

Programmierung der UCMote mini & UCMote Proton Base

A Programmierung der UCMote mini Sensorknoten

1.a Programm kompilieren

> **make ucmini**

1.b zum Übergeben der Knoten-ID (z.B. 5) folgenden Befehl nutzen

> **make ucmini install,5**

3. Den UCMote mini via MicroUSB-Kabel an den PC anschließen

4.a Innerhalb der nächsten 10 Sekunden den folgenden Befehl eingeben,
sonst gibt es einen Timeout im Bootloader (siehe 5.):

> **avrdude -cavr109 -pm128rfa1 -P/dev/ttyUSB0 -Uflash:w:build/ucmini/main.srec:a -b57600**

4.b Wenn eine Knoten-ID (siehe 1.b) übergeben werden soll, muss das entsprechende
main.srec.out-File übergeben werden:

> **avrdude -cavr109 -pm128rfa1 -P/dev/ttyUSB0 -Uflash:w:build/ucmini/main.srec.out-5:a -b57600**

5. Falls die Fehlermeldung kommt:

> **Connecting to programmer: .avrdude: butterfly_recv(): programmer is not responding**
, muss entweder die Zieladresse angepasst werden (z.B. /dev/ttyUSB1 statt /dev/ttyUSB0)
oder es gab einen Timeout im Bootloader. In beiden Fällen muss der Sensorknoten
vom PC getrennt werden (ein einfaches Reset half bei mir nicht) und ab 3. erneut versuchen.

6. Dies ist die Ausgabe bei einem erfolgreichen Durchlauf:

```
sd@ubuntu:/opt/tinyos-2.x/apps/Blink$ make ucmini
```

```
mkdir -p build/ucmini
```

```
compiling BlinkAppC to a ucmini binary
```

```
ncc -o build/ucmini/main.exe -Os -fnesc-separator=__ -Wall -Wshadow -Wnesc-all -target=ucmini -  
fnesc-cfile=build/ucmini/app.c -board= -DDEFINED_TOS_AM_GROUP=0x22 --param max-inline-insns-  
single=100000 -DIDENT_APPNAME="BlinkAppC" -DIDENT_USERNAME="sd" -  
DIDENT_HOSTNAME="ubuntu" -DIDENT_USERHASH=0xa182c757L -  
DIDENT_TIMESTAMP=0x527a4290L -DIDENT_UIDHASH=0xb66b6b38L -fnesc-dump=wiring -fnesc-  
dump='interfaces(!abstract())' -fnesc-dump='referenced(interfacedefs, components)' -fnesc-  
dumpfile=build/ucmini/wiring-check.xml BlinkAppC.nc -lm
```

```
compiled BlinkAppC to build/ucmini/main.exe
```

```
7010 bytes in ROM
```

```
233 bytes in RAM
```

```
avr-objcopy --output-target=srec build/ucmini/main.exe build/ucmini/main.srec
```

```
avr-objcopy --output-target=ihex build/ucmini/main.exe build/ucmini/main.ihex
```

```
writing TOS image
```

```
sd@ubuntu:/opt/tinyos-2.x/apps/Blink/$ avrdude -cavr109 -pm128rfa1 -P/dev/ttyUSB0 -
Uflash:w:build/ucmini/main.srec:a -b57600
Connecting to programmer: .
Found programmer: Id = "UC BOOT"; type = S
    Software Version = 0.3; Hardware Version = 2.0
Programmer supports buffered memory access with buffersize=256 bytes.
Programmer supports the following devices:
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.01s
avrdude: Device signature = 0x1ea701
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
    To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "main.srec"
avrdude: input file main.srec auto detected as Motorola S-Record
avrdude: writing flash (7010 bytes):
Writing | ##### | 100% 1.44s
avrdude: 7010 bytes of flash written
avrdude: verifying flash memory against main.srec:
avrdude: load data flash data from input file main.srec:
avrdude: input file main.srec auto detected as Motorola S-Record
avrdude: input file main.srec contains 7010 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 1.38s
avrdude: verifying ...
avrdude: 7010 bytes of flash verified
avrdude done. Thank you.
sd@ubuntu:/opt/tinyos-2.x/apps/Blink/
```

B Programmierung der UCMote Proton Base Sensorknoten

0. Proton Base Plattform in TinyOS hinzufügen. Dazu die folgenden Dateien von

<http://www.ucmote.com/images/uploaded/File/ucbase.zip>

herunterladen und in die entsprechenden Verzeichnisse kopieren.

1.a Programm kompilieren

> make ucbase

1.b zum Übergeben der Knoten-ID (z.B. 5) folgenden Befehl nutzen

> make ucbase install,5

3. Den UCMote Proton Base via USB an den PC anschließen

4.a Innerhalb der nächsten 10 Sekunden den folgenden Befehl eingeben,

sonst gibt es einen Timeout im Bootloader (siehe 5.):

> avrdude -cavr109 -pm128rfa1 -P/dev/ttyACM0 -Uflash:w:build/ucbase/main.srec:a -b57600

4.b Wenn eine Knoten-ID (siehe 1.b) übergeben werden soll, muss das entsprechende

main.srec.out-File übergeben werden:

> avrdude -cavr109 -pm128rfa1 -P/dev/ttyACM0 -Uflash:w:build/ucbase/main.srec.out-5:a -b57600

5. Falls die Fehlermeldung kommt:

> Connecting to programmer: .avrdude: butterfly_recv(): programmer is not responding

, muss entweder die Zieladresse angepasst werden (z.B. /dev/ttyACM1 statt /dev/ttyACM0)

oder es gab einen Timeout im Bootloader. In beiden Fällen muss der Sensorknoten

vom PC getrennt werden (ein einfaches Reset half bei mir nicht) und ab 3. erneut versuchen.

6. Dies ist die Ausgabe bei einem erfolgreichen Durchlauf:

```
sd@ubuntu:/opt/tinyos-2.x/apps/Blink$ make ucbase
```

```
mkdir -p build/ucbase
```

```
compiling BlinkAppC to a ucbase binary
```

```
ncc -o build/ucbase/main.exe -Os -fnesc-separator=__ -Wall -Wshadow -Wnesc-all -target=ucbase -
```

```
fnesc-cfile=build/ucbase/app.c -board= -DDEFINED_TOS_AM_GROUP=0x22 --param max-inline-
```

```
insns-single=100000 -DIDENT_APPNAME=\"BlinkAppC\" -DIDENT_USERNAME=\"sd\" -
```

```
DIDENT_HOSTNAME=\"ubuntu\" -DIDENT_USERHASH=0xa182c757L -
```

```
DIDENT_TIMESTAMP=0x527a4290L -DIDENT_UIDHASH=0xb66b6b38L -fnesc-dump=wiring -fnesc-
```

```
dump='interfaces(labstract())' -fnesc-dump='referenced(interfacedefs, components)' -fnesc-
```

```
dumpfile=build/ucbase/wiring-check.xml BlinkAppC.nc -lm
```

```
compiled BlinkAppC to build/ucbase/main.exe
```

```
7010 bytes in ROM
```

```
233 bytes in RAM
```

```
avr-objcopy --output-target=srec build/ucbase/main.exe build/ucbase/main.srec
```

```
avr-objcopy --output-target=ihex build/ucbase/main.exe build/ucbase/main.ihex
```

```

writing TOS image
sd@ubuntu:/opt/tinyos-2.x/apps/Blink/$ avrdude -cavr109 -pm128rfa1 -P/dev/ttyUSB0 -
Uflash:w:build/ucbase/main.srec:a -b57600
Connecting to programmer: .
Found programmer: Id = "UC BOOT"; type = S
    Software Version = 0.3; Hardware Version = 2.0
Programmer supports buffered memory access with buffersize=256 bytes.
Programmer supports the following devices:
avrdude: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.01s
avrdude: Device signature = 0x1ea701
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
    To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "main.srec"
avrdude: input file main.srec auto detected as Motorola S-Record
avrdude: writing flash (7010 bytes):
Writing | ##### | 100% 1.44s
avrdude: 7010 bytes of flash written
avrdude: verifying flash memory against main.srec:
avrdude: load data flash data from input file main.srec:
avrdude: input file main.srec auto detected as Motorola S-Record
avrdude: input file main.srec contains 7010 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 1.38s
avrdude: verifying ...
avrdude: 7010 bytes of flash verified
avrdude done. Thank you.
sd@ubuntu:/opt/tinyos-2.x/apps/Blink/

```

Weitere Hinweise: http://ucmote.uniconp.hu/images/uploaded/File/UCMote%20mini%20-%20User%20Manual%20v0_1.pdf (**User's manual UCMote Mini (English)**)