

SHIMMER Getting Started Guide

[Konrad Lorincz](#)
November 2, 2006
Harvard University

Disclaimers

1. Please note that this page is very dated (written in 2006) and much of the information described here may no longer be accurate!
2. The descriptions here were based upon the mainstream tinyos codebase (specifically TinyOS 1.1.15). Note that this was different from the one that the SHIMMER designers branched to provide their platform support.
3. The official codebase support as of this writing is at sourceforge
 - for tinyos-1.x - under tinyos-1.x/contrib/handhelds.
 - for tinyos-2.x - in the main tinyos-2.x branch
4. SHIMMER is now a commercial product. Please see <http://shimmer-research.com>
5. The original SHIMMER was replaced by SHIMMER2 last year, so any noted hardware characteristics may be different in version 2.

Table of Contents

1. [Introduction](#)
2. [Installing the SHIMMER Platform](#)
 - a. Download and Install the SHIMMER Platform Code
3. [Running Blink](#)
 - a. Build and Run Blink
4. [Running TestSamplingStorageRadio](#)
 - a. Build and Run TOSBase
 - b. Build and Run the Java Receive
 - c. Build and Run TestSamplingStorageRadio
5. [Migrating from TelosB/TMote Sky to SHIMMER](#)
 - a. Major similarities
 - b. Major differences
 - c. Major pin differences
6. [Support](#)
 - a. References
 - b. Mailing lists



Introduction

This document describes how to configure TinyOS 1.1.15, in order to compile and run standard TinyOS applications on the SHIMMER platform. This guide assumes that you are using the standard TinyOS 1.1.15 components (such as the radio stack from "\$TOSROOT/tos/lib/CC2420Radio").

SHIMMER was designed by **Benjamin Kuris** at Intel Digital Health Advanced Technology Group, Cambridge MA.

Installing the SHIMMER Platform

This section was outdated and removed!

Running Blink

At this point you should be able to compile and install standard tinyos applications such as Blink for the shimmer platform. If you can't, you may need to manually review the changes made by "makeShimmerPlatform.pl".

Build Blink

- 1 Compile and install Blink. For example:

```
$ cd $TOSROOT/apps/Blink
$ make shimmer
$ make shimmer reinstall bsl,X
```

At this point you should see Blink running on your SHIMMER.

Running TestSamplingStorageRadio

As the name suggests "TestSamplingStorageRadio", test the sampling, SD storage, and radio on SHIMMER. The application samples the three accelerometers on SHIMMER, saves the values in a TOS_Msg, and then writes the TOS_Msg to the SD. It then reads the last TOS_Msg from SD and broadcasts it over the radio. A separate mote running TOSBase receives the messages, and forwards them to a java application which displays the ADC values. Note: Currently the second mote cannot be SHIMMER because it can't send forward messages over the UART. I use a TelosB/Tmote Sky mote.

- 1 Download and decompress the "TestSamplingStorageRadio" application [\[download\]](#)

Build and Run TOSBase

- 1 Compile and install Blink on a TelosB mote. For example:

```
$ cd $TOSROOT/apps/TOSBase
$ make telosb install.0 bsl,X
```

Build and Run the Java Receive

- 1 Start SerialForwarder and point it to your TOSBase port. For example:

```
$ java net.tinyos.sf.SerialForwarder -comm serial@COMX:telos
```

- 1 Compile and run the java Receiver. For example:

```
$ cd testSamplingStorageRadio/java
$ make
$ make run
```

Build and Run TestSamplingStorageRadio

- 1 If you haven't already, connect SHIMMER to your PC.
- 1 In a separate console window, type the following:

```
$ cd testSamplingStorageRadio
$ make shimmer install bsl,X
```

After it compiles and installs, you should start seeing the accelerometer ADC values in your java Receiver console window. The format is as follows:

```
...
# srcAddr= 10 timeRaw= 635857 timeSec= 19.405 nbrSamples= 9
19.405 2318 2038 2037
19.605 2320 2044 2036
19.805 2323 2049 2035
# srcAddr= 10 timeRaw= 655519 timeSec= 20.005 nbrSamples= 9
20.005 2317 2046 2037
20.205 2321 2047 2041
20.405 2322 2042 2035
...
```

The line beginning with "#" indicates a new packet (this format is very convenient for plotting with gnuplot). The samples are displayed in subsequent lines and have the format

```
<timeSec> <accX> <accY> <accZ>
```

You can modify the sampling rate by editing the "testSamplingStorageRadio/TestSamplingStorageRadio.h". If you edit this file, you **MUST** also re-make the java Receiver, because it extracts the sampling rate from this file!

Migrating from TelosB/TMote Sky to SHIMMER

This section describes the major similarities and differences between TelosB/TMote Sky and SHIMMER.

Major Similarities

- 1 Same processor, MSP430 (version F1611)
- 1 Same radio chip, CC2420

Major Differences

- 1 Smaller form factor
- 1 Optional Bluetooth Radio

- 1 MicroSD flash storage (currently up to 2GB available)
- 1 Built in 3-axis accelerometer (Freescale MMA7260Q, with selectable gain 1.5/2/4/6G)
- 1 Extra led (orange)
- 1 Support for expansion board via Hirose DF12 20 pin connector
- 1 Li-Ion battery management
- 1 Programming over USB via programming board

Major Pin Differences

- 1 USART1 used by CC2420 and Bluetooth Radios (On TelosB USART1 is used for USB UART/RS232 Tx/Rx)
- 1 USART0 used by MicroSD and external expansion (On TelosB USART1 is shared by CC2420, ST Flash, IDC expansion connector)
- 1 RADIO_SFD (timed version) is wired to port P1.2, which is also the TimerA capture/compare 1 (CC1A) output. NOTE: This means that you can't use the MSP430ADC12 component for sampling because it requires explicit use of the main TimerA register and the CC1A output.
- 1 Requires UART to PC bridge (using UART0 on external expansion connector) for serial connection to PC
 - 1 The currently programming board does not support transmission over the UART to a PC. For example, you can't send a message to TOS_LOCAL_ADDRESS via GenericComm. This is disabled from GenericComm via the wrapper in "\$TOSROOT/tos/platform/shimmer/UARTFramedPacket.nc". A programming board with dual port FTDI USB chip is in development by Intel Research Seattle and will be available soon.

Support

References

- 1 [SHIMMER Hardware Guide](#)
- 1 [MMA7260Q Datasheet](#) (accelerometers)
- 1 [IDG300 Datasheet](#) (gyroscopes - on daughterboard)
- 1 [Antenova Rufa Datasheet](#) (antenna)
- 1 MSP430: [datasheet](#) | [user's guide](#)
- 1 [TMote Sky Datasheet](#)

Mailing Lists

- 1 [Shimmer-Users](#) (requires subscription)
- 1 [TinyOS](#)