A

MINI PORJECT REPORT

ON

“SMART SOCKET”

FOR FULFILLMENT
OF THE REQUIREMENTS FOR THE
MINI PROJECT SUBJECT
OF T.E. E&TC – 2019 COURSE, SPPU, PUNE

By

Disha Chinchole	**T190053046**
Chinmay Deshmukh	**T190053054**
Aditi Funde	**T190053075**

GUIDE
Prof. J. M. Sandur

DEPARTMENT OF
ELECTRONICS AND TELECOMMUNICATION ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE – 43

ACADEMIC YEAR: 2022 - 2023

Department of Electronics and Telecommunication Engineering

Pune Institute of Computer Technology, Pune – 43

CERTIFICATE

This is to certify that the Mini Project report entitled

SMART SOCKET

has been successfully completed by

Disha Chinchole	T190053046
Chinmay Deshmukh	T190053054
Aditi Funde	T190053075

Is a bonafide work carried out by them under the supervision of Prof. J. M. Sandur and it is approved for the partial fulfillment of the requirements for the Mini Project subject of T.E. E&TC – 2019 Course of the Savitribai Phule Pune University, Pune.

Prof. J. M. Sandur	Prof. M. V. Munot
Project Guide Dept. of E&TC	HOD, E&TC

Place: Pune
Date :

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who has helped us in the development of Smart Socket. Firstly, we would like to thank our guide, Prof. Jyoti Sandur ma’am, for their guidance and support throughout the project. Their valuable insights and feedback have been instrumental in shaping the project's direction and ensuring its success.

We would also like to thank the assistant teachers who have assisted us throughout this project. We would also like to acknowledge the support and encouragement our peers and family members provided. Their motivation and assistance have helped us stay focused throughout this project.

Lastly, we would like to express our gratitude to all the resources that were used during the development of this project. The knowledge gained during the project has been invaluable and will undoubtedly assist us in future endeavors.

Disha Chinchole
Chinmay Deshmukh
Aditi Funde

ABSTRACT

The intelligent control system has developed rapidly with the development of Internet of Things technology. The Home Wi-Fi smart socket consists of three parts: hardware platform, server, and client device. We can realize control of home equipment and real-time monitoring of appliances from the terminal via Wi-Fi. A smart socket is a device that connects to a home's Wi-Fi network and allows users to control their electrical devices remotely using a web server. It provides convenience and energy efficiency by enabling users to remotely turn on and off the appliances at specific times.

Disha Chinchole

Chinmay Deshmukh

Aditi Funde

List of figures:

Sr. No	Figure number and Figure caption	Page no.
1	Block Schematic of System	5
2	Implementation on Bread Board	12
3	Testing of Power Supply	12
4	Simulation of Circuit Diagram	13
5	Simulation of Smart Socket	13
6	PCB Design	13
7	Final Project Photograph and Working	14

List of tables:

Sr. No	Table number and Table caption	Page no.
1	ESP Module Selection Table	7
2	List of Components Required	9
3	Bill of Materials	17

Abbreviations

Symbol	**Meaning**
PCB	Printed Circuit Board
IoT	Internet Of Things
GPIO	General Purpose Input Output
TCP/IP	Transmission Control Protocol
Wi-Fi	Wireless Fidelity
MCU	Microcontroller Unit

CONTENTS

	Abstract		i
	List of Figures		ii
	List of Tables		iii
	Abbreviations		iv
	Feasibility Report		v
1	Introduction		
	1.1	Back ground and Context	1
	1.2	Relevance	1
	1.3	Literature Survey	2
	1.4	Motivation	2
	1.5	Aim of the Project	3
	1.6	Technical Approach	3
2	Block Diagram		
	2.1	Theoretical background	5
	2.2	Block diagram	5
	2.3	Requirements	6
	2.4	Selection of Sensors and Major Components	7
3	System Design		
	3.1	Calculations Block1	8
	3.2	List of Components	9
4	Implementation, Testing and Debugging		
	4.1	Implementation on breadboard	12
	4.2	Testing, Debugging	12
	4.3	Simulation Results, Observation table	13
	4.4	PCB Design	13
5	Results and discussion		15
	Conclusions and Future Scope		15
	References		16
	Bill of Material		17
	Datasheets		

Feasibility report

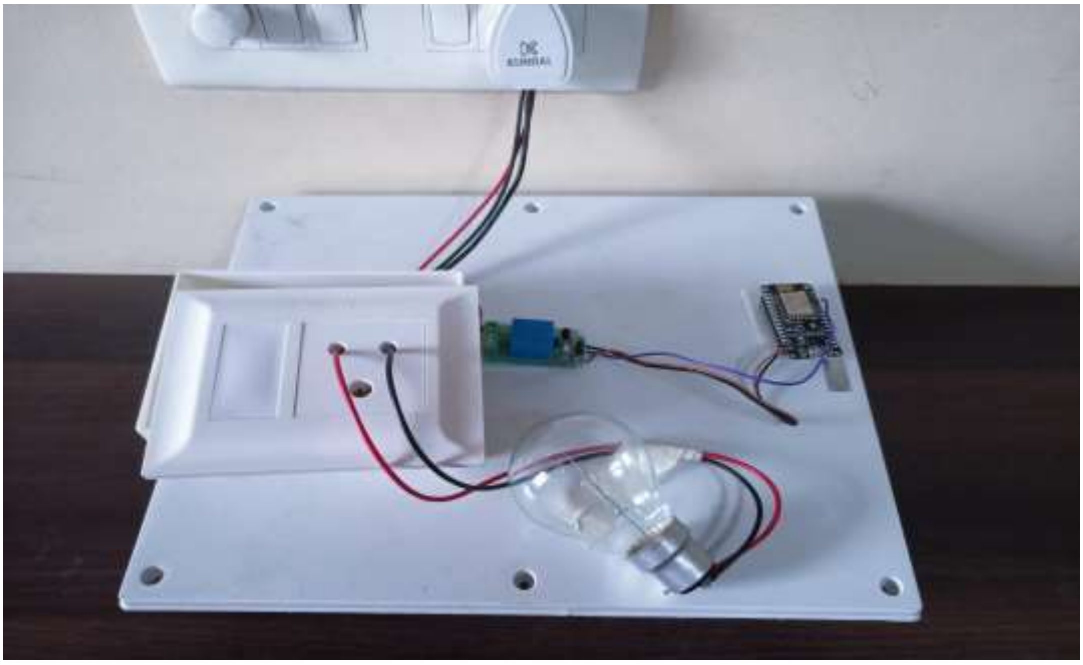
Title: Smart Socket

Group members:

1)32404_Aditi Funde
2)32412_Chinmay Deshmukh
3)32414_Disha Chinchole

Tools required	Testing possibility	Controller	Cost
Hardware/components: 1.5V Power Supply 2.Relay Circuit 3.AC Socket 4.Bulb 5.ESP Module	Hardware : Yes	ESP8266 Module (The ESP8266 module enables microcontrollers to connect to 2.4 GHz Wi-Fi. It can be used with ESP-AT firmware to provide Wi-Fi connectivity to external host MCUs, or it can be used as a self-sufficient MCU by running an RTOS-based SDK)	1) 350/- 2) 200/- 3) 120/- 4) 220/- 5) 180/- Total Cost: 1070/-
Software Proteus Arduino IDE	Software Yes	………	
Tools available within campus or outside	Sensors required	Signal conditioning if any	
Outside	No	Yes	
Applications if any	PCB design and fabrication	Datasheets/ application notes available	
Remote Control Power Saving Automatic Access to Power Devices	Yes	Yes	
Mechanical design	Enclosure design	Demonstration	
No	Yes	Yes	

Title of project: SMART SOCKET



Electric specification

Input specifications: 3.3V

Output specifications: 3.3V,12mA per pin

Features if any:
Global Remote Access
24/7 Monitoring
Power Supply Control

Mechanical specification

Enclosure design, any special mechanical arrangement:
Yes

It plugs into a regular wall outlet and controls the flow of electricity to connected devices. Once set up, you can control a smart plug from a companion app on your phone or tablet or from a smart speaker or display.
Cost of system**:**
Total: 1070/-

Group members:

T190053046	**Disha Chinchole**
T190053054	**Chinmay Deshmukh**
T190053075	**Aditi Funde**

CHAPTER 1: Introduction

1.1 Background

The background for a smart socket project is the increasing demand for home automation and the Internet of Things (IoT) devices. Smart sockets are an essential component of a smart home, allowing users to control the power supply to their appliances and devices remotely, through a web server. Smart sockets provide many benefits, such as energy savings, convenience, and enhanced security. With a smart socket, users can remotely turn off lights, TVs, and other appliances when not in use, which can help reduce their energy consumption and lower their electricity bills. Additionally, smart sockets can be programmed to turn on and off at specific times, creating the illusion of an occupied home and deterring potential burglars.

The technology behind smart sockets involves integrating Wi-Fi connectivity, sensors, and power management capabilities into a single device. This requires expertise in hardware design, firmware development, and software development for the web server. Smart socket projects typically involve collaboration between electrical engineers, software developers, and designers to create an intuitive and user-friendly product that meets the needs of consumers.

1.2 Relevance

A report on smart sockets would be relevant for a variety of audiences, including consumers, home automation enthusiasts, technology companies, and investors. For consumers, a report on smart sockets would provide valuable information on the features and benefits of these devices, such as energy savings, convenience, and enhanced security. The report could also provide comparisons between different types of smart sockets available on the market, helping consumers make informed purchasing decisions. For home automation enthusiasts, a report on smart sockets could provide technical details on the design and operation of these devices, including the hardware and software components, and how they integrate with other smart home systems.

For technology companies, a report on smart sockets could provide market insights on the demand for these devices, as well as information on the competitive landscape, including key players and emerging trends. This information could help companies develop more effective marketing strategies and product development plans. For investors, a report on smart sockets could provide insights into the growth potential and financial performance of companies operating in this space. The report could also highlight key risks and opportunities for investors to consider.

1.3 Literature Survey

A literature survey for a smart socket project report could include the following:

Smart Home Technologies: A Review of Concepts and Technologies - This paper provides an overview of the concepts and technologies used in smart homes, including smart sockets, and discusses the benefits and challenges of these technologies.

A Review of IoT-Based Home Automation Systems[1] - This paper provides a review of IoT-based home automation systems, including smart sockets, and discusses the current state of the art, challenges, and opportunities in this field.

Design and Implementation of a Smart Socket for Home Energy Management[2] - This paper describes the design and implementation of a smart socket for home energy management, including the hardware and software components, and presents results from a field trial.

Analysis of Energy Consumption in Smart Homes Using Smart Plugs [3]- This paper analyzes the energy consumption patterns of smart homes using smart plugs, including smart sockets, and discusses the potential for energy savings through the use of these devices.

1.4 Motivation

The motivation for a smart socket can be based on several factors, including the increasing demand for home automation and the Internet of Things (IoT) devices, the potential benefits of energy savings, convenience, and security, and the need for innovative and user-friendly solutions in this growing market.

One key motivation for a smart socket is the growing interest in smart home technology. With the rise of IoT devices, more and more consumers are looking for ways to control their homes remotely and automate daily tasks. Smart sockets are an essential component of a smart home, providing users with greater control over their energy usage, appliances, and devices. A project report on smart sockets can help to meet this growing demand and provide valuable insights into the design, operation, and benefits of these devices.

Another motivation for a project is the potential benefits of energy savings, convenience, and security. Smart sockets can help to reduce energy consumption and lower electricity bills by allowing users to remotely turn off appliances and devices when not in use. Additionally, smart sockets can be programmed to turn on and off at specific times, creating the illusion of an occupied home and deterring potential burglars. By highlighting these benefits, a smart socket project report can help to increase awareness and adoption of these devices. Finally, a smart socket can be motivated by the need for innovative and user-friendly solutions in this growing market.

1.5 Aim and Objectives

The aim of a smart sockets is to provide a comprehensive overview of these devices, including their design, operation, benefits, and challenges, and to highlight their potential for energy savings, convenience, and security.
Also, to describe the design and operation of smart sockets, including how they can be controlled through web server, voice assistants, or home automation systems, and how they can be programmed to turn on and off at specific times.

Objective

To provide a comprehensive overview of the design and operation of smart sockets, including their hardware and software components, communication protocols, and integration with other smart home systems.

To highlight the benefits of smart sockets, including their potential for energy savings, convenience, and security, and to provide examples of how they can be used in real-world scenarios.

To identify the challenges and limitations of smart sockets, including compatibility issues with different devices and systems, security risks, and concerns about data privacy.

1.6 Technical Approach

Research and analyze existing literature on smart sockets, including technical specifications, design considerations, and user feedback. Identify the key components and technologies required for a smart socket, including microcontrollers, sensors, wireless communication modules, and power management circuits. Document the design and development process, including schematics, circuit diagrams, and code snippets, and provide a detailed explanation of how the smart socket works. Analyze the energy consumption and savings potential of the smart socket, and compare its performance to that of traditional sockets and other smart home devices.

CHAPTER 2: Block Schematic and Requirements

2.1 Introduction

The block schematic is an essential component in the design process of a smart socket. It provides an overview of the various components and how they are interconnected to create a functional device. The block schematic is a visual representation of the electrical circuitry of the smart socket, which helps designers and engineers to understand the functionality of the device and its various subsystems. Before embarking on the design and development of a smart socket, it is important to identify the requirements and specifications that the device must meet. These requirements can include technical specifications, such as voltage and current ratings, as well as functional requirements, such as the ability to connect to other smart home systems and mobile devices.

In this report, we will provide an overview of the block schematic for a smart socket, and discuss the key requirements that the device must meet. We will also discuss the challenges and opportunities associated with designing and developing a smart socket that meets these requirements, and provide recommendations for companies and individuals looking to enter the smart home market.

2.2 Block Diagram

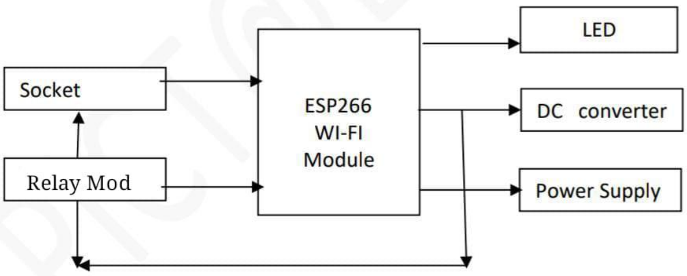


Fig.1. Block schematic of system

Socket: A socket is a physical device that provides electrical power to an appliance or electronic device. A smart socket goes beyond the traditional electrical socket and allows users to control the power supply remotely through a wireless connection.

5V Relay Module: A 5V relay is a device that allows low-voltage digital signals to control high-voltage electrical circuits. In a smart socket project, a 5V relay can be used to control the power supply to an electrical outlet or socket.

ESP266 WI-FI Module: The ESP Wi-Fi module can be used as a communication gateway between the smart socket and other devices, such as smartphones, tablets, or computers. It can also be used to connect the smart socket to the internet, enabling remote control of the socket from anywhere in the world.

LED: An LED can be used as an indicator to show the status of the socket or to provide feedback to the user.

5V DC Converter: In a smart socket project, a 5V DC converter can be used to convert higher voltage levels to a stable 5V DC voltage that can be used to power the microcontroller and other low-power components in the project.

Power Supply: In a smart socket project, a power supply can be used to provide the required voltage and current levels to power the microcontroller, the Wi-Fi module, the relay, and any other components in the project. The power supply can be designed to accept a wide range of input voltages, such as 100-240V AC, which is the typical range of mains power in most countries.

2.3 Requirements

Hardware Requirements:

AC power socket
Microcontroller board (e.g. Arduino, ESP32)
Wi-Fi module (e.g. ESP8266, ESP32)
5V DC converter
5V relay module
LED(s)
Power supply
Jumper wires
Breadboard or prototyping board
Capacitors.

Software Requirements:

Arduino IDE or other programming software

ESP8266 or ESP32 board support package (if using these microcontrollers)

Wi-Fi library (e.g. ESP8266WiFi, WiFi.h)

Relay control library (e.g. Arduino Relay, ESP8266Relay)

Web server library (optional, for controlling the socket via a web interface)

2.4 Selection of Sensors and major components

ESP8266

List Parameter	**Design Requirement**	**Unit**
I_C	80	mA
V_{CEO}	3.3	V
Temp	-40 to -125	.℃
Freq	2.4	GHz
Tx power	20	dBm
Rx power	91	dBm

Table 1: ESP Module selection table

CHAPTER 3: System Design

3.1 Calculations of Block

V output=12 Vds

Iout =1 amp

% Load regulation = 0.3%

Ripple Factor=0.4

<u>Step 1: Calculation of Voltage</u>

Input to regulatiom=14v

Vdc=14V

$V_m = \pi V_{dc} / 2 = 21.99$

Vm= 21.99-1.4 V

= 20.59

$V_{rms} = V_m/\sqrt{2}$ =14.56V

If Vrms=15	If Vrms=18
Vm=21.21 V	Vm=25.46
Vdc=13.50 V	Vdc=16.205V

<u>STEP 2: Filter Capacitor</u>

62=20 log 10{V ripple(in)/V ripple (out)}

V ripple(in)= 21.21 V

3.1 = log 10[21.21/V ripple(out)]

1258.925= 21.21/V ripple(out)

V ripple(out)= 16.847 mv

Therefore,

12Vdc →16.847 mv ripple

0.14% ripples

C=Idc /2f*V ripple

V ripple=21.21 V

C=1/2*100*21.21

C=235.7 uf

Xc =i/2πfc=1/π (100)(100uf)

Xc=15.9= 16 ohm

For 1000 uf→Vc=1.59

Capacitor Voltage=1.6 * 21.21

=33.9 v

= 35 V

3.2 List of Components

List of all the components selected:

Sr. No	Name and description of part	Value / model number
1	A compact ESP WI-FI Module - is required which can be used as a station or access point.	ESP8266
2	Power Supply	5V Regulated Supply
3	Relay - circuit that uses a low level triggered control signal.	5V Relay Module
4	Bulb - that ensures the working of components eventually ensures the output of the circuit.	25 Watt
5	AC Socket and plug along with connecting wires.	----

Table 2: List of components required in project.

Testing of Block

A] Test of ESP module

1. Connect the ESP8266 module to power source :The ESP8266 module typically requires a 3.3V-3.6V power supply .We can use voltage regulator to convert higher voltage to 3.3V or we can use 3.3 v power adapter.
2. Connect the ESP8266 module to a computer :To program the ESP8266 module and communicate with it, we need to connect ESP to computer via USB-to serial convertor
3. Install the programming environment :The ESP8266 can be programmed using Arduino IDE .Installed the appropriate development environment and set up the necessary drivers and libraries for ESP8266.
4. Write and upload a test program :Create a simple test program that blinks an LED or connect to wifi network to WI-FI network ,and upload it to ESP8266 using programming environment.
5. Monitor the output :Use the serial monitor to monitor the output from ESP8266.Verify that the ESP8266 is executing the test program correctly or not.
6. Test the WI-FI connectivity: The ESP8266's primary feature is it's ability to connect to wifi network. Test this functionality by connecting the esp8266 to a WI-Fi network and verify that it can send and receive data over the network.

B] Test of relay module

1. Connect the relay module to a power source: Depending on the specific model of the relay module, it may require a specific voltage and current to operate properly. Check the datasheet or manufacturer's instructions for the recommended power supply specifications.

2. Connect the relay module to a device: Connect a device such as an LED or motor to the relay module. This device should be connected to the relay output pins, which will switch on or off when the relay is activated.

3. Connect a control signal: Connect a control signal to the relay module. This signal can be a simple push button, a microcontroller output, or any other source that can provide a signal to activate the relay.

4. Test the relay operation: Activate the control signal and verify that the relay switches on and off as expected. You should be able to see the device.

5. Test the relay rating: Make sure that the relay is capable of handling the voltage and current of the device you are using. Check the relay datasheet for the maximum voltage and current rating, and make sure that the device does not exceed these ratings

CHAPTER 4: Implementation, testing and debugging

4.1 Implementation on bread board

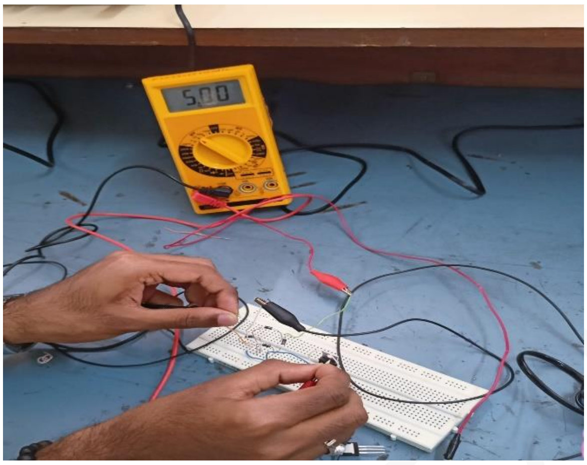


Fig 2. Implementation on bread board

4.2 Testing, Debugging:

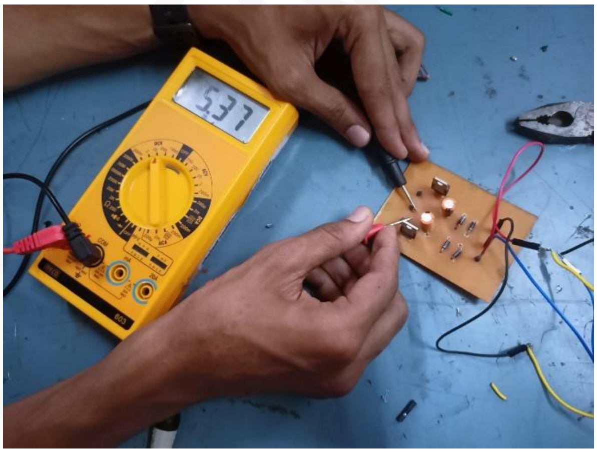


Fig 3. Testing of Power Supply

4.3 Simulation results:

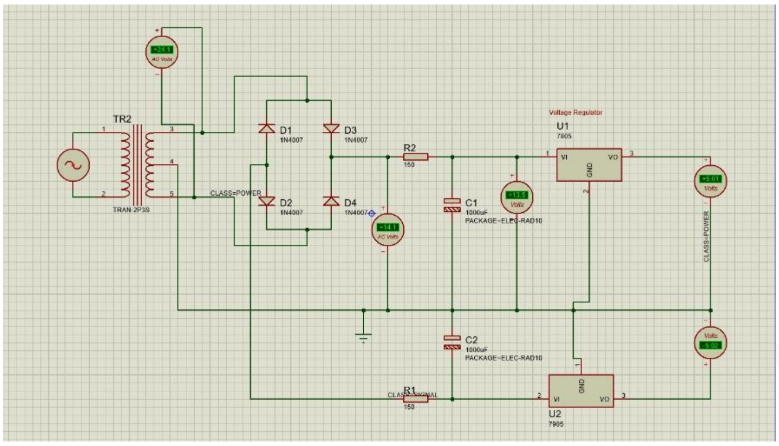


Fig 4. Simulation of Circuit Diagram

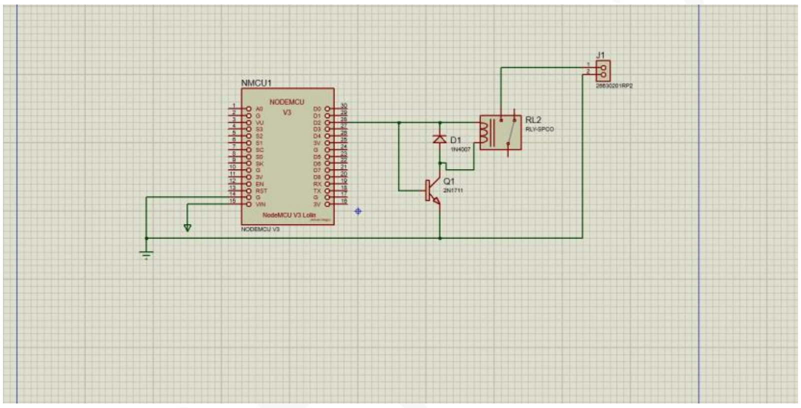


Fig 5.Schematic of Smart Socket

4.4 PCB design

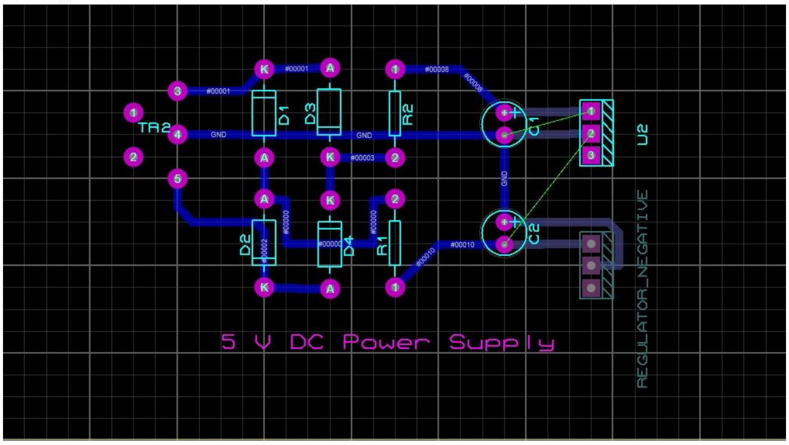


Fig.6 PCB Design

4.5 Final project photograph and working

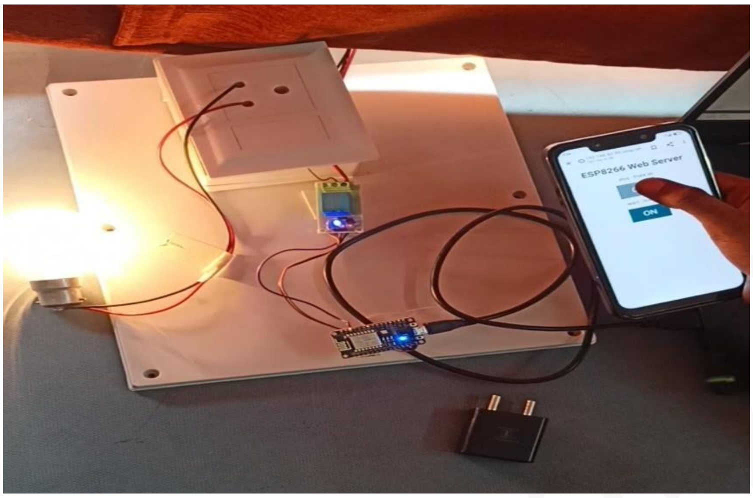


Fig 7. Final Project Photograph and Working

CHAPTER 5: Results and Discussion

A smart socket project can be a fun and rewarding project for electronics and IoT enthusiasts. It requires knowledge of electronics, programming, and networking, and it involves integrating hardware and software components to achieve the desired functionality. Additionally, the project can be customized to meet specific needs and preferences, and it has the potential to improve energy efficiency and safety in homes and cities.

A well-designed and functional smart socket project can provide convenience and energy savings, and customization options for users.

5.1.Conclusions

A smart socket project can be a useful and practical application of IoT technology, allowing users to control their electrical appliances remotely using a web server or other interface. The project typically involves using a microcontroller, a Wi-Fi module, a relay, and other components to build a device that can be plugged into a standard AC power socket and controlled using a web server or other interface.

A smart socket project can provide a practical and convenient way to control electrical appliances and improve energy efficiency, while also demonstrating the potential of IoT technology to improve our daily lives.

5.2. Future Scope

Smart Socket has the potential to play a significant role in the development of smart homes and smart cities. Here are some potential future developments and applications for the smart socket project.

- Integration with other smart home devices: Smart sockets can be integrated with other smart home devices, such as smart thermostats and lighting systems, to create a complete smart home automation system that can be controlled remotely via web server or other interface.
- Energy monitoring: Smart sockets can be equipped with energy monitoring features, allowing users to monitor and analyze their energy usage and make adjustments to improve energy efficiency.

- Integration with renewable energy sources: Smart sockets can be integrated with renewable energy sources, such as solar panels, to enable the efficient and sustainable use of renewable energy.
- Smart city applications: Smart sockets can be used in smart city applications, such as street lighting and public transportation systems, to improve energy efficiency and reduce environmental impact.

5.3.References

[1] “A Review of IoT-Based Home Automation Systems” by Radhika Garg and Swati Gupta, International Journal of Engineering Research & Technology (IJERT),2020.

[2] “Design and Development of Smart Socket Using Internet of Things” by R. H. Goudar and S.S.Patil, International Journal of Engineering Research and Technology, Vol. 6, No. 6, pp. 147-151, 2017.

[3] “A Smart Home Energy Management System using Smart Plugs” by Progress Mtshali, Freedom Khubia ICT and Society Research Group, Conference on Information Communications Technology and Society (ICTAS),2019.

[4] “Literature Review of IoT based Home Automation System” by Sivapriyan R, K Manisha Rao and Harijyothi M, International Conference on Inventive Systems and Control (ICISC 2020).

[5] “Smart Socket using ESP 8266” by Mr. Purushothaman B. , Mr. Prem S N. Ms. Aarthi T. , Ms.Gayathri R . and Mrs.Pushpalatha. N. IJSRED ISSN : 2581-7175.

Website:

[1] https://ieeexplore.ieee.org/document/8805944

[2] https://ieeexplore.ieee.org/document/9112377

5.3. Bill of Material

Components	Specification	Total Cost
Power supply	5V	350/-
ESP Module	ESP8266	180/-
Relay Circuit	5V	200/-
AC Socket	--	120/-
Bulb	100W	220/-

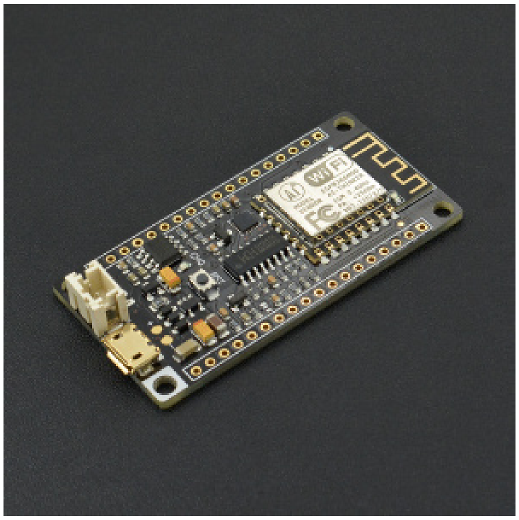
Table 3: Bill of Materials

FireBeetle ESP8266 IOT Microcontroller SKU: DFR0489

(http://www.dfrobot.com/) **Home** (https://www.dfrobot.com/) > **Arduino** (https://www.dfrobot.com/index.php?route=product/category&path=35) > **Microcontroller** (https://www.dfrobot.com/index.php?route=product/category&path=35_104)

Contents

1 Introduction
2 Specification
3 Board Overview
4 Tutorial
4.1 Requirements
4.2 Setup Arduino IDE Development Environment
4.3 Sample Code - **Blink**
4.4 Sample Code - **Scan WiFi**
5 FAQ
6 More Documents



(https://www.dfrobot.com/product-1634.html)

FireBeetle ESP8266 IOT Microcontroller (https://www.dfrobot.com/product-1634.html)

Introduction

DFRobot FireBeetle is a series of low-power-consumption development hardware designed for Internet of Things (IoT).
Firebeetle ESP8266 is a development board integrated with IoT WiFi, TCP/IP, 32-bit MCU, 10-bit ADC and multiple interfaces such as HSPI, UART, PWM, I2C and I2S. In DTIM10, the full power consumption to maintain WiFi connection reached to 1.2mW. Equipped with 16MB outer SPI flash memory, ESP8266 is available for programs and firmware storage.
Compatible with Arduino programming enables Firebeetle ESP8266 to lower the barrier of programming. Operator can implement Arduino programming codes directly onto ESP8266 to reduce the difficulty of operating and increase the stability of board.

⚠ (/wiki/index.php/File:Warning_yellow.png)

NOTE: There still remains some bugs to be detected and fixed by developers. In some cases some peripherals may not work perfectly by embedding Arduino sample codes in ESP8266. Much more functions implemented in Arduino are still under development and improvement. An alternative way is to change development tool such as RTOS and MicroPython towards a more operating-friendly experience in some conditions.

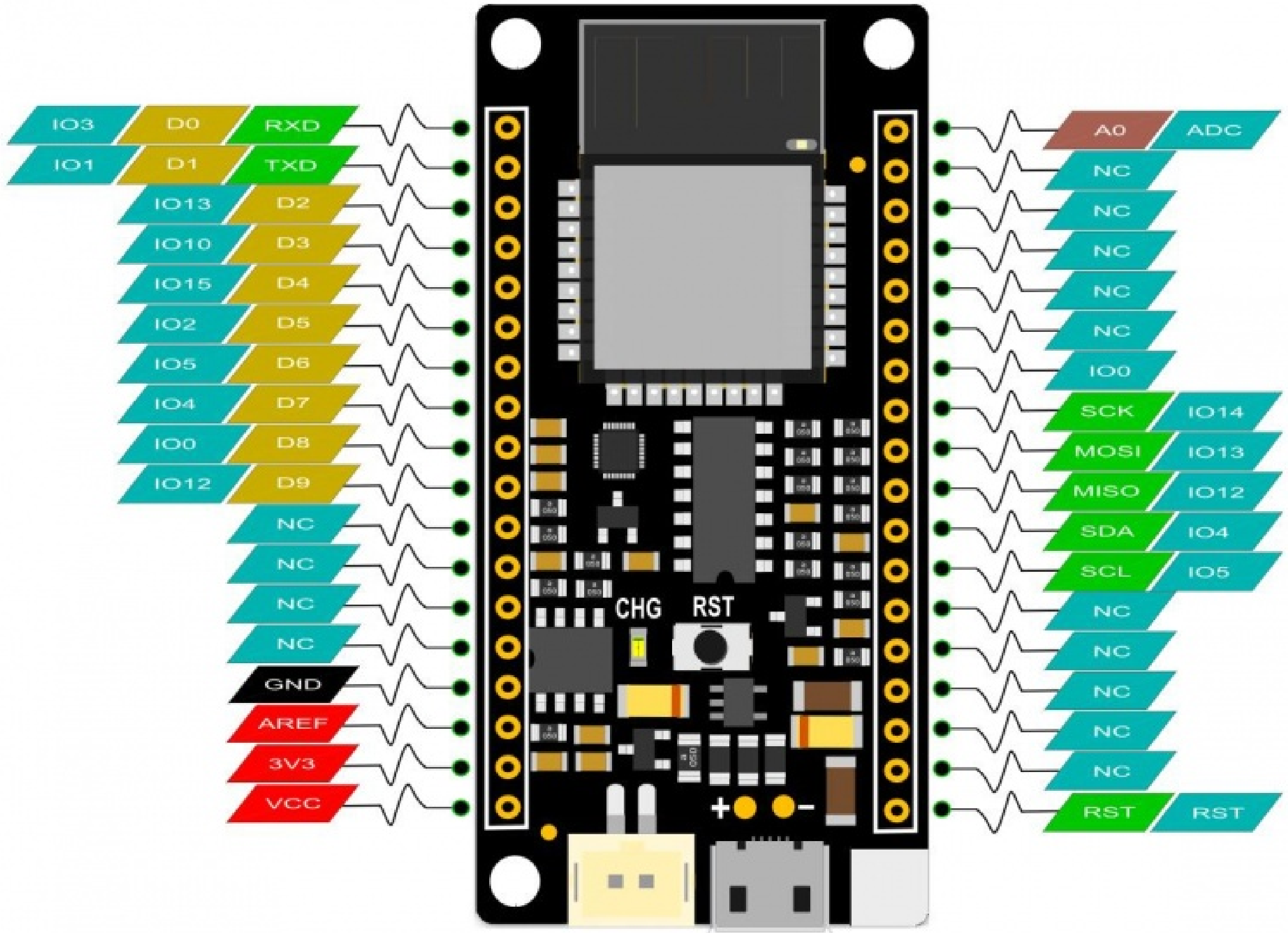
Specification

- Operating Voltage: 3.3V
- Input Voltage (limits): 3.3~5V (Lithium Battery:3.7V & USB:5V)
- Microcontroller: Tensilica L106 (32-bit MCU)
- Clock Speed: 80MHz (Maximum: 160MHz)
- SRAM：50KB
- External Flash Memory: 16MB

- DC Current in the Low-Power-Consumption: 46uA
- Average Operating Current: 80mA
- Maximum Discharging Current: 600mA (LDO-3.3 Output)
- Maximum Charging Current: 500mA
- Digital Pin x10
- Analog Pin x1
- SPI interface x1
- I2C interface x1
- IR interface x1
- I2S interface x1
- Interface: XH2.54mm Pin (No soldering default)
- In Combination of Wi-Fi MAC/ BB/RF/PA/LNA
- WiFi: IEEE802.11 b/g/n (2.4 GHz~2.5 GHz), not support 5GHz WiFi
- Operating Temperature: -10℃~+55℃
- Dimension: 58 × 29(mm)
- Weight: 24g

Board Overview

FireBeetle Board - ESP8266 is not only compatible with ESP8266 PinMap, but also make a special compatible with Arduino IDE PinMap. Dx (x=0,1,2,3...9)



(/wiki/index.php/File:DFR0489_pinout.jpg)

Fig1: FireBeetle Board-ESP8266 PinOUT

- **CHG** Blink = not connect battery; Light on = Charging; Light off = Charge complete

⚠ (/wiki/index.php/File:Warning_yellow.png) Note: NC = Empty; VCC = VCC (5V under USB power supply, Around 3.7V under 3.7V lipo battery.power supply)

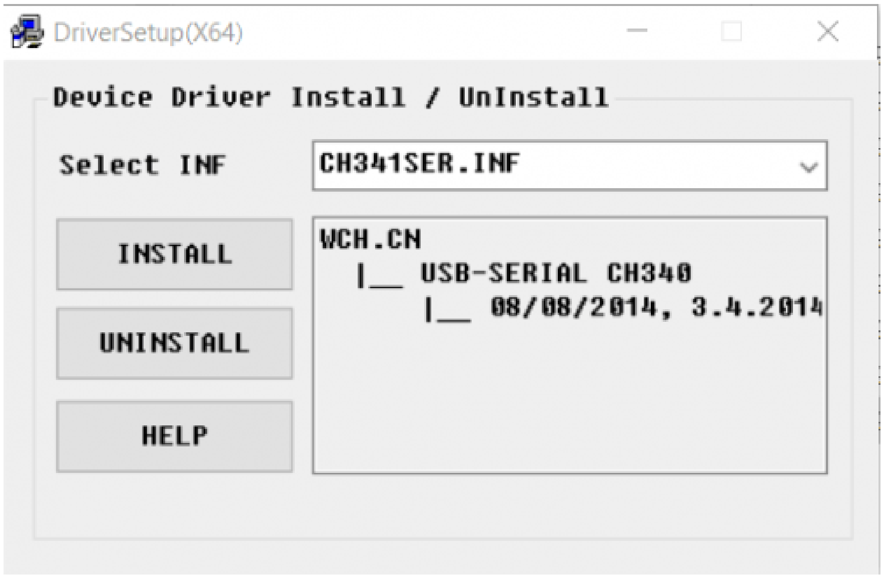
Tutorial

In this tutorial, we'll show you some basic operation with FireBeetle-ESP8266 Microcontroller.

Requirements

- **Hardware**
  - FireBeetle ESP8266 IOT Microcontroller x 1
  - Micro USB Cable x1
- **Software**
  - Arduino IDE (newest), Click to Download Arduino IDE from Arduino® (https://www.arduino.cc/en/Main/Software%7C)
  - Download CH340 FireBeetle ESP8266 Window Driver (https://github.com/Arduinolibrary/DFRobot_FireBeetle_ESP8266_DFR0489/raw/master/CH340%20Driver.zip)

Note: CH340 driver is free to install for most of Windows OS, if you find there is no COM Port in Device Manager, please download the driver and install it.



DriverSetup(X64)

Device Driver Install / UnInstall

Select INF: CH341SER.INF

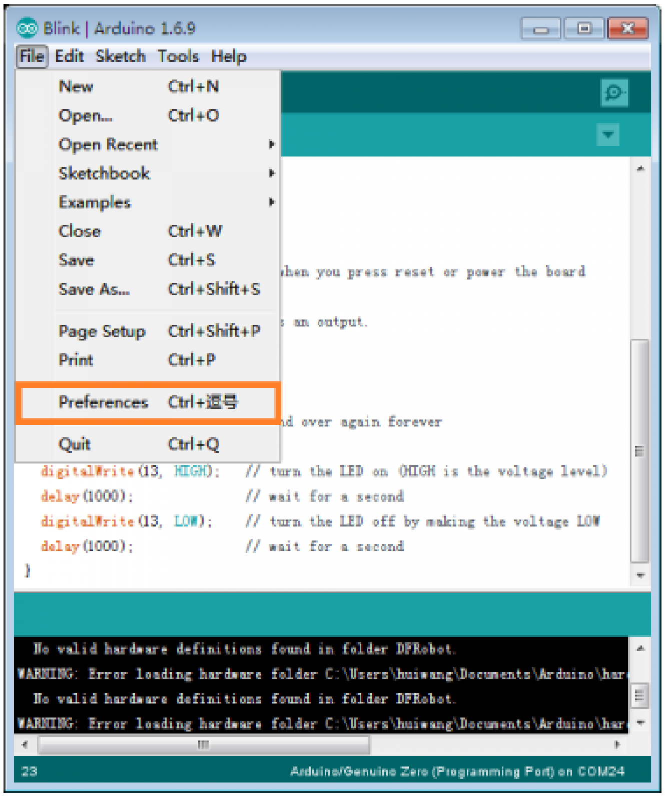
```
WCH.CN
  |__ USB-SERIAL CH340
       |__ 08/08/2014, 3.4.2014
```

(/wiki/index.php/File:CH340_Driver_install.png)

Setup Arduino IDE Development Environment

- Plug FireBeetle to your computer, install the driver manually.
- Add FireBeetle Board URL to Arduino IDE
- Open Arduino IDE, **File->Preferences**, find **Additional Boards Manager URLs**, copy the below link, and paste in the blank.

https://raw.githubusercontent.com/DFRobot/FireBeetle-ESP8266/master/package_firebeetle8266_index.json (https://raw.githubusercontent.com/DFRobot/FireBeetle-ESP8266/master/package_firebeetle8266_index.json)



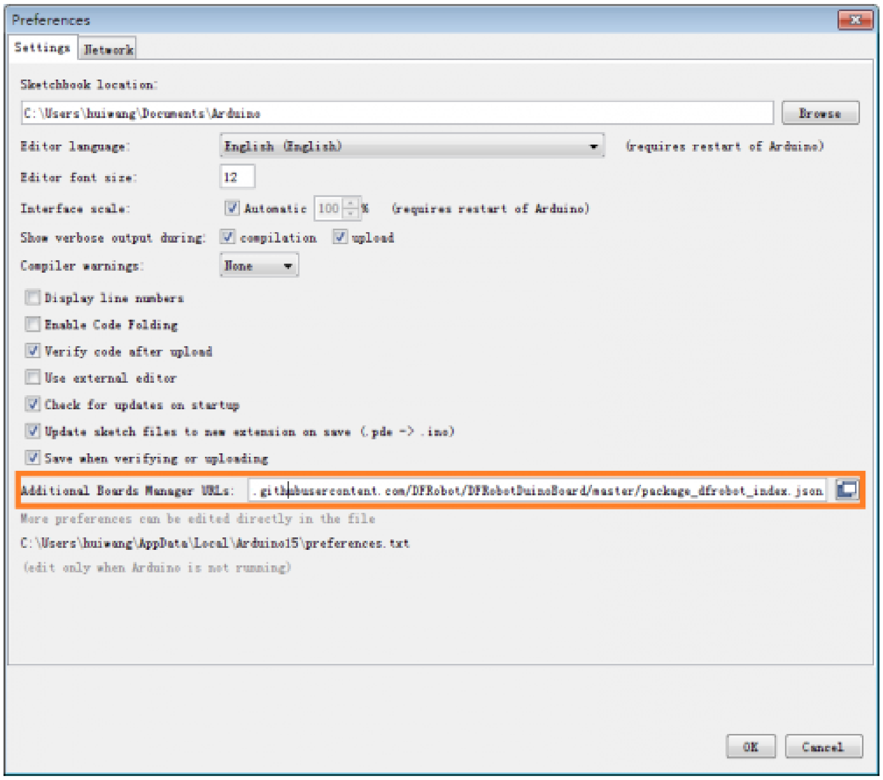
```
hen you press reset or power the board
 an output.
nd over again forever
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

```
 No valid hardware definitions found in folder DFRobot.
WARNING: Error loading hardware folder C:\Users\huiwang\Documents\Arduino\har
 No valid hardware definitions found in folder DFRobot.
WARNING: Error loading hardware folder C:\Users\huiwang\Documents\Arduino\har
```

23 Arduino/Genuino Zero (Programming Port) on COM24

(/wiki/index.php/File:Filepreferences.png)

File->Preferences



Sketchbook location:
C:\Users\huiwang\Documents\Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

☑ Verify code after upload

☐ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: .githubusercontent.com/DFRobot/DFRobotDuinoBoard/master/package_dfrobot_index.json

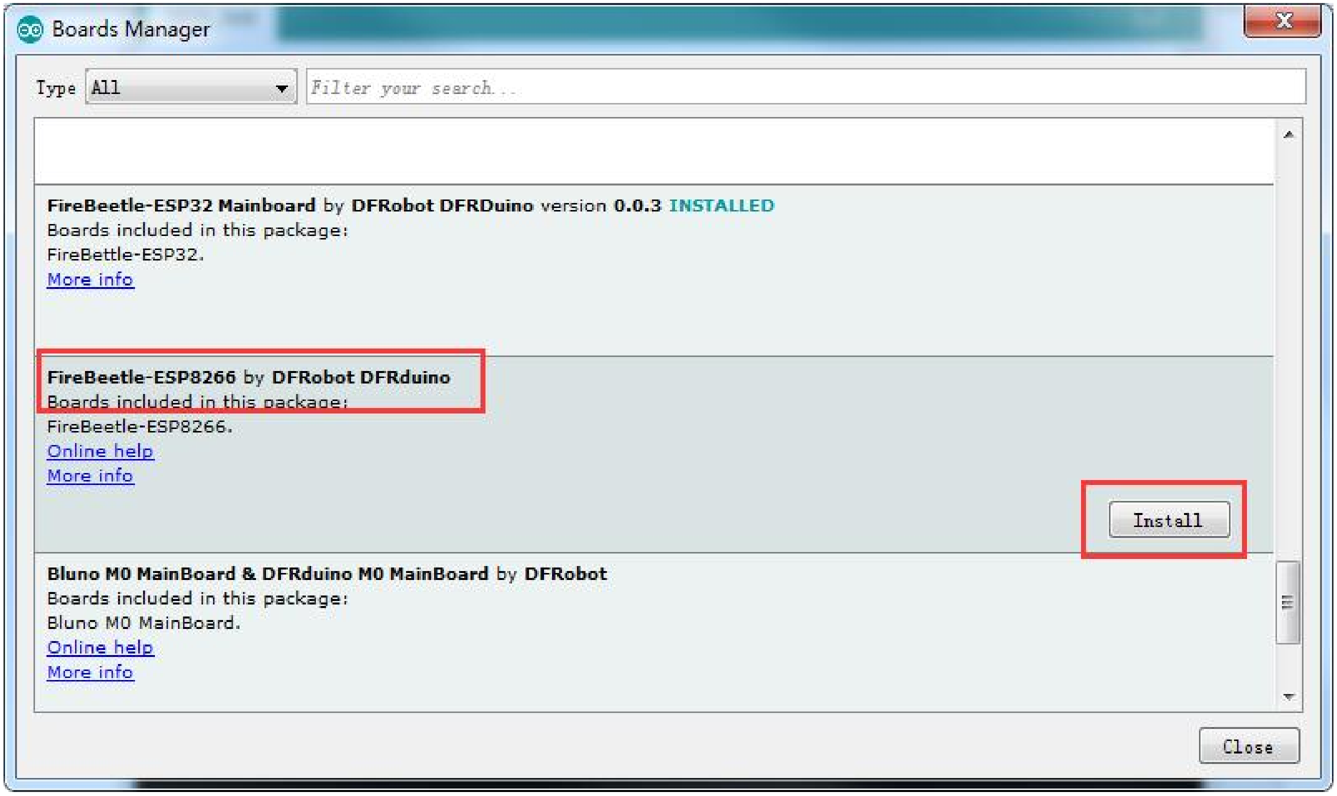
More preferences can be edited directly in the file
C:\Users\huiwang\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

(/wiki/index.php/File:Preferencesjson.png)

paste url here

- Click **OK**

- Open **Tools->Board->Boards Manager**, waiting automatic update. You'll find **FireBeetle-ESP8266**



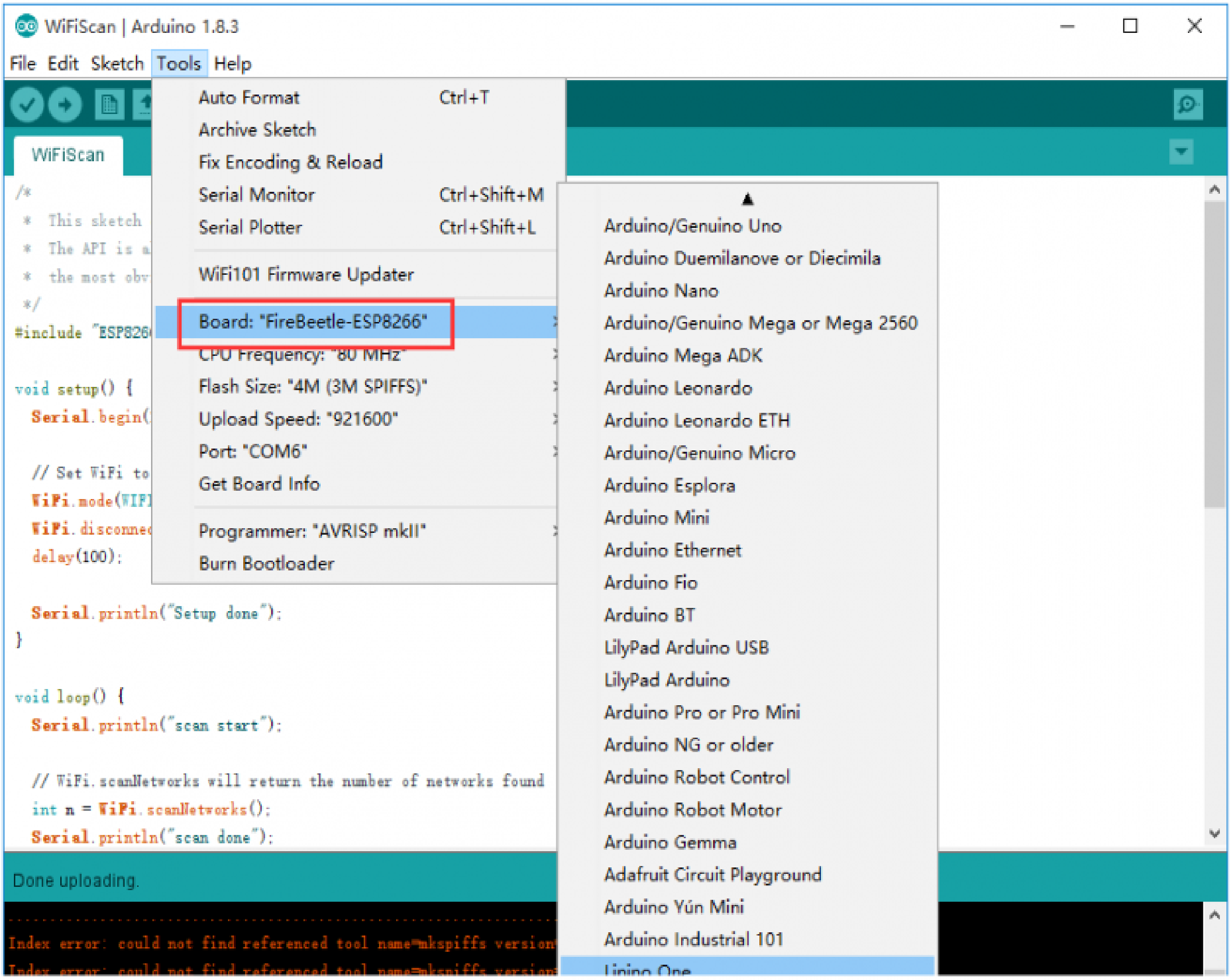
FireBeetle-ESP32 Mainboard by **DFRobot DFRDuino** version **0.0.3** **INSTALLED**
Boards included in this package:
FireBettle-ESP32.
More info

FireBeetle-ESP8266 by **DFRobot DFRduino**
Boards included in this package:
FireBeetle-ESP8266.
Online help
More info

Bluno M0 MainBoard & DFRduino M0 MainBoard by **DFRobot**
Boards included in this package:
Bluno M0 MainBoard.
Online help
More info

(/wiki/index.php/File:DFR0489_FireBeetle_ESP8266_BoardManager.jpg)

Now, the development environment has been installed, you can use it like a normal Arduino board.



Board: "FireBeetle-ESP8266"
CPU Frequency: "80 MHz"
Flash Size: "4M (3M SPIFFS)"
Upload Speed: "921600"
Port: "COM6"
Programmer: "AVRISP mkII"

```
/*
 * This sketch
 * The API is a
 * the most obv
 */
#include "ESP826

void setup() {
  Serial.begin(

  // Set WiFi to
  WiFi.mode(WIFI
  WiFi.disconne
  delay(100);

  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
```

Done uploading.

```
Index error: could not find referenced tool name=mkspiffs version
Index error: could not find referenced tool name=mkspiffs version
```

(/wiki/index.php/File:DFR0489_IDE_DFRduino_FirBeetle_ESP8266.png)

Sample Code - **Blink**

The default LED for FireBeetle Board-ESP8266 is D5 (IO2), input following code:

```
// GPIO 2 (D5) has a LED_BLINK attached to it. Give it a name:
int LED_BLINK = 2;
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BLINK, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BLINK, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BLINK, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

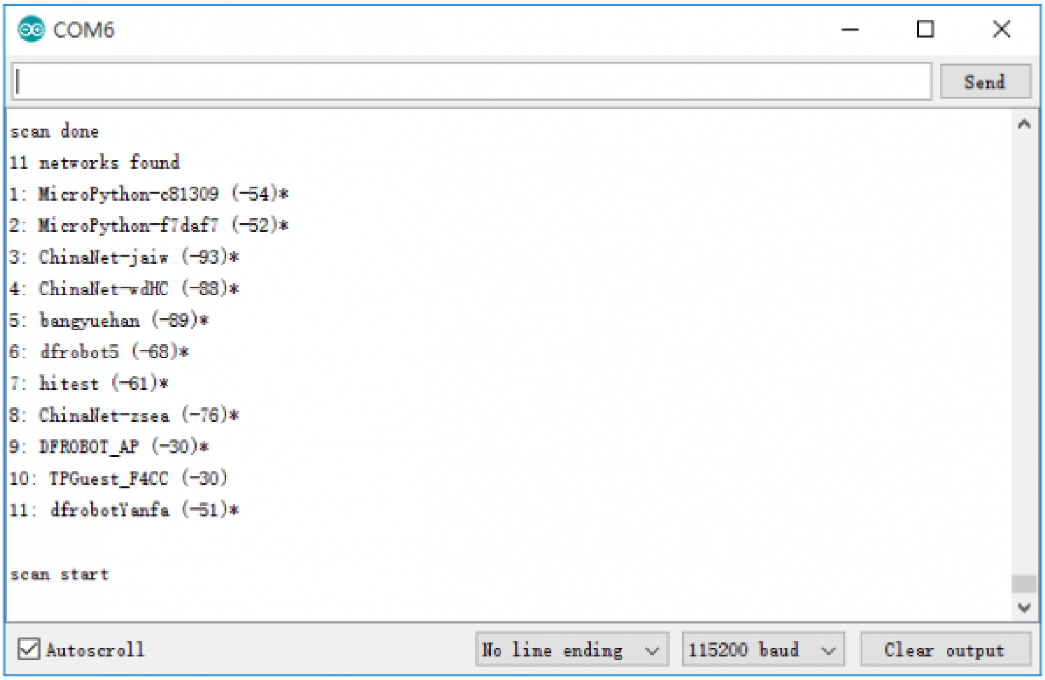
Note: ESP8266 has different pinmap in different development environment, For example: the LED connects IO2, which maps **D5** in Arduino IDE It is totally different mean with **2** and **D2** !

Sample Code - **Scan WiFi**

After you have installed the FireBeetle ESP8266 development environment, it will comes with a lot of sample code in Arduino IDE, you can find them in **File > Examples**. The follow sample code scans the around WiFi:

```
/*
 *  This sketch demonstrates how to scan WiFi networks.
 *  The API is almost the same as with the WiFi Shield library,
 *  the most obvious difference being the different file you need to include:
 */
#include "ESP8266WiFi.h"

void setup() {
  Serial.begin(115200);

  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);

  Serial.println("Setup done");
}

void loop() {
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0)
    Serial.println("no networks found");
  else
  {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i)
    {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE)?" ":"*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}
```

Open your Arduino IDE serial monitor:



```
scan done
11 networks found
1: MicroPython-c81309 (-54)*
2: MicroPython-f7daf7 (-52)*
3: ChinaNet-jaiw (-93)*
4: ChinaNet-wdHC (-88)*
5: bangyuehan (-89)*
6: dfrobot5 (-68)*
7: hitest (-61)*
8: ChinaNet-zsea (-76)*
9: DFROBOT_AP (-30)*
10: TPGuest_F4CC (-30)
11: dfrobotYanfa (-51)*

scan start
```

(/wiki/index.php/File:DFR0489_FireBeetle_WiFi_Scan.png)

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (http://www.dfrobot.com/forum/).

More Documents

- Schematic (https://github.com/Arduinolibrary/DFRobot_FireBeetle_ESP8266_DFR0489/raw/master/FireBeetle%20Board-ESP8266%20Schematic.pdf)
- Dimension (https://github.com/Arduinolibrary/DFRobot_FireBeetle_ESP8266_DFR0489/raw/master/FireBeetle%20Board-ESP8266%20Dimension.pdf)
- Datasheet (https://github.com/Arduinolibrary/DFRobot_FireBeetle_ESP8266_DFR0489/raw/master/esp-wroom-02_datasheet_en.pdf)
- Hardware Design (https://github.com/Arduinolibrary/DFRobot_FireBeetle_ESP8266_DFR0489/raw/master/esp-wroom-02_pcb_design.pdf)
- Espressif Download (http://espressif.com/en/support/download/documents)

(http://www.dfrobot.com/) Get **FireBeetle ESP8266 IOT Microcontroller** (https://www.dfrobot.com/product-1634.html) from DFRobot Store or **DFRobot Distributor**. (http://www.dfrobot.com/index.php?route=information/distributorslogo)

1.Features

- Output current in excess of 1A
- Output voltages of 5V
- Thermal overload protection
- Output transition SOA protection
- 2% output voltage tolerance
- Guaranteed in extended temperature range

2.Absolute maximum ratings

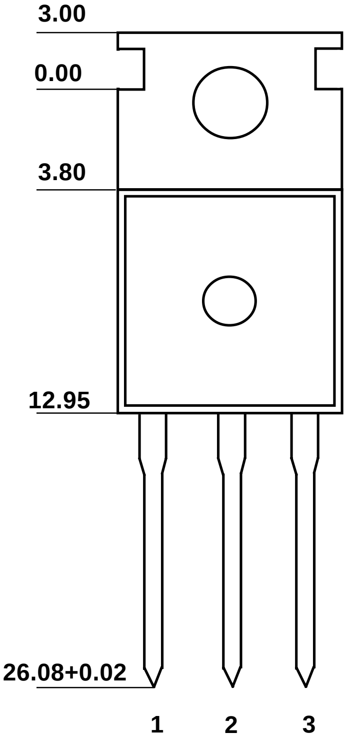
Table1: Maximum ratings(T_A=25 °C)

Parameter	Symbol	Rating	Unit
DC input voltage (V_O=5~18)	V_i	35	V
Thermal resistance junction-case	R_OJC	5	°C/W
Thermal resistance junction-ambient	R_OJA	65	°C/W
Operating juntion temperature range	T_OPR	0~125	°C
Storage temperature range	T_{stg}	-65~150	°C

3.Pin information & Package information

Table2: Pin information & Package information

Pin	Description
1	Input
2	GND
3	Output



4.Electrical characteristics

Table3: Electrical characteristics(0°C≤T_j≤125 °C,I_O=500mA,V_I=11V,C_I=0.33uF,C_O=0.1uF)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Output voltage	V_O	T_j=25 °C	4.8	5.0	5.2	V
		5.0mA≤I_O≤1.0A, P_D≤15W, 7.0V≤V_D≤20V	4.75	5.0	5.25	V
Line regulation	△V_O	T_j=25 °C,7.0V≤V_I≤25V		4.0	100	mV
		T_j=25 °C,9.0V≤V_I≤13V		1.6	50	mV
Load regulation	△V_O	T_j=25 °C,5.0mA≤I_O≤1.5A		9	100	mV
		T_j=25 °C,250mA≤I_O≤750mA		4	50	mV
Quiescent	I_q	T_j=25°C		5.0	8	mA
Quiescent current change	△I_q	5.0mA≤I_o≤1.0A		0.03	0.5	mA
		7.0V≤V_I≤25V		0.3	1.3	mA
Output voltage drift	△V_O/△T	I_O=5mA		-0.8		mV/ °C
Output noise voltage	V_n	T_A=25°C,10Hz≤f≤100KHz		42		uV
Supply voltage rejection	RR	f=120Hz,120V≤V_I≤18V	62	73		dB
Dropout	V_d	I_O=1A,T_j=25°C		2		V
Output resistance	R_O	f=1KHz		15		mohm
Short circuit current	I_{SC}	V_I=35V,T_j=25 °C		0.23		A
Short circuit peak current	I_{px}	T_j=25°C		2.2		A