



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE ON SKETCH WITH ARDUINO

STARTS ON January 2, 2022 In association with TLC

Registration : Free Course Duration : 4 Week Weekend Course (Saturday) Invited Participants: Third Year ECE, EEE, CSE Restricted to 30 Participants/Slot Resource Persons: In-house Trainers

Coordinators Mr. E.Parasuramu 9989575859 Convener Prof.k.Ashok Babu Principal Dr.G.Suresh

HAUS OUSE ON Diate From COURSE Sector with Artuino Bate From COL OL 2022 (3 Week Course) ATTENDATED STUDENTS I BOH 100447 Neek 3 Neek 3 ATTENDATED STUDENTS ATTENDATED STUDENT 5 No Hall Theit Number I 1804100400 Expendio Reaction On on Saturday ATTENDATED STUDENT Week 3 Neek 3 Week 4 Neek 4 5 No Hall Theit Number I 1804100400 Expendio Reaction Attended Attended Attended Attended Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Beneration Ben			DEPARTMENT OF ELECT	RONICS AND COMMU	JNICATION ENGINEERI	NG			
Section with Arduino Date Form 0.01.202 (a Week Course Only on Saturday) Sature Form 0.01.202 (a Week Course Only on Saturday) Sature Form 0.01.202 (a Week Course Only on Saturday) Sature Form 0.01.202 (a Week Course Only on Saturday) ATTEX DATE Sature Form 0.01.202 (a Week Course Only on Saturday) ATTEX DATE ATTEX ATTEX DATE <td cols<="" td=""><td></td><td></td><td>HAI</td><td>NDS ON TRAINING CO</td><td>URSE</td><td></td><td></td><td></td></td>	<td></td> <td></td> <td>HAI</td> <td>NDS ON TRAINING CO</td> <td>URSE</td> <td></td> <td></td> <td></td>			HAI	NDS ON TRAINING CO	URSE			
SHORTUSED STUDENTS SHORTUSED STUDENT Sind Hall Tucket Number NAME OF THE STUDENT Neek 1 1 1804140447 Collikur Teja Vardhan Reddy Tip/NAVCE STUDENT Week 3 Week 4 2 1804140447 Collikur Teja Vardhan Reddy Tip/NAVCE STUDENT Week 3 Week 4 3 1804140447 Collikur Teja Vardhan Reddy Tip/NAVCE Student Methods Tip/NAVCE Student 5 1804140447 Salwate Bharath Kumar Mumber Auch Mana Student Mumber Auch Mana Student Mumber Auch Mana Student Methods 7 1904140427 Bheemreedy Sanjara E San San R Roman Student Methods 8 1904140427 Bheemreedy Sanjara E San San R Roman Student Mumber Auch Mana Student 1 1904140428 Manani Sindhu H.cn/H.eu/H.eu/H.eu/H.eu/H.eu/H.eu/H.eu/H.eu	-20		SI Date: From 02.01	cetch with Ardui 2022 (4 Week Course	i no , Only on Saturday)			5.201	
S-No Hall Ticket Number AME OF THE STUDENT Week 1 Week 2 Week 3 Week 4 1 18D41A0447 Chilkur Teja Varihan Reddy Tij, Luhhan Tij, Luhhan Tij, Luhhan Meek 3 Week 4 2 18D41A0447 Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 3 18D41A0447 Saluvala Bharath Kumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 5 18D41A0447 Saluvala Bharath Kumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 6 18D41A0447 Saluvala Bharath Kumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 7 18D41A0447 Saluvala Bharath Kumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 6 18D41A0447 Saluvala Bharath Kumar Elkyala Aurukumar Elkyala Aurukumar Elkyala Aurukumar 7 18D41A0462 Raminal Bharath Kumar Ru Aurukurar Elkyala Aurukumar Elkyala Aurukumar 8 18D41A0462 Raman Sindhu H. Aurukurar Elkyala Aurukukukurar Elkyala Aurukurar 1 18D41A0463 Raman Sindhu H. Aurukurar Elkyala Aurukukukukukukukukukukukukukukukukukuku				SHORTLISTED STUDEN	TS				
 ³ No. Hall Tickert Number NAME OF THE STUDENT 18041A0447 Chilkur Teja Vardnan Reddy 18041A0447 Stilvanda Elityala Anunkumar 18041A0447 Stilvanda Elityala Anunkumar 18041A0447 Stilvanda <listilvanda< li=""> Stilvanda Stilvanda</listilvanda<>			V	ATTENDANCE SHEET				[
1 18041A0441 Chilkur Teja Vardhan Redoy 70x Vlavdikon 70x Vlavdikon 2 18045A0403 Etkiyala Aunhumar 7 7 7 7 7 7 3 18041A0441 Saluvala Aunhumar 7 7 7 7 7 7 4 17041A0402 Jataveth Subhash Naik 2 140x41 2 7 7 5 18041A0427 Saluvala Bharath Kumar 2 2 7 7 7 7 19041A0427 Saluvala Bharath Kumar 2 2 2 2 2 6 18041A0427 Saluvala Bharath Kumar 2 2 2 2 2 7 19041A0427 Saluvala Bharath Kumar 2 2 2 2 2 7 19041A0428 Kumar 2 2 2 2 2 8 17041A0428 Kamarath Kumar 1 1 1 1 9 17041A0428 Kamarath Kumar 2 2 2 2 10 17041A0448 Kathreek Reddy Kandrek Leddu 1 1 11 17041A0448 Kathreek Reddy Kandrek Leddu 2	S. No	Hall Ticket Number	NAME OF THE STUDENT	Week 1	Week 2	Week 3	Week 4		
2 18D45A0403 Eikyala Arunkumar A.	-	18D41A0447	Chilkur Teja Vardhan Reddy	Toja Vardla	Ter ladea	4	Top Vardha	-	
3 18D41A0490 Kamsethy Vinay Luth of the second momentary Vinay 4 17D41A0402 Jatavath Subhash Naik Juth of the second momentary Vinay 5 18D41A0417 Saluvala Bharath Kumar Juth of the second momentary Vinay 6 19D41A0402 Jatavath Subhash Naik Juth of the second momentary Vinay 7 19D41A0402 Bhenmedy Sanjana R. Saluvala Bharath Kumar R. Sanyara R. Sanyar	2	18D45A0403	Etikyala Arunkumar	8 A	SA.	SAS -	A A		
4 17D41A0402 Jatavath Subhash Naik Julbush Jul	ო	18D41A0490	Kamisetty Vinay	Khin	mary	Marin		100	
5 18D41A04/7 Saluvala Bharath Kumar Nusardh Lund Rusardh Lund Rhardh Rhardh Rhardh Lund Rhardh Rhardh Lund Rhardh Rhardh Lund Rhardh	4	17D41A0402	Jatavath Subhash Naik	Judraph	Jubhast	Jublah	Ruthal	-	
6 19D41A0427 Bheemredy Sanjana P. Sanjana P. Sanjana P. Sanjana 7 19D41A0429 Namani Sindhu H. Scholan P. Sanjana P. Sanjana P. Sanjana 8 18D41A0429 Sowmya Boddu H. Scholan P. Sanjana Ni Sindhu Ni Sindhu 9 17D41A0429 Sowmya Boddu H. Scholan P. Sanjana P. Sanjana 9 17D41A0436 Katni Kartheek Reddy K. moduculuk buk k. Mukuka Rahu K. moteth bodu Rutheek 10 17D41A0448 T. Sanjana Sowny R. Mahuka Rahu K. moteth bodu Rutheek 11 17D41A0448 T. Sanjana Sowny Rowny Manued Rutheek 12 17D41A0448 T. Sanjana Sowny Rutheek Rutheek Rutheek 13 18D45A0432 Janglii Raghavendra Rowny Rutheek Rutheek Rutheek 13 18D45A0432 Janglii Raghavendra Rutheek Rutheek Rutheek Rutheek 13 18D45A0432 Janglii Raghavendra Rutheek Rutheek Rutheek	5	18D41A04J7	Saluvala Bharath Kumar	Pl - A long	Chrandla Vinnex	Ch-Ab V.	the date of	-	
7 19D41A04C9 Namani Sindhu N.S.Alhu N.S.Alhu N.S.Alhu 8 18D41A04C9 Sowmya Boddu N.S.Alhu N.S.Alhu N.S.Alhu 9 17D41A0496 Kalavakuri Mahesh Babu K. Muducuk buk k. Mudu k. Muducuk buk k.	9	19D41A0427	Bheemreddy Sanjana	A.S. Man	To she was		R Conform		
8 18D41A0429 Sowmya Bodu Area Area Area Area Area Area Area 9 17D41A0436 Kalavakuri Mahesh Babu K. neulverk budu K. morek hodku K. imdreek ladu 10 17D41A0444 Kathi Kartheek Reddy K. neulverk budu K. morek hodku K. imdreek ladu 11 17D41A0448 T. Sanjana Sowmy E. W R. Murkek Routheek 12 17D41A0448 T. Sanjana Sowmy E. W Routheek Routheek 13 18D45A0432 Jangili Raghavendra Rowm Rowm Rowm Rowm 14 17D41A04H3 R Sai Abhinav Goud Rowm Rowm Rowm Rowm 15 17D41A04H3 R Sai Abhinav Goud Rowm Rowm Rowm Rowm 16 17D41A04H3 R Sai Abhinav Goud Rowm Rowm Rowm Rowm 16 17D41A04H3 Siddique Rowm Rowm Rowm Rowm 16 17D41A04H2 Vernori Morse Abhinav Arthreak M Rowm 16 17D41A04H2 Kentron Arthreak M Rowm 16 17D41A04H2 Artentron Arthreak M </td <td>7</td> <td>19D41A04C9</td> <td>Namani Sindhu</td> <td>M. C. M.</td> <td>w cale</td> <td>an Crafter</td> <td>and not</td> <td>_</td>	7	19D41A04C9	Namani Sindhu	M. C. M.	w cale	an Crafter	and not	_	
9 17D41A0496 Kalavakuri Mahesh Babu K- Meducut, buk K- Mulusuri Raduu K- Mulusuri Raduu	ø	18D41A0429	Sowmya Boddu	- Jes	(4)	Shine	- A		
10 17D41A04A4 Kathi Kartheek Reddy How The UK Patheek Patheek Partheek 11 17D41A04L8 T.Sanjana Cowy E.W Swy E.W 12 17D41A04L8 T.Sanjana Cowy E.W Swy E.W 13 17D41A04E9 Muppidi Pramod Manwad Markeek Rew Rwy 13 18D45A0432 Jangii Raghavendra Row Row Row Row Row 14 17D41A04H9 R Sai Abhinav Goud Row Row Row Row Row 15 17D41A04H7 Quazi Mohammad Abdul Raheem Awk Awk Awk Awk 16 17D41A04H2 Vemuri Moses Abhishek MSY Awk Awk Awk 17 19D41A0412 Vemuri Moses Abhishek MSY Arkirishe Arkirishe A	6	17D41A0496	Kalavakuri Mahesh Babu	K. Mediedube	k.mahurth Rabu	K. movel bab	t mulsels laber	-	
11 17D41A04L8 T.Sanjana Count	10	17D41A04A4	Kathi Kartheek Reddy	Kartheele	Kartheek	Butheek	Kartheefe	~	
12 17D41A04E9 Muppidi Pramod Manuel Inamed Manuel Inamed Manuel Inamed 13 18D45A0432 Jangili Raghavendra Ramed Manuel Inamed Manuel Inamed 13 18D45A0432 Jangili Raghavendra Ramed Manuel Inamed Manuel Inamed 14 17D41A04H3 R Sai Abhinav Goud R.M.M.M.M.M.M.M.M.M.M.M.M.M.M.M.M.M.M.M	11	17D41A04L8	T.Sanjana	Court	the	Sand.	' Mart		
13 18D45A0432 Jangili Raghavendra Daw Paw 14 17D41A04H9 R Sai Abhinav Goud CarMbhriau (m. Earthhriau) Ran Paw 15 17D41A04H7 Quazi Mohammad Abdul Raheem Ahurk Ahurk Ahurk Ahurk 15 17D41A04H7 Siddique Ahurk Ahurk Ahurk Ahurk 16 17D41A04H2 Vemuri Moses Abhishek MSV APSV Arbit 17 19D41A0412 Anemoni Shirisha Mudhiraj Achisrishe Arbit Ashisrishe	12	17D41A04E9	Muppidi Pramod	James	Manual	pamed	Pramad.		
14 17D41A04H9 R Sai Abhinav Goud Cut Mhunav Cud Cut Mhunav Gud Cut Mhunav Gud 15 17D41A04H7 Quazi Mohammad Abdul Raheem Anver Auver Auver Auver 16 17D41A04H2 Vemuri Moses Abhishek MASY MASY V mSy MAN 17 19D41A0412 Anemoni Shirisha Mudhiraj Arbiteche Arbiteche Arbiteche Arbiteche	13	18D45A0432	Jangili Raghavendra	Dan	Jarl	James 1	Dard		
15 17D41A04H7 Quazi Mohammad Abdul Raheem Alwer Alwer 16 17D41A04N2 Vemuri Moses Abhishek MMSV MMSV MMSV 17 19D41A04N2 Vemuri Moses Abhishek MMSV MMSV MSV MSV 17 19D41A0412 Anemoni Shirisha Mudhiraj M. Chiarishe A. Shirishe A. Shirishe A. Shirishe	14	17D41A04H9	R Sai Abhinav Goud	Carlebuiar In	Cei Adriadional	Gift, informe	1 S. Abhieverad	1	
16 17D41A04N2 Vemuri Moses Abhishek MASY MASY V MS M 17 19D41A0412 Anemoni Shirisha Mudhiraj //dhishcha //dhis	15	17D41A04H7	Quazi Mohammad Abdul Raheem	+ time-	Aluk	Almer	Ahur		
17 19D41A0412 Anemoni Shirisha Mudhiraj A.Shirisha A.Shirisha A.Shirisha A.Shirisha A.Shirisha	16	17D41A04N2	Vemuri Moses Abhishek	VES	1840	24	E I	120	
	17	19D41A0412	Anemoni Shirisha Mudhiraj	1. chische	A. shirishe	Architricha	A Shash	2.2	

1.2.2.							2 7 A A A	1	1		T	-	T	- A STR
T. Kawa	, Rangazaza		·	Cainally.	shive	Frankler.	mon	Mahool	Rudhinka,	Ninigh Paddy.	Delarge	Rathy kura	Lazya (feddig.	· (br
T. Kenne	Ruthganas	sisi	- The	Sairath	Shive	Grayathi	onent	Mahal	Rudhinke	Nunighe feddy	Dedeepye	Prattin paren	Lagya Peddy	30
T.Kawe	Rang arady.	212	Mit	Cainath.	shira	Sayati	-thmm.	Mahegh	Rudhinke	Nimithe fedly	Dedeepye	Practine Livan	Lasy Reddy	R
T. Kawler	DONDOND A	43 42	PO MA	Sairet	shive	(Reyeler:	madure	Maheph	Publicko	Nemisha Padly	Dedeepye	Prathic pice	[asye Reddy	(F)
T Kavya	Rangaraju Rahul	Dachepally Saisri	Lunavath Rahul Naik	B.Sainath	Pasunuri Shiva	Buduru Gayatri	Ashwala Madhuri	Mahesh Bhukya	Rudhinika	Nimisha Reddy	Beemagani Dedeepya	L. Pruthvi Kiran	Kontham Lasya Reddy	
17D41A04L7	17D41A04J5	19D41A0441	19D41A04A0	19D41A0435	17D41A04G9	19D41A0436	19D41A0417	19D41A04A3	18D41A0451	17D41A04G1	19D41A0424	17D41A04C5	17D41A04B2	
18	19	20	21	22	23	24	25	26	27	28	29	30	31	

NCLAULT

Convener

Son h

Coordinator







DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDS ON TRAINING COURSE ON SKETCH WITH ARDUINO

STARTS ON January 2, 2022

In association with TLC

Registration : Free Course Duration : 4 Week Weekend Course (Saturday) Invited Participants: Third Year ECE, EEE, CSE Restricted to 30 Participants/Slot Resource Persons: In-house Trainers

Coordinators Mr. E.Parasuramu 9989575859 Convener Prof.k.Ashok Babu Principal Dr.G.Suresh

What is Arduino?

- Arduino is an open-source electronics platform used for building electronics projects.
- · Arduino consists of both a physical programmable circuit board or microcontroller and a software IDE (Integrated Development Environment) that runs on the computer.
- · It is used to write and upload computer code to the physical board.
- · It is intended for making interactive projects.
- · Download Arduino IDE from www.arduino.cc

Features of Arduino IDE

- Works on Linux, Windows and Mac operating systems
- Has many in-built functions that make programming simple and easy
- · Easy to write code and upload it to the physical board
- · Arduino IDE can be used with any Arduino board
- Can be easily adapted for IoT applications
- · Arduino can be turned into IoT product by adding ESP8266 wifi module

Benefits of using Arduino Kit

- · Arduino boards are less expensive compared to other microcontrollers platform.
- . The Arduino programming environment is easyto-use for beginners.
- · For advanced users, the language can be expanded through C++ libraries and AVR-GCC programming language can be added to Arduino programs.
- The modules are published under a Creative Commons license, so circuit designers can make their own version of the module.

- · Arduino platform was designed for hobbyists, students and professionals to create IoT applications that play in the human interface world using sensors, motors, etc.
- · Arduino can interact with buttons, LEDs, LCDs, motors, speakers, cameras, TV and smartphones, etc.
- · Arduino can be connected to one or more sensors to capture the data.



Arduino serie

- **Basic Level** · Overview of Arduino
- · Electronic components and connections
- Introduction to Arduino
- Arduino components and IDE
- First Arduino Program
- Arduino with Tricolor LED and Push button Arduino with LCD
- Display counter using Arduino
- · Seven segment display Pulse Width Modulation
- Analog to Digital Conversion
- Wireless Connectivity to Arduino

Intermediate Level

- Digital logic design with Arduino
- AVR-GCC programming through Arduino Interfacing LCD through AVR-GCC
- programming

Popular uses of Arduino

- · Home automation (controlling lights, fans and other appliances) via Android smartphone
- Traffic light control
- · PC controlled robotic arm
- Temperature controller
- · Anti-theft camera system
- Automated irrigation system
- Feeder for Aquarium
- Garage parking
- · Line follower robot

Components required to practise

Arduino

- 1. Arduino UNO or Compatible Board (1 no.)
- 2. USB Power Cable (1 no.)
- 3. Resistor 220 ohms (6 nos.)
- 4. Resistor 10K Ohms (2 nos.)
- 5. Resistor 1K Ohms (4 nos.)
- 6. Breadboard (1 no.)
- 7. Tricolor LED Common Cathode (1 no.)
- 8. Red LED Common Cathode (1 no.)
- 9. Seven segment display Common cathode (1 no.)
- 10. Seven segment display Common anode (1 no.)
- 11 Decoder IC 7447 (1 no.)
- 12. LCD 16 X 2 soldered with pin header (1 no.)
- 13. Jumper wires Male to Male (20 nos.)
- 14. Jumper wires Male to Female (8 nos.)
- 15. Potentiometer 10K Ohms (1 no.)
- 16. ESP8266 es01 WiFi Black color Module (1 no.)
- 17. DHT11 Temp_Humidity Sensor Module (1 no.)
- 18. L293D H-Bridge Motor driver IC (1 no.)
- 19. Toy Motor (1 no.)
- 20. Buzzer (1 no.)
- 21. Push Button Switch (2 nos.)

The easiest way for beginners to get started with Arduino is by creating circuits using a solderless breadboard. These simple projects will teach you the basics of Arduino Uno, electronics and programming. In this tutorial, you will be creating circuits using the following electronic components:

- LED •
- **RGB LED**
- Temp Sensor
- Pushbutton •
- Potentiometer .
- Photoresistor •
- Servo
- Motor •
- Buzzer •
- LCD screen •

This tutorial is going to allow you to jump right in and start building circuits. If you need some background on the Arduino Uno board or the tools that are needed, please check out post – Arduino Uno For Beginners.

Mixing Assembly and C programming

Assembly programming through Arduino

Getting Started

Before you can start working with Arduino, you need to make sure you have the IDE software installed on your computer. This program allows you to write, view and upload the code to your Arduino Uno board. You can download the IDE for free on Arduino's website.

Once the IDE is installed, you will need to connect your Arduino to your computer. To do this, plug one end of the USB cable to the Arduino Uno and then the other end of the USB to your computer's USB port.

Select The Board

Once the board is plugged in, you will need to open the IDE and click on **Tools** > **Board** > **Arduino Uno** to select the board.

💿 Makerspaces_co	m Arduino 1.8.1			— —	I	×
File Edit Sketch To	ols Help					
Makerspaces	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T			£	2
// This is th	Serial Monitor Serial Plotter WiFi101 Firmware Updater	Ctrl+Shift+M Ctrl+Shift+L	-			^
<pre>void setup() // put your }</pre>	Board: "Arduino/Genuino Uno" Port Get Board Info	2		Boards Manager Arduino AVR Boards Arduino Yún		
void loop() { // put your	Programmer: "AVRISP mkil" Burn Bootloader	2	•	Arduino/Genuino Uno Arduino Duemilanove or Diecimila Arduino Nano		
}				Arduino Mario Arduino Mega or Mega 2560 Arduino Mega ADK Arduino Leonardo Arduino Leonardo ETH Arduino/Genuino Micro		

Select Serial Port

Next, you have to tell the Arduino which port you are using on your computer. To select the port, go to **Tools > Port** and then select the port that says **Arduino.**

🥺 Makerspaces	_com Arduino 1.8.1			— C		\times	
File Edit Sketch	Tools Help						
	Auto Format	Ctrl+T				Ø	
Makerspaces_	Fix Encoding & Reload				l	•	
	Serial Monitor	Ctrl+Shift+M					^
// This is th	Serial Plotter	Ctrl+Shift+L					
	WiFi101 Firmware Updater						
<pre>void setup()</pre>	Board: "Arduino/Genuino Uno"	>			1		
// put your	Port: "COM4 (Arduino/Genuino Uno)"	3		Serial ports			
}	Get Board Info		~	COM4 (Arduino/Genuino Uno)			
<pre>void loop() {</pre>	Programmer: "AVRISP mkll"	>					
// put your	Burn Bootloader						
}							

Project Code

To complete the projects in this tutorial, you will need to download the project code which are known as sketches. A sketch is simply a set of instructions that tells the board what functions it needs to perform. For some of these projects, we are using open-source code that was released by the good people at Sparkfun and Arduino. Use the link below to download the zip folder containing the code. Download Project Code – (ZIP File)

Once the file has been downloaded, you will need to unzip/extract the folder in order to use it.

#1 – Test Arduino

The first project is one of the most basic and simple circuits you can create with Arduino. This project will test your Arduino by blinking an LED that is connected directly to the board.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) LED 5mm
- (1) 220 Ω Resistor

Project Diagram



fritzing

Project Steps

- 1. Twist a 220 Ω resistor to the long leg (+) of the LED.
- 2. Push the short leg of the LED into the ground (GND) pin on the board.
- 3. Push the resistor leg that's connected to the LED into the #13 pin.

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_01_TestArduino
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#2 – Blink an LED

This project is identical to project #1 except that we will be building it on a breadboard. Once complete, the LED should turn on for a second and then off for a second in a loop.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires



Made with fritzing

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_02_Blink
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#3 – Push Button

Using a push button switch, you will be able to turn on and off an LED.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) 10K Ω Resistor
- (1) Push Button Switch
- (6) Jumper Wires

Project Diagram



Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_03_Pushbutton

- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#4 – Potentiometer

Using a potentiometer, you will be able to control the resistance of an LED. Turning the knob will increase and decrease the frequency the LED blinks.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (1) Potentiometer (10k Trimpot)
- (6) Jumper Wires

Project Diagram





Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_04_Potentiometer
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#5 – Fade an LED

By using a PWM pin on the Arduino, you will be able to increase and decrease the intensity of brightness of an LED.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LED 5mm
- (1) 220 Ω Resistor
- (2) Jumper Wires



- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_05_Fade
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#6 – Scrolling LED

This project will blink 6 LEDs, one at a time, in a back and forth formation. This type of circuit was made famous by the show Knight Rider which featured a car with looping LEDs.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (6) LED 5mm
- (6) 220 Ω Resistor
- (7) Jumper Wires



- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_06_Scrolling
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#7 – Bar Graph

Using a potentiometer, you can control a series of LEDs in a row. Turning the potentiometer knob will turn on or off more of the LEDs.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) Potentiometer Rotary
- (10) LED 5mm
- (10) 220 Ω Resistor
- (11) Jumper Wires

Project Diagram



Madewith

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_07_BarGraph
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#8 – Multiple LEDs

This project will use 8 pins on the Arduino board to blink 8 LEDs at the same time.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (8) LED 5mm
- (8) 330 Ω Resistor
- (9) Jumper Wires



- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_08_MultipleLEDs
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#9 – RGB LED

This project will be using an RGB LED to scroll through a variety of colors. RGB stands for Red, Green and Blue and this LED has the ability to create nearly unlimited color combinations.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) RGB LED
- (3) 330 Ω Resistor
- (5) Jumper Wires

Project Diagram



Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_09_RGBLED

- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#10 – Photoresistor

A photoresistor changes the resistance a circuit gets based on the amount of light that hits the sensor. In this project, the brightness of the LED will increase and decrease based on the amount of light present.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LED 5mm
- (1) 330 Ω Resistor
- (1) 10K Ω Resistor
- (1) Photoresistor
- (6) Jumper Wires

Project Diagram



Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_10_Photoresistor
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#11 – Temp. Sensor

A temperature sensor measures ambient temperatures of the world around it. In this project, we will be displaying the temperature in the serial monitor of the Arduino IDE.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) Temperature Sensor TMP36
- (5) Jumper Wires



- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_11_TempSensor
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#12 – Tone Melody

The project will use a piezo buzzer/speaker to play a little melody.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) Piezo Buzzer/Speaker
- (2) Jumper Wires

Project Diagram



fritzing

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_12_ToneMelody
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#13 – Servo

In this project, you will be able to sweep a servo back and forth through its full range of motion.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) Servo
- (6) Jumper Wires

Project Diagram



Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_13_Servo
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#14 – Motor

Using a switching transistor, we will be able to control a DC motor. If everything is connected correctly, you should see the motor spinning.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) DC Motor
- (1) 330 Ω Resistor
- (1) Diode 1N4148
- (1) NPN Transistor
- (6) Jumper Wires



Made with fritzing

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_14_Motor
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

#15 – LCD Screen

An LCD is a liquid crystal display that is able to display text on its screen. In this project, you should see the words "hello,world!" displayed on the screen. The potentiometer is used to adjust the contrast of the display.

Parts Needed

- (1) Arduino Uno
- (1) USB A-to-B Cable
- (1) Breadboard Half Size
- (1) LCD Screen
- (1) Potentiometer
- (16) Jumper Wires

Project Diagram



fritzing

Project Code

- 1. Connect the Arduino board to your computer using the USB cable.
- 2. Open project code Circuit_15_LCD
- 3. Select the board and serial port as outlined in earlier section.
- 4. Click upload button to send sketch to the Arduino.

Troubleshooting

- Make sure your board and serial port is selected in the IDE. To do this, plug your board in and go to **Tools > Board >Arduino** to select your board. Next, go to **Tools > Port >Com** (Arduino) to select your serial port.
- The long leg of the LED is the (+) positive and the short leg is the (-) negative. Make sure the correct leg of the LED is in the proper pin of the Arduino or breadboard as directed.
- It can be easy to put a component or jumper into the wrong pin on the Arduino or the breadboard. Double check the correct pin is being used.

Experiment 1: Turn an LED

Turn an LED on for one second, off for one second, and repeat forever.

```
void setup()
{
    pinMode(13, OUTPUT);
}
void loop()
{
    digitalWrite(13, HIGH); // Turn on the LED
    delay(1000); // Wait for one second
    digitalWrite(13, LOW); // Turn off the LED
    delay(1000); // Wait for one second
}
/*
```

```
Experiment 2: Turns on and off LED
```

Turns on and off a light emitting diode(LED) connected to digital pin 13, when pressing a pushbutton attached to pin 2.

The circuit:

- * LED attached from pin 13 to ground
- * pushbutton attached to pin 2 from +5V
- * 10K resistor attached to pin 2 from ground

* Note: on most Arduinos there is already an LED on the board attached to pin 13.

// set pin numbers: const int buttonPin = 2; // the number of the pushbutton pin const int ledPin = 13; // the number of the LED pin

// variables will change: int buttonState = 0; // variable for reading the pushbutton status

void setup() {
 // initialize the LED pin as an output:

```
pinMode(ledPin, OUTPUT);
// initialize the pushbutton pin as an input:
pinMode(buttonPin, INPUT);
}
void loop() {
// read the state of the pushbutton value:
```

```
buttonState = digitalRead(buttonPin);
```

```
// check if the pushbutton is pressed.
// if it is, the buttonState is HIGH:
if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
    else {
        // turn LED off:
        digitalWrite(ledPin, LOW);
    }
}
```

Experiment 3: Display RGB LED

const int RED_PIN = 9; const int GREEN_PIN = 10; const int BLUE_PIN = 11;

const int DISPLAY_TIME = 1000; // used in mainColors() to determine the // length of time each color is displayed.

```
void setup()
            //Configure the Arduino pins to be outputs to drive the LEDs
pinMode(RED_PIN, OUTPUT);
pinMode(GREEN_PIN, OUTPUT);
pinMode(BLUE PIN, OUTPUT);
}
void loop()
{
              // Red, Green, Blue, Yellow, Cyan, Purple, White
mainColors();
// showSpectrum(); // Gradual fade from Red to Green to Blue to Red
}
* void mainColors()
* This function displays the eight "main" colors that the RGB LED
* can produce. If you'd like to use one of these colors in your
* own sketch, you can copy and paste that section into your code.
void mainColors()
{
// all LEDs off
 digitalWrite(RED_PIN, LOW);
 digitalWrite(GREEN_PIN, LOW);
```

digitalWrite(BLUE_PIN, LOW); delay(DISPLAY_TIME);

// Red digitalWrite(RED_PIN, HIGH); digitalWrite(GREEN_PIN, LOW); digitalWrite(BLUE_PIN, LOW); delay(DISPLAY_TIME);

// Green digitalWrite(RED_PIN, LOW); digitalWrite(GREEN_PIN, HIGH); digitalWrite(BLUE_PIN, LOW); delay(DISPLAY_TIME);

// Blue

digitalWrite(RED_PIN, LOW); digitalWrite(GREEN_PIN, LOW); digitalWrite(BLUE_PIN, HIGH); delay(DISPLAY_TIME);

// Yellow (Red and Green)
digitalWrite(RED_PIN, HIGH);
digitalWrite(GREEN_PIN, HIGH);
digitalWrite(BLUE_PIN, LOW);
delay(DISPLAY_TIME);

// Cyan (Green and Blue)
digitalWrite(RED_PIN, LOW);
digitalWrite(GREEN_PIN, HIGH);
digitalWrite(BLUE_PIN, HIGH);
delay(DISPLAY_TIME);

// Purple (Red and Blue)
digitalWrite(RED_PIN, HIGH);
digitalWrite(GREEN_PIN, LOW);
digitalWrite(BLUE_PIN, HIGH);
delay(DISPLAY_TIME);

// White (turn all the LEDs on)
digitalWrite(RED_PIN, HIGH);
digitalWrite(GREEN_PIN, HIGH);
digitalWrite(BLUE_PIN, HIGH);
delay(DISPLAY_TIME);

```
* void showSpectrum()
```

*

}

* Steps through all the colors of the RGB LED, displaying a rainbow.

* showSpectrum() calls a function RGB(int color) that translates a number

* from 0 to 767 where 0 = all RED, 767 = all RED

*

```
* Breaking down tasks down into individual functions like this
* makes your code easier to follow, and it allows.
* parts of your code to be re-used.
void showSpectrum()
ł
 for (int x = 0; x \le 767; x++)
 ł
            // Increment x and call RGB() to progress through colors.
  RGB(x);
  delay(10); // Delay for 10 ms (1/100th of a second) - to help the "smoothing"
 }
}
* void RGB(int color)
* RGB(###) displays a single color on the RGB LED.
* Call RGB(###) with the number of a color you want
* to display. For example, RGB(0) displays pure RED, RGB(255)
* displays pure green.
* This function translates a number between 0 and 767 into a
* specific color on the RGB LED. If you have this number count
* through the whole range (0 to 767), the LED will smoothly
* change color through the entire spectrum.
*
* The "base" numbers are:
*0 = pure red
* 255 = pure green
* 511 = pure blue
*767 = pure red (again)
* Numbers between the above colors will create blends. For
* example, 640 is midway between 512 (pure blue) and 767
* (pure red). It will give you a 50/50 mix of blue and red,
* resulting in purple.
void RGB(int color)
ł
int redIntensity;
 int greenIntensity;
 int blueIntensity;
 color = constrain(color, 0, 767); // constrain the input value to a range of values from 0 to 767
  // if statement breaks down the "color" into three ranges:
                  // RANGE 1 (0 - 255) - red to green
 if (color \le 255)
 {
  redIntensity = 255 - color; // red goes from on to off
  greenIntensity = color;
                         // green goes from off to on
  blueIntensity = 0;
                        // blue is always off
```

```
}
```

```
else if (color \leq 511) // RANGE 2 (256 - 511) - green to blue
{
 redIntensity = 0;
                              // red is always off
                                   // green on to off
 greenIntensity = 511 - color;
 blueIntensity = color - 256;
                                  // blue off to on
}
else
                // RANGE 3 ( >= 512)- blue to red
{
 redIntensity = color - 512;
                                 // red off to on
 greenIntensity = 0;
                              // green is always off
 blueIntensity = 767 - color;
                                 // blue on to off
}
// "send" intensity values to the Red, Green, Blue Pins using analogWrite()
analogWrite(RED PIN, redIntensity);
```

```
analogWrite(RED_PIN, redIntensity);
analogWrite(GREEN_PIN, greenIntensity);
analogWrite(BLUE_PIN, blueIntensity);
}
```

Experiment 4: Dancing LED

int ledPins[] = {2,3,4,5,6,7,8,9}; // Defines an array to store the pin numbers of the 8 LEDs. // An array is like a list variable that can store multiple numbers. // Arrays are referenced or "indexed" with a number in the brackets []. See the examples in // the pinMode() functions below.

```
void setup()
```

```
{
```

ł

```
// setup all 8 pins as OUTPUT - notice that the list is "indexed" with a base of 0.
pinMode(ledPins[0],OUTPUT); // ledPins[0] = 2
pinMode(ledPins[1],OUTPUT); // ledPins[1] = 3
pinMode(ledPins[2],OUTPUT); // ledPins[2] = 4
pinMode(ledPins[3],OUTPUT); // ledPins[3] = 5
pinMode(ledPins[4],OUTPUT); // ledPins[3] = 6
pinMode(ledPins[5],OUTPUT); // ledPins[5] = 7
pinMode(ledPins[6],OUTPUT); // ledPins[6] = 8
pinMode(ledPins[7],OUTPUT); // ledPins[7] = 9
```

```
void loop()
{
```

// This loop() calls functions that we've written further below.
// We've disabled some of these by commenting them out (putting
// "//" in front of them). To try different LED displays, remove
// the "//" in front of the ones you'd like to run, and add "//"
// in front of those you don't to comment out (and disable) those
// lines.

oneAfterAnother(); // Light up all the LEDs in turn

//oneOnAtATime(); // Turn on one LED at a time

```
//pingPong(); // Same as oneOnAtATime() but change direction once LED reaches edge
//marquee(); // Chase lights like you see on theater signs
//randomLED(); // Blink LEDs randomly
}
```

```
* oneAfterAnother()
* This function turns all the LEDs on, pauses, and then turns all
* the LEDS off. The function takes advantage of for() loops and
* the array to do this with minimal typing.
void oneAfterAnother()
ł
int index:
int delayTime = 100; // milliseconds to pause between LEDs
          // make this smaller for faster switching
// Turn all the LEDs on:
 for(index = 0; index \leq 7; index = ++index) // step through index from 0 to 7
 ł
 digitalWrite(ledPins[index], HIGH);
 delay(delayTime);
 }
// Turn all the LEDs off:
 for(index = 7; index \ge 0; index = --index) // step through index from 7 to 0
 {
 digitalWrite(ledPins[index], LOW);
 delay(delayTime);
 }
}
* oneOnAtATime()
* This function will step through the LEDs, lighting only one at
* a time. It turns each LED ON and then OFF before going to the
* next LED.
void oneOnAtATime()
{
int index:
int delayTime = 100; // milliseconds to pause between LEDs
          // make this smaller for faster switching
 for(index = 0; index \leq 7; index = ++index) // step through the LEDs, from 0 to 7
 {
```

```
digitalWrite(ledPins[index], HIGH); // turn LED on
 delay(delayTime);
                          // pause to slow down
 digitalWrite(ledPins[index], LOW); // turn LED off
 }
}
* pingPong()
* This function will step through the LEDs, lighting one at at
* time in both directions. There is no delay between the LED off
* and turning on the next LED. This creates a smooth pattern for
* the LED pattern.
void pingPong()
int index;
int delayTime = 100; // milliseconds to pause between LEDs
for(index = 0; index \leq 7; index = ++index) // step through the LEDs, from 0 to 7
 {
  digitalWrite(ledPins[index], HIGH); // turn LED on
 delay(delayTime);
                  // pause to slow down
 digitalWrite(ledPins[index], LOW); // turn LED off
 }
 for(index = 7; index \ge 0; index = --index) // step through the LEDs, from 7 to 0
 ł
 digitalWrite(ledPins[index], HIGH); // turn LED on
 delay(delayTime);
                          // pause to slow down
 digitalWrite(ledPins[index], LOW); // turn LED off
 }
}
* marquee()
* This function will mimic "chase lights" like those around
* theater signs.
void marquee()
{
int index;
int delayTime = 200; // milliseconds to pause between LEDs
// Step through the first four LEDs
// (We'll light up one in the lower 4 and one in the upper 4)
 for(index = 0; index \leq 3; index++) // Step from 0 to 3
 ł
 digitalWrite(ledPins[index], HIGH); // Turn a LED on
 digitalWrite(ledPins[index+4], HIGH); // Skip four, and turn that LED on
 delay(delayTime);
                           // Pause to slow down the sequence
```

```
digitalWrite(ledPins[index], LOW); // Turn the LED off
 digitalWrite(ledPins[index+4], LOW); // Skip four, and turn that LED off
 }
}
* randomLED()
*
* This function will turn on random LEDs. Can you modify it so it
* also lights them for random times?
void randomLED()
{
int index;
int delayTime;
 index = random(8); // pick a random number between 0 and 7
 delayTime = 100;
 digitalWrite(ledPins[index], HIGH); // turn LED on
 delay(delayTime);
                       // pause to slow down
digitalWrite(ledPins[index], LOW); // turn LED off
}
Experiment 5: Running Motor
```

```
void setup()
{
    pinMode(motorPin, OUTPUT); // set up the pin as an OUTPUT
    Serial.begin(9600); // initialize Serial communications
}
```

void loop()

```
{ // This example basically replicates a blink, but with the motorPin instead.
int onTime = 3000; // milliseconds to turn the motor on
int offTime = 3000; // milliseconds to turn the motor off
```

```
analogWrite(motorPin, 255); // turn the motor on (full speed)
delay(onTime); // delay for onTime milliseconds
analogWrite(motorPin, 0); // turn the motor off
delay(offTime); // delay for offTime milliseconds
```

 $/\!/$ Uncomment the functions below by taking out the $/\!/.$ Look below for the $/\!/$ code examples or documentation.

```
// speedUpandDown();
// serialSpeed();
```

```
}
```

```
// This function accelerates the motor to full speed,
// then decelerates back down to a stop.
void speedUpandDown()
{
 int speed;
 int delayTime = 20; // milliseconds between each speed step
 // accelerate the motor
 for(speed = 0; speed \leq 255; speed++)
 {
  analogWrite(motorPin,speed);
                                        // set the new speed
  delay(delayTime);
                         // delay between speed steps
 }
 // decelerate the motor
 for(speed = 255; speed >= 0; speed--)
 {
  analogWrite(motorPin,speed);
                                         // set the new speed
  delay(delayTime);
                        // delay between speed steps
 }
}
// Input a speed from 0-255 over the Serial port
void serialSpeed()
ł
 int speed;
 Serial.println("Type a speed (0-255) into the box above,");
 Serial.println("then click [send] or press [return]");
 Serial.println(); // Print a blank line
 // In order to type out the above message only once,
 // we'll run the rest of this function in an infinite loop:
 while(true) // "true" is always true, so this will loop forever.
 {
  // Check to see if incoming data is available:
  while (Serial.available() > 0)
   speed = Serial.parseInt(); // parseInt() reads in the first integer value from the Serial Monitor.
   speed = constrain(speed, 0, 255); // constrains the speed between 0 and 255
                         // because analogWrite() only works in this range.
   Serial.print("Setting speed to "); // feedback and prints out the speed that you entered.
   Serial.println(speed);
   analogWrite(motorPin, speed); // sets the speed of the motor.
 }
}
```

Experiment 6: Potentiometer

int sensorPin = A0; // select the input pin for the potentiometer

```
int ledPin = 13; // select the pin for the LED
int sensorValue = 0; // variable to store the value coming from the sensor
void setup() {
// declare the ledPin as an OUTPUT:
pinMode(ledPin, OUTPUT);
}
void loop() {
// read the value from the sensor:
 sensorValue = analogRead(sensorPin);
 // turn the ledPin on
 digitalWrite(ledPin, HIGH);
 // stop the program for <sensorValue> milliseconds:
 delay(sensorValue);
 // turn the ledPin off:
 digitalWrite(ledPin, LOW);
// stop the program for for <sensorValue> milliseconds:
 delay(sensorValue);
}
```

Experiment 7: Scrolling LED

int time = 100;// The higher the number, the slower the timing. void setup() { // use a for loop to initialize each pin as an output: for (int thisPin = 2; thisPin < 8; thisPin++) { pinMode(thisPin, OUTPUT); } } void loop() { // loop from the lowest pin to the highest: for (int this Pin = 2; this Pin < 8; this Pin++) { // turn the pin on: digitalWrite(thisPin, HIGH); delay(timer); // turn the pin off: digitalWrite(thisPin, LOW); } // loop from the highest pin to the lowest: for (int this Pin = 7; this Pin >= 2; this Pin--) { // turn the pin on: digitalWrite(thisPin, HIGH); delay(timer); // turn the pin off: digitalWrite(thisPin, LOW); } }

Experiment 8: Potentiometer

int sensorPin = A0; // select the input pin for the potentiometer int ledPin = 13; // select the pin for the LED int sensorValue = 0; // variable to store the value coming from the sensor

```
void setup() {
    // declare the ledPin as an OUTPUT:
    pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    // turn the ledPin on
    digitalWrite(ledPin, HIGH);
    // stop the program for <sensorValue> milliseconds:
    delay(sensorValue);
    // turn the ledPin off:
    digitalWrite(ledPin, LOW);
    // stop the program for for <sensorValue> milliseconds:
    delay(sensorValue);
}
```

Experiment 9: LED with PWM

```
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);
}
```

Experiment 10: To measure the temperature sensor's // signal pin.

```
const int temperature Pin = A0;
```

void setup()

{

Serial.begin(9600); //Initialize serial port & set baud rate to 9600 bits per second (bps)

}

void loop()

{

float voltage, degreesC, degreesF; //Declare 3 floating point variables

voltage = getVoltage(temperaturePin); //Measure the voltage at the analog pin

degreesC = (voltage - 0.5) * 100.0; // Convert the voltage to degrees Celsius

degreesF = degreesC * (9.0 / 5.0) + 32.0; //Convert degrees Celsius to Fahrenheit

//Now print to the Serial monitor. Remember the baud must be 9600 on your monitor!
// These statements will print lines of data like this:
// "voltage: 0.73 deg C: 22.75 deg F: 72.96"

Serial.print("voltage: "); Serial.print(voltage); Serial.print(" deg C: "); Serial.print(degreesC); Serial.print(" deg F: "); Serial.println(degreesF);

delay(1000); // repeat once per second (change as you wish!)

float getVoltage(int pin)

//Function to read and return

//floating-point value (true voltage)
//on analog pin

{

return (analogRead(pin) * 0.004882814); // This equation converts the 0 to 1023 value that analogRead() // returns, into a 0.0 to 5.0 value that is the true voltage // being read at that pin.

}

// Other things to try with this code:

// Turn on an LED if the temperature is above or below a value.

// Read that threshold value from a potentiometer - now you've

// created a thermostat!