

Rádió kezelése a TinyOS-ben

- Adást és vételt lehetővé tevő interfészek:

–Adás:

```
interface SendMsg
{
    command result_t send(uint16_t address, uint8_t length,
                           TOS_MsgPtr msg);
    event result_t sendDone(TOS_MsgPtr msg, result_t success);
}
address: fogadó címe
length : üzenet hossza
msg     : üzenetre mutató adott típusú pointer
```

–Vétel:

```
interface ReceiveMsg
{
    event TOS_MsgPtr receive(TOS_MsgPtr m);
}
```

m : üzenetre mutató pointer.

Visszatérés: az új üzenet helyére mutató pointert kell visszatérési értéként megadni

- Ezen interface-ek paraméterezhetőek amennyiben a GenericComm modulon keresztül használjuk őket

Rádió kezelése a TinyOS-ben

- Alapvető üzenetstruktúra (AM.h fileban):

```
#define TOSH_DATA_LENGTH 29
typedef struct TOS_Msg
{
    /* The following fields are transmitted/received on the radio. */
    uint16_t addr;
    uint8_t type;
    uint8_t group;
    uint8_t length;
    int8_t data[TOSH_DATA_LENGTH];
    uint16_t crc;

    /* The following fields are not actually transmitted or received
     * on the radio! They are used for internal accounting only.
     * The reason they are in this structure is that the AM interface
     * requires them to be part of the TOS_Msg that is passed to
     * send/receive operations.
     */
    uint16_t strength;
    uint8_t ack;
    uint16_t time;
    uint8_t sendSecurityMode;
    uint8_t receiveSecurityMode;
} TOS_Msg;

typedef TOS_Msg *TOS_MsgPtr;
```

Rádió kezelése a TinyOS-ben

- Példa: CntToLedsAndRfm,
RfmToLeds alkalmazások

Alap üzenettípus (mindkét alkalmazásban deklarálván):

```
typedef struct IntMsg {  
    uint16_t val;    //küldött adat  
    uint16_t src;    //küldő címe  
} IntMsg;  
  
enum {    AM INTMSG = 4};
```

Rádió kezelése a TinyOS-ben

- Adó oldal

```
uses interface SendMsg as Send; //a SendMsg interface átnevezése Send-re
```

Üzenet küldése a program megfelelő szakaszában:

```
IntMsg *message = (IntMsg *)data.data; //Üzenet struktúra típuskonverziója  
message->val = value; //az üzenet adatmezőjébe beírjuk a küldendő értéket  
message->src = TOS_LOCAL_ADDRESS; //forrásként megadjuk a saját címünket  
call Send.send(TOS_BCAST_ADDR, sizeof(IntMsg), &data); //üzenet küldése
```

```
//Az üzenet elküldésének visszajelzése:
```

```
event result_t Send.sendDone(TOS_MsgPtr msg, result_t success){  
    return SUCCESS;  
}
```

- Vevő oldal

```
uses interface ReceiveMsg as ReceiveIntMsg; //interface átnevezése
```

Üzenet fogadása a ReceiveIntMsg.receive esemény segítségével:

```
event TOS_MsgPtr ReceiveIntMsg.receive(TOS_MsgPtr m) {  
    IntMsg *message = (IntMsg *)m->data;  
    //message->val hivatkozással elérhető a vett üzenet adatmezője  
    //message->src hivatkozással elérhető a vett üzenet küldőjének címe  
    return m;  
}
```