

```

#include <LiquidCrystal.h>
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// 2,5 GHz counter na 512x prescaler incl -50 tot +10 dBm powermeet aanduiding via 0-5V uitgang
AD8317 //
// mei 2013 Fred, PA4TIM, met dan aan Pros //
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 9, 6, 4, 3, 2);

// globale variabele
float power = 0; // power variabele
float powerRAW = 0; // power variabele
float correctie = 1; // eventuele correctie voorpower
int counterPin = 5; // ingang counter vanaf prescaler
int powerPin = 0; // analoge ingang voor powermeter
int klokPin = 8; // Op PB0 (Digital pin 8) komt een klok van 1Hz
volatile unsigned long tellerstand;
volatile unsigned char intflag;
char tekst[32];
register unsigned char ovf_cnt asm("r2");

void setup()
{
    // toekennen in en output pinnen Arduino duemilanove met atmel 328
    pinMode(counterPin, INPUT);
    pinMode(powerPin, INPUT);
    pinMode(klokPin, INPUT);

    lcd.begin(16, 2); // 16x2 lcd display initialisatie
}

// Berekenen van de 0-5V powermeter, de uitgang van de 1MHz to 7 GHz AD8317 is dmv opamps
geinvertreed en 0-5V gemaakt
// 5V=1024 punten, 1 punt is 4,88828mV, hij doet dan 83.33mV/dB nu is 5V = +10 dBm 0V =
-50dBm, 1dB = 17.04757 punten
// power = -1*(1024 - powerRAW) / 17.04757 + 10
// als powerRAW >1023 dan powerRAW--1 if powerRAW < 1 dan powerRAW ++1 om delen door
nul te voorkomen.
// 0V = 0 punten is -50dBm, 5V = 1024 punten is +10 dBm

ISR(TIMER1_CAPT_vect)
{
    tellerstand = (unsigned long) (ICR1) + ((unsigned long) (ovf_cnt) * 0x10000UL);
    intflag = 0x05;
}

// Timer1-overflow interrupt

```

```

ISR(TIMER1_OVF_vect, ISR_NAKED)
{
    ovf_cnt++;
    reti();
}
// ISR_NAKED, want we moeten enkel
// een register met 1 verhogen
// Niet vergeten bij ISR_NAKED!

```

```

// Plaats een unsigned long (max = 9999999) in een string, voorzien van dp en komma
// 1234567 wordt dan 1.234,567

```

```

void UL2string(unsigned long getal, char *string)
{
    unsigned long  deeltal = 1000000;
    char          tmp, cnt, zeroflag = 0;

    for (cnt = 7; cnt > 0; cnt--) {
        if ((cnt == 6) && (zeroflag == 1)) {
            *string = '.';
            string++;
        } else if (cnt == 3) {
            if (zeroflag == 0) {
                *string = '0';
                string++;
                zeroflag = 1;
            }
            *string = ',';
            string++;
        }
        tmp = 0;
        while (getal >= deeltal) {
            tmp++;
            getal -= deeltal;
        }
        if (tmp > 0) {
            zeroflag = 1;
        }
        afgedrukt
        *string = tmp + '0';
    } else if (zeroflag == 1) {
        *string = '0';
    } else {
        *string = ' ';
    }
    deeltal /= 10;
    decade
    string++;
}
/* *string = ' ';
string++;
*string = 'M';
string++;

```

```

*string = 'H';
string++;
*string = 'z';
string++;
*string = 0;    /* // string netjes afsluiten met een 0!!!

}

void loop()
{
  unsigned long  frequentie, vorige_tellerstand = 0;

  CLKPR = _BV(CLKPCE);
  CLKPR = 0; // Set max. system-clk

  // Timer 1 instellen
  TCCR1A = 0; // Gewoon tellen, niets meer
  TIMSK1 = _BV(ICIE1) | _BV(TOIE1); // Enable ICP-interrupt en OVF-
interrupt

  // Een opgaande flank op de ICP1-pin genereert een Input Capture Interrupt
  //TCCR1B = _BV(CS10) | _BV(CS11) | _BV(CS12) | _BV(ICES1) | _BV(ICNC1); // T1 (PD5
) = CLK

  TCCR1B = _BV(CS10) | _BV(CS11) | _BV(CS12) | _BV(ICES1); // T1 (PD5) = CLK

  //PORTD |= _BV(PD5); // Pull-up's inschakelen zodat hij niet wild gaat doen, (dit werkt niet bij
Arduino, DD5 was not declared in this scope)
  //PORTB |= _BV(PB0); // als er niets aangesloten is op PD5 of PB0 (dit werkt niet bij
Arduino, DD5 was not declared in this scope)

  sei(); // Global interrupts toestaan

  while (1)
  {
    LOOP:
    if (intflag == 0x05)
    {
      intflag++;
      // Om een onverklaarbare reden moet ik intflag twee maal testen
      // Zonder dat gaat hij regelmatig in de fout. Optimalisatie-probleempje?
      if (intflag == 0x0B)
      {
        goto LOOP;
      }

      if (tellerstand > vorige_tellerstand)
      {
        frequentie = tellerstand - vorige_tellerstand;
      }
    }
  }
}

```

```

else if (tellerstand == vorige_tellerstand)
{
    frequentie = 0;
}
else
{
    // tellerstand < vorige_tellerstand
    // We kunnen maar tellen van 0 tot en met 0xFFFFFFFF
    // Voorbeeld: vorige_tellerstand = 0xFFFFF80; tellerstand = 0x0000AB
    // frequentie = (0x1000000 - 0xFFFFF80) + 0x0000AB
    frequentie = 0x1000000UL - vorige_tellerstand;
    frequentie += tellerstand;
}
frequentie =( frequentie * 128) / 250; // frequentie aanpassen ivm 512x prescaler
vorige_tellerstand = tellerstand;
UL2string(frequentie, tekst);
lcd.setCursor(3, 0);
lcd.print(tekst);
lcd.setCursor(13,0);
lcd.print("MHz");

delay(20);
//powermeet routine
powerRAW = analogRead(powerPin);
power = (-1*(1024-powerRAW)/17.0475 + 10 )*correctie; // berekenen dB -50 tot -10dBm
delay(20);
lcd.setCursor(0, 1);
lcd.print("Power: ");
lcd.setCursor(7, 1);
lcd.print(power);
lcd.setCursor(12, 1);
lcd.print(" dBm");
delay(20);
intflag = 0x0A;
}
}
}

```