

```
/*****
```

This is a test example for the Adafruit Trellis w/HT16K33

Designed specifically to work with the Adafruit Trellis

----> <https://www.adafruit.com/products/1616>

----> <https://www.adafruit.com/products/1611>

These displays use I2C to communicate, 2 pins are required to interface

Adafruit invests time and resources providing this open source code, please support Adafruit and open-source hardware by purchasing products from Adafruit!

Written by Limor Fried/Ladyada for Adafruit Industries.

MIT license, all text above must be included in any redistribution

```
*****/
```

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

```
#include "Adafruit_Trellis.h"
```

```
/*****
```

This example shows reading buttons and setting/clearing buttons in a loop

"momentary" mode has the LED light up only when a button is pressed

"latching" mode lets you turn the LED on/off when pressed

Up to 8 matrices can be used but this example will show 4 or 1

```
*****/
```

```
#define MOMENTARY 0
```

```
#define LATCHING 1
```

```
// set the mode here
```

```
#define MODE LATCHING
```

```
Adafruit_Trellis matrix0 = Adafruit_Trellis();
```

```
Adafruit_Trellis matrix1 = Adafruit_Trellis();
```

```
Adafruit_Trellis matrix2 = Adafruit_Trellis();
```

```
Adafruit_Trellis matrix3 = Adafruit_Trellis();
```

```
// you can add another 4, up to 8
```

```
// Just one
```

```
//Adafruit_TrellisSet trellis = Adafruit_TrellisSet(&matrix0);
```

```
// or use the below to select 4, up to 8 can be passed in
```

```
Adafruit_TrellisSet trellis = Adafruit_TrellisSet(&matrix0, &matrix1,  
&matrix2, &matrix3);
```

```

// set to however many you're working with here, up to 8
#define NUMTRELLIS 4

#define numKeys (NUMTRELLIS * 16)

// Connect Trellis Vin to 5V and Ground to ground.
// Connect the INT wire to pin #A2 (can change later!)
#define INTPIN A2
// Connect I2C SDA pin to your Arduino SDA line
// Connect I2C SCL pin to your Arduino SCL line
// All Trellises share the SDA, SCL and INT pin!
// Even 8 tiles use only 3 wires max

void setup() {
    Serial.begin(9600);
    // Serial.println("Trellis Demo");

    // INT pin requires a pullup
    pinMode(INTPIN, INPUT);
    digitalWrite(INTPIN, HIGH);

    // begin() with the addresses of each panel in order
    // I find it easiest if the addresses are in order
    trellis.begin(0x71,0x72,0x73,0x74); // only one
    // trellis.begin(0x70, 0x71, 0x72, 0x73); // or four!
    trellis.setBrightness(0);
    // light up all the LEDs in order
    for (uint8_t i=0; i<numKeys; i++) {
        trellis.setLED(i);
        trellis.writeDisplay();
        delay(50);
    }
    // then turn them off
    for (uint8_t i=0; i<numKeys; i++) {
        trellis.clrLED(i);
        trellis.writeDisplay();
        delay(50);
    }
}

void loop() {
    delay(30); // 30ms delay is required, dont remove me!
}

```

```

if (MODE == MOMENTARY) {
    // If a button was just pressed or released...
    if (trellis.readSwitches()) {
        // go through every button
        for (uint8_t i=0; i<numKeys; i++) {
// if it was pressed, turn it on
if (trellis.justPressed(i)) {
    Serial.print("v"); Serial.println(i);
    trellis.setLED(i);
}
// if it was released, turn it off
if (trellis.justReleased(i)) {
    Serial.print("^"); Serial.println(i);
    trellis.clrLED(i);
}
        }
        // tell the trellis to set the LEDs we requested
        trellis.writeDisplay();
    }
}

if (MODE == LATCHING) {
    // If a button was just pressed or released...
    if (trellis.readSwitches()) {
        // go through every button
        for (uint8_t i=0; i<numKeys; i++) {
            // if it was pressed...
if (trellis.justPressed(i)) {
    /*Serial.print("v");*/ Serial.write(i);
    // Alternate the LED
    if (trellis.isLED(i))
        trellis.clrLED(i);
    else
        trellis.setLED(i);
        }
        }
        // tell the trellis to set the LEDs we requested
        trellis.writeDisplay();
    }
}
}
}

```