TinyOS 2.1.2.1 Installation on Raspbian Jessie

Open a terminal

0. Install prerequisites

```
sudo apt-get update
sudo apt-get install emacs gperf bison flex git automake autoconf
libtool
```

Reboot Raspberry PI

1.Install NesC and TinyOS from source repositories in your home directory:

Go to your home directory for example: `/home/pi`

1.1 Install NesC:

```
git clone https://github.com/tinyos/nesc.git
cd nesc/
./Bootstrap
./configure
make
sudo make install
```

go back to your home folder: `cd ..`

1.2 Install TinyOS:

```
git clone https://github.com/tinyos/tinyos-main.git
cd tinyos-main/tools
./Bootstrap
./configure
make
sudo make install
```

2. Install TinyOS Essentials e.g. msp430-gcc compiler

```
sudo apt-get install build-essential avarice avr-libc msp430-libc
avrdude binutils-avr binutils-msp430 gcc-avr gcc-msp430 gdb-avr
subversion graphviz python-docutils checkinstall
```

3. Set up TinyOS environment and variables

3.1 Open .bashrc file on leafpad (a text editor program that comes with Raspbian)

```
sudo leafpad ~/.bashrc
```

3.2 Add the following lines to the end of the file and save
** make sure that TOSROOT is set to *tinyos-main* folder

```
export TOSROOT="/home/pi/tinyos-main"
export TOSDIR="$TOSROOT/tos"
export
CLASSPATH=$CLASSPATH:$TOSROOT/tools/tinyos/java/tinyos.jar:.
export MAKERULES="$TOSROOT/support/make/Makerules"
export PYTHONPATH=$PYTHONPATH:$TOSROOT/tools/tinyos/python
echo "setting up TinyOS on source path $TOSROOT"
```

4. Change permission of TOSROOT folder

got to your home folder where tinyos-main is located e.g. `home/pi/`

```
sudo chmod -R a=wrx tinyos-main/
```

5. Java serial communication: copy libtoscomm.so and libgetenv.so to /usr/lib

```
cp /home/pi/tinyos-main/tools/tinyos/jni/serial/libtoscomm.so /usr/lib
cp /home/pi/tinyos-main/tools/tinyos/jni/env/libgetenv.so /usr/lib
```

Reboot Raspberry PI

TinyOS Testing:

1. Plug a mote to a USB port, check device name by using *motelist* e.g. /dev/ttyUSB0, change permission of the serial port

```
motelist
sudo chmod 777 /dev/ttyUSB0
```

2.Compile and run Blink application

go to your home directory

```
cd  /tinyos-main/apps/Blink
make telosb
make reinstall.1 telosb
```

** Mote's LEDs should be blinking if the program is successfully installed

3. Test java serial forwarder

```
java net.tinyos.tools.Listen -comm serial@dev/ttyUSB0:telosb
```

** dev/ttyUSB0 is a device name obtained from a command *motelist*
** the terminal should show --> serial@/dev/ttyUSB0:115200: resynchronizing
press Ctrl-c to quit

Back-end Installation on AWS:

Components: AWS IoT, AWS DynamoDB
AWS DynamoDB guide:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SettingUp.DynamoWebService.html
Create a DynamoDB database, use default setting

AWS IoT guide: http://www.cse.wustl.edu/~lu/cse521s/Slides/aws-iot.pdf

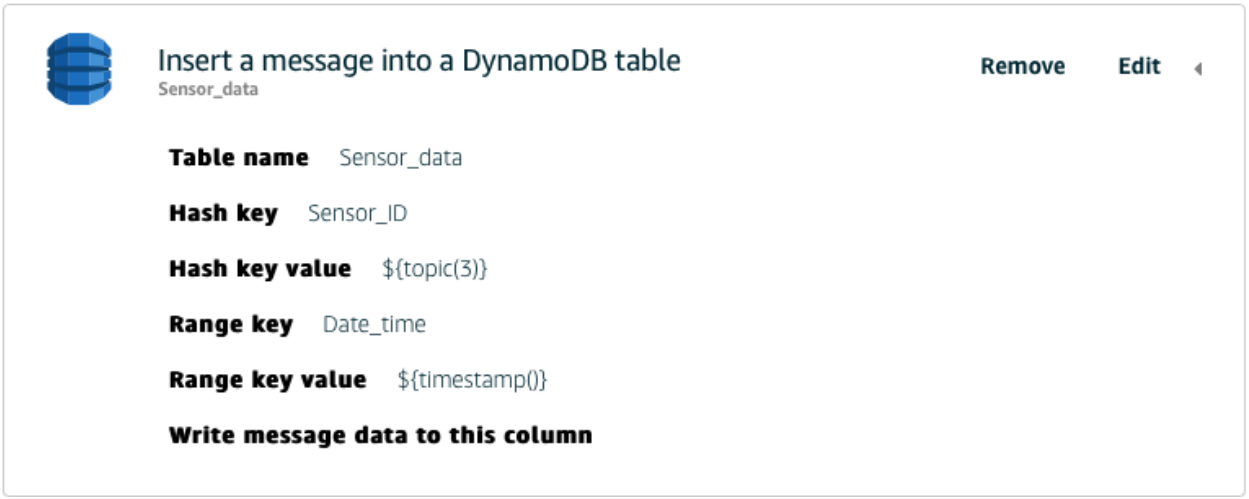
1 Create a virtual thing for your Raspberry PI.

2 Connect with your Raspberry PI.
Create and get certificate and private key, download to your Raspberry PI,
(**Note: Remember where you put, you will need that when you run Main.java**)

3 Attach a policy
Set your policy to allow sub/pub

4 Attach a rule (Important, connect your thing to DynamoDB)
a. Name: Rule name
b. Topic: The shadow address from your thing,the one PI publish to
e.g.$aws/things/sensor_01/shadow/update

5 Add action: Choose **Insert a message into a DynamoDB table**, configured as follows



Hints:

1. In order to access programmatically DynamoDB, you may need to get an access key for your account. Check here

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SettingUp.DynamoWebService.html

Then you can write a script to read or update the Database, for example, create a client:

```
client = boto3.client('dynamodb',region_name='us-west-2',aws_access_key_id='**',aws_secret_access_key='**')
```
(Python)

2.