ARDUINO

LEVEL I

SAMPLE

EDUCATION

تُطلق IP Education المُختصة بالخدمات التعليمية النسخة الاولى من كتاب (Arduino) وذلك كمستوى اول ضمن السلسلة المتكاملة (المهندس في الاردوينو). يتم تأسيس المُتعلم من خلاله بشكل راسخ وبإطار هندسي في المنصة الأكثر شهرة في عالم الالكترونيات التفاعلية (الاردوينو), بحيث يكون قادراً على تصميم وبرمجة دارات وأنظمة إلكترونية بسيطة ومتوسطة التعقيد مبنية على الاردوينو , وذلك من خلال تعّلم كيفية رسم وقراءة المخططات (Schematic diagrams) للدارات قبل تصميمها, القيام بالحسابات الهندسية (البسيطة) المهمة والضرورية لاختيار القطع والحساسات المناسبة, معرفة المواصفات والميزات التقنية لكل قطعة او حساس مُستخدم, بالإضافة الى تعّلم اساسيات البرمجة (Text Coding) ومن ثم استخدام برمجية (Arduino IDE) لاغراض البرمجة, كل ما سبق سَيجدُه ويَتعَّلمه المُتعلم بأسلوب مُبسط عملي قائم على تقنية (التعلم من خلال المشروع) ومرتبط بتطبيقات الحياة اليومية وضمن استراتيجية تعمل على ربط المعرفة النظرية بالتطبيق العملي الواقعي. يشتمل الكتاب على المحاور الرئيسية التالية:

- مقدمة حول المعالجات والمتحكمات الدقيقة والالواح الالكترونية
- مقدمة للاساسيات البرمجة (Text Coding)
- مدخل الى منصة الالكترونيات التفاعلية (Arduino)
- اللوحة الالكترونية Arduino UNO
- أنواع الإشارات الكهربائية (Analog and Digital)
- تقنية (Pulse Width Modulation)
- مشاريع وتجارب تطبيقية اساسية
- مشاريع وتجارب تطبيقية متوسطة التعقيد

هذا الكتاب جزء من سلسلة منهجية مُنظمة تهدف الى الاخذ بيد المُتعلم كمبتدىء في عالم التصميم الالكتروني باستخدام منصة الاردوينو وحتى وصوله الى الاحتراف والقدرة على تصميم مشاريع وأنظمة مُعقدة. يُرجى العلم بأن هذا الكتاب مُوجه لطلاب المدارس والجامعات والمهتمين من ذوي الاعمار 10 فما فوق وقد تم اعداده فنيا ليكون شاملاً ومُبسطاً في آن معاً ليُناسب بشكل اساسي غير المتخصصين وليكون رافداً مفيداً ومرجعاً ذو قيمة للمختصين المهتمين بعالم الاردوينو. بالاضافة الى ذلك يمكن استخدامه من قبل المعلمين والمدربين كمادة تدريبية في حصصهم او دوراتهم .

Preface

IP Education launches the first version of (Arduino 101) book as the beginning level of the integrated series (Arduino Engineer). Students are going to be firmly and deeply equipped with an extensive batch of the basic knowledge of the (Arduino) platform within an engineering framework, they are ensured to be able to design and program simple and average-complexity Arduino-based circuits and systems through learning how to draw and read schematic diagrams of circuits before being designed, doing (simple) math which is required to determine and choose the proper components and sensors and realizing the key specifications of them, and learning the basics of text Coding to use the (Arduino IDE) and for programming issues. All mentioned, are being assimilated and recognized by students within a simplified hands-on, project-based, real-life implementations methodology and in the context of a strategy that narrows the gap between the theoretical knowledge and the practical application. This book mainly contains the below themes (not limited to):

- Introduction to microprocessors, microcontrollers, and electronic boards
- Introduction to the basics of text coding
- A general picture of the Arduino platform
- Arduino UNO as an electronic board
- Electrical signals (Digital and Analog)
- The Pulse Width Modulation technique
- Simple essential projects and experiments
- Stepped up average-complexity projects and experiments

This book is a part of a methodical and systematic series that aims to elevate and level up students from being beginners to be professionals when it comes to Arduino-based electronic design. Be acquainted that this book targets school students, university students, interested people who are 15 years old and above, it's technically designed to be comprehensive and facilitated at the same time, on which this book best fits non-specialists and it also so useful and has a lot of benefits for specialists who are interested in Arduino.

Table of content

Worthy introduction of being read: in this section, you are going to be introduced to basics of Arduino-based programming and circuitry, by learning how to:

1. Generate Digital/Analog output-signals to power on/off loads that may be connected to an Arduino UNO board.

2. Receive Digital / Analog input-signals from an external source (a circuit or a component) that may be connected to an Arduino UNO board. And how to process the received data to take action.

3. Carry out a serial communication between a computer and an Arduino board to exchange data.

That's will be done by designing and programing real circuits that are based on employing the foregoing concepts.

Experiment 1: Digital output-signals

Description:
In this workshop, you will design simple **LED-circuits** and program them so that, a LED/LEDs will be powered on/off depending on a digital output signal/signals that comes from the Arduino UNO board. This experiment is broken into three parts:

- In the first part, a LED should be powered-on for 2 seconds, then powered-off for another 2 seconds.

- In the second part, two LEDs should be powered-on for 2 seconds, then powered-off for another 2 seconds, one by one.

- In the third part, a one-digit common-anode 7-SEG will be used to design a digital seconds-counter that counts from 1 to 3 repeatedly.

Light Emitting Diode (LED):

This semiconductor is a light source, which emits lights when activated, that happens when appropriate voltage (known as forward voltage) is applied to the LED terminals.

When voltage is applied forwardly, that means the **positive terminal** of a source is connected to the anode leg of a diode and the negative terminal to the cathode leg, when that, the LED will bright. If this connection is inverted, a reverse voltage is applied, and the LED will be off. The Forward voltage as well the forward Current of a LED varies according to the color of a LED (a 633nm Red LED will be used in this experiment).

Technical details & Specifications of (a 633nm Red LED):

- Superior weather resistance.
- Forward Voltage (VF): 1.7V (Max. 2V).
- Forward Current (IF): 5mA (Max. 20mA).
- Maximum Reverse Voltage: 5V.
- Luminous Intensity: 20mcd.

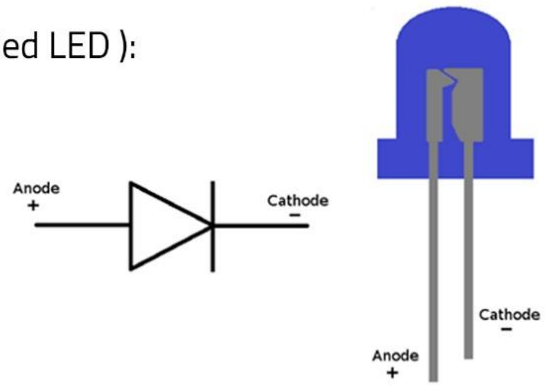


Figure 3.1.1 :LED - schematic

7-Segments (7-SEG):

it's nothing but a group of LED-segments that connected parallelly, by turning some on and others off, you can display a certain number from (1 to 9).

a (One-digit 7-SEG) is used to show one-digit number, a (Two-digits 7-SEG) is used to show two-digits number and so on. **7-Segments** are classified according to the **internal-connection** to:

1. Common-cathode 7-SEGs
2. Common-anode 7-SEGs

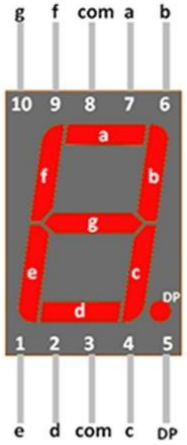


Figure 3.1.2
One-digit 7-SEG display

1. Common-cathode 7-SEG, it's characterized with:

- All **the cathodes-terminals** of the LEDs are connected with each other and treated as **one terminal** that has two ports (COM1 and COM2).

- Each **LED's anode-terminal** is treated as a standalone terminal.

- To show a number, connect a **COM terminal** to the **GND pin**, then any LED you want to light, all you have to do is connecting that **LED's anode-terminal** to an **I/O pin** and give it a **HIGH** signal.

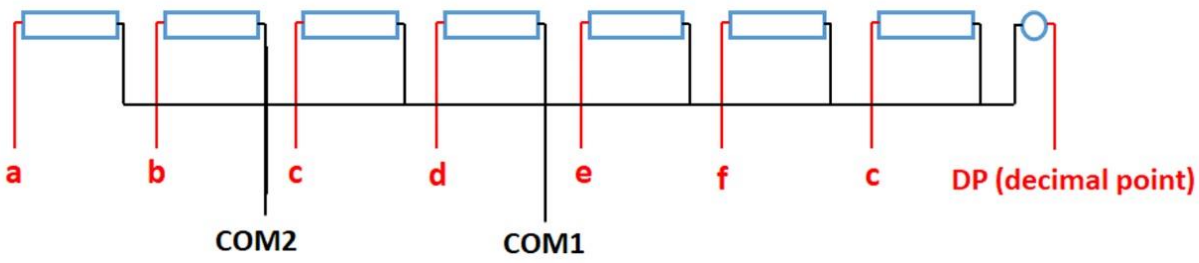


Figure 3.1.3 :common-cathode 7-SEG - schematic

2. Common-anode 7-SEG, it's characterized with:

- All **the anodes-terminals** of the LEDs are connected with each other and treated as **one terminal** that has two ports **(COM1 and COM2).**

- Each LED's cathode-terminal is treated as a standalone terminal.

- To show a number, connect a **COM terminal** to the **5V pin,** then any LED you want to light, all you have to do is connecting that **LED's anode-terminal** to an **I/O pin** and give it a **LOW** signal.

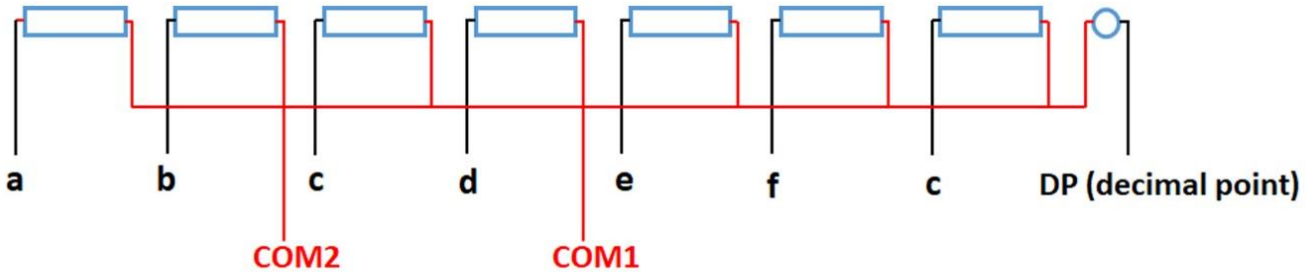
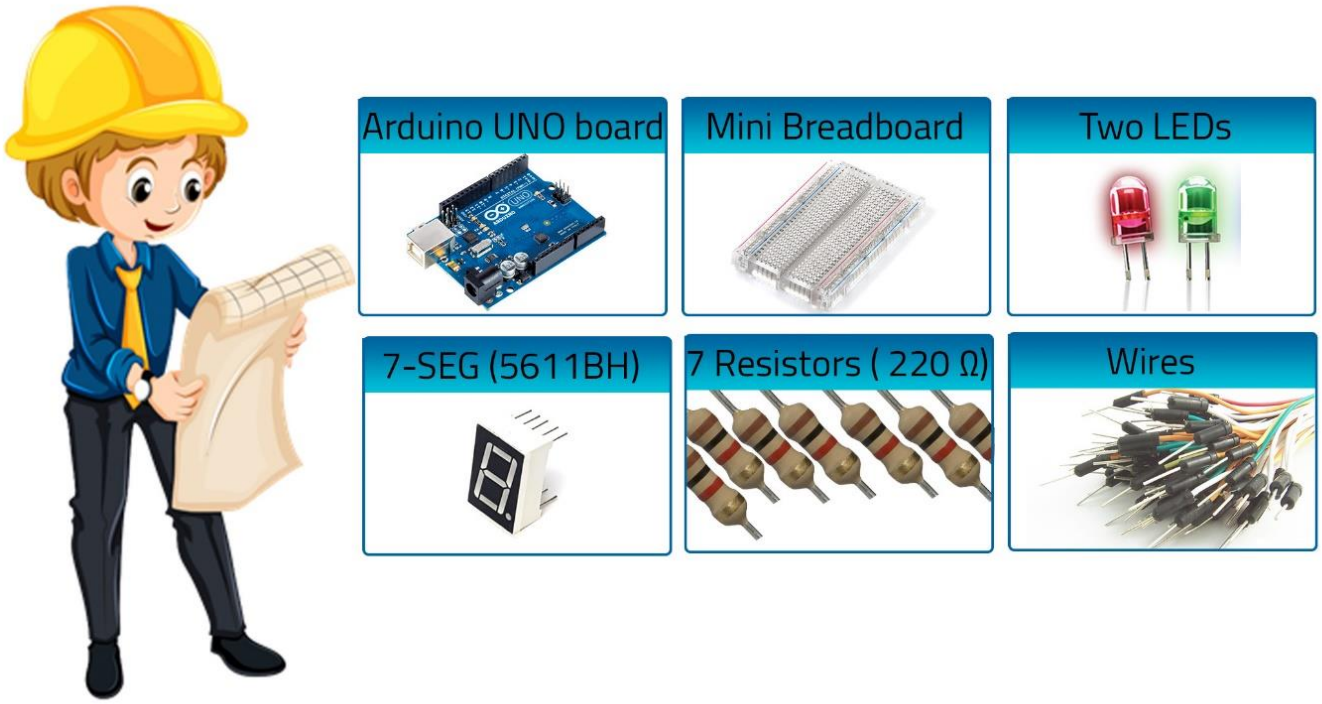


Figure 3.1.4: common-anode 7-SEG - schematic

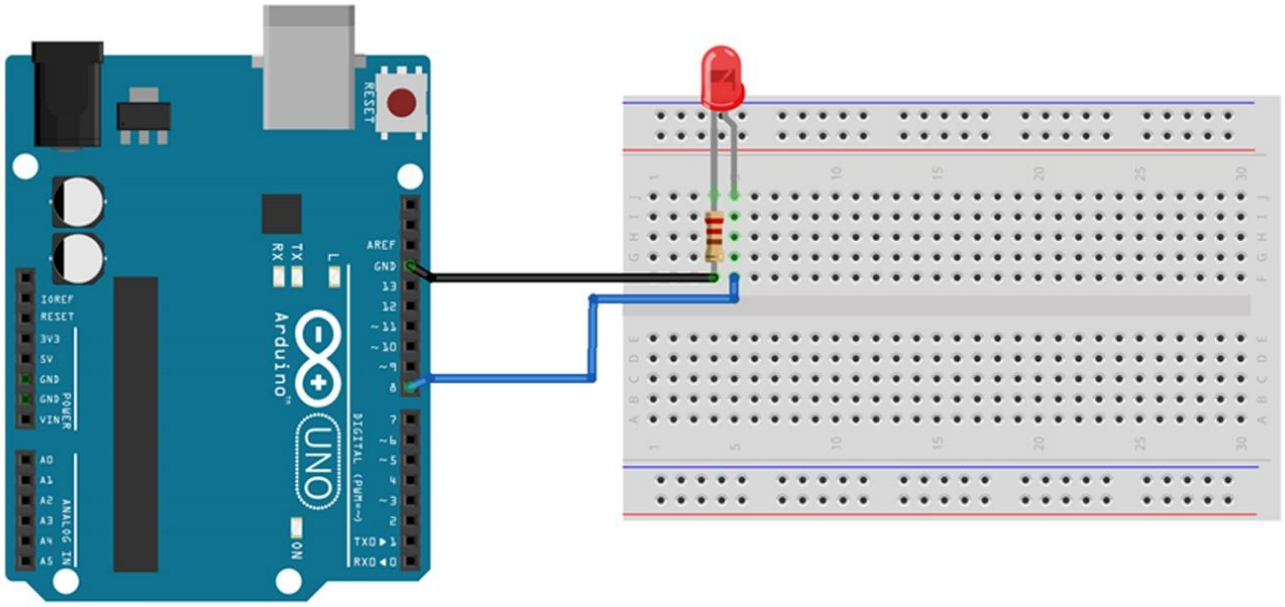
Note:
In this experiment, you are going to use **5611BH 7-SEG** which is one-digit common-anode.

The Required Components:

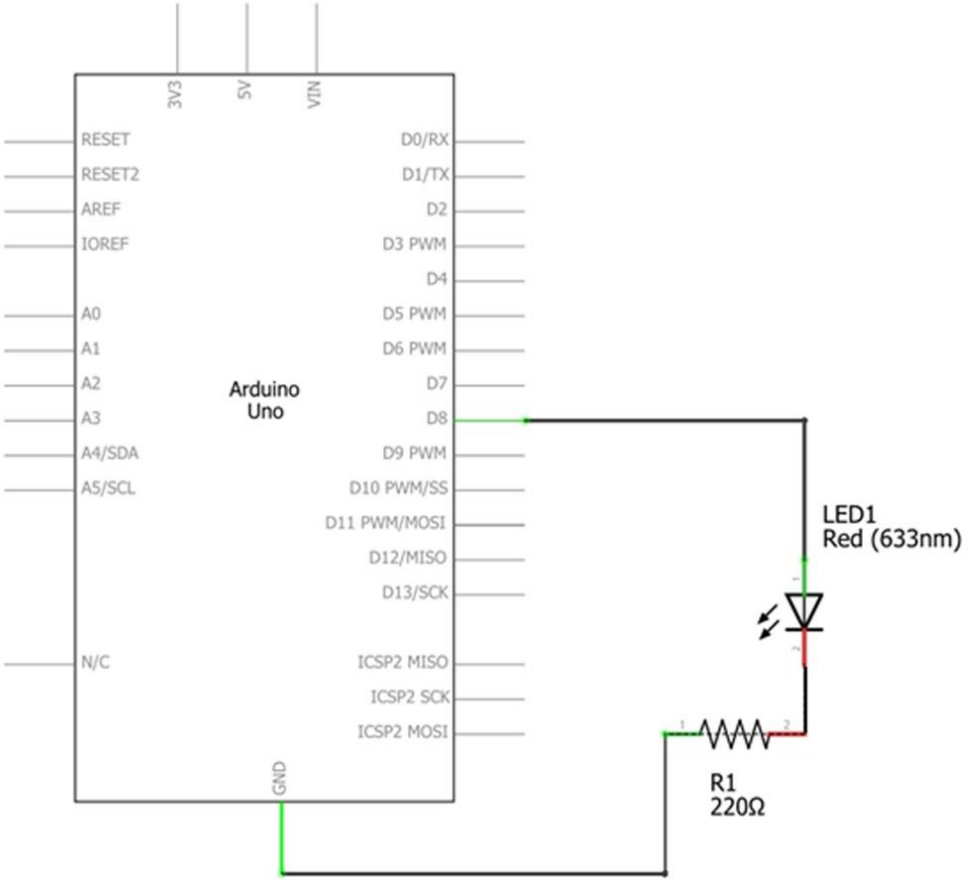


The First Part

The Breadboard Diagram



The Schematic Diagram



The Code

Digital output-signals

```
int LED = 8;

void setup()
{
pinMode(LED,OUTPUT);
}


void loop()
{
digitalWrite(LED,HIGH);




delay(2000);



digitalWrite(LED,LOW);
delay(2000);
}
```

By writing this line, there will be a global int variable with the value of 8, and pin #8 - on the Arduino board- is marked and labeled with the name (LED).

By calling this function, you tell the Arduino board that a certain pin will be used as an input or an output pin. In this code, pin #8 which is labeled as (LED) will be used as an OUTPUT pin.

By calling this function, you order your Arduino board to generate a HIGH or a LOW digital signal and send it through a certain output pin. In this code, a HIGH digital signal will be sent through pin #8 that is labeled as (LED).

By calling this function, you order your Arduino board to delay the executing of any line-of code after, as per how many milliseconds are written between the parentheses.

By calling this function, you order your Arduino board to generate a LOW digital signal and send it through pin #8.

An Engineer's Calculations

In general, the LEDs we use, are mainly made of diodes, those LEDs have various colors, each color has its own wavelength which determines its specifications such as (forward current, forward voltage, reverse voltage, ...etc.). Let's take a Red LED which is 633nm wavelength into account, the LED has:

- VF= 1.7V (Max-VF= 2V),
- IF= 5mA (Max-IF= 20mA)
- Max-VR= 5V