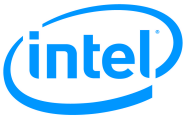
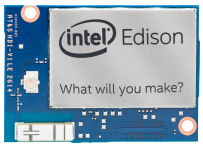




Intel® Edison Tutorial: IoT Coordinated Motion Sensing Reference Design

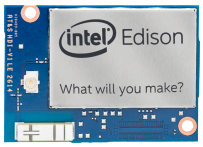
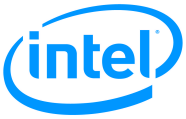
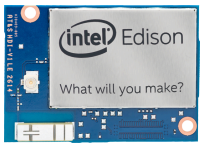


Table of Contents

Revision history		
Version	**Date**	**Comment**
1.0	07/15/2016	Final Draft v1.0
1.1	08/11/2016	Final Draft v1.1



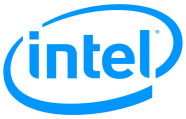


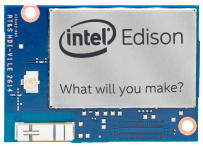
Introduction

This reference design aims to provide instructions on how to use existing code to build a coordinated motion sensing system. To do this, **at least one** Intel Edison kit will be set up as a **client** node. The client Intel Edison(s) will collect data from a 9 Degrees of Freedom Inertial Measurement Unit (**9DOF IMU**). The 9DOF IMU produces three-dimensional (x, y, z) data collected from a micro-accelerometer, micro-gyroscope, and micro-magnetometer. This data will be transferred to a central Intel Edison, equipped with a 9DOF IMU, operating as a **server**. The server will log data read from the 9DOF IMU to a **comma separated value** (CSV) file. The server will spawn a separate thread to service each incoming TCP/IP client connection. Upon receipt of data from a client, the thread servicing the client will append a data entry to a **comma separated value** (CSV) file. This file will be correlated with the IP address assigned to the client Intel Edison. The key concepts used to build this project are listed below.

1. Motion sensing.
2. TCP data transport systems.
3. Multi-threaded multi-node client-server architecture.
4. Data logging.

Before proceeding with this reference design, users must ensure they are familiar with the Intel Edison and the SparkFun 9DOF kit. For more information, refer to the document labelled ***Intel Edison Tutorial – 9DOF Sensor***.





List of Required Materials and Parts

Intel Edison – Server

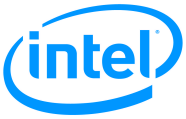
This reference design requires exactly one Intel Edison to act as a server.

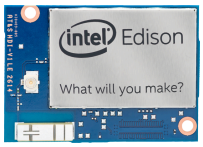
1. 1x Intel Edison Compute Module.
2. 2x USB 2.0 A-Male to Micro-B Cable (micro USB cable).
3. 1x powered USB hub **OR** 1x external power supply.
4. 1x SparkFun 9DOF IMU kit:
    1. 1x Base Block https://www.sparkfun.com/products/13045
    2. 1x 9DOF IMU https://www.sparkfun.com/products/13033
    3. 1x Battery Block https://www.sparkfun.com/products/13037
    4. 1x Hardware Pack https://www.sparkfun.com/products/13187
5. 1x Personal Computer.
6. 1x Roll of Electric Tape
7. Access to a network with an internet connection.

Intel Edison – Client(s)

This reference design will demonstrate how to build a system with one client. To incorporate more than one client node, scale the number of components as indicated.

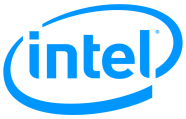
1. 1x Intel Edison Compute Module **per client node**.
2. 2x USB 2.0 A-Male to Micro-B Cable (micro USB cable) **per client node**.
3. 1x powered USB hub **OR** 1x external power supply **per client node**.
4. 1x SparkFun 9DOF IMU kit **per client node**:
    1. 1x Base Block **per client node** https://www.sparkfun.com/products/13045
    2. 1x 9DOF IMU **per client node** https://www.sparkfun.com/products/13033
    3. 1x Battery Block **per client node** https://www.sparkfun.com/products/13037
    4. 1x Hardware Pack **per client node** https://www.sparkfun.com/products/13187
5. 1x Personal Computer.
6. Access to a network with an internet connection.

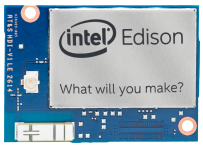




Hardware Setup

1. Assemble all SparkFun 9DOF IMU kits with an Intel Edison Compute Module.
2. Designate **exactly one** Sparkfun 9DOF IMU kit with Intel Edison Compute Module as a server. This will be referred to as **the server Intel Edison**.
3. Designate **all remaining** Sparkfun 9DOF IMU kits with Intel Edison Compute Modules as clients. These will be referred to as **the client Intel Edisons**.





Software Setup

1. Open the below link on a web browser on your personal computer.

   https://drive.google.com/drive/folders/0B4NGslzPqDhvdHFEYXVMMjdJWTQ

2. Download and extract the contents of the **zip archive** labelled **FILES** to your personal computer.

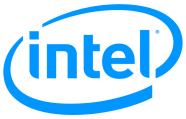
3. Examine the contents of the **zip archive**. It should contain the below files.

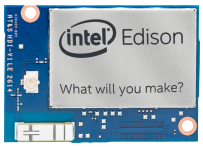
   Files within the subfolder labelled **client**:

   9DOF.c
   9DOF.h
   client.c
   client.h
   LSM9DS0.c
   LSM9DS0.h
   main.c
   Makefile

   Files within the subfolder labelled **server**:

   9DOF.c
   9DOF.h
   server.c
   server.h
   LSM9DS0.c
   LSM9DS0.h
   main.c
   Makefile





Intel Edison – Server

Follow the below steps to set up the software for the **server** Intel Edison.

1. Access the shell program on your Intel Edison through an SSH connection. For more details, refer to the document labelled ***Intel Edison Tutorial – Introduction, Linux Operating System Shell Access, and SFTP***.

2. Create a new directory labelled **IoT-CM**.

   $ cd
   $ mkdir IoT-CM

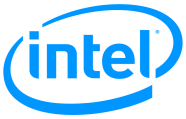
3. Transfer all files present in the subfolder **server** within the zip archive **FILES** to the Intel Edison via SFTP to the folder labelled **IoT-CM**.

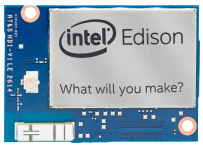
4. Compile the source files into an executable binary file.

   $ make

```
root@server_edison:~/9DOF/server# make
gcc -c main.c -o main.o
gcc -c 9DOF.c -o 9DOF.o
gcc -c server.c -o server.o
gcc -c LSM9DS0.c -o LSM9DS0.o
gcc -lmraa -lm -pthread main.o 9DOF.o server.o LSM9DS0.o -o server
root@server_edison:~/9DOF/server#
```

Figure 1: Successful compilation of server files





Intel Edison – Client(s)

Repeat the steps below for each Intel Edison client node.

1. Access the shell program on your Intel Edison through an SSH connection. For more details, refer to the document labelled ***Intel Edison Tutorial – Introduction, Linux Operating System Shell Access, and SFTP***.

2. Create a new directory labelled **IoT-CM**.

   $ cd
   $ mkdir IoT-CM

3. Transfer all files present in the subfolder **client** within the zip archive **FILES** to the Intel Edison via SFTP to the folder labelled **IoT-CM**.

4. Compile the source files into an executable binary file.

   $ make

```
root@client_edison_1:~/9DOF/client# make
gcc -c main.c -o main.o
gcc -c client.c -o client.o
gcc -c 9DOF.c -o 9DOF.o
gcc -c LSM9DS0.c -o LSM9DS0.o
gcc -lmraa -pthread -lm main.o client.o 9DOF.o LSM9DS0.o -o client
root@client_edison_1:~/9DOF/client#
```

Figure 2: Successful compilation of client files

Code Execution

Intel Edison – Server

1. Access the shell program on the server Intel Edison. Record the server's IP address.

   $ configure_edison --showWiFiIP

```
ison:~/9DOF/server# configure_edison --showWiFiIP
10.0.1.7
root@server_edison:~/9DOF/server#
```

Figure 3: IP address of the server Intel Edison

2. Execute the **main** executable binary file on the server Intel Edison.

   $ ./server

```
root@server_edison:~/9DOF/server# ./server
server init: 0.0.0.0 5000
Calculating the offsets...
ERROR on accept: Resource temporarily unavailable
Latest child process is waiting for an incoming client connection.
```

Figure 4: Successful execution of the binary file main. Record the port number, it is required for a later step

If the main function exits and shows a message similar to below, follow the below steps.

```
root@server_edison:~/9DOF/server# ./server
ERROR on binding: Address already in use
```

Figure 5: Error on port binding, follow the below steps to resolve the issue

$ vi main.c

```
 1 #include <stdio.h>
 2 #include <string.h>
 3
 4 #include <sys/types.h>
 5 #include <signal.h>
 6 #include <pthread.h>
 7
 8 #include "server.h"
 9 #include "9DOF.h"
10 #include "LSM9DS0.h"
11
12 #define PORTNO 5000
13
14 static volatile int run_flag = 1;
15
16 void do_when_interrupted()
17 {
18         run_flag = 0;
19         printf("\n Threads exiting. Please wait for cleanup operations to complete...\n");
20 }
```

Figure 6: Editing the port number

Change line 12 such that it has a port in the range of 2000-8000. Let's try port 8000

```
 1 #include <stdio.h>
 2 #include <string.h>
 3
 4 #include <sys/types.h>
 5 #include <signal.h>
 6 #include <pthread.h>
 7
 8 #include "server.h"
 9 #include "9DOF.h"
10 #include "LSM9DS0.h"
11
12 #define PORTN   8000
13
14 static volatile int run_flag = 1;
15
16 void do_when_interrupted()
17 {
18         run_flag = 0;
19         printf("\n Threads exiting. Please wait for cleanup operations to complete...\n");
20 }
```

Figure 7: main.c after the port number has been changed

Save and exit the main.c file, then recompile the project.

$ make

```
root@server_edison:~/9DOF/server# make
gcc -c main.c -o main.o
gcc -lmraa -lm -pthread main.o 9DOF.o server.o LSM9DS0.o -o server
```

Figure 8: Successful recompilation of project

Execute the compiled binary file.

```
root@server_edison:~/9DOF/server# ./server
server init: 0.0.0.0 8000
Calculating the offs
ERROR on accept: Resource temporarily unavailable
Latest child process is waiting for an incoming client connection.
```

Figure 9: Successful execution of the binary file main. Record the port number, it is required for a later step

If the above steps fail to resolve the issue, try to edit the port number again.

If the issue persists, reboot your Intel Edison. If the issue is not resolved, please contact your course instructor.

Intel Edison – Client(s)

1. Run the **main** executable binary file by issuing the below command.

   $ ./client <IP_ADDR_SERVER> <PORT>

```
root@client_edison_1:~/9DOF/client# ./client 10.0.1.7 5000
Calculating the offsets...
msg from server: 10.0.1.4 sent the server: time (epoch), accel_x, accel_y,
```

Figure 10: <IP_ADDR_SERVER> will vary based on step 1. <PORT> will vary based on step 2

Code Execution – Terminating the Processes

1. To terminate a process, issue an interrupt by pressing **ctrl-C**.

```
^Cmsg from server: 10.0.1.4 sent the server: 1471026945.8410170078,-0.203979,
2878,-0.039551,0.000000
root@client_edison_1:~/9DOF/client#
```

Figure 11: Terminating the process "client" on the client

```
^Cmsg from server: 10.0.1.4 sent the server: 1471026945.8410170078,-0.203979,
2878,-0.039551,0.000000
root@client_edison_1:~/9DOF/client#
```

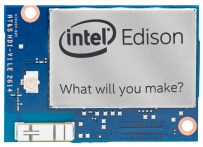
Figure 12: Terminating the process "server" on the server

2. Once the process **server** has been terminated on the server Intel Edison, examine the first few lines of one of the CSV files generated.

   $ ls *.csv
   This will print a list of CSV files containing data aggregated by the server Intel Edison.
   $ head <file_name>
   This command will print the first 10 lines of the file specified by <file_name>

```
root@server_edison:~/9DOF/server# ls -l *.csv
-rw-r--r-- 1 root root 33821 Aug 12 18:20 output_file_IP_10.0.1.4_TIMESTAMP_1471026030.csv
-rw-r--r-- 1 root root 60574 Aug 12 18:20 server_test_data.csv
root@server_edison:~/9DOF/server# head server_test_data.csv
time (epoch), accel_x, accel_y, accel_z, gyro_x, gyro_y, gyro_z, mag_x, mag_y, mag_z, temperature
1471026029.1333210468,-0.210327,-0.056030,2.007202,-1.732029,-1.426960,-1.601296,0.078613,0.021545,0.032043,0.000000
1471026029.1582860947,-0.200806,-0.059937,2.005737,-1.649785,-1.718556,-2.012520,0.060913,0.014526,0.046265,0.000000
1471026029.1761469841,-0.198120,-0.068115,2.013794,-1.948857,-2.144734,-1.571389,0.064026,0.016296,0.045166,0.000000
1471026029.1936249733,-0.195068,-0.057129,1.957275,-1.956334,-1.703602,-1.608773,0.067993,0.017090,0.042969,0.000000
1471026029.2108180523,-0.165527,-0.048340,2.049561,-1.410527,-2.152210,-2.543374,0.080994,0.016907,0.031860,0.000000
1471026029.2283260822,-0.222900,-0.062866,2.002075,-2.113347,-2.369038,-1.698495,0.076416,0.014282,0.036987,0.000000
1471026029.2448389530,-0.187866,-0.093018,2.051514,-1.926426,-1.284901,-2.042428,0.071960,0.014221,0.039917,0.000000
1471026029.2616579533,-0.185303,-0.064819,1.980469,-1.889042,-2.122303,-1.922799,0.071716,0.013428,0.038086,0.000000
1471026029.2791419029,-0.195923,-0.062012,1.994385,-1.784367,-2.189594,-1.698495,0.072388,0.013306,0.049927,0.000000
root@server_edison:~/9DOF/server# head output_file_IP_10.0.1.4_TIMESTAMP_1471026030.csv
time (epoch), accel_x, accel_y, accel_z, gyro_x, gyro_y, gyro_z, mag_x, mag_y, mag_z, temperature
1471026030.9939210415,-0.157593,-0.051392,1.900757,1.107974,-2.476895,-0.129919,0.095764,-0.008789,-0.047058,0.000000
1471026031.0306940079,-0.159912,-0.065796,1.933105,0.562167,0.050265,-0.840215,0.105347,-0.005676,-0.041382,0.000000
1471026031.0575029850,-0.195068,-0.050903,1.871704,0.307956,-1.766599,-0.017767,0.100647,-0.002869,-0.036194,0.000000
1471026031.0801920891,-0.191650,-0.049438,1.882690,0.764041,-2.252591,-0.279455,0.091125,-0.008667,-0.047485,0.000000
1471026031.1035521030,-0.193970,-0.044312,1.955322,0.913577,-2.125485,-0.167303,0.101135,-0.009033,-0.041260,0.000000
1471026031.1280848980,-0.175293,-0.062866,1.892456,0.801425,-2.073148,0.057001,0.115051,-0.002869,-0.038269,0.000000
1471026031.1539030075,-0.169312,-0.045288,1.936279,0.233187,-2.155393,0.236445,0.108704,-0.006042,-0.036560,0.000000
1471026031.1781570911,-0.206543,-0.050537,1.905151,0.906100,-1.848844,0.199061,0.109253,-0.002075,-0.036377,0.000000
1471026031.2030138969,-0.165405,-0.054321,1.933838,0.936007,-2.357266,-0.257024,0.103943,-0.007751,-0.036926,0.000000
root@server_edison:~/9DOF/server#
```

Figure 13: First 10 lines of CSV file



Tasks

Perform the following modifications to gain a deeper understanding of the system.

1. Attach both the server and the client Intel Edison's to a rigid body such as a box. Ensure they have the same orientation.

   Start the data collection process on the server Intel Edison by issuing the command below

   $ ./server

   Start the data collection process on each client Intel Edison by issuing the command below

   $ ./client <IP_ADDR> <PORT>

   Carefully apply an impulse force to the rigid body such that the Intel Edison's will record a sharp acceleration in one axis.

   Extract the generated CSV log files from the Intel Edison's.

   Calculate the time at which each Intel Edison experienced the peak of this acceleration.

   Submit these results as waveforms displayed on the same set of axes, and a table showing the time at which each Intel Edison observed the impulse.

2. Edit, test and run the code in main.c of the **client** such that it performs the following tasks
   a. Calculates **pitch angle** and **roll angle** using the equations below.
   b. No longer sends **raw sensor data** to the **server**.
   c. Sends **calculated pitch and roll** data to the **server Intel Edison**.

   Equations to calculate angles are
   a. Pitch

$$= -\operatorname{atan2}\left(a_x, \sqrt{a_x^2 + a_y^2 + a_z^2}\right) \times \frac{180}{\pi}$$

   b. Roll

$$= \operatorname{atan2}\left(a_y, \sqrt{a_x^2 + a_y^2 + a_z^2}\right) \times \frac{180}{\pi}$$

   c. Where the three dimensional acceleration vector is defined as

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}$$

   Submit a screenshot of the server receiving the **calculated pitch** and **roll** data from a client Intel Edison.