Application Note   AN2207

Lock-in Milliohmmeter

Author: Oleksandr Karpin
Associated Project: Yes
Associated Part Family: CY8C24xxxA, CY8C27xxx
PSoC Designer Version: 4.1 SP1
Associated Application Notes: AN2028, AN2044, AN2120, AN2148, AN2158

Abstract

This Application Note describes a lock-in milliohmmeter. A milliohmmeter is a handy tool for tasks such as measuring a switch's contact resistance and checking the trace resistance on a PCB. Lock-in modulation technology, which is used in this design, reduces noise and allows users to measure resistance with a sufficiently low test current.

Introduction

A milliohmmeter is used for measuring very small resistances. Examples include trace resistance on a PCB, shorted traces, and contact resistance of a switch or connector. The main issues for the milliohmmeter design are:

- o How to get stable resistance readings with a suitably low test current? For example, $50\,mA \times 1m\Omega = 50\,\mu V$. This means, that the milliohmmeter would have to be able to measure and display voltages in the range of fifty microvolts per count in a stable and repeatable manner.
- o Active semiconductor devices exhibit a low frequency phenomenon know as flicker noise, often called 1/f noise. How is flicker noise eliminated in the measurement?

Lock-in modulation (also called synchronous modulation/demodulation) provides high gain for the signal at a specific frequency and phase while reducing other signals. Other signals include 1/f noise. Therefore, the lock-in modulation technique is a good answer to issues associated with milliohmmeter design.

The milliohmmeter technical specifications are given in Table 1.

Table 1. Milliohmmeter Specifications

Item	Specification
Power Supply Voltage	5V
Power Consumption	55 mA
Modulation Frequency	5 kHz
Measured Resistance Range	0 – 10 ohm
Measurement Error	0,5%
Measurement Voltage at Probes	0,2V
Display Type	Compatible with industry standard HD44780-Based LCD Module

Measure Resistance

The technique used to measure resistance is explained in detail in Application Note AN2028 – Standard - "Ohmmeter."

A milliohmmeter must be able to measure resistance with resolution of several Milliohms. Therefore, the connection from the device to the measured resistor is calculated and introduced as an error in the result.

To cancel this error, a 4-wire connection is used. In the system shown in Figure 1, the drop in wiring resistances Rc and Rd will be negligible due to the high input impedance of the measurement system. Equation (1) shows how the measured resistance is determined:

$$R_t = R_{ref} \times \left(\frac{V_2 - V_3}{V_0 - V_1} \right) \qquad \textbf{(1)}$$

As seen in the equation, the voltage drop across Ra (V_1-V_2) and Rb (V_3-V_4) cancel out and is therefore not affected by the length and resistance of the connecting cable.

A voltage drop across the resistor device uses an instrumental amplifier. Therefore, the measured resistance is determined using Equation (2):

$$R_t = R_{ref} \times \left(\frac{V_{Rt}}{V_{Rref}} \right) \qquad \textbf{(2)}$$

- o V_{Rt} is voltage drop across the measured resistor Rt.
- o V_{Rref} is voltage drop across the reference resistor R_{ref}.

When we have to measure small voltages, the problems of offset, gain and signal-to-noise ratio come into consideration. To overcome these problems we will use:

- o Correlated double sampling (CDS).
- o Full-scale calibration.
- o Lock-in modulation technique.
- o Analog and digital filters.

Correlated Double Sampling (CDS):
Correlated double sampling eliminates offset errors present in the signal conditioning circuit and the ADC. The following steps are involved.

1. Short the input. Measure ADC output. Store as zero.
2. Connect inputs to measured resistor. Measure ADC output. Store as signal.
3. Subtract zero from signal.

This method eliminates gain errors present in the signal conditioning circuit and the ADC.

With CDS and reference calibration, the measured resistance can be calculated by the following equation:

$$R_t = R_{ref} \times \left(\frac{V_{Rt} - V_{zero}}{V_{Rref} - V_{zero}} \right) \qquad \textbf{(3)}$$

As is shown in Equation (3), any offset errors in the measurement system are removed by the subtraction of two measured voltages. This is CDS. The ratio of these two different values removes any measurement path gain error.

Reference Full-Scale Calibration:
The parameter of R_{ref} designates reference resistance for this device. For better resolution resistance, milliohms are used as the unit of measure. This leaves the measurement error to be determined by only the accuracy of R_{ref}.

Lock-in Modulation Technology:
This technique provides a signal gain at a specific frequency and phase, while reducing other signals, including noise. It involves:

- o Selecting the specific signal.
- o Modulating the specific signal.
- o Amplifying the specific signal.
- o Filtering the specific signal with a band pass filter (BPF).
- o Demodulating the specific signal.

The demodulated signal obtains a signal-to-noise ratio that is greater than one (SNR>1). This method eliminates all of the drift produced inside the lock-in circuit.

For a background on switched capacitor modulators, review Application Note AN2044 – Standard - "Signal Rectification, using Switched Capacitor Modulators."

Analog and Digital Filters:
Analog filters limit the bandwidth and reduce the effect of noise on the input signal.

A digital filter routine simply collects several ADC samples then finds the average. This is used to determine the average measurement of the ADC noise.

Milliohmmeter Block Diagram
Figure 1 depicts the milliohmmeter flowchart. The carrier generator produces the modulation signal that is applied to a sense resistor. High impedance analog multiplexers allow measurement of the sense resistor and shorting of the inputs to measure systems error. The column multiplexer allows users to automatically short the input (in program) for CDS execution.
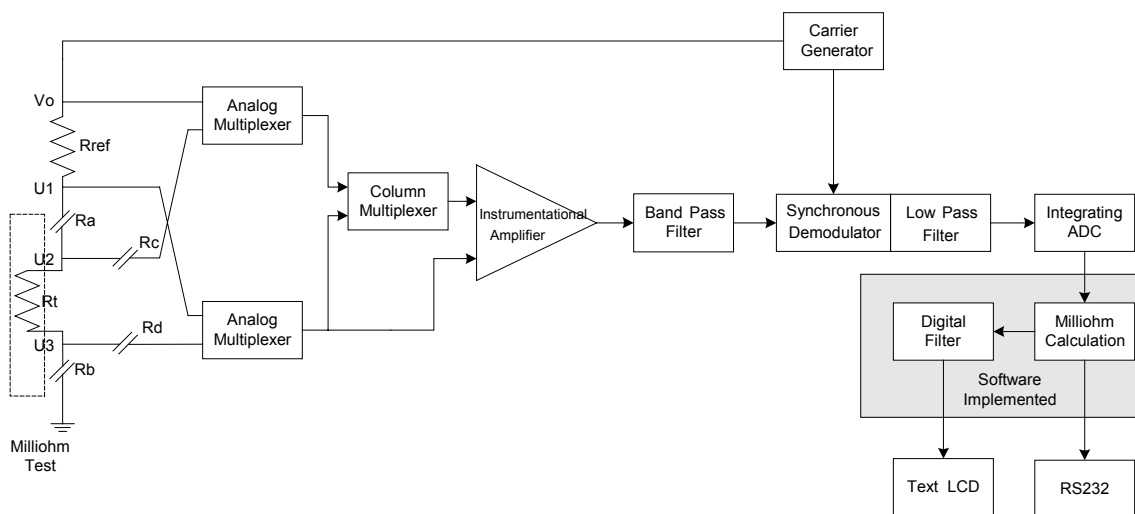
Figure 1. Milliohmmeter Flowchart

A voltage drop taken across the resistor device's instrumental amplifier is used. It amplifies the signal and makes it relative to AGND. The instrumental amplifier output is fed to a band pass filter (BPF). After passing through a band pass filter, the output is amplified and sent through a filtered signal. Then a synchronous demodulator rectifies it. The demodulator reference signal is set to the modulation signal. The demodulator output is then fed to a low pass filter (LPF). The low pass filter amplifies and filters this rectified signal to produce a filtered signal. This filtered signal is sampled directly by the integrating ADC. The integrating ADC utilized has a low pass bandwidth set by the output sample rate. The Nyquist sampling rate is at the input sample clock rate. For an n-bit converter, this is 2^n times higher than the output rate and effectively eliminates any biasing problems.

Using the PSoC™ device, we can make a demodulator based on a low pass filter, which provides more accurate output data. Note that the ADC conversion time is exactly the same as the carrier-signal period's integer number so that the ADC's gain frequency matches the carrier signal harmonics.

The following processing of ADC data stream is implemented in the firmware. The software calculates the measured resistance in milliohms as shown in Equation (3). The digital filter routine collects several measured resistances and then finds the average. The result of resistance average output is then sent to the LCD. For debugging purposes and external resistance evaluation, intermediate data can be sent via RS232 port.

Device Schematic
The complete milliohmmeter schematic is shown in Figure 2 for measurement, and Figure 3 for display and output.

R_1 is the reference resistor. As shown in figures 2 and 3, probes from J_1 connect with measured resistor Rt. The modulation frequency signal from port P2[0] goes through series-dropping resistor R_2, R_1 and the measured resistor. The R_2 is destined to reduce current and voltage of measurements. Voltages from the measured and reference resistor R_1 take the readings to the ports P0[0], P0[1], P0[6], P0[7] with high input impedance. They come through high pass filters based on C_8R_4, C_7R_5, C_5R_6, and C_2R_7 with reference to AGND. The $R_{10}R_{11}C_{14}$ determines the analog ground potential. To provide a low voltage at probes, diode D4 is used. Diodes D4 and D5 protect the device from the outside high voltage.

Speaker U4 makes a voice signal to indicate that the device measurement is in range. The milliohmmeter used is compatible with the industry standard HD44780-Based LCD Module, which is a 16-character by 2-line display driver, U3. The optional RS232 Level Translator U2 with auto-shutdown can be used for sending the ADC raw data or measured resistance values to the external PC. The test points TP1 and TP2 are intended for observing some PSoC device internal signals. TP1 is output of the band pass filter. TP2 is output of the low pass filter. The milliohmmeter can be powered from nonstabilized 6-12V DC/AC supply with maximum current of 100 mA. Normal operational current is significantly less.
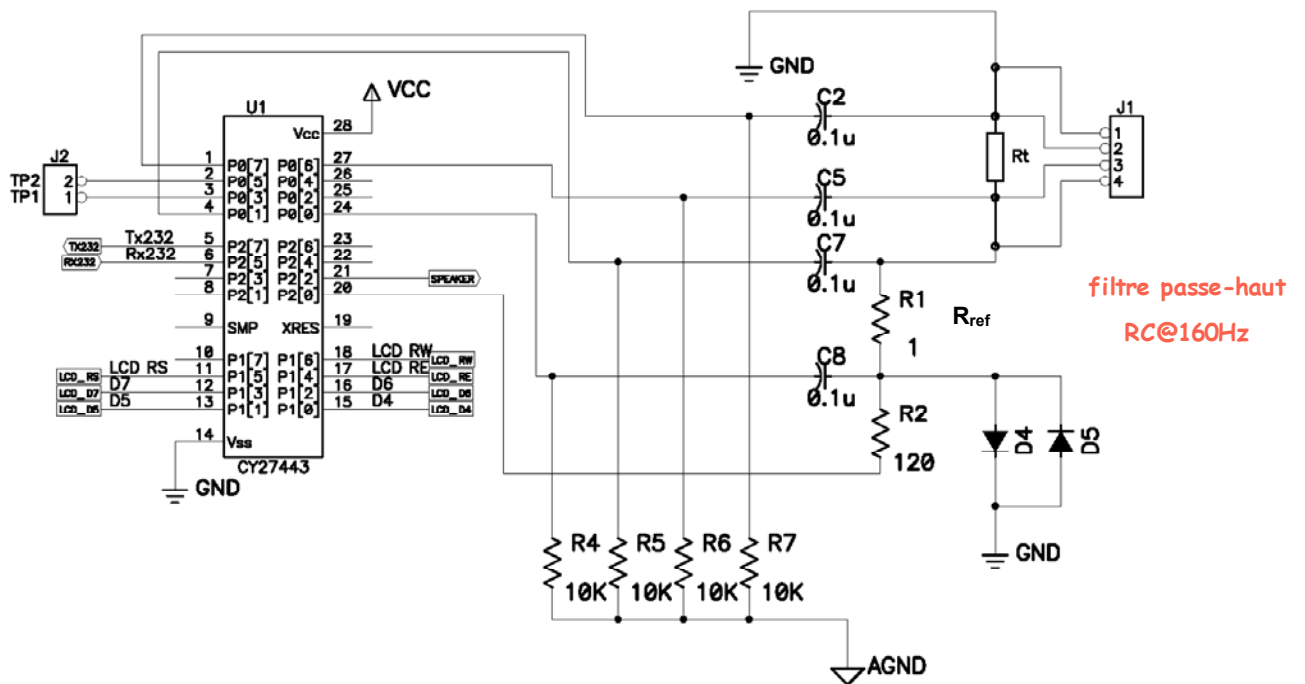
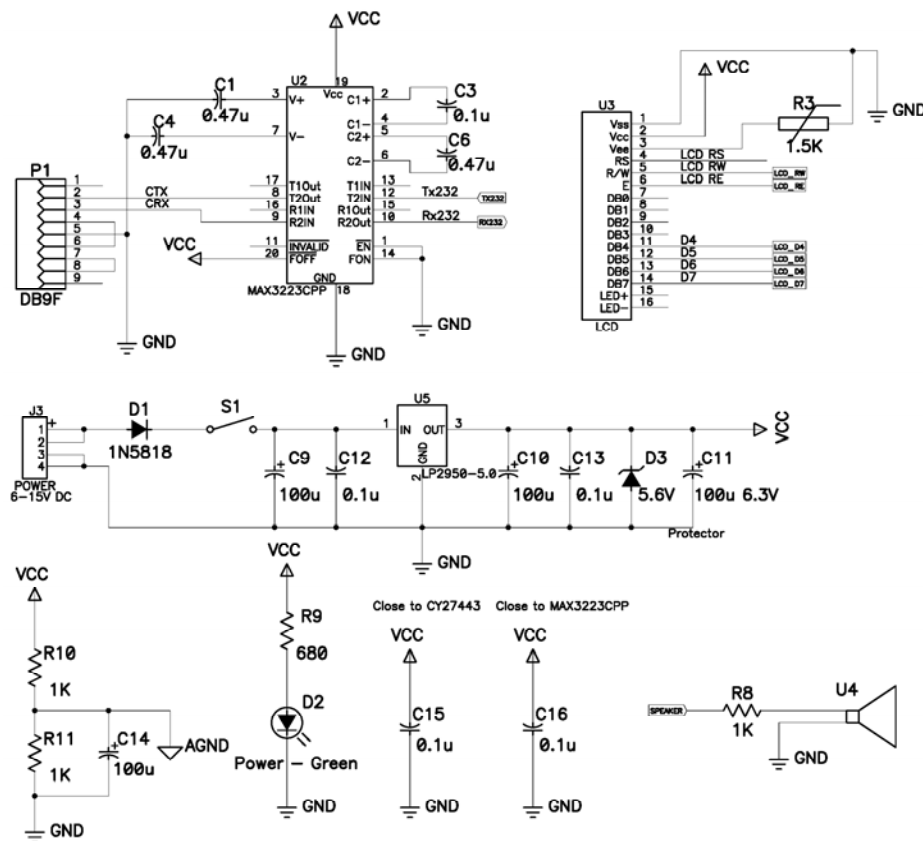Figure 2. Milliohmmeter Schematic: CPU and Measuring Equipment

Figure 3. Milliohmmeter Schematic: LCD, RS232 and Analog Components

PSoC Internals

The PSoC internal structure is shown in Figure 4. The instrumental amplifier is placed in ACB00 and ACB01. The amplifier is connected with BPF, which is placed in ASC10 and ASD20. The BPF output is sent via Port = P0[3] to checkpoint TP1 for testing and debugging purposes. The filter output signal is passed to a synchronous rectifier, which is build around LPF and is placed in ASD11 and ASC21. The rectifier reference comes from *GlobalOut_Even0*. The LPF output is sent via Port = P0[5] to checkpoint TP2 for testing and debugging purposes. Note that the column buffers BUF0 and BUF1 can be disabled to save power in the production release. In the 5V-powered milliohmmeter, the AGND level is set to Vdd/2 ± BandGap.

The LPF output is connected with ADC input. In this design, the incremental ADC is used. The ADC resolution is set to 14 bits, but the integration time is adjusted to precisely equal the integer number of the modulation signal in the firmware by directly writing to the ADC 17-bit PWM Period and Compare registers. As a result, the effective resolution is 13.6 bits. The modulation frequency is set to 5 kHz, filter over-sample ratio is 50, ADC integration time is 50 ms, or 250 modulation signal periods, and the ADC sample frequency is very close to 20 Hz. All switched capacitor modules use the same column frequency to prevent errors.
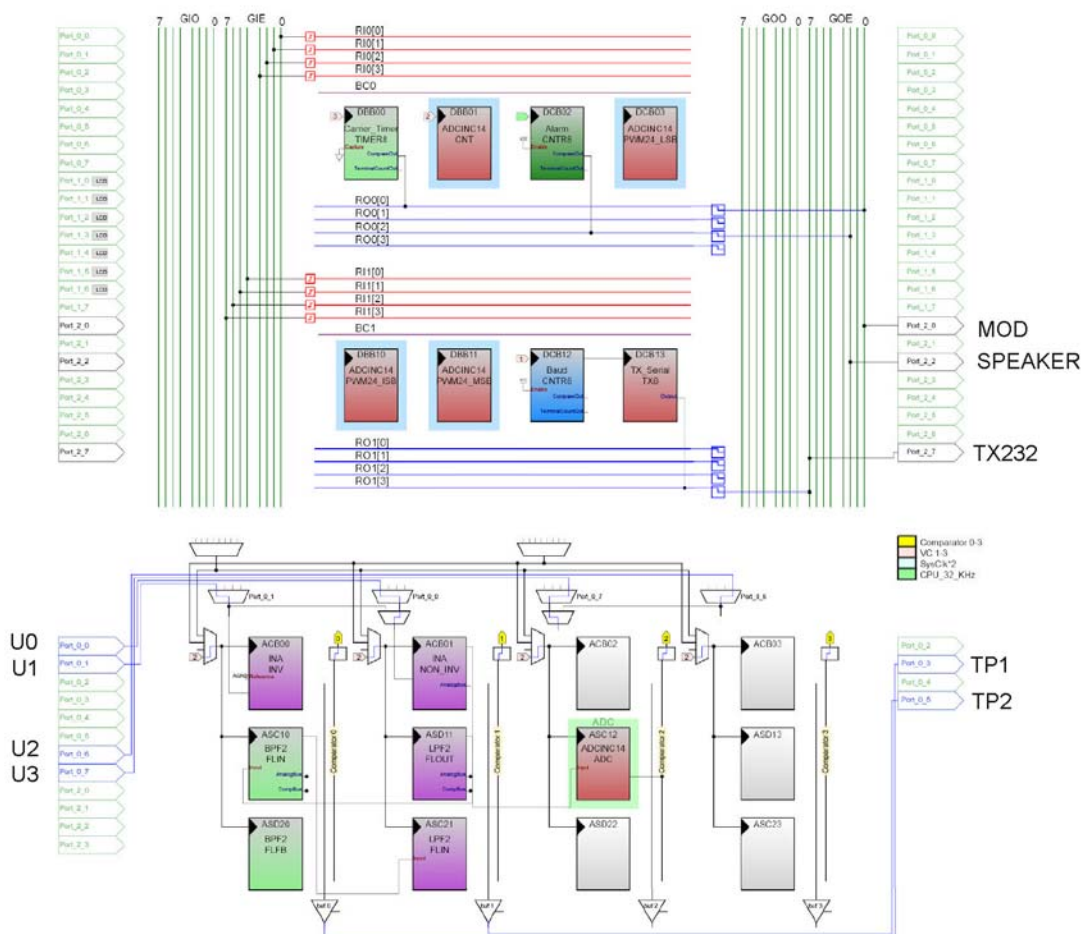
Figure 4. PSoC Internal User Module Configuration

The serial transmitter (placed in DCB13), along with the baud-rate timer (placed in DCB12), is used for transmitting debug ADC sample stream or other information to the PC. Both can be omitted in the production release. The timer (placed in DBB00) forms the modulation signal; the timer clock source is VC3. The LCD interface is implemented in software. The alarm counter (placed in DCB02) is used for signal-to-speaker generation. Note that the project can be used with the lower-cost CY8C24xxxA PSoC device without any problems. The serial transmitter and alarm counter modules should be eliminated, because the debug functionality is not required in the production release.

A synchronous rectifier should be built around the configurable switched capacitor block. The project must use an incremental ADC with a resolution of 13 bits. This technique is explained in detail in Application Note AN2158 - Standard - "Optical PulsOmeter with PSoC."

The Software
The milliohmmeter software consists mainly of three components:

- o readAdc
- o processAdc
- o main loop

The logic of the program is explained by program flowcharts in figures 5, 6 and 7.

Start

getADCSamples(1)

Is Data Ready?

No

Yes

Return ADCData
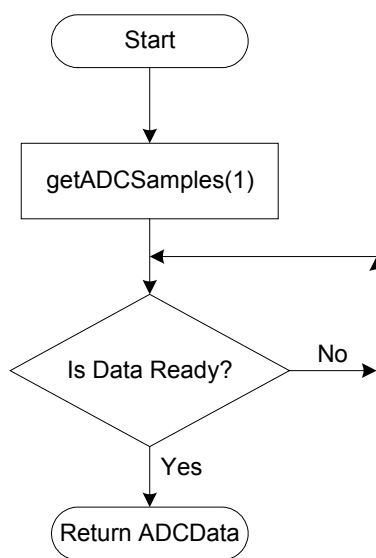
Figure 5. Program Flowchart for readAdc

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │ AMuxPos_In(PORT0_0)          │
            │ AMuxNeg_In(PORT0_1)          │
            │       Vref=readAdc           │
            │   Flip col1_mux on col0      │
            │       Vzero=readAdc          │
            │       Vref=Vref-Vzero        │
            │   Flip col1_mux on col1      │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │ AMuxPos_In(PORT0_6)          │
            │ AMuxNeg_In(PORT0_7)          │
            │        Vt=readAdc            │
            │   Flip col1_mux on col0      │
            │       Vzero=readAdc          │
            │        Vt=Vt-Vzero           │
            │   Flip col1_mux on col1      │
            └──────────────────────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │      lRt=Vt*1000/Vref        │
            │        Rt=(int)lRt           │
            └──────────────────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │  Return Rt  │
                    └─────────────┘
```
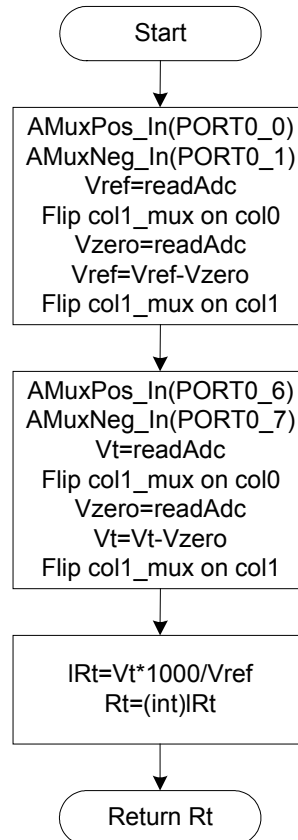
Figure 6. Program Flowchart for processAdc

```
                    ┌──────────┐
                    │  Start   │
                    └──────────┘
                         │
              ┌──────────────────────┐
              │      count=0;        │
              │       pos=0;         │
              │   Rt_Sample = 10;    │
              │    MAX_Rref=955      │
              └──────────────────────┘
                         │
              ┌──────────────────────┐
              │ iaRt[pos]=processAdc │
              └──────────────────────┘
                         │
         Yes      ◇ pos==Rt_Sample-1 ◇      No
       ┌─────────────────────────────────────┐
       │                                     │
  ┌─────────┐                           ┌─────────┐
  │  pos=0  │                           │  pos++  │
  └─────────┘                           └─────────┘

              ◇ count<Rt_Sample ◇   No

                    │ Yes
              ┌─────────────┐
              │   count++    │
              └─────────────┘

              ┌─────────────┐
              │   lRt=0      │
              │    i=0       │
              └─────────────┘

              ┌─────────────────┐
              │ lRt=lRt+iaRt[i] │
              │      i++        │
              └─────────────────┘

    Yes    ◇ i<Rt_Sample-1 ◇
                    │ No
              ┌─────────────┐
              │ lRt=lRt/count│
              └─────────────┘

     Yes    ◇ lRt<=MAX_Rref ◇    No
  ┌──────────────┐        ┌──────────────────┐
  │ LCD_out(lRt) │        │ LCD_out("Error") │
  └──────────────┘        └──────────────────┘
```

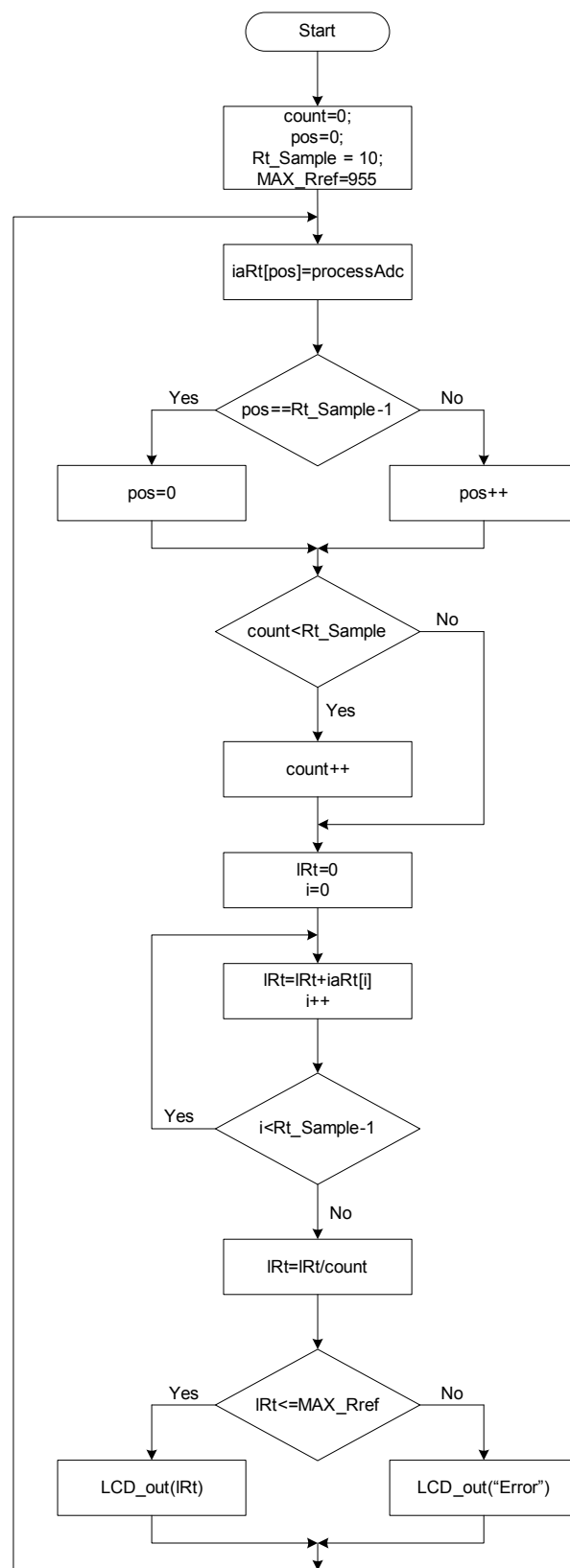**Figure 7. Program Flowchart for main loop**

Application Domains

Using two additional buttons, we can set the resistance range of dynamic speaker actuation. After this, we can easily check the trace resistance on a PCB.

The milliohmmeter can be used for measuring shorted traces as shown in Figure 8.
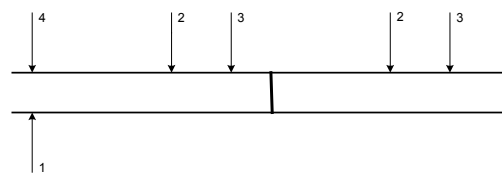

Figure 8. Tracking Shorted Traces
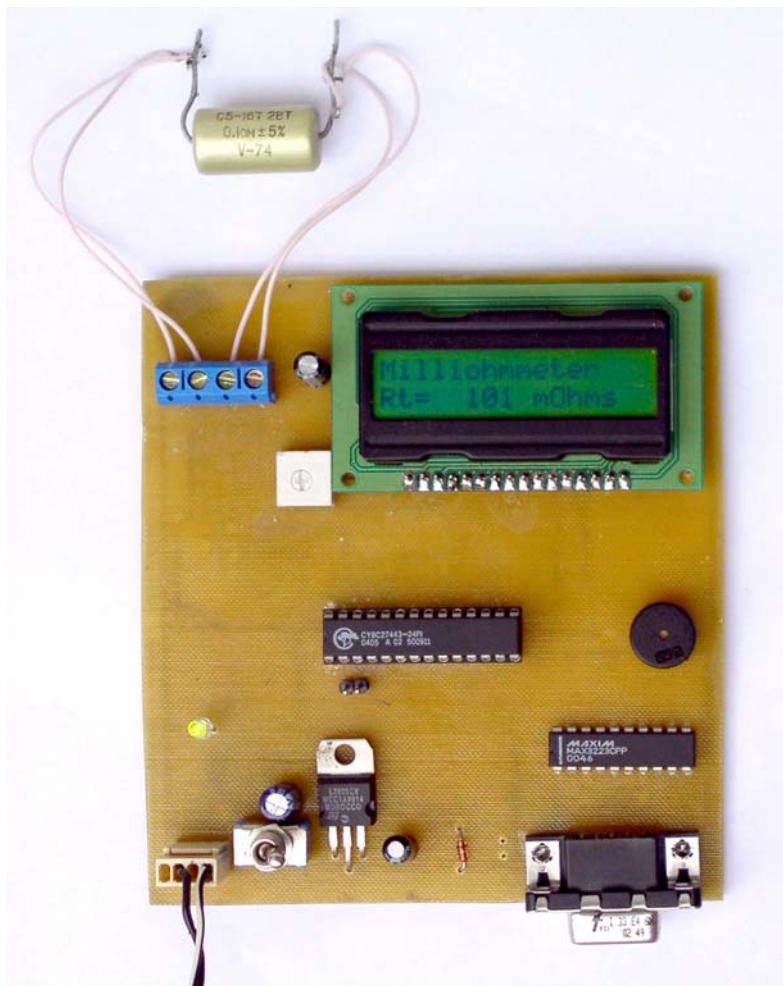
Device Photograph

Figure 9. Board Photograph

About the Author

Name:	Oleksandr Karpin
Title:	Post-Graduate Student
Background:	Oleksandr earned his computer-engineering diploma in 2001 from National University "Lvivska Polytechnika" (Lviv, Ukraine), then studied as a post-graduate student at the same university. His interests include embedded systems design and new technologies.
Contact:	karpinoo@polynet.lviv.ua