Programming of Arduino UNO, PWM pins & Controlling Arduino Uno’s PWM pins using ADC

Date: 19-08-2022

1. We know that, the Arduino UNO board comes with 6 PWM (Pulse Width Modulation) pins.
2. It is pins 3,5,6,9,10,11 respectively
3. Also learned earlier that, we use function **analogWrite(PinNo,Value)**
4. An example of “Fade” present in Arduino IDE Example Sketch was already been worked out and it is given below for practice.

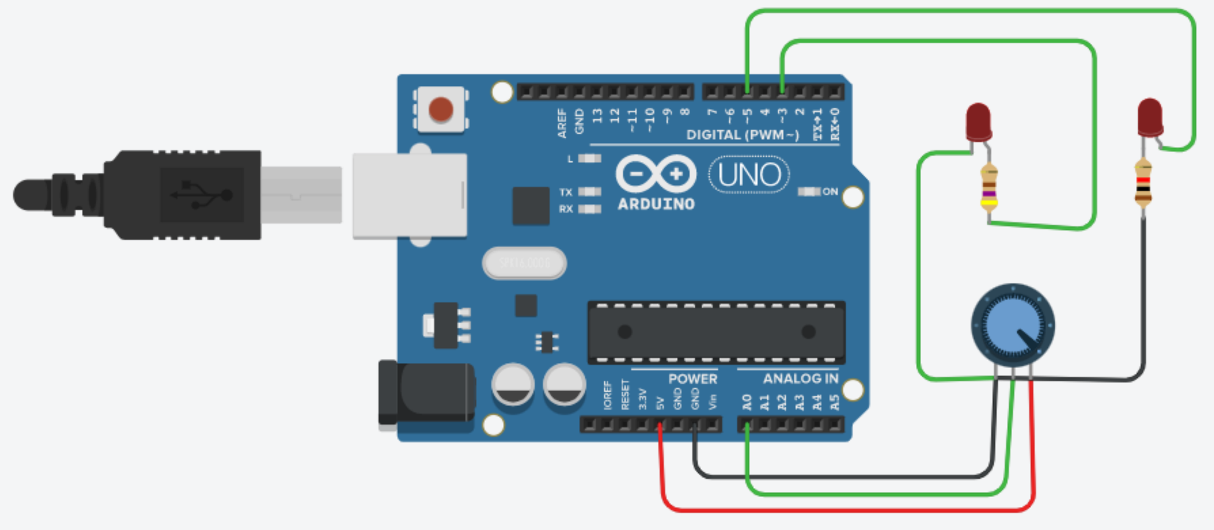
```
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
/*
  Fade

  This example shows how to fade an LED on pin 9 using the analogWrite()
  function.

  The analogWrite() function uses PWM, so if you want to change the pin you're
  using, be sure to use another PWM capable pin. On most Arduino, the PWM pins
  are identified with a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Fade
*/

int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;    // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);
```

```
  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

5. Now after learning the concept of PWM pins and its capabilities, now we can go forward in controlling these pins using the ADC output. This type of controlling is really required in IoT and associated applications.
6. In this session, we use the already learned functions namely, analogRead() and analogWrite() and control the PWM pins.
7. These concepts find applications in controlling the boiler/furnace temperatures and to control various similar parameters.
8. Following is the Arduino Sketch and circuit, which controls the PWM pins based on the ADC output.

```
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
int sensorPin = A0;  // input pin for the potentiometer
int digitalValue = 0;// variable to store the value coming from the sensor
int y =0;

void setup() {
  Serial.begin(9600);
}

void loop() {
  digitalValue = analogRead(sensorPin);// read the value from the analog channel
  Serial.print("digital value = ");
  Serial.println(digitalValue);        //print digital value on serial monitor
  delay(1000);

  if (digitalValue>=0 && digitalValue <=511)
  {
        analogWrite(3,0);
  }
```

```
else if (digitalValue>=512 && digitalValue <=1023)
{
        analogWrite(3,255);
}

y = map(digitalValue, 0, 1023, 0, 255);

analogWrite(5,y);

}
//~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▸1
RX◂0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5