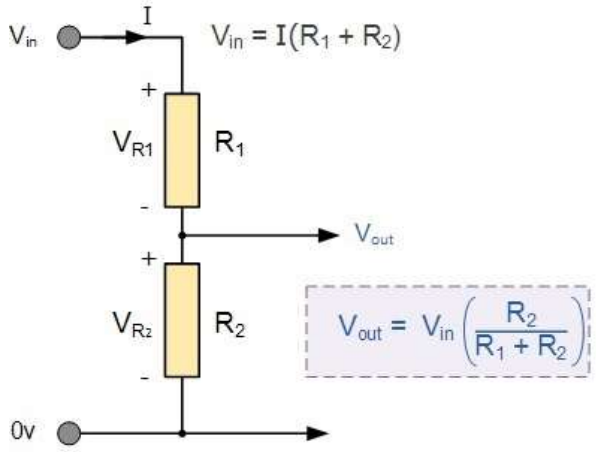
United International University
QUEST FOR EXCELLENCE

CSE 426 Experiment 9: Analog input-output

Objective:
• To understand the analog to digital conversion and digital to analog conversion.
• How to use analog input-output using Arduino uno board

Equipment needed:
• Arduino Uno board
• Arduino IDE (Compiler)
• Proteus (Simulator)

Voltage divider circuit:



The circuit shown consists of just two resistors, R_1 and R_2 connected together in series across the supply voltage V_{in}. One side of the power supply voltage is connected to resistor, R_1, and the voltage output, V_{out} is taken from across resistor R_2. The value of this output voltage is given by the corresponding formula.

Reading Sensors

Many sensors in the real world are simple resistive devices. A photocell is a variable resistor, which produces a resistance proportional to the amount of light it senses. Other devices like flex sensors, force-sensitive resistors, and thermistors, are also variable resistors.

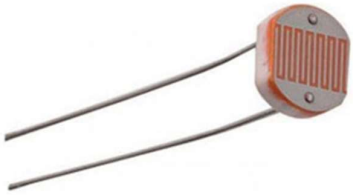
There are two types of sensors: Analog sensor and Digital sensor.

- Analog Sensors
  There are different types of sensors that produce continuous analog output signal and these sensors are considered as analog sensors. Practical examples of various types of analog sensors are as follows: accelerometers, pressure sensors, light sensors, sound sensors, temperature sensors, and so on.

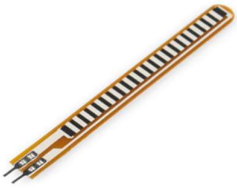
- Pros
  - Usually simple sensor design

- Cons
  - Analog signal produced by the analog sensor could be distorted during long distance transmission.

Thermistor (temperature sensor)



LDR (Light sensor)



Flex sensor

- Digital Sensors

  Electronic sensors or electrochemical sensors in which data conversion and data transmission takes place digitally are called as digital sensors. These digital sensors are replacing analog sensors as they are capable of overcoming the drawbacks of analog sensors. The digital sensor consists of majorly three components: senor, cable, and transmitter. In digital sensors, the signal measured is directly converted into digital signal output inside the digital sensor itself then this digital signal is transmitted through cable digitally.

  - Pros
    - Digital signal produced by the digital sensor could be transmitted long distance.
  - Cons
    - Usually complex sensor design.

PIR sensor – Motion detector sensor

Ultrasonic sensor

Light dependent resistor (LDR)

- Light dependent resistor (LDR) is used to detect change in light intensity or as a light sensor. LDR is basically a variable resistor.
- LDR resistance changes with the change in intensity of light. If intensity of light falling on LDR is high, LDR will have low resistance.
- When intensity of light decreases, LDR offer high resistance. Hence there is an inverse relationship between intensity of light and resistance of LDR.
- The LDR is a resistor, and its resistance varies according to the amount of light falling on its surface.

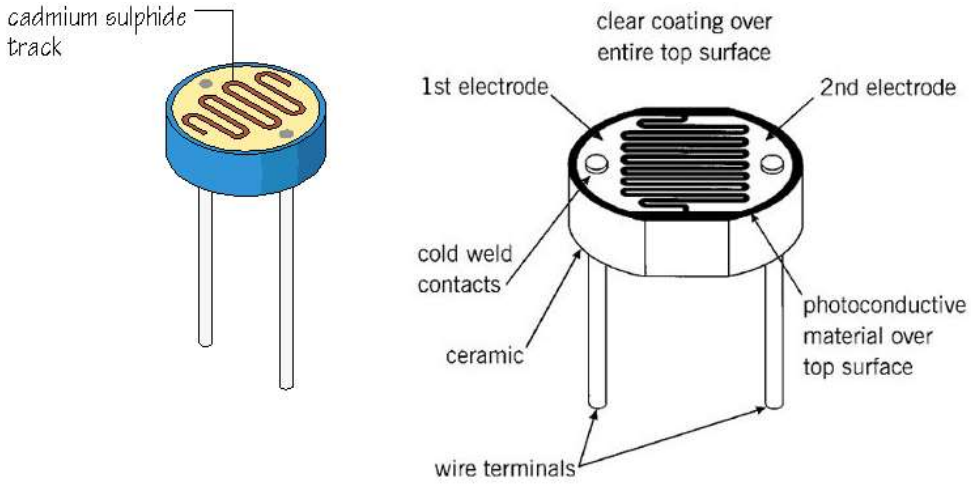
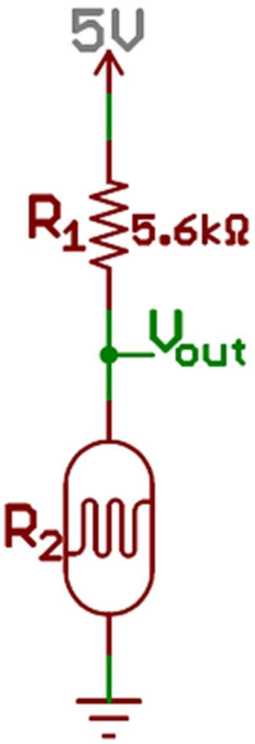
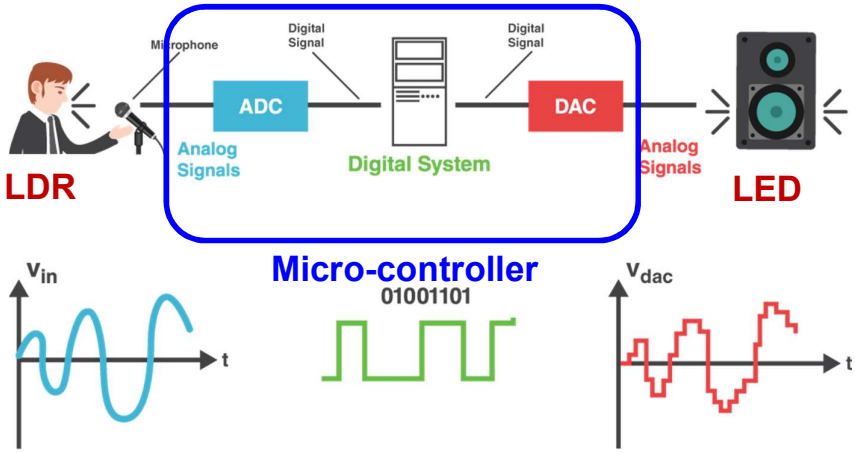


Figure: LDR



Light Level	R_2 (Sensor)	R_1 (Fixed)	V_{out}
Light	1kΩ	5.6kΩ	0.76 V
Dim	7kΩ	5.6kΩ	2.78 V
Dark	10kΩ	5.6kΩ	3.21 V

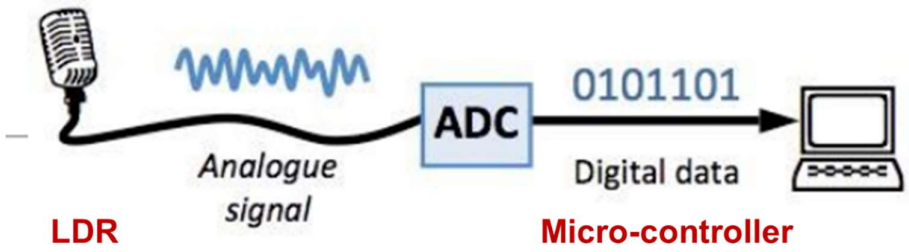
LDR makes up half of this voltage divider. The voltage is measured to find the resistance of the light sensor. As the light intensity increases, the resistance decrease.



Micro-controller has both ADC and DAC capability

Analog to digital conversion

The ADC translates the analog waves of your voice into digital data that the computer can understand. To do this, it samples, or digitizes, the sound by taking precise measurements of the wave at frequent intervals.



Microcontrollers are capable of detecting binary signals: is the button pressed or not? These are digital signals. When a microcontroller is powered from five volts, it understands zero volts (0V) as a binary 0 and a five volts (5V) as a binary 1. The world however is not so simple and likes to use shades of gray. What if the signal is 2.72V? Is that a zero or a one? We often need to measure signals that vary; these are called analog signals. A 5V analog sensor may output 0.01V or 4.99V or anything inbetween. Luckily, nearly all microcontrollers have a device built into them that allows us to convert these voltages into values that we can use in a program to make a decision.

An Analog to Digital Converter (ADC) is a very useful feature that converts an analog voltage on a pin to a digital number. By converting from the analog world to the digital world, we can begin to use electronics to interface to the analog world around us.

Not every pin on a microcontroller has the ability to do analog to digital conversions. On the Arduino board, these pins have an ‘A’ in front of their label (A0 through A5) to indicate these pins can read analog voltages.

Pulse-code modulation (PCM)

The most widely used technique for digitizing information signals for electronic data transmission is pulse-code modulation (PCM). PCM signals are serial digital data.

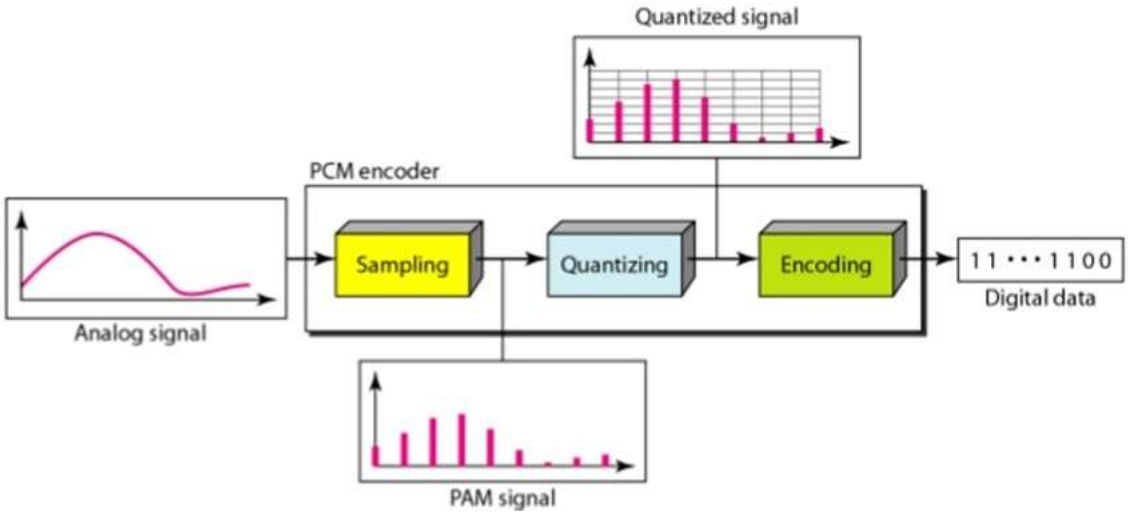


Figure: Block diagram showing Pulse code modulation process

A. Sampling

- Sample is a single measurement of amplitude.
- The larger sampling rate, the better accuracy of conversion.

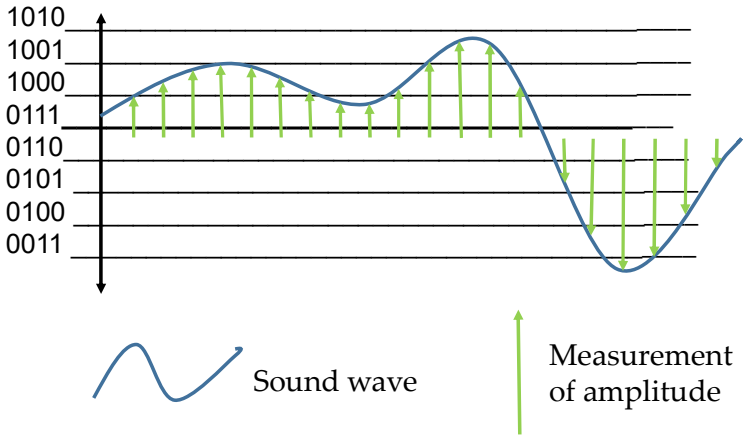
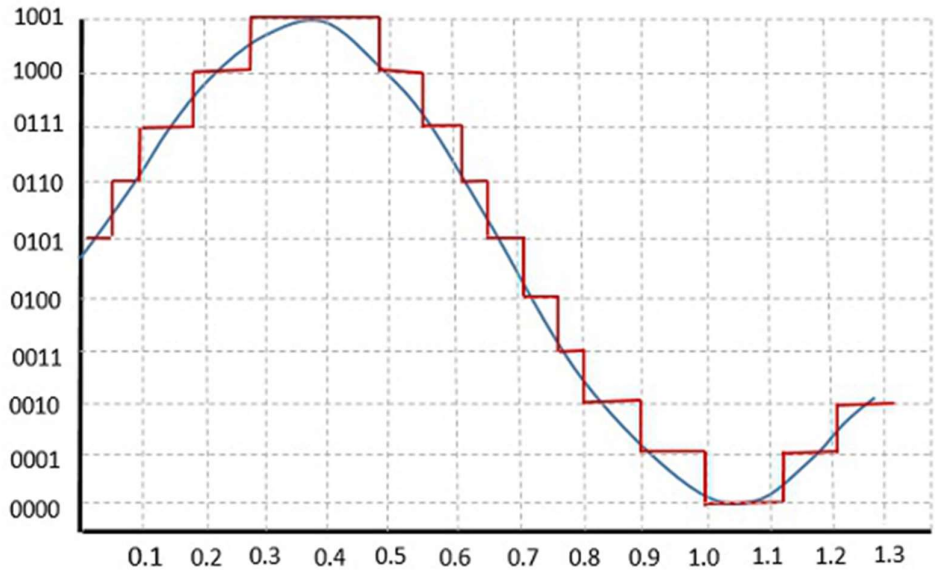


Figure: Sampling

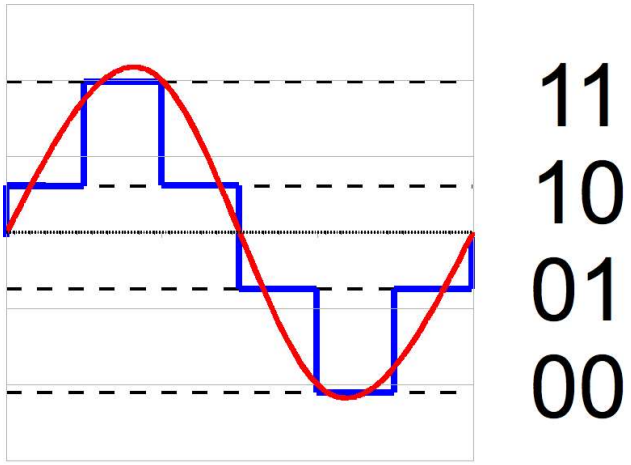
B. Quantization

The quantizing of an analog signal is done by discretizing the signal with a number of quantization levels. **Quantization** is representing the sampled values of the amplitude by a finite set of levels, which means converting a continuous-amplitude sample into a discrete-time signal.

- Both sampling and quantization result in the loss of information. The quality of a Quantized output depends upon the number of quantization levels used.
- The difference between an input value and its quantized value is called a **Quantization Error**.

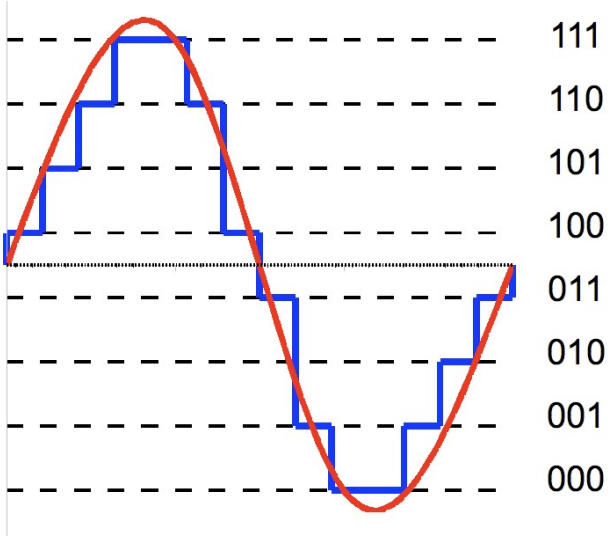


The figure shows how an analog signal gets quantized. The blue line represents analog signal while the brown one represents the quantized signal.



2-bit resolution with four levels of quantization

Red line: Actual signal
Blue line: Re-constructed signal



3-bit resolution with eight levels of quantization

Red line: Actual signal
Blue line: Re-constructed signal

C. Encoding
After the quantization process, the corresponding level is represented using binary numbers.
Encoded result: 10, 11, 11, 10, 01, 00, 00, 01

Arduino ADC
ADCs can vary greatly between microcontroller. The ADC on the Arduino is a 10-bit ADC meaning it has the ability to detect 1,024 (2^{10}) discrete analog levels. Some microcontrollers have 8-bit ADCs (2^8 = 256 discrete levels) and some have 16-bit ADCs (2^{16} = 65,535 discrete levels).

The way an ADC works is fairly complex. There are a few different ways to achieve this feat (see Wikipedia for a list), but one of the most common technique uses the analog voltage to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete.

Relating ADC Value to Voltage

The ADC reports a *ratiometric value*. This means that the ADC assumes 5V is 1023 and anything less than 5V will be a ratio between 5V and 1023.

$$\frac{Resolution\ of\ the\ ADC}{System\ Voltage} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

Analog to digital conversions are dependant on the system voltage. Because we predominantly use the 10-bit ADC of the Arduino on a 5V system, we can simplify this equation slightly:

$$\frac{1023}{5} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

If your system is 3.3V, you simply change 5V out with 3.3V in the equation. If your system is 3.3V and your ADC is reporting 512, what is the voltage measured? It is approximately 1.65V.

If the analog voltage is 2.12V what will the ADC report as a value?

$$\frac{1023}{5.00V} = \frac{x}{2.12V}$$

Rearrange things a bit and we get:

$$\frac{1023}{5.00V} * 2.12V = x$$

$$x = 434$$

The ADC should report 434. No fraction value is allowed. Take either floor or celling value.

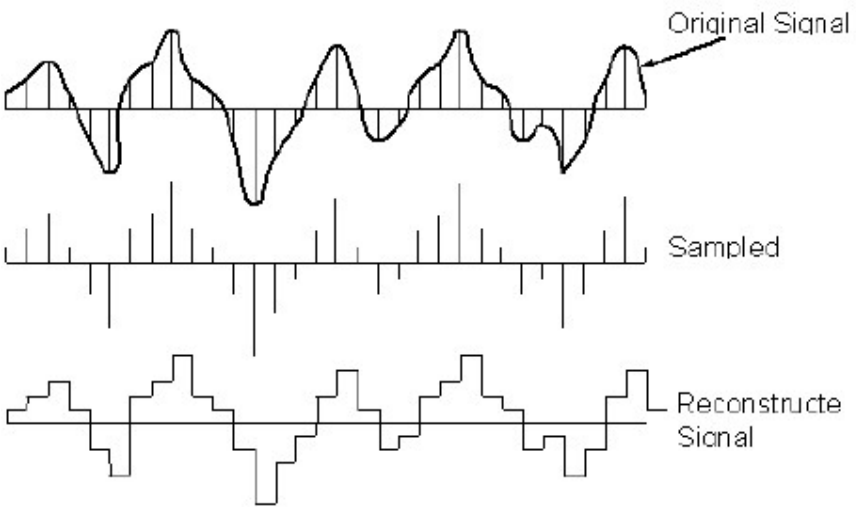


Figure : Analog to digital conversion

Arduino ADC Example

Figure: Analog pins of Arduino board

The analog to digital version by using the analogRead()command:

```
int x = analogRead(A3); //Reads the analog value on pin A3 into x
```

The value that is returned and stored in x will be a value from 0 to 1023. The Arduino has a 10-bit ADC ($2^{10} = 1024$). We store this value into an int because x is bigger (10 bits) than what a byte can hold (8 bits).

Let's print this value to watch it as it changes:

```
Serial.print("Analog value: ");
Serial.println(x);
```

As we change the analog value, x should also change. For example, if x is reported to be 334, and we're using the Arduino at 5V, what is the actual voltage? Pull out your digital multimeter and check the actual voltage. It should be approximately 1.63V.

Code

```
// the setup routine runs once when you press reset:
void setup()
{
      // initialize serial communication at 9600 bits per second:
      Serial.begin(9600);
}
// the loop routine runs over and over again forever:
void loop()
{
  // read the input on analog pin 0:
     int sensorValue = analogRead(A0);
  // print out the value you read:
    Serial.println(sensorValue); // LDR sensor value
```

```
    delay(50); //short delay for faster response to light.
}
```

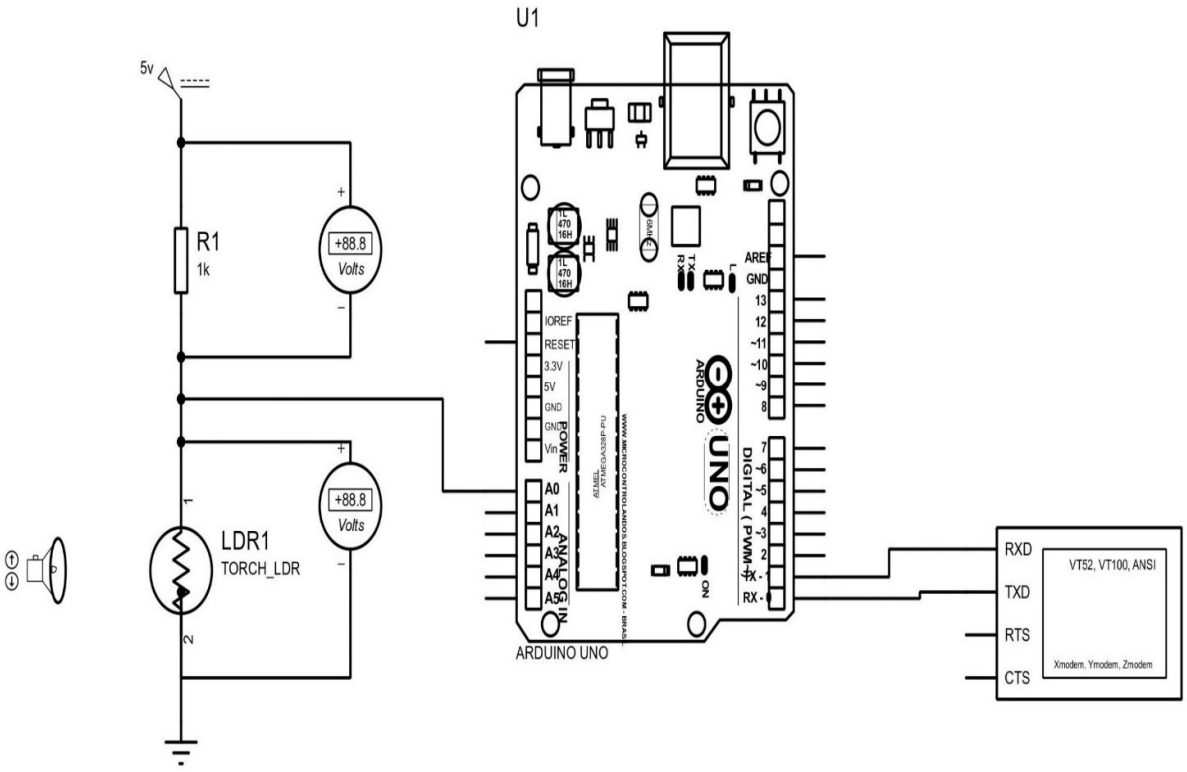
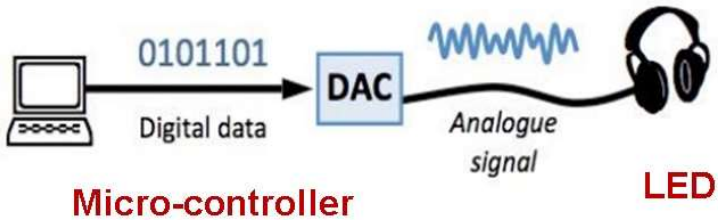


Figure: Arduino interfacing with LDR

Analog to Digital conversion

If you were to play your recording back through the speakers, the DAC would perform the same basic steps in reverse. With accurate measurements and a fast sampling rate, the restored analog signal can be nearly identical to the original sound wave.



- Pulse Width Modulation, or PWM, is a technique for getting digital to analog conversion.
- Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern (duty cycle) 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. The duration of "on time" is called the pulse width.

- To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.
- Arduino digital to analog resolution is 8 bit. (0-255)

Duty cycle

A duty cycle is the fraction of one period in which a signal or system is active. Duty cycle is commonly expressed as a percentage or a ratio. A period is the time it takes for a signal to complete an on-and-off cycle.

In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

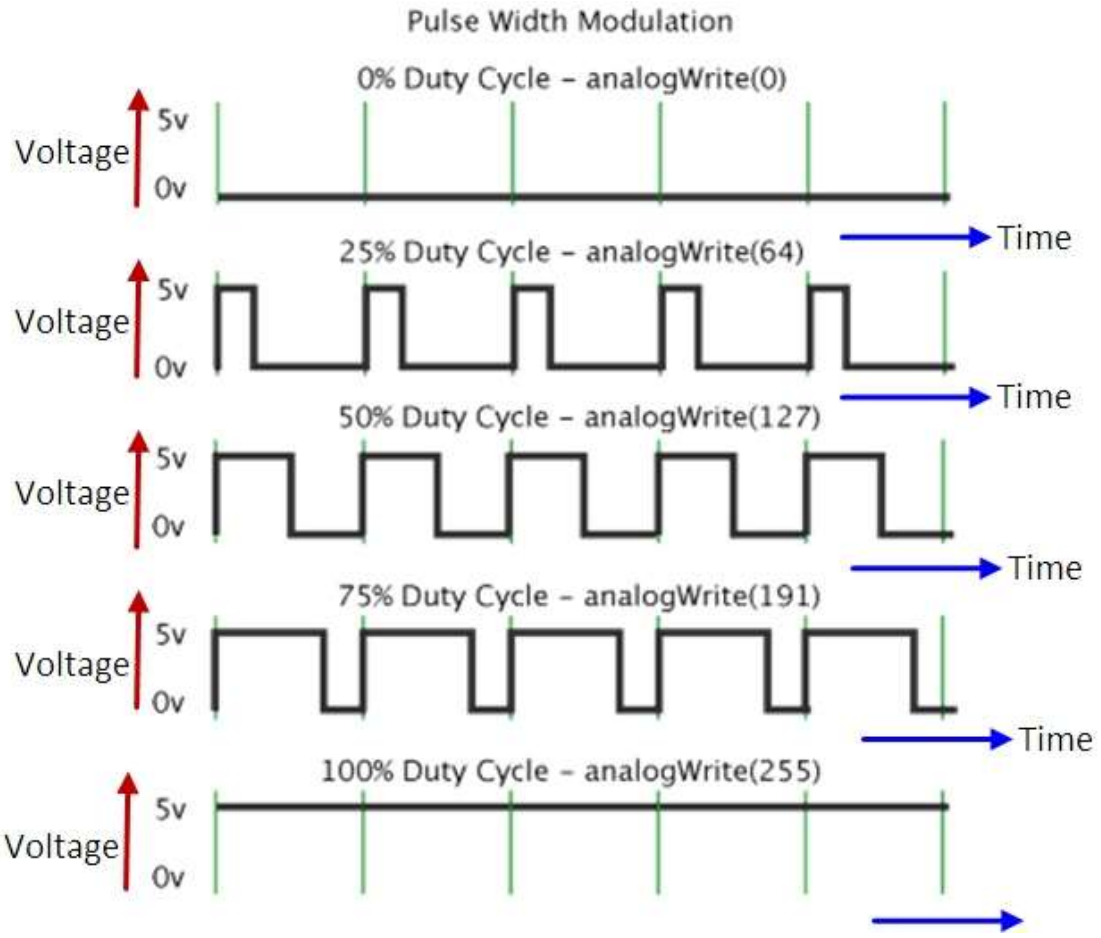


Figure: Digital to analog conversion

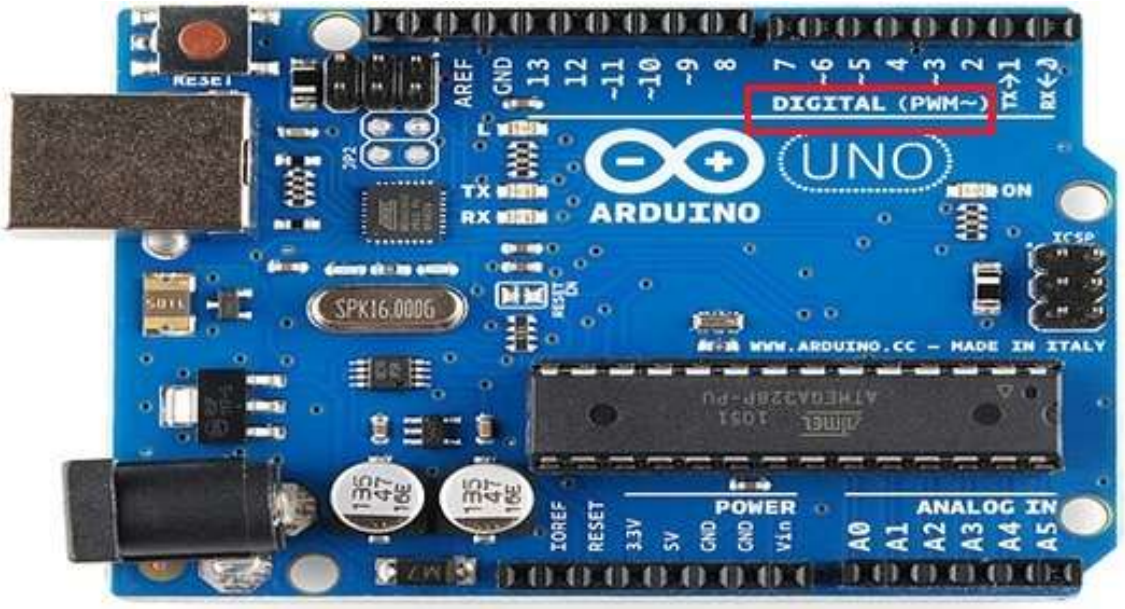
Code: This example shows how to fade an LED using the analogWrite() function.

```
int ledPin = 9;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue<= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue>= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```



PWM supporting digital pins are: 3, 5, 6, 9, 10, 11

Exercise

1. Explain the steps, how Analog to Digital converter in Arduino work? Draw the block diagram.
2. What is Duty cycle? Explain with a diagram how duty cycle will be used to control the brightness of LED.