ค่าคงที่ทุกตัวต้องมาจาก config):

```
1. เปิดพอร์ตด้วย timeout สั้น ๆ
2. อ่านทีละไบต์จนเจอ 0x42 ตามด้วย 0x4D          ← sync หาหัวเฟรม
3. อ่านอีก 30 ไบต์
4. ตรวจว่า frame_length == 28                     ← ถ้าไม่ใช่ กลับไปข้อ 2
5. คำนวณ checksum ของ 30 ไบต์แรก เทียบกับไบต์ 30-31
6. ถ้าไม่ตรง → ทิ้ง กลับไปข้อ 2
7. แกะ Data4, Data5, Data6 เป็น big-endian uint16
8. ถ้ายังไม่ครบ 30 วินาทีนับจากเปิดพัดลม → ทำเครื่องหมายว่ายัง warming up
```

สิ่งที่ต้องระวังในโค้ด:

- **ล้าง input buffer ก่อนอ่าน** ถ้าเว้นช่วงนาน — ไม่งั้นจะได้เฟรมเก่าค้างในบัฟเฟอร์
- **ตั้ง timeout เสมอ** ไม่งั้น thread ค้างถาวรเมื่อสายหลุด
- อย่าเปิด/ปิดพอร์ตซ้ำ ๆ ทุกครั้งที่อ่าน — เปิดครั้งเดียวตอนเริ่ม thread

---

5. การ calibrate

ไม่มีขั้นตอน calibrate สำหรับผู้ใช้ — PMS7003 ปรับเทียบมาจากโรงงานและ datasheet ระบุชัดว่า *"the sensor should not be checked with any third party equipment"*

สิ่งที่ต้องทำมีแค่:

รายการ	รายละเอียด
Warm-up	รอ **30 วินาที** หลังจ่ายไฟหรือปลุกจาก sleep ก่อนเชื่อค่าใด ๆ
ทำความสะอาด	เป่าลมแรงดันต่ำผ่านช่องลมทุก 6–12 เดือน · **ห้ามใช้ของเหลว**
อายุการใช้งาน	MTTF ≥ 3 ปี — หลังจากนั้นค่าจะ drift และควรเปลี่ยนตัวใหม่

6. ข้อจำกัดการใช้งานตาม datasheet

datasheet ระบุว่าเซนเซอร์นี้ออกแบบมาสำหรับ**สภาพในร่มทั่วไป** และต้องเพิ่มการป้องกันถ้าใช้ใน:

- สภาพที่ความเข้มข้น ≥300 μg/m³ นานเกิน 50% ของปี หรือ ≥500 μg/m³ นานเกิน 20% ของปี
- **ห้องครัว** (ไอมันเคลือบเลนส์)
- **ห้องน้ำ / ที่มีละอองน้ำ** (ควบแน่นบนออปติก)
- **กลางแจ้ง**

> ! ในบริบทประเทศไทย ช่วงฤดูฝุ่น (ธ.ค.–มี.ค.) ค่า PM2.5 กลางแจ้งอาจเกิน 300 μg/m³ ได้ ถ้าติดตั้งในห้องที่ไม่มีเครื่องฟอกอากาศและเปิดหน้าต่าง อายุเซนเซอร์จะสั้นกว่าที่ระบุ

ข้อกำหนดการติดตั้ง

1. ติดตั้งให้ระนาบช่องลมเข้า-ออกแนบกับผนังเครื่อง หรือมีแผงกั้นระหว่างช่องเข้ากับช่องออก **เพื่อไม่ให้อากาศวนกลับ**
2. ช่องเจาะบนฝาครอบต้อง**ไม่เล็กกว่า**ช่องลมเข้าของเซนเซอร์
3. ติดตั้ง**สูงจากพื้นอย่างน้อย 20 cm** เพื่อไม่ให้ฝุ่นหยาบและขนสัตว์อุดตัน
4. **ห้ามวางในทางลมของเครื่องฟอกอากาศ** — ต้องมีโครงสร้างกำบัง
5. เปลือกโลหะต่อกับ GND — ระวังไม่ให้ไปแตะวงจรส่วนอื่น
6. **ห้ามแกะเซนเซอร์**

7. เกณฑ์ค่าที่แนะนำสำหรับแดชบอร์ด

อ้างอิงมาตรฐานฝุ่น PM2.5 ในบรรยากาศทั่วไปของไทย (ประกาศคณะกรรมการสิ่งแวดล้อมแห่งชาติ มีผลบังคับใช้ 1 มิถุนายน 2566) และแนวทางองค์การอนามัยโลก (WHO AQG 2021)

PM2.5 (μg/m³)

ระดับ	ช่วง	ที่มา
ดีมาก	0–15	WHO AQG 2021 ค่าเฉลี่ย 24 ชม.
ดี	15–25	WHO interim target 4
ปานกลาง	25–37.5	**37.5 = มาตรฐานไทย 24 ชม. (พ.ศ. 2566)**
เริ่มมีผลต่อสุขภาพ	37.5–75	WHO interim target 1
มีผลต่อสุขภาพ	> 75	

PM10 (µg/m³)

ระดับ	ช่วง
ดี	0–45
ปานกลาง	45–120
เริ่มมีผลต่อสุขภาพ	120–180
มีผลต่อสุขภาพ	> 180

> ! มาตรฐานเหล่านี้เป็น**ค่าเฉลี่ย 24 ชั่วโมงในบรรยากาศทั่วไป** ไม่ใช่ค่าชั่วขณะในอาคาร การเอาค่าอ่านทุก 5 วินาทีมาเทียบตรง ๆ จะทำให้เตือนบ่อยเกินจริง **แนะนำ:** แสดงค่าปัจจุบันบนแดชบอร์ด แต่ใช้ค่าเฉลี่ยเคลื่อนที่ (rolling average) 1 ชั่วโมงในการตัดสินระดับเตือน

ค่าเกณฑ์เหล่านี้ต้องไปอยู่ใน `config/sensors.js → THRESHOLDS` เท่านั้น ห้ามเขียนตัวเลขซ้ำในคอมโพเนนต์ UI → ดู `docs/05-data-schema.md`

---

8. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
อ่านได้แต่ค่าเป็นขยะ	baud rate ผิด หรือ TX/RX ไม่ได้ไขว้	ยืนยัน 9600 8N1 · สลับสาย TX ↔ RX
ไม่มีข้อมูลเลย	serial console ยังยึด `/dev/ttyAMA0` อยู่	`sudo raspi-config nonint do_serial_cons 1` แล้วรีบูต
ไม่มีข้อมูลเลย + พัดลมไม่หมุน	ไฟไม่ถึง 5V หรือขา SET ถูกดึงลง LOW	วัดแรงดันที่ขา 1 · ปล่อยขา SET ลอย
ค่า PM ต่ำผิดปกติเสมอ (~0)	ยังไม่ครบ warm-up 30 วินาที · ช่องลมถูกบัง	รอให้ครบ · ตรวจสิ่งกีดขวางช่องลม
checksum ผิดบ่อย	สายยาวเกินไป หรือกราวด์ไม่ร่วม	สายสั้นลง · ตรวจ GND · ลอง `dtoverlay=disable-bt`
ค่าสูงผิดปกติต่อเนื่อง	ควันบุหรี่/ธูป/การทำอาหารใกล้เคียง หรือเลนส์สกปรก	ย้ายตำแหน่ง · เป่าลมทำความสะอาด
พัดลมมีเสียงครวญ	ตลับลูกปืนเสื่อม (ใกล้หมดอายุ)	เตรียมเปลี่ยนตัวใหม่

SCD40 — เซนเซอร์ CO_2 / อุณหภูมิ / ความชื้น

> **บทบาทในระบบนี้:** SCD40 เป็น**แหล่งข้อมูลหลัก**ของทั้ง CO_2, อุณหภูมิ และความชื้น มาแทน DHT22 เดิมทั้งหมด — แม่นกว่า, มี CRC, และไม่มีปัญหา timing ที่ DHT มี

1. สเปกจาก datasheet (v1.7, เมษายน 2025)

การวัด CO_2

รายการ	ค่า
หลักการทำงาน	Photoacoustic NDIR
ช่วงค่าที่ส่งออก	0–40,000 ppm
ความแม่นยำ	**±(50 ppm + 5% ของค่าที่อ่าน)** ที่ 400–2,000 ppm
ความสามารถทำซ้ำ	±10 ppm
เวลาตอบสนอง (τ63%)	60 วินาที (เปลี่ยนจาก 400 → 2,000 ppm)
การ drift หลังปีที่ 5	±(5 ppm + 0.5% ของค่าที่อ่าน) ต่อปี

การวัดความชื้น

รายการ	ค่า
ช่วงวัด	0–100 %RH
ความแม่นยำ	±6 %RH (ที่ 15–35 °C, 20–65 %RH) · ±9 %RH (ที่ −10–60 °C, 0–100 %RH)
ความสามารถทำซ้ำ	±0.4 %RH
เวลาตอบสนอง (τ63%)	90 วินาที
การ drift	< 0.25 %RH/ปี

การวัดอุณหภูมิ

รายการ	ค่า
ช่วงวัด	−10 ถึง 60 °C
ความแม่นยำ	±0.8 °C (ที่ 15–35 °C) · ±1.5 °C (ที่ −10–60 °C)
ความสามารถทำซ้ำ	±0.1 °C
เวลาตอบสนอง (τ63%)	120 วินาที
การ drift	< 0.03 °C/ปี

ไฟฟ้า

รายการ	ค่า
แรงดันไฟเลี้ยง	2.4 – 5.5V (ปกติ 3.3V หรือ 5.0V)
Ripple ที่ยอมรับได้	**≤ 30 mV peak-to-peak** !
กระแสสูงสุด (peak)	175 mA ปกติ / **205 mA สูงสุด** ที่ VDD = 3.3V
กระแสเฉลี่ย (รอบ 5 วินาที)	15 mA ปกติ / 18 mA สูงสุด ที่ VDD = 3.3V
กระแสเฉลี่ย (โหมดประหยัด รอบ 30 วินาที)	3.2 mA ปกติ / 3.5 mA สูงสุด
แรงดัน input HIGH	≥ 0.65 × VDD
แรงดัน input LOW	≤ 0.3 × VDD
เวลาบูตหลังจ่ายไฟ	30 ms
ความถี่ SCL	0 – 400 kHz
ที่อยู่ I2C	`0x62` (ตายตัว เปลี่ยนไม่ได้)
อายุการใช้งาน	> 10 ปี
ความชื้นที่ทำงานได้	0–95 %RH (ไม่ควบแน่น)

2. ! คำเตือนเรื่องแรงดัน

SCD40 ในระบบนี้ต้องจ่ายไฟ 3.3V เท่านั้น

datasheet ระบุว่ารับไฟได้ 2.4–5.5V ซึ่งถูกต้อง — แต่ระดับลอจิกของขา SDA/SCL **อ้างอิงกับ VDD**:

VDD	V_{IH} (ต้องการอย่างน้อย)	ระดับสัญญาณที่เซนเซอร์ปล่อยออกมาบนบัส
3.3V	2.15V	0 – 3.3V ✓ ปลอดภัยกับ Pi
5.0V	3.25V	0 – 5.0V ✗ **ทำ GPIO ของ Pi พัง**

Raspberry Pi ทนได้สูงสุด 3.3V บนขา GPIO **การจ่ายไฟ 5V ให้ SCD40 จะทำให้ pull-up ภายในเซนเซอร์ ดึงบัส I2C ขึ้นไป 5V และทำลาย GPIO ของ Pi อย่างถาวร**

> ✓ ต่อ VDD ของ SCD40 เข้าขา 1 (3V3) ของ Pi ✗ **ห้าม**ต่อเข้าขา 2 หรือ 4 (5V) เด็ดขาด CHECK วัดด้วยมัลติมิเตอร์ยืนยันก่อนเสียบไฟครั้งแรก

ผลข้างเคียงที่ต้องยอมรับ: ที่ 3.3V เซนเซอร์กินกระแส peak สูงกว่า (205 mA เทียบกับ 137 mA ที่ 5V) จึงต้องมี bulk capacitor → ดู `docs/03-power-budget.md`

3. การต่อสาย

SCD40	หน้าที่	→ Pi 5 (ขา)
VDD	ไฟเลี้ยง	**ขา 1 (3V3)**
VDDH	ไฟเลี้ยงแหล่ง IR	ต่อร่วมกับ VDD (บอร์ด breakout ต่อให้แล้ว)
GND	กราวด์	ขา 20 (GND)
SDA	ข้อมูล I2C	ขา 3 (GPIO2)
SCL	นาฬิกา I2C	ขา 5 (GPIO3)
DNC	ห้ามต่อ	—

พร้อม capacitor 100 nF คร่อม VDD–GND และ 100 µF ที่ปลายสายฝั่งเซนเซอร์

---

4. คำสั่ง I2C ที่ใช้

ทุกคำสั่งเป็นรหัส 16 บิต · ข้อมูลทุก word (2 ไบต์) ตามด้วย **CRC-8 หนึ่งไบต์** (polynomial `0x31`, init `0xFF`, ไม่ reflect, final XOR `0x00` — เหมือนกับ SGP30)

คำสั่ง	รหัส	ใช้ตอนไหน
`start_periodic_measurement`	`0x21B1`	เริ่มวัดต่อเนื่อง (อัปเดตทุก 5 วินาที)
`read_measurement`	`0xEC05`	อ่านค่า CO_2 + T + RH (9 ไบต์)
`stop_periodic_measurement`	`0x3F86`	หยุดก่อนส่งคำสั่งตั้งค่า
`set_temperature_offset`	`0x241D`	ชดเชยความร้อนจากบอร์ด KEY
`set_sensor_altitude`	`0x2427`	ตั้งความสูงจากระดับน้ำทะเล (เมตร)
`set_ambient_pressure`	`0xE000`	ตั้งความดันบรรยากาศ (Pa) — แม่นกว่า altitude
`perform_forced_recalibration`	`0x362F`	calibrate แบบบังคับ (FRC)
`set_automatic_self_calibration_enabled`	`0x2416`	เปิด/ปิด ASC
`get_data_ready_status`	`0xE4B8`	เช็คว่ามีค่าใหม่พร้อมหรือยัง
`get_serial_number`	`0x3682`	ตรวจว่าเซนเซอร์ตอบสนอง
`persist_settings`	`0x3615`	บันทึกค่าตั้งลง EEPROM ! ใช้ให้น้อยที่สุด
`reinit`	`0x3646`	โหลดค่าจาก EEPROM ใหม่

รูปแบบข้อมูลที่อ่านได้ (`read_measurement`)

ตอบกลับ 9 ไบต์:

```
[CO2_MSB][CO2_LSB][CRC]  [T_MSB][T_LSB][CRC]  [RH_MSB][RH_LSB][CRC]
```

สูตรแปลงค่า:

```
CO₂ (ppm)  = ค่าดิบ                              (ใช้ได้ตรง ๆ)
T  (°C)    = −45 + 175 × (ค่าดิบ / 2¹⁶)
RH (%)     = 100 × (ค่าดิบ / 2¹⁶)
```

ลำดับการทำงานที่ถูกต้อง

```
1. รอ 30 ms หลังจ่ายไฟ (power-up time)
2. stop_periodic_measurement      ← เผื่อค้างจากรอบก่อน · รอ 500 ms
3. get_serial_number              ← ยืนยันว่าเซนเซอร์ตอบสนอง
4. set_temperature_offset(ค่าจาก config)
5. set_sensor_altitude(ค่าจาก config)  หรือ set_ambient_pressure()
6. set_automatic_self_calibration_enabled(ค่าจาก config)
7. start_periodic_measurement
8. วนลูป: ทุก 5 วินาที → get_data_ready_status → read_measurement
```

> ! **ระหว่างที่ periodic measurement ทำงานอยู่ ห้ามส่งคำสั่งอื่น** ยกเว้น `read_measurement`, `get_data_ready_status`, `stop_periodic_measurement` และ `set_ambient_pressure` เท่านั้น

> ! **`persist_settings` เขียน EEPROM ซึ่งมีจำนวนรอบจำกัด** เรียกเฉพาะตอนตั้งค่าครั้งแรกหรือหลัง calibrate เท่านั้น **ห้ามเรียกทุกครั้งที่บูต**

---

5. ไลบรารี Python

```
pip install sensirion-i2c-scd4x sensirion-i2c-driver
# หรือ
pip install adafruit-circuitpython-scd4x
```

ไลบรารี	ข้อดี	ข้อเสีย
`sensirion-i2c-scd4x`	ทางการจากผู้ผลิต · ครบทุกคำสั่ง	เอกสารน้อยกว่า
`adafruit-circuitpython-scd4x`	ใช้ง่าย · ตัวอย่างเยอะ	ต้องพึ่ง Blinka

> **แนะนำ `sensirion-i2c-scd4x`** เพราะเป็นของผู้ผลิตโดยตรง จึงตามการเปลี่ยนแปลงของ datasheet ทัน และรองรับคำสั่ง FRC/ASC ครบถ้วนซึ่งจำเป็นสำหรับโปรเจกต์นี้

---

6. KEY Temperature offset — ขั้นตอนที่สำคัญที่สุด

SCD40 วัดอุณหภูมิที่ตัวชิปเอง ซึ่ง**ร้อนกว่าอากาศจริงเสมอ** เพราะ:

1. ความร้อนที่ตัวเซนเซอร์สร้างเอง (self-heating)
2. ความร้อนจาก SoC ของ Pi 5 ที่แผ่มาตามแผ่น PCB และอากาศ

ค่านี้ไม่กระทบแค่อุณหภูมิ — เซนเซอร์ใช้ค่าอุณหภูมิไปคำนวณความชื้นสัมพัทธ์ด้วย ดังนั้น offset ที่ผิดจะทำให้ **RH ผิดตามไปด้วย** และ datasheet ระบุว่า ค่าความแม่นยำที่โฆษณาไว้ทั้งหมดจะเป็นจริง**ก็ต่อเมื่อตั้ง temperature offset ถูกต้อง**

ค่าเริ่มต้นจากโรงงานคือ 4 °C ซึ่งเหมาะกับการติดตั้งบน PCB เล็ก ๆ ไม่ใช่กรณีนี้

วิธีหาค่า offset ที่ถูกต้อง

```
1. ประกอบระบบให้เสร็จสมบูรณ์ในกล่อง/ตำแหน่งจริงที่จะใช้งาน
2. เปิดระบบทิ้งไว้ให้อุณหภูมิเข้าสู่สมดุล  ≥ 30 นาที  (ไม่ใช่ 5 นาที)
3. วางเทอร์โมมิเตอร์อ้างอิงที่แม่นยำไว้ข้าง ๆ ตัวเซนเซอร์
4. offset ใหม่ = offset ปัจจุบัน + (T ที่ SCD40 อ่าน − T อ้างอิง)
5. set_temperature_offset(offset ใหม่)
6. persist_settings()       ← เรียกครั้งเดียวเท่านั้น
7. รออีก 30 นาที แล้วตรวจซ้ำ
```

ค่าที่พบบ่อยเมื่อวางใกล้ Pi 5: **6–10 °C** (สูงกว่าค่าเริ่มต้น 4 °C มาก)

> **วิธีที่ดีกว่าการปรับ offset สูง ๆ คือย้ายเซนเซอร์ให้ห่างจากแหล่งความร้อน** ต่อ SCD40 ผ่านสาย I2C ยาว 10–20 cm ออกไปนอกกล่อง แล้ว offset จะเหลือแค่ 1–3 °C ซึ่งแม่นยำกว่ามาก เพราะการชดเชยด้วยตัวเลขคงที่จะไม่แม่นเมื่อโหลด CPU เปลี่ยน

---

7. การชดเชยความดันบรรยากาศ

การวัด CO_2 แบบ NDIR ขึ้นกับความหนาแน่นของอากาศ จึงต้องบอกความสูงหรือความดันให้เซนเซอร์ทราบ

วิธี	คำสั่ง	ความแม่นยำ	หมายเหตุ
ตั้งความสูง (คงที่)	`set_sensor_altitude` (เมตร)	พอใช้	ตั้งครั้งเดียว · ต้อง persist
ตั้งความดัน (แบบไดนามิก)	`set_ambient_pressure` (Pa)	**แม่นกว่า**	ส่งใหม่ได้ตลอดขณะวัด · ไม่ต้อง persist

ค่าความสูงคร่าว ๆ ในไทย: กรุงเทพฯ ~2 m · เชียงใหม่ ~310 m · เชียงราย ~400 m · นครราชสีมา ~180 m

> ที่ระดับความสูงต่ำอย่างกรุงเทพฯ ผลกระทบแทบไม่มีนัยสำคัญ (< 1%) แต่ควรตั้งไว้เพื่อความถูกต้อง และ**จำเป็น**ถ้าติดตั้งบนภาคเหนือ

ทางเลือกขั้นสูง: โปรเจกต์นี้ดึงข้อมูลกรมอุตุฯ (TMD) อยู่แล้วผ่าน `lib/gov.js` สามารถส่งค่าความดันจริงเข้า `set_ambient_pressure` แบบไดนามิกได้ — แต่เป็นงาน optimization ที่ทำทีหลังได้ ไม่ต้องรีบ

---

8. การ calibrate CO_2 — ASC เทียบกับ FRC

รายละเอียดเต็มอยู่ใน `docs/04-calibration.md` — สรุปตรงนี้:

	ASC (Automatic Self-Calibration)	FRC (Forced Recalibration)
หลักการ	สมมติว่าค่าต่ำสุดที่เจอในรอบ ~7 วันคืออากาศบริสุทธิ์ (400 ppm)	บอกเซนเซอร์ตรง ๆ ว่าตอนนี้คือกี่ ppm
ต้องการอะไร	ต้องได้สัมผัสอากาศ ~400 ppm **สัปดาห์ละครั้ง**	ต้องพาไปที่อากาศบริสุทธิ์ครั้งเดียว
ข้อดี	ตั้งแล้วลืมได้	แม่นทันที ควบคุมได้
ข้อเสีย	**ห้องที่ปิดตลอดจะทำให้ค่าเพี้ยนสะสม**	ต้องทำเอง
ค่าเริ่มต้น	เปิดอยู่	—

! ประเด็นสำคัญสำหรับห้องปรับอากาศในไทย

ห้องแอร์ที่ปิดตลอดเวลามักไม่เคยลงถึง 400 ppm เลย (มักอยู่ที่ 600–1,200 ppm) ASC จะเข้าใจผิดว่าค่าต่ำสุดที่เจอ (เช่น 650 ppm) คือ 400 ppm แล้ว**ลบทุกค่าลงไป 250 ppm** ทำให้ค่าที่แสดงต่ำกว่าความจริงอย่างเป็นระบบ

แนวทางที่แนะนำ:

สถานการณ์	ทำอย่างไร
ห้องเปิดหน้าต่าง/ระบายอากาศทุกสัปดาห์	เปิด ASC ไว้ (ค่าเริ่มต้น)
ห้องแอร์ปิดตลอด	**ปิด ASC** แล้วทำ FRC ทุก 6–12 เดือน
ไม่แน่ใจ	ปิด ASC + ทำ FRC — ปลอดภัยกว่า

> datasheet ระบุว่าควรทำ FRC **หลังประกอบเสร็จอย่างน้อย 5 วัน** ไม่ใช่ทันที เพราะสารระเหยจากบัดกรีและกาวยังรบกวนค่าอยู่

---

9. เกณฑ์ค่า CO_2 ที่แนะนำสำหรับแดชบอร์ด

ระดับ	ช่วง (ppm)	ความหมาย
อากาศภายนอก	400–450	ค่าอ้างอิงอากาศบริสุทธิ์
ดี	< 800	การระบายอากาศเพียงพอ
ปานกลาง	800–1,000	เริ่มควรระบายอากาศ
แย่	1,000–1,500	เริ่มมีผลต่อสมาธิและความง่วง
แย่มาก	1,500–2,500	ปวดหัว เหนื่อย สมาธิลดลงชัดเจน
อันตราย	> 2,500	ต้องระบายอากาศทันที

> 1,000 ppm เป็นเกณฑ์ที่ใช้กันแพร่หลายในมาตรฐานคุณภาพอากาศภายในอาคาร และเหมาะกับโปรเจกต์นี้เป็นพิเศษ เพราะเชื่อมโยงโดยตรงกับ **Focus feature** — CO_2 สูงสัมพันธ์กับสมาธิที่ลดลงและความง่วง ซึ่งเป็นสิ่งที่ webcam EAR detection วัดอยู่แล้ว

ค่าเหล่านี้ต้องอยู่ใน `config/sensors.js → THRESHOLDS.co2` เท่านั้น

---

10. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
`i2cdetect` ไม่เจอ `0x62`	ไฟไม่ถึง · SDA/SCL สลับ · กราวด์ไม่ร่วม	ตรวจ 3.3V ที่ VDD · ตรวจสาย
อุณหภูมิสูงกว่าจริง 5–10 °C	temperature offset ผิด	ทำตาม §6
ความชื้นต่ำกว่าจริงตลอด	เป็นผลพวงจาก offset ผิด	แก้ offset แล้ว RH จะถูกตาม
CO_2 กระโดดเป็นฟันเลื่อย	ripple ราง 3V3 เกิน 30 mV	เพิ่ม bulk capacitor 100 µF
CO_2 ต่ำกว่าจริง ~200–300 ppm	ASC เพี้ยนเพราะห้องปิดตลอด	ปิด ASC + ทำ FRC
CO_2 ค้างที่ค่าเดิม	อ่านเร็วกว่ารอบ 5 วินาที	เช็ค `get_data_ready_status` ก่อนอ่าน
CO_2 กระโดดขึ้นเป็นพัน ๆ เป็นครั้งคราว	มีคนหายใจใส่เซนเซอร์โดยตรง	ย้ายตำแหน่งติดตั้ง
CRC ผิดบ่อย	บัส I2C ไม่นิ่ง	ลดความเร็วเป็น 50 kHz · สายสั้นลง
คำสั่งตั้งค่าไม่มีผล	ยังอยู่ใน periodic measurement mode	`stop_periodic_measurement` ก่อน แล้วรอ 500 ms

SGP30 — เซนเซอร์ TVOC / eCO_2

> **บทบาทในระบบนี้:** ตรวจจับสารอินทรีย์ระเหยง่าย (VOC) — กลิ่น น้ำยาทำความสะอาด สี ควัน **ไม่ใช่**เซนเซอร์ CO_2 แม้จะมีค่า "eCO_2" ให้ก็ตาม → ใช้ SCD40 สำหรับ CO_2 จริง

1. สเปกจาก datasheet (v0.9, สิงหาคม 2017)

สัญญาณคุณภาพอากาศ

รายการ	TVOC	eCO_2 (CO_2eq)
ช่วงค่าที่ส่งออก	0 – 60,000 ppb	400 – 60,000 ppm
ความละเอียด	1 ppb (0–2,008) · 6 ppb (2,008–11,110) · 32 ppb (11,110–60,000)	1 ppm (400–1,479) · 3 ppm (1,479–5,144) · 9 ppm (5,144–17,597) · 31 ppm (17,597–60,000)
อัตราการสุ่ม	**1 Hz**	**1 Hz**

สัญญาณดิบ (ที่มาของค่าข้างบน)

รายการ	ค่า
สัญญาณ Ethanol	0–1,000 ppm (ระบุสเปกที่ 0.3–30 ppm) · ความแม่นยำ ~15% ของค่าที่วัด
สัญญาณ H_2	0–1,000 ppm (ระบุสเปกที่ 0.5–10 ppm) · ความแม่นยำ ~10% ของค่าที่วัด
การ drift ระยะยาว	~1.3% ของค่าที่วัด ต่อปี

ไฟฟ้า

รายการ	ค่า
แรงดันไฟเลี้ยง (ตัวชิป)	**1.62 – 1.98V (ปกติ 1.8V)**
กระแสตอนวัด	48.2 mA
กระแสตอน sleep	2 µA (สูงสุด 10 µA)
เวลาบูตหลังจ่ายไฟ	0.4 ms (สูงสุด 0.6 ms)
ความถี่ SCL	0 – 400 kHz (รองรับ I2C fast mode)
ที่อยู่ I2C	`0x58` (ตายตัว เปลี่ยนไม่ได้)
ช่วงอุณหภูมิที่แนะนำ	5 – 55 °C
ความชื้นสัมบูรณ์ที่แนะนำ	4 – 20 g/m³
ขนาด	2.45 × 2.45 × 0.9 mm (DFN 6 ขา)

2. ⚠ คำเตือนเรื่องแรงดัน

ตัวชิป SGP30 ทำงานที่ 1.8V เท่านั้น — แรงดันสูงสุดที่ทนได้คือ 2.16V

หมายความว่า **การต่อชิปเปล่าเข้ากับราง 3.3V ของ Pi จะทำให้ชิปพังทันที**

> ✓ **ต้องซื้อเป็นบอร์ด breakout** ที่มี LDO regulator ลด 3.3V/5V → 1.8V พร้อม level shifter สำหรับบัส I2C มาให้เรียบร้อย

บอร์ดที่ใช้ได้และตรวจสอบแล้วว่ามีวงจรครบ:

บอร์ด	ไฟเข้า	หมายเหตุ
Adafruit SGP30 Breakout (#3709)	3.3V หรือ 5V	มี STEMMA QT/Qwiic · เอกสารดีที่สุด
Grove VOC and eCO2 Gas Sensor (Seeed)	3.3V หรือ 5V	ใช้สาย Grove
โมดูล SGP30 ทั่วไปจากจีน	มักระบุ 3.3–5V	⚠ **ต้องตรวจว่ามี LDO จริง**ก่อนต่อไฟ

วิธีตรวจโมดูลที่ไม่แน่ใจ: มองหาชิป 5–6 ขาขนาดเล็กใกล้ขา VIN (คือ LDO) ถ้าเห็นแค่ตัวชิป SGP30 กับตัวต้านทานสองสามตัว **อย่าเสี่ยง** — วัดแรงดันที่ขา VDD ของชิป ด้วยมัลติมิเตอร์ขณะจ่ายไฟ 3.3V เข้า VIN ต้องอ่านได้ ~1.8V

---

3. การต่อสาย

SGP30 breakout	→ Pi 5 (ขา)
VIN / VCC	ขา 2 (5V) หรือ ขา 1 (3V3) — บอร์ดจัดการเอง
GND	ขา 14 (GND)
SDA	ขา 3 (GPIO2)
SCL	ขา 5 (GPIO3)

พร้อม capacitor **100 nF** คร่อม VDD–GND ให้ใกล้ที่สุด (datasheet บังคับข้อนี้)

---

4. คำสั่ง I2C

คำสั่ง	รหัส	พารามิเตอร์ (ไบต์)	ตอบกลับ (ไบต์)	เวลาปกติ / สูงสุด (ms)
`Init_air_quality`	`0x2003`	—	—	2 / 10
`Measure_air_quality`	`0x2008`	—	6	10 / 12
`Get_baseline`	`0x2015`	—	6	10 / 10
`Set_baseline`	`0x201E`	6	—	10 / 10
`Set_humidity`	`0x2061`	3	—	1 / 10
`Measure_test`	`0x2032`	—	3	200 / 220
`Get_feature_set_version`	`0x202F`	—	3	1 / 2
`Measure_raw_signals`	`0x2050`	—	6	20 / 25
`Get_Serial_ID`	`0x3682`	—	9	—
Soft reset (General Call)	`0x0006`	—	—	0.4 / 0.6

CRC-8: polynomial `0x31` ($x^8+x^5+x^4+1$) · init `0xFF` · ไม่ reflect · final XOR `0x00` ตัวอย่าง: `CRC(0xBEEF) = 0x92`

รูปแบบข้อมูล `Measure_air_quality`

ตอบกลับ 6 ไบต์ **เรียงลำดับ eCO_2 ก่อน TVOC**:

```
[eCO2_MSB][eCO2_LSB][CRC]  [TVOC_MSB][TVOC_LSB][CRC]
```

ค่าใช้ได้ตรง ๆ ไม่ต้องแปลง (eCO_2 = ppm · TVOC = ppb)

> ⚠ **แต่ `Set_baseline` รับพารามิเตอร์เรียงกลับกัน คือ (TVOC, eCO_2)** ขณะที่ `Get_baseline` คืนค่าเรียง (eCO_2, TVOC) — datasheet ระบุไว้ชัดเจน **นี่คือจุดที่โค้ดมักผิดโดยไม่รู้ตัว** เพราะไม่มี error ให้เห็น แค่ค่า drift เพี้ยนไปเรื่อย ๆ

⚠ ห้ามใช้ `Measure_test` ในการทำงานปกติ

datasheet ระบุว่าคำสั่งนี้มีไว้สำหรับ**ทดสอบสายการผลิตเท่านั้น** และ **ห้ามเรียกหลังจากส่ง `Init_air_quality` ไปแล้ว** เพราะจะทำให้เซนเซอร์กลับไป sleep mode และ baseline ที่สะสมมาหายไป

---

5. ⚠ ข้อบังคับ: ต้องอ่านทุก 1 วินาทีเป๊ะ ๆ

datasheet ระบุไว้ชัด:

> *"After the 'Init_air_quality' command, a 'Measure_air_quality' command has to be sent in regular intervals of 1s to ensure proper operation of the dynamic baseline compensation algorithm."*

อัลกอริทึม baseline ภายในชิปถูกปรับจูนมาสำหรับ **1 Hz เท่านั้น** ถ้ารอบไม่สม่ำเสมอ:

- baseline จะ drift ผิดทิศ
- ค่า TVOC จะสูงหรือต่ำเกินจริงอย่างเป็นระบบ

- **ไม่มี error หรือสัญญาณเตือนใด ๆ ให้เห็น** — ค่ายังคงออกมาดูสมเหตุสมผล

การเขียนโค้ดที่ถูกต้อง:

- ใช้ thread แยกที่วนลูปด้วยตัวจับเวลาแบบ absolute (คำนวณเวลาถัดไปจากเวลาเริ่ม) **ไม่ใช่** `sleep(1)` ต่อท้ายงาน เพราะเวลาที่ใช้อ่านค่าจะสะสมเป็น drift
- อย่าเอา SGP30 ไปวนลูปรวมกับเซนเซอร์อื่นที่มีรอบต่างกัน
- ถ้าอ่านพลาดหนึ่งรอบ ให้ข้ามไปรอบถัดไปตามตาราง **ไม่ใช่**อ่านทันทีเพื่อชดเชย

```
# แนวคิด — ค่าคงที่ต้องมาจาก config
next_at = time.monotonic()
while running:
    next_at += config["sgp30"]["interval_s"]   # ไม่สะสม drift
    read_and_store()
    time.sleep(max(0, next_at - time.monotonic()))
```

---

6. Baseline — สิ่งที่ต้องทำให้ถูก ไม่งั้นค่าไร้ความหมาย

SGP30 ใช้ **dynamic baseline compensation** คือเรียนรู้ว่า "อากาศปกติของที่นี่" เป็นอย่างไร แล้วรายงานส่วนที่เบี่ยงเบนจากนั้น

ช่วงเริ่มต้น

ช่วงเวลา	สิ่งที่เกิดขึ้น
0–15 วินาที หลัง `Init_air_quality`	คืนค่าคงที่ **400 ppm / 0 ppb — ต้องทิ้ง อย่าส่งขึ้นแดชบอร์ด**
15 วินาที – 12 ชั่วโมง (ครั้งแรกที่ติดตั้ง)	baseline ยังสร้างไม่เสร็จ ค่าเชื่อถือได้บางส่วน
หลัง 12 ชั่วโมง	baseline ใช้ได้ — **เริ่มบันทึกลงไฟล์ได้**

วงจรบันทึกและกู้คืน baseline

ขั้น	ทำอะไร	เงื่อนไข / หมายเหตุ
1	เริ่มระบบ	—
2	`Init_air_quality`	ต้องเรียกทุกครั้งหลังจ่ายไฟหรือ soft reset
3	ตรวจไฟล์ `baseline.json`	อายุน้อยกว่า 7 วันหรือไม่
3ก	**มี →** `Set_baseline(tvoc, eco2)`	! **สลับลำดับ!** ไม่ใช่ `(eco2, tvoc)`
3ข	**ไม่มี →** ข้ามไป	ปล่อยให้เรียนรู้ baseline ใหม่ 12 ชั่วโมง
4	ทิ้งค่า 15 วินาทีแรก	เซนเซอร์คืนค่าคงที่ 400 ppm / 0 ppb
5	วนลูปอ่านทุก 1.000 วินาที	ต้องเป๊ะ — ดูหัวข้อ 5
6	ทุก 1 ชั่วโมง: `Get_baseline()` → เขียนไฟล์	เริ่มบันทึกหลังผ่าน 12 ชั่วโมงแรกแล้วเท่านั้น

กฎที่ต้องยึด:

กฎ	เหตุผล
บันทึกทุก **1 ชั่วโมง** ไม่ใช่ทุกนาที	ถนอม SD card
baseline ที่เก่ากว่า **7 วัน ให้ทิ้ง**	datasheet แนะนำ — ค่าเก่าเกินไปแย่กว่าไม่มีเลย
เก็บ path ของไฟล์ไว้ใน config	ไม่ hardcode
อย่าบันทึกก่อนครบ 12 ชั่วโมงแรก	จะบันทึก baseline ที่ยังไม่สมบูรณ์
เขียนไฟล์แบบ atomic (เขียนไฟล์ชั่วคราวแล้ว rename)	กันไฟล์เสียตอนไฟดับ

7. Humidity compensation

SGP30 มีวงจรชดเชยความชื้นในตัว แต่ต้อง**ป้อนค่าความชื้นสัมบูรณ์จากภายนอกให้** ซึ่งในระบบนี้เอามาจาก SCD40 ได้เลย

สูตรแปลง RH + T → ความชื้นสัมบูรณ์ (g/m³)

```
AH = 216.7 × [ (RH/100) × 6.112 × exp(17.62 × T / (243.12 + T)) ] / (273.15 + T)
```

โดย `T` = °C และ `RH` = %

รูปแบบข้อมูลที่ส่งให้ `Set_humidity`

เป็น **fixed-point 8.8 bit** (8 บิตบนคือจำนวนเต็ม · 8 บิตล่างคือเศษส่วน):

```
ค่าที่ส่ง = round(AH × 256)
```

ตัวอย่าง	ค่า hex	ความหมาย
16.50 g/m³	`0x0F80`	16 + 128/256
11.57 g/m³	`0x0B92`	ค่าเริ่มต้นจากโรงงาน
ต่ำสุด	`0x0001`	1/256 g/m³
ปิดการชดเชย	`0x0000`	0 g/m³

ความถี่ในการอัปเดต: ทุก 1–5 นาทีก็เพียงพอ ความชื้นในห้องไม่เปลี่ยนเร็วขนาดนั้น ค่าที่ตั้งจะถูกใช้ต่อเนื่องจนกว่าจะส่งค่าใหม่

> ! ค่าจะรีเซ็ตกลับเป็น `0x0B92` ทุกครั้งที่จ่ายไฟใหม่หรือ soft reset **ต้องส่ง `Set_humidity` ใหม่ทุกครั้งหลังบูต**

ลำดับที่ถูกต้อง: อ่าน RH+T จาก SCD40 → คำนวณ AH → `Set_humidity` → แล้วค่อยอ่าน `Measure_air_quality`

8. ! สารที่ทำให้เซนเซอร์เสื่อมถาวร

SGP30 เป็นเซนเซอร์แบบ metal-oxide ซึ่ง**เสียหายถาวร**เมื่อสัมผัสสารบางชนิด — ไม่สามารถซ่อมหรือ calibrate กลับได้

กลุ่ม	ตัวอย่าง
ซิลิโคน ! อันตรายที่สุด	กาวซิลิโคน · ซิลิโคนหล่อลื่น · sealant · ยางซิลิโคนบางชนิด
ตัวทำละลาย	ทินเนอร์ · อะซิโตน · น้ำยาล้างเล็บ · แอลกอฮอล์เข้มข้น
น้ำหอม	สเปรย์ปรับอากาศ · น้ำหอมปรับผ้านุ่ม · เทียนหอม · น้ำมันหอมระเหย
สารเคมีอื่น	ฟอร์มาลดีไฮด์เข้มข้น · ไอกรด

ข้อปฏิบัติ:

- ประกอบด้วยสกรูและ standoff **ห้ามใช้กาวซิลิโคนหรือ hot glue ใกล้เซนเซอร์**
- เก็บในถุงกันไฟฟ้าสถิตที่ปิดสนิทก่อนติดตั้ง
- ห้ามล้างบอร์ดด้วยตัวทำละลายหลังบัดกรี
- ถ้าติดตั้งในกล่อง ให้ระบายอากาศได้ — อย่าใช้กล่องที่ปล่อยไอพลาสติก

9. ไลบรารี Python

```
pip install adafruit-circuitpython-sgp30
```

หรือเขียนเองด้วย `smbus2` ซึ่งไม่ยากเพราะคำสั่งมีไม่กี่ตัว — ข้อดีคือควบคุมจังหวะ 1 Hz และการจัดการ baseline ได้เต็มที่ ซึ่งเป็นสองเรื่องที่สำคัญที่สุดของเซนเซอร์ตัวนี้

> Sensirion ไม่มีไลบรารี Python ทางการสำหรับ SGP30 (มีสำหรับ SGP40/41 รุ่นใหม่กว่า) ไลบรารีของ Adafruit จึงเป็นตัวเลือกที่ดูแลดีที่สุดที่มีอยู่

10. เกณฑ์ค่า TVOC ที่แนะนำ

อิงตามแนวทางของหน่วยงานสิ่งแวดล้อมเยอรมนี (UBA) ซึ่งเป็นเกณฑ์ TVOC ที่อ้างอิงกันแพร่หลายที่สุด

ระดับ	TVOC (ppb)	ความหมาย	คำแนะนำ
ดีเยี่ยม	0–65	อากาศสะอาด	—
ดี	65–220	ปกติ	—
ปานกลาง	220–660	เริ่มมีกลิ่น/สารระเหย	ระบายอากาศ
แย่	660–2,200	ไม่ควรอยู่นาน	ระบายอากาศทันที
แย่มาก	> 2,200	ไม่เหมาะกับการอยู่อาศัย	ออกจากพื้นที่ · หาต้นตอ

> **eCO_2 ไม่ควรใช้ตั้งเกณฑ์เตือนเลย** เพราะเป็นค่าที่คำนวณจากสัญญาณ VOC ไม่ได้วัด CO_2 ในระบบนี้เรามี SCD40 ที่วัด CO_2 จริงอยู่แล้ว ถ้าจะแสดง eCO_2 บนแดชบอร์ด ควรกำกับให้ชัดว่าเป็นค่าประมาณจาก VOC

11. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
ค่าค้างที่ 400 ppm / 0 ppb ตลอด	ยังอยู่ในช่วง init 15 วินาที · หรือลืมส่ง `Init_air_quality`	รอให้ครบ · ตรวจว่าส่ง init แล้ว
TVOC สูงผิดปกติเสมอ	baseline ยังไม่สร้างเสร็จ · หรือโดนสารทำลาย	รอ 12 ชั่วโมง · ถ้ายังไม่ดีขึ้นแปลว่าเซนเซอร์เสีย
TVOC เป็น 0 ตลอดทั้งวัน	baseline ปรับตัวตามอากาศที่นิ่งเกินไป	เป็นพฤติกรรมปกติ · ค่าจะขึ้นเมื่อมีการเปลี่ยนแปลงจริง
ค่า drift เรื่อย ๆ วันต่อวัน	รอบการอ่านไม่ใช่ 1 Hz เป๊ะ	ตรวจ timing ของ thread ให้ใช้ absolute timer
Set_baseline แล้วค่าแย่ลง	สลับลำดับ (TVOC, eCO_2) ผิด	ตรวจลำดับพารามิเตอร์ให้เป็น TVOC ก่อน
`i2cdetect` ไม่เจอ `0x58`	โมดูลไม่มี LDO และชิปพังไปแล้ว	วัดแรงดันที่ขา VDD ของชิป ต้องได้ 1.8V
ค่าเพี้ยนหลังย้ายห้อง	baseline ผูกกับสภาพแวดล้อมเดิม	ลบ baseline.json แล้วให้เรียนรู้ใหม่ 12 ชั่วโมง
ค่าไม่ตอบสนองต่อกลิ่นเลย	เซนเซอร์เสื่อมจากซิลิโคน/ตัวทำละลาย	เปลี่ยนตัวใหม่ — กู้ไม่ได้

BH1750 — เซนเซอร์วัดความสว่าง (lux)

> **ทำไมถึงเลือกตัวนี้แทน LDR:** BH1750 ส่งค่าเป็น **lux โดยตรง** ไม่ต้องแปลงจากแรงดัน ไม่ต้องใช้ ADC (ซึ่ง Pi 5 ไม่มี) และการตอบสนองต่อสเปกตรัมใกล้เคียงสายตามนุษย์ LDR ให้แค่ความต้านทานที่ไม่เป็นเชิงเส้น ต้องปรับเทียบเอง และเพี้ยนตามอุณหภูมิ

1. สเปกจาก datasheet

รายการ	ค่า
ชื่อชิปเต็ม	BH1750FVI
ช่วงวัด	**1 – 65,535 lx**
ความแม่นยำ	±20% ทั่วไป (ค่าสัมประสิทธิ์การวัด 0.96 – 1.44)
การตอบสนองสเปกตรัม	ใกล้เคียงการมองเห็นของมนุษย์ (peak ~560 nm)
อิทธิพลจากแสงอินฟราเรด	ต่ำมาก
แรงดันไฟเลี้ยง (ตัวชิป)	2.4 – 3.6V
แรงดันไฟเลี้ยง (โมดูล GY-302)	3.3 – 5V (มี regulator บนบอร์ด)
กระแสตอนทำงาน	~120 µA ทั่วไป
กระแสตอน power-down	~0.01 µA
แรงดันสูงสุดของขา ADDR	**3.6V** ⚠
ที่อยู่ I2C	`0x23` (ค่าเริ่มต้น) หรือ `0x5C`
ความถี่ SCL	สูงสุด 400 kHz
อุณหภูมิใช้งาน	−40 ถึง +85 °C

2. ที่อยู่ I2C

สถานะขา ADDR	ที่อยู่
ต่อลง GND หรือ **ปล่อยลอย**	`0x23` ← ใช้ตัวนี้
ดึงขึ้น HIGH (> 0.7 × VCC)	`0x5C`

โมดูล GY-302 ส่วนใหญ่มี pull-down บนขา ADDR อยู่แล้ว **ปล่อยลอยได้เลย ไม่ต้องต่ออะไร**

> ⚠ ถ้าจะดึง ADDR ขึ้น HIGH ต้องใช้ **3.3V ไม่ใช่ 5V** เพราะขานี้ทนได้สูงสุด 3.6V

3. การต่อสาย

BH1750 / GY-302	→ Pi 5 (ขา)
VCC	ขา 1 (3V3)
GND	ขา 14 (GND)
SDA	ขา 3 (GPIO2)
SCL	ขา 5 (GPIO3)
ADDR	ปล่อยลอย

> แม้โมดูลจะรับ 5V ได้ แต่**ให้ใช้ 3.3V** เพื่อให้ระดับลอจิกบนบัสตรงกับอุปกรณ์ตัวอื่นทั้งหมด และลดโอกาสที่ level shifter บนบอร์ดจะทำงานผิดพลาด

---

4. คำสั่งและโหมดการวัด

คำสั่ง	รหัส	ความละเอียด	เวลาวัด
Power Down	`0x00`	—	—
Power On	`0x01`	—	—
Reset	`0x07`	—	—
Continuous High Res Mode	`0x10`	**1 lx**	**120 ms** ← แนะนำ
Continuous High Res Mode 2	`0x11`	0.5 lx	120 ms
Continuous Low Res Mode	`0x13`	4 lx	16 ms
One Time High Res Mode	`0x20`	1 lx	120 ms
One Time High Res Mode 2	`0x21`	0.5 lx	120 ms
One Time Low Res Mode	`0x23`	4 lx	16 ms

โหมดที่แนะนำ: Continuous High Resolution Mode (`0x10`)

เหตุผล	รายละเอียด
ความละเอียด 1 lx เพียงพอ	ห้องทำงานทั่วไปอยู่ที่ 300–750 lx — ความละเอียด 0.5 lx ไม่ได้ให้ประโยชน์เพิ่ม
ไม่ต้องส่งคำสั่งซ้ำ	ต่างจาก One Time mode ที่เซนเซอร์กลับไป power-down หลังวัดเสร็จ
กระแสที่ใช้ต่างกันน้อยมาก	120 µA เทียบกับ ~0 µA — ไม่คุ้มกับความซับซ้อนที่เพิ่มขึ้น

> **High Res Mode 2 (`0x11`) มีประโยชน์เฉพาะกรณีวัดแสงสลัวมาก** (< 10 lx) เช่นตรวจว่าไฟในห้องปิดหรือยัง ถ้าโปรเจกต์ต้องการแยกแยะระดับแสงกลางคืนละเอียด ๆ ค่อยพิจารณาสลับโหมด

การอ่านค่า

เซนเซอร์คืน **2 ไบต์ big-endian** แปลงด้วย:

```
lux = ค่าดิบ / 1.2
```

ตัวหาร 1.2 คือค่าคงที่จาก datasheet (measurement accuracy typ.) **ไม่ใช่ค่าที่ตั้งเอง** สำหรับ High Res Mode 2 ต้องหารเพิ่มอีก 2 (รวมเป็น `/2.4`)

ลำดับที่ถูกต้อง:

```
1. เขียน 0x01 (Power On)
2. เขียน 0x10 (Continuous High Res Mode)
3. รอ ≥ 180 ms  ← เผื่อจากเวลาวัด 120 ms
4. วนลูป: อ่าน 2 ไบต์ → หาร 1.2 → ได้ lux
```

---

5. ไลบรารี Python

```
pip install smbus2
```

โปรโตคอลง่ายมาก (เขียนคำสั่ง 1 ไบต์ · อ่าน 2 ไบต์) **ไม่ต้องใช้ไลบรารีเฉพาะทาง** เขียนเองด้วย `smbus2` ราว 20 บรรทัดก็ครบ และควบคุมโหมดกับ timing ได้ชัดเจนกว่า

ถ้าต้องการไลบรารีสำเร็จรูป: `pip install adafruit-circuitpython-bh1750`

---

6. การ calibrate

BH1750 ปรับเทียบมาจากโรงงานแล้ว **สิ่งที่ต้อง calibrate คือการชดเชยฝาครอบ ไม่ใช่ตัวเซนเซอร์**

ปัจจัยการส่งผ่านแสงของฝาครอบ (transmission factor)

ถ้าติดตั้งหลังกระจก อะคริลิก หรือฝาครอบใด ๆ แสงจะถูกลดทอนลง:

วัสดุ	การส่งผ่านโดยประมาณ
ไม่มีฝาครอบ	1.00
อะคริลิกใส 2–3 mm	0.90 – 0.92
กระจกใส	0.88 – 0.92
อะคริลิกฝ้า	0.40 – 0.70
ฟิล์มกรองแสง	0.20 – 0.50

วิธีหาค่าจริง:

```
1. วัด lux โดยไม่มีฝาครอบ ในแสงคงที่          → L_open
2. ใส่ฝาครอบ วัดซ้ำในแสงเดิม                  → L_covered
3. transmission = L_covered / L_open
4. lux ที่ถูกต้อง = ค่าที่อ่านได้ / transmission
```

หรือถ้ามีเครื่องวัดแสงอ้างอิง (หรือแอปวัดแสงบนมือถือที่แม่นพอสมควร):

```
factor = lux อ้างอิง / lux ที่ BH1750 อ่านได้
```

> ค่า `transmission` หรือ `factor` นี้ **ต้องอยู่ในไฟล์ config** ไม่ใช่คูณตรง ๆ ในโค้ด เช่น `BH1750_TRANSMISSION_FACTOR=0.91`

สิ่งที่ต้องระวังตอนวัด

- อย่าให้เงาของสาย ฝาครอบ หรือชิ้นส่วนอื่นทับหน้าเซนเซอร์
- อย่าวางหันหน้าเข้าหาหลอดไฟโดยตรง — จะได้ค่าที่ไม่ได้สะท้อนแสงในห้องจริง
- ถ้าใช้หลอด LED ที่มี PWM dimming ค่าอาจกระพริบ → ใช้ค่าเฉลี่ยเคลื่อนที่ 3–5 ตัวอย่าง

---

7. เกณฑ์ค่าความสว่างที่แนะนำ

อ้างอิงแนวทางความสว่างสำหรับพื้นที่ทำงาน (มาตรฐานสากลด้านการส่องสว่างในสำนักงานอยู่ที่ 300–500 lx)

ระดับ	lux	สภาพ	เหมาะกับ
มืด	< 50	มืดมาก	นอนหลับ
สลัว	50–150	ไฟห้องนอนตอนกลางคืน	พักผ่อน · ดูหนัง
พอดี	**300–500**	**ไฟห้องทำงาน/ห้องเรียน**	**อ่านหนังสือ · ทำงาน** ✅
สว่าง	500–1,000	ห้องที่แสงธรรมชาติเข้าเยอะ	งานที่ต้องการรายละเอียด
สว่างมาก	> 1,000	ใกล้หน้าต่างตอนกลางวัน	อาจแสบตาถ้าจ้องจอ

ค่าอ้างอิงเพิ่มเติม: แสงเทียน ~10 lx · ห้องนั่งเล่นตอนกลางคืน ~150 lx · วันมีเมฆกลางแจ้ง ~1,000–10,000 lx · แดดจัดกลางแจ้ง ~100,000 lx (**เกินช่วงวัดของ BH1750**)

> `TIP` **เชื่อมโยงกับ Focus feature:** แสงต่ำกว่า 300 lx สัมพันธ์กับความล้าของสายตาและความง่วง ซึ่งเป็นสิ่งที่ EAR detection จากกล้องวัดอยู่แล้ว การมีทั้งสองค่าทำให้แดชบอร์ดบอกได้ว่า "ง่วงเพราะแสงน้อย" ไม่ใช่แค่ "ง่วง"

---

8. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
`i2cdetect` ไม่เจอ `0x23`	ไฟไม่ถึง · ADDR ถูกดึง HIGH โดยไม่ตั้งใจ	ลองสแกนหา `0x5C` ด้วย · วัดแรงดัน VCC
อ่านได้ `0` ตลอด	ยังไม่ได้ส่งคำสั่ง Power On + เลือกโหมด	ส่ง `0x01` แล้วตามด้วย `0x10`
อ่านได้ `65535` ตลอด	ค่าล้นช่วงวัด (แสงจ้าเกิน)	ย้ายออกจากแสงแดดตรง
ค่าเปลี่ยนกระตุกตลอดเวลา	หลอด LED มี PWM dimming	ใช้ค่าเฉลี่ยเคลื่อนที่ 3–5 ตัวอย่าง
ค่าต่ำกว่าที่ควรเป็นเสมอ	ฝาครอบลดทอนแสง · หน้าเซนเซอร์สกปรก	ตั้ง transmission factor · เช็ดหน้าเซนเซอร์
ค่าอ่านได้ช้า/ค้าง	อ่านถี่กว่าเวลาวัด 120 ms	เว้นช่วงอย่างน้อย 180 ms ระหว่างการอ่าน

ไมโครโฟน USB — วัดระดับเสียง (dBA)

> **ยืนยันแล้ว: ใช้ไมค์ USB ร่วมกับ BH1750 ได้ ไม่มีปัญหาใด ๆ** ทั้งสองใช้บัสคนละเส้นโดยสิ้นเชิง — ไมค์อยู่บน USB, BH1750 อยู่บน I2C ไม่แย่งขา GPIO ไม่แย่งที่อยู่ I2C และไม่แย่ง bandwidth กัน

1. ทำไมเลือก USB แทน I2S หรือ analog

ทางเลือก	ข้อดี	ข้อเสีย	สรุป
USB (UAC)	เสียบใช้ได้ทันที · ไม่กิน GPIO · ALSA รองรับในตัว · หาซื้อง่าย	กินโควตากระแส USB · latency สูงกว่าเล็กน้อย (ไม่สำคัญกับงานนี้)	✅ **เลือกตัวนี้**
I2S (เช่น INMP441)	คุณภาพดี · กินไฟน้อย · ไม่กินโควตา USB	ต้องใช้ GPIO 3 ขา · ต้องตั้ง device tree overlay · ต่อสายยุ่งกว่า	ทางเลือกสำรอง
Analog (เช่น MAX9814)	ถูกที่สุด	**Pi 5 ไม่มี ADC** ต้องเพิ่ม ADS1115 · ADS1115 เร็วสุด 860 SPS ซึ่ง**ช้าเกินไปสำหรับเสียง**	❌ ไม่แนะนำ

> **เหตุผลหลักที่ตัด analog ออก:** การวัด dB ที่ถูกต้องต้องสุ่มสัญญาณเสียงที่ ≥8 kHz ADS1115 ทำได้แค่ 860 ตัวอย่างต่อวินาที ซึ่งวัดได้แค่ "แรงดันเฉลี่ยของ envelope" ไม่ใช่ระดับเสียงจริง ผลคือค่าที่ได้จะบอกได้แค่ "ดัง/เบา" คร่าว ๆ ไม่ใช่ dBA ที่มีความหมาย

2. การเลือกไมโครโฟน

ข้อกำหนดขั้นต่ำ:

รายการ	ค่าที่ต้องการ
มาตรฐาน	**USB Audio Class 1.0 หรือ 2.0 (UAC)** — ต้องไม่ต้องลงไดรเวอร์
Sample rate	รองรับ 44.1 kHz หรือ 48 kHz
ความละเอียด	16 บิตขึ้นไป
กระแส	≤ 100 mA
ทิศทางรับเสียง	Omnidirectional (รอบทิศ) — เพราะวัดเสียงห้อง ไม่ใช่บันทึกเสียงพูด

ประเภทที่แนะนำ:

แบบ	เหมาะกับ	หมายเหตุ
Mini USB mic (แบบเสียบตรง)	✓ แนะนำสำหรับโปรเจกต์นี้	ราคาถูก · เล็ก · omnidirectional · เพียงพอสำหรับวัดระดับเสียง
USB conference mic	ห้องใหญ่	แพงกว่าโดยไม่จำเป็น
USB sound card + ไมค์ 3.5mm	ต้องการเลือกไมค์เอง	ยืดหยุ่นแต่ยุ่งกว่า
USB condenser mic (ขาตั้ง)	✗ ไม่แนะนำ	ออกแบบมาสำหรับบันทึกเสียง มักมี AGC ที่ทำให้วัดระดับไม่ได้

! AGC — ศัตรูตัวจริงของการวัดระดับเสียง

AGC (Automatic Gain Control) ปรับความดังขึ้นเองอัตโนมัติเมื่อเสียงเบา ซึ่งทำให้ค่าที่วัดได้**ไร้ความหมายโดยสิ้นเชิง** — ห้องเงียบก็จะถูกขยายจนดูเหมือนดัง

ไมค์สำหรับประชุม/สตรีมมักเปิด AGC มาเป็นค่าเริ่มต้น **ต้องปิดให้ได้**

```
# ตรวจว่ามี control อะไรบ้าง
amixer -c 1 scontrols

# ปิด AGC / auto gain ถ้ามี
amixer -c 1 sset 'Auto Gain Control' off

# ตั้ง capture gain ให้คงที่แล้วจดค่าไว้ (ห้ามเปลี่ยนอีกหลัง calibrate)
amixer -c 1 sset 'Mic' 50%
```

> KEY **หลัง calibrate เสร็จแล้ว ห้ามเปลี่ยนค่า gain อีกเด็ดขาด** — ค่า dB offset ที่หาไว้จะใช้ไม่ได้ทันที ค่า gain ที่ตั้งไว้ควรบันทึกไว้ในไฟล์ config และตั้งกลับอัตโนมัติทุกครั้งที่บูต

---

3. การตั้งค่าบน Raspberry Pi OS

```
# ตรวจว่าระบบเห็นไมค์
arecord -l
# ตัวอย่างผลลัพธ์:
#   card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]

lsusb                      # ยืนยันว่าเป็น UAC device

# ทดสอบอัดเสียง 5 วินาที
arecord -D plughw:1,0 -f S16_LE -r 48000 -c 1 -d 5 /tmp/test.wav
aplay /tmp/test.wav        # ถ้ามีลำโพงต่ออยู่
```

สิ่งที่ต้องบันทึกไว้ใน config:

ค่า	ตัวอย่าง	หมายเหตุ
ชื่ออุปกรณ์	`plughw:1,0`	! หมายเลข card **เปลี่ยนได้**เมื่อเสียบ USB ต่างพอร์ต
Sample rate	`48000`	
จำนวนช่อง	`1` (mono)	
รูปแบบ	`S16_LE`	
ขนาดหน้าต่าง RMS	`1.0` วินาที	

KEY ทำให้ชื่ออุปกรณ์คงที่

หมายเลข card ของ ALSA เปลี่ยนได้ตามลำดับการ enumerate ของ USB ทำให้ `plughw:1,0` กลายเป็น `plughw:2,0` หลังรีบูตหรือเสียบ webcam เพิ่ม (เพราะ webcam ก็มีอุปกรณ์เสียงในตัว)

วิธีแก้ที่ถูกต้อง — ผูกชื่อกับ USB ID:

```
# หา vendor:product id
lsusb        # เช่น Bus 001 Device 005: ID 0d8c:0134 C-Media ...
```

สร้าง `/etc/udev/rules.d/85-env-monitor-mic.rules`:

```
SUBSYSTEM=="sound", ATTRS{idVendor}=="0d8c", ATTRS{idProduct}=="0134", ATTR{id}="envmic"
```

จากนั้นอ้างถึงด้วย `plughw:CARD=envmic,DEV=0` ซึ่งจะคงที่เสมอ

> วิธีนี้จำเป็นมากถ้าใช้ USB webcam ร่วมด้วย เพราะจะมีการ์ดเสียงสองใบในระบบ

---

4. การคำนวณ dB

สูตรพื้นฐาน

```
dB = 20 × log₁₀(RMS / REF) + OFFSET
```

โดย:

- **RMS** = รากที่สองของค่าเฉลี่ยกำลังสองของตัวอย่างเสียงในหน้าต่าง 1 วินาที
- **REF** = ค่าอ้างอิง full-scale (สำหรับ 16 บิต = 32768)
- **OFFSET** = ค่าคงที่จากการ calibrate เทียบกับเครื่องวัดจริง (**ต้องอยู่ใน** config)

ขั้นตอนการคำนวณ

```
1. อ่านบล็อกเสียงจาก ALSA  (48000 ตัวอย่าง = 1 วินาที)
2. แปลงเป็น float แล้วหารด้วย 32768   → ช่วง -1.0 ถึง 1.0
3. ลบค่า DC offset ออก  (ลบค่าเฉลี่ยของบล็อก)
4. (ทางเลือก) กรอง A-weighting
5. RMS = sqrt(mean(x²))
6. dB = 20 × log10(max(RMS, ค่าเล็กสุดกันหาร 0)) + OFFSET
7. จำกัดค่าให้อยู่ในช่วงที่สมเหตุสมผล (เช่น 25–120 dB)
```

สิ่งที่ต้องระวังในโค้ด:

ประเด็น	ทำอย่างไร
RMS = 0 (ห้องเงียบสนิท)	ใช้ `max(rms, 1e-10)` กัน `log10(0)` = −∞
DC offset	ไมค์ราคาถูกมักมี DC bias — ต้องลบค่าเฉลี่ยออกก่อนคำนวณ RMS
Buffer overrun	ถ้า thread อื่นทำงานหนัก ALSA อาจ drop ข้อมูล — จับ error แล้ว restart stream
ค่าคงที่ทั้งหมด	`REF` , `OFFSET` , ช่วง clamp, ขนาดหน้าต่าง — **ทุกตัวต้องมาจาก config**

A-weighting — จำเป็นแค่ไหน

A-weighting คือฟิลเตอร์ที่ลดน้ำหนักความถี่ต่ำและสูงมาก ให้ใกล้เคียงกับการได้ยินของมนุษย์ เป็นสิ่งที่ทำให้ตัวเลขเรียกว่า "dBA" ได้อย่างถูกต้อง

แนวทาง	ความแม่นยำ	ความซับซ้อน	เหมาะกับ
ไม่กรองเลย (dBFS + offset)	พอใช้ — เปรียบเทียบกันเองได้	ต่ำมาก	✅ **เริ่มด้วยแบบนี้**
A-weighting ด้วย IIR filter	ดี	ปานกลาง — `scipy.signal`	ถ้าต้องการเรียกว่า dBA จริง ๆ
ใช้เครื่องวัด SPL แยกต่างหาก	แม่นที่สุด	สูง (ค่าอุปกรณ์)	งานที่ต้องอ้างอิงมาตรฐาน

> **คำแนะนำ:** เริ่มจากแบบไม่กรองก่อน ทำให้ระบบทำงานได้ครบวงจร แล้วค่อยเพิ่ม A-weighting ทีหลัง ถ้ายังไม่ได้กรอง A-weighting **ควรกำกับบนแดชบอร์ดว่าเป็น "ระดับเสียงสัมพัทธ์" ไม่ใช่ "dBA"** เพื่อไม่ให้เข้าใจผิดว่าเป็นค่าที่เทียบกับมาตรฐานได้

---

5. การ calibrate

รายละเอียดเต็มอยู่ใน `docs/04-calibration.md` — สรุปตรงนี้:

```
1. ติดตั้งไมค์ในตำแหน่งจริง ตั้ง gain แล้วจดค่าไว้
2. หาเครื่องอ้างอิง – เครื่องวัด SPL หรือแอปวัดเสียงบนมือถือ (แม่นราว ±3 dB)
3. วางเครื่องอ้างอิงข้างไมค์ ห่างไม่เกิน 10 cm
4. สร้างเสียงคงที่ระดับปานกลาง (เช่น เปิดเพลง หรือ white noise)
5. อ่านค่าพร้อมกันทั้งสองตัว
6. OFFSET = dB จากเครื่องอ้างอิง – dB ที่คำนวณได้จากไมค์
7. ตรวจซ้ำที่ระดับเสียงอื่นอีก 2 ระดับ (เงียบ / ดัง)
8. บันทึก OFFSET ลงไฟล์ config
```

ข้อจำกัดที่ต้องยอมรับ: ไมค์ USB ราคาถูกไม่ใช่เครื่องมือวัดมาตรฐาน

- การตอบสนองความถี่ไม่แบน (โดยเฉพาะความถี่ต่ำ)
- ช่วงที่เป็นเชิงเส้นจำกัด — ค่าจะเพี้ยนที่ระดับเงียบมากหรือดังมาก
- ความแม่นยำที่คาดหวังตามจริง: **±5 dB** ในช่วงกลาง (40–85 dB)

> ✅ **ค่านี้เหมาะกับการดูแนวโน้มและเปรียบเทียบกับตัวเอง** — "วันนี้เสียงดังกว่าเมื่อวาน" ใช้ได้ ❌ **ไม่เหมาะกับการอ้างอิงทางกฎหมายหรือมาตรฐานความปลอดภัย** ควรระบุข้อจำกัดนี้ไว้บนแดชบอร์ดด้วย

---

6. เกณฑ์ระดับเสียงที่แนะนำ

อ้างอิงแนวทางระดับเสียงในสภาพแวดล้อมทั่วไป

ระดับ	dBA	เทียบเคียง	ผลต่อสมาธิ
เงียบมาก	< 30	ห้องสมุดตอนกลางคืน	—
เงียบ	**30–45**	**ห้องนอน · ห้องทำงานเงียบ**	✅ เหมาะกับงานที่ต้องใช้สมาธิสูง
ปกติ	**45–60**	**สำนักงานทั่วไป · การสนทนา**	✅ ยอมรับได้
ดัง	60–75	ร้านอาหาร · การจราจร	⚠ เริ่มรบกวนสมาธิ
ดังมาก	75–85	ถนนพลุกพล่าน · เครื่องดูดฝุ่น	⚠ รบกวนชัดเจน
อันตราย	> 85	เสี่ยงต่อการได้ยินถ้าสัมผัสนาน	🔴 ควรใส่อุปกรณ์ป้องกัน

> TIP **เชื่อมโยงกับ Focus feature:** เสียงเกิน 60 dBA สัมพันธ์กับสมาธิที่ลดลง เมื่อรวมกับข้อมูล EAR/blink จากกล้อง แดชบอร์ดจะอธิบายได้ว่า "สมาธิลดลงเพราะเสียงรบกวน" ไม่ใช่แค่รายงานว่าสมาธิลด

---

7. ผลกระทบต่องบไฟและ USB

รายการ	ค่า
กระแสที่ไมค์ใช้	~50–100 mA
เพดาน USB รวมของ Pi 5 (PSU ธรรมดา)	600 mA
เพดาน USB รวมของ Pi 5 (PSU 5V/5A)	1.6 A

ไมค์อย่างเดียวไม่มีปัญหาแน่นอน แต่ถ้าจะเพิ่ม USB webcam ด้วย ต้องดู `docs/03-power-budget.md`

---

8. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
`arecord -l` ไม่เจออุปกรณ์	ไมค์ไม่ใช่ UAC · พอร์ต USB มีปัญหา	`lsusb` ตรวจว่าระบบเห็นไหม · ลองพอร์ตอื่น
`Permission denied`	ผู้ใช้ไม่อยู่ในกลุ่ม `audio`	`sudo usermod -aG audio $USER` แล้ว logout/login
ค่า dB นิ่งไม่เปลี่ยนเลย	AGC ทำงานอยู่	ปิด AGC ด้วย `amixer`
ค่า dB สูงตลอดแม้ห้องเงียบ	gain สูงเกิน · พัดลม PMS7003/Pi อยู่ใกล้	ลด gain แล้ว calibrate ใหม่ · ย้ายไมค์ห่างพัดลม ≥30 cm
ค่ากระโดดขึ้นลงรุนแรง	หน้าต่าง RMS สั้นเกินไป	เพิ่มเป็น 1 วินาที · เพิ่ม smoothing
ค่าเปลี่ยนไปเฉย ๆ หลังรีบูต	หมายเลข card ของ ALSA เปลี่ยน	ตั้ง udev rule ผูกชื่อกับ USB ID
`Device or resource busy`	มีโปรเซสอื่นถือไมค์อยู่	`fuser -v /dev/snd/*` หาตัวที่ถืออยู่
เสียง drop เป็นระยะ (overrun)	CPU ถูกใช้หนักจาก thread อื่น	เพิ่มขนาด ALSA buffer · ลดความสำคัญของงานอื่น

กล้อง — สำหรับ Focus feature (face / EAR / motion)

> **บทบาทในระบบนี้:** ป้อนข้อมูลให้ `webcam_json` ในตาราง `environment` และแท็บ Focus บนแดชบอร์ด โดยตรวจจับใบหน้า, คำนวณ Eye Aspect Ratio (EAR) เพื่อดูความง่วง, และนับการขยับตัว เกณฑ์ที่ใช้อยู่แล้วในโปรเจกต์: `WEBCAM.drowsyEarBelow = 0.18` และ `blinkNormal = [8, 21]` ครั้ง/นาที (ดู `config/sensors.js`)

1. สรุปคำแนะนำ

อันดับ	ตัวเลือก	ราคาโดยประมาณ	เหมาะกับ
1	**Raspberry Pi Camera Module 3 (Standard)**	~1,000–1,300 บาท	✓ **ตัวเลือกที่ดีที่สุดสำหรับโปรเจกต์นี้**
2	Raspberry Pi Camera Module 3 **Wide**	~1,200–1,500 บาท	ถ้ากล้องต้องอยู่ใกล้หน้า (< 40 cm)
3	Logitech C270 (USB, 720p)	~700–900 บาท	ถ้าต้องการวางกล้องห่างจาก Pi มาก
—	Logitech C920 (USB, 1080p)	~2,000–2,800 บาท	เกินความจำเป็นสำหรับงานนี้
!	Camera Module 3 **NoIR**	~1,000–1,300 บาท	เฉพาะกรณีต้องทำงานในความมืดสนิท (ต้องมีไฟ IR เพิ่ม)

2. ทำไม Camera Module 3 ถึงเหมาะกว่า USB webcam

ประเด็น	Camera Module 3 (CSI-2)	USB webcam
โควตากระแส USB	✓ **ไม่กินเลย** — ใช้ขั้วต่อ CSI แยก	✗ กิน 150–400 mA จากเพดาน 600 mA / 1.6 A
ความเข้ากันได้กับไมค์ USB	✓ ไม่แย่งอะไรกัน	! มีการ์ดเสียงในตัว ทำให้เลข ALSA card ของไมค์เปลี่ยน
Autofocus	✓ Phase Detection AF (เร็ว)	C270 ไม่มี · C920 มี AF แบบ contrast (ช้ากว่า)
ประสิทธิภาพในแสงน้อย	✓ ดี — พิกเซล 1.4 µm · F1.8 · back-illuminated	ปานกลาง
การควบคุมจากซอฟต์แวร์	✓ `libcamera` / `picamera2` ควบคุม exposure/AF ได้ละเอียด	จำกัดตามที่ V4L2 เปิดให้
ภาระ CPU	✓ ต่ำกว่า — ISP ในตัว Pi 5 ช่วยประมวลผล	ต้อง decode MJPEG ด้วย CPU
ความยาวสายที่ทำได้	สาย FPC 200 mm (ต่อยาวได้แต่มีข้อจำกัด)	สาย USB ยาวได้ 3–5 m
ราคา	ปานกลาง	C270 ถูกกว่า

> **KEY เหตุผลชี้ขาดคือโควตา USB** — โปรเจกต์นี้มีไมค์ USB อยู่แล้ว ถ้าเพิ่ม USB webcam 1080p เข้าไปด้วย จะใช้ไปราว 500 mA ซึ่งเกินเพดาน 600 mA ของ PSU ที่ไม่ใช่ 5A ทำให้ภาพกระตุกหรืออุปกรณ์หลุด การใช้ CSI camera ทำให้ปัญหานี้หายไปทั้งหมด

3. สเปก Raspberry Pi Camera Module 3

จาก product brief ทางการ (มิถุนายน 2024)

รายการ	ค่า
เซนเซอร์	Sony IMX708 (back-illuminated, stacked CMOS)
ความละเอียด	11.9 ล้านพิกเซล (4608 × 2592)
ขนาดพิกเซล	1.4 × 1.4 µm
ขนาดเซนเซอร์	เส้นทแยงมุม 7.4 mm
โหมดวิดีโอที่ใช้บ่อย	1080p50 · 720p100 · 480p120
ระบบโฟกัส	**Phase Detection Autofocus (PDAF)**
HDR	รองรับ (สูงสุด 3 ล้านพิกเซล)
อินเทอร์เฟซ	CSI-2 (Pi 5 ใช้ขั้วต่อ 22 ขา)
ความยาวสายริบบอน	200 mm
อุณหภูมิใช้งาน	0 – 50 °C
ผลิตถึงอย่างน้อย	มกราคม 2030

เปรียบเทียบรุ่นย่อย

	Standard	Wide	NoIR	Wide NoIR
ระยะโฟกัส	10 cm – ∞	**5 cm – ∞**	10 cm – ∞	5 cm – ∞
ความยาวโฟกัส	4.74 mm	2.75 mm	4.74 mm	2.75 mm
มุมรับภาพ (แนวนอน)	**66°**	**102°**	66°	102°
มุมรับภาพ (แนวทแยง)	75°	120°	75°	120°
รูรับแสง	**F1.8**	F2.2	F1.8	F2.2
ฟิลเตอร์ตัด IR	✓ มี	✓ มี	✗ ไม่มี	✗ ไม่มี
ขนาด	25 × 24 × 11.5 mm	25 × 24 × 12.4 mm	เท่ากัน	เท่ากัน

เลือกรุ่นไหนดี

ระยะจากกล้องถึงใบหน้า	รุ่นที่แนะนำ	เหตุผล
50 cm – 1.5 m (บนโต๊ะทำงานทั่วไป)	**Standard** ✅	มุม 66° ครอบคลุมพอ · F1.8 รับแสงได้ดีกว่า
30 – 50 cm (ติดขอบจอ)	**Wide**	มุม 102° เก็บใบหน้าได้ครบแม้อยู่ใกล้
ต้องทำงานในห้องมืดสนิท	NoIR + ไฟ IR 850nm	⚠ ดูข้อควรระวังด้านล่าง

> ⚠ **NoIR ไม่ใช่ "กล้องมองกลางคืน" ในตัวเอง** — มันแค่ไม่มีฟิลเตอร์ตัด IR จึงต้องมี**ไฟ IR ส่องเพิ่ม**ถึงจะเห็นภาพในความมืด และเมื่อใช้ในเวลากลางวันสีจะเพี้ยน สำหรับโปรเจกต์นี้ที่วัด **แสง (BH1750) ควบคู่ไปด้วย** การใช้ NoIR จะทำให้ตีความยากขึ้น **แนะนำ Standard** แล้วให้ระบบรายงานว่า "แสงน้อยเกินไปสำหรับการตรวจจับ" แทน

---

4. การติดตั้ง Camera Module 3

การต่อสาย

Raspberry Pi 5 ใช้ขั้วต่อ **CSI 22 ขา** (เล็กกว่า Pi รุ่นก่อนที่ใช้ 15 ขา) Camera Module 3 มาพร้อมสาย 15 ขา → **ต้องซื้อสายแปลง 15-to-22 pin เพิ่ม** (Pi 5 บางชุดแถมมาให้แล้ว — ตรวจในกล่องก่อนสั่งซื้อ)

Pi 5 มีขั้วต่อ CSI/DSI สองช่อง (`CAM/DISP 0` และ `CAM/DISP 1`) ใช้ช่องไหนก็ได้

> ⚠ ปิดเครื่องและถอดสายไฟก่อนเสียบสายกล้องทุกครั้ง
> ดึงตัวล็อกขึ้น → เสียบสายให้หน้าสัมผัสโลหะหันถูกด้าน → กดตัวล็อกลง

ตรวจสอบ

```
rpicam-hello --list-cameras          # ต้องเห็น imx708
rpicam-hello -t 5000                 # แสดงภาพพรีวิว 5 วินาที
```

> Pi 5 กับ Raspberry Pi OS Bookworm ขึ้นไป **ตรวจจับกล้องอัตโนมัติ** ไม่ต้องใส่ `dtoverlay` หรือเปิดอะไรใน `raspi-config` เหมือนรุ่นเก่า

---

5. ซอฟต์แวร์

ไลบรารีที่แนะนำ

```
sudo apt install -y python3-picamera2      # ✅ ทางการ · มากับ Raspberry Pi OS อยู่แล้ว
pip install opencv-python mediapipe        # ประมวลผลภาพ
```

ไลบรารี	ใช้ทำอะไร
picamera2	เข้าถึงกล้อง CSI · ควบคุม AF/exposure · ดึงเฟรมเป็น numpy array
OpenCV	แปลงสี · ปรับขนาด · วาดผลลัพธ์
MediaPipe Face Mesh	หา landmark 468 จุดบนใบหน้า → คำนวณ EAR

> ถ้าใช้ USB webcam แทน ให้ใช้ `cv2.VideoCapture()` ธรรมดา — โค้ดส่วนประมวลผลเหมือนกันทั้งหมด **ออกแบบให้สลับแหล่งภาพได้ผ่าน config** (`CAMERA_SOURCE=csi` หรือ `usb`) จะดีที่สุด

การคำนวณ EAR (Eye Aspect Ratio)

```
        p2      p3
         •       •
   p1 •            • p4          EAR = (|p2−p6| + |p3−p5|) / (2 × |p1−p4|)
         •       •
        p6      p5
```

- ตาเปิดปกติ: EAR ≈ 0.25 – 0.35
- **ตาปิด/ง่วง: EAR < 0.18** ← ตรงกับ `WEBCAM.drowsyEarBelow` ใน `config/sensors.js`
- นับการกะพริบ: EAR ตกลงต่ำกว่าเกณฑ์แล้วกลับขึ้นภายใน ~100–400 ms
- อัตราการกะพริบปกติ: **8–21 ครั้ง/นาที** ← ตรงกับ `WEBCAM.blinkNormal`

> ค่าเกณฑ์เหล่านี้**มีอยู่ใน `config/sensors.js` แล้ว** — โค้ดฝั่ง Pi ต้องอ่านค่าจาก config ของตัวเอง ที่ตั้งให้ตรงกัน **ห้ามเขียนตัวเลข 0.18 ซ้ำในโค้ด**

การตั้งค่าที่แนะนำ

พารามิเตอร์	ค่าที่แนะนำ	เหตุผล
ความละเอียดที่ประมวลผล	**640 × 480**	MediaPipe ไม่ต้องการภาพใหญ่ · ประหยัด CPU มาก
เฟรมเรต	**10 – 15 fps**	เพียงพอสำหรับนับการกะพริบ (การกะพริบกินเวลา ~100–400 ms)
Autofocus	`Continuous` หรือ ล็อกไว้ที่ระยะโต๊ะทำงาน	ถ้าคนนั่งอยู่กับที่ ล็อกไว้จะนิ่งกว่าและประหยัด CPU
รอบส่งสรุปขึ้น API	ทุก 5 วินาที (พร้อมข้อมูลเซนเซอร์อื่น)	ไม่ต้องส่งทุกเฟรม — สรุปเป็นค่าเฉลี่ยในหน้าต่างนั้น

> ❗ **อย่าประมวลผลที่ 1080p 30fps** — จะกิน CPU ของ Pi 5 เกือบเต็ม ทำให้ thread ของ SGP30 พลาดจังหวะ 1 Hz และค่า TVOC เพี้ยน (ดู sgp30.md § 5)

---

6. PRIVACY ความเป็นส่วนตัว — สำคัญมาก

โปรเจกต์นี้ติดกล้องในพื้นที่ส่วนตัว จึงต้องออกแบบให้ปลอดภัยตั้งแต่แรก

หลักการ	ปฏิบัติอย่างไร
ห้ามส่งภาพออกนอกเครื่อง	ประมวลผลบน Pi ทั้งหมด · ส่งขึ้น API เฉพาะ**ตัวเลขสรุป** (EAR, จำนวนกะพริบ, มี/ไม่มีคน)
ห้ามบันทึกภาพลงดิสก์	ไม่เขียนไฟล์ภาพหรือวิดีโอเลย นอกจากตอน debug ที่เปิดด้วย flag ชัดเจน
ห้ามเก็บ face embedding	ใช้ landmark เพื่อคำนวณ EAR แล้วทิ้งทันที ไม่เก็บข้อมูลที่ระบุตัวตนได้
มีสวิตช์ปิดกล้อง	ให้ปิดกล้องได้จาก config หรือสวิตช์จริง โดยระบบส่วนอื่นยังทำงานต่อ
มีไฟแสดงสถานะ	LED ที่บอกชัดเจนว่ากล้องกำลังทำงาน
แจ้งทุกคนในพื้นที่	บอกให้ทุกคนที่เข้ามาในห้องทราบว่ามีกล้อง

> LEGAL **ถ้าติดตั้งในพื้นที่ที่มีคนอื่นนอกจากตัวเอง** (ห้องเรียน · ออฟฟิศ · พื้นที่ส่วนกลาง) ควรขอความยินยอมจากทุกคนก่อน — และในหลายบริบทถือเป็นข้อกำหนดทางกฎหมาย ไม่ใช่แค่มารยาท ประเทศไทยมี พ.ร.บ. คุ้มครองข้อมูลส่วนบุคคล (PDPA) ที่ครอบคลุมข้อมูลชีวมิติ **หากเป็นโครงการที่ใช้จริงกับผู้อื่น ควรปรึกษาผู้รู้ด้านกฎหมายก่อน** — เอกสารนี้ไม่ใช่คำแนะนำทางกฎหมาย

7. ผลกระทบต่องบไฟและ CPU

	Camera Module 3	USB webcam 1080p
กระแส	~250 mA @3.3V (จากบอร์ด ไม่ใช่โควตา USB)	250–400 mA จากโควตา USB
CPU ตอนประมวลผล 640×480 @15fps	~25–35% ของหนึ่งคอร์	~35–45% (ต้อง decode MJPEG เพิ่ม)
กำลังไฟรวมที่เพิ่ม	~2 W	~2.5 W

ทั้งสองแบบอยู่ในงบของ PSU 27W → ดู `docs/03-power-budget.md`

8. การแก้ปัญหา

อาการ	สาเหตุที่น่าจะเป็น	วิธีแก้
`--list-cameras` ไม่เจอกล้อง	สายเสียบกลับด้าน · ใช้สาย 15 ขากับ Pi 5 โดยไม่มีตัวแปลง	ตรวจทิศหน้าสัมผัส · ใช้สายแปลง 15→22 ขา
ภาพเบลอตลอด	AF ล็อกไว้ผิดระยะ	ตั้งเป็น continuous AF หรือล็อกใหม่ที่ระยะโต๊ะ
ตรวจจับใบหน้าไม่เจอในที่แสงน้อย	แสงต่ำกว่า ~100 lx	ใช้ค่าจาก BH1750 เตือนผู้ใช้ให้เปิดไฟ
เฟรมเรตตก / ระบบหน่วง	ประมวลผลความละเอียดสูงเกินไป	ลดเป็น 640×480 · ลดเป็น 10 fps
ค่า TVOC เริ่มเพี้ยนหลังเปิดกล้อง	CPU เต็มจน thread SGP30 พลาดจังหวะ 1 Hz	ลดภาระกล้อง · เพิ่ม priority ให้ thread SGP30
USB webcam ทำให้ไมค์หาย	เลข ALSA card เปลี่ยน	ตั้ง udev rule (microphone-usb.md § 3)
USB webcam หลุดเป็นระยะ	เกินโควตากระแส USB	ใช้ PSU 5A · หรือเปลี่ยนไปใช้ CSI camera
กล้องร้อนจนภาพมี noise	อยู่ในกล่องปิดใกล้ Pi	ระบายอากาศ · ย้ายออกห่างจากบอร์ด

ทบทวนวรรณกรรม — สภาพแวดล้อมภายในอาคารกับสมาธิและสมรรถนะการคิด

> **บทนี้ทำหน้าที่เป็นบทที่ 2 ของรายงานโครงงาน** — อธิบายว่าเกณฑ์เตือนทุกค่าใน `config/sensors.js` มาจากไหน มีหลักฐานอะไรรองรับ หลักฐานแข็งแค่ไหน และมีข้อโต้แย้งอะไรบ้าง
>
> รูปแบบการอ้างอิง: **APA 7** · การอ้างอิงในเนื้อหาเป็น (ชื่อผู้แต่ง, ปี)

2.1 ที่มาและความสำคัญ

การเรียนและการทำงานส่วนใหญ่เกิดขึ้น**ภายในอาคาร** ซึ่งเป็นพื้นที่ปิดที่สภาพแวดล้อม เบี่ยงเบนจากธรรมชาติได้มาก — คาร์บอนไดออกไซด์สะสมจากลมหายใจ อุณหภูมิถูกควบคุมด้วย เครื่องปรับอากาศ ฝุ่นละอองลอดเข้ามาจากภายนอก และเสียงรบกวนจากคนรอบข้าง

งานวิจัยในช่วงสองทศวรรษที่ผ่านมาแสดงให้เห็นว่าปัจจัยเหล่านี้**ไม่ได้ส่งผลแค่ความสบาย** แต่กระทบถึงสมรรถนะการคิดและความสามารถในการจดจ่ออย่างวัดได้ อย่างไรก็ตาม งานวิจัยส่วนใหญ่**ศึกษาทีละปัจจัยในห้องปฏิบัติการ** ขณะที่สภาพจริงปัจจัยเหล่านี้เกิดขึ้นพร้อมกัน

โครงงานนี้จึงสร้างระบบที่วัดปัจจัยแวดล้อมหลายตัวพร้อมกันแบบต่อเนื่อง ควบคู่กับตัวชี้วัด พฤติกรรมความจดจ่อจากกล้อง เพื่อให้เห็นความสัมพันธ์ในสภาพการใช้งานจริง

2.2 วิธีสืบค้นวรรณกรรม

การทบทวนนี้**ไม่ใช่การทบทวนอย่างเป็นระบบตามระเบียบวิธี PRISMA** แต่เป็นการสืบค้น แบบมีเกณฑ์เพื่อหาหลักฐานที่ใช้กำหนดเกณฑ์เตือนของระบบ

หัวข้อ	รายละเอียด
คำค้น	CO_2 / temperature / PM2.5 / noise / illuminance / TVOC × cognitive performance / concentration / office / classroom
แหล่ง	วารสารที่มีการทบทวนโดยผู้ทรงคุณวุฒิ — *Environmental Health Perspectives*, *Indoor Air*, *Building and Environment*, *Applied Acoustics*, *Environmental Research Letters*, *Scientific Reports* · รายงานจาก Lawrence Berkeley National Laboratory
เกณฑ์คัดเข้า	มีตัววัดสมรรถนะเชิงวัตถุวิสัย · ระบุระดับความเข้มข้น/ระดับที่ทดสอบชัดเจน · ให้ผลเชิงปริมาณที่นำมากำหนดเกณฑ์ได้
เกณฑ์คัดออก	ใช้เฉพาะการประเมินตนเอง · ไม่ระบุระดับที่ทดสอบ · เป็นบทความทั่วไปที่ไม่มีข้อมูลปฐมภูมิ
ให้น้ำหนักพิเศษ	การทบทวนอย่างเป็นระบบและ meta-analysis · งานที่มีทั้งฝ่ายสนับสนุนและโต้แย้ง

> **ข้อจำกัดที่ต้องระบุ:** การสืบค้นทำผ่านเครื่องมือค้นหาทั่วไป ไม่ได้ใช้ฐานข้อมูลเฉพาะทาง เช่น Scopus หรือ Web of Science จึงอาจมีงานวิจัยสำคัญที่ตกหล่น

2.3 ตารางสังเคราะห์งานวิจัย

ตัวแปร	เซนเซอร์	งานวิจัยหลัก	ระดับที่ทดสอบ	ผลที่พบ	ความแข็งของหลักฐาน
CO_2	SCD40	Satish et al. (2012)	600 · 1,000 · 2,500 ppm	ที่ 1,000 ppm แย่ลง 6/9 ด้าน · ที่ 2,500 ppm แย่ลง 7/9 ด้าน	ปานกลาง — มีข้อโต้แย้ง
CO_2 + การระบายอากาศ	SCD40	Allen et al. (2016)	Conventional vs Green vs Green+	คะแนนสูงกว่า 61% และ 101% ตามลำดับ	ปานกลาง
อุณหภูมิ	SCD40	Seppänen et al. (2006)	15–35 °C (รวม 24 งานวิจัย)	จุดสูงสุด 21.75 °C · ที่ 30 °C เหลือ 91.1%	**แข็ง**
PM2.5	PMS7003	Cedeño-Laurent et al. (2021)	3.7 vs 18.0 μg/m³	Stroop ช้าลง · throughput ลดลง	**แข็ง**
PM2.5	PMS7003	Shehab & Pope (2019)	เทียบก่อน-หลังสัมผัสฝุ่น	MMSE ลดลง · ความสนใจแบบเลือกเฟ้นช้าลง	แข็ง
เสียง	ไมโครโฟน USB	Jahncke et al. (2011)	39 vs 51 dB(A)	จำคำได้น้อยลง · เหนื่อยขึ้น · แรงจูงใจลดลง	**แข็ง**
แสง	BH1750	EN 12464-1	มาตรฐาน 500 lx	เกณฑ์ความสว่างพื้นที่ทำงาน	ปานกลาง (คนละหน่วยกับงานวิจัยล่าสุด)
TVOC	SGP30	Mølhave (1987) · Nakaoka et al. (2014)	—	สัมพันธ์กับอาการ SBS รวมถึงปัญหาด้านการรับรู้	ปานกลาง

2.4 คาร์บอนไดออกไซด์ (CO_2)

2.4.1 หลักฐานที่พบผลกระทบ

การศึกษาที่ถูกอ้างอิงมากที่สุดคือ Satish et al. (2012) ซึ่งดำเนินการที่ Lawrence Berkeley National Laboratory ผู้เข้าร่วม 24 คนอยู่ในห้องจำลองสำนักงานครั้งละ 2.5 ชั่วโมง ภายใต้ความเข้มข้น CO_2 สามระดับคือ 600, 1,000 และ 2,500 ppm วัดผลด้วยแบบทดสอบ Strategic Management Simulation (SMS) ซึ่งประเมินการตัดสินใจ 9 ด้าน

ผลการศึกษาพบว่าที่ 1,000 ppm คะแนนลดลงอย่างมีนัยสำคัญใน 6 จาก 9 ด้าน และที่ 2,500 ppm ลดลงมากใน 7 จาก 9 ด้าน โดยด้านที่ได้รับผลกระทบรุนแรงที่สุดคือ **การริเริ่ม (initiative)** และ **การคิดเชิงกลยุทธ์ (strategic thinking)** ซึ่งถูกจัดอยู่ในระดับที่ผู้วิจัยเรียกว่า "dysfunctional" (Satish et al., 2012)

Allen et al. (2016) ศึกษาต่อยอดในโครงการ COGfx โดยให้พนักงานออฟฟิศทำงานภายใต้ สภาพอาคารสามแบบ พบว่าคะแนนการทำงานของสมองในวัน Green สูงกว่าวัน Conventional ร้อยละ 61 และในวัน Green+ ซึ่งเพิ่มอัตราการระบายอากาศ สูงกว่าถึงร้อยละ 101 (Allen et al., 2016)

ในสภาพการทำงานจริง Cedeño-Laurent et al. (2021) ติดตามพนักงานออฟฟิศหลายประเทศ และพบว่าระดับ CO_2 และ PM2.5 ภายในอาคารที่สูงขึ้นสัมพันธ์กับสมรรถนะที่ลดลงในแบบทดสอบ Stroop และ ADD ซึ่งเป็นหลักฐานจากสภาพแวดล้อมจริงไม่ใช่ห้องปฏิบัติการ

2.4.2 ⚠ หลักฐานที่โต้แย้ง

Zhang et al. (2017) ออกแบบการทดลองแยกอิทธิพลของ **CO_2 บริสุทธิ์** ออกจาก **bioeffluents** (สารที่ร่างกายมนุษย์ปล่อยออกมา) โดยเติม CO_2 บริสุทธิ์ให้ถึง 1,000 และ 3,000 ppm เปรียบเทียบกับการลดอัตราระบายอากาศจนกระทั่ง CO_2 จาก bioeffluents ถึงระดับเดียวกัน ผลชี้ว่าผลกระทบส่วนใหญ่อาจมาจาก bioeffluents ไม่ใช่ตัว CO_2 เอง

Du et al. (2020) ทบทวนวรรณกรรมอย่างวิพากษ์และสรุปว่าผลการศึกษาที่ผ่านมา **ไม่สอดคล้องกัน** และหลายงานมีข้อจำกัดด้านระเบียบวิธี เช่น ขนาดกลุ่มตัวอย่างเล็ก และการควบคุมตัวแปรแทรกซ้อนที่ไม่เพียงพอ

การทบทวนอย่างเป็นระบบและ meta-analysis (2023) ซึ่งรวม 15 งานวิจัยที่เข้าเกณฑ์ เปรียบเทียบกลุ่มควบคุม (< 1,000 ppm) กับสามช่วงความเข้มข้น สรุปว่า **CO_2 ที่ระดับต่ำกว่า 5,000 ppm ส่งผลต่อสมรรถนะการคิดจริง** โดย **งานที่ซับซ้อนได้รับผลกระทบมากกว่างานง่าย** อย่างมีนัยสำคัญ

2.4.3 การตีความที่ใช้ในโครงงานนี้

โครงงานนี้ใช้ CO_2 เป็น **ตัวบ่งชี้คุณภาพการระบายอากาศ (ventilation proxy)** มากกว่าจะถือว่าเป็นสารพิษโดยตรง เหตุผลคือ เมื่อ CO_2 สะสมสูง แปลว่า bioeffluents และมลพิษอื่นที่เกิดจากคนในห้องก็สะสมสูงตามไปด้วย การใช้ CO_2 กระตุ้นให้ระบายอากาศ จึงสมเหตุสมผลไม่ว่ากลไกที่แท้จริงจะเป็นอย่างไร

เกณฑ์ที่ใช้:

ระดับ	ppm	ที่มา
ดี	< 800	ต่ำกว่าระดับที่ Satish et al. (2012) เริ่มพบผล
ปานกลาง	800–1,000	เข้าใกล้เกณฑ์วิกฤต
แย่	1,000–1,500	Satish et al. (2012) พบผลชัดเจนที่ 1,000 ppm
แย่มาก	1,500–2,500	ช่วงที่ meta-analysis (2023) พบผลต่อเนื่อง
อันตราย	> 2,500	Satish et al. (2012) — ระดับ "dysfunctional"

2.5 อุณหภูมิ

Seppänen et al. (2006) รวบรวมและวิเคราะห์ 24 งานวิจัย (ภาคสนาม 11 งาน · ห้องปฏิบัติการ 9 งาน) ที่ใช้ตัววัดสมรรถนะเชิงวัตถุวิสัย เช่น การประมวลผลข้อความ การคำนวณอย่างง่าย และเวลาที่ใช้ต่อสายในศูนย์บริการลูกค้า จากนั้นคำนวณอัตราการเปลี่ยนแปลงสมรรถนะต่ออุณหภูมิ และทำ regression ถ่วงน้ำหนักด้วยขนาดกลุ่มตัวอย่างและความเกี่ยวข้องของงาน

ผลการศึกษา:

- สมรรถนะเพิ่มขึ้นตามอุณหภูมิจนถึง **21–22 °C**
- สมรรถนะลดลงเมื่อเกิน **23–24 °C**
- จุดที่เส้นโค้งตัดแกนนอน (สมรรถนะสูงสุด) อยู่ที่ **21.75 °C**
- ที่ **30 °C** สมรรถนะเหลือ **ร้อยละ 91.1** ของค่าสูงสุด (ลดลงร้อยละ 8.9)
- การทบทวนก่อนหน้าพบว่าสมรรถนะลดลงราว **ร้อยละ 2 ต่ออุณหภูมิ** ในช่วง 25–32 °C

สมการความสัมพันธ์ที่ได้จากการถ่วงน้ำหนักแบบผสม:

```
P = 0.1647524·T − 0.0058274·T² + 0.0000623·T³ − 0.4685328

โดย P = สมรรถนะเทียบกับค่าสูงสุด · T = อุณหภูมิห้อง (°C)
```

! ประเด็นที่ต้องยอมรับตรงไปตรงมา

เกณฑ์ "สบาย" ที่โครงงานนี้ใช้ (`THRESHOLDS.temp.okLo = 20` , `okHi = 28`) **กว้างกว่าช่วงที่ให้สมรรถนะสูงสุดตามงานวิจัยอย่างมาก** — เมื่อแทน T = 28 ลงในสมการข้างต้น จะพบว่าสมรรถนะลดลงไปแล้วราวร้อยละ 3–5

เหตุผลที่ยังคงเกณฑ์เดิมไว้:

1. ในบริบทภูมิอากาศไทย การตั้งเกณฑ์แคบที่ 21–24 °C จะทำให้ระบบแจ้งเตือนเกือบตลอดเวลา จนผู้ใช้เพิกเฉย (alarm fatigue)
2. การรักษาอุณหภูมิห้องที่ 22 °C ตลอดเวลามีต้นทุนพลังงานสูงมาก

ข้อเสนอสำหรับการพัฒนาต่อ: แยกการแสดงผลเป็นสองระดับ — "ช่วงสบาย" (กว้าง) และ "ช่วงสมรรถนะสูงสุด" (21–24 °C) แล้วแสดงทั้งสองค่าบนแดชบอร์ดพร้อมกัน

2.6 ฝุ่นละอองขนาดเล็ก (PM2.5)

Cedeño-Laurent et al. (2021) ศึกษาพนักงานออฟฟิศหลายประเทศแบบติดตามระยะยาว ในสถานที่ทำงานจริง พบว่าระดับ PM2.5 ภายในอาคารที่สูงขึ้นสัมพันธ์กับเวลาตอบสนอง ในแบบทดสอบ Stroop ที่ช้าลงและ throughput ที่ลดลง ที่สำคัญคือเมื่อใช้เครื่องฟอกอากาศ ลด PM2.5 จาก **18.0 μg/m³ เหลือ 3.7 μg/m³** สมรรถนะบางด้านดีขึ้นอย่างมีนัยสำคัญ

Shehab และ Pope (2019) ใช้การทดลองแบบ double cross-over พบว่าหลังสัมผัสฝุ่น จากการจุดเทียนและจากการเดินทางกลางแจ้ง คะแนน MMSE ลดลงอย่างมีนัยสำคัญทางสถิติ และความเร็วในการตรวจจับอัตโนมัติจากแบบทดสอบ Ruff 2 & 7 ซึ่งวัดความสนใจแบบเลือกเฟ้น ลดลงเช่นกัน

> **KEY** **ข้อค้นพบที่สำคัญที่สุดสำหรับโครงงานนี้:** ตัวเลข 3.7 เทียบกับ 18.0 μg/m³ แสดงว่าผลกระทบเกิดขึ้นที่ระดับ**ต่ำกว่ามาตรฐานของประเทศไทย (37.5 μg/m³) มาก** กล่าวคือไม่จำเป็นต้องรอให้ถึงระดับที่ประกาศว่า "มีผลต่อสุขภาพ" จึงจะเห็นผลต่อสมรรถนะ

เกณฑ์ที่ใช้:

ระดับ	μg/m³	ที่มา
ดีมาก	0–15	WHO Air Quality Guideline (2021), ค่าเฉลี่ย 24 ชั่วโมง
ดี	15–25	WHO interim target 4
ปานกลาง	25–37.5	มาตรฐานประเทศไทย 24 ชั่วโมง (บังคับใช้ 1 มิถุนายน 2566)
เริ่มมีผลต่อสุขภาพ	37.5–75	WHO interim target 1

> ! **ข้อควรระวังเชิงระเบียบวิธี:** มาตรฐานเหล่านี้เป็นค่าเฉลี่ย 24 ชั่วโมงในบรรยากาศทั่วไป ไม่ใช่ค่าชั่วขณะภายในอาคาร ระบบนี้จึงใช้**ค่าเฉลี่ยเคลื่อนที่ 1 ชั่วโมง**ในการตัดสินระดับเตือน เพื่อไม่ให้แจ้งเตือนบ่อยเกินจริงจากค่าที่กระโดดชั่วขณะ เช่น ตอนจุดธูปหรือทำอาหาร

2.7 เสียงรบกวน

Jahncke et al. (2011) ให้ผู้เข้าร่วมทำงานเป็นเวลา 2 ชั่วโมงในสำนักงานแบบเปิดจำลอง ภายใต้เสียงรบกวนสองระดับคือ 39 และ 51 dB(A) พบว่าเมื่ออยู่ในสภาพเสียงสูง ผู้เข้าร่วม **จำคำได้น้อยลง ประเมินตนเองว่าเหนื่อยมากขึ้น และมีแรงจูงใจในการทำงานน้อยลง** โดยสมรรถนะลดลงทั้งในงานที่ใช้ความจำและงานที่ไม่ใช้ความจำ

Jahncke et al. (2013) ศึกษาต่อและพบว่างานที่อาศัยความจำระยะสั้นแบบ episodic และการทวนซ้ำข้อมูลมีความไวต่อการรบกวนจากเสียงพูดมากกว่างานประเภทอื่น

กลไก: Irrelevant Speech Effect

เสียงพูดที่ไม่เกี่ยวข้องกับงานถูกสมองประมวลผลความหมาย**โดยอัตโนมัติและห้ามไม่ได้** กระบวนการนี้ไปแย่งทรัพยากรที่ใช้ในการทวนซ้ำข้อมูลในความจำใช้งาน (working memory) จึงทำให้สมรรถนะลดลงแม้ระดับเสียงจะไม่สูงมาก

> **51 dB(A) ไม่ถือว่าเป็นเสียงดัง** — เป็นระดับของสำนักงานทั่วไป แต่ก็เพียงพอที่จะลดสมรรถนะได้แล้ว เกณฑ์ 60 dB ที่โครงงานนี้ใช้จึงเป็นเกณฑ์ที่**ผ่อนปรน** กว่าระดับที่งานวิจัยพบผลเสียเสียอีก

ข้อจำกัดของการวัดในโครงงานนี้: ไมโครโฟน USB วัดได้เพียงระดับความดัง แต่**แยกไม่ได้ว่าเป็นเสียงพูดหรือเสียงคงที่** ทั้งที่งานวิจัยชี้ชัดว่าเสียงพูดรบกวน มากกว่าเสียงคงที่ที่ระดับความดังเท่ากัน

2.8 แสงสว่าง

มาตรฐาน EN 12464-1 กำหนดความสว่างบนพื้นที่ทำงานสำหรับงานอ่าน เขียน พิมพ์ และประมวลผลข้อมูลไว้ที่ **500 lx** และสูงถึง 750 lx สำหรับงานเขียนแบบเทคนิค

อย่างไรก็ตาม งานวิจัยด้านความตื่นตัวและนาฬิกาชีวภาพในระยะหลังใช้หน่วย **melanopic Equivalent Daylight Illuminance (mel-EDI)** แทน photopic lux เนื่องจากเซลล์รับแสงชนิด melanopsin ในจอตาซึ่งควบคุมความตื่นตัวตอบสนองต่อสเปกตรัมแสง ต่างจากเซลล์ที่ใช้ในการมองเห็น คำแนะนำปัจจุบันคือควรได้รับแสงอย่างน้อย **250 lx mel-EDI** ในเวลากลางวัน

! ข้อจำกัดเชิงหน่วยวัด

BH1750 วัดได้เฉพาะ photopic lux และไม่สามารถแปลงเป็น mel-EDI ได้ เพราะการแปลงต้องทราบสเปกตรัมของแหล่งกำเนิดแสง หลอด LED สีขาวเย็น (6500 K) กับสีขาวอุ่น (2700 K) ที่ให้ค่า lux เท่ากันอาจมีค่า mel-EDI ต่างกันได้เกินเท่าตัว

> **ขอบเขตที่ระบบนี้ตอบได้:** "แสงเพียงพอต่อการมองเห็นตามมาตรฐานหรือไม่" **ขอบเขตที่ตอบไม่ได้:** "แสงเหมาะสมต่อการกระตุ้นความตื่นตัวหรือไม่" การพัฒนาต่อควรใช้เซนเซอร์ที่วัดสเปกตรัมได้ เช่น AS7341 หรือ TCS34725

เกณฑ์ที่ใช้:

ระดับ	lux	ที่มา
พอดี	300–500	EN 12464-1 — งานอ่าน เขียน พิมพ์
สลัว	150–300	ต่ำกว่ามาตรฐานงานสำนักงาน
มืด	< 150	ต่ำกว่า ~100 lx ระบบตรวจจับใบหน้าเริ่มทำงานไม่ได้

2.9 สารอินทรีย์ระเหยง่ายรวม (TVOC)

Mølhave (1987) เป็นผู้วางนิยามอาการ Sick Building Syndrome (SBS) โดยระบุว่า อาการระคายเคืองทางประสาทสัมผัสต้องเป็นอาการเด่น อาการหลักที่พบประกอบด้วย **อ่อนเพลีย ระคายเคืองเยื่อเมือก ผิวหนังและตา ปวดศีรษะ เจ็บคอ ไอแห้ง ล้า และปัญหาด้านการรับรู้ (cognitive problems)**

Nakaoka et al. (2014) ศึกษาความสัมพันธ์ระหว่างอาการ SBS กับความเข้มข้นของ VOC ภายในอาคารและกลิ่น และพบความสัมพันธ์ที่มีนัยสำคัญ

Allen et al. (2016) เป็นหนึ่งในไม่กี่งานที่วัด TVOC ควบคู่กับคะแนนการทำงานของสมองโดยตรง ดังปรากฏในชื่อเต็มของงานที่ระบุ "Ventilation, and Volatile Organic Compound Exposures"

ข้อจำกัดของ SGP30

1. ค่า TVOC เป็น**สัญญาณเชิงเปรียบเทียบ ไม่ใช่ความเข้มข้นสัมบูรณ์** เพราะใช้ dynamic baseline ที่ปรับตัวตามสภาพแวดล้อม
2. **แยกชนิดสารไม่ได้** — น้ำหอมกับฟอร์มาลดีไฮด์ให้สัญญาณคล้ายกันทั้งที่อันตรายต่างกันมาก
3. ค่า **eCO_2 ไม่ใช่ CO_2** เป็นค่าประมาณที่คำนวณจากสัญญาณ VOC จึงไม่ควรใช้แทน SCD40

เกณฑ์ที่ใช้ อ้างอิงแนวทางของหน่วยงานสิ่งแวดล้อมเยอรมนี (Umweltbundesamt) ซึ่งเป็นเกณฑ์ TVOC ที่ถูกอ้างอิงแพร่หลายที่สุด: ดีเยี่ยม 0–65 ppb · ดี 65–220 ppb · ปานกลาง 220–660 ppb · แย่ 660–2,200 ppb

2.10 ช่องว่างของงานวิจัย (Research Gap)

จากการทบทวนวรรณกรรมข้างต้น พบช่องว่างสำคัญสี่ประการ

#	ช่องว่าง	รายละเอียด	โครงงานนี้ตอบสนองอย่างไร
1	**ศึกษาทีละตัวแปร**	งานวิจัยส่วนใหญ่ควบคุมตัวแปรอื่นให้คงที่แล้วเปลี่ยนทีละตัวในห้องปฏิบัติการ แต่สภาพจริงตัวแปรเกิดพร้อมกันและอาจเสริมกัน เช่น CO_2 สูงมักมาพร้อมอุณหภูมิสูง	วัด 8 ค่าพร้อมกันแบบต่อเนื่องทุก 5 วินาที
2	**วัดสมรรถนะเป็นครั้งคราว**	ต้องหยุดให้ผู้เข้าร่วมทำแบบทดสอบ ซึ่งรบกวนงานจริงและได้ข้อมูลเป็นจุด ๆ	ใช้ตัวชี้วัดพฤติกรรมจากกล้อง (EAR · อัตราการกะพริบ · การหันหน้า) ที่วัดต่อเนื่องโดยไม่รบกวน
3	**บริบทเขตอบอุ่น**	งานวิจัยส่วนใหญ่ทำในยุโรปและอเมริกาเหนือ เกณฑ์อุณหภูมิและการระบายอากาศจึงอาจไม่ตรงกับภูมิอากาศร้อนชื้นและฤดูฝุ่นของไทย	เก็บข้อมูลในบริบทไทย พร้อมเชื่อมข้อมูลจากกรมอุตุนิยมวิทยาและกรมควบคุมมลพิษ
4	**ต้นทุนสูง**	เครื่องมือวัดระดับงานวิจัยมีราคาสูงมาก ทำให้ทำซ้ำในโรงเรียนหรือบ้านได้ยาก	ใช้เซนเซอร์ราคาเข้าถึงได้ (~3,400–5,500 บาท) พร้อมเอกสารและโค้ดที่ทำซ้ำได้

2.11 กรอบแนวคิดการวิจัย (Conceptual Framework)

ตัวแปรต้น — สภาพแวดล้อม (วัดทุก 5 วินาที)

กลุ่ม	ตัวแปร	หน่วย	เซนเซอร์
คุณภาพอากาศ	CO_2	ppm	SCD40
	PM1.0 · PM2.5 · PM10	µg/m³	PMS7003
	TVOC	ppb	SGP30
สภาวะทางกายภาพ	อุณหภูมิ	°C	SCD40
	ความชื้นสัมพัทธ์	%RH	SCD40
	ความสว่าง	lx	BH1750
	ระดับเสียง	dB	ไมโครโฟน USB

↓ **ตัวแปรควบคุมที่ต้องบันทึกควบคู่:** ช่วงเวลาของวัน · จำนวนคนในห้อง · ประเภทงานที่ทำ

ตัวแปรตาม — พฤติกรรมความจดจ่อ (สรุปทุก 15 วินาที)

ตัวแปร	หน่วย	ที่มา	ตีความอย่างไร
อัตราการหันหน้า	ครั้ง/นาที	กล้อง + MediaPipe	สูง = เสียสมาธิบ่อย
Eye Aspect Ratio (EAR)	สัดส่วน	กล้อง + MediaPipe	ต่ำ = ตาปรือ/ง่วง
อัตราการกะพริบตา	ครั้ง/นาที	กล้อง + MediaPipe	นอกช่วง 8–21 = ผิดปกติ
ระยะเวลาหลับตาต่อเนื่อง	วินาที	กล้อง + MediaPipe	> 1.2 วินาที = ง่วง

> ! **ความสัมพันธ์ระหว่างสองกลุ่มนี้เป็นเพียง "ความสัมพันธ์ที่ตั้งสมมติฐาน"** การออกแบบเชิงสังเกตแบบนี้**ยืนยันความสัมพันธ์เชิงสาเหตุไม่ได้** การจะสรุปว่าเป็นเหตุเป็นผลต้องออกแบบการทดลองที่มีการสุ่มและกลุ่มควบคุม

2.12 สมมติฐานการวิจัย

สมมติฐานต่อไปนี้**ทดสอบได้ทางสถิติ**จากข้อมูลที่ระบบเก็บได้ หากขยายโครงงานเป็นงานวิจัยเต็มรูปแบบ

สมมติฐาน	ข้อความ	ฐานจากวรรณกรรม	วิธีทดสอบที่เสนอ
H1	เมื่อ CO_2 เกิน 1,000 ppm อัตราการหันหน้าต่อนาทีจะสูงกว่าช่วงที่ CO_2 ต่ำกว่า 800 ppm อย่างมีนัยสำคัญ	Satish et al. (2012) · meta-analysis (2023)	เปรียบเทียบค่าเฉลี่ยสองกลุ่มด้วย independent t-test หรือ Mann–Whitney U
H2	อุณหภูมิที่สูงกว่า 26 °C สัมพันธ์กับอัตราการหันหน้าที่สูงขึ้น โดยความสัมพันธ์เป็นแบบไม่เชิงเส้น	Seppänen et al. (2006)	Polynomial regression กำลังสอง เทียบกับ linear model ด้วยค่า R^2
H3	ระดับเสียงเกิน 55 dB สัมพันธ์กับอัตราการหันหน้าที่สูงขึ้นภายในหน้าต่างเวลา 15 วินาทีถัดไป	Jahncke et al. (2011)	Cross-correlation แบบมี time lag
H4	ความสว่างต่ำกว่า 300 lx สัมพันธ์กับค่า EAR เฉลี่ยที่ต่ำลง (สัญญาณความง่วง)	EN 12464-1 · งานวิจัย mel-EDI	Pearson correlation พร้อมควบคุมช่วงเวลาของวัน
H5	เมื่อรวมตัวแปรแวดล้อมหลายตัวเข้าด้วยกัน จะทำนายอัตราการหันหน้าได้ดีกว่าใช้ตัวแปรเดียว	ช่องว่างงานวิจัยข้อ 1	Multiple regression เทียบ adjusted R^2 กับ simple regression

> **ขนาดตัวอย่างขั้นต่ำที่แนะนำ:** เก็บข้อมูลต่อเนื่องอย่างน้อย 14 วัน ครอบคลุมทั้งวันทำงาน และวันหยุด เพื่อให้มีความแปรปรวนของสภาพแวดล้อมเพียงพอต่อการวิเคราะห์

2.13 ข้อจำกัดของโครงงาน

ข้อจำกัด	รายละเอียด	ผลต่อการตีความ
ความสัมพันธ์ ≠ สาเหตุ	เป็นการศึกษาเชิงสังเกต ไม่มีการสุ่มและไม่มีกลุ่มควบคุม	สรุปได้เพียงว่า "สัมพันธ์กัน" ไม่ใช่ "เป็นสาเหตุ"
ตัวชี้วัดเป็น proxy	EAR และการหันหน้าเป็นตัวแทนของความจดจ่อ ไม่ใช่การวัดสมรรถนะการคิดโดยตรง	ต้องเพิ่มแบบทดสอบ เช่น Stroop test จึงจะเทียบกับงานวิจัยที่อ้างอิงได้โดยตรง
กลุ่มตัวอย่างเล็ก	ใช้กับผู้ใช้จำนวนน้อย	สรุปเป็นข้อค้นพบทั่วไปไม่ได้
ไม่มีเครื่องมืออ้างอิง	ไม่ได้สอบเทียบกับเครื่องมือมาตรฐาน	ควรเทียบกับสถานีตรวจวัดของกรมควบคุมมลพิษที่ใกล้ที่สุดเป็นอย่างน้อย
หน่วยแสงไม่ตรงกับงานวิจัยล่าสุด	วัด photopic lux ไม่ใช่ melanopic EDI	ข้อสรุปเรื่องแสงกับความตื่นตัวทำได้จำกัด
แยกประเภทเสียงไม่ได้	วัดเฉพาะระดับ dB	ไม่สามารถแยกผลของ irrelevant speech ออกจากเสียงคงที่
หลักฐาน CO_2 ยังถกเถียง	ดูหัวข้อ 2.4.2	ควรนำเสนอ CO_2 ในฐานะตัวบ่งชี้การระบายอากาศ

2.14 รายการอ้างอิง

#	รายการ
1	Allen, J. G., MacNaughton, P., Satish, U., Santanam, S., Vallarino, J., & Spengler, J. D. (2016). Associations of cognitive function scores with carbon dioxide, ventilation, and volatile organic compound exposures in office workers: A controlled exposure study of green and conventional office environments. *Environmental Health Perspectives, 124*(6), 805–812.
2	Cedeño-Laurent, J. G., Williams, A., MacNaughton, P., Cao, X., Eitland, E., Spengler, J., & Allen, J. (2021). Associations between acute exposures to PM2.5 and carbon dioxide indoors and cognitive function in office workers: A multicountry longitudinal prospective observational study. *Environmental Research Letters.*
3	Du, B., Tandoc, M. C., Mack, M. L., & Siegel, J. A. (2020). Indoor CO_2 concentrations and cognitive function: A critical review. *Indoor Air.*
4	European Committee for Standardization. *EN 12464-1: Light and lighting — Lighting of work places — Part 1: Indoor work places.*
5	Jahncke, H., Hongisto, V., & Virjonen, P. (2013). Cognitive performance during irrelevant speech: Effects of speech intelligibility and office-task characteristics. *Applied Acoustics.*
6	Jahncke, H., Hygge, S., Halin, N., Green, A. M., & Dimberg, K. (2011). Open-plan office noise: Cognitive performance and restoration. *Journal of Environmental Psychology.*
7	Mølhave, L. (1987). The sick buildings — A sub-population among the problem buildings? *Indoor Air.*
8	Nakaoka, H., Todaka, E., Seto, H., Saito, I., Hanazato, M., Watanabe, M., & Mori, C. (2014). Correlating the symptoms of sick-building syndrome to indoor VOCs concentration levels and odour. *Indoor and Built Environment.*
9	Satish, U., Mendell, M. J., Shekhar, K., Hotchi, T., Sullivan, D., Streufert, S., & Fisk, W. J. (2012). Is CO_2 an indoor pollutant? Direct effects of low-to-moderate CO_2 concentrations on human decision-making performance. *Environmental Health Perspectives.*
10	Seppänen, O., Fisk, W. J., & Lei, Q. H. (2006). *Room temperature and productivity in office work* (LBNL-60952). Lawrence Berkeley National Laboratory.
11	Shehab, M. A., & Pope, F. D. (2019). Effects of short-term exposure to particulate matter air pollution on cognitive performance. *Scientific Reports.*
12	Zhang, X., Wargocki, P., Lian, Z., & Thyregod, C. (2017). Effects of exposure to carbon dioxide and bioeffluents on perceived air quality, self-assessed acute health symptoms, and cognitive performance. *Indoor Air.*
13	*Short-term exposure to indoor carbon dioxide and cognitive task performance: A systematic review and meta-analysis.* (2023). *Building and Environment.*

#	รายการ
14	World Health Organization. (2021). *WHO global air quality guidelines.*
15	คณะกรรมการสิ่งแวดล้อมแห่งชาติ. (2566). *ประกาศเรื่อง กำหนดมาตรฐานฝุ่นละอองขนาดไม่เกิน 2.5 ไมครอน ในบรรยากาศโดยทั่วไป* (บังคับใช้ 1 มิถุนายน 2566).
16	กรมควบคุมมลพิษ. (2566). *ประกาศดัชนีคุณภาพอากาศของประเทศไทย พ.ศ. 2566.*

ก่อนส่งรายงานฉบับจริง

1. **เปิดเปเปอร์ต้นฉบับอ่านเองอย่างน้อยรายการที่ 1, 6, 9 และ 10** — เป็นงานที่ใช้อ้างเกณฑ์โดยตรง และกรรมการมักถามรายละเอียดระเบียบวิธี
2. **เติมเลขเล่ม เลขฉบับ เลขหน้า และ DOI ให้ครบ** — รายการที่ยังไม่มีข้อมูลครบ (2, 3, 5, 6, 7, 8, 9, 11, 12, 13) ต้องเปิดจากวารสารต้นทางเพื่อเติม
3. **ตรวจรูปแบบให้ตรงกับที่สถาบันกำหนด** — บทนี้ใช้ APA 7 หากสถาบันใช้ IEEE ต้องแปลงเป็นเลขในวงเล็บเหลี่ยมและเรียงตามลำดับการอ้างถึง

งบประมาณ

> **สำคัญ: ตัวเลขในหน้านี้ใช้อ้างอิงเพื่อวางแผนเท่านั้น ไม่ใช่ใบเสนอราคา** ราคาชิ้นส่วนอิเล็กทรอนิกส์เปลี่ยนบ่อยมาก **ต้องเช็คราคาจริงก่อนสั่งซื้อทุกครั้ง** ราคาที่ทำเครื่องหมาย **[ยืนยันแล้ว]** ตรวจสอบจากหน้าเว็บร้านเมื่อ 31 กรกฎาคม 2569 ที่เหลือเป็น **[ประมาณการ]** จากช่วงราคาตลาดทั่วไป — อาจคลาดเคลื่อนได้มาก

1. ขอบเขตของงบนี้

ชุด Raspberry Pi 5 มีอยู่แล้ว — ไม่นับรวมในงบ

มีอยู่แล้ว (ไม่ต้องซื้อ)	มูลค่าโดยประมาณ
Raspberry Pi 5 Model B · PSU 27W · Active Cooler · microSD · เคส	~6,200–7,000 บาท

งบในหน้านี้จึงครอบคลุมเฉพาะ **เซนเซอร์ · กล้อง · อุปกรณ์ประกอบ** ที่ยังต้องซื้อเพิ่ม

> NOTE มูลค่าชุดที่มีอยู่แล้วยังมีประโยชน์ตอนทำรายงานโครงงาน — ใช้ระบุ "มูลค่ารวมของระบบ" ได้ที่ราว **9,600–12,500 บาท** แม้จ่ายจริงน้อยกว่านั้นมาก

! ปัจจัยที่ทำให้ราคาแกว่งมากที่สุด

ของแท้เทียบกับโมดูลจีน — SCD40 และ SGP30 ต่างกันได้เกินเท่าตัว

	โมดูลจีน (AliExpress/Shopee)	ของแท้ (Adafruit / Sparkfun / Seeed)
ราคา	ปลายล่างของช่วงในตาราง	ปลายบนของช่วง
ความเสี่ยง	! **SGP30 บางล็อตไม่มี LDO** ต่อ 3.3V แล้วชิปพังทันที	มีวงจรครบตามที่ระบุ
เอกสาร	มักไม่มี · schematic ไม่เปิด	มี schematic + ไลบรารีที่ดูแลต่อเนื่อง
คุ้มไหม	คุ้มสำหรับ BH1750 (วงจรง่าย)	**คุ้มสำหรับ SGP30** เพราะพังแล้วกู้ไม่ได้

ข้อสรุป: ประหยัดได้กับ BH1750 และ PMS7003 · **อย่าเสี่ยงกับ SGP30** ถ้าจะซื้อโมดูลจีน ต้องวัดแรงดันที่ขา VDD ของชิปให้ได้ ~1.8V ก่อนเสียบใช้จริง

2. รายการที่ต้องซื้อ

2.1 เซนเซอร์

รายการ	จำเป็น	ราคา (บาท)	สถานะราคา	หมายเหตุ
PMS7003 (PM1.0/2.5/10)	✓	**706**	**[ยืนยันแล้ว]** ArtronShop	ราคารวม VAT · ควรได้ adapter board มาด้วย
SCD40 breakout (CO_2/T/RH)	✓	~700–1,300	[ประมาณการ]	แทน DHT22 ทั้งหมด — วัดได้ 3 ค่าในตัวเดียว
SGP30 breakout (TVOC/eCO_2)	✓	~350–650	[ประมาณการ]	**ต้องเป็น breakout ที่มี LDO** ไม่ใช่ชิปเปล่า
BH1750 GY-302 (lux)	✓	**74**	**[ยืนยันแล้ว]** ArtronShop	ถูกที่สุดในชุด
ไมโครโฟน USB (mini, UAC)	✓	~150–400	[ประมาณการ]	ต้องปิด AGC ได้ · แบบเสียบตรงเพียงพอ
รวมหมวดนี้		**~1,980–3,130**		

2.2 กล้อง (Focus feature)

รายการ	ราคา (บาท)	สถานะราคา	หมายเหตุ
Camera Module 3 (Standard) ← แนะนำ	~1,000–1,500	[ประมาณการ]	**ไม่กินโควตากระแส USB**
สายแปลง CSI 15→22 ขา	~100–250	[ประมาณการ]	**Pi 5 ใช้ขั้ว 22 ขา — สายที่แถมมากับกล้องเป็น 15 ขา**
ทางเลือก: Logitech C270 (USB, 720p)	~700–900	[ประมาณการ]	ถูกกว่า แต่กินโควตา USB และทำให้เลข ALSA ของไมค์เปลี่ยน
ทางเลือก: Logitech C920 (USB, 1080p)	~2,000–2,800	[ประมาณการ]	เกินความจำเป็นสำหรับงานนี้
รวมหมวดนี้ (Camera Module 3)	**~1,100–1,750**		

> ! **อย่าลืมสายแปลง 15→22 ขา** — เป็นของที่คนลืมสั่งบ่อยที่สุด กล้องมาถึงแล้วเสียบกับ Pi 5 ไม่ได้ ต้องรออีกรอบ

2.3 อุปกรณ์ประกอบ

รายการ	ราคา (บาท)	หมายเหตุ
สายจัมเปอร์ female-female	~60–120	สำหรับ I2C และ UART
ตัวเก็บประจุ 100 nF × 5 · 100 µF × 1 · 220 µF × 1	~50–100	**จำเป็น** ไม่ใช่ของเสริม — SCD40 กระชากถึง 205 mA เป็นพัลส์
แผ่นปริ้นอเนกประสงค์ / เบรดบอร์ด	~60–150	
สาย I2C ยาว (สำหรับย้าย SCD40 ออกห่าง Pi)	~50–150	ทำให้ temperature offset เหลือ 1–3 °C แทน 6–10 °C
น็อต/standoff	~50–100	**ห้ามใช้กาวซิลิโคน** ใกล้ SGP30 — ทำให้เซนเซอร์เสื่อมถาวร
รวมหมวดนี้	**~270–620**	

3. สรุปงบตามชุดที่เลือก

ชุด	องค์ประกอบ	ช่วงราคา (บาท)
A — ประหยัด	เซนเซอร์โมดูลจีน · USB webcam C270 แทน CSI	**~2,950–4,650**
B — แนะนำ	เซนเซอร์ · **Camera Module 3** + สายแปลง	**~3,350–5,500**
+ อัปเกรดภายหลัง	NVMe SSD 256GB + M.2 HAT	+1,300–2,100

ทำไมชุด B ถึงแนะนำ: ต่างจากชุด A แค่ราว 400–850 บาท แต่ได้กล้อง CSI ที่**ไม่กินโควตากระแส USB** Pi 5 จำกัดพอร์ต USB ทั้งสี่รวมกันไว้ที่ 1.6 A (เมื่อใช้ PSU 5A) — เมื่อมีไมค์ USB อยู่แล้ว การเพิ่ม USB webcam เข้าไปทำให้เบียดโควตาและอาจหลุดเป็นระยะ

NVMe คุ้มเมื่อไหร่: ถ้าจะรันต่อเนื่อง 24/7 เป็นปี ๆ — microSD จะเสื่อมจากการเขียนซ้ำ (baseline ของ SGP30 เขียนทุกชั่วโมง + log ระบบ) ถ้าเป็นโปรเจกต์ระยะสั้น **ไม่จำเป็น**

4. สิ่งที่ตัดออกไปแล้วและประหยัดได้

รายการที่ตัด	ประหยัด (บาท)	เหตุผล
MQ-2	~80–150	PMS7003 + SGP30 ครอบคลุมกว่าและแม่นกว่า
ADS1115	~150–350	ไม่ต้องใช้แล้วเมื่อไม่มีเซนเซอร์ analog
ตัวต้านทาน 1% สำหรับ voltage divider	~20–50	ไม่ต้องใช้แล้ว
DHT22	~150–250	SCD40 วัดอุณหภูมิและความชื้นได้แม่นกว่า
รวมที่ประหยัดได้	**~400–800**	และลดกระแสราง 5V ลง 150 mA · ลดความร้อนในกล่อง

5. ค่าใช้จ่ายต่อเนื่อง

รายการ	ค่าใช้จ่าย	หมายเหตุ
ค่าไฟ	~26 บาท/เดือน	ที่ ~9 W ต่อเนื่อง 24/7 · คิดที่ ~4 บาท/kWh
Supabase	0 บาท	Free tier เพียงพอสำหรับข้อมูลทุก 5 วินาที จาก 1 อุปกรณ์
Vercel	0 บาท	Hobby plan
Gemini API	0 บาท	Free tier มีโควตาจำกัดต่อวัน — พอสำหรับใช้ส่วนตัว
เปลี่ยน PMS7003	~706 บาท / 3 ปี	MTTF ตาม datasheet คือ ≥3 ปี
เปลี่ยน microSD	~300 บาท / 1–2 ปี	ถ้าไม่ได้บูตจาก NVMe

คำนวณค่าไฟ: 9 W × 24 ชม. × 30 วัน = 6.48 kWh/เดือน → ที่ 4 บาท/kWh ≈ **26 บาท/เดือน** (ค่าไฟจริงขึ้นกับอัตราก้าวหน้าของการไฟฟ้าและปริมาณใช้รวมของบ้าน — ตัวเลขนี้เป็นการประมาณคร่าว ๆ)

6. ร้านค้าที่ตรวจสอบราคา

ร้าน	เว็บไซต์	หมายเหตุ
ArtronShop	artronshop.co.th	ราคาที่แสดงรวม VAT แล้ว · มีสินค้าที่ยืนยันราคาในหน้านี้ 2 รายการ
Arduitronics	arduitronics.com	
ThaiEasyElec	thaieasyelec.com	
Cytron Thailand	th.cytron.io	
Global Byte	globalbyteshop.com	

> `TIP` **เทียบราคาก่อนซื้อ:** BigGo (biggo.co.th) และ Priceza (priceza.com) รวบรวมราคาจากหลายร้าน · โมดูล SCD40/SGP30 ถูกกว่ามากบน Lazada/Shopee แต่ต้องยอมรับความเสี่ยงเรื่องคุณภาพ ตามที่อธิบายในหัวข้อ 1

7. ลำดับการซื้อที่แนะนำ

ไม่ต้องซื้อทุกอย่างพร้อมกัน — ระบบออกแบบให้เพิ่มเซนเซอร์ทีละตัวได้ โค้ดฝั่ง Pi เพิ่มเซนเซอร์ = สร้างไฟล์ใหม่ใน `sensors/` แล้วลงทะเบียนใน config ไม่ต้องรื้อของเดิม

ลำดับ	ซื้ออะไร	ทำอะไรได้	งบก้อนนี้
1	SCD40 + BH1750 + สายจัมเปอร์ + คาปาซิเตอร์	ได้ CO_2 · อุณหภูมิ · ความชื้น · แสง — **ครบพอให้แดชบอร์ดมีความหมายแล้ว**	~880–1,600
2	PMS7003	เพิ่ม PM1.0/2.5/10 — สำคัญมากในไทยช่วงฤดูฝุ่น	~706
3	กล้อง + สายแปลง 15→22 ขา	เปิดใช้ Focus feature (ตรวจจับใบหน้า/สมาธิ)	~1,100–1,750
4	SGP30 + ไมโครโฟน USB	เพิ่ม TVOC และระดับเสียง	~500–1,050

เริ่มจากขั้นที่ 1 ก่อน ด้วยงบไม่ถึง 1,600 บาท ก็ได้ระบบที่ใช้งานได้จริงแล้ว เพราะ SCD40 ตัวเดียวให้ทั้ง CO_2 อุณหภูมิ และความชื้น ซึ่งเป็นสามค่าหลักของแดชบอร์ด
ขั้นที่ 1 + 2 รวม **~1,590–2,300 บาท** ได้ครบทุกค่าด้านคุณภาพอากาศ

สรุปอุปกรณ์ทั้งหมด

ตารางเปรียบเทียบ

อุปกรณ์	วัดค่า (หน่วย)	อินเทอร์เฟซ	ไฟเลี้ยง	กระแส เฉลี่ย / พีค	รอบอ่าน	Warm-up
PMS7003	PM1.0 · PM2.5 · PM10 (μg/m³)	UART 9600 8N1 · `/dev/ttyAMA0`	**5V** (ลอจิก 3.3V)	100 / 150 mA	~2.3 s (push เอง)	**30 วิ**
SCD40	CO_2 (ppm) · T (°C) · RH (%)	I2C `0x62`	**3.3V เท่านั้น**	18 / **205** mA	5 s	~5 วิ
SGP30	TVOC (ppb) · eCO_2 (ppm)	I2C `0x58`	3.3–5V (ผ่าน breakout)	60 / 65 mA	**1.000 s เป๊ะ**	15 วิ · **12 ชม.**
BH1750	ความสว่าง (lx)	I2C `0x23`	3.3V	0.12 / 0.19 mA	120 ms	~1 วิ
USB mic	ระดับเสียง (dB)	USB Audio Class	จาก USB bus	50–100 mA	หน้าต่าง RMS 1 s	ทันที
กล้อง	ใบหน้า · EAR · การขยับ	CSI-2 (แนะนำ) หรือ USB	จากบอร์ด / USB	~250 mA	10–15 fps	ทันที

แผนที่ทรัพยากรบน Raspberry Pi 5

ทรัพยากร	ขา	ใช้โดย
I2C-1 @ 100 kHz	3 (SDA) · 5 (SCL)	SCD40 `0x62` · SGP30 `0x58` · BH1750 `0x23` — **ไม่มีที่อยู่ชนกัน**
UART0 @ 9600	8 (TXD) · 10 (RXD)	PMS7003 — Pi TXD→เซนเซอร์ RX, Pi RXD→เซนเซอร์ TX (**ไขว้กัน**) · SET/RESET *(ทางเลือก)* ที่ขา 11 · 13
ราง 3V3 · 5V	1 · 17 · 2 · 4	3V3 → SCD40 · BH1750 (+ SGP30 breakout) · 5V → PMS7003 (ต่อ VCC ทั้งขา 1+2 และ GND ทั้งขา 3+4)
USB · CSI-2	พอร์ต USB · ขั้ว CAM/DISP	ไมโครโฟน (+ webcam ถ้าไม่ใช้ CSI) · Camera Module 3 **ต้องมีสายแปลง 15→22 ขา**

ขาที่ห้ามใช้: 27–28 (HAT EEPROM) · 12 (สำรองไว้ให้ I2S) · 19/21/23 (SPI0)

งบกำลังไฟ

ราง	3V3	5V	USB	Pi 5	รวม
เฉลี่ย	~78 mA (0.26 W)	100 mA (0.50 W)	150–350 mA	~7 W	**~9 W**
พีค	~270 mA (0.89 W)	150 mA (0.75 W)	~500 mA	~13 W	**~16 W**

พีคของราง 3V3 มาจากพัลส์ IR ของ SCD40 ทุก 5 วินาที · USB มีเพดาน 600 mA และปลดเป็น 1.6 A เมื่อเจอ PSU 5A · **ใช้ PSU 27W (5.1V/5A) ทางการ ตัวเก็บประจุที่ต้องมี:** 100 nF คร่อม VDD–GND ของทุกบอร์ด I2C · 100 μF ที่ปลายสาย 3V3 ฝั่งเซนเซอร์ · 220 μF ที่ขั้ว VCC ของ PMS7003

คำเตือนที่ทำให้ของพังหรือข้อมูลผิด

#	เรื่อง	ผลถ้าทำผิด
1	**SCD40 ต้องจ่าย 3.3V เท่านั้น** — ระดับลอจิก I2C อ้างอิงกับ VDD	จ่าย 5V แล้วบัสเป็น 5V → **GPIO ของ Pi พังถาวร**
2	**SGP30 ตัวชิปใช้ 1.8V** ทนสูงสุด 2.16V — ต้องเป็น breakout ที่มี LDO	ต่อ 3.3V เข้าชิปตรง ๆ → **ชิปพังทันที**
3	**PMS7003 อ่าน Data4/5/6** (สภาพบรรยากาศ) ไม่ใช่ Data1/2/3 (CF=1 สำหรับโรงงาน)	ค่า PM ผิดเมื่อฝุ่นสูง · ไลบรารีสำเร็จรูปหลายตัวหยิบผิด
4	**SGP30 ต้องอ่านทุก 1 วินาทีเป๊ะ** และ `Set_baseline` รับ **(TVOC, eCO_2)** สลับกับที่ `Get_baseline` คืนมา	baseline drift · **ไม่มี error ให้เห็น** ค่ายังดูสมเหตุสมผล
5	**SCD40 ต้องตั้ง temperature offset** หลังประกอบเสร็จในตำแหน่งจริง	อุณหภูมิสูงเกินจริง 5–10 °C **และ RH ผิดตามไปด้วย**
6	**pull-up I2C รวมกันเหลือ ~1.05 kΩ (3.1 mA)** เกินเงื่อนไขทดสอบของ SGP30 (2 mA)	บัสไม่นิ่งเมื่อสายยาว → ถอด pull-up ออกจากบอร์ด 1–2 ตัว

การ calibrate

อุปกรณ์	ต้องทำไหม	ทำอะไร · เก็บผลที่ไหน	ความถี่
SCD40	**ต้องทำ**	temperature offset (เทียบเทอร์โมมิเตอร์หลังอุ่น 30 นาที) → EEPROM · แล้ว FRC ที่อากาศบริสุทธิ์ 420 ppm — **ห้องแอร์ปิดตลอด ควรปิด ASC**	offset ครั้งเดียว · FRC ทุก 6–12 เดือน
SGP30	**ต้องทำ**	บันทึก/กู้คืน baseline อัตโนมัติ → `calibration/baseline.json` · ทิ้งเมื่อเก่าเกิน 7 วัน	ทุก 1 ชม.
USB mic	**ต้องทำ**	หา dB offset เทียบเครื่องวัด → ไฟล์ config · **ห้ามเปลี่ยน gain หลังจากนั้น**	ครั้งเดียว
BH1750	ถ้ามีฝาครอบ	transmission factor (อะคริลิกใส ~0.91) → ไฟล์ config	ครั้งเดียว
กล้อง	ทางเลือก	ปรับ EAR threshold ให้เข้ากับรูปตาแต่ละคน → `config/sensors.js`	ครั้งเดียวต่อคน
PMS7003	**ไม่ต้อง**	ปรับเทียบจากโรงงาน — แค่เป่าลมทำความสะอาดทุก 6–12 เดือน	—

ค่าที่ส่งเข้า `POST /api/ingest`

`pm1` · `pm25` · `pm10` (μg/m³ — PMS7003) · `co2` (ppm) · `temperature` (°C) · `humidity` (% — ทั้งสามจาก SCD40) · `tvoc` (ppb) · `eco2` (ppm — SGP30) · `lux` (lx — BH1750) · `sound_db` (dB — USB mic) · `webcam_json` (jsonb — กล้อง)

กฎสำคัญ: อ่านค่าไม่ได้หรือยัง warm-up อยู่ → ส่ง `null` **ห้ามส่ง `0`** เพราะฝั่งเซิร์ฟเวอร์ใช้ `Number.isFinite()` กรอง ถ้าส่ง 0 จะถูกนับเป็นค่าจริงและทำให้คะแนนสุขภาพห้องเพี้ยน · อัปโหลดทุก 5 วินาทีให้ตรงกับ `CLIENT_POLL_MS_DEFAULT`