829H1 Real-Time Embedded Systems Exercise 5

Candidate No: 105936

12th January 2019

Contents

1. Introduction

This report focuses on using Ethernet communications on the board and shows the work completed during the laboratory sessions and what was learnt. Code listings for some of the created programs can be found in section 3.

2. Experiments

2.1. Connect 2 mbed boards via Ethernet

2.1.1. Exercise 1

This was completed using the code provided however did not work at the first attempt this was due to a faulty board. Once the board was replaced it worked as expected. Figure 1 and Figure 2 shows the outputs from the application once the new board was substituted.

2.2. Exercise 2

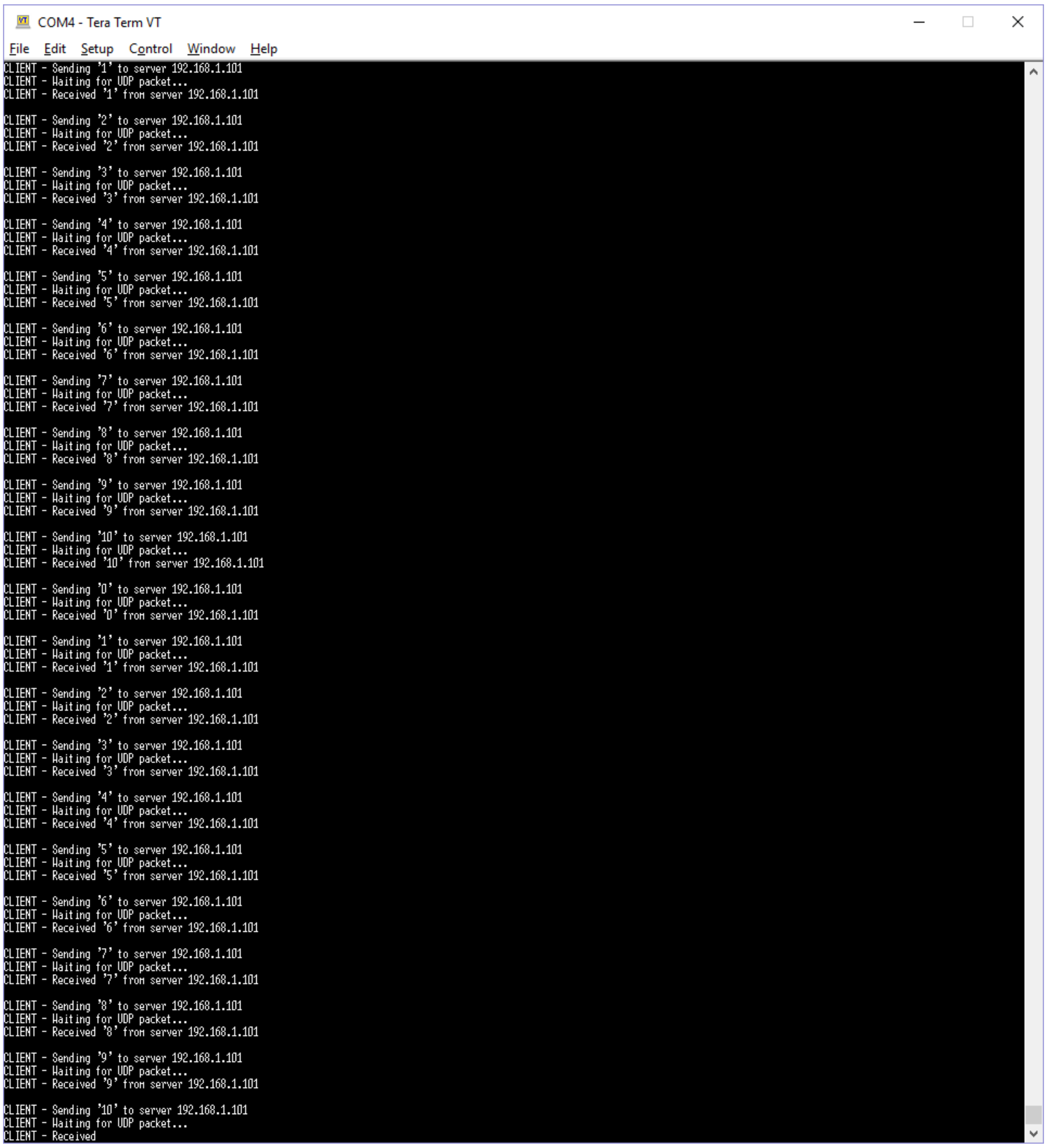
This involved me attempting to send the accelerometer data from one board to the other and vice versa. However the first problem I hit was that the sockets library we are using does not support the sending of floats. which is not too surprising however after this I attempted to convert the float into a char array so I could send that over the Ethernet cable however whenever I attempted to read that char array I had weird values in it. Unfortunately I ran out of time due to the problem earlier meaning that I was unable to get this working. The code for this can be found in section A.1 and section A.2 also fig. 3 and fig. 4 shows the gibberish output I was talking about.

2.3. Connecting to IBMs Internet of Things

I had to use the program at `https://os.mbed.com/users/steveshun/code/IoTClientEthernet/` to allow me to manually set the MAC address of the board allowing me to run it over the University's network. After that it was quite simple to work and I got the image shown in fig. 5 which shows the values of the sensors on the board.

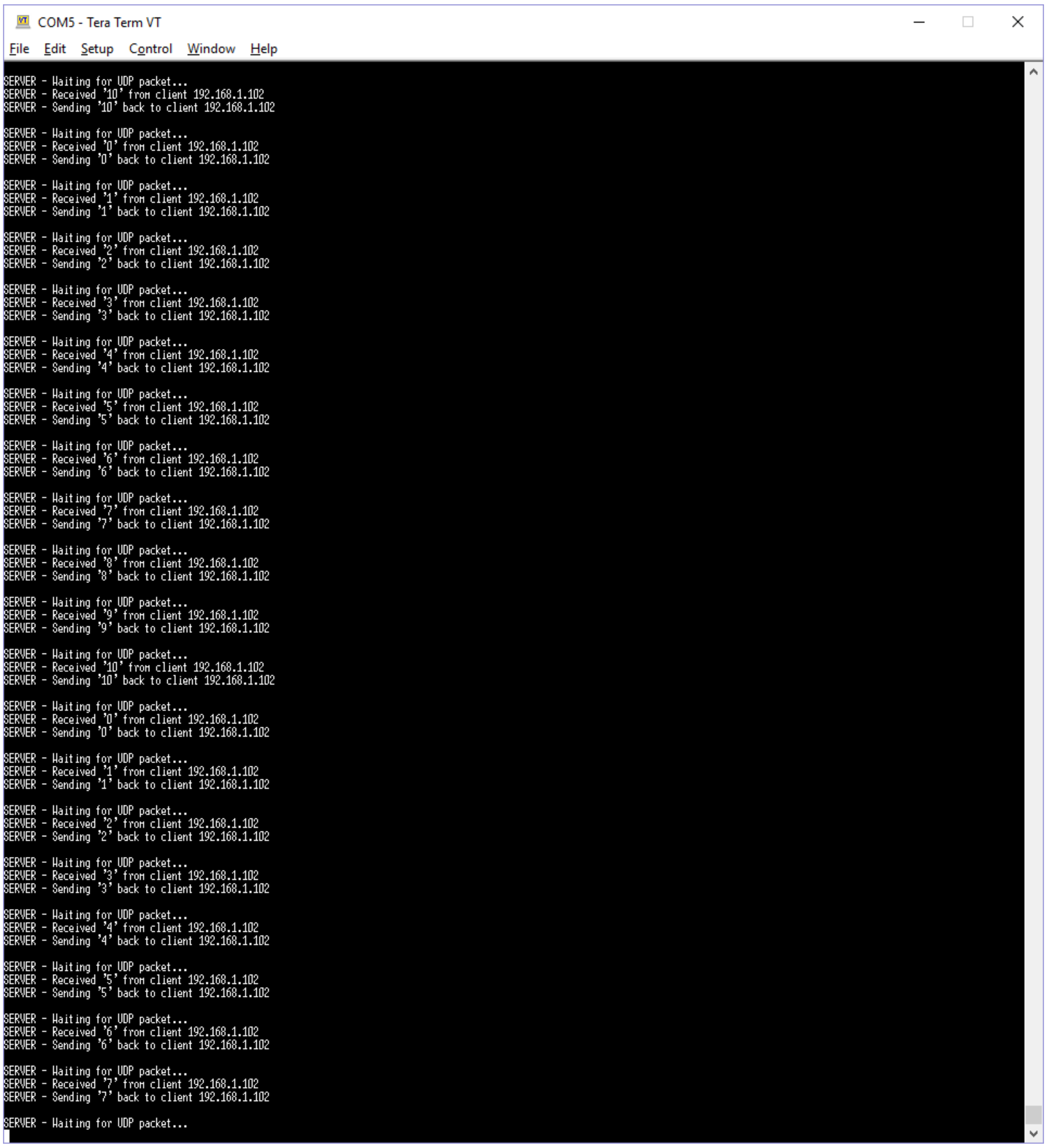
3. Conclusion

In conclusion, this was a useful exercise on using Ethernet communications and then how to use and set-up the device as an Internet of things devices. This served as a useful start for the project I plan on completing over the winter.



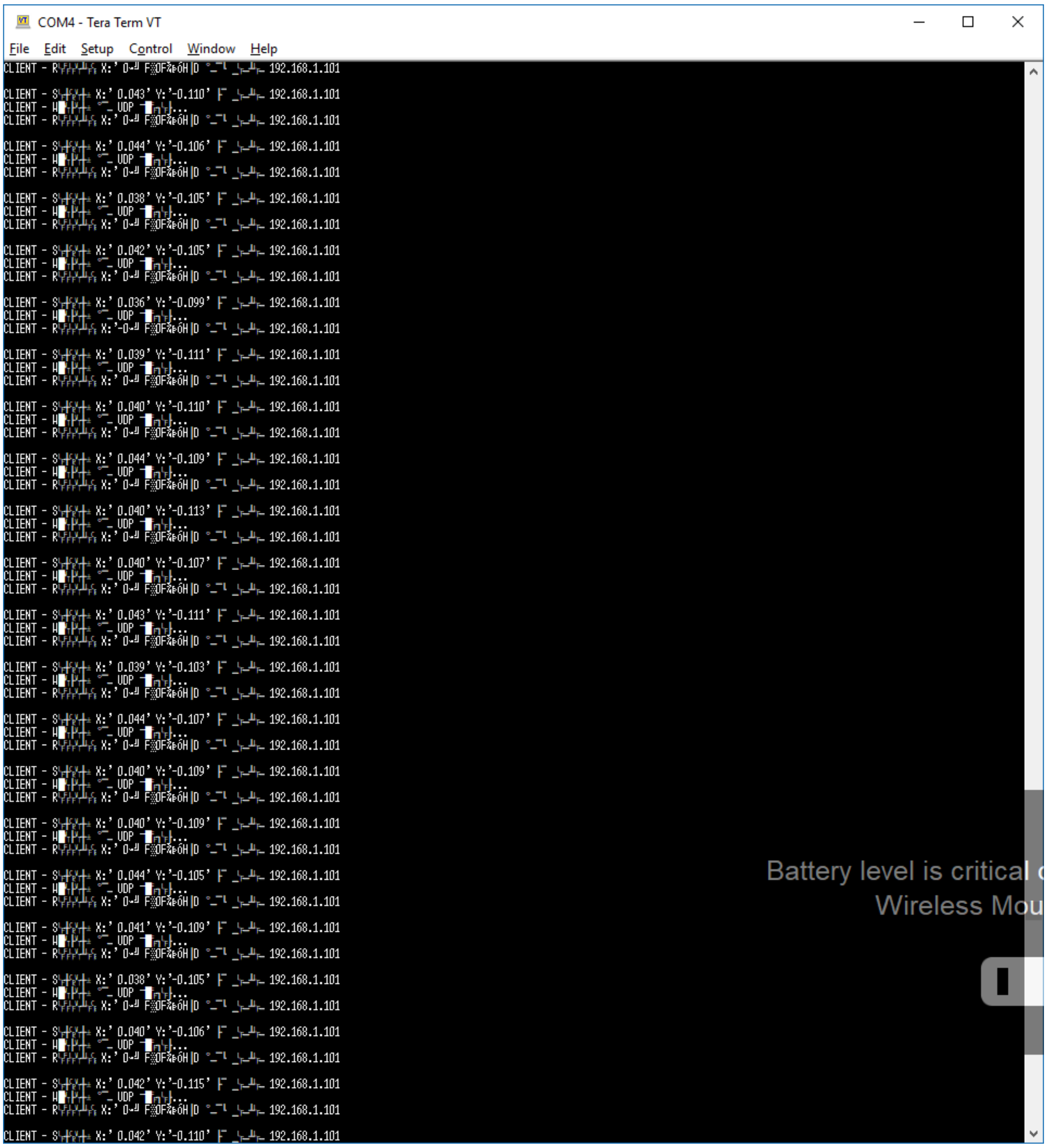
```
CLIENT - Sending '1' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '1' from server 192.168.1.101

CLIENT - Sending '2' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '2' from server 192.168.1.101

CLIENT - Sending '3' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '3' from server 192.168.1.101

CLIENT - Sending '4' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '4' from server 192.168.1.101

CLIENT - Sending '5' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '5' from server 192.168.1.101

CLIENT - Sending '6' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '6' from server 192.168.1.101

CLIENT - Sending '7' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '7' from server 192.168.1.101

CLIENT - Sending '8' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '8' from server 192.168.1.101

CLIENT - Sending '9' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '9' from server 192.168.1.101

CLIENT - Sending '10' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '10' from server 192.168.1.101

CLIENT - Sending '0' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '0' from server 192.168.1.101

CLIENT - Sending '1' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '1' from server 192.168.1.101

CLIENT - Sending '2' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '2' from server 192.168.1.101

CLIENT - Sending '3' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '3' from server 192.168.1.101

CLIENT - Sending '4' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '4' from server 192.168.1.101

CLIENT - Sending '5' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '5' from server 192.168.1.101

CLIENT - Sending '6' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '6' from server 192.168.1.101

CLIENT - Sending '7' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '7' from server 192.168.1.101

CLIENT - Sending '8' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '8' from server 192.168.1.101

CLIENT - Sending '9' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received '9' from server 192.168.1.101

CLIENT - Sending '10' to server 192.168.1.101
CLIENT - Waiting for UDP packet...
CLIENT - Received
```

Figure 1: shows the tera term output from the client application



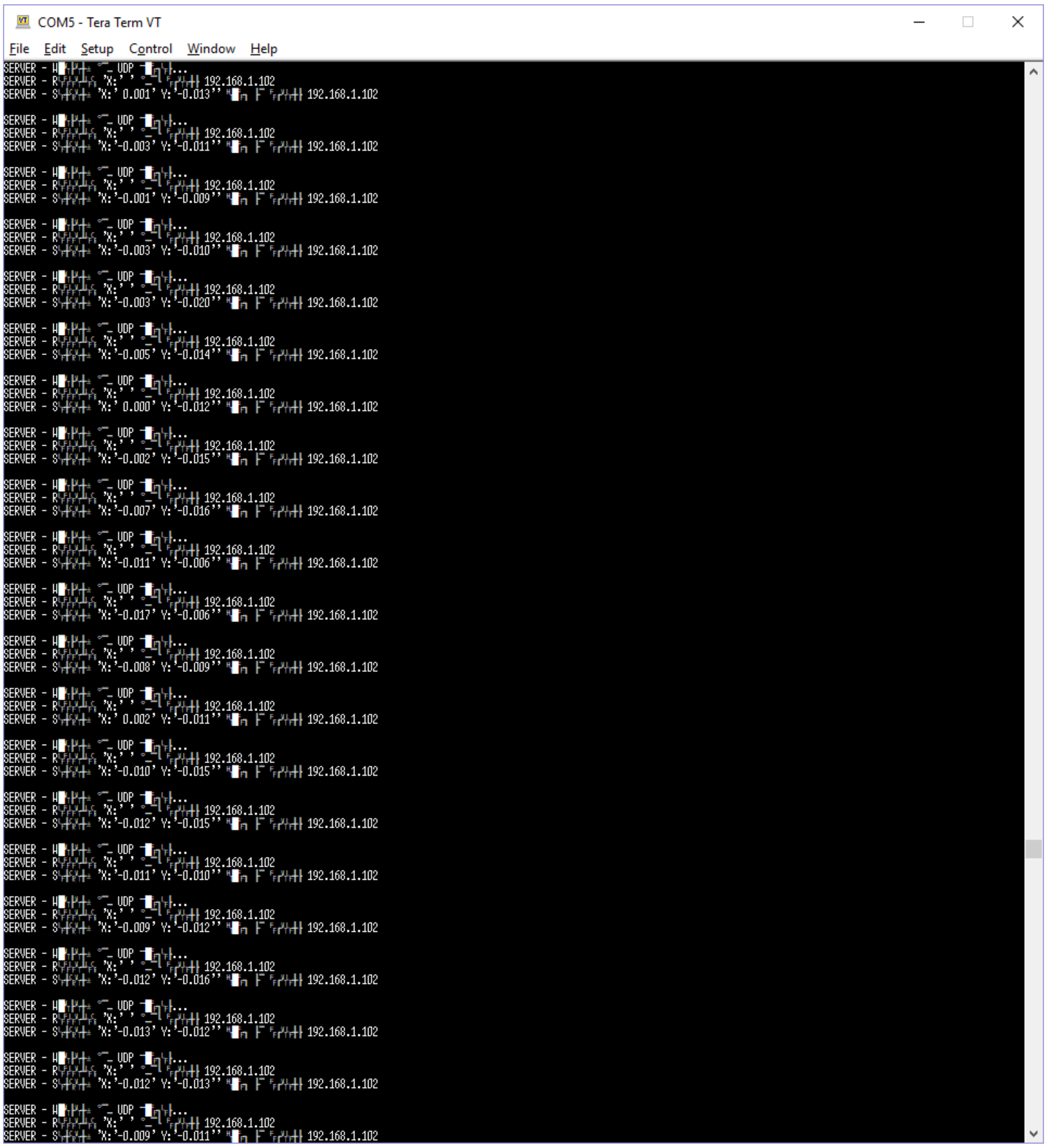
```
SERVER - Waiting for UDP packet...
SERVER - Received '10' from client 192.168.1.102
SERVER - Sending '10' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '0' from client 192.168.1.102
SERVER - Sending '0' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '1' from client 192.168.1.102
SERVER - Sending '1' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '2' from client 192.168.1.102
SERVER - Sending '2' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '3' from client 192.168.1.102
SERVER - Sending '3' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '4' from client 192.168.1.102
SERVER - Sending '4' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '5' from client 192.168.1.102
SERVER - Sending '5' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '6' from client 192.168.1.102
SERVER - Sending '6' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '7' from client 192.168.1.102
SERVER - Sending '7' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '8' from client 192.168.1.102
SERVER - Sending '8' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '9' from client 192.168.1.102
SERVER - Sending '9' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '10' from client 192.168.1.102
SERVER - Sending '10' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '0' from client 192.168.1.102
SERVER - Sending '0' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '1' from client 192.168.1.102
SERVER - Sending '1' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '2' from client 192.168.1.102
SERVER - Sending '2' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '3' from client 192.168.1.102
SERVER - Sending '3' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '4' from client 192.168.1.102
SERVER - Sending '4' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '5' from client 192.168.1.102
SERVER - Sending '5' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '6' from client 192.168.1.102
SERVER - Sending '6' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
SERVER - Received '7' from client 192.168.1.102
SERVER - Sending '7' back to client 192.168.1.102

SERVER - Waiting for UDP packet...
```

Figure 2: shows the tera term output from the server application


COM4 - Tera Term VT
File Edit Setup Control Window Help
Battery level is critical
Wireless Mou

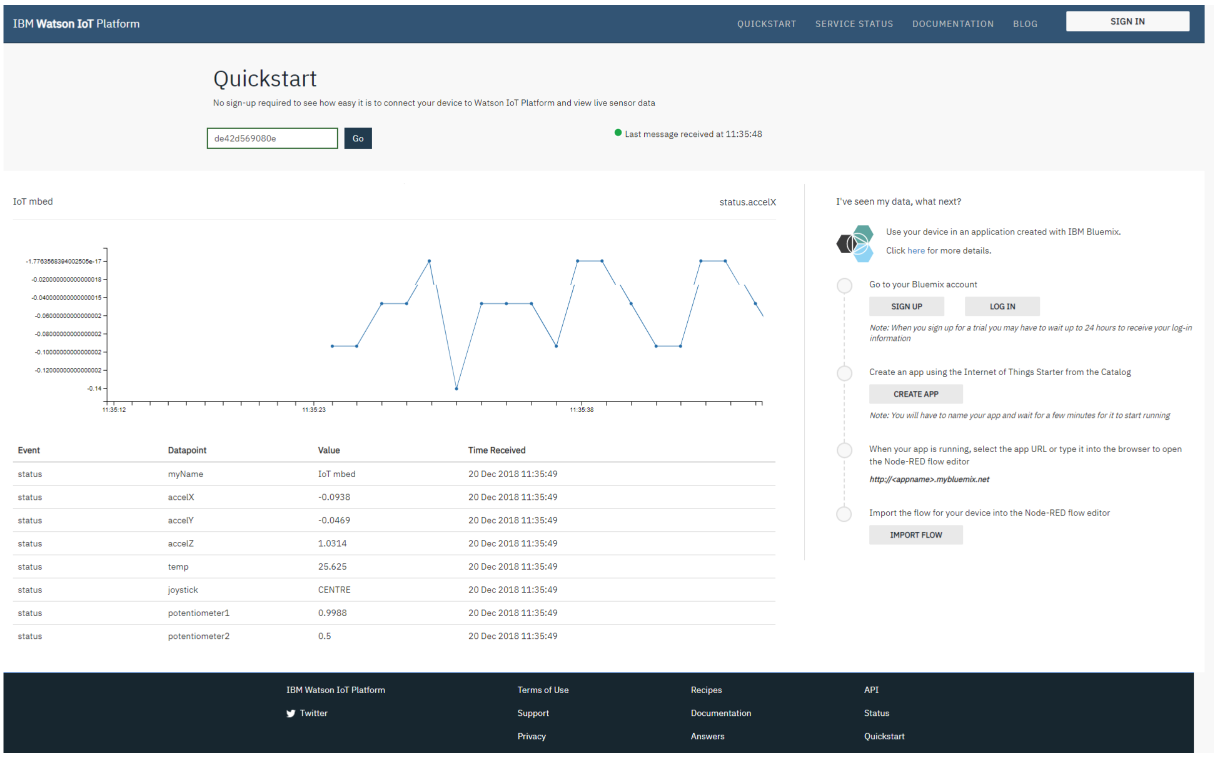
```
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.043' Y:'-0.110' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.044' Y:'-0.106' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.038' Y:'-0.105' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.042' Y:'-0.105' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.036' Y:'-0.099' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:'-0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.039' Y:'-0.111' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.110' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.044' Y:'-0.109' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.113' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.107' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.043' Y:'-0.111' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.039' Y:'-0.103' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.044' Y:'-0.107' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.109' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.109' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.044' Y:'-0.105' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.041' Y:'-0.109' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.038' Y:'-0.105' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.040' Y:'-0.106' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.042' Y:'-0.115' ╞ ╘╤╧╤ 192.168.1.101
CLIENT - H█╓╟╫╪ °¯_ UDP ╤█╖╕╛...
CLIENT - R╙╒╙╫╨╓ X:' 0↵╝ F▒0F¾ÕH|D °_¯╙ ╘╤╧╤ 192.168.1.101

CLIENT - S╙╪╒╫╪ X:' 0.042' Y:'-0.110' ╞ ╘╤╧╤ 192.168.1.101
```

Figure 3: shows the tera term output from the client application


COM5 - Tera Term VT

```
SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:' 0.001' Y:'-0.013'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.003' Y:'-0.011'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.001' Y:'-0.009'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.003' Y:'-0.010'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.003' Y:'-0.020'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.005' Y:'-0.014'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:' 0.000' Y:'-0.012'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.002' Y:'-0.015'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.007' Y:'-0.016'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.011' Y:'-0.006'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.017' Y:'-0.006'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.008' Y:'-0.009'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:' 0.002' Y:'-0.011'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.010' Y:'-0.015'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.012' Y:'-0.015'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.011' Y:'-0.010'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.009' Y:'-0.012'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.012' Y:'-0.016'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.013' Y:'-0.012'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.012' Y:'-0.013'' [illegible] 192.168.1.102

SERVER - H[illegible] UDP [illegible]...
SERVER - R[illegible] 'X:' ' [illegible] 192.168.1.102
SERVER - S[illegible] 'X:'-0.009' Y:'-0.011'' [illegible] 192.168.1.102
```

Figure 4: shows the tera term output from the server application



IBM Watson IoT Platform

Quickstart

No sign-up required to see how easy it is to connect your device to Watson IoT Platform and view live sensor data

de42d569080e

Last message received at 11:35:48

IoT mbed

status.accelX

Event	Datapoint	Value	Time Received
status	myName	IoT mbed	20 Dec 2018 11:35:49
status	accelX	-0.0938	20 Dec 2018 11:35:49
status	accelY	-0.0469	20 Dec 2018 11:35:49
status	accelZ	1.0314	20 Dec 2018 11:35:49
status	temp	25.625	20 Dec 2018 11:35:49
status	joystick	CENTRE	20 Dec 2018 11:35:49
status	potentiometer1	0.9988	20 Dec 2018 11:35:49
status	potentiometer2	0.5	20 Dec 2018 11:35:49

I've seen my data, what next?

Use your device in an application created with IBM Bluemix.

Click here for more details.

Go to your Bluemix account

Note: When you sign up for a trial you may have to wait up to 24 hours to receive your log-in information

Create an app using the Internet of Things Starter from the Catalog

Note: You will have to name your app and wait for a few minutes for it to start running

When your app is running, select the app URL or type it into the browser to open the Node-RED flow editor

http://<appname>.mybluemix.net

Import the flow for your device into the Node-RED flow editor

IBM Watson IoT Platform

Twitter

Terms of Use

Support

Privacy

Recipes

Documentation

Answers

API

Status

Quickstart

Figure 5: shows the display from the IBM IoT Application which describles the state of the board

Appendices

A. Lab Exercise 2

A.1. Client

The full repo can be accessed from `https://os.mbed.com/users/jamesfernando/code/frdm_labex_5_1_UDPClient/`

```
#include "mbed.h"
#include "EthernetInterface.h"
#include "FXOS8700Q.h"
#include <stdio.h>
#include <iostream>


const int PORT = 7;

static const char* SERVER_IP = "192.168.1.101"; //IP of server board
static const char* CLIENT_IP = "192.168.1.102"; //IP of client board
static const char* MASK = "255.255.255.0";      //mask
static const char* GATEWAY = "192.168.1.1";     //gateway


/*--INITS--------------------------------------------------------------------
    */
Serial pc(USBTX, USBRX);            //create PC interface
```

```
EthernetInterface eth;          //create ethernet
UDPSocket sock;                 //creat Ethernet socket
Endpoint server;                //create endpoint

DigitalOut red(LED_RED);        //debug led
DigitalOut green(LED_GREEN);    //debug led
FXOS8700Q_acc acc( PTE25, PTE24, FXOS8700CQ_SLAVE_ADDR1);


/*--VARIABLES------------------------------------------------------------------------------*/
int n;                  //size of received message
char* recieveBuffer = new char[23];     //create receive buffer
char* sendBuffer = new char[23];  //sample send buffer
std::string inStr;
std::string outStr;


/*--FUNCTION DECLARATIONS----------------------------------------------------------------*/
void init_usb(void);    //initializes pc.printf if required
void init_eth(void);    //initializes Ethernet
void end_eth(void);     //closes Ethernet socket
void printCharArr(char* arr);
int main(void);         //main


/*--FUNCTION DEFINITIONS---------------------------------------------------------------**********************
INIT_USB***/
void init_usb(void)
{
    pc.baud(9600);    //baud

}   //end init_usb()

/*************************************************************************************INIT_ETH***/
void init_eth(void)
{
    eth.init(CLIENT_IP, MASK, GATEWAY);         //set up IP
    eth.connect();                              //connect ethernet
    pc.printf("\nCLIENT - Client IP Address is %s\r\n", eth.getIPAddress());    //get client IP address

    sock.init();                                //initialize socket

    server.set_address(SERVER_IP, PORT);        //set address of server

}   //end init_eth()

```

```
/*
    ****************************************************************************
    END_ETH***/
void end_eth(void)
{
    sock.close();        //close socket
    eth.disconnect();    //close Ethernet

}    //end end_eth()

void printCharArr(char* arr){
    for(int i = 0; i < sizeof(arr); i++){
        pc.printf("%c", arr[i]);
    }
}

/*
    ******************************************************************************
    MAIN***/
int main(void)
{
    red = 0;
                        //client
    green = 1;

    acc.enable();
    float faX, faY;

    init_usb();
                        //initialize the PC interface
    init_eth();
                        //initialize the Ethernet connection

    while(true)
                        //repeat forever
    {
        acc.getX(&faX);
        acc.getY(&faY);
        sprintf(sendBuffer, "X:'% 5.3f' Y:'% 5.3f'\0", faX, faY);
        pc.printf("\nCLIENT - Sending");
        printCharArr(sendBuffer);
        pc.printf(" to server %s\r\n", SERVER_IP);        //print message
    to send
        sock.sendTo(server, sendBuffer, sizeof(sendBuffer)+1);
                                //send message
        pc.printf("CLIENT - Waiting for UDP packet...\r\n");
                        //wait for message
        n = sock.receiveFrom(server, recieveBuffer, 23);
             //receive message from server
        pc.printf("CLIENT - Received ");
        printCharArr(recieveBuffer);
        pc.printf(" from server %s\r\n", SERVER_IP);    //print message
    received


        wait(1);
                        //wait 1 second
    }

}    //end main()
```

A.2. Server

The full repo can be accessed from `https://os.mbed.com/users/jamesfernando/code/frdm_labex_5_1_UDPServer/`

```
/*---INCLUDES---------------------------------------------------------------------------------------------------*/
#include "mbed.h"
#include "EthernetInterface.h"
#include "FXOS8700Q.h"
#include <stdio.h>
#include <iostream>

/*---CONSTANTS--------------------------------------------------------------------------------------------------*/
const int PORT = 7;                                      //arbitrary port

static const char* SERVER_IP = "192.168.1.101";  //IP of server board
static const char* MASK = "255.255.255.0";       //mask
static const char* GATEWAY = "192.168.1.1";      //gateway


/*---INITS------------------------------------------------------------------------------------------------------*/
Serial pc(USBTX, USBRX);          //create PC interface
EthernetInterface eth;            //create ethernet
UDPSocket server;                 //creat server
Endpoint client;                  //create endpoint
FXOS8700Q_acc acc( PTE25, PTE24, FXOS8700CQ_SLAVE_ADDR1);


DigitalOut red(LED_RED);          //debug led
DigitalOut green(LED_GREEN);      //debug led


/*---VARIABLES--------------------------------------------------------------------------------------------------*/
int n;                    //size of received message
char* recieveBuffer;      //Receive Buffer
char* sendBuffer;         //Send Buffer
std::string inStr;
std::string outStr;
float faX, faY;

/*---FUNCTION DECLARATIONS------------------------------------------------------------------------------------*/
void init_usb(void);     //initializes pc.printf if required
void init_eth(void);     //initializes Ethernet
void receive(void);      //receives packets
void printCharArr(char* arr);
```

```c
int main(void);          //main


/*---FUNCTION DEFINITIONS --------------------------------------------------------------------------------*/

/**************************************************************************INIT_USB***/
void init_usb(void)
{
    pc.baud(9600);     //baud

}   //end init_usb()

/**************************************************************************INIT_ETH***/
void init_eth(void)
{
    eth.init(SERVER_IP, MASK, GATEWAY);         //set up IP
    eth.connect();         //connect ethernet
    pc.printf("\nSERVER - Server IP Address is %s\r\n", eth.getIPAddress());     //get server IP address;

    server.bind(PORT);         //bind server

}   //end init_eth()

/***************************************************************************RECEIVE***/
void receive(void)
{


    pc.printf("\nSERVER - Waiting for UDP packet...\r\n");                       //wait for packet
    server.receiveFrom(client, recieveBuffer, 23);              //receive message from client
    pc.printf("SERVER - Received '");
    printCharArr(recieveBuffer);
    pc.printf("' from client %s\r\n", client.get_address());    //print message and client

    acc.getX(&faX);
    acc.getY(&faY);
    sprintf(sendBuffer, "X:'% 5.3f' Y:'% 5.3f'\0", faX, faY);
    pc.printf("SERVER - Sending '");
    printCharArr(sendBuffer);
    pc.printf("' back to client %s\r\n", client.get_address()); //print sending back
    server.sendTo(client, sendBuffer, sizeof(sendBuffer)+1);                                         //send message
}   //end receive()
```

```

void printCharArr(char* arr){
    for(int i = 0; i < sizeof(arr); i++){
        pc.printf("%c", arr[i]);
    }
}

/* *********************************************************************************MAIN***/
int main(void)
{
    red = 1;
    green = 0;      //server

    init_usb();     //initialize the PC interface
    init_eth();     //initialize the Ethernet connection
    acc.enable();

    while (true)    //repeat forever
    {
        receive();  //wait for message
    }

}   //end main()
```