

```

1 // Test_pilot_tone_RXTX_OZ1LN_25-04-2018_b
2 //
3 // Display forbundet med D12 til RS, D11 til E, RW til stel.
4 // D10 til display D4, D9 til D 5, D8 til D6 og D7 til D7.
5 //
6 #include <LiquidCrystal.h>
7 LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
8 //-----
9 boolean TX_Tast = 1;
10 const int Test_Tast = 3;
11 const int tone_ud = 5;
12 const byte ledPin = 13; // LED på Nano
13 const byte interruptPin = 2; // Frekvens input
14 const byte countTime = 100; // tiden hvor interrupt er åben og der tælles
15 float Antal_micros_ialt;
16 float Antal_micros_pr_puls;
17 float uKorrigeret_Frekvens;
18 float Frekv;
19 float FrekvensKonstant = 1001665.0; // kompensering. Tjek med nøjagtig tæller
20 volatile unsigned long StartPeriode; // begyndelse på micros-optælling
21 volatile unsigned long SlutPeriode; // slutning på micros-optælling
22 volatile unsigned long AntalPerioder; // antal perioder, som blev nået på countTime
23 //-----
24 void setup()
25 {
26   lcd.begin(16, 2); // start af LCD display 16 x 2 karakterer
27   pinMode(tone_ud, OUTPUT);
28   pinMode(ledPin, OUTPUT); // LED på Nano
29   pinMode(interruptPin, INPUT); // Frekvens input
30   pinMode(Test_Tast, INPUT);
31 }
32 //-----
33 void HentFrekvens() // tænd interrupt og tæl
34 {
35   AntalPerioder = 0; // så fanger vi den første RISING flanke
36   attachInterrupt(digitalPinToInterrupt(interruptPin), pin2ISR, RISING);
37   // der trigges på stigende flanke, og interrupt rutine pin2ISR køres
38   delay(countTime); // her tælles micros
39   detachInterrupt(interruptPin); // sluk interrupt
40 }
41 //-----
42 void loop() {
43   TX_Tast = !digitalRead(Test_Tast);
44   if (TX_Tast)
45   {
46     tone(tone_ud, 885);
47   }
48   else
49   {
50     noTone(tone_ud);
51     HentFrekvens(); // kald HentFrekvens funktionen
52     Antal_micros_ialt = (SlutPeriode - StartPeriode);
53     Antal_micros_pr_puls = ((Antal_micros_ialt)/(AntalPerioder-2));
54     uKorrigeret_Frekvens = ((AntalPerioder - 2)/Antal_micros_ialt);
55     Frekv = (FrekvensKonstant * uKorrigeret_Frekvens);
56     // -----
57     if (Frekv <= 30 || Frekv >= 150) Frekv = 0; // Det er jo en pilottone tæller ;o)
58     if (Frekv >= 88 && Frekv <= 90){
59       digitalWrite(ledPin,HIGH); // korrekt Frekvens detekteret: tænd LED
60       // gør noget, så radio åbner
61     }
62     else digitalWrite(ledPin,LOW); // sluk LED
63     // Udskrifter -----
64     lcd.setCursor(0,0); // sæt cursor på 1. linje, 1 karakter
65     lcd.print("OZ1LN"); // skriv
66     lcd.setCursor(6,0); // sæt cursor på 1. linje, 7 karakter
67     lcd.print("Fkv: "); // skriv
68
69     if (Frekv == 0)
70     {
71       lcd.print("----"); // skriv
72       lcd.print(" "); // skriv
73       lcd.setCursor(0,1); // sæt cursor på 2. linje, 1. karakter

```

```

74     lcd.print("ikke gyldig tone");
75 }
76 else
77 {
78     lcd.print(Frekv);           // skriv Frekvens
79     lcd.print(" ");            // skriv
80     lcd.setCursor(0,1);        // sæt cursor på 2. linje, 1. karakter
81     lcd.print("                "); // blank 2. linje
82     lcd.setCursor(0,1);        // sæt cursor på 2. linje, 1. karakter
83     lcd.print(AntalPerioder - 2); // skriv antal RISING flanker
84     lcd.setCursor(2,1);        // sæt cursor på 2. linje, 4. karakter
85     lcd.print(Antal_micros_ialt); // skriv antal talte mikrosekunder
86     lcd.setCursor(11,1);       // sæt cursor på 2. linje, 13. karakter
87     lcd.print(Antal_micros_pr_puls); // mikrosekunder i en bølgelængde
88 }
89 }
90 }
91 ////////////////////////////////////////
92 //DETTE KØRES HVER GANG DER KOMMER EN INTERRUPT
93 void pin2ISR() {
94     if (AntalPerioder == 1) StartPeriode = micros(); // start tælling
95     else SlutPeriode = micros();                     // slut tælling
96     AntalPerioder = AntalPerioder + 1;
97 }
98 ////////////////////////////////////////
99
100
101
```