



GET LEARNING!

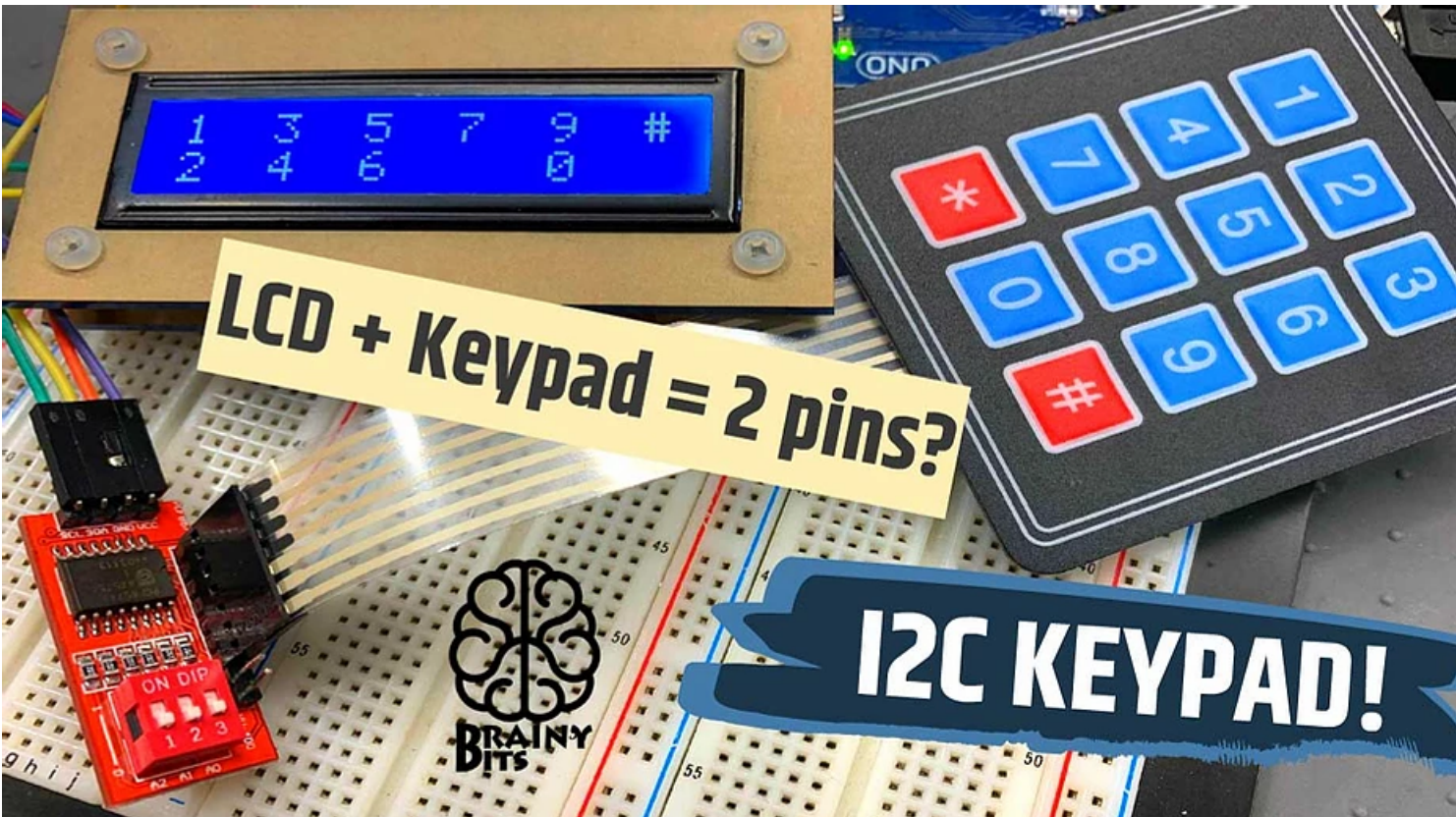
Home Tutorials

Safari browser sometimes has issues displaying images...
*I.e: *you have to click on the images to see them...*

For a better browsing experience on Brainy-Bits
Please use Chrome, Edge or Firefox browser.

Brainy-Bits Oct 26, 2020 5 min read

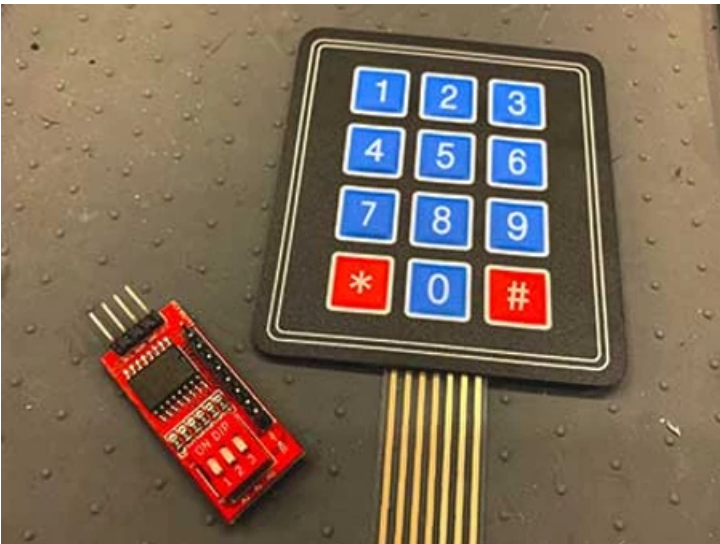
A Keypad + LCD with only 2 Pins?! No problem with the I2C Protocol!



OVERVIEW

In the last tutorial we saw how to use a **i2c Expander module** to connect a **Bourns encoder** and use 2 pins instead of eight to connect it to an Arduino.

In this one we'll use it to connect a Keypad to an Arduino and again save some pins, and also have a quick overview on what and how the i2c protocol works.



PARTS USED

16x2 LCD with Backpack
[Amazon usa](#)
[Amazon canada](#)

I2C Expander Module
[Amazon usa](#)
[Amazon canada](#)

Membrane Keypad
[Amazon usa](#)
[Amazon canada](#)

Arduino UNO
[Amazon usa](#)
[Amazon canada](#)

These are Amazon affiliate links...



GET LEARNING!

Home Tutorials



QUICK OVERVIEW OF THE I2C PROTOCOL

I2C stands for “Inter-Integrated Circuit”, it allows to connect multiple modules or “slave”, and requires only 2 wires no matter the amount of connected modules, on an Arduino you can have up to 128 slave devices.

Each I2C device need to have a different address so the Arduino can send or get the information from the right module.

Those address are in hex values (ex. 0x20), and most module give you the possibility to change the address by either soldering some pads or using dip switches like the I2C Expander module we used.

Some module have fixed address that cannot be changed, so you couldn’t use two of those modules on the same I2C bus.

We will see how to get around that in a future tutorial!

Btw you can also have more than one Arduino connected to the same I2C modules and be able to access them, of course not at the same time.

Even though I2C is pretty fast, it does have some overhead since for every 8 bits of information, there will be one extra bit for the i2c protocol.

PS615 - Available - Starting at: \$Submit for Price

Tripp Lite ® Authorized Distributor

onlinecomponents.com

Learn More

I2C uses two pins or signals, one is SCL and the other is SDA. Most Arduinos have only one I2C bus, one exception would be the Arduino Due which has two seperate I2C bus.

The pins (SCL and SDA) are predefined on Arduinos, for example on the UNO those are A4=SDA A5=SCL, and on the MEGA2560 those are 20=A4 and 21=SCL.

SCL provides the clock, and SDA is the data.

It’s good to know that the I2C protocol can communicate both ways, so can be used to send as well as receive information.

THE I2C EXPANDER MODULE VS THE I2C LCD BACKPACK

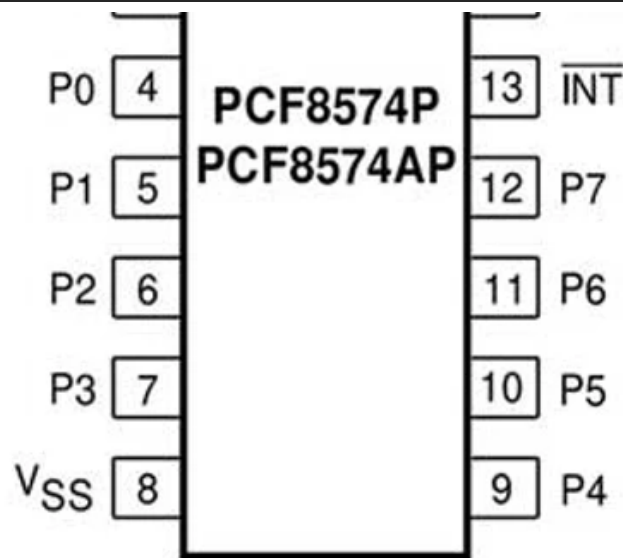


Both modules use the same chip at their core, the PCF8574T.



🔍 Search...

Tutorials



Each one of those 8 pins/ports (P0-P7) can be independently used as an input, like we used it for when connecting the Bourns encoder **or** as an output to control some LEDs for example.

So in the case of the I2C LCD Backpack module, those are used as output to send information to the LCD.

Offers a great collection of electronic parts datasheet.

Learn

The main difference between those two modules, is their pinouts, the LCD Backpack pinout is made to fit on an LCD with additional outputs for the backlight.

The I2C-Expander on the other hand, has all 8 pins in order and clearly marked, make it easy to connect.



Since both the I2C port Expander and the I2C LCD Backpack are pretty much the same, couldn't I just use the LCD Backpack to connect the Bourns encoder?

Well that's what I thought at first....

You see the LCD Backpack pinout is different, but it's easy to see which pin is connected to which pin on the PCF8574 by using a multimeter.

But to my surprise, it seems that the PCF8574 I/O pin #3 is not connected to anything?!

I don't know if the're all like that, but all the ones i've tested had the same problem.

Maybe yours has all the pins connected properly but you'll have to check that out with a multimeter.

Tripp Lite® Authorized Distributor

Learn

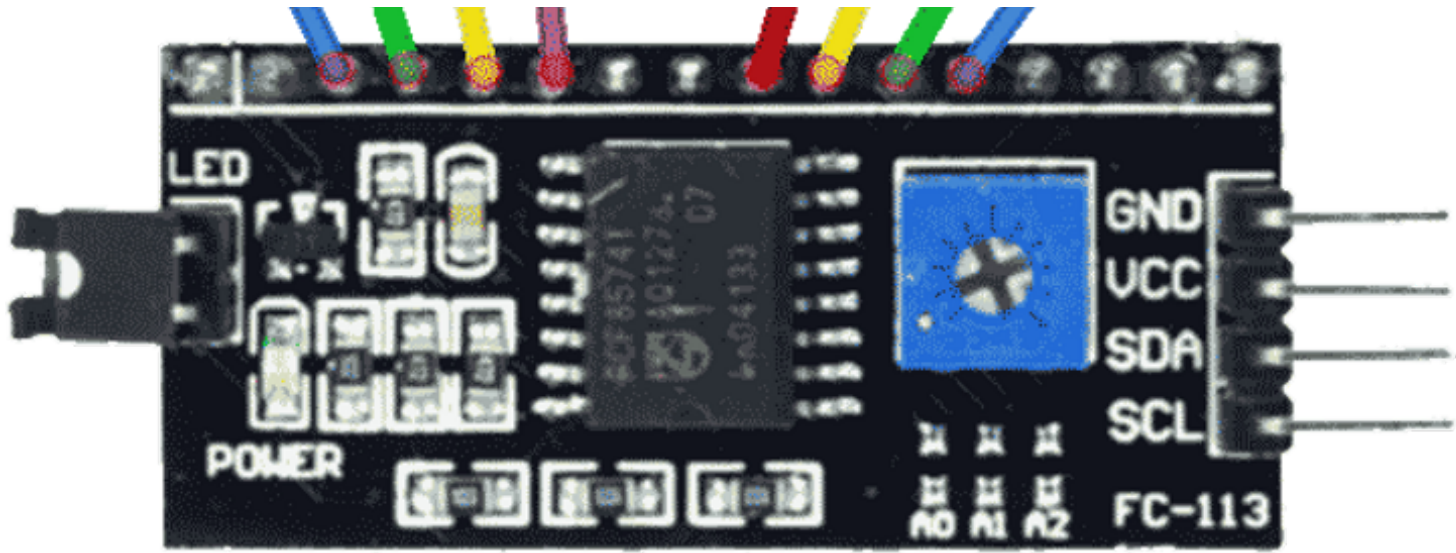
Don't know if those chinese version take bad or defect PCF8574 chips to create those cheap LCD Backpacks, maybe it's a luck of the draw.

But if your project can make due with only 7 I/O pins then you can use the LCD Backpack as an I2C Expander since those other pins are properly connected.

Here's the pinout for the I2C LCD backpack if you want to give it a try:

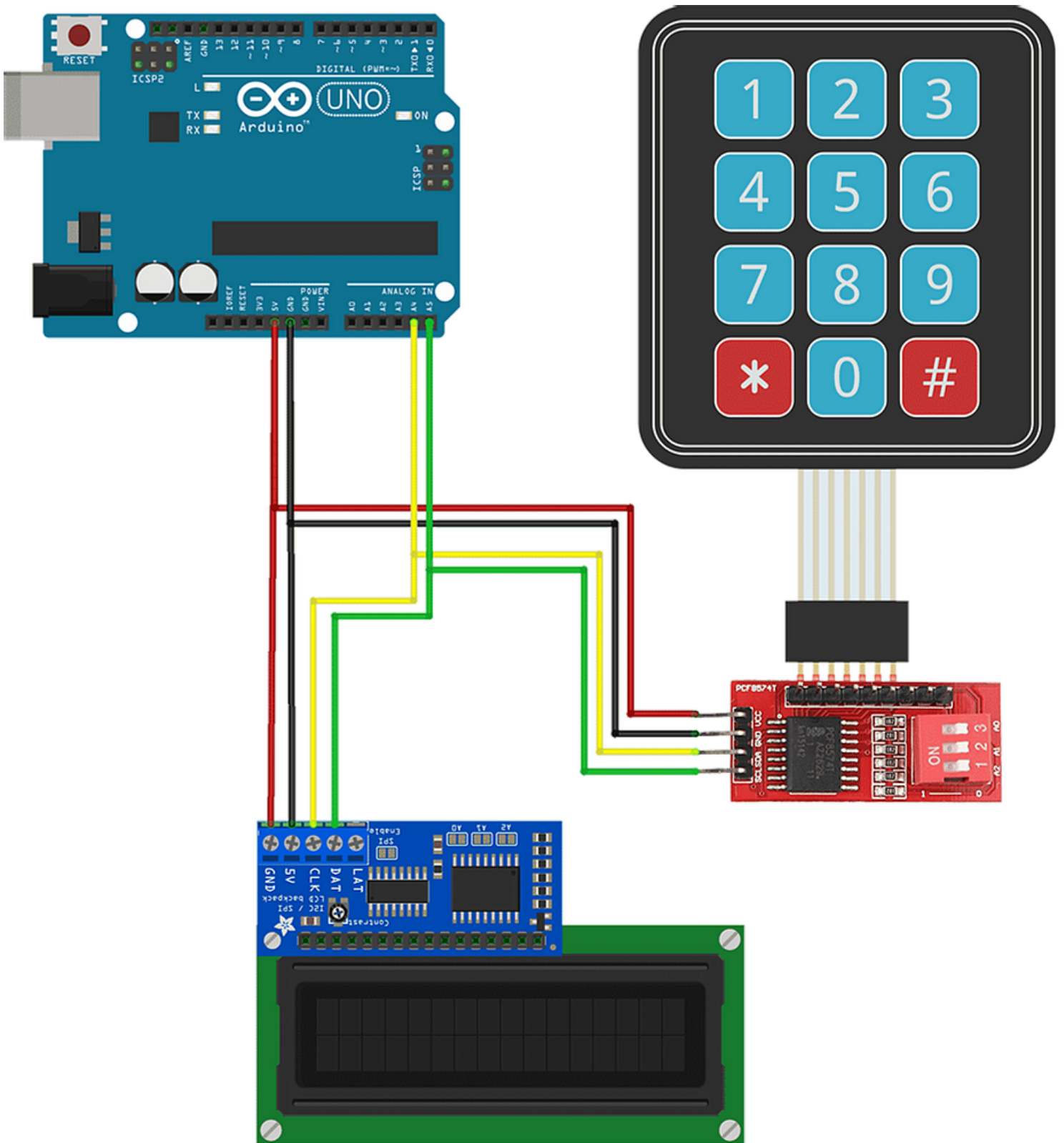


GET LEARNING!



*NOTE: P3 is not connected!

CONNECTIONS



fritzing

As you can see we are only using 2 Pins on the Arduino to control both the Keypad and the LCD:

I2C Expansion module connections:
5V and Ground from the Arduino and A4 – A5 to SDA and SCL on the I2C module

LCD Connections:
5v and Ground from the Arduino and A4 – A5 to SDA and SCL of the backpack

Keypad Connections:
The keypad has 7 pins, 4 of those from left to right are:
R1-R4 (These are the Rows of digits)

And the following 3 pins are:
C1-C3 (These are the Columns of digits)

We connect these in order (from left to right) to the I2C-Expander pins:
R1-R2-R3-R4-C1-C2-C3 —> P0-P1-P2-P3-P4-P5-P6

THE CODE

In this tutorial we will be using both the I2c LCD Backpack at address 0x27 and the I2C Expander at address 0x20.

We have a 12 keys keypad connected to the I2C Expander module P0 to P6 pins.



GET LEARNING!

Home Tutorials

[datasheetspdf.com](#)

Lea

So instead of using 7 Pins on the Arduino, were’s using the I2C protocol and using only 2 Pins to read the Keypad as well as display the results on the LCD screen.

We are using a library for both the Keypad and the LCD, you can find the link to the libraries at the bottom of this page.

As always please check out the tutorial video for more information.

```
/* Arduino I2C-Keypad using an I2C_Expander module

Created by Yvan / https://Brainy-Bits.com
This code is in the public domain...
You can: copy it, use it, modify it, share it or just plain ignore it!
Thx!

*/

#include <Keypad_I2C.h> // I2C Keypad Library by Joe Young
https://github.com/joeyoung/arduino\_keypads
#include <LiquidCrystal_I2C.h> // I2C LCD Library by Francisco
Malpartida https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home

#define lcd_addr 0x27 // I2C address of typical I2C LCD Backpack
#define keypad_addr 0x20 // I2C address of I2C Expander module (A0-A1-A2 dip switch to off position)

// LCD Pins to I2C LCD Backpack - These are default for HD44780 LCD's
#define Rs_pin 0
#define Rw_pin 1
#define En_pin 2
#define BACKLIGHT_PIN 3
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

// Create instance for LCD called: i2c_lcd
LiquidCrystal_I2C
i2c_lcd(lcd_addr,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

// Define the keypad pins
const byte ROWS = 4;
const byte COLS = 3;
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

// Keypad pins connected to the I2C-Expander pins P0-P6
byte rowPins[ROWS] = {0, 1, 2, 3}; // connect to the row pinouts of the keypad
byte colPins[COLS] = {4, 5, 6}; // connect to the column pinouts of the keypad

// Create instance of the Keypad name I2C_Keypad and using the PCF8574 chip
Keypad_I2C I2C_Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS, keypad_addr, PCF8574 );

void setup() {

  i2c_lcd.begin (16,2); // our LCD is a 16x2, change for your LCD if needed

  // LCD Backlight ON
  i2c_lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
  i2c_lcd.setBacklight(HIGH);
  i2c_lcd.clear(); // Clear the LCD screen

  I2C_Keypad.begin();

}

void loop() {
```




GET LEARNING!

Home Tutorials

```
        i2c_lcd.setCursor(0,0),
        i2c_lcd.print(key);
        break;

    case '2':
        i2c_lcd.setCursor(0,1);
        i2c_lcd.print(key);
        break;

    case '3':
        i2c_lcd.setCursor(3,0);
        i2c_lcd.print(key);
        break;

    case '4':
        i2c_lcd.setCursor(3,1);
        i2c_lcd.print(key);
        break;

    case '5':
        i2c_lcd.setCursor(6,0);
        i2c_lcd.print(key);
        break;

    case '6':
        i2c_lcd.setCursor(6,1);
        i2c_lcd.print(key);
        break;

    case '7':
        i2c_lcd.setCursor(9,0);
        i2c_lcd.print(key);
        break;

    case '8':
        i2c_lcd.setCursor(9,1);
        i2c_lcd.print(key);
        break;

    case '9':
        i2c_lcd.setCursor(12,0);
        i2c_lcd.print(key);
        break;

    case '0':
        i2c_lcd.setCursor(12,1);
        i2c_lcd.print(key);
        break;

    case '*':
        i2c_lcd.clear();
        break;

    case '#':
        i2c_lcd.setCursor(15,0);
        i2c_lcd.print(key);
        break;
    }
}
```

CONCLUSION

As you saw, using an I2C Expander is pretty easy and if your project needs a Keypad, now you won't have to sacrifice a bunch of pins to use one.

The I2C expander module can also be used as an output device to control some LEDs for example, I'll might do a tutorial on this soon.

Thank you for stopping by!

TUTORIAL VIDEO



GET LEARNING!

Home Tutorials



DOWNLOAD

Copy and Paste the above code/sketch in your Arduino IDE software.

Link to the libraries used in this tutorial:

- I2C Keypad library by Joe Young: https://github.com/joeyoung/arduino_keypads
-
- I2C LCD library by Francisco Malpartida: <https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home>

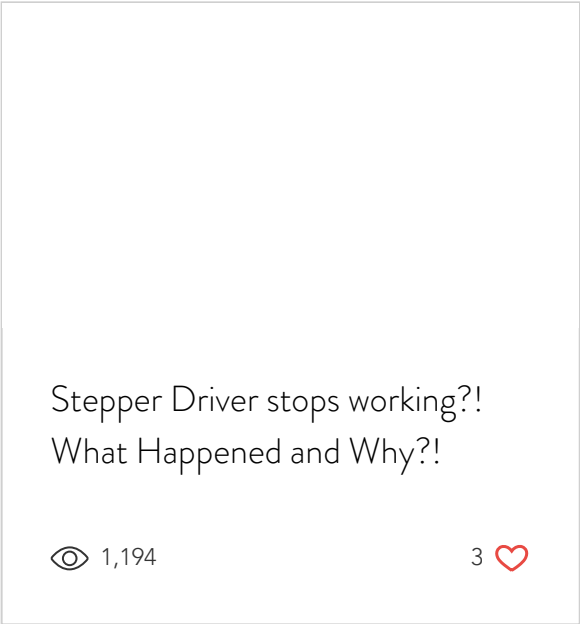
Recent Posts

See All

A Better Ultrasonic Sensor for the Arduino? JSN-SR04T

2,025

4



Stepper Driver stops working?!
What Happened and Why?!

1,194

3

Stepper Motor MicroStepping!
What to keep in mind when doin...

3,164

6

All my content is and will always be Free.

If you feel that my Videos / Tutorials are helping, and you would like to contribute...
You can toss some coins in the Tip Jar via PayPal.

Select amount then click the “Donate” button.

Brainy-Bits Tip Jar Amount

Thanks! \$2.00 CAD

DONATE

Safe & Secure

PayPal