ACme Quick Start Guide
For Developers

http://acme.cs.berkeley.edu/

Xiaofan Jiang :: Fred
fxjiang@eecs.berkeley.edu
Computer Science Department
University of California, Berkeley
LICENSE

“Copyright (c) 2008 The Regents of the University of California. All rights reserved.”

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without written agreement is hereby granted, provided that the above copyright notice, the following two paragraphs and the author appear in all copies of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS."
Step 1: Installing TinyOS

- Follow instructions at http://docs.tinyos.net/index.php/Getting_started to install the TinyOS development environment.
  - Xubuntu's VMware virtual machine image is often the easiest way to obtain a working TinyOS development environment.
  - This guide will assume you have successfully installed TinyOS 2.1 or later.

Step 2: Setting up ACme source tree

- Install tinyos-contrib source tree (example code installs it in the same level as $TOS_ROOT):

  cd $TOSROOT/..
  cvs -d:pserver:anonymous@tinyos.cvs.sourceforge.net:/cvsroot/tinyos login
  cvs -z3 -d:pserver:anonymous@tinyos.cvs.sourceforge.net:/cvsroot/tinyos contrib

- Compiling an example application

  cd $TOSROOT/../tinyos-2.x-contrib/berkeley/acme/apps/ACMeterApp
  make epic

- This application should compile without error (warning is ok).

Step 3: Programming an ACme

- Insert Epic Programmer into the programming header as shown below. The side marked with Vcc should align with the pin pointed by the arrow.
• Make sure that Makefile has the right version of ACme uncommented and the same radio channel as the BaseStation you are using.

```make
COMPONENT = ACMeterAppC
include $(MAKERULES)
# Specify radio channel here
PFLAGS += -DCC2420_DEF_CHANNEL=18
# Uncomment this line for ACme Rev.A
# CFLAGS += -I../../tos/sensorboards/ACme/
# Uncomment this line for ACmeX2 (Rev.B)
CFLAGS += -I../../tos/sensorboards/ACmeX2/
```

• Connect the other end to a USB port and program ACme (assuming you only have one ACme connected via USB):

```bash
cd $TOSROOT/../tinyos
cd $TOSROOT/apps/
```

```make
make epic install miniprogs
```

• At this point, the ACme should be broadcasting power readings at 1Hz.

• IMPORTANT: NEVER PLUG IN ACME INTO AC SOCKET WHILE ATTACHED TO THE PROGRAMMER!

• Unplug programmer from both the laptop and the ACme and securely screw in the cover. Do not leave any internal parts exposed. Plug into an AC socket and plug in the appliance under monitoring.

**Step 4: Monitoring Energy Wirelessly**

• Plug in a CC2420 compatible mote (telosb in this example) and program with BaseStation (make sure Makefile specify the same channel used to program ACme).

```bash
cd $TOSROOT/apps/BaseStation
```

```make
make telosb install
```

• The mote’s green LED should be blinking at 1Hz indicating reception of packets from ACme. Try pressing the “reset” button on the mote if not working right away.

• Start decoding packets:

```bash
cd $TOSROOT/../tinyos-2.x-contrib/berkeley/acme/tools/
java net.tinyos.tools.Listen -comm serial@/dev/ttyUSB0:telos | ./powerParse.pl
```

• Note: `ttyUSB0` may be different depending on the OS and which USB port the mote is assigned too. Use “motelist” command to find the right port.

• At this point, energy readings should be streaming to stdout and can be piped / redirected as desired.
IPv6/6LoWPAN Connectivity

ACme can be easily programmed with the Berkeley blip IPv6/6LoWPAN stack to enable Internet connectivity via UDP/IP. In this part of the guide, we will install the /app/ACme application which sends energy statistics to a user-specified IP address and allows remote administration.

**Step 1: Installing and configuring blip Edge Router**

Please refer to the *blip* document for installation and basic usage of the *blip* stack: http://docs.tinyos.net/index.php/BLIP_Tutorial

This guide will assume you have installed the Edge Router on a mote successfully and have activated *ip-driver*.

Make sure to specify the same channel in the *serial_tun.conf* file as the ACme nodes.

It’s a good idea to familiarize with the built-in commands of the *ip-driver*, such as *help, conf, links, routes*.

**Step 2: Configuring and installing ACme/IP application**

- The ACme/IP application opens two UDP ports, one for reporting energy/power readings to a user-specified IP address, and the other for interacting with the onboard shell.
- If you have established global IP connectivity in the previous step, you can specify any IPv6 address. The computer at that IP address can monitor a UDP port and process the packets.

```
COMPONENT=ACMeterAppC
BOOTLOADER=tosboot

# Report locally
CFLAGS += -DREPORT_ADDR="fec0::64"

# Report to other IPv6 address
# CFLAGS += -DREPORT_ADDR="2001:283:6f04:5c8::64"

# Use same channel as serial_tun.conf
CFLAGS += -DCC2420_DEF_CHANNEL=21

......
```

- Configuring the reporting frequency and UDP port number in the ACMeterAppP file:
......

// PERIOD specifies the reporting frequency
#define PERIOD 60

// this line specifies the reporting UDP port
report_dest.sin6_port = hton16(7001);
......

- Install ACme/IP application. Make sure Makefile specify the same channel as in serial_tun.conf
  ```
  cd $TOSROOT/../tinyos-2.x-contrib/berkeley/acme/apps/ACme
  make epic blip install.xx miniprog ### replace xx with node ID
  ```
- Replace xx with the node ID you wish to use (such as 1).
- Screw in cover and plug into AC outlet.
- At this point, if you type `links` in the ip-driver terminal, you should see the ACme node showing up. Issue `newroutes` command to update link distance.

**Step 3: Interacting with ACme/IP application**
- You can interact with ACme directly using standard networking tools such as netcat and ping. The address for the ACme node is fec0::xx (or optionally the global IP prefix). In the following example, the node ID is 1.
  ```
  xubuntos@xubuntos-tinyos:$ ping6 fec0::1
  PING fec0::1(fec0::1) 56 data bytes
  64 bytes from fec0::1: icmp_seq=1 ttl=65 time=78.2 ms
  64 bytes from fec0::1: icmp_seq=2 ttl=65 time=72.6 ms
  
  xubuntos@xubuntos-tinyos:$ nc6 -u fec0::1 2000
  nc6: using datagram socket
  help
  sdsh-0.9 builtins: [help, echo, ping6, uptime, ident] [set, read, period, reset]
  read power
  Instantious power: 775 watts
  read energy
  Energy: 1381 jouls
  set off
  uptime
  up 99 seconds
  ```

Congratulations! You have reached the end of the quick start guide. Please email me at fxjiang@eecs.berkeley.edu if you have any questions or suggestions.