


# Embedded Systems Programming on STM32 MCU

การโปรแกรมระบบสมองกลฝังตัวบน


ไมโครคอนโทรลเลอร์ STM32

# ครั้งที่ 2 : การใช้งานพอร์ต GPIO เบื้องต้น

## การอ่านพอร์ต GPIO

-  การอ่านพอร์ตแบบการหยั่งสัญญาณ (polling) และแบบขัดจังหวะ (interrupt)


## switch bouncing

-  debouncing circuit .vs. software debouncing

## การอ่าน switch matrix

-  การใช้ pull-up กับ switch matrix

## การคำนวณคาบเวลาของลอจิกสัญญาณขาเข้าด้วยซอฟต์แวร์

-  การใช้งานเซ็นเซอร์อัลตราโซนิก



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics

# การทำงานแบบการหยั่งสัญญาณ (polling)

🌿 อาศัยการวนรอบอ่านข้อมูลจากช่องทางสัญญาณเข้าจนกว่าจะได้ข้อมูลที่ต้องการ

🌿 ข้อดี

- 🔵 ผู้พัฒนาโปรแกรมเข้าใจง่าย

- 🔵 สามารถประยุกต์กับการเขียนโปรแกรมแบบหลายเธรด (แบ่งเธรดวนอ่านสัญญาณเข้า) ได้

- 🔵 ลดปัญหากับอินพุตที่ไม่ต้องการได้ข้อมูลทุกตัว อินพุตที่อาจจะมีข้อมูลเข้าอย่างรวดเร็วจนอาจทำให้ฟังก์ชันขัดจังหวะทำงานไม่ทัน (เกิด reentrance ซึ่งต้องอาศัยการ disable interrupt เพื่อแก้ปัญหา)

🌿 ข้อด้อย

- 🔵 ประยุกต์กับโปรแกรมที่มีการทำงานอย่างซับซ้อน (และไม่มีการแตกเธรด) ได้ยาก

- 🔵 มีโอกาสไม่สามารถวนไปรับข้อมูลเข้าได้ทันเวลา

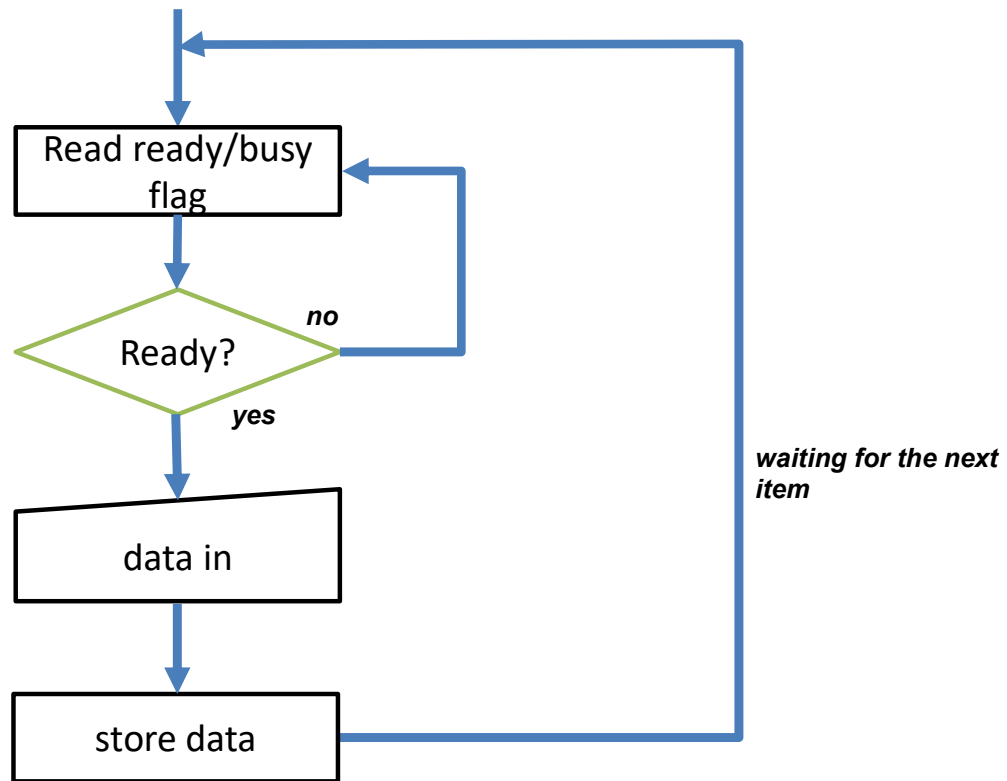


Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics

# การทำงานแบบการหยั่งสัญญาณ (polling- input)



# การทำงานแบบการขัดจังหวะ (interrupt)

- 🌿 อาศัยกลไกทางฮาร์ดแวร์(จากอุปกรณ์รับส่งข้อมูลเข้าออก)ที่สร้างสัญญาณขัดจังหวะ
  - 💡 เมื่อมีข้อมูลใหม่เข้ามา หรือเมื่อส่งข้อมูลออกไปได้เรียบร้อยแล้ว
- 🌿 การกำหนดฟังก์ชันบริการการขัดจังหวะ (Interrupt service routine- ISR) ไว้ล่วงหน้า
  - 💡 ฟังก์ชันดังกล่าวจะถูกเรียกใช้งานเมื่อเกิดการขัดจังหวะตามหมายเลข/อุปกรณ์ที่กำหนด
    - 🌍 CPU จะหยุดการทำงานของงานที่กำลังประมวลผลในปัจจุบัน แล้วกระโดดไปทำงานในฟังก์ชันบริการการขัดจังหวะ เมื่อทำงานเสร็จสิ้นจึงกระโดดกลับมาทำงานเดิมต่อไป
  - 💡 หน่วยประมวลผลกลางมีการจัดเก็บตารางเก็บเลขที่อยู่ฟังก์ชันขัดจังหวะ (Interrupt vector table –IVT)
    - 🌍 ค่าในตารางดังกล่าวโดยปกติจะเปลี่ยนแปลงโดยระบบปฏิบัติการ แต่สำหรับระบบปฏิบัติการสมัยเก่าหรือใน MCU อาจจะออกแบบให้ผู้พัฒนาโปรแกรมเปลี่ยนแปลงค่าดังกล่าวได้โดยตรง
    - 🌍 ใน **STM-32 จัดการผ่านไลบรารี**ที่ ST เตรียมไว้ให้ (Hardware application layer- HAL)
      - 🌍 อุปกรณ์ I/O แต่ละประเภท ผู้พัฒนาโปรแกรมจะเขียนฟังก์ชันเรียกกลับ (Callback function) เพื่อจัดการแยกกันไป



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

**GPIO Basics**

# การทำงานแบบการขัดจังหวะ (interrupt)

## ข้อดี

- สามารถรับข้อมูลที่เข้ามาได้โดยไม่ตกหล่น
  - แต่ในทางปฏิบัติจะต้องออกแบบไม่ให้ ISR ทำงานซ้ำเกินกว่าชุดข้อมูลถัดไปจะเข้ามา ไม่งั้นนั่นจะเกิด reentrance (สถานะที่ CPU จะเรียก ISR ตัวเดิมทำงานซ้ำในขณะที่ยังทำงานกับข้อมูลเดิมไม่สำเร็จ)
  - แก้ปัญหาโดยการยกเลิกการขัดจังหวะ I/O ดังกล่าวในระหว่างเกิด ISR แต่ก็ส่งผลทำให้ข้อมูลอาจตกหล่นได้
- สามารถออกแบบระบบปฏิบัติการให้บริหารจัดการกับ I/O จำนวนมากได้
  - ระบบปฏิบัติการยุคใหม่ที่ใช้การขัดจังหวะเป็นหลัก จึงถูกเรียกว่าเป็น Interrupt driven OS

## ข้อด้อย

- โปรแกรมอาจดูว่ามีความซับซ้อนมากขึ้นสำหรับนักโปรแกรมมือใหม่
- การพัฒนาโปรแกรมแบบหลายเธรด อาจจะต้องมีการกำหนดว่าจะให้เธรดใดทำหน้าที่ตอบสนองต่อ I/O ใด
- ต้องออกแบบฮาร์ดแวร์/ซอฟต์แวร์ ในการควบคุมไม่ให้ฟังก์ชันจัดการอินเทอร์รัปต์ใช้เวลาที่ไม่แน่นอน หรือเวลานานเกินไปที่จะส่งผลต่อการตอบสนอง I/O ต่างๆ ของระบบทันเวลา (Real-time systems)
- การพัฒนาฟังก์ชันบริการการขัดจังหวะ ต้องออกแบบให้ใช้เวลาที่สั้นที่สุด (ลดโอกาส reentrance)

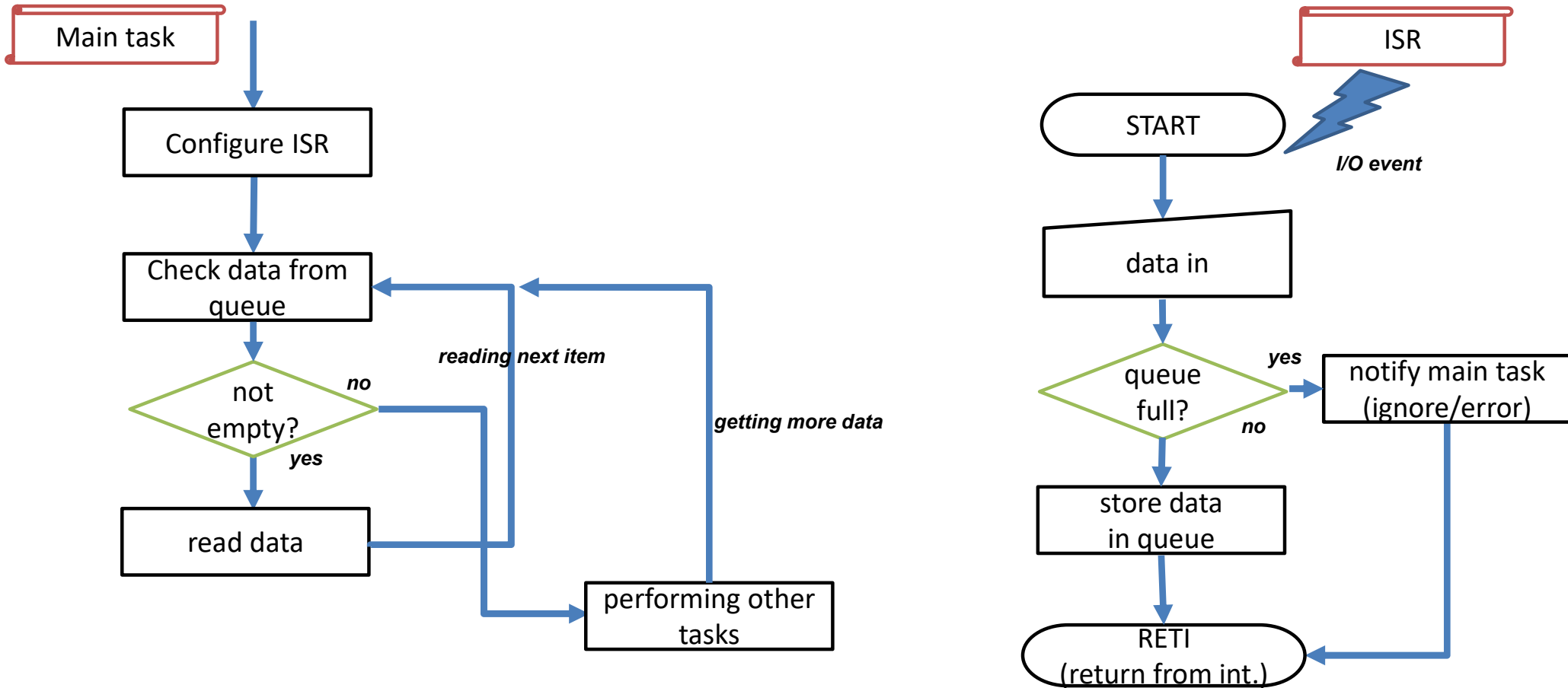


Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

**GPIO Basics**

# การทำงานแบบการขัดจังหวะ (interrupt - input)



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics



# Switch debouncing

🌿 การกดสวิตช์โดยผู้ใช้งานทั่วไป อาจส่งผลทำให้เกิดการเปลี่ยนแปลงสัญญาณลอจิกอย่างรวดเร็วในจังหวะที่กดหรือปล่อยสวิตช์

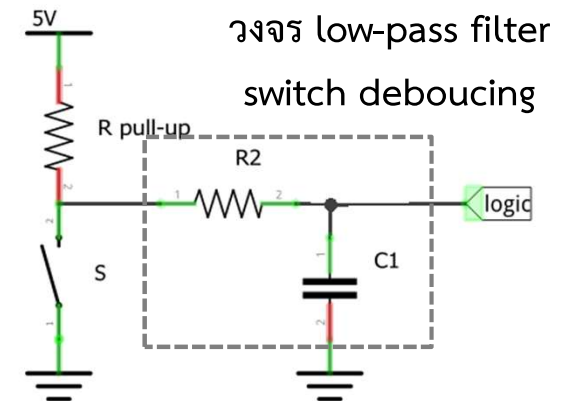
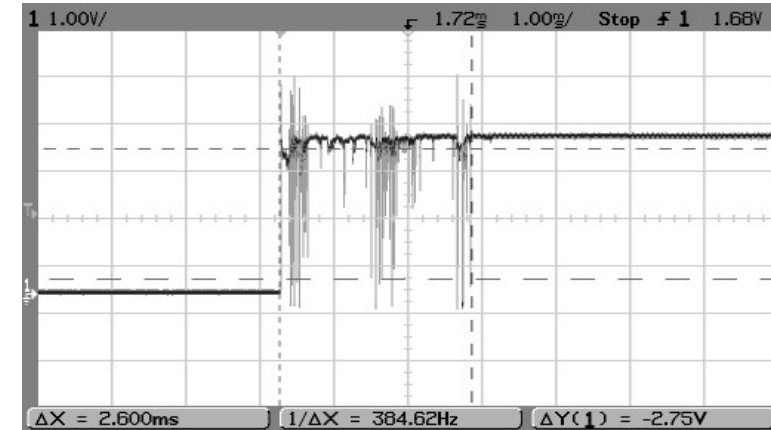
💡 หน้าสัมผัสไม่สะอาด, มือผู้ใช้ไม่นิ่ง ฯลฯ

🌿 แก้ปัญหาได้ทั้งวิธีทางฮาร์ดแวร์และซอฟต์แวร์

💡 ใช้วงจร Set-Reset latch (SR debouncer)

💡 ใช้วงจร Low-pass filter

💡 ใช้ซอฟต์แวร์จับเวลาหน่วง (ไม่ถือว่าเป็นการกดสวิตช์หากคาบเวลาต่ำกว่าที่กำหนด)



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics



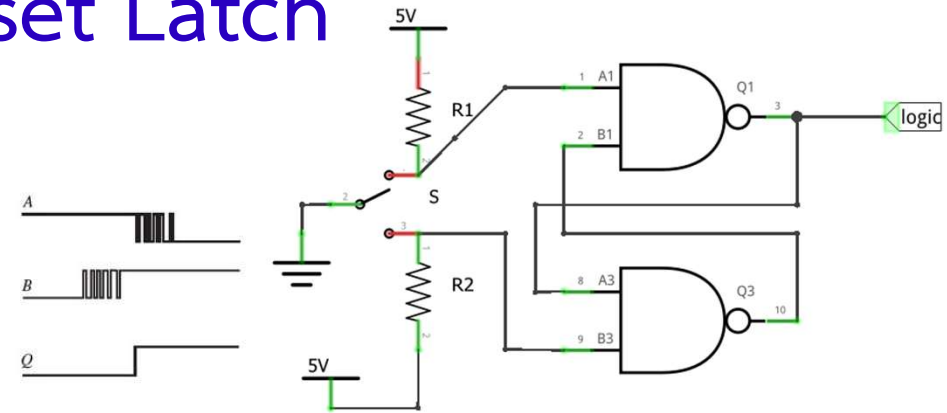
# Switch debouncing : Set-Reset Latch

🌿 ใช้สวิตช์แบบสองทาง

🔵 ตัวสวิตช์จะไม่สัมผัสที่หน้าสัมผัสทั้งสองทางพร้อมกัน

🎨 ส่งผลให้เกิด switch bounce เฉพาะที่ขา A1 หรือ B3 ในเวลาใดเวลาหนึ่งแต่จะไม่พร้อมกัน

🔵 ใช้ NAND gate จำนวนสองตัว



วงจร SR debouncing  
switch debouncing

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0

NAND truth-table



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS  
GPIO Basics

# Switch debouncing : Low-pass filter

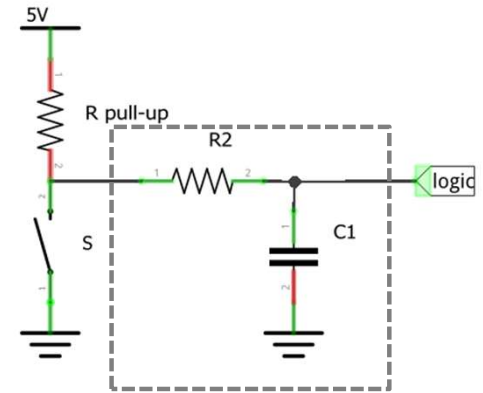
🌿 ใช้สวิตช์แบบทางเดียว ร่วมกับวงจร low-pass filter

💡 ใช้กับ R pull-up หรือ pull-down

💡 R2 ร่วมกับ C1 ทำหน้าที่เป็น Low-pass filter (C เก็บประจุในจังหวะที่หน้าสัมผัสสวิตช์ติด ทำให้ระดับแรงดันขาออกไม่เปลี่ยนแปลงอย่างรวดเร็วตามสวิตช์)

💡 ตัวอย่างค่าที่อาจใช้ R pull-up 10k R2 = 1k C1 = 0.1 $\mu$ F

🌈 ค่าที่แตกต่างจากนี้จะส่งผลต่อความไวของการเปลี่ยนแปลงระดับแรงดัน (ส่งผลต่อความเร็วในการตอบสนองของการกดปุ่ม)



วงจร Low-pass filter  
switch debouncing



# Switch debouncing : Software

🌿 ใช้สวิตช์แบบทางเดียว ร่วมกับ R pull-up

💧 อาจใช้ผสมผสานกับวงจรแบบ low-pass filter แต่ไม่จำเป็นนัก

💧 ใช้วิธีการเขียนโปรแกรมแบบ polling หรือ interrupt

🌍 เก็บค่าฐานเวลาเมื่อพบว่าการกดสวิตช์ (ตรวจระดับลอจิกจาก 1 เป็น 0)

🌍 เมื่อมีการตรวจพบการปล่อยสวิตช์ (เช่น ตรวจระดับลอจิกจาก 0 เป็น 1 = ปล่อยสวิตช์) ให้อ่านค่าฐานเวลามาเพื่อเทียบกับค่าฐานเวลาก่อนหน้า

🌍 หากค่าความแตกต่างน้อยกว่าที่กำหนด พิจารณาว่าเป็น switch bounce และไม่ทำงานใดๆ ต่อ

🌍 หากค่าความแตกต่างมากกว่าที่กำหนด พิจารณาว่าเป็นการกดสวิตช์



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics

# Switch matrix

🌿 สวิตช์แต่ละตัวต่อขาข้างหนึ่งร่วมกันในแถวเดียวกัน และอีกข้างหนึ่งต่อร่วมกันในคอลัมน์เดียวกัน

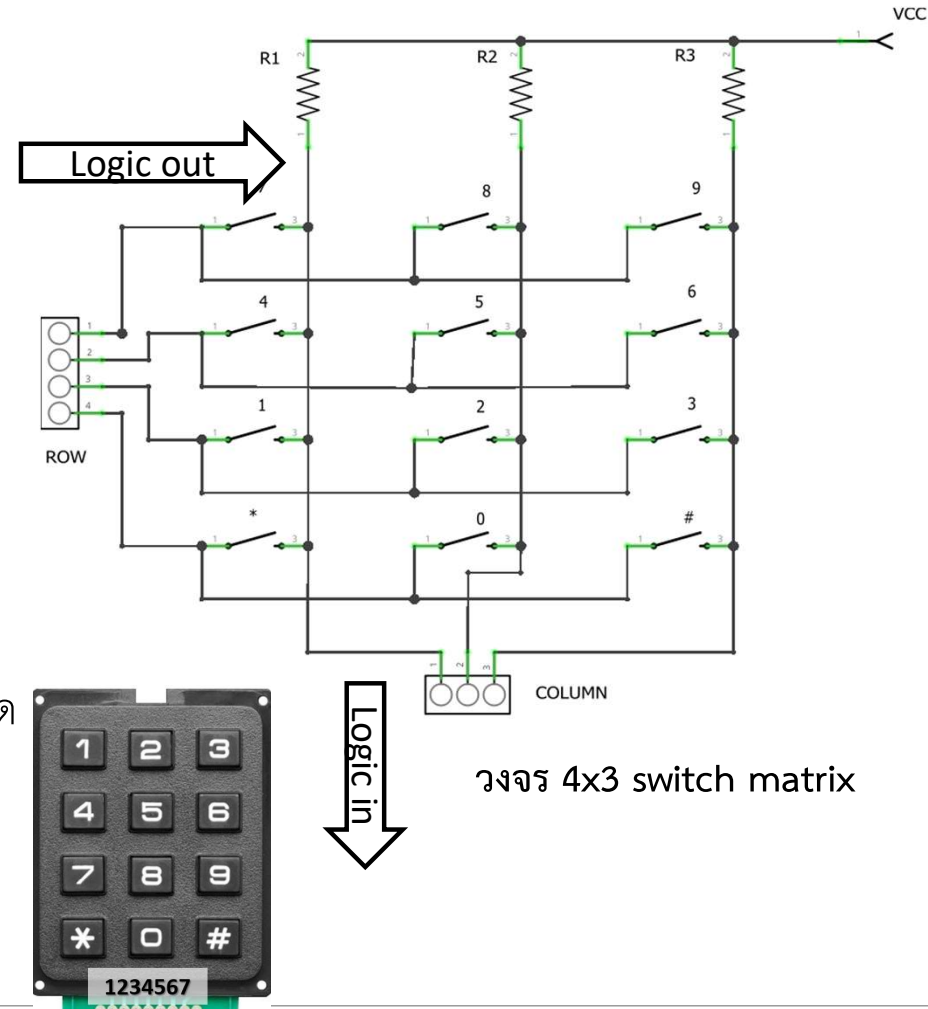
🌿 นิยมใช้ R pull-up ฝั่งอ่านค่าด้วย GPIO input

🌿 หลักการทำงาน

💡 ปกติส่งสัญญาณลอจิก 1 ให้ทุกขา output

🎨 เมื่ออ่านค่าจากฝั่ง input จะได้ลอจิก 1 ทุกตำแหน่ง

💡 อ่านค่าด้วยการหยั่งสัญญาณ (polling) โดยส่งลอจิก 0 ให้ขา output ทีละขา แล้วอ่านค่าจากฝั่ง input เพื่อดูว่ามีตำแหน่งใดที่มีลอจิก 0 แสดงว่าผู้ใช้กดปุ่มที่ลัดวงจรระหว่างแถว-คอลัมน์ดังกล่าว



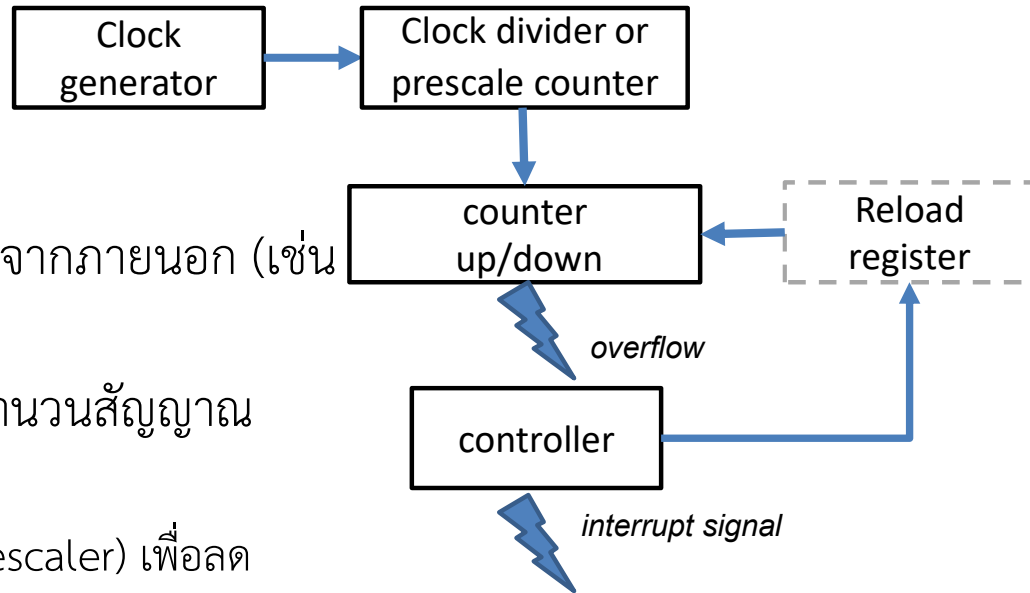
Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics

# Timer / Counter

- 🌿 วงจรฐานเวลาใน MCU และคอมพิวเตอร์โดยทั่วไป
- 🌿 ใช้ฐานเวลาจากภายในระบบ หรือรับสัญญาณนาฬิกาจากภายนอก (เช่น จาก GPIO input)
- 🌿 มีวงจรนับขึ้น / ลง ที่จะถูกเปลี่ยนค่าไปเรื่อยๆ ตามจำนวนสัญญาณนาฬิกาที่ได้รับ
  - 💧 อาจจะมีวงจรหาร (divider) หรือวงจรนับคั่นกลาง (prescaler) เพื่อลดจำนวนสัญญาณนาฬิกาลงให้อยู่ในคาบเวลาที่เหมาะสม
    - 🎨 สมมติเช่น สัญญาณนาฬิกาเข้าเป็น 84MHz ผ่านวงจร prescale counter ที่กำหนดค่านับลงเป็น 83 (84-1) จะทำให้เกิดสัญญาณนาฬิกาออก 1 ครั้งต่อสัญญาณนาฬิกาเข้า 84 ครั้ง
- 🌿 วงจรควบคุมหลักทำหน้าที่โหลดค่านับใหม่เมื่อนับครบ และทำหน้าที่สร้างสัญญาณขัดจังหวะให้กับ CPU ต่อไป



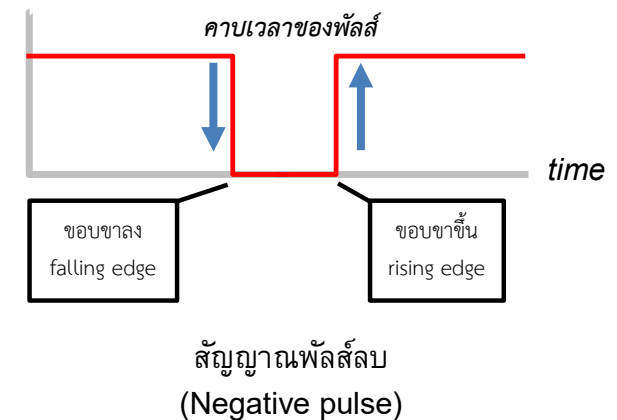
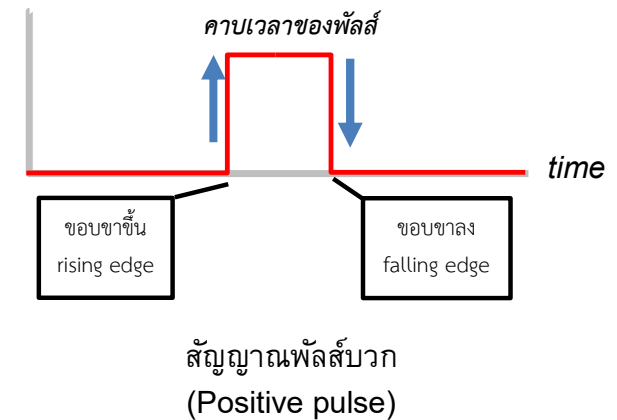
Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics

# Pulse (signal)

- 🌿 สัญญาณพัลส์ หมายถึงลักษณะการเปลี่ยนแปลงทางลอจิกเป็นคาบเวลาสั้นๆ เมื่อเทียบกับระดับลอจิกของขาสัญญาณดังกล่าวตามปกติ
- 🌿 สำหรับขาสัญญาณที่ปกติมีลอจิกเป็นศูนย์ จะเกิดพัลส์บวก หมายถึงช่วงคาบเวลาที่ระดับลอจิกเปลี่ยนจาก 0 เป็น 1 และเปลี่ยนกลับลงมาจาก 1 เป็น 0
  - 💡 ช่วงเวลาที่ระดับลอจิกเปลี่ยนจาก 0 เป็น 1 เรียกว่าขอบขาขึ้น (rising edge)
  - 💡 ช่วงเวลาที่ระดับลอจิกเปลี่ยนจาก 1 เป็น 0 เรียกว่าขอบขาลง (falling edge)
- 🌿 วงจรลอจิกโดยทั่วไปใช้สัญญาณพัลส์เพื่อเป็นสัญญาณกระตุ้นไปยังวงจรอื่น
  - 💡 การกระตุ้นเพื่อให้เริ่มทำงาน
  - 💡 การส่งสัญญาณกลับมาในลักษณะของสัญญาณขัดจังหวะ (interrupt signal) เพื่อแจ้งว่าได้ทำงานเสร็จสิ้นแล้ว

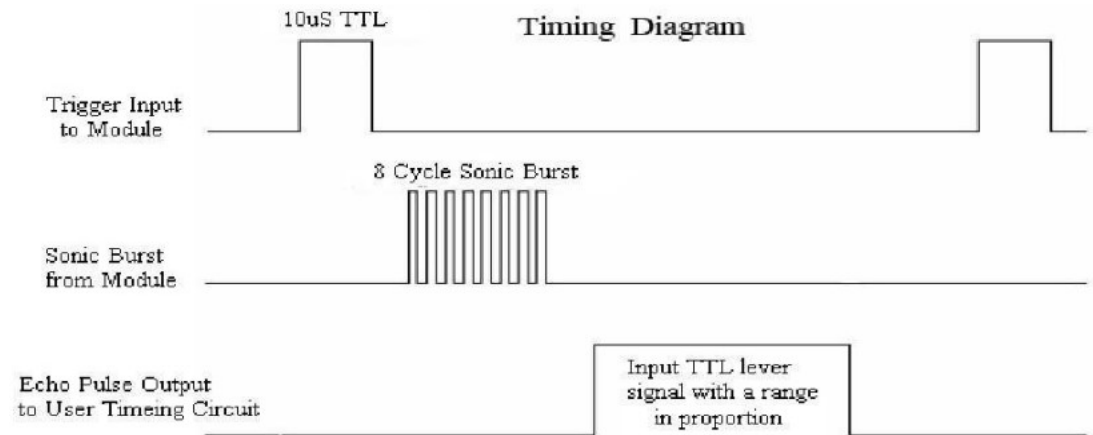


Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS  
GPIO Basics

# เซ็นเซอร์อัลตราโซนิก

- ตัวเซ็นเซอร์ประกอบไปด้วยวงจรสร้างสัญญาณความถี่สูง (ประมาณ 40kHz) โดยจะส่งสัญญาณออกไปทาง transmitter เมื่อได้รับสัญญาณพัลส์กระตุ้นจากภายนอก
- เสียงที่ถูกสร้าง เดินทางไปกระทบวัตถุและสะท้อนกลับเข้าสู่ receiver ทำหน้าที่ตอบสนองต่อแรงดันอากาศที่เปลี่ยนไป(ในย่านความถี่ 40kHz) และแปลงเป็นกระแส/แรงดัน ผ่านวงจรขยายสัญญาณที่นำสัญญาณทั้งฝั่งที่ส่งสัญญาณออกไปและสัญญาณที่สะท้อนกลับ ได้สัญญาณพัลส์ที่มีคาบเวลาเท่ากับช่วงเวลาที่ปล่อยสัญญาณออกไปจนถึงเวลาที่รับสัญญาณกลับเข้ามา
- คาบเวลาดังกล่าวคือช่วงเวลาที่เสียงเดินทางไป  
ในอากาศ



Asst.Prof. Thanwa SRIPRAMONG

PRESENTER

TODAY TOPIC IS

GPIO Basics





## สรุปหัวข้อ

- 🌿 การทำงานแบบหยั่งสัญญาณ (polling) เป็นลักษณะการวนรอบการอ่านค่าจาก input จนกว่าจะได้ข้อมูลที่ต้องการ
- 🌿 การทำงานแบบขัดจังหวะ (interrupt) เป็นลักษณะการกำหนดฟังก์ชันบริการการขัดจังหวะไว้ล่วงหน้าให้กับสัญญาณขัดจังหวะของ I/O ที่เกี่ยวข้อง เมื่อ I/O ทำงานเสร็จตามขั้นตอน จะเกิดสัญญาณขัดจังหวะ และในที่สุด ฟังก์ชันบริการการขัดจังหวะจะดำเนินการตอบสนอง (เช่นการอ่านค่าจาก input ที่ได้)



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics



## สรุปหัวข้อ (ต่อ)

- 🌿 การใช้งานสวิตช์เพื่อสร้างสัญญาณลอจิกให้กับ MCU มักเกิดปัญหา switch bounce
  - 💡 สามารถแก้ปัญหาได้โดยการใช่วงจร RS-Latch หรือ lowpass filter หรือใช้กลไกทางซอฟต์แวร์จัดการได้
- 🌿 เราสามารถนำสวิตช์หลายตัวมาต่อกันในรูปแบบของ switch matrix เพื่อลดจำนวนขาสัญญาณที่ต้องใช้กับสวิตช์จำนวนมาก
  - 💡 ส่วน input สามารถต่อ lowpass filter เพื่อแก้ปัญหา switch bounce ได้เช่นเดียวกัน
- 🌿 วงจร timer/counter เป็นวงจรพื้นฐานใน MCU โดยทั่วไป ใช้สร้างสัญญาณนาฬิกาที่มีคาบเวลาตามที่ต้องการ หรือใช้กำเนิดสัญญาณขั้วจังหวะที่มีคาบเวลาที่คงที่ สำหรับนำไปใช้งานต่างๆ ต่อไป



Asst.Prof. Thanwa SRIPRAMONG  
PRESENTER

TODAY TOPIC IS

GPIO Basics