PART 8: Embedded Programming STM32

ADC, Standard communication (I2C/SPI/UART)

ISEN | école d'ingénieurs
ALL IS DIGITAL!

Analog To Digital Converter

What is an ADC?

- ✓ ADC stands for Analog to Digital Converter
- ✓ Converts an analog voltage on a pin to a digital number
- ✓ Converting from the analog world to the digital world, electronics can be used to interface to the analog world (the real world)

n-bit ADC?

- ✓ The converted voltage is stored in a n-bit register. The size of this register defines the resolution of the ADC
- ✓ STM32 integrates a 12-bit ADC. 12-bit ADC means it can detect up to 4,096 (= 2^12) discrete analog levels.
- ✓ An 8-bit ADC has 2^8 = 256 discrete levels while a 16-bit ADCs has 2^16 = 65,536 discrete levels

ADC Categories

ADCs has many architectures. The three most popular categories are

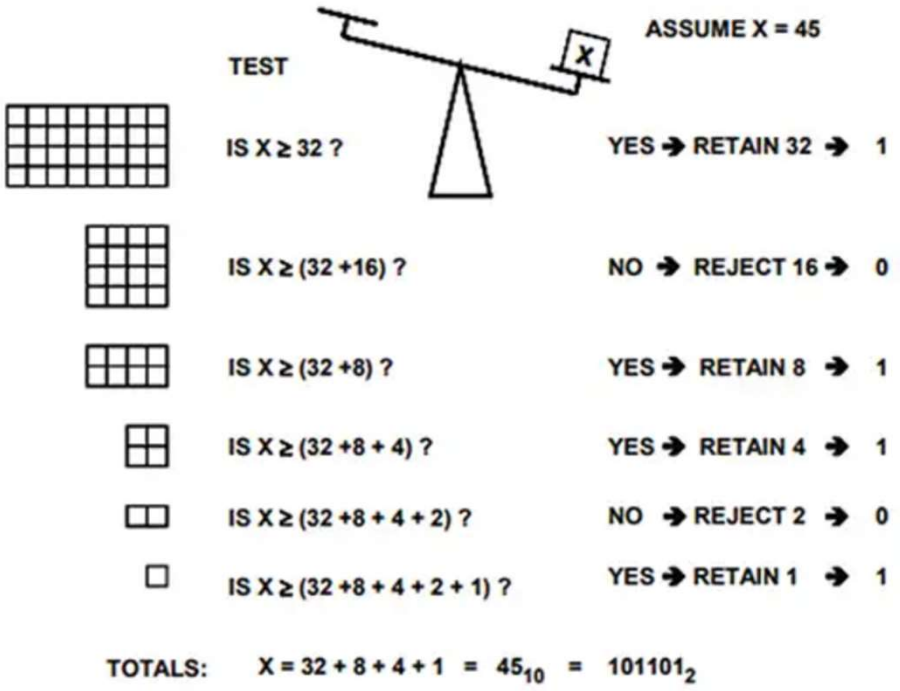
- ✓ Successive Approximation Registers (SAR) (architecture used in microcontrollers)
- ✓ Delta-Sigma (Δ∑)
- ✓ Pipeline converters

ADC Categories Comparison

ADC Category	Advantages	Disadvantages
SAR	✓ Low power ✓ Low latency ✓ Good accuracy	✓ Medium resolution
Delta Sigma	✓ High resolution ✓ High Stability	✓ Low Speed
Pipeline	✓ High speed	✓ Low resolution ✓ High power

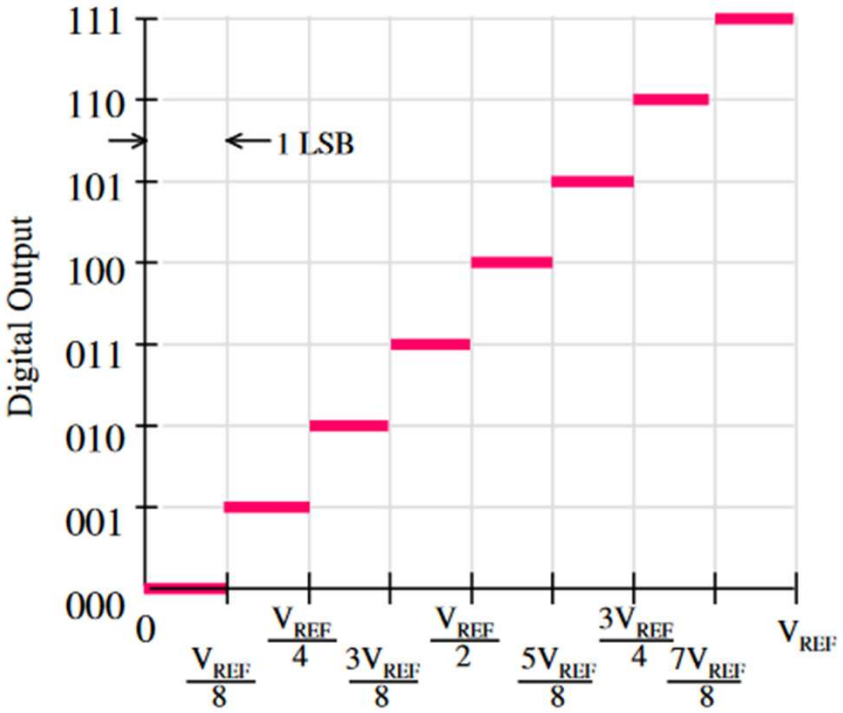
A Microcontroller ADC cannot meet all the requirements and there is a big market for stand alone ADCs. Leaders are TI, Analog Device and Maxim Integrated

SAR ADC Analogy



The block = 45 units, the first test is to use 32 small blocks each being a unit of 1. The block is still heavier than the smaller blocks. Since the block is still heavier the smaller blocks will remain. The next test is to add 16 more small blocks. This would be 32 + 16 which equals 48, hence being too heavy. As a result of being too heavy, the 16 blocks are discarded. The next test is to add 8 more blocks to the scale, since 32 + 8 will equal 40 that quantity will be added. And so on…

Quantization Error



Error

The Magnitude of the Error Ranges from Zero to 1 LSB

SAR ADC Analogy

- ✓ To process the analog signal SAR ADCs will have a sample and hold to keep the signal constant.
- ✓ A comparator that measures the analog input against an internal voltage reference. This voltage reference will be set to ½ of its potential voltage at this point.
- ✓ If the input is higher than the DAC, the comparator will output a 1 to be stored in the MSB in the “successive approximation register”.
- ✓ After this the reference voltage will be set to ¼ of its potential voltage and the process repeats. The next value for the DAC would be 1/8 then 1/16 and so on until all of the bits had been loaded into the register.

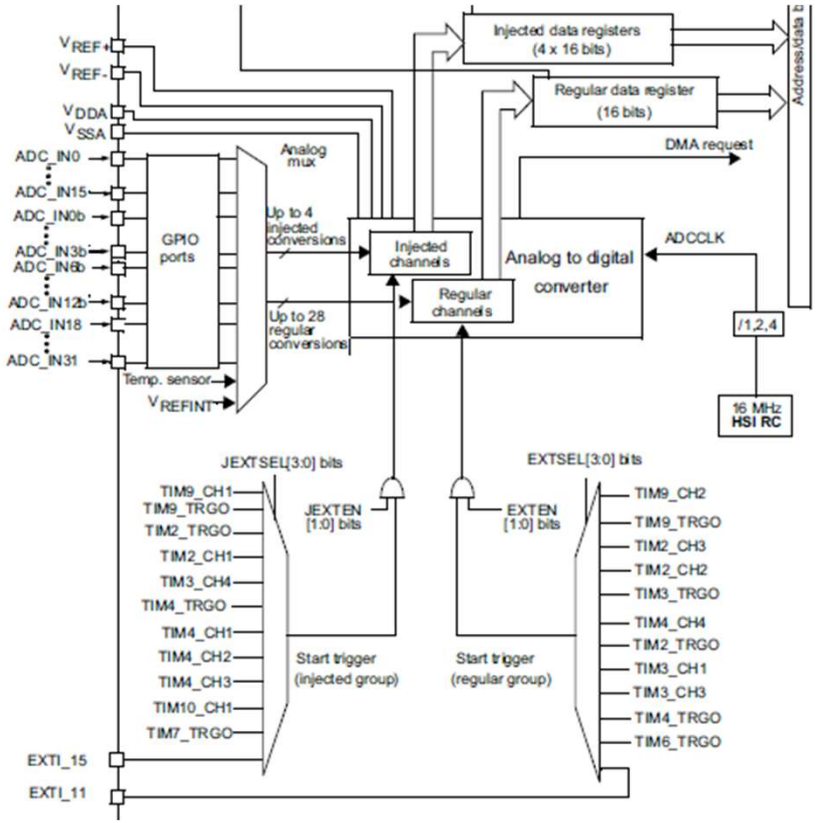
ISEN ALL IS DIGITAL! OUEST | yncréa

STM32L152 ADC Characteristics

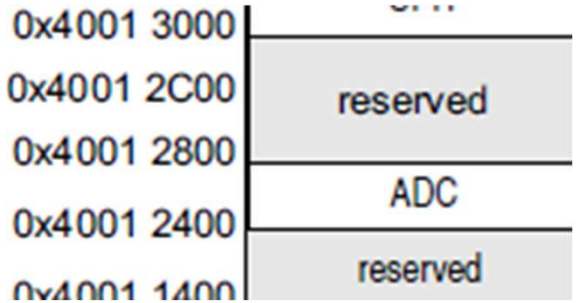
The STM32L152 has one Analog to Digital Converter with the following main features

- ✓ 12-bit resolution
- ✓ Single and continuous conversion modes. Scan Mode in a programmable order
- ✓ Buffered continuous conversion mode
- ✓ External trigger option with configurable edge detection
- ✓ Interrupt generation at End of Conversion
- ✓ Conversion time: 1µs at full speed (ADC clocked at 16MHz) down to 4µs at low speed (ADC clocked at 4MHz), independent of the APB clock

STM32 ADC as Peripheral



- ✓ ADC is a peripheral
- ✓ The base address for the registers is 0x4001 2400



ISEN ALL IS DIGITAL! OUEST | yncréa

ADC Inputs

Pins					Pin name	Pin Type(1)	I / O structure	Main function(2) (after reset)	Pin functions	
LQFP144	UFBGA132	LQFP100	LQFP64	WLCSP104					Alternate functions	Additional functions
41	K4	30	21	M8	PA5	I/O	TC	PA5	TIM2_CH1_ETR/ SPI1_SCK	ADC_IN5/ DAC_OUT2/ COMP1_INP
42	L4	31	22	H6	PA6	I/O	FT	PA6	TIM3_CH1/TIM10_CH1/SPI1_MISO/ LCD_SEG3	ADC_IN6/ COMP1_INP/ OPAMP2_VINP
43	-	32	23	K7	PA7	I/O	FT	PA7	TIM3_CH2/TIM11_CH1/ SPI1_MOSI/ LCD_SEG4	ADC_IN7/ COMP1_INP/ OPAMP2_VINM
-	J5	-	-	-	PA7	I/O	FT	PA7	TIM3_CH2/TIM11_CH1/ SPI1_MOSI/ LCD_SEG4	ADC_IN7/ COMP1_INP

- ✓ AINx stands for Analog Input
- ✓ The Datasheet shows the available pins for the ADC in the additional functions column

Selection Channel

- ✓ The number of input channels depends on devices
- ✓ The channel to convert is programmed in ADC_SQR registers

12.15.14 ADC regular sequence register 5 (ADC_SQR5)

Address offset: 0x40

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_0	SQ3[4:0]					SQ2[4:0]					SQ1[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Conversion Data Storage

12.15.17 ADC regular data register (ADC_DR)

Address offset: 0x58

Reset value: 0x0000 0000

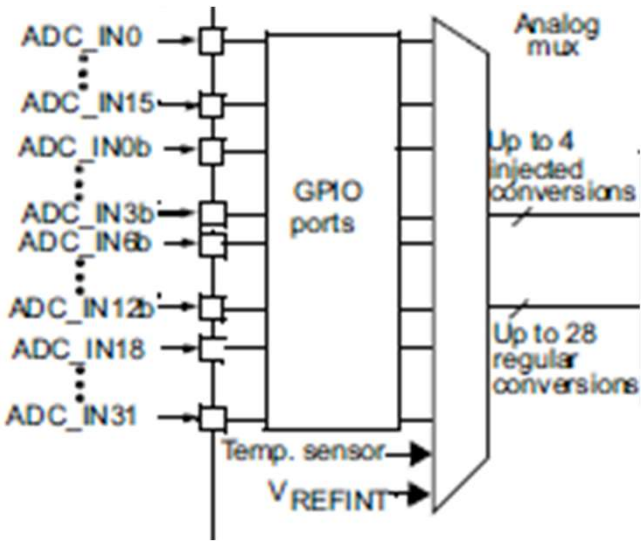
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bits 31:16 Reserved.

Bits 15:0 **DATA[15:0]:** Regular data

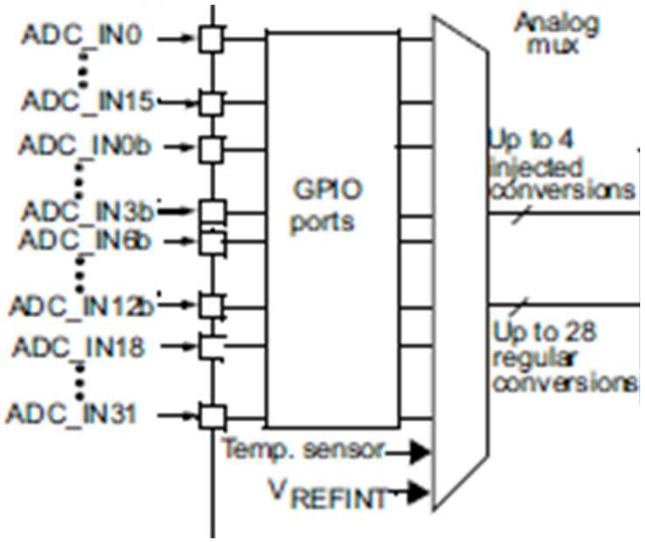
These bits are read-only. They contain the conversion result from the regular channels. The data are left- or right-aligned as shown in *Figure 44* and *Figure 45*.

ADC: Single Conversion Mode



- ✓ The ADC does **one conversion** on the selected channel in the S1 bits of the ADC_SQR5 register
- ✓ This mode is started by setting the ADON and the CONT bit is 0 in the ADC_CR2 register
- ✓ Once the conversion is complete, the converted data are stored in the ADC_DR register, the EOC (End of Conversion) flag is set and an interrupt is optionally generated if the EOCIE bit is set

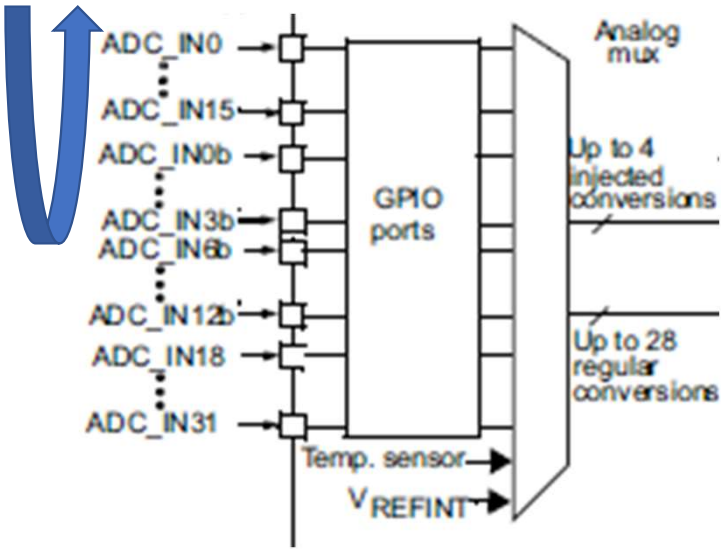
ADC: Continuous Conversion Mode



- ✓ The ADC starts another conversion as soon as it finishes one.
- ✓ This mode is started by setting the ADON bit in the ADC_CR2 register while the CONT bit is set

ADC: ScanMode

All the inputs are converted sequentially



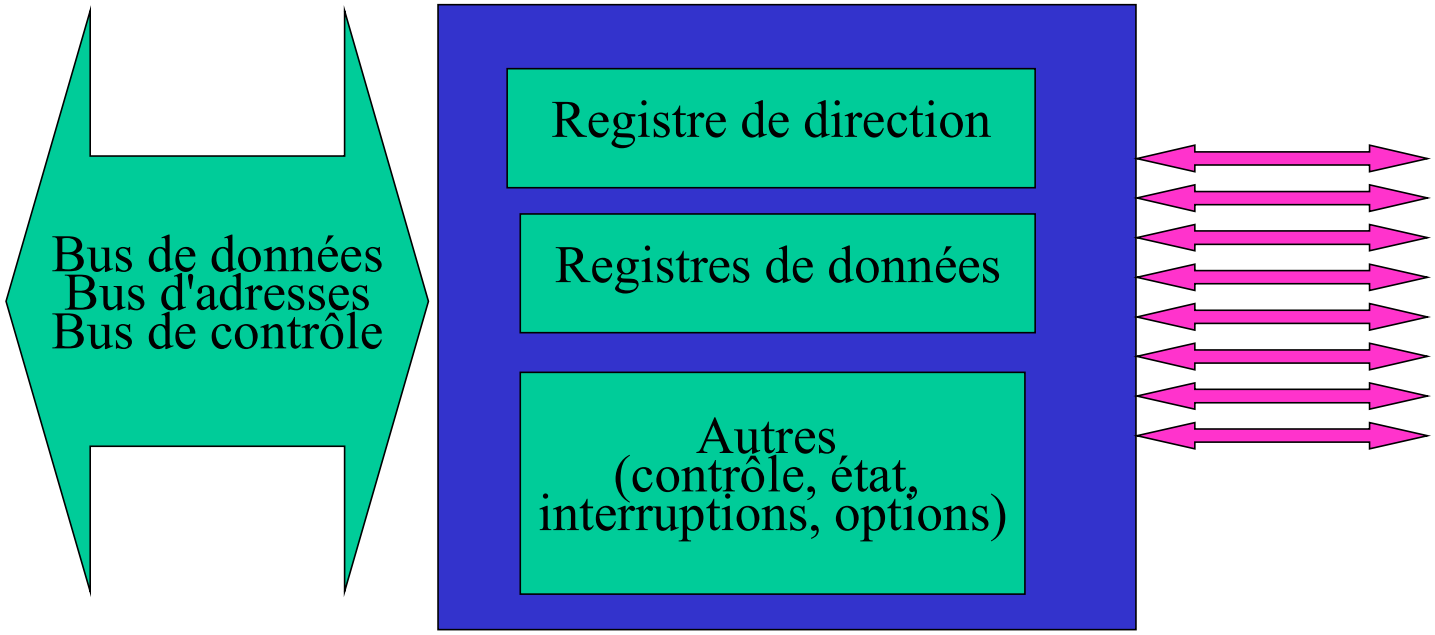
- ✓ In scan mode is used to convert a sequence of analog channels from AINO to AINn
- ✓ This sequence is programmable through ADC_SQRx registers
- ✓ The number of channels to convert is programmed in the L[4:0] bits of the ADC_SQR1 register
- ✓ Single scan mode is started by setting the SCAN bit (in the ADC_CR1 register)

ISEN | école d'ingénieurs
ALL IS DIGITAL!

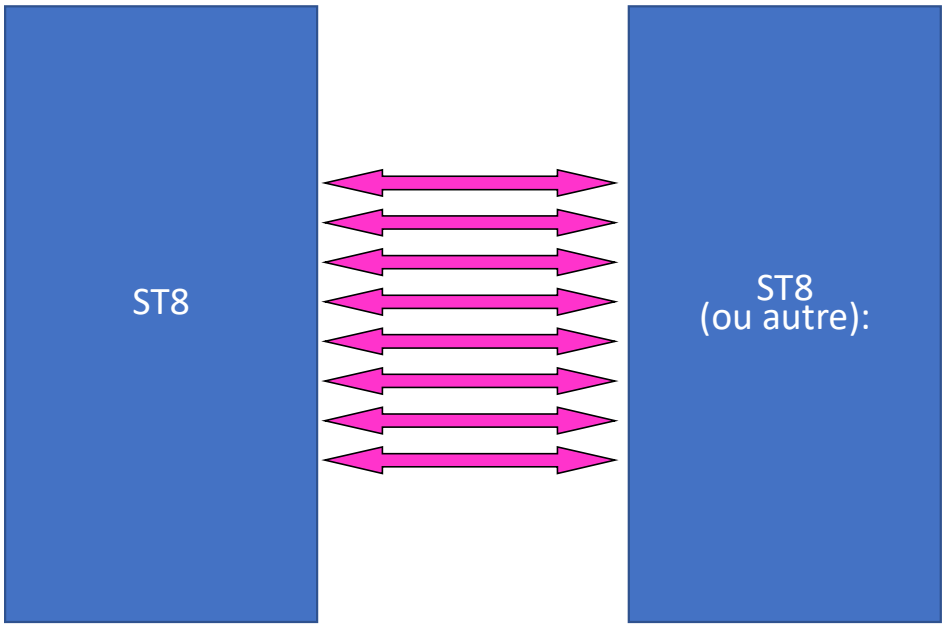
Standard Communication

Serial Communication

Parallel Interface Usage



Parallel Interface Advantages /Disadvantages



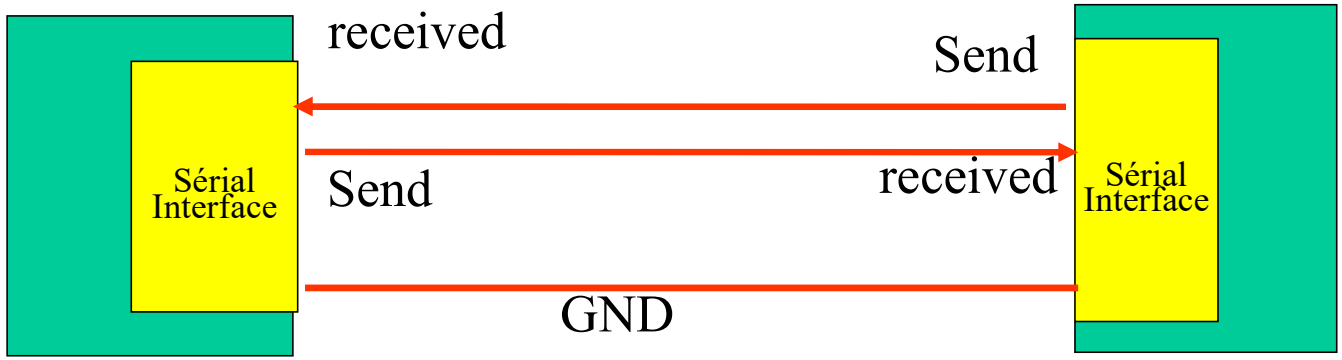
Advantages:

- ✓ Several bits at the same time (speed)
- ✓ Easy implementation

Disadvantages:

- ✓ Cost
- ✓ Size implementation on board and large packages
- ✓ Very limited in distance

Serial Interface



- ✓ One or two wires to transmit bits
- ✓ Bits are sent one after one
- ✓ One clock wire if the transmission is synchronous

Common Serial Interfaces

- ✓ In microcontrollers
  - SPI
  - I2C
  - UART
- ✓ In automotive application
  - CAN
  - LIN
- ✓ Industrial Applications
  - RS232 (ability to send to tens of meters)
  - RS422/RS485 (ability to send to hundreds of meters)

Communication

Simplex Communication

Communication channel sends information in one direction only

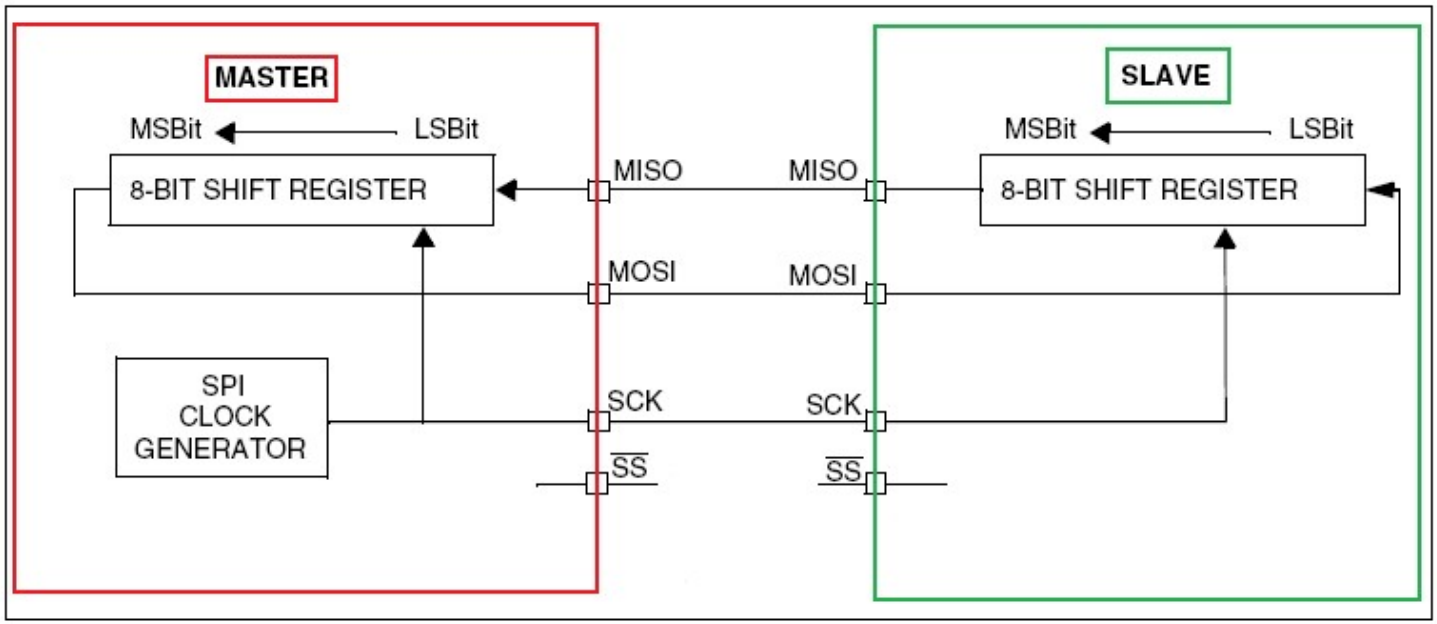
Half Duplex Communication (transmit or receive)

Communication in both directions, but only one at a time

Full Duplex Communication (transmit and receive)

Communication in both directions at the same time

Principal Serial Communication



In serial communication, registers of master and slave are shifted synchronously

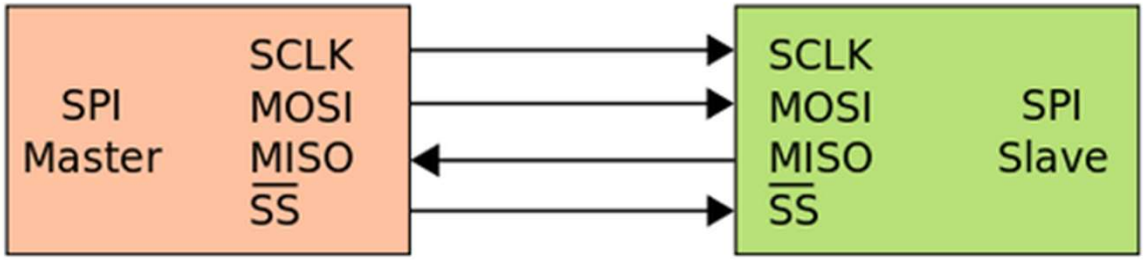
ISEN | école d'ingénieurs
ALL IS DIGITAL!

Standard Communication

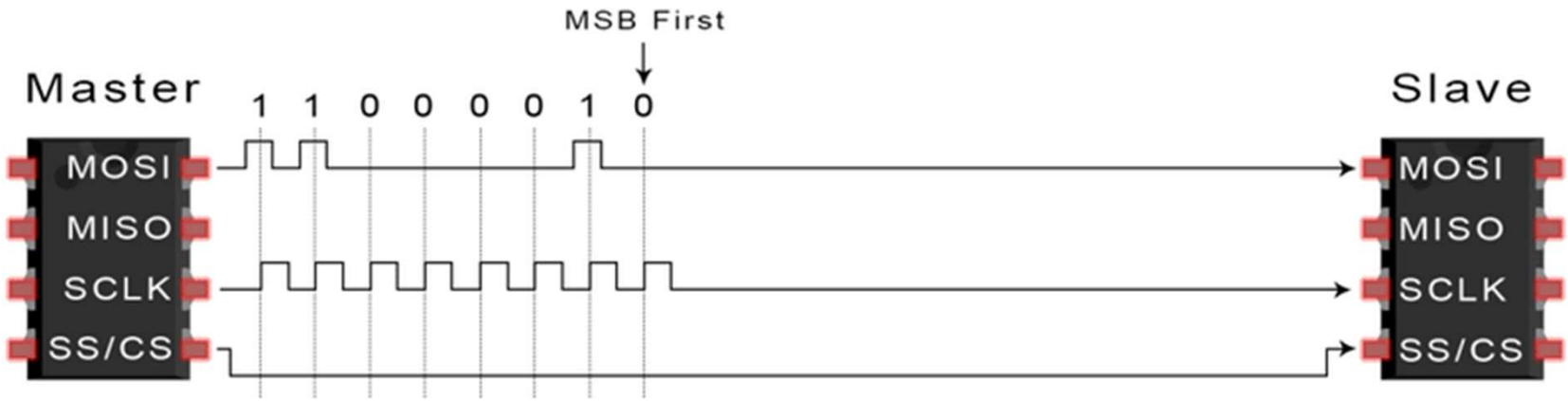
SPI

SPI

- ✓ SPI stands for Serial Peripheral Interface
- ✓ Full duplex synchronized bus designed by Motorola in mid 80's
- ✓ Circuits communicate according to a master-slave scheme
- ✓ The bus is composed of 4 wires
  - SCLK (or SCK, SCL): Serial Clock (controlled by the master)
  - MOSI (or SDO): Master Output, Slave Input (controlled by the master)
  - MISO (or SDI): Master Input, Salve Output (controlled by the slave)
  - SS: Slave Select

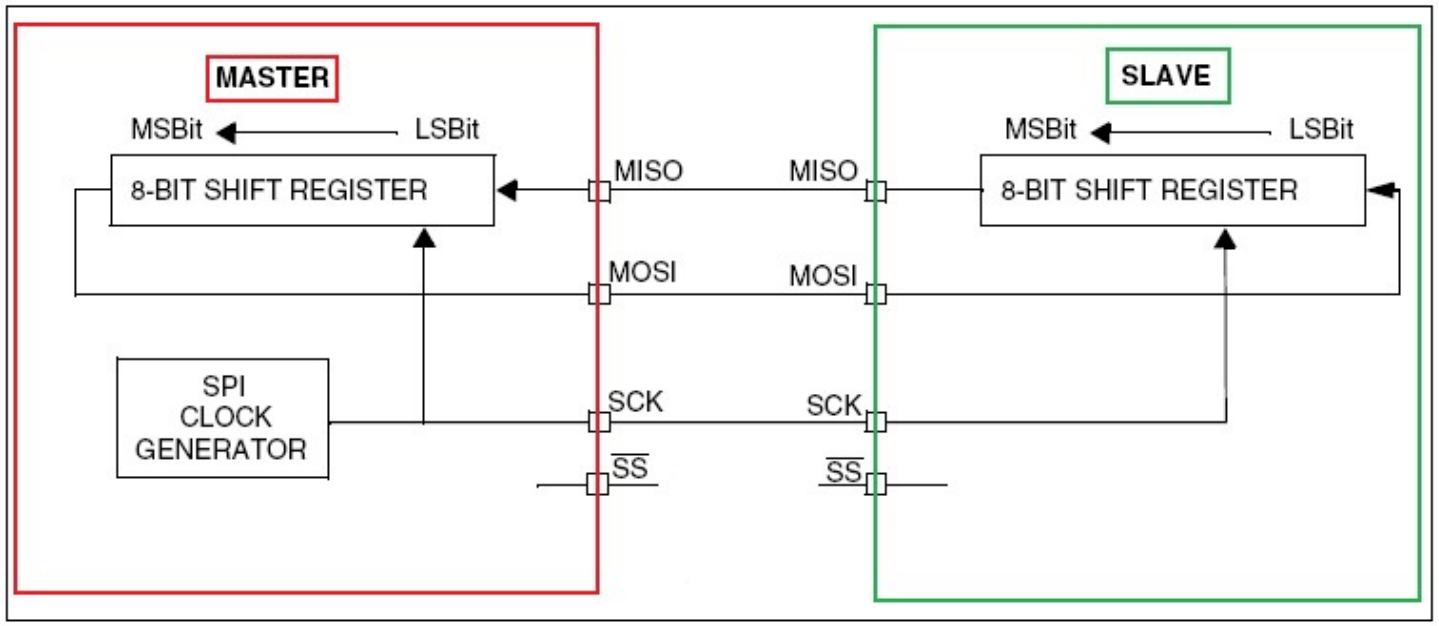


SPI Communication



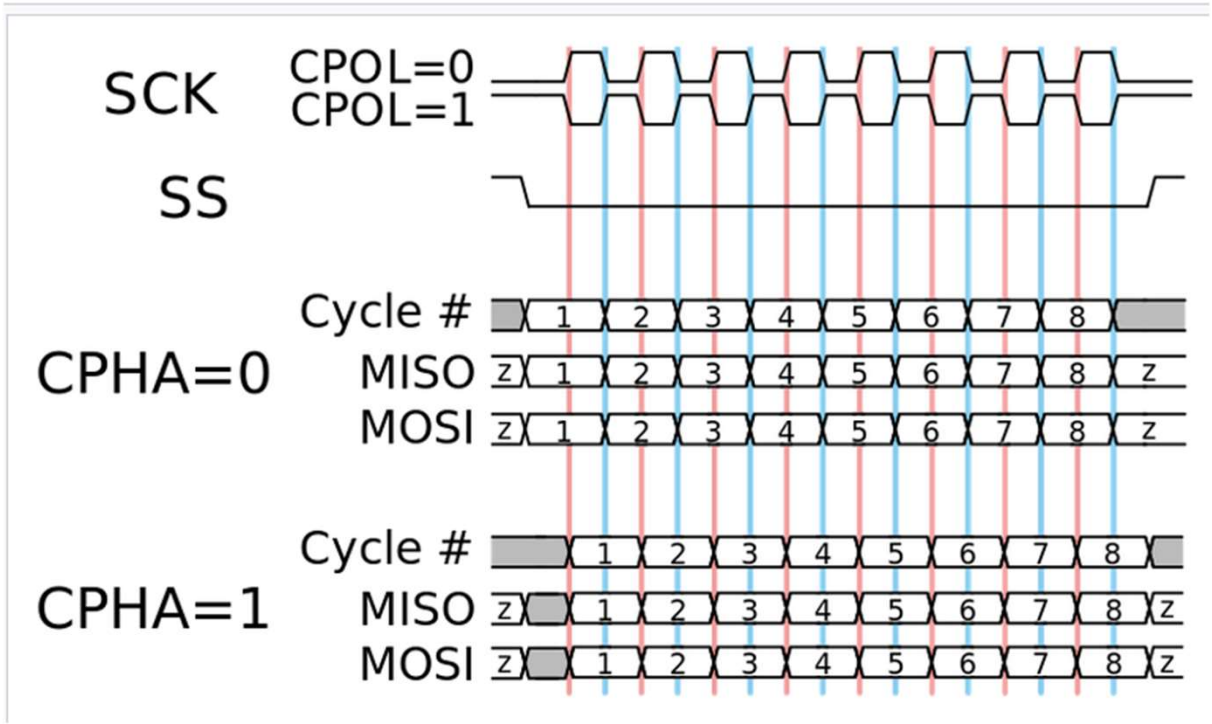
At each clock, a bit is transmitted. The bus is synchronous.

SPI Communication



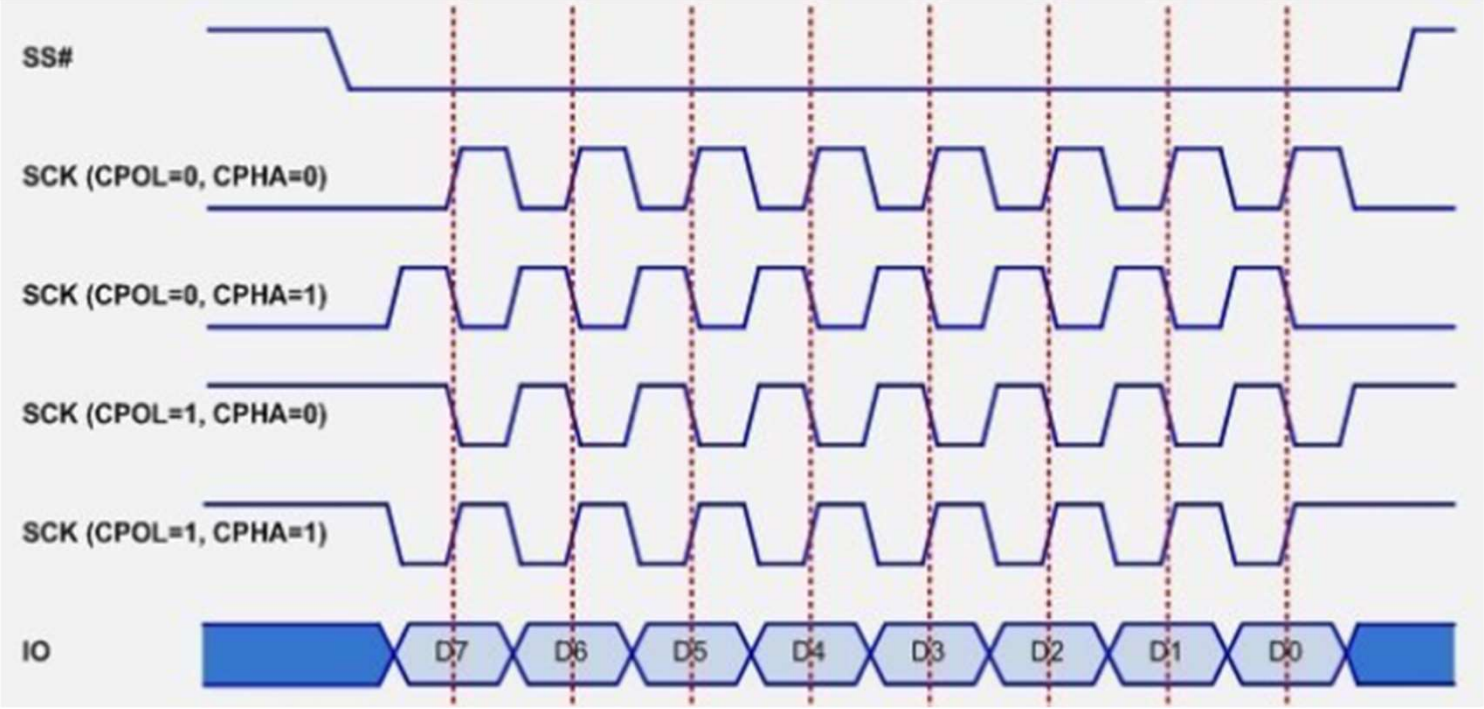
In serial communication

Polarity



- ✓ CPOL is the Clock Polarity: assigns the clock level when the clock is not active
- ✓ CPHA: used to shift the capturing phase
- ✓ Four possible modes to correspond to slave device

Polarity


SS#
SCK (CPOL=0, CPHA=0)
SCK (CPOL=0, CPHA=1)
SCK (CPOL=1, CPHA=0)
SCK (CPOL=1, CPHA=1)
IO
D7
D6
D5
D4
D3
D2
D1
D0

MAX7219 SPI Specifications

PIN	NAME	FUNCTION
1	DIN	Serial-Data Input. Data is loaded into the internal 16-bit shift register on CLK's rising edge.

Serial-Addressing Modes

For the MAX7219, serial data at DIN, sent in 16-bit packets, is shifted into the internal 16-bit shift register with each rising edge of CLK regardless of the state of LOAD. For the MAX7221, $\overline{CS}$ must be low to clock data in or out. The data is then latched into either the digit or control registers on the rising edge of LOAD/$\overline{CS}$. LOAD/$\overline{CS}$ must go high concurrently with or after the 16th rising clock edge, but before the next rising clock edge or data will be lost. Data at DIN is propagated through the shift register and appears at DOUT 16.5 clock cycles later. Data is clocked out on the falling edge of CLK. Data bits are labeled D0–D15 (Table 1). D8–D11 contain the register address. D0–D7 contain the data, and D12–D15 are "don't care" bits. The first received is D15, the most significant bit (MSB).

- ✓ 16-bit
- ✓ Data received on rising edge
- ✓ Data latched on rising edge of LOAD
- ✓ Data received with MSB first

SPI Registers

Address offset	Register name	7	6	5	4	3	2	1	0
0x00	SPI_CR1 Reset value	LSB FIRST 0	SPE 0	BR2 0	BR1 0	BR0 0	MSTR 0	CPOL 0	CPHA 0
0x01	SPI_CR2 Reset value	BDM 0	BDOE 0	CRCEN 0	CRCNEXT 0	- 0	RXONLY 0	SSM 0	SSI 0
0x03	SPI_SR Reset value	BSY 0	OVR 0	MODF 0	CRCERR 0	WKUP 0	- 0	TXE 1	RXNE 0
0x04	SPI_DR Reset value	DR[7:0] 0							

- ✓ Two configuration registers (Polarity, speed, clock master, first bit, enable peripheral and CRC, hardware….)
- ✓ One Status register (on going communication, error in CRC…)
- ✓ One interrupt register
- ✓ One Data register

MAX7219 SPI Specifications

- ✓ 10MHz speed
- ✓ 16-bit
- ✓ Data received on rising edge
- ✓ Data latched on rising edge of LOAD
- ✓ Data received with MSB first

Le registre SPI_CR1 est un registre de 8 bits défini comme suit :

7	6	5	4	3	2	1	0
LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
rw	rw	rw			rw	rw	rw

- LSBFIRST = 0 : transmet le MSB en premier
- SPE = 0 : désactive le SPI le temps de la configuration.
- BR[2 :0] = 101 : indique la vitesse de communication (baudrate) de la liaison : ici fréquence égale à fMASTER/64 soit 31250 Hz
- MSTR = 1 : indique que le ucontrôleur est maître
- CPOL-CPHA = 11 : indique la polarité et la phase de l'horloge

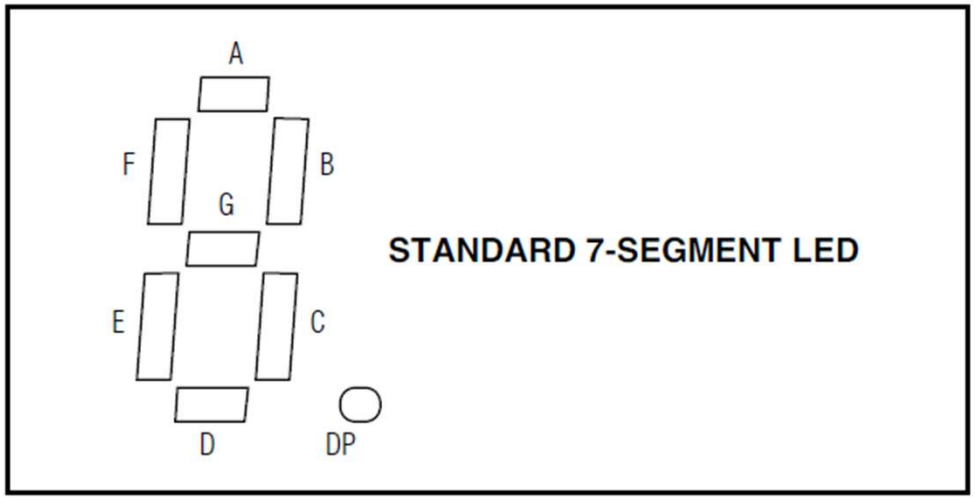
Le registre SPI_CR2 est un registre de 8 bits défini comme suit :

7	6	5	4	3	2	1	0
BDM	BDOE	CRCEN	CRCNEXT	Reserved	RXOnly	SSM	SSI
rw	rw	rw	rw	rw	rw	rw	rw

- BDM = 0 : communication unidirectionnelle sur 2 fils
- BDOE = 0 ou 1 : pas d'influence sur une communication unidirectionnelle
- CRCEN = 0 : pas de calcul de CRC (moyen de valider les données)
- CRCNEXT = 0 : désactive l'envoi du CRC au message suivant
- RXOnly = 0 : communication full duplex
- SSM-SSI = 11 : "software slave management enable" en mode master

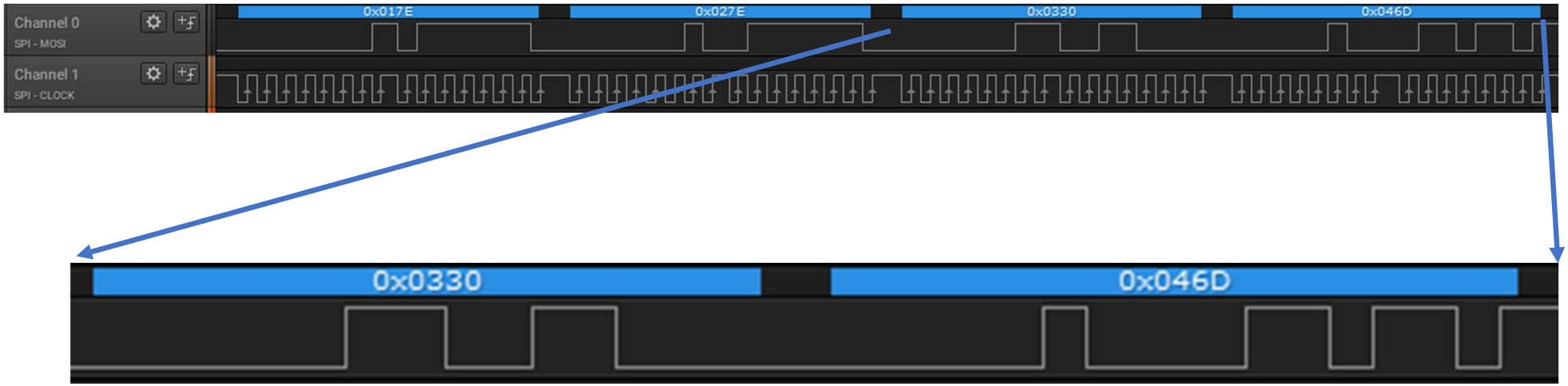
Matrix Digit / 7-segment

Digit Value	Hex Value to decode
0	0x7E
1	0x30
2	0x6D
3	0x79
4	0x33
5	0x5B
6	0x5F
7	0x70
8	0x7F
9	0x7B



	REGISTER DATA							
	D7	D6	D5	D4	D3	D2	D1	D0
Corresponding Segment Line	DP	A	B	C	D	E	F	G

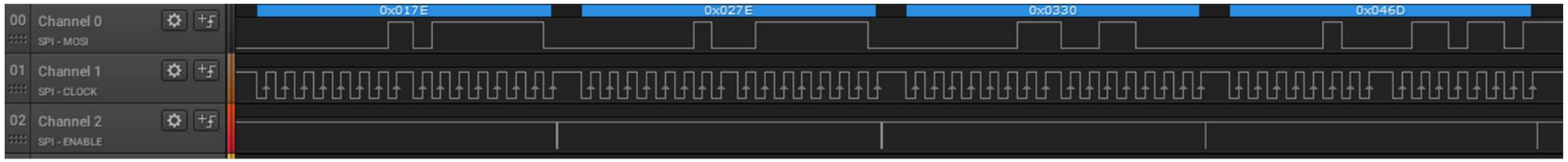
Frame Example

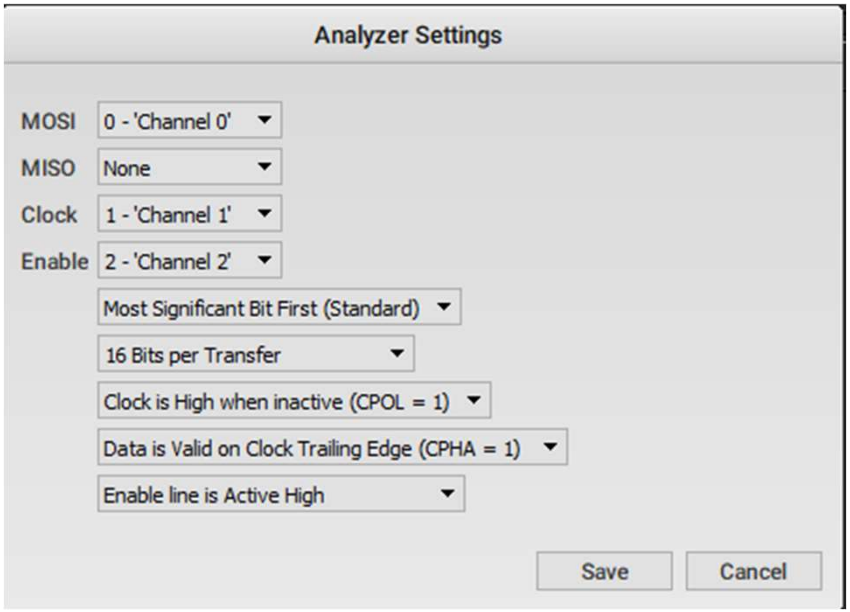


This frame sends the information

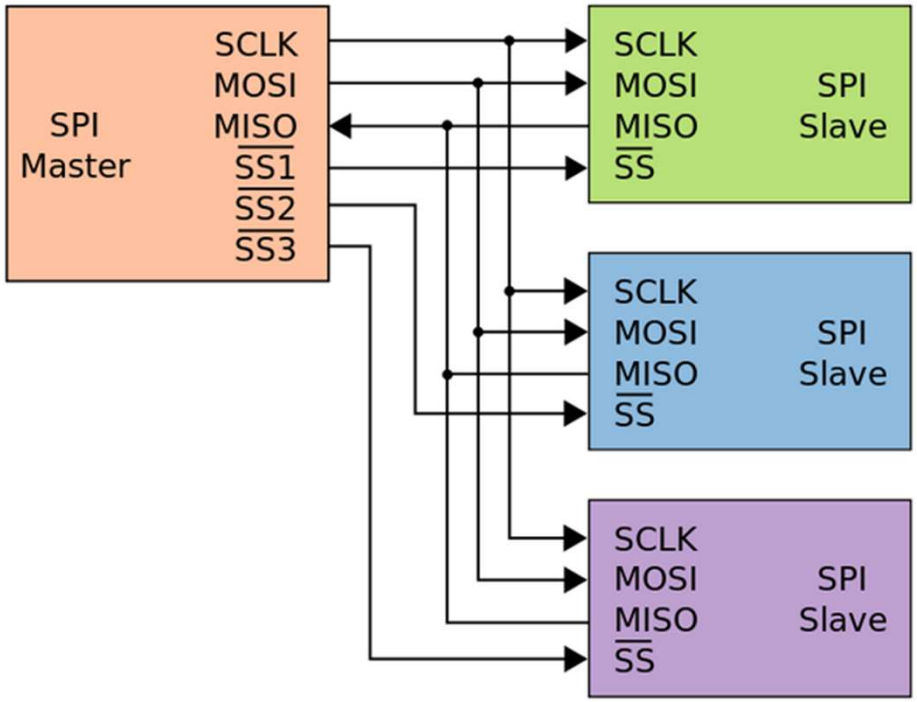
- ✓ 0 for 1st digit
- ✓ 0 for 2nd digit
- ✓ 1 for 3rd digit
- ✓ 2 for 4th digit

Analyzer Setting





Multiple Slaves



The Slave Select pin selects which slave communicates with the master

Advantages / Disavantages SPI

ADVANTAGES	DISADVANTAGES
No start and stop bits, so data can be streamed continuously without interruption	Uses four wires (Vs I2C and UARTs that need two wires)
No complicated slave addressing system	No acknowledgement that the data has been successfully received
High data transfer rate (several MHz)	No form of error checking (need a CRC)
Full Duplex	Allows a single master

ISEN | école d'ingénieurs
ALL IS DIGITAL!

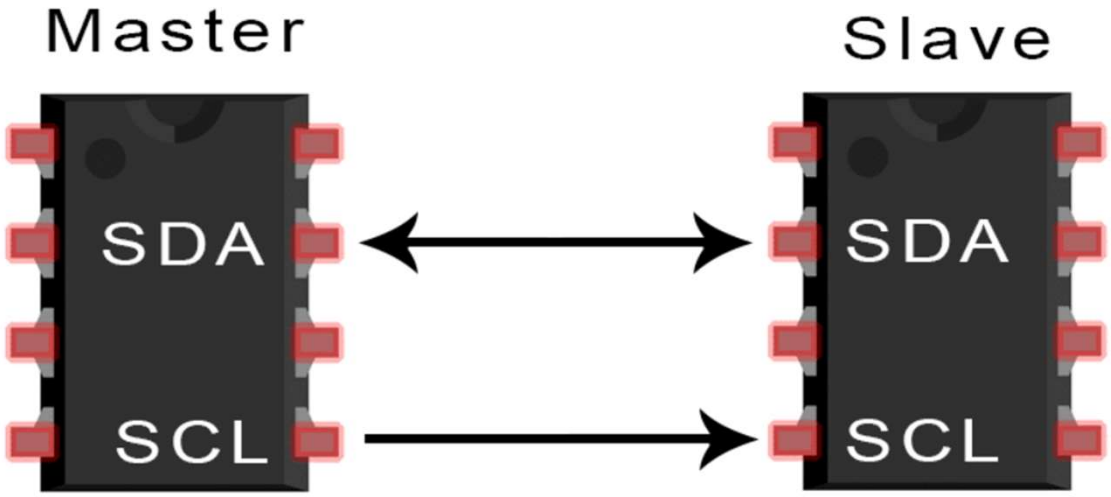
Standard Communication

I2C

I2C

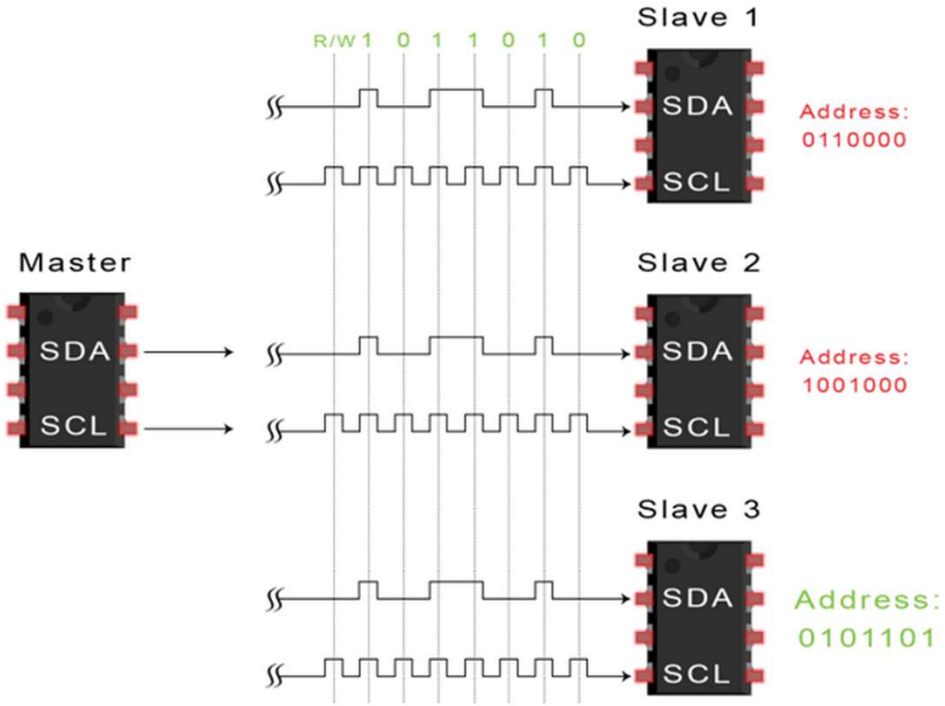
- ✓ I2C stands for Inter-Integrated Circuit
- ✓ Designed by Philips
- ✓ Synchronous bi direction half-duplex bus
- ✓ Standard speed of 100kHz, 400kHz, 1MHz
- ✓ The bus is composed of 2 wires
  - SDA: Serial Data
  - SCL: Serial Clock
- ✓ I2C peripheral is present on most microcontrollers. But a designer often needs to program it because pins for I2C are used for a more complex function

I2C Communication



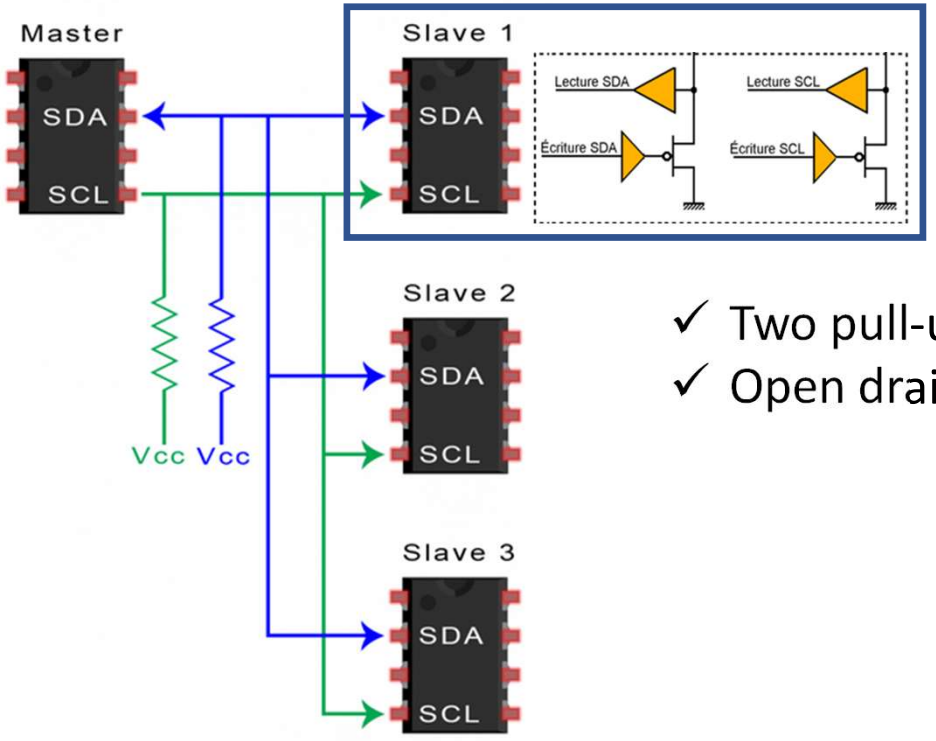
I2C is synchronized.

Multiple Slaves



Each slave has a different address.
The master selects the slave by sending the address of a given slave.
2 devices cannot have the same address! → address is often programmable by hardware

Multiple Slaves Electrical Connection



- ✓ Two pull-up resistors are required
- ✓ Open drain configuration allows

Start Condition

Start Condition

HIGH to LOW transition on the SDA line while the SCL line is held HIGH

Stop Condition

LOW to HIGH transition on the SDA line while the SCL line is held HIGH

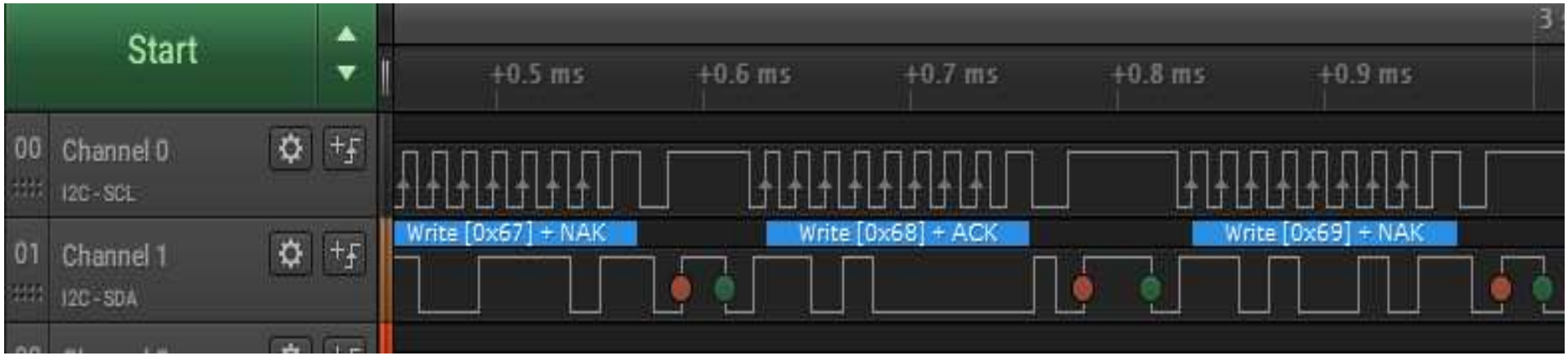
The SDA signal can only change when the SCL signal is low. When the clock is high the data should be stable

Addressing

- ✓ I2C does not have slave select line
- ✓ A device is selected by sending the address of the device
- ✓ The address frame is always the first frame after the start bit
- ✓ Communication
  - The master send the address to every connected slaves
  - Each slave compares the address sent to its own address
  - If the address matches, it sends a low voltage ACK bit back to the master
  - If the address does not match, the slave does nothing

Scanning devices on bus

✓ If slave address is unknow, it possible to scan the bus by trying addresses

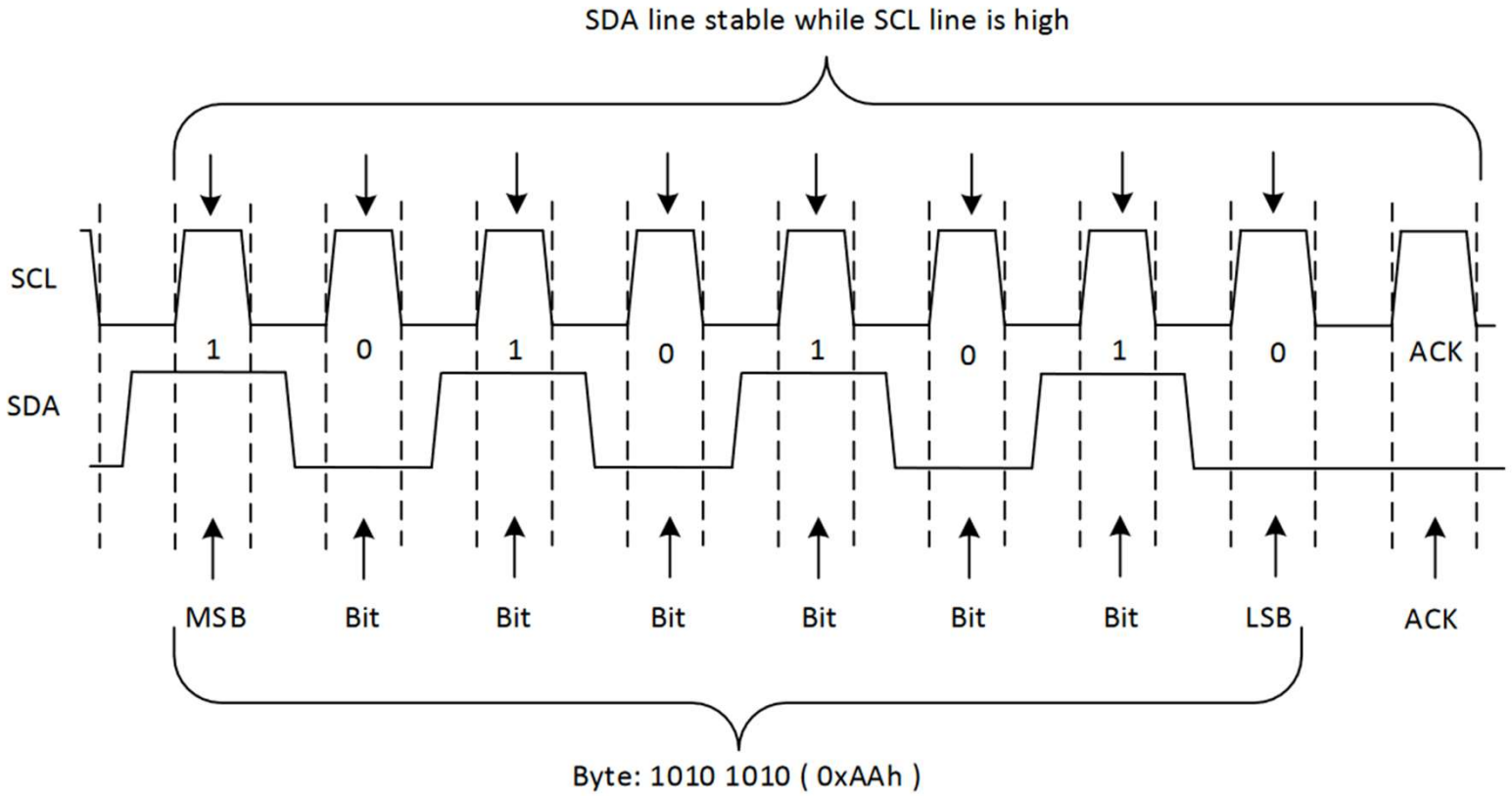


In the above example, a device with address 0x68 is available, unlike addresses 0x67 and 0x69

Acknowledge (1/2)

- ✓ Each byte of data(including the address byte) is followed by one ACK bit from the receiver.
- ✓ The ACK bit allows the receiver to communicate to the transmitter hat the byte was successfully received and another byte may be sent.
- ✓ Before the receiver can send an ACK, the transmitter must release the SDA line.
- ✓ To send an ACK bit, the receiver shall pull down the SDA line during the low phase of the ACK/NACK-related clock period (period 9),
- ✓ so that the SDA line is stable low during the high phase of the ACK/NACK-relatedclockperiod.
- ✓ When the SDA line remains high during the ACK/NACK-related clock period, this is interpreted as a NACK.

Acknowledge (2/2)



Read/Write

- ✓ The address frame includes a single bit at the end that informs the salve whether the master wants to write data to it or receive data from it.
- ✓ If the master wants to send data to the slave, the read/write bit is a low voltage level.
- ✓ If the master is requesting data from the slave, the bit is high voltage level

Advantages / Disavantages I2C

ADVANTAGES	DISADVANTAGES
Uses only two wires	Slower data transfer than SPI due to open Drain configuration
Supports multiples masters and multiple slaves	Size of data frame is limited to 8 bits
ACK/NACK bit gives confirmation that each frame is transferred successfully	Current consumption due to pull-up resistors

LPS22H

- ✓ The LPS22H is a pressure sensor from STMicroelectronics that integrates an I2C bus
- ✓ This sensor address is 0x5D
- ✓ LPS22H has a register called WHO_AM_I. This register has a fixed value and allows checking the communication is good between the microcontroller and

Table 16. Registers address map

Name	Type	Register Address	Default	Function and comment
		Hex	Binary	
Reserved		00 - 0A	-	Reserved
INTERRUPT_CFG	R/W	0B	00000000	Interrupt register
THS_P_L	R/W	0C	00000000	Pressure threshold registers
THS_P_H	R/W	0D	00000000	
Reserved		0E	-	Reserved
WHO_AM_I	R	0F	10110001	Who am I

The WHO_AM_I register is at the address 0F in the matrix and equal to 0xB1

How to Read WHO_AM_I register

- ✓ Send the address of the LPS22H and let it know we want to send information
  - The LPS22H will acknowledge
- ✓ Send the address of the register where the WHO_AM_I register is located
- ✓ Read the value of the WHO_AM_I register

Table 14. Transfer when master is receiving (reading) one byte of data from slave

Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

Meaning of the Frame

Table 14. Transfer when master is receiving (reading) one byte of data from slave

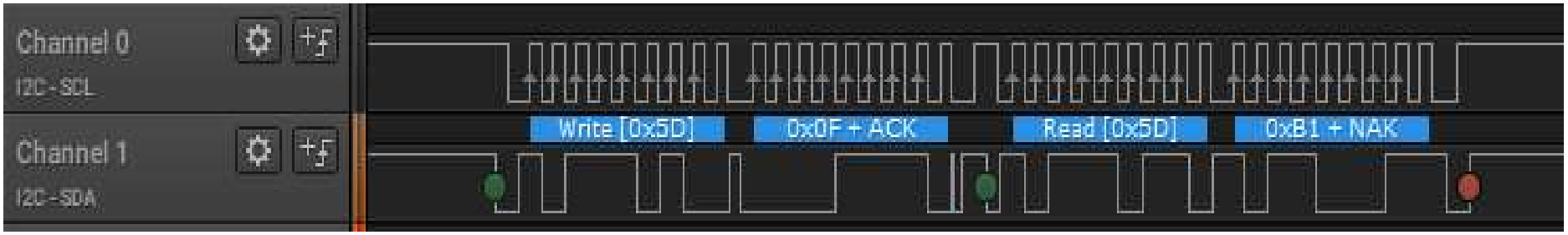
Master	ST	SAD + W		SUB		SR	SAD + R			NMAK	SP
Slave			SAK		SAK			SAK	DATA		

ST: Start
SAD+W : Slave Address + Write bit
SAK: Slave Acnowledge
SUB: Sub-address
SR: Start repeated

SAD+R: Slave Address + Read bit
DATA: Value of the register we want to read
NMAK: No Master Acknowledge
SP: STP

Depending on data organization, protocol can be different. **Read device specifications**

Meaning of the Frame



- ✓ Master sends address of the LPS22H with writing bit : 0x5D + write
- ✓ Master sends address of the matrix for the data it wants to read (0x0F: WHO_AM_I register)
- ✓ Master sends address of the LPS22H with reading bit : 0x5D + read
- ✓ Master reads value of the WHO_AM_I register: 0xB1

ISEN | école d'ingénieurs
ALL IS DIGITAL!

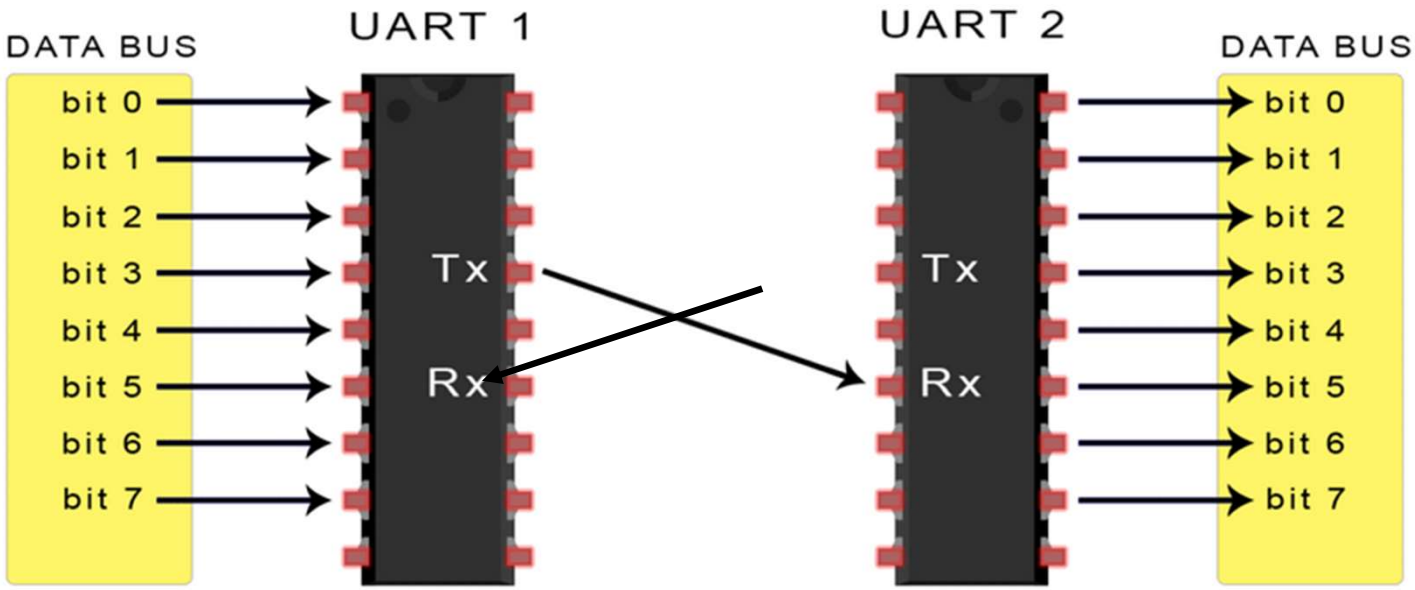
Standard Communication

UART

UART

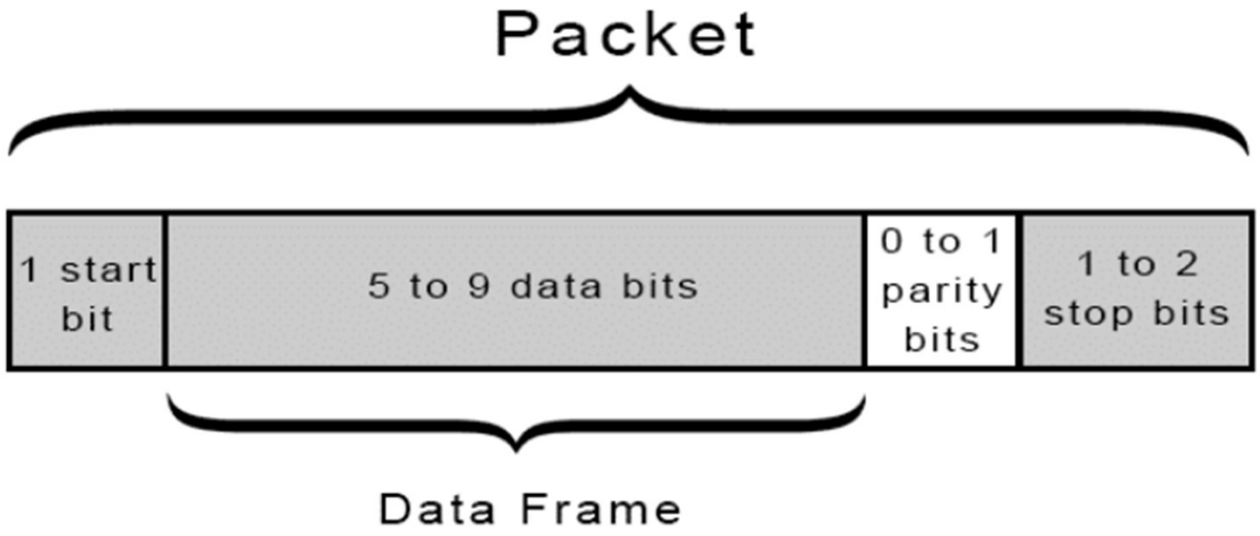
- ✓ UART stands for Universal Asynchronous Receiver Transmitter
- ✓ Was designed for PC to convert parallel bus into serial link (replaced by USB today)
- ✓ Is asynchronous (no clock) bus full duplex
- ✓ Synchronous bi direction half-duplex bus
- ✓ Standard speed is up to 115200 baud, usually 9600 baud
- ✓ The bus is composed of 2 wires
  - Tx : Transmitter
  - Rx : Receiver

UART Communication



UART is asynchronized (there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART)

packet



UART transmitted data is organized into packets; Each packet contains 1 start bit, 5 to 9 data bits (depending on device), an optional parity bit, and 1 or 2 stop bits

Start bit

- ✓ The UART data transmission line is normally held at a high voltage level when it's not transmitting data.
- ✓ To start the transfer of data, the transmitting UART pulls the transmission line from high to low for one clock cycle.
- ✓ When the receiving UART detects the high to low voltage transition, it begins reading the bits in the data frame at the frequency of the baud rate.

Data Frame

- ✓ The data frame contains the actual data being transferred.
- ✓ It can be 5 bits up to 8 bits long if a parity bit is used.
- ✓ If no parity bit is used, the data frame can be 9 bits long. I
- ✓ In most cases, the data is sent with the least significant bit first

Parity and Stop Bit

Parity

- ✓ Parity bit increases robustness of communication
- ✓ If the parity bit is a 0 (even parity), the 1 bits in the data frame should total to an even number.
- ✓ If the parity bit is a 1 (odd parity), the 1 bits in the data frame should total to an odd number.

Stop bit

- ✓ To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage for at least two bit durations

Advantages / Disavantages UART

ADVANTAGES	DISADVANTAGES
Uses only two wires	Size of data frame is limited to a maximum of 9 bits
No clock signal is necessary	Does not support multiple slaves or multiple master systems
Has a parity bit to allow for error checking	Baud rates of each UART must be within 10% of each other
Structure of data packet can be changed as long as both sides are set up for it (it is just a PHY layer. No protocol)	