

# Lab1: Robot Basic

## Pre-Requisite Knowledge and Skills

1. Understand the basics of computer.
2. Be able to use computer and write a code in C.

## Learning Objective:

1. Understand the structure of robot programming.
2. Be able to write and upload a code to robot.

## Recommended Running Environment and Software:

1. Computers Running Windows OS, OSX, or Linux
2. SparkiDuino IDE

## Instructional Material:

1. Sparki Robot
2. Instructions of this activity
3. Sample codes

## Video Demonstration:

1. to be developed

## Lab Assessment:

1. Exercises
2. Quiz

## Lab Instructions

The Saprki is pre-assembled and packed with more than 12 sensors. Since it is based on Arduino, programming Sparki follows the programming in Arduino, which is based on C language. Sparki provides the programming interface, called “SparkiDuino” as shown in Figure 1, for user to write a code.

**Step 1:** Go to this link “<http://arcbotics.com/lessons/sparkiduino-install-choose-os/>” and select a OS to download and install “SparkiDuino” IDE.



**Figure 1.** SparkiDuino interface and basic code structure

Sparki also provides a set of functions included in header file “sparki.h”. It should be included in the code to use the functions. The code should be structured in two part, setup and loop. The setup will be running once while loop will be continuously repeating. The basic structure of the Sparki robot programming is shown in Figure 2.

```
#include <Sparki.h> // include the sparki library

void setup() // code inside these brackets runs first, and only once
{
}

void loop() // code inside these brackets runs over and over forever
{
}
```

**Figure 2.** Basic code structure

Initial parameters and conditions can be written in “setup()” function such as aligning the sensors, initializing the motors, or etc. The code in the “loop()” function will be kept executing over and over as long as robot is turned on. Therefore, the repeating commands should be included in this function.

Sparki programming is based on C and we will follow the example to learn how to write and upload a code to Sparki.

**Step 2:** Open “SparkiDuino” IDE and type following code in the SparkiDuino.

```
#include <Sparki.h>      // include the sparki library

void setup()
{
}

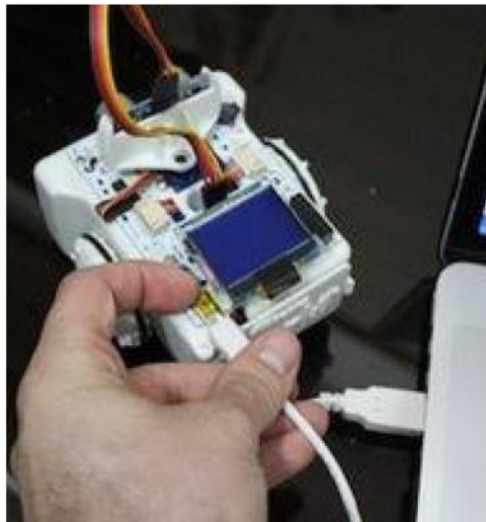
void loop()
{
  int i;
  for (i = 0; i < 3; i=i+1)
  {
    sparki.beep();      // Sparki beeps!
    delay(300);        // wait 0.3 seconds
  }

  delay(2000);         // wait two seconds (2000 milliseconds)
}
```

**Figure 3.** Code example

Once code is written in this SparkiDuino, code should be compiled and uploaded to Sparki robot. To compile and upload the code, Sparki has to be connected to PC where the interface is running.

**Step 3:** Use included cable to connect Sparki to PC as shown in Figure 4.



**Figure 4.** Connect Sparki to PC

**Step 4:** Configure the port in SparkiDuino. Go to “Tool > Serial port” and select the port that has been assigned by OS to Sparki.

**Step 5:** Compile the written code – Click the check mark icon (i.e., ✓) and code will be compile. During compiling, any syntax error in the code will be check and identified. However, logical error will not be detected.

**Step 6:** Now click arrow icon (i.e., ↗) to upload a code. Then, upload process will begin and red and green lights will be flashing. When the code is successfully uploaded, red light will be turned off and only green light will be on. And SparkiDuino will show “Done uploading” as shown in Figure 5.



**Figure 5.** Successful uploading

**Step 6:** Disconnect the cable from Sparki and Turn on the switch. You will have to wait some time for Sparki to initialize and run. Then, observe what Sparki does.

### Discussion

- **What will make beeping more?**

Change the parameters in for loop. For example, for 5 beeping, for loop should look like “for (i=0; i<5; i++)”

- **How to create rhythm using beeping?**

Add more beep and use several different numbers in delay functions.