

Embedded Systems Programming on STM32 MCU

การโปรแกรมระบบสมองกลฝังตัวบน

ไมโครคอนโทรลเลอร์ STM32

1-wire

- 🌿 1-wire ถูกคิดค้นโดย Dallas Semiconductor Corp.
- 🌿 ถูกออกแบบมาเพื่อใช้กับอุปกรณ์ที่กินไฟน้อยและรับส่งข้อมูลด้วยความเร็วต่ำผ่านสายสัญญาณเพียงเส้นเดียว (ไม่นับ ground)
 - 💡 ตามปกติมักจะพบสาย VCC แยกออกมาต่างหาก แต่อุปกรณ์หลายตัวสามารถใช้วิธีต่อวงจรเพิ่มเพื่อนำไฟเลี้ยงจากขาสัญญาณ ผ่านวงจรเก็บประจุ (Diode และ Capacitor) เพื่อนำไปเลี้ยงวงจรได้
- 🌿 ลักษณะการจัดการพื้นฐานของสัญญาณคล้ายคลึงกับ I2C
 - 💡 วงจรขับใช้ Open-drain (open-collector) และใช้ R pull-up เพื่อดึงสัญญาณขึ้น VCC ในขณะที่วงจรขาออกไม่ทำงาน (อุปกรณ์จึงสามารถนำขาสัญญาณนี้ไปจ่ายวงจรเก็บประจุเพื่อใช้เลี้ยงวงจรได้)
 - 💡 การรับส่งข้อมูลกระทำเป็นรายบิต แต่การจัดการแตกต่างจาก I2C ตรงที่ข้อมูลทั้งหมดจะส่งไปในคราวเดียว (คล้ายคลึงกับ SPI ในจุดนี้) แต่อุปกรณ์บางตัวอาจจะต้องการให้ master ส่งสัญญาณลอจิก 0 เพื่อกระตุ้นให้ slave ส่งข้อมูลแต่ละบิต หรืออาจจะใช้วิธีส่งรวดเดียวทุกบิต แต่กำหนด timing ของสัญญาณลอจิก 0/1 ไว้



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique

1-wire : Protocol

การรีเซ็ตอุปกรณ์ slave

- อุปกรณ์ master ดึงลอจิก 0 ด้วยคาบเวลาที่กำหนด
- slave ดึงลอจิก 0 ด้วยคาบเวลาที่กำหนด เพื่อตอบรับว่าพร้อมทำงาน

การเขียนข้อมูลจาก master ไปยัง slave

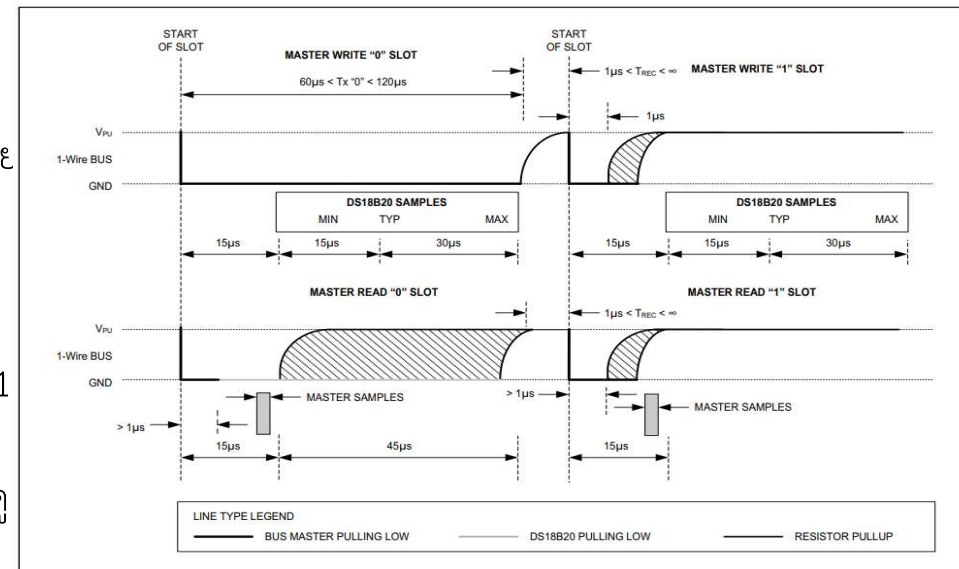
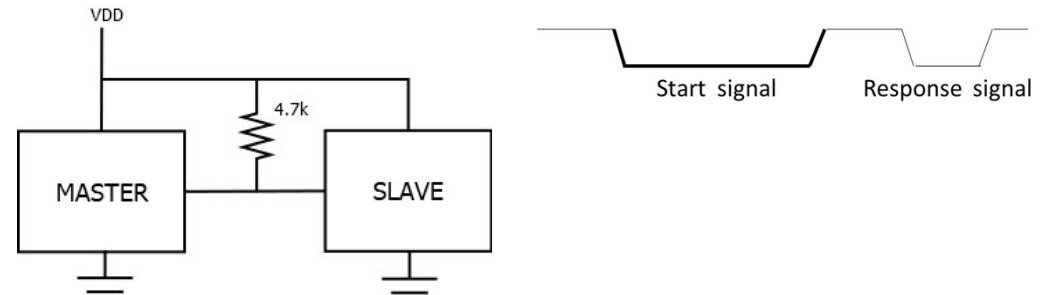
- master ส่งบิตข้อมูลแต่ละบิตไปยัง slave โดยแต่ละบิตประกอบไปด้วยระดับลอจิก 0 และ 1 ที่มีคาบเวลาที่แตกต่างกัน

บิต 0 ส่งโดยส่งลอจิก 0 ไปด้วยคาบเวลายาวมากกว่า แล้วตามด้วยลอจิก 1 ที่คาบเวลาประมาณค่าหนึ่ง

บิต 1 ส่งโดยส่งลอจิก 0 ด้วยคาบเวลาที่สั้นๆ แต่คาบเวลาลอจิก 1 จะมีความยาวมากกว่า

slave จะใช้จังหวะขอบขาลงเพื่อเริ่มต้นจับเวลาเพื่ออ่านค่า เพื่อดูสถานะว่าได้ลอจิก 0 หรือ 1

ดูตัวอย่างการส่งข้อมูลของ DS18B20 จะเห็นว่า master ส่งลอจิก 0 ด้วยพัลส์ลบที่มีความยาวประมาณ 60-120 ไมโครวินาที ส่วนลอจิก 1 จะส่งพัลส์ลบที่มีความยาวประมาณ 1 ไมโครวินาที



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique

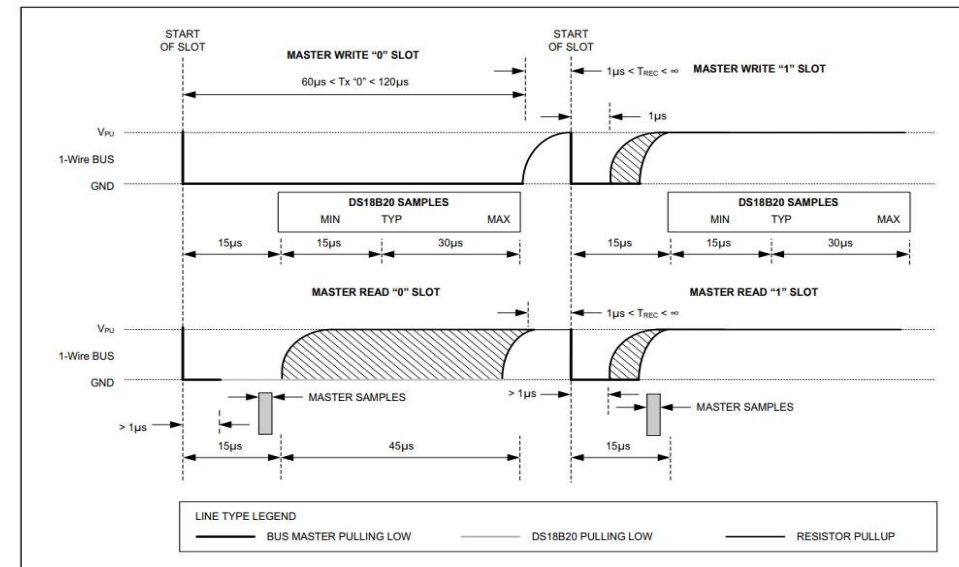
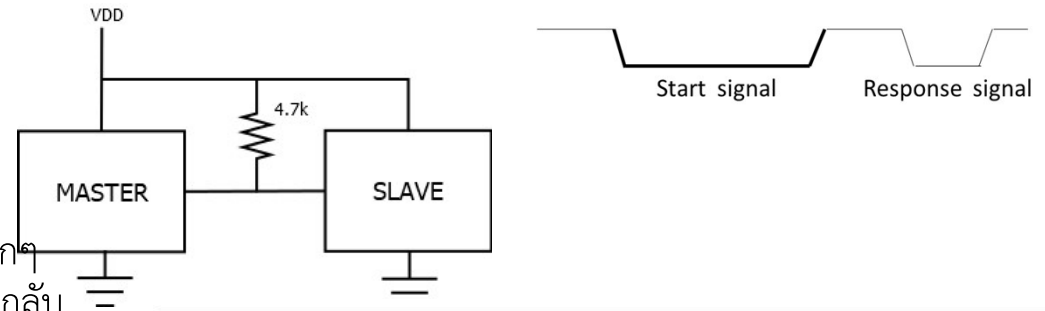
1-wire : Protocol

🌿 การส่งข้อมูลกลับจาก slave ไปยัง master

🌐 กรณีของ DS18B20 master ส่งพัลส์ลบด้วยคาบเวลาสั้นมากๆ ประมาณ 1 ไมโครวินาที จากนั้นรอรออ่านสัญญาณที่ slave ส่งกลับ

🌐 slave ส่งบิต 0 ส่งด้วยพัลส์ลบที่ความยาวประมาณ 15-60 ไมโครวินาที ส่วนลอจิก 1 จะส่งพัลส์ลบที่ความยาวสั้นมากๆ ประมาณ 1 ไมโครวินาที

🌐 master หลังจากส่งพัลส์ลบออกไป รอประมาณ 7-15 ไมโครวินาทีแล้วอ่านสถานะบนบัส



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

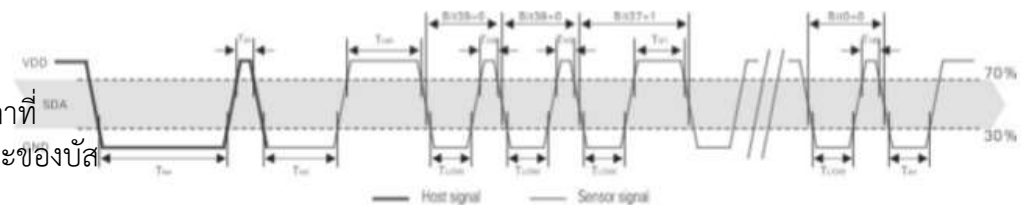
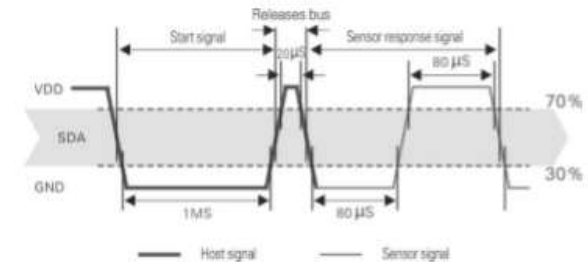
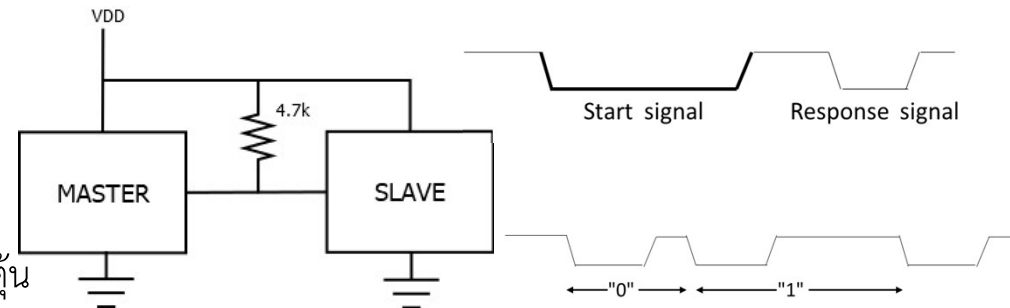
1-wire & Bit-banging technique

1-wire : Protocol

🌱 อีกตัวอย่างของการเขียนข้อมูลจาก master ไปยัง slave

- 🌐 สำหรับ DHT11/22 การรับส่งข้อมูลจะไม่มีให้ master กระตุ้นการรับข้อมูลในแต่ละบิต
- 🌐 การร้องขอข้อมูลเริ่มต้นด้วยการที่ master ส่งพัลส์ลบไปด้วยความยาว 1 มิลลิวินาที แล้วรอให้ slave ตอบกลับเป็นพัลส์ลบด้วยความยาว 80 ไมโครวินาที (คล้ายคลึงกับการ reset ของ DS18B20)
- 🌐 slave ส่งข้อมูลกลับเป็นชุดบิตข้อมูลคำตอบทั้งหมดคืนกลับมา โดยในที่นี้จะส่งกลับมาทั้งหมด 40 บิต (5 ไบต์)

- 🌐 บิต 0 นำเสนอด้วยรูปคลื่นสัญญาณที่ประกอบไปด้วยลอจิกศูนย์คาบเวลา 50 ไมโครวินาที และลอจิกหนึ่งคาบเวลา 70 ไมโครวินาที
- 🌐 บิต 1 นำเสนอด้วยรูปคลื่นสัญญาณที่ประกอบไปด้วยลอจิกศูนย์คาบเวลา 50 ไมโครวินาที และลอจิกหนึ่งคาบเวลา 26 ไมโครวินาที
- 🌐 ดังนั้นเมื่อ master จับจังหวะขอขาขึ้นได้ จะจับเวลาต่อไปด้วยเวลาที่มากกว่า 26 ไมโครวินาที แต่น้อยกว่า 70 ไมโครวินาที แล้วจับสถานะของบัสเพื่อให้ได้คำตอบของบิตข้อมูลนั้นๆ



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique

การจำลองบัสด้วยซอฟต์แวร์ (Bit-banging)

- 🌿 คือการจำลองสัญญาณการรับส่งข้อมูลเพื่อเลียนแบบการทำงานของบัสประเภทต่างๆ ด้วยการเปลี่ยนสถานะของ GPIO ไปมาหรือการอ่านค่าสถานะอินพุตแทนการใช้วงจรรณเฉพาะทาง
- 🌿 ใช้ในกรณีที่ MCU ไม่รองรับบัสที่ต้องการใช้ หรือมีข้อจำกัดอื่นๆ
- 🌿 เนื่องจากการจำลองทางซอฟต์แวร์ การทำ Bitbanging จึงไม่เหมาะกับบัสความเร็วสูง เนื่องจากทำงานไม่ทัน
- 🌿 การจำลองกระทำในระดับ physical และสัญญาณพื้นฐาน
 - 💡 การจำลองทางฮาร์ดแวร์ โดยใช้ GPIO ของ MCU และกำหนดคุณลักษณะทางฮาร์ดแวร์ให้เทียบเคียงกับบัสที่จำลอง
 - 💡 การจำลองลักษณะทางสัญญาณ โดยการสร้างพัลส์/ระดับสัญญาณต่างๆ ให้ได้ตามคาบเวลาที่กำหนดของบัสที่จำลอง



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique

การทำ bit-banging กับ STM32

- 🌿 STM32 ไม่มีบัส 1-wire ดังนั้นจึงต้องจัดการกับอุปกรณ์ 1-wire โดยอาศัยการทำ bit-banging
- 🌿 เนื่องจากบัส 1-wire มีคุณลักษณะพื้นฐานเป็น open-drain ดังนั้นการกำหนด GPIO จึงกระทำในลักษณะของการสลับโหมตไปมาระหว่าง
 - 🔵 โหมตปกติจะเป็นอินพุต ซึ่งอาจกำหนดเป็น R Pull-up หรืออาจใช้ R pull-up ภายนอก
 - 🔵 หากต้องการส่งสัญญาณออกไป จะเปลี่ยนโหมตเป็นเอาต์พุตแบบ push-pull หรือ open-drain (และกำหนด R Pull-up ประกอบ หรือใช้ R pull-up ภายนอกประกอบ)
 - 🔵 เนื่องจากการเปลี่ยนโหมต GPIO ใน STM32 ใช้เวลา (เมื่อเทียบกับการทำงานของอุปกรณ์โดยทั่วไป) ในกรณีที่ทำงานไม่ทัน อาจใช้วิธีการกำหนดขา GPIO ไว้สองตัว ตัวหนึ่งเป็น output แบบ open-drain อีกตัวเป็น input (อาจมี R-pull up) แล้วต่อขาทั้งสองเข้าด้วยกันเพื่อใช้เป็นบัส 1-wire



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique



สรุปหัวข้อ

- 1-wire เป็นบัสสื่อสารข้อมูลแบบอนุกรมความเร็วต่ำที่ถูกนำมาใช้กับอุปกรณ์ที่ไม่ต้องการการสื่อสารแบบซับซ้อน
- อุปกรณ์เหล่านี้ส่วนมากมักกินไฟน้อย ดังนั้นจึงอาจใช้กระแสไฟจากบนสายข้อมูลนำมาเป็นไฟเลี้ยงให้กับอุปกรณ์ ทำให้เหลือสายเพียงเส้นเดียว (ไม่รวม ground) ซึ่งเป็นที่มาของคำว่า 1-wire
- มาตรฐานของการรับส่งข้อมูลอาจมีลักษณะที่แตกต่างกันไปขึ้นกับการกำหนดของอุปกรณ์แต่ละตัว
 - แต่โดยทั่วไปแล้ว จะรับส่งข้อมูลในลักษณะบิตข้อมูลหลายบิตต่อเนื่องไปในคราวเดียว
 - การจำลองบิต 0 และ 1 ทำได้โดยการกำหนดคาบเวลาของช่วงพัลส์บวกให้มีความยาวสั้น (สำหรับบิต 0) หรือยาวมากกว่า (สำหรับบิต 1)



Asst.Prof. Thanwa SRIPRAMONG
PRESENTER

TODAY TOPIC IS

1-wire & Bit-banging technique