



**VIT**<sup>®</sup>  
**Vellore Institute of Technology**  
(Deemed to be University under section 3 of UGC Act, 1956)

# Eye Health Aid

*IoT Domain Analyst, ECE3502*

*A Project Report*

*25<sup>th</sup> May, 2021*

## **Team Members –**

<b>Name</b>	<b>Registration no.</b>
Vikram Baruah	18BEC0967
Md Ayaz Haque	18BEC0556
Sadhinoor Jyoti	18BEC0916

*Under the guidance of -*

*Aparna Mohanty*

## **Introduction –**

Computer vision syndrome (CVS) is a condition caused by using digital displays which include laptops, mobile phones, E-readers, computer monitors for a prolonged period of time without blinking and without a break, thereby causing dry eyes which leads to headache, strain and overall fatigue.

CVS is a major cause of concern in today's society due to increased smart phone and computer usage with 5-6 hours being spent on these devices by an average teenager every day. In order to tackle this problem, an Eye Health Aid System has been developed which detects the number of blinks within a specific time interval and notifies the user via E-mail and mobile phone notification when the number of blinks is less than the threshold.

## **Abstract –**

An Eye Health Aid System has been developed using NodeMCU ESP8266, and an IR sensor which detects the number of blinks. On an average, 15 blinks per minute is considered healthy; if the number of blinks per minute is less than the threshold, an E-mail and mobile notification is sent to the user via Blynk.

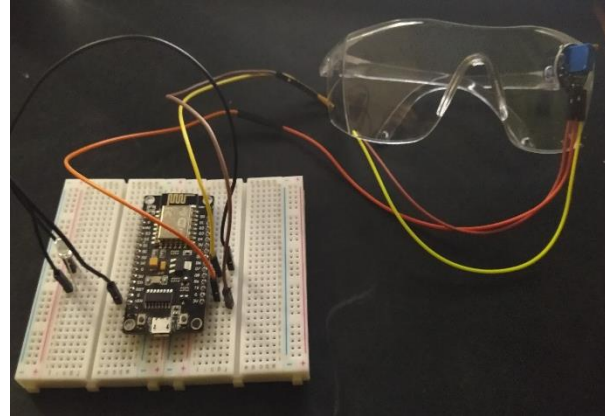
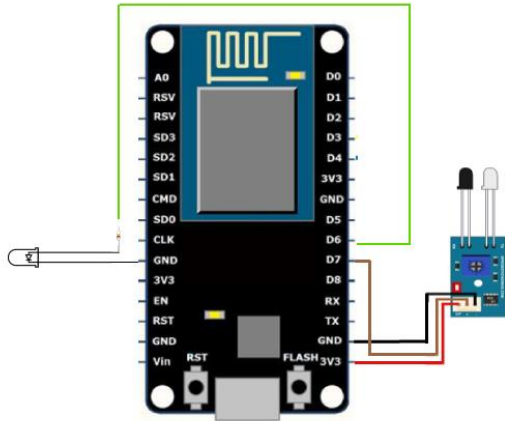
NodeMCU has been chosen as it contains a pre-attached WiFi Module (ESP8266), that is required to send the data collected by the IR sensor over to our IoT protocol, Blynk.

## **Components & Software Required –**

- NodeMCU ESP8266
- IR sensor
- LED
- Arduino IDE
- Blynk

## Proposed Methodology –

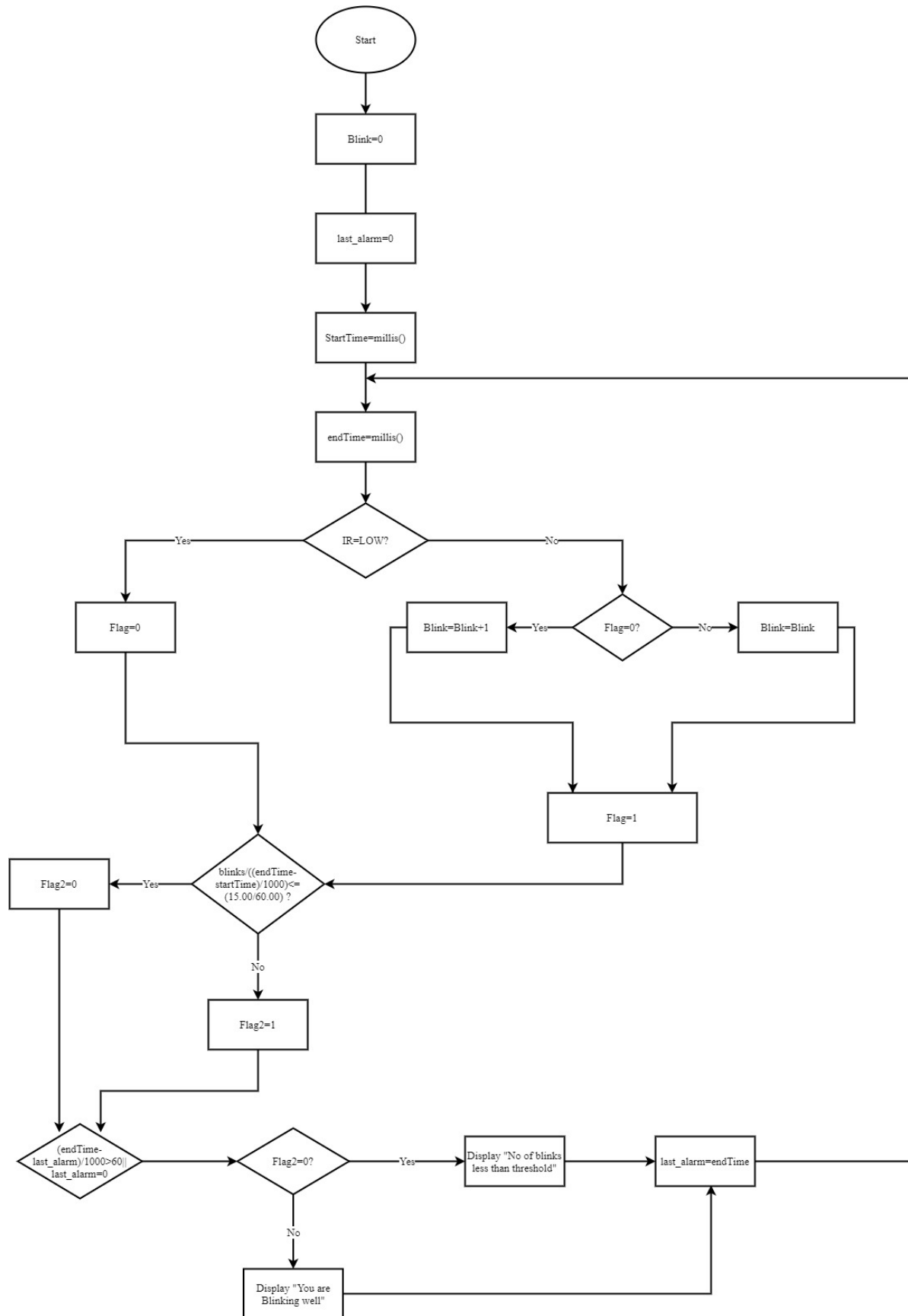
The IR sensor is connected to the D7 pin of NodeMCU and the LED is connected to the D6 pin of Node MCU (as shown in the figure).



A flag is initialized at 0; when IR reading is low, flag is set to 0 and when IR reading is high, the number of blinks is incremented by 1 if flag in the previous statement was 0 and then flag is set to 1.

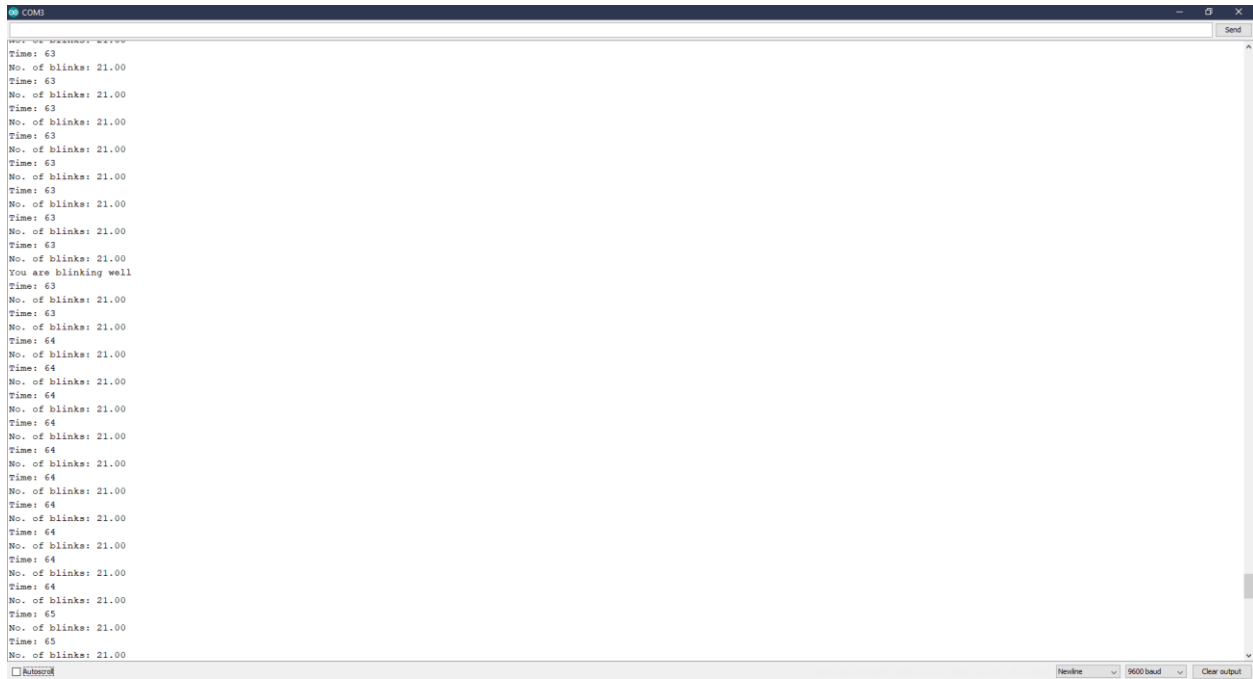
Millis() function has been used to keep track of the current time and the condition  $\text{blinks}/((\text{millis}() - \text{start Time})/1000) \leq 15.00/60.00$  is checked every minute, and if the condition is satisfied then an alarm is sent which is programmed to transmit the data over to Blynk.

Blynk sends an Email and mobile notification depending on the data received.



## Results –

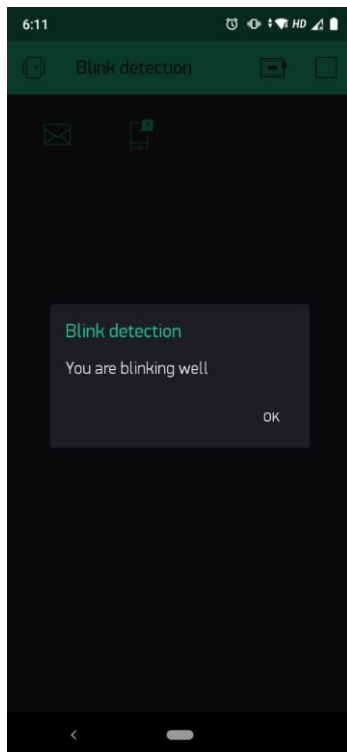
When no. of blinks  $\geq 15$  in 60 seconds



The screenshot shows a serial monitor window titled 'COM3'. The output text is as follows:

```
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
You are blinking well  
Time: 63  
No. of blinks: 21.00  
Time: 63  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 64  
No. of blinks: 21.00  
Time: 65  
No. of blinks: 21.00  
Time: 65  
No. of blinks: 21.00
```

At the bottom of the window, there is a checkbox labeled 'Autoscroll' which is checked, and two buttons labeled 'Newline' and 'Clear output'.



When no. of blinks <15 in 60 seconds

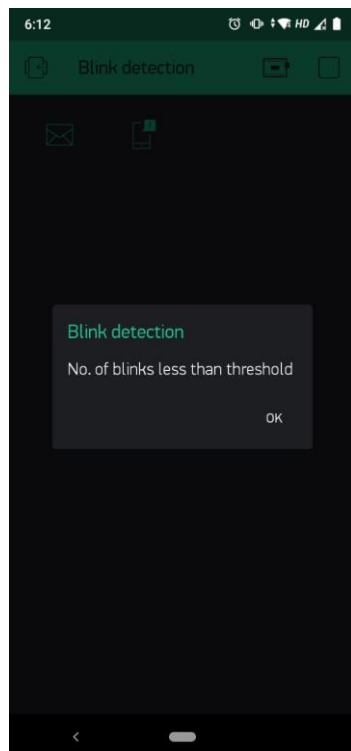
```

COM3
No. of blinks: 21.00
Time: 124
No. of blinks: 21.00
Time: 124
No. of blinks: 21.00
Time: 124
No. of blinks: 21.00
Time: 124
No. of blinks: 21.00
Time: 124
No. of blinks less than threshold
Time: 124
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 125
No. of blinks: 21.00
Time: 126
No. of blinks: 21.00
Time: 126
No. of blinks: 21.00
Time: 126
No. of blinks: 21.00
Time: 126
No. of blinks: 21.00
Time: 126

```

Autosave Newline 9600 baud Clear output

**Blynk** <dispatcher@blynk.io>  
to me ▾  
📎  
No of blinks less than threshold. Blink more



## **Conclusion –**

In this paper we have seen the system designed to reduce eye dryness caused due to blinking less while staring at electronic displays. It helps detect the number of blinks and can be checked from time to time to make the user conscious of his/her blinking pattern, thereby reducing the possibility of being subjected to eye dryness and strain.

## **References –**

[1] [www.arduino.cc](http://www.arduino.cc)

[2] Ueno H., Kanda, M. and Tsukino, M. “Development of Drowsiness Detection System”, IEEE Vehicle Navigation and Information Systems Conference Proceedings, (1994), ppA1-3,15-20.

[3] International Journal of Advance Engineering and Research Development (IJAERD) Volume 5, Issue 03, March-2018, e-ISSN: 2348 - 4470, print-ISSN: 2348-6406.

[4] [https://www.nvisioncenters.com/conditions/computer-vision-syndrome/#:~:text=Computer%20vision%20syndrome%20\(CVS\)%20is,improper%20distance%2C%20or%20before%20sleep.](https://www.nvisioncenters.com/conditions/computer-vision-syndrome/#:~:text=Computer%20vision%20syndrome%20(CVS)%20is,improper%20distance%2C%20or%20before%20sleep.)