บทที่ 3

การออกแบบระบบเชิงเวลาจริงที่ตระหนักถึงกำลังงานที่ใช้

หลังจากได้ทำการศึกษาค้นคว้าข้อมูลที่เกี่ยวข้องกับระบบเชิงเวลาจริง เทคนิคในการลดการใช้กำลังงานโพรโตคอล Modbus ระบบปฏิบัติการเชิงเวลาจริง uC/OS-II และ ARM-7 embedded controller board ทำให้เข้าใจถึงคุณสมบัติที่สำคัญของระบบรวมถึงเทคนิคสำหรับลดการใช้กำลังงาน จึงได้นำข้อมูลดังกล่าวมาเป็นแนวทางในการออกแบบระบบ โดยมีวัตถุประสงค์สำคัญในการออกแบบคือการปรับระบบปฏิบัติการเชิงเวลาจริง uC/OS-II ให้สามารถทำงานบน ARM-7 embedded controller board ใช้กำลังงานต่ำที่สุด และได้คำตอบที่ถูกต้อง ซึ่งแนวคิดและกระบวนการสำคัญสำหรับการออกแบบเพื่อทดสอบระบบที่จะกล่าวถึงในบทนี้คือ ฮาร์ดแวร์ ซอฟต์แวร์ ความถูกต้องในการทำงานของระบบ และการเลือกใช้เครื่องมือและอุปกรณ์ ส่วนรายละเอียดในการทดสอบจะกล่าวถึงในบทต่อไป

3.1 การเลือกใช้เครื่องมือและอุปกรณ์

ระบบเชิงเวลาจริงที่ออกแบบประกอบด้วยส่วนประกอบหลัก 2 ส่วน คือ ฮาร์ดแวร์ และซอฟต์แวร์ ขั้นแรกพิจารณาซอฟต์แวร์ทั้งหมดที่ต้องโปรแกรมลงบอร์ดเพื่อประเมินขนาดโปรแกรมและข้อมูลที่ใช้ซึ่งนำมาใช้ในการพิจารณาฮาร์ดแวร์หรือบอร์ดให้มีขนาดหน่วยความจำเพียงพอต่อความต้องการของระบบ และสุดท้ายคือการเลือกคอมไพเลอร์ให้เหมาะสมกับบอร์ดที่ใช้

3.1.1 ซอฟต์แวร์

การทำงานเชิงเวลาจริงต้องการระบบปฏิบัติการเชิงเวลาจริงในการทำงานเพื่อให้ได้งานถูกต้องและความรวดเร็วในการพิจารณา นอกจากนี้เวลาที่ใช้ในการทำงานเป็นปัจจัยหนึ่งที่สำคัญอย่างมากในการพิจารณาความถูกต้องของคำตอบในระบบเชิงเวลาจริง นอกเหนือจากระบบปฏิบัติการเชิงเวลาจริงแล้วโพรโตคอลก็เป็นส่วนที่สำคัญสำหรับ embedded system ดังนั้น

ซอฟต์แวร์หลักที่ใช้เชื่อมต่ออุปกรณ์เข้าด้วยกันผ่านระบบเครือข่ายในงานวิจัยนี้คือระบบปฏิบัติการเชิงเวลาจริงและโพรโตคอล

3.1.1.1 ระบบปฏิบัติการเชิงเวลาจริง

ระบบปฏิบัติการเชิงเวลาจริงที่ใช้ในปัจจุบันมีหลายตัวแต่ตัวที่เป็นที่รู้จักและใช้มากในงานวิจัยได้แก่ Echidna, NOS และ uC/OS-II คุณสมบัติแสดงดังตาราง 3-1 [40]

ตาราง 3-1 คุณสมบัติของระบบปฏิบัติการเชิงเวลาจริง

uC/OS-II	Echidna	NOS
Preemptive multitasking	Cooperative multitasking RTOS based on Chimera [41]	Nonpreemptive RTOS
ROMable and scalable	Supports reconfigurable component-based software for microcontrollers and digital signal processors	Bare-bones
Deterministic	Dynamic-priority cooperative RTOSs	Just a task scheduler and not a full OS
footprints small enough for microcontroller systems	Footprints small enough for microcontroller systems (Echidna has a footprint of ~6KB)	Multirate executive based on descriptions of “roll-your-own” RTOSs given by embedded-systems designers in industry
Small size (1,700 lines of code)	NA	NA

ตาราง 3-1 (ต่อ)

uC/OS-II	Echidna	NOS
Offers such services as mailboxes, queues, semaphores, time-related functions, etc	NA	NA
DO-178B Validation Suite™ for MicroC/OS-II [42]	NA	NA

ระบบปฏิบัติการเชิงเวลาจริง NOS เป็นระบบปฏิบัติการที่ยังไม่สมบูรณ์และ Echidna เป็นระบบปฏิบัติการที่ต้องทำงานร่วมกับระบบปฏิบัติการ Chimera ดังนั้นจึงไม่ได้เลือกใช้ในงานวิจัยนี้ แต่เลือกระบบปฏิบัติการ uC/OS-II เพราะมีคุณสมบัติที่เหมาะสมที่สุดอีกทั้งได้รับการตรวจสอบรับรองมาตรฐานแล้วและนิยมใช้กันมากทั้งงานทางด้านวิจัยและอุตสาหกรรม นอกจากนี้ยังมีคู่มือสำหรับศึกษาเพิ่มเติม รายละเอียดและคุณสมบัติของระบบปฏิบัติการ uC/OS-II ได้กล่าวไว้แล้วในหัวข้อ 2.3 ระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

3.1.1.2 โพรโตคอล

การติดต่อสื่อสารกับระบบคอมพิวเตอร์ฝังตัวส่วนมากใช้ในระบบอุตสาหกรรมที่สามารถติดต่ออุปกรณ์หลาย ๆ ตัวจากหลายแหล่งได้ด้วยเครื่องเพียงเครื่องเดียว โพรโตคอลสำหรับติดต่อสื่อสารบนระบบเชิงเวลาจริงจึงพิจารณาถึงโพรโตคอลที่ใช้ในอุตสาหกรรม ได้แก่ DNP, Modbus, MMS/UCA 2.0 และ IEC 60870-5-101 โดยโพรโตคอลแต่ละตัวมีความแตกต่างกันแสดงดังตาราง 3-2 [43]

จากตาราง 3-2 DNP สนับสนุนการทำงานได้หลากหลายแต่จากการศึกษาคุณสมบัติของ DNP เมื่อเปรียบเทียบกับการใช้ในงานวิจัยพบว่ามีคุณสมบัติมากเกินความจำเป็น นอกจากนี้ความซับซ้อนและขนาดของโพรโตคอลจะแปรผันตามคุณสมบัติที่โพรโตคอลนั้น ๆ สนับสนุน Modbus เป็นโพรโตคอลที่มีความซับซ้อนน้อยที่สุด ขนาดเล็กที่สุด และถูกนำมาใช้งานจริงมากที่สุด อีกทั้งสามารถรองรับการทำงานสำหรับงานวิจัยนี้ได้ ดังนั้นโพรโตคอล

Modbus จึงถูกเลือกให้เป็นเครื่องมือหนึ่งในงานวิจัยนี้ รายละเอียดของโพรโตคอล Modbus ได้กล่าวถึงแล้วในหัวข้อ 2.4 โพรโตคอล Modbus

ตาราง 3-2 คุณสมบัติของโพรโตคอลที่แตกต่างกัน

Feature	**DNP**	**Modbus RTU**	**MMS/ UCA 2.0**	**IEC 60870-5-T101**
OSI 3-layer model	✓	✗	✓	✓
Users	> 500	1000	< 100	100
Designed for utility environment	✓	✗	✓	✓
User group & technical committee	✓	✗	✗	✗
Revision control on final documentation	✓	✗	✓	✓
Defined protocol test documentation	✓	✗	✗	✗
Independent protocol verification programs	✓	✗	✗	✗
Migration path to enhanced architectures	✓	✗	✓	✓
Time synchronization & time-stamped events	✓	✗	✓	✓
Multiple masters & peer-to-peer operations	✓	✗	✓	✗
Unsolicited slaves do not need to be polled	✓	✗	✓	✗
Segmentation of messages	✓	✗	✓	✗
Secure file transfer	✓	✗	✓	✓
Broadcast messages	✓	✗	✓	✓
User defined data objects	✓	✗	✓	✗

จากการคอมไพล์แอพลิเคชันที่รองรับโพรโตคอล Modbus บนระบบปฏิบัติการเชิงเวลาจริง uC/OS-II พบว่าขนาดโค้ดที่ได้ประมาณ 9,000 ไบต์ และขนาดข้อมูลประมาณ 3,000 ไบต์ ซึ่งเป็นขนาดของโปรแกรมและขนาดของข้อมูลที่ใช้ในการเลือก embedded controller board

3.1.2 ฮาร์ดแวร์

ในการพัฒนาและทดสอบระบบเชิงเวลาจริงฮาร์ดแวร์หลักที่เกี่ยวข้องคือ เครื่องไมโครคอมพิวเตอร์ และ embedded controller board สำหรับเครื่องไมโครคอมพิวเตอร์สามารถใช้เครื่องรุ่นใดก็ได้ เครื่องที่นำมาทดสอบคือ Intel Pentium 4 ความเร็ว 1.8 GHz หน่วยความจำ 256 MB โดยความเร็วของหน่วยประมวลผลและขนาดของหน่วยความจำไม่มีผลต่อระบบเชิงเวลาจริง ส่วน embedded controller board จำเป็นต้องระบุให้ชัดเจนเนื่องจากการเขียนโปรแกรมในระดับล่างหรือติดต่อกับบอร์ดจะมีความแตกต่างกันในบอร์ดต่างชนิดและต่างรุ่น

บริษัทผู้ผลิตไมโครคอนโทรเลอร์ที่สนับสนุนการทำงานของระบบปฏิบัติการ uC/OS-II [44] คือ ADC, Altera, Analog Devices, ARC, Atmel, Freescale, Fujitsu, Hynix, IBM, Infineon, Intel, LSIL, Microchip, MIPS, National, NEC, OKI, Openrisc, Philips, Rabbit, Renesas, Samsung, Sharp, ST Micro, Texas Instruments, Xilinx และ Zilog ประเด็นสำคัญในการพิจารณาเลือก embedded microcontroller board ที่เหมาะสมคือ สนับสนุนการทำงานของระบบปฏิบัติการ uC/OS-II มีหน่วยความจำเพียงพอสำหรับการทำงานของโปรแกรม และมีจำหน่ายทั่วไปในท้องตลาดของประเทศไทยเพราะไม่เสียเวลาในการขนส่งสินค้า กรณีที่บอร์ดมีปัญหาสามารถหาซื้อเปลี่ยนได้สะดวก และราคาไม่สูงมาก สำหรับบอร์ดที่มีจำหน่ายในประเทศไทยและรองรับการทำงานของระบบปฏิบัติการ uC/OS-II แสดงดังตาราง 3-3 [45]

ตาราง 3-3 หน่วยความจำของบอร์ดรุ่นต่าง ๆ ที่รองรับระบบปฏิบัติการ uC/OS-II

บอร์ด	หน่วยความจำ FLASH	RAM
ET-dsPIC30F2010	12 Kbyte (4 Kword)	512 Byte
ET-BASE51 V2.0 (AT89C51ED2)	64 Kbyte	2 Kbyte
ET-BASE MSP430F11	2 Kbyte	128 Byte
ET-BASE MSP430F21	4 Kbyte	256 Byte
ET-BASE MSP430 F133	8 Kbyte	256 Byte
ET-BASE MSP430 F169	60 Kbyte	2 Kbyte
ET-ARM7 LPC2119	128 Kbyte	16 Kbyte
ET-ARM7 LPC2138	512 Kbyte	32 Kbyte

จากการประเมินขนาดของโค้ดและข้อมูลที่ใช้ดังที่ได้กล่าวมาแล้วในข้างต้น เพื่อความยืดหยุ่นควรใช้หน่วยความจำแฟลชไม่ต่ำกว่า 10 Kbyte และ RAM ไม่ต่ำกว่า 4 Kbyte บอร์ดที่มีคุณสมบัติตามที่ต้องการมี 2 รุ่น คือ ET-ARM7 LPC2119 และ ET-ARM7 LPC2138 เปรียบเทียบราคาบอร์ดทั้ง 2 รุ่นเพิ่มเติมได้ว่า LPC2119 ราคา 820 บาท และ LPC2138 ราคา 950 บาท ฟังก์ชันของบอร์ดทั้ง 2 รุ่นสนับสนุนงานวิจัยนี้ ดังนั้นจึงเลือกบอร์ดที่ราคาถูกกว่าคือ ET-ARM7 LPC2119 รายละเอียดของ ARM-7 embedded controller board ได้กล่าวไว้แล้วในหัวข้อ 2.2 สำหรับการพัฒนาโปรแกรมเลือกใช้ภาษาซีที่รองรับบน ARM-7 LPC2119 การเลือกใช้ตัวแปลภาษาจะกล่าวถึงในหัวข้อถัดไป

3.1.3 ตัวแปลภาษา

การเลือกใช้ตัวแปลภาษาต้องเลือกให้ถูกต้องตรงตามหน่วยประมวลผลที่นำมาใช้ ตัวแปลภาษาที่สามารถใช้ในการพัฒนาโปรแกรมเพื่อพอร์ตลงบนหน่วยประมวลผล ARM [46] คือ ARM [47], IAR [48], iMAGEcraft [49], Keil [50], Metroworks [51], Microcross [52] และ TASKING [53] สำหรับ ARM-7 LPC2119 มีเพียง Keil และ IAR เท่านั้นที่สนับสนุนการทำงานของไมโครคอนโทรเลอร์เบอร์ LPC2119 สำหรับการพอร์ต uC/OS-II ลงบน embedded controller board ให้สามารถทำงานได้อย่างถูกต้อง บอร์ดที่ใช้ต้องสนับสนุนการทำงานของ uC/OS-II ซึ่งต้องปรับหรือแก้ไขระบบปฏิบัติการ uC/OS-II บางส่วนเพื่อให้สามารถพอร์ตและทำงานบนบอร์ดที่ต้องการได้

กลุ่มผู้ผลิตและพัฒนาระบบปฏิบัติการ uC/OS-II ได้กล่าวถึงรายละเอียดเกี่ยวกับบอร์ดที่สามารถนำมาใช้ร่วมกับระบบปฏิบัติการ uC/OS-II และตัวแปลภาษาที่เหมาะสมที่สุดในการพัฒนาโปรแกรม รายละเอียดแสดงดังตาราง 3-4 [54]

ตาราง 3-4 ตัวแปลภาษาที่ใช้สำหรับบอร์ดที่มีหน่วยประมวลผลต่าง ๆ กัน

Processor	OS version	Compiler
LPC2104		GNU ADS V1.2
LPC2106	V2.76	IAR
LPC2129	V2.76	IAR
LPC2138	V2.76	IAR
XA (Large Model)		Tasking

จากตาราง 3-4 ไม่ได้กล่าวถึง LPC2119 ที่ได้เลือกไว้สำหรับงานวิจัยนี้ เมื่อศึกษาคุณสมบัติของบอร์ดเบอร์ต่าง ๆ ที่ได้กล่าวไว้ในตาราง 3-4 LPC2129 มีสถาปัตยกรรมเดียวกันกับ LPC2119 แต่ขนาดหน่วยความจำแฟลชต่างกันเพียงอย่างเดียวเท่านั้น นั่นคือ LPC2119 มีหน่วยความจำแฟลช 128 Kbyte และ LPC2129 มีหน่วยความจำแฟลช 256 Kbyte ดังนั้นจึงสามารถใช้ตัวแปลภาษาตัวเดียวกันได้

IAR เป็นตัวแปลภาษาที่เลือกใช้สำหรับงานวิจัยนี้ ทำงานร่วมกับบอร์ด ARM-7 LPC2119 โดยใช้ระบบปฏิบัติการ uC/OS-II V2.76 เครื่องมือและอุปกรณ์ที่เลือกสามารถทำงานได้อย่างถูกต้อง ทดสอบโดยปรับระบบปฏิบัติการ uC/OS-II ให้สามารถพอร์ตและทำงานบน ARM-7 LPC2119 และได้พัฒนาโปรแกรมให้ควบคุม LED บนบอร์ดติดหรือดับได้ตามต้องการ

3.2 การปรับปรุงระบบปฏิบัติการเชิงเวลาจริง uC/OS-II ให้ตระหนักถึงกำลังงานที่ใช้

การปรับปรุงระบบปฏิบัติการ uC/OS-II ให้ตระหนักถึงกำลังงานที่ใช้พิจารณาจากคุณสมบัติและการทำงานของระบบซึ่งแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ บอร์ด ARM-7 LPC2119 และระบบปฏิบัติการ uC/OS-II บอร์ด ARM-7 LPC2119 มีคุณสมบัติที่รองรับการโปรแกรมเพื่อควบคุมการใช้กำลังงาน การปรับการทำงานของระบบปฏิบัติการ uC/OS-II ในการจัดการตารางงานและระยะเวลาของ timer tick ได้

3.2.1 คุณสมบัติการควบคุมกำลังงานของบอร์ด ARM-7

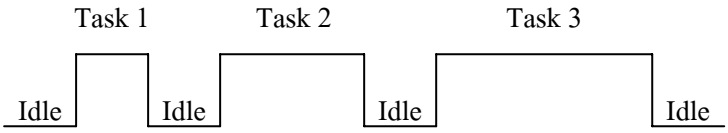
จากคุณสมบัติของ ARM-7 ดังที่ได้กล่าวไว้ในหัวข้อ 2.2 ARM-7 embedded controller board พบว่ามีคุณสมบัติในการควบคุมการใช้กำลังงานให้ต่ำลงได้ นั่นคือโปรแกรมให้ระบบอยู่ในสภาวะ idle หรือ power down ในช่วงเวลาที่เหมาะสม และนำเทคนิคดังกล่าวปรับการทำงานของระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

3.2.2 แนวทางการปรับการทำงานของระบบปฏิบัติการเชิงเวลาจริง uC/OS-II

สำหรับการปรับระบบปฏิบัติการเชิงเวลาจริงจะนำเทคนิคในการควบคุมกำลังงานของบอร์ดใช้ร่วมกับการทำงานของระบบปฏิบัติการโดยปรับให้มีการจัดตารางงานให้ดียิ่งขึ้นและปรับระยะเวลา timer tick ให้เหมาะสมกับการทำงาน

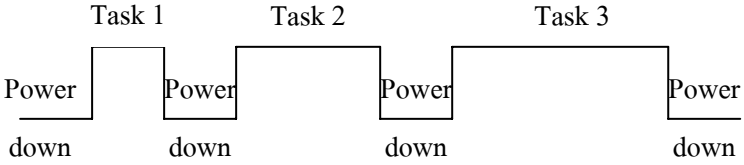
3.2.2.1 การจัดตารางงาน

uC/OS-II เป็นระบบปฏิบัติการที่ทำงานแบบ multitasking นั่นคือหน่วยประมวลผลสามารถรองรับการทำงานได้มากกว่า 1 งาน ช่วงที่มีการทำงานจะใช้กำลังงานสูงแต่เมื่อไม่มีงานระบบจะเข้าสู่สภาวะ idle สมมติให้ระบบปฏิบัติการให้บริการงาน 3 งานคือ Task1, Task2 และ Task3 โดยใช้เวลาในการทำงานในแต่ละรอบเป็น 1, 2 และ 3 ms และมีลำดับความสำคัญของงานเป็น 0, 1 และ 2 ตามลำดับ แสดงดังภาพประกอบ 3-1

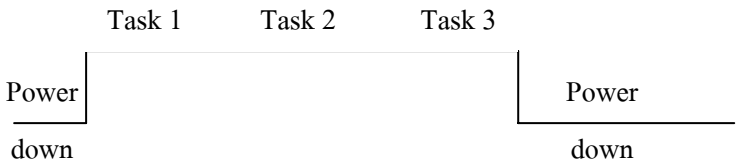


ภาพประกอบ 3-1 การทำงานแบบ multitasking

ข้อดีของสภาวะ idle คือสามารถเปลี่ยนสถานะเข้าสู่การทำงานปกติได้อย่างรวดเร็วทำให้ไม่ส่งผลกระทบต่อการทำงานเชิงเวลาจริงแต่สภาวะ idle ใช้กำลังงานสูงกว่า power down จึงปรับช่วงที่อยู่ในสถานะ idle เปลี่ยนเป็น power down ดังภาพประกอบ 3-2 แต่สภาวะ power down ใช้เวลานานในการเปลี่ยนสถานะเข้าสู่การทำงานปกติอาจส่งผลกระทบต่อการทำงานเชิงเวลาจริงได้ สำหรับการเปลี่ยนสถานะจะสูญเสียเวลาและกำลังงานส่วนหนึ่งโดยไม่มีงานเกิดขึ้นดังนั้นการเปลี่ยนสถานะบ่อยจะส่งผลกระทบต่อการทำงานเชิงเวลาจริงและกำลังงานที่ใช้จึงปรับตารางการทำงานให้ทำงานอย่างต่อเนื่องและมีช่วงเวลาสำหรับการเข้าสู่สภาวะ power down นานยิ่งขึ้น ดังภาพประกอบ 3-3



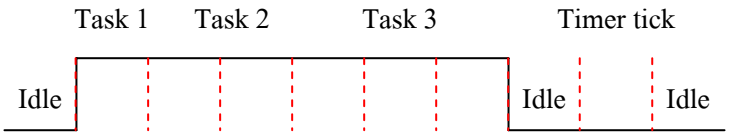
ภาพประกอบ 3-2 การทำงานแบบ multitasking ที่ปรับให้อยู่ในสภาวะ power down



ภาพประกอบ 3-3 การทำงานแบบ multitasking ที่มีการทำให้ทำงานอย่างต่อเนื่อง

3.2.2.2 การปรับ Timer tick

Timer tick เป็น interrupt ที่เกิดขึ้นในทุกรอบของช่วงเวลาหนึ่ง ช่วงเวลาดังกล่าวกำหนดโดยระบบปฏิบัติการ สำหรับการ tick ในแต่ละครั้งถ้าไม่มีงานที่ต้องการประมวลผลระบบก็จะเข้าสู่สถานะเดิมก่อนที่จะมีการ tick ดังภาพประกอบ 3-4 โดยเส้นประแสดงถึงการเกิด timer tick แต่ถ้ามีงานรอประมวลผลระบบจะเปลี่ยนสถานะในการทำงานทันทีโดยให้งานที่รอและมีลำดับความสำคัญสูงสุดได้เข้าไปทำงานก่อน ในระหว่างที่มีการทำงานยังคงเกิด interrupt เนื่องจาก timer tick ตลอดทุกช่วงเวลา แต่เมื่อระบบเข้าสู่สถานะ power down ซึ่งเป็นสถานะที่ใช้กำลังงานต่ำมากระบบจะสูญเสียกำลังงานส่วนหนึ่งเมื่อมี interrupt จาก timer tick ถ้า timer tick มีความถี่สูงจะส่งผลดีต่อการตอบสนองเชิงเวลาจริงแต่ส่งผลให้สูญเสียกำลังงานอันเนื่องจาก interrupt ของ timer tick สูงมากยิ่งขึ้น แต่ interrupt นี้ไม่เกิดในระบบที่อยู่ในสภาวะ power down ดังนั้นเมื่อระบบเข้าสู่สถานะ idle จึงสามารถลดกำลังงานของระบบด้วยการลดความถี่ในการ interrupt ของ timer tick แต่การปรับดังกล่าวอาจก่อให้เกิดความผิดพลาดต่อระบบเชิงเวลาจริงได้



ภาพประกอบ 3-4 interrupt ที่เกิดจาก timer tick ทุก ๆ 1 ms

การกำหนดความถี่ของ interrupt ที่เกิดจาก timer tick ที่เหมาะสมช่วยให้ระบบสูญเสียกำลังงานลดลงและสามารถตอบสนองต่อการทำงานเชิงเวลาจริงได้

3.3 แนวทางการทดสอบการทำงานเชิงเวลาจริง

การปรับระบบปฏิบัติการเชิงเวลาจริง uC/OS-II ให้ตระหนักถึงกำลังงานที่ใช้ เมื่อปรับระบบปฏิบัติการเพื่อควบคุมการใช้กำลังงานแล้วต้องตรวจสอบความถูกต้องในการทำงานของระบบเชิงเวลาจริงด้วย นั่นคือ ความถูกต้องของคำตอบที่ได้และเวลาที่ใช้ในการทำงานสำหรับการตอบสนอง

สำหรับการทดสอบความถูกต้องในการทำงานของระบบเชิงเวลาจริงทดสอบจากการรับส่งข้อมูลระหว่างอุปกรณ์โดยใช้ Modbus เป็นกติกาสำหรับการสื่อสารระหว่างกัน การตรวจสอบและพิจารณาความผิดพลาดที่เกิดขึ้นจากการรับส่งข้อมูลบนเงื่อนไขของการปรับเปลี่ยนความถี่ในการส่งข้อมูลของ master และเวลาการตอบสนองกลับที่แตกต่างกัน ความผิดพลาดที่พิจารณาคือความผิดพลาดของข้อมูลที่ได้รับและเวลาที่ใช้ในการตอบสนอง