Sensors & Buses

ELEC-D0301 Protopaja

Aalto University
School of Electrical
Engineering

Aleksi Zubkovski
(Based on slides by Juha Biström & Mikko Simenius)

12.6.2023

Understand how sensors “sense”

Understand limitations & accuracy of sensors

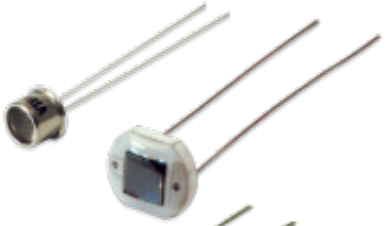
Sensor integration

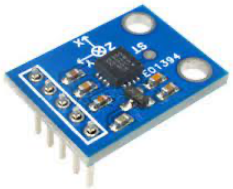
Buses & protocols

Aalto University
School of Electrical
Engineering

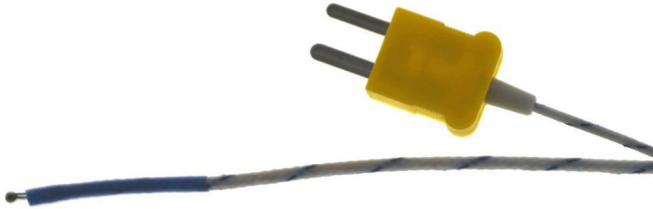
Sensors

- **Light, Colour**
  - LDR (photoresistor), Phtodiodes, complex semiconductors
- **Sound**
  - Microphone, Piezo
- **Acceleration**

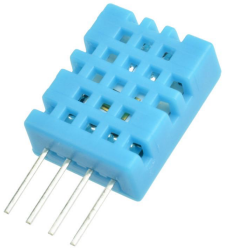


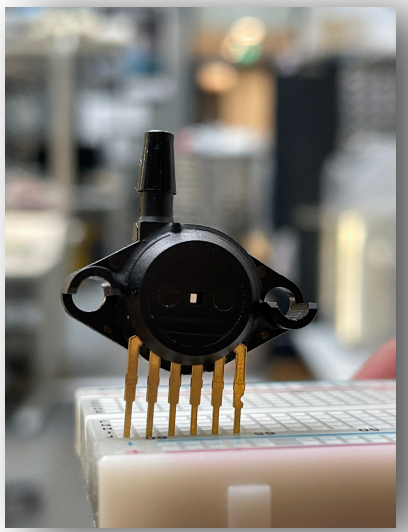


Sensors



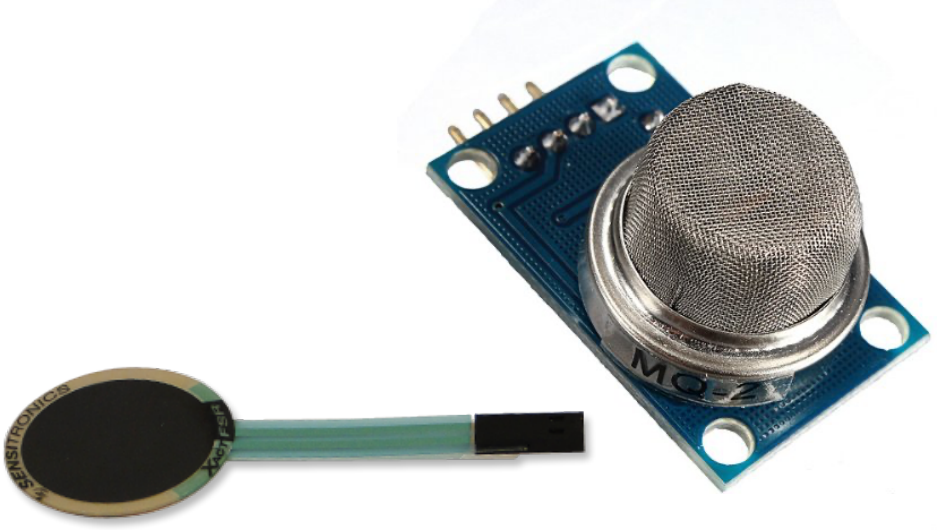
- **Temperature**
  - Thermistor, Thermocouple, PT100, LM35, DS18x20…
- **Humidity, Moisture**
  - E.G. DHT11
- **Pressure**
  - E.G. MPX5100AP

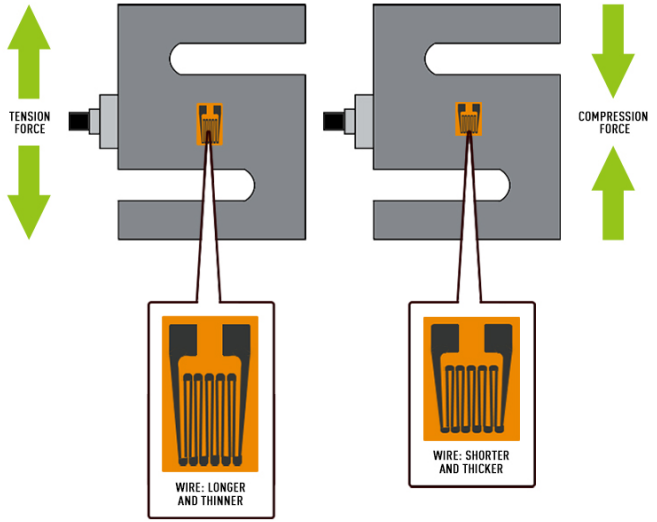




Sensors

- **Gas Contents**
  - For example MQx for different gases
  - MQ3 - for ethanol
- **Force Sensors**
  - Force Sensing Resistor (Sensing Stretch)
  - Load Cell
- **Magnetic Field**
  - E.G. HMC5883





Sensor selection?

< Which sensor is good?

Aalto University
School of Electrical
Engineering

Compare	Mfr Part #	Quantity Available	Price	Series	Package	Product Status
☐	SDP31-500PA-TR-1500PCS SENSOR 0.07PSID 0.08" 16BIT 1SMD *Sensirion AG*	22 529 In Stock	1 : €19.52000 Cut Tape (CT) 1 500 : €19.52472 Tape & Reel (TR)	SDP3x	Tape & Reel (TR) Cut Tape (CT)	Active
☐	SDP31-500PA-TR-250PCS SENSOR 0.07PSID 0.08" 16BIT 1SMD *Sensirion AG*	1 691 In Stock	1 : €19.52000 Cut Tape (CT) 250 : €21.28632 Tape & Reel (TR)	SDP3x	Tape & Reel (TR) Cut Tape (CT) Digi-Reel®	Active
☐	SDP36-500PA-TR-250PCS SENSOR 0.07PSID 0.08" .9V 16SMD *Sensirion AG*	843 In Stock	1 : €19.52000 Cut Tape (CT) 250 : €21.28632 Tape & Reel (TR)	SDP3x	Tape & Reel (TR) Cut Tape (CT) Digi-Reel®	Active
☐	SDP32-125PA-TR-250PCS SENSOR 0.018PSID 0.08" 16BIT SMD *Sensirion AG*	1 170 In Stock	1 : €19.52000 Cut Tape (CT) 250 : €21.28632 Tape & Reel (TR)	SDP3x	Tape & Reel (TR) Cut Tape (CT) Digi-Reel®	Active
☐	SDP33-1500PA-TR-250PCS SENSOR 0.22PSID 0.08" 16BIT 1SMD *Sensirion AG*	1 278 In Stock	1 : €19.52000 Cut Tape (CT) 250 : €21.28632 Tape & Reel (TR)	SDP3x	Tape & Reel (TR) Cut Tape (CT) Digi-Reel®	Active
☐	SDP816-125PA SENSOR 0.018PSID 0.2" 4.5V *Sensirion AG*	3 407 In Stock	1 : €19.52000 Tray	SDP800	Tray	Active
☐	SDP810-500PA SENSOR 0.07PSID 0.2" 16BIT *Sensirion AG*	2 531 In Stock	1 : €19.52000 Tray	SDP800	Tray	Active
☐	SDP810-125PA SENSOR 0.018PSID 0.2" 16BIT *Sensirion AG*	1 464 In Stock	1 : €19.52000 Tray	SDP800	Tray	Active
☐	SDP816-500PA SENSOR 0.07PSID 0.2" 4.5V *Sensirion AG*	762 In Stock	1 : €19.52000 Tray	SDP800	Tray	Active

Datasheet

Aalto University
School of Electrical
Engineering

NXP Semiconductors
Data Sheet: Technical Data

Document Number: MPX5100
Rev. 14, 12/2018

RoHS

MPX5100, 0 to 100 kPa, Differential, Gauge, and Absolute, Integrated, Pressure Sensors

MPX5100

The MPX5100 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high-level, analog output signal that is proportional to the applied pressure.

Features

- 2.5% maximum error over 0 to 85 °C
- Ideally suited for microprocessor or microcontroller-based systems
- Patented silicon shear stress strain gauge
- Available in absolute, differential and gauge configuration
- Durable epoxy unibody element
- Easy-to-use chip carrier option

Typical applications

- Patient monitoring
- Process control
- Pump/motor control
- Pressure switching
- White goods

Unibody packages

MPX5100AP/GP
98ASB42796B

MPX5100DP
98ASA42797B

Small outline packages

MPXV5100DP
98ASA99255D

MPXV5100GP
98ASA99303D

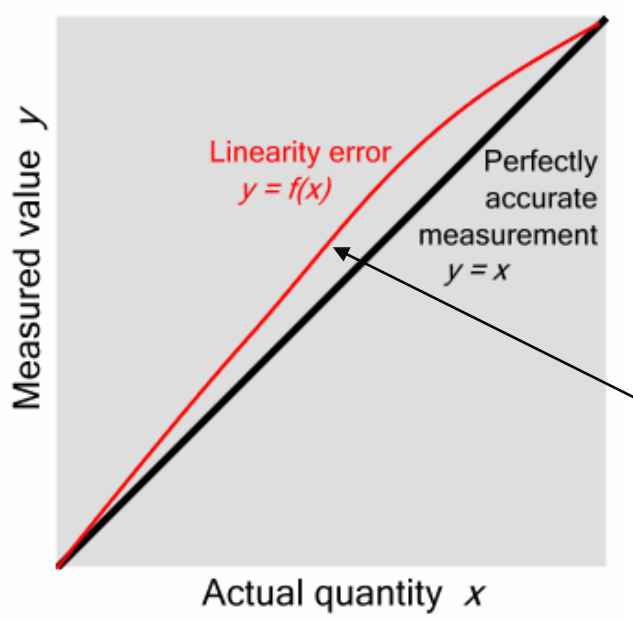
MPXV5100GC6U
98ASB17757C

MPXV5100GC7U
98ASB17759C

Ordering information

Device name	Shipping	Package	# of Ports			Pressure type			Device marking
			None	Single	Dual	Gauge	Differential	Absolute	
Unibody Package (MPX5100 Series)									
MPX5100AP	Tray	98ASB42796B		•				•	MPX5100AP
MPX5100DP	Tray	98ASB42797B			•		•		MPX5100DP
MPX5100GP	Tray	98ASB42796B		•		•			MPX5100GP
Small Outline Package (MPXV5100 Series)									
MPXV5100DP	Tray	98ASA99255D			•		•		MPXV5100DP
MPXV5100GC6U	Rail	98ASB17757C		•		•			MPXV5100G
MPXV5100GC7U	Rail	98ASB17759C		•		•			MPXV5100G
MPXV5100GP	Tray	98ASA99303D		•		•			MPXV5100GP

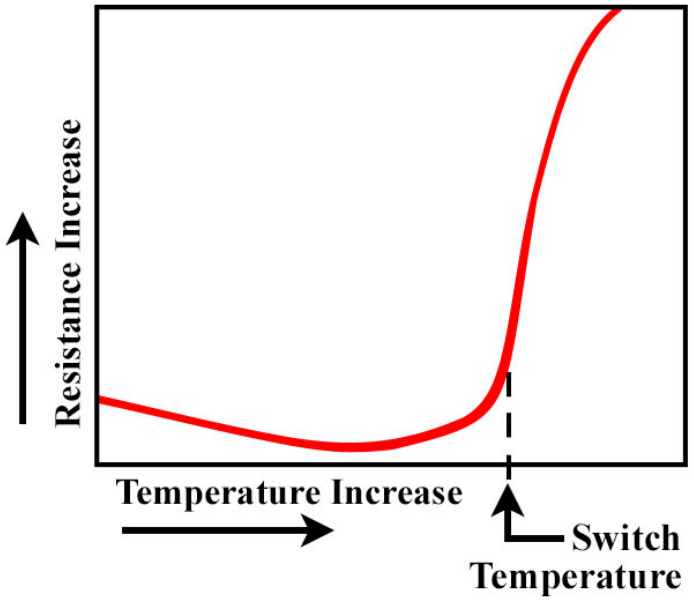
Sensor Characteristics: Linearity



Non-linearity

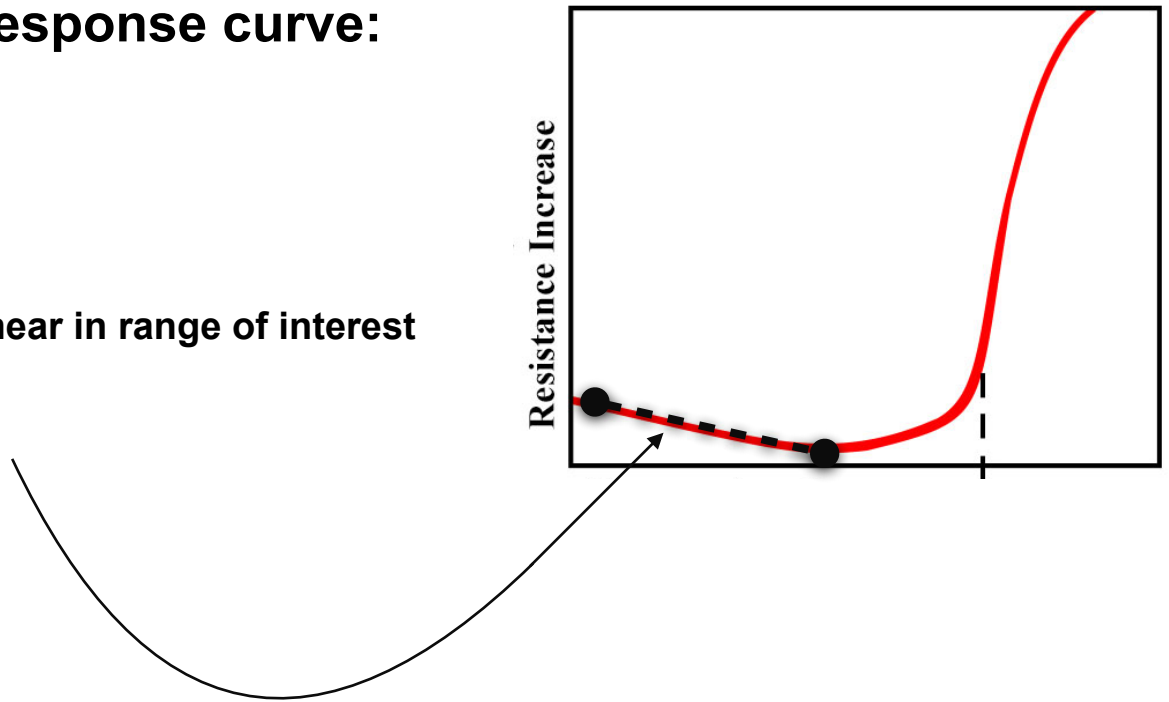
Sensor Characteristics: Linearity

E.G.Thermistor response curve:



Sensor Characteristics: Linearity

E.G.Thermistor response curve:



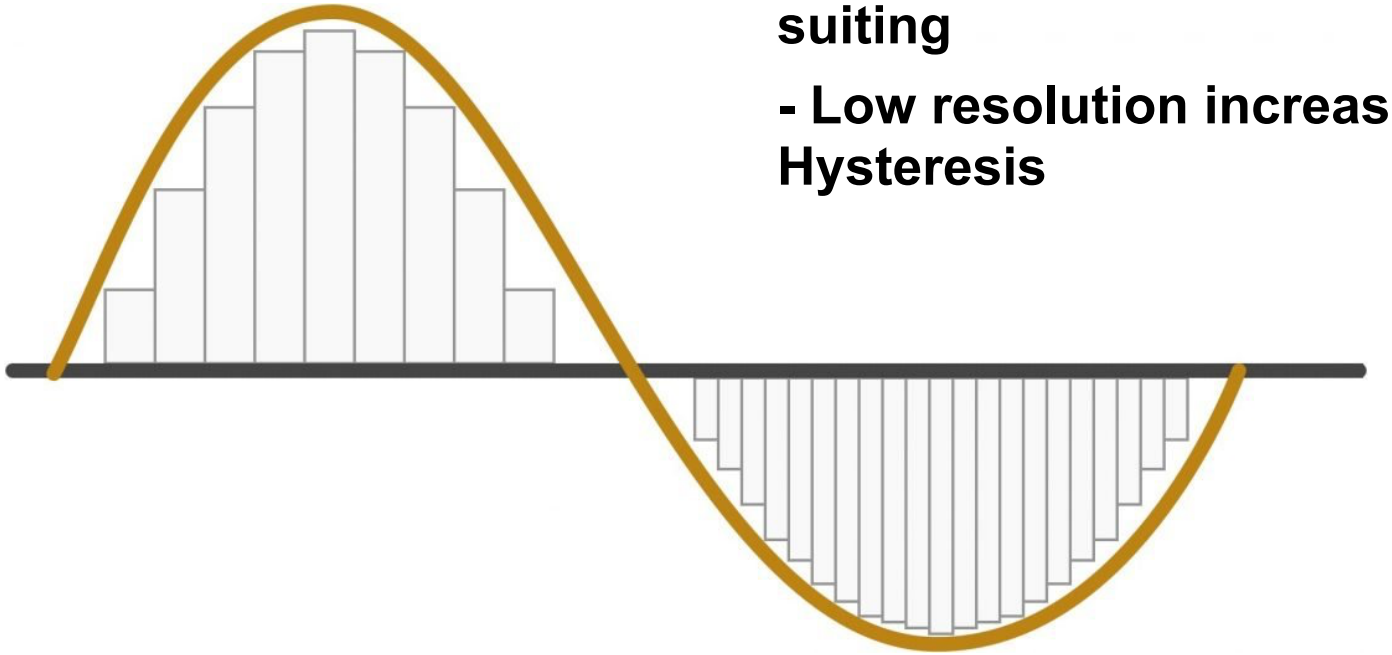
Chose sensors that are linear in range of interest

Sensor Characteristics: Stability & Drift

- A sensor is stable if it is able to produce consistent measurements for constant environment
- Drift = differentiation around the constant

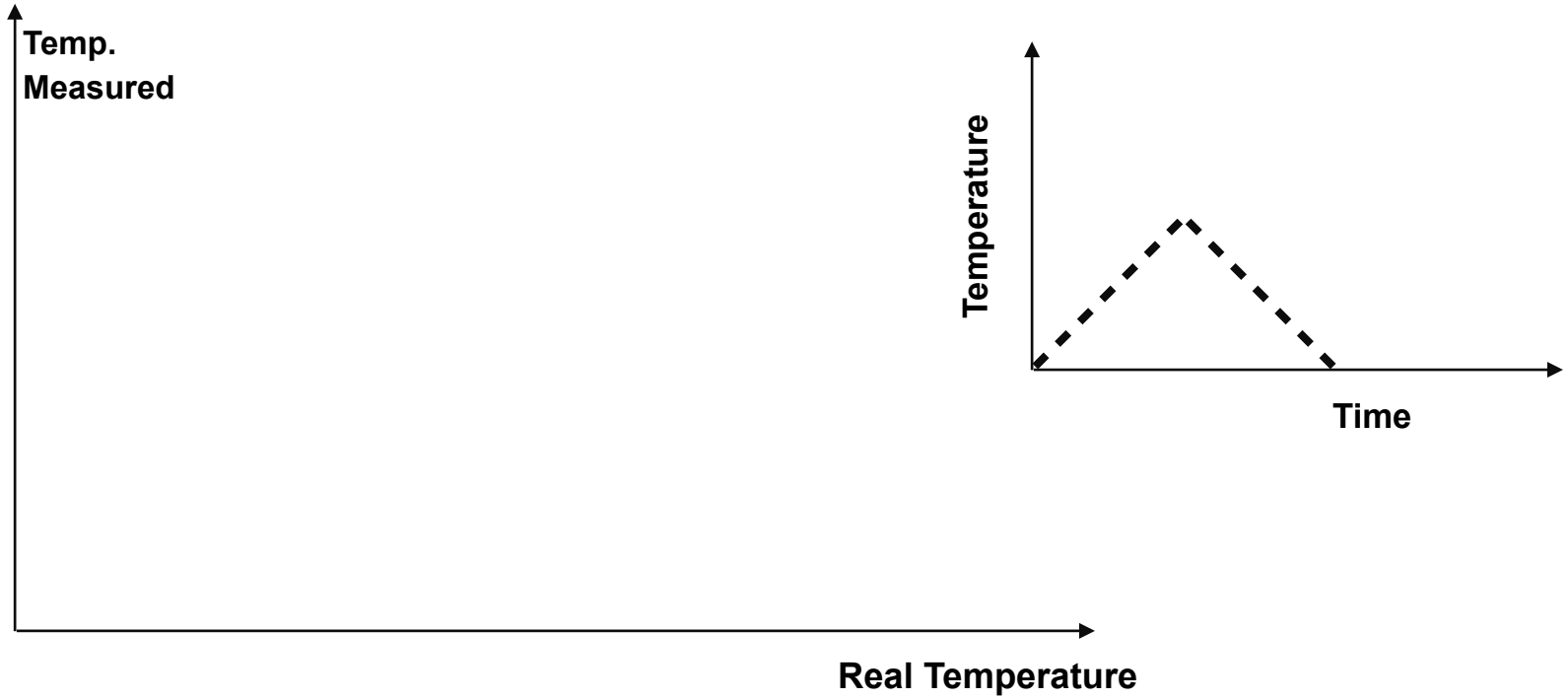
Important in process monitoring

Sensor Characteristics: Resolution

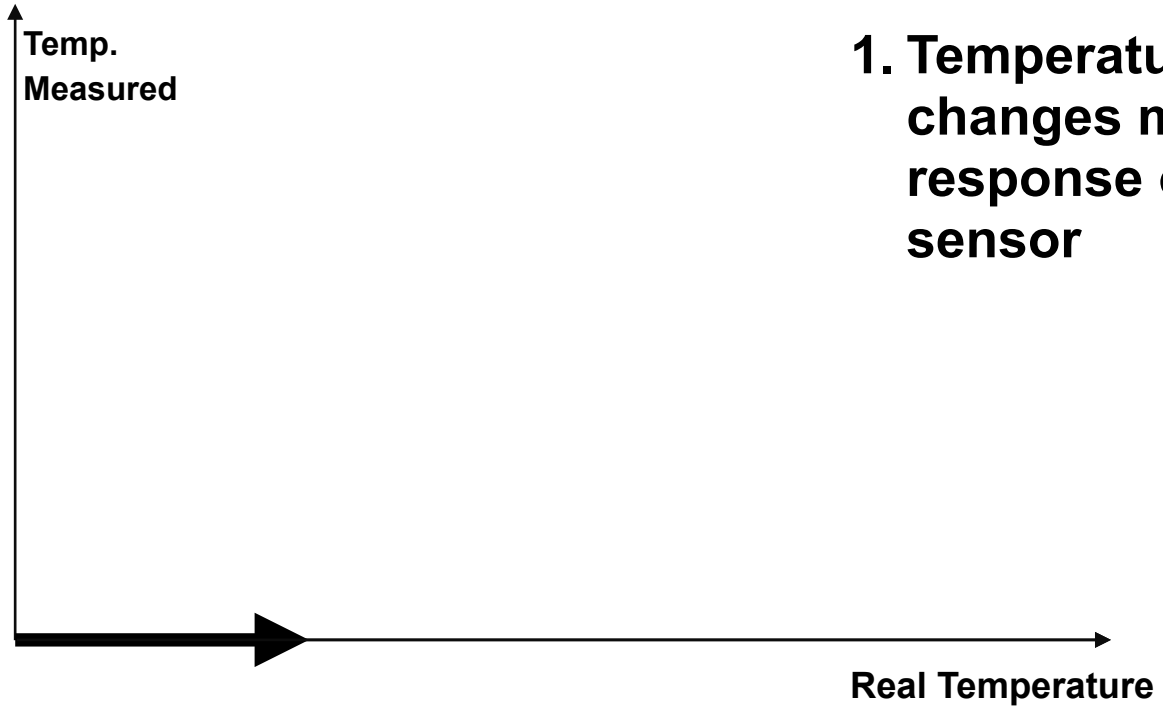


- Sensor resolution should be suiting
- Low resolution increases Hysteresis

Sensor Characteristics: Hysteresis

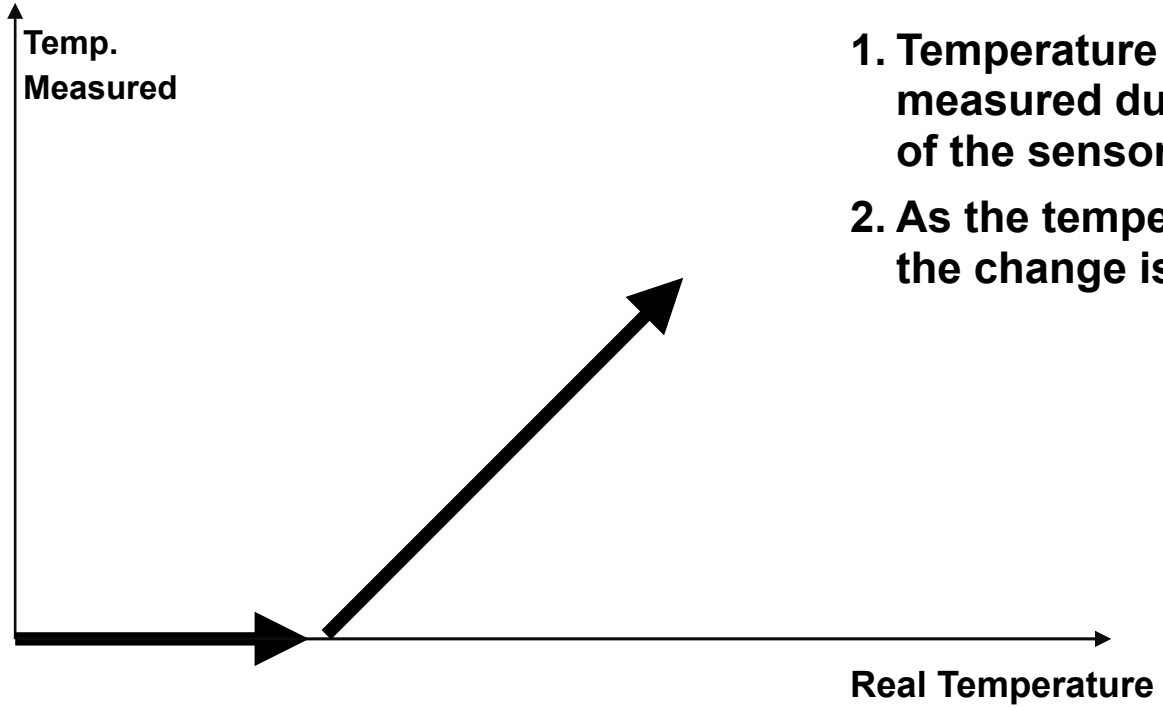


Sensor Characteristics: Hysteresis



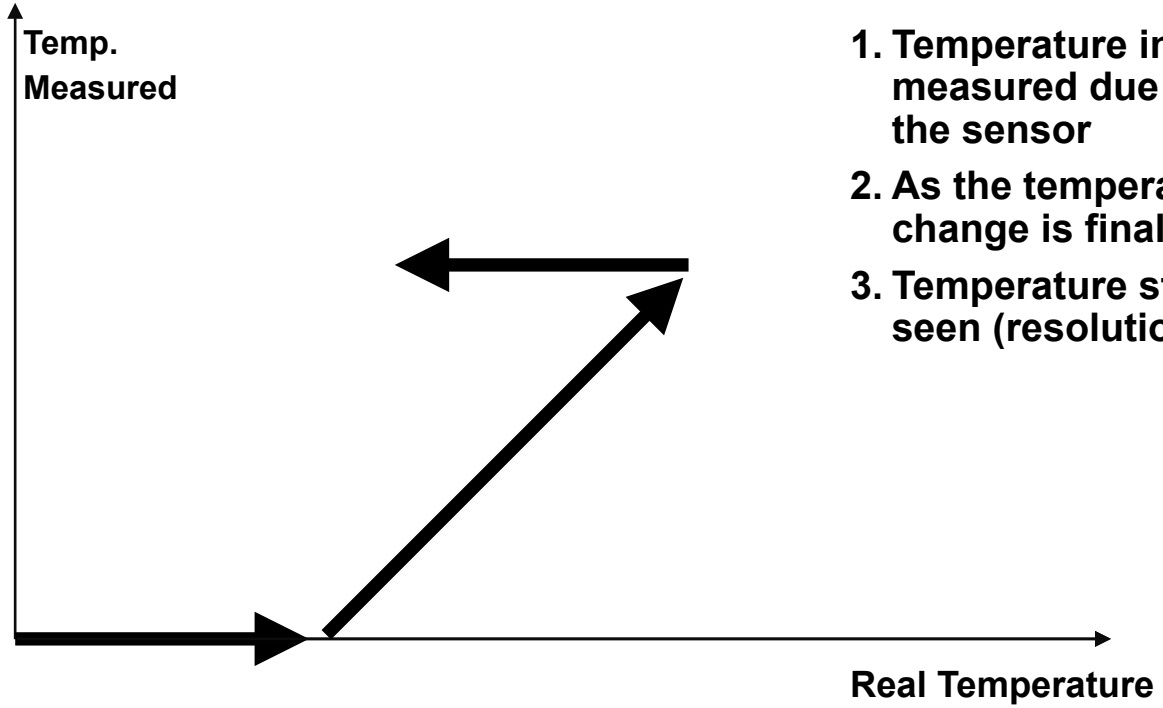
1. Temperature increases, but no changes measured due to slow response or resolution of the sensor

Sensor Characteristics: Hysteresis



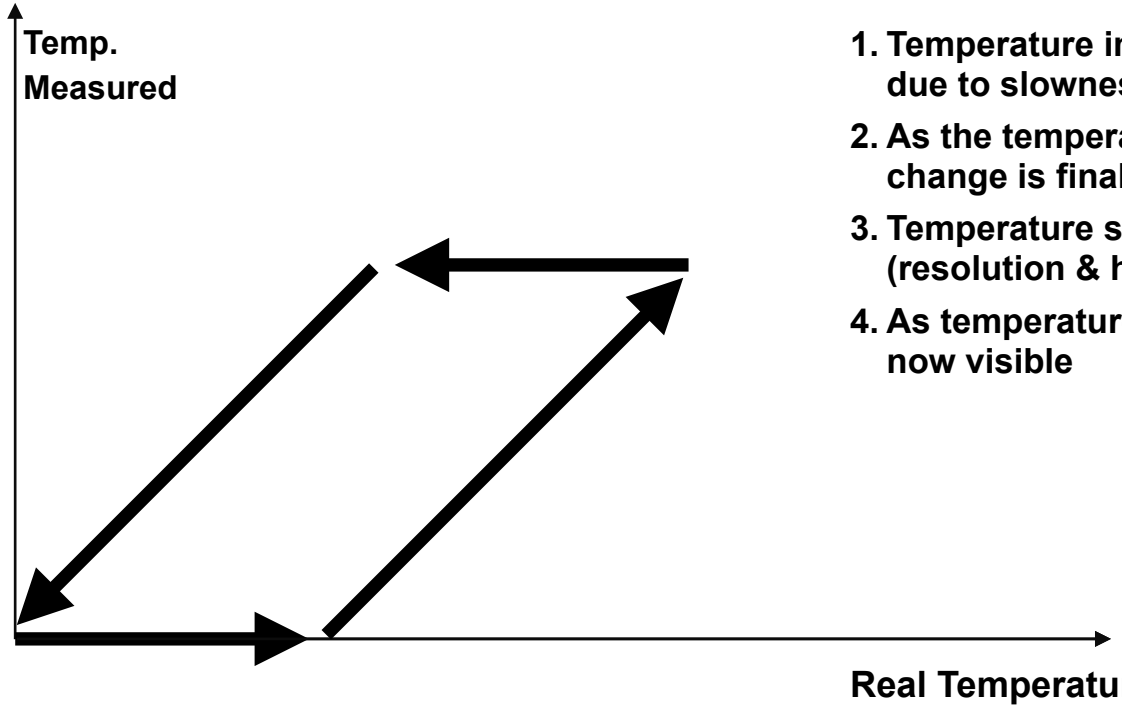
1. Temperature increases, but no changes measured due to slowness or resolution of the sensor
2. As the temperature continues increasing the change is finally seen

Sensor Characteristics: Hysteresis



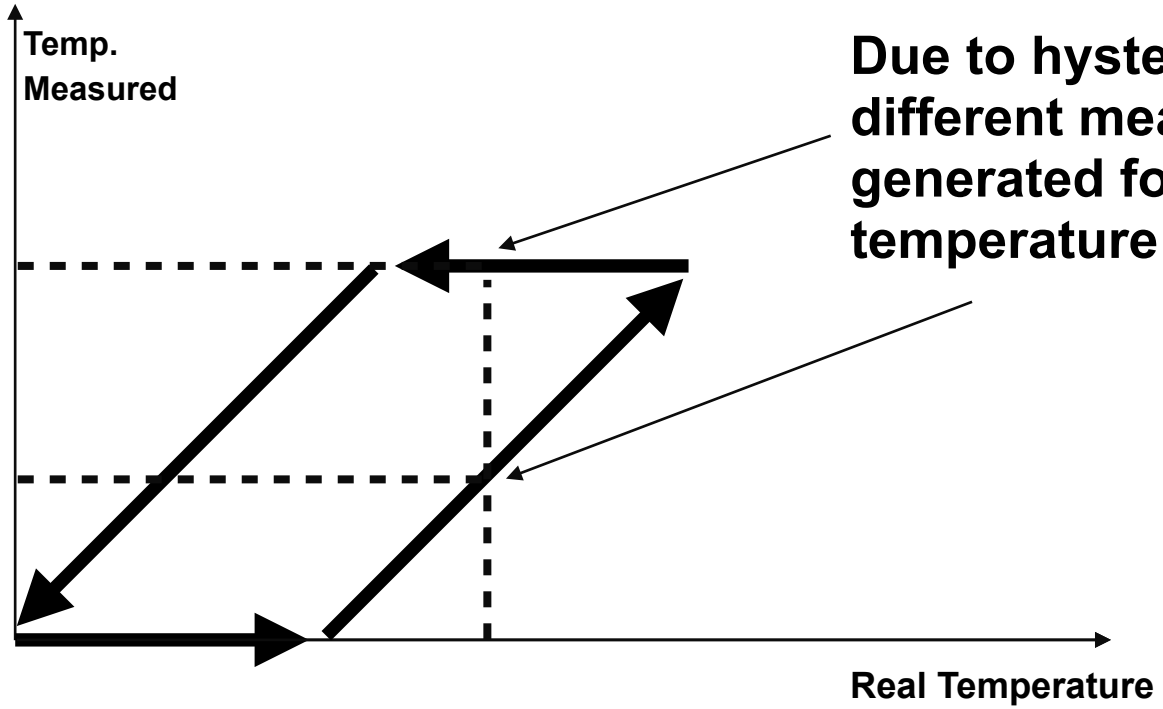
1. Temperature increases, but no changes measured due to slowness or resolution of the sensor
2. As the temperature continues increasing the change is finally seen
3. Temperature starts dropping, but no change seen (resolution & hysteresis)

Sensor Characteristics: Hysteresis



1. Temperature increases, but no changes measured due to slowness or resolution of the sensor
2. As the temperature continues increasing the change is finally seen
3. Temperature starts dropping, but no change seen (resolution & hysteresis)
4. As temperature decreases enough the change is now visible

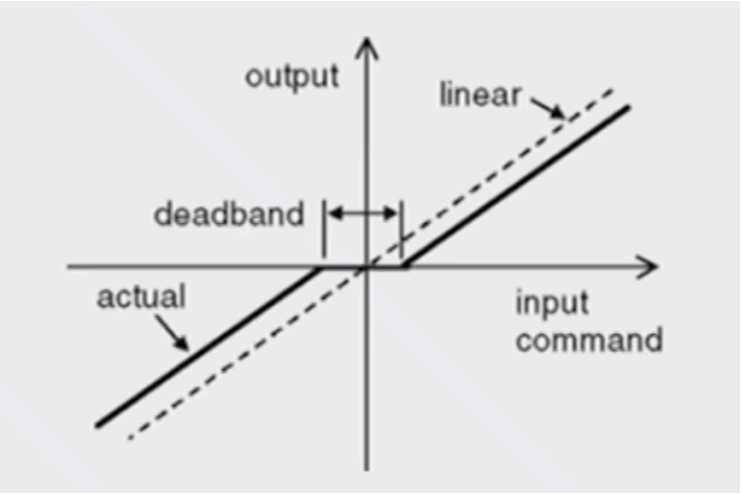
Sensor Characteristics: Hysteresis



Due to hysteresis we get two different measurements are generated for the same temperature!

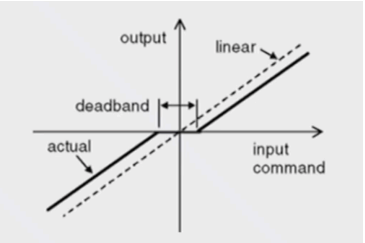
Sensor Characteristics: Dead Space

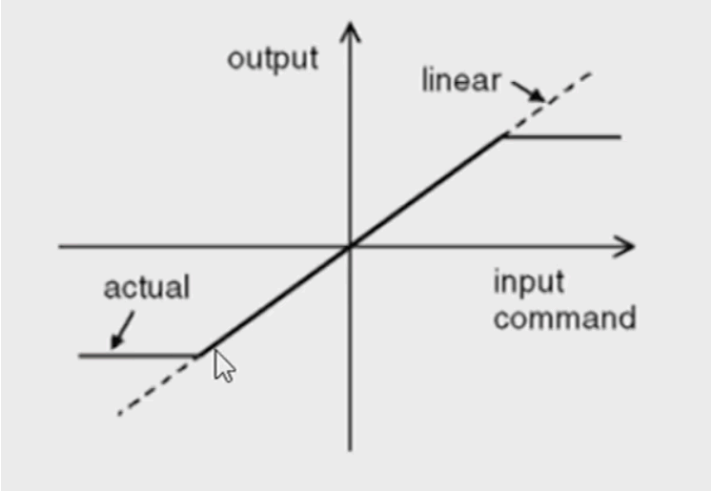
- **Some sensors often have Dead Zones (usually near zero value)**
  - *In dead zone the sensor will not respond to changes*
  - *Can be added on purpose to counter hysteresis*



Sensor Characteristics: Saturation

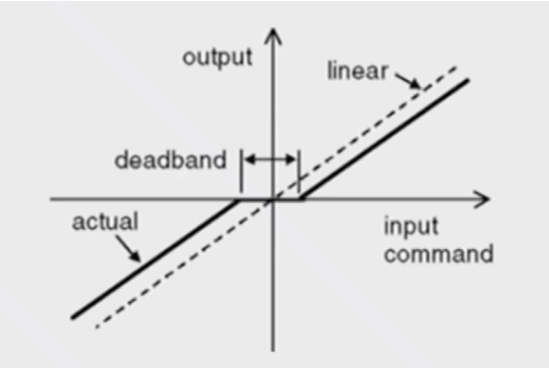
- Some sensors often have Dead Zones (usually near zero value)
- Saturation

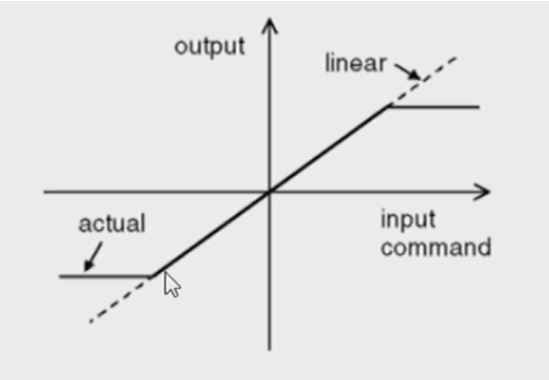




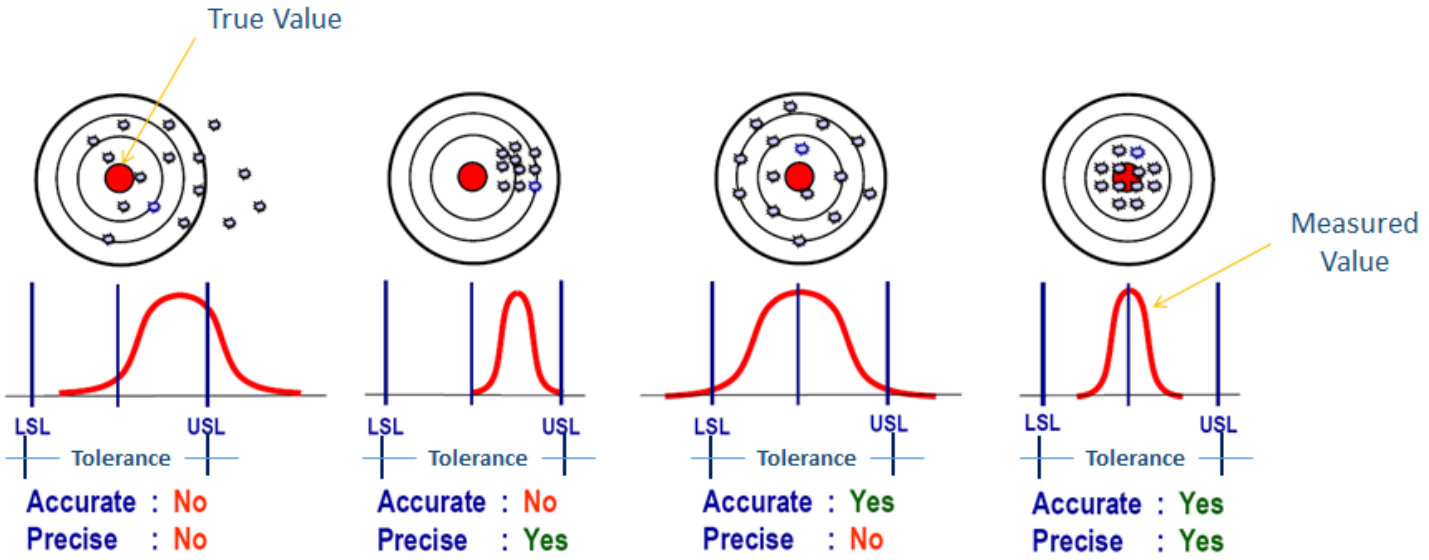
Sensor Characteristics: Data-sheet

- Some sensors often have Dead Zones (usually near zero value)
- Saturation
- Characteristics are documented in sensor data-sheet





Calibration: Accurate vs Precise



Nothing can be done.

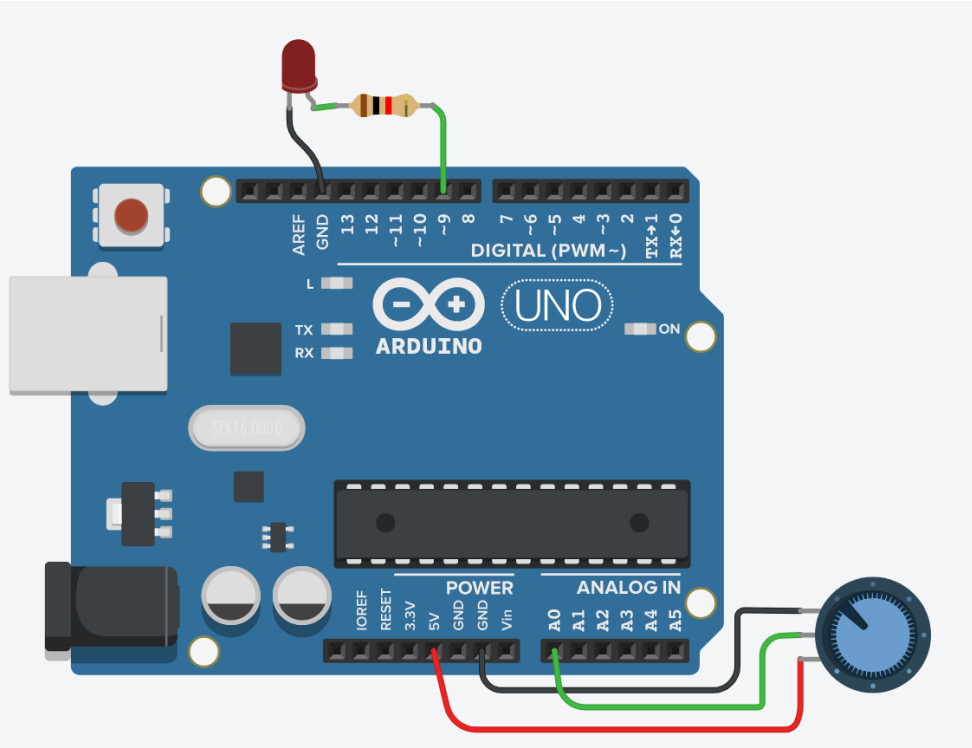
Sensor Calibration

Data Averaging

No action necessary

Analog & Digital Sensor Outputs

- Most of sensors we have discussed sensed the environment that varied its *analog signal*
- If analog sensor data is processed by digital hardware it must be converted.



Analog & Digital Sensor Outputs

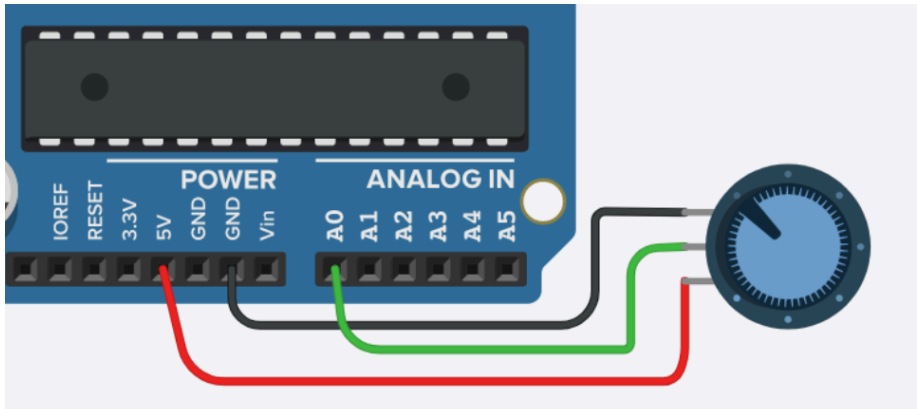
- A/D converters are used for conversion
- Many sensors have A/D converters built in (easier to use)

Analog & Digital Sensor Outputs

- A/D converters are used for conversion
- Many sensors have A/D converters built in (easier to use)
- High performance sensors usually have only analog inputs

A/D Converters

- Convert analog data to digital through quantisation
  - Leads to some data loss
- MCU's usually have builtin A/D's
  - E.g. in Arduino UNO whenever we use analog pins (*analog_read (___)*) MCU's A/D unit is used.
  - External converters can be used too
- Noise considerations:
  - Component noise
  - Layout considerations
  - ADC saturation: amplify signal before feeding it to the ADC. <u>Add resistor?</u>



Signal Sampling

- Sampling frequency:
  - How frequently samples are taken (Value Differentiation Accuracy)
    E.g. Arduino UNO: almost 1kHz
- Bit depth (“Resolution”):
  - How precise values can be measured. (Value Accuracy)
  - E.g. Arduino UNO: 10 bit (= 1024)

Too small sampling freq > Aliasing

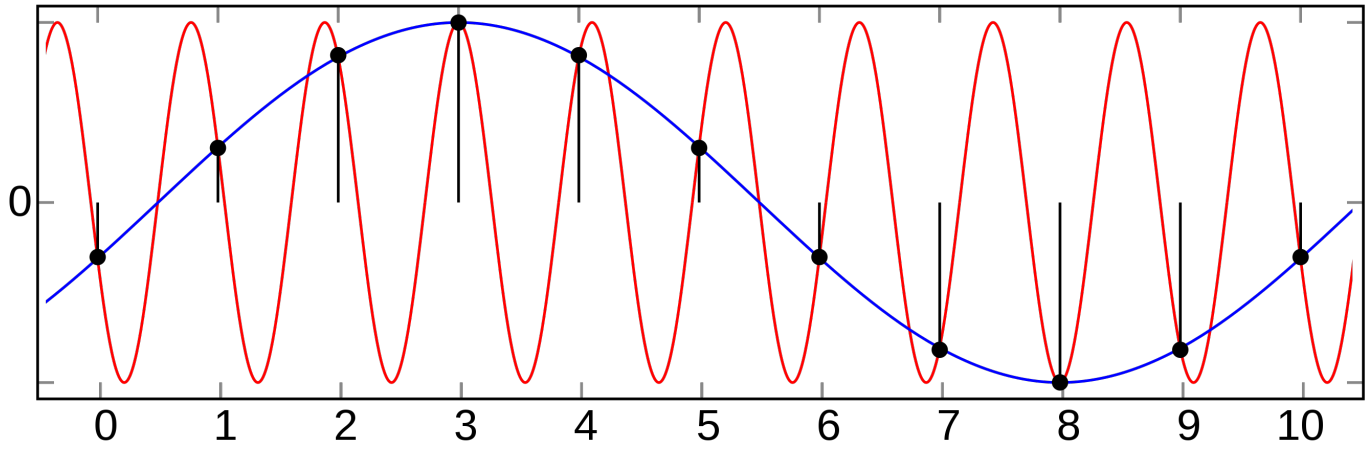
Too small bit depth > Quantization error

Signal Sampling

- **Nyquist Theorem:**
  Sampling Frequency must be higher than 2 x maximum frequency being sampled.
  - **E.g Audio freq. range is** 20 Hz — 20 kHz**,**
    Sampling frequency must be 40 kHz

Signal Sampling

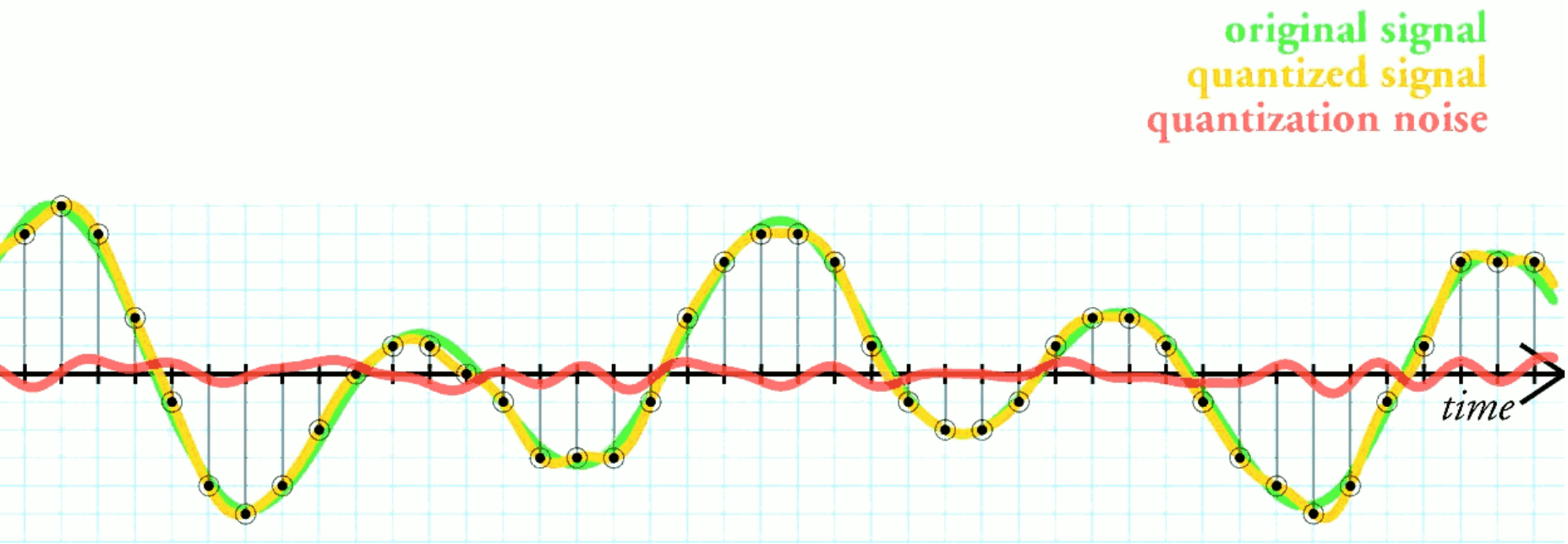
- Aliasing
  - With low sampling frequency aliasing may occur



  - Low pas filter may prevent aliasing, Aliasing filter

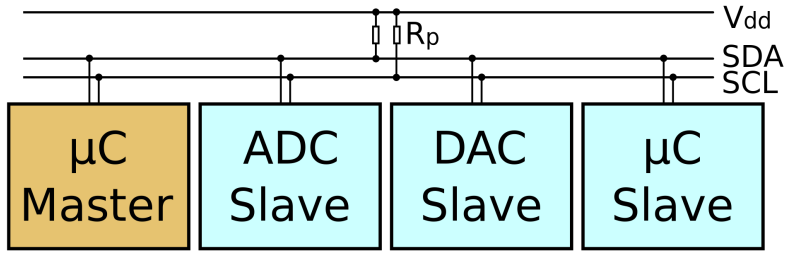
Signal Sampling

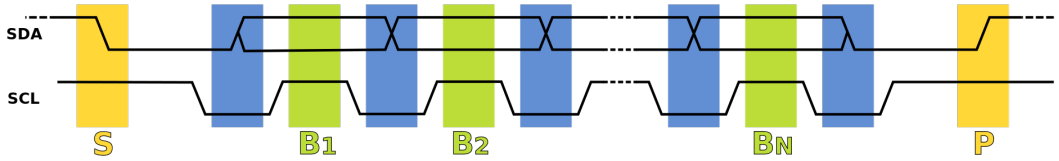
- Aliasing
  - With low sampling frequency aliasing may occur



Buses

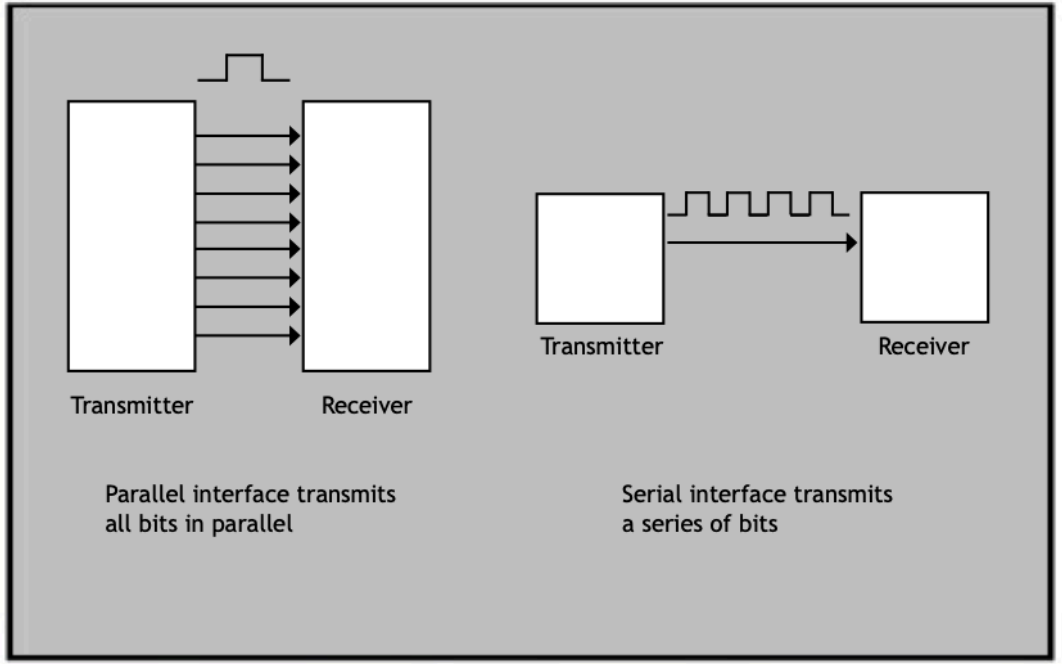
- Needed to communicate to sensors & other peripherals (& other devices)
- Implemented with hardware or software drivers
- Operate according to protocols
- Arduino libraries (Do not bitbang SPI)



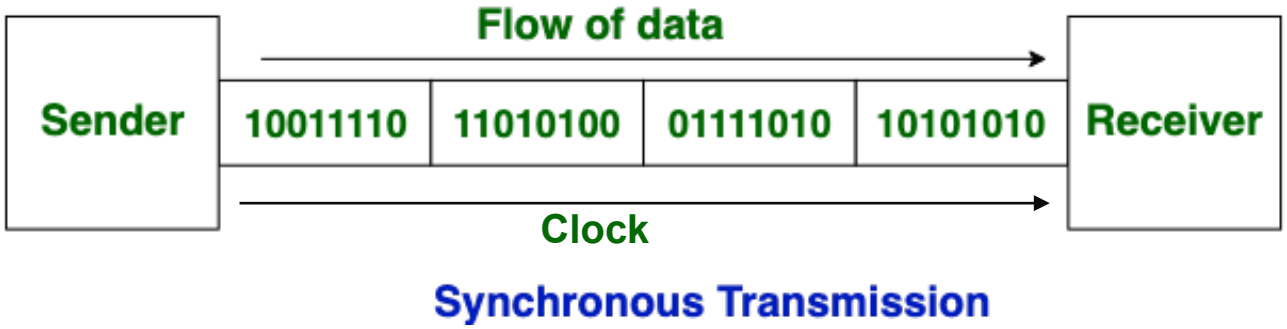


Buses

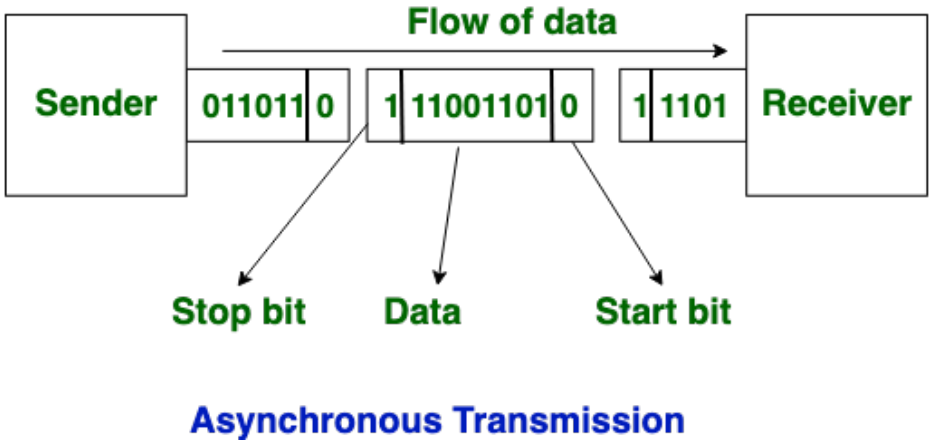
- Serial vs parallel



Buses



- Serial vs parallel
- Synchronous
  vs
  Asynchronous



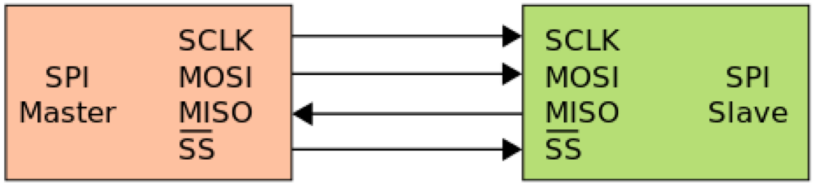
Buses

PCB Scale
- SPI, I2C/DDC/
SMBus, I2S,
1-Wire, **…**

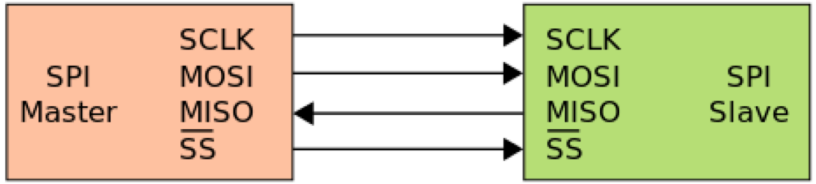
Device Scale
- **UART/serial,**
RS232, RS422,
RS485, CAN, LIN,
MIDI, USB, …

SPI

- Serial Peripheral Interface Bus
  - High speed (up to > 10 MB/s), full duplex capable
- Master initiated, simultaneous bidirectional data transfer capable
  - MISO (master in slave out), MOSI (master out slave in), SCK (serial clock), SS/CS (slave / chip select)
- Easy to use with Arduino libraries (SPI Library)



SPI



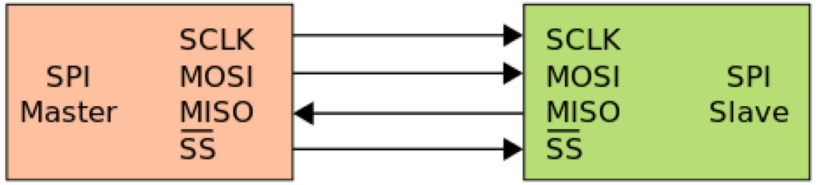
Advantages of using SPI

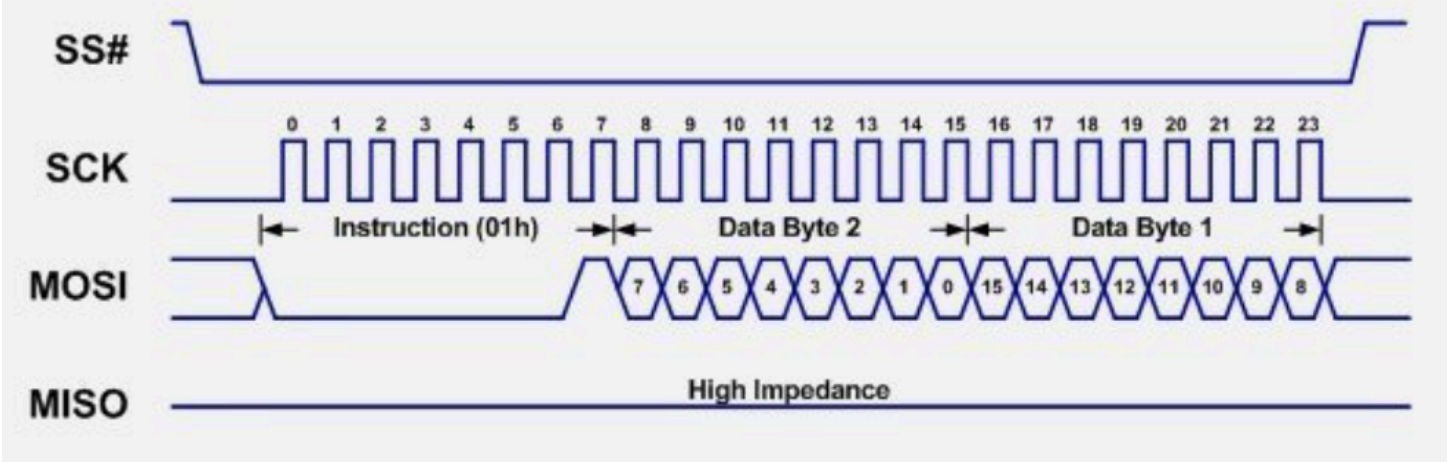
- The protocol is simple as there is no complicated slave addressing system like I2C.
- It is the fastest protocol compared to UART and I2C.
- No start and stop bits unlike UART which means data can be transmitted continuously without interruption
- Separate MISO and MOSI lines which means data can be transmitted and received at the same time

Disadvantages of using SPI

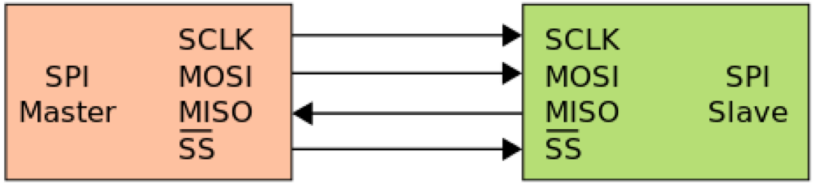
- More Pin ports are occupied, the practical is the number of devices + MOSI, MISO, NCLK, NSS
- There is no acknowledgement mechanism confirms whether data is received unlike I2C
- No form of error check unlike in UART (using parity bit)
- Only 1 master

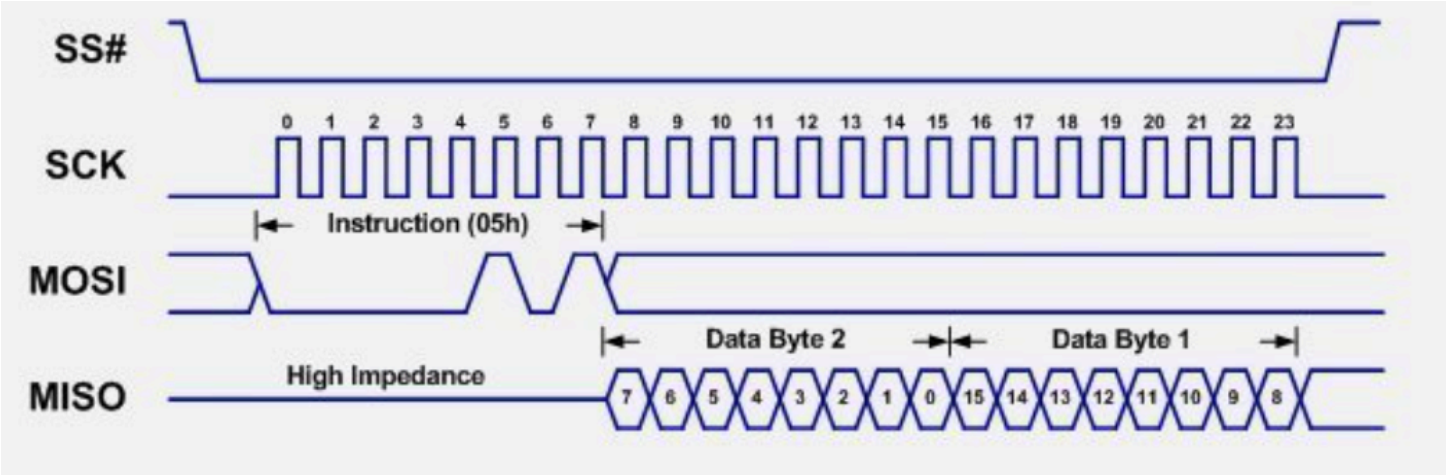
SPI


SPI
Master
SCLK
MOSI
MISO
SS
SCLK
MOSI
MISO
SS
SPI
Slave

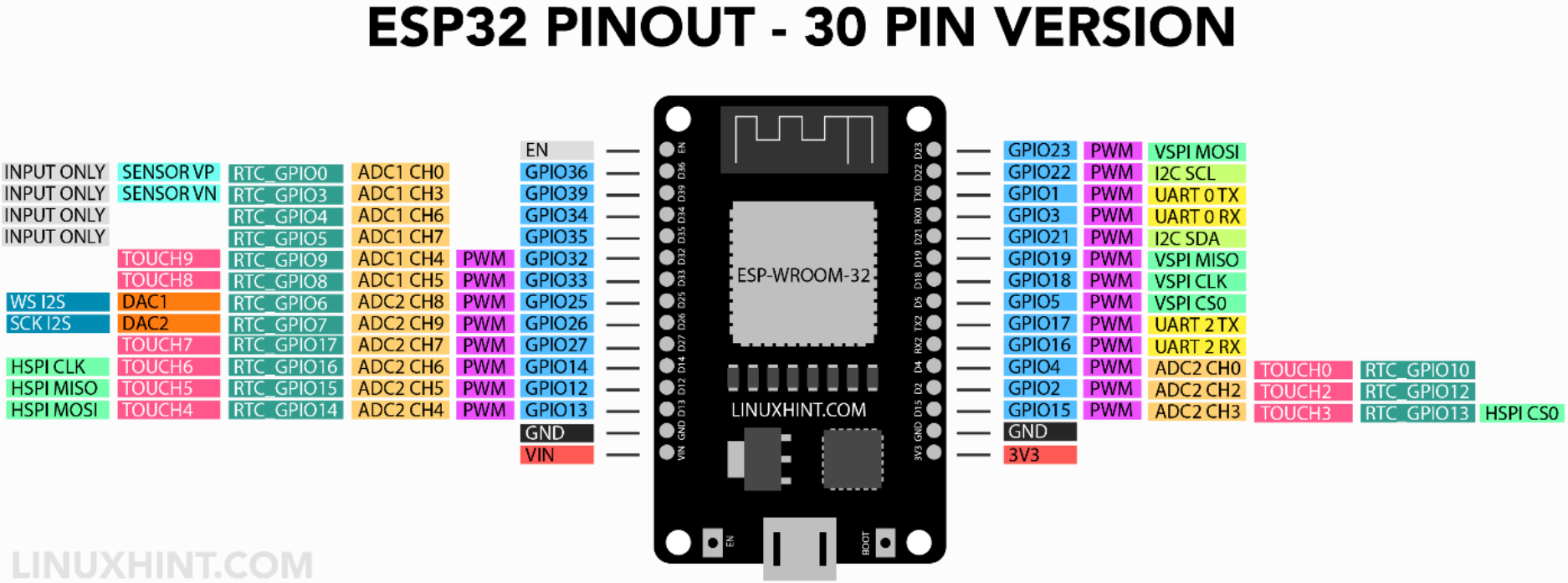

SS#
SCK
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Instruction (01h)
Data Byte 2
Data Byte 1
MOSI
7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8
MISO
High Impedance

SPI


SPI
Master
SCLK
MOSI
MISO
SS
SCLK
MOSI
MISO
SS
SPI
Slave


SS#
SCK
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Instruction (05h)
MOSI
Data Byte 2
Data Byte 1
MISO
High Impedance
7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8

Bit-Banging


ESP32 PINOUT - 30 PIN VERSION
ESP-WROOM-32
LINUXHINT.COM
EN
GPIO36 INPUT ONLY SENSOR VP RTC_GPIO0 ADC1 CH0
GPIO39 INPUT ONLY SENSOR VN RTC_GPIO3 ADC1 CH3
GPIO34 INPUT ONLY RTC_GPIO4 ADC1 CH6
GPIO35 INPUT ONLY RTC_GPIO5 ADC1 CH7
GPIO32 TOUCH9 RTC_GPIO9 ADC1 CH4 PWM
GPIO33 TOUCH8 RTC_GPIO8 ADC1 CH5 PWM
GPIO25 WS I2S DAC1 RTC_GPIO6 ADC2 CH8 PWM
GPIO26 SCK I2S DAC2 RTC_GPIO7 ADC2 CH9 PWM
GPIO27 TOUCH7 RTC_GPIO17 ADC2 CH7 PWM
GPIO14 HSPI CLK TOUCH6 RTC_GPIO16 ADC2 CH6 PWM
GPIO12 HSPI MISO TOUCH5 RTC_GPIO15 ADC2 CH5 PWM
GPIO13 HSPI MOSI TOUCH4 RTC_GPIO14 ADC2 CH4 PWM
GND
VIN
GPIO23 PWM VSPI MOSI
GPIO22 PWM I2C SCL
GPIO1 PWM UART 0 TX
GPIO3 PWM UART 0 RX
GPIO21 PWM I2C SDA
GPIO19 PWM VSPI MISO
GPIO18 PWM VSPI CLK
GPIO5 PWM VSPI CS0
GPIO17 PWM UART 2 TX
GPIO16 PWM UART 2 RX
GPIO4 PWM ADC2 CH0 TOUCH0 RTC_GPIO10
GPIO2 PWM ADC2 CH2 TOUCH2 RTC_GPIO12
GPIO15 PWM ADC2 CH3 TOUCH3 RTC_GPIO13 HSPI CS0
GND
3V3
LINUXHINT.COM

Bit-Banging

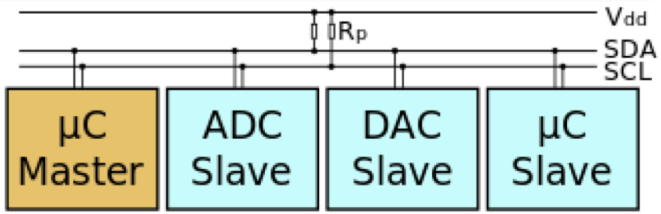
- ***If suitable hardware peripheral is not available*** **(or free), many interfaces can be emulated with bitbanging.**
  - **Interface behaviour is emulated by swinging GPIO pins to proper state at proper timing, using MCU processing time to control pins *WITH SOFTWARE***
  - **Essentially interface needs to be considerably slower than MCU/CPU speed**
    - *E.g. earlier versions of Arduino Software serial were able to run at max 19200 bps on 16MHz MCU (although now the library has been optimised for better speed)*
- **Many exotic interfaces need to be bit-banged, in lack of proper hardware peripherals, e.g. 1-Wire**

Bit-Banging SPI in C

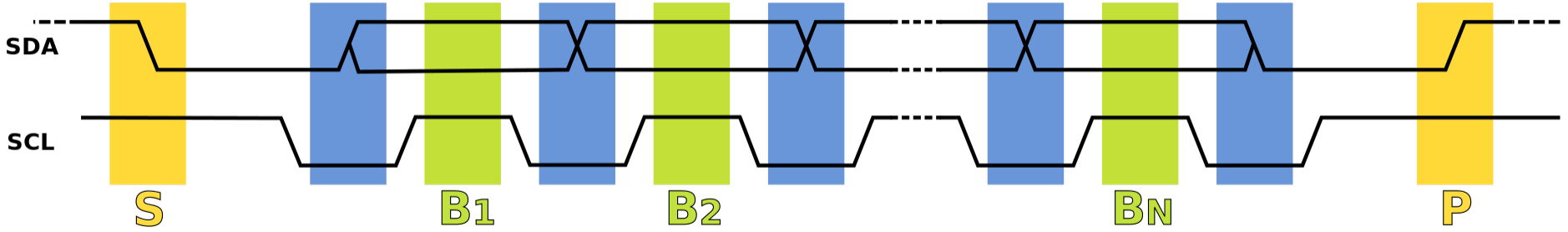
```c
// transmit byte serially, MSB first
void send_8bit_serial_data(unsigned char data)
{
   int i;

   // select device (active low)
   output_low(SD_CS);

   // send bits 7..0
   for (i = 0; i < 8; i++)
   {
       // consider leftmost bit
       // set line high if bit is 1, low if bit is 0
       if (data & 0x80)
           output_high(SD_DI);
       else
           output_low(SD_DI);

       // pulse the clock state to indicate that bit value should be read
       output_low(SD_CLK);
       delay();
       output_high(SD_CLK);

       // shift byte left so next bit will be leftmost
       data <<= 1;
   }

   // deselect device
   output_high(SD_CS);
}
```

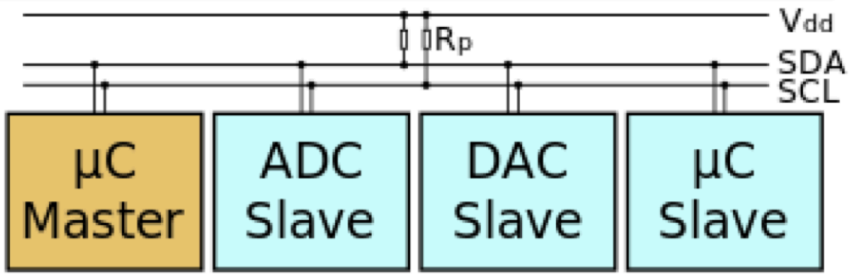
I2C

- Inter-Integrated Circuit, Display Data Channel, System Management Bus…
  - Low speed: 400 / 100 kHz usually, but higher speed devices available (>1 MHz)
  - Developed, Patented & Controlled by Philips Semiconductors
- Master initiated, half-duplex
  - SDA (SerialData), SCL (SerialClock), by default pulled up(pull-up resistors)
  - Several devices can share same bus, (each has 7-bit unique address)
  - Available at VGA, DVI, HDMI-Connectors
  - On PC motherboards RAM connectors (DIMMs)
- Easy to use with Arduino (Wire Library)

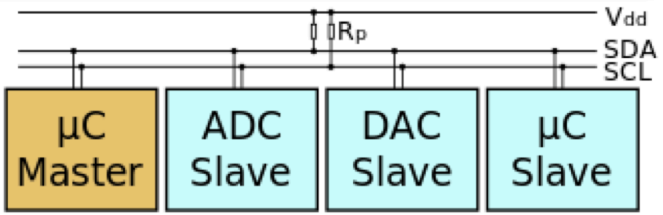


I2C


SDA
SCL
S
B1
B2
BN
P


Vdd
Rp
SDA
SCL
µC Master
ADC Slave
DAC Slave
µC Slave

I2C



Advantages of using I2C

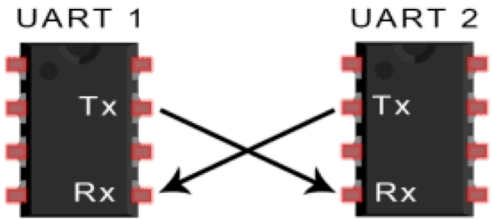
- Has a low pin/signal count even with numerous devices on the bus Simple!
- Flexible, as it supports multi-master and multi slave communication.
- Adaptable as it can adapt to the needs of various slave devices.

Disadvantages of using I2C

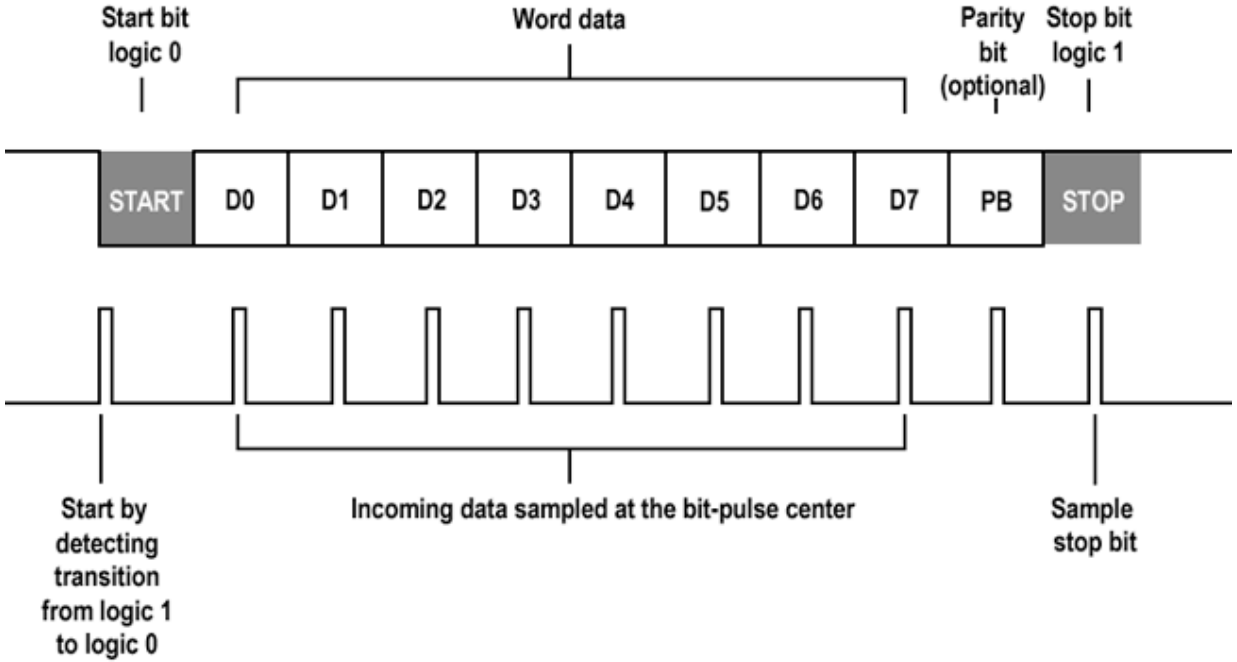
- Slower speed as it requires pull-up resistors rather than push-pull *transistors* used by SPI. It also has an open-drain design = limited speed.
- Requires more space as the resistors consume valuable PCB real estate.
- May become complex as the number of devices increases.

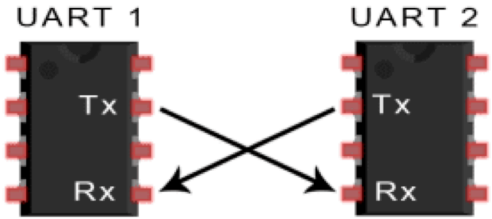
UART

- Serial, Asynchronous, Bidirectional, half-/fullduplex
  - Only TX and RX, no separate clock signal
  - Simple, Easy to use
- Protocol not defined, several standard electrical interfaces
- Usually used for specific peripherals, E.g. Bluetooth transmitters, GPS, GSM or Inter-device communication
- Arduino library: (Serial, SoftwareSerial)



UART

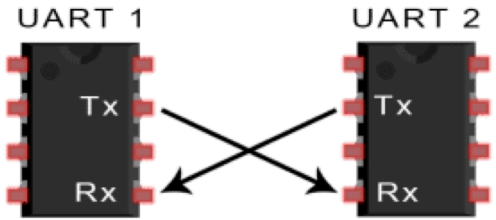

Start bit
logic 0
Word data
Parity
bit
(optional)
Stop bit
logic 1
START
D0
D1
D2
D3
D4
D5
D6
D7
PB
STOP
Start by
detecting
transition
from logic 1
to logic 0
Incoming data sampled at the bit-pulse center
Sample
stop bit


UART 1
UART 2
Tx
Rx
Tx
Rx

UART

Advantages of using UART

- Simple to operate, well documented as it is a widely used method with a lot of resources online
- No clock needed
- Parity bit to allow for error checking



Disadvantages of using UART

- Size of the data frame is limited to only 9 bits
- Only two devices
- Baud rates (“clocks”) of each UART must be within 10% of each other to prevent data loss.
- Low speed

Other

- 1-Wire: Low speed single datawire bus by Dallas/Maxim
  - Several devices can share same data bus
  - E.g. used in DS18x20 digital interface temperature sensors
  - Arduino library (OneWire)

- MIPI: Camera & Display Serial interface, HD resolutions
  - Requires driver to work
  - Found on Raspberry Pi platforms
- USB: Hard, complicated protocol
  - Always requires a driver (usually it is easier to use a USB-UART emulating adapter)
  - Supplies power, max 500 mA