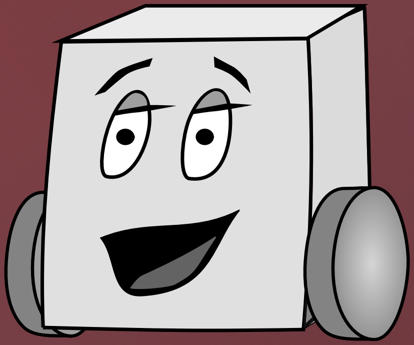


E11 Lecture 6: Even More C!!!

Prof. David Money Harris
Fall 2014

Outline

- Control Statements
- Arrays
- Function Calls
- Timing

Control Statements

- if
- if / else
- switch / case
- while
- do / while
- for

if Statement

```
if (i == 25) {
  Serial.println(“You guessed the magic number!”);
}
y = 42;
```

if / else Statement

```
if (i == 25) {
  Serial.println(“You guessed the magic number!”);
}
else {
  Serial.println(“Try again!”);
}
```

switch / case Statement

```
switch (var) {
  case 0:
    Serial.println(“Nice choice!”);
    break;
  case 1:
    Serial.println(“I wouldn’t have done that!”);
    break;
  default:
    Serial.println(“You pressed an invalid number”);
}
```

Equivalent to

```
if (var == 0)       Serial.println(“Nice choice!”);
else if (var == 1)  Serial.println(“I wouldn’t have done that!”);
else                Serial.println(“You pressed an invalid number”);
```

while Statement

```
int x = 1;
while (x < 1000) {
  Serial.println(x);
  x = x*2;
}
```

do / while Statement

```
int x = 0;

do {
  delay(100);  // delay 100 ms between readings
  x = analogRead(0);
} while (x < 300);
```

for Loop

```
for (initialization; condition; loop operation)
  loop body
```

- `initialization`: executes before the loop begins
- `condition`: is tested at the beginning of each iteration
- `loop operation`: executes at the end of each iteration
- `loop body`: executes each time the condition is met

for Loop

```
int i;
int x = 1;

for (i = 2; i < 10; i++)
  x = x * i;
```

Arrays

- Collection of similar items
- Example syntax:

```
int stuff[5];      // a 5-element array from index 0 – 4
```

Arrays: Example 1

```
// store the powers of 2 from 0-9 in an array
int powof2[10];

void setup() {
  int x = 1;

  Serial.begin(9600);

  for (int i = 0; i < 10; i++) {
    powof2[i] = x;
    x = x * 2;
  }
  for (i = 0; i < 10; i++) Serial.println(powof2[i]);
}

void loop() {
}
```

Your turn!

Write code that sums all of the elements of a 100-entry array called `data`. (You may assume `data` has been initialized.)

```
int data[100];

...
```

Your turn!

Write code that sums all of the elements of a 100-entry array called `data`.

```
int data[100];

...

int i, total = 0;

for (i = 0; i < 100; i++)
    total += data[i];
```

Arrays: Example 2

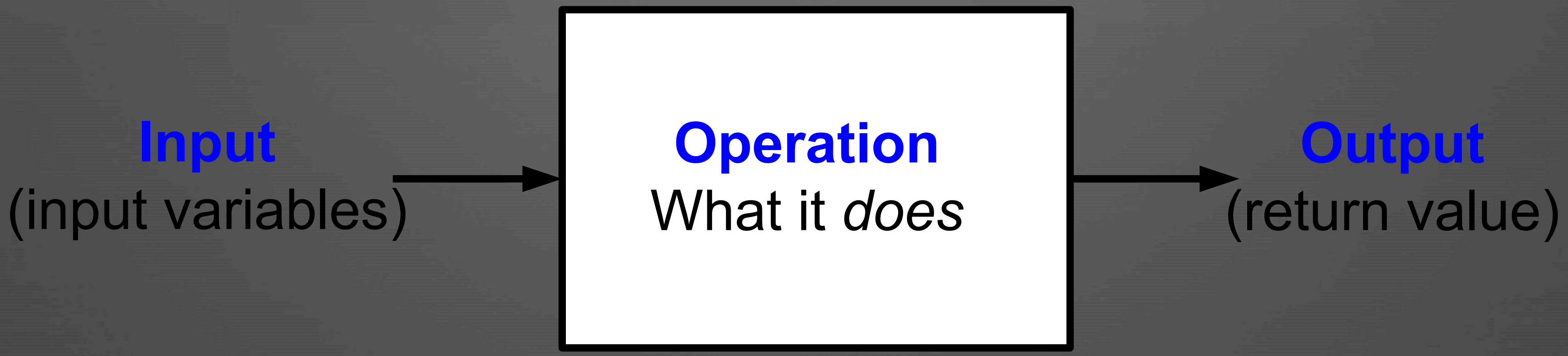
```
// This program initializes an array to random values
// between 0 and 100 and then finds the average value in
// the array.
int randVals[200];

void setup() {

  unsigned int i, average, total = 0;
  Serial.begin(9600);

  for (i = 0; i < 200; i++)
    randVals[i] = random(0,101);  // initialize array

  for (i = 0; i < 200; i++)
    total += randVals[i];  // compute sum

  average = total/200;
  Serial.print("Average value: ");
  Serial.println(average);
}
```

…Now using a function!

```
#define ARRAYSIZE 200
int randVals[ARRAYSIZE];

void setup() {
  unsigned int i;
  Serial.begin(9600);

  for (i = 0; i < ARRAYSIZE; i++)
    randVals[i] = random(0,101);
  getAverage(randVals, ARRAYSIZE);
}

int getAverage(int arr[], int len) {
  int i, average, total = 0;

  for (i = 0; i < len; i++)
    total += arr[i];
  average = total/len;
  Serial.print("Ave: "); Serial.println(average);
  return average;
}
```

Functions

- What is a function?
  - A function has:
    - Inputs
    - Outputs
    - Operation

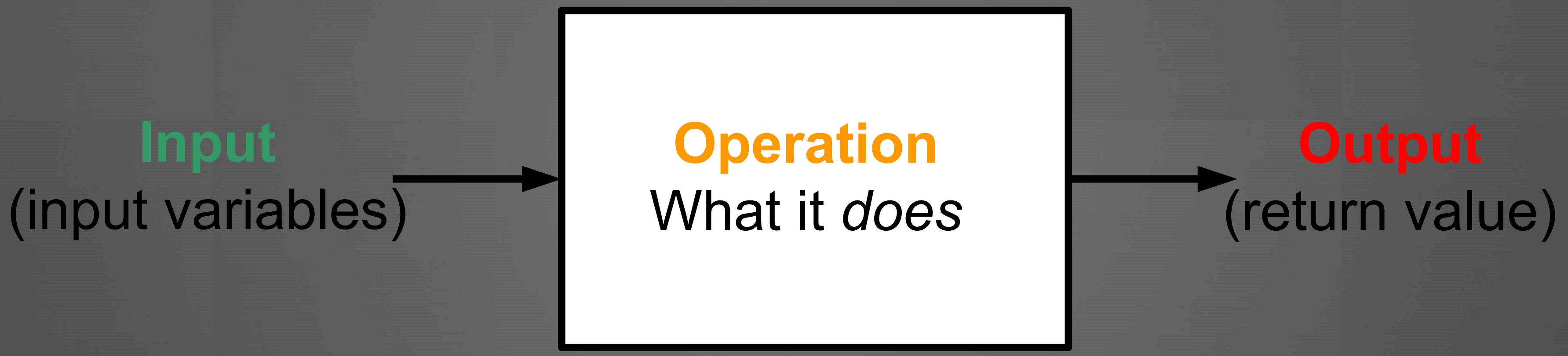


Input
(input variables)

Operation
What it *does*

Output
(return value)

Function Syntax



Input
(input variables)

Operation
What it *does*

Output
(return value)

```
output function_name(inputs)
{
  // operation
}
```

Function Syntax

no outputs

no inputs

```
void setup()
{
  // what it does...
}

void loop()
{
  // what it does…
}

int getAverage(int arr[], int len)
{
  // what it does...
}
```

Functions – Example Program

```
int getAverage(int arr[], int len) {
  int i, average, total = 0;

  for (i = 0; i < len; i++)
    total += arr[i];
  average = total/len;
  Serial.print("Ave: "); Serial.println(average);
  return average;
}
```

Your turn!

Write a function `getMax()` that returns the maximum of three numbers (that are inputs to the function).

Your turn again!

Write a function `getMaxArray()` that returns the maximum value in an array.

Functions – Example Program

```
void setup() {
   int avg;
   int distarray[100], distbinarray[100];
   ...

   readSensor(DISTSENSOR, distarray, 100);
   avg = getAverage(distarray, 100);
   convertToBin(avg, distarray, distbinarray, 100);
   printArray(distarray, 100);
   printArray(distbinarray, 100);
}
```

Functions – Example Program

```
void readSensor(int pin, int data[], int len) {
   for (int i=0; i<len; i++)
       data[i] = analogRead(pin - 14);
}

void convertToBin(int avg, int array[],
                  int arrayBin[], int len)
{
   for (int i=0; i<len; i++)
     if (array[i] < avg) arrayBin[i] = 0;
     else                arrayBin[i] = 1;
}
```

Functions – Example Program

```
int getAverage(int arr[], int len) {
  int i, average, total = 0;

  for (i = 0; i < len; i++)
    total += arr[i];
  average = total/len;
  Serial.print("Ave: "); Serial.println(average);
  return average;
}

void printArray(int array[], int len)
{
  for (int i=0; i<len; i++) {
    Serial.print(array[i]);
    Serial.print(" ");
  }
  Serial.println("");
}
```

Timing

- delay(time)
  - delays for time ms until continuing execution
- delayMicroseconds(time)
  - delays for time us until continuing execution
- millis()
  - returns time since program started in ms
  - returns unsigned long
- micros()
  - returns time since program started in us
  - returns unsigned long

Using timing for randSeed()

```
void setup() {
  int startTime;
  Serial.begin(9600);

  // prompt user
  Serial.println("Press any key to begin");
  while (Serial.available() == 0) ;  // wait for key press
  Serial.read();

  // get time from start of program to user key press (in ms)
  startTime = millis();
  Serial.print("startTime is: "); Serial.println(startTime);
  randomSeed(startTime);       // set the random seed
}

void loop() {
  int randNum = random(0, 100);
  Serial.print("Random number: "); Serial.println(randNum);
  delay(300);
}
```

Timing: frequency

```
#define REDLED 13

void setup()
{
  Serial.begin(9600); // set up Serial communication speed
  pinMode(REDLED, OUTPUT);   // red led is output
}

void loop()
{
  Serial.println("Starting loop\n");
  digitalWrite(REDLED, HIGH); // turn red LED on
  delay(1000);
  digitalWrite(REDLED, LOW);  // turn red LED off
  delay(1000);
}
```

But how long does printing take?

Timing: measuring time?

```
#define REDLED 13

void loop() {
  unsigned long startTime, endTime;

  startTime = micros();
  Serial.println("Starting loop\n");
  endTime = micros();
  Serial.print("Elapsed time to print:");
  Serial.println(endTime-startTime);

  digitalWrite(REDLED, HIGH); // turn red LED on
  delay(1000);
  digitalWrite(REDLED, LOW);  // turn red LED off
  delay(1000);
}
```

Your turn!

Write code that reads the distance sensor roughly every 250 ms and prints out the reading.

```
#define DISTSENSOR 14

void setup() {
  Serial.begin(9600); // set up Serial communication speed
  pinMode(DISTSENSOR, INPUT);  // distance sensor as input
}

void loop()
{
  int reading = analogRead(DISTSENSOR-14);
  Serial.print("Reading: "); Serial.println(reading);
  delay(250);
}
```

Your turn!

Write code that reads the distance sensor exactly every 250 ms and then prints out the readings.

```
#define DISTSENSOR 14
```

Reading Sensor Data

Write code that reads the distance sensor exactly every 250 ms.

```
void readDistData()
{
   unsigned long time;
   int i;

   time = millis(); // time at start of function in ms

   // record distance sensor data
   // sampling time = 250 ms (sampling rate = 4 bits/second)
   for (i=0; i<ARRAYSIZE; i++) {
     distData[i] = analogRead(DISTSENSOR-14); // read sensor
     while (millis() < (time + 250)); // wait until time done
     time += 250; // compute finish time for next sample
   }
}
```