

Infusing Cyber-Physical Systems Concepts into Computer Science Curricular Module: Simple Model

Release 0.1

May 6, 2013

Copyright© 2013. The SysNet Group. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation. A copy of the license can be found at <http://www.gnu.org/licenses/fdl.html>.
Contact: huichen@ieee.org

1 Overview

You will revise a given program to control a robotic vehicle to follow a circular track of a certain size repeatedly. From the experiments, you will learn to use a simple modeling technique to solve the problem. Besides embedded control using computer programs and solution evaluation to a technical problem, you are to construct a mathematical model using experimental data and apply the model to solve the given problem.

1.1 Prerequisite

We recommend that you complete the “Trial and Error” and “Binary Search” modules before you start this module. In the “Trial and Error” module you can obtain experience in programming a Boe-Bot and experimenting with program *circlerun.bs2*. In the “Binary Search” module you can gain knowledge of algorithm development and algorithm efficiency. However, it does not preclude that you treat this module as a standalone one without trying the “Trial and Error” module and the “Binary Search” module. We refer you to the “Trial and Error” module for a basic introduction to programming Boe-Bot and program *circlerun.bs2*.

1.2 Lab Requirement

- A *Boe-Bot* robot.
- A PC with the *BASIC Stamp Editor* software installed
- A large circular track and a small circular track
- Program *circlerun.bs2*
- Gridded paper¹

1.3 Programming Boe-Bot

See the “Trial and Error” module.

¹You may print the gridded paper PDF file from <http://sysnetgrp.net/cpsedu/>.

1.4 Running Circles

We introduce Program *circlerun.bs2* in the “Trial and Error” module and use it also in the “Binary Search” module. We will use this program in this module as well. For convenience, we list the program in program listing 1.

Listing 1: Program *circlerun.bs2*

```

1  ' {$STAMP BS2}
2  ' {$PBASIC 2.5}
3
4  DEBUG "Program Running"
5  DO
6      PULSOUT 13, 850
7      PULSOUT 12, 750
8      PAUSE 20
9  LOOP

```

2 Tasks

You are to solve the same problem in the “Trial and Error” module and the “Binary Search” module, i.e., to revise program *circlerun.bs2* so that the robot will travel endlessly following a given circular track whose radius is different from those obtained from previous experiments. Figure 1 demonstrates a circular track and the robot must run within the circular track. As discussed in the “Trial and Error” module and the “Binary Search” module, the problem reduces to the finding of an appropriate value of the pulse width in Line 7 of program *circlerun.bs2*, i.e., the second parameter in Line 7 that controls the revolution speed of the servo controlled by pin 12. However, different from the “Trial and Error” module and the “Binary Search” module, you are to use a new method built upon a simple mathematical model that characterizes the robotic vehicle’s movements. For convenience, we call this method, a “Simple Model” method.

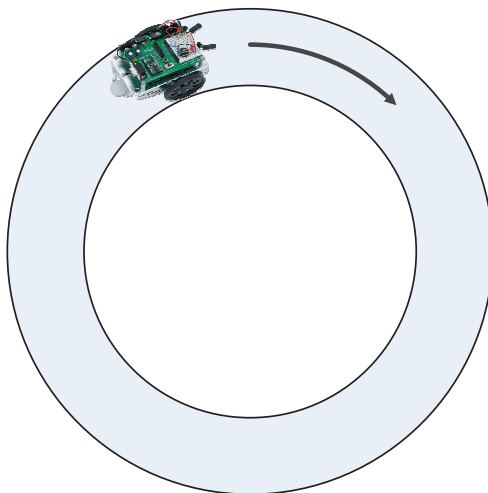


Figure 1: The robot runs within a circular track.

2.1 Task 1: Simple Model

In the “Trial and Error” module and the “Binary Search” module, you have observed that different values of the pulse width at Line 7 of program *circlerun.bs2* lead to circular paths of different radius. Can we use the experience obtained from past experiments to find a sufficiently good pulse width value when you want to guide the robotic vehicle to follow a new circular track of different radius? The answer is *Yes!* In this task, you are to construct a mathematical model that describes the relationship between the radius of circular paths along which the robotic vehicle travels and the value of the pulse width.

The model in effect is a mathematical function, $r : X \rightarrow R$, or $r = r(x)$ where $x \in X = [650, 750]$ is the value of the pulse and $r \in R = [0, \infty]$ represents the radius of the circular path. Figure 2 is an example of the graph of the simple model obtained from experiments. Table 1 lists the values of the pulse width in Line 7 of program *circlerun.bs2* versus the corresponding radius of the resulting circular paths along which the robotic vehicle travels. Figure 2 is the corresponding graph of the two columns in Table 2. Table 1 and Figure 2 characterizes how the radius of the resulting circular paths changes with the value of the pulse width. Given a circular track of a certain radius, you only need to look up what the corresponding value of the pulse width is.

Pulse width (μs)	Radius (inch)
714	18.0
716	14.3
720	11.0
722	9.8
725	8.0
727	5.4

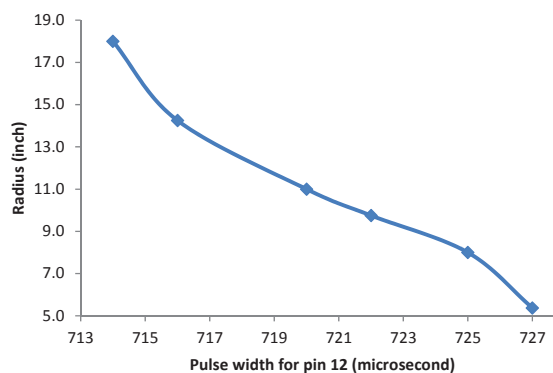


Table 1: Pulse width of pin 12 (Line 7) and radius of circular path

Figure 2: Function of parameter value and radius of circular track

You are to follow the steps below to construct a model similar to Table 1 and Figure 2 through a few experiments. Your task is to fill up Table 2 and graph the results in the table using a paper, pencil and a ruler, using Microsoft Excel, or using any other graphing software you are familiar with or you would like to learn.

1. Connect the Boe-Bot to the PC.
2. Launch the BASIC Stamp Editor from the PC
3. Input program *circlerun.bs2* as shown in Program Listing 1 to the Basic Stamp Editor.
4. Choose a pulse with value from Table 2. Replace the pulse width value with the selected one.
5. Click the “Run” menu entry form the “Run Menu”.

If the 3-position switch on the Boe-Bot is at the center position (i.e., position 1), the Boe-Bot should be running and its trajectory is the smallest circular path since only one wheel is turning. An individual can choose to disconnect the Boe-Bot from the PC to allow it runs more freely.

You shall measure the radius of the circular path along which the vehicle travels. Write down your measure in Table 2.

6. Repeat Steps 4–5 to complete Table 2

Pulse width (μs)	Radius (inch)
735	
730	
725	
720	
715	

Table 2: Relationship of the pulse width of pin 12 and radius of the resulting circular paths

- Graph the results in Table 2 using provided gridded paper. You may use software such as Microsoft[®] Excel or OriginLab[®] Origins to graph the results. In your graph, treat the pulse width as x -axis and the radius of the resulting circular paths as y -axis.

2.2 Task 2: Following New Circular Track

Your task is to complete Table 3. You will use the simple model constructed in Task 1, i.e., Table 2 to find the value of the pulse width for each radius of a given circular track along which the robotic vehicle should follow. You will then replace the value of the pulse width in Line 7 of program *circlerun.bs2* with the value you find and then run the robotic vehicle. Having measured the radius of the resulting circular path, you will calculate the absolute error and the relative error between the resulting circular path and the radius of the circular track.

Radius of Circular Track (inch)	Pulse width (μs)	Observed Radius (inch)	Absolute Error	Relative Error
13				
9				

Table 3: Application of the simple model

3 Submission

The instructor requires you to submit the following items,

- A brief recitation of the problem description and the technical solution;

- Table 2 and Table 3 that you have filled up.
- The answers to the two evaluation questions, 1) how accurate is the circle? 2) and how many trials have been conducted? For question 2, what if you are given 10 different circular tracks along which the robotic vehicle must follow? You should also compare the answers to the two questions to those of the “Trial and Error” method and the “Binary Search” method.