



## รายงานสรุปผลโครงการ

รายวิชา FRA 262 Robotics Studio III : Industrial Automation

สถาบันวิทยาการหุ่นยนต์ภาคสนาม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ชื่อโครงการ หุ่นยนต์พิกัดฉากเพื่อจำลองการเคลื่อนย้ายวัสดุระหว่างสถานที่

### คณะกรรมการ

- |    |                          |                          |
|----|--------------------------|--------------------------|
| 1. | นางสาวฐิติพร สุวรรณวงศ์  | รหัสนักศึกษา 64340500014 |
| 2. | นายธรรมสรณ์ ใจดี         | รหัสนักศึกษา 64340500030 |
| 3. | นายวสุพล เยงศรีสวัช      | รหัสนักศึกษา 64340500049 |
| 4. | นายสรรวัธ บุญยงค์        | รหัสนักศึกษา 64340500051 |
| 5. | นายปภินวิทย์ รัตนศิริ    | รหัสนักศึกษา 64340500065 |
| 6. | นายมนูวิทย์ นาเมืองรักษ์ | รหัสนักศึกษา 64340500077 |

ระยะเวลาโครงการ 5 เดือน (ตั้งแต่ 16 มกราคม 2566 – 12 มิถุนายน 2566)

งบประมาณโครงการ 20,000 บาท

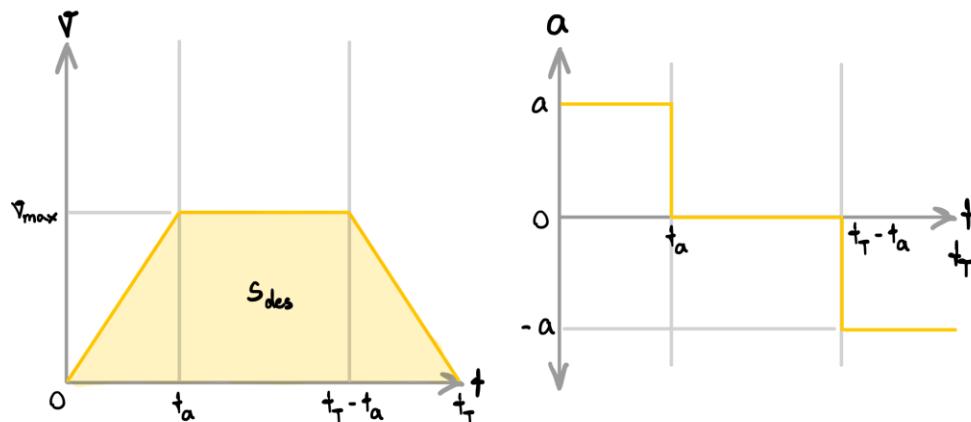
## การออกแบบทางกล (Mechanical Design)

### 1. การคำนวณความเร็วและความเร่งที่ระบบต้องทำ

พิจารณาความเร็วและความเร่งของ End effector ให้สอดคล้องกับความต้องการให้ระบบเคลื่อนที่หยิบและวางของ 9 ชิ้น ระหว่างถ้าดหลุม 9 ช่อง 2 ถาด ภายใน 25 วินาที จะได้ว่าแขนกลจะต้องเคลื่อนที่อย่างน้อย 17 รอบ (เมื่อแต่ละรอบแขนกลเคลื่อนที่ไปในทิศทางเดียว) ซึ่งจะสามารถคำนวณเวลาที่มากที่สุดที่แขนกลต้องเคลื่อนที่ภายในแต่ละรอบได้ว่า

$$t_{round} = \frac{25}{17} = 1.47\text{s}$$

เพื่อให้การเคลื่อนที่ของแขนกลในแต่ละรอบอยู่ภายใน 1.47 วินาที ทางทีมวิศวกรรมเลือกใช้วิถีโคจรแบบสี่เหลี่ยมคงที่ (Trapezoidal Trajectory) มาใช้ในการควบคุมของเคลื่อนที่ของ End effector เนื่องจากเป็นรูปแบบของ วิถีโคจรที่เร็วที่สุดเมื่อเทียบกับวิถีโคจารูปแบบอื่น ๆ จากการกำหนดความเร่งในช่วงต่าง ๆ ให้เป็นค่าคงที่  $a$  โดยหากกำหนดให้ช่วงเวลาทั้งหมดในการเคลื่อนที่ 1 รอบเป็น  $t_T$  ช่วงเวลาที่ใช้ในการเร่งเป็น  $t_a$  และระยะทางที่ต้องการให้เคลื่อนที่เป็น  $s_{des}$  จะทำให้ลักษณะของกราฟความเร็วเทียบกับเวลาเป็นรูปสี่เหลี่ยมคงที่ดังรูป



รูปที่ 1 ความเร็ว (ซ้าย) และความเร่ง (ขวา) ในวิถีโคจรแบบสี่เหลี่ยมคงที่

เมื่อกำหนดให้ระบบสามารถเข้าสู่ Set point ได้ภายใน 0.25 วินาทีจะได้ว่า เวลาที่สามารถเคลื่อนที่ใน 1 รอบจะเหลือ 1.22 วินาที หรือประมาณลงเหลือ 1.2 วินาที พิจารณาเวลาที่ต้องใช้ในการเร่ง End effector ( $t_a$ ) หากกำหนดให้ระบบทำความเร็วเชิงเส้นได้สูงที่สุด ( $v_{max}$ ) 1000 มิลลิเมตรต่อวินาที โดยพิจารณาที่ระยะสูงสุดที่แขนกลต้องเคลื่อนที่ จากพื้นที่ใต้กราฟจะได้ว่า

$$\begin{aligned}
 s_{des} &= \frac{1}{2} v_{max} (2t_T - 2t_a) \\
 t_a &= t_T - \frac{s_{des}}{v_{max}} \\
 &= 1.2 - \frac{700}{1000} \\
 &= 0.5 \text{ s}
 \end{aligned}$$

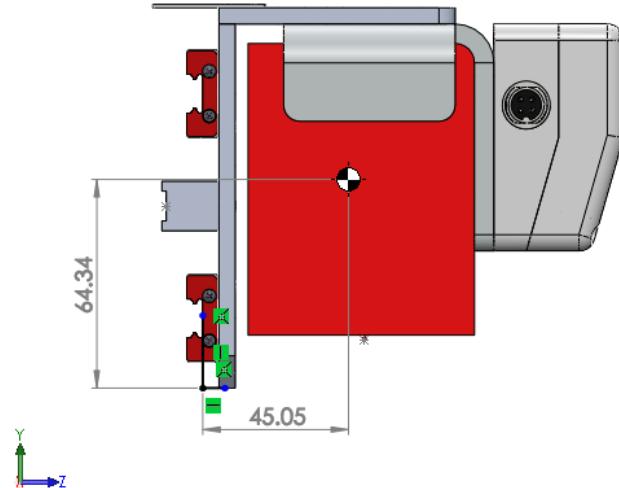
ดังนั้น จากความชันกราฟเวลา – ความเร็วจึงได้ว่า

$$\begin{aligned}
 a &= \frac{v_{max}}{t_a} \\
 &= \frac{1000}{0.5} \\
 &= 2000 \text{ mm/s}^2
 \end{aligned}$$

ในการออกแบบและการวิเคราะห์พลวัตของระบบเพื่อนำไปพิจารณาเลือกใช้อุปกรณ์ต่าง ๆ จึงเลือกใช้ความเร็วของ End effector เป็น 1,000 mm/s และความเร่ง 2,000 mm/s<sup>2</sup>

## 2. การคำนวณโครงสร้างและวัสดุที่ใช้ในการออกแบบ

### 2.1 การคำนวณเพื่อตรวจสอบโมเมนต์ที่เกิดขึ้นกับ Linear Guide



รูปที่ 2 รูปจุดอ้างอิง Center of Mass ของ End Effector

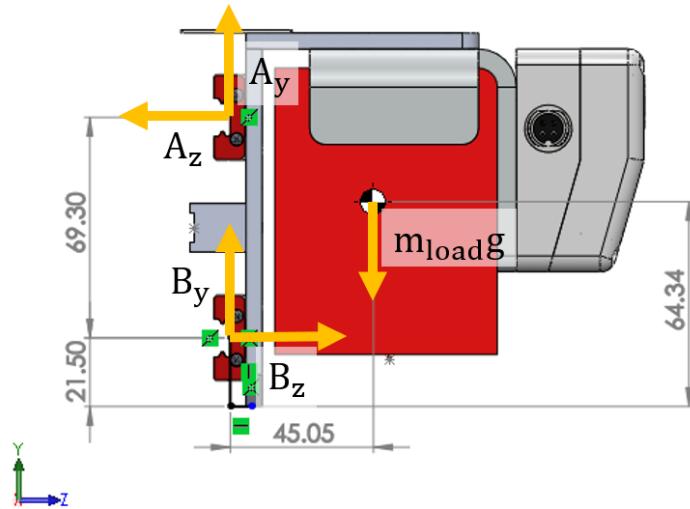
พิจารณามวลรวม End effector โดยจากการประมาณมวลด้วยโปรแกรม SolidWorks จะได้ว่า มวลของ End Effector แบบไม่มีโหลดมีค่าประมาณ 645 กรัม และจากความต้องการของลูกค้าที่ระบบจะต้องรับมวลโหลดได้สูงสุดอีก 1.5 กิโลกรัม จึงออกแบบให้ระบบสามารถรับน้ำหนักได้สูงสุด 2 กิโลกรัม

ดังนั้นมวลรวมของ End effector จึงสามารถคำนวณได้ว่า

$$m_e = 645 + 2000 = 2645 \text{ g} = 2.645 \text{ kg}$$

จะได้ว่า End Effector มีมวลรวม 2.645 กิโลกรัม โดยมีจุด Center of Mass ของ End Effector เทียบจากจุดอ้างอิงดังแสดงในรูปที่ 2

เมื่อพิจารณาแรงที่กระทำกับ End Effector จึงสามารถเขียน Free-Body Diagram ของระบบได้ดังนี้



รูปที่ 3 Free-Body Diagram ของ Load

จาก Free-Body Diagram และเนื่องจากโหลดไม่หมุนในแนวแกน x จึงได้ว่า

$$\Sigma M = 0$$

$$45.05 \cdot 2.645 \cdot 9.81 = 63.9 \cdot A_z$$

$$A_z = 18.29 \text{ N}$$

นอก จาก นี้ ยัง สามารถ พิจารณา  $B_z$  ได้ดังนี้

$$\Sigma F_z = 0$$

$$B_z = A_z$$

$$= 18.29 \text{ N}$$

จะได้ว่า  $A_z$  และ  $B_z$  มีค่าเท่ากับ 18.29 N ซึ่งเป็นแรงคู่คู่ โมเมนต์ที่กระทำกับ Linear Guide จึงสามารถคำนวณได้ดังนี้

$$M_x = 18.29 \cdot 63.9 \cdot 10^{-3}$$

$$= 1.169 \text{ N} \cdot \text{m}$$

ทางทีมจึงเลือกใช้ Linear Guide รุ่น MGN12H เนื่องจากสามารถรับ  $M_x$  ได้สูงสุด 38.2 N·m ซึ่ง MGN12H สามารถรับโมเมนต์ที่เกิดขึ้นได้เพียงพอ

## 2.2 การคำนวณเพื่อเลือกหาสายพานที่เหมาะสม

ในส่วนของการเลือกสายพานที่เหมาะสมสามารถเริ่มได้จากการพิจารณาแรงที่เกิดขึ้นภายในสายพาน (Peripheral Force) โดยเริ่มจากการพิจารณาแรงเสียดทานที่เกิดขึ้นระหว่าง End Effector กับ Linear Guide ซึ่งสามารถทำได้โดยพิจารณาแรงในแนวตั้งเมื่อกำหนดให้ Linear Guide หันคู่ส่วนเข้าไปใน Linear Rail ได้พอดี จะได้ว่า  $A_y = B_y$  ดังนั้น

$$\Sigma F_y = 0$$

$$2 \cdot A_y = 2.645 \cdot 9.81$$

$$A_y = 12.97 \text{ N}$$

ดังนั้น  $A_y$  และ  $B_y$  จึงมีค่าเท่ากับ 12.97 N ซึ่งโดยทั่วไปแล้ว สัมประสิทธิ์แรงเสียดทานของ Linear Guide มีค่าประมาณ 0.004 - 0.006 (อ้างอิง: [https://us.misumi-ec.com/pdf/fa/2012/p1\\_0469.pdf](https://us.misumi-ec.com/pdf/fa/2012/p1_0469.pdf)) ดังนั้นในการคำนวณ จึงกำหนดให้  $\mu_k$  เท่ากับ 0.006 เนื่องจากแรงเสียดทานที่เกิดขึ้นจากแรงปฎิกิริยาตั้งฉากทั้ง 4 เป็นไปในทิศทางเดียวกัน ผลรวมของแรงเสียดทานที่เกิดจากแรงปฎิกิริยาตั้งฉากที่กระทำบน Linear Guide ทั้งสองตัวจะได้ว่า

$$\begin{aligned} f_k &= \mu_k(A_z + A_y + B_z + B_y) \\ &= 0.006(18.29 + 12.97 + 18.29 + 12.97) \\ &= 0.38 \text{ N} \end{aligned}$$

จากแรงเสียดทานที่ได้ เมื่อพิจารณาแรงภายในสายพาน จึงได้ว่า

$$\begin{aligned} \text{แรงที่ทำให้เกิดความเร่ง ; } \quad F_{ua} &= m_e a \\ &= 2.645 \cdot 2 \\ &= 5.29 \text{ N} \end{aligned}$$

$$\text{แรงเสียดทาน ; } \quad F_f = 0.38 \text{ N}$$

$$\begin{aligned} \text{Peripheral Force ; } \quad F_u &= F_{ua} + F_f \\ &= 5.29 + 0.38 \\ &= 5.67 \text{ N} \end{aligned}$$

ดังนั้นแรงที่เกิดขึ้นภายในสายพานจึงมีค่าประมาณ 5.67 N

เนื่องจากเครื่องต้องการความแม่นยำเชิงตำแหน่งสูงแต่ Load ไม่หนักมาก ซึ่งทำให้กำลังส่งที่ใช้มีค่าไม่สูงมาก จึงออกแบบโดยเลือกใช้สายพานที่มีพิธช์สั้น โดยเลือกสายพานรุ่น 2GT ที่มีพิธช์ 2 มิลลิเมตร กว้าง 6 มิลลิเมตร ( $P = 2 \text{ mm}$ ,  $b = 6 \text{ mm}$ )

ในส่วนของการหาจำนวนฟันของรอก สามารถทำได้โดยการคำนวณจากเส้นผ่านศูนย์กลาง โดยจากความต้องการให้ End Effector สามารถเคลื่อนที่ได้ 700 มิลลิเมตร แต่เครื่องมีความยาวได้สูงสุด 1000 มิลลิเมตร เนื่องจากทางทีมวิศวกรต้องการใช้รอกสายพานขนาดเท่ากันทั้งหมดจึงประมาณให้ให้เส้นผ่านศูนย์กลางของรอกมีค่าเท่ากับ 30 มิลลิเมตร จำนวนฟันของรอกจึงสามารถหาได้ดังนี้

$$\begin{aligned} \text{จำนวนฟันของรอก ; } Z_p &= \frac{\pi d_{pulley}}{P} \\ &= \frac{30\pi}{2} \\ &= 47.12 \text{ tooth} \end{aligned}$$

จะได้ว่าจากการคำนวณจะได้ว่าจำนวนฟันของรอกเท่ากับ 47.12 ซึ่งต้องเลือกรอกที่มีจำนวนฟัน 48 ซึ่งจะได้เส้นผ่านศูนย์กลางของรอกสายพานจาก

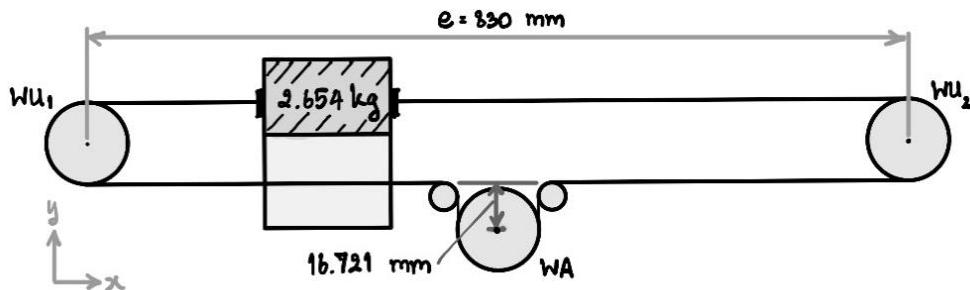
$$\begin{aligned} \text{เส้นผ่านศูนย์กลางของรอก ; } d_{pulley} &= \frac{Z_p P}{\pi} \\ &= \frac{48 \cdot 2}{\pi} \\ &= 30.56 \text{ mm} \end{aligned}$$

จะได้ว่าเส้นผ่านศูนย์กลางของรอกที่ต้องใช้มีจำนวน 48 ซึ่งจากเส้นผ่านศูนย์กลางรอกที่ได้และจากความต้องการให้ End Effector ทำความเร็วได้สูงสุด 1000 มิลลิเมตรต่อวินาที จึงสามารถคำนวณหาความเร็วรอบสูงสุดที่มอเตอร์และรอกขับต้องหมุนได้ ดังนี้

$$\begin{aligned} v &= \omega r \\ 1 &= \omega \cdot \left( \frac{1}{2} \cdot 30.568 \cdot 10^{-3} \right) \\ \omega &= 65.45 \frac{\text{rad}}{\text{s}} \\ &= 65.45 \cdot \frac{60}{2\pi} \approx 625 \text{ rpm} \end{aligned}$$

ตั้งนั้นมอเตอร์และรอกขับจะต้องหมุนที่ความเร็วสูงสุด 625 รอบต่อนาที

เนื่องจากระบบต้องสามารถควบคุมตำแหน่งของ End Effector ได้ End Effector จึงต้องเคลื่อนที่ไป-กลับ เพื่อความสะดวกต่อการออกแบบระบบควบคุม ทางทีมวิศวกรจึงออกแบบให้ระบบสายพานเป็นแบบ Center-Drive เพื่อให้แรงที่ใช้ในการดึงรอกไปทิศ  $+x$  และ  $-x$  เท่ากัน ดังแสดง



รูปที่ 4 ระยะห่างระหว่างรอกของระบบส่งกำลัง

เพื่อให้ End Effector สามารถเคลื่อนที่ได้ 700 มิลลิเมตร ทางทีมวิศวกรจึงออกแบบให้ระยะห่างระหว่างรอกตามทั้งสอง (รอก  $WU_1$  และ  $WU_2$ ) เท่ากับ 830 มิลลิเมตร ซึ่งระยะห่างระหว่างรอกจะนำไปสู่การคำนวณจำนวนฟันของสายพานที่จะนำมาใช้กับรอกทั้ง 3 ตัว จะได้ว่า

$$\begin{aligned} \text{จำนวนฟันของสายพาน ; } Z_b &= \frac{2(e - 0.5d_{pulley} + 16.721)}{P} + (1.5Z_p) \\ &= \frac{2 \cdot (830 - 30.558 + 16.721)}{2} + (1.5 \cdot 48) \\ &= 903.442 \text{ tooth} \end{aligned}$$

จากการคำนวณจะได้ว่าจำนวนฟันของรอกเท่ากับ 903.442 ซึ่งดังนั้นจึงเลือกสายพานที่มีจำนวนฟัน 904 ซึ่ง จากนั้นจึงคำนวณหาความยาวของสายพานเมื่อมีจำนวนฟันทั้งหมด 904 ซึ่งจะได้ว่า

$$\begin{aligned} \text{ความยาวของสายพาน ; } l_o &= Z_b P \\ &= 904 \cdot 2 \\ &= 1808 \text{ mm} \end{aligned}$$

ดังนั้นความยาวของสายพานที่ต้องใช้เท่ากับ 1808 มิลลิเมตร

จากการออกแบบในข้างต้น เมื่อกลับมาพิจารณาแรงภายในสายพาน จากการที่แรงตึงด้านหย่อนจะอยู่ในช่วง 0.1 - 0.3 เท่าของแรงภายในสายพานจะสามารถหาแรงตึงด้านที่หย่อนในสายพานได้ว่า

$$\text{แรงตึงด้านที่หย่อน ; } F_2 = 0.2 \cdot F_u = 0.2 \cdot 5.67 = 1.13 \text{ N}$$

จากการที่ระบบสายพานเป็นแบบ Center-Drive ทำให้  $\frac{l_1}{l_0} = 0.5$  จึงสามารถหาแรงตึงเมื่อสายพานยังไม่หมุนได้จาก

$$\begin{aligned}\text{แรงตึงเมื่อสายพานยังไม่หมุน ; } \quad F_0 &= F_2 + (F_u \frac{l_1}{l_0}) \\ &= 1.13 + (5.67 \cdot 0.5) \\ &= 3.97 \text{ N}\end{aligned}$$

และแรงตึงด้านที่ตึงของสายพานสามารถคำนวณได้ดังนี้

$$\begin{aligned}\text{แรงตึงด้านที่ตึง ; } \quad F_1 &= F_0 + (F_u \frac{l_1}{l_0}) \\ &= 3.97 + (5.67 \cdot 0.5) \\ &= 6.80 \text{ N}\end{aligned}$$

จะได้ว่าแรงตึงเมื่อสายพานยังไม่หมุน แรงตึงด้านที่ตึง และแรงตึงด้านที่หยอดนของสายพานมีค่าเท่ากับ 3.97 N, 6.80 N และ 1.13 N ตามลำดับ

จากแรงที่เกิดขึ้นภายในสายพานจึงสามารถนำมาใช้ในการพิจารณาความหนาแน่นขั้นต่ำของสายพานได้ ซึ่งในการพิจารณาความหนาแน่นขั้นต่ำของสายพานจะพบว่าความเร็วในการหมุนของสายพานมีผลต่อการเสื่อมสภาพของสายพาน ดังนั้นลำดับแรกจึงต้องพิจารณาความถี่ในการหมุนครบรอบของสายพาน ซึ่งจะได้ว่า

$$\begin{aligned}\text{ความถี่ในการหมุนของสายพาน ; } \quad f_R &= \frac{V}{l_o} \\ &= \frac{600}{1808} \\ &= 0.332 \text{ s}^{-1}\end{aligned}$$

ดังนั้นจึงสามารถคำนวณหาค่าสัมประสิทธิ์ความเร็วเพื่อนำไปใช้ในการคำนวณความหนาแน่นขั้นต่ำของสายพาน ได้ดังนี้

$$\begin{aligned}\text{ค่าสัมประสิทธิ์ความเร็ว ; } \quad C_v &= 1 - \frac{50V}{l_o} \\ &= 1 - \frac{50 \cdot 0.6}{1808} \\ &= 0.9834\end{aligned}$$

จะได้ว่าค่าสัมประสิทธิ์ความเร็วสำหรับนำไปคำนวณความหนาแน่นขั้นต่ำของสายพานมีค่า 0.9834

พิจารณาแรงที่สายพาน 2GT สามารถรับได้เมื่อสายพานมีความกว้าง 4 มิลลิเมตร โดยที่อกขับจำนวนฟัน 48 ซี.หมุนที่ความเร็ว 375 รอบต่อนาที ได้จากตารางรูปที่ 5

2GT基準許容張力(Ta)-ベルト幅4mm当り

歯数	12	14	16	18	20	22	24	26	28	30	32	36	40	44	48	50	60	72	単位 N
ピッチ径(mm)	7.64	8.91	10.19	11.46	12.73	14.01	15.28	16.55	17.83	19.10	20.37	22.92	25.46	28.01	30.56	31.83	38.20	45.84	
20	56.09	57.27	58.45	59.63	60.80	61.98	63.16	64.33	65.51	66.69	67.86	70.22	72.57	74.92	77.28	78.45	84.34	84.34	
40	50.60	51.78	52.96	54.13	55.31	56.49	57.66	58.84	60.02	61.20	62.37	64.73	67.08	69.43	71.79	72.96	78.85	78.85	
60	47.39	48.57	49.75	50.92	52.10	53.28	54.45	55.63	56.81	57.98	59.16	61.51	63.87	66.22	68.57	69.75	75.64	75.64	
100	43.35	44.52	45.70	46.88	48.05	49.23	50.41	51.58	52.76	53.94	55.11	57.47	59.82	62.17	64.53	65.70	71.59	71.59	
200	37.85	39.03	40.21	41.38	42.56	43.74	44.92	46.09	47.27	48.45	49.62	51.98	54.33	56.68	59.04	60.21	66.10	66.10	
300	34.64	35.82	37.00	38.17	39.35	40.53	41.70	42.88	44.06	45.23	46.41	48.76	51.12	53.47	55.82	57.00	62.89	62.89	
400	32.36	33.54	34.72	35.89	37.07	38.25	39.42	40.60	41.78	42.95	44.13	46.49	48.84	51.19	53.55	54.72	60.61	60.61	
500	30.60	31.77	32.95	34.13	35.30	36.48	37.66	38.83	40.01	41.19	42.36	44.72	47.07	49.42	51.78	52.96	58.84	58.84	
600	29.15	30.33	31.51	32.68	33.86	35.04	36.21	37.39	38.57	39.74	40.92	43.27	45.63	47.98	50.33	51.51	57.39	57.39	
700	27.93	29.11	30.28	31.46	32.64	33.81	34.99	36.17	37.34	38.52	39.70	42.05	44.41	46.76	49.11	50.29	56.17	56.17	
800	26.87	28.05	29.23	30.40	31.58	32.76	33.93	35.11	36.29	37.46	38.64	40.99	43.35	45.70	48.06	49.23	55.12	55.12	
小	26.21	27.39	28.56	29.74	30.92	32.09	33.27	34.45	35.62	36.80	37.98	40.33	42.68	45.04	47.39	48.57	54.45	54.45	



รูปที่ 5 แรงที่สายพาน 2GT สามารถรับได้เมื่อสายพานมีความกว้าง 4 มิลลิเมตร

(ที่มา: [https://th.c.misumi-ec.com/en/book/UNT1\\_M01\\_SS/digitalcatalog.html?page\\_num=01-46](https://th.c.misumi-ec.com/en/book/UNT1_M01_SS/digitalcatalog.html?page_num=01-46))

จะได้ว่าที่อกจำนวนฟัน 48 ซี.หมุนด้วยความเร็ว 400 rpm (เลือกจากความเร็วที่มากกว่าแต่ใกล้เคียงกับ 375 รอบต่อนาทีมากที่สุด) แรงที่สายพานรุ่น 2GT โดยที่มีพิพิธ์ 2 มิลลิเมตร กว้าง 4 มิลลิเมตรรับได้จะมีค่า  $F_{adm} = 53.55 \text{ N}$  เนื่องจากความสัมพันธ์ระหว่างความกว้างของสายพานกับแรงที่สายพานรับได้เป็นแบบเชิงเส้น

(อ้างอิง: <http://www.esabelt.it/habasync.pdf>) ดังนั้นถ้าความกว้างของสายพานเท่ากับ 6 มิลลิเมตรจะได้ว่า

$$F_{adm} = \frac{53.55}{4} \cdot 6 \\ = 80.325 \text{ N}$$

ความหนาขั้นต่ำของสายพานเมื่อพิจารณาจากแรงดึงที่สายพานรับได้จึงสามารถคำนวณได้จากสูตร

$$b_{req1} = \frac{F_1 b}{F_{adm} \cdot C_v} \\ = \frac{6.80 \cdot 6}{80.325 \cdot 0.9334} \\ = 0.52 \text{ mm}$$

นอกจากความเสียหายที่เกิดขึ้นจากการฉีกขาดของสายพาน ความเสียหายยังสามารถเกิดขึ้นได้บนฟันของรอกที่ขบอยู่กับสายพาน ดังนั้นจึงต้องพิจารณาความหนาขั้นต่ำของสายพานผ่านการพิจารณาจำนวนฟันที่ขบกันระหว่างรอกกับสายพาน (Teeth In Mesh) ควบคู่ไปด้วย ซึ่งจะสามารถพิจารณาจำนวนฟันที่ขบกันระหว่างรอกกับสายพานได้ดังนี้

$$\text{Teeth in Mesh ; } Z_m = \frac{Z_p}{2} \\ = \frac{48}{2} \\ = 24 \text{ tooth}$$

จะได้ว่าที่รอกขับจะมีฟันมากกับสายพานอยู่ที่ 24 ซี่ ซึ่งจะสามารถนำไปเทียบเป็นสัมประสิทธิ์การขบกันระหว่างรอกกับสายพาน ( $t_m$ ) เพื่อนำไปใช้พิจารณาความหนาขั้นต่ำของสายพานได้ดังตารางรูปที่ 6

No. of teeth in mesh $Z_m$	Tooth-in-mesh factor $t_m$
1	0.15
2	0.30
3	0.40
4	0.50
5	0.60
6	0.70
7	0.80
8	0.85
9	0.90
10	0.95
11	0.97
> 11	1.00

รูปที่ 6 ตารางสัมประสิทธิ์การขบกันระหว่างรอกและสายพาน

(ที่มา: <https://www.habasit.com/-/media/Project/Habasit/PublicWebSite/.pdf>)

ดังนั้นหากจำนวนฟันของรอกที่มากกับสายพานอยู่ที่ 24 ซี่ ค่าสัมประสิทธิ์การขบกันระหว่างรอกและสายพานจะมีค่า 1.00 ความหนาขั้นต่ำของสายพานเมื่อพิจารณาจากความแข็งแรงของฟันจึงสามารถคำนวณได้ ดังนี้

$$b_{req2} = \frac{F_u b}{F_{adm} t_m C_v} \\ = \frac{5.67 \cdot 6}{80.325 \cdot 0.9334} \\ = 0.44 \text{ mm}$$

จากการพิจารณาความหนาขั้นต่ำของสายพานจากการพิจารณาความเสียหายที่เกิดขึ้นทั้ง 2 กรณี ได้ว่าความหนาขั้นต่ำที่เกิดจากการพิจารณาแรงภายในสายพานมีค่า 0.52 มิลลิเมตร และความหนาขั้นต่ำที่เกิดจากการพิจารณาแรงที่กระทำกับพื้นของรองสายพานมีค่า 0.44 มิลลิเมตร ดังนั้นสายพานจึงควรมีความหนาต่ำที่สุด 0.52 มิลลิเมตร เพื่อให้สามารถรับแรงที่เกิดขึ้นและหลีกเลี่ยงความเสียหายจากการนีทั้ง 2 ได้ ดังนั้นความหนาในเบื้องต้นที่ได้ประมาณสำหรับการเลือกซื้อไว้ 6 มิลลิเมตรจึงเพียงพอ

ดังนั้นทางทีมวิศวกรจึงเลือกสายพานรุ่น 2GT กว้าง 6 มิลลิเมตร ยาว 1808 มิลลิเมตร โดยใช้รองสายพานสำหรับรุ่น 2GT กว้าง 7 มิลลิเมตร จำนวนพื้น 48 ชิ้น ในการขึ้นรูประบบเพื่อให้สามารถทำงานได้ตรงกับความต้องการของผู้ว่าจ้าง

### 2.3 การคำนวณเพื่อเลือกหมายอเตอร์ที่เหมาะสม

เริ่มพิจารณาจากการคำนวนหาแรงบิดโดยประมาณให้มวลของรองสายพานมีค่า 100 กรัมและกำหนดให้เส้นผ่านศูนย์กลางของรองขั้บมีค่า 30.56 มิลลิเมตรตามการคำนวนที่ได้จากหัวข้อ 1.2 เพื่อใช้ประกอบการเลือกหมายอเตอร์ จากกฎการเคลื่อนที่ของนิวตัน

$$\begin{aligned}\Sigma F &= ma \\ F_1 - F_2 - f &= ma \\ F_1 - F_2 &= ma + f\end{aligned}$$

เมื่อพิจารณาการหมุนโดยมีความเร่งเชิงมุมของเพลาที่เกิดจากการเร่งเชิงเส้น 2000 mm/s<sup>2</sup> จึงได้ว่า

$$\begin{aligned}\Sigma \tau &= J\alpha \\ \tau + (F_2 - F_1)R &= \frac{1}{2}Mr^2 \left(\frac{a}{r}\right) \\ \tau - (ma + f)r &= \frac{1}{2}Mra \\ \tau &= \frac{1}{2}Mra + (ma + f)r \\ \tau &= \left(\frac{1}{2}Ma + ma + f\right)r \\ &= \left(\frac{1}{2}(0.1)(0.5) + 3(2) + 0.44\right)\left(\frac{1}{2} \cdot 30.56 \cdot 10^{-3}\right) \\ &= 0.1 \text{ N} \cdot \text{m}\end{aligned}$$

ดังนั้นมอเตอร์ที่เลือกจึงควรออกแบบบิดได้สูงสุด 0.1 นิวตันเมตร และสามารถวิ่งได้ที่ความเร็ว 625 รอบต่อนาที (จาก การคำนวณในข้อที่ 2.2) ดังนั้นกำลังสูงสุดที่มอเตอร์ควรทำได้จึงสามารถคำนวณได้จากสูตร

$$\begin{aligned} P &= \tau\omega \\ &= 0.1 \cdot 625 \cdot \frac{2\pi}{60} \\ &= 6.545 \text{ W} \end{aligned}$$

เนื่องจากระบบจริง เกิดการสูญเสียพลังงานจากการเสียดสีของเบริ่ง เพลา รอกสายพานและอื่นๆ จึงประมาณ ให้ระบบมีประสิทธิภาพ 50% ดังนั้นกำลังต่ำสุดที่มอเตอร์ควรจะทำได้จึงมีค่า

$$\text{Estimated Minimum Power ; } P_{min,est} = \frac{6.545}{0.5} = 13.09 \text{ W}$$

การเลือกซึ่อมอเตอร์จึงควรเลือกซึ่อไม่ต่ำกว่า 625 rpm และกำลังไม่ต่ำกว่า 13.09 W หรือแรงบิดไม่ต่ำกว่า 0.2 N·m ตามความสัมพันธ์เชิงเส้นระหว่างกำลังและแรงบิด

เมื่อเราได้คำนวณเพื่อหาคุณสมบัติของมอเตอร์ที่ต้องใช้มาแล้ว จึงได้เลือกมอเตอร์ที่มีแปรผักร้าน ใช้ไฟฟ้า กระแสตรง 12V ความเร็ว 1400 RPM และมีกำลังสูงสุด เป็นรุ่น IG42E-04K ซึ่งเป็นรุ่นที่มีคุณสมบัติเพียงพอตามที่ได้ คำนวณไว้ข้างต้น

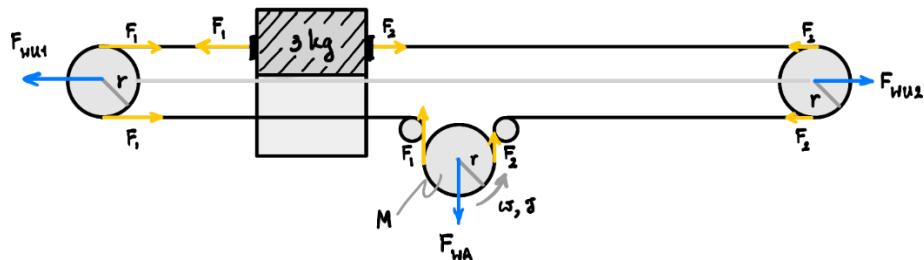


Motor Specification	
Voltage	12VDC
Speed (RPM)	1001 - 2000
Torque (kgf.cm)	2.01 - 3.00
Gear Ratio	4:1
Current	5.5A
Diameter Motor	42mm/45mm
Encoder	Yes

รูปที่ 7 มอเตอร์เกียร์ดิชี 12V 1400 RPM IG42E-04K (ซ้าย) และ Specifications ของ IG42E-04K (ขวา)

#### 1.4 การคำนวณเพื่อหาสัมผ่านศูนย์กลางขั้นต่ำของเพลา

เพื่อพิจารณาแรงต่าง ๆ ที่กระทำกับเพลา สามารถหา Free-Body Diagram ของระบบรอกและเพลาต่าง ๆ ได้ดังนี้



รูปที่ 8 Free-Body Diagram ของระบบส่งกำลัง

จากรูปที่ 8 สามารถคำนวณหาแรงที่กระทำกับเพลารอกตัวขับขณะอยู่นิ่งได้ดังนี้

$$\begin{aligned} F_{WAS} &= 2F_0 \\ &= 2 \cdot 3.97 \\ &= 7.94 \text{ N} \end{aligned}$$

และสามารถคำนวณหาแรงที่กระทำกับเพลารอกตัวขับขณะเคลื่อนที่ได้ว่า

$$\begin{aligned} F_{WAd} &= F_1 + F_2 \\ &= 6.80 + 1.13 \\ &= 7.93 \text{ N} \end{aligned}$$

จะได้ว่าแรงที่กระทำกับเพลารอกตัวขับขณะอยู่นิ่งและเคลื่อนที่มีค่า 2.38 นิวตัน

พิจารณาแรงที่กระทำกับเพลารอกตัวตามทั้งสองขณะอยู่นิ่ง

$$\begin{aligned} F_{WUs} &= 2F_0 \\ &= 2 \cdot 3.97 \\ &= 7.94 \text{ N} \end{aligned}$$

แรงที่กระทำกับเพลารอกตัวตามผิ่งตึงขณะเคลื่อนที่สามารถคำนวณได้จาก

$$\begin{aligned} F_{WU1d} &= 2F_1 \\ &= 2 \cdot 6.80 \\ &= 13.60 \text{ N} \end{aligned}$$

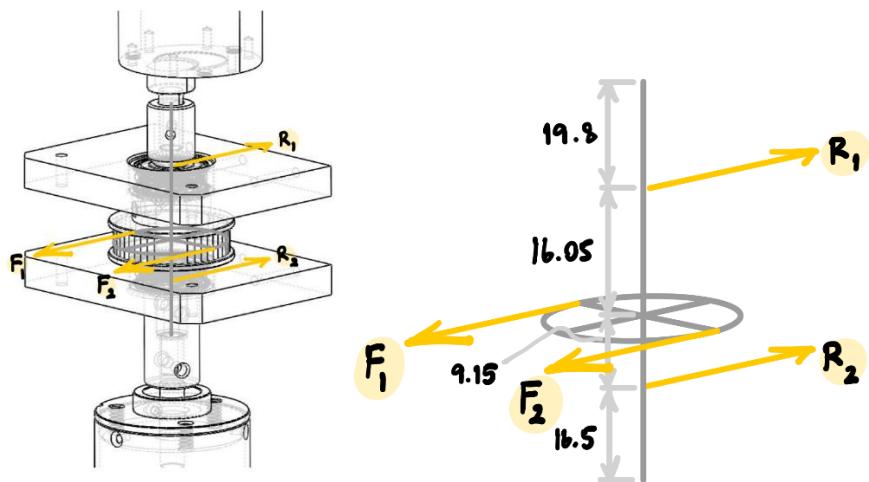
และสามารถคำนวณหาแรงที่กระทำกับเพลา rotor ตัวตามฝั่งหย่อนขณะเคลื่อนที่

$$\begin{aligned}
 F_{WU2d} &= 2F_2 \\
 &= 2 \cdot 1.13 \\
 &= 2.26 \text{ N}
 \end{aligned}$$

เนื่องจากระบบมีความสมมาตรเพลาของ rotor ตามทั้งสองจังจะได้รับแรงสูงสุดเท่ากันคือขณะที่ rotor ของเพลาฝั่งนั้นเป็นฝั่งตึง ซึ่งเพลาจะต้องรับแรงสูงสุด 4.08 นิวตัน

จากแรงสูงสุดที่ rotor สายพานแต่ละตัวได้รับจึงสามารถพิจารณาเส้นผ่านศูนย์กลางขั้นต่ำของเพลา rotor สายพานแต่ละชิ้นได้ดังนี้

#### 1.4.1 เพลา rotor ขับ



รูปที่ 9 Free Body Diagram แรงที่กระทำกับเพลา rotor ขับ

ถึงแม้ว่าแรงลักษณะที่กระทำกับเพลา rotor ขับจะเท่ากันในทั้งขณะที่เพลาอยู่นิ่งและในขณะที่เพลาหมุน แต่จะมีแรงบิด (โมเมนต์/ทอร์ก) กระทำกับเพลาขับมากที่สุดเมื่อเพลาขับหมุน หรือก็คือมีแรง  $F_1$  และ  $F_2$  มากระทำกับเพลาขับ

พิจารณา  $R_1$  จากสมดุลการหมุนโดยให้  $R_2$  เป็นจุดหมุน จะได้ว่า

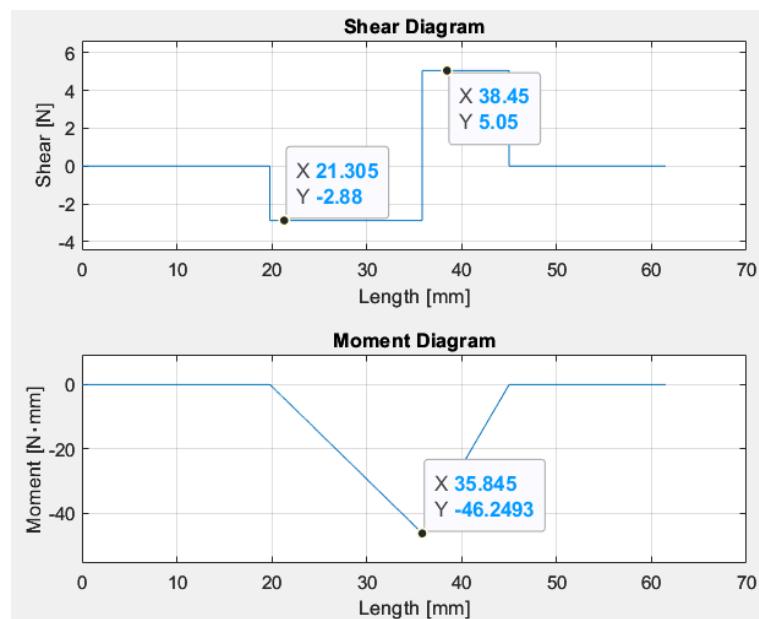
$$\begin{aligned}
 \sum M &= 0 \\
 9.15 \cdot (F_1 + F_2) &= (9.15 + 16.05) \cdot R_1 \\
 R_1 &= \frac{9.15 \cdot (6.80 + 1.13)}{9.15 + 16.05} \\
 &= 2.88 \text{ N}
 \end{aligned}$$

และพิจารณาหา  $R_2$  จากสมดุลแรง

$$\begin{aligned}\sum F &= 0 \\ R_1 + R_2 &= F_1 + F_2 \\ 2.88 + R_2 &= 6.80 + 1.13 \\ R_2 &= 5.05 N\end{aligned}$$

จะได้ว่าโมเมนต์จะมีค่าสูงสุดที่บริเวณที่รอกถูกดึง ดังนั้นจึงสามารถคำนวณโมเมนต์สูงสุดที่กระทำกับเพลาได้จาก

$$\begin{aligned}M &= 16.05 \cdot 10^{-3} \cdot R_1 \\ &= 16.05 \cdot 10^{-3} \cdot 2.88 \\ &= 0.0462 N \cdot m\end{aligned}$$



รูปที่ 10 Shear-Moment Diagram ของเพลารอกขับ

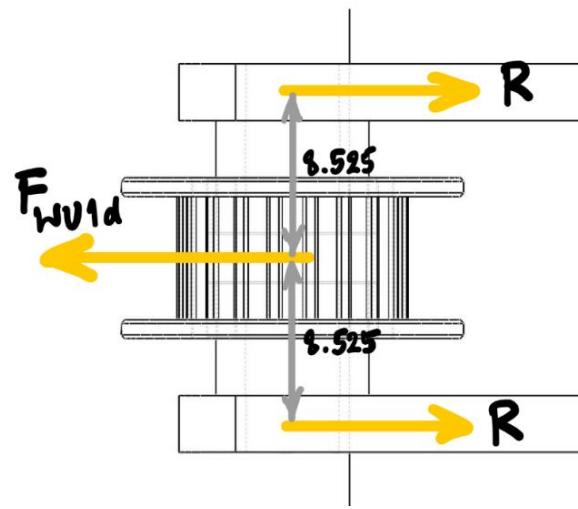
โดยแรงบิดที่มอเตอร์ส่งไปที่รอกผ่านเพลานมีค่าเท่ากับ  $T = 0.2$  นิวตันเมตร ตามการคำนวณจากหัวข้อ 2.3

พิจารณาเส้นผ่านศูนย์กลางขั้นต่ำของเพลารองตาม โดยเพลาที่ใช้มีร่องลิมจะได้ว่า  $\tau_d = 41 \text{ N/mm}^2$  เพลาที่ใช้มีแรงมากจะทำกับเพลาอย่างสมำเสมอจะได้ว่า  $C_t = 1.0$  และ  $C_m = 1.5$  ซึ่งสามารถนำว่าความเส้นผ่านศูนย์กลางของเพลาได้ดังนี้

$$\begin{aligned} d^3 &= \frac{16}{\pi \tau_d} \sqrt{(C_t \cdot T)^2 + (C_m \cdot M)^2} \\ &= \frac{16}{\pi \cdot 41 \cdot 10^6} \sqrt{(0.2)^2 + (1.5 \cdot 0.0462)^2} \\ &= 2.629 \cdot 10^{-8} \\ d &= 2.97 \text{ mm} \end{aligned}$$

จากการคำนวณจึงได้ว่าเพลาของรองข้อจะต้องมีเส้นผ่านศูนย์กลางอย่างน้อย 2.97 มิลลิเมตร ซึ่งเพลาที่ออกแบบไว้ที่ขนาดเส้นผ่านศูนย์กลาง 8 มิลลิเมตร (ที่เลือกไว้เนื่องจากเป็นเส้นผ่านศูนย์กลางที่พอดีกับรอง) จึงเพียงพอ

#### 1.4.2 เพลารองตาม



รูปที่ 11 Free Body Diagram แรงที่กระทำกับเพลารองตาม

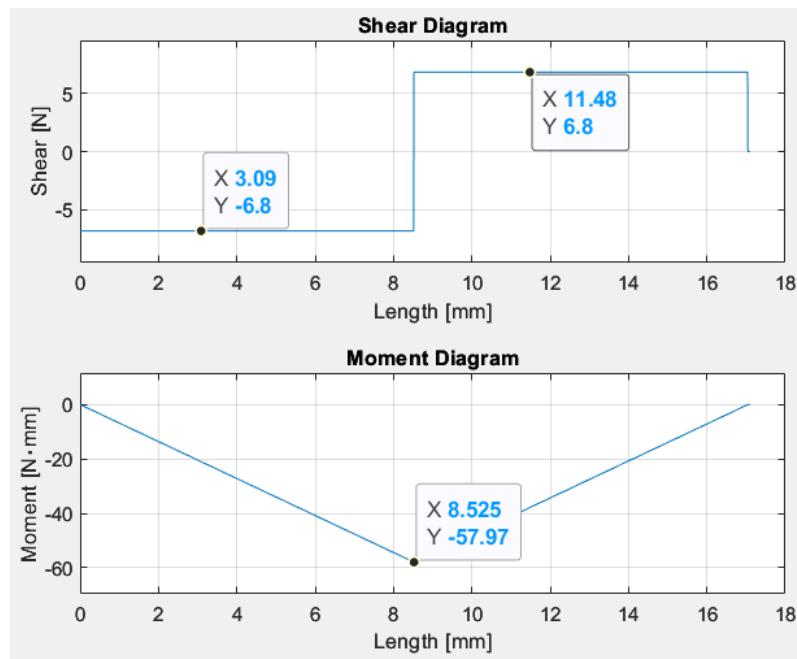
จากรูปที่ 11 เพลาของรอกตามจะสามารถรับแรงสูงสุดเมื่อได้รับแรงดึงจากสายพานฝั่งตึง จึงพิจารณาความแข็งแรง เมื่อแรงที่กระทำกับรอกตัวตามที่ฝั่งตึง ซึ่งมีค่าเท่ากับ 4.08 N

$$\begin{aligned}
 \sum F_x &= 0 \\
 F_{wu1d} &= 2R \\
 R &= \frac{F_{wu1d}}{2} \\
 &= \frac{13.60}{2} \\
 &= 6.80 \text{ N}
 \end{aligned}$$

จะได้โมเมนต์ที่สูงสุดที่ทำให้เพลาถูกดึงโดยสายพาน

$$\begin{aligned}
 M &= 8.525 \cdot 10^{-3} \cdot R \\
 &= 8.525 \cdot 10^{-3} \cdot 6.80 \\
 &= 57.97 \cdot 10^{-3} \text{ N} \cdot m
 \end{aligned}$$

และเนื่องจากไม่มีการส่งกำลังผ่านเพลานี้ ดังนั้น  $T = 0 \text{ N}\cdot\text{m}$



รูปที่ 12 Shear-Moment Diagram ของเพลารอกตาม

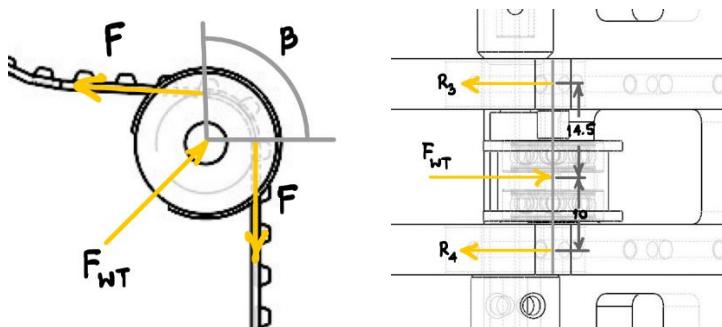
พิจารณาเส้นผ่านศูนย์กลางขั้นต่ำของเพลาอกรตาม โดยเพลาที่ใช้มีร่องลิมจะได้ว่า  $\tau_d = 55 \text{ N/mm}^2$  เพลาที่ใช้มีแรงกระดูกอย่างเบาจากการเริ่มหมุนของสายพานมากรทำเป็นระยะๆ จะได้ว่า  $C_t = 1.2$  และ  $C_m = 1.7$  ซึ่งสามารถนำว่าค่านวนเส้นผ่านศูนย์กลางของเพลาได้จากสูตร

$$\begin{aligned} d^3 &= \frac{16}{\pi \tau_d} \sqrt{(C_t \cdot T)^2 + (C_m \cdot M)^2} \\ &= \frac{16}{\pi \cdot 55 \cdot 10^6} \sqrt{(1.2 \cdot 0)^2 + (1.7 \cdot 57.97 \cdot 10^{-3})^2} \\ &= 9.126 \cdot 10^{-9} \end{aligned}$$

$$d = 2.09 \text{ mm}$$

จากการคำนวณจึงได้ว่าเพลาของรอกตามจะต้องมีเส้นผ่านศูนย์กลางอย่างน้อย 2.09 มิลลิเมตร ดังนั้นการเลือกใช้เพลาเส้นผ่านศูนย์กลาง 5 มิลลิเมตรจึงเพียงพอ

#### 1.4.3 เพลาลูกกลิ้ง



รูปที่ 13 free Body Diagram แรงที่กระทำกับลูกกลิ้ง (ซ้าย) Shear-Moment Diagram ของเพลาลูกกลิ้ง (ขวา)

เนื่องจากเพลาลูกกลิ้งจะรับแรงมากที่สุดเมื่อสายพานผ่านที่ดึงลูกกลิ้งอยู่คือสายพานผ่านตึงดังนั้น  $F = F_1 = 6.80 \text{ N}$  จึงสามารถพิจารณาแรงที่กระทำกับลูกกลิ้งเนื่องจากแรงดึงของสายพานสูงสุด ได้ดังนี้

$$\begin{aligned} F_{WT} &= 2F \sin\left(\frac{\beta}{2}\right) \\ &= 2 \cdot 6.80 \cdot \sin 45^\circ \\ &= 9.62 \text{ N} \end{aligned}$$

พิจารณาแรงที่กระทำกับเพลาลูกกลิ้ง จึงสามารถคำนวณหา  $R_3$  ได้จากสมดุลการหมุน ดังนี้

$$\sum M = 0$$

$$10 \cdot F_{WT} = 24.5 \cdot R_3$$

$$R_3 = \frac{10 \cdot 9.62}{24.5}$$

$$= 3.93 N$$

และสามารถหา  $R_4$  ได้จากสมดุลการเคลื่อนที่เชิงเส้น

$$\sum F = 0$$

$$R_4 = F_{WT} - R_3$$

$$= 9.62 - 3.93$$

$$= 5.69 N$$

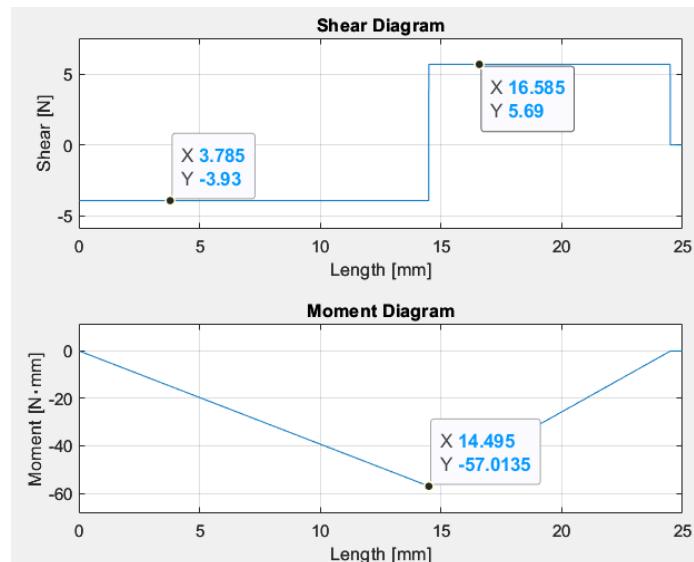
จะได้โมเมนต์ที่สูงสุดที่บริเวณที่รอกถูกดึง ดังนั้นจึงสามารถคำนวณโมเมนต์สูงสุดที่กระทำกับเพลาได้จาก

$$M = 14.5 \cdot 10^{-3} \cdot R_3$$

$$= 14.5 \cdot 10^{-3} \cdot 3.93$$

$$= 57.00 \cdot 10^{-3} N \cdot m$$

และเนื่องจากไม่ได้มีการส่งกำลังผ่านเพลานี้ ดังนั้น  $T = 0 N \cdot m$



รูปที่ 14 Shear-Moment Diagram ของเพลาลูกกลิ้ง

พิจารณาเส้นผ่านศูนย์กลางขั้นต่ำของเพลาอ ก ตาม โดยเพลาที่ใช้มีร่องลิ่มจะได้ว่า  $\tau_d = 41 \text{ N/mm}^2$  เพลาที่ใช้มีแรงกระดูกอย่างเบาจากการเริ่มหมุนของสายพานมีการทำเป็นระยะๆ จะได้ว่า  $C_t = 1.2$  และ  $C_m = 1.7$  ซึ่งสามารถคำนวณเส้นผ่านศูนย์กลางของเพลาได้ดังนี้

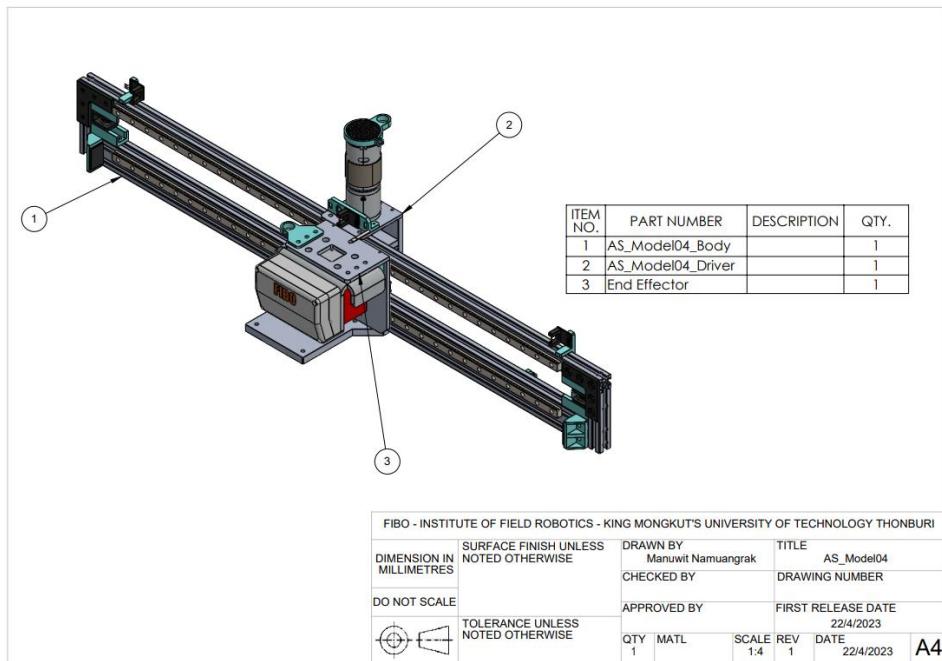
$$\begin{aligned}
 d^3 &= \frac{16}{\pi \tau_d} \sqrt{(C_t \cdot T)^2 + (C_m \cdot M)^2} \\
 &= \frac{16}{\pi \cdot 41 \cdot 10^6} \sqrt{(1.2 \cdot 0)^2 + (1.7 \cdot 57.00 \cdot 10^{-3})^2} \\
 &= 1.204 \cdot 10^{-8} \\
 d &= 2.29 \text{ mm}
 \end{aligned}$$

จึงได้ว่าเพลาของลูกกลิ้งจะต้องมีเส้นผ่านศูนย์กลางอย่างน้อย  $2.29 \text{ มิลลิเมตร}$  ดังนั้นเพลาขนาดเส้นผ่านศูนย์กลาง  $5 \text{ มิลลิเมตร}$  ที่เลือกมาจึงเพียงพอ

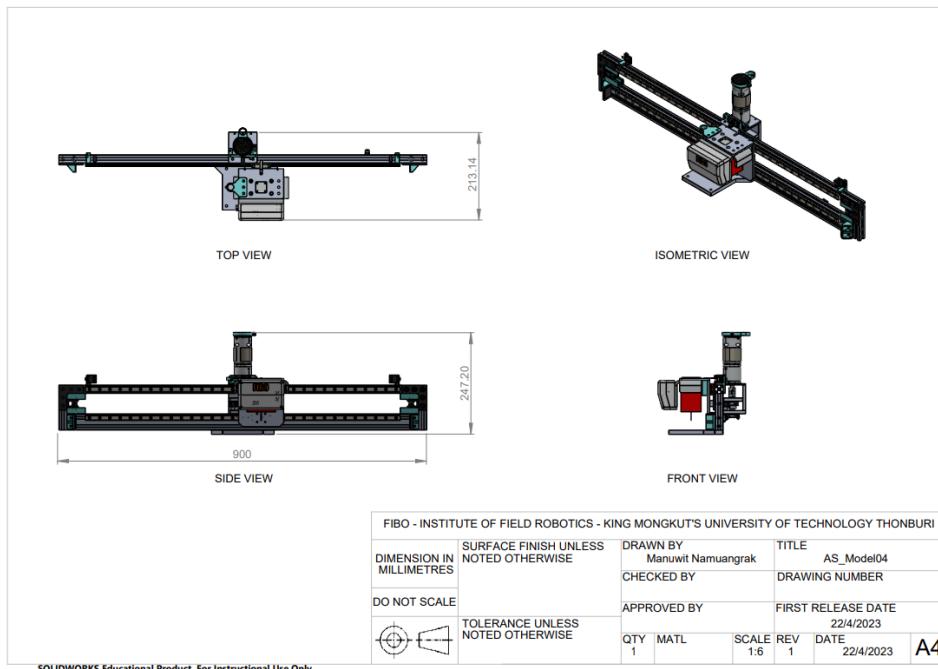
## 2. การออกแบบและการเลือกใช้วัสดุอุปกรณ์ของโครงสร้าง

### 2.1 ภาพรวมของหุ่นยนต์(CAD และ BOM)

หุ่นยนต์ประกอบไปด้วย 3 ส่วนหลักๆ คือ ส่วนของโครงสร้างหลัก(Body) ตัวขับเคลื่อน(Driver) และส่วนของ End Effector ดังรูปที่ 14 ซึ่งมีน้ำหนักอยู่ที่ 4 kg และมีขนาด  $900 \times 213.14 \times 247.2$  mm ดังรูปที่ 15



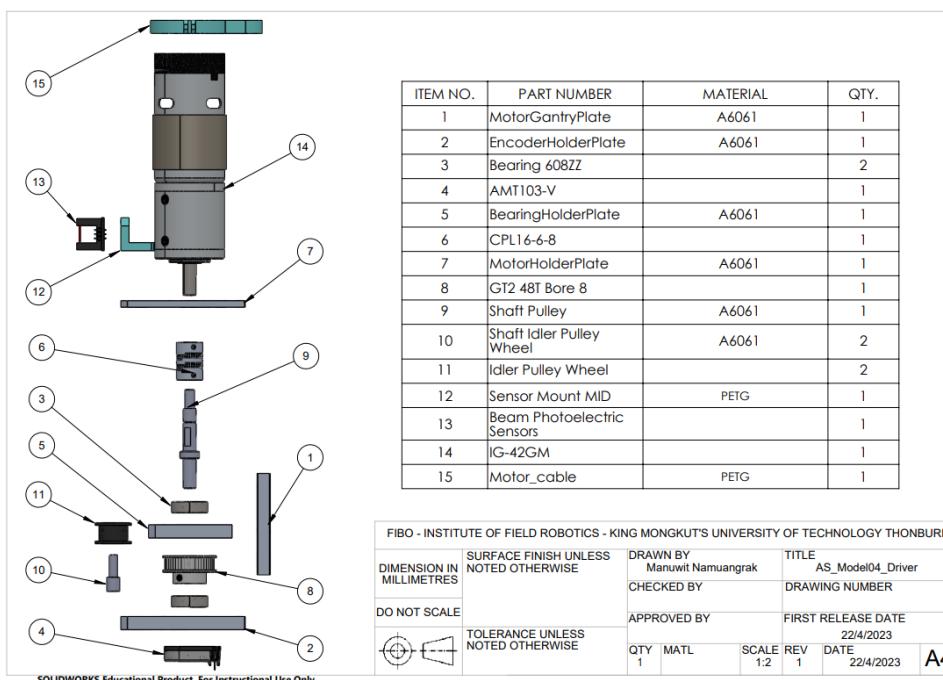
รูปที่ 15 BOM แสดงส่วนประกอบหลักของหุ่นยนต์



รูปที่ 16 รูปแสดง View ต่างๆของหุ่นยนต์ขนาดโดยรวม

### 2.1.1 Driver

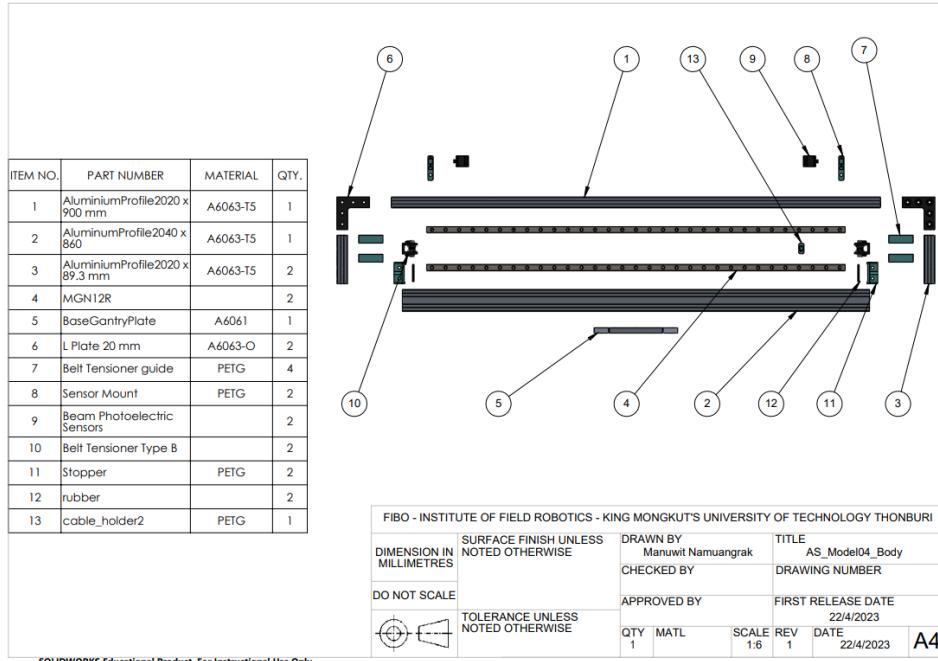
ในส่วนของการขับเคลื่อนจะใช้มอเตอร์ในการขับเคลื่อนโดยใช้ Coupling แบบ Slit ต่อกับเพลาส่งกำลังที่ขับเคลื่อน Pulley โดยมี Bearing สองตัวระหว่าง Pulley และติด Incremental Encoder ไว้ที่ปลายของเพลาส่งกำลัง



รูปที่ 17 รูป BOM แสดงส่วนประกอบต่างๆของตัว Driver

## 2.1.2 Body

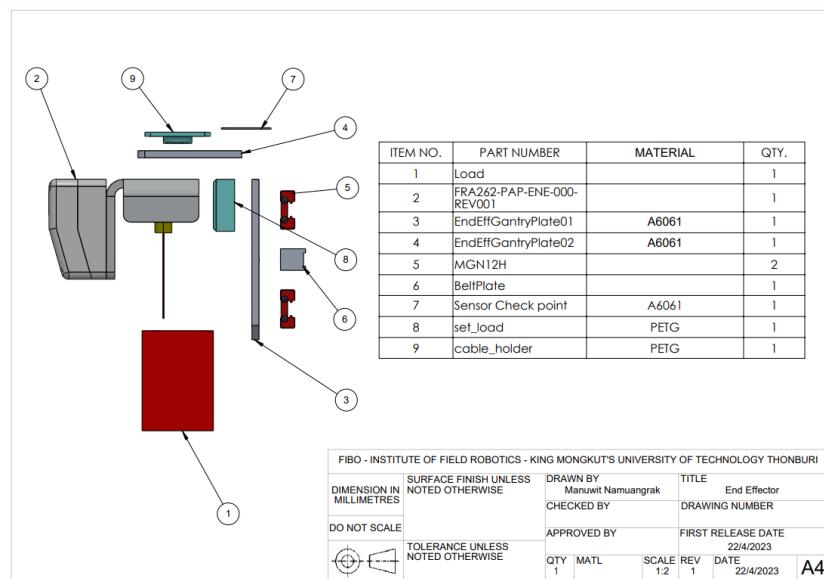
ในส่วนของ body จะมีการติดตั้ง Photo Electric Sensor ไว้ด้านบน และใช้ตัวขับเคลื่อนเป็น Linear Rail Guide 1 คู่ในการขับเคลื่อน โดยมี Belt Tensioner และ Stopper อุปกรณ์ที่จะสองฝั่งของเครื่อง



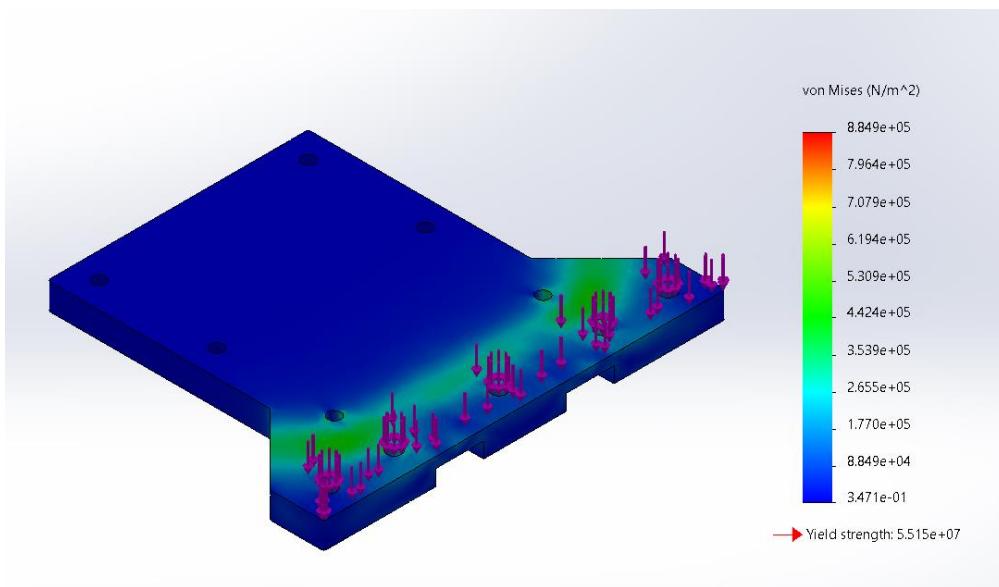
รูปที่ 18 รูป BOM แสดงส่วนประกอบต่างๆของตัว Body

## 2.1.3 End Effector

ในส่วนของ End Effector จะมี Linear Block อุปกรณ์ 1 คู่เพื่อนำไปติดตั้งบน Linear Rail



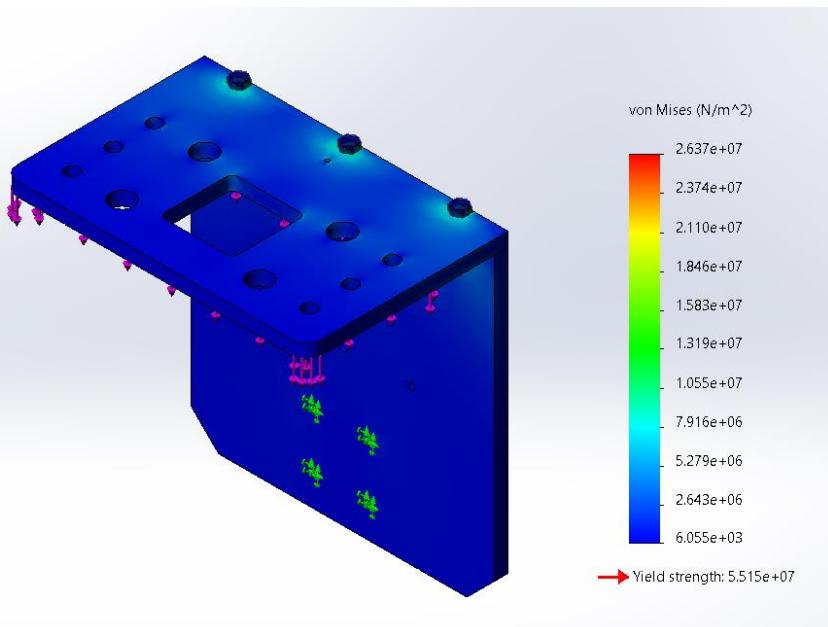
รูปที่ 19 รูป BOM แสดงส่วนประกอบต่างๆของตัว End Effector



รูปที่ 20 FEA ของ BaseGantryPlate

จากการทำ simulation จะสังเกตเห็นว่า Von stress ของชิ้นส่วนไม่เกินค่า Yield Strength ของ Aluminum 6061 หมายความว่าวัตถุจะไม่เกิดการเปลี่ยนรูปถาวร(Deformation)

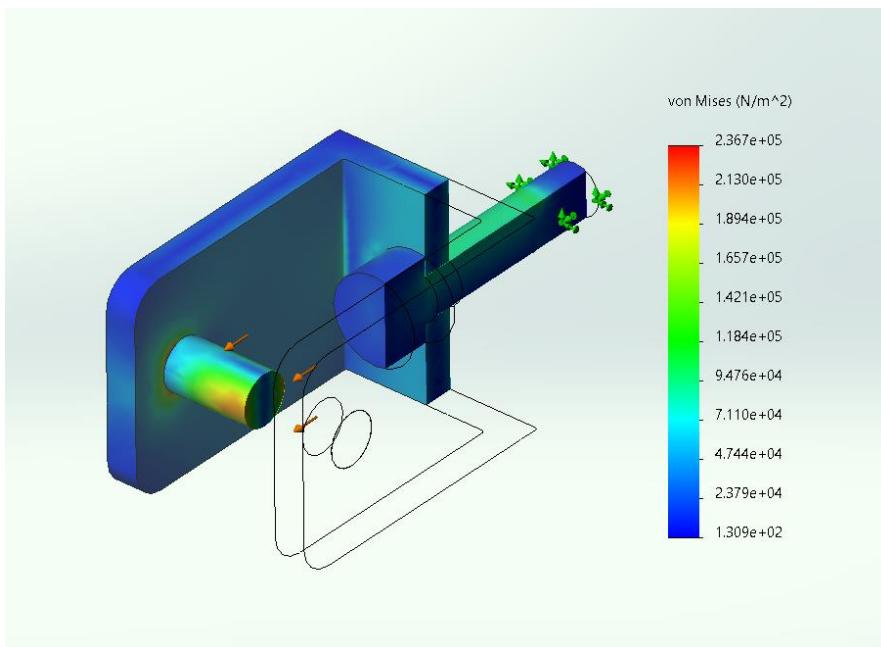
ทางที่มีได้ทำการทดสอบความแข็งแรงของการเชื่อมต่อระหว่าง EndEffGantryPlate01 และ EndEffGantryPlate02 ด้วย bolt ขนาด M3 โดยการทำ Simulation ในโปรแกรม Solidworks โดยทดสอบชิ้นโดยใช้ น้ำหนักของ End Effector และ Load ของทางบริษัท FIBO ซึ่งได้ผลลัพธ์การจำลองดังนี้



รูปที่ 21 FEA ของการยึด Bolt ของ EndEffGantryPlate

จากการทำ simulation จะสังเกตเห็นว่า Von stress ของชิ้นส่วนไม่เกินค่า Yield Strength ของ Aluminum 6061 หมายความว่าวัสดุจะไม่เกิดการเปลี่ยนรูปถาวร (Deformation) และ bolt M3 จะสามารถรับแรงได้เพียงพอ

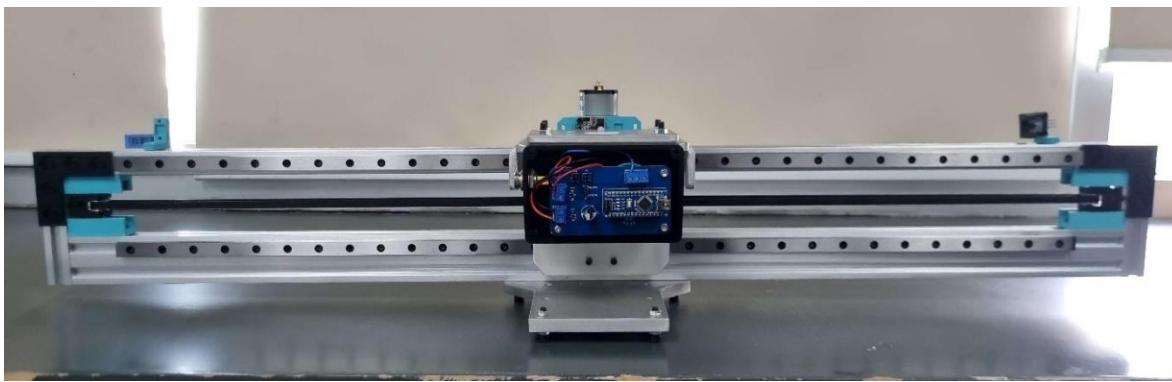
ทางคณะผู้เชี่ยวชาญได้ทำการทดสอบความแข็งแรงของการตึง belt ด้วย Bolt ขนาด M5 โดย Simulation ในโปรแกรม Solidworks โดยทดสอบโดยการใส่แรง  $F_{wud1}$  ที่ได้จากการคำนวณแรงดึงของ Belt ซึ่งได้ผลลัพธ์การจำลองดังนี้



รูปที่ 22 FEA ของ Bolt ของ Tensioner

จากการทำ Simulation จะสังเกตเห็นว่า Von stress ของชิ้นส่วนไม่เกินค่า Yield Strength ของ Bolt หมายความว่า Bolt M5 สามารถรับแรงได้อย่างเหมาะสม

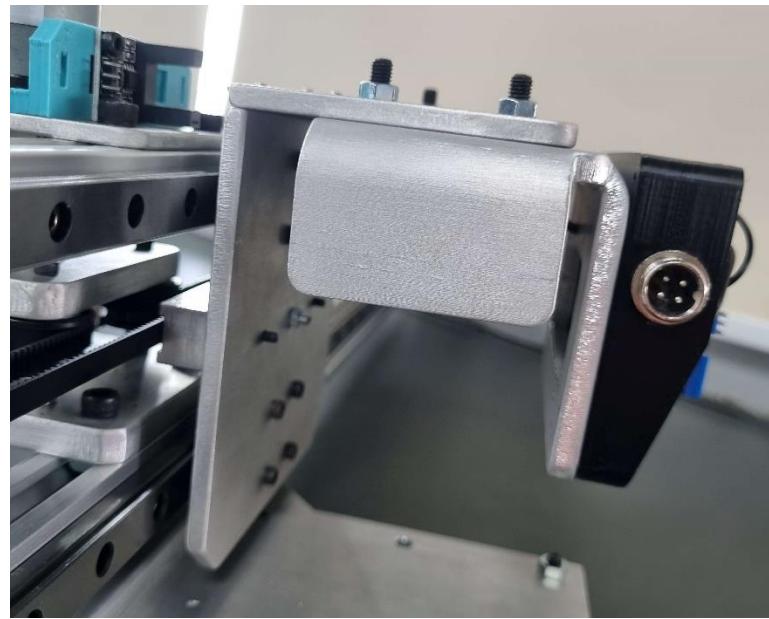
### 3. การประกอบชิ้นงานจริง



รูปที่ 23 ภาพของหุ่นยนต์แขนกลแกน Y ที่ประกอบสำเร็จแล้ว



รูปที่ 24 ภาพ Driver ที่ทำการประกอบสำเร็จแล้ว



รูปที่ 25 ภาพของ EndEffGantryPlate ที่ทำการประกอบกับ End Effector สำเร็จแล้ว

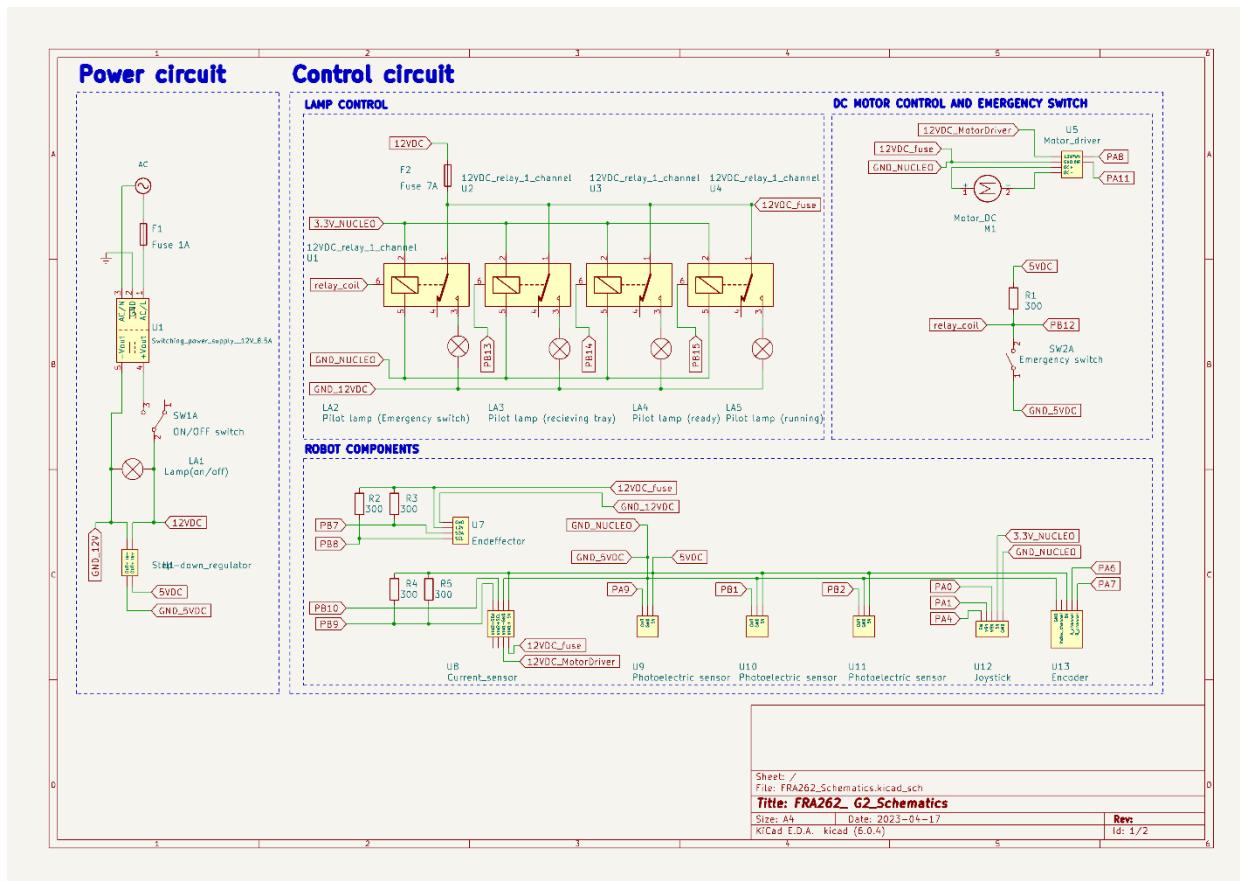


รูปที่ 26 ภาพของสกรูที่ใช้ในการตั้งสายพาน

## ความคืบหน้าของการออกแบบทางไฟฟ้า (Electrical Module)

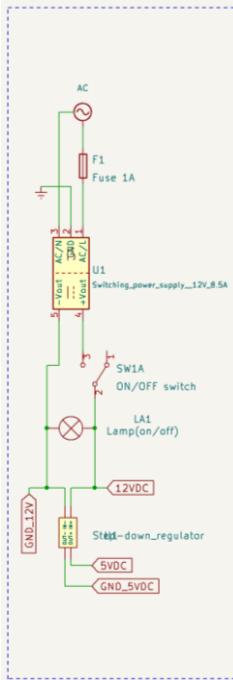
### 1. การออกแบบวงจรไฟฟ้า

ภายในตู้ไฟประกอบไปด้วย 2 วงจร ได้แก่ วงจรกำลัง และวงจรควบคุม ดังรูปที่ 27 โดยวงจรกำลังทำหน้าที่แปลงไฟฟ้ากระแสสลับขนาด 220V ให้เป็นไฟฟ้ากระแสตรงขนาด 12V เพื่อนำไปจ่ายให้กับอุปกรณ์ที่ใช้ไฟฟ้ากระแสตรง 12V และแปลงไฟฟ้ากระแสตรงขนาด 12V ให้เป็นไฟฟ้ากระแสตรงขนาด 5V เพื่อนำไปจ่ายให้กับอุปกรณ์ที่ใช้ไฟฟ้ากระแสตรง 5V ดังรูปที่ 28 ส่วนวงจรควบคุมทำหน้าที่ควบคุมการทำงานของแต่ละอุปกรณ์ในระบบตามโปรแกรมที่ถูกเขียนเอาไว้ ดังรูปที่ 29



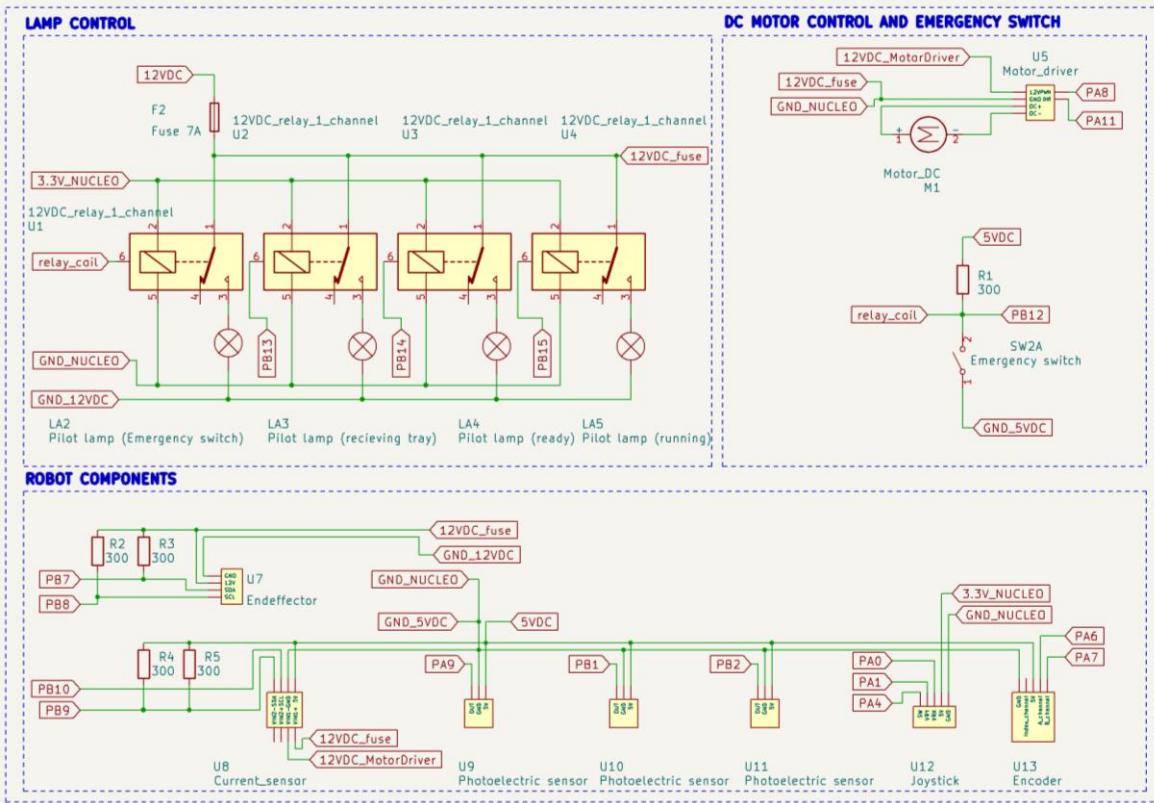
รูปที่ 27 รูปแสดงวงจรไฟฟ้าทั้งระบบ

## Power circuit



รูปที่ 28 รูปแสดงวงจรกำลัง

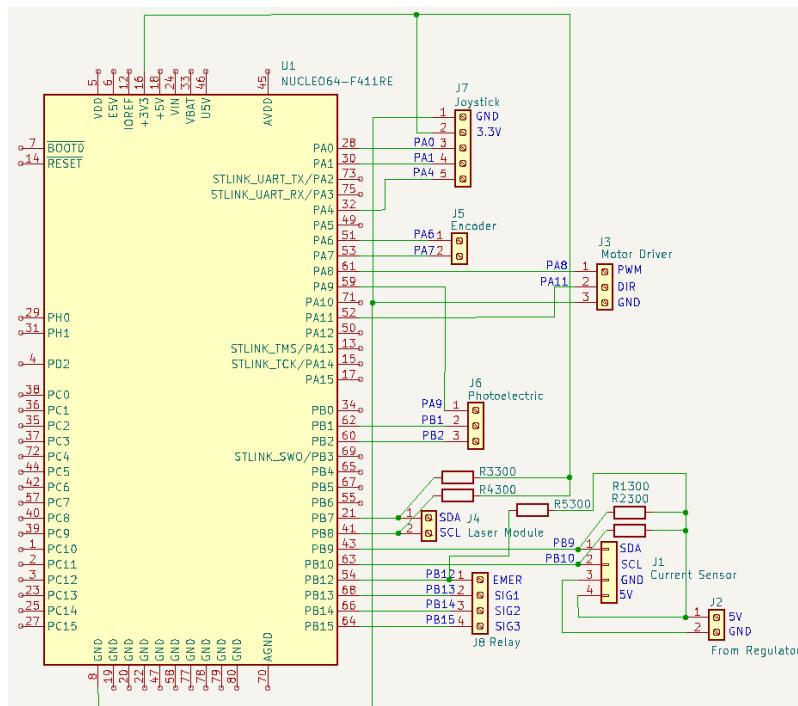
## Control circuit



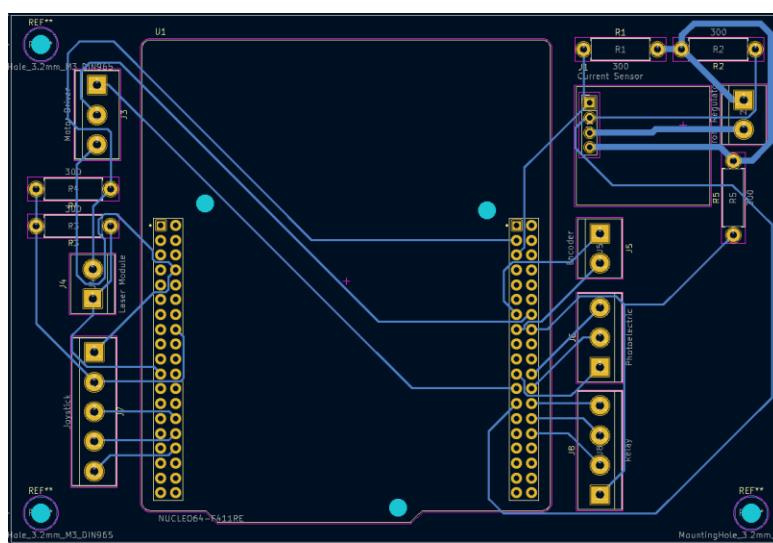
รูปที่ 29 รูปแสดงวงจรควบคุม

## 2. การออกแบบ PCB

ทางกลุ่มผู้จัดทำได้ออกแบบแผ่น PCB โดยใช้โปรแกรม KiCad ดังรูปที่ 30 และ 31 ซึ่งมีการติดตั้งบอร์ด Nucleo F411RE, Terminal Block และเซนเซอร์วัดกระแสไฟฟ้า (Current sensor) เพื่อความสะดวกในการต่อวงจรควบคุมจากพินต่างๆของ Nucleo F411RE ผ่าน Terminal Block ไปยังอุปกรณ์อื่นๆในวงจรควบคุม รวมถึงเซนเซอร์วัดกระแสไฟฟ้า (Current sensor)



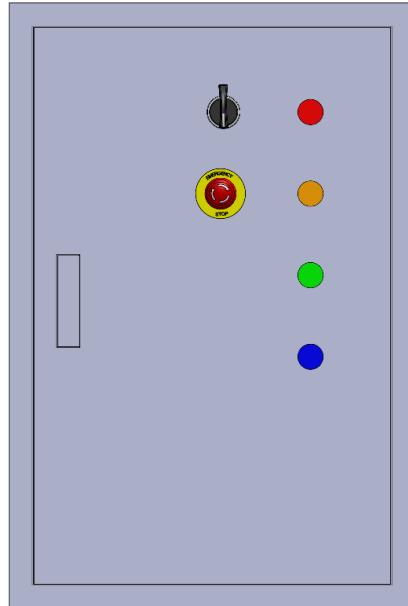
รูปที่ 30 รูปแสดง Schematic ของ PCB



รูปที่ 31 รูปแสดง PCB

### 3. การจัดวางตำแหน่งอุปกรณ์หน้าตู้ไฟและภายในตู้ไฟ

หน้าตู้ไฟจะประกอบไปด้วยปุ่มสวิตซ์เปิดและสวิตซ์ปิด (ON/OFF) สีดำ บริเวณด้านซ้ายบน, ปุ่มควบคุมฉุกเฉิน (Emergency switch) แบบมี lamp แสดงสถานะในตัว บริเวณซ้ายล่าง หลอดไฟแสดงสถานะ Pilot lamp (ON/OFF) สีแดง, Pilot lamp (receiving position) สีส้ม, Pilot lamp (homing x) สีเขียว และ Pilot lamp (homing y) สีฟ้า ดังรูปที่ 32



รูปที่ 32 ตำแหน่งอุปกรณ์หน้าตู้ไฟ



รูปที่ 33 รูปหน้าตู้ไฟจริง

ภายในตู้ไฟจะประกอบไปด้วย 3 Section ในส่วนบนสุดประกอบด้วย Switching Power Supply, fuse 1A, fuse 7A และโมดูลลดระดับแรงดัน ในส่วนตรงกลางประกอบด้วย PCB ที่มีบอร์ด Nucleo F411RE และ INA219 Current Sensor ติดอยู่ และโมดูลขั้บมอเตอร์ และในส่วนล่างสุดจะมี Terminal Bar และโมดูลวีเลอร์



รูปที่ 34 ตำแหน่งอุปกรณ์ภายในตู้ไฟ

#### 4. รายการอุปกรณ์ในตู้ไฟ

##### โมดูลขั้บมอเตอร์ (Motor Driver)

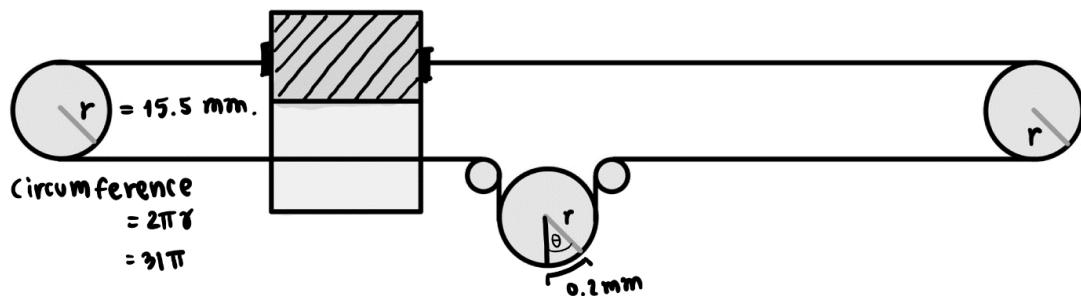
ทางกลุ่มผู้จัดทำได้เลือกใช้โมดูลขั้บมอเตอร์ รุ่น MDD10A ดังรูปที่ 42 มาเป็นอุปกรณ์ในการขับเคลื่อน มอเตอร์ ซึ่งสามารถรองรับการจ่ายกระแสไฟฟ้าอย่างต่อเนื่องให้กับมอเตอร์ได้ 10A และจ่ายกระแสไฟฟ้าสูงสุด 30A เนื่องจากมอเตอร์ที่เลือกใช้คือรุ่น IG42 ที่ใช้ไฟกระแสเดียวที่มีแรงดัน 12V และใช้กระแสไฟฟ้าสูงสุด 10A ทำให้โมดูลขั้บมอเตอร์ รุ่น MDD10A มีความเหมาะสมในการใช้งาน



รูปที่ 35 รูปโมดูลขั้บมอเตอร์ รุ่น MDD10A

#### 4.1 เอ็นโค้ดเดอร์ (Encoder)

จาก Performance Requirement ที่กำหนดกล่าวว่า ค่าความผิดพลาดเมื่อเคลื่อนที่ไปยังตำแหน่งที่กำหนดต้องน้อยกว่า 0.2 mm. หรือ +- 20% จะสามารถตรวจสอบภาพแสดงลักษณะตำแหน่งของเอ็นโค้ดเดอร์และจุดหมายได้ดังนี้



รูปที่ 36 แสดงตำแหน่งของ เอ็นโค้ดเดอร์ และจุดหมาย  
จากรูปที่ 36 สามารถคำนวณองศาคลาดเคลื่อนสูงสุดที่ยอมรับได้เป็น

$$0.2 \cdot 360 / 31\pi = 0.739^\circ$$

แสดงว่า Incremental Encoder ที่ใช้จำเป็นจะต้องอ่านค่าได้อย่างน้อย 0.739 องศา ต่อ 1 pulse โดยหากต้องการให้ Encoder สามารถอ่านค่าได้ 360 องศา หรือ 1 Revolution ดังนั้น เอ็นโค้ดเดอร์ ที่เลือกใช้จะต้องสามารถอ่านค่าได้อย่างน้อย

$$360 / 0.739 \approx 487.145 \text{ Pulse / revolution}$$

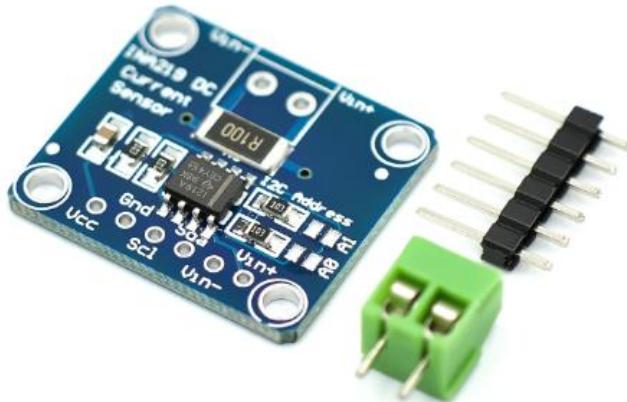
เมื่อได้พิจารณาเอ็นโค้ดเดอร์ จากการสมการข้างต้นที่หากความคลาดเคลื่อนเมื่อเคลื่อนที่ไปยังตำแหน่งที่กำหนดน้อยกว่า 0.2 mm. หรือ +- 20% จึงได้เลือกใช้ Incremental Encoder รุ่น AMT103-V จากบริษัท CUI Devices ซึ่งมีความละเอียดในการอ่านค่าที่ 2048 P/R รองรับความเร็ว 7500 RPM ดังรูปที่ 37



รูปที่ 37 Incremental Encoder AMT103-V

#### 4.2 เซ็นเซอร์วัดกระแสไฟฟ้า (Current sensor)

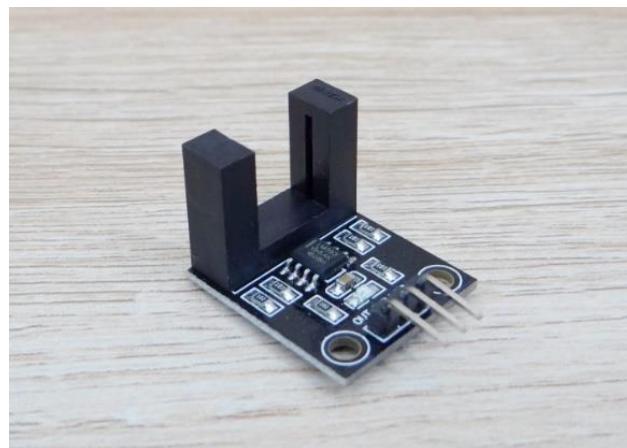
ทางกลุ่มผู้จัดทำได้เลือกใช้เซ็นเซอร์วัดกระแสไฟฟ้า รุ่น INA219 ดังรูปที่ 38 มาใช้ในการวัดค่ากระแสไฟฟ้าที่จ่ายให้กับมอเตอร์ ซึ่งใช้กระแสไฟเพียงแค่ 1 mA สามารถรองรับแรงดันไฟฟ้ากระแสตรงได้ 0V–26V และมีความแม่นยำสูงมากโดยคาดเคลื่อนจากความจริงไม่เกิน 0.5%



รูปที่ 38 รูปเซ็นเซอร์วัดกระแสไฟฟ้า รุ่น INA219

#### 4.3 Photoelectric sensor

ทางกลุ่มผู้จัดทำได้เลือกใช้ LM393 Beam Photoelectric Sensors ดังรูปที่ 39 เนื่องจากมีขนาดที่เล็ก ไม่ถูก grub กวนด้วยโลหะภายนอก รองรับการตรวจจับวัตถุที่เคลื่อนที่ผ่านด้วยความเร็วสูง เพราะมีอัตราการอ่านที่สูง และใช้งานง่ายมีเพียง 3 Pin คือ OUT, VCC และ GND



รูปที่ 39 LM393 Beam Photoelectric sensor

#### 4.4 พิวส์ (Fuse)

สำหรับอุปกรณ์ป้องกันกระแสไฟเกินและกระแสไฟลัดวงจรในวงจรควบคุม ทางกลุ่มผู้จัดทำได้เลือกใช้พิวส์หลอดแก้วที่รับกระแสไฟฟ้าสูงสุดได้ 7A และแรงดันไฟฟ้า 250V ดังรูปที่ 40 โดยพิวส์จะทำหน้าที่ป้องกันกระแสไฟเกินเมื่อหุ่นยนต์แขนกลชนกับวัตถุใดๆ จนมอเตอร์หยุดไม่ได้แต่มีการจ่ายไฟอยู่ และเกิดไฟฟ้าลัดวงจรในวงจรควบคุม เนื่องจากมอเตอร์สามารถใช้กระแสไฟฟ้าได้มากที่สุดประมาณ 10A และอุปกรณ์อื่นๆ ใช้กระแสไฟสูงสุดประมาณ 0.6A ภายในวงจรควบคุมจึงใช้กระแสรวมสูงสุดประมาณ 10.6A



รูปที่ 40 รูปพิวส์ 7A (fuse 7A)

สำหรับอุปกรณ์ป้องกันไฟลัดวงจรสำหรับวงจรกำลัง ทางกลุ่มผู้จัดทำได้เลือกพิวส์หลอดแก้วที่รับกระแสไฟฟ้าสูงสุดได้ 1A AC ดังรูปที่ 41 เนื่องจากแหล่งจ่ายไฟฟ้ากรุงเทพ (Switching Power Supply) 12V 15A มีกำลังไฟฟ้า (P) เท่ากับ 180W โดยคำนวณกระแสไฟฟ้าจากสูตร

$$\begin{aligned} I &= \frac{P}{V} \\ I &= \frac{180}{220} \\ I &= 0.82A \end{aligned}$$

ทางกลุ่มผู้จัดทำจึงได้เลือกพิวส์หลอดแก้วรับกระแสไฟฟ้าได้ที่ค่าใกล้เคียงที่สุดที่ 1A AC



รูปที่ 41 รูปพิวส์ 1A (fuse 1A)

#### 4.5 แหล่งจ่ายไฟฟ้ากระแสสลับ (Switching Power Supply)

เนื่องจาก Physical Requirement ของโครงการได้กำหนดไว้ว่าต้องใช้แหล่งจ่ายไฟกระแสสลับที่มีแรงดัน 220V จึงจำเป็นต้องเปลี่ยนไฟฟ้ากระแสสลับที่มีแรงดัน 220V เป็นไฟฟ้ากระแสตรงที่มีแรงดัน 12V เพื่อนำไปจ่ายให้กับมอเตอร์รวมถึงโหลดอื่นๆ โดยมอเตอร์เป็นโหลดที่ใช้กระแสไฟฟ้ามากที่สุด จึงคำนวณกำลังไฟฟ้า ( $P$ ) ที่มอเตอร์ใช้ ดังนี้

จากสูตร

$$P_{\text{motor}} = I_{\text{motor}} \cdot V_{\text{motor}}$$

$$P_{\text{motor}} = 10 \cdot 12$$

$$P_{\text{motor}} = 120 \text{ W}$$

$$P_{\text{other}} = 6.04 \text{ W}$$

$$P_{\text{Total}} = 126.04 \text{ W}$$

$$P_{\text{Total}_{\text{Eff}}} = 126.04(1.25)$$

$$P_{\text{Total}_{\text{Eff}}} = 157.55 \text{ W}$$

จะได้ว่ามอเตอร์ใช้กำลังไฟฟ้า 120W และโหลดอื่นๆ ใช้กำลังไฟฟ้ารวมกันประมาณ 6.04W ทั้งระบบ จึงใช้กำลังไฟฟ้าทั้งหมดประมาณ 126.04W เมื่อเทียบกับ Switching Power Supply มีประสิทธิภาพ 80% จึงได้ว่าต้องใช้ Switching Power Supply ที่มีกำลังไฟฟ้าอย่างน้อย 157.55W ดังนั้นจึงได้เลือกใช้ Switching power supply 12V 15A ที่มีกำลังไฟฟ้า ( $P$ ) เท่ากับ 180W ซึ่งเพียงพอต่อการจ่ายไฟให้กับอุปกรณ์ในระบบทั้งหมด



รูปที่ 42 แหล่งจ่ายไฟฟ้ากระแสสลับ (Switching Power Supply)

#### 4.6 โมดูลลดระดับแรงดัน (Step-down regulator module)

เนื่องจากภายในระบบมีอุปกรณ์อิเล็กทรอนิกส์หลายอุปกรณ์ที่ใช้แรงดันต่ำ (5V) จึงมีความจำเป็นที่จะต้องใช้ไฟฟ้ากระแสสลับที่มีขนาดแรงดันขนาด 5V ด้วย ดังนั้นจึงได้เลือกใช้โมดูล XL4015 ตั้งรูปที่ 43 เพื่อแปลงไฟกระแสสลับแรงดันไฟฟ้า 12V ลงเป็นไฟฟ้ากระแสสลับแรงดันไฟฟ้า 5V โดยโมดูลนี้สามารถรับแรงดันไฟฟ้ากระแสสลับที่มีขนาดตั้งแต่ 5V-36V และเป็นแรงดันไฟฟ้ากระแสสลับที่มีขนาดตั้งแต่ 1V-32V โดยรับกระแสไฟฟ้าสูงสุดได้ 5A



รูปที่ 43 Step – Down Regulator Module XL4015

#### 4.7 ปุ่มควบคุมฉุกเฉิน (Emergency switch)

เนื่องจาก Functional Requirement ของโครงการกำหนดไว้ว่าต้องมีปุ่มควบคุมฉุกเฉิน ทางกลุ่มผู้จัดทำจึงเลือก ปุ่มควบคุมฉุกเฉินที่สามารถทนแรงดันไฟฟ้ากระแสสลับและกระแสแรงขนาด 600V 10A ตั้งรูปที่ 44 เมื่อนำมาใช้กับแรงดันไฟฟ้ากระแสสลับ 12V จะสามารถทนกระแสไฟฟ้าได้ประมาณ 10A จากโหลด ที่สามารถรองรับแรงดันไฟฟ้ากระแสสลับ 12V ได้ ซึ่งเหมาะสมในการใช้งาน



รูปที่ 44 รูป Emergency switch

#### 4.8 ปุ่มสวิตช์เปิดและสวิตช์ปิด (ON/OFF switch)

เนื่องจากทางกลุ่มผู้จัดทำต้องการให้สามารถเปิดและปิดตู้ไฟ โดยการตัดต่อวงจรในส่วนของไฟฟ้ากระแสสลับ จึงได้เลือกใช้ สวิตช์เปิดและสวิตช์ปิดที่สามารถงานแรงดันไฟฟ้ากระแสสลับและกระแสตรงที่มีขีนหาด 600V 10A ดังรูปที่ 45 เมื่อนำมาใช้กับแรงดันไฟฟ้ากระแสตรง 12V จะสามารถงานกระแสไฟฟ้าได้ประมาณ 10A จากโหลดซึ่งเหมาะสมในการใช้งาน



รูปที่ 45 รูปปุ่มสวิตช์เปิดและสวิตช์ปิด (ON/OFF switch)

#### 4.9 โมดูลรีเลย์ (Relay module)

ทางกลุ่มผู้จัดทำได้เลือกใช้โมดูลรีเลย์ที่ใช้แรงดันไฟฟ้ากระแสตรงที่มีพิกัด 3.3V 4 ช่อง ดังรูปที่ 46 มาใช้ในการเปิดและปิดไฟแสดงสถานะของทุ่นยนต์และไฟสถานะของปุ่มควบคุมฉุกเฉิน (Emergency switch) หน้าตู้ไฟ เนื่องจากแรงดันไฟฟ้าที่ออกจากการ GPIO ของ NUCLEO-F411RE มีขีนหาด 3.3V ทำให้ง่ายต่อการใช้งาน



รูปที่ 46 รูปปุ่มสวิตช์เปิดและสวิตช์ปิด (ON/OFF switch)

#### 4.10 หลอดไฟแสดงสถานะ (Pilot lamp)

ทางกลุ่มผู้จัดทำต้องการเห็นสถานะต่างๆ ของหุ่นยนต์และตู้ไฟ ได้แก่ สถานะเมื่อเปิดและปิดการทำงานของตู้ไฟ(ON/OFF), สถานะขณะหุ่นยนต์กำลังรับค่าตำแหน่ง(receiving position), สถานะขณะหุ่นยนต์กำลังเคลื่อนที่เข้าสู่ตำแหน่งโฮมในแกน x, สถานะขณะหุ่นยนต์กำลังเคลื่อนที่เข้าสู่ตำแหน่งโฮมในแกน y จึงได้เลือก หลอดไฟแสดงสถานะ (Pilot lamp) 220V ซึ่งสามารถใช้ได้กับกระแสตรงที่มีแรงดัน 12V โดยมีเส้นผ่านศูนย์กลาง 22 มม. ดังรูปที่ 47



รูปที่ 47 รูป Pilot lamp

#### 4.11 โมดูลจอยสติ๊ก (Joystick module)

ทางกลุ่มผู้จัดทำได้เลือกใช้โมดูลจอยสติ๊ก PS2 ตั้งรูปที่ 62 มี 2 แกน แบบอนาล็อกที่มีปุ่มกดที่รวมมาด้วย เนื่องด้วยความง่ายในการใช้งาน ความกะทัดรัดของโมดูล และความจำเป็นที่จะต้องใช้ปุ่มเพื่อบันทึกตำแหน่ง เทrey์ โดยโมดูลนี้ใช้ไฟ 3.3VDC



รูปที่ 48 รูปโมดูลจอยสติ๊ก (Joystick module)

## 5. การเลือกสายไฟ

เนื่องจากขนาดของสายไฟแต่ละประเภทสามารถรองรับกระแสได้แตกต่างกัน ทางกลุ่มผู้จัดทำจึงจำเป็นต้องคำนวณขนาดสายไฟ ดังนี้

หากกำลังไฟฟ้าสูงสุด ( $P_{max}$ ) ที่มอเตอร์ใช้ จากสูตร

$$\begin{aligned} \text{Torque} &= \frac{9.548 \cdot P_{max}}{\omega} \\ 0.196 &= \frac{9.548 \cdot P_{max}}{400} \\ P_{max} &= 8.21 \text{ W} \end{aligned}$$

ดังนั้น กำลังไฟฟ้า ( $P$ ) ที่มอเตอร์ใช้ เท่ากับ 8.21W

หากกระแสสูงสุด ( $I_{max}$ ) ที่โหลดใช้จากมอเตอร์ จากสูตร

$$\begin{aligned} P &= IV \\ I &= 8.21/12 \\ I &= 0.684 \text{ A} \end{aligned}$$

ดังนั้นมอเตอร์จะใช้กระแสสูงสุดขณะมีโหลด เท่ากับ 0.684A แต่เนื่องจากขณะที่มอเตอร์เริ่มทำงาน จะเกิดการกระแสโซกได้ประมาณ 6-8 เท่า จึงนำ 8 เท่าของกระแสโซกมาคำนวณเพื่อหาความปลอดภัยของสายไฟที่ต้องการจะใช้ หากกระแสกระแสโซกสูงสุดของมอเตอร์ จากค่ากระแสสูงสุด ( $I_{max}$ ) ที่โหลดใช้จากมอเตอร์

$$\begin{aligned} I &= 0.684 \cdot 8 \\ I &= 5.472 \text{ A} \end{aligned}$$

ดังนั้นกระแสกระแสโซกสูงสุดของมอเตอร์คือ 5.472A

หากขนาดของสายไฟที่เหมาะสมกับการทนกระแส จากสูตร

$$\begin{aligned} I &= 1.25 \cdot 5.472 \\ I &= 6.84 \text{ A} \end{aligned}$$

ดังนั้นจึงต้องเลือกสายไฟที่สามารถทนกระแสได้มากกว่า 6.84A ดังนั้นทางกลุ่มผู้จัดทำจึงเลือกใช้สายไฟสำหรับระบบไฟฟ้ากระแสสลับ ประเภท IEC02 THWF ดังรูปที่ 49 ที่มีขนาดพื้นที่หน้าตัดเท่ากับ  $2.5 \text{ mm}^2$  รองรับกระแสได้มากที่สุด 16A ซึ่งเพียงพอ กับกระแสที่โหลดในระบบใช้ทั้งหมด เลือกใช้สายไฟประเภท IEC06 VSF ดังรูปที่ 50 ที่มีขนาดพื้นที่หน้าตัดเท่ากับ  $1 \text{ mm}^2$  รองรับกระแสได้มากที่สุด 10A ซึ่งเพียงพอ กับกระแสที่มอเตอร์ใช้ และเลือกใช้สายไฟประเภท IEC06 VSF ดังรูปที่ 51 ที่มีขนาดพื้นที่หน้าตัดเท่ากับ  $0.5 \text{ mm}^2$  รองรับกระแสได้มากที่สุด 3A ซึ่งเพียงพอ กับกระแสที่อุปกรณ์อื่นๆ ในระบบใช้



## 6. ขั้นตอนและเงื่อนไขการทำงานของวงจร

### 6.1 เงื่อนไขการทำงานของวงจร

1. เมื่อผู้ใช้งานกดปุ่มสวิตช์เปิดและสวิตช์ปิด (ON/OFF switch) จะเป็นการเปิดหรือปิดการทำงานของหุ่นยนต์ หากกดปุ่มเปิด วงจรควบคุมจะได้รับไฟฟ้ากระแสสลับขนาด 12V ที่แหล่งจ่ายไฟฟ้ากระแสสลับ (Switching Power Supply) และมาจากการแรงดันไฟฟ้ากระแสสลับขนาด 220V ทำให้ Pilot lamp (ON/OFF) ติดสว่าง และเมื่อกดปิด จะไม่มีไฟเลี้ยงวงจรควบคุม และ Pilot lamp (ON/OFF) จะดับ
2. การกดปุ่มควบคุมฉุกเฉิน (Emergency switch) จะเป็นเพียงการหยุดการทำงานของเตอร์กษทันทันทีทันใด รวมถึงมีสัญญาณส่งไปยัง NUCLEO-F411RE เพื่อหยุดการทำงานของอุปกรณ์อื่นๆด้วยโปรแกรม Interrupt ที่ถูกเขียนไว้ก่อนหน้า
3. ในแต่ละสถานะการทำงานของหุ่นยนต์ หลอดไฟแสดงสถานะ (Pilot lamp) แต่ละดวงจะติดสว่างตามที่ได้ออกแบบไว้ ได้แก่ Pilot lamp (receiving position), Pilot lamp (homing x) และ Pilot lamp (homing y)
4. หากเกิดเหตุการณ์กระแสไฟเกิน เนื่องจากมอเตอร์ดึงกระแสมากเกินไป ซึ่งมากกว่ากระแสที่พิวร์สอร์รองรับได้พิวร์ส์จะขาด
5. หากเกิดเหตุการณ์ลัดวงจร พิวร์ส์จะขาด เพื่อป้องกันความเสียหายของวงจร

### 6.2 ขั้นตอนการทำงานของวงจร

1. ผู้ใช้งานเสียบสายไฟที่มีแรงดันไฟฟ้ากระแสสลับขนาด 220V เข้าตู้ไฟ แหล่งจ่ายไฟฟ้ากระแสสลับ (Switching Power Supply) จะแปลงแรงดันไฟฟ้ากระแสสลับขนาด 220V ให้เป็นไฟฟ้ากระแสสลับขนาด 12V
2. ผู้ใช้งานกดปุ่มสวิตช์เปิดและสวิตช์ปิด (ON/OFF switch) เพื่อทำการเปิดเครื่อง ไฟฟ้ากระแสสลับขนาด 12V จะถูกจ่ายให้กับอุปกรณ์ในวงจรควบคุม ได้แก่ มอเตอร์ (DC motor) และโมดูลขับมอเตอร์ (Motor Driver) และ Pilot lamp (ON/OFF) จะติดสว่าง หุ่นยนต์จึงพร้อมใช้งาน
3. โมดูลลดระดับแรงดัน (Step-down regulator) จะลดระดับแรงดันไฟฟ้ากระแสสลับขนาด 12V ให้เหลือแรงดันไฟฟ้ากระแสสลับขนาด 5V เพื่อจ่ายให้กับอุปกรณ์ในวงจรควบคุม ได้แก่ เซนเซอร์วัดกระแสไฟฟ้า (Current sensor), Photoelectric sensor และเอ็นโคเดอร์ (Encoder)
4. เสียบสาย USB เพื่อจ่ายไฟให้กับ NUCLEO-F411RE และเชื่อมต่อการทำงานระหว่าง NUCLEO-F411RE กับ Base system

5. เมื่อ NUCLEO-F411RE ได้รับไฟเลี้ยงแล้ว จะจ่ายแรงดันไฟฟ้ากระแสตรงขนาด 3.3V ให้กับโมดูลจอยสติ๊ก (Joystick module) และทำหน้าที่ตามที่ได้เขียนโปรแกรมไว้ ทั้งการสั่งงานและรับค่าจากอุปกรณ์อื่น ดังนี้

5.1 ทำหน้าที่สั่งงานอุปกรณ์อื่น ได้แก่ ส่งค่าไปยัง Base system, ควบคุมมอเตอร์ โดยใช้โมดูลข้อมูลมอเตอร์ในการขับ, ควบคุมโมดูลเลเซอร์ของทาง FIBO โดยการสื่อสารกับ Arduino Nano และส่งออกสัญญาณไปยังคอล์ยของโมดูลรีเลย์ (Relay module) เพื่อควบคุมการติดสว่างของหลอดไฟแสดงสถานะ ตามสถานะการทำงานของหุ่นยนต์

5.2 ทำหน้าที่รับค่าจากอุปกรณ์อื่น ได้แก่ รับค่าที่ถูกส่งมาจากรหัสบาร์โค้ด (Barcode), รับค่ากระแสจาก Current sensor, รับค่าองศาที่มอเตอร์หมุนจาก Encoder, รับสัญญาณ 0 และ 1 จาก Inductive proximity sensor และรับค่าตำแหน่งจาก Joystick

6. เมื่อผู้ใช้งานเห็นว่าหุ่นยนต์เกิดการขัดข้อง แล้วกดปุ่มควบคุมฉุกเฉิน (Emergency switch) จะทำให้ Motor หยุดหมุนกะทันหัน แต่อุปกรณ์อื่นๆในระบบยังมีไฟเลี้ยงอยู่

7. เมื่อเกิดกระแสเกินหรือลัดวงจร Fuse จะขาดทำให้วงจรถูกตัดไฟ เพื่อป้องกันวงจรเสียหาย

8. ผู้ใช้งานกดปุ่มสวิตซ์เปิดและสวิตซ์ปิด (ON/OFF switch) เพื่อทำการปิดเครื่อง เมื่อกดปิด จะเป็นการหยุดจ่ายไฟให้กับระบบ และ Pilot lamp (ON/OFF) จะดับ หุ่นยนต์จึงหยุดทำงาน

## 7. การเขียนโปรแกรม

ในการทดสอบตู้ไฟ ทางผู้จัดทำได้เขียนโปรแกรมเพื่อทดสอบระบบต่างๆ โดยจะมีการทดสอบระบบทั้งหมด 5 ระบบ ดังนี้

- ระบบการเปิดปิดไฟแสดงสถานะของ Pilot lamp หน้าตู้ด้วย Relay Module โดยทดสอบด้วยโปรแกรมดังนี้

```
void led_fnc(){
    if(emer_pushed == 1){
        if(rx[0] == '1' && action == 0){
            action = 1;
            HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_13);
        }
        if(rx[0] == '2' && action == 0){
            action = 1;
            HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_14);
        }
        if(rx[0] == '3' && action == 0){
            action = 1;
            HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_15);
        }
    }
}
```

รูปที่ 52 โปรแกรมทดสอบ Pilot lamp

โดยการทำงานของโปรแกรมคือเมื่อได้รับคำสั่งจากคอมพิวเตอร์ผ่าน UART เป็นเลข 1 ให้เปิด Pilot lamp (receiving tray) สีส้ม เมื่อกดเลข 2 เปิด Pilot lamp (ready to run) สีเขียว และเมื่อกดเลข 3 ให้เปิด Pilot lamp (running) สีฟ้า

- ระบบการสั่งการเปิด/ปิด เพิ่ม/ลดความเร็ว และการเปลี่ยนทิศทางของมอเตอร์ โดยจะทดสอบด้วยโปรแกรมดังนี้

```
void motor_fnc(){
    if(emer_pushed == 1){
        if(rx[0] == '5' && action == 0){
            action = 1;
            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_11);
        }
        if(rx[0] == '7' && action == 0){
            action = 1;
            duty += 50;
        }
        if(rx[0] == '4' && action == 0){
            action = 1;
            duty -= 50;
        }
        if(rx[0] == '9' && action == 0){
            action = 1;
            duty = 0;
        }
    }
    HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,duty);
}
```

รูปที่ 53 โปรแกรมทดสอบ Motor

โดยการทำงานของโปรแกรมคือเมื่อได้รับคำสั่งจากคอมพิวเตอร์ผ่าน UART เป็นเลข 5 จะทำการเปลี่ยนทิศทางของมอเตอร์ด้วยการสั่งงาน Pin A11 หรือ Pin direction ของ Motor driver เมื่อกด

คำสั่ง 7 หรือ 4 จะเป็นการเพิ่มลดความเร็วของเตอร์ด้วยการปรับสัญญาณ PWM และการกดคำสั่ง 9 จะเป็นการหยุดมอเตอร์

### 3. ระบบการตรวจจับวัตถุของ Photoelectric Sensor โดยจะทดสอบด้วยโปรแกรมดังนี้

```
void check_pe(){
    if(emer_pushed == 1){
        pe1_st = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_1);
        pe2_st = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2);
        pe3_st = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);
    }
}
```

รูปที่ 54 โปรแกรมทดสอบ Photoelectric sensor

โดยการทำงานของโปรแกรมคือเมื่อมีวัตถุมาบังที่ Photoelectric Sensor ตัวใดตัวหนึ่ง ตัวแปร photoelectric state (peX\_st) ที่สอดคล้องกับตัวนั้นจะอ่านสัญญาณได้เป็น LOW

### 4. ระบบ Joystick โดยจะทดสอบด้วยโปรแกรมดังนี้

```
void check_js(){
    if(emer_pushed == 1){
        if(joystick_reading == 0){
            HAL_ADC_Start_DMA(&hadc1, (uint32_t*)joy_xy, 2);
            joystick_reading = 1;
        }
        sw = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4);
    }
}
```

รูปที่ 55 โปรแกรมทดสอบ Joystick

โดยการทำงานของโปรแกรมคือในครั้งแรกตั้งแต่เปิดเครื่อง จะสั่งให้ Nucleo F411RE รับค่า Joystick แบบ DMA เรื่อยๆ และหลังจากนั้นจะอ่านค่า Joystick Switch เรื่อยๆ ทุก 10 ms

### 5. ระบบ Emergency Switch Interrupt

ระบบ Emergency Switch Interrupt จะทำงานด้วยการใช้เป็น External Interrupt เมื่อปุ่ม Emergency ถูกกด Nucleo F411RE จะหยุดการทำงานตามปกติและมาทำงานในฟังก์ชัน Callback ของ Interrupt ก่อน โดยฟังก์ชัน Callback เป็นไปดังนี้

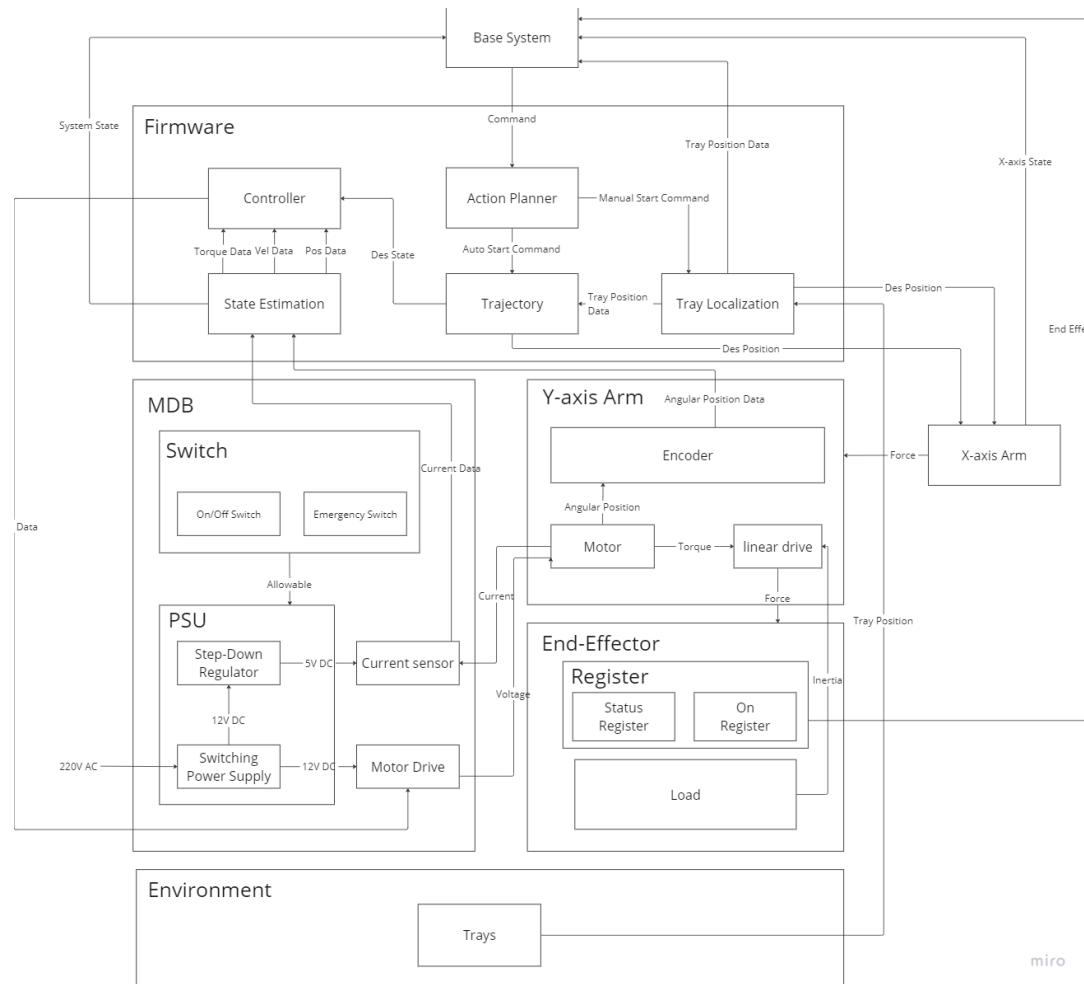
```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
    if(GPIO_Pin == GPIO_PIN_12 && HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12) == 0){
        emer_pushed = 0;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0);
        __HAL_TIM_SET_COMPARE(&htim1, TIM_CHANNEL_1, 0);
    }
    if(GPIO_Pin == GPIO_PIN_12 && HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12) == 1){
        emer_pushed = 1;
    }
}
```

รูปที่ 56 โปรแกรม Emergency Switch Interrupt

โดยการทำงานของโปรแกรมก็คือเมื่อเกิดการ Interrupt ขึ้นและเป็น EXTI ของ Emergency Switch จะเข้ามาในฟังกชั่น Callback นี้ เมื่อรับรู้ว่าเป็นการกดปุ่ม (Falling Edge) จะทำการสั่ง Pilot lamp ทั้งหมดให้ปิด และสั่งให้มอเตอร์หยุดการทำงาน หลังจากนั้นก็จะเปลี่ยนตัวแปร emer\_pushed ให้เป็น 0 เพื่อไม่ให้เกิดการทำงานต่อไปของระบบทั้งหมด เนื่องจากระบบทุกรอบที่ได้กล่าวมาของโปรแกรมนี้จะมีการเช็คเงื่อนไข emer\_pushed เท่ากับ 1 หรือไม่ ก่อนทำงานทุกรอบ และเมื่อปุ่มปุ่ม Emergency ออก (Rising Edge) ก็จะคืนค่า emer\_pushed ให้เป็น 1 อีกครั้ง ก็จะทำงานต่อไปได้ตามปกติ

# ความคืบหน้าของการออกแบบระบบควบคุม (Control Engineering)

## 1. System Architecture Diagram



รูปที่ 57 System Architecture Diagram ของระบบแขนกล

ระบบโดยภาพรวมจะรับคำสั่งจาก Base System ในการเลือกว่าช่วงไหนของระบบควรเป็นแบบ Manual ช่วงไหนของระบบควรเป็นแบบ Auto โดยในช่วงที่เป็น Manual ระบบจะ Localization ตำแหน่งต่าง ๆ ได้แก่ ชุดถ้วยหรือตำแหน่งที่ต้องการให้ถ้วยไปวางผ่านการบังคับ Joy Stick และระบบจะเข้าสู่โหมด Auto เมื่อผู้ใช้กดปุ่ม Play ที่ Base System หลังจากกำหนดตำแหน่งของถ้วยและเป้าหมายเรียบร้อยแล้ว โดยในโหมด Auto ส่วน Action Plan ก็จะทำการป้อนค่าให้แขนกลแกน X และแขนกลแกน Y ซึ่งในส่วนของแขนกลแกน Y ก็จะมีการคำนวณวิถีโคจร และระบบควบคุมเพื่อให้ End Effector เคลื่อนย้ายของระหว่างถ้วยได้ตามที่ต้องการ โดยที่แขนกลแกน X แขนกลแกน Y และ End Effector จะมีการติดต่อกับ Base System ตลอด

## 2. System Identification

### 2.1 พลวัตของมอเตอร์

พิจารณาพลวัตของมอเตอร์เพื่อนำไปสร้างแบบจำลองความสัมพันธ์ระหว่างแรงดันที่จ่ายให้มอเตอร์ และความเร็วในการหมุนของมอเตอร์สำหรับนำไปใช้ในการคำนวณเพื่ออ้างอิงค่า Gain ในพจน์ต่าง ๆ ของระบบควบคุม PID ต่อไป

โดยกำหนดให้

$$v_{in} = \text{แรงดันที่จ่ายให้มอเตอร์}$$

$$R = \text{ความต้านทานภายในมอเตอร์}$$

$$L = \text{ค่าความเหนี่ยวนำภายในมอเตอร์}$$

$$k_b = \text{สัมประสิทธิ์ Back EMF ของมอเตอร์}$$

$$k_m = \text{สัมประสิทธิ์トルกของมอเตอร์}$$

$$\tau_m = \text{トルกที่เกิดขึ้นจากมอเตอร์}$$

$$B_m = \text{แเดมป์รูมของมอเตอร์}$$

$$J_m = \text{โมเมนต์ความเฉื่อยของมอเตอร์}$$

$$i = \text{กระแสที่เหลากายในวงจร}$$

$$\omega = \text{ความเร็วเชิงมุมของมอเตอร์}$$

พิจารณาพลวัตของมอเตอร์จะสามารถตั้งสมการได้ดังนี้

$$\text{KVL ; } v_{in} - Ri - L \frac{di}{dt} - k_b = 0$$

$$\text{Motor Torque ; } \tau_m = k_m i$$

$$\Sigma \tau = J\alpha ; \quad \tau_m - B_m = J_m \frac{d\omega}{dt}$$

ดังนั้นจึงสามารถลดรูปสมการได้เป็น

$$v_{in} - Ri - k_b = L \frac{di}{dt}$$

$$k_m i - B_m = J_m \frac{d\omega}{dt}$$

เมื่อ Laplace Transform จึงสามารถจัดรูปสมการใน Frequency Domain

$$\Theta(s) = \frac{k_m}{J_m s^2 + B_m s} I(s)$$

$$V_{in}(s) - k_b s \Theta(s) = (L_s + R) I(s)$$

จากสมการที่ได้ จึงสามารถเขียน Transfer Function ได้ดังนี้

$$\begin{aligned} G(s) &= \frac{K_m}{1 + \frac{K_m}{(Ls + R)(J_m s + B_m)}} \\ G(s) &= \frac{K_m}{J_m L s^2 + (J_m R + B_m L)s + B_m R + K_m K_b} \end{aligned}$$

ซึ่งเมื่อนำไปคำนวณจริงจึงสามารถ Discretize ด้วยวิธีการประมาณด้วยการหาผลต่างข้างหลังได้ดังนี้

แทน  $s \approx \frac{1-z^{-1}}{\Delta t}$  จะได้ว่า

$$\begin{aligned} G(z) &= \frac{K_m}{J_m L \left(\frac{1-z^{-1}}{\Delta t}\right)^2 + (J_m R + B_m L) \left(\frac{1-z^{-1}}{\Delta t}\right) + B_m R + K_m K_b} \\ &= \frac{\frac{K_m}{\frac{J_m L}{(\Delta t)^2} + \frac{J_m R + B_m L}{\Delta t} + B_m R + K_m K_b}}{1 - \left( \frac{\frac{2J_m L}{(\Delta t)^2} + \frac{J_m R + B_m L}{\Delta t}}{\frac{J_m L}{(\Delta t)^2} + \frac{J_m R + B_m L}{\Delta t} + B_m R + K_m K_b} \right) z^{-1} + \left( \frac{\frac{J_m L}{(\Delta t)^2}}{\frac{J_m L}{(\Delta t)^2} + \frac{J_m R + B_m L}{\Delta t} + B_m R + K_m K_b} \right) z^{-2}} \end{aligned}$$

เนื่องจาก Parameters ของมอเตอร์ทั้งหมดมีค่าเป็นบวก ดังนั้น จึงสามารถสรุป Transfer Function ของ มอเตอร์ออกมาได้ว่า

$$G(z) = \frac{N}{1 - D_1 z^{-1} + D_2 z^{-2}} \quad \text{เมื่อ } N > 0, D_1 > 0 \text{ และ } D_2 > 0$$

สุดท้ายจึงสามารถสรุปได้ว่า จำนวน Poles ของมอเตอร์เท่ากับ 2 และ Zeros ของมอเตอร์เท่ากับ 0

## 2.2 Transfer Function ของมอเตอร์ SPG30HP

เมื่อลองสั่ง SPG30HP ให้หมุนด้วยสัญญาณ Ramp ตั้งแต่ 0V – 10V เป็นระยะเวลา 120 วินาที โดยวัดความเร็วด้วย Encoder รุ่น AMT103-V ความละเอียด 2048 PPR ได้ Transfer Function ที่ได้จากแบบจำลองที่มี 0 Poles และ 2 Zeros ดังรูป

```

tf4 =
From input "u1" to output "y1":
-9.715e-07
-----
1 - 2 z^-1 + z^-2

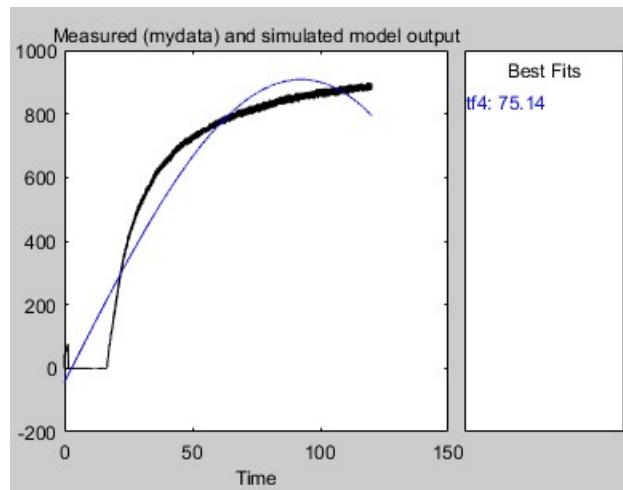
Name: tf4
Sample time: 0.01 seconds
Discrete-time identified transfer function.

Parameterization:
Number of poles: 2 Number of zeros: 0
Number of free coefficients: 3
Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

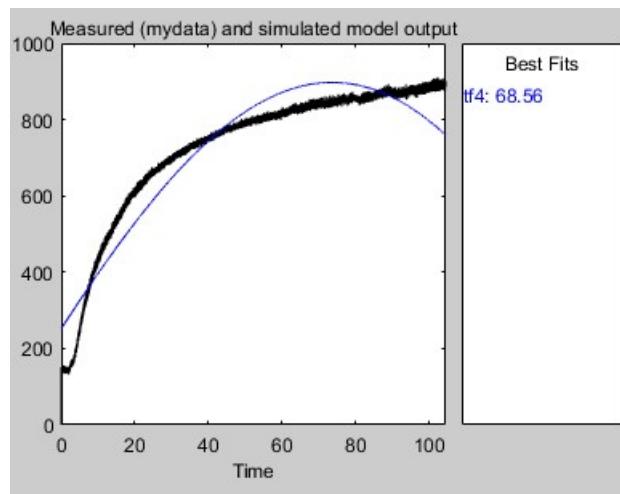
```

รูปที่ 72 Transfer Function ของ SPG30HP จากการทดลอง

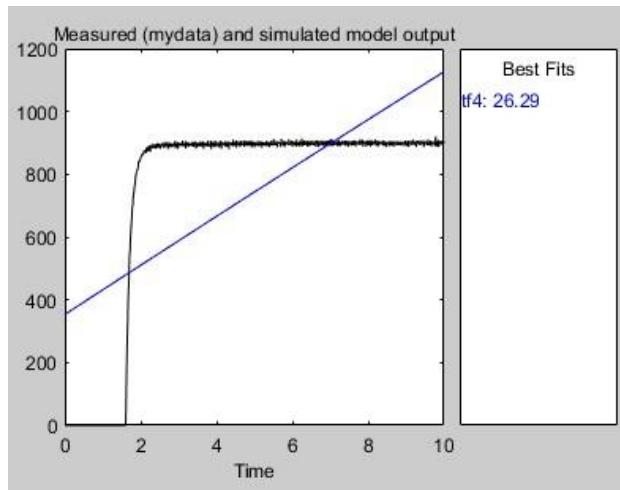
โดย Transfer Function ดังกล่าวมีค่า Best Fit อยู่ที่ 75.14% เมื่อ Input เป็นสัญญาณ Ramp ตั้งแต่ 0V – 10V มีค่า Best Fit ที่ 68.56% เมื่อ Input เป็น สัญญาณ Ramp ตั้งแต่ 1.3V – 10V (มอเตอร์เริ่มหมุนเมื่อจ่ายไฟ 1.3V) และมีค่า Best Fit ที่ 26.29% เมื่อ Input เป็น สัญญาณ Step ที่ 10V



รูปที่ 58 Output ของ Transfer Function ที่ได้จากการทดลองเทียบกับ Output จริง เมื่อจ่ายสัญญาณ Ramp ตั้งแต่ 0V – 10V ให้มอเตอร์



รูปที่ 59 Output ของ Transfer Function ที่ได้จากการทดลองเทียบกับ Output จริง เมื่อจ่ายสัญญาณ Ramp ตั้งแต่ 1.3V – 10V ให้มอเตอร์



รูปที่ 60 Output ของ Transfer Function ที่ได้จากการทดลองเทียบกับ Output จริง เมื่อจ่ายสัญญาณ Step ที่ 10V ให้มอเตอร์

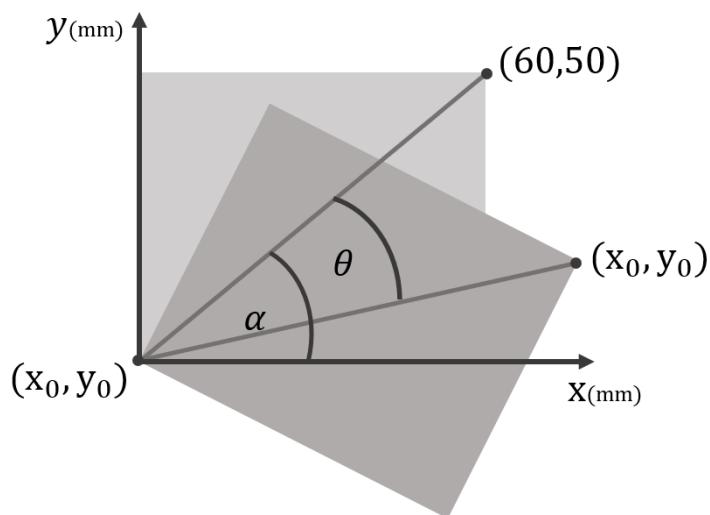
### 3. Action plan

ส่วนของ Action Plan จะเป็นส่วนที่รับค่าตำแหน่งของถาดวางที่ระบุในระบบพิกัดฉากมาแปลงเป็นตำแหน่งจุด 9 จุดที่แสดงถึงรูของถาดวางสำหรับนำไปใช้ในการกำหนดเป็นจุดหมายในการคำนวณวิธีโครงสร้างเพื่อให้แขนกลสามารถเคลื่อนตำแหน่งไปหยิบและวางวัตถุได้ตามที่ต้องการ

เพื่อคำนวณตำแหน่งรูทั้ง 9 รู และส่งตำแหน่งและลักษณะการจัดวางของถาดวางให้กับ Base System ส่วน Action Plan จึงประกอบไปด้วย 2 ส่วน ได้แก่ ส่วน Tray Localization ที่จะรับตำแหน่งของถาดวางในระบบพิกัดฉากมาแปลงเป็นจุดอ้างอิงของถาดวางและมุมที่ถาดวางทำกับแกนอ้างอิง และส่วน Tray Holes Calculation ที่จะนำจุดอ้างอิงของถาดวางและมุมที่ถาดวางทำกับแกนอ้างอิงมาคำนวณเป็นตำแหน่งรูทั้ง 9 รูของถาดวาง

#### 3.1 Tray Localization

ทางทีมวิศวกรได้เลือกทำ Action Plan แบบเลือกจุด 2 จุดที่ท้ายมุมกันเพื่อที่จะระบุตำแหน่งของอีก 9 จุดของถาดวางอีกมา โดยการเลือกจุดจะนำมาเปรียบเทียบกันระหว่างจุดของแกน Y โดยจะเลือกจากจุดที่มีค่าน้อยที่สุด แต่ถ้าจุดของแกน Y ทั้งสองมีค่าเท่ากัน จะเลือกจากจุดของแกน X ที่มีค่าน้อยที่สุดเป็นจุด reference



รูปที่ 61 การวางตัวของถาดวางที่จุดอ้างอิง  $(x_0, y_0)$  โดยทำมุกับแกนอ้างอิง  $\theta$

เมื่อได้จุด Reference และจึงสามารถคำนวณหมุนที่ทำกับแกนอ้างอิงได้จาก

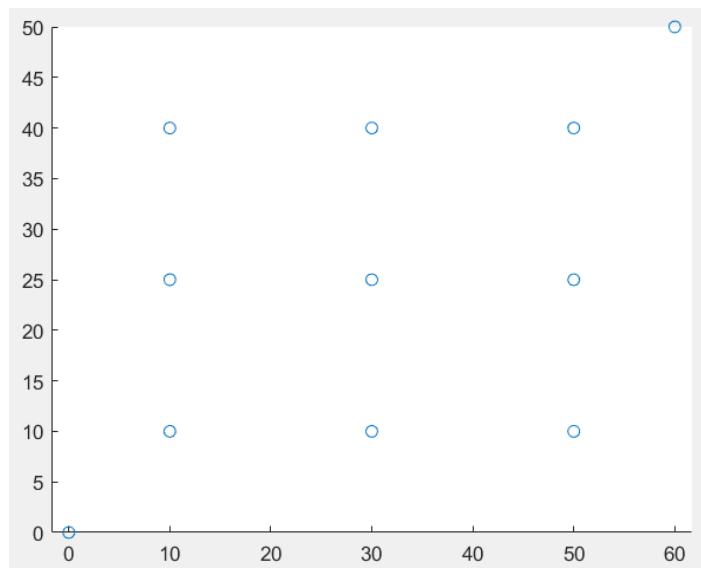
$$\theta = \tan^{-1} \left( \frac{50}{60} \right) - \tan^{-1} \left( \frac{y_1 - y_0}{x_1 - x_0} \right)$$

### 3.2 Tray Holes Calculation

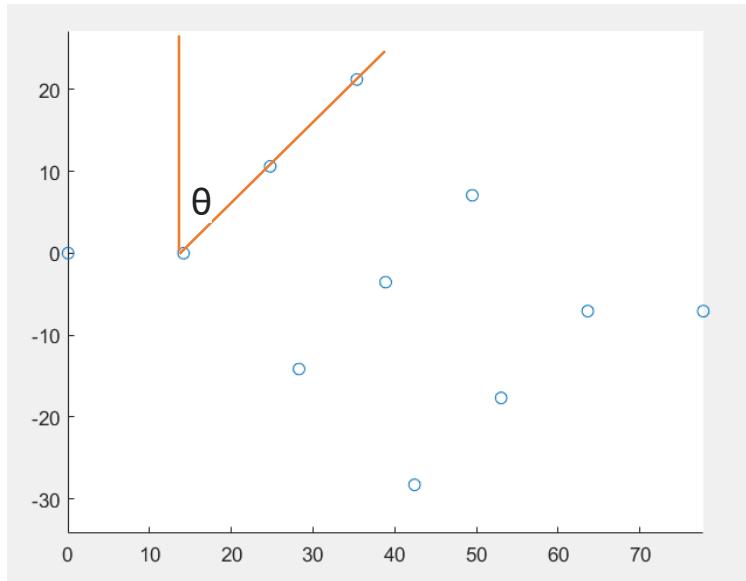
เมื่อได้จุด Reference และมุมที่ทำกับแกนอ้างอิงจากการทำ Tray Localization มาแล้ว การคำนวณจุดทั้ง 9 จุดของถอดวางจึงนำมุมที่ถอดวางทำกับแกนอ้างอิงมาใส่ในรูปของ Inverse Rotation Matrix ดังสูตรต่อไปนี้

$$R'_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

จากนั้นนำ Rotation Matrix มาคูณจุด 9 จุดของถอดจะได้ตำแหน่ง Pick And Place ทั้ง 9 จุดที่หันยนต์แกน X และ Y



รูปที่ 62 แสดงการจำลองตำแหน่งของ tray ทั้ง 9 จุด โดยที่มีจุดข้ายล่าง (0,0) เป็นจุด reference และจุดขวาบน (60,50) เป็นจุดท้ายน้ำมุนกับจุด reference ที่ใช้ในการเลือกเพื่อหาองศาและแสดง 9 จุดของ tray โดยองศาที่ได้คือ 0



รูปที่ 63 แสดงการจำลองตำแหน่งของ tray ทั้ง 9 จุด โดยที่มีจุดล่างสุด  $(0,0)$  เป็นจุด reference และจุดขวาบน  $(77.78, -7.07)$  เป็นจุดท้ายมุมกับจุด reference ที่ใช้ในการเลือกเพื่อหาองศาและแสดง 9 จุดของ tray โดยองศาที่ได้คือ 45

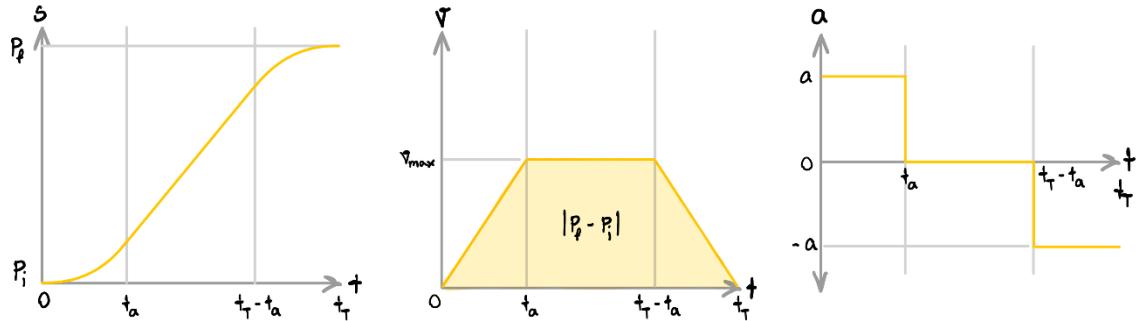
## 4. Trajectory Generation

### 4.1 Trapezoidal Trajectory

ส่วนของการคำนวณวิถีโคจร (Trajectory Generation) จะเป็นส่วนที่รับตำแหน่งจากส่วน Action planning หลังจากที่ผู้ใช้ระบุตำแหน่งและการจัดวางของสถานที่เรียบร้อยแล้ว โดยส่วนของการคำนวณวิถีโคจรจะทำการระบุสถานะของแขนกล ได้แก่ ตำแหน่ง ความเร็ว และความเร่งที่เวลาต่าง ๆ ให้สอดคล้องกับความสมพันธ์ในทางพิสิกส์ และทำการจ่ายค่าสถานะเหล่านั้นตามเวลาที่คำนวณได้ให้กับส่วนควบคุม (Controller) ต่อไป

โดยรูปแบบของวิถีโคจรที่ทางทีมวิศวกรเลือกใช้จะเป็นวิถีโคจรรูปแบบสี่เหลี่ยมคงที่เนื่องจากเป็นรูปแบบที่เร็วที่สุดเมื่อเทียบกับวิถีโคจรแบบ Polynomial และแบบ 7-Segments เพื่อให้สามารถเคลื่อนที่ในแต่ละรอบได้ภายใน 1.47 วินาทีตามความต้องการของระบบดังที่แสดงในการคำนวณในหัวข้อที่ 1 ดังนั้นในส่วนของการคำนวณวิถีโคจรจึงแบ่งได้ออกเป็น 2 ส่วน ดังนี้

#### 4.1.1 Trajectory Evaluation



รูปที่ 64 กราฟตำแหน่ง (ซ้าย) ความเร็ว (กลาง) และความเร่ง (ขวา) ของวิถีโคจรแบบสี่เหลี่ยมคงที่

ในส่วนของ Trajectory Evaluation จะเป็นส่วนที่จะคำนวณหาค่าระยะเวลาทั้งหมดที่จะต้องใช้ในการเคลื่อนที่แต่ละรอบ ( $t_T$ ) และระยะเวลาที่จะต้องใช้ในการเร่งส่วน End effector ของแขนกล ( $t_a$ ) โดยพิจารณาจากระยะทางระหว่างตำแหน่ง 2 ตำแหน่งที่ถูกป้อนเข้ามา และความเร็วสูงสุด ( $v_{max}$ ) 1,000 mm/s กับความเร่ง (a) 2,000 mm/s<sup>2</sup> ที่ระบบจะต้องทำดังที่ได้แสดงไว้ในหัวข้อที่ 1 ซึ่งจากความชันของกราฟ เวลา - ความเร็วในช่วงที่ End effector กำลังถูกเร่ง จะได้ว่า

$$t_a = \frac{v_{max}}{a}$$

และจากพื้นที่ใต้กราฟ จะสามารถคำนวณเวลาที่ต้องใช้ในการเคลื่อนที่แต่ละรอบได้ ดังนี้

$$\begin{aligned} |P_f - P_i| &= \frac{1}{2} v_{max} (2t_T - 2t_a) \\ &= v_{max} (t_T - \frac{v_{max}}{a}) \\ t_T &= \frac{v_{max}^2 + a|P_f - P_i|}{av_{max}} \end{aligned}$$

จากสมการดังกล่าวจึงสามารถนำมาเขียนเป็นฟังก์ชันภาษา C เพื่อนำไปใช้งานบน STM32 โดยจะรับจุดที่ไม่ซ้ำกัน 2 จุดเพื่อทำการคำนวณเวลาทั้งหมดและเวลาที่ใช้เร่ง ดังภาพ

```
void TrapezoidalTraj_PreCal(int16_t start_pos, int16_t final_pos)
{
    if (start_pos != final_pos)
    {
        float s = final_pos - start_pos;

        t_acc = v_max/a;
        t_total = (pow(v_max, 2)+a*fabs(s)) / (a*v_max);
    }
}
```

รูปที่ 77 ฟังก์ชัน TrapezoidalTraj\_PreCal ซึ่งทำหน้าที่เป็นส่วน Trajectory evaluation  
ของวิถีโคจรแบบสี่เหลี่ยมคงที่

#### 4.1.2 Trajectory State Calculation

ส่วนของ Trajectory State Calculation จะเป็นส่วนที่รับเวลา ( $t$ ) มาเพื่อคำนวณสถานะต่าง ๆ ของระบบได้แก่ ตำแหน่ง ความเร็ว และความเร่ง ซึ่งลักษณะของสมการจะขึ้นอยู่กับเวลาที่ถูกป้อนเข้ามาเปรียบเทียบกับเวลาที่ใช้ในการเร่งและเวลาทั้งหมดใน 1 รอบตามที่ได้คำนวณมาจาก Trajectory Evaluation โดยสำหรับในกรณีทั่วไป ( $2t_a < t_T$ ) ลักษณะของการเคลื่อนที่จะแบ่งออกเป็น 3 ช่วง ดังนี้

##### 4.1.2.1 กรณีทั่วไป ( $2t_a < t_T$ )

สำหรับในกรณีทั่วไปลักษณะของการเคลื่อนที่จะแบ่งออกเป็น 3 ช่วง ดังนี้

###### 4.1.2.1.1 เมื่อ $t \leq t_a$

จะมีลักษณะเป็นการเคลื่อนที่แบบมีความเร่งคงที่โดยมีความเร็วต้นเป็น 0 ความเร่งในช่วงนี้จึงมีค่าเป็น  $a$  ตลอด และกราฟความเร็วจะมีลักษณะเป็นเส้นตรงที่มีความชันเป็นบวก ดังสมการ

$$\dot{q} = at$$

และมีตำแหน่งดังนี้

$$q = P_i + \frac{1}{2}at^2$$

###### 4.1.2.1.2 เมื่อ $t_a < t < t_T - t_a$

จะมีลักษณะเป็นการเคลื่อนที่แบบมีความเร็วคงที่โดย ดังนั้นความเร่งในช่วงเวลานี้จะเป็น 0 และมีความเร็วเป็นค่าคงที่และตำแหน่งเป็นสมการเส้นตรงตามเวลาที่ถูกเลื่อนไป  $t_a$  ดังสมการ

$$\dot{q} = at_a$$

และ

$$q = P_i + \frac{1}{2}at_a^2 + at_a(t - t_a)$$

#### 4.1.2.1.3 เมื่อ $t_T - t_a \leq t \leq t_T$

จะมีลักษณะของเคลื่อนที่เป็นแบบมีความหน่วงคงที่โดยมีความเร็วตันเป็น  $at_a$  ความเร่งในช่วงนี้จึงมีค่าเป็น  $-a$  และกราฟความเร็วและตำแหน่งเป็นกราฟที่ถูกเลื่อนไป  $t_T - t_a$  ดังนี้

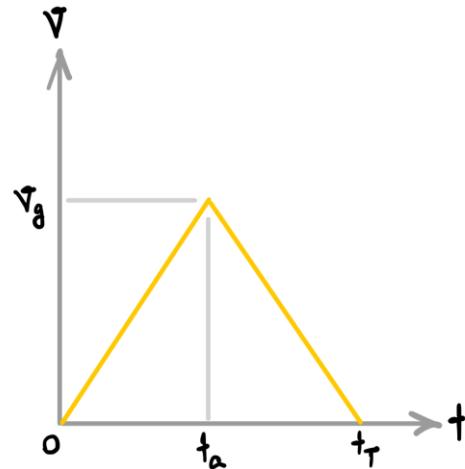
$$\begin{aligned}\dot{q} &= at_a - a(t - (t_T - t_a)) \\ &= at_T - at\end{aligned}$$

และ

$$q = P_i + at_T t + at_a t_T - at_a^2 - \frac{1}{2}a(t^2 + t_T^2)$$

#### 4.1.2.2 กรณีพิเศษ ( $2t_a \geq t_T$ )

สำหรับกรณีที่  $2t_a \geq t_T$  จะทำให้ลักษณะของกราฟเวลา - ความเร็วเป็นรูปสามเหลี่ยม การเคลื่อนที่จะแบ่งออกเป็น 2 ช่วง ได้แก่ ช่วงเร่งและช่วงหน่วง นอกจากนี้ความเร็วของ End effector จะไม่ได้ขึ้นไปจนถึง  $V_{max}$  เวลาที่ใช้เร่งและเวลาที่ต้องใช้ในการเคลื่อนที่ในแต่ละรอบจึงเปลี่ยนไปดังนี้



รูปที่ 65 กราฟความเร็วในกรณีพิเศษ

จากพื้นที่ใต้กราฟ จะสามารถคำนวณเวลาที่ต้องใช้ในการเคลื่อนที่แต่ละรอบในกรณีพิเศษได้ ดังนี้

$$\begin{aligned}|P_f - P_i| &= \frac{1}{2}at_a t_T \\ t_T &= \sqrt{\frac{4|P_f - P_i|}{a}}\end{aligned}$$

โดย  $t_T$  จะเป็น 2 เท่าของ  $t_a$  จึงได้ว่า

$$t_a = \sqrt{\frac{|P_f - P_i|}{a}}$$

เมื่อแก้ไขเวลาเร่งและเวลาทั้งหมดเมื่อการเคลื่อนที่เข้าข่ายกรณีพิเศษแล้ว จึงสามารถแบ่งการเคลื่อนที่ออกเป็น 2 ช่วงได้ ดังนี้

### 3.2.2.1 เมื่อ $t \leq t_a$

การเคลื่อนที่จะเป็นการเคลื่อนที่แบบความเร่งคงที่และความเร่งเป็นบวก ดังนั้นความเร่งจึงมีค่า  $a$  โดยมีสมการความเร็วและตำแหน่งดังแสดง

$$\dot{q} = at$$

และ

$$q = P_i + \frac{1}{2}at^2$$

### 4.1.2.2.2 เมื่อ $t_a \leq t \leq t_T$

การเคลื่อนที่จะมีลักษณะเป็นการเคลื่อนที่แบบมีความหน่วงคงที่ ดังนั้นความเร่งจึงมีค่า  $-a$  โดยมีสมการความเร็วและตำแหน่งที่ถูกเลื่อนไป  $t_a$  ดังแสดง

$$\dot{q} = at_a - at$$

และ

$$q = P_i + 2at_a t - \frac{1}{2}at^2 - at_a^2$$

นอกจากนี้ฟังก์ชัน Trajectory State Calculation ยังต้องสามารถรองรับการคำนวณเมื่อมีการเคลื่อนที่จากตำแหน่งที่มีค่ามากกว่าไปยังตำแหน่งที่มีค่าน้อยกว่า (การกระจัดเป็นลบ) จึงได้มีการเพิ่มตัวแปร dir ที่จะคุณอยู่ด้านหน้าพจน์ของตำแหน่งที่เพิ่มมาจาก  $P_i$  ในทุกรณี ซึ่ง dir จะมีค่าเป็น 1 เมื่อการกระจัดเป็นบวก และมีขนาดเป็น -1 เมื่อการกระจัดเป็นลบ

```

void TrapezoidalTraj_GetState(int16_t start_pos, int16_t final_pos, uint32_t t_us)
{
    if (start_pos != final_pos)
    {
        float t = t_us/1000000.0;

        float s = final_pos - start_pos;
        int8_t dir = 1;
        if (s < 0)
        {
            dir = -1;
        }

        if (2*t_acc < t_total) // General Case
        {
            if (t <= t_acc)
            {
                qddot_des = dir*a;
                qdot_des = dir*a*t;
                q_des = start_pos + dir*(0.5*a*pow(t,2));
            }
            else if (t_acc < t && t < (t_total - t_acc))
            {
                qddot_des = 0;
                qdot_des = dir*a*t_acc;
                q_des = start_pos + dir*(0.5*a*pow(t_acc,2) + a*t_acc*(t - t_acc));
            }
            else if ((t_total - t_acc) <= t && t <= t_total)
            {
                qddot_des = -dir*a;
                qdot_des = dir*a*(t_total - t);
                q_des = start_pos + dir*(a*t_total*t+a*t_acc*t_total-a*pow(t_acc,2)-0.5*a*(pow(t,2)+pow(t_total,2)));
            }
        }
        else // Triangle Case
        {
            t_acc = 0.5*sqrt(4*fabs(s)/a);
            t_total = 2*t_acc;

            if (t <= t_acc)
            {
                qddot_des = dir*a;
                qdot_des = dir*a*t;
                q_des = start_pos + dir*(0.5*a*pow(t,2));
            }
            else if (t_acc < t && t < t_total)
            {
                qddot_des = -dir*a;
                qdot_des = dir*a*(2*t_acc - t);
                q_des = start_pos + dir*(2*a*t_acc*t-0.5*a*pow(t,2)-a*pow(t_acc,2));
            }
        }
    }
}

```

รูปที่ 66 พังก์ชัน TrapezoidalTraj\_GetState ซึ่งทำหน้าที่เป็นส่วน Trajectory state calculation ของวิถีโคจรแบบสี่เหลี่ยมคงที่

## 4.2 Quintic Polynomial Trajectory

ภายหลังที่ได้มีการทดสอบความต้องการให้ระบบสามารถรับได้ภายในระยะเวลา 25 วินาทีอกไป เพื่อพัฒนาให้แขนกลมีความแม่นยำสูงที่สุดและลดภาระให้กับ Hardware ของระบบ ทางทีมวิศวกรจึงเปลี่ยนมาใช้วิถีโคจรแบบ Quintic Polynomial ที่สามารถกำหนดให้ระบบค่อย ๆ เร่งจากความเร็ว 0 mm/s<sup>2</sup> และค่อย ๆ หนงจนหยุดกับที่ได้ ต่างจากวิถีโคจรแบบสี่เหลี่ยมคงที่จะเริ่มเร่งด้วยความเร็วคงที่ค่าหนึ่งทันทีที่เริ่มเคลื่อนที่ ทำให้ Quintic Polynomial มีการกระชากระถางจากมอเตอร์น้อยกว่าและมีความแม่นยำมากกว่าเนื่องจากความเร็วที่จุดสุดท้ายของวิถีโคจรมะเป็น 0 ทำให้ Overshoot ต่ำกว่าวิถีโคจรแบบสี่เหลี่ยมคงที่

เช่นเดียวกับวิถีโคจรแบบสี่เหลี่ยมคงที่ วิถีโคจรแบบ Quintic Polynomial ประกอบไปด้วยส่วนการคำนวณค่าคงที่ และส่วนการคำนวณสถานะที่เวลาต่าง ๆ ดังนี้

#### 4.2.1 Trajectory Evaluation

วิถีโคจรแบบ Quintic Polynomial จะพิจารณาตำแหน่งของระบบที่เวลาต่าง ๆ ด้วยสมการพหุนามกำลัง 5 ดังนั้นสถานะของระบบที่เวลาต่าง ๆ จึงสามารถเขียนเป็นสมการพหุนามได้ดังนี้

$$\text{ตำแหน่ง: } q = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

$$\text{ความเร็ว: } \dot{q} = 5c_5 t^4 + 4c_4 t^3 + 3c_3 t^2 + 2c_2 t + c_1$$

$$\text{ความเร่ง: } \ddot{q} = 20c_5 t^3 + 12c_4 t^2 + 6c_3 t + 2c_2$$

เมื่อแทนเงื่อนไขขอบเขตโดยกำหนดให้ความเร็วและความเร่งที่จุดเริ่มต้นและจุดสิ้นสุดมีค่าเป็น 0 กำหนดให้ตำแหน่งเริ่มต้นเป็น  $q_i$  ตำแหน่งสิ้นสุดเป็น  $q_f$  เวลาที่จุดสิ้นสุดมีค่าเป็น  $T$  จะสามารถจัดรูปสมการให้ในรูปของแมทริกซ์ได้ว่า

$$\begin{bmatrix} q_i \\ q_f \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ T^5 & T^4 & T^3 & T^2 & T & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20T^3 & 12T^2 & 6T & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}$$

ซึ่งเมื่อแก้สมการอกรากแล้วจะค่าของสัมประสิทธิ์ต่าง ๆ ดังนี้

$$\begin{bmatrix} c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} \frac{6(q_f - q_i)}{T^5} \\ \frac{-15(q_f - q_i)}{T^4} \\ \frac{10(q_f - q_i)}{T^3} \\ 0 \\ 0 \\ q_i \end{bmatrix}$$

จะเห็นได้ว่าตัวแปร  $q_i$  และ  $q_f$  เป็นตัวแปรที่เราต้องกำหนดค่าเข้าไปในการคำนวณวิถีโคจรในแต่ละครั้งอยู่แล้ว ดังนั้นตัวแปรไม่ทราบค่าจึงมีเพียงแค่เวลาที่ใช้ในการเคลื่อนที่ทั้งหมด  $T$  ซึ่งสามารถพิจารณาได้จากการกำหนดความเร็วและความเร่งสูงสุด โดยจากค่าสัมประสิทธิ์ที่ได้และสมการความเร็วจะได้ว่า

$$\dot{q} = \frac{30(q_f - q_i)t^4}{T^5} - \frac{60(q_f - q_i)t^3}{T^4} + \frac{30(q_f - q_i)t^2}{T^3}$$

เมื่อทำการหาจุดสูงสุดจากการสมการดังกล่าวจะได้ว่าที่เวลา  $T/2$  ความเร็วจะมีค่าสูงสุด ดังนั้นการพิจารณาเวลาที่ใช้ในการเคลื่อนที่ทั้งหมดจากความเร็วสูงสุดจะสามารถทำได้โดยการแทน  $T/2$  เข้าไปในสมการความเร็ว ได้ดังนี้

$$\begin{aligned} v_{max} &= \frac{30(q_f - q_i)}{16T} - \frac{60(q_f - q_i)}{8T} + \frac{30(q_f - q_i)}{4T} \\ &= \frac{15(q_f - q_i)}{8T} \\ T &= \frac{15(q_f - q_i)}{8v_{max}} \end{aligned}$$

ในทำนองเดียวกัน การพิจารณาเวลาทั้งหมดที่ใช้ในการเคลื่อนที่จากความเร่งสูงสุดจะสามารถพิจารณาได้โดยสมการ

$$T = \frac{1}{2} \sqrt{\frac{40\sqrt{3}|q_f - q_i|}{3a_{max}}}$$

เมื่อได้เวลาที่ใช้ในการเคลื่อนที่ทั้งหมดที่พิจารณาจากความเร็วสูงสุดและความเร่งสูงสุดแล้ว จึงนำมาเปรียบเทียบกัน โดยจะเลือกใช้เวลาที่ใช้ในการเคลื่อนที่ทั้งหมดที่มีค่าสูงสุดเพื่อให้ความเร็วและความเร่งของ End Effector ไม่เกินจากค่าสูงสุดที่กำหนดไว้

```
#include "QuinticTraj.h"
#include "Trajectory.h"

void QuinticTraj_PreCal(int16_t start_pos, int16_t final_pos, Traj* trajectory)
{
    if (start_pos != final_pos)
    {
        float s = final_pos - start_pos;

        trajectory->t_acc = 0;
        float t_total_v = (15.0*fabs(s))/v_max;
        float t_total_a = 0.5*sqrt((40*sqrt(3)*fabs(s))/(3*a));

        if(t_total_v > t_total_a)
        {
            trajectory->t_total = t_total_v;
        }
        else
        {
            trajectory->t_total = t_total_a;
        }
    }
}
```

รูปที่ 67 ฟังก์ชัน QuinticTraj\_PreCal ซึ่งทำหน้าที่เป็นส่วน Trajectory evaluation

ของวิถีโคจรแบบ Quintic Polynomial

#### 4.2.2 Trajectory State Calculation

การคำนวณสถานะที่เวลาต่าง ๆ ของระบบสามารถทำได้โดยการคำนวณค่าสัมประสิทธิ์แล้วแทนเข้าไปในสมการดังที่ได้อธิบายไว้ในหัวข้อ 4.2.1

```
void QuinticTraj_GetState(int16_t start_pos, int16_t final_pos, Traj* trajectory, uint32_t t_us)
{
    if (start_pos != final_pos)
    {
        float t = t_us/1000000.0;

        float t_total = trajectory->t_total;
        float s = final_pos - start_pos;

        float C5 = 6*s/pow(t_total,5);
        float C4 = -15*s/pow(t_total,4);
        float C3 = 10*s/pow(t_total,3);
        float C0 = start_pos;

        q_des = C5*pow(t,5) + C4*pow(t,4) + C3*pow(t,3) + C0;
        qdot_des = 5*C5*pow(t,4) + 4*C4*pow(t,3) + 3*C3*pow(t,2);
        qddot_des = 20*C5*pow(t,3) + 12*C4*pow(t,2) + 6*C3*t;
    }
}
```

รูปที่ 68 พังก์ชัน QuinticTraj\_GetState ซึ่งทำหน้าที่เป็นส่วน Trajectory state calculation

ของวิถีโคจรแบบ Quintic Polynomial

## 5. Controller

ทางทีมวิศวกรได้เลือกทำ PID Controller แบบ Velocity Form เพื่อเป็นลดปัญหาของการเกิดสัญญาณรบกวนของการทำอนุพันธ์จากพจน์ derivative โดยรับค่าจาก End effector เคลื่อนที่ไปจากวิถีโจร และส่งค่า Error ที่ได้จากการวัดค่าจากระบบเข้า PID controller จะพิสูจน์ได้เป็นสมการ ดังนี้

กำหนดให้  $K_p$  คือ ค่า gain ของ Proportional Control

$K_i$  คือ ค่า gain ของ Integral Control

$K_d$  คือ ค่า gain ของ Derivative Control

$e[k]$  คือ ค่าความผิดพลาดปัจจุบันที่ได้จากการทำ feedback loop

$e[k - 1]$  คือ ค่าความผิดพลาดอดีตที่ได้จากการรับค่าความผิดพลาดก่อนหน้า ( $e[k]$ )

$e[k - 2]$  คือ ค่าความผิดพลาดอดีตที่ได้จากการรับค่าความผิดพลาดก่อนหน้า ( $e[k - 1]$ )

$t$  คือ เวลาในหน่วยวินาที (s)

จากสมการระบบควบคุมแบบ PID ที่การคำนวณครั้งที่  $k$

$$\begin{aligned}\vec{u}[k] &= \vec{K}_p(e[k]) + \vec{K}_i \left( \sum_{i=0}^k e[i] \right) + \vec{K}_d(e[k] - e[k-1]) \\ &= \vec{K}_p e[k] + \vec{K}_i e[k] + \vec{K}_i \left( \sum_{i=0}^{k-1} e[i] \right) + \vec{K}_d(e[k] - e[k-1])\end{aligned}$$

และจากสมการระบบควบคุมแบบ PID ที่การคำนวณครั้งที่  $k-1$

$$\vec{u}[k-1] = \vec{K}_p e[k-1] + \vec{K}_i \left( \sum_{i=0}^{k-1} e[i] \right) + \vec{K}_d(e[k-1] - e[k-2])$$

จะได้ว่า

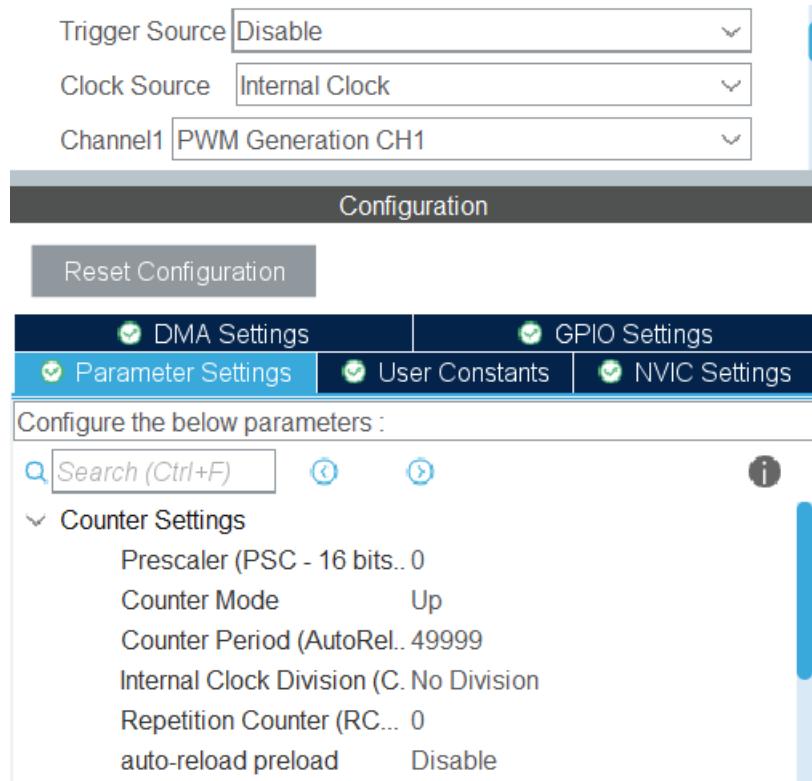
$$\vec{u}[k-1] = \vec{K}_p e[k-1] + \vec{u}[k] - \vec{K}_p e[k] - \vec{K}_i e[k] - \vec{K}_d(e[k] - e[k-1]) + \vec{K}_d(e[k-1] - e[k-2])$$

$$= \vec{u}[k] - (\vec{K}_p - \vec{K}_i e - \vec{K}_d)e[k] + (\vec{K}_p + 2\vec{K}_d)e[k-1] - \vec{K}_d e[k-2]$$

$$\vec{u}[k] = \vec{u}[k-1] + (\vec{K}_p - \vec{K}_i e - \vec{K}_d)e[k] - (\vec{K}_p + 2\vec{K}_d)e[k-1] + \vec{K}_d e[k-2]$$

## การเลือกใช้ Timer

- Timer 1 : ใช้ในการจ่าย Duty Cycle ให้กับมอเตอร์ และไม่ทำการ Pre-scale เพราะฉะนั้นจะได้ค่าความถี่ที่ตั้งเท่ากับ 100MHz จากนั้นตั้ง counter period เท่ากับ 49999 ซึ่งสามารถปรับค่าความละเอียดของ Duty Cycle ได้ถึง 0.5 เพราะฉะนั้น Timer 1 ที่มี 16 bits จึงเพียงพอต่อการจ่าย Duty Cycle ให้กับมอเตอร์



รูปที่ 69 แสดงการตั้งค่าของการจ่าย Duty cycle ให้กับมอเตอร์

- Timer 3 : ใช้ในการอ่านค่า Encoder เนื่องจาก requirement กำหนดให้มี พื้นที่ในการทำงานอยู่ในช่วง 700 มิลลิเมตร และมี Percent overshoot ไม่เกิน 2% จึงได้คำนวณเพื่อซ้ายและขวา 25 มิลลิเมตร ซึ่งจะได้ว่าระบบเบนกลจะมีช่วงพื้นที่การทำงานอยู่ที่ 750 มิลลิเมตรโดยขนาดของข้อมูลที่ต้องใช้ในการเก็บข้อมูลตำแหน่งตั้งกลล่าง โดย encoder mode จะเลือกเป็น TI1 และ TI2 เพื่อเพิ่มความละเอียดเป็น 4 จึงสามารถคำนวณได้โดย

กำหนดให้ พื้นที่ในการทำงาน คือ 750 mm

เส้นรอบวงของเพลา คือ  $30.558\pi$

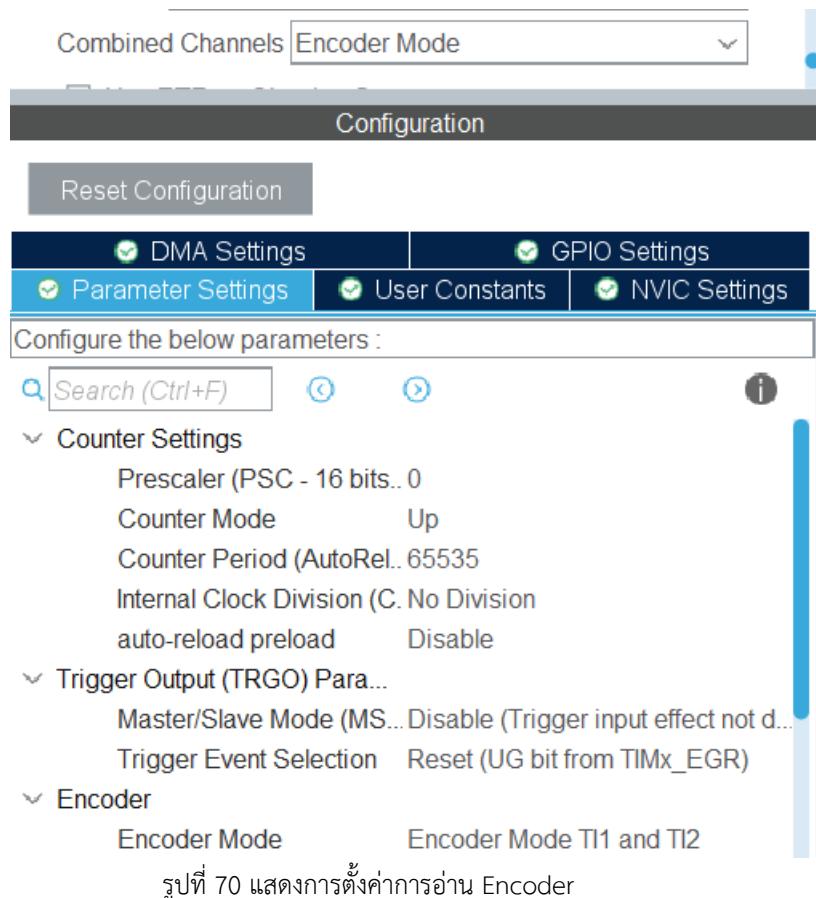
จำนวน Pulse ใน 1 รอบที่มอเตอร์หมุน คือ 2048 Pulse/Round

ค่าความละเอียดของ encoder คือ 4

จะได้ขนาดของข้อมูลว่า

$$\frac{750}{30.558 \cdot \pi} \cdot 2048 \cdot 4 = 63999.47446$$

จากสมการจะเห็นได้ว่า Timer 3 ที่มี 16 bits ซึ่งสามารถเก็บข้อมูลได้ 65536 ค่าจึงเพียงพอต่อการรับค่าจาก Encoder



- Timer 5 : ใช้ในการนับเวลาในหน่วย micro second โดยที่นำ Timer 32 bits มาเพิ่มจำนวน bits ให้เป็น 64 bits ซึ่งเหมาะสมต่อการใช้งาน

Internal Clock

## Configuration

Reset Configuration

<input checked="" type="checkbox"/> NVIC Settings	<input checked="" type="checkbox"/> DMA Settings
<input checked="" type="checkbox"/> Parameter Settings	<input checked="" type="checkbox"/> User Constants

Configure the below parameters :

Search (Ctrl+F)   

Counter Settings

- Prescaler (PSC - 16 bits .. 0)
- Counter Mode Up
- Counter Period (AutoRel... 99999)
- Internal Clock Division (C.. No Division)
- auto-reload preload Disable

Trigger Output (TRGO) Param...

- Master/Slave Mode (MS... Disable (Trigger input effect not de..)
- Trigger Event Selection Reset (UG bit from TIMx\_EGR)

รูปที่ 71 แสดงการตั้งค่าการการนับเวลาในหน่วย microsecond

## 6. End effector

```
void eff_write(uint8_t* cmd2){  
    HAL_I2C_Master_Transmit_IT(&hi2c1, 0x15 << 1, cmd2, 4);  
}  
  
void eff_write2(uint8_t* cmd3){  
    HAL_I2C_Master_Transmit_IT(&hi2c1, 0x15 << 1, cmd3, 8);  
}  
  
void eff_read(){  
    HAL_I2C_Master_Receive_IT(&hi2c1, 0x15 << 1, effstatus, 1);  
}
```

รูปที่ 72 พัฟ์ชั่นสั่งงานและรับค่าสถานะของ End effector

```
//EndEffector  
uint8_t softReset_cmd[] = {0x00,0xFF,0x55,0xAA};  
uint8_t emerMode_cmd[] = {0xF1,0x99,0x99,0x99};  
uint8_t exitEmer_cmd[] = {0xE5,0x7A,0xFF,0x81};  
uint8_t testMode_cmd[] = {0x01,0x10,0x99,0x99};  
uint8_t exitTest_cmd[] = {0x01,0x00,0x99,0x99};  
uint8_t runMode_cmd[] = {0x10,0x13,0x99,0x99};  
uint8_t exitRun_cmd[] = {0x10,0x8C,0x99,0x99};  
uint8_t pickup_cmd[] = {0x10,0x5A,0x99,0x99};  
uint8_t place_cmd[] = {0x10,0x69,0x99,0x99};  
uint8_t AllOff_cmd[] = {0x01,0x00,0x99,0x99,0x10,0x8C,0x99,0x99};  
  
//uint8_t exitEmerAndAllOff = exitEmer  
uint8_t exitEmerAndtestMode_cmd[] = {0xE5,0x7A,0xFF,0x81,0x01,0x10,0x99,0x99};  
uint8_t exitEmerAndrunMode_cmd[] = {0xE5,0x7A,0xFF,0x81,0x10,0x13,0x99,0x99};  
uint8_t exitEmerAndpickup_cmd[] = {0xE5,0x7A,0xFF,0x81,0x10,0x5A,0x99,0x99};  
uint8_t exitEmerAndplace_cmd[] = {0xE5,0x7A,0xFF,0x81,0x10,0x69,0x99,0x99};
```

รูปที่ 73 ชุดคำสั่งต่างๆที่ใช้กับพัฟ์ชั่นสั่งงาน End effector

### 6.1 การสื่อสารกับ End effector เป็นต้น

ในการสั่งงานและรับค่าสถานะของ End effector จะใช้ทั้งหมด 3 พัฟ์ชั่นด้วยกัน คือ

1. eff\_write สำหรับการสั่งงาน End effector ด้วย 4 Sequence หรือน้อยกว่า
2. eff\_write2 สำหรับการสั่งงาน End effector ด้วย 8 Sequence หรือน้อยกว่า
3. eff\_read สำหรับการรับค่า End effector status

โดยพัฟ์ชั่นในการสั่งงานและรับค่าสถานะทั้งหมดจะใช้เป็นแบบ Interrupt เพื่อไม่ให้ไปขัดการทำงานของส่วนอื่น

## 6.2 การสั่งงาน End effector ด้วย Base system

หลักการทำงานของการสั่งงาน End effector คือ การเช็คขอบของ Register เมื่อได้รับรู้ว่าค่าใน Register เกิดการเปลี่ยนแปลง (เกิดขอบ) ก็จะยก Flag สำหรับการสั่งงานขึ้น และเข้าไปทำงานตาม Flag นั้นๆ

```
eff_c = registerFrame[2].U16;
if(eff_l != eff_c)
{
    eff_action = 1;
}
```

รูปที่ 74 โปรแกรมส่วนการเช็คขอบ

```
EffRegState = registerFrame[2].U16;

switch(EffRegState){
case 0b0000: //everything off
    if(eff_action == 1){
        EffAllOff_Flag = 1;
        eff_action = 0;
    }
    break;

case 0b0001: //laser on
    if(eff_action == 1){
        EffLaserOn_Flag = 1;
        eff_action = 0;
    }
    break;
```

รูปที่ 75 โปรแกรมส่วนยก Flag

```
case 0b0010: //gripper on
    if(eff_action == 1){
        EffGripperOn_Flag = 1;
        eff_action = 0;
    }
    break;

case 0b0110: //gripper picking
    if(eff_action == 1){
        EffGripperPick_Flag = 1;
        eff_action = 0;
    }
    break;

case 0b1010: //gripper placing
    if(eff_action == 1){
        EffGripperPlace_Flag = 1;
        eff_action = 0;
    }
    break;
}
```

รูปที่ 76 โปรแกรมส่วนยก Flag

```

if (emer_pushed)
{
    BaseSystem_EffAllOff();
    BaseSystem_EffLaserOn();
    BaseSystem_EffGripperOn();
    BaseSystem_EffGripperPick();
    BaseSystem_EffGripperPlace();
}

eff_l = eff_c;

```

รูปที่ 77 โปรแกรมส่วนการทำงานตาม Flag ที่ถูกยก

```

void BaseSystem_EffAllOff(){
    if(EffAllOff_Flag == 1){
        eff_write2(AllOff_cmd);
        EffAllOff_Flag = 0;
    }
}
void BaseSystem_EffLaserOn(){
    if(EffLaserOn_Flag == 1){
        eff_write(testMode_cmd);
        EffLaserOn_Flag = 0;
    }
}
void BaseSystem_EffGripperOn(){
    if(EffGripperOn_Flag == 1){
        eff_write(runMode_cmd);
        EffGripperOn_Flag = 0;
    }
}
void BaseSystem_EffGripperPick(){
    if(EffGripperPick_Flag == 1){
        eff_write(pickup_cmd);
        EffGripperPick_Flag = 0;
        registerFrame[2].U16 = 0b0010;
    }
}
void BaseSystem_EffGripperPlace(){
    if(EffGripperPlace_Flag == 1){
        eff_write(place_cmd);
        EffGripperPlace_Flag = 0;
        registerFrame[2].U16 = 0b0010;
    }
}

```

รูปที่ 78 โปรแกรมส่วนการทำงานตาม Flag ที่ถูกยก

### 6.3 การอ่านค่าสถานะของ End effector และการส่งไปยัง Base system

```
void eff_st(){
    //          READ   REG
    //alloff     0000   0000
    //laser on   1000   0001
    //gripper on 0100   0010
    //pick       0101   0110
    //picked     0111   N/A
    //place      0110   1010

    eff_read();
    effstatus = effstatus & 0b00001111;
    if (effstatus == 0b0000) {effst_mb = 0b0000;}
    else if(effstatus == 0b1000) {effst_mb = 0b0001;}
    else if(effstatus == 0b0100) {effst_mb = 0b0010;}
    else if(effstatus == 0b0101) {effst_mb = 0b0110;}
    //else if(effstatus == 0b0111) {effst_mb = }
    else if(effstatus == 0b0110) {effst_mb = 0b1010;}

    registerFrame[2].U16 = effst_mb;
}
```

รูปที่ 79 ฟังก์ชันส่งสถานะ End effector ไปที่ Base system

หลักการทำงานคืออ่านค่าสถานะแล้วใส่ไปที่ตัวแปร Global effstatus หลังจากนั้นก็ทำการ bitwise operation เพื่อเอาแค่ bit ที่ 0 – 3 และหลังจากนั้นก็ใช้ if else เพื่อเช็คและเปลี่ยน Register เป็นค่าที่ Base system ต้องการ พอกเปลี่ยนเรียบร้อยแล้วก็รอให้ Modbus Protocol Worker ทำงานตาม Routine ปกติ

### 6.4 การทำงานของ End effector เมื่อเกิดการกดปุ่ม Emergency

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
    if(GPIO_Pin == GPIO_PIN_12 && HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12) == 0){
        emer_pushed = 1;
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_13, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_14, 0);
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0);
        effreg_temp = registerFrame[2].U16;
        effstatus_temp = effstatus[0];
        eff_write(emergency_cmd);
        HAL_TIM_SET_COMPARE(&htim1,TIM_CHANNEL_1,0);
        ControllerFlagMem = ControllerFlag;
        ControllerFlag = 0;
        state = Idle;
        emerpass = 1;
    }
}
```

รูปที่ 80 โปรแกรม Interrupt กรณีปุ่ม Emergency ถูกกด

```

if(GPIO_Pin == GPIO_PIN_12 && HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12) == 1){
    registerFrame[2].U16 = effreg_temp;
    effstatus[0] = effstatus_temp;

    EffRegState = registerFrame[2].U16;
    switch(EffRegState){
        case 00000: //everything off
            eff_write(exitEmer_cmd);
            break;
        case 00001: //laser on
            eff_write2(exitEmerAndtestMode_cmd);
            break;
        case 00010: //gripper on
            eff_write2(exitEmerAndrunMode_cmd);
            break;
        case 00010: //gripper picking
            registerFrame[2].U16 = 0b0010;
            eff_write2(exitEmerAndpickup_cmd);
            break;
        case 0b1010: //gripper placing
            registerFrame[2].U16 = 0b0010;
            eff_write2(exitEmerAndplace_cmd);
            break;
    }
}

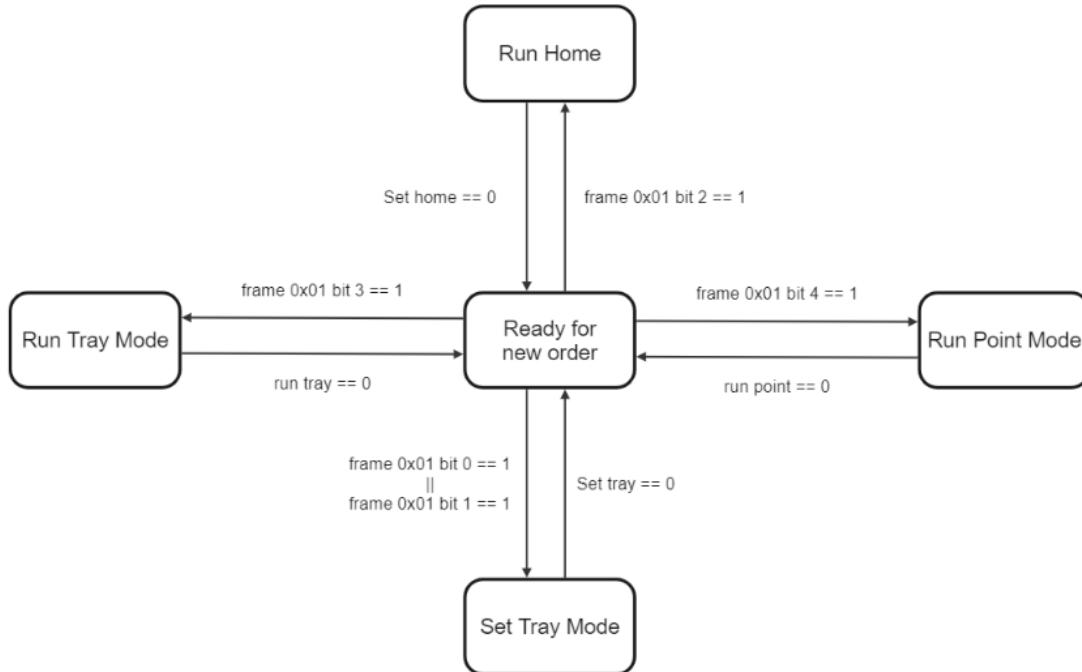
```

รูปที่ 81 โปรแกรม Interrupt กรณีปุ่ม Emergency ถูกปล่อย

หลักการทำงานคือเมื่อกดปุ่ม Emergency และให้เก็บค่าสถานะ และค่าใน Register เอ้าไว้ในตัว แปรซึ่งคราว แล้วสั่งให้ End effector เข้าโหมด Emergency เมื่อปล่อยปุ่ม Emergency ก็ให้คืนค่าสถานะและค่าใน Register และสั่งให้ End effector ออกจากโหมด Emergency และสั่งให้ทำงานตามที่ก่อนหน้าเคยทำค้าง เอ้าไว้โดยดูจากค่าใน Register ที่ได้คืนมา

## 7. Base System Integration

overall

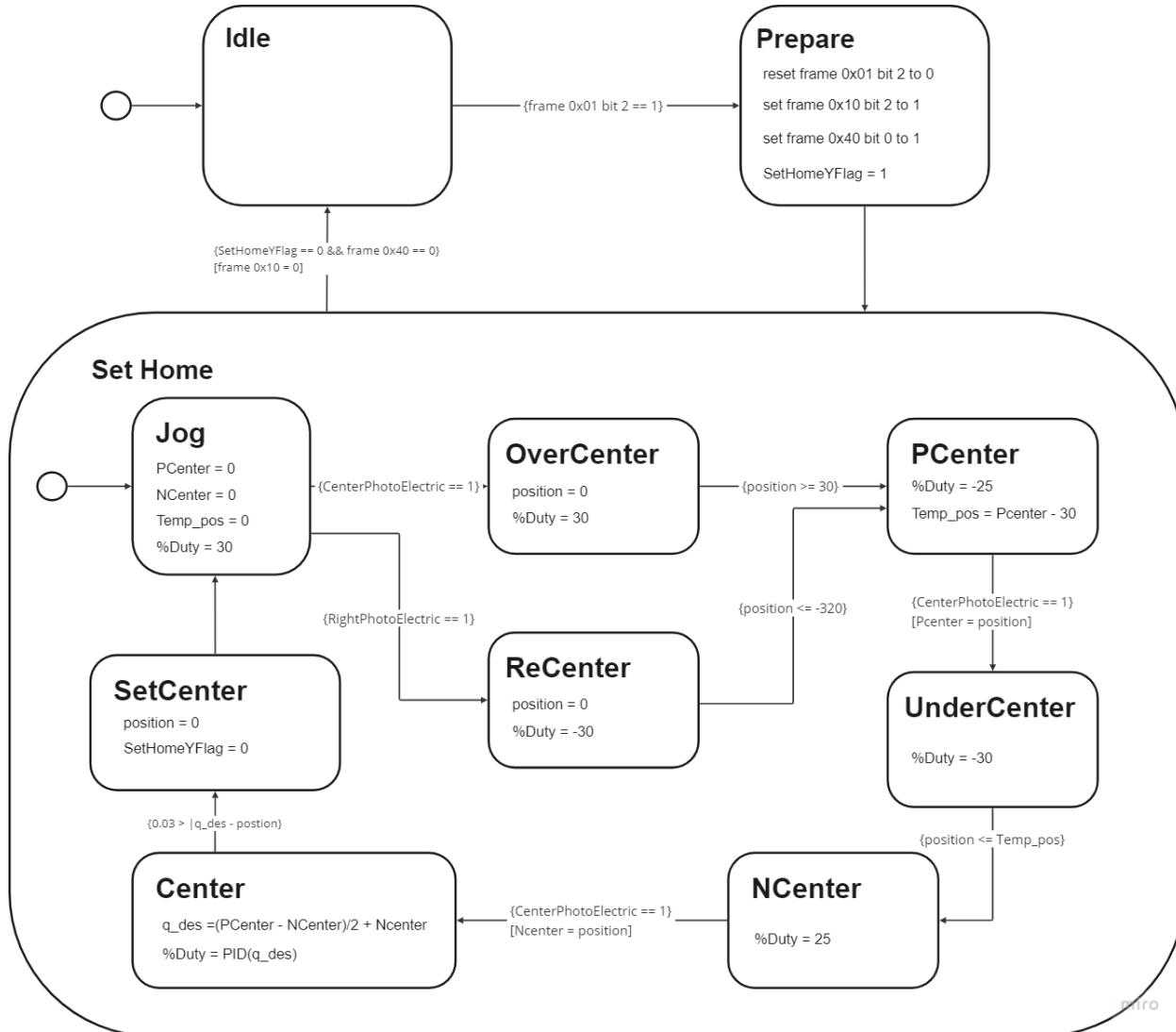


รูปที่ 82 แสดงภาพรวมของการสื่อสาร Modbus protocol

จากรูปที่ 82 เนื้องจากระบบของแกน y จะต้องติดต่อสื่อสารกับระบบของแกน x ได้ โดยระบบ Base system ที่นำมาใช้งานนั้นถูกกำหนดการเชื่อมต่อด้วย Modbus Protocol โดยต้องมีการส่ง Heartbeat เพื่อเป็นการสื่อสารกันระหว่างหุ่นยนต์แกน x และหุ่นยนต์แกน y โดยที่ Base system จะอ่านสถานะของหุ่นยนต์แกน y ในทุกๆ 200 ms ถ้าหุ่นยนต์แกน y ส่งสัญญาณมาว่า “Ya” หุ่นยนต์แกน x จะตอบกลับว่า “Hi” ก่อนจะรีเซ็ตกลับเป็น “Ya” ตามเดิม ซึ่งมีข้อมูลการทำงานต่างๆดังต่อไปนี้

## 7.1 Set Home

เมื่อกดปุ่ม Home ที่ Base system แล้ว หุ่นยนต์แกน x และหุ่นยนต์แกน y จะต้องเข้าสู่ Set point ตรงกลางที่ให้เป็นจุดเริ่มต้นหรือ home โดยสามารถแสดงการทำงานตอน Set home ได้ดังนี้



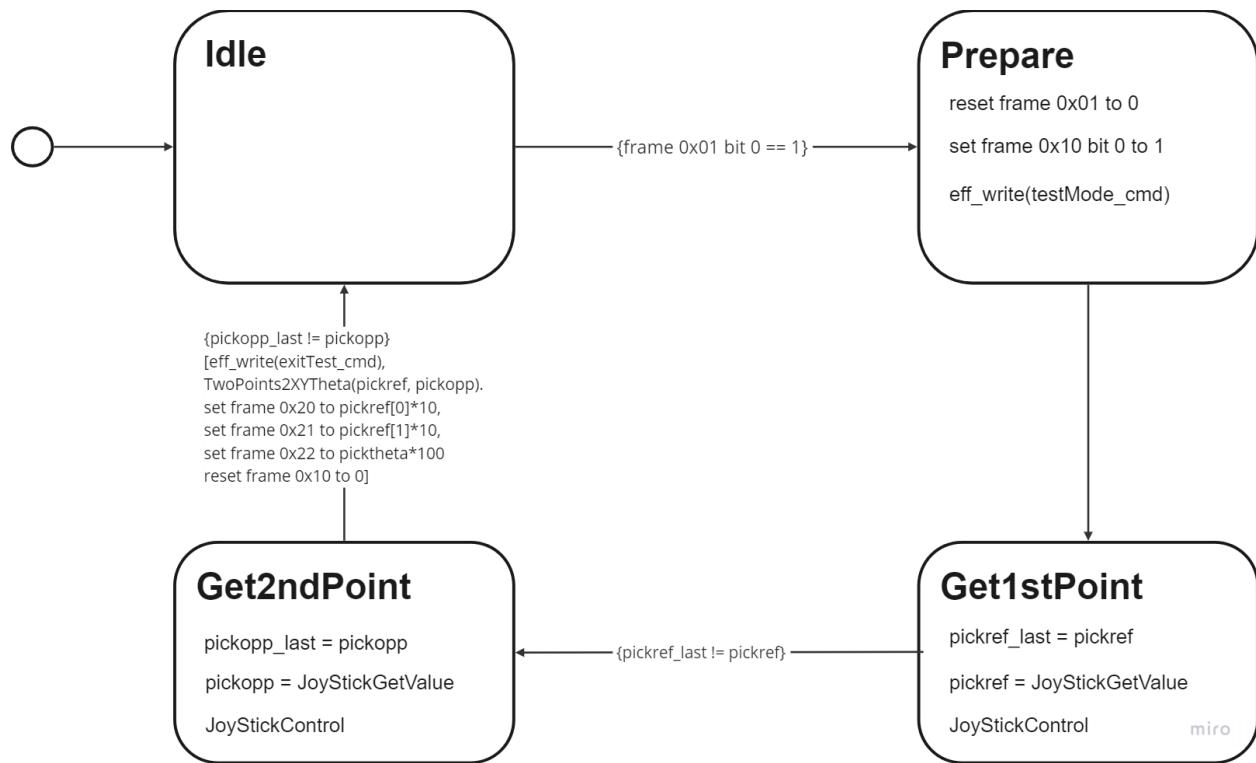
รูปที่ 83 Finite State Machine Diagram แสดงการทำงานภายในของการ Set Home

โดยการ Set Home ของแกน Y จะทำโดยการให้ End Effector เคลื่อนที่เข้าหา Photo Electric Sensor ตรงกลางจากทางด้านขวาและทางด้านซ้ายเพื่อทำการหาค่ากึ่งกลาง จากนั้นจึงใช้ PID ในการควบคุมตำแหน่ง End Effector ไปยังตำแหน่งกึ่งกลางที่คำนวณไว้

## 7.2 Set Pick/Place Tray

การกำหนดถุงหุบและถอดความสามารถทำได้โดยการใช้ Joy Stick โดยการเลื่อนตำแหน่งจะใช้ DMA ใน การเก็บค่าตำแหน่งของ Joy Stick เพื่อนำมาหาค่าเฉลี่ยสำหรับนำไปพิจารณาในการขับเคลื่อนมอเตอร์ทั้งของแกน X และแกน Y ไปยังทิศทางที่ต้องการ

การรับค่าตำแหน่งของ End Effector สามารถทำได้โดยการกดปุ่มที่ Joy Stick ต่อเนื่องกัน 10 ครั้ง โดย เมื่อกดครบ 10 ครั้งแล้ว ระบบจะทำการอ่านค่าตำแหน่งของแกน X จาก Modbus Protocol และของแกน Y จาก Encoder และหากกด Joy Stick ค้างเป็นเวลา 2 วินาที ระบบจะทำการ Set Home แกน Y

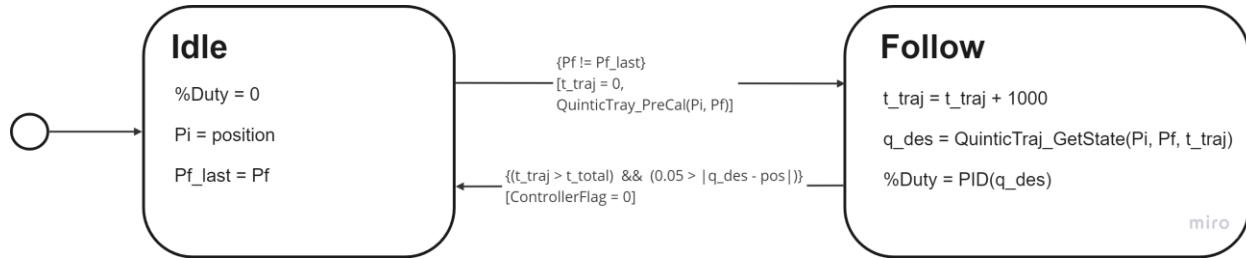


รูปที่ 84 Finite State Machine Diagram แสดงการทำงานภายในการ Set Pick Tray

เมื่อเข้าสู่โหมด Set Pick Tray หรือ Set Place Tray ระบบจะทำการเปิดเลเซอร์ที่ End Effector และ จะเปิดให้รับค่าจุด 2 จุด โดยระหว่างการรับค่าจุดแต่ละจุด ผู้ใช้จะสามารถควบคุมและกรับค่าจาก Joy Stick ได้ เมื่อผู้ใช้กรับครบทั้ง 2 จุดแล้ว ระบบจะทำการแปลงตำแหน่งจุด 2 จุดจากระบบพิกัดจากเป็นจุดอ้างอิงและมุ่งที่ ทำกับแกนอ้างอิงดังแสดงในหัวข้อ 3.1 (Tray localization) เพื่อส่งไปให้ Base System ผ่าน Modbus Protocol

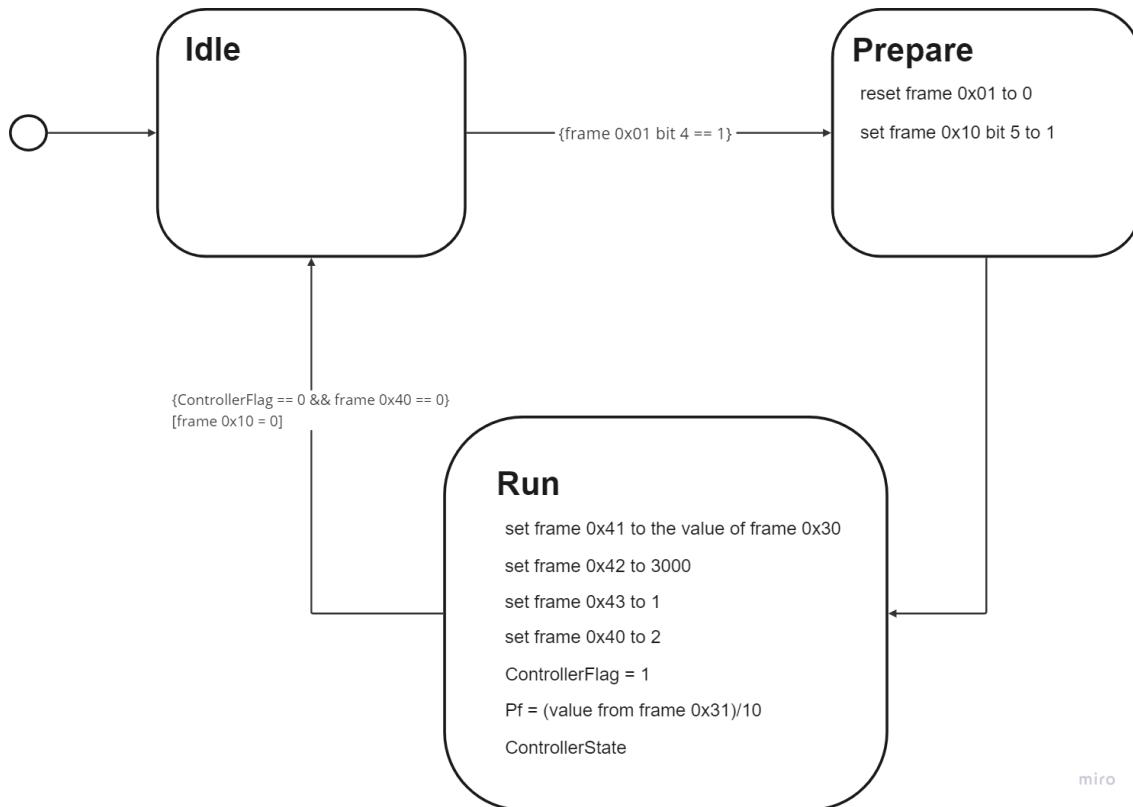
### 7.3 Run Point

ในการควบคุมตำแหน่งที่ไม่ใช่การคุ้มตำแหน่งเพื่อ Set Home จะมีการคำนวณวิธีการของเคลื่อนที่ไปที่จุดสิ้นสุดก่อน แล้วจึงนำค่าสถานะต่าง ๆ จากวิธีการมาเข้าระบบควบคุมดังแสดงในแผนภาพ



รูปที่ 85 Finite State Machine Diagram แสดงการทำงานของระบบควบคุมตำแหน่งแบบใช้วิธีการ (Controller State)

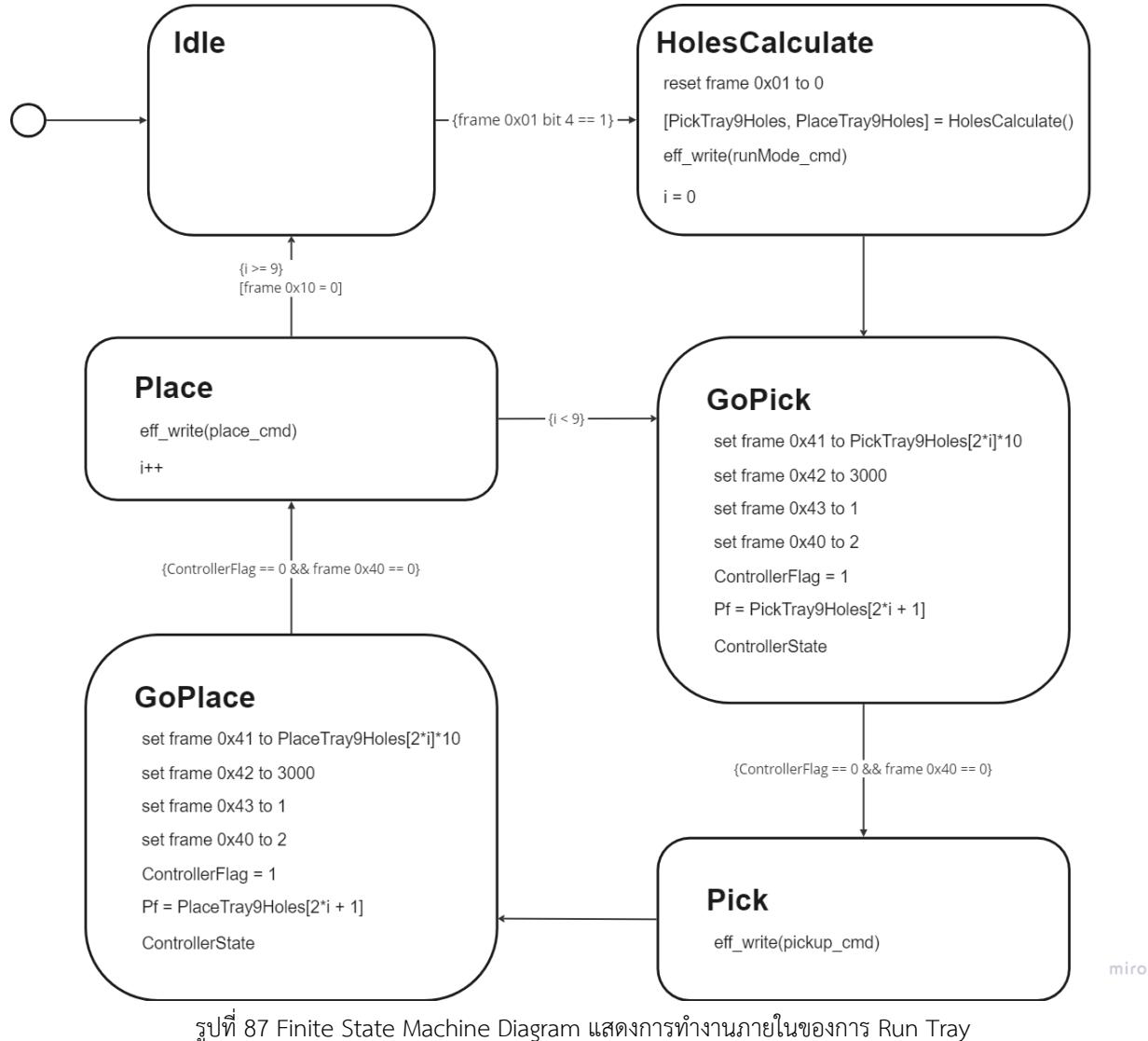
โดย Run Point จะเป็นการกำหนดตำแหน่งให้แกน X และแกน Y เคลื่อนที่จากจุดนึ่งไปยังอีกจุดที่กำหนด โดยจะใช้คำสั่งใน Frame 0x30 หรือ Goal Point X และจะทำการ Reset เมื่อ Frame 0x40 == 0 หรือว่า Run Point เสร็จเรียบร้อยแล้วโดยมีการทำงานเป็นแผนผังได้ดังนี้



รูปที่ 86 Finite State Machine Diagram แสดงการทำงานภายในของการ Run Point

## 7.4 Run Tray

การ Run Tray จะเริ่มต้นจากการคำนวณตำแหน่งรู 9 รูของแต่ละถาดวางเพื่อนำไปใช้ในการกำหนดเป้าหมายในการเคลื่อนที่ของ End Effector โดยจะกำหนดเป็นรูของถาดหยิบและถาดวางสลับกันไปเรื่อยๆ จนครบ 18 จุด

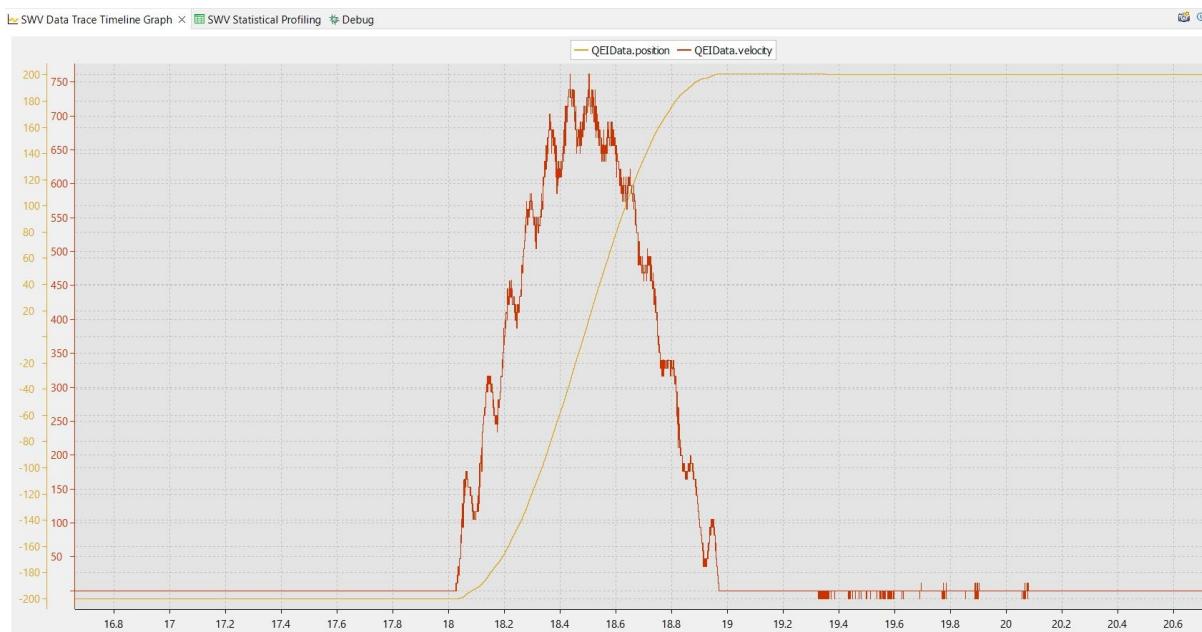


รูปที่ 87 Finite State Machine Diagram แสดงการทำงานภายในของการ Run Tray

miro

## ผลการทดสอบ

### 1. ผลการทดสอบโดยไม่ติดมวล

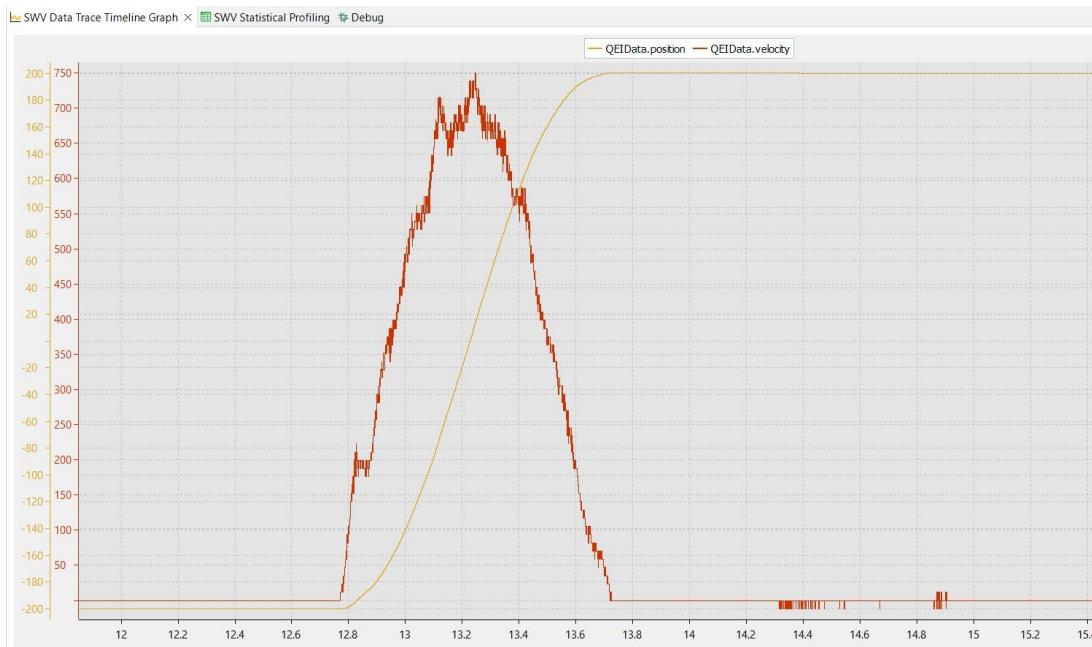


รูปที่ 88 กราฟตำแหน่งและความเร็วของการเคลื่อนที่จากตำแหน่ง -200 mm ไป 200 mm ในแกน Y

จากการทดสอบระบบโดยไม่ติดมวลโดยการสั่งให้ระบบวิ่งจากตำแหน่ง -200 mm ไป 200 mm พบร่วมระบบมี Steady State Error อยู่ที่ 0.01 mm และมี Overshoot ขึ้นไปถึงค่า 200.66 mm หรือคิดเป็นร้อยละ 0.165 ของระยะการเคลื่อนที่ทั้งหมด

ในส่วนของความเร็วระบบสามารถเคลื่อนที่ได้ด้วยความเร็วสูงสุด 761.72 mm/s และสามารถประมาณความเร่งได้จากผลต่างของความเร็วที่เวลา 18.50 วินาที (ความเร็ว 761.72 mm/s) และที่เวลา 18.35 วินาที (ความเร็ว 656.25 mm/s) จะได้ความเร่ง 703.13 mm/s<sup>2</sup> ซึ่งเป็นไปตามวิถีโคจรที่กำหนดไว้และตรงตามความต้องการของลูกค้า

## 2. ผลการทดสอบแบบติดมวล 1.5 กิโลกรัม



รูปที่ 89 กราฟตำแหน่งและความเร็วของการเคลื่อนที่จากตำแหน่ง -200 mm ไป 200 mm ในแกน Y เมื่อติดมวล 1.5 กิโลกรัม

จากการทดสอบระบบโดยไม่ติดมวลโดยการสั่งให้ระบบบว่งจากตำแหน่ง -200 mm ไป 200 mm โดยติดมวล 1.5 กิโลกรัมพบว่าระบบมี Steady State Error อุ่ยที่ 0.03 mm และมี Overshoot ขึ้นไปถึงค่า 200.39 mm หรือคิดเป็นร้อยละ 0.098 ของระบบการเคลื่อนที่ทั้งหมด

ในส่วนของความเร็วระบบสามารถเคลื่อนที่ได้ด้วยความเร็วสูงสุด 738.28 mm/s และสามารถประมาณความเร่งได้จากผลต่างของความเร็วที่เวลา 13.10 วินาที (ความเร็ว 664.651 mm/s) และที่เวลา 13.24 วินาที (ความเร็ว 738.28 mm/s) จะได้ความเร่ง  $525.92 \text{ mm/s}^2$  ซึ่งตรงตามความต้องการของลูกค้า จึงสามารถสรุปได้ว่าระบบสามารถทำงานโดยรับโหลด 1.5 กิโลกรัมได้

## Management Section

1. Progress 1 (26 April 2023) : ประมาณ 33% ของโครงการทั้งหมด ซึ่งมีรายละเอียดดังตารางที่ 1  
ตารางที่ 1 ตารางแสดงรายละเอียดการส่งมอบงาน Progress 1

ฝ่าย	รายละเอียดการดำเนินงาน
Mechanical Design (30 %)	<ol style="list-style-type: none"> <li>ออกแบบและเลือกวัสดุของโครงสร้าง (100 %)</li> <li>คำนวณโครงสร้างและวัสดุที่ใช้ในการออกแบบ (100%)</li> <li>จัดทำ CAD, DRAWING และ BOM ของโครงสร้างที่ออกแบบทั้งหมด (100 %)</li> <li>จัดซื้ออุปกรณ์ (100 %)</li> </ol>
Electrical Module (%)	<ol style="list-style-type: none"> <li>ออกแบบวงจรและเงื่อนไขการทำงาน (100 %)</li> <li>คำนวณเพื่อเลือกอุปกรณ์ที่เหมาะสมกับการใช้งาน (100 %)</li> <li>จัดทำ Schematic ของวงจร (100 %)</li> <li>จัดซื้ออุปกรณ์ (80 %)</li> </ol>
Control Engineering (15 %)	<ol style="list-style-type: none"> <li>จัดทำ System Identification (100 %)</li> <li>ออกแบบระบบ PID Controller (0 %)</li> <li>ออกแบบ Trajectory (0 %)</li> </ol>
Document & Presentation	<p>จัดทำเอกสารและสื่อการนำเสนอ สำหรับการนำเสนอ Progress 1 ซึ่งประกอบด้วย</p> <ol style="list-style-type: none"> <li>Mechanical Design <ul style="list-style-type: none"> <li>การคำนวณโครงสร้างและวัสดุที่ใช้ในการออกแบบทั้งหมดของหุ่นยนต์</li> <li>CAD, DRAWING และ BOM ของโครงสร้างทั้งหมด</li> <li>งบประมาณของอุปกรณ์ในด้าน Mechanical Design</li> </ul> </li> <li>Electrical Module <ul style="list-style-type: none"> <li>รายละเอียดการออกแบบ Schematic</li> <li>ขั้นตอนการทำงานโดยรวมของระบบ</li> <li>การคำนวณเพื่อเลือกอุปกรณ์ที่เหมาะสมกับการใช้งาน</li> <li>งบประมาณของอุปกรณ์ในด้าน Electronical Module</li> </ul> </li> <li>Control Engineering <ul style="list-style-type: none"> <li>System Identification</li> </ul> </li> </ol>

2. Progress 2 (15 March 2023) : ประมาณ 67% ของโครงการทั้งหมด ซึ่งมีรายละเอียดดังตารางที่ 2

ตารางที่ 2 ตารางแสดงรายละเอียดการส่งมอบงาน Progress 2

ฝ่าย	รายละเอียดการดำเนินงาน
Mechanical Design (100 %)	1. ประกอบโครงสร้างของหุ่นยนต์ทั้งหมด (100 %)
Electrical Module (100 %)	1. จัดทำตู้ไฟโดยจะต้องสามารถแปลงไฟฟ้าจาก 220V เป็น 12V และ 5V 2. จัดทำตู้ไฟโดยจะต้องมีระบบสายดิน (Grounding System) (100 %) 3. จัดทำตู้ไฟโดยจะต้องมีสวิตช์สำหรับ ON/OFF และมีปุ่ม Emergency สำหรับกรณีฉุกเฉิน (100 %) 4. พัฒนาโปรแกรมให้หุ่นยนต์สามารถทำงานตามคำสั่งเบื้องต้น ซึ่งได้แก่ 4.1 ควบคุมแขนของหุ่นยนต์ให้หยุดหมุนในตำแหน่งของศากี้ต้องการ (100 %) 4.2 คำสั่งในการอ่าน Encoder และ Protocol ในการควบคุม End-Effector (100 %) 4.3 ระบบ Base System ติดต่อกันด้วย MODBUS RS232 Protocol (100 %) 5. คำสั่งในการอ่าน Encoder และ Protocol ในการควบคุม End-Effector (100 %) 6. ระบบ Base System ติดต่อกันด้วย MODBUS RS232 Protocol (100 %)
Control Engineering (75 %)	1. จัดทำระบบ PID Controller (100 %) 2. จัดทำ Trajectory (100 %)
Document & Presentation	จัดทำเอกสารและสื่อการนำเสนอ สำหรับการนำเสนอ Progress 2 ซึ่งประกอบด้วย 1. Mechanical Design 1.1 รายละเอียดที่แก้ไขและปรับปรุงในฝ่าย Mechanical Design 1.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในฝ่าย Mechanical Design ที่ได้รับการแก้ไขและปรับปรุง 2. Electrical Module 2.1 รายละเอียดอุปกรณ์ที่ใช้จัดทำตู้ไฟ

	<p>2.2 รายละเอียดการพัฒนาโปรแกรมให้หุ่นยนต์สามารถทำงานคำสั่ง</p> <p>2.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในด้าน Electronical Module</p> <p>3. Control Engineering</p> <p>3.1 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในระบบ PID Controller</p> <p>3.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ใน Trajectory</p>
--	--

3. Testing 1 (29 March 2023) : ประมาณ 98% ของโครงการทั้งหมด ซึ่งมีรายละเอียดดังตารางที่ 3

ตารางที่ 3 ตารางแสดงรายละเอียดการส่งมอบงาน Testing 1

ฝ่าย	รายละเอียดการดำเนินงาน
Mechanical Design	แก้ไขและปรับปรุงโครงสร้างหุ่นยนต์เพื่อให้สามารถทำงานได้ตามเป้าหมาย
Electrical Module	แก้ไขและพัฒนาวงจรเพื่อให้หุ่นยนต์สามารถทำงานได้ตามเป้าหมายและสามารถสื่อสารรับ-ส่งข้อมูลกับ Base System ได้อย่างเสถียร
Control Engineering	แก้ไขและพัฒนาให้ระบบ PID Controller และ Trajectory สามารถควบคุมหุ่นยนต์ให้ทำงานได้ตามเป้าหมาย
Document & Presentation	<p>จัดทำเอกสารและสื่อการนำเสนอ สำหรับการนำเสนอ Testing 1 ซึ่งประกอบด้วย</p> <p>1. Mechanical Design</p> <p>1.1 การคำนวณและวิเคราะห์โครงสร้างรวมถึงวัสดุทั้งหมดของหุ่นยนต์</p> <p>1.2 CAD, DRAWING และ BOM ของโครงสร้างทั้งหมดแบบสมบูรณ์</p> <p>1.3 งบประมาณสุทธิของอุปกรณ์ในด้าน Mechanical Design</p> <p>1.4 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์พร้อมระบบทั้งหมด</p> <p>2. Electrical Module</p> <p>2.1 การคำนวณและวิเคราะห์วงจรรวมถึงอุปกรณ์ทั้งหมด</p> <p>2.2 Schematic Diagram โดยรวมของระบบ</p> <p>2.3 งบประมาณสุทธิของอุปกรณ์ในด้าน Electronical Module</p> <p>2.4 รายละเอียดอุปกรณ์ที่ใช้จัดทำตู้ไฟ</p>

	<p>2.5 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในด้าน Electronical Module</p> <p>3. Control Engineering</p> <p>3.1 System Identification</p> <p>3.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในระบบ PID Controller</p> <p>3.3 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ใน Trajectory</p>
--	--

4. Testing 2 (5 June 2023) : ประเมณ 100% ของโครงการทั้งหมด ซึ่งมีรายละเอียดดังตารางที่ 4 ตารางที่ 4 ตารางแสดงรายละเอียดการส่งมอบงาน Testing 2

ฝ่าย	รายละเอียดการดำเนินงาน
Mechanical Design	แก้ไขและปรับปรุงโครงสร้างหุ่นยนต์เพื่อให้สามารถทำงานได้ตามเป้าหมาย
Electrical Module	แก้ไขและพัฒนาวงจรเพื่อให้หุ่นยนต์สามารถทำงานได้ตามเป้าหมายและสามารถสื่อสารรับ-ส่งข้อมูลกับ Base System ได้อย่างเสถียร
Control Engineering	แก้ไขและพัฒนาให้ระบบ PID Controller และ Trajectory สามารถควบคุมหุ่นยนต์ให้ทำงานได้ตามเป้าหมาย และจัดทำ Action Plan สำหรับการเคลื่อนที่ของแขนกล
Document & Presentation	<p>จัดทำเอกสารและสื่อการนำเสนอ สำหรับการนำเสนอ Testing 2 ซึ่งประกอบด้วย</p> <p>1. Mechanical Design</p> <p>1.1 รายละเอียดที่แก้ไขและปรับปรุงในฝ่าย Mechanical Design</p> <p>1.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในฝ่าย Mechanical Design ที่ได้รับการแก้ไขและปรับปรุง</p> <p>2. Electrical Module</p> <p>2.1 รายละเอียดที่แก้ไขและปรับปรุงในฝ่าย Electrical Module</p> <p>2.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในฝ่าย Electrical Module ที่ได้รับการแก้ไขและปรับปรุง</p> <p>3. Control Engineering</p> <p>3.1 รายละเอียดที่แก้ไขและปรับปรุงในฝ่าย Control Engineering</p> <p>3.2 ข้อมูลการทดสอบการใช้งานของหุ่นยนต์ในฝ่าย Control Engineering ที่ได้รับการแก้ไขและปรับปรุง</p>

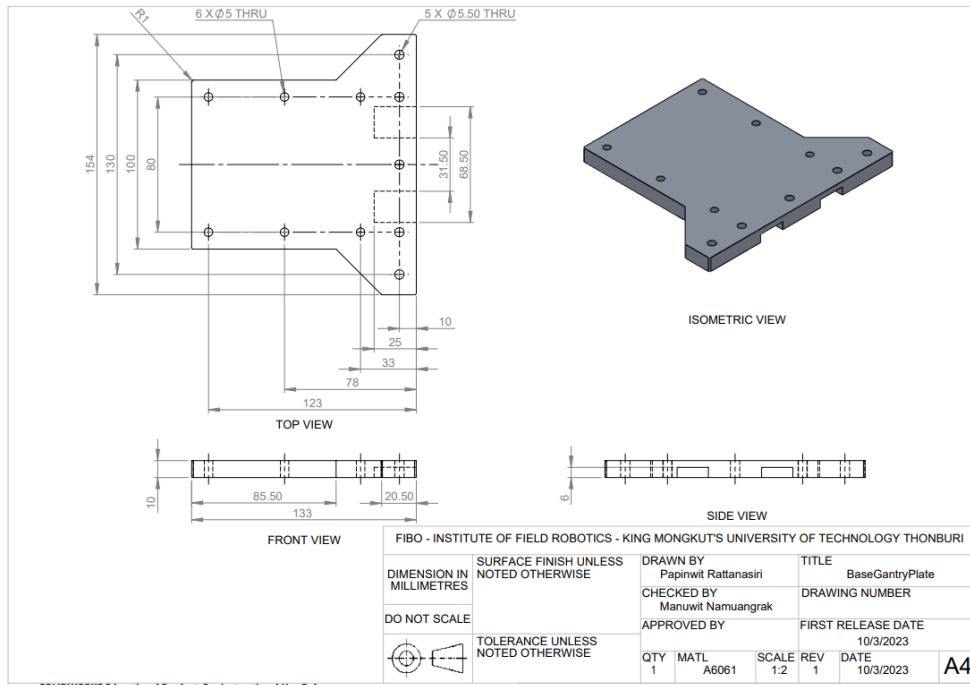
## Cost Section

งบประมาณที่ใช้ในการสร้างหุ่นยนต์โดยเป็นระบบแข็งกล มีรายละเอียด ดังตารางที่ 5  
ตารางที่ 5 ตารางแสดงงบประมาณในการสร้างหุ่นยนต์

ลำดับ	รายละเอียด	ราคาต่อหน่วย	จำนวน	ราคารวม
<b>Mechanic Design</b>				
1	Pulley GT2 48T Bore 8	100	1	100
2	Flanged Idlers GT2 20T Bore 5	130	2	260
3	Idler Pulley Wheel Bore	40	2	80
4	Bearing 608ZZ	10	2	20
5	clamp belt GT2 2mm 9.2mm 40mm	25	1	25
6	Belt 2GT PU Timing Belt Steel Core (1M)	40	2	80
7	6x8 Rigid Top Wire Coupling	20	1	20
8	Aluminum Profile 2020 (1M)	150	1	150
9	Aluminum Profile 2040 (1M)	230	1	230
10	3D Filament	470	1	470
11	ค่า CNC	800	10	8000
<b>Electronic Module</b>				
12	มอเตอร์ (DC motor)	2000	1	2000
13	โมดูลขับมอเตอร์ (Motor Driver)	1000	1	1000
14	เอ็นโคเดอร์ (Encoder)	1500	1	1500
15	เซนเซอร์วัดกระแสไฟฟ้า (Current sensor)	700	1	700
16	Photoelectric sensor	300	1	300
17	ฟิวส์ (Fuse)	100	3	300
18	แหล่งจ่ายไฟฟ้ากระแสสลับ Switching Power Supply	600	1	600
19	โมดูลลดระดับแรงดัน (Step-down regulator)	300	1	300

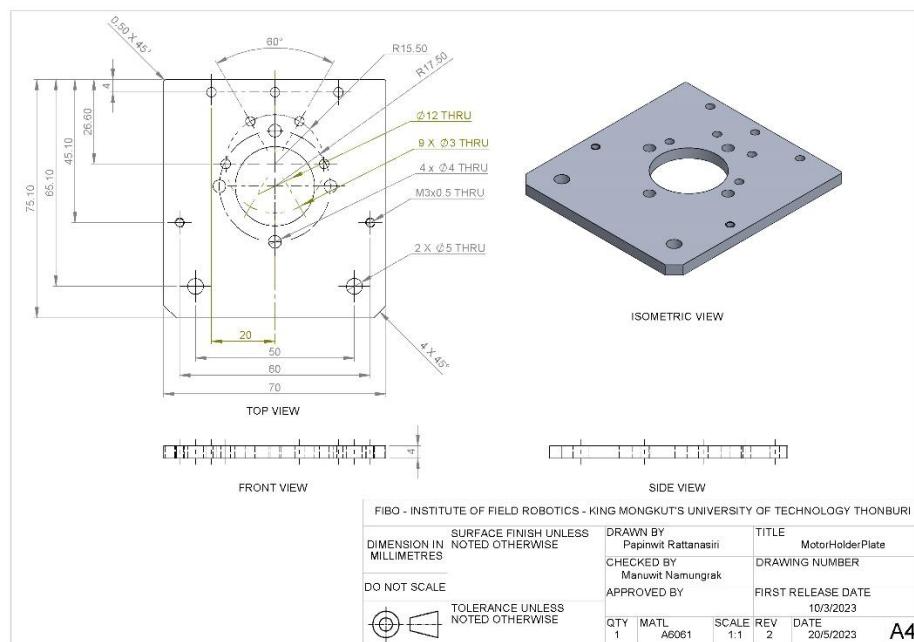
ลำดับ	รายละเอียด	ราคาต่อหน่วย	จำนวน	ราคารวม
Electronic Module				
20	ปุ่มควบคุมฉุกเฉิน (Emergency switch)	300	1	300
21	ปุ่มสวิตซ์เปิดและสวิตซ์ปิด (ON/OFF switch)	150	1	150
23	Joystick	115	1	115
22	โมดูลรีเลย์ (Relay module)	300	1	300
24	Pilot lamp	80	5	400
25	สายไฟวงจรกำลัง IEC02 THWF	500	1	500
26	สายไฟวงจรควบคุม IEC02 VSF	500	1	500
27	ตู้ไฟ	600	1	600
28	PCB	1000	1	1000
รวมเป็นเงิน				20000

ภาคผนวก

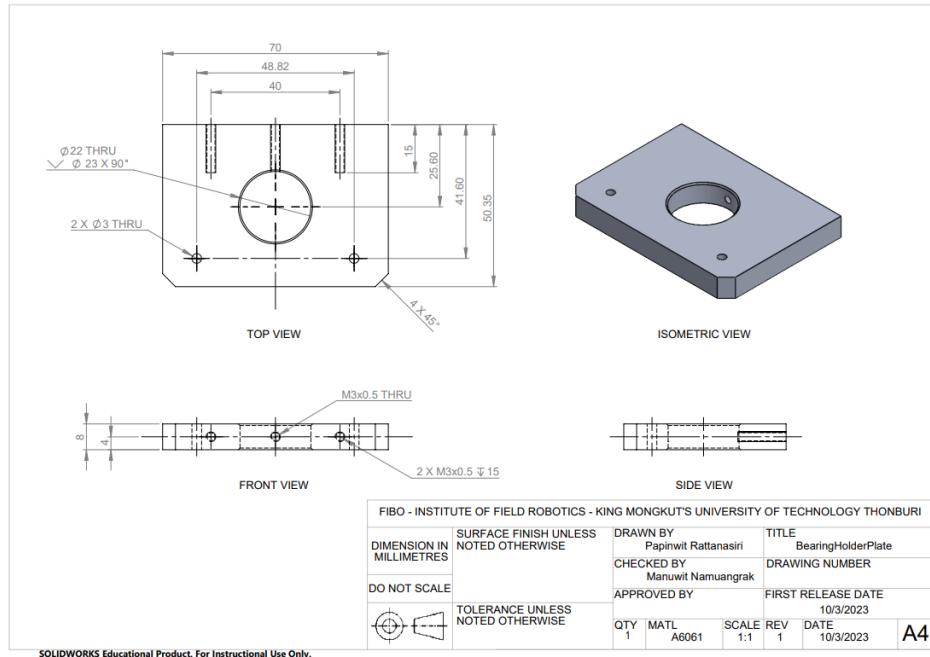


SOLIDWORKS Educational Product. For Instructional Use Only.

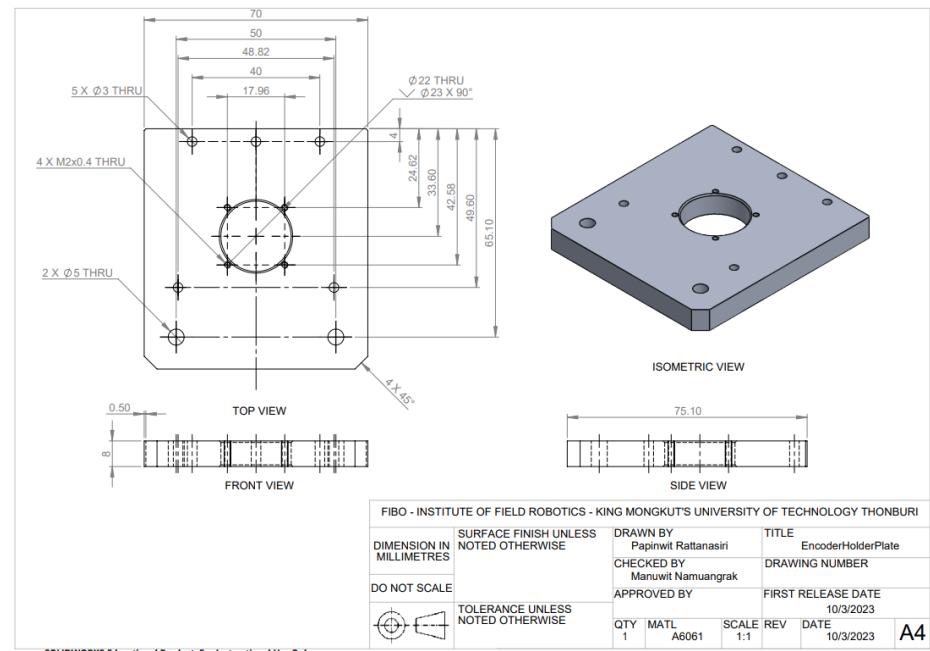
รูปที่ 1 รูปแสดง Drawing ของ BaseGantryPlate เป็นชิ้นส่วนที่ใช้ยึดตัวฐานของ FIBO กำหนดให้กับตัวหุ่นยนต์ของกลุ่มเรา เพื่อทำให้เป็นลายเป็นหินยนต์ตัวเดียว



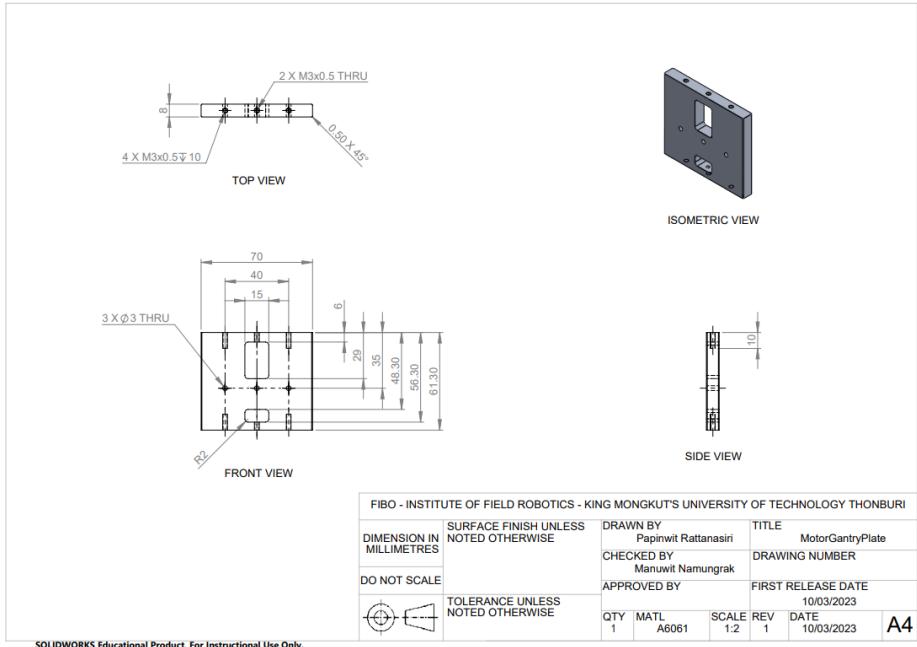
รูปที่ 2 รูปแสดง Drawing ของ MotorHolderPlate เป็นชิ้นส่วนที่ใช้ยึดตัวมอเตอร์



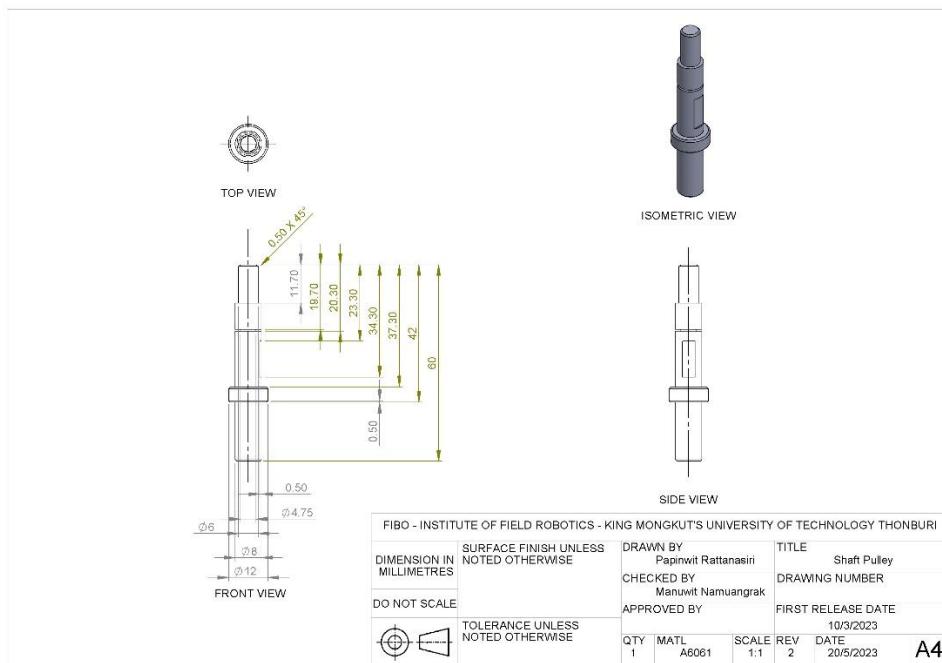
รูปที่ 3 รูปแสดง Drawing ของ BearingHolderPlate เป็นชิ้นส่วนที่ใส่ Bearing



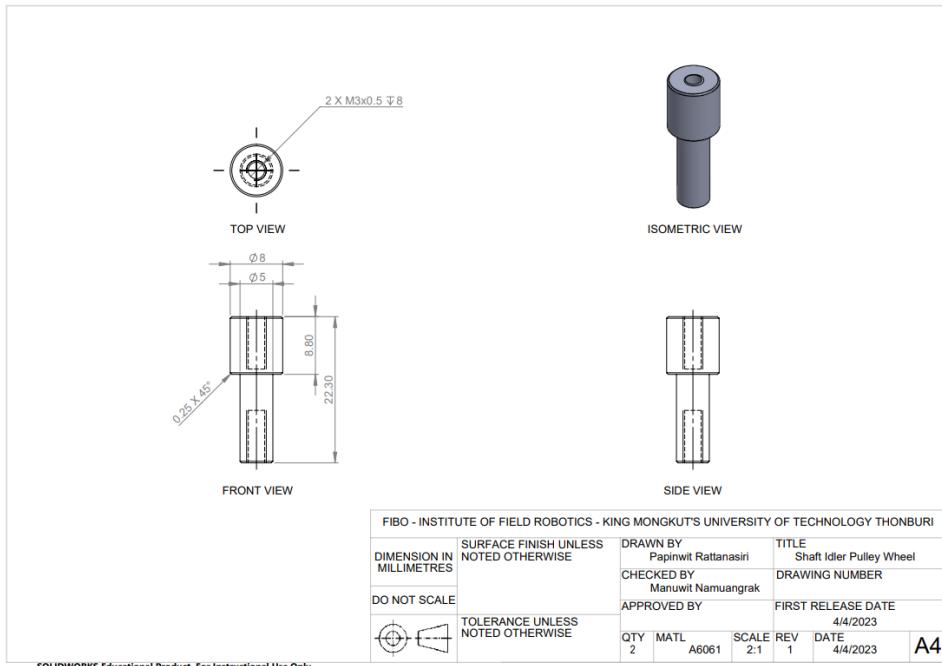
รูปที่ 4 รูปแสดง Drawing ของ EncoderHolderPlate เป็นชิ้นส่วนที่ใส่ Bearing และ Encoder  
เพื่อใช้อ่านตำแหน่งหรือความเร็วของมอเตอร์



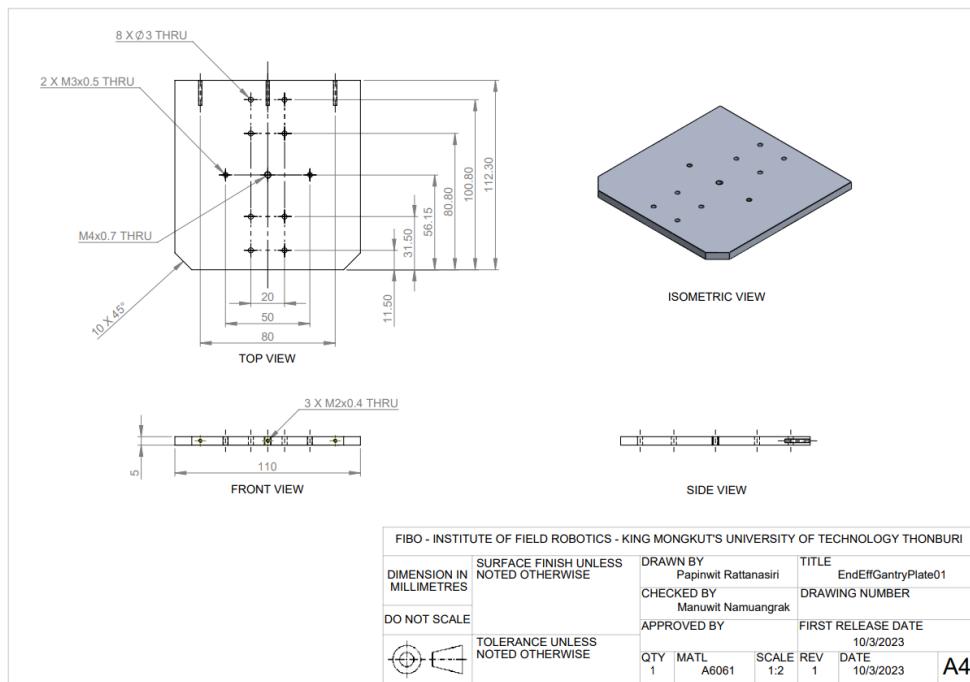
รูปที่ 5 รูปแสดง Drawing ของ MotorGantryPlate เป็นแกนกลางในการยึดทั้ง 3 ข้อก่อนหน้านี้ให้เป็นชิ้นเดียวกัน



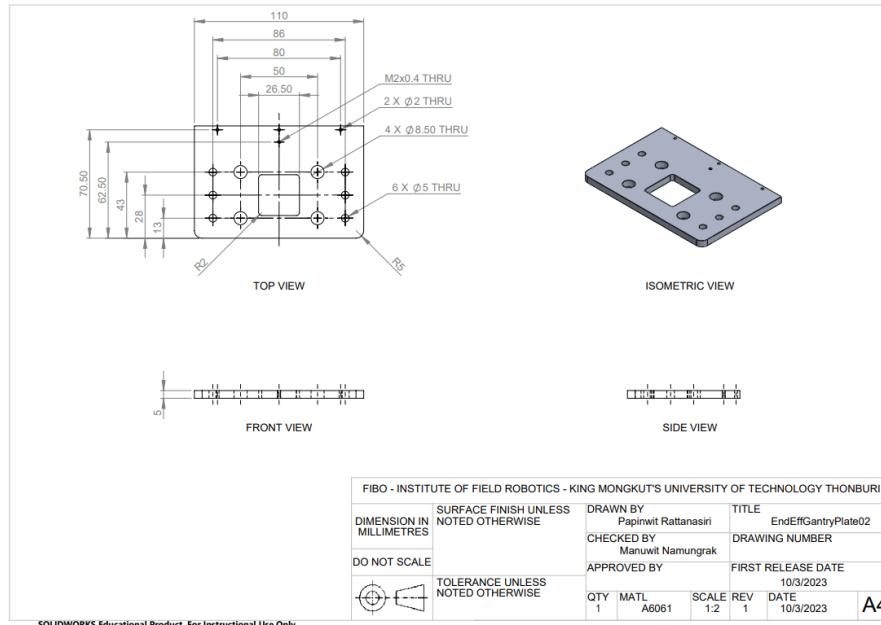
รูปที่ 6 รูปแสดง Drawing ของ Shaft Pulley เป็นเพลาแกนหลักที่ใช้ในการควบคุมตัวหมุนต์



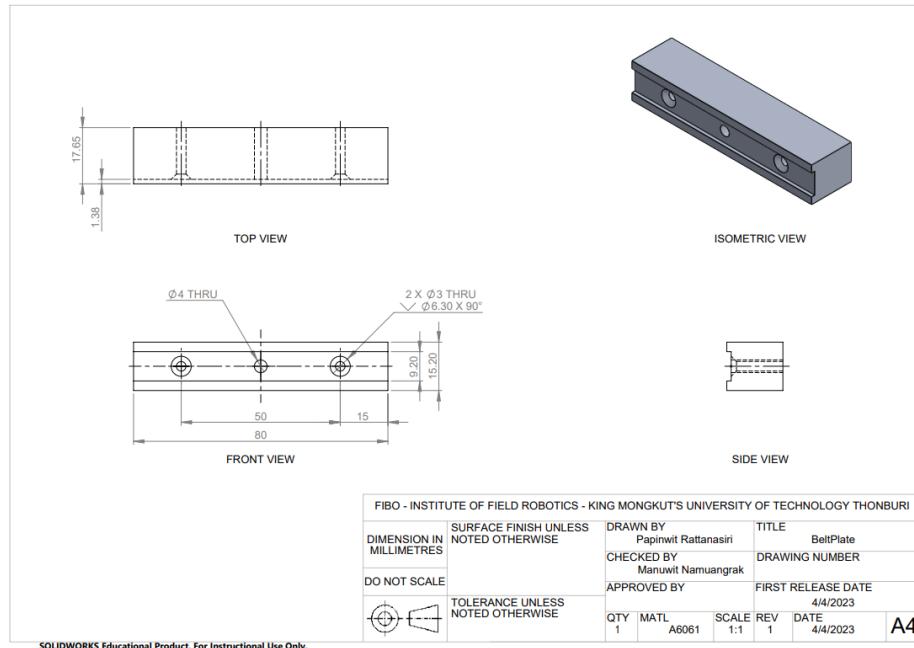
รูปที่ 7 รูปแสดง Drawing ของ Shaft Idler Pulley Wheel เป็นเพลาที่ว่าไถ Idler Pulley Wheel เพื่อเปลี่ยนตำแหน่งของ Belt



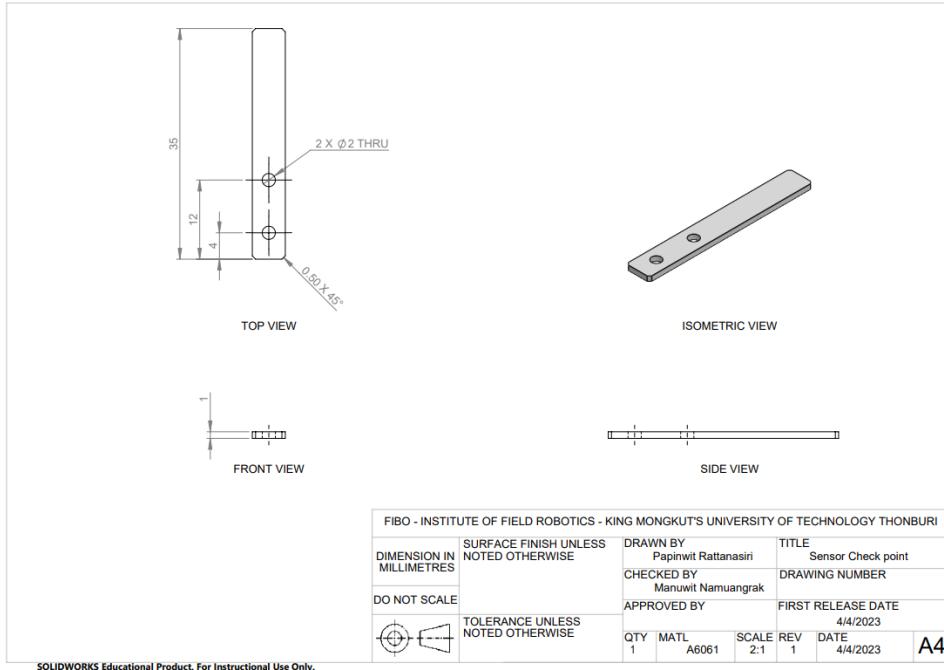
รูปที่ 8 รูปแสดง Drawing ของ EndEffGantryPlate01 เป็นชิ้นส่วนกล่างที่ใช้ยึด BeltPlate กับ Liner guide



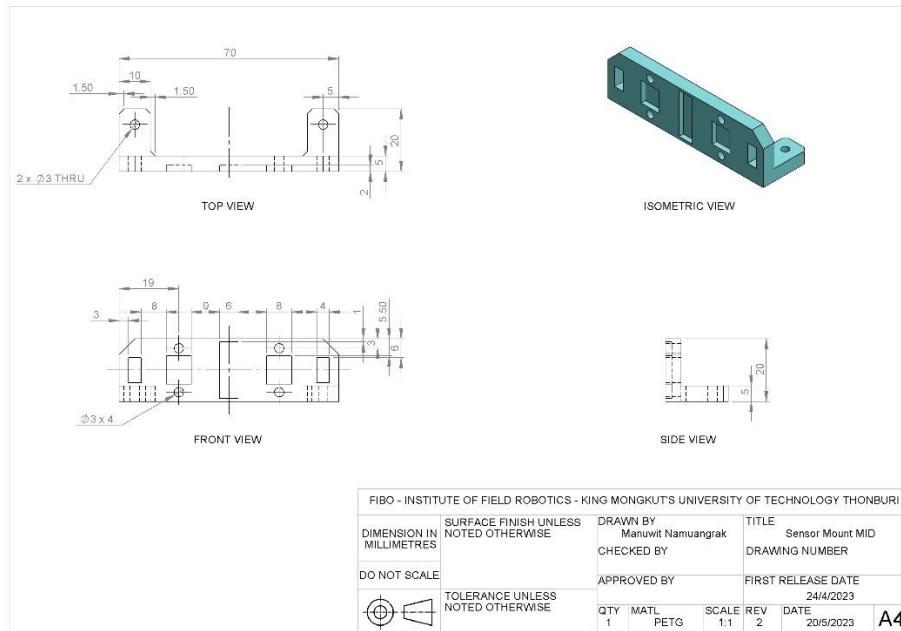
รูปที่ 9 รูปแสดง Drawing ของ EndEffGantryPlate02 เป็นชิ้นส่วนที่ใช้ด้วย FRA262-PAP-ENE-000-REV001  
เข้ากับชิ้นงานแล้วนำไปต่อ กับ EndEffGantryPlate01



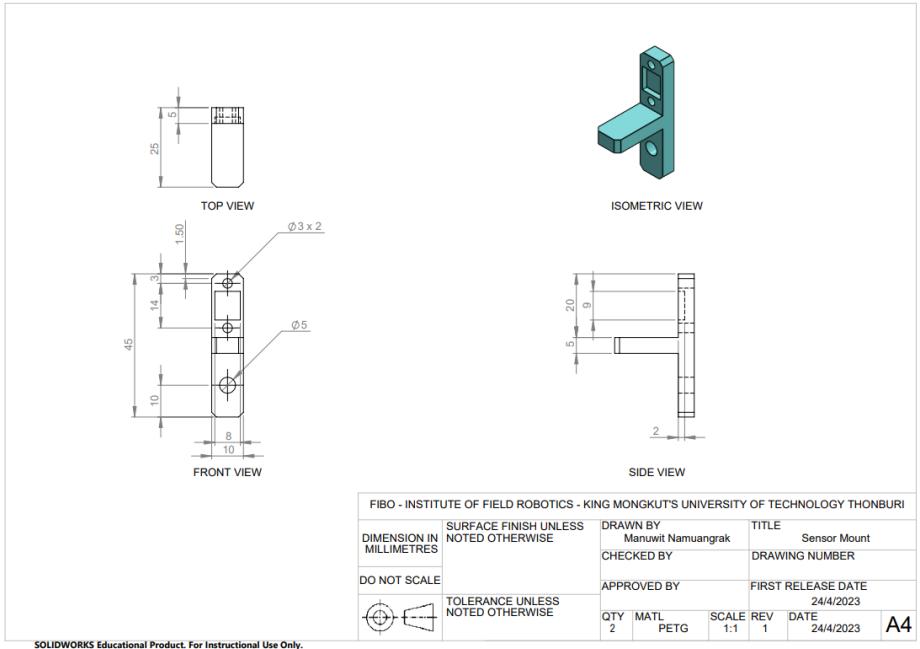
รูปที่ 10 รูปแสดง Drawing ของ BeltPlate เป็นชิ้นส่วนที่ช่วยในการล็อคตัว Belt เข้ากับ EndEffGantryPlate  
เพื่อที่จะทำให้ FRA262-PAP-ENE-000-REV001 สามารถเคลื่อนที่ได้



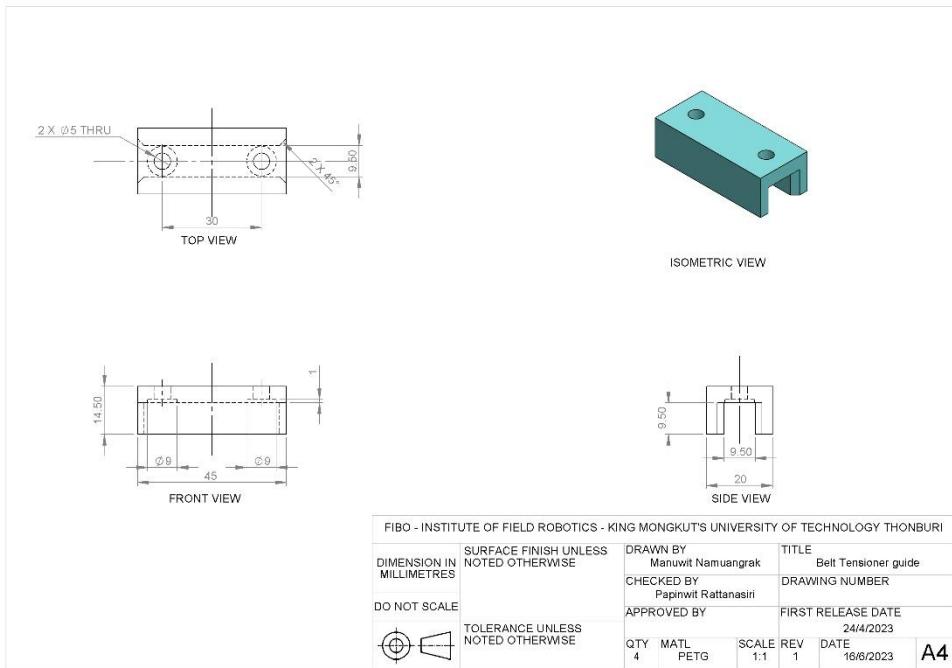
รูปที่ 11 รูปแสดง Drawing ของ Sensor Check point เป็นชิ้นส่วนที่ยื่นออกมาเพื่อให้ Sensor ตรวจจับว่าถึงตำแหน่งที่กำหนดแล้ว



รูปที่ 12 รูปแสดง Drawing ของ Sensor Mount MID เป็นชิ้นส่วนที่ไว้ใส่ Sensor ตรงกลางหุ่นเพื่อเป็นจุด Set zero

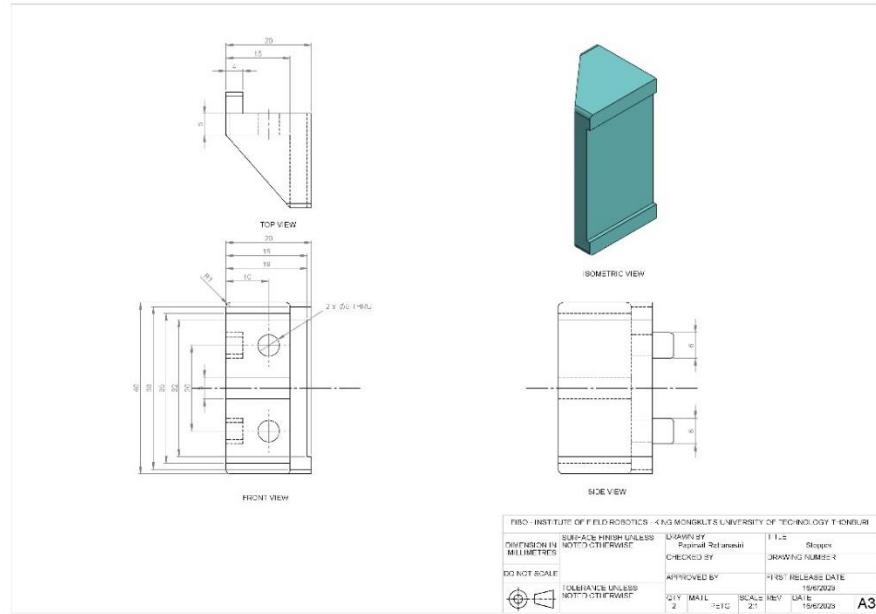


รูปที่ 13 รูปแสดง Drawing ของ Sensor Mount เป็นชิ้นส่วนที่ไว้ใส่ Sensor ด้านข้าง  
เพื่อกำหนด limit สูงสุดที่หุ่นเคลื่อนที่ได้

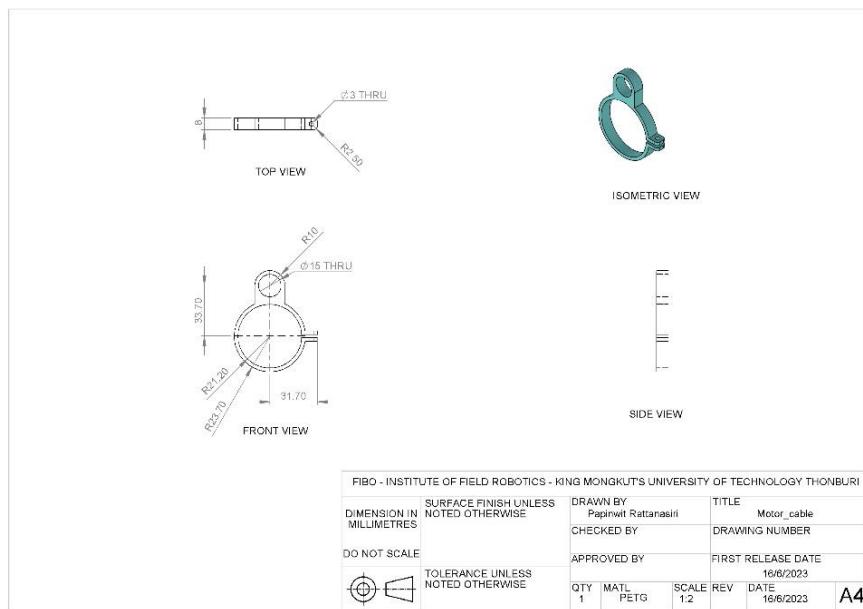


รูปที่ 14 รูปแสดง Drawing ของ Belt Tensioner guide เป็นชิ้นส่วนที่ใช้ guide ตัว Belt Tensioner

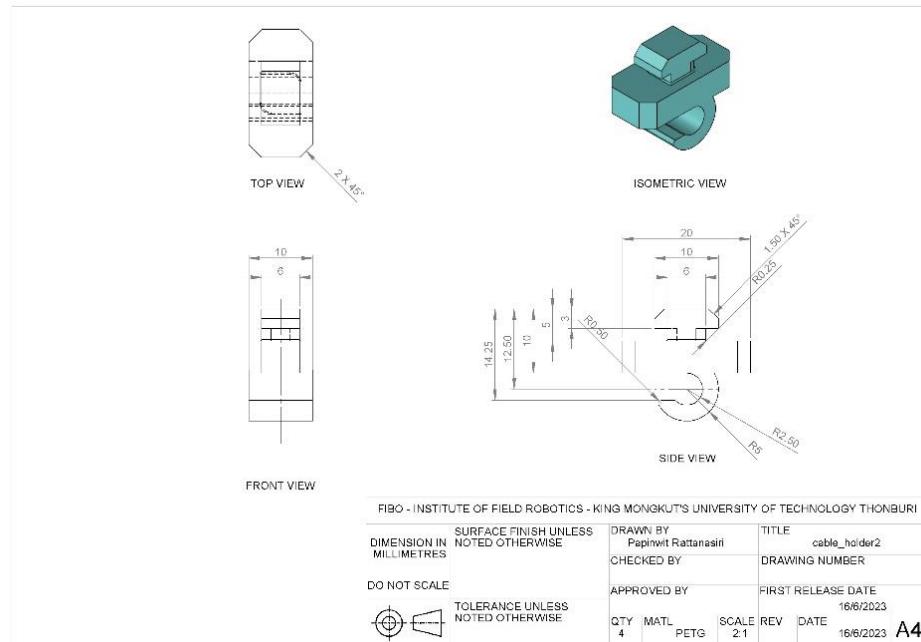
เพื่อให้ Belt Tensioner ไม่ส่าย และเคลื่อนที่เป็น Linear



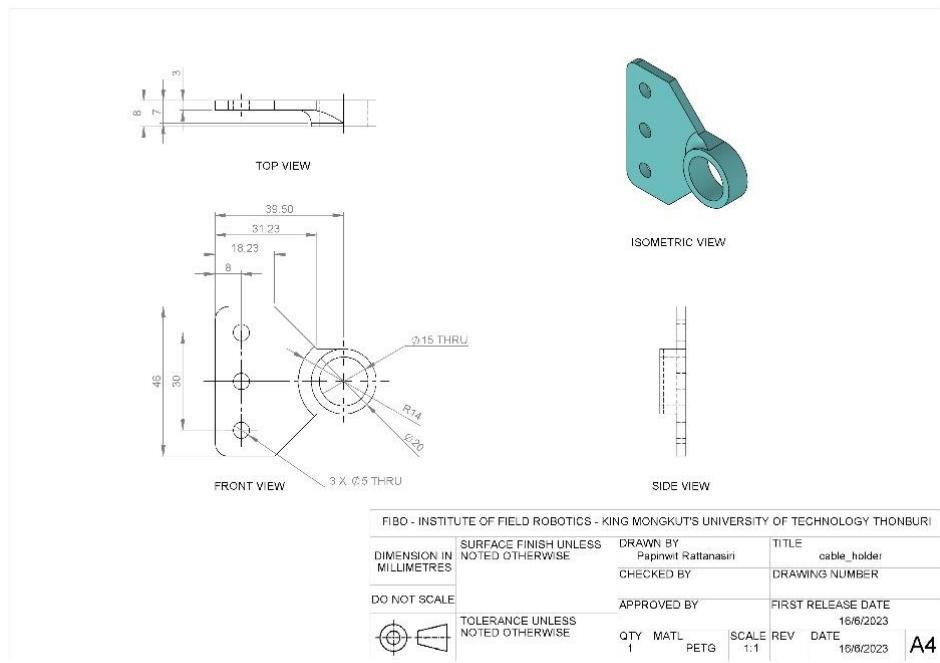
รูปที่ 15 รูปแสดง Drawing ของ Stopper เป็นชิ้นส่วนที่ใช้ในการติดยาง Stopper เพื่อยุดการเคลื่อนที่



รูปที่ 16 รูปแสดง Drawing ของ Motor\_cable เป็นชิ้นส่วนที่ใช้ในการเก็บสายไฟตรงมอเตอร์



รูปที่ 17 รูปแสดง Drawing ของ cable\_holder2 เป็นชิ้นส่วนที่ใช้ในการเก็บสายไฟตามแนว Aluminum Profile



รูปที่ 18 รูปแสดง Drawing ของ cable\_holder เป็นชิ้นส่วนที่ใช้ในการเก็บสายไฟที่ End effector