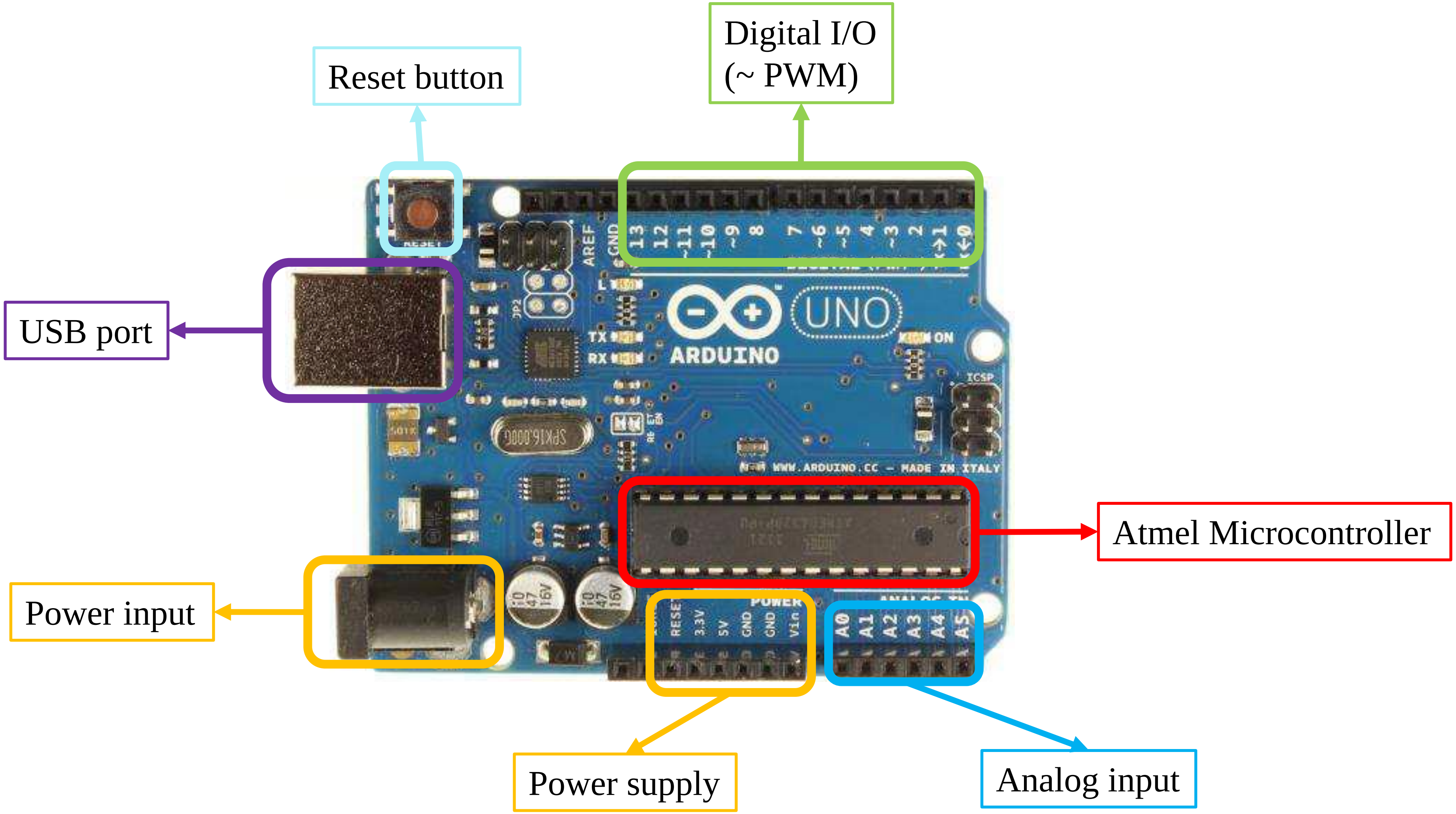
Arduino and LabVIEW

Training Project Laboratory 2020/2021

I. **What is Arduino ?**



Reset button

Digital I/O (~ PWM)

USB port

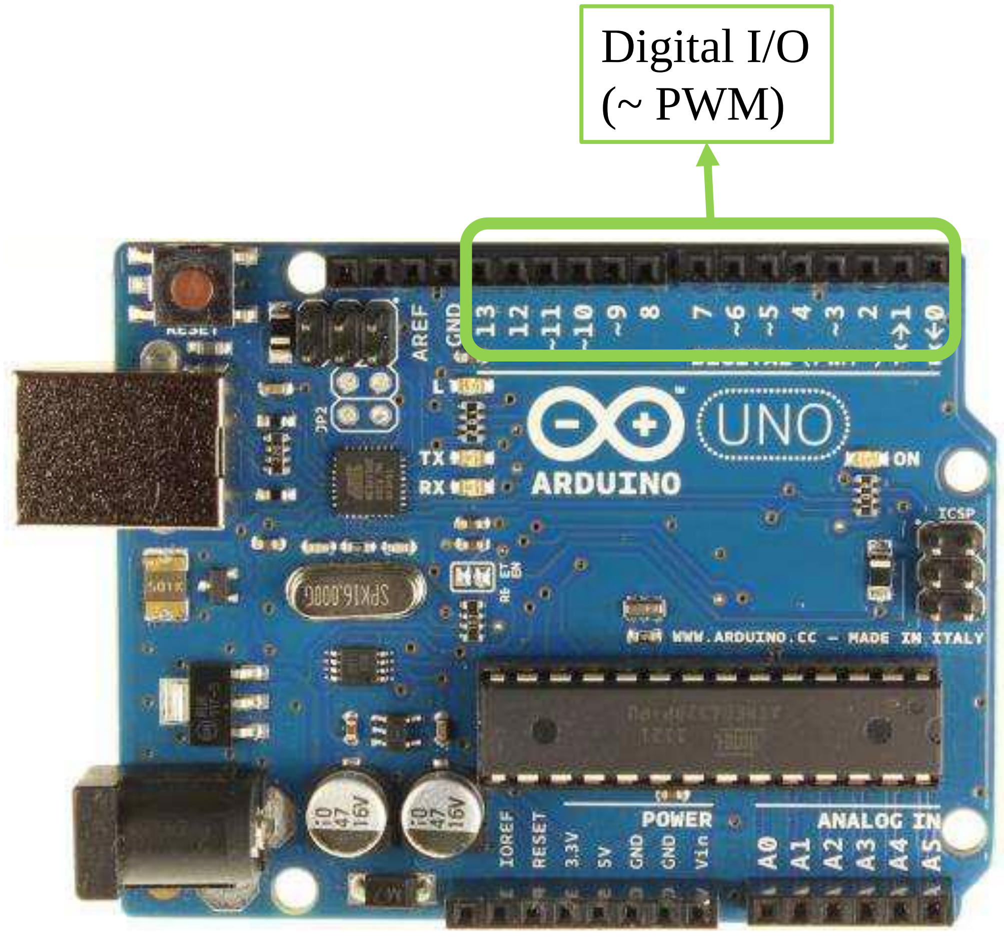
Atmel Microcontroller

Power input

Power supply

Analog input

Digital Pins:

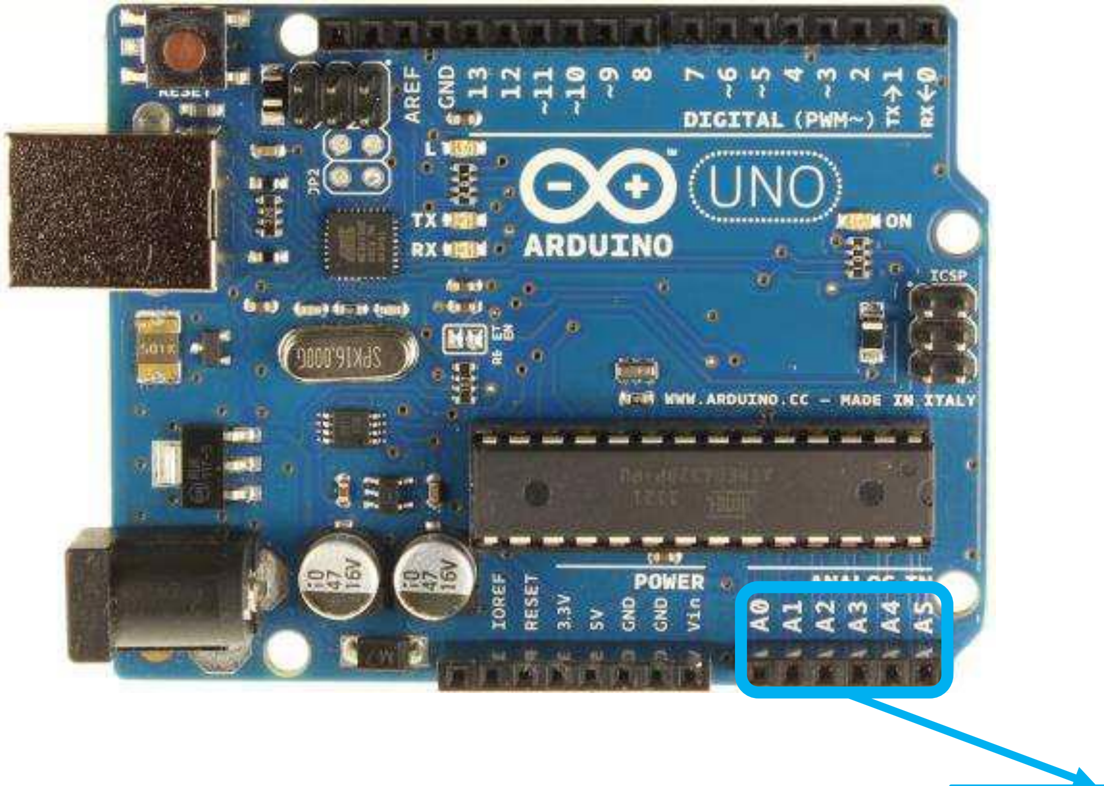


Digital I/O
(~ PWM)

- ➢ They can be ON or OFF,
- ➢ Can provide substantial amount of current to other circuits
- ➢ **digitalRead(pin)** => Reads **HIGH** or **LOW** from a pin
- ➢ **digitalWrite(pin, value)** => Writes **HIGH** or **LOW** to a pin

Analog Pins:

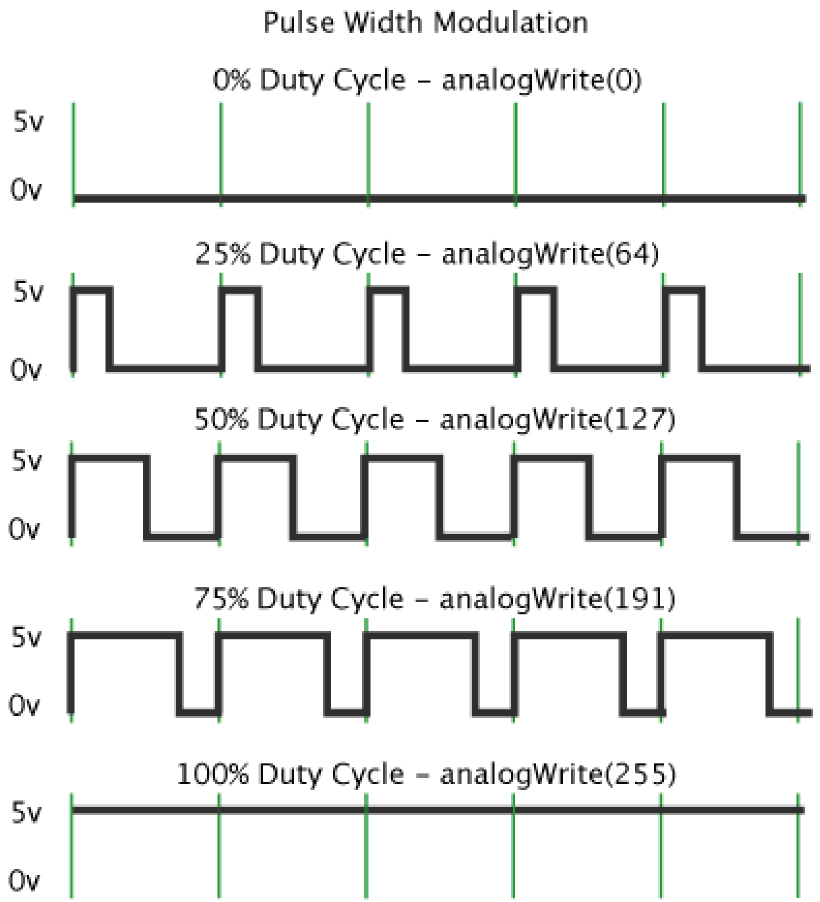
- They are used to read analog sensors such as the voltage,
- Maps input voltages between 0 and 5V into integer values between 0 and 1023 (converter has 10 bit),
- **analogRead(pin)** => Reads the value from the specified analog pin.
- **analogWrite(pin,value)** => Writes an analog value (**PWM wave**) to a digital pin.



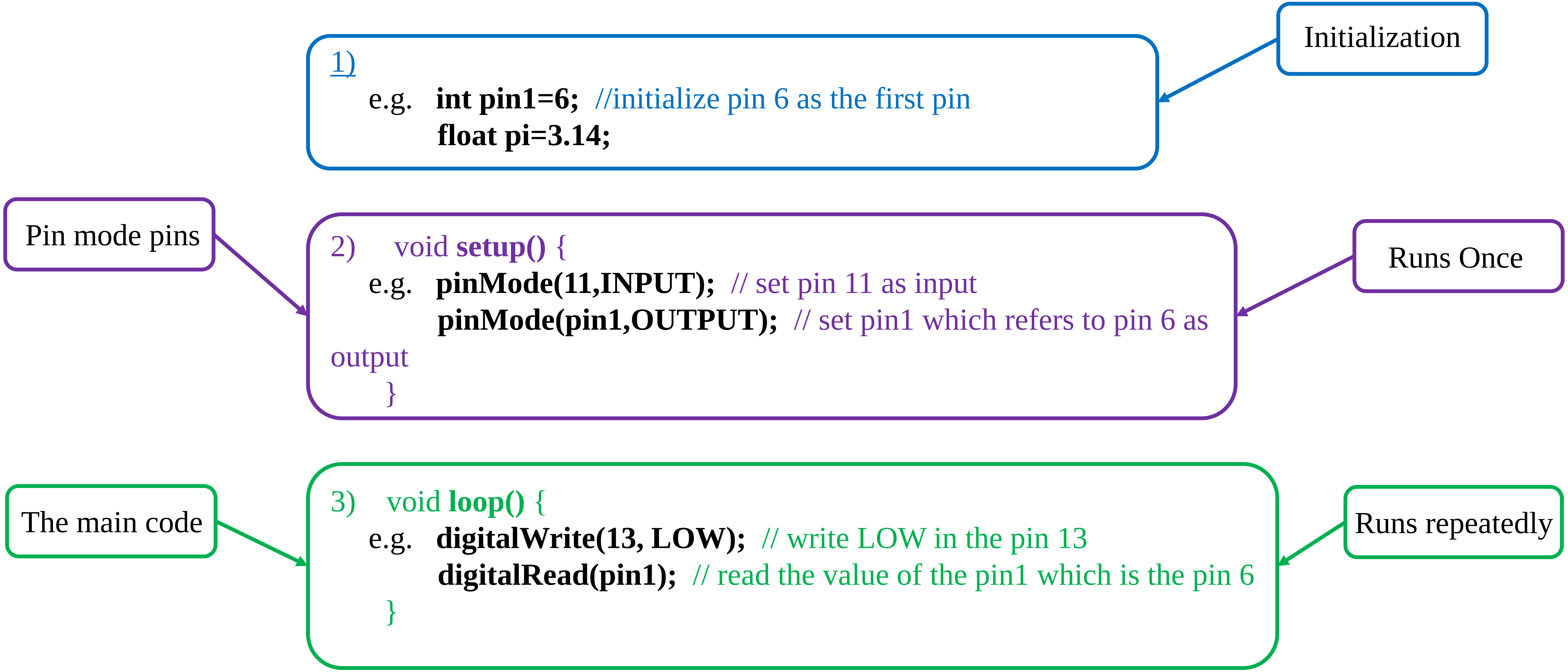
Analog input

Pulse Width Modulation (PWM)

- It is a technique for getting analog results with **digital means**,
- Digital control is used to create a square wave, a signal switched between on and off
- The on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off
- A call to **analogWrite()** is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on)
- Digital outputs 2-11 of the Arduino Mega board support PWM
- Devices like LEDs or DC motors can be driven by PWM


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
25% Duty Cycle – analogWrite(64)
50% Duty Cycle – analogWrite(127)
75% Duty Cycle – analogWrite(191)
100% Duty Cycle – analogWrite(255)
5v
0v

Arduino: Code structure



Initialization

```
1)
     e.g.   int pin1=6;  //initialize pin 6 as the first pin
            float pi=3.14;
```

Pin mode pins

Runs Once

```
2)    void setup() {
     e.g.   pinMode(11,INPUT);  // set pin 11 as input
            pinMode(pin1,OUTPUT);  // set pin1 which refers to pin 6 as
output
      }
```

The main code

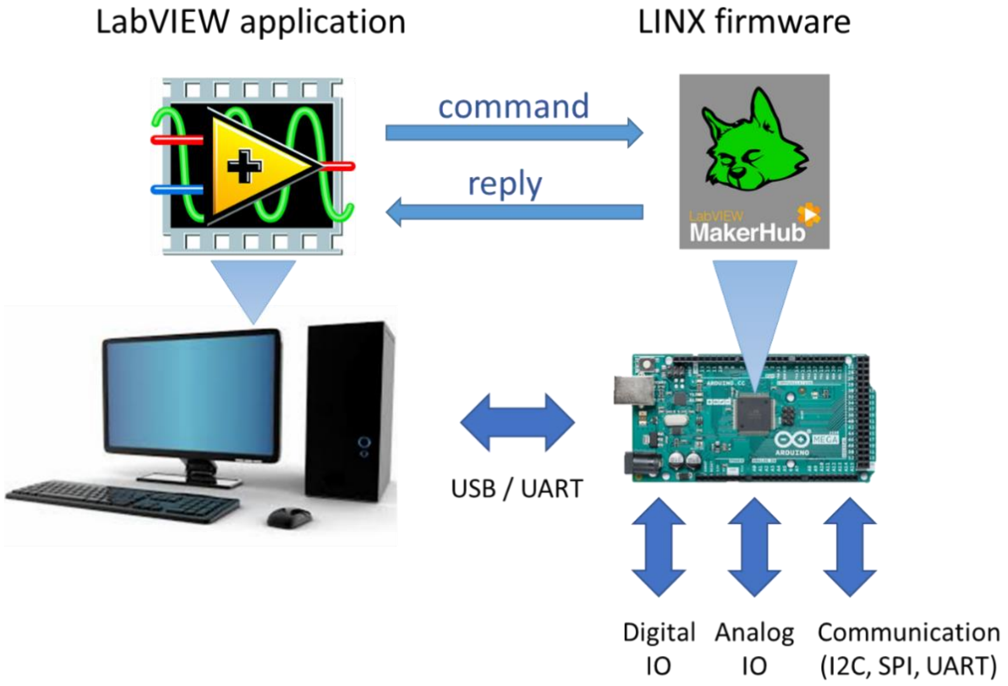
Runs repeatedly

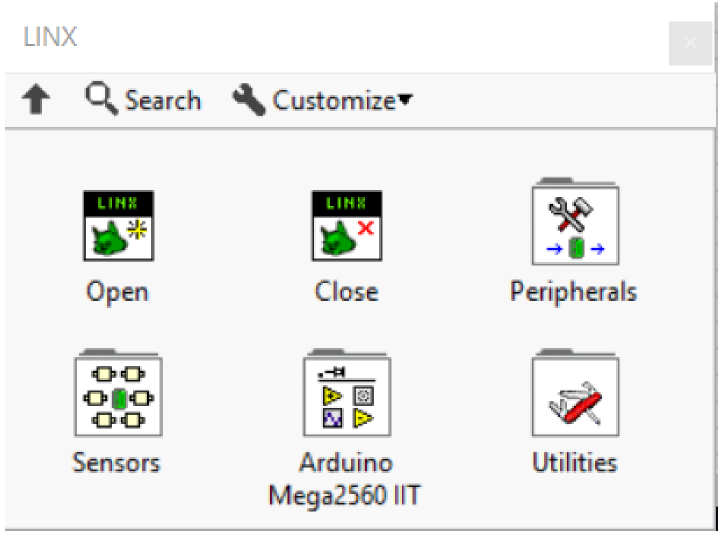
```
3)   void loop() {
     e.g.   digitalWrite(13, LOW);  // write LOW in the pin 13
            digitalRead(pin1);  // read the value of the pin1 which is the pin 6
      }
```

II. LabVIEW and Arduino

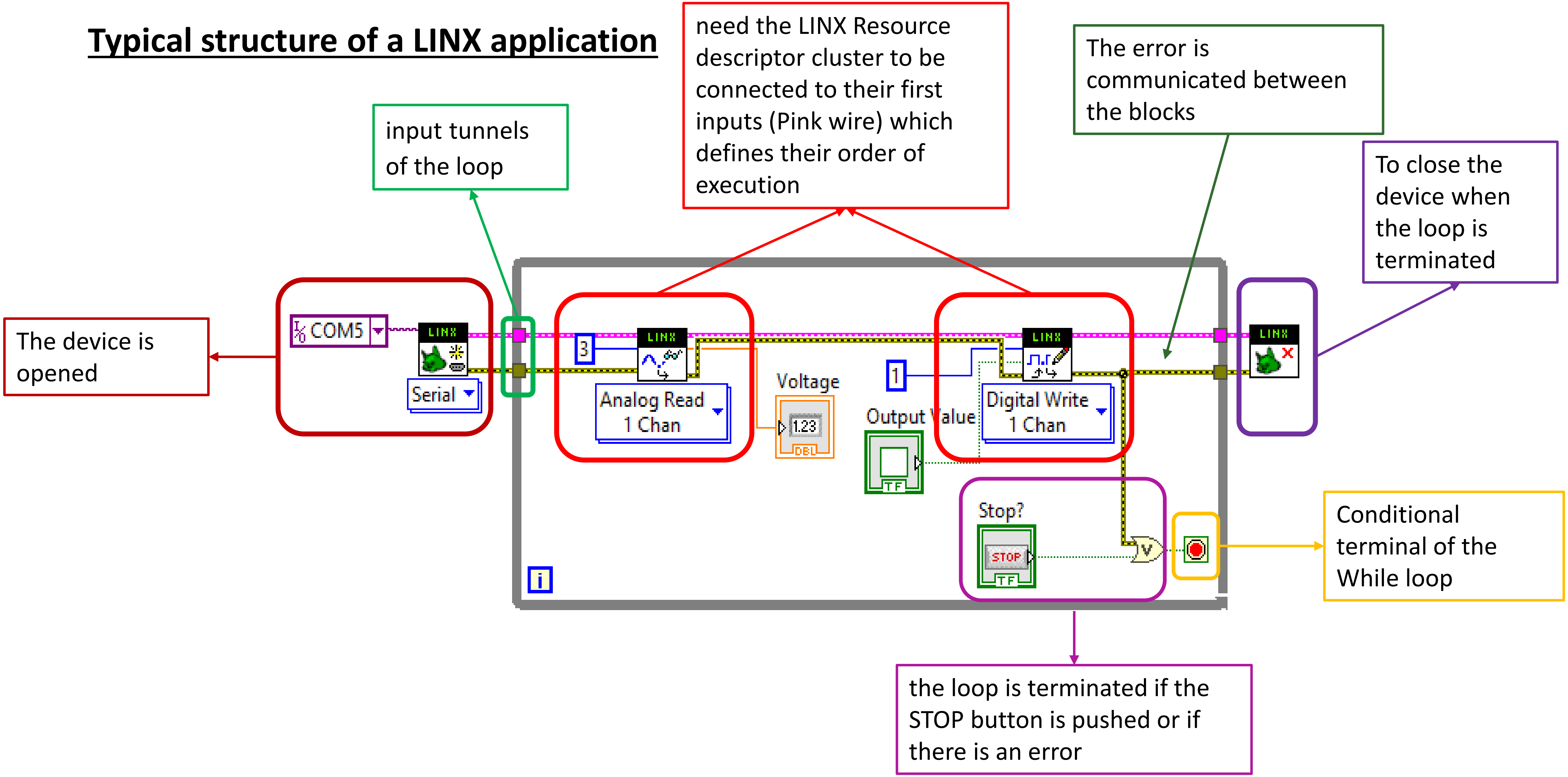
- The application is designed in LabVIEW and communicates with Arduino through the LINX firmware,

- Arduino executes after receiving commands sent by the PC





Typical structure of a LINX application



The device is opened

input tunnels of the loop

need the LINX Resource descriptor cluster to be connected to their first inputs (Pink wire) which defines their order of execution

The error is communicated between the blocks

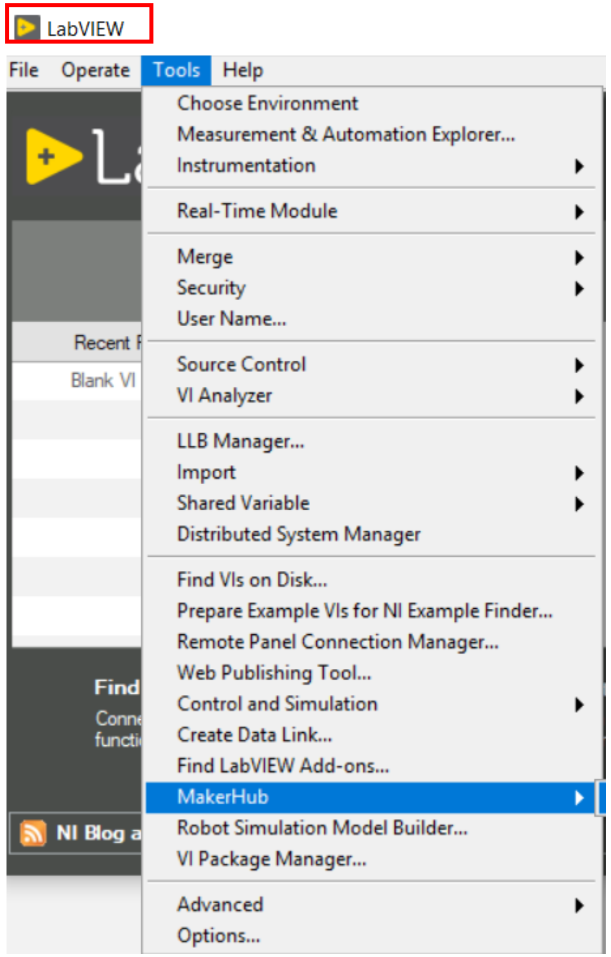
To close the device when the loop is terminated

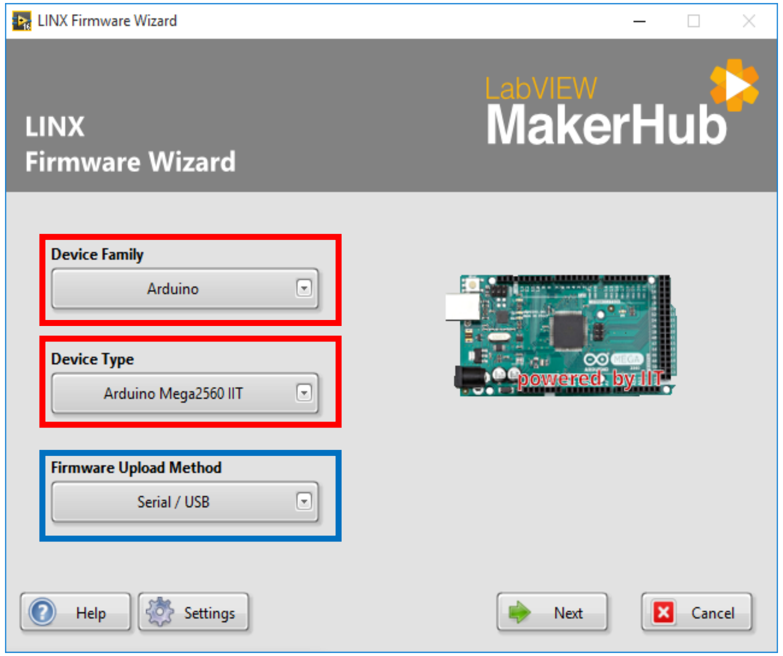
Conditional terminal of the While loop

the loop is terminated if the STOP button is pushed or if there is an error

Firmware Upload:

✓ In order to use LINX, the firmware needs to be uploaded to the Arduino board as follow:
Tools> MakerHub > Linx > Linx Firmware Wizard

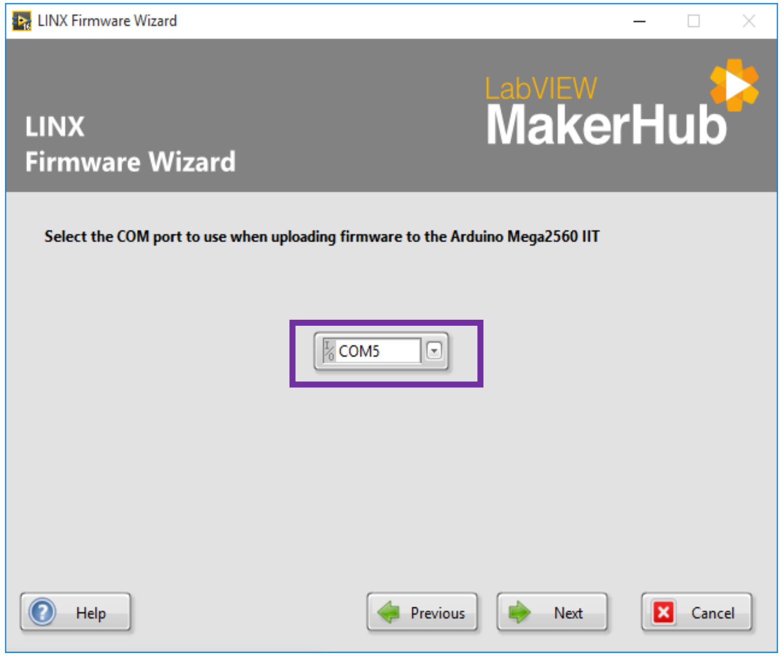




Device Family: Arduino

Device Type: Arduino Mega2560 IIT

Firmware Upload Method: Serial / USB



Select the COM port to use when uploading firmware to the Arduino Mega2560 IIT

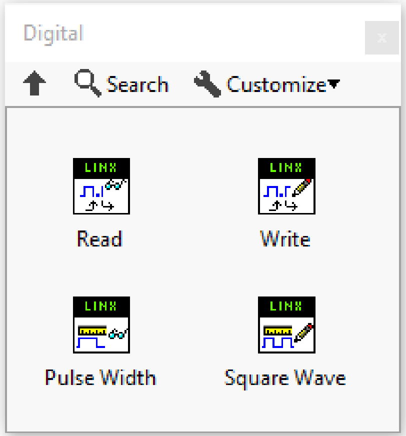
COM5

PS: Arduino Mega supports only serial communication, you shall not change the **Serial / USB** setting of the Firmware Upload Method option.

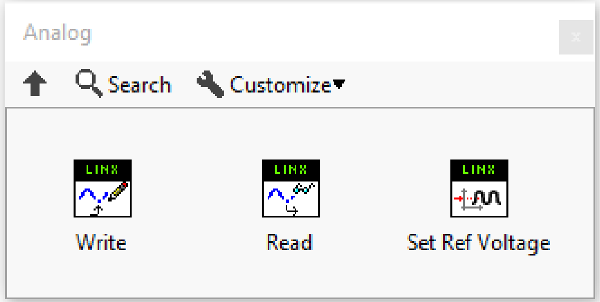
Connect the Arduino and Select the COM port where your Arduino is connected to.

LINX Toolkit Functions

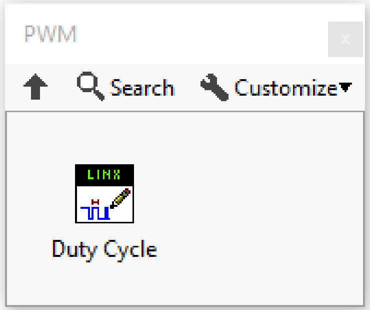
Digital tools



Analog tools



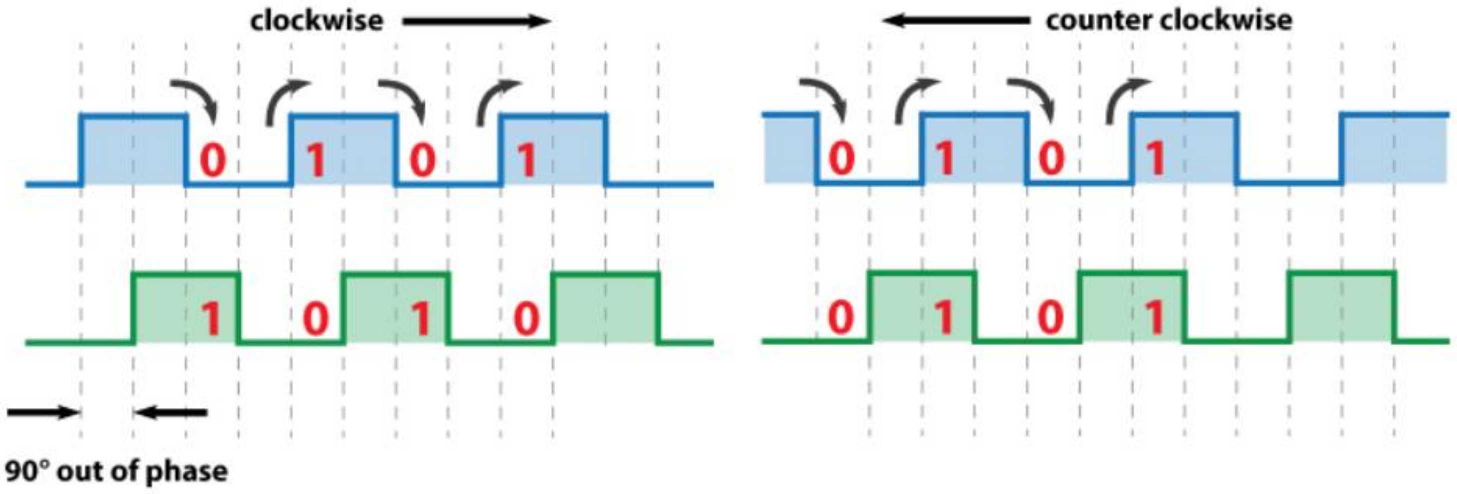
PWM tools

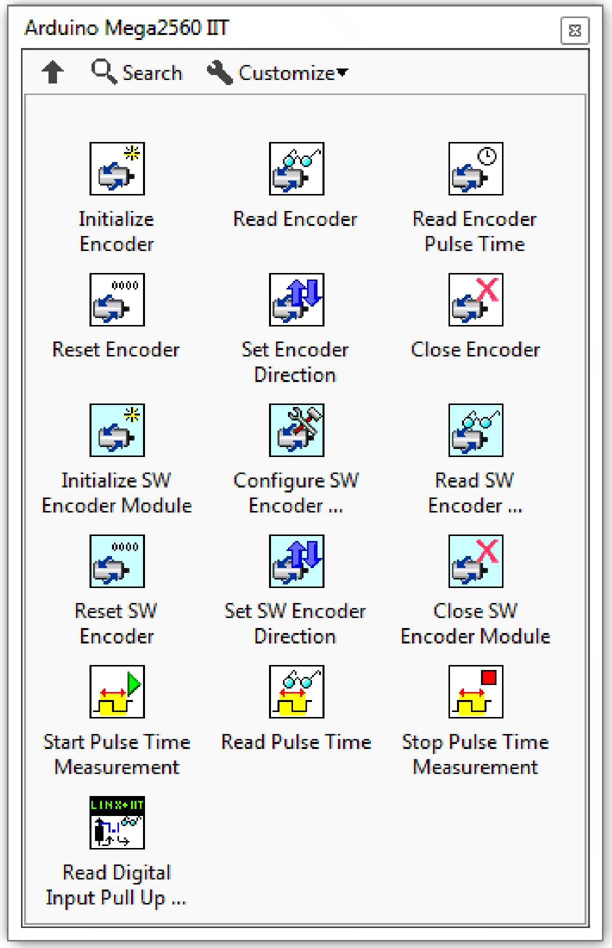


Encoder support

- Used for displacement and speed measurements **MakerHub > LINX > Arduino Mega 2560 IIT**,

- incremental encoder is a linear or rotary electromechanical device that has two output signals, A and B, which issue pulses when the device is moved. Together, the A and B signals indicate both the occurrence and direction of movement,





Example

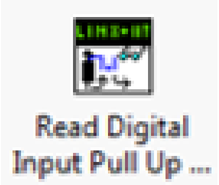
- Create an application in LabVIEW to turn on each LED with a separate button, only one LED should be on at the same time. Indicate which LED is on in the interface with the same color.

Peripheral	Pin
Red LED	6
Green LED	8
Blue LED	7

Task 1: Digital IOs

- Develop a LabVIEW application which lights the onboard red LED when Button 1 is pressed.

PS: Pushbutton is used with the pull-up resistor.



Peripheral	Pin
Pushbutton BTN1	22
Pushbutton BTN2	23
Red LED	6
Green LED	7
Blue LED	8
Trimmer potentiometer	A3
Encoder channel A	A8
Encoder channel B	A9
H-bridge PWM input	4
H-bridge input A	26
H-bridge input B	27

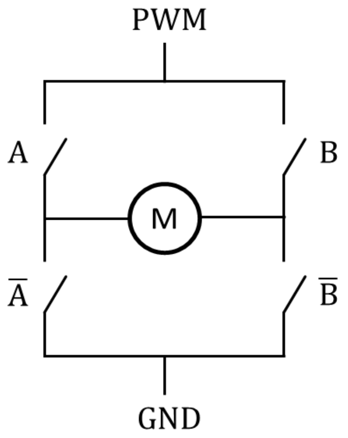
Task 2: Analog input and pulse width modulation

- Develop a LabVIEW application which displays the voltage of the analog channel connected to a trimmer potentiometer! Use a gauge indicator to display the actual voltage (0…5V) and a waveform chart to plot the change of voltage vs time.
- Add a slider to the front panel to set the luminance of the onboard red LED using pulse width modulation!

Peripheral	Pin
Pushbutton BTN1	22
Red LED	6
Green LED	7
Blue LED	8
Trimmer potentiometer	A3
Encoder channel A	A8
Encoder channel B	A9
H-bridge PWM input	4
H-bridge input A	26
H-bridge input B	27

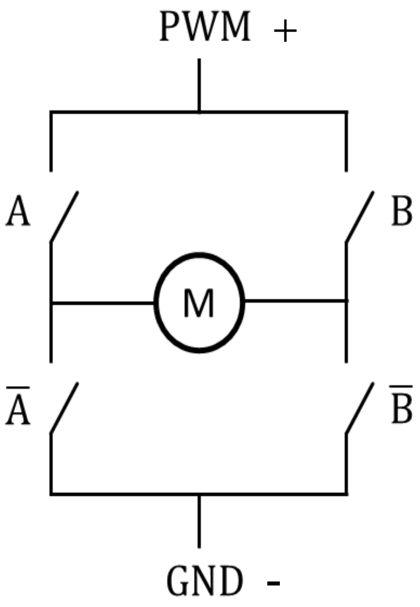
Task 4: DC motor drive

The switches, realized by power transistors, are closed according to the states of digital inputs A and B of the bridge circuit. The voltage applied to the motor can be set by the PWM input while the direction of the rotation depends on the signals A and B according to the table below. The DC motor is also equipped with an incremental encoder connected to encoder Channel 0.

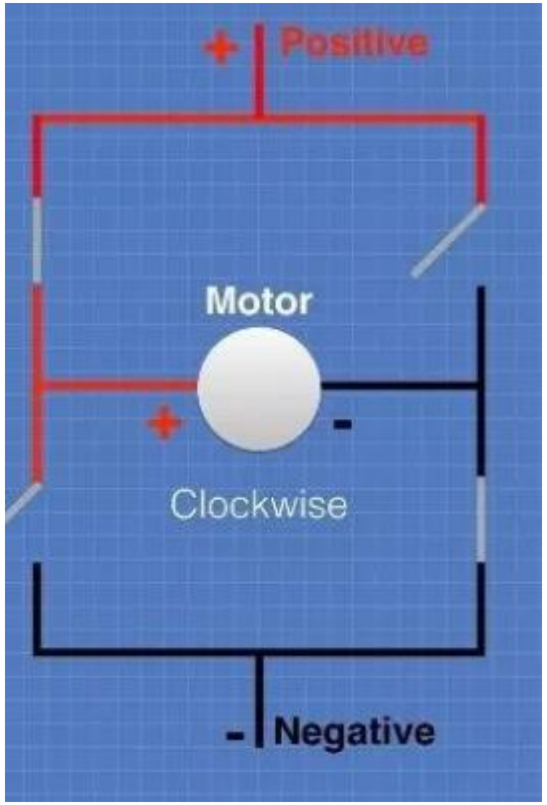


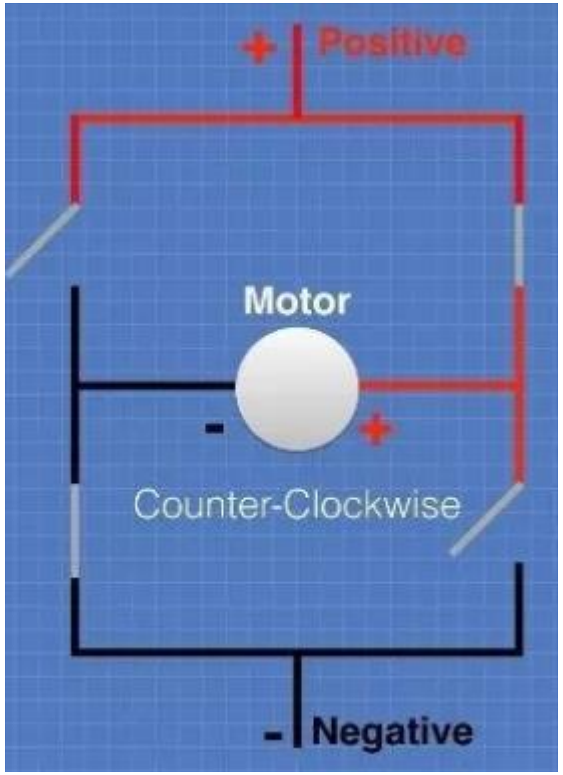
A	B	Operation
1	1	Freewheeling
0	1	Motor turns in CCW direction
1	0	Motor turns in CW direction
0	0	Brake

- Define a control which takes a floating point percentage value between -100 and +100 and drives the motor with a voltage proportional to the absolute value and a direction corresponding to the sign of the input!
- Measure the speed of the motor in RPM (revolution per minute)! (30 pulses per cycle)
- Develop an application which allows the user to set the voltage of the motor using a slider and displays the pulse counts.



the H-bridge circuit





(0=close, 1=open).

Peripheral	Pin
H-bridge PWM input	4
H-bridge input A	26
H-bridge input B	27

A	B	Operation
1	1	**Freewheeling**
0	1	**Motor turns in CCW direction**
1	0	**Motor turns in CW direction**
0	0	**Brake**