

Arduino Programming – Part 5: User-defined functions

ME 120

Mechanical and Materials Engineering

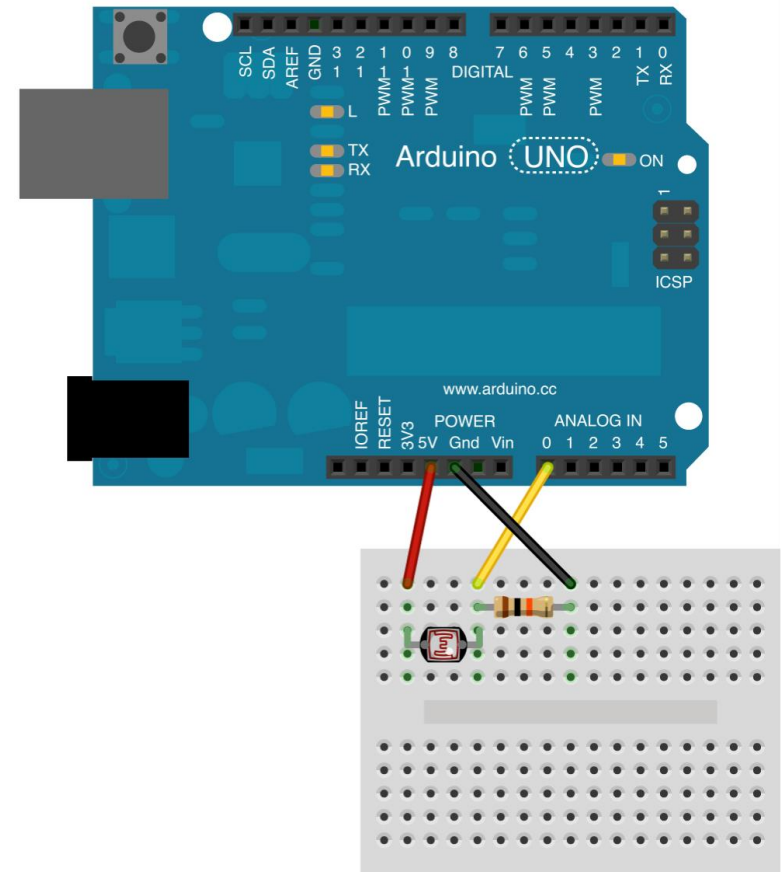
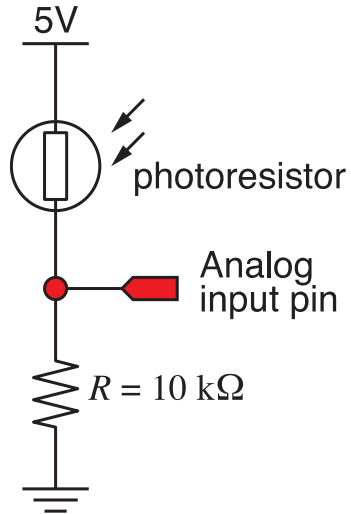
Portland State University

<http://web.cecs.pdx.edu/~me120>

Overview

- Continue case study of analog input with a photoresistor
- Use loop to compute the average of multiple readings
- Create a user-defined function
 - ❖ Perform a generic task
 - ❖ Reuse in other programs
 - ❖ Encapsulate code to separate the details
- See on-line reference:
 - ❖ <http://arduino.cc/en/Reference/HomePage>
 - ❖ <http://www.arduino.cc/en/Reference/FunctionDeclaration>

Voltage divider circuit for photoresistor



Display voltage divider output on the serial monitor

Connect the voltage divider output to analog pin 0

```
void setup() {  
  Serial.begin(9600);      // Initialize serial port object  
}  
  
void loop() {  
  int reading;  
  float voltage;  
  
  reading = analogRead(0);    // Read analog input channel 0  
  voltage = reading*(5.0/1023.0); // and convert to voltage  
  
  Serial.print("Reading: ");  
  Serial.print(reading);      // Print the raw reading  
  Serial.print(" Voltage: "); // Make a horizontal space  
  Serial.println(voltage);    // Print voltage value  
}
```

See <http://arduino.cc/en/Reference/AnalogRead>

Average multiple readings

Computing the average

- Basic procedure
 - ❖ Make several readings with `analogRead`
 - ❖ Add the readings and divide by the number of readings
- Let n be the number of readings

$$n = 2: \quad \bar{x} = \frac{1}{2}(x_1 + x_2)$$

$$n = 3: \quad \bar{x} = \frac{1}{3}(x_1 + x_2 + x_3)$$

$$\text{any } n: \quad \bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n)$$

- Express with summation notation

$$\bar{x} = \frac{1}{n} \sum x_i$$

Evaluation summation with a loop

- Basic loop structure for n readings

```
int i,n=10;
float ave,sum;

sum = 0.0;    // initial value of sum
for ( i=1; i<=n; i++ ) {
    sum = sum + analogRead(0);
}
ave = sum/float(n);
```

- Add this code to the basic reading code presented earlier

Display average of voltage divider output on the serial monitor

```
void setup() {
  Serial.begin(9600);          // Initialize serial port object
}

void loop() {
  // we can get rid of "int reading;"
  float voltage;

  // let's add the average of n readings block instead of one reading. Before we
  // had reading = analogRead(A0);
  int i,n=10;
  float ave,sum;

  sum = 0.0;    // initial value of sum
  for ( i=1; i<=n; i++ ) {
    sum = sum + analogRead(0);
  }
  ave = sum/float(n);
  voltage = ave*(5.0/1023.0); // replace "reading" by "ave"

  Serial.print("Average: "); // replace "Reading" by "Average"
  Serial.print(ave);         // replace "reading" by "ave"
  Serial.print(" Voltage: ");
  Serial.println(voltage);
}
```


Writing a function to compute
the average of n readings

The average_reading function

```
void setup() {
  Serial.begin(9600);          // Initialize serial port object
}

void loop() {
  float voltage, ave_photo;
  ave_photo = average_reading();
  voltage = ave_photo*(5.0/1023.0); // replace "ave" by "ave_photo"

  Serial.print("Average: ");
  Serial.print(ave_photo);      // replace "ave" by "ave_photo"
  Serial.print(" Voltage: ");
  Serial.println(voltage);
}

float average_reading(){
  int i,n=10;
  float ave,sum;

  sum = 0.0;    // initial value of sum
  for ( i=1; i<=n; i++ ) {
    sum = sum + analogRead(0);
  }
  ave = sum/float(n);
  return(ave); // add this line: return the value "ave"
}
```

Update the function to allow the pin to be a variable

Add an input to make average_reading work with any analog pin

```
void setup() {
  Serial.begin(9600);          // Initialize serial port object
}

void loop() {
  float voltage, ave_photo;
  ave_photo = average_reading(0);
  voltage = ave_photo*(5.0/1023.0);

  Serial.print("Average: ");
  Serial.print(ave_photo);
  Serial.print(" Voltage: ");
  Serial.println(voltage);
}

float average_reading(int sensor_pin) {
  int i,n=10;
  float ave,sum;

  sum = 0.0;    // initial value of sum
  for ( i=1; i<=n; i++ ) {
    sum = sum + analogRead(sensor_pin);
  }
  ave = sum/float(n);
  return(ave);
}
```

Update the function to allow the number of readings to be a variable

Add a second input to make average_reading work with any number of readings to average

```
void setup() {
  Serial.begin(9600);          // Initialize serial port object
}

void loop() {
  float voltage, ave_photo;
  ave_photo = average_reading(0, 10);
  voltage = ave_photo*(5.0/1023.0);

  Serial.print("Average: ");
  Serial.print(ave_photo);
  Serial.print(" Voltage: ");
  Serial.println(voltage);
}

float average_reading(int sensor_pin, int nave){
  int i;          // get rid of "int n=10;"
  float ave,sum;

  sum = 0.0;      // initial value of sum
  for ( i=1; i<=nave; i++ ) {
    sum = sum + analogRead(sensor_pin);
  }
  ave = sum/float(nave);
  return(ave);
}
```

Even better: pin number and number of readings to average are defined at top

```
void setup() {
  Serial.begin(9600);          // Initialize serial port object
}

void loop() {
  int photo_pin=0, number_readings=10;
  float voltage, ave_photo;
  ave_photo = average_reading(photo_pin, number_readings);
  voltage = ave_photo*(5.0/1023.0);

  Serial.print("Average: ");
  Serial.print(ave_photo);
  Serial.print(" Voltage: ");
  Serial.println(voltage);
}

float average_reading(int sensor_pin, int nave){
  int i;
  float ave,sum;

  sum = 0.0;    // initial value of sum
  for ( i=1; i<=nave; i++ ) {
    sum = sum + analogRead(sensor_pin);
  }
  ave = sum/float(nave);
  return(ave);
}
```

Summary of user-defined functions

- You chose the name
 - ❖ Make sure the name is not already used
- You chose the type of return value
- You choose the number and type of inputs
 - ❖ Input types are declared in the function definition
 - `float average_reading(int sensor_pin, int nave) { ... }`
 - ❖ Input variables are used in the body of the function
 - ❖ When function is called in another section of code, any variable that matches the type of the declared input can be used
 - `reading = average_reading(photo_pin, 15);`
- Variables in the function are local
 - ❖ Calling function is not affected by local variables and logic in the function