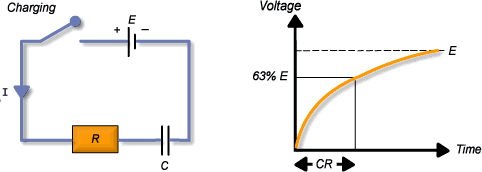
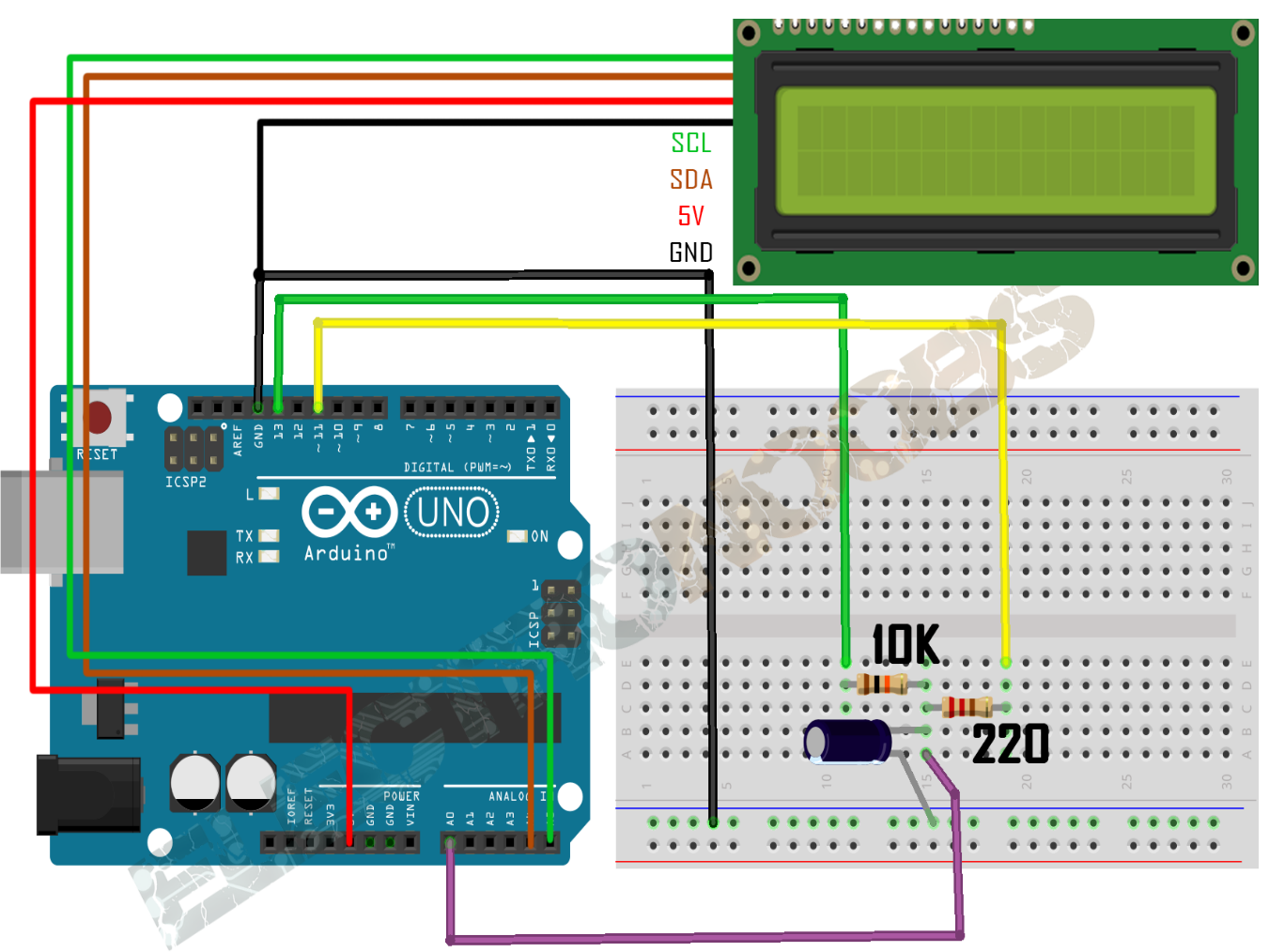
La capacitancia es una medida de la capacidad de "algo" para almacenar carga eléctrica. El medidor de capacitancia Arduino se basa en la misma propiedad básica de los capacitores: la constante de tiempo. **La constante** de **tiempo** de un capacitor se define como el tiempo que tarda el voltaje a través del capacitor en alcanzar el **63.2%** de su voltaje cuando está completamente cargado. Los condensadores más grandes tardan más en cargarse y, por lo tanto, tienen constantes de tiempo más grandes. Un Arduino puede medir la capacitancia porque el tiempo que tarda un capacitor en cargarse está directamente relacionado con su capacitancia por la siguiente ecuación:  
  
TC=RxC  
TC es la constante de tiempo del capacitor (en segundos).  
  
R es la resistencia del circuito (en ohmios).  
  
C es la capacitancia del condensador (en Faradios).  
  
  
Entonces, lo que vamos a hacer es cargar el capacitor a través de una resistencia usando uno de los pines de Arduino. Tenemos que conocer el valor de resistencia de esa resistencia. Usando el ADC del arduino podemos medir el voltaje que alcanzó el capacitor. Comenzamos a contar el sime y cuando el voltaje alcanza el 63.2% de la carga completa paramos la carga y calculamos la capacitancia. Podemos obtener el valor de capacitancia porque conocemos la resistencia del circuito, la tensión medida y el tiempo que tardó en alcanzar el 63,2% de la carga completa. De la ecuación anterior obtenemos C como:

**C = TC / R**

El Arduino mide la capacitancia del capacitor desconocido registrando el tiempo que tarda el capacitor en alcanzar el 63.2% de su voltaje cuando está completamente cargado, luego dividiendo ese valor por la resistencia conocida del circuito. Ahora deberíamos estar listos para configurar los medidores de capacitancia y comenzar a medir algunos capacitores.

**Rango de 0,1 μF a 3900 μF**

El principal problema de esta configuración es el rango de valores que podemos medir. Es casi imposible poder medir capacitores de 1pF a 100F usando un circuito simple y el Arduino. Por eso vamos a montar dos circuitos. Uno para rangos de 0.1uF a 3.9F y el otro de 10pF a 4.7nF. Con estas dos configuraciones podríamos medir capacitores desde 10pF hasta casi 4F.  
  
  
Por lo tanto, usaremos el pin 13 para cargar el condensador a través de la resistencia de 10K ohmios. Una vez que comienza la carga, también iniciamos un contador de tiempo en microsegundos. Medimos la tensión con la entrada analógica A0. El ADC de Arduino tiene 10 bits, por lo que 0 voltios sería 0 y 5 voltios sería 1024. Entonces, 63.2% de 1024 es 648. Cuando la lectura analógica alcanza esos valores, dejamos de cargar el capacitor y también el contador de tiempo. Obtenemos el valor C dividiendo el tiempo transcurrido por el valor de la resistencia utilizada. Conectamos los pines LCD i2c como se muestra en el esquema anterior y cargamos el siguiente código. Para utilizar la pantalla LCD con comunicación i2c, necesitamos instalar la biblioteca lyquid crystal i2c a continuación. Descárguelo e instálelo en su Arduini IDE.

/\*Thanks. Remember to visit my Youtube channel

If you don't whant to Serial print the valeus just delete the serial. print lines

and leave just the LCD print ones.

I've used a i2c LCD screen module.

\*/

//LCD config

#**include** <Wire.h>

#**include** <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3f,20,4); //sometimes the adress is not 0x3f. Change to 0x27 if it dosn't work.

#**define** analogPin 0

#**define** chargePin 13

#**define** dischargePin 8

#**define** resistorValue 10000.0F //Remember, we've used a 10K resistor to charge the capacitor

**unsigned** **long** startTime;

**unsigned** **long** elapsedTime;

**float** microFarads;

**float** nanoFarads;

**void** setup(){

pinMode(chargePin, OUTPUT);

digitalWrite(chargePin, LOW);

lcd.init();

lcd.backlight();

}

**void** loop(){

digitalWrite(chargePin, HIGH);

startTime = micros();

while(analogRead(analogPin) < 648){

}

elapsedTime= micros() - startTime;

microFarads = ((**float**)elapsedTime / resistorValue) ;

if (microFarads > 1){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SCALE: 0.1uF-4F");

lcd.setCursor(0,1);

lcd.print(microFarads);

lcd.setCursor(14,1);

lcd.print("uF");

delay(500);

}

else{

nanoFarads = microFarads \* 1000.0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SCALE: 0.1uF-4F");

lcd.setCursor(0,1);

lcd.print(nanoFarads);

lcd.setCursor(14,1);

lcd.print("nF");

delay(500);

}

digitalWrite(chargePin, LOW);

pinMode(dischargePin, OUTPUT);

digitalWrite(dischargePin, LOW); //discharging the capacitor

while(analogRead(analogPin) > 0){

}//This while waits till the capaccitor is discharged

pinMode(dischargePin, INPUT); //this sets the pin to high impedance

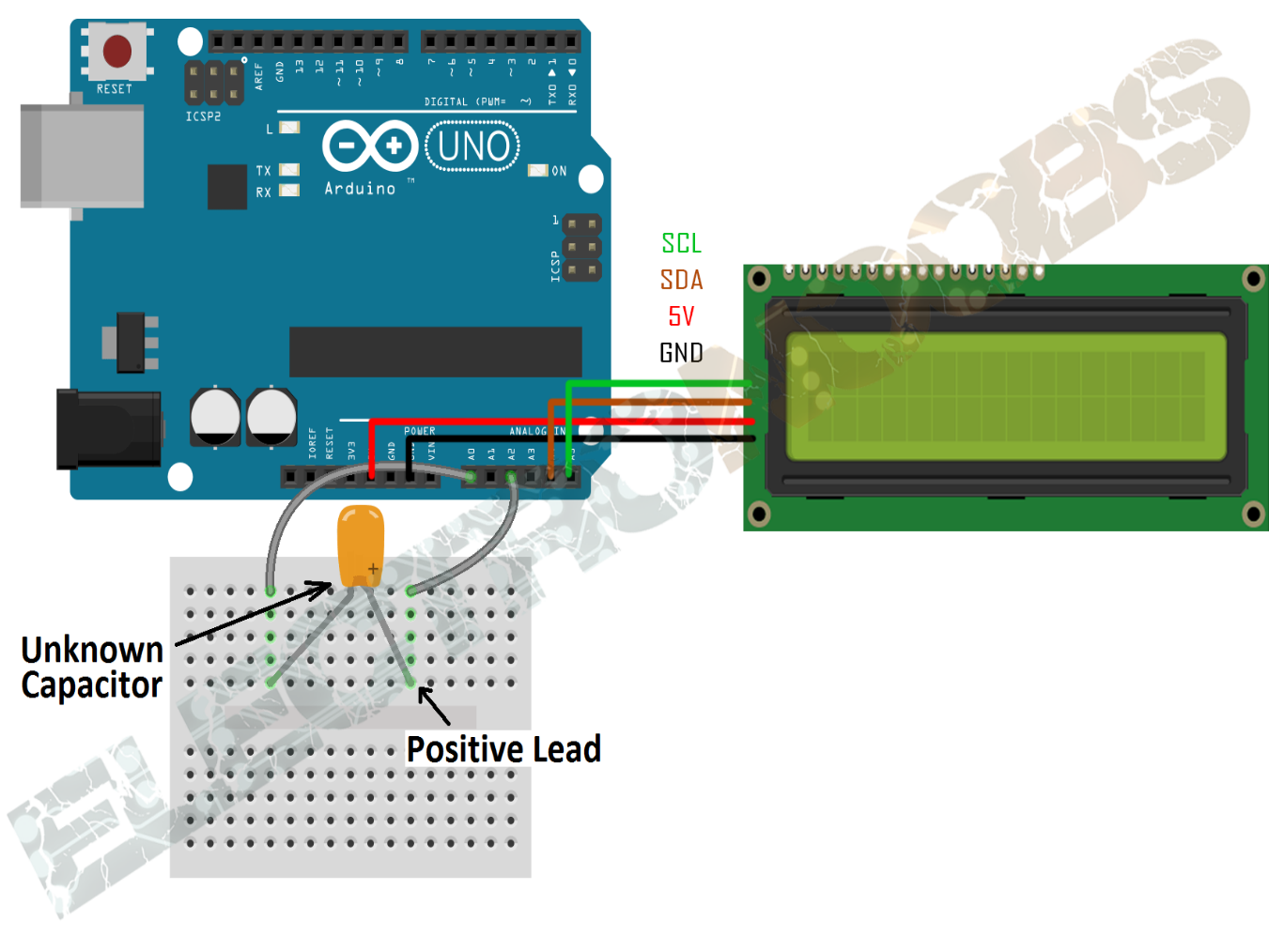
lcd.setCursor(0,0);

lcd.print("DISCHARGING.....");

lcd.setCursor(0,1);

}

## Rango de 1 pF a 1 nF

Vimos en el ejemplo anterior que medimos la capacitancia en colud simplemente cargando el capacitor. Pero el ejemplo anterior tiene muchos errores para condensadores pequeños con un rango entre 1pF y 1nF. Así que este próximo ejemplo medirá con mucha más precisión pequeños condensadores. Finalmente, todo lo que tenemos que hacer es cambiar entre los dos rangos usando un conector de interruptor para hacer una de las entradas de Arduino.  
  
  
Usaremos A2 para cargar el condensador y A0 para descargarlo. Este circuito es bastante simple y todo está basado en el código. No hay resistencia adicional, solo las resistencias internas pullup y pulldown. Observe y cargue el código a continuación. Conecte la pantalla LCD a los pines i2c y comience a medir capacitores. En el siguiente paso, intentaremos combinar los dos ejemplos para tener un rango total de 1pF a 4F. Recuerde instalar la biblioteca lyquid crystal i2c a continuación.

/\*Thanks. Remember to visit my Youtube channel

If you don't whant to Serial print the valeus just delete the serial. print lines

and leave just the LCD print ones.

I've used a i2c LCD screen module.

\*/

//LCD config

#**include** <Wire.h>

#**include** <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3f,20,4); //sometimes the adress is not 0x3f. Change to 0x27 if it dosn't work.

**const** **int** OUT\_PIN = A2;

**const** **int** IN\_PIN = A0;

**const** **float** IN\_STRAY\_CAP\_TO\_GND = 24.48;

**const** **float** IN\_CAP\_TO\_GND = IN\_STRAY\_CAP\_TO\_GND;

**const** **float** R\_PULLUP = 34.8;

**const** **int** MAX\_ADC\_VALUE = 1023;

**void** setup()

{

pinMode(OUT\_PIN, OUTPUT);

pinMode(IN\_PIN, OUTPUT);

lcd.init();

lcd.backlight();

}

**void** loop()

{

pinMode(IN\_PIN, INPUT);

digitalWrite(OUT\_PIN, HIGH);

**int** val = analogRead(IN\_PIN);

digitalWrite(OUT\_PIN, LOW);

if (val < 1000)

{

pinMode(IN\_PIN, OUTPUT);

**float** capacitance = (**float**)val \* IN\_CAP\_TO\_GND / (**float**)(MAX\_ADC\_VALUE -

val);

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

lcd.setCursor(0,1);

lcd.print(capacitance,3);

lcd.setCursor(14,1);

lcd.print("pF");

}

else

{

pinMode(IN\_PIN, OUTPUT);

delay(1);

pinMode(OUT\_PIN, INPUT\_PULLUP);

**unsigned** **long** u1 = micros();

**unsigned** **long** t;

**int** digVal;

do

{

digVal = digitalRead(OUT\_PIN);

**unsigned** **long** u2 = micros();

t = u2 > u1 ? u2 - u1 : u1 - u2;

}

while ((digVal < 1) && (t < 400000L));

pinMode(OUT\_PIN, INPUT);

val = analogRead(OUT\_PIN);

digitalWrite(IN\_PIN, HIGH);

**int** dischargeTime = (**int**)(t / 1000L) \* 5;

delay(dischargeTime);

pinMode(OUT\_PIN, OUTPUT);

digitalWrite(OUT\_PIN, LOW);

digitalWrite(IN\_PIN, LOW);

**float** capacitance = -(**float**)t / R\_PULLUP / log(1.0 - (**float**)val / (**float**)MAX\_ADC\_VALUE);

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

if (capacitance > 1000.0)

{

lcd.setCursor(0,1);

lcd.print(capacitance / 1000.0, 2);

lcd.setCursor(14,1);

lcd.print("uF ");

}

else

{

lcd.setCursor(0,1);

lcd.print(capacitance);

lcd.setCursor(14,1);

lcd.print("nF");

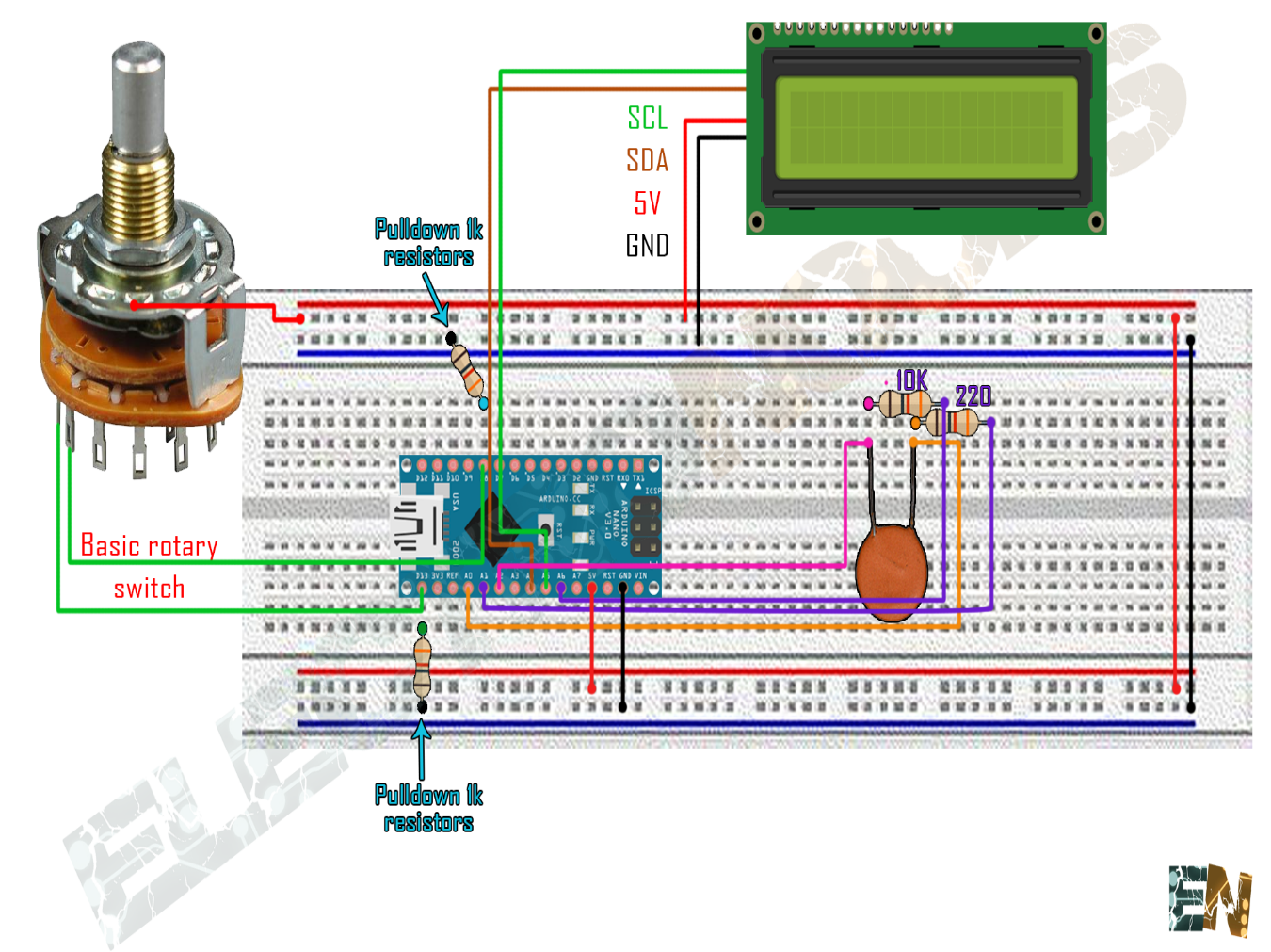
}

}

while (micros() % 1000 != 0);

}

# Medidor de capacitancia todo en uno

Lo que haremos ahora es combinar los dos esquemas anteriores en uno solo. Tendremos que alternar entre escalas y para eso usaremos un interruptor giratorio conectado a 5 voltios y remolcaremos los pines del interruptor a los pines digitales D13 y D8 como podemos ver en el esquema a continuación. Usamos estos pines porque en el futuro quiero combinar este medidor de capacitancia con el [medidor de resistencia](http://electronoobs.com/eng_arduino_tut10.php) que hemos creado antes.  
  
  
Entonces, para el rango de valores altos usaremos los pines A1 (violeta) y A6 (violeta) para cargar y descargar el capacitor y el pin analógico A0 (naranja) para medir el voltaje. Para los valores bajos usaremos A1 (violeta) y A0 (naranja) para cargar y descargar el condensador pequeño. En el código cuando cambiemos la escala tendremos que definir como entradas los pines que no usaremos para darles una impedancia infinita para asegurarnos de que ninguna corriente fluirá por esos pines.

/\*Thanks. Remember to visit my Youtube channel

I've used a i2c LCD screen module.

\*/

//LCD config

#**include** <Wire.h>

#**include** <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3f,20,4); //sometimes the adress is not 0x3f. Change to 0x27 if it dosn't work.

**int** ScalepF=8;

**int** ScalenF=13;

//High values//

#**define** analogPin A0

#**define** chargePin 5

#**define** dischargePin 6

#**define** resistorValue 10000.0F //Remember, we've used a 10K resistor to charge the capacitor

**unsigned** **long** startTime;

**unsigned** **long** elapsedTime;

**float** microFarads;

**float** nanoFarads;

//Low values//

**const** **int** OUT\_PIN = A2;

**const** **int** IN\_PIN = A0;

**const** **float** IN\_STRAY\_CAP\_TO\_GND = 50.28; //We have to change te resistance in this configuration. The 10K and 220 resistors

//are changeing the values

**const** **float** IN\_CAP\_TO\_GND = IN\_STRAY\_CAP\_TO\_GND;

**const** **float** R\_PULLUP = 30.0;

**const** **int** MAX\_ADC\_VALUE = 1023;

**void** setup() {

lcd.init();

lcd.backlight();

pinMode(ScalepF,INPUT);

pinMode(ScalenF,INPUT);

pinMode(OUT\_PIN, OUTPUT);

pinMode(IN\_PIN, OUTPUT);

pinMode(chargePin, OUTPUT);

}

**void** loop() {

//////////////////////////nF/////////////////////////////////////////

if(digitalRead(ScalenF))

{

pinMode(OUT\_PIN,OUTPUT);

digitalWrite(OUT\_PIN,LOW); //to make it GND

pinMode(analogPin,INPUT); //This pin will read the voltage

digitalWrite(chargePin, HIGH);

startTime = micros();

while(analogRead(analogPin) < 648){

}

elapsedTime= micros() - startTime;

microFarads = ((**float**)elapsedTime / resistorValue) ;

if (microFarads > 1){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SCALE: 100nF-4F");

lcd.setCursor(0,1);

lcd.print(microFarads);

lcd.setCursor(14,1);

lcd.print("uF");

delay(500);

}

else{

nanoFarads = microFarads \* 1000.0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("SCALE: 100nF-4F");

lcd.setCursor(0,1);

lcd.print(nanoFarads);

lcd.setCursor(14,1);

lcd.print("nF");

delay(500);

}

digitalWrite(chargePin, LOW);

pinMode(dischargePin, OUTPUT);

digitalWrite(dischargePin, LOW); //discharging the capacitor

while(analogRead(analogPin) > 0){

}//This while waits till the capaccitor is discharged

pinMode(dischargePin, INPUT); //this sets the pin to high impedance

lcd.setCursor(0,0);

lcd.print("DISCHARGING.....");

lcd.setCursor(0,1);

}

//////////////////////////pF/////////////////////////////////////////

if(digitalRead(ScalepF))

{

pinMode(chargePin,INPUT);

pinMode(dischargePin,INPUT);//We give high impedance to the two pins. We don't use this pins

pinMode(IN\_PIN, INPUT);

digitalWrite(OUT\_PIN, HIGH);

**int** val = analogRead(IN\_PIN);

digitalWrite(OUT\_PIN, LOW);

if (val < 976)

{

pinMode(IN\_PIN, OUTPUT);

**float** capacitance = (**float**)val \* IN\_CAP\_TO\_GND / (**float**)(MAX\_ADC\_VALUE - val);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

lcd.setCursor(0,1);

lcd.print(capacitance,3);

lcd.setCursor(14,1);

lcd.print("pF");

delay(200);

}

else

{

pinMode(IN\_PIN, OUTPUT);

delay(1);

pinMode(OUT\_PIN, INPUT\_PULLUP);

**unsigned** **long** u1 = micros();

**unsigned** **long** t;

**int** digVal;

do

{

digVal = digitalRead(OUT\_PIN);

**unsigned** **long** u2 = micros();

t = u2 > u1 ? u2 - u1 : u1 - u2;

}

while ((digVal < 1) && (t < 400000L));

pinMode(OUT\_PIN, INPUT);

val = analogRead(OUT\_PIN);

digitalWrite(IN\_PIN, HIGH);

**int** dischargeTime = (**int**)(t / 1000L) \* 5;

delay(dischargeTime);

pinMode(OUT\_PIN, OUTPUT);

digitalWrite(OUT\_PIN, LOW);

digitalWrite(IN\_PIN, LOW);

**float** capacitance = -(**float**)t / R\_PULLUP / log(1.0 - (**float**)val / (**float**)MAX\_ADC\_VALUE);

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

if (capacitance > 1000.0)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

lcd.setCursor(0,1);

lcd.print(capacitance / 1000.0, 3);

lcd.setCursor(14,1);

lcd.print("uF ");

delay(200);

}

else

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Scale: 1pF-1nF");

lcd.setCursor(0,1);

lcd.print(capacitance,3);

lcd.setCursor(14,1);

lcd.print("nF");

delay(200);

}

}

while (micros() % 1000 != 0);

}

}