

/\*\*\*\*\*

Bare Conductive MPR121 library

-----

SimpleTouch.ino - simple MPR121 touch detection demo with serial output

Based on code by Jim Lindblom and plenty of inspiration from the Freescale Semiconductor datasheets and application notes.

Bare Conductive code written by Stefan Dzisiewski-Smith and Peter Krige.

This work is licensed under a MIT license <https://opensource.org/licenses/MIT>

Copyright (c) 2016, Bare Conductive

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

\*\*\*\*\*/

```
#include <MPR121.h>
```

```
#include <Wire.h>
```

```
#define numElectrodes 12
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  while(!Serial); // only needed if you want serial feedback with the
```

```
// Arduino Leonardo or Bare Touch Board
```

```
pinMode(6, OUTPUT);
```

```
Wire.begin();
```

```
// 0x5C is the MPR121 I2C address on the Bare Touch Board
```

```
if(!MPR121.begin(0x5A)){
```

```
  Serial.println("error setting up MPR121");
```

```
  switch(MPR121.getError()){
```

```
    case NO_ERROR:
```

```
      Serial.println("no error");
```

```
      break;
```

```
    case ADDRESS_UNKNOWN:
```

```
      Serial.println("incorrect address");
```

```
      break;
```

```
    case READBACK_FAIL:
```

```
      Serial.println("readback failure");
```

```
      break;
```

```
    case OVERCURRENT_FLAG:
```

```
      Serial.println("overcurrent on REXT pin");
```

```
      break;
```

```
    case OUT_OF_RANGE:
```

```
      Serial.println("electrode out of range");
```

```
      break;
```

```
    case NOT_INITED:
```

```
      Serial.println("not initialised");
```

```
      break;
```

```
    default:
```

```
      Serial.println("unknown error");
```

```
      break;
```

```
  }
```

```
  while(1);
```

```
}
```

```
// pin 4 is the MPR121 interrupt on the Bare Touch Board
```

```
MPR121.setInterruptPin(4);
```

```
// FOR PROXIMITY USE 1 AND 0.5
```

```
// this is the touch threshold - setting it low makes it more like a proximity  
trigger
```

```
// default value is 40 for touch
```

```
MPR121.setTouchThreshold(5);
```

```

// this is the release threshold - must ALWAYS be smaller than the touch
threshold
// default value is 20 for touch
MPR121.setReleaseThreshold(2);

// initial data update
MPR121.updateTouchData();
}

void loop()
{
  if(MPR121.touchStatusChanged()){
    MPR121.updateTouchData();
    for(int i=0; i<numElectrodes; i++){
      if(MPR121.isNewTouch(i)){
        Serial.println(i, DEC);
        delay(20);
        // Serial.println(" t");
        //digitalWrite(6, HIGH);
      } else if(MPR121.isNewRelease(i)){
        // Serial.println(i, DEC);
        // delay(20);
        //Serial.println(" r");
        //digitalWrite(6, LOW);
      }
    }
  }
}
}
}

```