การประยุกต์ใช้งานสมองกลฝังตัวสำหรับการออกแบบ PLC บนเครือข่ายอีเธอร์เน็ต

Applying embedded system to design programmable logic controller based on Ethernet network

สามารถ เลิศเสรี สุพรรณ กุลพาณิชย์

สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

บทความนี้กล่าวถึงการออกแบบและพัฒนา คอนโทรลเลอร์แบบสมองกลฝังตัว(Embedded Controller) ให้เป็นเครื่องควบคุม PLC ที่สามารถเชื่อมต่อกับโฮสคอมพิวเตอร์ผ่านระบบโครงข่ายแบบอีเธอร์เน็ต (Ethernet) การออกแบบแบ่งเป็นส่วนของฮาร์ดแวร์ และ ซอฟท์แวร์ โดยส่วนของฮาร์ดแวร์ได้เลือกเอาสมองกลฝังตัวตระกูล ARM7TDMI-S ที่ประมวลผลข้อมูลขนาด 32 บิต เบอร์ LPC2378 เป็นตัวควบคุมการทำงานหลักซึ่งประกอบด้วยหน่วยอินพุตเอาต์พุตแบบดิจิตอล ระบบสื่อสารข้อมูลที่เป็นทั้งแบบอนุกรมและขนาน หน่วยความจำภายในและภายนอก ในส่วนของซอฟท์แวร์ ได้ใช้ระบบปฏิบัติการเวลาจริง (Real time OS) ที่มีอยู่ในระบบสมองกลฝังตัวให้เป็นระบบปฏิบัติการของเครื่องควบคุม โปรแกรมตัวแปลงภาษา (Interpreter) เป็นส่วนหนึ่งของระบบปฏิบัติการที่แปลงจากไฟล์ตัวอักษร (Text File) ให้เป็นรหัสการทำงาน (Operating Code) ในรูปของไฟล์แบบไบนารี (Binary File) การพัฒนาระบบปฏิบัติการนี้อาศัยโปรแกรมตัวแปรภาษา C ตามมาตรฐาน ANSI (ANSI Standard C) การทดสอบประสิทธิภาพการทำงานของเครื่องควบคุมทำได้โดยการวัดเวลาในการสแกน (Scan Time) เมื่อมีการ Online และ Offline พอร์ทอีเธอร์เน็ต

คำหลัก: สมองกลฝังตัว, อีเธอร์เน็ต, ระบบปฏิบัติการเวลาจริง, โปรแกรมตัวแปลงภาษา, เวลาในการสแกน

Abstract

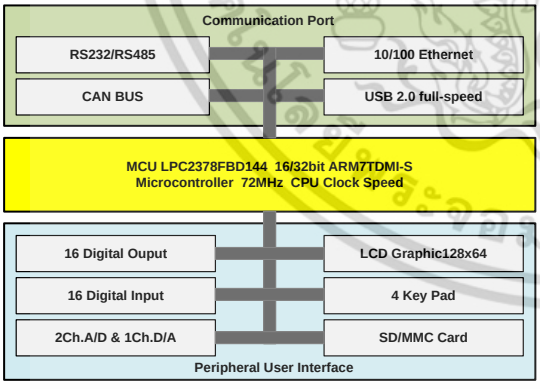
This paper proposes the designation and development of embedded controller to programmable logic controller (PLC) linked to host computer via Ethernet network. The designation consisted of Hardware and Software parts. Hardware part, the 32 bit microcontroller (LPC2378) that is a family of ARM7TDMI-S embedded system has been applied to be the main controller comprising of digital input and output, both serial and parallel communication, internal and external memories. Software part uses the real-time Operating System (OS) based on embedded system, developed to be the OS of controller. The interpreter is a part of OS, generates code from text file to a binary file. The development uses C Complier program according to ANSI Standard. Efficiency of the controller measured by scanned time when on-line and off-line Ethernet port has been illustrated

Keywords: Embedded controller, Ethernet, Real time OS, Interpreter, Scan time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.บทนำ

ระบบสมองกลฝังตัว(Embedded system) จะมีตัวควบคุมไมโครโปรเซสเซอร์แบบชิพเดียว(LSI : Large Scale Integrated Circuit) ที่เป็นส่วนประกอบสำคัญทางด้านฮาร์ดแวร์ การออกแบบ เริ่มตั้งแต่ การเลือกใช้หน่วยประมวลผลกลาง (Micro Processor Unit หรือ MPU) เป็นหน่วยประมวลผลที่มีความสามารถในการคำนวณและประมวลผลได้ครั้งละ 32 บิต ทำงานที่ความถี่สูงถึง 72 MHz ส่งผลให้เวลาในการปฏิบัติคำสั่ง(Execution Time) สามารถตอบสนองได้รวดเร็วและมีประสิทธิภาพมาก ลำดับต่อมา หน่วยเชื่อมต่ออุปกรณ์ภายนอก ถูกออกแบบให้รองรับส่วนที่เป็นข้อมูลทางเข้าโดยตรงไม่ว่าจะเป็นข้อมูลอินพุตเอาต์พุตแบบดิจิตอลหรืออนาล็อกที่เชื่อมต่อกับตัวตรวจจับ(Sensor)และตัวแปลงสัญญาณ (Transducer) หรือ จะเป็นอุปกรณ์ที่ใช้อำนวยความสะดวก เช่นจอภาพ LCD คีย์แพ็ต ป้อนข้อมูล หรือ หน่วยความจำ SD/MMC Card ทั้งหมดนี้เป็นหน่วยเชื่อมต่ออุปกรณ์ภายนอก หน่วยสื่อสารข้อมูลเป็นอีกหน่วยหนึ่งที่ออกแบบไว้ให้สามารถสื่อสารข้อมูลได้หลายช่องทางเช่น UART ,Can bus, Usb และ Ethernet port การออกแบบ PLC มีโครงสร้างตามรูปที่ 1

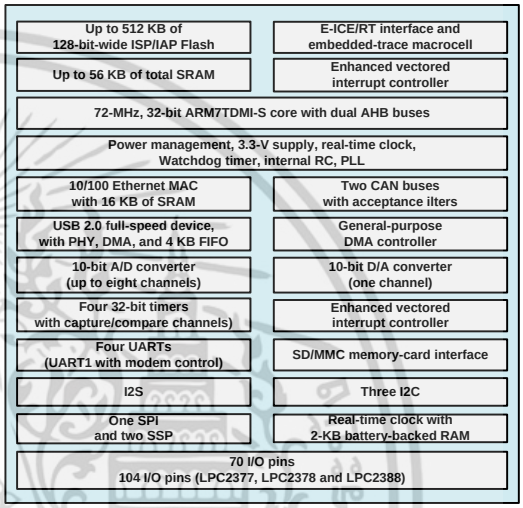


รูปที่ 1 บล๊อคไดอะแกมโครงสร้างของ PLC

สิ่งต่างๆเหล่านี้มีอยู่ในระบบสมองกลฝังตัว ทำให้ความสามารถในการทำงานไม่ต่างไปจากเครื่องคอมพิวเตอร์ แต่มีขนาดเล็กกว่าหลายเท่า อีกสิ่งที่อยากจะกล่าวถึงคือความสามารถ และสถาปัตยกรรมทางด้านซอฟท์แวร์ ที่ทำงานเป็นแบบมัลติแทสค์ (Multi Tasking) สามารถบริการงานได้หลายงานพร้อมๆกัน ซึ่งเป็นระบบปฏิบัติการเวลาจริง (Real time Operating System) ดังนั้นระบบสมองกลฝังตัวจึงได้ถูกนำมาประยุกต์ใช้เป็นเครื่องควบคุมอัตโนมัติให้กับอุปกรณ์ไฟฟ้าต่างๆมากมายหลายชนิด สำหรับบทความนี้ก็เช่นกันได้นำระบบสมองกลฝังตัวมาประยุกต์ใช้เป็นเครื่องควบคุม PLC เพื่อควบคุมเครื่องจักรกลในงานอุตสาหกรรม

2 . โครงสร้างของ PLC

2.1หน่วยประมวลผลกลาง



รูปที่ 2[1] โครงสร้าง MCU LPC2378

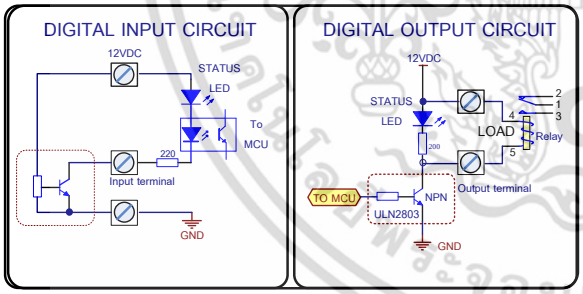
หน่วยประมวลผลกลาง (Central processing unit) หรือที่นิยมเรียกย่อ ๆ ว่า ซีพียู (CPU) เป็นหน่วยที่ทำหน้าที่เปรียบเสมือนสมองของระบบ โดยมีระบบปฏิบัติการเริ่มต้นตั้งแต่ โปรแกรมอ่านข้อมูลจาก อินพุททำหน้าที่รับข้อมูลเข้า จากนั้นนำข้อมูลที่ได้ ไปทำการ ประมวลผลคำนวณ ตามโปรแกรมประยุกต์ของผู้ใช้งานหรือโปรแกรม PLC แล้วจึงส่งข้อมูลผลลัพธ์ที่ได้ไปที่เอาท์พุท การทำงานจะวนรอบลักษณะเช่นนี้ต่อเนื่องตลอดเวลา และเวลาที่ใช้ในหนึ่งรอบการทำงานจะเรียกว่า เวลาในการสแกน (Scan Time)[2] ซึ่งจะเป็นตัวชี้วัดขีดความสามารถหรือประสิทธิภาพของ เครื่องควบคุม PLC ว่าสามารถตอบสนองต่อการเปลี่ยนแปลงสถานะของอินพุทเอาท์พุทได้เร็วเพียงใด จากข้อกำหนดต่างๆเหล่านี้ จึงได้ออกแบบเครื่องควบคุม PLC ให้ใช้ ไมโครคอนโทรลเลอร์ 32 บิต ในตระกูล ARM7TDMI-S เบอร์ LPC2378 ของ NXP มีโครง สร้างภายในตามรูปที่ 2 เป็นซีพียู ขนาด 32 บิต 144 ขา ทำงานที่ความถี่สัญญาณนาฬิกา 72 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมีหน่วยความจำแบบแฟลชความจุ 512 กิโลไบท์ เก็บ โปรแกรมส่วนระบบปฏิบัติการ หน่วยความจำแบบสแตติกแรม 58 กิโลไบท์ใช้จัดเก็บหน่วยความจำต่างๆ PLC นอกจากนี้ยัง มีโมดูล USB, RTC, Ethernet, SD/MMC memory card, CAN controller, SRAM powered backup 2 กิโลไบท์ จากสถาปัตยกรรมที่มีอยู่อย่างครบถ้วน จึงถูกเลือกให้เป็นซีพียู ของงานวิจัยนี้ การควบคุมสามารถกระทำได้ทั้งการทำงานเป็นแบบอิสระ (Stand alone) หรือการทำงานเป็นแบบเครือข่าย (Network) ได้อย่างสอดคล้องและลงตัว

2.2 หน่วยอินพุต/เอาต์พุต

หน่วยอินพุตเอาต์พุตดิจิตอลมีขนาด 16 อินพุต/16 เอาต์พุต เนื่องจาก MCU มีขาอินพุต-เอาต์พุต อยู่จำนวนมากจึงสามารถต่อเข้าโดยตรงเพื่อให้เป็น อินพุตเอาต์พุตของเครื่องควบคุม PLC อุปกรณ์ด้านอินพุต จะทำหน้าที่รับสัญญาณจาก ภายนอก เช่น สวิตช์ และตัวตรวจจับต่างๆ ซึ่งอินพุตจะเป็น Sink input [2] ตามรูปที่ 3 สำหรับส่วนของชุดเชื่อมต่อสัญญาให้กับตัวควบคุมจะอาศัยออปโตคลัปเปิล (Opto-couple) เป็นสื่อเชื่อมสัญญาแบบแสงทำให้ไม่มีการเชื่อมต่อทางไฟฟ้าโดยตรง (Isolate)



รูปที่ 3 รูปแสดงวงจรอินพุต/วงจรเอาต์พุตแบบดิจิตอล

ด้านเอาต์พุตแบบดิจิตอล จะทำหน้าที่ ขับสัญญาณที่สั่งการมาจากหน่วยประมวลผลกลางออกไปยังอุปกรณ์ทางด้านเอาท์พุต ในที่นี้จะออกแบบให้เป็นทรานซิสเตอร์เอาท์พุตชนิด NPN แบบอาร์เรย์ Open Collector เบอร์ ULN2308 วงจรตามรูปที่ 3 โดยมีรีเลย์เป็นภาระ ในการขับเพื่อเปิด/ปิดวงจรไฟฟ้ากระแสตรงอีกชั้นหนึ่งและเพื่อแยกกราวด์ของระบบออกจากกัน

3. การจัดวางเม็มโมรีของ PLC

3.1 หน่วยความจำแรม(Random Access Memory)

เป็นหน่วยความจำที่สามารถ อ่าน และเขียน ซ้ำได้หลายครั้ง ภายใน MCU LPC2378 มีขนาดความจุ 58 กิโลไบท์ ซึ่งมีขนาดเพียงพอที่จะจัดให้เป็นหน่วยความจำเพื่อใช้งานของหน่วยความจำ PLC ตามตารางที่ 1 มีขนาดขนาด 8 กิโลไบท์ เพื่อกำหนดให้เป็นหน่วยความจำ จำลอง (Visual Memory) สำหรับเครื่องควบคุม PLC ดังนั้นหน่วยความจำส่วนนี้จึงเรียกว่า หน่วยความจำ PLC โดยแบ่งตามชนิดใช้งาน และการทำงาน แสดงในตารางที่ 1

Area	Size	Range(word)
Input bits	160 bit	IR000 to IR009
Output bits	160 bit	IR010 to IR019
Work relay (IR Area)	2000 bit	IR106 to IR231
Holding relay (HR Area) * (Nonvolatile RAM)	320 bit	HR 00 to HR 19
Auxiliary relay (AR Area)	256 bit	AR 00 to AR 15
Link relay (LR Area)	256 bit	LR 00 to LR 15
Temporary relay (TR Area)	8 bit	TR 0 to TR 7
Special relay (SR Area)	384 bit	SR232 to SR255
Timers	256 Points	TIM 000 to 256
Counters	256 Points	CNT 000 to 256
Data memory	800 Byte	DM000-DM399
Data memory * (Nonvolatile RAM)	224 Byte	DM400-DM512

* หมายเหตุ Nonvolatile RAM เป็นหน่วยความจำที่คงค่าข้อมูลไว้ได้แม้ไฟดับ

ตารางที่ 1 แสดงหน่วยความจำ PLC[7]

3.2 หน่วยความจำ SD การ์ด (Secure Digital Memory Card)

SD การ์ด เป็นหน่วยความจำที่สามารถ อ่าน และเขียนซ้ำใหม่ได้ตลอดเวลาที่ผู้ใช้งานต้องการจึงมีความยืดหยุ่นสูงในการที่จะนำมาจัดเก็บโปรแกรมประยุกต์ของเครื่องควบคุม PLC การออกแบบจะอาศัยบัสข้อมูลแบบ SD เป็นช่องทางสื่อสารร่วมกับหน่วยประมวลผล จัดเก็บข้อมูลในรูปแบบ FAT16 ข้อมูลที่จัดเก็บส่วนแรกได้แก่โปรแกรมประยุกต์ผู้ใช้งานที่อยู่ในรูปแบบไฟล์ตัวอักษร (Text File) ผู้ใช้งานสร้างได้จากโปรแกรมอิดิเตอร์ทั่วไป เช่น โปรแกรม Notepad ที่มีอยู่ใน MS. Windows เป็นต้น และส่วนที่สองโปรแกรมโค๊ตคำสั่ง(Mechanic Code)ที่ผ่านการแปลงความหมายมาแล้วจัดเก็บในรูปแบบไฟล์ไบนารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

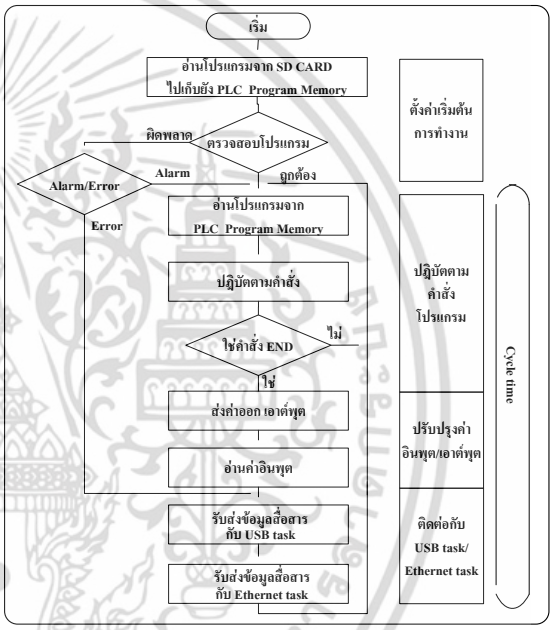
(Binary File) ที่หน่วยประมวลผลการสามารถอ่านและนำไปประมวลผลต่อได้ทันที่

4.การติดต่อสื่อสารแบบ Ethernet

ในการติดต่อสื่อสารระหว่างโฮสคอมพิวเตอร์กับเครื่องควบคุมผ่านเครือข่ายแบบ Ethernet ชุดประมวลผลกลาง LPC2378 ตามรูปที่ 2 จะมี Ethernet MAC (Media Access Controller) [3] อยู่ภายในตัว นอกจากนี้ยังมีฮาร์ดแวร์ ตัวควบคุม DMA (Direct Memory Access) เพื่อการโอนถ่ายข้อมูลจากหน่วยความจำโดยตรง ทำให้ลด ภาระการทำงานของหน่วยประมวลผลในการเชื่อมต่อ Ethernet ลงได้ มีความเร็วในการสื่อสารมีระดับ Fast Ethernet 100 เมกะบิตต่อวินาที ผ่านสายสัญญาณชนิด UPT ทางพอร์ท RJ-45 ที่ติดตั้งอยู่บนแผงวงจรหลัก ไปยังระบบ Ethernet โดยมีโปรโตคอลแบบ TCP/IP และ UTP/IP เป็นพื้นฐานในการรับส่งข้อมูล และการสื่อสารรูปแบบ Embedded Web Server ที่เป็นอีกส่วนหนึ่งของระบบปฏิบัติการ ถูกพัฒนาด้วยภาษา HTML ทำให้เครื่องควบคุม PLC สื่อสารผ่านโครงข่าย Ethernet ได้อีกช่องทางหนึ่งโดยอาศัยโปรแกรมเว็บเบราเซอร์ (Web Browser) เพียงแค่กรอกตำแหน่ง IP address ลงไปก็จะสามารถ อ่าน/เขียนหน่วยความจำ PLC, Up-Down load โปรแกรมประยุกต์และควบคุม PLC ในระยะไกลผ่านโปรแกรม เว็บเบราเซอร์ ได้ทันทีทำให้สะดวกรวดเร็ว จึงเป็นจุดเด่นของงานวิจัยชิ้นนี้ทำให้ลดความยุ่งยากในการพัฒนาโปรแกรมบนคอมพิวเตอร์เพื่อควบคุมงานประยุกต์เฉพาะอย่างของเครื่องควบคุม PLC

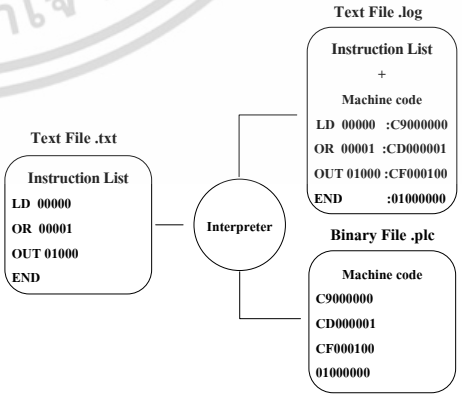
5.โปรแกรมการทำงานของ PLC

โปรแกรมระบบปฏิบัติการจะทำงานเป็นแบบมัลติแทสค์ (Multi Tasking) ที่บริการได้หลายงานพร้อมๆกัน ระบบปฏิบัติการนี้มีจะมีการตอบสนองของ หน่วยประมวลผลกลางที่มีต่อโปรแกรมผู้ใช้งานที่มีผลตอบสนองต่ออินพุทเอาท์พุท ต้องทำงานไปพร้อมกันกับการรับส่งข้อมูลผ่านทางพอร์ท Ethernet พอร์ท USB และ อ่านเขียนข้อมูลกับหน่วยความจำ SD และอื่นๆทั้งหมด เราเรียกระบบปฏิบัติการนี้ว่า ระบบปฏิบัติการเวลาจริง (Real time Operating System :RTOS) สำหรับ แทสค์(task) โปรแกรม PLC ตามแผนภูมิการทำงานดังรูปที่ 4 จะเริ่มจากโปรแกรมจะทำการอ่านไฟล์ข้อมูลแบบไบนารีไฟล์จากหน่วยความจำ SD Card มาเก็บยังหน่วยความจำโปรแกรมของ PLC จากนั้นจะอ่านโปรแกรมคำสั่งและประมวลผลตั้งแต่คำสั่งแรกจนถึงคำสั่งสุดท้ายแล้ว จึงส่งข้อมูลผลลัพธ์ที่ได้จากการประมวลผลไปยังหน่วยความจำเอาท์พุท และ สู่เอาต์พุตภายนอก จากนั้นอ่านข้อมูลอินพุตจากภายนอกเข้ามาสู่หน่วยความจำอินพุท พร้อมกับรับส่งข้อมูลให้กับหน่วยสื่อสารข้อมูล Ethernet แทสค์(task) และ USB แทสค์(task) เป็นการทำงานครบหนึ่งรอบ และจะเริ่มต้นรอบใหม่อีกครั้งในทำลักษณะเช่นนี้เรื่อยไปจนกว่าจะหยุดการทำงาน[4]



รูปที่ 4 แผนภูมิการทำงานของแทสค์โปรแกรม PLC

5.1 อินเทอร์พรีเตอร์ (Interpreter)



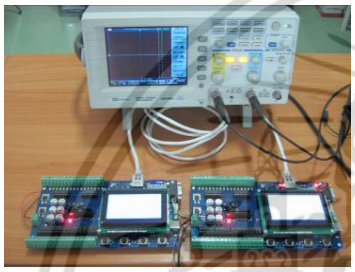
รูปที่ 5 แสงดงการทำงานของอินเทอร์พรีเตอร์

โปรแกรมอินเทอร์พรีเตอร์ (Interpreter) เป็นส่วนสำคัญมีหน้าที่ทำการแปลงคำสั่งโปรแกรม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปแบบ คำสั่ง Instruction List : IL ตามมาตรฐาน IEC 1131-3[5] โปรแกรมจะจัดเก็บข้อมูลในรูปแบบไฟล์ตัวอักษร ในหน่วยความจำ SD Card แล้วทำการแปลงโปรแกรมทั้งหมดให้ได้ผลลัพธ์จัดเก็บได้ออกเป็น 2 ไฟล์ ตามรูปที่ 5 ได้แก่ไฟล์ข้อมูลแบบไบนารี หรือข้อมูลรหัสเครื่อง (Machine Code) ที่ PLC สามารถเข้าใจแล้วนำไปใช้งานได้ทันที ส่วนไฟล์ข้อมูลแบบตัวอักษรอีกไฟล์หนึ่ง (Textfile.log) เพื่อใช้รายงานความผิดพลาดของโปรแกรมที่ผ่านการทำงานของอินเทอร์พรีเตอร์มาแล้ว

6.การทดลอง



รูปที่ 6 อุปกรณ์และการทดสอบประสิทธิภาพของ PLC

ในงานวิจัยนี้สามารถทดสอบประสิทธิภาพการทำงานได้ 2 การทดลองโดยแบ่งการทดลองที่ 6.1ใช้เครื่องควบคุม PLC เพียงชุดเดียว ส่วนการทดลองที่ 6.2 จะใช้เครื่องควบคุมจำนวน 2 ชุด

6.1 การทดสอบประสิทธิภาพของแสกนไทม์ตามจำนวนคำสั่ง

การทดสอบประสิทธิภาพการทำงานของ PLC โปรแกรมที่ใช้ทดสอบเป็นโปรแกรมผู้ใช้งานที่มีความจุคำสั่งผู้ใช้งานได้มากถึง 16 Kstep โดยการจำลองคำสั่งให้เพิ่มจากน้อยไปหามากเพิ่มครั้งละ 0.5 Kstep มีสัดส่วนคำสั่งระดับบิต 70% และคำสั่งระดับเวิลด์ 30% สัดส่วนนี้ได้มีการวิจัยมาแล้วว่าเหมาะสมกับการทดสอบ[6] ผลการทดสอบ เริ่มต้นที่คำสั่ง 10 step, 0.5 , 1, 1.5, 2 ,….16 Kstep แล้วบันทึกเวลาการสแกนของ PLC โดยเวลาการสแกนจะอ้างอิงจาก ฐานเวลา 1 µsec ที่สร้างไว้ภายในหน่วยความจำตัวตั้งเวลาเพื่อใช้ตรวจจับเวลาในแต่ละคาบที่ซีพียูทำงานครบ 1 รอบว่านานเท่าใด จากนั้นค่อยนำผลมาแสดงยังหน่วย ความจำ DM0380 ของ PLC

จากตารางที่ 2 แสดงบางส่วนของคำสั่ง IL ระดับบิทและเวิลด์ จะเห็นว่าเวลาการปฏิบัติคำสั่งระดับบิทจะมากกว่าเป็นเพราะการพัฒนาโปรแกรมบน ARM7TDMI-S ไม่ค่อยสนับสนุนคำสั่งในระดับบิทจำเป็นต้องเอาคำสั่งเวิลด์มาประยุกต์ใช้แทน เวลาการปฏิบัติคำสั่งมีหน่วยเป็น µsec

บางส่วนคำสั่งระดับบิท

คำสั่ง IL	เวลาปฏิบัติ(µsec)
LD	4
LD NOT	4
AND	2
OR	2
OUT	3

บางส่วนคำสั่งระดับเวิลด์

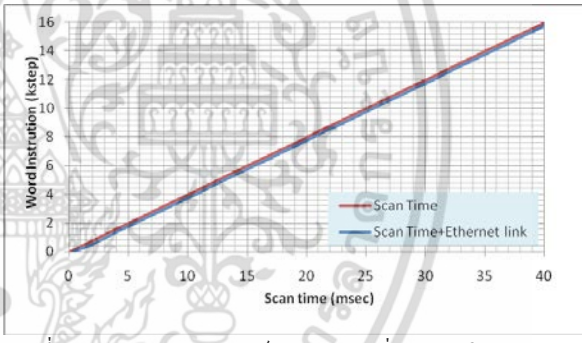
คำสั่ง IL	เวลาปฏิบัติ(µsec)
CMP S,D	2
MOV S,D	2
ADD S,A,R	1
SUB S,S,R	1
MUL S,M,R	1

ตารางที่ 2

บางส่วนจำนวนคำสั่งกับเวลาการสแกนที่ Online และ Offline พอร์ท Ethernet

จำนวนสเต็ป(Kstep)	Online (msec)	Offline(msec)
0.01	0.389	0.116
0.5	1.578	1.341
4	10.328	10.091
8	20.328	20.091
12	30.328	30.091
16	40.328	40.091

ตารางที่ 3



รูปที่ 7 กราฟแสดงความสัมพันธ์ของจำนวนคำสั่งและเวลาในการสแกน

จากตารางที่ 3 เป็นผลการทดลองจำนวนโปรแกรมที่ 0.5 Kstep ใช้เวลาในการสแกน 1.578 msec และ1.341 msec และจำนวนโปรแกรมที่16 Kstepใช้เวลาในการสแกน 40.328 msec และ 40.091 msec เมื่อทำการOnline และ Offline พอร์ท Ethernet ตามลำดับ ในการทดลองวัดค่าเวลาในการสแกน ขณะมีการติดต่อรับส่งข้อมูลทาง Ethernet นั้นจะทำให้เวลาเพิ่มขึ้นจากเดิมอีก 273 µSec ซึ่งนับว่าน้อยมากเมื่อเทียบกับโปรแกรมการทำงานทั้งหมดของ PLC

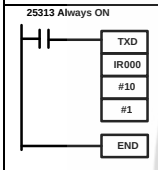
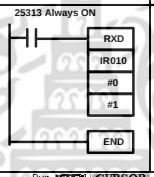
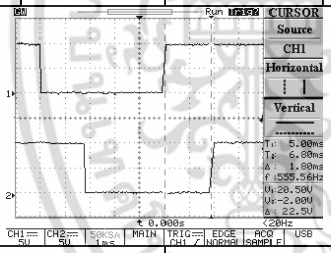
6.2 การทดสอบเวลาการตอบสนองต่ออินพุต/เอาต์พุต ของ PLC ผ่านพอร์ท Ethernet (I/O Response Time)

ทดสอบโดยใช้ PLC จำนวน 2 ชุดต่อผ่านพอร์ท Ethernet ตามรูปที่ 6 และเขียนโปรแกรมด้วยคำสั่ง IL ตามตารางที่ 4 โดยให้ชุดที่ 1 โหนดหมายเลข #10 เป็นชุดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอินพุตจากสวิทช์ลงหน่วยความจำอินพุทแล้วเขียนโปรแกรมส่งข้อมูลด้วยคำสั่ง TXD ส่งข้อมูลผ่าน พอร์ท Ethernet ในทางกลับกันเขียนโปรแกรมรับข้อมูลด้วยคำสั่ง RXD ให้กับ PLC ชุดที่ 2 โหนดหมายเลข #00 เพื่อรับข้อมูลจากพอร์ท Ethernet ส่งมายังหน่วยความจำเอาท์พุทและส่งข้อมูลออกไปอุปกรณ์ภายนอก ทางฝั่งอินพุตของ PLC ชุดที่ 1 ป้อนสัญญาณพัลส์ความถี่ 100Hz หรือขนาด 10 msec

เป็นความถี่ที่เครื่องควบคุมตอบสนองได้ทัน แล้วใช้ออสซิลโลสโคป(Oscilloscope) ทำการเปรียบเทียบระหว่างสัญญานอินพุตของ PLC ชุดที่ 1 ด้วย OSC. Ch1 และเอาต์พุตของ PLC ชุดที่ 2 ด้วย OSC. Ch2 ตามลำดับ [7]

PLCชุดที่ 1 Unit #10 โปรแกรมส่งข้อมูลอินพุตผ่านพอร์ทEthernet ขนาด 1 word		PLC ชุดที่ 2 Unit #00 โปรแกรมรับข้อมูลเอาต์พุตผ่านพอร์ทEthernet ขนาด 1 word	
โปรแกรม LD	โปรแกรม IL	โปรแกรม LD	โปรแกรม IL
	LD 25313 TXD -- (A) 000 (B) #10 (C) #01 END		LD 25313 RXD -- (A) 010 (B) #00 (C) #01 END
			
Ch1 Input PLC1		Ch2 Output PLC2 Delay time	
10 msec.		1.80 msec.	

ตารางที่ 4 ตารางโปรแกรม PLC ทดสอบที่ 6.2

จากผลการทดลองตารางที่ 4 เมื่อเปรียบเทียบอินพุตจาก PLC ชุดที่1 และเอาต์พุตจาก PLC ชุดที่2 สัญญาณเอาต์พุตจะล้าหลังตามหลังสัญญาณอินพุตอยู่ 1.80 msec.

7. สรุป

ในบทความนี้ได้นำเสนอการใช้เทคโนโลยีด้านสมองกลฝังตัว โดยใช้ ไมโครคอนโทรลเลอร์ 32 บิต ในตระกูล ARM7TDMI-S เบอร์ LPC2378 ซึ่งมีโครงสร้างภายในเอื้อต่อการนำมาออกแบบเป็นเครื่องควบคุม PLC ได้อย่างลงตัว และเนื่องจาก PLC สามารถทำงานแปลงคำสั่ง IL ในรูปแบบไฟล์ตัวอักษร ให้ทำงานได้ทันทีจึงลดความยุ่งยากในการใช้เครื่องป้อนโปรแกรม (Programming console) และสะดวกในการใช้งานอย่างมาก ที่สำคัญยังสามารถเชื่อมต่อบนเครือข่าย Ethernet ให้ทำงานได้ทั้งโหมดการปรับปรุง แก้ไข โปรแกรม และโหมดการลิงค์สัญญาณเครื่องควบคุมเข้าด้วยกัน จากการทดลองผลที่ได้รับเป็นที่พอใจในระดับหนึ่ง และสามารถปรับปรุงให้มีประสิทธิภาพ และ มีเสถียรภาพมากขึ้นทั้งในด้านการเขียนโปรแกรมผู้ใช้งาน การสื่อสารข้อมูล และการควบคุมการทำงาน นอกจากนี้ชุดคำสั่งของ PLC ที่มีอยู่ขณะนี้มีเพียง 49 คำสั่งสามารถใช้งานได้ในระดับหนึ่งเช่นกัน อาจจะไม่ครอบคลุมการประยุกต์ใช้งานในทุกรูปแบบ ดังนั้นสามารถจะพัฒนาต่อให้มีชุดคำสั่งที่หลากหลายมากขึ้นไป อีกทั้ง ทางด้านการสื่อสารข้อมูลซอฟท์แวร์ สามารถเพิ่มเติมโปรโตคอลคำสั่ง ที่ใช้ในการสื่อสารข้อมูลทั้งในส่วนพอร์ทอนุกรม และ ส่วนของ Can bus จะทำให้เครื่องควบคุมในงานวิจัยนี้มีความสมบูรณ์มากยิ่งขึ้น

8. เอกสารอ้างอิง

[1] www.nxp.com, UM10211, LPC2378 User manual,

[2] Warnock, G Programmable Controllers: Operation and Application,Prentice-Hall,Englewood Cliffs NJ,pp35,1998

[3] Philip J, Koopman, Jr, Beyond the Embedded Web Server ,pp 143, 1996

[4] Kim,Jomg-il,Park,J and Kwon,W H'Architecture of a ladder solving processor for Programmable Controllers, Microprocessors Microsystems. Vol. 16,pp 369-379,1992

[5] Programmable Controllers-Part 3: Programming Languages ,IEC 1131-3,pp 15,1993

[6] Gab Seon Rho ,Kyeong-hoon Koo and Naehyuck Chang, Implementation of a RISC microprocessor for programmable logic controllers, Microprocessors Microsystems. Vol. 19 , pp 599-671,1992

[7] Suphan Gulpanich, Ajin Numsomran,Prapas Roengruen,Viriya Kongratana and Kitti Tirasesth, Desing of programmable logic controllers and I/O Expansions, ICCAS2005 June 2-5,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KINTEX,Gyeonggi-Do,Korea, pp1107-1111,2005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้