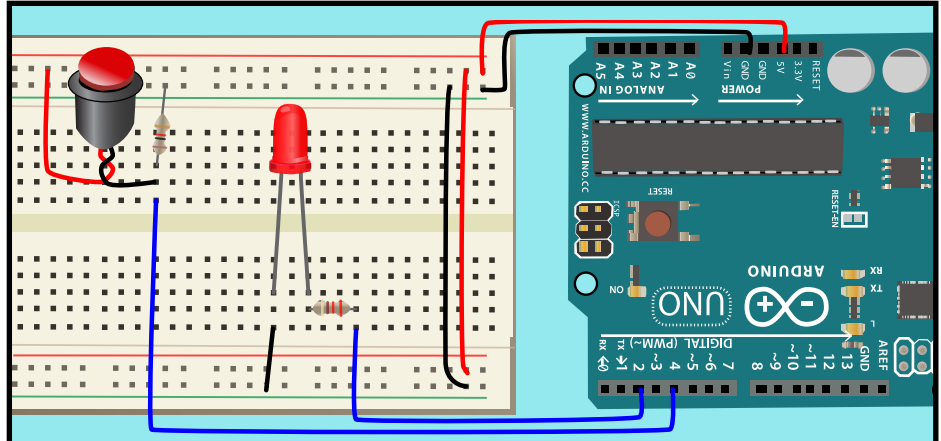NEXT WE WILL ADD A SWITCH, A DIGITAL INPUT, SO WE CAN TURN THE LED OFF AND ON.

CONNECT ONE END OF A MOMENTARY SWITCH TO PIN 4 ON THE ARDUINO, WITH A 10K RESISTOR CONNECTED TO GROUND ATTACHED TO THE SAME END. ATTACH THE OTHER END TO POWER. WE WILL LEAVE THE LED ATTACHED TO THE SAME PIN.

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
}

void loop() {
  if(digitalRead(4)){
    digitalWrite(2, HIGH);
  }else{
    digitalWrite(2, LOW);
  }
}
```

NEXT WE'LL WRITE THE CODE. IN SETUP, WE DECLARE PIN 2 AN OUTPUT AND PIN 4 AN INPUT. IN LOOP, WE USE AN IF STATEMENT, IF WE READ PIN 4 AS HIGH, WE SET THE LED PIN TO HIGH, OTHERWISE WE SET THE LED PIN TO LOW, TURNING IT OFF.
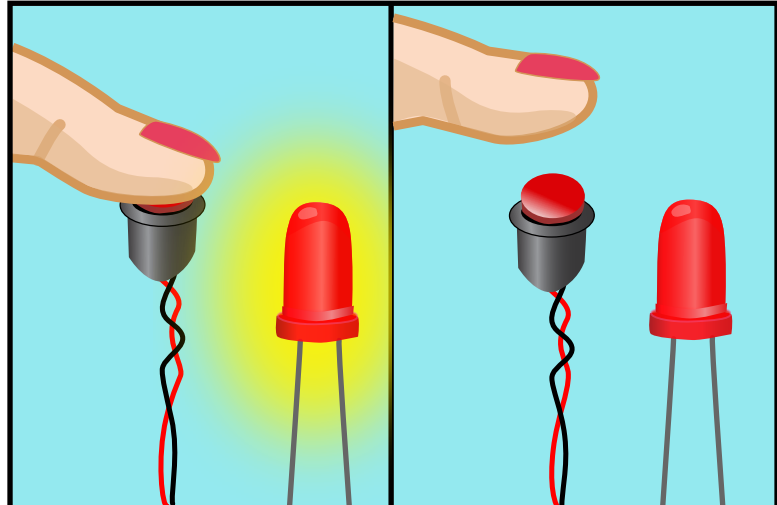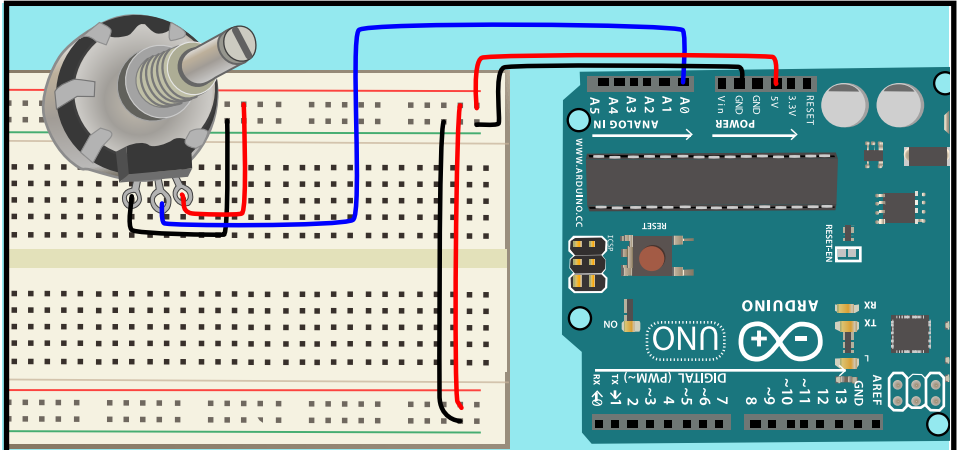
THE LED LIGHTS WHEN THE SWITCH IS HELD DOWN.

Panel 1 (speech bubble):

A *POTENTIOMETER*, OR *POT*, IS A *VARIABLE RESISTOR*. THE AMOUNT OF RESISTANCE CHANGES AS IT IS TURNED, INCREASING OR DECREASING DEPENDING ON WHICH DIRECTION IT IS TURNED.

Panel 1 (caption):

NOW WE WILL SET UP AN ANALOG INPUT. WE'LL USE A POTENTIOMETER.

Panel 2 (caption):

ATTACH THE MIDDLE PIN ON THE POTENTIOMETER TO ANALOG PIN A0. ATTACH ONE END OF THE POT TO POWER, THE OTHER TO GROUND.

Panel 3 (code):

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
}
```

Panel 3 (caption):

FIRST WE WILL LOOK AT THE RANGE OF VALUES WE GET BY TURNING THE POT USING THE *SERIAL MONITOR*. IN OUR CODE, WE INITIALIZE THE SERIAL OBJECT IN SETUP, SETTING A BAUD RATE OF 9600. IN LOOP, WE READ THE VALUE FROM ANALOG PIN A0 AND PRINT IT TO THE SERIAL OBJECT USING THE *PRINTLN* FUNCTION.
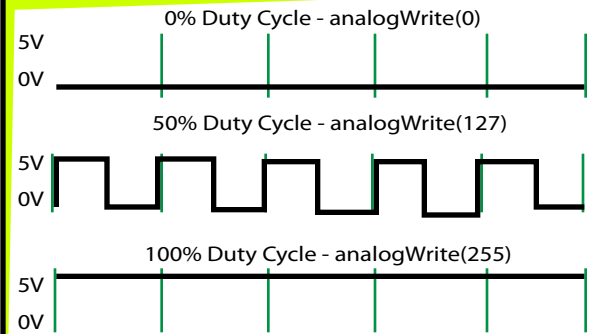
Panel 4:

Serial Monitor

/dev/tty.usbmodemfa141

128
129
130
131
131
132
133
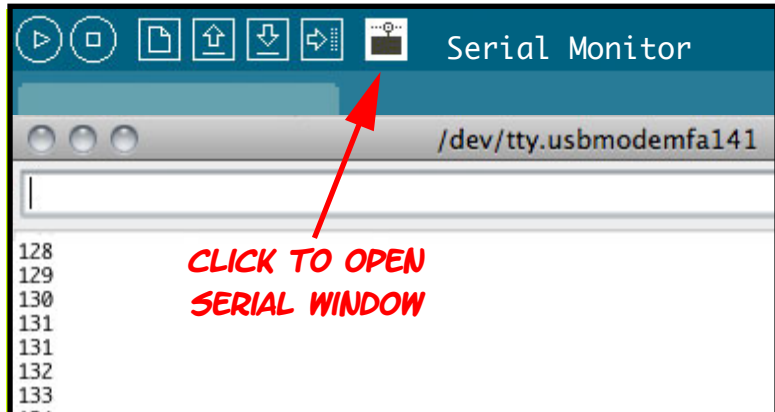
CLICK TO OPEN SERIAL WINDOW

Panel 4 (caption):

AFTER YOU HAVE UPLOADED THE SCRIPT TO THE ARDUINO, CLICK THE *SERIAL MONITOR* BUTTON IN ORDER TO SEE THE VALUES AS YOU TURN THE POT. A WINDOW WILL OPEN, AND YOU WILL SEE VALUES RANGING FROM 0 TO 1024 AS THE POT IS TURNED.
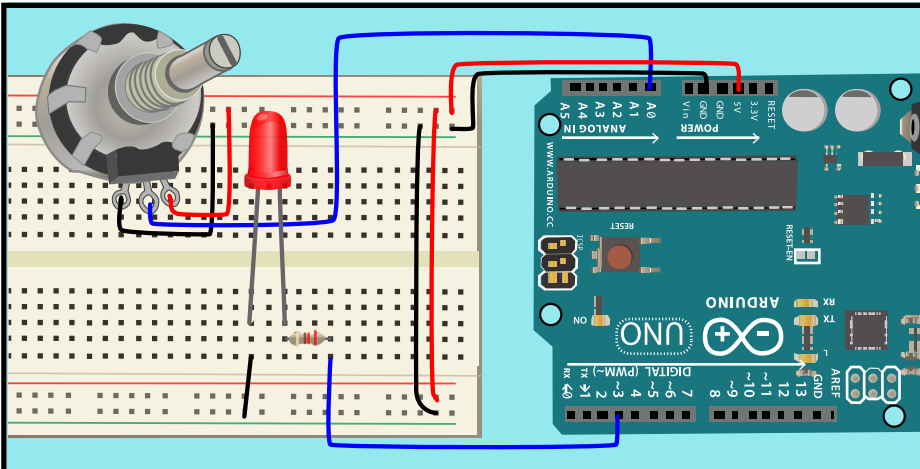
LET'S USE THE CHANGING VALUES WE RECEIVE FROM THE POT AS A DIMMER TO LIGHT AN LED. PUT THE LED BACK INTO THE BOARD, ATTACHED TO THE ARDUINO AT PIN 3.


0% Duty Cycle - analogWrite(0)

50% Duty Cycle - analogWrite(127)

100% Duty Cycle - analogWrite(255)

WE'LL USE *PULSE WIDTH MODULATION* (PWM). THIS IS A METHOD OF SIMULATING AN ANALOG VALUE BY MANIPULATING THE VOLTAGE, TURNING IT ON AND OFF AT DIFFERENT RATES, OR *DUTY CYCLES*. YOU CAN USE PWM WITH PINS 3, 5, 6, 9, 10, AND 11.

```
int sensorValue = 0;

void setup() {
  pinMode(3,OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

FIRST WE CREATE A *VARIABLE* TO STORE THE VALUE OF THE POT. IN SETUP WE MAKE PIN 3 AN OUTPUT. IN LOOP, WE STORE THE VALUE WE HAVE READ FROM PIN A0 IN OUR VARIABLE. THEN WE WRITE THE VALUE TO PIN 3, OUR LED PIN. WE HAVE TO DIVIDE THE VARIABLE BY 4, SO WE WILL HAVE A RANGE OF VALUES FROM 0 TO 255, OR A *BYTE*.
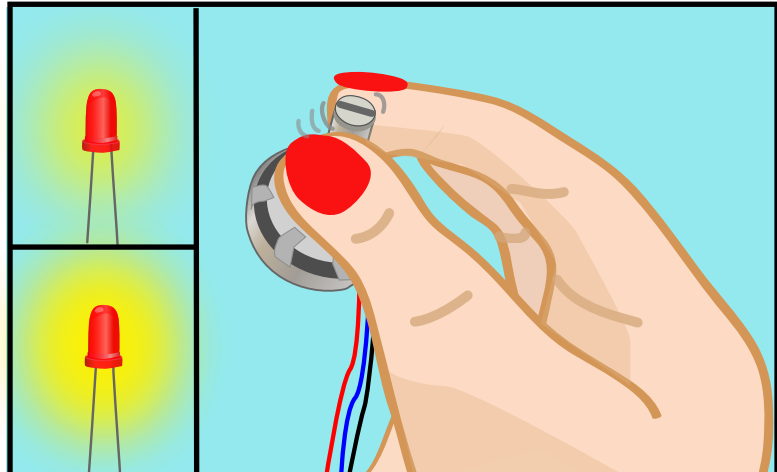
THE BRIGHTNESS OF THE LED CHANGES, RANGING FROM COMPLETELY OFF TO VERY BRIGHT AS YOU TURN THE POT.