

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ความรู้ทั่วไปเกี่ยวกับสัญญาณไฟจราจร

สัญญาณไฟจราจร (Traffic Signalization) หมายถึง สัญญาณไฟสำหรับให้ผู้ใช้ทางปฏิบัติตามสัญญาณนั้น อาจเป็นสัญญาณไฟทางแยก หรือสัญญาณไฟทางข้ามหรือสัญญาณไฟกระพริบ หรือสัญญาณไฟควบคุมช่องเดินรถ หรือสัญญาณไฟเตือนอื่นๆ (ตามข้อกำหนดกรมตำรวจ เรื่องสัญญาณจราจร เครื่องหมายจราจร และความหมายของสัญญาณจราจรและเครื่องหมายจราจร ข้อ 3 เมื่อวันที่ 24 เมษายน 2522 ออกตาม พ.ร.บ.จราจรทางบกพ.ศ.2522 “สัญญาณไฟจราจร หมายถึง โคมสัญญาณไฟที่ใช้ควบคุมการจราจรมีขนาดและติดตั้งหรือทำให้ปรากฏไว้ในทาง ในลักษณะที่ทำให้ผู้ขับขี่หรือผู้ที่ต้องปฏิบัติตามสัญญาณมองเห็นได้โดยชัดเจน การติดตั้งโคมสัญญาณไฟจราจรต้องประกอบด้วยดวงโคมอย่างน้อยสามดวง โดยมีโคมสัญญาณไฟจราจรสีแดงอยู่ตอนบนหรือด้านขวาของผู้ขับขี่หรือผู้ที่ต้องปฏิบัติตามสัญญาณโคมสัญญาณไฟจราจรสีเหลืองอำพันอยู่ตอนกลางและโคมสัญญาณไฟจราจรสีเขียวอยู่ตอนล่างหรือด้านซ้ายมือของผู้ขับขี่ หรือผู้ที่ต้องปฏิบัติตามสัญญาณ ในบางกรณีอาจมีโคมสัญญาณไฟจราจรลูกศรเขียวประกอบได้”)

สัญญาณไฟจราจร

สัญญาณไฟจราจรประกอบด้วย 3 สี คือ

- สีเหลืองอำพัน (Amber) หมายถึง การเตือนให้ผู้ขับขี่ชะลอความเร็วและเพิ่มความระมัดระวังให้มากยิ่งขึ้นในบางครั้งให้หยุดเมื่อเห็นว่าจะไม่ปลอดภัย
- สีแดง (Red) หมายถึง การบังคับให้ยานพาหนะทุกคันหยุด
- สีเขียว (Green) หมายถึง การอนุญาตให้ยานพาหนะแล่นผ่านไป

จุดประสงค์ในการติดตั้งสัญญาณไฟจราจร

การติดตั้งสัญญาณไฟจราจรควบคุมที่ทางแยกจะช่วยลดปัญหาการจราจรติดขัดและทำให้การใช้รถใช้ถนนให้ปลอดภัยมากยิ่งขึ้น และเป็นแนวทางที่สำคัญอันหนึ่งที่ใช้ควบคุมการใช้รถใช้ถนน ในโครงข่ายจุดประสงค์หลักในการติดตั้งสัญญาณไฟจราจรอาจจะเป็นข้อหนึ่งข้อใดหรือหลายข้อดังต่อไปนี้

- ลดความขัดแย้งในการจราจร (TRAFFIC CONFLICTS)
- ลดความล่าช้าการจราจร
- เพิ่มความจุถนน
- ลดอุบัติเหตุ
- ลดกำลังเจ้าหน้าที่ตำรวจ
- อำนวยความสะดวกในการข้ามถนนของคนเดินเท้าและรถจักรยาน
- ให้สิทธิพิเศษแก่ยานพาหนะบางประเภทในบางโอกาส เช่น รถโดยสารสาธารณะรถจักรยาน

- ส่งเสริมให้มีการใช้ถนนตรงตามชนิดของถนนที่กำหนดไว้ เช่น ลดจุดขัดแย้งและความล่าช้าบนถนนบางสาย เพื่อดึงดูดให้ผู้ขับขี่หันมาใช้ถนนนั้น เพื่อลดจำนวนการขับขี่ตามซอยต่าง ๆ ที่ผ่านพื้นที่อยู่อาศัย
- ลดความยากลำบาก และความกดดันในการขับขี่ผ่านทางแยกที่มีการจราจรหนาแน่น

ข้อดีของการติดตั้งสัญญาณไฟจราจร

- การจราจรเป็นไปอย่างมีระเบียบ ลดปัญหาการติดขัด
- ลดอุบัติเหตุบางชนิด เช่น การประสานงาน อุบัติเหตุของคนเดินถนน
- ให้ความปลอดภัยแก่รถทางโทในการผ่านหรือเข้าสู่ทางเอก
- เสริมความมั่นใจให้กับผู้ใช้ยานพาหนะ

ข้อเสียของการติดตั้งสัญญาณไฟจราจร

- ทำให้เกิดความล่าช้าบริเวณทางแยก โดยเฉพาะนอกเหนือเวลาเร่งด่วน
- เพิ่มอุบัติเหตุบางประเภท เช่น ชนด้านหลังของรถยนต์
- การติดตั้งสัญญาณไฟในบริเวณไม่เหมาะสมอาจเพิ่มความล่าช้าโดยไม่จำเป็น
- การติดตั้งสัญญาณไฟที่ระยะเวลาไม่เหมาะสมอาจสร้างความล่าช้าและความเบื่อหน่าย

การไหลอิ่มตัว (Saturation Flow)

การไหลอิ่มตัว หมายถึง ปริมาณการจราจรสูงสุดที่ผ่านเส้นหยุดที่ทางแยกไปได้ในช่วงสัญญาณไฟเขียวมีหน่วยนับเป็นปริมาณยานพาหนะต่อชั่วโมงหรือหน่วยยานพาหนะเทียบรถยนต์ส่วนบุคคล เริ่มจากการเคลื่อนที่ การเร่งความเร็วเพื่อให้ความเร็วเพิ่มขึ้นคงที่ ปริมาณการจราจรที่สามารถผ่านทางแยกส่วนหนึ่งจะเสียไปจากประสิทธิภาพของทางแยก

ปัจจัยที่ใช้พิจารณากำหนดให้มีสัญญาณไฟจราจร

1. ความเร็วของการจราจรที่คาดหวัง ปริมาณการจราจร และจำนวนคนเดินเท้าในช่วงเวลา 16 ชั่วโมง นอกจากนี้จะต้องพิจารณาถึงสัดส่วนของยานพาหนะในช่วงเวลาเร่งด่วน ความสมดุลระหว่างปริมาณการจราจรในถนนสายหลักกับปริมาณจราจรในถนนสายรอง และจุดขัดแย้ง (CONFLICT) ระหว่างรถเลี้ยวกับคนข้ามถนน
2. สถิติของอุบัติเหตุ (จำนวน อัตรา และคุณลักษณะ) และความสัมพันธ์ของอุบัติเหตุกับรูปแบบของพื้นที่บริเวณนั้น
3. ความเป็นไปได้ในการจัดรูปแบบทางแยกในลักษณะต่างๆ และแผนผังของทางแยก
4. ทางแยกนั้นอยู่ในพื้นที่การควบคุมสัญญาณไฟเป็นโครงข่าย UPC หรือไม่ และอยู่ใกล้กับทางแยกถัดไปมากน้อยเพียงใด

การควบคุมสัญญาณไฟ

การควบคุมสัญญาณไฟ มี 2 ประเภท คือ

การควบคุมสัญญาณไฟแบบเดี่ยว (Isolated Signal Control) ทางแยกที่มีสัญญาณไฟจราจรควบคุมสามารถที่จะทำงานได้เดี่ยวๆ ไม่ว่าจะเป็นการทำงานแบบกำหนดช่วงเวลา ควบคุมสัญญาณไฟจราจร สามารถที่จะทำงานได้เดี่ยวๆ ไม่ว่าจะเป็นการทำงานแบบกำหนดช่วงเวลาสัญญาณไฟล่วงหน้า (Fixed Time Operation) หรือจัดสัญญาณไฟตามสภาพจริง การควบคุมในลักษณะนี้ จะใช้ในกรณีที่ว่าทางแยกนั้นเป็นทางแยกที่มีสัญญาณไฟจราจรอยู่แห่งเดียวในบริเวณนั้นหรือทางแยกอื่นๆ อยู่ห่างออกไปมากเกินกว่า 500 เมตร เป็นต้น

การควบคุมสัญญาณไฟแบบต่อเนื่อง (Linked Signal Control) บริเวณในเมืองไม่ควรใช้การควบคุมสัญญาณไฟแบบเดี่ยว ดังนั้นควรพิจารณาออกแบบสัญญาณไฟแบบต่อเนื่อง ซึ่งจะให้ประสิทธิภาพสูงสุด สัญญาณไฟแบบนี้จะเชื่อมต่อกันกับทางแยกใกล้เคียงภายใต้ระบบควบคุมสัญญาณไฟแบบโครงข่าย UPC (Urban Traffic Control)

การใช้งานอื่น ๆ ของสัญญาณไฟจราจร

สัญญาณไฟกระพริบที่เรียกว่า “WIG-WAG” ใช้ในการหยุดการจราจรในบริเวณต่าง ๆ เช่นทางรถไฟ บริเวณใกล้กับสนามบิน สถานีดับเพลิง และสถานีฉุกเฉินต่างๆ สัญญาณไฟนี้จะประกอบด้วยไฟแดง 2 ดวง ที่อยู่ในแนวขนานกันและกระพริบไป-มา มีไฟเหลือง 1 ดวง อยู่ตรงกลางระหว่างไฟแดงทั้งสองขณะที่เริ่มทำงานจะเปิดไฟเหลืองก่อนที่จะมีการเปิดไฟกระพริบสีแดง โดยทั่วไปสัญญาณนี้จะปิดไว้ตลอดจนกว่าจะมีการสั่งให้ทำงานโดยมือหรือโดยอุปกรณ์อัตโนมัติได้

ในถนนที่มีความเร็วสูงการควบคุมช่องทางจราจรแบบพิเศษและสัญญาณบอกเหตุอื่นๆ สามารถใช้เพื่อหยุดการจราจรเมื่อมีเหตุการณ์บางอย่างเกิดขึ้นและอาจจะบังคับให้เปลี่ยนช่องทาง ,ลดความเร็ว, เตือนอันตรายหรือบังคับให้ออกที่ทางออกข้างหน้า

สัญญาณอาจจะวางไว้ที่ข้างถนนหรือวางบนทั้งบนช่องทางที่ต้องการ ส่วนมากการดำเนินการแบบนี้จะใช้กับแผนจัดการจราจรตามปริมาณการจราจรซึ่งทิศทางการจราจรจะเปลี่ยนไปตามชั่วโมงเร่งด่วน การเปลี่ยนทิศทางนี้จะใช้สัญญาณไฟที่ติดตั้งไว้กับโครงเหล็กที่ติดตั้งไว้เหนือช่องทางจราจรเป็นตัวบ่งบอก ในงานก่อสร้างหรือบำรุงรักษาดถนนอาจจะเป็นที่จะต้องใช้สัญญาณไฟจราจรชั่วคราว (สามารถเคลื่อนย้ายได้) เพื่อช่วยจัดการจราจร เช่น จัดเดินรถทางเดียว เป็นต้น

หลักในการควบคุมสัญญาณไฟจราจร

1. วงรอบเวลาในการปิด-เปิดสัญญาณไฟจราจร แต่ละรอบต้องไม่นานเกินไป โดยแต่ละรอบไม่ควรเกิน 4 นาที โดยประมาณ
2. การปิด-เปิดสัญญาณไฟจราจร แต่ละด้านต้องไม่นานเกินไป แต่ละด้านไม่ควรเกิน 1 นาที
3. การปล่อยแต่ละด้านไม่ควรข้ามจังหวะการปล่อยปกติ
4. การควบคุมการปิด-เปิด ต้องประสานสัมพันธ์กับแยกใกล้เคียงทุกด้าน

5. ควบคุมทิศทางการจราจรที่เหมาะสม เช่น เร่งระบายรถด้านเข้าเมืองในช่วงเช้าในช่วงเช้าในชั่วโมงเร่งด่วน และ ระบายรถออกเมืองในช่วงเย็น

การจราจร เป็นระบบที่ค่อนข้างซับซ้อน มีองค์ประกอบหลักอยู่ 4 ส่วน ได้แก่ ผู้ใช้รถใช้ถนน (Drivers) ยานพาหนะ (Vehicles) ถนนหนทาง (Roadways) และอุปกรณ์ควบคุมการจราจร (Traffic Controls) ซึ่งปัจจัยทั้งสี่มีความเกี่ยวเนื่องสัมพันธ์กันจนก่อให้เกิดเป็นสภาพการจราจร จึงจำเป็นต้องทำความเข้าใจในองค์ประกอบทั้งสี่อย่างลึกซึ้งเพื่อสามารถนำความรู้ดังกล่าวไปใช้ในการออกแบบ วางแผน จัดการ ควบคุมระบบการจราจรได้อย่างเหมาะสมต่อไป

2.2 คุณลักษณะของผู้ใช้รถใช้ถนน

ผู้ใช้รถใช้ถนน คือ คนที่เดินทางอยู่บนท้องถนนทุกคน ไม่ว่าจะเป็นผู้ขับขี่ยานพาหนะ (รถยนต์ รถมอเตอร์ไซค์ รถเมล์) ผู้โดยสาร ผู้ขับขี่จักรยาน คนเดินเท้า และคนพิการ ผู้ใช้รถใช้ถนนนี้เป็นองค์ประกอบของระบบจราจรที่มีความซับซ้อนมากที่สุด เพราะคนแต่ละคนมีความรู้ความนึกคิด ความรู้ ความพร้อม สภาพร่างกายและอารมณ์แตกต่างกันไปในแต่ละบุคคลและยังแตกต่างกันไปในแต่ละช่วงเวลาด้วย ดังนั้นการออกแบบอุปกรณ์ควบคุมสภาพการจราจรต่างๆ ให้ใช้งานได้อย่างเหมาะสมกับผู้ใช้อใช้ถนนทุกคนได้อย่างมีประสิทธิภาพและปลอดภัย ซึ่งต้องคำนึงถึงบุคคลกลุ่มต่างๆ ไม่ว่าจะเป็นผู้ใหญ่ เด็ก คนชรา คนพิการ และผู้ใช้รถใช้ถนนอื่นๆ ด้วย

คุณลักษณะของผู้ใช้รถใช้ถนนที่สำคัญบางตัวถูกนำมาใช้ในการวิเคราะห์วิศวกรรมจราจร ได้แก่ ระยะเวลาการรับรู้และตอบสนอง (Perception and reaction time) และ visual acuity ซึ่งคุณลักษณะเหล่านี้สามารถทำการตรวจวัดได้ไม่ยากนักสำหรับคุณลักษณะอื่นๆ เช่น ความสมบูรณ์แข็งแรงของร่างกาย การได้ยิน ความเหนื่อยล้า และสภาพจิตใจ ก็มีผลต่อการตัดสินใจของผู้ใช้รถใช้ถนนเช่นกัน แต่ปัจจัยเหล่านี้ทำการตรวจวัดได้ยากจึงไม่ค่อยนิยมนำมาใช้ในการวิเคราะห์มากนัก

2.2.1 ระยะเวลาการรับรู้และตอบสนอง (Perception and reaction time)

การขับขี่บนท้องถนนเป็นการกระทำที่รับรู้ทางสายตาและเสียงอย่างต่อเนื่อง โดยผู้ขับขี่จะต้องคอยสนใจอยู่ตลอดเวลาและมีปฏิกิริยาตอบสนองอย่างทันท่วงที ถ้าวิเคราะห์กันอย่างละเอียดแล้วการรับรู้และตอบสนองจะประกอบด้วย 4 ขั้นตอน คือ

1. การรับรู้ (Perception) - การรับรู้ว่ามีเหตุการณ์หนึ่งเกิดขึ้น
2. การเข้าใจหรือจำได้ (Intellection or Identification) - ผู้ขับขี่จำได้ว่าเหตุการณ์นั้นคืออะไร จากการสะสมประสบการณ์ในอดีต
3. การตัดสินใจ (Decision or Emotion) - ผู้ขับขี่วิเคราะห์หาวิธีการตอบสนองที่เหมาะสมกับเหตุการณ์นั้นๆ
4. ปฏิกริยาตอบสนอง (Reaction or Volition) - การดำเนินการอย่างใดอย่างหนึ่งตามการตัดสินใจข้างต้นตัวอย่างเช่น ถ้าผู้ขับขี่กำลังขับรถยนต์มาถึงทางแยกสัญญาณไฟจราจรแห่งหนึ่ง ซึ่งเมื่อกำลังเข้ามาใกล้ทางแยกนั้น สัญญาณไฟได้เปลี่ยนจากสีเขียว

เป็นสีแดง ผู้ขับขี่เริ่มต้นจากการรับรู้ นั่นคือการมองเห็นสัญญาณไฟจราจรเปลี่ยนจากสีเขียวเป็นสีแดง จำได้ว่าสัญญาณไฟสีแดงเป็นการให้สัญญาณหยุดรถ ณ เส้นหยุดรถ เพื่อให้รถยนต์จากทางด้านอื่นไปก่อน ซึ่งเป็นสิ่งที่ผู้ขับขี่ได้สัมผัสประสบการณ์เรื่อยมาจากนั้นจึงตัดสินใจที่จะชะลอความเร็วเพื่อหยุดรถ ณ เส้นหยุดรถ เนื่องจากคาดว่าจะไม่สามารถข้ามทางแยกได้ทันการณ์ จึงทำปฏิกิริยาตอบสนอง โดยการเหยียบเบรกเพื่อชะลอรถยนต์จนรถหยุดที่เส้นหยุด

ระยะเวลาตลอดกระบวนการสี่ขั้นตอนนี้ รวมกันเรียกว่าระยะเวลาการรับรู้และตอบสนอง หรือค่าเวลา PIEV (ตัวย่อของขั้นตอนทั้งสี่) ซึ่งค่านี้ได้นำไปใช้ในการออกแบบด้านวิศวกรรมจราจรมากมายจากกรณีดังกล่าวข้างต้นจึงสังเกตว่า ในขณะที่ผู้ขับขี่กำลังรับรู้และตอบสนองอยู่นั้น รถยนต์ก็ยังคงวิ่งต่อไปเรื่อยๆ ด้วยความเร็วคงที่ (เพราะกำลังรับรู้และตัดสินใจอยู่ แต่ยังไม่ได้เหยียบเบรกจริง) ซึ่งในกรณีที่ยานพาหนะใช้ความเร็วสูงหรือผู้ขับขี่บางกลุ่มที่มีเวลา PIEV มาก รถยนต์คันดังกล่าวก็จะวิ่งเป็นระยะทางอีกไกลพอสมควร จนผู้ขับขี่ไม่สามารถหยุดรถได้ทันการณ์ อาจจะทำให้เกิดอุบัติเหตุได้ ระยะทางรวมที่รถยนต์กำลังวิ่งต่อเนื่องไป ในขณะที่ผู้ขับขี่อยู่ในกระบวนการรับรู้และตอบสนองนั้น จะเรียกว่า ระยะทาง PIEV

โดยทั่วไป ระยะทาง PIEV (ในหน่วยเมตร) มีสูตรคำนวณดังนี้

$$d_p = \frac{1}{3.6} vt \quad (2-1)$$

โดยที่ d_p - ระยะทางการรับรู้และการตอบสนอง (เมตร)

v - ความเร็วของยานพาหนะขณะนั้น (กม./ชม.)

t - ระยะเวลาการรับรู้และตอบสนอง หรือค่าเวลา PIEV (วินาที)

ในสภาพการณ์จราจรจริง ระยะเวลาการรับรู้และตอบสนองของแต่ละบุคคลจะแตกต่างกันไป ทั้งนี้ค่าดังกล่าวจะเพิ่มสูงขึ้นสำหรับเหตุการณ์ที่มีความซับซ้อน

โดยปรกติแล้ว ระยะเวลาการรับรู้และตอบสนอง จะขึ้นอยู่กับปัจจัยหลายประการ เช่น

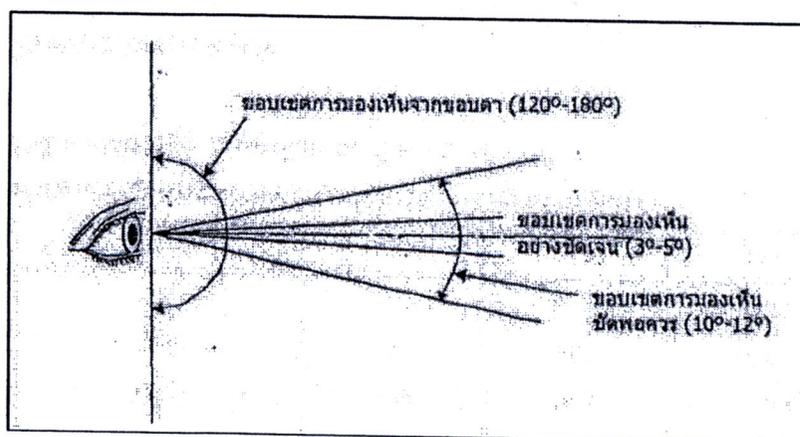
1. อายุ
2. ความล้า
3. ความซับซ้อนของเหตุการณ์
4. ความพิการทางกายภาพ
5. การอยู่ในสภาพมึนเมา

American Association of State Highway and Transportation Officials (AASHTO) ได้แนะนำให้ใช้ค่าเวลาการรับรู้และตอบสนอง (หรือค่า PIEV) เท่ากับ 2.5 วินาที ในการคำนวณระยะการเบรกหรือระยะการหยุดรถในสภาพการณ์ต่างๆ

2.2.2 การมองเห็นและการขับขี

เป็นที่ทราบกันดีว่า ผู้ขับขีต้องสามารถมองเห็นท้องถนนจึงสามารถขับขีรถยนต์ได้ อย่างไรก็ตามที่ผ่านมาไม่ค่อยมีการวิเคราะห์เกี่ยวกับรายละเอียดการขับขีและ Visual Acuity กันมากนัก โดยส่วนใหญ่ในปัจจุบันการตรวจสอบสายตาในการสอบใบขับขีของกรมการขนส่งทางบกจะดำเนินการตรวจสอบเฉพาะความคมชัดในการมองวัตถุหนึ่ง เช่น การอ่านค่าตัวเลขขนาดต่างๆ จากโปสเตอร์ที่ติดบนฝาผนัง ซึ่งในความเป็นจริงแล้ว การมองวัตถุหนึ่งจะมีผลต่อการขับขีน้อยกว่าการมองเห็นในรูปแบบอื่นๆ เช่น ความชัดเจนในการมองวัตถุเคลื่อนที่ (Dynamic visual acuity) การเห็นความใกล้ไกลของวัตถุ (Depth perception) การที่สายตากลับคืนสู่สภาพปกติหลังจากมองแสงจ้า (Glare recovery) และการมองเห็นวัตถุบริเวณขอบตา (Peripheral vision) เป็นต้น คุณลักษณะทางสายตาที่สำคัญเหล่านี้ยังไม่ได้ได้รับการศึกษาวิจัยกันมากนัก ทั้งนี้เพราะว่าโดยปกติแล้วคนส่วนใหญ่ที่มีปัญหาทางด้านสายตาเหล่านี้จะตระหนักถึงความบกพร่องของตนเองและทำการปรับเปลี่ยนพฤติกรรมเพื่อให้แก้ปัญหาในชีวิตประจำวันอยู่แล้ว เช่น ผู้มีปัญหาทางด้านสายตากลับคืนสู่สภาพปกติหลังจากมองแสงจ้าจะต้องใช้เวลาในการคืนสภาพก็จะขับขีรถยนต์ช้าลง และเพิ่มความระมัดระวังในการขับขีในเวลาากลางคืน

รูปที่.2.1 แสดงขอบเขตการมองเห็นที่สำคัญสามประเภท ที่มีความสำคัญในการขับขียานพาหนะ ประเภทแรกคือขอบเขตการมองเห็นอย่างชัดเจน (Field of clear or acute vision) ซึ่งจะครอบคลุมพื้นที่ประมาณ 3-5 องศาโดยรอบของแนวการมองของของตา (พื้นที่การมองเห็นจะมีลักษณะเป็น cone) ผู้ขับขีสามารถอ่านตัวอักษรหรือสัญลักษณ์ต่างๆ ในขอบเขตการมองเห็นนี้ได้อย่างชัดเจน ประเภทที่สองได้แก่ขอบเขตการมองเห็นชัดพอควร (Field of fairly clear vision) ซึ่งจะมีพื้นที่การมองเห็นกว้างกว่าการมองเห็นประเภทแรก นั่นคือครอบคลุมพื้นที่ประมาณ 10-12 องศาโดยรอบแนวการมองของตาภายในขอบเขตนี้ ผู้ขับขีสามารถระบุได้ว่าวัตถุมีสีอะไรและมีรูปร่างอย่างไร แต่ไม่สามารถอ่านตัวอักษรได้ ส่วนประเภทสุดท้าย ได้แก่ขอบเขตการมองเห็นจากขอบตา (Field of peripheral vision) ซึ่งโดยปกติ จะครอบคลุมพื้นที่ประมาณ 120-180 องศาจากศูนย์กลางจากแนวการมองของสายตาในขอบเขตการมองเห็นดังกล่าวผู้ขับขีไม่สามารถบอกสีและรูปร่างของวัตถุได้ แต่สามารถรับรู้ถึงการเคลื่อนที่ของวัตถุในขอบเขตดังกล่าวได้



รูปที่.2.1 ขอบเขตการมองเห็น

ขอบเขตการมองเห็นนี้ มีความสำคัญในการออกแบบติดตั้งป้ายจราจร สัญญาณไฟจราจร และอุปกรณ์ควบคุมการจราจรประเภทต่างๆ เช่นในการวางป้ายเตือนที่มีตัวอักษรกำกับอยู่ ก็ควรจะติดตั้งที่ผู้ขับขี่สามารถมองเห็นด้วยขอบเขตการมองเห็นอย่างชัดเจน หรือ 3-5 องศาจากศูนย์กลางตาต่ำเพื่อให้ผู้สัญจรไปมาสามารถมองเห็นได้อย่างชัดเจนโดยที่ไม่ต้องละสายตาจากถนนเบื้องหน้า

สีและรูปร่างของวัตถุสามารถมองเห็นได้จากขอบเขตการมองเห็นอย่างชัดเจนพอควร ซึ่งมีมุมมองกว้างขึ้น ดังนั้น ป้ายจราจรต่างๆ จึงได้ถูกออกแบบให้มีสีและรูปร่างที่เป็นมาตรฐาน เช่น ป้ายหยุด จะมีพื้นสีแดงและมีรูปร่างแปดเหลี่ยม เป็นต้น

2.2.3 คุณลักษณะอื่นๆ ของผู้ขับขี่

การได้ยินก็เป็นปัจจัยอันหนึ่งที่ช่วยให้การขับขี่เป็นไปอย่างมีประสิทธิภาพและปลอดภัยมากยิ่งขึ้น อย่างไรก็ตามการศึกษาสถิติอุบัติเหตุในอดีตก็ไม่มีที่ยืนยันว่าผู้ที่พิการทางหูจะมีสถิติอุบัติเหตุจราจรมากกว่าผู้ขับขี่อื่นๆ ในปัจจุบันผู้พิการทุกสภาพ ก็สามารถขับขี่ยานพาหนะได้ โดยจะต้องมีการดัดแปลงสภาพรถยนต์ เพื่อให้มีความเหมาะสมกับความพิการนั้นๆ ได้

2.3 คุณลักษณะของยานพาหนะ

คุณลักษณะของยานพาหนะบนท้องถนนมีความแตกต่างกัน มียานพาหนะหลายๆ ประเภทซึ่งมีรูปร่างและสมรรถนะที่แตกต่างกันไป ตั้งแต่ขนาดเล็ก เช่น รถมอเตอร์ไซด์ รถเก๋งสี่ล้อ รถเก๋งอเนกประสงค์ รถกระบะ รถสองแถว จนไปถึงยานพาหนะขนาดใหญ่ เช่น รถประจำทาง รถบรรทุก 6 ล้อ 10 ล้อ 18 ล้อ และรถพ่วง ดังนั้น วิศวกรจราจรจำเป็นต้องคำนึงถึงปัจจัยเหล่านี้ ในการออกแบบและจัดการระบบการจราจรบนท้องถนน

2.3.1 รูปร่างยานพาหนะมาตรฐาน (Design Vehicles)

ในประเทศสหรัฐอเมริกา American Association of State Highway and Transportation Officials (AASHTO) ได้กำหนดมาตรฐานรูปร่างยานพาหนะ (Design Vehicles) สำหรับยานพาหนะไว้ทั้งหมด 10 ประเภท สำหรับใช้ในการออกแบบคุณลักษณะทางเรขาคณิตของท้องถนน (Geometric Design) และโครงสร้างที่อำนวยความสะดวกต่างๆ เช่น ความกว้างช่องจราจร ความกว้างไหล่ทาง รัศมีของขอบทาง และมุมถนน ระดับความสูงของโครงสร้างคร่อมช่องทางจราจรต่างๆ (Clearance heights) และองค์ประกอบอื่นๆ

ตารางที่ 2.1 แสดงขนาดและรูปร่างของยานพาหนะมาตรฐาน 10 ประเภท ที่กำหนดโดย AASHTO ดังแสดงในตาราง จะเห็นได้ว่ายานพาหนะแต่ละประเภทก็จะมีคุณลักษณะความกว้าง ความสูง ความยาวและขนาดอื่นๆ ที่กำหนดเป็นมาตรฐานสำหรับการออกแบบ ทั้งนี้ในกระแสรจราจรจริง ยานพาหนะแต่ละประเภทก็มีรูปร่าง และขนาดแตกต่างกันอยู่บ้างเล็กน้อย รูปร่างของยานพาหนะมาตรฐานที่กำหนดขึ้น ถือว่าเป็นตัวแทนของยานพาหนะนั้นๆ สำหรับใช้ในการออกแบบเท่านั้น

ตารางที่ 2.1 ขนาดของยานพาหนะมาตรฐานที่ใช้สำหรับออกแบบ โดย AASHTO

ประเภทยานพาหนะ	สัญลักษณ์	ขนาด (ฟุต)				
		โดยรวม			เฉพาะส่วนที่ยื่น	
		ความสูง	ความกว้าง	ความยาว	ด้านหน้า	ด้านหลัง
รถยนต์ส่วนบุคคล	P	4.25	7	19	3	5
รถบรรทุก	SU	13.5	8.5	30	4	6
รถประจำทาง	BUS	13.5	8.5	40	7	8
รถประจำทางพ่วง	A-BUS	10.5	8.5	60	8.5	9.5
รถบรรทุก						
รถบรรทุกกึ่งพ่วง - กลาง	WB-40	13.5	8.5	50	4	6
รถบรรทุกกึ่งพ่วง - ใหญ่	WB-50	13.5	8.5	55	3	2
รถบรรทุกพ่วง	WB-60	13.5	8.5	65	2	3
ยานพาหนะสำหรับพักผ่อน						
รถแบบอยู่อาศัยได้	MH		8	30	4	6
รถพ่วงบ้านแคมป์	P/T		8	49	3	10
รถพ่วงเรือ	P/B		8	42	2	8

2.3.2 อัตราการเร่งของยานพาหนะ

ยานพาหนะแต่ละประเภท จะมีขนาด น้ำหนักและกำลังขับเคลื่อนที่แตกต่างกันไป ทำให้มีสมรรถภาพการเร่งความเร็วที่แตกต่างกันด้วย ซึ่งสรุปไว้ใน ตารางที่ 2.2

ตารางที่ 2.2 อัตราการเร่งยานพาหนะแต่ละประเภท

ประเภท ยานพาหนะ	น้ำหนักสุทธิ (ปอนด์)	อัตราความเร็วสูงสุด		
		0-15 ไมล์/ชม.	จาก 40ไมล์/ชม.	จาก 60ไมล์/ชม.
รถยนต์ - ใหญ่	4,800	10.0	4.0	2.5
รถยนต์ - กลาง	4,000	8.0	4.0	2.0
รถคอมแพคต์	3,000	8.0	3.0	1.1
รถยนต์-เล็ก	2,100	6.0	1.2	0.7
รถปิคอัพ	5,000	8.0	1.8	1.5
รถบรรทุก	12,000	2.0	0.6	0.6
รถบรรทุก - ใหญ่	45,000	2.0	0.4	-

จากตารางจะเห็นได้ว่า โดยทั่วไปแล้ว ยานพาหนะขณะที่ใช้ความเร็วต่ำ สามารถเร่งความเร็วได้ดีกว่าขณะที่ใช้ความเร็วสูง และยานพาหนะที่มีขนาดเล็ก และน้ำหนักเบาจะสามารถเร่งความเร็วได้ดีกว่า

ยานพาหนะที่มีขนาดใหญ่และมีน้ำหนักมาก ซึ่งคุณลักษณะนี้มีผลต่อการออกแบบท้องถนนเป็นอย่างมาก เพราะค่าที่ต้องใช้ในการคำนวณระยะทางที่สามารถหยุดรถได้โดยปลอดภัย โดยใช้สมการ

$$d_a = \frac{1}{7.2} at^2 \quad (2-2)$$

โดยที่ d_a - ระยะทางที่รถยนต์เคลื่อนที่ไปขณะที่เร่งความเร็ว (เมตร)
 a - อัตราการเร่งความเร็ว (กม./ชม./วินาที)
 t - ระยะเวลาในการเร่งความเร็ว (วินาที)

ความแตกต่างในการเร่งสูงสุดของยานพาหนะแต่ละประเภท ส่งผลให้ประสิทธิภาพของการจราจรในระบบลดต่ำลง ตัวอย่างเช่น รถบรรทุกขนาดใหญ่ ติดรถสัญญาณไฟจราจร ณ ทางแยกแห่งหนึ่ง เมื่อได้รับสัญญาณไฟเขียว ก็จะเร่งความเร็วได้ช้า ทำให้อานพาหนะประเภทอื่นที่ตามหลังมาเกิดความล่าช้าไปด้วย

2.3.3 การชะลอตัวของยานพาหนะ

การชะลอตัวของยานพาหนะถือเป็นคุณลักษณะของยานพาหนะที่มีความสำคัญมากที่สุด ในด้านการออกแบบบนท้องถนน และความปลอดภัยการจราจร ทั้งระยะเวลา และระยะทางที่จำเป็นต้องใช้ในการชะลอรถยนต์จนหยุดนิ่งสนิท จะนำมาใช้ประกอบการออกแบบและปฏิบัติการของการจราจรในแทบทุกๆ ด้าน ประสิทธิภาพในการชะลอตัวของยานพาหนะ จะขึ้นอยู่กับปัจจัยหลายๆ ประการ ได้แก่ ระบบเบรกของยานพาหนะ ชนิดและสภาพของยางรถยนต์ ตลอดจนประเภทและสภาพของพื้นผิวถนน โดยปรกติการคำนวณที่จำเป็นต้องใช้ในการชะลอรถยนต์ จะใช้สมการ

$$d_b = \frac{v^2 - u^2}{254.0(f+g)} \quad (2-3)$$

โดยที่ d_b - ระยะทางที่รถยนต์เคลื่อนที่ไประหว่างการชะลอความเร็ว (เมตร)
 v - ความเร็วเริ่มต้นก่อนการชะลอความเร็ว (กม./ชม.)
 u - ความเร็วสุดท้ายหลังจากชะลอความเร็ว (กม./ชม.)
 f - ค่าสัมประสิทธิ์ของความต้านทานการเลื่อนไถล (Coefficient of skidding friction)
 g - ความชันของถนน (Decimal)

ทั้งนี้ ระยะทางชะลอรถยนต์ที่คำนวณได้จากสมการ (2-3) ข้างต้นนี้ เป็นระยะทางในช่วงระหว่างการชะลอความเร็วหลังจากเริ่มเหยียบเบรกแล้วเท่านั้น จะไม่รวมถึงเวลารับรู้ตอบสนอง ในกรณีที่มีความชันเป็นแบบลงเขา ก็ให้ใส่ค่าความลาดชันพร้อมด้วยเครื่องหมายลบ

ตารางที่ 2.3 แสดงค่าสัมประสิทธิ์ความต้านทานการลื่นไถลที่นิยมใช้ในการออกแบบ ซึ่งจะแปรผันตามชนิดของพื้นผิวถนน ความเร็วก่อนเริ่มชะลอความเร็ว สภาพยางของรถยนต์ และสภาพพื้นผิวถนนว่าแห้งหรือเปียก

ตารางที่ 2.3 ค่าสัมประสิทธิ์ความต้านทานการลื่นไถลที่ใช้ในการออกแบบ

ประเภทของพื้นผิว	พื้นผิวแห้ง		พื้นผิวเปียก ค่าออกแบบที่แนะนำ โดย AASHTO
	样ใหม่ที่ได้มาตรฐาน	样ที่เสื่อมสภาพแล้ว	
	ความเร็ว = 10 mph (17.7 km/h)		
Dry bit. Conc.	0.74	0.61	-
Sand asphalt	0.75	0.66	-
Rock asphalt	0.78	0.73	-
Port. Cem. Conc.	0.76	0.68	-
	ความเร็ว = 20 mph (32.2 km/h)		
Dry bit. Conc.	0.76	0.60	0.40
Sand asphalt	0.75	0.57	0.40
Rock asphalt	0.76	0.65	0.40
Port. Cem. Conc.	0.73	0.50	0.40
	ความเร็ว = 30 mph (48.2 km/h)		
Dry bit. Conc.	0.79	0.57	0.36
Sand asphalt	0.79	0.48	0.36
Rock asphalt	0.74	0.59	0.36
Port. Cem. Conc.	0.78	0.47	0.36
	ความเร็ว = 40 mph (64.4 km/h)		
Dry bit. Conc.	0.75	0.48	0.33
Sand asphalt	0.75	0.39	0.33
Rock asphalt	0.74	0.50	0.33
Port. Cem. Conc.	0.76	0.33	0.33
ทุกพื้นผิว	ความเร็ว = 50 mph (80.5 km/h)		0.31
	ความเร็ว = 60 mph (96.5 km/h)		0.30
	ความเร็ว = 70 mph (112.6 km/h)		0.29
	ความเร็ว = 80 mph (128.7 km/h)		0.27

2.3.4 คุณลักษณะในการขับเคลื่อน

ในขณะที่ยานพาหนะเคลื่อนที่นั้น จะมีแรงต้านทานภายนอกต่างๆ มาทำให้รถช้าลง ได้แก่ แรงต้านทานจากอากาศ (Air resistance) แรงต้านทานจากความลาดชัน (Grade resistance) แรงต้านทานในการ

ขับเคลื่อน (Rolling resistance) แรงต้านทานจากการเข้าโค้ง (Curve resistance) แรงต้านทานจากความต้านทานบนผิวถนน (Friction resistance)

2.3.5 ตัวอย่างการคำนวณระยะทางการรับรู้และตอบสนอง และระยะทางการชะลอความเร็ว

ดังที่กล่าวมาแล้ว ระยะทางการชะลอรถยนต์ถือเป็นตัวแปรหลักในการออกแบบท้องถนน เพราะมีความสำคัญทางด้านความปลอดภัย ยานพาหนะจะต้องชะลอตัวได้ทันทั่วทั้งที่เพื่อหลีกเลี่ยงการเกิดอุบัติเหตุ ระยะทางดังกล่าวจะประกอบด้วย 2 ส่วน คือ ระยะทางการรับรู้และตอบสนอง ซึ่งเริ่มต้นขึ้นจากการที่ผู้ขับขี่สามารถมองเห็นเหตุการณ์ข้างหน้า จนกระทั่งเริ่มเหยียบเบรก และระยะทางที่ใช้ในการชะลอความเร็ว ยานพาหนะภายหลังจากการเหยียบเบรกแล้ว ดังนั้น สามารถคำนวณได้โดยใช้สูตร

$$d_s = d_p + d_b$$

$$d_s = \frac{1}{3.6} vt + \frac{v^2 - u^2}{254.0(f + g)} \quad (2-4)$$

ระยะทางการหยุดรถอย่างปลอดภัย

โดยหลักการแล้วการออกแบบท้องถนนนั้น จะต้องให้ผู้ขับขี่มีระยะการมองเห็นได้ไม่น้อยกว่าระยะการหยุดรถอย่างปลอดภัย ทั้งนี้เพื่อให้ผู้ขับขี่ เมื่อพบเห็นสิ่งที่กีดขวางถนนข้างหน้า เช่นอุบัติเหตุ ของตกหล่น เป็นต้น ก็สามารถชะลอความเร็วยานพาหนะได้อย่างทันทั่วทั้งที่

ระยะเวลาการเปลี่ยนสัญญาณไฟ ณ ทางแยก

โดยทั่วไปจะเห็นว่าการเปลี่ยนสัญญาณไฟจราจรจากไฟเขียวเป็นไฟแดงนั้น จะต้องให้สัญญาณไฟเหลืองในระยะเวลาสั้นๆ ก่อนเสมอ ทั้งนี้ เพื่อให้ยานพาหนะคันที่ไม่สามารถหยุดรถที่เส้นหยุดได้ทัน สามารถเร่งความเร็วให้ผ่านทางแยกได้อย่างปลอดภัย ก่อนที่สัญญาณไฟเขียวทางด้านอื่นจะเริ่มดับ ถ้าไม่มีสัญญาณไฟเหลือง กล่าวคือ เปิดไฟเขียวให้ด้านอื่นไปทันที จะส่งผลให้เกิดอุบัติเหตุขึ้นมากมาย ดังนั้น จึงต้องมีการพิจารณาระยะเวลาการเปลี่ยนสัญญาณไฟ (Clearance interval) ในการออกแบบไฟสัญญาณจราจรด้วย

ในการวิเคราะห์หาระยะเวลาการเปลี่ยนสัญญาณไฟจราจร (ช่วงเวลาสั้นๆ ที่แสดงสัญญาณไฟเหลือง และไฟแดงทุกด้าน) นั้น จะพิจารณาจากระยะทางการหยุดรถอย่างปลอดภัย ในขณะที่สัญญาณไฟจราจรเปลี่ยนจากสีเขียวเป็นสีแดงนั้น ยานพาหนะที่ยังคงอยู่ห่างจากทางแยกมาก ระยะห่างจากทางแยกมากกว่าระยะการหยุดรถอย่างปลอดภัยนั้น จะสามารถชะลอความเร็วและจอดหยุดนิ่งที่เส้นหยุดได้ทัน ส่วนยานพาหนะที่กำลังเข้าสู่ทางแยก ระยะห่างจากทางแยกน้อยกว่าระยะการหยุดรถอย่างปลอดภัย จะไม่สามารถหยุดรถที่ทางแยกได้ทัน ดังนั้น วิศวกรจะต้องออกแบบสัญญาณไฟจราจรให้รถยนต์เหล่านี้ สามารถเคลื่อนตัวผ่านทางแยกได้อย่างปลอดภัย ก่อนที่จะเปิดสัญญาณไฟเขียวให้กับขาอื่นของทางแยก



2.4 ทฤษฎีการวิเคราะห์การออกแบบสัญญาณไฟจราจร

การวิเคราะห์การออกแบบสัญญาณไฟจราจรมีด้วยกันหลายวิธี ซึ่งได้แก่ รอบเวลาสัญญาณไฟ (Signal Cycle) เวลาล่าช้าของการเริ่มต้นและสุดท้าย (Start and End Lag Time) การสูญเสียของการเคลื่อนที่ (Movement Lost Time) การเคลื่อนที่วิกฤต (Critical Movement) เวลาสูญเสียของทางแยก (Intersection Lost Time) สามารถสรุปได้ดังนี้

2.4.1 รอบเวลาสัญญาณไฟ (Signal Cycle)

ลำดับเวลาจังหวะสัญญาณไฟช่วงเวลาที่ครบ 1 รอบของวงรอบ เรียกว่า รอบเวลาสัญญาณไฟ (c) ผลรวมของเวลาระหว่างไฟเขียวและเวลาแสดงไฟเขียวของทุกเฟส คือรอบเวลาสัญญาณไฟ (Cycle Time) สามารถคำนวณได้จากสมการ

$$c = \sum(I + G) \quad (2-5)$$

เมื่อ C คือ รอบเวลาสัญญาณไฟ (วินาที)

I คือ เวลาระหว่างไฟเขียว (วินาที)

G คือ เวลาแสดงไฟเขียว (วินาที)

2.4.2 เวลาล่าช้าของการเริ่มต้นและสุดท้าย (Start and End Lag Time)

Start Lag (a) คือ ผลรวมของเวลาช่วง Intergreen บวกกับเวลาที่สูญเสียช่วงเริ่มต้น

End Lag (b) คือ เวลาล่าช้าของการสิ้นสุด

ดังนั้นจะมีสมการดังนี้

$$\text{Start Lag, } a = I + ee' \quad (2-6)$$

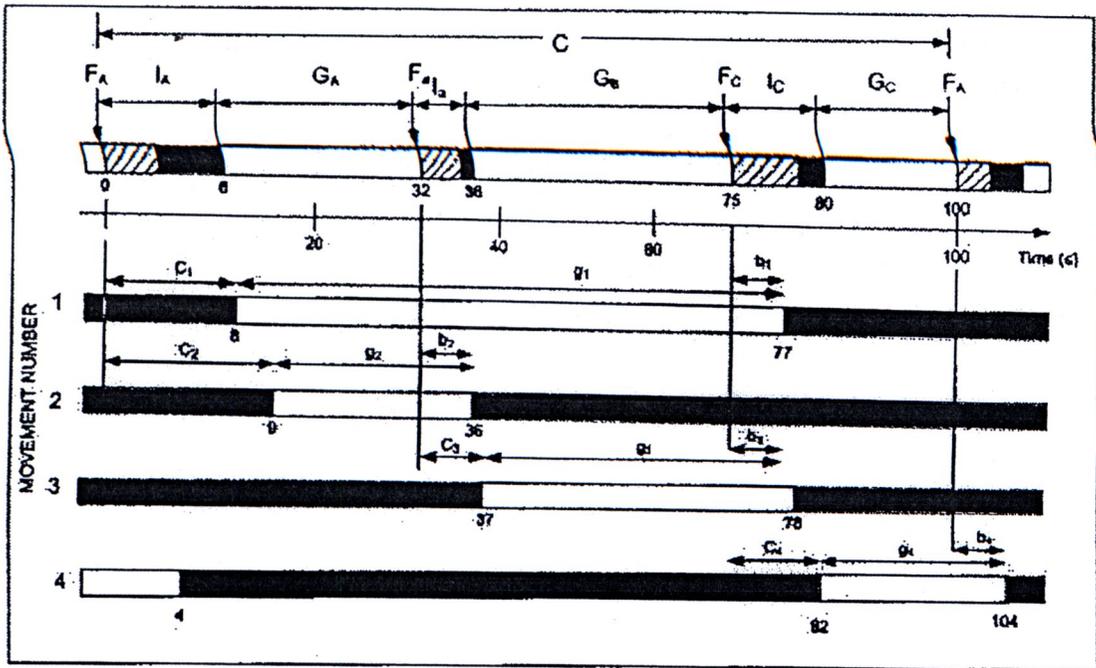
$$\text{End Lag, } b = ff' \quad (2-7)$$

เมื่อ I คือ เวลาระหว่างไฟเขียว (วินาที)

ee' คือ เวลาล่าช้าของการเริ่มต้น (วินาที)

ff' คือ เวลาล่าช้าการสิ้นสุด (วินาที)

สำหรับ เวลาระหว่างไฟเขียวของการเคลื่อนที่ ก็คือ เวลาระหว่างไฟเขียวของเฟสที่กำลังจะเริ่มต้นของการเคลื่อนที่ โดยช่วงประสิทธิภาพเขียวจะเริ่ม ($F_i + a$) และเวลาสิ้นสุด คือ ($F_k + b$) เมื่อ F_i และ F_k เป็นเวลาของการเปลี่ยนเฟส ซึ่งเริ่มต้นและหยุดตามลำดับ



รูปที่.2.2 แสดงเวลาล่าช้าของการเริ่มต้นและสุดท้าย (Start and End Lag Time)

2.4.3 การสูญเสียเวลาของการเคลื่อนที่ (Movement Lost Time)

การสูญเสียเวลาของการเคลื่อนที่ (Movement Lost Time) คือ ผลต่างระหว่าง Start Lag Time กับ End Lag Time สามารถคำนวณได้ดังสมการ

$$l = a - b$$

$$= I + ee' - ff'$$

(2-8)

เมื่อ l คือ เวลาสูญเสียเวลาของการเคลื่อนที่ (วินาที)
 ee' คือ เวลาล่าช้าของการเริ่มต้น (วินาที)
 ff' คือ เวลาล่าช้าการสิ้นสุด (วินาที)



จากการสูญเสียเวลาของการเคลื่อนที่ สามารถสร้างความสัมพันธ์ระหว่าง Movement Lost Time กับ Display Green Time (G) และเวลาไฟเขียวประสิทธิภาพ (g) ได้ดังสมการ

$$g + l = G + I$$

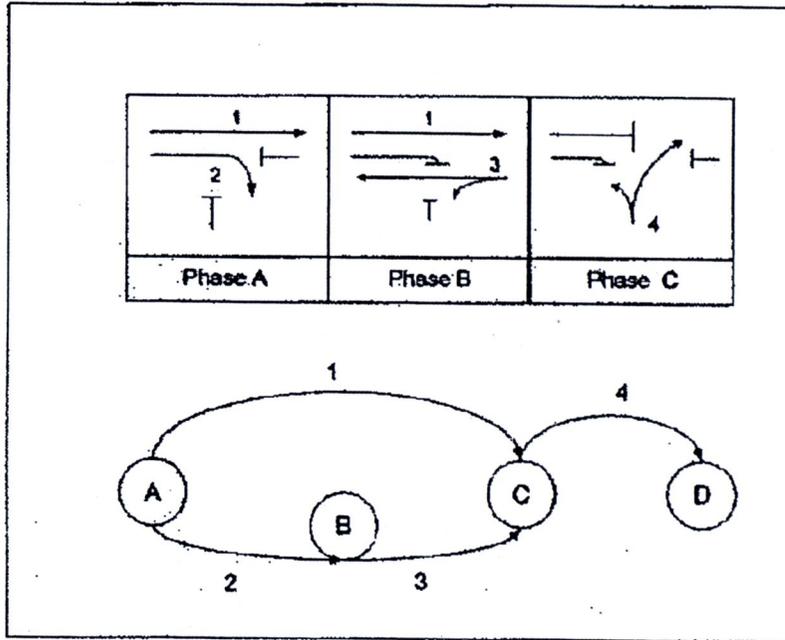
(2-9)

2.4.4 การเคลื่อนที่วิกฤต (Critical Movement)

เป็นการเคลื่อนที่ที่ใช้ในการคำนวณความจุและจังหวะสัญญาณไฟของทางแยกจากสมการ 2-5 การเคลื่อนที่วิกฤตได้อธิบายรอบสัญญาณไฟในรูปแบบผลบวก เวลาไฟเขียวประสิทธิภาพและเวลาที่สูญเสียไฟของการเคลื่อนที่ ในขณะที่การเคลื่อนที่วิกฤตสามารถคำนวณได้จากสมการ

$$c = \sum(g + l) \quad (2-10)$$

- เมื่อ C คือ รอบสัญญาณไฟ (วินาที)
 g คือ เวลาไฟเขียวประสิทธิภาพ สำหรับการเคลื่อนที่วิกฤต (วินาที)
 l คือ เวลาสูญเสียไปของการเคลื่อนที่สำหรับการเคลื่อนที่วิกฤต (วินาที)



รูปที่.2.3 การสูญเสียของการเคลื่อนที่ (Movement Lost Time)

2.4.5 เวลาสูญเสียของทางแยก (Intersection Lost Time)

เวลาสูญเสียของทางแยก (L) สามารถคำนวณได้จากสมการ

$$L = \sum l \quad (2-11)$$

- เมื่อ L คือ เวลาสูญเสียของทางแยก (วินาที)
 l คือ เวลาสูญเสียไปของการเคลื่อนที่สำหรับการเคลื่อนที่วิกฤต (วินาที)

2.5 ทฤษฎีการวิเคราะห์ประสิทธิภาพการทำงานของทางแยกที่ควบคุมด้วยสัญญาณไฟ

2.5.1 การพิจารณาการเคลื่อนที่ (Movement Consideration)

ความจุของทางแยกขึ้นอยู่กับอัตราสูงสุดที่รถยนต์สามารถเคลื่อนที่ออกจากทางแยก คือ ปริมาณจราจรอิ่มตัว (Saturation Flow, s) ความจุของการเคลื่อนที่ที่ทางแยก คำนวณได้จาก

$$Q = s\left(\frac{g}{c}\right) \quad (2-12)$$

เมื่อ	Q	คือ ความจุ (คัน/ชั่วโมง)
	S	คือ ปริมาณจราจรอ้อมตัว (คัน/ชั่วโมง)
	C	คือ รอบสัญญาณไฟ (วินาที)
	g	คือ ระยะเวลาไฟเขียว (วินาที)

2.5.2 ความล่าช้า (Delay)

ความล่าช้าโดยประมาณสำหรับการเคลื่อนที่ที่ทางแยกเดี่ยวแบบกำหนดเวลาแน่นอน สามารถคำนวณจากสมการการจำลองความล่าช้าสม่ำเสมอของ Webster ดังสมการที่ 2-13

$$UD = \frac{0.50C \left[1 - \frac{g}{c}\right]^2}{1 - \left(\frac{g}{c}\right)x} \quad (2-13)$$

เมื่อ	UD	คือ ความล่าช้าเฉลี่ยสม่ำเสมอต่อคัน (วินาที/คัน)
	C	คือ รอบเวลาสัญญาณไฟ (วินาที)
	g	คือ ประสิทธิภาพไฟเขียว (วินาที)
	x	คือ ความจุความอ้อมตัว
	C	คือ ความจุของทางแยก (คัน/ชั่วโมง)

2.6 ระบบควบคุมสัญญาณไฟจราจร

รูปแบบระบบควบคุมสัญญาณไฟจราจรมี 2 ประเภท

1. Fixed Time Plan (ระบบสัญญาณไฟจราจรแบบตั้งเวลาไว้ล่วงหน้า)

เป็นระบบที่ใช้การเก็บข้อมูลสถิติสภาพจราจรที่ผ่านมาในอดีต เพื่อมาคำนวณรอบสัญญาณไฟ ตั้งเวลาของตู้ควบคุม โดยสามารถแยกได้เป็นช่วงเวลาในแต่ละวัน และสามารถแยกได้ระหว่างวันทำงาน (จันทร์ – ศุกร์) และวันหยุด (เสาร์ – อาทิตย์)

2. Vehicle Actuated (ระบบสัญญาณไฟจราจรแบบปรับตามยานพาหนะ)

เป็นระบบที่มีการติดตั้ง Detector บริเวณทางแยกเพื่อประมวลการเดินทางของรถ โดยทำการตรวจสอบรถที่เดินทางเข้ามาในบริเวณทางแยกว่ามีหรือไม่ ถ้ามีจะทำการปล่อยสัญญาณไฟจนกระทั่งหมดรถหรือ ถึงค่าสัญญาณไฟเขียวสูงสุดที่ตั้งไว้ ในกรณีที่ไม่มีรถเดินทางเข้ามา ระบบจะทำการตัดเวลาไฟเขียวดังกล่าวออกไป มีการทำงานดังนี้

2.1 ระบบจะใช้ค่าตั้งต้นจากแผนที่ตั้งไว้ในแต่ละช่วงเวลาของวัน โดยการกำหนดเวลาไฟเขียวต่ำสุดและเวลาไฟเขียวมากที่สุด

2.2 ระบบมีการติดตั้งลูบ (Loop Detector) ในช่องจราจรที่มีความต้องการในการผ่านทางแยกน้อยที่สุด (มุ่งสู่ทางโท) เพื่อส่งข้อมูลไปคำนวณการให้เวลารอบสัญญาณไฟเขียวแก่ทิศทางนั้น โดยระบบจะเริ่มให้จากให้เวลาไฟเขียวต่ำสุด และจะเพิ่มเวลาให้ทีละน้อยในกรณีที่ยังมีรถทับลูบอยู่เรื่อยๆ จนกระทั่งถึงค่าสูงสุดของเวลาที่ตั้งไว้ แต่ถ้า

ไม่มีรถทับลูปนานเกินช่วงเวลาหนึ่ง ระบบจะหยุดการให้ไฟเขียวในจังหวะนั้น เพื่อลดการสูญเสียเวลาโดยเปล่าประโยชน์

2.3 ความยาวรอบสัญญาณไฟ (Cycle Length) จะเป็นผลจากการเพิ่มหรือลดเวลาสัญญาณไฟในแต่ละจังหวะเพื่อตอบสนองสภาพจราจร ไม่มีการควบคุม

2.4 เนื่องจากไม่มีการควบคุมรอบสัญญาณไฟ จึงสามารถพิจารณาค่าเหลื่อมเวลา (Offset) ได้เมื่อสัญญาณไฟในแต่ละทางแยกไม่เท่ากัน สัญญาณไฟจราจรของทั้งระบบจะไม่มีรูปแบบ (Pattern) ทำให้การพยากรณ์การมาถึงของรถจากแยกที่แล้วทำได้ยาก ระบบจึงไม่สามารถคำนวณค่าเหลื่อมเวลาได้

3. Area Traffic Control (ระบบสัญญาณไฟจราจรแบบเป็นพื้นที่)

เป็นระบบควบคุมสัญญาณไฟจราจรที่สามารถปรับเปลี่ยนได้เองอัตโนมัติ (Adaptive Signal Control) การควบคุมในลักษณะนี้อาศัยการตรวจวัดปริมาณการจราจรที่เข้ามาที่ทางแยกโดยผ่านเครื่องตรวจจับ (Detector) โดยจะปรับสัญญาณไฟให้เหมาะสมกับสภาพการจราจรที่วัดได้ในขณะนั้น

ตารางที่ 2.4 แสดงข้อดี-ข้อเสียของระบบ VA และ ATC

	VA (Vehicle Actuated)	ATC (Area Traffic Control)
ความซับซ้อนของระบบ	เรียบง่าย เนื่องจากสามารถเชื่อมต่อข้อมูลจากลูปโดยตรงไปใช้การประมวลผลจาก CPU ของตู้ควบคุมสัญญาณไฟจราจรได้โดยตรง	ซับซ้อนกว่า เนื่องจากในแต่ละทางแยกจะต้องเพิ่มคอมพิวเตอร์ เพื่อประมวลผลเพิ่มเติมจาก CPU ของตู้ควบคุม (CPU ตู้จราจรมีประสิทธิภาพพอจะคำนวณได้)
อุปกรณ์เพิ่มเติมจากตู้จราจรปกติ	การ์ด CPU รุ่นที่สามารถทำงานในโหมด VA ได้ + การ์ดเชื่อมต่อข้อมูลลูป + (อุปกรณ์เชื่อมโยงสัญญาณ)	การ์ด CPU รุ่นที่รองรับระบบ ATC + Embedded Computer + การ์ดเชื่อมต่อข้อมูลลูป + (อุปกรณ์เชื่อมโยงสัญญาณ) * CPU ที่รองรับ ATC สามารถทำงานในโหมด VA ได้เช่นกัน
แนวคิดของระบบ	ลดการสูญเสียเวลาไฟเขียวจากการไม่ได้ใช้ประโยชน์ (ปกติจะใช้เฉพาะในทิศทางรองที่มีการจราจรเบาบาง)	ลดเวลาการรอคอยให้น้อยที่สุดในทุกทิศทางของทางแยก และให้รถติดไฟแดงน้อยที่สุดในทิศทางหลักของทางแยกที่เชื่อมต่อกัน
หลักการทำงาน	ระบบจะเริ่มจากให้เวลาไฟเขียวต่ำที่สุด และจะเพิ่มเวลาให้ทีละน้อยในกรณีที่ยังคงมีรถทับลูปอยู่เรื่อยๆ จนกระทั่งถึงค่าสูงสุด แต่ถ้าหากไม่มีรถทับลูปนานเกินช่วงเวลาหนึ่ง ระบบจะหยุดการให้ไฟเขียวในจังหวะนั้น	ประเมินสภาพจราจรในทุกทิศทางในระยะเวลาหนึ่ง เพื่อคำนวณและปรับปรุงสัญญาณไฟจราจรทีละน้อย (Adaptive) เพื่อให้ได้สัญญาณไฟจราจรที่ดีที่สุดของทั้งทางแยกและของทั้งเส้นทาง (กรณีที่มีการประสานสัมพันธ์ทางแยก)

	VA (Vehicle Actuated)	ATC (Area Traffic Control)
การใช้ข้อมูลจากลูบ	จะดูในช่วงเวลาก่อนหน้าสั้นๆ ว่ามีรถทับลูบหรือไม่ เพื่อเพิ่มเวลาไฟเขียวไปเรื่อยๆ	มีการเก็บเก็บข้อมูลจากลูบทั้งช่วงเวลาไฟเขียวในทุกจังหวะ เพื่อใช้ประเมินสภาพจราจรนำมาคำนวณการปรับปรุงรอบสัดส่วนของสัญญาณไฟเมื่อสิ้นสุดรอบสัญญาณไฟ
การชี้วัดสภาพจราจร	ไม่มีตัวชี้วัดที่สามารถเปรียบเทียบสภาพจราจรระหว่างสองทิศทางได้	ชี้วัดสภาพจราจรได้จากค่า Degree of Saturation ทำให้สามารถปรับสัดส่วนสัญญาณไฟจราจรให้เหมาะสมในแต่ละทิศทางได้
การเพิ่มและตัดเวลาไฟเขียวเฉพาะหน้า	จะเพิ่มเมื่อยังมีรถทับลูบ และจะตัดเวลาไฟเขียวเมื่อไม่มีรถในช่วงเวลาหนึ่ง	มีเฉพาะการตัดเวลาไฟเขียวเมื่อไม่มีรถทับลูบในช่วงเวลาหนึ่ง การเพิ่มเวลาจะเกิดขึ้นจากการคำนวณรอบสัญญาณไฟจราจร
เวลารอคอย	สามารถลดเวลารอคอยได้จากการตัดเวลาไฟเขียวเมื่อไม่มีรถทับลูบ แต่จะไม่คำนึงถึงผลกระทบต่อการรอคอยของรถในจังหวะอื่น จากการเพิ่มเวลาไฟเขียวไปเรื่อยๆ ในกรณีที่ยังมีรถทับลูบ	ประมวลผลโดยคำนึงถึงภาพรวมของรอบสัญญาณไฟและสัดส่วนของจังหวะสัญญาณไฟจราจร มีการคำนึงว่าการเพิ่มเวลาไฟเขียวในทิศทางหนึ่งจะมีผลกระทบต่อเวลารอคอยรถในอีกทิศทาง (ระบบพยายามลดเวลารอคอยของทุกทิศทาง)
การประสานสัมพันธ์ทางแยก	ไม่สามารถทำได้เนื่องจากไม่มีการควบคุมสัญญาณไฟ รอบสัญญาณไฟเปลี่ยนแปลงไปตามการมีหรือไม่มีทับลูบทำให้รอบสัญญาณไฟไม่เท่ากันในแต่ละทางแยก จึงไม่สามารถจัด Offset ได้	สามารถประสานสัมพันธ์ทางแยกได้ เนื่องจากมีระบบคำนวณกลาง (อยู่ที่ศูนย์ควบคุมที่ตู้ควบคุมสัญญาณไฟจราจรก็ได้) ที่คอยคำนวณรอบสัญญาณไฟกลาง (Common Cycle Length) และค่าเหลื่อมเวลา Offset และแต่ละทางแยกสามารถปรับรอบสัญญาณไฟให้เข้ากับรอบสัญญาณไฟกลางได้
ความเหมาะสมกับสภาพจราจร	เหมาะสมกับสภาพจราจรที่ไม่สมดุล (เช่น จังหวะหนึ่งรถมาก อีกจังหวะหนึ่งรถน้อย) หรือสภาพจราจรที่ไม่คับคั่ง เนื่องจากเมื่อมีรถมาก จะมีรถทับลูบตลอดเวลา ไฟเขียวจะไม่เกิดขึ้นซึ่งจะเท่ากับการใช้ระบบ Fixed time	เหมาะสมกับทุกสภาพจราจร โดยสามารถรองรับสภาพการจราจรติดขัดได้ในระดับหนึ่ง เนื่องจากระบบสามารถปรับปรุงรอบและสัดส่วนไฟสัญญาณไฟเอง เพื่อตอบสนองต่อสภาพจราจรได้ในจังหวะนั้น ทั้งนี้ในกรณีที่สภาพการจราจรเบาบางและไม่มีปัญหามาก การใช้ระบบ Fixed time จะคุ้มค่ากว่า
การเปลี่ยนแปลงสภาพการจราจร	ไม่รองรับสภาพการจราจรที่เปลี่ยนแปลง (เช่น ในวันหยุดติดต่อยาว) เนื่องจากพื้นฐานของระบบคือแผนตามช่วงเวลา และระบบไม่สามารถชี้วัดสภาพจราจรได้	มีการปรับปรุงเวลาสัญญาณไฟให้เหมาะสมกับสภาพจราจรในแต่ละช่วงเวลา (Adaptive) ทำให้สัญญาณไฟสามารถปรับได้ตามสภาพจราจรที่เปลี่ยนแปลงไป

การควบคุมสัญญาณไฟจราจร โดยระบบ VA และ ATC เป็นที่ยอมรับอย่างกว้างขวางว่าสามารถเพิ่มประสิทธิภาพในการควบคุมการจราจรได้ โดยนิยมเรียกระบบการควบคุมสัญญาณไฟจราจรแบบนี้ว่า ระบบควบคุมสัญญาณไฟจราจรอัจฉริยะ และสามารถลดการใช้พลังงานในภาคขนส่งได้ โดยเฉพาะอย่างยิ่ง บริเวณจุดตัดถนนสายหลักและถนนสายรอง (ตรอก/ซอย) โดยใช้หลักการคือ การเพิ่มประสิทธิภาพการใช้ประโยชน์จากไฟเขียวให้มากที่สุด โดยที่การควบคุมสัญญาณไฟจราจร โดยระบบ Fixed Time Plan ไม่สามารถทำได้ เพราะบ่อยครั้งจะพบว่าจะมีการเปิดสัญญาณไฟเขียวโดยไม่มีรถยนต์จอดติดสัญญาณไฟจราจร อยู่เลย

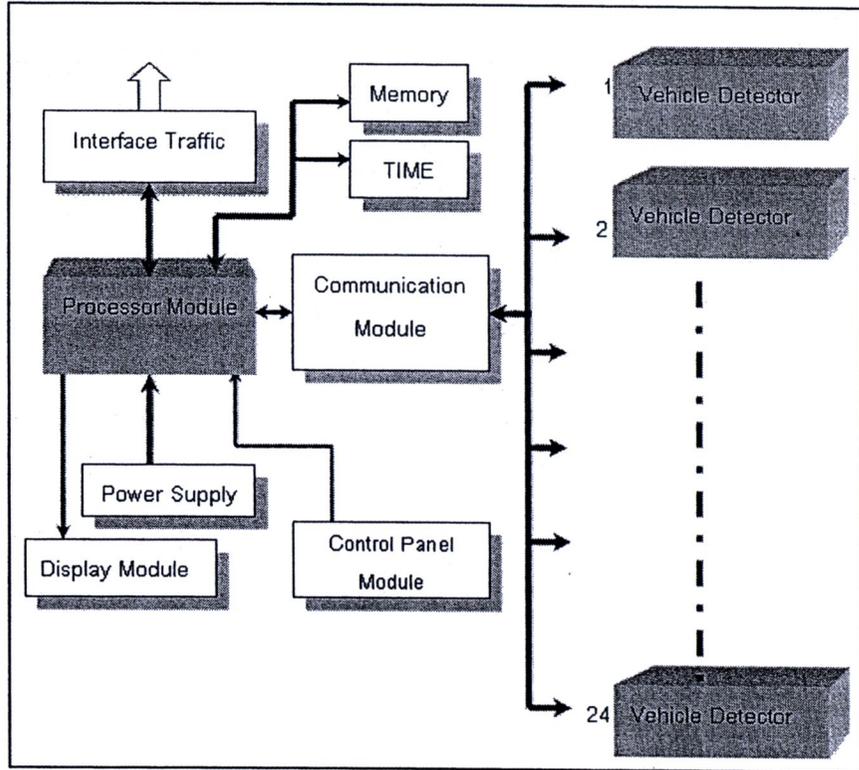
ระบบควบคุมสัญญาณไฟจราจรอัจฉริยะ(ATC) ได้รับการติดตั้งอยู่หลายจุดทั้งกรุงเทพฯ ปริมณฑล และเมืองใหญ่ๆ เช่น พัทยา หาดใหญ่ เชียงใหม่ ฯลฯ และยังไม่มีกรายงานและประเมินผลถึงการประหยัดพลังงานอย่างชัดเจน อันเนื่องมาจากสาเหตุ คือ

1. บริเวณกรุงเทพมหานครและปริมณฑล เป็นบริเวณที่มีจราจรหนาแน่น มีการใช้ประโยชน์จากสัญญาณไฟเขียวเต็มที่
2. การจราจรในตัวเมืองของประเทศไทยมียานพาหนะประเภท รถมอเตอร์ไซด์ จำนวนมากและอุปกรณ์ตรวจจับ (Detector) มีความไวไม่เพียงพอ

2.7 องค์ประกอบของเครื่องควบคุมไฟจราจร

จากการทบทวนถึงงานวิจัยที่เกี่ยวข้องกับเครื่องควบคุมการจราจรในประเทศไทยสามารถออกแบบ หลักการทำงานของฮาร์ดแวร์ไว้ดังรูปที่.2.4

- ส่วนโปรเซสเซอร์ (Processor Module) ทำหน้าที่เก็บรวบรวมข้อมูลและโปรแกรมควบคุม ทั้งยังเป็นเสมือนสมองของเครื่องควบคุมสัญญาณไฟจราจร ที่คอยสั่งการ และรับรู้เงื่อนไขต่างๆ เช่น เงื่อนไขทางเวลาเพื่อนำไปเลือกโปรแกรมทำงาน, เงื่อนไขจากปุ่มควบคุมเพื่อกำหนดรูปแบบการจราจรตามที่คนควบคุมต้องการ เป็นต้น
- ส่วนแผงควบคุม (Control Panel Module) เป็นส่วนที่รวบรวมปุ่มควบคุม และเป็นสื่อกลางในการติดต่อระหว่างผู้ใช้กับเครื่อง (Man-Machine Interface) ส่วนขับหลอดไฟจราจร (Output Drive Module) ทำหน้าที่แปลงสัญญาณ ที่ได้รับจากส่วนโปรเซสเซอร์ให้เป็นไฟฟ้ากระแสสลับ 220 โวลต์ เพื่อไปขับหลอดไฟจราจร
- ส่วนติดต่อสื่อสาร (Communication Module) ทำหน้าที่ในการเชื่อมโยงเครื่องควบคุมกับอุปกรณ์สื่อสารเช่น โมเด็ม (Modem) เพื่อการส่งข้อมูลระยะไกล หรือติดต่อกับส่วนป้อนโปรแกรมแบบมือถือ (Hand Held Programmer) หรือกับคอมพิวเตอร์เพื่อป้อนข้อมูลต่างๆ รวมทั้งโปรแกรมควบคุมให้กับควบคุม
- ส่วนบันทึกข้อมูล (Memory Module) เพื่อนำมาใช้ประมวลผล
- ส่วนฐานเวลา (Time Module)
- ส่วนแสดงผล (Display Module) ใช้ในการแสดงข้อมูลการทำงาน และตั้งข้อมูล
- ส่วนจ่ายกำลังไฟฟ้า (Power Supply Module)



รูปที่.2.5 ลักษณะพื้นฐานของส่วนประมวลผลกับตัวตรวจจับยานพาหนะ

2.8 ระบบตัวตรวจวัด (Detectors)

เป็นปัจจัยสำคัญในการตรวจสอบยานพาหนะบนท้องถนน ตัวตรวจวัดจะใช้หลักการทำงานของขดลวดเหนี่ยวนำ จากขดลวดหรือสายไฟมาพันเป็นรูปสี่เหลี่ยมผืนผ้า, สี่เหลี่ยมจัตุรัส, หรือเป็นรูปวงกลม โดยมีจำนวนรอบตั้งแต่หนึ่งรอบขึ้นไปจะทำให้ขดลวดหรือสายไฟเส้นนั้นมีคุณสมบัติเป็นตัวเหนี่ยวนำ (Inductor) ค่าความเหนี่ยวนำ (Inductance) ที่ได้อยู่ที่ขนาด และจำนวนรอบการคำนวณทั่วไปของค่าเหนี่ยวนำคือ

$$L = P (t^2 + t) / 4 \quad (2-14)$$

โดย L = ค่าความเหนี่ยวนำ (ไมโครเฮนรี)
 P = เส้นรอบรูป (ฟุต)
 t = จำนวนรอบ

ค่าความไวในการตรวจวัดยานพาหนะ

จากสมการ

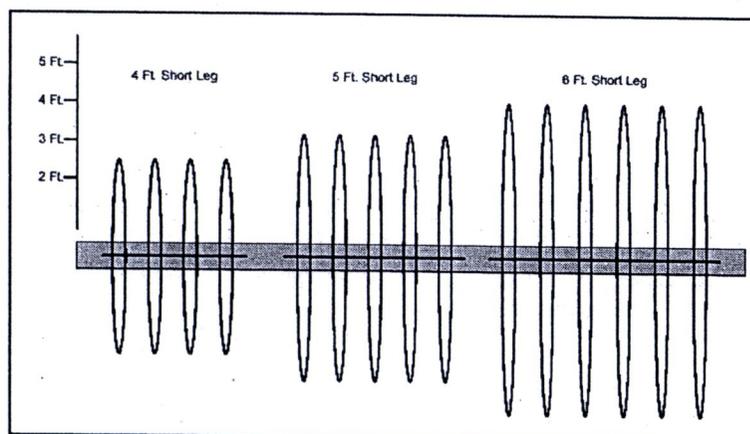
$$\text{Amount of Change} \approx \frac{\text{Vehicle Size}}{(\text{Loop Size}) \times (\text{Vehicle Height})} \quad (2-15)$$

โดย Vehicle Size = ขนาดของยานพาหนะ
 Loop Size = ขนาดของลูป
 Vehicle Height = ความสูงของยานพาหนะ

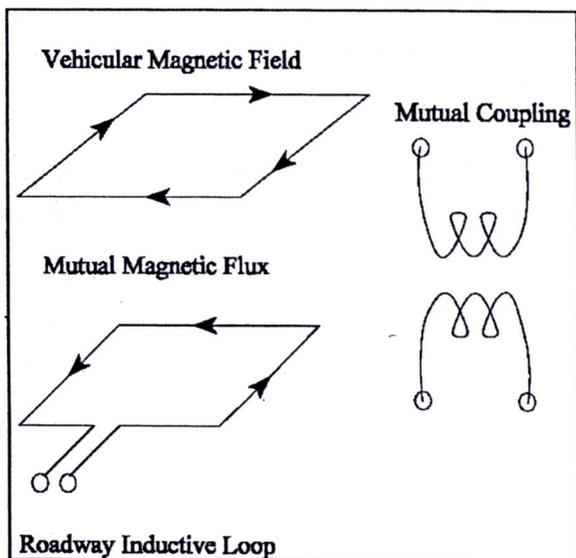


ดังนั้นจากสมการที่ (2-14) และ (2-15) จะทำให้ได้ว่า

- การเพิ่มขนาดของลูป จะลดการเปลี่ยนแปลงที่กระทำจากยานพาหนะ
- ถ้ายานพาหนะมีความสูง ความเหนียววน่าที่เกิดจากยานพาหนะก็จะลดลง
- ถ้ายานพาหนะมีขนาดเล็กการเปลี่ยนแปลงที่กระทำจากยานพาหนะก็จะน้อยลง
- การเพิ่มจำนวนรอบไม่มีผลต่อความสูงที่ต้องการตรวจวัด
- ขนาดของลูปมีผลต่อความสูงในการตรวจวัด



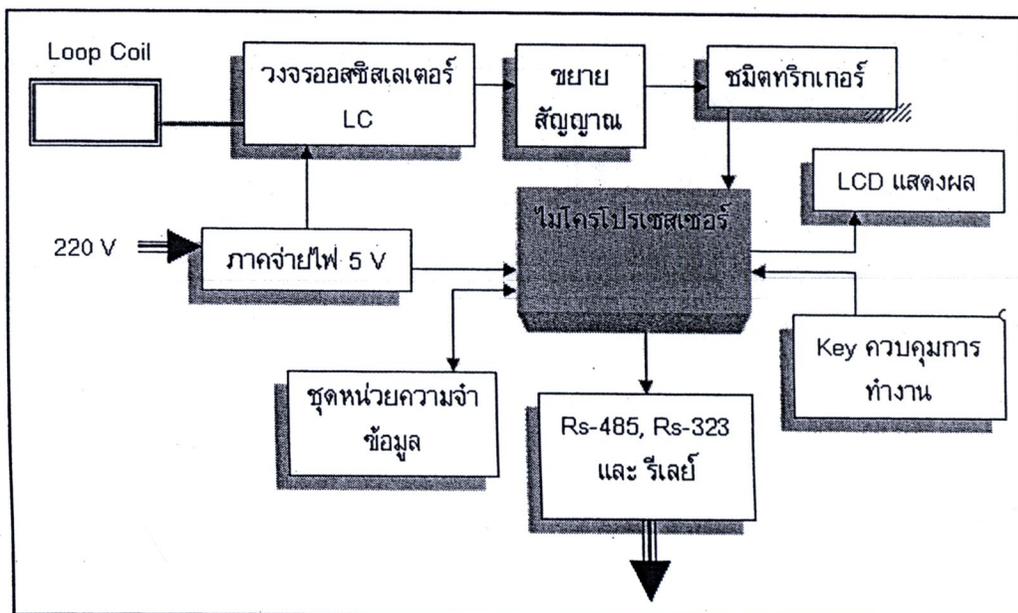
รูปที่.2.6 แสดงลำดับความสูงที่เหมาะสมในการตรวจวัด



รูปที่.2.7 แสดงผลการกระทำระหว่างยานพาหนะกับขดลวดเหนี่ยวนำ

วงจรการทำงาน

ขดลวดจะถูกฝังลงบริเวณพื้นดิน และเมื่อตัวตรวจจับได้ไปพบกับวัตถุที่เป็นยานพาหนะ จะทำให้ความถี่ในภาคผลิตความถี่อิสระดังที่ได้กล่าวมาแล้วแต่ตอนต้นเกิดการเปลี่ยนแปลงความถี่ การเปลี่ยนแปลงความถี่ดังกล่าวนี้จะถูกขยายสัญญาณและแปลงสัญญาณเป็นรูปสี่เหลี่ยมเข้าไปในส่วนวิเคราะห์ข้อมูลด้วยอัลกอริทึมในไมโครคอนโทรลเลอร์ บล็อกไดอะแกรมดังแสดงในรูปที่.2.8



รูปที่.2.8 ลักษณะวงจรพื้นฐานของส่วนเครื่องตรวจจับยานพาหนะแบบลูปเหนี่ยวนำ

2.9 วิวัฒนาการของระบบตรวจวัดยานพาหนะ

อุปกรณ์ตรวจวัดการจราจร (Traffic flow sensors) คือ อุปกรณ์ที่ใช้ในการตรวจวัดการมีอยู่ (presence) หรือการผ่าน (passage) ของยานพาหนะ ณ บริเวณจุดสำรวจให้ข้อมูลที่ใช้ในการจัดการจราจร และข้อมูลที่ใช้ในการวางแผน

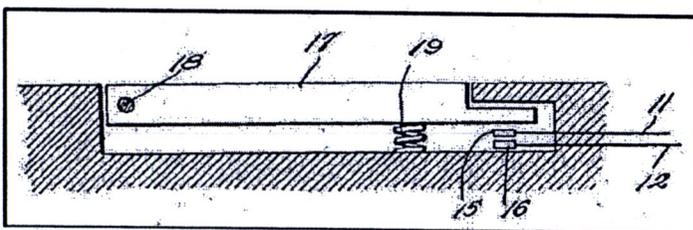
พัฒนาการของเทคโนโลยีตรวจวัดการจราจร

ค.ศ. 1928 Horn-activated traffic signal (developed by Charles Adler, Jr.) เป็นเครื่องมือวัดปริมาณรถโดยอาศัยการติดตั้ง Micro Phone



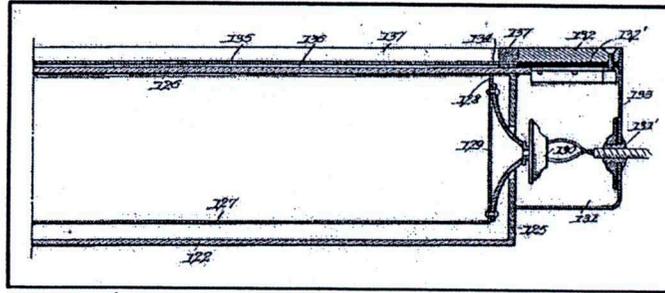
รูปที่.2.9 การตรวจวัดยานพาหนะโดยใช้ Micro Phone

ค.ศ. 1928 In-roadway pressure sensitive sensor (developed by Henry A. Haugh) เป็นเครื่องมือวัดปริมาณรถโดยอาศัยกระแสไฟฟ้าที่ได้จากแรงกดทับของแผ่นเหล็กที่ติดตั้งไว้ใต้ผิวถนน



รูปที่.2.10 การตรวจวัดยานพาหนะโดยการกดทับของแผ่นเหล็ก

ค.ศ. 1930 In-roadway pressure sensitive sensor (developed by Charles Adler, Jr.) เป็นเครื่องมือวัดปริมาณรถโดยอาศัยเสียงซึ่งได้มาจากการติดตั้งเครื่องวัดเสียงที่ติดตั้งไว้ใต้ผิวถนน



รูปที่.2.11 การตรวจวัดยานพาหนะโดยอาศัยเสียง

ซึ่ง In-roadway pressure sensitive sensor ได้รับความนิยมมากแต่ก็ยังมีปัญหาซึ่งสรุปได้ดังนี้

1. อุปกรณ์มีราคาแพง
2. ต้องติดตั้งอุปกรณ์ใหม่หากมีการปรับปรุงผิวถนน
3. ความเสียหายเนื่องจากรถกวดหิมะ

2.9.1 อุปกรณ์ตรวจวัดการจราจรที่ติดตั้งอยู่บนหรือใต้ผิวทาง (In-roadway sensors)

1. Pneumatic road tube

เริ่มใช้ตั้งแต่ปี 1920s เป็นเครื่องตรวจวัดการจราจรที่ติดตั้งโดยอาศัยท่อยางเพื่อตรวจวัด

หลักการทำงาน

น้ำหนักรถกดทับท่อลมเป็นเครื่องตรวจวัดการจราจรที่ติดตั้งบนผิวทางวางขวางช่องทางเดินรถ เคลื่อนที่อาศัยน้ำหนักรถกดทับท่อลม ซึ่งทำด้วยยางแล้วเกิดแรงดันกระจายไปในแนวท่ออย่างโดยปลายด้านหนึ่งของท่อปิดตันและอีกด้านต่อเข้ากับอุปกรณ์ระบบไฟฟ้า ซึ่งทำงานด้วยแรงดันที่เปลี่ยนแปลงในเรือท่อ ทำให้ทราบจำนวนการกดทับและวิ่งผ่านท่อลมซึ่งสามารถจัดเก็บข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- แยกประเภทรถโดยใช้จำนวนเพลลาและระยะห่างระหว่างเพลลา, เวลาห่างระหว่างรถ (Gap)

การติดตั้ง Pneumatic road tube

- เลือกตำแหน่งที่ต้องการติดตั้ง
- ติดตั้ง End plug เพื่อกันน้ำหรือฝุ่นเข้า
- ยึดปลายด้านหนึ่งของท่อลมให้แน่นกับพื้นถนน
- ติดท่อลมกับผิวถนนด้วยเทปกาวยาส์ฟิลท์
- วัดระยะเพื่อติดตั้งท่อลมอีกเส้น (ถ้ามี)
- ติดปลายท่ออีกด้านเข้ากับเครื่องนับและถ้าทำการติดตั้ง Two-pneumatic road tubes connected together จะช่วยลดข้อผิดพลาดจากการวัดระยะห่างท่อลมขณะติดตั้ง



รูปที่.2.12 การตรวจวัดยานพาหนะโดยใช้ท่อลม

ข้อดี

- ใช้งานง่าย
- ราคาถูก

ข้อเสีย

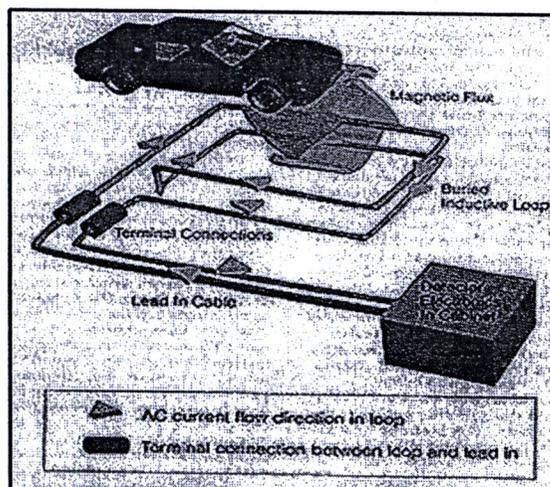
- ท่อลมชำรุดได้ง่ายถ้ามีปริมาณการจราจรสูง
- การติดตั้งเหมาะสมกับการติดตั้งแบบชั่วคราวและเป็นบริเวณที่
- ไม่ซับซ้อน
- การติดตั้งต้องปิดการจราจรถึงจะทำการติดตั้งได้

2. Inductive loop detectors

เริ่มใช้ตั้งแต่ปี 1960s เป็นเครื่องมือสำหรับจัดการจราจรที่ตรวจวัดการเปลี่ยนแปลงของสนามแม่เหล็กไฟฟ้า

หลักการทำงาน

เปลี่ยนแปลงค่าความต้านทานการเหนี่ยวนำของกระแสไฟฟ้าในขดลวดเหนี่ยวนำ



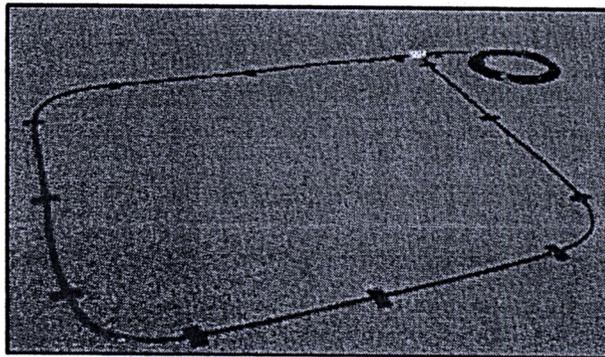
รูปที่.2.13 การตรวจวัดยานพาหนะโดยใช้ Inductive loop

ซึ่งประเภทของ Inductive loop detectors มีดังนี้

- Saw cut loops เป็นการกัดพื้นผิวของถนนแล้วทำการฝังสาย Loop ซึ่งมี

วิธีการติดตั้ง Saw-cut loops ดังนี้

1. วาง layout ของ loop บนผิวถนน
 2. กัดผิวถนน
 3. ติดตั้งสาย Loop
 4. ปิด Loop ด้วย Sealant
- Trenched-in loops เป็นการวางสาย Loop ก่อนการทำถนน
 - Pre-formed loops เป็นการทำสาย Loop สำเร็จรูปแล้วนำมาวางบนพื้นถนนแล้วนำเทปกาวติดเพื่อยึดสาย Loop ไว้บนถนนซึ่งมีข้อเสียคือจะไม่มั่นคงแต่จะมีข้อดีตรงที่สามารถเคลื่อนย้ายง่ายเมื่อต้องการนำไปติดตั้ง



รูปที่.2.14 Inductive loop บนถนน

หมายเหตุ ตัวอย่าง Layout ของ Loop detectors ต่างๆ เช่น รูปวงกลม, รูปสี่เหลี่ยม, รูปหกเหลี่ยม เป็นต้น

เครื่องตรวจวัดสามารถจัดเก็บข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- ระยะเวลาครอบครอง (Occupancy)
- ระยะห่างของรถแต่ละคัน (Gap)
- Headway
- แยกประเภทรถโดยใช้ vehicle signature

ข้อดี

- เป็นเทคโนโลยีที่มีการพัฒนามานานมีคนใช้เยอะ
- ให้ข้อมูลการนับรถที่ถูกต้องที่สุด
- สามารถนำไปประยุกต์ใช้กับงานได้หลายรูปแบบ
- ไม่ไวต่อสภาพอากาศที่เปลี่ยนแปลงใช้งานง่าย

ข้อเสีย

- การติดตั้งจำเป็นต้องมีการกีดผิวทาง
- การติดตั้งที่ไม่ดีทำให้ผิวทางมีอายุการใช้งานสั้นลง
- ต้องมีการปิดการจราจรขณะติดตั้งและซ่อมแซม
- ในแต่ละจุดต้องใช้ loop หลายตัวในการตรวจวัดการจราจร
- น้ำหนักกดทับและอุณหภูมิส่งผลต่อ loop ที่ฝังไว้

3. Magnetic sensors

หลักการทำงาน

การเปลี่ยนแปลงของสนามแม่เหล็กโลกซึ่งประเภทของ Magnetic sensors มีดังนี้

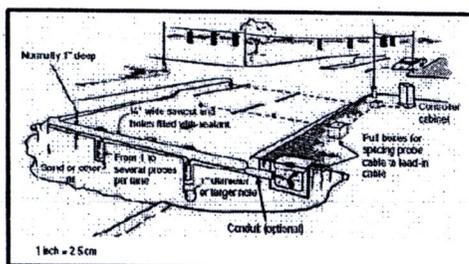
- Two-axis fluxgate magnetometer
- Induction or search coil magnetometer

เครื่องตรวจวัดสามารถจัดเก็บข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- ระยะเวลาครอบครอง (Occupancy)
- แยกประเภทรถโดยใช้ vehicle magnetic signature

การติดตั้ง Magnetic Sensors

เจาะรูบนผิวถนนความลึกประมาณ 5 cm แล้วนำ Portable magnetic sensor วางไว้แล้วนำแผ่นยางปิดบริเวณดังกล่าวโดยใช้ตะปูตอกเพื่อยึดแผ่นยางไม้ได้แผ่นยางเปิดวิธีนี้ง่ายต่อการติดตั้ง



รูปที่.2.15 การติดตั้ง Magnetic Sensors

ข้อดี

- สามารถใช้ในบริเวณที่ไม่สามารถติดตั้ง loop ได้
- ไม่ไวต่อสภาพอากาศ
- ผลจากแรงกดทับจากปริมาณจราจรมีน้อยกว่า loop

ข้อเสีย

- การติดตั้งจำเป็นต้องมีการกีดผิวทางหรือเจาะเป็นช่องใต้ผิวทาง
- ไม่สามารถตรวจจับรถที่จอดนิ่งอยู่กับที่ได้หากไม่มีการวาง layout
- ของอุปกรณ์ตรวจวัดแบบพิเศษและใช้ software เสริม

2.9.2 อุปกรณ์ตรวจวัดการจราจรที่ติดตั้งอยู่เหนือผิวทาง (Over-roadway sensors)

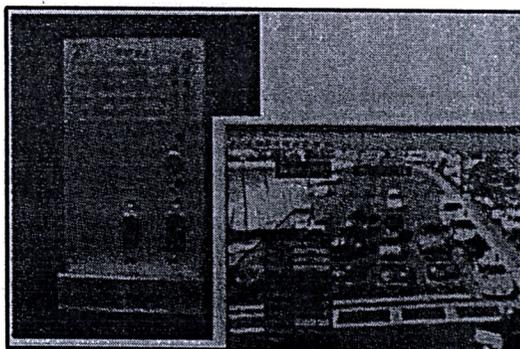
1. Video Image Processors

เริ่มใช้ตั้งแต่ปี 1970s

หลักการทำงาน

วิเคราะห์ภาพจาก video frames เครื่องตรวจวัดสามารถให้ข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- ระยะเวลาครอบครอง (Occupancy)
- ระยะห่างของรถแต่ละคัน (Gap)
- Headway
- แยกประเภทรถโดยใช้ความยาว
- ความหนาแน่นของปริมาณรถ
- ความยาวของแถวรถที่ติดอยู่บนท้องถนน



รูปที่.2.16 การตรวจวัดยานพาหนะโดยใช้ Video Image Processors

ข้อดี

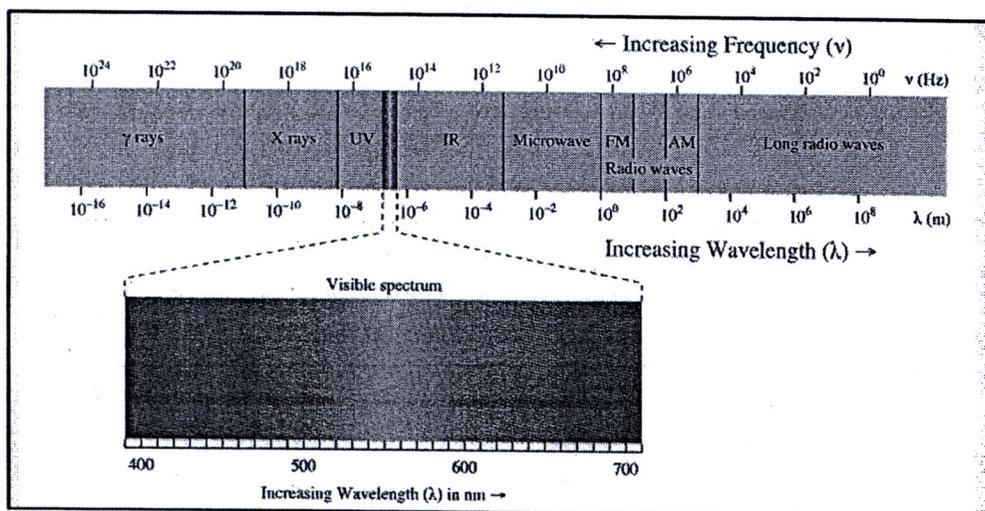
- สามารถตรวจวัดได้หลายช่องจรรยาพร้อมๆกัน
- ง่ายในการเพิ่มพื้นที่ตรวจวัด
- ให้ข้อมูลจรรยาหลายอย่าง
- สามารถตรวจวัดได้ในบริเวณกว้างเมื่อมีการเชื่อมข้อมูลระหว่างกล้อง

ข้อเสีย

- ต้องมีการปิดช่องจรรยาหากติดตั้งกึ่งกลางถนน
- ต้องมีการทำความสะอาดเลนส์ตามระยะเวลา
- สภาพอากาศและความชื้นสัมพัทธ์มีผลต่อประสิทธิภาพของ กล้อง
- ต้องมีไฟส่องสว่างที่เพียงพอในเวลากลางคืน
- ต้องติดตั้งกล้องจากมุมสูง
- จะคุ้มค่างี้ต่อเมื่อมีการตรวจวัดหลายพื้นที่พร้อมกันในกล้องเดียวหรือต้องการข้อมูลบางอย่างเป็นพิเศษ

2. Microwave radar sensors

คลื่นไมโครเวฟ (Microwaves) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในช่วง 3000 MHz คลื่นไมโครเวฟได้ถูกนำมาประยุกต์ใช้ในอุปกรณ์ต่างๆ เช่น เตาไมโครเวฟ โทรศัพท์มือถือ รวมถึงเครื่องมือในการตรวจวัดการจราจร



รูปที่.2.17 คลื่นแม่เหล็กไฟฟ้าที่มีความถี่ต่างๆ

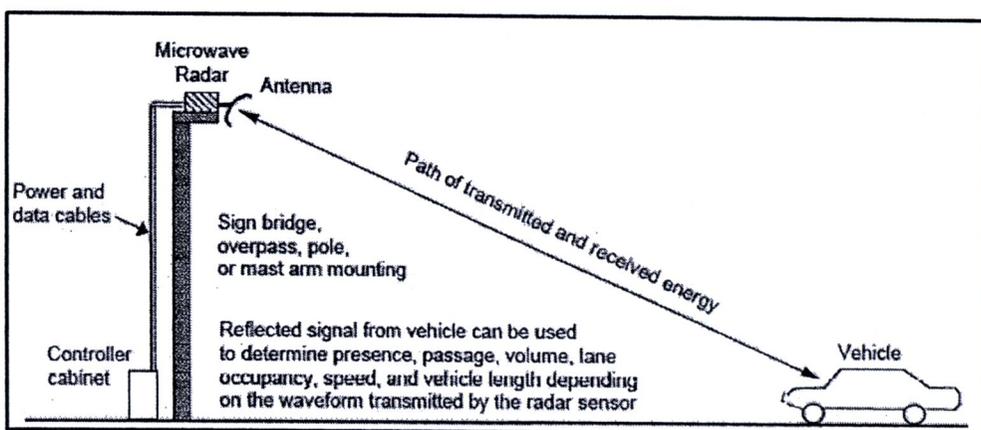
เครื่องมือตรวจวัดการจราจรแบบ Microwave Radar Sensors มี 2 ประเภทคือ

1. แบบ Doppler Microwave
2. แบบ Frequency – modulated continuous wave

หลักการทํางาน

เครื่องตรวจวัดจะส่งสัญญาณและสะท้อนกลับของคลื่น Microwave ออกมาเพื่อตรวจจําบรถยนต์ที่วิ่งผ่านแนวสัญญาณเครื่องตรวจวัดประมวลผลออกมาเป็นข้อมูลดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- ระยะเวลาครอบครอง (Occupancy)
- ระยะห่างของรถแต่ละคัน (Gap)
- Headway
- ประเภทรถยนต์จําแนกตามความยาวรถแต่ละคัน



รูปที่.2.18 การตรวจวัดยานพาหนะโดย Microwave radar sensors

ข้อดี

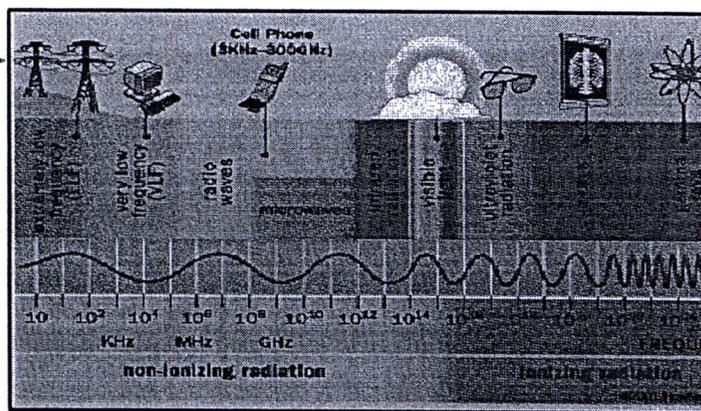
- ไม่ไวต่อสภาพอากาศสำหรับการใช้งานตรวจจับในช่วงสั้น
- สามารถวัดความเร็วได้โดยตรง
- สามารถใช้ตรวจวัดการจราจรได้หลายช่องทางในเวลาเดียวกัน

ข้อเสีย

- Continuous Wave Doppler Sensors ไม่สามารถตรวจวัดรถที่จอดนิ่งอยู่กับที่

3. Active infrared sensors (laser sensors)

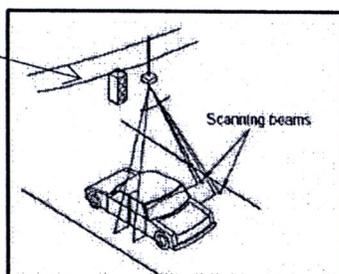
คลื่นอินฟราเรด (Infrared) เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในช่วง 10¹¹ Hz - 10¹⁴ Hz ซึ่งมีความถี่สูงกว่าคลื่นไมโครเวฟเครื่องตรวจวัดการจราจรแบบ Active Infrared Sensors



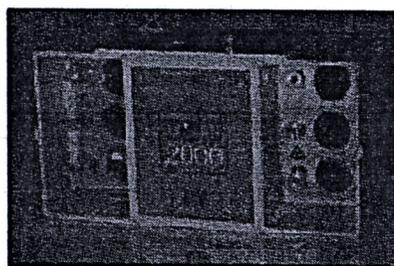
รูปที่.2.19 การตรวจวัดยานพาหนะโดยใช้ Active infrared sensors

เครื่องตรวจวัดการจราจรแบบ Active Infrared Sensors มีลักษณะการทำงานคล้ายๆ กับการทำงานของเครื่องตรวจวัดการจราจรแบบ Microwave Radar Sensors คือมีการปล่อยคลื่นอินฟราเรดออกมาเพื่อตรวจจับรถยนต์ที่วิ่งผ่านแนวสัญญาณดังรูป

Active Infrared Sensors



รูปที่.2.20 การตรวจวัดการจราจรแบบ Microwave Radar Sensors



รูปที่.2.21 เครื่องตัววัดจราจร แบบ Active Infrared Sensors

หลักการทำงาน

เป็นอุปกรณ์ลำแสง 2 ชุดกวาดผ่านช่องจราจรที่จะทำการตรวจวัดแล้วอุปกรณ์ตรวจจับพลังงานแสงที่เกิดการกระจายเมื่อยานพาหนะเคลื่อนที่ผ่านแล้วแปรสัญญาณส่งผ่านอุปกรณ์ประมวลผลซึ่งเครื่องตรวจวัดสามารถจัดเก็บข้อมูลได้ดังนี้

- ปริมาณการจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- ประเภทรถยนต์จำแนกตามความยาวรถแต่ละคัน

ข้อดี

- สัญญาณที่ส่งออกมามีหลายลำทำให้สามารถตรวจจับตำแหน่งรถความเร็ว
- และประเภทของได้อย่างแม่นยำสูง
- สามารถใช้ตรวจวัดการจราจรได้หลายช่องทางในเวลาเดียวกัน

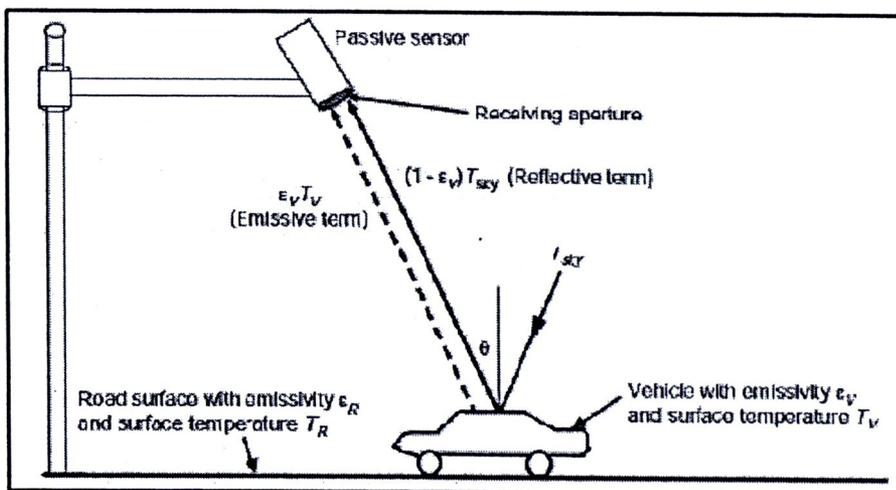
ข้อเสีย

- ประสิทธิภาพในการทำงานลดหากมีหมอกหรือมีหิมะตก
- ต้องมีระยะเวลาในการทำความสะอาดเลนส์เป็นประจำ
- ต้องมีการปิดช่องการจราจรเพื่อทำการติดตั้ง

4. Passive infrared sensors

หลักการทำงาน

เป็นเครื่องมือวัดการจราจรตรวจจับพลังงานที่สะท้อนจากวัตถุเข้าสู่ตัวกล้องที่เกิดจากวัตถุที่เคลื่อนที่ผ่านหรือวัตถุที่อยู่ในบริเวณพื้นที่ตรวจวัด เช่น ผิวจราจรรถยนต์และพลังงานที่เกิดจากบรรยากาศบริเวณนั้นสะท้อนผ่านวัตถุที่เคลื่อนที่ เครื่องมือจะตรวจจับการเปลี่ยนแปลงของ Infrared Spectrum แปรข้อมูลส่งผ่านเครื่องประมวลผล



รูปที่.2.22 การตรวจวัดยานพาหนะโดยใช้ Passive infrared sensors

ซึ่งประเภทของ Passive infrared sensors มีดังนี้

- Non-imaging
- Imaging

เครื่องตรวจจับสามารถให้ข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- แยกประเภทรถโดยใช้ความยาว
- ความยาวของแถวรถที่ติดอยู่บนท้องถนน

ข้อดี

- ระบบตรวจจับแบบหลายพื้นที่ใช้วัดความเร็ว

ข้อเสีย

- ประสิทธิภาพลดลงช่วงฝนตกหิมะตกหมอกกลาง
- บางรุ่นไม่สามารถตรวจจับการมีรถในบริเวณที่กำหนดได้

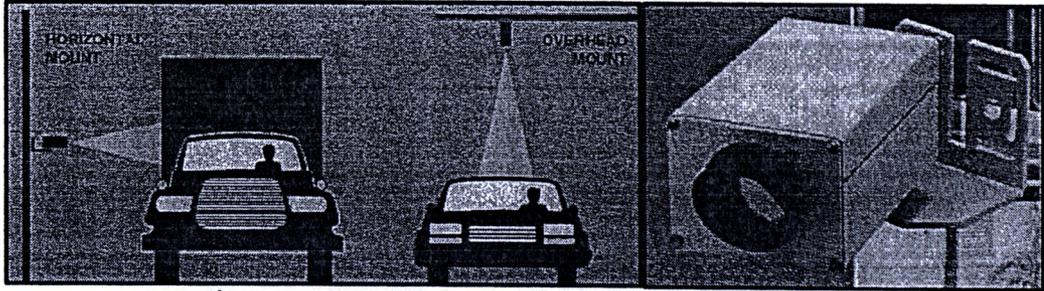
5. Ultrasonic sensors

เริ่มใช้ตั้งแต่ปี 1950s เป็นเครื่องมือตรวจจับการจราจรที่ใช้การส่งสัญญาณเสียงในระดับความถี่ช่วง 25-50 KHZ ใช้ตรวจจับการสะท้อนกลับของคลื่นเสียง

หลักการทำงาน

เป็นการส่งสัญญาณเสียงในระดับความถี่ดังที่กล่าวมาโดยวัดระยะเวลาในการส่งสัญญาณจนถึงระยะเวลานี้เสียงสะท้อนกลับมาซึ่งจะสามารถแสดงผลเป็นระยะห่างระหว่างรถกับสัญญาณเครื่องตรวจจับสามารถให้ข้อมูลได้ดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- แยกประเภทรถโดยใช้ความยาว



รูปที่.2.23 การตรวจวัดยานพาหนะโดยใช้ Ultrasonic sensors

ข้อดี

- ตรวจวัดได้หลายช่องจราจรพร้อมๆกัน
- สามารถตรวจวัดรถที่มีความสูงเกินกำหนดได้
- ใช้งานแพร่หลายในประเทศญี่ปุ่น

ข้อเสีย

- อุณหภูมิและลมมีผลต่อประสิทธิภาพ
- คลื่นช่วงกว้างส่งผลต่อความถูกต้องในการวัด occupancy บน freeway

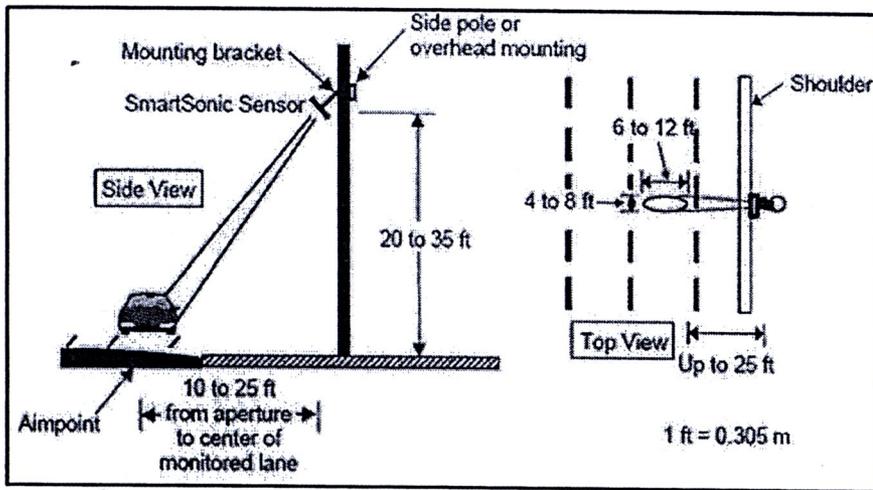
6. Passive acoustic sensors

เป็นเครื่องมือตรวจวัดการจราจรที่ใช้พลังงานเสียง

หลักการทำงาน

ทำงานโดยไม่ต้องมีการส่งสัญญาณเสียงออกไป แต่จะต้องอาศัยการตรวจจับเสียงที่เกิดขึ้นจากยานพาหนะที่เคลื่อนที่ผ่าน เช่น เสียงเครื่องยนต์เสียงล้อรถเมื่อมีเสียงเกิดขึ้นในพื้นที่การตรวจวัดซึ่งจะสามารถจำกัดเป็นช่องทางจราจรได้ โดยเสียงที่เกิดนอกเขตพื้นที่ตรวจวัดจะไม่ส่งผลกระทบต่ออุปกรณ์การวัดจากระดับเสียงที่ตรวจวัดได้จะถูกแปรสัญญาณโดยเครื่องประมวลผลได้ข้อมูลดังนี้

- ปริมาณจราจร
- ความเร็วรถ
- การมีรถบริเวณนั้น (Presence)
- การผ่านของรถบริเวณนั้น (Passage)



รูปที่.2.24 การตรวจวัดยานพาหนะโดยใช้ Passive acoustic sensors

ข้อดี

- เป็นเทคโนโลยีแบบ passive
- ไม่ไวต่อสภาพฝน
- บางรุ่นสามารถใช้ตรวจวัดได้หลายช่องจราจรพร้อมๆกัน

ข้อเสีย

- ความถูกต้องของการนับรถลดลงที่อุณหภูมิต่ำ
- บางรุ่นไม่เหมาะกับสภาพการจราจรติดขัด



2.9.3 เทคโนโลยีการตรวจวัดการจราจรที่ใช้ในประเทศไทย

1. Inductive loop detector: ทล, กทม, สนข
2. Video image processor: ทล, สนข, กทพ, ทช
3. Magnetic sensor: ทล
4. Microwave radar: สนข (ระหว่างดำเนินการ)
5. Ultrasonic sensor: สตช (ระหว่างดำเนินการ)

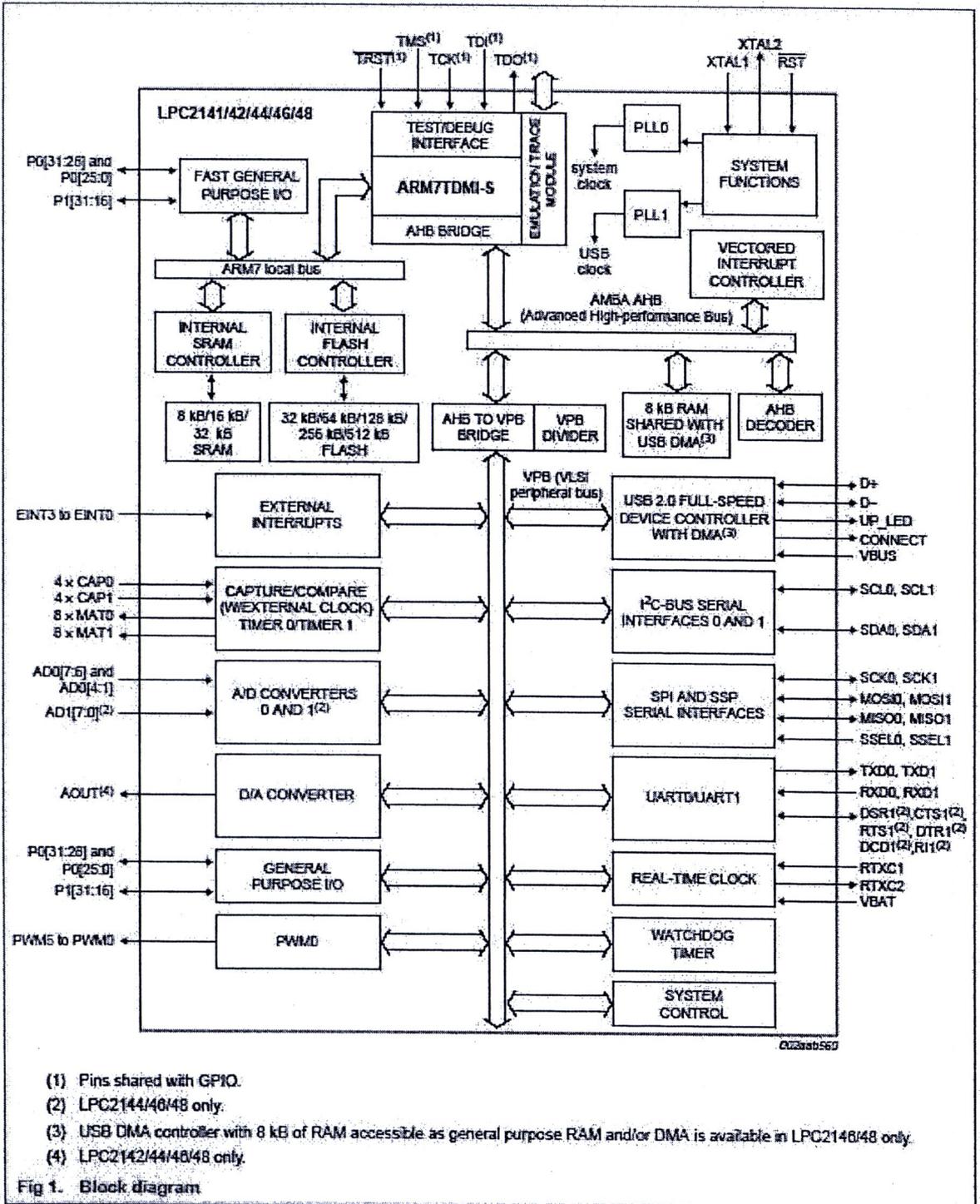
2.10 ไมโครคอนโทรลเลอร์ ARM7

2.10.1 ไมโครคอนโทรลเลอร์ ARM7 Philips LCP2148

ความสามารถของไมโครคอนโทรลเลอร์ Philips LCP2148 มีดังนี้

- ไมโครคอนโทรลเลอร์ขนาด 16/32 บิต ARM7TDMI-S มี LQFP 64 ขา
- หน่วยความจำ Static RAM มีขนาด 40kB
- หน่วยความจำ Flash Program Memory มีขนาด 512kB อยู่ภายในชิปที่สามารถ ลบ/เขียนซ้ำได้ถึง 10000 ครั้ง โปรแกรมชิปสามารถได้ทั้งที่ผ่าน In-System Programming (ISP) และ In-Application Programming (IAP) โดยใช้ซอฟต์แวร์ boot-loader ที่อยู่ภายในชิปตัวควบคุมอุปกรณ์ USB 2.0 Full-speed โดยมี ARM สำหรับ endpoint ขนาด 8 kB สำหรับการติดต่อแบบ DMA วงจรแปลงแอนะล็อกเป็นดิจิทัลความละเอียด 10 บิต จำนวน 2 ชุด ที่รับอินพุตได้ถึง 14 อินพุต โดยมีเวลาในการแปลงค่าต่ำถึง 2.44 μ s วงจรแปลงดิจิทัลเป็นแอนะล็อกความละเอียด 10 บิต 1 ตัว วงจรไทมเมอร์ขนาด 32 บิต 2 ชุด (มี 4 capture และ 4 compare channel) PWM (Pulse width modulation) 6 เาต์พุท โมดูลนาฬิกาเวลาจริง (Real Time Clock) ที่สามารถต่อกับคริสตอลความถี่ 32 kHz และแบตเตอรี่ภายนอกได้ และวอชด็อกซ์ (Watchdog)
- วงจรสื่อสารอนุกรม UART(16C550) จำนวน 2 ชุด วงจรสื่อสารอนุกรม I²C ความเร็วสูง (400Kbits/s) วงจรสื่อสารอนุกรม SPI และ SSP มีวงจร Phase lock loop ภายในเพื่อคุณค่าให้สัญญาณนาฬิกาภายในทำงานที่ความถี่สูงสุด 60 MHz Vectored Interrupt Controller ที่สามารถกำหนดลำดับความสำคัญ และกำหนดแอดเดรสของเวกเตอร์ได้ใช้กับแหล่งจ่ายไฟชุดเดียวขนาด 3.0 V และ 3.6V (3.3 \pm 10%)
- มี I/O pin อเนกประสงค์ที่สามารถใช้กับระดับแรงดัน 5 V ได้สูงสุด 45 ขา โดยสามารถจัดเป็นขาอินเทอร์รัปต์จากภายนอกได้สูงสุด 21 ขา มีโหมดประหยัดพลังงาน 2 โหมดได้แก่ Idle และ Power-down

Block Diagram LPC2148



รูปที่.2.25 Block Diagram LPC2148

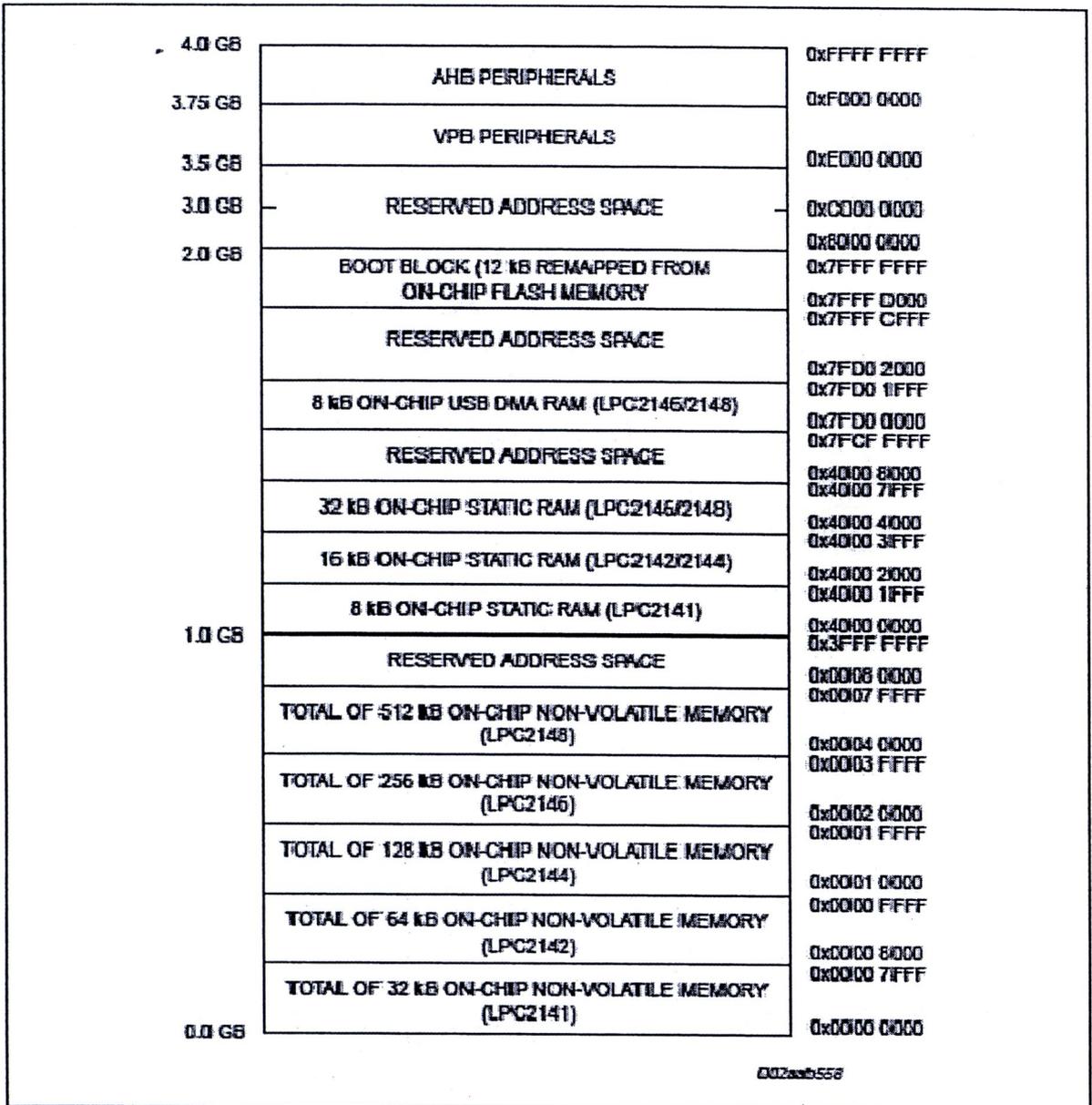
จากรูปที่.2.25 เป็นไมโครโปรเซสเซอร์ ARM7-TDMI-S ซึ่งเป็นปัจจัยหลัก ด้านซ้ายมือเป็นส่วน ของ AMR7 Local Bus ที่ใช้ในการติดต่อกับหน่วยความจำแบบ Flash ที่ใช้เก็บโปรแกรมและหน่วยความจำ SRAM ที่ใช้เก็บข้อมูล ส่วนที่ใช้ในการติดต่อกับหน่วยความจำภายนอกมีการติดต่อผ่านบัส AMBA AHB (Advanced High-performance Bus) ซึ่งใน LPC2148 ไม่สามารถต่อกับหน่วยความจำภายนอกได้

ในการติดต่อกับอุปกรณ์ที่เกี่ยวข้อง เช่น GPIO, I²C, SPI, UART ฯลฯ จะติดต่อผ่านบัส VPB (VLSI peripheral BUS) ซึ่ง VPB บัสต่อกับ AHB บัส ผ่าน AHB to VPB Bridge โดยสามารถปรับลดค่าความถี่ของ VPB บัสให้ทำงานช้ากว่าความถี่ของซีพียูได้เพื่อให้ทำงานร่วมกับอุปกรณ์เสริมต่างๆ ที่มีความเร็วต่ำกว่าได้

การจัดหน่วยความจำของ LPC2148

เนื่องจาก ARM7 เป็นซีพียูขนาด 32 บิต ที่มีขาแอดเดรสต่อกับหน่วยความจำ จำนวน 32 ขา ทำให้สามารถอ้างถึงหน่วยความจำได้ 4 GB ($2^{32} = 4\text{GB}$) อุปกรณ์หลักของ ARM7TDMI จะมีสถาปัตยกรรมแบบ Von Neumann ที่ใช้บัสขนาด 32 บิต ชุดเดียวกันสำหรับตัวคำสั่งของโปรแกรม และข้อมูล โดยมีแค่คำสั่ง load, store, swap เท่านั้น ที่ใช้การเรียกข้อมูลที่เก็บในหน่วยความจำ การติดต่อกับคอมพิวเตอร์อินพุต หรือ เอาต์พุตก็จะใช้คำสั่งเดียวกันกับการใช้คำสั่งจัดการเกี่ยวกับหน่วยความจำในไมโครคอนโทรลเลอร์ LPC2148 ได้จัดสรรหน่วยความจำดังรูป 2.26 เมื่อซีพียูถูกรีเซ็ตจะเริ่มต้นทำงานที่หน่วยความจำ 0x000 0000 จากหน่วยความจำทั้งหมด 4GB ได้แบ่งออกเป็น 4 ส่วน ดังนี้

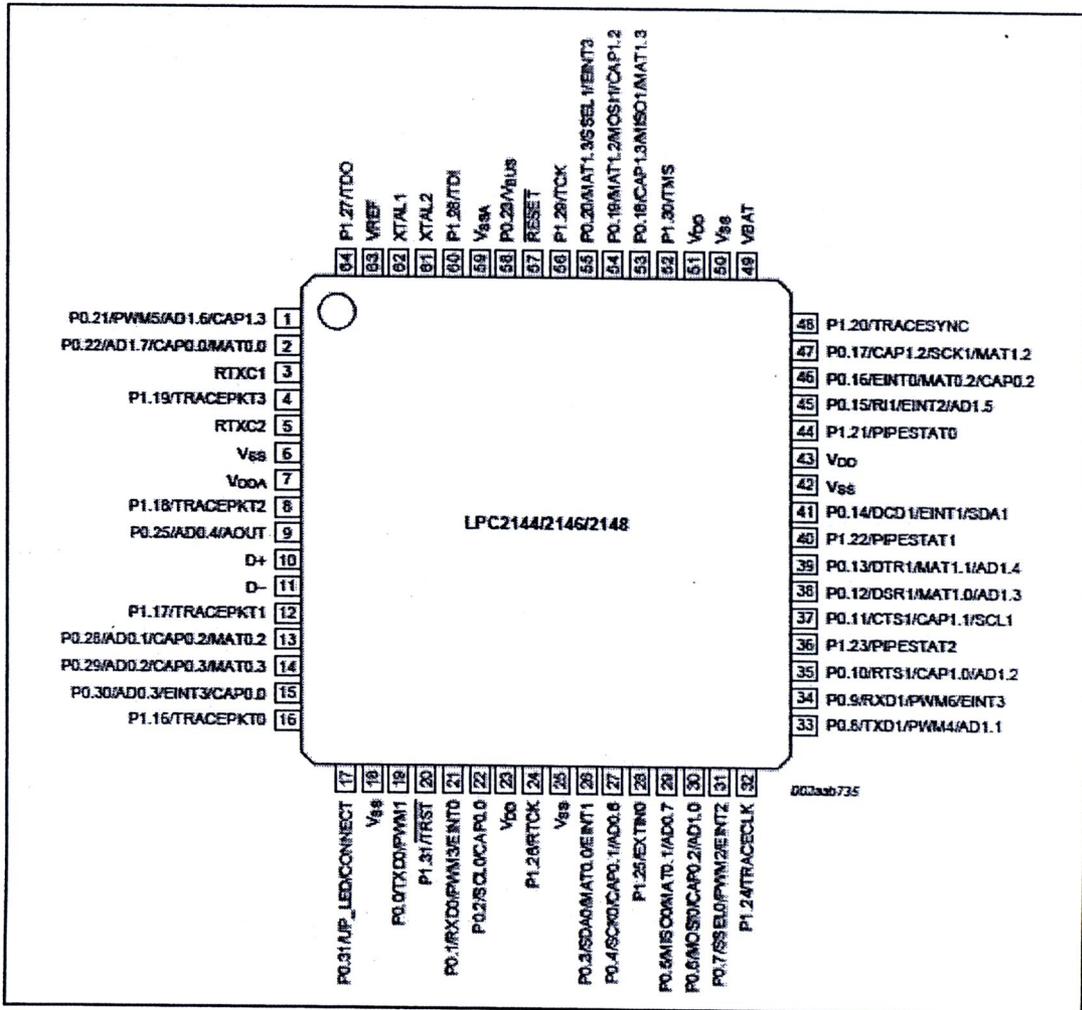
- 1 GB แรก (แอดเดรส 0x0000 0000 – 0x3FFF FFFF) จัดเป็นส่วนของหน่วยความจำสำหรับเก็บโปรแกรม คือ หน่วยความจำ Flash memory ขนาด 512kB ซึ่งมีแอดเดรส 0x0000 0000 – 0x0007 FFFF
- หน่วยความจำในช่วง 1 GB - 2 GB (แอดเดรส 0x4000 0000 – 0x7FFF FFFF) จัดเป็นส่วนของหน่วยความจำ ARM จะเป็น SRAM ขนาด 40kB โดยแบ่งออกเป็น 2 ส่วน
- ส่วนที่ 1 คือ 32kB แรกอยู่ที่แอดเดรส 0x4000 0000 – 0x4000 7FFFF
- ส่วนที่ 2 คือ 8kB เป็นหน่วยความจำใช้สำหรับ USB โดยมีแอดเดรส 0x7FD0 0000- 0x7FD0 1FFF ในกรณีที่ไมใช้การติดต่อกับ USB สามารถนำหน่วยความจำ ARM ส่วนนี้มาใช้เป็น ARM สำหรับงานทั่วไปได้ หน่วยความจำที่ใกล้ 2 GB จะเป็นส่วนของ Boot Block ขนาด 12 kB ซึ่งเป็นโปรแกรมที่ทำงานเพื่อเขียนโปรแกรมลงในหน่วยความจำ Flash
- หน่วยความจำ 2GB - 3 GB (แอดเดรส 0x8000 0000 – 0xDFFF FFFF) สงวนไว้สำหรับต่อกับหน่วยความจำภายนอก ซึ่งไม่ได้ใช้งาน
- หน่วยความจำ 2GB - 3GB จะเป็นพื้นที่สำหรับติดต่อกับอุปกรณ์อื่นๆ ที่อยู่ในชิป โดยแบ่งเป็น อุปกรณ์ที่ต่อกับ VPB บัส จะติดต่อกับหน่วยความจำช่วง 0xE000 00000 – 0xEFFF FFFF ถ้าเป็นอุปกรณ์ที่ติดต่อกับผ่านทาง AHB จะมีช่วงแอดเดรส 0xF000 0000 – 0xFFFF FFFF ไม่สามารถติดต่อกับหน่วยความจำภายนอก จึงไม่ได้ใช้หน่วยความจำส่วนนี้



รูปที่.2.26 memory map LPC 2148

การจัดขาของ LPC2148

มีจำนวนขาทั้งหมด 64 ขา โดยมีการจัดขาแสดงดังรูปที่.2.27 ดังนี้



รูปที่.2.27 LPC2148 pinning

หลังจากการรีเซตขาพอร์ตทั้งหมดจะถูกกำหนดให้ทำหน้าที่เป็นอินพุตขาแต่ละขา จะมีการทำงาน เช่นขาที่ 14 สามารถทำหน้าที่ได้ 4 หน้าที่ คือ

- เป็น P0.29/AD0.2/CAP0.3/MAT0.3
- ถ้าเป็นอินพุตเอาต์พุตจะเรียกว่า General Purpose Input Output : GPIO ก็คือขา P0.23.
- ถ้าใช้วงจรแปลงแอนะล็อกเป็นดิจิตอลขานี้คือ AD0.2 เป็นขาอินพุตสำหรับสัญญาณแอนะล็อก ADC0 อินพุตสอง
- ถ้าใช้งาน Time0 ขานี้จะเป็น CAP0.3 คือ Capture input for timer 0, Channel3 หรือ MAT0.3 ซึ่งเป็น Match output for timer0, Channel3

2.10.2 ฮาร์ดแวร์ และซอฟต์แวร์ของไมโครคอนโทรลเลอร์

ฮาร์ดแวร์ และซอฟต์แวร์ที่ใช้ในการเขียนไมโครคอนโทรลเลอร์ Philips LPC 2148 เป็นไมโครคอนโทรลเลอร์ตระกูล ARM7TDMI-S ฮาร์ดแวร์ที่ใช้ประกอบด้วย แผงวงจร CP-JR ARM7 USB-LPC2148 EXP บริษัท อีทีที จำกัด ส่วนของซอฟต์แวร์คอมไพเลอร์เพื่อแปลภาษาซีเป็นภาษาเครื่องของ LPC2148 จะใช้ชุดพัฒนา RealView Microcontroller Development Kit Version 3.02a โดยใช้คอมไพเลอร์ CARMของบริษัท Keil เป็นรุ่น Evaluation ที่อนุญาตให้ดาวน์โหลดมาใช้ทดสอบโปรแกรมฟรี โดยมีข้อจำกัดว่าโปรแกรมที่แปลเป็นภาษาเครื่องแล้ว จะมีขนาดไม่เกิน 16kB หลังจากคอมไพล์เป็นภาษาเครื่องจะทำการลงโปรแกรมชิปไมโครคอนโทรลเลอร์ด้วยโปรแกรม LPC2000 Flash Utility ของบริษัท Philips ซึ่งอนุญาตให้ใช้งานได้ฟรี

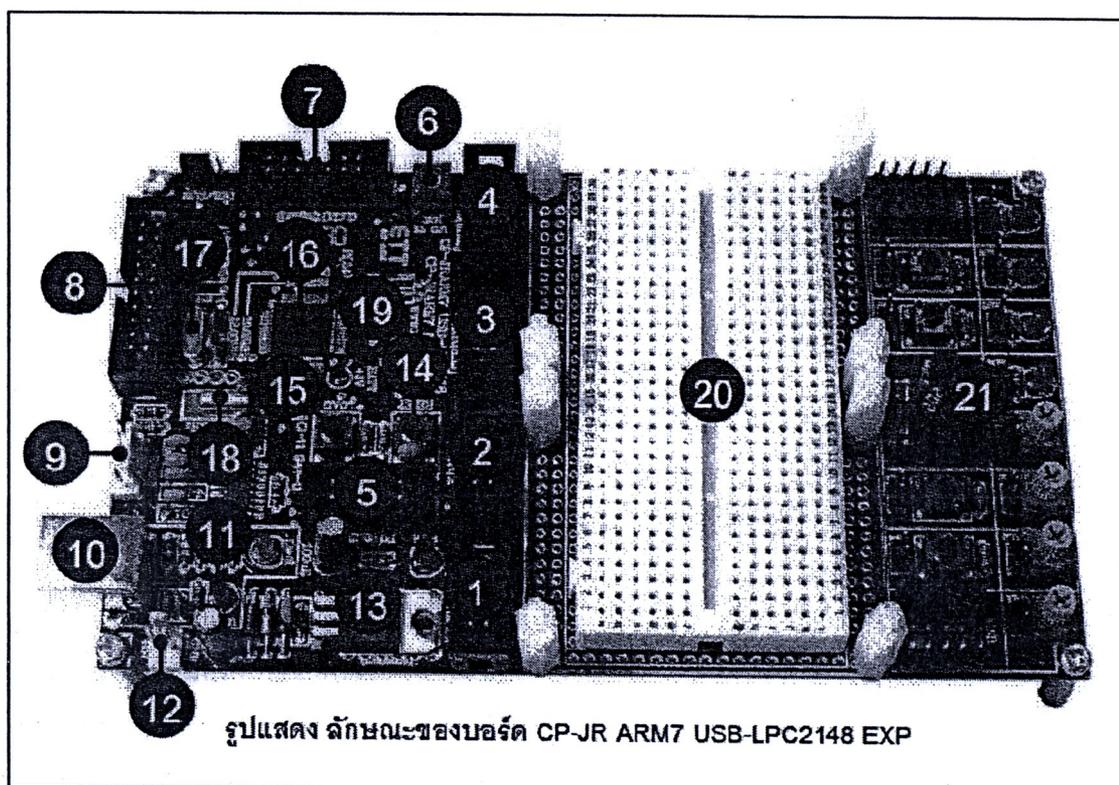
คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์ CP-JR ARM7 USB-LPC2148

1. ใช้ MCU ตระกูล ARM7TDMI-S เบอร์ LPC2148 ของ Philips ซึ่งเป็น MCU ขนาด 16/32-Bit
2. ใช้ Crystal 12.00MHz โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 60MHz เมื่อใช้งานร่วมกับ Phase-Locked Loop (PLL) ภายในตัว MCU เอง
3. รองรับการโปรแกรมแบบ In-System Programming (ISP) และ In-Application Programming (IAP) ผ่านทาง On-Chip Boot-Loader Software ทางพอร์ต UART0 (RS232)
4. Power Supply ใช้แรงดันไฟฟ้า +5VDC โดยใช้ขั้วต่อแบบ CPA-2PIN จากภายนอก หรือ ใช้พลังงานจาก USB Port ได้ (ในกรณีใช้กระแสไม่เกิน 500mA)
5. ภายใน MCU มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 512KB, Static RAM ขนาด 40KB
6. มีวงจร USB มาตรฐาน 2.0 แบบ Full Speed ภายในตัว (USB Function มี 32 End Point)
7. จำนวน GPIO สูงสุดถึง 47 I/O Pins สามารถเชื่อมต่อกับระบบ I/O ที่เป็นสัญญาณ 5V ได้ ซึ่งขาสัญญาณ GPIO จะมีการใช้งานร่วมกันของ Function อื่นๆอีกดังนี้วงจรสื่อสารอนุกรมแบบ SPI จำนวน 2 ช่อง และ วงจรสื่อสารอนุกรมแบบ I2C จำนวน 2 ช่อง วงจร ADC ขนาด 10 Bit จำนวน 14 ชุด และ วงจร DAC ขนาด 10 Bit จำนวน 1 ชุด วงจร UART แบบ Full-Duplex จำนวน 2 ช่อง คือ UART-0 มาตรฐาน 4 Pin ETT เป็นสัญญาณระดับ RS232 Level และ UART-1 เป็นสัญญาณระดับ TTL Level Timer 32-bit จำนวน 2 ช่อง (4 Input Capture / 4 Output Compare), 6-Channel PWMOutput, Watchdog Timer และ Real Time Clock
8. มีวงจรเชื่อมต่อกับ Character LCD โดยใช้วงจรการเชื่อมต่อแบบ 4 บิต จาก GPIO1[25..31] พร้อมวงจรปรับความสว่างหน้าจอ
9. มีวงจรเชื่อมต่อกับ JTAG ARM ขนาด 20 Pin มาตรฐาน เพื่อทำการ Debug แบบ Real Time ได้
10. มีวงจรทดลองขั้นพื้นฐานสำหรับสนับสนุนการใช้งานและทดลองเรียนรู้ ขั้นพื้นฐานอย่างครบถ้วน จัดเตรียมไว้ภายในบอร์ด (ติดตั้งไว้เฉพาะรุ่น CP-JR ARM7 USB-LPC2148 EXP) ซึ่ง ได้แก่
 - LED Output แบบ Sink Current สำหรับแสดงสถานะของ Output จำนวน 4 ชุด
 - Push Button Switch แบบ Active Logic "0" สำหรับทดสอบ Input Logic จำนวน 4 ชุด
 - Volume ปรับค่าแรงดัน 0-3.3V สำหรับทดสอบการทำงานของ ADC จำนวน 4 ชุด

- ชุดกำเนิดสัญญาณเสียง Mini Speaker สำหรับทดสอบการเสียงแบบต่างๆ จำนวน 1 ชุด
- แผงต่อวงจร Project Board รุ่น AD-100 ขนาด 360 จุด สำหรับเป็นพื้นที่ต่อทดลองวงจรขนาด
- เล็กๆ เพื่อใช้งานร่วมกับ CPU ได้อย่างอิสระ
- จุดต่อแหล่งจ่ายไฟ +3.3V และ GND สำหรับเชื่อมต่อไปยังวงจรภายนอกอื่นๆ .

11. ทนอุณหภูมิใช้งานระหว่าง -40 to +85°C

โครงสร้างบอร์ดไมโครคอนโทรลเลอร์ CP-JR ARM7 USB-LPC2148



รูปที่.2.28 ลักษณะโครงสร้างของบอร์ด CP-JR ARM7 USB-LPC2148 / EXP

จากรูปที่.2.28 รายละเอียดของการจัดตำแหน่งส่วนต่าง ๆ ของบอร์ดไมโครคอนโทรลเลอร์ มีดังนี้

หมายเลข 1 คือ ขั้วต่อ Port1[16..23] จำนวน 8 บิต

หมายเลข 2 คือ ขั้วต่อ Port0[2..7] จำนวน 6 บิต

หมายเลข 3 คือ ขั้วต่อ Port0[8..15] จำนวน 8 บิต

หมายเลข 4 คือ ขั้วต่อ Port0[16..23] จำนวน 8 บิต

หมายเลข 5 คือ ขั้วต่อ Port0[25..31] จำนวน 7 บิต

หมายเลข 6 คือ ตัวต้านทานสำหรับปรับค่าความสว่าง (Contrast) ของหน้าจอ LCD

หมายเลข 7 คือ ขั้วต่อ Character LCD โดยใช้สัญญาณ Port1[25..31] ในการเชื่อมต่อ

หมายเลข 8 คือ ขั้วต่อ JTAG โดยใช้สัญญาณ Port1[26..31] และ Reset ของ CPU

หมายเลข 9 คือ ขั้วต่อ RS232 สำหรับใช้งาน และ Download Hex File ให้ CPU

หมายเลข 10 คือ ขั้วต่อ USB สำหรับเชื่อมต่อกับ USB Hub รุ่น 2.0

หมายเลข 11 คือ LED แสดงสถานะ Power จาก USB และ สถานะของการเชื่อมต่อกับ USB

หมายเลข 12 คือ ขั้วต่อ Power ขนาด +5VDC และ GND เพื่อจ่ายให้กับบอร์ด

หมายเลข 13 คือ LED แสดงสถานะของแหล่งจ่ายไฟ Power ของบอร์ด

หมายเลข 14 คือ Switch RESET สำหรับสั่ง Reset การทำงานของ CPU

หมายเลข 15 คือ Switch LOAD ใช้ร่วมกับ Switch RESET เพื่อ Download Hex ให้ CPU

หมายเลข 16 คือ CPU เบอร์ LPC2148 ของ Philips ซึ่งเป็น CPU ประจำบอร์ด

หมายเลข 17 คือ Crystal 12.00 MHz สำหรับป้อนให้เป็นสัญญาณนาฬิกาของ LPC2148

หมายเลข 18 คือ Crystal 32.768 KHz สำหรับ Real Time Clock (RTC) ในตัวของ LPC2148

หมายเลข 19 คือ จุดเชื่อมต่อ ลังถ่าน Battery ขนาด +3V (อยู่ด้านใต้บอร์ด) สำหรับต่อให้กับ RTC เพื่อเก็บรักษาค่าเวลาของ RTC ในกรณีที่ไม่ได้จ่ายไฟเลี้ยงให้กับบอร์ด

หมายเลข 20 คือ แผง Project Board รุ่น AD-100 ขนาด 360 จุด สำหรับต่อวงจร (มีติดตั้งไว้เฉพาะในรุ่น CP-JR ARM7 USB-LPC2148 EXP)

หมายเลข 21 คือ ส่วนของวงจร I/O พื้นฐาน สำหรับใช้ทดสอบการทำงานของ Function ต่างๆของ CPU (มีติดตั้งไว้เฉพาะในรุ่น CP-JR ARM7 USB-LPC2148 EXP) โดยมีรายละเอียดดังนี้คือ

LED สำหรับแสดงผลการทำงานของ Output แบบ Sink Current มีทั้งหมด 4 ชุด

Push Button Switch สำหรับกำเนิด Logic เพื่อทดสอบการทำงานของ Input มีทั้งหมด 4 ชุด

Volume สำหรับปรับค่าแรงดัน 0-3.3V เพื่อใช้ทดสอบการทำงานของ A/D มีทั้งหมด 4 ชุด

Mini-Speaker สำหรับใช้กำเนิดเสียง เช่น Beep จำนวน 1 ชุด

จุดต่อแหล่งจ่ายไฟ +3.3V และ GND

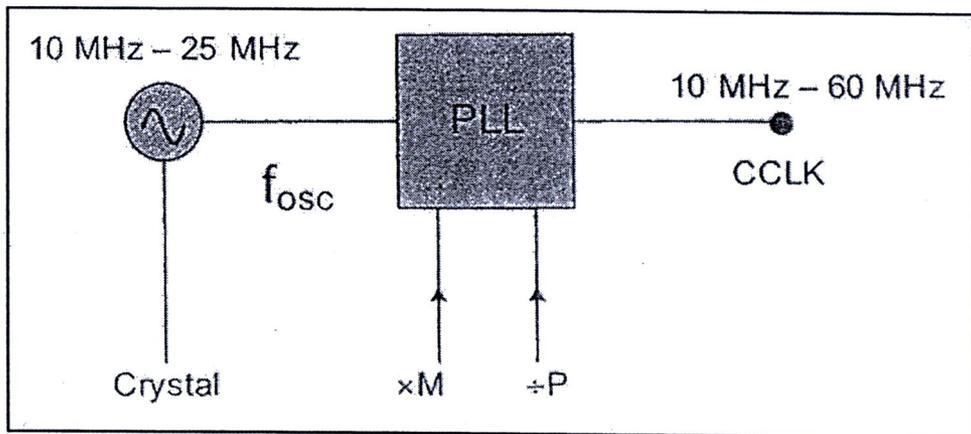
2.10.3 การกำหนดค่าเริ่มต้นให้กับไมโครคอนโทรลเลอร์ ARM7

การกำหนดค่าเริ่มต้นให้ก่อนโดยค่าที่ต้องการกำหนด คือ ส่วนของหน่วยความจำ RAM ที่ใช้เป็น stack pointer การกำหนดค่าของ stack pointer ต้องเขียนคำสั่งเป็นโปรแกรมภาษาแอสเซมบลีภายในของไมโครคอนโทรลเลอร์ ARM มีวงจร Phase Lock Loop : PLL สำหรับคูณค่าความถี่ของสัญญาณนาฬิกาจากภายนอก เพื่อให้ตัวไมโครคอนโทรลเลอร์สามารถทำงานที่ความถี่สูงๆ ได้โดยการกำหนดค่าตัวคูณความถี่ สำหรับ LPC2148 ภายในจะมีวงจร PLL อยู่สองวงจรโดยตั้งชื่อเป็น PLL0 และ PLL1 โดย PLL0 ใช้คูณค่าความถี่ของสัญญาณนาฬิกาภายนอกให้กับซีพียู ARM 7 และ PLL1 ที่ทำหน้าที่คูณความถี่ให้กับวงจรส่วนของ USB โดยต้องคูณค่าให้ได้ความถี่ 48 MHz หลังจากที่ไมโครคอนโทรลเลอร์ทำงานที่ความถี่สูงแล้วอุปกรณ์ประกอบ(Peripheral)ต่างๆ ที่อยู่ภายนอกไมโครคอนโทรลเลอร์ เช่น UART , I²C ฯลฯ จะทำงานไม่ทัน ดังนั้นภายในไมโครคอนโทรลเลอร์ ARM7 จะมีวงจรหารความถี่เพื่อลดความถี่ที่ป้อนให้อุปกรณ์ต่างๆหน่วยความจำแบบ Flash ภายในของ LPC2148 มีการทำงานช้ากว่าตัวซีพียู ดังนั้น

ภายในของ LPC2148 จะมีวงจร Memory Accelerator Module (MAM) เพื่อทำหน้าที่เร่งความเร็วของการติดต่อกับหน่วยความจำแฟลช(Flash)ภายในชิป

วงจร Phase Lock Loop : PLL

วงจรคริสตัลลออสซิลเลเตอร์ภายในของไมโครคอนโทรลเลอร์ตระกูล LPC2000 ใช้ได้กับคริสตัลค่าความถี่ 1 MHz - 30MHz เอาต์พุตของวงจรเรียกว่า f_{osc} ส่วนความถี่สัญญาณนาฬิกาของไมโครคอนโทรลเลอร์จะมีชื่อเรียกว่า CCLK โดยปกติถ้าไม่ได้เปิดใช้วงจร PLL ไมโครคอนโทรลเลอร์ ARM7 จะนำความถี่ f_{osc} ไปใช้กับไมโครคอนโทรลเลอร์ ARM7 เช่นเดียวกัน (กรณีนี้ค่าของ f_{osc} และ CCLK จะมีค่าเท่ากัน)



รูปที่.2.29 ผังวงจร PLL ภายในไมโครคอนโทรลเลอร์ ARM 7

วงจร PLL จะรับค่าความถี่สัญญาณนาฬิกาจากคริสตัลลออสซิลเลเตอร์ที่ถูกควบคุมค่าโดยคริสตัลภายนอก แล้วนำมาคูณด้วยค่าคงที่ M ให้เป็นความถี่ 10 MHz - 60 MHz โดยใช้วงจร Current Controlled Oscillator (CCO) ทำหน้าที่คูณความถี่ ค่าตัวคูณ M จะมีค่าตั้งแต่ 1 ถึง 32 วงจร CCO ทำงานในช่วงความถี่ 156 MHz - 320MHz ดังนั้นภายในรูปของ CCO จะต้องมียังวงจรหารค่าอีกหนึ่งตัวเพื่อให้เอาต์พุตของ PLL สร้างความถี่ให้ได้ค่าตามที่ต้องการ ค่าตัวหาร คือ P กำหนดค่าได้เป็น 2,4,8 หรือ 16 เมื่อไมโครคอนโทรลเลอร์ถูกรีเซ็ต วงจร PLL จะถูกปิดการทำงาน ซึ่งต้องให้ซอฟต์แวร์สั่งเปิดการทำงานของโปรแกรม ตัวโปรแกรมจะต้องกำหนดค่าตัวคูณ M ตัวหาร P และกระตุ่นการทำงานของ PLL และรอให้ PLL ล็อกความถี่ได้ก่อน จึงจะสั่งต่อให้ PLL เป็นสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ ARM7 ค่าเวลาในการเซต PLL เป็น $10\mu s$ เอาต์พุตของ PLL คำนวณได้ดังนี้

$$CCLK = M * f_{osc} \quad \text{หรือ} \quad CCLK = f_{cco}/(2 * P)$$

ค่าความถี่ CCO คำนวณได้ดังนี้

$$f_{cco} = CCLK * 2 * P \quad \text{หรือ} \quad f_{cco} = f_{osc} * 2 * M * P$$

ค่า P เมื่อนำไปคูณแล้ว f_{cco} จะมีเงื่อนไขดังนี้

$$156\text{MHz} < f_{\text{cco}} < 320\text{MHz}$$

แผงวงจร CP-JR ARM7 USB-LPC2148 EXP ใช้คริสตัลความถี่ 12.00MHz ดังนั้น

$f_{\text{osc}} = 12\text{MHz}$ ต้องการคูณให้ได้ความถี่สูงสุดไม่เกิน 60MHz จะได้ตัวคูณ $M = 5$ ดังนั้น

$$\text{CCLK} = 5 * 12.00 = 60.00\text{MHz}$$

ในการกำหนดค่า PLL1 เพื่อกำเนิดสัญญาณนาฬิกาสำหรับวงจร USB จะต้องคำนวณ ค่า M ที่นำไปคูณกับความถี่ของสัญญาณนาฬิกาให้ได้ความถี่ 48 MHz จาก แผงวงจร CP-JR ARM7 USB-LPC2148 EXP จะคำนวณได้ $M = 48\text{MHz} / f_{\text{osc}} = 48\text{MHz} / 12\text{MHz} = 4$

การหาค่าของ P ให้ใช้ตามค่าที่ได้จากกรณีของ PLL0 คือให้ค่า $P = 2$ ได้

$f_{\text{cco}} = 48\text{MHz} * 2 * 2 = 192\text{MHz}$ ซึ่งตรงตามเงื่อนไขในการกำหนดค่าการทำงานของวงจร PLL จะสั่งผ่านทางรีจิสเตอร์ ที่เกี่ยวข้องกับ PLL ดังแสดงในตารางที่ 2.5

ตารางที่ 2.5 PLL Register

ชื่อทั่วไป	ความหมาย	ติดต่อ	ค่าหลังรีเซต	ชื่อและแอดเดรสของ PLL0	ชื่อและแอดเดรสของ PLL1
PLLCON	PLL Control Register เก็บค่ารีจิสเตอร์สำหรับการอัปเดตค่า PLL control bit ค่าที่เขียนให้ยังไม่มีผลจนกว่าจะเขียน PLL feed sequence ที่ถูกต้องให้	อ่าน/เขียน	0	PLL0CON 0xE01FC080	PLL1CON 0xE01FC0A0
PLLCFG	PLL Configuration Register เก็บค่าสำหรับอัปเดตค่าของ PLL Configuration ค่าที่เขียนให้ยังไม่มีผลจนกว่าจะเขียน PLL feed sequence	อ่าน/เขียน	0	PLL0CFG 0xE01FC084	PLL1CFG 0xE01FC0A4
PLLSTAT	PLL Configuration อ่านค่าสถานะของ PLL Control และ configuration	อ่าน/เขียน	0	PLL0STAT 0xE01FC088	PLL1STAT 0xE01FC0A8
PLLFEED	PLL Feed Register เป็นการสั่งโหลดค่าที่เก็บใน	อ่าน/เขียน	ไม่มี	PLL0FEED	PLL1FEED

ชื่อทั่วไป	ความหมาย	ติดต่อ	ค่าหลังรีเซ็ต	ชื่อและแอดเดรสของ PLL0	ชื่อและแอดเดรสของ PLL1
	PLLCON และ PLLCFG เพื่อสั่งให้ทำงานตามค่าที่กำหนดให้			0xE01FC08C	0xE01FC0AC

การกำหนดค่าของ P , M กำหนดที่รีจิสเตอร์ PLLCFG ซึ่งแสดงรายละเอียดแต่ละบิตของรีจิสเตอร์ PLLCFG ได้ดังตาราง 2.6

ตารางที่ 2.6 PLLCFG Register

ค่าบิตของ PLLCFG	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
4:0	MSEL 4:0	ค่าตัวคูณ M ของวงจรร PLL 00000 คือ M=1 , 00001 คือ M=2,.....,11110 คือ M =31 และ 11111 คือ M=32	0
6:5	PSEL 1:0	ค่าตัวหาร P ของวงจรร PLL 00 คือ P=1 , 01 คือ P=2 10 คือ P=2 , 11 คือ P=4	0
7	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตนี้	ไม่ได้กำหนด

เมื่อกำหนดค่าตัวหาร P และตัวคูณ M แล้วคำสั่งให้วงจรร PLL ทำงานและต่อ PLL เป็นสัญญาณนาฬิกาหลักได้ที่รีจิสเตอร์ PLLCON ซึ่งมีค่าดังตารางที่ 2.7

ตารางที่ 2.7 PLLCON Register

ค่าบิตของ PLLCON	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
0	PLLE	PLL Enable สั่งให้ PLL ทำงาน โดยเขียนค่าเป็น 1 และต้องเขียนค่า PLLFEED ที่เหมาะสมด้วย	0
1	PLL C	PLL Connect เมื่อ PLLC และ PLLE มีค่าเป็น 1 และหลังจากเขียนค่า PLLFEED ที่เหมาะสมจะเป็นการสั่งให้ต่อ PLL เป็นสัญญาณนาฬิกาหลักของวงจรร	0
7:2	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตนี้	ไม่ได้กำหนด

ค่าแต่ละบิตของรีจิสเตอร์ PLLSTAT เพื่อดูสถานะของ PLL แสดงได้ในตารางที่ 2.8
ตารางที่ 2.8 ค่าแต่ละบิตของรีจิสเตอร์ PLLSTAT

ค่าบิตของ PLLSTAT	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
4:0	MSEL 4:0	อ่านค่าว่าปัจจุบันวงจร PLL ใช้ ค่าตัวคูณความถี่เท่ากับเท่าไร	0
6:5	PSEL 1:0	อ่านค่าว่าปัจจุบันวงจร PLL ใช้ ค่าตัวหารความถี่เท่ากับเท่าไร	0
7	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตนี้	ไม่กำหนด
8	PLLE	อ่านค่าว่าปัจจุบันบิต PLLE มีค่า เป็น 0 หรือ 1	0
9	PLLC	อ่านค่าว่าปัจจุบันบิต PLLC มีค่า เป็น 0 หรือ 1	0
10	PLOCK	ถ้าเป็น 0 แสดงว่า PLL ยังไม่ สามารถล็อกค่าความถี่ ถ้าเป็น 1 แสดงว่าสามารถล็อกค่าความถี่ ตามที่กำหนดได้แล้ว	0
15:11	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตนี้	ไม่กำหนด

เราสามารถกำหนดโหมดการทำงานของวงจร PLL ได้จากบิต PLLC และ PLLE ของรีจิสเตอร์
PLLCON ได้ดังตารางที่ 2.9

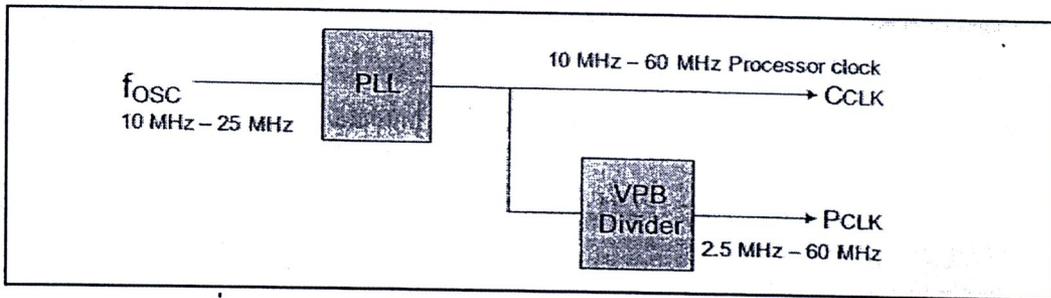
ตารางที่ 2.9 โหมดการทำงานของวงจร PLL

บิต PLLC	บิต PLLE	การทำงานของ PLL
0	0	PLL ไม่ทำงาน ไม่ได้ต่อเข้ากับไมโครคอนโทรลเลอร์
0	1	PLL ทำงานแต่ไม่ได้ต่อกับไมโครคอนโทรลเลอร์
1	0	เหมือนกับกรณีค่า 00 เพื่อป้องกันไม่ให้ต่อ PLL เข้ากับระบบ ถ้ายังไม่สั่งให้มันทำงาน
1	1	PLL ทำงาน และต่อเป็นสัญญาณนาฬิกาหลัก

การกำหนดค่าความถี่ให้กับ VLSI Peripheral Bus

เมื่อกำหนดให้กับวงจร PLL ภายในไมโครคอนโทรลเลอร์ ARM7 ทำงานสร้างความถี่สูง และ
ต่อให้เป็นสัญญาณนาฬิกาของโปรเซสเซอร์แล้ว อุปกรณ์ประกอบต่างๆ เช่น UART, GPIO, I²C ที่ต่อกับบัส

VPB จะทำงานไม่ทัน ดังนั้นภายในไมโครคอนโทรลเลอร์ ARM7 จะมีวงจรหารลดความถี่ของ CCLK ให้เป็นความถี่ของ VPB Bus ที่เรียกว่า PCLK โดยบล็อกไดอะแกรมภายในแสดงดังรูป 2.30



รูปที่. 2.30 Block Diagram วงจรกำเนิดสัญญาณภายใน ARM7

รีจิสเตอร์ที่เกี่ยวข้องคือ VPBDIV ที่อยู่แอดเดรส 0xE10FC100 โดยสามารถอ่านและเขียนได้ ค่าของรีจิสเตอร์ VPBDIV จะมีค่า 2 บิตสุดท้าย โดยกำหนดได้ดังตารางที่ 2.10

ตารางที่ 2.10 ค่าแต่ละบิตของรีจิสเตอร์ VPBDIV

ค่าบิตของ VPBDIV	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
1:0	VPBDIV	VPB bus clock = CCLK/4 VPB bus clock = CCLK VPB bus clock = CCLK/2 สงวนไว้	0
7:2	สงวนไว้	ห้ามเขียนค่า 1 ให้บิตเหล่านี้	0

การกำหนดค่าให้กับ Memory Accelerator Module (MAM)

การกำหนดค่าการทำงาน MAM มีรีจิสเตอร์ที่เกี่ยวข้องสองตัว คือ MAMCR, MAMTIM ดังแสดงในตารางที่ 2.11

ตารางที่ 2.11 MAM Register

แอดเดรส	ชื่อ	ความหมาย	ติดต่อ
0xE01FC000	MAMCR	Memory Accelerator Module Control Register ใช้กำหนดโหมดการทำงานของ MAM	อ่าน/ เขียน
0xE01FC004	MAMTIM	Memory Accelerator Module Timing Control ใช้กำหนดจำนวนสัญญาณนาฬิกาที่ใช้ในการติดต่อกับหน่วยความจำ Flash	อ่าน/ เขียน

การกำหนดค่าโหมดการทำงานของ MAM ผ่านทาง MAMCR ได้ดังตาราง 2.12

ตารางที่ 2.12 การกำหนดค่าให้กับ MAMCR Register

MAMCR bit	ฟังก์ชัน	ความหมาย
1:0	MAN mode control	00-MAM Disable หยุดการทำงานของ MAM 01-MAM partially enable เปิดการทำงานของ MAM บางส่วน 10-MAM fully enable เปิดการทำงานของ MAM สมบูรณ์แบบ 11-สงวนไว้
7:2	สงวนไว้	สงวนไว้ ห้ามเขียนค่า 1 ให้บิตเหล่านี้

การกำหนดค่ารีจิสเตอร์ MAMTIM กำหนดได้ดังตารางที่ 2.13

ตารางที่ 2.13 การกำหนดค่าให้กับ MAMTIM Register

MAMTIM bit	ฟังก์ชัน	ความหมาย
2:0	MAN Fetch Cycle timing	000-สงวนไว้ 001-MAM fetch cycles = 1 * CCLK 010-MAM fetch cycles = 2 * CCLK 011-MAM fetch cycles = 3 * CCLK 100-MAM fetch cycles = 4 * CCLK 101-MAM fetch cycles = 5 * CCLK 110-MAM fetch cycles = 6 * CCLK 111-MAM fetch cycles = 7 * CCLK
7:3	สงวนไว้	สงวนไว้ ห้ามเขียนค่า 1 ให้บิตเหล่านี้

ในการกำหนดค่าเริ่มต้นการทำงานของ MAM เริ่มต้นด้วยการหยุดทำงานของ MAM ด้วยการโหลดค่า 0 ให้กับ MAMCR แล้วจึงกำหนดค่าของ MAMTIM ตามที่ต้องการแล้วจึงเปิดการทำงานของ MAM ด้วยการเขียนค่า 1 หรือ 2 ให้กับ MAMCR

ถ้าซีพียูมีความถี่ CCLK น้อยกว่า 20MHz ให้ใช้ MAMTIM = 1*CCLK ถ้า CCLK มีค่าระหว่าง 20 -40 MHz ให้กำหนด MAMTIM = 2*CCLK ถ้าความถี่มากกว่า 40MHz ให้กำหนดค่า MAMTIM = 3*CCLK

2.10.4. การต่อ GPIO เป็นเอาต์พุต

ภายในไมโครคอนโทรลเลอร์ มีพอร์ตอเนกประสงค์ ขนาด 32 บิต ให้ใช้งาน 2 พอร์ต คือ Port 0 , Port1 โดยให้ Port0 มีขาต่อใช้งานได้ 29 ขา คือ P0.0 –P0.31 โดยไม่มีขา P0.24, P0.26, P0.27 ให้ใช้งาน และขา P0.31 ทำหน้าที่เป็นเอาต์พุตได้อย่างเดียว ส่วน Port1 จะมีต่อให้ใช้งานแค่ 16 ขา คือ P1.16-P1.31 โดยแต่ละขาของพอร์ต P0 และ P1 มีหน้าที่การทำงานได้หลายประเภท ซึ่งต้องกำหนดรีจิสเตอร์ PINSEL

การกำหนดหน้าที่การทำงานของพอร์ต

หลังจากเกิดการรีเซ็ตไมโครคอนโทรลเลอร์ ARM7 วงจรภายในสังรีเซตค่ารีจิสเตอร์ PINSEL ทุกตัวกำหนดค่าให้ Port0 ,Port1 ทุกขาเป็น GPIO และให้ทุกขาเป็นอินพุต

ขาแต่ละขาของ Port0,Port1 มีหน้าที่การทำงานได้หลายหน้าที่ โดยกำหนดการทำงานของ Port ที่รีจิสเตอร์ PINSEL0, PINSEL1 และกำหนดหน้าที่การทำงานของ Port1 ที่รีจิสเตอร์ PINSEL2 โดยรีจิสเตอร์แต่ละตัวที่มีแอดเดรสแสดงในตารางที่ 2.14

ตารางที่ 2.14 รีจิสเตอร์ PINSEL0, PINSEL1, PINSEL2

ชื่อ	ความหมาย	การติดต่อ	แอดเดรส
PINSEL0	Pin-function select register กำหนดการทำงานของ P0.0-P0.15	อ่าน/เขียน	0xE002 C000
PINSEL1	Pin Function select register 1 กำหนดการทำงานของ P0.16-P0.31	อ่าน/เขียน	0xE002 C004
PINSEL2	Pin Function select register 2 กำหนดการทำงานของ P01.0 –P1.31	อ่าน/เขียน	0xE002 C014

ตารางที่ 2.15 ค่าประจำบิตของรีจิสเตอร์ PINSEL0

PINSEL0	ชื่อขา	การทำงาน เมื่อมีค่า 00	การทำงาน เมื่อมีค่า 01	การทำงาน เมื่อมีค่า 10	การทำงาน เมื่อมีค่า 11	ค่าหลัง รีเซต
1:0	P0.0	GPIO Port0.0	TxD0(UART0)	PWM1	Reserved	00
3:2	P0.1	GPIO Port0.1	RxD0(UART0)	PWM3	EINT0	00
5:4	P0.2	GPIO Port0.2	SCL0(I ² C)	Capture 0.0(TIMERO)	Reserved	00
7:6	P0.3	GPIO Port0.3	SDA0(I ² C)	Match0.0 (TIMERO)	EINT1	00
9:8	P0.4	GPIO Port0.4	SCK0(SPI0)	Capture 0.1(TIMERO)	AD0.6	00

PINSELO	ชื่อขา	การทำงาน เมื่อมีค่า 00	การทำงาน เมื่อมีค่า 01	การทำงาน เมื่อมีค่า 10	การทำงาน เมื่อมีค่า 11	ค่าหลัง รีเซ็ต
11:10	P0.5	GPIO Port0.5	MISO0(SPI0)	Match0.1 (TIMER0)	AD0.7	00
13:12	P0.6	GPIO Port0.6	MISO0(SPI0)	Capture 0.2(TIMER0)	AD1.0	00
15:14	P0.7	GPIO Port0.7	SSEL0(SPI0)	PWM2	EINT2	00
17:16	P0.8	GPIO Port0.8	TxD1 (UART1)	PWM4	AD1.1	00
19:18	P0.9	GPIO Port0.9	RxD1 (UART1)	PWM6	EINT3	00
21:20	P0.10	GPIO Port0.10	RTS1(UART1)	Capture 1.0(TIMER1)	AD1.2	00
23:22	P0.11	GPIO Port0.11	CTS1(UART1)	Capture 1.1(TIMER1)	SCL1(I ² C1)	00
25:24	P0.12	GPIO Port0.12	DSR1(UART1)	Match1.0 (TIMER1)	AD1.3	00
27:26	P0.13	GPIO Port0.13	DTR1(UART1)	Match1.1 (TIMER1)	AD1.4	00
29:28	P0.14	GPIO Port0.14	DCD1(UART1)	EINT1	SDA1(I ² C1)	00
31:30	P0.15	GPIO Port0.15	RI1(UART1)	EINT2	AD1.5	00

หมายเหตุ :

- ขา P0.0 และ P0.1 ทำหน้าที่เป็น UART0 เพื่อติดต่อกับคอมพิวเตอร์ในการเขียนโปรแกรม
- การกำหนด ค่ารีจิสเตอร์ PINSELO ต้องระวังให้ P0.0 , P0.1 มีหน้าที่การทำงานเป็นขา
- TxD0 , RxD0, ของ UART0 เสมอ มิฉะนั้นอาจจะทำให้วงจรเสียได้ ดังนั้นจึงกำหนดค่าเริ่มต้นของ PINSELO ได้ดังนี้
- PINSELO = 0x00000005 ; // set P0.2 –P0.15 to GPIO Function

ตารางที่ 2.16 ค่าประจำบิตของรีจิสเตอร์ PINSEL1

PINSEL 1	ชื่อขา	การทำงาน เมื่อมีค่า 00	การทำงาน เมื่อมีค่า 01	การทำงาน เมื่อมีค่า 10	การทำงาน เมื่อมีค่า 11	ค่าหลัง รีเซ็ต
1:0	P0.16	GPIO Port0.16	EINT0	Match0.2(TIMER0)	Capture0.2 (TIMER0)	00
3:2	P0.17	GPIO Port0.17	Capture1.2 (TIMER1)	SCK1(SPI1)	Match1.2 (TIMER1)	00
5:4	P0.18	GPIO Port0.18	Capture1.3 (TIMER1)	MISO1(SPI1)	Match1.3 (TIMER1)	00
7:6	P0.19	GPIO Port0.19	Match1.2 (TIMER1)	MOIS1(SPI1)	Capture1.2 (TIMER1)	00
9:8	P0.20	GPIO Port0.20	Match1.3 (TIMER1)	SSEL(SPI1)	EINT3	00
11:10	P0.21	GPIO Port0.21	PWM5	AD1.6	Capture1.3 (TIMER1)	00
13:12	P0.22	GPIO Port0.22	V _{BUS}	Capture0.0 (TIMER0)	Match0.0 (TIMER0)	00
15:14	P0.23	GPIO Port0.23	AD0.7	สงวนไว้	สงวนไว้	00
17:16	P0.24	สงวนไว้				00
19:18	P0.25	GPIO Port0.25	AD0.4	AOUT	สงวนไว้	00
21:20	P0.26	สงวนไว้				00
23:22	P0.27	สงวนไว้				00
25:24	P0.28	GPIO Port0.28	AD0.1	Capture0.2 (TIMER0)	Match0.2 (TIMER0)	00
27:26	P0.29	GPIO Port0.29	AD0.2	Capture0.3 (TIMER0)	Match0.3 (TIMER0)	00
29:28	P0.30	GPIO Port0.30	AD0.3	EINT3	Match0.0 (TIMER0)	00
31:30	P0.31	GPIO Port0.31	UP_LED	CONNECT		00

ตารางที่ 2.17 ค่าประจำบิตของรีจิสเตอร์ PINSEL2

PINSEL2	ความหมาย	ค่าหลังรีเซ็ต
1:0	สงวนไว้	00
2	เป็น 0 ขา P1.31:26 จะใช้เป็น GPIO ถ้าเป็น1ขา P1.31:26จะเป็น Debug port	P1.26/RTCK
3	เป็น 0 ขา P1.25:16 จะใช้เป็น GPIO ถ้าเป็น1ขา P1.25:16จะเป็นTrace port	P1.20/ TRACESYNC

ตัวอย่างการกำหนดค่าขาให้ ขา P0.16-P0.31 มีการทำงานเป็น GPIO ทำโดยการเขียนค่า 0 ทุกบิตให้กับ PINSEL 1 ดังนี้

```
PINSEL1 = 0x00000000; //set P0.16-P0.31 to GPIO Function
```

ค่าแต่ละบิตของรีจิสเตอร์ PINSEL2 ที่ใช้กำหนดหน้าที่การทำงาน Port P1 แสดงได้ในตารางที่ 2.2.14 ตัวอย่างการกำหนดค่าให้ขา P1.16-P1.31 มีการทำงานเป็น GPIO ทำการเขียนค่า 0 ทุกบิตให้กับ PINSEL1

```
PINSEL2 = 0x00000000; //set P1.16-P1.31 to GPIO Function
```

การกำหนดค่าควบคุมการทำงานพอร์ตอเนกประสงค์ (GPIO)

หลังจากที่กำหนดค่าให้พอร์ตแต่ละตัวมีการทำงานเป็น GPIO แล้ว ควบคุมการทำงานของ GPIO จะส่งงานผ่านทางรีจิสเตอร์ 4 ตัวได้แก่ IOPN , IOSET, IOCKR, IODIRโดยที่รีจิสเตอร์แต่ละตัวจะมีแอดเดรสแสดงในตารางที่ 2.18

ตารางที่ 2.18 ชื่อและแอดเดรสของรีจิสเตอร์ที่ใช้ในการควบคุม GPIO

ชื่อทั่วไป	ความหมาย	การติดต่อ	ชื่อ และ แอดเดรสของ Port0	ชื่อ และ แอดเดรสของPort0
IOPIN	GPIO Port Pin value register ใช้อ่านสถานะปัจจุบันของพอร์ต	อ่าน	0XE002 8000 IOPIN0	0XE002 8010 IOPIN1
IOSET	GPIO Port Output set register ใช้กำหนดค่าให้ขาของพอร์ตมีค่าเป็น 1 ถ้าบิตใดเป็น 1 ขาของพอร์ตที่ตรงกันจะมีค่าเป็น 1	อ่าน/ เขียน	0XE002 8004 IOSET0	0XE002 8014 IOSET1

IODIR	GPIO Port Direction control register ใช้กำหนดว่าขาใดเป็นอินพุตขาใดเป็นเอาต์พุต	อ่าน/ เขียน	0XE002 8008 IODIR0	0XE002 8018 IODIR1
IOCLR	GPIO Port Output clear register ใช้กำหนดให้ขาของพอร์ตมีค่าเป็น 0	เขียน	0XE002 800C IOCLR0	0XE002 801C IOCLR1

ในไมโครคอนโทรลเลอร์ LPC2148 มีพอร์ตอเนกประสงค์ที่ทำงานได้เร็วขึ้นจะเรียกว่าเป็น Fast GPIO โดยให้กำหนดรีจิสเตอร์เพิ่มเติมขึ้นอีก แต่การทำงานที่รวดเร็วขึ้นจะต้องเขียนโปรแกรมเป็นภาษาแอสเซมบลีเท่านั้น

การทำงานของ GPIO เริ่มต้นด้วยการกำหนดทิศทางของพอร์ตก่อนว่าจะให้เป็นอินพุตหรือเอาต์พุต โดยกำหนดค่ารีจิสเตอร์ IODIR โดยค่าบิตใดเป็น 0 คือให้ขาของบิตนั้นเป็นอินพุต ถ้าให้ค่าเป็น 1 ขาของบิตนั้นจะเป็นเอาต์พุต

ตัวอย่างเช่น ต้องการให้ขา P0.22, P0.20, P0.19, และ P0.16 เป็นเอาต์พุต โดยขาของ P0 ที่เหลือเป็นอินพุต จะต้องกำหนดค่าให้กับรีจิสเตอร์ IODIR0 แต่ละบิตดังนี้

บิตที่	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ค่า	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1
	0			0			5			9						

บิตที่	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ค่า	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0			0			0			0						

เขียนเป็นโปรแกรมภาษาซีได้ดังนี้

```
IODIR 0 = 0X00590000;
```

หลังจากการกำหนดให้พอร์ตเป็นเอาต์พุตแล้ว ถ้าต้องการสั่งให้พอร์ตที่เป็นเอาต์พุตนี้มีค่าเป็น 1 ต้องสั่งที่รีจิสเตอร์ IOSET จากตัวอย่างต้องสั่งให้ขา P0.16 และ P.19 เป็น 1 จะต้องกำหนดให้ค่ารีจิสเตอร์ IOSET0 ดังนี้

บิตที่	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ค่า	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

| 0 | 0 | 0 | 9 |

บิตที่	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ค่า	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

| 0 | 0 | 0 | 0 |

เขียนเป็นโปรแกรมภาษาซีได้ดังนี้

IOSET 0 = 0X00090000;

การเขียนค่าเป็น 0 ให้รีจิสเตอร์ IOSET จะไม่มีผลต่อขาของพอร์ตที่ตรงกับบิตนั้น ขาของพอร์ตจะมีค่าคงเดิม ถ้าต้องการส่งให้พอร์ตที่เป็นเอาต์พุตนี้มีค่าเป็น 0 ต้องสั่งที่รีจิสเตอร์ IOCLR ด้วยจากตัวอย่าง ต้องการสั่งให้ขา P0.22 และ P0.20 เป็น 0 จะต้องกำหนดค่าให้กับ รีจิสเตอร์ IOCLR0 ดังนี้

บิตที่	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ค่า	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

| 0 | 0 | 5 | 0 |

บิตที่	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ค่า	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

| 0 | 0 | 0 | 0 |

เขียนโดยใช้โปรแกรมภาษาซีได้ดังนี้

IOCLR = 0X00500000;

ข้อควรระวัง ในการสั่งให้ขาพอร์ตมีค่าเป็น 0 นี้ไม่ได้สั่งให้เขียนค่า 0 ให้กับรีจิสเตอร์ IOCLR ต้องเขียนค่าเป็น 1 เท่านั้น

ในการเขียนค่าให้พอร์ต จะต้องแยกข้อมูล ถ้าบิตใดเป็น 0 จะต้องเขียนสั่งที่รีจิสเตอร์ IOCLR ถ้าบิตใดเป็น 1 จะต้องเขียนสั่งที่รีจิสเตอร์ IOSET

2.10.5 อินเทอร์เน็ตไมโครคอนโทรลเลอร์ ARM7

ในไมโครคอนโทรลเลอร์ ARM7 มีโมดูล Vectored Interrupt Controller (VIC) เป็นตัวควบคุมกลไกของการอินเทอร์เน็ต โดยสามารถรับอินเทอร์เน็ตจากอุปกรณ์ทั้งภายในและภายนอกไมโครคอนโทรลเลอร์ ARM7 ได้ทั้งหมด 32 อินพุตตามที่แสดงในตารางที่ 2.19

ตารางที่ 2.19 แหล่งกำเนิดอินเทอร์เน็ตทั้ง 32 ตัว

อุปกรณ์	แหล่งกำเนิดอินเทอร์เน็ต	VIC Channel#
WDT	Watchdog Interupt (WDINT)	0
-	Reserved for software interrupts only	1
ARM Core	Embedded ICE, DbgCommRx	2
ARM Core	Embedded ICE, DbgcommTx	3
TIME0	Match 0-3 (MR0, MR1, MR2, MR3) Capture0-3 (CR0, CR1, CR2, CR3)	4
TIME1	Match0-3 (MR0, MR1, MR2, MR3) Capture 0-3 (CR0,CR1,CR2 CR3)	5
UART0	Rx Line Status (RLS) Transmit Holding Register Empty(THRE) Rx Data Available (RDA) Charater Time-out Indicator(CTI)	6
UART1	Rx Line Status (RLS) Transmit Holding Register Empty(THRE) Rx Data Available (RDA) Charater Time-out Indicator(CTI) Modem Status Interrupt(MSI)	7
PWM0	Match 0-6 (MR0, MR1, MR2, MR3, MR4, MR5, MR6)	8
I ² C0	SI(state change)	9
SPI0	SPI Interrupt Flag (SPIF) Mode Fault(MODF)	10
SPI1(SSP)	SPI Interrupt Flag(SPIF) Mode Fault (MODF)	11
PLL	PLL Lock (PLOCK)	12
RTC	Counter Indrement(RTCCIF) Alarm(RTCALF)	13

อุปกรณ์	แหล่งกำเนิดอินเทอร์รัปต์	VIC Channel#
System Control	External Interrupt 0 (EINT0)	14
System Control	External Interrupt 1 (EINT1)	15
System Control	External Interrupt 2 (EINT2)	16
System Control	External Interrupt 3 (EINT3)	17
ADC0	A/D Converter 0 end of conversion	18
I ² C1	SI(state change)	19
BOD	Brown our detect	20
ADC1	A/D Converter 1 end of conversion	21
USB	USB interrupt, DMA Interrupt	22
	Reserved	23-31

VIC จะนำอินพุตจากการอินเทอร์รัปต์ทั้ง 32 แหล่งมาจัดแบ่งตามประเภทได้เป็น 3 ประเภท ทำให้สามารถปรับลำดับความสำคัญของอินเทอร์รัปต์จากอุปกรณ์ประกอบต่างๆ ได้ตามต้องการ คือ

- Fast Interrupt request(FIQ) จะลำดับความสำคัญสูงสุดโดยตอบสนองเร็วสุด
- Vectored IRQs จะมีการลำดับความสำคัญอยู่กึ่งกลาง โดยสามารถนำอินเทอร์รัปต์แค่ 16 บิต หรือ 32 แหล่งมาจัดให้เป็น Vectored IRQs ได้ 16 ตัว โดย Slot 0 จะมีความสำคัญสูงสุด Slot 15 มีความสำคัญต่ำสุด
- Non-vectored IRQ มีความสำคัญต่ำสุด

รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์มีทั้งหมด 43 ตัว ดังแสดงในตารางที่ 2.19 (ภาคผนวก) โดยในหัวข้อนี้จะเป็นการเลือกเฉพาะรีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์ในแบบ Vector IRQ คือรีจิสเตอร์ VICVectAddr0-15 , VICVectCntl0-15 และ VICIntEanble

รีจิสเตอร์ VICVectAddr0-15 เป็นรีจิสเตอร์ที่เก็บค่าแอดเดรสของโปรแกรมที่ตอบสนองต่ออินเทอร์รัปต์โดย VICVectAddr0 จะเป็นอินเทอร์รัปต์ Slot 0 ที่มีความสำคัญสูงสุด ตามลำดับไปจนถึง VICVectAddr15 จะเป็นอินเทอร์รัปต์ Slot 15 ที่มีความสำคัญต่ำสุด

VICVectCntl 0- 15 ใช้ในการควบคุม คือ Slot หมายเลขที่ตรงกับ VICVectCntl นี้จะรับอินเทอร์รัปต์จากแหล่งใด จากทั้งหมด 32 แหล่ง ตามตารางที่ 2.19 โดยกำหนดค่าในบิตที่ 4:0 โดยบิตที่ 5 ถ้าเป็น 1 หมายถึง อนุญาตให้อินเทอร์รัปต์จากอุปกรณ์ที่กำหนดใน Slot นี้

VICInEnable ใช้เปิดอินเทอร์รัปต์จากแหล่งต่างๆ ทั้ง 32 แหล่งตามตารางที่ 2.19 โดยบิตใดมีค่าเป็น 1 หมายถึงอนุญาตให้อินเทอร์รัปต์ได้ ถ้าบิตใดเป็น 0 ไม่อนุญาตให้อุปกรณ์นั้นๆ อินเทอร์รัปต์

ตารางที่ 2.20 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์

ชื่อ	ความหมาย	การติดต่อ	ค่าหลังรีเซ็ต	แอดเดรส
VICIRQStatus	IRQ Status Request อ่านค่าสถานะว่าอนุญาตให้มี interrupt request จากตัวใด และถูกจัดเป็น IRQ	อ่าน	0	0xFFFF F000
VICFIQStatus	FIQ Status Request อ่านค่าสถานะว่าอนุญาตให้มีอินเทอร์รัปต์รีเคสจากตัวใด และถูกจัดเป็น FIQ	อ่าน	0	0xFFFF F004
VICRawItr	Raw Interrupt Status Register ใช้อ่านสถานะของอินเทอร์รัปต์จากทั้ง 32 แหแหล่งหรือจากซอฟต์แวร์โดยไม่ว่าถูกเปิดใช้หรือถูกจัดประเภทหรือไม่	อ่าน/เขียน	0	0xFFFF F008
VICIntselect	Interrupt Select Register ใช้ในการระบุว่าอินเทอร์รัปต์ทั้ง 32 แหแหล่งแต่ละตัวเป็น FIQ หรือ IRQ	เขียน	0	0xFFFF F00C
VICIntEnable	Interrupt Enable Register ควบคุมว่าให้อินเทอร์รัปต์จากทั้ง 32 แหแหล่งตัวไหนทำงาน และให้เป็น FIQ หรือ IRQ	อ่าน/เขียน	0	0xFFFF F010
VICIntEnClr	Interrupt Enable Clear Register ใช้ล้างค่าบิตหนึ่งบิตหรือมากกว่าหนึ่งบิตของรีจิสเตอร์ Interrupt Enable Register	เขียน	0	0xFFFF F014
VicSoftInt	Software Interrupt Register ค่าของรีจิสเตอร์นี้จะถูกนำไป OR กับอินเทอร์รัปต์จากอุปกรณ์ต่างๆ ทั้ง 32 แหแหล่ง	อ่าน/เขียน	0	0xFFFF F018
VICSoftIntClear	Software Interrupt Clear Register ใช้ล้างค่าบิตหนึ่งบิตหรือมากกว่าหนึ่งบิตของรีจิสเตอร์ Software Interrupt Register	เขียน	0	0xFFFF F01C
VICProtection	Protection enable register ใช้จำกัดการเข้าถึง VIC Register	อ่าน/เขียน	0	0xFFFF F020



ชื่อ	ความหมาย	การติดต่อ	ค่าหลังรีเซ็ต	แอดเดรส
VICVectAddr	Vector Address Register เมื่อเกิดการอินเตอร์รัปต์แบบ IRQ ตัว IRQ Sevice routine จะอ่านค่าจากรีจิสเตอร์เพื่อกระโดดไปยังแอดเดรสที่ระบุไว้	อ่าน/เขียน	0	0xFFFF F030
VICDefVectAddr	Default Vector Address Register ใช้เก็บค่าแอดเดรสของ Interrupt service routine(ISR) สำหรับ IRQ non-vectorred IRQ	อ่าน/เขียน	0	0xFFFF F100
VICVEctAddr1	Vector address 1 register	อ่าน/เขียน	0	0xFFFF F104
VICVEctAddr2	Vector address 2 register	อ่าน/เขียน	0	0xFFFF F108
VICVEctAddr3	Vector address 3 register	อ่าน/เขียน	0	0xFFFF F10C
VICVEctAddr4	Vector address 4 register	อ่าน/เขียน	0	0xFFFF F110
VICVEctAddr5	Vector address 5 register	อ่าน/เขียน	0	0xFFFF F114
VICVEctAddr6	Vector address 6 register	อ่าน/เขียน	0	0xFFFF F118
VICVEctAddr7	Vector address 7 register	อ่าน/เขียน	0	0xFFFF F11C
VICVEctAddr8	Vector address 8 register	อ่าน/เขียน	0	0xFFFF F120
VICVEctAddr9	Vector address 9 register	อ่าน/เขียน	0	0xFFFF F124
VICVEctAddr10	Vector address 10 register	อ่าน/เขียน	0	0xFFFF F128
VICVEctAddr11	Vector address 11 register	อ่าน/เขียน	0	0xFFFF F12C
VICVEctAddr12	Vector address 12 register	อ่าน/เขียน	0	0xFFFF F130
VICVEctAddr13	Vector address 13 register	อ่าน/เขียน	0	0xFFFF F134
VICVEctAddr14	Vector address 14 register	อ่าน/เขียน	0	0xFFFF F138
VICVEctAddr15	Vector address 15 register	อ่าน/เขียน	0	0xFFFF F13C
VICVectCntl0	Vector control 0 register รีจิสเตอร์ Vector Control Register0-15 แต่ละตัวควบคุม IRQ Slot แต่ละตัว โดย Slot 0 มีความสำคัญสูงสุดและ Slot 15 มีความสำคัญต่ำสุด	อ่าน/เขียน	0	0xFFFF F200
VICVectCntl1	Vector control 1 register	อ่าน/เขียน	0	0xFFFF F204
VICVectCntl2	Vector control 2 register	อ่าน/เขียน	0	0xFFFF F208

ชื่อ	ความหมาย	การติดต่อ	ค่าหลังรีเซ็ต	แอดเดรส
VICVectCntl3	Vector control 3 register	อ่าน/เขียน	0	0xFFFF F20C
VICVectCntl4	Vector control 4 register	อ่าน/เขียน	0	0xFFFF F210
VICVectCntl5	Vector control 5 register	อ่าน/เขียน	0	0xFFFF F214
VICVectCntl6	Vector control 6 register	อ่าน/เขียน	0	0xFFFF F218
VICVectCntl7	Vector control 7 register	อ่าน/เขียน	0	0xFFFF F21C
VICVectCntl8	Vector control 8 register	อ่าน/เขียน	0	0xFFFF F220
VICVectCntl9	Vector control 9 register	อ่าน/เขียน	0	0xFFFF F224
VICVectCntl10	Vector control10 register	อ่าน/เขียน	0	0xFFFF F228
VICVectCntl11	Vector control 11 register	อ่าน/เขียน	0	0xFFFF F22C
VICVectCntl12	Vector control 12 register	อ่าน/เขียน	0	0xFFFF F230
VICVectCntl13	Vector control 13 register	อ่าน/เขียน	0	0xFFFF F234
VICVectCntl14	Vector control 14 register	อ่าน/เขียน	0	0xFFFF F238
VICVectCntl15	Vector control 15 register	อ่าน/เขียน	0	0xFFFF F23C

การอินเทอร์รัปต์จากอินพุตภายนอกของไมโครคอนโทรลเลอร์ ARM7

ในไมโครคอนโทรลเลอร์ LPC2148 จะรับอินเทอร์รัปต์จากภายนอกได้สูงสุด 4 แหล่ง คือ EINT0, ENIT1, ENIT2 และ ENIT3 โดยคิดเป็นอินพุตได้ทั้งหมด 9 ตัว

- EINT0 อยู่ที่ขา P0.1 และ P0.16
- EINT1 อยู่ที่ขา P0.3 และ P0.14
- EINT2 อยู่ที่ขา P0.7 และ P0.15
- EINT3 อยู่ที่ขา P0.9, P0.20 และ P0.30

โดยสามารถนำอินพุตจากการอินเทอร์รัปต์ทั้งสิ้นนี้ มาใช้ในการกระตุ้นให้ไมโครคอนโทรลเลอร์ ออกจากการทำงานในโหมด Power Down ได้

รีจิสเตอร์ที่เกี่ยวข้องกับอินเทอร์รัปต์จากภายนอก

รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์จากภายนอกมีทั้งหมด 4 ตัว คือ EXTINT, EXTWAKE, EXXTMODE และ EXTPOLAR ดังแสดงในตารางที่ 2.21

ตารางที่ 2.21 รีจิสเตอร์ที่เกี่ยวข้องกับการอินเทอร์รัปต์จากภายนอก

แอดเดรส	ชื่อ	ความหมาย	การติดต่อ
0xE01FC140	EXTINT	External Interrupt Flag Register เก็บค่าอินเทอร์รัปต์แฟล็กของ EINT0, EINT1 และ EINT2	อ่าน/เขียน
0xE01FC144	EXTWAKE	External Interrupt Wakeup Register เก็บค่าบิตที่ควบคุมว่าจะให้อินเทอร์รัปต์จากภายนอกนี้สามารถกระตุ้นซีพียูจาก Power mode หรือไม่	อ่าน/เขียน
0xE01FC148	EXTMODE	External Interrupt Mode Register ควบคุมให้แต่ละระดับจะตอบสนองต่อระดับลอคจิกหรือขอบสัญญาณ	อ่าน/เขียน
0xE01FC14C	EXTPOLAR	External Interrupt Polarity Register ควบคุมว่าแต่ละขาจะตอบสนองระดับลอคจิก 0 หรือ 1 หรือตอบสนองต่อขอบขาขึ้นขอบขาลงของสัญญาณ	อ่าน/เขียน

เมื่อเกิดการอินเทอร์รัปต์จากภายนอกเกิดขึ้น จะทำให้ค่าบิตบางบิตของรีจิสเตอร์ EXTINT มีค่าเป็น 1 และส่งต่อการอินเทอร์รัปต์ไปยัง VIC เพื่อสร้างอินเทอร์รัปต์จัดการทำงานของไมโครคอนโทรลเลอร์ เมื่อโปรแกรมตอบสนองต่อการอินเทอร์รัปต์ ทำงานเสร็จแล้วจะต้องเขียนค่า 1 ให้กับบิตเหล่านี้เพื่อหยุดการอินเทอร์รัปต์ ค่าแต่ละบิตของรีจิสเตอร์ EXTINT แสดงในตารางที่ 2.22

ตารางที่ 2.22 ค่าประจำบิตของรีจิสเตอร์ EXTINT

ค่าบิตของ EXTINT	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
0	EINT0	เมื่อขา EINT0 ทำงานมีระดับลอคจิกหรือมีขอบของสัญญาณตามที่กำหนดเกิดขึ้น บิตนี้จะมีค่าเป็น 1 เมื่อเขียนค่า 1 ให้มันเป็นการล้างค่า ยกเว้นกรณีที่ทำงานตามระดับลอคจิกต้องรอให้ค่าลอคจิกหยุดการทำงานก่อน ขาที่เป็น EINT0 คือ P0.1 และ P0.16	0
1	EINT1	เมื่อขา EINT1 ทำงานมีระดับลอคจิกหรือมีขอบของสัญญาณตามที่กำหนดเกิดขึ้น บิตนี้จะมีค่าเป็น 1 เมื่อ	0

		เขียนค่า 1 ให้มันเป็นการล้างค่า ยกเว้นกรณีที่ทำงานตามระดับลอจิกต้องรอให้ค่าบอจิกหยุดการทำงานก่อน ขาที่เป็น EINT1 คือ P0.3 และ P0.14	
2	EINT2	เมื่อขา EINT2 ทำงานมีระดับลอจิกหรือมีขอบของสัญญาณตามที่กำหนดเกิดขึ้น บิตนี้จะมีค่าเป็น 1 เมื่อเขียนค่า 1 ให้มันเป็นการล้างค่า ยกเว้นกรณีที่ทำงานตามระดับลอจิกต้องรอให้ค่าบอจิกหยุดการทำงานก่อน ขาที่เป็น EINT2 คือ P0.7 และ P0.15	0
3	EINT3	เมื่อขา EINT3 ทำงานมีระดับลอจิกหรือมีขอบของสัญญาณตามที่กำหนดเกิดขึ้น บิตนี้จะมีค่าเป็น 1 เมื่อเขียนค่า 1 ให้มันเป็นการล้างค่า ยกเว้นกรณีที่ทำงานตามระดับลอจิกต้องรอให้ค่าบอจิกหยุดการทำงานก่อน ขาที่เป็น EINT3 คือ P0.9, P0.20 และ P0.30	0
7:4	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตเหล่านี้ ค่าที่อ่านได้ไม่มี ความหมาย	ไม่กำหนด

ในการกำหนดว่าสัญญาณที่อินเทอร์รัปต์นี้ทำงานที่ระดับสัญญาณหรือที่ขาขอบของสัญญาณกำหนดได้ที่รีจิสเตอร์ EXTMODE ซึ่งมีค่าดังตารางที่ 2.23

ตารางที่ 2.23 ค่าประจำบิตของรีจิสเตอร์ EXTMODE

ค่าบิตของ EXTWAKE	หน้าที่	ความหมาย	ค่าหลัง รีเซ็ต
0	EXTWAKE0	โหมดการทำงานของ EINT0 เมื่อมีค่าเป็น 0 ทำงานที่ระดับสัญญาณ เมื่อมีค่าเป็น 1 ทำงานที่ขอบของสัญญาณ	0
1	EXTWAKE1	โหมดการทำงานของ EINT1 เมื่อมีค่าเป็น 0 ทำงานที่ระดับสัญญาณ เมื่อมีค่าเป็น 1 ทำงานที่ขอบของสัญญาณ	0
2	EXTWAKE2	โหมดการทำงานของ EINT2 เมื่อมีค่าเป็น 0 ทำงานที่ระดับสัญญาณ เมื่อมีค่าเป็น 1 ทำงานที่ขอบของสัญญาณ	0
3	EXTWAKE3	โหมดการทำงานของ EINT3 เมื่อมีค่าเป็น 0 ทำงานที่ระดับสัญญาณ เมื่อมีค่าเป็น 1 ทำงานที่ขอบของสัญญาณ	0

7:4	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตเหล่านี้ ค่าที่อ่านได้ไม่มี ความหมาย	ไม่กำหนด
-----	---------	---	----------

หลังจากที่กำหนดโหมดการทำงานของสัญญาณที่ใช้อินเตอร์รัปต์ว่าเป็นระดับของสัญญาณหรือขอบสัญญาณแล้วที่รีจิสเตอร์ EXTMODE แล้วต้องกำหนดว่าสัญญาณที่อินเตอร์รัปต์นี้ทำงานที่ระดับลอจิก 1 หรือ 0 หรือทำงานที่ขอบขาขึ้นหรือขาลงของสัญญาณ โดยกำหนดที่รีจิสเตอร์ EXTPOLAR ซึ่งแสดงรายละเอียดดังตารางที่ 2.24

ตารางที่ 2.24 ค่าประจำบิตของรีจิสเตอร์ EXTPOLAR

ค่าบิตของ EXTPOLAR	หน้าที่	ความหมาย	ค่าหลัง รีเซ็ต
0	EXTPOLAR0	เป็น 0 EINT0 ทำงานที่ลอจิก 0 หรือขอบขาลงของ สัญญาณ เป็น 1 EINT0 ทำงานที่ลอจิก 1 หรือขอบขาขึ้นของ สัญญาณ	0
1	EXTPOLAR1	เป็น 0 EINT1 ทำงานที่ลอจิก 0 หรือขอบขาลงของ สัญญาณ เป็น 1 EINT1 ทำงานที่ลอจิก 1 หรือขอบขาขึ้นของ สัญญาณ	0
2	EXTPOLAR2	เป็น 0 EINT2 ทำงานที่ลอจิก 0 หรือขอบขาลงของ สัญญาณ เป็น 1 EINT2 ทำงานที่ลอจิก 1 หรือขอบขาขึ้นของ สัญญาณ	0
3	EXTPOLAR3	เป็น 0 EINT3 ทำงานที่ลอจิก 0 หรือขอบขาลงของ สัญญาณ เป็น 1 EINT3 ทำงานที่ลอจิก 1 หรือขอบขาขึ้นของ สัญญาณ	0
7:4	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตเหล่านี้ ค่าที่อ่านได้ไม่มี ความหมาย	ไม่ กำหนด

ถ้าต้องการให้สัญญาณที่ขา EINT0-EINT3 สามารถกระตุ้นให้ไมโครคอนโทรลเลอร์ ARM7 ออกจากการทำงานในโหมด Power Down จะต้องสั่งที่รีจิสเตอร์ EXTWAKE ซึ่งมีค่าบิตของรีจิสเตอร์แสดงในตารางที่ 2.25

ตารางที่ 2.25 ค่าประจำบิตของรีจิสเตอร์ EXTPILAR

ค่าบิตของ EXTMODE	หน้าที่	ความหมาย	ค่าหลัง รีเซ็ต
0	EXTMODE0	เป็น 1 สัญญาณที่ขา EINT0 สามารถกระตุ้นให้ออกจาก Power Down ได้	0
1	EXTMODE1	เป็น 1 สัญญาณที่ขา EINT1 สามารถกระตุ้นให้ออกจาก Power Down ได้	0
2	EXTMODE2	เป็น 1 สัญญาณที่ขา EINT2 สามารถกระตุ้นให้ออกจาก Power Down ได้	0
3	EXTMODE3	เป็น 1 สัญญาณที่ขา EINT0 สามารถกระตุ้นให้ออกจาก Power Down ได้	0
7:4	สงวนไว้	ห้ามเขียนค่า 1 ให้กับบิตเหล่านี้ ค่าที่อ่านได้ไม่มี ความหมาย	ไม่กำหนด

2.10.6 การกำหนดค่าเริ่มต้นสำหรับ UART0

UART0 มีขาสัญญาณสองขา คือ

1. ขาสัญญาณ TxDO ไว้สำหรับส่งข้อมูลอยู่ที่ขา
2. ขาสัญญาณ RxDO ไว้สำหรับรับข้อมูลอยู่ที่ขา P0.1

การใช้งาน UART0 เริ่มต้นด้วยการเขียนค่ายังรีจิสเตอร์ PINSELO เพื่อกำหนดให้ขา P0.0 และ P0.1 มีการทำงานเป็น UART0 โดยต้องกำหนดบิตที่ 3-0 ของ PINSELO ให้มีค่าเป็น 1010 ซึ่งเขียนเป็นคำสั่งได้ดังนี้

```
PINSELO = 0x00000005;
```

รีจิสเตอร์ที่เกี่ยวข้องกับ UART0 มีทั้งหมด 10 ตัวโดยแต่ละตัวมีขนาด 8 บิต ดังแสดงในตารางที่

2.26

ตารางที่ 2.26 รีจิสเตอร์ที่เกี่ยวข้องกับ UART0

ชื่อ	ความหมาย	บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	การติดต่อ	ค่าหลังรีเซ็ต	แอดเดรส	
U0RBR	Receiver Buffer Register	MSB	อ่านข้อมูล							LSB	อ่าน	ไม่กำหนด	0xE000C000 DLAB=0
U0THR	Transmit Holding Register	MSB	เขียนข้อมูล							LSB	เขียน	ไม่กำหนด	0xE000C000 DLAB=0
U0IER	Interrupt Enable Register	0	0	0	0	0	Enable Rx Line status Interrupt	Enable THRE Interrupt	Enable Rx Data Available Interrupt	อ่าน/เขียน	0	0xE000C004 DLAB=0	
U0IIR	Interrupt ID Register	FIFOs Enable		0	0	IIR3	IIR2	IIR1	IIR0	อ่าน	0x01	0xE000C008	
U0FCR	FIFO Control Register	Rx Trigger		สงวนไว้			-	Tx FIFO Reset	Rx FIFO Reset	FIFO Enable	เขียน	0	0xE000C008
U0LCR	Line Control Register	DLAB	Set Break	Stick Parity	Even Parity Select	Parity Enable	Number of Stop Bit	Word Length Select		อ่าน / เขียน	0	0xE000C00C	
U0LSR	Line Status Register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	อ่าน	0x60	0xE000C014	
U0SCR	Scratch Pad Register	MSB LSB							อ่าน / เขียน		0	0xE000C01C	
U0DLL	Divisor Latch LSB	MSB LSB					อ่าน / เขียน				0x01	0xE000C000 DLAB=1	
U0DLM	Divisor Latch MSB	MSB LSB					อ่าน / เขียน				0	0xE000C004 DLAB=1	

หลังจากที่กำหนดให้ขา P0.0 และ P0.1 ทำงานเป็น UART0 แล้วถัดมาเป็นการกำหนดรูปแบบการติดต่อ เช่น ติดต่อแบบ 8 ใช้การตรวจสอบบิตพลาตแบบใด เช่น even parity จำนวนของ Stop bit โดยการกำหนดค่าผ่านทางรีจิสเตอร์ UART Line Control Register : LCR : ซึ่งมีรายละเอียดของรีจิสเตอร์ดังแสดงในตารางที่ 2.27

ตารางที่ 2.27 ค่าประจำบิตของรีจิสเตอร์ UART0 Line Control Register (U0LCR)

ค่าของบิต U0LCR	หน้าที่	ความหมาย	ค่าหลัง รีเซ็ต
1:0	Word Length Select	ตัวอักษรขนาด 5 บิต ตัวอักษรขนาด 6 บิต ตัวอักษรขนาด 7 บิต ตัวอักษรขนาด 8 บิต	0
2	Stop Bit Select	0 Stop bit 1 บิต 1 Stop Bit 2 บิต(1.5บิต ถ้า U0LCR[1:0]=00)	0
3	Parity Enable	0 ยกเลิกการเพิ่มพาริตีบิตและการตรวจพาริตี 1 ใช้งานการเพิ่มพาริตีบิตและการตรวจพาริตี	0
5:4	Parity Select	Odd parity Even Parity ให้พาริตีบิตมีค่าเป็น 1 ตลอดเวลา ให้พาริตีบิตมีค่าเป็น 0 ตลอดเวลา	0
6	Break Control	ยกเลิกการหยุดการส่ง อนุญาตให้หยุดการส่งให้ ขาที่เอาต์พุตของ UART0 TxD จะมีค่าลอจิก 0	0
7	Divisor Latch Access Bit	ไม่อนุญาตให้แก้ไขค่าตัวหาร Divisor Latch อนุญาตให้แก้ไขค่าตัวหาร Divisor Latch	0

ในรีจิสเตอร์ LCR มีบิตที่เรียกว่า DLAB (Divisor Latch Access bit) ถ้าต้องการปรับค่าของวงจรถ่าย Broad Rate ต้องเซตบิตนี้ให้เป็น 1 ค่าของ Baud rate generator เป็นค่าของตัวหารขนาด 16 บิต เพื่อนำไปใช้หารค่าของ PCLK เพื่อให้ได้ความถี่สูงกว่าความเร็ว Broad Rate 16 เท่า ทำให้สมการค่าตัวหารเป็นดังนี้

$$\text{Divisor} = \text{PCLK}/(16*\text{Baud})$$

แผงวงจร CP-JR ARM7 USB-LPC2148 EXP ใช้คริสตัลความถี่ 12.000MHz สั่ง PLL คูณ 5 จะได้ CCLK = 60.0 MHz และกำหนดให้ VPBDIV = 2 จะได้ PCLK = 30.0 MHz ด้วย ถ้าต้องการอัตราบอดที่ 9600 bps ตัวหารจะมีค่าเท่ากับ

$$\text{Divisor} = 30,000,000 / (60 * 9600) = 195.3125 \quad \text{ปัดเศษลง}$$

$$\text{Divisor} = 195 \quad \text{หรือ} \quad 0xC3 \quad \text{นำค่าที่ได้ไปคำนวณหาอัตราบอดจะได้}$$

$$\text{Baud} = \text{PCLK} / (16 * \text{Divisor}) = 30,000,000 / (16 * 195) = 9615 \text{ bps}$$

ซึ่งผิดพลาดไป 0.156% สามารถใช้งานได้ เนื่องมาตรฐานของการสื่อสารแบบอนุกรมสามารถรับอัตราบอดที่ผิดพลาดได้ถึง 5%

นำค่าหารที่ได้ไปเก็บลงในรีจิสเตอร์ขนาด 8 บิตสองตัว คือ Divisor Latch MAB(DLM) และ Divisor Latch LSB (DLL) ในขณะที่เขียนค่าเก็บในรีจิสเตอร์ DLM และ DLL ค่าบิต DLAB ต้องมีค่าเป็น 1 เมื่อเขียนเสร็จแล้วต้องรีเซ็ตค่าบิตนี้ให้กลับเป็น 0 ซึ่งเขียนเป็นคำสั่งภาษาซีได้ดังนี้

```
UODLM = 0x00;
```

```
UODLL = 0xC3;
```

```
UOLCR = 0x7F;
```

ในการเรียกใช้ฟังก์ชัน `uart0_init()` จะต้องส่งค่าอัตราบอดที่ต้องการให้ฟังก์ชัน ตัวอย่างเช่น ต้องการอัตราบอดที่ 9600 bps จะต้องเรียกใช้ฟังก์ชันดังนี้

```
uart0_init(9600);
```

เมื่อกำหนดการทำงานให้กับ UART แล้ว จะสามารถรับส่งค่าผ่านพอร์ตอนุกรมได้ในการส่งข้อมูลต้องเขียนข้อมูลไปยังรีจิสเตอร์ Transmit Holding Register (THR) ถ้าต้องการอ่านข้อมูลที่รับพอร์ตอนุกรมต้องอ่านค่าจากรีจิสเตอร์ Receiver Buffer Register (RBR) จากตารางที่ 2.2.24 (ภาคผนวก) จะพบว่าค่าแอดเดรสของรีจิสเตอร์ทั้งสอง มีค่าอยู่ที่ตำแหน่งเดียวกัน แสดงว่าการเขียนค่าให้กับ THR เป็นการเขียนค่าลงในบัฟเฟอร์แบบ FIFO ของ UART0 การอ่านค่าจากรีจิสเตอร์ RBR เป็นการอ่านค่าจากบัฟเฟอร์แบบ FIFO

ก่อนที่จะอ่านหรือเขียนค่าลงในรีจิสเตอร์ THR หรือ RBR จะต้องอ่านค่าสถานะของ UART ก่อนว่ามีการผิดพลาดหรือไม่ โดยอ่านค่าสถานะที่รีจิสเตอร์ Line Status Register (LSR) ก่อน โดยค่าประจำบิตของรีจิสเตอร์แสดงได้ในตารางที่ 2.28

ตารางที่ 2.28 แสดงค่าประจำบิตของ UART0 Line Status Register (UOLSR)

ค่าบิตของ UOLSR	หน้าที่	ความหมาย	ค่าหลังรีเซ็ต
0	Receiver Data Ready (RDR)	0 : UORBR ว่าง 1 : UORBR มีรับข้อมูลที่ถูกต้อง UOLSR0 เป็น 1 เมื่อ UORBR มีค่าตัวอักษรที่ยังไม่อ่านและถูกล้างค่าเมื่อ UART0 RBR FIFO ว่าง	0
1	Overrun Error (OE)	0 : ไม่มี Overrun error 1 : มี Overrun error เกิดขึ้น	0

ค่าบิตของ UOLSR	หน้าที่	ความหมาย	ค่าหลังรีเซต
		Overrun error เกิดขึ้นเมื่อ UART0 RSR ได้รับตัวอักษรตัวใหม่ในขณะที่บัฟเฟอร์ FIFO เต็ม ในกรณีนี้ข้อมูลตัวใหม่ใน UART0 RSR จะสูญหายไป เมื่ออ่านค่าของ UOLSR จะล้างค่าบิตนี้	
2	Parity Error (PE)	0 : ไม่มี Parity Error 1 : มี Parity Error เกิดขึ้น ใช้ตรวจสอบว่าข้อมูลที่รับมาใน UART0 RBR FIFO มีการผิดพลาดที่พาริตีบิตหรือไม่ เมื่ออ่านค่าของ UOLSR จะล้างค่าของบิตนี้	0
3	Framing Error (FE)	0 : ไม่มี Framing Error 1 : มี Framing Error เกิดขึ้น เมื่อ Stop bit ของข้อมูลที่รับมามีค่าเป็น 0 แสดงว่าเกิดการผิดพลาดที่เฟรมข้อมูลในขณะที่เกิดการผิดพลาดที่เฟรมข้อมูลนี้ UART0 จะพยายามรับข้อมูลโดยนำ Stop bit ที่ผิดพลาดนี้มาใช้เป็น Start bit ข้อมูลตัวถัดไป ซึ่งอาจจะอ่านข้อมูลได้ถูกต้องหรือไม่ถูกต้อง	0
4	Break Interrupt (BI)	0 : ไม่ใช่ Break Interrupt 1 : ใช้ Break Interrupt เมื่อ RxD0 ได้รับข้อมูลที่เป็น 0 ทุกบิต (start, data, parity, stop) จะเกิด Break Interrupt ภาครับจะหยุดทำงาน จนกว่าจะได้รับข้อมูลที่เป็น 1 ทุกบิตอีกครั้งหนึ่ง	0
5	Transmitter Holding Register Empty (THRE)	0 : U0THR มีข้อมูลที่ถูกต้อง 1 : U0THR ว่าง บิต THRE จะถูกเซตค่าเป็น 1 ทันทีที่พบว่า UART0 THR ว่าง และถูกล้างค่าทันทีที่มีการเขียนค่าไปยัง U0THR	1
6	Transmitter Empty (TEMT)	0 : U0THR และ/หรือ U0TSR มีข้อมูลที่ถูกต้อง 1 : U0THR และ U0TSR ว่าง บิตนี้จะถูกล้างค่าเมื่อ U0THR หรือ U0TSR มีข้อมูลที่ถูกต้อง	1

ค่าบิตของ UOLSR	-หน้าที่	ความหมาย	ค่าหลัง รีเซ็ต
7	Error in Rx FIFO (RXFE)	0 : ข้อมูลที่เก็บใน UART0 RBR FIFO ไม่ผิดพลาด 1 : ข้อมูลที่เก็บใน UART0 RBR FIFO อย่างน้อยหนึ่งตัว มีการผิดพลาด	0

ก่อนที่จะเขียนข้อมูลให้ UART ต้องตรวจสอบดูที่บิต Transmitter Empty (TEMT) ของ รีจิสเตอร์ LSR ก่อนว่าบัพเฟอร์สำหรับส่งว่างหรือไม่ ถ้าว่างจะได้ค่าเป็น 1 จึงส่งข้อมูลได้ก่อนที่จะอ่านข้อมูลจาก UART ต้องตรวจสอบบิต Receiver Data Ready (RDR) ก่อน ถ้ามีข้อมูลพร้อมแล้วบิตนี้จะมีค่าเป็น 1 จึงอ่านค่าจาก UART ได้