

# Firmware-based Capacitor and Equivalent Series Resistance (ESR) Meter

John Davis, Lucky A. Ishaku, Abdelrahman Abdelazim, Margarita Georgieva  
Department of Advanced Engineering  
Blackpool and The Fylde College, United Kingdom, FY2 0HB  
E-mail: lucky.ishaku@blackpool.ac.uk; luckyishaku@yahoo.co.uk

## Abstract

This paper presents the development of a capacitor ESR meter, which covers capacitance and ESR measurements as well as the interrelationship of the constituent parts/circuits. The idea is to keep the hardware design simple, transferring the complexity to firmware which controls the hardware. In particular, it uses automatic functions controlled by relays thus there is no need for switches and its automatic calibration ensures there is no need for menus. This work covers voltmeter function, C/R time constant function and ESR function using Arduino Uno boards. Each firmware code was developed one after the other, starting with the voltmeter function then C/R time constant function and finally ESR function. Results were checked by serial data transfer from Arduino Uno board to the laptop that is running the Arduino compiler with terminal. One Arduino Nano interface was constructed to interface the project shield. The firmware was developed to provide the voltmeter function, C/R time constant function and ESR function. An organic light-emitting diode (OLED) display was used to show measurement results. Logic control was added to provide the required functions to work as intended, as well as an automatic short circuit detection at DUT terminals firmware which includes a calibration cycle. Conventional ESR meters are usually have complex hardware, in contrast the design presented in this work shifts the focus to the firmware so that hardware is uncomplicated. This reduces the cost that typically comes with more hardware components.

**Keywords:** Capacitor, ESR Meter, C-R Time Constant, Arduino, OLED Display, Calibration

**DOI:** 10.7176/CEIS/13-1-02

**Publication date:** January 31<sup>st</sup> 2022

## 1. Introduction

This measurement device uses simple circuit design to measure capacitors with a Capacitor and ESR Meter with high accuracy. Controlled by a microcontroller with a display to show capacitance in farads and ESR in ohms. The measurement to determine capacitance uses the CR time constant method. The measurement to determine the ESR of a capacitor is the pulsed method. To get the required measurement resolution the ADC is oversampled to get an increase in resolution. Software control is fully automatic to determine capacitance, ESR and calibration. It is important to state that most capacitor values tested are between 1uF and 470uF.

### 1.1 Methods of Measuring a Capacitor's Value

#### 1.1.1 Capacitor/Resistor Time Constant

A resistor-capacitor connection in series across a voltage supply is the easiest circuit to implement and measure the time taken to charge a capacitor to 63.2% of charge. The equation for it is given by:

$$\tau = RC \quad 1$$

Where RC is the Time Constant. If resistance R is fixed, then the value of capacitance is proportional to time constant in seconds. If R = 1000 ohms and Capacitance is 1uF then time constant would be 1mS. If R = 1000 ohms and Capacitance is 1000uF then time would be 1s. So, the down side is that the display update time is about one second. This circuit typology was chosen because it is easy to implement.

#### 1.1.2 Frequency Resonance.

The resonance frequency method being parallel LC tank circuit. Were the capacitive and inductive reactance is equal and 180 degrees opposite in phase. The formula for frequency resonance is given as,

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad 2$$

The circuit typology that provides the oscillation does not matter. It is the relationship with frequency and capacitance when the inductance is a fixed value that matters. Frequency is inversely proportional to the capacitance with a fixed inductance value. This means the frequency is a low value when capacitance is a high value. The circuit typology that uses this method usually is ok when capacitance has a value from 10pF to 1uF. The added circuit complexity and that the capacitance range is not good enough for the proposed project.

### 1.2 Methods of Measuring Capacitor's ESR.

#### 1.2.1 100 kHz Impulse.

There is no direct formula for this method but an ADC measurement and Ohm's law calculation to define the ESR

of a capacitor. The capacitor is subjected to a series of 100 kHz pulses through a constant current circuit as shown in Figure 1. That impresses a small dc voltage (millivolts) that is captured by the ADC from which the resistance can be calculated from a known current. This resistance is in the range of 0.03 ohms to 10 ohms. Because of the low value captured the ADC is oversampled to give a 16bit value from a 10bit ADC. Because this method is easy to implement it was chosen for the project.



Figure 1: ESR 100 kHz Impulses [1].

### 1.2.2 100 kHz Sinewave.

Auto balancing bridge (Figure 2) is one way to measure ESR as well as other parameters. Like resistance, capacitance and inductance using a sinewave for the source. DUT is measured by voltmeter v1 and v2 to determine the impedance Z.

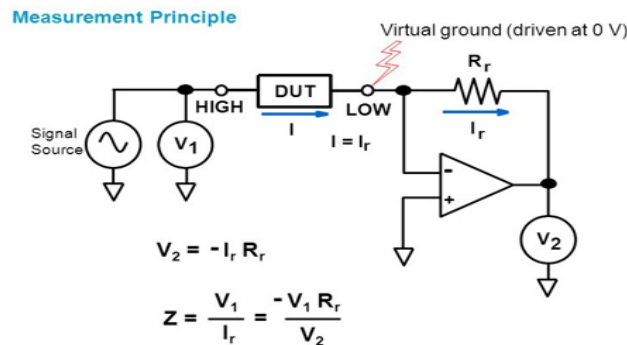


Figure 2: Auto Balancing Bridge [2]

From the vector impedance diagram (Figure 3) you can determine ESR of a capacitor. This method was not used as it is a complex typology.

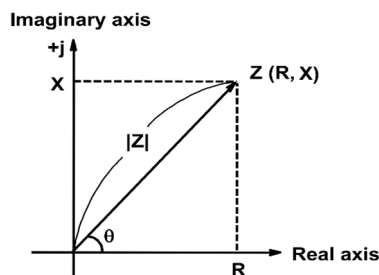


Figure 3: Vector Impedance Diagram [3]

$$Z = R + jX \text{ (Rectangular-coordinate)} \quad Z = |Z| \angle \theta \text{ (Polar form)}$$

$$R = |Z| \cos \theta \text{ (ESR)} \quad X = |Z| \sin \theta \quad 3$$

$$|Z| = \sqrt{R^2 + X^2} \quad \theta = \tan^{-1} \left( \frac{X}{R} \right)$$

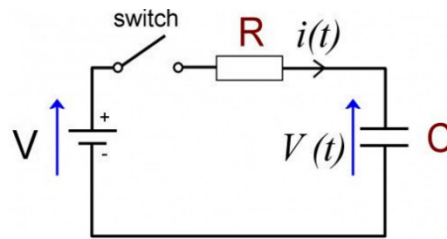
## Measurement Theory

### 1.3 Capacitor/Resistor Time Constant

As the voltage V is applied across a series RC circuit (Figure 4), Voltage across the capacitor  $V_o$  charges at an exponential rate. Thus the capacitance is defined as the ratio of charge stored to the voltage developed across it. This gives:

$$q(t) = Cv(t) \quad 4$$

Where: q = is charge in coulombs, v = Voltage in Volts and C = Capacitance in Farads.



**Figure 4: RC Series Circuit**

If we differentiate each side of the equation we obtain.

$$\frac{dq(t)}{dt} = \frac{Cdv(t)}{dt} \quad 5$$

From the series RC circuit it can be seen that the current is the rate of flow of charge,  $dq(t)/dt$  is the current flowing into the capacitor, therefore:

$$i(t) = \frac{Cdv(t)}{dt} \quad 6$$

If we assume that there is no current being drawn from the capacitor, then the current through the resistor is equal to the current through the capacitor. By ohm's law the voltage across the resistor is equal to  $(V-v(t))$ . By Kirchoff's law we obtain,

$$V - v(t) = Ri(t) \quad 7$$

From equations (6) and (7) we obtain,

$$\frac{V-v(t)}{R} = \frac{Cdv(t)}{dt} \quad 8$$

Separating the variables gives,

$$\frac{d(t)}{RC} = \frac{dv(t)}{V-v(t)} \quad 9$$

Solving by integrating leads to,

$$\frac{t}{RC} = -\ln(V - v(t)) + K \quad 10$$

Since at  $t = 0$ ,  $v(t) = 0$ , then by substitution K becomes,

$$K = \ln(V) \quad 11$$

Substituting equation (11) into equation (10) gives,

$$\frac{t}{RC} = -\ln(V - v(t)) + \ln(V) \quad 12$$

Rearranging equation (12) gives,

$$\ln\left(\frac{V-v(t)}{V}\right) = -\frac{t}{RC} \quad 13$$

In exponential form it becomes,

$$V - v(t) = Ve^{-\frac{t}{RC}} \quad 14$$

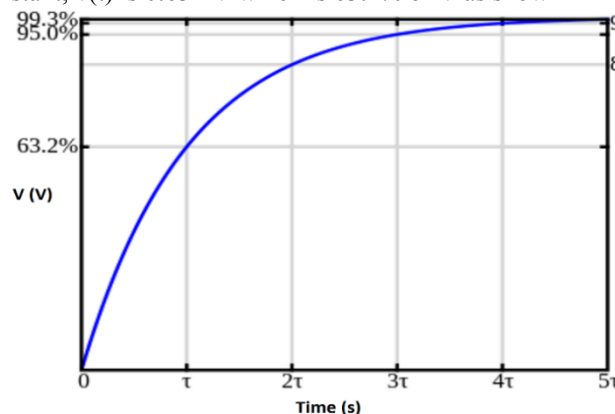
This resolves to,

$$v(t) = V(1 - e^{-\frac{t}{RC}}) \quad 15$$

For one time constant  $t = 1$  and  $RC = 1$  or  $t = RC$

So,  $v(t) = V(1 - e^{-1}) = V(0.632)$  16

Therefore at one time constant,  $v(t)$  is 0.632 V which is 63.2% of V as shown in Figure 5



**Figure 5: Time Constant Graph**

The product  $R \times C$  is used to find values of V at different times.

### 1.4 Equivalent Series Resistance (ESR)

ESR or Equivalent Series Resistance is the normally referred to as the loss angle  $\delta$  (Greek letter delta). The model of which consists of a perfect capacitor in series with a resistor (Figure 6). The resistance comprises the loss of the capacitor which manifests as heating of the capacitor. This usually happens to electrolytic capacitors more so than other types of capacitor like ceramic. Switch mode power supplies are prime example of capacitors ESR due to their long use Also due to currents charging and discharging capacitors within the switch mode power supplies. Also poor quality capacitors can cause poor performance of switched mode power supplies.

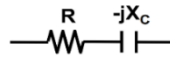


Figure 6: ESR Model

The vector diagram figure 7 showing R and  $-jX_c$  are at right angles and would indicate a perfect capacitor. But since no capacitor is perfect has loss, which is the loss angle  $\delta$ .

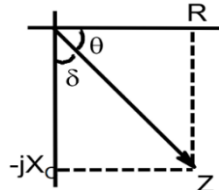


Figure 7: Capacitor Loss Tangent

The loss angle is the ratio of ESR and the capacitive reactance  $X_c$ .

$$DF = \tan \delta = \frac{ESR}{|X_c|} \quad 17$$

Where:  $\delta$  = Loss Angle, DF = Dissipation Factor, ESR = Equivalent Series Resistance,  $X_c$  = Reactance of the capacitor in Ohms. The quality factor is the energy losses in a resonant system. It can also be referred to as the ratio of energy stored to that dissipated per cycle. Thus:

$$Q = \frac{X_c}{ESR} \quad 18$$

The dissipation factor DF is the tangent of the loss angle  $\delta$ . The lower this value is the lower the loss in the capacitor.

## 2. Hardware Planning

### 2.1 Arduino Nano

#### 2.1.1 Voltmeter Function

A potential divider is an electric circuit with two resistors in series. The supply potential  $V_{in}$  is across both resistors in which a common current flows through both resistors. Therefore, the potential  $V_{out}$  is a divisor of  $V_{in}$  set by the ratio of the two resistors  $R_1/R_2$  as in Figure 8. This is the same circuit in the Arduino Nano Based Project  $R_1/R_2$ . The resistors  $R_1, R_2$  are 0.1% resistors to minimise the overall measurement error budget.

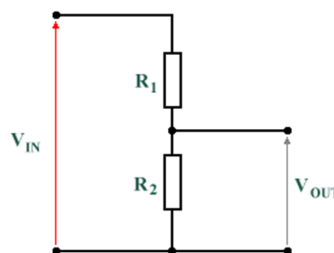


Figure 8: Potential Divide.

The formula to work out  $V_{out}$  when  $V_{in}, R_1$  and  $R_2$  is given as Figure 9.

$$V_{out} = \frac{V_{in} \times R_2}{R_1 + R_2} \quad 19$$

This potential divider is used to measure the supply voltage by analogue read (A0). The reason for this is the supply voltage changes under load. For example, if supply voltage was 5.12V there would be an error of  $5.12V - 5V = 0.12V$ . The Arduino ADC is 10 bits which equals  $2^{10} = 1024$  steps. The step voltage is  $5V/1024 = 4.88mV$ . If the supply voltage is 5.12V then the step voltage would be  $5.12V/1024 = 5mV$  which is an error of  $120\mu V$  per step which gives a total error of 3% which is greater than the measurement goal of 1%. The software in other designs do not follow this approach but enter the measured supply voltage as value in the firmware code.

#### 2.1.2 Capacitor / Resistor Time Constant Function

This circuit is just two resistors  $R_3, R_4$  to control the charge and discharge of a capacitor. The control of the resistors set by the port pins 5, 6 (Figure 12). The resistor  $R_3$  1k 0.1% is used to control the charge of a capacitor by the time constant and monitored by an analogue read(A1) to 63.2% of charge.  $R_4$  controls the discharge after each measurement. The reasoning of this circuit is that it is simple but care is needed in the discharge current to

less than 20mA per port pin. Other circuits may use a transistor to control the discharge of a capacitor which is an additional cost (Figure 12).

### 2.1.3 ESR Function

This circuit comprises of a constant current source Q1, R6 and pulsed controlled from port pin 8 with diode clamps to limit the pulsed voltage to +/-0.7V. The constant current source does not have an issue that it is not completely a constant current source because the short circuit current and full load current are not the same. There is slope error that can be calibrated for in firmware. To implement a true constant current source would have been more circuit complexity and addition cost. This was noted in the earlier design on the Arduino Uno project board. The other components that comprise the circuit of the ESR function was functioning as intended.

### 2.1.4 OLED Display

The OLED display was chosen because of its small size and I2C control. Other designs tend to be LCD which have viewing issues when there is little lighting. The I2C is a 2 wire master/slave control system that has a clock data in/out in half duplex mode.



Figure 9: The OLED display which is chosen over LCD usually have viewing issues when there is little lighting [4]

## 2.2 The I2C Protocol

The two wire interface to connect to microcontrollers, ADC, DAC and EEPROMS. This interface is popular because you have more than one master in the system. This interface was invented by Philips and now used by most other chip manufacturers. Each I2C slave device needs an address – they must still be obtained from [NXP](#) (formerly Philips semiconductors).

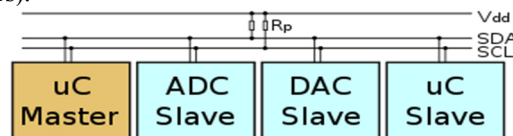


Figure 10: I2C Bus [5]

The original specification for transmission/receive speeds was 100 kHz. But because newer devices are capable of higher speeds later increased to 400 KHz fast mode. There is also high speed mode 3.4MHz as well as ultra-speed mode 5MHz. Each slave device has a unique address and transfer of 8 bit data packets is serial with a synchronous clock.

### 2.2.1 I2C Interface

I2C uses only two wires: SCL (serial clock) and SDA (serial data). Both need to be pulled up with a resistor to supply rail.

### 2.2.2 I2C Addresses

Each slave device has a unique 7bit address on the bus. Data transfer is 8bit / bytes to give basic communication. Devices can have a fixed I2C address others can have additional addresses to determine the lower address. Therefore very easy to have all I2C devices on the bus with a unique address. Master device does not need an address because it generates the clock. Also addresses all the slave devices on the bus. The initial state of the lines SCL and SDA are high. Communication starts by the master device generates the start condition (S). Followed by the address of the slave device (B1). If the address byte bit 0 is set to 0 the master device will write to slave (B2). Otherwise, the next byte will be read from the slave device. When all bytes have been read or written (Bn) the master device generates a stop condition (P). This allows other devices to use the bus because communication has ended (I2C Protocol).

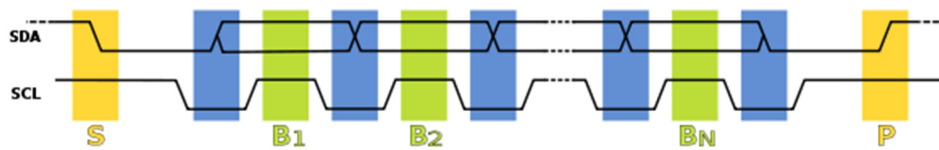


Figure 11: I2C Timing Diagram [3]

### 3. Firmware Planning

Before firmware can be planned the I/O is requirement had to be planned and resolved first. From the circuit diagram (Figure 12) it can be seen that the following I/O is required.

Arduino	Uno	Nano	I/O
Analogue Port	ADC(0)	ADC(0)	Input
Analogue Port	ADC(1)	ADC(1)	Input
Analog Port	ADC(2)	ADC(2)	Input
Digital I/O Pin	2	2	Output
Digital I/O Pin	3	3	Output
Digital I/O Pin	4	4	Output
Digital I/O Pin	5	5	Input / Output
Digital I/O Pin	6	6	Output
Digital I/O Pin	7	7	Output
Digital I/O Pin	8	8	Output
Digital I/O Pin		SCL	Output
Digital I/O Pin		SDA	Output

Since from the circuit diagrams (Figure 12) it can be seen that there are three circuit functions Voltmeter, C/R Time Constant and ESR. These functions are common to the Arduino Uno and Arduino Nano platforms. The only difference is Arduino Nano platform has OLED display. Also the Arduino Ono platform has data sent to the UART serial port. Three Arduino Ono Boards were used to write the firmware code for each function. With data required is sent from the UART to the USB port of a Laptop to show in the terminal function within the Arduino compiler.

#### 3.1 Voltmeter Arduino Uno

The firmware was written to read a voltage on the analogue port (ADC0) and use serial print function to send characters and data values from the UART to the laptop to show data on the terminal screen. The reason for doing this is that there is no display to show the results. Also the error voltage was calculated, showing the result of the supply voltage. The reason for doing was the supply voltage can change and result in an error C/R Time Constant and ESR functions.

#### 3.2 Voltmeter C/R Time Constant Arduino Uno

The firmware was written to control a charge and discharge of a capacitor. From which calculations give the capacitor value in micro farads. The result was sent via serial UART to the laptop USB port to show the result in the Arduino compiler terminal function. This was a good practice to debugging the code and showing the results. Also short circuit detection was written so that the action was automatic. The reason for this was there would be no need for a switch to do the zero calibration.

#### 3.3 ESR Arduino Uno

The firmware was written to control 100 KHz current pulse to a capacitor. From the calculations which results in the ESR of the capacitor. The result was sent via serial UART to the laptop USB port to show the result in the Arduino compiler terminal function. The important part of the result is the oversampling of the ADC to give a 16bit result from a 12bit ADC (oversampling). This result was found to be slow about one result update per second. This can be easily change in the code to give a 14bit result without compromising the result. Thus, this would give an increase in the result update rate.

#### 3.4 Arduino Nano Final

The three functions of the Arduino Uno firmware was added together to give the final code. OLED display control added to give results on the display as Capacitor value and ESR value. Logic code added to provide control of relays for each function. Also code added to detect if a capacitor is connected or not. Automatic short circuit detected to give automatic zero calibration. Control code added to display capacitor value and ESR value if connected. Also place zero values for capacitor value and ESR value if not connected.

## 4. Circuit Description.

### 4.1 Hardware.

Since both Arduino Uno [6] and Arduino Nano [7] projects are functionally the same. Except for the Arduino Nano which has a OLED display. Even the pin definitions are the same so the circuit description will focus on the Arduino Nano project. There is basically three circuit functions voltmeter, C/R time constant and ESR.

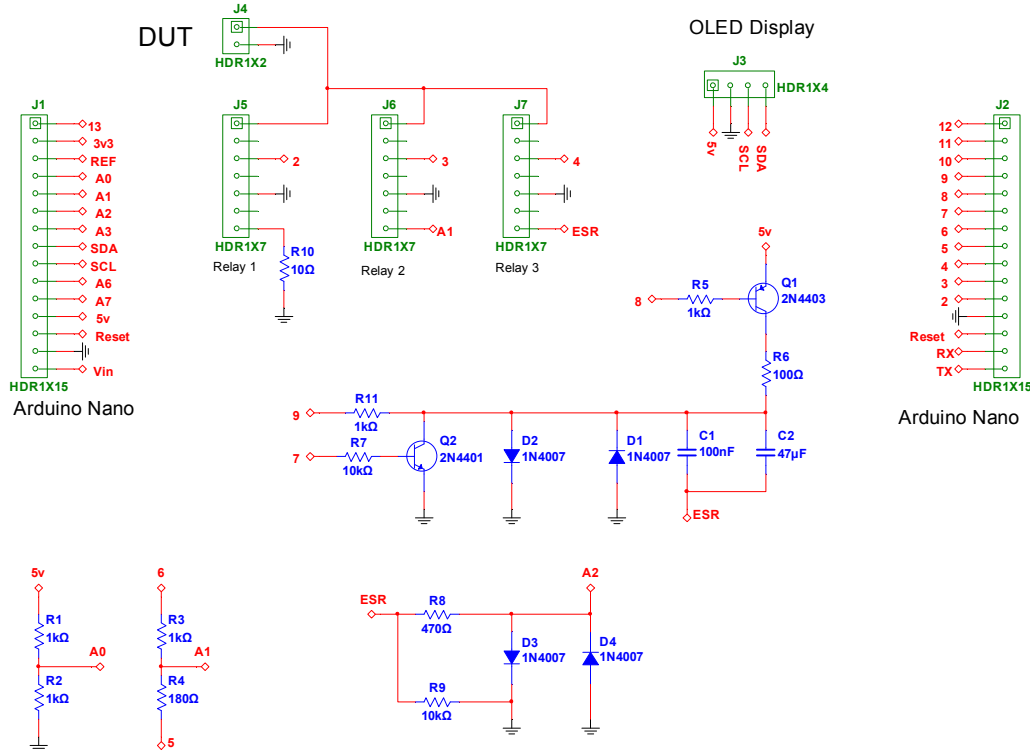
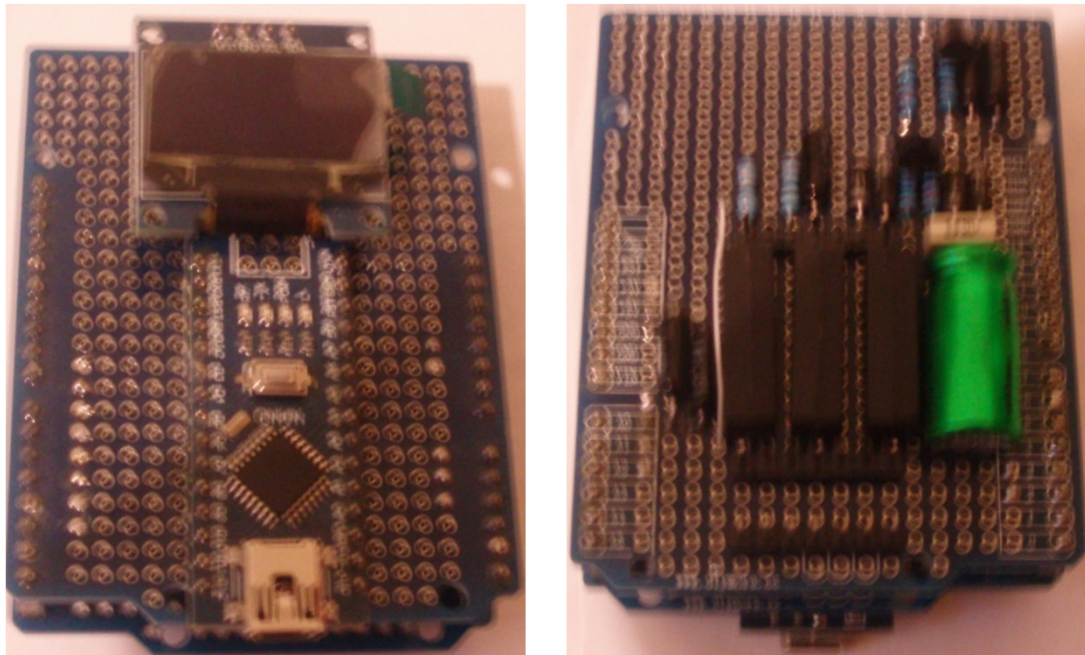


Figure 12: Circuit Diagram Arduino Nano Based Project

- 4.1.1 Voltmeter Function:** R1, R2 of equal resistance (1k) form a potential divider which is connected to the ADC(0). The supply voltage is nominally 5v so that the voltage to the ADC (0) will be 2.5V.
- 4.1.2 C/R Time Constant Function:** When relay 2 is energised by pin 3 this connects the capacitor under test (DUT) to the junction of R3, R4. When pin 6 is high current flows through the precision 0.1% resistor R3 (1k) and charges the capacitor under test. Note that R4 is not connected as pin 5 is assigned as a port input so is high impedance. This has no effect on the charging of the capacitor under test. The junction of R3, R4 is connected to the ADC(1) so the microcontroller continually reads the ADC(1) until the charge on the capacitor is 63.2%. When that charge value has been reached pin 6 is set low. R4 pin 5 is connected as an output port pin 5 which set low so the combination of R3, R4 discharges the capacitor under test. The microcontroller continually reads the ADC(1) to detected that the capacitor is fully discharged. Microcontroller re-assigns pin 5 as a port input and pin 3 is driven low to turn the relay 2 off. The calculated value is displayed on the OLED Display.
- 4.1.3 ESR function:** Microcontroller sets pin 4 high to energize the relay 3 connects the capacitor under test to one side of C1, C2. Also to junction R8, R9 is connected to ground so to discharge any charge on a capacitor under test. R8 is serially connected to ADC (2) and to a pair of parallel connected diodes D3, D4. The diodes D3, D3 function is to limit the voltage to the ADC(2) to less than +/- 0.7v which acts as a clamp. This allows for in circuit testing without the possibility to damage any semiconductor junctions. When a capacitor under test is connected it is ac coupled via the parallel connection of C1, C2. The capacitor C2 is a low value to improve bypass at higher frequencies. When the microcontroller set pin 8 is pulsed at a 100 KHz rate Q1 turns on and off at the rate. The pulse is coupled via R6 and C1, C2 to the capacitor under test. The microcontroller sets pin 8 to pulse to turn off/on Q2 to discharge the capacitor after each sample of the ADC. Also the pulse is clamped via diodes D1, D2. The pulse action at 100 KHz rate means the capacitor is not charged and discharged. But gives a DC level that corresponds to the loss of the capacitor in ohms the effective ESR. The microcontroller ADC(2) reads the DC level and calculates the ESR in ohms. The value is displayed on the OLED Display (Figure 12).



**Figure 13: Prototype Arduino Nano Top and Bottom Views**

#### **4.2 Firmware Description**

Since both Arduino Uno and Arduino Nano projects are functionally the same. Except for the Arduino Nano which has an OLED display. Even the pin definitions are the same so the firmware description will focus on the Arduino Nano project. There is basically three circuit functions voltmeter, C/R time constant and ESR. But added glue logic to drive relays, OLED display with each function voltmeter, C/R Time Constant and ESR.

Various include statements for 1-Wire, I2C Display functionary and OLED display setting height and width. Various define statements to allocate microcontroller pins 2, 3, 4, 5, 6, 7 and 9. Define resistor value 1K for C/R time constant calculation. Define float values milivolt, vref and current for ESR calculation. Define float values vout, vin, and int val all set to zero for voltmeter function. Define unsigned long values for starTime, elapsedTime, Define floats microFarads, nanoFards, ESRval and shortval all set to zero. Define int shortFlag and int NocapFlag all set to zero.

Void setup() function, run serial.begin(9600) to start UART at a baud rate of 9600 for debugging. Run display.begin(SSD1306\_SWHCAPVCC, 0x3C) function to check that I2C OLED display is connected. The following functions clears the display, set text size, set text colour (WHITE) and set cursor position. Display "Capacitor ESR Meter" on the display for two seconds by the delay function delay(2000). Set chargepin low, set pin mode to output for relays (3) and digitalwrite low to turn relays off.

void loop() start main loop function to return to. Set chargepin pin mode to output and digitalwrite chargepin low. Set relayCR, relayCR and relayESR pin mode to output and digitalwrite all three relays low so they are off. shortFlag and NocapFlag set to zero.

##### **4.2.1 Start voltmeter function**

Set analogue reference to default which is 5v. Read analogue port A0 and pass result to variable val. Do calculation  $(val * 5)/1024$  and pass result to variable vout. The analogue port is 10bit so  $2^{10} = 1024$  and the result should be 2.5V. Variable Vout is multiplied by two and is passed to variable Vin. Calculate variable Error is supply voltage by checking if variable Vin is greater than or less than 5V. If variable Vin is less than 5V then the variable Error = 5.00 minus Vin. If variable Vin is greater than 5v then the variable Error = 5.00 plus Vin.

##### **4.2.2 Start CR function**

Write to relayCR pin and to chargePin and set high. Start time function startTime = micros(). Read analogue port A(1) in a while loop waiting for a value of 647 which is 63.2% of charge. Within the while loop read analogue port A(1) to check if value less than 10. If value less than 10 then shortFlag is set to 1, followed by a break to exit the while loop. Also this condition is checking for a short circuit.

If there is a valid analogue A(1) read value of 647 then there is a capacitor connected. Then the variable elapsedTime is calculated from function micros() - startTime. From variable elapsedTime calculate the capacitance in microfarads from the variable microFarads = float elapsedTime/ resistorValue. If shortFlag is equal to zero then print message to serial port "Output Short Circuit".

If variable microFarads is greater than 1 print out serial port. Also clear display then print out to OLED display Capacitor = XXX.X  $\mu$ F, XX.X is the value of the variable microFarads. If variable microFarads is less than 1



convert microfarads to nanofarads by multiplying the variable microFarads by 1000. This pass to variable nanoFarads. Print variable nanoFarads to serial port "nanofarads XXX.XX", XXX.XX is the value of the variable nanoFarads.

If variable nanoFarads is less than 200.00 then there is no capacitor connected, the variable NocapFlag is set to 1. The variable microFarads is set to 0.00. The display is cleared then prints out "Capacitor = 0.00".

Discharge the capacitor by setting chargePin low and dischargePin pin mode to output and set dischargePin low. Read analog port A(1) in a while loop until value is zero. Set relayCR to low to turn it off and set dischargePin to INPUT this sets this pin to high impedance.

#### 4.2.3 Start ESR function

Set analogue reference to internal this is 1.1v because this improves the common mode for the ESR circuit with a maximum voltage measurement of 0.7v. The variable esrSamples receives the results from the measureESR() function. The variable milivolt is calculated from the variable  $(\text{esrSamples} * \text{vRef}) / 65535$ . The variable ESRval is calculated from the variable  $(\text{milivolt} / \text{current}) / 1000$  to give a value in miliohms. Check the variable NocapFlag is equal to zero. The print out on the display "ESR = XX.X Ohms" XX.X is the value of the variable ESRval. If shortFlag is set to 1 variable ESRval is calculated as  $\text{ESRval} - \text{shortval}$  this removes the lead resistances from the ESR value. relayESR is set to zero to turn off the relayfunction.

#### 4.2.4 End of void loop()

The unsigned long measuresESR() is a function prototype called by the variable esrSamples in the main loop. This is the ADC oversampling routine to get a 16bit value from a 10 bit ADC  $(4^6) = 4096$ .

1. Assign two unsigned long samples and int accumulator and set to zero.
2. Integer  $i$  is set to 0.
3. While loop  $i$  less than 4096.
4. Set DISCHARGE\_PIN low to disable discharging the capacitor low.
5. Set PULSE\_PIN low and delay  $1\mu\text{S}$  to make a  $1\mu\text{S}$  pulse.

The variable accumulator is the result from the analogueread(A2).

Pulse is stopped by setting DISCHARGE\_PIN high followed by a delay of  $600\mu\text{s}$ . The variable samples is added from the variable accumulator and the variable  $i$  is increment by 1. This is repeated 4095 times to accumulate the variable samples. The variable samples is right shifted by 6 to decimate the value. The variable samples is returned to the main loop.

#### 4.2.5 An Output Result



Figure 14: Output result of an ESR measurement

## 5. Conclusion

Accurate measurement of capacitance and ESR values has been presented. This method for automatic short circuit detection and correction is novel as it has not used before, to the know knowledge of the authors. The design has

very small form factor by choice of parts used. Simple control and fast measurement speed are achieved by the use of microcontroller which is programmed to perform those specific tasks. The use of ADC oversampling to give a 16bit value for a 10bit ADC improves resolution considerably, thus there is no need for an external ADC which reduces the cost. For future research regarding this project, we would be looking to apply the ESR meter to determine the intrinsic value of the equivalent series resistance (ESR) of the electrolytic capacitors which usually introduces a high AC component in the output voltage of switch mode DC-DC converter.

## References

- [1]Teach Me Micro, "Arduino Nano Pinout," 2 04 2020. [Online]. Available: <https://www.teachmemicro.com/arduino-nano-pinout-diagram>.
- [2]Teach Me Micro, "Arduino Uni Pinout," 01 04 2020. [Online]. Available: <https://www.teachmemicro.com/arduino-uno-pinout-diagram>.
- [3]Slide Player, "Auto Balancing Bridge," 22 03 2020. [Online]. Available: <https://slideplayer.com/slide/10968584/>.
- [4]Electronic Notes, "Capacitor Loss Tangent," 27 03 2020. [Online]. Available: [https://www.electronics-notes.com/articles/basic\\_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php](https://www.electronics-notes.com/articles/basic_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php).
- [5]J. Lewis, "ESR 100KHz Impulses," 21 03 2020. [Online]. Available: [https://www.electronics-notes.com/articles/basic\\_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php](https://www.electronics-notes.com/articles/basic_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php).
- [6]Electronic Notes, "ESR Model," 26 03 2020. [Online]. Available: [https://www.electronics-notes.com/articles/basic\\_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php](https://www.electronics-notes.com/articles/basic_concepts/capacitance/esr-dissipation-factor-loss-tangent-q.php).
- [7]Eevblog, "ESR Table," 05 04 2020. [Online]. Available: <https://www.eevblog.com/forum/beginners/esr-values-for-electrolytic-caps/>.
- [8]I2C, "I2C Protocol," 29 03 2020. [Online]. Available: <https://i2c.info/>.
- [9]G. Granet, "Vector Impedance Diagram," 23 03 2020. [Online]. Available: <https://slideplayer.com/slide/10968584/>.
- [10]Display Module, "OLED Display," 04 04 2020. [Online]. Available: <https://www.google.com/search?q=1.3%22oled+spi+display&tbm=isch&source=univ&sa=X&ved=2ahUKEwjw1r7w6-roAhUThlwKHWZ1B0EQsAR6BAGFEAE&biw=1366&bih=584>.
- [11]The Cave Pearl Project, "Give your Arduino a high resolution ADC by oversampling with noise from a toggled pin," 30 03 2020. [Online]. Available: <https://thecavepearlproject.org/2017/02/27/enhancing-arduinos-adc-resolution-by-dithering-oversampling/>.
- [12]Kitronik, "How a Potential Divider / Voltage Divider Works," 28 03 2020. [Online]. Available: <https://www.kitronik.co.uk/blog/potential-divider-voltage-divider/>.
- [13]Reference Designer, "RC Time Constant Derivation," 24 03 2020. [Online]. Available: [http://referencedesigner.com/rfcal/cal\\_05\\_01.php](http://referencedesigner.com/rfcal/cal_05_01.php).
- [14]Sengpielaudio, "Conversion Calculator Time Constant," 23 03 2020. [Online]. Available: <http://www.sengpielaudio.com/calculator-timeconstant.htm>.