Arduino Programming – Part 5: User-defined functions

ME 120

Mechanical and Materials Engineering

Portland State University

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

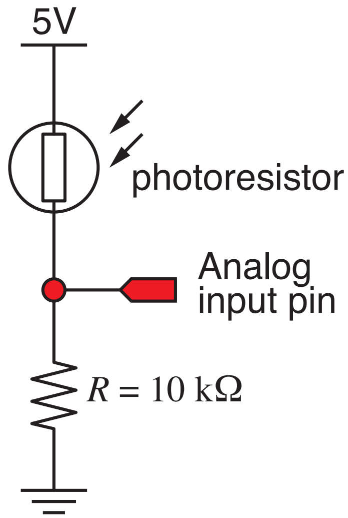
Fall 2013

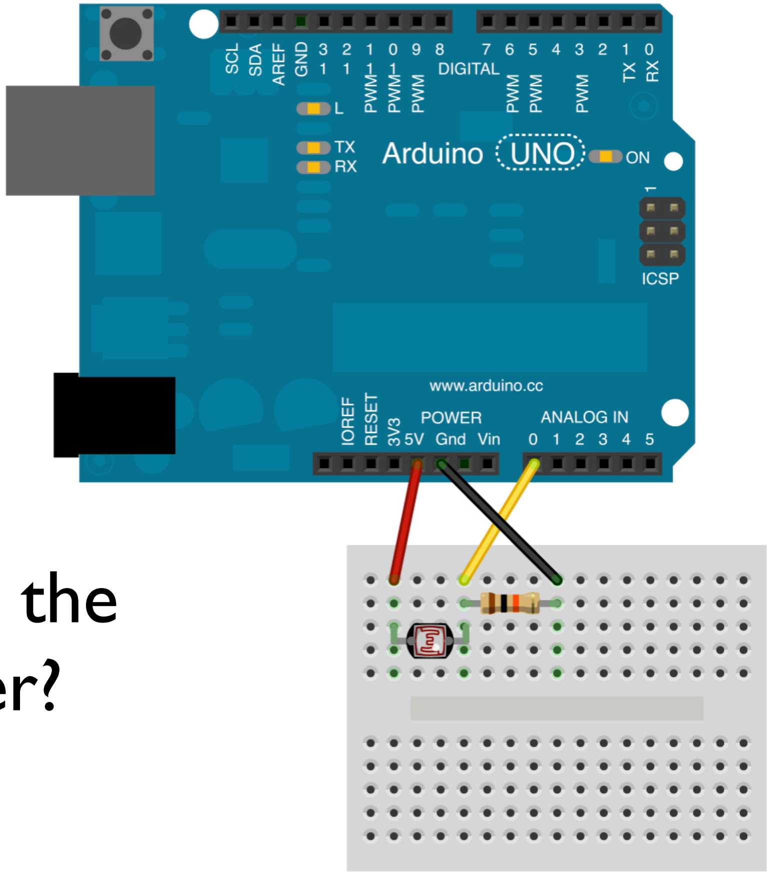
Overview

- 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





Why is the fixed resistor on the bottom of the voltage divider?

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(A0);        //  Read analog input channel 0
  voltage = reading*(5.0/1023.0);  //  and convert to voltage

  Serial.print(reading);           //  Print the raw reading
  Serial.print(”  ”);              //  Make a horizontal space
  Serial.println(voltage);         //  Print voltage value
}
```

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

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

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

$$\text{any}\, n: \quad \overline{x} = \frac{1}{n}\left(x_1 + x_2 + \ldots + x_n\right)$$

- Express with summation notation

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

Evaluation summation with a loop

- Basic loop structure for n readings

```
int i,n=10,sensor_pin=0;
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);
```

- Add this code to the basic reading code presented earlier

Writing a function to compute the average of n readings

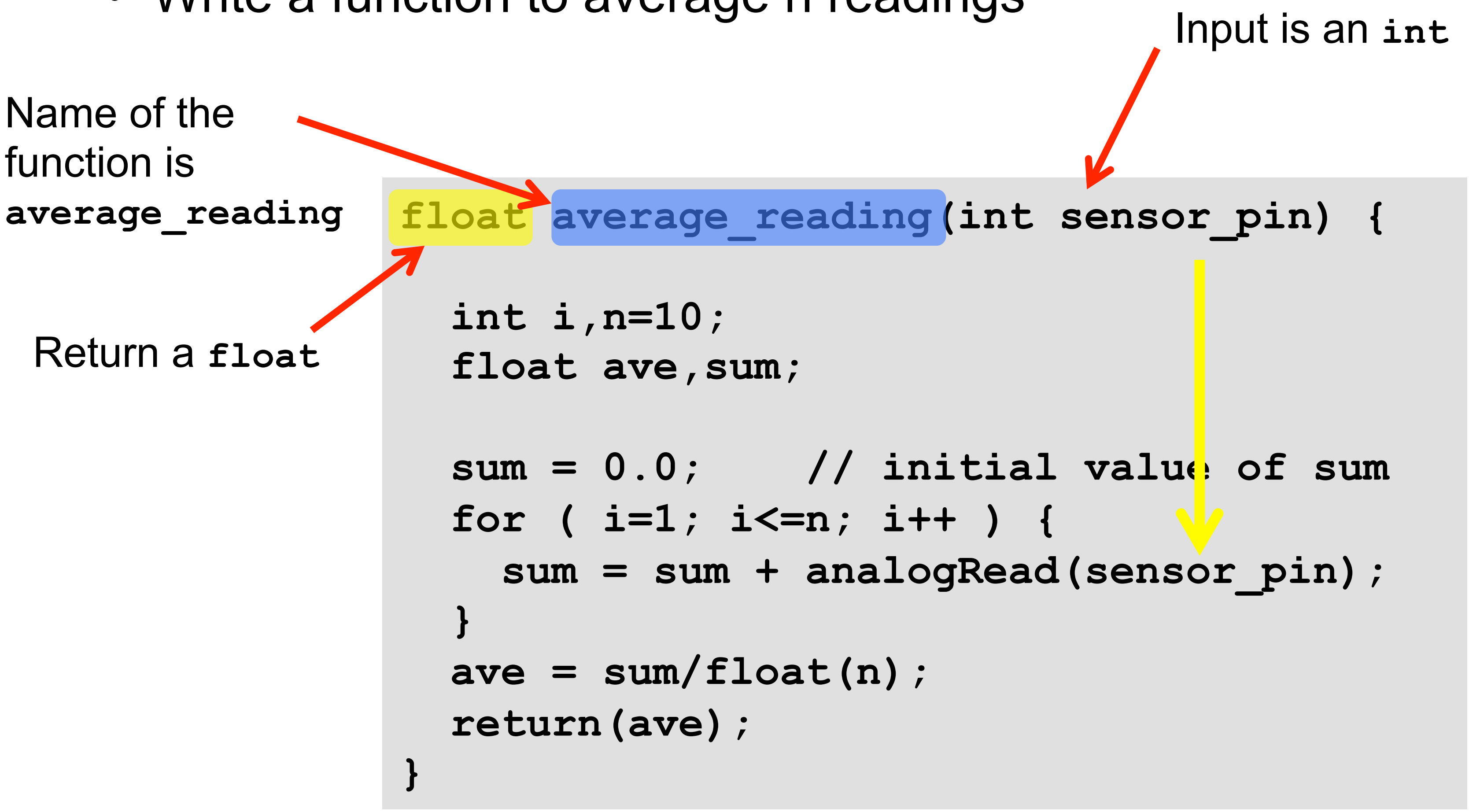
User-defined functions

- Functions are reusable code modules
  - Functions encapsulate details of a task into larger building blocks
  - Well-written functions can be reused
  - Functions can accept input (or not) and return output (or not)
  - All Arduino sketches have at least two functions
    - setup: runs once to configure the system
    - loop: runs repeatedly after setup is complete

- Reference on the Arduino web site
  - http://www.arduino.cc/en/Reference/FunctionDeclaration

The average_reading function

- Write a function to average n readings



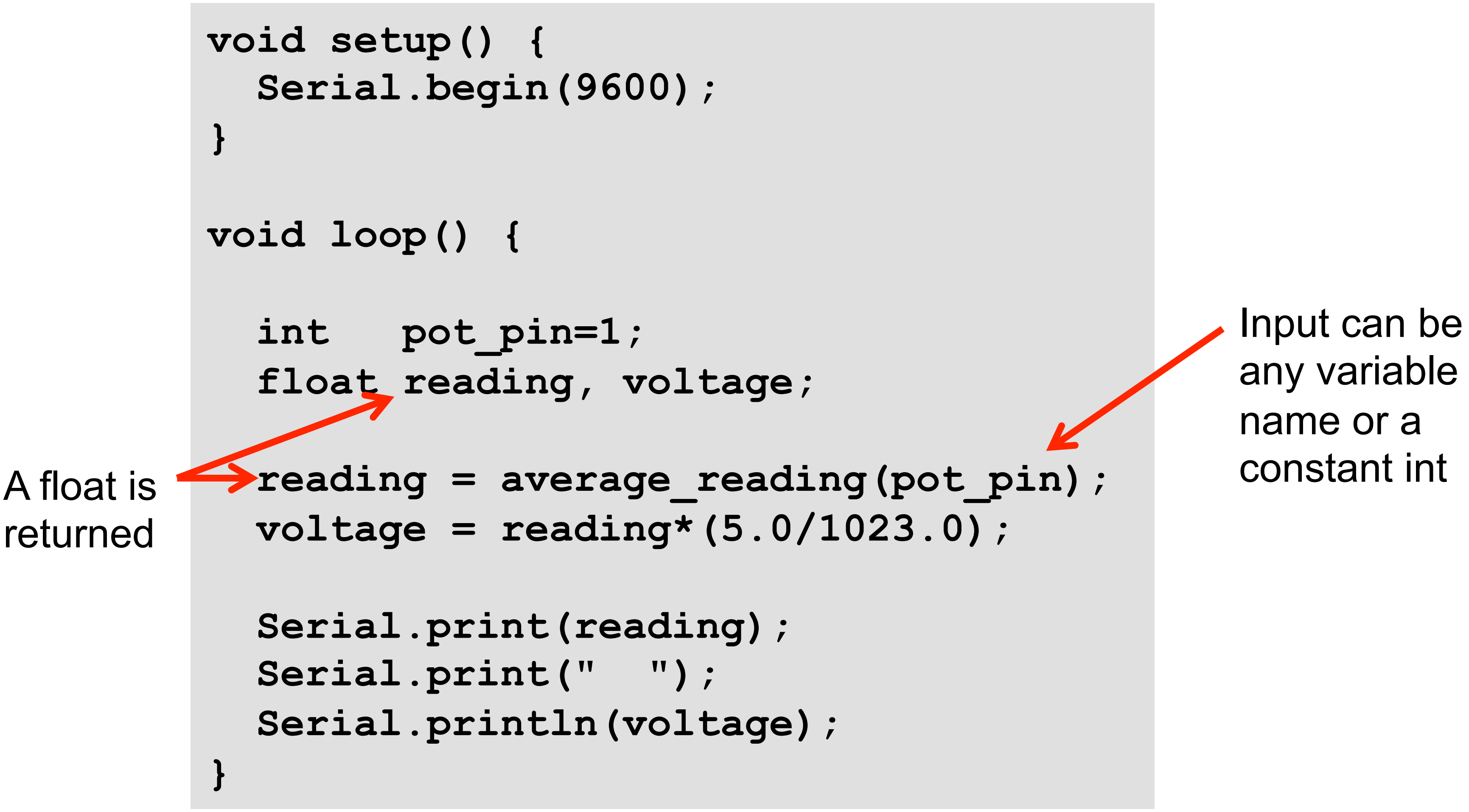
Input is an **`int`**

Name of the function is **`average_reading`**

Return a **`float`**

```
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);
}
```

Use the average_reading function



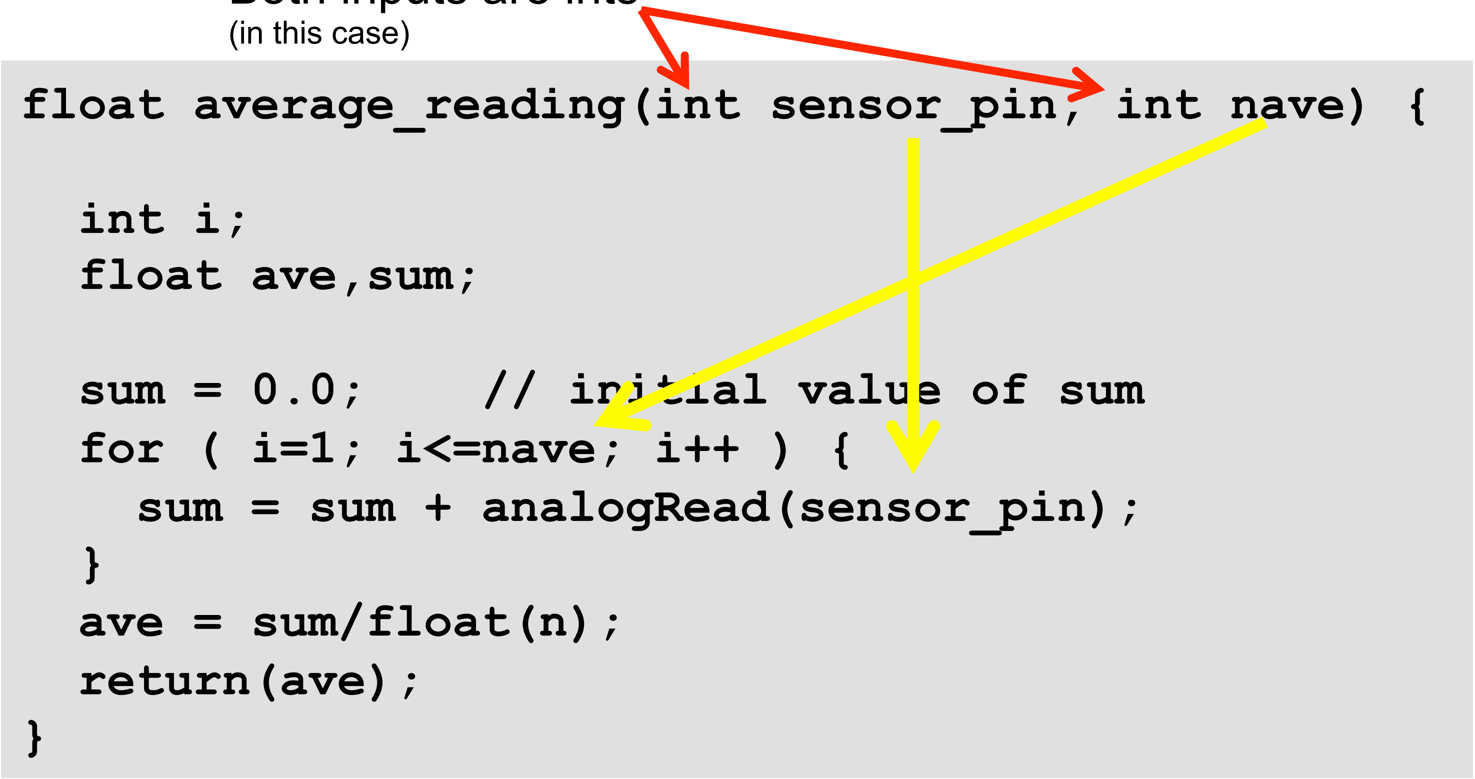
```
void setup() {
  Serial.begin(9600);
}

void loop() {

  int   pot_pin=1;
  float reading, voltage;

  reading = average_reading(pot_pin);
  voltage = reading*(5.0/1023.0);

  Serial.print(reading);
  Serial.print("  ");
  Serial.println(voltage);
}
```

A float is returned

Input can be any variable name or a constant int

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

Update the average_reading function



Both inputs are ints
(in this case)

```
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(n);
  return(ave);
}
```

Use the updated average_reading function

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

void loop() {

  int   n=15, pot_pin=1;
  float reading, voltage;

  reading = average_reading(pot_pin,n);
  voltage = reading*(5.0/1023.0);

  Serial.print(reading);
  Serial.print("  ");
  Serial.println(voltage);
}
```

Inputs can be any variable name or a constant int

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(pot_pin, 15);
- Variables in the function are local
  - ❖ Calling function is not affected by local variables and logic in the function