

การใช้งานโปรโตคอลมอดบัส RTU และ TCP โดยใช้ ESP32 สำหรับตัวควบคุมแบบฟัซซี่

Implementation of Modbus RTU and Modbus TCP Protocol using ESP32 for Fuzzy Controller

สมพร เตียเจริญ*

Somporn Tiacharoen *

ภาควิชาเทคโนโลยีวิศวกรรมอิเล็กทรอนิกส์ วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Department of Electronics Engineering Technology, College of Industrial Technology,

King Mongkut's University of Technology North Bangkok

*Email: tiacharoen@yahoo.com

Received: March 13, 2020; Revised: May 28, 2020; Accepted: June 19, 2020

บทคัดย่อ

การใช้งานโปรโตคอลมอดบัส RTU และ TCP โดยใช้ ESP32 สำหรับตัวควบคุมแบบฟัซซี่ได้ถูกนำเสนอในบทความนี้ ตัวควบคุมแบบฟัซซี่ถูกออกแบบมาสำหรับระบบควบคุมความเร็วของมอเตอร์กระแสตรงระยะไกลที่อยู่คนละตำแหน่งโดยใช้โปรโตคอลมอดบัส TCP (Modbus TCP) ตัวควบคุมแบบฟัซซี่จะใช้ไมโครคอนโทรลเลอร์ Arduino MEGA2560 ทำหน้าที่เป็นมอดบัส RTU (Modbus RTU) พารามิเตอร์ต่างๆของการควบคุมแบบฟัซซี่สามารถตั้งค่าผ่านการสื่อสารแบบมอดบัส โดยมอดบัส TCP ตัวควบคุมหลักจะสามารถกำหนดค่าให้กับตัวควบคุมตัวลูกได้ซึ่งต้นแบบที่สร้างขึ้นมาจะใช้ไมโครคอนโทรลเลอร์ ESP32 ผู้ใช้งานจะสื่อสารกับตัวควบคุมหลักผ่านโปรโตคอลมอดบัส TCP และตัวควบคุมตัวลูกสื่อสารกับตัวควบคุมหลักผ่านการสื่อสารแบบมอดบัส ระบบควบคุมความเร็วของมอเตอร์กระแสตรงระยะไกลและจากผลการทดสอบพบว่าการทำงานของระบบควบคุมสอดคล้องตามหลักทฤษฎี

คำสำคัญ : มอดบัส, มอดบัส TCP, ไมโครคอนโทรลเลอร์ ESP32, ระบบควบคุมความเร็วของมอเตอร์กระแสตรง

Abstract

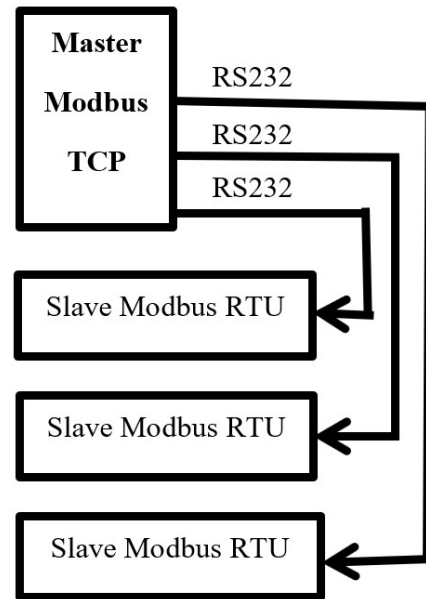
In this article, a fuzzy controller is designed for the speed control of a DC motor, and then is applied to a remote tracking system based on Modbus TCP protocol. The fuzzy controller is implemented onto an Arduino MEGA2560 microcontroller as a slave Modbus RTU. The parameters of fuzzy control can set via the Modbus communication. The master Modbus TCP is able to assign values to a controller, this master implemented onto an ESP32 microcontroller. The user communicates with the master over a TCP protocol and the slaves communicate with the master over a Modbus communication. The tracking system and results verify that the Implementation of control system is satisfactory and reliable.

Keywords : Modbus, Modbus TCP, ESP32 microcontroller, speed control of a DC motor

1. บทนำ

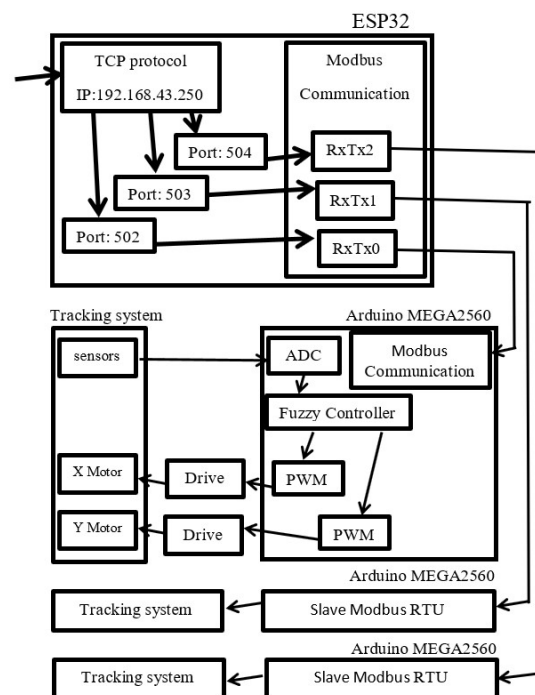
มอดบัส (Modbus) เป็นคือโปรโตคอลที่ใช้การสื่อสารแบบอนุกรม [1] ปกติจะใช้งานกับ PLC (Programmable Logic Controllers) ซึ่งเป็นส่วนสำคัญในระบบควบคุมและเครื่องมือวัดในอุตสาหกรรม ใช้ติดต่อสื่อสารกันเป็นเครือข่ายแบบมีสายด้วยมาตรฐาน RS232 RS422 หรือ RS485 เป็นต้น ซึ่งในปัจจุบันการสื่อสารผ่านเครือข่ายไร้สายมีการใช้งานอย่างกว้างขวางมากขึ้น [2] แต่ระบบในอุตสาหกรรมเดิมส่วนใหญ่ยังเป็นแบบมอดบัสแบบใช้สาย มีการนำระบบควบคุมระยะไกลด้วยอินเทอร์เน็ต modbus / TCP โดยใช้โปรแกรม PCAuto ซึ่งเป็นซอฟต์แวร์สำหรับการกำหนดค่า SCADA ส่งให้ตัวควบคุมพีไอดีเพื่อนำไปใช้กับระบบควบคุมอุณหภูมิระยะไกล [3] ซึ่งงานวิจัยนี้จะใช้มอดบัส RTU และโปรโตคอลมอดบัส TCP เช่นกันโดยใช้ไมโครคอนโทรลเลอร์ ESP32 เชื่อมต่อผ่านอินเทอร์เน็ต modbus / TCP และส่งค่าให้ควบคุมแบบพีซีซึ่งที่ผ่านมามีการออกแบบและการจำลองระบบควบคุมอุณหภูมิด้วยตัวควบคุมแบบพีซี - พีไอดีในโปรแกรม MATLAB และมีการใช้โปรโตคอล Modbus-RTU เพื่อตรวจสอบและควบคุมระบบ โดยส่งค่าอุณหภูมิที่ต้องการไปยังตัวควบคุม [4] และมีการพัฒนาตัวควบคุมพีซีบน PLC (Programmable Logic Controller) การเชื่อมต่อของ PLC กับเซิร์ฟเวอร์ OPC (Object Process Control) [5] แต่ไม่ได้มีการส่งค่าพารามิเตอร์ของตัวควบคุมไปเพื่อทำการปรับระบบของตัวควบคุมใหม่ซึ่งในงานวิจัยนี้จะทำการส่งค่าพารามิเตอร์ของตัวควบคุมไปยังหน่วยความจำที่กำหนดเพื่อให้ตัวควบคุมใช้ในการปรับรูปร่างของเซตพีซีในตัวควบคุม

ไดอะแกรมของระบบควบคุมโดยรวมสำหรับการนำไปประยุกต์ใช้งานแสดงดังรูปที่ 1 ซึ่งแสดงให้เห็นว่าตัวควบคุมหลัก (Master Modbus TCP) หนึ่งตัวสามารถสื่อสารกับตัวควบคุมลูก (Modbus RTU) หลาย ๆ ตัวได้ โดยระบบต้นแบบจะมีตัวควบคุมหลักหนึ่งตัวที่สามารถสื่อสารกับตัวควบคุมลูกใดๆที่เชื่อมต่อในระบบได้โดยเลือกได้จากการกำหนดค่าพอร์ทของโปรโตคอล TCP



รูปที่ 1 ไดอะแกรมของระบบควบคุม

ในการตั้งค่าระบบจะอยู่บนพื้นฐานของการสื่อสารแบบมอดบัส TCP ทั้งนี้มีส่วนประกอบหลักสองส่วนคือตัวควบคุมมอดบัสหลัก (Master Modbus) และตัวควบคุมมอดบัสลูก (Slave Modbus) สามารถดูได้ในรูปที่ 2 โดยในชั้นแอปพลิเคชันจะเป็นโปรโตคอลมอดบัส TCP / IP และในชั้นกายภาพของโปรโตคอล TCP / IP จะใช้การสื่อสารแบบ RS232 [6]

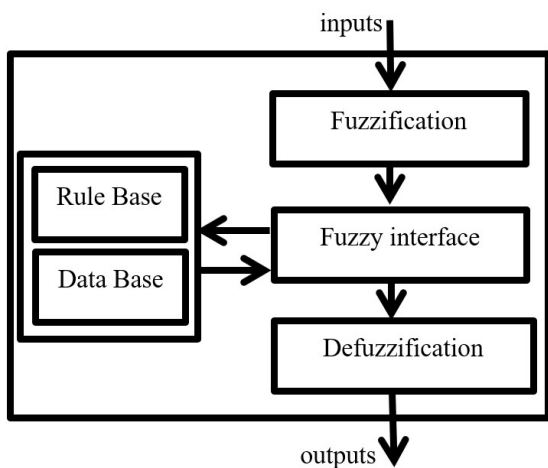


รูปที่ 2 ไดอะแกรมของการควบคุมและสื่อสารแบบมอดบัส

2. ตัวควบคุมแบบฟัซซี (Fuzzy Controller)

การควบคุมแบบฟัซซีมีหลายส่วน [7]: ข้อมูลกฎการควบคุม (Rule Base) กลไกการอนุมานฟัซซี (Fuzzy Inference Mechanism) การป้อนข้อมูลแบบฟัซซี (Input Fuzzification) และการหาผลลัพธ์ (Output Defuzzification) โครงสร้างของตัวควบคุมแบบฟัซซีนี้แสดงในรูปที่ 3 สำหรับการควบคุมฟัซซีขั้นพื้นฐาน LL หมายถึงสัญญาณอินพุตเซ็นเซอร์ด้านซ้าย LR แสดงถึงสัญญาณอินพุตเซ็นเซอร์ด้านขวา PWM แทนสัญญาณเอาต์พุตควบคุมความเร็ว โดยอินพุตและเอาต์พุตมีฟัซซีเซต 3 เซตคือ เซต L, เซต M และเซต H ทั้งนี้ TN จะแทนสัญญาณเอาต์พุตควบคุมทิศทางมอเตอร์มีฟัซซีเซต 3 เซตคือ เซต L, เซต S และเซต R ดังแสดงในรูปที่ 4

สำหรับตัวควบคุมแบบฟัซซีมีสัญญาณอินพุตทั้งหมดสองตัวคือ LL และ LR และโดยมีฟังก์ชันการเป็นสมาชิกฟัซซีแบบสามเหลี่ยม สำหรับแต่ละอินพุตจะมีกฎในการควบคุมทั้งหมด 9 กฎเพื่อควบคุมความเร็วมอเตอร์แต่ละตัว ซึ่งกฎในการควบคุมจะกำหนดไว้ในข้อมูลกฎการควบคุม (Rule Base) (ตารางที่ 1 สำหรับมอเตอร์แกน X และตารางที่ 2 สำหรับมอเตอร์แกน Y) ยกตัวอย่างจากกฎการควบคุมในตารางที่ 1 เช่น เมื่อมีสัญญาณอินพุตจากเซ็นเซอร์ด้านซ้าย (LL) อยู่ในฟัซซีเซตชื่อ M และมีสัญญาณอินพุตจากเซ็นเซอร์ด้านขวา (LR) อยู่ในฟัซซีเซตชื่อ H จะทำให้ตัวควบคุมสั่งให้มอเตอร์หมุน (PWM) ด้วยความเร็วของฟัซซีเซตชื่อ F และหมุนในทิศทาง (TN) ของฟัซซีเซตชื่อ R



รูปที่ 3 โครงสร้างของตัวควบคุมแบบฟัซซี

3. การดำเนินงาน

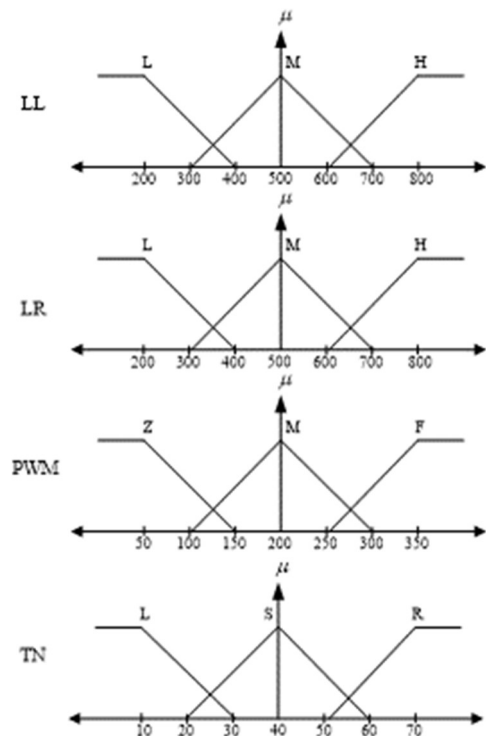
ตัวควบคุมหลักด้วยโปรโตคอลมอดบัส TCP / IP ที่แสดงดังในรูปที่ 5 ซึ่งตัวควบคุมหลักต้นแบบนี้พัฒนาด้วยไมโครคอนโทรลเลอร์ ESP32 [8]

พอร์ตของโปรโตคอลมอดบัส TCP / IP:

192.168.43.250:502 ---> COM0

192.168.43.250:503 ---> COM1

192.168.43.250:504 ---> COM2



รูปที่ 4 ฟังก์ชันการเป็นสมาชิกของการควบคุมแบบฟัซซี แกน y เป็นระดับการเป็นสมาชิกของฟัซซีเซตจาก 0 ถึง 1 และแกน x เป็นค่าอินพุตของแต่ละฟัซซีเซต

ตารางที่ 1 กฎการควบคุมฟัซซีสำหรับมอเตอร์แกน X

		LL		
		H	M	L
LR	H	TN=S & PWM=Z	TN=R & PWM=F	TN=R & PWM=F
	M	TN=L & PWM=F	TN=S & PWM=Z	TN=R & PWM=M
	L	TN=L & PWM=F	TN=L & PWM=M	TN=S & PWM=Z

ตารางที่ 2 กฎการควบคุมพีชสำหรับมอเตอร์แกน Y

		LL		
		H	M	L
LR	H	TN=S & PWM=Z	TN=R & PWM=F	TN=R & PWM=F
	M	TN=L & PWM=F	TN=S & PWM=Z	TN=R & PWM=M
	L	TN=L & PWM=F	TN=L & PWM=M	TN=S & PWM=Z



รูปที่ 5 ตัวควบคุมหลักด้วยโปรโตคอลมอดบัส TCP / IP

ตัวควบคุมลูกด้วยโปรโตคอลมอดบัส RTU แสดงในรูปที่ 6 ซึ่งตัวควบคุมลูกต้นแบบทุกตัวนี้พัฒนาด้วยไมโครคอนโทรลเลอร์ Arduino MEGA2560

พอร์ตของโปรโตคอลมอดบัส RTU:

debug <---> COM0

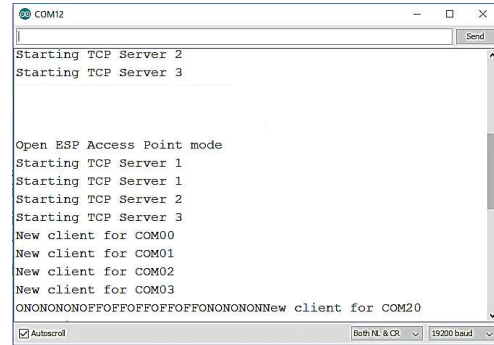
modbus ---> COM1



รูปที่ 6 ตัวควบคุมลูกด้วยโปรโตคอลมอดบัส RTU

3.1 ตัวควบคุมหลักด้วยโปรโตคอลมอดบัส TCP / IP

การเชื่อมต่อ TCP/IP จะใช้การเชื่อมต่อแบบไร้สายผ่านสัญญาณ WiFi เพื่อส่งข้อมูลไปยังรูปแบบการสื่อสารแบบอนุกรม (UART Bridge) ดังแสดงในรูปที่ 7

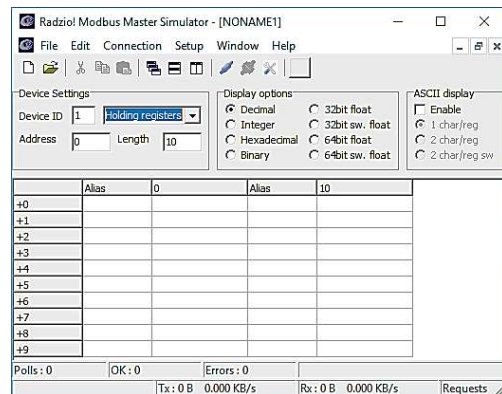


รูปที่ 7 การเชื่อมต่อ WiFi เพื่อส่งข้อมูลไปยังการสื่อสารแบบอนุกรม

การทดสอบเป็นการส่งข้อมูลผ่านทางเว็บเบราว์เซอร์ไปยังตัวควบคุมหลัก โดยกำหนดตำแหน่งที่อยู่ IP เป็น 192.168.43.250 และจะทำการเลือกพอร์ตที่ต้องการจะส่งข้อมูลไปยังเครื่องลูกข่ายที่ต้องการ เช่น ต้องการติดต่อกับเครื่องลูกหมายเลข 2 ก็จะกำหนดพอร์ตในการส่งข้อมูลเป็นหมายเลข 503 ซึ่งเครื่องลูกหมายเลข 2 จะต่อกับพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ESP32 ที่พอร์ต COM1 โดยใช้สายตามาตรฐาน RS232

3.2 ตัวควบคุมลูกด้วยโปรโตคอลมอดบัส

การเชื่อมต่อมอดบัสจะใช้การเชื่อมต่อการสื่อสารแบบอนุกรมด้วยมาตรฐาน RS-232 แสดงดังในรูปที่ 8



รูปที่ 8 การทดสอบการเชื่อมต่อมอดบัสด้วย RS-232

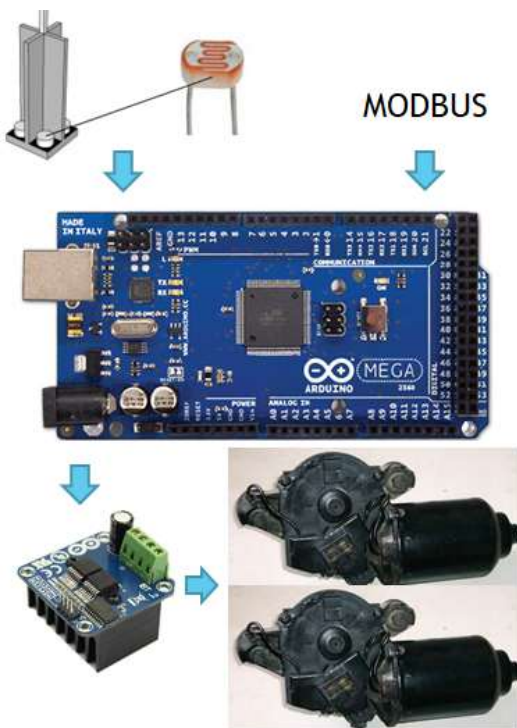
การทดสอบเป็นการส่งข้อมูลผ่านมอดบัสโดยใช้โปรแกรมจำลองตัวควบคุมหลัก โดยได้ทำการโปรแกรมให้ตัวควบคุมลูกมีระบบการควบคุมแบบพีชซึ่งอยู่ภายในซึ่งจะอ่านค่าจากโฮลตั้ง

รีจิสเตอร์(Holding Register)เพื่อใช้กำหนดค่าข้อมูลพารามิเตอร์ เริ่มต้นให้กับตัวควบคุมแบบพีซีซีก่อนที่จะเริ่มรูปในการควบคุม โดยการทดลองจะกำหนดให้ตัวควบคุมลูก RTU มีตำแหน่งเป็น 1 และทดลองเขียนค่าพารามิเตอร์ของพีซีซีในโฮลดีงรีจิสเตอร์ ภายในที่กำหนดไว้

แล้วทำการส่งต่างๆของการควบคุมแบบพีซีซีสามารถตั้งค่า ผ่านการสื่อสารแบบมอดบัส โดยมอดบัส TCP เลือกพอร์ตที่ต้องการจะส่งข้อมูลไปยังเครื่องลูกข่ายที่ต้องการ เช่น ต้องการติดต่อกับเครื่องลูกหมายเลข 2 ก็กำหนดพอร์ตในการส่งข้อมูล เป็นหมายเลข 503 ซึ่งเครื่องลูกหมายเลข 2 จะต่อกับพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ ESP32 ที่พอร์ต COM1 โดยใช้สายตามาตรฐาน RS232

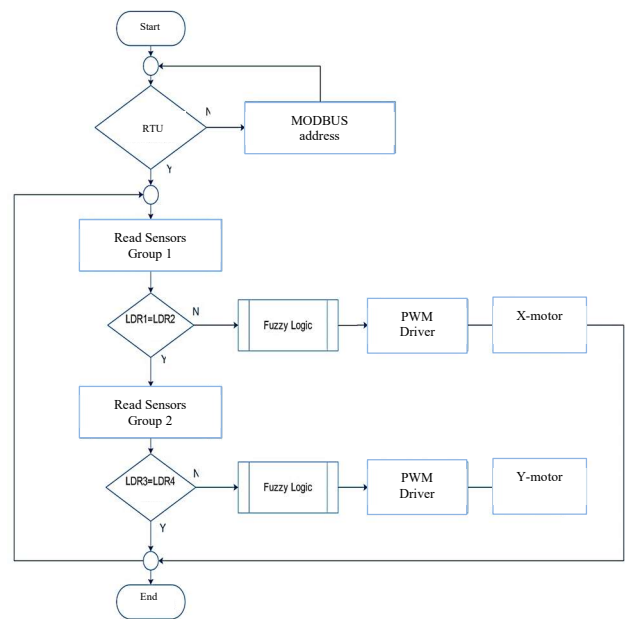
3.3 ระบบควบคุมความเร็วของมอเตอร์กระแสตรง

ตัวควบคุมตัวลู่กนอกจากจะมีโปรแกรมในส่วนการติดต่อกับมอดบัสแล้วยังมีโปรแกรมควบคุมความเร็วของมอเตอร์กระแสตรงด้วยพีซีซี โดยมีไดอะแกรมของการควบคุมดังรูปที่ 9



รูปที่ 9 ไดอะแกรมของระบบควบคุม

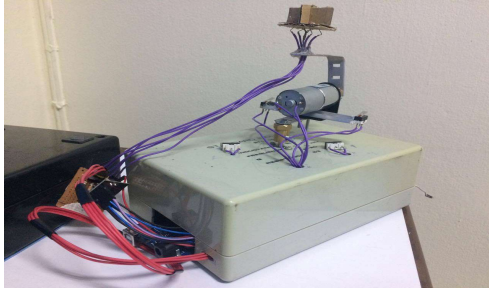
จากไดอะแกรมของระบบควบคุมจะมีอินพุตจากเซนเซอร์ทั้งหมด 2 กลุ่ม ซึ่งจะมีการรับค่าแบบอนาล็อกและส่งค่าไปประมวลผลที่ตัวควบคุมตัวลู่ก (Arduino MEGA2560) จากค่าที่อ่านได้จะส่งให้กับระบบควบคุมแบบพีซีซีเพื่อสร้างสัญญาณ PWM ให้กับมอเตอร์แกน X และแกน Y โดยใช้บอร์ดขับมอเตอร์ BTS7960B สำหรับมอเตอร์แต่ละตัว ความเร็วและทิศทางการหมุนของมอเตอร์จะเป็นไปตามระบบควบคุมแบบพีซีซี ซึ่งผังงาน (Flowchart)ระบบการควบคุมแสดงดังรูปที่ 10



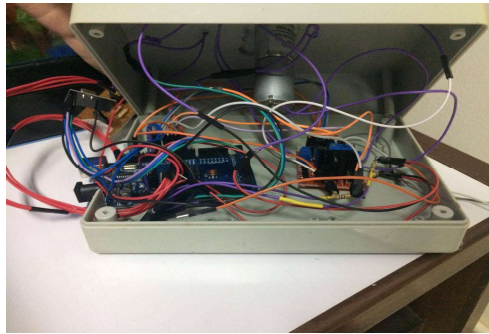
รูปที่ 10 ผังงานระบบการควบคุม

4. ผลการดำเนินงาน

จากรูปที่ 11(ก) และ (ข) เป็นรูปต้นแบบตัวควบคุมมอดบัสตัวลู่ก RTU หมายเลข 1 ประกอบด้วยบอร์ดควบคุม บอร์ดขับมอเตอร์และมอเตอร์ในแนวแกน X และแกน Y โดยเมื่อได้รับพารามิเตอร์เริ่มต้นจากตัวควบคุมมอดบัส TCP ตัวหลักแล้วตัวลู่กจะเข้าสู่ลูการควบคุมด้วยพีซีซี จากการทดลองเป็นไปตามเป้าหมายที่กำหนดไว้ โดยได้ทดลองกับต้นแบบตัวควบคุมมอดบัสตัวลู่ก RTU จำนวน 2 ชุด คือหมายเลข 1 และหมายเลข 2



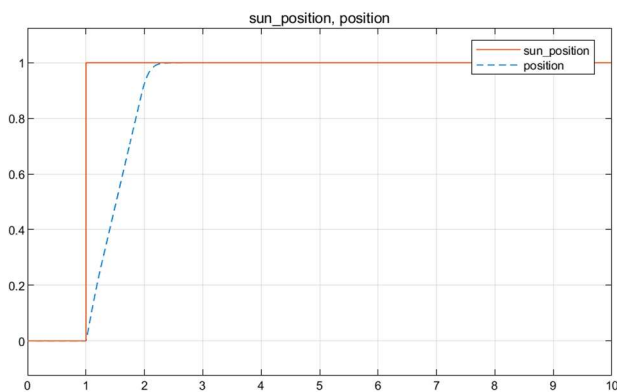
(ก)



(ข)

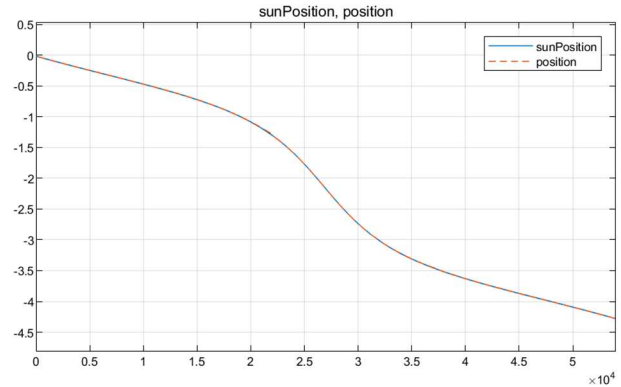
รูปที่ 11 ต้นแบบตัวควบคุมมอเตอร์ตัวลูก RTU หมายเลข 1
 (ก) รูปภายนอกจะเห็นเฉพาะมอเตอร์ในแนวแกน X ส่วนมอเตอร์ในแนวแกน Y จะอยู่ในกล่อง (ข) รูปภายในจะเห็นบอร์ดควบคุมและบอร์ดขับมอเตอร์

ผลการทดลองของตัวควบคุมพีซีลلاجิกเมื่อมีการส่งค่าเซตพีซีให้ตัวควบคุมเพื่อให้ติดตามตำแหน่งของดวงอาทิตย์ในหนึ่งแกนแสดงดังรูปที่ 12 เมื่อกำหนดให้มีการเปลี่ยนตำแหน่งของดวงอาทิตย์แบบสลับ ซึ่งระบบสามารถติดตามได้ถูกต้อง โดยใช้เวลาดตอบสนอง 1.2 วินาที และไม่มีค่าความผิดพลาด



รูปที่ 12 ตัวควบคุมพีซีลلاجิกสามารถติดตามตำแหน่งของดวงอาทิตย์ได้เมื่อมีการเปลี่ยนแปลงตำแหน่ง

รูปที่ 13 แสดงผลการทดลองของตัวควบคุมพีซีลلاجิกเมื่อมีการติดตามตำแหน่งของดวงอาทิตย์ในหนึ่งแกนแบบต่อเนื่อง ผลการทดลองพบว่าสามารถติดตามตำแหน่งของดวงอาทิตย์ได้อย่างถูกต้องตามการเคลื่อนที่ที่กำหนด



รูปที่ 13 ตัวควบคุมพีซีลلاجิกสามารถติดตามตำแหน่งของดวงอาทิตย์ได้อย่างต่อเนื่อง

5. สรุปผลและข้อเสนอแนะ

ผลการวิจัยแสดงให้เห็นว่าสามารถพัฒนาการใช้งานมอเตอร์ RTU และโปรโตคอลมอเตอร์ TCP ขึ้นใช้ได้เอง โดยสามารถปรับโครงสร้างของตัวควบคุมได้ตามต้องการไม่ใช่ปรับได้แค่เพียงค่าเซตพอยต์ ทำให้มีความยืดหยุ่นในการนำไปใช้งาน สามารถปรับตัวควบคุมให้เหมาะสมกับกระบวนการในอุตสาหกรรมได้หลากหลายเพราะการปรับโครงสร้างจะทำให้ตัวควบคุมมีความเร็วในการตอบสนองที่แตกต่างกัน เช่นการควบคุมระบบมอเตอร์ก็ต้องใช้โครงสร้างที่ตอบสนองเร็ว เป็นต้น บทความนี้ได้พัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์ ESP32 และพัฒนาโปรแกรมตัวควบคุมแบบ Fuzzy บนไมโครคอนโทรลเลอร์ Arduino MEGA 2560 ซึ่งจะเป็นประโยชน์สำหรับภาคอุตสาหกรรมในการเปลี่ยนถ่ายไปสู่อนาคต

ต้นแบบงานวิจัยยังเป็นการประกอบวงจรและอุปกรณ์เพื่อการทดสอบเบื้องต้นเท่านั้น การจะนำไปใช้ได้จริงยังต้องการการออกแบบด้านวงจรที่ทนต่อสภาพแวดล้อมทางอุตสาหกรรม เช่น ทนต่อระบบไฟฟ้าในโรงงาน อุณหภูมิ ความชื้น เป็นต้น

6. เอกสารอ้างอิง

- [1] Contemporary Control Systems, Inc. "Introduction to Modbus Serial and Modbus TCP", the Extension a technical supplement to control network, Vol. 9, Issue 5, Sept-Oct, 2008.
- [2] Sadik Tamboli, Mallikarjun Rawale, Rupesh Thoraiet and Sudhir Agashe, "Implementation of Modbus RTU and Modbus TCP communication using Siemens S7-1200 PLC for batch process" International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp. 258 – 263, 2015.
- [3] Qingbao Huang, Qianzhong She and Xiaofeng Lin, "Adaptive fuzzy PID temperature control system based on OPC and modbus/TCP protocol," 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), Wuhan, pp. 238-241, 2010.
- [4] D. Gao, "Design and Implementation of Intelligent Control System for Green Building," 2019 11th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Qiqihar, China, pp. 28-31, 2019.
- [5] M. A. Ali, A. H. Miry and T. M. Salman, "IoT Based Water Tank Level Control System Using PLC," 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, pp. 7-12, 2020.
- [6] P. V. Vyatkin, "Software Engineering in Industrial Automation: State-of-the-Art Review," IEEE Trans. Ind. Informatics, Vol. 9, no. 3, Aug. 2013, pp. 1234-1249.
- [7] T. Somporn, M. Prachayanont and P. Krissana, "Wireless Solar Tracking for Solar Cells by using Fuzzy Logic," in Proceeding of The 13th Conference on Energy Network of Thailand, Thailand, May 31 - June 2, 2017, pp. 484-487.
- [8] Espressif Systems (Shanghai) CO., LTD. ESP32 Hardware Reference. [Online]. (2019). [Cited October 10, 2019]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/hwreference/index.html>