AAVC 2026 — สคริปต์พูดพรีเซนต์ (ไทย + ศัพท์เทคนิคอังกฤษ)

ทีม KMUTNB · Autonomous Aerial Vehicle Challenge 2026 คู่กับสไลด์ `AAVC2026_Presentation.html` (14 สไลด์) · เวลารวม ~15 นาที + Q&A 5 นาที *อัปเดต 2026-08-26 ตาม progress ล่าสุด: ผล SITL บน competition envelope (24 ส.ค.), takeover drill บนเครื่องจริง (21 ส.ค.), ไฟลท์จริง 3 เที่ยวของวันที่ 26 ส.ค. — กล้อง decode ได้ในอากาศแล้ว*

วิธีใช้สคริปต์นี้

- **สไลด์เป็นภาษาอังกฤษ, พูดเป็นภาษาไทย** แทรกศัพท์เทคนิคภาษาอังกฤษตามธรรมชาติ (ArUco, geofence, OFFBOARD ฯลฯ) — กรรมการฟังศัพท์เทคนิคออกอยู่แล้ว
- **เลื่อนสไลด์:** ลูกศร → / `Spacebar` ไปหน้า, ← ย้อน, `Home` กลับสไลด์แรก. มุมขวาบนบอกเลขสไลด์
- **จับเวลาต่อสไลด์ ~60 วินาที** (ตัวเลข ⏱ กำกับแต่ละสไลด์). ถ้าเริ่มช้า ให้ตัดสไลด์ 5, 6, 11 ให้สั้นลงก่อน — สไลด์ 2, 7, 9, 10, 12 คือหัวใจคะแนน อย่ารวบ
- **โทนเสียง:** เล่าเป็น "ระบบทำอะไร + ผลลัพธ์ + ความปลอดภัย" ไม่ต้องลงสมการหรือ derivation — ถ้ากรรมการอยากลึก เดี๋ยวถามใน Q&A (มี prep ให้ด้านล่าง)
- **ตัวเลขทุกตัวในสคริปต์นี้มีที่มา** — SITL = `docs/evidence/sweep_overlap_2026-08-24.txt`, takeover = `docs/evidence/G7_takeover_dri` `08-21.txt`, ภาพ decode จริง = `captures/decoded_2026-08-26_flight3/`. ถ้ากรรมการถามว่า "วัดยังไง" ตอบได้ว่ามีไฟล์
- **ถ้าพูดคนเดียว** อ่านรวดเดียวได้เลย. **ถ้าแบ่ง 2 คน** แนะนำ: คนที่ 1 สไลด์ 1–7 (mission + ตัวเครื่อง + vision), คนที่ 2 สไลด์ 8–14 (autonomy + safety + results + ปิด)

สไลด์ 1 — Title: Sky-Field Egg Delivery ⏱ 0:40

สวัสดีครับ/ค่ะ คณะกรรมการทุกท่าน พวกเราทีม **KMUTNB** ครับ

วันนี้เราจะนำเสนอ **โดรน hexacopter ที่บินส่งไข่แบบอัตโนมัติเต็มรูปแบบ** สำหรับโจทย์ AAVC 2026 ครับ

หัวใจของระบบเราคือคำเดียว — **autonomous** ครับ ตั้งแต่ takeoff จนถึง landing โดรนหา landing pad เอง เลือกเส้นทางเอง ลงจอดเอง และปล่อยไข่เอง ภายในเวลา 20 นาที โดยที่มนุษย์ไม่ต้องเข้าไปบังคับเลย

ตัวเครื่องคือ **EFT X6100** ครับ สมองสองชั้นคือ **Pixhawk 6X** เป็น flight controller และ **Raspberry Pi CM4** เป็น companion computer ที่รันสมองภารกิจทั้งหมด

(กด → ไปสไลด์ Mission)

สไลด์ 2 — The Mission ⏱ 1:05 · [Concept Development — 30 คะแนน]

โจทย์คือแบบนี้ครับ ในสนามมี landing pad ทั้งหมด **6 แผ่น** แต่ละแผ่นเป็นสี่เหลี่ยมขาว 1 คูณ 1 เมตร มี **ArUco marker** ขนาด 400 มิลลิเมตรอยู่ตรงกลาง

กรรมการจะ **assign มาให้ทีมเรา 4 แผ่นจาก 6** ครับ อีก 2 แผ่นที่เหลือเป็นตัวลวง — decoy — ที่เราต้อง "ไม่" ไปลง

สำหรับแต่ละ pad ที่ได้รับมอบหมาย โดรนต้องบินผ่าน **corridor** ที่กำหนด หา marker ให้เจอ **ลงจอดบนแผ่น** แล้วปล่อยไข่ของ pad นั้น — ทั้งหมดนี้ภายใน window **20 นาที** เดียว

จุดสำคัญของ scoring คือ **ทำได้เองเต็มคะแนน** ครับ ไม่มีแต้มพิเศษถ้าทำเสร็จเร็ว แต่ได้คะแนนเต็มถ้าทำได้โดย **ไม่มี human intervention** เพราะฉะนั้นเราออกแบบทุกอย่างเพื่อความ "เชื่อถือได้" มากกว่า "เร็ว" ครับ

(ตัวเลขบนสไลด์: 6→4 pad · 20 นาที · full autonomy · เพดาน 20 เมตร)

สไลด์ 3 — Concept of Operation ⏱ 1:10

นี่คือ **flight profile** ของหนึ่งเที่ยวบินครับ อ่านจากซ้ายไปขวา

เริ่มที่ **Launch & Recovery** ไต่ขึ้นไปที่ **20 เมตร** แล้วบินผ่าน transit corridor จุด **P1 P2 P3** — อันนี้เป็น corridor บังคับตามกติกา แต่ละจุดถูกให้คะแนนตอนบินผ่าน ทั้งขาไปและขากลับ

จากนั้นเข้าสู่ **search area** โดรนบินกวาดแบบ **boustrophedon** — คือกวาดไปกลับเหมือนไถนา — เพื่อ decode pad ทุกแผ่นที่มองเห็น

พอเจอ pad ที่ต้องส่ง โดรนจะ **ลงจอดบนแผ่นนั้น** และปล่อยไข่ **หลังจากแตะพื้นแล้วเท่านั้น**

แล้วบินกลับผ่าน corridor เดิม ลงจอดที่ L&R และ **disarm** ครับ ครบหนึ่งรอบ

(ชี้ diagram: เส้นเขียว = เส้นทางบิน, เส้นประ = เพดาน 20 เมตร, จุดเหลือง = pad)

สไลด์ 4 — System Architecture ⏱ 1:05 · [Concept + Engineering]

ระบบเราแบ่งเป็นสองส่วนครับ — **air segment** กับ **ground segment**

บนอากาศ สมองภารกิจทั้งหมดรันบน **CM4** ครับ — **ไม่มี cloud ไม่มี network ไม่มี LLM ในการบิน** เพราะกติกาห้ามใช้ internet และ 4G เด็ดขาด ผิดคือ **disqualify** ทันที ทุกอย่างจึง deterministic รันจบในเครื่อง

Pixhawk 6X คุมการบินและเป็นเจ้าของ failsafe ทั้งหมด, กล้อง **nadir** ตัวเดียวที่มองลงเป็นตัวชี้ว่าจะลงตรงไหน, และมี **TFmini-S lidar** วัดความสูงตอนลงจอด

ส่วนภาคพื้น — **safety pilot** ถือ RC พร้อม takeover ได้ทุกวินาที และ **ground station** ที่เราเอาไว้ "มอนิเตอร์" เฉยๆ ไม่จำเป็นต้องสั่งการตัวโดรนบินเองครับ

สไลด์ 5 — The Aircraft: EFT X6100 ⏱ 0:55 · [Engineering Design — 40 คะแนน]

ตัวเครื่องเป็น hexacopter **6 ใบพัด ขนาด 18 นิ้ว** ครับ

น้ำหนักพร้อมบินรวมไข่ทั้ง 4 ฟอง **ชั่งจริง** อยู่ที่ **7.2 กิโลกรัม** ในขณะที่ MTOW ที่กติกากำหนดคือ **ไม่เกิน 25 กิโลกรัม** — เราบินอยู่ที่แค่ **29 เปอร์เซ็นต์** ของลิมิต มี margin เยอะมากครับ

จุดที่ออกแบบมาเฉพาะสำหรับโจทย์นี้มี 3 อย่าง: - หนึ่ง เราไม่มี RTK ครับ GPS จึงหยาบ เราเลยใช้ **TFmini-S lidar** วัดความสูงตอน descent และตอนแตะพื้น ไม่พึ่ง GPS ในช่วงสำคัญ - สอง กล้อง **global-shutter** ยึดแน่นมองลงตรงๆ ไม่มี gimbal — เราชดเชยมุมเอียงของตัวเครื่องด้วยซอฟต์แวร์แทน ประหยัดน้ำหนักและจุดเสีย และที่สำคัญ **เลนส์ 74.2 องศา กับมุมติดตั้งกล้อง 180 องศา เราวัดจากตัวเครื่องจริงทั้งคู่** ไม่ได้ใช้ค่าจากสเปก — เพราะถ้าค่านี้ผิด การแปลงพิกเซลเป็นพิกัดจะผิดทั้งหมด - สาม ระบบปล่อยไข่ **4 ตัว** ปล่อยแบบ **ทแยงมุม** เพื่อให้จุดศูนย์ถ่วงไม่เสียสมดุลตอนปล่อย

สไลด์ 6 — Propulsion & Power ⏱ 0:55 · [DCOS — power margin]

เรื่องพลังงานครับ ใบพัดทั้ง 6 ตัว hover ตัวเครื่อง 7.2 กิโลได้ที่กระแสประมาณ **31 แอมป์** ซึ่งอยู่ในพิกัดสบายๆ ถ้ามอเตอร์ดับหนึ่งตัว allocator ของ PX4 ยัง **คุมทรงและแรงยกได้ เสียแค่ yaw** ซึ่งนักบินกู้คืนได้

แบตเป็น **6S 17000 มิลลิแอมป์ 10C semi-solid** ครับ ถ้าคิดเป็น usable ที่กันสำรอง 25 เปอร์เซ็นต์ไว้แล้ว จะเหลือใช้ประมาณ **12.75 แอมป์-ชั่วโมง** ในขณะที่หนึ่งเที่ยวส่ง 4 ไข่ใช้แค่ประมาณ **4.5 แอมป์-ชั่วโมง** — ตามแผนจบเที่ยวแบตยังเหลือราวๆ **74 เปอร์เซ็นต์**

ที่สำคัญคือเรามี **energy-reserve policy** ครับ — ถ้าพลังงานไม่พอจบภารกิจ ระบบจะ **ปฏิเสธไม่เริ่มบิน** หรือไม่เริ่ม delivery นั้น เพราะฉะนั้นไม่มีทางที่ไข่จะค้างอยู่บนแบตที่กำลังจะหมดครับ

และเกจแบตของเรา **อ่านแบบ conservative** — เป็นเกจแรงดันล้วน ไม่มี current sensor พอมีโหลดแรงดันจะย้อยและเกจอ่าน "ต่ำกว่าจริง" ดังนั้น floor 30 เปอร์เซ็นต์ที่สั่ง Return จะทำงาน **เร็วกว่า** ไม่ใช่ช้ากว่า — เมื่อวันที่ 26 ที่ผ่านมามันก็ทำงานจริง สั่งแบตที่บินมาแล้ว 3 เที่ยวกลับบ้านเองครับ

สไลด์ 7 — Finding the Pads: Vision ⏱ 1:15 · [Concept — จุดแข็ง, มีภาพจริงยืนยัน]

เรื่อง computer vision ครับ อันนี้เป็นจุดที่เราเลือกอย่างตั้งใจ และ **มีภาพจากการบินจริงมาให้ดู**

ทุก pad มี **ArUco marker** แบบ DICT_4X4_50 เรา **decode มันตรงๆ** ด้วย OpenCV โดยเช็คร่วมกันสองอย่าง — **white-pad cue** คือแผ่นขาวที่มีจุดดำตรงกลาง กับ **fiducial decode** ที่ยืนยันทั้งตำแหน่งและ **id** ว่าเป็น pad หมายเลขอะไร

สิ่งที่เรา **ไม่มี** เลยคือ machine-learning model ในลูปการบินครับ — **ไม่มี deep learning ไม่มี black box** input เดิมให้ผลลัพธ์เดิมทุกครั้ง ตรวจสอบได้ รันไม่กี่มิลลิวินาทีบน CM4

โจทย์ยากคือขนาดครับ — marker 400 มิล ที่ความสูงกวาด 12 เมตร ในภาพมันกว้างแค่ **28 พิกเซล** เราวัดแล้วว่า detector อ่านได้ที่ขนาดนี้ **ทุกตำแหน่งขวางแนวบิน** ออกไปถึง 94 เปอร์เซ็นต์ของครึ่ง swath — ดังนั้นไม่มีระยะห่างของแนวกวาดไหนที่จะเอา pad ไปวางตรงที่กล้องอ่านไม่ได้

และบทเรียนใหญ่จากสนามซ้อมครับ — **กล้องต้องวัดแสงที่ pad ไม่ใช่ที่หญ้า** auto-exposure ปกติวัดทั้งภาพซึ่งเป็นหญ้า แผ่นขาวเลยสว่างจนขาวโพลน 255 ลายดำหายหมด — เที่ยวหนึ่งเราเก็บ 2,788 เฟรมแล้ว decode ได้ **ศูนย์** ทั้งที่ pad อยู่ในภาพ เราเลยเขียน **highlight-priority AE** ให้คุม exposure จากจุดสว่างสุด 0.2 เปอร์เซ็นต์ของภาพ ให้ขาวอยู่ที่ 190–225 ไม่ clip — บ่ายวันเดียวกัน เที่ยวถัดไป decode ได้ **155 เฟรม 4 id** จากความสูง 8 เมตร ซึ่ง pad สนามซ้อมเล็กกว่าของจริงครึ่งหนึ่ง เท่ากับ **28 พิกเซลเดียวกับที่ KMITL จะให้ที่ 12 เมตร** พอดี

(ชี้ภาพ: 4 เฟรมจากการบินจริง 26 ส.ค. — หญ้ามืด แผ่นขาวชัด กรอบเขียวคือ id ที่ decode ได้: 1, 5, 2, 4)

สไลด์ 8 — Precision Delivery ⏱ 1:00

ตอนส่งของครับ โดรนไม่ได้ทิ้งไข่จากอากาศ — มัน **ลงจอดบนแผ่นจริงๆ** แล้วค่อยปล่อย

วิธีคือ descent เป็น **ขั้นบันไดความสูง** — altitude rung — ปรับให้ marker อยู่กลางภาพทุกขั้น จนแตะพื้น ความคลาดเคลื่อนขั้นสุดท้าย **0.2 เมตร** บนแผ่นขนาด 1 เมตร แล้วช่องไข่จะเปิด **หลังยืนยันว่าแตะพื้นแล้วเท่านั้น** อัตราลงช่วงสุดท้าย **pin ไว้ที่ 0.4 เมตรต่อวินาที** — หนึ่งในสี่ของค่า default ของ PX4 — เป็นค่าที่ทุกการลงจอดที่ validate มาใช้จริง

มี **สองด่านป้องกัน** ครับ — หนึ่ง โดรนจะ **ไม่ยอมลงจอด** ถ้ายังไม่ได้ decode id ที่ถูกต้องระหว่างเข้าใกล้ และสอง **ไม่มีการปล่อยกลางอากาศ** เด็ดขาด pad ผิด id จะไม่มีทางถูกใช้นำการลง

รายละเอียดหนึ่งที่เราเจอจาก log เครื่องจริงครับ — PX4 เวอร์ชัน 1.17 **เขียนความสูง home ใหม่กลางอากาศ** ใน 2 นาทีแรกหลัง takeoff ทุกไฟลท์ของเราโดนบวกลบ 1 ถึง 4.7 เมตร ถ้าปล่อยไว้ ตัวตัดสิน "แตะพื้นแล้ว" จะอ่านผิด เราเลย **latch ค่า home ตอน arm** แล้วไม่ยอมให้มันขยับ

ผลในซิมูเลชันบน envelope ของสนามแข่ง — เราปล่อยได้ห่างจากจุดจริง **0.09 ถึง 0.22 เมตร** ครับ

สไลด์ 9 — Autonomy & Tactics ⏱ 1:05 · [Concept — tactics]

ความท้าทายที่แท้จริงคือ กรรมการให้เรามาแค่ **id ไม่ได้ให้พิกัด** ครับ โดรนต้องหาเอง

เที่ยวบินที่ยังไม่รู้ว่า pad อยู่ไหน จะบิน **sweep แบบ boustrophedon** — บนพื้นที่ค้นหาของ KMITL ขนาด 226 คูณ 73 เมตร คือ **5 แนวที่ความสูง 12 เมตร** — decode **ทุก** pad ที่เห็นเข้าไปเก็บใน registry ที่ key ด้วย marker id ยืนยันด้วยการเห็นซ้ำหลายครั้ง

รายละเอียดที่ทำให้กล้องใช้ได้จริงคือ โดรน **ถือหัวทิศเดียวตลอดการกวาด** ให้ด้านกว้าง 1280 พิกเซลของภาพวางขวางแนวบิน — ถ้าปล่อยให้หันหัวตาม waypoint กล้องที่ยึดกับลำตัวจะหมุนตาม เราเคยเจอมาแล้ว ระยะห่างระหว่างแนวเราก็ไม่ได้เดา — **บินเทียบกัน** ใน SITL ที่ optics ของ KMITL ระหว่าง overlap 0.30 กับ 0.44: เจอ pad 6 จาก 6 ส่ง 4 จาก 4 **เท่ากัน** แต่ 0.30 เร็วกว่า **111 วินาที**

พอเจอครบ 4 โดรนจะเสิร์ฟตาม **เส้นทางสั้นที่สุด** — เป็นการ optimize จริงๆ ไม่ใช่ลำดับที่พิมพ์เข้าไป ในสนามทดสอบของเรา มันลดระยะจาก **429 เมตรเหลือ 266 เมตร** ครับ และแผนถูก **สร้างใหม่สดๆ ทุกเที่ยว** — เที่ยวหลังที่รู้ตำแหน่ง pad แล้ว จะบิน **ตรง** ไม่ต้องกวาดซ้ำ

สไลด์ 10 — Safety & Redundancy ⏱ 1:15 · [DCOS — redundancy + emergency — จุดแข็งจริง เน้นตรงนี้]

เรื่องความปลอดภัยครับ อันนี้เราภูมิใจที่สุด — เรามี **3 ชั้นที่เป็นอิสระต่อกัน** และ **นักบินชนะเสมอ**

ชั้นที่ 1 — safety pilot ถือ RC สามารถ takeover ได้ทุกวินาที ทันทีที่ระบบเห็นว่านักบินเข้าคุม companion จะ **stand down** หยุดสั่งการทันที — และอันนี้ **ซ้อมบนเครื่องจริงแล้ว** เมื่อ 21 สิงหา: ตัวจับทั้งสองตัวทำงานภายใน 1 วินาที และคำสั่งทุกเส้นทางถูกปฏิเสธหมด

ชั้นที่ 2 — flight controller PX4 เป็นเจ้าของ failsafe: หลุด geofence สั่ง **Return** — ค่านี้เรา **ตรวจแล้วว่าที่เก็บในบอร์ดคือ Return จริง** ไม่ใช่แค่เขียนลงไป, ขาด datalink สั่ง Return, ขาด RC สั่ง Return, แบตต่ำ Return แล้ว Land และมี vertical fence 50 เมตรไว้เป็นตาข่ายกันหลุดใหญ่

ชั้นที่ 3 — companion watchdog เป็นตัวตรวจอิสระอีกชั้น เช็คเพดาน 20 เมตร พื้น 10 เมตร no-fly zone แบต GPS และความสดของ telemetry — และถ้า FC สั่ง Return หรือ Land เอง companion จะ **ถอยให้** ไม่ไปแย่งคุม

ทำไมเพดาน 20 เมตรถึงอยู่ที่ companion ไม่ใช่ที่ fence ของ FC ครับ — เพราะ PX4 1.17 **เลื่อนความสูง home กลางอากาศ** อย่างที่เล่าไป fence ที่อ้าง home จึงกลายเป็นตัว trigger สุ่ม watchdog ของเราวัด AGL จาก home ที่ **latch ตอน arm** แทน

สรุปคือ **envelope protection บังคับใช้จริง ไม่ใช่หวังเอา** — ค่าทั้งหมดถูก **เขียนลง FC และอ่านกลับมายืนยัน** และมี **board check 24 พารามิเตอร์** ก่อนทุกวันบิน และ **emergency mode พร้อมใช้ตลอดเวลา** ตรงตามที่ DCOS กำหนดครับ

สไลด์ 11 — Communication ⏱ 0:55 · [Engineering — comms freq/power]

การสื่อสารครับ เราใช้ลิงก์วิทยุ **NOMAD** แบบ ELRS-MAVLink ตัวเดียวส่งทั้ง telemetry, สถานะเครื่อง และ lidar stream ระหว่างโดรนกับ ground station ครอบคลุมพื้นที่ปฏิบัติการตามที่กติกากำหนด

เพราะกติกาห้าม SIM 4G ระบบทั้งหมดจึง **offline by design** ครับ — ภารกิจรันบน CM4, ground console ดึงข้อมูลจากวิทยุ ดังนั้น WiFi หลุดก็ไม่มีทางทำให้จอค้างหรือเข้าใจผิดว่าโดรนมีปัญหา

เราใช้ย่านความถี่ที่แนะนำ — 920 ถึง 925, 2400 ถึง 2500 และ 5725 ถึง 5850 เมกะเฮิรตซ์ ครับ ภาพจากกล้องถูก **บันทึกบนเครื่องที่ 5 เฟรมต่อวินาที** พร้อมส่งลงมาให้กรรมการติดตามได้ และตามกติกาเรื่อง standby area — เรา **ไม่เสียบแบต ไม่เปิดวิทยุ** จนกว่าจะถึงจุดปล่อยครับ

สไลด์ 12 — Validation & Results ⏱ 1:10

ผลการทดสอบครับ เราทดสอบแบบ **end-to-end ในซิมูเลชันก่อน แล้วสร้างและบินจริง**

ในซิมูเลชันบน **envelope ของสนามแข่ง** — transit 20 เมตร กวาดที่ 12 เมตร marker 28 พิกเซล — **ส่งได้ 4 จาก 4 ถูก id ทุกแผ่น**, ความแม่นในการปล่อย **0.09 ถึง 0.22 เมตร** เทียบกับตำแหน่งจริงที่ survey ไว้, ผ่าน transit ครบ **6 จาก 6 จุด ถูกลำดับทั้งสองทาง**, ใช้เวลา **536 วินาที จาก window 1200 วินาที** — ตรวจโดยตัวตรวจอิสระหลังบินซึ่ง **fail แบบ closed** คือถ้าสงสัยจะตัดสินว่าไม่ผ่านไว้ก่อน

ฝั่งฮาร์ดแวร์ **พิสูจน์แล้วไม่ใช่เดา** ครับ — motor map 6 ตัว, battery gauge, TFmini, เลนส์และมุมกล้องที่วัดจริง, servo ไข่ทั้ง 4 มุม และ geofence ที่ **อัปโหลดและอ่านกลับบนบอร์ดจริง**

เราบินจริงที่สนาม KMUTNB มาแล้ว **5 วันสนาม กว่าสิบเที่ยวบิน** — transit corridor ผ่าน 3 จาก 3 จุด, takeover บน FC จริง, และ **decode pad ได้ 4 id จากในอากาศ** ที่ 8 เมตร

และเรา **อ่าน log ทุกไฟลท์** ครับ — แค่วันที่ 26 วันเดียวเจอและแก้ไป 3 เรื่องที่ซิมูเลชันไม่มีทางโชว์: PX4 เลื่อน home altitude กลางอากาศ, นาฬิกาภารกิจที่ป้อนด้วย GPS timestamp ที่มาห่างๆ, และ auto-exposure ที่วัดหญ้าแทน pad — นี่คือเหตุผลที่เราบินจริงให้มากที่สุดก่อนวันแข่งครับ

สไลด์ 13 — Field Operation ⏱ 0:55 · [Engineering — procedure]

ขั้นตอนหน้างานครับ ก่อนวันบินเรา **อ่านพารามิเตอร์ 24 ตัวจากบอร์ด** — motor map, เกจแบต, fence, แหล่งอ้างอิงความสูง — ให้แน่ใจว่าเครื่องเป็นอย่างที่เราคิด เพราะเราเคยเจอ motor map หายไปเองมาแล้วครั้งหนึ่ง

หน้างาน operator เปิด ground console ยืนยัน id ทั้ง 4 ที่ได้รับมอบหมาย เทียบกับการ์ดของกรรมการ — เราโชว์เป็น **รูป marker จริง** ไม่ใช่แค่ตัวเลข ให้จับคู่ภาพต่อภาพได้เลย ลดโอกาสพิมพ์ผิด

ก่อนบินมี **checklist** เป็นด่าน: GPS ตรงกับ FC, โดรนอยู่ใน geofence, แบต, ลิงก์ และ RC พร้อมครบ จากนั้น **สไลด์เพื่อยืนยัน** การอัปโหลด — ตั้งใจให้เป็นการยืนยัน ไม่ใช่กดพลาด แล้ว safety pilot arm และสลับเข้า **OFFBOARD** จอจะเขียวทันทีที่โดรนรับภารกิจ

สรุปเป็น 2 เส้นทางครับ — **ปกติ:** stage แล้ว checklist แล้วสไลด์ แล้ว arm แล้ว OFFBOARD แล้วบินเอง. **ฉุกเฉิน:** นักบินสลับไป POSCTL companion ถอย FC failsafe รับช่วง เป็นตาข่ายนิรภัย

สไลด์ 14 — Summary + Q&A ⏱ 0:45

สรุปครับ ระบบเราเป็นระบบที่ **เรียบง่าย ตรวจสอบได้ และทำคะแนนตรงจุดที่กติกาให้**

- ด้าน **concept** — เรามี concept of operation ที่ชัดเจน: ส่งตาม id ที่ได้รับ, corridor บังคับ, ค้นหาแบบ blind, ลงจอดบนแผ่นแล้วปล่อย
- ด้าน **engineering** — hexacopter ที่มี margin ทั้งแรงและพลังงาน, วัดความสูงได้โดยไม่ต้อง RTK, envelope protection บังคับใช้บน FC จริง และอ่าน log ทุกไฟลท์เพื่อหาสิ่งที่ซิมูเลชันไม่โชว์
- ด้าน **safety** — 3 ชั้นอิสระ, นักบินชนะเสมอ, และ fail-safe พร้อมทุกวินาที

Deterministic. Offline. Full-auto — ภายใน 20 นาที ครับ

ขอบคุณครับ/ค่ะ ยินดีรับคำถามครับ

Q&A Prep — คำถามที่น่าโดน + คำตอบที่ป้องกันได้

หลักการตอบ: **ตอบตรง สั้น มั่นใจ** ยกผลลัพธ์จริงมาหนุน. ถ้าไม่รู้ ให้บอกว่า “เป็นสิ่งที่เราจะ verify หน้างาน” อย่าเดา. เลี่ยงการลงสมการ — ดึงกลับมาที่ระบบและผลลัพธ์.

1. ทำไมใช้ classical CV ไม่ใช้ deep learning / YOLO?

เพราะเป้าหมายคือ **ArUco marker** ซึ่งเป็น fiducial ที่ถูกออกแบบมาให้ decode ได้แม่นและ deterministic อยู่แล้วครับ deep learning จะเพิ่มความไม่แน่นอน, ต้องใช้ข้อมูลเทรน, และกินทรัพยากรบน CM4 โดยไม่จำเป็น. ทางเราเลือกสิ่งที่ **input เดิมให้ผลเดิมทุกครั้ง ตรวจสอบได้ และรันไม่กี่มิลลิวินาที** ซึ่งเหมาะกับภารกิจที่ต้องการความเชื่อถือได้มากกว่าความยืดหยุ่นครับ

2. ไม่มี RTK แล้วจะลงจอดแม่นได้ยังไง? GPS หยาบไม่ใช่เหรอ?

ถูกต้องครับ GPS เราหยาบ เราเลย **ไม่ให้ GPS คุมเมตรสุดท้าย** — ช่วง descent และแตะพื้นใช้ **กล้อง** จัด marker ให้อยู่กลางภาพ (visual servo) และ **TFmini lidar** วัดความสูง. GPS ใช้แค่พาไปถึงบริเวณ search เท่านั้น ตอนลงจริงเป็น vision ล้วน — ผลในซิมคือปล่อยได้แม่น 0.09 ถึง 0.22 เมตรครับ

3. ถ้ามอเตอร์ดับกลางอากาศจะเป็นยังไง?

ตอบตรงๆ ครับ — hexacopter บนใบพัด 5 ตัวยัง **คุมทรงและ thrust ได้ เสียแค่ yaw** ซึ่งนักบินกู้คืนได้บน POSCTL. แต่เราไม่เคลม automatic motor-out detection เพราะ ESC ของเราเป็น PWM ไม่มีสาย telemetry จึงไม่มีสัญญาณกระแสต่อมอเตอร์ให้ตรวจ. **การป้องกันจริงของเราคือ safety pilot** ที่ takeover ได้ทันที บวกกับ margin แรงขับที่เหลือเยอะ อันนี้เป็นข้อจำกัดที่เรารู้ตัวและเลือกจัดการด้วยคนแทนครับ

4. รับประกันยังไงว่าไข่จะไม่ถูกปล่อยกลางอากาศ หรือปล่อยผิด pad?

มีสองด่านที่ hard-code ไว้ครับ — หนึ่ง **touchdown-gated release**: ช่องไข่เปิดหลังยืนยันแตะพื้นเท่านั้น ถ้า telemetry อ่านว่ายังลอยอยู่ การปล่อยจะถูก **ข้าม**. สอง **id-verified land gate**: โดรนจะไม่ลงถ้ายังไม่ decode id ที่ถูกต้อง. ทั้งสองอย่างถูก **audit** บันทึกทุกครั้ง ตรวจย้อนหลังได้ครับ

5. ไม่มี current sensor แล้วรู้ได้ยังไงว่าแบตเหลือเท่าไหร่? ปลอดภัยไหม?

เราใช้ **voltage-only gauge** ครับ อ่านจากแรงดันเทียบกับ endpoint ของแพ็คจริง (4.18 / 3.77 โวลต์ต่อเซลล์) ที่ **verify บนบอร์ดแล้ว**. ไม่มี current sensor แปลว่าไม่มี load compensation — ใต้โหลดแรงดันย้อย เกจจึงอ่าน "ต่ำกว่าจริง" ราว 30 จุด ซึ่งเป็นทิศทางที่ **ปลอดภัย**: floor สั่ง Return เร็วกว่าไม่ใช่ช้ากว่า วันที่ 26 มันก็ทำงานจริงกับแพ็คที่บินไปแล้ว 3 เที่ยว. บวกกับ **energy-reserve policy** ที่ปฏิเสธไม่เริ่มเที่ยวหรือ delivery ที่พลังงานไม่พอจบ. คำตอบเชิงปฏิบัติคือ **แพ็คชาร์จเต็มทุกเที่ยว** ไม่ใช่ลด floor ครับ

6. ถ้าสัญญาณวิทยุหลุดระหว่างบิน?

ไม่กระทบภารกิจครับ เพราะสมองรันบน **CM4 บนเครื่อง** ไม่ได้พึ่งลิงก์ในการตัดสินใจ. ถ้า datalink หลุด **FC failsafe สั่ง Return** อัตโนมัติ — และ companion ของเรา **รู้ว่า FC สั่ง Return** แล้วถอยให้ ไม่ส่งคำสั่งซ้อน. ลิงก์เราไว้ **มอนิเตอร์** ไม่ใช่ควบคุม ครับ

7. "อัตโนมัติเต็มรูปแบบ" แล้ว operator ทำอะไรบ้าง?

operator ทำแค่ 2 อย่างครับ — **stage** ยืนยัน id 4 ตัวที่กรรมการ assign, และ safety pilot **arm + สลับ OFFBOARD** เพื่อปล่อยบิน. หลังจากนั้น **hands-off** ทั้งหมด โดรนบิน corridor หา pad ลงจอด ปล่อยไข่ กลับบ้าน เอง. สิ่งเดียวที่คนทำระหว่างบินคือ **พร้อม takeover** ถ้าจำเป็นครับ

8. รับประกันยังไงว่าจะไม่บินออกนอกเขต?

สามชั้นครับ — PX4 **geofence action = Return** ที่เรา verify ค่าที่เก็บบนบอร์ดแล้ว, **companion watchdog** ที่เช็ค no-fly zone กับเพดาน 20 เมตรเป็นอิสระอีกชั้น, และ vertical fence ของ FC เองที่ 50 เมตรเป็นตาข่ายใหญ่. ถ้าชั้นใดชั้นหนึ่งพลาด อีกชั้นรับ ครับ

9. delivery แม่นแค่ไหน วัดยังไง?

0.09 ถึง 0.22 เมตรครับ วัดจากจุดปล่อยเทียบกับตำแหน่ง pad จริงที่ survey ด้วยพิกัด ground-truth ในซิมูเลชัน บน envelope ของสนามแข่ง. บนแผ่น 1 เมตร ถือว่าอยู่กลางแผ่นสบายๆ ครับ

10. ทำ 4 ส่งใน 20 นาทีทันจริงเหรอ?

ทันครับ — รันเต็มรูปแบบบน envelope ของสนามแข่งใช้ **536 วินาที ราว 9 นาที** สำหรับส่งครบ 4 ในเที่ยวเดียว. งบเวลาที่เราคิดจากส่วนประกอบจริงสำหรับพื้นที่ KMITL ซึ่งใหญ่กว่าสนามซ้อม อยู่ราว 15 นาที ยังเหลือ margin ราว 5 นาทีเผื่อลมหรือการกวาดซ้ำ. และมี **gate ต่อ delivery** — ถ้าเวลาไม่พอสำหรับไข่ฟองถัดไป ระบบจะไม่เริ่มลง จะกลับบ้านพร้อมไข่ที่เหลือ ไม่ปล่อยให้เกิน window ครับ

11. ถ้าหา pad ไม่เจอล่ะ?

blind sweep ของเรากวาดทั่ว search polygon ครับ pad ขาวที่เห็นแต่ยัง decode id ไม่ได้ จะถูก **บินกลับไปดูซ้ำที่ระดับต่ำ**. ถ้าสุดท้ายยังไม่เจอจริงๆ delivery นั้นจะถูก **ข้ามอย่างปลอดภัย** — เราเลือกที่จะ "ไม่ส่ง" ดีกว่า "ส่งผิดแผ่น" ครับ เพราะลงผิด pad เสียคะแนนมากกว่า

12. ทำไมต้อง 17000 mAh — power margin เป็นยังไง?

hover ที่ ~31 แอมป์ ใช้ได้จริงราว 12.75 แอมป์-ชั่วโมงหลังกันสำรอง 25 เปอร์เซ็นต์ หนึ่งเที่ยว 4 ส่งกินแค่ ~4.5 จบเที่ยวเหลือ ~74 เปอร์เซ็นต์ตามแผนครับ — เผื่อไว้เยอะเพื่อความปลอดภัย ไม่ได้บินจนแบตตึง

13. (ถ้าโดน) ทำไม eggs_aboard = 4 ส่งเที่ยวเดียว ทั้งที่กติกาบอก assign ต่อเที่ยว?

อ้างอิงจาก **event-briefing override วันที่ 24 กรกฎา** ที่กรรมการปรับให้วางครบ 6 pad และ assign 4 ต่อทีมครับ ระบบเรารองรับทั้งสองโหมด — ส่ง 4 เที่ยวเดียว หรือ **fallback เป็น 1 ไข่ต่อเที่ยว** ด้วยการเปลี่ยนค่า integer ตัวเดียว ถ้าหน้างานกรรมการยืนยันเป็นแบบ per-flight เราสลับได้ทันทีครับ

14. อะไรคือจุดที่ทำให้ทีมคุณต่างจากทีมอื่น?

ความเชื่อถือได้ที่พิสูจน์ได้ครับ — เราไม่ได้แค่บอกว่าปลอดภัย เรา **อ่านค่า failsafe กลับจากบอร์ดจริงเพื่อยืนยัน**, เรามีตัวตรวจหลังบินที่ fail แบบ closed, เราอ่าน ULog ทุกไฟล์ท์แล้วเจอบั๊กของ PX4 เองที่ซิมไม่โชว์, และทุกการตัดสินใจใน flight core เป็น deterministic ที่อธิบายได้ทีละบรรทัด ไม่มี black box. ระบบที่ **ตรวจสอบได้** คือระบบที่เชื่อใจให้ถือไข่บินเหนือหัวคนได้ครับ

15. บินภารกิจเต็มรูปแบบบนเครื่องจริงครบหรือยัง? (ถ้าโดน ตอบตรง อย่าเลี่ยง)

ครบทั้งลูปใน SITL ครับ บนเครื่องจริง **แต่ละส่วนพิสูจน์แยกกันแล้ว** — transit corridor ผ่าน 3 จาก 3, takeover ของนักบินทำงานบน FC จริง, กล้อง decode pad ได้ 4 id จากในอากาศเมื่อวันที่ 26, geofence อ่านกลับจากบอร์ด. สิ่งที่ยังไม่ได้ต่อกันในเที่ยวเดียวบนเครื่องจริงคือ **ลงจอดบน pad แล้วปล่อยไข่ต่อจาก sweep** ซึ่งเป็นเป้าของ trial slot วันศุกร์ครับ. เหตุผลที่ช้าไม่ใช่ตัวระบบ แต่เป็นบั๊กที่เจอจาก log จริง 3 เรื่องซึ่งเราแก้แล้วทั้งหมด — และแก้ก่อนวันแข่งดีกว่าเจอในวันแข่งครับ

16. ทำไมก่อนหน้านี้กล้องอ่าน marker ไม่ได้ แล้วมั่นใจได้ยังไงว่าตอนนี้ได้แล้ว?

สาเหตุคือ **exposure** ไม่ใช่ความคม ครับ — auto-exposure ของ driver วัดค่าเฉลี่ยทั้งภาพซึ่งเป็นหญ้า pad ขาวที่สว่างกว่าหญ้า 4-5 เท่าเลยขาวโพลน 255 ขอบดำ-ขาวที่ ArUco ต้องใช้หายไป. เราวัดจากเฟรมจริง: มี pad ในภาพ 2,788 เฟรม decode ได้ 0. แก้ด้วย **highlight-priority AE** ที่วัดจากจุดสว่างสุดของภาพแทน คุมให้ขาวอยู่ที่ 190–225. หลักฐานคือเที่ยวถัดไปบ่ายเดียวกัน decode ได้ 155 เฟรม 4 id ที่ 8 เมตร ซึ่งขนาดพิกเซลเท่ากับ KMITL ที่ 12 เมตรพอดี — **มีภาพจริงในสไลด์** ครับ

17. ความสูงไม่มี RTK เชื่อถือได้ยังไง? ทำไมถึงพูดเรื่อง home altitude?

ความสูงหลักอ้าง **barometer** (EKF2_HGT_REF = 0) ไม่ใช่ GPS ครับ เพราะเราวัดแล้วว่า GPS altitude แกว่ง 10 เมตรได้ในไฟล์ท์เดียว ส่วนเมตรสุดท้ายใช้ **TFmini lidar**. เรื่อง home คือ PX4 1.17 มี feature ที่ "แก้" ความสูง home ตาม GPS ใน 2 นาทีแรกหลังบิน — ออกแบบมาสำหรับ EKF ที่อ้าง GPS แต่กับเราที่อ้าง baro มันกลับสร้าง error ±1-4.7 เมตร เราเจอจาก ULog ทุกไฟล์ท์ตั้งแต่ 21 สิงหา เลย **latch home ตอน arm** แล้วไม่ยอมให้ขยับ และย้ายเพดาน 20 เมตรมาอยู่ที่ companion ครับ

18. ทำไม sweep ที่ 12 เมตร ไม่ลงไปที่ 10 เมตรซึ่งเป็น floor จะได้เห็นชัดกว่า?

เพราะ floor 10 เมตรเป็นเส้นกติกา ถ้าบินชิดเส้น baro แกว่งนิดเดียวก็ละเมิดได้ครับ 12 เมตรให้ margin 2 เมตร และเราวัดแล้วว่าที่ 12 เมตร marker 28 พิกเซล detector อ่านได้ทุกตำแหน่งขวางแนวบิน ก็เลยไม่มีเหตุผลต้องเสี่ยงลงไปอีก ส่วน pad ที่เห็นแต่ยัง decode ไม่ได้ ค่อยบินกลับไปดูซ้ำที่ระดับต่ำเฉพาะจุดครับ

เช็กลิสต์ก่อนขึ้นเวที

- ☐ เปิด `AAVC2026_Presentation.html` แบบ **fullscreen** (กด F11) — ฟอนต์และภาพฝังในไฟล์แล้ว ไม่ต้องต่อเน็ต
- ☐ ทดลองเลื่อน → / ← ให้ชิน ก่อนขึ้นจริง
- ☐ ซ้อมจับเวลา 1 รอบเต็ม — เป้า 14 นาที เหลือ buffer 1 นาที (สไลด์ 7 กับ 12 ยาวขึ้นกว่าเดิม ระวังเวลา)
- ☐ เตรียมตอบ Q&A ข้อ 1, 3, 4, 8, 15, 16 ให้คล่อง (โดนบ่อยสุด — 15 กับ 16 คือของใหม่)
- ☐ มีไฟล์สำรองใน USB เผื่อคอมมีปัญหา + วิดีโอ `captures/real_flight_KMUTNB_2026-08-26_flight3_decoded_realtime.mp4` เผื่อกรรมการอยากดูภาพจริงต่อ