



รายงานวิจัยฉบับสมบูรณ์ ประจำปี 2564

โครงการวิจัยที่ รหัส 3060-A180

การพัฒนาเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

Development of Automated Workstation for Toxic Residue Analysis

หัวหน้าโครงการวิจัย

นฤพล วัฒนภาพ

แผนกงานอารักขาพืช

ได้รับทุนวิจัยสนับสนุนจากมูลนิธิโครงการหลวง

กันยายน พ.ศ. 2564



รายงานวิจัยฉบับสมบูรณ์ ประจำปี 2563

โครงการวิจัยที่ รหัส 3060-A180

การพัฒนาเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

Development of Automated Workstation for Toxic Residue Analysis

หัวหน้าโครงการวิจัย

นฤพล วัฒนภาพ

แผนกงานอารักขาพืช

คณะวิจัย

มูลนิธิ

วิฑูรย์ บุญสง่า

ศูนย์พัฒนาโครงการหลวงแม่สาใหม่

ได้รับทุนวิจัยสนับสนุนจากมูลนิธิโครงการหลวง

กันยายน พ.ศ. 2564

กิตติกรรมประกาศ

งานวิจัยฉบับนี้ ได้รับการสนับสนุนจากมูลนิธิโครงการหลวง ประจำปีงบประมาณ พ.ศ. 2564 ผู้วิจัยขอขอบพระคุณคณะผู้บริหาร และหัวหน้าศูนย์ฯ แม่สาใหม่ที่กรุณาให้การสนับสนุนสถานที่ทำวิจัย และผู้ตรวจงานวิจัยทุกท่านที่อนุญาติให้ผู้วิจัยได้จัดทำ งานวิจัยเรื่องดังกล่าวนี้ และให้โอกาสแก่ผู้วิจัยมา ณ ที่นี้

การวิจัยในครั้งนี้ สำเร็จลงได้ด้วยความช่วยเหลือสนับสนุน แนะนำอย่างดียิ่งจาก รองศาสตราจารย์.ดร.นุชนาฏ จงเลขา ที่ปรึกษางานวิจัย ที่ได้ให้คำปรึกษา แนะนำ และให้ข้อคิดเห็นต่างๆ ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาโดยตลอด และขอขอบคุณ นางสาวกาญจนา วิชิตระกูลถาวร หัวหน้าแผนกงานอารักขาพืช ในการอำนวยความสะดวกและสนับสนุนการทำวิจัยมาโดยตลอด รวมถึงผู้ที่ช่วยตอบแบบสอบถามประเมินความพึงพอใจทุกท่าน ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

คุณค่าและประโยชน์ที่เกิดจากงานวิจัยฉบับนี้ ผู้วิจัยขอมอบให้เป็นกตัญญู กตเวทิตา แก่ พ่อแม่ ครู อาจารย์ และผู้มีพระคุณทุกท่าน

คณะผู้วิจัย

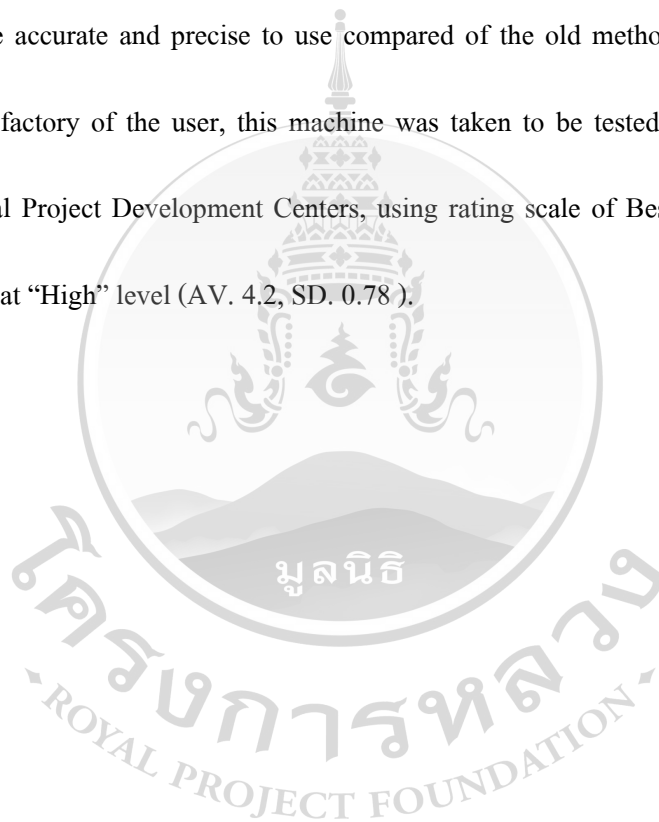
บทคัดย่อ

ทำการพัฒนาเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติในขั้นตอนการทดสอบด้วยชุดทดสอบ GT. เครื่องที่พัฒนาขึ้นนี้ประกอบด้วยไมโครโปรเซสเซอร์อาร์ดิวโน อูโน เซอร์โวมอเตอร์และสเตปปีงมอเตอร์ สามารถดูค่าของเหลวและตรวจวิเคราะห์สารตกค้างได้อย่างมีประสิทธิภาพและรวดเร็ว ซึ่งมีความถูกต้องแม่นยำสูง ผลการประเมินความพึงพอใจของผู้ปฏิบัติการวิเคราะห์สารตกค้างมูลนิธิโครงการหลวง จำนวน 27 คน แบบมาตรประมาณค่าตามวิธีของเบสท์ พบว่ามีความพึงพอใจต่อเครื่องวัดค่าสี ในระดับมาก (ค่าเฉลี่ย 4.2 ค่าเบี่ยงเบนมาตรฐาน 0.78)



Abstract

The automated workstation for Toxic residue analysis was developed to using for analysing for the food toxic residue in the process of checking pesticide residues in the samples by GT pesticide test kit. This machine consists of Arduino UNO microprocessor board, servo and stepping motors are included. The developed machine can pipetting and analysing the level of toxic residue from the extract effective and rapidly. It is more accurate and precise to use compared of the old method which pipetted by the tester. To test on satisfactory of the user, this machine was taken to be tested by 27 pesticide residue testers located at Royal Project Development Centers, using rating scale of Best. It was found that the value of satisfactory is at “High” level (AV. 4.2, SD. 0.78).



สารบัญ

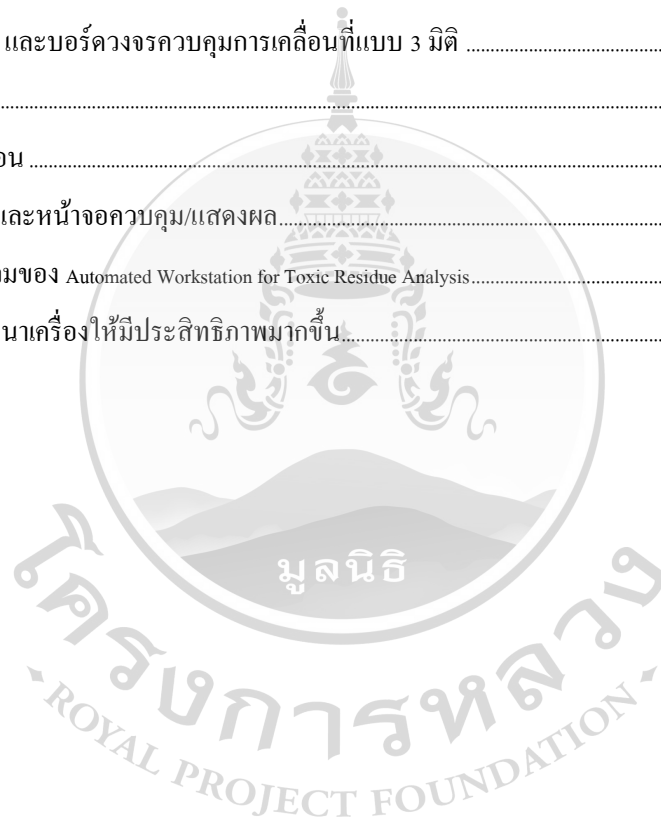
| | หน้า |
|--|------|
| กิตติกรรมประกาศ | ก |
| บทคัดย่อ | ข |
| Abstract | ค |
| สารบัญ | ง |
| สารบัญภาพ | จ |
| สารบัญตาราง | ช |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความสำคัญและที่มาของปัญหา | 1 |
| 1.2 วัตถุประสงค์ของการทดลอง และขอบเขตของการวิจัย | 7 |
| 1.3 ประโยชน์ที่คาดว่าจะได้รับ | 9 |
| บทที่ 2 ทฤษฎีและแนวคิดที่เกี่ยวข้อง | 11 |
| 2.1 ทฤษฎีหรือแนวความคิดที่ใช้ในการแก้ปัญหา | 11 |
| 2.2 ผลงานวิจัยที่เคยทำมาก่อน หรืองานวิจัยที่เกี่ยวข้อง | 43 |
| บทที่ 3 กรรมวิธีการทดลอง | 45 |
| 3.1 สถานที่ทดลอง ระยะเวลาทำการทดลอง | 45 |
| 3.2 วัสดุทดลองที่ใช้ | 45 |
| 3.3 วิธีทดลองที่ใช้ การบันทึกข้อมูล วิธีการวิเคราะห์ข้อมูล | 46 |
| บทที่ 4 ผลการวิจัย | 51 |
| 4.1 การออกแบบและประดิษฐ์ | 51 |
| 4.2 การทดสอบประสิทธิภาพการทำงานของเครื่องที่สร้างขึ้น | 56 |
| 4.4 ผลการประเมินความพึงพอใจที่มีต่อเครื่องวัดค่าสี | 59 |
| 4.5 ปรับปรุงแก้ไข | 59 |
| 4.6 ผลสัมฤทธิ์จากการประเมินตัวชี้วัดของงานวิจัย | 60 |
| บทที่ 5 สรุปผลและข้อเสนอแนะ | 61 |
| 5.1 สรุปผล | 61 |
| 5.2 ข้อเสนอแนะ | 62 |
| เอกสารอ้างอิง | 63 |
| ภาคผนวก ก คู่มือประกอบการใช้งาน | 65 |
| ภาคผนวก ข แบบประเมินความพึงพอใจ | 71 |

สารบัญภาพ

หน้า

| | |
|--|----|
| ภาพที่ 1 ชุดตรวจหายาฆ่าแมลง/ สารพิษตกค้าง “GT” | 1 |
| ภาพที่ 2 การอ่านผลการทดสอบโดยการเทียบความเข้มของสีของสารละลายในหลอด | 7 |
| ภาพที่ 3 รูปแบบของความผิดพลาดจาก Human Error ในการเติมน้ำยาลิควิเดสสารตกค้าง GT | 11 |
| ภาพที่ 4 ความผิดพลาดจากการกระทำของมนุษย์ | 14 |
| ภาพที่ 5 เครื่อง Beckman Coulter Biomek 4000 Automated Liquid Handling Workstations | 16 |
| ภาพที่ 6 เครื่อง PerkinElmer JANUS G3 Automated Workstation | 17 |
| ภาพที่ 7 สเตปป์มอเตอร์ | 21 |
| ภาพที่ 8 สเตปป์มอเตอร์ 4 เฟส แบบยูนิโพลาร์ (Uni-polar stepper motor) | 22 |
| ภาพที่ 9 ลักษณะภายนอกและโครงสร้างภายในของสเตปป์มอเตอร์แบบ Unipolar | 22 |
| ภาพที่ 10 ลักษณะภายนอกและโครงสร้างภายในของสเตปป์มอเตอร์แบบ Bipolar | 22 |
| ภาพที่ 11 ลำดับขั้นการทำงานของสเตปป์มอเตอร์ | 23 |
| ภาพที่ 12 แสดงกระบวนการทำงานของ Stepper Motor | 26 |
| ภาพที่ 13 การประยุกต์ใช้งาน Stepping Motor Drive | 27 |
| ภาพที่ 14 ส่วนประกอบภายในของ RC Servo | 28 |
| ภาพที่ 15 ส่วนประกอบภายนอก RC Servo Motor | 28 |
| ภาพที่ 16 RC Servo Motor Ref: www.pololu.com | 29 |
| ภาพที่ 17 Servo Motor Block Diagram | 29 |
| ภาพที่ 18 Remote Control ที่ใช้ควบคุมมอเตอร์ด้วยสัญญาณวิทยุต่าง | 30 |
| ภาพที่ 19 องศาการหมุนของ RC Servo Motor | 31 |
| ภาพที่ 20 การกำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา | 31 |
| ภาพที่ 21 เซอร์โวไดรฟ์เวอร์ (Servo Driver) | 33 |
| ภาพที่ 22 เซอร์โวมอเตอร์ (Servo Motor) | 34 |
| ภาพที่ 23 แสดงความกว้างของสัญญาณ ON หรือ %Duty Cycle สำหรับควบคุมตำแหน่งของ RC Servo | 35 |
| ภาพที่ 24 การหมุน Servo Motor ที่ 0 และ 90 องศา | 38 |
| ภาพที่ 25 การหมุน Servo Motor ที่ 90 และ 180 องศา | 39 |
| ภาพที่ 26 ตัวอย่างการควบคุมตำแหน่ง RC Servo Motor | 39 |
| ภาพที่ 27 การกำหนดสัญญาณความกว้างของพัลส์ที่ใช้ควบคุมให้ servo motor หมุนเป็นมุม | 40 |
| ภาพที่ 28 การใช้ Arduino ควบคุม servo motor | 40 |

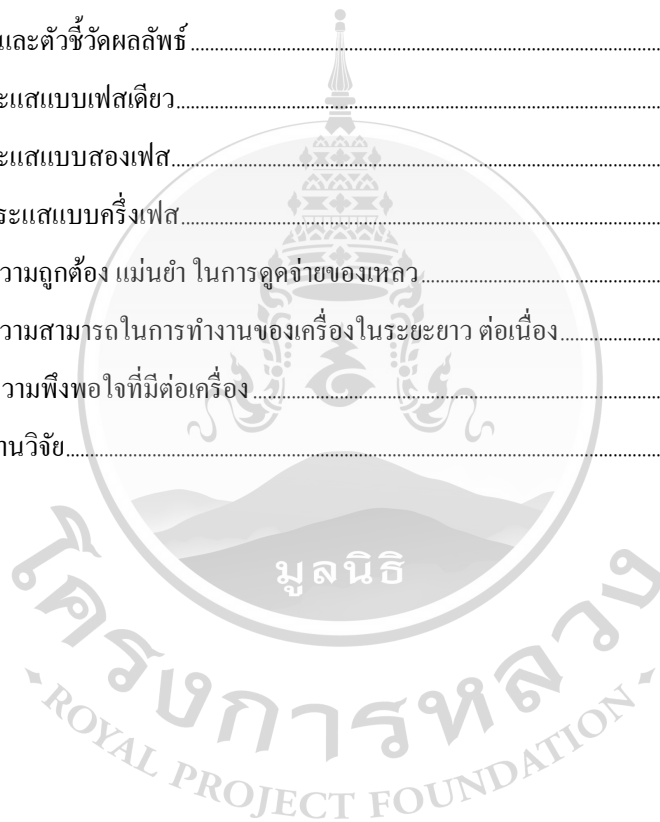
| | |
|--|----|
| ภาพที่ 29 รางเลื่อน linear rail | 42 |
| ภาพที่ 30 เครื่อง CNC router..... | 42 |
| ภาพที่ 31 Sumitomo's robotic reaction workstation[19] | 43 |
| ภาพที่ 32 Robotic ARM Workstation[19]..... | 44 |
| ภาพที่ 33 แผนผังกิจกรรมการดำเนินงาน | 46 |
| ภาพที่ 34 ส่วนประกอบภายในเครื่องที่ประดิษฐ์ขึ้น..... | 51 |
| ภาพที่ 35 servo motor ควบคุมการหมุนของเฟืองเพื่อส่งกำลังให้เพลาคือเคลื่อนที่ | 52 |
| ภาพที่ 36 ชุดแกนเลื่อน Linear Rail Axis Slide Stroke Long Stage Actuator r..... | 52 |
| ภาพที่ 37 ชุดชุดจ่ายของเหลว..... | 53 |
| ภาพที่ 38 ชุด micro processor และบอร์ดวงจรควบคุมการเคลื่อนที่แบบ 3 มิติ | 53 |
| ภาพที่ 39 ชุดโครงสร้าง | 54 |
| ภาพที่ 40 ระบบทำความร้อน | 54 |
| ภาพที่ 41 ระบบปั๊มสวิตซ์และหน้าจอควบคุม/แสดงผล..... | 55 |
| ภาพที่ 42 โครงสร้างโดยรวมของ Automated Workstation for Toxic Residue Analysis..... | 55 |
| ภาพที่ 46 การปรับปรุงพัฒนาเครื่องให้มีประสิทธิภาพมากขึ้น..... | 58 |



สารบัญตาราง

หน้า

| | |
|---|----|
| ตาราง 1 สัดส่วนปริมาตรในการเติมน้ำยา GT..... | 4 |
| ตาราง 2 วิธีการสกัดตัวอย่างสำหรับพืชต่างชนิด..... | 5 |
| ตาราง 3 การเติมน้ำยาทดสอบ..... | 6 |
| ตาราง 4 การอ่านผล..... | 7 |
| ตาราง 5 ขอบเขตของงานวิจัย..... | 8 |
| ตาราง 6 ผลผลิต (Out put) และตัวชี้วัดของ โครงการวิจัย..... | 9 |
| ตาราง 7 ผลลัพธ์ (Outcome) และตัวชี้วัดผลลัพธ์..... | 10 |
| ตาราง 8 ลำดับการป้อนกระแสแบบเฟสเดียว..... | 24 |
| ตาราง 9 ลำดับการป้อนกระแสแบบสองเฟส..... | 24 |
| ตาราง 10 ลำดับการป้อนกระแสแบบครึ่งเฟส..... | 25 |
| ตาราง 11 ผลการทดสอบความถูกต้อง แม่นยำ ในการดูจ่ายของเหลว..... | 56 |
| ตาราง 13 ผลการทดสอบความสามารถในการทำงานของเครื่องในระยะยาว ต่อเนื่อง..... | 57 |
| ตาราง 14 ผลการประเมินความพึงพอใจที่มีต่อเครื่อง..... | 59 |
| ตาราง 16 ผลสัมฤทธิ์ของงานวิจัย..... | 60 |



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ชุดน้ำยาตรวจสอบยาฆ่าแมลงตกค้าง “GT pesticide test kit” ถูกคิดค้น โดย นางกอบทอง รูปหอม กรมวิทยาศาสตร์การแพทย์ ใช้ในการวิเคราะห์สารพิษตกค้างและคัดกรองตัวอย่างที่ไม่ปลอดภัย จากสารพิษ โดยรวมที่มีอยู่ในตัวอย่างที่มีผลยับยั้งการทำงานของเอนไซม์อะซิติลโคลีนเอสเทอเรส ซึ่งเป็นเอนไซม์ที่ได้จากซีรัมมนุษย์และสัตว์ทั่วไป ซึ่งจะแสดงความไวต่อการรับรู้ความเป็นพิษได้ โดยปกติเอนไซม์อะซิติลโคลีนเอสเทอเรส (Acetylcholinesterase, AChE) มีหน้าที่สลายอะซิติลโคลีน (Acetylcholine, ACh) ให้เป็นโคลีนและกรดอะซิติล



ภาพที่ 1 ชุดตรวจหาฆ่าแมลง/ สารพิษตกค้าง “GT”

สารกำจัดแมลงกลุ่มออกแกโนฟอสเฟตและคาร์บาเมท เป็นกลุ่มสารที่สามารถยับยั้งการทำงานของเอนไซม์อะซิติลโคลีนเอสเทอเรส โดยสารแต่ละชนิดในกลุ่มนี้จะมีคุณสมบัติในการยับยั้งการทำงานของเอนไซม์ได้ไม่เท่ากัน ทั้งนี้ขึ้นอยู่กับระดับความเป็นพิษของแต่ละสาร ดังนั้นจึงสามารถตรวจหาสารกำจัดแมลงโดยวิธีทางอ้อม ด้วยการตรวจหาปริมาณ ACh ที่เหลือจากการถูก AChE ย่อยสลาย โดยให้ ACh ทำปฏิกิริยากับไฮดรอกซิลเอมีน (Hydroxylamine) จะได้กรดอะซิติลไฮดรอกซามิก (Acetylhydroxamic acid) ซึ่งเมื่อเกิดปฏิกิริยากับสารเฟอร์ริกคลอไรด์ (Ferric chloride, $FeCl_3$) จะได้สารประกอบสีน้ำตาลแดง สีที่เกิดขึ้นนี้จะมีสีเข้มหรืออ่อนขึ้นขึ้นอยู่กับปริมาณ ACh ที่เหลือจากการถูก AChE ย่อยสลาย

ในตัวอย่างผักหรือผลไม้ที่ไม่มีสารกำจัดแมลง เอนไซม์จะไม่ถูกยับยั้งการทำงาน($I_{0\%}$) ACh จะถูกย่อยสลายได้มากจึงเหลือปริมาณ ACh น้อยเป็นเหตุให้สีที่เกิดขึ้นเป็นสีน้ำตาลอ่อน ในทางกลับกันถ้าหากใน

ตัวอย่างมีสารกำจัดแมลงมาก เอนไซม์จะถูกยับยั้งการทำงานมาก ทำงานได้ไม่ดี ทำให้มีปริมาณ ACh เหลืออยู่มาก สีที่เกิดขึ้นจะเป็นสีน้ำตาลแดงเข้ม ดังนั้นเมื่อเทียบกับหลอดมาตรฐานทำให้ทราบระดับของ สารกำจัดแมลงได้ ดังนั้นเมื่อทดสอบแล้วว่าตัวอย่างผักหรือผลไม้ที่ให้สีเกินค่าที่ 50% ขึ้นไป ($I_{50\%}$) แสดงว่า ผักหรือผลไม้เหล่านั้นมีสารพิษในปริมาณที่เมื่อผู้บริโภครับประทานแล้วจะเป็นผลให้เอนไซม์ที่อยู่ในร่างกาย ชนิดนี้ทำงานได้ไม่เต็มที่ จึงประเมินได้ว่าน่าจะปลอดภัยต่อผู้บริโภค

โดยปกติการอ่านค่าระดับความเป็นพิษตามวิธีของนาง กอบทอง ฐูปหอม จะอ่านค่าหรือตรวจวัดได้ โดยเทียบความเข้มของสีกับหลอดควบคุม(ไม่พบสารพิษ) และหลอดตัดสิน ($I_{50\%}$) เท่านั้น ซึ่งค่ากำหนด ดังกล่าวได้มาจากการประเมินเปรียบเทียบค่าความไม่ปลอดภัย กับค่ากำหนด Codex MRL ที่ให้ผลใกล้เคียง กัน และค่านี้นำไปใช้ในการประเมินความไม่ปลอดภัยของผักผลไม้ทั่วไปที่กำหนดในประเทศไทย แต่สำหรับ ผลผลิตของโครงการหลวงเน้นถึงคุณภาพและความปลอดภัยต่อผู้บริโภคยิ่งขึ้น ดังนั้น รองศาสตราจารย์ ดร. นุชนาฏ จงเลขา แผนกงานอารักขาพืช มูลนิธิโครงการหลวง จึงได้เกิดแนวคิดให้แบ่งระดับสารพิษตกค้าง ออกเป็น 6 ระดับ จากเดิม 2 ระดับคือหลอดควบคุม (0) กับหลอดตัดสิน (+5) โดย 0 คือตรวจไม่พบสารพิษ ตกค้าง +1, +2, +3, +4 และ +5 คือตรวจพบสารพิษตกค้างที่ยับยั้งการทำงานของเอนไซม์ 10, 20, 30, 40 และ 50 เปอร์เซ็นต์ตามลำดับและเพื่อให้เกิดความปลอดภัยต่อผู้บริโภคยิ่งขึ้น มูลนิธิโครงการหลวง ได้ ถือเอาตัวอย่างผลิตภัณฑ์ที่มีปริมาณสารพิษที่ยับยั้งการทำงานของเอนไซม์ตั้งแต่ 30% ขึ้นไป (ตั้งแต่ +3 ขึ้นไป) เป็นตัวอย่างที่ไม่ปลอดภัยต่อผู้บริโภค

หลักการสกัดตัวอย่าง

ใช้เทคนิค solvent extraction โดยการเติมน้ำยาสกัด-1 (dichloromethane) ลงในตัวอย่างพืชที่ บดละเอียดแล้ว ทำการเขย่าประมาณ 1 นาที เพื่อสกัดสารพิษตกค้างในตัวอย่างให้ละลายออกมา และวางทิ้ง ไว้ประมาณ 15 นาที เพื่อทำให้เกิดการแยกชั้นและง่ายต่อการดูดชั้นตัวทำละลาย dichloromethane ซึ่งอยู่ ด้านล่างของขวดตัวอย่างไปใช้

เนื่องจาก dichloromethane เป็นตัวทำละลายอินทรีย์ที่สามารถยับยั้งการทำงานของเอนไซม์จึงต้อง เปลี่ยนตัวทำละลายเป็นน้ำยาสกัด-2 (5%ethanol) โดยทำการเติมน้ำยาสกัด-2 ลงไปในแล้วเป่าระเหยด้วย อากาศ ทำให้ dichloromethane ระเหยออกไป, สารพิษจะละลายใน 5%ethanol และจะได้สารสกัดตัวอย่าง (sample extract) ที่พร้อมนำไปตรวจวิเคราะห์

หลักการตรวจวิเคราะห์

เติมน้ำยา GT-1 (AChE enzyme) ปริมาณ 2 ส่วน ผสมกับสารสกัดตัวอย่าง 1 ส่วน รอให้ เกิดปฏิกิริยาการยับยั้งการทำงานของเอนไซม์ให้สมบูรณ์ประมาณ 15 นาที

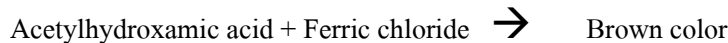
เติมน้ำยา GT-2 (Acetylcholine, Ach substrate) ปริมาณ 1 ส่วน เพื่อเป็นการวัดกิจกรรมของเอนไซม์ (enzyme activity) ในช่วงเวลาประมาณ 60 นาทีในอ่างน้ำอุ่น หาก AChE ถูกสารพิษยับยั้งการทำงานมาก enzyme hydrolysis activity จะลดลงทำให้มีปริมาณ Ach เหลืออยู่มาก (หลุดตัดตีนระดับ $I_{50\%}$ จึงมีการ เติมเติมน้ำยา GT-2 เป็น 1.5 ส่วน เพื่อป้องกันการมีปริมาณ Ach ที่เหลืออยู่ เนื่องจากถูกสารพิษยับยั้ง การทำงานเอนไซม์)

เติมน้ำยา GT-3 (Hydroxyl amine hydrochloride in NaOH) ปริมาณ 4 ส่วน เข้าไปทำปฏิกิริยากับ Acetylcholine ที่เหลือจากการ hydrolysis ด้วยเอนไซม์ เกิดสารประกอบ Acetylhydroxamic acid ในสภาวะ เบสแก่

เติมน้ำยา GT-4 (Hydrochloric acid, HCl) ปริมาณ 2 ส่วน เพื่อทำปฏิกิริยาสะท้อนจากเบสเป็นกลาง

เติมน้ำยา GT-5 (Ferric chloride in HCl) ปริมาณ 2 ส่วน เพื่อให้ Ferric ion(Fe^{3+}) ทำปฏิกิริยา coordination กับ Acetylhydroxamic acid เกิด complex compound ที่มีสีน้ำตาล

ปฏิกิริยาเคมีของการตรวจสารกำจัดแมลง



สัดส่วนของการเติมน้ำยาจาก Gobthong Thoophom, 2554, Hand book GT-Pesticide test kit ใช้การ วัดระดับจากขีดของหลอดหยดพลาสติก 1 ขีด ซึ่งประมาณ 0.25 ซีซี = 1 ส่วน ซึ่งปริมาตรของ GT-2 ที่ $I_{10\%}$, $I_{30\%}$ และ $I_{50\%}$ จะมีความยากในการ ปิเปต และมี error ได้ง่ายที่ปริมาตรทศนิยมตำแหน่งที่ 3 จึงทำการ ปรับปรุงปริมาณใหม่ในสัดส่วนเดิมและปิเปตได้ง่ายขึ้นจึงช่วยลด error จากการปิเปต ดังตารางที่ 1

ตาราง 1 สัดส่วนปริมาตรในการเติมน้ำยา GT

| Reagent | สัดส่วนน้ำยา | สัดส่วนน้ำยาแบบเดิม | สัดส่วนน้ำยาที่ปรับปรุง |
|-------------------------|--------------|----------------------|-------------------------|
| สารสกัดตัวอย่าง | 1 ส่วน | 1 ซีดี =0.25 ซีซี | 0.2 mL |
| GT-1 | 2 ส่วน | 2 ซีดี =0.5 ซีซี | 0.4 mL |
| GT-2 | 1 ส่วน | 1 ซีดี =0.25 ซีซี | 0.2 mL |
| GT-2(I _{10%}) | 1.1 ส่วน | 1.5 ซีดี =0.275 ซีซี | 0.22 mL |
| GT-2(I _{20%}) | 1.2 ส่วน | 1.5 ซีดี =0.3 ซีซี | 0.24 mL |
| GT-2(I _{30%}) | 1.3 ส่วน | 1.5 ซีดี =0.325 ซีซี | 0.26 mL |
| GT-2(I _{40%}) | 1.4 ส่วน | 1.5 ซีดี =0.35 ซีซี | 0.28 mL |
| GT-2(I _{50%}) | 1.5 ส่วน | 1.5 ซีดี =0.375 ซีซี | 0.3 mL |
| GT-3 | 4 ส่วน | 4 ซีดี =1 ซีซี | 0.8 mL |
| GT-4 | 2 ส่วน | 2 ซีดี =0.5 ซีซี | 0.4 mL |
| GT-5 | 2 ส่วน | 2 ซีดี =0.5 ซีซี | 0.4 mL |

สารเคมีและการเตรียม (chemical reagent and preparation)

- 1 ชุดน้ำยาตรวจหาฆ่าแมลงชนิดสำเร็จรูป GT-Pesticide Test Kit ประกอบด้วย น้ำยา GT-1, น้ำยา GT- 2 และ 2.1, น้ำยา GT-3 และ 3.1, น้ำยา GT-4, น้ำยา GT-5
- 2 น้ำยาสกัด-1 คือ dichloromethane หรือ methylene chloride AR grade
- 3 น้ำยาสกัด-2 คือ สารละลาย ethanol 5%(v/v) ตวง absolute ethanol 99.9% ผสมกับน้ำปราศจากไอออนหรือน้ำกลั่น ในอัตราส่วน 5 : 95 โดยปริมาตร ตัวอย่างการเตรียม เช่น ตวง ethanol 50 mL ด้วยกระบอกตวง เทใส่ขวดวัดปริมาตรขนาด 1,000 mL เติมน้ำกลั่นถึงขีดกำหนดปริมาตร จากนั้นจึงปรับปริมาตรให้ถึงขีดกำหนดปริมาตร ปิดฝาและเขย่าให้เข้ากัน เก็บในขวดแก้วฝาเกลียว

ขั้นตอนการสกัดตัวอย่าง

1. ชั่งตัวอย่าง ใสลงในหลอด centrifuge พลาสติกฝาเกลียวและเติมน้ำยาสกัด-1 (ตามตารางที่ 2) ปิดฝาให้สนิท เขย่าแรงๆ ด้วยมือหรือเครื่องเขย่า เป็นเวลาอย่างน้อย 30 วินาที วางไว้ให้แยกชั้น ประมาณ 15 นาที หรือ นำไปหมุนเหวี่ยงให้ตกตะกอนด้วยเครื่อง centrifuge ที่ความเร็วรอบ ประมาณ 3500 RPM ประมาณ 1 นาที
2. คูดน้ำยาสกัด-2 จำนวน 1 mL ใสลงในหลอดทดลอง จากนั้นคูดสารสกัดตัวอย่างเฉพาะส่วนใส ด้านล่าง(ชั้นของ dichloromethane) ให้ได้ปริมาตรตามตารางที่ 2 ใสลงในหลอดเดียวกัน (กรณีที วางไว้แล้วการแยกชั้นระหว่างสารสกัด-1 กับตัวอย่าง ไม่สมบูรณ์ ไม่สามารถคูดสารสกัด-1 ได้ ให้หมุนเหวี่ยงตกตะกอนด้วยเครื่อง centrifuge หากตัวอย่างประเภทพืช ถั่ว แป้ง ที่บดเป็น ผงละเอียดให้ใช้เทคนิคการกรอง โดยใช้กรวยแก้วขนาดเล็กคูดด้วยสำลีสระอาดปริมาณเล็กน้อย หรือกรองผ่านกระดาษกรอง Whatman no. 1)
3. นำไปประเหยสารสกัด-1 ออก ด้วยเครื่องเป่าระเหยสารสกัดตัวอย่าง โดยค่อยๆ รุ่มปลายท่อเป่า อากาศลงในหลอดตัวอย่าง ปล่อยให้มีการระเหยน้ำยาสกัด-1 ออกให้หมด ใช้เวลาประมาณ 1 นาที จนเหลือแต่ชั้นของน้ำยาสกัด-2 เท่านั้น (ควรทำในตู้ดูดควัน) สารละลายที่ได้จากการสกัดนี้ คือ สารสกัดตัวอย่าง (Sample extract)
4. คูดสารสกัดตัวอย่าง ปริมาตร 0.2 mL ใสลงในหลอดทดลองอีกอัน เพื่อใช้เป็นหลอดตัวอย่าง สำหรับทดสอบ

ตาราง 2 วิธีการสกัดตัวอย่างสำหรับพืชต่างชนิด

| ตัวอย่าง | น้ำหนักที่ชั่ง (g) | ปริมาตรน้ำยาสกัด-1 (mL) | ปริมาณที่คูดไปใช้ (mL) |
|---|--------------------|-------------------------|------------------------|
| ผักสด ผลไม้สดทั่วไป เช่น กะหล่ำปลี ผักกาดขาวปลี | 5 | 5 | 1 |
| ผักผลไม้สด ที่มีปริมาณน้ำมาก ที่มีปัญหาการแยกชั้นของตัวอย่างกับสารสกัดและไม่ใช้ centrifuge ช่วยการตกตะกอน | 5 | 10 | 2 |
| ผักพื้นบ้าน เครื่องเทศ เช่น จิง ข่า หอม กระเทียม ตะไคร้ ใบมะกรูด พริก สด กระเพรา | 2.5 | 10 | 1 |
| สมุนไพรสด | 1 | 10 | 1 |
| สมุนไพรแห้ง | 0.5 | 10 | 1 |
| ธัญพืช และของแห้ง เช่น ถั่วลิสงแห้ง ข้าวสาร งาม | 2.5 | 10 | 1 |

ขั้นตอนปฏิบัติการทดสอบตัวอย่าง

1. ควบคุมอุณหภูมิอ่างน้ำอุ่น (water bath) อยู่ที่ 32-36 องศาเซลเซียส
2. นำขวดน้ำยา GT-1 มาแช่ใน water bath ให้อุ่น หากเป็นน้ำแข็งต้องรอให้ละลายทั้งหมด
3. เตรียมหลอดควบคุม(negative control) และหลอดตัดสิน(Judgment, positive control) ดังนี้ :
 หลอดควบคุม(control) I_{0%} หรือ 0
 หลอดตัดสินระดับ I_{10%} หรือ +1
 หลอดตัดสินระดับ I_{30%} หรือ +3 (เป็นระดับที่พบไม่ปลอดภัย)
 ทุกหลอดใส่น้ำยาสกัด-2 จำนวน 0.2 mL
4. นำหลอดตัวอย่าง หลอดควบคุม หลอดตัดสิน ลงแช่ในอ่างน้ำอุ่น
5. ใส่น้ำยาในขวด GT-1 ปริมาตร 0.4 mL ใสลงในทุกหลอด เขย่าและวางไว้ประมาณ 15 นาที
6. ระหว่างรอเวลา ให้ผสมน้ำยาจากขวด GT-2.1 ลงในขวด GT-2 และผสมน้ำยา GT-3.1 ลงในขวด GT-3
7. เมื่อครบเวลา เริ่มใส่น้ำยาผสมจากขวด GT-2 ใสหลอดตัวอย่าง หลอดควบคุม และหลอดตัดสิน ปริมาตรตามตารางที่ 2 เขย่าและตั้งเวลาที่ 60 นาที
8. เมื่อครบ 60 นาที ใส่น้ำยาที่ผสมแล้วจากขวด GT-3 ปริมาตร 0.8 mL ใสทุกหลอด เขย่า
9. ใส่น้ำยา GT-4 ปริมาตร 0.4 mL ใสลงในทุกหลอด เขย่า
10. ใส่น้ำยา GT-5 ปริมาตร 0.4 mL ใสลงในทุกหลอด เขย่า และเปรียบเทียบความเข้มของสีที่เกิดขึ้น

ตาราง 3 การเติมน้ำยาทดสอบ

| น้ำยา | หลอดตัวอย่าง | control | +1 | +3 | ข้อปฏิบัติ |
|----------|---------------------------|------------------|------------------|------------------|--|
| สารละลาย | สารสกัดตัวอย่าง 0.2 mL | สกัด-2 0.2 mL | สกัด-2 0.2 mL | สกัด-2 0.2 mL | ปิเปตใสลงในหลอดทดลอง วางในอ่างควบคุมอุณหภูมิ 32-36 °C |
| GT-1 | 0.4 mL | 0.4 mL | 0.4 mL | 0.4 mL | เขย่า ตั้งไว้ 15 นาทีในอ่างควบคุมอุณหภูมิ |
| GT-2 | 0.2 mL | 0.2 mL | 0.22 mL | 0.26 mL | เขย่า ตั้งไว้ 60 นาทีในอ่างควบคุมอุณหภูมิ |
| GT-3 | 0.8 mL | 0.8 mL | 0.8 mL | 0.8 mL | เขย่า |
| GT-4 | 0.4 mL | 0.4 mL | 0.4 mL | 0.4 mL | เขย่า |
| GT-5 | 0.4 mL | 0.4 mL | 0.4 mL | 0.4 mL | เขย่า อ่านผล แทนความหมายเป็นตัวเลข |

การอ่านผล

การอ่านค่าระดับความเป็นพิษจะอ่านค่าได้โดยเทียบความเข้มของสีของหลอดตัวอย่างที่ทดสอบกับหลอดควบคุม(ไม่พบสารพิษ) และหลอดตัดสินในระดับต่างๆ คือ +1, +2, +3, +4 และ +5 และถือเอาตัวอย่างผลิตภัณฑ์ที่มีปริมาณสารพิษที่ยับยั้งการทำงานของเอ็นไซม์ตั้งแต่ 30% ขึ้นไป (ตั้งแต่ +3 ขึ้นไป) เป็นตัวอย่างที่ไม่ปลอดภัยต่อผู้บริโภค ดังตารางที่ 4

ตาราง 4 การอ่านผล

| ผลทดสอบ | การอ่านผล | แปลความหมาย |
|---|--|---|
| หลอดตัวอย่างมีสีจางกว่า หรือสีเท่ากับหลอดควบคุม negative control | บันทึกผลเป็นตัวเลข 0 | ตรวจไม่พบสารพิษตกค้าง |
| หลอดตัวอย่างมีสีเข้มกว่าหลอดควบคุม และสีอ่อนกว่าหลอดตัดสิน (Judgment) | บันทึกผลเป็นตัวเลขตามระดับที่พบ เช่น +1, +2 เป็นต้น | ตรวจพบสารพิษตกค้างในปริมาณที่ปลอดภัย |
| หลอดตัวอย่างมีสีเท่ากับหรือมีสีเข้มกว่าหลอดตัดสิน (Judgment) | บันทึกผลเป็นตัวเลขตามระดับที่พบ เช่น +3, >+3, +5 เป็นต้น | ตรวจพบสารพิษตกค้างในปริมาณที่ไม่ปลอดภัย |



ภาพที่ 2 การอ่านผลการทดสอบ โดยการเทียบความเข้มของสีของสารละลายในหลอด

1.2 วัตถุประสงค์ของการทดลอง และขอบเขตของการวิจัย

วัตถุประสงค์ของการทดลอง

พัฒนาเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ สำหรับวิเคราะห์หาสารพิษตกค้างด้วยชุดทดสอบ GT Pesticide test kit ระบบอัตโนมัติ ใช้สำหรับกระบวนการควบแน่นยา GT-1 ถึง GT-5 ตามปริมาณที่กำหนดซึ่งสามารถปรับค่าได้ เติมลงในหลอดตัวอย่าง มีระบบการตั้งเวลาทำงาน และมีระบบควบคุมอุณหภูมิหลอดทดลอง

ขอบเขตของการวิจัย

ออกแบบ ประดิษฐ์ ทดสอบการใช้งาน เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ ในขั้นตอนวิเคราะห์สารตกค้างด้วยชุดสารพิษตกค้างในอาหารประเภทผักสดและผลไม้สด ด้วยชุดตรวจหายาฆ่าแมลง ในอาหาร GT ซึ่งโดยปกติจะใช้บุคลากรเป็นผู้เติมน้ำยา GT การออกแบบเครื่องให้สามารถควบคุมเปิดในการดูดจ่ายน้ำยา GT 1-5 แทนคนโดยอัตโนมัติ ให้ผลการวัดที่ถูกต้องแม่นยำเที่ยงตรง ป้องกันการผิดพลาดหลงลืม ทำการแก้ไขปรับปรุงให้มีประสิทธิภาพสูงสุด ประเมินความพึงพอใจของเครื่องที่พัฒนาขึ้น และประเมินประสิทธิภาพ ด้านความถูกต้องแม่นยำ

กลุ่มตัวอย่างที่ใช้ในการวัดประสิทธิภาพของเครื่องมือและความพึงพอใจของผู้ใช้งาน โดยนำเครื่องมือที่ประดิษฐ์ขึ้นไปให้เจ้าหน้าที่ และนักวิทยาศาสตร์ ห้องปฏิบัติการวิเคราะห์สารตกค้าง อย่างน้อย 10 คน โดยฝึกวิธีการใช้ ทำการทดสอบในตัวอย่างจริงและรวบรวมผลการใช้ปัญหาอุปสรรค ตลอดจนทดสอบผลการปฏิบัติงานเป็นที่น่าพอใจของผู้ใช้ รายละเอียดแสดงดังตารางที่ 1

ตาราง 5 ขอบเขตของงานวิจัย

| ขอบเขตการศึกษา | วิธีดำเนินการ | ตัวชี้วัด |
|---|--|---|
| 1.ออกแบบ ประดิษฐ์ ทดสอบการใช้งาน เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ | ชุด servo motor, stepper motor และ motor driver ชุดรางขับเคลื่อน สายพาน อุปกรณ์เชื่อมต่อ ก๊อปปี้ หมุน เลื่อน ส่วนยึดจับ บอร์ดคอนโทรลสำเร็จรูป พร้อมไมโครชิพ micro processor ชุดควบคุมแรงดัน จ่ายไฟฟ้าและระบบอิเล็กทรอนิกส์ สวิทช์ สายไฟ จุดเชื่อมต่อไฟฟ้า โครงสร้างตัวถัง ฐานรองรับ เสา คาน ฝาครอบ ไมโคร ไปเปิด ใช้ชุดจ่ายของเหลว ระบบเขย่า ระบบให้ความร้อนและควบคุมอุณหภูมิ ระบบปุ่มควบคุมและแสดงผล | เครื่องต้นแบบ 1 เครื่อง สามารถดูดจ่ายน้ำยา GT-1 ถึง GT-5 เติมลงในหลอดตัวอย่างตามปริมาตรที่กำหนด สามารถเขย่าหลอด ตั้งเวลาทำงาน สามารถควบคุมอุณหภูมิหลอดทดลอง ตามที่กำหนดได้ |
| 2.ทดสอบการใช้งาน | ประเมินประสิทธิภาพตรวจวิเคราะห์ของเครื่องที่พัฒนาขึ้น ด้านความถูกต้องแม่นยำ เก็บข้อมูลทางสถิติจากการทดสอบซ้ำ (replication) ความถูกต้อง: ปริมาตร ตำแหน่ง เวลา อุณหภูมิ ความแม่นยำ:การทำซ้ำให้ผลเหมือนเดิม RSD ต่ำ | ทดสอบอย่างน้อย 100 testing data, Error = 0% |
| 3.นำมาปรับปรุงแก้ไขประสิทธิภาพการทำงาน | นำผลการประเมินประสิทธิภาพมาปรับปรุง | ผลการปรับปรุงสำเร็จ |

| | | |
|---|---|--|
| 4. วัดประสิทธิภาพของเครื่องมือและทดสอบความพึงพอใจของผู้ใช้งานเบื้องต้น | โดยนำเครื่องมือที่ประดิษฐ์ขึ้นไปให้เจ้าหน้าที่และนักวิทยาศาสตร์ ห้องปฏิบัติการวิเคราะห์สารตกค้าง อย่างน้อยจำนวน 15 คน โดยศึกษาวิธีการใช้ ทำการทดสอบในตัวอย่างจริงและรวบรวมผลการใช้ปัญหาอุปสรรคตลอดจนทดสอบผลการปฏิบัติ | รวบรวมผลการใช้งาน ปัญหาอุปสรรค อย่างน้อย 15 ข้อมูล |
| 5. เก็บ ข้อมูลผลการวัดประสิทธิภาพของเครื่องมือและความพึงพอใจเบื้องต้นของผู้ใช้งาน นำมาปรับปรุงแก้ไขเครื่องมือ | นำผลการประเมินประสิทธิภาพมาปรับปรุง | ผลการปรับปรุงสำเร็จ |
| 6. ปรับปรุงเครื่องมือให้มีประสิทธิภาพสูงสุดตามผลการทดสอบ และผลการสำรวจความพึงพอใจ | ได้เครื่องต้นแบบที่ผ่านการทดสอบประสิทธิภาพ ความถูกต้อง แม่นยำและผ่านการประเมิน และปรับปรุงแล้ว | คะแนนต้องอยู่ในระดับ 3.51 มากขึ้นไป |

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- ได้เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ ใช้สำหรับกระบวนการคัดจ่ายน้ำยา ชุดตรวจหายาฆ่าแมลง GT-1 ถึง GT-5 เดิมลงในหลอดตัวอย่าง โดยการทำงานมีความถูกต้อง แม่นยำ ตามปริมาตรที่กำหนด โดยสามารถเขย่าหลอด ตั้งเวลาทำงานในการเติมน้ำยาแต่ละชนิด สามารถควบคุมอุณหภูมิหลอดทดลอง มีประสิทธิภาพสูงกว่าทำงานด้วยคนทำงานแทนคนได้อย่างสมบูรณ์และสามารถนำไปประยุกต์ใช้ในการปฏิบัติงานวิเคราะห์สารตกค้างได้อย่างมีประสิทธิภาพ เป็นที่พึงพอใจของผู้ใช้งาน สามารถนำไปขยายผลโดยผลิตเพิ่มตามเครื่องต้นแบบ ได้รายละเอียดแสดงดังตารางที่ 6

ตาราง 6 ผลผลิต (Out put) และตัวชี้วัดของโครงการวิจัย

| ผลผลิต | ตัวชี้วัด | | | |
|--|---|--|------|--|
| | เชิงปริมาณ | เชิงคุณภาพ | เวลา | ต้นทุน |
| เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ | เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ 1 ชนิดเครื่องมือ | สามารถคัดจ่ายน้ำยา GT-1 ถึง GT-5 เดิมลงในหลอดตัวอย่าง ตามปริมาตรที่กำหนด สามารถเขย่าหลอด ตั้งเวลาทำงาน สามารถควบคุมอุณหภูมิหลอดทดลอง ทำงานแทนคนได้อย่างสมบูรณ์ | - | สามารถตั้งค่าให้ลดสัดส่วนน้ำยาที่ใช้ลงอย่างน้อยครั้งหนึ่ง ทำให้ต้นทุนลดลงอย่างน้อย 50% |

ตาราง 7 ผลลัพธ์ (Outcome) และตัวชี้วัดผลลัพธ์

| ผลผลิต | ตัวชี้วัด | | | |
|--|---------------|--|---|--|
| | เชิงปริมาณ | เชิงคุณภาพ | เวลา | ต้นทุน |
| เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ 1 ชนิด เครื่องมือ | Human Error=0 | มีความถูกต้องแม่นยำสูง ปราศจาก Human Error มีความเชื่อมั่น น่าเชื่อถือ ลดความลังเลสงสัยในผล | ลดเวลาจากการทำงานผิดพลาด ต้องทำงานซ้ำจาก Human Error =0 | ลดต้นทุนจากการทำงานผิดพลาด ต้องทำงานซ้ำจาก Human Error=0 |



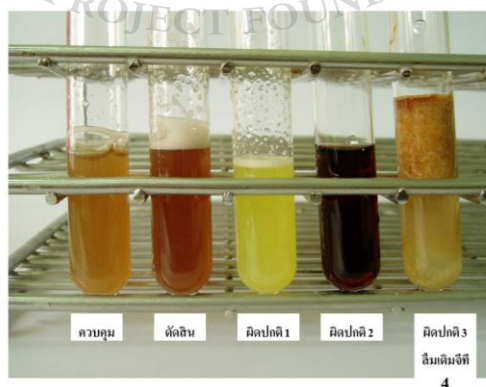
บทที่ 2

ทฤษฎี และแนวคิดที่เกี่ยวข้อง

2.1 ทฤษฎีหรือแนวความคิดที่ใช้ในการแก้ปัญหา

ชุด น้ำยาตรวจสอบยาฆ่าแมลงตกค้าง “GT pesticide test kit” ถูกคิดค้น โดยกรมวิทยาศาสตร์การแพทย์ ใช้ในการวิเคราะห์สารพิษตกค้างและคัดกรองตัวอย่างที่ไม่ปลอดภัยจากสารพิษโดยรวมที่มีอยู่ในตัวอย่างที่มีผลยับยั้งการทำงานของเอนไซม์อะซิติลโคลีนเอสเตอเรส ขั้นตอนที่สำคัญคือการวางหลอดทดลองในน้ำอุ่นเพื่อให้เอนไซม์ทำงานที่ 32-36°C เติมน้ำยาตามขั้นตอน จำนวน 5 ชนิด รอเวลา 1 ชั่วโมง มักจะเกิดความผิดพลาดจากผู้วิเคราะห์ (Human Error) ซึ่งรูปแบบของความผิดพลาดมักจะเป็นการพลั้งเผลอ (Slips) และการลืมนึก (Lapses), การทำผิด (Mistakes) ทำให้ผิดขั้นตอน หรือละเอียดไม่ได้ปฏิบัติตามขั้นตอน ตัวอย่างเช่น หากมีการเติมน้ำยาแต่ละชนิด น้อยหรือมากเกินไป การลืมนึก การเติมน้ำในหลอดเติมน้ำยาผิด ลำดับการเติมน้ำยาผิด หรือเติมในระยะเวลาที่ไม่ตรงกับที่กำหนดไว้เช่นเลยเวลา หรือเร่งรัดเวลาให้เร็วขึ้น เทคนิคการดูค่าที่ไม่ถูกทำให้ปริมาณมากหรือน้อยกว่าปกติ จะทำให้ผลการวิเคราะห์มีสิ่งผิดพลาด การอ่านผลและเทียบสีจะยากและมีความคลาดเคลื่อนสูง ทั้งนี้เนื่องจากจะมีการทำงานอย่างอื่นร่วมด้วยหรือสภาพอย่างอื่นมาหั่นความสนใจจนไม่สามารถจะทำงานได้ถูกต้อง เมื่อผลการวิเคราะห์ผิดพลาดก็จะแก้ไขไม่ทันซึ่งมีผลกระทบต่อคุณภาพของงาน ซึ่งยังมีกระบวนการมากขึ้นตอนโอกาสผิดพลาดย่อมมีมากขึ้นตามไปด้วย

ความผิดพลาดที่เกิดจากการตรวจ



ภาพที่ 3 รูปแบบของความผิดพลาดจาก Human Error ในการเติมน้ำยาวิเคราะห์สารตกค้าง GT

การที่มีการพัฒนาเครื่องมือระบบอัตโนมัติที่สามารถทำงานแทนคนในกระบวนการควบน้ำยา GT-1 ถึง GT-5 เดิมลงในหลอดตัวอย่างตรงตามปริมาตร ตามอุณหภูมิ และตามเวลาที่กำหนด กระบวนการทำงานเป็นมาตรฐานเดียวกัน โดยไม่เกิดความคลาดเคลื่อน ไม่ผิดพลาดหลงลืม จะทำให้ผลการตรวจมีความถูกต้องแม่นยำ สร้างความเชื่อมั่นไม่ลังเลสงสัยในผลการวิเคราะห์ ไม่เป็นภาระกังวลให้ผู้ปฏิบัติงานรวมถึงยังใช้เวลาที่เหลือจากขั้นตอนดังกล่าวไปปฏิบัติอื่นๆ เพิ่มขึ้นและมีประสิทธิภาพมากขึ้น รวมถึงลดค่าใช้จ่ายในส่วนของการทำซ้ำกรณีเกิดความลังเลสงสัยจากการหลงลืมหรือทำผิดพลาด นอกจากนี้การใช้ระบบอัตโนมัติสามารถลดสัดส่วนของการใช้น้ำยาลง เช่น ลดลง 50% ทำให้สามารถลดต้นทุนการตรวจวิเคราะห์เป็นการประหยัดค่าใช้จ่ายในระยะยาวได้อย่างมาก

ในการปฏิบัติงานวิเคราะห์สารตกค้าง สิ่งสำคัญคือต้องเชื่อมั่นว่ามีความถูกต้องแม่นยำ ปราศจากความผิดพลาดในการตรวจวิเคราะห์ แต่อย่างไรก็ตาม การปฏิบัติมีหลายขั้นตอน มีกระบวนการที่ซับซ้อน มีความจำเพาะด้านปริมาตร อุณหภูมิและเวลา ต้องอาศัยความชำนาญของผู้ปฏิบัติงาน และก็มีมักจะพบปัญหาการเกิดความลังเลสงสัยในผลเนื่องจากการหลงลืม และต้องทำการทดสอบซ้ำในกรณีที่ทำผิดพลาด (ERROR) หรือไม่มั่นใจในผลทำให้เสียเวลาและค่าใช้จ่าย ชนิดของความผิดพลาด (Type of Error)

การทำผิดหรือพลั้งเผลอของมนุษย์ ว่าเป็นความผิดพลาดของมนุษย์ (Human Error) โดยมีการจำกัดความในศัพท์เฉพาะ Dependability (ความน่าเชื่อถือ) JIS Z8115 ไว้ว่าหมายถึง “ การกระทำของมนุษย์ที่ก่อให้เกิดผลที่ไม่ได้ตั้งใจ” กล่าวคือ ความผิดพลาดของมนุษย์ คือ การเกิดผลที่ไม่ได้ตั้งใจ ได้แก่ อุบัติเหตุปัญหาที่มีสาเหตุจากการกระทำของมนุษย์ เช่น ทำผิดพลาด ซึ่งความผิดพลาดของการปฏิบัติงาน แบ่งออกได้เป็น 3 แบบหลักๆ คือ

- 1.ความผิดพลาดเชิงระบบ (Systematic Error) ซึ่งเกิดจากเครื่องมือ สารเคมี วัสดุดิบ มาตรฐาน สภาวะแวดล้อม
- 2.ความผิดพลาดแบบสุ่ม (Random Error) เกิดจากความแปรปรวนของการทำซ้ำ เกิดความไม่แน่นอนของผลการทดสอบ
- 3.ความผิดพลาดจากบุคคล (Human Error) ปόอยครั้ง ที่เมื่อเกิดความผิดพลาดขึ้น สาเหตุเกิดจากความผิดพลาดของคน ซึ่งเป็นความผิดพลาดที่เกิดจากการกระทำของคนที่ทำมากกว่าหรือน้อยกว่าระดับที่สามารถยอมรับได้ของระบบ ลักษณะของความผิดพลาดที่เกิดจากคน รูปแบบความผิดพลาดของมนุษย์ขั้นพื้นฐาน (Basic Error Types) โดยใช้กรอบโมเดล GEMS : Generic Error Modelling Systems ของ Professor

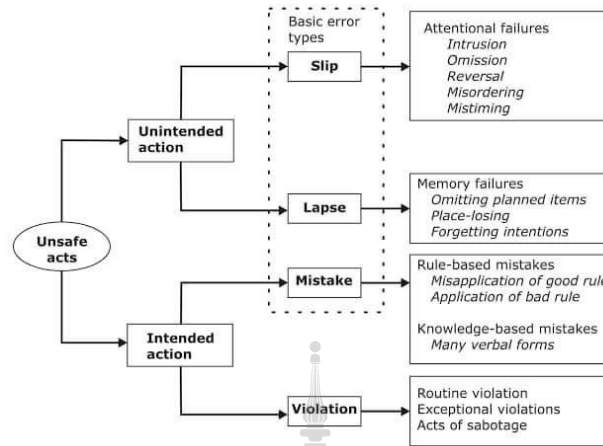
James Reason Reason (J. *Human error*. New York: Cambridge University Press; 1990.) จะแบ่งประเภท ความผิดพลาดของมนุษย์ออกได้ 2 ประเภท ประกอบด้วย

1. ความผิดพลาดที่ไม่ตั้งใจ (**unintended action**) เกิดจากการทำหรือไม่ทำโดยไม่ได้มีการคิดไว้ล่วงหน้า เช่น ซึ่งรูปแบบของความผิดพลาดมักจะเป็นการพลั้งเผลอ (Slips) เป็นความผิดพลาดของการปฏิบัติงานใดๆที่มีความคุ้นเคยกับงานนั้นเป็นอย่างดี มีความชำนาญในระดับทักษะ แต่บางครั้งปฏิบัติโดยไม่ได้ตั้งใจและไม่รู้ตัว เผลอเรอ ขาดสติชั่วขณะ เช่น ตั้งใจที่จะเหยียบคันเร่งไปข้างหน้า แต่กลับเหยียบขึ้นเร่งถอยหลัง เป็นความผิดพลาดจากความบกพร่องของความตั้งใจและรู้ตัว(Attention Failure). การหลงลืม (Lapses) เป็นความผิดพลาดจากความบกพร่องของความจำ (Memory Failure) ลืมที่จะปฏิบัติต่อไป หรือลืมการปฏิบัติก่อนหน้า เช่น ลืมปิดฝา ลืมจดบันทึก ลืมตรวจสอบระบบ เป็นต้น.

2. ความผิดพลาดที่จงใจให้เกิด (**intended action**) เป็นความผิดพลาดที่เกิดจากการทำ หรือไม่ทำ โดยที่พนักงานเชื่อว่าเป็นการตัดสินใจที่ถูกต้อง ปฏิบัติในสิ่งที่ตนเชื่อ หรืออยากทดลองทำที่ต่างไปจากเดิม ตัวอย่าง เช่น การเร่งอุณหภูมิ ลดเวลา เพิ่มหรือลดปริมาณ เพิ่มรอบ การกะเผื่อ ทำให้ผิดขั้นตอน หรือละเลยไม่ได้ปฏิบัติตามขั้นตอน ตัวอย่างเช่น หากมีการเติมน้ำยาแต่ละชนิด น้อยหรือมากเกินไป หรือเติมในระยะเวลาที่ไม่ตรงกับที่กำหนดไว้เช่นเลยเวลา หรือเร่งรัดเวลาให้เร็วขึ้น จะทำให้ผลการวิเคราะห์มีสิ่งที่ผิดพลาด การอ่านผลและเทียบสีจะยากและมีความคลาดเคลื่อนสูง ทั้งนี้อาจเนื่องจากการทำงานอย่างอื่นร่วมด้วยหรือสภาพอย่างอื่นมาหันเหความสนใจจนไม่สามารถจะทำงานได้ถูกต้อง เมื่อผลการวิเคราะห์ผิดพลาดก็จะแก้ไขไม่ทันซึ่งมีผลกระทบต่อคุณภาพของงาน ซึ่งยังมีกระบวนการมากขึ้นตอน โอกาสผิดพลาดมีมากขึ้นตามไปด้วย

การทำผิดเนื่องจากความเข้าใจที่ผิด (Mistakes) เป็นการปฏิบัติโดยตั้งใจและรู้ตัว (Intended) แต่ผลของการปฏิบัตินั้นไปเป็นตามแผนที่ตั้งใจ การทำผิดพลาด (Mistake) เป็นความผิดพลาดที่เกี่ยวกับการวางแผน (Planning Failure) การวิเคราะห์ การวินิจฉัยปัญหาและการตัดสินใจ แบ่งได้อีก 2 แบบ คือ 1 ความผิดพลาดบนพื้นฐานของการใช้กฎระเบียบวิธีปฏิบัติ (Rule-based) สำหรับงานที่มี Procedure รองรับ แต่ใช้ Procedure ที่ไม่ถูกต้อง ไม่เหมาะสมกับสถานการณ์ 2 ความผิดพลาดบนพื้นฐานของความรู้ (Knowledge-based) เป็นการทำผิดหรือตัดสินใจผิดพลาดอันเนื่องมาจากการขาดความรู้และความเข้าใจที่เพียงพอ ซึ่งจะเกิดขึ้นกับงานหรือสถานการณ์ที่ไม่มีความคุ้นเคยมาก่อนและไม่มี Procedure รองรับ

การละเมิดกฎ (Violations) เป็นการปฏิบัติละเมิดฝืนทำผิดโดยตั้งใจและรู้ตัว (Intended) การละเมิดไม่ใช้ความผิดพลาดของมนุษย์โดยธรรมชาติ



ภาพที่ 4 ความผิดพลาดจากการกระทำของมนุษย์

ปัจจัยที่มีผลกระทบต่อการเกิดความผิดพลาดจากบุคคล (Human Error) สามารถแบ่ง ออกได้เป็น 3 ประเภท ประกอบด้วย

1. ปัจจัยภายในเกิดจากองค์ประกอบที่เกิดขึ้นภายในตัวของพนักงานเอง ซึ่งจะส่งผลต่อการทำงานของพนักงาน ได้แก่ การฝึกอบรม ทักษะ ประสบการณ์ แรงจูงใจ ทัศนคติในการทำงาน สภาพทางอารมณ์ วัฒนธรรม บุคลิกภาพ ความรู้ในงาน สุขภาพร่างกาย ชาติกำเนิด อิทธิพลจากสังคมและครอบครัว

2. ปัจจัยภายนอก 2 กลุ่มหลักๆ ได้แก่

2.1 ลักษณะของปัจจัยแวดล้อม เช่น สภาพแวดล้อม เช่น อุณหภูมิ ความชื้น คุณภาพอากาศ แสงสว่าง เสียง ความสั่นสะเทือน ความสะอาด ชั่วโมงการทำงาน ชั่วโมงพัก การหมุนเวียนกะการผลิต ความพร้อมใช้ของเครื่องมือพิเศษ อุปกรณ์ ระดับของพนักงาน การจัดโครงสร้างขององค์กร อำนาจหน้าที่ความรับผิดชอบ และการสื่อสาร นโยบายการบริหารงาน

2.2 ภารกิจ เครื่องมือ และวิธีการทำงาน เช่น วิธีการทำงานที่เป็นเอกสาร หรือไม่มีการเขียนไว้ การสื่อสารด้วยลายลักษณ์อักษร หรือด้วยวาจา คำเตือน และข้อควรระวัง โครงสร้างการทำงานเป็นทีม และการสื่อสารความต้องการในการรับรู้ โครงสร้างทางกายภาพ การทำงานร่วมกับอุปกรณ์ การออกแบบอุปกรณ์ควบคุม หรืออุปกรณ์ช่วย ความถี่ในการทำงาน

3. ปัจจัยจากความกดดันที่เกิดขึ้นในการทำงาน ซึ่งส่งผลกระทบต่อการทำงานของพนักงาน สามารถแบ่งออกได้เป็น 2 ลักษณะคือ

3.1 ความกดดันต่อจิตใจ ได้แก่ สภาพการทำงานที่เสี่ยงอันตราย ความเร่งรีบในการทำงาน ความผิดพลาดในการทำงาน การถูกลดตำแหน่งหรือความสำคัญในการทำงาน ความคาดหวังในผลการทำงานที่ไม่ตรงกัน การสนับสนุนที่ไม่ถูกต้อง เสี่ยงหรือการเคลื่อนไหวที่รบกวนการทำงาน การไม่ได้รับผลประโยชน์หรือรางวัลจากการทำงาน

3.2 ความกดดันต่อสรีระร่างกาย ได้แก่ ความเครียดเป็นเวลานาน ความเมื่อยล้า ความเจ็บปวด หรือความไม่สะดวกสบาย ความหิวกระหาย ความร้อนที่มากเกินไป ความเจ็บไข้ได้ป่วย การเคลื่อนไหวในพื้นที่จำกัด การเคลื่อนไหวในลักษณะท่าซ้ำ การถูกรบกวนเป็นระยะๆ

แนวทางในการลดความผิดพลาดจากคน

การคัดเลือกคนให้เหมาะกับงาน ปรับลักษณะการทำงาน เครื่องมือ และสภาพแวดล้อมในการทำงาน โดยใช้หลักการดังนี้

1 การใช้หลักการยศาสตร์ (Ergonomics) ออกแบบวิธีการทำงาน เครื่องจักร และสภาพแวดล้อมให้เหมาะสมกับความสามารถและข้อจำกัดของคนทำงาน

2 การจัดทำคู่มือและวิธีการทำงานที่ชัดเจน ถูกต้อง

3 การจัดให้มีการฝึกอบรมและพัฒนาทักษะในการทำงาน ช่วยในการลดความผิดพลาดที่จะเกิดขึ้นจากการทำงาน

4 การสร้างระบบการตรวจจับและแก้ไขความผิดพลาด เช่น วางระบบการทำงานเป็นคู่ (Buddy System) ในการช่วยกันควบคุม ระบบการทวนสอบการทำงานด้วยตนเอง (Self-checking)

5 การตอบสนองต่อความต้องการทางด้านสังคม และจิตวิทยาของพนักงาน เช่น การสร้างแรงจูงใจ

เทคนิคหนึ่งทางด้านปรับปรุงคุณภาพคือ การใช้เทคนิค POKA-YOKE ซึ่งเป็นระบบป้องกันความผิดพลาดในขบวนการผลิตที่เกิดจากคน (Human Error) ที่มีรากฐานมาจากภาษาญี่ปุ่น POKA คือ ความผิดพลาดจากการไม่เอาใจใส่ YOKE คือ ป้องกัน/ไม่ให้เกิด/หลีกเลี่ยง หรือที่เรียกกันเป็นที่แพร่หลายว่า

ระบบป้องกันความผิดพลาดจากการพลั้งเผลอหรือใช้คำว่า Error- Proofing ซึ่งระบบ POKA-YOKE นี้ ควบคุมให้งานในกระบวนการมีความถูกต้องมากที่สุดก่อนที่จะสามารถผ่านไปสู่วกระบวนการต่อไป

อย่างไรก็ตามการจัดการให้ความผิดพลาดจากคนที่ได้กล่าวข้างต้น มีความยุ่งยากหลากหลาย ขั้นตอนและเป็นเพียงลดโอกาสการเกิดเท่านั้น ไม่สามารถกำจัด error ได้ทั้งหมด ซึ่งปัจจุบันมักจะใช้ เทคโนโลยีหรือ IT และระบบการจัดการอัตโนมัติเข้ามาดำเนินการแทนคนเพื่อลด Human Error ออกไป อย่างมีประสิทธิภาพ และมีความสามารถทำให้ของเสียเป็นศูนย์ (Zero-Defect) อย่างเช่น ระบบ Robotic Process Automation (RPA) เป็นการใช้งานซอฟต์แวร์ในการบริหารจัดการข้อมูลแทนมนุษย์ RPA นั้นเป็น ซอฟต์แวร์ที่ใช้ระบบข้อมูลขององค์กรเหมือนอย่างที่มีมนุษย์ใช้ โดยซอฟต์แวร์เหล่านี้ทำหน้าที่เป็นเหมือน หุ่นยนต์ในระบบสายการผลิตที่คอยทำกิจกรรมต่าง ๆ ให้อย่างรวดเร็วและไม่เหน็ดเหนื่อย เหมาะสำหรับการ ทำกิจกรรมหรือกระบวนการทางคอมพิวเตอร์ในรูปแบบของระบบอัตโนมัติ ที่สามารถทำซ้ำได้ มีการ ทำงานแบบ Rule-based งานที่มนุษย์ทำแล้วมีข้อผิดพลาดได้ง่าย หรืองานที่เป็นกิจวัตร

เครื่องดูดจ่ายของเหลวอัตโนมัติ (Automated Liquid Handling Workstations) สำหรับ ห้องปฏิบัติการมีความสามารถในการดูดจ่ายของเหลวตามปริมาตรตามที่ต้องการในช่วง 1-1000 μL ที่มีความถูกต้องแม่นยำที่สูง สำหรับเครื่องดูดจ่ายของเหลวอัตโนมัติ ที่วางจำหน่ายในท้องตลาดมีราคาสูง เนื่องจากนำเข้าจากต่างประเทศ ซึ่งมีราคาประมาณ 57,000 -6,500 ดอลลาร์สหรัฐ หรือ 1,900,000- 2,200,000 บาท



ภาพที่ 5 เครื่อง Beckman Coulter Biomek 4000 Automated Liquid Handling Workstations ที่จำหน่ายใน หน้าเว็บเพจ <https://us.vwr.com/store/product/23725021/biomek-4000-automated-workstation-beckman-coulter>.



ภาพที่ 6 เครื่อง PerkinElmer JANUS G3 Automated Workstation (<https://www.perkinelmer.com/product/janus-varispan-automated-workstation-ajv001>)

การออกแบบระบบ automation ขึ้นมาใช้งานเป็นระบบปฏิบัติการที่ช่วยพัฒนางาน และยังช่วยลดต้นทุน สามารถควบคุมคุณภาพของงานได้อย่างมีประสิทธิภาพ ซึ่งระบบนี้ใช้หน่วยประมวลผลอย่างตั้งโปรแกรม เพื่อจัดการควบคุม สั่งงาน ปฏิบัติการด้วยระบบเซนเซอร์ ทำให้ทำงานด้วยระบบอัตโนมัติ จุดประสงค์หลักของการใช้งานเครื่องมือระบบอัตโนมัติดังนี้

ลดต้นทุน – ออโตเมชันช่วยลดเวลาในการทำงานได้เป็นอย่างดี รวมถึงการอำนวยความสะดวกให้กับแรงงานทำให้ไม่เกิดค่าใช้จ่ายจากการบาดเจ็บหรือ Downtime

เพิ่มคุณภาพ – ระบบออโตเมชันนั้นเป็นสิ่งที่มีความแม่นยำสูง ลดความผิดพลาด สามารถผลิตชิ้นงานที่มีความละเอียดสูงและทำงานในรูปแบบพิเศษที่มนุษย์ไม่อาจทำได้

เพิ่มความสามารถในการผลิต – การทำงานภายใต้เงื่อนไขอันจำกัด เช่น สถานที่ สภาพแวดล้อม ระยะเวลา ต้องมีการบริหารจัดการที่ดีไม่อย่างนั้นจะสูญเสียอย่างมาก เช่น การบริหารจัดการคลังสินค้าที่ดีจะไม่ก่อให้เกิดการส่งชิ้นส่วนที่ล่าช้า หรือการทำงานซ้ำๆ อย่างต่อเนื่องได้โดยไม่ก่อให้เกิดความเสี่ยงจากความเมื่อยล้า

รักษาสุขภาพและความปลอดภัยที่ดี – การใช้งานระบบออโตเมชันสามารถส่งเสริมความปลอดภัยของกระบวนการผลิตได้เป็นอย่างดีด้วยระบบเซนเซอร์ตรวจจับการทำงาน ทดแทนแรงงานมนุษย์ในการปฏิบัติงานที่มีความเสี่ยงต่อสุขภาพและความปลอดภัยของผู้ปฏิบัติงานในพื้นที่ เช่น ระบบหยุดการทำงานของหุ่นยนต์เมื่อมีคนเข้าไปใกล้

ส่วนประกอบของเครื่องมือระบบอัตโนมัติ

ระบบควบคุมอัตโนมัติ (Automation System) โดยใช้เครื่องจักรกลอัตโนมัติ (Automation Machine) ในการปฏิบัติงานในขั้นตอนต่างๆ ตั้งแต่การเตรียมจนถึงขั้นตอนการจัดส่ง (Logistic) หน่วยประมวลผลที่สามารถโปรแกรมได้ หรือ พีแอลซี (PLC : Programmable Logic Controller) และ ไมโครคอนโทรลเลอร์ ทำหน้าที่จัดการควบคุม สั่งงาน รับค่าเซนเซอร์ ต่าง ๆ และกำหนดการทำงานของระบบควบคุมอัตโนมัติ เช่น มอเตอร์ ลูกสูบ ปิดเปิดการจ่ายไฟฟ้า การหมุนการเลื่อน สิ้น สายพานนำส่ง เครื่องจ่าย เครื่องนำเข้า เครื่องบีบ การहारอยรั่ว การตรวจสอบการมองเห็น การไล่แก๊ส การล้าง การควบคุมแรงดันอากาศ และแรงดันน้ำ เอซีมอเตอร์และการควบคุมมอเตอร์ การควบคุมน้ำและความชื้น เป็นต้น ระบบต่างๆเหล่านี้ได้รับการควบคุมผ่านโปรแกรมควบคุม หรือ programmable logic controller (PLCs) หรือ คอมพิวเตอร์ กระบวนการผลิตในงานอุตสาหกรรม ส่วนใหญ่นั้นมักใช้กระบวนการควบคุมที่เป็นแบบอัตโนมัติ โดยมีการควบคุมตัวแปรต่าง ๆ อันได้แก่ อุณหภูมิ, ความดัน, ระดับ, อัตราการไหล, ตำแหน่งการเคลื่อนที่, แรง, น้ำหนัก เป็นต้น ซึ่งถ้าใช้ระบบการควบคุมที่ดีก็จะเป็นการเพิ่มประสิทธิภาพของการผลิตให้มีสินค้าที่คุณภาพดีช่วยลดต้นทุนและประหยัดพลังงาน

การควบคุม หมายถึง วิธีการที่ทำให้เครื่องจักรทำงานได้ถูกต้องตามเงื่อนไขที่กำหนด ซึ่งการควบคุมในงานอุตสาหกรรมทำเพื่อควบคุมเครื่องจักรให้ทำงานในกระบวนการผลิตตรงตามจุดประสงค์ของโรงงานอุตสาหกรรม

การควบคุมอัตโนมัติทางอุตสาหกรรม เป็นการควบคุมเครื่องจักรที่ใช้ในกระบวนการผลิตให้สามารถทำงานแทนมนุษย์ได้ โดยมีความผิดพลาดในการผลิตน้อยลง เวลาการผลิตเร็วขึ้น ลดการใช้แรงงาน และความสิ้นเปลืองของพลังงานลง

การควบคุมลำดับ เป็นการควบคุมขั้นตอนการทำงานของเครื่องจักรการผลิตในงานอุตสาหกรรมให้ทำงานสอดคล้องกันตามขั้นตอนการผลิตของโรงงานอุตสาหกรรม โดยมีอุปกรณ์ตรวจวัด (Sensors) ทำหน้าที่ตรวจสอบการทำงานของเครื่องจักร และอุปกรณ์ควบคุม (Control Devices) ที่ทำหน้าที่ควบคุมเครื่องจักรการผลิตให้ได้ผลผลิตตามต้องการ

การควบคุมในงานอุตสาหกรรมแบ่งตามอุปกรณ์การควบคุม ได้ดังนี้

1. การควบคุมด้วยไฟฟ้า
2. การควบคุมด้วยนิวแมติกส์และไฮดรอลิกส์
3. การควบคุมด้วยอิเล็กทรอนิกส์
4. การควบคุมด้วยคอมพิวเตอร์
5. การควบคุมด้วยโปรแกรมเมเบิลคอนโทรลเลอร์

การควบคุมด้วยโปรแกรมเมเบิลคอนโทรลเลอร์ (Programmable Controllers)

เป็นอุปกรณ์ที่ใช้ควบคุมการทำงานของเครื่องจักร หรือกระบวนการต่าง ๆ โดยใช้โปรแกรมคำสั่งที่ทำหน้าที่เหมือนวงจรรีเลย์ มีส่วนของอินพุตและเอาต์พุตที่สามารถต่อใช้งานได้ทันที ดังนั้นโปรแกรมเมเบิลคอนโทรลเลอร์ จึงถูกใช้ในงานอุตสาหกรรมเพื่อเพิ่มประสิทธิภาพในการทำงานของเครื่องจักร ทำให้เครื่องจักรทำงานได้โดยอัตโนมัติ

ข้อดีของโปรแกรมเมเบิลคอนโทรลเลอร์

1. สิ้นเปลืองเนื้อที่น้อยเพราะมีขนาดเล็ก
2. สามารถใช้ควบคุมเครื่องจักรหรือระบบกระบวนการใด ๆ ก็ได้ ถ้าเลือกขนาดของ PLC ที่เหมาะสม
3. การเปลี่ยนลำดับขั้นตอนหรือเงื่อนไขของการทำงาน ทำได้ตามต้องการเพราะใช้หลักการทาง

โปรแกรม

4. ตัวตั้งเวลา ตัวนับ เป็นซอฟต์แวร์ทำให้กำหนดค่าต่าง ๆ ได้ง่าย สามารถเปลี่ยนแปลงค่าได้ตลอดเวลา โดยไม่ต้องมีฮาร์ดแวร์ร่วมและทำให้มีราคาถูกลง

5. รีเลย์ภายใน (Internal Relay) เป็นซอฟต์แวร์ ทำให้ลดค่าใช้จ่ายในการเดินสาย ลดฮาร์ดแวร์และทำให้ขนาดเล็กลง

6. การติดตั้งทำได้สะดวกและง่าย
7. การขยายระบบให้ใหญ่ขึ้นทำได้โดยง่าย
8. ราคาถูกกว่าระบบรีเลย์
9. ความน่าเชื่อถือดีเพราะเป็นอุปกรณ์สารกึ่งตัวนำ ไม่มีการเดินสายมาก ไม่มีปัญหาเรื่องหน้าสัมผัส
10. มีระบบตรวจสอบหาที่ผิดพลาดด้วยตัวเอง การตรวจสอบแก้ไขเมื่อมีปัญหาจึงทำได้เร็ว
11. ลดการเดินสายยาว ๆ เพราะมีอินพุต-เอาต์พุตแบบบริโมท

12. การบำรุงรักษาทำได้ง่าย
13. เวลาในการทำงานเร็วกว่าระบบรีเลย์
14. มีฟังก์ชันทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร และอื่น ๆ
15. ต่อเข้ากับระบบคอมพิวเตอร์ได้
16. ใช้ได้ในทุกสภาพแวดล้อมของงานอุตสาหกรรม

หลักการออกแบบการควบคุมด้วยไมโครคอนโทรลเลอร์

1. **ขั้นเตรียมการออกแบบ** เป็นขั้นตอนที่ต้องมีการศึกษาการทำงานของระบบที่จะควบคุม โดยกำหนดรายละเอียดของเครื่องที่จะใช้ควบคุม เช่น จำนวนอินพุต เอาต์พุต ขนาดของเครื่อง โปรแกรมที่ใช้ออกแบบ เป็นต้น
2. **ขั้นดำเนินการ** เป็นขั้นตอนที่ต้องจัดทำทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ โดยมีการออกแบบวงจรภายนอกของเครื่อง โปรแกรมเมเบิลคอนโทรลเลอร์ วิธีการติดตั้ง การกำหนดตำแหน่งของหน่วยความจำ การออกแบบโปรแกรมและทดสอบการทำงานของโปรแกรมเบื้องต้น
3. **ขั้นการติดตั้งและทดสอบ** เป็นการทดสอบการทำงานร่วมกันทั้งระบบด้านฮาร์ดแวร์และซอฟต์แวร์ เพื่อหาข้อบกพร่องในการทำงานแล้วนำมาปรับปรุงการทำงานให้เกิดเสถียรภาพดียิ่งขึ้น

7.4.2. สเตปเปอร์มอเตอร์ (Stepper Motor)

สเตปเปอร์มอเตอร์ เป็นมอเตอร์ที่มีการหมุนเป็นขั้น ๆ (Step) ซึ่งสามารถนำไอซีไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก สเตปเปอร์มอเตอร์จะแตกต่างจากมอเตอร์กระแสตรงทั่วไป (DC MOTOR) โดยการทำงานของมอเตอร์กระแสตรงทั่วไปเมื่อให้กระแสไฟฟ้าจะหมุนไปแบบต่อเนื่อง ไม่สามารถหมุนเป็นแบบทีละขั้วได้ ดังนั้นการกำหนดตำแหน่งของมอเตอร์กระแสตรงทั่วไปจึงควบคุมได้ยาก สเตปเปอร์มอเตอร์เป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงานควบคุมการหมุนที่ต้องการกำหนดตำแหน่ง และทิศทางที่แน่นอน การทำงานของสเตปเปอร์มอเตอร์ คือ แกนของมอเตอร์จะหมุนด้วยมุมค่าคงที่เมื่อมีสัญญาณ Pulse มากระตุ้น เช่น จะขับเคลื่อนทีละขั้น ๆ ละ 0.9, 1.8, 5, 7.5, 15 หรือ 50 องศา ซึ่งขึ้นอยู่กับชนิดและการออกแบบ เนื่องจากการควบคุมสเตปเปอร์มอเตอร์สามารถใช้สัญญาณดิจิทัลควบคุมได้โดยตรง และสามารถบังคับทิศทางและความเร็วของแกนหมุน สเตปเปอร์มอเตอร์เป็นที่นิยมมาใช้งานที่ต้องการควบคุมองศาการหมุนของมอเตอร์ ปริ้นเตอร์ (Printer) พล็อตเตอร์ (X-Y Plotter)

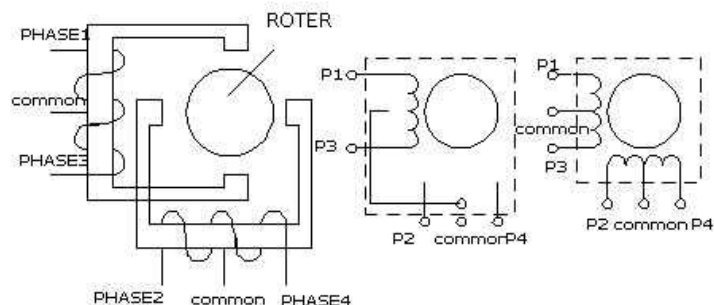
ข้อดีของสเตปเปอร์มอเตอร์เมื่อเปรียบเทียบกับมอเตอร์กระแสตรง (DC MOTOR)

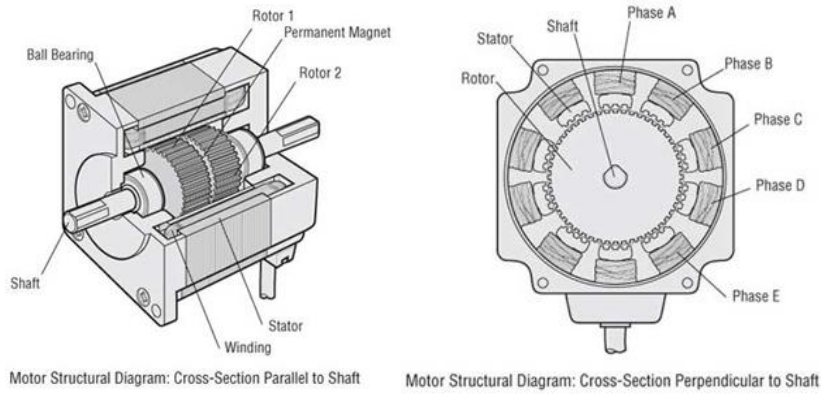
1. หมุนได้ครบ 360 องศา ต่อเนื่อง การควบคุมไม่ต้องอาศัยตัวตรวจจับการหมุน เนื่องจากสามารถกำหนดตำแหน่ง และทิศทางที่แน่นอน การทำงานของ สเตปป์มอเตอร์จะขับเคลื่อนทีละขั้นๆ ละ (Step)
2. ไม่ต้องใช้แปรงถ่าน ดังนั้นจึงทำให้ไม่มีส่วนที่จะต้องสึกหรอ และปัญหาของการสปาร์ค (ที่เกิดจากหน้าสัมผัสของแปรงถ่านแหวนตัวนำใน โรเตอร์) ที่ทำให้เกิดสัญญาณรบกวน
3. การควบคุมโดยทางวงจรดิจิทัลหรือไมโครโพรเซสเซอร์ ทำได้ง่าย และสะดวก



สเตปเปอร์มอเตอร์จะมีส่วนประกอบที่สำคัญ 2 ส่วนด้วยกันคือ

- 1 ส่วนที่ทำการหมุน (Rotor) จะเป็นแม่เหล็กถาวร โดยที่ Rotor 1 เป็นขั้ว N และ Rotor 2 คือ S
- 2 ส่วนที่อยู่กับที่ (winding) เป็นขดลวดที่พันไว้จำนวนหลายๆขด
3. Stator : คือ โลหะหรือวัสดุจำเพาะของผู้ผลิตแต่ละราย

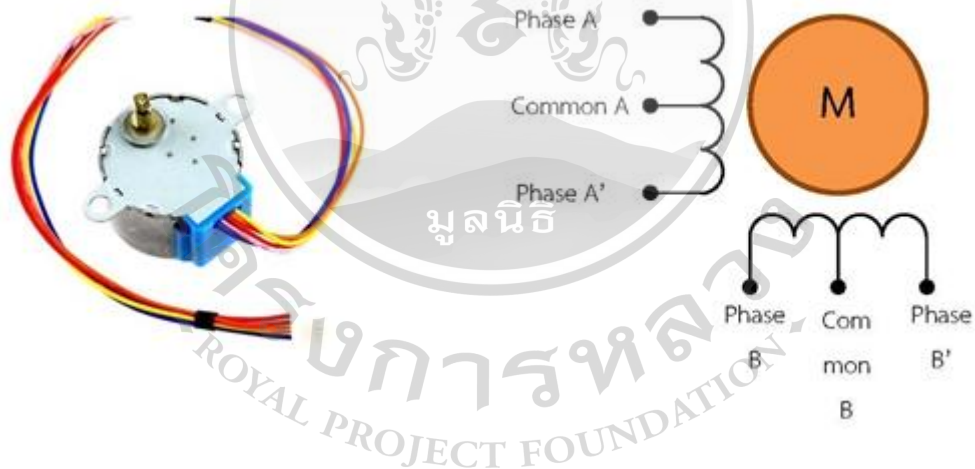




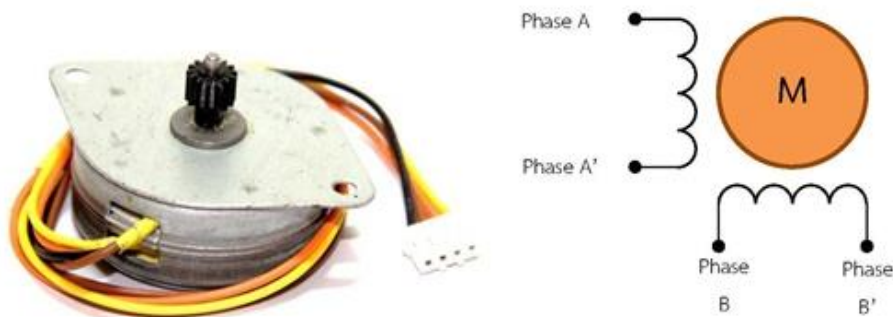
ภาพที่ 8 สเตปป์มอเตอร์ 4 เฟส แบบยูนิโพลาร์ (Uni-polar stepper motor)

วิธีการจับสเตปป์มอเตอร์ให้หมุน โดยการกระตุ้นเฟส

โดยทั่วไป สเตปเปอร์มอเตอร์ (Two-phase Stepper Motor) สามารถแบ่งได้ 2 ประเภท คือ Unipolar และ Bipolar แบบ Unipolar จะมีสายไฟต่อ 5 (Common A และ Common B เชื่อมด้วยกัน) หรือ 6 เส้น และแบบ Bipolar จะมีสายไฟต่อ 4 เส้น เท่านั้น



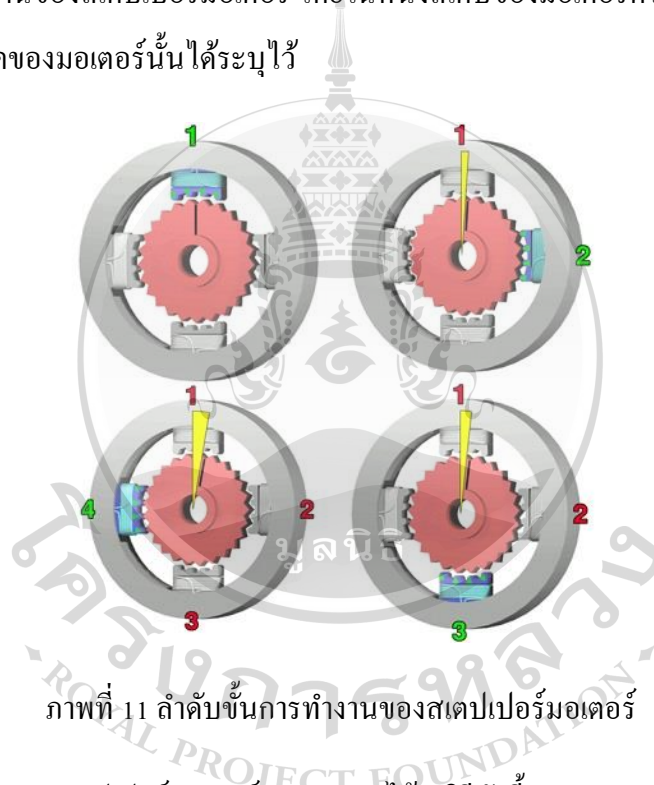
ภาพที่ 9 ลักษณะภายนอกและโครงสร้างภายในของสเตปเปอร์มอเตอร์แบบ Unipolar



ภาพที่ 10 ลักษณะภายนอกและโครงสร้างภายในของสเตปเปอร์มอเตอร์แบบ Bipolar

การควบคุมการหมุนของสเตปเปอร์มอเตอร์แบบ Unipolar สามารถทำได้โดย ควบคุมกระแสไฟที่จ่ายให้กับ ขดลวดในแต่ละเฟส อย่างเป็นลำดับที่แน่นอน โดยถ้าหากเราต้องการให้กระแสไหลในเฟสใดๆ ก็จะทำให้ สถานะของเฟสนั้นๆ เป็นสถานะลอจิกเป็นเลข 1 แทนการบ่งชี้ว่ามีกระแสไฟฟ้าไหลในเฟสนั้น และเลข 0 แทน การไม่มีไฟฟ้าไหล การที่เราจะให้สเตปเปอร์มอเตอร์หมุนนั้นเราต้องจ่ายไฟให้กับขดลวด เอง ซึ่งนั่นก็หมายถึงเพื่อที่ ใช้ทำให้ stator เกิดสนามแม่เหล็กเพื่อดูด Rotor ให้เข้าหาขั้วอย่างถูกต้องกว่าการจ่ายไฟให้ขดลวดแต่ละขด ต้องมีลำดับการจ่ายซึ่งเราเรียก commutation stepper motor

มอเตอร์ที่มีการหมุนเป็นสเต็ป การทำงานของมอเตอร์ชนิดนี้คือจ่ายกระแสไฟไปที่ขดลวดต่างๆตามลำดับ ดังรูป ลำดับขั้นการทำงานของสเตปเปอร์มอเตอร์ โดยในหนึ่งสเต็ปของมอเตอร์ตัวนั้นๆจะเคลื่อนที่ไปเป็น จำนวนองศาตามที่สเป็คของมอเตอร์นั้นได้ระบุไว้



ภาพที่ 11 ลำดับขั้นการทำงานของสเตปเปอร์มอเตอร์

ซึ่งการควบคุมการหมุนของสเตปเปอร์มอเตอร์สามารถทำได้ 3 วิธีดังนี้

1.การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive

เป็นการป้อนกระแสไฟให้กับขดลวดของสเตปเปอร์มอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกัน ไปตามตาราง (ตัวเลข 1 หมายถึง มีการจ่ายแรงดันให้กับ ขดลวด 0 คือไม่จ่ายแรงดันให้กับขดลวด) ดังนั้น กระแสที่ไหลในขดลวดจะทำการไหลในทิศทางเดียวกันทุกขด ลักษณะเช่นนี้จึงทำให้แรงขับของสเตปเปอร์ มอเตอร์มีน้อย จึงไม่เป็นที่นิยมใช้

ตาราง 8 ลำดับการป้อนกระแสแบบเฟสเดียว

| Step | Phase A | Phase B | Phase A' | Phase B' |
|------|----------------|---------|----------|----------|
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 0 | 0 | 0 | 1 |
| 5 | Back to Step 1 | | | |

2.การควบคุมการหมุนแบบสองเฟส(Two Phase) หรือ Full Step Drive

เป็นการป้อนกระแสไฟให้กับขดลวดของสเตปเปอร์มอเตอร์ทั้ง 2 ขด พร้อมๆ กันไป และป้อนกระแสเรียงตามลำดับกันไป ตามตาราง ดังนั้นจึงมีกระแสไหลในขดลวดของมอเตอร์มากขึ้น และจะทำให้มอเตอร์มีแรงบิดมากขึ้น จึงมักนิยมใช้ในงานที่มีลักษณะการหมุนทั่วไปเช่นการเคลื่อนที่เป็นเส้นตรง

ตาราง 9 ลำดับการป้อนกระแสแบบสองเฟส

| Step | Phase A | Phase B | Phase A' | Phase B' |
|------|----------------|---------|----------|----------|
| 1 | 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | Back to Step 1 | | | |

3.การควบคุมการหมุนแบบครึ่งเฟส (Half Step)

เป็นการป้อนกระแสแบบ Two Phase และ Single Phase สลับกันไป ตามตาราง ซึ่งเป็นการเพิ่มความละเอียดของตำแหน่งในการหมุน ใช้ในงานที่ต้องการควบคุมมอเตอร์ให้หมุนตรงตำแหน่งที่กำหนดได้อย่างละเอียด

ตาราง 10 ลำดับการป้อนกระแสแบบครึ่งเฟส

| Step | Phase A | Phase B | Phase A' | Phase B' |
|------|----------------|---------|----------|----------|
| 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 |
| 5 | 0 | 0 | 1 | 0 |
| 6 | 0 | 0 | 1 | 1 |
| 7 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 1 |
| 9 | Back to Step 1 | | | |

ความเร็วการหมุนของแกนสเตปเปอร์มอเตอร์ จะขึ้นอยู่กับการหน่วงเวลา (Time Delay) ในแต่ละลำดับ ถ้า Time Delay มีค่าน้อย แกนจะหมุนเร็วขึ้น ถ้า Time Delay มีค่ามาก แกนจะหมุนช้าและหมุนไม่สม่ำเสมอ การควบคุมอุปกรณ์ขับเคลื่อนเช่นสเตปเปอร์มอเตอร์ จะใช้กระแสขั้วสูงเพื่อสร้างแรงบิด การใช้งานมอเตอร์กระแสตรงกับบอร์ดไมโครคอนโทรลเลอร์ไม่สามารถต่อใช้งานกันโดยตรงได้ เพราะว่าขา GPIO ของตัวบอร์ดไมโครคอนโทรลเลอร์นั้นไม่สามารถจ่ายกระแสออกมาขนาดที่จะทำให้มอเตอร์ทำงานได้ ต้องใช้งานบอร์ดขับเคลื่อนและแหล่งจ่ายไฟเลี้ยงที่เหมาะสมจากภายนอกเพิ่มเข้ามาด้วย โดยปกติแล้วบอร์ดขับเคลื่อน 1 บอร์ดสามารถควบคุมมอเตอร์ได้ 1 ตัว หรือ 2 ตัวตามสเปคของบอร์ดขับเคลื่อนตัวนั้นๆ และการในการควบคุมมอเตอร์ 1 ตัวจะใช้ต้องสายสัญญาณ 3 สายในการควบคุมมอเตอร์ ดังตารางรายละเอียดของขาสัญญาณที่ใช้ในการควบคุมมอเตอร์จึงไม่สามารถต่อกับ Microcontroller ได้โดยตรง ดังนั้นต้องมีอุปกรณ์ขับเคลื่อนเชื่อมต่อระหว่าง Microcontroller กับ สเตปเปอร์มอเตอร์ หรือ บอร์ด driver

ในการที่จะให้สเตปเปอร์มอเตอร์นั้นทำงานก็จะมี 3 องค์ประกอบหลักๆด้วยกันก็คือ

1. ชุดคอนโทรล เช่น areduino ที่ใช้ Grbl หรือ Pc base เช่น Mach3 Linux

ชุดอุปกรณ์ขับเคลื่อนหรือ Motor driver

Stepper moter

การควบคุมตำแหน่งของแกน Stepper Motor มีขั้นตอนการทำงานดังนี้

กำหนดตำแหน่งที่ต้องการ (Desired Step) และตรวจสอบตำแหน่งปัจจุบัน (Current Step) ซึ่งได้มาจาก Subsystem ดังรูป 2-13 ในการทดลองนี้จะกำหนดให้ ทิศทวนเข็มนาฬิกาเป็นทิศ + (บวก) และทิศตามเข็มนาฬิกาเป็นทิศ – (ลบ)

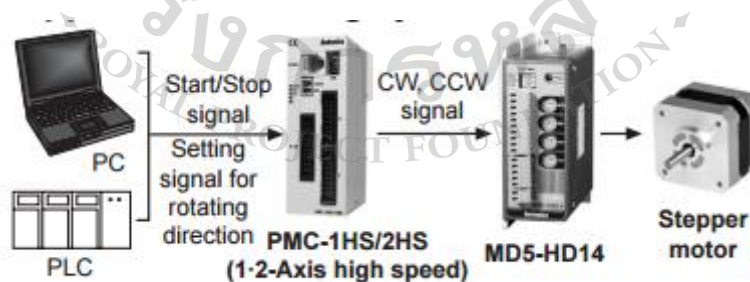
เปรียบเทียบและคำนวณหาจำนวน StepToGo ได้จากสมการ: $StepToGo = Desired Step - Current Step$

สั่ง StepperMotor ให้หมุน จนกว่า StepToGo จะมีค่าเท่ากับศูนย์

จากคุณสมบัติของ Stepper Motor ที่ใช้ในการทดลอง Step Angle: 5.625 องศา / 64 หมายความว่ามอเตอร์ หมุน 1 Step แกนจะหมุน 0.0879 องศา ดังนั้น หากต้องการให้หมุน 90 องศา $Desired Step = 90 / 0.0879 = 1024 Step$

การประยุกต์ใช้งาน Stepping Motor Drive

Stepping Motor Drive (สเตปเปอร์มอเตอร์ไดรฟ์) เป็นอุปกรณ์ที่มีหน้าที่ในการขับเคลื่อนให้สเตปเปอร์ มอเตอร์นั้นสามารถเคลื่อนที่หรือทำงานได้ ซึ่ง Stepping Motor Drive นี้จะรับสัญญาณพัลส์จากตัวควบคุม (Controller) แล้วทำการส่งสัญญาณ ไปขับสเตปเปอร์มอเตอร์ให้ทำการหมุน เคลื่อนที่ตำแหน่งตามที่ต้องการ ได้อย่างแม่นยำ โดยกระบวนการทั้งหมดนี้ จะต้องประกอบไปด้วย 3 อุปกรณ์หลักๆ คือ Controller, Stepping Motor Drive และ Stepping Motor



ภาพที่ 12 แสดงกระบวนการทำงานของ Stepper Motor

ควบคุมสายพานในขณะที่ทำการตรวจจับผลิตภัณฑ์

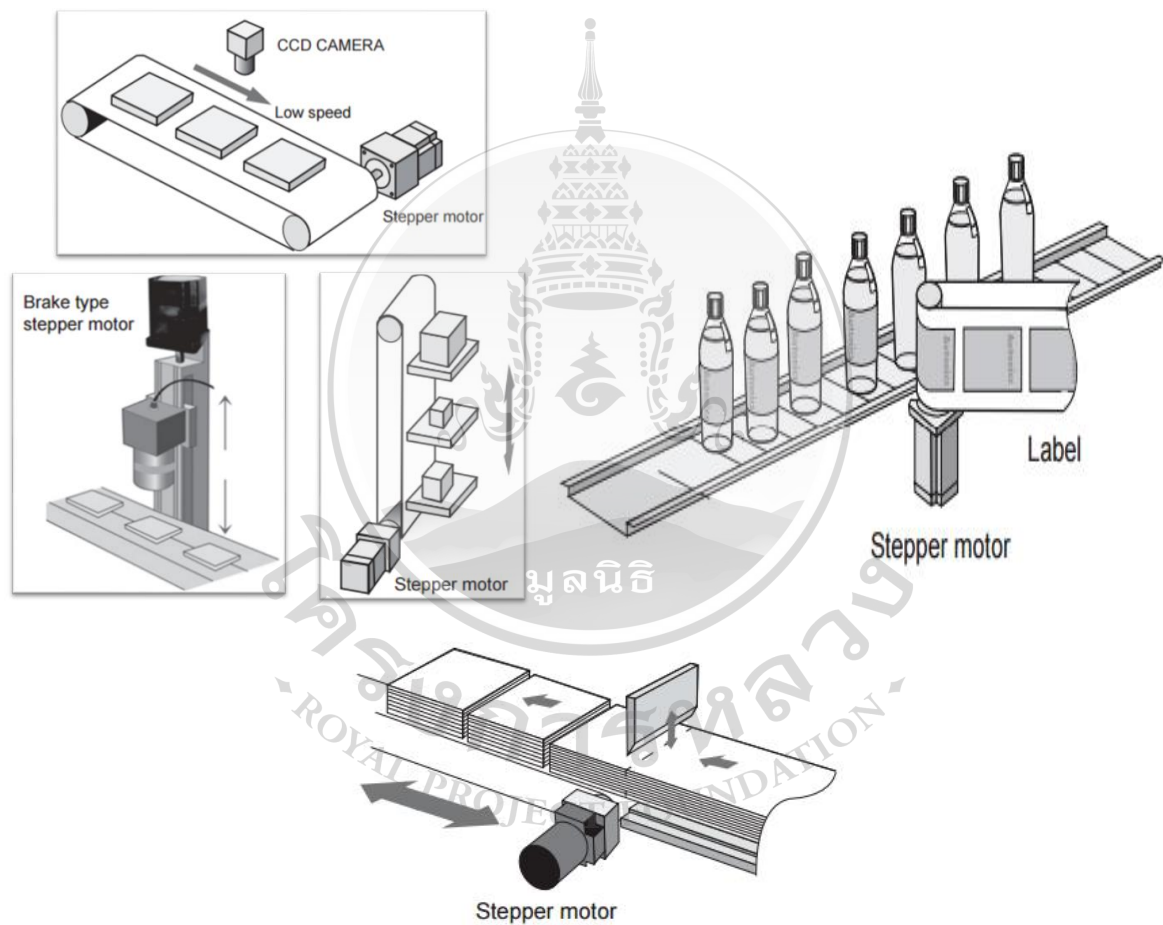
ในอุตสาหกรรมก็ได้นำสเตปเปอร์มอเตอร์ไปใช้งานในหลายๆ ด้าน เช่น นำไปใช้ในการควบคุมสายพานในขณะที่ทำการตรวจจับผลิตภัณฑ์

ควบคุมตำแหน่งในการติดฉลาก

นำไปควบคุมเครื่องจักรที่ใช้ในการติดฉลากบนขวด ซึ่งสามารถควบคุมตำแหน่งในการติดฉลากได้อย่างแม่นยำ

ควบคุมตำแหน่งในการตัดชิ้นส่วนงานต่างๆ

นำสเตปเปอร์มอเตอร์ไปใช้งานในการควบคุมตำแหน่งในการตัดชิ้นส่วนงานต่างๆ ซึ่งให้ความแม่นยำในการตัดสูง



ภาพที่ 13 การประยุกต์ใช้งาน Stepping Motor Drive

RC Servo Motor

RC Servo (คำว่า RC ย่อมาจาก Radio Control) เป็นชุดมอเตอร์ขนาดเล็กที่สามารถหมุนแกนไปยังตำแหน่ง (มุม) ต่างๆ ได้อย่างแน่นอน และโดยปกติสามารถตอบสนองได้เร็วกว่า สเตปเปอร์มอเตอร์ตามคำสั่งที่เป็น สัญญาณไฟฟ้าจากตัวรับสัญญาณวิทยุ รูป 1-3 แสดงส่วนประกอบหลักของ RC Servo

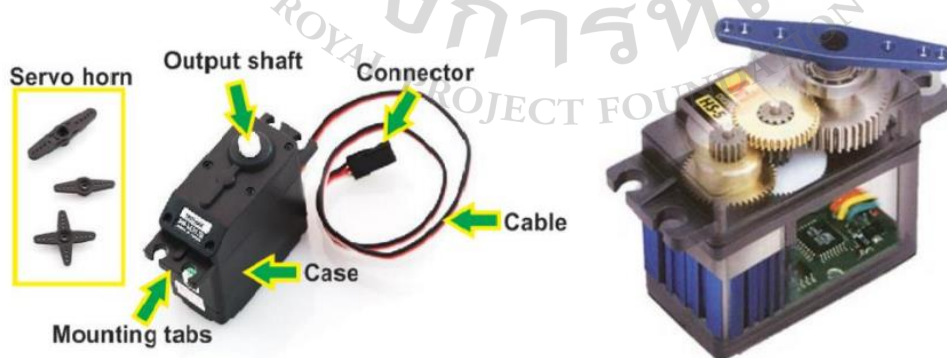


ภาพที่ 14 ส่วนประกอบภายในของ RC Servo

เนื่องจากมีขนาดเล็ก น้ำหนักเบา และมีแรงบิดสูง จึงเป็นที่นิยมใช้ในอุปกรณ์บังคับวิทยุ เช่นการปรับองศาของล้อหน้าในรถบังคับวิทยุ หรือการปรับมุมปีกและหางของเครื่องบินบังคับวิทยุ อย่างไรก็ตามอุปกรณ์ชนิดนี้ ไม่สามารถหมุนได้ต่อเนื่องครบ 360 องศา จึงไม่เหมาะใช้งานขับเคลื่อนแบบล้อ

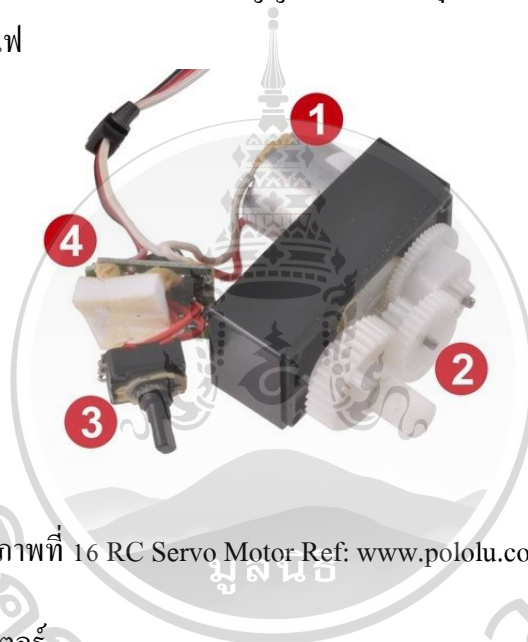
มูลนิธิ

ส่วนประกอบภายนอก RC Servo Motor



ภาพที่ 15 ส่วนประกอบภายนอก RC Servo Motor

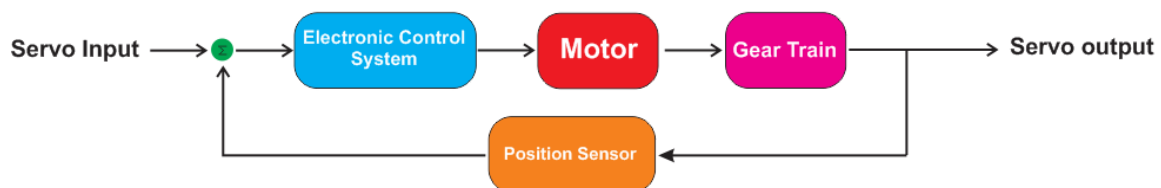
- Case ตัวถัง หรือ กรอบของตัว Servo Motor
- Mounting Tab ส่วนจับยึดตัว Servo กับชิ้นงาน
- Output Shaft เฟลาส่งกำลัง
- Servo Horns ส่วนเชื่อมต่อกับ Output shaft เพื่อสร้างกลไก
- Cable สายเชื่อมต่อเพื่อ จ่ายไฟฟ้า และ ควบคุม Servo Motor จะประกอบด้วยสายไฟ 3 เส้น และ ใน RC Servo Motor จะมีสีของสายแตกต่างกันไปดังนี้
 - o สายสีแดง คือ ไฟเลี้ยง (4.8-6V)
 - o สายสีดำ หรือ น้ำตาล คือ กราวด์
 - o สายสีเหลือง (ส้ม ขาว หรือฟ้า) คือ สายส่งสัญญาณพัลส์ควบคุม (3-5V)
- Connector จุดเชื่อมต่อสายไฟ



ภาพที่ 16 RC Servo Motor Ref: www.pololu.com

1. Motor เป็นส่วนของตัวมอเตอร์
2. Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง
3. Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน
4. Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล

Servo Motor Block Diagram



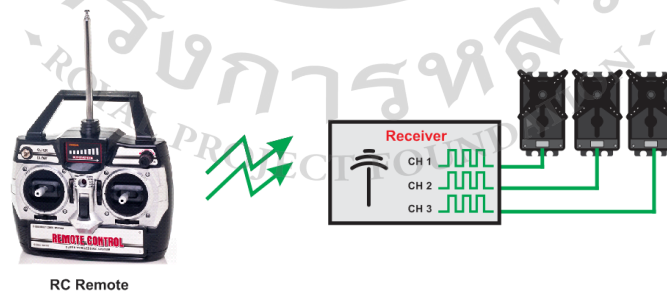
ภาพที่ 17 Servo Motor Block Diagram

หลักการทำงานของ RC Servo Motor

เมื่อจ่ายสัญญาณพัลส์เข้ามายัง RC Servo Motor ส่วนวงจรควบคุม (Electronic Control System) ภายใน Servo จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ Motor หมุนเคลื่อนที่ไปยังตำแหน่งนั้น แล้วส่งคำสั่งไปทำการควบคุมให้ Motor หมุนไปยังตำแหน่งที่ต้องการ โดยมี Position Sensor เป็นตัวเซ็นเซอร์คอยวัดค่ามุมที่ Motor กำลังหมุน เป็น Feedback กลับมาให้อ่านวงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องแม่นยำ เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของมัน (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน สามารถควบคุมเครื่องจักรกล หรือระบบการทำงานนั้นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตำแหน่ง (Position), ระยะทางในการเคลื่อนที่ (หมุน) (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมในลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยำสูง

สัญญาณ RC ในรูปแบบ PWM

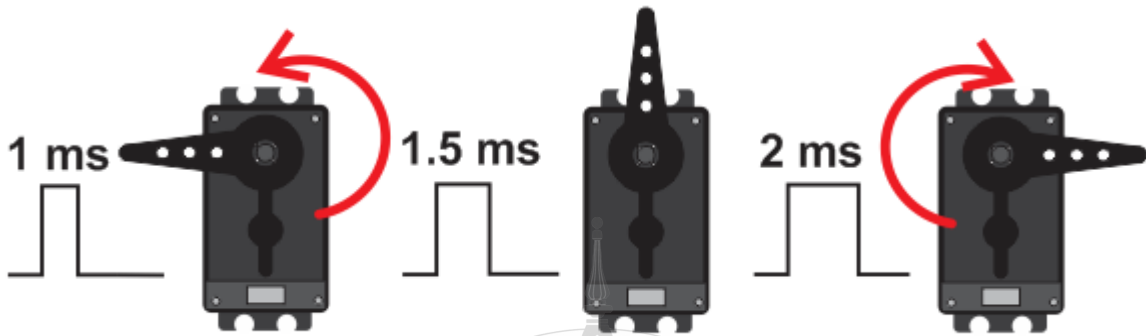
ตัว RC Servo Motor ออกแบบมาใช้สำหรับรับคำสั่งจาก Remote Control ที่ใช้ควบคุมของเล่นด้วยสัญญาณวิทยุต่างๆ เช่น เครื่องบินบังคับ รถบังคับ เรือบังคับ เป็นต้น ซึ่ง Remote จำพวกนี้ที่ภาครับจะแปลงความถี่วิทยุออกมาในรูปแบบสัญญาณ PWM (Pulse Width Modulation)



ภาพที่ 18 Remote Control ที่ใช้ควบคุมมอเตอร์ด้วยสัญญาณวิทยุต่าง

มุมหรือองศาจะขึ้นอยู่กับความกว้างของสัญญาณพัลส์ ซึ่งโดยส่วนมากความกว้างของพัลส์ที่ใช้ใน RC Servo Motor จะอยู่ในช่วง 1-2 ms หรือ 0.5-2.5 ms

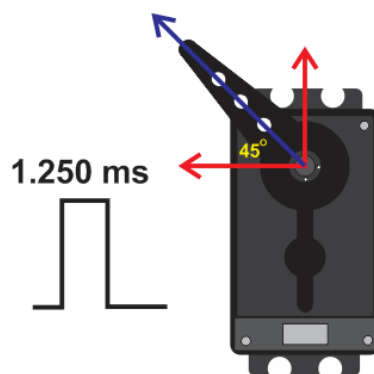
ยกตัวอย่างเช่นหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1 ms ตัว Servo Motor จะหมุนไปทางซ้ายสุด ในทางกลับกันหากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 2 ms ตัว Servo Motor จะหมุนไปยังตำแหน่งขวาสุด แต่หากกำหนดความกว้างของสัญญาณพัลส์ไว้ที่ 1.5 ms ตัว Servo Motor ก็จะหมุนมาอยู่ที่ตำแหน่งตรงกลางพอดี



ภาพที่ 19 องศาการหมุนของ RC Servo Motor

ดังนั้นสามารถกำหนดองศาการหมุนของ RC Servo Motor ได้โดยการเทียบค่า เช่น RC Servo Motor สามารถหมุนได้ 180 องศา โดยที่ 0 องศาใช้ความกว้างพัลส์เท่ากับ 1000 us ที่ 180 องศาความกว้างพัลส์เท่ากับ 2000 us เพราะฉะนั้นค่าที่เปลี่ยนไป 1 องศาจะใช้ความกว้างพัลส์ต่างกัน $(2000-1000)/180$ เท่ากับ 5.55 us

จากการหาค่าความกว้างพัลส์ที่มุม 1 องศาข้างต้น หากต้องกำหนดให้ RC Servo Motor หมุนไปที่มุม 45 องศาจะหาค่าพัลส์ที่ต้องการได้จาก 5.55×45 เท่ากับ 249.75 us แต่ที่มุม 0 องศาเราเริ่มที่ความกว้างพัลส์ 1ms หรือ 1000 us เพราะฉะนั้นความกว้างพัลส์ที่ใช้กำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา คือ $1000 + 249.75$ เท่ากับประมาณ 1250 us



ภาพที่ 20 การกำหนดให้ RC Servo Motor หมุนไปที่ 45 องศา

วิธีควบคุม RC Servo Motor ด้วย Arduino

Arduino มีไลบรารีสำหรับสั่งงาน RC Servo Motor มาให้ใช้งานอยู่แล้วเป็นฟังก์ชันสำเร็จรูปและใช้งานได้ง่าย Servo Library ของ Arduino สามารถสั่งงาน RC Servo Motor ได้ทั้งแบบหมุนไป-กลับได้ 0-180 องศา และแบบต่อเนื่องที่หมุนครบรอบได้เรียกว่าเป็น Continuous Rotation Servo โดยสามารถรองรับการเชื่อมต่อ RC Servo Motor ได้ถึง 12 ตัวกับบอร์ด Arduino UNO และรองรับสูงสุดถึง 48 ตัวหากใช้บอร์ด Arduino Mega

ประยุกต์ใช้ Servo motor

servo motor ถูกประยุกต์ใช้ในโรงงานเช่นพวกหุ่นยนต์ ระบบอัตโนมัติ เครื่องยนต์และการบรรจุหีบห่อ การตอบสนองกลับมามีทำให้ตัวเคลื่อนไหวรู้ถึงตำแหน่ง ความเร็วและแรงเสียดทานเพื่อตรวจพบกับการเคลื่อนไหวที่ไม่ต้องการ อุตสาหกรรมผลิตยานั้นต้องการสร้างอุปกรณ์ที่เล็กกว่าเป็นอย่างมาก อุปกรณ์ที่ง่ายกว่าในการใช้งานและปฏิบัติงานได้อย่างมีประสิทธิภาพมากขึ้น

ทำงาน Servo motor

ระบบกลไกของ เซอร์โวมอเตอร์ที่ว่า ไปนั้นไม่ซับซ้อน เซอร์โวมอเตอร์มีแผงควบคุมและโพเทนโอมิเตอร์ซึ่งต่อเข้ากับท่อแสดงผล โพเทนโอมิเตอร์ถูกส่งสัญญาณจากแผงควบคุมเพื่อที่จะควบคุมมุมของท่อ ที่ไหนก็ได้จาก 0 – 180 องศา บางครั้งก็มากกว่านั้น โพเทนโอมิเตอร์ช่วยในการควบคุมแผงวงจรเพื่อที่จะควบคุม ปัจจุบันของ เซอร์โวมอเตอร์ถ้าท่ออยู่ที่มุมถูกต้อง เซอร์โวมอเตอร์ จะอยู่นิ่งๆจนกระทั่งสัญญาณจุดตำแหน่งต่อไปได้รับ เซอร์โวมอเตอร์ตัวนั้นก็หมุนไปที่จุดที่ถูกต้องจนกระทั่งมุมนั้นถูกต้อง

เซอร์โวมอเตอร์แต่ละชิ้นทำงานด้วยระบบชนิดหนึ่งเรียกว่า พัลส์โค้ดเดด โมดูเลชันหรือพีซีเอ็ม มอเตอร์ตัวนั้นจะมีสายไฟสำหรับควบคุมที่จะให้สัญญาณเป็นระยะเวลาหนึ่งซึ่ง เซอร์โวมอเตอร์คอยที่จะมองหาสัญญาณทุกๆ 2 วินาที เซอร์โวปกตินี้ไม่สามารถหมุนได้มากกว่าปกติเพราะมีกลไกการหยุดที่ตัวเครื่องจักรหลัก พลังที่ให้ไปแก่มอเตอร์นั้นเป็นส่วนที่มันต้องการจะเดินทาง ดังนั้นถ้าท่อของเซอร์โวมอเตอร์ต้องการหมุนที่ระยะไกลๆ เซอร์โวมอเตอร์จะวิ่งที่ความเร็วเต็มที่ ถ้าเซอร์โวมอเตอร์ต้องการที่จะหมุนเล็กๆน้อยๆ เครื่องยนต์จะวิ่งที่ความเร็วช้าลง สิ่งนี้หมายถึงการควบคุมที่เป็นสัดส่วนกัน เซอร์โวมอเตอร์คาดว่าจะเห็นสัญญาณทุกๆ 20 มิลลิวินาที และความยาวของแต่ละสัญญาณจะวัดว่า servo motor จะหมุนไกลได้เท่าไร

โครงสร้างของระบบควบคุมเซอร์โวมอเตอร์

ลักษณะของระบบควบคุมเซอร์โวมอเตอร์จะเป็นระบบควบคุมแบบลูปปิด(Closed loop control) ซึ่งประกอบด้วย 3 โหมดการควบคุมคือ โหมดการควบคุมแรงบิด (Torque Control Mode) ซึ่งอยู่วงรอบหรือลูปในสุด โหมดการควบคุมอัตราเร็ว(Velocity Control Mode) และ โหมดการควบคุมตำแหน่ง(Position Control Mode) ซึ่งอยู่ลูปด้านนอกสุด โดยมีองค์ประกอบที่สำคัญๆดังรูป

องค์ประกอบในการทำงานของเซอร์โวมอเตอร์

การทำงานเพียงตัว Servo Motor เพียงอย่างเดียวนั้นไม่สามารถทำงานได้ การที่จะให้ Servo Motor จะควบคุมลักษณะที่กล่าวมาข้างต้นนั้นต้องมีองค์ประกอบดังนี้

1.คอนโทรลเลอร์ (Controller)

หลักการทำงานหลักๆ หน้าที่ของ Controller คือ มีหน้าที่รับคำสั่งจากผู้ใช้งานว่าต้องการให้ Servo Motor นั้นเคลื่อนที่ด้วยความเร็วเท่าไรและระยะทาง ไกลหรือใกล้แค่ไหน หน้าที่ตรงจุดนี้จะเป็น Controller จะเป็นตัวกำหนดให้กับตัว Servo Motor

2. เซอร์โวไดรฟ์เวอร์ (Servo Driver)

หน้าที่ของ Servo Driver คือ จะรับสัญญาณมาจาก Controller และสั่งการให้กับตัว Servo Motor เคลื่อนที่ตามที่ Controller สั่งการมา แต่ทำไม Controller ไม่สั่งการควบคุมไปที่ Servo Motor โดยตรง เนื่องจาก Servo Driver จะเป็นตัวที่ปรับตั้งค่าของตัว Servo Motor ให้ทำงานตามรูปแบบของการควบคุมไม่ว่า จะเป็นการควบคุม ความเร็ว(Speed Control) , แรงบิด(Torque) และ ตำแหน่ง(Position Control) ตัว Servo Driver จะเป็น ตัวกำหนดค่าตัวแปรหรือพารามิเตอร์ต่างๆ ให้กับตัว Servo Motor ให้ทำงานได้อย่างถูกต้องและแม่นยำ เพราะฉะนั้นเมื่อใช้ Servo Motor ก็จะต้องมี Servo Driver เสมอ



ภาพที่ 21 เซอร์โวไดรฟ์เวอร์ (Servo Driver)

3. เซอร์โวมอเตอร์ (Servo Motor)

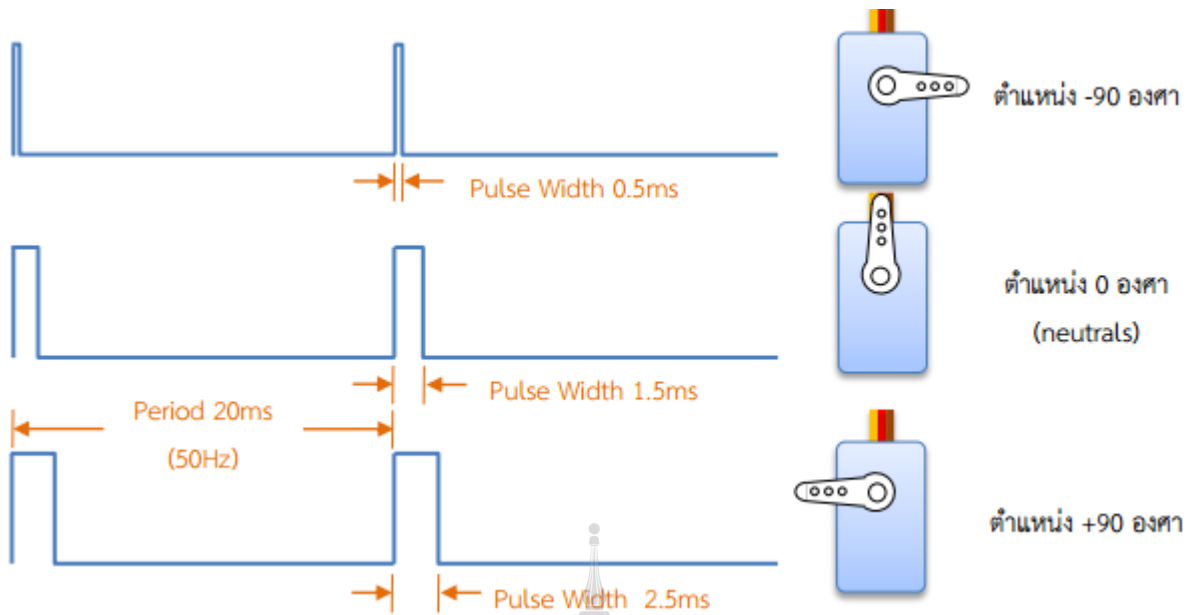
หน้าที่ของเซอร์โวมอเตอร์ คือ ขับเคลื่อนอุปกรณ์ของเครื่องจักรกลหรือระบบของการทำงานนั้นๆ ให้เป็นไปตามรูปแบบที่ได้รับคำสั่งจากตัว Servo Driver พร้อมกับส่งสัญญาณป้อนกลับให้กับตัว Servo Driver ว่าตอนนี้ Servo Motor เคลื่อนที่ด้วย ความเร็วเท่าไรและระยะทางในการเคลื่อนที่เป็นระยะทางเท่าไรแล้ว ด้วยสัญญาณของตัว Encoder ที่อยู่ภายในตัว Servo Motor ทำให้การเคลื่อนที่ของ Servo Motor นั้นมีความแม่นยำสูง



ภาพที่ 22 เซอร์โวมอเตอร์ (Servo Motor)

ด้วยองค์ประกอบข้างต้นทั้งหมดทั้งหมดนั้น พอจะทำให้ผู้ที่ใช้งานหรือผู้ที่กำลังศึกษา พอที่จะมองภาพของการ ทำงานของระบบ Servo Motor ว่าองค์ประกอบของระบบหรือการที่จะใช้งาน Servo Motor นั้นต้องมีองค์ประกอบอะไรบ้างจึงจะใช้งาน Servo Motor ได้อย่างถูกต้องและมีประสิทธิภาพ

RC Servo รับสัญญาณดิจิทัลเพื่อปรับมุมมองของแกน โดยมีลักษณะของสัญญาณเรียกว่า Pulse Width Modulation (PWM) ชุดควบคุมตำแหน่งของ RC Servo จะตรวจสอบช่วงเวลาสัญญาณอยู่ในสถานะ ON หรือ Pulse Width แล้วปรับตำแหน่งตามที่ถูกผลิต RC Servo โปรแกรมไว้ ดังรูป 3-2 แสดงความกว้างโดยประมาณของสัญญาณ ON สำหรับควบคุมตำแหน่งของ RC Servo รุ่น SG90 โดยมี Period ของสัญญาณมีค่า 20 ms หากใช้งาน RC Servo รุ่นอื่นๆ ผู้ใช้งานควรศึกษารายละเอียดเพิ่มเติมจาก Datasheet เช่นระดับแรงดันของสัญญาณ PWM และ ความกว้างของสัญญาณ



ภาพที่ 23 แสดงความกว้างของสัญญาณ ON หรือ %Duty Cycle สำหรับควบคุมตำแหน่งของ RC Servo

วิธีการใช้งาน Basic PWM Block

ในบท Digital Input/Output แสดงการใช้งาน Pulse Generator Block สำหรับสร้างสัญญาณ PWM ซึ่งผู้ใช้งานสามารถตั้งค่า %Duty Cycle ภายใน Block ได้เท่านั้น ในบทนี้จะกล่าวถึงวิธีใช้งาน Basic PWM Block ซึ่งทำหน้าที่สร้างสัญญาณ PWM โดยใช้ Timer ของ STM32F4DISCOVERY ผู้ใช้งานสามารถเลือกใช้งาน Timer และ Channel สำหรับส่งสัญญาณไปยังอุปกรณ์ภายนอก เช่น RC Servo หรือชุดควบคุม DC Motor และกำหนด %Duty Cycle ผ่าน Simulink Model ได้

ฟังก์ชันภายใน Servo Library

- attach()
- write()
- writeMicroseconds()
- read()
- attached()
- detach()

attach()

Description

คือฟังก์ชันที่ใช้ในการกำหนดขาสัญญาณที่ Servo Motor ต่อกับ Arduino และกำหนดความกว้าง

ของพัลส์ที่ 0 องศาและ 180 องศา

Syntax

```
Servo.attach(pin)
```

```
Servo.attach(pin,min,max)
```

Parameters

Pin: คือ ขาสัญญาณของ Arduino ที่ใช้เชื่อมต่อกับ Servo Motor

Min: คือ ความกว้างของพัลส์ที่ 0 องศาของ Servo ตัวที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติแล้ว หากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 544 us

Max: คือ ความกว้างของพัลส์ที่ 180 องศาของ Servo ตัวที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติแล้ว หากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 2400 us

Write()

Description

คือฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ต้องการให้ Servo Motor หมุนไปยังองศาที่กำหนดสามารถกำหนดเป็นค่าองศาได้เลย คือ 0-180 องศา แต่ใน Servo Motor ที่เป็น Full Rotation คำสั่ง write จะเป็นการกำหนดความเร็วในการหมุน โดย

ค่าเท่ากับ 90 คือคำสั่งให้ Servo Motor หยุดหมุน

ค่าเท่ากับ 0 คือการหมุนด้วยความเร็วสูงสุดในทิศทางหนึ่ง

ค่าเท่ากับ 180 คือการหมุนด้วยความเร็วสูงสุดในทิศทางตรงกันข้าม

Syntax

```
servo.write(angle)
```

Parameters

Angle: คือมุมที่ต้องการให้ RC Servo Motor แบบ 0-180 องศาหมุนไป แต่หากเป็น RC Servo Motor แบบ Full Rotation ค่า Angle คือ การกำหนดความเร็วและทิศทางในการหมุน

writeMicroseconds()

Description

คือฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ให้ Servo Motor หมุนไปยังตำแหน่งองศาที่กำหนดโดยกำหนดเป็น

ค่าความกว้างของพัลส์ในหน่วย us ซึ่งปกติแล้ว RC Servo Motor จะใช้ความกว้างของพัลส์อยู่ที่ 1000-2000 us ตามที่ได้กล่าวไปข้างต้นแล้ว แต่ RC Servo Motor บางรุ่นหรือบางยี่ห้อไม่ได้ใช้ ช่วงความกว้างของพัลส์ตามที่ได้กล่าวเอาไว้เนี่ย อาจจะใช้ช่วง 700-2300 แทนก็สามารถใช้ฟังก์ชัน writeMicroseconds นี้เพื่อกำหนดความกว้างพัลส์ได้เอง

การใช้ฟังก์ชัน writeMicroseconds สามารถกำหนดค่าได้อิสระ ตรงนี้ ”ต้องระวังในการใช้งาน” หากสั่งงาน RC Servo Motor (แบบ 0 – 180 องศา) จนหมุนไปเกินจุดสิ้นสุดคือเกินทั้งฝั่ง 0 หรือ 180 องศา จะทำให้เกิดเสียงครางดังจากการหมุนไปต่อไม่ได้และมอเตอร์จะกินกระแสสูงขึ้นด้วยในเวลาเดียวกันนั้น ซึ่งอาจทำให้ RC Servo Motor เกิดความเสียหายได้

Syntax

```
servo.writeMicroseconds(uS)
```

Parameters

uS: คือค่าความกว้างของพัลส์ที่ต้องการกำหนดในหน่วยไมโครวินาที (โดยตัวแปร int)

read()

Description

คือฟังก์ชันอ่านค่าองศาที่ตั้งเข้าไปด้วยฟังก์ชัน write() เพื่อให้รู้ว่าตำแหน่งองศาสุดท้ายที่เราสั่งเข้าไปนั้นมีค่าเท่าไรซึ่งค่าที่อ่านออกมานั้นจะมีค่าอยู่ในช่วง 0 – 180

Syntax

```
servo.read()
```

Parameters

ไม่มี: จะ Return ค่า 0-180

attached()

Description

คือฟังก์ชันตรวจสอบว่า Servo ที่เราต้องการใช้กำลังต่ออยู่กับขั้วสัญญาณของ Arduino หรือไม่

Syntax

```
servo.attached()
```

Parameters

ไม่มี: จะ Return ค่า True ออกมา หาก Servo Motor เชื่อมต่ออยู่กับ Arduino แต่ถ้าหาก Return

ออกมาเป็นค่าอื่นถือว่าไม่เชื่อมต่อ

detach()

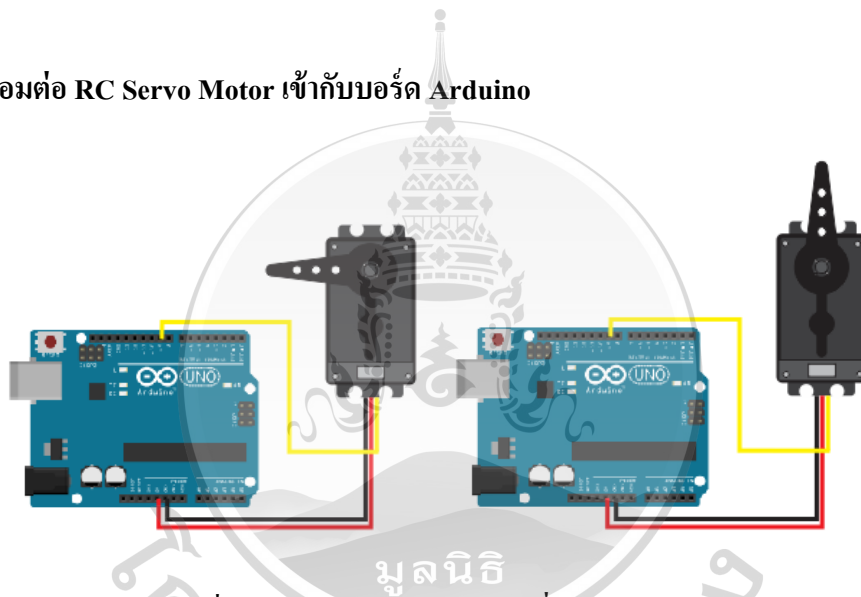
Description

คือฟังก์ชันคืนสถานะของขาที่เรากำหนดให้เป็นขาควบคุม Servo Motor ด้วยคำสั่ง attached() ให้กลับคือสู่การใช้งานปกติ

Syntax

```
servo.detach()
```

ตัวอย่างการเชื่อมต่อ RC Servo Motor เข้ากับบอร์ด Arduino



ภาพที่ 24 การหมุน Servo Motor ที่ 0 และ 90 องศา

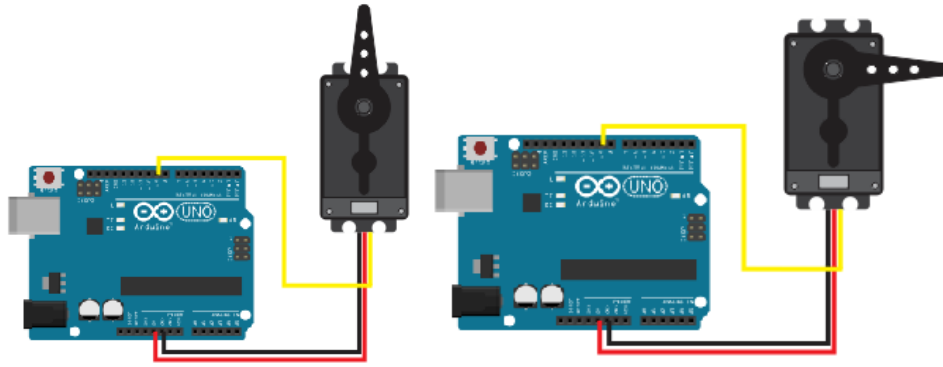
```
myservo.write(0);
```

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 0 องศา และ หยุดเป็นเวลา 1 วินาที myservo.write(90);

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 90 องศา และ หยุดเป็นเวลา 1 วินาที



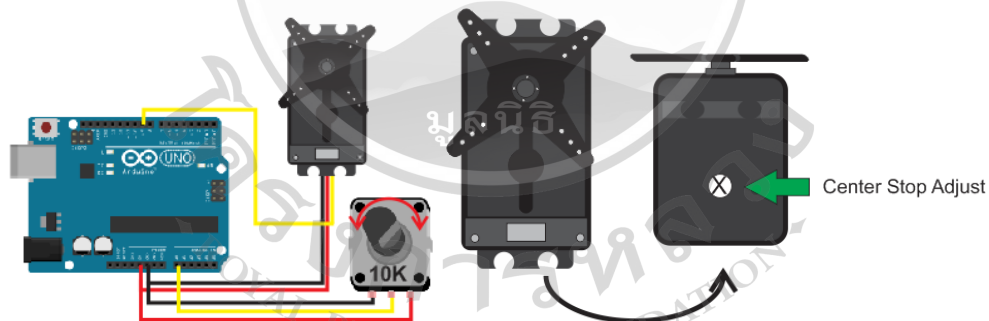
ภาพที่ 25 การหมุน Servo Motor ที่ 90 และ 180 องศา

```
myservo.write(90);
```

```
delay(1000);
```

Servo Motor จะหมุนไปที่ตำแหน่ง 180 องศา และ หยุดเป็นเวลา 1 วินาที จากนั้นจะหมุนกลับไปตำแหน่ง 0 องศา และวนรอบไปเช่นนี้เรื่อยๆ

ตัวอย่างการควบคุมตำแหน่ง RC Servo Motor

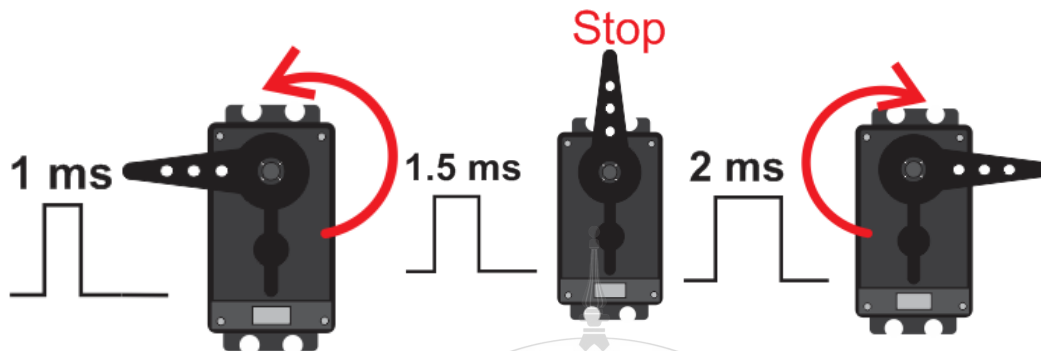


ภาพที่ 26 ตัวอย่างการควบคุมตำแหน่ง RC Servo Motor

Continuous Rotation Servo คือ RC Servo Motor แบบที่สามารถหมุนได้ 360 องศา ส่วนประกอบภายนอกนั้นจะมีหน้าตาคล้ายกับ RC Servo Motor แบบที่หมุนได้ 180 องศา เพียงแต่จะมี Potentiometer เพื่อใช้สำหรับปรับ ตำแหน่ง Center Stop Adjust ของตัว Servo

ลักษณะการใช้งาน RC Servo Motor ชนิดนี้จะแตกต่างจากการใช้งาน RC Servo Motor แบบ 180 องศาตรงที่ Servo ชนิดนี้จะใช้ความกว้างของสัญญาณพัลส์ในการกำหนดความเร็วและทิศทางในการหมุน ไม่ได้ใช้เพื่อกำหนดมุมจึงไม่สามารถกำหนดให้ Motor หมุนไปยังตำแหน่งมุมต่างๆ ตามความต้องการได้

สัญญาณความกว้างของพัลส์ที่ใช้ควบคุมจะอยู่ในช่วง 1000-2000 us แต่จะมีความแตกต่างกันความหมายของแต่ละความกว้างของพัลส์ดังนี้ ความกว้าง 1000 us หมายถึงการหมุนไปทางซ้ายด้วยความเร็วสูงสุดที่ Servo Motor จะหมุนได้ ความกว้าง 1500 us หมายถึงการสั่งให้ Servo Motor หยุดหมุน ความกว้าง 2000 us หมายถึงการหมุนไปทางขวาด้วยความเร็วสูงสุดที่ Servo Motor จะหมุนได้

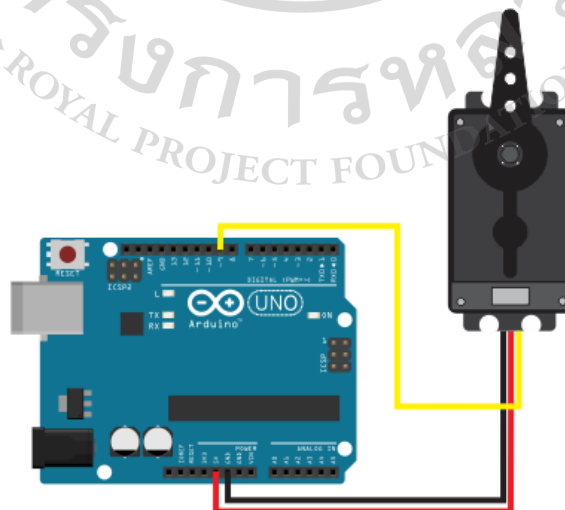


ภาพที่ 27 การกำหนดสัญญาณความกว้างของพัลส์ที่ใช้ควบคุมให้ servo motor หมุนเป็นมุม

การ Calibrate Center Stop

ในการใช้งาน Continuous Rotation Servo เมื่อซื้อใหม่หรือใช้งานไปสักระยะหนึ่งจุด Center Stop อาจมีการคลาดเคลื่อนได้ ซึ่งแม้เราสั่งให้สัญญาณพัลส์มีความกว้างเท่ากับ 1500 us ไป Continuous rotation servo ก็จะไม่หยุดหมุน เราจึงต้องปรับตั้งค่า Center Stop ดังนี้

– ต่อ Continuous rotation servo เข้ากับ Arduino ดังนี้



ภาพที่ 28 การใช้ Arduino ควบคุม servo motor

Servo motor มักใช้เป็นมอเตอร์ขับเคลื่อนแกนหมุน x y หรือ z ของเครื่อง CNC (Computer Numerical Control) ที่ใช้เจาะ ผลิต ตัด แกะสลัก กลึงวัสดุตามขนาดที่กำหนดทั้งความกว้าง ยาว ลึก ได้อย่างอัตโนมัติ ซึ่งเครื่อง CNC จะมีระบบการทำงานดังนี้

ระบบขับเคลื่อน ประกอบด้วยสองระบบหลัก คือ ระบบส่งกำลังหลักและระบบขับเคลื่อนแกน-ระบบส่งกำลัง เครื่องกลึง CNC จะใช้เซอร์โวมอเตอร์ (Servo Motor) ซึ่งสามารถควบคุมความเร็วการหมุน แรงบิด ตำแหน่ง ให้เป็นไปตามความต้องการได้

- ระบบขับเคลื่อนแกน ในระบบนี้จะใช้ สเตปเปอร์ หรือ สเตปเปอร์มอเตอร์ (Stepping Motor) ทำงานร่วมกับระบบควบคุมระยะในการควบคุมแกนให้เข้าตัดชิ้นงานต่างๆ

ระบบจับยึด ระบบจับยึดของเครื่องกลึง CNC มีสองส่วน คือส่วนที่ใช้จับยึดชิ้นงาน และส่วนที่ใช้จับยึดเครื่องมือ-ระบบจับยึดชิ้นงาน หรือ หัวจับ (JAW) ทำหน้าที่ยึดชิ้นงานอย่างมั่นคง ทำให้สามารถตกแต่งกลึงชิ้นงานได้อย่างราบรื่น ซึ่งหัวจับชิ้นงานนี้มีทั้งแบบธรรมดาที่ไม่ตกแต่งรูปร่างอะไร หรือแบบพิเศษที่ใช้ผลิตงานจำนวนไม่มาก แต่มีความเฉพาะตัวสูง

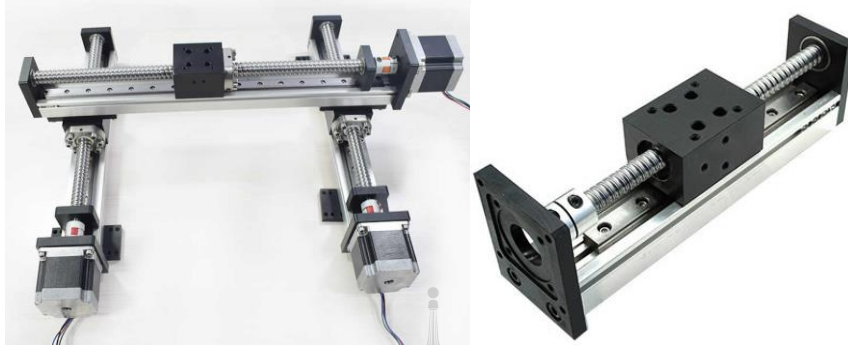
- ระบบจับยึดเครื่องมือตัด (Turret) ระบบจับยึดเครื่องมือ เป็นอีกสิ่งที่ทำให้ CNC มีความแตกต่างกับการกลึงแบบ Manual เพราะใน Turret จะประกอบด้วยเครื่องมือตัดเป็นจำนวนมาก สามารถหมุนเพื่อใช้งานในรูปแบบต่างๆ ได้เลยตามที่คอมพิวเตอร์กำหนดคำสั่งเอาไว้ โดยไม่ต้องมีการเปลี่ยนหัวตัดด้วยมือ

ระบบตรวจวัด ระบบตรวจวัดสำหรับเครื่อง CNC จะมีการติดตั้ง Linear Scale ซึ่งคืออุปกรณ์ตรวจวัดระยะเส้นตรงความละเอียดสูง (สูงสุด 0.001 mm) หรืออุปกรณ์อื่น ๆ ที่มีความสามารถใกล้เคียงกัน เพื่อวัดระยะของชิ้นงาน หรือใช้เพื่อคำนวณในฟังก์ชันต่างๆ เพื่อการใช้งานเครื่อง CNC

ระบบไฟฟ้า ส่วนใหญ่ระบบไฟฟ้าของเครื่องกลึง CNC จะเป็นระบบไฟฟ้าแบบ 3 เฟส 380 โวลต์เป็นหลัก ยกเว้นเป็นเครื่องแบบ MINI CNC ที่จะใช้ไฟฟ้าแบบ 220 โวลต์

ระบบควบคุม เครื่องกลึง CNC จะมีการควบคุมการทำงานทั้งหมดผ่านคอมพิวเตอร์ โดยจะอาศัย G Code และ M Code ในการป้อนข้อมูล ซึ่งสามารถควบคุมได้ตั้งแต่การเคลื่อนที่ง่ายๆ ไปจนถึงการปรับรายละเอียดการกลึง

สำหรับระบบขับเคลื่อนจะมีอุปกรณ์ที่เป็นรางเลื่อนซึ่งสามารถเลื่อนไปในทิศทางต่างๆได้ตามลักษณะของราง โดยการหมุนเกลียวด้วยมอเตอร์ ดังนั้นสามารถควบคุมได้โดยใช้มอเตอร์ร่วมกับรางนี้ในการประดิษฐ์พวกแขนกลต่างๆ



ภาพที่ 29 รางเลื่อน linear rail

ซึ่งระบบควบคุม CNC สามารถโปรแกรมเพิ่มเติมได้โดยการเขียนโค้ด ซึ่งในปัจจุบันนิยมเขียนลงในโปรแกรม CAD (Computer Aided Design) ทั้ง 2D CAD และ 3D CAD แล้วใช้โปรแกรมประเภท CAM (Computer Aided Manufacturing) ในการปรับแต่งให้เหมาะสมกับงาน



ภาพที่ 30 เครื่อง CNC router

เครื่อง CNC router ที่ควบคุมการหมุนของ stepper motor เพื่อกำหนดระยะการเคลื่อนที่ไปตามรางเลื่อนได้ ใน 3 มิติ กว้าง ยาว และใช้ servo motor กำหนดระยะมิติความลึกอย่างละเอียดในการตัดเจาะกลึงของหัวตัด เจาะ

ในการศึกษาครั้งนี้จะออกแบบเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ ใช้ micro processor ร่วมกับ บอร์ดวงจรควบคุมการเคลื่อนที่ของหัวพิมพ์ แบบ 3 มิติ (3D printer) เช่น Raspberry, Arduino, KFB2.0 3D Printer Controller Board เพื่อให้สามารถควบคุมการเคลื่อนที่ของมอเตอร์ทั้ง สเตปเปอร์มอเตอร์ (Stepper Motor) และเซอร์โวมอเตอร์ (Servo Motor) โดยการใช้ Stepper Motor เคลื่อนย้ายและยกวัตถุไปตาม แนวแกนระนาบ ทั้งแนวนอน แนวตั้งและแนวลึก และใช้ Servo Motor ในการหมุนเพื่อปรับการเคลื่อนที่ที่ละเอียดต้องการความแม่นยำ โดยควบคุมการเคลื่อนที่ของตำแหน่งไมโครปีเปิดไปตามแนวรางแกน x, y และควบคุมการปีเปิดดูจ่ายน้ำยาตรวจหาสารตกค้าง 5 ชนิดที่วางไว้ตามตำแหน่งที่กำหนดไว้ตามปริมาตร และเวลาที่กำหนดไว้ในโปรแกรม โดยมีระบบเข้าและระบบให้ความร้อนด้านฐานของ rack อีกด้วย

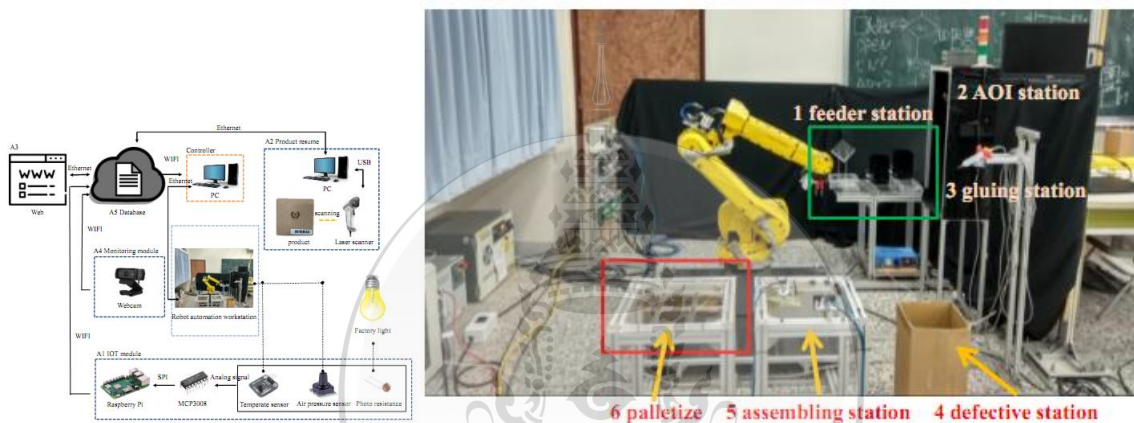
2.2 ผลงานวิจัยที่เคยทำมาก่อน หรืองานวิจัยที่เกี่ยวข้อง

ผลงานวิจัยที่เกี่ยวข้องกับการประดิษฐ์เครื่องดูจ่ายของเหลวอัตโนมัติสำหรับห้องปฏิบัติการ โดย Okamoto, H.; Deuchi, K., 2000 Design of a Robotic Workstation for Automated Organic Synthesis, LRA, Vol. 12, pp. 2–11. [19] ได้ศึกษาพัฒนาเครื่องมือดูจ่ายของเหลว นำมาผสมและให้ความร้อนเพื่อทำปฏิกิริยาสังเคราะห์ทางเคมีอินทรีย์ ซึ่งให้ผลการสังเคราะห์สารที่เหมาะสม ดังภาพ



ภาพที่ 31 Sumitomo's robotic reaction workstation[19]

Chia-Ying Hsieh, Ying-Jie Jhao, Chen-Huan Chang, and Lin-Yin Chen, Development of Intelligence Automated Robotic Arm Workstation, International Journal of Mechanical Engineering and Robotics Research Vol. 9, No. 5, May 2020 [20] ได้ทำการวิจัยประดิษฐ์ Robotic workstation และ IC ชนิด MCP3008 ไมโครชิปทำหน้าที่ แปลงสัญญาณ Analog ให้เป็น Digital แบบ 8 Channel (8 ช่องสัญญาณ) โดยความละเอียดแบบ 10 Bit (0-1023) ด้วยการสื่อสารแบบ SPI Interface เหมาะสำหรับนำไปใช้งานในการขยายขาสัญญาณ analog ให้กับ ไมโครคอนโทรลเลอร์ ชนิด Raspberry Pi ให้สั่งงานและควบคุมมอเตอร์ให้ทำงาน ดังภาพ



ภาพที่ 32 Robotic ARM Workstation[19]

มูลนิธิ

โครงการหลวง
ROYAL PROJECT FOUNDATION

บทที่ 3

กรรมวิธีการทดลอง

3.1 สถานที่ทดลอง ระยะเวลาทำการทดลอง

ศูนย์พัฒนาโครงการหลวงแม่สาใหม่ กันยายน 2563- ตุลาคม 2564 รวบรวมอุปกรณ์ ประกอบ สร้างประดิษฐ์ชิ้นงาน

ห้องปฏิบัติการวิเคราะห์สารตกค้าง แผนกงานอารักขาพืช มูลนิธิโครงการหลวง กันยายน 2563- ตุลาคม 2564 วางแผน ออกแบบ ทดสอบสารตกค้างในห้องปฏิบัติการ ทดสอบประสิทธิภาพทางด้านเคมี ปรับปรุงแก้ไขพัฒนา ประเมินความพึงพอใจ จัดทำเอกสาร คู่มือสรุปผล รายงานผล

3.2 วัสดุทดลองที่ใช้

ชุด โครง Aluminum Profile 2020 V-slot พื้น Stainless Steel ขนาด 700X550x550 mm

ชุดเลื่อนแกน x Linear Guide Rail 550 mm 2GT-6mm PU Timing Belt Steel

ชุดเลื่อนแกน Y Solid v wheel 16 Teeth Timing Pulley 2GT-6mm PU Timing Belt Steel

ชุด servo motor Digital Metal Gear Servo 47.8kg และ Digital Metal Gear Servo 5.1kg

ชุด Stepper Motor 1.8 degree

ชุดชุดจ่ายของเหลว ระบบเลื่อนแกน Z และ micropipette Gilson pipetman 1000P

ชุด Aluminum Heat block for 53 test tube 13 mm Powder Coating

ชุดควบคุมอุณหภูมิ Aluminum 220x220mm Heat Bed

ชุด GT reagent rack and sample rack Duran Bottle 50 ml

ชุด tip box 1000 ไมโครลิตร Rainin

ชุด power supply Cable Drag Chain ระบบเดินไฟฟ้า UPS สำรองไฟ

ชุด microcontroller Arduino nano ควบคุม เซอร์โว มอเตอร์, Arduino Mega 2560/KFB 2.0

ควบคุมสเตปเปอร์มอเตอร์ board 20x4 LCD backlight, Arduino UNO ควบคุมอุณหภูมิ

ชุดหน้าจอแสดงผล LCD 128x64 Backlight 3.5 นิ้ว

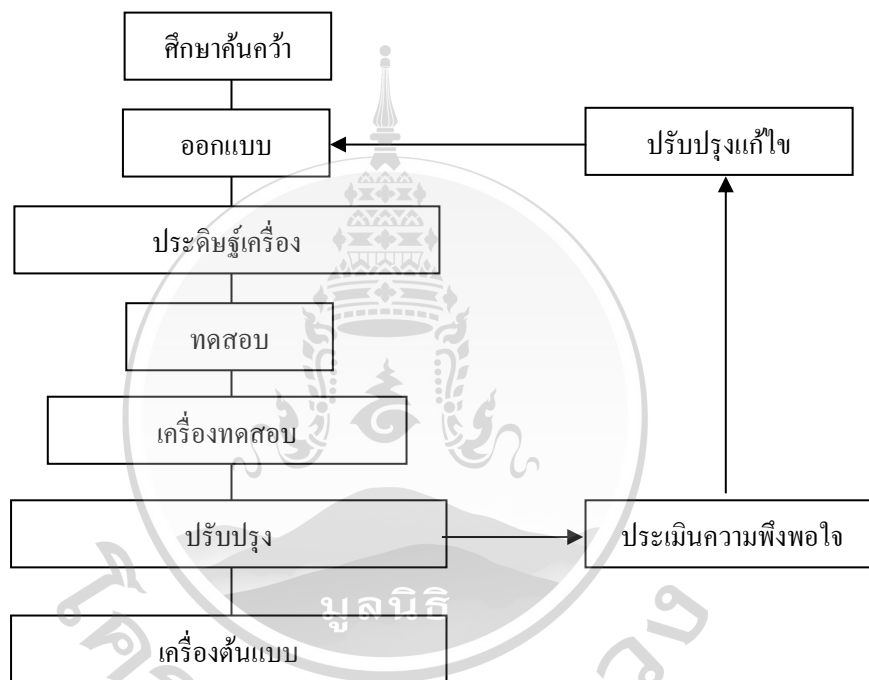
3.3 วิธีทดลองที่ใช้ การบันทึกข้อมูล วิธีการวิเคราะห์ข้อมูล

3.3.1 วิธีทดลองที่ใช้

3.3.1.1 ออกแบบและประกอบตัวเครื่อง

ออกแบบส่วนประกอบด้วยโปรแกรม Ultimaker Cura Version 15.0.4, เขียนโปรแกรมควบคุม microcontroller ด้วย Arduino IDE

ส่วนภายนอกจะประกอบด้วยฝาครอบตัวถังด้านหน้าและด้านข้าง ฝาครอบตัวถังด้านหลัง ฐานและ ขากรองเครื่อง ตามแผนผังในภาพ



ภาพที่ 33 แผนผังกิจกรรมการดำเนินงาน

ในการออกแบบ มีหลักการดังนี้

- 1 ใช้ Power supply adaptor แปลงไฟฟ้ากระแสสลับ 220 V AC 50Hz ไปเป็นไฟฟ้ากระแสตรง DC 9 V. 2 A. ส่งไปยังสวิทช์เปิดที่ต่อเข้ากับบอร์ด Step down high voltage เพื่อกรองกระแสให้คงที่ เพื่อส่งไฟไปเลี้ยงบอร์ดวงจรต่างๆ ใช้ microcontroller Arduino nano ควบคุมเซอร์โวมอเตอร์, Arduino Mega 2560/KFB 2.0 ควบคุมสเตปเปอร์มอเตอร์
- 2 servo motor ควบคุมการหมุนของเฟืองเพื่อส่งกำลังให้เพลาคือเคลื่อนที่ตามรางเลื่อน ชุดเลื่อนแกน x Linear Guide Rail 550 mm 2GT-6mm PU Timing Belt Steel และชุดเลื่อนแกน Y Solid v wheel 16 Teeth Timing Pulley 2GT-6mm PU Timing Belt Steel เพื่อควบคุมการกด Tip ejection, และปรับระดับความสูงต่ำของปีเปิด(แกน-z),

- 3 stepper motor และ linear ball screw ควบคุมการกด plunger ควบคุมการกดของบีเปิด
- 4 ชุดแกนเลื่อน Linear Rail Axis Slide Stroke Long Stage Actuator ขับเคลื่อนรอกสายพานด้วย stepper motor เป็นระบบขับเคลื่อนที่มีอุปกรณ์ที่เป็นรางเลื่อนซึ่งสามารถเลื่อนไปในทิศทางต่างๆตามแนวแกน (x,y) ได้ตามลักษณะของรางโดยการหมุนเกลียวด้วยมอเตอร์
- 5 ชุดชุดจ่ายของเหลวประกอบด้วยไมโครบีเปิดขนาด 1000 ไมโครลิตรและ rack สำหรับบรรจุทึบ ควบคุมการดูดจ่ายด้วย servo และ step motor มี rack สำหรับวางหลอดตัวอย่าง และช่องวางขวดน้ำยา GT-1 ถึง GT-5
- 6 ชุด micro processor Arduino UNO, และบอร์ดวงจรควบคุมการเคลื่อนที่แบบ 3 มิติ Arduino Mega 2560/KFB2.0 3D Printer Controller Board และระบบไฟฟ้าที่จ่ายพลังงานให้แก่อุปกรณ์ไฟฟ้าในระบบขับเคลื่อนต่างๆเพื่อให้สามารถควบคุมการเคลื่อนที่ของมอเตอร์ทั้ง สเต็ปเปอร์มอเตอร์ (Stepper Motor) และเซอร์โวมอเตอร์ (Servo Motor) โดยการใช้ Stepper Motor เคลื่อนย้ายและยกวัตถุไปตามแนวแกน ระบาย ทั้งแนวนอน แนวตั้งและแนวทึบ และใช้ Servo Motor ในการหมุนเพื่อปรับการเคลื่อนที่ที่ละเอียดต้องการความแม่นยำ โดยควบคุมการเคลื่อนที่ของตำแหน่งไมโครบีเปิดไปตามแนวรางแกน x, y และควบคุมการบีเปิดชุดจ่ายน้ำยาตรวจสอบหาสารตกค้าง 5 ชนิดที่วางไว้ตามตำแหน่งที่กำหนดไว้ ตามปริมาตรและเวลาที่กำหนด ระบบสายไฟร้อยไปตามรางกระดูกงู (Cable Carrier)
- 7 ชุดโครงสร้างเป็นเหล็กกล่องชุบสีขนาด 1x1 นิ้ว หนา 25 ม.ม. กว้าง 55 cm. ยาว 68 cm. สูง 51 cm และ ราง aluminium profile
- 8 ระบบทำความร้อนด้วย Aluminum heater plate และควบคุมอุณหภูมิด้วยบอร์ดและเซนเซอร์ ส่งถ่าย ความร้อนไปยัง aluminum block ที่เจาะรูให้พอดีกับหลอดทดลอง 53 ช่อง
- 9 ระบบปุ่มสวิทช์และหน้าจอควบคุม/แสดงผล LCD Display 3.5 นิ้ว ควบคุมโดยบอร์ด Arduino Mega 2560/KFB2.0

3.3.2 การทดสอบหาสภาวะที่เหมาะสมในการทำงาน

ทดสอบหาสภาวะที่เหมาะสมที่ทำให้การทำงานให้มีประสิทธิภาพสูงสุด ดังนี้

3.3.2.1 ศึกษาความสามารถในการทำงานของเครื่อง โดยทดสอบความถูกต้อง แม่นยำ ในการคัดจ่ายของเหลวโดยการสอบเทียบ

ตั้งงานให้เครื่องคัดจ่ายน้ำกลั่นตลอดช่วงการใช้งาน 4 จุด ประกอบด้วยปริมาตร 0.1, 0.2, 0.4, 0.8 mL จุดละ 10 ซ้ำ นำไปชั่งน้ำหนักด้วยเครื่องชั่งไฟฟ้า ความละเอียดทศนิยม 5 ตำแหน่ง คำนวณหาค่าเฉลี่ย (mean) และค่าเบี่ยงเบนมาตรฐาน (SD) และ %error

3.3.2.2 ศึกษาความสามารถในการทำงานของเครื่อง โดยทดสอบตัวอย่างควบคุมที่ทราบค่าจริงที่ระดับความเข้มข้น 0(blank)จะให้ผลตรวจไม่พบ, Spiked sample (profenophos 0.7 mg/kg จะให้ผลพบปลอดภัย $I_{20\%}$, 1 mg/kg จะให้ผล พบไม่ปลอดภัย $I_{30\%}$), 3 mg/kg จะให้ผล พบไม่ปลอดภัย $\geq I_{50\%}$) ด้วยน้ำยา GT-1 – GT-5 ชุดละ 25 ซ้ำ

3.3.2.3 ศึกษาความสามารถในการทำงานของเครื่องในระยะยาว โดยนำไปใช้งานจริง วันละ 25 ตัวอย่าง และหาค่าเบี่ยงเบนมาตรฐานระดับ 0, +1, +3 เก็บข้อมูลต่อเนื่อง 3 เดือน (กรกฎาคม-กันยายน 2564) ตรวจสอบเช็คข้อบกพร่องและปัญหาที่ต้องปรับแก้

3.3.3 ประเมินความพึงพอใจ

ใช้แบบสอบถามความพึงพอใจแบบมาตราประมาณค่า (Rating scale) ประชากรและกลุ่มตัวอย่างได้จากการสุ่มแบบเจาะจง (Purposive Sampling) โดยนำเครื่องมือที่ประดิษฐ์ขึ้นไปให้เจ้าหน้าที่ปฏิบัติการวิเคราะห์สารตกค้าง ที่ประจำอยู่ตามศูนย์พัฒนาโครงการหลวง ที่คุ้นเคยวิธีการตรวจวิเคราะห์ด้วยชุดทดสอบยาฆ่าแมลง GT อย่างน้อยจำนวน 10 ศูนย์/สถานี โดยศึกษาวิธีการใช้และตามผลการทดสอบประสิทธิภาพ รวบรวมผลการใช้ปัญหาอุปสรรค ตลอดจนทดสอบผลการปฏิบัติงานเป็นที่น่าพอใจของผู้ใช้ทดสอบประสิทธิภาพ กำหนดเกณฑ์สำหรับการให้ความหมายของผลจากค่าเฉลี่ยที่ได้ โดยวิธีของเบสท์ (Best, 1986) ดังนี้

| | | |
|---------|---------|---|
| ระดับ 1 | หมายถึง | เหมาะสม / เห็นด้วยอยู่ในระดับน้อยที่สุด |
| ระดับ 2 | หมายถึง | เหมาะสม / เห็นด้วยอยู่ในระดับน้อย |
| ระดับ 3 | หมายถึง | เหมาะสม / เห็นด้วยอยู่ในระดับปานกลาง |

ระดับ 4 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับมาก

ระดับ 5 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับมากที่สุด

กำหนดเกณฑ์สำหรับการให้ความหมายของผลจากค่าเฉลี่ยที่ได้ โดยวิธีของเบสท์ (Best, 1986) ดังนี้

1.00 - 1.50 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับน้อยที่สุด

1.51 - 2.50 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับน้อย

2.51 - 3.50 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับปานกลาง

3.51 - 4.50 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับ มาก

4.51 - 5.00 หมายถึง เหมาะสม / เห็นด้วยอยู่ในระดับมากที่สุด

โดยหัวข้อการประเมิน ความเหมาะสม ประสิทธิภาพและความพึงพอใจของเครื่องที่พัฒนาขึ้น กำหนดไว้ดังนี้ โดยแบ่งหัวข้อการประเมิน ไว้ดังนี้

1. มีประสิทธิภาพสูงในการสกัด สามารถวัดผลได้ถูกต้องแม่นยำ ในการทดสอบเมื่อเทียบกับการวัดสายตากคน
2. วิธีการไม่ยุ่งยาก ความเหมาะสมของโครงสร้าง และการใช้งาน

สถิติที่ใช้ในการวิเคราะห์ข้อมูลความพึงพอใจ

1. ค่าเฉลี่ย โดยใช้สูตร

$$\bar{X} = \frac{\sum X}{N}$$

\bar{X} = ค่าเฉลี่ยของคะแนน

$\sum X$ = ผลรวมของคะแนน

N = จำนวนข้อมูล

2. ส่วนเบี่ยงเบนมาตรฐาน, S โดยใช้สูตร

$$S^2 = \frac{N\sum X^2 - (\sum X)^2}{N(N-1)}$$

S = ค่าเบี่ยงเบนมาตรฐาน

N = จำนวนข้อมูล

3.3.4 ปรับปรุงแก้ไขและผลิตเครื่องต้นแบบ

รวบรวมข้อมูลจากแบบประเมินความพึงพอใจ ข้อเสนอแนะต่างๆ นำมาปรับปรุงตามข้อเสนอแนะแก้ไขข้อบกพร่องต่างๆที่สามารถแก้ไขได้ จนได้เครื่องต้นแบบที่สมบูรณ์ สามารถทำงานได้อย่างมีประสิทธิภาพเป็นไปตามวัตถุประสงค์ที่ตั้งไว้



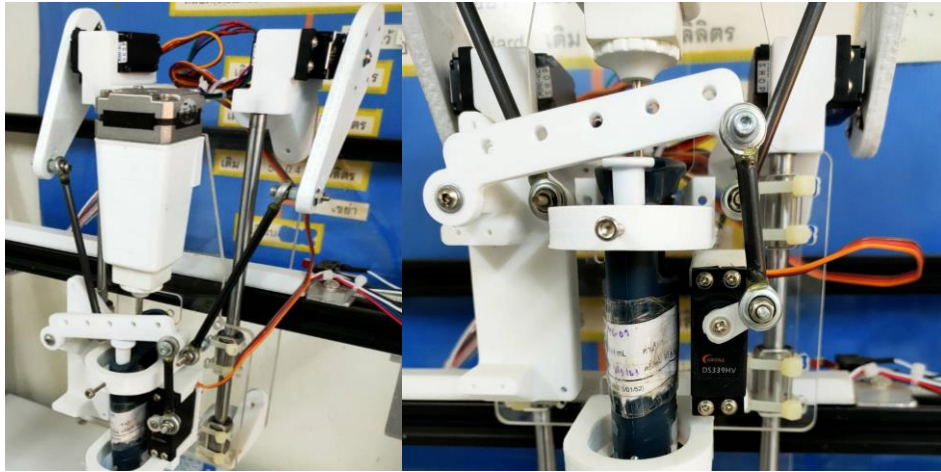
บทที่ 4 ผลการวิจัย

4.1 การออกแบบและประดิษฐ์

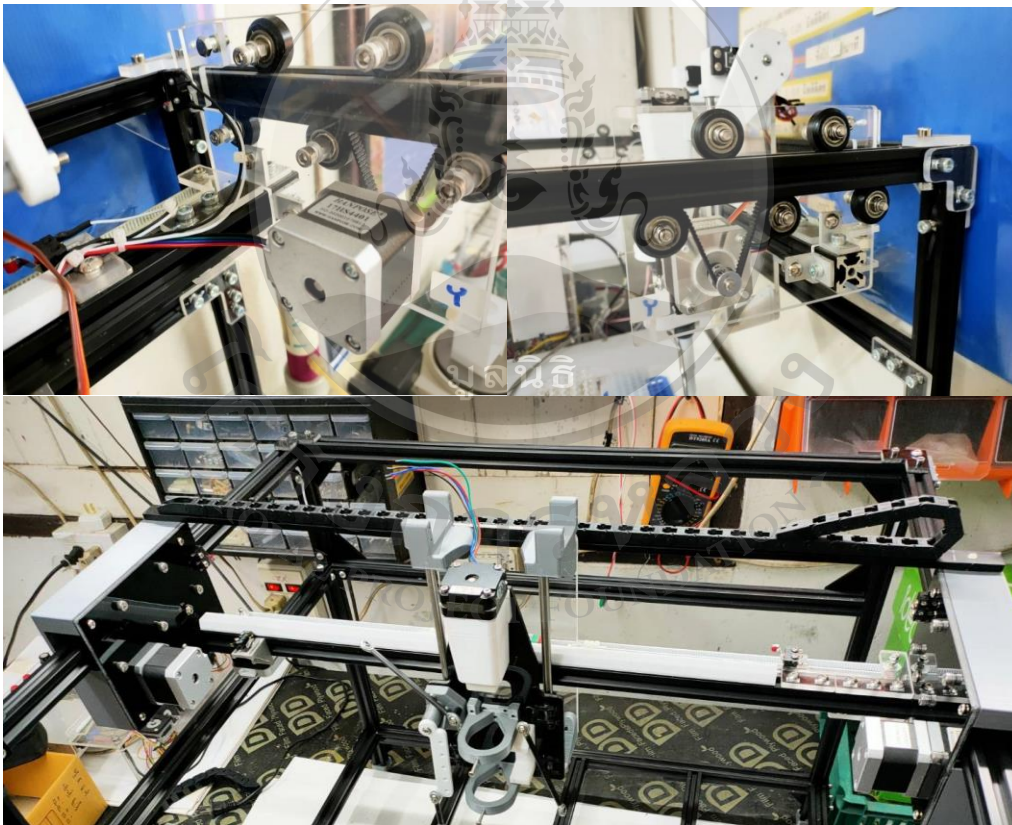
ประกอบชุดอุปกรณ์สำหรับภายใน



ภาพที่ 34 ส่วนประกอบภายในเครื่องที่ประดิษฐ์ขึ้น



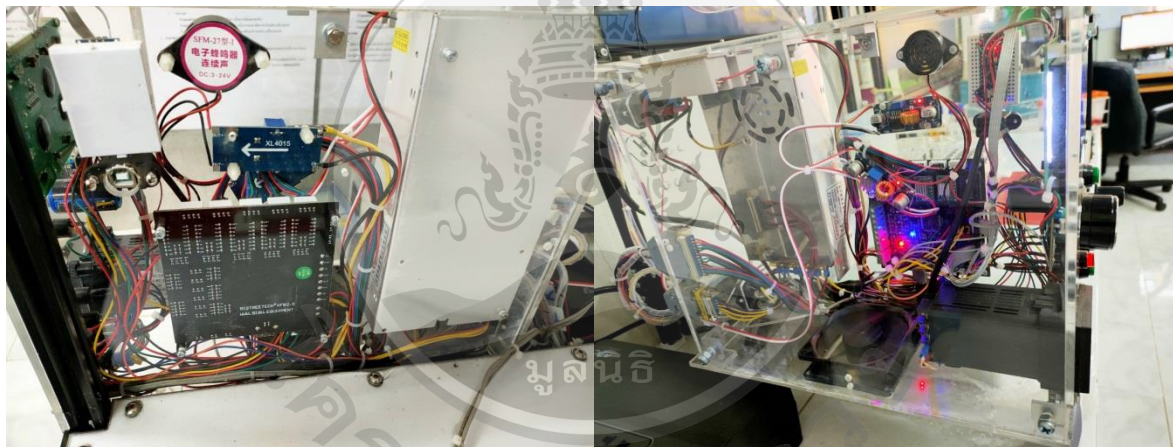
ภาพที่ 35 servo motor ควบคุมการหมุนของเฟืองเพื่อส่งกำลังให้เพลาลูกเบี้ยวที่เพื่อควบคุมการกด Tip ejection, และระดับความสูงต่ำของปีเปต(แกน-z), stepper motor และ linear ball screw ควบคุมการกด plunger คู่อ้ายของปีเปต



ภาพที่ 36 ชุดแกนเลื่อน Linear Rail Axis Slide Stroke Long Stage Actuator ขับเคลื่อนรอกสายพานด้วย stepper motor เป็นระบบขับเคลื่อนที่มีอุปกรณ์ที่เป็นรางเลื่อนซึ่งสามารถเลื่อนไปในทิศทางต่างๆตามแนวแกน (x,y) ได้ตามลักษณะของราง โดยการหมุนเกลียวด้วยมอเตอร์ สายไฟจะร้อยไปตามรางกระดูกงู (Cable Carrier)



ภาพที่ 37 ชุดชุดจ่ายของเหลวประกอบด้วยไมโครปิเปตขนาด 1000 ไมโครลิตรและ rack สำหรับบรรจุที่ควบคุมการดูดจ่ายด้วย servo และ step motor มี rack สำหรับวางหลอดตัวอย่าง และช่องวางขวดน้ำยา GT-1 ถึง GT-5



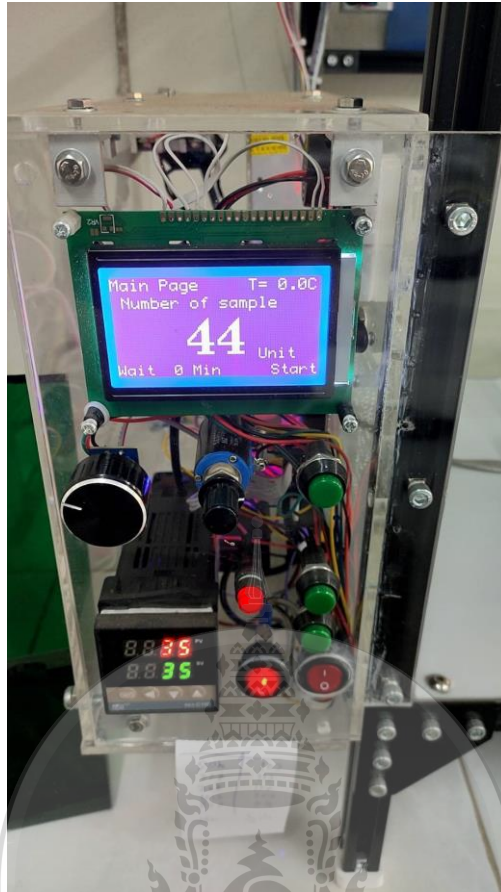
ภาพที่ 38 ชุด micro processor และบอร์ดวงจรควบคุมการเคลื่อนที่แบบ 3 มิติ Arduino Mega 2560/KFB2.0 3D Printer Controller Board และระบบไฟฟ้าที่จ่ายพลังงานให้แก่อุปกรณ์ไฟฟ้าในระบบขับเคลื่อนต่างๆ เพื่อให้สามารถควบคุมการเคลื่อนที่ของมอเตอร์ทั้ง สเต็ปเปอร์มอเตอร์ (Stepper Motor) และเซอร์โวมอเตอร์ (Servo Motor) โดยการใช้ Stepper Motor เคลื่อนย้ายและยกวัตถุไปตามแนวแกนระนาบ ทั้งแนวนอน แนวตั้งและแนวทึ่ก และใช้ Servo Motor ในการหมุนเพื่อปรับการเคลื่อนที่ที่ละเอียดต้องการความแม่นยำ โดยควบคุมการเคลื่อนที่ของตำแหน่งไมโครปิเปตไปตามแนวรางแกน x, y และควบคุมการปิเปตดูดจ่ายน้ำยาตรวจหาสารตกค้าง 5 ชนิดที่วางไว้ตามตำแหน่งที่กำหนดไว้ ตามปริมาตรและเวลาที่กำหนด



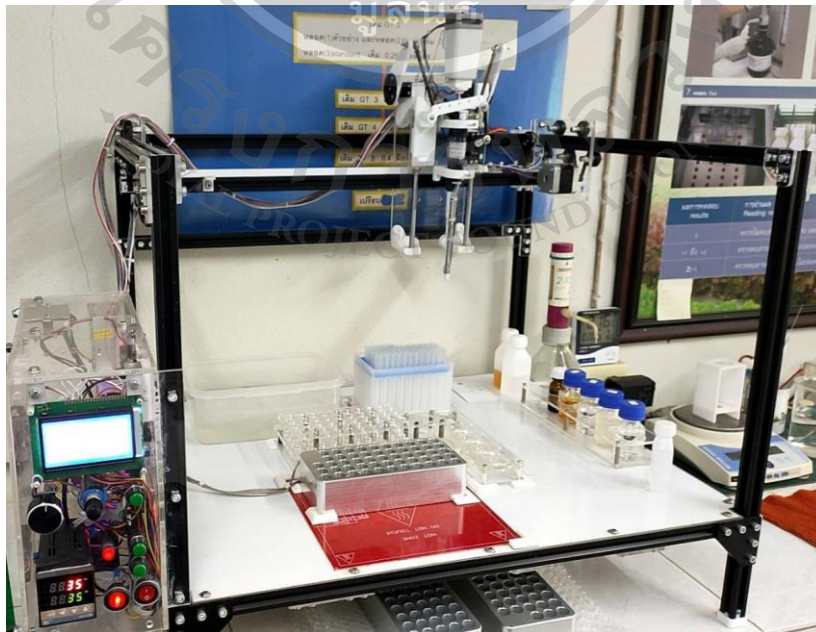
ภาพที่ 39 ชุดโครงสร้างเป็นเหล็กกล่องชุบสีขนาด 1x1 นิ้ว หนา 25 ม.ม. และราง aluminium profile



ภาพที่ 40 ระบบทำความร้อนด้วย Aluminum heater plate และควบคุมอุณหภูมิด้วยบอร์ดและเซนเซอร์ ส่ง
ถ่ายความร้อนไปยัง aluminum block ที่เจาะรูให้พอดีกับหลอดทดลอง 53 ช่อง



ภาพที่ 41 ระบบปุ่มสวิตช์และหน้าจอควบคุม/แสดงผล LCD Display 3.5 นิ้ว ควบคุมโดยบอร์ด Arduino Mega 2560/KFB2.0



ภาพที่ 42 โครงสร้างโดยรวมของ Automated Workstation for Toxic Residue Analysis

4.2 การทดสอบประสิทธิภาพการทำงานของเครื่องที่สร้างขึ้น

4.2.1 ศึกษาความสามารถในการทำงานของเครื่อง โดยทดสอบความถูกต้อง แม่นยำ ในการดูดจ่ายของเหลวโดยการสอบเทียบ สั่งงานให้เครื่องดูดจ่ายน้ำกลั่นตลอดช่วงการใช้งาน 4 จุด ประกอบด้วย ปริมาตร 0.1, 0.2, 0.4, 0.8 mL จุดละ 10 ซ้ำ นำไปชั่งน้ำหนักด้วยเครื่องชั่งไฟฟ้า ความละเอียดทศนิยม 5 ตำแหน่ง คำนวณหาค่าเฉลี่ย (mean) และค่าเบี่ยงเบนมาตรฐาน (SD) และ %error

ตาราง 11 ผลการทดสอบความถูกต้อง แม่นยำ ในการดูดจ่ายของเหลวที่ปริมาตร 0.1, 0.2, 0.4, 0.8 mL

| ครั้งที่ | จุดสอบเทียบปริมาตร (mL) | | | |
|----------|-------------------------|----------|----------|----------|
| | 0.1 | 0.2 | 0.4 | 0.8 |
| 1 | 0.08435 | 0.19099 | 0.41409 | 0.80311 |
| 2 | 0.10123 | 0.19670 | 0.39662 | 0.80638 |
| 3 | 0.11973 | 0.18808 | 0.38160 | 0.79137 |
| 4 | 0.10690 | 0.20077 | 0.41056 | 0.80377 |
| 5 | 0.11647 | 0.18852 | 0.39594 | 0.78935 |
| 6 | 0.10440 | 0.20110 | 0.40581 | 0.78486 |
| 7 | 0.10188 | 0.19860 | 0.39382 | 0.79208 |
| 8 | 0.09301 | 0.21433 | 0.38756 | 0.78687 |
| 9 | 0.09245 | 0.20773 | 0.38751 | 0.79960 |
| 10 | 0.08728 | 0.22013 | 0.40316 | 0.81446 |
| Mean | 0.10077 | 0.20069 | 0.39767 | 0.79719 |
| SD | 0.011733 | 0.010725 | 0.010618 | 0.009699 |
| Error | 0.00077 | 0.00069 | -0.00233 | -0.00281 |
| %Error | 0.77 | 0.35 | -0.58 | -0.35 |

4.2.2 ทดสอบตัวอย่างควบคุมที่ทราบค่าจริงที่ระดับความเข้มข้น 0 ($I_{0\%}$) จะให้ผลตรวจไม่พบ, Spiked sample (profenophos 0.7 mg/kg จะให้ผลพบปลอดภัย $I_{20\%}$, 1 mg/kg จะให้ผล พบไม่ปลอดภัย $I_{30\%}$), 3 mg/kg จะให้ผล พบไม่ปลอดภัย $\geq I_{50\%}$ ด้วยน้ำยา GT-1 – GT-5 จำนวนชุดละ 25 ตัวอย่าง

| จำนวนตัวอย่าง | ผลทดสอบตัวอย่างควบคุมที่ทราบค่าจริงโดยเดิมสาร profenophos | | | |
|----------------|---|------------|------------|------------|
| | $I_{0\%}$ | $I_{20\%}$ | $I_{30\%}$ | $I_{50\%}$ |
| Fault negative | 0 | 0 | 0 | 0 |
| Fault positive | 0 | 0 | 0 | 0 |
| True negative | 25 | 0 | 0 | 0 |
| True positive | 0 | 25 | 25 | 25 |
| %Accuracy | 100 | 100 | 100 | 100 |

4.2.3 ผลการทดสอบความสามารถในการทำงานของเครื่องในระยะยาว โดยนำไปใช้งานจริง เก็บข้อมูล ต่อเนื่อง 3 เดือน (กรกฎาคม-กันยายน 2564)

ตาราง 12 ผลการทดสอบความสามารถในการทำงานของเครื่องในระยะยาว ต่อเนื่อง 3 เดือน (กรกฎาคม-กันยายน 2564)

| เดือน | จำนวนตัวอย่าง | ผลการทำงาน |
|---------|---------------|-------------------------|
| กรกฎาคม | 600 | ทำงานได้ปกติ ไม่พบปัญหา |
| สิงหาคม | 600 | ทำงานได้ปกติ ไม่พบปัญหา |
| กันยายน | 600 | ทำงานได้ปกติ ไม่พบปัญหา |

3. ปรับปรุงพัฒนาให้เครื่องทำงานได้มีประสิทธิภาพสูงสุด

- 3.1 เพิ่มขนาดของ รางกระดุกงู (Cable Carrier) เพื่อความสวยงาม และทนทานมากขึ้น
- 3.2 เปลี่ยนตัวถัง case box ของ controller จากพลาสติก เป็น stainless เพื่อให้ทนทาน สวยงาม
- 3.3 ปรับ module ตัว lock ไมโครเปิดให้กระชับ แน่นมากขึ้น



ภาพที่ 43 การปรับปรุงพัฒนาเครื่องให้มมีประสิทธิภาพมากขึ้น

4.4 ผลการประเมินความพึงพอใจที่มีต่อเครื่องวัดค่าสี

จากผู้ปฏิบัติงานวิเคราะห์สารตกค้างประจำศูนย์/สถานี จำนวน 27 คน แบบมาตราประมาณค่า (Rating scale)

ช่วงคะแนน 1-5 ตามวิธีของเบสท์ ดังตาราง

ตาราง 13 ผลการประเมินความพึงพอใจที่มีต่อเครื่อง จากการใช้แบบสอบถามความพึงพอใจ จากเจ้าหน้าที่วิเคราะห์สารตกค้างประจำศูนย์/สถานี จำนวน 27 คน

| Satisfaction Survey | Mean | Mean | Result |
|--|------|------------|------------|
| 1. ด้านประสิทธิภาพ และความถูกต้องแม่นยำในการทดสอบ | | 4.4 | มาก |
| 1.1 หลักการทำงานของเครื่องมือมีประสิทธิภาพสูง | 4.4 | | |
| 1.2 สามารถช่วยให้วัดผลได้ถูกต้องแม่นยำในการทดสอบ | 4.4 | | |
| 1.3 ลดปัญหา error จากการทำงานได้ ทำให้มีความเชื่อมั่นสูงขึ้น | 4.4 | | |
| 2. ด้านความสะดวกในการใช้งาน | | 4.3 | มาก |
| 2.1 เครื่องมือใช้งานได้ง่าย สะดวก ไม่ซับซ้อน | 4.1 | | |
| 2.2 เครื่องมือช่วยผ่อนแรง และลดขั้นตอนลง ทำให้ทำงานได้ดีขึ้น | 4.7 | | |
| 2.3 ความสะดวกในการดูแลทำความสะอาดหลังการใช้งาน | 4.4 | | |
| 3. ด้านโครงสร้าง | | 3.8 | มาก |
| 3.1 โครงสร้างเครื่องมือ มีความแข็งแรง หนาแน่น คงทน เหมาะกับงาน | 4.3 | | |
| 3.2 รูปร่าง ขนาดของเครื่องมือมีความเหมาะสมกับงานทดสอบ | 4.3 | | |
| 3.3 เครื่องมือมีรูปลักษณะภายนอกทันสมัย เหมาะสมกับงานทางวิทยาศาสตร์ | 2.8 | | |
| 4. ด้านเวลาและค่าใช้จ่าย | | 4.2 | มาก |
| 4.1 ช่วยลดค่าใช้จ่ายและเวลาในการทำซ้ำเมื่อทำผิดพลาด(ลืม ทำผิด) | 4.4 | | |
| 4.2 สามารถลดค่าใช้จ่ายจากค่าน้ำยา GT ลง 50 % | 4.0 | | |
| 4.3 อายุการใช้งานเครื่องมากกว่า 5 ปี | 4.3 | | |
| 5. ด้านคุณค่าด้านอื่น | | 4.5 | มาก |
| 5.1 มีประโยชน์ต่อผู้ปฏิบัติงาน ตอบสนองต่อความต้องการ นำไปใช้งานได้จริง | 4.4 | | |
| 5.2 เครื่องมือปลอดภัยต่อผู้ใช้งาน | 4.5 | | |
| 5.3 เกิดภาพลักษณ์ที่ดีต่อองค์กร ทันสมัย น่าภาคภูมิใจ | 4.6 | | |
| Mean | | 4.2 | มาก |

4.5 ปรับปรุงแก้ไข

- ไม่พบหัวข้อแนะนำให้ทำการปรับปรุงเพิ่มเติมจากผลการสำรวจความคิดเห็น

4.6 ผลสัมฤทธิ์จากการประเมินตัวชี้วัดของงานวิจัย

ตาราง 14 ผลสัมฤทธิ์ของงานวิจัย

| ผลผลิต | ตัวชี้วัด | ผลสัมฤทธิ์ |
|--|---|-----------------------|
| เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ 1 เครื่อง | เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ 1 เครื่อง 1 ชนิดเครื่องมือ | เป็นไปตามวัตถุประสงค์ |
| สามารถดูดจ่ายน้ำยา GT-1 ถึง GT-5 เดิมลงในหลอดตัวอย่าง ตามปริมาตรที่กำหนด สามารถเข้าหลอด ตั้งเวลาทำงาน สามารถควบคุมอุณหภูมิหลอดทดลอง ทำงานแทนคนได้อย่างสมบูรณ์ | สามารถดูดจ่ายน้ำยา GT-1 ถึง GT-5 เดิมลงในหลอดตัวอย่าง ตามปริมาตรที่กำหนด สามารถเข้าหลอด ตั้งเวลาทำงาน สามารถควบคุมอุณหภูมิหลอดทดลอง ทำงานแทนคนได้อย่างสมบูรณ์ | เป็นไปตามวัตถุประสงค์ |
| สามารถตั้งค่าให้ลดสัดส่วนน้ำยาที่ใช้ลงอย่างน้อยครั้งหนึ่ง ทำให้ต้นทุนลดลงอย่างน้อย 50% | สามารถตั้งค่าให้ลดสัดส่วนน้ำยาที่ใช้ลงตามที่ต้องการ ตั้งแต่ 10-90% จากปริมาตรปกติ | เป็นไปตามวัตถุประสงค์ |
| Human Error=0 | เครื่องทำงานอัตโนมัติตั้งแต่ GT-1 ถึง GT-5 | เป็นไปตามวัตถุประสงค์ |
| มีความถูกต้องแม่นยำสูงปราศจาก Human Error มีความน่าเชื่อถือ ลดความลังเลสงสัยในผล ลดเวลาจากการทำงานผิดพลาด ต้องทำงานซ้ำ จาก Human Error ลดต้นทุนจากการทำงานผิดพลาด ต้องทำงานซ้ำ จาก Human Error | จากผลการสอบเทียบปริมาตร การตรวจตัวอย่าง ควบคุม และการเฝ้าดูการทำงานระยะยาว เครื่องสามารถทำงานได้ถูกต้องแม่นยำ ไม่พบข้อผิดพลาด | เป็นไปตามวัตถุประสงค์ |

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ได้ทำการออกแบบ จัดหาวัสดุอุปกรณ์ที่เหมาะสม และประกอบเครื่องตรวจวิเคราะห์สารระบบอัตโนมัติ โดยตัวเครื่องประกอบด้วย Power supply adaptor แปลงไฟฟ้ากระแสสลับ 220 V AC 50Hz ไปเป็นไฟฟ้ากระแสตรง DC 9 V. 2 A. ส่งไปยังสวิทช์ปิดเปิดที่ต่อเข้ากับบอร์ด Step down high voltage เพื่อกรองกระแสให้คงที่ เพื่อส่งไฟไปเลี้ยงบอร์ดวงจรต่างๆ ใช้ microcontroller Arduino nano ควบคุมเซอร์โวมอเตอร์, Arduino Mega 2560/KFB 2.0 ควบคุมสเตปเปอร์มอเตอร์ servo motor ควบคุมการหมุนของเฟืองเพื่อส่งกำลังให้เพลาคือเคลื่อนที่ตามรางเลื่อน ชุดเลื่อนแกน x Linear Guide Rail 550 mm 2GT-6mm PU Timing Belt Steel และชุดเลื่อนแกน Y Solid v wheel 16 Teeth Timing Pulley 2GT-6mm PU Timing Belt Steel เพื่อควบคุมการกด Tip ejection, และปรับระดับความสูงต่ำของปีเปด(แกน-z), stepper motor และ linear ball screw ควบคุมการกด plunger คู่อจ่ายของปีเปด ชุดแกนเลื่อน Linear Rail Axis Slide Stroke Long Stage Actuator ขับเคลื่อนรอกสายพานด้วย stepper motor เป็นระบบขับเคลื่อนที่มีอุปกรณ์ที่เป็นรางเลื่อนซึ่งสามารถเลื่อนไปในทิศทางต่างๆตามแนวแกน (x,y) ได้ตามลักษณะของรางโดยการหมุนเกลียวด้วยมอเตอร์ ชุดคู่อจ่ายของเหลวประกอบด้วยไมโครปีเปดขนาด 1000 ไมโครลิตรและ rack สำหรับบรรจุทิป ควบคุมการคู่อจ่ายด้วย servo และ step motor มี rack สำหรับวางหลอดตัวอย่าง และช่องวางขวดน้ำยา GT-1 ถึง GT-5 , Stepper Motor เคลื่อนย้ายและยกวัตถุไปตามแนวแกนราบ ทั้งแนวนอน แนวตั้งและแนวลึก และใช้ Servo Motor ในการหมุนเพื่อปรับการเคลื่อนที่ที่ละเอียดต้องการความแม่นยำ โดยควบคุมการเคลื่อนที่ของตำแหน่งไมโครปีเปดไปตามแนวรางแกน x, y และควบคุมการปีเปดคู่อจ่ายน้ำยาตรวจหาสารตกค้าง 5 ชนิดที่วางไว้ตามตำแหน่งที่กำหนดไว้ ตามปริมาตรและเวลาที่กำหนด ระบบสายไฟร้อยไปตามรางกระดูกงู (Cable Carrier) ชุดโครงสร้างเป็นเหล็กกล่องชุบสีขนาด 1x1 นิ้ว หนา 25 ม.ม. กว้าง 55 cm. ยาว 68 cm. สูง 51 cm และราง aluminium profile ระบบทำความร้อนด้วย Aluminum heater plate และควบคุมอุณหภูมิด้วยบอร์ดและเซนเซอร์ ส่งถ่ายความร้อนไปยัง aluminum block ที่เจาะรูให้พอดีกับหลอดทดลอง 53 ช่องระบบปุ่มสวิทช์และหน้าจอควบคุม/แสดงผล LCD Display 3.5 นิ้ว ควบคุมโดยบอร์ด Arduino Mega 2560/KFB2.0

ผลการทำงานของเครื่องเมื่อทำการสอบเทียบปริมาตรชุดจ่ายของเหลวมีความถูกต้องแม่นยำสูง ด้วยต้นทุน 120,000 บาท ซึ่งถูกกว่าซื้อเครื่องมือจากต่างประเทศซึ่งมีราคาประมาณ 2 ล้านบาท สามารถผลิตให้เจ้าหน้าที่วิเคราะห์สารตกค้างมูลนิธิโครงการหลวงได้ใช้งานประจำ

ผลการประเมินความพึงพอใจ กลุ่มตัวอย่างที่เป็นผู้ปฏิบัติการวิเคราะห์สารตกค้างประจำศูนย์/สถานี มีความพึงพอใจอยู่ในระดับมาก (เฉลี่ย 4.2) และในแต่ละด้าน มีความพึงพอใจในระดับมากที่สุดทุกหัวข้อ เช่นเดียวกัน โดยมีความพึงพอใจสูงสุดคือ ด้านคุณค่าด้านการตอบสนองตรงต่อความต้องการของเจ้าหน้าที่วิเคราะห์สารตกค้าง นำไปปฏิบัติงานได้จริง และมีความปลอดภัยต่อการใช้งาน สำหรับระดับความพึงพอใจต่ำสุดคือ รูปลักษณะภายนอก เนื่องจากตัวเครื่องไม่ได้ตกแต่งด้านนอกด้วยกล่องครอบ แต่อย่างไรก็ตาม การไม่ได้ตกแต่งรูปลักษณะภายนอกไม่มีผลกระทบต่อการทำงานและยังลดต้นทุนการผลิต การใช้งานเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติสามารถลดต้นทุนการตรวจวิเคราะห์ได้ถึง 50% หรือประมาณ 15 บาทต่อตัวอย่าง

5.2 ข้อเสนอแนะ

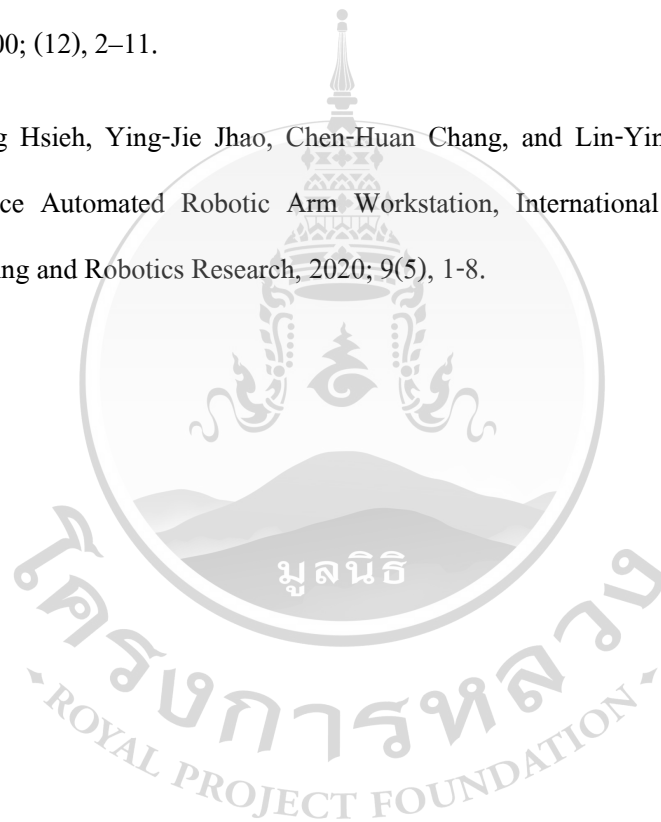
จากพัฒนาเครื่อง การประเมินประสิทธิภาพเครื่อง และการประเมินความพึงพอใจจากผู้ใช้งาน มีความพึงพอใจต่อเครื่องวัดค่าสี ในระดับมาก จึงมีคุณลักษณะตรงวัตถุประสงค์ที่ต้องการและมีประสิทธิภาพเหมาะสมต่อการใช้งาน คือ ความถูกต้องแม่นยำสูง ทำงานเองเป็นอัตโนมัติ จึงลดความคลาดเคลื่อนจากการอ่านด้วยสายตาได้เป็นอย่างดี โครงสร้างทำจากวัสดุที่มีความแข็งแรงคงทน สวยงาม ขนาดและน้ำหนักเหมาะสมสำหรับปฏิบัติงาน มีความสวยงามในการออกแบบ สามารถใช้ได้ทั้ง งานวิเคราะห์สารตกค้างด้วยชุดทดสอบ GT จึงมีความเหมาะสมต่อการนำไปใช้ในการปฏิบัติงานประจำวัน มีประสิทธิภาพและเป็นประโยชน์โดยตรงต่อผู้ปฏิบัติงาน

อย่างไรก็ตาม การทำงานด้วยระบบ micro controller จะต้องหล่อเลี้ยงด้วยกระแสไฟฟ้าตลอดระยะเวลาการทำงาน หากไฟฟ้าขัดข้อง เช่น ไฟตก ไฟดับ จะทำให้ระบบหยุดทำงาน และไม่สามารถทำงานต่อไปได้ จึงต้องมีอุปกรณ์สำรองไฟฟ้าต่อพ่วงกับตัวเครื่องด้วย

เอกสารอ้างอิง

- [1] กอบทอง รุฬาทอม.2547. SOP No.10 02 113.การวิเคราะห์สารเคมีกำจัดศัตรูพืชกลุ่มสารประกอบฟอสเฟตและ/หรือสารคาร์บาเมทในผักและผลไม้ โดยใช้ชุดน้ำยาตรวจหายาฆ่าแมลง ชนิดสำเร็จรูป. กองอาหาร กรมวิทยาศาสตร์การแพทย์ กระทรวงสาธารณสุข.
- [2] แม้น อมรสิทธิ์ และคณะ. *หลักการและเทคนิคการวิเคราะห์เชิงเครื่องมือ*. พิมพ์ครั้งที่ 1 ชวนพิมพ์. กรุงเทพฯ, 2553, หน้า 187-389.
- [3] Baltussen AH. New concepts in sorption based sample preparation for chromatography. Eindhoven: Technische Universiteit Eindhoven; New concepts in sorption based sample preparation for chromatography; 2000.
- [4] Ahmed FE. Analyses of pesticides and their metabolites in foods and drinks. Trends Anal Chem. 2001; 20:649-660.
- [5] LeBlanc G. A Review of EPA Sample Preparation Techniques for Organic Compound Analysis of Liquid and Solid Samples. LCGC. 2001; 19:1120-1130.
- [6] ศูนย์บริการเครื่องมือวิทยาศาสตร์. แก๊สโครมาโทกราฟี-แมสสเปกโตรเมตรี. [ออนไลน์]. เข้าถึงได้จาก : <http://www.kmitl.ac.th/sisc/GC-MS/main.html> สืบค้น 12 พค. 2561.
- [12] Zhang L, Shaowen Liu S, Cui X, Pan C,Zhang A, Chen F. A review of sample preparation methods for the pesticide residue analysis in foods. Cent Eur J Chem. 2012; 10(3):900-925.
- [13] Sheng Tang S, Hong Zhang H, Hian Kee Lee HK. Advances in sample extraction. Anal Chem. 2016; 88:228 – 249.
- [14] Vidya K, Maheshwara-Rao VU, Ajitha A. A review on current trends and advancements in bioanalytical method. Int J Phama Res Anal. 2014; 4(6):369-38.
- [15] Reddy KV, Yachawad A. Overview on recent extraction techniques in bioanalysis. Int Res J Pharm. 2016; 7(2):15-24.
- [16] Tan GH, Chai MK. Sample preparation in the analysis of pesticides residue in food by chromatographic techniques. In: Stoytcheva M, editors. Pesticides-Strategies for pesticides analysis. Rijeka: Intech; 2011. p. 27-58.

- [17] T.D. Lekkas, *et al* .Optimization of analytical methods for the determination of trace concentrations of toxic pollutants in drinking and surface waters . Global Nest: the Int. J. 2003; 5 (3), 165-175.
- [18] Yulkifli, P Kahar, R Ramli, S B Etika, C Imawan. Development of color detector using colorimetry system with photodiode sensor for food dye determination application. Journal of Physics Conf. Series 1185,2019; 012031.
- [19] Okamoto, H.; Deuchi, K., Design of a Robotic Workstation for Automated Organic Synthesis, LRA, 2000; (12), 2–11.
- [20] Chia-Ying Hsieh, Ying-Jie Jhao, Chen-Huan Chang, and Lin-Yin Chen, Development of Intelligence Automated Robotic Arm Workstation, International Journal of Mechanical Engineering and Robotics Research, 2020; 9(5), 1-8.



ภาคผนวก ก

คู่มือประกอบการใช้งาน



คู่มือการใช้งาน

เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

1. วัตถุประสงค์เครื่องวัดค่าสี

เพื่อใช้เป็นคู่มือและเป็นข้อปฏิบัติในการใช้ เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

2. ขอบข่าย

ใช้เป็นคู่มือประกอบในการใช้งานกับ เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติใช้ในขั้นตอน วิเคราะห์สารตกค้างด้วยชุดทดสอบ GT

3. เอกสารอ้างอิง

3.1 กอบทอง รูปหอม,ชุดทดสอบยาฆ่าแมลงในอาหาร pesticide test kit

4. หลักการ

ชุดน้ำยาตรวจสอบยาฆ่าแมลง “จีที” และ “เอ็ม เจ ที เค” ถูกคิดค้น โดย กรมวิทยาศาสตร์การแพทย์ ใช้ในการวิเคราะห์สารพิษตกค้างและคัดกรองตัวอย่างที่ไม่ปลอดภัย จากค่า ความเป็นพิษโดยรวมของสารพิษตกค้างที่มีอยู่ในตัวอย่างที่วิเคราะห์ ที่มีผลยับยั้งการทำงานของเอ็นไซม์ โคลีนเอสเตอเรสลดลงตั้งแต่ร้อยละ 50 ขึ้นไปเป็นเกณฑ์ หลักการทำงานของชุดตรวจหาฆ่าแมลง/ สารพิษตกค้างของชุดทดสอบใช้หลักการการทำงานของ Cholinesterase Inhibition technique ตรวจวัดระดับ ความเป็นพิษของสารพิษตกค้างโดยรวมทุกชนิดที่มีอยู่ในตัวอย่างที่ตรวจวิเคราะห์ ซึ่งปริมาณความเป็นพิษ ของสารพิษโดยรวมนั้น จะต้องมึปริมาณความเป็นพิษที่ตก ค้างอยู่ในตัวอย่างวิเคราะห์ไม่เกินค่าที่มีผลทำให้ ประสิทธิภาพการทำงานของเอ็นไซม์โคลีนเอสเตอเรสลดลงร้อยละ 50%

ในการทดสอบซึ่งต้องอาศัยความชำนาญในการดูค่า ในบางกรณีจะมีการหลงลืมหรือผิดพลาด จากผู้ทดสอบ ตลอดจนการเกิดความล้าเลงสงสัยในผล ดังนั้นจึงได้ทำการประดิษฐ์เครื่องทดสอบสารพิษ ตกค้าง เพื่อใช้ในการดูค่าของเหลวจากน้ำยา GT โดยอัตโนมัติทดแทนการทำงานด้วยคน เพื่อลดความ ผิดพลาดจาก Human error

เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ มีหลักการการทำงานคือ ใช้ microcontroller Arduino nano ควบคุมเซอร์โวมอเตอร์ควบคุมการหมุนของเฟืองเพื่อส่งกำลังให้เพลาคือที่ตามรางเลื่อน, Arduino

Mega 2560/KFB 2.0 ควบคุมสเตปเปอร์มอเตอร์ เพื่อควบคุมการกดไม้โครปีเปตุดูดจ่ายของเหลวจากน้ำยา GT-1 ถึง GT-5 ที่วางไว้ตามตำแหน่งที่กำหนดไว้ ตามปริมาตรและเวลาที่กำหนด มีระบบทำความร้อนด้วย Aluminum heater plate และควบคุมอุณหภูมิด้วยบอร์ดและเซนเซอร์ ส่งถ่ายความร้อนไปยัง aluminum block ที่เจาะรูให้พอดีกับหลอดทดลอง 53 ช่องระบบปั๊มสวิตซ์และหน้าจอควบคุม/แสดงผล LCD Display 3.5 นิ้ว



เครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

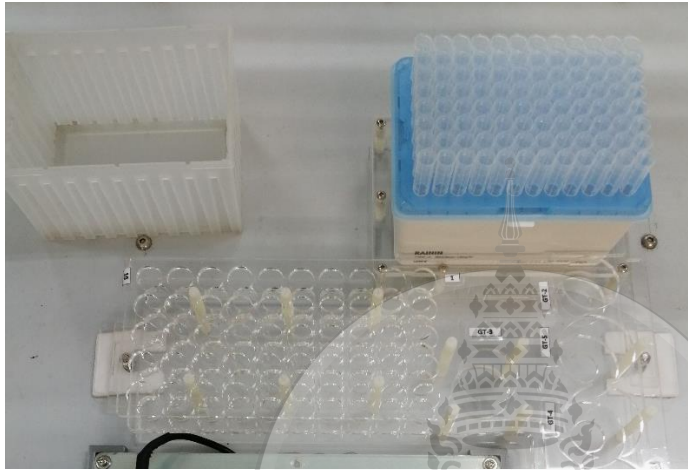
การพัฒนาเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัตินี้จึงเป็นการพัฒนาคิดค้นประดิษฐ์เครื่องมือออกมาให้เป็นนวัตกรรมใหม่ สามารถใช้ได้กับการทดสอบสารพิษตกค้างในอาหารประเภทผักสดและผลไม้สด ด้วยชุดตรวจหายาฆ่าแมลงในอาหาร (สารพิษตกค้าง) มุลนิธิฯ สามารถผลิตขึ้นใช้งานได้เองสามารถปฏิบัติงานได้จริงในห้องปฏิบัติการ เป็นการพัฒนา สนับสนุนงานอารักขาพืชด้านความปลอดภัยและการตรวจวัดเฝ้าระวังใช้สารเคมีกำจัดศัตรูพืช ทั้งนี้ยังสามารถนำผลิตภัณฑ์นำไปใช้ประโยชน์ให้กับหน่วยงานต่างๆที่ต้องการใช้ต่อไป ได้เครื่องมือที่มีประสิทธิภาพโดยไม่ต้องนำเข้าสั่งซื้อจากต่างประเทศ

5. ขั้นตอนการใช้งาน

5.1 ต่อเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติเข้ากับอุปกรณ์สำรองไฟ (UPS)

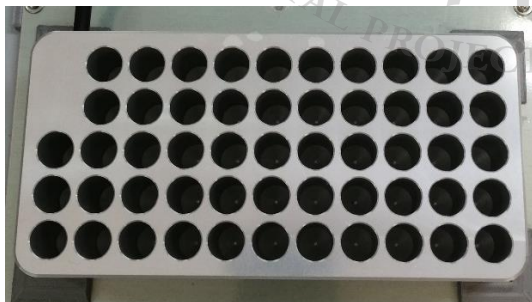
5.2 เสียบปลั๊กไฟฟ้า 220V 50Hz ของอุปกรณ์สำรองไฟ (UPS)

5.3 เติม Tip ให้เต็ม rack, วางภาชนะรองรับสำหรับทิ้ง Tip ที่ใช้แล้วตรงจุดที่ mark ไว้, วางหลอดสารสกัดตัวอย่างลงในช่องพลาสติก rack ซึ่งอยู่ด้านหลัง block aluminum



| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |
| 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 |
| 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 |
| 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 |

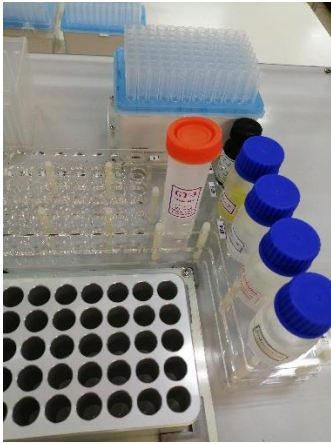
5.4 วางหลอดเปล่าไว้ในช่อง block aluminum



ลำดับช่องการวางหลอดเปล่า

| | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|
| | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 |
| 0 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 |
| +1 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 |
| +3 | 50 | 49 | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 |

5.5 เทน้ำยา GT 1-5 ในขวด Duran ที่เตรียมให้ ให้ของเหลวถึงระดับความสูงที่ 25 mL, วางลงในช่องที่ระบุไว้, เปิดฝาขวดออก



ตำแหน่งการวางขวดน้ำยา GT

| | |
|-----|-----------|
| | GT-2 |
| | GT-5 |
| GT3 | GT-4 |
| | GT-1 |
| | Solvent-2 |

5.1 กดสวิทช์เปิดตำแหน่ง on/off และ กดสวิทช์เปิดตำแหน่ง power, รอ heater ทำงาน 5 นาที



5.2 หมุนปุ่ม volume เพื่อตั้งค่าจำนวนหลอดตัวอย่าง แล้วกดปุ่ม select เพื่อ start เริ่มสั่งให้ทำงาน



6. ข้อควรระวัง

- 6.1 ระวังไม่ให้เกิดไฟฟ้าดับ
- 6.2 ความสูงของน้ำยา GT ต้องเติมให้อยู่ในระดับ 25 mL ตอนเริ่มต้น
- 6.3 Tip ต้องใช้ Pipette Tip, RAININ, 1000 uL, BioClean Ultra P/N: 30389164 และเติมให้เต็ม Rack เสมอ
- 6.4 Test tube ที่ใช้ต้องขนาด 13x100 mm, PYREX 9820-13, White marking spot

7. วิธีบำรุงรักษา

- 7.1 ควรจัดวางไว้ใน พื้นที่มั่นคง ไม่ควรสัมผัสน้ำโดยตรง
- 7.2 ทุก 1 ปี ต้องตรวจเช็คหรือเปลี่ยน stepper motor, servo motor, สายพาน

8. ผู้ประดิษฐ์ชิ้นงาน

นฤพล วัฒนภาพ และ วิฑูรย์ บุญสง่า

เบอร์โทร 0898508359, 088-529 0279

E-mail msn7029rpf@gmail.com



ภาคผนวก ข
แบบประเมินความพึงพอใจ



แบบสอบถาม ประเมินความพึงพอใจ

คำชี้แจง แบบประเมินนี้จัดทำขึ้นเพื่อรวบรวมข้อมูลและข้อเสนอแนะสำหรับการปรับปรุงพัฒนาการพัฒนาระบบเครื่องตรวจวิเคราะห์สารตกค้างระบบอัตโนมัติ

ส่วนที่ 1: ข้อมูลทั่วไป

- ชื่อ.....ตำแหน่ง..... ลिंगค์.....
- ประสบการณ์ด้านงานวิเคราะห์สารตกค้าง 1. 1-5 ปี 2. 6-10 ปี 3. 10 ปีขึ้นไป
- ท่านทำการอ่านสิทธิทดสอบเป็นประจำหรือไม่ 1. นานๆ ที 2. ทำบ้าง 3. เป็นประจำ

ส่วนที่ 2 : ความพึงพอใจต่อการใช้เครื่อง

โปรดทำเครื่องหมาย ✓ ลงในช่องที่ตรงกับความเป็นจริงมากที่สุด

| รายการ | ระดับความพึงพอใจ | | | | |
|--|------------------|----------|--------------|-----------|-----------------|
| | มากที่สุด 5 | มาก 4 | ปานกลาง 3 | น้อย 2 | น้อยที่สุด 1 |
| 1. ด้านประสิทธิภาพ และความถูกต้องแม่นยำในการทดสอบ | | | | | |
| 1.1 หลักการทำงานของเครื่องมือมีประสิทธิภาพสูง | | | | | |
| 1.2 สามารถช่วยให้วัดผลได้ถูกต้องแม่นยำในการทดสอบ | | | | | |
| 1.3 ลดปัญหา error จากการทำงานได้ ทำให้มีความเชื่อมั่นสูงขึ้น | | | | | |
| 2. ด้านความสะดวกในการใช้งาน | | | | | |
| 2.1 เครื่องมือใช้งานได้ง่าย สะดวก ไม่ซับซ้อน | | | | | |
| 2.2 เครื่องมือช่วยผ่อนแรง และลดขั้นตอนลง ทำให้ทำงานได้คล่อง | | | | | |
| 2.3 ความสะดวกในการดูแลทำความสะอาดหลังการใช้งาน | | | | | |
| 3. ด้านโครงสร้าง | | | | | |
| 3.1. โครงสร้างเครื่องมือ มีความแข็งแรง ทนทาน คงทน เหมาะกับงาน | | | | | |
| 3.2 รูปร่าง ขนาดของเครื่องมือมีความเหมาะสมกับงานทดสอบ | | | | | |
| 3.3 เครื่องมือมีรูปลักษณะภายนอกทันสมัย เหมาะสมกับงานทางวิทยาศาสตร์ | | | | | |
| 4. ด้านเวลาและค่าใช้จ่าย | | | | | |
| 4.1 ช่วยลดค่าใช้จ่ายและเวลาในการทำซ้ำเมื่อทำผิดพลาด(ลืม ทำผิด) | | | | | |
| 4.2 สามารถลดค่าใช้จ่ายจากค่าน้ำยา GT ลง 50 % | | | | | |
| 4.3 อายุการใช้งานเครื่องมากกว่า 5 ปี | | | | | |
| 5. ด้านคุณค่าด้านอื่น | | | | | |
| 5.1 มีประโยชน์ต่อผู้ปฏิบัติงาน ตอบสนองต่อความต้องการ นำไปใช้งานได้จริง | | | | | |
| 5.2 เครื่องมือปลอดภัยต่อผู้ใช้งาน | | | | | |
| 5.3 เกิดภาพลักษณ์ที่ดีต่อองค์กร ทันสมัย น่าภาคภูมิใจ | | | | | |

ส่วนที่ 3: ข้อเสนอแนะอื่นๆ

คำติ

คำชม

แนะนำเพื่อปรับปรุง

ลงชื่อ ผู้กรอกข้อมูลแบบสอบถาม

ขอขอบคุณที่ให้ความร่วมมือในการตอบแบบประเมิน

Aduino Code

```

#define THERMISTORNOMINAL 10000
#define TEMPERATURENOMINAL 25
#define BCOEFFICIENT 3950
#define SERIESRESISTOR 4700
#include <math.h>
#define pep_step 128
#define backlight 41
#define VR 14
#define buzzer 31
#define en_p1 18
#define en_p2 37
#define en_push 35
#define bt 33
#define heater_pin 10
#define ORIG_X_MAX_PIN 2
#define ORIG_Y_MAX_PIN 15
#define ORIG_Z_MIN_PIN 19
#define ORIG_X_MIN_PIN 3
#define ORIG_Y_MIN_PIN 14
#define ORIG_X_STEP_PIN 54
#define ORIG_X_DIR_PIN 55
#define ORIG_X_ENABLE_PIN 38
#define ORIG_Y_STEP_PIN 60
#define ORIG_Y_DIR_PIN 61
#define ORIG_Y_ENABLE_PIN 56
#define ORIG_Z_STEP_PIN 46
#define ORIG_Z_DIR_PIN 48
#define ORIG_Z_ENABLE_PIN 62
#define ORIG_E_STEP_PIN 26
#define ORIG_E_DIR_PIN 28
#define ORIG_E_ENABLE_PIN 24
#define tip_servo_pin 8
#define z_servo1_pin 9
#define z_servo2_pin 4
#define p_adj_up_pin 23
#define p_adj_down_pin 27
#define ENCODER_OPTIMIZE_INTERRUPTS
#define use_eep
// table zone
#define AAA 0
#define BBB 1
#define CCC 4
#define DDD 5
#define EEE 2
#define FFF 3
#include <Arduino.h>
#include <U8g2lib.h>
#include <SPI.h>

```



```

#include <Encoder.h>
#include <TimerOne.h>
#include <EEPROM.h>

const int servo_max = (112 * 5), servo_min = (-106 * 5);
uint16_t adc_t1[40];
uint8_t adc_index = 0;

const uint8_t THERMISTER_1_pin = 13;
Encoder knob(en_p1, en_p2);
U8G2_ST7920_128X64_F_HW_SPI u8g2(U8G2_R2, /* CS=*/53, /* reset=*/8);

uint8_t f_start = 0;
int servo_t_value = -20;
int servo_t_reject = 140;
int servo_z_value = -106;

int tip_count = 0;
int a_count = 0;
int b_count = 0;
int c_count = 0;
int d_count = 0;
int e_count = 0;
int f_count = 0;
struct MyObject
{
  uint16_t pipe_cal;
  uint16_t wait_time;
  uint16_t pipe_26;
  uint16_t pipe_40;
  uint16_t pipe_80;
  uint16_t lev_a;
  uint16_t lev_b;
  uint16_t lev_c;
  uint16_t lev_d;
  uint16_t lev_e;
  uint16_t lev_f;
  uint16_t lev_t1;
  uint16_t lev_t2;
  uint16_t lev_t3;
  int Kp;
  int Ki;
  int Kd;
  int Ki_min;
  int Ki_max;
  int temp_ref;
};

MyObject parameter;
struct MyObject_check
{
  uint8_t x;
  uint8_t y;
  uint8_t z;
  uint8_t p;
  uint8_t t;

```



```

};
MyObject_check Myaxis;
int sampling = 0;
float temp_ref;
uint8_t run = 0;
uint8_t test_value = 0;
uint8_t test_level = 0;
int count = 0;
uint8_t page = 0;
uint32_t t0, t1, t2, t3, t4, t5, buzzer_time;
uint32_t bt_count = 0;
uint8_t bt_press = 0;
uint8_t bt_hold = 0;
uint8_t bt_state = 0;
uint8_t bt_prev_state = 0;
uint8_t bt1_press = 0;
uint8_t bt1_state = 0;
uint8_t bt1_prev_state = 0;
uint8_t cerser = 0;
const int line[] = {8, 19, 30, 41, 52, 63};
int32_t test_tube_M1[50][2] = {0};
int32_t test_tube_M2[50][2] = {0};
int32_t test_tube_M2_s[3][2] = {0};
int32_t tip[192][2] = {0};
int32_t A_F[6][2] = {0};
int32_t A_F_lev[6] = {0};
uint8_t process = 0;
uint8_t sub_process = 0;
uint8_t sub2_process = 0;
uint8_t sub3_process = 0;
int32_t tip_drop[2] = {21000, 6400}; // 4000 31000
uint32_t show_wait_time = 0;
volatile int32_t x_mechine;
volatile int32_t y_mechine;
volatile int32_t z_mechine = servo_max;
volatile int32_t t_mechine;
volatile int32_t p_mechine;
volatile int32_t x_target;
volatile int32_t y_target;
volatile int32_t z_target = 0;
volatile int32_t t_target;
volatile int32_t p_target;
static uint8_t run_count;
volatile uint32_t start_time_for_wake;
float temp = 0;
uint16_t solution_count[6] = {0};
uint8_t cal_tip = 0;
int air_gab = 15;
int x_acc = 0;
int y_acc = 0;
int p_acc = 0;

```



```

float PID_error = 0;
float previous_error = 0;
int PID_value = 0;
float PID_p = 0.0;
float PID_i = 0.0;
void beep_ms(uint8_t ms)
{
    digitalWrite(buzzer, 1);
    buzzer_time = millis() + ms;
}
int x_acc_pre = 10;
int y_acc_pre = 10;
int p_acc_pre;
const int acc_config = 100;
const int acc_config_max = 8;
const int acc_config_min = 0;
const int acc_distance = 1000;
int8_t test_tip_Hold = 0;
int8_t fin = 0;
uint8_t re_screen = 1;
void setup(void)
{
    for (int index = 0; index < 50; index++)
    {
        test_tube_M1[index][0] = 13500 + (index % 10) * 1600; // x
        test_tube_M1[index][1] = 12000 + (index / 10) * 1600; // y
    }
    for (int index = 0; index < 50; index++)
    {
        test_tube_M2[index][0] = 12200 + (index % 10) * 1600; // x
        test_tube_M2[index][1] = 24250 + (index / 10) * 1600; // y
    }
    for (int index = 0; index < 96 * 2; index++)
    {
        if (index < 96)
        {
            tip[index][0] = 4170 + (index % 12) * 895; // x
            tip[index][1] = 2900 + (index / 12) * 895; // y
        }
        else
        {
            tip[index][0] = tip[index - 96][0] + 12900; // offset
            tip[index][1] = tip[index - 96][1]; // offset
        }
    }
    A_F[0][0] = 4900;
    A_F[0][1] = 27100;
    A_F[1][0] = 4900;
    A_F[1][1] = 23350;
    A_F[2][0] = 4900;
    A_F[2][1] = 19600;

```



```

A_F3[0] = 4900;
A_F3[1] = 15850;
A_F4[0] = 4900;
A_F4[1] = 12050;
A_F5[0] = 10500;
A_F5[1] = 17300;
test_tube_M2_s[0][0] = test_tube_M2[0][0] + (10 * 1600);
test_tube_M2_s[0][1] = test_tube_M2[0][1] + (2 * 1600);
test_tube_M2_s[1][0] = test_tube_M2[0][0] + (10 * 1600);
test_tube_M2_s[1][1] = test_tube_M2[0][1] + (3 * 1600);
test_tube_M2_s[2][0] = test_tube_M2[0][0] + (10 * 1600);
test_tube_M2_s[2][1] = test_tube_M2[0][1] + (4 * 1600);
Serial.begin(115200);
u8g2.begin();
u8g2.clearBuffer();
u8g2.sendBuffer();
pinMode(p_adj_up_pin, INPUT_PULLUP);
pinMode(p_adj_down_pin, INPUT_PULLUP);
pinMode(13, OUTPUT);
pinMode(heater_pin, OUTPUT);
pinMode(buzzer, OUTPUT);
pinMode(en_p1, INPUT_PULLUP);
pinMode(en_p2, INPUT_PULLUP);
pinMode(en_push, INPUT_PULLUP);
pinMode(bt, INPUT_PULLUP);
pinMode(en_p1, INPUT_PULLUP);
pinMode(en_p2, INPUT_PULLUP);
pinMode(en_push, INPUT_PULLUP);
pinMode(ORIG_X_MAX_PIN, INPUT_PULLUP);
pinMode(ORIG_Y_MAX_PIN, INPUT_PULLUP);
pinMode(ORIG_Z_MIN_PIN, INPUT_PULLUP);
pinMode(ORIG_X_STEP_PIN, OUTPUT);
pinMode(ORIG_X_DIR_PIN, OUTPUT);
pinMode(ORIG_X_ENABLE_PIN, OUTPUT); //y
digitalWrite(ORIG_X_DIR_PIN, HIGH);
pinMode(ORIG_Y_STEP_PIN, OUTPUT);
pinMode(ORIG_Y_DIR_PIN, OUTPUT);
pinMode(ORIG_Y_ENABLE_PIN, OUTPUT);
pinMode(ORIG_Z_STEP_PIN, OUTPUT);
pinMode(ORIG_Z_DIR_PIN, OUTPUT);
pinMode(ORIG_Z_ENABLE_PIN, OUTPUT);
digitalWrite(ORIG_Z_DIR_PIN, HIGH); //x
pinMode(ORIG_E_STEP_PIN, OUTPUT);
pinMode(ORIG_E_DIR_PIN, OUTPUT);
pinMode(ORIG_E_ENABLE_PIN, OUTPUT);
for (int adc_index = 0; adc_index < 40; adc_index++)
{
  adc_t1[adc_index] = analogRead(THERMISTER_1_pin);
}
EEPROM.get(0, parameter);

```



```

pinMode(blacklight, OUTPUT);
digitalWrite(blacklight, HIGH);
int raw_vr = map(analogRead(VR) + analogRead(VR)) / 2, 0, 1023, 0, 880);
sampling = raw_vr / 20;
go_home();
Timer1.initialize(65);
Timer1.attachInterrupt(hw_handle);
}
void loop(void)
{
t0 = millis();
if (t0 - t5 >= 5000)
{
t5 = t0;
if (fin == 1)
beep_ms(5);
}
if (t0 - t4 >= 1)
{
t4 = t0;
adc_t1[adc_index] = analogRead(THERMISTER_1_pin);
adc_index++;
if (adc_index >= 40)
adc_index = 0;
digitalWrite(13, 1);
digitalWrite(13, 0);
}
if (buzzer_time < t0)
{
digitalWrite(buzzer, 0);
}
if (t0 - t1 >= 100)
{
t1 = t0;
if (digitalRead(p_adj_down_pin) == 0)
p_mechine -= 1500;
if (digitalRead(p_adj_up_pin) == 0)
p_mechine += 1500;
////////////////////////////////////pid////////////////////////////////////
temp = (int)((float)get_tr1() * 10.0f);
if ((temp < 10) || (temp > 1000))
temp = 0;
if (temp != 0)
{
PID_error = (float)parameter.temp_ref * 0.1f - ((float)temp * 0.1f);
//Calculate the P value
PID_p = ((float)parameter.Kp * 1.0f) * PID_error;
PID_i = PID_i + (((float)parameter.Ki * 0.1f) * PID_error * 0.01f);
if (PID_i > ((float)parameter.Ki_max * 1.0f))
PID_i = ((float)parameter.Ki_max * 1.0f);
if (PID_i < -((float)parameter.Ki_max * 1.0f))

```



```

PID_i = -((float)parameter.Ki_max * 1.0f);
PID_value = PID_p + PID_i;
if (PID_value < 0)
    PID_value = 0;
if (PID_value > 200)
    PID_value = 200;
analogWrite(heater_pin, PID_value);
}
else
{
    analogWrite(heater_pin, 0);
    PID_i = 0;
    PID_value = 0;
}
// Serial.print(" temp_ref= ");
// Serial.print((float)parameter.temp_ref * 0.1f);
// Serial.print(" temp= ");
// Serial.print((float)temp * 0.1f);
// Serial.print(" PID_error= ");
// Serial.print(PID_error);
// Serial.print(" PID_value= ");
// Serial.print(PID_value);
// Serial.print(" PID_i= ");
// Serial.print(PID_i);
// Serial.print(" Kp= ");
// Serial.print(((float)parameter.Kp * 1.0f));
// Serial.print(" Ki= ");
// Serial.print(((float)parameter.Ki * 0.1f));
// Serial.println("");
// Serial.print(" p_target= ");
// Serial.print((p_target));
// Serial.print(" p_mechine= ");
// Serial.print((p_mechine));
// Serial.println("");
////////////////////////////////////pid////////////////////////////////////
bt_prev_state = bt_state;
bt_state = 1 - digitalRead(bt);
if (bt_state == 0 && bt_prev_state == 1 && bt_hold == 0 && bt_press == 0)
{
    beep_ms(2);
    bt_press = 1;
    re_screen = 1;
#ifdef use_eep
    EEPROM.put(0, parameter);
#endif
}
if (bt_count == 10)
{
    beep_ms(100);
    bt_hold = 1;
    re_screen = 1;

```



```

#ifdef use_eep
    EEPROM.put(0, parameter);
#endif

}

bt1_prev_state = bt1_state;
bt1_state = 1 - digitalRead(en_push);
if (bt1_state == 0 && bt1_prev_state == 1)
{
    beep_ms(2);
    bt1_press = 1;
    re_screen = 1;
#ifdef use_eep
    EEPROM.put(0, parameter);
#endif
}

else
{
    bt1_press = 0;
}

int tmp1 = knob.read() / 4;
int raw_vr = map((analogRead(VR) + analogRead(VR)) / 2, 0, 1023, 0, 880);
if (run == 0)
{
    if ((raw_vr - (sampling * 20)) < -15)
    {
        sampling--;
        re_screen = 1;
    }
    if ((raw_vr - (sampling * 20)) > 15)
    {
        sampling++;
        re_screen = 1;
    }
}

u8g2.clearBuffer();
u8g2.setFont(u8g2_font_profont11_tr);
char tmp[100];
const int l1 = 8, l2 = 19, l3 = 30, l4 = 41, l5 = 52, l6 = 63;
if (bt_hold == 0 && run == 0)
    page = constrain(page + tmp1, 0, 4);
static int l6_t temp_tmp, show_wait_time_tmp, run_tmp;
if (re_screen != 0 || tmp1 != 0 || temp_tmp != temp || show_wait_time_tmp != show_wait_time || run != run_tmp)
{
    re_screen = 0;
    temp_tmp = temp;
    show_wait_time_tmp = show_wait_time;
    run_tmp = run;
    switch ((uint8_t)page)
    {
        case 0:

```



```

static uint8_t bt_press_tmp0;
if (bt_press == 1 && bt_press_tmp0 == 0)
{
    run = 1 - run;
    re_screen = 1;
}
bt_press_tmp0 = bt1_press;
sprintf(tmp, "Main Page   T=%2d.%dC", (int)temp / 10, (int)(temp) % 10);
u8g2.drawStr(0, 11, tmp);
sprintf(tmp, " Number of sample");
u8g2.drawStr(0, 12, tmp);
sprintf(tmp, "Unit");
u8g2.drawStr(90, 15, tmp);
sprintf(tmp, "Wait %2d Min", show_wait_time / 60000);
u8g2.drawStr(0, 16, tmp);
show_wait_time = 0;
if (run)
{
    sprintf(tmp, "Stop");
    u8g2.drawStr(98, 16, tmp);
}
else
{
    sprintf(tmp, "Start");
    u8g2.drawStr(98, 16, tmp);
}
sprintf(tmp, "%d", sampling);
u8g2.setFont(u8g2_font_ncenB24_tn);
u8g2.drawStr(45, 49, tmp);
break;
case 1:
static uint8_t bt_press_tmp1;
if (bt_press == 1 && bt_press_tmp1 == 0)
{
    test_value = 1 - test_value;
}
bt_press_tmp1 = bt1_press;
static uint8_t bt1_press_tmp1;
if (bt1_press == 1 && bt1_press_tmp1 == 0)
{
    cerser = cerser + 1;
    re_screen = 1;
}
if (cerser > 4)
    cerser = 0;
bt1_press_tmp1 = bt1_press;
sprintf(tmp, ">>");
u8g2.drawStr(0, line[cerser + 1], tmp);
if (bt_hold)
{
    switch ((uint8_t)cerser)

```



```

{
case 0:
parameter.pipe_cal = (double)parameter.pipe_cal + tmp1;
break;
case 1:
parameter.pipe_cal = (double)parameter.pipe_cal + tmp1;
break;
case 2:
parameter.pipe_cal = (double)parameter.pipe_cal + tmp1;
break;
case 3:
parameter.pipe_cal = (double)parameter.pipe_cal + tmp1;
break;
case 4:
parameter.pipe_cal = (double)parameter.pipe_cal + tmp1;
break;
}
}
sprintf(tmp, "Pipette Calibration");
u8g2.drawStr(0, 11, tmp);
sprintf(tmp, " 0.20ml = %d", (uint32_t)((float)parameter.pipe_cal * 0.2f));
u8g2.drawStr(0, 12, tmp);
sprintf(tmp, " 0.22ml = %d", (uint32_t)((float)parameter.pipe_cal * 0.22f));
u8g2.drawStr(0, 13, tmp);
sprintf(tmp, " 0.26ml = %d", (uint32_t)((float)parameter.pipe_cal * 0.26f));
u8g2.drawStr(0, 14, tmp);
sprintf(tmp, " 0.40ml = %d", (uint32_t)((float)parameter.pipe_cal * 0.4f));
u8g2.drawStr(0, 15, tmp);
sprintf(tmp, " 0.80ml = %d", (uint32_t)((float)parameter.pipe_cal * 0.8f));
u8g2.drawStr(0, 16, tmp);
sprintf(tmp, "Test");
u8g2.drawStr(100, 16, tmp);
break;
case 2:
static uint8_t bt_press_tmp2;
if (bt_press == 1 && bt_press_tmp2 == 0)
{
if (process == 0)
test_level = 1 - test_level;
}
bt_press_tmp2 = bt1_press;
static uint8_t bt1_press_tmp2;
if (bt1_press == 1 && bt1_press_tmp2 == 0)
{
censer = censer + 1;
re_screen = 1;
}
if (censer > 4)
censer = 0;
bt1_press_tmp2 = bt1_press;
sprintf(tmp, ">>");

```



```

u8g2.drawStr(0, line[cerser + 1], tmp);
if (bt_hold)
{
switch ((uint8_t)cerser)
{
case 0:
parameter.lev_a = parameter.lev_a + tmp1;
break;
case 1:
parameter.lev_b = parameter.lev_b + tmp1;
break;
case 2:
parameter.lev_c = parameter.lev_c + tmp1;
break;
case 3:
parameter.lev_d = parameter.lev_d + tmp1;
break;
case 4:
parameter.lev_e = parameter.lev_e + tmp1;
break;
}
}
sprintf(tmp, "Level Setting 1");
u8g2.drawStr(0, 11, tmp);
sprintf(tmp, " Ex-2 = %d", parameter.lev_a);
u8g2.drawStr(0, 12, tmp);
sprintf(tmp, " GT-1 = %d", parameter.lev_b);
u8g2.drawStr(0, 13, tmp);
sprintf(tmp, " GT-2 = %d", parameter.lev_c);
u8g2.drawStr(0, 14, tmp);
sprintf(tmp, " GT-3 = %d", parameter.lev_d);
u8g2.drawStr(0, 15, tmp);
sprintf(tmp, " GT-4 = %d", parameter.lev_e);
u8g2.drawStr(0, 16, tmp);
sprintf(tmp, "Test");
u8g2.drawStr(100, 16, tmp);
break;
case 3:
static uint8_t bt_press_tmp3;
static uint8_t bt1_press_tmp3;
if (bt1_press == 1 && bt1_press_tmp3 == 0)
{
cerser = cerser + 1;
re_screen = 1;
}
bt1_press_tmp3 = bt1_press;
if (cerser > 4)
cerser = 0;
if (bt_press == 1 && bt_press_tmp3 == 0)
{
if (process == 0 && cerser != 4)

```

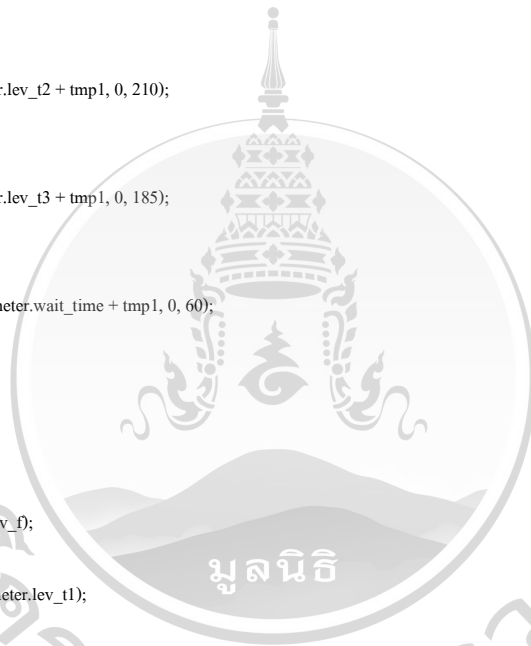


มูลนิธิโครงการหลวง
ROYAL PROJECT FOUNDATION

```

    test_level = 1 - test_level;
}
bt_press_tmp3 = bt1_press;
sprintf(tmp, ">>");
u8g2.drawStr(0, line[cerser + 1], tmp);
if (bt_hold)
{
    switch ((uint8_t)cerser)
    {
    case 0:
        parameter.lev_f = parameter.lev_f + tmp1;
        break;
    case 1:
        parameter.lev_t1 = constrain(parameter.lev_t1 + tmp1, 0, 217);
        break;
    case 2:
        parameter.lev_t2 = constrain(parameter.lev_t2 + tmp1, 0, 210);
        break;
    case 3:
        parameter.lev_t3 = constrain(parameter.lev_t3 + tmp1, 0, 185);
        break;
    case 4:
        parameter.wait_time = constrain(parameter.wait_time + tmp1, 0, 60);
        break;
    }
}
sprintf(tmp, "Level Setting 2");
u8g2.drawStr(0, 11, tmp);
sprintf(tmp, " GT-5 = %d", parameter.lev_f);
u8g2.drawStr(0, 12, tmp);
sprintf(tmp, " Test Tube 1 = %d", parameter.lev_t1);
u8g2.drawStr(0, 13, tmp);
sprintf(tmp, " Test Tube 2 = %d", parameter.lev_t2);
u8g2.drawStr(0, 14, tmp);
sprintf(tmp, " Tip = %d", parameter.lev_t3);
u8g2.drawStr(0, 15, tmp);
sprintf(tmp, " Wait Time = %d", parameter.wait_time);
u8g2.drawStr(0, 16, tmp);
sprintf(tmp, "Test");
u8g2.drawStr(105, 16, tmp);
break;
case 4:
    static uint8_t bt_press_tmp4;
    if (bt1_press == 1 && bt_press_tmp4 == 0)
    {
        cerser = cerser + 1;
        re_screen = 1;
    }
}
if (cerser > 4)
    cerser = 0;
bt_press_tmp4 = bt1_press;

```



มูลนิธิโครงการหลวง
ROYAL PROJECT FOUNDATION

```

sprintf(tmp, ">>");
u8g2.drawStr(0, line[cerser + 1], tmp);
if (bt_hold)
{
    switch ((uint8_t)cerser)
    {
    case 0:
        parameter.Kp = constrain(parameter.Kp + tmp1, 0, 177);
        break;
    case 1:
        parameter.Ki = constrain(parameter.Ki + tmp1, 0, 225);
        break;
    case 2:
        parameter.Kd = constrain(parameter.Kd + tmp1, 0, 210);
        break;
    case 3:
        parameter.Ki_max = constrain(parameter.Ki_max + tmp1, 1, 48);
        tip_count = (parameter.Ki_max - 1);
        break;
    case 4:
        parameter.temp_ref = constrain(parameter.temp_ref + tmp1, 0, 1000);
        break;
    }
}
sprintf(tmp, "Level Setting 3");
u8g2.drawStr(0, 11, tmp);
sprintf(tmp, "  Max Solution = %d", (int)parameter.Kp);
u8g2.drawStr(0, 12, tmp);
sprintf(tmp, "  Max GT-3 = %d", (int)parameter.Ki);
u8g2.drawStr(0, 13, tmp);
sprintf(tmp, "  Mix Solution = %d", (int)parameter.Kd);
u8g2.drawStr(0, 14, tmp);
sprintf(tmp, "  Tip Start = %d", (int)parameter.Ki_max);
u8g2.drawStr(0, 15, tmp);
sprintf(tmp, "  Set Temp = %d.%01d C", (int)parameter.temp_ref / 10, (int)parameter.temp_ref % 10);
u8g2.drawStr(0, 16, tmp);
break;
default:
    break;
}
u8g2.sendBuffer();
}
if (bt_state == 1)
{
    bt_count++;
}
else
{
    bt_count = 0;
    bt_hold = 0;
}

```

```

    bt_press = 0;
}
if (tmp1 != 0)
    knob.write(0);
}

if (t0 - t3 >= 20)
{
    t3 = t0;
    update_AF_lev();
    if (abs(x_mechine - x_target) > acc_distance)
    {
        if (x_acc_pre > acc_config_min)
        {
            x_acc_pre--;
        }
    }
    else
    {
        if (x_acc_pre < acc_config_max)
        {
            x_acc_pre++;
        }
    }
    if (abs(y_mechine - y_target) > acc_distance)
    {
        if (y_acc_pre > acc_config_min)
        {
            y_acc_pre--;
        }
    }
    else
    {
        if (y_acc_pre < acc_config_max)
        {
            y_acc_pre++;
        }
    }
}

if (abs(p_mechine - p_target) > 400)
{
    if (p_acc_pre > 0)
    {
        p_acc_pre--;
    }
}
else
{
    if (p_acc_pre < 2)
    {
        p_acc_pre++;
    }
}

```



```

}
}
// Serial.print("xf= ");
// Serial.print(x_target);
// Serial.print(" xm= ");
// Serial.print(x_mechine);
// Serial.print(" yf= ");
// Serial.print(y_target);
// Serial.print(" ym= ");
// Serial.print(y_mechine);
// Serial.print(" zf= ");
// Serial.print(z_target);
// Serial.print(" zm= ");
// Serial.print(z_mechine);
// Serial.print(" page= ");
// Serial.print(page);
// Serial.print(" process= ");
// Serial.print(process);
// Serial.print(" sub_process= ");
// Serial.print(sub_process);
// Serial.println("");
// Serial.print("aa= ");
// Serial.print(solution_count[AAA]);
// Serial.print(" bb= ");
// Serial.print(solution_count[BBB]);
// Serial.print(" cc= ");
// Serial.print(solution_count[CCC]);
// Serial.print(" dd= ");
// Serial.print(solution_count[DDD]);
// Serial.print(" ee= ");
// Serial.print(solution_count[EEE]);
// Serial.print(" ff= ");
// Serial.print(solution_count[FFF]);
// Serial.print(" z_target= ");
// Serial.print(z_target);
// Serial.print(" z_mechine= ");
// Serial.print(z_mechine);
// Serial.println(p_mechine);
if (Myaxis.x == 1 && Myaxis.y == 1)
{
    const int max_step = 2;
    if (z_mechine == z_target)
    {
        Myaxis.z = 1;
    }
    else if (z_target > z_mechine)
    {
        Myaxis.z = 0;
        if (abs(z_target - z_mechine) > max_step)
        {
            z_mechine += max_step;

```



```

}
else
{
    z_mechine++;
}
}
else
{
    Myaxis.z = 0;
    if (abs(z_mechine - z_target) > max_step)
    {
        z_mechine -= max_step;
    }
    else
    {
        z_mechine--;
    }
}
}
else
{
    Myaxis.z = 0;
}
if (t_mechine == t_target)
{
    Myaxis.t = 1;
}
else if (t_mechine < t_target)
{
    Myaxis.t = 0;

    if (abs(t_target - t_mechine) > 10)
    {
        t_mechine = t_mechine + 10;
    }
    else
    {
        t_mechine++;
    }
}
}
else
{
    if (abs(t_target - t_mechine) > 10)
    {
        t_mechine = t_mechine - 10;
    }
    else
    {
        t_mechine--;
    }
}
}

```



```

int servo_t_target = constrain(servo_t_value + t_mechine, -120, 120);
int servo_z_target = constrain(servo_z_value + z_mechine, -120, 120);
write_3_servo(servo_t_target, servo_z_target);
}
if (t0 - t2 >= 40)
{
t2 = t0;
if (run == 1 && fin == 0)
{
switch ((uint8_t)process)
{
case 0: ////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
if (cal_tip == 1)
{
if (drop_tip(tip_count))
{
tip_count++;
sub_process++;
cal_tip = 0;
}
}
else
{
//if (go_to(1000, 2000, 0))
sub_process++;
}
break;
case 1:
if (get_tip(tip_count))
sub_process++;
break;
case 2: // ฤดู
if (get_solution_from_test_tube(run_count, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 3:
if (drop_solution_2_test_tube(run_count, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 4: // ปล่อย
if (drop_tip(tip_count))
{
sub_process++;
tip_count++;
}
break;
case 5: // fin
sub_process = 0;

```



```

run_count++;
break;
}
if (run_count >= sampling)
{
run_count = 0;
sub_process = 0;
process++;
}
break;
case 1: //////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
if (get_tip(tip_count))
sub_process++;
break;
case 1: // ฝึก
if (get_solution_from_AF_tube(AAA, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 2:
if (drop_solution_2_sp_tube(0, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 3: // ฝึก
if (get_solution_from_AF_tube(AAA, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 4:
if (drop_solution_2_sp_tube(1, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 5: // ฝึก
if (get_solution_from_AF_tube(AAA, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 6:
if (drop_solution_2_sp_tube(2, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 7:
if (drop_tip(tip_count))
{
sub_process++;
tip_count++;
}
break;
case 8: // fin
sub_process = 0;
process++;

```



```

    break;
}
break;
case 2: ///////////////////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
    if (get_tip(tip_count))
        sub_process++;
    break;
case 1: // ฉุกเฉิน
    if (get_solution_from_AF_tube(BBB, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 2:
    if (drop_solution_2_sp_tube(0, parameter.pipe_cal * 0.4f))
    {
        start_time_for_wake = millis(); //เริ่มจับเวลา
        sub_process++;
    }
    break;
case 3: // ฉุกเฉิน
    if (get_solution_from_AF_tube(BBB, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 4:
    if (drop_solution_2_sp_tube(1, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 5: // ฉุกเฉิน
    if (get_solution_from_AF_tube(BBB, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 6:
    if (drop_solution_2_sp_tube(2, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 7: // fin
    sub_process = 0;
    process++;
    break;
}
break;
case 3: ///////////////////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0: // ฉุกเฉิน
    if (get_solution_from_AF_tube(BBB, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 1:

```



```

if (drop_solution_2_test_tube(run_count, parameter.pipe_cal * 0.4f))
    sub_process++;
break;
case 2: // fin
    sub_process = 0;
    run_count++;
    break;
}
if (run_count >= sampling)
{
    run_count = 0;
    sub_process = 0;
    process++;
}
break;
case 4: ///////////////////////////////////////////////////////////////////
switch (sub_process)
{
case 0:
if (drop_tip(tip_count))
{
    sub_process++;
    tip_count++;
}
break;
case 1:
if (waiting_min(15))
{
    sub_process = 0;
    process++;
}
break;
}
break;
case 5: ///////////////////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
if (get_tip(tip_count))
    sub_process++;
break;
case 1: // ฉุกเฉิน
if (get_solution_from_AF_tube(CCC, parameter.pipe_cal * 0.26f))
    sub_process++;
break;
case 2:
if (drop_solution_2_sp_tube(0, parameter.pipe_cal * 0.26f))
{
    start_time_for_wake = millis(); //เริ่มจับเวลา
    sub_process++;
}
}

```



```

break;
case 3: // ๑๑๑
if (get_solution_from_AF_tube(CCC, parameter.pipe_cal * 0.22f))
sub_process++;
break;
case 4:
if (drop_solution_2_sp_tube(1, parameter.pipe_cal * 0.22f))
sub_process++;
break;
case 5: // ๑๑๑
if (get_solution_from_AF_tube(CCC, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 6:
if (drop_solution_2_sp_tube(2, parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 7: // fin
sub_process = 0;
process++;
break;
}
break;
case 6: ///////////////////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0: // ๑๑๑
if (get_solution_from_AF_tube(CCC, (double)parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 1:
if (drop_solution_2_test_tube(run_count, (double)parameter.pipe_cal * 0.2f))
sub_process++;
break;
case 2: // fin
sub_process = 0;
run_count++;
break;
}
if (run_count >= sampling)
{
run_count = 0;
sub_process = 0;
process++;
}
break;
case 7: ///////////////////////////////////////////////////////////////////
switch (sub_process)
{
case 0:

```



```

if (drop_tip(tip_count))
{
    sub_process++;
    tip_count++;
}
break;
case 1:
if (waiting_min(parameter.wait_time))
{
    sub_process = 0;
    process++;
}
break;
}
break;
case 8: ////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
if (get_tip(tip_count))
    sub_process++;
break;
case 1: // ฤๅ
if (get_solution_from_AF_tube(DDD, parameter.pipe_cal * 0.8f))
{
    sub_process++;
}
break;
case 2:
if (drop_solution_2_sp_tube(0, parameter.pipe_cal * 0.8f))
{
    sub_process++;
}
break;
case 3: // ฤๅ
if (get_solution_from_AF_tube(DDD, parameter.pipe_cal * 0.8f))
{
    sub_process++;
}
break;
case 4:
if (drop_solution_2_sp_tube(1, parameter.pipe_cal * 0.8f))
    sub_process++;
break;
case 5: // ฤๅ
if (get_solution_from_AF_tube(DDD, parameter.pipe_cal * 0.8f))
{
    sub_process++;
}
break;
case 6:

```



```

if (drop_solution_2_sp_tube(2, parameter.pipe_cal * 0.8f))
    sub_process++;
break;
case 7: // fin
    run_count = 0;
    sub_process = 0;
    process++;
    break;
}
break;
case 9: ////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0: // ฤๅ
    if (get_solution_from_AF_tube(DDD, parameter.pipe_cal * 0.8f))
    {
        sub_process++;
    }
    break;
case 1:
    if (drop_solution_2_test_tube(run_count, parameter.pipe_cal * 0.8f))
        sub_process++;
    break;
case 2: // fin
    sub_process = 0;
    run_count++;
    break;
}
if (run_count >= sampling)
{
    run_count = 0;
    sub_process = 0;
    process++;
}
break;
case 10:
switch ((uint8_t)sub_process)
{
case 0:
    if (drop_tip(tip_count))
    {
        sub_process = 0;
        tip_count++;
        process++;
    }
    break;
}
break;
case 11: ////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{

```



```

case 0:
    if (get_tip(tip_count))
    {
        sub_process++;
    }
    break;
case 1: // ฤๅ
    if (get_solution_from_AF_tube(EEE, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 2:
    if (drop_mix_solution_2_sp_tube(run_count,parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 3:
    if (mix_get_drop_solution_from_sp_tube(run_count, parameter.pipe_cal * 0.8f))
        sub_process++;
    break;
case 4: // ปลั๊ก
    if (drop_tip(tip_count))
    {
        sub_process++;
        tip_count++;
    }
    break;
case 5: // fin
    run_count++;
    sub_process = 0;
    break;
}
if (run_count >= 3)
{
    run_count = 0;
    sub_process = 0;
    process++;
}
break;
case 12: //////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0:
    if (get_tip(tip_count))
    {
        sub_process++;
    }
    break;
case 1: // ฤๅ
    if (get_solution_from_AF_tube(EEE, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 2:

```



```

if (drop_mix_solution_2_test_tube(run_count,parameter.pipe_cal * 0.4f))
    sub_process++;
break;
case 3:
if (mix_get_drop_solution_from_test_tube(run_count, parameter.pipe_cal * 0.8f))
    sub_process++;
break;
case 4: // ปล่อย
if (drop_tip(tip_count))
{
    sub_process++;
    tip_count++;
}
break;
case 5: // fin
run_count++;
sub_process = 0;
break;
}
if (run_count >= sampling)
{
    run_count = 0;
    sub_process = 0;
    process++;
}
break;
case 13:
switch ((uint8_t)sub_process)
{
case 0:
if (get_tip(tip_count))
{
    sub_process = 0;
    process++;
}
break;
}
break;
case 14: //////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0: // ฝึก
if (get_solution_from_AF_tube(FFF, parameter.pipe_cal * 0.4f))
    sub_process++;
break;
case 1:
if (drop_solution_2_sp_tube(0, parameter.pipe_cal * 0.4f))
{
    sub_process++;
}
}
break;

```



```

case 2: // ฤๅ
    if (get_solution_from_AF_tube(FFF, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 3:
    if (drop_solution_2_sp_tube(1, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 4: // ฤๅ
    if (get_solution_from_AF_tube(FFF, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 5:
    if (drop_solution_2_sp_tube(2, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 6: // fin
    sub_process = 0;
    process++;
    break;
}
break;
case 15: ///////////////////////////////////////////////////////////////////
switch ((uint8_t)sub_process)
{
case 0: // ฤๅ
    if (get_solution_from_AF_tube(FFF, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 1:
    if (drop_solution_2_test_tube(run_count, parameter.pipe_cal * 0.4f))
        sub_process++;
    break;
case 2: // fin
    sub_process = 0;
    run_count++;
    break;
}
if (run_count >= sampling)
{
    run_count = 0;
    sub_process = 0;
    process++;
}
break;
case 16: ///////////////////////////////////////////////////////////////////
switch (sub_process)
{
case 0:
    if (drop_tip(tip_count))
    {

```



```

    sub_process++;
    tip_count++;
}
break;
case 1:
if (go_to(1000, 2000, 1))
{
    sub_process++;
}
break;
case 2:
go_to(1000, 2000, servo_max);
sub_process = 0;
process = 0;
run = 0;
fin = 1;
break;
}
break;
}
else if (test_value == 1)
{
static uint16_t pipe_tmp;
switch (sub2_process)
{
case 0:
if (cal_tip == 0)
{
if (get_tip(tip_count))
{
sub2_process++;
cal_tip = 1;
}
}
else
{
sub2_process++;
}
break;
case 1:
switch (censer)
{
case 0:
pipe_tmp = (double)parameter.pipe_cal * 0.2f;
break;
case 1:
pipe_tmp = (double)parameter.pipe_cal * 0.22f;
break;
case 2:
pipe_tmp = (double)parameter.pipe_cal * 0.26f;

```



```

break;
case 3:
    pipe_tmp = (double)parameter.pipe_cal * 0.4f;
    break;
case 4:
    pipe_tmp = (double)parameter.pipe_cal * 0.8f;
    break;
}
if (get_solution_from_test_tube(49, pipe_tmp))
    sub2_process++;
break;
case 2:
if (drop_solution_2_test_tube(49, pipe_tmp))
    sub2_process++;
break;
case 3:
if (go_to(1000, 2000, 1))
    sub2_process++;
break;
case 4:
sub2_process = 0;
test_value = 0;
break;
}
}
else if (test_level == 1)
{
switch (sub3_process)
{
case 0:
if (cal_tip == 0)
{
if (get_tip(tip_count))
{
sub3_process++;
cal_tip = 1;
}
}
else
{
sub3_process++;
}
break;
case 1:

switch ((uint8_t)page)
{
case 2:
switch (censer)
{
case 0:

```



```

if (get_solution_from_AF_tube(AAA, 1))
    sub3_process++;
break;
case 1:
if (get_solution_from_AF_tube(BBB, 1))
    sub3_process++;
break;
case 2:
if (get_solution_from_AF_tube(CCC, 1))
    sub3_process++;
break;
case 3:
if (get_solution_from_AF_tube(DDD, 1))
    sub3_process++;
break;
case 4:
if (get_solution_from_AF_tube(EEE, 1))
    sub3_process++;
break;
}
break;
case 3:
switch (censer)
{
case 0:
if (get_solution_from_AF_tube(FFF, 1))
    sub3_process++;
break;
case 1:
if (get_solution_from_test_tube(0, 1))
    sub3_process++;
break;
case 2:
if (drop_solution_2_test_tube(0, 1))
    sub3_process++;
break;
case 3:
test_tip_Hold = 1;
// if (get_tip(0))
// {
sub3_process++;
// }
break;
}
break;
}
break;
case 2:
if (go_to(1000, 2000, 1))
    sub3_process++;
break;

```



โครงการหลวง
ROYAL PROJECT FOUNDATION

```

case 3:
    sub3_process = 0;
    test_level = 0;
    if (test_tip_Hold == 1)
    {
        cal_tip = 0;
        test_tip_Hold = 0;
    }
    break;
}
}
else
{
}
}
}

void go_home(void)
{
    int go_zero = 0;
    while (go_zero == 0)
    {
        if (digitalRead(ORIG_Z_MIN_PIN) == 0)
            digitalWrite(ORIG_Z_STEP_PIN, HIGH);
        if (digitalRead(ORIG_Y_MAX_PIN) == 0)
            digitalWrite(ORIG_X_STEP_PIN, HIGH);
        if (digitalRead(ORIG_X_MAX_PIN) == 0)
            digitalWrite(ORIG_Y_STEP_PIN, HIGH);
        delayMicroseconds(100);
        digitalWrite(ORIG_Z_STEP_PIN, LOW);
        digitalWrite(ORIG_X_STEP_PIN, LOW);
        digitalWrite(ORIG_Y_STEP_PIN, LOW);
        if ((digitalRead(ORIG_Z_MIN_PIN) == 1) && (digitalRead(ORIG_Z_MIN_PIN) == 1) && (digitalRead(ORIG_X_MAX_PIN) == 1) &&
(digitalRead(ORIG_X_MAX_PIN) == 1) && (digitalRead(ORIG_Y_MAX_PIN) == 1) && (digitalRead(ORIG_Y_MAX_PIN) == 1))
            go_zero = 1;
        delayMicroseconds(100);
    }
}

void hw_handle(void)
{
    static uint8_t x1, x2, x3;
    if (f_start == 0)
        go_to(1000, 2000, 0);
    if (run == 1 || f_start == 0 || test_level == 1 || test_value == 1)
    {
        x_acc++;
        if (x_acc > x_acc_pre)
        {
            x_acc = 0;
            if (x_mechine == x_target)
            {
                Myaxis.x = 1;
            }
        }
    }
}

```

```

x_acc_pre = acc_config_max;
}
else if (x_target > x_mechine)
{
x_mechine++;
Myaxis.x = 0;
digitalWrite(ORIG_Z_DIR_PIN, LOW);
digitalWrite(ORIG_Y_DIR_PIN, HIGH);
x1 = 1 - x1;
digitalWrite(ORIG_Z_STEP_PIN, x1);
digitalWrite(ORIG_Y_STEP_PIN, x1);
}
else
{
x_mechine--;
Myaxis.x = 0;
digitalWrite(ORIG_Z_DIR_PIN, HIGH);
digitalWrite(ORIG_Y_DIR_PIN, LOW);
x1 = 1 - x1;
digitalWrite(ORIG_Z_STEP_PIN, x1);
digitalWrite(ORIG_Y_STEP_PIN, x1);
}
}
y_acc++;
if (y_acc > y_acc_pre)
{
y_acc = 0;
if (y_mechine == y_target)
{
Myaxis.y = 1;
f_start = 1;
y_acc_pre = acc_config_max;
}
else if (y_target > y_mechine)
{
y_mechine++;
Myaxis.y = 0;
digitalWrite(ORIG_X_DIR_PIN, LOW);
x2 = 1 - x2;
digitalWrite(ORIG_X_STEP_PIN, x2);
}
else
{
y_mechine--;
Myaxis.y = 0;
digitalWrite(ORIG_X_DIR_PIN, HIGH);
x2 = 1 - x2;
digitalWrite(ORIG_X_STEP_PIN, x2);
}
}
}
}

```



```

p_acc++;
if (p_acc > p_acc_pre)
{
p_acc = 0;
if (p_mechine == p_target)
{
Myaxis.p = 1;
}
else if (p_target > p_mechine)
{
p_mechine++;
Myaxis.p = 0;
digitalWrite(ORIG_E_DIR_PIN, LOW);
x3 = 1 - x3;
digitalWrite(ORIG_E_STEP_PIN, x3);
}
else
{
p_mechine--;
Myaxis.p = 0;
digitalWrite(ORIG_E_DIR_PIN, HIGH);
x3 = 1 - x3;
digitalWrite(ORIG_E_STEP_PIN, x3);
}
}
}
uint8_t get_tip(int number)
{
uint8_t tmp = 0;
static uint8_t sub_process;
switch ((uint8_t)sub_process)
{
case 0:
if (go_to(tip[number][0], tip[number][1], parameter.lev_t3))
sub_process++;
break;
case 1:
if (go_to(tip[number][0], tip[number][1], 0))
sub_process++;
break;
case 2:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
uint8_t drop_tip(int number)
{
uint8_t tmp = 0;
static uint8_t sub_process;

```



```

switch ((uint8_t)sub_process)
{
case 0:
if (go_to(tip_drop[0], tip_drop[1], 10))
sub_process++;
break;
case 1:
if (set_servo_tip(servo_t_reject))
sub_process++;
break;
case 2:
if (set_servo_tip(0))
sub_process++;
break;
case 3:
if (go_to(tip_drop[0], tip_drop[1], 0))
sub_process++;
break;
case 4:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
uint8_t get_solution_from_AF_tube(int solution_number, int volume)
{
uint8_t tmp = 0;
static uint8_t run_count;
static uint8_t sub_process;
int level_tmp = 0;
int32_t x_tmp = A_F[solution_number][0];
if (run == 1)
{
switch ((uint8_t)solution_number)
{
case AAA:
level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[AAA] * 0.45f), 0, parameter.Kp); // start 133 max 177
break;
case BBB:
level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[BBB] * 0.9f), 0, parameter.Kp); // start 133 max 177
break;
case CCC:
level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[CCC] * 0.9f), 0, parameter.Kp); //max 177
break;
case DDD:
level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[DDD] * 3.2f), 0, parameter.Ki); // start 65 max 225
break;
case EEE:
level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[EEE] * 0.9f), 0, parameter.Kp); // start 133 max 177

```



```

    break;
case FFF:
    level_tmp = constrain(A_F_lev[solution_number] + (int)((float)solution_count[FFF] * 0.9f), 0, parameter.Kp); // start 133 max 177
    break;
}
}
else
{
    level_tmp = A_F_lev[solution_number];
    x_tmp = 1000;
}
switch ((uint8_t)sub_process)
{
case 0:
    set_servo_pep(volume + air_gab);
    sub_process++;
    break;
case 1:
    if (go_to(x_tmp, A_F[solution_number][1], level_tmp) && set_servo_pep(volume + air_gab))
        sub_process++;
    break;
case 2:
    if (set_servo_pep(air_gab))
        sub_process++;
    break;
case 3:
    if (go_to(x_tmp, A_F[solution_number][1], 10))
        sub_process++;
    break;
case 4:
    if (set_servo_pep(0))
        sub_process++;
    break;
case 5:
    if (go_to(x_tmp, A_F[solution_number][1], 0))
        sub_process++;
    break;
case 6:
    tmp = 1;
    sub_process = 0;
    if (run == 1)
        solution_count[solution_number]++;
    break;
}
return tmp;
}
uint8_t get_solution_from_test_tube(int solution_number, int volume)
{
    uint8_t tmp = 0;
    static uint8_t run_count;

```

```

static uint8_t sub_process;
switch (((uint8_t)sub_process)
{
case 0:
    set_servo_pep(volume + air_gab);
    sub_process++;
    break;
case 1:
    if (go_to(test_tube_M1[solution_number][0], test_tube_M1[solution_number][1], parameter.lev_t1) && set_servo_pep(volume + air_gab))
        sub_process++;
    break;
case 2:
    if (set_servo_pep(air_gab))
        sub_process++;
    break;
case 3:
    if (go_to(test_tube_M1[solution_number][0], test_tube_M1[solution_number][1], 10))
        sub_process++;
    break;
case 4:
    if (set_servo_pep(0))
        sub_process++;
    break;
case 5:
    if (go_to(test_tube_M1[solution_number][0], test_tube_M1[solution_number][1], 0))
        sub_process++;
    break;
case 6:
    tmp = 1;
    sub_process = 0;
    break;
}
return tmp;
}
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
uint8_t mix_get_drop_solution_from_test_tube(int solution_number, int volume)
{
    uint8_t tmp = 0;
    static uint8_t run_count;
    static uint8_t sub_process;
    switch (((uint8_t)sub_process)
    {
    case 0:
        set_servo_pep(volume + air_gab);
        sub_process++;
        break;
    case 1:
        if (set_servo_pep(volume + air_gab))
            sub_process++;
        break;
    case 2:

```

```

if ( go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], parameter.Kd))
    sub_process++;
break;
case 3:
if (set_servo_pep(air_gab))
    sub_process++;
break;
case 4:
if (go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], parameter.Kd))
    sub_process++;
break;
case 5:
if (set_servo_pep(volume + air_gab))
    sub_process++;
break;
case 6:
if (go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], 0))
    sub_process++;
break;
case 7:
// if (set_servo_pep(0))
set_servo_pep(0);
sub_process++;
break;
case 8:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
uint8_t mix_get_drop_solution_from_sp_tube(int solution_number, int volume)
{
uint8_t tmp = 0;
static uint8_t run_count;
static uint8_t sub_process;
switch (((uint8_t)sub_process)
{
case 0:
set_servo_pep(volume + air_gab);
sub_process++;
break;
case 1:
if (set_servo_pep(volume + air_gab))
sub_process++;
break;
case 2:
if ( go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], parameter.Kd))
sub_process++;
break;
case 3:

```



```

if (set_servo_pep(air_gab))
    sub_process++;
break;
case 4:
if (go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], parameter.Kd))
    sub_process++;
break;
case 5:
if (set_servo_pep(volume + air_gab))
    sub_process++;
break;
case 6:
if (go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], 0))
    sub_process++;
break;
case 7:
// if (set_servo_pep(0))
set_servo_pep(0);
sub_process++;
break;
case 8:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
uint8_t drop_solution_2_test_tube(int solution_number, int volume)
{
uint8_t tmp = 0;
static uint8_t run_count;
static uint8_t sub_process;
switch ((uint8_t)sub_process)
{
case 0:
if (go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], parameter.lev_t2))
    sub_process++;
break;
case 1:
if (set_servo_pep(volume + air_gab * 3))
    sub_process++;
break;
case 2:
if (go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], 0))
    sub_process++;
break;
case 3:
// if (set_servo_pep(0))
set_servo_pep(0);
sub_process++;

```



```

break;
case 4:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
uint8_t drop_solution_2_sp_tube(int solution_number, int volume)
{
uint8_t tmp = 0;
static uint8_t run_count;
static uint8_t sub_process;
switch ((uint8_t)sub_process)
{
case 0:
if (go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], parameter.lev_t2))
sub_process++;
break;
case 1:
if (set_servo_pep(volume + air_gab * 3))
sub_process++;
break;
case 2:
if (go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], 0))
sub_process++;
break;
case 3:
// if (set_servo_pep(0))
set_servo_pep(0);
sub_process++;
break;
case 4:
tmp = 1;
sub_process = 0;
break;
}
return tmp;
}
uint8_t drop_mix_solution_2_test_tube(int solution_number, int volume)
{
uint8_t tmp = 0;
static uint8_t run_count;
static uint8_t sub_process;

switch ((uint8_t)sub_process)
{
case 0:
if (go_to(test_tube_M2[solution_number][0], test_tube_M2[solution_number][1], parameter.lev_t2))
sub_process++;
break;

```



```

case 1:
    if (set_servo_pep(volume + air_gab))
        sub_process++;
    break;
case 2:
    // if (set_servo_pep(0))
    set_servo_pep(0);
    sub_process++;
    break;
case 3:
    tmp = 1;
    sub_process = 0;
    break;
}
return tmp;
}
uint8_t drop_mix_solution_2_sp_tube(int solution_number, int volume)
{
    uint8_t tmp = 0;
    static uint8_t run_count;
    static uint8_t sub_process;
    switch ((uint8_t)sub_process)
    {
    case 0:
        if (go_to(test_tube_M2_s[solution_number][0], test_tube_M2_s[solution_number][1], parameter.lev_t2))
            sub_process++;
        break;
    case 1:
        if (set_servo_pep(volume + air_gab))
            sub_process++;
        break;
    case 2:
        // if (set_servo_pep(0))
        set_servo_pep(0);
        sub_process++;
        break;
    case 3:
        tmp = 1;
        sub_process = 0;
        break;
    }
    return tmp;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
uint8_t go_to(int32_t x, int32_t y, int32_t z)
{
    uint8_t tmp = 0;
    x_target = x * 2;
    y_target = y * 2;
    z_target = z;
}

```



```

if (z_mechine != z_target)
{
    Myaxis.z = 0;
}
if (Myaxis.x == 1 && Myaxis.y == 1)
    if (Myaxis.z == 1)
        tmp = 1;
return tmp;
}
uint8_t set_servo_tip(int32_t val)
{
    uint8_t tmp = 0;
    t_target = val;
    if (t_mechine != t_target)
    {
        Myaxis.t = 0;
    }
    if (Myaxis.t == 1)
        tmp = 1;
    return tmp;
}
uint8_t set_servo_pep(int32_t val)
{
    uint8_t tmp = 0;
    p_target = -val * pep_step;
    if (p_mechine != p_target)
    {
        Myaxis.p = 0;
    }
    if (Myaxis.p == 1)
        tmp = 1;
    return tmp;
}
void update_AF_lev(void)
{
    A_F_lev[0] = parameter.lev_a;
    A_F_lev[1] = parameter.lev_b;
    A_F_lev[2] = parameter.lev_e;
    A_F_lev[3] = parameter.lev_f;
    A_F_lev[4] = parameter.lev_c;
    A_F_lev[5] = parameter.lev_d;
}
uint8_t waiting_min(uint32_t m)
{
    uint8_t tmp = 0;
    show_wait_time = (m * 60000) + start_time_for_wake - millis();
    if (millis() - start_time_for_wake >= (m * 60000))
    {
        tmp = 1;
    }
}

```



```

return tmp;
}
float get_tr1(void)
{
uint32_t sum = 0;
for (int adc_index = 0; adc_index < 40; adc_index++)
{
sum += adc_t1[adc_index];
}
return Thermistor((float)sum / 40);
}
float Thermistor(float average)
{
//average = 1023 / average - 1; // pulldown
average = average / (1023 - average);
average = SERIESRESISTOR * average;
float steinhart;
steinhart = average / THERMISTORNOMINAL; // (R/Ro)
steinhart = log(steinhart); // ln(R/Ro)
steinhart /= BCOEFFICIENT; // 1/B * ln(R/Ro)
steinhart += 1.0 / (TEMPERATURENOMINAL + 273.15); // + (1/To)
steinhart = 1.0 / steinhart; // Invert
steinhart -= 273.15; // convert to C
steinhart = (int)(steinhart * 10);
steinhart /= 10;
return steinhart;
}
void write_3_servo(int16_t val1, int16_t val2)
{
val2 = constrain(val2, servo_min / 5, servo_max / 5);
val2 = val2 * 5;
int16_t servo_write1 = val1 * 5;
int16_t servo_write2 = val2 - 100 + map(constrain(val2, 0, servo_max), 0, servo_max, 0, 5) + map(constrain(val2, servo_min, 0), servo_min, 0, 18, 0);
int16_t servo_write3 = -val2;
uint8_t tmp[6] = {0};
tmp[0] = (uint8_t)((uint16_t)servo_write1 >> 8);
tmp[1] = (uint8_t)servo_write1;
tmp[2] = (uint8_t)((uint16_t)servo_write2 >> 8);
tmp[3] = (uint8_t)servo_write2;
tmp[4] = (uint8_t)((uint16_t)servo_write3 >> 8);
tmp[5] = (uint8_t)servo_write3;
Serial.write(tmp[0]);
Serial.write(tmp[1]);
Serial.write(tmp[2]);
Serial.write(tmp[3]);
Serial.write(tmp[4]);
Serial.write(tmp[5]);
Serial.write(0x79);
Serial.write(0x80);
Serial.write(0x81);
}

```