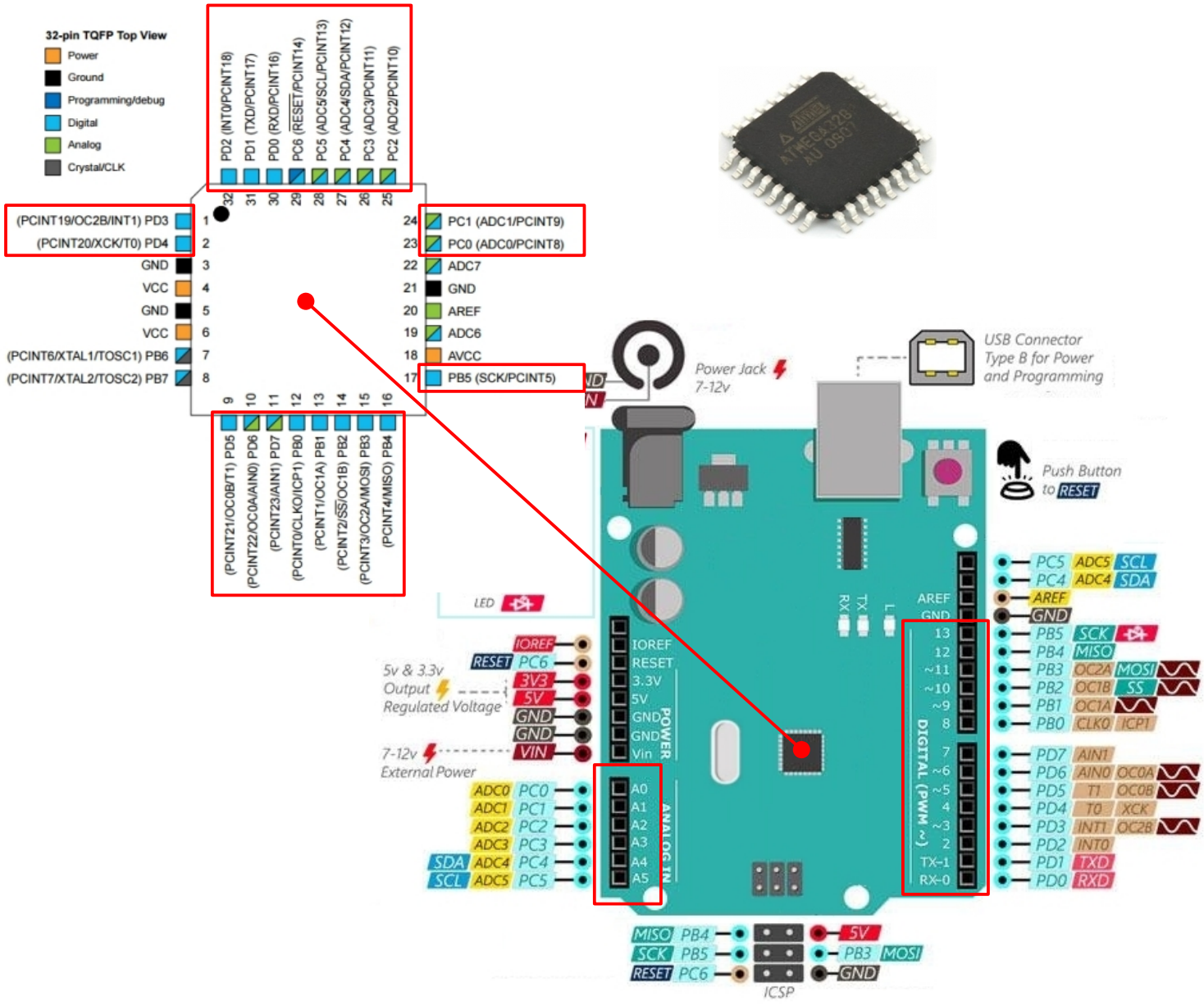
ATMEGA328/Arduino Uno – I/O Pins – INPUT
https://github.com/teaksoon/lmaewapm

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 other Pins are known as **I/O Pins** connected to **Arduino Uno Board with label A0 to A5 and 0 to 13**

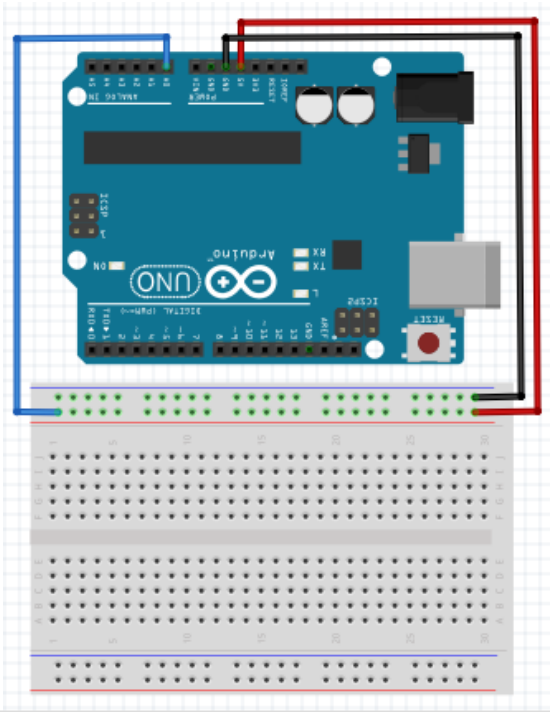


Once the I/O Pins are set to be an INPUT Pin (from our Program), our Program can now read the VOLTAGE that is currently on the INPUT I/O Pin.

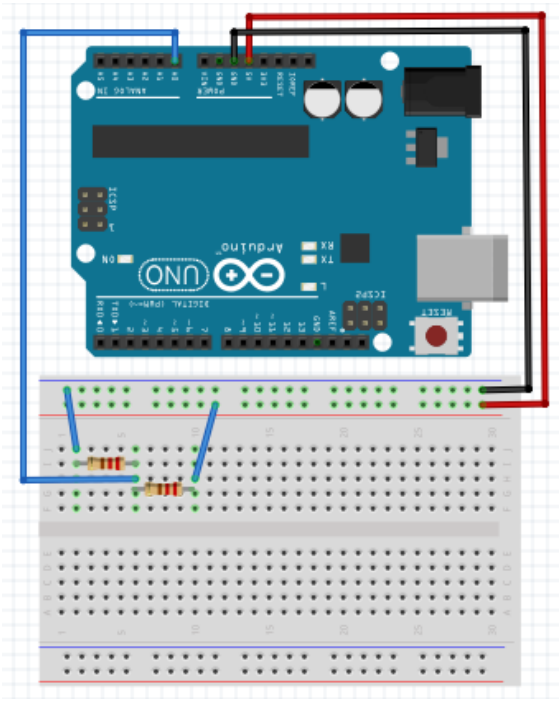
Programming "INPUT I/O Pin" is all about reading the VOLTAGE on the I/O pin from our PROGRAM

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

The INPUT I/O Pin does not automatically have Voltage in them. We need to physically feed Voltage into the INPUT I/O Pin. IF no Voltage is fed into the INPUT I/O Pin, it will just have some small random Voltage



In the circuit above, the INPUT I/O Pin A0 will receive 5V from Arduino Uno 5V Pin



In the circuit above, the INPUT I/O Pin A0 will receive about half of 5V from the Arduio Uno 5V Pin (Voltage Divider with 2 same value Resistor, from a 5V Source)

Refer to the "2021_11_23_resistor" Tutorial topic on how to calculate the Voltage from Resistor based Voltage Divider

All I/O Pins by default, is already an INPUT I/O Pin. We do not need to code pinMode() for INPUT I/O Pin. However, we can still code the pinMode() if we wish to do so

Once an I/O Pin is already an INPUT I/O Pin, we can read VOLTAGE from the INPUT I/O Pin by using these two functions,

1. digitalRead(pinNumber) function – This function **can work on any I/O Pin that is set as INPUT I/O Pin.** This function will tell is whether it is **LOW VOLTAGE (Less than 3V)** or **HIGH VOLTAGE(More than 3V).** LOW is represented by 0 and HIGH represented by 1

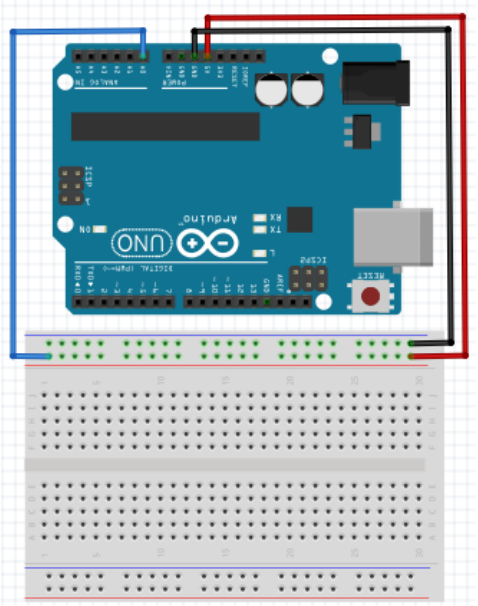
2. analogRead(pinNumber) function – This function **can only work with I/O Pins labelled A0,A1,A2,A3,A4,A5, set as INPUT I/O Pin.** These are also known as Analog Pins or ADC Pins. This function gives us the **Actual Voltage represented by a 10-bit number** (maximum value in 10-bit = 1111111111 in binary = 1023 in decimal. Therefore, a 10-bit ADC value will range from 0 to 1023)

5V on this Pin is represented by 1023
0V on this Pin is represented by 0

The VOLTAGE in-between 0V and 5V can be calculated accordingly

Example: 4V will be represented by (4x1023)/5 = 818

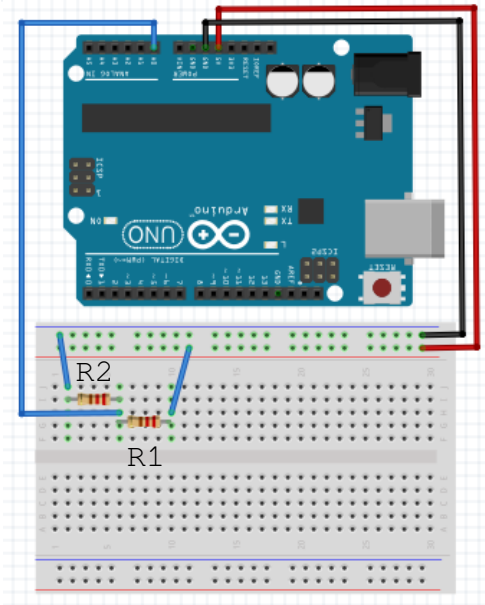
ATMEGA328/Arduino Uno - I/O Pins - INPUT
https://github.com/teaksoon/lmaewapm



Setup 1:
1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board

1x Solderless Breadboard
Jumper wires

Pin A0 is connected to Arduino Uno 5V



Setup 2:
1x Computer with Arduino IDE Software
1x USB 2.0 Type A/B Data Cable
1x Arduino Uno Board

1x Solderless Breadboard
Jumper wires
2x 10K Ohm Resistor (2 same value Resistor)

Resistor 1 is connected to 5V
Resistor 2 is connected to GND

In-Between Resistor 1 and Resistor 2, make connection to Pin A0 (This is called a Voltage Divider, we want to split 5V before sending it to A0)

Program: io_pin_input_read_voltage (use this same program both setup)

```
void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int pinDigital = digitalRead(A0);
  int pinAnalog  = analogRead(A0);

  Serial.print("\nINPUT I/O Pin A0: ");
  Serial.print("Digital Value=");Serial.print(pinDigital);
  Serial.print(", Analog Value=");Serial.print(pinAnalog);
  Serial.print(", Voltage=");Serial.print( (float) (pinAnalog*5)/1023,2 );
  delay(500);
}
```

Open the Serial Monitor from the Arduino IDE Software and watch the Output on the serial monitor screen. Observe the Voltage Reading. While still watching the Serial Monitor Screen do the following,
for Setup 1: Remove A0 Pin connection to 5V (A0 is free floating)
for Setup 1: Connect A0 Pin to Arduino Uno GND, instead of 5V Pin

for Setup 2: Remove Resistor R2 (no more Voltage Divider, A0 gets 5V)