บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ข้อมูลเบื้องต้นของบอร์ด FPGA

บอร์ด WARRIOR CYCLONE3 – EB01/02 เป็นบอร์ดทดลอง FPGA ที่มาพร้อมกับเทคโนโลยี 65 นาโนเมตรของชิป Cyclone III จาก ALTERA บอร์ดเหมาะสำหรับผู้ใช้งานในทุกระดับตั้งแต่เริ่มต้นจนถึงขั้นสูง สามารถใช้งาน FPGA ในรูปแบบการเขียนวงจรภาษา VHDL และ ยังให้ความสามารถในอีกระดับของการใช้งาน FPGA กับซอฟต์ไมโครโปรเซสเซอร์อย่าง NIOS II เป็นการผสม FPGA กับไมโครคอนโทรลเลอร์ ขนาด 32 บิต ที่สามารถกำหนดคุณสมบัติภายในตัวของไมโครโปรเซสเซอร์ได้ด้วยตัวคุณเอง

รูปภาพ 2.1 ภาพของชิพ Cyclone III
http://www.astronlogic.com/content/product/WarriorCyclone3_EB1.php

รูปภาพ 2.2 ภาพของบอร์ดรุ่น Warrior LAB01
http://www.astronlogic.com/content/product/WarriorCyclone3_EB1.php

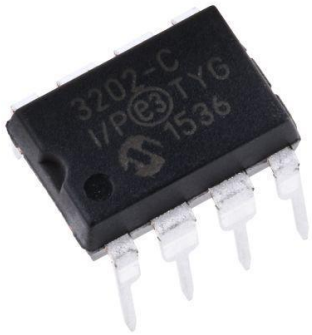
สำหรับในส่วนของตัว LAB

เป็นส่วนที่มีการติดตั้งอุปกรณ์รอบข้างรวมถึงแหล่งจ่ายไฟที่จำเป็นสำหรับโมดูล DEV

หรือตัวชิพนั่นเอง โดยอุปกรณ์รอบข้างต่าง ๆ สำหรับทดลองมีดังนี้

1. โมดูล LCD แบบ 16 ตัวอักษร 2 บรรทัด พร้อมไฟส่องหลัง (Backlight) LED ขนาด 8 บิต
2. รีเลย์ 1 หน้าสัมผัส
3. Buzzer
4. หน่วยความจำแบบ IC2 ขนาด 32 Kbit และ Expansion I2C Port ขนาด 3.3V
5. Serial Port
6. VGA Port สำหรับทดลองเชื่อมต่อกับจอมอนิเตอร์
7. PS/2 Port สำหรับทดลองเชื่อมต่อกับคีย์บอร์ด และเมาส์
8. Oscillator ขนาด 50 MH_z (การสร้างความถี่ภายในอื่น ๆ สามารถใช้วงจรเฟสล็อกลูปซึ่งสร้างจาก MegaCore Wizard ของ Quartus II ได้)
9. สวิตช์เลื่อน 8 บิต
10. สวิตช์กดติด-ปล่อยดับ 4 บิต (Active Low)
11. สวิตช์รีเซ็ต (สำหรับ NIOS II Soft-Core Processor)
12. Expansion Port A ขนาด 14 บิต แบบอิสระ สำหรับเชื่อมต่ออุปกรณ์ทั่วไป
13. Expansion Port B ขนาด 38 บิต แบบอิสระ สำหรับเชื่อมต่ออุปกรณ์ทั่วไป หรือโมดูล SRAM และSDRAM เพื่อการใช้งาน NIOS II Soft-Core Processor

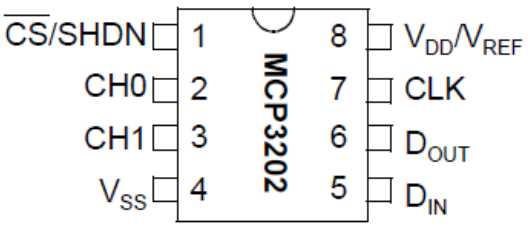
ข้อมูลเบื้องต้นของ MCP3202



รูปภาพ 2.3 ภาพของ Microchip MCP3202
http://th.rs-online.com/web/p/general-purpose-adcs/0403115/

MCP3202 เป็นไอซีตระกูล ADC หรือ Analog-to-Digital Converter ใช้สำหรับแปลงสัญญาณ Analog ที่ป้อนเข้ามาให้เป็นข้อมูล Digital ผ่านบัส SPI มีความละเอียดของข้อมูลสูงสุดคือ 12-bit ซึ่งสามารถอ่านค่าแรงดัน Analog โดยมีรายละเอียดดังนี้

- ขนาดของข้อมูล (bit resolution): 12-bit (ค่าที่อ่านได้อยู่ในช่วง 0..4095)
- ขา Analog-Input สามารถเลือกใช้งานทั้งแบบ single-ended หรือ pseudo-differential pair
- สามารถรับค่าแรงดันได้ 2-channel
- เชื่อมต่อแบบ SPI โหมด 0,0 หรือ 1,1 (SPI Mode 0 or 3)
- ใช้แรงดันไฟเลี้ยงในช่วง 2.7V ถึง 5.5V
- อัตราการแปลงข้อมูลสูงสุด (max. sampling rate, ksps = 1000 Samples-per-second):
  100 ksps @Vdd = 5 V,
  50 ksps @Vdd = 2.7 V



รูปภาพ 2.4 ภาพขาของ IC MCP3202

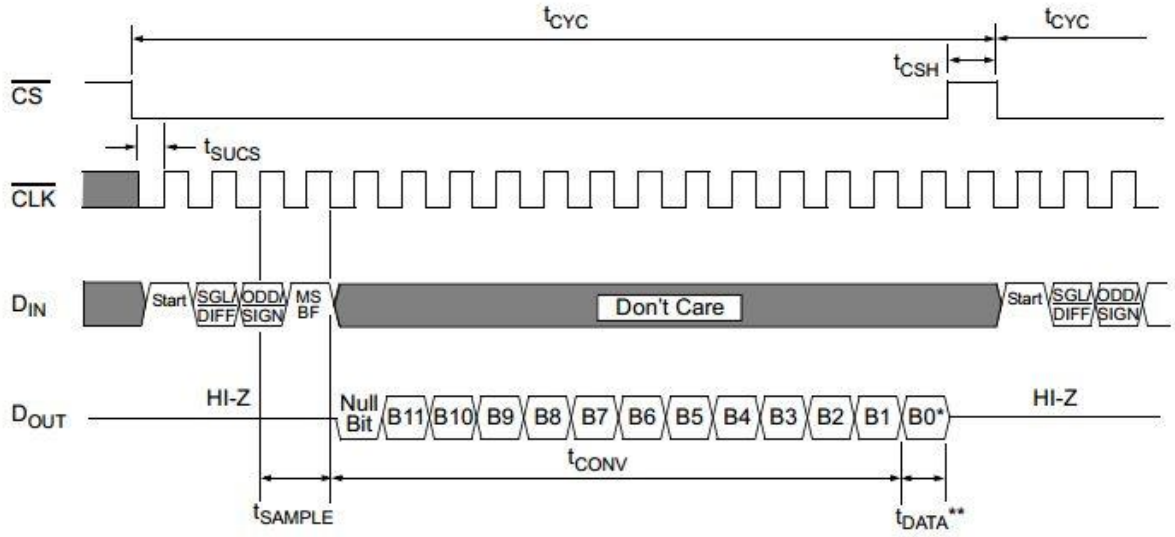
http://cpre.kmutnb.ac.th/esl/learning/index.php?article=mcp320x-adc-spi

ขาของ MCP3202 ทั้ง 8 มีการทำงานดังนี้

- Pin1:**#CS/SHDN** เป็นสัญญาณอินพุต Chip Select (active-low) --- นำไปต่อกับขา Arduino /SS
- Pin2:**CH0** เป็นอินพุตแอนะล็อกช่องที่0 (Analog Input, Channel 0)
- Pin3:**CH1** เป็นอินพุตแอนะล็อกช่องที่1 (Analog Input, Channel 1)
- Pin4:**Vss** เป็นขาสำหรับต่อกับ Gnd ของระบบ
- Pin5:**DIN** เป็นอินพุตดิจิทัลสำหรับ SPI (Serial Data In) --- นำไปต่อกับขา Arduino MOSI
- Pin6:**DOUT** เป็นเอาต์พุตดิจิทัลสำหรับ SPI (Serial Data Out) --- นำไปต่อกับขา Arduino MISO
- Pin7:**CLK** เป็นอินพุตสำหรับสัญญาณ CLK (Serial Clock) สำหรับ SPI --- นำไปต่อกับขา Arduino SCK
- Pin8:**Vdd/Vref** เป็นแรงดันไฟเลี้ยงและใช้เป็นแรงดันอ้างอิงด้วย (2.7V .. 5.5V)

จากตารางข้างต้นใช้สำหรับกำหนด 2-bit เพื่อใช้ควบคุมการอ่านค่าแรงดันแต่ละ Channel

	Config Bits		Channel Selection		GND
	SGL/DIFF	ODD/SIGN	0	1	
Single-Ended Mode	1	0	+	—	-
	1	1	—	+	-
Pseudo-Differential Mode	0	0	IN+	IN-	
	0	1	IN-	IN+	



รูปภาพ 2.5 ภาพตารางและการทำงานของบัส SPI

โดย 2-bit ที่ทำการกำหนดข้างต้นจะนำมารวมกับ Start ซึ่งเป็นส่วนหัว และ MSBF (ส่งหลักนัยสำคัญสูงสุดก่อน) ซึ่งเป็นส่วนท้ายแล้วส่งไปที่ MCP3202 เพื่อทำการแปลงค่าแรงดัน Analog ตามช่องสัญญาณที่เราเลือกไปแล้วให้ส่งกลับมาเป็นข้อมูล Digital 12-bit เพื่อนำไปใช้โดยการกำหนด Channel จะแบ่ง Header เป็น 2 ค่าคือ 1101 สำหรับอ่านค่าแรงดันจาก Channel 0 และ 1111 สำหรับอ่านค่าแรงดันจาก Channel 1 โดยภาพตารางการเลือก Channel และภาพการทำงานของ SPI นั้นสามารถศึกษาข้อมูลเพิ่มเติมได้จาก Datasheet ที่อยู่ในลิ้งค์ข้างต้น

MCP3202 Datasheet : ww1.microchip.com/downloads/en/DeviceDoc/21034D.pdf

Processing

รูปภาพ 2.6 รูป icon processing
https://forum.processing.org/processing-org.jpg

Processing ไม่ใช่ภาษาใหม่ แต่เป็นการกำหนดรูปแบบของการเขียนโค้ดหรือที่เรียกว่า Sketch โดยอาศัยภาษา Java เป็นพื้นฐาน ซึ่งภาษา Processing ก็มีข้อดีดังนี้

- ใช้ได้กับระบบปฏิบัติการ Windows, Linux, Mac OS X
- เป็นซอฟต์แวร์ประเภท Opensource (เปิดเผยโค้ดต้นฉบับ)
- สามารถใช้สร้างรูปกราฟิกแบบ 2D และ 3D (2 และ 3 มิติ) หรือแบบมีปฏิสัมพันธ์ (Interactive) กับผู้ใช้ได้ เช่น ในการเรียนรู้การสร้างเกมส์คอมพิวเตอร์เบื้องต้น หรือการนำเสนอข้อมูลในรูปแบบต่างๆ (Data Visualization)
- ทำให้ผู้เรียนได้เห็นความเชื่อมโยงระหว่างการเขียนโค้ดและสิ่งที่ปรากฏเห็นได้อันเป็นผลมาจากการทำงานของโปรแกรม
- รองรับการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming)
- มีตัวอย่างการใช้งานมากมาย (ลองดูได้จาก http://openprocessing.org/) และแหล่งข้อมูลอ้างอิง รวมทั้งหนังสือให้ศึกษาได้
- มีความเชื่อมโยงกับภาษา Java และเป็นพื้นฐานในการเรียนรู้ภาษา Java ต่อไป

อ้างอิงจาก http://cpre.kmutnb.ac.th/blog/เรียนรู้การเขียนโปรแกร/

Serial / Libraries

Serial เป็น Libraries หนึ่ง ของ Processing
ซึ่งสามารถช่วยในการจัดการ การอ่านและการเขียน ข้อมูลจาก อุปกรณ์ภายนอก โดย

จัดการด้วยอัตรา 1 Byte/1 ครั้งและยังอนุญาติให้คอมพิวเตอร์ 2 เครื่องรับและส่งข้อมูลได้รวมถึงMicrocontroller อีกด้วย โดยเมื่อเราต้องการใช้งาน Libraries นี้ใน processing เราต้องทำการ Import ดังนี้

```
import processing.serial.*;
```

โดยสามารถอ่านรายละเอียดและการใช้งานเพิ่มเติมได้ที่

- https://processing.org/reference/libraries/serial/index.html

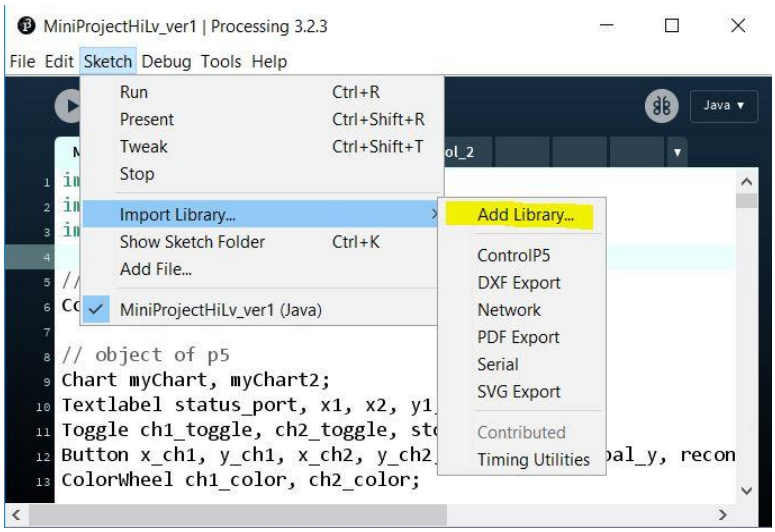
ControlP5 / Libraries

ControlP5 เป็น Libraries หนึ่งของ Processing ที่ถูกพัฒนาโดย Andreas Schlegel โดยที่ Libraries ตัวนี้นั้นจะเน้นไปในทางเสริมสร้างความสดวกสบายในการเขียนโปรแกรมภาษา

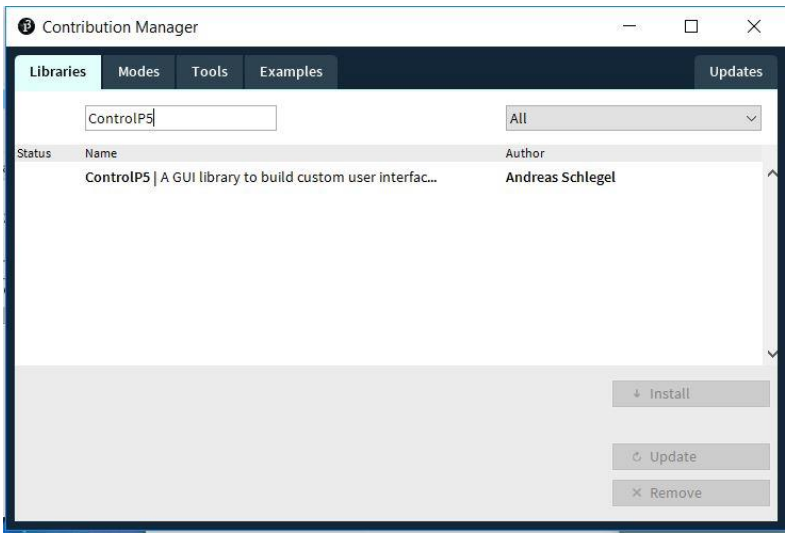
processing ในส่วนของการทำ Gui ปุ่มกด หรือการทำกราฟ เป็นต้น โดยการใช้งานต้องทำการ Import ดังนี้

```
import controlP5.*;
```

แต่ว่าก่อนหน้านั้นต้องไปทำการดาวน์โหลดก่อน



รูปภาพ 2.7



รูปภาพ 2.8

ศึกษาเพิ่มเติมได้ที่ http://www.sojamo.de/libraries/controlP5/

รูปแบบการรับและส่งข้อมูล

ในการทำงานครั้งนี้ได้มีการนำวิธีการส่งข้อมูล (Data Transmission) มาประยุกต์ใช้ซึ่งแบ่งเป็น 2 แบบหลักๆ คือ

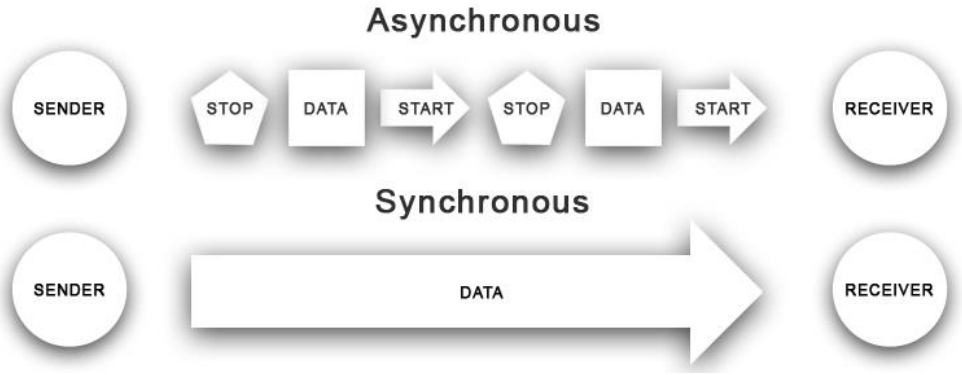
- การส่งข้อมูลแบบ Asynchronous
- การส่งข้อมูลแบบ Synchronous

การส่งข้อมูลแบบ Asynchronous

เป็นการส่งข้อมูลที่ทั้งตัวของผู้รับและผู้ส่งไม่จำเป็นต้องใช้สัญญาณ Clk ร่วมกัน
แต่ข้อมูลที่จะรับมานั้นต้องมีการตกลงกันกับผู้ส่งก่อน
ดังนั้นผู้ส่งจึงจำเป็นต้องแจ้งให้ผู้รับทราบก่อนว่าจะมีการส่งข้อมูลไปให้โดยเพิ่มบิตที่เรียกว่า Start bit
เพื่อให้ผู้รับทราบว่ามีการส่งข้อมูลแล้ว และ Stop bit เพื่อให้ผู้รับทราบจุดสิ้นสุดของการส่งข้อมูล

การส่งข้อมูลแบบ Synchronous

เป็นการส่งข้อมูลที่ทั้งตัวของผู้รับและผู้ส่งต้องทำการกำหนดจังหวะการรับและส่งของข้อมูลด้วยสัญญาณ Clk ทำให้การส่งรูปแบบนี้ไม่จำเป็นต้องใช้ Start bit และ Stop bit



รูปภาพ 2.8 ภาพการส่งข้อมูลทั้ง 2 แบบ
https://www.eyerys.com/articles/news/meet-project-adam-microsofts-artificial-brain-deep-learning

ทิศทางของการสื่อสารข้อมูล

ทิศทางการสื่อสารของข้อมูลนั้นสามารถแบ่งได้ 3 รูปแบบ ดังนี้

- การสื่อสารแบบทิศทางเดียว (Simplex)
- การสื่อสารแบบกึ่งสองทิศทาง (Half Duplex)
- การสื่อสารแบบสองทิศทาง (Full Duplex)

การสื่อสารแบบทิศทางเดียว (Simplex)

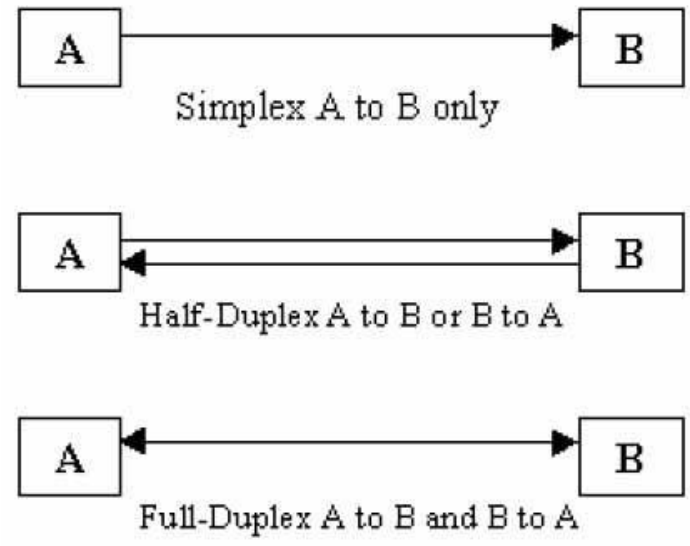
เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลจะถูกส่งจากทิศทางหนึ่งไปยังอีกทิศทางโดยไม่สามารถส่งข้อมูลย้อนกลับมาได้ เช่น ระบบวิทยุ หรือโทรทัศน์, การส่งข้อมูลจาก Mouse หรือ Keyboard เป็นต้น

การสื่อสารแบบกึ่งสองทิศทาง (Half Duplex)

เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลสามารถส่งสลับกันได้ 2 ทิศทาง แต่ไม่สามารถส่งพร้อมกันทั้ง 2 ฝั่งได้ โดยต้องผลัดกันส่งครั้งละทิศทางเท่านั้น เช่น วิทยุสื่อสารแบบผลัดกันพูด

การสื่อสารแบบสองทิศทาง (Full Duplex)

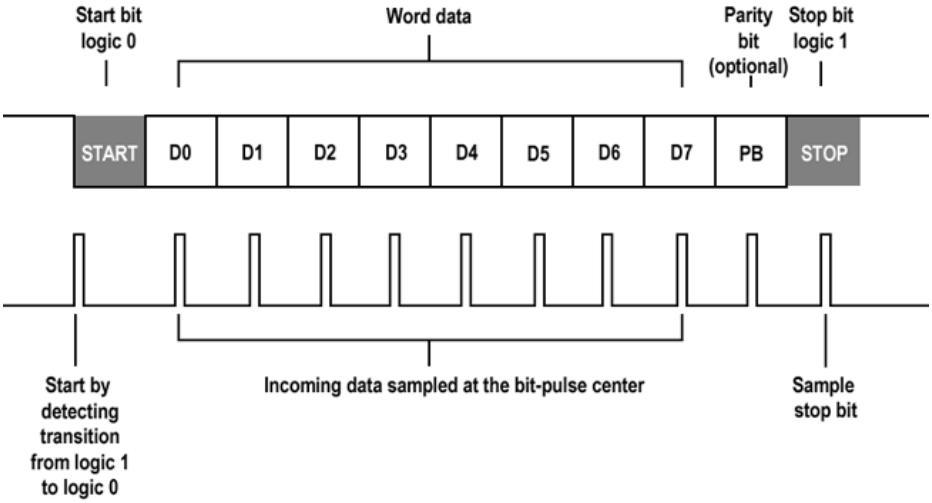
เป็นทิศทางการสื่อสารข้อมูลแบบที่ข้อมูลสามารถส่งพร้อม ๆ กันได้ทั้ง 2 ทิศทาง ในเวลาเดียวกัน เช่น ระบบโทรศัพท์



รูปภาพ 2.9 รูปภาพทิศทางการส่งข้อมูลทั้ง 3 รูปแบบ

http://cedtinet.blogspot.com/2013/02/transmission-modes.html

UART (Universal Asynchronous Receiver Transmitter)



รูปภาพ 2.10 รูปแบบ Uart Data Frame

การส่งข้อมูลแบบอนุกรมทีละบิตตามลำดับโดยในหนึ่งเฟรมของการส่งข้อมูลจะประกอบด้วย

- Start Bit (บิตเริ่มต้นการส่งข้อมูล) จำนวน 1 บิต
- Data Bits (บิตข้อมูล) จำนวน 5 - 8 บิต
- Parity Bit (บิตสำหรับค่าพาริตี้) จำนวน 1 บิต (หรืออาจจะไม่มีก็ได้)
- Stop Bit (บิตจบการส่งข้อมูล) สามารถเลือกความยาวได้ (1 หรือ 2 บิต)

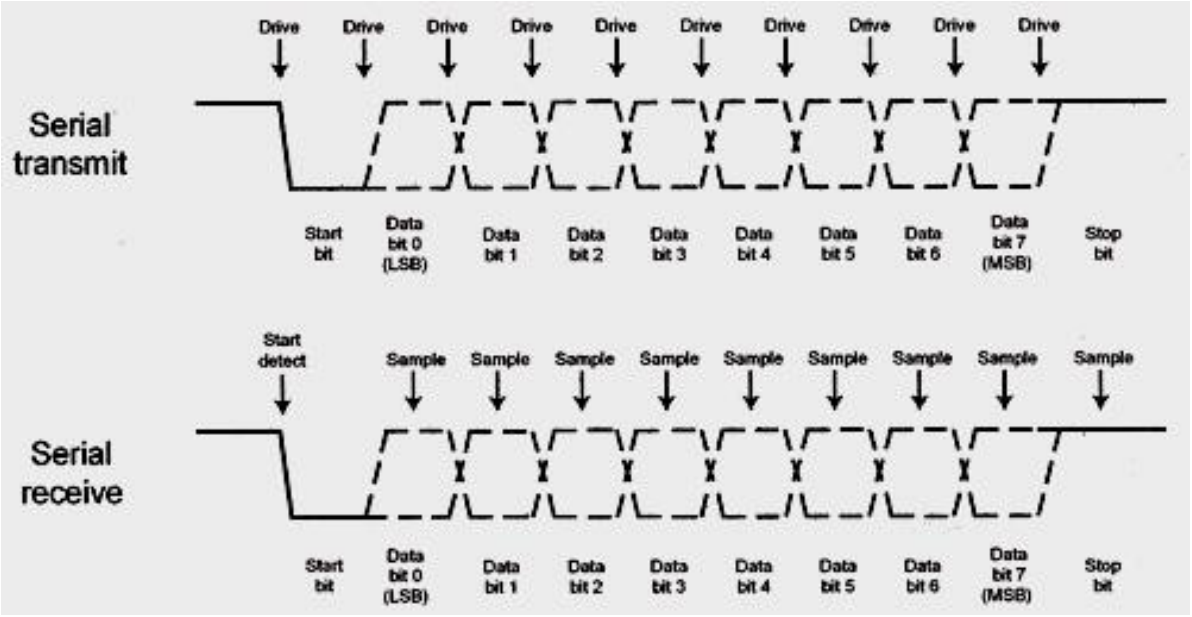
ลำดับของสัญญาณในการส่งข้อมูลแบบ UART (Tx)

- ในสภาวะเริ่มต้นสัญญาณจากตัวส่ง Tx จะมีลอจิกเป็น ‘1’
- เมื่อเริ่มทำการส่งข้อมูลตัวส่งจะส่ง Start Bit ที่มีลอจิกเป็น ‘0’ จำนวน 1บิต
- จากนั้นจะส่งข้อมูลขนาด 8 บิต โดยส่ง LSB ไปก่อน จนถึง MSB
- แล้วตามด้วย Parity Bit (Odd หรือEven Parity) จำนวน 1 บิต ในส่วนนี้จะมีหรือไม่มีก็ได้
- จากนั้นจะส่ง Stop Bit ที่มีลอจิกเป็น ‘1’ จำนวน 1 บิต

เพื่อเป็นการบอกว่าสิ้นสุดการสงข้อมูลในหนึ่งเฟรม

ลำดับของสัญญาณในการส่งข้อมูลแบบ UART (Rx)

- ในสภาวะเริ่มต้นสัญญาณในบัสข้อมูลจะมีลอจิกเป็น ‘1’ ซึ่งหมาความว่ายังไม่มีการส่งข้อมูล
- เมื่อเริ่มทำการส่งข้อมูลสัญญาณในบัสข้อมูลจะมีการเปลี่ยนแปลงจาก ลอจิก ‘1’ เป็น ‘0’ โดยมีความกว้างของสัญญาณ ขนาด 1 บิตข้อมูล
- จากนั้นจะส่งข้อมูลขนาด 8 บิต โดยส่ง LSB ไปก่อน จนถึง MSB
- จากนั้นจะส่ง Stop Bit ที่มีลอจิกเป็น ‘1’ มีความกว้างของสัญญาณ ขนาด 1 บิตข้อมูล เพื่อเป็นการบอกว่าสิ้นสุดการสงข้อมูลในหนึ่งเฟรม



รูปภาพ 2.11 รูปแบบ Uart Data Frame Rx และ Tx

http://www.embedded.com/design/other/4025995/Implementing-your-MCU-based-system-s-serial-UART-in-software

SPI (Serial Peripheral Interface)

รูปภาพ 2.12 รูปแบบการทำงานของ SPI
https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

SPI หรือ Serial Peripheral Interface เป็นวิธีการสื่อสารรูปแบบหนึ่ง ที่ใช้ในการติดต่อสื่อสารกับอุปกรณ์อิเล็กทรอนิกส์ โดยมีการรับส่งข้อมูลแบบ อนุกรมซิงโครนัส (synchronous seral data link) ในการรับส่งข้อมูลเป็นการสื่อสารแบบสองทิศทาง (full duplex)

การติดต่อส่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์ที่เชื่อมต่อแบบ SPI จะมี การทำงานในรูปแบบที่ให้อุปกรณ์ตัวหนึ่งทำหน้าที่เป็น MASTER ในขณะที่อีกตัวหนึ่งทำหน้าที่เป็น SLAVE และส่งข้อมูลในโหมด Full-duplex นั่นหมายความว่า สัญญาณสามารถส่งหากันได้ระหว่าง MASTER และ SLAVE ได้อย่างต่อเนื่อง ในการสื่อสารแบบ SPI นี้ ไม่ได้มีมาตรฐานก าหนดตายตัวว่าข้อมูลที่ส่งหากันต้องอยู่ในรูปแบบหรือ format แบบไหน เป็นการคิด protocol การสื่อสาร

รูปแบบของสายสัญญาณที่ใช้ในการติดต่อสื่อสารแบบ SPI ในการสื่อสาร แบบ SPI โดยทั่วไปจะใช้สายสัญญาณในการสื่อสาร 4 เส้น (four wire) ได้แก่

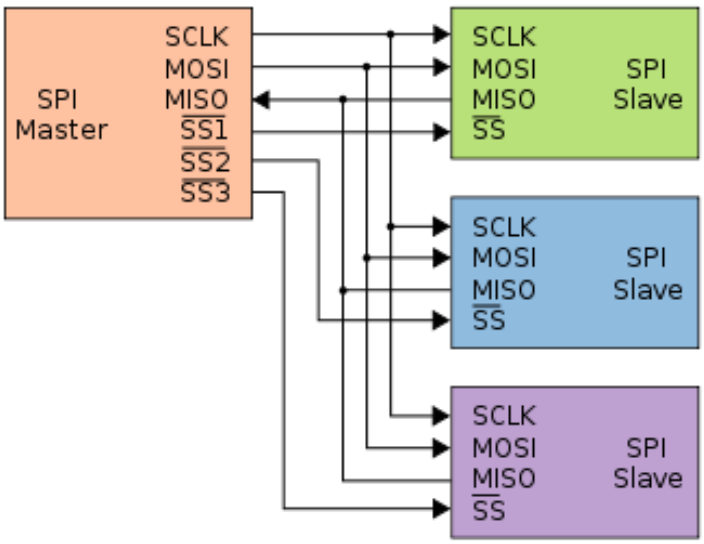
- SCLK : Serial Clock เป็นสัญญาณที่ใช้ในการกำหนดจังหวะการรับส่งข้อมูลซึ่งในการควบคุมจังหวะการติดต่อสื่อสาร อุปกรณ์ที่เป็น Master จะเป็นผู้กำหนดจังหวะโดยความถี่ของสัญญาณ Clock ขึ้นอยู่กับผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ที่เชื่อมต่อ โดยทั่วไปตวามถี่ของสัญญาณ Clock สามารถมีความถี่ในช่วง เมกะเฮิร์ตซ์ (MHz)

- MOSI : Master Output Slave Input เป็นสัญญาณข้อมูลที่ฝั่งอุปกรณ์ที่เป็น Master ส่งไปยังอุปกรณ์ที่เป็น Slave โดยข้อมูลดังกล่าวจะเป็นไปตามรูปแบบ protocol การสื่อสารข้อแต่ละอุปกรณ์ ซึ่งในบางผลิตภัณฑ์จะเรียกสัญญาณนี้ว่า Din, Di เป็นต้น

• MISO : Mater Input Slave Output เป็นสัญญาณข้อมูลที่ฝั่งอุปกรณ์ที่เป็น Slave ส่งไปยังอุปกรณ์ที่เป็น Master โดยข้อมูลดังกล่าวจะเป็นไปตามรูปแบบ protocol การสื่อสารของแต่ละอุปกรณ์ ซึ่งในบางผลิตภัณฑ์จะเรียกสัญญาณนี้ว่า Dout, Do เป็นต้น

• SS: Slave Select (active-low) เป็นสัญญาณที่เลือกอุปกรณ์ที่จะติดต่อสื่อสารด้วยโดยการควบคุมจังหวะกาติดต่อสื่อสารฝั่ง Master จะเป็นตัวกำหนดจังหวะเพื่อเลือกการติดต่อสื่อสาร ซึ่งในบางผลิตภัณฑ์จะเรียกสัญญาณนี้ว่า CS เป็นต้น

รูปแบบการเชื่อมต่อระหว่างอุปกรณ์ Master 1 ตัว กับ Slave หลายตัวโดยใช้สัญญาณ Slave Select ในการกำหนดอุปกรณ์ที่ต้องการชื่อมต่อ



รูปภาพ 2.13 รูปแบบการเชื่อมต่อแบบ 1 Master กับ Slave หลายตัว
https://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

จังหวะการเชื่อมต่อและส่งข้อมูลของระบบ SPI บัส

1. อุปกรณ์ที่เป็น Master จะส่งสัญญาณ SS ที่มีลอจิกเป็น Low เพื่อเป็นการเลือกว่าจะติดต่อกับอุปกรณ์ Slave
2. หลังจากนั้นข้อมูลของการสื่อสารระหว่าง Master และ Slave จะถูกส่งเพื่อสื่อสารกันโดยในขบวนการดังกล่าวจะถูกกำหนดจากจังหวะของสัญญาณ Clock ที่ Master เป็นตัวกำหนด
3. โดยจังหวะที่ Master ท าการส่งข้อมูลไปยัง Slave ผ่าน MOSI ทีละหนึ่งบิตและ Slaveจะรับข้อมูลดังกล่าว และในช่วงเวลาขณะเดียวกัน Slave ก็จะทำการส่งข้อมูลที่ละหนึ่งบิตให้กับ Master โดยในขบวนการดังกล่าวจะมีสัญญาณ Clock เป็นตัวควบคุมจังหวะการทำงาน โดยรูปแบบการรับและส่งข้อมูลดังกล่าวจะมีข้อกำหนดตามรูปแบบของ Clock polarity and phase
4. เมื่อสิ้นสุดการส่งข้อมูล Master จะส่งสัญญาณ SS ที่มีลอจิกเป็น High เพื่อยกเลิกการติดต่อกับ Slave

ลำดับการติดต่อสื่อสารในระบบ SPI Bus

1. ก่อนการเริ่มต้นการติดต่อ สัณณาณ SS มีลอจิกเป็น High, SCK มีลอจิกเป็น Low MISO และ MOSI ยังไม่มีการเริ่มส่งข้อมูล
2. ในการเริ่มต้นการติดต่อ Master จะเปลี่ยนสัญญาณ SS ที่มีลอจิกเป็น High ให้เป็น Low เพื่อบอกกับอุปกรณ์ Slave ว่าจะเริ่มขบวนการติดต่อ
3. จากนั้นในการส่งข้อมูลจาก Master ไปยัง Slave และ Slave ไปยัง Master ได้เริ่มขึ้นโดยรูปแบบจะเป็นไปตาม Clock polarity and phase
4. ที่ Master และ Slave ได้กำหนดแบบเดียวกัน ซึ่งในตัวอย่างนี้ได้ก าหนด CPOL=0 และ CPHA=1
5. โดยลักษณะการส่งข้อมูลดังกล่าวสามารถอธิบายได้ดังนี้ ในการเริ่มต้นส่งข้อมูลจาก Master ไปยัง Slave จะเริ่มจาก SS เปลี่ยนจาก High เป็น Low
   โดยข้อมูลจะถูกส่งไปที่ละหนึ่งบิตทุกขอบขาลงของ Clock ส่วนการส่งข้อมูลจาก Slave ไป Master จะมีการส่งข้อมูลทีละบิตในทุกขอบขาขึ้นของ Clock