

```

/*
 * @author IPP HURRAY http://www.hurray.isep.ipp.pt/art-wise
 * @author Andre Cunha
 */

/* definovany fyzicke konstanty aMaxPHYPacketSize, aTurnaroundTime, INIT_CURRENTCHANNEL,
INIT_CHANNELSSUPPORTED, INIT_TRANSMITPOWER, INIT_CCA_MODE
a definovana nova struktura phyPIB, umisten v "\tos\lib\phy" */
includes phy_const;

/* umisten v "\tos\lib\phy", obsahuje dva výčty
a) phy status enumerations: PHY_BUSY, PHY_BUSY_RX, PHY_BUSY_TX, PHY_FORCE_TRX_OFF,
PHY_IDLE, PHY_INVALID_PARAMETER, PHY_RX_ON, PHY_SUCCESS, PHY_TRX_OFF, PHY_TX_ON,
PHY_UNSUPPORTED_ATTRIBUTE
b) phy PIB attributes enumerations: PHYCURRENTCHANNEL, PHYCHANNELSSUPPORTED,
PHYTRANSMITPOWER, PHYCCAMODE */
includes phy_enumerations;

// v "\tos\system", definice struktur MPDU, beacon, ACK, cmd, dest, source.. atd.
includes frame_format;

configuration Phy {
provides {
    interface StdControl as Phy_control; // konfigurace poskytuje interface StdControl
                                         // ale dava mu lokalni jmeno Phy_control

    interface PD_DATA; //
    //PLME-SAP // interfacey, které poskytuje Phy.nc
                // pomoci modulu PhyM.nc (viz nize)

    interface PLME_ED; //
    interface PLME_CCA; // umistene v "\tos\interfaces\ieee802154\phy"
    interface PLME_GET; //
    interface PLME_SET; //
    interface PLME_SET_TRX_STATE; //
    }
}

implementation
{
components Main, /* - komponenta, ktera je spustena jako prvni v aplikaci, obsahuje
Main.StdControl.init(), Main.StdControl.start(), Main.StdControl.stop()
- aplikace musi mit Main komponentu v konfiguraci, StdControl je spolecny interface
pouzivany k inicializaci a startu TinyOS komponent */

    PhyM, // dalsi dve komponenty, na ktere se tato konfigurace odkazuje
    HPLCC2420C; // umistene v "\tos\lib\phy" a "\tos\platforms\micaz"

    /*StdControl*/
    Main.StdControl -> PhyM; // muze byt Main.StdControl -> PhyM.StdControl
    // svazani a pouziti interfaceu k poskytovateli implementace
    // Main.StdControl uziva interface, ktery poskytuje komponenta PhyM

    //poskytuje
    Phy_control = PhyM; // Phy_control (StdControl) interface poskytovany Phy je
    // ekvivalentni s implementaci v PhyM

    PhyM.HPL_Control->HPLCC2420C;
    /* pozor HPLCC2420C neobsahuje HPL_Control ale StdControl
    je to jako PhyM.StdControl->HPLCC2420C.StdControl ale PhyM uziva
    interface StdControl jako HPL_Control a obsahuje HPL_Control.init(),
    start(), stop() */
    // HPLCC2420C poskytuje interfacey: StdControl, HPLCC2420,
    // PLCC2420FIFO, HPLCC2420RAM

```

```

//poskytuje
PD_DATA = PhyM; //
//PLME-SAP //
PLME_ED = PhyM; // PD_DATA, PLME_ED, PLME_CCA, PLME_GET, PLME_SET,
// PLME_SET_TRX_STATE
PLME_CCA = PhyM; // interfacey jsou ekvivalentni s implementaci v PhyM
PLME_GET = PhyM; //
PLME_SET = PhyM; //
PLME_SET_TRX_STATE = PhyM;

//HPLCC2420
PhyM.HPLCC2420 -> HPLCC2420C.HPLCC2420; /* svazani a pouziti interfaceu k
poskytovateli implementace
PhyM.xxx uziva interfacey,
kttere poskytuje komponenta HPLCC2420C */
PhyM.HPLCC2420RAM -> HPLCC2420C.HPLCC2420RAM;
PhyM.HPLCC2420FIFO -> HPLCC2420C.HPLCC2420FIFO;

```

}