

**Structure Design and Platform Development of Universal Template for**   
**Humanoid Algorithm Interface (UTHAI)**

**การออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์ฮิวมานอยด์**

**เพื่อการศึกษาและวิจัย**

**นายจิรัฏฐ์ ศรีรัตนอาภรณ์**

**นายเจษฎากร ทาไชยวงค์**

**นายวุฒิภัทร โชคอนันตทรัพย์**

**โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร ปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมหุ่นยนต์ และระบบอัตโนมัติ สถาบันวิทยาการหุ่นยนต์ภาคสนาม  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
ปีการศึกษา 2560**

สารบัญ

[บทที่ 1 บทนำ 4](#_Toc499871955)

[1.1 ที่มาและความสำคัญ 4](#_Toc499871956)

[1.2 วัตถุประสงค์ 5](#_Toc499871957)

[1.3 ขอบเขตการดำเนินงาน 6](#_Toc499871958)

[1.4 ประโยชน์ที่คาดว่าจะได้รับ 6](#_Toc499871959)

[บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 7](#_Toc499871960)

[2.1 หุ่นยนต์ฮิวมานอยด์ 7](#_Toc499871961)

[2.2 ส่วนประกอบของหุ่นยนต์ฮิวมานอยด์ 8](#_Toc499871962)

[2.3 การวิเคราะห์องศาอิสระของมนุษย์ [1] 8](#_Toc499871963)

[2.4 การวิเคราะห์กลไกการเดินของมนุษย์ 9](#_Toc499871964)

[2.5 การออกแบบกลไกการเดินของหุ่นยนต์ฮิวมานอยด์ 10](#_Toc499871965)

[2.6 จลนศาสตร์ของหุ่นยนต์ 11](#_Toc499871966)

[2.7 การคำนวณพลังงานรวมในการใช้ของหุ่นยนต์ฮิวมานอยด์ 14](#_Toc499871967)

[2.8 เซนเซอร์ที่ใช้ในหุ่นยนต์ฮิวมานอยด์ 14](#_Toc499871968)

[2.8.1 Ground Contact Force Sensor 14](#_Toc499871969)

[2.8.2 Inertial Measurement Units (IMU) 15](#_Toc499871970)

[2.9 เครื่องมือที่ใช้พัฒนาระบบฮิวมานอยด์ 16](#_Toc499871971)

[2.9.1 Robot Middleware 16](#_Toc499871972)

[2.9.2 การจำลอง 18](#_Toc499871973)

[2.10 วิพากษ์วิจารณ์หุ่นยนต์ฮิวมานอยด์ 24](#_Toc499871974)

[2.10.1 Open source platform 24](#_Toc499871975)

[2.10.2 Non-Open source platform 29](#_Toc499871976)

[บทที่ 3 วิธีการดำเนินงาน 32](#_Toc499871977)

[3.1 หน้าที่ความรับผิดชอบ 32](#_Toc499871978)

[3.2 ขั้นตอนการดำเนินงาน 32](#_Toc499871979)

[3.2.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง 34](#_Toc499871980)

[3.2.2 ออกแบบสถาปัตยกรรมของหุ่นยนต์ 34](#_Toc499871981)

[3.2.3 ออกแบบระบบโครงสร้างทางกลของหุ่นยนต์ 37](#_Toc499871982)

[3.2.4 นำแบบจำลองของหุ่นยนต์ฮิวมานอยด์เข้าโปรแกรมจำลองการเคลื่อนไหว 42](#_Toc499871983)

[3.2.5 จัดทำชิ้นส่วนโครงสร้างและประกอบ 43](#_Toc499871984)

[3.2.6 พัฒนาซอฟแวร์ส่วนที่ติดต่อกับเซนเซอร์และตัวขับเคลื่อน 43](#_Toc499871985)

[3.2.7 จัดทำคู่มือและเอกสารการใช้งาน 43](#_Toc499871986)

[บทที่ 4 ผลการดำเนินงาน 44](#_Toc499871987)

[4.1 รายการอุปกรณ์ที่ได้จัดซื้อ 44](#_Toc499871988)

[4.2 เซนเซอร์ตรวจจับการสัมผัสพื้น 45](#_Toc499871989)

[4.3 การออกแบบโครงสร้างส่วนขา 47](#_Toc499871990)

[เอกสารอ้างอิง 49](#_Toc499871991)

# บทที่ 1 บทนำ

## 1.1 ที่มาและความสำคัญ

หุ่นยนต์ฮิวมานอยด์เป็นหุ่นยนต์ที่สร้างขึ้นเพื่อเลียนแบบสรีระของมนุษย์ ซึ่งมีแขนและขาที่มีข้อต่อจำนวนมาก มีลักษณะที่เป็นจุดเด่นของหุ่นยนต์ฮิวมานอยด์คือการเคลื่อนที่ด้วยขาทั้งสองขา ด้วยการเคลื่อนที่โดยใช้ขา ทำให้สามารถเคลื่อนที่ได้อย่างคล่องแคล่วในทุกสภาพพื้นผิว ทางเรียบ ทางขรุขระและพื้นต่างระดับ[[1]](#footnote-2)ในขณะที่หุ่นยนต์เคลื่อนที่แบบล้อไม่สามารถทำได้ และด้วยโครงสร้างที่คล้ายมนุษย์นั้นเอง จึงทำให้หุ่นยนต์ฮิวมานอยด์สามารถทำงานได้หลากหลายและยืดหยุ่น สามารถใช้อุปกรณ์ที่ถูกออกแบบขึ้นมาเพื่อมนุษย์ได้ นั่นหมายความว่าในอนาคตนั้นหุ่นยนต์ฮิวมานอยด์จะสามารถทำงานทดแทนงานของมนุษย์ได้[[2]](#footnote-3) งานที่หุ่นยนต์ฮิวมานอยด์จะมาทำทดแทนมนุษย์จะเป็นงานที่ต้องทำซ้ำ ๆจนเกิดความเมื่อยล้า งานที่อยู่พื้นที่ที่เสี่ยงต่อการเกิดความอันตราย เช่น ภายในโรงไฟฟ้านิวเคลียร์ การทหาร การดับเพลิง รักษาดินแดน หรือพื้นที่ที่ยากต่อการเข้าถึง

สถาบันวิจัยหลายแห่งทั่วโลกกำลังให้ความสนับสนุนด้านการศึกษาวิจัยและพัฒนาหุ่นยนต์ฮิวมานอยด์ เพื่อวัตถุประสงค์ต่าง ๆ ยกตัวอย่างเช่น DARPA Robotics Challenge (DRC)[[3]](#footnote-4) เป็นงานแข่งขันหุ่นยนต์กึ่งอัตโนมัติเพื่อทำภารกิจกู้ภัยในสถานการณ์ภัยพิบัติที่อันตราย และที่มีหลายสถาบันทั่วโลกได้ส่งหุ่นยนต์ฮิวมานอยด์ของตัวเองเข้ามาร่วมแข่งขัน ปัจจุบันมีหุ่นยนต์ฮิวมานอยด์ที่ถูกพัฒนาขึ้นหลากหลายตัวเช่น ASIMO[[4]](#footnote-5), HRP-3[[5]](#footnote-6), LOLA[[6]](#footnote-7) และ WATHLETE-1[[7]](#footnote-8) การพัฒนาหุ่นยนต์ฮิวมานอยด์นั้นก่อให้เกิดงานศึกษาวิจัย และทฤษฎีต่อยอดต่าง ๆมากมาย ไม่ว่าจะเป็น motion planning, การเดินแบบสถิต (Static walking), การเดินแบบพลวัต (Dynamic walking), การติดต่อสื่อสาร (Communication), การมองเห็นและการประมวลผลภาพ หรือปัญญาประดิษฐ์ (artificial intelligence: AI) ซึ่งงานวิจัยเหล่านี้สามารถที่จะประยุกต์ให้ใช้กับหุ่นยนต์ฮิวมานอยด์ได้

แม้ว่าจะมีการพัฒนาหุ่นยนต์ฮิวมานอยด์มามากมายแล้ว แต่การเริ่มต้นทำงานวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์ฮิวมานอยด์นั้น ต้องใช้ความรู้ความสามารถ เครื่องมือ ระยะเวลา งบประมาณ และความพยายยามเป็นอย่างมาก ในการสร้างหุ่นยนต์ฮิวมานอยด์ขึ้นมาใหม่นั้นต้องใช้งบประมาณสูง ดังนั้นการสร้างหุ่นยนต์ฮิวมานอยด์และระบบจำลองของหุ่นยนต์ขึ้นมาเป็น ระบบพื้นฐานที่พร้อมสำหรับการพัฒนาต่อยอดให้แก่นักศึกษาหรือนักวิจัย จะสามารถช่วยประหยัดเวลาและงบประมาณที่ต้องใช้ในการพัฒนา ซึ่งนั่นหมายความว่าจะทำให้นักศึกษาหรือนักวิจัยสามารถทำงานวิจัยได้อย่างมีประสิทธิภาพมากขึ้น รวบรวมเป็นหมวดหมู่ เกิดเป็นหุ่นยนต์ฮิวมานอยด์สัญชาติไทยที่สามารถทำให้ฉลาดขึ้นได้

ในงานวิจัยนี้เป็นการออกแบบและพัฒนาระบบพื้นฐานของหุ่นยนต์ฮิวมานอยด์สำหรับนักศึกษาและนักวิจัย โดยหุ่นยนต์ฮิวมานอยด์ที่ออกแบบและพัฒนาขึ้นมาเป็นพื้นฐานเพื่อการพัฒนาต่อยอดนั้น โครงสร้างระบบทางกลของหุ่นยนต์จำเป็นที่จะต้องมีความเสถียร สามารถปรับปรุงได้ง่าย มีการติดตั้งอุปกรณ์ เซนเซอร์สำหรับการทำงานวิจัยที่หลากหลาย ไม่ว่าจะเป็น กล้อง ไมโครโฟน ลำโพง เซนเซอร์วัดแรงที่ฝ่าเท้าเซนเซอร์วัดมุมเอียง เซนเซอร์วัดแรงบิดที่กระทำต่อข้อต่อ เซนเซอร์วัดอุณหภูมิ หรือ เซนเซอร์วัดกระแสของตัวขับเคลื่อน มีการจัดทำคู่มือหรือเอกสารวิธีใช้งานอย่างชัดเจน และสื่อการสอนเกี่ยวกับเซนเซอร์ที่ใช้ในหุ่นยนต์ฮิวมานอยด์ เพื่อที่จะทำให้ง่ายต่อการศึกษาทำความเข้าใจ ปรับปรุง บำรุงรักษา พัฒนาต่อยอด ผู้วิจัยมีความคาดหวังว่าผู้ที่จะใช้ระบบพื้นฐานนี้ เป็นผู้ที่มีความรู้ในการพัฒนาโปรแกรมคอมพิวเตอร์ด้วยภาษา Python, C++ หรือ MATLAB ความรู้ในการติดต่อสื่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์หรือเซนเซอร์ต่างๆเบื้องต้นได้ รู้จักทฤษฎีระบบควบคุมพื้นฐาน และการใช้งาน Robot Operating System (ROS) ในการ Publish Subscribe และ Service ได้

## 1.2 วัตถุประสงค์

ออกแบบโครงสร้างและพัฒนาระบบพื้นฐานสำหรับหุ่นยนต์ฮิวมานอยด์เพื่อการวิจัยและพัฒนาต่อยอด และเปิดให้บุคคลทั่วไปสามารถเข้าถึงได้

## 1.3 ขอบเขตการดำเนินงาน

1. ใช้ ROS เป็นกรอบการทำงานสำหรับพัฒนาระบบพื้นฐาน
2. ออกแบบหุ่นยนต์ให้มีโครงสร้างที่มีความแข็งแรง สามารถรับน้ำหนักอุปกรณ์ต่าง ๆที่ติดตั้งอยู่บนตัวหุ่นยนต์ได้ไม่เกิน 5 กิโลกรัม โดยใช้โปรแกรมออกแบบสามมิติ (3D Solidworks)
3. หุ่นยนต์มีความสูงไม่ต่ำกว่า 100 เซนติเมตร และสูงไม่เกิน 120 เซนติเมตร
4. หุ่นยนต์มี 2 แขน 2 ขา โดยมีองศาอิสระของขาข้างละ 6 และแขนข้างละ 2 องศาอิสระ
5. หุ่นยนต์สามารถทำงานภายในสภาพแวดล้อมแบบปิด
6. หุ่นยนต์ใช้พลังงานจากแหล่งจ่ายพลังงานที่มีขนาดแรงดันไฟฟ้า 12 โวลต์
7. ใช้ Dynamixel Digital Servo เป็นตัวขับเคลื่อนสำหรับแต่ละข้อต่อ
8. นำ CAD Model เข้าโปรแกรม Gazebo เพื่อใช้ทำระบบจำลองสำหรับทดลองระบบ
9. ติดตั้ง Ground contact sensor สำหรับตรวจฝ่าเท้าของหุ่นยนต์
10. ติดตั้ง IMU สำหรับการรักษาสมดุลในการเดินของหุ่นยนต์
11. จัดทำคู่มือ เอกสารวิธีการใช้งาน การติดตั้งระบบ และ รายละเอียดส่วนประกอบส่วนต่าง ๆ

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. มีต้นแบบหุ่นยนต์ฮิวมานอยด์สำหรับใช้ในงานวิจัยในแขนงต่าง ๆ
2. มีระบบพื้นฐานสำหรับพัฒนาหุ่นยนต์ฮิวมานอยด์เป็นของสถาบัน
3. มีคู่มือ เอกสารวิธีใช้งาน และรายระเอียดของหุ่นยนต์สำหรับผู้พัฒนาต่อยอด

# **บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง**

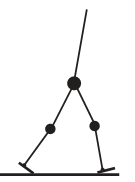
## 2.1 หุ่นยนต์ฮิวมานอยด์

หุ่นยนต์ฮิวมานอยด์ คือ หุ่นยนต์ที่ถูกสร้างขึ้นมาให้มีรูปร่างคล้ายคลึงกับโครงสร้างของมนุษย์ มักได้รับการออกแบบขึ้นมาเพื่อจุดประสงค์บางอย่าง เช่น เพื่อให้ใช้เครื่องมือของมนุษย์และให้อยู่ในสภาพแวดล้อมของมนุษย์ได้ เพื่อศึกษาการเคลื่อนไหวของร่างกาย ศึกษาการทำแผนที่ เพื่อให้ทำงานในสิ่งที่มนุษย์ทำได้ยาก หรือเพื่อจุดประสงค์อื่น ๆ โดยทั่วไปแล้ว หุ่นยนต์ฮิวมานอยด์จะประกอบไปด้วย ส่วนของลำตัว ส่วนของหัว ส่วนของแขน และส่วนของขา แต่การสร้างหุ่นยนต์ฮิวมานอยด์นั้นก็ไม่จำเป็นที่จะต้องมีส่วนประกอบทุกส่วนเหมือนมนุษย์ ในบางครั้งจะออกแบบอาจมีเพียงแค่บางส่วนของมนุษย์เท่านั้น เช่น หุ่นยนต์ที่มีแค่ส่วนข้างบนนับจากเอวขึ้นไป หุ่นยนต์ที่มีแค่ส่วนข้างล่างนับจากเอวลงมา หรือหุ่นยนต์ที่มีใบหน้าเหมือนมนุษย์ มีตา มีปาก พูด ปฏิสัมพันธ์กับมนุษย์ได้

หุ่นยนต์ฮิวมานอยด์ สามารถแบ่งองค์ประกอบที่สำคัญออกเป็น 3 ส่วนประกอบหลักดังนี้

1. ระบบทางกล คือ ส่วนของโครงสร้างทางกลของหุ่นยนต์ เช่น ก้านต่อ ข้อต่อ
2. ระบบไฟฟ้าหรืออิเล็กทรอนิกส์ คือ ส่วนของวงจรไฟฟ้าต่าง ๆ แหล่งพลังงาน วงจรควบคุม รวมไปถึงอุปกรณ์ที่ใช้ในการขับเคลื่อน หรือการรับรู้สถาพแวดล้อมต่าง ๆของหุ่นยนต์ฮิวมานอยด์ เช่นกล้อง เซนเซอร์วัดแรง ไมโครโฟน ลำโพง
3. ระบบซอฟแวร์ คือ ส่วนที่ใช้สำหรับประมวลผลการทำงานและสั่งการการเคลื่อนไหวท่าทางของหุ่นยนต์ฮิวมานอยด์

## 2.2 ส่วนประกอบของหุ่นยนต์ฮิวมานอยด์



ขา(Leg)

ลำตัว(Torso)

สะโพก(Hip)

รูปที่ 2.1 ส่วนประกอบของหุ่นยนต์ฮิวมานอยด์

หุ่นยนต์ฮิวมานอยด์ประกอบด้วยก้านต่อที่ต่อกัน ในลักษณะโครงสร้างแบบโซ่เปิด โดยมีก้านต่อย่อย 2 ส่วน ส่วนแรกคือ ส่วนก้านต่อของลำตัวหุ่นยนต์ (Torso) ซึ่งจะรวมไปถึงส่วนแขนกับหัวด้วย และในส่วนที่สองคือ ส่วนก้านต่อของขาหุ่นยนต์ (Legs) ซึ่งเป็นส่วนที่สัมผัสกับพื้น ทั้งสองก้านต่อจะเชื่อมต่อกันด้วยส่วนของสะโพก (Hip) ดังรูปที่ 1

## 2.3 การวิเคราะห์องศาอิสระของมนุษย์ [1]

การที่มนุษย์เราสามารถเคลื่อนที่ได้นั้น เป็นผลเนื่องมาจากการเคลื่อนที่ของข้อต่อต่าง ๆ ที่อยู่บนขา ซึ่งประกอบไปด้วย ข้อต่อส่วนสะโพก ข้อต่อส่วนหัวเข่า และข้อต่อส่วนข้อเท้า แรงบิดที่เกิดขึ้นของแต่ละข้อต่อมีความสัมพันธ์ต่อกัน ส่งผลให้เกิดเสถียรภาพในการเดินของมนุษย์ เมื่อวิเคราะห์ลักษณะโครงสร้างในแต่ละส่วน พบว่าข้อต่อส่วนสะโพกมีลักษณะเป็นทรงกลม ทำให้ข้อต่อส่วนสะโพกสามารถหมุนได้ 3 องศาอิสระ ส่วนหัวเข่าของมนุษย์ มีจุดต่อของข้อที่มีลักษณะเป็นทรงกลมสองลูกประกอบเข้าด้วยกันทำให้การเคลื่อนที่ถูกบังคับให้สามารถเคลื่อนที่ได้เพียง 1 องศาอิสระ ในส่วนของข้อเท้ามีลักษณะการเคลื่อนที่เหมือนสะโพกคือสามารถเคลื่อนที่ได้ 3 องศาอิสระ

จากทั้งหมดที่ได้ทำการวิเคราะห์มาข้างต้นพบว่าในขาหนึ่งข้างของมนุษย์ประกอบไปด้วย 7 องศาอิสระ ซึ่งส่งผลให้การเคลื่อนที่ของมนุษย์มีความคล่องแคล่วสูง แต่ในทางออกแบบกลไกการเดินและการควบคุมของหุ่นยนต์สองขาถือว่ามีจำนวนองศาอิสระเกินความจำเป็นในการเคลื่อนที่บนปริภูมิ(space) และยากต่อการควบคุม(under actuated) ดังนั้นการกำหนดจำนวนองศาอิสระเพื่อให้หุ่นยนต์เดินได้เสมือนมนุษย์จึงมีผลในการออกแบบกลไกทางกลและการควบคุมของหุ่นยนต์สองขา

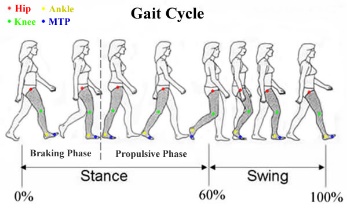
ตารางที่ 2.1 แสดงความสามารถในการหมุนของแต่ละข้อต่อของมนุษย์

|  |  |  |  |
| --- | --- | --- | --- |
| ข้อต่อ | องศาอิสระ | ความสามารถด้านการหมุน (องศา) | |
| สูงสุด | ต่ำสุด |
| หัว |  | +60 | -30 |
|  | +70 | -70 |
|  | +80 | -80 |
| หลัง |  | +30 | -60 |
|  | +55 | -55 |
|  | +45 | -45 |
| หัวไหล่ |  | +180 | -80 |
|  | +45 | -135 |
|  | +30 | 0 |
| ข้อศอก |  | 0 | -155 |
| สะโพก |  | +120 | -40 |
|  | +40 | -50 |
|  | +60 | -50 |
| หัวเข่า |  | 0 | -130 |
| ข้อเท้า |  | +30 | -60 |
|  | +45 | -20 |
|  | +20 | -60 |

## 2.4 การวิเคราะห์กลไกการเดินของมนุษย์

การเคลื่อนที่ของหุ่นยนต์ฮิวมานอยด์นั้นจะเลียนแบบการเคลื่อนที่ของมนุษย์โดยการก้าวเดินของขาและเท้าไปในทิศทางที่ต้องการให้หุ่นยนต์เคลื่อนที่ ขณะเดินหุ่นยนต์จะมีพื้นที่รองรับน้ำหนักทั้งตัวเพียงแค่บริเวณฝ่าเท้าของหุ่นยนต์เท่านั้น

การวิเคราะห์ลักษณะการเดินของมนุษย์ เป็นการศึกษาเพื่อทำความเข้าใจถึงธรรมชาติการเดินก่อนนำไปทำการออกแบบกลไกทางกลและระบบควบคุมหุ่นยนต์ฮิวมานอยด์ การก้าวเดินโดยปกติของมนุษย์มีลักษณะเป็นวัฏจักร วนซ้ำไปเรื่อย ๆในทิศทางที่ต้องการจนกว่าจะทำการหยุดเดิน การทรงตัวในระหว่างการยืนหรือการก้าวเดินนั้นเป็นไปตามสัญชาตญาณซึ่งเกิดจากการรักษาความสมดุลของระดับน้ำในหู[[8]](#footnote-9) ส่งสัญญาณผ่านเส้นประสาทไปยังกล้ามเนื้อส่วนต่าง ๆ ที่ทำหน้าที่ให้เกิดการเคลื่อนที่ การเคลื่อนที่ของมนุษย์ในการเดินไปข้างหน้าสามารถแบ่งออกเป็นช่วงต่าง ๆ ดังนี้



รูปที่ 2.2 วัฏจักรการเดินของมนุษย์

1. ช่วงเริ่มการวางเท้าเพื่อเข้าสู่ช่วงเริ่มต้นเหวี่ยงเท้า เป็นช่วงที่เท้าเกิดการกระแทกลงบนพื้นหลังจากทำการเหวี่ยงมาจากด้านหลัง โดยธรรมชาติมนุษย์จะทำการวางส้นเท้าลงเพื่อลดแรงกระแทกที่เกิดขึ้นในช่วงนี้ ดังนั้นทางกายภาพในส่วนของส้นเท้ามนุษย์จึงมีลักษณะอ่อนนุ่ม
2. ช่วงเริ่มต้นเหวี่ยงเท้าเพื่อเข้าสู่ช่วงเหวี่ยงเท้า หลังจากทำการวางส้นเท้าลงกับพื้นแล้ว ข้อเข้าจะปรับมุมเพื่อให้ฝ่าเท้าแนบพื้นสนิท ขณะเดียวกันขาอีกข้างจะยกสูงขึ้นเพื่อถ่ายเทน้ำหนักไปยังเท้าที่เพิ่งวางลง
3. ช่วงเหวี่ยงเท้า เป็นช่วงที่ขาหนึ่งยกลอยอยู่ในอากาศและขาที่วางแนบกับพื้นจะรองรับน้ำหนักทั้งหมดของร่างกาย
4. ช่วงเตรียมการวางเท้า เป็นช่วงที่ขาข้างที่ลอยอยู่เหวี่ยงไปข้างหน้าเพื่อเตรียมเข้าสู่ช่วงรองรับ   
   ในขณะเดียวกันขาที่รับน้ำหนักอยู่จะทำการผลักตัวเพื่อเริ่มทำการถ่ายเทน้ำหนักไปข้างหน้า

## 2.5 การออกแบบกลไกการเดินของหุ่นยนต์ฮิวมานอยด์

ในการออกแบบหุ่นยนต์ฮิวมานอยด์ให้สามารถเดินได้เสมือนมนุษย์ โดยใช้จำนวนองศาอิสระของข้อต่อให้มีเทียบเท่ามนุษย์นั้น พบว่า จะมีข้อจำกัดทางด้านการออกแบบ เนื่องมาจากอุปกรณ์ที่ใช้ในการขับเคลื่อนข้อต่อต่างๆ มีอยู่อย่างจำกัด รวมถึงข้อจำกัดทางด้านตัวรับรู้สถานะตัวขับของหุ่นยนต์ ดังนั้นผู้เขียนจึงได้ออกแบบหุ่นยนต์ให้มีองศาอิสระของข้อต่อของขาหนึ่งข้างเท่ากับ 6 องศาอิสระ โดยในส่วนนี้ได้ต่างจากมนุษย์คือ ตัดส่วนของการหมุนรอบน่องเท้าออกไป ทั้งนี้หุ่นยนต์ยังสามารถเคลื่อนที่ได้ในปริภูมิ เนื่องจากองศาอิสระของข้อต่อเพียงพอต่อการใช้งาน

จากการศึกษาการเคลื่อนที่ของมนุษย์และทฤษฏีทางด้านหุ่นยนต์แล้ว ผู้เขียนได้ข้อสรุปในการออกแบบขาหนึ่งข้างของหุ่นยนต์ให้มีองศาอิสระเท่ากับ 6 องศาอิสระ และได้ใช้ตัวขับเคลื่อนเป็นแบบดิจิตอล (Digital Servo) ของบริษัท Robotics เนื่องจากภายในมีตัวรับรู้สถานะของตัวขับเคลื่อนต่างๆ และตัวขับเคลื่อนนี้ถูกออกแบบมาให้สามารถติดตั้ง และสั่งการได้ง่าย

## 2.6 จลนศาสตร์ของหุ่นยนต์

จลนศาสตร์ เป็นการศึกษาเกี่ยวกับตำแหน่ง (position), ทิศทางการหมุน (orientation) และการเคลื่อนที่ ทั้งเชิงเส้น (translation) และเชิงมุม (rotation) โดยไม่ได้คำนึงถึงแรงที่ก่อให้เกิดการเคลื่อนที่ โดยพื้นฐานแล้วหุ่นยนต์เป็นระบบพลวัต (dynamic) ที่เกี่ยวข้องกับการเคลื่อนที่ ซึ่งต้องคำนวณถึงแรงที่ก่อให้เกิดการเคลื่อนที่ด้วย

หุ่นยนต์ฮิวมานอยด์ตรงส่วนขา เป็นหุ่นยนต์แบบอนุกรมประกอบขึ้นจากการต่อกันของก้านต่อต่าง ๆด้วยข้อต่อ ไล่เรียงลำดับจากส่วนสะโพกถึงส่วนปลายเท้า ในลักษณะโครงสร้างแบบโซ่เปิด ขาของหุ่นยนต์ฮิวมานอยด์จะสามารถเคลื่อนที่ได้ด้วยการขับเคลื่อนของตัวขับ ซึ่งมักติดตั้งอยู่ที่ข้อต่อ ซึ่งทำให้ท่าทางของขาถูกกำหนดได้ด้วยค่าตัวแปรของข้อต่อ หากต้องการที่จะรู้ว่าขาของหุ่นยนต์จะเคลื่อนที่อย่างไรในปริภูมิสามมิติ จำเป็นที่จะต้องรู้ความสัมพันธ์ของตำแหน่งและทิศทางการหมุนของขาหุ่นยนต์ และตัวแปรของข้อต่อ

ดังนั้นท่าทางของขาจะถูกกำหนดด้วยค่าตัวแปรของข้อต่อ เพื่อความสะดวกในการใช้งานจึงมักจะเขียนให้อยู่ในรูปของเวกเตอร์ เรียกว่าเวกเตอร์ของข้อต่อ (joint vector)

ปริภูมิของเวกเตอร์ของข้อต่อทั้งหมด จะเรียกว่าปริภูมิของข้อต่อ (joint space)

บางครั้งในทางปฏิบัติ อาจจะมีกลไกบางอย่างเพื่อเปลี่ยนแปลงการเคลื่อนที่จากตัวขับไปยังข้อต่อ เพื่อประโยชน์ต่าง ๆ เช่น ทดแรง ทดรอบ ลดมวล นั่นทำให้ ตัวแปรข้อต่อไม่ใช่ตัวแปรที่เกิดจากตัวขับโดยตรง จึงมีการเขียนค่าของตัวแปรตัวขับให้อยู่ในรูปของเวกเตอร์ เรียกว่าเวกเตอร์ของตัวขับ (actuator vector)

ปริภูมิของเวกเตอร์ของตัวขับทั้งหมด จะเรียกว่าเวกเตอร์ของตัวขับ (actuator space)

โดยทั่วไปแล้วการอธิบายส่วนปลายขาของหุ่นยนต์ซึ่งมีทั้งตำแหน่งและทิศทางการหมุน นิยมอธิบายด้วยเวกเตอร์ตำแหน่ง และมุมออยเลอร์ โดยเราสามารถเขียนให้อยู่ในรูปเวกเตอร์รวมได้

ปริภูมิของตำแหน่งและการหมุนนี้จะเรียกว่าปริภูมิของการทำงาน (task space)

จลนศาสตร์ของหุ่นยนต์จะแบ่งออกเป็น 2 ประเภท คือ จลนศาสตร์ไปข้างหน้า และจลนศาสตร์ผกผัน ซึ่งจะนำมาใช้ในการกำหนดมุมในแต่ละข้อต่อของหุ่นยนต์

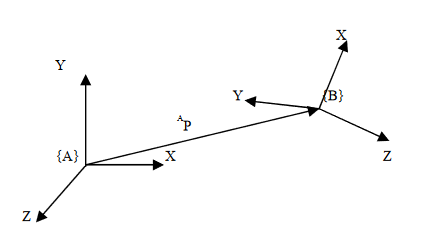
1. จลนศาสตร์ไปข้างหน้า (Forward Kinematics)

จลนศาสตร์ไปข้างหน้าเป็นการวิเคราะห์หาฟังก์ชันของตำแหน่งและทิศทางการหมุน (task space) ในพจน์ที่มีตัวแปรเป็น ค่าของข้อต่อ (joint space)

รูปที่ 2.3 แสดงความสัมพันธ์ระหว่างปริภูมิตัวขับ ปริภูมิข้อต่อ และปริภูมิการทำงาน โดยจลนศาสตร์ไปข้างหน้า

ในการคำนวณการหาสมการพื้นฐานของโครงสร้างของหุ่นยนต์นั้น เราจะต้องกำหนดแกนสมมุติขึ้นมา (x,y,z) ซึ่งเราจะได้ตัวแปรที่ไม่ทราบค่าของแต่ละข้อต่อ เพื่อหาตำแหน่งปลายสุดของขาหุ่นยนต์ จะบ่งบอกถึงตำแหน่ง (Position vector)

และการหมุนของแกนข้อต่อ เมื่อเปรียบเทียบกับแกนข้อต่อที่ผ่านมา (Rotation matrix)



รูปที่ 2.4 แสดงความสัมพันธ์ระหว่างเวกเตอร์ กับแกน

เมื่อมีการหมุน หรือเลื่อนตำแหน่งของข้อต่อจะมีการเปลี่ยนรูปของแกนใหม่ ดังนั้นเราสามารถเขียนให้อยู่ในรูปของการเปลี่ยนรูปแบบ (Transformation matrix) ได้โดยการ กำหนด (Rotation matrix) ให้อยู่ในรูปแบบ 4x4 และ (Position vector) ให้อยู่ในรูปแบบ 4x1 หรือก็คือ Homogenuos Tranformation Matrix

1. จลนศาสตร์ผกผัน (Inverse Kinematics)

จลนศาสตร์ผกผันเป็นการวิเคราะห์หาฟังก์ชันของค่าของข้อต่อ (joint space) ในพจน์ที่มีตัวแปรเป็นค่าของตำแหน่งและทิศทางการหมุน (task space)



รูปที่ 2.5 แสดงความสัมพันธ์ระหว่างปริภูมิตัวขับ ปริภูมิข้อต่อ และปริภูมิการทำงาน โดยจลนศาสตร์ผกผัน

วิธีการคำนวณจลนศาสตร์ผกผันนี้ ในบางครั้งมีได้หลายคำตอบ หรือไม่สามารถหาคำตอบได้ การคำนวณนี้ค่อนข้างยุ่งยากกว่าการคำนวณจลศาสตร์ไปข้างหน้า ซึ่งคำตอบของวิธีการคำนวณจลศาสตร์ผกผันอาจแบ่งได้ 2 รูปแบบคือ closed form และ numerial form ซึ่งโดยรูปแบบของ closed form นั้นเราสามารถหาคำตอบให้ออกมาในรูปของฟังก์ชั่น ซึ่งจะง่ายต่อการคำนวณหาค่าเพราะสามารถแทนค่าในตัวแปรของฟังก์ชั่นนั้น

ส่วนรูปแบบของ numerical iterative form นั้นใช้วิธีการสมมติค่าตัวเลข เริ่มต้นแล้วทำซ้ำไปเรื่อยๆ จนกว่าค่านั้นไม่มีการเปลี่ยนแปลง หรือเปลี่ยนแปลงน้อยมาก หรืออีกนัยหนึ่งคือ ค่านั้นลู่สู่ค่าหนึ่งซึ่งเป็นการยากต่อการคำนวณเพราะในบางครั้งอาจหาคำตอบไม่ได้

## 2.7 การคำนวณพลังงานรวมในการใช้ของหุ่นยนต์ฮิวมานอยด์

หุ่นยนต์ฮิวมานอยด์นับเป็นหุ่นยนต์ที่สามารถเคลื่อนที่ได้ ดังนั้นแล้วการบริหารจัดการการใช้พลังงาน ย่อมเป็นส่วนสำคัญที่ส่งผลต่อระยะเวลาการใช้งานต่อเนื่อง เพื่อที่จะหาว่าหุ่นยนต์ฮิวมานอยด์ใช้พลังงานไปเท่าใดในการทำกิจกรรมบางอย่าง ซึ่งเป็นข้อมูลสำคัญในการกำหนดคุณสมบัติของแบตเตอรี่ ภายในระยะเวลาการเดิน 1 คาบ จะใช้พลังงานในการเดินเท่ากับ กำลังของตัวขับทั้ง N ตัวมารวมกัน

โดยที่

คือ พลังงานที่ใช้ในการเดินหนึ่งคาบ

คือ กระแสไฟฟ้าของมอเตอร์ที่

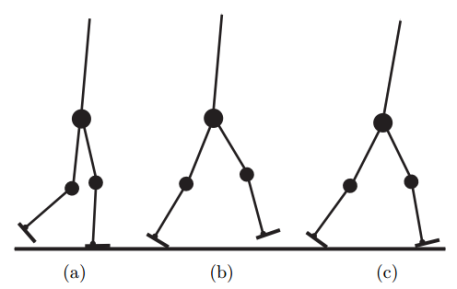
คือ ความต่างศักย์ไฟฟ้าตกคร่อมที่มอเตอร์ที่

คือ เวลา

คือ ระยะเวลาของวงจรเดินหนึ่งคาบ

## 2.8 เซนเซอร์ที่ใช้ในหุ่นยนต์ฮิวมานอยด์

### 2.8.1 Ground Contact Force Sensor

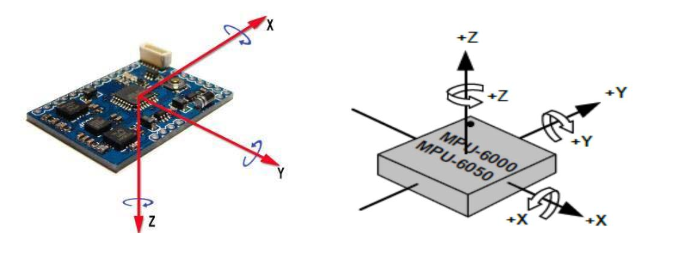


รูปที่ 2.6 ช่วงการเดินของหุ่นยนต์สองขา

การเดินของมนุษย์เราจะแบ่งออกเป็น 2 ช่วงคือ ช่วงที่ฝ่าเท้าของมนุษย์สัมผัสกับพื้นเพียงเท้าเดียว (Single Support Phase) ซึ่งขาที่สัมผัสกับพื้นจะเรียกว่า Stance leg ขาที่ลอยจากพื้นเรียกว่า Swing leg และ ช่วงที่ฝ่าเท้าของมนุษย์สัมผัสกับพื้นทั้งสองเท้า (Double Support Phase) ในแต่ละช่วงมีผลต่อพื้นที่สัมผัสระหว่างฝ่าเท้ากับพื้น และ พื้นที่สัมผัสระหว่างฝ่าเท้ากับพื้นนั้นมีผลต่อการควบคุม Zero Moment Point(ZMP)

โดยทั่วไปการควบคุมหุ่นยนต์สองขาให้สามารถเดินได้อย่างมีเสถียรภาพนั้นมักจะนิยมใช้การควบคุมตำแหน่งของจุด ZMP ของหุ่นยนต์ ให้อยู่ภายในขอบเขตของพื้นที่ที่รองรับน้ำหนักตัวอยู่ทั้งหมด (Support polygon) ซึ่งก็คือกรอบของหน้าสัมผัสฝ่าเท้าที่สัมผัสอยู่กับพื้น โดยจุด ZMP เป็นจุดที่เกิดจากผลรวมของโมเมนต์ที่เกิดจากความเฉื่อยและผลรวมโมเมนต์ของระบบหักล้างกันทำให้ผลลัพธ์ออกมาเป็นศูนย์ โดยพื้นที่สัมผัสนี้สามารถหาได้จากกลศาสตร์ของตัวหุ่นยนต์ และจุด ZMP สามารถหาได้จากเซนเซอร์วัดแรง (Force sensor) ที่นำไปติดไว้ใต้ฝ่าเท้าของหุ่นยนต์

### 2.8.2 Inertial Measurement Units (IMU)



รูปที่ 2.7 แสดงการตั้งแกนของ IMU

Inertial Measurement Unit (IMU) เป็นส่วนประกอบหลักที่ใช่ในการนำร่องเครื่องบิน ยาน-อวกาศ ดาวเทียม เรือ ขีปนาวุธ ซึ่งในตัวของ IMU ประกอบไปด้วยสองส่วนหลักคือ Accelerometers 3 ทิศทาง ในการรับความเร่งเชิงเส้น และ Gyroscopes 3 ทิศทาง ในการบอกความเร็วเชิงมุม เซนเซอร์ตัวนี้สามารถนำมาใช้ในการหาทิศทางการหมุนของตัวหุ่นยนต์ฮิวมานอยด์ได้

เซนเซอร์วัดความเร็ว (Gyroscope)[[9]](#footnote-10) เป็นอุปกรณ์สำหรับการวัดความเร็ว หรือการรักษาการปรับทิศทาง ขึ้นอยู่กับหลักการของการอนุรักษ์โมเมนตัมเชิงมุม ถ้าไม่มีการเคลื่อนที่ อัตราการเปลี่ยนแปลงมุมจะมีค่าเท่ากับศูนย์

เซนเซอร์วัดความเร่ง (Accelerometer)[[10]](#footnote-11) เป็นอุปกรณ์ที่ใช้วัดความเร่งเชิงเส้น โดยอาศัยการวัดแรงที่กระทำต่อน้ำหนัก อ้างอิงที่เกิดจากแรงโน้มถ่วงโลก ซึ่งแรงโน้มถ่วงของโลกจะเป็นเวกเตอร์ชี้ไปที่แกนกลางโลกเสมอ ตามกฎของนิวตัน

## 2.9 เครื่องมือที่ใช้พัฒนาระบบฮิวมานอยด์

### 2.9.1 Robot Middleware

Robot Middleware เป็นกรอบการทำงาน(framework) ที่มีความยืดหยุ่นสำหรับการพัฒนาซอฟแวร์ที่เกี่ยวข้องกับหุ่นยนต์ โดยมีเครื่องมือที่ช่วยติดต่อสื่อสารระหว่างอุปกรณ์ต่าง ๆของหุ่นยนต์ Robot Middleware ส่วนใหญ่จะใช้การติดต่อสื่อสารผ่านระบบเครือข่ายเน็ตเวิร์ค ทำให้การสื่อสารในระบบพื้นฐานเป็นอิสระต่อกัน และสามารถติดต่อสื่อสารกันกับอุปกรณ์ที่อยู่ภายนอกผ่านเครือข่ายเดียวกันได้

ปัจจุบันมี Robot Middleware ที่ถูกพัฒนาขึ้นมาให้ใช้อยู่หลายตัวเช่น

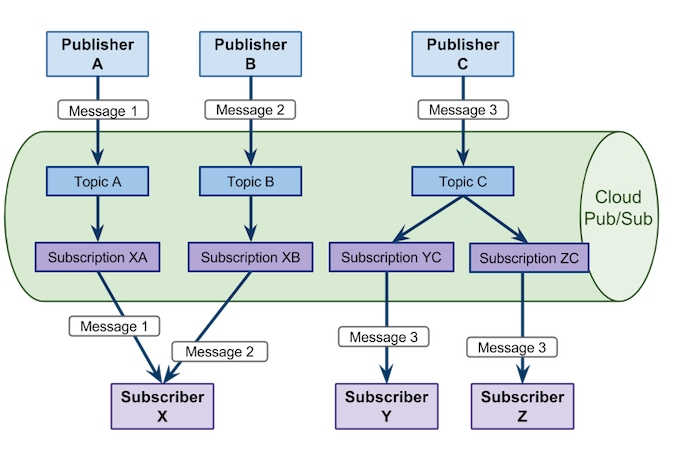
* Player Project: เป็นโปรเจคที่ใช้ในการสร้างซอฟแวร์เพื่อการศึกษาวิจัยที่มีความเกี่ยวข้องกับหุ่นยนต์และระบบเซนเซอร์
* YARP: เป็น open source ที่เขียนด้วยภาษา C++ ในการเชื่อมต่อกับเซนเซอร์ หน่วยประมวลผล และตัวขับเคลื่อนของหุ่นยนต์
* URBI: เป็น open source สำหรับพัฒนาแอพพลิเคชั่นที่เกี่ยวข้องกับหุ่นยนต์หรือระบบที่มีความซับซ้อน ใช้ภาษาพื้นฐานเป็นภาษา C++ ติดต่อสื่อสารได้ภายในเครือข่ายเดียวกันเท่านั้น (Local Network)
* MIRO: เป็นกรอบการทำงานของหุ่นยนต์ที่เคลื่อนที่ได้โดยเขียนในลักษณะเป็น OOP
* OpenRDK: เป็น open source สำหรับพัฒนาระบบที่มีความเป็นอิสระต่อกัน (Modules) สามารถใช้ช่องทางการติดต่อสื่อสารและหน่วยความจำร่วมกันได้
* ROS: เป็นแหล่งรวมเครื่องมือที่ใช้พัฒนาซอฟแวร์หุ่นยนต์

**Robot Operating System (ROS)**

ROS เป็นกรอบการทำงานที่ได้รับความนิยมและมีประสิทธิภาพในการทำงานมากที่สุดในปัจจุบัน เนื่องจาก ROS ได้รวบรวมเครื่องมือที่หลากหลายเอาไว้เป็นหมวดหมู่ ทำให้ช่วยลดความซับซ้อนและเพิ่มประสิทธิภาพในการทำงานกับแพลตฟอร์มของหุ่นยนต์ ระบบของ ROS ประกอบไปด้วยแพ็กเกจต่าง ๆมาประกอบกัน แต่ละแพ็กเกจจะประกอบไปด้วย โหนด(Node) โดย ROS นี้ทำหน้าที่เป็นตัวกลางในการติดต่อสื่อสารระหว่างอุปกรณ์ที่เป็นโหนดต่าง ๆ มีรูปแบบการสื่อสารแบบ Publish/Subscribe มีภาษาให้เลือกพัฒนาที่หลากหลาย เช่น C++, Python, Lisp, MATLAB หรือ JavaScript การสื่อสารรองรับโปรโตคอลทั้ง TCP และ UDP

**การรับและส่งข้อมูล (Publishing and Subscribing)**

การรับและส่งข้อมูลของ ROS เป็นรูปแบบของการรับส่งข้อมูลหรือข้อความที่ ผู้ส่ง(Publishers) จะส่งข้อความ(Message) ออกไปในลักษณะเหมือนการเผยแพร่โฆษณา โดยที่ไม่ได้บอกว่าผู้รับเป็นใคร แต่จะส่งข้อความไปยังผู้รับ(Subscribers) ที่กำลังรอรับข้อความนั้นอยู่ ผู้รับจะรอรับเฉพาะหัวข้อ(Topics) ที่ตัวเองรอรับเอาไว้เท่านั้น



รูปที่ 2.8 Google’s Cloud Pub/Sub Real-Time Messaging Service Diagram

คุณสมบัติของการ Publish และ Subscribe

* ผู้ส่งสามารถส่งข้อความไปยังหัวข้อไหนก็ได้
* ผู้รับสามารถรับข้อความจากหัวข้อไหนก็ได้
* ผู้ส่งหลายตัวสามารถส่งข้อความไปยังหัวข้อเดียวกันได้
* ผู้รับหลายตัวสามารถรับข้อความจากหัวข้อเดียวกันได้
* ผู้ส่งหนึ่งตัวสามารถส่งข้อความไปยังกี่หัวข้อก็ได้
* ผู้รับหนึ่งตัวสามารถรับข้อความจากกี่หัวข้อก็ได้

ประโยชน์ของการรับและส่งข้อมูลแบบนี้คือ เป็นตัวที่ช่วยในการทำให้การส่งข้อมูลมีความเป็นระเบียบเรียบร้อย และสามารถทำให้โปรแกรมแต่ละ Node สามารถรับข้อมูลเดียวกันได้โดยไม่ต้องมีการต่อคิว หรือรอเปลี่ยนการส่งผ่าน

### 2.9.2 การจำลอง

โปรแกรมจำลองการทำงานของหุ่นยนต์นั้นเป็นเครื่องมือที่สำคัญสำหรับนักหุ่นยนต์ การใช้โปรแกรมจำลองนั้นจะช่วยเพิ่มประสิทธิภาพในการทำงานหลายอย่าง เช่น ให้รู้ว่าหุ่นยนต์ที่ออกแบบนั้นสามารถทำงานได้อย่างที่ต้องการหรือไม่ กระบวนการคิดถูกต้องหรือไม่ โปรแกรมจำลองระบบส่วนใหญ่จะคำนวณพลวัตของหุ่นยนต์โดยใช้เครื่องมือคำนวณ open dynamics engine (ODE)

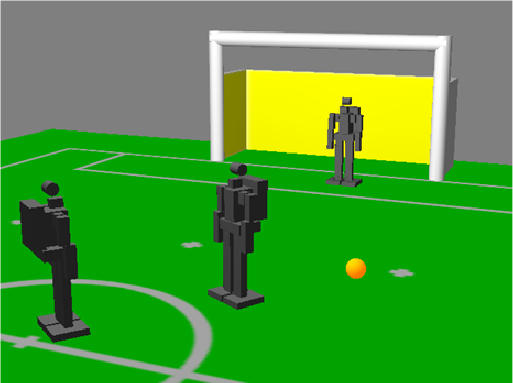
**USARSim**



รูปที่ 2.9 ผลลัพธ์จากการใช้โปรแกรม USARSim

USARSim เป็นโอเพนซอร์ซและเหมาะสำหรับทำหุ่นยนต์ประเภทกู้ภัยในซากเมือง โดยมีฐานการพัฒนามาจาก Unreal Tournament game engine ภายในโปรแกรมมีเครื่องมือสำหรับการทำงานวิจัย มีเซนเซอร์ของหุ่นยนต์ที่หลากหลาย เช่น เซนเซอร์รับภาพ หรือเซนเซอร์ตรวจความเคลื่อนไหว

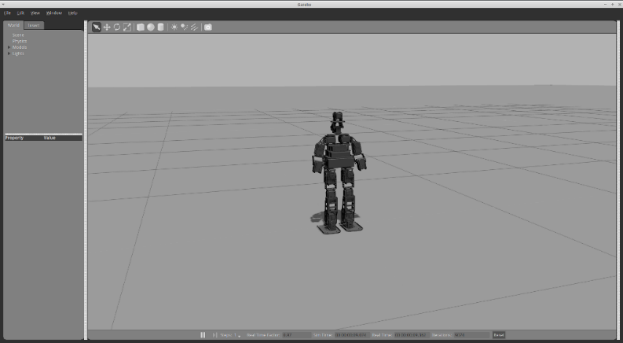
**MuRoSimF**



รูปที่ 2.10 ผลลัพธ์จากโปรแกรม MuRoSimF

MuRoSimF ย่อมาจากคำว่า Multi-Robot Simulation Framework เป็นเครื่องมือที่ช่วยทำระบบจำลองจาก Darmstadt University โปรแกรมระบบจำลองนี้มีการใช้งานที่ง่าย เหมาะสำหรับหุ่นยนต์หลายประเภท เช่น หุ่นยนต์เคลื่อนที่ด้วยล้อ หุ่นยนต์สองขา หรือหุ่นยนต์หลายขา สามารถคำนวณพลวัตร และการขัดกันของก้านต่อต่าง ๆได้

**Gazebo**



รูปที่ 2.11 โปรแกรม Gazebo ที่กำลังจำลอง HR OS5 Humanoid Research Platform

Gazebo เป็นโปรแกรมจำลองการทำงานของหุ่นยนต์ ที่มีความสามารถในการคำนวณการเดินและการเคลื่อนที่ของหุ่นยนต์ที่สลับซับซ้อนได้ สามารถเห็นภาพกราฟฟิคของหุ่นยนต์ขณะทำงาน โดยผู้ใช้สามารถกำหนดค่าตัวแปรทางฟิสิกส์ต่าง ๆได้ เช่นน้ำหนัก ค่าความเฉื่อย แรงเสียดทานของข้อต่อ ทำให้การออกแบบหุ่นยนต์หรือทดลองโปรแกรมได้เหมือนกับโลกจริง มีแสง มีเงา และ พื้นผิวของวัตถุ และที่พิเศษคือสามารถสังเคราะห์ค่าของเซนเซอร์ เซนเซอร์พร้อมสัญญาณรบกวน ค่าระยะทาง แรงบิด และอื่นๆ คำนวณพลศาสตร์ของหุ่นยนต์โดยใช้ตัวคำนวณทางฟิสิกส์เป็น Bullet หรือ Simbody ในการจำลองหุ่นยนต์ในโปรแกรมนี้จำเป็นต้องได้รับไฟล์ข้อมูลของหุ่นยนต์มาก่อนซึ่งอยู่ในรูปแบบของ URDF ซึ่ง URDF คือ ประเภทของไฟล์ที่บ่งบอกถึงความสัมพันธ์ของข้อต่อและก้านต่อแต่ละชิ้นในตัวหุ่นยนต์ มีความสามารถในการอธิบายถึงกลศาสตร์และการเคลื่อนที่ของหุ่นยนต์ รวมถึงตรวจสอบการขัดกันของก้านต่อในหุ่นยนต์ได้ ภายในไฟล์นี้จะประกอบไปด้วย

* **Link:** คือก้านต่อของหุ่นยนต์ซึ่งภายในจะสามารถบอกขนาด รูปร่าง สี และสามารถ import 3d mesh เข้ามาได้ด้วย อีกทั้งยังสามารถใส่รายละเอียดของการเคลื่อนที่ ของก้านต่อได้เช่น inertial matrix และ collision properties
* **Joint:** คือข้อต่อของหุ่นยนต์สามารถกำหนดกลศาสตร์และการเคลื่อนที่ได้เช่น Joint limits ของข้อต่อที่กำลังหมุนและความเร็วการหมุน ซึ่งข้อต่อมีหลายแบบที่สามารถกำหนดได้เช่น ข้อต่อแบบหมุน, ข้อต่อแบบเลื่อน, ข้อต้อต่อแบบยึดติด, ข้อต่อแบบต่อเนื่อง

**2.9.3 หน่วยประมวลผลระบบควบคุม**

ในการควบคุมหุ่นยนต์ฮิวมานอยด์ให้สามารถทำกิจกรรมต่าง ๆ ได้นั้น ส่วนที่มีความสำคัญที่ขาดไปไม่ได้ คือ หน่วยประมวลผลระบบควบคุม ถ้าไม่มีระบบควบคุมแล้ว อุปกรณ์ต่าง ๆ ที่ติดตั้งอยู่บนตัวของหุ่นยนต์ฮิวมานอยด์จะไม่สามารถติดต่อสื่อสารกันได้ ซอฟท์แวร์ของหุ่นยนต์ที่พัฒนามาทั้งหมดจะไม่สามารถใช้ได้ ทำให้หุ่นยนต์ฮิวมานอยด์ไม่สามารถทำงานในสิ่งที่ต้องการได้ การวางแผนระบบควบคุมที่นิยมใช้ในระบบหุ่นยนต์ฮิวมานอยด์ส่วนใหญ่ จะแบ่งออกเป็น 2 ส่วนด้วยกัน คือส่วนของหน่วยประมวลผลควบคุมระดับสูง และหน่วยประมวลผลควบคุมระดับต่ำ

**หน่วยประมวลผลควบคุมระดับสูง (High level controller)**

หน่วยประมวลผลควบคุมระดับสูงเป็นส่วนที่ใช้ประมวลผลการทำงานที่มีความซับซ้อนของระบบเช่น จลนศาสตร์ของหุ่นยนต์ การคำนวณหาเส้นทางการเดิน ในการคำนวณทางคณิตศาสตร์ของระบบเหล่านี้จำเป็นต้องมีการประมวลผลที่เร็ว และมีประสิทธิภาพ ย้อนไปในสมัยที่มีการพัฒนาหุ่นยนต์ฮิวมานอยด์ยุคแรกเริ่มนั้น หน่วยประมวลผลควบคุมระดับสูง จะใช้คอมพิวเตอร์เป็นตัวในการประมวลผลการคำนวณ ซึ่งคอมพิวเตอร์สมัยนั้นมีขนาดใหญ่ น้ำหนักมาก และต้องใช้หลังงานสูง ซึ่งต่างจากปัจจุบันนี้ที่มีการพัฒนาของเทคโนโลยีที่ก้าวหน้ามากขึ้น ทำให้คอมพิวเตอร์มีขนาดเล็กลงเทียบเท่ากับบอร์ดคอนโทรเลอร์ทั่วไป ในที่นี้จะทำการยกตัวอย่างของบอร์ดคอมพิวเตอร์ที่มีวางจำหน่ายในปัจจุบัน และทำการรวมรวมเทียบเคียงประสิทธิภาพ ของบอร์ดคอมพิวเตอร์แต่ละชนิดไว้ดังนี้

ตาราง 2.2 ตารางเปรียบเทียบประสิทธิภาพของบอร์ดคอมพิวเตอร์

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ODROID-XU4 | ODROID-C2 | ODROID-C1+ | RPi 3 Model B |
| CPU | [Samsung Exynos-5422](http://www.samsung.com/global/business/semiconductor/minisite/Exynos/products5octa_5422.html) : Cortex™-A15 and Cortex™-A7 big.LITTLE processor with 2GByte LPDDR3 RAM | Amlogic S905 SoC  4 x ARM Cortex-A53 1.5GHz  64bit ARMv8 Architecture @28nm | Amlogic S805 SoC  4 x ARM Cortex-A5 1.5GHz  32bit ARMv7 Architecture @28nm | Broadcom BCM2837  4 x ARM Cortex-A53 1.2Ghz  64bit ARMv7 Architecture @40nm |
| GPU | Mali-T628 MP6(OpenGL ES 3.1/2.0/1.1 and OpenCL 1.2 Full profile) | 3 x ARM Mali-450 MP 700MHz | 2 x ARM Mali-450 MP 600MHz | 1 x VideoCore IV 250MHz |
| RAM | 2Gbyte LPDDR3 RAM PoP stacked | 2GB 32bit DDR3 912MHz | 1GB 32bit DDR3 792MHz | 1GB 32bit LPDDR2 450MHz |
| Flash Storage | Micro-SD slot, eMMC 5.0 module connector | Micro-SD UHS-1 @83Mhz/SDR50 or  eMMC5.0 storage option | Micro-SD UHS-1 @78Mhz/SDR50 or  eMMC4.5 storage option | Micro-SD @ 50Mhz/SDR25  No eMMC storage option |
| USB2.0 Host | USB 3.0 Host x 2, USB 2.0 Host x 1 | 4 Ports | 4 Ports | 4Ports |
| USB2.0 Device /  OTG | N/A | 1 Port for Linux USB Gadget device or  USB host | 1 Port for Linux USB Gadget device or  USB host | No |

ตาราง 2.3 ตารางเปรียบเทียบประสิทธิภาพบอร์ดคอมพิวเตอร์

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ODROID-XU4 | ODROID-C2 | ODROID-C1+ | RPi 3 Model B |
| Ethernet / LAN | 10 / 100 / 1000 Mbit/s NO WIFI | 10 / 100 / 1000 Mbit/s | 10 / 100 / 1000 Mbit/s | 10 / 100 Mbit/s + WIFI |
| Video Output | HDMI 1080p | HDMI 2.0 4K / 60Hz | HDMI 1.4 | HDMI 1.4 / RCA / DSI |
| Audio Output | HDMI / I2S | HDMI / I2S | HDMI / I2S | MDMI / 3.5mm Jack / I2S |
| Camera Input | n/d | USB 720p | USB 720p | MIPI CSI 1080p |
| Real Time Clock | RTC and 38 LDOs | No (unless using an add-on module) | Yes (on-board RTC) | No (unless using an add-on module) |
| IR Receiver | n/d | Yes (on-board IR sensor) | Yes (on-board IR sensor) | No (unless using an add-on module) |
| IO Expansion | 30Pin : GPIO/IRQ/SPI/ADC, 12Pin : GPIO/I2S/I2C | 40 + 7 pin port  GPIO / UART / I2C / I2S / ADC | 40 + 7 pin port  GPIO / UART / SPI / I2C / I2S / ADC | 40 pin port  GPIO / UART / SPI / I2S |
| ADC | GPIO/IRQ/SPI/ADC | 10bit SAR 2 channels | 10bit SAR 2 channels | No (unless using an add-on board) |
| Heat sink | n/d | Included | Included | Optional |
| Size | 83 x 58 mm | 85 x 56 mm | 85 x 56 mm | 85 x 56 mm |
| Weight | 38g | 40g | 40g | 42g |
| Price | 59USD | 46USD | 35USD | 35USD |

**หน่วยประมวลผลควบคุมระดับต่ำ (Low level controller)**

หน่วยประมวลผลควบคุมระดับต่ำเป็นส่วนที่รับคำสั่งบางอย่างมาจาก หน่วยประมวลผลควบคุมระดับสูง มีประสิทธิภาพในการประมวลผลการคำนวณที่น้อยกว่า เนื่องจากการออกแบบสถาปัตยกรรมภายในระบบไม่เอื้ออำนวยต่อการคำนวณที่มีความซับซ้อน แต่มีความสามารถในการประมวลผลระบบที่เป็นคาบได้อย่างแม่นยำ ในด้านการทำหุ่นยนต์ฮิวมานอยด์นั้นมักจะใช้หน่วยประมวลผลควบคุมระดับต่ำ ในการติดต่อกับอุปกรณ์ต่าง ๆบนตัวของหุ่นยนต์ฮิวมานอยด์โดยตรง เช่น ตัวขับเคลื่อน เซนเซอร์รับค่า หรือไฟแสดงสถานะต่างๆของหุ่นยนต์

ตาราง 2.4 ตารางเปรียบเทียบบอร์ดไมโครคอลโครเลอร์

|  |  |  |
| --- | --- | --- |
|  | Arduino 2560 | Nucleo |
| CPU | ATmega2560 | ARM 32-bit Cortex M4 CPU |
| Core clock | 16 MHz | 100 MHz |
| Logic Level | 5V | 3.3V |
| Flash memory | 256 KB | 512 KB |
| GPIO | 54 | 50 |
| ADC | 10 bit with 16 channels | 12 bit with 16 channels |
| I2C | 1 | 3 |
| UART | 4 | 3 |
| SPI | 1 | 5 |
| RTC | ไม่มี | มี |

จากตารางข้างต้นเป็นตารางเปรียบเทียบประสิทธิภาพและความเหมาะสมกับการใช้งาน จะเห็นได้ว่า Nucleo มีประสิทธิภาพมากกว่าในหลายๆด้านไม่ว่าจะเป็น Core Clock ที่เร็วถึง 100 MHz หรือ GPIO ที่มีมาให้ 50 ช่องการเชื่อมต่อ อีกทั้งยังมี I2C ซึ่งเป็นรูปแบบที่ใช้ในการติดต่อกับ IMU ที่ต้องใช้ ทั้งนี้ ทั้ง 2 รุ่นที่ได้ทำการเปรียบเทียบนี้ไม่รองรับ RS – 485 โดยตรง ซึ่งเป็นรูปแบบการสื่อสารที่จะใช้กับการติดต่อกับตัวขับเคลื่อน ฉะนั้นแล้วทางผู้วิจัยจึงเลือกที่จะใช้ Nucleo แล้วมีการใช้โมดูลการสื่อสาร UART to RS-485 เพิ่มเติม

## 2.10 วิพากษ์วิจารณ์หุ่นยนต์ฮิวมานอยด์

### 2.10.1 Open source platform

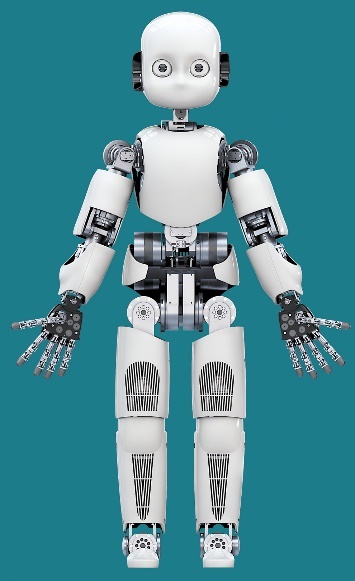
หุ่นยนต์ Open source platform คือ หุ่นยนต์ที่เปิดให้ผู้ใช้ทั่วไปสามารถเข้าถึง ปรับปรุง เรียนรู้ และพัฒนาตามได้ด้วยตนเอง ซึ่งข้อมูลที่กล่าวมานั้นสามารถหาได้จากเว็บไซต์ของผู้พัฒนาหุ่นยนต์ ปัจจุบันมีหุ่นยนต์ฮิวมานอยด์ที่เป็นเปิดให้เข้าถึงหลายรูปแบบแตกต่างกันไป โดยจะยกตัวอย่างหุ่นยนต์ที่เปิดให้บุคคลทั่วไปสามารถเข้าถึงได้ 4 ตัวอย่างดังนี้

* **Poppy** [2]

รูปที่ 2.12 Poppy Humanoid Robot

หุ่นยนต์ฮิวมานอยด์ Poppy ถูกสร้างขึ้นมาเพื่อใช้ในงานศิลปะ การวิจัยและการศึกษาโดยเฉพาะ หุ่นยนต์ Poppy ประกอบด้วยส่วนของฮาร์ทแวร์และซอฟแวร์ที่เปิดเป็นโอเพนซอร์ซ ใช้โมดูลที่มีชื่อว่า Pypot ภายในภาษา Python ในการพัฒนา ทุกคนสามารถเข้าถึงข้อมูลของหุ่นยนต์ฮิวมานอยด์ได้ในส่วน รายละเอียด การสอน การจำลอง และการพัฒนาต่าง ๆผ่านทาง  http://www.poppy-project.org หุ่นยนต์ Poppy เป็นหุ่นยนต์ที่มีโครงสร้างที่ผลิตจาก PLA, ABS ด้วยเครื่องพิมพ์สามมิติ ควบคุมการสั่งงานตัวขับด้วย Odroid UX4 ในระบบปฎิบัติการ Ubuntu 14.04 มีความสูง 83 เซนติเมตร น้ำหนัก 3.5 กิโลกรัม ใช้ IMU 9 องศาอิสระ ในการควบคุมเสถียรภาพในการเดินของตัวเอง มีองศาอิสระทั้งหมด 25 องศา ประกอบไปด้วย ขาข้างละ 6 องศาอิสระ แขนข้างละ 4 องศาอิสระ ลำตัว 3 องศาอิสระ และ หัว 2 องสาอิสระ

* **iCub** [3]



รูปที่ 2.13 Icup Humanoid โดย Istituto Italiano di Tecnologia

หุ่นยนต์ฮิวมานอยด์ Icup ถูกออกแบบโดยมหาวิทยาลัยหลายแห่งในยุโรปรวมตัวกันในชื่อ the RobotCub และถูกสร้างขึ้นโดย Istituto Italiano di Tecnologia (IIT) [[11]](#footnote-12) และถูกสร้างขึ้นโดย Istituto Italiano di Tecnologia (IIT) ตัวหุ่นยนต์มีความสูง 1 เมตร น้ำหนักโดยรวมประมาณ 22 กิโลกรัม ใช้วัสดุในการสร้างแตกต่างกันไปในแต่ละส่วนของร่างกายโดยจะใช้ aluminum alloy AI6082 สำหรับส่วนที่ต้องรับภาระความเครียด(Stress) น้อย aluminum alloy 7075(Ergal) สำหรับส่วนที่ต้องรับภาระความเครียดปานกลางถึงสูง และใช้ Stainless Steel 17-4PH ในส่วนของเพลาข้อต่อต่าง ๆเพื่อให้มีความแข็งแรงสูง โดยหุ่นยนต์ถูกออกแบบให้มีลักษณะเหมือนเด็กอายุ 3-4 ขวบ ควบคุมโดย PC104 controller ใช้ภาษา C++ ซึ่งติดต่อสื่อสารกับ มอเตอร์ และเซนเซอร์ ผ่านทาง CANBus ใช้เส้นเอ็นในการขับเคลื่อนข้อต่อส่วนมือและไหล่ ซึ่งนิ้วถูกร้อยด้วย เคเบิลเคลือบ  [Teflon](https://en.wikipedia.org/wiki/Teflon) อยู่ภายในและคลายกลับได้ด้วยแรงของสปริง มุมของข้อต่อใช้การออกแบบให้มี Hall-effect sensor ในการอ่านค่าของแรงบิดที่เกิดขึ้น มีองศาอิสระทั้งหมด 53 องศา ประกอบไปด้วย แขนข้างละ 7 องศาอิสระ มือข้างละ 9 องศาอิสระ หัว 6 องศาอิสระ ลำตัว 3 องศาอิสระ และขาข้างละ 6 องศาอิสระ

โดยในส่วนของหัวจะประกอบไปด้วย กล้องสองตัว ไมโครโฟน และไฟแสดงอารมณ์บริเวณปากและคิ้ว หุ่นยนต์นี้ไม่ได้ถูกออกแบบให้มีการทำงานอัตโนมัติ ซึ่งผลตามมาก็คือไม่มีแบตเตอรี่ภายในตัว แต่ใช้แหล่งพลังงานจากสายเคเบิล และสาย LAN

* **DARWIN-OP** [4]

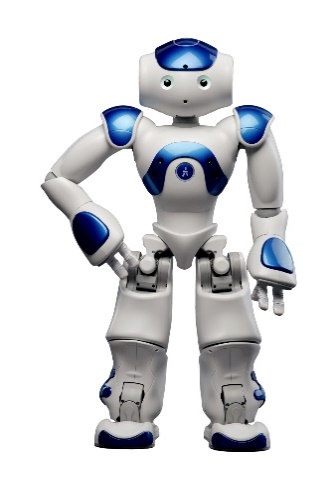
Figure 2 Darwin-op

รูปที่ 2.14 Darwin-op

หุ่นยนต์ฮิวมานอยด์ Darwin-OP เป็นชื่อย่อมาจาก Dynamic [Anthropomorphic](https://en.wikipedia.org/wiki/Anthropomorphic) [Robot](https://en.wikipedia.org/wiki/Robot) with Intelligence–[Open Platform](https://en.wikipedia.org/wiki/Open_Platform) ซึ่งสามารถรับภาระโหลดได้สูง และมีความสามารถในการเคลื่อนที่แบบ Dynamic หุ่นยนต์ได้ถูกออกแบบและพัฒนา โดย [Korean](https://en.wikipedia.org/wiki/Korea) robot manufacturer [Robotis](https://en.wikipedia.org/wiki/Robotis) โดยความร่วมมือกับ [Virginia Polytechnic Institute and State University](https://en.wikipedia.org/wiki/Virginia_Tech), [Purdue University](https://en.wikipedia.org/wiki/Purdue_University) และ [University of Pennsylvania](https://en.wikipedia.org/wiki/University_of_Pennsylvania) หุ่นยนต์ Darwin-OP มีองศาอิสระทั้งหมด 20 องศา ประกอบไปด้วย ขาข้างละ 6 องศาอิสระ แขนข้างละ 3 องศาอิสระ และหัว 2 องศาอิสระ

ขับเคลื่อนด้วย [DYNAMIXEL](https://en.wikipedia.org/wiki/DYNAMIXEL) MX-28T มีความสูง 45.45 เซนติเมตร น้ำหนัก 2.9 กิโลกรัม ออกแบบระบบภายในด้วย 1.6 GHz Intel Atom Z530 (32 bit) ใช้ controller ARM CortexM3 STM32F103RE 72 MHz และมีเซนเซอร์คุมเสถียร 3-axis gyro, 3-axis accelerometer

* **Nao** [5]



รูปที่ 2.15 Nao Humanoid robot

หุ่นยนต์ฮิวมานอยด์ Nao เป็นหุ่นยนต์ฮิวมานอยด์ขนาดกลาง จากประเทศฝรั่งเศษ พัฒนาโดย  Aldebaran Robotics เมื่อปี 2004 ในปี 2007 นาโอได้นำไปแทนที่หุ่นยนต์สุนัขของ Sony ชิอ Aibo ขณะถูกใช้ใน  RoboCup Standard Platform League (SPL) Nao ได้ถูกนำไปใช้ใน Robocup 2008 และ 2009 หลายหลายรุ่นของ Nao มีองศาอิสระตั้งแต่ 14 องศาอิสระ 21 องศาอิสระ และ 25 องศาอิสระ สำหรับเพื่องานวิจัยนั้นมีถึง 25 องศาอิสระ โดยเพิ่มเติมมือสองข้างเอาเข้าไปเพื่อให้สามารถหยิบจับสิ่งของได้ ภายในหุ่นยนต์ถูกควบคุมด้วยระบบปฎิบัติการ NAO 2.0 (Linux-based) มีความสูง 58 เซนติเมตร น้ำหนัก 4.3 กิโลกรัม ส่วน Sensor จะประกอบไปด้วย IMU, Ultrasound captors, ไมโครโฟน 4 ตัว ลำโพง 2 ตัว กล้อง 2 ตัว เพื่อใช้ประโยชน์ในการทำงานวิจัยต่างๆ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | iCub | Poppy | DARWIN-OP | Nao |
| วัตถุประสงค์ | ใช้ในการวิจัยกระบวนการคิดของมนุษย์ และปัญญาประดิษฐ์ | ใช้ในด้านการเรียนรู้และวิจัยในหลากหลายด้าน | ใช้ในงานวิจัยหลายด้านเช่น ปัญญาประดิษฐ์ วิธีการเดิน และการมองเห็น | ใช้ในงานวิจัย การศึกษา และ ให้ความบันเทิง |
| High Level Controller | PC104 controller | Ordroid XU4 | 1.6 GHz Intel Atom Z530 (32 bit) | [Intel Atom](https://en.wikipedia.org/wiki/Intel_Atom) @ 1.6 GHz |
| Low Level Controller | - | - | ARM CortexM3 STM32F103RE | - |
| ระบบปฎิบัติการ | Linux | Linux(Ubuntu 14.04) | Linux | NAO qi 2.0 ([Linux](https://en.wikipedia.org/wiki/Linux)-based) |
| Sensor | กล้อง stereo ,ไมโครโฟน,force sensor, Hall effect | IMU 9 DoF  กล้องเลนส์กว้าง | 3-axis gyro, 3-axis accelerometer,กล้อง,ไมโครโฟน,force sensor(FRS X4) | กล้อง,ไมโครโฟน,IMU, Infrared Sensor, Ultrasonic Sensor |
| วัสดุของโครงสร้าง | Aluminum alloy (AI6082) Stainless Steel 17-4PH | 3D - printed PLA | Aluminum Alloy 5052 | Plastic |
| องศาอิสระ | 53 | 25 | 20 | 25 |
| ความสูง (เมตร) | 1.04 | 0.83 | 0.4545 | 0.58 |
| น้ำหนัก (กิโลกรัม) | 22 | 3.5 | 2.9 | 4.3 |
| แหล่งพลังงาน | แหล่งพลังงานภายนอกจากสาย Cable | แหล่งพลังงานภายนอกจากสาย Cable | แบตเตอรี่ 12V | แบตเตอรี่ |

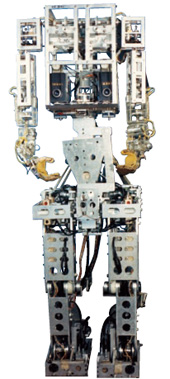
ตารางที่ 2.5 ตารางเปรียบเทียบหุ่นยนต์ฮิวมานอยด์แบบ open source

จากตารางข้างต้น ซึ่งเป็นตารางที่นำหุ่นยนต์ฮิวมานอยด์แบบ Open source ที่มีวัตถุประสงค์เพื่อใช้ในงานวิจัยและการเรียนรู้ จะพบว่าเป็น Linux based ทั้งหมด และจากวัสดุแสดงให้เห็นว่ามีตัวอย่างที่ใช้ PLA 3D print มาทำโครงสร้างในหุ่นยนต์ poppy ที่มีความสูง 83 ซม. หนัก 3.5 กก.

### 2.10.2 Non-Open source platform

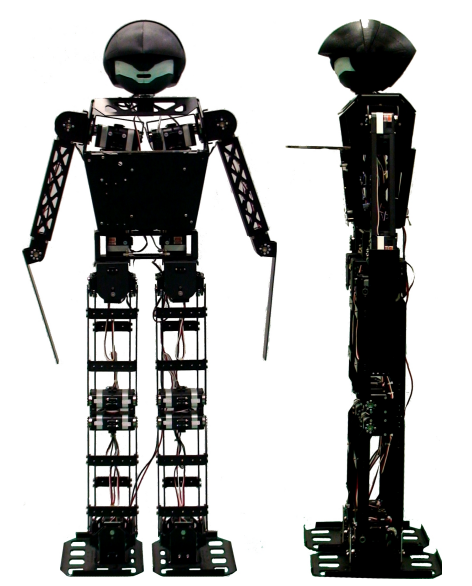
หุ่นยนต์ Non-open source platform คือหุ่นยนต์ที่สร้างมาเฉพาะเจาะจงสำหรับการวิจัย การสำรวจ หรือการแข่งขันโดยเฉพาะ ไม่เปิดให้บุคคลภายนอกเข้าศึกษาหรือแก้ไขปรับปรุง ซึ่งทำให้หุ่นยนต์ประเภทนี้ไม่เหมาะสำหรับผู้วิจัยที่จะเรียนรู้และศึกษาด้วยตนเอง เพราะมีขนาดใหญ่ ใช้ทรัพยากรมาก และการออกแบบมีความซับซ้อน เรียนรู้ยากกว่าหุ่นยนต์แบบ Open source โดยจะยกตัวอย่างหุ่นยนต์แบบ Non-Open source platform ทั้งหมด 3 ชนิดด้วยกันคือ

* Wabot [6]



รูปที่ 2.16 Wabot-1

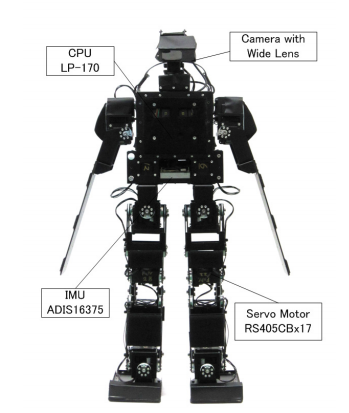
หุ่นยนต์ฮิวมานอยด์มีการพัฒนาในช่วงแรกเริ่มมาตั้งแต่ปี 1973 หุ่นยนต์ฮิวมานอยด์ ตัวแรกชื่อ Wabot-1 เริ่มสร้างโดยมหาวิทยาลัย Waseda ที่ประเทศญี่ปุ่นมีความสูง 180 เซนติเมตร น้ำหนัก 210 กิโลกรัม โดยหุ่นยนต์สามารถติดต่อสื่อสารกับมนุษย์ได้ด้วยภาษาญี่ปุ่น สามารถวัดระยะและทิศทางได้โดยใช้การรับรู้ผ่านทางตาและหูเทียม หุ่นยนต์ Wabot-1 นั้นเดินได้ด้วยขาของตนเอง สามารถหยิบและเคลื่อนย้ายวัตถุด้วยมือ ต่อมาในปี 1984 มหาวิทยาลัย Waseda ได้พัฒนาหุ่นยนต์ฮิวมานอยด์ที่ชื่อ Wabot-2 โดยหุ่นยนต์สามารถสื่อสารกับมนุษย์ได้ สามารถอ่านโน๊ตเพลงและเล่นดนตรีโดยใช้ electronic organ แบบง่ายๆได้ และในปี 1985 บริษัท Hitachi ได้สร้างหุ่นยนต์ WHL-11 ที่มีสองขาเหมือนมนุษย์ ซึ่งสามารถเดินแบบ Static Walking บบนพื้นราบได้ด้วยความเร็ว 13 วินาทีต่อหนึ่งก้าว และสามารถเลี้ยวได้

* Robo-Erectus Tr-2010 [7]

รูปที่ 2.17 Robo-Erectus Tr-2010

Robo-Erectus Tr-2010 เป็นหุ่นยนต์เลียนแบบมนุษย์เพื่อใช้ในการแข่งขันหุ่นยนต์ฮิวมานอยด์เตะฟุตบอล สร้างจากการใช้อลูมิเนียมพับ (Sheet Metal) ส่วนสูง 106 ซม. ใช้ระบบการประมวลผลขั้นสูงด้วยบอร์ด Gumstix และระบบประมวลผลขั้นต่ำ (Low Level Controller) ด้วยบอร์ด PIC มีน้ำหนักโดยรวมทั้งตัว 7.8 กก. Robo-Erectus มีองศาอิสระทั้งหมด 18 องศา ประกอบไปด้วย ขาข้างละ 5 องศาอิสระ แขนข้างละ 3 องศาอิสระ และ หัว 2 องศาอิสระ

ความเร็วในการเดิน 40 เซนติเมตรต่อวินาที มีเซนเซอร์ทั้งหมดจำนวน 2 ตัวคือ กล้องที่มีความละเอียดของภาพเป็น 640 x 480 และ IMU

* CIT Brains [8]

รูปที่ 2.18 CIT Brains

หุ่นยนต์ฮิวมานอยด์ CIT Brains เป็นหุ่นยนต์ที่ใช้ในการแข่งขัน RoboCup soccer ขนาดเล็ก ซึ่งมีการร่วมมือกันระหว่าง Hajime Research Institute และ Chiba Institute of Technology (CIT) มีความสูง 60 เซนติเมตร น้ำหนัก 3.5 กิโลกรัม ใช้มอเตอร์ Futaba RS405CB จำนวน 17 ตัวในการควบคุม ซึ่งทำให้มีองศาอิสระทั้งหมด 17 องศา ประกอบไปด้วย ขาข้างละ 5 องศาอิสระ แขนข้างละ 2 องศาอิสระ และ หัว 1 องศาอิสระ

ความเร็วในการเดินสูงสุด 40 เซนติเมตร/วินาที ใช้การประมวลผลขั้นสูงด้วยบอร์ด COMMEL LP-170C ในระบบปฎิบัติการ Linux (Ubuntu12.04LTS) พร้อมทั้งมีแหล่งพลังงานภายในตัวด้วยแบตเตอรี่ 11.1V 5000mAh

# บทที่ 3 วิธีการดำเนินงาน

## 3.1 หน้าที่ความรับผิดชอบ

* นายจิรัฏฐ์ ศรีรัตนอาภรณ์ และ นายเจษฎากร ทาไชยวงค์

ออกแบบและจัดทำโครงสร้างของหุ่นยนต์ฮิวมานอยด์

ติดตั้งตัวควบคุมและเซนเซอร์ที่ใช้ในหุ่นยนต์ฮิวมานอยด์

* นายวุฒิภัทร โชคอนันตทรัพย์

ออกแบบและพัฒนาระบบพื้นฐานสำหรับการติดต่อสื่อสารระหว่างอุปกรณ์

## ขั้นตอนการดำเนินงาน

การออกแบบหุ่นยนต์ฮิวมานอยด์ที่จะออกแบบนั้น ผู้วิจัยได้มีการตั้งชื่อให้หุ่นยนต์ โดยใช้ชื่อว่า อุทัย (UTHAI) มาจากภาษาอังกฤษคำว่า Universal Template for Humanoid Algorithm Interface เพื่อให้สมกับเป็นหุ่นยนต์ฮิวมานอยด์ที่ใช้สำหรับงานวิจัยและพัฒนาต่อยอด

การดำเนินงานและการออกแบบการสร้างหุ่นยนต์ฮิวมานอยด์อุทัย มีการลักษณะการทำงานออกเป็น 2 ส่วน ส่วนแรกคือ ส่วนของฮาร์ดแวร์ที่เกี่ยวกับโครงสร้างทางกลของหุ่นยนต์ฮิวมานอยด์ เช่น ข้อต่อ ก้านต่อ ฝ่าเท้า รวมไปถึงระบบอิเล็กทรอนิกส์ ตัวประมวลผลการควบคุม เซนเซอร์ ตัวขับเคลื่อนต่าง ๆ และส่วนที่สองคือ ส่วนของซอฟท์แวร์ที่เกี่ยวข้องกับการติดต่อสื่อสารกันเบื้องต้น การควบคุมตัวขับเคลื่อนที่ข้อต่อ การอ่านค่าเซนเซอร์ และระบบพื้นฐานสำหรับการใช้งาน ขั้นตอนในการทำงานวิจัยครั้งนี้

1. ออกแบบระบบโครงสร้างทางกลของหุ่นยนต์ฮิวมานอยด์ (Solidworks)
2. นำแบบจำลองของหุ่นยนต์เข้าทดสอบการเคลื่อนไหวด้วยโปรแกรมจำลองระบบ (Gazebo)
3. ทดสอบเซนเซอร์ต่าง ๆ เพื่ออ่านค่าออกมาใช้งาน
4. จัดทำชิ้นส่วนโครงสร้างทางกล และประกอบหุ่นยนต์ขึ้นจริง
5. วางแผนระบบการเชื่อมต่อสื่อสาร และการส่งข้อมูลของอุปกรณ์ต่าง ๆ
6. ติดตั้งอุปกรณ์บอร์ดควบคุมการทำงาน และเซนเซอร์เข้ากับตัวหุ่นยนต์
7. จัดทำคู่มือและเอกสารการใช้งานส่วนต่าง ๆ ของหุ่นยนต์

ตาราง 3.1 แสดงระยะเวลาการดำเนินงานของงานวิจัย

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| สิ่งที่ต้องทำ | | ปีพ.ศ.2560 | | | | | | | | | | | | | | | | | | | |
| ส.ค. | | | | ก.ย. | | | | ต.ค. | | | | พ.ย. | | | | ธ.ค. | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | ศึกษาทฤษฏีและงานวิจัยที่เกี่ยวข้อง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | วางขอบเขตของหุ่นยนต์ UTHAI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | ออกแบบโครงสร้างของหุ่นยนต์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | ออกแบบระบบส่งกำลังของหุ่นยนต์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | ออกแบบระบบไฟฟ้าของหุ่นยนต์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | วาดแบบ 3 มิติบนโปรแกรม Solidworks |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | นำแบบ 3 มิติของหุ่นยนต์ที่ได้ไปใช้ในระบบจำลอง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | สั่งซื้ออุปกรณ์และชิ้นส่วนทั้งหมด |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | ประกอบชิ้นงาน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | พัฒนาโปรแกรมสำหรับติดต่อฮาร์ดแวร์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | ทดลองโปรแกรมกับหุ่นยนต์จริง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| สิ่งที่ต้องทำ | | ปีพ.ศ. 2561 | | | | | | | | | | | | | | | | | | | |
| ม.ค. | | | | ก.พ. | | | | มี.ค. | | | | เม.ย. | | | | พ.ค. | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 12 | จัดทำคู่มือรายละเอียดหุ่นยนต์ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | นำระบบควบคุมการเดินจากระบบจำลองใส่ลงหุ่นยนต์จริง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | ทำการทดลองการเดินของหุ่นยนต์และเก็บผลทดลอง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการที่จะเริ่มต้นทำงานวิจัยเกี่ยวกับฮิวแมนนอยด์นั้นสิ่งจำเป็นที่ต้องทำในอันดับแรกเลยคือการศึกษาสิ่งที่เคยมีอยู่ หรืองานวิจัยได้ทำเอาไว้แล้ว ศึกษาทำความเข้าใจในข้อดี - ข้อเสีย ของวิธีหรือกระบวนการต่าง ๆ เพื่อนำมาประยุกต์ใช้กับหุ่นยนต์ฮิวมานอยด์ UTHAI เพื่อให้ได้สิ่งที่ดีที่สุด

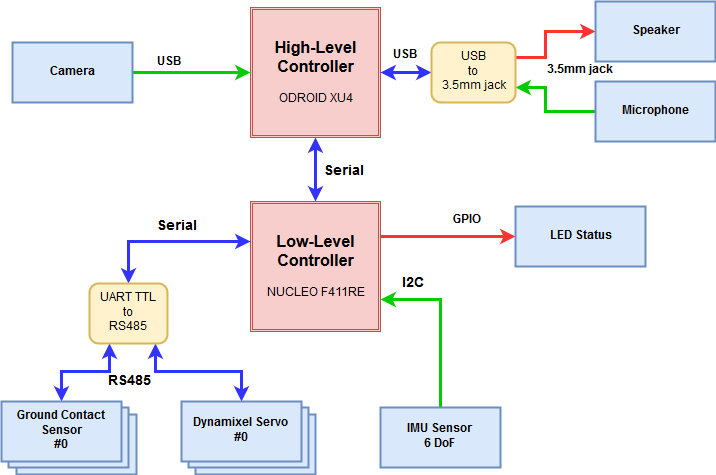
โดยการศึกษานั้นจะเริ่มต้นจากการศึกษาโครงสร้างทางกลของหุ่นยนต์ฮิวมานอยด์ที่มีอยู่แล้วและมีสิ่งที่ต้องดูเป็นพิเศษคือ วิธีการเชื่อมต่อกันระหว่างก้านต่อและข้อต่อ, จุดที่ใช้ติดตั้ง IMU และการเลือกใช้วัสดุให้เหมาะสม รวมถึงการทำงานของเซ็นเซอร์ที่จำเป็นต้องใช้ในการควบคุมการทำงานของหุ่นยนต์ และการทำงานของอุปกรณ์ขับเคลื่อนต่างๆ เช่น การสั่งงานมอเตอร์ Digital servo ผ่าน Protocol ของ Dynamixel

### ออกแบบสถาปัตยกรรมของหุ่นยนต์

หลักการออกแบบสถาปัตยกรรมของหุ่นยนต์ฮิวมานอยด์ UTHAI จะออกแบบระบบให้อยู่บนระบบพื้นฐาน ROS เนื่องจากการใช้กรอบการทำงานที่มีประสิทธิภาพ และความยืดหยุ่นสูง จะช่วยทำให้สามารถปรับเปลี่ยนระบบการควบคุมของหุ่นยนต์ฮิวมานอยด์ได้ง่ายและรวดเร็ว

ดังนั้นแล้วผู้วิจัยจึงได้แบ่งการประมวลผลออกเป็น 2 ส่วนคือ

1. หน่วยประมวลผลควบคุมระดับสูง (High Level Controller)
2. หน่วยประมวลผลควบคุมระดับต่ำ (Low Level Controller)

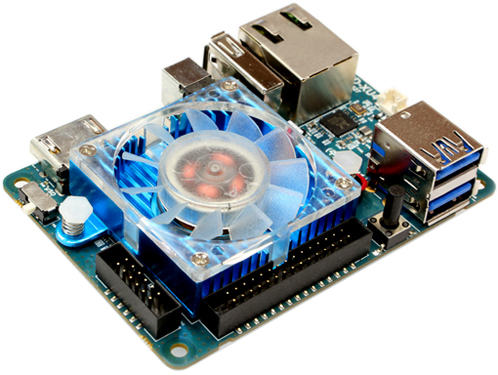


รูปที่ 3.1 สถาปัตยกรรมของหุ่นยนต์ฮิวมานอยด์ UTHAI

ตารางที่ 3.2 ตารางแสดงรายละเอียด Hardware ของหุ่นยนต์ UTHAI

|  |  |
| --- | --- |
| **รายละเอียด** | **สิ่งที่ใช้ในหุ่นยนต์** |
| High Level Controller | Ordroid XU4 |
| Low Level Controller | Nucleo F411RE |
| ระบบปฏิบัติการ | Linux (Ubuntu 16.04) |
| Sensor | กล้อง, ไมโครโฟน, IMU9250, Ground Contact Sensor |
| วัสดุของโครงสร้าง | 3D print PLA |
| องศาอิสระ | 20 |
| ความสูง (เมตร) | 1.03 |
| น้ำหนัก (กิโลกรัม) | 5.5 |
| แหล่งพลังงาน | แบตเตอรี่ 12V และสามารถใช้แหล่งพลังงานจากภายนอกได้ |

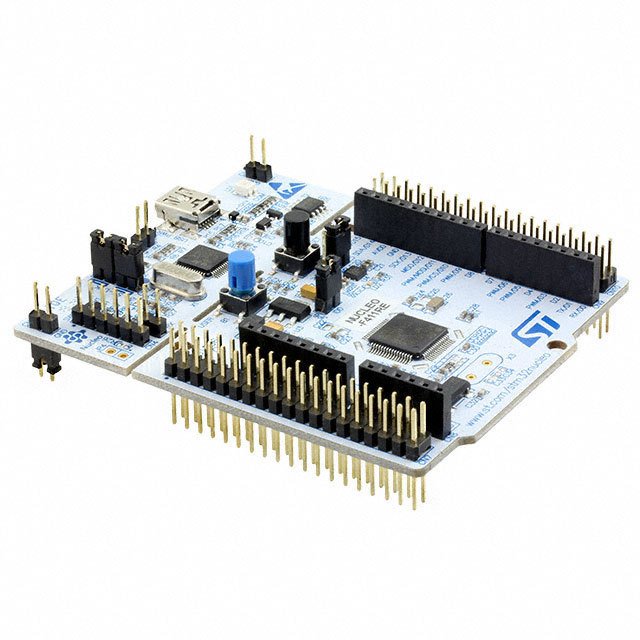
**หน่วยประมวลผลควบคุมระดับสูง (High level controller)**



รูปที่ 3.2 บอร์ดคอนโทรลเลอร์ Odroid XU4

ระบบควบคุมหลักของหุ่นยนต์ UTHAI นั้นจะอยู่ที่หน่วยประมวลผลขั้นสูง ใช้เป็นบอร์ดคอมพิวเตอร์ ODROID-XU4 ตัวประมวลผลหลักนี้ มีหน้าที่ในการทำการคำนวณ เส้นทางการเดินของหุ่นยนต์ให้มีเสถียรภาพ ตรวจการขัดกันของโครงสร้างของหุ่นยนต์ รวมไปถึงรับค่าข้อมูลภาพจากกล้อง และข้อมูลเสียงจากไมโครโฟนมาประมวลผล หลังจากนั้นจะทำการนำค่าทั้งหมดที่ได้จากการคำนวณ มาแปลงให้อยู่ในรูปของชุดข้อมูล แล้วส่งออกไปให้ระบบกลาง (ROS) ในการส่งต่อไปให้อุปกรณ์อื่นต่อไป

**หน่วยประมวลผลควบคุมระดับต่ำ (Low level controller)**



รูปที่ 3.3 บอร์ดไมโครคอนโทรลเลอร์ Nucleo F411RE

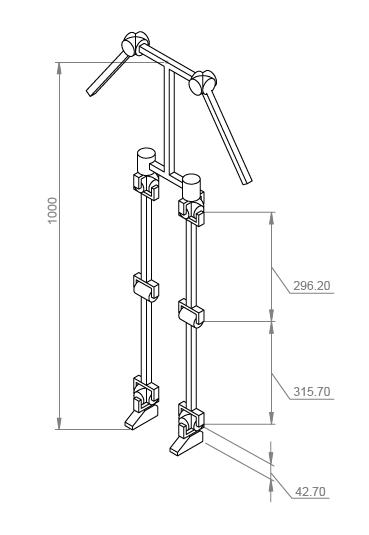
ระบบควบคุมขั้นต่ำเป็นหน่วยประมวลผลที่รองลงมาจาก บอร์ดคอมพิวเตอร์ โดยจะใช้บอร์ดไมโครคอลโทรเลอร์ Nucleo F411RE เป็นหน่วยประมวลผลขั้นต่ำ สำหรับในการติดต่อกับอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ที่อยู่ภายในตัวของหุ่นยนต์ เช่น ค่าเซนเซอร์ที่ฝ่าเท้าซึ่งสามารถบอกได้ว่าควรจะใช้สมการไหนในการคำนวณพลวัต หรือค่าของเซนเซอร์ IMU มีความสำคัญมาก ในการทำให้หุ่นยนต์ฮิวมานอยด์เดินได้อย่างมีเสถียรภาพ เมื่ออ่านค่าเซนเซอร์ต่าง ๆได้แล้ว หน่วยประมวลผลขั้นต่ำจะนำค่าที่ได้จากการอ่านเซนเซอร์เหล่านี้แปลงให้อยู่ในลักษณะของชุดข้อมูล แล้วส่งออกไปในระบบกลาง(ROS) นอกเหนือจากนี้หน่วยประมวลผลขั้นต่ำยังทำหน้าที่รับค่าคำสั่งมาจากระบบกลาง ในการสั่งงานให้หุ่นยนต์มีท่าทางต่าง ๆตามต้องการได้

### 3.2.3 ออกแบบระบบโครงสร้างทางกลของหุ่นยนต์

การออกแบบทางโครงสร้างทางกลนั้น ผู้วิจัยได้เลือกที่จะใช้โปรแกรม Solidworks เป็นเครื่องมือที่ช่วยในการพัฒนาโมเดลของหุ่นยนต์ฮิวมานอยด์ เนื่องจากโปรแกรม Solidworks เป็นโปรแกรมที่มีความสามารถในการขึ้นรูปและวาดแบบทางวิศวกรรม สามารถวิเคราะห์โครงสร้างทางกลของแบบจำลองได้ และมีการใช้งานอย่างแพร่หลาย อีกทั้งยังสามารถดาวน์โหลดโมเดลต่าง ๆ ที่มีคนพัฒนาเข้าเข้ามาใช้ร่วมกับการออกแบบของเราได้ และด้วยทางผู้วิจัยมีความคำนึงถึงความสามารถในการพัฒนาต่อยอดเป็นหลัก ดังนั้นการออกโครงสร้างทางกลของหุ่นยนต์ฮิวมานอยด์ UTHAI จึงถูกออกแบบมาเพื่อให้สามารถรองรับการเปลี่ยนแปลง แก้ไขส่วนต่าง ๆของตัวหุ่นยนต์เองได้ในอนาคตอีกด้วย

**3.2.3.1 โครงสร้างหุ่นยนต์**

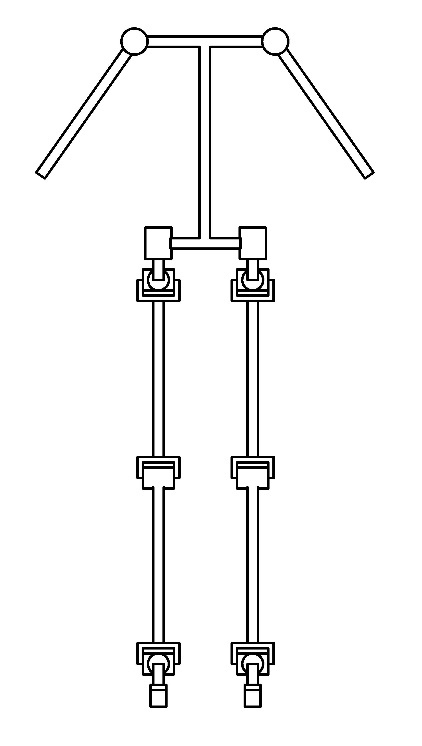
หุ่นยนต์ที่สร้างขึ้นประกอบด้วยส่วนของลำตัวและส่วนขา ในขาแต่ละข้างออกแบบให้เป็นลักษณะของข้อต่อหมุน (Revolute joint) เลียนแบบโครงสร้างของมนุษย์ซึ่งประกอบด้วย ส่วนของสะโพกที่มีองศาอิสระจำนวน 3 องศาอิสระ ส่วนของหัวเข่า 1 องศาอิสระและส่วนของข้อเท้า 2 องศาอิสระ รวมขาข้างละ 6 องศาอิสระ ระบบต้นกำลังที่ใช้เป็น Dynamixel servo การออกแบบหุ่นยนต์นั้น สิ่งแรกที่ต้องทำ คือ การกำหนดโครงสร้างของข้อต่อและก้านต่อขึ้นมาก่อน โดยวาดขึ้นมาเป็นเหมือนโครงกระดูก ซึ่งโครงสร้างนั้นทางผู้วิจัยจะอ้างอิงมาจากสัดส่วนของมนุษย์จริง ที่ประกอบด้วยส่วนของขาข้างละ 6 องศาอิสระ และมีจุด CoM อยู่บริเวณกระดูกเชิงกรานของตัวหุ่นยนต์เอง



หน่วย : mm

รูปที่ 3.4 แผนภาพแสดงแสดงโครงสร้างของหุ่นยนต์ UTHAI

เมื่อเราได้แบบจำลองของหุ่นยนต์ฮิวมานอยด์ออกมาแล้ว ลำดับต่อไปคือการกำหนดตำแหน่งการติดตั้งเซนเซอร์และตัวขับเคลื่อนต่างๆเข้าไป โดยมี Ground contact sensor ติดตั้งที่ใต้ฝ่าเท้าของหุ่นยนต์, IMU sensor ติดตั้งไว้ให้ใกล้กับจุด COM ของหุ่นยนต์ และ Dynamixel servo ติดตั้งไว้ที่ข้อต่อในแต่ละข้อต่อ



Ground Contact Sensor

IMU

รูปที่ 3.5 แผนภาพแสดงการติดตั้งเซนเซอร์ในจุดต่าง ๆ

ส่วนโครงสร้างหุ่นยนต์ฮิวมานอยด์ UTHAI ทางผู้วิจัยจะเลือกใช้วัสดุหลักเป็น PLA ที่ขึ้นรูปด้วยวิธีการขึ้นรูปสามมิติ และมีวัสดุเสริมบางชิ้นส่วนจากอลูมิเนียม เนื่องจากจะทำให้โครงสร้างมีน้ำหนักเบา สามารถปรับปรุงแก้ไขง่าย และมีราคาที่สมเหตุสมผล

ตารางที่ 3.3 ตารางแสดงสมบัติทางกลของวัสดุต่าง ๆ

|  |  |  |
| --- | --- | --- |
| Material | Longitudinal Tensile Strength (ksi) | Density (g/cm^3) |
| Carbon Fiber | 300 | 1.55 |
| Steel | 100 | 7.7 |
| Titanium | 120 | 4.34 |
| Aluminum | 35 | 2.7 |
| PLA 3D printing (50 % infill) | 3.5 | 1.26 |
| PLA 3D printing (100 % infill) | 5.5 | 1.26 |

**3.2.3.2 ระบบขับเคลื่อนของหุ่นยนต์ (Actuator)**

ในการสร้างหุ่นยนต์ฮิวมานอยด์นั้นระบบการขับเคลื่อนก็ถือว่าเป็นเรื่องสำคัญ เนื่องจากว่าถ้าหากระบบขับเคลื่อนไม่สามารถทำงานได้อย่างเต็มประสิทธิภาพ หรือหากมีการออกแบบที่ผิดพลาด จะส่งผลทำให้หุ่นยนต์ฮิวมานอยด์นั้นมีประสิทธิภาพในการทำงานลดลงตามไปด้วย ภายในงานวิจัยนี้ทางผู้จัดทำได้ใช้ตัวขับเคลื่อนเป็น Dynamixel digital servo EX-106 ซึ่งเป็นเซอร์โวมอเตอร์ที่เหมาะสำหรับทำหุ่นยนต์โดยเฉพาะ ภายในประกอบไปด้วย มอเตอร์กระแสตรง ชุดเฟืองมอเตอร์ ไดรเวอร์คอนโทรเลอร์ สามารถเชื่อมต่อกันผ่าน BUS RS-485 มีการควบคุมแบบ PID และแรงบิดที่สูง[[12]](#footnote-13)

การทำงานของตัวขับเคลื่อนนี้สามารถทำได้ 2 รูปแบบคือ joint mode และ wheel mode

1. Joint Mode สามารถที่จะควบคุม Torque Speed และ position ได้ความละเอียดในการควบคุม 10-bit (0-1023) หมุนได้อยู่ในช่วง 0-250 องศา
2. Wheel Mode สามารถที่จะควบคุม Torque Speed และ direction ได้ ความละเอียดของความเร็วมอเตอร์เท่ากับ 10bit (0-1023) สามารถหมุนได้ครบ 360 องศาได้[[13]](#footnote-14)



รูปที่ 3.6 แสดงประสิทธิภาพของมอเตอร์ EX-106+

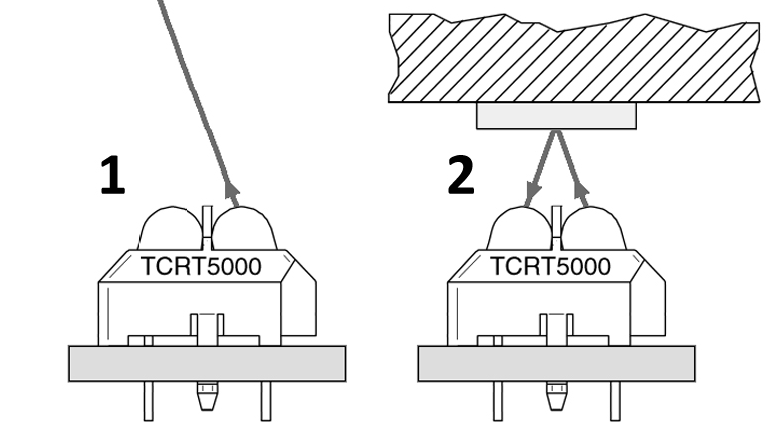
**3.2.3.2 ระบบตัวรับรู้ของหุ่นยนต์ (Sensor)**

เซนเซอร์ที่ได้นำมาใช้ในการสร้างหุ่นยนต์ฮิวมานอยด์ UTHAI มีทั้งหมด 2 ชนิดคือ

1. Ground contact sensor
2. Inertial measurement unit

**Ground contact sensor**

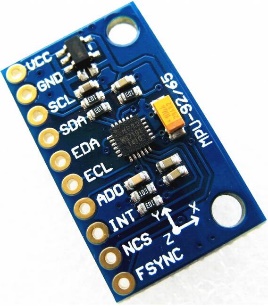
เซนเซอร์ตรวจหน้าสัมผัสที่พื้นเป็นเซนเซอร์ที่ถูกติดตั้งบริเวณฝ่าเท้า เพื่อตรวจสอบการเดินของหุ่นยนต์ฮิวมานอยด์ว่าขณะนี้มีการสัมผัสของฝ่าเท้าของหุ่นยนต์กับพื้นหรือไม่ ซึ่งในงานวิจัยนี้ได้ใช้หลักการของแสงสะท้อนนำมาประยุกต์ใช้ในการทำ Ground Contact Sensor



รูปที่ 3.7 หลักการทำงานของ Light Reflective Sensor

หลักการทำงานของอุปกรณ์นี้คือ การที่มีตัวปล่อยแสง (LED) ปล่อยแสงออกไป และมีตัวรับแสง (Phototransistor) รอวัดค่าความเข้มแสงที่สะท้อนกลับมาแปลงเป็นค่าความต้านทานของวงจร ดังนั้นแล้วถ้านำอุปกรณ์ตัวนี้มาใช้งานกับการตรวจสอบการเดินบนพื้นโดยตรง อาจจะมีปัญหาเรื่องสีที่ไม่เหมือนกันในแต่ละพื้นที่ และความสว่างของสิ่งแวดล้อมรอบข้าง ซึ่งอาจจะทำให้เกิดความผิดพลาดในการอ่านค่าได้ ดังนั้นแล้วจึงทำการแก้ไขโดยให้อุปกรณ์นี้อยู่ในระบบปิด กล่าวคือฝ่าเท้าของหุ่นยนต์จะมีวัสดุที่ทำจากยางยืดหยุ่นสีขาว ปกคลุมบริเวณที่อุปกรณ์ตรวจรับอยู่[[14]](#footnote-15)

เมื่อมีการเหยียบของฝ่าเท้าเกิดขึ้น ตัวยางที่ปกคลุมอยู่จะถูกบีบให้มีระยะห่างจากตัวเซนเซอร์น้อยลง และทำให้ค่าที่อ่านได้จากเซนเซอร์เปลี่ยนแปลงไป เนื่องจากความเข้มของแสงสะท้อนจะแปรผันตรงกับระยะห่างระหว่างเซนเซอร์กับแผ่นยาง โดยจะติดตั้งระบบเซนเซอร์ตัวนี้ให้มีจำนวน 3 ตัวต่อเท้า เพราะเพียงจุด 3 จุด ก็สามารถอธิบายระนาบของฝ่าเท้าที่สัมผัสกับพื้นได้ โดยเซนเซอร์จะติดบริเวณส้นเท้า 1 ตัวและปลายเท้า 2 ตัว เพื่อเพิ่มความแม่นยำในการตรวจสอบ

**Inertial Measurement Unit (IMU)**

รูปที่ 3.8 โมดูล IMU รุ่น MPU-9250

ในการทำวิจัยครั้งนี้ผู้จัดทำได้เลือกนำเซนเซอร์ MPU-9250มาใช้ในการอ่านค่ามุมเอียงของหุ่นยนต์เพื่อใช้ในการคุมเสถียรภาพของหุ่นยนต์ โดยเซนเซอร์ตัวนี้สามารถวัดค่าได้ 9 แกนคือ วัดค่าความเร็วเชิงมุม(gyroscope) 3 แกน วัดค่าความเร่งเชิงเส้น(accelerometer) 3 แกน และ วัดค่าสนามแม่เหล็กโลก(magnetometer) 3 แกน ซึ่งเซนเซอร์ตัวนี้จะติดตั้งบริเวณส่วนของลำตัวหุ่นยนต์ เนื่องจากว่าจะเป็นจุดที่สามารถบ่งบอกได้ถึงการเคลื่อนที่และมุมเอียงของหุ่นยนต์ในขณะนั้นได้ดีที่สุด[[15]](#footnote-16)

### 3.2.4 นำแบบจำลองของหุ่นยนต์ฮิวมานอยด์เข้าโปรแกรมจำลองการเคลื่อนไหว

ในส่วนนี้จะเป็นการนำ CAD model จากโปรแกรม Solidworks ไปใส่โปรแกรม Gazebo ผ่านเครื่องมือเสริมของ Solidworks

Solidworks to URDF exporter ช่วยเพิ่มความสะดวกสบายในการแปลง SW Parts และ Assemblies เป็น URDF การแปลงนั้นจะสร้าง ROS package ที่รวมไฟล์ meshes, textures และ robots (urdf file) สำหรับไฟล์ที่เป็นพาร์ทชิ้นเดียว จะสามารถใส่คุณสมบัติของวัสดุได้และสร้างพาร์ทนั้นเป็นลิ้งค์ใน URDF แต่สำหรับไฟล์ที่เป็นพาร์ทประกอบ จะสามารถสร้างลิ้งค์เป็นแบบต้นไม้ โดยเครื่องมือนี้จะสามารถใส่ชนิดของ Joint และแกนการหมุนให้ และนำไปใส่ใน Gazebo

การทำงานในส่วนนี้จะแบ่งงานออกเป็น 2 ส่วน ในส่วนแรกจะเป็นการทดสอบนำโมเดลอย่างง่ายเข้าไปในโปรแกรม อีกส่วนจะเป็นการนำโมเดลของจริงเข้าไปในโปรแกรม

### 3.2.5 จัดทำชิ้นส่วนโครงสร้างและประกอบ

ส่วนของโครงสร้างหุ่นยนต์ที่ผ่านการทดสอบในโปรแกรมจำลองแล้วจะถูกสร้างขึ้นจริง ด้วยเทคนิคการขึ้นรูปสามมิติในส่วนที่ไม่ต้องการความแข็งแรงมาก และการขึ้นรูปอลูมิเนียมในส่วนที่ต้องการความแข็งแรงสูง

### 3.2.6 พัฒนาซอฟแวร์ส่วนที่ติดต่อกับเซนเซอร์และตัวขับเคลื่อน

ระบบการทำงานส่วนหน่วยประมวลผลควบคุมระดับสูงที่ใช้บอร์ดคอมพิวเตอร์ Odroid-XU4 ได้ติดตั้ง ROS เอาไว้แล้ว ดังนั้น การพัฒนาโปรแกรมจะอยู่ในรูปแบบของ Node ที่มีการ Public และ Subscribe มีการให้หน่วยประมวลผลควบคุมระดับต่ำเป็นอีก Node หนึ่งในระบบโปรแกรม

### 3.2.7 จัดทำคู่มือและเอกสารการใช้งาน

คู่มือจะเป็นส่วนที่ผู้มาพัฒนาต่อยอดสามารถที่จะอ่านทำความเข้าใจได้ โดยจะเขียนให้อยู่ในรูปของไฟล์ Markdown (.md) และเก็บเอาไว้ในเว็บไซท์ GitHub ซึ่งเป็นแหล่งรวม Source code ออนไลน์ สามารถเข้าไปดาวน์โหลดไฟล์ลงเครื่องผู้ใช้ แล้วทำการติดตั้งใช้งานได้เลย อีกทั้งผู้ใช้ยังสามารถส่ง Code ของตัวเองเข้าระบบเพื่อเพิ่มประสิทธิภาพในการทำงานของโปรแกรม

ส่วนที่เน้นในการทำคู่มือคือ

1. รายการวัสดุที่ใช้ในการทำหุ่นยนต์ฮิวมานอยด์ UTHAI
2. รายระเอียดการเชื่อมต่อระหว่างอุปกรณ์ที่อยู่ในตัวหุ่นยนต์
3. รายระเอียดการประกอบชิ้นส่วนทางกล
4. รายละเอียดการใช้งานโปรแกรมพื้นฐาน

# บทที่ 4 ผลการดำเนินงาน

## รายการอุปกรณ์ที่ได้จัดซื้อ

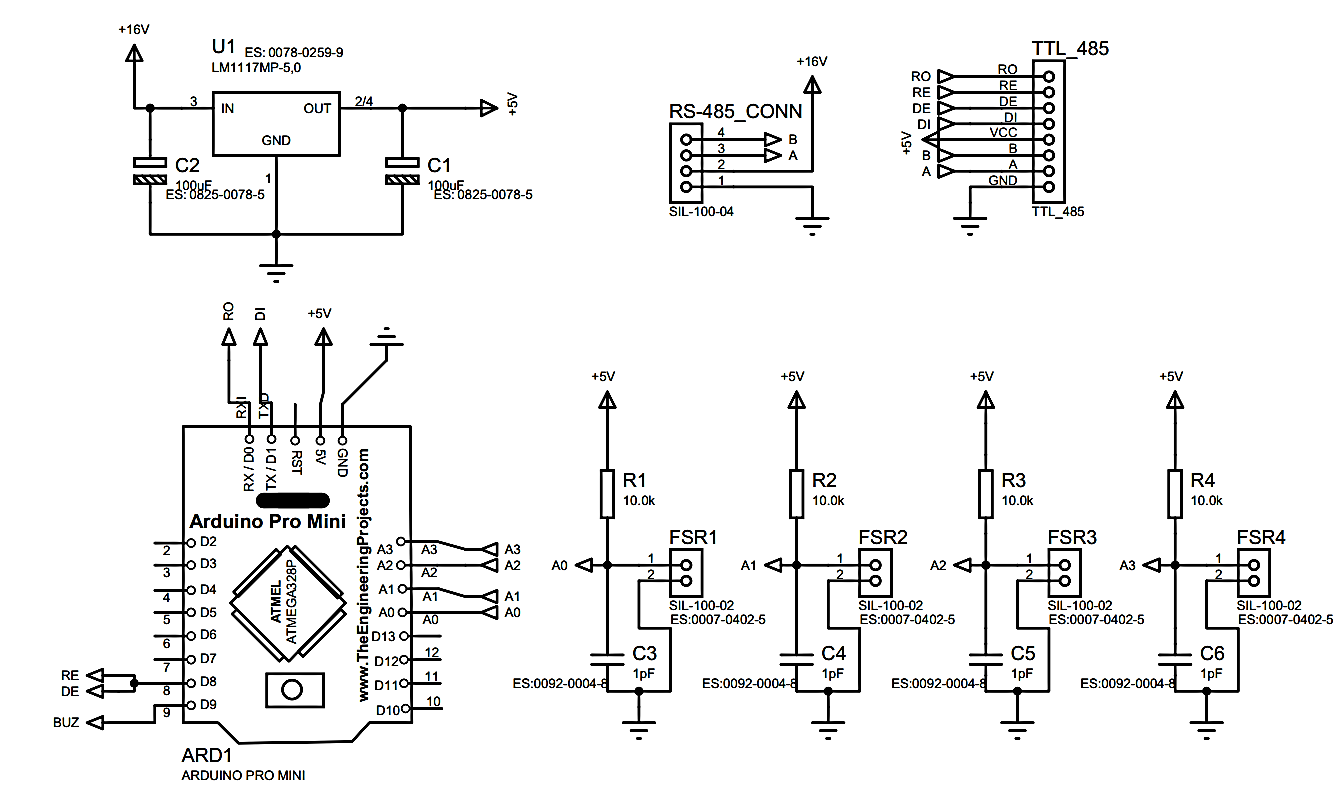
ตารางที่ 4.1 ตารางแสดงรายการของวัสดุต่าง ๆ

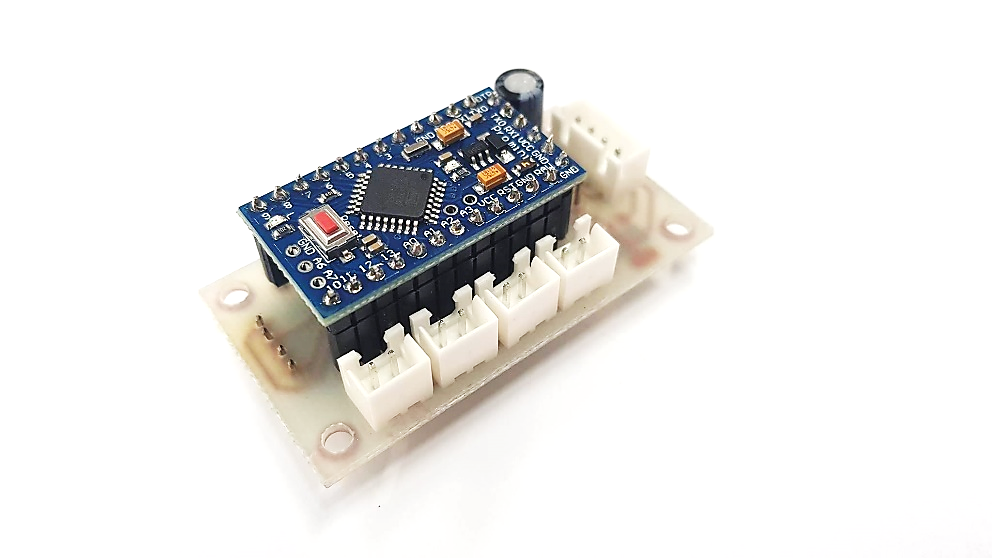
|  |  |  |  |
| --- | --- | --- | --- |
| รายการ | จำนวน(หน่วย) | ราคา/หน่วย(บาท) | ราคารวม(บาท) |
| Processing Unit |  |  |  |
| * Odroid XU4 Embeded Computer | 1 | 3,800 | 3,800 |
| * Shifter Shield for Odoird XU4 | 1 | 1,000 | 1,000 |
| Sensor |  |  |  |
| * Force sensitive Resistor | 8 | 300 | 2,400 |
| * Electronic Component | 1 | 2,000 | 2,000 |
| * MPU9255 9 Axis IMU Module | 1 | 500 | 500 |
| โครงสร้าง |  |  |  |
| * ค่าจัดทำชิ้นงาน CNC, Laser Cutting | 1 | 8,000 | 8,000 |
| * อุปกรณ์ส่งกำลัง | 1 | 3,000 | 3,000 |
| * ค่าวัสดุ เช่น Filament 3D printer , Carbon Fiber | 1 | 8,000 | 7,000 |
| * สปริง | 14 | 50 | 700 |
| * อุปกรณ์สิ้นเปลือง เช่น กระดาษทราย ฯลฯ | 1 | 1,000 | 1,000 |
| อุปกรณ์เสริม Motor Dynamixel |  |  |  |
| * Frame สำหรับต่อพ่วงมอเตอร์ | 4 | 2,000 | 8,000 |
| * Horn Bearing | 4 | 1,400 | 5,600 |
| อุปกรณ์จ่ายพลังงาน |  |  |  |
| * Power Supply | 1 | 2,000 | 2,000 |
| * Battery Li-Po 4 cell | 1 | 3,000 | 3,000 |
| รวม | | | 48,000 |

\*\* ใช้สำหรับแจกแจงค่าใช้จ่ายเบื้องต้นเท่านั้น ไม่สามารถใช้อ้างอิงงบประมาณแบบละเอียดได้

## เซนเซอร์ตรวจจับการสัมผัสพื้น

แนวคิดการออกแบบหลัก คือการออกแบบให้สามารถติดตั้งกับตัวหุ่นยนต์ได้เลย ไม่ต้องเชื่อมต่อสายไฟและสายส่งข้อมูลใหม่ โดยให้ใช้สายไฟไฟเลี้ยง และสายสัญญาณชุดเดียวกับตัวขับเคลื่อน Dynamixel Servo Motor ซึ่งมีการติดต่อกันในลักษณะเป็นบัสแบบ RS-485 ดังนั้นแล้วผู้เขียนจึงเลือกที่จะทำโมดูลขึ้นมาใหม่ 1 โมดูล เพื่อที่ใช้ในการอ่านค่า Ground Contact Sensor ของหุ่นยนต์โดยเฉพาะ โดยมีการติดต่อรูปแบบบัส RS-485 ใช้ลักษณะการติดต่อสื่อสาร(Protocol) เดียวกับตัวขับเคลื่อน Dynamixel และมีการพัฒนาจาก Arduino ซึ่งทำให้สามารถอ่านค่าได้ทั้ง Analog และ Digital ได้ อีกทั้งรองรับการต่อ Sensor แบบ Force sensitive resistor จำนวน 4 ตัว



รูปที่ 4.1 Schematic ของวงจร Ground Contact Sensor

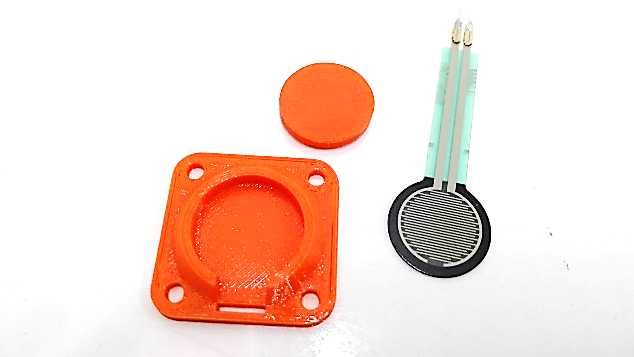
รูปที่ 4.2 แผงวงจร Ground Contact Sensor ที่ประกอบเสร็จแล้ว

เซนเซอร์ที่เลือกใช้คือ Force Sensitive Resistor (FSR) เป็นเซนเซอร์ที่มีค่าความต้านทานภายในตัวเอง โดยเซนเซอร์นี้มีหลักการทำงานคือ ค่าความต้านทานทางไฟฟ้าของตัวเซนเซอร์จะเปลี่ยนแปลงเมื่อมีแรงเข้ามากระทำกับหน้าสัมผัส เมื่อมีแรงเข้ามากระทำมาก จะทำให้ค่าความต้านทานต่ำ หากไม่มีแรงเข้ามากระทำจะทำให้มีค่าความต้านทานสูง และเมื่อมีการนำเซนเซอร์นี้มาต่อกับตัวต้านทานที่มีค่าคงที่ ในรูปแบบของ Voltage Divider จะทำให้สามารถอ่านค่าแรงดันไฟฟ้าที่เปลี่ยนแปลงตามแรงที่เกิดขึ้นกับหน้าสัมผัสของเซนเซอร์ FSR ได้



รูปที่ 4.3 Force Sensitive Resistor (FSR) ขนาด 0.5 นิ้ว

ข้อดีของ FSR นั้นคือ เป็นเซนเซอร์ที่ถูกพัฒนาและออกแบบมาเพื่อใช้สำหรับการวัดแรงโดยตรง จึงทำให้ใช้งานได้ง่าย และสะดวก ในราคาที่ถูกกว่า เมื่อเทียบกับเซนเซอร์ Load cell ที่มีราคาสูงและการใช้งานจำเป็นที่จะต้องมีวงจรขยายสัญญาณที่ใช้ในการอ่านค่าการบิดของวัสดุจาก แต่ FSR นั้นก็มีข้อเสียเช่นกันคือ ความไม่ทนทานต่อการขีดข่วน เนื่องจากตัวเซนเซอร์ถูกทำมาจากฟิล์มพลาสติกบางๆ ซึ่งหากเกิดการขีดข่วนเกิดขึ้นแล้วอาจทำให้ฟิล์มฉีกขาดได้ หากฟิล์มขาดจะทำให้ค่าความต้านทานออกมาไม่เหมือนเดิม ดังนั้นทางผู้เขียนจึงเลือกที่จะออกแบบโครงครอบสำหรับเซนเซอร์ FSR เพื่อป้องกันจากการถูกขีดข่วนจากภายนอก

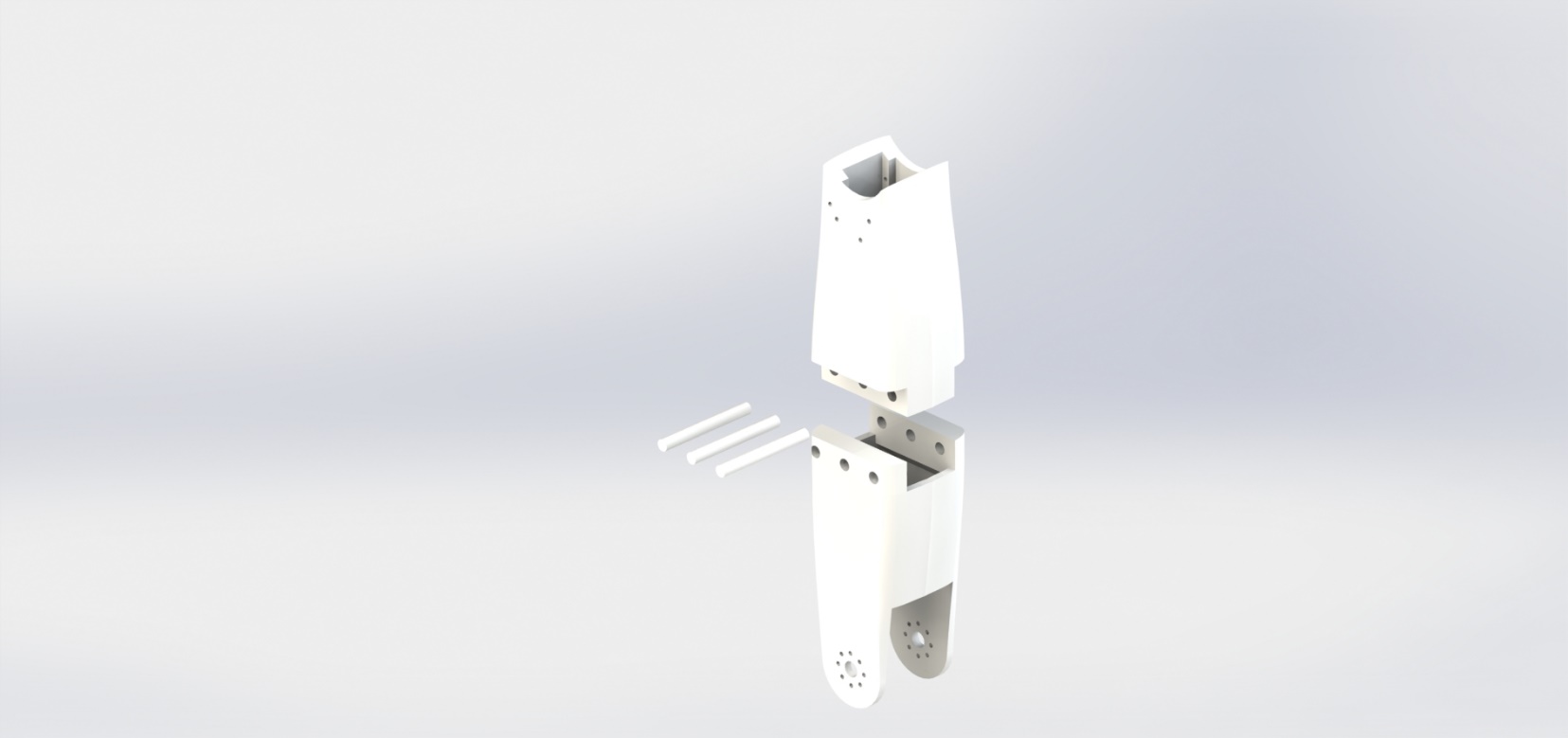
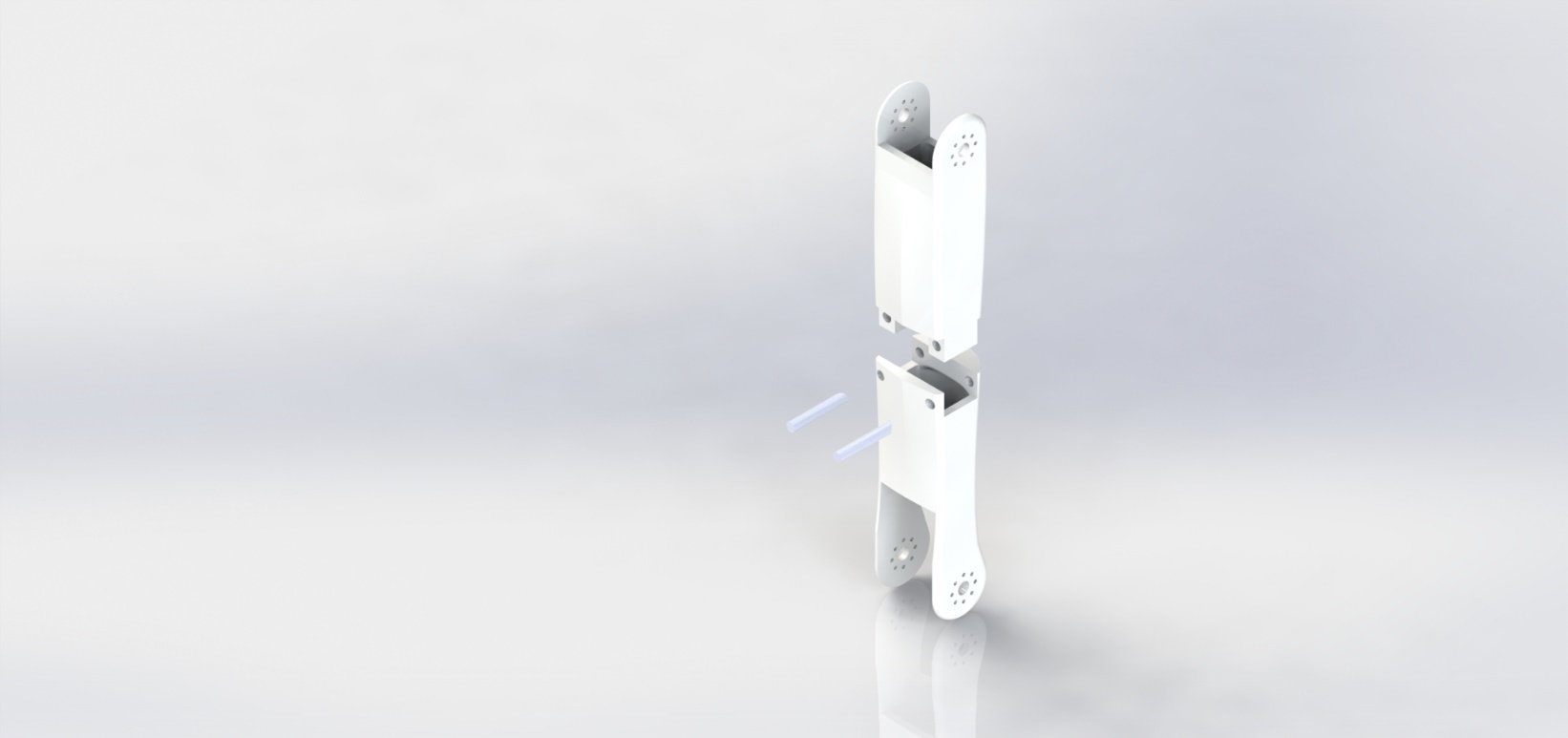


รูปที่ 4.4 โครงครอบของ Force Sensitive Resistor

อย่างไรก็ตาม ถึงแม้ว่าค่าความต้านทานของตัวเซนเซอร์จะสามารถเปลี่ยนแปลงไปตามแรงที่เข้ามากระทำได้ แต่ก็ไม่สามารถวัดค่าแรงที่มากระทำได้อย่างเที่ยงตรง จะสามารถวัดได้แค่ มีแรงเข้ามากระทำมากหรือน้อยเท่านั้น ซึ่งถือว่าเพียงพอสำหรับการตรวจเช็คว่าฝ่าเท้าติดอยู่กับพื้นหรือไม่

## การออกแบบโครงสร้างส่วนขา

การออกแบบโครงสร้างส่วนขาของหุ่นยนต์ฮิวมานอยด์ ผู้เขียนได้ออกแบบโดยคำนึงถึงการขึ้นรูปด้วยเครื่องพิมพ์สามมิติ (3D Printer) ทั่วไปก็สามารถที่จะขึ้นรูปชิ้นงานส่วนนี้ได้ ดังนั้นผู้เขียนจึงเลือกที่จะแบ่งส่วนของขาออกเป็นจำนวน 2 ส่วน ในก้านต่อของขาท่อนบนและขาท่อนล่าง หลังจากนั้นนำมาตอกสลักเพื่อยึดติดชิ้นส่วนเข้าด้วยกัน



รูปที่ 4.5 โครงสร้างส่วนขาท่อนบนและขาท่อนล่าง

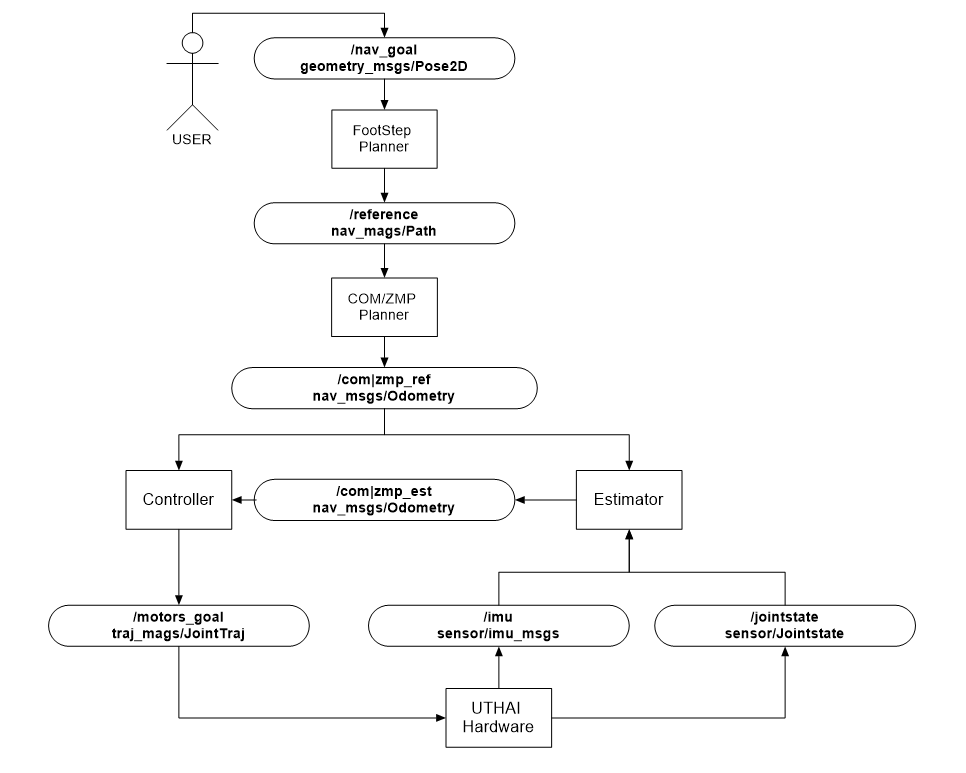
ทดสอบโครงสร้างและการขับเคลื่อน

จากการทดสอบความสามารถในการเคลื่อนที่ พบว่าตัวขับเคลื่อนสามารถเคลื่อนที่เข้าตำแหน่งได้ถูกต้องตามมุม ที่ป้อนเข้าไปให้ระบบ แต่หากทำให้ชิ้นส่วนของขาเคลื่อนที่ด้วยความเร็วไปกลับๆด้วยความเร็วที่มาก อาจจะทำให้ตัวขับเคลื่อนเกิดการโอเวอร์โหลด ซึ่งมีผลทำตัวขับเคลื่อนหยุดการทำงาน ต้องทำการปิดเปิดตัวขับเคลื่อนใหม่

จากการทดสอบความแข็งแรงของชิ้นงาน พบว่าชิ้นงานมีความแข็งแรงเพียงพอที่จะทำให้ตัวขับเคลื่อนมีค่าแรงบิดเป็นค่าแรงบิดสูงสุด(Stall Torque) แล้วทำให้ชิ้นงานไม่เกิดความเสียหาย

จากการทดสอบระยะเวลาการทำงานของตัวขับเคลื่อน ด้วยการเขียนโปรแกรมให้ตัวขับเคลื่อน เคลื่อนที่ไปมา สลับตำแหน่งไปเรื่อยๆอย่างต่อเนื่อง เป็นเวลา 20 นาที พบว่า ตัวขับเคลื่อนทำงานได้เป็นปกติ

## การติดต่อสื่อสารระหว่าง Node ใน ROS



รูปที่ 4.6 โครงสร้างการติดต่อสื่อสารระหว่าง Node ใน ROS

|  |  |
| --- | --- |
| geometry\_msgs/Pose2D | |
| float64 | x |
| float64 | y |
| float64 | theta |

ใช้สำหรับในการส่งตำแหน่งที่หุ่นยนต์ต้องการจะเดินไปเป็นในระบบ 2 มิติ

|  |  |
| --- | --- |
| nav\_msgs/Path | |
| Header | header |
| geometry\_msgs/PoseStamped[] | poses |

ใช้สำหรับส่งตำแหน่งของฝ่าเท้าของหุ่นยนต์ให้ไปในตำแหน่งที่โปรแกรมสร้างขึ้น

ใน nav\_msgs/Path ประกอบไปด้วยลิสของ geometry\_msgs/Pose

|  |  |
| --- | --- |
| nav\_msgs/Odometry | |
| std\_msgs/Header | header |
| string | child\_frame\_id |
| geometry\_msgs/PoseWithCovariance | pose |
| geometry\_msgs/PoseWithCovariance | twist |

ใช้สำหรับการส่งตำแหน่งและความเร็วของ CoM และ ZMP ของหุ่นยนต์เพื่อใช้ควบคุม

|  |  |
| --- | --- |
| trajectory\_msgs/JointTrajectory | |
| std\_msgs/Header | header |
| string[] | joint\_names |
| trajectory\_msgs/JointTrajtoryPoint[] | points |

ใช้ในการส่งค่าตำแหน่ง ความเร็ว หรือแรงบิดให้กับมอเตอร์เพื่อใช้ในการควบคุมข้อต่อ

|  |  |
| --- | --- |
| sensor\_msgs/Imu | |
| std\_msgs/Header | header |
| geometry\_msgs/Quaternion | orientation |
| float64[9] | orientation\_covariance |
| geometry\_msgs/Vector3 | angular\_velocity |
| float64[9] | angular\_velocity\_covariance |
| geometry\_msgs/Vector3 | linear\_acceleration |
| float64[9] | linear\_acceleration\_covariance |

ใช้ในการอ่านค่าเซนเซอร์ IMU จากหุ่นยนต์แล้วส่งให้ Estimater Node

|  |  |
| --- | --- |
| sensor\_msgs/JointState | |
| std\_msgs/Header | header |
| string[] | name |
| float64[] | position |
| float64[] | velocity |
| float64[] | effort |

ใช้สำหรับอ่านค่าตำแหน่ง ความเร็ว แรง ของตัวขับเคลื่อนแล้วส่งให้ Estimater Node

## การขึ้นสร้างแบบจำลองสามมิติด้วย ROS [URDF]

หลังจากที่ได้ออกแบบและโมเดลหุ่นยนต์เดินสองขาขึ้นมาแล้วโดยที่ใช้โปรแกรมเครื่องมือ CAD ต่างๆ เช่น AutoCAD, SolidWorks, Blender หรืออื่นๆ เราจะนำโมเดลที่สร้างขึ้นนี้มาใช้ในการทำระบบจำลองเสมือนจริง(Simulation) การที่เราทำ Simulation นั้นก็เพื่อที่จะทำให้เราสามารถมองเห็นโครงสร้างของหุ่นยนต์และเห็นการทำงานต่างๆของหุ่นยนต์ ก่อนที่เราจะสร้างมันขึ้นมาจริงๆ

โมเดลหุ่นยนต์จำลองที่เราสร้างขึ้นมานั้นควรที่จะมีลักษณะใกล้เคียงกับหุ่นยนต์ของจริงมากที่สุด ไม่ว่าจะเป็นรูปร่าง รูปทรง น้ำหนักต่างๆ

**ROS packages for robot modeling**

ROS นั้นได้ให้เครื่องมือที่ช่วยให้เราสามารถสร้าง 3D robot models ได้

* + robot\_model: ใน ROS มี meta package ที่ชื่อว่า robot\_model ซึ่งข้างในมี package ต่างๆที่ใช้สำหรับสร้าง 3D robot models
    - urdf: เป็น 1 ในหลายๆ package ที่อยู่ใน robot\_model, urdf เป็น xml ไฟล์ที่เอาไว้ใช้บอกลักษณะของหุ่นยนต์ ย่อมาจาก Unified Robot Description Format(URDF)
  + เราสามารถระบุ robot model, sensors และ working environment โดยใช้ URDF การบอกนั้นจะสามารถบอกเป็นเหมือน tree structure ของ link ต่างๆในตัวหุ่นยนต์ สามารถบอก rigid link เชื่อมต่อกันผ่าน joints แต่ถ้าเป็น flexible link จะไม่สามารถบอกได้โดยใช้ urdf
    - joint\_state\_publisher: เครื่องมือนี้มีประโยชน์มากในการ model robot URDF เพราะมันสามารถหา joints ทุก joint ที่ไม่ใช่ fixed joints มาแสดงเป็น GUI sliders ทำให้เราสามารถเลื่อนๆหมุนๆไปมาได้ อีกทั้งยังสามารถใช้งานร่วมกับ visualize RViz
    - robot\_state\_publisher: เป็นเครื่องมือที่ใช้ในการ publish 3d pose ของ link ต่างๆใน urdf การ ยublish นั้นจะใช้ ROS tf(transform) ROStf คือการหาความสัมพันธ์ระหว่าง frame ของหุ่นยนต์
    - xacro: ย่อมาจาก XML Macros หรือเราสามารถเรียกอีกอย่างว่า URDF plus add-ons. ซึ่งการทำงานเหมือนกับ urdf แต่ทำให้ไฟล์ urdf สั้นกว่า อ่านง่ายกว่า และสามารถใช้เพื่อทำให้สร้างหุ่นยนต์ที่มีความซับซ้อนง่ายขึ้น เราสามารถแปลงไฟล์ xacro เป็น urdf ได้

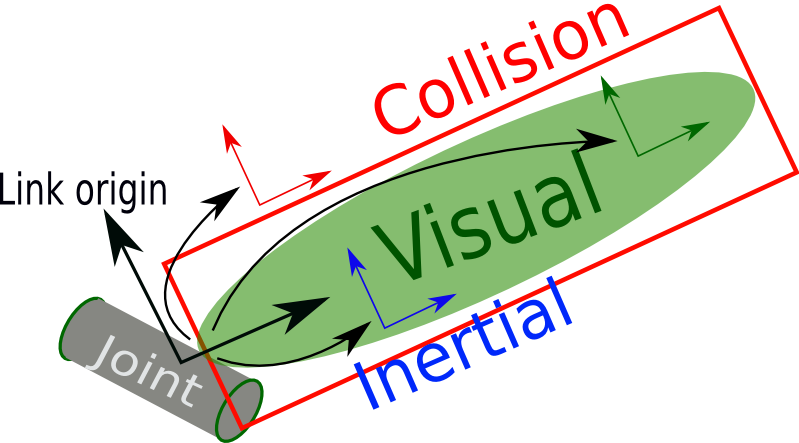
**Understanding robot modeling [URDF]**

จากที่กล่าวไปแล้วเกี่ยวกับไฟล์ URDF ที่จะช่วยสร้างแบบจำลองโมเดลหุ่นยนต์แต่ไฟล์นี้เราจำเป็นที่จะต้องสร้างขึ้นมาก่อน

ไฟล์ URDF สามารถใส่ kinematic dynamic ของหุ่นยนต์ได้ visual collision

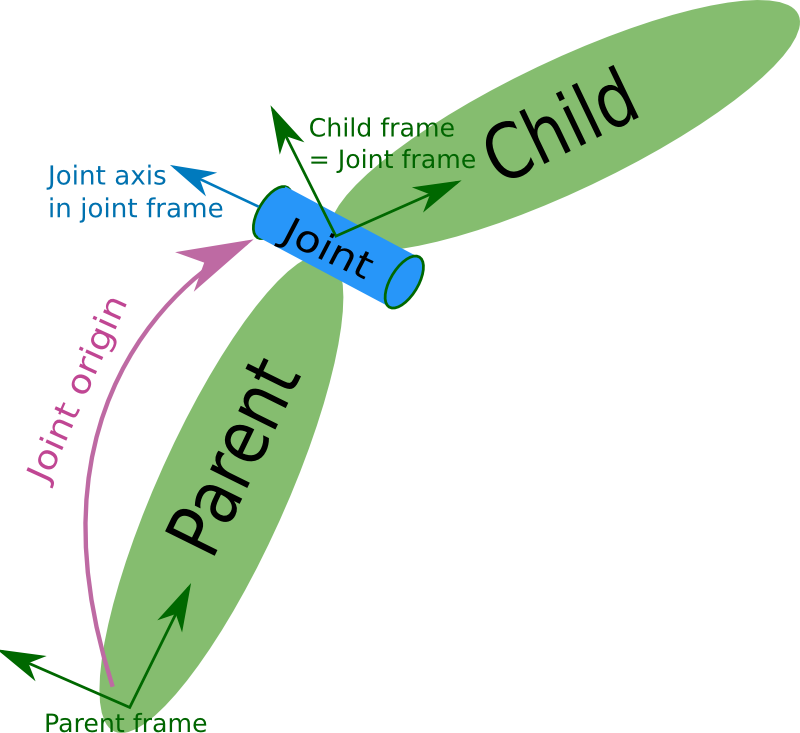
ส่วนประกอบของ tags แต่ละอันใน urdf robot model

link: ลิ้งจะแสดงถึง single link ของหุ่นยนต์ สามารถใช้ tag นี้ได้ อีกทั้งเรายังสามารถใส่ properties ได้อีกด้วย เช่น size shape color หรือเรายังสามารถใส่ 3d mesh ลงไปใน link ได้ สามารถใส่ dynamic properties เช่น inertial matrix และ collision properties



ตัว code ด้านบนจะบอกเพียงแค่ 1 link เท่านั้น มีส่วนของ visual ที่เป็น real link ของหุ่นยนต์จริงๆ ส่วนที่รอบๆจะเป็นส่วน Collision ควรจะทำให้ใหญ่กว่านิดหน่อยเพราะจะได้ตรวจพบได้ก่อนที่จะเกิดการชนกันจริงๆ

joint: ข้อต่อ จะแสดงถึง joint ของหุ่นยนต์ สามารถกำหนด kinematics dynamics ของ joint ได้ เช่น ใส่ limit ไม่ให้ joint เคลื่อนที่เกิน กำหนดความเร็วของ joint ได้ ชนิดของ joint ใน urdf มีอยู่ทั้งหมด 6 ชนิด คือ revolute, continous, prismatic, fixed, floating และ planar



URDF joint จะเป็นการเปลี่ยนของ 2 link ซึ่งลิ้งแรกจะเรียกว่า Parrent link ส่วนอีกอันจะเรียก Child link

# เอกสารอ้างอิง

|  |  |
| --- | --- |
| [1] | D. Seward, B. A. และ M. F., “The anatomy of a humanoid robot,” *Robotica,* pp. 437-443, 1995. |
| [2] | C. Antoine Petit, “Poppy-Project,” Inria, 2016. [ออนไลน์]. Available: https://www.poppy-project.org. [%1 ที่เข้าถึง9 2017]. |
| [3] | t. R. Consortium, “iCub.org - an opensource cognitive humanoid robotic platform,” the RobotCub Consortium, 2017. [ออนไลน์]. Available: http://www.icub.org/. [%1 ที่เข้าถึง9 2017]. |
| [4] | “DARwIn-OP,” ROBOTIS, 2010. [ออนไลน์]. Available: http://support.robotis.com/en/product/darwin-op.htm. [%1 ที่เข้าถึง9 2017]. |
| [5] | “Discover Nao,” SoftBank Robotics, [ออนไลน์]. Available: https://www.ald.softbankrobotics.com/en/robots/nao. |
| [6] | “Humanoid History - WABOT-,” Waseda University, 1970. [ออนไลน์]. Available: http://www.humanoid.waseda.ac.jp/booklet/kato\_2.html. [%1 ที่เข้าถึง9 2017]. |
| [7] | B. S. Ng, “Robo-Erectus Tr-2010 TeenSize Team Description Paper.,” 2010. |
| [8] | Y. Hayashibara, “CIT Brains (Kid Size League),” 2015. |
| [9] | DARPA, “DARPA Robotics Challenge (DRC),” June 2015. [ออนไลน์]. Available: www.darpa.mil. |
| [10] | C. a. C. o. I. Antoine Petit, “Poppy-Project,” Inria, 2016. [ออนไลน์]. Available: https://www.poppy-project.org/. [%1 ที่เข้าถึง25 9 2017]. |

**ภาคผนวก**

**ติดตั้ง Ubuntu ลงบน Odroid board**

1. Download Odroid images จากลิ้งค์ข้างล่างนี้

Download Odroid-XU4:

<https://odroid.in/ubuntu_16.04lts/ubuntu-16.04.3-4.9-mate-odroid-xu4-20170824.img.xz>

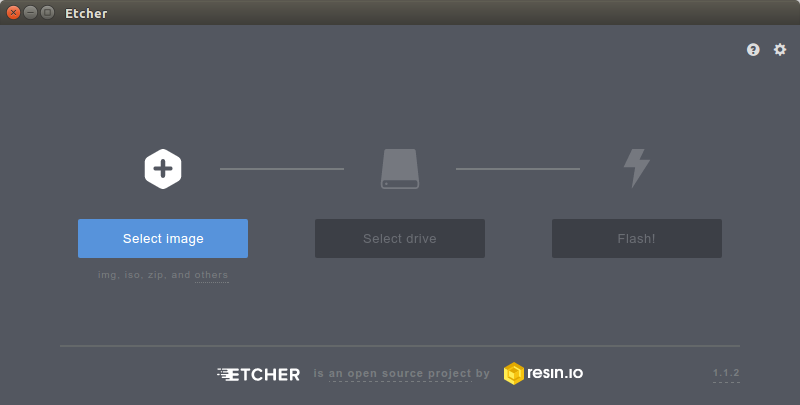
2. Download Burn images to SD card จากลิ้งค์เว็บข้างล่างนี้

Download Etcher:

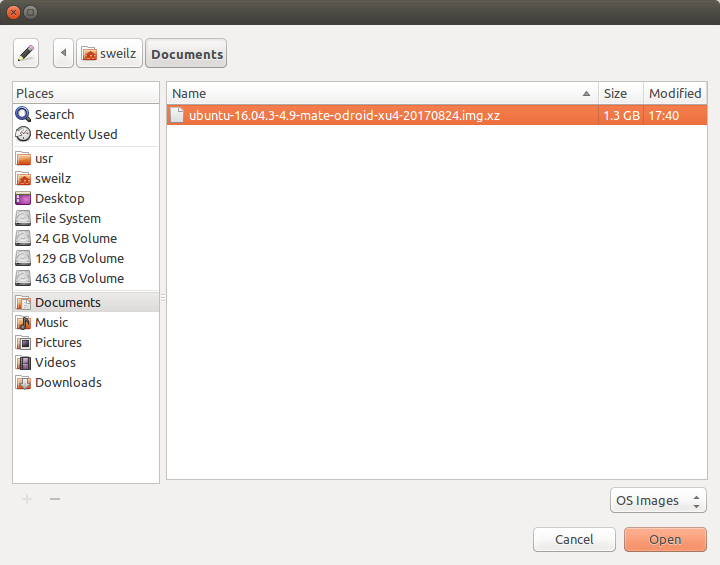
<https://github.com/resin-io/etcher/releases/download/v1.1.2/>

3. ใส่ SDcard เข้าเครื่องคอมพิวเตอร์

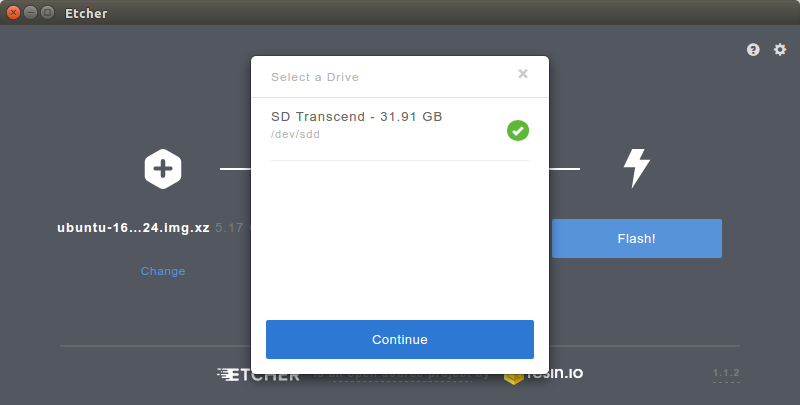
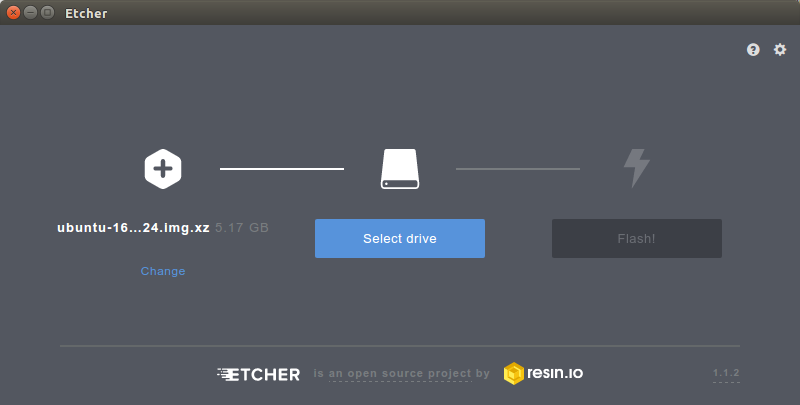
4. เปิดโปรแกรม Etcher ขึ้นมา กดเลือก Select image



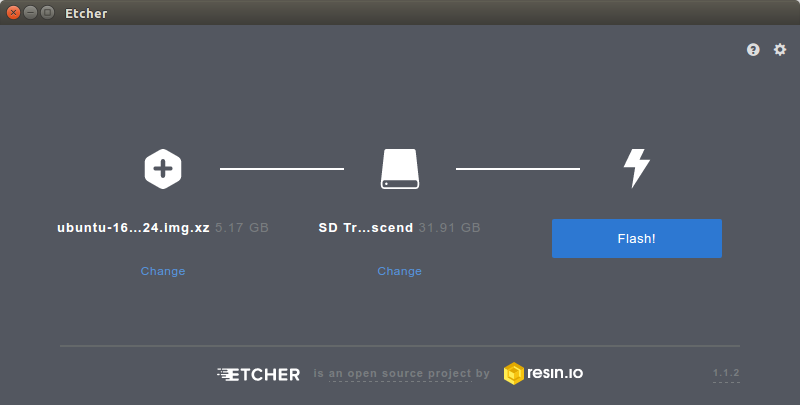
เลือกไฟล์ Image ที่ต้องการลงใน Odroid



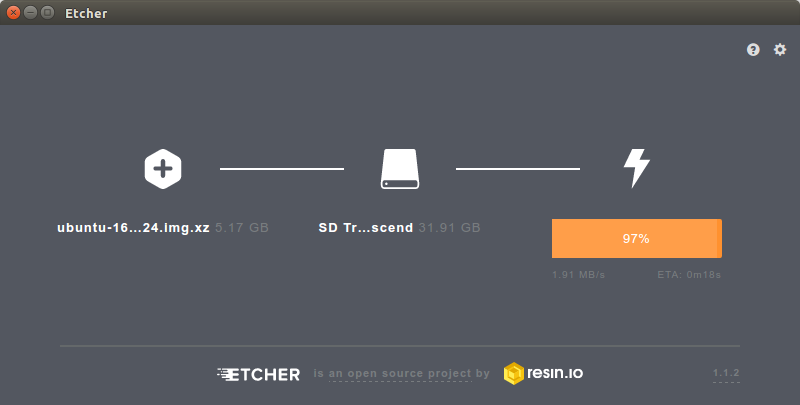
กด Select drive แล้วเลือก SDcard ที่ต้องการ แล้วกด Continue



กด Flash แล้วรอให้โปรแกรม Burn เสร็จ

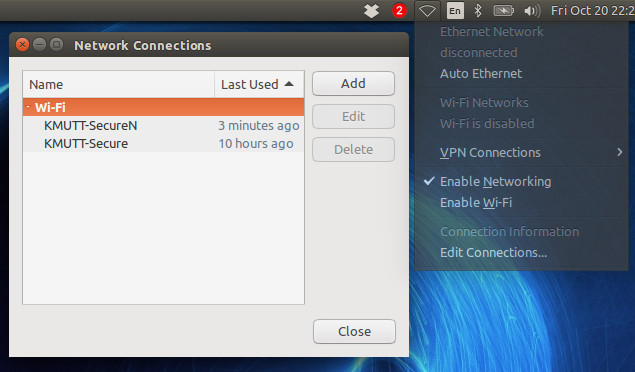


ใช้ระยะเวลาประมาณ 30 นาที



เชื่อมต่อ Odroid กับ PC

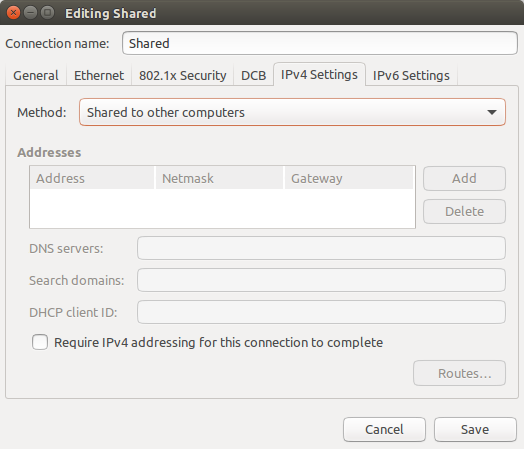
1. คลิกที่ Edit Connection... ตรง Network Option ตามรูป หลังจากนั้นคลิก Add เพื่อที่จะสร้าง Connection อันใหม่ขึ้นมา



เลือกเป็น Ethernet แล้วกด Create...



2. Connection name ตั้งชื่อว่า Shared และกดไปที่ Tab IPv4 Setting เลือก Method เป็น Shared to other computers



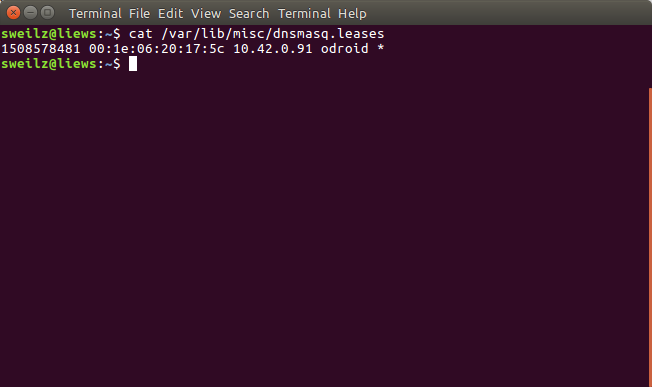
3. หลังจากเสร็จแล้วก็ในนำ SDcard ใส่เข้าไปใน Odroid เสียบสาย power 5V แล้วต่อสาย LAN เข้ากับ PC

4. เมื่อ Board boots up ขึ้นมาแล้วมันจะเชื่อมต่อ Share Network โดยอัตโนมัติถ้าหากเชื่อมต่อแล้ว หมายความว่าตอนนี้ Board Odroid นั้นจะมี IP Address เป็นของตัวเอง

5. ถึงตอนนี้เราจะต้องหาว่า IP address ของ Odroid เป็นอะไรถึงจะเข้าไปควบคุมได้ ซึ่งคำสั่งที่จะหาก็คือ

$ cat /var/lib/misc/dnsmasq.leases

จะได้ค่าตามแบบที่เห็นนี้

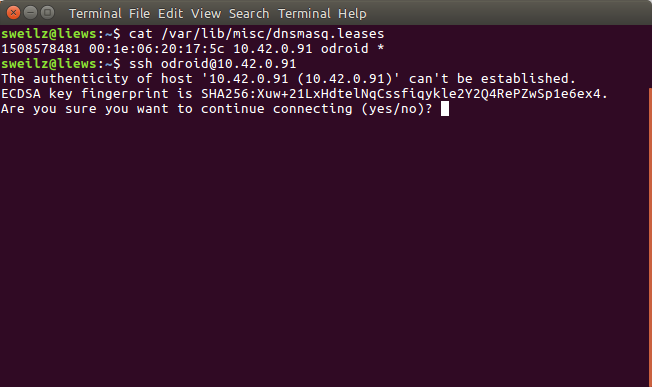


6. หลังจากได้ IP address มาแล้วก็แสดงว่าเราสามารถที่จะติดต่อกับ Odroid ผ่าน SSH ได้

$ ssh odroid@%ip*\_address\_*ของ\_odroid%

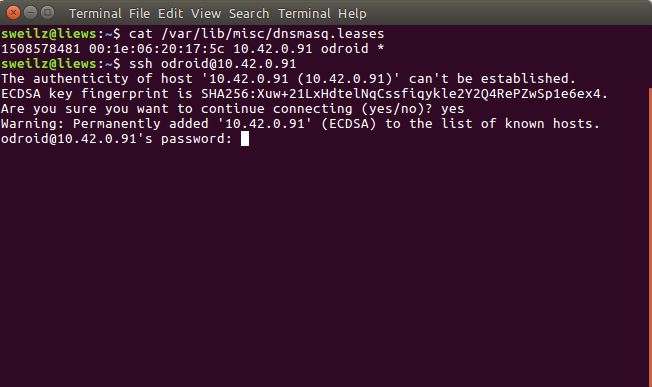
ในที่นี้คือ 10.42.0.91 จะใช้คำสั่งเป็น

$ ssh odroid@10.42.0.91

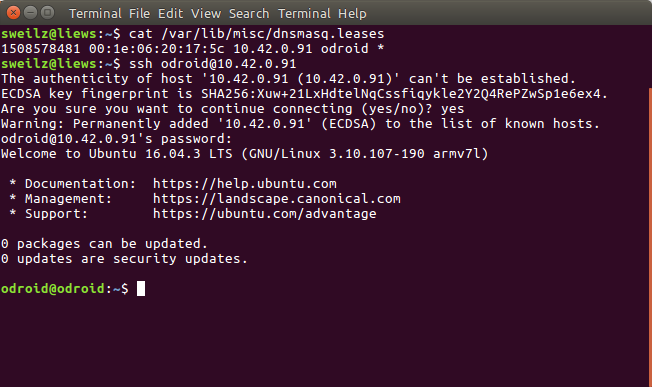


ระบบต้องการที่จะเพิ่ม Key Access เข้าระบบให้พิมพ์ `yes` หลังจากนั้นระบบจะขอ password

Password Default จะเป็น `odroid`



7. ถ้าทุกอย่างถูกต้อง เราก็จะสามารถเข้าถึงและควบคุม Odroid ผ่านทาง SSH ได้



8. เมื่อผ่านแล้วก็ให้ทำการอัพเดตระบบ โดยการพิมพ์คำสั่งตามนี้เข้าไปทีละคำสั่ง

$ sudo apt update && apt -y upgrade

ลง ROS บน Odroid

1. Download ROS Installer

$ wget https://raw.githubusercontent.com/SweiLz/UTHAI-Humanoid/master/ros\_installer.sh

2. Change permission

$ chmod +x ros\_installer.sh

3. Run installer

$ sh ros\_installer.sh

ทดลองใช้คำสั่ง `roscore` หากเห็นไม่มีปัญหาก็แสดงว่าสามารถใช้งาน ROS ได้แล้ว

**เริ่มเขียนโปรแกรมโดยใช้ STM32 board**

1. สมัครสมาชิกที่เว็บไซต์ mbed <https://os.mbed.com/>

2. หลังจากสมัครสมาชิกเสร็จแล้ว เข้าไปดู board ที่ระบบรองรับ <https://os.mbed.com/platforms/>

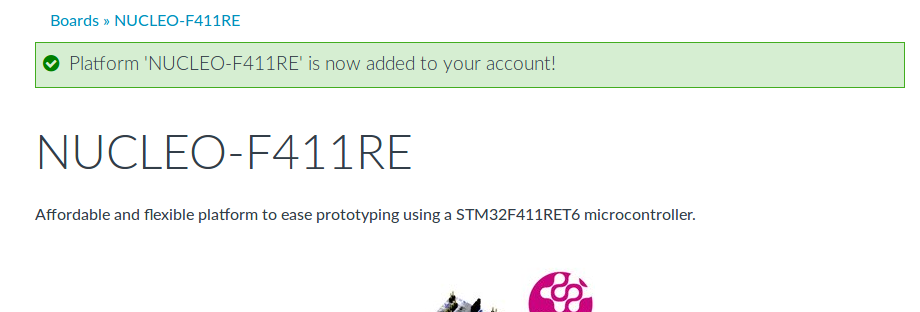
3. เราจะเลือก board ของเราได้จากหน้านี้ ในที่นี้เราจะเลือก NUCLEO F411RE board



หรือเข้าไปที่หน้านี้ <https://os.mbed.com/platforms/ST-Nucleo-F411RE/>



4. เราจะเห็นปุ่ม Add to your mbed compiler อยู่ตรงทางขวาของเพจ ให้กดปุ่มเพื่อที่จะเพิ่ม board นี้เข้า mbed compiler

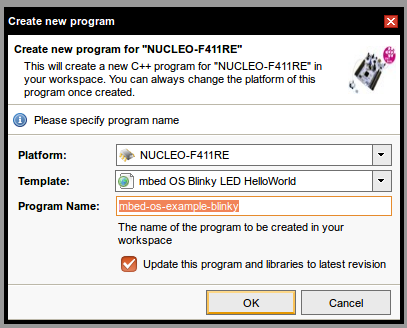


5. หลังจากเพิ่ม board เข้า compiler แล้ว เราจะเขียนโปรแกรมได้โดยการเข้าไปที่ mbed compiler

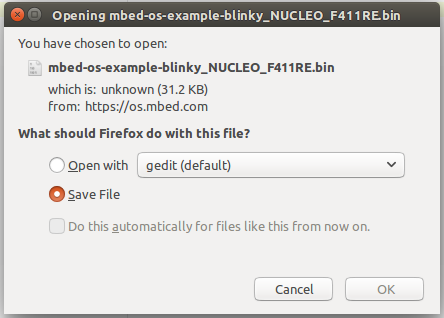
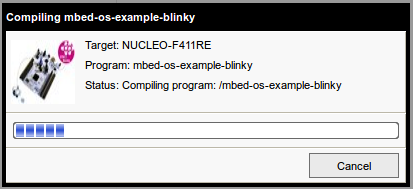
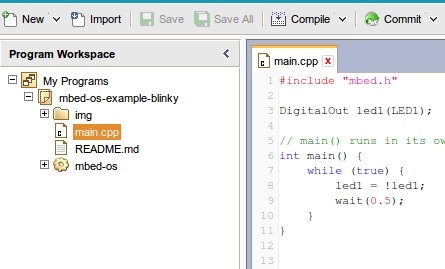
หรือเข้าไปที่ <https://os.mbed.com/compiler/>



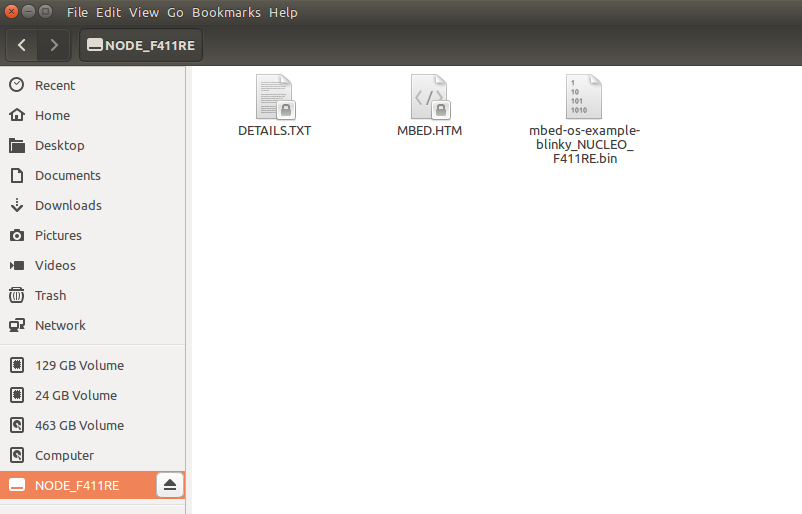
6. กดสร้างโปรแกรมใหม่ขึ้นมา เพื่อทดสอบการทำงานของ board โดยใช้โปรแกรมตัวอย่าง mbed blinkly



เปิดไฟล์ maim.cpp แล้วกด compile

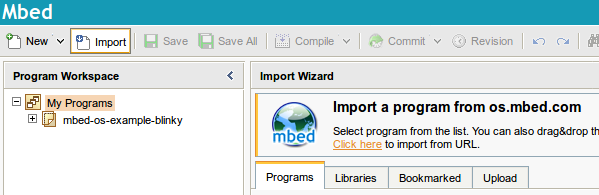


หลังจากนั้นจะได้ไฟล์ binary (.bin) ซึ่งสามารถนำไปใส่ไว้ใน flash drive ของ board

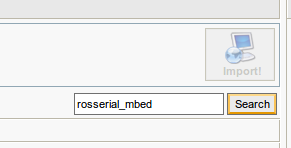


ถ้าเห็นไฟสีเหลืองกระพริบๆ แสดงว่า STM32 ของเราพร้อมสำหรับการทำ ROS Node แล้ว

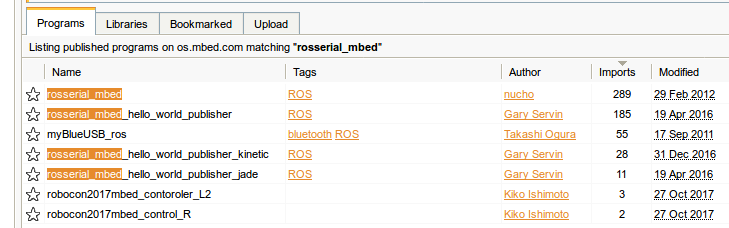
**เริ่มเขียนโปรแกรม STM32 board ให้เป็น ROS Node**  
1. เปิด mbed compiler ขึ้นมาหรือเข้าไปที่ <https://os.mbed.com/compiler/> แล้วกด Import



2. ทางช่อง Search ขวามือ ให้พิมพ์คำว่า rosserial\_mbed ลงไป แล้วกด Search

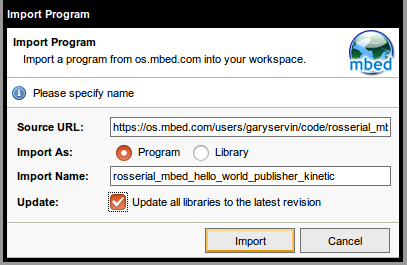


จากนั้นรายชื่อ ให้กดที่ rosserial\_mbed\_hello\_world\_publisher\_kinetic ของ Gary Servin

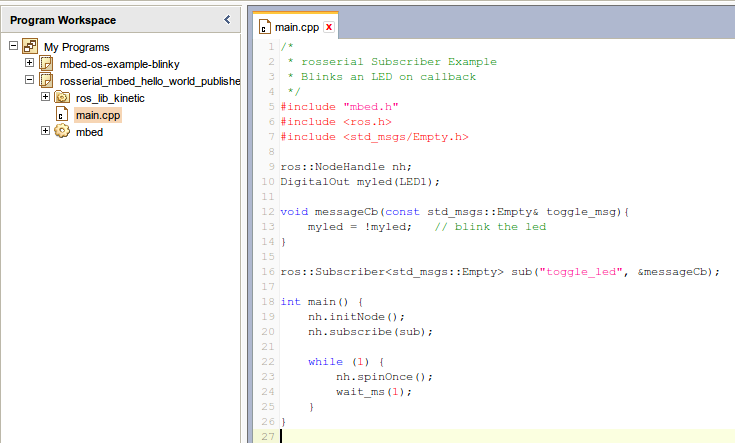


และให้ Import! ที่อยู่เหนือปุ่ม Search

ระบบจะเด้งหน้าต่างขึ้นมา ให้ติ๊กที่ Update all แล้วกด Import



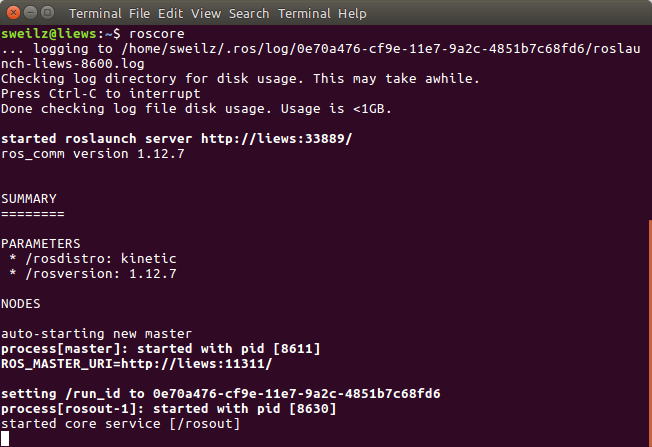
3. เปิดไฟล์ main.cpp แล้วกด Compile



จากนั้นก็ทำแบบเดิมเซฟไฟล์ .bin แล้วก๊อปไปไว้ใน mbed drive

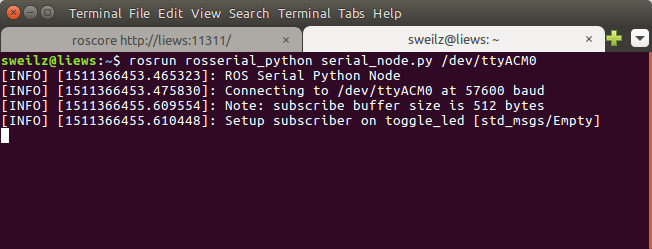
4. เมื่อก๊อปไฟล์เสร็จแล้ว board จะรัน code นั้นโดยอัตโนมัติ เป็นอันจบขั้นตอน เหลือแค่ทดสอบรัน ROS ทางฝั่ง PC

Start roscore: เปิด Terminal



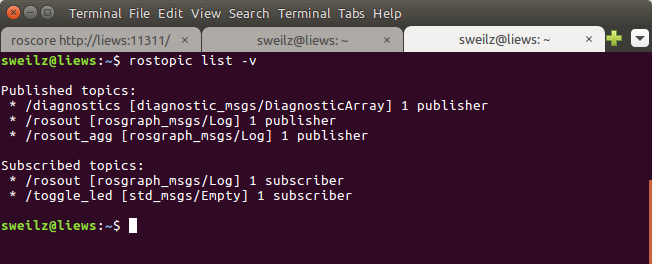
Start the ROS server: เปิด Teminal tab ใหม่ขึ้นมา

$ rosrun rosserial*\_python serial\_*node.py /dev/ttyACM0



List ROS topics: เปิด Terminal tab ใหม่ขึ้นมา

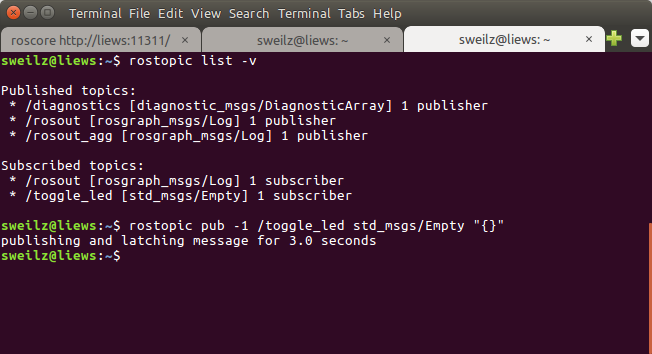
$ rostopic list –v



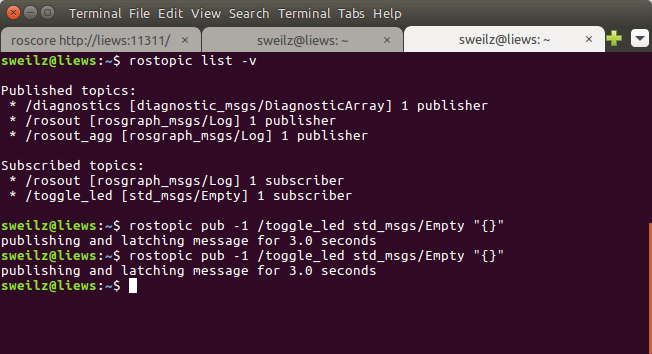
จะเห็นว่ามี Subcribed topics ที่ชื่อว่า `/toggle\_led` ขึ้นมา ซึ่งเป็นการรอสั่ง toggle led หลังจากได้รับ message ที่เป็นชนิด `std\_msgs/Empty`

**Test toggle led:**

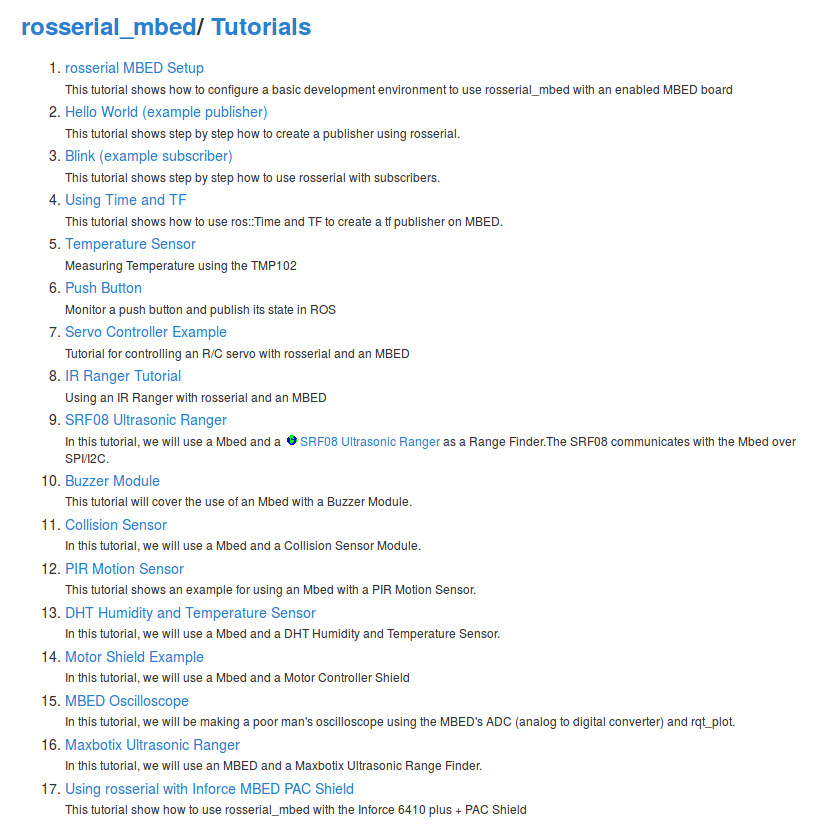
$ rostopic pub -1 /toggle*\_led std\_*msgs/Empty "{}"



หลอดไฟสีเหลืองที่ board จะติดขึ้นมา หากใช้คำสั่งเดิมอีกครั้งหลอดไฟจะดับ



จากนี้เราสามารถเขียนโปรแกรม mbed board โดยใช้ภาษา C++ ซึ่งสามารถทำตาม Tutorials ได้จากลิ้งค์นี้ <http://wiki.ros.org/rosserial_mbed/Tutorials>



1. การเคลื่อนที่ของหุ่นยนต์ [http://dusithost.dusit.ac.th/~juthawut\_cha/download/Lession%2010%20Robotics.pdf] [↑](#footnote-ref-2)
2. ณัฐพงษ์ วารีประเสริฐ และณรงค์ ล่ำดี (2552: 374) [↑](#footnote-ref-3)
3. DARPA robotics challenge (DARPA, 2015) [https://www.darpa.mil/about-us/about-darpa] [↑](#footnote-ref-4)
4. ASIMO Humanoid robot [http://asimo.honda.com/asimo-specs/] [↑](#footnote-ref-5)
5. HRP-3 [http://global.kawada.jp/mechatronics/hrp3.html] [↑](#footnote-ref-6)
6. LOLA [http://www.amm.mw.tum.de/en/research/current-projects/humanoid-robots/lola/] [↑](#footnote-ref-7)
7. งานวิจัยหุ่นยนต์วิ่งสองขา [http://www.takanishi.mech.waseda.ac.jp/top/research/running/index.htm] [↑](#footnote-ref-8)
8. Rose, J. and Gamble, J., 1993, Human Walking, Williams & Wilkins, Philadelphia, pp. 10-44. [↑](#footnote-ref-9)
9. Mechanic gyroscope two-degree of freedom [https://www.bosch-sensortec.com/bst/products/motion/gyroscope/overview\_gyroscopesensors] [↑](#footnote-ref-10)
10. Accelerometer and Gyroscopes Sensor [https://www.maximintegrated.com/en/app-notes/index.mvp/id/5830] [↑](#footnote-ref-11)
11. http://www.antonigracia.com/nl/project/icub-humanoide-platform/ [↑](#footnote-ref-12)
12. Robot Actuator [http://support.robotis.com/en/product/actuator/dynamixel/ex\_series/ex-106.htm] [↑](#footnote-ref-13)
13. EX-106+ Mode [http://www.trossenrobotics.com/dynamixel-ex-106-robot-actuator.aspx] [↑](#footnote-ref-14)
14. TCRT5000 [http://francois-ouellet.ca/using-a-tcrt-5000-sensor-with-your-arduino/] [↑](#footnote-ref-15)
15. MPU-9250 [http://www.arduiner.com/en/gy-series-axis-accellerometers/6924-gy9255-mpu9255-sensor-module-alternative-mpu9150-mpu9250-3809200640200.html] [↑](#footnote-ref-16)