

CS12020

Arduino Serial Networking Project Assignment

Documentation

Muntazir Rashid – mrm19

17th December 2021

Introduction:

This assignment aims to program an **Arduino** with the **Software Serial Shield** to analyse the transmission of the messages sent and received through software serial ports connected to a virtual network of Hosts running on the connected **Nano**.

This project relies on **Aberystwyth Software Serial Protocol (ASSP)** to listen for messages on more than 1 Port. **ASSP** implements a reliable communication mechanism by ensuring a receiving Arduino is listening before a message is sent. It also allows the names of the destination and sender to be included in the message.

The overall aim of the assignment is to determine which Hosts are present on what ports and analyse the network's performance.

*"Also need to set the **Arduino Nano** to emulate a small network which is achieved by calling 'SerialShield.setEmuMode(6)' from setup function."*

Analysis:

Assignment consists of 4 Tasks divided into more minor problems.

Task 1: Does the port connect to that Host?

An array definition with the list of 26 Host Names was provided (i.e., **names**).

This task aims to check which Host was connected to **Port 4**.

To tackle this problem, we were advised to create a '**Boolean**' function named **"isHostOnPort"** which send a '**ping**' message to the particular Host on the specified Port and returns '**true**' if the host is reachable and '**false**' if otherwise. To receive the ping response '**fetchASSPMessage**' is looped to try **50** times until it returns '**true**' and if it doesn't return '**true**' within 50 times the Host is not connected to the Port.

For testing **"isHostOnPort"** a '**void**' function called **task1()** is written which goes through all the Hosts in the '**names array**' and prints whether the corresponding Host is connected on the Port 4 or not (as directed in the Task).

Task 2: Timing Port 2.

This task aims to return the time it takes for each Host connected on Port 2 to respond.

To achieve this an '**unsigned long**' function named **"getTimeToHost"** is created which capitalizes on **"isHostOnPort"** function and the built in Arduino timer '**millis()**'.

To test this a '**void**' function named **task2()** is created which goes through all the Hosts in the '**names array**' and prints the respond time if they are connected to **Port 2** and **NC** if not.

Task 3: Mapping the Network.

This task aims to return each Host connected to **Ports 1-4** and **NC** (Not Connected) for Hosts not connected to any port (i.e., **0**).

To get the Port number of every Host an *'int'* function named **"getPortForHost"** is created which returns the **Port Number (1-4)** or **0** if Host is not connected to any port at all. A *'void'* function named **"Details ()"** is created which gets the details (Port Number, Name and Time Taken) of all the Hosts and saves it in an ***'Array of Struct (HostDetail)'*** named **"hostDetails_list"**.

For testing and to get the data in desired format i.e., **1-4 Port Numbers** and **NC** for Hosts not connected at all, a *'void'* function **"task3()"** is written which loops through all the Hosts stored in **"hostDetails_list"** and prints the result in required format i.e., to what Port each Host is connected to.

***I used struct "HostDetail" to collect all the Host details to increase the efficiency of the program.**

Task 4: Network response time analysis.

Aim of this task is to analyse the response time of hosts on different Ports.

Same ***'Details()'*** function created during Task 3 is used and the details obtained are looped to calculate the number of Hosts on each Port, fastest and slowest Host on each Port and to print the details in required format.

***I utilised the struct created in task3() for efficiency.**

***I couldn't figure out How to print the Name of fastest and Slowest Hosts on each Port.**

Problems Encountered:

Task 1 and Task 2 were straight forward and went smooth.

Getting Task 3 results was not that hard but **formatting** the Task 3 results as described in the assignment was complicated and required a lot of time. First, I came up with an idea to use loops to format it in the desired way, but this was extremely inefficient as it had to recollect the data every single time during the loop and took a lot of time to print the result. After that I created a struct named **"HostDetail"** which collected all the details required to get the result and used the loops to access the details from **"HostDetail"** instead of recalculating it again and again.

Same struct **"HostDetail"** was also used in Task 4 to collect the time it took for each host on a particular Port to respond.

***I couldn't figure out How to get the Name of fastest and Slowest Hosts in Task 4.**

Your assessment of the mark you would award yourself for the work (based on the assessment criteria detailed):

1st: 70-79%.

I believe that the written components are professionally presented in both layout on the page and logical structure with a high grammatical standard. Each Task is implemented using appropriate technology and will at least completely fulfil the functional requirements.

Output

Test Outputs:

Task 1:

```
Software serial LED test program
HOST Port 4?
Ann NO
Bob NO
Cat NO
Del NO
Eva NO
Fey NO
Software serial LED test program
HOST Port 4?
Ann NO
Bob NO
Cat NO
Del NO
Eva NO
Fey NO
ZÿjGSoftware serial LED test program
HOST Port 4?
Ann NO
Bob NO
Cat NO
Del NO
Eva NO
Fey NO
Gil NO
Han NO
Ivy NO
Jem YES
Ken YES
Leo NO
Mae NO
Ned NO
Obe NO
Pam NO
```

☒ Autoscroll ☐ Show timestamp

Newline

57600 baud

Task 2:

```
u@`ZÿSoftware serial LED test program
HOST Port 2?
Ann NO
Bob NO
Cat NO
Del NO
Eva NO
Fey NO
Gil NO
Han NO
Ivy NO
Jem NO
Ken NO
Leo NO
Mae NO
Ned NO
Obe YES
Pam NO
HOST Port2Time
Ann NC
Bob NC
Cat NC
Del NC
Eva NC
Fey NC
Gil NC
Han NC
Ivy NC
Jem NC
Ken NC
Leo NC
Mae NC
Ned NC
Obe 1724
Pam NC
```

☒ Autoscroll ☐ Show timestamp

Newline

57600 baud

Task 3:

Port	Hosts
NC	Ann
4	Bob
NC	Cat
1	Del
2	Eva
3	Fey
2	Gil
NC	Han
3	Ivy
4	Jem

Results before Formatting

Task 4:

/dev/cu.usbmodem14301

Fetching Details...
Fetching Details...
Complete

Port	Hosts	Slowest	Time
1	2		
2	2		
3	1		
4	0		
Port	Hosts	Fastest	Time
1	2		
2	1		
3	1		
4	0		

☒ Autoscroll ☐ Show timestamp

Newline ⬆⬇⬇⬆

 57600 b

//Actual Formatted Output is Below

Desired Output:

Task 1:

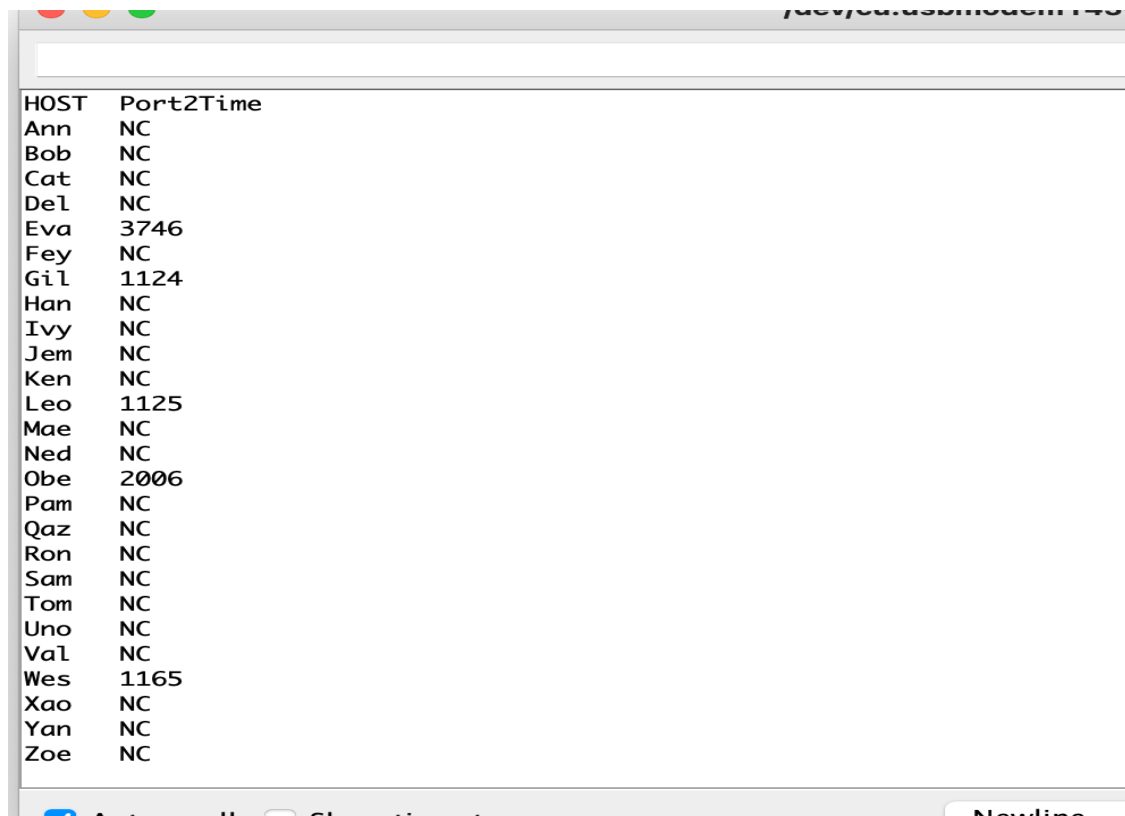


A terminal window titled `/dev/cu.usbmodem143` showing the output of a command. The output is a list of names and their corresponding port status. The names are: Ann, Bob, Cat, Del, Eva, Fey, Gil, Han, Ivy, Jem, Ken, Leo, Mae, Ned, Obe, Pam, Qaz, Ron, Sam, Tom, Uno, Val, Wes, Xao, Yan, and Zoe. The port status is either "NO" or "YES".

HOST	Port 4?
Ann	NO
Bob	YES
Cat	NO
Del	NO
Eva	NO
Fey	NO
Gil	NO
Han	NO
Ivy	NO
Jem	YES
Ken	YES
Leo	NO
Mae	NO
Ned	NO
Obe	NO
Pam	NO
Qaz	NO
Ron	NO
Sam	YES
Tom	NO
Uno	NO
Val	NO
Wes	NO
Xao	NO
Yan	NO
Zoe	YES

The terminal window has a status bar at the bottom with the following options: ☒ Autoscroll, ☐ Show timestamp, and ☐ Newline.

Task 2:



A terminal window titled `/dev/cu.usbmodem143` showing the output of a command. The output is a list of names and their corresponding port status. The names are: Ann, Bob, Cat, Del, Eva, Fey, Gil, Han, Ivy, Jem, Ken, Leo, Mae, Ned, Obe, Pam, Qaz, Ron, Sam, Tom, Uno, Val, Wes, Xao, Yan, and Zoe. The port status is either "NC" or a number.

HOST	Port2Time
Ann	NC
Bob	NC
Cat	NC
Del	NC
Eva	3746
Fey	NC
Gil	1124
Han	NC
Ivy	NC
Jem	NC
Ken	NC
Leo	1125
Mae	NC
Ned	NC
Obe	2006
Pam	NC
Qaz	NC
Ron	NC
Sam	NC
Tom	NC
Uno	NC
Val	NC
Wes	1165
Xao	NC
Yan	NC
Zoe	NC

The terminal window has a status bar at the bottom with the following options: ☒ Autoscroll, ☐ Show timestamp, and ☐ Newline.

Task 3:

```

/dev/cu.usbmodem

```

```

Fetching Details...
6 Completed out of 26
12 Completed out of 26
18 Completed out of 26
22 Completed out of 26
Complete
Port  Hosts
1     Ann Del Pam Tom
2     Gil Leo Obe Val
3     Fey Ivy Mae Ned Ron Xiao
4     Bob Ken Sam Zoe
NC    Cat Eva Han Jem Qaz Uno Wes Yan

```

```

//
printing
progress
message
//

```

```

☒ Autoscroll ☐ Show timestamp Newl

```

Task 4:

```

/dev/cu.usbmodem14101

```

```

Fetching Details...
6 Completed of of 26
12 Completed of of 26
18 Completed of of 26
22 Completed of of 26
Complete
Port  Hosts      Slowest      Time
1     4         Name        1401
2     6         Name        3799
3     6         Name        3239
4     3         Name        2795
Port  Hosts      Fastest      Time
1     4         Name        824
2     6         Name        803
3     6         Name        904
4     3         Name        1155

```

```

☒ Autoscroll ☐ Show timestamp Newline
/*Wasn't Able to figure out how to get Names of Fastest and Slowest Hosts*/

```