ATMEGA328/Arduino Uno – I/O Pins
https://github.com/teaksoon/lmaewapm

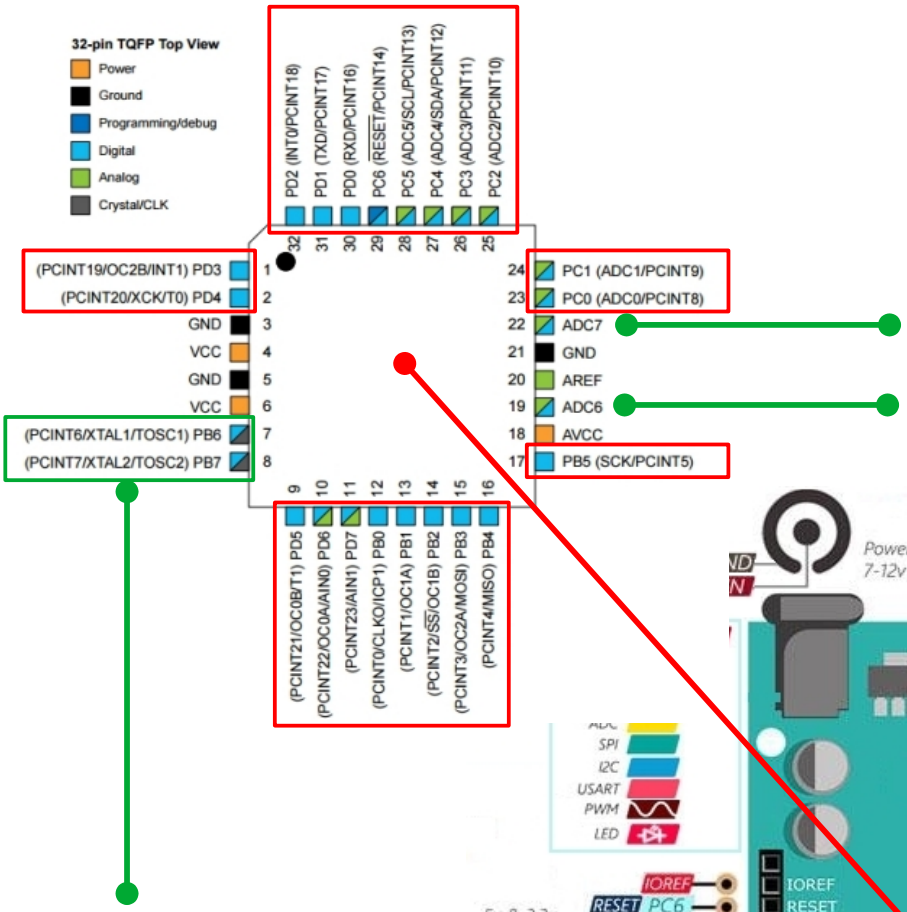
Apart from the **Power Supply Pins (GND, VCC, AVcc)**, the ATMEGA328P micro-controller have many other Pins coming out from its physical chip packaging. Those Pins are known as **INPUT/OUTPUT or I/O Pins**

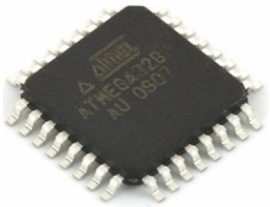
Each of the ATMEGA328P I/O Pin is connected a female header on the Arduino Uno board(labelled 0 to 13 and A0 to A5), except for

- Pin PB6 and PB7, which are connected to the 16Mhz Clock Crystal
- Pin ADC6 and ADC7, which are not connected to anything on Arduino Uno Board (you have them on the Arduino Nano board)

The I/O Pins are also dealing with the basic properties of electricty: CURRENT, VOLTAGE and RESISTANCE

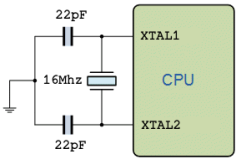
What we are interested from the I/O Pins is "VOLTAGE"

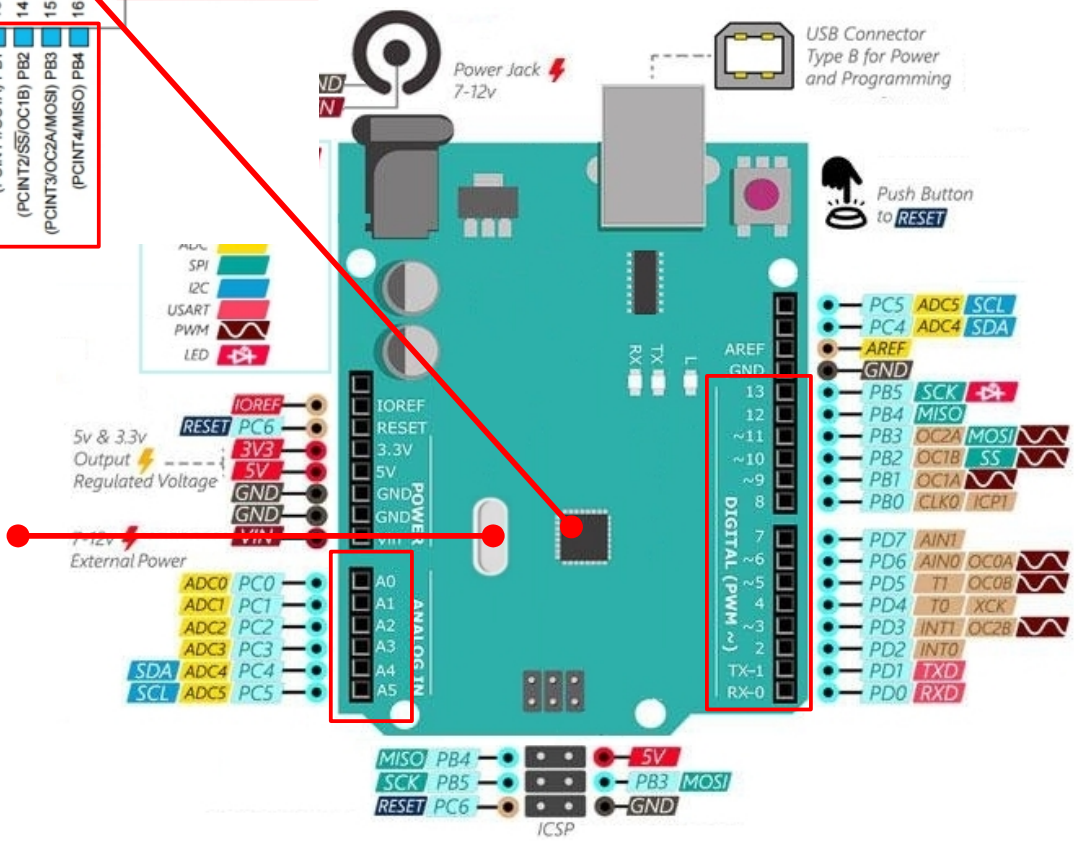




ADC6 and ADC7 are not connected to anything on the Arduino Uno board

These two are also I/O pin, however they are dedicated to the External 16Mhz Clock Crystal on Arduino Uno board





16Mhz Clock Crystal when powered up, will consistently generate “pulse” 16 million times in every one seconds. From there, we get our timed interval. This device is optional, because the ATMEGA328 micro-controller chip has an internal 8Mhz Clock Device. Since we want to run the ATMEGA328P micro-controller at 16Mhz, this external 16Mhz Clock Crystal is required

The "I/O Pins" can be set to become "INPUT PIN" or "OUTPUT PIN"
We can do it by an instruction from our PROGRAM to the micro-controller

When I/O Pin is set as "INPUT PIN"

- This Pin will **"receive" VOLTAGE** from the connected external device

- The **VOLTAGE "received"** from the external device can **range from 0V to 5V**

Our PROGRAM can ask the micro-controller for the VOLTAGE received from the external device connected to this Pin

Since the VOLTAGE is coming from the external device, we know what is happening in the connected external device

Our PROGRAM can then give more instructions to the micro-controller to respond to the connected external device activities

When I/O Pin is set as "OUTPUT PIN"

- This Pin will **"supply" VOLTAGE** to the connected external device

- The **VOLTAGE "supplied"** to the external device can be **either 0V or 5V**

Our PROGRAM can ask the micro-controller to supply either 0V or 5V to this Pin, to be "picked-up" by the external device

We can supply the 0V or 5V in different ways,

1.Permanent – We can just leave the Pin to Supply either 0V or 5V

2.Conditional – We can have this Pin supplying either 0V or 5V based on "conditions" set in our PROGRAM

3.Timed – We can have this Pin supplying either 0V or 5V for a period of time

We can also combine the "Permanent", "Timed" and "Conditional" in our PROGRAM to do perform more complicated task

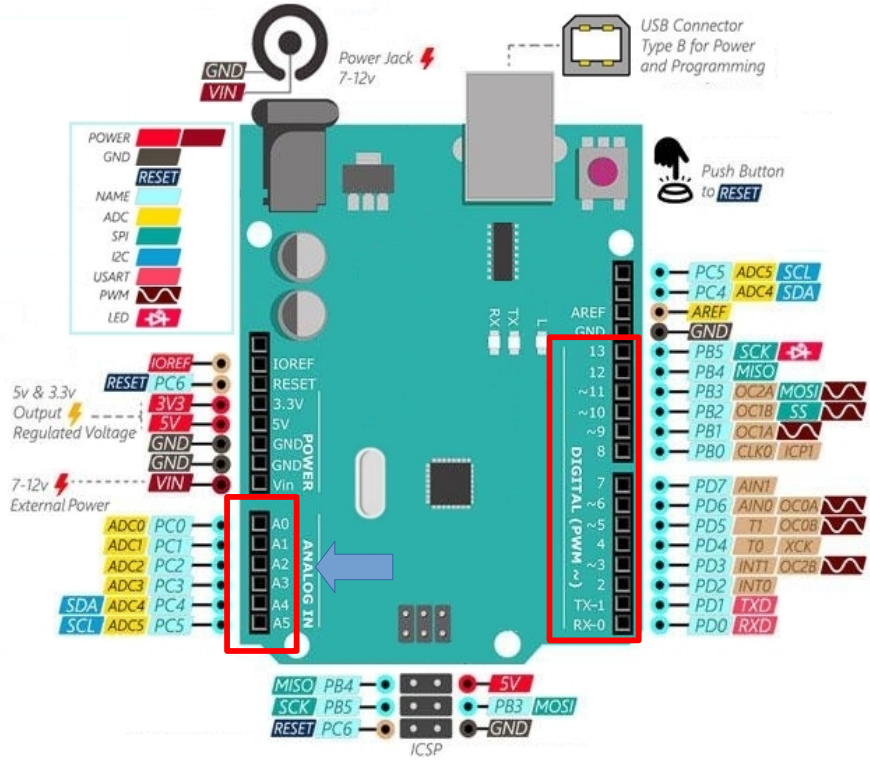
We have probably heard things like Robotics, Machine Automation, Artifical Intelligence, IOT, Industrial 4.0, Drones, Car Electronics, Driverless Tecnology, Aerospace Technology and so many other fanciful names,...

Soon you will discover that,

Bulk of them are just some creative way of supplying and reading of "VOLTAGE" on the I/O Pins

After our PROGRAM set an I/O Pin to become an INPUT PIN

Our PROGRAM can ask the micro-controller to
Read VOLTAGE from the INPUT PIN



There are two types of VOLTAGE reading that our PROGRAM can do
DIGITAL READING and ANALOG READING

DIGITAL READING – We can do DIGITAL READING from all the I/O Pins (A0 to A5, 0 to 13) when set as INPUT PIN

– if the VOLTAGE is 3V to 5V, DIGITAL READING will be HIGH or 1
– if the VOLTAGE is below 3V, DIGITAL READING will be LOW or 0

We use DIGITAL READING from INPUT I/O Pin when we only need two distinct state from the device, HIGH VOLTAGE or LOW VOLTAGE

ANALOG READING – We can only do ANALOG READING from Arduino Uno Pin A0,A1,A2,A3,A4 and A5 when set as INPUT PIN

They are also known as "10-BIT ADC Pins". The micro-controller ADC will converts the ACTUAL VOLTAGE on those Pin to a "10-BIT Whole Number" automatically for us

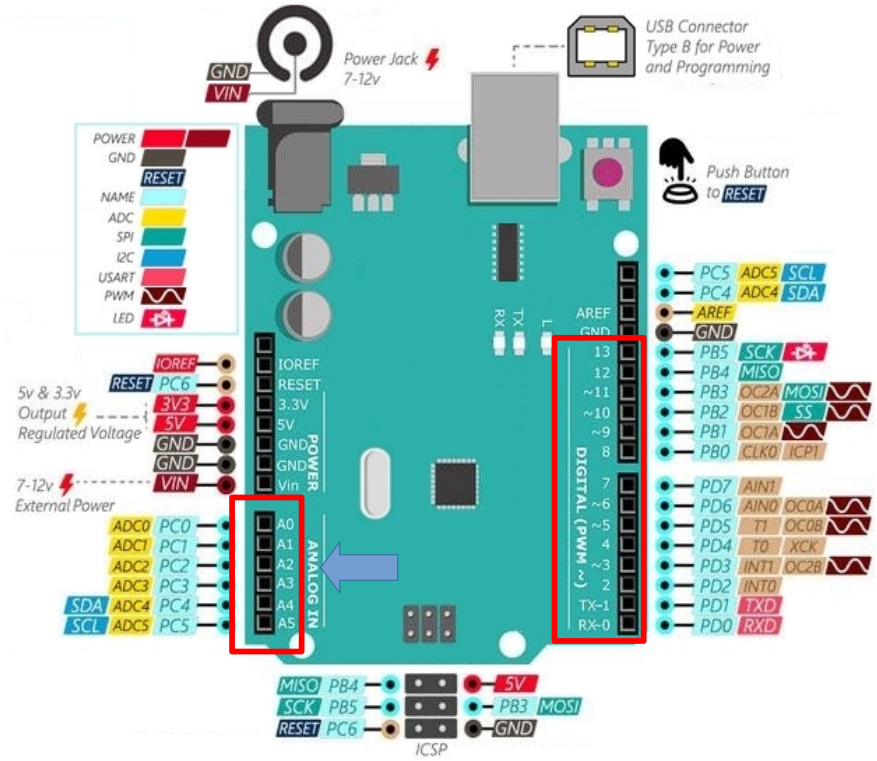
– if the VOLTAGE is 0V, ANALOG READING will be 0
– if the VOLTAGE is 5V, ANALOG READING will be 1023 (why 1023? The maximum value for 10-BIT in binary is 1111111111, which is 1023 in decimal)

From the above, we can calculate the rest,
How about 2V ? 2V ANALOG READING will be (2*1023)/5 = 409
How about 3V ? 3V ANALOG READING will be (3*1023)/5 = 614
and so on...

We use ANALOG READING from INPUT I/O Pin (A0 to A5 only) when we need the actual VOLTAGE from the device, for example: the variable VOLTAGE from Analog Sensors (different VOLTAGE tells us a different story)

After our PROGRAM set an I/O Pin to become an OUTPUT PIN

Our PROGRAM can ask the micro-controller to
Supply VOLTAGE to the OUTPUT PIN



There are two types of VOLTAGE supply that our PROGRAM can do
DIGITAL and PWM

DIGITAL – We can supply 0V or 5V to all the I/O Pins (A0 to A5, 0 to 13) when set as OUTPUT PIN

– if the supplied VOLTAGE is 5V, it is DIGITAL HIGH or 1
– if the supplied VOLTAGE is 0V, it is DIGITAL LOW or 0

The OUTPUT Pin will supply exactly 0V or 5V. To supply other voltage, we will need extra hardware devices

PWM – We can only do "PWM Power Supply" to Arduino Uno Pin ~3,~5,~6,~9,~10 and ~11 when set as OUTPUT PIN

The micro-controller has something called "PWM" or Pulse Wave Modulation on those Pins. Which will automatically alternate between "0V" and "5V" at a specific pattern for the connected device

These pin are able to alternate between 0V and 5V with a specific pattern based on something called "Duty Cycle"

100% Duty Cycle, alternate 5V,5V,5V, PWM Value = 255 (8-BIT, 11111111 = 255)
0% Duty Cycle, alternate 0V,0V,0V, PWM Value = 0
50% Duty Cycle, alternate 0V,5V,0V,5V, PWM Value (255-0)/2 = 127
25% Duty Cycle, alternate 0V,0V,5V,0V,0V,5V, PWM Value 0+(127/2) = 63
75% Duty Cycle, alternate 0V,5V,5V,0V,5V,5V, PWM Value 127+(127/2) = 190

The PWM signal can be sent to the the external device. When the external device receive this kind of VOLTAGE pattern, the device will response accordingly