```
/* FILE:    TTP229_16_Key_Capacitive_Touch_Example
   DATE:    25/02/15
   VERSION: 0.1

REVISIONS:

25/02/15 Created version 0.1

This is an example of how to use the Hobby Components 16 key capacitive touch
keypad (HCMODU0079). This example sketch will read the current state of the
of the keypad and if a key is pressed output its key number to the serial port.

The sketch assumes that the keypad is configured to 16 key active low mode
by shorting pads P1-3 and P1-P4 together (see schematic or sport forum for more
information). Connect the keypad to your Arduino as follows:

Keypad......Arduino
VCC.........+5V
GND.........GND
SCL.........Digital pin 8
SDO.........Digital pin 9

You may copy, alter and reuse this code in any way you like, but please leave
reference to HobbyComponents.com in your comments if you redistribute this code.

This software may not be used directly for the purpose of promoting products that
directly compete with Hobby Components Ltd's own range of products.

THIS SOFTWARE IS PROVIDED "AS IS". HOBBY COMPONENTS MAKES NO WARRANTIES,
WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OR
LACK OF NEGLIGENCE. HOBBY COMPONENTS SHALL NOT, IN ANY CIRCUMSTANCES, BE LIABLE
FOR ANY DAMAGES INCLUDING, BUT NOT LIMITED TO, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES FOR ANY REASON WHATSOEVER. */



/* Define the digital pins used for the clock and data */
#define SCL_PIN 8
#define SDO_PIN 9

/* Used to store the key state */
byte Key;

void setup()
{
  /* Initialise the serial interface */
  Serial.begin(9600);
  /* Configure the clock and data pins */
  pinMode(SCL_PIN, OUTPUT);
  pinMode(SDO_PIN, INPUT);
}
```

```
/* Main program */
void loop()
{
  /* Read the current state of the keypad */
  Key = Read_Keypad();

  /* If a key has been pressed output it to the serial port */
  if (Key)
    Serial.println(Key);

  /* Wait a little before reading again
     so not to flood the serial port*/
  delay(100);
}


/* Read the state of the keypad */
byte Read_Keypad(void)
{
  byte Count;
  byte Key_State = 0;

  /* Pulse the clock pin 16 times (one for each key of the keypad)
     and read the state of the data pin on each pulse */
  for(Count = 1; Count <= 16; Count++)
  {
    digitalWrite(SCL_PIN, LOW);

    /* If the data pin is low (active low mode) then store the
       current key number */
    if (!digitalRead(SDO_PIN))
      Key_State = Count;

    digitalWrite(SCL_PIN, HIGH);
  }

  return Key_State;
}
```