Appendix A – Arduino Code

Step 10 – ReadButton.ino

```
#define buttonPin 2

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
  Serial.println("Setup Complete");
}

void loop() {
  if (digitalRead(buttonPin))
    Serial.println("Button Pressed");
  else
    Serial.println("Button Not Pressed");
  delay(2000);
}
```

Step 12 – ReadAnalog.ino

```
#define buttonPin 2

void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
  Serial.println("Setup Complete");
}

void loop() {
  if (digitalRead(buttonPin))
    Serial.println("Button Pressed");
  else
    Serial.println("Button Not Pressed");
  delay(2000);
}
```

Step 13 – ReadVoltage.ino

```
void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
  Serial.println("Setup Complete");
}

void loop() {
  int analogValue = analogRead(A0);
  float voltage = analogValue * (5.0 / 1023.0);
  Serial.print(voltage);
  Serial.println(" volts");
  delay(250);
}
```

Step 17 – LEDdimmer.ino

```
#define LEDpin 3

void setup() {
  pinMode(LEDpin, INPUT);
}

void loop() {
  analogWrite(LEDpin, 250);  // bright
  delay(1000);
  analogWrite(LEDpin, 150);  // dim
  delay(1000);
  analogWrite(LEDpin, 50);   // dimmer
  delay(1000);
}
```

Step 19 – ReadTemp.ino

```
// Read Temperature Sensor TMP36GT9Z on Pin A0

void setup() {
  pinMode(A0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int   analogValue = analogRead(A0);

  float volts = analogValue * (5.0 / 1023.0);
  float millivolts = volts * 1000;
  float tempC = (millivolts - 500) / 10;
  float tempF = (tempC * 1.8) + 32;

  Serial.print(tempC);
  Serial.print(" \xC2\xB0"); // degree symbol
  Serial.print("C  ");

  Serial.print(tempF);
  Serial.print(" \xC2\xB0"); // degree symbol
  Serial.println("F");

  delay(1500);
}
```

Step 20 – ControlFlow.ino

```
// anything after a double slash is a comment
// comments are not run in the program
// a comment documents what the program is doing as a
//           note to others or even you in the future
/* anything inside a pair of slash-stars is a comment */

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

void loop() {
  int i;

  i=0;
  if (i<1) {
    Serial.println("Yes, zero is less than one.");
  }

  if (i>1) {
    Serial.println("No!  This line should never be reached.");
  } else {
    Serial.println("Yes, zero is not greater than one.");
  }

  Serial.println("Let's count to five using a for loop...");
  for (i=1; i < 6; i++) {
    Serial.println(i);
  }

  i=1;
  Serial.println("Let's count to five using a while loop...");
  while (i <= 5) {
    Serial.println(i);
    i++;
  }

  Serial.println("Let's get stuck in an infinite loop...");
  while (true);  // true is always true
}
```

Step 21 – DataArrays.ino

```
void setup() {
  Serial.begin(9600);
  Serial.println("Setup Complete");
}

void loop() {
  Serial.println("intialize an array of five integers:");
  Serial.println("int myArray[5] = {3, 4, 5, 6, 7}");
  int myArray[5] = {3, 4, 5, 6, 7};

  for (int c=0; c<5; c++){
    Serial.print("element ");
    Serial.print(c);
    Serial.print(" of myArray is ");
    Serial.println(myArray[c]);
  }

  Serial.println("The first element is index 0");
  Serial.println("The last element is index 4");
  Serial.println("There is no index 5");

  Serial.println(); //skip a line
  Serial.println("An array of characters is a string of text");
  Serial.println("char myString[]=\"my pet is a cat\"");
  char myString[]="my pet is a cat";
  Serial.println(myString);
  Serial.println("change element 12 from c to r");
  myString[12]='r';
  Serial.println(myString);

  while(1);  //just wait forever
}
```

Step 22 – Sounds.ino

```
#define NOTE_D7  2349
#define NOTE_E7  2637
#define NOTE_C7  2093
#define NOTE_C6  1047
#define NOTE_G6  1568

#define noteDuration 800
#define notePause 900

#define buzzerPin 8

void setup() {

}

void loop() {
  tone(buzzerPin, NOTE_D7, noteDuration);
  delay(notePause);
  noTone(buzzerPin);

  tone(buzzerPin, NOTE_E7, noteDuration);
  delay(notePause);
  noTone(buzzerPin);

  tone(buzzerPin, NOTE_C7, noteDuration);
  delay(notePause);
  noTone(buzzerPin);

  tone(buzzerPin, NOTE_C6, noteDuration);
  delay(notePause);
  noTone(buzzerPin);

  tone(buzzerPin, NOTE_G6, noteDuration);
  delay(notePause);
  noTone(buzzerPin);

  delay(4000); //wait 4 seconds = 4,000 ms
}
```

Step 23 – Ultrasound.ino

```
#define echoPin 2
#define trigPin 3
#define SpeedOfSound 0.0343  // in cm-per-microsecond

void setup() {
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  digitalWrite(trigPin, LOW);
  Serial.begin(9600);
}

void loop() {
  float echoTime, distance;

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  echoTime = pulseIn(echoPin, HIGH);
  distance = (echoTime/2.0) * SpeedOfSound;
  Serial.print("distance in cm: ");
  Serial.println(distance);
  delay(200);
}
```

Step 26 – OLEDtext.ino

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

void setup() {
  Serial.begin(115200);

  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("SSD1306 allocation failed");
    while(1);
  }

  delay(2000);
  display.clearDisplay();
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(10, 25);
  display.println("HACKERBOX");
  display.display();
  while(1);
}
```

Step 27 – CommonCathodeRGB.ino

```
#define RED_LED_PIN    6
#define GREEN_LED_PIN  5
#define BLUE_LED_PIN   3

void setup() {
  pinMode(RED_LED_PIN, OUTPUT);
  pinMode(GREEN_LED_PIN, OUTPUT);
  pinMode(BLUE_LED_PIN, OUTPUT);
}

void loop() {
  RGB_LED(255, 0, 0);     // Red
  RGB_LED(0, 255, 0);     // Green
  RGB_LED(0, 0, 255);     // Blue
  RGB_LED(255, 255, 125); // Purple
  RGB_LED(0, 255, 255);   // Cyan
  RGB_LED(255, 0, 255);   // Magenta
  RGB_LED(255, 255, 0);   // Yellow
  RGB_LED(255, 255, 255); // White
}

void RGB_LED(int red_LED_value, int green_LED_value, int blue_LED_value)
{
  analogWrite(RED_LED_PIN, red_LED_value);
  analogWrite(GREEN_LED_PIN, green_LED_value);
  analogWrite(BLUE_LED_PIN, blue_LED_value);
  delay(2000);
}
```

Step 29 – Capacitance.ino

```
// Simple Capacitance Meter
// For capacitors between 18 pF and 470 uF
// Results outputs to the Arduino Serial Monitor
//
// Sketch from Circuit Basics:
// https://www.circuitbasics.com/how-to-make-an-arduino-capacitance-meter/
/////////////////////////////////////////////////////////////////////

const int OUT_PIN = A2;
const int IN_PIN = A0;
const float IN_STRAY_CAP_TO_GND = 24.48;
const float IN_CAP_TO_GND  = IN_STRAY_CAP_TO_GND;
const float R_PULLUP = 34.8;
const int MAX_ADC_VALUE = 1023;

void setup(){
  pinMode(OUT_PIN, OUTPUT);
  pinMode(IN_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop(){
    pinMode(IN_PIN, INPUT);
    digitalWrite(OUT_PIN, HIGH);
    int val = analogRead(IN_PIN);
    digitalWrite(OUT_PIN, LOW);

    if (val < 1000){
      pinMode(IN_PIN, OUTPUT);

      float capacitance = (float)val * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE -
val);

      Serial.print(F("Capacitance Value = "));
      Serial.print(capacitance, 3);
      Serial.print(F(" pF ("));
      Serial.print(val);
      Serial.println(F(") "));
    }

    else{
      pinMode(IN_PIN, OUTPUT);
      delay(1);
      pinMode(OUT_PIN, INPUT_PULLUP);
      unsigned long u1 = micros();
      unsigned long t;
      int digVal;

      do{
        digVal = digitalRead(OUT_PIN);
        unsigned long u2 = micros();
        t = u2 > u1 ? u2 - u1 : u1 - u2;
      } while ((digVal < 1) && (t < 400000L));
```

```
      pinMode(OUT_PIN, INPUT);
      val = analogRead(OUT_PIN);
      digitalWrite(IN_PIN, HIGH);
      int dischargeTime = (int)(t / 1000L) * 5;
      delay(dischargeTime);
      pinMode(OUT_PIN, OUTPUT);
      digitalWrite(OUT_PIN, LOW);
      digitalWrite(IN_PIN, LOW);

      float capacitance = -(float)t / R_PULLUP / log(1.0 - (float)val /
(float)MAX_ADC_VALUE);

      Serial.print(F("Capacitance Value = "));
      if (capacitance > 1000.0){
        Serial.print(capacitance / 1000.0, 2);
        Serial.print(F(" uF"));
      }
      else{
        Serial.print(capacitance, 2);
        Serial.print(F(" nF"));
      }

      Serial.print(F(" ("));
      Serial.print(digVal == 1 ? F("Normal") : F("HighVal"));
      Serial.print(F(", t= "));
      Serial.print(t);
      Serial.print(F(" us, ADC= "));
      Serial.print(val);
      Serial.println(F(")"));
    }
    while (millis() % 1000 != 0);
}
```

Step 31 – DiodeTest.ino

```
////////////////////////////////////////////////////////
//
// Measure direction of current flow between pins A1 and A2
//
// (Requires two 1K voltage sensing resistors)
//
////////////////////////////////////////////////////////

#define A1_R A0  //Pin A0 controls the 1K resistor on A1
#define A2_R A3  //Pin A3 controls the 1K resistor on A2

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

void loop() {

  float V1TO2, V2TO1;

  // FIRST: measure the voltage between A1 to A2
  // Connect A2 to GND through the 1K resistor at A3
  pinMode(A2, INPUT);
  pinMode(A2_R, OUTPUT);
  digitalWrite(A2_R, 0);
  // Connect A1 directly to 5V
  pinMode(A1_R, INPUT);
  pinMode(A1, OUTPUT);
  digitalWrite(A1, 1);
  // Read voltage at A2
  delay(100);
  V1TO2 = 5-5*((float)analogRead(A2)/1023);
  Serial.print("Voltage difference between A1 and A2: ");
  Serial.print(V1TO2);
  if (V1TO2 < 3)
    Serial.println(" - current is flowing from A1 to A2");
  else
    Serial.println(" - current is not flowing from A1 to A2");

  delay(3000);

  // SECOND: measure the voltage between A2 to A1
  // Connect A2 directly to 5V
  pinMode(A2_R, INPUT);
  pinMode(A2, OUTPUT);
  digitalWrite(A2, 1);
  // Connect A1 to GND through the 1K resistor at A0
  pinMode(A1, INPUT);
  pinMode(A1_R, OUTPUT);
  digitalWrite(A1_R, 0);
  // Read voltage at A1
  delay(100);
  V2TO1 = 5-5*((float)analogRead(A1)/1023);
  Serial.print("Voltage difference between A2 and A1: ");
```

```
  Serial.print(V2TO1);
  if (V2TO1 < 3)
    Serial.println(" - current is flowing from A2 to A1");
  else
    Serial.println(" - current is not flowing from A2 to A1");

  delay(3000);
}
```