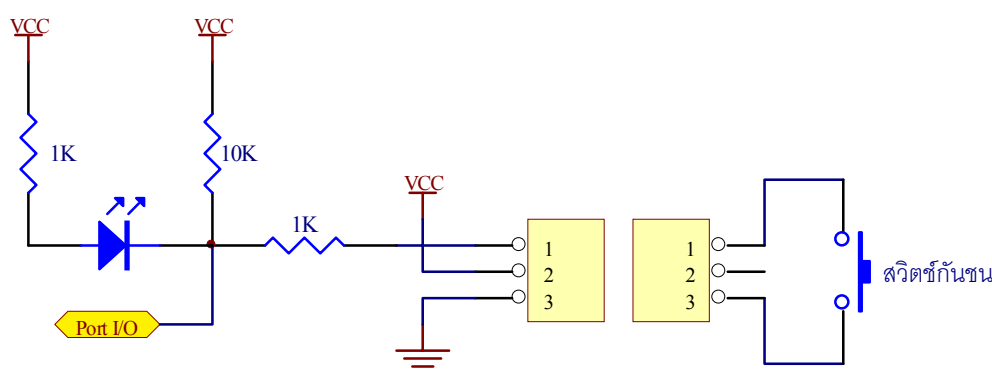


## การตรวจสอบการชนของรถหุ่นยนต์ด้วย Switch

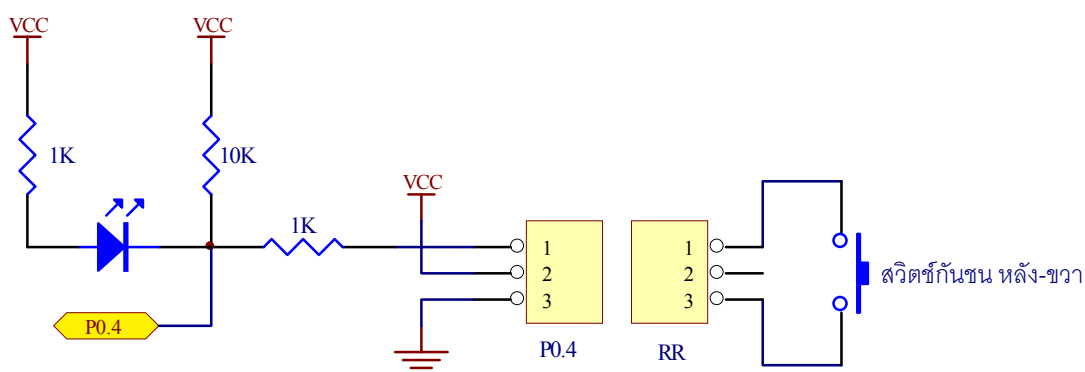
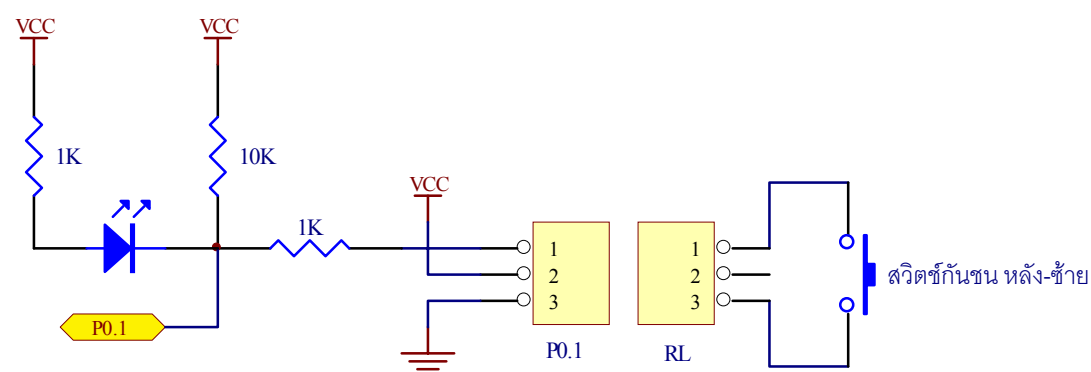
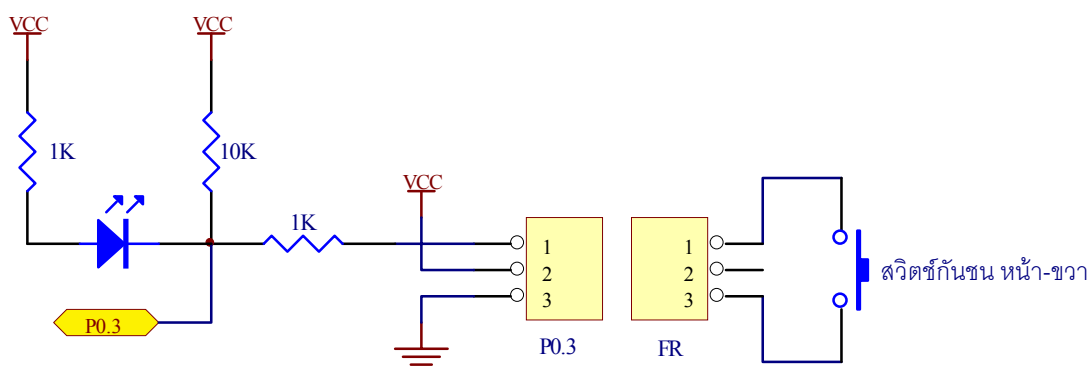
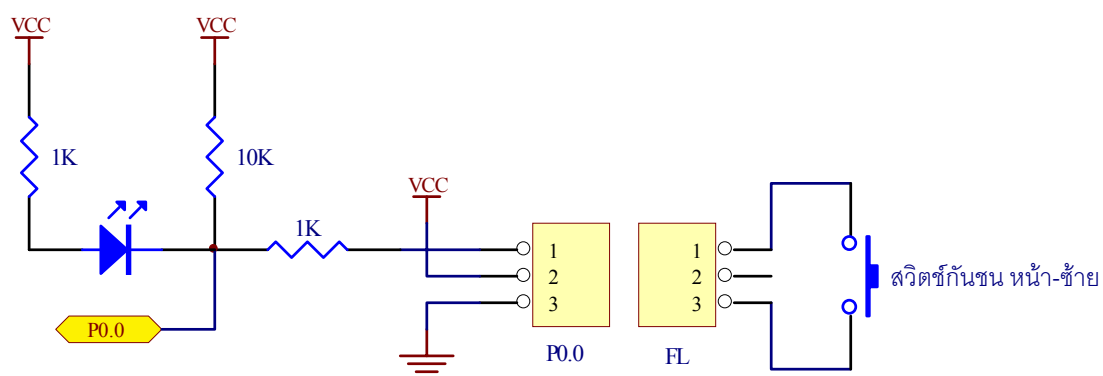
การศึกษางานของหุ่นยนต์นั้น หัวใจสำคัญที่สุดอยู่ที่ขั้นตอนของการเขียนโปรแกรมสำหรับควบคุมการทำงานของหุ่นยนต์ให้สามารถทำงานตามเงื่อนไขที่เรากำหนดไว้ได้ ซึ่งการที่จะสามารถสร้างระบบหุ่นยนต์ให้มีความเฉลียวฉลาดได้มากนักน้อยเพียงใดนั้น ปัจจัยหลักอย่างหนึ่งก็คือ การติดตั้งอุปกรณ์ตรวจจับแบบต่างๆ ให้กับหุ่นยนต์ เพื่อใช้ในการตรวจจับสภาพแวดล้อมภายนอกรอบๆ ตัวหุ่นยนต์อันจะเป็นหนทางเพื่อนำไปสู่การวิเคราะห์และตัดสินใจสั่งงานระบบกลไกต่างๆ เพื่อตอบสนองต่อเงื่อนไขต่างๆ ที่ตรวจจับได้ สำหรับระบบโครงสร้างของรถหุ่นยนต์ ET-ROBOT RD2 ก็เช่นเดียวกัน ในชุดมาตรฐานจะมีการติดตั้งอุปกรณ์สำหรับตรวจจับ Input แบบต่างๆ ไว้ 2 ส่วนด้วยกัน คือ Switch สำหรับตรวจสอบการชน ด้านหน้าและด้านหลัง และแผงวงจรสำหรับตรวจจับเส้นขนาด 3 จุด สำหรับใช้นำทางให้รถหุ่นยนต์เคลื่อนที่ไปตามแนวเส้น ซึ่งในบทนี้เราจะมาทดลองศึกษาเรียนรู้เกี่ยวกับวิธีการในการตรวจสอบการชนของรถหุ่นยนต์รุ่น ET-ROBOT RD2 ด้วย Switch กันดู เพื่อเป็นแนวทางในการนำไปประยุกต์ใช้งานต่อไป

### การทำงานของ Switch ตรวจสอบการชน

สำหรับ Switch ตรวจสอบการชน ของรถหุ่นยนต์ รุ่น ET-ROBOT-RD2 นั้น จะเป็น สวิตช์ ชนิดหน้าสัมผัสแบบ Micro Switch ซึ่งจะมีการติดตั้งไว้ในตำแหน่ง ทางฝั่ง ซ้าย และ ขวา ของตัวรถ โดยแบ่งออกเป็น 4 ตำแหน่ง คือ ด้านหน้าฝั่งซ้าย, ด้านหน้าฝั่งขวา, ด้านหลังฝั่งซ้าย และ ด้านหลังฝั่งขวา ตามลำดับ โดยวงจรการทำงานของสวิตช์ ตรวจสอบการชน ทั้ง 4 จุด จะมีลักษณะเหมือนกัน เพียงแต่เชื่อมต่อกับสัญญาณ Pin Port ที่แตกต่างกัน โดยลักษณะของวงจร เป็นดังรูป



รูปแสดง วงจรสวิตช์ตรวจสอบกันชน ขนาด 1 จุด ของรถหุ่นยนต์ ET-ROBOT-RD2



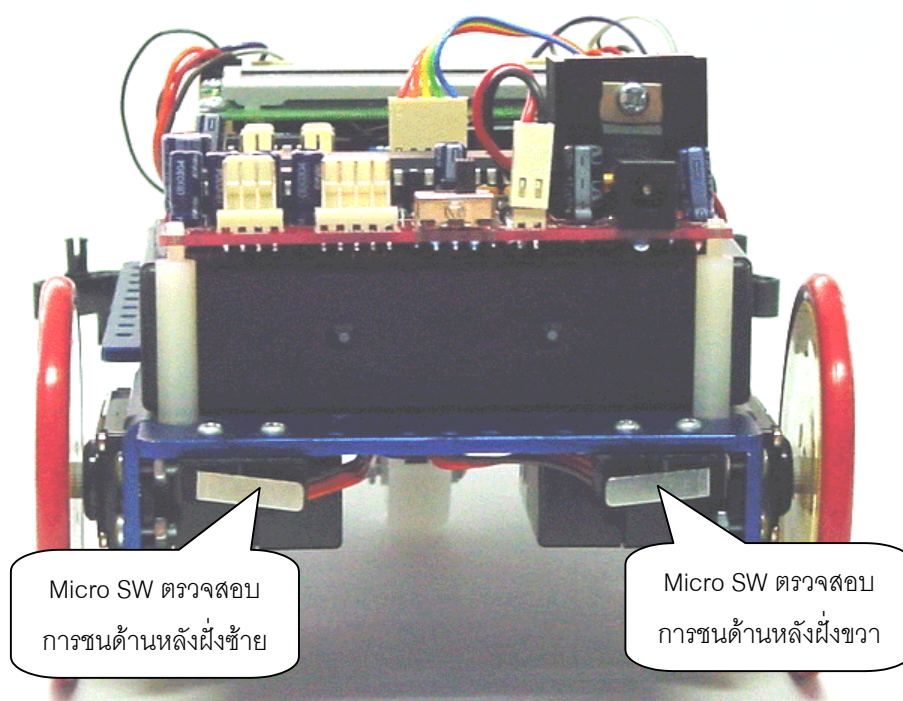
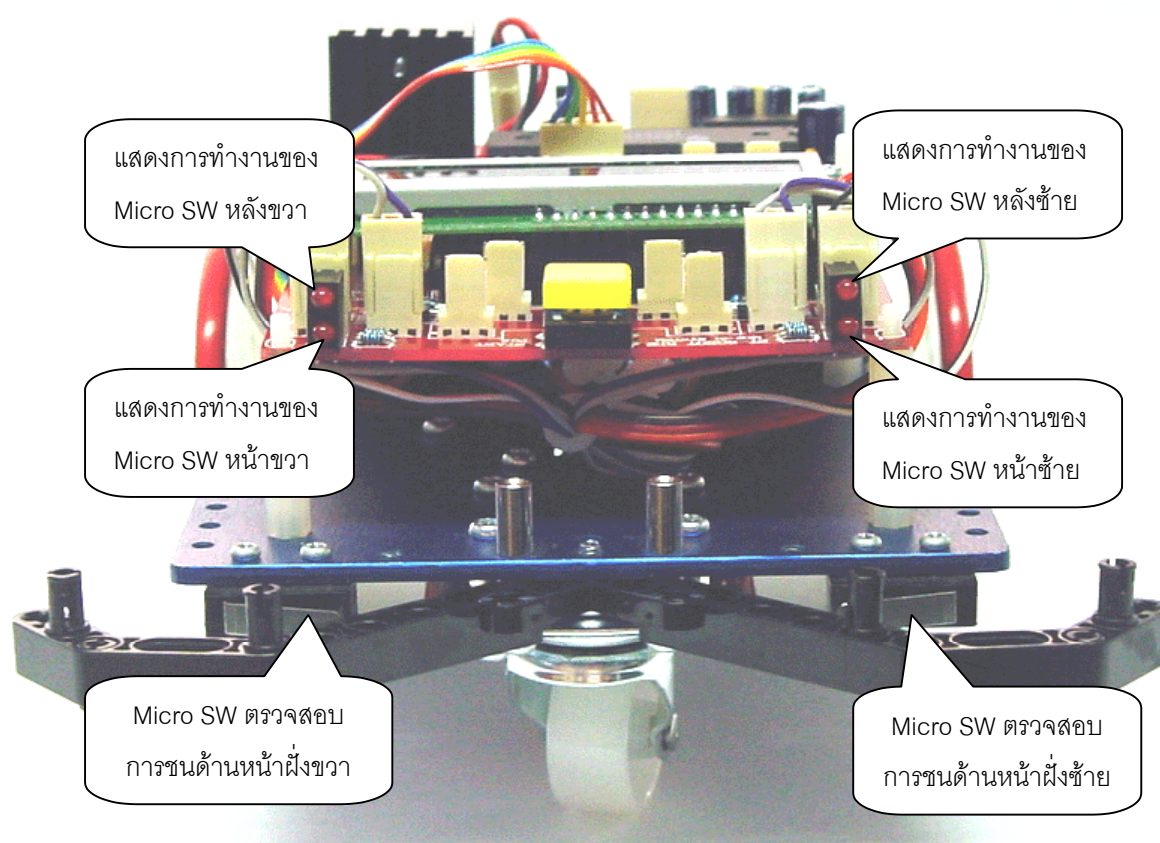
รูปแสดง วงจรสวิตช์ตรวจสอบการชน ของรถหุ่นยนต์ รุ่น ET-ROBOT-RD2

สำหรับการทำงานของวงจรนั้น จะเห็นได้ว่า ตามปกติแล้ว พอร์ต I/O สำหรับใช้ตรวจจบการทำงานของสวิตช์ตรวจสอบการชน จะถูกต่อวงจร Pull-Up รอไว้เรียบร้อยแล้ว ดังนั้นในสถานะปกติสถานะของ Input จะมีค่าเป็น “1” เสมอ แต่เมื่อสวิตช์กันชนถูกกด จะทำให้น้ำสัมผัสของสวิตช์ลัดวงจรเข้าหากัน ซึ่งจะส่งผลให้เกิดการลัดวงจรจาก พอร์ต I/O ลง Ground โดยผ่านตัวต้านทานค่า 1Kohm ซึ่งในกรณีนี้จะทำให้สถานะ Input ของพอร์ตมีค่าเป็น “0” ดังนั้นเราจึงสามารถตรวจสอบการทำงานของสวิตช์ได้โดยการตรวจสอบสถานะทางลอจิกของพอร์ต I/O ที่เชื่อมต่ออยู่กับสวิตช์กันชน ตำแหน่งต่างๆ โดยถ้าบิตใดมีค่าเป็น “0” ก็แสดงว่าสวิตช์ตำแหน่งนั้นๆถูกกดให้ทำงานหรือเกิดการชนในตำแหน่งนั้นๆแล้ว

### การตรวจสอบสวิตช์ สำหรับตรวจสอบการชน

สำหรับการทดสอบการทำงานของไมโครสวิตช์ ที่ใช้ตรวจสอบการชน ของรถหุ่นยนต์นั้น สามารถทำได้โดยการกดก้านสวิตช์เพื่อให้สวิตช์ทำงาน โดยสถานะปกติหน้าสัมผัสของสวิตช์จะเปิดอยู่ แต่เมื่อทำการกดก้านสวิตช์หน้าสัมผัสของสวิตช์จะปิดเข้าหากัน ซึ่งจะทำให้สถานะทางลอจิกของหน้าสัมผัสสวิตช์ด้านที่ต่อกับ Port I/O อยู่มีค่าเป็น “0” และ LED แสดงผลจะติดสว่าง สำหรับการทดสอบการทำงานของไมโครสวิตช์ อย่างง่ายๆ สามารถทำได้โดยการกดก้านสวิตช์เพื่อให้หน้าสัมผัสสวิตช์ลัดวงจรเข้าหากันแล้ว สังเกตผลการทำงานจากหลอดแสดงผล LED โดยมีวิธีการตรวจสอบดังนี้

- ตรวจสอบการชนด้านหน้าฝั่งซ้าย ให้ทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งซ้าย พร้อมกับสังเกต LED แสดงผลที่อยู่มุมด้านหน้าซ้าย ซึ่งถ้าทุกอย่างถูกต้อง LED ดวงที่อยู่ด้านล่างจะติดสว่าง เมื่อทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งซ้าย และ LED จะดับเมื่อ OFF ไมโครสวิตช์
- ตรวจสอบการชนด้านหน้าฝั่งขวา ให้ทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งขวา พร้อมกับสังเกต LED แสดงผลที่อยู่มุมด้านหน้าขวา ซึ่งถ้าทุกอย่างถูกต้อง LED ดวงที่อยู่ด้านล่างจะติดสว่าง เมื่อทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งซ้าย และ LED จะดับเมื่อ OFF ไมโครสวิตช์
- ตรวจสอบการชนด้านหลังฝั่งซ้าย ให้ทำการ ON ไมโครสวิตช์ด้านหลังฝั่งซ้าย พร้อมกับสังเกต LED แสดงผลที่อยู่มุมด้านหน้าซ้าย ซึ่งถ้าทุกอย่างถูกต้อง LED ดวงที่อยู่ด้านบนจะติดสว่าง เมื่อทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งซ้าย และ LED จะดับเมื่อ OFF ไมโครสวิตช์
- ตรวจสอบการชนด้านหลังฝั่งขวา ให้ทำการ ON ไมโครสวิตช์ด้านหลังฝั่งขวา พร้อมกับสังเกต LED แสดงผลที่อยู่มุมด้านหน้าขวา ซึ่งถ้าทุกอย่างถูกต้อง LED ดวงที่อยู่ด้านบนจะติดสว่าง เมื่อทำการ ON ไมโครสวิตช์ด้านหน้าฝั่งซ้าย และ LED จะดับเมื่อ OFF ไมโครสวิตช์



รูปแสดง Micro Switch สำหรับตรวจสอบการชนของ ET-ROBOT RD2

## การเขียนโปรแกรมตรวจสอบการชนของรถหุ่นยนต์ด้วยสวิทช์

สำหรับตัวอย่างโปรแกรมของการตรวจสอบการชนด้วยภาษาเบสิกของ BASCOM-8051 นั้นสามารถทำได้ 2 แบบ คือการอ่านค่าสถานะ Pin Port เข้ามาตรวจสอบ โดยการอ่านทั้ง Byte และการสังตรวจสอบสถานะของ Pin Port เฉพาะ Bit ที่ต้องการ ซึ่งในกรณีที่ต้องการตรวจสอบสถานะของ Input จากพอร์ตเพียงบิตใดบิตหนึ่งนั้น ในกรณีที่ใช้โปรแกรมภาษาเบสิก ของ BASCOM-8051 สามารถใช้คำสั่งเฉพาะในการตรวจสอบสถานะของ Input บิตที่ต้องการได้ โดยใช้คำสั่งต่างๆดังนี้

<b>BITWAIT Px.y,Set</b>	‘รอจนกระทั่งสถานะบิต y ของพอร์ต Px ถูกเซตเป็น “1”
<b>BITWAIT Px.y,Reset</b>	‘รอจนกระทั่งสถานะบิต y ของพอร์ต Px ถูกรีเซตเป็น “0”
<b>Debounce Px.y , 1 , Sub1</b>	‘ถ้าพบว่าสถานะบิต y ของพอร์ต Px เป็น “1” (หลัง Debounce) ‘จึงจะกระโดดไปโปรแกรมย่อย Sub1
<b>Debounce Px.y , 0 , Sub1</b>	‘ถ้าพบว่าสถานะบิต y ของพอร์ต Px เป็น “0” (หลัง Debounce) ‘จึงจะกระโดดไปโปรแกรมย่อย Sub1

การใช้คำสั่ง BITWAIT ในการตรวจสอบการทำงานของ Input Pin สามารถกำหนดสถานะการทำงานของคำสั่งได้ 2 รูปแบบคือ ถ้าใช้เงื่อนไข Set จะเป็นการสั่งให้โปรแกรมวนรอบตรวจสอบสถานะของ Input บิตที่กำหนดไว้จนกว่าจะพบว่าสถานะของ Input บิตนั้นมีค่าเป็น “1” โปรแกรมจึงจะข้ามไปทำงานยังคำสั่งถัดไป ซึ่งถ้าหากพบว่าสถานะของ Input ยังคงเป็น “0” อยู่โปรแกรมก็ยังคงวนรอบตรวจสอบสถานะของ Input อยู่เช่นเดิม แต่ถ้าใช้เงื่อนไขเป็น Reset ก็จะเป็นการสั่งให้โปรแกรมวนรอบตรวจสอบสถานะของ Input จนกว่าจะพบว่า Input บิตที่กำหนดไว้มีสถานะเป็น “0” จึงจะจบการทำงานจากคำสั่งนี้ เพื่อข้ามไปทำงานยังคำสั่งที่อยู่ถัดไป

การใช้คำสั่ง Debounce นั้นจะใช้กับการตรวจสอบสถานะทาง Input ของพอร์ตที่ต่อใช้งานร่วมกับอุปกรณ์จำพวกที่เป็นหน้าสัมผัสต่างๆ เช่น สวิตช์ หรือ รีเลย์ เป็นต้น เนื่องจากเมื่อหน้าสัมผัสเริ่มต้นทำงานนั้น ในช่วงเวลาเริ่มต้นที่หน้าสัมผัสจะปิด (Short) หรือเปิด (Open) นั้นจะเกิดการ กระชาก ของสัญญาณ Input ซึ่งเรียกว่าอาการ Bounce โดยจะมีลักษณะเหมือนกับการเปิดและปิดของหน้าสัมผัสจำนวนหลายสิบหลายร้อยครั้งในช่วงระยะเวลาสั้นๆ แต่เนื่องจากการทำงานของ CPU นั้นสามารถกระทำได้อย่างรวดเร็วมาก (หลายล้านคำสั่งภายใน 1 วินาที) ดังนั้นจึงอาจทำให้โปรแกรมตรวจพบว่ามีการทำงานของหน้าสัมผัสสวิตช์ (มีการกดและปล่อยสวิตช์) หลายๆครั้ง ทั้งๆที่ในความเป็นจริงแล้วเกิดจากการทำงานของสวิตช์หรือหน้าสัมผัสเพียงครั้งเดียว ซึ่งการแก้ปัญหาคาร Bounce นี้ โดยเมื่อตรวจพบว่า Input ทำงานในครั้งแรกนั้นจะมีการหน่วงเวลาทิ้งไว้ชั่วขณะหนึ่งก่อนแล้วจึงทำการอ่านค่าสถานะของ Input เข้ามาตรวจสอบใหม่อีกครั้งหนึ่งเพื่อตรวจสอบให้แน่ใจ ซึ่งลักษณะของการหน่วงเวลาเพื่อรอตรวจสอบการทำงานของ Input ชั่วนี้เรียกว่าการ Debounce ซึ่งเป็นเทคนิคหนึ่งของการแก้ปัญหาคาร Bounce ของหน้าสัมผัส

## ตัวอย่างโปรแกรมการตรวจสอบการชน แบบตรวจสอบเป็น Bit

```

$regfile = "89c51rd.dat"      'กำหนดใช้งานกับ CPU เบอร์ P89C51RD2(Philips)
$ramstart = 0
$ramsize = 256
$crystal = 36864000          'กำหนด XTAL = 18.432MHz แบบ X2 Mode
Config Timer0=Timer,Gate=Internal,Mode=2 'ให้ Timer0 = 8Bit Auto Reload
Load Timer0 , 240             'กำหนดค่า Reload = 240x325.52ns = 78.125uS
Disable Timer0                'ห้ามไม่ให้เกิดการ Interrupt จากการทำงานของ Timer0
Start Timer0                  'เริ่มการทำงานของ Timer0 เพื่อ Trig PCA Counter
Cmod = &B00000100            'ให้ CPS(1:0) = 10 (PCA นับจาก Overflow ของ Timer0)
Ccapm1 = &B01000010          'Set Bit ECOM1,PWM1 (Enable PWM1)
Ccapm2 = &B01000010          'Set Bit ECOM2,PWM2 (Enable PWM2)
Ccap1h = 0                   'กำหนดค่า PWM1 Duty Cycle = 20mS
Ccap2h = 0                   'กำหนดค่า PWM2 Duty Cycle = 20mS
Ccon = &B01000000            'Set Bit CR (เริ่มการทำงานของระบบ PCA Timer)

Declare Sub Robot_forward     'โปรแกรมย่อยสำหรับบังคับให้รถเดินหน้า
Declare Sub Robot_backward    'โปรแกรมย่อยสำหรับบังคับให้รถถอยหลัง
Declare Sub Robot_fast_left   'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว
Declare Sub Robot_slow_left   'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ
Declare Sub Robot_fast_right  'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว
Declare Sub Robot_slow_right  'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ
Declare Sub Robot_stop        'โปรแกรมย่อยสำหรับบังคับให้รถหยุด

Sw_front_left Alias P0.0      'กำหนด Pin Port ที่ใช้ตรวจสอบการชน หน้า-ซ้าย
Sw_front_right Alias P0.3     'กำหนด Pin Port ที่ใช้ตรวจสอบการชน หน้า-ขวา
Sw_back_left Alias P0.1       'กำหนด Pin Port ที่ใช้ตรวจสอบการชน หลัง-ซ้าย
Sw_back_right Alias P0.4      'กำหนด Pin Port ที่ใช้ตรวจสอบการชน หลัง-ขวา

Robot_stop                   'สั่งให้รถหยุดอยู่กับที่ก่อน เพื่อรอการกดสวิตช์ Start
Bitwait P0.2 , Reset         'รอการกดสวิตช์ Start เพื่อเริ่มต้นการทำงาน
Robot_forward                'สั่งให้รถเดินหน้า

```

Do

Debounce Sw\_front\_left,0,Back\_turn\_right,Sub 'พบการชน Sw หน้า-ซ้าย

Debounce Sw\_front\_right,0,Back\_turn\_left,Sub 'พบการชน Sw หน้า-ขวา

Debounce Sw\_back\_left,0,Next\_turn\_right,Sub 'พบการชน Sw หลัง-ซ้าย

Debounce Sw\_back\_right,0,Next\_turn\_left,Sub 'พบการชน Sw หลัง-ขวา

Loop

**'การบริการเมื่อพบการชนของสวิทช์ หน้า-ซ้าย**

Back\_turn\_right: 'ให้ถอยหลังกลับ แล้วหลบโดยการเลี้ยวขวา เดินหน้าต่อไป

Robot\_backward 'สั่งให้รถถอยหลังกลับ

Wait 1 'หน่วงเวลาการถอยเพื่อให้พ้นระยะการเลี้ยวหลบ

Robot\_fast\_right 'สั่งเลี้ยวขวาเพื่อหลบสิ่งกีดขวาง

Wait 1 'หน่วงเวลาของการเลี้ยว

Robot\_forward 'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป

Return

**'การบริการเมื่อพบการชนของสวิทช์ หน้า-ขวา**

Back\_turn\_left: 'ให้ถอยหลังกลับ แล้วหลบโดยการเลี้ยวซ้าย เดินหน้าต่อไป

Robot\_backward 'สั่งให้รถถอยหลังกลับ

Wait 1 'หน่วงเวลาการถอยเพื่อให้พ้นระยะการเลี้ยวหลบ

Robot\_fast\_left 'สั่งเลี้ยวซ้ายเพื่อหลบสิ่งกีดขวาง

Wait 1 'หน่วงเวลาของการเลี้ยว

Robot\_forward 'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป

Return

**'การบริการเมื่อพบการชนของสวิทช์ หลัง-ซ้าย (ใช้สำหรับเมื่อถอยแล้วชน)**

Next\_turn\_right: 'ให้เดินหน้า แล้วหลบโดยเลี้ยวขวา เดินหน้าต่อไป

Robot\_forward 'สั่งให้รถเดินหน้า

Wait 1 'หน่วงเวลาการเดินหน้าเพื่อให้พ้นระยะการเลี้ยวหลบ

Robot\_fast\_right 'สั่งให้รถเลี้ยวหลบไปทางขวา

Wait 1 'หน่วงเวลาของการเลี้ยว

Robot\_forward 'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป

Return

**'การบริการเมื่อพบการชนของสวิทช์ หลัง-ขวา (ใช้สำหรับเมื่อถอยแล้วชน)**

```

Next_turn_left:      'ให้เดินหน้า แล้วหลบโดยเลี้ยวซ้าย เดินหน้าต่อไป
Robot_forward        'สั่งให้รถเดินหน้า
Wait 1               'หน่วงเวลาการเดินหน้าเพื่อให้พ้นระยะการเลี้ยวหลบ
Robot_fast_left      'สั่งให้รถเลี้ยวหลบไปทางขวา
Wait 1               'หน่วงเวลาของการเลี้ยว
Robot_forward        'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
Return

' /*****/;
' /* ROBOT Forward */;
' /*****/;
'
Sub Robot_forward      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเคลื่อนที่ไปข้างหน้า
  Ccap1h = 256 - 26    'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
  Ccap2h = 256 - 13    'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub                'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเคลื่อนที่ไปข้างหน้า

' /*****/;
' /* ROBOT Backward */;
' /*****/;
'
Sub Robot_backward     'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์ถอยหลัง
  Ccap1h = 256 - 13    'กำหนดให้ SERVO1(ซ้าย) = 1mS = ถอยหลัง
  Ccap2h = 256 - 26    'กำหนดให้ SERVO2(ขวา) = 2mS = ถอยหลัง
End Sub                'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์ถอยหลัง

' /*****/;
' /* ROBOT Turn Left (Fast) */;
' /*****/;
'
Sub Robot_fast_left    'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว
  Ccap1h = 256 - 13    'กำหนดให้ SERVO1(ซ้าย) = 1mS = ถอยหลัง
  Ccap2h = 256 - 13    'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub                'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว

```



```

' /***** */;
' /* ROBOT Turn Left (Slow) */;
' /***** */;
'

Sub Robot_slow_left      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ
    Ccap1h = 0            'กำหนดให้ SERVO1 (ซ้าย) = "1" = หยุดการเคลื่อนที่
    Ccap2h = 256 - 13     'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ

' /***** */;
' /* ROBOT Turn Right (Fast) */;
' /***** */;
'

Sub Robot_fast_right     'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว
    Ccap1h = 256 - 26     'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
    Ccap2h = 256 - 26     'กำหนดให้ SERVO2(ขวา) = 2mS = ถอยหลัง
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว

' /***** */;
' /* ROBOT Turn Right (Slow) */;
' /***** */;
'

Sub Robot_slow_right     'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ
    Ccap1h = 256 - 26     'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
    Ccap2h = 0            'กำหนดให้ SERVO2 (ขวา) = "1" = หยุดการเคลื่อนที่
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ

' /***** */;
' /* ROBOT Stop */;
' /***** */;
'

Sub Robot_stop           'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์หยุด
    Ccap1h = 0            'กำหนดให้ SERVO1 (ซ้าย) = "1" = หยุดการเคลื่อนที่
    Ccap2h = 0            'กำหนดให้ SERVO2 (ขวา) = "1" = หยุดการเคลื่อนที่
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์หยุด

```

แสดง ตัวอย่างโปรแกรมตรวจสอบการชนโดยการตรวจสอบ Input แบบ Bit

สำหรับการทำงานของโปรแกรมนั้น จะขอก้าว อธิบาย เฉพาะส่วนของการตรวจสอบการทำงานของสวิทช์เท่านั้น สำหรับการทำงานของโปรแกรมนั้นส่วนของการควบคุมการเคลื่อนที่ของตัวรถนั้น ขอให้ศึกษารายละเอียดจากตัวอย่างในหัวข้อ “การควบคุมการเคลื่อนที่ของรถหุ่นยนต์” เอง

โดยการทำงานของโปรแกรมนั้นส่วนของการตรวจสอบการชนนั้น จะอยู่ภายในส่วนของวงรอบคำสั่ง DO..LOOP โดยจะใช้คำสั่ง Debounce สำหรับตรวจสอบการทำงานของสวิทช์ ซึ่งก่อนหน้าที่จะเข้าสู่วงรอบการทำงานของคำสั่ง DO..LOOP นั้นโปรแกรมนั้นจะรอการกดสวิทช์ Start ก่อน ซึ่งเมื่อตรวจพบว่าสวิทช์ Start ถูกกดแล้วก็จะสั่งให้รถหุ่นยนต์เริ่มต้นเคลื่อนที่ไปข้างหน้าแล้วจึงเข้าทำงานในวงรอบ DO..LOOP

โดยภายในวงรอบคำสั่งของ DO..LOOP นั้นจะใช้คำสั่ง Debounce เพื่อตรวจสอบการทำงานของสวิทช์ ทั้ง 4 ตำแหน่ง โดยการทำงานของคำสั่งนี้เมื่อตรวจพบการทำงานของสัญญาณตรงตามเงื่อนไขที่กำหนดไว้แล้ว จะทำการหน่วงเวลาไว้ชั่วขณะหนึ่ง (ประมาณ 25mS) แล้วจึงตรวจสอบซ้ำว่าเงื่อนไขการทำงานของสัญญาณยังตรงตามที่กำหนดไว้หรือไม่ โดยเมื่อตรวจพบว่าสวิทช์ตัวใดทำงานก็จะกระโดดเข้าไปทำงานยังโปรแกรมน้อยที่สร้างขึ้นมารองรับการทำงานเมื่อตรวจพบว่า Switch นั้นๆถูกกด โดยเงื่อนไขการทำงานของโปรแกรมน้อยทั้ง 4 โปรแกรมมีดังนี้

- ถ้าตรวจพบว่า Switch กันชน ตำแหน่ง หน้า-ซ้าย ถูกกด จะสั่งให้รถถอยหลังกลับออกมาระยะเวลาหนึ่งเพื่อรอให้พ้นระยะสิ่งกีดขวาง จากนั้นจึงสั่งให้รถเลี้ยวหลบไปทางขวาพร้อมกับหน่วงเวลาของการเลี้ยวไว้ระยะหนึ่งเพื่อให้ทิศทางการเคลื่อนที่ของรถพ้นจากแนวสิ่งกีดขวางแล้วจึงสั่งให้รถเคลื่อนที่เดินทางต่อไปตามปกติ
- ถ้าตรวจพบว่า Switch กันชน ตำแหน่ง หน้า-ขวา ถูกกด จะสั่งให้รถถอยหลังกลับออกมาระยะเวลาหนึ่งเพื่อรอให้พ้นระยะสิ่งกีดขวาง จากนั้นจึงสั่งให้รถเลี้ยวหลบไปทางซ้ายพร้อมกับหน่วงเวลาของการเลี้ยวไว้ระยะหนึ่งเพื่อให้ทิศทางการเคลื่อนที่ของรถพ้นจากแนวสิ่งกีดขวางแล้วจึงสั่งให้รถเคลื่อนที่เดินทางต่อไปตามปกติ
- ถ้าตรวจพบว่า Switch กันชน ตำแหน่ง หลัง-ซ้าย ถูกกด จะสั่งให้เดินทางไประยะหนึ่ง จากนั้นจึงสั่งให้รถเลี้ยวหลบไปทางขวาพร้อมกับหน่วงเวลาของการเลี้ยวไว้ระยะหนึ่งก่อนที่จะสั่งให้รถเคลื่อนที่เดินทางต่อไปตามปกติ
- ถ้าตรวจพบว่า Switch กันชน ตำแหน่ง หลัง-ขวา ถูกกด จะสั่งให้เดินทางไประยะหนึ่ง จากนั้นจึงสั่งให้รถเลี้ยวหลบไปทางซ้ายพร้อมกับหน่วงเวลาของการเลี้ยวไว้ระยะหนึ่งก่อนที่จะสั่งให้รถเคลื่อนที่เดินทางต่อไปตามปกติ

## ตัวอย่างโปรแกรมการตรวจสอบการชน แบบตรวจสอบเป็น Byte

```

$regfile = "89c51rd.dat"      'กำหนดใช้งานกับ CPU เบอร์ P89C51RD2(Philips)
$ramstart = 0
$ramsize = 256
$crystal = 36864000          'กำหนด XTAL = 18.432MHz แบบ X2 Mode
Config Timer0=Timer,Gate=Internal,Mode=2 'ให้ Timer0 = 8Bit Auto Reload
Load Timer0 , 240             'กำหนดค่า Reload = 240x325.52ns = 78.125uS
Disable Timer0                'ห้ามไม่ให้เกิดการ Interrupt จากการทำงานของ Timer0
Start Timer0                  'เริ่มการทำงานของ Timer0 เพื่อ Trig PCA Counter
Cmod = &B00000100            'ให้ CPS(1:0) = 10 (PCA นับจาก Overflow ของ Timer0)
Ccapm1 = &B01000010          'Set Bit ECOM1,PWM1 (Enable PWM1)
Ccapm2 = &B01000010          'Set Bit ECOM2,PWM2 (Enable PWM2)
Ccap1h = 0                    'กำหนดค่า PWM1 Duty Cycle = 20mS
Ccap2h = 0                    'กำหนดค่า PWM2 Duty Cycle = 20mS
Ccon = &B01000000            'Set Bit CR (เริ่มการทำงานของระบบ PCA Timer)

Declare Sub Robot_forward      'โปรแกรมย่อยสำหรับบังคับให้รถเดินหน้า
Declare Sub Robot_backward     'โปรแกรมย่อยสำหรับบังคับให้รถถอยหลัง
Declare Sub Robot_fast_left    'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว
Declare Sub Robot_slow_left    'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ
Declare Sub Robot_fast_right   'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว
Declare Sub Robot_slow_right   'โปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ
Declare Sub Robot_stop         'โปรแกรมย่อยสำหรับบังคับให้รถหยุด
Declare Sub Back_turn_right     'โปรแกรมย่อยบริการเมื่อพบการชน หน้า-ซ้าย
Declare Sub Back_turn_left     'โปรแกรมย่อยบริการเมื่อพบการชน หน้า-ขวา
Declare Sub Back_eturn         'โปรแกรมย่อยบริการเมื่อพบการชน หน้า-ซ้าย,ขวา
Declare Sub Next_turn_right    'โปรแกรมย่อยบริการเมื่อพบการชน หลัง-ซ้าย
Declare Sub Next_turn_left     'โปรแกรมย่อยบริการเมื่อพบการชน หลัง-ขวา

Dim Input_new As Byte          'ประกาศตัวแปรสำหรับเก็บค่าสถานะ Input ปัจจุบัน
Dim Input_old As Byte          'ประกาศตัวแปรสำหรับเก็บค่าสถานะ Input ครั้งก่อน
Port_input Alias P0            'กำหนดพอร์ต Input

```

## ตัวอย่างการควบคุมรถหุ่นยนต์ ET-ROBOT RD2 ด้วยภาษาเบสิก BASCOM-8051

```

Robot_stop      'สั่งให้รถหยุดอยู่กับที่ก่อน เพื่อรอการกดสวิตช์ Start
Bitwait P0.2 , Reset      'รอการกดสวิตช์ Start เพื่อเริ่มต้นการทำงาน
Robot_forward    'สั่งให้รถเดินหน้า
Input_old = &B00011011    'กำหนดค่าเริ่มต้นของค่าอ้างอิง Input Bit 0,1,3,4

Do
  Input_new = Port_input    'สั่งอ่านค่าสถานะ Input จากพอร์ต P0
  Input_new = Input_new And &B00011011    'ตรวจสอบเฉพาะ Bit 0,1,3,4
  If Input_new <> Input_old Then    'ถ้าสถานะไม่เหมือนเดิมให้ทำดังนี้
    Waitms 30    'หน่วงเวลาเพื่อ Debounce
    Input_new = Port_input    'สั่งอ่านสถานะ Input ใหม่อีกครั้งหนึ่ง
    Input_new = Input_new And &B00011011    'ตรวจสอบเฉพาะ Bit 0,1,3,4
    If Input_new <> Input_old Then    'ถ้ายังไม่เหมือนเดิมอีกให้ทำดังนี้
      Input_old = Input_new    'Update Input_old เพื่อใช้อ้างอิงครั้งต่อไป
      Select Case Input_new    'เริ่มตรวจสอบสถานะ Input ที่เปลี่ยนแปลง
        Case &B00010010 : Back_urn    'กรณีพบ Sw หน้า ซ้ายและขวา
        Case &B00011010 : Back_turn_right    'กรณีพบ Sw หน้า-ซ้าย
        Case &B00010011 : Back_turn_left    'กรณีพบ Sw หน้า-ขวา
        Case &B00011001 : Next_turn_right    'กรณีพบ Sw หลัง-ซ้าย
        Case &B00001011 : Next_turn_left    'กรณีพบ Sw หลัง-ขวา
        Case Else Robot_forward    'กรณีอื่นๆให้สั่งเดินหน้า
      End Select
    End If
  End If
Loop

' /***** */;
' /* ROBOT Forward */;
' /***** */;
'

Sub Robot_forward    'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเคลื่อนที่ไปข้างหน้า
  Ccap1h = 256 - 26    'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
  Ccap2h = 256 - 13    'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub    'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเคลื่อนที่ไปข้างหน้า

```

```

' /*****/;
' /* ROBOT Backward */;
' /*****/;
'
Sub Robot_backward      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์ถอยหลัง
    Ccap1h = 256 - 13    'กำหนดให้ SERVO1(ซ้าย) = 1mS = ถอยหลัง
    Ccap2h = 256 - 26    'กำหนดให้ SERVO2(ขวา) = 2mS = ถอยหลัง
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์ถอยหลัง

' /*****/;
' /* ROBOT Turn Left (Fast) */;
' /*****/;
'
Sub Robot_fast_left      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว
    Ccap1h = 256 - 13    'กำหนดให้ SERVO1(ซ้าย) = 1mS = ถอยหลัง
    Ccap2h = 256 - 13    'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายอย่างรวดเร็ว

' /*****/;
' /* ROBOT Turn Left (Slow) */;
' /*****/;
'
Sub Robot_slow_left      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ
    Ccap1h = 0           'กำหนดให้ SERVO1 (ซ้าย) = "1" = หยุดการเคลื่อนที่
    Ccap2h = 256 - 13    'กำหนดให้ SERVO2(ขวา) = 1mS = เคลื่อนที่ไปข้างหน้า
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวซ้ายแบบปกติ

' /*****/;
' /* ROBOT Turn Right (Fast) */;
' /*****/;
'
Sub Robot_fast_right     'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว
    Ccap1h = 256 - 26    'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
    Ccap2h = 256 - 26    'กำหนดให้ SERVO2(ขวา) = 2mS = ถอยหลัง
End Sub                  'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาอย่างรวดเร็ว

```

```

' /***** */;
' /* ROBOT Turn Right (Slow) */;
' /***** */;
'

Sub Robot_slow_right      'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ
    Ccap1h = 256 - 26      'กำหนดให้ SERVO1(ซ้าย) = 2mS = เคลื่อนที่ไปข้างหน้า
    Ccap2h = 0             'กำหนดให้ SERVO2 (ขวา) = "1" = หยุดการเคลื่อนที่
End Sub                   'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถเลี้ยวขวาแบบปกติ

' /***** */;
' /* ROBOT Stop */;
' /***** */;
'

Sub Robot_stop            'จุดเริ่มต้นโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์หยุด
    Ccap1h = 0            'กำหนดให้ SERVO1 (ซ้าย) = "1" = หยุดการเคลื่อนที่
    Ccap2h = 0            'กำหนดให้ SERVO2 (ขวา) = "1" = หยุดการเคลื่อนที่
End Sub                   'จุดสิ้นสุดโปรแกรมย่อยสำหรับบังคับให้รถหุ่นยนต์หยุด

'การบริการเมื่อพบการชนของสวิตช์ หน้า-ซ้าย
Back_turn_right:         'ให้ถอยหลังกลับ แล้วหลบโดยการเลี้ยวขวา เดินหน้าต่อไป
    Robot_backward        'สั่งให้รถถอยหลังกลับ
    Wait 1                'หน่วงเวลาการถอยเพื่อให้พ้นระยะการเลี้ยวหลบ
    Robot_fast_right      'สั่งเลี้ยวขวาเพื่อหลบสิ่งกีดขวาง
    Wait 1                'หน่วงเวลาของการเลี้ยว
    Robot_forward         'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
Return

'การบริการเมื่อพบการชนของสวิตช์ หน้า-ขวา
Back_turn_left:         'ให้ถอยหลังกลับ แล้วหลบโดยการเลี้ยวซ้าย เดินหน้าต่อไป
    Robot_backward        'สั่งให้รถถอยหลังกลับ
    Wait 1                'หน่วงเวลาการถอยเพื่อให้พ้นระยะการเลี้ยวหลบ
    Robot_fast_left      'สั่งเลี้ยวซ้ายเพื่อหลบสิ่งกีดขวาง
    Wait 1                'หน่วงเวลาของการเลี้ยว
    Robot_forward         'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
Return

```

**'การบริการเมื่อพบการชนของสวิทช์ หน้า ขวา และ ซ้าย พร้อม ๆ กัน**

```

Sub Back_urn          'ให้ถอยหลังแล้วเลี้ยวกลับแบบ U-Turn
  Robot_backward      'สั่งให้รถถอยหลังกลับ
  Wait 1              'หน่วงเวลาการถอยเพื่อให้พ้นระยะการเลี้ยวหลบ
  Robot_fast_left     'สั่งเลี้ยวขวาเพื่อ U-Turn กลับ
  Wait 2              'หน่วงเวลาของการเลี้ยว
  Robot_forward       'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
End Sub

```

**'การบริการเมื่อพบการชนของสวิทช์ หลัง-ซ้าย (ใช้สำหรับเมื่อถอยแล้วชน)**

```

Next_turn_right:      'ให้เดินหน้า แล้วหลบโดยเลี้ยวขวา เดินหน้าต่อไป
  Robot_forward        'สั่งให้รถเดินหน้า
  Wait 1              'หน่วงเวลาการเดินหน้าเพื่อให้พ้นระยะการเลี้ยวหลบ
  Robot_fast_right     'สั่งให้รถเลี้ยวหลบไปทางขวา
  Wait 1              'หน่วงเวลาของการเลี้ยว
  Robot_forward       'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
Return

```

**'การบริการเมื่อพบการชนของสวิทช์ หลัง-ขวา (ใช้สำหรับเมื่อถอยแล้วชน)**

```

Next_turn_left:       'ให้เดินหน้า แล้วหลบโดยเลี้ยวซ้าย เดินหน้าต่อไป
  Robot_forward        'สั่งให้รถเดินหน้า
  Wait 1              'หน่วงเวลาการเดินหน้าเพื่อให้พ้นระยะการเลี้ยวหลบ
  Robot_fast_left     'สั่งให้รถเลี้ยวหลบไปทางขวา
  Wait 1              'หน่วงเวลาของการเลี้ยว
  Robot_forward       'สั่งให้รถเคลื่อนที่เดินหน้าต่อไป
Return

```

**แสดง ตัวอย่างโปรแกรมตรวจสอบการชน โดยการตรวจสอบ Input แบบ Byte**

สำหรับตัวอย่างโปรแกรมนี้นี้ จะใช้วิธีการตรวจสอบสถานะการทำงานของ Switch ตรวจสอบการชนโดยการอ่าน Input เข้ามาทั้งพอร์ต ซึ่งวิธีการนี้จะมีข้อดีกว่าวิธีการตรวจสอบ Input แบบ Bit เนื่องจากสามารถตรวจสอบลักษณะของ Input ซึ่งทำงานได้พร้อมๆกันในเวลาเดียวกันได้มากกว่า 1 Input แต่จะต้องทำการหน่วงเวลาเพื่อทำการ Debounce หน้าสัมผัสสวิทช์เอง

สำหรับการทำงานของโปรแกรมนั้นจะอาศัยหลักการอ่านค่าสถานะ Input จากพอร์ต P0 เพื่อนำมาตรวจสอบสถานะ Input ของบิตที่ต่อกับ Micro Switch สำหรับตรวจสอบการชน แต่การอ่านนั้นจะอ่านค่าได้มาทั้ง 8 บิตของพอร์ต P0 ดังนั้นจึงต้องทำการ AND ด้วยค่า 00011011B เพื่อให้ข้อมูลบิตอื่นๆที่ไม่เกี่ยวข้องมีค่าเป็น "0" เพื่อให้เหลือเฉพาะสถานะของ Micro Switch ที่ต้องการคือ บิต 0,1,3 และ 4 เท่านั้น จากนั้นก็จะทำการเปรียบเทียบค่า สถานะของ Input ที่อ่านได้ในปัจจุบัน กับค่าสถานะ Input ครั้งสุดท้ายที่อ่านได้มายังมีค่าเหมือนกันอยู่หรือไม่ ซึ่งถ้าค่า Input ยังเหมือนเดิมก็แสดงว่ายังไม่มีมีการเปลี่ยนแปลงของ Input ก็จะไม่สนใจ แต่ถ้าพบว่า Input เกิดการเปลี่ยนแปลง ในครั้งแรกโปรแกรมจะยังไม่ตัดสินใจในทันที แต่จะทำการสั่งหน่วงเวลาทิ้งไประยะหนึ่งก่อน จากนั้นจึงทำการอ่านค่าสถานะ Input กลับเข้ามาตรวจสอบใหม่อีกครั้งหนึ่งเพื่อให้แน่ใจว่าเกิดการเปลี่ยนแปลงจริง ซึ่งถ้าพบว่า Input เกิดการเปลี่ยนแปลงจริงๆก็จะทำการตรวจสอบลักษณะการทำงานของ Micro Switch โดยใช้คำสั่ง Select Case ในการเลือกตรวจสอบเงื่อนไขและสั่งงาน ซึ่งจากตัวอย่างจะสั่งตรวจสอบอยู่ 6 เงื่อนไขคือ

- ถ้าพบว่า Micro Switch ด้านหน้า ฝั่งซ้าย ทำงาน (Bit 0 และ 3 เป็น "0") พร้อมกัน ก็ จะสั่งให้รถถอยหลังแล้ว U-Turn กลับ
- ถ้าพบว่า Micro Switch ด้านหน้า ฝั่งซ้าย ทำงาน (Bit 0 = "0") จะสั่งให้รถถอยหลังแล้วเลี้ยว หลบไปทางขวา
- ถ้าพบว่า Micro Switch ด้านหน้า ฝั่งขวา ทำงาน (Bit 3 = "0") จะสั่งให้รถถอยหลังแล้วเลี้ยว หลบไปทางซ้าย
- ถ้าพบว่า Micro Switch ด้านหลัง ฝั่งซ้าย ทำงาน (Bit 1 = "0") จะสั่งให้รถเดินหน้าพร้อมกับ เลี้ยวหลบไปทางขวา
- ถ้าพบว่า Micro Switch ด้านหลัง ฝั่งขวา ทำงาน (Bit 4 = "0") จะสั่งให้รถเดินหน้าพร้อมกับ เลี้ยวหลบไปทางซ้าย
- ถ้าพบว่า Micro Switch มีสถานะนอกเหนือจาก 5 สถานะ ที่กล่าวมาแล้วในข้างต้น จะสั่งให้ รถเดินหน้าต่อไปตามปกติ

ซึ่งในการนำไปใช้งานจริงนั้น อาจจำเป็นต้องพิจารณาถึงลักษณะ ของสถานะการชน และ สิ่งแวดล้อมต่างๆประกอบด้วย โดยอาจต้องมีการปรับปรุงเงื่อนไขของการของโปรแกรมย่อยต่างๆ ที่ใช้สำหรับรองรับการทำงานของโปรแกรมเมื่อตรวจพบการทำงานของ Micro Switch แบบต่างๆให้เหมาะสมด้วย เช่น จากตัวอย่างเมื่อเกิดการชนหน้าหน้าฝั่งซ้ายจะสั่งให้เลี้ยวหลบไปทางขวา และถ้าเกิดการชนด้านขวาจะสั่งให้เลี้ยวหลบไปทางซ้าย ซึ่งถ้านำรถหุ่นยนต์ไปทดสอบกับมุมห้องที่เป็นสี่เหลี่ยมมุมฉากก็จะพบว่ารถจะ หลบจากซ้ายไปชนด้านขวาแล้วหลบจากด้านขวากลับมาชนทางซ้ายอีกสลับกันไปมาจนไม่สามารถหลบ ออกจากมุมห้องได้ ซึ่งอาจต้องปรับเปลี่ยนรูปแบบการตอบสนองการชนใหม่ให้เหมาะสมด้วย