

Portierung von TinyOS Blip2.0 auf die UCMotes Hardware

0. Ausgangsrepository für die Portierung:

<https://github.com/mmaroti/tinyos-main/tree/blip2-patch>

1. Hinzufügen der aktuellen Plattform und Chip-Definitionen aus dem *Main development repository for TinyOS* (<https://github.com/tinyos/tinyos-main>)

d.h. Ersetzen/Hinzufügen der folgenden Verzeichnisse/Dateien:

```
/tos/platforms/ucmini  
/tos/platforms/ucbase  
/tos/platforms/ucprotonb  
/tos/chips/atm128rfa1 -> UDPEcho fehlerhaft  
/tos/chips/rf212  
/support/make/ucmini.target  
/support/make/ucbase.target  
/support/make/ucprotonb.target
```

2. Folgendes Configuration File anpassen:

/tos/system/**MultiplexedReadC.nc**:

```
provides interface Read<width_t>[uint8_t client];  
    uses {  
        interface Read<width_t> as Service;  
        interface Resource[uint8_t client];  
+        interface ArbiterInfo;  
    }  
}
```

3. Kopieren der Dateien **SerialAutoControlC.nc** und **SerialAutoControlP.nc** von der "alten" ucmini-Plattform in die aktuellen Plattformdefinitionen von ucmini, ucbase, ucprotonb unter /tos/platforms/ucmini, /tos/platforms/ucbase und /tos/platforms/ucprotonb, da diese im Development Repository fehlen.

- ~~4. Kopieren der Datei **RFA1leee154MessageC.nc** von der "alten" atm128rfa1-Chipdefinition in die aktuelle unter /tos/chips/atm128rfa1/radio, da diese im Development TinyOS main Tree fehlt.~~

5. Kopieren der Datei **LocalleeeEui64C.nc** von der ucmini-Plattform in die ucbase- und ucprotonb-Plattform

6. Hinzufügen der ReadLqi-Configuration für den RFA1-Chip. Dazu unter /tos/lib/net/blip/platform die Datei **RFA1ReadLqiC.nc** mit folgendem Inhalt anlegen:

```
uint16_t adjustLQI(uint8_t val) {  
    uint16_t result = 64 - (val / 4);  
    result = (((result * result) >> 3) * result) >> 3; // result = (result ^ 3) / 64  
    return result;  
}
```

```

}

module RFA1ReadLqiC {
  provides interface ReadLqi;
  uses interface PacketField<uint8_t> as SubLqi;
  uses interface PacketField<uint8_t> as SubRssi;
} implementation {
  command uint8_t ReadLqi.readLqi(message_t *msg) {
    return call SubLqi.get(msg);
  }

  command uint8_t ReadLqi.readRssi(message_t *msg) {
    return call SubRssi.get(msg);
  }
}

```

7. Bearbeiten der ReadLqi-Configuration in /tos/lib/net/blip/ReadLqiC.nc

```

-#elif defined(PLATFORM_UCMINI)
-  components RF230ReadLqiC, RFA1Ieee154MessageC;
-  ReadLqi = RF230ReadLqiC;
-  RF230ReadLqiC.SubLqi -> RFA1Ieee154MessageC.PacketLinkQuality;
-  RF230ReadLqiC.SubRssi -> RFA1Ieee154MessageC.PacketRSSI;
+##elif defined(PLATFORM_UCMINI) || defined(PLATFORM_UCBASE) || defined(PLATFORM_UCPROTONB)
+  components RFA1ReadLqiC, RFA1Ieee154MessageC;
+  ReadLqi = RFA1ReadLqiC;
+  RFA1ReadLqiC.SubLqi -> RFA1Ieee154MessageC.PacketLinkQuality;
+  RFA1ReadLqiC.SubRssi -> RFA1Ieee154MessageC.PacketRSSI;
#else
#error "No radio support is available for your platform"
#endif

```

8. Plattformen ucbase und ucprotonb in Blip-spezifische Dateien einfügen. Dazu in folgenden Dateien die "if defined"-Platform-Anweisungen anpassen:

```

/tos/lib/net/blip/IPAddressP.nc
/tos/lib/net/blip/IPDispatchC.nc
/tos/lib/net/blip/IPDispatchP.nc
/tos/lib/net/blip/Ieee154AddressC.nc
/tos/lib/net/blip/Ieee154AddressP.nc

```

hierfür sämtliche "`|| defined(PLATFORM_UCMINI)`"-Ausdrücke durch folgende ersetzen:
"`|| defined(PLATFORM_UCMINI) || defined(PLATFORM_UCBASE) || defined(PLATFORM_UCPROTONB)`"

9. In der Datei /tos/lib/net/blip/IPDispatchP.nc im Event **BareReceive.receive**(message_t *msg) {} folgende Abfrage hinzufügen, um Duplikate von Nachrichten zu verhindern.

```

/* unpack and Glowpan headers */

```

```

lowmsg.data = buf;
lowmsg.len = len;
lowmsg.headers = getHeaderBitmap(&lowmsg);
if (lowmsg.headers == LOWMSG_NALP) {
goto fail;
}

+ // discard packets that are not addressed to the local node address or the broadcast address
+ if (!(frame_address.ieee_dst.ieee_addr.saddr == AM_BROADCAST_ADDR ||
frame_address.ieee_dst.ieee_addr.saddr == TOS_NODE_ID)) {
+ return msg;
+ }

if (hasFrag1Header(&lowmsg) || hasFragNHeader(&lowmsg)) {
// start reassembly

```