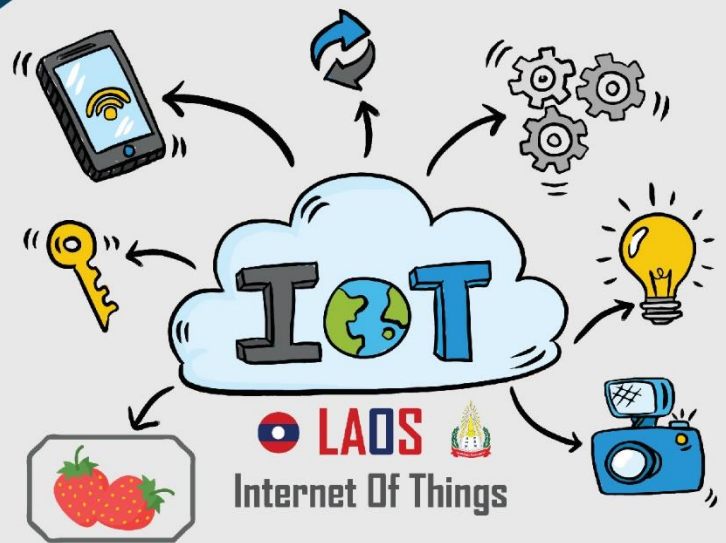


ວິທີການນຳໃຊ້ NodeMCU

ໃນການເຊື່ອມຕໍ່ ແລະ ຄວບຄຸມອຸປະກອນ IoT
ຜ່ານອິນເຕີເນັດ ດ້ວຍ
Mobile Apps & Blynk Server



ESP8266 V2



ຄຳນຳ

ສຳລັບ E-Book ວິທີການນຳໃຊ້ NodeMCU ໃນການເຊື່ອມຕໍ່ ແລະ ຄວບຄຸມອຸປະກອນ IoT ຜ່ານອິນເຕີເນັດ ດ້ວຍ Mobile Apps & Blynk Server ຫົວນີ້ ເນື້ອໃນຈະລົງເລິກໄປໃນທາງການທຳຄວາມເຂົ້າໃຈ ແລະ ວິເຄາະແບບເປັນຂັ້ນເປັນຕອນ ພ້ອມທັງລົງມືປະຕິບັດໄປພ້ອມກັນ ດັ່ງນັ້ນຜູ້ຂຽນຫວັງຢ່າງຍິ່ງວ່າ E-Book ຫົວນີ້ ຈະເປັນຕົວຊ່ວຍໃນການຮຽນຮູ້ຜູ້ນຳ ແລະ ເປັນແຮງຊີ້ນຳໃຫ້ກັບຜູ້ອ່ານທີ່ກຳລັງສົນໃຈການພັດທະນາອຸປະກອນສຳຫລັບດ້ານ IoT ໄດ້ເຂົ້າໃຈ ແລະ ສາມາດນຳໄປຕໍ່ຍອດຫລືປະຍຸກໃຊ້ໃນດ້ານຕ່າງໆ

ສາລະບານ

ບົດທີ 1 ການກະກຽມອຸປະກອນ ແລະ ໂປຣແກມໃຫ້ພ້ອມ

- I. ສ່ວນປະກອບຂອງບອດ NodeMCU ESP82661
- II. ການກະກຽມຄວາມພ້ອມກ່ອນລົງມືພັດທະນາ IoT2

ບົດທີ 2 ການເຊື່ອມຕໍ່ ແລະຄວບຄຸມອຸປະກອນ IoT ຜ່ານອິນເຕີເນັດ ດ້ວຍ Mobile Apps ແລະ Blynk Server

- I. Blynk ແມ່ນຫຍັງ? ແລະການເຮັດວຽກແບບໃດ?3
- II. ກະກຽມຄວາມພ້ອມກ່ອນເລີ່ມຕົ້ນໃຊ້ງານ Blynk4
 - 1. ດາວໂຫລດ ແລະ ຕິດຕັ້ງ Blynk App 4
 - 2. ລົງທະບຽນສ້າງ Account ດ້ວຍອີເມວແທ້ພ້ອມກຳນົດລະຫັດຜ່ານໃນ Blynk App 5
 - 3. ຕິດຕັ້ງ Blynk Libraries ໃຫ້ກັບ Arduino IDE ໄວ້ສຳລັບຮ້ອງໃຊ້ເວລາຂຽນໂປຣແກມ 5

ບົດທີ 3 ເລີ່ມຕົ້ນໃຊ້ງານ Blynk ດ້ວຍການສ້າງໂປຣເຈັກໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່

- I. ເລີ່ມຕົ້ນໃຊ້ງານ Blynk ດ້ວຍການສ້າງໂປຣເຈັກໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່6
 - ຕົວຢ່າງ: ວົງຈອນຄວບຄຸມການເປີດ ແລະ ປິດດອກໄຟ(ໄອໂດດເປັ້ງແສງ) LED 6
- II. ໃຊ້ງານ Blynk ກັບ ເຊັນເຊີ (Sensor) ດ້ວຍການສ້າງໂປຣເຈັກໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່.....12
 - ຕົວຢ່າງ: ວົງຈອນວັດອຸນຫະພູມ (Temperature) ແລະ ຄວາມຊຸ່ມຂອງອາກາດ (Humidity)12

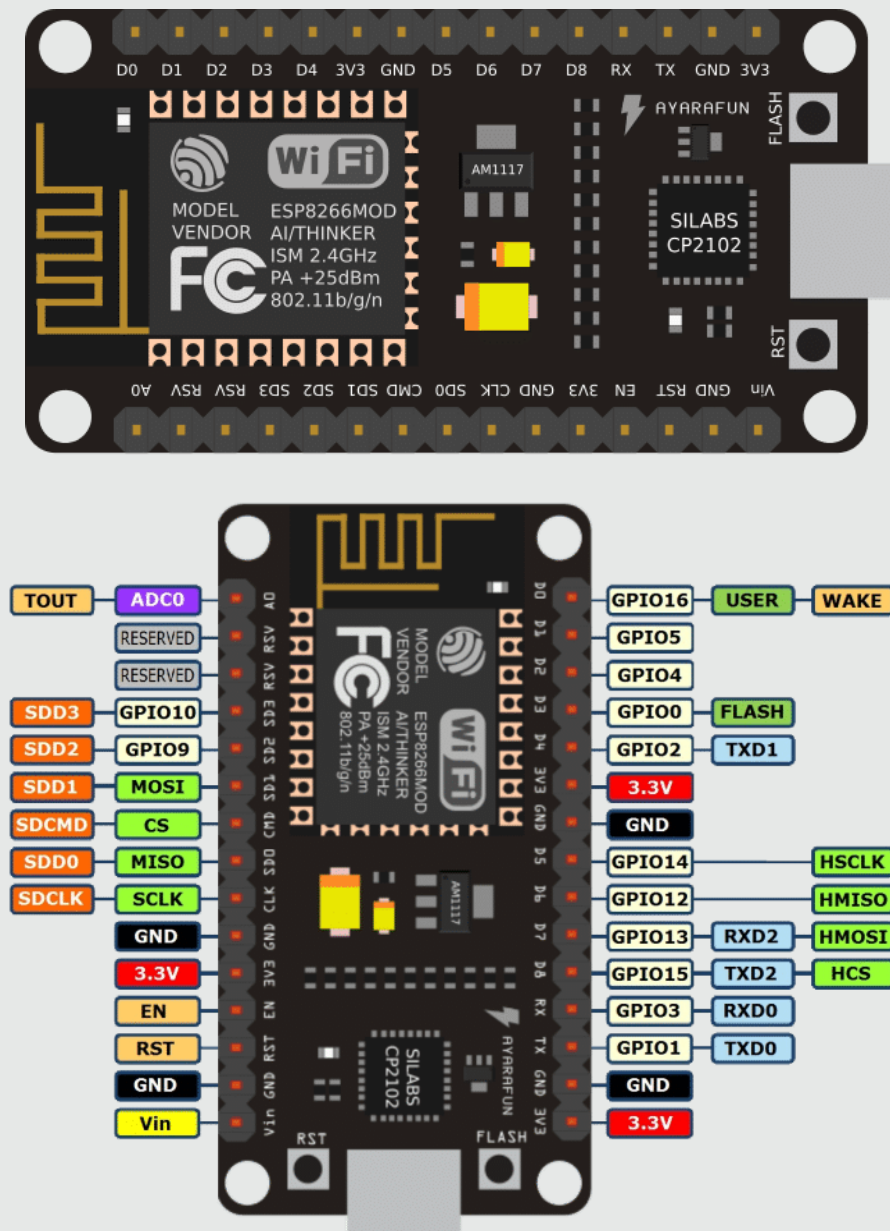
ບົດທີ 1

ການກະກຽມອຸປະກອນ ແລະ ໂປຣແກມໃຫ້ຝ່ອມ

I. ສ່ວນປະກອບຂອງບອດ NodeMCU ESP8266

ESP8266 ແມ່ນຊື່ເອີ້ນຊັບຂອງໂມດູນ ESP8266 ສໍາລັບການສື່ສານບິນມາດຕະຖານ WiFi, ເຊິ່ງໃຊ້ງານດ້ວຍແຮງດັນໄຟຟ້າ 3.0-3.6V, ໃຊ້ກະແສໄຟຟ້າສະເລ່ຍ 80mA, ຮອງຮັບຄໍາສັ່ງ deep sleep ເພື່ອປະຢັດພະລັງງານ, ໃຊ້ກະແສນ້ອຍກວ່າ 10 ໂມໂຄແອມ ສາມາດ wake up ກັບມາສິ່ງຂໍ້ມູນໂດຍໃຊ້ເວລາໜ້ອຍກວ່າ 2 ມິນລີ/ວິນາທີ, ພາຍໃນມີ Low power MCU 32bit ເຮັດໃຫ້ເຮົາຂຽນໂປຣແກມສ້າງງານໄດ້ແບບມີຮອບວຽນ analog digital converter ເຮັດໃຫ້ສາມາດອ່ານຄ່າຈາກ analog ໄດ້ຄວາມລະອຽດ 10bit ແລະ ສາມາດເຮັດວຽກໄດ້ໃນອຸນຫະພູມ -40 ເຖິງ 125 ອົງສາ.

ໂດຍລາຍລະອຽດ ແລະ ສ່ວນປະກອບຕ່າງໆຂອງ NodeMCU ESP8266 V2 ມີດັ່ງນີ້:



- NodeMCU V2 ແມ່ນ ESP8266-12E ປະສົມປະສານກັບ USB TTL ທີ່ໃຊ້ຊິບ CP2102 ແລະ ຂະຫຍາຍຂາໃຫ້ສາມາດຕໍ່ທົດລອງໄດ້ງ່າຍຂຶ້ນ ມີປຸ່ມ reset ແລະ flash ສໍາລັບໃຊ້ໂປຣແກມໂດຍໃຊ້ Arduino IDE ຫຼືໂປຣແກມອື່ນໆ ໄດ້ຢ່າງສະດວກ.

II. ການກະກຽມຄວາມພ້ອມກ່ອນລົງມືພັດທະນາ IoT

ສິ່ງທີ່ກ່ຽວຂ້ອງກັບການພັດທະນາ IoT ຈະມີທັງ ຮາດແວ (Hardware) ແລະ ຊອບແວ (Software) ໂດຍຮາດແວ ຈະເປັນການນໍາເອົາອຸປະກອນແລະຊັ້ນສ່ວນຕ່າງໆ ມາເຊື່ອມຕໍ່ກັນເປັນວົງຈອນ ໂດຍມີບອດຫລັກທີ່ເຮັດຫນ້າທີ່ເປັນຕົວກາງໃນການເຊື່ອມຕໍ່ລະຫວ່າງອຸປະກອນ I/O ຕ່າງໆ ເຊິ່ງໃນຫນັງສືເຫລົ່ານີ້ໄດ້ເລືອກໃຊ້ NodeMCU ESP8266 ມາປະກອບເປັນບົດຕົວຢ່າງ ເນື່ອງຈາກມີຂະໜາດນ້ອຍ ລາຄາບໍ່ແພງ ກົນໄຟຫນ້ອຍ ມີຂາ I/O ແລະອິນເຕີເຟສສໍາລັບໃຊ້ງານ IoT ມາໃຫ້ຂ້ອນຂ້າງຄົບ ແຖມຍັງສາມາດເຊື່ອມຕໍ່ກັບເຄືອຂ່າຍ Wi-Fi ໄດ້ດ້ວຍ ເຊິ່ງເຫມາະສໍາລັບການເລີ່ມຕົ້ນຮຽນຮູ້ການພັດທະນາ IoT ໄດ້ເປັນຢ່າງດີ ສ່ວນຊອບແວ ຈະເປັນການຂຽນໂປຣແກມຄວບຄຸມດ້ວຍພາສາຊີລິງໃນບອດ ເພື່ອໃຫ້ສາມາດອ່ານຄ່າຈາກເຊັນເຊີ ແລະຄວບຄຸມອຸປະກອນ I/O ຕ່າງໆ ໄດ້ຕາມຕ້ອງການ.

ສະນັ້ນ, ກ່ອນທີ່ເຮົາຈະເລີ່ມຕົ້ນລົງມືພັດທະນາ IoT ເຮົາຄວນຈະຕ້ອງກະກຽມຄວາມພ້ອມທາງດ້ານ ຮາດແວ ແລະ ຊອບແວກ່ອນ ໂດຍເລີ່ມຈາກ

- ດາວໂຫລດ ແລະຕິດຕັ້ງໂປຣແກມ Arduino IDE ທີ່ໃຊ້ສໍາລັບຂຽນໂຄດ ກວດສອບ/ປະມວນຜົນໂຄດ ແລະອັບໂຫລດໂຄດລົງໃນບອດ
 - ລົງວິທີການດາວໂຫລດ
<https://www.arduino.cc/en/guide/windows#toc2>
 - ລົງການດາວໂຫລດໂປຣແກມ Arduino IDE
<https://www.arduino.cc/en/Main/Software>
- ຕິດຕັ້ງຊຸດຟາຍໄລບາລີ (Libraries) ທີ່ສະໜັບສະໜູນບອດທີ່ເຮົາໃຊ້ ເພື່ອຊ່ວຍໃນການຂຽນໂປຣແກມ
 - ລົງວິທີການຕິດຕັ້ງຊຸດຟາຍໄລບາລີ (Libraries)
<https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>
- ດາວໂຫລດ ແລະຕິດຕັ້ງໄດຣເວີ (Driver) ໃຫ້ກັບບອດທີ່ເຮົາໃຊ້
ໝາຍເຫດ: ສະເພາະຄອມພິວເຕີບາງເຄື່ອງທີ່ບໍ່ມີ
 - ລົງຕິດຕັ້ງໄດຣເວີ CP2102
<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
- ກວດສອບພອດ (COM Port) ຫລືຊ່ອງທາງທີ່ໃຊ້ຕິດຕໍ່ສື່ສານກັບບອດ (ຈາກຄອມພິວເຕີ)
- ເລືອກພອດ (COM Port) ທີ່ໃຊ້ຕິດຕໍ່ສື່ສານກັບຄອມພິວເຕີ (ຈາກ Arduino IDE)

ບົດທີ 2

ການເຊື່ອມຕໍ່ ແລະຄວບຄຸມອຸປະກອນ IoT ຜ່ານອິນເຕີເນັດ ດ້ວຍ Mobile Apps ແລະ Blynk Server

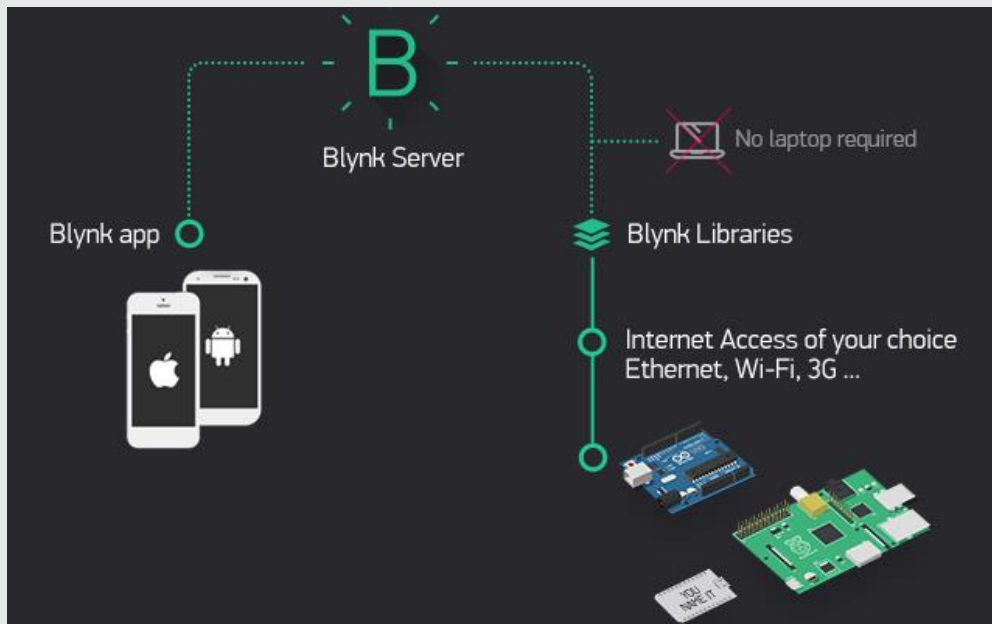
I. Blynk ແມ່ນຫຍັງ? ແລະການເຮັດວຽກແບບໃດ?

Blynk Platform ຖືກອອກແບບມາເພື່ອຄວບຄຸມ Internet of Things ເຊິ່ງຄຸນນະສົມບັດໃນການຄວບຄຸມຈາກໄລຍະໄກຜ່ານເຄືອຂ່າຍອິນເຕີເນັດ ແລະຍັງສາມາດສະແດງຜິດພາດຈາກເຊັ່ນເຊີຕ່າງໆໄດ້ອີກດ້ວຍ.



ການເຮັດວຽກຈະປະກອບດ້ວຍ 3 ອົງປະກອບດັ່ງຕໍ່ໄປນີ້

- **Blynk App** - ແອັບພລິເຄຊັນທີ່ສາມາດຕິດຕັ້ງໃນໂທລະສັບມືຖືຂອງພວກເຮົາເພື່ອສ້າງອິນເຕີເຟດ (Interface) ເພື່ອໃຊ້ໃນການຄວບຄຸມຫລືສະແດງຄ່າຕ່າງໆຈາກອຸປະກອນ Internet of Things.
- **Blynk Server** - ເຮັດໜ້າທີ່ເປັນຕົວກາງໃນການສື່ສານລະຫວ່າງແອັບພລິເຄຊັນ ແລະ Internet of Things (ໃນສ່ວນນີ້ພວກເຂົາໄດ້ໃຫ້ບໍລິການຟຣີ ແຕ່ມີຂໍ້ຈຳກັດຈຳນວນໜຶ່ງ).
- **Blynk Libraries** - ຖືກອອກແບບມາສຳລັບອຸປະກອນ Internet of Things ຕ່າງໆ ເພື່ອໃຫ້ສື່ສານກັນໄດ້ຢ່າງມີປະສິດທິພາບ.



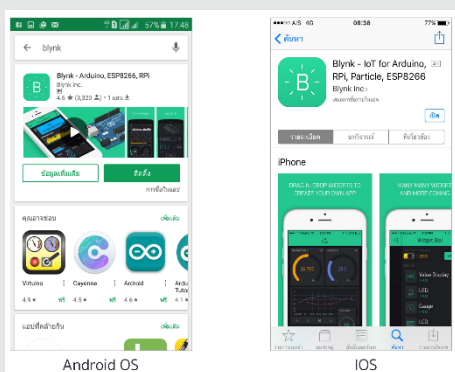
ແຜນຮ່າງ (Blynk Platform)

ການເຮັດວຽກຂອງ Blynk ເລີ່ມຕົ້ນຈາກການທີ່ເຮົາລົງທະບຽນໃຊ້ງານ Blynk ຜ່ານ Blynk App ທີ່ເຮົາຕິດຕັ້ງໄວ້ໃນໂທລະສັບ Smart Phone ຕ່າງໆ ເຊິ່ງຈະຕ້ອງມີການລະບຸອີເມວ (Email) ທີ່ຕ້ອງໃຊ້ງານໄດ້ຈິງໃນການສະຫມັກ ແລະອີເມວນີ້ແມ່ນລະບົບຈະໄດ້ໃຫ້ໃຊ້ລະຫັດ TOKEN ໄປໃຫ້ ເວລາທີ່ເຮົາລົງມືສ້າງແຕ່ລະ Project ເຊິ່ງລະຫັດທີ່ວ່ານີ້ປຽບສະເໝືອນກັບເປັນເລກບັດປະຈຳໂຕປະຊາຊົນທີ່ຈະຕ້ອງບໍ່ຊ້ຳກັນໃນແຕ່ລະອຸປະກອນ ເພື່ອໃຊ້ຢືນຢັນຕົວຕົນວ່າເປັນອຸປະກອນໂຕໃດ ໂດຍເຮົາຈະຕ້ອງລະບຸລະຫັດ TOKEN ນີ້ ໄວ້ໃນໂຄດໂປຣແກມສະເໝີ ພ້ອມທັງລະບຸຊື່ SSID ແລະລະຫັດເພື່ອເຊື່ອມຕໍ່ WiFi ຈາກນັ້ນກໍອັບໂຫລດໂຄດໂປຣແກມໃນບອດ ເມື່ອບອດ ຫລື ອຸປະກອນ IoT ເຊື່ອມຕໍ່ກັບອິນເຕີເນັດຜ່ານທາງ WiFi ໄດ້ແລ້ວກໍຈະເຊື່ອມຕໍ່ໄປຍັງ Blynk Server ເຊິ່ງ Blynk Server ຈະແຍກອຸປະກອນທີ່ເຊື່ອມຕໍ່ເຂົ້າມາ ຖ້າລະຫັດ TOKEN ຄືກັນ ກໍຈະໄດ້ຮັບການຍິນຍອມໃຫ້ເຊື່ອມຕໍ່ກັບ Blynk App ທີ່ຢູ່ໃນໂທລະສັບ Smart Phone ຂອງເຮົາຜ່ານທາງ Blynk Server ໄດ້.

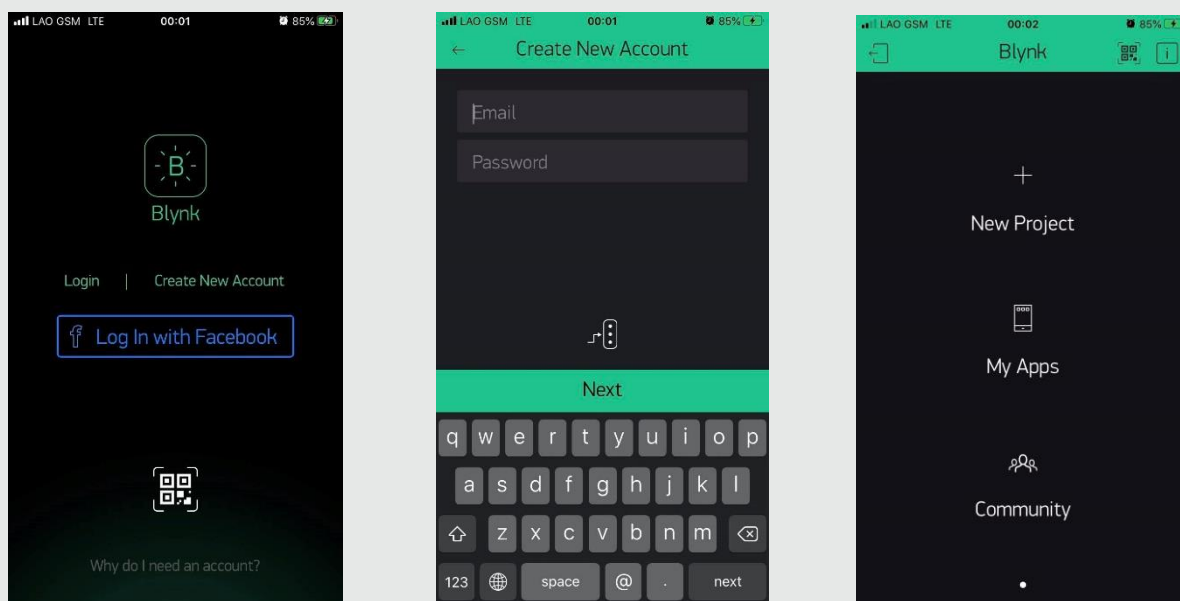
II. ກະກຽມຄວາມພ້ອມກ່ອນເລີ່ມຕົ້ນໃຊ້ງານ Blynk

ກ່ອນເລີ່ມຕົ້ນໃຊ້ງານ Blynk ໃນຄັ້ງທຳອິດ ສິ່ງທີ່ເຮົາຕ້ອງກະກຽມມີຄື:

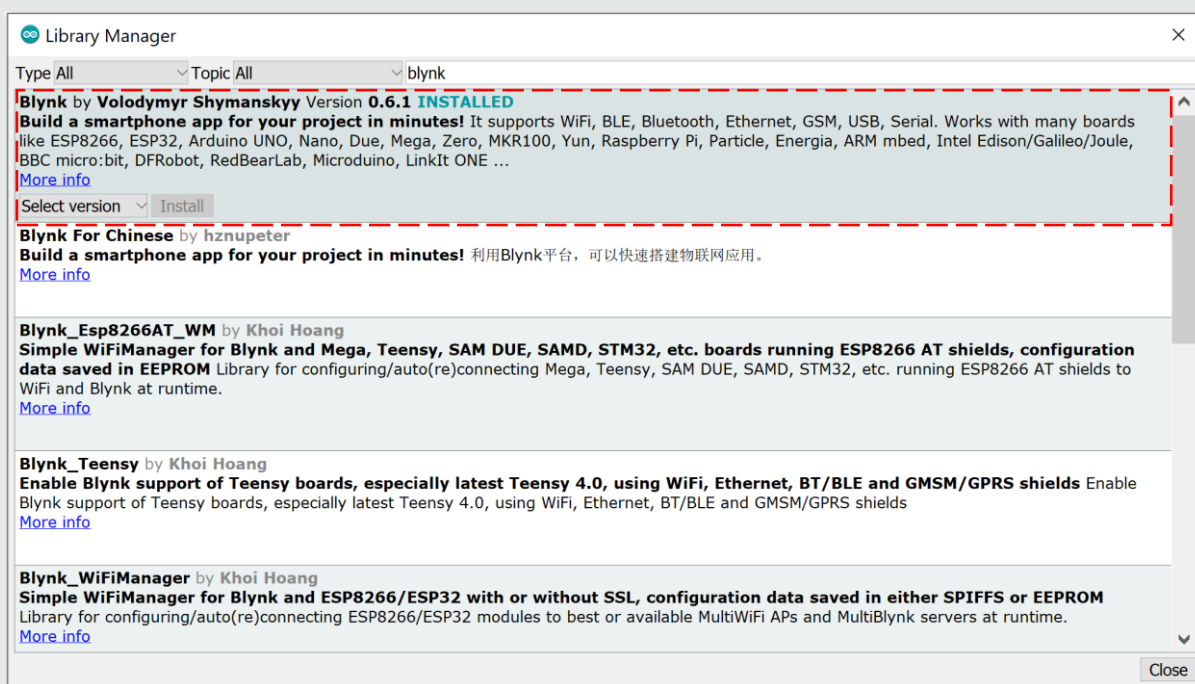
1. ດາວໂຫລດ ແລະ ຕິດຕັ້ງ Blynk App



2. ລົງທະບຽນສ້າງ Account ດ້ວຍອີເມວແທ້ຜ່ານກຳນົດລະຫັດຜ່ານໃນ Blynk App



3. ຕິດຕັ້ງ Blynk Libraries ໃຫ້ກັບ Arduino IDE ໄວ້ສຳລັບເອີ້ນໃຊ້ເວລາຂຽນໂປຣແກມ

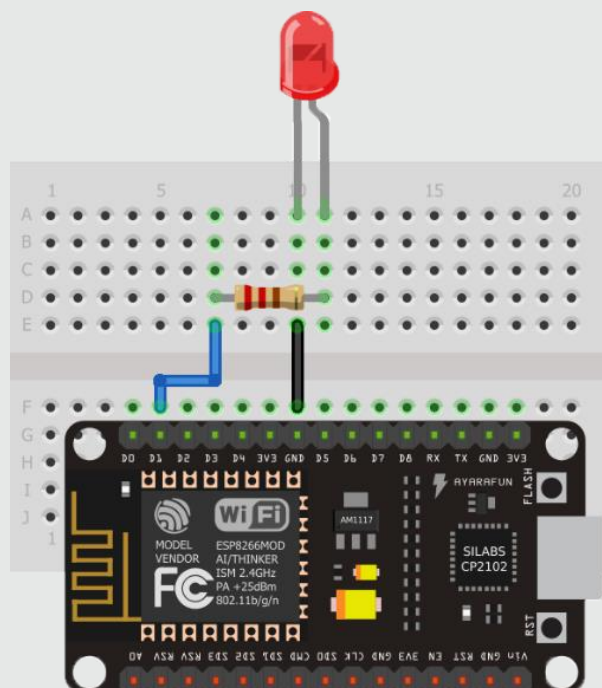


ບົດທີ 3

ເລີ່ມຕົ້ນໃຊ້ງານ Blynk ດ້ວຍການສ້າງໂປຣເຈັກໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່

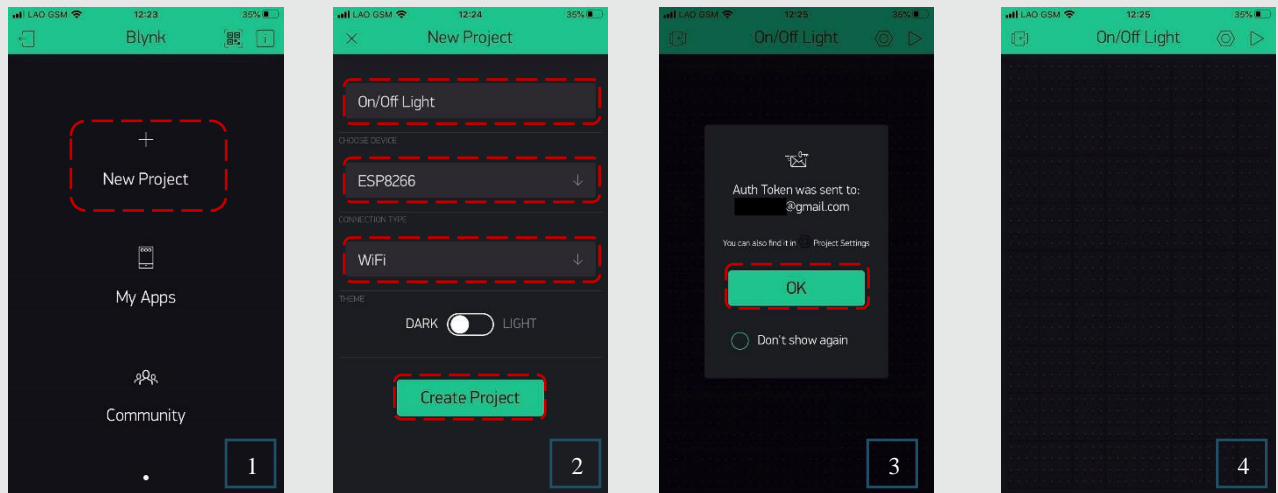
I. ເລີ່ມຕົ້ນໃຊ້ງານ Blynk ດ້ວຍການສ້າງໂປຣເຈັກໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່

ຕົວຢ່າງ: ວົງຈອນຄວບຄຸມການເປີດ ແລະ ປິດດອກໄຟ(ໄອໂດດເບິ່ງແສງ) LED

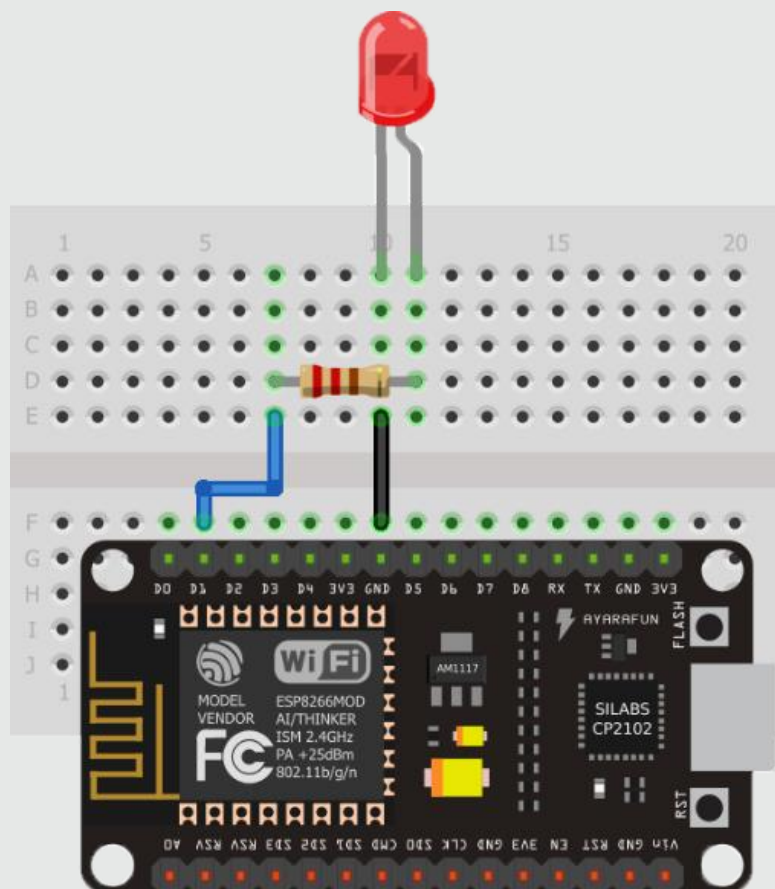


ວິທີການຕໍ່ວົງຈອນ ແລະຂັ້ນຕອນການຂຽນໂປຣແກມມີດັ່ງນີ້:

1. ເມື່ອ Login ເຂົ້າມາໄດ້ແລ້ວ ໃຫ້ກົດທີ່ +New Project ເພື່ອເລີ່ມຕົ້ນສ້າງໂປຣເຈັກ
2. ໃນຫນ້າຕ່າງ New Project ໃຫ້ເຮົາຕັ້ງຊື່ໂປຣເຈັກ (ໃນທີ່ນີ້ຕັ້ງຊື່ວ່າ On/Off Light), ເລືອກອຸປະກອນທີ່ໃຊ້ (ໃນທີ່ນີ້ໃຊ້ບອດ NodeMCU ESP8266 ດັ່ງນັ້ນເລືອກ ESP8266), ເລືອກຊະນິດເຊື່ອມຕໍ່ (ໃນທີ່ນີ້ເລືອກ WiFi) ຈາກນັ້ນກົດປຸ່ມ Create Project
3. ລະບົບຈະບັນທຶກຂໍ້ມູນຜ່ານທັງສ້າງລະຫັດ Token ໃຫ້ ແລະສົ່ງລະຫັດ Token ນີ້ ໄປຫາອີເມວທີ່ສະຫມັກຫລື Login ເຂົ້າມາ (ເຮົາຈະຕ້ອງນຳເອົາລະຫັດທີ່ວ່ານີ້ໄປໃສ່ໃນໂຄດໂປຣແກມເຊິ່ງຈະດຳເນີນການໃນພາຍຫລັງ) ກົດປຸ່ມ OK
4. ຕໍ່ມາກໍຈະເຫັນເປັນຜື່ນທີ່ຫວ່າງເປົ່າ



5. ລົງມືຕໍ່ວົງຈອນຄວບຄຸມການເປີດປິດຫລອດໄຟ LED ຕາມຮູບລຸ່ມນີ້ (ໃນທີ່ນີ້ເລືອກໃຊ້ຂາ D1 ຫລື GPIO05 ເປັນຂາເອົາພຸດທີ່ເຊື່ອມຕໍ່ກັບຕົວຕ້ານທານ 220Ω ແລະ ຂາບວກຫລືຂາ Anode ຂອງ ດອກໄຟ)



6. ເປີດ Arduino IDE ເພື່ອຂຽນໂຄດລົງໄປໃນບອດ NodeMCU ESP8266

```

#define BLYNK_PRINT Serial

// ລາຍງານສະຖານະການເຊື່ອມຕໍ່ຂອງ Blynk ອອກ Serial Monitor

#include <ESP8266WiFi.h>

// ເອີ້ນໃຊ້ Libraries ESP8266WiFi.h

#include <BlynkSimpleEsp8266.h>

// ເອີ້ນໃຊ້ Libraries BlynkSimpleEsp8266.h

char auth[] = "YourAuthToken";

// Copy Token ທີ່ໄດ້ຮັບຈາກອີເມວກ່ອນໜ້ານີ້ (ຫຼັງຈາກສ້າງ New Project ແລ້ວ) ມາໃສ່ໃນ “ ”

// ຕົວຢ່າງ char auth[] = "6eoDkkPhHy_yMgD2ez3mOkUfxyIdMnjT";

char ssid[] = "YourNetworkName";

// ໃສ່ຊື່ SSID ຂອງ WiFi Network ທີ່ຈະໃຫ້ອຸປະກອນ IoT ເຊື່ອມຕໍ່

char pass[] = "YourPassword";

// ໃສ່ລະຫັດຜ່ານທີ່ຈະເຊື່ອມຕໍ່ກັບ WiFi Network

void setup()
{
  Serial.begin(9600);

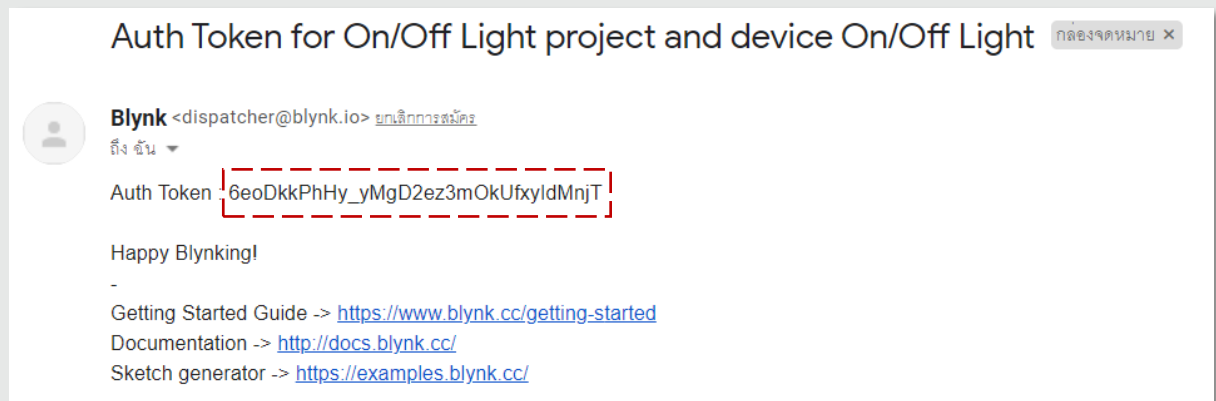
  Blynk.begin(auth, ssid, pass);

  // ເລີ່ມຕົ້ນລະບົບການເຊື່ອມຕໍ່ກັບ WiFi Network ແລະ Blynk Server
}

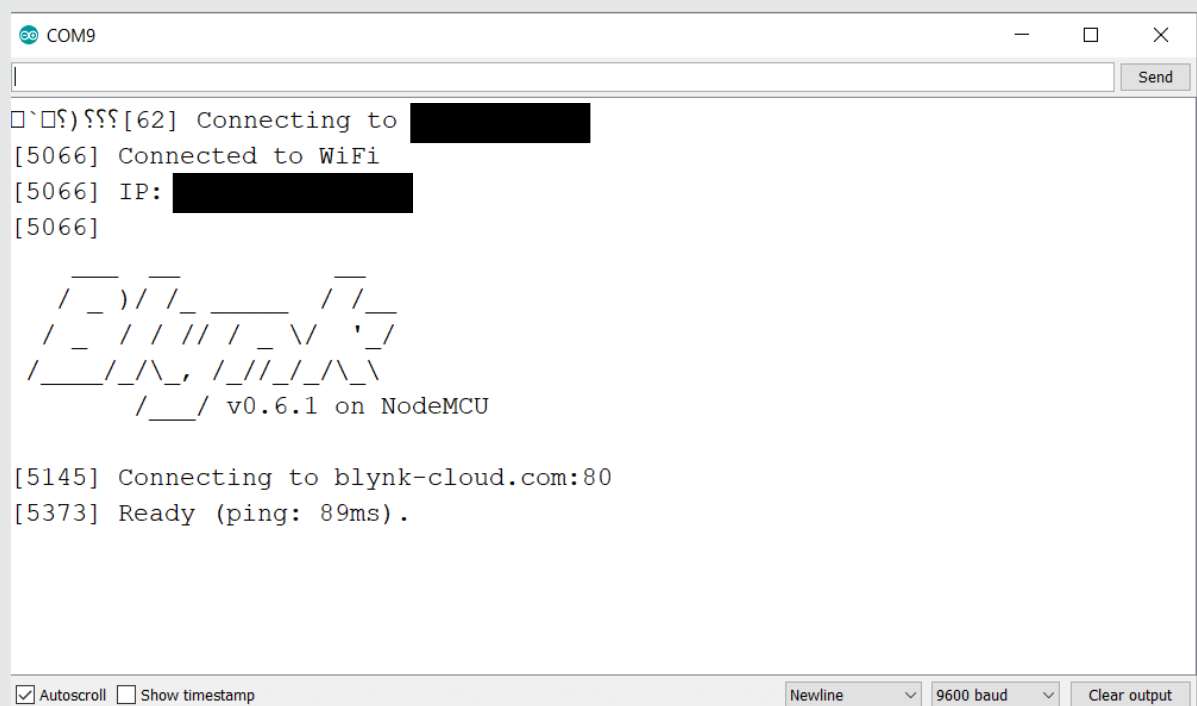
void loop()
{
  Blynk.run();

  // ລະບົບ Blynk ເລີ່ມການເຮັດວຽກ
}

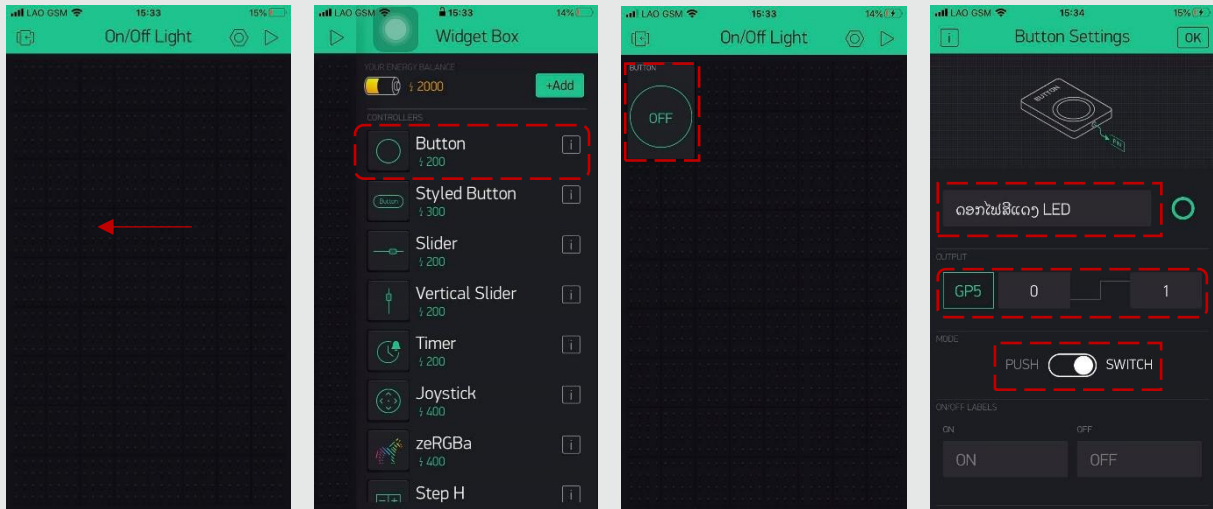
```



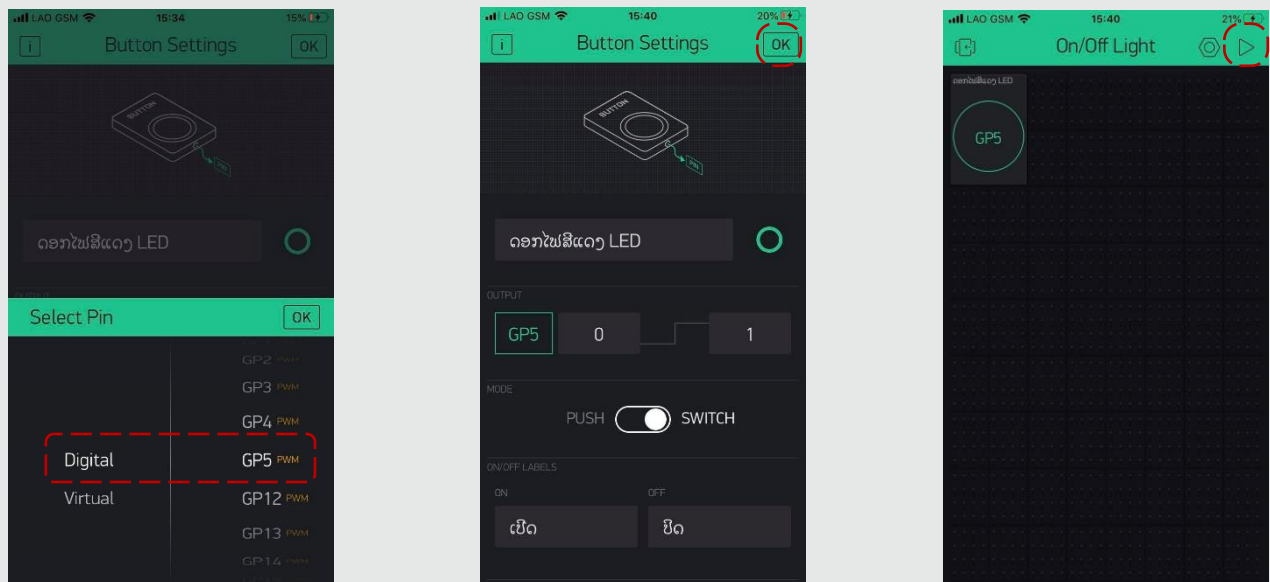
7. ພາຍຫລັງແກ້ໄຂໂຄດ ແລ້ວເພີ່ມ Token ເຂົ້າໄປເປັນຮຽບຮ້ອຍ ຕໍ່ມາແມ່ນໃຫ້ໂຫລດໂປຣແກມລົງໄປທີ່ ບອດຫລືອຸປະກອນ IoT
8. ເປີດຫນ້າຕ່າງ Serial Monitor ຂຶ້ນມາກວດສອບວ່າເຊື່ອມຕໍ່ແລ້ວບໍ່ ຖ້າສາມາດເຊື່ອມຕໍ່ກັບອິນເຕີເນັດ ຜ່ານທາງ WiFi ແລະສາມາດເຊື່ອມຕໍ່ກັບ Blynk Server ໄດ້ແລ້ວ ຈະໄດ້ຮັບຫມາຍເລກ IP Address ແລະ ຂໍ້ຄວາມ Ready ດັ່ງຮູບ ເປັນອັນວ່າເຮົາສາມາດເຊື່ອມຕໍ່ໄປຍັງອຸປະກອນ IoT ຜ່ານທາງ Blynk Server ໄດ້ແລ້ວ



ສ້າງສ່ວນຄວບຄຸມ ແລະສະແດງຜົນໃນ Blynk App ເພື່ອໃຊ້ງານມືດັ່ງນີ້:

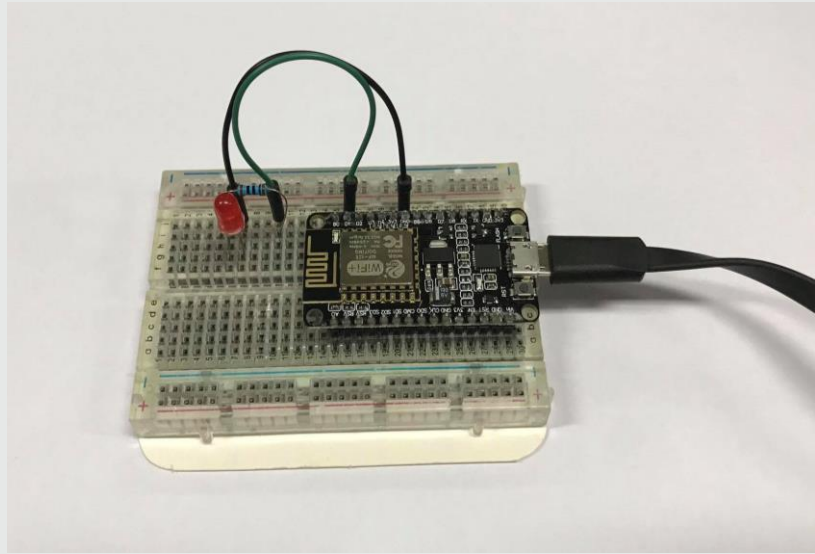
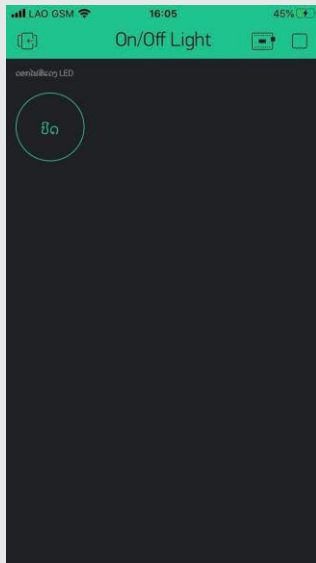


1. ເຂົ້າໄປທີ່ App ທີ່ເຮົາໄດ້ສ້າງໄວ້
2. ປັດມາທາງຊ້າຍ ເລືອກທີ່ Button ກົດເພື່ອເອີ້ນໃຊ້
3. ກົດເຂົ້າໄປທີ່ Button ທີ່ເຮົາໄດ້ເອີ້ນອອກມາ
4. ຕັ້ງຊື່ໃຫ້ອັບດອກໄຟ ແລະ ເລືອກ MODE ເປັນ Switch

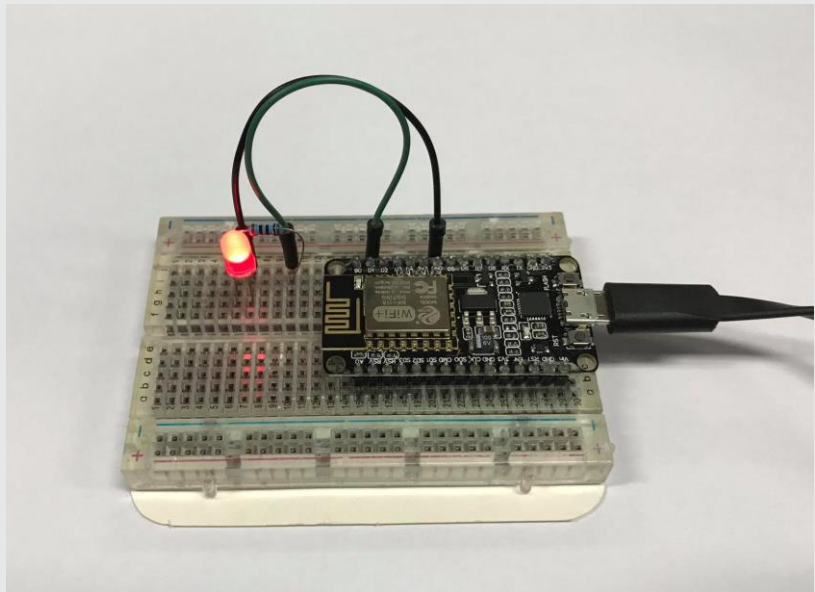
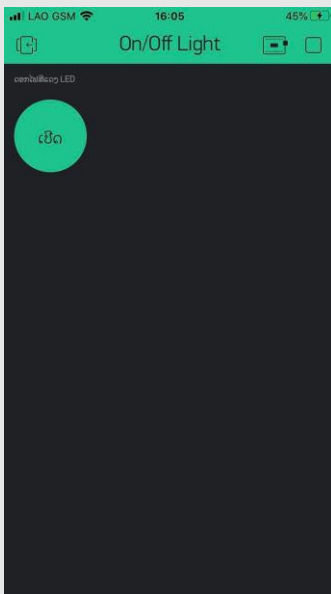


5. ເລືອກຂາ Pin ເປັນ Digital GP5 ແມ່ນຂາທີ່ເຮົາໄດ້ຕໍ່ສາຍໃສ່ກັບບອດ ແລະດອກໄຟ
6. ຕັ້ງຊື່ໃຫ້ກັບ ON/OFF LABS ເປັນ ເປີດ ແລະ ປິດ ພາຍຫລັງຕັ້ງຄ່າແລ້ວໃຫ້ກົດປຸ່ມ OK
7. ຫຼັງຈາກນັ້ນໃຫ້ກົດລົ້ນໄປຮແກມ

ເມື່ອກົດ ປິດ

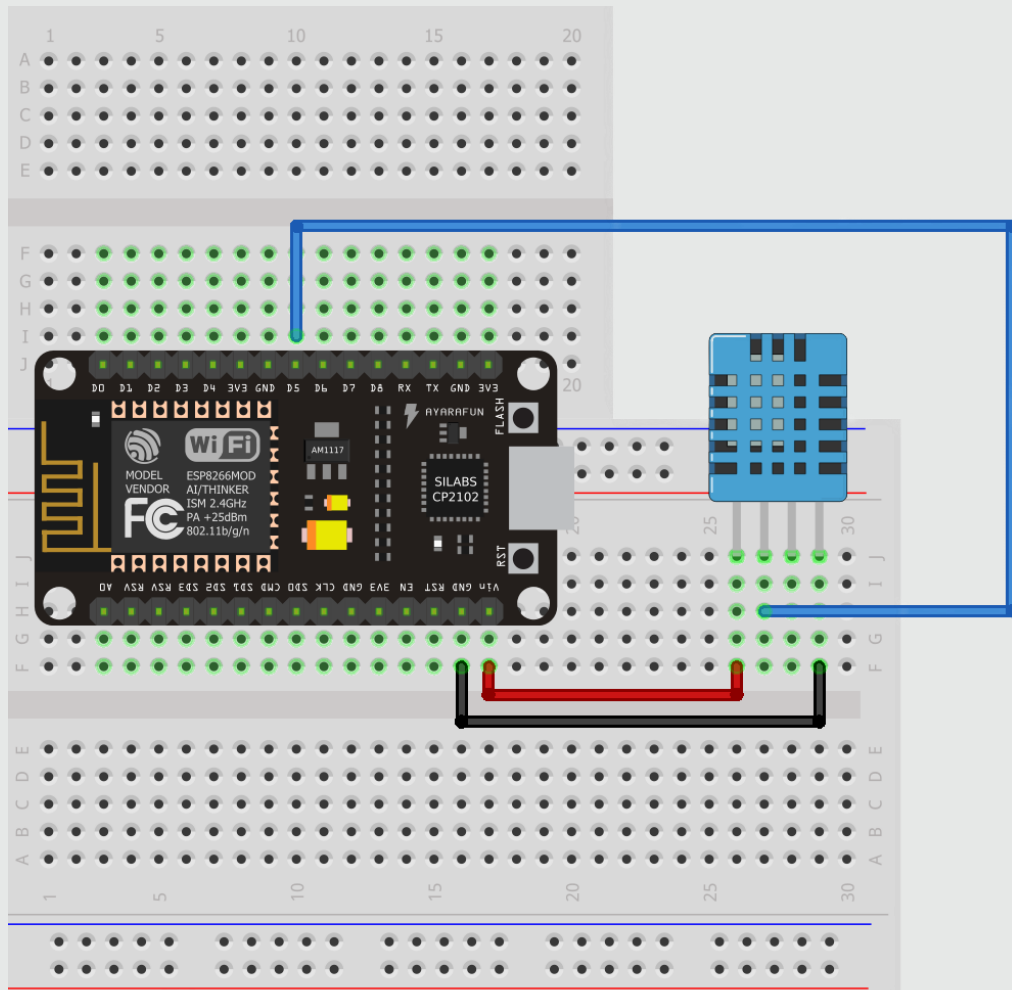


ເມື່ອກົດ ເປີດ



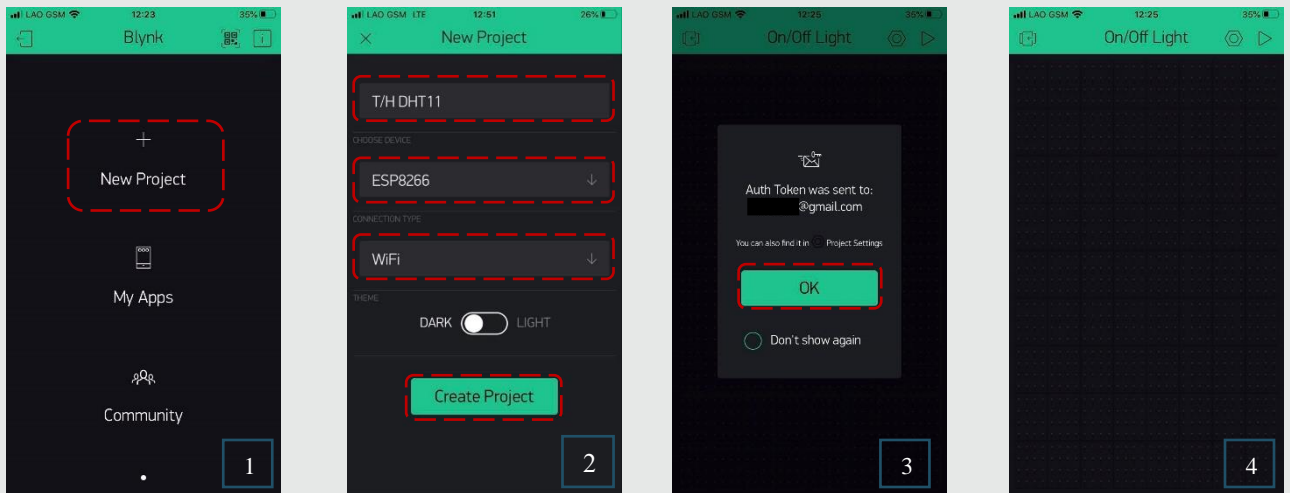
II. ໃຊ້ງານ Blynk ກັບ ເຊັນເຊີ (Sensor) ດ້ວຍການສ້າງໂປຣແກຣມໃໝ່ ແລະທົດລອງເຊື່ອມຕໍ່

ຕົວຢ່າງ: ວົງຈອນວັດອຸນຫະພູມ (Temperature) ແລະ ຄວາມຊຸ່ມຂອງອາກາດ (Humidity)

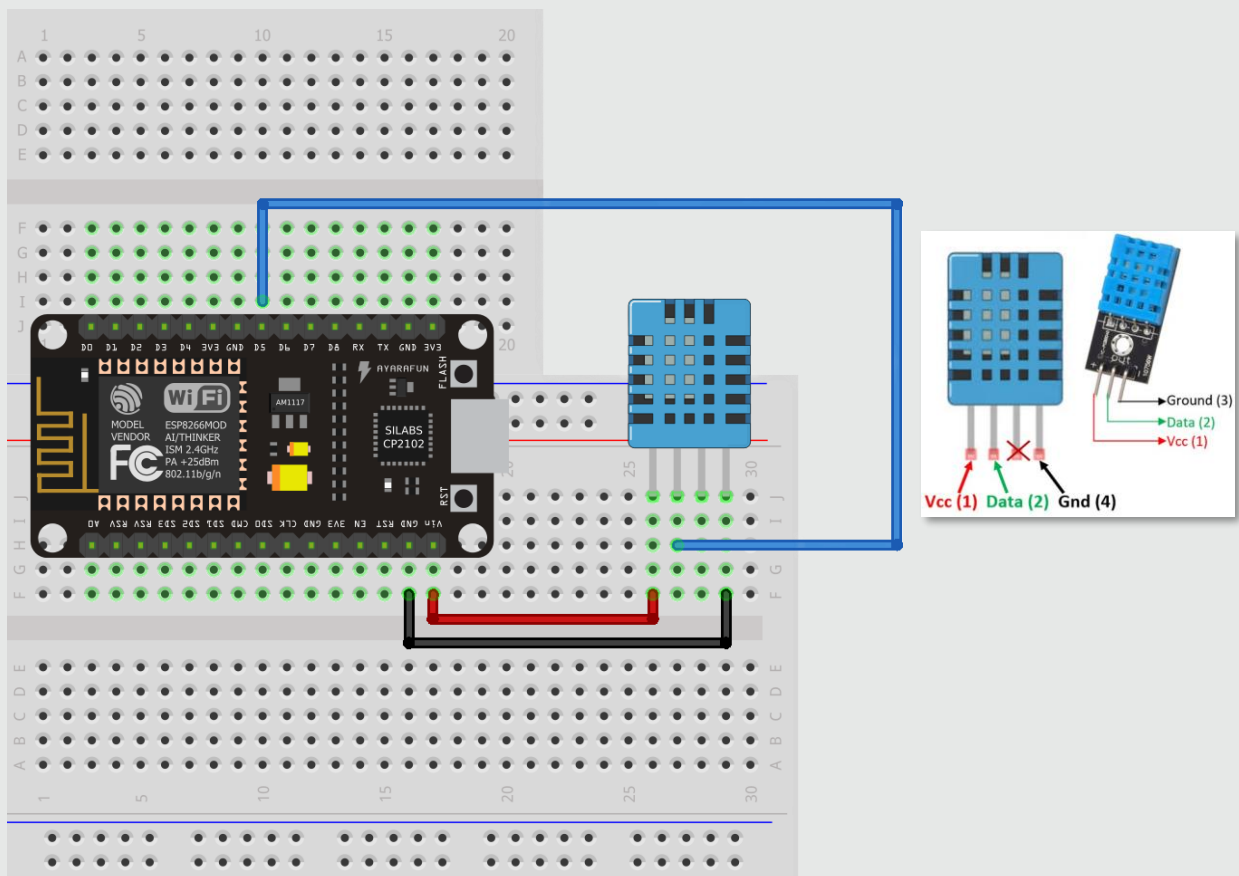


ວິທີການຕໍ່ວົງຈອນ ແລະຂັ້ນຕອນການຂຽນໂປຣແກມມີດັ່ງນີ້:

1. ເມື່ອ Login ເຂົ້າມາໄດ້ແລ້ວ ໃຫ້ກົດທີ່ +New Project ເພື່ອເລີ່ມຕົ້ນສ້າງໂປຣແກຣມ
2. ໃນຫນ້າຕ່າງ New Project ໃຫ້ເຮົາຕັ້ງຊື່ໂປຣແກຣມ (ໃນທີ່ນີ້ຕັ້ງຊື່ວ່າ T/H DHT11), ເລືອກອຸປະກອນທີ່ໃຊ້ (ໃນທີ່ນີ້ໃຊ້ບອດ NodeMCU ESP8266 ດັ່ງນັ້ນເລືອກ ESP8266), ເລືອກຊະນິດເຊື່ອມຕໍ່ (ໃນທີ່ນີ້ເລືອກ WiFi) ຈາກນັ້ນກົດປຸ່ມ Create Project
3. ລະບົບຈະບັນທຶກຂໍ້ມູນຜ່ອມທັງສ້າງລະຫັດ Token ໃຫ້ ແລະສິ່ງລະຫັດ Token ນີ້ ໄປຫາອີເມວທີ່ສະຫມັກຫລື Login ເຂົ້າມາ (ເຮົາຈະຕ້ອງນຳເອົາລະຫັດທີ່ວ່ານີ້ໄປໃສ່ໃນໂຄດໂປຣແກມເຊິ່ງຈະດຳເນີນການໃນພາຍຫລັງ) ກົດປຸ່ມ OK
4. ຕໍ່ມາກໍຈະເຫັນເປັນພື້ນທີ່ຫວ່າງເປົ່າ



5. ລົງມືຕໍ່ວົງຈອນວັດອຸນຫະພູມ (Temperature) ແລະ ຄວາມຊຸ່ມຂອງອາກາດ (Humidity) ດັ່ງຮູບ (ໃນທີ່ນີ້ເລືອກໃຊ້ຂາ D5 ຫລື GPIO14 ເປັນຂາອິນພຸດ)



6. ເປີດ Arduino IDE ເພື່ອຂຽນໂຄດລົງໄປໃນບອດ NodeMCU ESP8266

```

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
//ເລືອກໃຊ້ Libraries DHT.h

char auth[] = "YourAuthToken";
// Copy Token ທີ່ໄດ້ຮັບຈາກອີເມວກ່ອນໜ້ານີ້ (ຫຼັງຈາກສ້າງ New Project ແລ້ວ) ມາໃສ່ໃນ “ ”
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";
#define DHTPIN 14
//ກຳນົດໃຫ້ຂາ D5/GPIO14 ເປັນຂາອິນພຸດທີ່ຮັບຂໍ້ມູນມາຈາກຂາ Data Out ຂອງໂມດູນ DHT11 ມາເກັບໄວ້
ໃນຕົວປ່ຽນ DHTPIN

#define DHTTYPE DHT11
// ກຳນົດໃຫ້ຕົວປ່ຽນ DHTTYPE ໃຫ້ເກັບຊະນິດຂອງໂຕເຊັນເຊີ

DHT dht(DHTPIN, DHTTYPE);

// ສ້າງອອບເຈັກຈາກຄາດ DHT ແລ້ວນຳໄປເກັບໄວ້ໃນຕົວປ່ຽນ dht ໂດຍໃຊ້ຄ່າທີ່ເກັບໄວ້ໃນຕົວປ່ຽນ DHTPIN
ແລະ DHTTYPE

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  // ເລີ່ມຕົ້ນລະບົບການເຊື່ອມຕໍ່ກັບ WiFi Network ແລະ Blynk Server

  dht.begin();
  // ເລີ່ມຕົ້ນໃຊ້ງານເຊັນເຊີ DHT11
}

```

```

void loop()
{
  Blynk.run();
  // ລະບົບ Blynk ເລີ່ມການເຮັດວຽກ

  delay(2000);

  float t = dht.readTemperature();
  // ໂຕປ່ຽນ t ຊະນິດ float ໃຊ້ເກັບຄ່າອຸນຫະພູມທີ່ໄດ້ອ່ານຈາກເຊັນເຊີ

  float h = dht.readHumidity();
  // ໂຕປ່ຽນ h ຊະນິດ float ໃຊ້ເກັບຄ່າອຸນຫະພູມທີ່ໄດ້ອ່ານຈາກເຊັນເຊີ

  if (isnan(t) || isnan(h)) {
    // ຖ້າບໍ່ສາມາດອ່ານຄ່າອຸນຫະພູມຫລືຄວາມຊຸ່ມຂອງອາກາດທີ່ເກັບໄວ້ໃນໂຕປ່ຽນໄດ້ ແມ່ນໃຫ້ສະແດງ "DHT
    Failed!" ອອກທາງ Serial Monitor

    Serial.print("DHT Failed!");
    return; // ກັບໄປອ່ານຄ່າມາໃໝ່ ຈົນກວ່າອ່ານຄ່າໄດ້
  }

  Blynk.virtualWrite(V1, t);
  // ນຳຄ່າອຸນຫະພູມຈາກຕົວປ່ຽນ t ໄປເກັບໄວ້ໃນ V1 ແລ້ວສົ່ງໄປຫາ Blynk Server

  Blynk.virtualWrite(V2, h);
  // ນຳຄ່າເປີເຊັນຄວາມຊຸ່ມຈາກຕົວປ່ຽນ h ໄປເກັບໄວ້ໃນ V2 ແລ້ວສົ່ງໄປຫາ Blynk Server

  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.print("°C | ");
  Serial.print("Humidity ");
  Serial.print(h);
  Serial.println("%");
  delay(1000);
}

```

Auth Token for T/H DHT11 project and device T/H DHT11

ກຳລັງຈັດໝາຍ ✕



Blynk <dispatcher@blynk.io> ຍຸດເລີກກາຣສົມຄຸນ

ດຶງ ຂໍ້ມູນ ▼

Auth Token : eeL3DaDdhaVT8yn9pb3CJCSIm-gS3vhH

Happy Blynking!

-

Getting Started Guide -> <https://www.blynk.cc/getting-started>

Documentation -> <http://docs.blynk.cc/>

Sketch generator -> <https://examples.blynk.cc/>

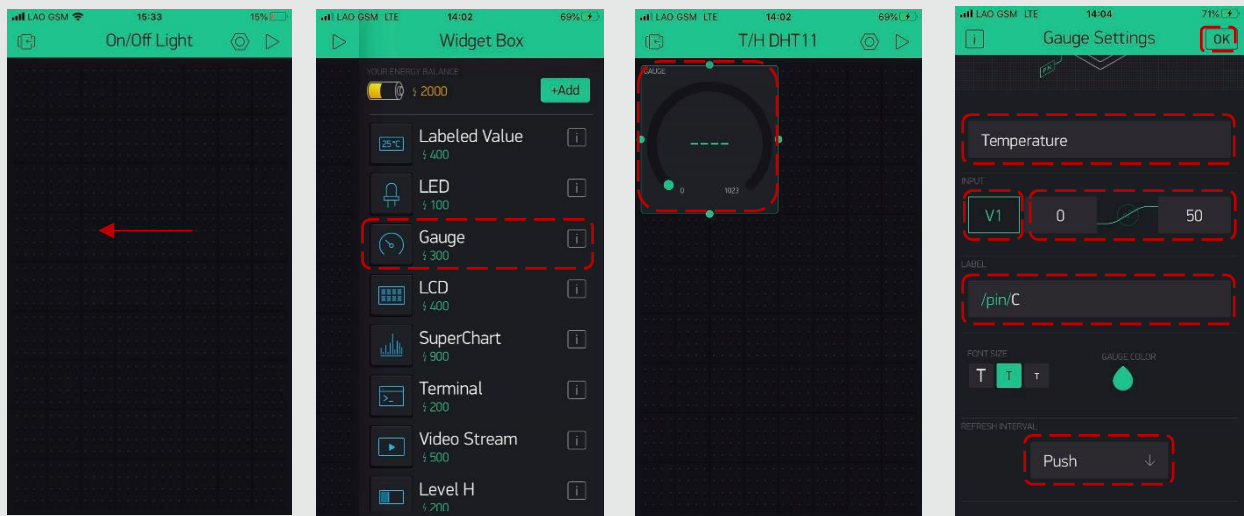
7. ພາຍຫລັງແກ້ໄຂໂຄດ ແລ້ວເພີ່ມ Token ເຂົ້າໄປເປັນຮູບຮ້ອຍ ຕໍ່ມາແມ່ນໃຫ້ໂຫລດໂປຣແກມລົງໄປທີ່ ບອດຫລືອຸປະກອນ IoT
8. ເປີດຫນ້າຕ່າງ Serial Monitor ຂຶ້ນມາກວດສອບວ່າເຊື່ອມຕໍ່ແລ້ວບໍ່ ຖ້າສາມາດເຊື່ອມຕໍ່ກັບອິນເຕີເນັດ ຜ່ານທາງ WiFi ແລະສາມາດເຊື່ອມຕໍ່ກັບ Blynk Server ໄດ້ແລ້ວ ຈະໄດ້ຮັບໝາຍເລກ IP Address ແລະ ຂໍ້ຄວາມ Ready ດັ່ງຮູບ ເປັນອັນວ່າເຮົາສາມາດເຊື່ອມຕໍ່ໄປຍັງອຸປະກອນ IoT ຜ່ານທາງ Blynk Server ໄດ້ແລ້ວ

```
COM9
[5066] Connecting to [redacted]
[5066] Connected to WiFi
[5066] IP: [redacted]
[5066]

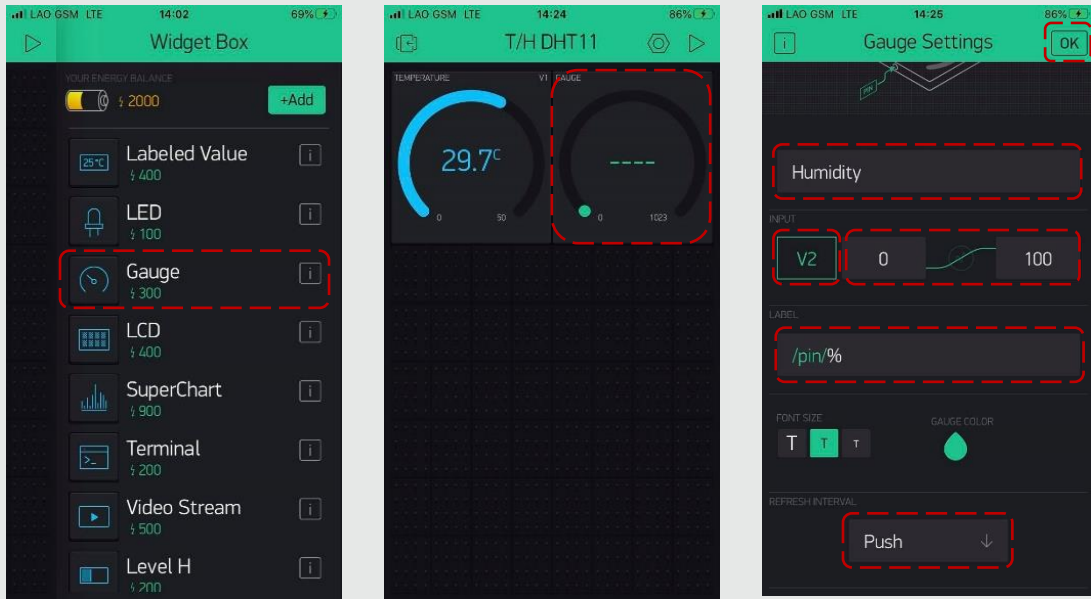
  _ _ _ _ _
 / _ ) / / _ _ _ _ _ / / _
 / _ / / / / _ _ _ _ _ \ / '
 / _ _ / _ \ _ _ _ _ _ \ \
      / _ _ / v0.6.1 on NodeMCU

[5145] Connecting to blynk-cloud.com:80
[5373] Ready (ping: 89ms).
```

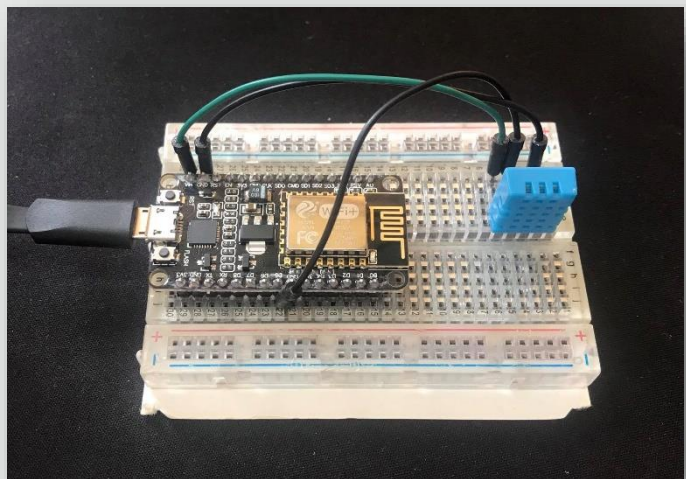
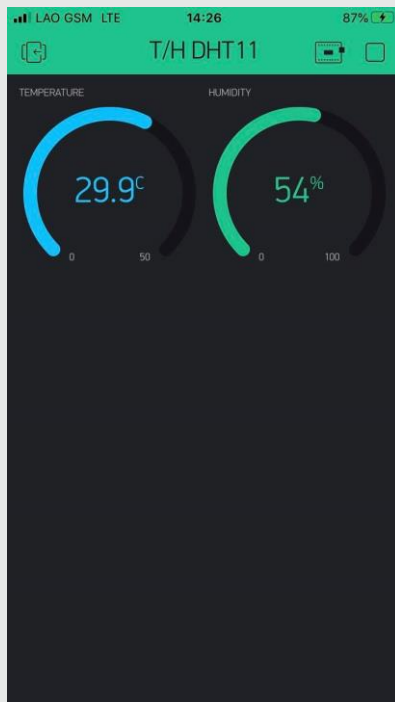
ສ້າງສ່ວນຄວບຄຸມ ແລະສະແດງຜົນໃນ Blynk App ເພື່ອໃຊ້ງານມືດັ່ງນີ້:



1. ເຂົ້າໄປທີ່ App ທີ່ເຮົາໄດ້ສ້າງໄວ້
2. ປັດມາທາງຊ້າຍ ເລືອກທີ່ Gauge ກົດເພື່ອເອົາໃຊ້
3. ຫຼັງຈາກນັ້ນກົດໄປທີ່ Gauge ເພື່ອເຂົ້າໄປຕັ້ງຄ່າການເຮັດວຽກ
4. ກົດປ່ຽນຊື່ເປັນ Temperature, ໄປທີ່ Input ກົດ PIN ແລ້ວເລືອກ Virtual V1 ຕໍ່ມາໄປທີ່ກຳນົດຄ່າ Min ແລະ Max (ໃນທີ່ນີ້ເຮົາຈະຕັ້ງເປັນ 0-50), ໄປທີ່ LABEL ເພື່ອໃສ່ຂໍ້ຄວາມ ຕົວອັກສອນຫຼື ສັນຍະລັກຕ່າງໆ (ໃນທີ່ນີ້ເຮົາຈະໃສ່ເປັນ C) ແລະໄປທີ່ REFRESH INTERVAL ເພື່ອເລືອກອັດຕາຫຼື ໄລຍະຫ່າງຂອງເວລາໃນການຮິດຂໍ້ມູນໃໝ່ (ໃນທີ່ນີ້ຈະເລືອກເປັນ Push) ພາຍຫລັງຕັ້ງຄ່າແລ້ວໃຫ້ກົດປຸ່ມ OK



5. ຫຼັງຈາກນັ້ນກົດໄປທີ່ Gauge (ໂຕທີ2) ເພື່ອເຂົ້າໄປຕັ້ງຄ່າການເຮັດວຽກ
6. ກົດປ່ຽນຊື່ເປັນ Humidity, ໄປທີ່ Input ກົດ PIN ແລ້ວເລືອກ Virtual V2 ຕໍ່ມາໄປທີ່ກຳນົດຄ່າ Min ແລະ Max (ໃນທີ່ນີ້ເຮົາຈະຕັ້ງເປັນ 0-100), ໄປທີ່ LABEL ເພື່ອໃສ່ຂໍ້ຄວາມ ຕົວອັກສອນຫຼືສັນຍະລັກ ຕ່າງໆ (ໃນທີ່ນີ້ເຮົາຈະໃສ່ເປັນ %) ແລະໄປທີ່ REFRESH INTERVAL ເພື່ອເລືອກອັດຕາຫຼືໄລຍະຫ່າງ ຂອງເວລາໃນການຮິຟເຟດຂໍ້ມູນໃໝ່ (ໃນທີ່ນີ້ຈະເລືອກເປັນ Push) ພາຍຫລັງຕັ້ງຄ່າແລ້ວໃຫ້ກົດປຸ່ມ OK
7. ຫຼັງຈາກນັ້ນໃຫ້ກົດລັ່ນໂປຣແກມ



ຮູບການຕໍ່ວົງຈອນວັດອຸນຫະພູມ (Temperature) ແລະ ຄວາມຊຸ່ມຂອງອາກາດ (Humidity)

ສໍາລັບ E-Book ວິທີການນໍາໃຊ້ NodeMCU ໃນການເຊື່ອມຕໍ່ ແລະ ຄວບຄຸມອຸປະກອນ IoT ຜ່ານ
ອິນເຕີເນັດ ດ້ວຍ Mobile Apps & Blynk Server ສະບັບພາສາລາວກໍ່ຈະອະທິບາຍ ແລະສອນພຽງເທົ່ານີ້ອາດ
ຈະບໍ່ມີຄວາມລະອຽດ 100% ກໍ່ຕ້ອງຂໍອະໄພ 😊

ໂດຍ: ພອນປະເສີດ ສີໃສແກ້ວ

ນັກສຶກສາ: ມະຫາວິທະຍາໄລ ແຫ່ງຊາດລາວ National University of Laos; ຄະນະວິສະວະກໍາສາດ Faculty
of Engineering; ພາກວິຊາ ວິສະວະກໍາຄອມພິວເຕີ ແລະ ເຕັກໂນໂລຊີຂໍ້ມູນຂ່າວສານ Department of
Computer Engineering and Information Technology; ສາຂາ ວິສະວະກໍາຄອມພິວເຕີ Major of
Computer Engineering.

Facebook: Phonpaserth SISAYKEO

E-Mail: Phonpaserth.ssk@gmail.com

Tel: 020 56599929

ກວດແກ້ໂດຍ: ອຈ. ແສງລັດສະໝີ ຈັນທະມົນາວົງ