

Control of PLC hunt range via MDIO

Bill Keasler

MDIO control of PLC acquisition/hunt range

To improve the support for worldwide deployment of EPoC CNU devices I propose that a small set of MDIO registers be allocated to control the grid of frequencies used during hunt/acquisition of the physical link channel

- Regional/Geographic differences
- Possible evolution from low-split to high-split

Registers (names are placeholders) (16-bit)

- **PLC_SRCH_FREQ_START (R/W)**
 - 50kHz increments (0 to 3.3 GHz)
- **PLC_SRCH_FREQ_STEP (R/W)**
 - 125 Hz increments (0 to 8.2 MHz)
- **PLC_SRCH_ENDCNT (R/W)**
 - Number of grid frequencies in search range
- **PLC_SRCH_CNTRL (R/W)**
- **PLC_SRCH_STATUS (RO)**

Comments on frequency resolution

- The frequency spacing for the raster of frequencies searched is determined by the register **PLC_SRCH_FREQ_STEP**
 - $\Delta f = \text{PLC_SRCH_FREQ_STEP} \cdot 125 \text{ Hz}$
- The initial frequency is determined by the register **PLC_SRCH_FREQ_START**
 - $\text{Initial_freq} = \text{PLC_SRCH_FREQ_START} \cdot 50000 \text{ Hz}$
- The number of grid points searched is simply **PLC_SRCH_ENDCNT**

Pseudo code to illustrate register use

```
long plc_hunt()  
{  
#define    RES_START 50000                /* 50 kHz resolution on start frequency */  
#define    RES_STEP 125                  /* 125 Hz resolution on frequency grid */  
  
    int i;  
    int n;  
    int start;  
    int step;  
    long dF;  
    long freq;  
  
    start = read_mdio_reg(plc_srch_freq_start);  
    step  = read_mdio_reg(plc_srch_freq_step);  
    n     = read_mdio_reg(plc_srch_endcnt);  
  
    freq = RES_START * start;  
    dF   = RES_STEP * step;  
  
    for(i=0; i<n; i++){  
        if( plc_trial(freq) ){  
            return freq;                /* return frequency on success */  
        } else {  
            freq += dF;  
        }  
    }  
    return PLC_ACQ_FAIL;                /* fall through on failure */  
                                        /* return failure code */  
}
```

Thank You