

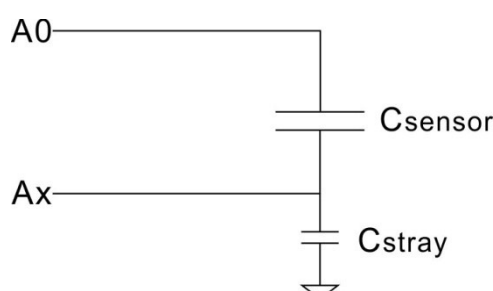
Supporting Information

Defect-Free, High Resolution Patterning of Liquid Metals by Reversibly Sealed, Reusable Polydimethylsiloxane Microchannels for Flexible Electronic Applications

Jinbin Yang,^{a,†} Tingjiao Zhou,^{a,†} Liyun Zhang,^a Deyong Zhu,^a Stephan Handschuh-Wang,^a Zhou Liu,^a Tiantian Kong,^b Yizhen Liu,^a Junmin Zhang,^a and Xuechang Zhou^{a,*}

Email: xczhou@szu.edu.cn

The working principle of the capacitive touch sensor with Arduino.



The circuit of capacitive touch sensor with Arduino

C_{sensor} is the capacitance of the capacitive touch sensor, and C_{stray} is the stray capacitance of the board and of the micro-controller. First, both capacitors are discharged and $A0$ is 0 V. When $A0$ is raised to 5 V the current will flow through both capacitors. Then the voltage on Ax will change. The value of Ax is proportional to the ratio of C_{sensor} divided by the total capacitance ($C_{\text{total}} = C_{\text{stray}} + C_{\text{sensor}}$). This dependence is depicted in Equation 1:

$$\frac{V_{Ax}}{V_{A0}} = \frac{C_{\text{sensor}}}{C_{\text{stray}} + C_{\text{sensor}}} \quad 1$$

Finally, this formula can be rearranged to Equation 2:

$$C_{\text{sensor}} = \frac{V_{Ax} \times C_{\text{stray}}}{V_{A0} - V_{Ax}} \quad 2$$

Where V_{Ax} is the voltage of Ax , and V_{A0} is the voltage of $A0$. V_{A0} , C_{stray} are constant. When the value of V_{Ax} is measured, the value of C_{sensor} is obtained. It is obvious that C_{sensor} will change when a finger touches the sensor, and this change can be detected by Arduino. As a result, touched and untouched state can be recognized.

The control of PowerPoint using capacitive touch keyboard

A capacitive touch keyboard is connected to the computer via a serial port. When finger-touch is detected, Arduino sends corresponding commands to the computer, which are received by the program in the computer. Then the program controls the PowerPoint using "Send Keys" command. In this way, the PowerPoint can be controlled by the capacitive touch keyboard.

Movie S1 shows the program-controlled lighting up of LED lamps in a Flash LED circuit.

Movie S2 illustrates the operation of PowerPoint slides with the as-made touch keyboard.

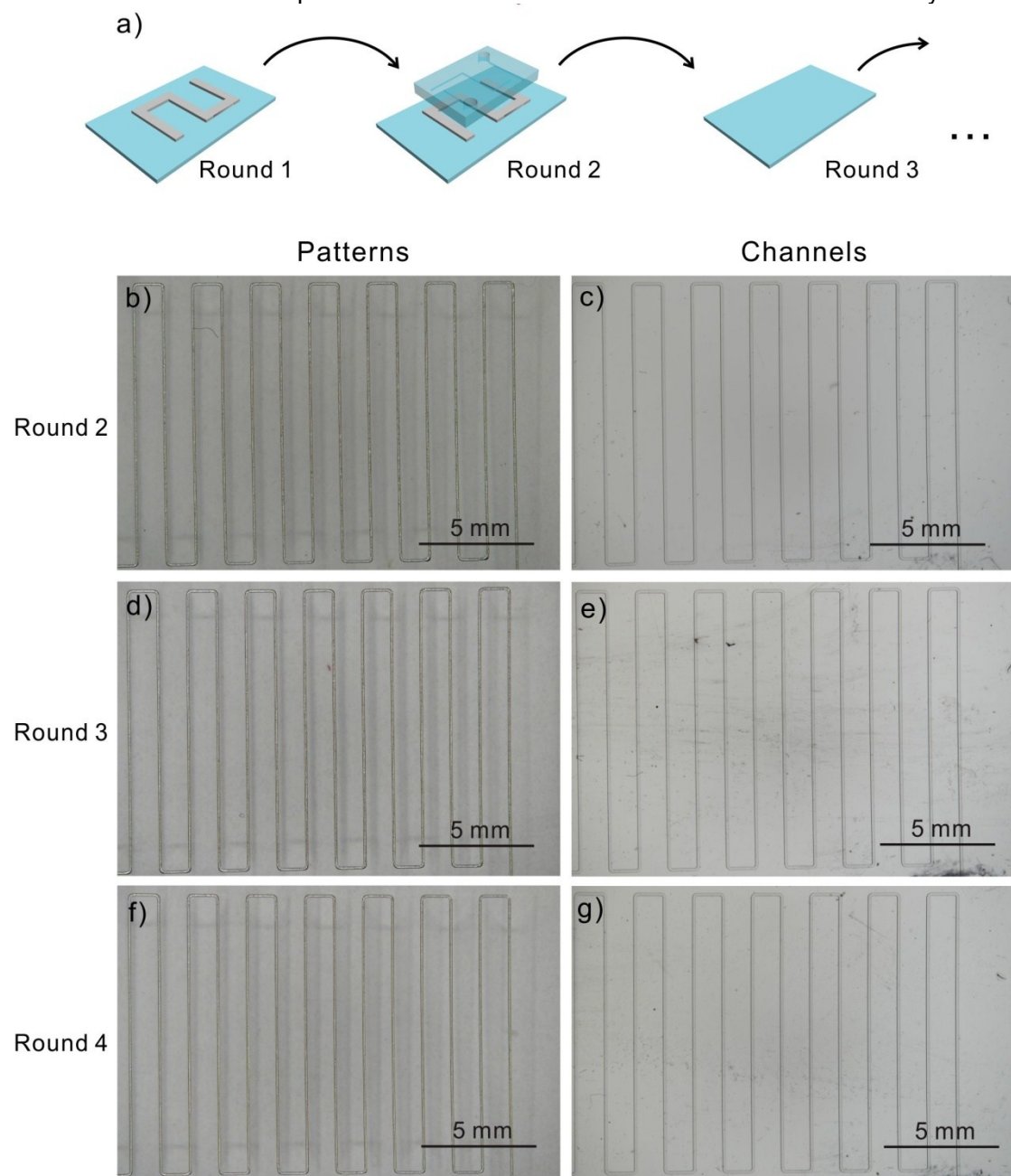


Figure S1. a) Schematic illustration of the recycling of a microchannel mold in the patterning process. b, d, f) Micrographs of the liquid metal patterns obtained using the same mold. c, e, g) Micrographs of the microchannel mold after recycling.

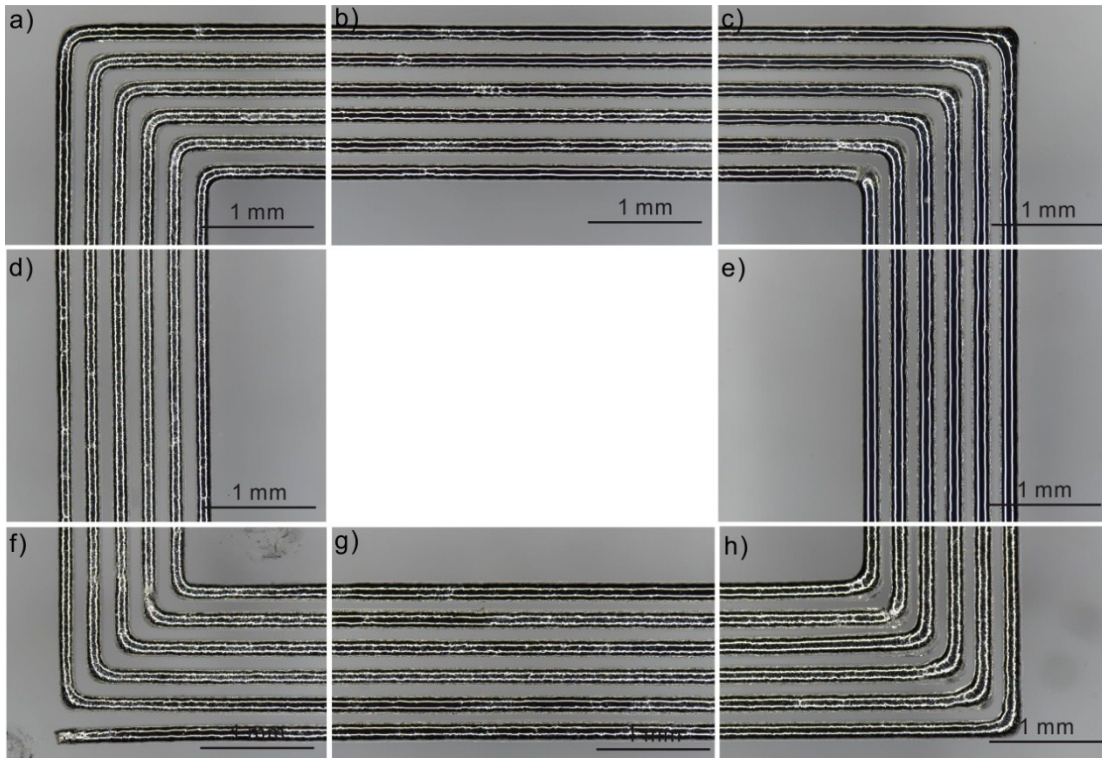


Figure S2. a-h) Micrographs of liquid metal coil pattern in detail.

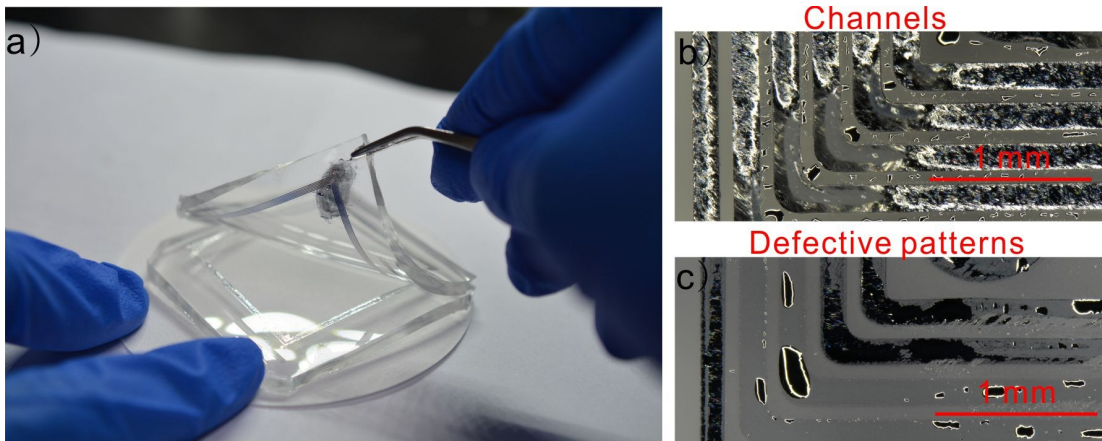


Figure S3. a) Digital image showing the peeling off of a PDMS microchannel mold without freezing the liquid metal. Micrographs of microchannels with lots of liquid metal residues (b) and the obtained defective patterns (c).

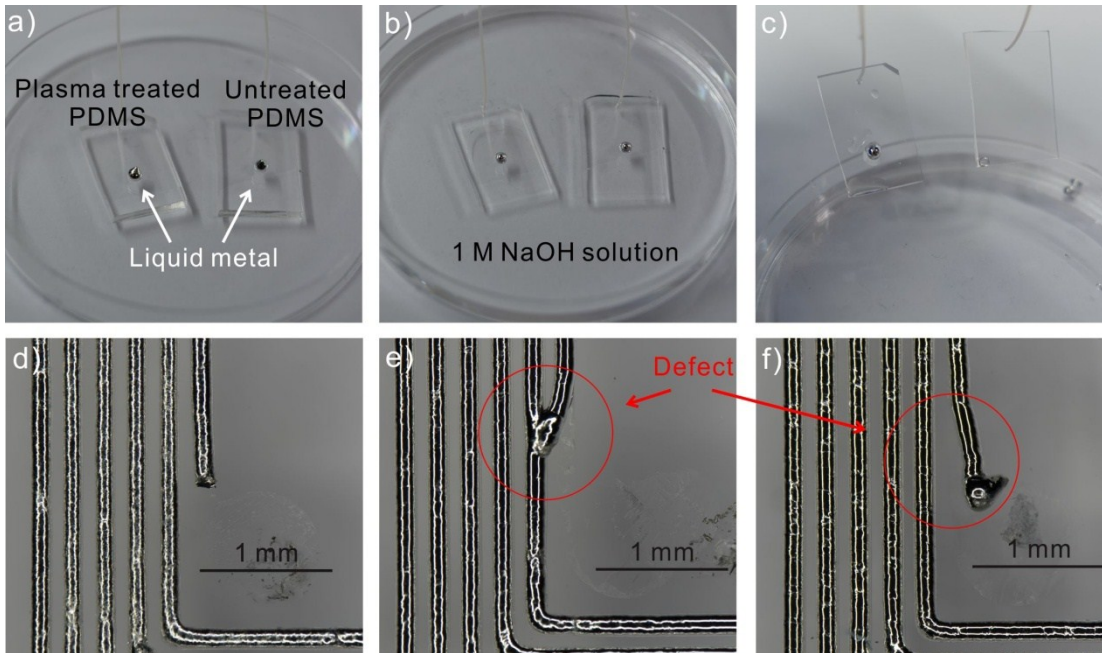


Figure S4. a) Plasma treated PDMS and untreated PDMS, both decorated with a liquid metal droplet. b) The droplets in a) were immersed in a 1 M NaOH solution. c) The two PDMS patches were removed from the NaOH solution. The droplet on air plasma treated PDMS was still adhesive to the substrate. d) The metal pattern obtained by peeling off on a hot plate. e, f) Metal patterns obtained without peeling off on a hot plate. Here, defects are clearly visible.

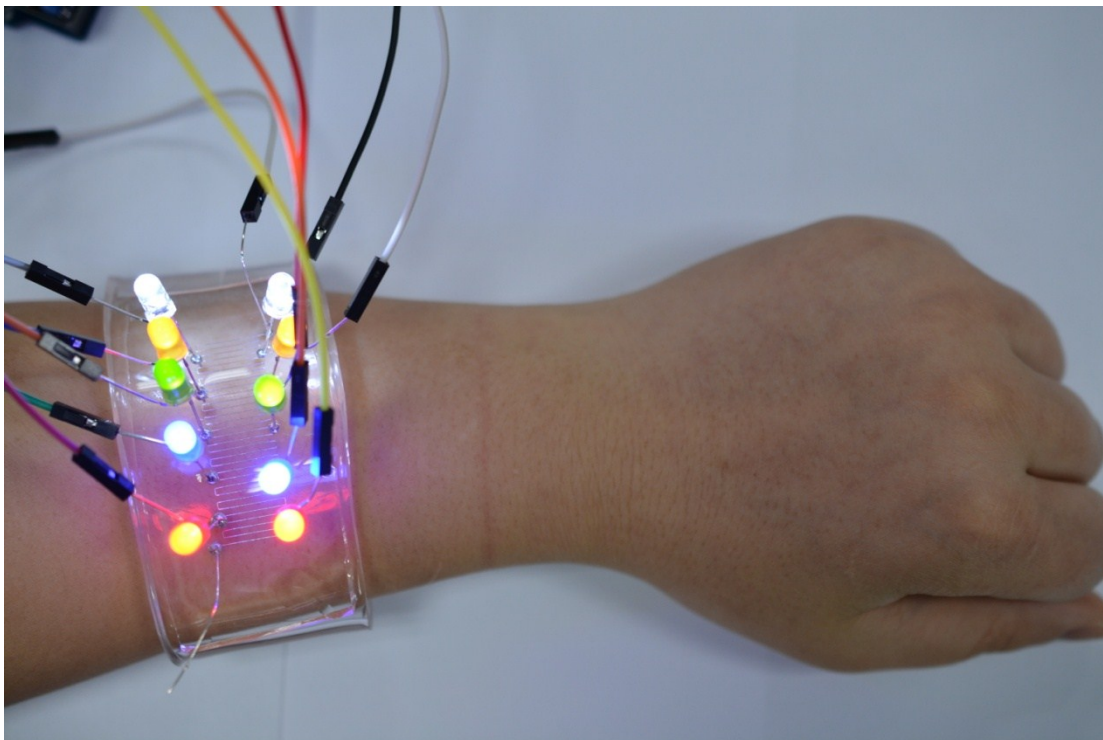


Figure S5. Digital image showing the wearable flash LED circuit as a wristband.

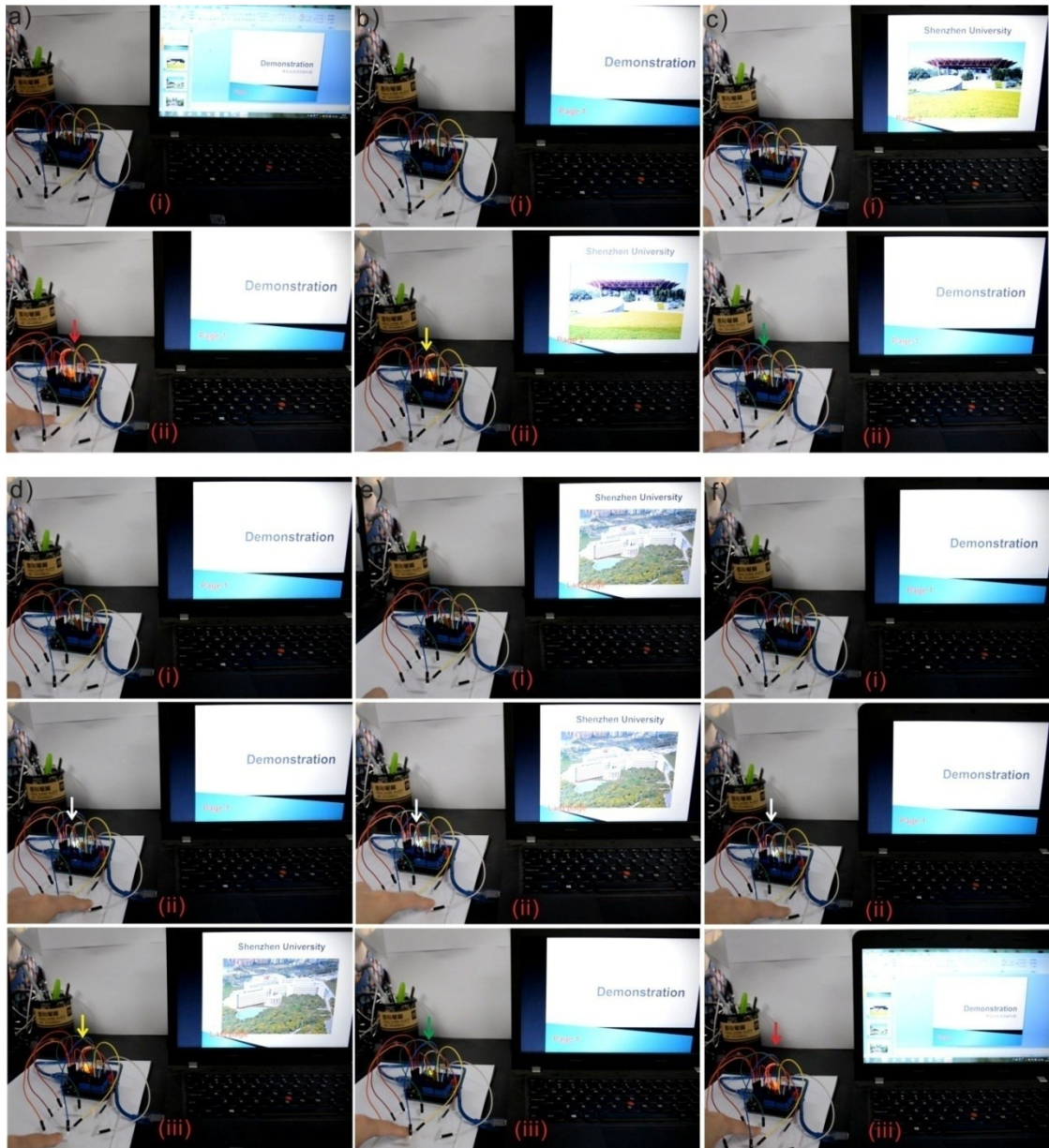


Figure S6. Capacitive touch keyboard connected to a computer for PowerPoint control. a) Key A was touched and the slide show began. b) Key C was touched and the software responds with “go to the next page”. c) Key B was touched and PowerPoint went to the previous page. d) First key D and then key C were touched. The software responded with “go to the last page”. e) First key D and then key B were touched. PowerPoint responded with “go to the first page”. f) First key D and then key A were touched. The software responded with “exit the slide show”.

```
//Arduino code for flash LED circuit
```

```
long r;
```

```
void setup() {  
  Serial.begin(9600);  
  pinMode(2,OUTPUT) ;  
  pinMode(3,OUTPUT) ;  
  pinMode(4,OUTPUT) ;  
  pinMode(5,OUTPUT) ;  
  pinMode(6,OUTPUT) ;  
  pinMode(7,OUTPUT) ;  
  pinMode(8,OUTPUT) ;  
  pinMode(9,OUTPUT) ;  
  pinMode(10,OUTPUT) ;  
  pinMode(11,OUTPUT) ;  
  digitalWrite(2,LOW);  
  digitalWrite(3,LOW);  
  digitalWrite(4,LOW);  
  digitalWrite(5,LOW);  
  digitalWrite(6,LOW);  
  digitalWrite(7,LOW);  
  digitalWrite(8,LOW);  
  digitalWrite(9,LOW);  
  digitalWrite(10,LOW);  
  digitalWrite(11,LOW);  
}
```

```
void ledflash(int i)  
{digitalWrite(i,HIGH);  
  delay(200);  
  digitalWrite(i,LOW);  
}
```

```
void loop() {  
  
  //Flash mode 1  
  for(int h=2;h<12;h++)  
  {ledflash(h);  
   Serial.println(h);}  
  delay(400);
```

```
//Flash mode 2
```

```
for(int l=11;l>1;l--)
{
  digitalWrite(l,HIGH);
  Serial.println(l);
  delay(400);
}
delay(400);
for(int j=2;j<12;j++)
{
  digitalWrite(j,LOW);
  Serial.println(j);
}
delay(400);
```

//Flash mode 3

```
for(int k=2; k<12;k=k+2)
{ledflash(k);
Serial.println(k);}
for(int g=11;g>1;g=g-2)
{ledflash(g);
Serial.println(g);}
delay(400);
```

//Flash mode 4

```
r=random(2,12);
ledflash(r);
Serial.println(r);
r=random(2,12);
ledflash(r);
Serial.println(r);
r=random(2,12);
ledflash(r);
Serial.println(r);
r=random(2,12);
ledflash(r);
Serial.println(r);
r=random(2,12);
ledflash(r);
Serial.println(r);
```

```
}
```

```
//Arduino code for capacitive touch keyboard
//Adapted from http://wordpress.codewrite.co.uk/pic/2014/01/21/cap-meter-with-arduino-uno/
```

```
const int OUT_PIN = A0;
const int IN_PIN1 = A1;
const int IN_PIN2 = A2;
const int IN_PIN3 = A3;
const int IN_PIN4 = A4;
float c1;
float c2;
float c3;
float c4;
```

```
//Capacitance between IN_PIN and Ground
//Stray capacitance is always present. Extra capacitance can be added to
//allow higher capacitance to be measured.
const float IN_STRAY_CAP_TO_GND = 24.48; //initially this was 30.00
const float IN_EXTRA_CAP_TO_GND = 0.0;
const float IN_CAP_TO_GND = IN_STRAY_CAP_TO_GND +
IN_EXTRA_CAP_TO_GND;
const int MAX_ADC_VALUE = 1023;
```

```
// Indicator light
int ledStart=4;
int ledPrevious=5;
int ledNext=6;
int ledCombine=7;
```

```
//Threshold. Threshold should be adjusted in different conditions.
float n1=6.80;
float n2=7.00;
float n3=6.40;
float n4=3.45;
```

```
void setup()
{
  pinMode(OUT_PIN, OUTPUT);
  pinMode(IN_PIN1, OUTPUT);
  pinMode(IN_PIN2, OUTPUT);
  pinMode(IN_PIN3, OUTPUT);
  pinMode(IN_PIN4, OUTPUT);
```



```

pinMode(ledStart, OUTPUT);
pinMode(ledPrevious,OUTPUT);
pinMode(ledNext,OUTPUT);
pinMode(ledCombine,OUTPUT);

Serial.begin(9600);
}

// Indicator light flash.
void led(int i)
{digitalWrite(i,HIGH);
delay(1000);
digitalWrite(i,LOW);}

//Measurement of capacitance.
void cap()
{
  for(int q=0;q<10;q++)
  {
    //Capacitor under test between OUT_PIN and IN_PIN
    //Rising high edge on OUT_PIN
    pinMode(IN_PIN1, INPUT);
    pinMode(IN_PIN2, INPUT);
    pinMode(IN_PIN3, INPUT);
    pinMode(IN_PIN4, INPUT);
    digitalWrite(OUT_PIN, HIGH);
    int val1 = analogRead(IN_PIN1);
    int val2 = analogRead(IN_PIN2);
    int val3 = analogRead(IN_PIN3);
    int val4 = analogRead(IN_PIN4);

    //Discharge for next measurement
    digitalWrite(OUT_PIN, LOW);
    pinMode(IN_PIN1, OUTPUT);
    pinMode(IN_PIN2, OUTPUT);
    pinMode(IN_PIN3, OUTPUT);
    pinMode(IN_PIN4, OUTPUT);

    //Calculate
    c1 = c1+(float)val1 * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE - val1);
    c2 = c2+(float)val2 * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE - val2);
    c3 = c3+(float)val3 * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE - val3);
    c4 = c4+(float)val4 * IN_CAP_TO_GND / (float)(MAX_ADC_VALUE - val4);
  }
}

```

```

// Averaging
c1=c1/10;
c2=c2/10;
c3=c3/10;
c4=c4/10;
}

void loop()
{
  cap();//Measure the capacitance

  if(c1<n1) //Detect the change of capacitance
  {delay(1);
  cap();
  if(c1<n1);
    {delay(1);
    cap();
    if(c1<n1)
    {Serial.println("1") ;//Send command
    led(ledStart); // Indicator light flash
    delay(500);
    }
  }
}

else if (c2<n2)
  {delay(1);
  cap();
  if(c2<n2);
    {delay(1);
    cap();
    if(c2<n2)
    {Serial.println("2") ;
    led(ledPrevious);
    delay(500);
    }
  }
}

else if(c3<n3)
  {delay(1);
  cap();
  if(c3<n3);
    {delay(1);

```

```
        cap();
        if(c3<n3)
        {Serial.println("3" );
          led(ledNext);
          delay(500);
        }
    }
}

else if(c4<n4)
{delay(1);
cap();
if(c4<n4);
    {delay(1);
      cap();
      if(c4<n4)
      {Serial.println("4" );
        led(ledCombine);
        delay(500);
      }
    }
}
while (millis() % 50 != 0)
;

}
```

'Visual Basic code for PowerPoint control program "COM2keys"
' Adapted from <http://www.crystalradio.cn/thread-321839-1-1.html>

```
Private Sub Form_Load()  
Form1.WindowState = vbMinimized  
MSComm1.Settings = "9600,n,8,1"  
MSComm1.CommPort = 3  
MSComm1.InputLen = 0  
MSComm1.InBufferSize = 512  
MSComm1.InBufferCount = 0  
MSComm1.OutBufferCount = 0  
MSComm1.RThreshold = 1  
MSComm1.PortOpen = True  
End Sub
```

```
'Receive commands form touch keyboard through serial port  
Private Sub MSComm1_OnComm()  
Select Case MSComm1.CommEvent  
Case comEvReceive  
Inputsignal = MSComm1.Input  
Case Else  
End Select
```

```
'Control the PowerPoint in the computer  
If Inputsignal = 1 Then  
SendKeys ("+F5")  
Elseif Inputsignal = 2 Then  
SendKeys ("UP")  
Elseif Inputsignal = 3 Then  
SendKeys ("DOWN")  
Elseif Inputsignal = 4 Then  
Do  
Select Case MSComm1.CommEvent  
Case comEvReceive  
Inputsignal2 = MSComm1.Input  
Case Else  
End Select  
  
Select Case Inputsignal2  
Case 2  
SendKeys ("HOME")  
Case 3  
SendKeys ("END")  
Case 1
```

```
SendKeys ("{ESC}")
```

```
End Select
```

```
Loop While Inputsignal2 <> 2 And Inputsignal2 <> 3 And Inputsignal2 <> 1
```

```
End If
```

```
End Sub
```