ARDUINO PROGRAMMING

OUTREACH PROGRAM LESSON PLAN II

WAAW foundation

Working To Advance STEM Education for African Girls

WAAW Foundation is an international non-profit organization dedicated to bringing hands-on STEM education to girls all over Africa.

Our Mission: To increase the pipeline of African women in Science, Technology, Engineering and Math (STEM) disciplines and to ensure this talent is engaged in African innovation.

Our Vision: To eradicate poverty in Africa through female education and science and technology innovation.

WAAW foundation

Working To Advance STEM Education for African Girls

P.O. Box 1691 Wylie,

Texas 75098

1-972-763-5924

www.waawfoundation.org

"LIKE" us on facebook— www.facebook.com/waawfoundation
"FOLLOW" us on twitter— www.twitter.com/waaw_foundation
"SUBSCRIBE" to our newsletter— http://eepurl.com/ihwpU

Arduino Programming

CLASS DESCRIPTION:

In this class, students will participate in activities of doing arduino projects as they explore the basics of the previous knowledge. They will work in a maximum of two groups for effective involvement.

TOTAL CLASS TIME: **1 hour 30 Minutes**

CLASS OUTCOMES:

- Students will learn how to write Arduino codes
- They will be able to build circuits using the required components.
- Students would be able to carry out Arduino projects that solves Real-world problems.

LIST OF MATERIALS:

- Computer systems (with the Arduino software already installed)
- Arduino board
- Breadboard
- Raspberry Pi (when using a lapdock)
- USB cable
- Components (Jumper wires, resistors, LEDs, potentiometer)

PRE-CLASS PREPARATION AND SET-UP:

- In preparation to teach this class, you should carry out the activities yourself so that you can get familiar with what the projects and check for possible problems that may rise during the class.
- Ensure you have installed the software (can be downloaded from https://www.arduino.cc/).
- Prior to the introduction, review and revise the previous lesson (From lesson plan 1) so as to enhance proper understanding and follow-up of the class. Then precede to the Arduino projects.

INTRODUCTION (5 minutes)

Ask the students – What do you know about Programming?

Let the students list the programming languages they know.

Programming is the process of instructing a computer on how to carry out a task. Programming can either be for Software or Hardware.

There various Software Programming Languages are:

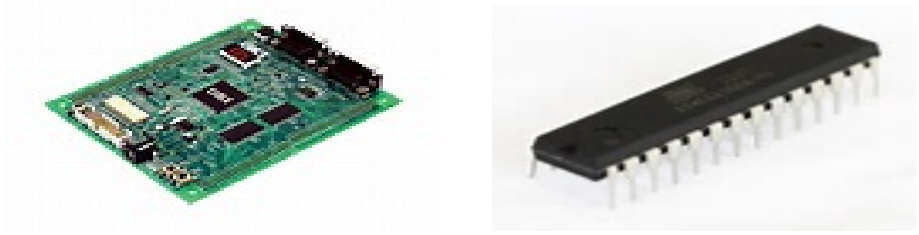
- C
- C#
- C++
- HTML (HyperTextMarkup Language)
- CSS (Cascading Style Sheet)
- PHP (HpertextPreprocessor)
- MYSQL (Structured Query Language)
- JavaScript
- Java
- Python … and so on.

Note that: they should have learned about Arduino Programming from the previous lesson plan for better understanding

Some of these languages above can also be used to program Hardware but for controlling micro-controllers, one easy method is using the Arduino program.

Show pictures to ensure they create a visual-picture in their minds for better understanding.

This is what we are going to explore today.
Who has heard or seen a Micro-Controller before?



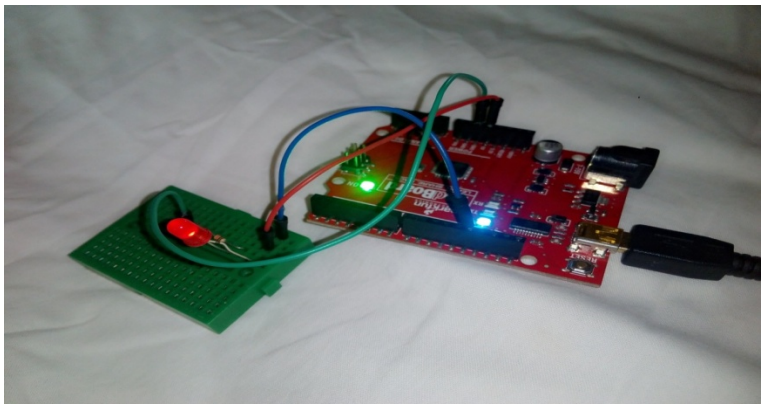
Sample Pictures of Micro-controllers

A micro-controller is a small control device that incorporates a microprocessor and is capable of carrying out a complex series of actions automatically when programmed with a computer.

PROJECT 1 (15 minutes):

Blinking an LED

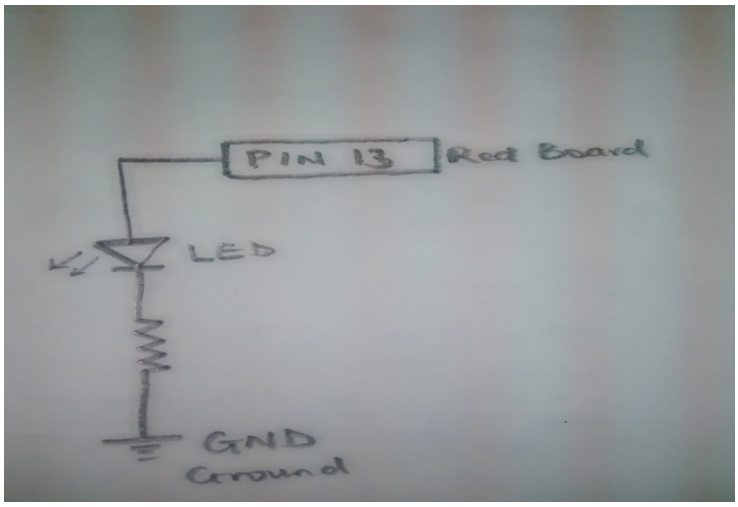
This project entails turning on and off an LED at one second interval repeatedly as shown in the comment (indicated /* --- */) in the Arduino codes below.
The circuit diagram for this project is shown below to help with connection.



ARDUINO CODE

```
/*Blink

Turns on an LED on for one second,

then off for one second, repeatedly.

*/

void            setup() {          .

//                initialize the digital pin as an output

//                Pin 13 has an LED connected on most Arduino boards:

} pinMode(13, OUTPUT);

void            loop() {  // set the LED on

digitalWrite(13, HIGH);

delay(1000); // wait for a second

digitalWrite(13, LOW);          // set the LED off

} delay(1000);// wait for a second
```

With the connection and code writing all done, proceed with asking the students to compile and verify the codes on the Arduino screen as the USB cord is connected to the computer



Circuit diagram

Procedure:

- Write the codes as shown above.
- Connect all components correctly.
- Connect the Arduino Board with the Computer using the USB cable.
- Verify the written codes and upload to the board.

What did you observe?

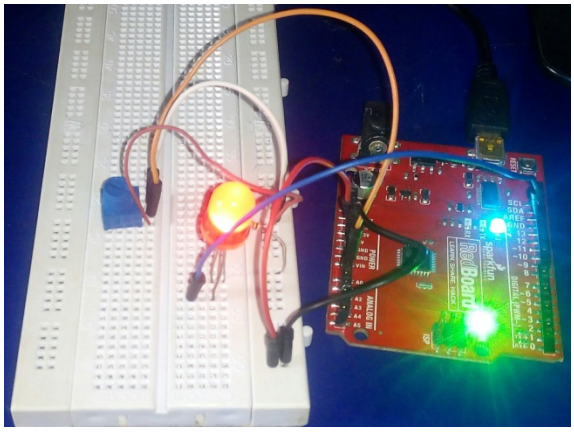
- State your observations.
- Let the students add codes that will make the LED blink faster or slower.
- Let them keep exploring different speed times.

Note: As seen in the circuit diagram, the positive of the LED (the longest pin) is connected to pin 13 and the negative to the resistor (resistors do not have negative or positive end).

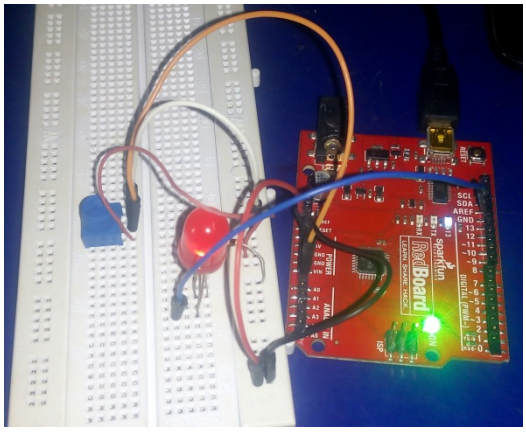
PROJECT 2 (15 minutes)

USING A POTENTIOMETER:

This is an extension to project one as potentiometer is added to the circuit to vary the voltage. The potentiometer controls the amount of voltage following into the output component. In this project, it controls the intensity of the LED bulb. This potentiometer has three pins (and can be blue in color) as shown below:



HIGH INTENSITY



LOW INTENSITY

The centre pin connects to the pin Ao,

The positive end to the 5volts supply and

The negative end connects to the ground.

Make additions to the codes as shown, compile and verify.

ARDUINO CODE

```
int sensorPin = 0;    // The potentiometer is connected to analog pin 0
int ledPin = 13;      // The LED is connected to digital pin 13
void setup() // this function runs once when it starts up
{
 pinMode(ledPin, OUTPUT);
}
void loop() // this function runs repeatedly after setup() finishes
{
  int sensorValue;
```

```
 sensorValue = analogRead(sensorPin);
  digitalWrite(ledPin, HIGH);     // Turn the LED on
  delay(sensorValue);             // Pause for sensorValue milliseconds
  digitalWrite(ledPin, LOW);      // Turn the LED off
  delay(sensorValue);             // Pause for sensorValue milliseconds
}
```

Procedure:

- Write the codes as shown above.
- Connect all components correctly.
- Connect the Arduino Board with the Computer using the USB cable.
- Verify the written codes and upload to the board.
- Turn the nub on the potentiometer in different directions and observe the LED bulb.

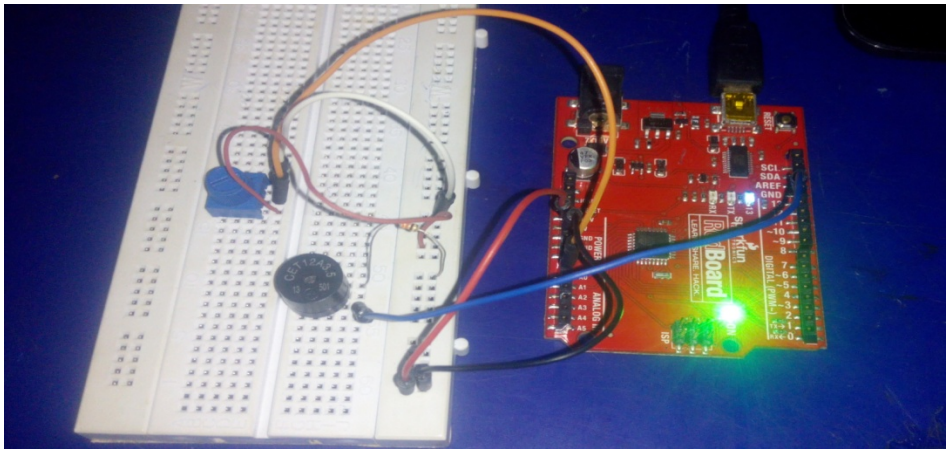
What did you observe?

- State your observations.
- Let them keep exploring different speed times using this component.

PROJECT 3 (15 minutes)

USING A POTENTIOMETER TO CONTROL THE VOUME OF A BUZZER

This is an extension to project 2 as potentiometer is added to the circuit to vary the voltage. The potentiometer controls the amount of voltage flowing into the output component. In this project, it controls the volume of the buzzer.



Procedure:

- Write the codes exactly as it is in project 2 above.
- Connect all components correctly as shown in the picture above (Replacing the LED with the Buzzer)
- Connect the Arduino Board with the Computer using the USB cable.
- Verify the written codes and upload to the board.
- Turn the nub on the potentiometer in different directions and observe the sound of the buzzer.

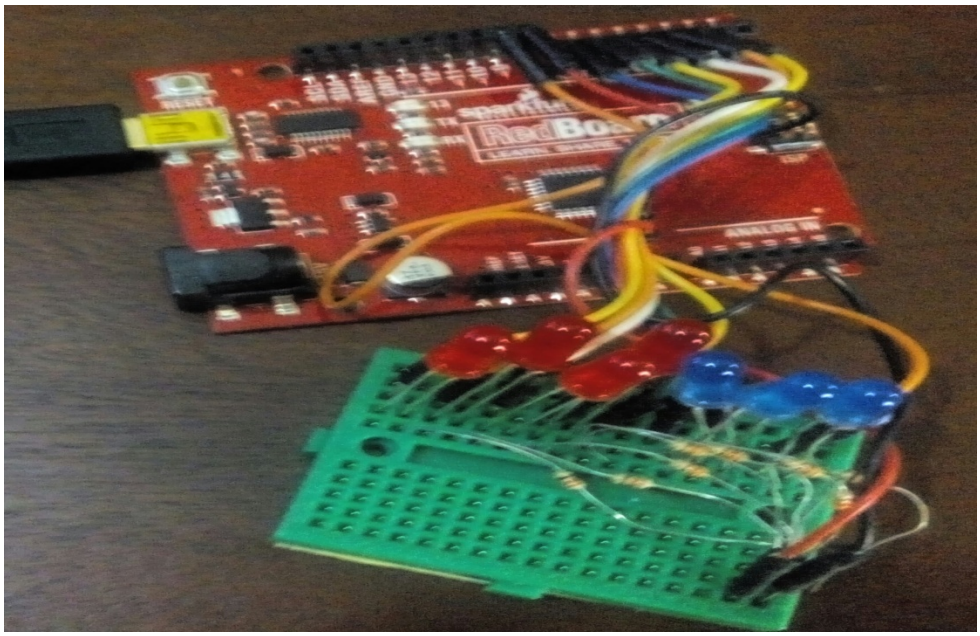
What did you observe?

- State your observations.
- Let them keep exploring different times using this component.

PROJECT 4 (30 minutes)

CONTROLLING MULTIPLE LEDs

This is similar to the projects we have been doing so far just the addition of more LEDs. Here we will be controlling 8 LED bulbs.



ARDUINO CODE

```
// To keep track of all the LED pins, we'll use an "array".
// An array lets you store a group of variables, and refer to them by their position
int ledPins[] = {2,3,4,5,6,7,8,9};
// We're using the values in this array to specify the pin numbers
// that the eight LEDs are connected to. LED 0 is connected to
// pin 2, LED 1 is connected to pin 3, etc
void setup()
{
  int index;
  // we will use "for() loops" to step variables from
  // one value to another, and perform a set of instructions for count up or down.
```

```
  // Every for() loop has three statements separated by
  // semicolons (;):
  //   1. Something to do before starting
  //   2. A test to perform; as long as it's true, keep looping
  //   3. Something to do after each loop (increase a variable)
  {
    pinMode(ledPins[index],OUTPUT);
    // ledPins[index] is replaced by the value in the array.
    // For example, ledPins[0] is 2
  }
}
void loop()
{
  oneAfterAnotherNoLoop();  // Light up all the LEDs in turn
  //oneAfterAnotherLoop();  // Same as one after another no Loop but with much less typing
  //oneOnAtATime();        // Turn on one LED at a time scrolling down the line
  //pingPong();            // Light the LEDs middle to the edges
  //marquee();             // Chase lights like you see on signs
  //randomLED();           // Blink LEDs randomly
}
void oneAfterAnotherNoLoop()
{
  int delayTime = 100; // time (milliseconds) to pause between LEDs
                       // make this smaller for faster switching
  // turn all the LEDs on:
  digitalWrite(ledPins[0], HIGH);  //Turns on LED #0 (pin 2)
  delay(delayTime);                //wait delayTime milliseconds
  digitalWrite(ledPins[1], HIGH);  //Turns on LED #1 (pin 3)
  delay(delayTime);                //wait delayTime milliseconds
```

```
digitalWrite(ledPins[2], HIGH);  //Turns on LED #2 (pin 4)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[3], HIGH);  //Turns on LED #3 (pin 5)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[4], HIGH);  //Turns on LED #4 (pin 6)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[5], HIGH);  //Turns on LED #5 (pin 7)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[6], HIGH);  //Turns on LED #6 (pin 8)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[7], HIGH);  //Turns on LED #7 (pin 9)
delay(delayTime);                //wait delayTime milliseconds
// turn all the LEDs off:
digitalWrite(ledPins[7], LOW);   //Turn off LED #7 (pin 9)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[6], LOW);   //Turn off LED #6 (pin 8)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[5], LOW);   //Turn off LED #5 (pin 7)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[4], LOW);   //Turn off LED #4 (pin 6)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[3], LOW);   //Turn off LED #3 (pin 5)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[2], LOW);   //Turn off LED #2 (pin 4)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[1], LOW);   //Turn off LED #1 (pin 3)
delay(delayTime);                //wait delayTime milliseconds
digitalWrite(ledPins[0], LOW);   //Turn off LED #0 (pin 2)
delay(delayTime);                //wait delayTime milliseconds
```

```
}
void oneAfterAnotherLoop()
{
  int index;
  int delayTime = 100; // milliseconds to pause between LEDs
                       // make this smaller for faster switching
  for(index = 0; index <= 7; index++)
  {
    digitalWrite(ledPins[index], HIGH);
    delay(delayTime);
  }
  // Turn all the LEDs off:
  // This for() loop will step index from 7 to 0
  // (putting "--" after a variable means subtract one from it)
  // and will then use digitalWrite() to turn that LED off.
  for(index = 7; index >= 0; index--)
  {
    digitalWrite(ledPins[index], LOW);
    delay(delayTime);
  }
}
void oneOnAtATime()
{
  int index;
  int delayTime = 100; // milliseconds to pause between LEDs
                       // make this smaller for faster switching
  // step through the LEDs, from 0 to 7
  for(index = 0; index <= 7; index++)
  {
```

```
    digitalWrite(ledPins[index], HIGH);  // turn LED on
    delay(delayTime);                    // pause to slow down
    digitalWrite(ledPins[index], LOW);   // turn LED off
  }
}
void pingPong()
{
  int index;
  int delayTime = 100; // milliseconds to pause between LEDs
                       // make this smaller for faster switching

  // step through the LEDs, from 0 to 7

  for(index = 0; index <= 7; index++)
  {
    digitalWrite(ledPins[index], HIGH);  // turn LED on
    delay(delayTime);                    // pause to slow down
    digitalWrite(ledPins[index], LOW);   // turn LED off
  }
  for(index = 7; index >= 0; index--)
  {
    digitalWrite(ledPins[index], HIGH);  // turn LED on
    delay(delayTime);                    // pause to slow down
    digitalWrite(ledPins[index], LOW);   // turn LED off
  }
}
void marquee()
{
  int index;
```

```
  int delayTime = 200;
  for(index = 0; index <= 3; index++) // Step from 0 to 3
  {
    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
  }
}
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
}
```

CONCLUSION(10 minutes)

Ask the students the basic real-world applications of the four projects carried out. For example,

- The blinking LED can be used for indicating faults in a machine.
- The Buzzer can be used as an alarm system
- The multiple LED project can be used for LED display in a device by creating a letter with the LEDs.

EXPLORE, PLAY & SHARE!

- Can you make the bulb blink faster?
- What Challenges did you encounter?
- How did you solve them?
- Have you tried witting fresh lines of codes all by yourself?
- What are the real-world applications of these projects?

Ensure the students discuss their observations.

Let them state various ways these projects can solve problems.

Relate this lesson to various Real-world Applications.

The Key here is **problem solving**.

"Today's Play is tomorrow's Pay"

REFERENCES

To learn more, check out these sites:

https://www.arduino.cc/

https://www.learn.sparkfun.com/.../redboard-hookup-guide/all.pdf

Arduino Programming Lesson Plan 1

Instructors should always make more research to stay updated and become more proficient.