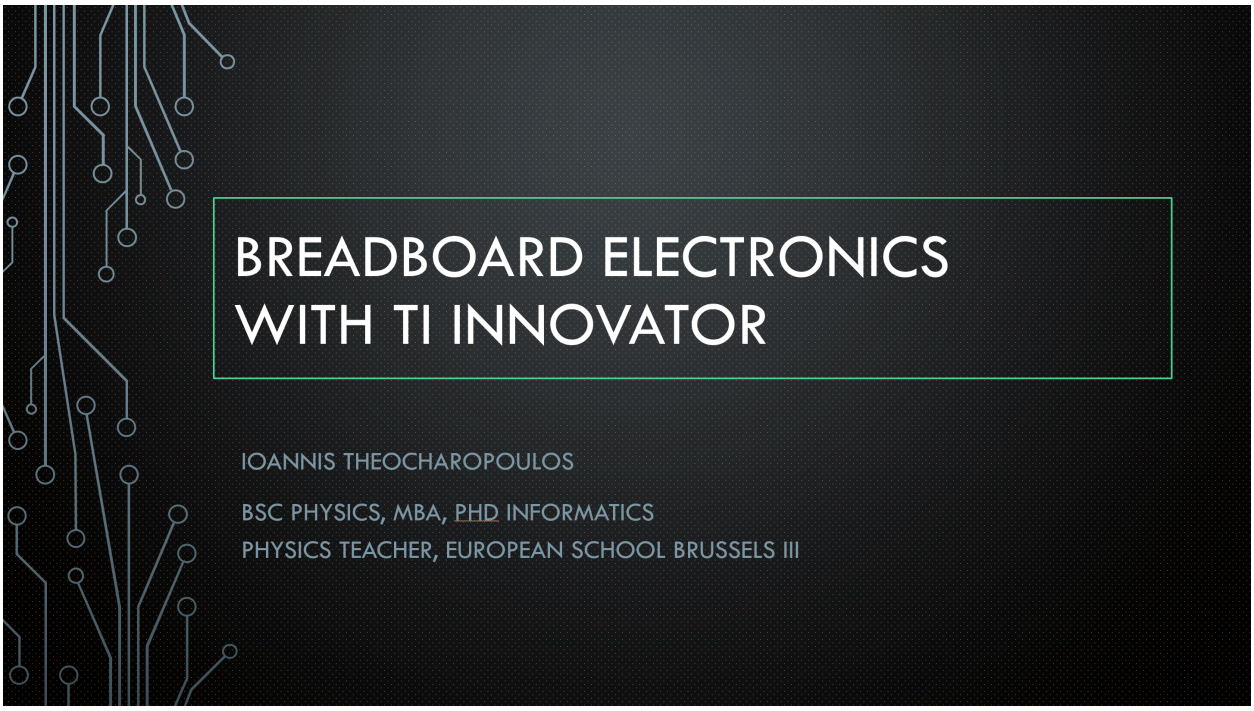
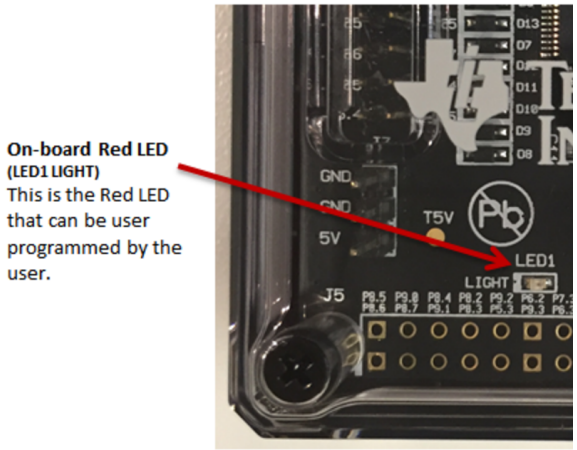


WORKSHOP ACTIVITY SET

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.1	Simple Blinker
SCOPE	Demonstrate simple control on the launchpad.

SCHEMATIC



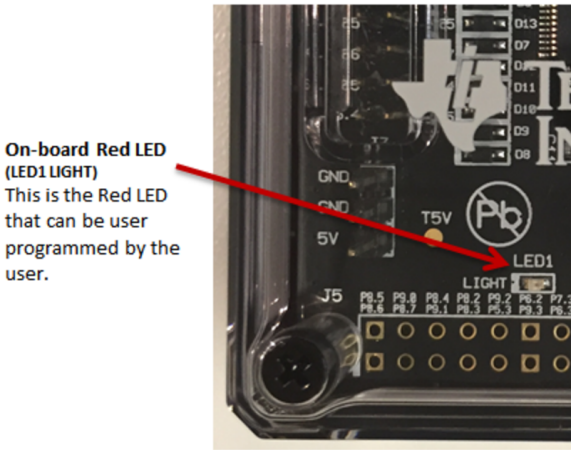
LISTING

```
Define simple_blinker()=
Prgm
  For n,1,10
    Send "SET LIGHT ON"
    Wait 1
    Send "SET LIGHT OFF"
    Wait 1
    EndFor
EndPrgm
```

SEND “XXXXXXXXXX” command communicates with the TI launcher pad.

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.2	Parametric Blinker
SCOPE	Demonstrate the way to use parameters that increase the level of control.

SCHEMATIC



LISTING

```
Define paramet_blinker(a,b)=
Prgm
For n,1,b
   Send "SET LIGHT ON"
   Wait a
   Send "SET LIGHT OFF"
   Wait a
   EndFor
EndPrgm
```

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.2	Buzzer
SCOPE	Create audible signals on the Ti Innovator Hub Speaker

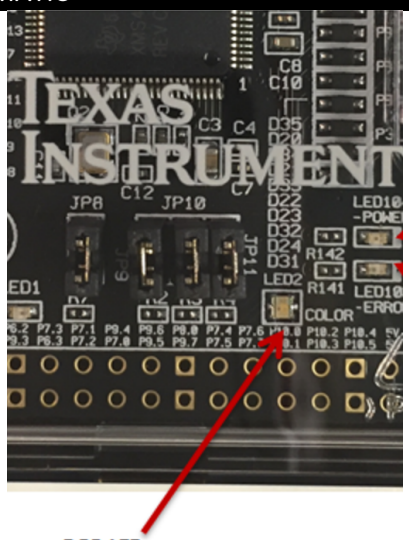
SCHEMATIC

LISTING

```
Define buzzer(a,b)=
Prgm
    Send "SET SOUND eval(a)   eval(b)"
EndPrgm
```

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.3	RGB Synthesizer
SCOPE	Create customized light signals on the launchpad board.

SCHEMATIC



RGB LED
(LED2 COLOR)
This is the Red-Green-Blue (RGB) LED that can be user programmed by the user.

LISTING

```
Define rgb_synth(r,g,b)=
Prgm

Send "SET COLOR.RED eval(r)"

Send "SET COLOR.GREEN eval(g)"

Send "SET COLOR.BLUE eval(b)"

EndPrgm
```

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.4	Light metering
SCOPE	Read an input from a sensor

SCHEMATIC

LISTING

```
Define sens_light()=
Prgm

 Local a

    For n,1,20

      Get "READ BRIGHTNESS",a

      Disp "Brightness level=",a

      Wait 0.5

     EndFor

EndPrgm
```

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.5	Sound synthesizer
SCOPE	Application of REQUEST command

SCHEMATIC

LISTING

```
Define sound_synth()=
Prgm
Request "Frequency?",f
Request "Duration?",t
Send "SET SOUND eval(f), TIME eval(t)"
EndPrgm
```

WORKSHOP 1 : COMMUNICATING WITH THE LAUNCHPAD	
ACTIVITY 1.6	APPLICATION
SCOPE	Create audible indicators depending on light level. User defined parameters

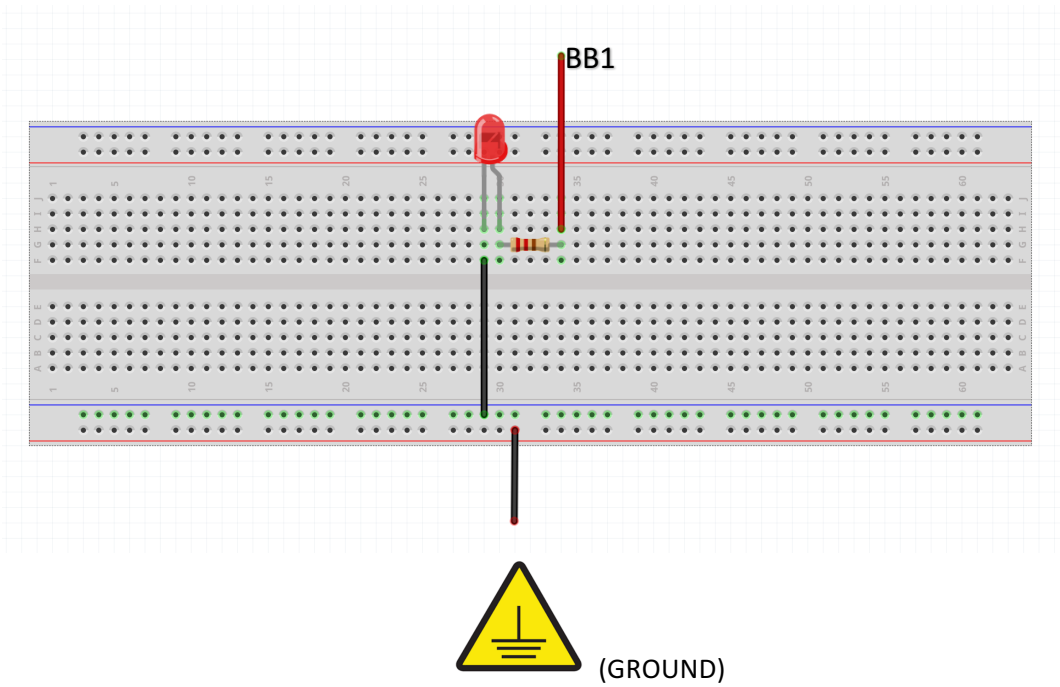
SCHEMATIC

LISTING

```
Define applic1()=
Prgm
Get "READ BRIGHTNESS",a
Disp a
Request "Threshold",t
      For n,1,200
          Get "READ BRIGHTNESS",a
           If a<t Then
              Send "SET SOUND 400,1"
              Wait 0.1
         EndIf :
   EndFor :
EndPrgm
```

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.1	LED attached to Digital Out
SCOPE	Demonstration of DIGITAL.OUT

SCHEMATIC

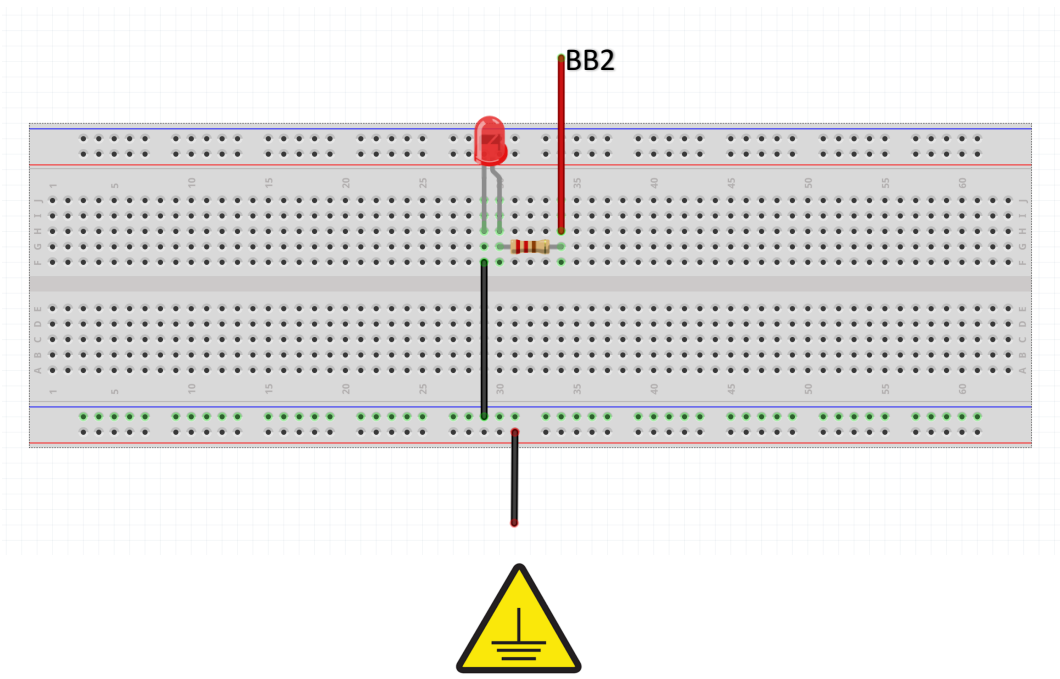


LISTING

Define bb_led_digital()=	
Prgm	
Send "BEGIN"	BEGIN initializes the Board (Works as a reset button)
Send "CONNECT DIGITAL.OUT 1 TO BB1"	CONNECT DIGITAL.OUT connects the DIGITAL.OUT object to a breadboard pin
Send "SET DIGITAL.OUT 1 1"	SET assigns a state to the DIGITAL.OUT DIGITAL.OUT has two states 1 (3,3V) and 0 (0V)
EndPrgm	

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.2	LED attached to Analog Out
SCOPE	Demonstration of ANALOG.OUT

SCHEMATIC

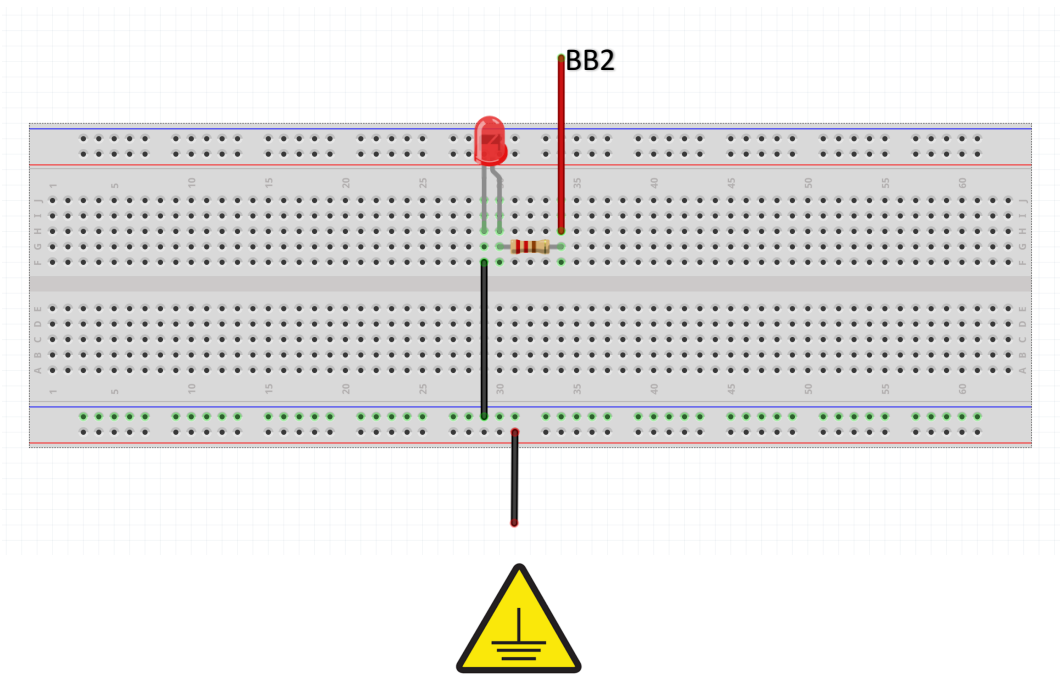


LISTING

Define bb_led_analog()= Prgm Send "BEGIN" Send "CONNECT ANALOG.OUT 1 TO BB2" Send "SET ANALOG.OUT 1 255" EndPrgm	BEGIN initializes the Board (Works as a reset CONNECT ANALOG.OUT connects the ANALOG.OUT object to a breadboard pin SET assigns a state to the ANALOG.OUT ANALOG.OUT has 255 states where 0V is OFF and 255 is 3,3V

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.3	Dimmer
SCOPE	External control of the intensity of the light of a led Use of eval()

SCHEMATIC



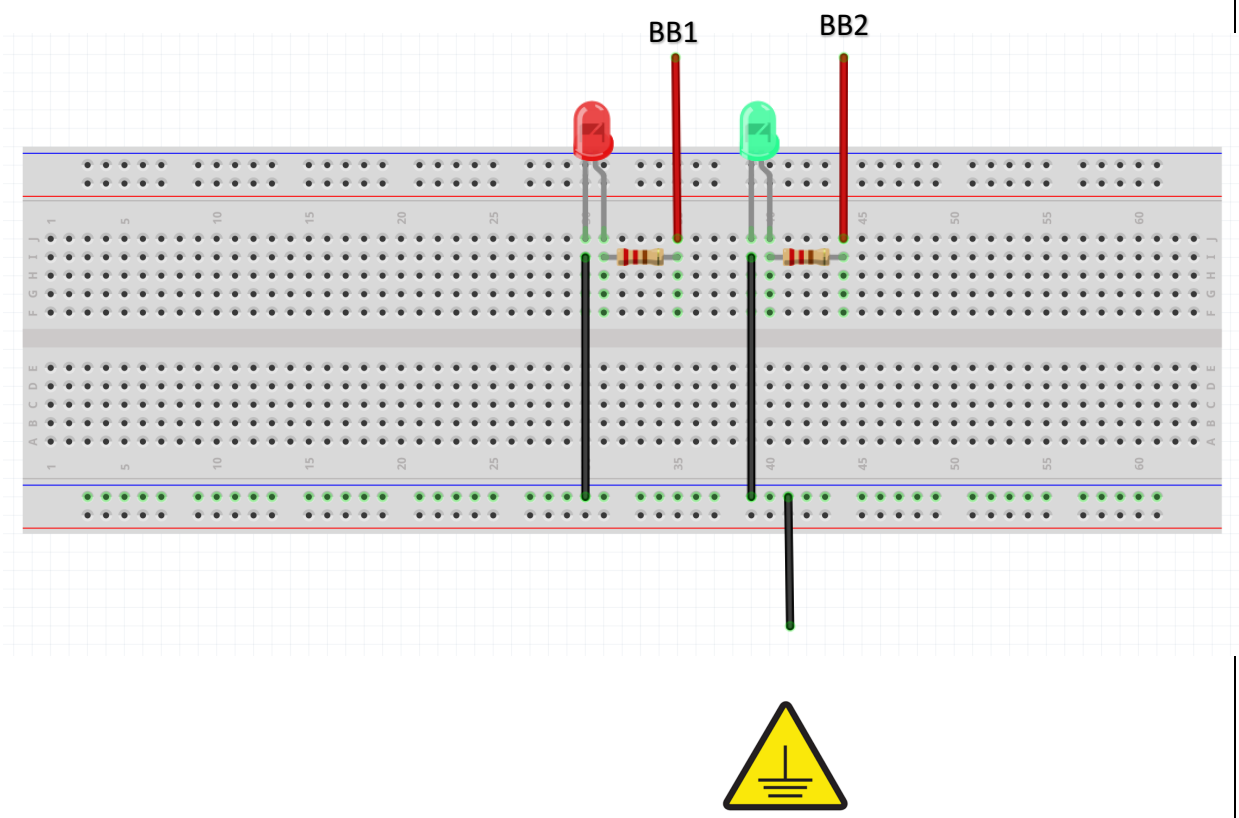
LISTING

```
Define dimmer()=
Prgm

Send "BEGIN"

Request "level 0-255",l

Send "CONNECT ANALOG.OUT 1 TO BB2"

Send "SET ANALOG.OUT 1 eval(l)"

EndPrgm
```

eval() transforms a string to its value

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.4	Alternating Blinking
SCOPE	Create a more complex behavior pattern

SCHEMATIC



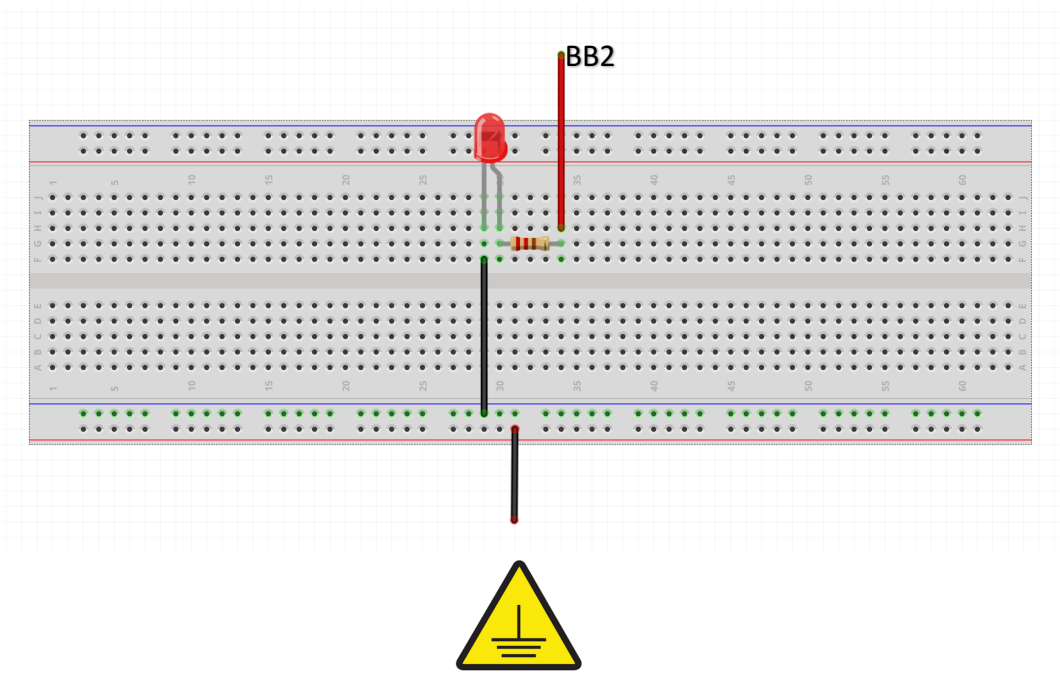
Note Resistors can be omitted

LISTING

```
Define alternating()=
Prgm
Send "BEGIN"
Send "CONNECT DIGITAL.OUT 1 TO BB1"
Send "CONNECT DIGITAL.OUT 2 TO BB2"
For n,1,100
   Send "SET DIGITAL.OUT 1 1"
   Send "SET DIGITAL.OUT 1 0"
   Send "SET DIGITAL.OUT 2 1"
   Send "SET DIGITAL.OUT 2 0"
EndFor
EndPrgm
```

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.5	Square wave generator
SCOPE	External control of the intensity of the light of a led Use of eval()

SCHEMATIC

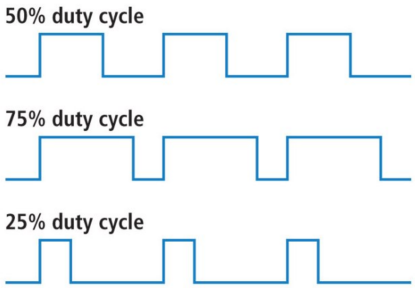


Resistors are 100Ω

LISTING

```
Define square_wave()=
Prgm
Request "duty 0-99",d
Request "freq",f
Send "CONNECT SQUAREWAVE 1 TO BB1"
Send "SET SQUAREWAVE 1 eval(f)  eval(d)  TIME 20"
EndPrgm
```

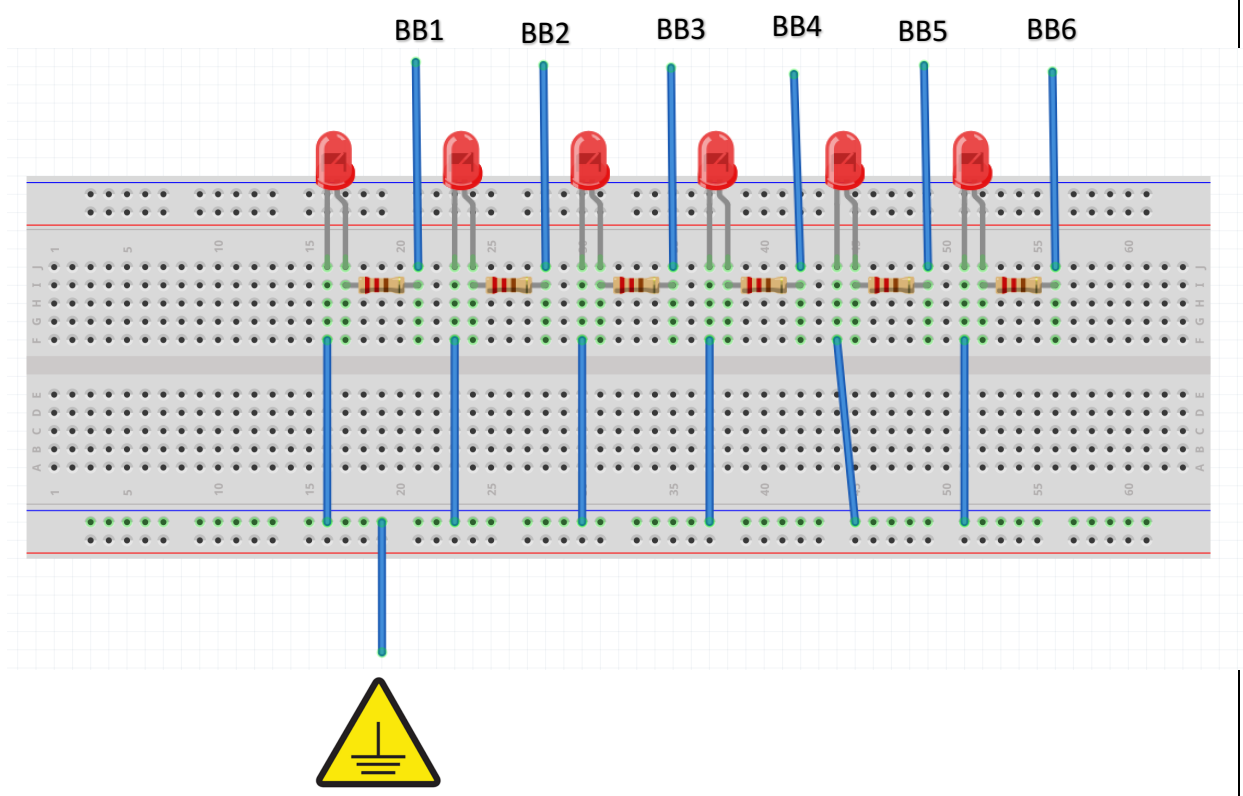
Duty cycle is interpreted as follows



eval() transforms a string to its value

WORKSHOP 2: THE BREADPOARD PINS	
ACTIVITY 2.6	APPLICATION Electronic dice
SCOPE	A game that demonstrates the capabilities of the Launchpad

SCHEMATIC



Resistors are 100Ω

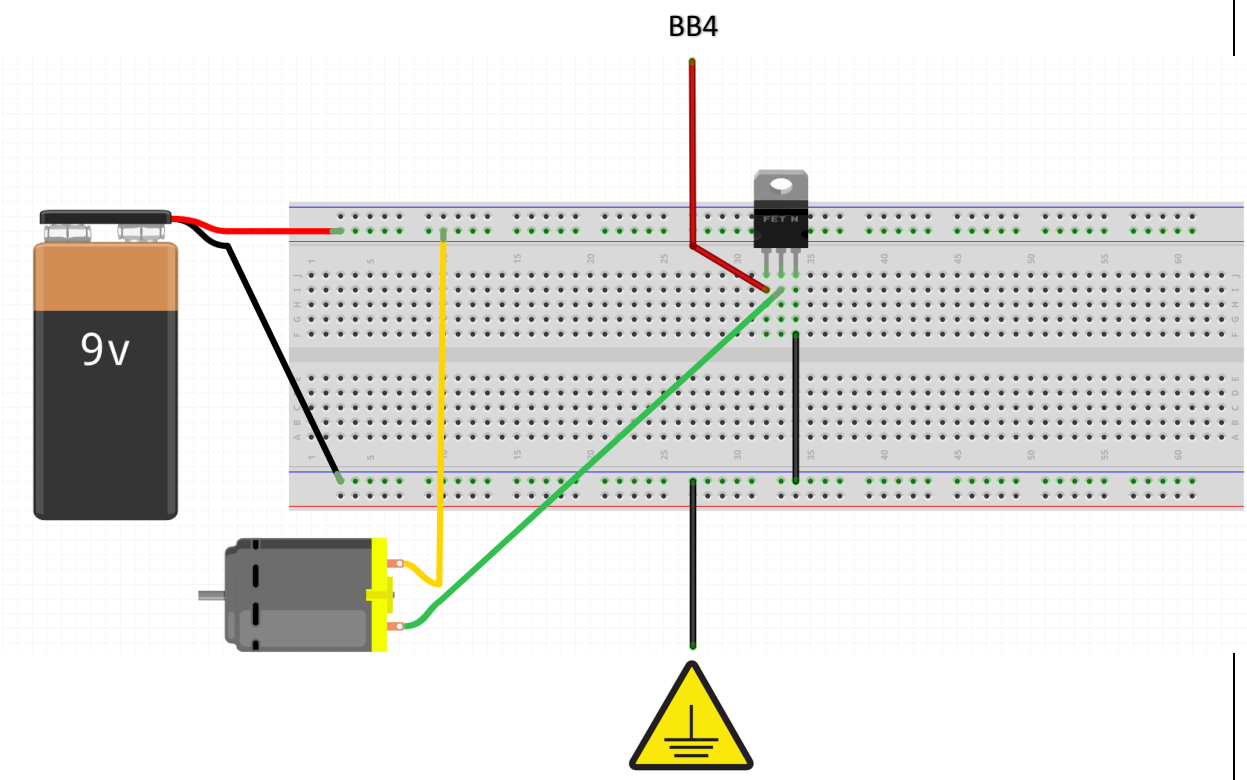
LISTING

```
Define game_analog()=
Prgm
Request "Give a number 0-6",k
Send "BEGIN" :Send "CONNECT ANALOG.OUT 1 TO BB1"
 Send "CONNECT ANALOG.OUT 2 TO BB2"
Send "CONNECT ANALOG.OUT 3 TO BB3"
Send "CONNECT ANALOG.OUT 4 TO BB4"
Send "CONNECT ANALOG.OUT 5 TO BB5"
Send "CONNECT ANALOG.OUT 6 TO BB6"
 For n,1,200
    randInt(1,6)→a :
    Send "SET ANALOG.OUT eval(a) 255"
    Send "SET ANALOG.OUT eval(a) 0"
      If n=200 Then :
      Send "SET ANALOG.OUT eval(a) 255" :
     Wait 1
```

```
    Disp a
      If k=a Then
          Disp "You win!!!"
      Else
           Disp "You loose..."
      EndIf
    EndIf
 EndFor
EndPrgm
```

WORKSHOP 3: BREADBOARD SCENARIOS	
ACTIVITY 3.1	N- Chanel MOSFET Transistor
SCOPE	Use the MOSFET transistor to modulate the voltage output to a motor

SCHEMATIC

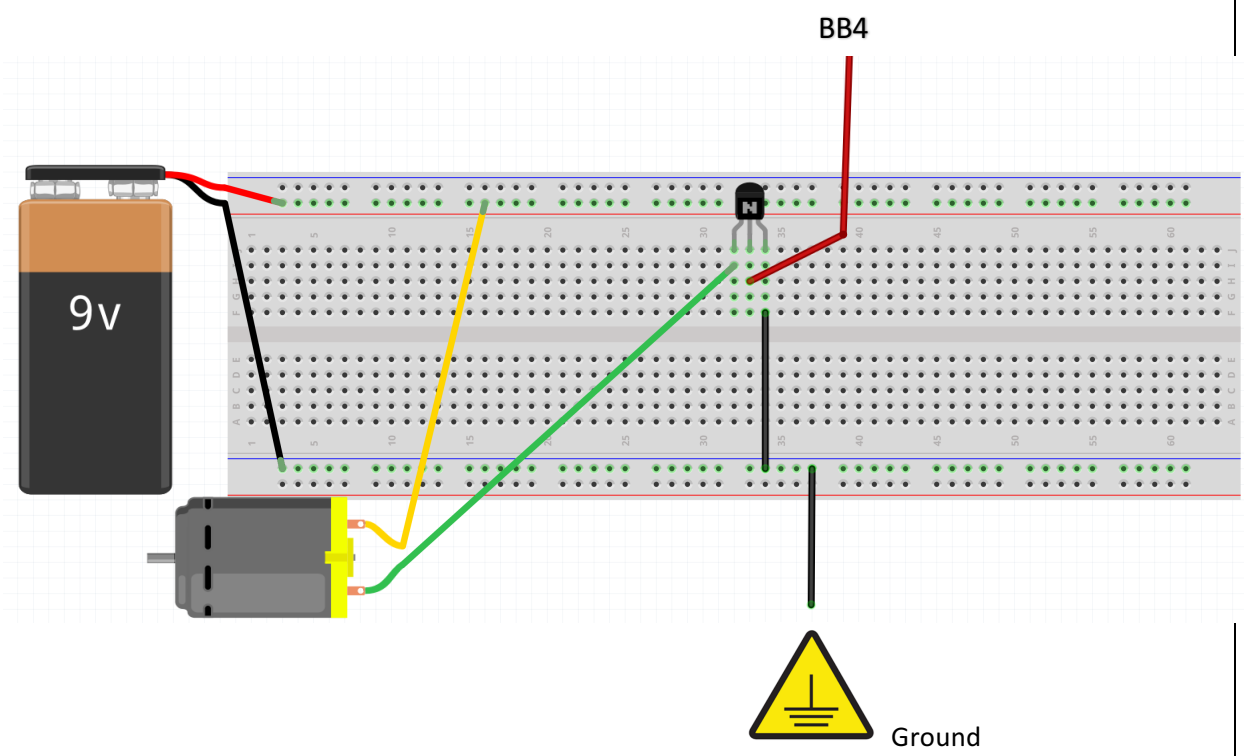


LISTING

Define dc1()= Prgm Send "CONNECT ANALOG.OUT 1 TO BB4" Request "ENTER SPEED 1 TO 255",s Send "SET ANALOG.OUT 1 TO eval(s)" EndPrgm	We actually request for a voltage level. The voltage attached to the GATE of the MOSFET is calculated through the converter $\frac{s}{255} \cdot 3.3V$

WORKSHOP 3: BREADBOARD SCENARIOS	
ACTIVITY 3.2	NPN Bipolar Junction Transistor
SCOPE	Use BJT Transistor to demonstrate and explore the control of current in a motor

SCHEMATIC



LISTING

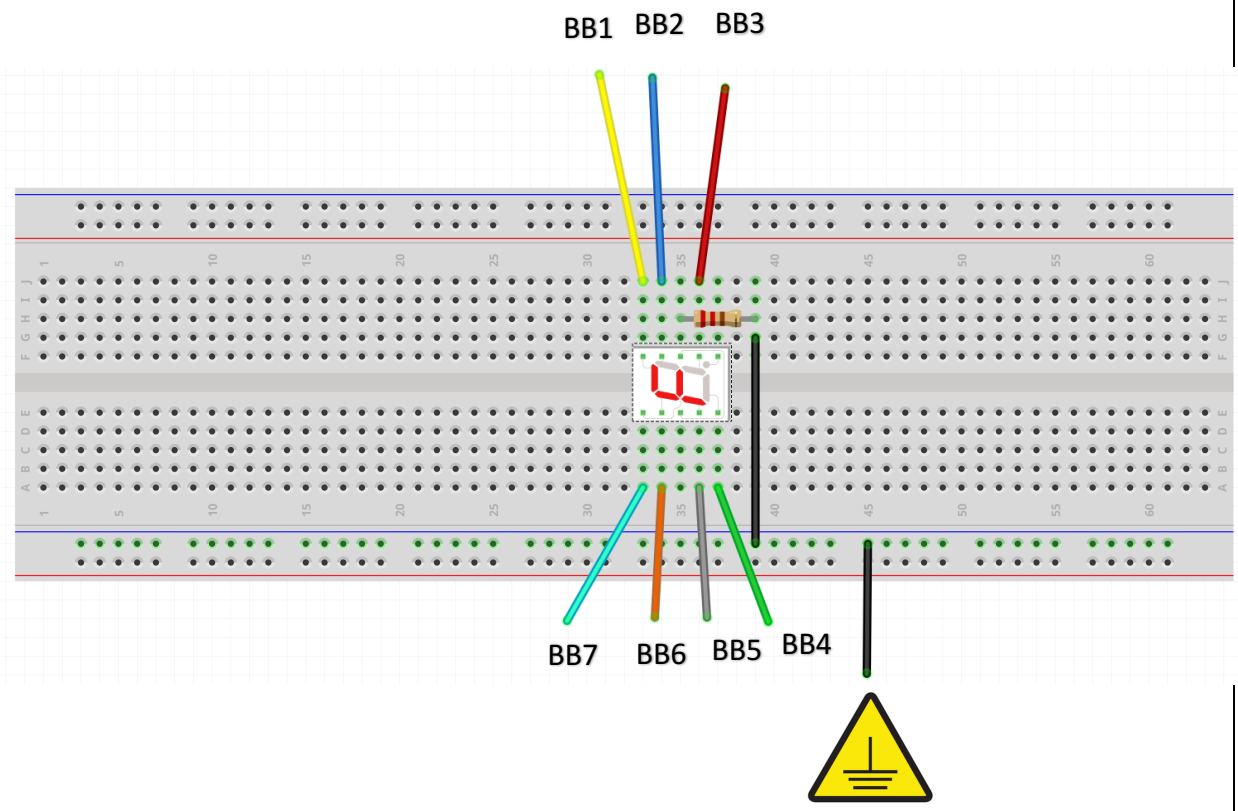
```
Define dc1()=
Prgm
Send "CONNECT ANALOG.OUT 1 TO BB4"
Request "ENTER SPEED",s
Send "SET ANALOG.OUT 1 TO eval(s)"
EndPrgm
```

We actually request for a voltage level. The voltage attached to the BASE of the BJT is calculated through the converter

$$\frac{s}{255} \cdot 3.3V$$

WORKSHOP 3: BREADBOARD SCENARIOS	
ACTIVITY 3.2	7-segments display (This demonstration requires a common cathode digital display)
SCOPE	Control the output of a common cathode display

SCHEMATIC



Resistors are 100Ω

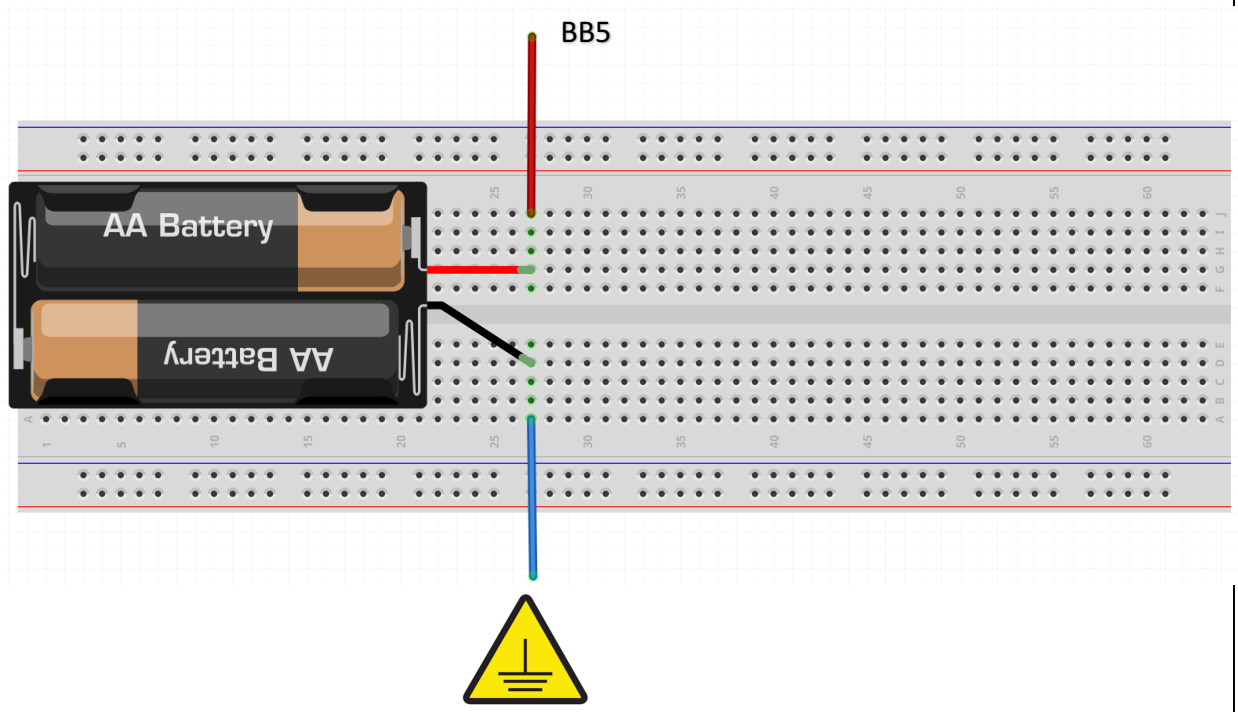
LISTING

```
Define digits()=
Prgm

Send "CONNECT ANALOG.OUT 1 TO BB1 "
Send "CONNECT  ANALOG.OUT 2 TO BB2 "
Send "CONNECT ANALOG.OUT 3 TO BB3"
Send "CONNECT ANALOG.OUT 4 TO BB4 "
Send "CONNECT ANALOG.OUT 5 TO BB5 "
Send "CONNECT ANALOG.OUT 6 TO BB6 "
Send "CONNECT ANALOG.OUT 7 TO BB7 "

Request "Give a number 1 to 9 ",n

If n=7 Then
```

```
Send "SET ANALOG.OUT 1  255"
Send "SET ANALOG.OUT 2  255"
Send "SET ANALOG.OUT 3  255"
Send "SET ANALOG.OUT 7  255"

ElseIf n=3 Then

Send "SET ANALOG.OUT 1  255"
Send "SET ANALOG.OUT 2  255"
Send "SET ANALOG.OUT 3  255"
Send "SET ANALOG.OUT 4  255"
Send "SET ANALOG.OUT 6  255"

ElseIf n=2 Then

Send "SET ANALOG.OUT 1  255"
Send "SET ANALOG.OUT 2  255"
Send "SET ANALOG.OUT 6  255"
Send "SET ANALOG.OUT 4  255"
Send "SET ANALOG.OUT 5  255"
EndIf
EndPrgm
```

WORKSHOP 3: BREADBOARD SCENARIOS	
ACTIVITY 3.3	Voltage meter
SCOPE	Measure a voltage 0 - 3.3V

SCHEMATIC

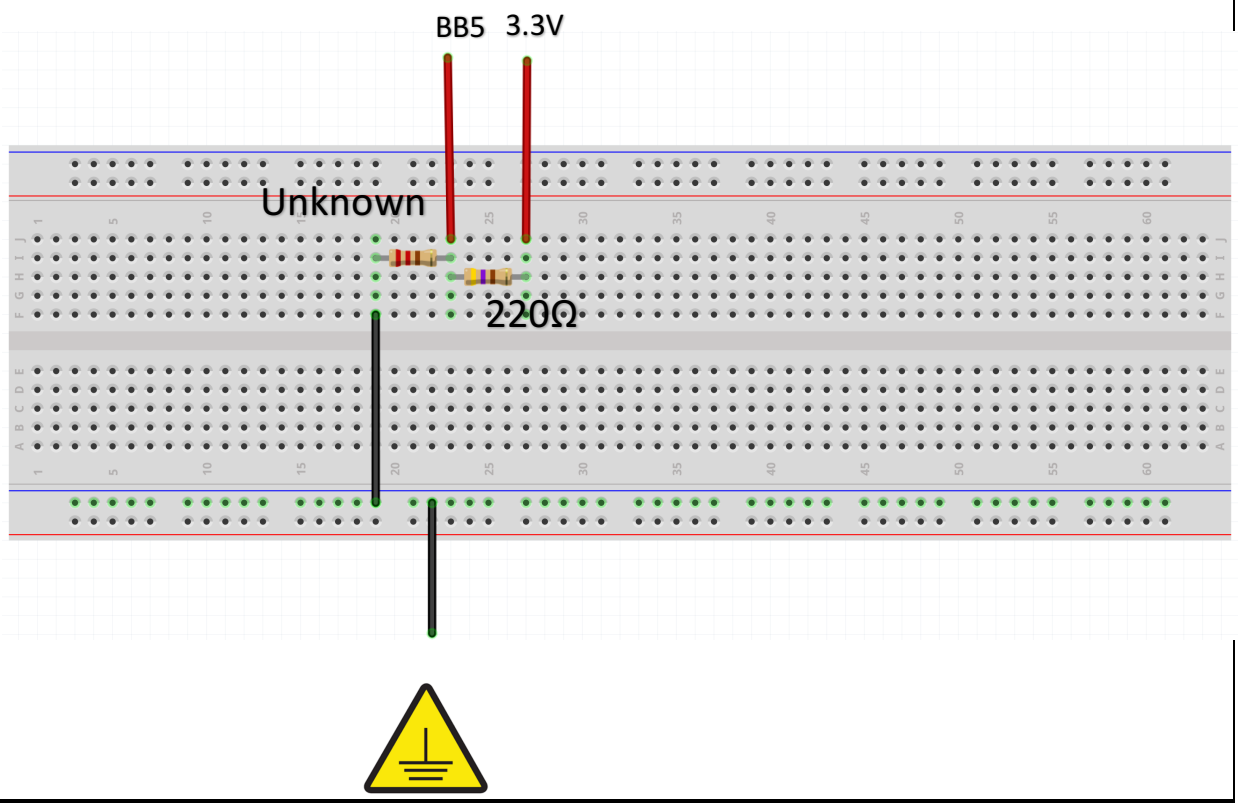


LISTING

Define voltage_meter()= Prgm Send "CONNECT ANALOG.IN 1 TO BB 5" Send "READ ANALOG.IN 1" Get a $b:=\frac{a}{2^{14}}\cdot 3.3$ Disp b EndPrgm	This is the conversion rule from the analog output to voltage.

WORKSHOP 3: BREADBOARD SCENARIOS	
ACTIVITY 3.3	Measure an unknown resistor
SCOPE	Measure the value of an unknown resistor

SCHEMATIC



LISTING

Define resist()= Prgm Send "CONNECT ANALOG.IN 1 TO BB 5" Send "READ ANALOG.IN 1" Get a b:=((a)/(2^(14)))*3.3 Disp b r:=((220*b)/(3.3-b)) Disp r EndPrgm	This is the voltage divider rule The known resistor is 220Ω