Promoting robotic design and entrepreneurship experiences among students and teachers

Lesson 13:
Advanced Arduino Programming - II
• Arrays
• Functions

• **TASK/ACTIVITY:** Advanced programming with motors and functions
• An **array** is a collection of one type of variables that are accessed with an index number.

• **Declaring an array:**
  
  o **Syntax:** `data_type array_name[size] = value;`
  o **Value** and **size** may/may not be given at declaration

• **Examples:**

  1. `int myNums[6];`
  2. `int myInts[ ] = {2, 4, 6};`
Arrays in Arduino are zero indexed, i.e., array elements numbering starts from “0”

Array elements can be accessed with index as follows:

- Example: int myNums[4] = {2, 4, 6};
  c = myNums[2];
- Value of c is 6

We can assign a value to an array as follows:

- myNums[3] = 8;
ACTIVITY 1

1. Write a program to create an array and display all elements on the serial monitor

1. Create an array of size 6 with integers and display elements at index 2 and 4
ARRAYS – SOLUTION FOR 1St

```
int myIntVals[5]={1,2,3,4,5};

void setup() {
    Serial.begin(9600);
    for (int x = 0; x < 3; x++)
    {
        Serial.println(myIntVals[x]);
    }
}

void loop() {}
```

**OUTPUT:**
The values of the array are: 1 2 3 4 5
INDEXED ELEMENTS IN ARRAY | Arduino 1.8.19

File Edit Sketch Tools Help

Indexed_elements_in_Array

```cpp
int myIntVals[6] = {10, 20, 30, 40, 50, 60};

void setup() {
  Serial.begin(9600);
  Serial.println("The value of the 2nd indexed element is: ");
  Serial.println(myIntVals[2]);
  Serial.println("The value of the 4th indexed element is: ");
  Serial.println(myIntVals[4]);
}

void loop() {} 
```
ARRAYS – SOLUTION FOR 2\textsuperscript{nd}

\textbf{OUTPUT:}
The value of the 2nd indexed element is: 30
The value of the 4th indexed element is: 50
FUNCTIONS

What is a function?

• A function is a block of code either built-in or written by the user that allows structuring the code into one or more segments to perform individual tasks

Types of functions:

1. Predefined functions: Built-in functions available in Arduino environment such as println(), pinMode(), analogRead()

2. User-defined functions: Functions written by the programmers to run a specific task
STRUCTURE OF A FUNCTION

```
return_type function_name(parameters)
{
    // declarations;
    // statements;
}
```

- **return_type**: The data type of the value being returned by the function; If there is no return value, “void” can be used
- **function_name**: The name of the function; It is user-defined; It should not start with a digit but can contain letter, number or an underscore; Pre-defined function names cannot be used
- **parameters**: The values being passed to the function; These are used inside the code
- **declarations**: Declaring the variables for the function
- **statements**: The code written for a specific task to be performed
Example:

```c
int my_add_func(int a, int b)
{
    int sum; //declaration
    sum = a + b; //statements
    return sum;
}
```

- **return_type**: `int`
- **function_name**: `my_add_func`
- **parameters**: `int a, int b`
- **declarations**: `int sum`
- **statements**: `sum = a + b; return sum;`
```cpp
void hello() {
    Serial.print("Hello from function!\n");
    delay(2000);
}
```

**OUTPUT:**
Hello from function!
Hello again from main loop.
FUNCTION WITH ONE PARAMETER

```cpp
void hello(int repeat)
{
    for(int i=1; i<=repeat; i++)
    {
        Serial.print("Hello from function!\n");
        delay(500);
    }
}
```

**OUTPUT:**
Hello from function!
Hello from function!
Hello from function!
Hello from function!
Hello again from main loop.
ACTIVITY 2

• Write a program to display all even numbers from 1 to n where n is the parameter to be sent, i.e., $n = 100$
ACTIVITY 2 - SOLUTION

```cpp
void setup() {
  Serial.begin(9600);
  even_nums(100);
}

void loop() {}

void even_nums(int n){
  Serial.println("Even number from 1 to 100 are");
  for (int a=1; a<=n; a++){ // n=100
    if (a%2 == 0)
      Serial.println(a);
    delay(500);
  }
}
```

OUTPUT:
Even numbers from 1 to 100 are:
2
4
6
...
100
FUNCTION WITH TWO PARAMETERS

```java
void hello(int repeat, int dlyTim) {
    for (int i = 1; i <= repeat; i++) {
        Serial.print("Hello from function!\n");
        delay(dlyTim);
    }
}
```

**OUTPUT:**
Hello from function!
Hello from function!
Hello from function!
Hello from function!
Hello again from main loop.
FUNCTION WITH RETURN PARAMETER

```java
int adder(int aa, int bb){
    int cc = aa + bb;
    return cc;
}
```

**OUTPUT:**
50
Promoting Robotic Design and Entrepreneurship Experiences Among Students and Teachers

Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, NYU Tandon School of Engineering, July 2017

FUNCTION WITH NO RETURN PARAMETERS

```c
void loop() {
    // pick some random numbers
    x = random(10);
    y = random(10);
    Serial.print("Values of x and y before swapping: ");
    Serial.print(x);
    Serial.print(',');
    Serial.println(y);
    swap();
    Serial.print("\nValues of x and y after swapping: ");
    Serial.print(x);
    Serial.print(',');
    Serial.println(y);
    delay(1000);
}
```
FUNCTION WITH NO RETURN PARAMETERS

```c
void swap(){
    int temp;
    temp = x;
    x = y;
    y = temp;
}
```

**OUTPUT:**
Values of x and y before swap 7,9
Values of x and y after swap 9,7
Task / Activity: Advanced Programming
ACTIVITY 3

Motor Direction Control using Functions

Write a program to make the robot’s wheels rotate clockwise and rotate anticlockwise using functions

1. Which rotation makes the robot move forward?

1. Which rotation makes the robot move backward?
ACTIVITY 3 SOLUTION

- Connect a DC motor with L293D IC (like you have previously done)
  
  **Setup function**

```cpp
int motor1Pin1 = 3; // pin 2 on L293D
int motor1Pin2 = 4; // pin 7 on L293D
int enablePin = 9; // pin 1 on L293D

void setup() {
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(enablePin, OUTPUT);
    digitalWrite(enablePin, HIGH);
}
```
ACTIVITY 3 SOLUTION

Loop function

```c
void loop() {
clockwise_rotate();
delay(2000);
anticlockwise_rotate();
delay(2000);
}
```
Function definition

define function clockwise_rotation:
define function anticlockwise_rotation:

```c
void clockwise_rotate()
{
        digitalWrite(motor1Pin1, LOW);  // set pin 2 on L293D LOW
        digitalWrite(motor1Pin2, HIGH);  // set pin 7 on L293D HIGH
}

void anticlockwise_rotate()
{
        digitalWrite(motor1Pin2, LOW);  // set pin 7 on L293D LOW
        digitalWrite(motor1Pin1, HIGH);  // set pin 2 on L293D HIGH
}
```
Video: Forward movement
Video: Backward movement
VEX CLAWBOT
Promoting Robotic Design and Entrepreneurship Experiences Among Students and Teachers
Innovative Technology Experiences for Students and Teachers (ITEST), Professional Development Program, NYU Tandon School of Engineering, July 2017

VEX CLAWBOT

Video
Thank You!

Questions and Feedback?