

INSTYTUT INFORMATYKI, AUTOMATYKI I ROBOTYKI  
POLITECHNIKA WROCŁAWSKA  
RAPORT SERII PREPRINTY

PRE/94/2011

**Instalacja i uruchomienie systemu TinyOS**

**Instalacja i uruchomienie systemu TinyOS**

Tomasz Surmacz

Słowa kluczowe:

sieci czujników  
czujniki bezprzewodowe  
transmisja radiowa  
zużycie energii

Wrocław, lipiec 2012

## **Abstrakt**

Instalacja systemu TinyOS

# Instalacja i uruchomienie systemu TinyOS

Tomasz Surmacz

Institute of Computer Engineering, Control and Robotics,  
Wrocław University of Technology  
ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław

24 lipca 2012

## 1 Instalacja w systemie Linux

Instalacja w systemie Linux sprowadza się do zainstalowania gotowych pakietów o nazwie `tinyos`. W systemach Ubuntu należy wykonać:

```
apt-get install tinyos
```

Wcześniej może być konieczne wykonanie także `apt-get update`. Jeśli okaże się, że pakiet `tinyos` jest nieznan, należy do pliku `/etc/apt/sources.list` dopisać:

```
deb http://tinyos.stanford.edu/tinyos/dists/ubuntu maverick main
```

(przy czym zamiast “maverick” oznaczającego Ubuntu 10.10 powinna się tam znaleźć nazwa kodowa właściwej wersji Ubuntu), po czym wykonać `apt-get update` i `apt-get install tinyos`. Przykład instalacji pokazany jest w dodatku 5. Standardowo pakiet instaluje się w katalogu `/opt`, np. `/opt/tinyos-2.1.1/`. W katalogu tym po instalacji znajdzie się m.in. skrypt o nazwie `tinyos.sh`, który należy wczytać w bieżącym shell-u, aby poustawiać właściwe zmienne środowiskowe konieczne do kompilacji.

Skrypt ten można wywoływać przez `source /opt/tinyos-2.1.1/tinyos.sh` ze swojego `.bashrc` albo ręcznie w miarę potrzeby.

## 1.1 Instalacja starszej wersji bibliotek dla węzłów TelOS

Bieżące archiwa nadają się wprost do obsługi węzłów Iris, jednak załączone z nimi biblioteki i nagłówki dla węzłów TelOS nie zostały dobrze przetestowane i nie pozwalają na poprawną kompilację programów. Aby wszystko działało, należy zainstalować starszą wersję bibliotek dla węzłów “msp430”.

Szczegółowe informacje na ten temat można znaleźć pod adresem [http://docs.tinyos.net/tinywiki/index.php/Installing\\_TinyOS\\_2.1.1#Manual\\_installation\\_on\\_your\\_host\\_OS\\_with\\_RPMs](http://docs.tinyos.net/tinywiki/index.php/Installing_TinyOS_2.1.1#Manual_installation_on_your_host_OS_with_RPMs)

Pakiety, które należy zastąpić starszymi to:

```
1 root@plonk:# dpkg -l | grep msp430
2 ii  msp430-binutils-tinyos  2.21.1-20110821  TinyOS-specific
   MSP430 binutils
3 ii  msp430-gcc-tinyos      4.5.3-20110821  TinyOS-specific
   MSP430 gcc
4 ii  msp430-libc-tinyos     20110612-20110821  TinyOS-specific
   MSP430 libc
5 ii  msp430-tinyos          20110821         Dummy package to
   pull in all the MSP430 packages
6 ii  msp430-tinyos-base     2.1-20080806     Dummy package
   required by all msp430-tinyos tools. Removing this package
   will remove all msp430-tinyos related packages.
7 ii  msp430mcu-tinyos      20110613-20110821  TinyOS-specific
   MSP430 headers and linker scripts
8 ii  tinyos-required-msp430  2.1-20090326     Dummy package to
   pull in all optional packages for tinyos
```

Starsze wersje tych pakietów można pobrać z repozytorium <http://tinyos.stanford.edu/tinyos/dists/ubuntu.old/pool/main/m/>

W razie kłopotów z niespełnionymi zależnościami (od nowszych wersji pakietów) należy użyć opcji `--force`

```
dpkg --force all -i tinyos-required-msp430_2.1-20090326_all.deb
```

Spowoduje to zainstalowanie pakietów:

```
1  msp430-binutils-tinyos
2  msp430-gcc-tinyos
3  msp430-libc-tinyos
4  msp430-optional-tinyos
5  msp430-tinyos
6  msp430-tinyos-base
```

Ostatnią rzeczą, jaką trzeba sprawdzić, to prawa dostępu do odpowiednich urządzeń. Urządzenia podłączone przez USB będą pojawiać się jako emulowane

porty szeregowo o nazwach typu `/dev/ttyUSBn` z kolejnymi numerami, a standardowe prawa dostępu zakładane dla tych urządzeń w systemach Linux to:

```
1 crw-rw---- 1 root dialout 188, 0 2012-05-22 20:09 /dev/ttyUSB0
2 crw-rw---- 1 root dialout 188, 1 2012-05-22 16:17 /dev/ttyUSB1
3 crw-rw---- 1 root dialout 188, 2 2012-05-22 16:17 /dev/ttyUSB2
4 ...
```

Aby bez problemów dostawać się do tych urządzeń należy dopisać nazwę swojego konta do grupy `dialout` w pliku `/etc/group`, np.:

```
1 ts@plonk:~% who am i
2 ts pts/6 2012-05-22 18:58 (laptok)
3 ts@plonk:~% grep dialout /etc/group
4 dialout:x:20:ts,mn
5 ts@plonk:~% id -a
6 uid=138(ts) gid=138(ts)
7 grupy=20(dialout),24(cdrom),29(audio),46(plugdev),115(pulse),116(
8 pulse-access),
9 ...
```

Zmiany w liście grup odniosą skutek przy następnym zalogowaniu się użytkownika.

## 1.2 Dodatkowe pliki dla węzłów TelOSB

Czujniki płytek pomiarowych DS1000 ( $CO_2$ ,  $CO$ , termistor), AR1000 ( $CO_2$ ,  $CO$ , cząsteczki kurzu), SE1000 (mikrofon, buzzer, kontaktron, czujnik ruchu), EM1000 (temperatura, wilgotność, światło widzialne i IR, akcelerometr, ciśnienie) mają swoje własne pliki `*.nc` definiujące dostępne moduły. Pliki „driverów” można pobrać ze stron <http://www.advanticsys.com/shop/> na podstronach poszczególnych produktów. Po rozpakowaniu wszystkich powinniśmy uzyskać podkatalogi:

gdzie to najlepiej rozpakować?  
w /opt/ti-nyOS? podkatalog drivers?

```
1 drwx----- 2 ts ts 4096 2012-07-20 01:14 MTS_AR1000
2 drwx----- 2 ts ts 4096 2011-09-14 17:19 MTS_DS1000
3 drwx----- 2 ts ts 4096 2012-07-20 01:15 MTS_EM1000
4 drwx----- 2 ts ts 4096 2011-05-13 09:16 MTS_SE1000
```

A w nich odpowiednio pliki analogiczne do poniższych:

```
1 -rwxr-xr-x 1 ts ts 4112 2011-09-14 17:18 MTS_AR1000C.nc
2 -rwxr-xr-x 1 ts ts 4101 2011-09-14 17:19 MTS_AR1000P.nc
3 -rwxr-xr-x 1 ts ts 0 2008-07-19 00:30 .sensor
```

opisać instalację XM-1000

Osobnego drzewa konfiguracyjnego wymagają węzły XM-1000

### 1.3 Biblioteka programów w C

Po instalacji warto od razu skompilować bibliotekę `libmote.a` i zestaw programów do komunikacji z portem szeregowym. Programy te znajdują się standardowo w katalogu `tinynos-2.1.1/support/sdk/c/sf`:

```
ts@plonk:% . /opt/tinynos-2.1.1/tinynos.sh
ts@plonk:% ./bootstrap
configure.ac:5: installing 'config-aux/install-sh'
configure.ac:5: installing 'config-aux/missing'
Makefile.am: installing 'config-aux/depcomp'
ts@plonk:% ./configure --prefix=/home/ts/
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
...
configure: creating ./config.status
config.status: creating Makefile
config.status: creating autoconf.h
config.status: executing depfiles commands
ts@plonk:% make
```

Następnie `make install` zainstaluje w podkatalogu `/home/ts/bin` program `sf` – serial forwarder. A w bieżącym zostaną ponadto skompilowane:

-rw-r--r--	1	ts	ts	84042	2012-07-20	19:59	libmote.a
-rwxr-xr-x	1	ts	ts	39965	2012-07-20	19:59	seriallisten
-rwxr-xr-x	1	ts	ts	40413	2012-07-20	19:59	serialsend
-rwxr-xr-x	1	ts	ts	43513	2012-07-20	19:59	prettylisten
-rwxr-xr-x	1	ts	ts	23320	2012-07-20	19:59	sflisten
-rwxr-xr-x	1	ts	ts	23426	2012-07-20	19:59	sfsend

## 2 Instalacja w systemie Windows

W wypadku Windows konieczna jest instalacja pakietu `Cygwin`. Można go pobrać spod adresu <http://cone.informatik.uni-freiburg.de/people/aslam/cygwin-files.zip>. Rozpakować należy go w osobnym katalogu, np. `c:\cygwin-files` i uruchomić program `setup`.

## 3 Przykładowe programy

W archiwum „startowym” znajdują się przykładowe programy:

oraz skrypt `polaczenie.py` uzupełniający działanie programów `RadioSerialComm`.

```
Blink
BlinkToRadio
RadioSerialComm
RadioSerialComm_telosb_sensor
```

### 3.1 Kompilacja programu `Blink`

Zawartość katalogu:

```
-rw-r--r-- 1 ts ts 2118 2009-10-26 09:34 BlinkAppC.nc
-rw-r--r-- 1 ts ts 2392 2008-06-26 05:38 BlinkC.nc
drwxr-xr-x 6 ts ts 4096 2012-04-24 12:16 build
-rw-r--r-- 1 ts ts 42 2006-07-12 18:58 Makefile
-rw-r--r-- 1 ts ts 678 2006-12-12 20:22 README.txt
```

1. W każdym przypadku należy rozpocząć od wczytania zmiennych środowiskowych, chyba że dopisaliśmy je do swoich skryptów startowych logowania (plik `.bashrc`):

```
ts@plonk:% . /opt/tinyos-2.1.1/tinyos.sh
Setting up for TinyOS 2.1.1
```

Komenda „`.`” (kropka), podobnie jak `source` powoduje wczytanie zmiennych w bieżącej powłóce, a nie w podpowłóce.

2. Wpisujemy `make` z nazwą platformy sprzętowej, na której chcemy zainstalować kompilowany program. Do wyboru mamy:

```
make iris
```

lub

```
make telosb
```

Poprawna kompilacja kończy się wydrukowaniem statystyki zajętości pamięci RAM/ROM i komunikatem ”writing TOS image”.

3. Aby zainstalować program na płytce węzła podłączamy go przez programator i wpisujemy odpowiednio:

```
make iris install mib510,/dev/ttyUSB0
```

albo

```
telosb install.7 bsl,/dev/ttyUSB0
```

W przypadku węzłów TelOS sufix `.7` oznacza identyfikator nadawany węzłowi w trakcie instalacji (można oczywiście nadać inny). W obu przypad-

kich podać należy właściwe urządzenie służące do komunikacji z węzłem. Do sprawdzenia, jakie węzły są dostępne służy program `motelist`:

```
ts@laptopok:~% motelist
```

Reference	Device	Description
MFV60PYQ	/dev/ttyUSB0	FTDI MTM-CM5000MSP
MFV66TKI	/dev/ttyUSB1	FTDI MTM-CM5000MSP
FTVJDHFK	/dev/ttyUSB2	FTDI MTI-USB1000
(none)	/dev/ttyUSB3	FTDI USB <-> Serial

W przykładzie powyżej zostały wykryte 2 węzły TelOS CM5000, jeden programator USB1000 (z dołączonym węzłem lub nie) oraz nierozpoznany z typu węzeł podłączony do portu `ttyUSB3` (w tym przypadku XM-1000 wymagający instalacji dodatkowych plików). Węzły Iris i MicaZ mogą nie zostać wykryte w ten sposób. W razie wątpliwości najlepiej wylistować zawartość katalogu `/dev` sortując pliki po dacie modyfikacji (`ls -lart`) lub przeglądając komunikaty systemowe jądra pojawiające się po podłączeniu programatora:

```
1 dmesg | less
2 [5662.613775] usb 5-2: new full speed USB device using
   uhci_hcd and address 2
3 [5662.826367] usb 5-2: configuration #1 chosen from 1 choice
4 [5662.833304] ftdi_sio 5-2:1.0: FTDI USB Serial Device
   converter detected
5 [5662.833331] usb 5-2: Detected FT2232C
6 [5662.833334] usb 5-2: Number of endpoints 2
7 [5662.833336] usb 5-2: Endpoint 1 MaxPacketSize 64
8 [5662.833338] usb 5-2: Endpoint 2 MaxPacketSize 64
9 [5662.833340] usb 5-2: Setting MaxPacketSize 64
10 [5662.834323] usb 5-2: FTDI USB Serial Device converter now
   attached to ttyUSB0
11 [5662.838300] ftdi_sio 5-2:1.1: FTDI USB Serial Device
   converter detected
12 [5662.838323] usb 5-2: Detected FT2232C
13 [5662.838325] usb 5-2: Number of endpoints 2
14 [5662.838327] usb 5-2: Endpoint 1 MaxPacketSize 64
15 [5662.838329] usb 5-2: Endpoint 2 MaxPacketSize 64
16 [5662.838331] usb 5-2: Setting MaxPacketSize 64
17 [5662.839446] usb 5-2: FTDI USB Serial Device converter now
   attached to ttyUSB1
```

W przedstawionym przykładzie podłączony pprogramator został wykryty jako urządzenia `/dev/ttyUSB0` i `/dev/ttyUSB1`. Urządzenia TelOS widoczne są jako jeden port (służący zarówno do programowania jak i komunikacji z węzłem), natomiast urządzenia Iris zgłaszają się jako 2 niezależne porty – niższy służy do programowania węzła, a wyższy do komunikacji z nim (należy otworzyć port jak każde inne urządzenie typu port szeregowy z prędkością pracy 57600 bps, 1 bit stopu, 8 bitów danych, bez parzystości). Węzły TelOS programowane są z prędkością 38400 bps, natomiast do komunikacji w trakcie działania aplikacji TinyOS używana jest prędkość



115200 bps.

Programu `motelist` można też używać w celu automatyzacji wykrywania węzłów przy instalowaniu programów. W tym celu lepiej posłużyć się opcją `-c`, wymuszającą format danych łatwiejszy do sparsowania:

```
ts@laptop:~% motelist -c
FTVJDHFK ,/dev/ttyUSB0 ,FTDI MTI-USB1000
MFV69BB6 ,/dev/ttyUSB1 ,FTDI MTM-CM5000MSP
```

## 4 Programowanie w TinyOS

### 4.1 Typowe problemy na starcie

Domyślnie komunikacja radiowa odbywa się na kanale XX. NetServ projektu WSN domyślnie nasłuchuje na kanale 12. Aby w programie TinyOS-owym wybrać kanał komunikacyjny, do pliku `Makefile` danego projektu należy dodać jeden z poniższych wpisów (albo oba):

```
PFLAGS += -DRF230_DEF_CHANNEL=12 # Węzły Iris
PFLAGS += -DCC2420_DEF_CHANNEL=12 # Węzły TelOSB i MicaZ
```

kanal

czy można zmienić kanał w trakcie działania programu? jak?

## 5 Log z instalacji Linux

```
1 ts@plonk:# apt-get install tinynos
2 Czytanie list pakietów... Gotowe
3 Budowanie drzewa zależności
4 Odczyt informacji o stanie... Gotowe
5 Note, selecting 'tinynos-2.1.1' instead of 'tinynos'
6 The following packages were automatically installed and are no
  longer required:
7   libjna-java librxtx-java libxerces-c3.1
8 Aby je usunąć należy użyć "apt-get autoremove".
9 Zostaną zainstalowane następujące dodatkowe pakiety:
10  avr-binutils-tinynos avr-gcc-tinynos avr-libc-tinynos avr-optional-
   tinynos avr-tinynos avr-tinynos-base avrdude-tinynos default-jdk
   default-jre default-jre-headless deputy-tinynos libxt-dev
   msp430-binutils-tinynos msp430-gcc-tinynos msp430-libc-tinynos
   msp430-tinynos msp430mcu-tinynos nesc openjdk-6-jdk tinynos
   -2.1.1 tinynos-base tinynos-required-all tinynos-required-avr
   tinynos-required-msp430 tinynos-tools
11 Sugerowane pakiety:
12   openjdk-6-demo openjdk-6-source visualvm
13 Polecane pakiety:
14   java-sdk
15 Następujące pakiety zostaną USUNIĘTE:
16   arduino arduino-core avr-libc avrdude binutils-avr gcc-avr
17 Zostaną zainstalowane następujące NOWE pakiety:
18  avr-binutils-tinynos avr-gcc-tinynos avr-libc-tinynos avr-optional-
   tinynos avr-tinynos avr-tinynos-base avrdude-tinynos default-jdk
   default-jre default-jre-headless deputy-tinynos libxt-dev
   msp430-binutils-tinynos msp430-gcc-tinynos msp430-libc-tinynos
   msp430-tinynos msp430mcu-tinynos nesc openjdk-6-jdk tinynos
   -2.1.1 tinynos-base tinynos-required-all tinynos-required-avr
   tinynos-required-msp430 tinynos-tools
```

## 6 Przydatne linki

Różne przydatne linki:

- TinyOS Wiki  
<http://en.wikipedia.org/wiki/TinyOS>
- Installing TinyOS 2.1.1:  
[http://docs.tinyos.net/tinywiki/index.php/Installing\\_TinyOS\\_2.1.1](http://docs.tinyos.net/tinywiki/index.php/Installing_TinyOS_2.1.1)
- Dokumentacja funkcji TinyOS (2.1.0) / Iris/MicaZ:  
<http://www.tinyos.net/tinyos-2.1.0/doc/nesdoc/iris/>
- TinyOS Tutorials  
[http://docs.tinyos.net/tinywiki/index.php/TinyOS\\_Tutorials](http://docs.tinyos.net/tinywiki/index.php/TinyOS_Tutorials)
- Pakiety MSP430 do ściągnięcia:  
<http://tinyos.stanford.edu/tinyos/dists/ubuntu.old/pool/main/m/>
- Całe mnóstwo przykładowych programów na TinyOS:  
[http://docs.tinyos.net/tinywiki/index.php/TinyOS\\_2.x\\_index\\_of\\_contributed\\_code](http://docs.tinyos.net/tinywiki/index.php/TinyOS_2.x_index_of_contributed_code)
- Lista dyskusyjna TinyOS-help  
<http://mail.millennium.berkeley.edu/pipermail/tinyos-help/2011-August/052187.html>
- Dash 7  
<http://en.wikipedia.org/wiki/DASH7>,  
<http://www.dash7.org/>