# Environment Dynamic Monitoring and Remote Control of Greenhouse with ESP8266 NodeMCU

Ziyu Wan*, Yunkai Song† and Zhuli Cao‡
School of Electronics and Information Engineering
*Beijing Jiaotong University*
Beijing, China
Email: *ziyuwan@bjtu.edu.cn, †yksong@bjtu.edu.cn, ‡zhulicao@bjtu.edu.cn

***Abstract*—Aiming to reduce the labour cost in managing greenhouses, an combined greenhouse management system including environmentally dynamic monitoring system and remote control system were developed and implemented in a greenhouse. The central unit of this combined system is the ESP8266 NodeMCU board. Environmental parameters including air temperature, air humidity, CO2 concentration and soil moisture were selected to represent the state of the interior environment state of the greenhouse. Based on HTTP protocol, the ESP8266 NodeMCU connected to the Wi-Fi and communicated with IoT platforms, Thingspeak and Bylnk. The data regarding environmental parameters was collected from sensors was processed and sent to cloud and IoT platforms though Wi-Fi. Environmental data was visualized in time-series form on IoT platforms allowing users to know about the internal environment conditions in real time. Moreover, the remote system has enabled users to change the operative condition of environment controllers remotely by altering between the on-off state of control buttons on Bylnk android application. This combined system has low-cost characteristics and is suitable for indoor garden, sized greenhouse. Moreover, it can be applied in modern smart agriculture.**

***Index Terms*—ESP8266, Internet of Things, Remote Control, Environment Dynamic Monitoring, Intelligent System, Wireless Sensor Network**

## I. INTRODUCTION

Agriculture is the basic industry for economic growth and development. The growth in demand of crops is a result of a rapid increase in the world population. For growth and cultivation of crops, parameters including fields and growing condition of plants are vital. Farmers have usually applied regular strategies for all crops which has led to less yield for some specific crops.[1-3] These specific crops need specific conditions, and in this regard, smart agriculture management system can grow crops under specific conditions. Smart agriculture management system consists of monitoring and controlling of environmental variables monitoring either automatic or remote, based on ESP8266 Wi-Fi Module and IoT technology. It can provide suitable growing conditions for crops by adjusting environmental variables inside the greenhouse which can improve crop yields and meet the needs for crop growth. Environmental variables including temperature, humidity, light intensity, air quality etc. can reflect the environment condition inside the greenhouse. These environmental variables can be monitored by using various environmental sensors placed inside the greenhouse which are managed as a Wireless Sensor Network (WSN). WSN is a network composed of nodes which transmit or receive data using wireless transmission technologies. WSN plays an increasingly vital role in constructing a smart agriculture management system because farmers require to capture the real-time environment data to develop an effective strategy for improving crop yields.These sensors can be powered and then used to transmit real-time data to ESP8266 which can then analyze and process the provided data. Wireless Sensor Network (WSN) can be connected to Internet using Wi-Fi Router, and then real-time data fetched by these sensors can be transmitted to IoT platform periodically[4,5]. Then, environmental variables become visible in real time online through laptop or on IoT app on mobile devices which is useful for farmers to observe environmental condition when they do not directly work on the farm. Electronic devices with IoT technology are used increasingly nowadays such as smart television, refrigerator, washing machine, air conditioner etc., which can connect to internet and transmit data to users. Moreover, they can be remotely controlled by users using mobile phone . Users can evaluate parameters inside refrigerators, such as temperature and humidity, to judge whether their food is well preserved or not by using IoT platform. If users forget to turn off their television or refrigerator after going out, they can turn them off using IoT platform, saving energy and money. Connection of various electronic devices is the main concept in the working of Internet of Thing (IoT). Data from these electronic devices are transmitted to IoT platform through internet[6,7]. Environmental variables including temperature, humidity, light intensity, air quality etc. inside the greenhouse are monitored by using different sensors, and then ESP8266 module transmits data to IoT platform which makes environmental variables visible to users. These environmental variables can be adjusted by using various devices placed inside the greenhouse which are managed by using relay. Relay is a switch for these various devices and works according to the instructions provided from WSN. Instructions are generated by users on IoT platform, and are then transmitted to relay through ESP8266 module using internet.

978-1-5386-6243-4/19/$31.00 ©2019 IEEE

## II. System Overview

The environment dynamic monitoring system and remote control system are two systems based on the central unit of ESP8266 NodeMCU. Figure 1 shows the overview of the structure of these two systems. The environment dynamic monitoring system is developed to remotely monitor the internal environment of a greenhouse. The ESP8266 Wi-Fi Module was interfaced with sensors to monitor environmental parameters inside greenhouse, and then this data was transmitted to IoT platforms. IoT platforms and OLED Module make environmental variables visible to users. In the remote control system, the ESP8266 NodeMCU is connected to a multi-channel relay which interfaces with water pumps, humidifiers, grow lights and windows. Moreover, it is programmed to communicate with IoT platforms. Control instruction to the multi-channel relay generated by users is transmitted to the ESP8266 Wi-Fi Module through IoT platforms which achieves remote control.

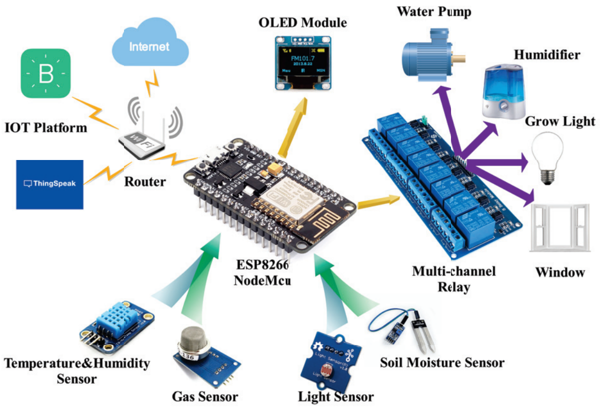


Fig. 1. System overview

## III. Environment Dynamic Monitoring System

In this part, the sensors measure the greenhouse environment parameters and send the data to the Wi-Fi Module. The environment parameters consist of air temperature, air humidity, illumination intensity and CO2 concentrations. Once the Wi-Fi module receives the data flow, the OLED Module displays the different environmental parameters at the same time. Meanwhile, ESP8266 NodeMCU sends data to the IoT platform. Therefore, the environmental parameters can be observed on the OLED Module and IoT platforms by users.

### A. The Central Unit

The ESP8266 NodeMCU which is an open-source firmware for ESP8266 Wi-Fi SOC from Espressif using an on-module and flash-based SPIFFS file system[9]. It can be programmed using Arduino IDE software and it can upload the data received from various sensors to the Internet.

Because of its characteristic of having a low cost, ESP8266 NodeMCU is selected in order to interface with sensors and to upload the environmental data to the IoT platforms though Wi-Fi. Based on the HTTP Protocol, NodeMCU can communicated with IoT platforms and cloud using API Key(Application Program Interface). Hence, users can have access to monitoring and managing their private environmental data. The pin layout of ESP8266 NodeMCU is shown in Figure 2.

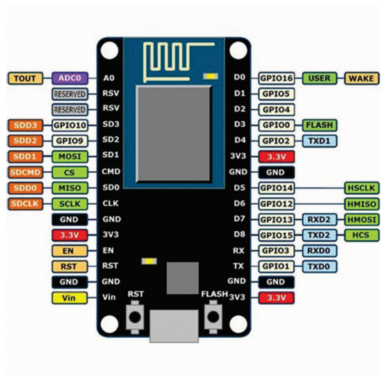

Fig. 2. Central unit: ESP8266 NodeMCU

### B. Sensors

Environmental parameters including air temperature, air humidity, soil moisture, illumination intensity and CO2 concentrations were selected to determine the environmental condition of greenhouse. The sensors used to measure the environmental parameters are shown in Figure 3. We used digital temperature humidity sensor(DHT11) to measure the parameters: temperature and humidity. DHT11 was chosen to measure these parameters because of its practicality. Soil moisture and illumination intensity parameters were measured by using soil moisture sensor and photoresistance, respectively. Moreover, MQ135 gas sensor was selected to measure CO2 concentration of the air in the interior of the greenhouse.

TABLE I
Environmental Parameters and Sensors

| Sensors | Environmental Parameters |
|---|---|
| DHT11 | Air Temperature and Humidity |
| Soil moisture sensor | Soil Moisture |
| Photoresistance | Illumination Intensity |
| MQ135 Gas Sensor | CO2 Concentration |

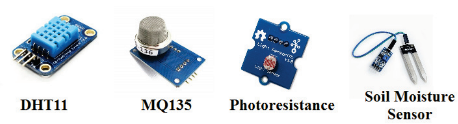


Fig. 3. Sensors

### C. Hardware Implementation

The hardware implementation for the system is shown in Figure 5. For this purpose, the digital or analog data pins were connected to the digital or analog input pins of NodeMCU and ADS1115. ADS1115 is a higher-precision ADC which provides 16-bit precision at the rate 860 samples/second over I2C[10]. It is shown in Figure 4. Since the NodeMCU has only one analog input pin and all the sensors selected, except DHT11, have analog output pins, ADS1115 Module was chosen to extend the analog input pins for NodeMCU.

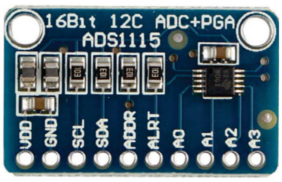


Fig. 4. ADS1115 adc module

Since NodeMCU has 9 digital input pins(from D0 to D8), the digital outputs of four sensors were directly connected to the digital input pins of NodeMCU directly. Then, the analog and digital outputs of soil moisture sensor were connected to the A0 and D3 pins of the NodeMCU, respectively. MQ135 gas sensor interfaced with D7 pin of NodeMCU and A0 pin of ADS1115. Moreover, the LDR sensor was connected to the A1 pin of ADS1115 and +5V power. All the sensors were powered by the Vin of the NodeMCU, and their GND was connected to GND of NodeMCU while ADS1115 was powered by 3.3V output of NodeMCU.

Therefore, multiple environmental sensors with analog outputs were used though the extension of analog inputs for central unit. Moreover, multi-parameter data acquisition for environmental monitoring system was implemented.

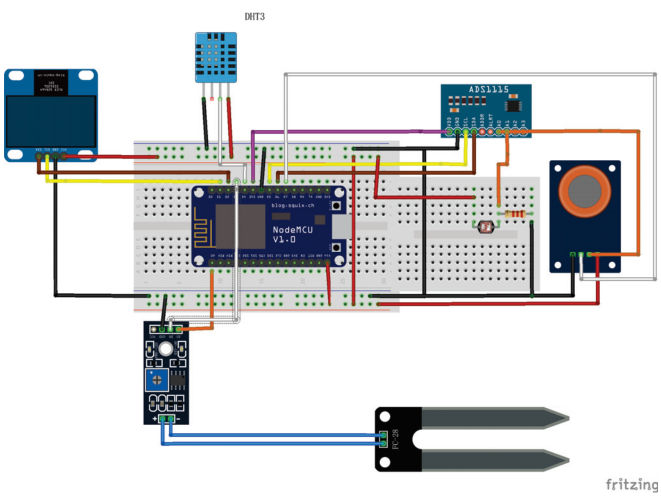


Fig. 5. Hardware implementation for environment dynamic monitoring system

### D. Software Implementation and Algorithm Design

The Arduino IDE(Integrated Development Environment) software was used to programme, compile the code written in C/C++ and upload it to flash memory of NodeMCU[5-8]. It also provides various libraries including built-in functions which can be accessed in firmware. In this project, the firmware was designed as the flow chart shown in Figure 6.


Fig. 6. Flow chart of programming

TABLE II
THE PROCEDURE OF PROGRAMMING

| Procedure of Programming | |
|---|---|
| Step 1: | Add the header files: ESP8266WiFi.h, Adafruit_ADS1015.h, BlynkSimpleEsp8266.h, ACROBOTIC_SSD1306.h, SimpleTimer.h |
| Step 2: | Set the Wi-Fi credentials: SSID and Password and set Auth Token in the Blynk App |
| Step 3: | Declare the Wi-Fi client, server(api.thingspeak.com) and API keys for private channels on Thingspeak. |
| Step 4: | Declare digital pins, analog pins and sensor types; initialize humidity, temperature, soilMoister and gasSensor to 0. |
| Step 5: | Setup the timer, declare the pinmode, debug console using *Blynk.begin(auth, ssid, pass)* and start the OLED with *oledStart()* |
| Step 6: | Set timer interval, define the address and wire of ADS1115 with *Adafruit_ADS1115 ads(0x48)* and *Wire.begin()* |
| Step 7: | Begin the ads1115, initiates SimpleTimer and connect to the Blynk using *Blynk.run()* |
| Step 8: | Note the reading of sensors "N" times, get the average and map value for proper levels. |
| Step 9: | Send sensors data to Blynk using *Blynk.virtualWrite()* and display data on OLED |
| Step 10: | Send Data to Thinkspeak Channel through API keys based on HTTP protocol. |

*E. Experimental Results*

The complete environment monitoring system was implemented as shown in Figure 7. The sensors, ADS1115 were connected to the NodeMCU using wires or Dupont lines while NodeMCU and OLED modules were fixed through female header connectors which were soldered on circuit board. Moreover, we installed MQ135 gas sensor on a button inside the greenhouse for measuring CO2 concentration more accurately. The soil moisture sensor was implemented in the soil where plants grow. Additional, DHT11 was installed on the sidewalls of greenhouse in order to sense the interior air temperature and humidity.

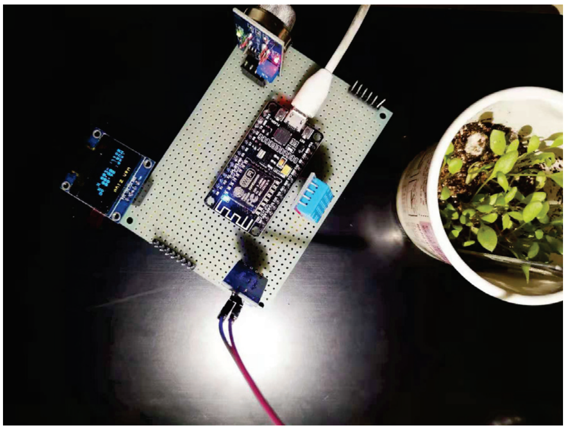

Fig. 7. Implementation of environment monitoring system

After processing of the data, the data for environmental parameters was sent to Thingspeak website in cycles through HTTP protocol. The parameters-air temperature, humidity, soil moisture and CO2 concentration was visualized in time-series form on Thingspeak as shown in Figure 8. The Thingspeak website acted as a server, and NodeMCU board was used as client. When NodeMCU was connected to Wi-Fi, the connection did not disconnect until the power went off or Wi-Fi signals disappeared.

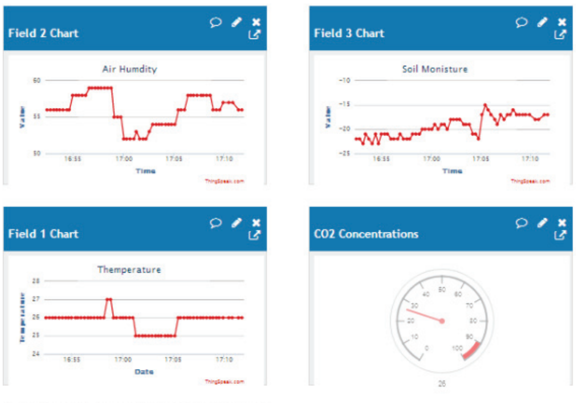


Fig. 8. Data visualization on ThingSpeak

Meanwhile, the client transmitted environmental data to Bylnk app through the Auth Token in order that users have access to these parameters. Then, we added a super chart in Bylnk app and created 4 data streams for displaying four parameters: air temperature, humidity, soil moisture and CO2 concentration in one chart as shown in Figure 9.

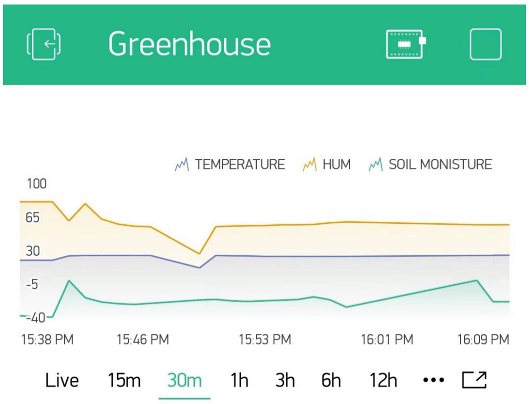


Fig. 9. Data visualization on Bylnk APP

## IV. Remote Control System

Remote control technology is essential for modern Internet of Things applications because it provides long-distance transmission which is a convenient feature for users. With the emergence of IoT technologies, remote control has been applied in various domains, such as modern agriculture. Therefore, in this research, with the aim to reduce labour cost in managing greenhouse, a remote control system was proposed. Bylnk, as one of IoT platforms, was connected with the ESP8266 NodeMCU. Mobile phone installed Bylnk application can display environmental parameters received from NodeMCU on the screen to help users know the situation in a greenhouse. Then, users can turn the controllers on/off to change environmental parameters within the greenhouse for better growth of plants by pressing the corresponding control buttons using Bylnk app.

*A. Components*

NodeMCU is once again used as the central unit. A 4-channel 5V relays module and a 2-channel 12V relays module were selected to control the switch of greenhouse controllers including humidifier, grow lights, water pump and fan. Since an humidifier is inconvenient to install in a greenhouse, an atomizer and its power circuit are shown in Figure 10(a),hence replacing it. The water pump was applied for irrigation of plants irrigation in order to change the soil moisture. Grow lights can provide the sufficient illumination intensity for plants growth. Nodemcu was thus linked with IoT platform to establish communication for facilitating direct programming required to execute remote control function. Meanwhile, Nodemcu interfaced with relay modules established within the greenhouse which include atomizer, power circuit, water pump, grow light and fan.

(a) Atomizer and power circuit (b) Water pump

(c) Grow light (d) Fan

Fig. 10. Components

### B. Hardware Implementation

The hardware implementation for the system is shown in Figure 8. These greenhouse controllers work at different voltages. In order to ensure proper working of the controllers, power relays with varying power supply ranges were selected. Hence, humidifier and grow lights were connected to 5V relays, while fan and water pump were switched to 12V relays. The input pins of relays modules were connected to GPIO(General Purpose Input/Output) pins of NodeMCU. The GPIO pins are uncommitted digital signal pins on IC, and their functions of input/outpus are defined by the user during run time. NodeMCU has 13 GPIO pins which were enough for this project.

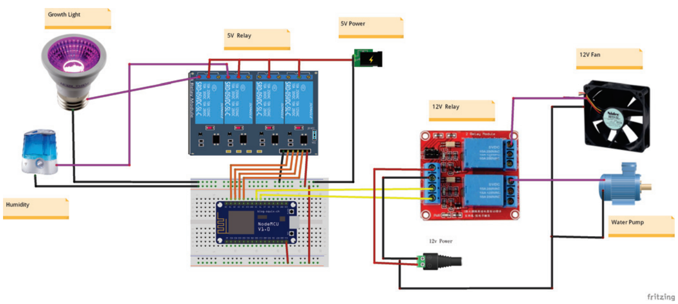


Fig. 11. Hardware implementation for remote control system

### C. Software Implementation

The firmware in this system is the example file: ESP8266_Standalone under Bylnk libraries in Arduino IDE[11]. We set the SSID and password and then created the communication between NodeMCU and Bylnk server using Auth Token provided by Bylnk. After uploading firmware to NodeMCU, the wireless channel between Bylnk and client(NodeMCU) was established. Next, we created 6 control buttons as shown in Figure 12. and set the output pins used fot giving control commands to NodeMCU. After a control command was received by NodeMCU, the states of channels on relay modules were changed from on to off or from off to on.

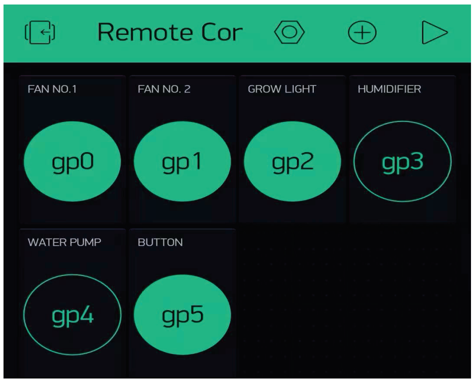


Fig. 12. The control buttons in Bylnk APP

## V. Combined System Prototype

We combined these two systems and implemented them in a greenhouse made up of wooden frame and PVC materials. Figure 13 shows the prototype of the combined system with the greenhouse.

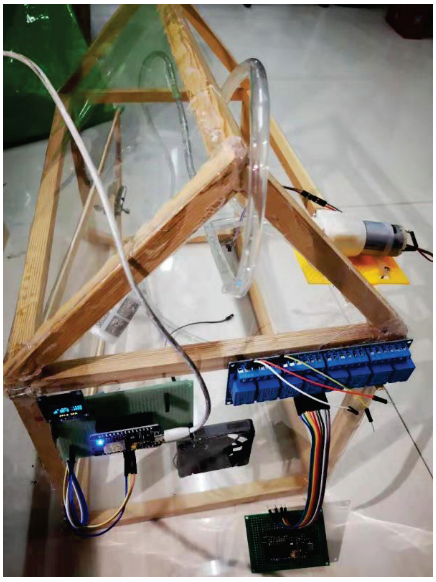

Fig. 13. Prototype of combined system

## VI. Conclusions and Future Work

This research paper presents a combined greenhouse management system including environment dynamic monitoring system and remote control system. Environmental parameters including air temperature, air humidity, CO2 concentration, soil moisture were selected to represent the interior environmental state of a greenhouse. These parameters were efficiently monitored and analyzed using the installed sensors which were in turn linked with NodeMCU. The ESP8266 NodeMCU was used to connect to Wi-Fi and to communicate with IoT platforms-Thingspeak and Bylnk based on HTTP protocol.

After uploading the firmware we designed for NodeMCU, the data for environmental parameters collected from sensors was processed and sent to cloud and IoT platforms though Wi-Fi. Meanwhile, data was visualized in time-series form on IoT platform, and users could gain access to it and know about the internal environment conditions in real time. After connecting the GPIO pins of NodeMCU to relay modules, users are enabled to remotely change the on-off state of relays on Bylnk app in order to control the operation condition of controllers.

This greenhouse management system has a low-cost characteristics and is suitable for indoor garden or small-sized greenhouse. Moreover, it can be extended to reduce the labour cost in modern agriculture because it realizes the wireless remote control for greenhouses. It also provides efficient visual representation of the parameters of the internal greenhouse environment. Moreover, this management system facilitates users to remotely control the operation condition of the controllers established inside the greenhouse. In the future, we will evaluate more environmental parameters (including water flow) for monitoring and will aim to implement system in modern smart agriculture field.

## REFERENCES


[1] Thakur, D., et al. (2018). ”Real Time Monitoring of Valeriana Jatamansi Plant for Growth Analysis.” Procedia Computer Science 132: 507-517.

[2] Liang, M.-H., et al. (2018). ”Greenhouse Environment dynamic Monitoring system based on WIFI.” IFAC-PapersOnLine 51(17): 736-740.

[3] Pallavi S., Jayashree D. Mallapur, Kirankumar Y. Bendigeri, Remote Sensing and Controlling of Greenhouse Agriculture Parameters based on IoT, 2017 International Conference on Big Data, IoT and Data Science (BID) Vishwakarma Institute of Technology, Pune, Dec 20-22, 2017.

[4] Saokaew, A., et al. (2018). A smart photovoltaic system with Internet of Thing: A case study of the smart agricultural greenhouse. 2018 10th International Conference on Knowledge and Smart Technology (KST).

[5] Pakalapati, S. S., et al. (2017). A prosthetic hand control interface using ESP8266 Wi-Fi module and Android application. 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).

[6] Saha, S. and A. Majumdar (2017). Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system. 2017 Devices for Integrated Circuit (DevIC).

[7] Singh, P. and S. Saikia (2016). Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module. 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC).

[8] Kodali, R. K. and S. Mandal (2016). IoT based weather station. 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT).

[9] NodeMCU Documentation. Available: https://nodemcu.readthedocs.io/en/master/

[10] ADS1115 16-BIT ADC - 4 CHANNEL WITH PROGRAMMABLE GAIN AMPLIFIER Available: https://www.adafruit.com/product/1085

[11] The blynk-library Available: https://github.com/blynkkk/blynk-library/releases/tag/v0.5.4