```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <EEPROM.h>
#include <Talkie.h>
#include <Vocab_US_Large.h>
Talkie voice;

/* ── Pin map ──────────────────────────────────────────────── */
const int CLbarPin  = 2;      // HIGH=C, LOW=L
const int RmodePin  = 8;      // LOW ⇒ Resistance mode
const int RelayPin  = 9;      // C2 relay
const int DecrPin   = 4;      // ↓ CF nudge
const int OscSigPin = 5;      // oscillator signal
const int IncrPin   = 6;      // ↑ CF nudge
const int CalLkPin  = 7;      // jumper enables nudging
const int SensePin  = A0;     // divider sense for Rx

/* ── Constants ────────────────────────────────────────────── */
const float FourPiSqrd = 39.4784;
const float uHmult = 1e6, mHmult = 1e3;
const float pFmult = 1e12, nFmult = 1e9, uFmult = 1e6;
const float C2val  = 1.000e-9;      // 1 nF reference
const float RREF   = 10000.0;       // 10 kΩ reference

float C1val, L1val, CF = 1.0;
float F1sqrd, F2sqrd;
long  Fcount, Freq3;  float F3sqrd;

/* ── 1-s polling frequency counter ─────────────────────────── */
void GetFrequency() {
  uint32_t cnt = 0; uint8_t last = PIND & _BV(PD5);
  uint32_t t0 = micros();
  while (micros() - t0 < 1000000UL) {
    uint8_t now = PIND & _BV(PD5);
    if (now && !last) cnt++;
    last = now;
  }
  Fcount = cnt;
}

/* ── Measure resistance with correct formula --------------- */
float MeasureRx() {
  uint32_t sum = 0;
  for (uint8_t i = 0; i < 32; i++) sum += analogRead(SensePin);
  float Vout = (sum / 32.0f) * (5.0f / 1023.0f);
  if (Vout < 0.01f || Vout > 4.99f) return -1.0f;      // open/short
  return RREF * (Vout / (5.0f - Vout));                // ohms
}

/* ── Talkie helpers (digits + units) ---------------------- */
void speakDigit(uint8_t d){
  static const uint16_t w[]={
    sp2_ZERO,sp2_ONE,sp2_TWO,sp2_THREE,sp2_FOUR,
    sp2_FIVE,sp2_SIX,sp2_SEVEN,sp2_EIGHT,sp2_NINE};
  voice.say(w[d]);
}
```

```
61
62  void speakNumber(float v,uint8_t dec=2){
63    int whole = int(v);
64    if (!whole) voice.say(sp2_ZERO);
65    else {
66      char buf[10]; itoa(whole, buf, 10);
67      for(char *p = buf; *p; p++) speakDigit(*p - '0');
68    }
69    if (dec) {
70      voice.say(sp2_POINT);
71      int frac = int(round((v - whole) * pow(10, dec)));
72      char buf2[8]; itoa(frac, buf2, 10);
73      for(char *p = buf2; *p; p++) speakDigit(*p - '0');
74    }
75  }
76
77  void speakCap(float C){
78    if (C < 1e-9) {                          // pF
79      speakNumber(C * pFmult, 2); voice.say(sp2_PICO);
80    }
81    else if (C < 1e-6) {                     // nF → announce as pF ×1000
82      speakNumber(C * nFmult * 1e3, 2); voice.say(sp2_PICO);
83    }
84    else {                                   // µF
85      speakNumber(C * uFmult, 2); voice.say(sp2_MICRO);
86    }
87    voice.say(sp2_FARAD);
88  }
89
90  void speakInd(float L){
91    if (L < 1e-3) { speakNumber(L * uHmult, 2); voice.say(sp2_MICRO); }
92    else if (L < 0.15) { speakNumber(L * mHmult, 2); voice.say(sp2_MILLI); }
93    else { voice.say(sp2_OVER); voice.say(sp2_RANGE); return; }
94    voice.say(sp2_HENRY);
95  }
96
97  /* ------------- Resistance speech  -------------- */
98  void speakRes(float R){
99    if (R < 1e6) {                            // say everything <1 MΩ in ohms
100     speakNumber(R, 0); voice.say(sp2_OHMS);
101   } else {
102     speakNumber(R / 1e6, 2); voice.say(sp2_MEGA); voice.say(sp2_OHMS);
103   }
104 }
105
106 /* ── setup ------------------------------------------------ */
107 void setup() {
108   pinMode(CLbarPin,INPUT_PULLUP); pinMode(RmodePin,INPUT_PULLUP);
109   pinMode(RelayPin,OUTPUT);       pinMode(DecrPin,INPUT_PULLUP);
110   pinMode(OscSigPin,INPUT);       pinMode(IncrPin,INPUT_PULLUP);
111   pinMode(CalLkPin,INPUT_PULLUP); pinMode(SensePin,INPUT);
112
113   lcd.init(); lcd.backlight();
114   lcd.setCursor(2,0); lcd.print("Talking Meter");
115   delay(1200); lcd.clear();
116
117   /* auto-calibration (S1 in C position) */
118   digitalWrite(RelayPin,LOW);  GetFrequency(); long F1=Fcount;
119   digitalWrite(RelayPin,HIGH); delay(20);      GetFrequency(); long F2=Fcount;
120   digitalWrite(RelayPin,LOW);
```

```
121
122    F1sqrd = sq(float(F1)); F2sqrd = sq(float(F2));
123    C1val  = C2val * (F2sqrd / (F1sqrd - F2sqrd));
124    L1val  = 1.0 / (FourPiSqrd * F1sqrd * C1val);
125  }
126
127  /* ── main loop ─────────────────────────────────────────── */
128  void loop() {
129
130    bool modeR = !digitalRead(RmodePin);   // D8 grounded?
131    bool modeC =  digitalRead(CLbarPin);   // S1 in C?
132    bool modeL = !modeC && !modeR;
133
134    lcd.clear();
135
136    /* ---------- RESISTANCE (top priority) ----------------- */
137    if (modeR) {
138      float Rx = MeasureRx();
139      if (Rx < 0) { lcd.print("Open/Short"); voice.say(sp2_OVER);
voice.say(sp2_RANGE); }
140      else {
141        if (Rx < 1e6) {
142          lcd.print("Rx = "); lcd.print(Rx, 0); lcd.print(" Ohm");
143        } else {
144          lcd.print("Rx = "); lcd.print(Rx / 1e6, 3); lcd.print(" MOhm");
145        }
146        speakRes(Rx);
147      }
148      delay(1500);
149      return;
150    }
151
152    /* ---------- C & L share frequency ---------------------- */
153    GetFrequency();
154    if (Fcount == 0) { lcd.print("Over Range"); voice.say(sp2_OVER);
voice.say(sp2_RANGE); delay(1500); return; }
155    Freq3  = Fcount; F3sqrd = sq(float(Freq3));
156
157    /* ----------------- CAPACITANCE ------------------------- */
158    if (modeC) {
159      float Cx = C1val * CF * (F1sqrd / F3sqrd - 1.0);
160      if      (Cx < 1e-9) { lcd.print("Cx = "); lcd.print(Cx * pFmult, 3); lcd.print("
pF"); }
161      else if (Cx < 1e-6) { lcd.print("Cx = "); lcd.print(Cx * nFmult, 3); lcd.print("
nF"); }
162      else                { lcd.print("Cx = "); lcd.print(Cx * uFmult, 3); lcd.print("
uF"); }
163      speakCap(Cx);
164    }
165    /* ----------------- INDUCTANCE -------------------------- */
166    else { // modeL
167      float Lx = L1val * CF * (F1sqrd / F3sqrd - 1.0);
168      if      (Lx < 1e-3) { lcd.print("Lx = "); lcd.print(Lx * uHmult, 3); lcd.print("
uH"); }
169      else if (Lx < 0.15) { lcd.print("Lx = "); lcd.print(Lx * mHmult, 3); lcd.print("
mH"); }
170      else { lcd.print("Over Range"); voice.say(sp2_OVER); voice.say(sp2_RANGE);
delay(1500); return; }
171      speakInd(Lx);
172    }
```

```
173
174   /* bottom line shows frequency */
175   lcd.setCursor(0,1); lcd.print("F = "); lcd.print(Freq3); lcd.print(" Hz");
176
177   /* optional CF nudging */
178   if (!digitalRead(CalLkPin)) {
179     if (!digitalRead(IncrPin)) { CF *= 1.005; EEPROM.put(0, CF); }
180     if (!digitalRead(DecrPin)) { CF *= 0.995; EEPROM.put(0, CF); }
181   }
182   delay(1500);
183 }
184
185 /* end of file */
186
187
```