<u>INDEX</u>

i	Introduction	2
ii	Kit Content	3
iii	Component Explanation/Pinouts	6
iv	Indentifying Resistor Colour Codes	19
v	Identifying Ceramic Capacitor Code	20
1	Getting Started with Arduino	21
2	Arduino Motion Detector	23
3	3-Way Traffic Light	25
4	Interfacing LCD with Arduino	27
5	Interfacing 7-Segment Display with Arduino	29
6	Controlling RGB LED with Arduino	31
7	Arduino Clap Circuit	33
8	Shift Register with Arduino	35
9	Distance measurement using US sensor	37
10	Arduino Digital Voltmeter	39
11	Light Detector using Arduino	41
12	IR Controlled DC Motor	43
13	DC Motor Speed controller	45
14	Digital Thermometer	47
15	Arduino Calculator using Keypad	49
16	Servo Motor Control	51
iv	Bonus Projects	53
V	Credits	54

INTRODUCTION

Thank you for Purchasing the "Professional Arduino Starter Kit" by Circuitloop. This kit aims to get you started with electronics and enjoy the world programming through Arduino. The contents of the kit are well planned to cover maximum number of projects with minimum number of components in hand. It also avoids the hassle of purchasing individual components for your project and each component is tested by our staff to ensure their working so you can stay fool proof.

How to use this Book?

This booklet will be helpful for beginners to get started with the kit in a sequential manner. The book covers about 16 Projects each of varying difficulty level. As a token of gratitude for purchasing our kit we have also linked 5 Bonus Projects which you can check of after finishing the basic projects.

Each project in this booklet is covered with a colourful easy to understand circuit diagram, a picture of the hardware and the code of the Project. Also along with the book you will receive a CD which will contain the entire source codes of all the projects. Every Project mentioned here is also tested by our staff for flaw in circuit diagram or code.

What Next?

Once you have received your kit, move to the "Kit Content" section on this book where you will find pictures of all the components include in the kit. So make sure you have received all the components correctly.

Next read through the component description of every component to get used to the components in the kit and also to indentify the pinouts and use of every component in the kit.

For beginners it is recommended to start with project one and proceed logically upto project 16. Use the circuit diagram in the book to make your connection you can refer the hardware picture if you have any doubt on how to connect it. Always scan the QR code to get into website where you will find the complete information of the project with much detailed explanation.

Kit Contents			
Arduino Uno	Programming Cable	Bread Board	16×2 LCD display
Servo (MG90)	Relay Module	IR Module	PIR Sensor
		C B B B B B B B B B B B B B B B B B B B	
Ultrasonic sensor	7-Segment Display	Keypad	LM35
Otto			
LDR	Shift Register	Thermistor	Trimmer Pot
		/	
Buzzer	Microphone	DC Motor	Push Button
		2.	

RGB Led	Red LED	Green LED	Blue LED
A CONTRACT OF A			
Potentiometer (100k)	BC547 Transistor	BC557 Transistor	1N4007 Diode
			-
1k Resistor	10k Resistor	100k Resistor	220R Resistor
ALL S	AL S	JA	AND
0.1uF	10uF	100uF	22pF
40 Pin Berg stick	Hook-Up wire	Male to Male wires	Female to Male wires
9V Battery	Motor Fan	Program File CD	Printed Guide

Arduino Uno. Arduino Uno, as you might know is the Microcontroller Development module. It will be the brain of the projects that you develop using this kit. It is a compact computer which can take inputs, perform calculations and give outputs. We will be learning more about this module as we go through this module.



Arduino Uno Rev 3

Programming Cable – The Program for the Arduino Module will be written using the Arduino Module on the Computer/Laptop. This Program will then be uploaded to the Arduino Module, thorough this Programming Cable. It has USB type B on one side which connects to the Arduino Module and Type A on other side which connects to the Computer. It also acts as a source of Power.



Bread Board – This is the board on which you can build projects. It has holes on them which are connected internally in a particular pattern as shown in the below picture. The holes which are connected with a line represents they are connected internally. The Red line indicates Power, which is normally connected to the power rail. The Blue line indicates Ground, which is normally connected to the ground of the circuit. IC's like shift registers can be placed in the middle breadboard to share the 1^{st} eight pins to the yellow line and the 2^{nd} eight pins to the green lines.

The Male to Male and Female to Male connectors provided along with this kit can be used to connect the module and Arduino Board to other components using the Bread board.



 16×2 LCD Display – The 16×2 Alphanumeric Display is the common display module that comes handy with microcontroller projects. It has 16 Rows and 2 columns, which gives together of 32 boxes. We can display one character at each box. These characters can be alphabets, numbers or even some custom symbols.

The module has a total of 16 pins. You will learn how to use this module with Arduino. These modules are also available in many size and colours. The one provided with this kit is green colour.



Servo Motor (SG90) – Servo Motors are really cool since they provide controlled rotation. They are commonly used in robotics projects like Robotic arm, hexapods etc. These motors have three pins out of which one is the data pin. The data pin is provided with a PWM signal, and based on the duty cycle of the PWM the rotation of the servo is decided. We will learn more about this through our projects

The one provided with this kit is the popular SG90 Servo motor with a torque of 2.5kg/cm. This motor can rotate only from 0° to 180°. There are also other motors available in the market with better torque.



IR Module – The IR modules are used to detect the presence of an object at close range. It has an IR LED and an IR receiver, the IR Led emits IR rays and the receiver checks if the rays are returned back. These rays will return back only if it hit an object before it, hence we can determine if there is an object in front of the IR module. This module can also be used to differentiate between white and black colour, since black completely observes IR light and white completely reflects it. So this module can be used to build Line follower Robots, Obstacle avoider robots and much more



PIR Sensor – The PIR stands for Passive Infrared Sensor. This sensor is used for motion detection, as it can differentiate between living and non-living things. All living beings emit a small amount IR rays from their body, this sensor reads this IR rays and gets triggered.

The module provided with it is the popular HC-SR501 PIR sensor which can operate in two modes, H mode and I mode, but commonly is used in H mode. Also it has two potentiometer (orange coloured) using which we can control the sensitivity and off time of the sensor as marked below. We will learn more about this sensor though our projects



Ultrasonic Sensor –The Ultrasonic sensor is another cool sensor which can be used to measure distances accurately upto 80cm (practically). As the name suggests the sensor uses Ultrasonic wave to detect the presence of objects ahead of it just like a SONAR system. This sensor is commonly used with Arduino to build robots, so that the bot know what is present in front it and avoid hitting on them. We will learn more about this sensor through our projects.

The sensor provided with this kit is the commonly used and popular HC-SR04. It has four pins of which two is for power and two is used for data. The pin out of the sensor is given below



7-Segment Display – The 7-Segement displays are also a very popular and commonly used in embedded projects. This displays work great in situation where the values displayed have to be viewed from a long range. As the name suggests a single seven segment display has 7 leds (minimum) each forming a single segment. These segmented Leds can be made to glow in a particular pattern to form a single character.

7-segment Displays are available in many colour and sizes. It also has a common anode and common cathode type. The one provided with the kit is a common cathode red colour seven segment display



Thermistor – A Thermistor is a sensor which is used to detect change in temperature. Technically it a resistor, but the value of its resistance changes based on the temperature. So by determining this change in resistance we can estimate the change in temperature. This sensor is commonly used for fire detection and other such related projects.

Since it is just like a resistor it does not have any polarity, it has two leads which can be used in any direction. The Thermistor is also rated in terms of ohms; the one provided with this kit is 10k ohms



LM35 – The LM35 is another type of temperature sensor which can be used to measure temperature from -55°C to 150°C with an accuracy of 1°C. It can be used to measure room temperature or enclosure temperature etc.. The sensor senses the temperature and outputs the result as a variable voltage. This variable voltage can then be measured using any microcontroller with ADC feature like our Arduino. Then we have a small formula which can be used to convert this measured voltage into a meaningful temperature value.

The LM35 is a 3-pin device which looks similar to a transistor, the pin out of the sensor is shown below. We will learn more about this through our projects



LDR – The term LDR stands for Light Dependent Resistor, meaning the resistance value depends on the light around it. This sensor is used to detect light as you might have already judged; it is commonly used to distinguish between day and night or to find from which direction the light is falling so that it can be used in projects like automatic street light, solar panel tracking etc..



Shift Register - The shift register 74HC595 is an integrated IC with 16 pins. It is an 8-bit Serial In Parallel out Shift register. Meaning it can take input serially and control 8 output pins based on the serial data received. It can be used to expand the GPIO functionality of the Arduino board. It is normally used to control LEDs so that it can be used in projects like LED cube, matrix LED etc..



DC Motor – Many of us could have come across this Hobby DC motors through kids toys such as remote controlled cars, trains etc, hence this motor is also called as **Toy motor**. Nevertheless this motor can also be used for many other common purposes and is mostly used by beginner level electronics hobbyist. They are very rugged and easy to use and control with commonly available and wide range of voltage levels. These motors can rotate in either direction and speed control is also possible, however do not expect a very fast speed and high torque from these small guys.



Buzzer – A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on breadboard and is a widely used component in most electronic applications.

The one given in this kit is a simple buzzer which when powered will make a Continuous Beeeeeeppp.... sound, but when it is connected to Arduino we can produce different sounds by controlling voltage supplied to it. The Buzzer is polarity sensitive so make sure you can figure out the positive and negative pin from the image below.



Microphone – An Electret Condenser microphone (a.k.a Condenser Microphone) is the most common type of transducer that is used to detect or measure sound signals. It functions exactly as opposite of a normal speaker, that is when a sound signal is detected it produces electrical signals.

By using a simple circuit we can easy measure the amount sound produced using an Arduino. This can b used to build projects like clap switch, noise detector etc. We will learn more about using this microphone through our tutorials. The Microphone also have a positive and ground terminal which can be identified using the below picture.



9V Battery – How could one ignore the famous 9V battery, when building electronic projects! These are commonly used batteries to power all your circuits. It can also be used the power the Arduino itself through the Vin pin. Do not expect it to run heavy loads from this small guy, but for the projects involved in this booklet, this battery will work just fine.

In this kit the battery is also provided with a battery clip, which can be connected to the battery directly, then the red wire will be positive and black wire will be the ground wire.



Push buttons - Push-Buttons a.k.a tactile switches are used to close or open a circuit. These buttons can be used as inputs for the Arduino, using which we can trigger an action through the program.

The buttons has 4-pins, out of which the horizontal pins are connected internally. So we are left with two pair of pins. These pair of pins will be connected to gather when the switch is pressed and will be disconnected when the button is released. The below image will help you to understand better



RGB Led - A RGB LED stands for Red, Green and Blue Led. It is commonly used component in electronics, generally, as it is used for indication purpose. You can use RGB LED in various projects like portable flashlight, LED indicator etc. An RGB LED can also be used for work according to condition like for condition 1st Red will glow, for condition 2nd green will glow and for condition 3rd blue will glow. Basically this single LED can be made to glow in three different colours based on the way we use it.

It has four pins out of which only two pins will be used at a time. There are two types of RGB Led then one given with this kit is the Common anode RGB LED. We will learn more about this Led when we interface this with our Arduino in our projects.



LED – Along with a RGB Led, the Kit also provides Individual Red, Green and Blue Leds. These Leds can be used as outputs with the Arduino and can be programmed as required. They are used to build simple projects like traffic light controller etc..

An LED has polarity and should always be used in combination with a resistor for Arduino. This resistor is called a current limiting resistor. The pin our of a LED is shown below, it applies the same for all coloured LEDs.



Potentiometer – Potentiometers also known as POT are nothing but variable resistors. They can provide a variable resistance by simply varying the knob on top of its head. The value or resistance decides how much opposition it provides to the flow of current. The greater the resistor value the smaller the current will flow. The potentiometer high with this kit is 100K 0.3W potentiometer.

These can be used to provide variable voltage to a microcontroller, by rotating the knob. It is commonly used in projects where the user has to vary a parameter manually.



BC547 Transistor – The BC547 is a NPN transistor. To put it simple, it is an electronic controlled switched that is, it can be closed or opened by using the GPIO pin on Arduino. For a NPN transistor the collector and emitter will be left open (Reverse biased) when the base pin (middle pin) is held at ground and will be closed (Forward biased) when a signal is provided to base pin.

A transistor is used with Arduino to control loads which cannot be controlled directly through the GPIO pins. The DC motor for example draws more current and hence it is mandatory to use a transistor to drive it using Arduino.



BC557 Transistor – The BC557 is a PNP transistor. This is also an electronic controlled through GPIO pin on Arduino like BC547 but it is of PNP type. A PNP transistor is just the opposite of NPN transistor in functionality, that is in this transistor the collector and emitter will be left closed (Forward biased) when the base pin (middle pin) is held at ground and will be open (Reverse biased) when a signal is provided to base pin.

Both the BC557 and BC547 have its own applications, but the BC547 is the most commonly used transistor since NPN transistors are preferred to avoid complexity. The pin out of a BC557 transistor is shown below.



1N4007 Diode - A diode is a device which allows current flow through only one direction. That is the current should always flow from the Anode to cathode. The cathode terminal can be identified by using a grey bar as shown in the picture above.

The 1N4007 Diode is the most commonly used diode, it is normally used a protection device to avoid reverse polarity or used as a free-wheeling diode to direct excess current to the ground.



Resistors – Resistors are commonly used passive devices on every electronic circuit. The function of a resistor is to limit the current or make a voltage drop across it. They are commonly with LED lights to limit the current flowing through it.

Along with this kit you are provided with four different types of resistors, which are of values 1K, 10K, 100K and 220R. The value of resistor can be identified using the colour code on top it. How to read this colour code will be explained later in this booklet. A resistor does not have any polarity so it does no matter in which direction we connect them



Ceramic Capacitor – Along with this kit you are provided with two types of capacitor. The orange coloured once are called ceramic capacitors. You will be provided with two values of ceramic capacitors, which is 0.1uF and 22pF. The value of a ceramic capacitor can be identified with the code printed on top it. How to read this code will be discussed further down in this booklet.

The Ceramic Capacitors has no polarity. Meaning they can be connected in any direction. They are breadboard friendly and can be easily used in the breadboard given with this kit.



Electrolytic Capacitor - The Electrolytic Capacitors have polarity. Meaning they have a positive and negative pin. The pin which is long is the positive pin and the pin which is short is the negative pin. You can also identify the polarity using the negative strip on the capacitor label. As shown in the picture above the negative pin will be directly under the negative symbol.

Along with this kit you will be provided with two value of electrolytic capacitor which is 10uF and 100uF. The value of this capacitor will be printed on the capacitor itself.



Indentifying Resistor Colour Codes

In order to identify the resistance value of a Resistor, we have to look at its colour code. Yeah! It would have been easy if the value was directly written, but still with little practise from below we can start reading the resistor values.



Now, with the help of this chart let's try to calculate the value of a resistor that we have in hand. The resistor I have chosen is shown below



So, the first digit is RED, the second digit is also RED, the multiplier is BROWN, the tolerance is GOLD.

This compared with the chart above give us the following numbers

So the value of the resistor is $22 \times 10\Omega$ which is 220Ω with a tolerance of ±5%. Similarly you can calculate the value of any resistor.

Indentifying Ceramic Capacitor Codes

The value of a ceramic capacitance will not be directly mentioned on the capacitor. There will always be a three digit number followed by a variable; let's learn how to identify the value using these numbers. Consider the following capacitor.



As you can notice, these three digits are split into two digits and the third one is the multiplier. In this case 68 is the digit and 3 is the multiplier. So 68 should be multiplied with 10^3. Simple put it is 68 followed by 3 zeros. Hence the value of this capacitor will be 68000 pF. Notice the unit should always be pF. Similarly a capacitor with 220 code means it is 22 Pico farad, since 10^0 is 0.

The voltage rating of the capacitor can be found by using the line under this code. If there is a line then the voltage value is 50/100V if there is no line then it is 500V.

Getting Started with Arduino



Scan me for Code, Circuit Diagram and more details on this project

Discover: Arduino IDE, Uploading pgms Difficulty: 0/5

Circuit Diagram





The name Arduino is typically buzz word among most of the electronics students or hobbyists alike. Its ability to build things faster and cheaper with a massive online community support has made it an ideal choice for many people who are just getting started with electronics or programming. Because of its simplified hardware design and programming environment, it is possible even for people without any electronics or computer science background to learn it with ease. So what actually is this Arduino? How can you get started with it? What can you make with it to improve your lifestyle?

All these questions will be tried to be answered in this tutorial, as we step through. We will set up the Arduino IDE on your PC/Laptop and upload a sample blink program to the Arduino. Then we will also build a small hardware using a bread board with a simple circuit consisting of a switch and LED and program our Arduino according to it. Sounds interesting enough?!!

In our hardware the push button is input which is connected to the second pin of the Arduino. This means that whenever we press the button the pin number 2 will be connected with ground. Next the LED is connected to pin number 3 through a resistor of 1k

```
void setup() {
pinMode(2,INPUT_PULLUP);
pinMode (3,OUTPUT);
}
void loop() {
   // put your main code here, to run repeatedly:
   if (digitalRead(2) == LOW)
   {
     digitalWrite(3,HIGH);
   }
   else
   {
     digitalWrite(3,LOW);
   }
}
```

Motion Detector Using PIR Sensor

Scan me for Code, Circuit Diagram and more details on this project

Discover: Digital Read, Digital Write Difficulty: 1/5

Circuit Diagram



fritzing



Detecting motions or movements has always been important in most projects. With the help of the PIR Sensor it has become very easy to detect human/animal movements. In this project we will learn how we can interface a PIR Sensor with a microcontroller like Arduino. We will interface an Arduino with PIR module and blink a LED and beep a Buzzer whenever a movement is detected.

We have powered the PIR sensor using the 5V Rail of the Arduino. The output pin of the PIR Sensor is connected to the 2^{nd} digital pin of Arduino. This pin will be the INPUT pin for Arduino. Then the 3^{rd} pin of Arduino is connected to the LED and Buzzer. This pin will act as the output pin of the Arduino. We will program the Arduino to trigger an Output on 3^{rd} pin if an Input has been detected at 2^{nd} pin. The complete Program is explained below.

The program for Arduino is pretty simple and straight forward. To connect Arduino PIR Sensor, we have to assign the pin number 2 as input and pin number 3 as output. Then we have to produce a discontinuous trigger whenever the pin 2 goes high. Each line is explained below.

```
void setup() {
  pinMode(2, INPUT); //Pin 2 as INPUT
  pinMode(3, OUTPUT); //PIN 3 as OUTPUT
}
void loop() {
  if (digitalRead(2) == HIGH)
  {
    digitalWrite(3, HIGH); // turn the LED/Buzz ON
    delay(100); // wait for 100 msecond
    digitalWrite(3, LOW); // turn the LED/Buzz OFF
    delay(100); // wait for 100 msecond
  }
}
```

Three Way Traffic Light Using Arduino

Discover: LEDs, Multiple outputs, Delay Difficulty: 2/5



Scan me for Code, Circuit Diagram and more details on this project

Circuit Diagram





This Arduino based 3-Way Traffic Light Controller is a simple Arduino DIY project which is useful to understand the working of traffic lights which we see around us. We will explore how to program the Arduino for multiple outputs by controlling them simultaneous.

The code for this Arduino Traffic Light Controller Project is also very interesting and fun to learn. We have demonstrated Traffic lights for the 3 ways road and the code glows LED's on all the three sides in a particular sequence, in which the actual Traffic Lights works. Like, at a time, there will be two Red signals on any of the two sides and one Green light on the remaining side. And yellow light will also glow, for 1 second each time, in between transition from Red to Green, means first red light glows for 5 second then yellow light glows for 1 second and then finally green light will be turned on..

```
void setup() {
  // configure the output pins
 pinMode(2,OUTPUT);
 pinMode(3,OUTPUT);
 pinMode(4,OUTPUT);
 pinMode(5,OUTPUT);
 pinMode(6,OUTPUT);
  pinMode(7,OUTPUT);
 pinMode(8,OUTPUT);
 pinMode(9,OUTPUT);
  pinMode(10,OUTPUT);
}
void loop()
{
  digitalWrite(4,1);
  digitalWrite(5,1);
  digitalWrite(8,1);
  digitalWrite(2,0);
  digitalWrite(6,0);
                       //enables the 1st set of signals
  digitalWrite(3,0);
  digitalWrite(7,0);
  digitalWrite(10,0);
  digitalWrite(9,0);
  delay(5000);
}
```

Interfacing LCD with Arduino



Scan me for
 Code, Circuit
 Diagram and
 more details
 on this project

Discover: Library, LCD display Difficulty: 2/5

Circuit Diagram





To establish a good communication between human world and machine world, display units play an important role. And so they are an important part of embedded systems. Display units - big or small, work on the same basic principle. Besides complex display units like graphic displays and 3D displays, one must know working with simple displays like 16x1 and 16x2 units. The 16x1 display unit will have 16 characters and are in one line. This LCD will have 32 characters in total 16in 1st line and another 16 in 2nd line. Here one must understand that in each character there are 5x10=50 pixels so to display one character all 50 pixels must work together. But we need not to worry about that because there is another controller (HD44780) in the display unit which does the job of controlling the pixels. (You can see it in LCD unit; it is the black eye at the back).

In this tutorial we are going to interface a 16x2 LCD with ARDUINO UNO. Unlike normal development boards interfacing a LCD to an ARDUINO is quite easy. Here we don't have to worry about data sending and receiving. We just have to define the pin numbers and it will be ready to display data on LCD.

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface
pins
LiquidCrystal lcd(0, 1, 8, 9, 10, 11); /// REGISTER SELECT
PIN, ENABLE PIN, D4 PIN, D5 PIN, D6 PIN, D7 PIN
void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
}
void loop() {
// set the cursor to column 0, line 1
lcd.print(" CIRCUIT DIGEST");//print name
lcd.setCursor(0, 1); // set the cursor to column 0, line 2
lcd.print("www.circuitdigest.com");//print name
delay(750);//delay of 0.75sec
lcd.scrollDisplayLeft();//shifting data on LCD
lcd.setCursor(0, 0);// set the cursor to column 0, line1
}
```

Interfacing 7-Segment display



Scan me for Code, Circuit Diagram and more details on this project

Discover: Library, 7-Seg display Difficulty: 2/5

Circuit Diagram





To establish a good communication between human world and machine world, display units play an important role. And so they are an important part of embedded systems. Display units - big or small, work on the same basic principle. Besides complex display units like graphic displays and 3D displays, one must know working with simple displays like 16x1 and 16x2 units. The 16x1 display unit will have 16 characters and are in one line. This LCD will have 32 characters in total 16in 1st line and another 16 in 2nd line. Here one must understand that in each character there are 5x10=50 pixels so to display one character all 50 pixels must work together. But we need not to worry about that because there is another controller (HD44780) in the display unit which does the job of controlling the pixels. (You can see it in LCD unit, it is the black eye at the back).

In this tutorial we are going to interface a 16x2 LCD with ARDUINO UNO. Unlike normal development boards interfacing a LCD to a ARDUINO is quite easy. Here we don't have to worry about data sending and receiving. We just have to define the pin numbers and it will be ready to display data on LCD.

```
#define segA 2//connecting segment A to PIN2
#define segB 3// connecting segment B to PIN3
#define segC 4// connecting segment C to PIN4
#define segD 5// connecting segment D to PIN5
#define segE 6// connecting segment E to PIN6
#define segF 7// connecting segment F to PIN7
#define seqG 8// connecting segment G to PIN8
int COUNT=0;//count integer for 0-9 increment
void setup()
{
for (int i=2;i<9;i++) {</pre>
    pinMode(i, OUTPUT);// taking all pins from 2-8 as
output
}
}
/*FULL CODE CAN BE FOUND USING OR CODE OR FROM
GIVEN CD */
```

Controlling RGB LED with Arduino

Scan me for Code, Circuit Diagram and more details on this project

Discover: RGB Led Difficulty: 2/5

Circuit Diagram



fritzing



In this project we are going to interface 5 RGB (Red Green Blue) LEDs to Arduino Uno. These LEDs are connected in parallel for reducing PIN usage of Uno.

The tricky part here is, say we want to turn the RED led in SET1 and GREEN LED in SET2. We power the PIN8 and PIN9 of UNO, and ground PIN7, PIN6.With that flow we will have RED in first SET and GREEN in second SET ON, but we will have GREEN in SET1 and RED in SET2 ON with it. By simple analogy we can seen all four LEDs close the circuit with above configuration and so they all glow. So to eliminate this problem we will turn only one SET on at a time. Say at t=0m SEC, SET1 is tuned ON. At t = 1m SEC, SET1 is tuned OFF and SET2 is turned ON. Again at t=6m SEC, SET5 is turned OFF and SET1 is turned ON. This goes on.

Here the trick is, the human eye cannot capture a frequency more than 30 HZ. That is if a LED goes ON and OFF continuously at a rate of 30HZ or more. The eye sees the LED as continuously ON. However this is not the case. The LED will be constantly turning ON and OFF. This technique is called multiplexing.

```
void setup() {
  for (int i = 0; i < 11; i++) // 11 pins are set as OUTPUT
    pinMode(i, OUTPUT);
  for (int i = 3; i < 8; i++)</pre>
    digitalWrite(i, LOW);//pins3-8 are set low
}
void loop() {
  for (int i = 3; i < 8; i++)</pre>
  Ł
    digitalWrite(i, HIGH); //power each set one time
    for (int j = 8; j < 11; j++)</pre>
      digitalWrite(j, LOW);
      delay(100);
      digitalWrite(j, HIGH); //blink three colors
    }
    digitalWrite(i, LOW); //pull down power set
  }
}
```

Arduino Clap Circuit



Scan me for Code, Circuit Diagram and more details on this project

Discover: Analog Read, High Pass filters Difficulty: 3/5

Circuit Diagram





In this project we are going to make Clapper circuit using the concept of ADC (Analog to Digital Conversion) in ARDUINO UNO. We are going to use a MIC and Uno to sense the sound and trigger a response. This Clap ON Clap OFF switch basically turns ON or OFF the device, by using the clap sound, as switch.

MIC or Microphone is a sound sensing transducer, which basically converts sound energy into electrical energy, so with this sensor we have sound as changing voltage. We usually record or sense sound through this device. This transducer is used in all mobile phones and laptops.

Whenever there is sound, the MIC picks it up and converts it into voltage, linear to the magnitude of sound. So for a higher sound we have higher value and for lower sound we have lower value. This value is first fed to the High Pass Filter for filtration. Then this filtered value is fed to the transistor for amplification and transistor provides the amplified output at the collector. This collector signal is fed to the ADC0 channel of the UNO, for Analog to Digital conversion. And lastly Arduino is programmed to toggle the LED, connected at PIN 7 of PORTD, each time ADC channel A0 goes beyond a particular level.

```
const int analogInPin = A0;
                              // Analog input pin 0
int sensorValue = 0;
void setup()
Ł
DDRD = 0 \times FF;
}
void loop()
Ł
 sensorValue = analogRead(analogInPin); //read ADC value
on channel 0
 if(sensorValue>60)
  PORTD ^=(1<<7); //If there is a peak toggle the LED on and
OFF on pin7.
  delay(250);
  }
}
```

Shift Register with Arduino



Scan me for Code, Circuit Diagram and more details on this project

Discover: Shift Register, for loops Difficulty: 3/5

Circuit Diagram





In ARDUINO we have 20 I/O pins, so we can program 20 pins of UNO to be used as either input or output. Although there are more pins on ATMEGA328P controller than on UNO, this is because while designing the board some pins are defaulted.

Now for some applications we need more than 30 pins, say if we want to design a 5x5x5 LED CUBE, so for this we need 5x5+5=30 pins. For such cases we use serial to parallel converter chips or shift register. A shift register chip takes data from UNO board serially and gives output in 8 bit parallel configuration.

Here we are going to send data in eight bit size through a single channel to shift register. The shift register takes the data serially and stores that data in its memory. Once the data is sent by the controller, we are going to send a command to shift register to show the data at the output, with this command the shift register puts out data parallel.

```
volatile int i=0;
void setup()
{
                pinMode(2, OUTPUT); // sets the pin2 as output
                pinMode(1, OUTPUT); // sets the pin1 as output
                pinMode(0, OUTPUT); // sets the pin0 as output
}
void loop()
{
                for (int i=0;i<255;i++) //if binary count is less</pre>
                  digitalWrite(2, HIGH);
                   shiftOut(0,1,2,i); //send eight bit data serially
                  digitalWrite(2,LOW);
                  delay(500); //wait for half a second
                 }
}
```

Distance measurement using US sensor



Scan me for Code, Circuit Diagram and more details on this project

Discover: Ultrasonic sensor, Pulse In Difficulty: 3/5

Circuit Diagram





Ultrasonic sensors are great tools to measure distance without actual contact and used at several places like water level measurement, distance measurement etc. This is an efficient way to measure small distances precisely. In this project we have used an Ultrasonic Sensor to determine the distance of an obstacle from the sensor. Basic principal of ultrasonic distance measurement is based on ECHO. When sound waves are transmitted in environment then waves are return back to origin as ECHO after striking on the obstacle. So we only need to calculate the travelling time of both sounds means outgoing time and returning time to origin after striking on the obstacle. As speed of the sound is known to us, after some calculation we can calculate the distance.

In circuit connections Ultrasonic sensor module's "trigger" and "echo" pins are directly connected to pin 18(A4) and 19(A5) of arduino. A 16x2 LCD is connected with arduino in 4-bit mode. Control pin RS, RW and En are directly connected to arduino pin 2, GND and 3. And data pin D4-D7 is connected to 4, 5, 6 and 7 of arduino.

```
#include <LiquidCrystal.h>
#define trigger 18
#define echo 19
LiquidCrystal lcd(2,3,4,5,6,7);
float time=0, distance=0;
void setup()
{
 lcd.begin(16,2);
 pinMode(trigger,OUTPUT);
pinMode(echo,INPUT);
 lcd.print(" Ultra sonic");
 lcd.setCursor(0,1);
 lcd.print("Distance Meter");
 delay(2000);
 lcd.clear();
 lcd.print(" Circuit Digest");
 delay(2000);
}
/*COMPLETE PROGRAM GIVEN IN THE QR CODE AND THE CD PROVIDED
```

Distance measurement using US sensor



Scan me for Code, Circuit Diagram and more details on this project

Discover: Ultrasonic sensor, Pulse In Difficulty: 3/5

Circuit Diagram





Ultrasonic sensors are great tools to measure distance without actual contact and used at several places like water level measurement, distance measurement etc. This is an efficient way to measure small distances precisely. In this project we have used an Ultrasonic Sensor to determine the distance of an obstacle from the sensor. Basic principal of ultrasonic distance measurement is based on ECHO. When sound waves are transmitted in environment then waves are return back to origin as ECHO after striking on the obstacle. So we only need to calculate the travelling time of both sounds means outgoing time and returning time to origin after striking on the obstacle. As speed of the sound is known to us, after some calculation we can calculate the distance.

In circuit connections Ultrasonic sensor module's "trigger" and "echo" pins are directly connected to pin 18(A4) and 19(A5) of arduino. A 16x2 LCD is connected with arduino in 4-bit mode. Control pin RS, RW and En are directly connected to arduino pin 2, GND and 3. And data pin D4-D7 is connected to 4, 5, 6 and 7 of arduino.

```
#include <LiquidCrystal.h>
#define trigger 18
#define echo 19
LiquidCrystal lcd(2,3,4,5,6,7);
float time=0, distance=0;
void setup()
{
 lcd.begin(16,2);
 pinMode(trigger,OUTPUT);
pinMode(echo,INPUT);
 lcd.print(" Ultra sonic");
 lcd.setCursor(0,1);
 lcd.print("Distance Meter");
 delay(2000);
 lcd.clear();
 lcd.print(" Circuit Digest");
 delay(2000);
}
/*COMPLETE PROGRAM GIVEN IN THE QR CODE AND THE CD PROVIDED
```

Arduino Digital Voltmeter

Discover: Library, LCD display Difficulty: 4/5

Circuit Diagram



Scan me for Code, Circuit Diagram and more details on this project





With a simple knowledge of Arduino and Voltage Divider Circuit, we can turn the Arduino into Digital Voltmeter and can measure the input voltage using Arduino and a 16x2 LCD display.

Arduino has several analog input pins that connect to an Analog-to-Digital converter (ADC) inside the Arduino. The Arduino ADC is a ten-bit converter, means that the output value will range from 0 to 1023. We will obtain this value by using the *analogRead()* function. If you know the reference voltage you can easily calculate the voltage present at the analog input. We can use voltage divider circuit to calculate the input voltage. Learn more about ADC in Arduino here.

The voltage measured is displayed on the 16x2 Liquid Crystal Display (LCD). We have also displayed the voltage in Serial Monitor of Arduino IDE and confirmed the measured voltage using Multimeter. Pin DB4, DB5, DB6, DB7, RS and EN of LCD are directly conneted to Pin D4, D5, D6, D7, D8, D9 of Arduino Uno

The Center point of two resistors R1 and R2, which makes the voltage divider circuit, is connected to Arduino Pin A0. While the other 2 ends are connected to the input volt (voltage to be measured) and gnd.

Source Code

```
#include <LiquidCrystal.h>
                            // LIBRARY TO ACCESS THE LCD DISPLAY
LiquidCrystal lcd( 4, 5, 6, 7,8,9);
float input volt = 0.0;
float temp=0.0;
float r1=10000.0;
                     //rl value
float r2=100000.0;
                       //r2 value
void setup() {
   Serial.begin (9600);//opens serial port, sets data rate to 9600 bps
   lcd.begin(16, 2); //// set up the LCD's number of columns and rows
   lcd.print("DC DIGI VOLTMETER");
}
void loop()
{
    int analogvalue = analogRead(A0);
    temp = (analogvalue * 5.0) / 1024.0;//FORMULA CONVERT THE VOLTAGE
```

/* FULL CODE CAN BE FOUND ON SOURCE CD OR THROUGH QR CODE */

Light Detector using Arduino

Scan me for Code, Circuit Diagram and more details on this project

Discover: LDR, Relay Module Difficulty: 4/5

Circuit Diagram





We all want our home appliances to be controlled automatically based on some conditions and that's called Home automation. Today we are going to control the light based of darkness outside, the light turns ON automatically when it is dark outside and turns off when it gets bright. For this, we need a light sensor to detect the light condition and some circuitry to control the Light sensor.

LDR is Light Dependent Resistor. LDRs are made from semiconductor materials to enable them to have their light-sensitive properties. Whenever light falls on the surface of the LDR (in this case) the conductance of the element increases or in other words, the resistance of the LDR falls when the light falls on the surface of the LDR. This property of the decrease in resistance for the LDR is achieved because it is a property of semiconductor material used on the surface.

As per the circuit diagram, we have made a voltage divider circuit using LDR and 100k resistor. The voltage divider output is feed to the analog pin of the Arduino. The analog Pin senses the voltage and gives some analog value to Arduino. The analog value changes according to the resistance of LDR. So, as the light falls on the LDR the resistance of it get decreased and hence the voltage value increase.

Source Code

```
#define relay 10
int LED = 9;
int LDR = A0;
void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(relay, OUTPUT);
  pinMode(LDR, INPUT);
  }
void loop() {
  int LDRValue = analogRead(LDR);
  Serial.print("sensor = ");
  Serial.print(LDRValue);
```

/* FULL PROGRAM IS GIVEN IN THE SOURCE CD AND THE QR CODE LINK*/

IR Controlled DC Motor

Scan me for Code, Circuit Diagram and more details on this project

Discover: IR Module, Relay Module, Difficulty: 4/5

Circuit Diagram





This is a simple project in which we use three basic components that are IR Sensor, Relay Module and DC Motor. Here we will interface IR sensor with Arduino to control DC Motor. The IR sensor will detect any object in front of it and Arduino will read the IR Sensor's output and make the Relay High. Relay is further connected to DC Motor, so DC Motor will be ON whenever IR Sensor Detects any objects in front it.

In circuit, IR sensor Module output pin is simple connected to the Pin 2 no of Arduino and Relay Module's input is connected to Pin 7 no of Arduino. Further a DC Motor is connected to the Relay.

Working of this project is straight forward. Whenever there is some object in front of IR sensor, it will detect that and make the output pin high. IR sensor's output pin is connected to Arduino, so Arduino will read it and activate the Relay module by making pin 7 high. As soon as relay is activated, it will turn on the DC motor.

When there is no object near IR sensor, the output of IR sensor will remain low and DC motor will also remain in off state. The sensitivity of IR Sensor can be adjusted using the potentiometer on the module itself. Sensitivity simply means the distance from which it can detect the object.

```
void setup() {
pinMode(2,INPUT);
pinMode(7,OUTPUT);
Serial.begin(9600);
}
void loop() {
    if (digitalRead(2) == 1)
    {
      Serial.println(digitalRead(2));
      digitalWrite(7,HIGH);
    }
else{
      digitalWrite(7,LOW);
      }
}
```

DC Motor Speed Controller

Discover: analogwrite, Potentiometer Difficulty: 4/5

Circuit Diagram



Scan me for Code, Circuit Diagram and more details on this project





For controlling the speed of DC motor we have various methods, but in this project we are controlling DC Motor speed using PWM. In this project we will be able to control the speed of DC motor with potentiometer and we can adjust the speed by rotating the knob of Potentiometer.

PWM is a technique by using we can control the voltage or power. To understand it more simply, if you are applying 5 volt for driving a motor then motor will moving with some speed, now if we reduces applied voltage by 2 means we apply 3 volt to motor then motor speed also decreases. This concept is used in the project to control the voltage using PWM.

For controlling the speed of DC motor, we use a 100K ohm potentiometer to change the duty cycle of the PWM signal. 100K ohm potentiometer is connected to the analog input pin A0 of the Arduino UNO and the DC motor is connected to the 12th pin of the Arduino (which is the PWM pin). The working of Arduino program is very simple, as it reads the voltage from the analog pin A0. The voltage at analog pin is varied by using the potentiometer. After doing some necessary calculation the duty cycle is adjusted according to it.

```
int pwmPin = 12; // assigns pin 12 to variable pwm
int pot = A0; // assigns analog input A0 to variable pot
int c1 = 0; // declares variable c1
int c2 = 0; // declares variable c2
void setup() // setup loop
{
 pinMode (pwmPin, OUTPUT);
  pinMode(pot, INPUT);
}
void loop()
{
  c2= analogRead(pot);
  c1= 1024-c2; // subtracts c2 from 1000 ans saves the
  digitalWrite(pwmPin, HIGH);
  delayMicroseconds(c1);
  digitalWrite(pwmPin, LOW);
  delayMicroseconds(c2);
}
```

Digital Thermometer

Scan me for Code, Circuit Diagram and more details on this project

Discover: LM35, LCD display, Arrays Difficulty: 4/5

Circuit Diagram





Thermometers are useful apparatus being used since long time for temperature measurement. In this project we have made an Arduino based digital thermometer to display the current ambient temperature and temperature changes on a LCD unit in real time. It can be deployed in houses, offices, industries etc. to measure the temperature. This project is based on Arduino which communicates here with LM35 temperature sensor and a 16x2 display unit. We can divide this Arduino based thermometer into three sections - The first senses the temperature by using temperature sensor LM 35, second section converts the temperature value into a suitable numbers in Celsius scale which is done by Arduino, and last part of system displays temperature on LCD.

Arduino reads output voltage of temperature sensor by using Analog pin A0 and performs the calculation to convert this Analog value to a digital value of current temperature. After calculations Arduino sends these calculations or temperature to 16x2 LCD units by using appropriate commands of LCD.

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);
#define sensor A0
byte degree[8] =
              {
                0b00011,
                 0b00011,
                0b00000,
                0b00000,
                 0b00000,
                0b00000,
                0b00000,
                 0b00000
              };
void setup()
{
  lcd.begin(16,2);
  lcd.createChar(1, degree);
  lcd.setCursor(0, 0);
                           ");
  lcd.print("
                Digital
  lcd.setCursor(0,1);
 /*THE COMPETE CODE CAN BE FOUND ON SOURCE CD OR THROUGH THE
                           QR LINK*/
```

Arduino Calculator using Keypad

Discover: Loops, Arrays, Keypad Difficulty: 5/5

Circuit Diagram



Scan me for Code, Circuit Diagram and more details on this project





Programming is always fun and Arduino is a wonderful platform if you are just getting started with Embedded programming. In this tutorial we will build our own calculator with Arduino. The values can be sent in through a keypad (4×4 keypad) and result can be viewed on a LCD screen (16×2 Dot-matrix). This calculator could perform simple operations like Addition, Subtraction, Multiplication and Division with whole numbers. But once you understand the concept you can implement even scientific functions with Arduino's built in functions.

The +5V and ground connection shown in the circuit diagram can be obtained from the 5V and ground pin of the Arduino. The Arduino itself can be powered from your laptop or through the DC jack using a 12V adapter or 9V battery.

We are operating the LCD in 4-bit mode with Arduino so only the last four data bits of the LCD is connected to Arduino. The Keyboard will have 8 output pins which have to be connected from pin 0 to pin 7 as shown above.

```
#include <LiquidCrystal.h> //Header file for LCD from
#include <Keypad.h> //Header file for Keypad from
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'7','8','9','D'},
  {'4','5','6','C'},
  {'1','2','3','B'},
  {'*','0','#','A'}
};
byte rowPins[ROWS] = { 0, 1, 2, 3 };// Connect keypad ROW0, ROW1,
byte colPins[COLS] = { 4, 5, 6, 7 }; // Connect keypad COL0, COL1 and
Keypad kpd = Keypad ( makeKeymap (keys), rowPins, colPins, ROWS, COLS
); // Create the Keypad
const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;
     /*COMPLETE PROGRAM IS GIVEN IN THE QR LINK AND ALSO IN THE CD*/
```

Servo Motor Control using Arduino



Scan me for Code, Circuit Diagram and more details on this project

Discover: Servo Motor, PWM control Difficulty: 3/5

Circuit Diagram





In this tutorial we are going to control a servo motor by ARDUINO UNO. Servo Motors are used where there is a need for accurate shaft movement or position. These are not proposed for high speed applications. These are proposed for low speed, medium torque and accurate position application. These motors are used in robotic arm machines, flight controls and control systems.

A servo motor is a combination of DC motor, position control system, gears. The position of the shaft of the DC motor is adjusted by the control electronics in the servo, based on the duty ratio of the PWM signal the SIGNAL pin. Simply speaking the control electronics adjust shaft position by controlling DC motor. This data regarding position of shaft is sent through the SIGNAL pin. The position data to the control should be sent in the form of PWM signal through the Signal pin of servo motor.

The frequency of PWM (Pulse Width Modulated) signal can vary based on type of servo motor. The important thing here is the DUTY CYCLE of the PWM signal.

```
Source Code
```

```
#include <LiquidCrystal.h> //Header file for LCD from
#include <Keypad.h> //Header file for Keypad from
const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'7','8','9','D'},
  {'4','5','6','C'},
  {'1','2','3','B'},
  { '*', '0', '#', 'A' }
};
byte rowPins[ROWS] = { 0, 1, 2, 3 };// Connect keypad ROW0, ROW1,
byte colPins[COLS] = { 4, 5, 6, 7 }; //Connect keypad COL0, COL1
Keypad kpd = Keypad ( makeKeymap (keys), rowPins, colPins, ROWS,
COLS ); // Create the Keypad
const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;
      /*COMPLETE PROGRAM IS GIVEN IN THE QR LINK AND ALSO IN THE CD*/
```

Bonus Projects

This booklet covers the very basic projects for a beginner to get started with Arduino. Once you have completed all the Projects, you can proceed on to these Bonus Projects. These Projects are meant for people who always want to push an extra mile, again all the Projects mentioned below can be built by just using the components provided through this kit. Scan the QR code to proceed to the project page and start building

S.No	Project Name	QR Code
1	Arduino Based LED Dimmer using PWM	
2	Arduino Ohm Meter	
3	Digital Dice Circuit	
4	Electronic Voting Machine using Arduino	
5	Digital Code Lock using Arduino	

CREDITS

This book would not have been possible if it was not for the strong Arduino community that keeps the platform alive. We also take this page to extend the credits to the following people



www.components101.com

Editor

John Macolf

Assistant Editor

Aisha Caile

Design

Emilie Thomas

We also extend our thanks to the hobbyists namely Antony Jackson, Aswinth Raj, Dilip Raja, Jayant and Pankaj Khatri for sharing their projects on CircuitDigest.

