

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

การวิจัยนี้ มีวัตถุประสงค์คือ การประยุกต์ใช้เซ็นเซอร์สามแกนกับกราฟิกสามมิติ เพื่อพัฒนาการออกกำลังกายให้มีรูปแบบที่น่าสนใจและแตกต่างไปจากเดิม โดยอาศัยการทำงานจากสองส่วนหลักๆ คือ ส่วนของอุปกรณ์และส่วนของกราฟิกสามมิติ อุปกรณ์ที่ใช้เป็นชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino ที่มี MCU เป็น ATmega328 ควบคุมร่วมกับเซ็นเซอร์วัดความเร่งทิศทางแกน X, Y, Z ที่เรียกว่า Accelerometer ADXL345 ใช้ลักษณะการส่งข้อมูลในรูปแบบอนุกรม I2C Bus โดยการทำงานของชุดอุปกรณ์ดังกล่าวจะเกิดขึ้นเมื่อมีการส่งค่าในรูปแบบอนาล็อกจากไดนาโมจิกรยาน ในส่วนของกราฟิกสามมิติมีการออกแบบกราฟิกโมเดลด้วยโปรแกรม Blender3D และควบคุมการทำงานระหว่างอุปกรณ์กับกราฟิกสามมิติด้วยโปรแกรมภาษา Python ดังนั้นเพื่อให้อุปกรณ์และโปรแกรมต่างๆสามารถติดต่อกันได้ตามขอบเขตที่วางไว้ จึงต้องศึกษาข้อมูลและทฤษฎีที่เป็นพื้นฐานสำคัญ โดยแบ่งออกเป็นกลุ่มต่างๆ ตามลักษณะการทำงานดังนี้

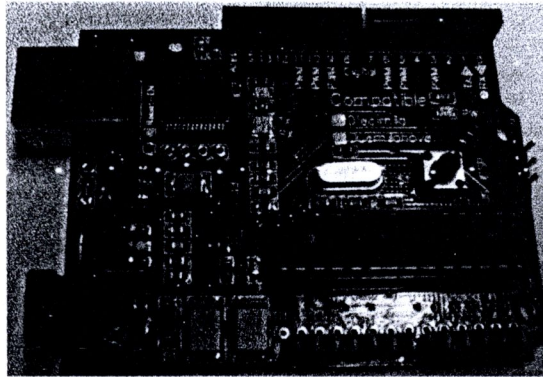
2.1 ทฤษฎีไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรส่งสัญญาณเอาต์พุต รวมถึงหน่วยความจำวงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี โดยไมโครคอนโทรลเลอร์มาจากคำสองคำรวมกันคือ “ไมโคร” ซึ่งหมายถึงไมโครโปรเซสเซอร์ เป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็กภายในประกอบด้วย หน่วยประมวลผลกลางหรือ CPU ประกอบด้วยหน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรเชื่อมต่อหน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า “คอนโทรลเลอร์” หมายถึง อุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างเป็นอิสระ

2.1.1 ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino รุ่น Duemilanove Compatible

เป็นไมโครคอนโทรลเลอร์ตระกูล AVR โดยใช้ไมโครคอนโทรลเลอร์เบอร์ ATmega 328 เป็น MCU ประจําบอร์ด โดย MCU รุ่นนี้ มีขา pin ทั้งหมด 28 ขาและมีจุดเด่นคือเป็น

ไมโครคอนโทรลเลอร์ขนาดเล็ก แต่เพียบพร้อมไปด้วยทรัพยากรพื้นฐานต่างๆ อย่างครบถ้วน จึงมีความเหมาะสมเป็นอย่างยิ่งในการใช้งานทั่วไป สำหรับภายในมีทั้งระบบฮาร์ดแวร์ของ SPI, UART, I2C, Watchdog, Timer/Counter, PWM เป็นต้นและสามารถใช้ในการพัฒนาโปรแกรมด้วย Arduino ได้ทันที



รูปที่ 2.1 ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino

2.1.2 คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์

ชุดโมดูลบอร์ดไมโครคอนโทรลเลอร์ Arduino มี ATmega328 เป็น MCU ตระกูล AVR ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Crystal Oscillator ที่มีค่า 16 MHz เพื่อให้สามารถใช้งานพอร์ทสื่อสารอนุกรมได้อย่างลงตัว

- บอร์ดสามารถเปลี่ยนการติดตั้ง MCU เป็นแบบ 28 pin หรือเบอร์อื่นในอนุกรมเดียวกันได้โดยไม่ต้องมีการดัดแปลงใดๆ เช่น ATmega88 เป็นต้น
- มีหน่วยความจำแบบ Flash 32 Kbyte โดยแบ่งเป็น Boot loader 2 Kbyte
- มี EPROM 1Kbyte/SRAM 2Kbyte
- มีพอร์ทเป็น 14 digital input/output pins ซึ่งมี 6 pin สามารถสร้างเป็น PWM outputs
- มีพอร์ท 6 Analog input/output pins
- ไฟกระแสตรง (DC) ขา I/O pin มีค่า 40 mA
- ไฟกระแสตรง (DC) ขา 3.3V pin มีค่า 50 mA
- MCU ประจำบอร์ดที่ได้รับการติดตั้ง Boots loader สามารถอัปโหลดโค้ดให้บอร์ดผ่านทางพอร์ทสื่อสารแบบอนุกรมได้ทันที
- มีขั้วต่อ USB Interface สื่อสารข้อมูลแบบอนุกรมเข้า คอมพิวเตอร์ได้ทันที
- มี LED สำหรับแสดงสถานะไฟเลี้ยง
- มี LED แสดงสถานะการรับส่งข้อมูล

- ใช้ไฟเลี้ยง ประจําบอร์ด 7-12 Volt33

2.1.3 ทฤษฎีภาษา Arduino

“Arduino” เป็นภาษาอิตาลีซึ่งเป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open source ที่ได้รับการปรับปรุงมาจากโครงการ Open source ของ AVR

Arduino มีจุดเด่นในเรื่องของความสะดวกในการเรียนรู้และใช้งาน เนื่องจากการออกคำสั่งต่างๆนั้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ซึ่งแม้ว่า Arduino จะมีรูปแบบการใช้งานคล้ายกับไมโครคอนโทรลเลอร์อย่าง Basic Stab ของ Parallax แต่ก็มีจุดเด่นกว่ารายอื่นคือ

- ราคาไม่แพง เนื่องจากมี Source Code และวงจร แจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้เองได้ รวมถึงมีการเปิดเผยวงจร Source code ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดได้ดี ทั้งด้านฮาร์ดแวร์และซอฟต์แวร์
- โปรแกรมที่ใช้พัฒนาของ Arduino สามารถรองรับการทำงานทั้ง Window, Linux และ OSx
- รูปแบบคำสั่งง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริงๆ กับส่วนที่มีความซับซ้อนมากๆได้ และยังสามารถสร้างคำสั่งรวมถึง Library ใหม่ๆ ขึ้นมาใช้งานได้ เมื่อมีความชำนาญมากขึ้น

2.1.4 เปรียบเทียบภาษาซีกับ Arduino

สำหรับการเขียนโปรแกรมของ Arduino นั้นใช้ภาษา C++ ซึ่งเป็นรูปแบบของภาษาซีประยุกต์รูปแบบหนึ่งที่มีโครงสร้างการทำงานของตัวภาษาโปรแกรมโดยรวมคล้ายกับ ภาษาซีมาตรฐานทั่วไป เพียงแต่ได้มีการปรับปรุงเพื่อลดความยุ่งยากในการใช้งานลดลง และให้ผู้ใช้สามารถใช้งานเขียนโปรแกรมได้ง่าย สะดวกมากกว่าการเขียนภาษาซีแบบมาตรฐาน แต่ในความเป็นจริงนั้นโปรแกรมหังกล่าวไม่ใช่ C-compiler โดยตรง เนื่องจาก Arduino จะมีลักษณะการทำงานเช่นเดียวกับ Text Editor ของภาษา C++ ตัวหนึ่งโดยจะทำงานร่วมกับ Utility บางส่วนที่ Arduino สร้างขึ้นมารองรับโดย Arduino จะใช้รูปแบบการทำงานของ Editor เป็นฉากหน้า ในการติดต่อสื่อสารกับผู้ใช้เท่านั้น ส่วนเบื้องหลังจริงๆแล้ว Arduino จะไปเรียกใช้ตัวแปลภาษาซี และ Utility อื่นที่ใช้เป็นเครื่องมือพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล AVR อีกทีหนึ่ง Arduino จะเลือกใช้คอมไพเลอร์ของ “GNU AVR-GCC Toolchain” ร่วมกับ Library function ของ “avr-libc” ส่วน Utility ที่ใช้ในการอัปโหลดโค้ดให้กับ AVR จะใช้ในส่วน of “AVRDude”

2.1.5 โครงสร้างการเขียนโปรแกรมภาษาซีของ Arduino

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมเป็นส่วนย่อยๆ หลายๆส่วน โดยเรียกแต่ละส่วนว่า “ฟังก์ชัน” และเมื่อนำฟังก์ชันมารวมเข้าด้วยกัน ก็จะเรียกว่า

“โปรแกรม” โดยโครงสร้างการเขียนโปรแกรมของ Arduino ทุกๆโปรแกรมจะประกอบไปด้วย ฟังก์ชันจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมี 2 ฟังก์ชันคือ `setup()` และ `loop()`

- **Setup()** : เป็นฟังก์ชันบังคับที่ต้องกำหนดให้มีทุกๆโปรแกรม ถึงแม้ว่าบางโปรแกรม จะไม่ต้องการใช้งานก็ยังจำเป็นต้องประกาศไว้เสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใดๆไว้ หลังวงเล็บปีกกา {} ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชัน โดยฟังก์ชันนี้จะใช้สำหรับ บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียว ตอนเริ่มต้นทำงานของ โปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่างๆ เช่น การกำหนดหน้าที่ของการใช้งานของ PinMode และค่า Baudrate สำหรับการใช้งาน สื่อสารพอร์ตอนุกรม เป็นต้น
- **Loop()** : เป็นส่วนฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆโปรแกรม เช่นเดียวกันกับ ฟังก์ชัน Setup() โดยฟังก์ชัน Loop() นี้จะใช้ในการบรรจุคำสั่งที่ต้องการให้โปรแกรม ทำงานเป็นวนรอบซ้ำๆ กันไปไม่รู้จบ ซึ่งเปรียบเทียบกับได้กับ ฟังก์ชัน `main ()` ใน ANSCI-C นั่นเอง

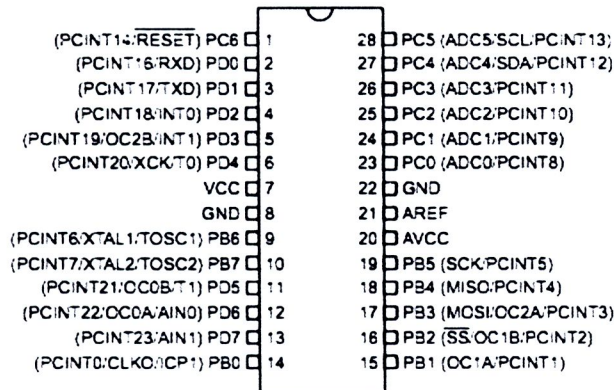
2.2 ทฤษฎีไมโครคอนโทรลเลอร์ตระกูล AVR รุ่น ATmega 328P

2.2.1 สถาปัตยกรรมขั้นสูงแบบ RISC(Reduce Instruction Set Computer)

- RISC คือตัวทำให้การประมวลผลมีความเร็ว 1 คำสั่ง/1 clock หรือ CPU สามารถ ประมวลผลคำสั่งได้ 1 MIPS/MHz
- ชุดคำสั่ง 131 คำสั่งต่อ 1 รอบนาฬิกา
- รีจิสเตอร์ขนาด 8 บิต 32 ตัว
- ความเร็วในการประมวลผลมากกว่า 20 ล้านคำสั่งต่อวินาที (MIPS) ที่ 20 MHz

- มีระบบการขัดจังหวะทั้งภายในและภายนอก
- มีระบบตรวจจับความผิดพลาดของ CPU
- มีโหมดอนุรักษ์พลังงาน 5 mode ได้แก่ Idle, ADC Noise Reduction, Power-Save, Power-Down, Standby

2.2.2 ส่วนประกอบต่างๆของไอซี ATmega 328P



รูปที่ 2.3 ไอซี ATmega 328P [1]

ตารางที่ 2.1 ตารางแสดงฟังก์ชันการทำงานของ ATmega 328P

ชื่อ	รายละเอียด	ขา
GND	ขากราวด์ต่อสายดิน	8,22
VCC	ไฟเลี้ยง 1.8 ถึง 5.5v	7
Port B(PB 7: 0) XTAL1/XTAL2/TOSC1/TOSC2	- เป็นพอร์ต 2 ทิศทาง ขนาด 8 บิต โดยสามารถกำหนดให้ขาของแต่ละพอร์ตสามารถตั้งค่าให้ Pull up Resistor ได้ (ภายในเป็นอิสระแยกจากกัน เพื่อตั้งแรงดันของลอจิก 1 ให้เท่ากับ 5 v) - สามารถใช้งานพิเศษตามความต้องการของ ATmega 328 โดยขึ้นอยู่กับค่าสัญญาณนาฬิกาที่ขา PB6 ที่ใช้เป็นแรงดัน Oscillator และขาอินพุตของวงจรสัญญาณ Clock Oscillator	9,10,14-19
PC6/RESET	ขา Reset	1

ชื่อ	รายละเอียด	ขา
Port D (PD7:0)	- เป็นพอร์ตสองทิศทางขนาด 8 บิต โดยสามารถกำหนดให้ขาของแต่ละพอร์ตสามารถตั้งค่าให้ Pull up Resistor ได้ - สามารถใช้งานพิเศษตามความต้องการของ ATmega 328	1-6,11-13
AVCC	ใช้จ่ายไฟให้กับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล มักจะต่อเข้ากับขา VCC	20
AREF	แรงดันอ้างอิงที่ใช้งานในส่วน of วงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลมักต่อกับ VCC	21
ADC7:6 (TQFP and QFN/MLF Package Only)	ขากำลังงานใช้แปลงสัญญาณอนาล็อกเป็นดิจิตอล	23-28
Port C (PC5:0)	- เป็นพอร์ตสองทิศทางขนาด 8 บิต โดยสามารถกำหนดให้ขาของแต่ละพอร์ตสามารถตั้งค่าให้ Pull up Resistor ได้ - สามารถใช้งานพิเศษตามความต้องการของ ATmega 328	23-28

2.3 ทฤษฎีการสื่อสารแบบอนุกรม

ทฤษฎีการสื่อสารแบบอนุกรม เป็นการรับส่งข้อมูลที่ละบิตจนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์คือ D0 - D7 อาจจะส่ง D0 ออกไปก่อน แล้วตามด้วย D1 ไปเรื่อยๆจนถึง D7 โดยการส่งข้อมูลแบบอนุกรมมีความแตกต่างกับแบบขนานทั้งข้อดีและข้อเสียดังนี้ การส่งข้อมูลแบบขนานสามารถรับส่งข้อมูลได้อย่างรวดเร็ว เพราะการส่งข้อมูลสามารถทำได้ทีละบิตครบ 1 ไบต์ แต่ถ้าต้องการส่งระยะทางไกลๆจะต้องสิ้นเปลืองสายสัญญาณมาก สำหรับการส่งข้อมูลแบบอนุกรม จะแตกต่างคือ ถ้าส่งระยะทางไกลๆจะประหยัดสายสัญญาณ จากการใช้สายสัญญาณอย่างน้อยสองเส้นคือ สายสัญญาณกับสายกราวด์ แต่การรับส่งข้อมูลจะใช้เวลาานกว่าเนื่องจากการรับส่งค่าทีละบิต

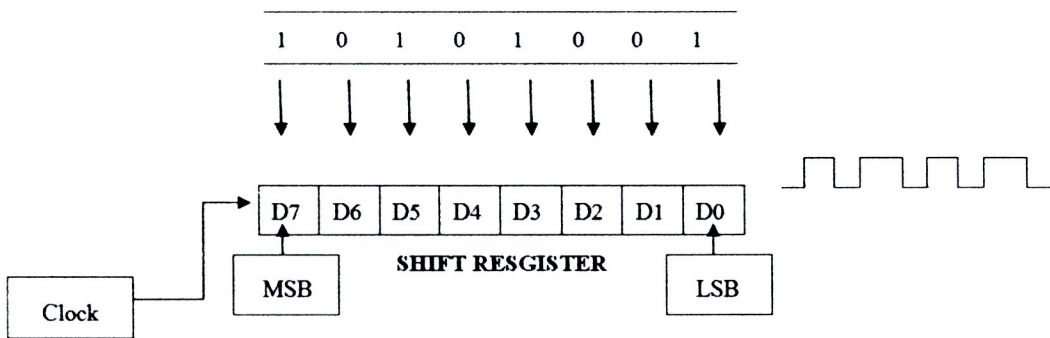
2.3.1 การสื่อสารข้อมูลแบบ Asynchronous

เป็นการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาไปด้วยแต่จะใช้การกำหนดค่าอัตราเร็วในการรับส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตรานี้ว่า Baud rate (Bit/second)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบ Asynchronous ประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรม มีขนาด 8 บิต
3. บิตตรวจสอบพาริตี (parity bit) มีขนาด 1 บิตหรือไม่ก็ได้
4. บิตหยุด (Stop bit) มีขนาด 1 บิต

เมื่อไมโครคอมพิวเตอร์ต้องการรับส่งข้อมูลแบบอนุกรม ตัวไมโครคอมพิวเตอร์จะส่งข้อมูลออกไปแบบขนานก่อน จากนั้นจะมีอุปกรณ์ที่มาต่อพอร์ต เพื่อแปลงข้อมูลแบบขนานให้เป็นอนุกรมทีหนึ่ง ตัวแปลงข้อมูลนี้อาจพิจารณาว่าเป็น Shift Register



รูปที่ 2.4 การส่งข้อมูลแบบอนุกรม [2]

สำหรับตัวรับข้อมูลแบบอนุกรมนั้นเมื่อตัวรับข้อมูลทำงานจะเป็นการรับเข้ามาใน shift Register แล้วส่งข้อมูลให้ไมโครคอมพิวเตอร์แบบขนานอีกที ระบบคอมพิวเตอร์ในปัจจุบันมีตัวแปลง Pararell to Serial และ Serial to Parallel อยู่ในชิพไอซีเรียกว่า Universal Asynchronous Receiver Transmitter (UART) การส่งข้อมูลแบบอนุกรมนั้นจะต้องมีการเพิ่มเติมข้อมูลบางอย่างเข้าไปเพื่อให้ข้อมูลมีความถูกต้องมากขึ้น

Start	D0	D1	D2	D3	D4	D5	D6	D7	Parity	Stop
-------	----	----	----	----	----	----	----	----	--------	------

รูปที่ 2.5 บิตต่างๆของการรับส่งข้อมูลแบบอนุกรม [2]

ถ้ามีการส่งข้อมูลแบบ 8 บิตจะต้องมีการส่งบิตแรกออกไปก่อน เรียกว่า บิตเริ่มต้น (Start Bit) ถ้ามีการส่งข้อมูลหลายๆไบต์ออกมา บิตนี้จะเป็นตัวบอกว่ามีข้อมูลใหม่มาแล้ว โดยทั่วไปบิตเริ่มต้นมักมีระดับลอจิกเป็น “0” ต่อจากนั้นจะเป็น ข้อมูลบิต D0 - D7 จากนั้นตามด้วยบิตตรวจสอบและบิตหยุด เพื่อบอกการสิ้นสุดของบิตข้อมูลซึ่งบิตหยุดอาจมากกว่า 1 บิตก็ได้

การสื่อสารแบบอนุกรมนี้ การกำหนดอัตราเร็วในการรับส่งข้อมูลจะบอกเป็นบิตต่อวินาที bps ที่เรียกว่าอัตราบอดเรต (BAUD RATE) โดยค่ามาตรฐานที่ใช้กรณีหลายค่า เช่น ค่า 9600 หรือ ค่า 115200 เป็นต้น

2.4 หลักการพื้นฐานของ USB

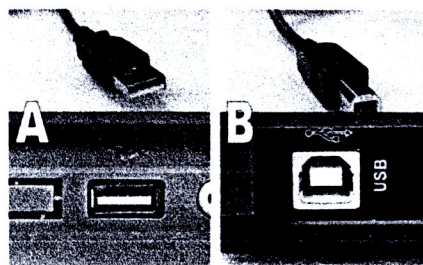
USB หรือ Universal Serial Bus ได้ถูกกำหนดขึ้นมาโดยบริษัทยักษ์ใหญ่ผู้นำทางด้านอุปกรณ์ไฟฟ้า อิเล็กทรอนิกส์และคอมพิวเตอร์ ซึ่งประกอบไปด้วย COMPAQ, IBM, DEC, Intel, Microsoft, NEC และ Northern Telecom ช่วยกันวางมาตรฐานให้อยู่ในรูปแบบเดียวกัน โดยในยุคเริ่มแรกนั้นมาตรฐานของ USB ที่ ออกสู่สาธารณะชนเป็นครั้งแรกเมื่อวันที่ 11 พฤศจิกายน ปี ค.ศ. 1994 คือ Revision 0.7 และได้ปรับปรุงแก้ไขเรื่อยมา จนกระทั่งเมื่อ วันที่ 15 มกราคม ค.ศ. 1996 ได้ออกมาเป็น Revision 1.0 (USB1.0) ได้เป็นผลสำเร็จและยังได้ปรับปรุง พัฒนาแก้ไข ปัญหาต่างๆ เรื่อยมาจนเมื่อวันที่ 23 กันยายน ค.ศ. 1998 ได้เป็น Revision 1.1 (USB1.1) เมื่อความเร็วที่ได้ ยังไม่เพียงพอกับความต้องการ

ดังนั้นทางกลุ่มผู้พัฒนา หรือ USB-IF (USB Implementers Forum, Inc.) ได้ร่าง มาตรฐาน USB รุ่นใหม่ และได้ข้อสรุป เป็นมาตรฐานที่แน่นอน คือ USB 2.0 ในเดือนเมษายน ปี ค.ศ. 2000 สำหรับความเร็วในการรับ-ส่งข้อมูลนั้น USB1.1 จะมีความเร็วอยู่ที่ 12Mbps ส่วน USB 2.0 นั้น รองรับระดับการรับส่งข้อมูลได้ถึง 3 ระดับ คือ

- ความเร็ว 1.5 Mbps (Low Speed) สำหรับการเชื่อมต่อกับอุปกรณ์ที่ไม่จำเป็นต้องส่งข้อมูลคราวละมากๆ
- ความเร็ว 12 Mbps (Full Speed) สำหรับการเชื่อมต่อกับ USB 1.1
- ความเร็ว 480 Mbps (Hi-Speed) สำหรับการเชื่อมต่อกับ USB 2.0 ด้วยกัน

2.4.1 ชนิดของหัว Connector

Connector มีอยู่ 2 แบบ คือ แบบ A และ แบบ B และ Socket ดังรูป



รูปที่ 2.6 ชนิดของหัว Connector แบบ A และแบบ B [3]

ลักษณะการทำงานของหัวทั้งสองแบบมีดังนี้

- แบบ A : เป็นการส่งข้อมูลจากอุปกรณ์ไปยังเครื่อง Computer เพื่อการประมวล เรียกว่า UpStream
- แบบ B : เป็นการส่งข้อมูลเข้าหาอุปกรณ์ เรียกว่า DownStream

2.4.2 ลักษณะการเชื่อมต่อ

การเชื่อมต่อใช้งานนั้นสามารถ เชื่อมต่อร่วมกันได้ทั้งที่เป็น USB1.1 และ USB 2.0 แต่จะได้ความเร็วที่ต่างกัน

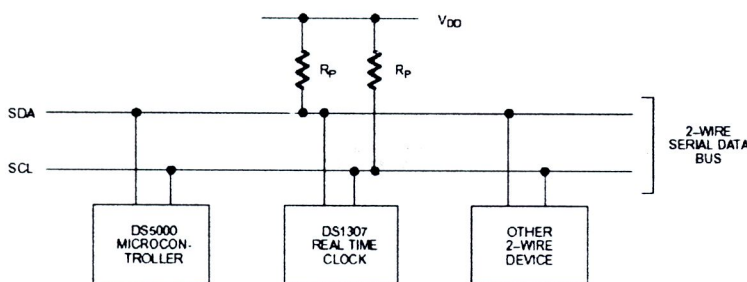
ถ้าหากต่ออุปกรณ์มาตรฐาน USB1.1 บนระบบบัสที่เป็น USB2.0 จะได้ความเร็ว = 12Mbps

ถ้าหากต่ออุปกรณ์มาตรฐาน USB2.0 บนระบบบัสที่เป็น USB1.1 จะได้ความเร็ว = 12Mbps

ถ้าหากต่ออุปกรณ์มาตรฐาน USB2.0 บนระบบบัสที่เป็น USB2.0 จะได้ความเร็ว = 480Mbps

2.5 ทฤษฎีการสื่อสารแบบ I2C Bus

Inter Integrate Circuit Bus (IIC Bus) เรียกสั้นๆว่า “I2C Bus” ซึ่งเป็นชื่อของการติดต่อสื่อสารแบบอนุกรมแบบหนึ่ง ถูกคิดค้นและพัฒนาโดย PHILIPS SEMICONDUCTOR เมื่อหลายปีก่อน แต่เพิ่งมาได้รับความนิยมแพร่หลายเมื่อไม่นานมานี้ ซึ่งปัจจุบันนี้ได้มีหลายบริษัทให้ความสนใจในการติดต่อสื่อสารแบบนี้เป็นอย่างมาก เนื่องจากรูปแบบการเชื่อมต่ออุปกรณ์ด้วยระบบบัสแบบนี้จะมีข้อดีคือใช้สัญญาณในการเชื่อมต่อเพียง 2 เส้น (SCL และ SDA) แต่สามารถเชื่อมต่ออุปกรณ์จำนวนหลายตัวในบัสเดียวได้ ซึ่งปัจจุบันถือว่าเป็นยุคสมัยของไมโครคอนโทรลเลอร์ขนาดเล็ก เนื่องจากระบบการทำงานของวงจรต่างๆมุ่งเน้นออกแบบให้มีขนาดเล็กกะทัดรัดและสามารถใช้งานได้หลายหลาย ดังนั้นพวก chip support ต่างๆจึงได้ออกแบบมาเพื่อตอบสนองการใช้งานแบบ I2C Bus มากยิ่งขึ้น



รูปที่ 2.7 ลักษณะการการเชื่อมต่ออุปกรณ์แบบ I²C BUS [4]

2.5.1 การเชื่อมต่อบัสแบบ I2C Bus

การเชื่อมต่อนี้จะใช้สัญญาณทั้งหมด 2 เส้นคือ SCL และ SDA โดยการติดต่อระหว่างอุปกรณ์จะเป็นแบบ 2 ทิศทาง โดยสัญญาณทั้งสองเส้นจะต้องต่อกับตัวต้านทาน Pull up ไว้

เพื่อให้สถานะของบัสในขณะที่ไม่ถูกใช้งานเป็นลอจิก “1” ทั้งคู่โดยอุปกรณ์ต่างๆที่ถูกออกแบบมาเพื่อเชื่อมต่อกับบัสแบบนี้ จะต้องสร้างวงจรภาคเอาพุตให้เป็นแบบ Open Drain หรือ Open collector เสมอเพื่อให้อุปกรณ์สามารถติดต่อกันได้มากกว่า 1 อุปกรณ์

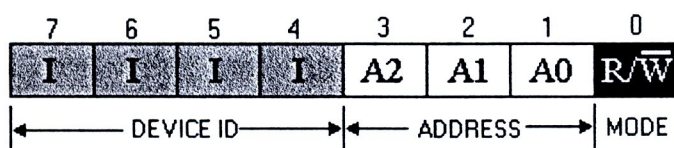
2.5.2 หลักการของ I2C

บัส I2C ประกอบด้วยสายสัญญาณ 2 เส้นดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้เป็นการอธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูลเรียกว่า ตัวส่ง (transmitter) อุปกรณ์ที่เป็นผู้รับข้อมูลเรียกว่า ตัวรับ (receiver) ในอุปกรณ์บนบัส I2C สามารถเป็นได้ทั้งตัวรับและตัวส่งบางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียวจะไม่มีอุปกรณ์ใดบนบัส I2C เรียกว่า มาสเตอร์ (master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I2C เรียกว่า สเลฟ (slave) โดยข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I2C คือ

- 1.) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น
- 2.) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูงสายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

การรับส่งข้อมูลของ I2C บัส จะเริ่มต้นด้วยการที่ตัวแม่สร้างสถานะเริ่มต้น (Start Condition) เพื่อขอใช้บัสจากนั้นจึงเริ่มการส่งรหัสควบคุม (Control Byte) เพื่อใช้ระบุตำแหน่งของแอดเดรสตัวลูกที่ต้องการใช้ติดต่อด้วยในระบบบัส โดยค่าตำแหน่งแอดเดรสนี้ อุปกรณ์แต่ละตัวจะมีรหัสแอดเดรสที่แตกต่างกันออกไป ไม่มีการซ้ำกันในระบบบัสเดียวกัน โดยรหัส Control Byte จะมีขนาด 8 บิตซึ่ง 7 บิตแรก เริ่มจาก MSB จะเป็นค่าตำแหน่งแอดเดรสของตัวลูก ส่วนบิตที่ 8 (LSB) จะเป็นบิตสุดท้ายของไบท์ที่ใช้สำหรับระบุทิศทางของการรับส่ง (R/W) โดยถ้าบิต LSB มีค่าเป็น “0” จะหมายถึงตัวแม่เขียนข้อมูลไปหาตัวลูก โดยข้อมูลจะรับส่งกันครั้งละ 1 ไบท์ และบิตท้ายข้อมูลของแต่ละไบท์ด้วยบิตแสดงการตอบรับ (Acknowledge Bit) โดยลักษณะโครงสร้าง Control Byte ของอุปกรณ์ แบบ I2C มีดังนี้



รูปที่ 2.8 ลักษณะของ Control Byte ของ I2C Bus [4]



สำนักงานคณะกรรมการวิจัยแห่งชาติ

วันที่ 23 พ.ย. 2555

เลขทะเบียน 250180

เลขเรียกหนังสือ

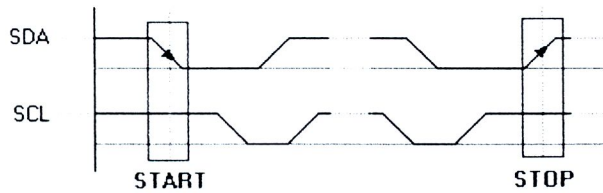
จากรูปที่ 2.8 จะเห็นได้ว่ารหัส Control Byte ของอุปกรณ์ I2C จะมีขนาด 8 บิต โดยที่บิตที่ 7 ถึงบิตที่ 4 จะเป็นรหัสประจำตัวของอุปกรณ์แต่ละตัวที่ถูกกำหนดไว้ตายตัวจากโรงงาน ซึ่งผู้ใช้งานต้องศึกษาข้อมูลจากคู่มือ Data sheet ของอุปกรณ์นั้นเองว่าอุปกรณ์ที่จะนำมาใช้งานมีรหัสประจำตัวเป็นเท่าใด ส่วนบิตที่ 3 ถึงบิตที่ 1 จะมีไว้สำหรับการเลือกต่ออุปกรณ์ที่อยู่ในบัส โดยค่าของทั้ง 3 บิตนี้ จะต้องเป็นค่าที่ ตรงกับการกำหนดสถานะทางลอจิก ให้กับขาสัญญาณของ A2,A1,A0 ของอุปกรณ์ด้วย ตัวอย่างเช่น อุปกรณ์ที่มีรหัสประจำตัวเป็น “0111” อาจถูกออกแบบให้สามารถต่อร่วมกันภายในบัสเดียวกันจำนวน 8 ตัว โดยการกำหนดสถานะของลอจิกให้กับขาสัญญาณ A2,A1,A0 ให้มีความแตกต่างกันจากวงจรที่ต่ออยู่

แต่อย่างไรก็ตาม อุปกรณ์ I2C บางตัว อาจถูกออกแบบให้ต่อได้ในระบบบัสเดียวกันเพียงตัวเดียวโดยไม่มีขาสัญญาณในการเลือกตำแหน่งของอุปกรณ์อยู่ด้วย ค่าของ Control Byte ในตำแหน่งบิตที่ 3 ถึงบิตที่ 1 ก็อาจถูกกำหนดไว้ตายตัวเป็น 000 เสมอก็ได้ ส่วนค่า Control Byte ของบิต 0 นั้นจะถูกกำหนดไว้เป็นทิศทางของข้อมูล โดยถ้าบิต 0 มีค่าเป็น 1 หมายถึงตัวแม่ต้องการอ่านค่าข้อมูลจากอุปกรณ์ ถ้ามีค่าเป็น 0 หมายถึงตัวแม่ต้องการเขียนค่า

สำหรับการรับส่งข้อมูลนั้นไม่มีการกำหนดตายตัวว่าจะต้องส่งกันครั้งละกี่ไบต์ ขึ้นอยู่กับข้อตกลงระหว่างอุปกรณ์แต่ละชนิดจะกำหนดขึ้น โดยในการรับส่งแต่ละครั้ง ตัวแม่จะเป็นตัวควบคุมการรับส่งเองทั้งหมด ซึ่งในกรณีที่ตัวแม่ต้องการติดต่อกับอุปกรณ์หลายๆตัวนั้น หลังจากตัวแม่สร้างสถานะเริ่มต้นเสร็จแล้ว และทำการติดต่อข้อมูลกับอุปกรณ์เรียบร้อยแล้ว ถ้าตัวแม่ต้องการติดต่อข้อมูลกับอุปกรณ์อื่นไม่จำเป็นต้องสร้างสถานะสิ้นสุด แต่ตัวแม่สามารถสร้างสถานะเริ่มต้นขึ้นมาใหม่พร้อมกับส่ง Control Byte ใหม่ไปติดต่อกับอุปกรณ์ที่ต้องการได้ทันที เมื่อทำการติดต่อรับส่งข้อมูลเรียบร้อยแล้วจึงสั่งหยุด Stop Condition เพื่อเป็นการเลิกใช้บัสที่มีอยู่และปล่อยให้บัสว่าง

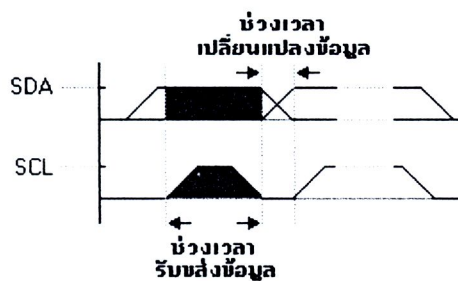
สถานะที่เกิดขึ้นบนบัส I2C มีด้วยกัน 5 สถานะ ดังนี้

- 1.) บัสว่าง (Bus not busy) สถานะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั้นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้
- 2.) เริ่มต้นการถ่ายทอดข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า สถานะเริ่มต้น (START)
- 3.) หยุดการถ่ายทอดข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า สถานะหยุด (STOP)



รูปที่ 2.9 เงื่อนไขของสภาวะเริ่มต้น และสิ้นสุดของ I2C Bus [4]

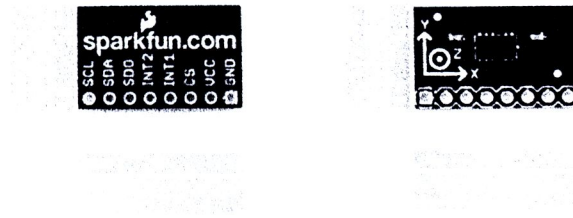
- 4.) ข้อมูลดำรงอยู่บนบัส (data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดเกิดความผิดพลาดขึ้น



รูปที่ 2.10 ช่วงเวลารับส่งบิตข้อมูลของ I2C BUS [4]

- 5.) รับข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับ เกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษ ซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจาก ตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนอง ให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

2.6 Accelerometer ADXL345

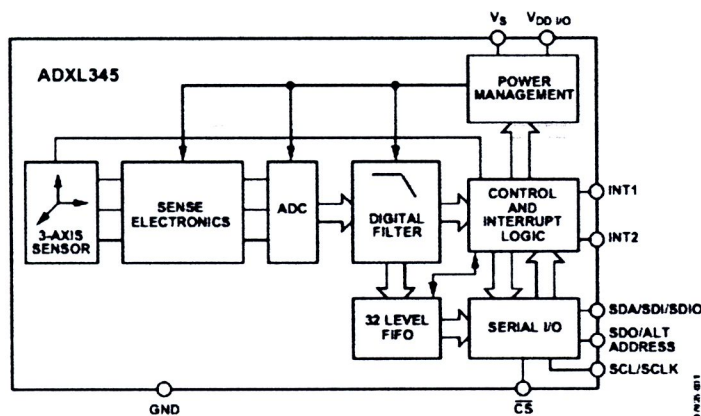


รูปที่ 2.11 เซนเซอร์วัดความเร่ง ADXL345 [5]

ADXL345 เป็นเซนเซอร์วัดความเร่งสามแกน มีขนาดเล็กบางและใช้กำลังน้อย เซนเซอร์นี้ให้ค่าความละเอียดสูงถึง 16g ซึ่งข้อมูลสามารถส่งได้ถึง 16 บิต สามารถส่งข้อมูลได้ทั้งการส่งข้อมูลแบบ SPI หรือ I2C อีกทั้ง ADXL345 สามารถนำมาประยุกต์ใช้กับ Application ในรูปแบบ Motion ได้ดีเพราะสามารถตอบสนองการเคลื่อนไหวได้อย่างละเอียดสูงถึง 3.9 mg/LSB นั้นหมายความว่าสามารถเปลี่ยนแปลงค่ามุมได้ครั้งละต่ำกว่า 1 องศา

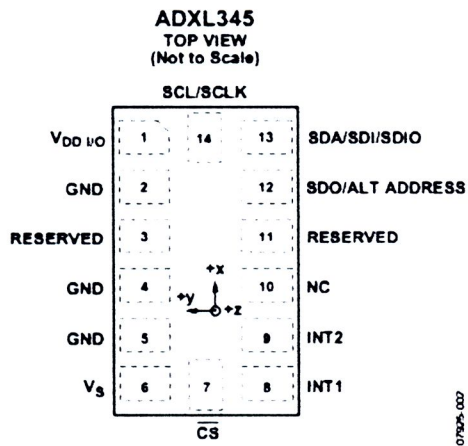
2.6.1 โครงสร้างภายในของ ADXL345

ภายในโครงสร้างของเซนเซอร์ ADXL345 มี ADC เพื่อแปลงจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล มีตัวกรองสัญญาณดิจิทัล รวมถึงการส่งข้อมูลสองแบบคือ แบบอินเทอร์รัพท์และแบบ 32 LEVEL FIFO อีกทั้งยังมีการส่งข้อมูลแบบอนุกรมได้ทั้งรูปแบบ SPI BUS และ I2C BUS นอกจากนี้ยังมีตัวควบคุมการเกิดการอินเทอร์รัพท์จากภายนอก และยังสามารถด้านจัดการทรัพยากรที่ใช้ภายในอีกด้วย



รูปที่ 2.12 โครงสร้างของ ADXL345 [5]

2.6.2 ฟังก์ชันการทำงานของขาต่างๆ ของ ADXL345



รูปที่ 2.13 โครงสร้างของ ADXL345 [5]

ตารางที่ 2.2 ตารางรายละเอียดฟังก์ชันขาต่างๆของ ADXL345 [5]

ชื่อ	รายละเอียด	ขา
VDD I/O	ไฟเลี้ยง -0.3 -3.9 volt	1
GND	ขา Ground ต่อสายดิน	2
RESERVED	ขาสำรองไฟ จะต่อเข้า Vs หรือเปิดว่างไว้	3
GND	ขา Ground ต่อสายดิน	4
GND	ขา Ground ต่อสายดิน	5
Vs	ไฟเลี้ยง -0.3 -3.9 volt	6
CS	Chip select	7
INT1	Interrupt 1 Output.	8
INT2	Interrupt 2 Output.	9
NC	ไม่มีการติดต่อ	10
RESERVED	ขาสำรองไฟ จะต่อเข้า Vs หรือเปิดว่างไว้	11
SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Address สำหรับ I2C	12
SDA/SDI/SDIO	Serial Data(I2C)/Serial Data Input(SPI 4-Wire)/Serial Data Input และ Output (SPI)	13
SCL/SCLK	Serial Data Output (SPI 4-	14

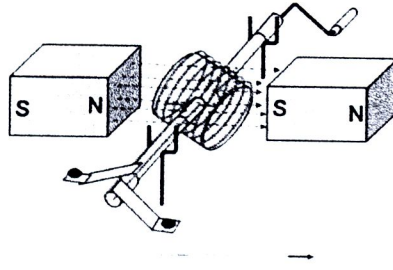
	Wire)/Alternate I2C Address Select (I2C)	
--	---	--

2.6.3 FEATURES

- Ultralow power มีค่าต่ำ 23 μA ในโหมดปกติและ 0.1 μA ในโหมด standby ที่ $V_S=2.5\text{ V}$
- Power consumption ควบคุมอัตโนมัติด้วย bandwidth
- ผู้ใช้สามารถเลือกใช้ความละเอียดได้ถึง 16 บิต
- ความละเอียดหลักที่ใช้งาน 10 bit
- Supply voltage range มี 2.0 V ถึง 3.6 V
- I/O voltage range มี 1.7 V ถึง V_S
- ส่งข้อมูลผ่าน serial port ใช้ได้ทั้ง SPI (3- และ 4-wire) และ I2C digital interfaces
- Wide temperature อยู่ในช่วง -40°C ถึง $+85^\circ\text{C}$
- 10,000 g shock survival
- ขนาดเล็ก 3 mm $\times$ 5 mm $\times$ 1 mm LGA package
- Bandwidth selectable via serial command
- Flexible interrupt modes mappable to either interrupt pin
- Measurement ranges selectable via serial command
- Wide temperature อยู่ในช่วง -40°C ถึง $+85^\circ\text{C}$

2.7 หลักการพื้นฐานของไดนาโม

เครื่องกำเนิดไฟฟ้าหรือไดนาโม (Generator or dynamo) สามารถเปลี่ยนพลังงานกลเป็นพลังงานไฟฟ้า โดยอาศัยการเหนี่ยวนำในขดลวดของเครื่องกำเนิดไฟฟ้าทำให้เกิดกระแสไฟฟ้าเกิดขึ้น ซึ่งสามารถจัดกระแสไฟได้สองแบบคือ กระแสตรง และกระแสสลับ พลังงานไฟฟ้าจากเครื่องกำเนิดไฟฟ้าเป็นไปตามกฎการอนุรักษ์พลังงาน โดยทั่วไปเครื่องกำเนิดไฟฟ้าจะประกอบด้วยขดลวดเหนี่ยวนำเคลื่อนที่สัมผัสกับแท่งแม่เหล็กทำให้เกิดไฟฟ้า ตัวอย่างการใช้งาน เช่น กรณีไฟฟ้าพลังน้ำ จะมีน้ำตกลงมาหมุนใบพัดของเครื่องกำเนิดไฟฟ้า



รูปที่ 2.14 การทำงานพื้นฐานของไดนาโม [6]

โดยสามารถคำนวณแรงเคลื่อนไฟฟ้าจากเครื่องกำเนิดไฟฟ้าได้ดังนี้

สมมติ ขดลวดมี N รอบ และแต่ละรอบมีพื้นที่ A ให้ขดลวดหมุนด้วยอัตราเร็วเชิงมุม รอบแกนที่ตั้งฉากกับสนามแม่เหล็ก B แรงเคลื่อนไฟฟ้าคือ

$$E = E_0 \sin(\omega t) \quad (2.1)$$

จากสมการที่ (2.1) เมื่อ E_0 คือแรงเคลื่อนไฟฟ้าสูงสุด

โดย
$$E_0 = NBA\omega \quad (2.2)$$

2.7.1 หลักการเหนี่ยวนำให้เกิดกระแสไฟฟ้าของไดนาโม

ไดนาโมเป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนพลังงานกลให้เป็นพลังงานไฟฟ้า มีส่วนประกอบสำคัญ ได้แก่ ขดลวดที่พันอยู่รอบแกน เรียกว่า อาเมเจอร์ (armature) แม่เหล็ก 2 แท่งหันขั้วต่างกันเข้าหากัน เพื่อให้เกิดสนามแม่เหล็กโดยจะมีเส้นแรงแม่เหล็กพุ่งจากขั้วเหนือไปยังขั้วใต้ และบริเวณขั้วจะมีความเข้มของสนามแม่เหล็กมากกว่าบริเวณอื่นๆ

หลักการเหนี่ยวนำให้เกิดกระแสไฟฟ้าจากไดนาโมอาจทำได้ดังนี้ การหมุนขดลวดตัดสนามแม่เหล็กจะทำให้สนามแม่เหล็กเปลี่ยนแปลงจึงเกิดกระแสไฟฟ้าขึ้น ไมเคิล ฟาราเดย์ (Michael Faraday) นักวิทยาศาสตร์ชาวอังกฤษ (พ.ศ.2334-2410) เป็นผู้ค้นพบหลักการที่ว่า “กระแสไฟฟ้าเหนี่ยวนำเกิดจากการเปลี่ยนแปลงสนามแม่เหล็กที่ผ่านขดลวด” หากต้องการสร้างไดนาโมให้สามารถผลิตกระแสไฟฟ้าได้มากขึ้น สามารถทำได้โดยเพิ่มจำนวนรอบของขดลวด และหมุนขดลวดให้เร็วขึ้น

ไดนาโมสามารถแบ่งออกเป็น 2 ชนิดคือ

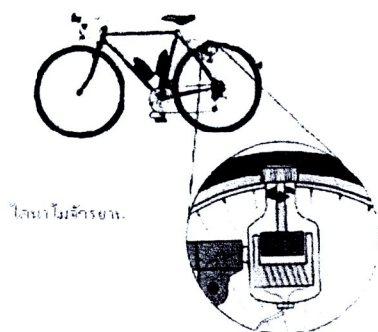
1. ไดนาโมไฟฟ้ากระแสสลับ AC Dynamo

ประกอบด้วยแท่งแม่เหล็ก 2 แท่ง ขดลวดและแหวนลื่นโดย แหวนลื่น 2 วงสัมผัสกับแปรงตัวนำไฟฟ้าซึ่งจะรับกระแสไฟฟ้าจากขดลวดออกสู่วงจรภายนอก

2. ไดนาโมไฟฟ้ากระแสตรง DC Dynamo

ประกอบด้วยแท่งแม่เหล็ก 2 แท่ง ขดลวดและแหวนแยกโดย แหวนแยกแต่ละอัน

สัมผัสกับแปรงตัวนำไฟฟ้าซึ่งจะรับกระแสไฟฟ้าจากขดลวดออกสู่วงจรภายนอก โดย ไดนาโมกระแสตรง (Direct current dynamo) หมายถึง ไดนาโมที่ผลิตไฟกระแสตรง (DC) ส่วนประกอบเหมือนกับไดนาโมกระแสสลับทุกอย่าง แตกต่างที่วงแหวนเท่านั้น



รูปที่ 2.15 ไดนาโมจักรยาน [7]

2.8 วงจรไฟฟ้าอนุกรม

สมการแบ่งแรงดัน (Voltage Divider Equation) หมายถึง วงจรอนุกรมที่ใช้ตัวต้านทานอย่างน้อย 2 ตัวเป็นตัวแบ่งแรงดันไฟฟ้า

$$(2.3) \quad V_{R_n} = V_T \frac{R_n}{R_T}$$

โดย V_{R_n} คือ แรงดันตกคร่อมตัวต้านทานใดๆ

V_T คือ แรงดันของแหล่งจ่ายไฟฟ้า

R_n คือ ตัวต้านทานใดๆ ในวงจร

R_T คือ ค่าความต้านทานรวมของวงจร

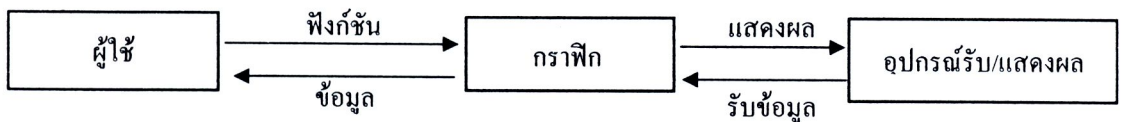
2.9 โปรแกรมออกแบบกราฟิก Blender

โปรแกรม Blender เป็นโปรแกรมที่นำเอาความสามารถต่างๆ ในการสร้างสรรค์ไฟล์รูปแบบสามมิติมารวมกัน เช่น การขึ้นรูปโมเดล , เรนเดอร์ , แอนิเมชัน ฯลฯ โดย Blender สามารถทำงานได้ในระบบปฏิบัติการที่หลากหลาย และมีขนาดไฟล์ติดตั้งที่ไม่มากนักเมื่อเทียบกับโปรแกรมกราฟิกสามมิติตัวอื่นๆ เนื่องจาก Blender มีรูปแบบซอฟต์แวร์แบบ Open Source เพราะตัวดั้งเดิมถูกสร้างขึ้นโดยบริษัท Not a Number หรือที่รู้จักกันในชื่อของ NaN ในปัจจุบันได้

	G5,Powerbook G4,iMac G5 Linux glibc 2.3.6,includes FFMPG Suits
- ความเร็วซีพียู	700 MHz หรือสูงกว่า
- RAM	512 MB
- พื้นที่สำหรับลงโปรแกรม	27 MB
- การ์ดแสดงผล	1024×768 16-bit color ,RAM 8 MB (แนะนำให้รองรับมาตรฐาน OpenGL)
- ระบบและอุปกรณ์อื่นๆ	เมาส์แบบ 3 ปุ่ม

2.9.3 ฟังก์ชันทางกราฟิก (Graphic Functions)

การทำงานของระบบกราฟิก ผู้ใช้ไม่จำเป็นต้องทราบว่าการทำงานต้องทำอะไร รู้เพียงแต่ อินพุต (input) และเอาต์พุต (output) ก็เพียงพอ โดยในระบบกราฟิกจะมีฟังก์ชันที่ถูกเรียกใช้ โดยมีการรับค่าของอินพุตจากอุปกรณ์ต่างๆ เช่น เมาส์หรือคีย์บอร์ด หรืออุปกรณ์อื่นๆ เช่น แมสเสจจากระบบปฏิบัติการ หรืออินเทอร์รัพท์ก็ได้ ผลลัพธ์ที่ได้จากการทำงานจะแสดงออกไปยังอุปกรณ์ในการแสดงผล เช่น จอภาพ CRT หรือจอ LCD ทั้งนี้เอาต์พุตอาจจะมองอินพุตเป็นฟังก์ชันที่ผู้ใช้เรียกใช้และเอาต์พุตคือ ผลของการทำงานที่แสดงไปยังจอภาพ ดังรูปด้านล่าง



รูปที่ 2.15 ระบบกราฟิกที่เปรียบเสมือนกล่องดำ (Black Box) [8]

2.10 โปรแกรมภาษา Python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่งที่มีอยู่ในตอนนี้ โดยความสามารถของภาษานี้สูงไม่แพ้ภาษาอื่นๆ Python นั้นเป็นภาษาชนิด Open Source ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมได้ฟรีโดยไม่ต้องเสียค่าใช้จ่าย เนื่องจากความเป็น Open Source จึงทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และสามารถใช้งานได้ครอบคลุมกับทุกลักษณะงาน

Python ถูกสร้างขึ้นโดย Guido Van Rossum โค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์คือจะประมวลผลไปทีละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันแรกคือเวอร์ชัน 0.9.0 ออกมาเมื่อปี 2533 และเวอร์ชันปัจจุบัน

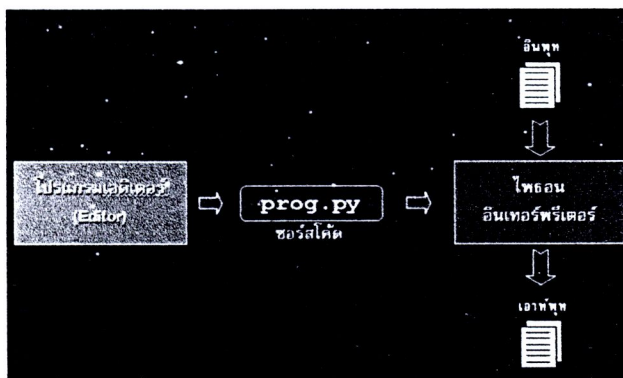
คือ 2.5.2 ออกเมื่อวันที่ 21 กุมภาพันธ์ 2551 และได้มีการพัฒนา Python ในรุ่นที่ 3 คือ Python 3000 หรือ Py3k โดยจะมีการปรับปรุงใหม่เกือบหมด และตอนนี้อยู่ในระหว่างการทดลองอยู่

2.10.1 จุดเด่นของโปรแกรมภาษา Python

โปรแกรมภาษา Python เป็นโปรแกรมภาษาแบบ Open Source ที่สามารถดาวน์โหลดมาใช้งานได้ฟรี และยังมีความสามารถทัดเทียมกับโปรแกรมภาษาอื่นๆโดยจุดเด่นที่น่าสนใจของโปรแกรมนี้นี้

- ภาษา Python สนับสนุนแนวคิดแบบออบเจกต์โอเรียนเทด หรือ OOP (Object Oriented Programming)
- Python เป็นภาษาคอมพิวเตอร์ที่ไม่คิดมูลค่าการใช้งานและเป็นภาษาที่มีความยืดหยุ่นสูงมาก
- โค้ดที่เขียนด้วย Python สามารถนำไปรันบนระบบปฏิบัติการอื่นๆ ได้ เช่น Linux, Ms-windows (95, 98, NT, 2000, XP), Amiga, Be-OS, OS/2, VMS, QNX, และระบบอื่นๆอีกมากมาย
- Python สนับสนุนเทคโนโลยี COM ของ Ms-windows
- Python รวมมาตรฐานการอินเทอร์เฟซ Tkinter ซึ่งสนับสนุนบนระบบ Windows, Ms-windows และ Macintosh การใช้คำสั่ง Tkinter API ช่วยให้โปรแกรมเมอร์ไม่ต้องแก้ไขโค้ดเมื่อนำไปรันบนระบบปฏิบัติการอื่นๆ
- Python เป็น Dynamic typing คือ สามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก
- Python มี Built-in Object Types คือ โครงสร้างของข้อมูลที่สามารถใช้ได้ภายใน Python ประกอบด้วย ลิสต์, ดิกชันนารี, สตริง ที่ง่ายต่อการใช้งานและมีประสิทธิภาพสูง
- Python มีเครื่องมือต่างๆ มากมาย เช่น การประมวลผลเท็กซ์ไฟล์ การเรียงข้อมูล การเชื่อมต่อสตริง การตรวจสอบเงื่อนไขของข้อความ การแทนค่า เป็นต้น
- Python มีโมดูลสำหรับจัดการ Regular Expression
- Python มีโมดูลที่สร้างขึ้นจากนักพัฒนาสนับสนุนมากมาย ได้แก่ COM, image, CORBA, ORBs, XML เป็นต้น
- Python จัดการหน่วยความจำอย่างอัตโนมัติ สามารถจัดการพื้นที่หน่วยความจำที่ไม่ต่อเนื่องให้ทำงานได้อย่างมีประสิทธิภาพ
- Python อนุญาตให้ฝังชุดคำสั่งของ Python เอาไว้ภายในโค้ดภาษา C/C++ ได้
- Python อนุญาตให้โปรแกรมเมอร์สร้าง Dynamic Link Library (DLL) เพื่อใช้ร่วมกับ Python

- Python มีโมดูลสนับสนุนเกี่ยวกับเน็ตเวิร์ก โปรเซส เรด regular, expression, xml, GUI และอื่นๆ
- Python ประกอบด้วยโมดูลสำหรับสร้าง Internet Script และติดต่อกับอินเทอร์เน็ตผ่าน Sockets, และทำหน้าที่เป็น CGI Script ตลอดจนใช้งานคำสั่ง FTP, Gopher, XML และอื่นๆอีกมาก
- Python สามารถประมวลผลทางด้านวิทยาศาสตร์ และวิศวกรรมศาสตร์ได้อย่างมีประสิทธิภาพ
- Python มีฟังก์ชันสนับสนุนฐานข้อมูล เช่น MySQL, Sybase, Oracle, Informix, ODBC และอื่นๆ
- Python มีไลบรารีสนับสนุนด้านการสร้างภาพกราฟิก เช่น ทำภาพเบลอ หรือภาพชัด หรือเขียนข้อความบนภาพ ตลอดจนบันทึกไฟล์ในรูปแบบต่างๆ ได้อย่างสะดวกและมีประสิทธิภาพ
- Python มีไลบรารีสนับสนุนด้านปัญญาประดิษฐ์
- Python มีไลบรารีสำหรับสร้างเอกสาร PDF โดยไม่ต้องติดตั้ง Acrobat Writer
- Python มีไลบรารีสำหรับสร้าง Shockwaves Flash (SWF) โดยไม่ต้องติดตั้ง Macromedia Flash



รูปที่ 2.16 ขั้นตอนการทำงานของโปรแกรม Python [9]

2.10.2 โปรแกรมภาษา Python ในแพลตฟอร์มต่างๆ

ผู้เขียนโปรแกรมภาษา Python สามารถเลือกใช้แพลตฟอร์มได้ดังนี้

CPython (ซีไพทอน) : เป็นแพลตฟอร์มภาษาไพทอนดั้งเดิม โปรแกรมอินเทอร์เน็ตถูกเขียนโดยภาษาซี ซึ่งคอมไพล์ใช้ได้บนหลายระบบปฏิบัติการ เช่น Windows, Linux, Unix การใช้งานสามารถทำได้โดยการติดตั้งโปรแกรมอินเทอร์เน็ตและแพ็คเกจที่จำเป็นต่างๆ

Jython (ไจทอน) : เป็นแพลตฟอร์มภาษาไพทอนที่ถูกพัฒนาบนแพลตฟอร์มจาวา เพื่อเพิ่มอำนวยความสะดวกในการใช้ความสามารถภาษาสคริปต์ของไพทอนลงในซอฟต์แวร์จาวาอื่นๆ การใช้งานสามารถทำได้โดยการติดตั้งจาวาและเรียนไลบรารีของไจทอนซึ่งมาในรูปแบบไบนารีเพื่อใช้งาน

Python.NET (ไพทอนดอตเน็ต) : เป็นการพัฒนภาษาไพทอนให้สามารถทำงานบนดอตเน็ตเฟรมเวิร์กของไมโครซอฟท์ได้ โดยโปรแกรมที่ถูกเขียนจะถูกแปลงเป็น CLR ปัจจุบันมีโครงการที่นำภาษาไพทอนมาใช้งานบน .NET Framework ของไมโครซอฟท์แล้วคือโครงการ IronPython

2.10.3 ไลบรารีภาษา Python

การเขียนโปรแกรมในภาษาไพทอนโดยใช้ไลบรารีต่างๆ เป็นการลดภาระของโปรแกรมเมอร์ได้เป็นอย่างดี ทำให้โปรแกรมเมอร์ไม่ต้องเสียเวลากับการเขียนคำสั่งที่ซ้ำๆ เช่นการแสดงผลข้อมูลออกสู่หน้าจอ หรือการรับค่าต่างๆ

ไพทอนมีชุดไลบรารีมาตรฐานมาให้ตั้งแต่อินเตอร์พรีเตอร์ นอกจากนั้นยังมีผู้พัฒนาจากทั่วโลกดำเนินการพัฒนาไลบรารีซึ่งช่วยอำนวยความสะดวกในด้านต่างๆ โดยจะเผยแพร่ในรูปแบบของแพ็คเกจต่างๆ ซึ่งสามารถติดตั้งเพิ่มเติมได้ ดังนี้

- wxPython : อีกทางเลือกหนึ่งสำหรับเขียนส่วนติดต่อกับผู้ใช้กับกราฟิก ซึ่งสามารถใช้ได้หลายระบบปฏิบัติการ
- SciPy : รวมโครงสร้างข้อมูลและการคำนวณต่าง ๆ ที่จำเป็นต้องใช้ในการเขียนโปรแกรมคำนวณทางวิทยาศาสตร์
- py2exe : ใช้สำหรับแปลงโปรแกรมที่เขียนในภาษาไพทอนให้อยู่ในรูปแบบของระบบปฏิบัติการวินโดวส์
- PyWin32 : ใช้สำหรับติดต่อเรียกใช้บริการบนระบบปฏิบัติการวินโดวส์และคลาสใน Microsoft Foundation Classes: MFC
- MySQLdb : ใช้สำหรับติดต่อกับระบบฐานข้อมูล MySQL
- psycopg2 : ใช้สำหรับติดต่อกับระบบฐานข้อมูล โพสต์เกรสคิวเอล
- PyGTK : GTK+ สำหรับ Python ใช้สำหรับสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก ซึ่งสามารถใช้ได้หลายระบบปฏิบัติการ
- PyQt : Qt สำหรับ Python ใช้สำหรับสร้างส่วนติดต่อกับผู้ใช้แบบกราฟิก ซึ่งสามารถใช้ได้หลายระบบปฏิบัติการ

2.10.4 ตัวอย่างการนำไปใช้ของโปรแกรมภาษา Python

จะเห็นได้ว่าจุดเด่นของโปรแกรมภาษา Python มีอยู่มากและใช้งานได้อย่างรวดเร็ว โดยโปรแกรมภาษา Python ได้ถูกนำไปใช้ประโยชน์ในต่างๆมากมาย เช่น

- Red Hat ใช้ Python เป็นเครื่องมือสำหรับการติดตั้งระบบปฏิบัติการ Linux (Installer)
- Google สร้างขึ้นด้วย Python
- Infoseek ใช้ Python จัดการ web search products
- Yahoo! ใช้ Python ในการจัดการด้าน Internet services
- NASA ใช้ Python สำหรับ mission-control-system
- Lawrence Livermore Labs ใช้ Python สำหรับงาน numeric programming
- Industrial Light and Magic ใช้ Python สร้างภาพแอนิเมชัน