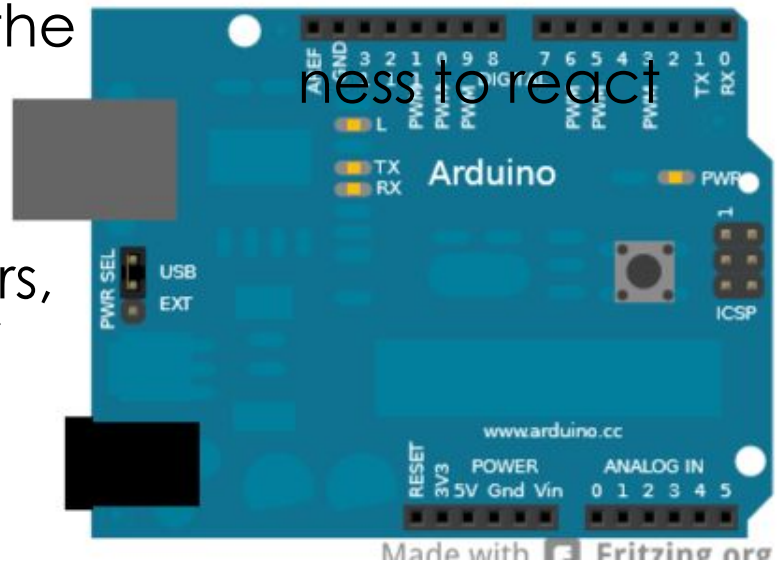
WHAT IS ARDUINO?

- Arduino is an open-source electronics prototyping platform based on flexible, easy-to use hardware and software.
- Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.
- The microcontroller on the board is programmed using the Arduino programming language(simplified c++) and the Arduino Development Environment. They can communicate with software running on a computer.
- The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

WHAT CAN YOU DO WITH ARDUINO

- An Arduino can basically do anything by interfacing sensors with a computer.
- This would allow you to take any sensor and have any action applied with the readings.
- For example (in one of our projects) we will read the level of light in a room and adjust an LED's bright ness to react based on that input.
- Think of the possibility of wiring your house with all sorts of different sensors (photocells, oxygen sensors, thermometers) and having it adjust your blinds, air conditioner and furnace and make your house a more comfortable place.



INSIDE THE ARDUINO

- Although there are many different types of Arduino boards available, this manual focuses on the Arduino Uno. This is the most popular Arduino board around
- Processor: 16Mhz ATmega328
- Flash memory : 32Kb
- Ram: 2Kb
- Operating Voltage: 5V
- Input Voltage: 7-12 V
- Number of Analog Inputs: 6
- Number of Digital I/O: 14(6 Of them PWM)

APPLICATIONS

- Home Automation Systems
- Remote controlled cars
- Robots
- Video Games
- Detecting things
- Remote control systems
- Controlling systems like chemical Reaction

PIN CONFIGURATION

- It consists power(USB/Barrel Jack).
- It is working with 5V of supply . USB connection is also useful for uploading code to the Arduino (dump).
- **PINS:** 5V,Gnd,3.3V,Analog,Digital,PWM,AREF.
- **Analog Pins:** A0-A5 Pins used for read values from analog sensors.
- **Digital Pins:** 0-13 pins used to perfrom digitalread,digitalwrite,analogoutputs.
- In digital pins to drive analog output using PWM Pins 11,10,9,6,5,3
- **Reset Button:** To restart the the code in the Arduino we use reset button
- **TX &RX LEDs** : indicates that thansmitting and receiving the data.

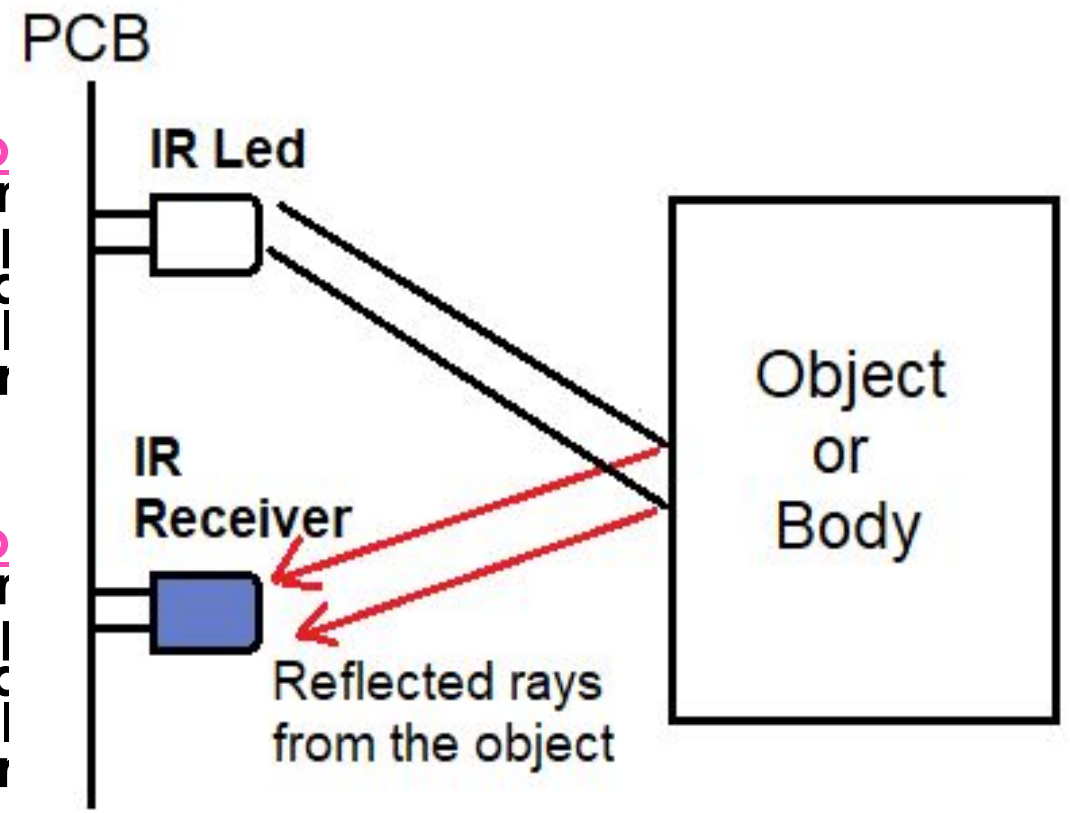
SENSORS & COMPONENTS

- IR Sensor
- Bluetooth module
- LDR
- Moisture Sensor
- LM293D
- Push Button
- 7segment display
- Dc motor
- LED (RGB)
- Thinker CAD
- APP

IR SENSOR

- An infrared senso
  aspects of the sur
  The emitter is simp
  IR photodiode tho
  the IR LED. When l
  voltages will char
  IR light received.

- An infrared senso
  aspects of the sur
  The emitter is simp
  IR photodiode tho
  the IR LED. When l
  voltages will char
  IR light received.



er to sense some

e detector is simply an
ength as that emitted by
nces and the output

er to sense some

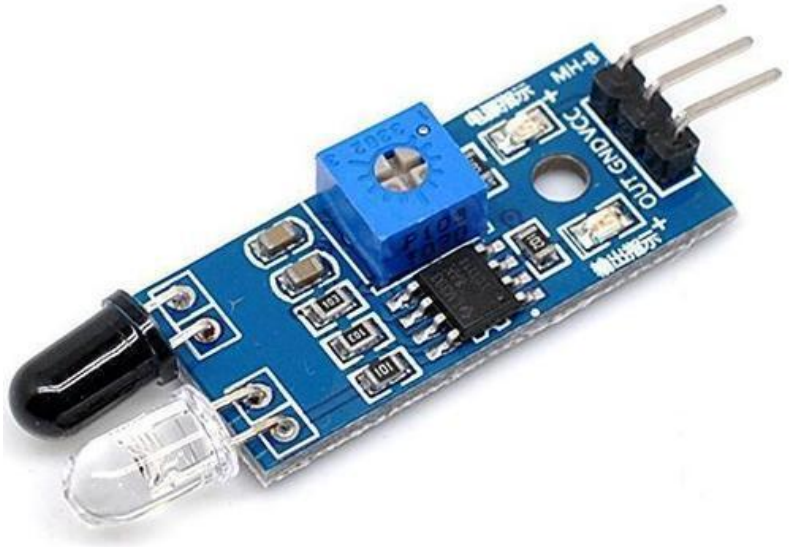
e detector is simply an
ength as that emitted by
nces and the output

PIN CONFIGURATION & CONNECTIONS

- In IR Sensor there are 3 pins.
- 1.vcc :supply to the sensor either 5v or 3.3v.
- 2.gnd: this pin connected to ground:
- 3.out: this pin used to read sensor reading values.

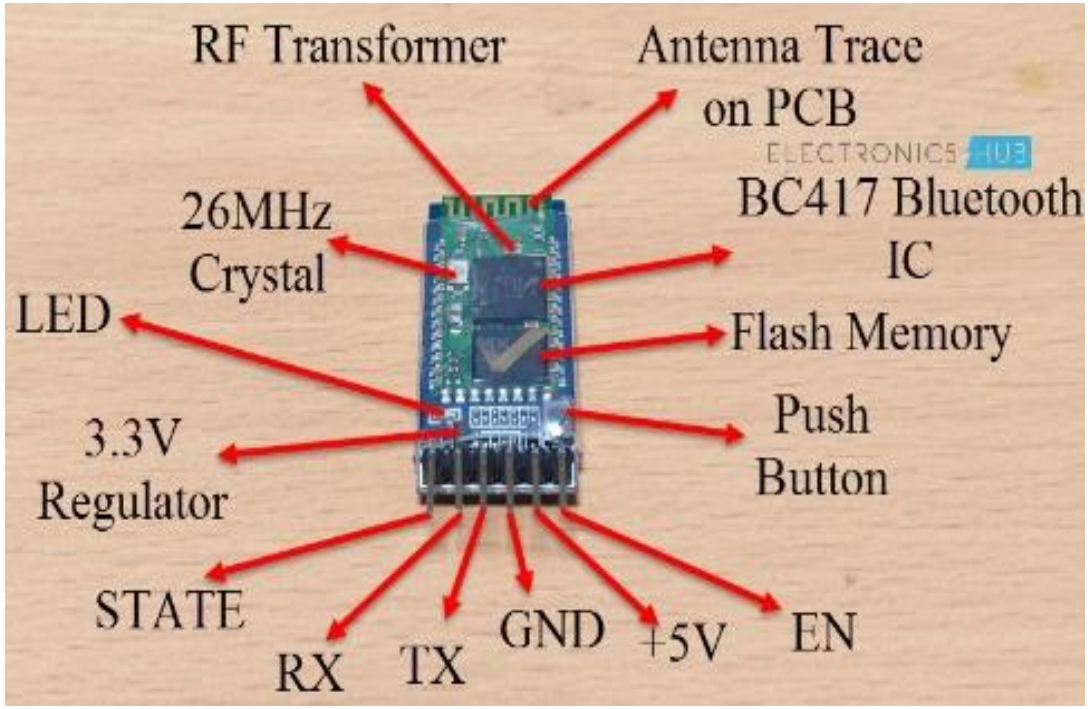
Application:

- **1.Radiation thermometers**
- **2.flame monitors**
- **3.moisture analyzers**
- **4.gas analyzers**



BLUETOOTH MODULE

- **Bluetooth module (HC-05) is designed for transparent wireless serial connection .its communication is via serial communication makes an easy way to interface with controller.**
- **It consists an antenna which is responsible sending and receiving the data.**
- **It is active low device.**
- It consists 6 pins:
- 1.enabe/key
- 2.vcc(supply either 5v or 3.3v)
- 3.ground
- 4.Tx
- 5.Rx
- 6.State



CONNECTIONS

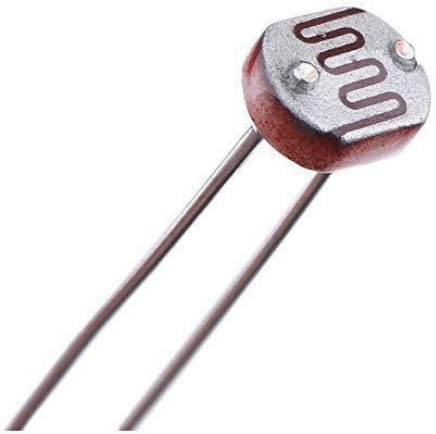
PIN	DESCRIPTION	FUNCTION
VCC	+5v	Connected supply 5v
GND	ground	Connected to ground
TXD	Transmitter pin ,bluetooth serial sending pin	Connect the pin to the ardino reciever pin(RX)
RXD	Receiver pin,bluetooth serial receiving pin	Connect the pin to the arduino transmitter pin.
KEY	Mode switch input	AT command mode
State	indicator	It shows HC-05 ON/OFF

LDR

- Light dependent Resistor are often used to detect the presence or the level of light
- LDR provide large change in resistance for changes in light levels.
- When light falls on LDR ,the resistance of LDR is decreased
- When no light falls on LDR , the resistance of LDR is increased
- It is also depend on wavelength of light
- **Connections:**
- It consists two terminals .first terminal is connected to vcc(supply) and second terminal pin is connected to ground through the resistors.
- Analog value of sensor is measured at second terminal using analog pins.

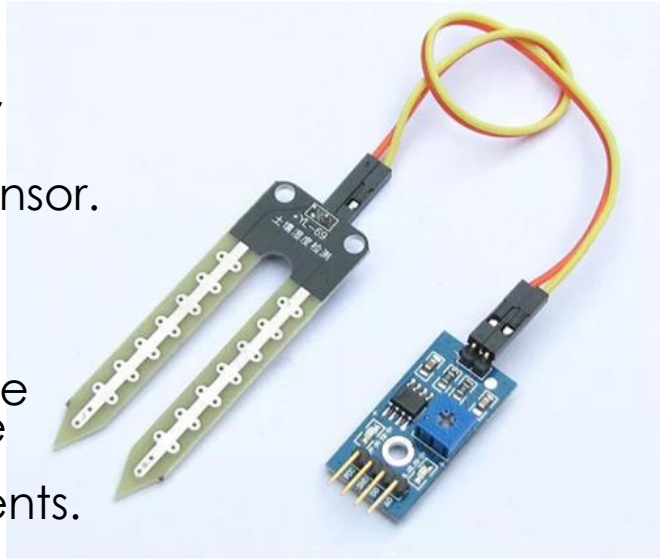
APPLICATION

- **1.** LDRs are used in Light Sensors
- **2.** LDR is also used in some cameras to detect the presence of the light.
- **3.** LDRs are used Light Intensity measurement meters.
- **4.** In the manufacturing industry, LDR is used as a sensor for the counting of the packets moving on a conveyor belt.
- **5.** LDRs are also used in Light Activated Control Circuits.
- **6.** LDRs are used in Street Lights which are automatically turn ON in the night time.



MOISTURE SENSOR

- The **Moisture sensor** is used to measure the water content(moisture) of soil. When the soil is having water shortage , the module output is at high level, else the output is at low level.
- **Working**:
- The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium.
- In soil dielectric permittivity is a function of the water content.
- The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil.
- The sensor averages the water content over the entire length of the sensor.
- There is a 2 cm zone of influence with respect to the flat surface
- of the sensor, but it has little or no sensitivity at the extreme edges.
- The Soil Moisture Sensor is used to measure the loss of moisture over time due to evaporation and plant uptake , evaluate optimum soil moisture contents for various species of plants , monitor soil moisture content to control irrigation in greenhouses and enhance bottle biology experiments.



PIN COFIGURATION AND CONNECTION

- Soil moisture sensor consists 4 pins.
- 1.vcc: supply to the sensor either 5v or 3.3v.
- 2.gnd: this terminal connected to ground.
- 3.A0: this pin used to take analog value of the sensor.
- 4.D0: this pin used to take digital value of the sensor.
- Note:
- We convert the analog value of sensor into percentage by using formula.
- 100-(sensorvalue /1023.00)*100.
- It gives low percentage when water content in soil is less.
- It gives high percentage when water content in soil is high.

LM293D MODULE

- It is
  spe
  dev
- This
  gro
  Mo
- It n
  has
  to c

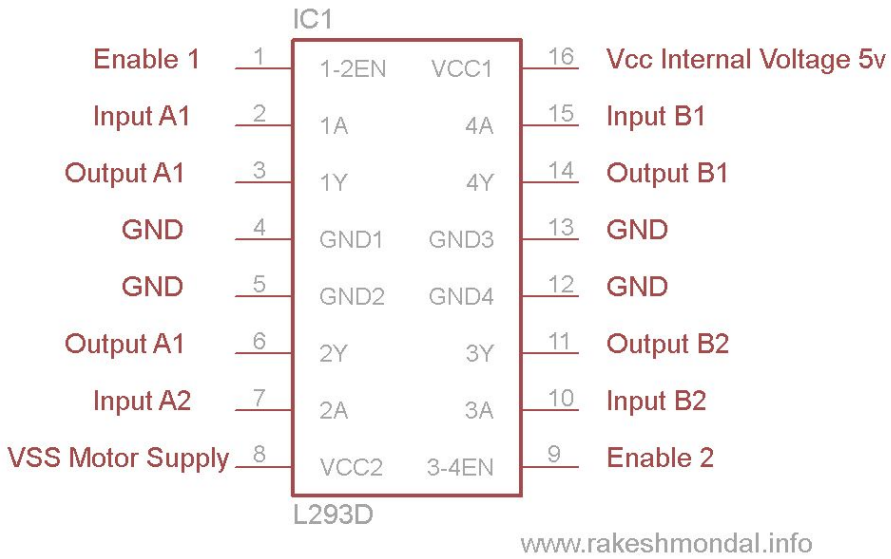
Motor working:

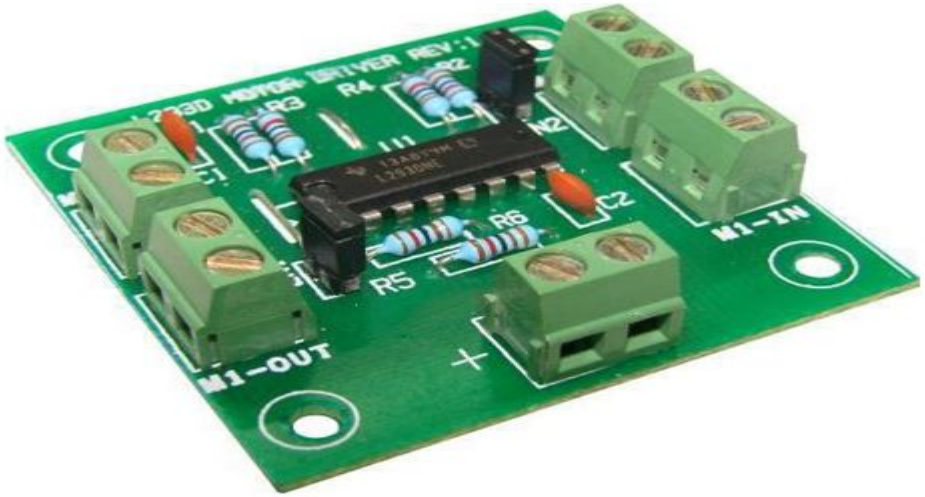
Pin8	Pin7	Output1	Output2	Rotating
High	Low	High	Low	clockwise
Low	High	Low	High	Anti-clockwise
Low	Low	Low	Low	stop
High	High	High	High	Stop(high impedence)

Output 4 Output 2
Output 3 Output 1

APPLICATION

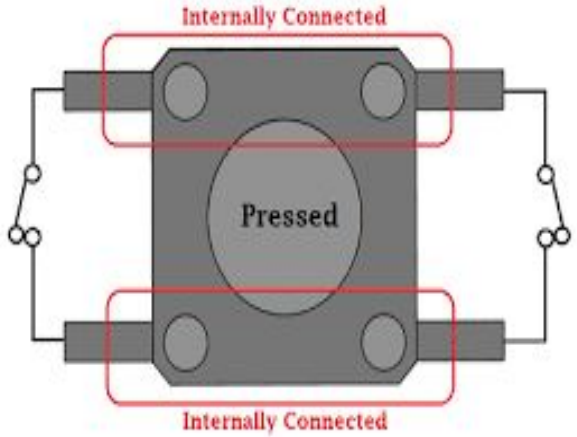
- To control DC motors in Robots.
- To generate more supply to DC motors

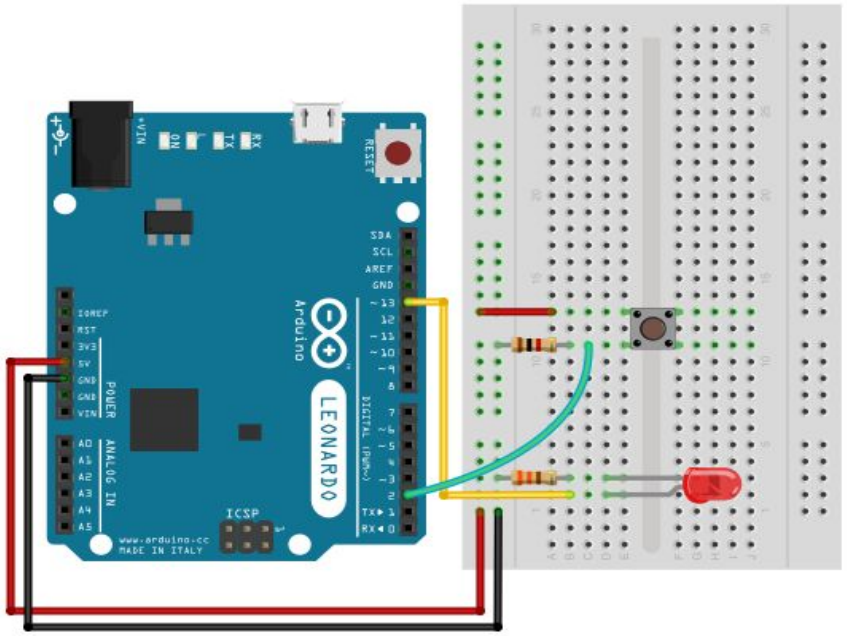


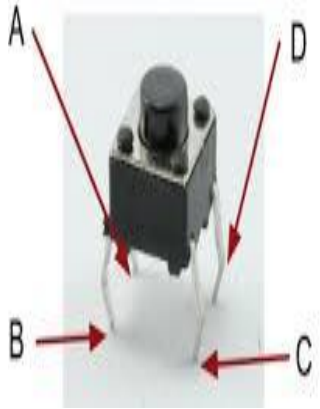


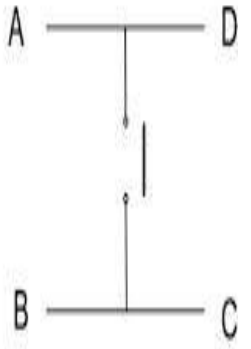
PUSH BUTTON

- Push-Buttons are normally-open **tactile switches**. Push buttons allow us to power the circuit or make any particular connection only when we press the button.

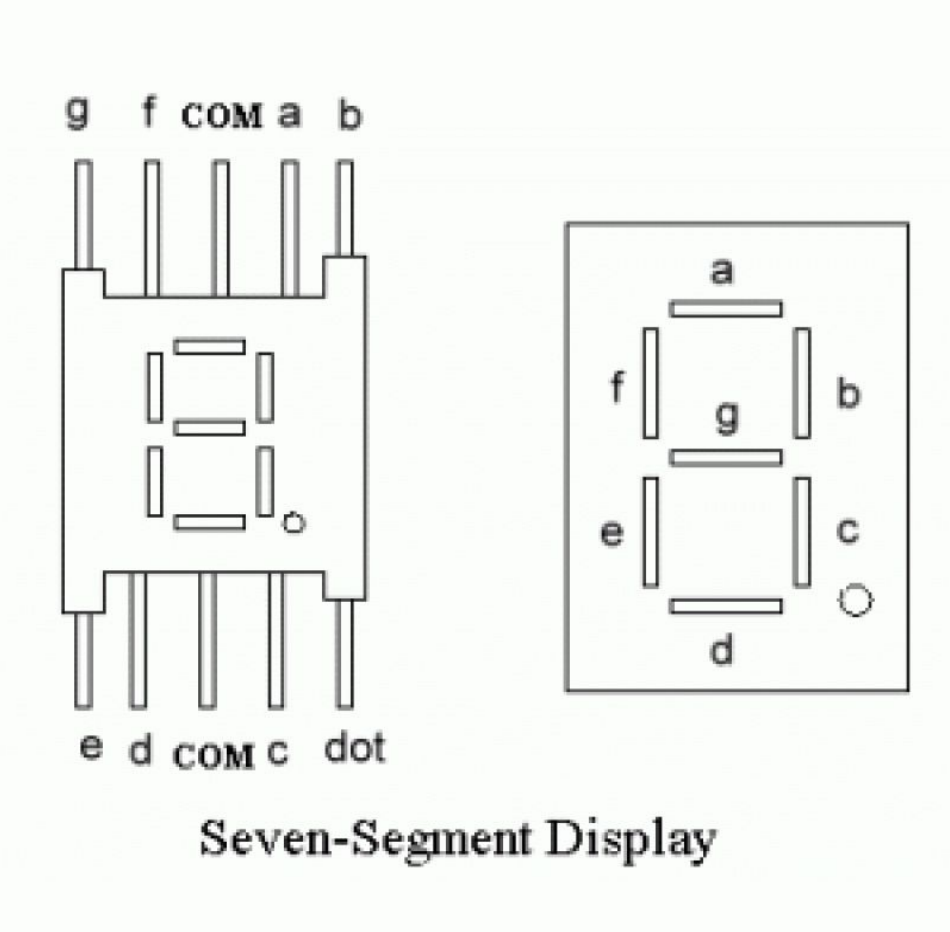








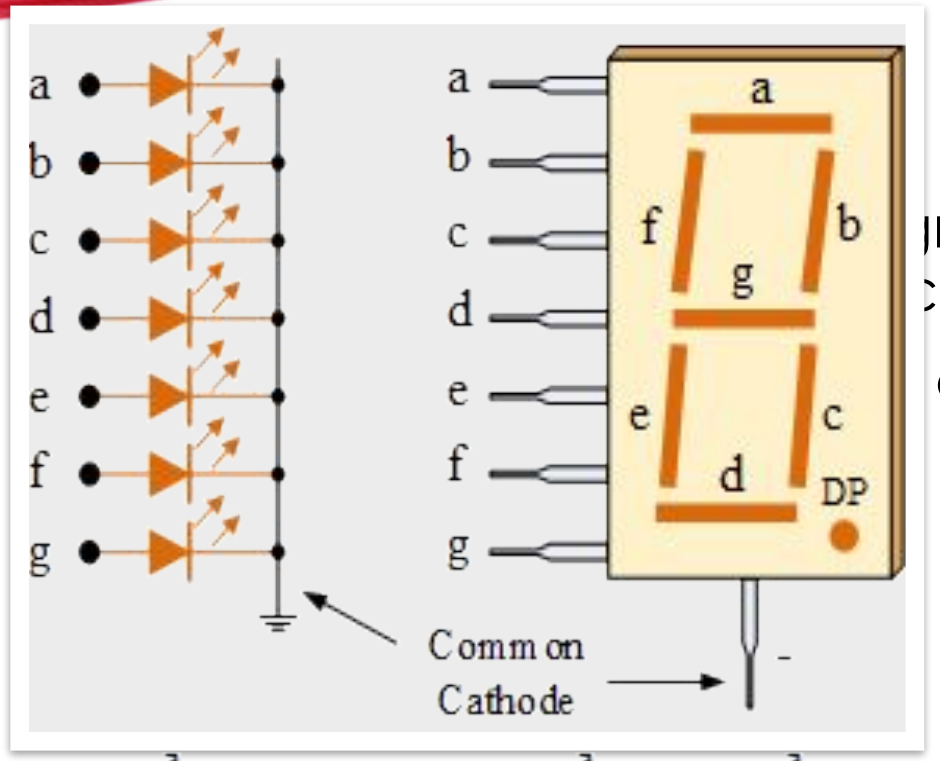
7 SEGMENT DISPLAY

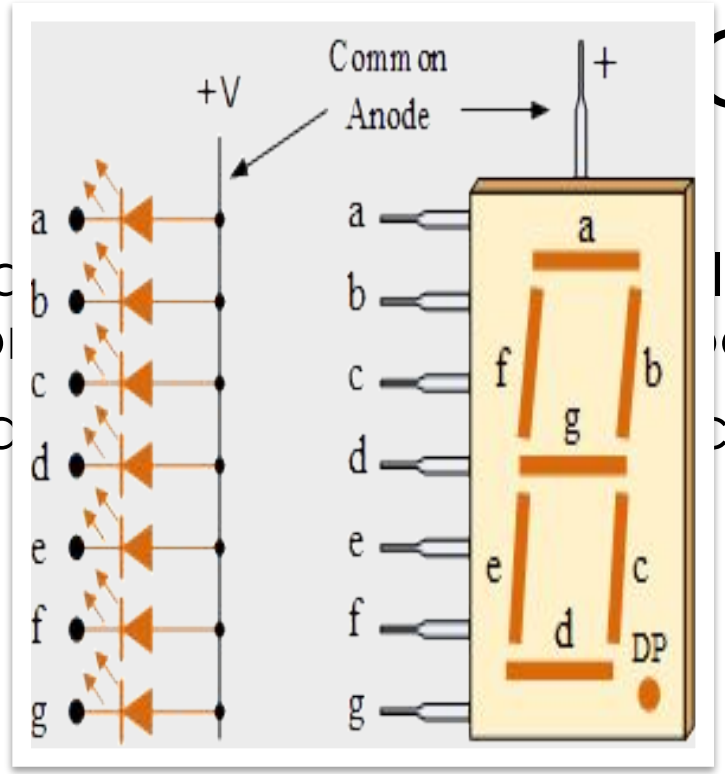


Seven-Segment Display

- seven segment display", consists of seven LEDs and one dot led (hence its name) arranged in a rectangular fashion as shown.
- Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package.
- These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.
- there are therefore two types of LED 7-segment display called: **Common Cathode** (CC) and **Common Anode** (CA).

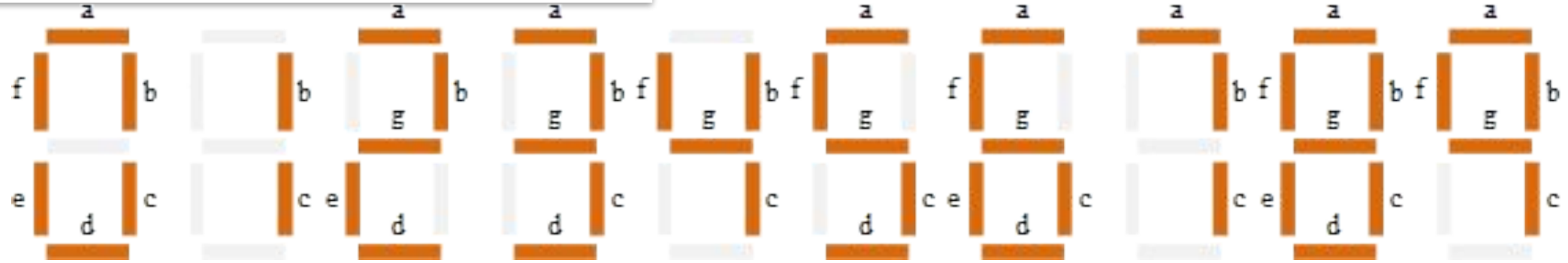
CATIONS





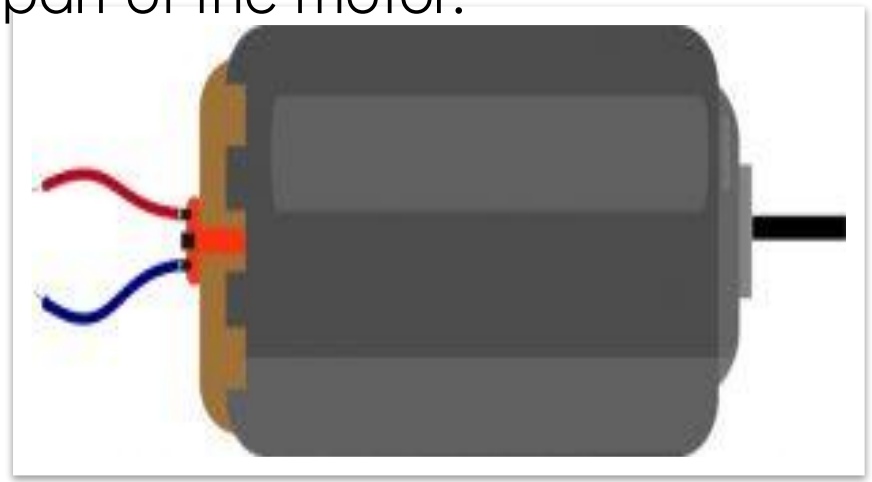
ɟments c
cks, odo
applic

lators,
ck radios, etc.
cause of low



DC MOTOR

- A car consists of two DC motors . DC Motor is a rotary electrical machine that converts direct electrical energy into mechanical energy.
- DC motor working is based on electromagnetism(The forces produced by magnetic fields).
- All type of dc motors have some internal mechanism that is electromechanical to periodically change the direction of current flow in part of the motor.
- DC motor speed is controlled by changing voltage *or* strength of current field windings.
- Small DC motors used in tools,toys and other applicances.
- Lager dc motors are currently used in propulsion electric vehicles and elevators



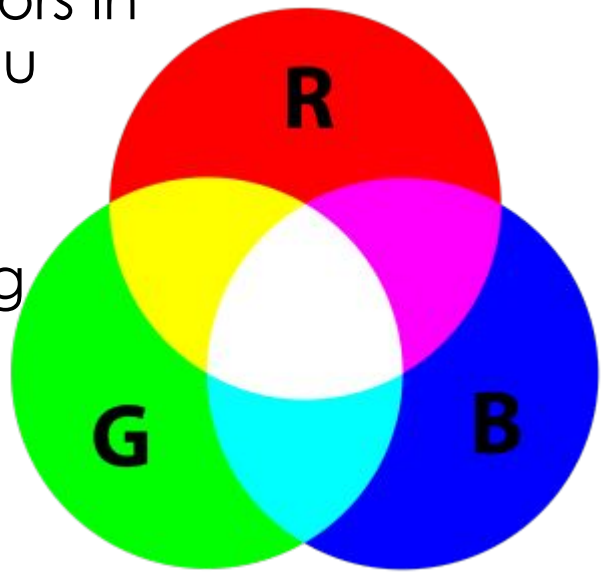
CONTD..

- DC motor consists of two terminals ,one is positive terminal and other is negative terminal .The power supply to DC motor given based on rpm.
- here the dc motor is 9000 rpm working by giving supply voltage 4.5 volts to 9 volts. The rpm of the motor is controlled by using Pulse Width Modulation.
- PWM is processed in driver ic by using Enable pins.
- By this you can control the car motors while turning right or left by doing first motor slow down and second motor moving with some speed greater than the first motor.

INPUT 1	INPUT 2	Result
0	0	Stop
0	1	Anti-clockwise rotation
1	0	Clockwise rotation
1	1	Stop

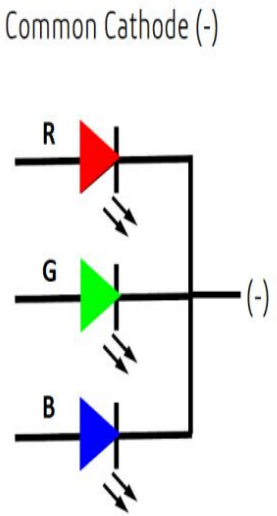
RGB LED

- With an RGB LED you can, of course, produce red, green, and blue light, and by configuring the intensity of each LED, you can produce other colors as well.
- To produce other colors, you can combine the three colors in different intensities. To adjust the intensity of each LED you can use a PWM signal.
- To have an idea on how to combine the colors, take a look at the following chart. This is the simplest color mixing chart, but gives you an idea how it works and how to produce different colors.



CONTD…

- There are two kinds of RGB LEDs: *common anode* LED and *common cathode* LED. The figure below illustrates a common anode and a common cathode LED.



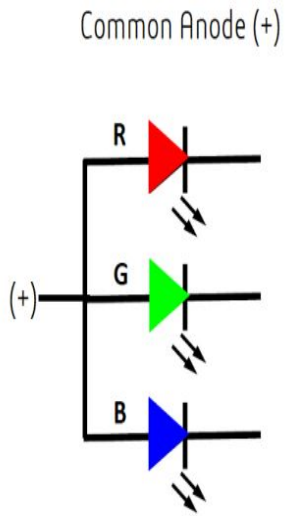
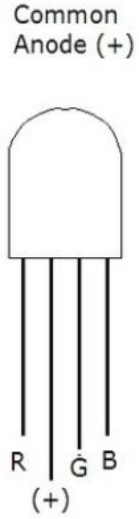
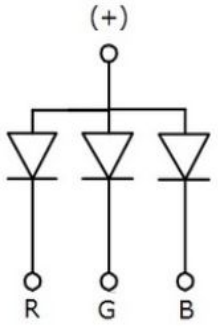


Fig. 5-4 Common Cathode RGB Diagram



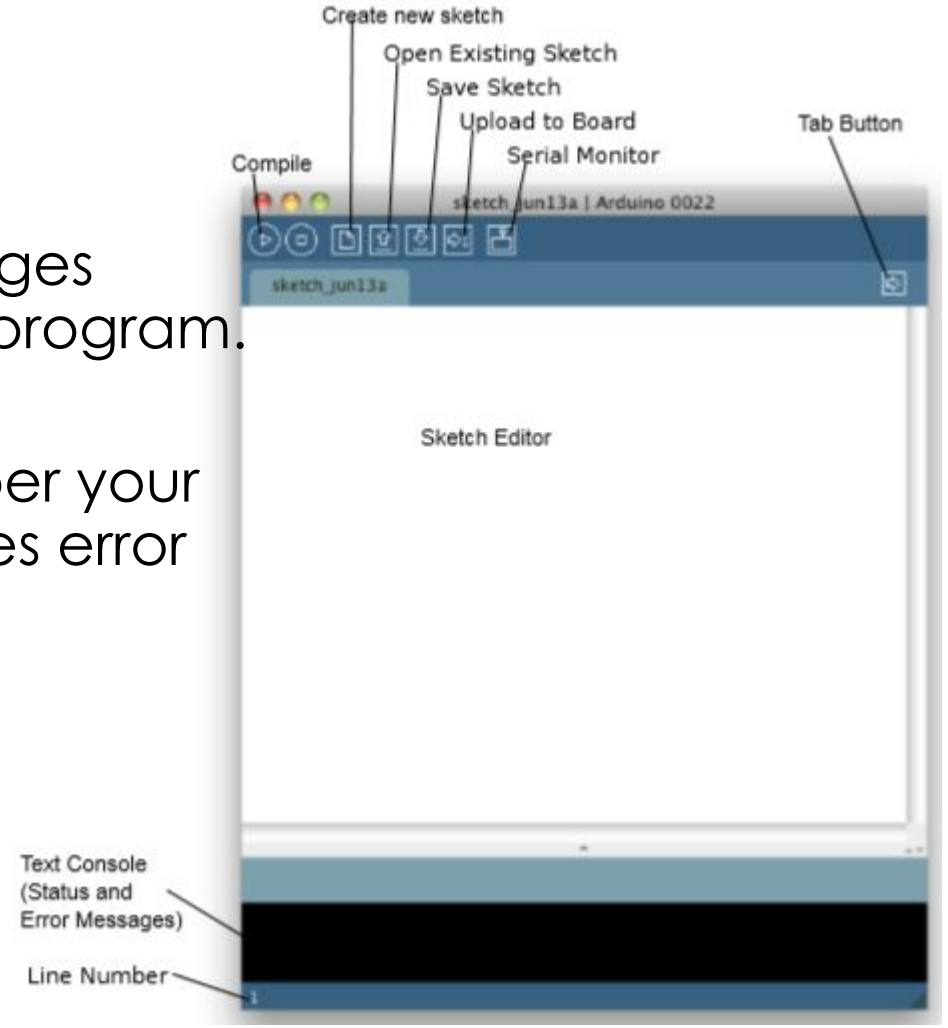


ARDUINO IDE

- You use the Arduino IDE on your computer (picture following) to create, open, and change sketches
- Sketches define what the board will do. You can either use the buttons along the top of the IDE or the menu items.
- **Compile** - Before your program "code" can be sent to the board, it needs tobeconvertedintoinstructionsthattheboardunderstands. Thisprocess is called compiling
- **Stop** - This stops the compilation process. (I have never used this button and you probably won't have a need to either.)
- **Create new Sketch** - This opens a new window to create a new sketch.
- **Open Existing Sketch** - This loads a sketch from a file on your computer.
- **Save Sketch** - This saves the changes to the sketch you are working on.
- **Upload to Board** - This compiles and then transmits over the USB cable to your board.

CONTD.....

- Text Console - This shows you what the IDE is currently doingand is also where error messages display if you make a mistake in typing your program. (often called a syntax error)
- Line Number - This shows you what line number your cursor is on. It is useful since the compiler gives error messages with a line number



PROGRAMING STRUCTURE

- Arduino programs can be divided in three main parts: **Structure, Values** (variables and constants), and **Functions**.

Software structure consist of two main functions −

  - Setup( ) function
  - Loop( ) function

- The **setup()** function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.
- After creating a **setup()** function, which initializes and sets the initial values, the **loop()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

Operator name	Operator simple	Description	Example
increment	++	Increment operator, increases integer value by one	A++ will give 11
decrement	--	Decrement operator, decreases integer value by one	A-- will give 9
compound addition	+=	Add AND assignment operator. It adds right operand to the left operand and assign the result to left operand	B += A is equivalent to B = B+ A
compound subtraction	-=	Subtract AND assignment operator. It subtracts right operand from the left operand and assign the result to left operand	B -= A is equivalent to B = B - A
compound multiplication	*=	Multiply AND assignment operator. It multiplies right operand with the left operand and assign the result to left operand	B*= A is equivalent to B = B* A
compound division	/=	Divide AND assignment operator. It divides left operand with the right operand and assign the result to left operand	B /= A is equivalent to B = B / A
compound modulo	%=	Modulus AND assignment operator. It takes modulus using two operands and assign the result to left operand	B %= A is equivalent to B = B % A
compound bitwise or	|=	bitwise inclusive OR and assignment operator	A |= 2 is same as A = A | 2
compound bitwise and	&=	Bitwise AND assignment operator	A &= 2 is same as A = A & 2
		right by the number of bits specified by the right operand.	which is 0000 1111

BASIC PROGRAMES

- digitalWrite
- digitalRead
- digitalRead & digitalWrite
- analogRead
- analogWrite
- analogRead & analogWrite

DIGITALWRITE

- By using dig t devices like LEDs,Motors and other c
- Basic Progr

```
int led_pin=8;
void setup()
{
  pinMode(led_pin,OUTPUT);
}

void loop()
{
  digitalWrite(led_pin,HIGH);
  delay(1000);
  digitalWrite(led_pin,LOW);
  delay(1000);
}
```

DIGITAL READ

- To g
- we c
- Basi

```
2§
int ir_sensor_pin=3;
int ir_value;
void setup()
{
  pinMode(ir_sensor_pin,INPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_value=digitalRead(ir_sensor_pin);
  Serial.print(ir_value);
  Serial.print("\n");
}
```

D & DIGITAL WRITE

- By using d et value from sensor and operate a
- For examp son
- Basic prog

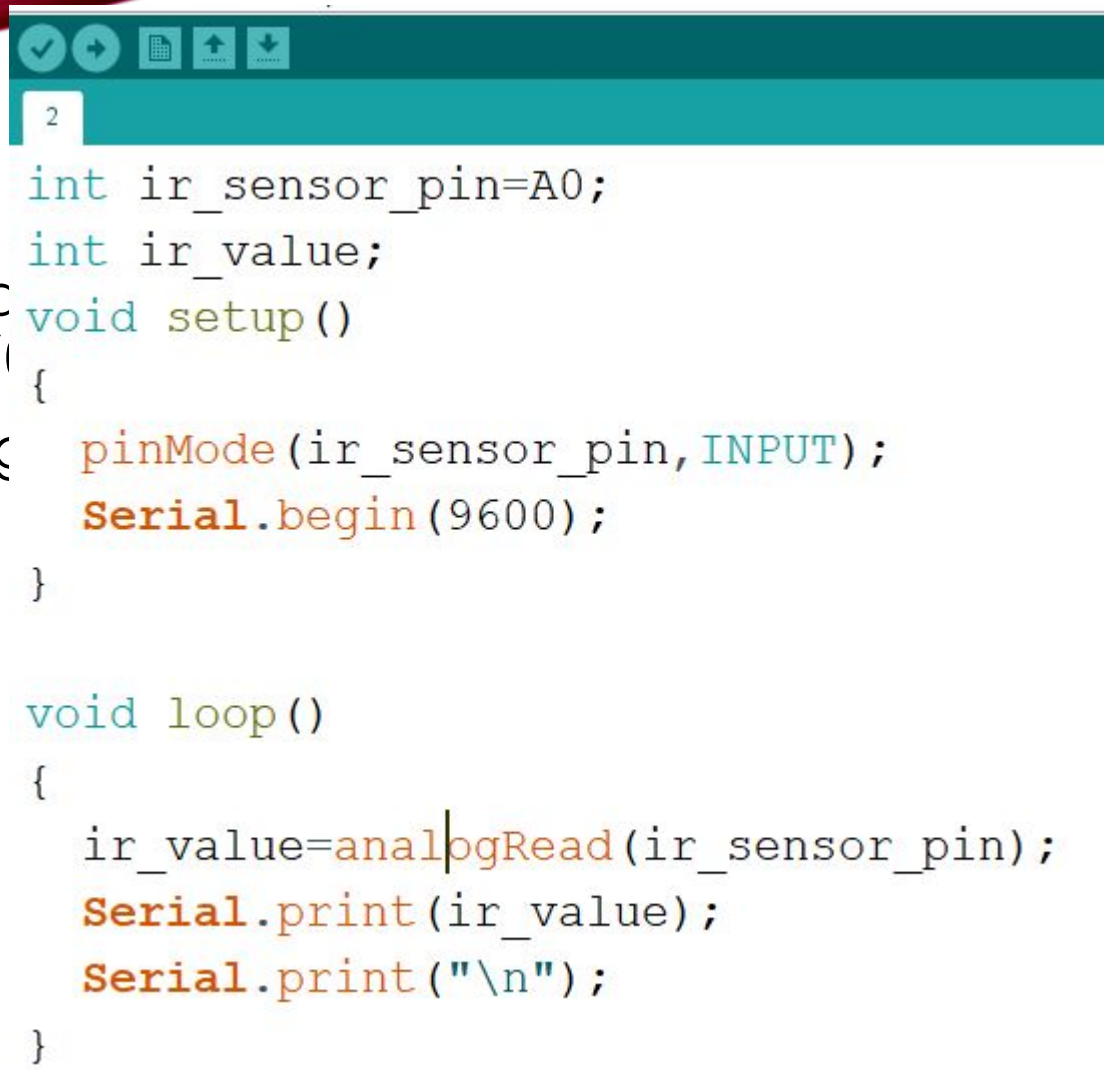
digital_ir_with_led

```
int ir_sensor=3;
int in_val=0;
int led_pin=11;
void setup()
{
  pinMode(ir_sensor,INPUT);
  pinMode(led_pin,OUTPUT);
  Serial.begin(9600);
}

void loop()
{
 in_val=digitalRead(ir_sensor);
 Serial.println(in_val);
 delay(9600);
 if(in_val==HIGH)
 {
  digitalWrite(led_pin,HIGH);
 }
 else
 {
  digitalWrite(led_pin,LOW);
 }

}
```

ANALOGREAD

- By using c
  range of(
- Basic Prog

for analog sensors in the

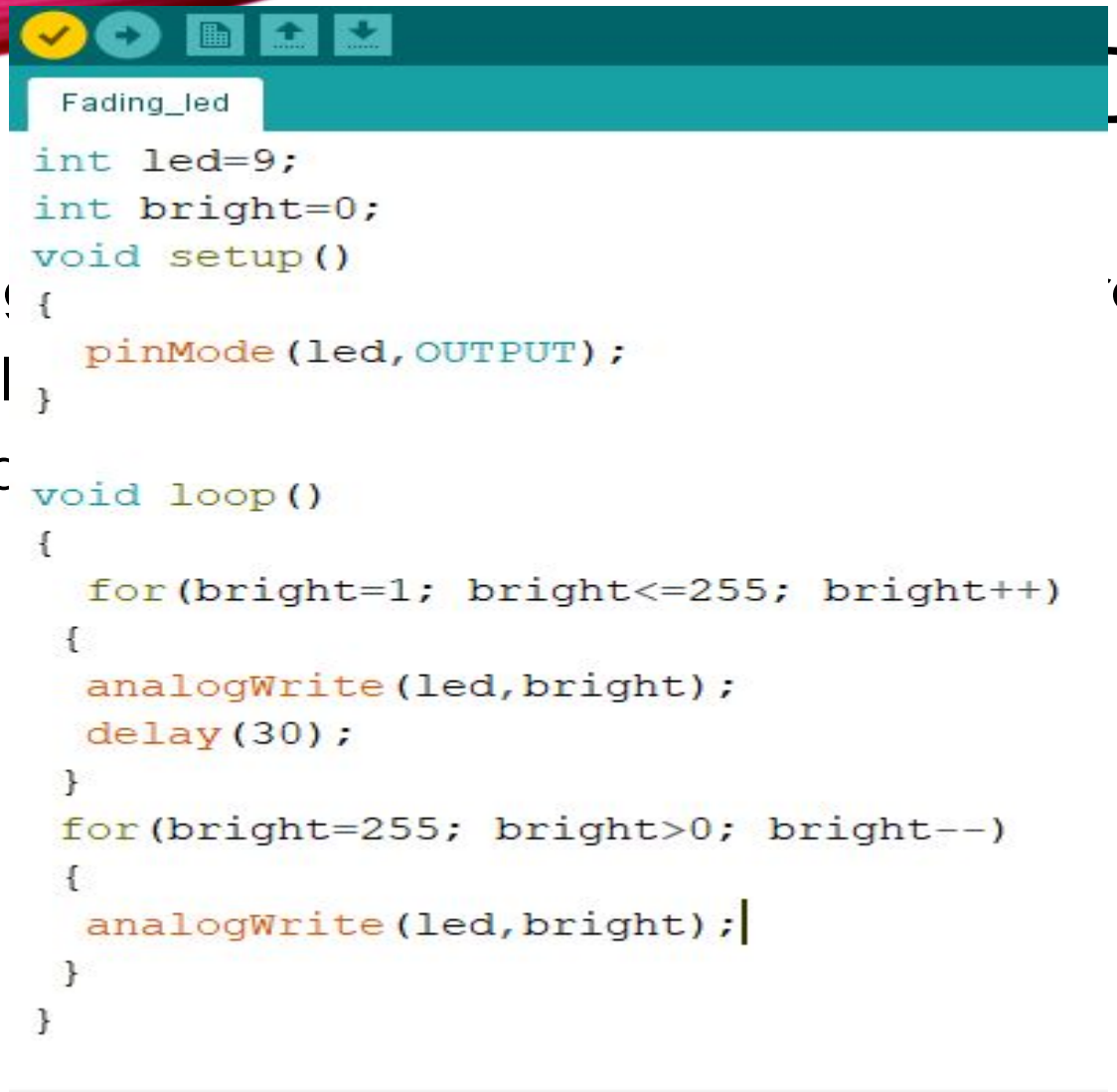


```
2

int ir_sensor_pin=A0;
int ir_value;
void setup()
{
  pinMode(ir_sensor_pin,INPUT);
  Serial.begin(9600);
}

void loop()
{
  ir_value=analogRead(ir_sensor_pin);
  Serial.print(ir_value);
  Serial.print("\n");
}
```

D & ANALOGWRITE

- In di e used to analogwrite.
- anal

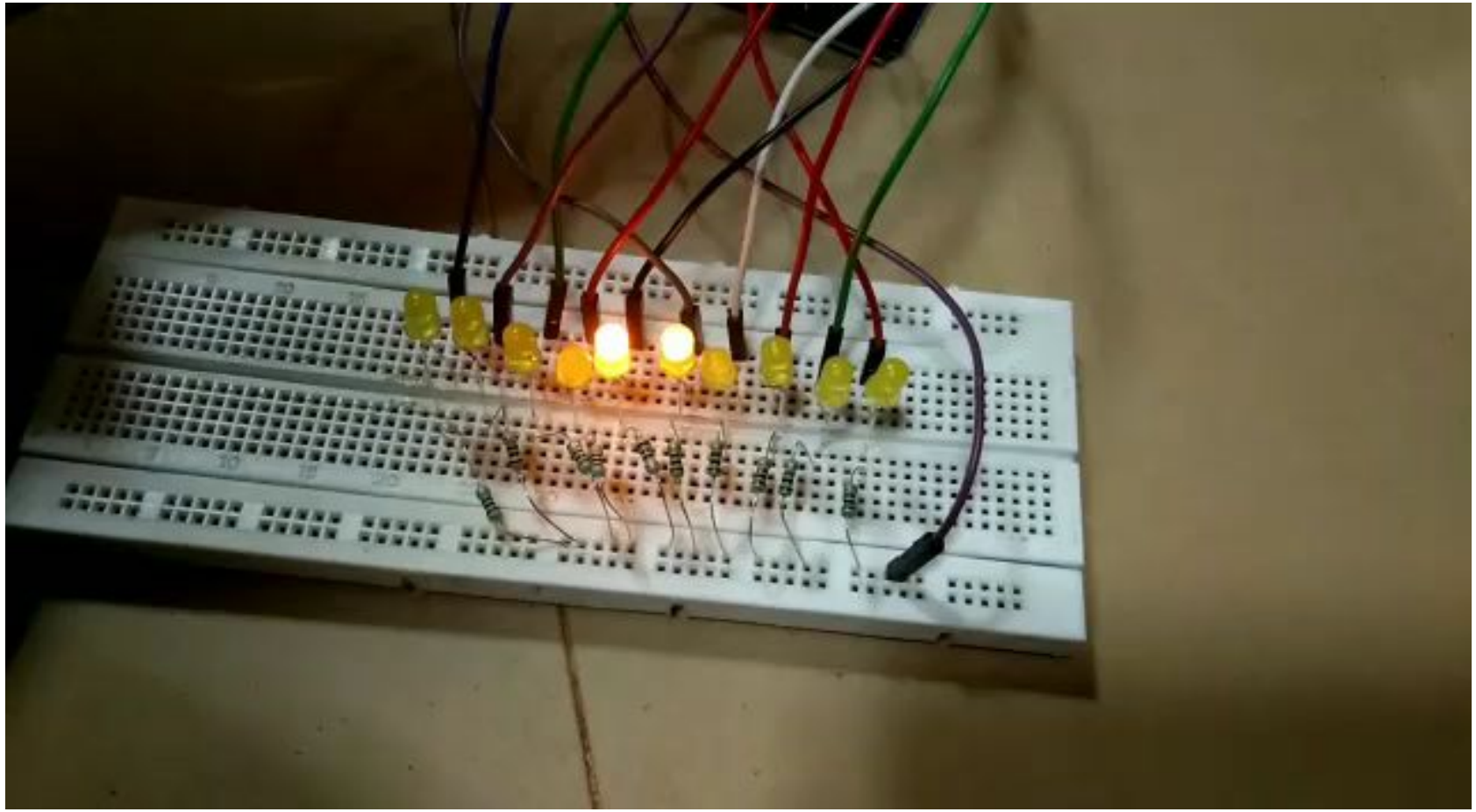
Progrc



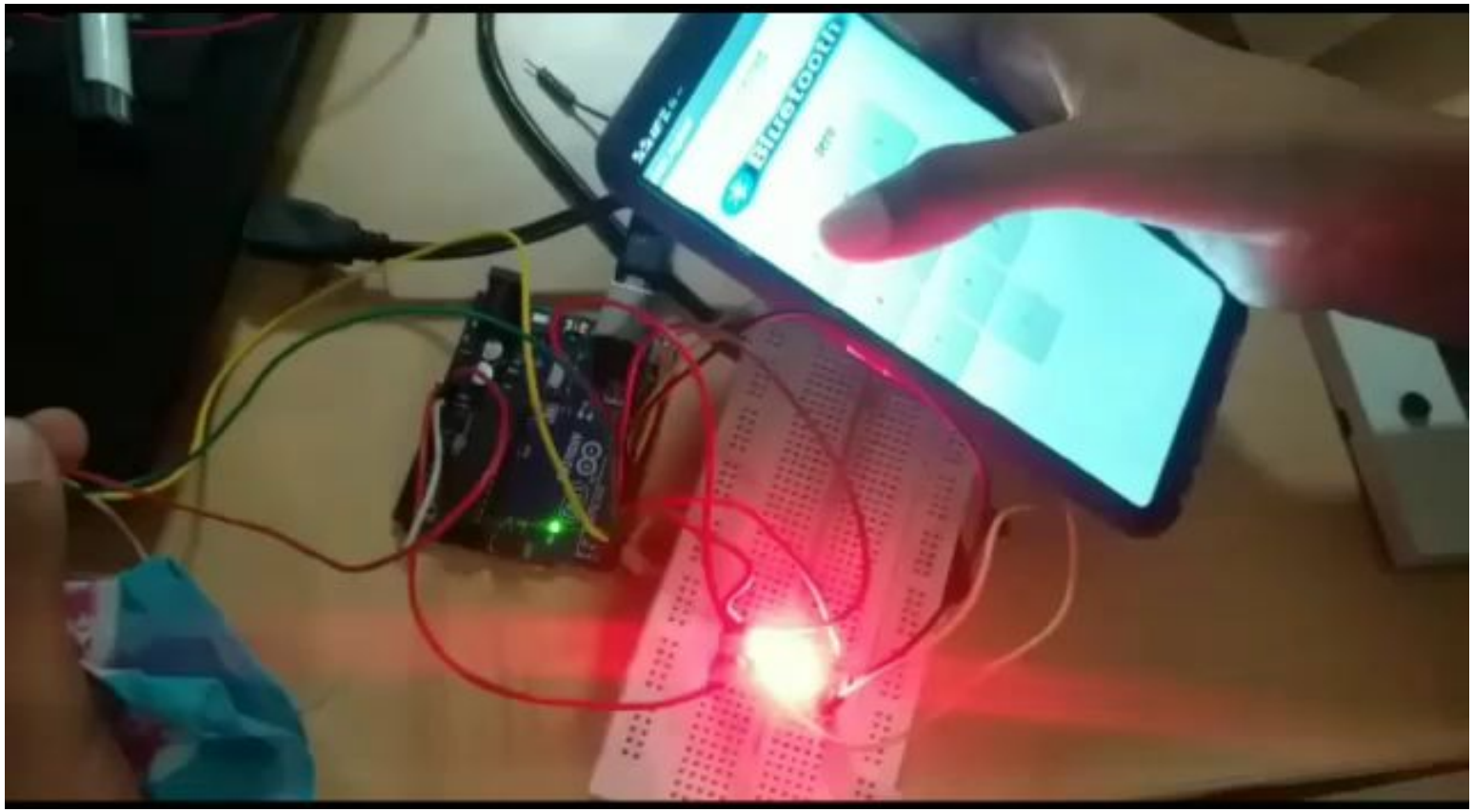
Fading_led

```
int led=9;
int bright=0;
void setup()
{
  pinMode(led,OUTPUT);
}

void loop()
{
  for(bright=1; bright<=255; bright++)
 {
  analogWrite(led,bright);
  delay(30);
 }
 for(bright=255; bright>0; bright--)
 {
  analogWrite(led,bright);
 }
}
```

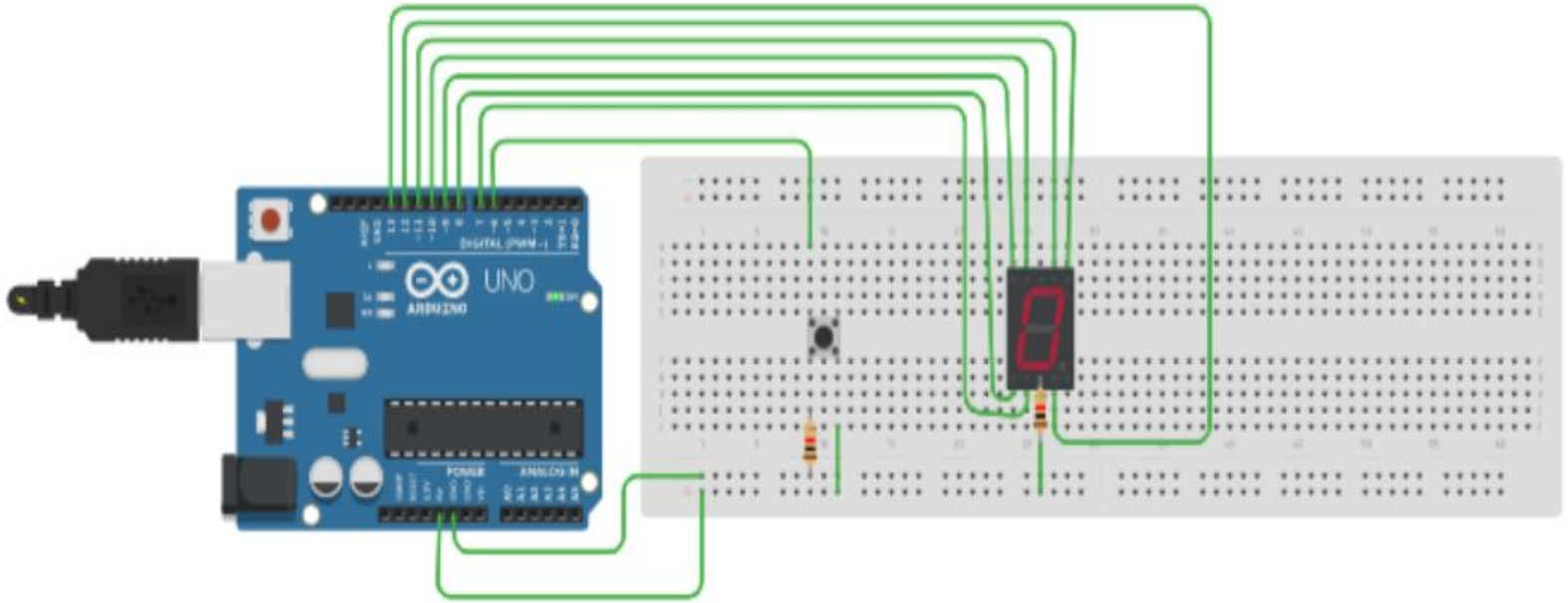
LED PATTERNS



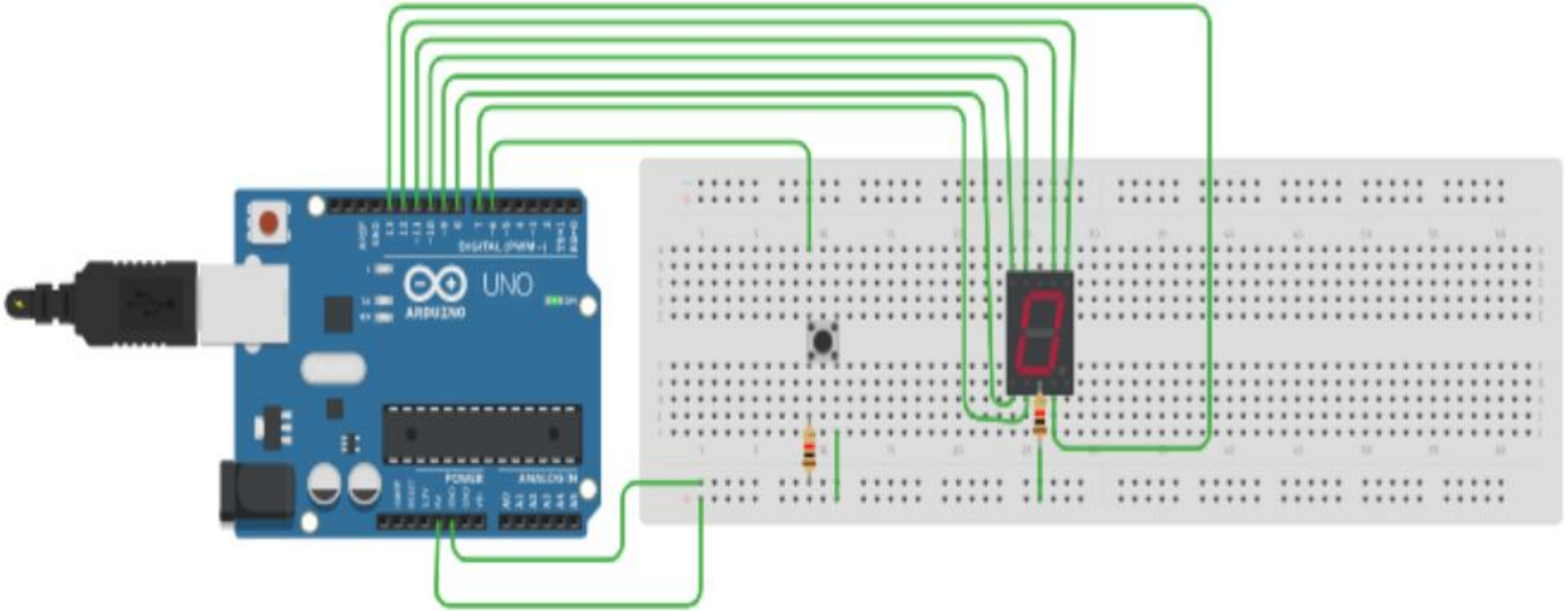
SEVEN SEGMENT DISPLAY



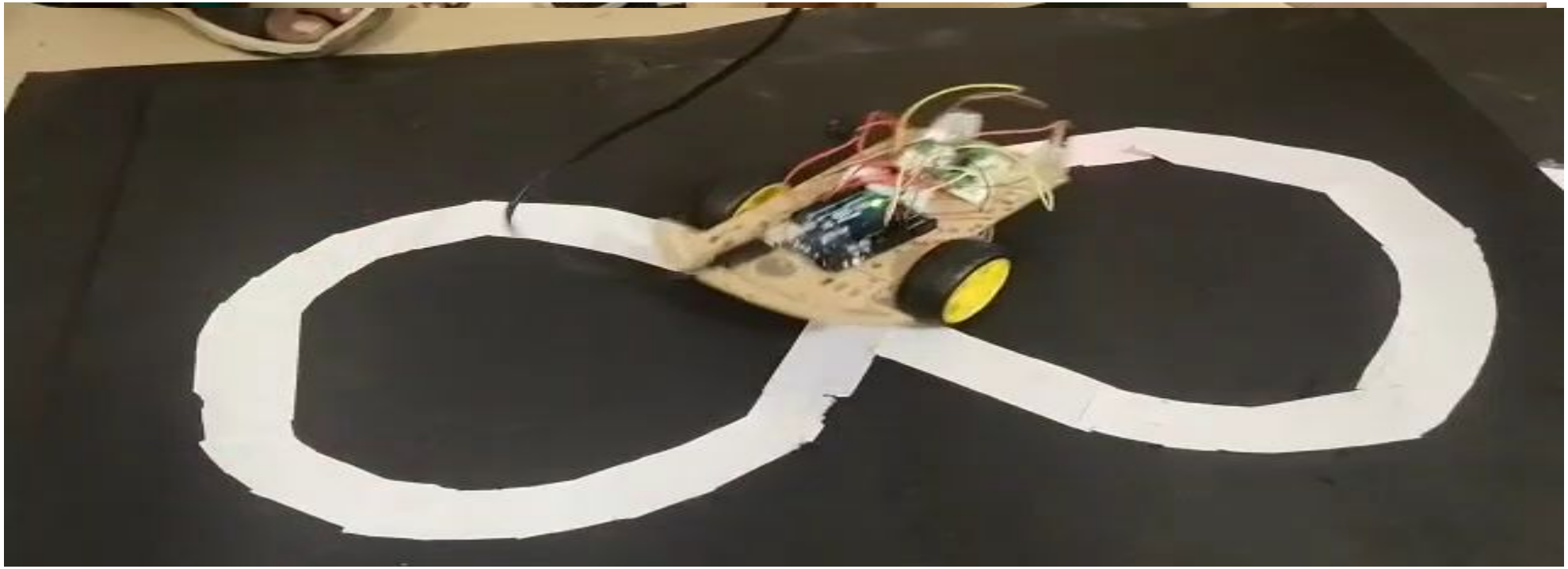
PUSHBUTTON VOTING



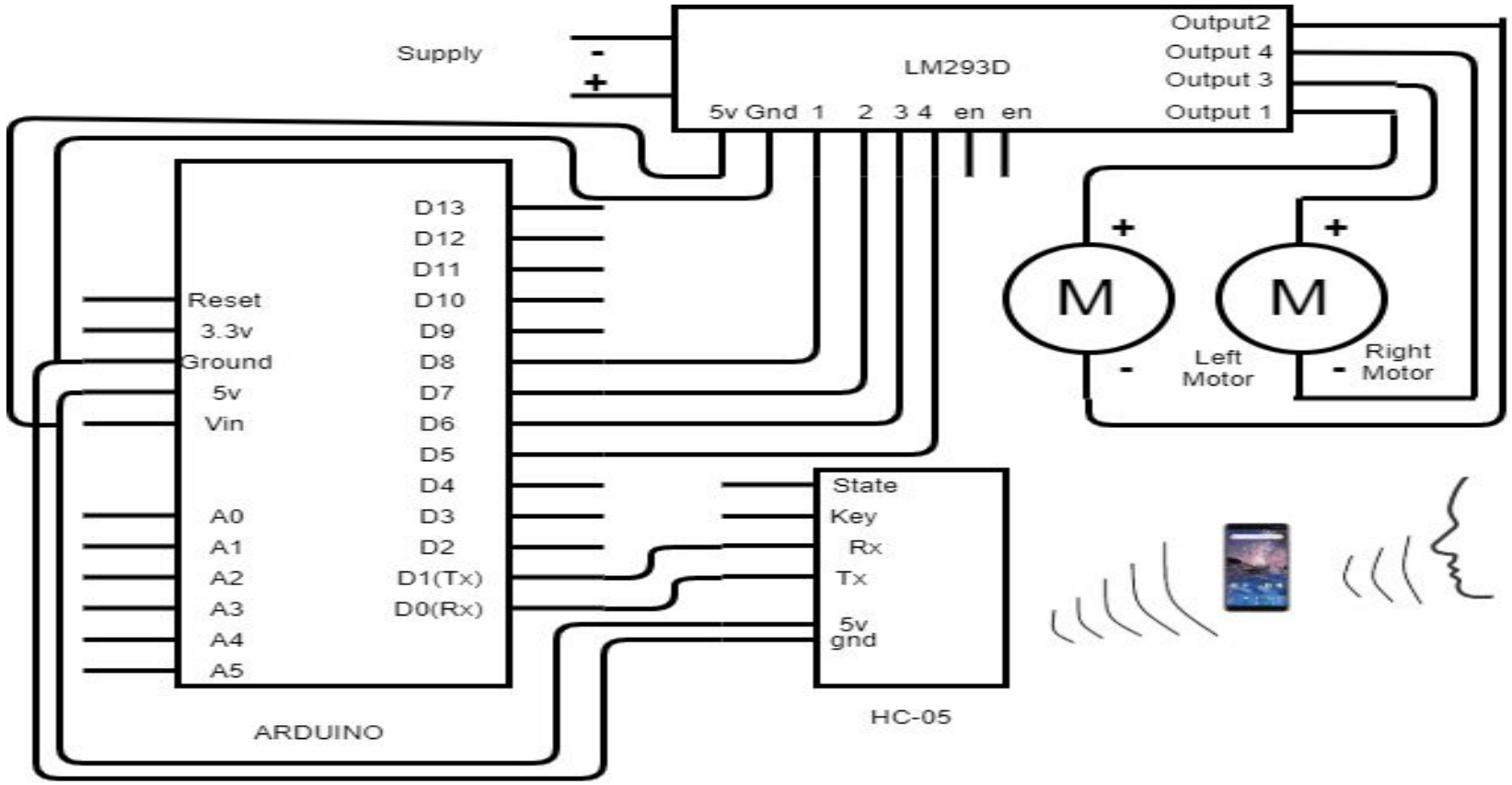
ELECTRONIC DIE



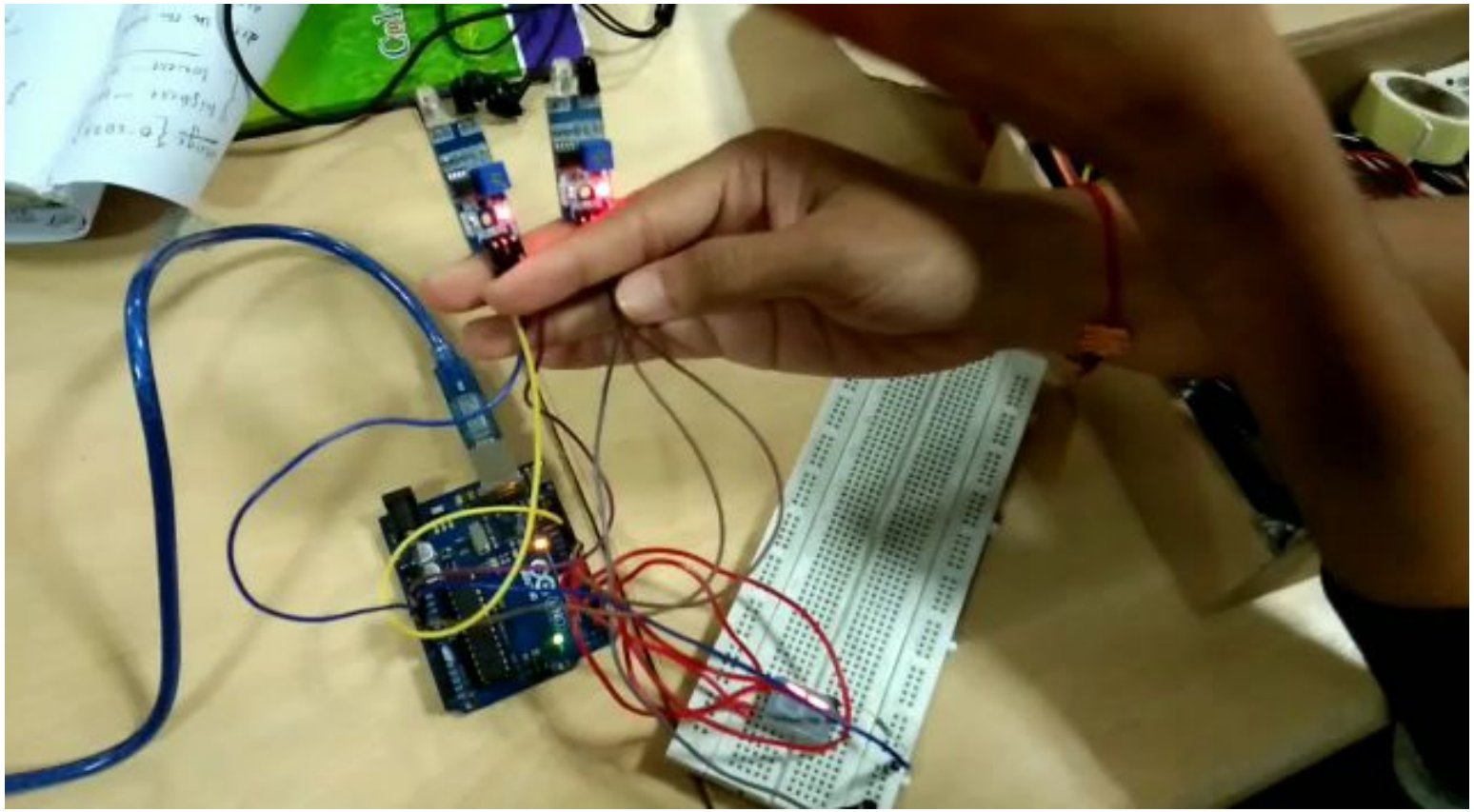
LINE FOLLOWER



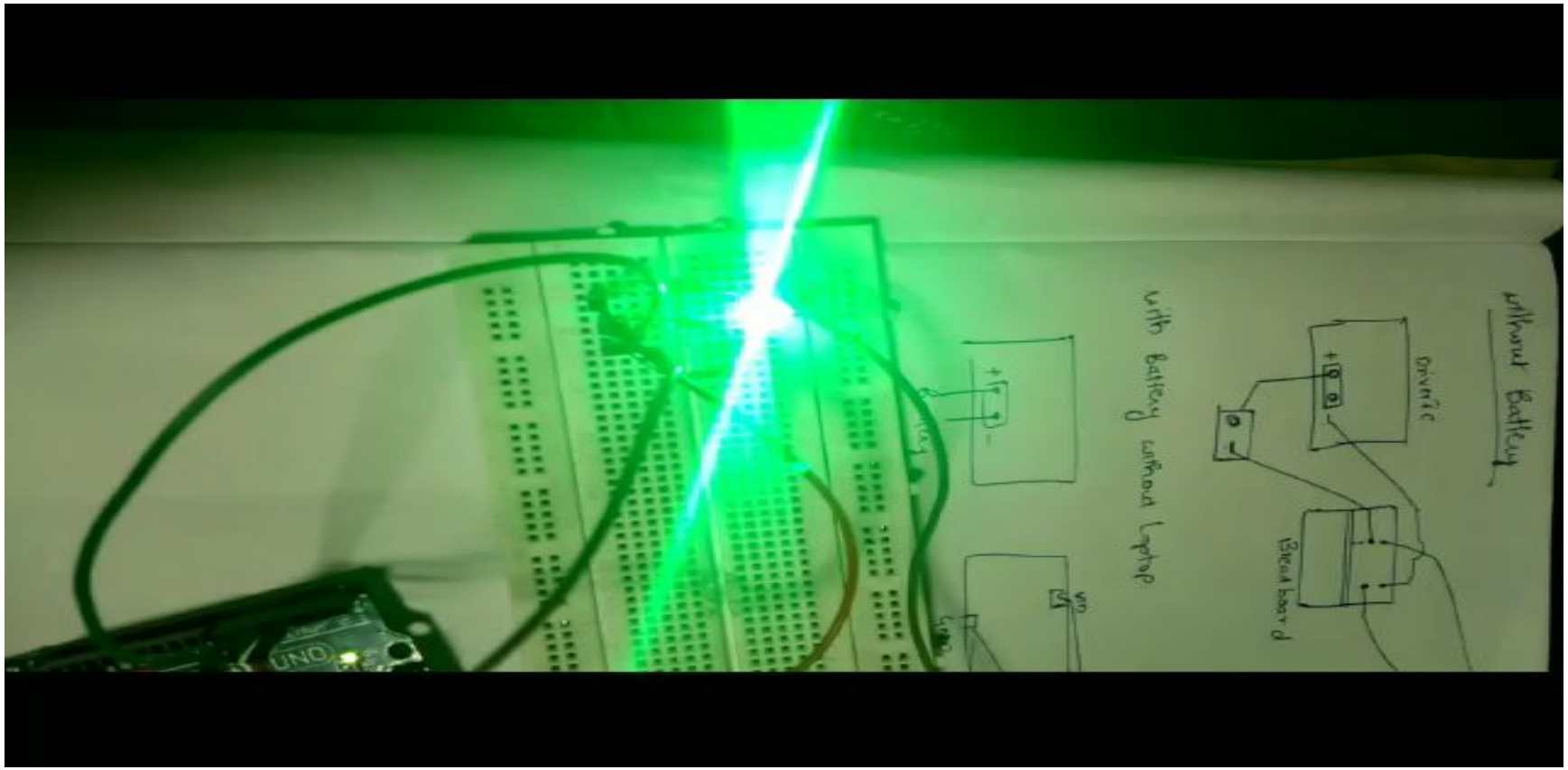
VOICE CONTROL ROBOT



OBJECTS COUNT USING IR

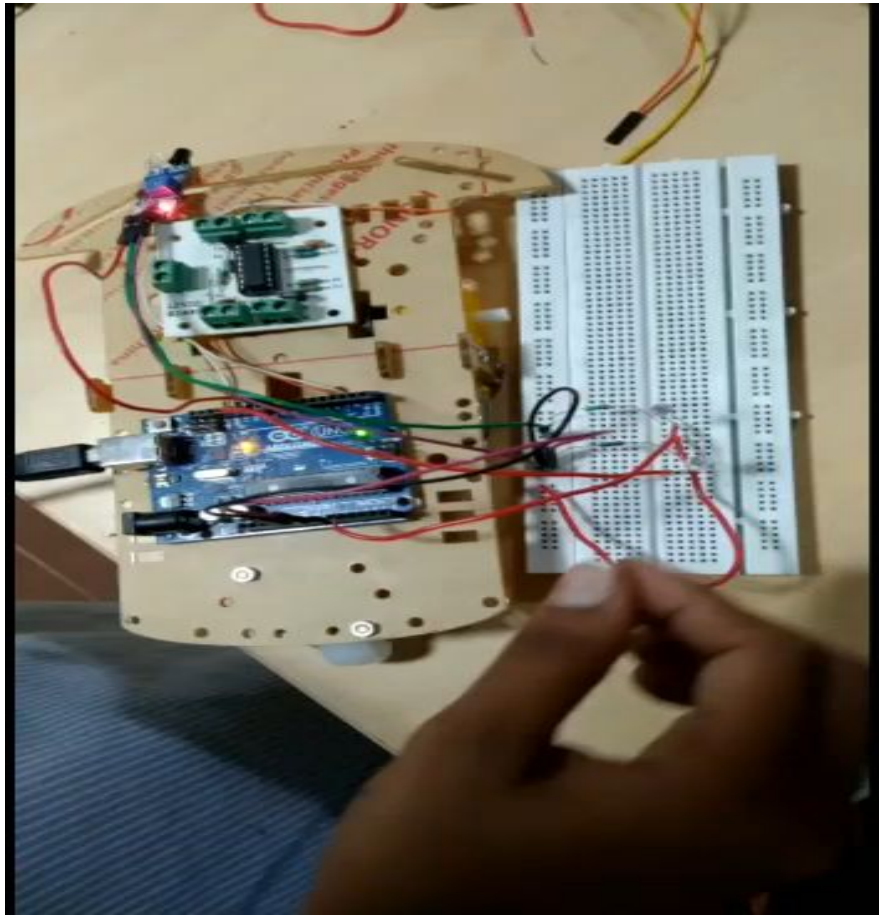


RGB


without Battery
Arduino
Bread board
with Battery without Laptop
UNO

IRRIGATION

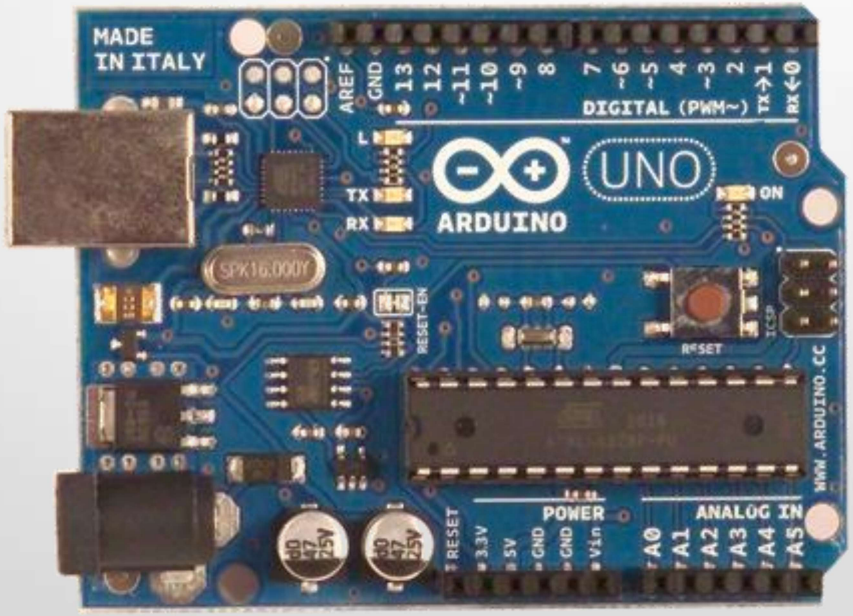
IR WITH LDR



Rudra Pratap Suman

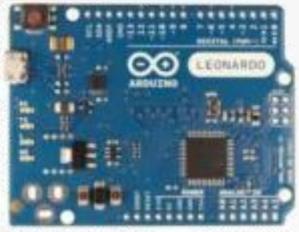
What is an Arduino ?

- **Open Source** electronic prototyping **platform** based on flexible **easy to use** hardware and software.



Arduino Family

Arduino Uno



Arduino Leonardo

Arduino Mega ADK

Arduino Ethernet

Arduino Due

Arduino Yún

Arduino Mega 2560

Arduino Mini

The Accessories

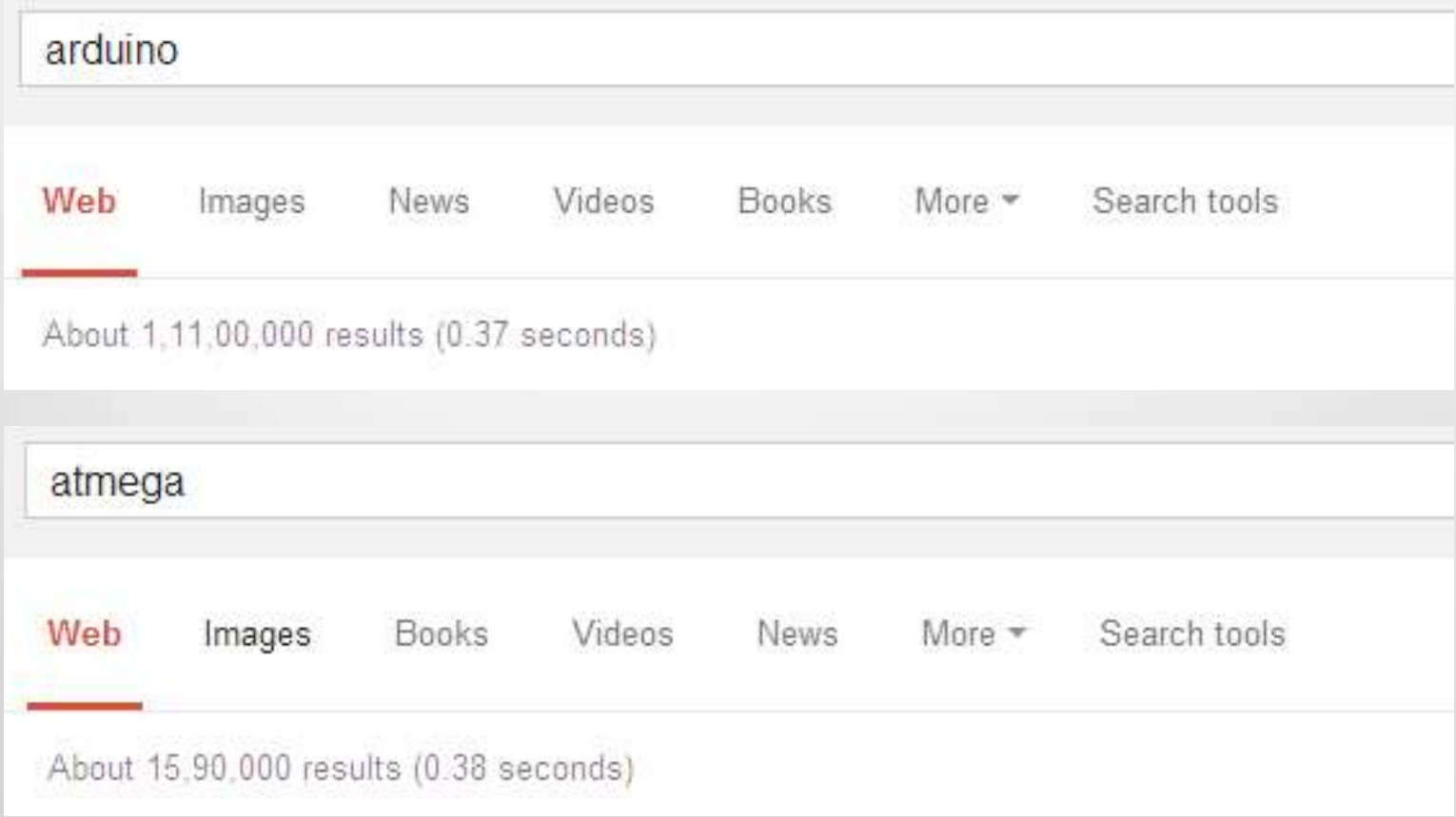
Arduino

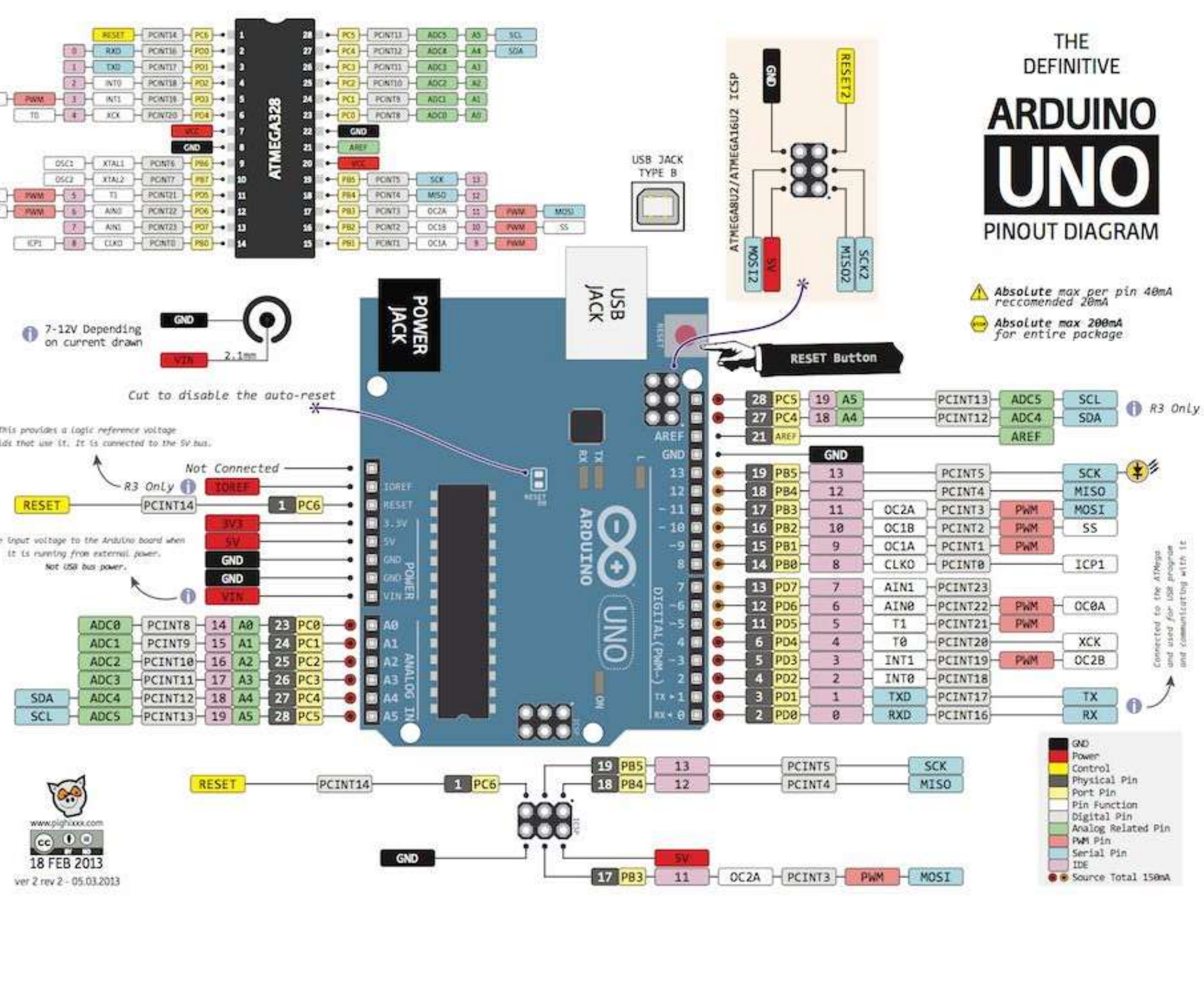
Arduino Shield

A Summary of Arduino power

Name	Processor	Operating Voltage/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [KB]	SRAM [KB]	Flash [KB]	USB	UART
Uno	ATmega328	5 V/7-12 V	16 Mhz	6/0	14/6	1	2	32	Regular	1
Due	AT91SAM3X8E	3.3 V/7-12 V	84 Mhz	12/2	54/12	-	96	512	2 Micro	4
Leonardo	ATmega32u4	5 V/7-12 V	16 Mhz	12/0	20/7	1	2.5	32	Micro	1
Mega 2560	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V/7-12 V	16 Mhz	16/0	54/15	4	8	256	Regular	4

Who is more popular Atmega or Arduino?




THE
DEFINITIVE
ARDUINO
UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK TYPE B
POWER JACK
USB JACK
RESET Button
Cut to disable the auto-reset
7-12V Depending on current drawn
2.1mm
Absolute max per pin 40mA reccomended 20mA
Absolute max 200mA for entire package
R3 Only
Not Connected
This provides a logic reference voltage
Input voltage to the Arduino board when it is running from external power.
Not USB bus power.
Connected to the ATMega and used for USB program and communicating with it
ARDUINO
UNO
DIGITAL (PWM~)
ANALOG IN
POWER
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA
www.pighixxx.com
18 FEB 2013
ver 2 rev 2 - 05.03.2013

Bare minimum code

```
void setup() {
    // put your setup code here, to run once:

}


void loop() {
    // put your main code here, to run repeatedly:

}
```

Bare minimum code

- setup : It is called only when the Arduino is powered on or reset. It is used to initialize variables and pin modes
- loop : The loop functions runs continuously till the device is powered off. The main logic of the code goes here. Similar to while (1) for micro-controller programming.

PinMode

- A pin on arduino can be set as input or output by using pinMode function.
- pinMode(13, OUTPUT); // sets pin 13 as output pin
- pinMode(13, INPUT); // sets pin 13 as input pin

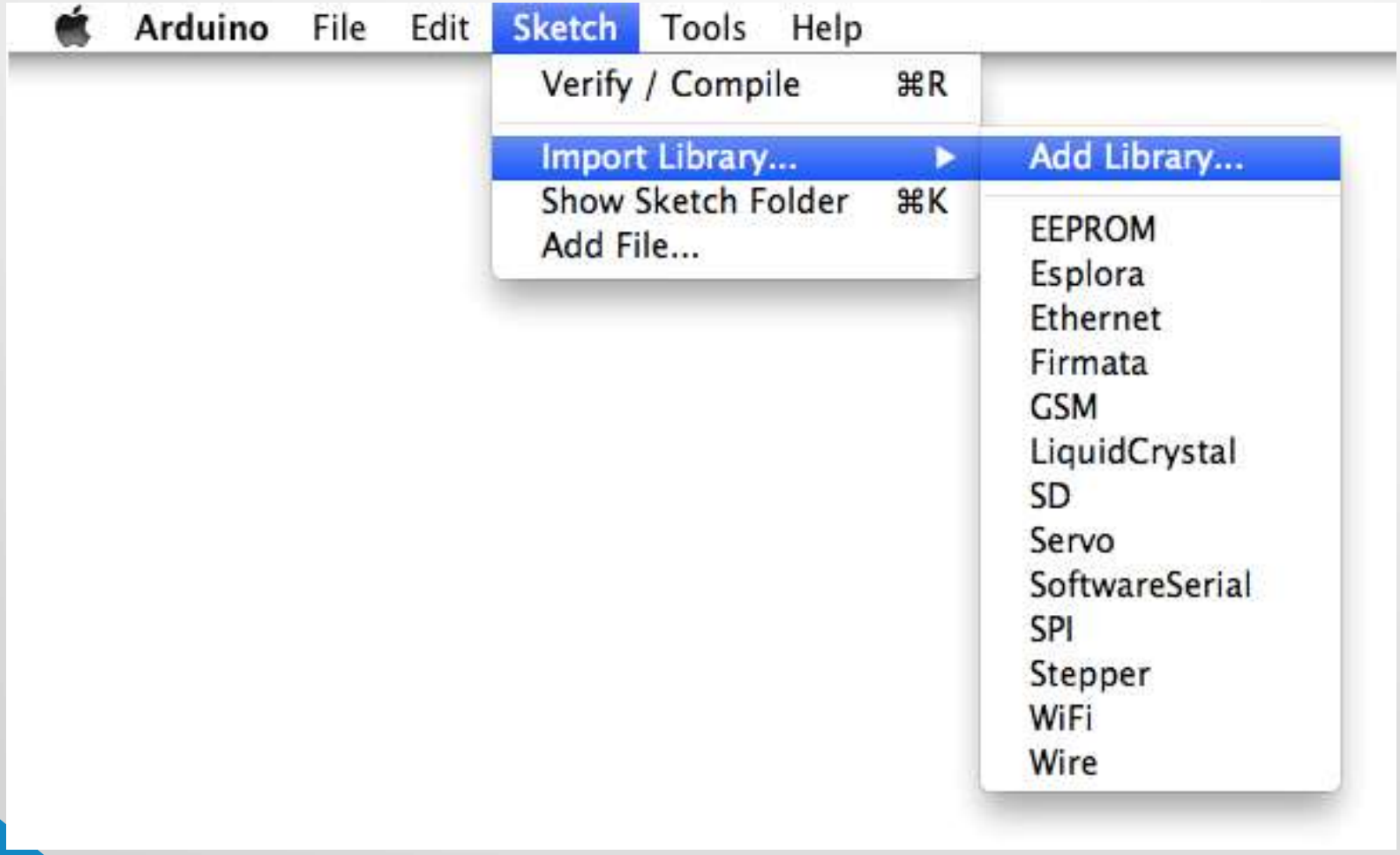
Reading/writing digital values

- digitalWrite(13, LOW); // Makes the output voltage on pin 13 , 0V
- digitalWrite(13, HIGH); // Makes the output voltage on pin 13 , 5V
- int buttonState = digitalRead(2); // reads the value of pin 2 in buttonState

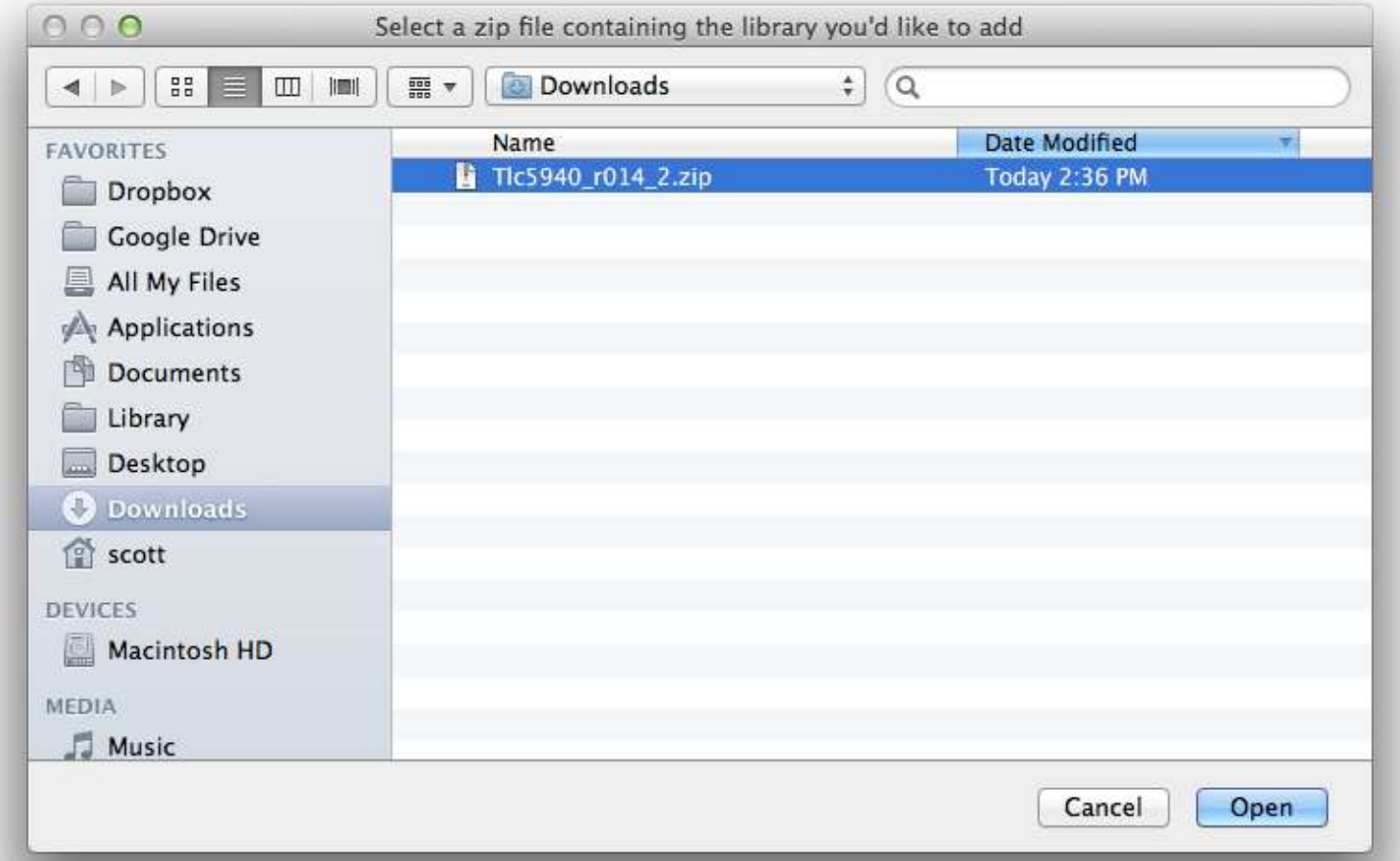
What are Libraries?

- Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download.

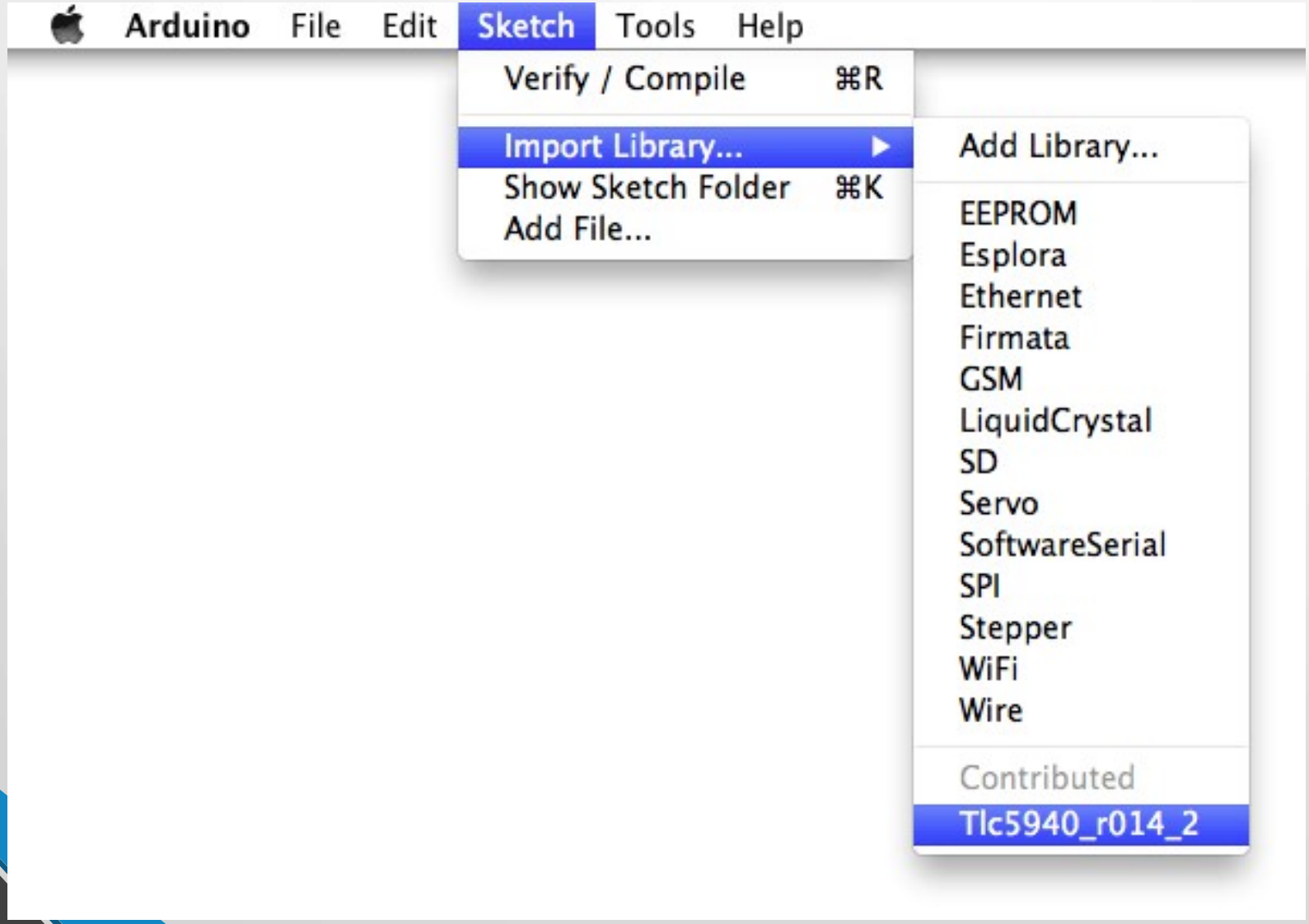
How to use them?



How to use them?



How to use them?



Arduino day

Think Make Share

Seven_segment Display with voice controller

Introduction:-

Seven_segment display(diodes) take on and off by interfacing it with bluetooth_module.Blue tooth module is a type of sensor which can recieve and transmitt the data to aurdino with the help of any interface .it takes commands from voice controller and responds according with that

Ex:-if command is “seven” then sevensegment _display displays seven number on it.

Description:-

Seven _segment_display:-A seven segment display is a form of electronic display device for displaying deciamal numerals.these segment contain seven led and those are assembled to structure like numeral .A light emitting diode is set according to shown in figure.seven segment displays are widely used in digital clocks,calculators,and other numerical display information.

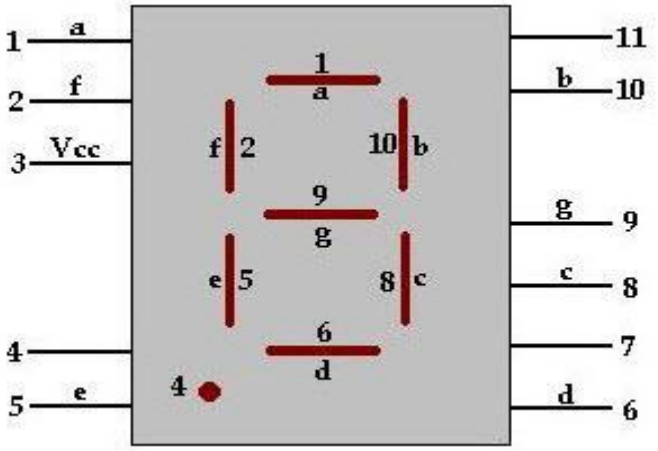
In simple led package,typically all of the cathodes(negative terminals) or all of the anodes(positive terminals) of the segment leds are connected to each other and connected to anode or cathod called “common anode” or “common cathod”.

We can display numbers from [0 to 9] and letters like b,c etc..

##Common anode:- All the anode connetions of the led segments are connected together to logic high .The seperate segments are lightened by applying of the logic low signal through a current limiting resistor to the cathode of the particular segment.

##Common cathode:-All the cathode connections of the led segments are connected togetehr to logic low.The seperate segments are lightened by applying of the logic high signa through a current limiting resistor to the anode of the particular segment.

PINS DESCRIPTION:-

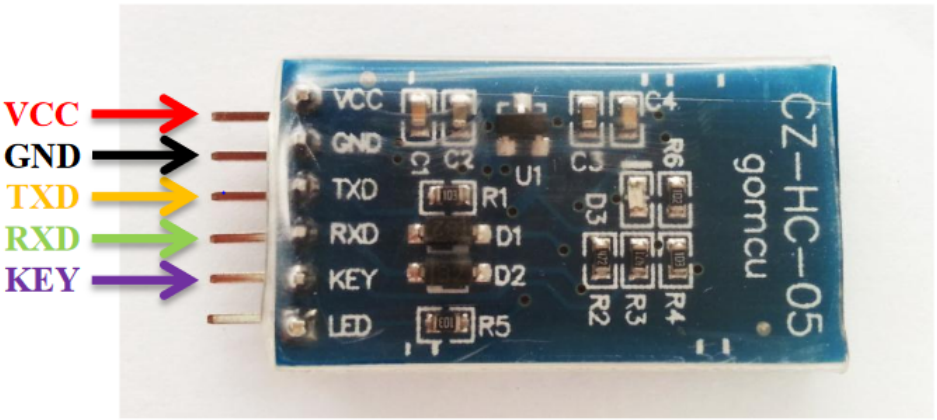


PIN	DESCRIPTION	FUNCTION
1	Connecetd to a	Led working
2	Connected to f	Led working
3	Vcc (+5v)	Supply pin
4	Dot pin	Led working
5	Connected to e	Led working
6	Connected to d	Led working
7	Groung(0v)	Supply pin
8	Connected to c	Led working
9	Connected to g	Led working
10	Connected to b	Led working

Bluetooth module:-

Bluetooth module (HC-05) is designed for transmission.wireless serial connection setup .its communication is via serail communication which makes an easy way to inerface with controller/PC/aurdino.you nedd an interface like any app or internet cloud for transmission of data. It consists of 5 pins. it takes inputs from user by usong any app interface Between HC-05module and user. you can use interface like cloud, app etc...by this yo can give commands to the sevensegment display to which number has to be displayed.. the voice commands taken through reciever of bluetooth and send to via transmitter to seven segmenet display.

PIN	DESCRIPTION	FUNCTION
VCC	+5v	supply
GND	Groung(0v)	Supply to 0v
TXD	Transmiietr pin	Transmiit the data to display
RXD	Reciver pin	Recieve the data from voice
KEY	Mode switch input	AT command pin
STATE	indicator	HC_05 on/off

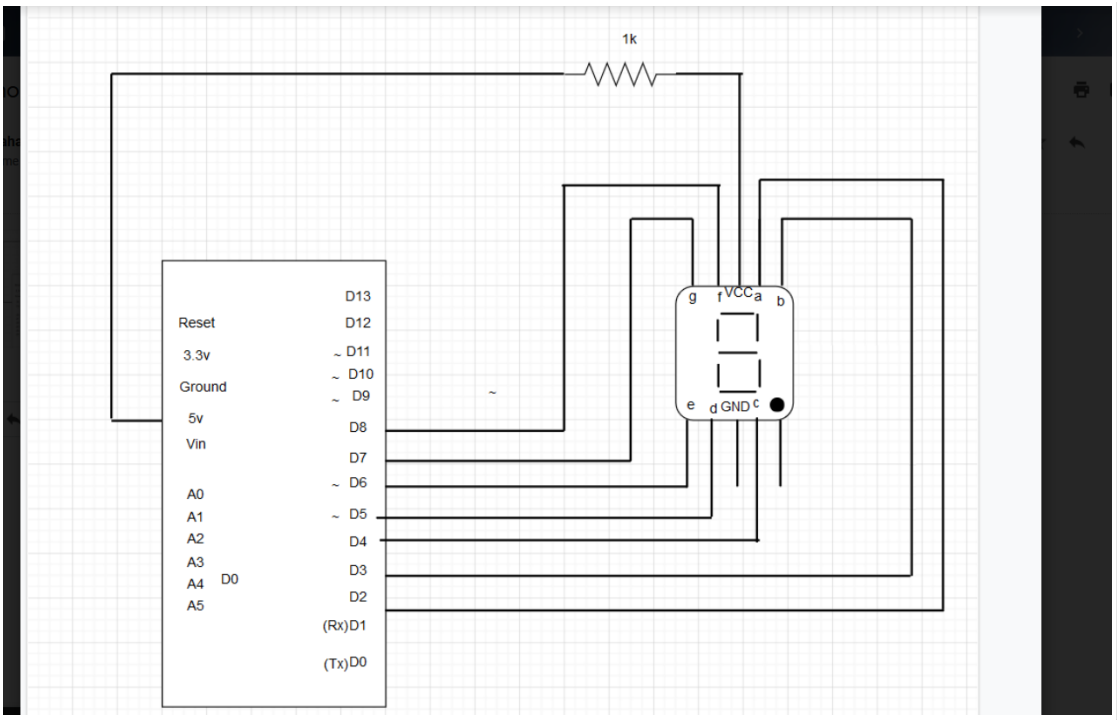


Aurdino:-

Aurdinois a open source electornic platform based on easy to use hardware and software.aurdino boards are able to take inputs from sensor
And trun it into output to activate output devices like led,motor etc..
It can take set of instructions to the microcontroller on the board.to do this module has a seperate provision to provide supply.

Cicuit operation:-

As the connections are according to the circuit.the blutooth module take the data(eg:-1,2,3)from the voice controller and transmiit to the aurdino board. There the segment displays the number based on the command given.we can check by it with serial monitor also.



Programme:-

```
char val;
int a=2;
int b=3;
int c=4;
int d=5;
int e=6;
int f=7;
int g=8;
void setup()
{
 Serial.begin(9600);
 pinMode(a,OUTPUT);//a
 pinMode(b,OUTPUT);//b
 pinMode(c,OUTPUT);//c
 pinMode(d,OUTPUT);//d
 pinMode(e,OUTPUT);//e
 pinMode(f,OUTPUT);//f
 pinMode(g,OUTPUT);//g
}
void loop()
{
  while(Serial.available())
  val=Serial.read();
  delay(3);
  if(val=='0')
 {
   Serial.println("zero");
   digitalWrite(a,LOW);
   digitalWrite(b,LOW);
   digitalWrite(c,LOW);
   digitalWrite(d,LOW);
```

```
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,HIGH);
  delay(1000);
}
else if(val=='1')
{
  Serial.println("one");
  digitalWrite(a,HIGH);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  delay(1000);
}
else if(val=='2')
{
  Serial.println("two");
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,HIGH);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,HIGH);
  digitalWrite(g,LOW);
  delay(1000);
}
else if(val=='3')
{
  Serial.println("three");
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,LOW);
  delay(1000);
}
else if(val=='4')
{
  Serial.println("FOUR");
  digitalWrite(a,HIGH);
  digitalWrite(b,HIGH);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
  digitalWrite(g,LOW);
  delay(1000);
}
else if(val=='5')
{
  Serial.println("FIVE");
  digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
```

```
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,HIGH);
  digitalWrite(f,LOW);
  digitalWrite(g,LOW);
  delay(1000);
 }
 else if(val=='6')
 {
  Serial.println("SIX");
  digitalWrite(a,LOW);
  digitalWrite(b,HIGH);
  digitalWrite(c,LOW);
  digitalWrite(d,LOW);
  digitalWrite(e,LOW);
  digitalWrite(f,LOW);
  digitalWrite(g,LOW);
  delay(1000);
 }
 else if(val=='7')
 {
  Serial.println("SEVEN");
  digitalWrite(a,LOW);
  digitalWrite(b,LOW);
  digitalWrite(c,LOW);
  digitalWrite(d,HIGH);
  digitalWrite(e,HIGH);
  digitalWrite(f,HIGH);
  digitalWrite(g,HIGH);
  delay(1000);
 }
 else if(val=='8')
 {
  Serial.println("EIGHT");
  digitalWrite(a,LOW);
  digitalWrite(b,0);
  digitalWrite(c,0);
  digitalWrite(d,0);
  digitalWrite(e,0);
  digitalWrite(f,0);
  digitalWrite(g,0);
  delay(1000);
 }
 else if(val=='9')
 {
  Serial.println("NINE");
  digitalWrite(a,0);
  digitalWrite(b,0);
  digitalWrite(c,0);
  digitalWrite(d,1);
  digitalWrite(e,1);
  digitalWrite(f,0);
  digitalWrite(g,0);
 }
}
```

Referennces:-

1. https://www.engineersgarage.com/contributions/bluetooth-controlled-seven-segment-display/

VOTING MACHINE

Working:

It counts the number of votes when you swich the button.here we use push button to count the number of votes per party or union or any counting purpose.

Whenever press the push button it raises the count and displayed in the seven segment display or LCD display.

Components required:

1.Arduino Board

2.seven segment display.

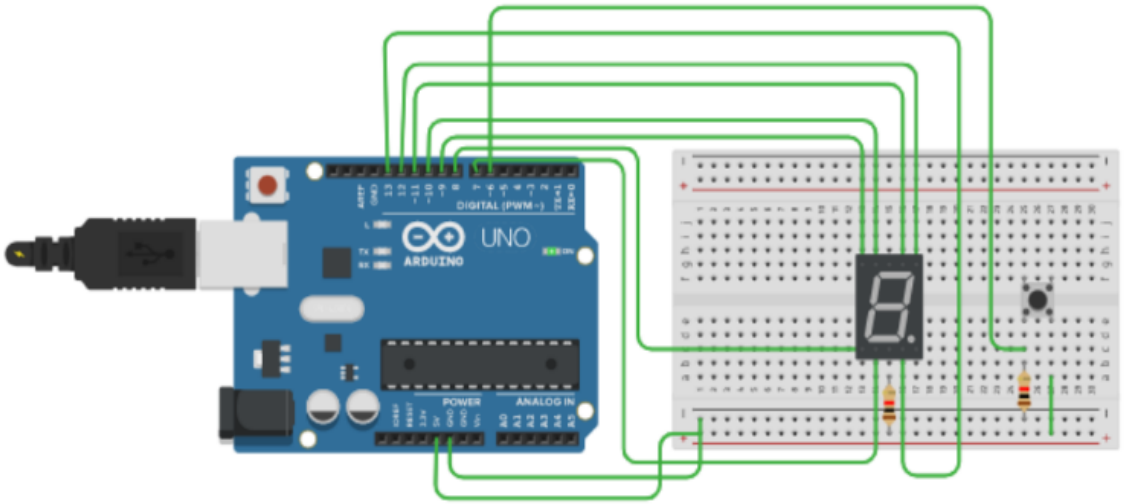
3.push button

4.resistors

5.connecting wires

6.bread board.

Circuit connections:



VOTING MACHINE

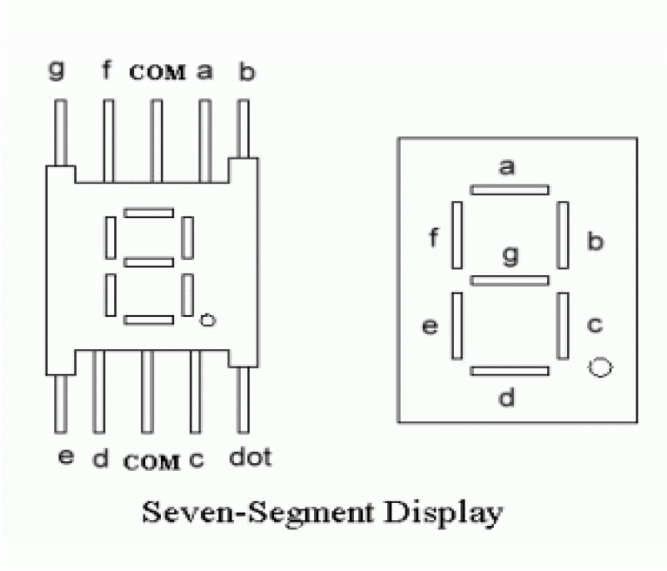
Arduino:

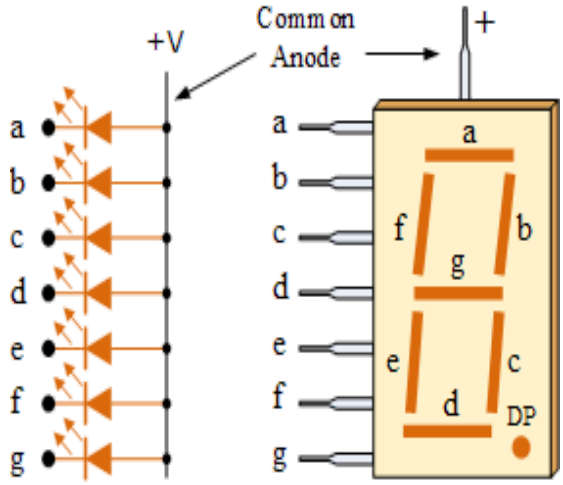
- Arduino is an open-source electronics prototyping platform based on flexible, easy-to use hardware and software.
- Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.
- The microcontroller on the board is programmed using the Arduino programming language(simplified c++) and the Arduino Development Environment. They can communicate with software running on a computer.
- The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

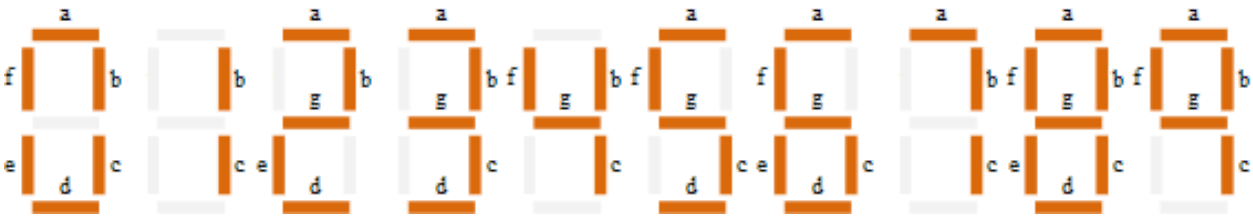
Seven segment Display:

- seven segment display", consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown.
- Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package.
- These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.
- there are therefore two types of LED 7-segment display called: **Common Cathode** (CC) and **Common Anode** (CA).

VOTING MACHINE







Push Button:

Push-Buttons are normally-open tactile switches. Push buttons allow us to power the circuit or make any particular connection only when we press the button.

Applications:

1.counting number of votes

2.counting number of persons entering and leaving a room

ELECTRONIC DIE

Working:

Generally we use a die to play games. Here we genrerate a number randomly using some random functions.we display a random in a seven segment display.

The random function is a inbuit function which can generate random number within a range of numbers.

Syntax:

R=random(x,y);

Components Required:

1.Arduino

2.Seven segment display

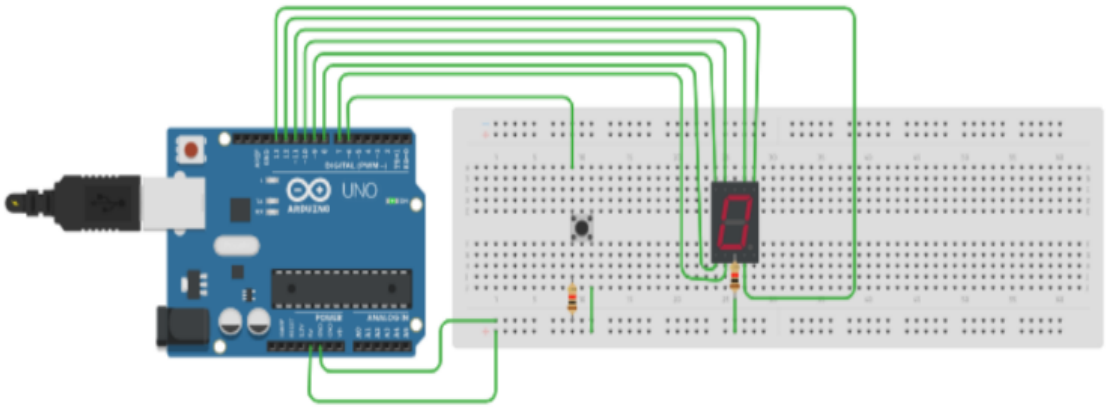
3.push button

4.resistors

5.connecting wires

6.breadboard

Circuit Connections:



ELECTRONIC DIE

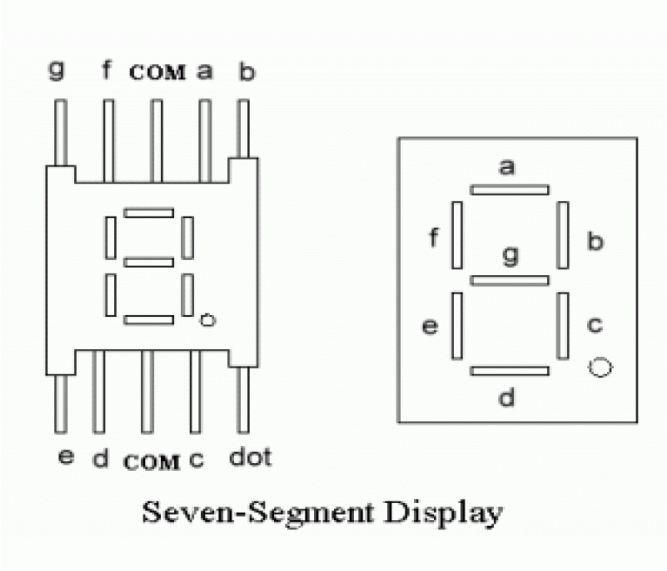
Arduino:

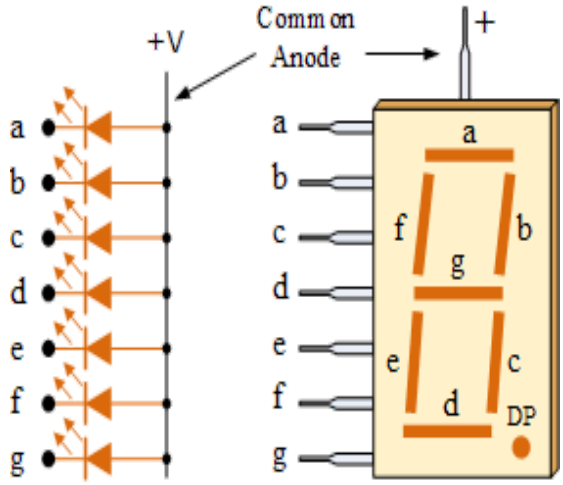
- Arduino is an open-source electronics prototyping platform based on flexible, easy-to use hardware and software.
- Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.
- The microcontroller on the board is programmed using the Arduino programming language(simplified c++) and the Arduino Development Environment. They can communicate with software running on a computer.
- The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

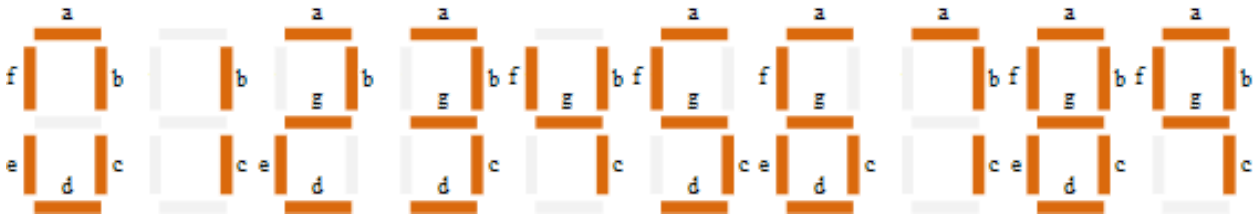
Seven segment Display:

- seven segment display", consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown.
- Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package.
- These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.
- there are therefore two types of LED 7-segment display called: **Common Cathode** (CC) and **Common Anode** (CA).

ELECTRONIC DIE







Push Button:

Push-Buttons are normally-open tactile switches. Push buttons allow us to power the circuit or make any particular connection only when we press the button.

Applicaion:

1.generating shuffle.

2.playing games.

3.selecting things randomly

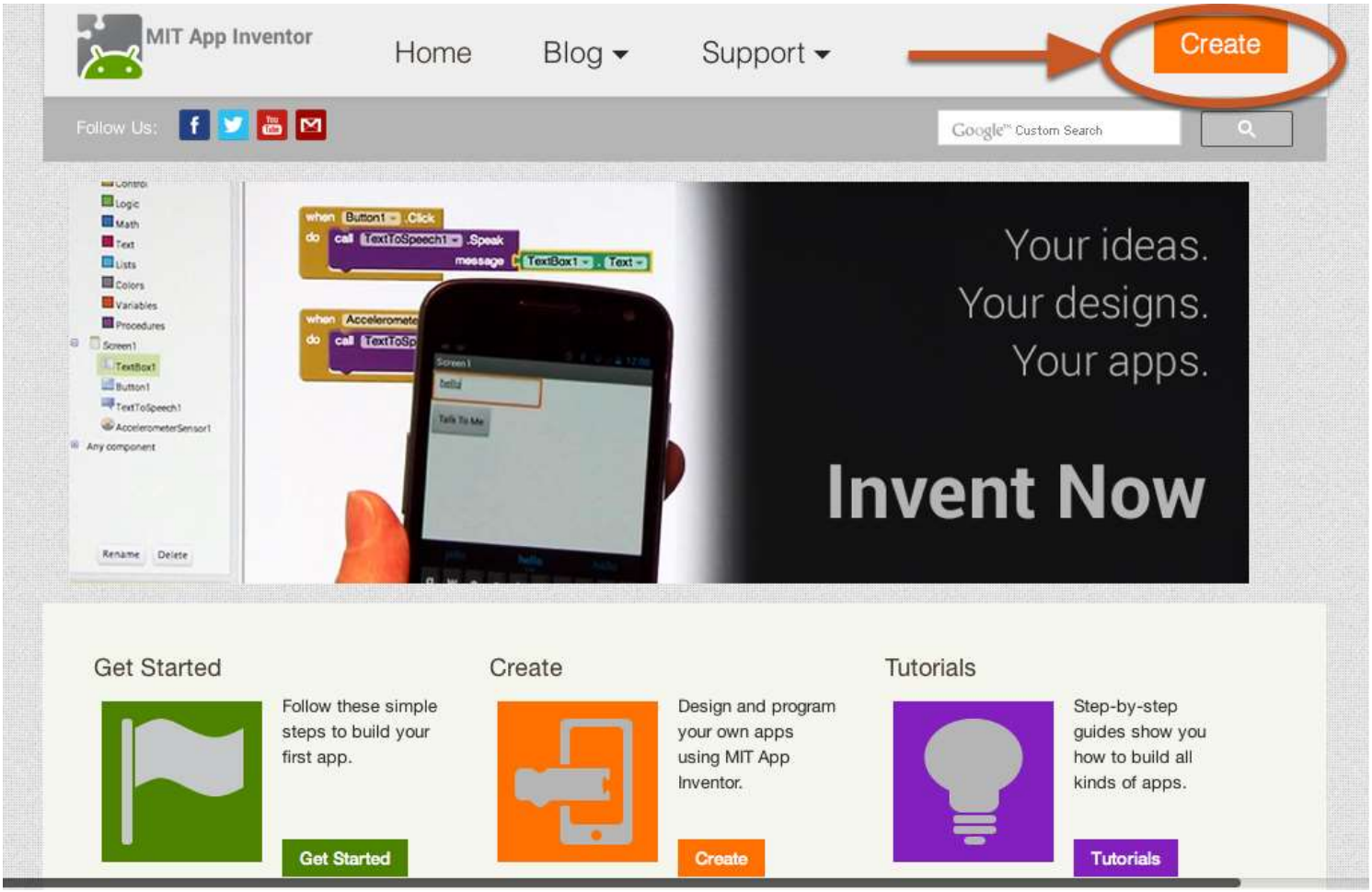
MIT App Inventor
appinventor.mit.edu

TalkToMe: Your first App Inventor app

This step-by-step picture tutorial will guide you through making a talking app.

To get started, go to App Inventor on the web.

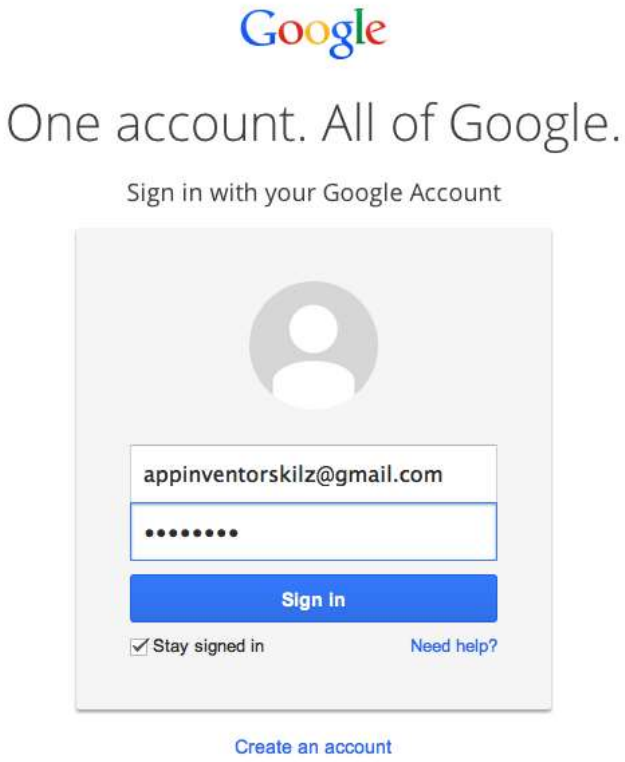
Go directly to **ai2.appinventor.mit.edu**, or click the orange "Create" button from the App Inventor website.



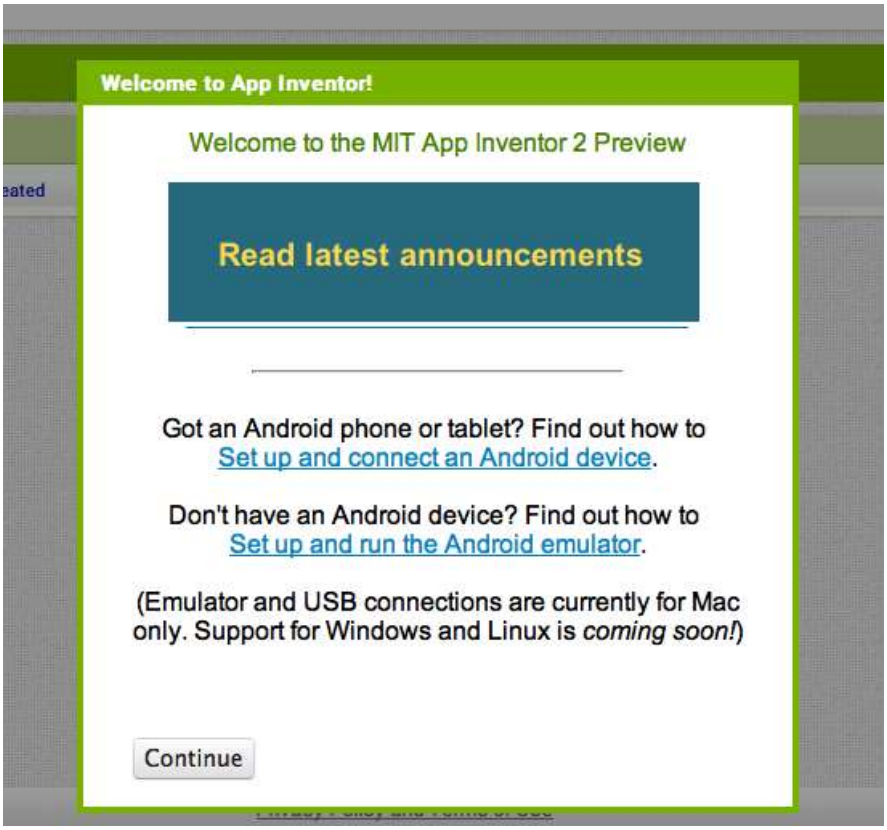
MIT App Inventor
appinventor.mit.edu

Log in to App Inventor with a gmail (or google) user name and password.

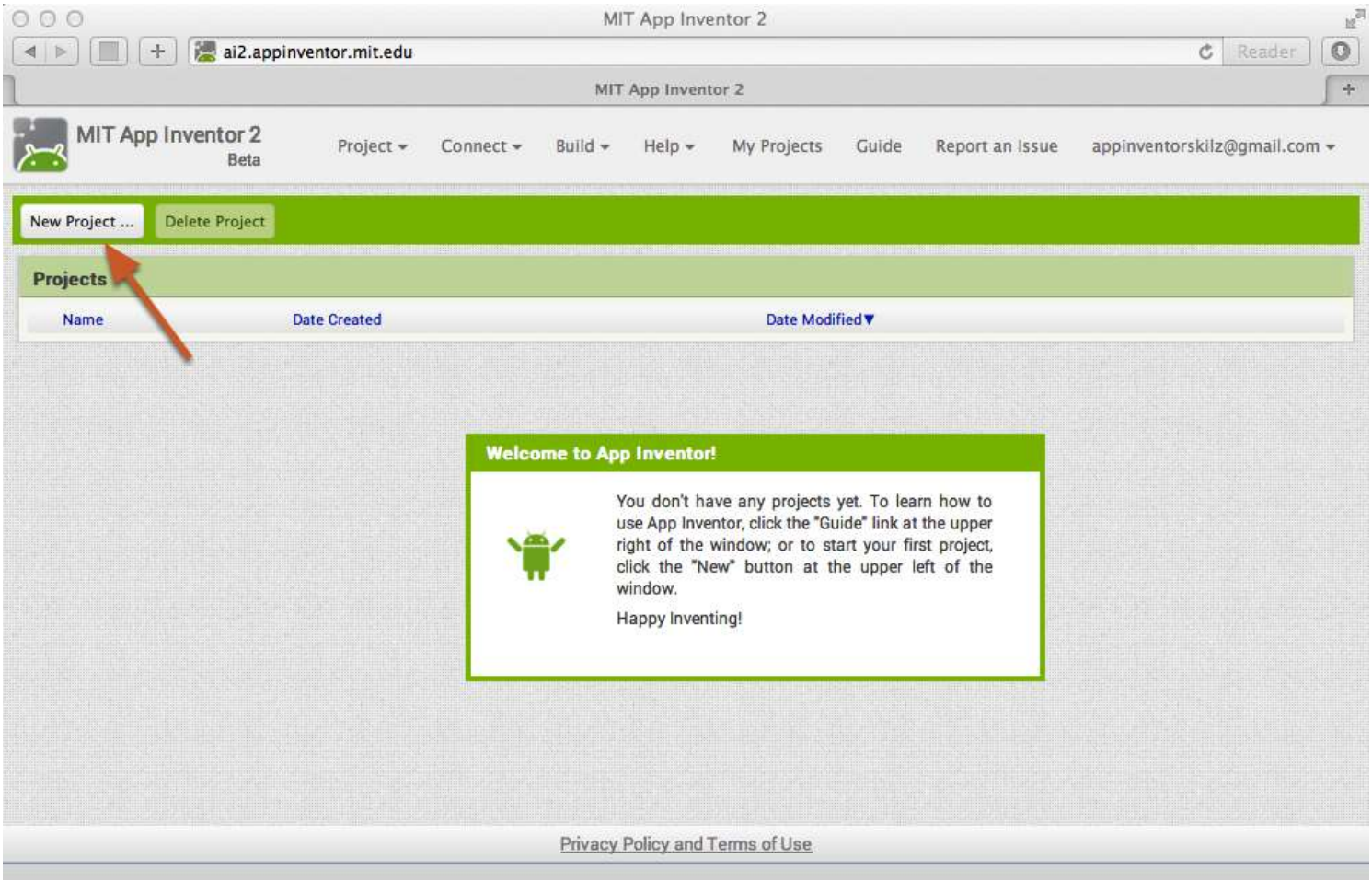
Use an existing gmail account or school-based google account to log in to ai2.appinventor.mit.edu
To set up a brand new gmail account, go to accounts.google.com/SignUp



Click "Continue" to dismiss the splash screen.

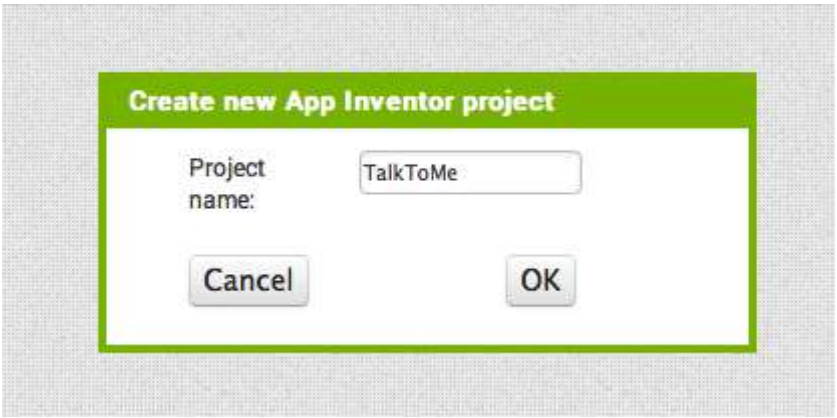


Start a new project.



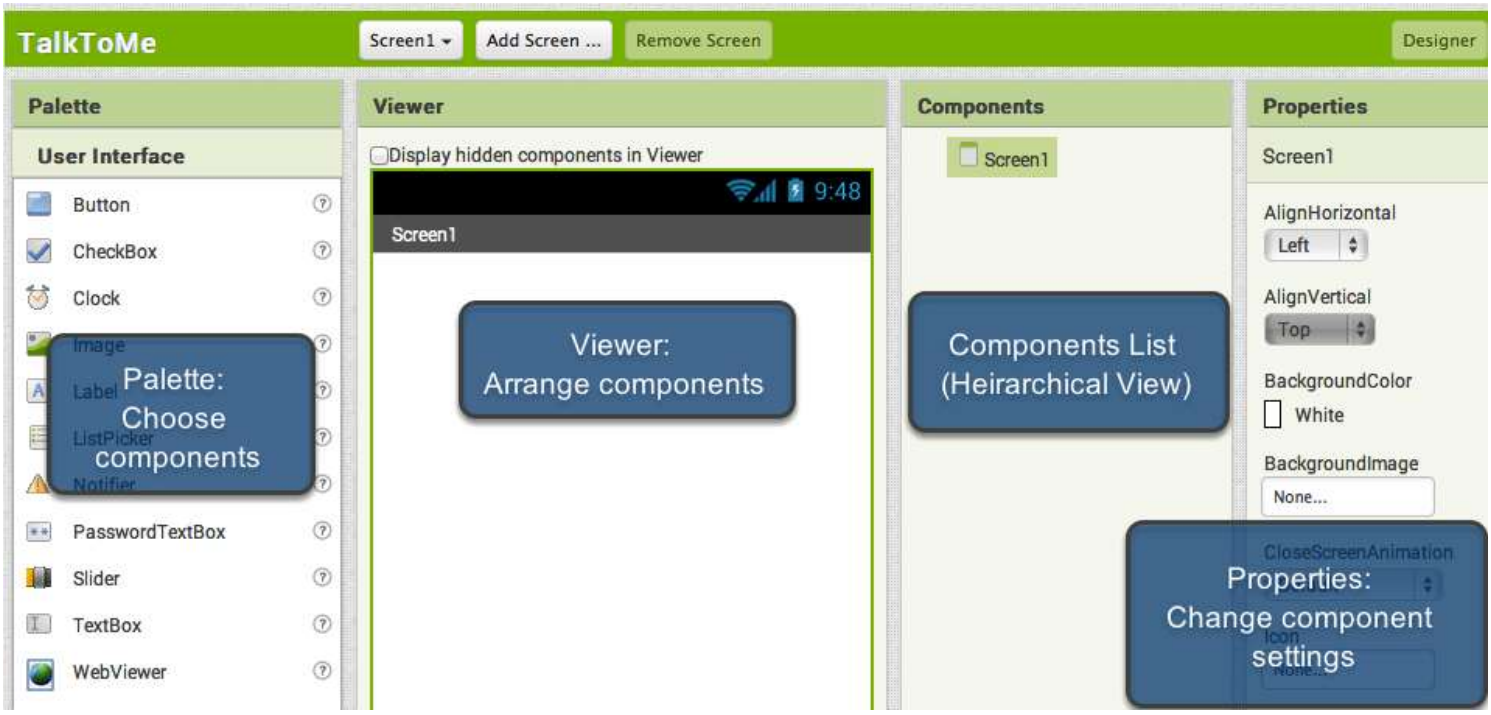
Name the project "TalkToMe" (no spaces!)

Type in the project name (underscores are allowed, spaces are not) and click OK.



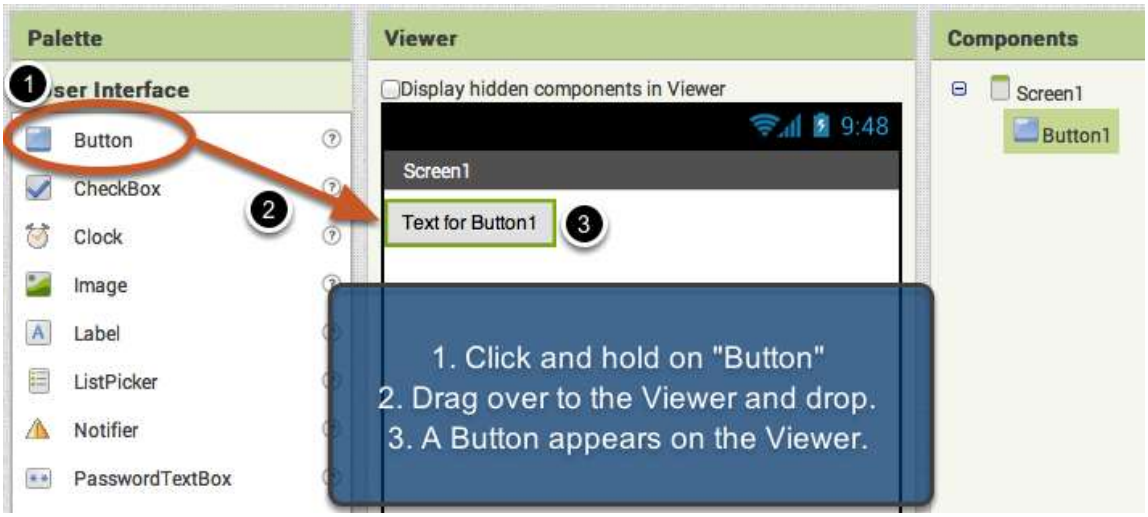
You are now in the Designer, where you lay out the "user interface" of your app.

The Design Window, or simply "Designer" is where you lay out the look and feel of your app, and specify what functionalities it should have. You choose things for the user interface things like Buttons, Images, and Text boxes, and functionalities like Text-to-Speech, Sensors, and GPS.



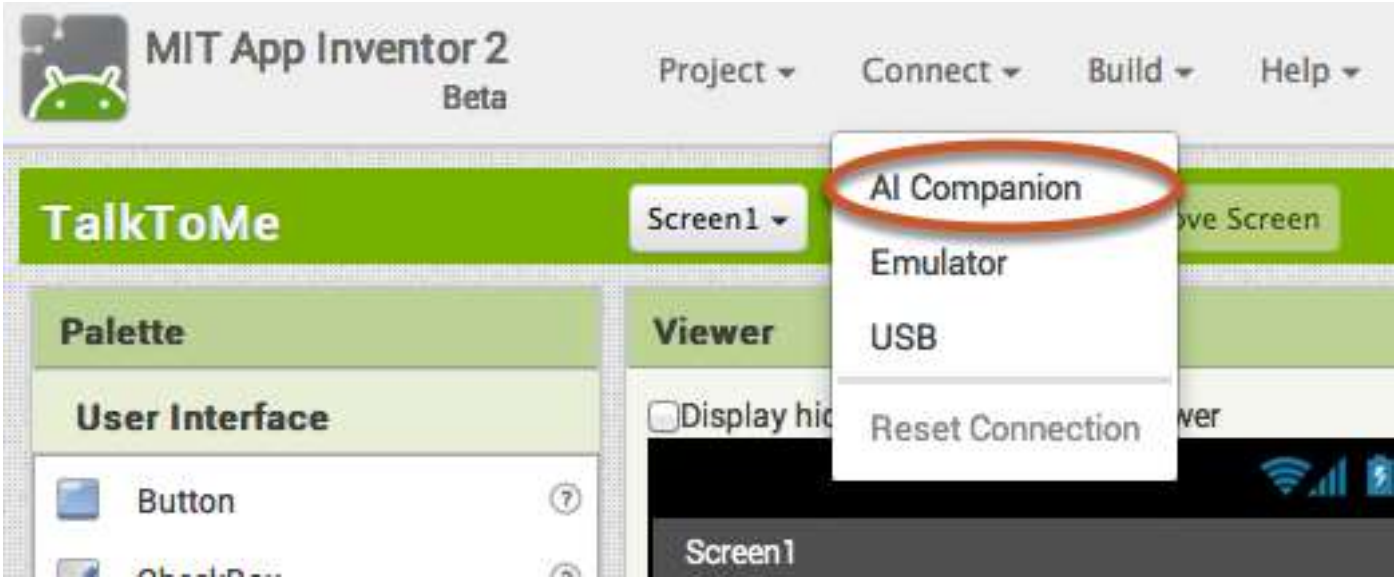
Add a Button

Our project needs a button. ***Click and hold*** on the word "Button" in the palette. ***Drag*** your mouse over to the Viewer. ***Drop*** the button and a new button will appear on the Viewer.



Connect App Inventor to your phone for live testing

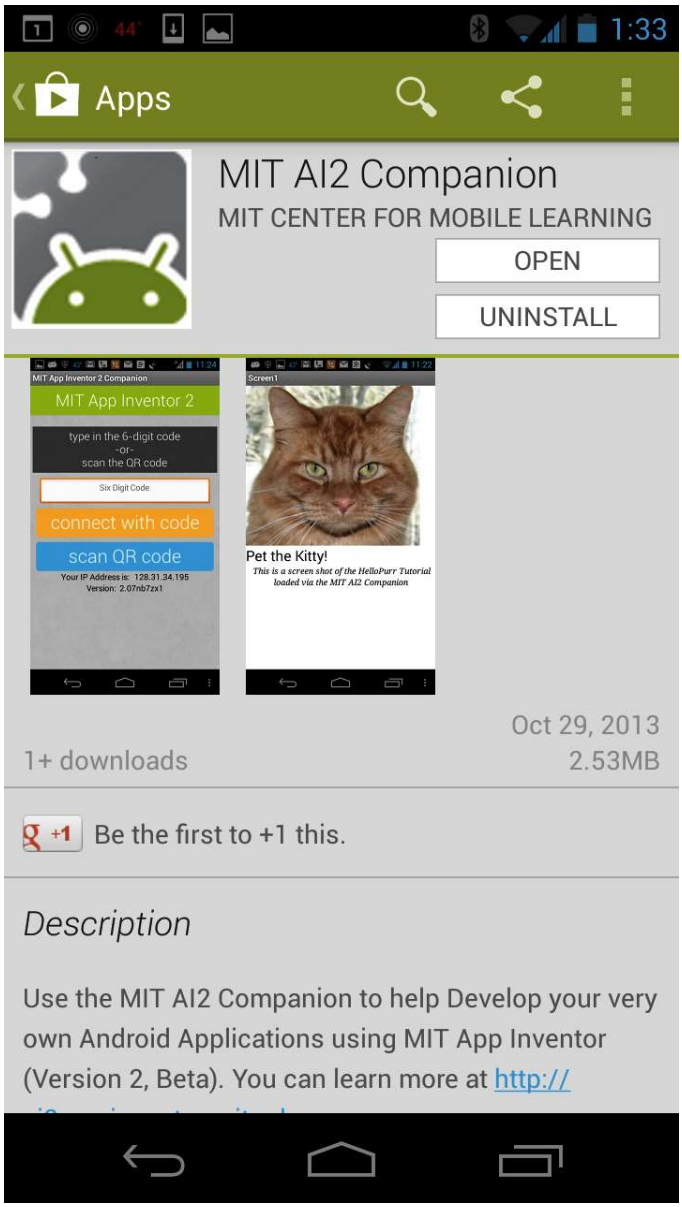
One of the neatest things about App Inventor is that you can see and test your app while you're building it, on a connected device. If you have an **Android phone or tablet, follow the steps below.** *If you do not have a device,* then follow the instructions for setting up the on-screen emulator (opens a new page) and then come back to this tutorial once you've gotten the emulator connected to App Inventor.



MIT App Inventor
appinventor.mit.edu

Get the MIT AI2 Companion from the Play Store and install it on your phone or tablet.

The preferred method for getting the AI2 Companion App is to **download the app from the Play Store by searching for "MIT AI2 Companion".**



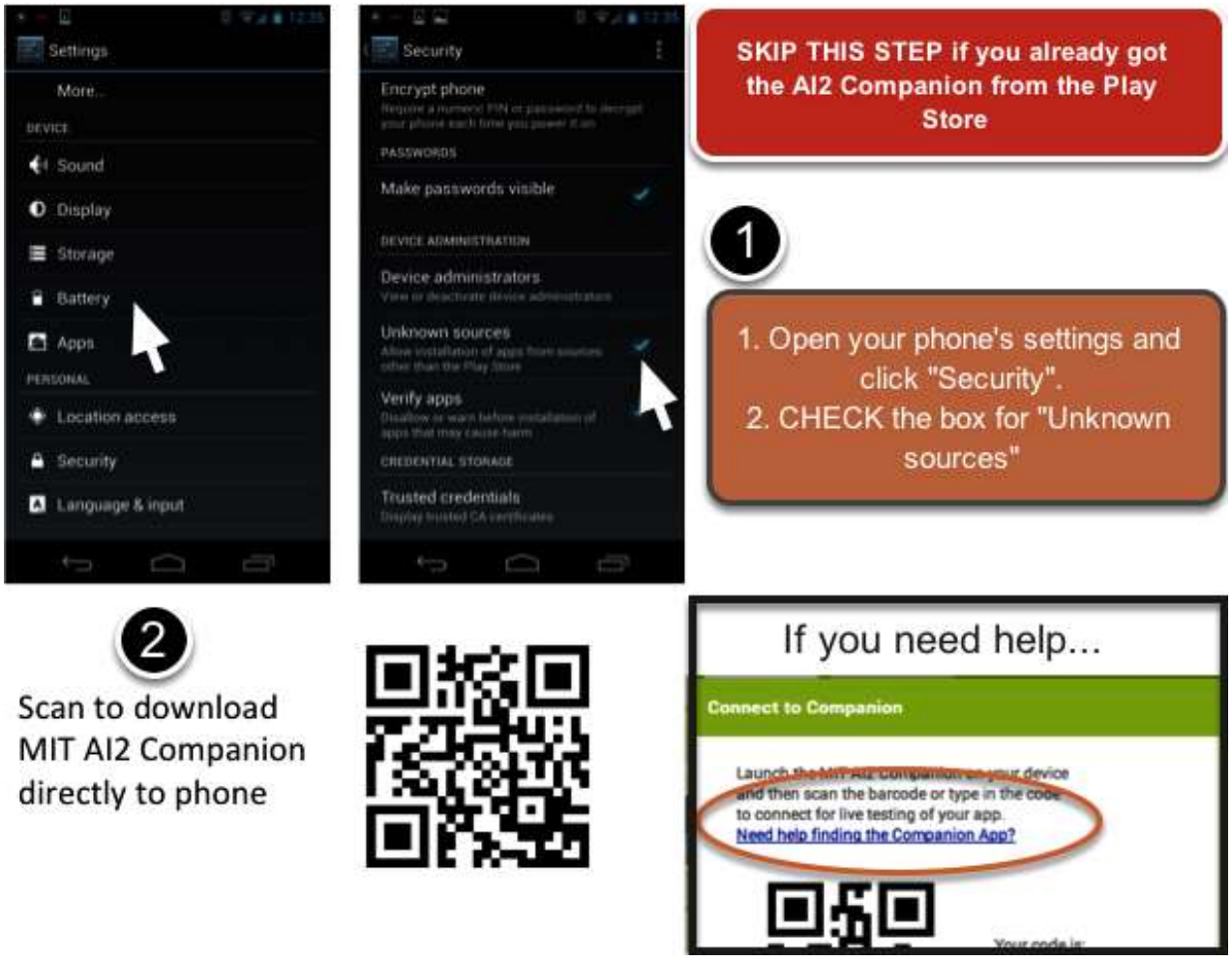
MIT App Inventor
appinventor.mit.edu

To download the AI2 Companion App to your device directly (SKIP THIS STEP IF YOU already got the app from Play Store)

If for some reason you can not connect to the Google Play store, you can download the AI2 Companion as described here.

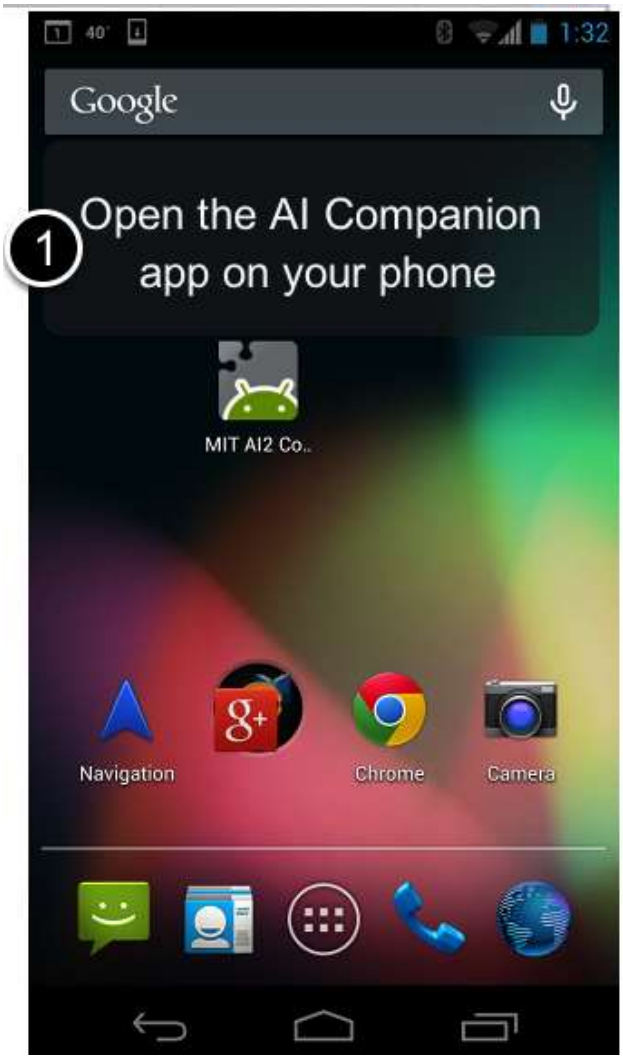
First, you will need to go into your phone's settings (#1), choose "Security", then scroll down to allow "Unknown Sources", which allows apps that are not from the Play Store to be installed on the phone.

Second, do one of the following:
A) **Scan the QR code above (#2)**
or
B) Click the "Need help finding..." link and you'll be taken to the download page. From there you can download the MITAI2Companion.apk file to your computer and then move it over to your device to install it.



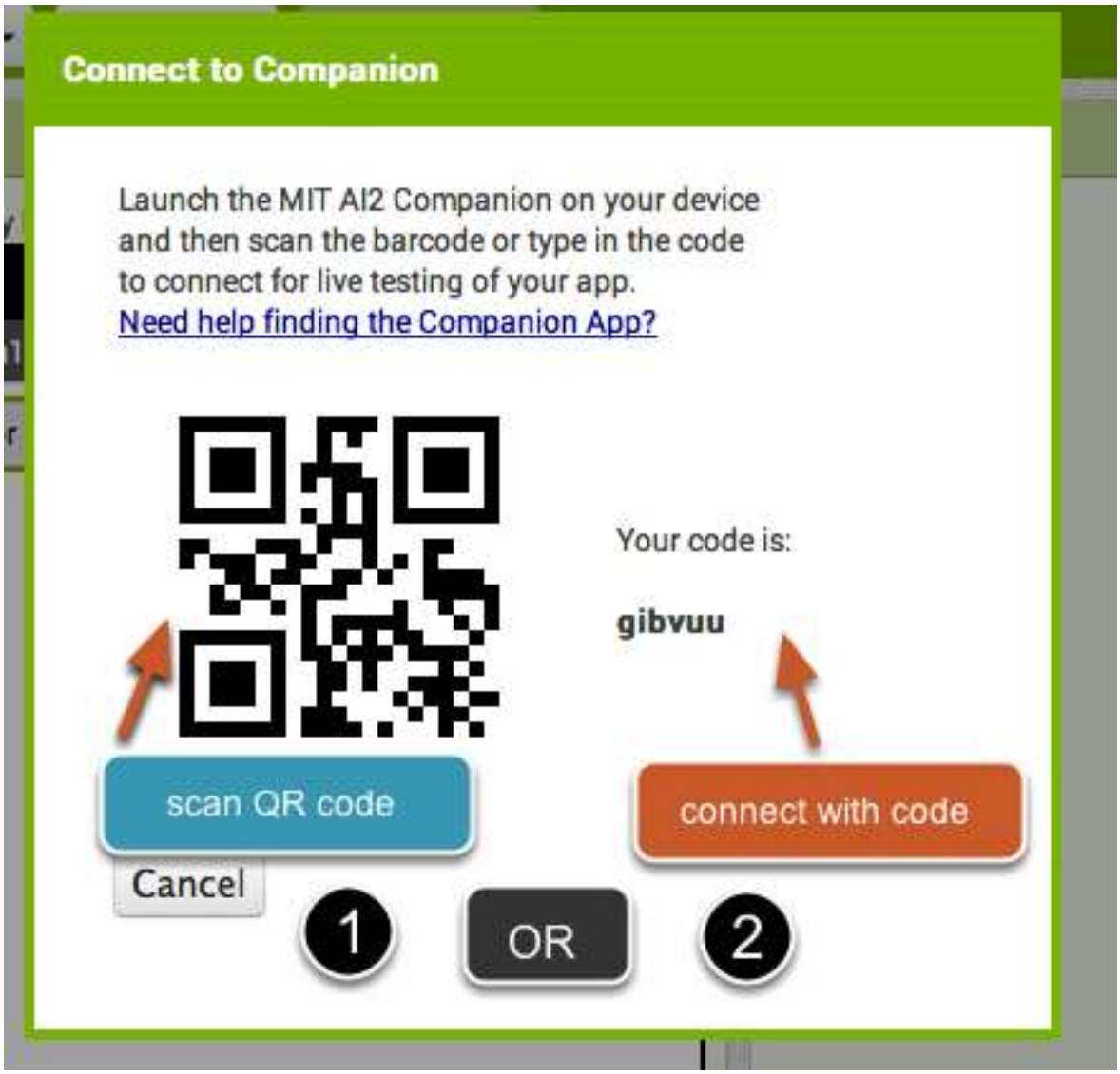
Start the AICompanion on your device

On your phone or tablet, click the icon for the MIT AI Companion to start the app. NOTE: Your **phone and computer must both be on the same wireless network.** Make sure your phone's wifi is on and that you are connected to the local wireless network. If you can not connect over wifi, go to the Setup Instructions on the App Inventor Website to find out how to connect with a USB cable.



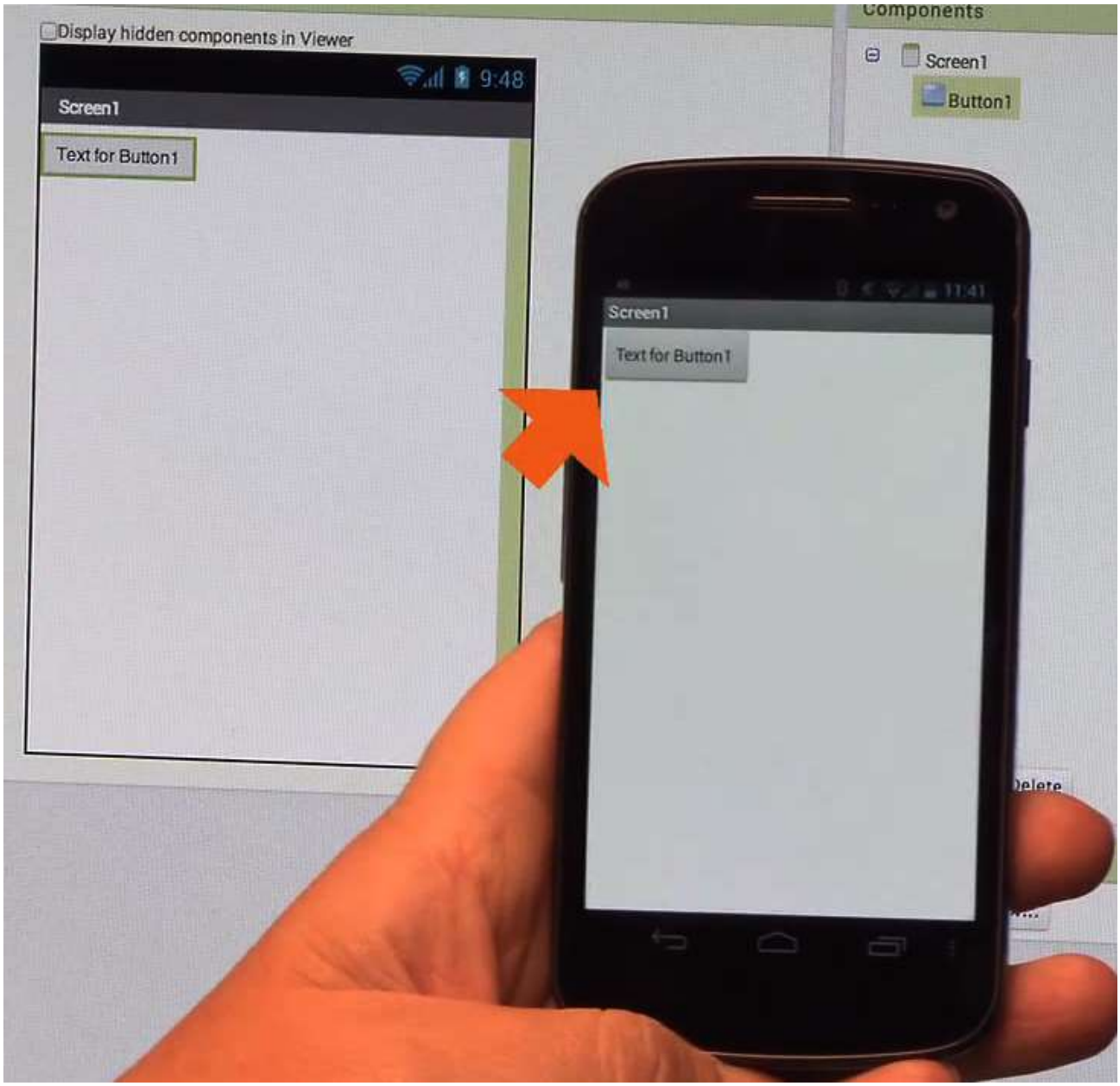
Get the Connection Code from App Inventor and scan or type it into your Companion app

On the Connect menu, choose "AI Companion". You can connect by:
1 - Scanning the QR code by clicking "Scan QR code" (#1).
or
2 - Typing the code into the text window and click "Connect with code" (#2).



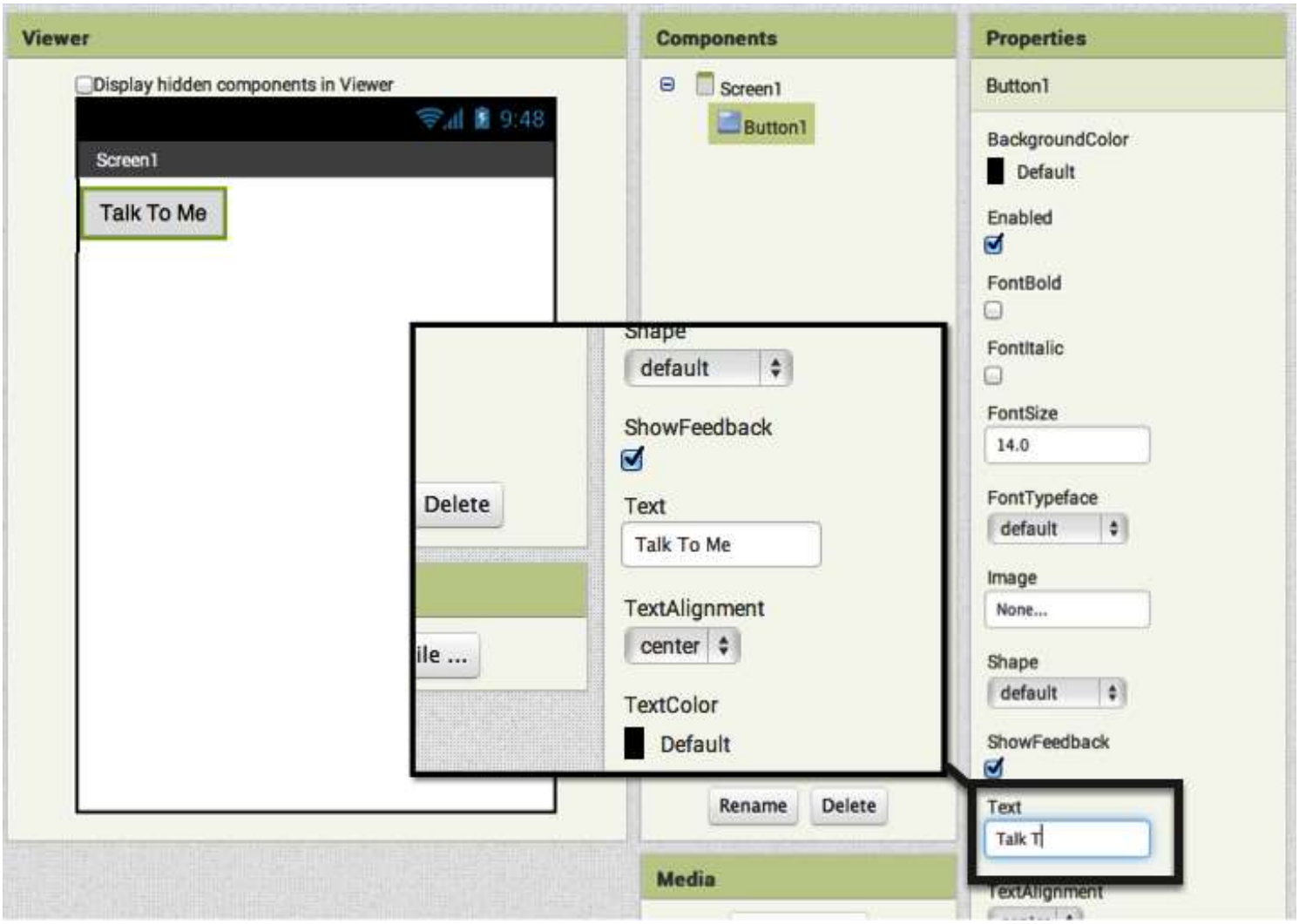
See your app on the connected device

You will know that your connection is successful when you see your app on the connected device. So far our app only has a button, so that is what you will see. As you add more to the project, you will see your app change on your phone.



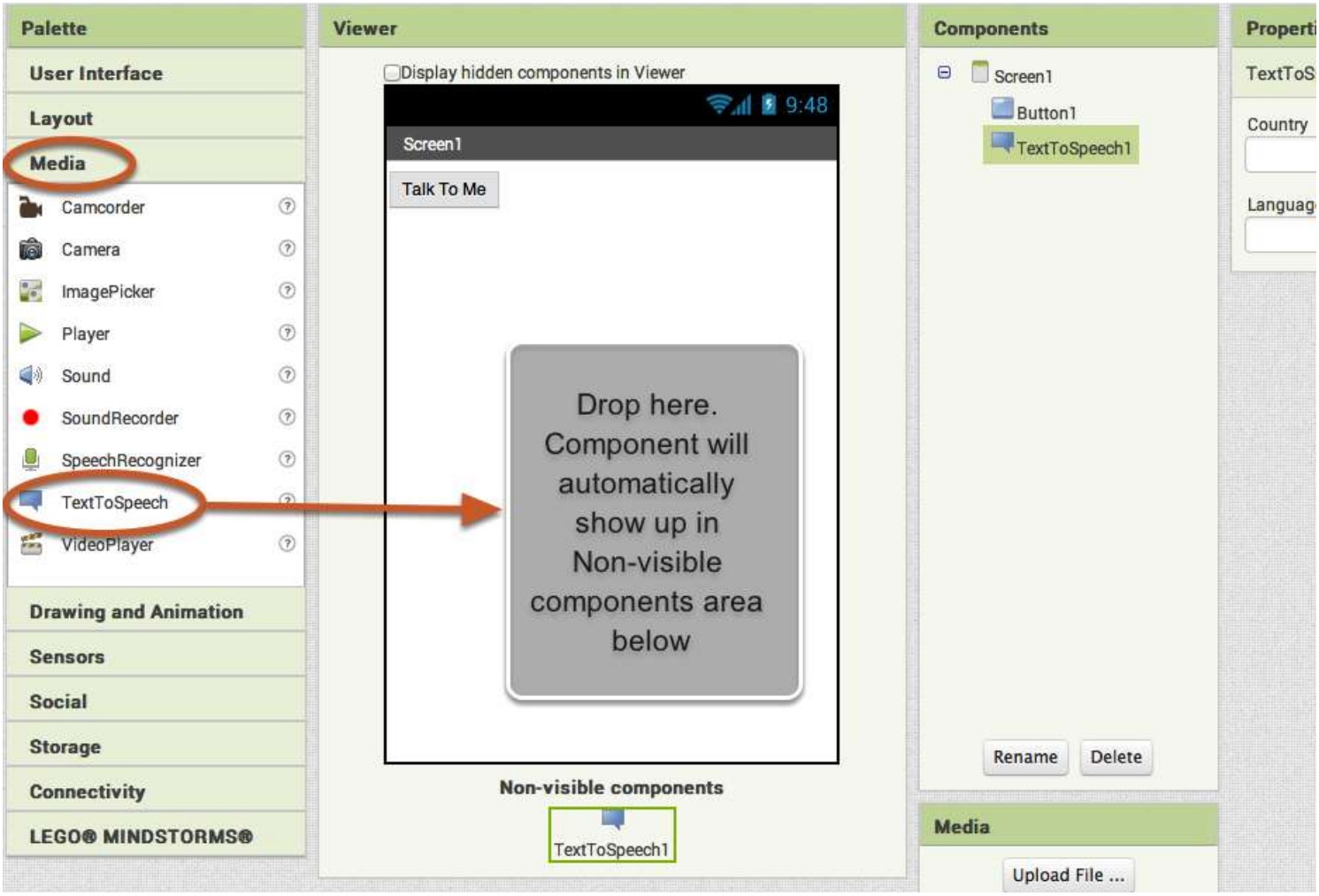
Change the Text on the Button

On the properties pane, change the text for the Button. Select the text "Text for Button 1", delete it and type in "Talk To Me". Notice that the text on your app's button changes right away.



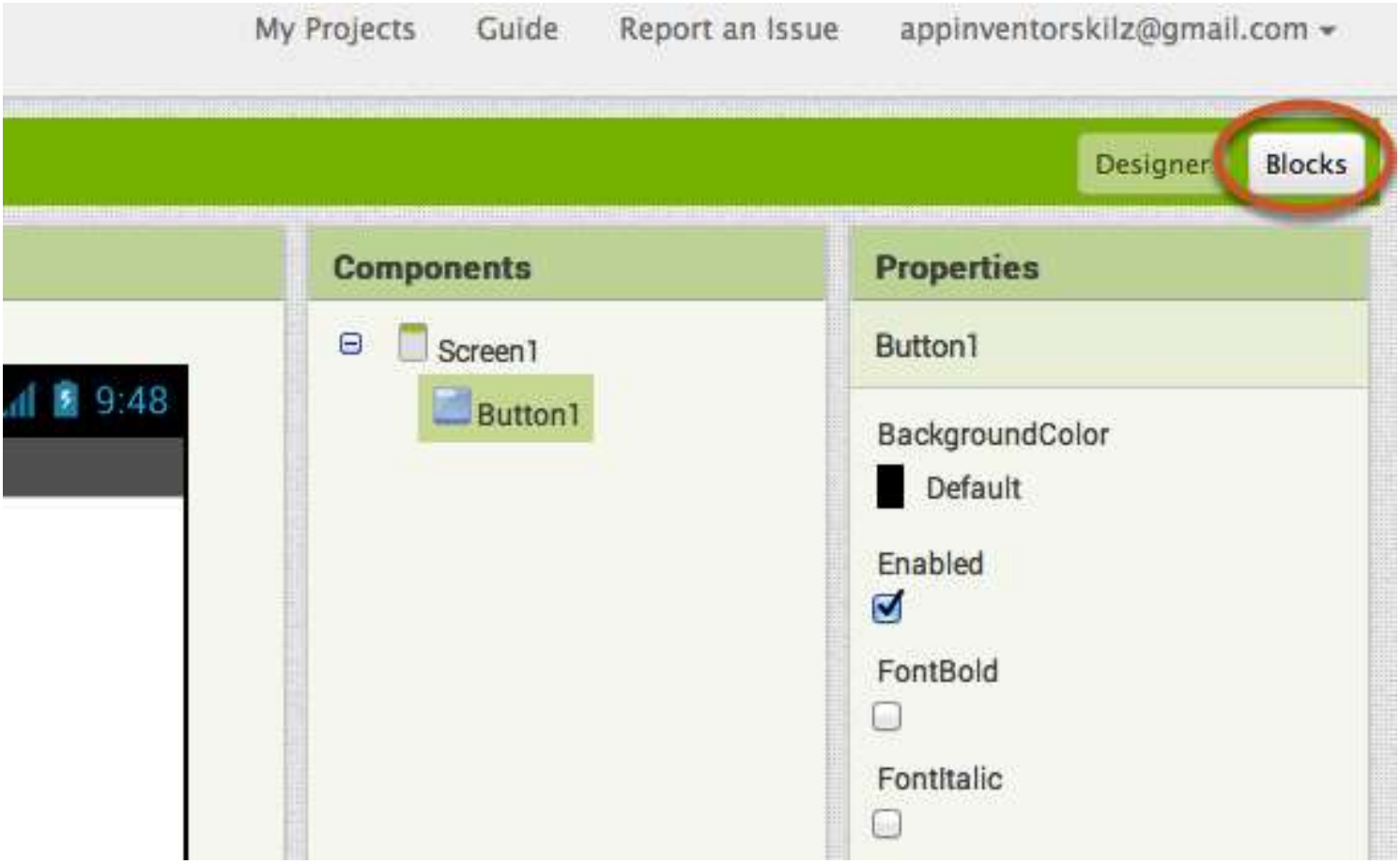
Add a Text-to-Speech component to your app

Go to the Media drawer and drag out a TextToSpeech component. Drop it onto the Viewer. Notice that it drops down under "Non-visible components" because it is not something that will show up on the app's user interface. It's more like a tool that is available to the app.



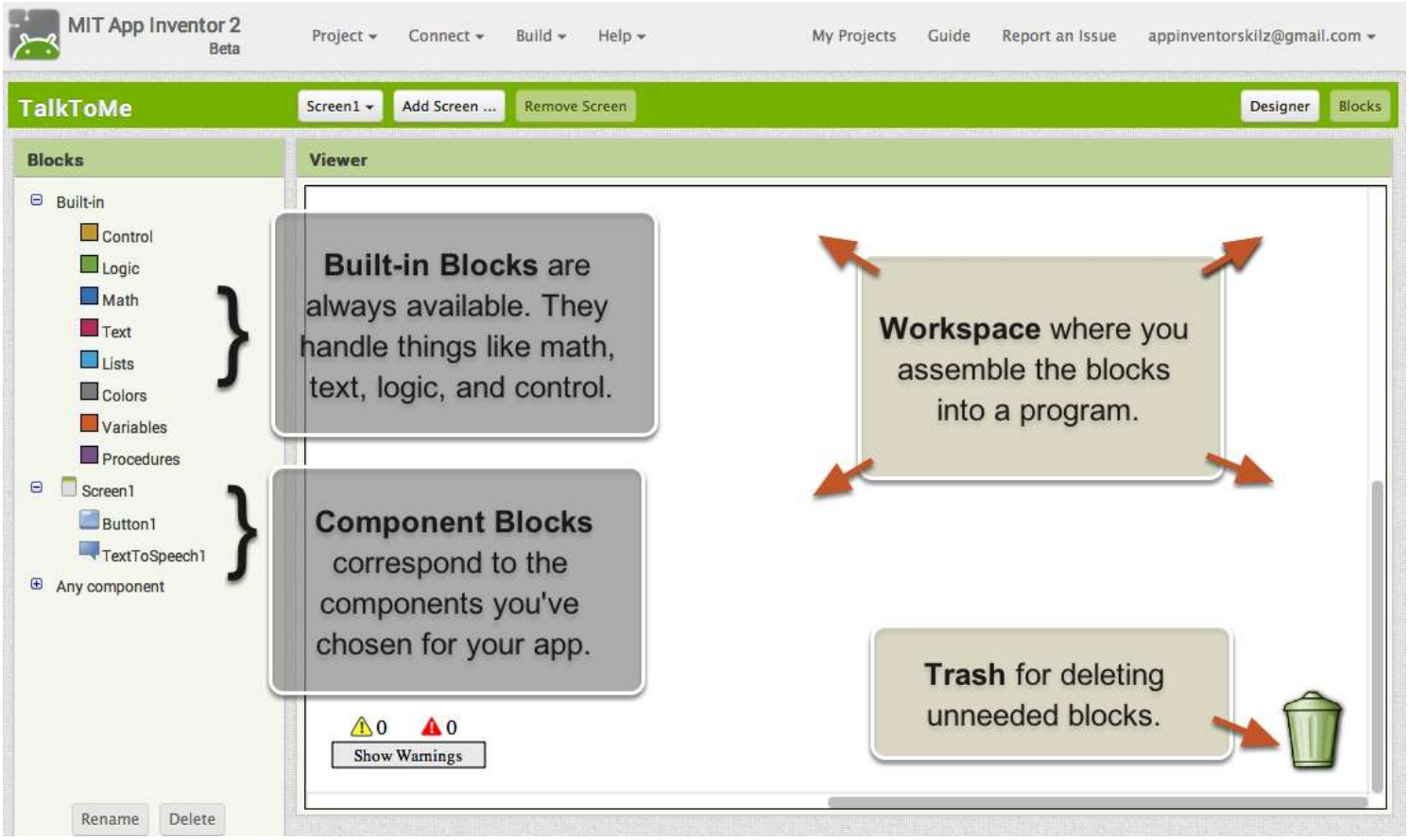
Switch over to the Blocks Editor

It's time to tell your app what to do! Click "Blocks" to move over to the Blocks Editor. Think of the Designer and Blocks buttons like tabs -- you use them to move back and forth between the two areas of App Inventor.



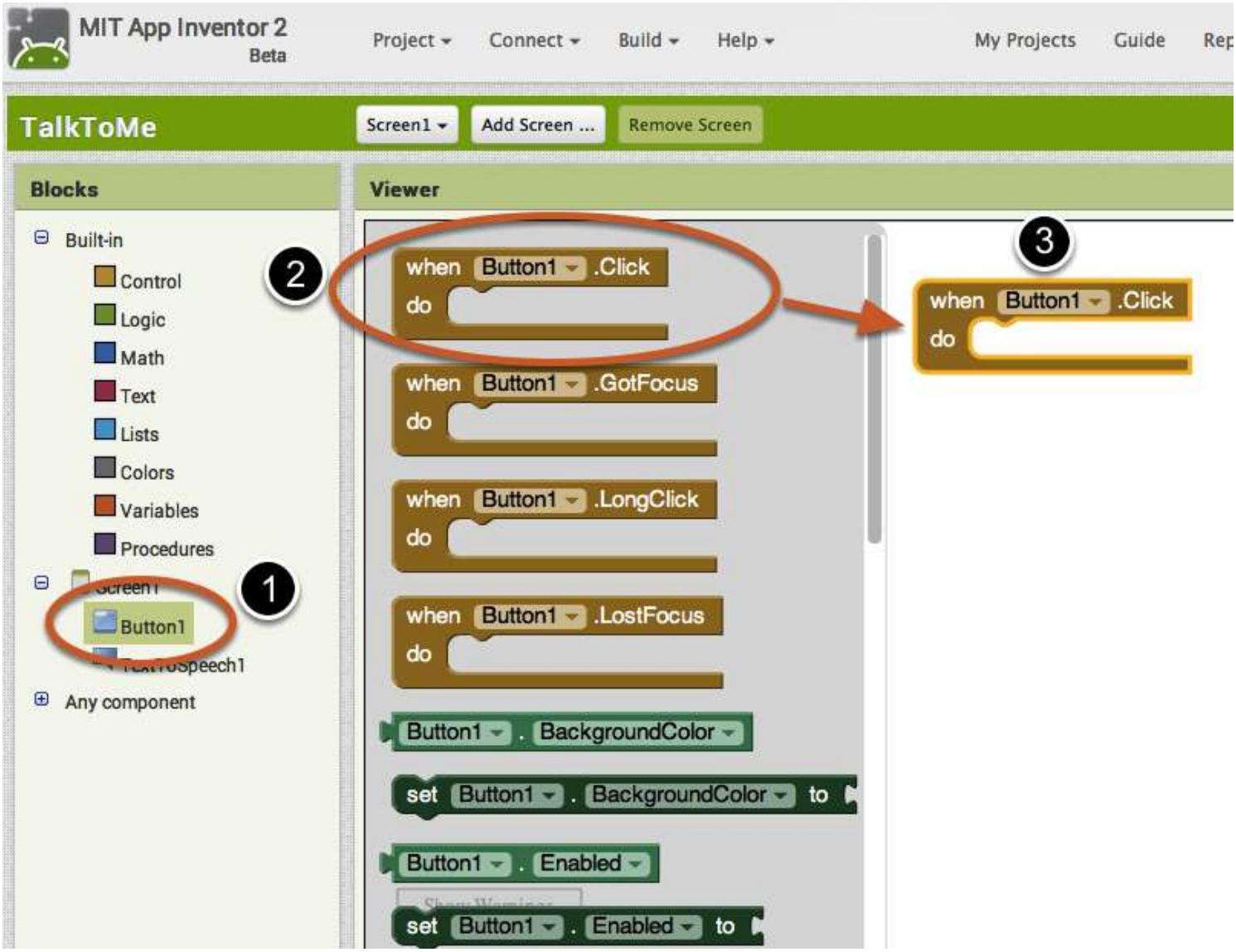
The Blocks Editor

The Blocks Editor is where you program the behavior of your app. There are Built-in blocks that handle things like math, logic, and text. Below that are the blocks that go with each of the components in your app. *In order to get the blocks for a certain component to show up in the Blocks Editor, you first have to add that component to your app through the Designer.*



Make a button click event

Click on the Button1 drawer. Click and hold the ***when Button1.Click do*** block. Drag it over to the workspace and drop it there. This is the block that will handle what happens when the button on your app is clicked. It is called an "Event Handler".



Program the TextToSpeech action

Click on the TextToSpeech drawer. Click and hold the ***call TextToSpeech1.Speak*** block. Drag it over to the workspace and drop it there. This is the block that will make the phone speak. Because it is inside the Button.Click, it will run when the button on your app is clicked.

Fill in the message socket on TextToSpeech.Speak Block

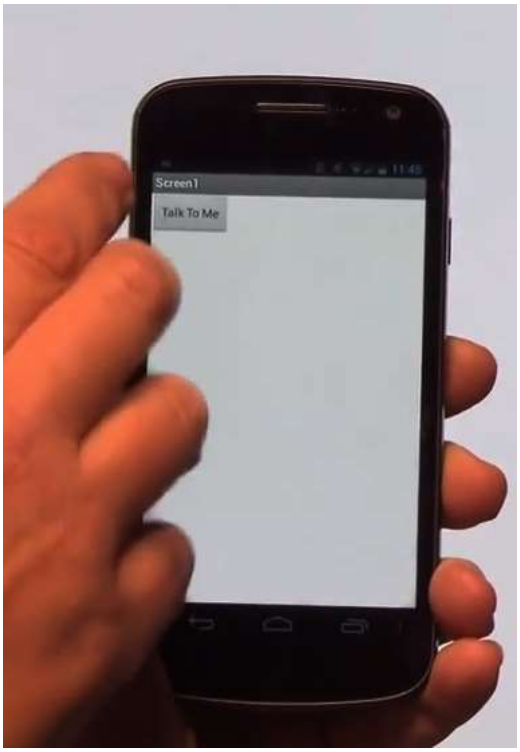
Almost done! Now you just need to tell the TextToSpeech.Speak block what to say. To do that, click on the Text drawer, drag out a ***text*** block and plug it into the socket labeled "message".

Specify what the app should say when the button is clicked

Click on the text block and type in "Congratulations! You've made your first app." (Feel free to use any phrase you like, this is just a suggestion.)

Now test it out!

Go to your connected device and click the button. Make sure your volume is up! You should hear the phone speak the phrase out loud. (This works even with the emulator.)



Great job!

Now move on to TalkToMe Part 2 to make the app respond to shaking and to let users put in whatever phrase they want.