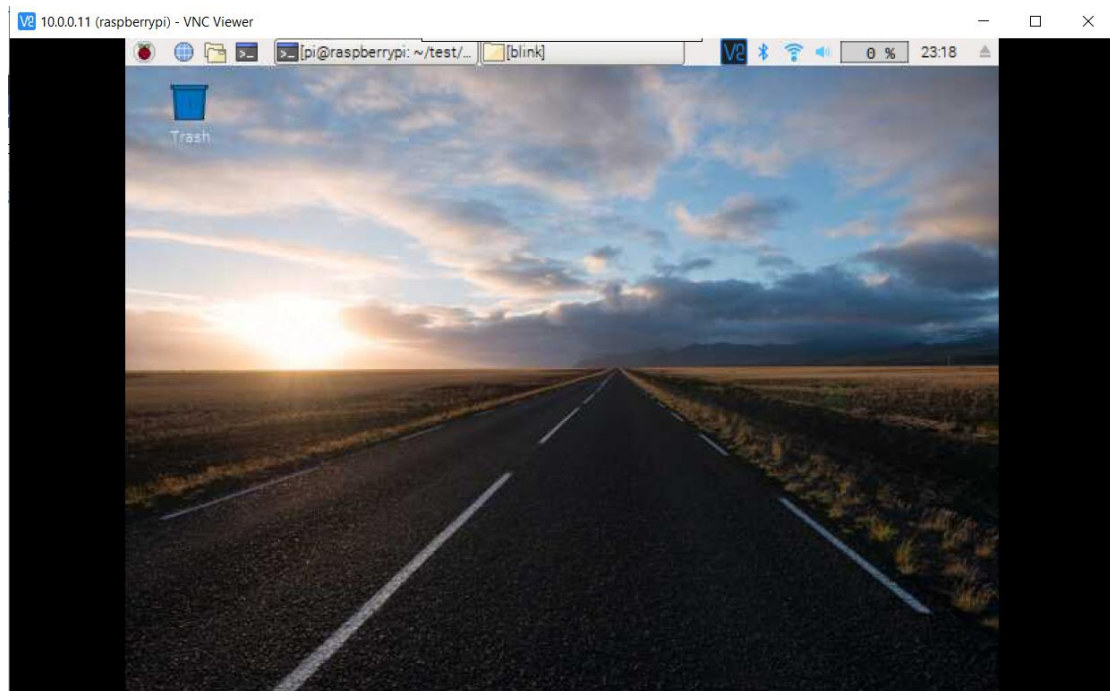


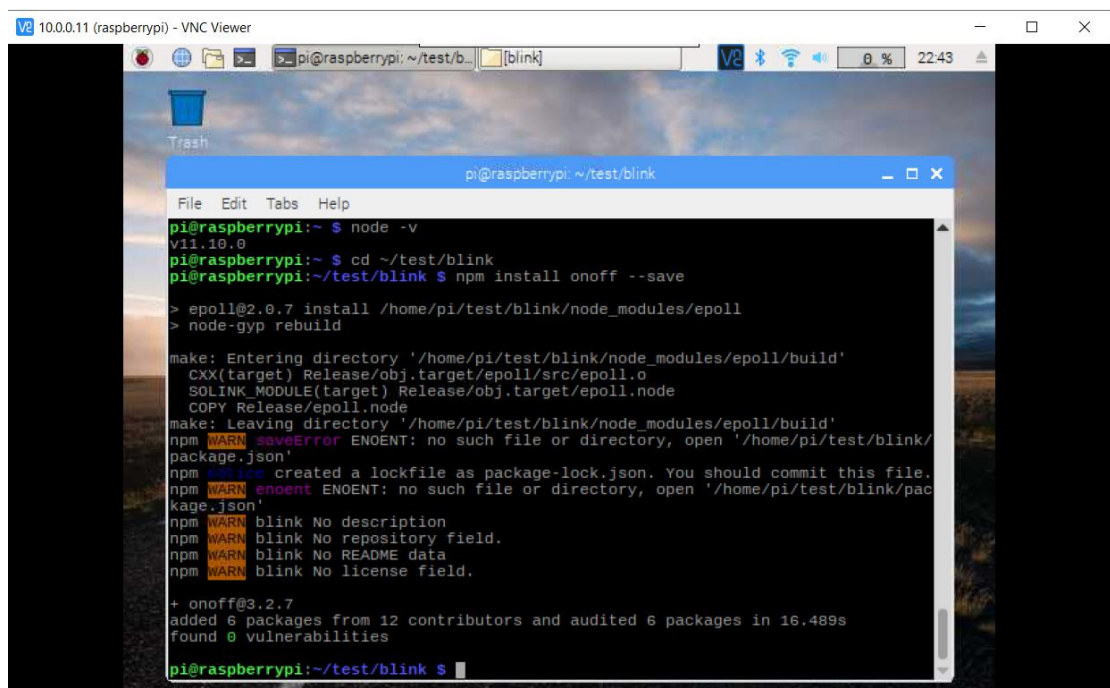
## EE517 Internet of Things

### Project: Controlling led blinking by node.js

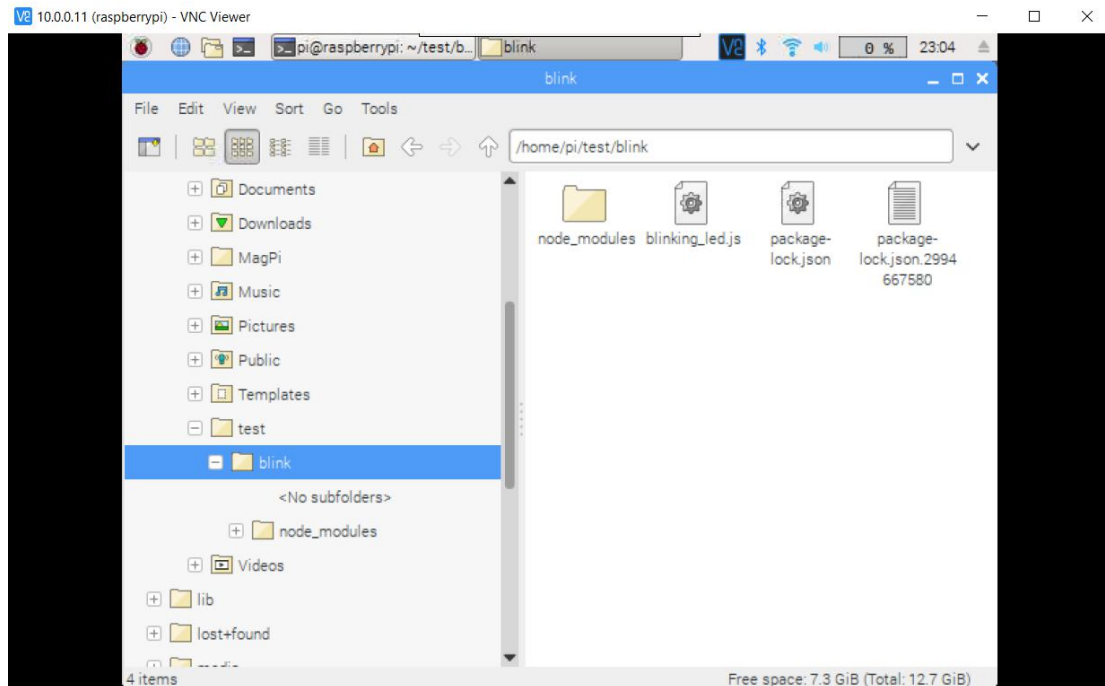
1. Access Raspberry Pi with your computer by using Putty or VNC. Here, I use VNC.



2. Install onoff packet under your direction



3. Create a blinking\_led.js file under your folder



Blinking\_led.js

```
// Import the onoff library
```

```
var onoff = require('onoff');
```

```
var Gpio = onoff.Gpio;
```

```
// Initialize GPIO 4 to be an output pin.
```

```
var led = new Gpio(4, 'out');
```

```
var interval;
```

```
    // The setInterval() method repeats a given function
```

```
    // at every given time-interval.
```

```
interval=setInterval(function () {
```

```
    // - readSync() read GPIO value synchronously. Returns the
```

```
    //   number 0 or 1 to represent the state of the GPIO.
```

```
    // - The state is reversed by (state + 1) % 2
```

```
    //       0 ==> 1
```

```
    //       1 ==> 0
```

```
    var value = (led.readSync() + 1) % 2;
```

```
    // write() asynchronously writes 0 or 1 to GPIO pin 4
```

```
    led.write(value, function() {
```

```
        console.log("Changed LED state to: " + value);
```

```
    });
```

```
}, 2000); // This interval will be called every two seconds
```

```
// Listen to the SIGINT event triggered by Ctrl-C
```

```
process.on('SIGINT', function () {
```

```
    clearInterval(interval);
```

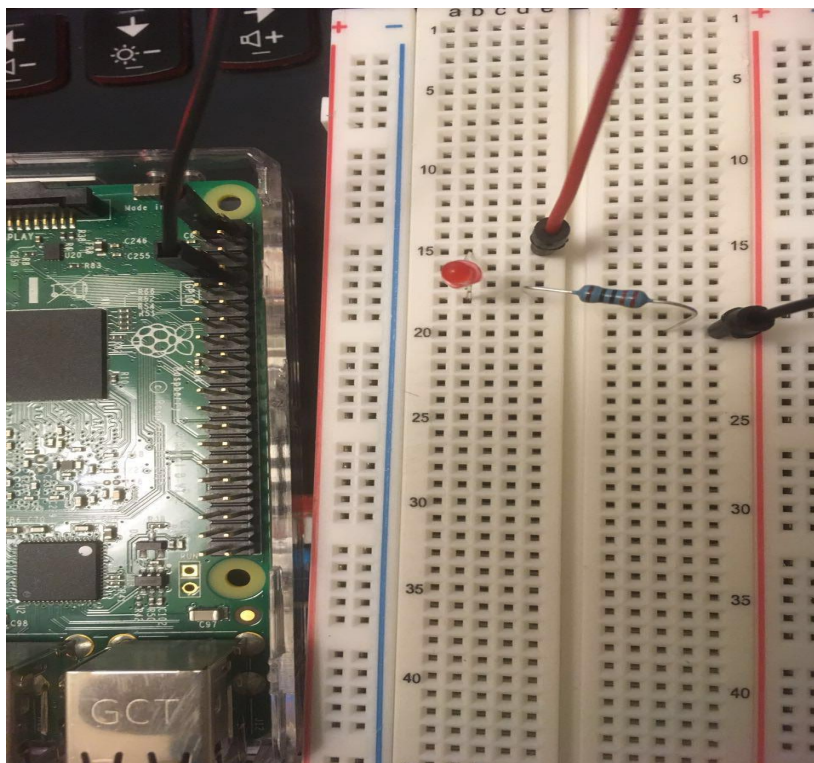
```
    // writeSync(value) write 0 or 1 to GPIO
```

```
led.writeSync(0);  
// Cleanly close the GPIO pin before existing.  
led.unexport();  
console.log('Bye, bye!');  
process.exit();  
});
```

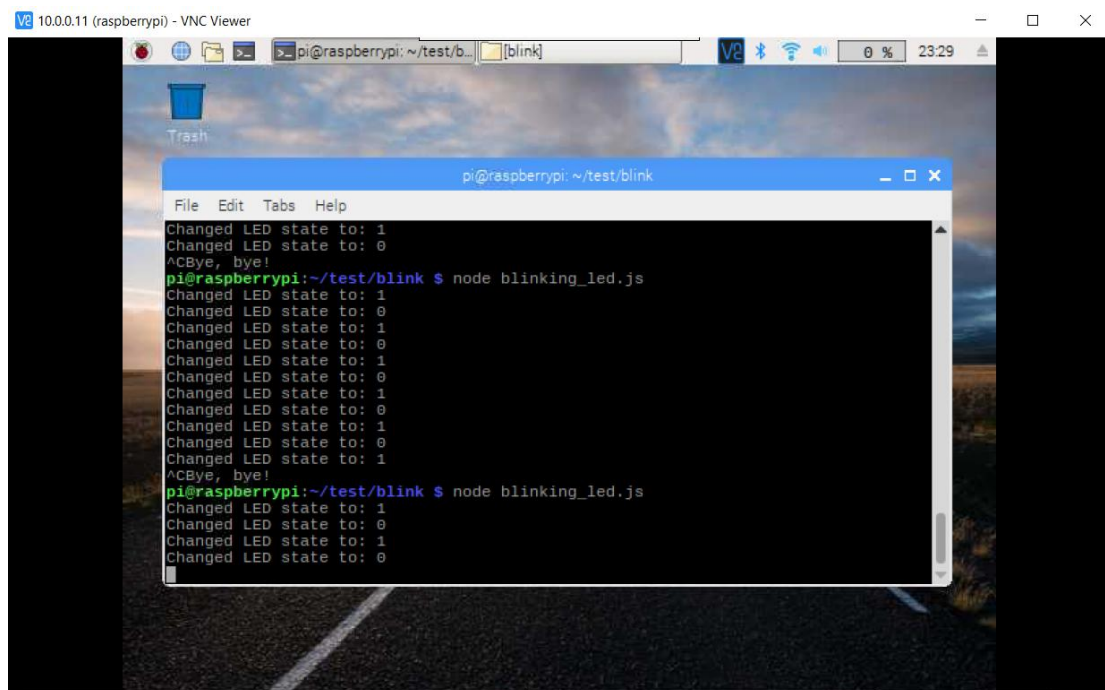
#### 4. Hardware connection

Pin 1 (3.3V, which is the red wire) connects with the anode of LED(which is the long leg)

Pin 7 (GPIO4, which is the black wire) connects with a 220 $\Omega$  resistor and the cathode of LED(which is the short leg)



## 5. Run node blinking\_led.js



```
pi@raspberrypi: ~ - VNC Viewer
pi@raspberrypi: ~/test/b... [blink]
pi@raspberrypi: ~/test/blink
File Edit Tabs Help
Changed LED state to: 1
Changed LED state to: 0
^Cbye, bye!
pi@raspberrypi:~/test/blink $ node blinking_led.js
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
^Cbye, bye!
pi@raspberrypi:~/test/blink $ node blinking_led.js
Changed LED state to: 1
Changed LED state to: 0
Changed LED state to: 1
Changed LED state to: 0
```

simulation:

When LED status is 1, the led is on; when LED status is 0, the led is off.

