Fibre photodetector

Matlab

```
clear all;
delete(instrfindall);
arduino=serial('COM36','BAUD', 9600);

fopen(arduino);


data = [0,0;0,0;0,0;0,0];
row=[1 4];
col=[1 2];
im=imagesc(row,col,data);
hold on;
map = [1 0 0 ;1 1 1 ];
colormap(map)



while(1)

    matlabreading = fscanf(arduino);

    reading=char(matlabreading);
    reading=str2num(reading);


    data(1,1)=1-bitget(reading,1);
    data(2,1)=1-bitget(reading,2);
    data(3,1)=1-bitget(reading,3);
    data(4,1)=1-bitget(reading,4);

    data(1,2)=1-bitget(reading,5);
    data(2,2)=1-bitget(reading,6);
    data(3,2)=1-bitget(reading,7);
    data(4,2)=1-bitget(reading,8);

    %disp(data);

    set(im, 'CData', data);
    drawnow;



end
```

Arduino

```
const int numreadings = 40; //sampling rate
```

```
int readings[8][numreadings];      // the readings[i] from the analog input
int readIndex = 0;              // the index of the current reading
int total[8] = {0};                // the running total[i]
int average[8] = {0};              // the average[i]
int average_map[8] = {0};
int readings_min[8] = {0, 0, 0, 12, 0, 0, 0, 0};
int readings_max[8] = {15, 12, 20, 450, 45, 300, 40, 100};
int average_map_last[8] = {0};
int threshold[8] = {25, 16, 15, 18, 13, 38, 15, 20};
int uv[8] = {0};
int count = 0;
void setup() {
  // initialize serial communication with computer:
  Serial.begin(9600);
  // initialize all the readings[i] to 0:
  for (int i = 0; i < 7; i++) {
    for (int thisReading = 0; thisReading < numreadings; thisReading++) {
      readings[i][thisReading] = 0;
    }
    total[i] = 0;
    average[i] = 0;
    average_map[i] = 0;
    average_map_last[i] = 0;
    uv[i] = {0};
  }
}

void loop() {
  //runing average of each pin
  for (readIndex = 0; readIndex < numreadings; readIndex++) {
```

```
  for (int i = 0; i < 7; i++) {
    // subtract the last reading:
    total[i] = total[i] - readings[i][readIndex];
    // read from the sensor:
    readings[i][readIndex] = analogRead(i);
    // add the reading to the total[i]:
    total[i] = total[i] + readings[i][readIndex];
    // calculate the average[i]:
    average[i] = total[i] / numreadings;
    //average[i] = constrain(average[i], readings_min[i], readings_max[i]);
    average_map[i] = map(average[i], readings_min[i], readings_max[i], 0, 32);
  }
  Serial.print(millis());
  Serial.print(",");
  Serial.print(readings[5][readIndex]);
  Serial.print(",");
  Serial.println(average[5]);





}

for (int i = 0; i < 7; i++) {
  if (average_map[i] > threshold[i] || (average_map[i] - average_map_last[i]) > 4  )  {
    uv[i] = 1;

  }
  else     uv[i] = 0;
}
```

```
  count++;
  if (count == 5) {
    for (int i = 0; i < 7; i++) {
      average_map_last[i] = average_map[i];
    }
    count = 0;
  }

  //Serial.print(uv[0]);
  byte matlabreading;
  matlabreading = uv[0] + uv[1] * 2 + uv[2] * 4 + uv[3] * 8 + uv[4] * 16 + uv[5] * 32 + uv[6] * 64 + uv[7]
* 128;
  //Serial.print(matlabreading);
  // Serial.println();
  delay(10);
  matlabreading = 0;

}
```

F-Touch sensor with 30 points

Mathlab Code

```
clear all;
delete(instrfindall);
arduino=serial('COM28','BAUD', 9600);

fopen(arduino);


data = [1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1];
row=[1 5];
col=[1 6];
im=imagesc(row,col,data);
hold on;
map = [0 0 1;1 1 1 ];
colormap(map)
titlename={
    'Fan ON',   'Light ON', 'Blinds Up',    'Aircon ON',    'Curtain Open';
    'Fan OFF',  'Light OFF','Blinds Down',  'Aircon OFF',   'Curtain Close';
    'Sprinkler ON', 'Hi Siri',  'Temp Up',  'ECG ON',   'Volume Up';
    'Sprinkler OFF',    'OK Google',    'Temp Down',    'ECG OFF',  'Volume
Down';
    'Door Lock',    'Hey Cortana',  'TV ON',    'Camera ON',    'Screen Out';
    'Door Unlock',  'Hi Bixby', 'TV OFF',   'Camera OFF',   'Turn OFF ALL'
};

axis off

while(1)

    matlabreading = fscanf(arduino);

    reading=char(matlabreading);
    reading=str2num(reading);
    maxrow=bitshift(reading,-4);
    maxcol=reading-bitshift(maxrow,4);
    disp(maxrow);
    disp(maxcol);

    title( 'No touch','FontSize', 24);

    data = [1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1;1,1,1,1,1];
    if maxrow~=0
        data(maxrow,6-maxcol)=[-1];

        title( [titlename(maxrow,6-maxcol)],'FontSize', 24);

    end


    set(im, 'CData', data);
```

```
    drawnow;
end
```

Arduino Code

```
#include <Wire.h>
#include <MPR121.h>
#include <Adafruit_GFX.h>
#include "Adafruit_LEDBackpack.h"
Adafruit_BicolorMatrix matrix = Adafruit_BicolorMatrix();

// mapping constrains
#define LOW_DIFF 0
#define HIGH_DIFF 31

//Global variables
int reading[11] = {0};
int row = 0;
int col = 0;
int brightness = 0;

int  gain[11] = {1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1};
#define THRESHOLD 9

/*********************        SETUP    ************************************/
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    delay(10);
  }
```

```
  // Serial.println("MPR121 Touch sensor");

  //Default address is 0x5A, if tied to 3.3V its 0x5B
  // If tied to SDA its 0x5C and if SCL then 0x5D
  if (!MPR121.begin(0x5A)) {
    Serial.println("MPR121 not found, check wiring?");
    while (1);
  }
  // Serial.println("MPR121 found!");

  // slow down some of the MPR121 baseline filtering to avoid
  // filtering out slow hand movements
  MPR121.setRegister(MPR121_NHDF, 0x01); //noise half delta (falling)
  MPR121.setRegister(MPR121_FDLF, 0x3F); //filter delay limit (falling)

  matrix.begin(0x70);  // pass in the address
  // Serial.println("8x8 LED Matrix found!");
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage level)
  digitalWrite(3, HIGH);    // turn the LED off by making the voltage LOW

}

/********************        MAIN    ************************************/
void loop() {
  readingtouch();
  process();
```

```
  LED();
  matlab();
}

void readingtouch() {
  //Get reading of each electrode
  // update all of the data from the MPR121
  MPR121.updateAll();
  // read the difference between the measured baseline and the measured continuous data
  for (int i = 0; i < 12; i++) {

    reading[i] = MPR121.getBaselineData(i) - MPR121.getFilteredData(i);
    //print out the reading value for debug
    //Serial.println(reading[i]);

    // map the LOW_DIFF..HIGH_DIFF range to 0..255 (8-bit resolution for analogWrite)
    reading[i] = constrain(reading[i], LOW_DIFF, HIGH_DIFF);
    //reading[i] = map(reading[i], LOW_DIFF, HIGH_DIFF, 0, 1023);

  }

  //Serial.println();
  // delay(500);

}
void process() {

  int max_rowreading = 0;
  int max_colreading = 0;
```

```
row = 0;
col = 0;

for (int i = 0; i < 12; i++) {
  reading[i] = gain[i] * reading[i];
  //print out the reading value for debug
  //  Serial.println(reading[i]);
}
//Serial.println();
//delay(300);

//Extract the index of the max reading from E0 to E5

for (int i = 0; i < 6; i++) {
  if ( max_rowreading < reading[i]) {
    max_rowreading = reading[i];
    row = i + 1;
  }
}

//Extract the index of the max reading from E6 to E11
for (int i = 6; i < 12; i++) {
  if ( max_colreading < reading[i]) {
    max_colreading = reading[i];
    col = i - 5;
  }
}

//Check if the max_colreading and max_rowreading are valid
```

```
  if (max_rowreading < THRESHOLD || max_colreading < THRESHOLD) {
    max_rowreading = 0;
    max_colreading = 0;
    row = 0;
    col = 0;
  }

  // map the signal strength to brightness
  brightness = map(max_colreading, LOW_DIFF, HIGH_DIFF, 0, 15);

}
void LED() {

  matrix.clear();      // clear display
  matrix.setBrightness(brightness);
  matrix.drawPixel(col - 1, row - 1, LED_RED);
  matrix.writeDisplay();  // write the changes we just made to the display

}

void matlab() {
  byte matlabreading;

  matlabreading = char(row) << 4;
  matlabreading = matlabreading + char(col);

  Serial.print(matlabreading);
  Serial.println();
  delay(10);
```

```
  if (row == 1 && col == 4) {
  digitalWrite(2, LOW);   // turn the LED on (HIGH is the voltage level)
  delay(10);                    // wait for a second

  }

  else if (row == 2 && col == 4) {
  digitalWrite(3, LOW);   // turn the LED on (HIGH is the voltage level)
  delay(10);                    // wait for a second
  }
  else{
     digitalWrite(2, HIGH);   // turn the LED on (HIGH is the voltage level)
       digitalWrite(3, HIGH);   // turn the LED on (HIGH is the voltage level)
  }

}
```

F-Energy

Matlab

```
clear all;

%Create the bar chart
y = [0 0 0 0 0 0 0 0; 0 0 0 0 0 0 0 0;10 5 5 5 5 5 5 12;0 0 0 0 0 0 0 0;0 0
0 0 0 0 0 0];
b=barh(y,'stacked');

%Make edges invisible
for i=1:1:8
    b(i).EdgeColor='none';
end

%Assigned the right color to the bar
b(1).FaceColor = [1 1 1];
b(7).FaceColor = [1 0 0];
b(6).FaceColor = [1 0.5 0];
b(5).FaceColor = [1 1 0];
b(4).FaceColor = [0.8 1 0];
b(3).FaceColor = [0.5 1 0];
b(2).FaceColor = [0 1 0];
b(8).FaceColor = [1 1 1];

%Hide all the bars, and show the suitable bars taht is charged
for i=1:1:8
    alpha(b(i),0);
end
hold on


%Load background image
I =  flipud(imread('discharge.png'));
I2=flipud(imread('charge.png'));
%I = imread('battery.png');
h = image([0 60],[0 6],I);
h2=image([0 60],[0 6],I2);

axis tight;
uistack(b,'top');
drawnow;


arduino=serial('COM3','BAUD', 9600);
fopen(arduino);


while(1)
    matlabreading = fscanf(arduino);
    reading=char(matlabreading);
    reading=str2num(reading);

    if reading>=128

    reading=reading-128;
```

```
    uistack(h,'bottom');
     else

        uistack(h2,'bottom');
     end



     %Hide all the bars, and show the suitable bars taht is charged
     for i=1:1:8
         alpha(b(i),0);
     end

     %Change the charging status
     for i=7:-1:9-reading
         alpha(b(i),1);

     end

     %for stability
     pause(0.01);
     drawnow;




end
```

Arduino

```
//PIN defination
int chargingpin=2;

void setup() {
  // initialize serial communication with computer:
  Serial.begin(9600);
 pinMode(chargingpin,INPUT);
}

void loop() {

 // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 900) to a voltage (0 - 4.5V):
  uint8_t voltage = map(sensorValue,0,900,2,7);
  // print out the value you read:

int charging=digitalRead(chargingpin);

  byte matlabreading;
  matlabreading = voltage+(1-charging)*128;

  Serial.print(matlabreading);
  Serial.println();
  delay(100);





}
```