

**BE1010 – Fall 2004**

**Prof. Mohamad Hassoun, WSU**

**Quiz # 9** (25 minutes. Closed book/webpage/notes)

- I. (3 points) Consider the following NQC instructions. What does each instruction do when executed by the RCX?

`CreateDatalog (0)`; Clears the datalog in the RCX.

`SendMessage (13)`; Sends an infrared message with a value of 13.

`Message ()`; Retrieves the most recently received message.

- II. (4 points) Assume that the NQC program shown below has been loaded into your RCX. Next, the RCX is switched off and then on (but no program is run, yet). Now, assume that you have pointed your remote toward the RCX and pressed the “message #3” button. What happens when you run the program? Explain.

```
task main()
{
  while(true)
  {
    until(Message() != 2);
    if(Message() ==1) {PlaySound(SOUND_UP); }
    if(Message() ==2) {PlaySound(SOUND_DOWN); }
    if(Message() ==3) {PlaySound(SOUND_DOUBLE_BEEP); }
    Wait(50);
  }
}
```

When the message #3 button is pressed on the remote, the RCX stores it in the buffer as the last message received. Now, once the program runs, it retrieves the last message with the `Message()` command and checks if it is not equal to 2. Since it is 3 (which  $\neq$  2) it checks if it is one of the values for which a sound command is given. It will then play the double beep sound wait  $\frac{1}{2}$  second and play it again and again until a different message value is received.

- III. (3 points) Your RCX has just executed the following program. Assume you use the Brick Command Center **datalog utility** to upload the datalog from the RCX to your PC. Write down the datalog value(s) you would see in the datalog window.

```
task main()
{
  CreateDatalog(1);
  SetWatch(12, 48);
  Wait(13000);
  AddToDatalog(Watch() );
}
```

The program sets the internal clock to 12:48 and then runs for 130 seconds (2min 10 seconds or until the clock reads 12:50) and saves the value to the datalog. The datalog only displays values in minutes so it will read 770 .  
(12hr x 60min/h + 50 min = 770 min)