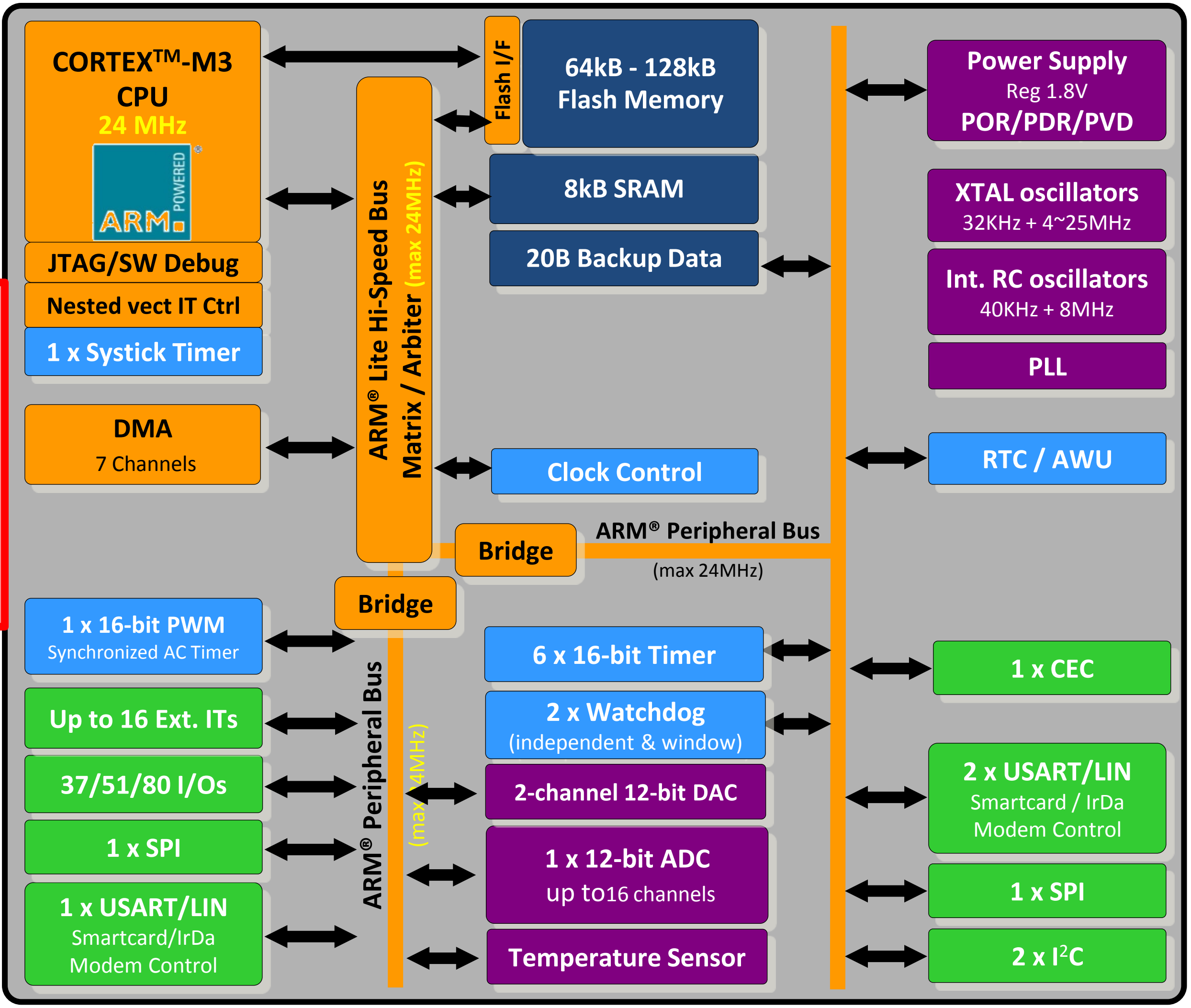
IO and Peripheral Interfaces

STM32 Value line 64K-128KBytes System Diagram

- **Core and operating conditions**
  - ARM® Cortex™-M3
  - 1.25 DMIPS/MHz up to 24 MHz
  - 2.0 V to 3.6 V range
  - -40 to +105 °C
- **Rich connectivity**
  - 8 communications peripherals
- **Advanced analog**
  - 12-bit1.2 µs conversion time ADC
  - Dual channel 12-bit DAC
- **Enhanced control**
  - 16-bit motor control timer
  - 6x 16-bit PWM timers
- **LQFP48, LQFP/BGA64, LQFP100**



CORTEX™-M3 CPU
24 MHz
JTAG/SW Debug
Nested vect IT Ctrl
1 x Systick Timer
DMA
7 Channels
Flash I/F
64kB - 128kB Flash Memory
8kB SRAM
20B Backup Data
ARM® Lite Hi-Speed Bus Matrix / Arbiter (max 24MHz)
Clock Control
Bridge
ARM® Peripheral Bus
(max 24MHz)
Bridge
ARM® Peripheral Bus (max 24MHz)
Power Supply
Reg 1.8V
POR/PDR/PVD
XTAL oscillators
32KHz + 4~25MHz
Int. RC oscillators
40KHz + 8MHz
PLL
RTC / AWU
1 x 16-bit PWM
Synchronized AC Timer
Up to 16 Ext. ITs
37/51/80 I/Os
1 x SPI
1 x USART/LIN
Smartcard/IrDa
Modem Control
6 x 16-bit Timer
2 x Watchdog
(independent & window)
2-channel 12-bit DAC
1 x 12-bit ADC
up to16 channels
Temperature Sensor
1 x CEC
2 x USART/LIN
Smartcard / IrDa
Modem Control
1 x SPI
2 x I^2C

Interfaces

- Digital Interface
  - Several protocols for inter-chip communication

    UART, I^{2}C, SPI, USB,…
  - Serial communication protocols
  - Meant for short distances “inside the box”
  - Low complexity
  - Low cost
  - Low speed (a few Mbps at the fastest)
  - Serial communication is employed where it is not practical, either in physical or cost terms, to move data in parallel between systems.
- Analog Interface
  - ADC (Analog Digital Converter)
  - DAC (Digital Analog Converter)
  - Comparator

Peripherals registers

- Microcontroller has peripherals registers mapped in the memory space address
- Each peripheral has registers to configure its functionality and read and write input/output data

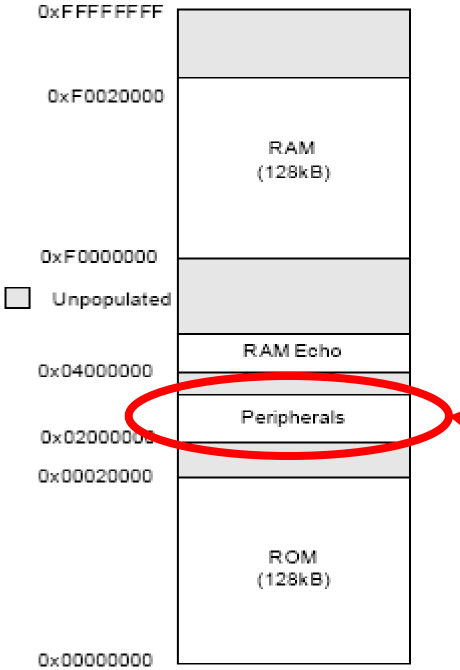
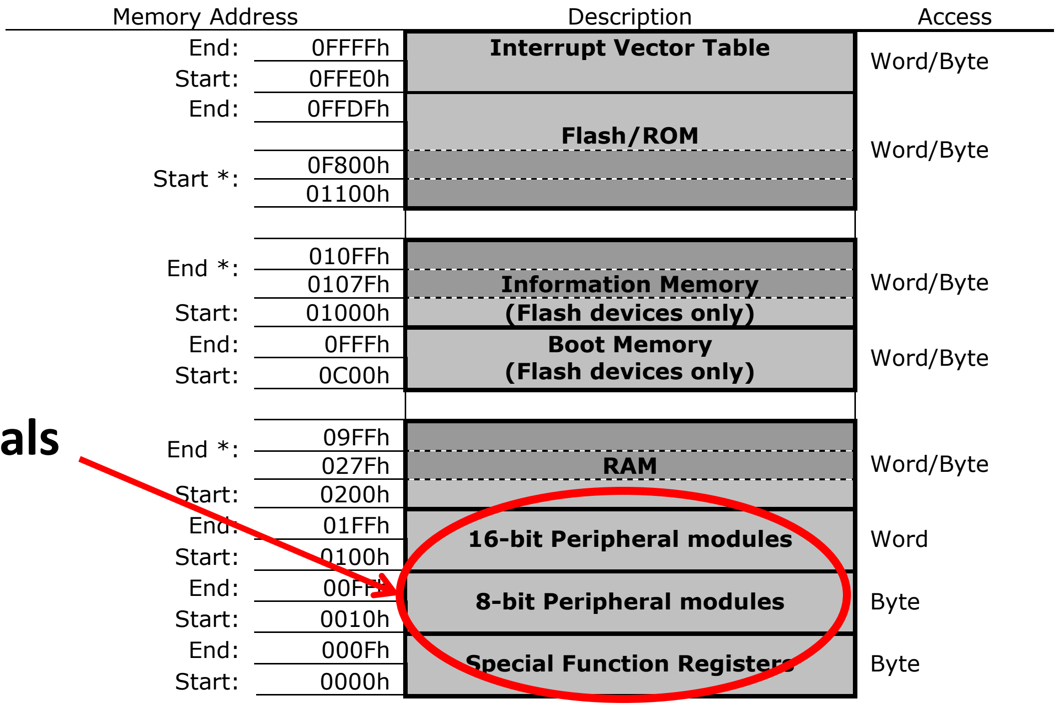


Figure 5: JN5148 Memory Map

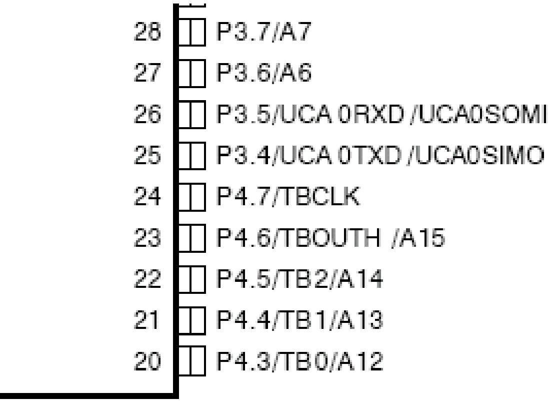
Peripherals

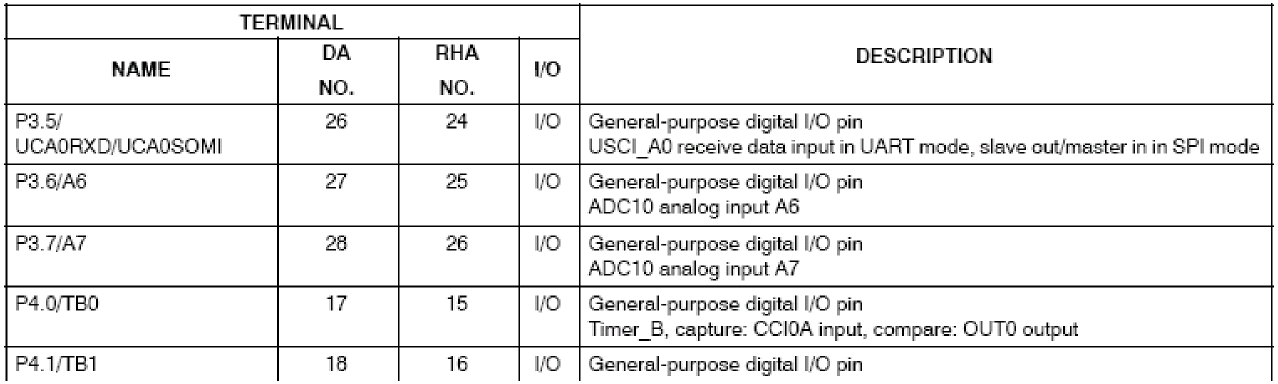


Memory Address		Description	Access
End:	0FFFFh	**Interrupt Vector Table**	Word/Byte
Start:	0FFE0h		
End:	0FFDFh	**Flash/ROM**	Word/Byte
	0F800h		
Start *:	01100h		
End *:	010FFh	**Information Memory (Flash devices only)**	Word/Byte
	0107Fh		
Start:	01000h		
End:	0FFFh	**Boot Memory (Flash devices only)**	Word/Byte
Start:	0C00h		
End *:	09FFh	**RAM**	Word/Byte
	027Fh		
Start:	0200h		
End:	01FFh	**16-bit Peripheral modules**	Word
Start:	0100h		
End:	00FFh	**8-bit Peripheral modules**	Byte
Start:	0010h		
End:	000Fh	**Special Function Registers**	Byte
Start:	0000h		

Microcontroller External Pin Configuration

- Microcontroller pins can be assigned to different function and peripherals
- Each pin can:
  - Perform input, output, and interrupt functionality (general purpose input output: GPIO);
  - Set pull-up/pull-down resistor;
  - Be assigned to digital/analog peripheral as UARTs, SPIs, I2C, ADC, DAC...



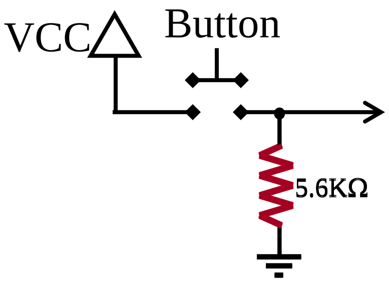


TERMINAL				DESCRIPTION
NAME	DA NO.	RHA NO.	I/O	
P3.5/ UCA0RXD/UCA0SOMI	26	24	I/O	General-purpose digital I/O pin USCI_A0 receive data input in UART mode, slave out/master in in SPI mode
P3.6/A6	27	25	I/O	General-purpose digital I/O pin ADC10 analog input A6
P3.7/A7	28	26	I/O	General-purpose digital I/O pin ADC10 analog input A7
P4.0/TB0	17	15	I/O	General-purpose digital I/O pin Timer_B, capture: CCI0A input, compare: OUT0 output
P4.1/TB1	18	16	I/O	General-purpose digital I/O pin

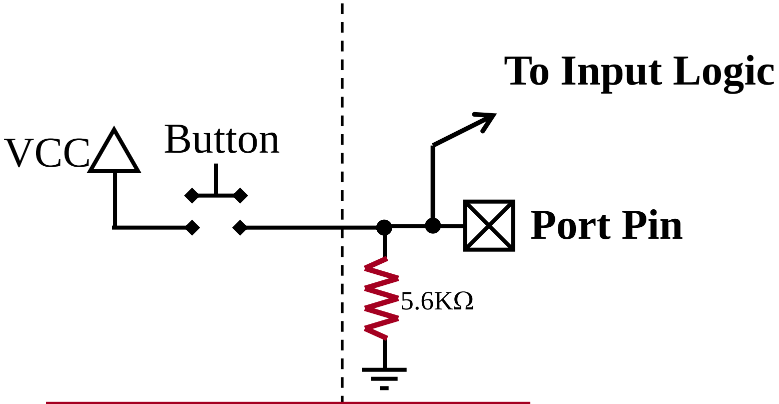
General Purpose I/O - GPIO

Avoid floating inputs!!!

- Use a pull-up/down resistor, GND, or internal programmable logic
- A floating inputs can cause short circuit current in the input circuit!



Button produces either Vcc or Floating input. Adding a pull-down resistor fixes it.



Some ports have internal programmable resistors

STM32 GPIO Features

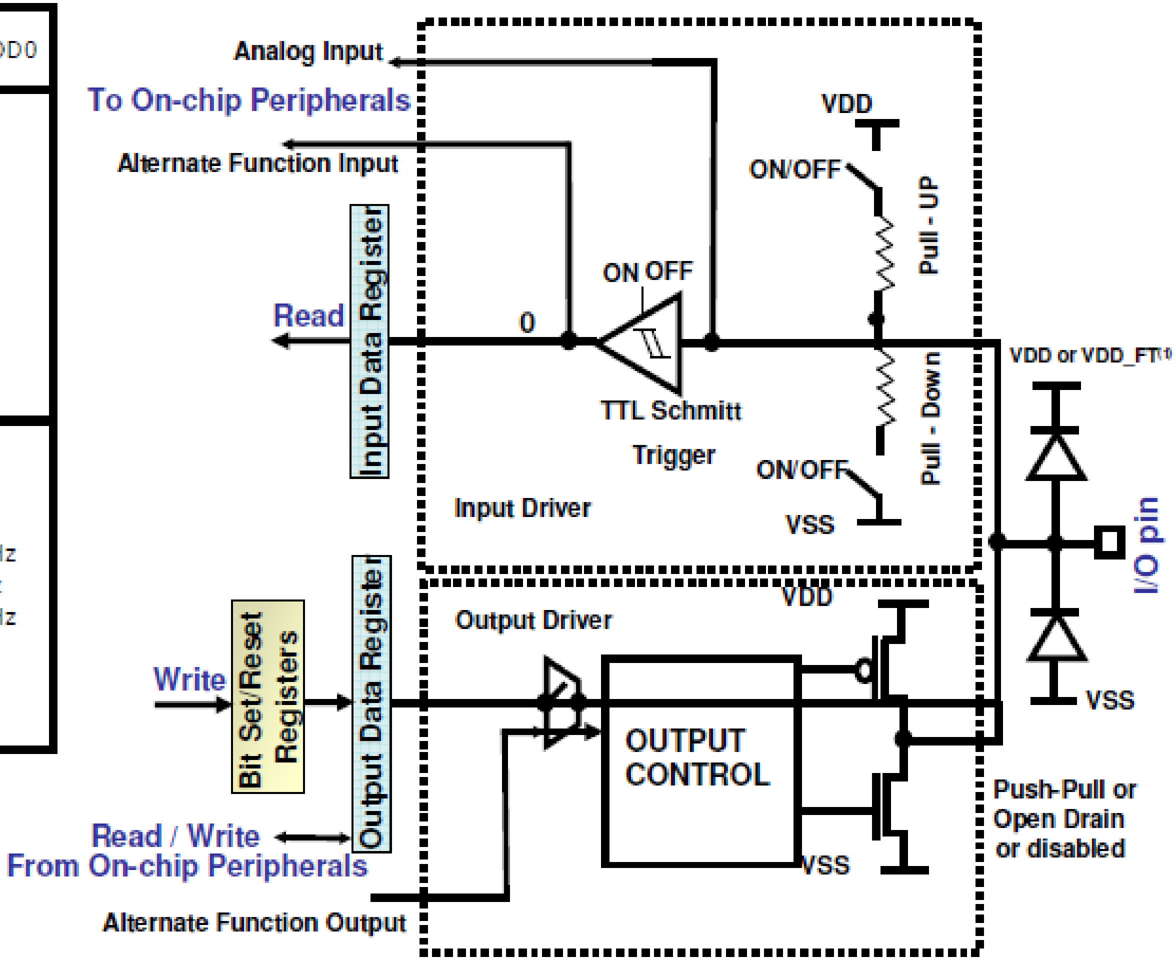
Up to 80 multifunction bi-directional I/O ports available: 80% IO ratio

- Standard I/Os 5V tolerant
- The GPIOs can sink 25mA (total currents sunk is 150mA)
- 18 MHz Toggling
- Configurable Output Speed up to 50 MHz
- Up to 16 Analog Inputs
- Alternate Functions pins (like USARTx, TIMx, I2Cx, SPIx, CAN, USB…)
- Up to 80 GPIOs can be set-up as external interrupt (up to 16 lines at time)
- One I/O can be used as Wake-Up from STANDBY (PA.00)
- One I/O can be set-up as Tamper Pin (PC.13)
- All Standard I/Os are shared in 5 ports (GPIOA..GPIOE)
- Atomic Bit Set and Bit Reset using BSRR and BRR registers
- Locking mechanism to avoid spurious write in the IO registers
  - When the LOCK sequence has been applied on a port bit, it is no longer possible to modify the configuration of the port bit until the next reset (no write access to the CRL and CRHregisters corresponding bit).

General-Purpose IO for STM32

Configuration Mode	CNF1	CNF0	MOD1	MOD0
Analog Input	0	0	00	
Input Floating **(Reset State)**	0	1		
Input Pull-Up	1	0		
Input Pull-Down	1	1		
Output Push-Pull	0	0	01: 10 MHz 10: 2 MHz 11: 50 MHz	
Output Open-Drain	0	1		
AF Push-Pull	1	0		
AF Open-Drain	1	1		

(1) ***VDD for standard I/Os and VDD_FT is a potential specific to five-volt tolerant I/Os and different from VDD.***



Digital (Serial) Interfaces

I^{2}C

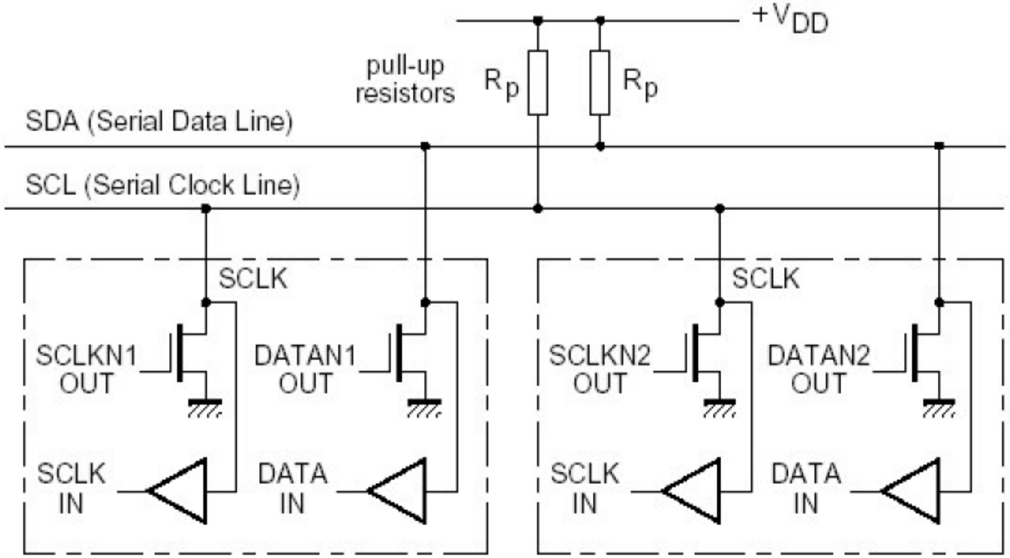
- Shorthand for an “Inter-integrated circuit” bus
- I2C devices include EEPROMs, thermal sensors, and real-time clocks
- Used as a control interface to signal processing devices that have separate data interfaces, e.g. RF tuners, video decoders and encoders, and audio processors.
- I^{2}C bus has three speeds:
  - Slow (under 100 Kbps)
  - Fast (400 Kbps)
  - High-speed (3.4 Mbps) – I^{2}C v.2.0
- Limited to about 3 meters for moderate speeds

I^{2}C (Inter-Integrated Circuit) protocol

- Communications is always initiated and completed by the master, which is responsible for generating the clock signal;
- In more complex applications, I^{2}C can operate in multi-master mode;
- The slave selection by the master is made using the seven-bit address of the target slave;
- The master (in transmit mode) sends:
  - Start bit;
  - 7-bit address of the slave it wishes to communicate with;
  - A single bit representing whether it wishes to write (0) to or read (1) from the slave;
  - The target slave will acknowledge its address.

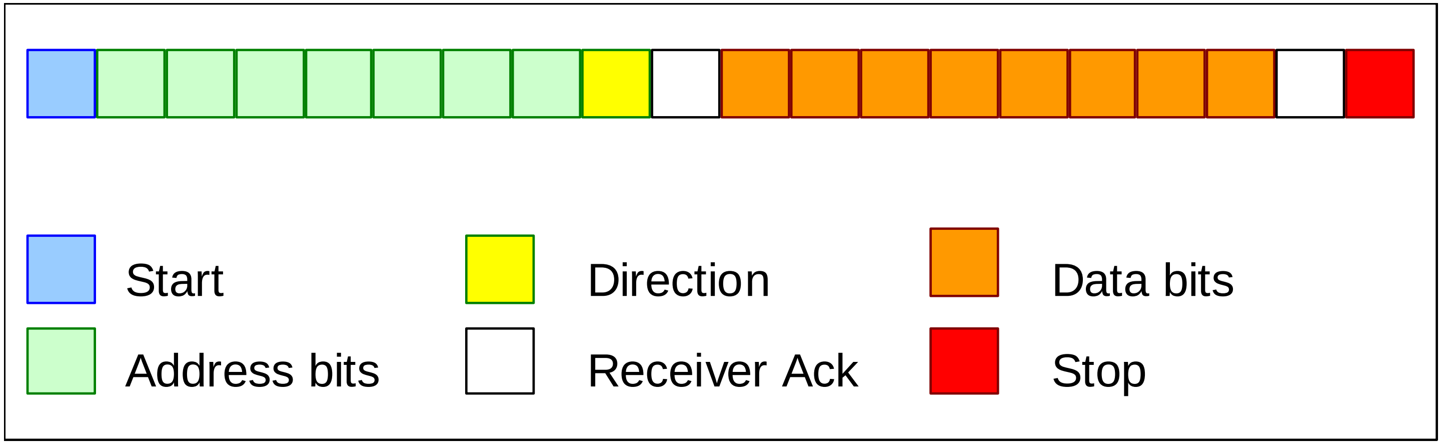
I^{2}C Bus Configuration

- 2-wire serial bus – Serial data (SDA) and Serial clock (SCL)
- Half-duplex, synchronous, multi-master bus
- No chip select or arbitration logic required
- Lines pulled high via resistors, pulled down via open-drain drivers (wired-AND, avoid short circuit among the bus)



I^{2}C Features

- "Clock stretching" – when the slave (receiver) needs more time to process a bit, it can pull SCL low. The master waits until the slave has released SCL before sending the next bit.
- "General call" broadcast – addresses every device on the bus
- 10-bit extended addressing for new designs. 7-bit addresses all exhausted



🟦 Start
🟩 Address bits
🟨 Direction
□ Receiver Ack
🟧 Data bits
🟥 Stop

I^{2}C Registers

The I2C peripheral has several registers to:

- Set external pin (enable I2C function in a specific chip pin)
- Set transmission clock frequency
- Set Read or Write transmission
- Write slave address
- Enable/Disable stop bit
- Write output data
- Read input data
- Start/Stop the transmission
- Enable/Disable DMA controller
- Enable/Disable Interrupt
- Check I2C status

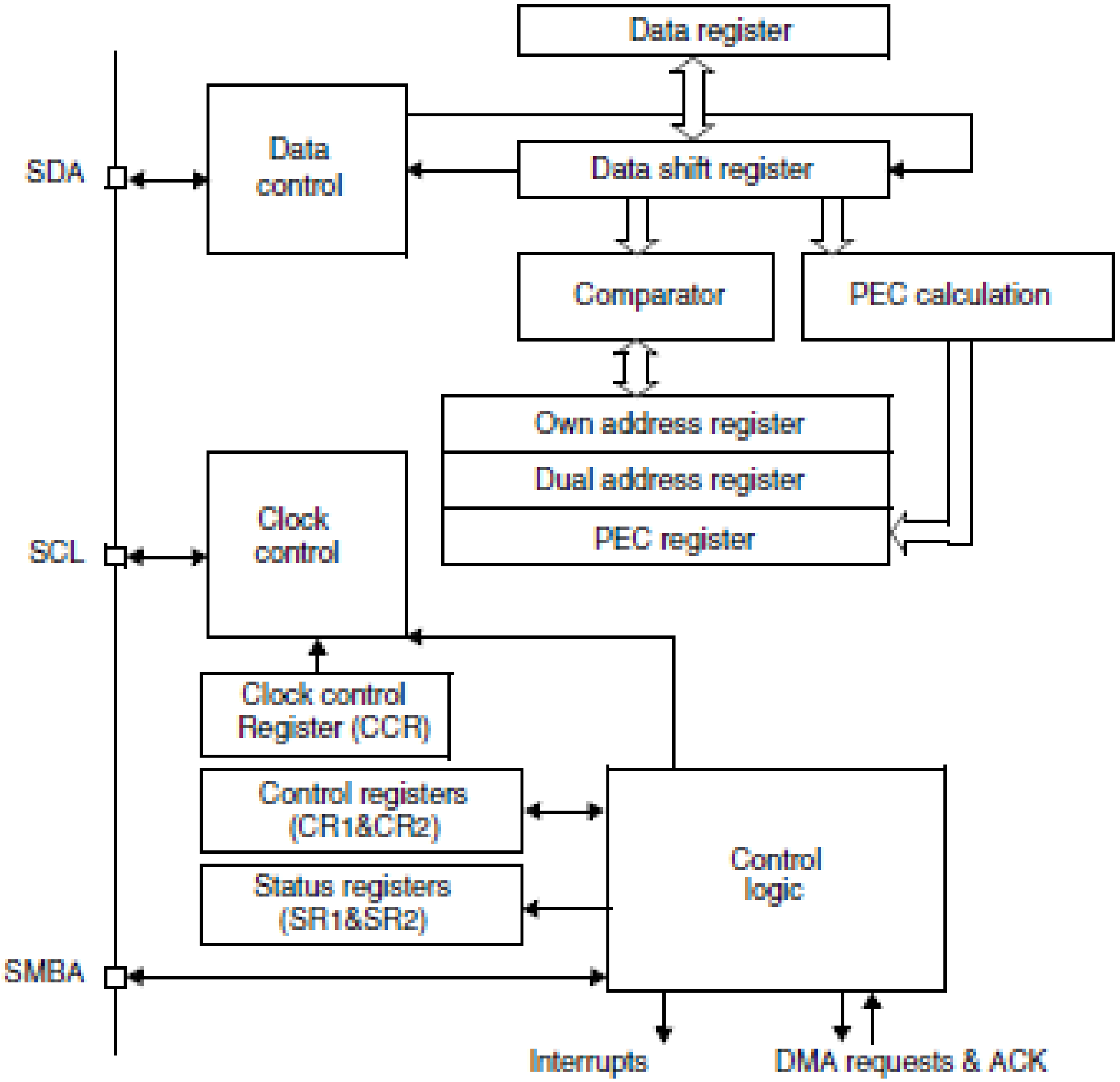
STM32 I2C Features (1/2)

- Multi Master and slave capability
- Controls all I²C bus specific sequencing, protocol, arbitration and timing
- Standard and fast I²C mode (up to 400kHz)
- 7-bit and 10-bit addressing modes
- Dual Addressing Capability to acknowledge 2 slave addresses
- Status flags:
  - Transmitter/Receiver mode flag
  - Byte transfer finished flag
  - I2C busy flag
- Configurable PEC (Packet Error Checking) Generation or Verification:
  - PEC value can be transmitted as last byte in Tx mode
  - PEC error checking for last received byte

STM32 I2C Features (2/2)

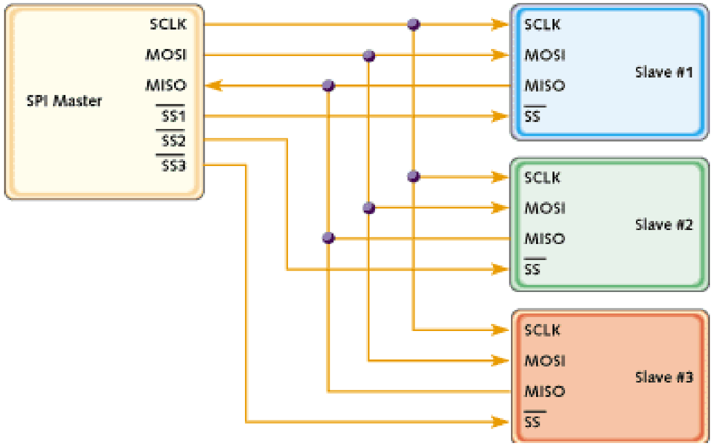
- Error flags:
  - Arbitration lost condition for master mode
  - Acknowledgement failure after address/ data transmission
  - Detection of misplaced start or stop condition
  - Overrun/Underrun if clock stretching is disabled
- 2 Interrupt vectors:
  - 1 Interrupt for successful address/ data communication
  - 1 Interrupt for error condition
- 1-byte buffer with DMA capability
- SMBus 2.0 Compatibility
- PMBus Compatibility

STM32 I2C Block diagram

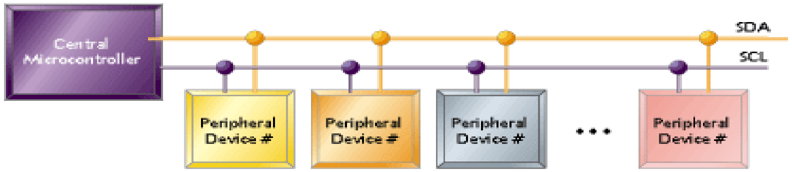


SPI vs. I^{2}C

- For point-to-point, SPI is simple and efficient
  - Less overhead than I^{2}C due to lack of addressing, plus SPI is full duplex.
- For multiple slaves, each slave needs separate slave select signal
  - SPI requires more effort and more hardware than I^{2}C



SPI



I^{2}C

SPI

- Shorthand for “Serial Peripheral Interface”
- Defined by Motorola on the MC68HCxx line of microcontrollers
- Generally faster than I^{2}C, capable of several Mbps

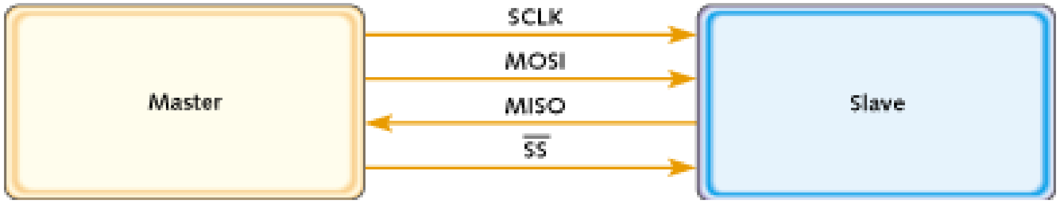
Applications:

- Like I^{2}C, used in EEPROM, Flash, and real time clocks
- Better suited for “data streams”, i.e. ADC converters
- Full duplex capability, i.e. communication between a codec and digital signal processor

Serial Peripheral Interface (SPI) protocol

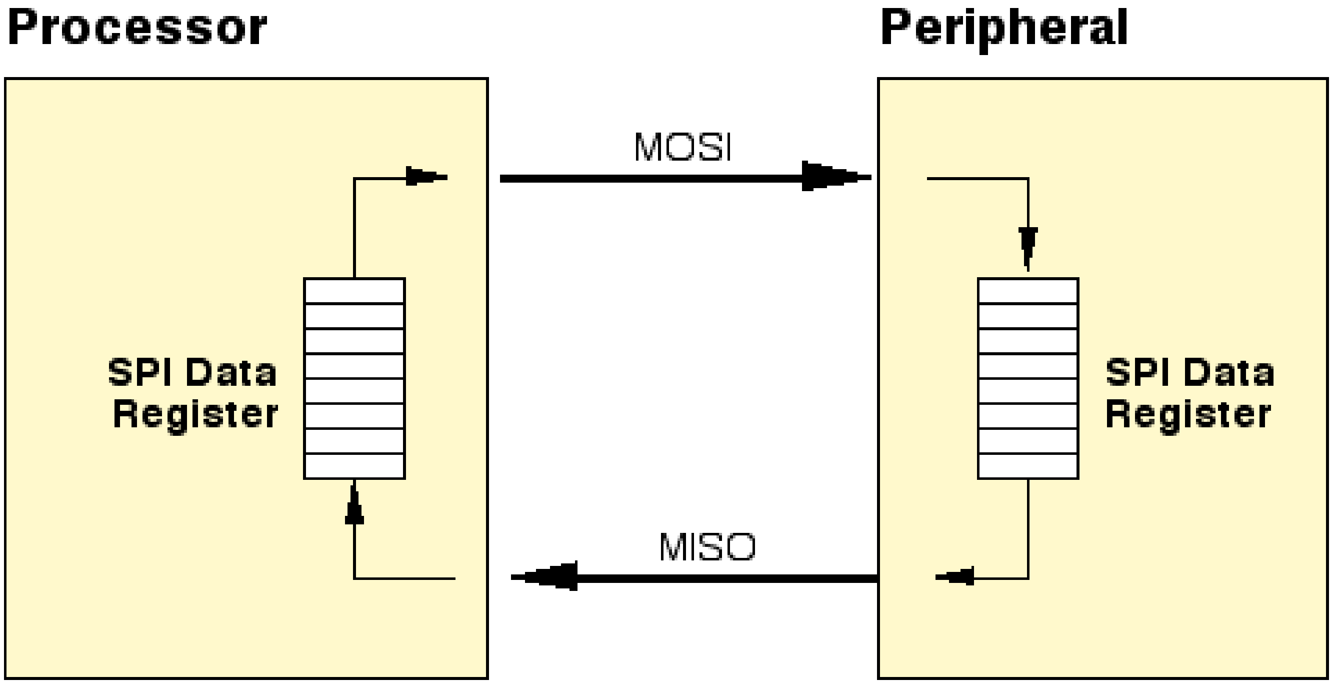
- Supports only one master;
- Can support more than a slave;
- Short distance between devices, e.g. on a printed circuit boards (PCBs);
- Special attention needs to be observed to the polarity and phase of the clock signal;
- The master sends data on one edge of clock and reads data on the other edge. Therefore, it can send/receive at the same time.

SPI Bus Configuration



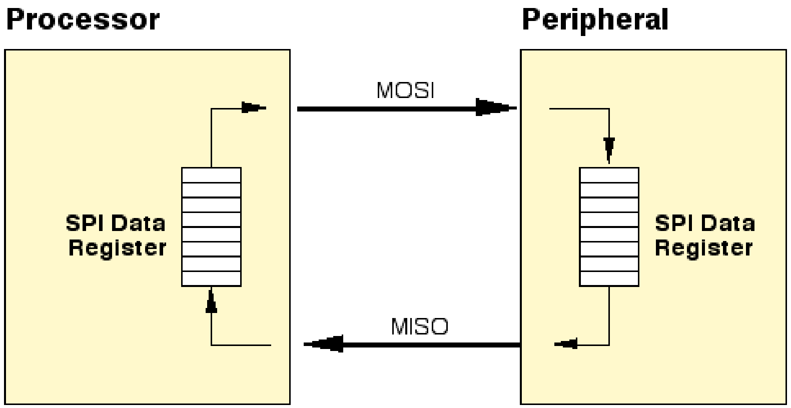
- Synchronous serial data link operating at full duplex
- Master/slave relationship
- 2 data signals:
  - MOSI – master data output, slave data input
  - MISO – master data input, slave data output
- 2 control signals:
  - SCLK – clock
  - /SS – slave select (no addressing)

SPI structure



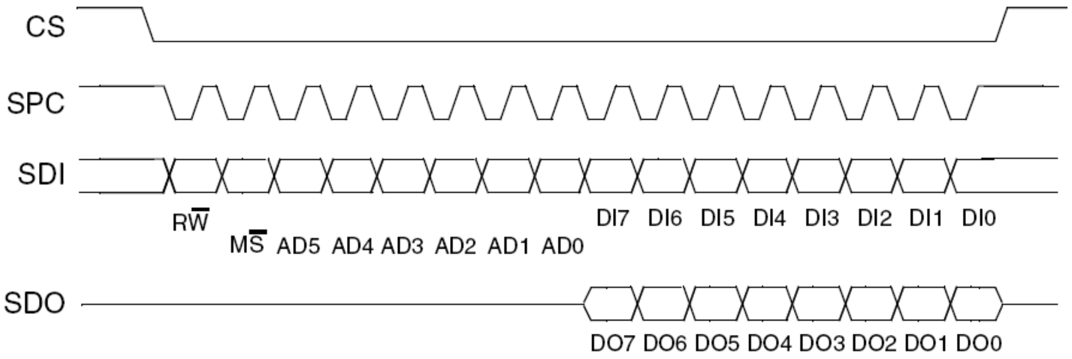
- As the register transmits the byte to the slave on the MOSI signal line, the slave transfers the contents of *its* shift register back to the master on the MISO signal line, exchanging the contents of the two shift registers.

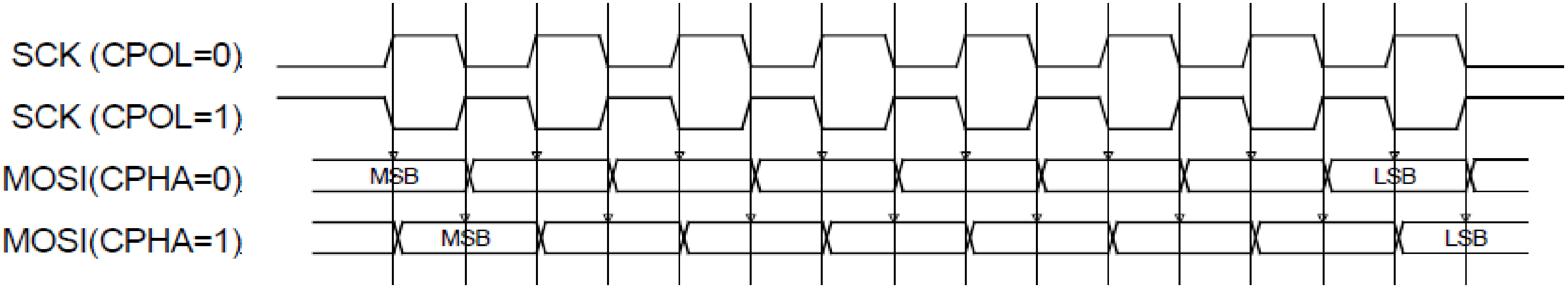
SPI Operating Mode



SPI has 4 operating modes, each of them differ to the others for clock polarity and phase.

- Clock polarity set if the clock will start with high or low value
- Clock phase set if the data will be latched on rising or falling edge of clock


CS
SPC
SDI
RW̄
MS̄ AD5 AD4 AD3 AD2 AD1 AD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0
SDO
DO7 DO6 DO5 DO4 DO3 DO2 DO1 DO0



SPI Registers

The SPI peripheral has several registers to:

- Set external pin (enable SPI function in a specific chip pin)
- Set transmission clock frequency
- Set clock polarity
- Set clock phase
- Enable/Disable slave select
- Write output data
- Read input data
- Start/stop transmission
- Enable/Disable interrupt
- Enable/Disable DMA
- Check SPI status

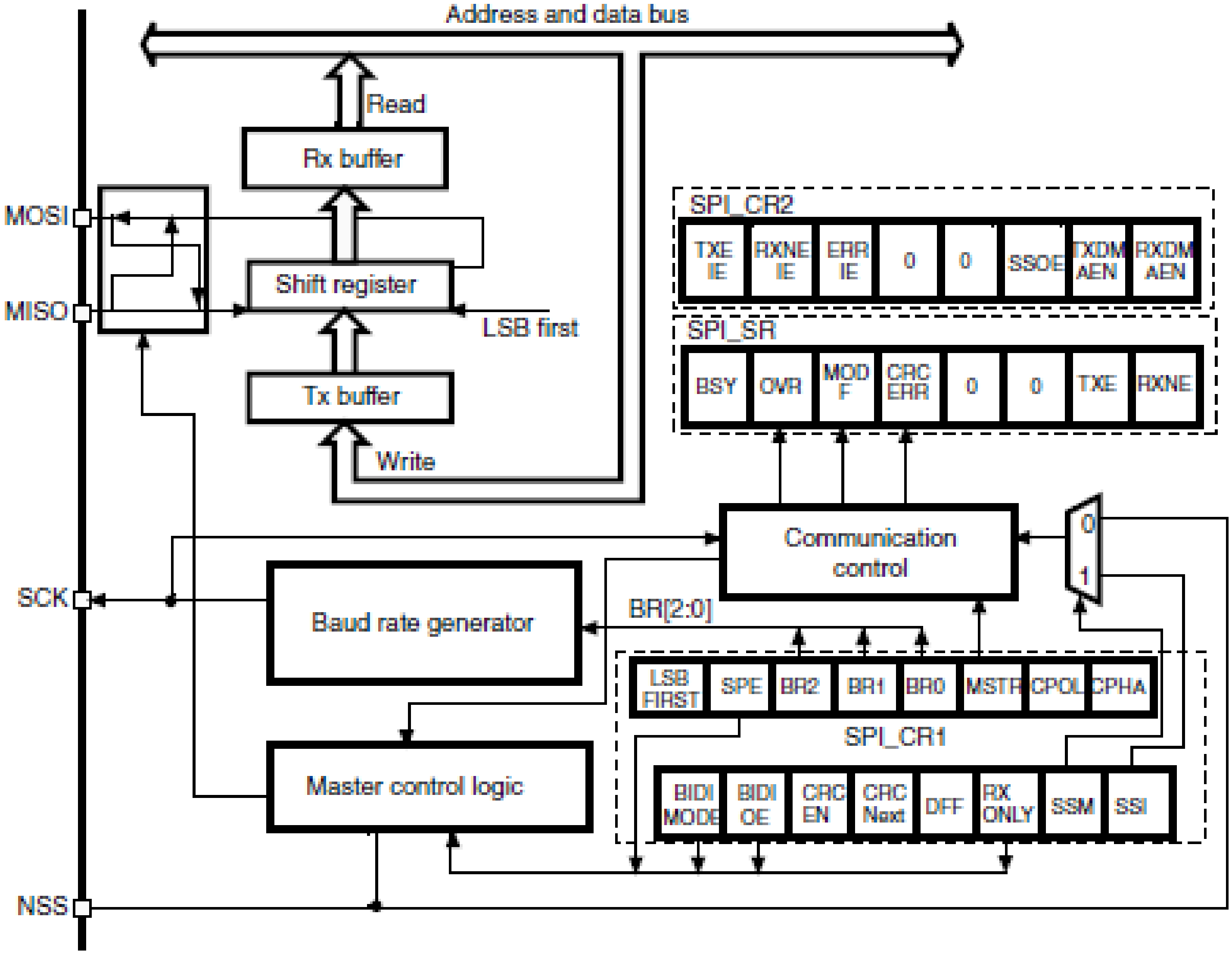
STM32 SPI Features (1/2)

- Two SPIs: SPI1 on high speed APB2 and SPI2 on low speed APB1
- Full duplex synchronous transfers on 3 lines
- Simplex synchronous transfers on 2 lines with or without a bi-directional data line
- Programmable data frame size :8- or 16-bit transfer frame format selection
- Programmable data order with MSB-first or LSB-first shifting
- Master or slave operation
- Programmable bit rate: up to 18 MHz in Master/Slave mode
- NSS management by hardware or software for both master and slave:
  - Dynamic change of Master/Slave operations

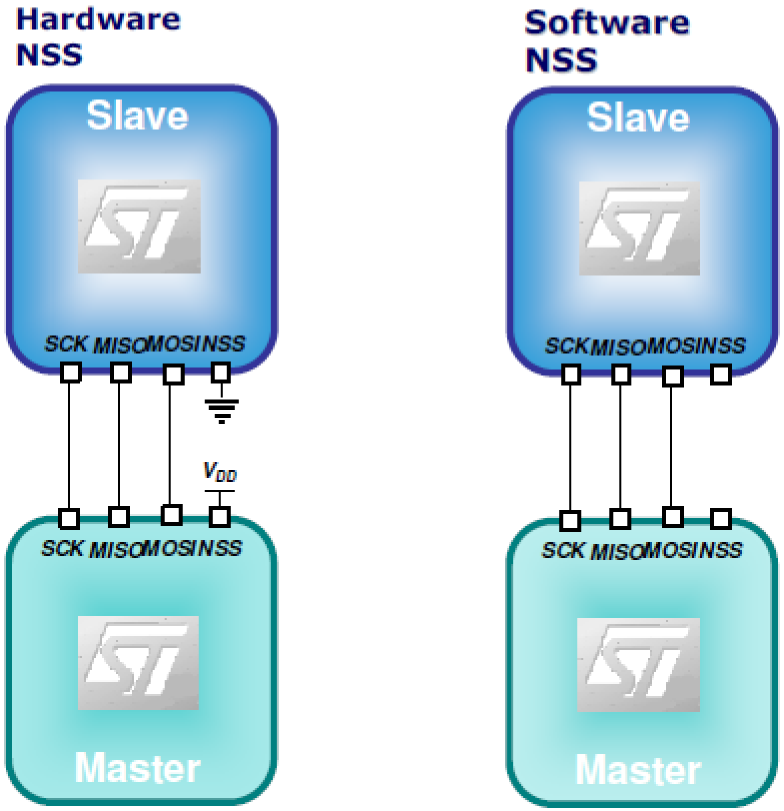
STM32 SPI Features (2/2)

- Programmable clock polarity and phase
- Dedicated transmission and reception flags (Tx buffer Empty and Rx buffer Not Empty) with interrupt capability
- SPI bus busy status flag
- Master mode fault and overrun flags with interrupt capability
- Hardware CRC feature for reliable communication
- Support for DMA
  - Each SPI has a DMA Tx and Rx requests
  - Each of the SPIs requests is mapped on a different DMA channel:
  - Possibility to use DMA for all SPIs transfer direction in the same time
  - Calculated CRC value is automatically transmitted at the end of data transfer

SPI Block Diagram (STM32)

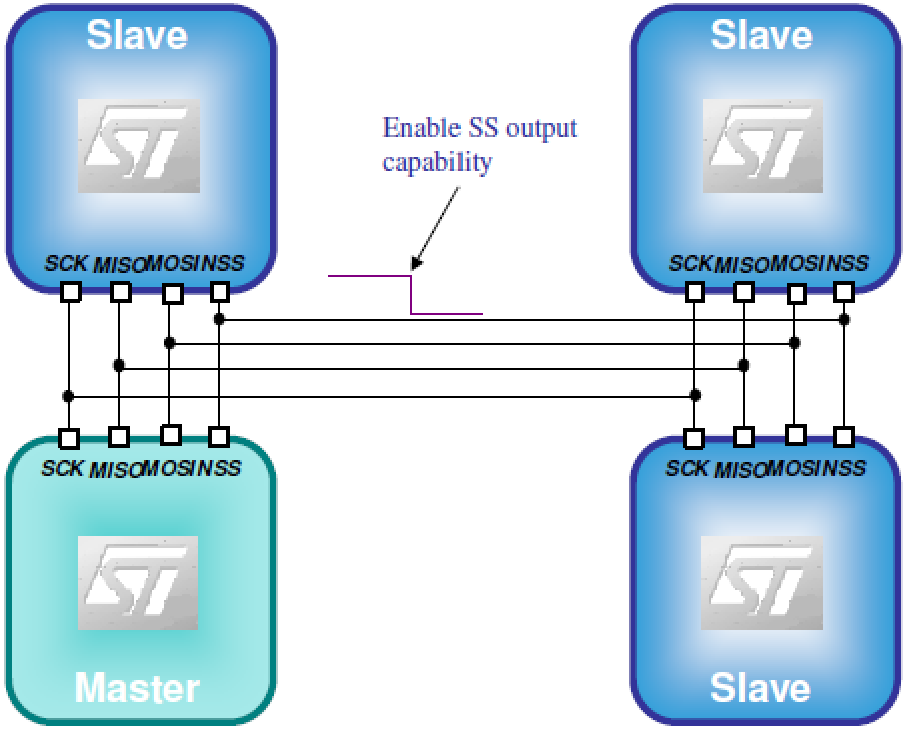


NSS HW & SW Management



- Both Master and Slave NSS pins could be used for other purpose
- Provides the possibility of dynamic change of Master/Slave operations: No hardware limitation to switch from master to slave or slave to master in the same application

Multi-Master NSS Management



- Each device can be a unique master by enabling its NSS as output and driving it low: all other devices became slaves.
- Rx-only mode No need for external GPIO pin to drive slaves NSS pins

UART

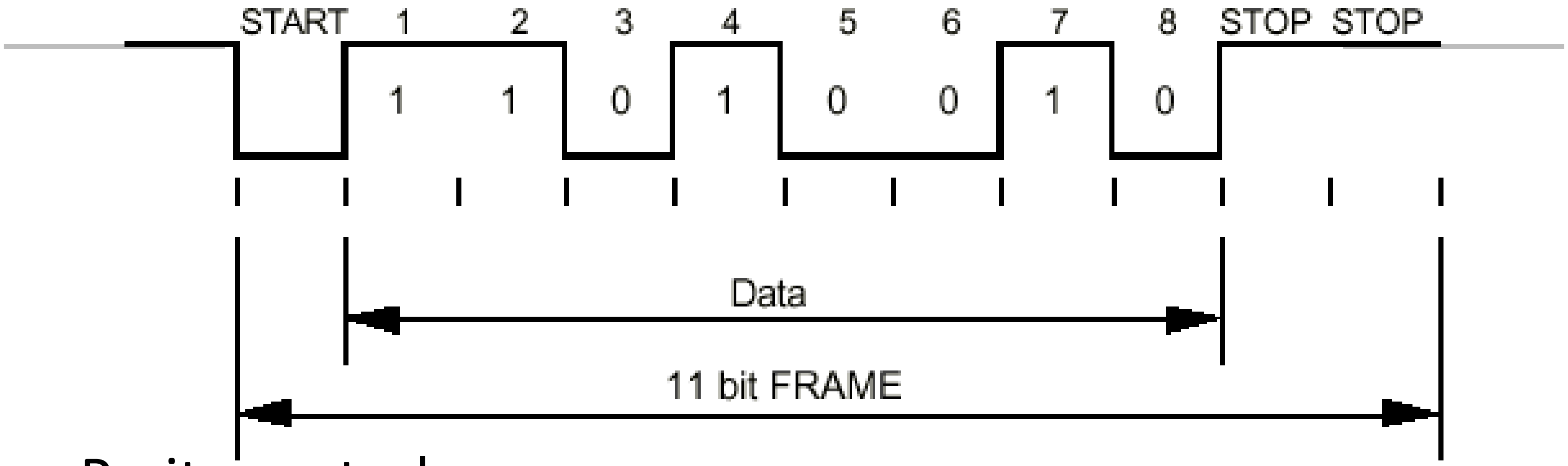
- Shorthand for “Universal Asynchronous Receiver-Transmitter “
- A UART’s transmitter is essentially just a parallel-to-serial converter with extra features.
- The UART bus is a full-duplex bus.
- The essence of the UART transmitter is a shift register that is loaded in parallel, and then each bit is sequentially shifted out of the device on each pulse of the serial clock.
- Application:
  - Communication between microprocessors, pc
  - Used to interface the microcontroller with others transmission bus as: RS232, RS485, USB, CAN BUS, KNX, LonWorks ecc.
  - Used to connect microntroller with modem and transceiver as: telephone modem, Bluetooth, WIFi, GSM/GPRS/HDPSA

UART

- Asynchronous serial devices, such as UARTs, do not share a common clock
- Each device has its own, local clock.
- The devices must operate at exactly the same frequency.
- Logic (within the UART) is required to detect the phase of the transmitted data and *phase lock* the receiver's clock to this.
- Bitrate: 2400, 19200, 57600,115200, 921600…
- One of the problems associated with serial transmission is reconstructing the data at the receiving end, because the clock is not transmitted.
- Difficulties arise in detecting boundaries between bits.

UART

- The transmission format uses:
  - 1 start bit at the beginning
  - Settable 5,6,7,8 data bits string length
  - Settable 1 or 0 even/odd parity bit control

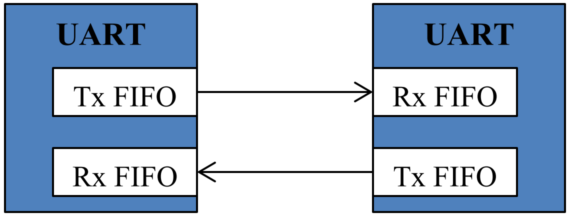


- Parity control
  - The parity bit control is accordingly set to 0 or 1 to have and odd number of frame 1 bits in odd parity either an even number of frame 1 bits in the even parity
  - The control can detect 1 bit error in the frame

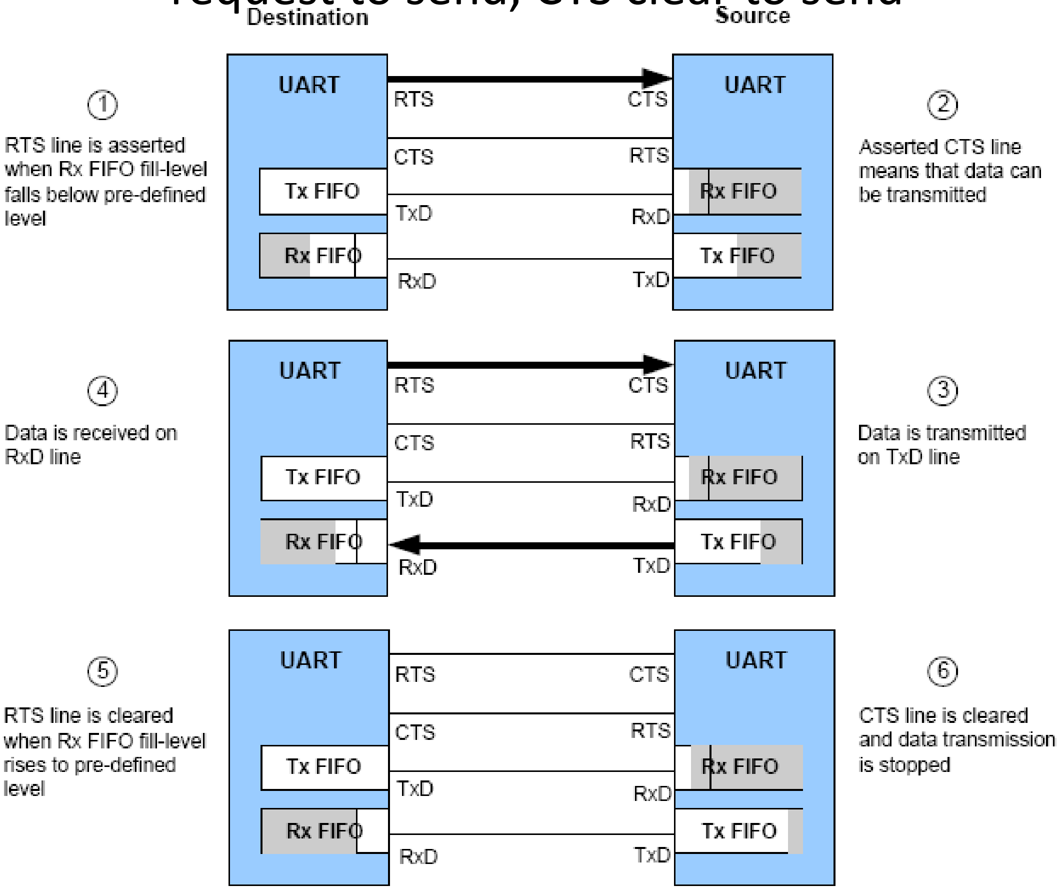
UART transmission

UART can transmit either with 2 or 4 wires

- 2 wires mode has transmit and receive lines



- 4 wires mode has transmit and receive lines plus 2 handshake signals, RTS request to send, CTS clear to send



① RTS line is asserted when Rx FIFO fill-level falls below pre-defined level

② Asserted CTS line means that data can be transmitted

③ Data is transmitted on TxD line

④ Data is received on RxD line

⑤ RTS line is cleared when Rx FIFO fill-level rises to pre-defined level

⑥ CTS line is cleared and data transmission is stopped

UART Registers

The UART peripheral has several registers to:

- Set external pin (enable UART function in a specific chip pin)
- Set transmission bitrate
- Set stop bit numbers, parity control
- Set data string length
- Set 2 or 4 wires mode
- Enable/Disable handshake control
- Write output data
- Read input data
- Start/stop transmission
- Enable/Disable interrupt
- Enable/Disable DMA
- Check UART status and bit errors

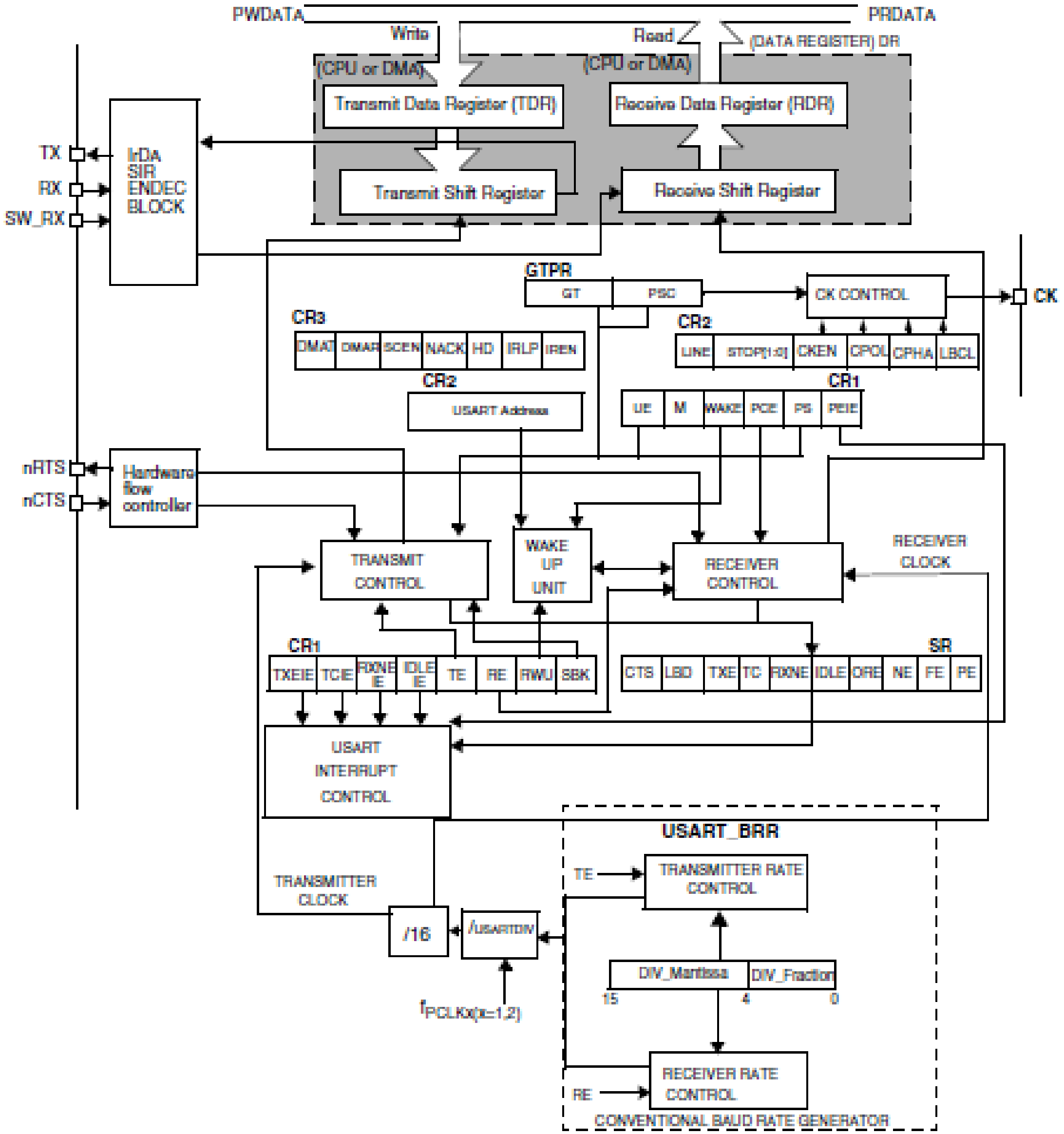
STM32 USART Features (1/2)

- Three USART: USART1 High speed APB2 and USART2,3 on Low speed APB1
- Data can be 8 or 9 bits
- Even, odd or no-parity bit generation and detection
- 0.5, 1, 1.5 or 2 stop bit generation
- Programmable baud rate generator
  - Integer part (12 bits)
  - Fractional part (4 bits)
- Support hardware flow control (CTS and RTS)
- Dedicated transmission and reception flags (TxE and RxNE) with interrupt capability
- Support for DMA
  - Receive DMA request
  - Transmit DMA request

STM32 USART Features (2/2)

- 10 interrupt sources to ease software implementation
- LIN Master/Slave compatible
- Synchronous Mode: Master mode only
- IrDA SIR Encoder Decoder
- Smartcard Capability
- Single wire Half Duplex Communication (a co simplex na SPI)
- Multi-Processor communication
  - USART can enter Mute mode
  - Mute mode: disable receive interrupts until next header detected
  - Wake up from mute mode (by idle line detection or address mark detection)

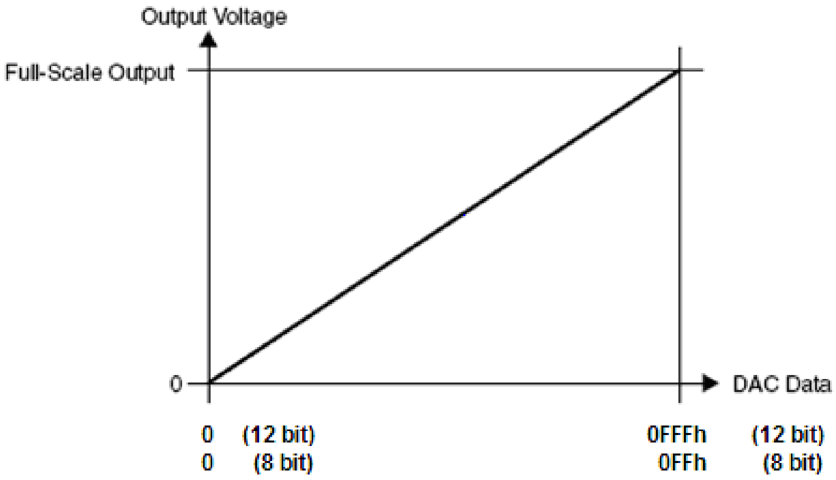
STM32 USART Block Diagram



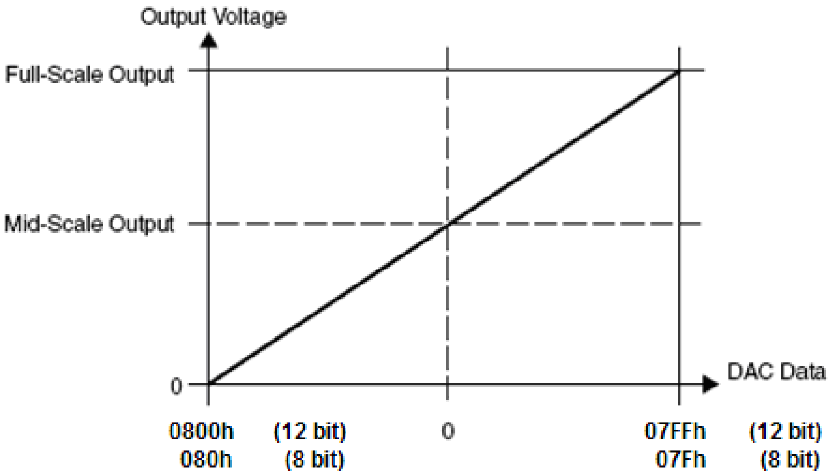
DAC & ADC Interfaces

Digital Analog Converter

- The inputs to a DAC are the digital value and a reference voltage V_{REF} to set the analogue output level;
- Provides a continuous time output signal, mathematically often treated as discrete Dirac pulses into a zero-order hold: a series of fixed steps;
- Filtering the discrete output signal can be used to approximate a continuous time signal, as well as:
  - Increasing the resolution;
  - Increasing the number of discrete levels and;
  - Reducing the level size (reduces the quantization error).



Straight binary



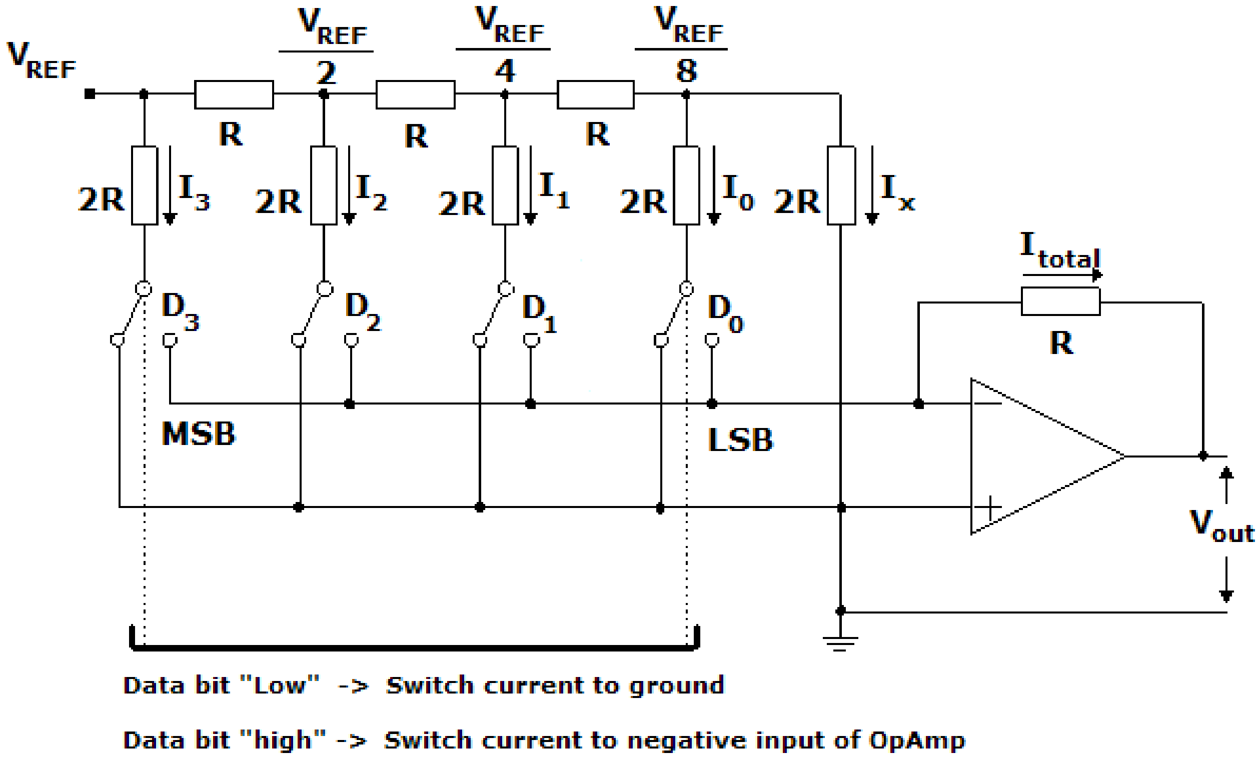
Two's complement

DAC types

- Binary Weighted DAC:
  - Contains one resistor (or current source) for each bit of the DAC connected to a common voltage source V_{REF};
  - There are accuracy problems (high precision resistors are required);
- R/2R Ladder DAC:
  - Binary weighted DAC that uses a repeating cascaded structure of resistors of value R and 2R;
- Pulse Width Modulator DAC:
  - A stable voltage (or current) is switched into a low-pass (LP) filter during a time period representative of the digital input value.

DAC types

- R/2R Ladder DAC:
  - Example: R/2R 4 bit DAC architecture:



Data bit "Low" -> Switch current to ground

Data bit "high" -> Switch current to negative input of OpAmp

  - Swit

und

DAC characteristic parameters

- Resolution (*n*):
  - Number of possible DAC output levels, 2^n (*n*: n.º of bits);
  - The Effective Number Of Bits (ENOB) is the actual resolution achieved by the DAC, taking into account errors like nonlinearity, signal-to noise ratio.
- Integral Non-Linearity (INL):
  - Deviation of a DAC's transfer function from a straight line.
- Differential NonLinearity (DNL):
  - Difference between an actual step height and the ideal value of 1 LSB;
  - DNL < 1 LSB, the DAC is monotonic, that is, no loss of data.

DAC characteristic parameters

- Offset error:
  - Analogue output voltage when the digital input is zero.
- Gain error:
  - Difference between the ideal maximum output voltage and the actual maximum value of the transfer function, after subtracting the offset error.
- Monotonicity:
  - Ability of the analogue output of the DAC to increase with an increase in digital code or the converse.
- Total Harmonic Distortion (THD):
  - Distortion and noise introduced to the signal by the DAC.
- Dynamic range:
  - Difference between the largest and the smallest signals.

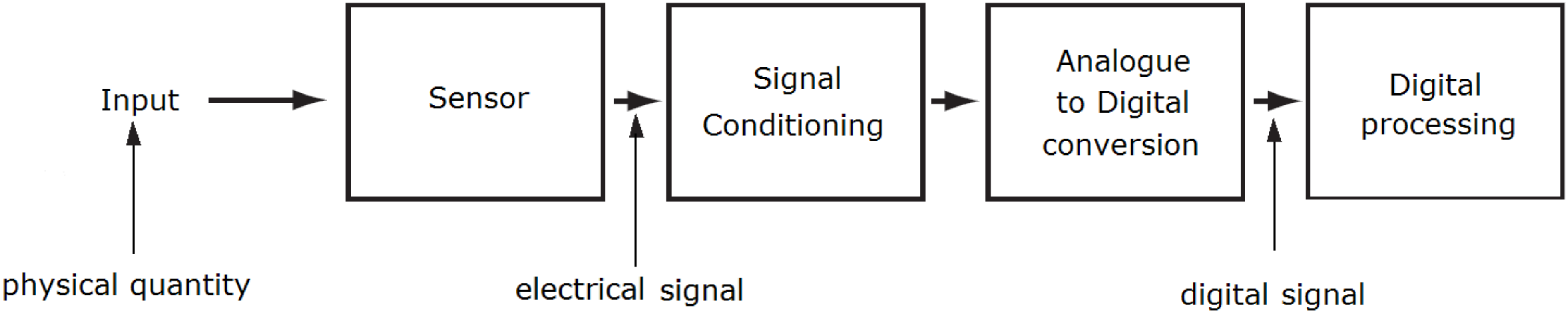
DAC Registers

The DAC peripheral has several registers to:

- Set external pin (enable DAC function in a specific chip pin)
- Set output signal refresh rate
- Set voltage reference value
- Write output value
- Enable/Disable output
- Enable/Disable DMA
- Check DAC status

Sensors data acquisition

Data acquisition system components:



Sensors:

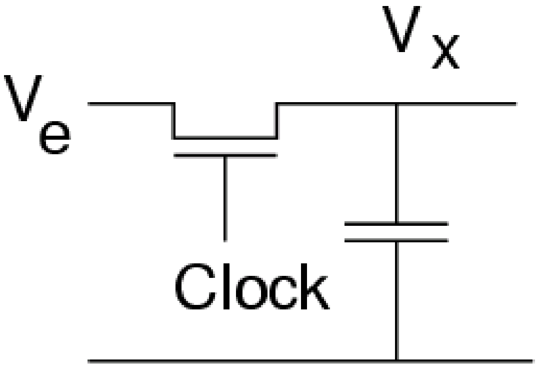
Convert analogue measurements of physical quantities (e.g. temperature, pressure, humidity, velocity, flow-rate, linear motion, position) into electrical signals (voltage or current).

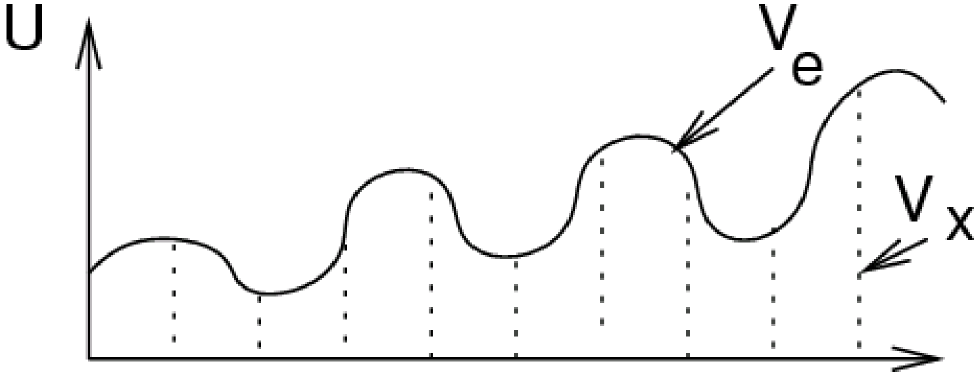
Data acquisition system components

- **Signal conditioning (filtering and amplification):**
  - The operations required to convert the measured analogue signal to the electrical signal range of the analogue-to-digital converter (ADC) may involve filtering, amplification, attenuation or impedance transformation.
- **Analogue-to-Digital Converter (ADC):**
  - Input: Signal to be measured;
  - Output: A digital code compatible with the digital processing system;
  - Requires:
    - Sample-and-hold: Used to take a snapshot of the continuously changing input signal and maintain the value over the sample interval set by a clock system;
    - A sampling frequency based on the Nyquist theorem.

ADC conversion Sample and Hold

: **Convert signal function in a series of value**



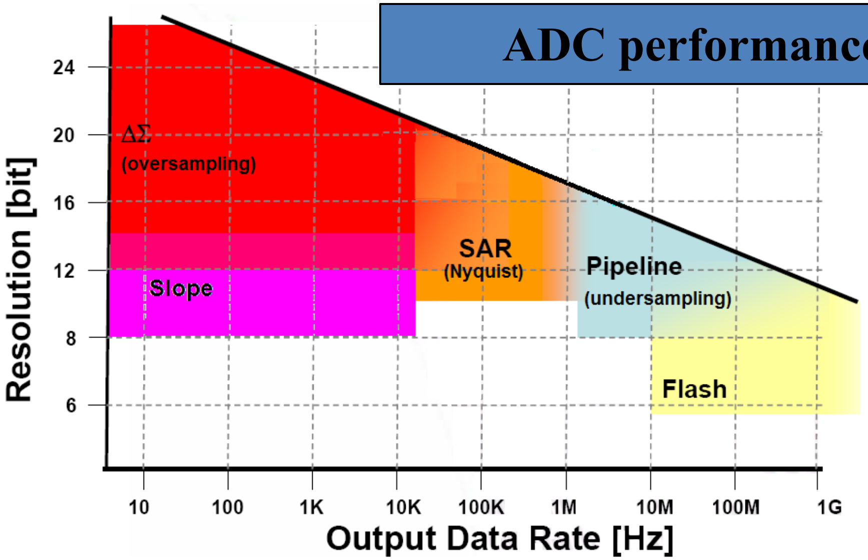


Restriction to digital information processing.
Known digital computers can only process discrete time series
Sample and hold-devices.
Ideally: width of clock pulse → 0

- *Sample and Hold circuit does the signal time quantization*

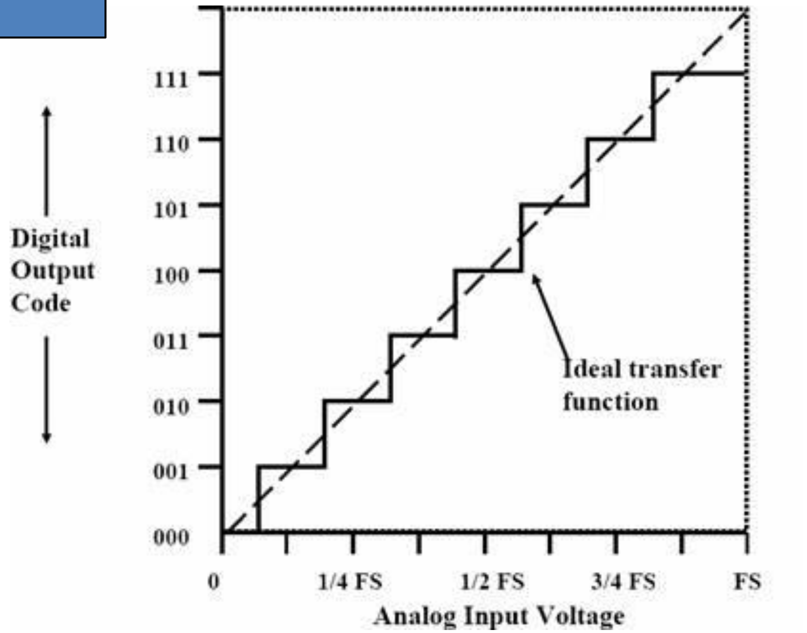
Analogue-to-Digital Converter (ADC)

- The ADC takes the voltage from the acquisition system (after signal conditioning) and converts it to an equivalent digital code;



ADC performance

-
  - Successive Approximation (SAR)
  - Sigma Delta (SD or ΔΣ)
  - Slope or Dual Slope
  - Pipeline
  - Flash



ADC ideal transfer function

ADC performance

- Resolution, *R*:
  - The resolution specifies the width of the digital output word;
    - 10, 12, 16 Bit ADC
  - The width of the word implies the smallest change to the analogue voltage that can be converted into a digital code;
  - The Least Significant Bit (LSB):

$$V_{LSB} = \frac{V_{ref}}{2^n}$$

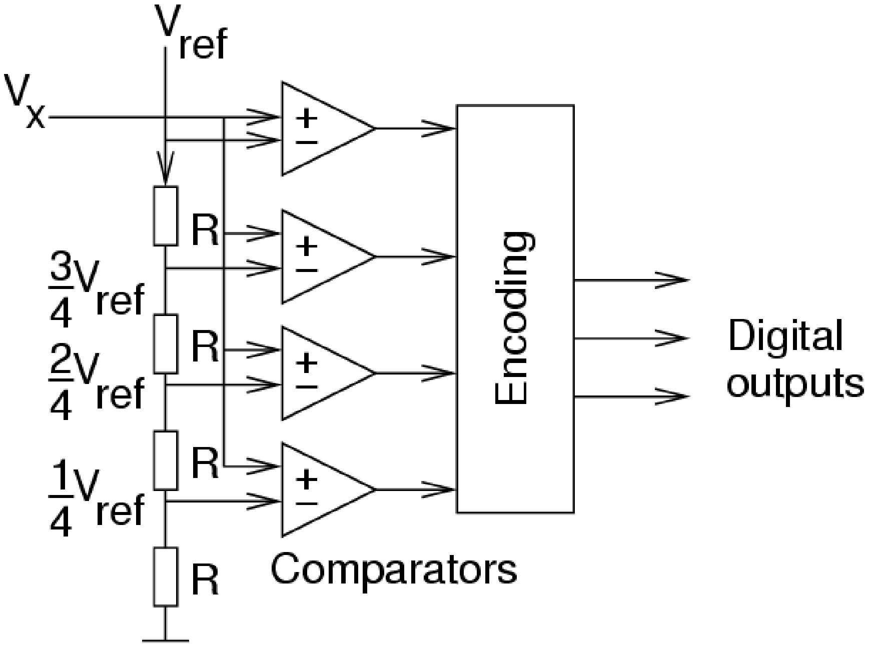
- Accuracy:
  - Degree of conformity of a digital code representing the analogue voltage to
- Speed:
  - Maximum output data rate expressed in sample per second (sps)

ADC Selection

- The selection of an ADC will depend on:
  - Voltage range to be measured;
  - Maximum signal frequency;
  - Minimum resolution needed *vs.* analogue input variation;
  - The need for differential inputs;
  - Voltage reference range;
  - The need for multiple channels for different analogue inputs.

ADC architecture	**Resolution**	**Conversion rate**	**Advantages**	**Disadvantages**
SAR	≤ 18 bit	< 5 Msps	Zero-cycle latency Low latency-time High accuracy Low power Simple operation	Sample rates 2-5 MHz
SD	≤ 24 bit ≤ 16-18 bit	< 625 ksps < 10 Msps	High resolution High stability Low power Moderate cost	Cycle-latency Low speed
Pipeline	≤ 16 bit	< 500 Msps	Higher speeds Higher bandwidth	Lower resolution Delay/Data latency Power requirements

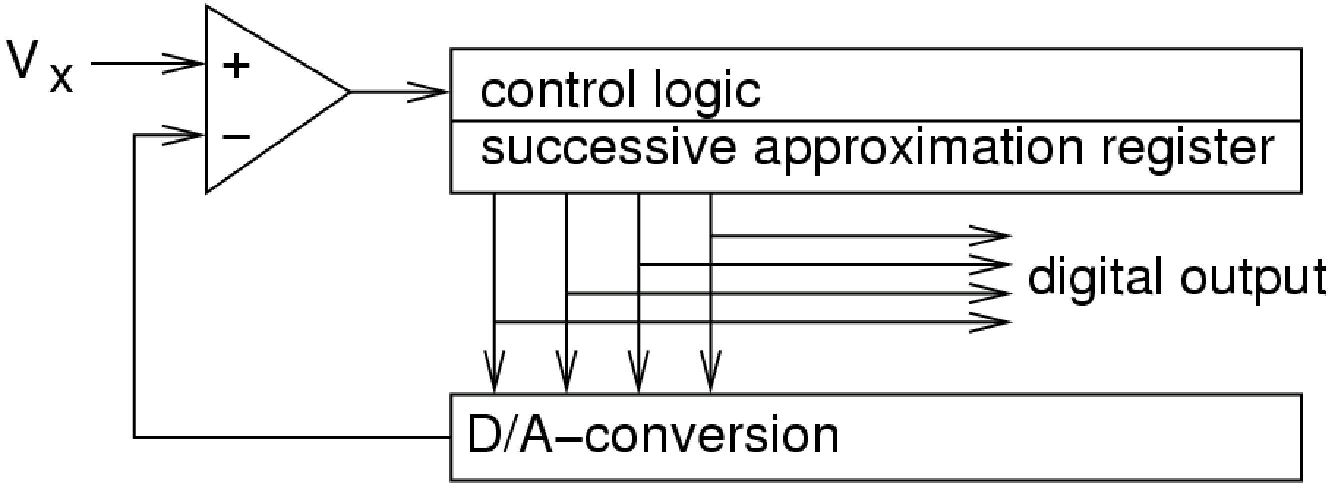
Flash A/D converter



Parallel comparison with reference voltage

- Speed: *O(1)*
- Hardware complexity: *O(n)*
- with *n*= # of distinguished voltage levels

Successive approximation



Key idea: binary search:
Set MSB='1'
if too large: reset MSB
Set MSB-1='1'
if too large: reset MSB-1

Speed: *O(ld(n))*

Hardware complexity: *O(ld(n))*

with *n*= # of distinguished voltage levels;

slow, but high accurate

ADC Registers

The ADC peripheral has several registers to:

- Set external pin (enable ADC function in a specific chip pin)
- Set sps (sample per second) rate
- Set sample/hold period
- Set voltage reference
- Read input value acquired
- Enable/Disable sampling
- Set interrupt
- Enable/Disable DMA
- Check ADC status

Interfacing to Sensors

Comparator Registers

The Comparator peripheral has several registers to:

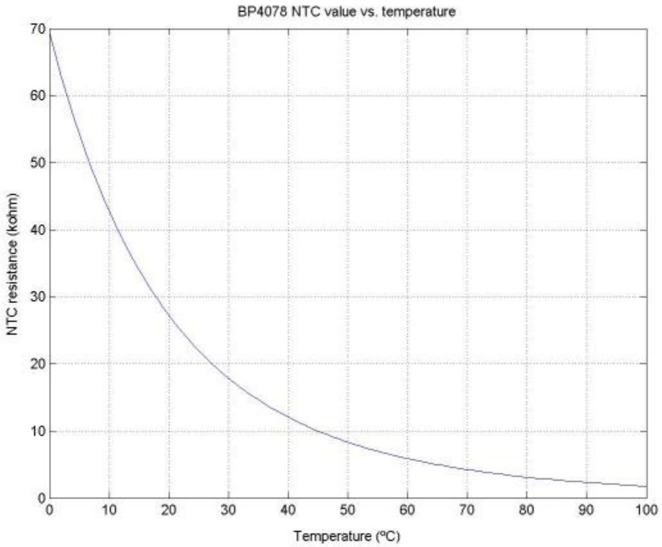
- Set external pin (enable comparator function in a specific chip pin)
- Voltage reference value
- Read input comparing acquired
- Enable/Disable comparator
- Set interrupt
- Check comparator status

Sensors data acquisition example

Case: read outdoor temperature

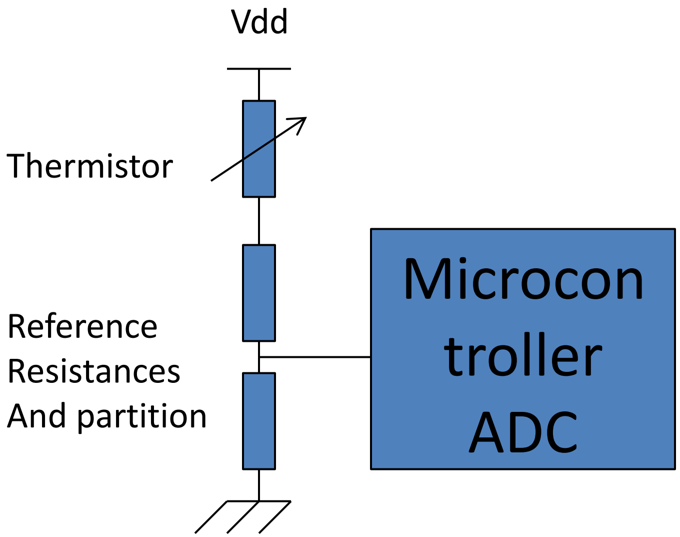
Analog implementation.

- To read outdoor temperature NTC is the best fitting solution
- Datasheet information:
  - Nominal value at 25°C: 20Kohm
  - Non linear characteristic
- Microcontroller ADC
  - Resolution 12bit
  - 250 ksps
  - Voltage reference 2.4V
  - Voltage supply 3.3V
- Conditioning circuit?



Sensors data acquisition

- Conditioning circuit:
  - The resistance value has to be converted in a voltage signal
  - Using a reference resistance it is possible to convert resistance value in a voltage signal
  - Vdd is higher than voltage reference and for small thermistor resistance it could be not possible for the ADC read the voltage value



  - Then splitting the reference resistance will make a voltage divider able to provide the output voltage thermistor value in the microcontroller ADC range
- To derive the temperature the value read from ADC has to be processed:
  - Using a formula
  - Using a lookup table

Analog vs Digital

Analog (thermistor)

- Neel to be calibrated
- Resolution is affected by reference resistance precision
- Conditioning circuit has to be accurately designed
- Use of microcontroller ADC
- Low cost

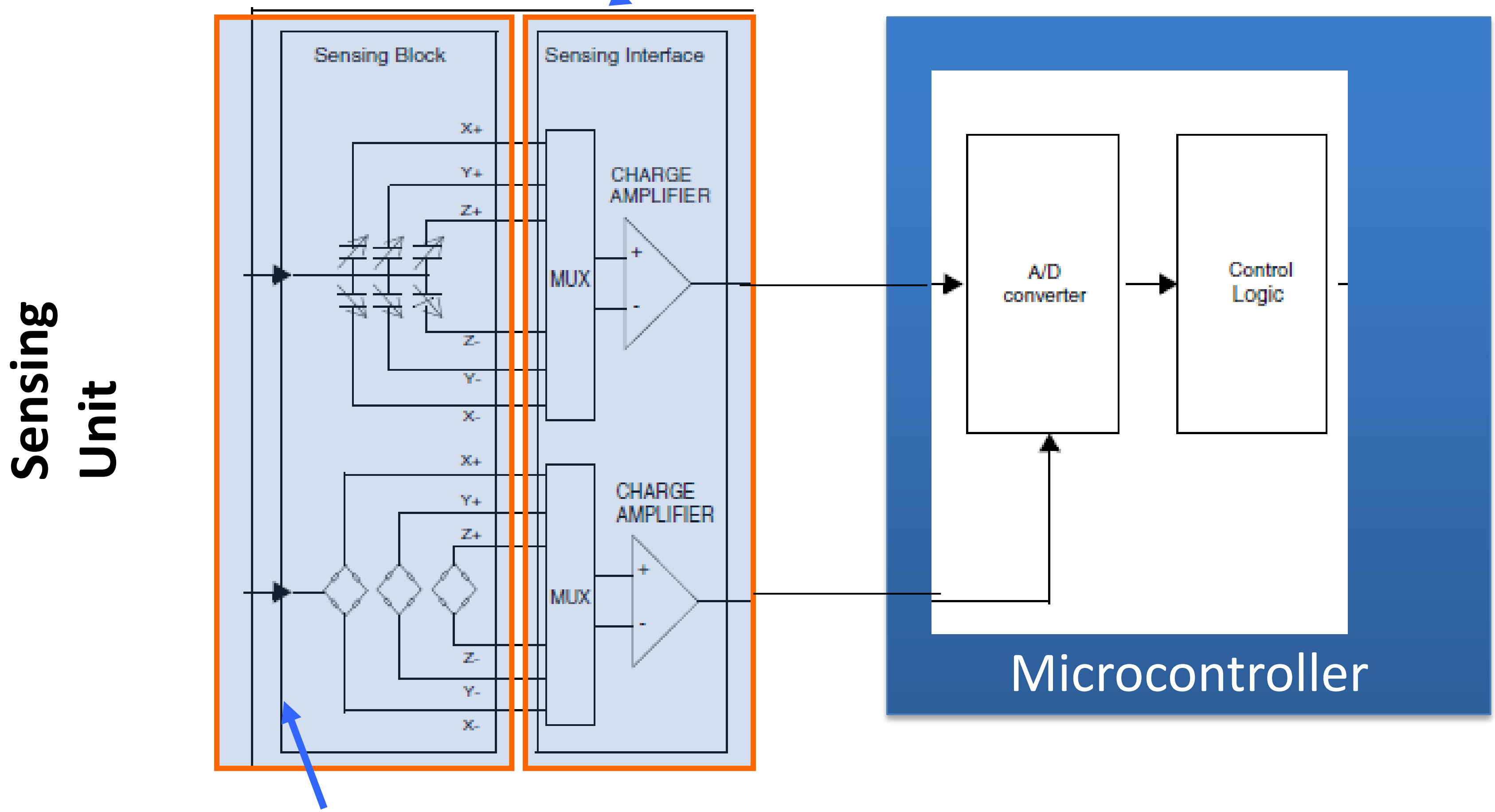
Digital (IC temperature)

- Factory calibrated
- Fixed resolution and high accuracy
- Expensive

Analog Sensors

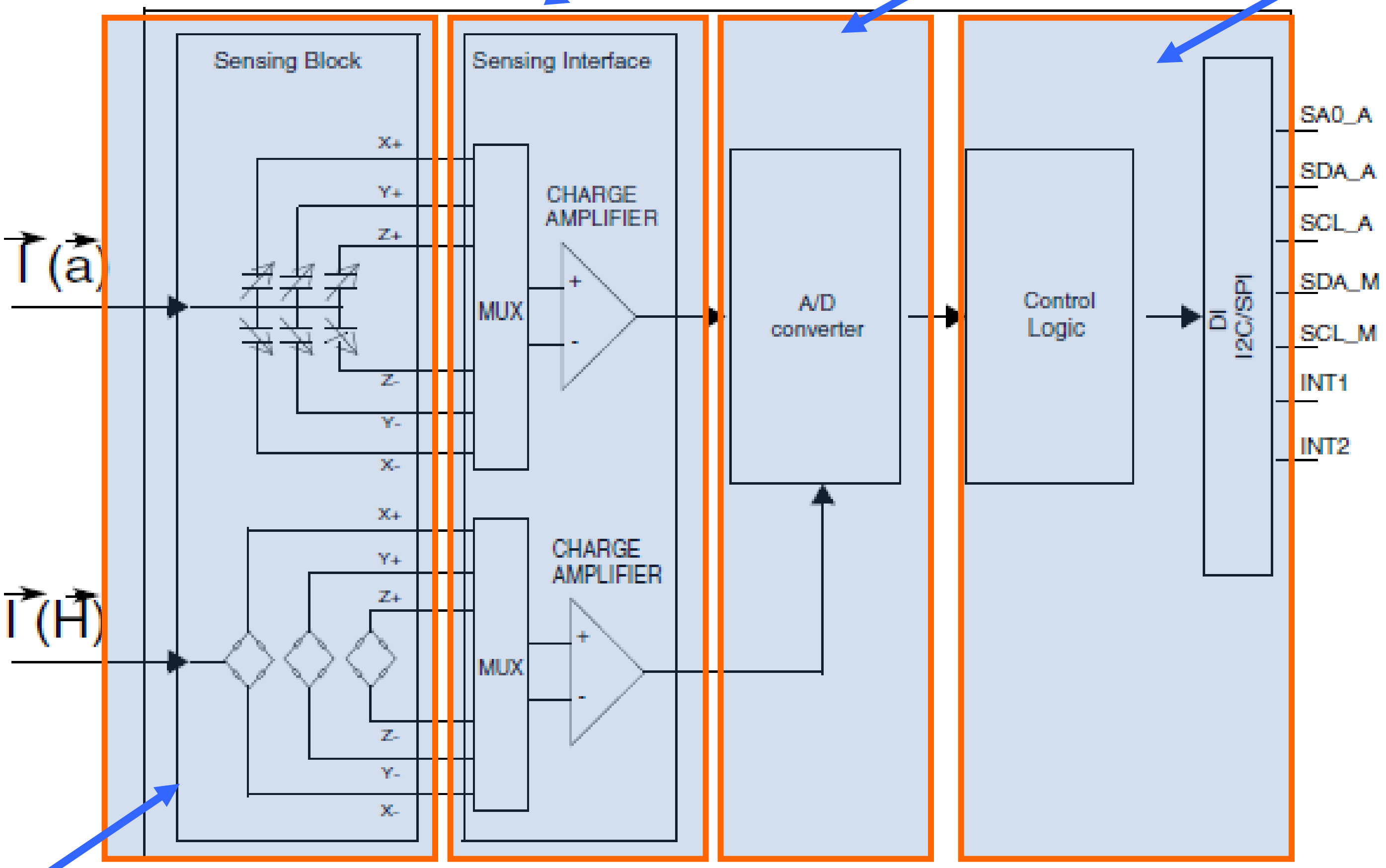
Signal Conditioning

Sensing Unit



Sensing Unit

Digital Sensors



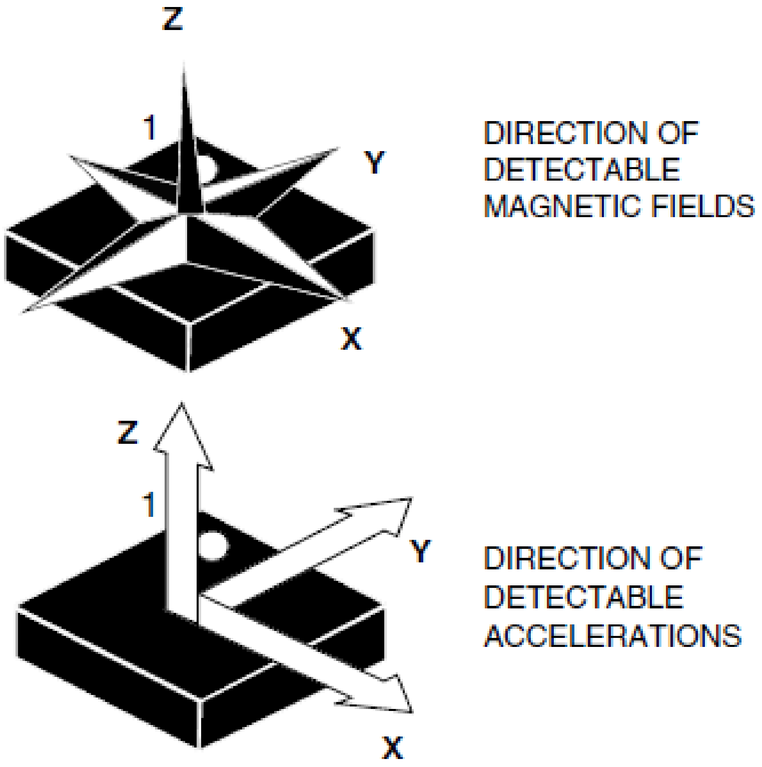
Signal Conditioning

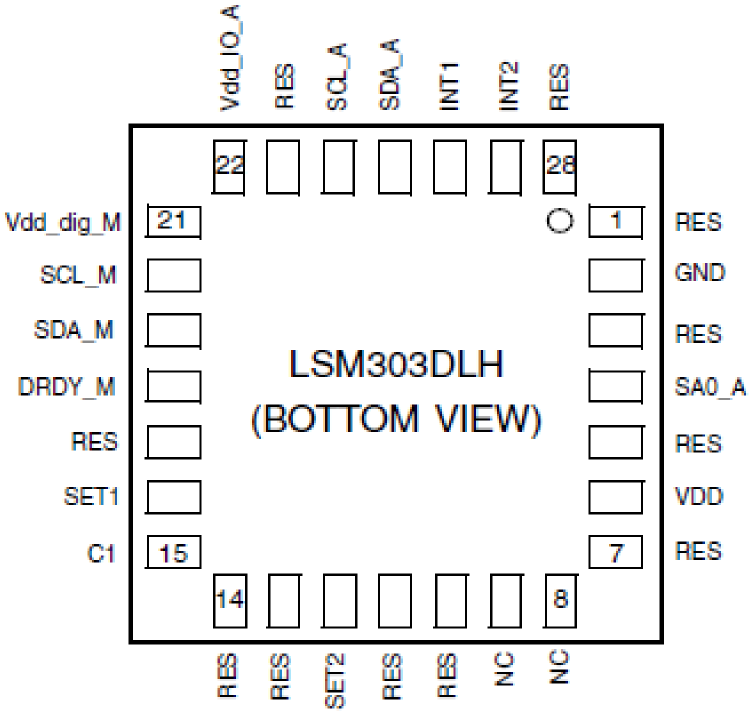
AD Converter

Digital Interface

Sensing Unit

Multi Sensors


Z
1
Y
X
DIRECTION OF DETECTABLE MAGNETIC FIELDS
Z
1
Y
X
DIRECTION OF DETECTABLE ACCELERATIONS



Multi-sensor (and functions) in one package : reduction of costs and size

Sensors (example from Perhl project)

	Axis	Range	Sensibility	Sampling Required for Perhl	Vcc (V)	Model	Price
Accelerometer	X,Y,Z	± 8g ± 4g ± 2g	3,9 mg 2 mg 1 mg	100Hz	2,5 – 3,6	LSM303DLH	7 €
Magnetometer	X,Y,Z	± 8,1Gauss	8 mGauss	100Hz	2,5 – 3,6 1,8		
Giroscope	X,Y,Z	±250 dps ±500 dps ±2000 dps	8.75 mdps/digit 17.50 mdps/digit 70 mdps/digit	100Hz	2,4 – 3,6	L3G4200	5 €
Barometer	-	130KPa	2.5Pa 1.2Pa 0.3Pa	10 Hz 5 Hz 1.25 Hz	2,7 – 3,3	XP6000CA	25 €

Sampling frequency is influenced by signal to noise ratio

Accuracy has a cost (e.g. barometer)
The package also.

Sensors data acquisition example

Case: read outdoor temperature

Commercial temperature sensors:

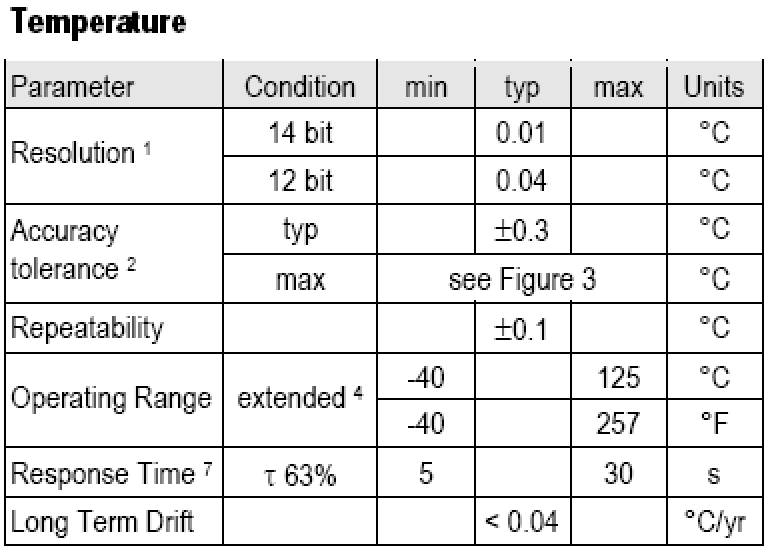
- Analog (thermistors, change resistance accordingly to the temperature):
  - NTC (negative temperature constant): R decrease hundreds of ohm each temp degree rise (-40 to 90° range), not linear, cheap
  - PTC (positive temperature constant): R increase hundreds of ohm each temp degree rise (80 to 160° range), not linear, cheap
  - PT100 (thermocouple): R increase few ohms each temp degree rise (-50 to 500°), linear, cheap

- IC temperature sensor
  - The temperature is provided to I2C or SPI data bus.
  - The microcontroller has to set and read internal sensor registers to acquire the temperature data.
  - Expensive, high resolution and accuracy

Sensors data acquisition example

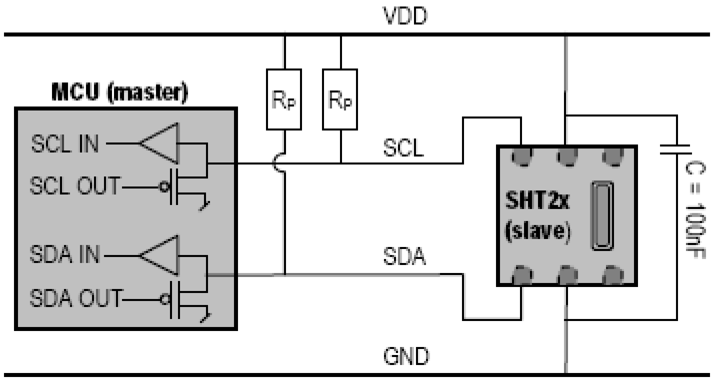
Realization with digital sensor:

- E.g. Sensirion SHT21:
  - Fully calibrated
  - Digital output,I2C interface
  - Low power consumption
  - Excellent long term stability
- The sensor is connected to the Microcontroller by I2C bus



Temperature

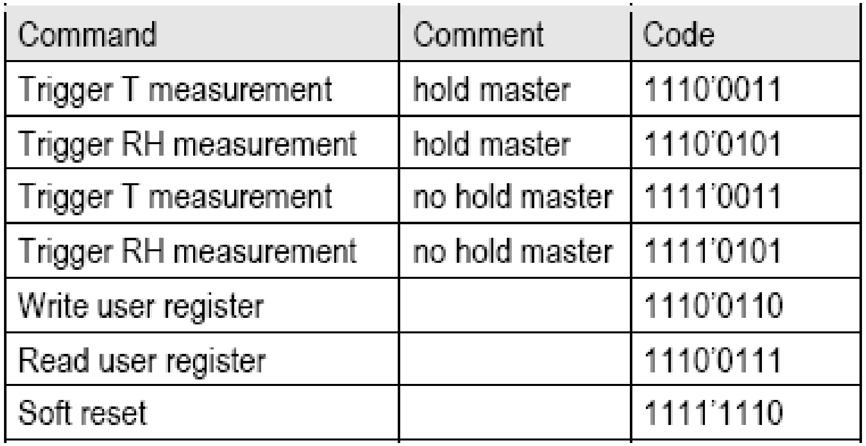
Parameter	Condition	min	typ	max	Units
Resolution [1]	14 bit		0.01		°C
	12 bit		0.04		°C
Accuracy tolerance [2]	typ		±0.3		°C
	max	see Figure 3			°C
Repeatability			±0.1		°C
Operating Range	extended [4]	-40		125	°C
		-40		257	°F
Response Time [7]	τ 63%	5		30	s
Long Term Drift			< 0.04		°C/yr



Sensors data acquisition example

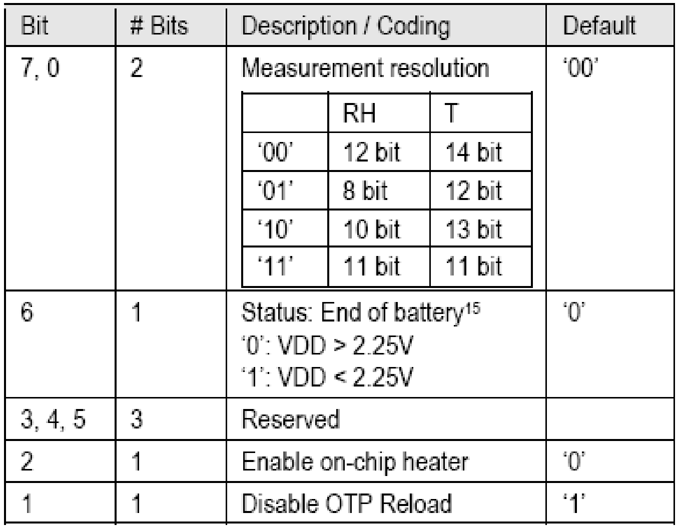
Realization with digital sensor:

- The SHT21 has internal register that microcontroller can read and/or write by I2C bus
  Those registers does:
  - Set the sensor measurement precisio
  - Start data acquisition
  - Read acquired value
  - Switch of the sensor



Command	Comment	Code
Trigger T measurement	hold master	1110’0011
Trigger RH measurement	hold master	1110’0101
Trigger T measurement	no hold master	1111’0011
Trigger RH measurement	no hold master	1111’0101
Write user register		1110’0110
Read user register		1110’0111
Soft reset		1111’1110

Table 6 Basic command set, RH stands for relative humidity, and T stands for temperature



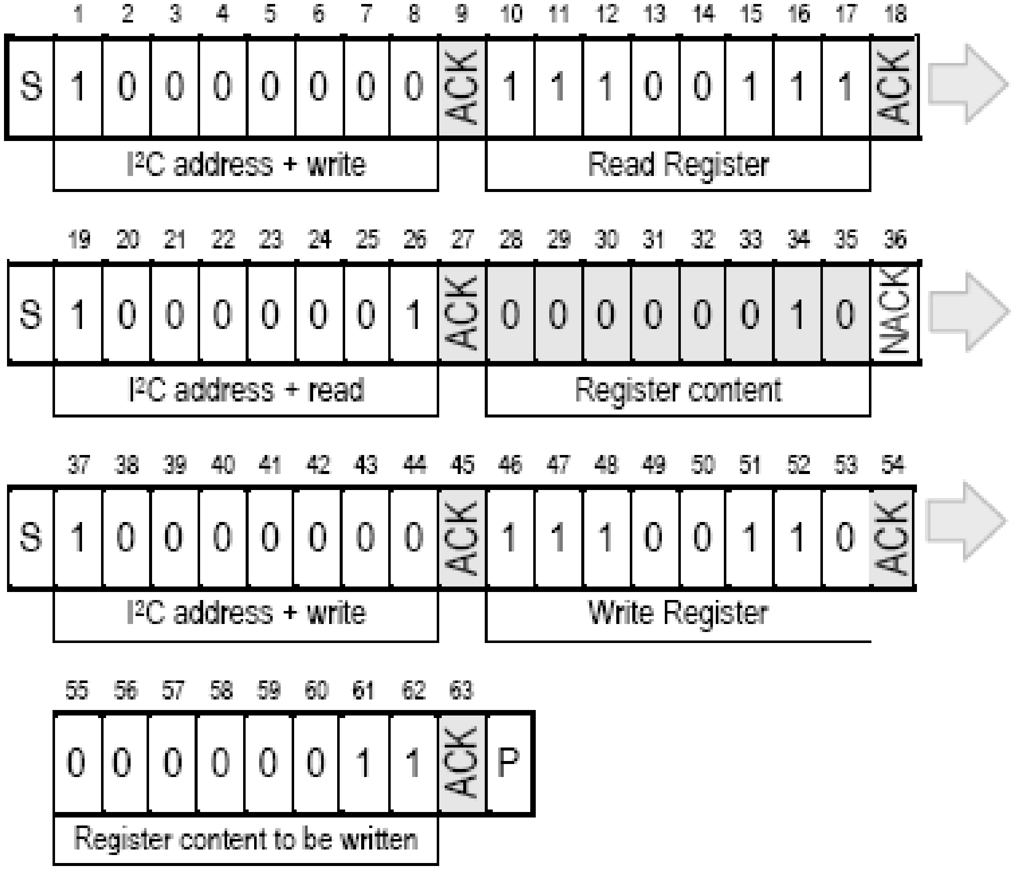
Bit	# Bits	Description / Coding	Default
7, 0	2	Measurement resolution RH / T ‘00’: 12 bit / 14 bit ‘01’: 8 bit / 12 bit ‘10’: 10 bit / 13 bit ‘11’: 11 bit / 11 bit	‘00’
6	1	Status: End of battery[15] ‘0’: VDD > 2.25V ‘1’: VDD < 2.25V	‘0’
3, 4, 5	3	Reserved	
2	1	Enable on-chip heater	‘0’
1	1	Disable OTP Reload	‘1’

Table: user register

Sensors data acquisition example

Realization with digital sensor:

- How to write user register:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	1	0	0	0	0	0	0	0	ACK	1	1	1	0	0	1	1	1	ACK
	I^{2}C address + write									Read Register								

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
S	1	0	0	0	0	0	0	1	ACK	0	0	0	0	0	0	1	0	NACK
	I^{2}C address + read									Register content								

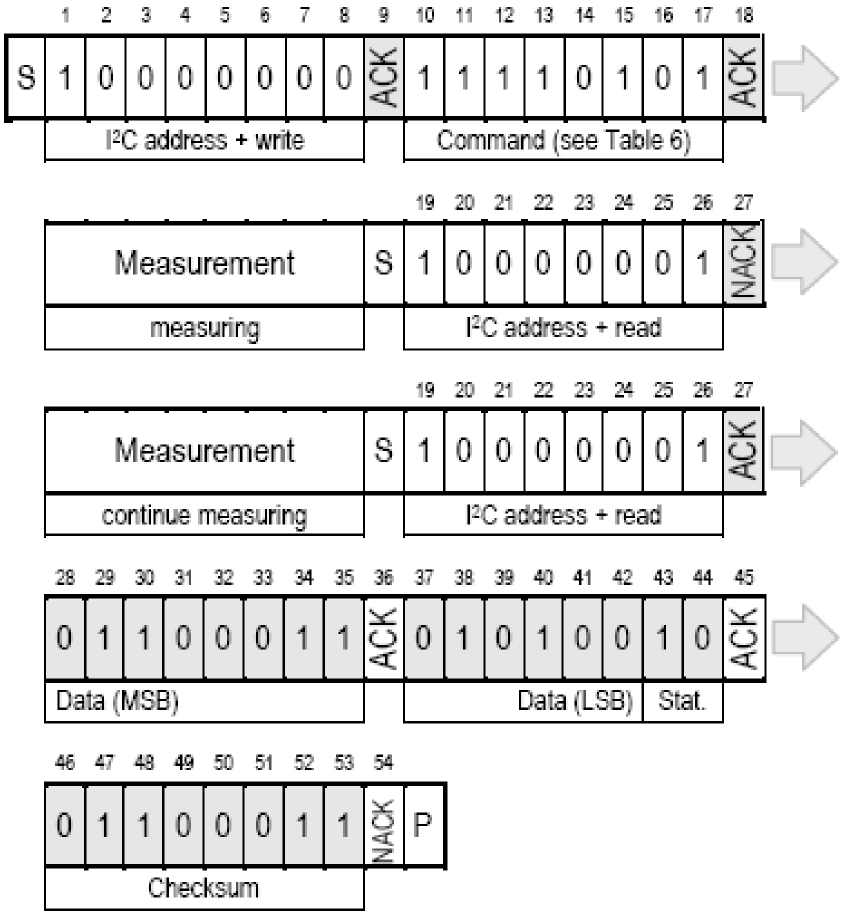
	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
S	1	0	0	0	0	0	0	0	ACK	1	1	1	0	0	1	1	0	ACK
	I^{2}C address + write									Write Register								

55	56	57	58	59	60	61	62	63	
0	0	0	0	0	0	1	1	ACK	P
Register content to be written									

Sensors data acquisition example

Realization with digital sensor:

- Data acquisition procedure:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	1	0	0	0	0	0	0	0	ACK	1	1	1	1	0	1	0	1	ACK
	I^2C address + write									Command (see Table 6)								

		19	20	21	22	23	24	25	26	27
Measurement	S	1	0	0	0	0	0	0	1	NACK
measuring		I^2C address + read								

		19	20	21	22	23	24	25	26	27
Measurement	S	1	0	0	0	0	0	0	1	ACK
continue measuring		I^2C address + read								

28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
0	1	1	0	0	0	1	1	ACK	0	1	0	1	0	0	1	0	ACK
Data (MSB)									Data (LSB)						Stat.		

46	47	48	49	50	51	52	53	54	
0	1	1	0	0	0	1	1	NACK	P
Checksum									