Introduction to Microcontrollers: Arduino Tutorial

Vlad—Mihai PLACINTA[1,2]

1. Horia Hulubei National Institute for R&D in Physics and Nuclear Engineering
2. University POLITEHNICA of Bucharest

| vlad-mihai.placinta@cern.ch | www.nipne.ro/dpp/Collab/LHCb/upgrade.html |

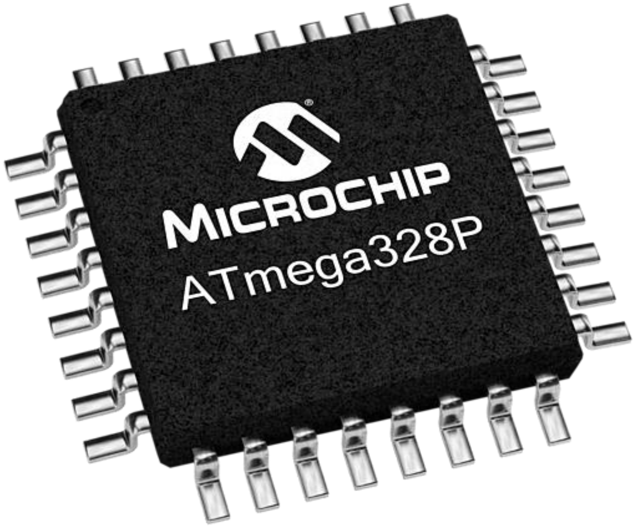
OUTLINE

- Introduction
- Microcontrollers Architecture
    - Arduino environment
        - Arduino boards
        - Sensors interface
        - Application control
        - Arduino UNO board
        - Coding example
        - Applications
- Bibliography

Introduction

What is a microcontroller?

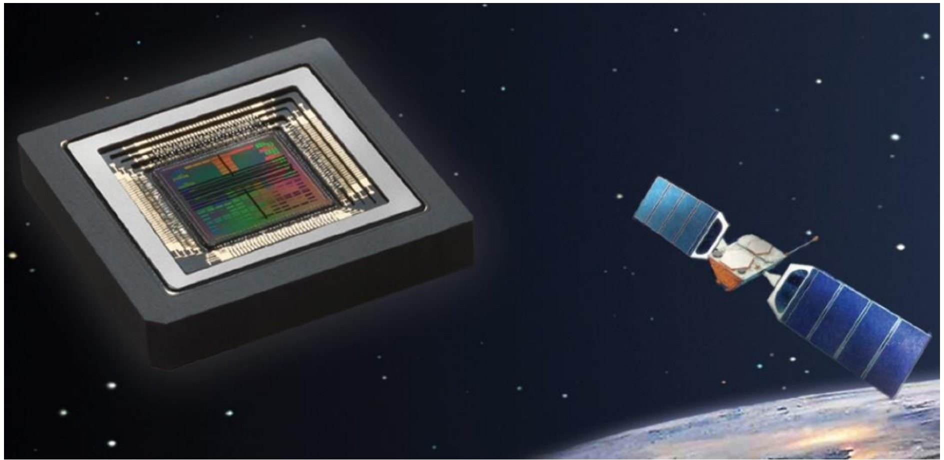
- Small computers integrated in a single chip:
  - CPU, RAM, EEPROM and other Peripherals in the same package.
- Excellent for embedded applications;
- Low cost and low power consumption; (ATMEGA328P ~ 1.5 $)
- Can work up to tens of MHz as clock frequency;
  - Can work with low frequency also, few MHz.
- Standalone devices;
  - most of them only need power and a clock source to run.



Introduction

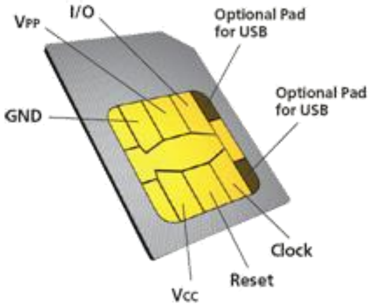
What are they used for?

- Data control;
- Data Acquisition Systems;
- Power control and monitoring;
- Motor control applications;
- Smartphones;
- Automotive industry;
- Nuclear instrumentation;
- Space applications.



https://atmelcorporation.wordpress.com/tag/avr-microcontrollers/

- In general are used in applications where high processing resources are not required, hence a general purpose microcontroller is considered suitable due the low power consumption and without any peripheral and external memories;
- Also, the cost may be an important factor.



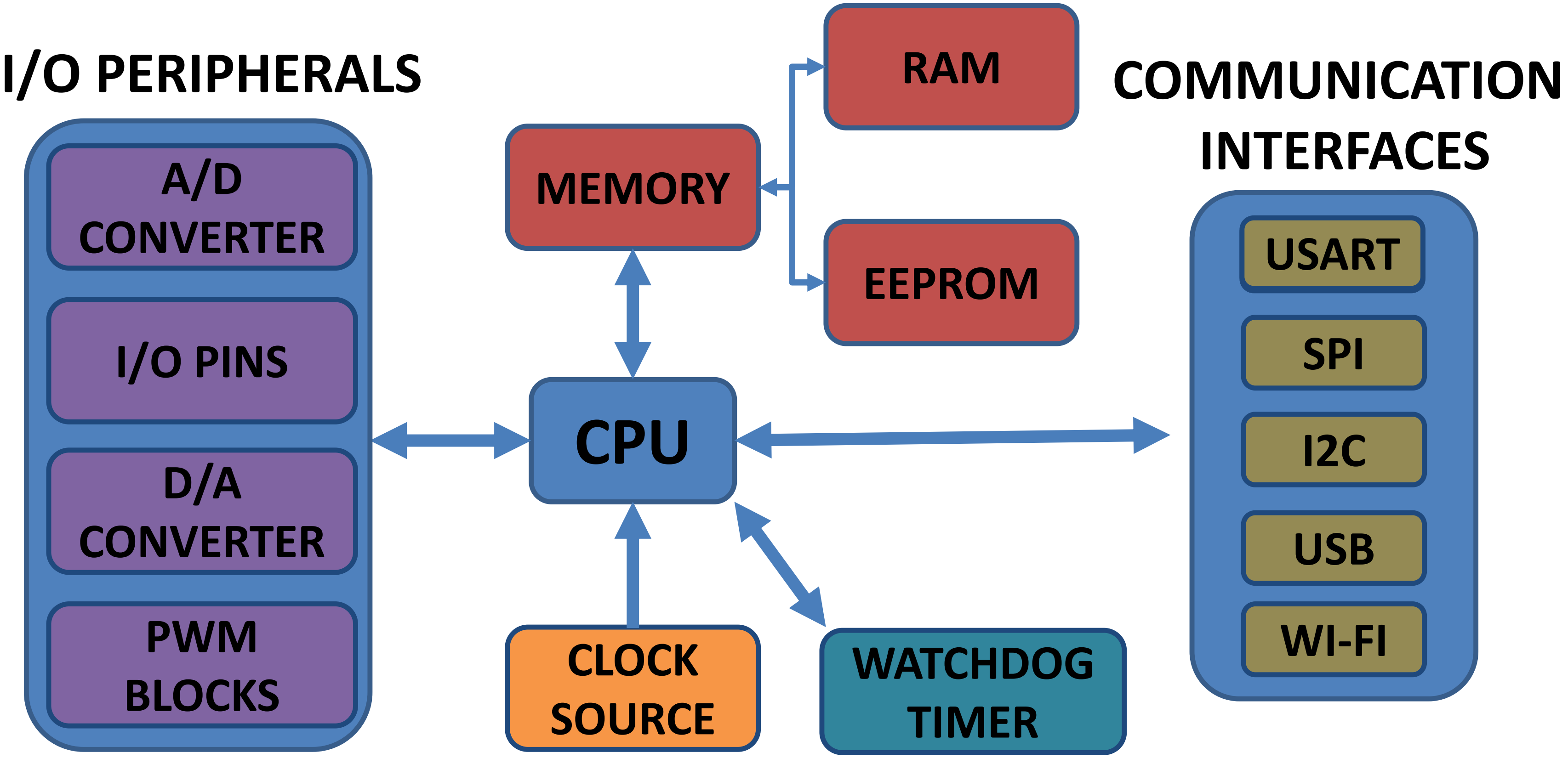
Introduction

Famous microcontroller manufacturers:

- **Microchip;**
- **Atmel (now is part of Microchip);**
- **Intel;**
- **Analog Devices;**
- **MAXIM Integrated;**
- **Renesas Electronics.**

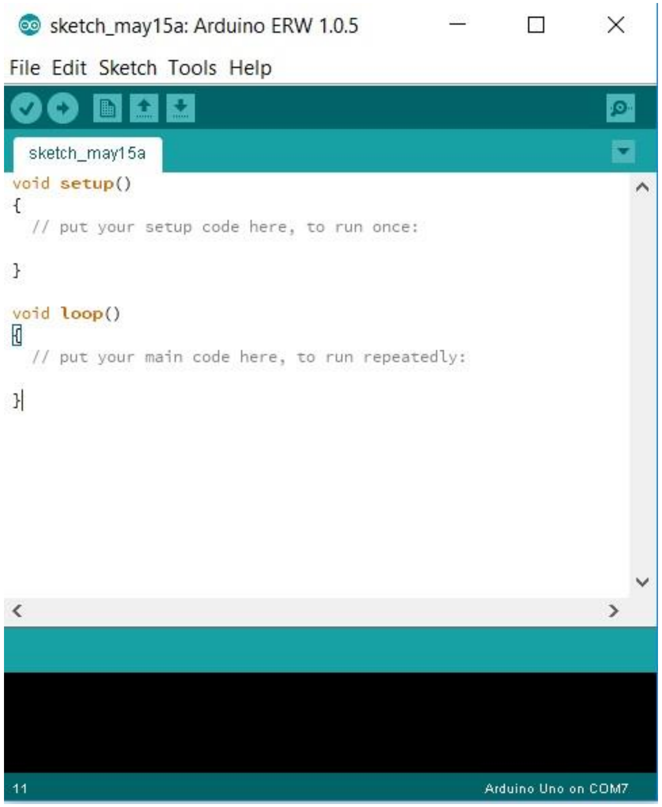
A detailed list can be found at:
http://embedeo.org/microcontroller_manufacturers/

Microcontrollers Architecture



Simplified Architecture

Arduino environment



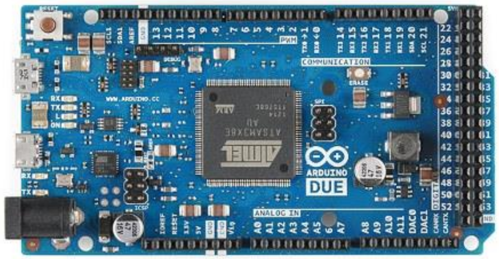
```
void setup()
{
  // put your setup code here, to run once:

}

void loop()
{
  // put your main code here, to run repeatedly:

}
```

- ➢ Open source-source electronics development board, based on easy-to-use hardware and software interface;
- ➢ Very easy to use and its has a broad range of libraries and example codes;
- ➢ Arduino language is merely C/C++;
- ➢ It has embedded an U(S)ART monitor in order to check the communication with microcontroller;
- ➢ Using Processing software (embedded in the latest Arduino compiler version) data can be displayed, GUIs can be made;
- ➢ More at: https://www.arduino.cc/ .

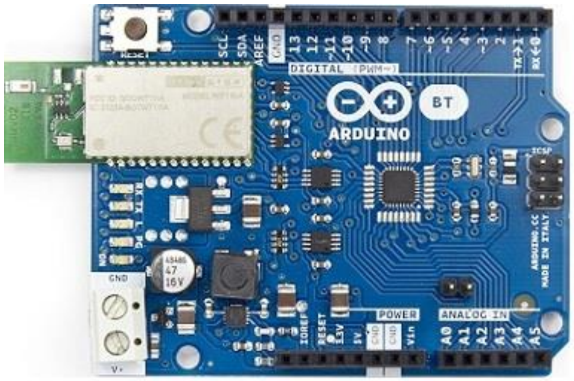
Arduino Boards

Arduino UNO

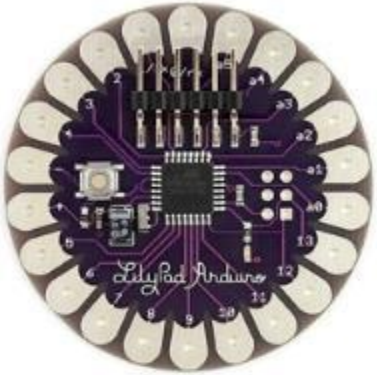
Arduino Mega 2560



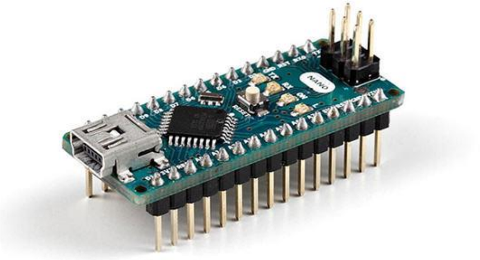
Arduino Due



Arduino BT

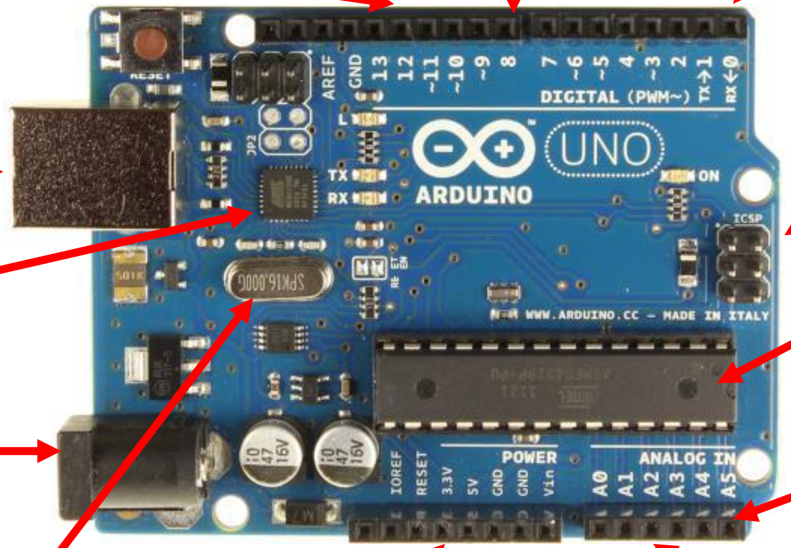


Arduino LilyPad



Arduino NANO

Arduino UNO Board



Hardware SPI block:
13=> SCLK
12=> MISO
11=> MOSI
10=> SS

Digital I/O PINS
Pulse Width Modulation
(PWM) included (~)

Direct connection to the hardware UART block

In circuit Serial programming

USB Port

USB-UART bridge

ATMEGA328P Microcontroller

Input voltage
(7-12 V recommended)

Hardware I2C
A4 => SDA
A5 => SCL

16 MHz quartz oscillator

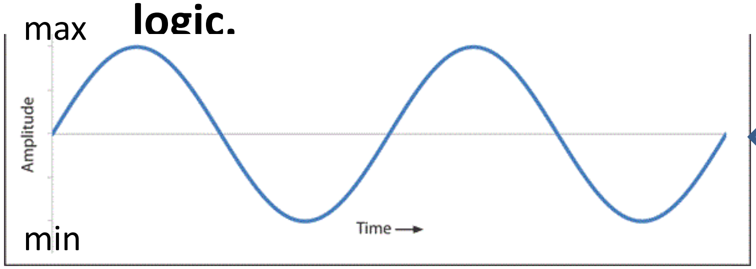
Power Supply Distribution
(can be used to power up peripherals)

Analog inputs
10-bit within the range 0 – Vref (max. 5 V)

Sensor interface

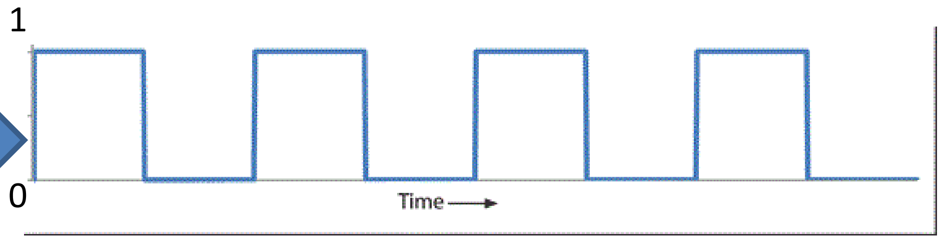
Analog or digital?

- Sensor: an analog/digital device which converts the physical quantity in to an electrical signal;
- All related physical quantities are related to ANALOG interface, where the quantity can take any value between the hardware minimum and maximum values;
- DIGITAL interface is used to describe the quantity which has only 2 levels, 1 or 0 logic.



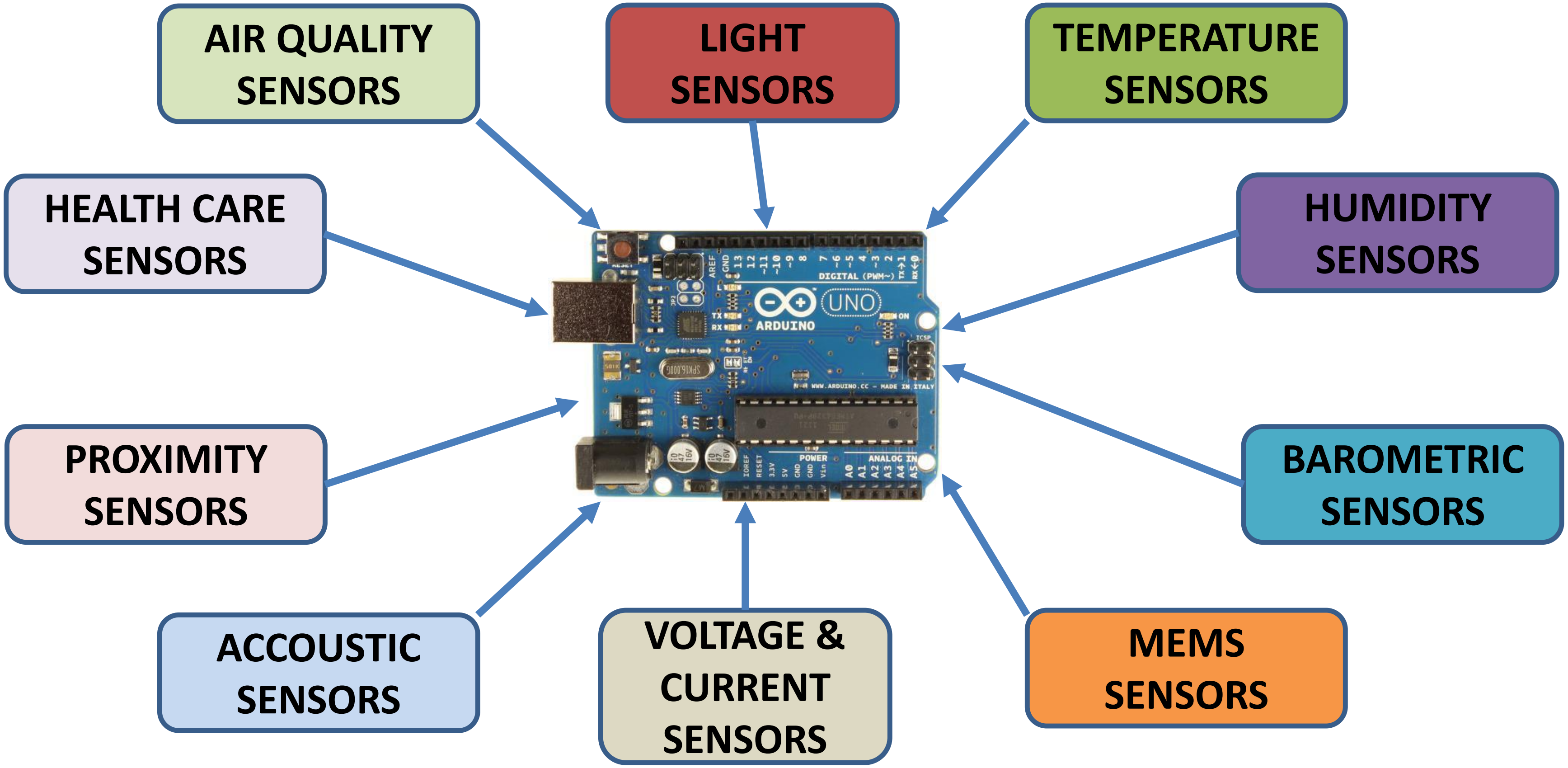
Sine waves are analog, because the amplitude can vary between minimum and a maximum value

Square waves are digital, because the amplitude can vary only between 2 values, 0 or 1 logic



Sensor interface

Classification



AIR QUALITY SENSORS

LIGHT SENSORS

TEMPERATURE SENSORS

HEALTH CARE SENSORS

HUMIDITY SENSORS

PROXIMITY SENSORS

BAROMETRIC SENSORS

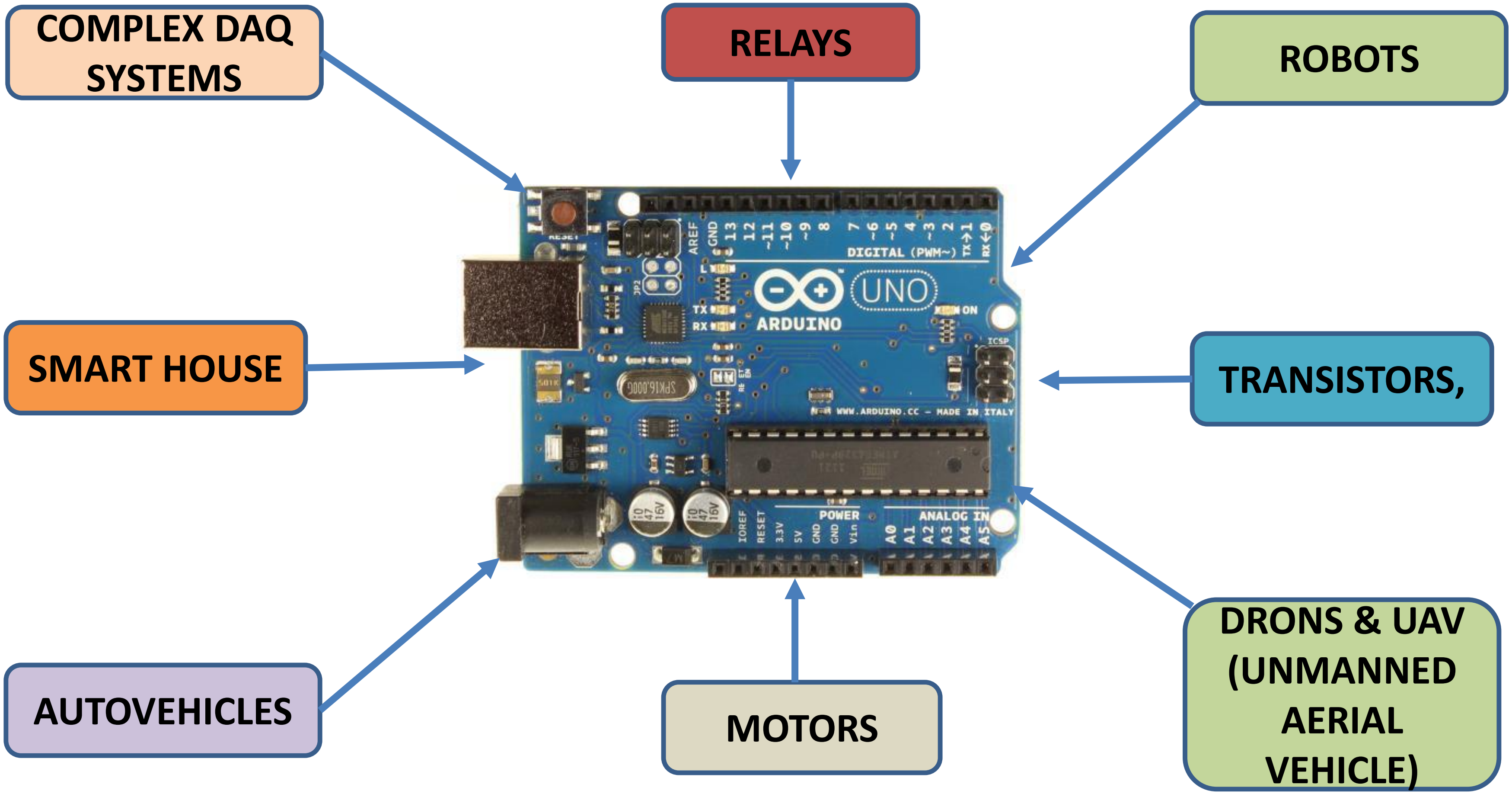
ACCOUSTIC SENSORS

VOLTAGE & CURRENT SENSORS

MEMS SENSORS

MEMS => Microelectromechanical systems.

Application control



COMPLEX DAQ SYSTEMS

RELAYS

ROBOTS

SMART HOUSE

TRANSISTORS,

AUTOVEHICLES

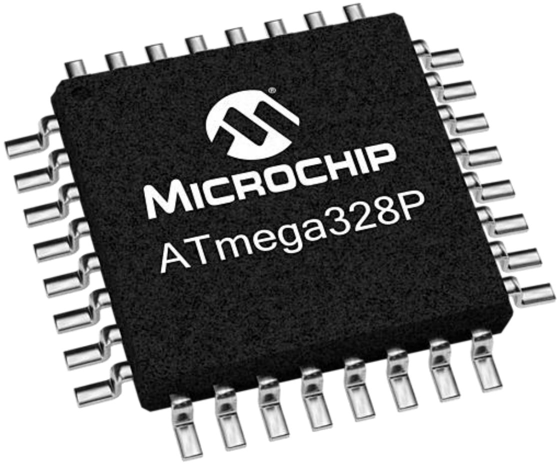
MOTORS

DRONS & UAV (UNMANNED AERIAL VEHICLE)

Arduino UNO Board

- 8-bit RISC architecture microcontroller;
- 32 KB In-System Self-Programmable Flash program memory;
- 1 KB EEPROM, 2 KB SRAM ;
- 6 PWM channels;
- 8 x 10-bit resolution ADC inputs;
- UART, SPI, I2C;
- On chip analog comparator;
- 3 Timers: 2 x 8-bit and 1 x 16-bit;
- Up to 23 Programmable I/O lines;
- Write/erase cycles: Up to 10.000 for Flash/ 100.000 for EEPROM;
- Speed Grade: 0 - 4 MHz @ 1.8 - 5.5 V, 0 – 10 MHz @ 2.7 – 5.5 V, 0 – 20 MHz @ 4.5 – 5.5 V;
- Power consumption: 0.2 mA in Active Mode @ 1 MHz, 1.8 V and 25 °C.

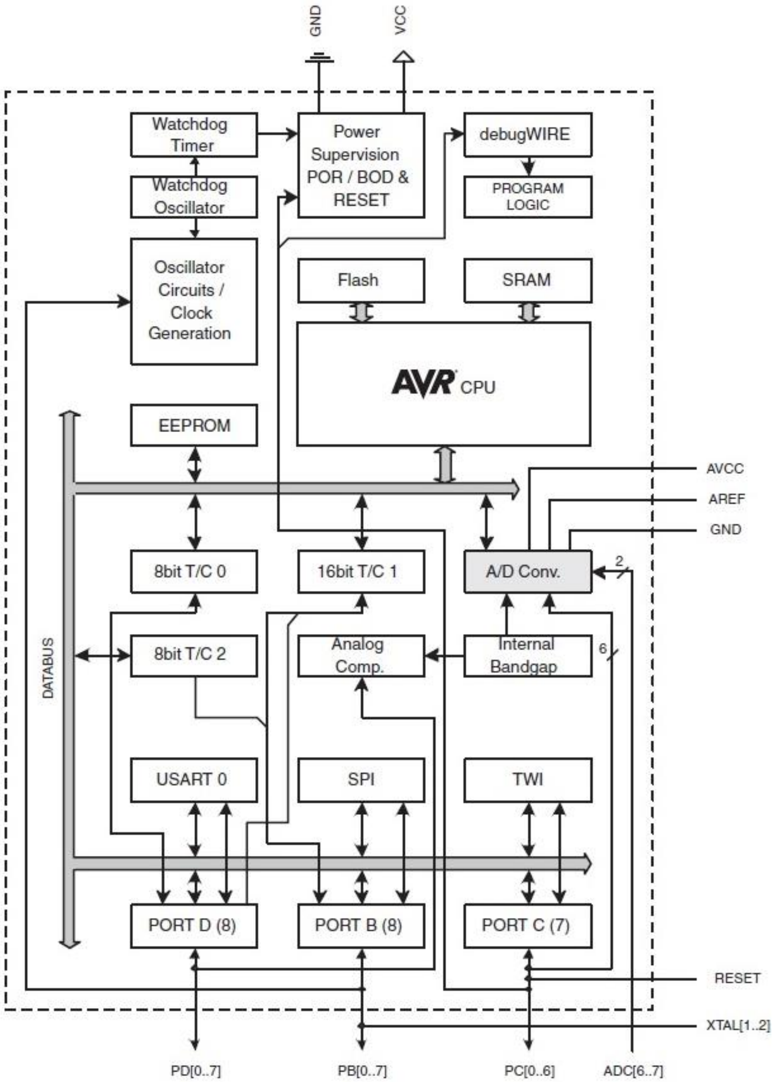
ATMEGA328P Microcontroller



Full datasheet

Arduino UNO Board

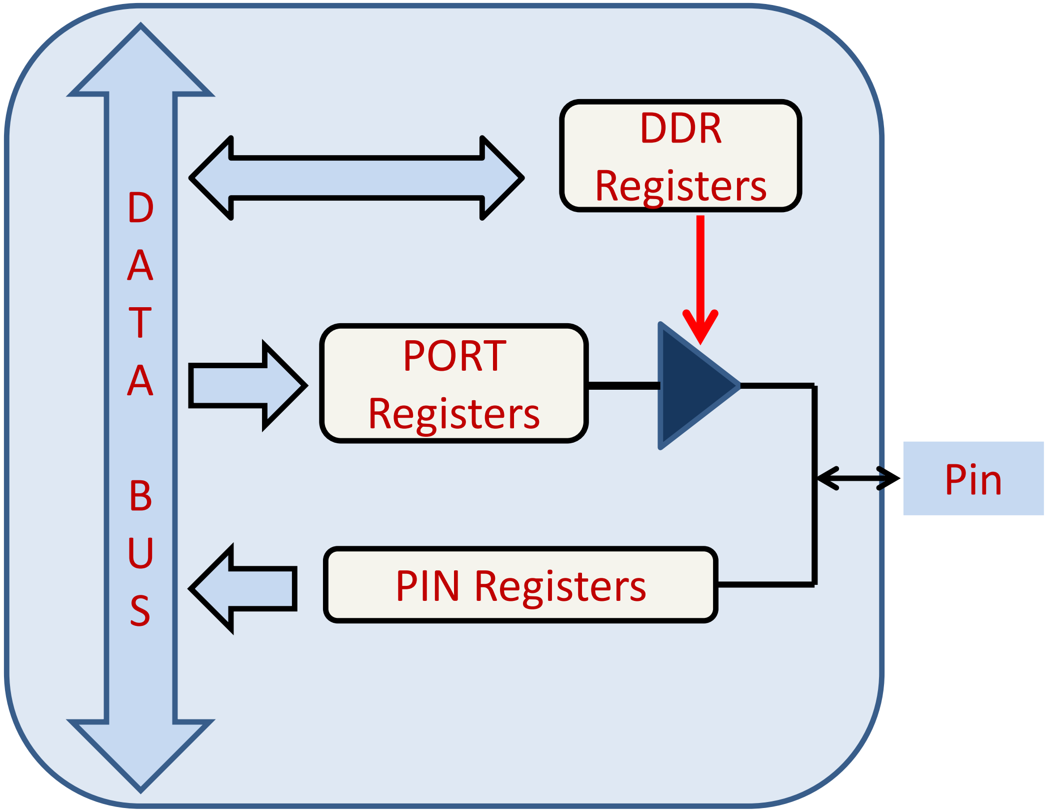
ATMEGA328P Architecture



Full datasheet

General purpose Input/Output (GPIO)

- Are the interface of logic software with external hardware;
- Can be programmable as Input or Output;
- Can read or write digital signals;
- logic 0 = 0 V and logic 1 = Vcc;
- Are controller by 3 registers:
  - DDR (Data Direction Register);
  - PORT (load data when the pin is set as output);
  - PIN (load data when the pin is set as input).



Timers / Counters

- Specific internal registers that increment data and be triggered by:
  - A clock source : Timer;
  - An external event : Counter.

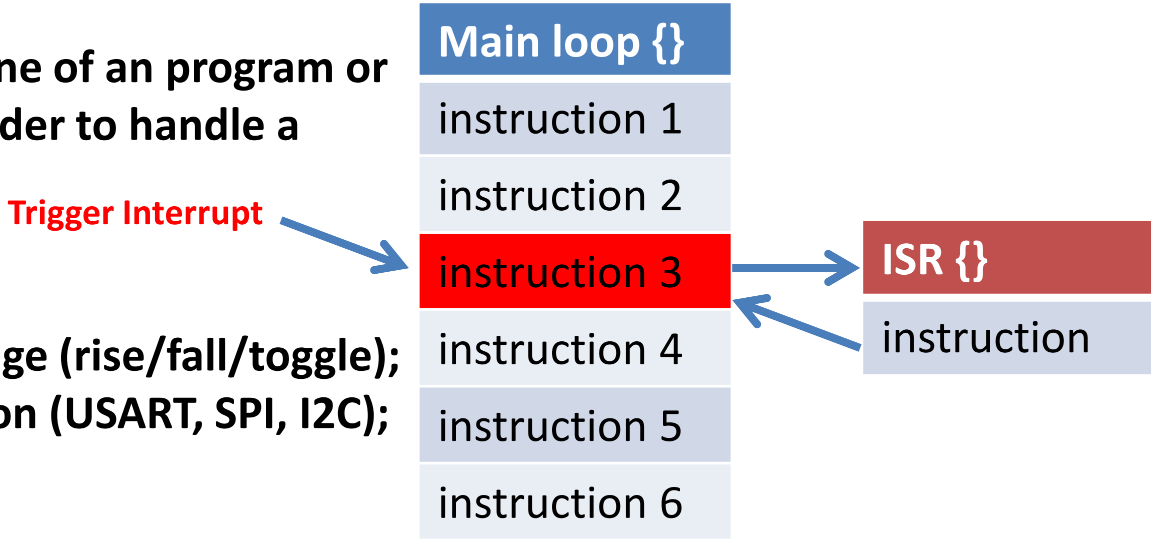
HC- SR04 ultrasonic distance sensor, it returns the distance measurement based on echo time of an ultrasonic pulse.

Utilities:

- Time domain measurements;
- Create PWM waveforms.

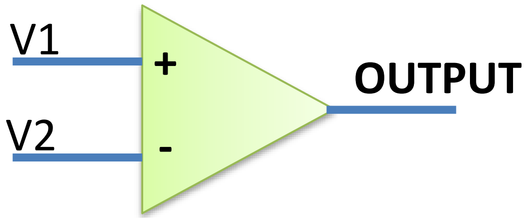
Interrupts

- Are used to break a routine of an program or entire program flow in order to handle a specific event.
- Are triggered by:
  - Input pin state change (rise/fall/toggle);
  - Serial communication (USART, SPI, I2C);
  - ADC state registers;
  - Analog comparator;
  - Timers or Counters.



ISR => Interrupt Service Routine

Internal Analog Comparator



OUTPUT
- 1 for V1 > V2
- 0 for V1 < V2

Utilities:

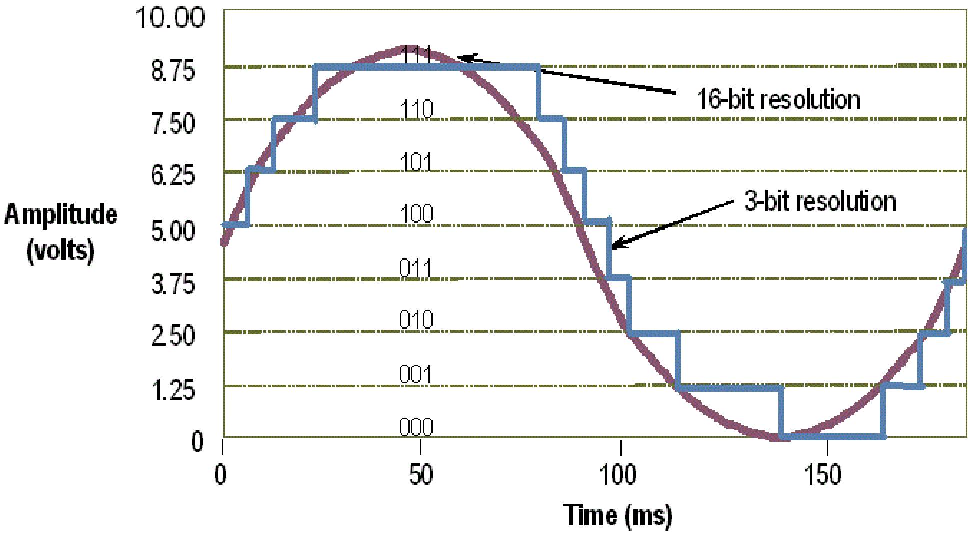
- Compare 2 analog voltages;
- Trigger a Timer/Counter1 Input Capture function;
- Trigger an Interrupt (rise, fall, toggle).

- ANI0 => V1;
- ANI1 => V2;
- Its output value is located in the ACSR register, named ACO bit;
- V2 can be tied to the output of the ADC multiplixer, hence can be tied to any of the analog inputs;
- As a general information is good to know that any comparator is an 1-bit ADC;

ACO => Analog Comparator Output;
ACSR => Analog Comparator and Status Register.

More information at page 234

Analog to Digital Converter (ADC)



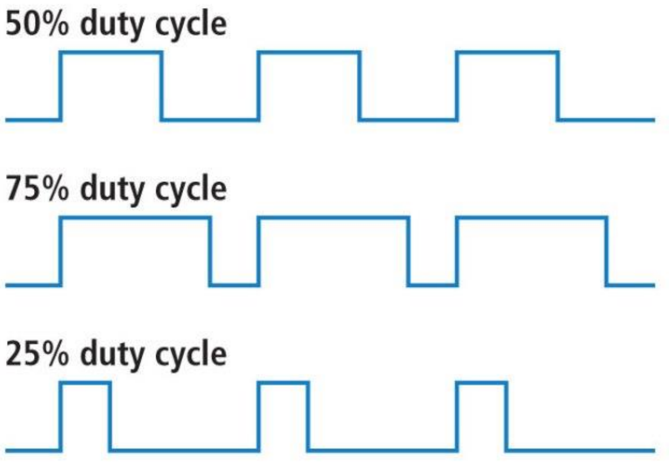
More about ADC

- 10-bit resolution => 0 – Vref => 0-1023;
- Successive-approximation (SAR) architecture;
- 1 ADC with up to 8 multiplexed input channels => they share the sampling rate;
- one channel is dedicated for internal temperature monitoring;

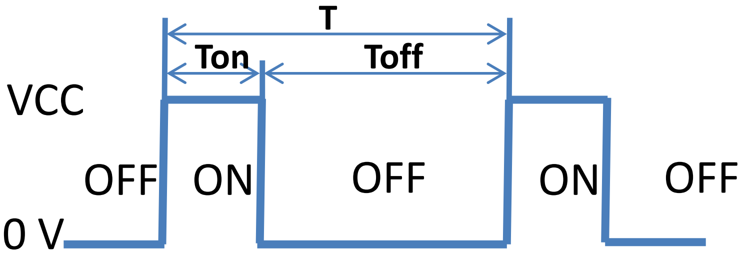
- Vref can be:
  - Vcc from power source; (not very recomended because the power supply can be noisy)
  - 1.1 V internal voltage reference;
  - External, from an external voltage reference.

Pulse width modulation (PWM)

- A square signal can be generated with a variable duty cycle;
- ATMEGA328P has six 8-bit PWM channels;
- Can be used for:
  - To control DC-DC converters;
  - To control the speed for motors;
  - To control the luminosity of Leds or lamps;
  - To generate an analog voltage direct proportional with duty cycle value.



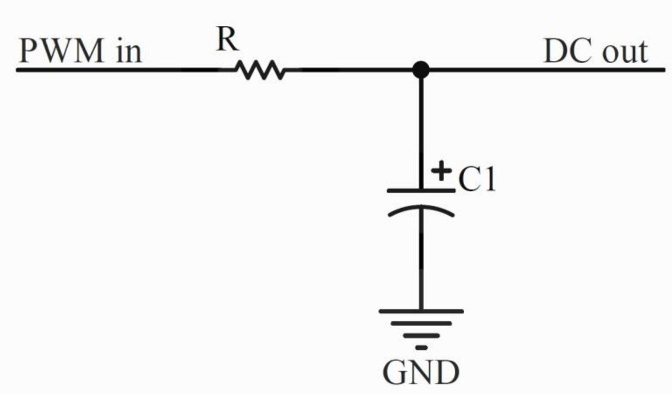
Duty cycle = $\frac{\text{Ton}}{\text{T}}$

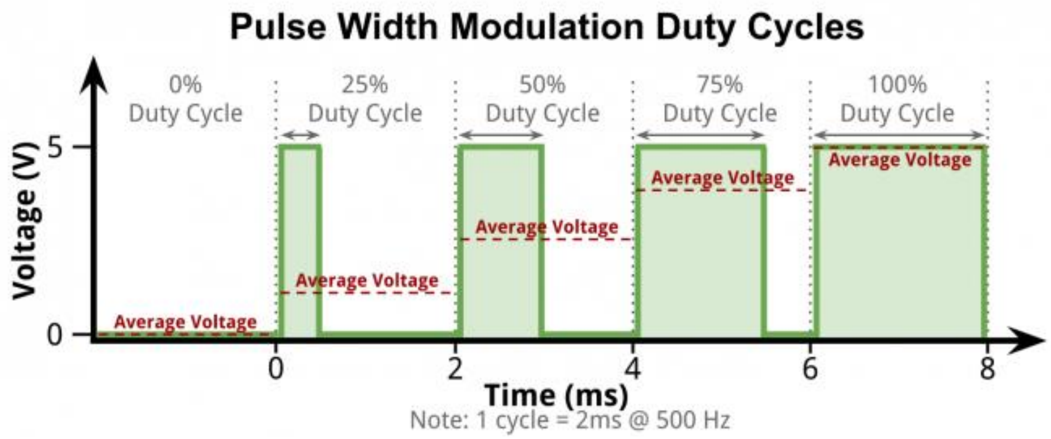


25 % Duty Cycle

Pulse width modulation (PWM)

- PWM can be used to generate a stable DC signal if its passed to a low pass filter, RC passive integrator;
- An 8 bit PWM signal will emulate an 8-bit DAC (Digital to Analog Converter).





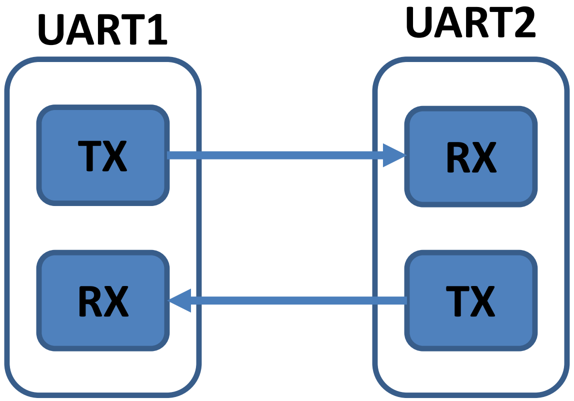
Source

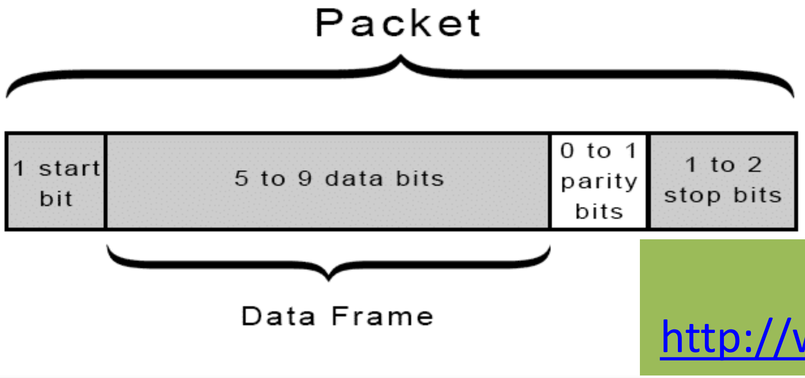
Read more

Communication interfaces

~USART~

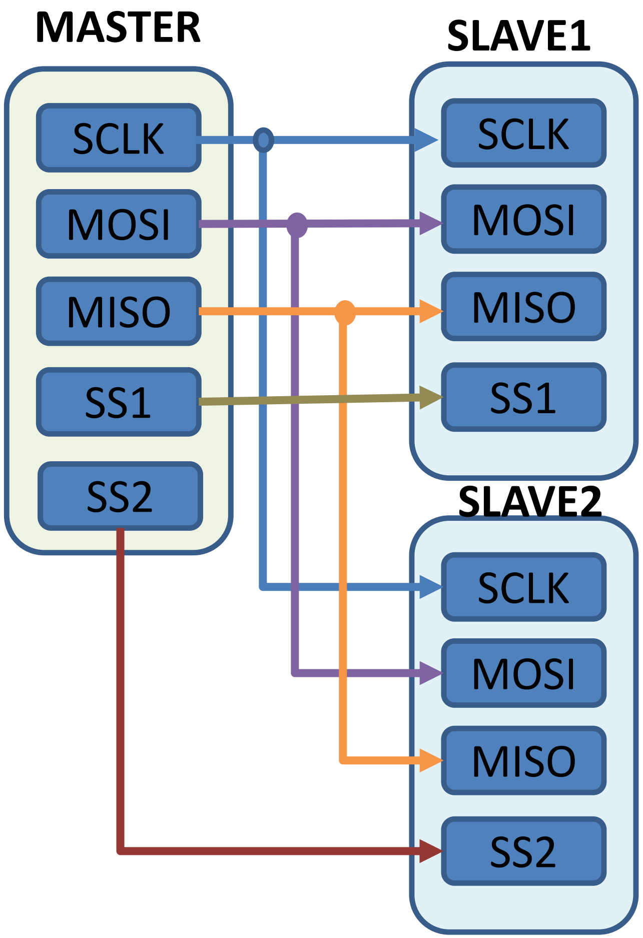
- Stands for Universal Synchronous/Asynchronous Receiver/Transmitter;(one to one communication)
- Most common is used UART => Universal Asynchronous Receiver/Transmitter;
- The data transfer speed is defined as baud rate (bit rate);
- Normally is defines as:
  - One start bit;
  - 8 data bits;
  - 1 stop bit.





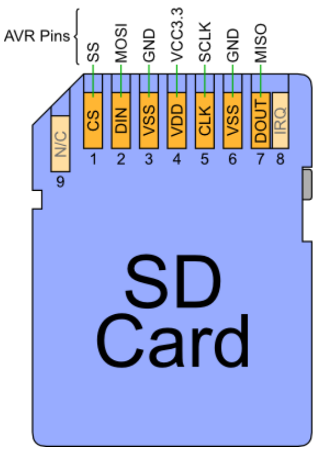
More at:
http://www.circuitbasics.com/basics-uart-communication/

Communication interfaces
~SPI~



- Stands for Serial Peripheral Interface, and its describes a serial communication used in embedded systems;
- Support multiple slaves communications, but only one at time;
- Can perform full duplex;
- One to many communication.

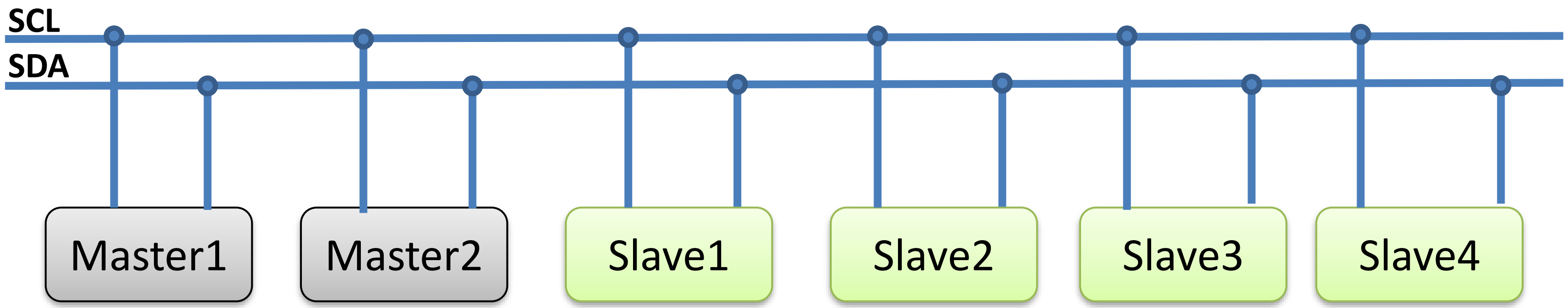
SCLK => Serial Clock;
MOSI => Master Output Slave Input;
MISO => Master Input Slave Output;
SS => Select Slave.



Read more about SPI

Communication interfaces
~I2C~

- Stands for Inter-Integrated Circuit, also know as Two Wire Interface (TWI);
- Serial protocol which allows multiples masters and slaves on the same bus, up to 128; (many to many communication)
- Normally each device has an unique 7-bit or 10-bit address in the I2C bus;
- Being used in variety of digital sensors, can reach speeds up to 400 kbps. (Arduino UNO)



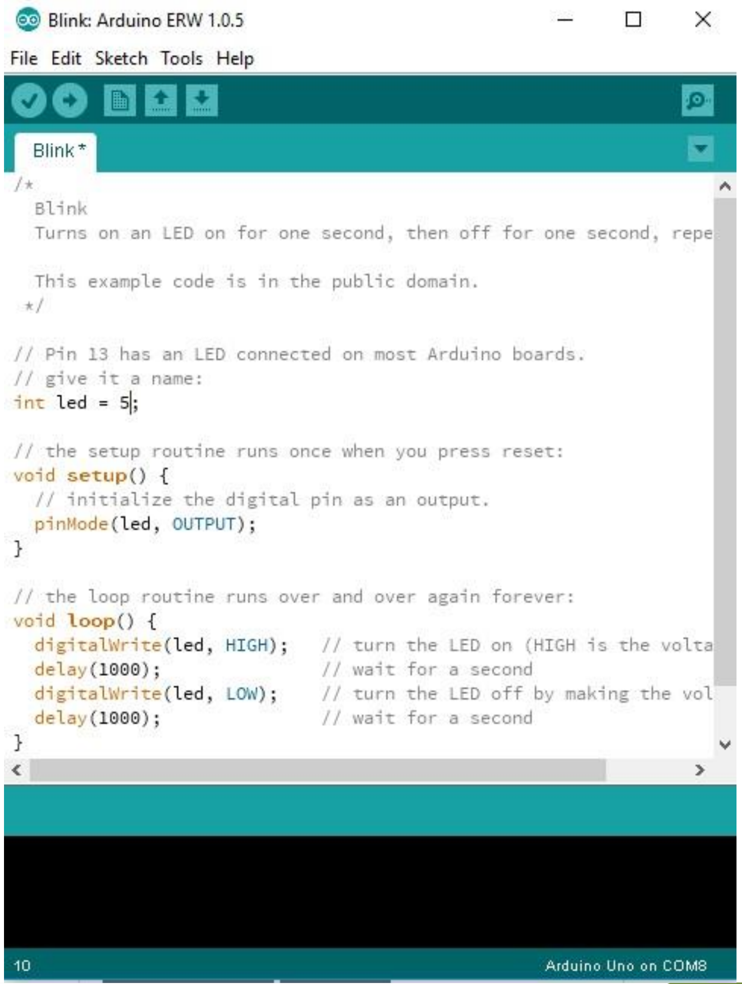
SCL=> Serial Clock Line;
SDA=>Serial Data Line.

Read more about I2C

Coding example

~blinking a led~

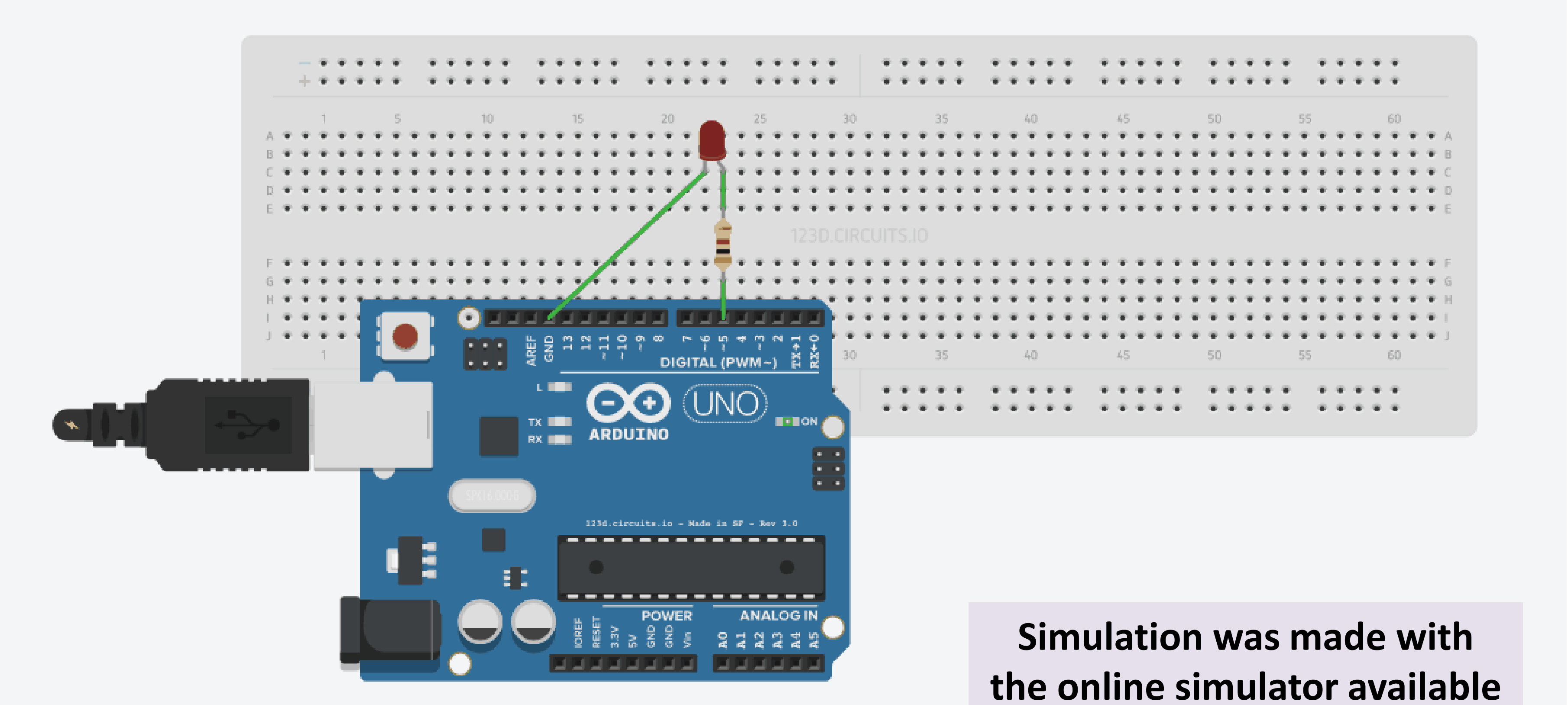
- First of all, the latest version of Arduino compiler must be installed from: https://www.arduino.cc/ ;
- Run the compiler and go to File=>Examples=>Basics=>Blink ;
- A new windows with the related software should appear on your screen;
- Compile the code and upload it on the board; (The board new to be connected to the PC already)
- Change the delay value, and observe what happen.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 5;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the volta
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the vol
  delay(1000);               // wait for a second
}
```

Coding example

~blinking a led~

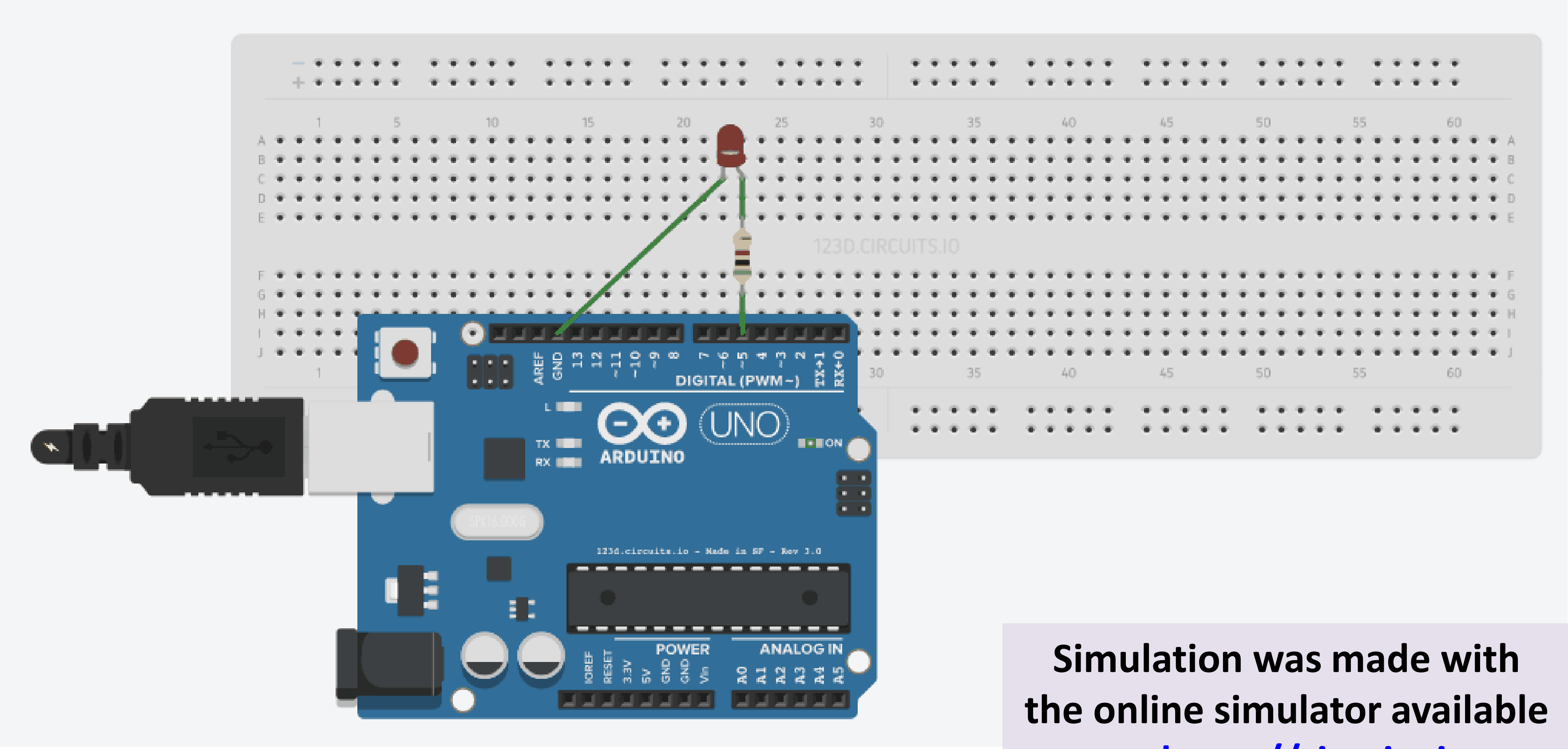


Simulation was made with the online simulator available at: https://circuits.io

Delay value = 1000 ms

Coding example

~blinking a led~



Delay value = 250 ms

Simulation was made with the online simulator available at: https://circuits.io

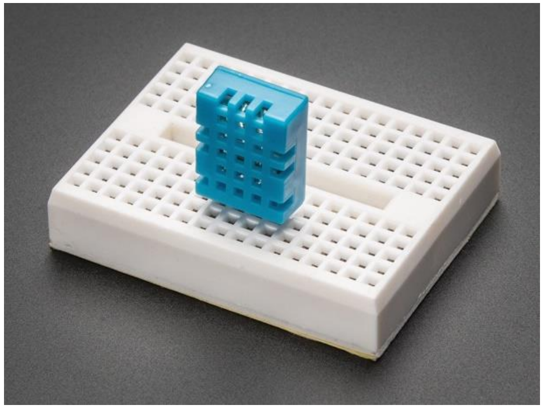
Applications

- ➢ Temperature & Humidity sensor readout;
- ➢ Proximity sensors readout => ultrasonic radar;
- ➢ Leds control;
- ➢ Light sensor application;
- ➢ Wearable multi-sensor data acquisition system.

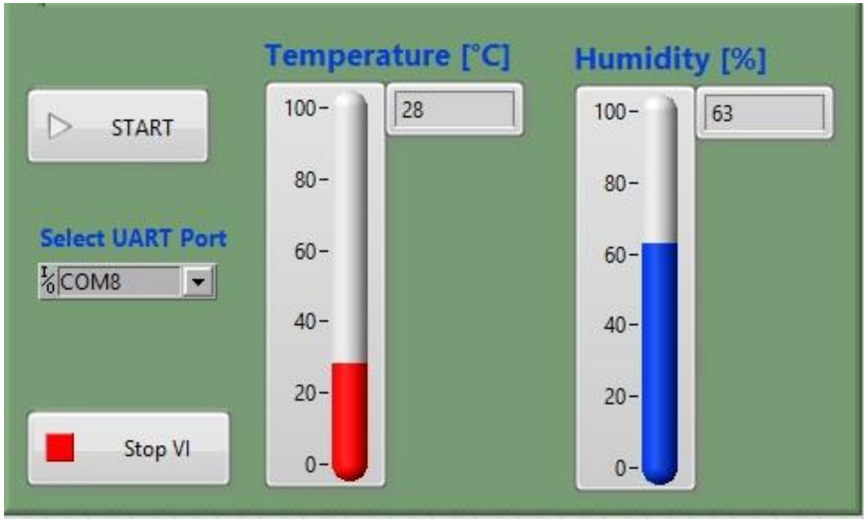
❖ Readout will be made with the Arduino UNO board and data will be send via UART or Bluetooth to an LabVIEW GUI (Graphical User Interface).

Temperature & Humidity Sensor Readout

- Based on low cost DHT11 sensor with 1 wire interface;
- Sampling rate need to be lower than 1 Hz (once every second);
- Data is acquired and displayed with a LabVIEW GUI over UART interface;

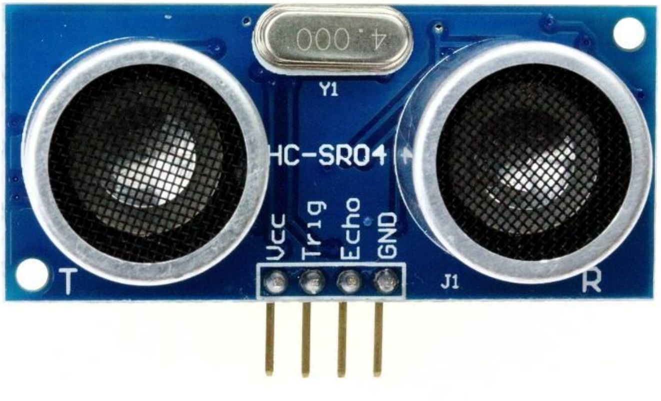


More info

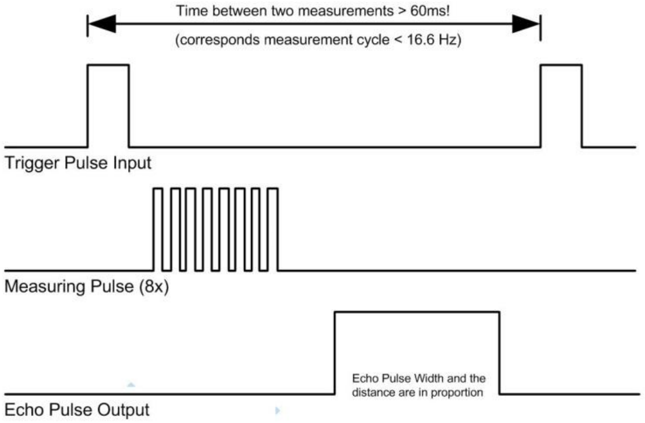


Ultrasonic Radar

- Provide good measurement between 2 and 400 cm;
- Easy to interface with any microcontroller;
- Widely used in robots and system were distance to a specific object is needed to be know.
- [A good reference article]

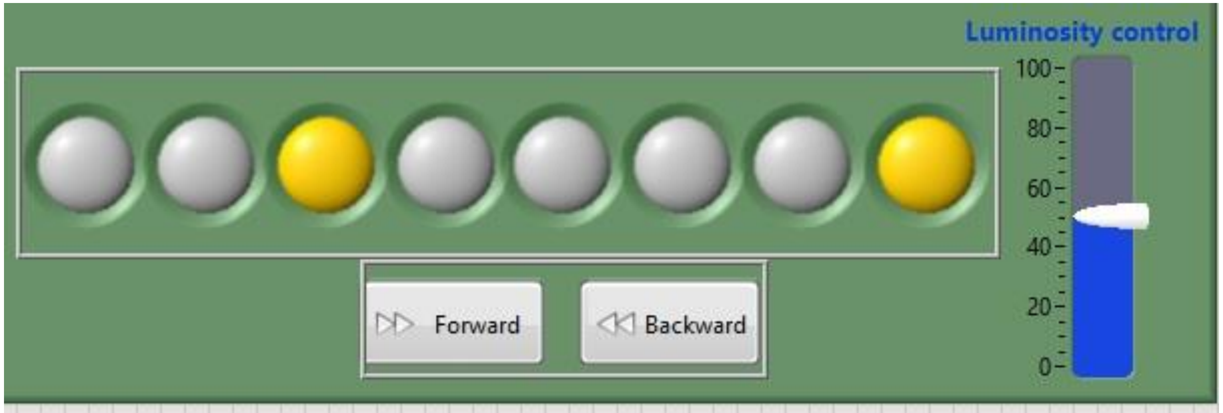


[Datasheet]


Time between two measurements > 60ms!
(corresponds measurement cycle < 16.6 Hz)
Trigger Pulse Input
Measuring Pulse (8x)
Echo Pulse Width and the distance are in proportion
Echo Pulse Output

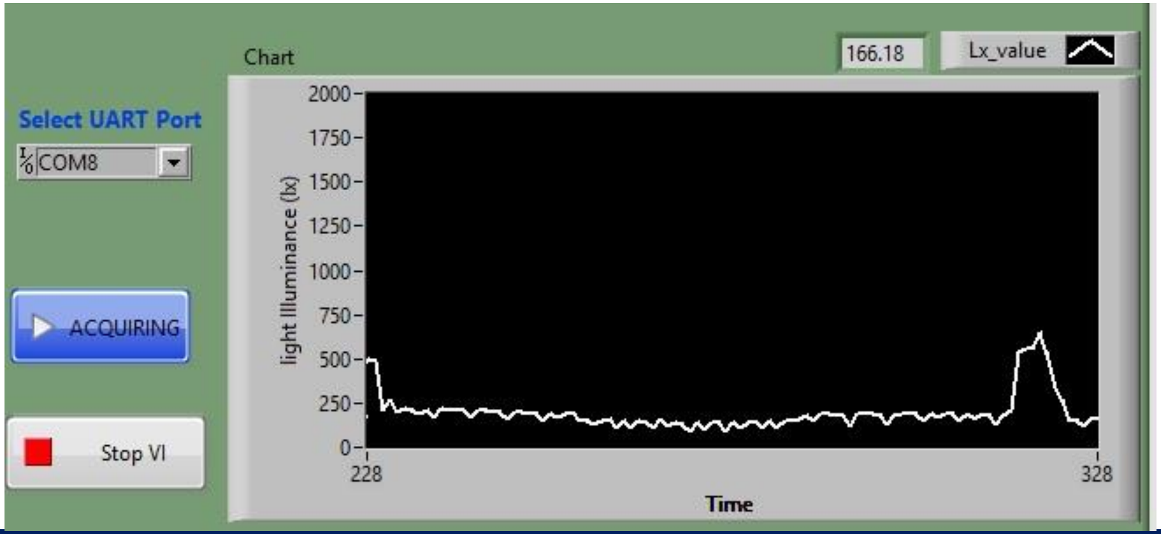
Leds Control

- Various Leds turned on and off with Arduino UNO board using commands received over UART from a LabVIEW GUI;
- Their luminosity will be also controller using PWM technique by calling the analogWrite function;
- ASCII commands are used to control the Arduino board.



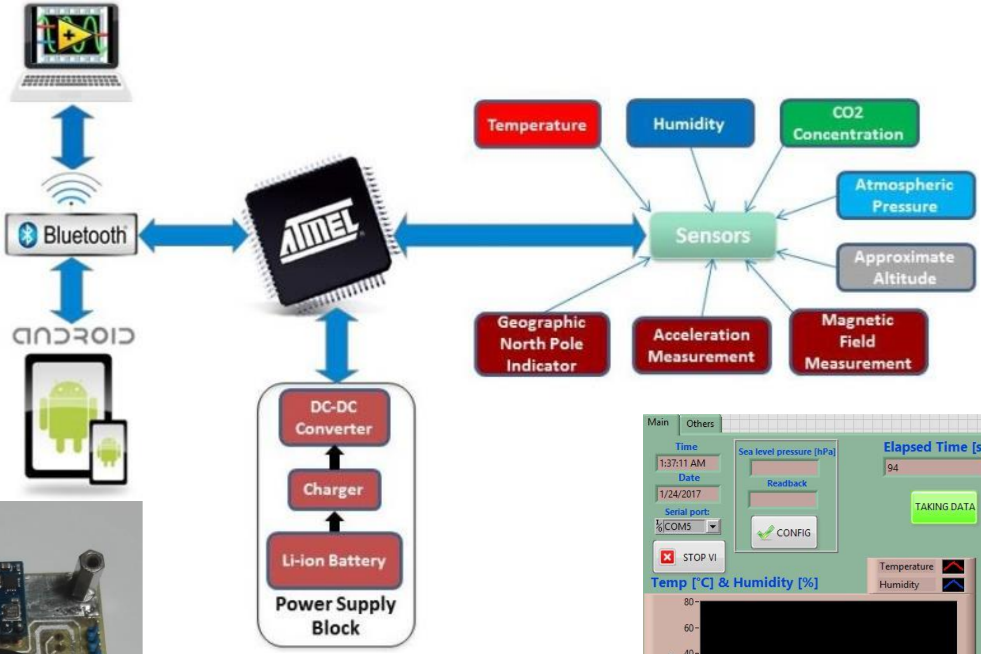
Light sensor application

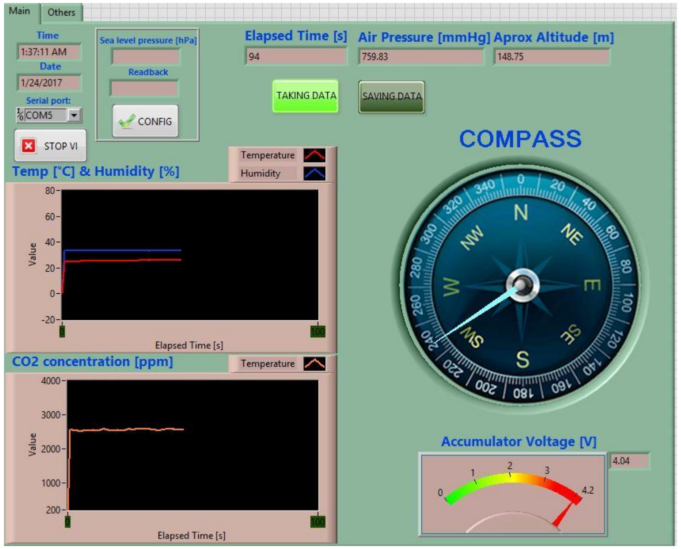
- The total luminous flux incident on a surface can be measured using light sensors, illuminance, expressed in SI unit as lux (lx);
- Ambient Light sensor TEMT6000 is used;
- TEMT6000 is a NPN phototransistor sensitive to the visible spectrum;
- The collector light current feed a 10 KΩ load resistor;
- Voltage drop across resistor can be transformed using a formula from datasheet.


Select UART Port
COM8

Datasheet

Smart, low power, wearable multi-sensor data acquisition system for environmental monitoring





Time: 1:37:11 AM

Date: 1/24/2017

Serial port: COM5

Sea level pressure [hPa]

Readback

Elapsed Time [s]: 94

Air Pressure [mmHg]: 759.83

Aprox Altitude [m]: 148.75

Temp [°C] & Humidity [%]

Temperature

Humidity

CO2 concentration [ppm]

Accumulator Voltage [V]: 4.04

- Full article at: https://doi.org/10.1109/ATEE.2017.7905059

Bibliography

1. https://www.arduino.cc/
2. https://forum.arduino.cc/
3. http://www.ladyada.net/learn/arduino/
4. https://learn.sparkfun.com/tutorials/tags/arduino?page=all
5. http://forefront.io/a/beginners-guide-to-arduino/
6. https://programmingelectronics.com/arduino-tutorials-all/
7. http://littlebits.cc/tips-tricks/introduction-to-arduino-programming-1-the-basics
8. https://www.tutorialspoint.com/arduino/arduino_overview.htm