

# **Title: *Smartwatch***

## **I. Objective**

The purpose of this project is to construct a smartwatch. The smartwatch will control other electrical appliances remotely through the internet. The smartwatch will also connect show the date, time, and temperature.

## **II. Summary**

To construct this smartwatch ESP8266 will be used. The main feature of this smartwatch is to display time, weather, connect to the internet, and control appliances. In order for the watch to receive data it must be connected to the internet. Also, the watch has to be connected to the nearest network time protocol (NTP) server to get the time and data. Lastly, in order for the watch to display the weather – the weather data must be obtained from the Yahoo Weather Application Program Interface (API). Above it all, the watch has to be connected to the Blynk server and use its bridge function to directly control the other ESP8266 Relay module.

## **III. Related to Industry**

Digital Logic Design is the representation of signals and sequences of a digital circuit through numbers. It is the basis for digital computing and provides a fundamental understanding on how circuits and hardware communicate within a computer. Digital logic is typically implanted into most electronic devices, including calculators, computers, video games, and watches. This project gives students an introduction to how to connect IoT (internet of things) devices to circuitry. This may

relate to a professional engineering job by an engineer may design smart devices. Many engineers may get in this field with the evolution of smart products, such as smart lights, smart homes, smart appliances, smart televisions, smart phones and more.

## **IV. Methodology**

### **a. Parts**

- i.** ESP-12e
- ii.** Organic Light-emitting Diode
- iii.** Three SMD Buttons
- iv.** 3.3V Voltage Regulator
- v.** TP4056 (Charger Circuit)
- vi.** Two Small SMD Buttons (buttons present on the NodeMCU)
- vii.** Two 0603 LE
- viii.** Two 0603 Resistor
- ix.** SMD Slide Switch
- x.** Eight 0805 10K Resistor
- xi.** Two 0805 10 $\mu$ F Capacitor
- xii.** 0603 112 Resistor
- xiii.** Wires
- xiv.** Breadboard
- xv.** Digital Multimeter
- xvi.** Power Supply

### **b. Procedure**

## **i. Solder the Components**

1. Design the PCB (Printed Circuit Board) using easyEDA from (<https://easyeda.com/>).
2. Order the PCB from JLCPCB (<http://www.jlpcb.com/>)
3. Solder all the components to the PCB.
4. Use a solder to solder the THT components.
5. Use a hot air soldering gun for the SMD components.
6. After Soldering the components, pin 9 of the GPIO cannot be used as an input.
7. Cut the copper trace using a Xacto knife and solder a thin wire from the button to pin 14 of the GPIO.

## **ii. Programming the Watch**

1. Get the Weather API from Yahoo's Weather API.
2. Visit the Page, under the YQL query and click the green text and scroll all the way to the end of the text until "nome, ak" is found.
3. Enter a city's name and the abbreviation of the country the city is in.
4. Click the "Test" button below and copy the link given under "Endpoint" from "/v1" till the end of the link.
5. Enter the Arduino Code.
6. In the code, first insert the Blynk Authentication code, that will be emailed after creating a new Blynk Project.
7. Type in the WIFI SSID and password.

8. Paste the copied Yahoo API link in the place where it says “const String URL =”
9. Under BLYNK\_CONNECTED, insert the authentication code of the other ESP8266, which controls the Relays.
10. Scroll Down and find “TimeChangeRule”
11. From there, add or subtract minutes from UTC according to your time zone.
12. Install all of the Arduino libraries and the latest Blynk library.
13. Wire pin 4 of the serial communicator to the USP-TTL converter accordingly
14. ESP-12EUSB-TTL convertor
15. GND-GND
16. RST – RT
17. RC – TXD
18. TX – RXD
19. On the Arduino IDE, select your device as NodeMCU 1.0 and choose the correct COM port. Then, upload the code.
20. While uploading the code make sure to press and hold the flash button on the watch.

### **iii. Strap for the Watch**

1. To make the watch strap, use an electric band from a pair of old safety goggles and attach it to the watch.
2. Place the battery in between the strap.

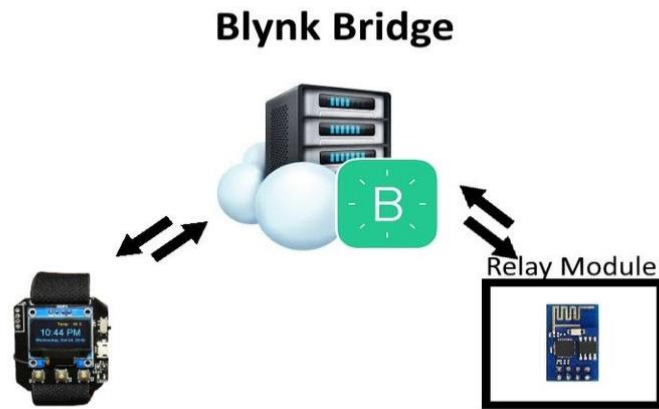
#### iv. Test the Watch

1. Connect the watch to Wi-Fi
2. Ensure that the date, time, and weather appear.
3. Connect the relay to any electrical appliance, which can be controlled by the watch.
4. Make sure to connect the Relay switch to the live wire and not the neutral.

#### c. Links

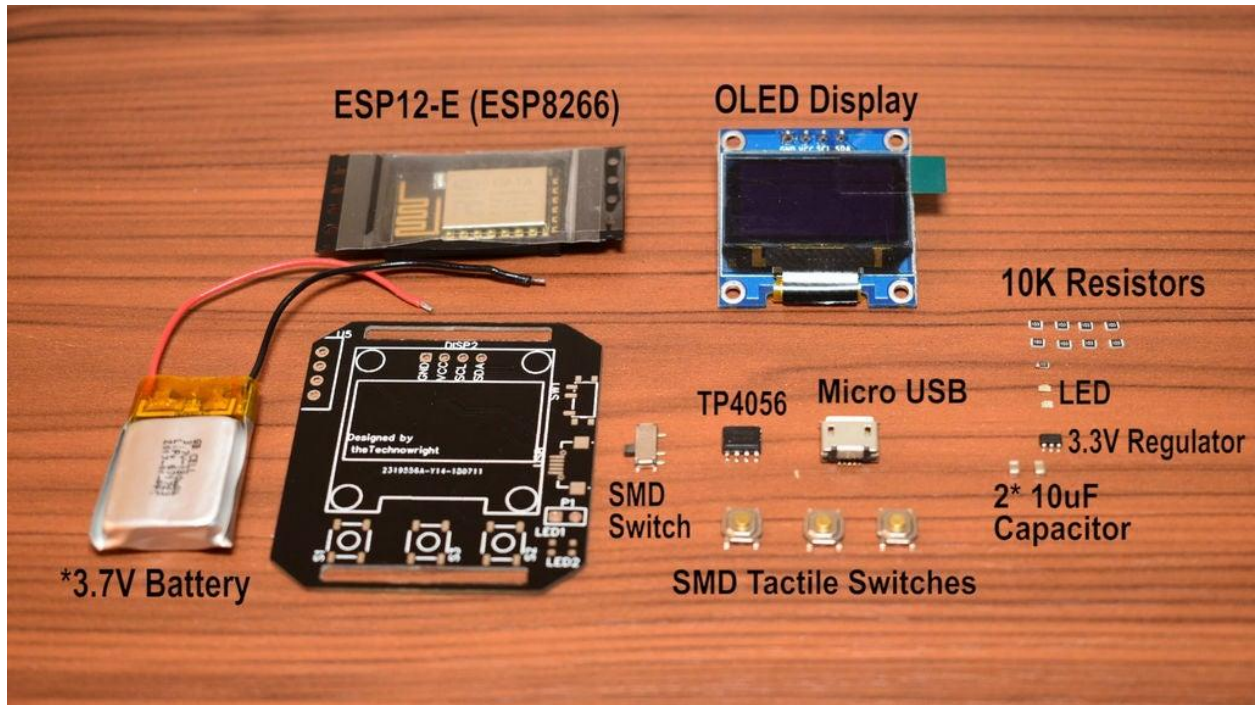
- i. **Code:** [https://github.com/theTechnowright/DIY-Smartwatch-using-ESP8266/blob/master/ESP8266\\_\\_SmartWatch.ino](https://github.com/theTechnowright/DIY-Smartwatch-using-ESP8266/blob/master/ESP8266__SmartWatch.ino)
- ii. **Video:** <https://www.youtube.com/watch?v=BjgFCOM3iMk>
- iii. **Video:**  
<https://www.youtube.com/watch?v=kPblG4l4gEs&feature=youtu.be>
- iv. **Reference:** <https://www.instructables.com/id/DIY-Smartwatch-Using-ESP8266/>
- v. **Arduino Codes:**
  1. Time.h & TimeLib.h: [www.github.com/PaulStoffregen/Time](http://www.github.com/PaulStoffregen/Time)
  2. Timezone.h: [www.github.com/JChristensen/Timezone](http://www.github.com/JChristensen/Timezone)
  3. SSD1306.H & SSD1306WIRE.H:  
[www.github.com/JChristensen/Timezone](http://www.github.com/JChristensen/Timezone)
  4. NTPClient.h: [www.github.com/JChristensen/Timezone](http://www.github.com/JChristensen/Timezone)
  5. ESP8266WIFI.H & WIFIUDP.H:  
[www.github.com/JChristensen/Timezone](http://www.github.com/JChristensen/Timezone)

d. Pictures

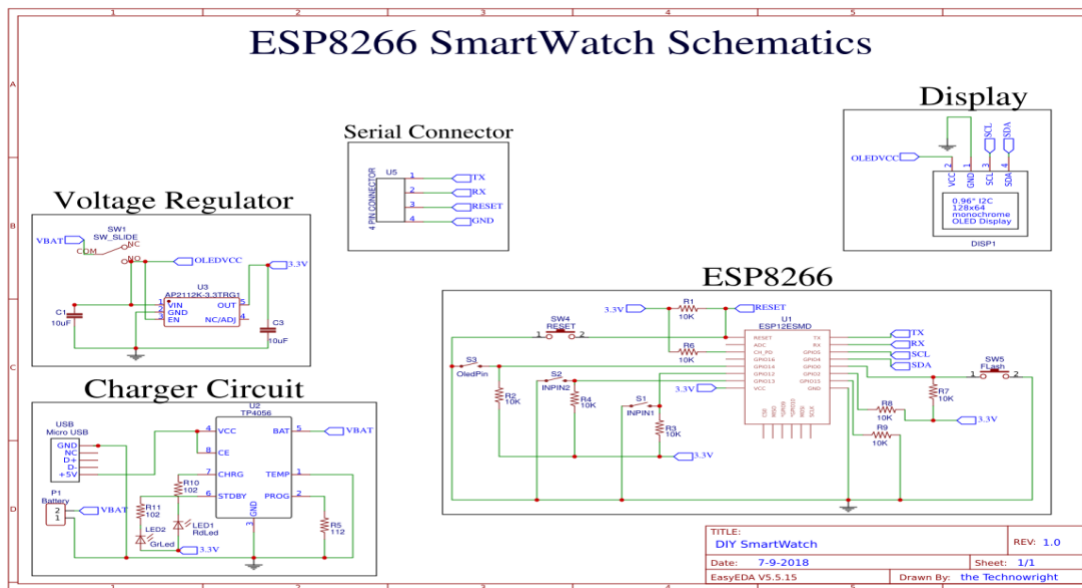


**Figure 1** – The Figure above shows the function of the Blynk Bridge to the relay module and smartwatch





**Figure 2** – This photo contains a picture of all the parts needed to build the smartwatch.



**Figure 3** – Circuit diagram for the watch