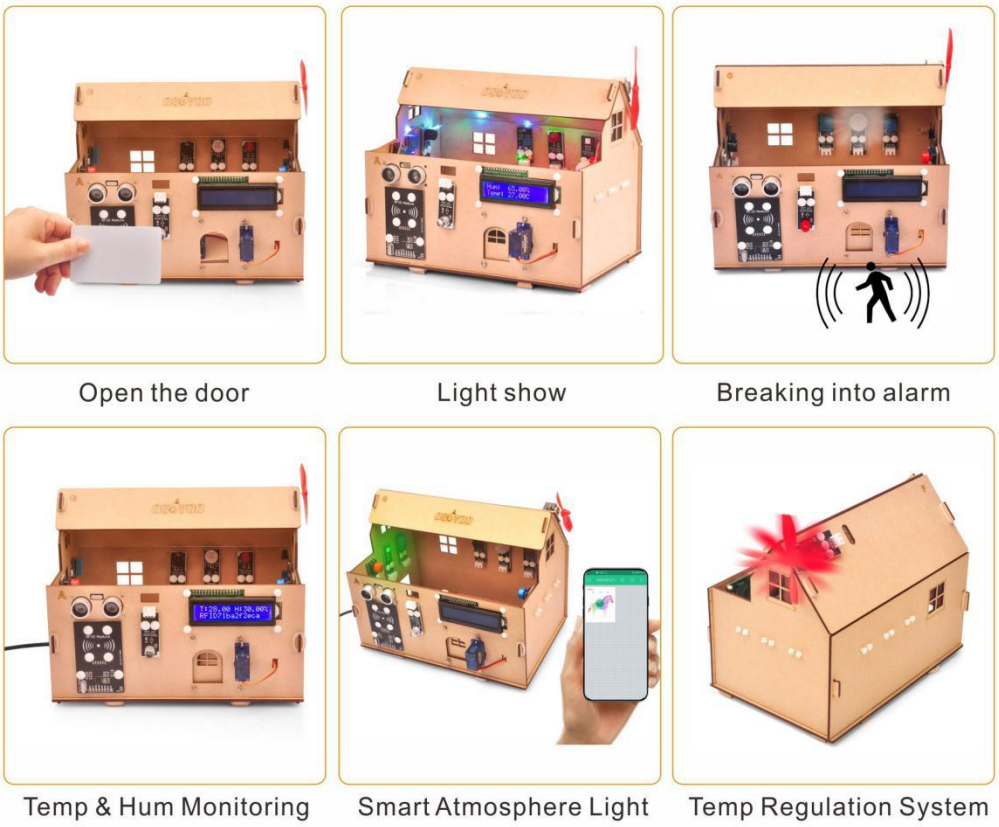
Table of Contents

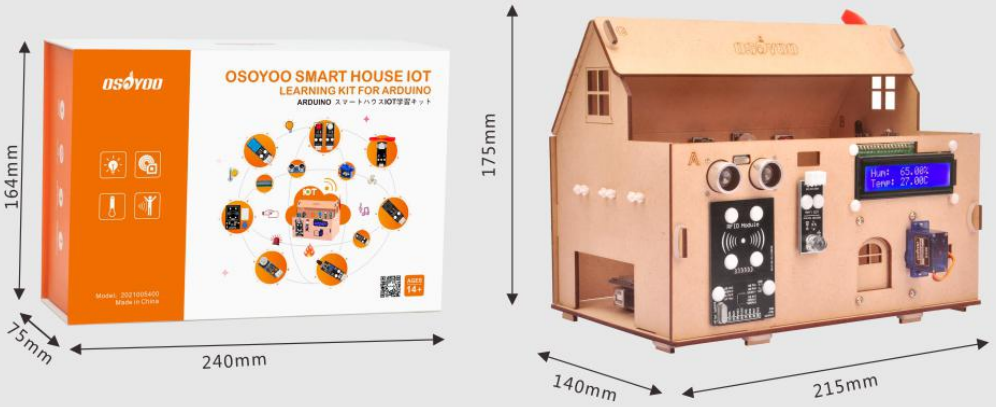

OSOYOO SMART HOUSE IOT
LEARNING KIT FOR ARDUINO


osoyoo smart house iot
learning kit for arduino

Curriculum experiment

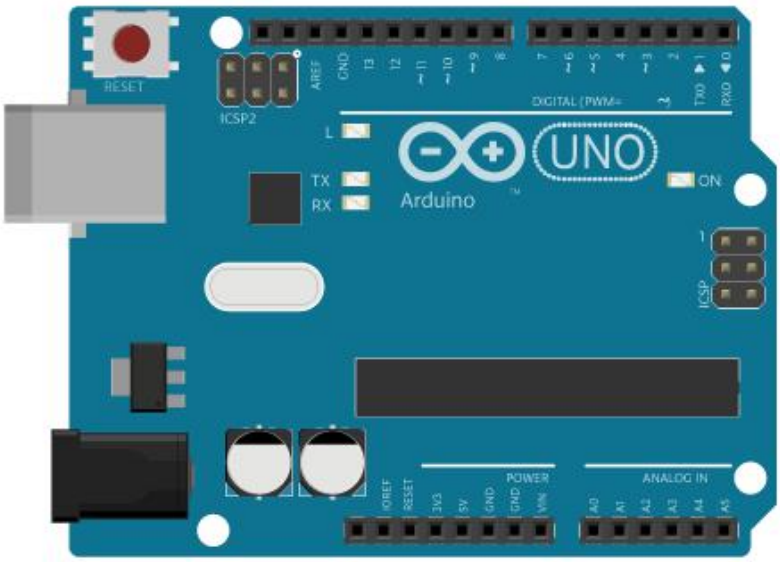


product parameters



1-1 What is Arduino?

What is Arduino?



Arduino is basically an open source electronics archetype platform for electronics engineers, hobbyist, designers or anyone interested in creating interactive electronics projects. It is a flexible platform and based on an easy to use software and hardware systems. Arduino comprises of a microcontroller and a software or **Integrated Development Environment (IDE)** that runs on laptops or computers, used for writing and uploading computer codes or programs to the physical board.

The Arduino boards are able to read inputs – light, proximity or air quality on a sensor, or an SMS or Twitter message – and turn it into an output – activating a motor, turning on a light, publishing content online or trigger external events. You can tell your board what to do by writing code and uploading it to the microcontroller on it using the **Arduino programming language** (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has powered thousands of projects. Arduino has gathered around a community where beginners and experts from around the world share ideas, knowledge and their collective experience. There are

thousands of makers, students, artists, designers, programmers, researchers, professionals and hobbyists worldwide who use Arduino for learning, prototyping, and finished professional work production.

Arduino was born at the Interaction Design Institute Ivrea IDII from the Wiring project as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. The main objective of both projects is to make the process of working with technology and electronics easier. The Arduino board has evolved to adapt to new needs ranging from simple 8-bit boards to products ready for IoT applications. **All Arduino boards are completely open-source**, empowering users to build them independently and eventually adapt them to their particular needs. **The software is open-source, and it is growing through the contributions of developers and the Arduino community worldwide**.

There have been many similar projects, but none of them succeeded as well as Arduino has, due to how easy it is to use the software, and the affordability of the hardware. **The Arduino software is easy-to-use for beginners**, yet flexible enough for advanced users needs. It runs on Mac, Windows, and Linux.

Why choose Arduino?

Today there are many different types of microcontrollers are available in the market. So why choose Arduino? It's an important question… Some points for Why choose arduino are given below.

- Unlike other programmable circuit boards or microcontroller board, arduino doesn't require a separate hardware or programmer parts to load a code on to the board, instead of these you just need a USB cable.
- Furthermore, the arduino IDE utilizes a simplified sort of C++, so you can easily learn the programming sections.
- Arduino supports a standard form factor.
- Working with arduino does not require any previous experience.
- For amateurs, Arduino offers a wonderful platform for their projects ideas.
- Inexpensive compared to similar boards.
- Simplicity of ardunio compared to others.
- It is an open source platform. So any one can modify and build your own Arduino board.

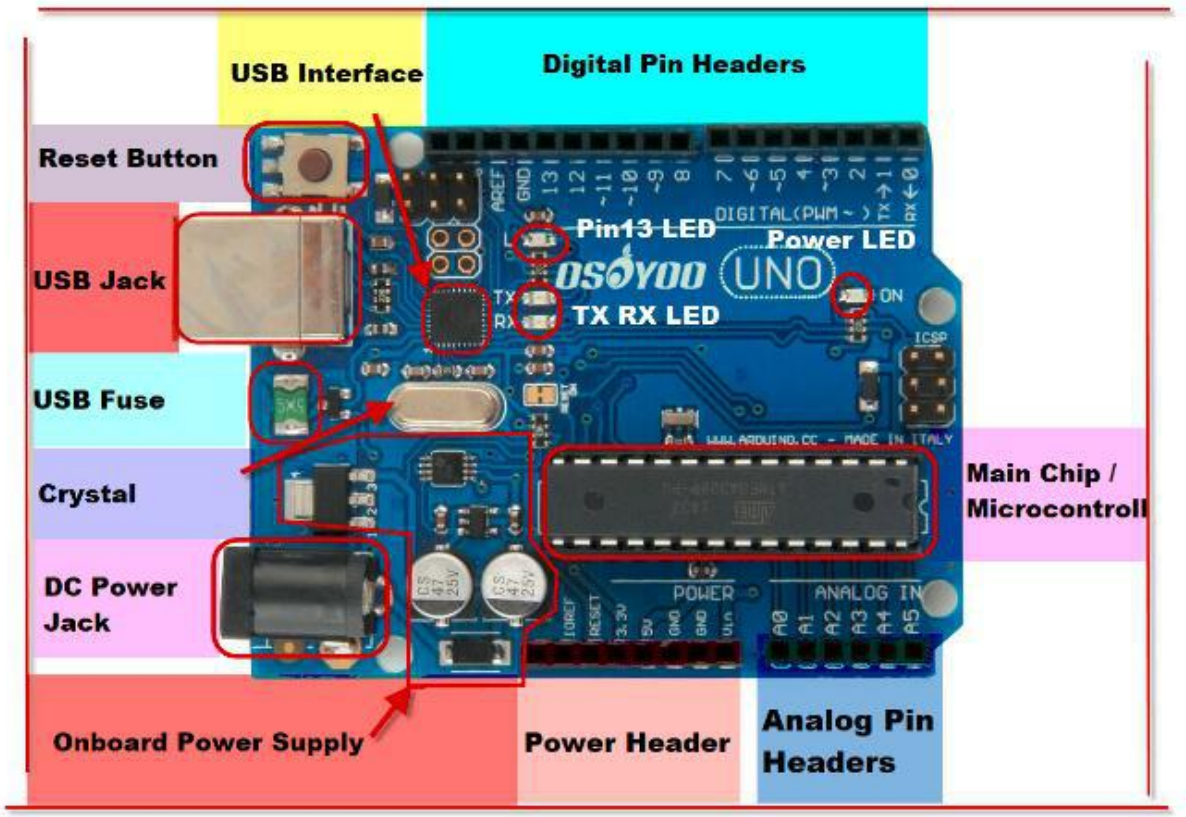
- Arduino is based on Atmel's ATMEGA168 and ATMEGA8 microcontrollers.
- Experts can easily make improvements and extensions on to the arduino board.

Types of Arduino

Arduino has many boards it starts from basic Arduino UNO and goes to Arduino mega, ArduinoFio, lily pad so on and so forth.

If you need more info you can compare the specs of each board here.
If you are wondering if your Arduino board is authentic you can learn how to spot a counterfeit board here.

What is Inside an Arduino / What's on the board?



Don't feel like you have to understand this part fully! Skim it for now,
and consider it a resource for you when you want to take a deeper dive
into understanding the hardware!

MAIN CHIP / MICROCONTROLLER

The black thing with all the metal legs is an IC, or Integrated Circuit **(13)**. Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.

NOTE: Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

ONBOARD POWER SUPPLY

The Arduino is designed for beginners so it has some protection and regulation circuitry so that it can use just about any power supply you throw at it. In particular there is a polarity protection diode (to avoid destroying the board if you have a Negative Tip adapter). It also has an **onboard 5V**

USB JACK & INTERFACE

USB Jack

As we talked about in the beginning, this is how you connect your Arduino to your computer. You can use any computer with a USB port. You will need a cable to connect! This cable is usually included in the OSOYOO pack.

USB Inteface Chip

OK so you plug your Arduino into a computer with a USB cable. But you may be surprised to learn, the main processor chip (ATmega328) cannot speak "USB". Instead it can talk an interface language called "Serial". Serial is a much simpler, much older interface. (It's also a lot less expensive to build into a chip) So, how do you connect a chip that does not speak USB to a USB port? Easy! you just need a **USB to Serial Interface Translator chip**. Much like a human translator, it can understand and speak both languages and can seamlessly translate between the two.

There's a lot of different translator chips, some common part numbers are **FTDI FT232**, **FTDI FT231X, CP2102** or**CP2104**, **PL2303**, **CH430** and probably a dozen others. They're all nearly identical but some require different operating system drivers.

ARDUINO LEDS

Likewise, the Arduino has **four** LEDs: **L**, **RX**, **TX**, and **ON**

ON LED

This LED will shine green whenever the Arduino is powered. Always check this LED if your Arduino is not acting right, if its flickering or off then you should check your power supply.

RX and TX LEDs

These are like the 'send' and 'receive' LEDs on your cable modem. They blink whenever information is sent from or to the Arduino through the USB connection

The **TX** LED lights up yellow whenever data is sent **from the Arduino to the computer** USB port

The **RX** LED lights up yellow whenever data is sent **to the Arduino from the computer** USB port

L LED

This is the one LED that you can control. The ON, RX and TX LEDs all light up automatically no matter what. The **L** LED, however, is connected to the Arduino main chip and you can turn it on or off when you start writing code. For future reference, **L** is connected to Digital Pin #13

POWER HEADER

- **Vin –** This is connected to the power input from the DC Jack, so it is going to range from 7 V to 12 VDC, depending on what is plugged into the DC Jack. If the DC Jack is not powered, it will provide the 5V from the USB connection. Provides as much as the DC power supply can.
- **GND** – You get two of these here, they are the common ground connection for all power and data
- **5V** – This is the clean regulated 5V power that the Arduino runs on, provided from the DC jack (if plugged in) or USB connection (if DC is not plugged in). Provides up to about 500mA current draw.
- **3.3V** – This is a clean regulated 3.3V power, sometimes you'll need exactly this voltage for some sensors. Provides up to about 100mA current draw.
- **Reset** – This is the same pin connected to the reset button
- **IOref** – Used by shields to know what the IO voltage is. You can ignore this pin.
- **Unnamed pin** – Reserved for future use, don't connect to it!

DIGITAL PIN HEADERS

The two pins labeled **0 (RX)** and **1 (TX)** are the two Serial pins that are used to send data to and from the Arduino to the USB-Serial translator chip.
Don't connect anything to Digital 0 or 1 unless you are super sure because it will affect your Arduino's ability to communicate!

- **Digital 2** through **Digital 12** are normal every day digital pins.
- **PWM Pins:** In the board, there is a ~ sign near to some of the pins – 3, 5, 6, 9, 10(digital pins) and 11 on the UNO Arduino. Actually these pins are normal digital pins but it can be used for Pulse-Width Modulation (PWM) also.
- **Digital 13** is a little special because it is also connected to the **L LED**. You can use this pin without affecting the Arduino just be aware that the **L** LED will also blink at the same time.

And a few extra straggler pins:

- A spare power **GND** Ground pin
- **AREF** – **A**nalog **Ref**erence pin. Used for advanced analog sensor reading (You'll learn about this later)
- Two unlabeled pins (the labels are on the bottom). These are the **SDA** and **SCL** pins, which are used for connecting I2C type sensors. They are connected inside the PCB to **A5** and **A4** We do not recommend usin these unless you have an I2C sensor

ANALOG PIN HEADERS

Shh! It's a secret but those 6 analog input pins? They can also be used as digital input/output pins, they really are the most versatile pins!
Each analog pin can read a voltage between 0 and 5 V (the same voltage used to power the Arduino.
Once you get advanced analog skills you can connect the **ARef** pin to a different voltage like 3.3V and direct the Arduino to use Aref as the max voltage, then you can get more precision. But we'll cover that some other day.
Do not connect a voltage higher than 5V to the analog input pins or you could damage them!

USB FUSE

The little USB fuse is a part that is used to protect your Arduino and computer. You'll be connecting all sorts of wires to your Arduino and there's a chance you will accidentally short out the power. To keep your electronics safe this resettable fuse will open up, much like the circuit breakers in your home.

Reset Button

Just like the original Nintendo, the Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND (3)**: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V (4) & 3.3V (5)**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog (6)**: The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

- **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

What is the Arduino IDE?

Arduino provides an open-source and easy-to-use programming tool for writing code and uploading it to your board. It is often referred to as the Arduino IDE (Integrated Development Environment). The **Arduino Software (IDE) is easy-to-use for beginners**, yet flexible enough for advanced users.

1-2 OSOYOO MEGA-IoT Shield

WHAT IS OSOYOO MEGA-IOT SHIELD?

This ESP8266 is an ultra-low power UART WiFi module. The OSOYOO MEGA-IoT Shield is based on the ESP-12s with the ESP8266 WiFi chip. This shield is designed to easily connect your Arduino to the IoT world.

It is a shield based on Arduino, which integrates famous ESP-12S WiFi module and provides low cost WiFi solution with any MEGA2560 projects. It is compatible with Arduino Mega2560, and possibly other pin compatible main boards.

Because of IoT (Internet of Thing), everything should be connected to Internet, a crucial element is WiFi that provides wireless connection to Internet. We have been carrying ESP module and many have used it. However it is not user friendly enough for beginner. The module is 3.3V powered and the pins are not properly broken out. Intergrating it to your microcontroller e.g.: Arduino will require some skill and electronics interface.

With this OSOYOO MEGA-IoT Shield, it will be plug and use for Arduino user. The shield come preassembled with ESP12S module which offers WiFi connection to your Arduino MEGA2560 board or project. No additional soldering or wiring needed, just stack the shield on to Arduino or any compatible Arduino main board, select the pins for serial communication with mini jumpers and the hardware connection is completed! Focus in program development, and the good news, there are plenty of libraries that you can utilize, we will show you the steps!

The ESP12S comes pre-flashed with an AT-command firmware, so it can be controlled by any UART/Serial, but it also breaks out and provides command access to all of the ESP8266 ' s I/O. The shield itself does not limit to AT command, we have breakout most of the IO and you can program the ESP12S module directly. Whether you want to add AT commands of your own, or flash custom firmware on the ESP8266, this feature may come in handy especially with it utilizing the UART pin selectors.

FEATURES

1. Standard pin interface to achieve full compatibility with the pitch Arduino

Uno, Mega and so on. (Unless the standard size Arduino, such as Nano, Mini)

2. 3.3V power supply using advanced chip (with the official expansion board power chip ESP12S same), so 3.3V Arduino board provided more accurate, so ESP8266 as official chip work as normal, there is no gap.

3. It leads to a WiFi module power supply / service interface to facilitate developers to extend the power supply and maintenance.

4. Onboard ESP8266-12 stable industrial grade serial WiFi module, using an enhanced PCB antenna, signal better.

5. You can quickly switch DIP switch ESP8266-12 industrial grade WiFi stable operation mode (such as operating mode and upgrade mode) module, integrated serial chip, quickly flashing the firmware.

6. Built-in 5V to 3.3V TTL level converter to prevent high-pressure Arduino TTL level of the WiFi module damage.

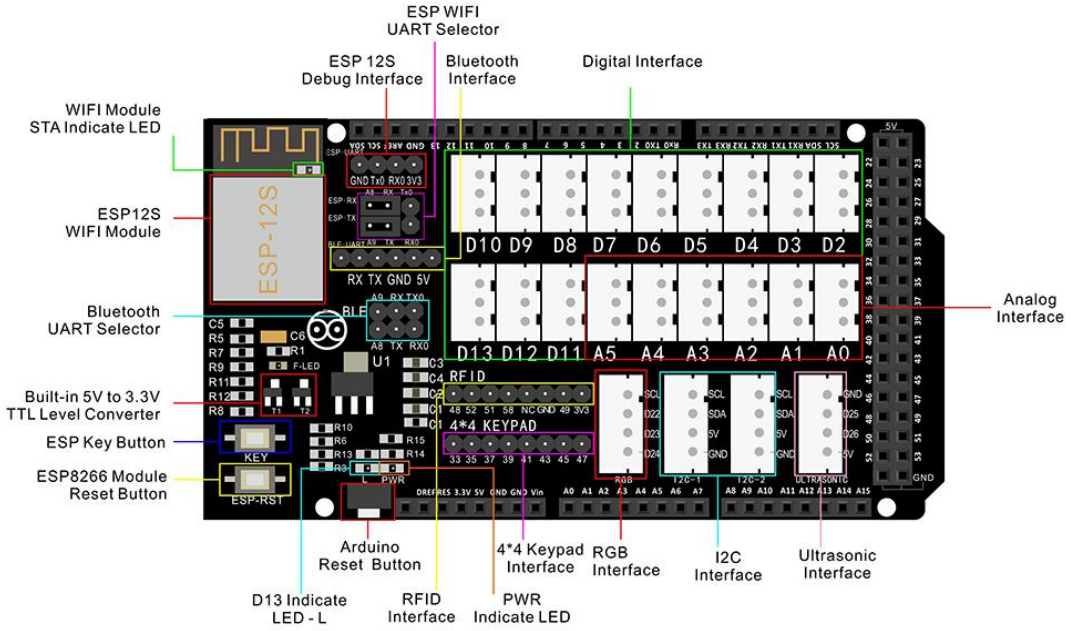
7. Onboard ESP8266-12 stable industrial grade WiFi module with a metal shield, immunity to interference.

8. Onboard ESP8266-12 standard pitch leads to a stable industrial grade WiFi module all interfaces, enabling developers to easily develop deep.

9. Use stackable design, it can continue to accumulate above the other modules, easy to use.

10. Onboard four LED displays the status of ESP8266, namely PWR, L, STA

SPECIFICATIONS



Input Voltage: +5VDC Internal Source+7V to 9VDC[External Source)

ESP8266 Module Supply: +3.3VDC

ESP Shield Default Baudrate Settings are 9600 kbps baudrate, 8 bit data, 1 stop bit and no parity

Onboard ESP12S WiFi module with preloaded AT firmware

Jumper selectors for software or hardware UART and PC-ESP communication.

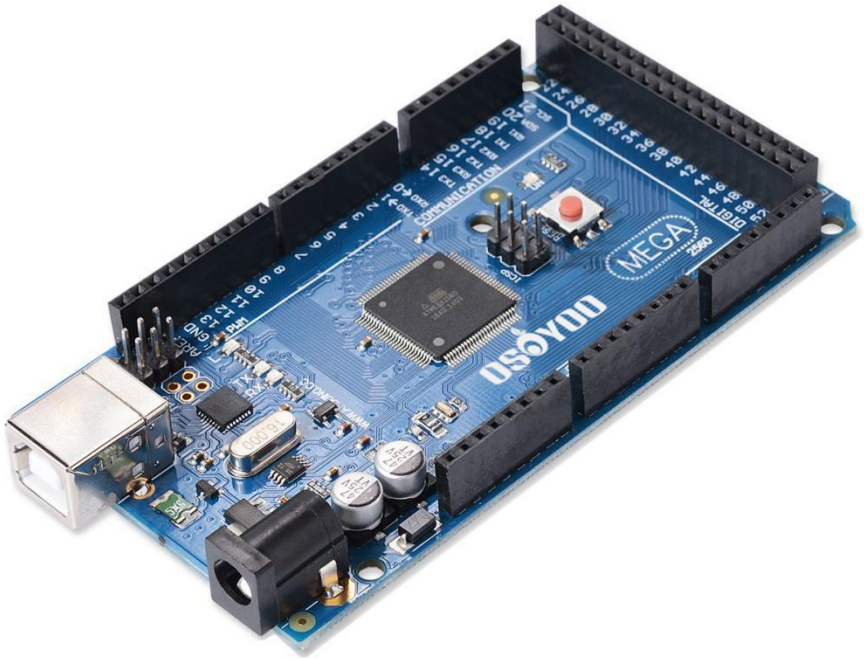
Stackable Digital I/O header: This header pin is Digital I/O pin stacked to the OSOYOO MEGA2560 board. Other Arduino shield can be stacked on top of this stackable header.

Stackable Analog Input pin header: This header pin is Analog Input pin stacked to the OSOYOO MEGA2560 board. Other Arduino shield can be stacked on top of this stackable header.

Serial Selector: User may select A8 as the RX pin, select A9 as the TX pin from OSOYOO MEGA2560 board with the mini jumpers.

1-3 OSOYOO Mega2560 Board — Fully compatible with Arduino Mega2560 Rev.3

“The MEGA 2560 is designed for more complex projects. With 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects. This gives your projects plenty of room and opportunities.” ————————- www.Arduino.cc



Overview

The OSOYOO Mega2560 Board is fully compatible with Arduino Mega2560 rev.3, it is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button.

It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed

for the Uno and the former boards Duemilanove or Diecimila.

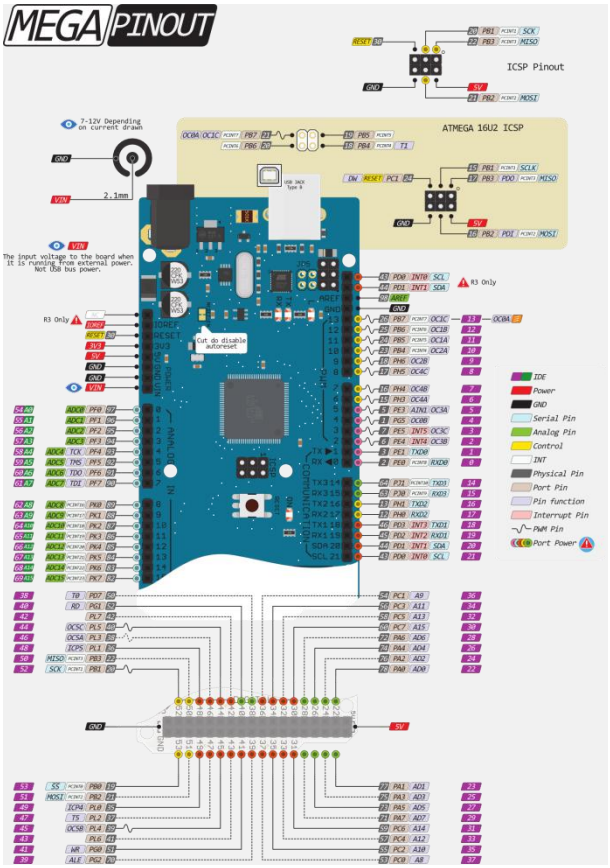
Note:

This is an Arduino Compatible board. It is NOT an original Arduino board, but is similar. None of the Arduino Mega2560 R3 boards sold on the internet at this price are original, they are all copies. This is perfectly legal, seeing that the whole arduino ecosystem is open source! Please note this board is manufactred by OSOYOO! We do have control of the brand and quality of components used! We have also carefully selected suppliers that consistently supply quality products. We strictly control the quality of the products before leaving the factory. The excellent after-sales service and professional technical support will ensure you have a good time with OSOYOO Mega2560 Board

Tech Spec

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Pinout of OSOYOO Mega2560



Documentations

Schematics

Arduino Mega2560 is open-source hardware! You can build your own board using the following files from the Arduino official site:

Arduino_Mega2560_Rev3 Eagle files

Arduino_Mega2560_Rev3 schematic PDF file

Arduino_Mega2560_Rev3 DXF file

Programming

The Mega 2560 board can be programmed with the Arduino Software (IDE). For details, see thereference and tutorials.

The ATmega2560 on the Mega 2560 comes preprogrammed with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available in theArduino repository. The ATmega16U2/8U2 is loaded with a

DFU bootloader, which can be activated by:
On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.

On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Warnings

The Mega 2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Power

The Mega 2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

Vin. The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 – 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current

draw is 50 mA.

GND. Ground pins.

I**OREF.** This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

See the mapping between Arduino pins and Atmega2560 ports:
PIN MAPPING ATmega2560

Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

In addition, some pins have specialized functions:
Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low level, a rising or falling edge, or a change in level. See the attachInterrupt() function for details.

PWM: 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.

SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using theSPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Arduino /Genuino Uno and the old Duemilanove and Diecimila Arduino boards.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is

HIGH value, the LED is on, when the pin is LOW, it's off.

TWI: 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the old Duemilanove or Diecimila Arduino boards.

See also the mapping Arduino Mega 2560 PIN diagram.

The Mega 2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:
AREF. Reference voltage for the analog inputs. Used with analogReference().
Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.
Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically.

The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 / ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Mega 2560's digital pins.

The Mega 2560 also supports TWI and SPI communication. The Arduino Software (IDE) includes a Wire library to simplify use of the TWI bus; see the documentation for details. For SPI communication, use the SPI library.
Physical Characteristics and Shield Compatibility
The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a

surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16″), not an even multiple of the 100 mil spacing of the other pins.

The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Diecimila or Duemilanove Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations.

Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega 2560 and Duemilanove / Diecimila boards. Please note that I2C is not located on the same pins on the Mega 2560 board (20 and 21) as the Duemilanove / Diecimila boards (analog inputs 4 and 5).

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Mega 2560 is designed in a way that allows it to be reset by software running on a connected computer.

One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip.

The Arduino Software (IDE) uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega 2560 board is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the ATMega2560.

While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega 2560 board contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to

re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

Revisions

The Mega 2560 does not use the FTDI USB-to-serial driver chip used in past designs. Instead, it features the ATmega16U2 (ATmega8U2 in the revision 1 and revision 2 Arduino boards) programmed as a USB-to-serial converter. Revision 2 of the Mega 2560 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the Arduino board and the current Genuino Mega 2560 have the following improved features:

- 1.0 pinout: SDA and SCL pins – near to the AREF pin – and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the board that uses ATSAM3X8E, that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

The OSOYOO Mega2560 Board is 100% Software and Hardware compatible with Arduino Mega2560 Board, you can get more info from www.arduino.cc. Thanks for their efforts, it's easier for us to learn Arduino!

1-4 OSOYOO UNO vs OSOYOO Mega 2560

The Following table shows the comparison between the two popular Arduino boards. It is primarily a comparison between OSOYOO Uno vs Mega 2560:

Features	**OSOYOO UNO**	**OSOYOO Mega 2560**
Microcontroller (MCU)	ATmega328P	ATmega2560
Operating Voltage of the Microcontroller	5 V	5 V
Typical Supply Voltage for the board	7 V – 12V	7 V – 12 V
Digital I/O Pins	**14 (includes 6 PWM outputs)**	**54 (includes 15 PWM outputs)**
PWM outputs	**6**	**15**
Analogue Input Pins	**6**	**16**
Max DC Current per I/O Pin	20 mA	20 mA
Max DC Current for 3.3V Pin	50 mA	50 mA
Flash memory of MCU	**32 KB**	**256 KB**
Bootloader footprint	**0.5 KB**	**8 KB**

SRAM of MCU	**2 KB**	**8 KB**
EEPROM of MCU	**1 KB**	**4 KB**
Clock Speed	16 MHz	16 MHz
Programming Interface	USB via ATMega16U2	USB via ATMega16U2
Board Dimensions (Length)	**68.6 mm**	**101.5 mm**
Board Dimensions (Width)	53.4 mm	53.4 mm
Photo		

Microcontrollers:

OSOYOO Mega 2560

OSOYOO Mega 2560 uses an ATmega2560 microcontroller which has:

- 256K Bytes of In-System Self-Programmable Flash.
- 8K Bytes RAM.
- 4K Byte Internal SRAM.

OSOYOO Uno

OSOYOO Uno uses an ATmega328p microcontroller which has:

- Flash Program Memory: 32 kbytes.
- EEPROM Data Memory: 1 kbytes.
- SRAM Data Memory: 2 kbytes.

Boards:

OSOYOO Mega 2560

OSOYOO Mega board has:

- 54 digital I/O pins, of which 14(D0 to D13) can be used as PWM outputs.
- 16 analog input pins, which can also be used as digital I/O pins, adding to the existing 54 digital I/O pins.
- 4 serial communication lines(pins D0, D1, and from D14 to D19).

OSOYOO UNO

The OSOYOO Uno board has:

- 14 digital I/O pins, of which 6 (D3, D5, D6, D9, D10, and D11) can be used as PWM outputs.
- 6 analog inputs, which can also be used as digital I/O pins, adding to the existing 14 digital I/O pins.
- Only one serial communication line (D0, D1).

Final points to remember while selecting one of these:

- An OSOYOO Uno board is best suited for beginners who have just started using microcontrollers, on the other hand, OSOYOO Mega board is for enthusiasts who require a lot of I/O pins for their projects.
- Due to smaller size of the OSOYOO Uno thanOSOYOO Mega, it can be use to make more compact projects, although boards like the Micro or the Nano are recommended for that purpose.

1-5 What is Blynk?

Intro

This guide will help you understand how to get started using Blynk and give a comprehensive overview of all the features.

How Blynk Works

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

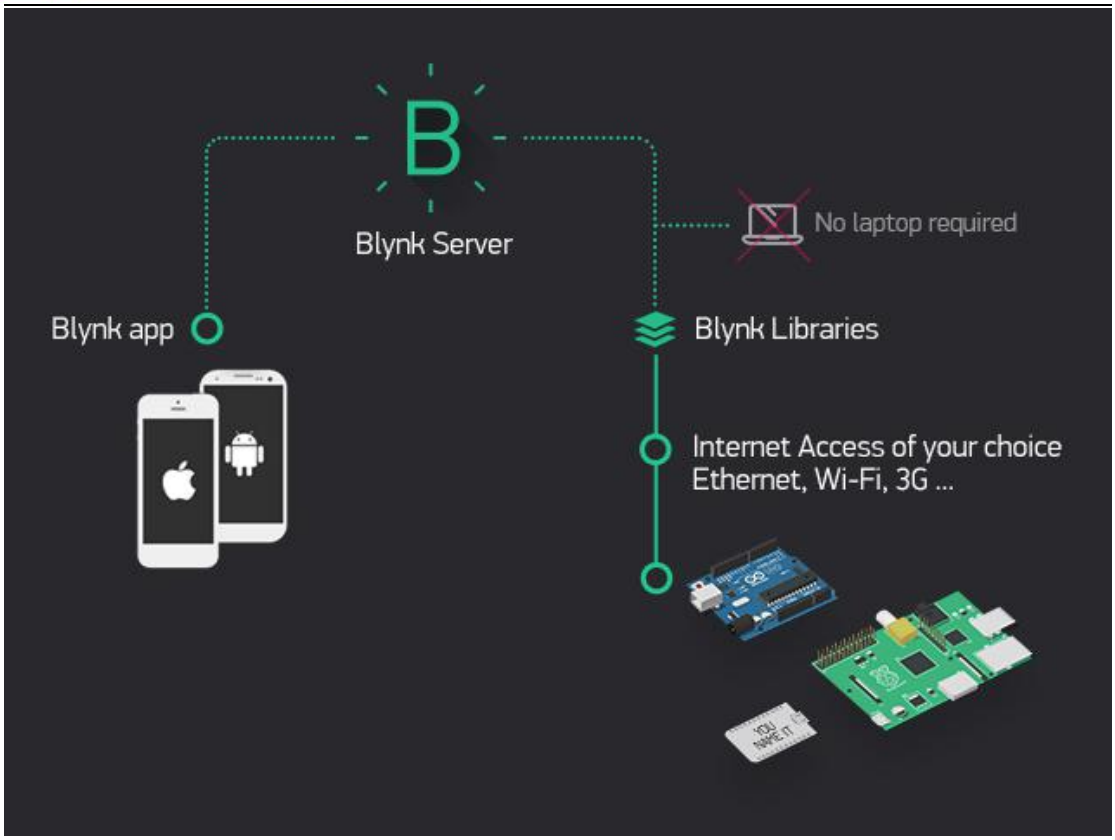
There are three major components in the platform:

- **Blynk App** - allows to you create amazing interfaces for your projects using various widgets we provide.

- **Blynk Server** - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private Blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

- **Blynk Libraries** - for all the popular hardware platforms - enable communication with the server and process all the incoming and outcoming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to space the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.



Features

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using:
  - WiFi
  - Bluetooth and BLE
  - Ethernet
  - USB (Serial)
  - GSM
  - …
- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins
- History data monitoring via SuperChart widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc.
- … new features are constantly added!

You can find example sketches covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

What do I need to Blynk?

At this point you might be thinking: **“Ok, I want it. What do I need to get started?”** – Just a couple of things, really:

1. Hardware.

An Arduino, Raspberry Pi, or a similar development kit.

Blynk works over the Internet. This means that the hardware you choose should be able to connect to the internet. Some of the boards, like Arduino Uno will need an Ethernet or Wi-Fi Shield to communicate, others are already Internet-enabled: like the ESP8266, Raspberri Pi with WiFi dongle, Particle Photon or SparkFun Blynk Board. But even if you don’t have a shield, you can connect it over USB to your laptop or desktop (it’s a bit more complicated for newbies, but we got you covered). What’s cool, is that the list of hardware that works with Blynk is huge and will keep on growing.

2. A Smartphone.

The Blynk App is a well designed interface builder. It works on both iOS and Android, so no holywars here, ok?

Downloads

Blynk Apps for iOS or Android

Download on the App Store GET IT ON Google Play

Blynk Library

DOWNLOAD THE BLYNK LIBRARY

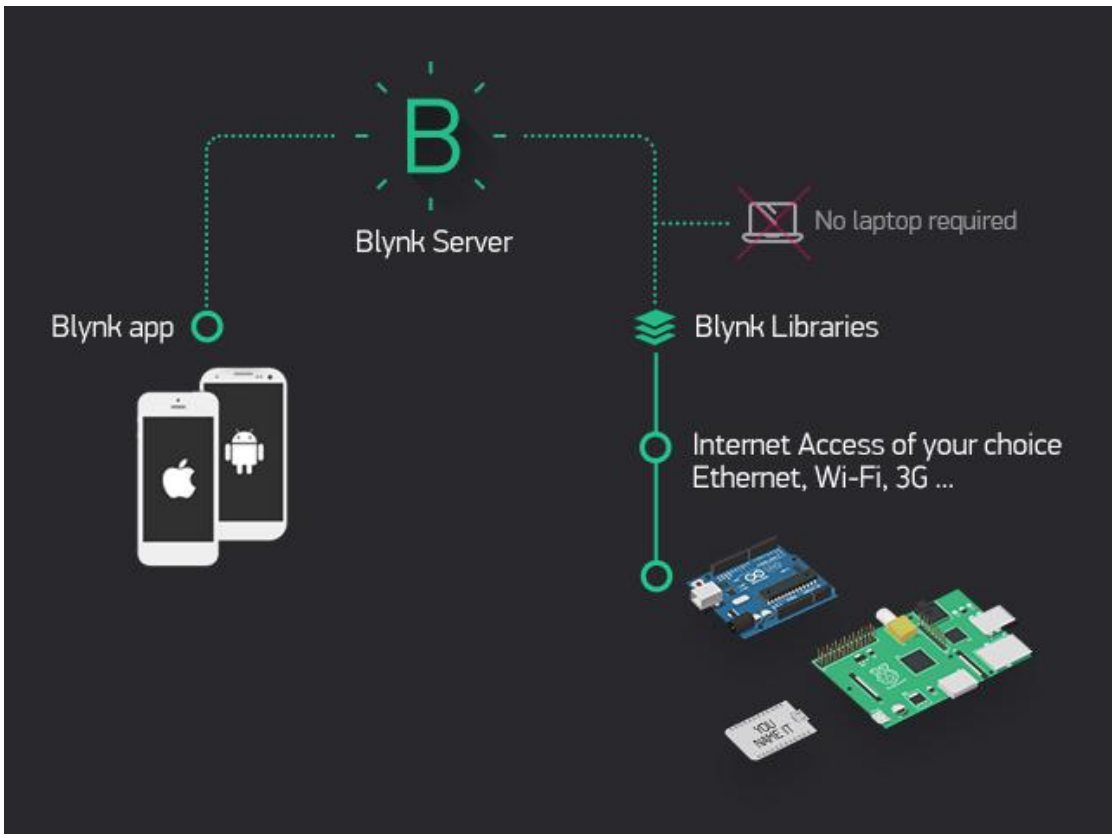
In case you forgot, or don’t know how to install Arduino libraries click here.

1-6 How to install a local Blynk Legacy Server in your PC and get a Local Blynk Token

Introduction

Blynk Legacy is a very powerful Internet of Things(IoT) software solution which allows you to design your own IoT APP without Android and iOS APP programming knowledge.
Following tutorial will guide to install Blynk legacy server in your local Windows, Mac computer or Linux like Raspberry Pi.
Picture 1.1 shows how Blynk systems works:

Picture 1.1



From Above picture, you can see Blynk software consists three parts:
Blynk Legacy APP which is installed in your Cell phone or tablet
Blynk Legacy Server
Blynk Library which is installed in IoT terminal device such as Arduino,

Raspberry Pi etc.

In this Lesson, we will teach students how to install Blynk Legacy local Server and setup APP and Arduino sketch code to make them working in Blynk Local Server.

Install Server on local PC

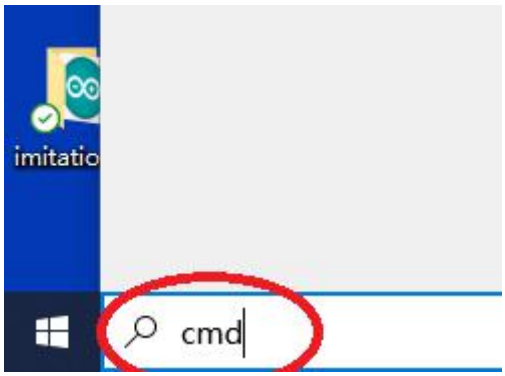
You can install **local Blynk server** on any computer which has Java 11 (or higher version). Java environment is a basic module which comes with those popular OS such as Windows, MacOS and Linux. So if your computer is not too old, you no need install any Java software to run Blynk server.

In this tutorial, we will teach how to install local Blynk server on Windows, MacOS and Raspbian(Raspberry Pi OS). If you use other OS, the installation is similar, just make sure you have Java 11 and JDK 8+ installed in your OS.

INSTALL BLYNK LOCAL SERVER ON WINDOWS

1)Download Blynk local server from
https://osoyoo.com/driver/blynk/blynk-server.jar

2)In Windows search bar , type **cmd** to open command windows



3)After open the command window, go to the fold where blynk-server.jar file is saved. Test your java environment by typing following command in cmd terminal:

```
java -version
```

You should see something as following

```
java version "1.8.0_202"
```

If you can't see java version, it means your haven't installed Java properly,

please read this article to install Java

4)Go to your download folder, replace the bold font path with your folder path

```
cd    C:\Users\Admin\your download folder
```

Then run following command to start the local Blynk server:

```
java -jar ./blynk-server.jar -dataFolder ./Blynk
```

you will see following result:

Now your local Blynk server has been successfully installed and running in your Windows.

INSTALL BLYNK LOCAL SERVER ON APPLE MACBOOK COMPUTER

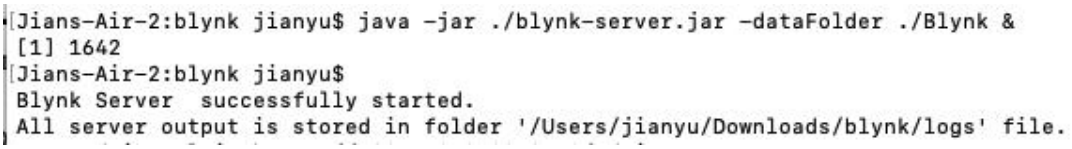
1) Download Blynk local server from
https://osoyoo.com/driver/blynk/blynk-server.jar

2) Open MacBook terminal window,

Go to your download folder, Run following command to start the local Blynk server:

```
java -jar ./blynk-server.jar -dataFolder ./Blynk
```

you will see following result:



INSTALL BLYNK LOCAL SERVER ON RASPBERRY PI OR SIMILAR LINUX COMPUTER

1)Open your Pi shell terminal or use SSH to connect Raspberry Pi remote

terminal

2)Download Blynk local server by following command

```
wget http://osoyoo.com/driver/blynk/blynk-server.jar
```

2)Go to your download folder, Run following command to start the local Blynk server:

```
java -jar ./blynk-server.jar -dataFolder ./Blynk
```

The java server will start in one or two minutes, then you will see following result:

```
Last login: Mon Jan 11 01:25:13 2021
pi@raspberrypi:~ $ java -jar ./blynk-server.jar -dataFolder ./Blynk &
[1] 726
pi@raspberrypi:~ $
Blynk Server  successfully started.
All server output is stored in folder '/home/pi/logs' file.
```

If you want the Raspberry Pi to start Blynk Server automatically when starting your Pi, you can run following command:

```
crontab -e
```

Then you can add following in the bottom crontab file

```
@reboot java -jar /home/pi/blynk-server.jar -dataFolder
/home/pi/Blynk
```

After that, use Ctrl-X to save and exit the file and reboot your computer. Your Blynk server will automatically start every time.
You can visit the admin page of Blynk Server url:

```
https://your_RaspberryPi_ip_address:9443/admin
```

Cell Phone APP download and configure:

Prerequisite:

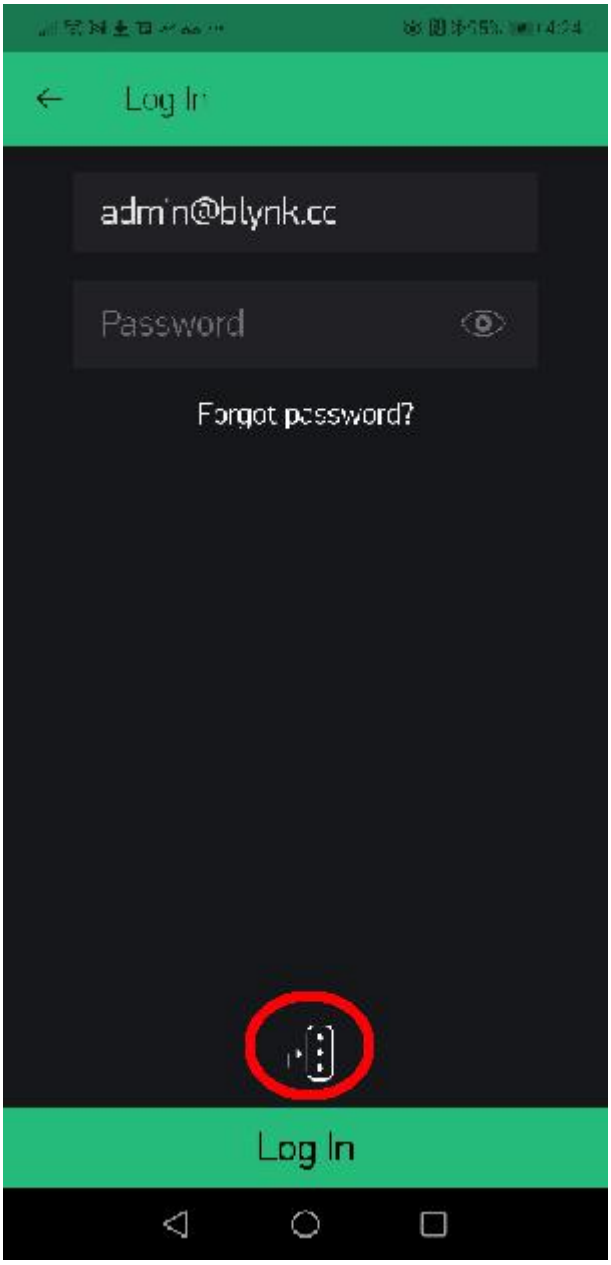
You have successfully installed Blynk server in your local computer. Next step is to install Blynk Legacy APP in your cell phone.
Before install and setup your Blynk APP, you need to know your local server's LAN IP address first, if you don't know how to do it, read following article:

How to know my computer's local IP address

Now let's start to install and setup Blynk Legacy APP

1)In Google Play or iOS APP store, search Blynk Legacy and install the APP.

2) Open Blynk App, click Login , you will see following UI:



3)Click the three dot icon in above UI to set local server IP and port.

Do following settings:

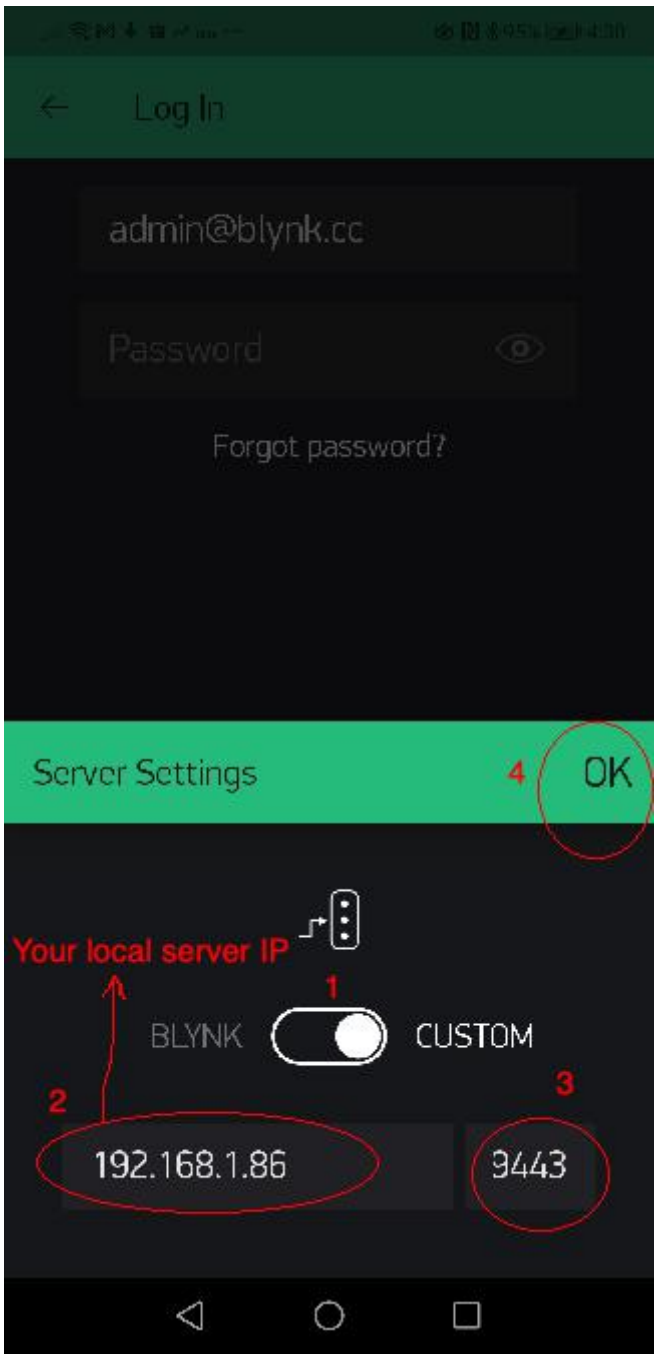
Switch server from BLYNK to CUSTOMS

Set your local blynk server IP

Set port to 9443

click OK

See following picture:

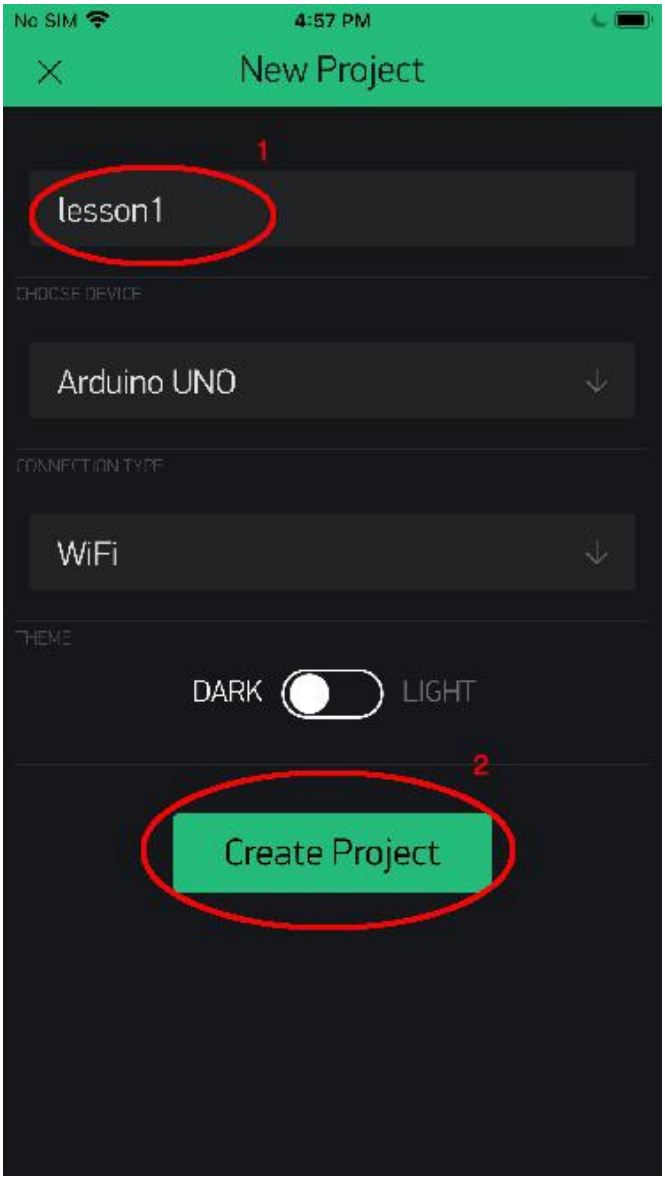


After you have set up the local server IP and port, you can use following steps to connect APP to local Blynk server:

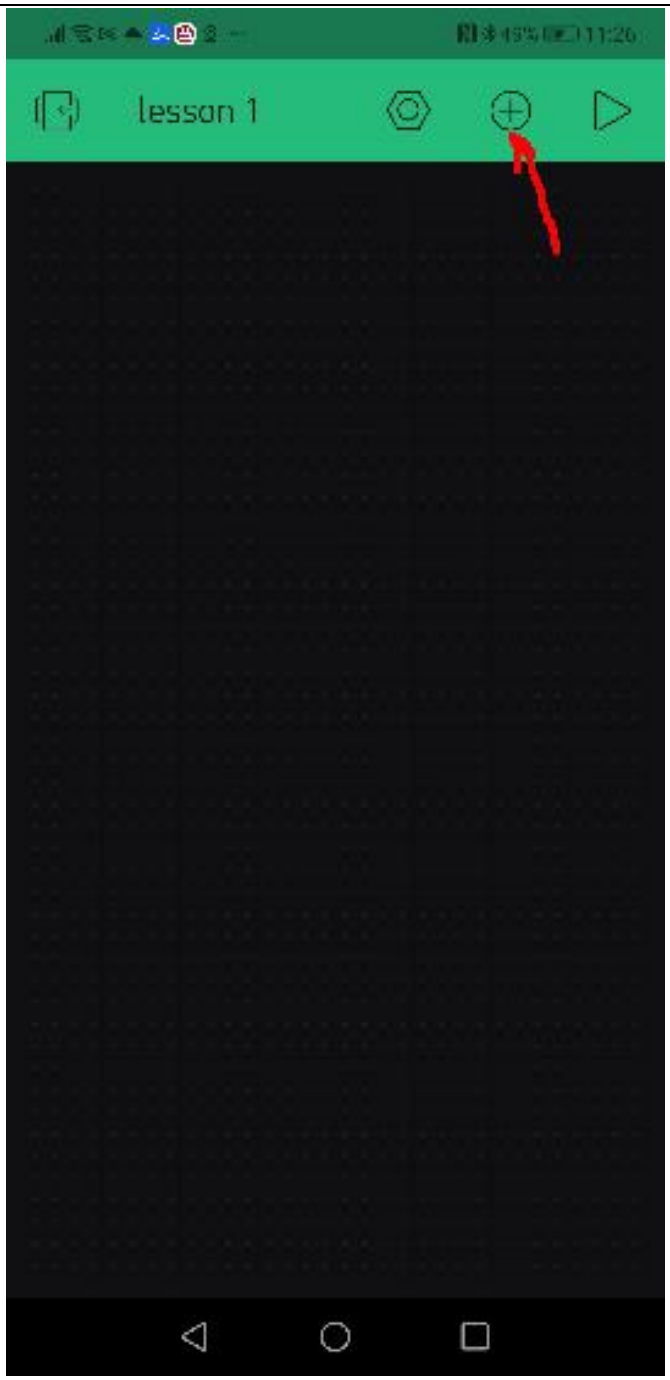
- Set Email field to **admin@blynk.cc**
- Set password field to **admin**
- Click Login

If your Local Blynk is not running properly or IP address setting is wrong, APP will show **Can't connect to Server, Try later…**If server running and IP setting are all ok, you will a New Project UI as following:

Congratulations! You have successfully installed local Blynk Server and connected your APP to the server.
Suppose you are using Arduino as terminal device, you click +New Project to add an Arduino project:
You can name your Project **Lesson1** and select **Arduino UNO** (or **Arduino MEGA** if you are using MEGA2560),



In above UI, Click **Create Project**, you will see a blank project named **lesson1** as following picture, In future, you can click (+) sign to add control Widget to create your APP UI

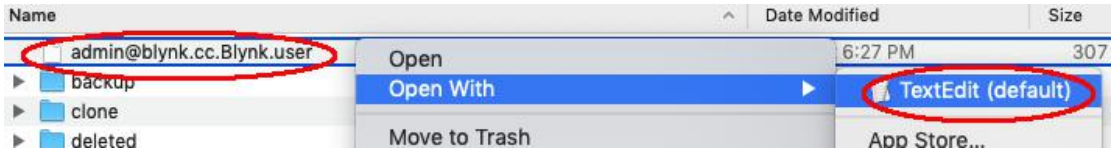


Get Auth Token

Blynk Auth token is the password to let your Arduino board to exchange project data with local Blynk server. You need to set this token in your Arduino code file in future. Token is associated with a project, so you must create a project in APP before getting a Token.

You have two methods to get Blynk Token:

Method 1) In your folder where you download and save blynk-server.jar file, you will see a sub-folder called blynk, enter that folder, you will find a file **admin@blynk.cc.Blynk.user** , please use a text editor to open this file:



After open the the file, you will see following content:

In above picture, you can see that, on the right of the word "**token**", there is a long string

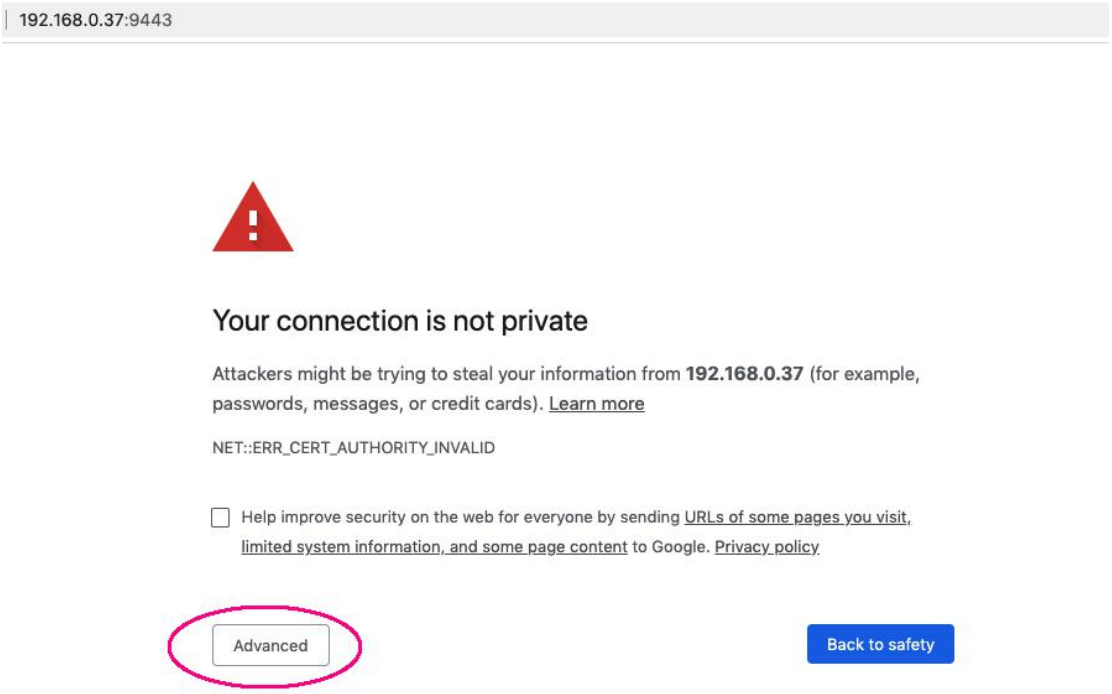
eVEnM2WCQykIlY9w5KPfxrzzWEUfk4N–

This is the token of the **lesson 1** project , copy this token to a file , and you will use it in your Arduino or Raspberry Pi client software code.

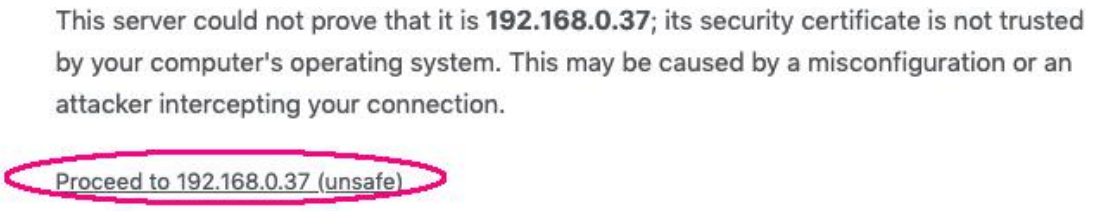
Method 2) Get token from local blynk control panel:

Please open your browser and visit following url:
https://**your_server_ip**:9443/admin
your_server_ip is the IP address of your local Blynk server.

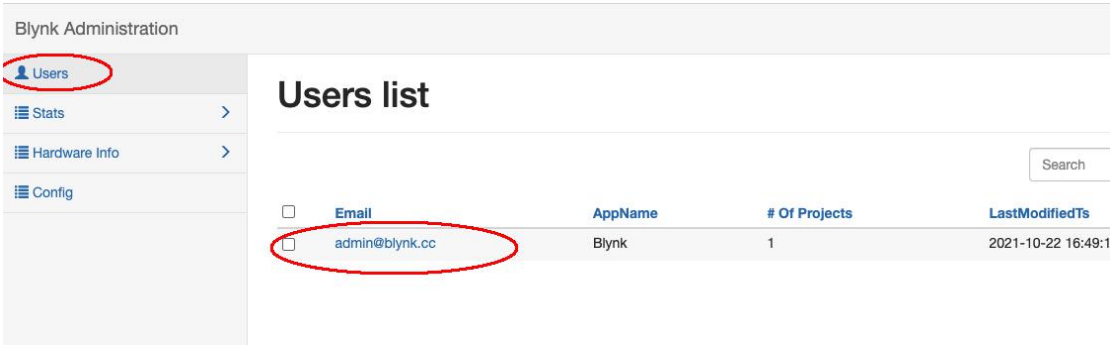
Above https:// url default certificate is self-signed. It will therefore be considered unreliable. You will have to accept the alert message displayed by your internet browser. If you are using Google Chrome browser, You will see following alert:



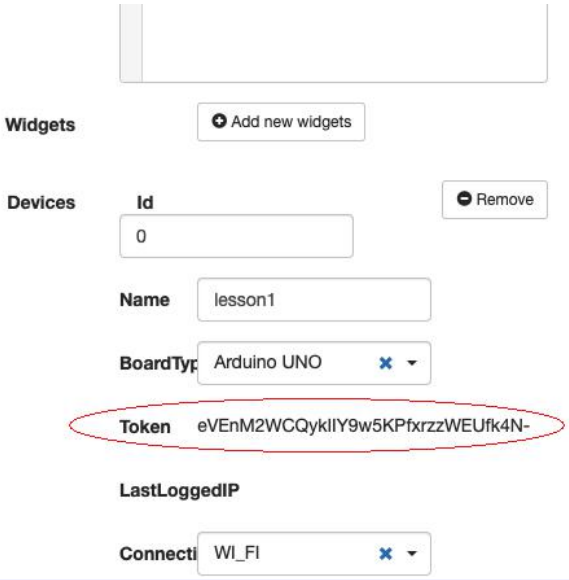
Click Advanced, then select proceed to the unsafe IP as following:



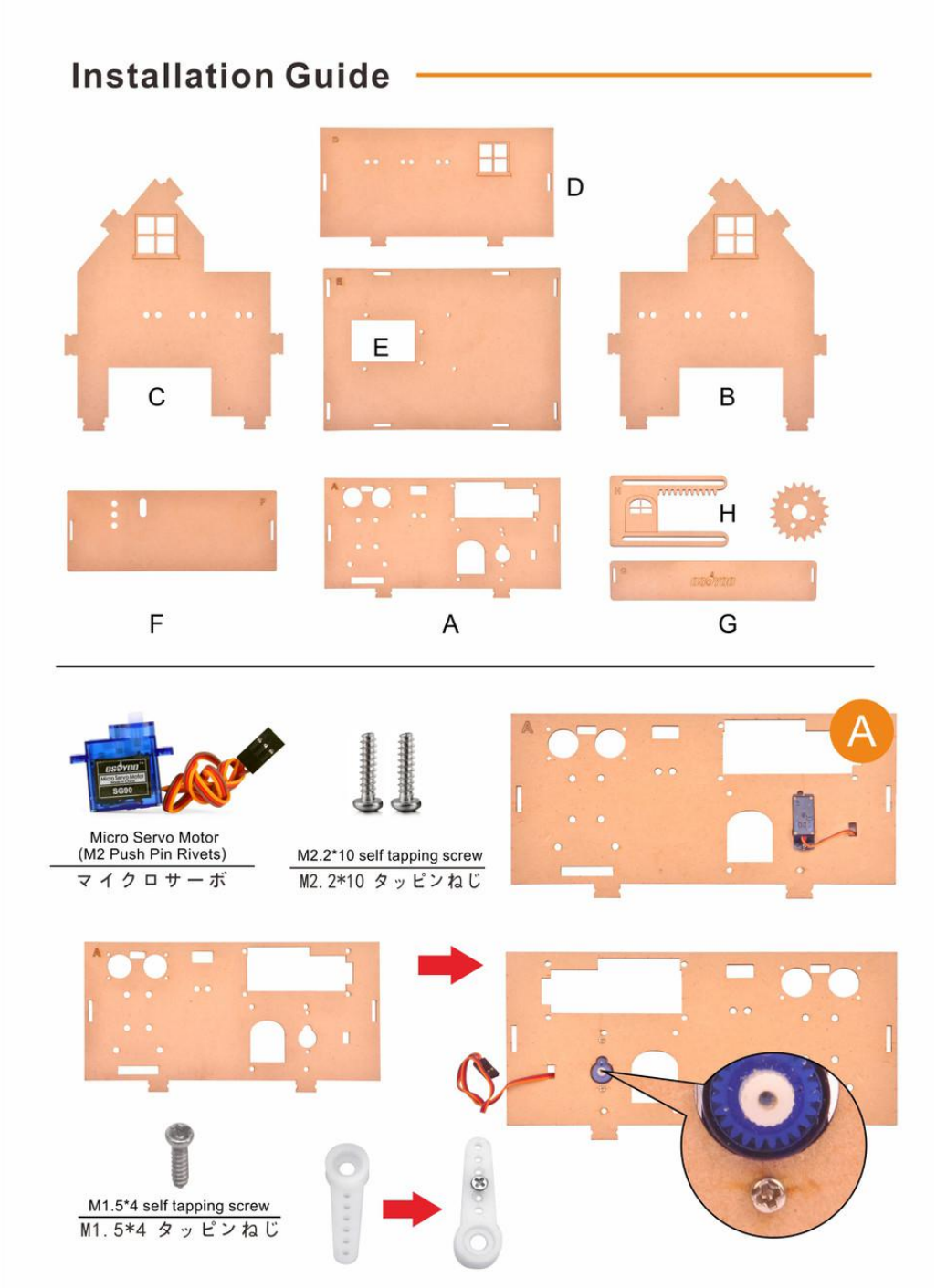
You will go to Server login page, please use user name **admin@blynk.cc** and password **admin** to login to control panel.
then please click Users->admin@blynk.cc:

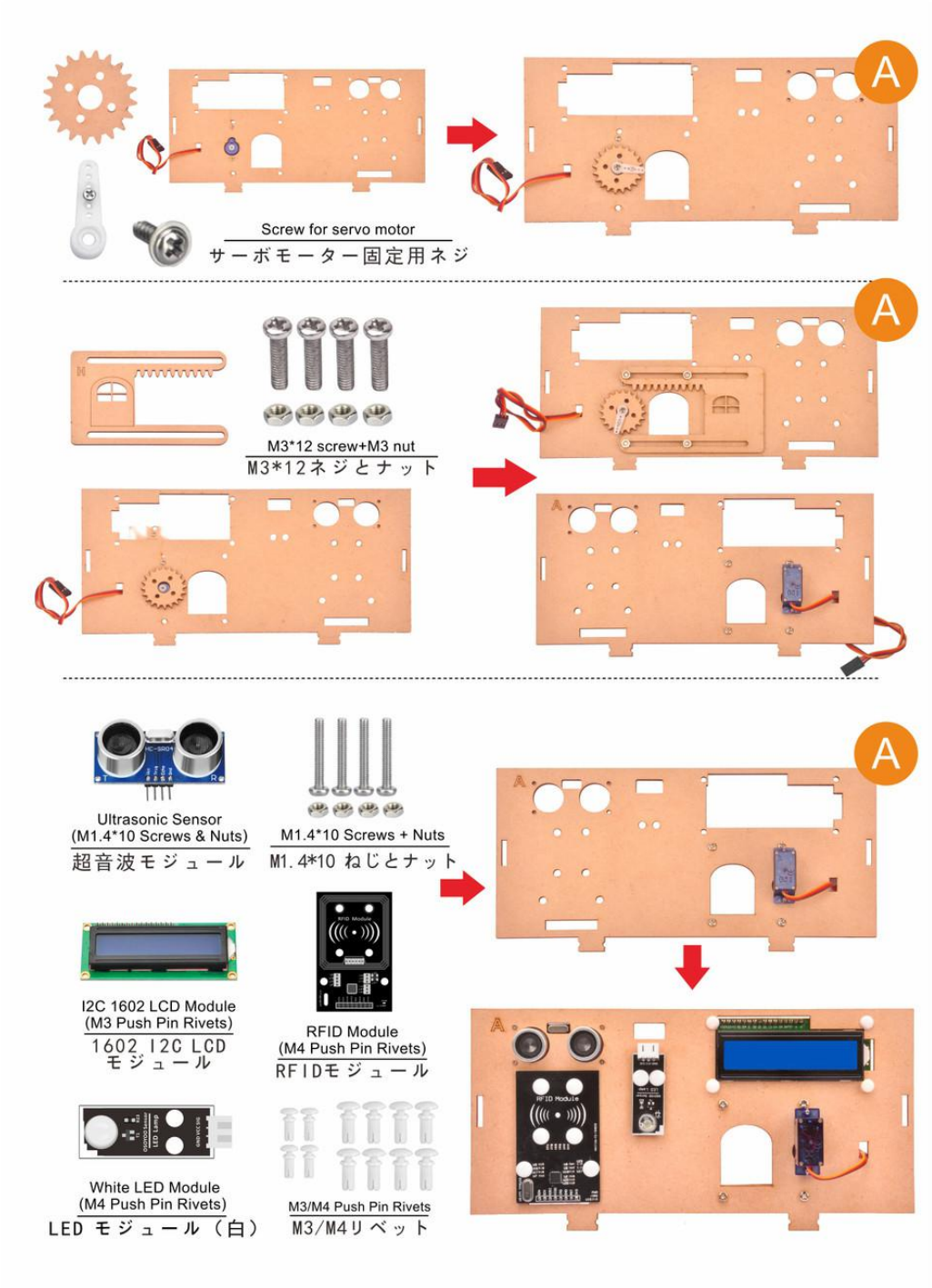


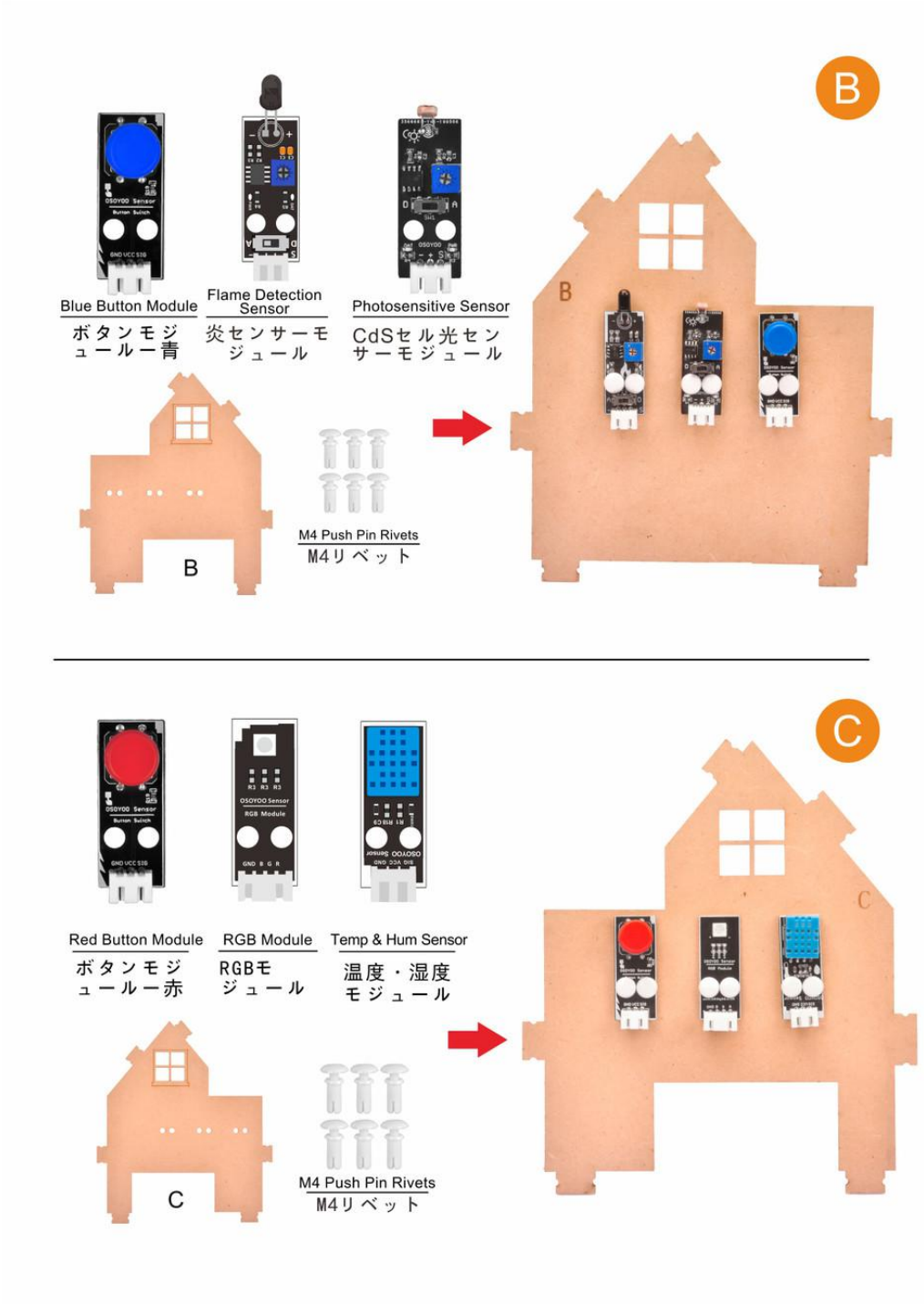
Click admin@blynk.cc, then you will find the Token which was just assigned for your project:

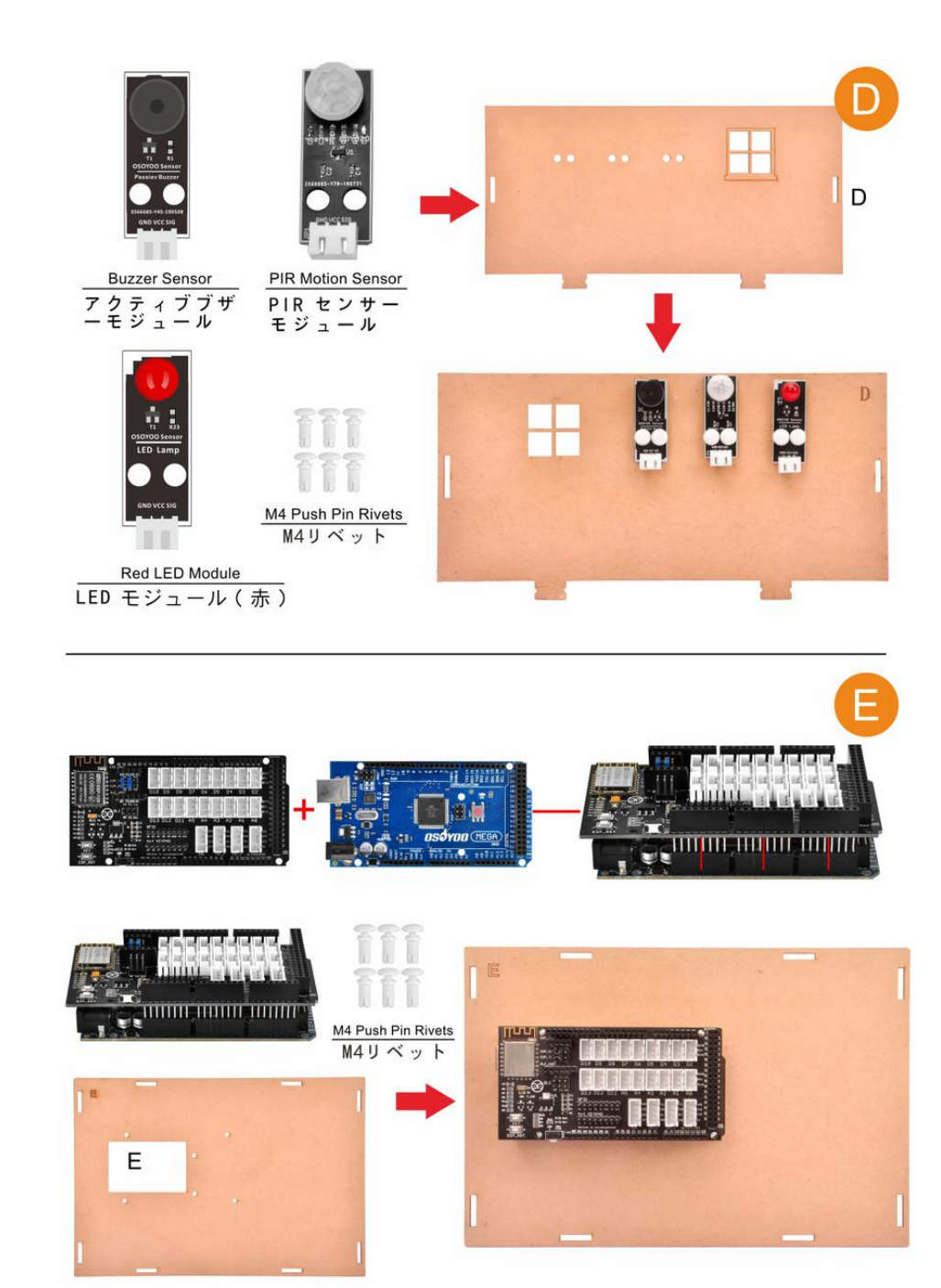


1-7 Model Installation

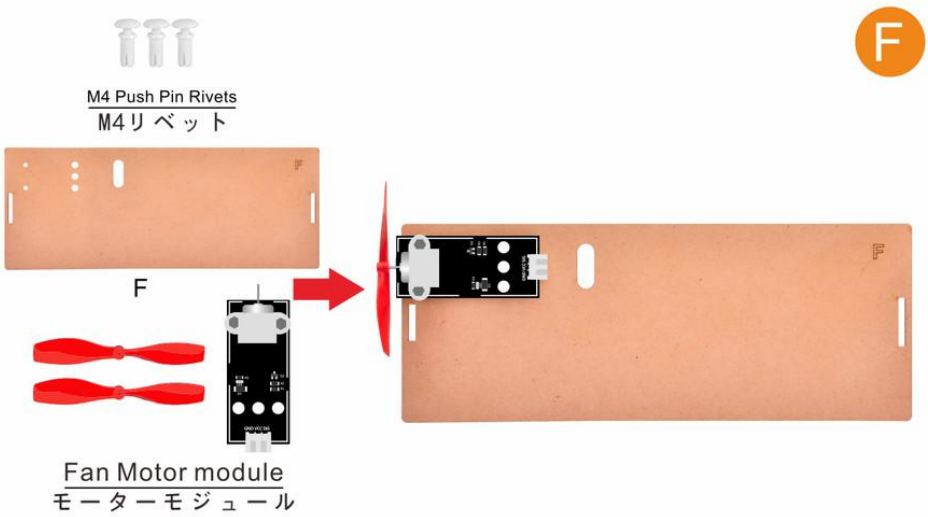



A
Screw for servo motor
サーボモーター固定用ネジ
A
M3*12 screw+M3 nut
M3*12ネジとナット
A
Ultrasonic Sensor
(M1.4*10 Screws & Nuts)
超音波モジュール
M1.4*10 Screws + Nuts
M1.4*10 ねじとナット
A
I2C 1602 LCD Module
(M3 Push Pin Rivets)
1602 I2C LCD
モジュール
RFID Module
(M4 Push Pin Rivets)
RFIDモジュール
White LED Module
(M4 Push Pin Rivets)
LED モジュール（白）
M3/M4 Push Pin Rivets
M3/M4リベット
A


B
Blue Button Module
ボタンモジュール青
Flame Detection Sensor
炎センサーモジュール
Photosensitive Sensor
CdSセル光センサーモジュール
B
M4 Push Pin Rivets
M4リベット
B
C
Red Button Module
ボタンモジュール赤
RGB Module
RGBモジュール
Temp & Hum Sensor
温度・湿度モジュール
C
M4 Push Pin Rivets
M4リベット
C


Buzzer Sensor
アクティブブザーモジュール
PIR Motion Sensor
PIR センサーモジュール
D
D
Red LED Module
LED モジュール（赤）
M4 Push Pin Rivets
M4リベット
E
M4 Push Pin Rivets
M4リベット
E


E
front
brack


F
M4 Push Pin Rivets
M4リベット
F
Fan Motor module
モーターモジュール

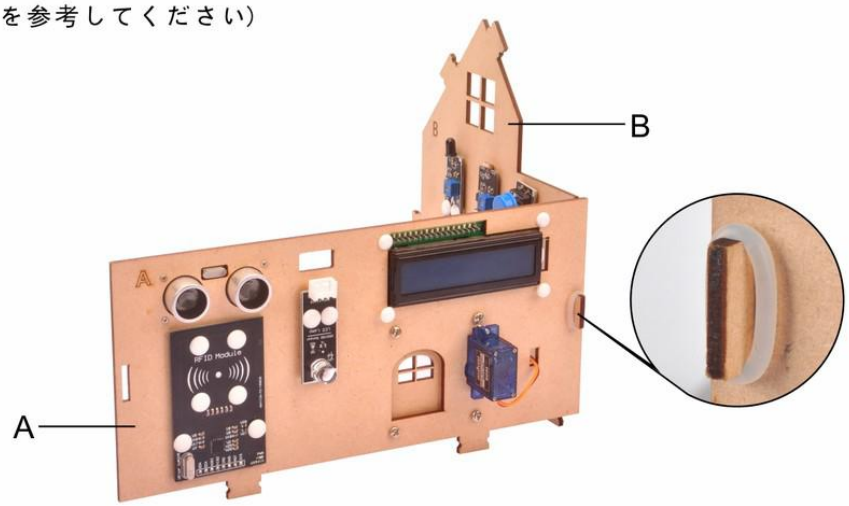
Installation Procedure

Step 1

Insert B board into A board and fasten it with rubber rings.

B板をA板に挿入してから、ゴム輪で固定してください。

（図を参考してください）

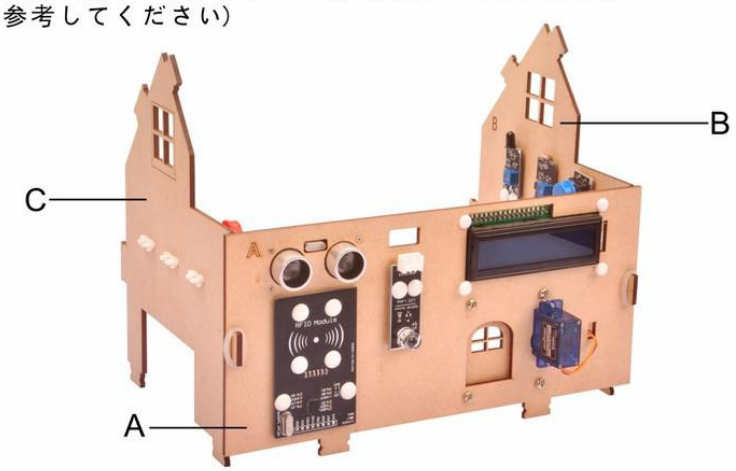


Step 2

Insert C board into A board and fasten it with rubber rings.

C板をA板に挿入してから、ゴム輪で固定してください。

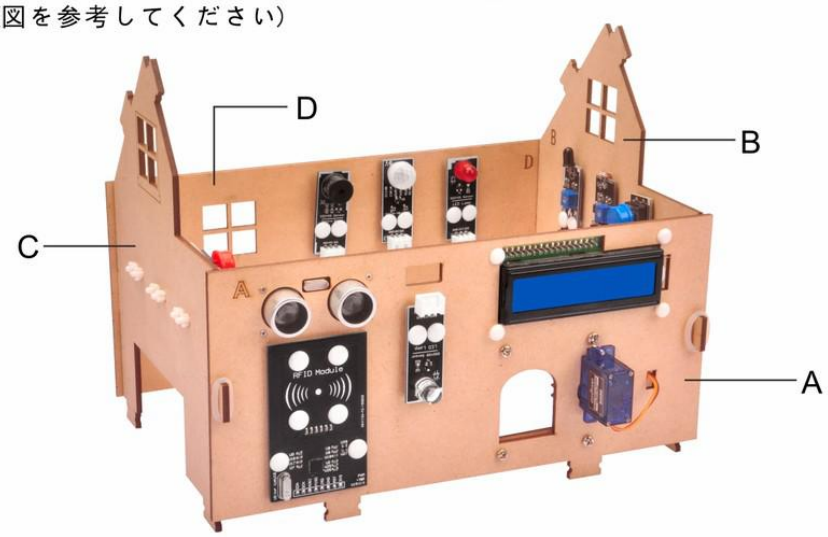
（図を参考してください）



Step 3

Insert D board into B & C board and fasten it with rubber rings.

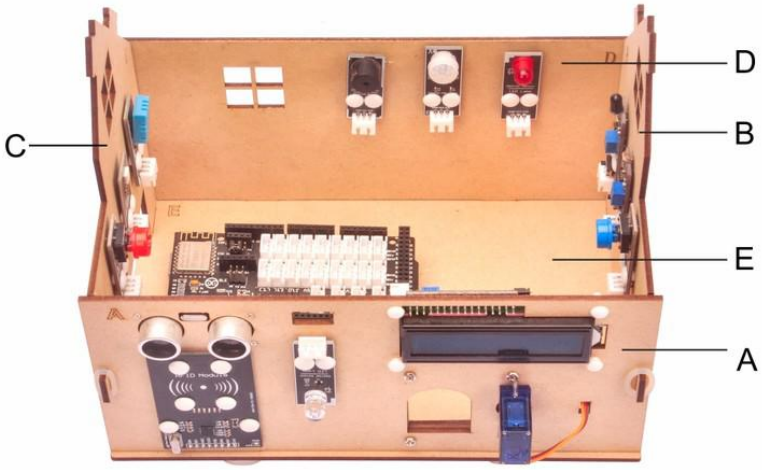
D板をBとC板に挿入してから、ゴム輪で固定してください。
（図を参考してください）



Step 4

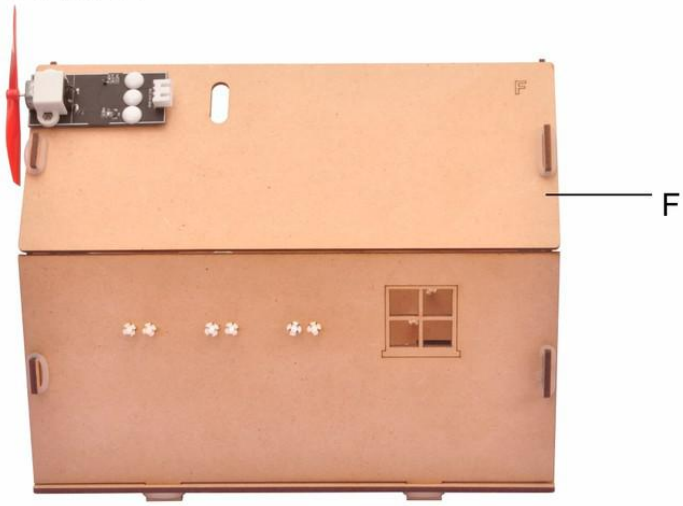
Insert E board into A B C D board and fasten it with rubber rings.

E板をA、B、C、D板に挿入してから、ゴム輪で固定してください。
（図を参考してください）



Step 5

Insert F board into B & C board and fasten it with rubber rings.
F板をBとC板に挿入してから、ゴム輪で固定してください。
(図を参考してください)



Step 6

Insert G board into B & C board and fasten it with rubber rings.
G板をBとC板に挿入してから、ゴム輪で固定してください。
(図を参考してください)

2-1 Wireless Lighting Control

OBJECTIVE

Here, we will show how to use OSOYOO Smart Home IoT Kit with Blynk APP to control some electronic devices, such as leds, active buzzers, fan modules,relay modules, etc.

In this lesson, we will learn how to use Blynk to remotely turn on or turn off the LED and control the brightness of the LED. . . Using the same principle, you can also control other electronic devices

PARTS & DEVICES

HARDWARE

OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
LED Module x 1
USB Cable x 1
3-pin PnP Cable x 1

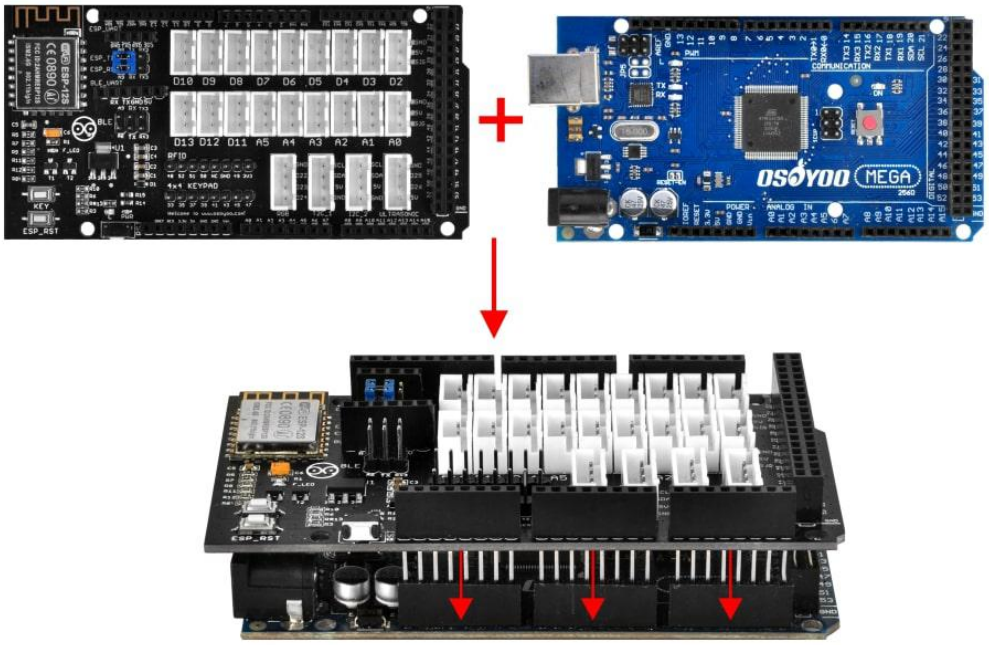
SOFTWARE

Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

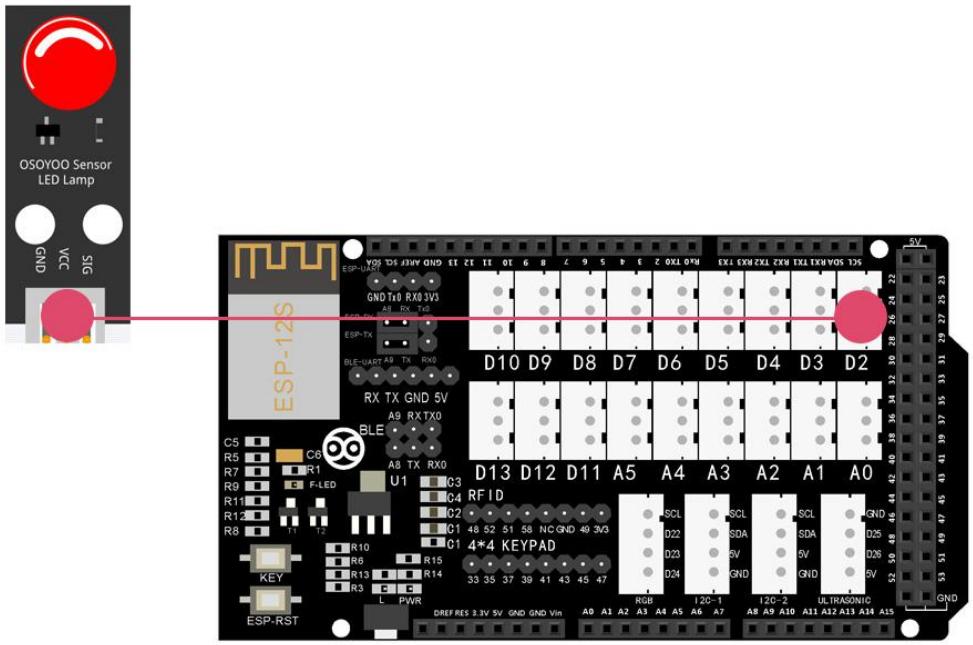
HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:

Make sure that jumper caps in following red circle are installed. These two jumper caps connect A9 to ESP_TX and A8 to ESP_RX.



Connect the LED Module to the D2 port as below:



HOW TO CODE

Step 1 Prerequisite

You must have installed Blynk Legacy APP in your cell phone and local Blynk Server in one of your local network computer. You also need to get your Blynk Server local IP address, create a new project and get Blynk Token.

If you haven't done above tasks yet, please read following tutorial to complete the tasks:
https://osoyoo.com/2021/01/15/how-to-install-a-local-blynk-server-in-your-pc-and-get-a-local-blynk-token/

Step 2 Install latest Arduino IDE (If you have Arduino IDE version after 1.1.16, please skip this step)

Download Arduino IDE from the following link, then install the software.
https://www.arduino.cc/en/software

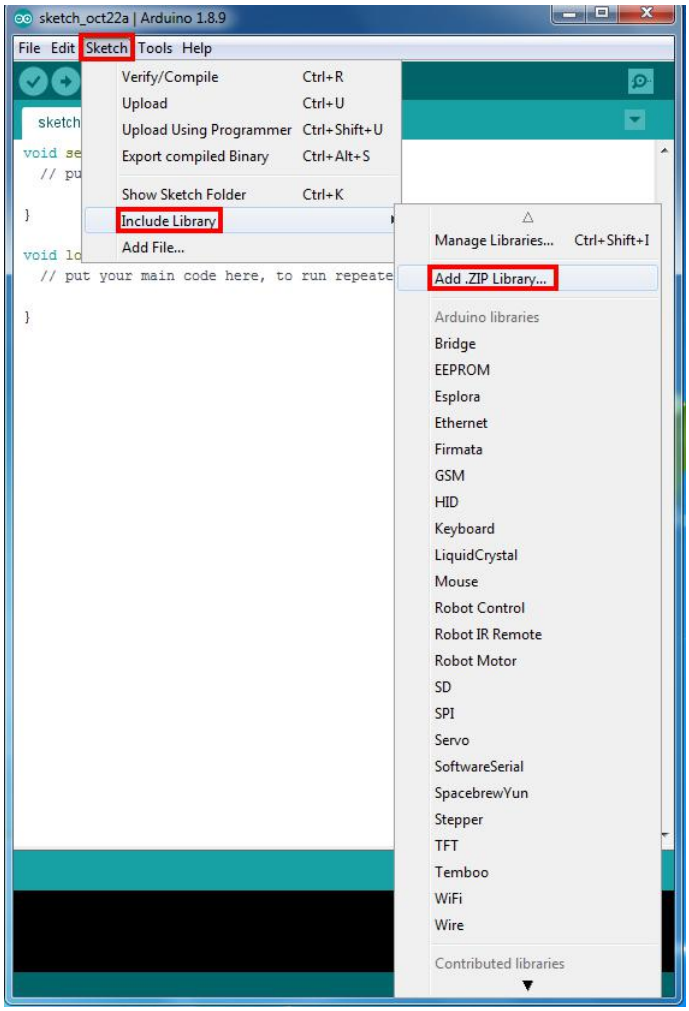
Step 3 Library Installation

Download Blynk library from:
https://osoyoo.com/driver/blynk/Blynk.zip

Download BlynkESP8266_Lib library from:
https://osoyoo.com/driver/blynk/BlynkESP8266_Lib.zip

You need go to Arduino IDE ->Sketch ->Include Library ->Add ,Zip library to

load above zip files into Arduino IDE.

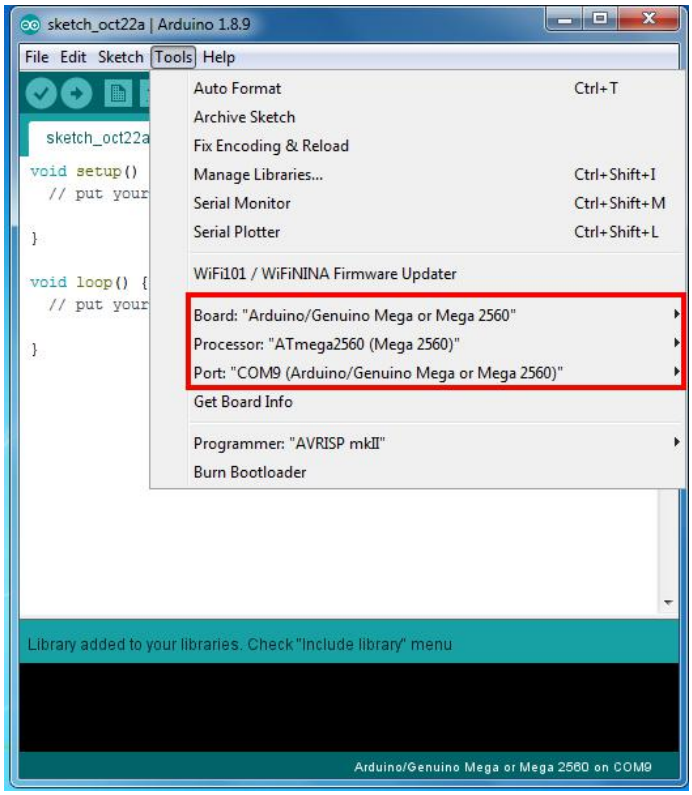


Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-1.zip

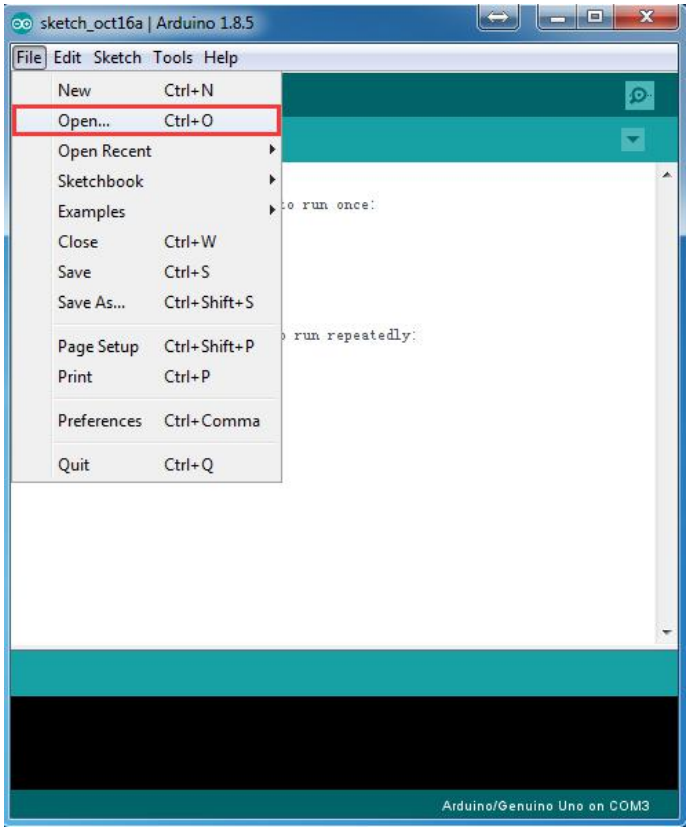
Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino IDE.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click **file** – **Open**, then choose code in the folder, load up the sketch onto your Arduino.



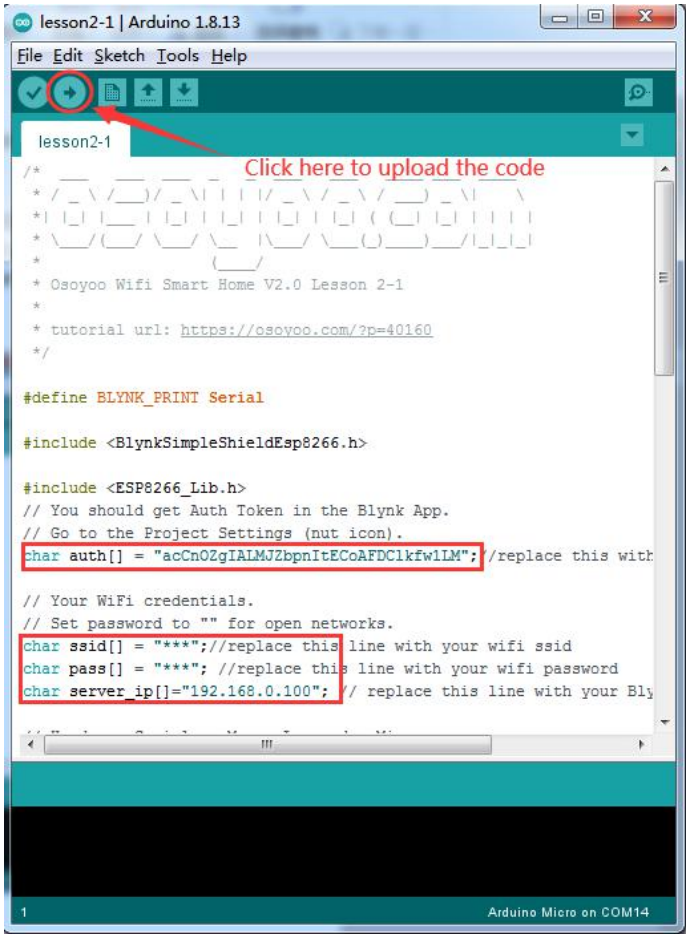
Note: In the sketch, find this line as following:
char auth[] =”sM1i_rSJjDlQxesfuhOzrL0h9NiivJkn”;

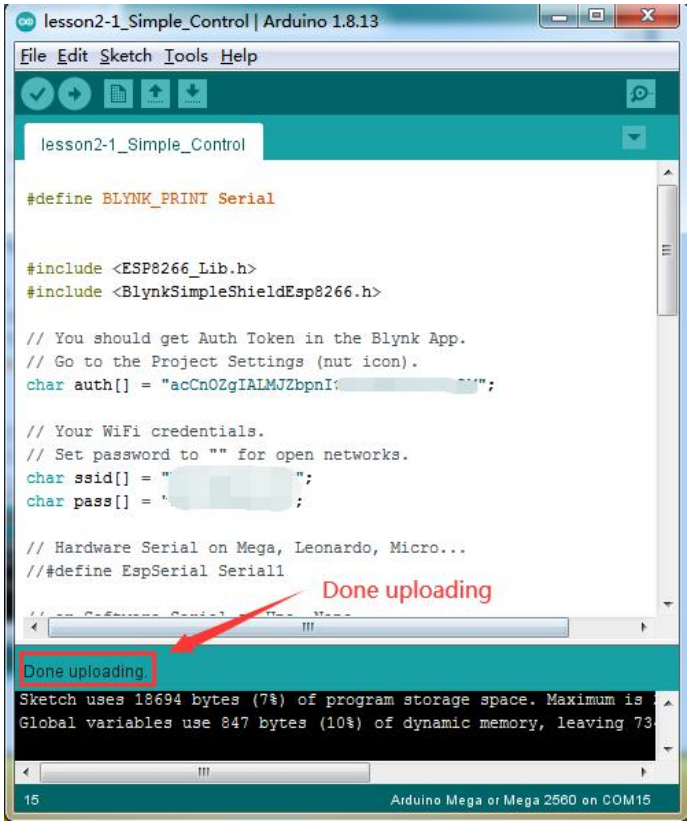
Replace **sM1i_rSJjDlQxesfuhOzrL0h9NiivJkn** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

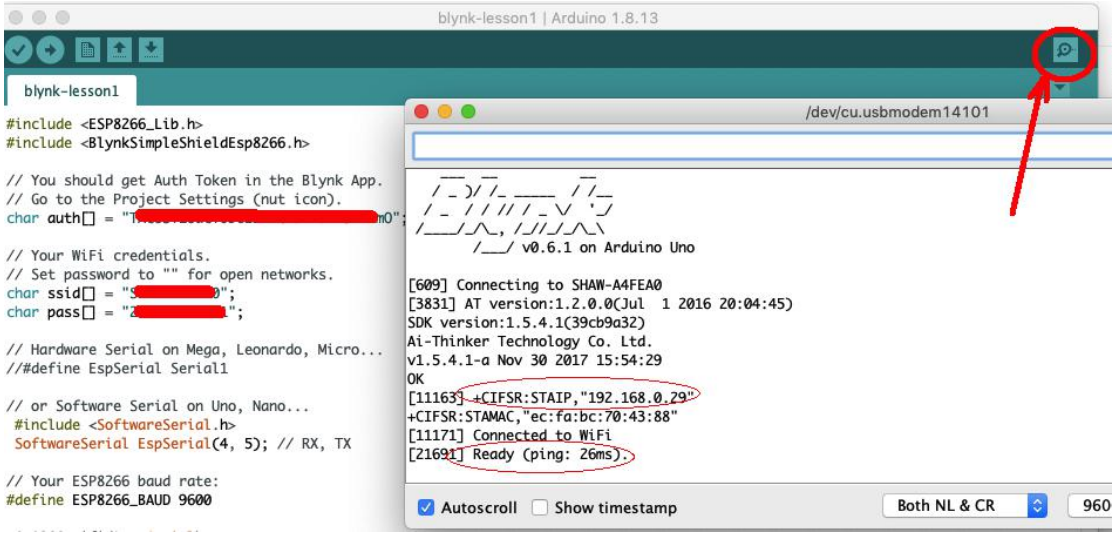
please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14


lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

Now open Serial monitor in the upper right corner, you should see your Arduino IP address and Ready message as following:



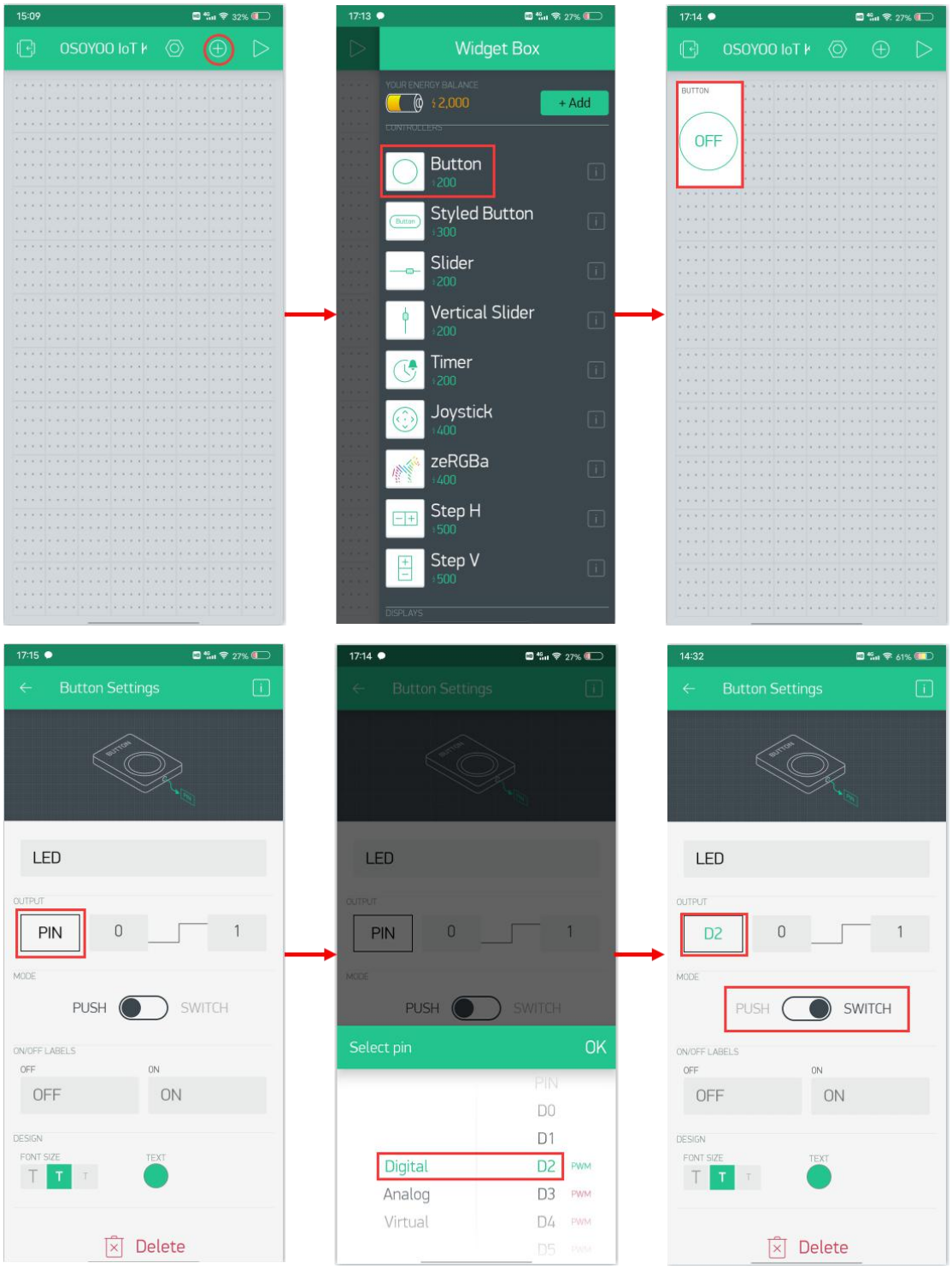
Congratulations! You are all ready! Your hardware is now connected to the Blynk cloud!

If you can't see your IP address, please check your Wifi SSID/Password setting in your code, If you can see IP address but no Ready message, please check your Blynk Token and Local Server IP address are correctly updated in your code file.

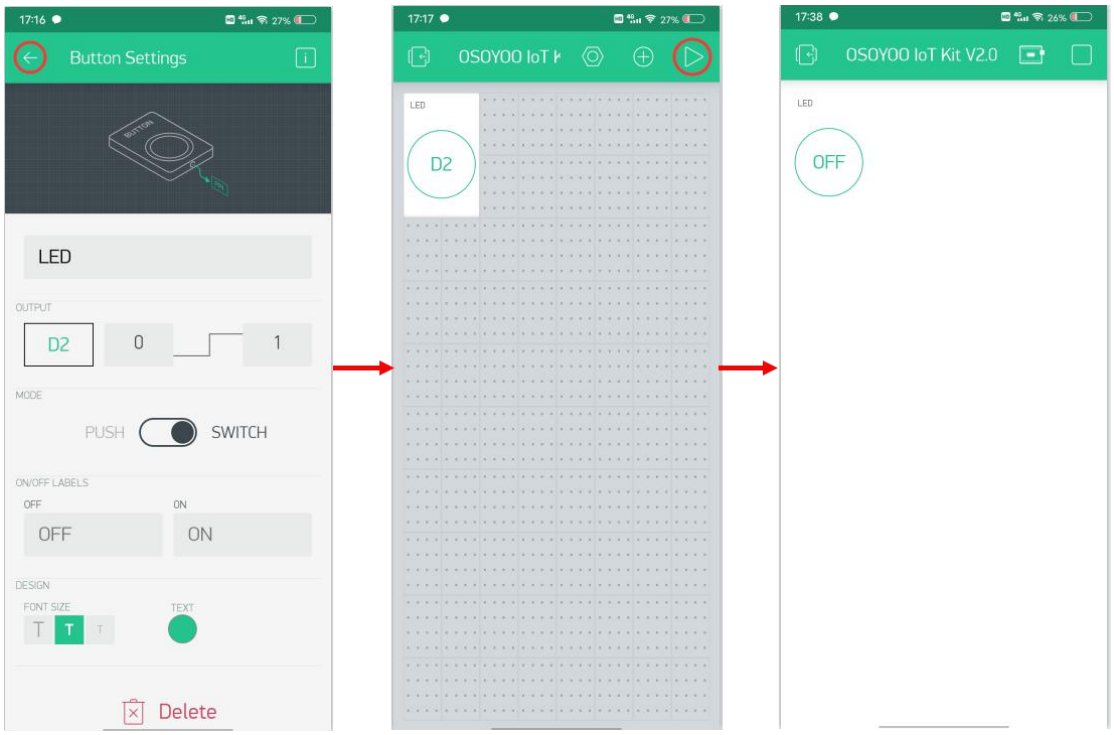
Blynk(legacy) APP PROJECT SETUP

Open your project page and press the "+" button to add the "Button" Widget.

Button Widget settings:
Name Column: You can name your button widget, whatever you like.
OUTPUT: D2 | 0-1
MODE: We choose the Switch mode here.
You can modify other options according to your own habits or keep them as default.



Then back to the project page. and press the "▷ " button to start your project.



HOW TO PLAY

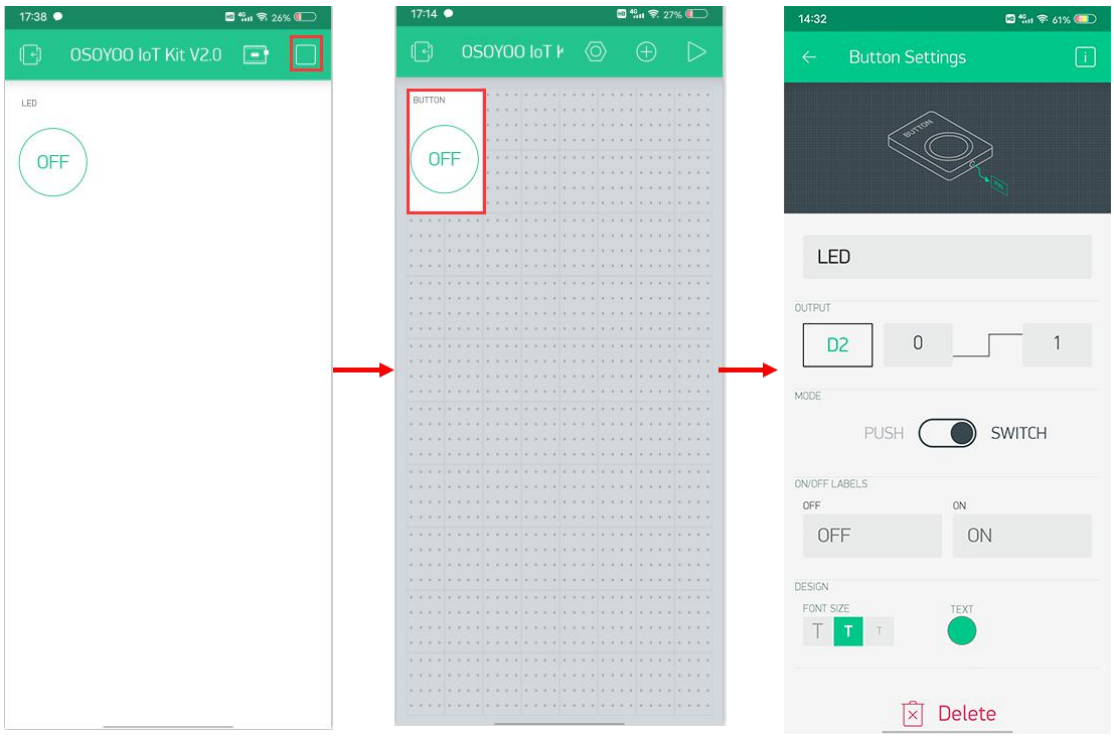
Press the Button on the project page and you will find that the LED connected to D2 has been lit.

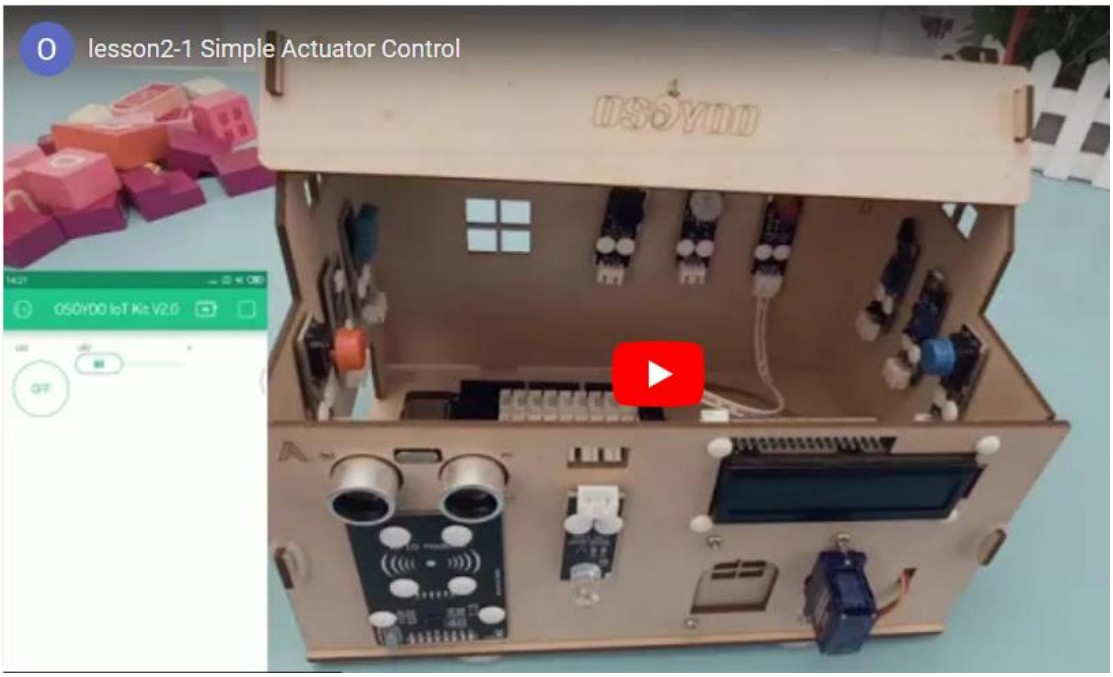
Press the Button again, the LED will be extinguished.

While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

If you find the device is offline, please check your code, wiring and app settings, and try again.

http://OSOYOO.com/?p=40440





https://youtu.be/XHJ3hvjUoqw

Remotely control the brightness of the LED

In this section we will introduce how to control the brightness of the light through this kit and Blynk APP.

The wiring and code of this experiment are the same as the previous part, we only need to add a new widget.

APP PROJECT SETUP

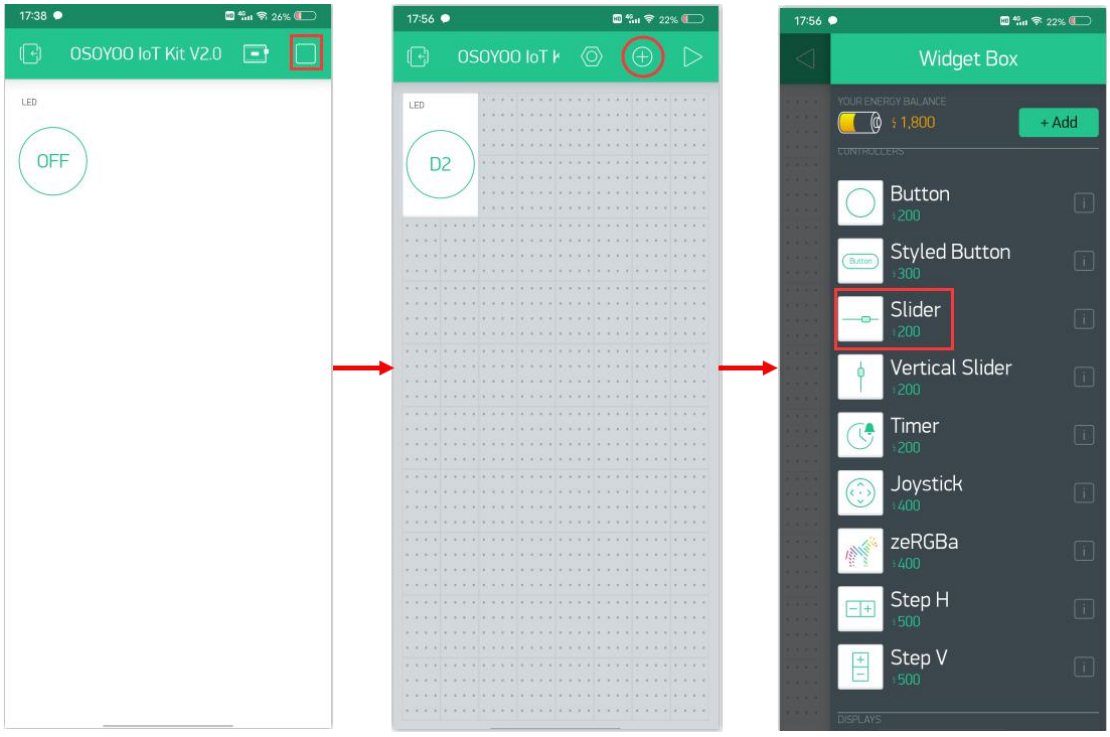
Open your project page and press the "+" button to add the "Slider" Widget.

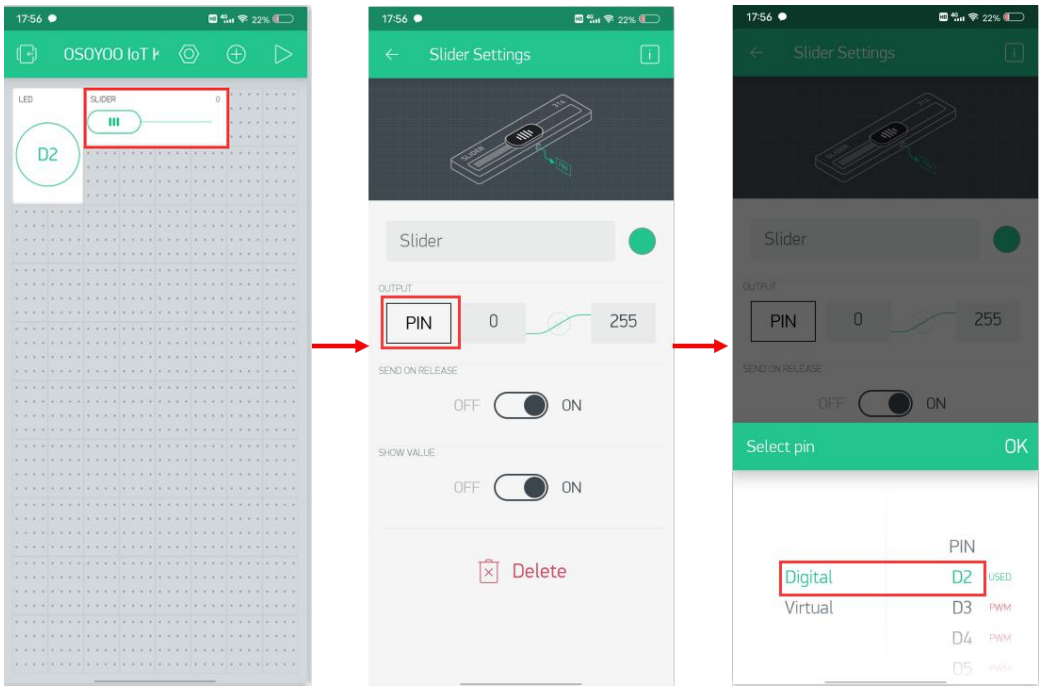
Slider Widget settings:

Name Column: You can name your button widget, whatever you like.

OUTPUT: D2 | 0-255

You can modify other options according to your own habits or keep them as default.

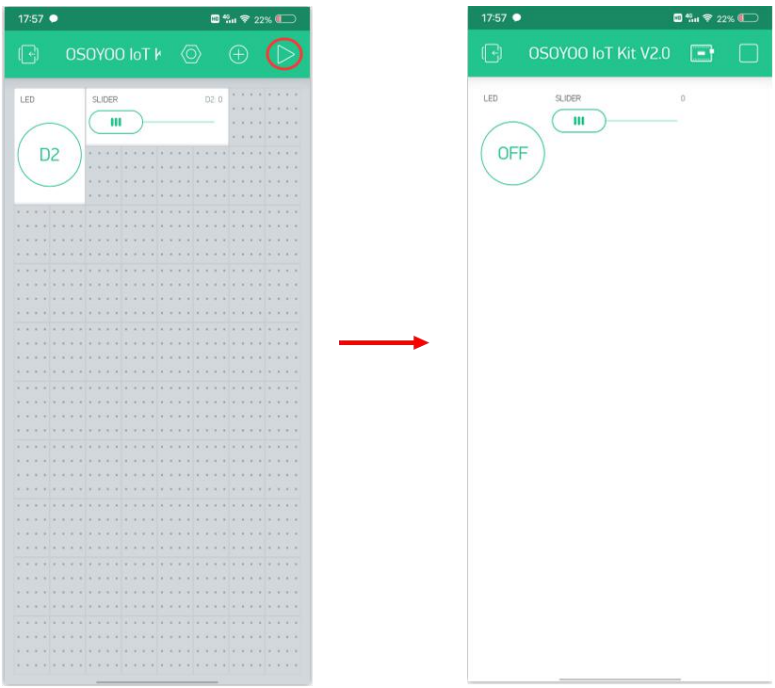




RUNNING RESULT

After you finished all above operations,open the Serial Monitor and you will see the connection situation,then open the Blynk APP, press the PLAY button "▷ ". This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

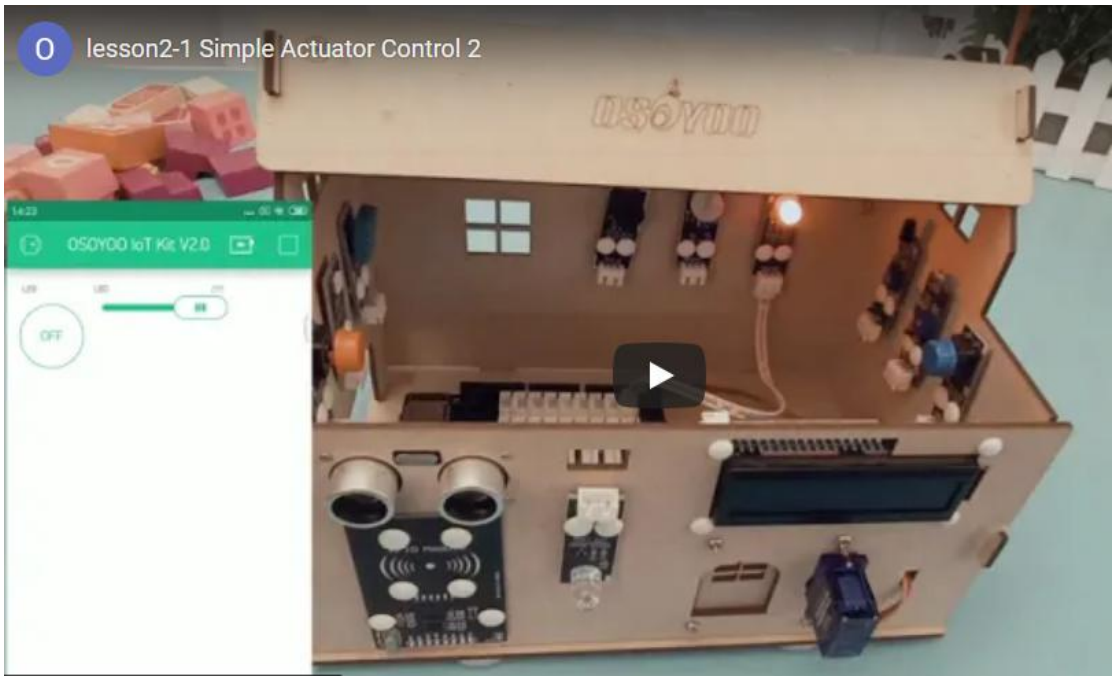
While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.



Slide the slider to change the brightness of the LED module connected to D2.

When the slider is on the far left, the LED is off, and when the slider is on the far right, the LED is brightest.

If you find the device is offline, please check your code, wiring and app settings, and try again.



https://youtu.be/Dx2ae--M98Y

2-2 Collect data remotely

OBJECTIVE

Overhere, we will demonstrate how to use this kit to remotely collect sensor data through the Blynk APP, In this way, we can use the mobile phone to check the environmental data information at home.

In this project, we will view the Arduino button status remotely through the Blynk APP. If the button is pressed, the LED widget on the APP will be lit, otherwise it will be off. Through the same principle, we can remotely measure the state of other sensors in digital mode, such as photosensitive sensor module, flame sensor module, human infrared sensor module, etc.

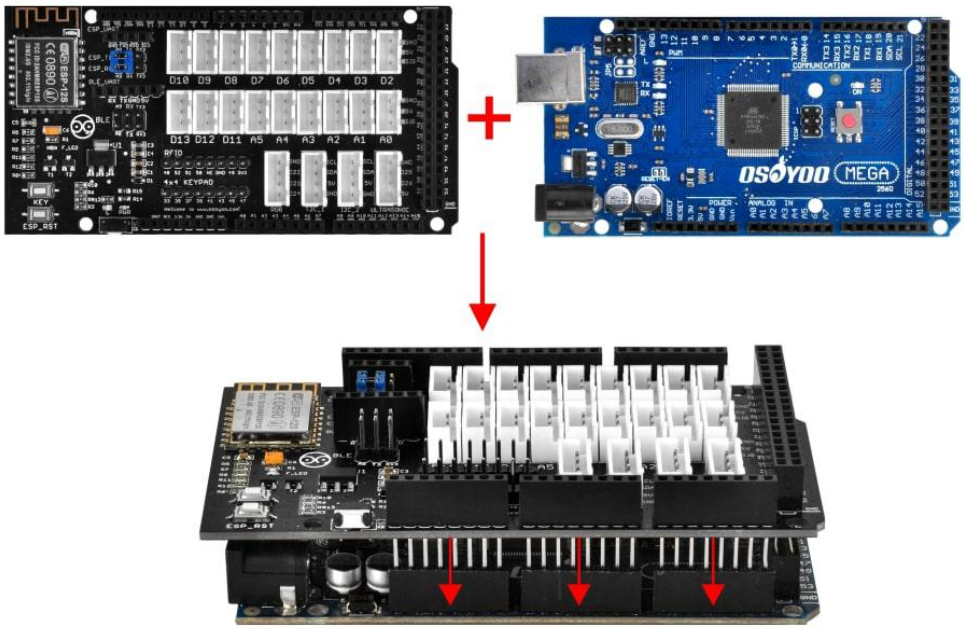
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 (Fully compatible with Arduino MEGA2560) x 1
OSOYOO IoT Wifi Shield x 1
Button Module x 1
LED Module x1
USB Cable x 1
3-pin PnP Cable x 2

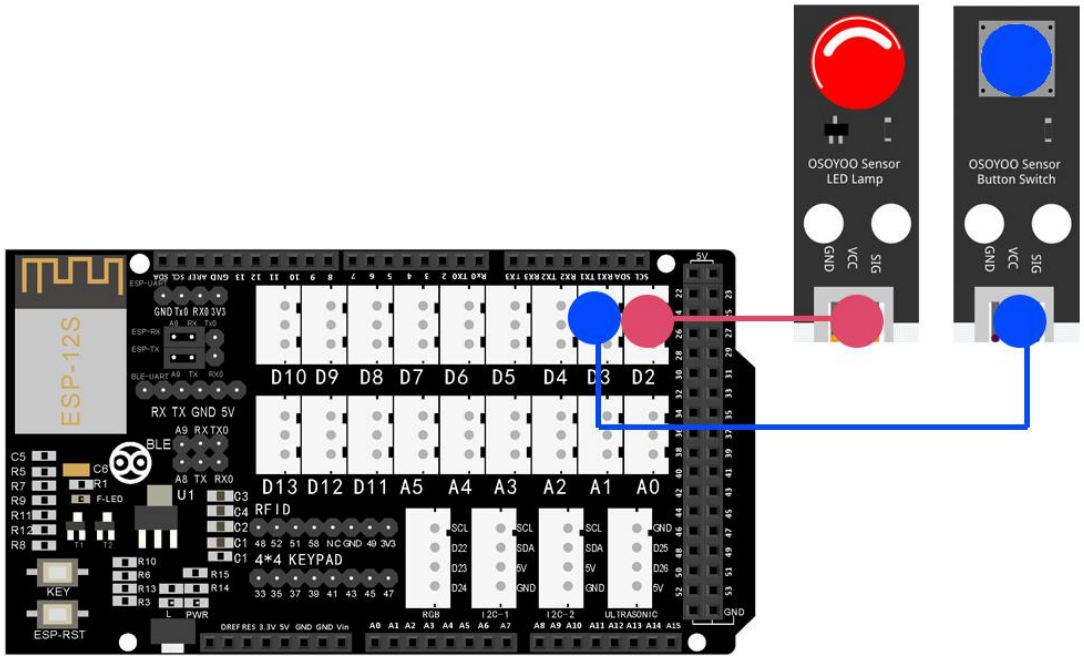
SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:
LED Module – D2
Button Module – D3



HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

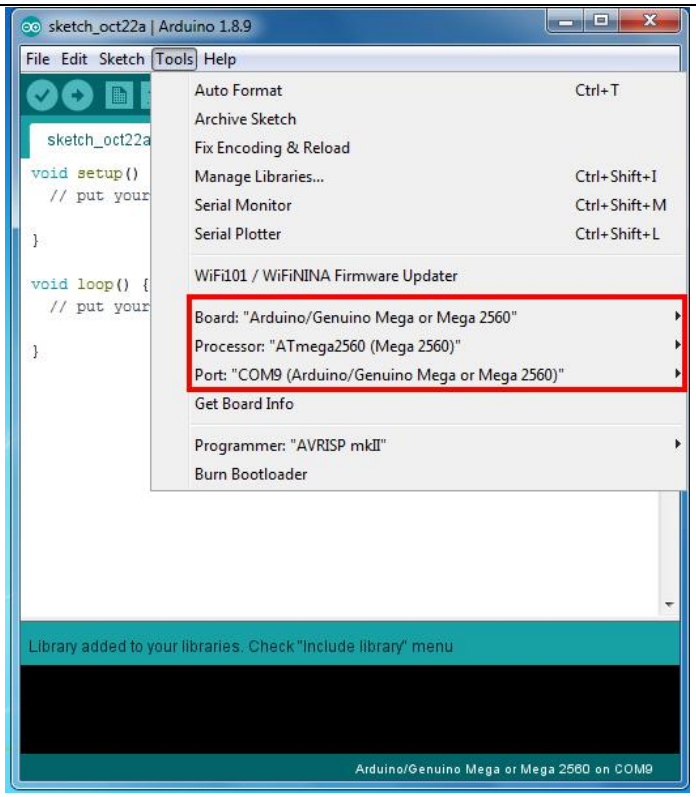
Step 3 Library Installation

Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-2.zip

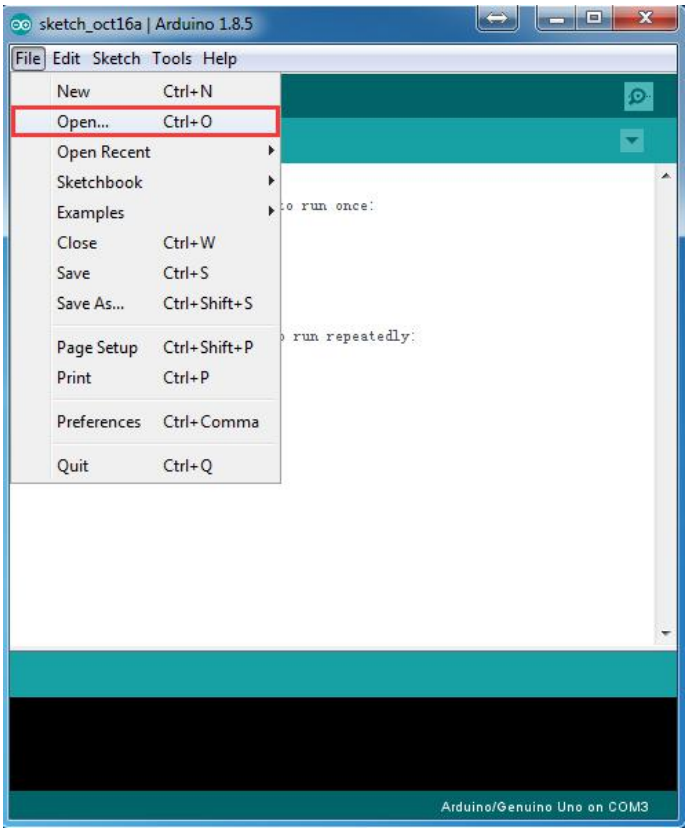
Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project.



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



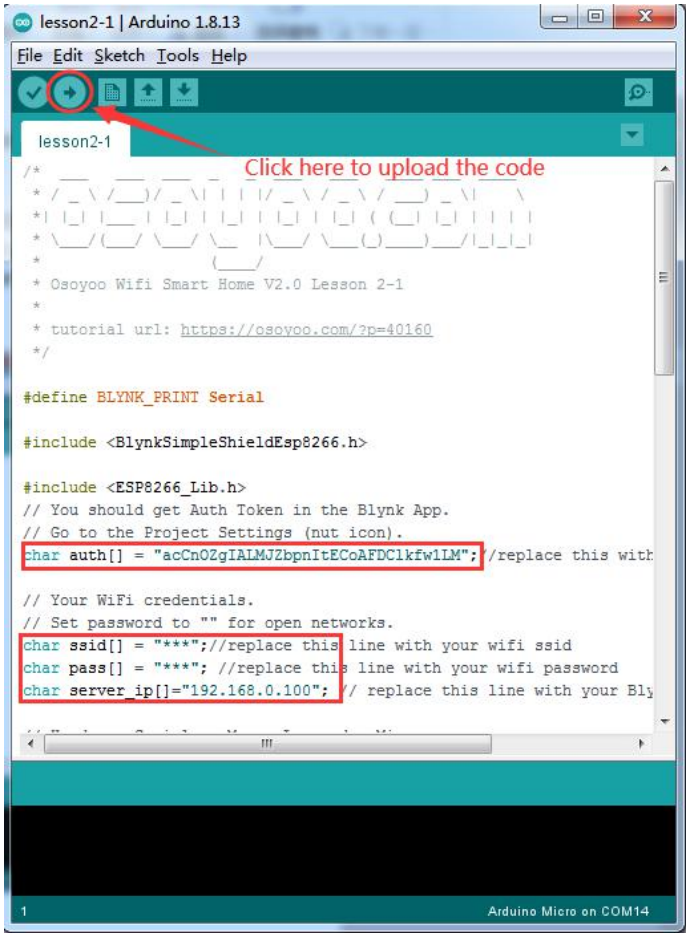
Note: In the sketch, find this line as following:
char auth[] =”sM1i_rSJjDlQxesfuhOzrL0h9NiivJkn”;

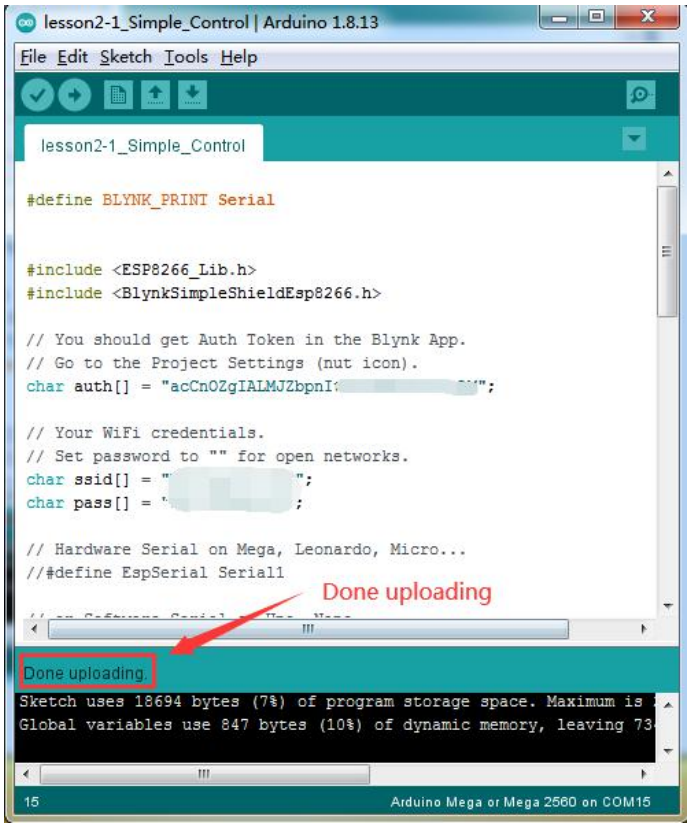
Replace **sM1i_rSJjDlQxesfuhOzrL0h9NiivJkn** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading





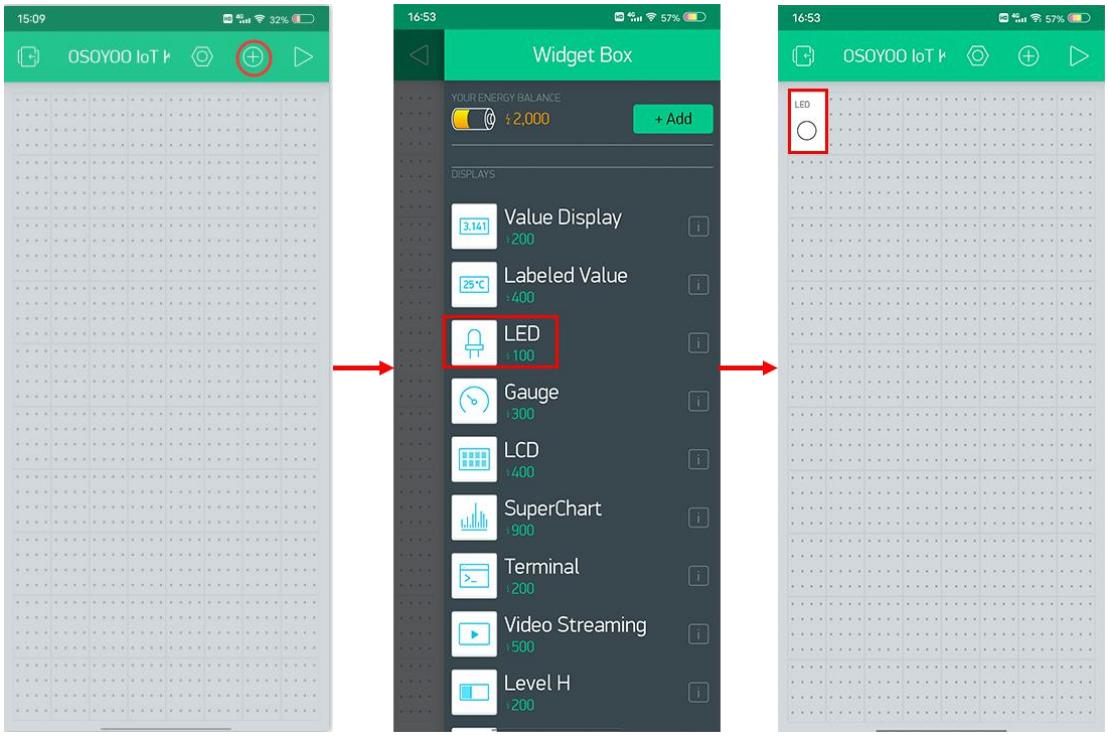
Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

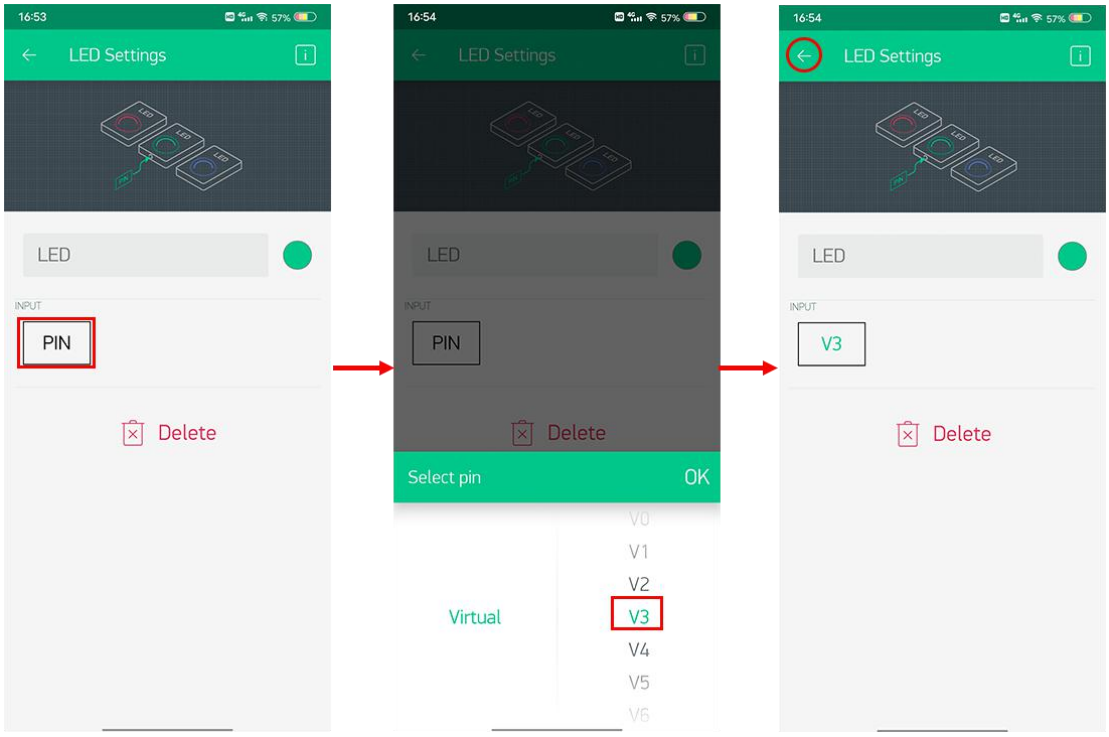
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

Overhere we need to add an LED wiget. Follow the next operations:
Open your project page and press the “+” button to add the “LED” Widget.

LED Widget settings:
Name Column: You can name your widget, whatever you like.
INPUT: V3

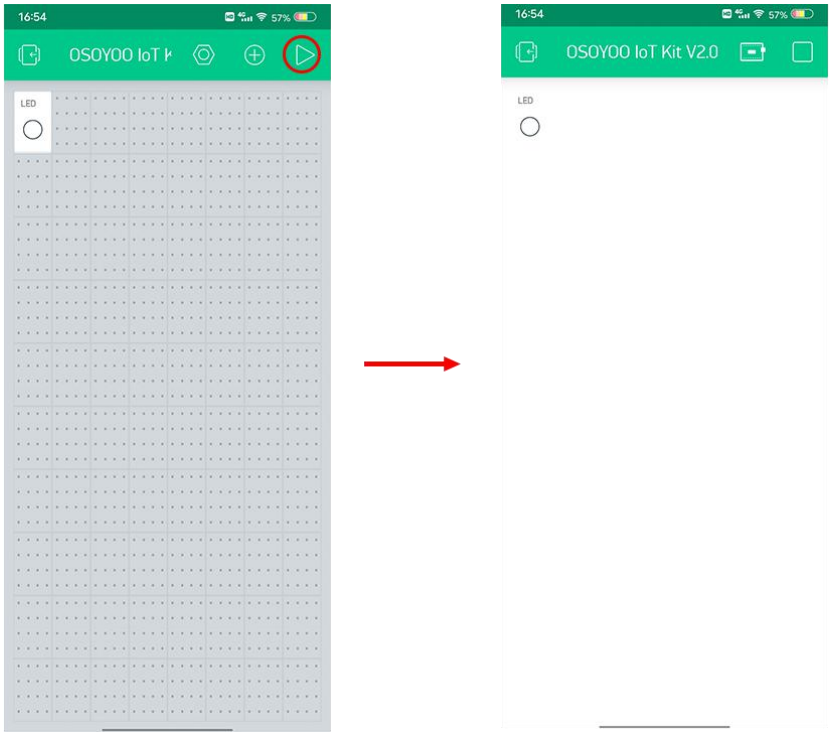




After you finished all above operations, open the Serial Monitor and you will see the connection status. Now open the Blynk APP, press the PLAY button "▷ ".

This will switch the APP from EDIT mode to PLAY mode where you can interact with the hardware.

While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.



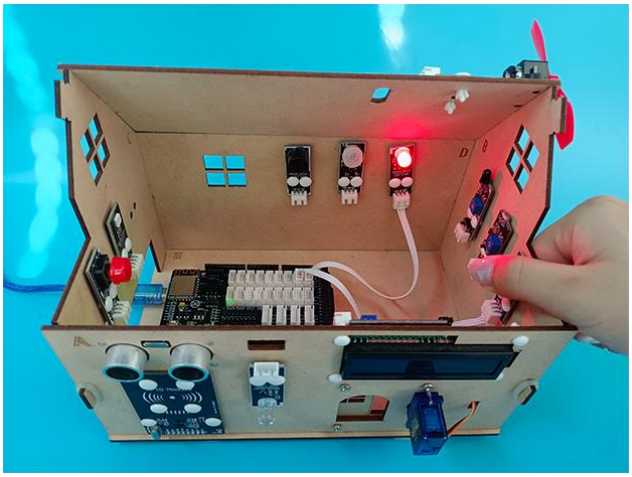
HOW TO PLAY

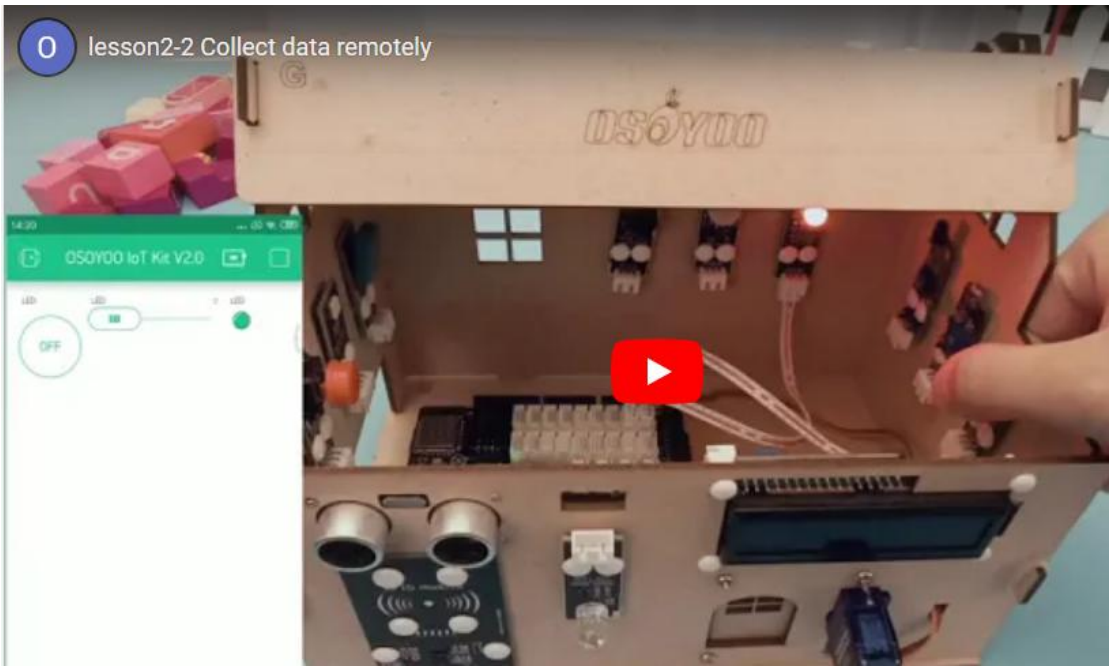
When you Press/Release the button, and then you will find that the virtual LED widget on the Blynk APP will be turned on/off.

At this time, you can try to use this code to detect the status of your other environmental sensors in digital mode.

Please note: If there is a status switch on your sensor, please flip the switch to the side with the letter "D".

If you find the device is offline, please check your code, wiring and app settings, and try again.





https://youtu.be/gu2GupfbA-k

2-3 Remote temperature and humidity monitoring

OBJECTIVE

In this lesson, we will use this IoT kit to remotely monitor the home environmental temperature and humidity, and display the data to the Blynk APP.

PARTS & DEVICES

HARDWARE

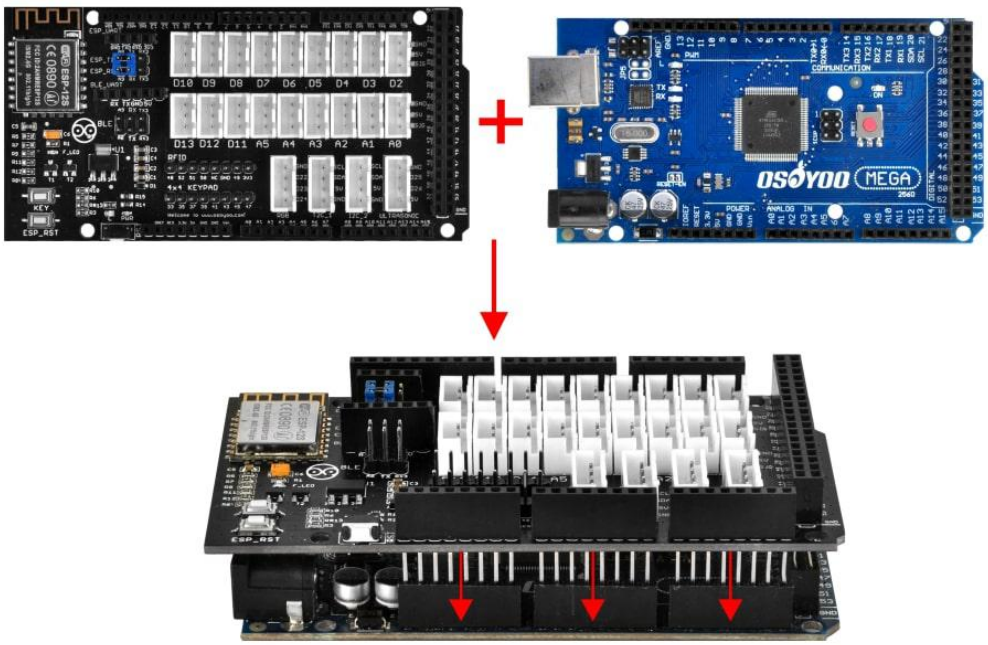
Osoyoo Mega2560 (fully compatible with Arduino MEGA2560) x 1
OSOYOO IoT Wifi Shield x 1
DHT11 Temperature/Sensor Module x 1
LCD Display Module x 1
USB Cable x 1
3-pin PnP Cable x 1
4-pin PnP Cable x 1

SOFTWARE

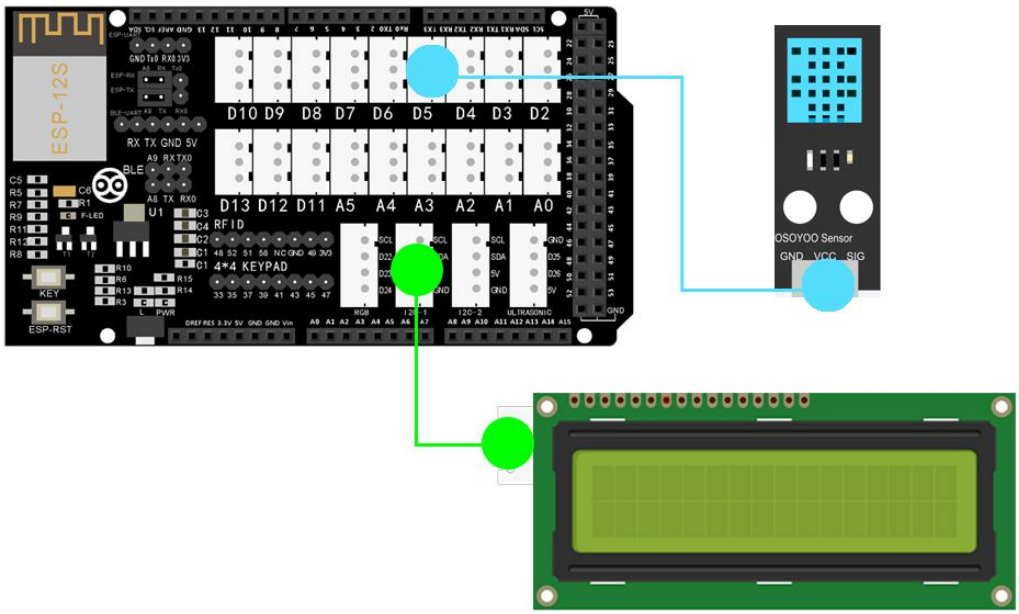
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
DHT11 library
Adafruit library
I2C library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:
Here we connect the signal pin (S) to digital pin D5.

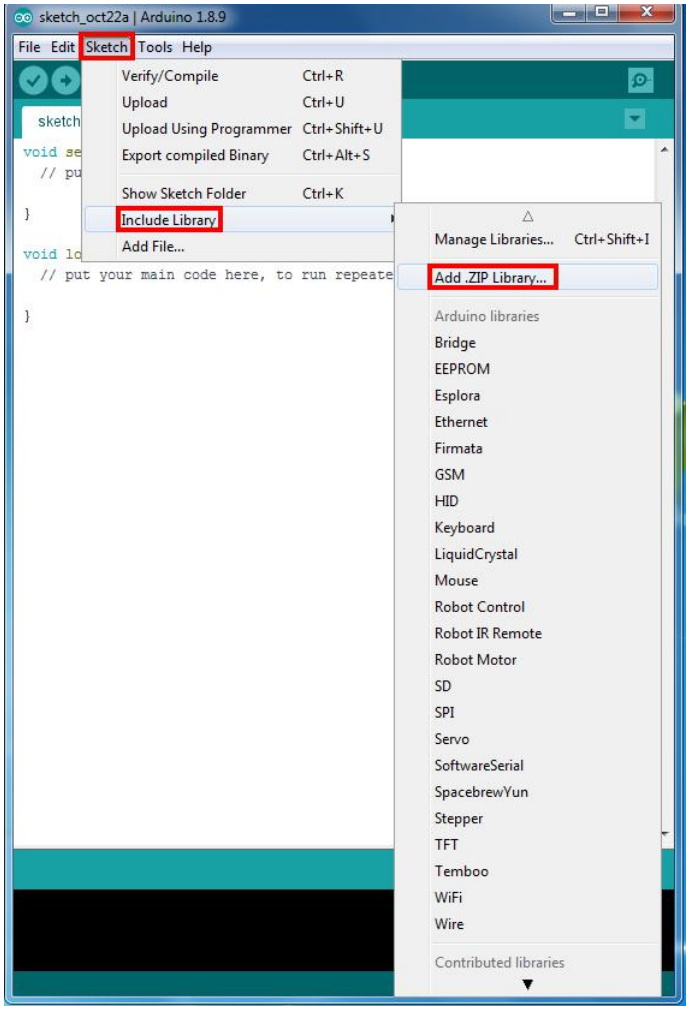


Step 1 Prerequisite

Step 2 Install latest Arduino IDE

Step 3 Library Installation

To use the DHT11 sensor here, you also need to install the DHT11 library, Adafruit library and I2C library as above operations.

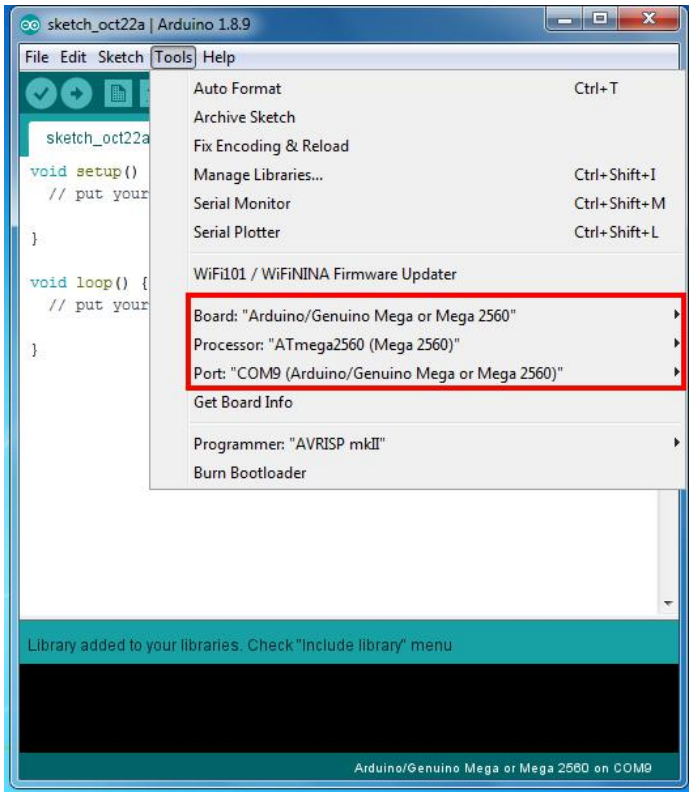


Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-3.zip

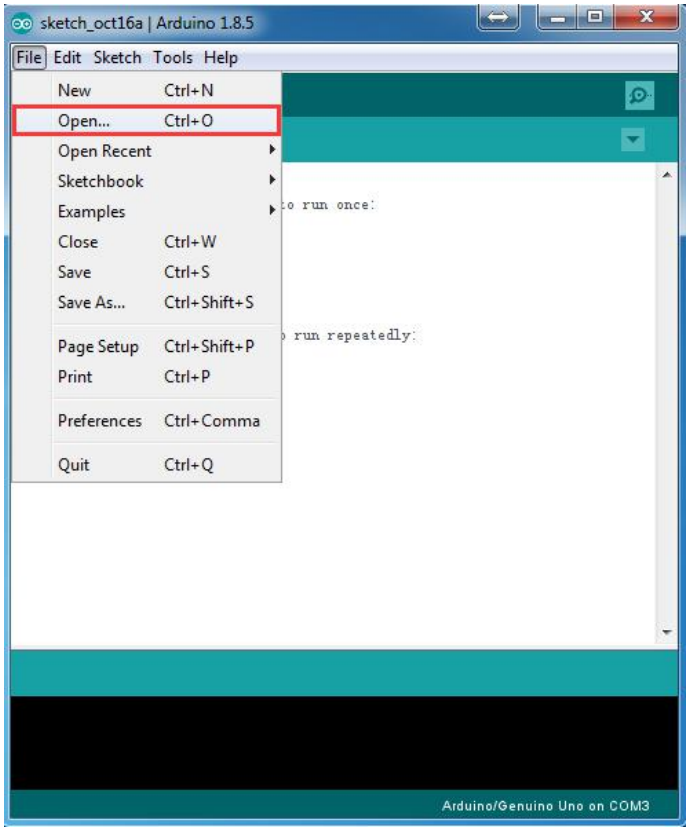
Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino IDE.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project.



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



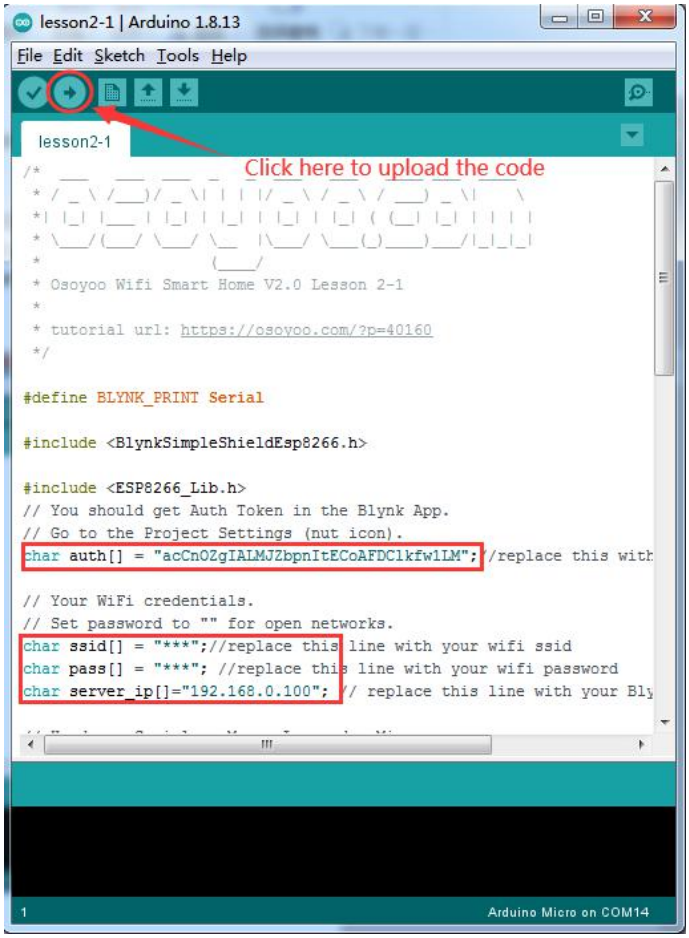
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

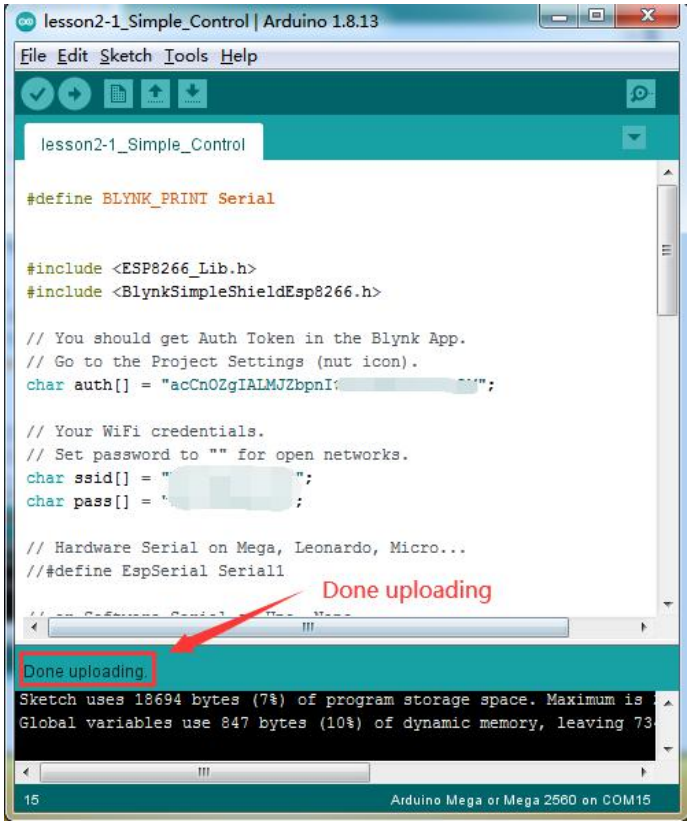
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14


lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "Ready (ping: 25ms)" ending message.

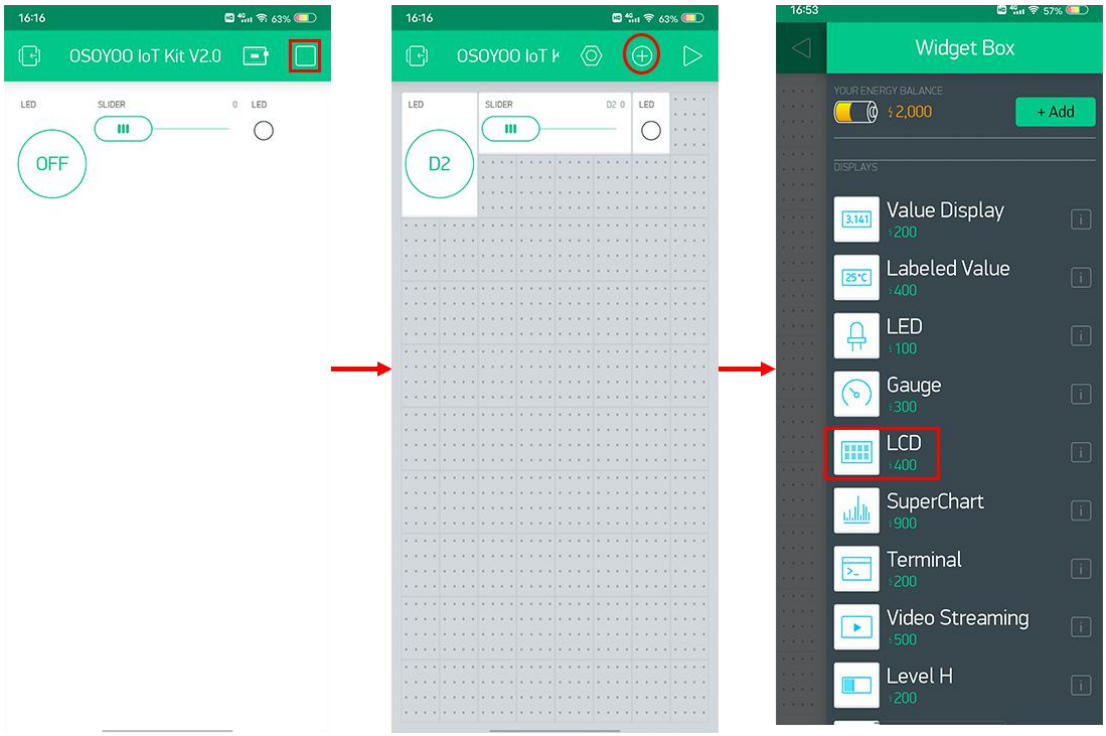
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

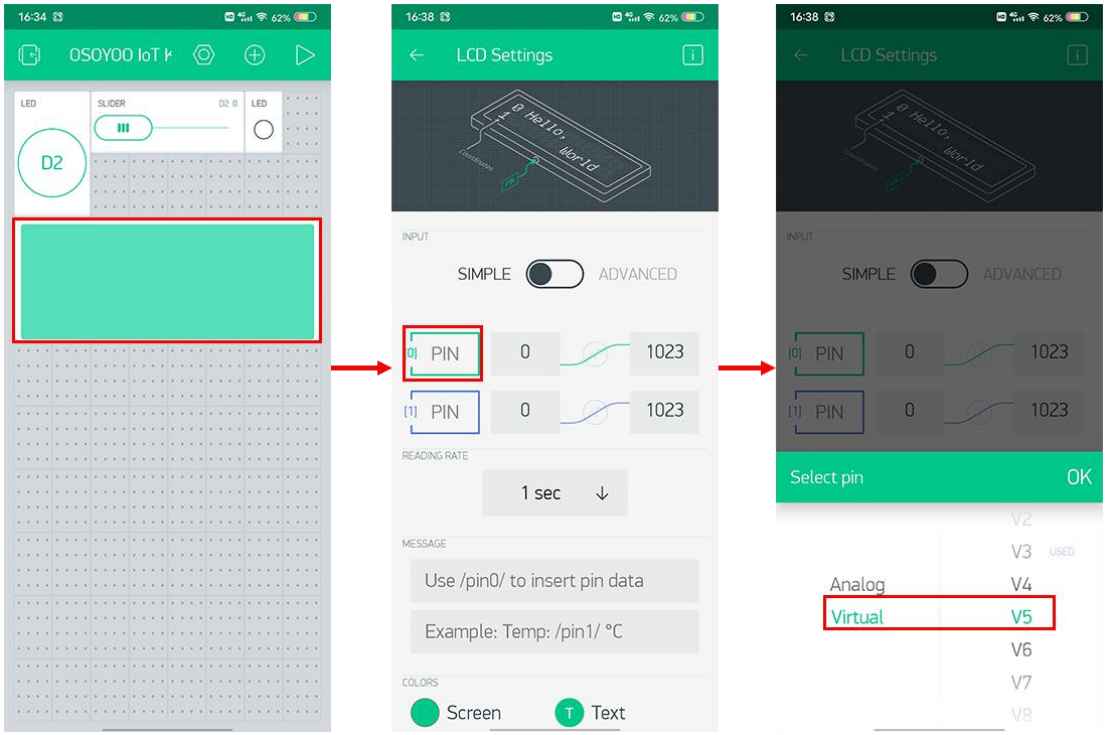
Add a Widget in Blynk (legacy) APP

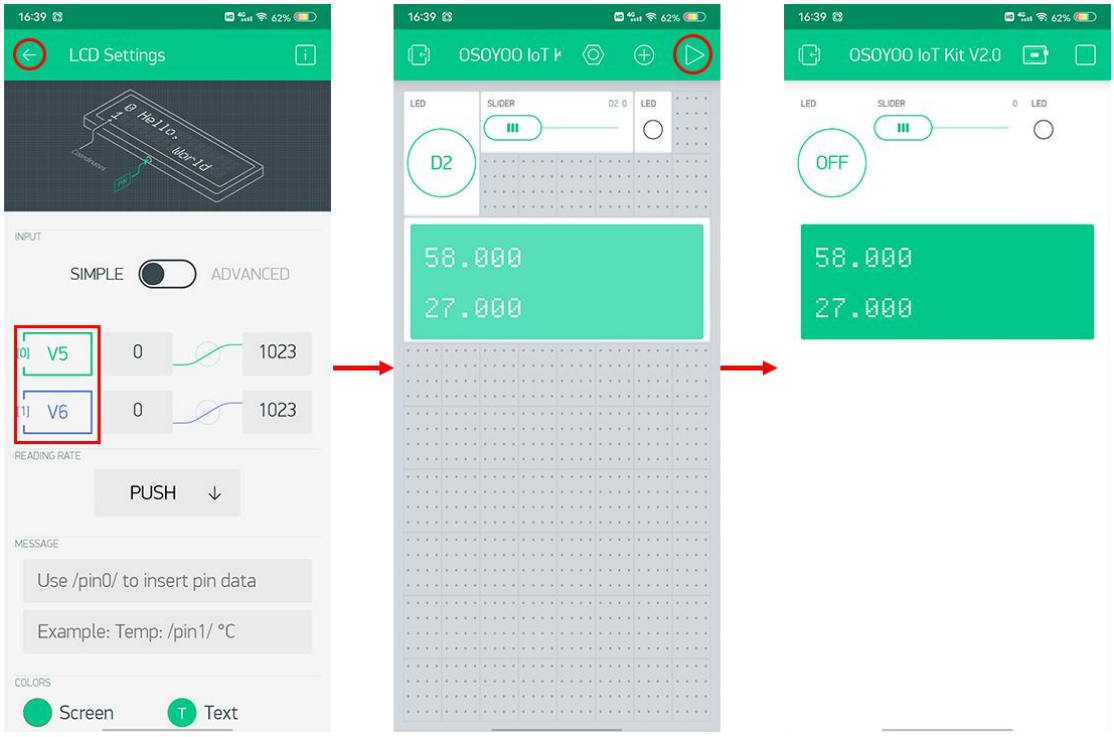
Open your project page and press the "+" button to add the "LCD" Widget.
Widget settings:

Name Column: You can name your widget, whatever you like.
PIN V5, value: H
PIN V6, value: T

Here, you can try to change the LCD Widget to Labeled Value Widget by yourself, so that you can add the symbols of the corresponding data, but you need more Blynk energy.



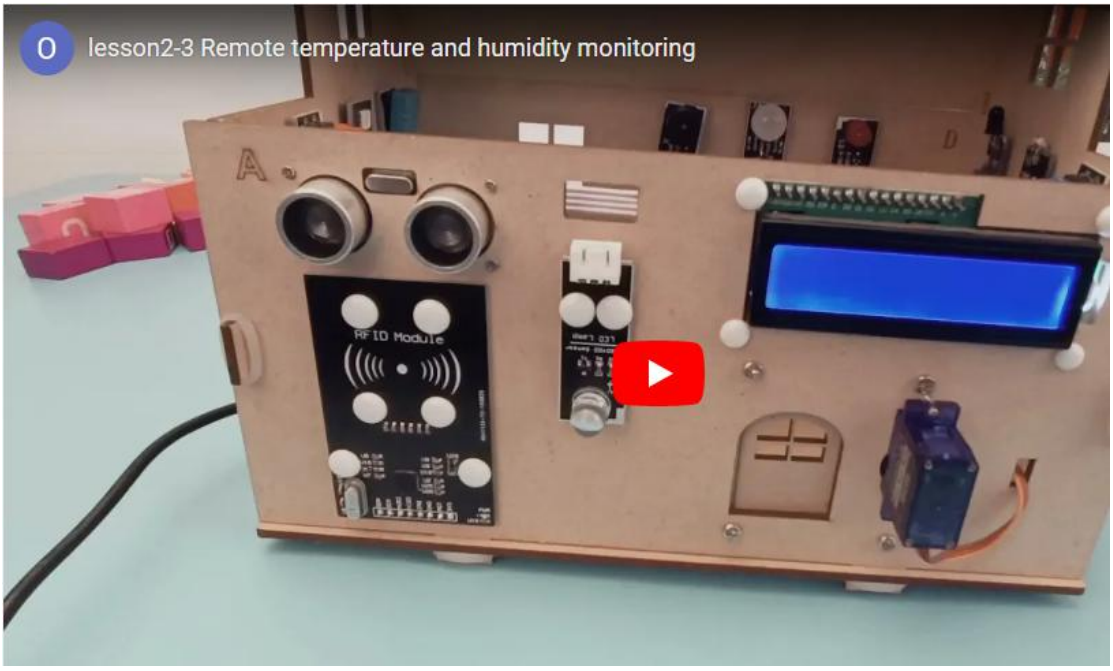




After you finished all above operations, press the "▷ " button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware. While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

HOW TO PLAY

Now, you can see the temperature and humidity remotely!
If you find the device is offline, please check your code, wiring and app settings, and try again.



https://youtu.be/T0XFHWZbVLo

2-4 Collect ultrasonic sensor

OBJECTIVE

Here, we will demonstrate how to use this kit to remotely collect the data detected by the ultrasonic ranging sensor and display it on the mobile phone through the Blynk APP.
Some people may think that this sensor is rarely seen in ordinary times. In fact, we often use it in apartment garage systems and smart security systems.

PARTS & DEVICES

HARDWARE

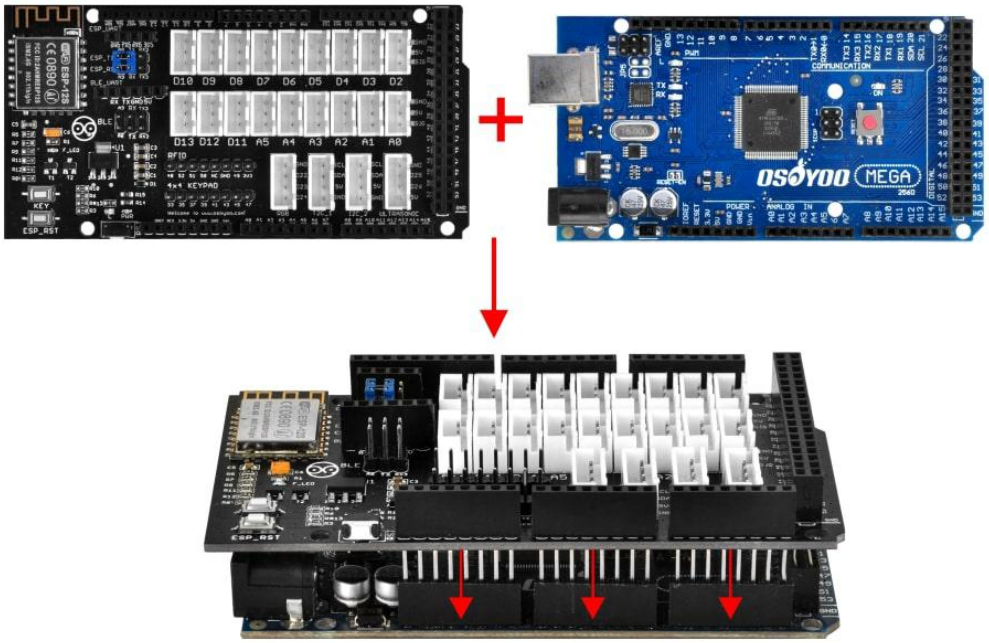
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
Ultrasonic Module x 1
USB Cable x 1
4-pin PnP Cable x 2

SOFTWARE

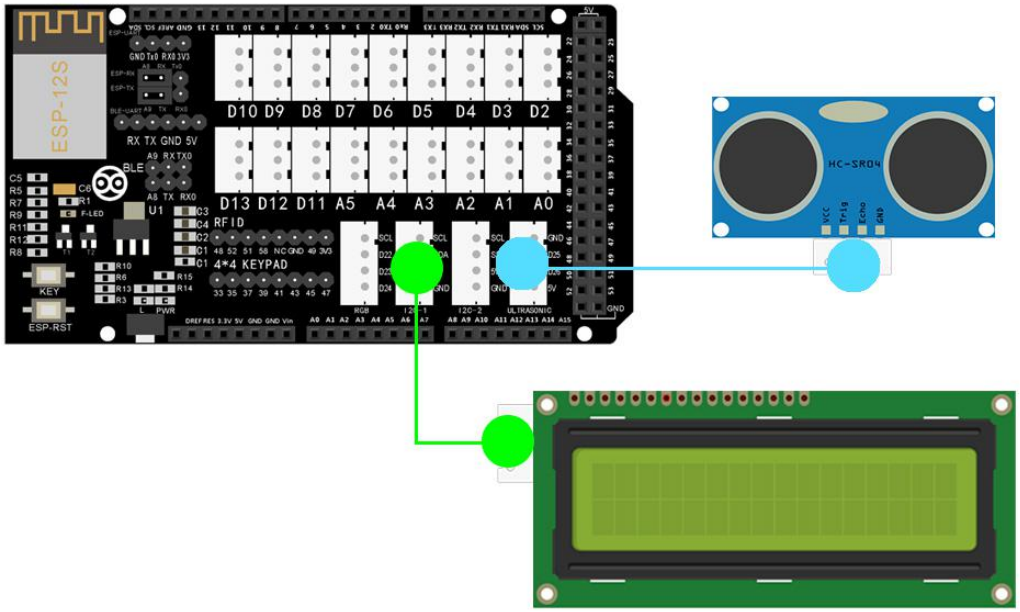
Arduino IDE (version 1.6.4+)
Blynk APP
WifiEsp Library
I2C library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:
Here we connect the Echo to port D25, connect the Trig to port D26.



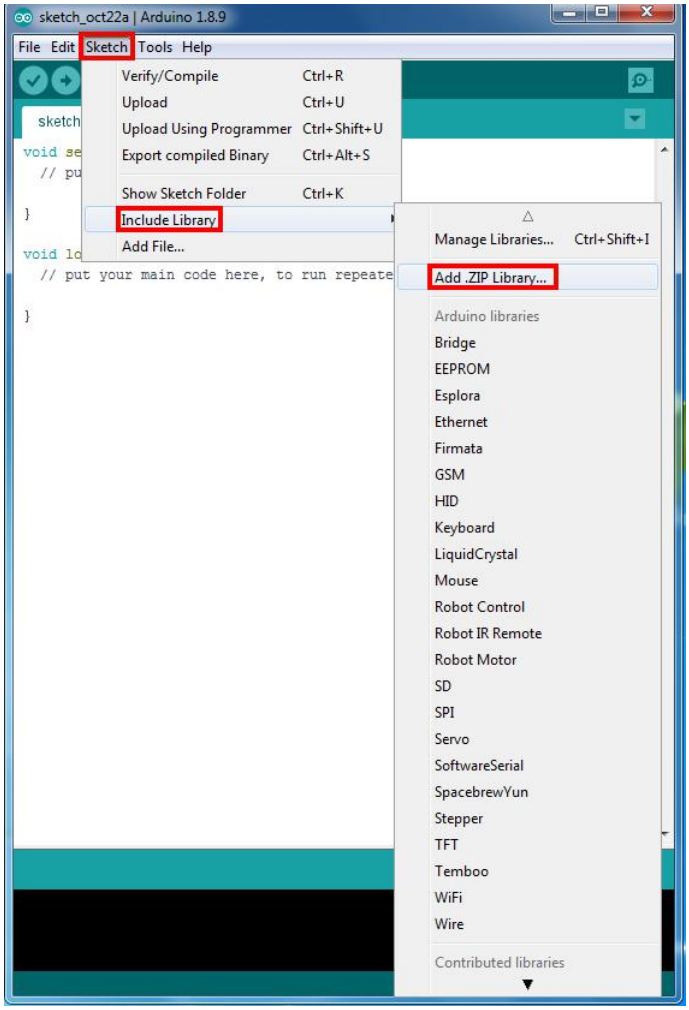
HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

Step 3 Library Installation

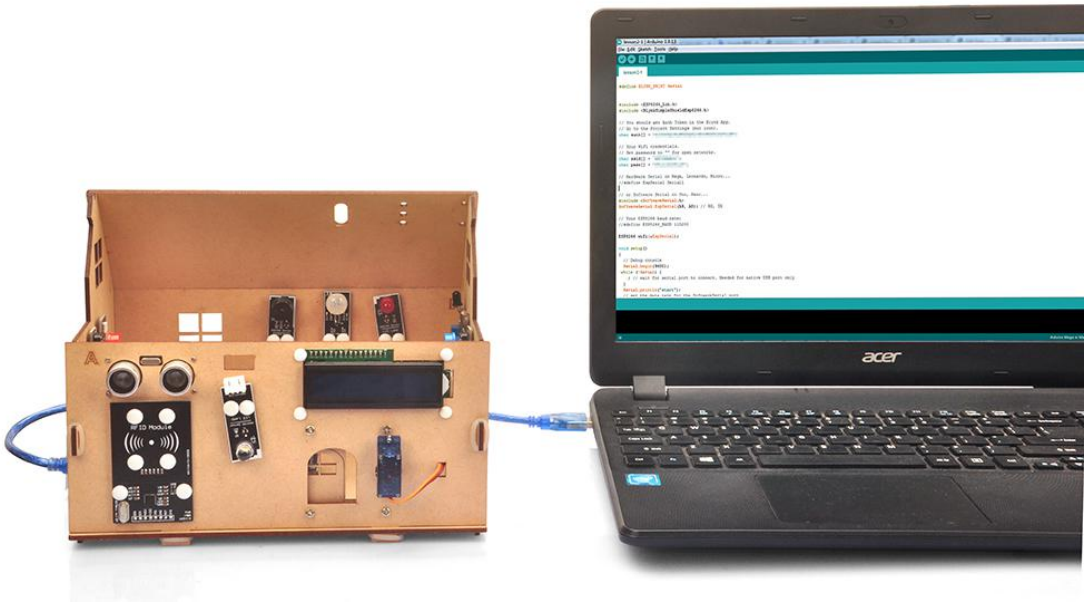
You also need to install the I2C library as above operations.



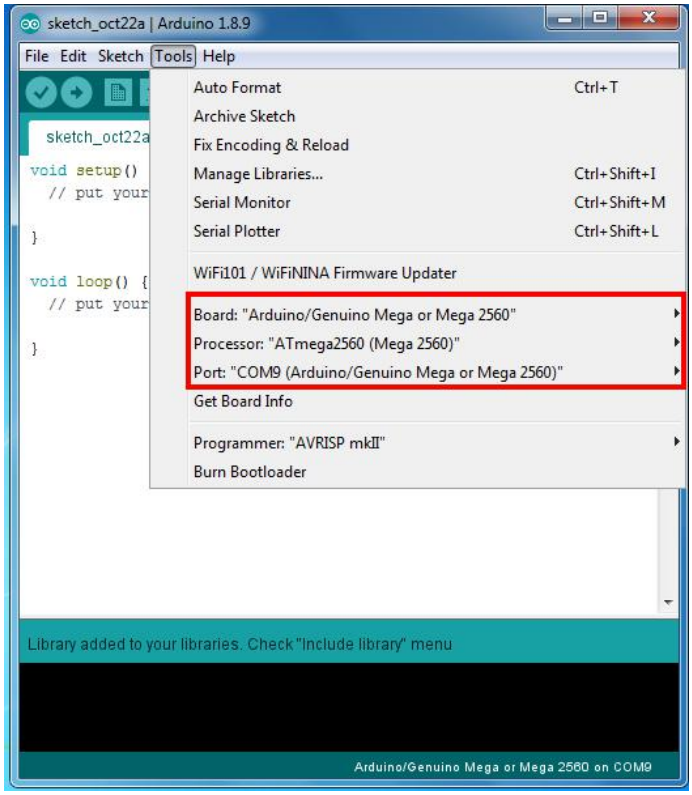
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-4.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

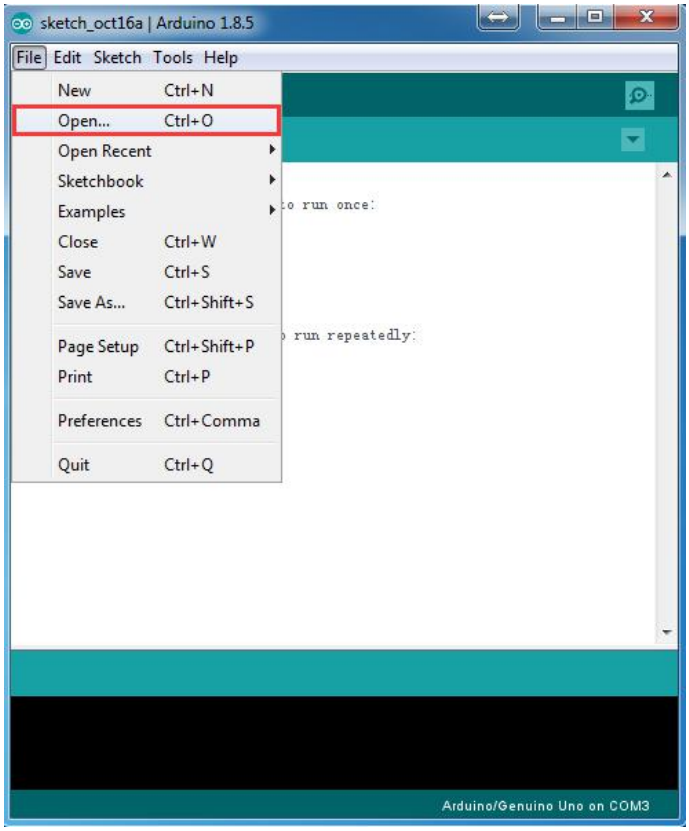
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino IDE.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project.



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino IDE.



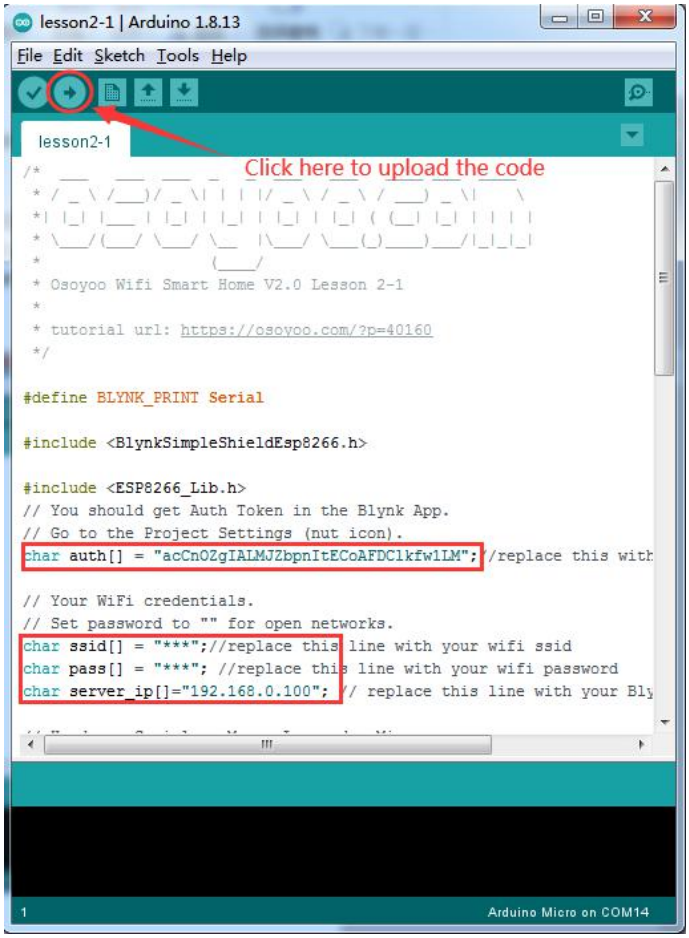
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

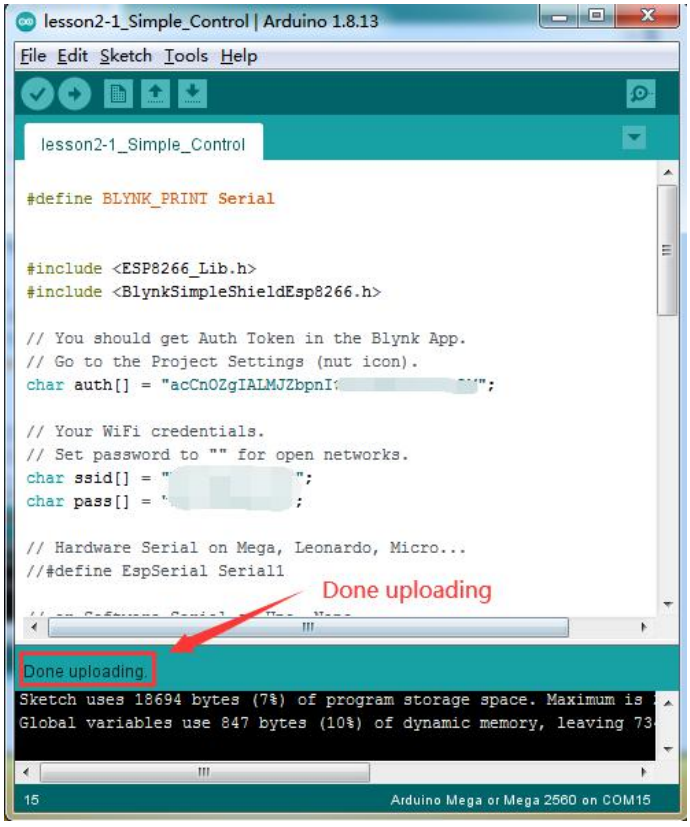
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14


lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

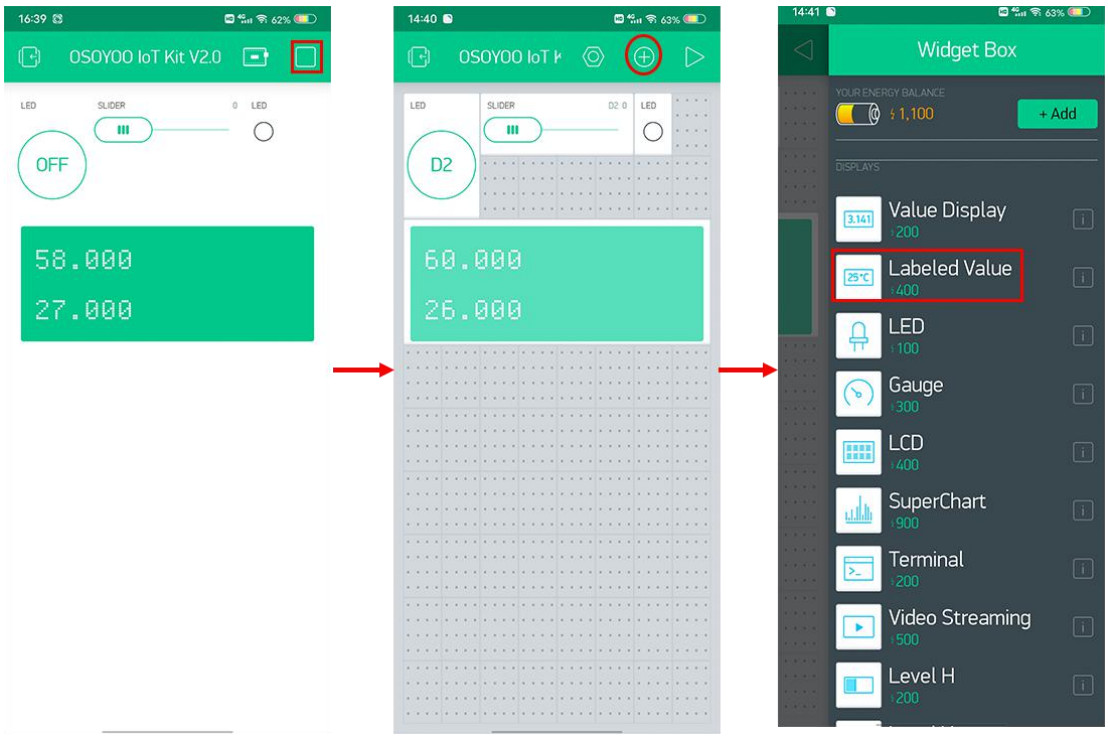
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

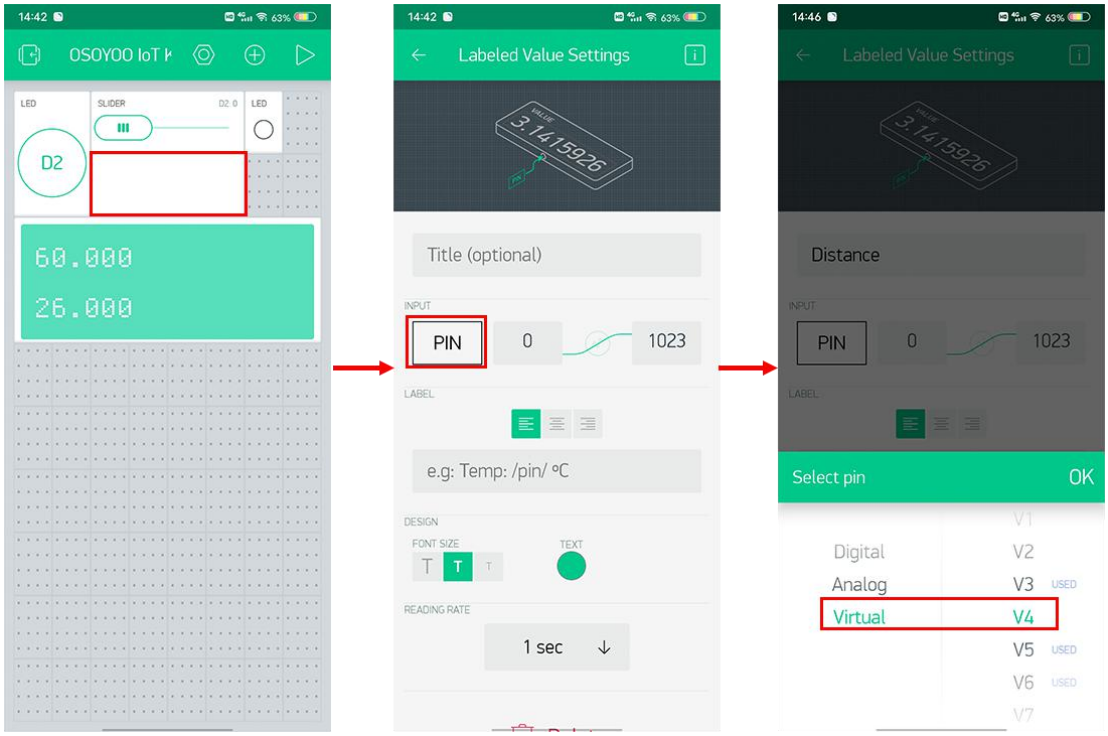
Add a Widget in Blynk (legacy) APP

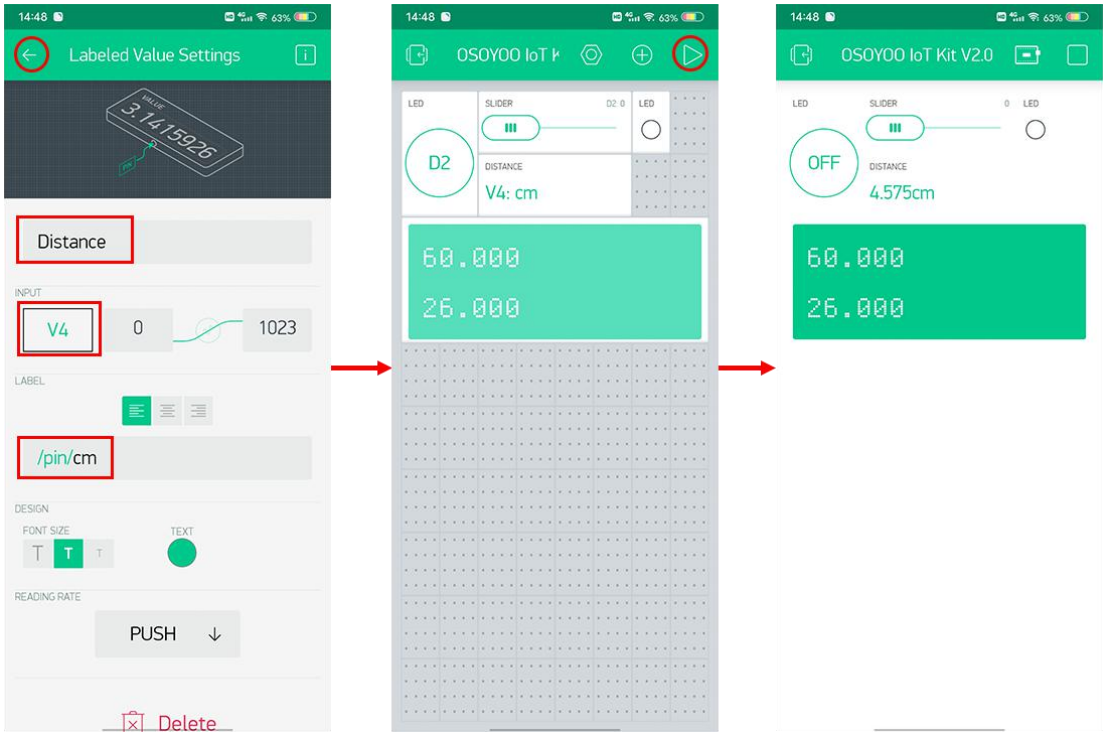
Overhere we need to add a Labeled Value wiget to get the real time distance value. Follow the next operations:

Widget Setting 1
Open your project page and press the "+" button to add the "Labeled Value" Widget.

Name Column: Distance
PIN: V4
LABEL: pin/cm







After you finished all above operations, press the "▷ " button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

Widget Setting 2

In addition, you can also add GAUGE widget or Level H widget to demonstrate the changes in the values collected by the sensor.

GAUGE widget settings:
Name: Distance
INPUT: V4:0-300
LABEL: Distance:/pin/cm

Widget Setting 3

Level H widget settings

Title:Distance

INPUT: V4:0-300

Keep orther settings as default

HOW TO PLAY

Now you can remotely view the value detected by the ultrasonic distance sensor through the Labeled Value widget, GAUGE widget or Level H widget, you can clearly see the changes in the values collected by the sensor

If you find the device is offline, please check your code, wiring and app settings, and try again.

https://youtu.be/GrvPf-Bu4kk

2-5 RFID Access Control

OBJECTIVE

In this lesson, we will show how to use the Osoyoo IoT Kit to build a RFID access control system. Before starting this lesson, you can get more information about the RFID module from this link.

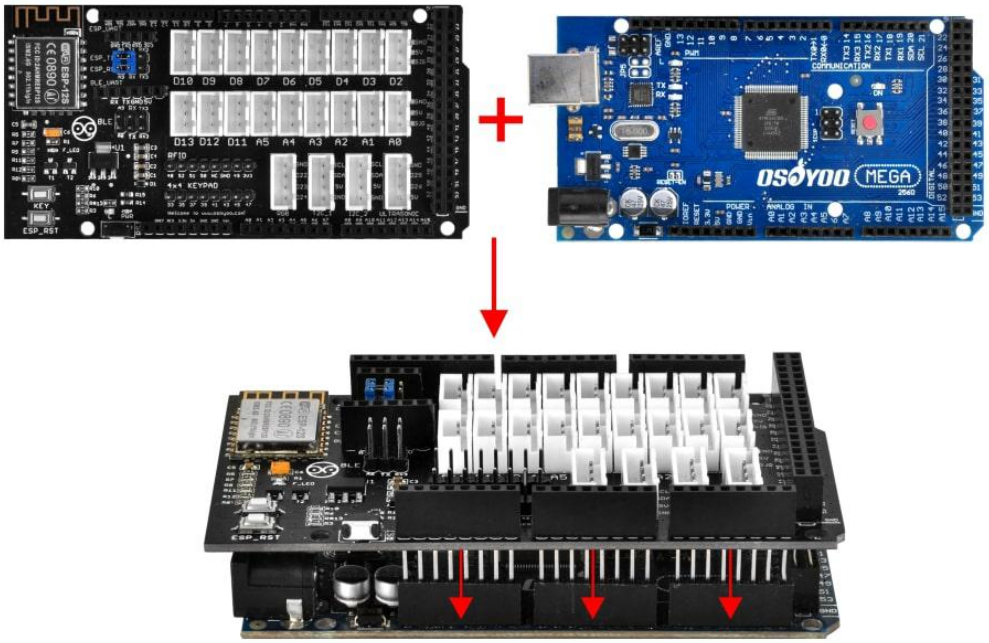
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
RFID Module x 1
RFID Card x 1
1602 LCD Screen PnP Module x 1
USB Cable x 1
8pin 12cm Female to Female Cable x 1
4-pin PnP Cable x 1

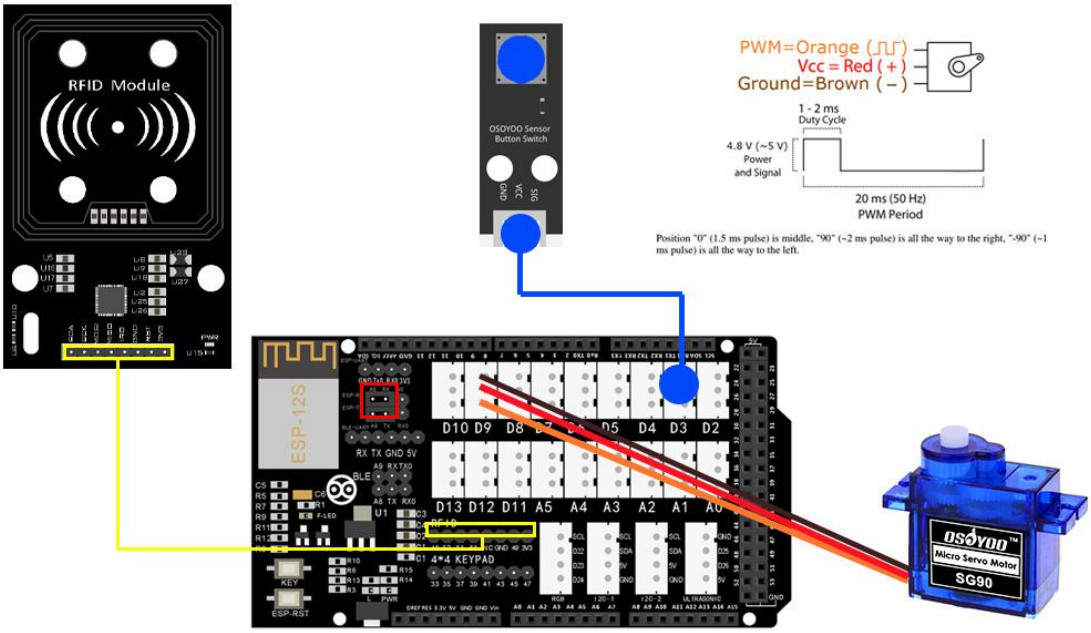
SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
RFID library
Servo library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:
RFID Module – RFID Port
Please confirm whether the wiring is correct according to the VCC and GND pins



HOW TO CODE

Step 1 Prerequisite

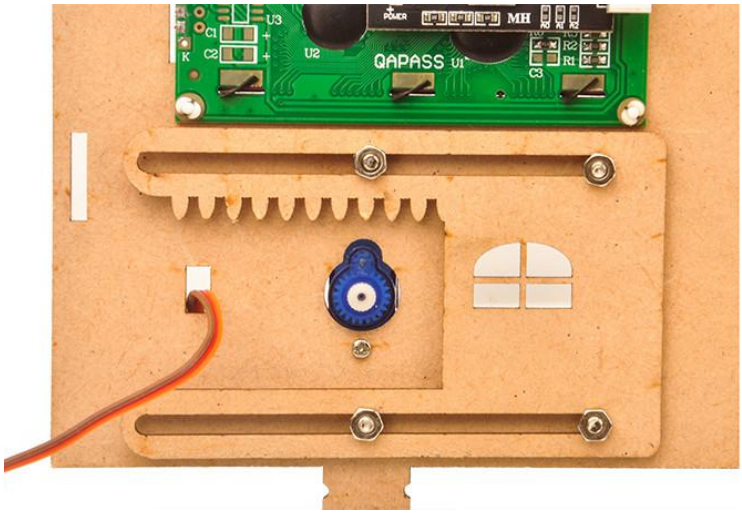
Step 2 Install latest Arduino IDE

Step 3 Library Installation

You also need to install the Servo library and RFID library as above operations.

Step 4 Servo Test

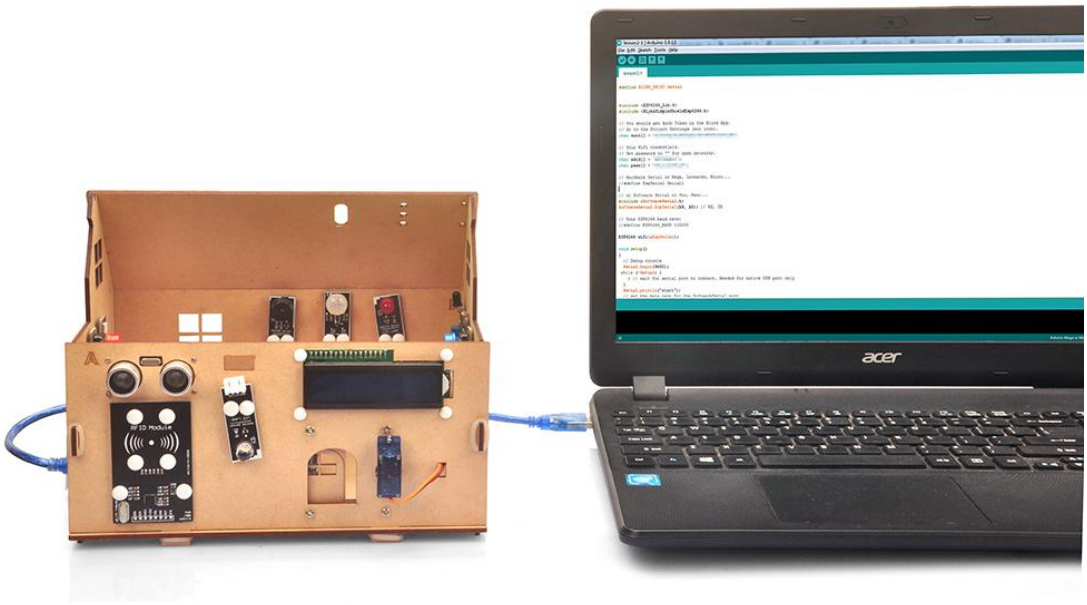
Before upload the code, we should adjust the servo angle first.
Please remove the items from the wooden house's door area:



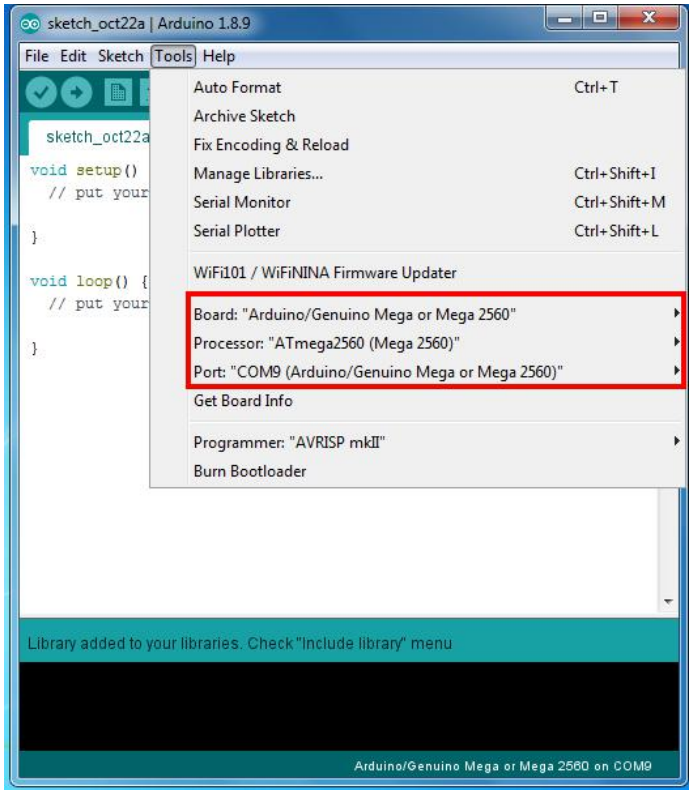
Then please download the servo test code from this link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/servo_config.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

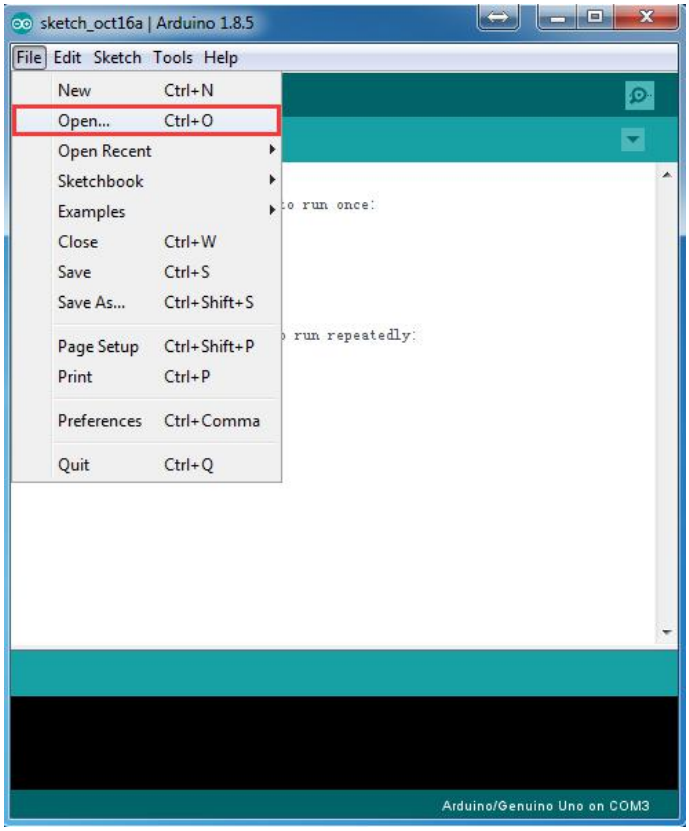
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino IDE.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project.

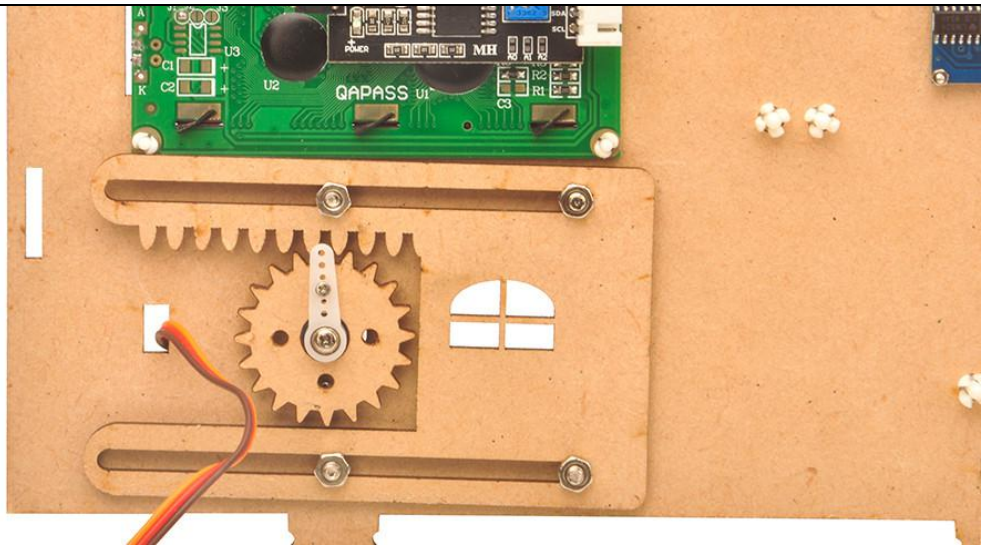


Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino IDE.

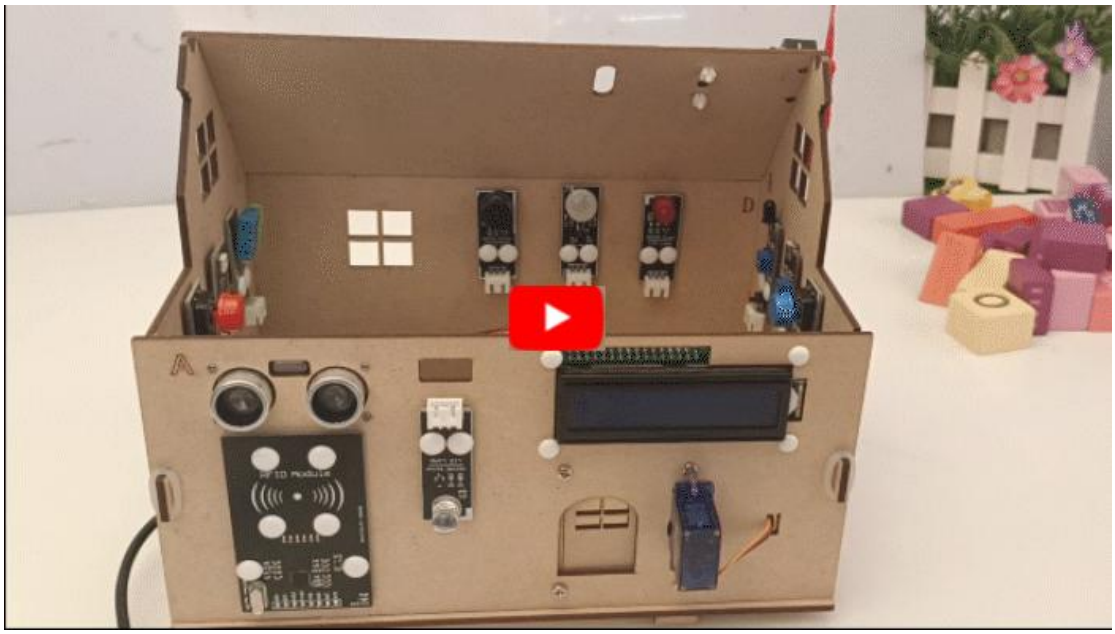


Upload the sketch to the board. Wait until you see something like this:
Done uploading

Step 8 After upload the code, the servo will turn to 0 degree position, now we install the wooden gear and the white fixed arm:



Step 9 Press the button, the door can be simulated to open, when released, the door can be simulated to close.



https://youtu.be/nb8Vamj7zi0

Step 10 After above operations are completed, please download the code from following link, unzip it, and follow the above Step 5 – Step 7 to upload the code:

https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-5.zip

You need to edit the code:

****

Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

****

Add your own RFID Card ID:

After completing the above steps and uploading the code, open the IDE serial monitor, you can see that the system is initialized successfully, and then put your RFID card in the RDID module recognition area, you will see your serial monitor will display your The card information is as follows:

RFID START!
Find the card!
Card type: UnknownThe card's number is : 71BA2F2ECA

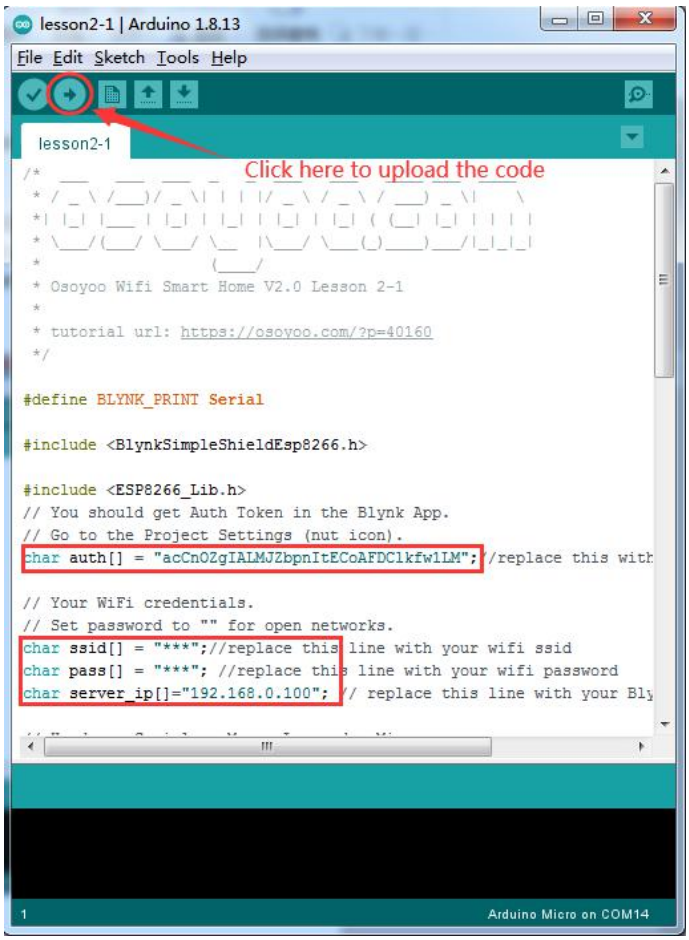
Then find these lines:
if(id[0]==0x32 && id[1]==0xDA && id[2]==0x94 && id[3]==0x10)

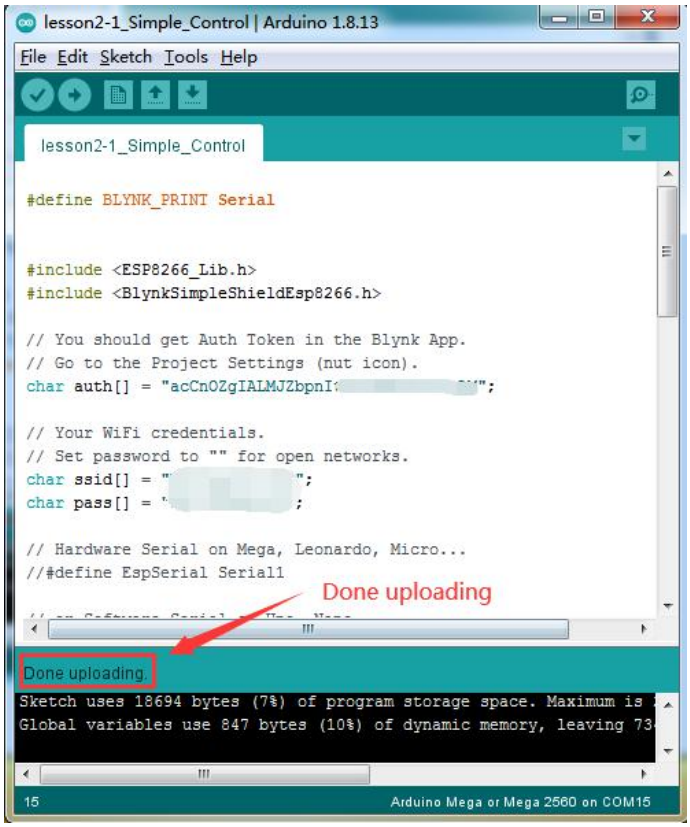
And change the id data to your own card number，so the changed code is:
if(id[0]==0x71 && id[1]==0xBA && id[2]==0x2F && id[3]==0x2E)

In the same way, you can add or modify your card information at will, and finally, re-upload the code to the Arduino IDE and continue the experiment.
****

Upload the sketch to the board. Wait until you see something like this:

Done uploading





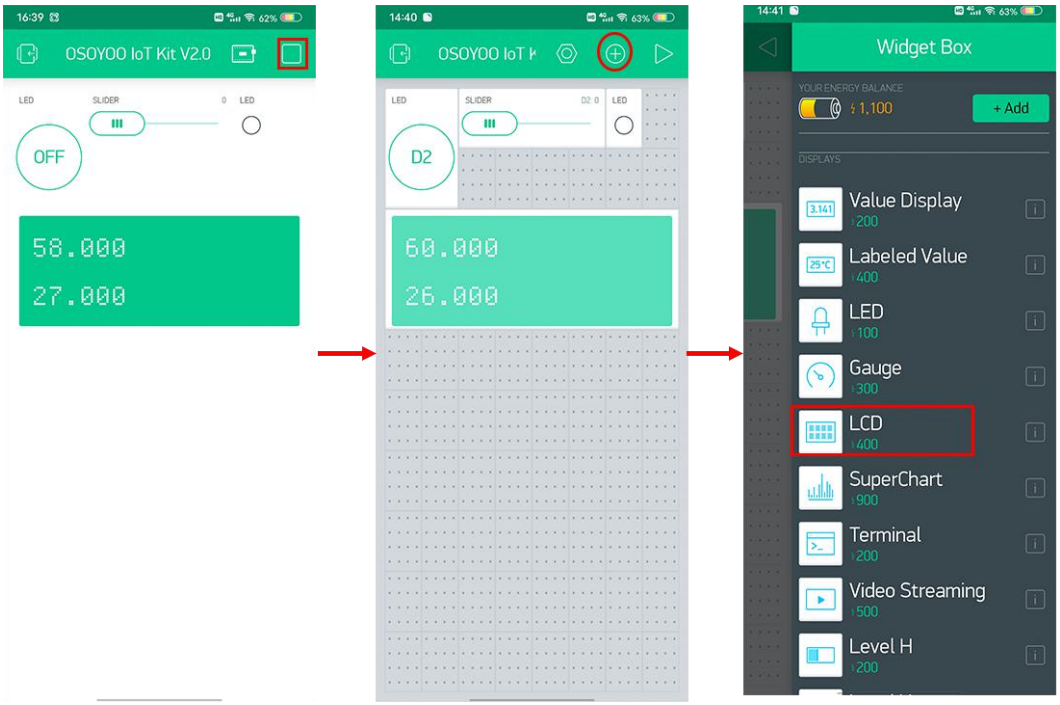
Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

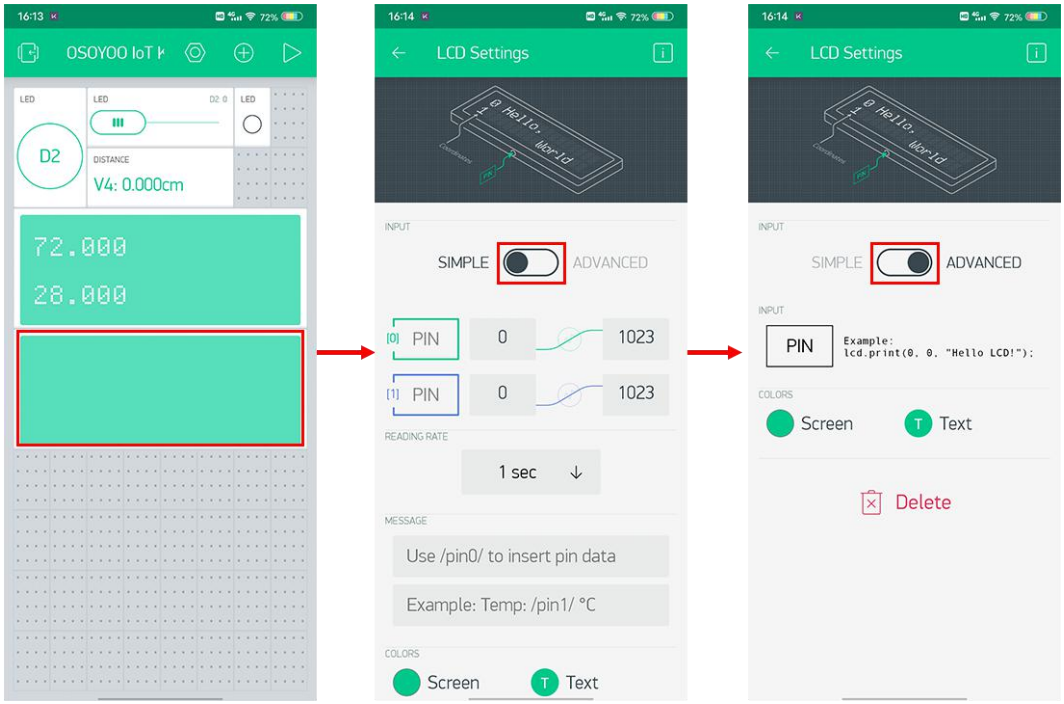
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

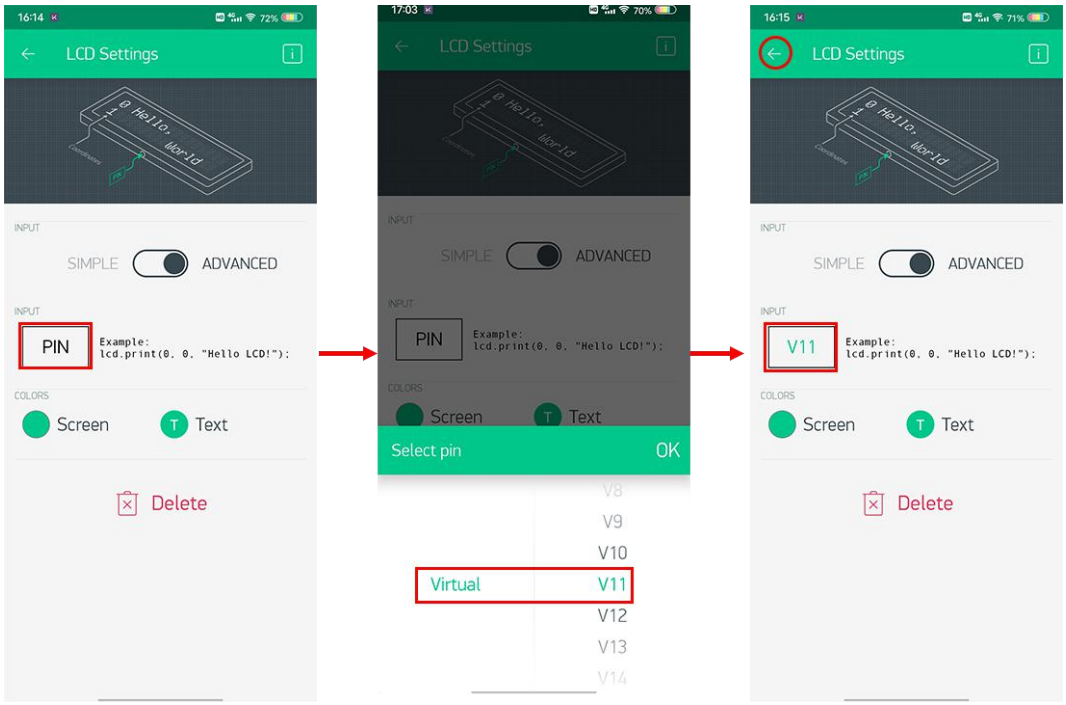
Add a Widget in Blynk (legacy) APP

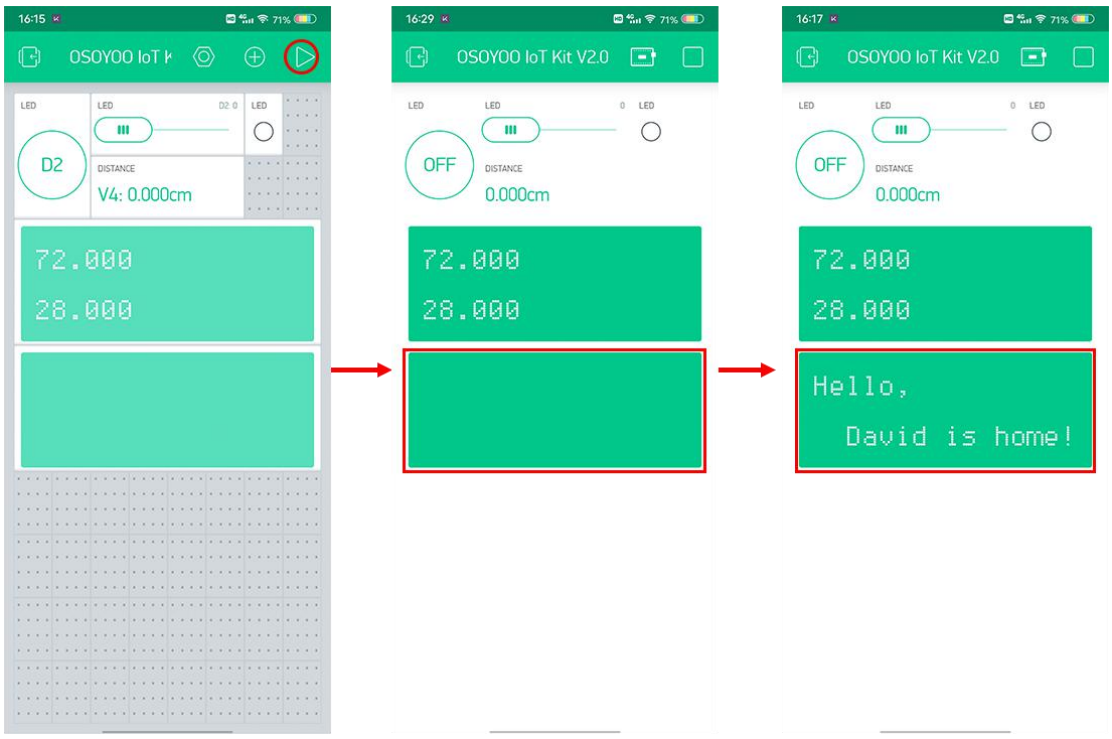
Open your project page and press the "+" button to add the "LCD" Widget.

Widget settings:
Name Column: You can name your widget, whatever you like.
INPUT ADVANCED
INPUT V11


OSOYOO IoT Kit V2.0
OFF
58.000
27.000
D2
60.000
26.000
Widget Box
+ Add
Value Display
Labeled Value
LED
Gauge
LCD
SuperChart
Terminal
Video Streaming
Level H


D2
V4: 0.000cm
72.000
28.000
LCD Settings
SIMPLE
ADVANCED
PIN
0
1023
READING RATE
1 sec
MESSAGE
Use /pin0/ to insert pin data
Example: Temp: /pin1/ °C
COLORS
Screen
Text
Delete





After you finished all above operations, press the "▷ " button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

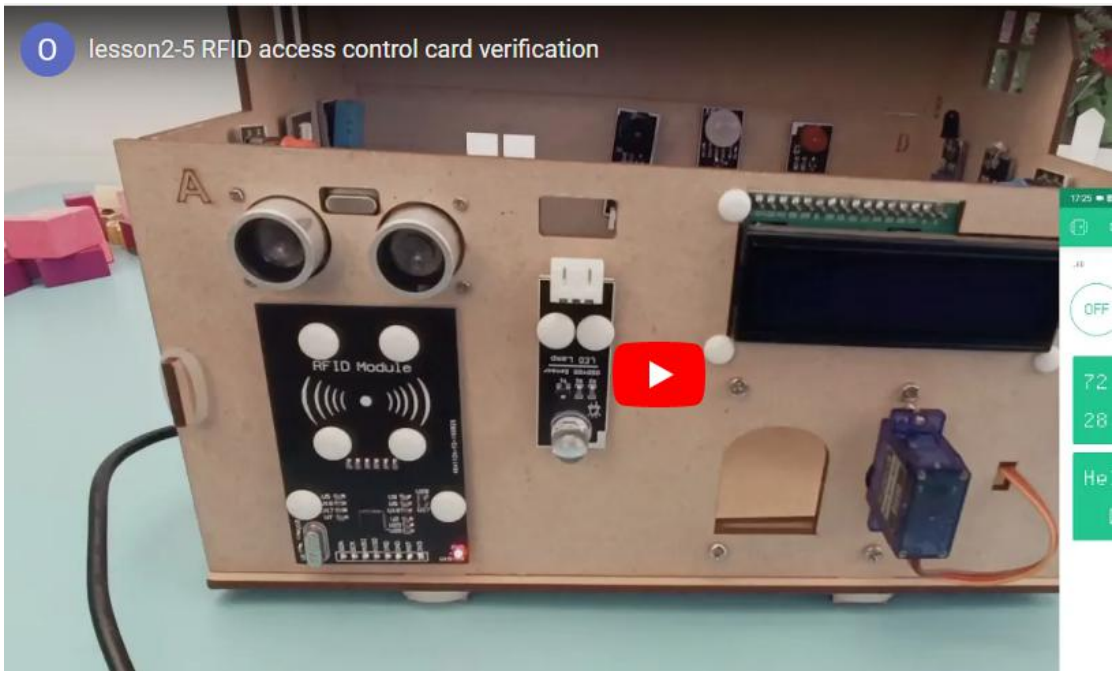
While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

HOW TO PLAY

After you finished all above operations, approximate the RFID card or the keychain to the reader.

Open the Serial Monitor and you will get some informations of the card.

If you update your card information to the code, when you use the corresponding card, let the reader and the tag closer, you will see that the corresponding name is printed in the serial monitor, and LED will be lit for a second.



https://youtu.be/_MqUcmUHwFo

2-6 Real-time Family Message Board

OBJECTIVE

Here, we will demonstrate how to use this kit to make a remote family message board. We can send messages to this message board remotely through the Blynk APP.

Before starting this lesson, you can get more information about the I2C 1602 LCD display from this link.

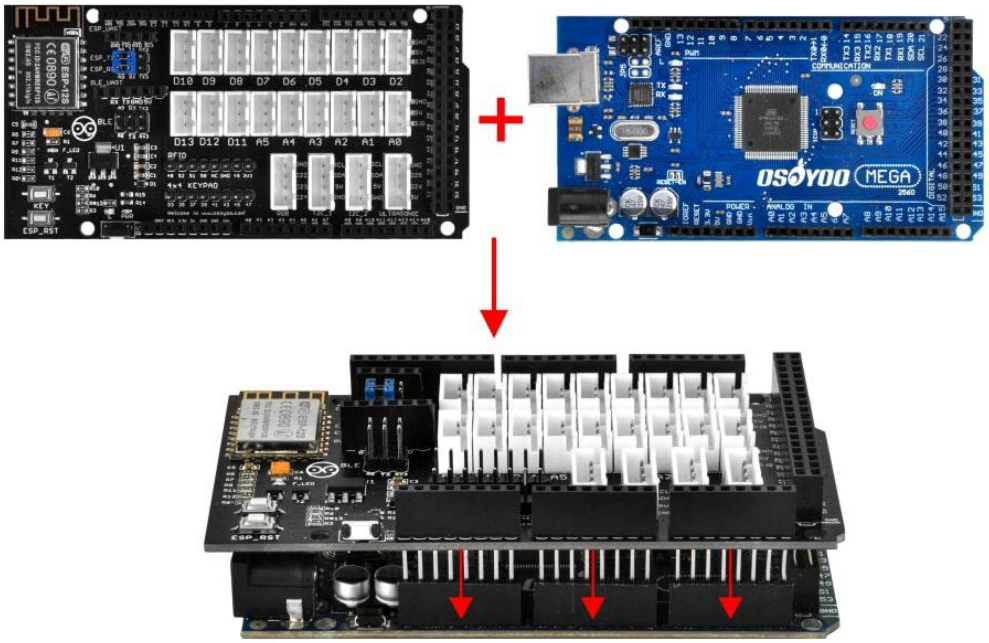
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
I2C 1602 LCD Display x 1
USB Cable x 1
4-pin PnP Cable x 1

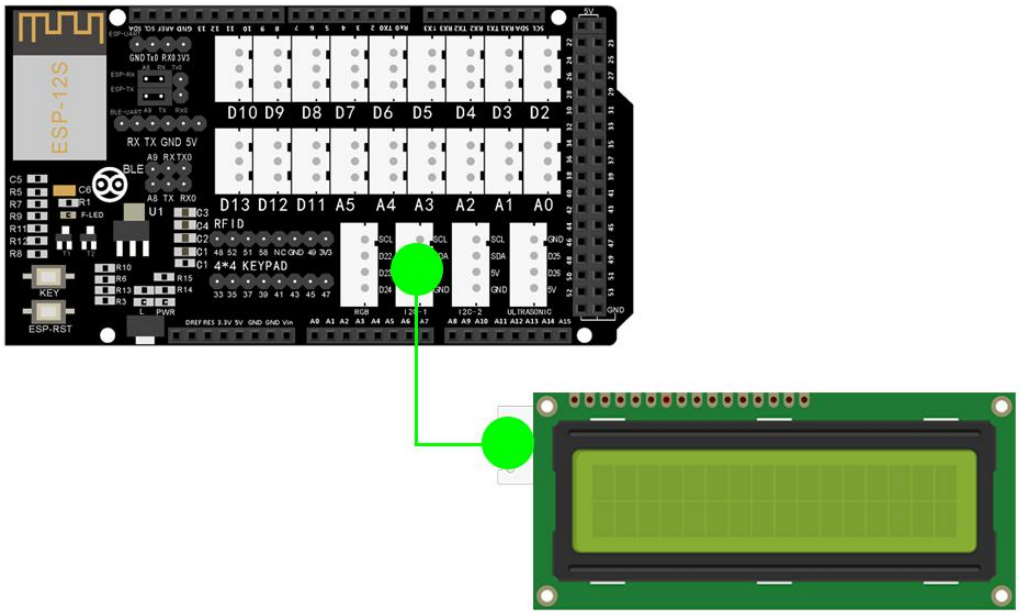
SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
I2C library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:



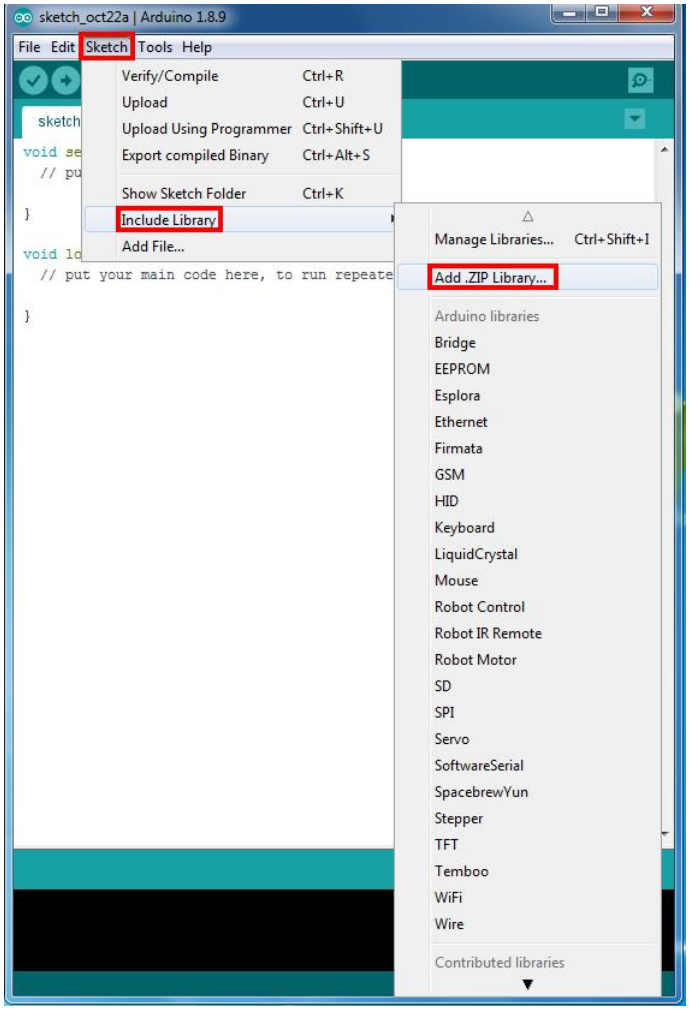
HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

Step 3 Library Installation

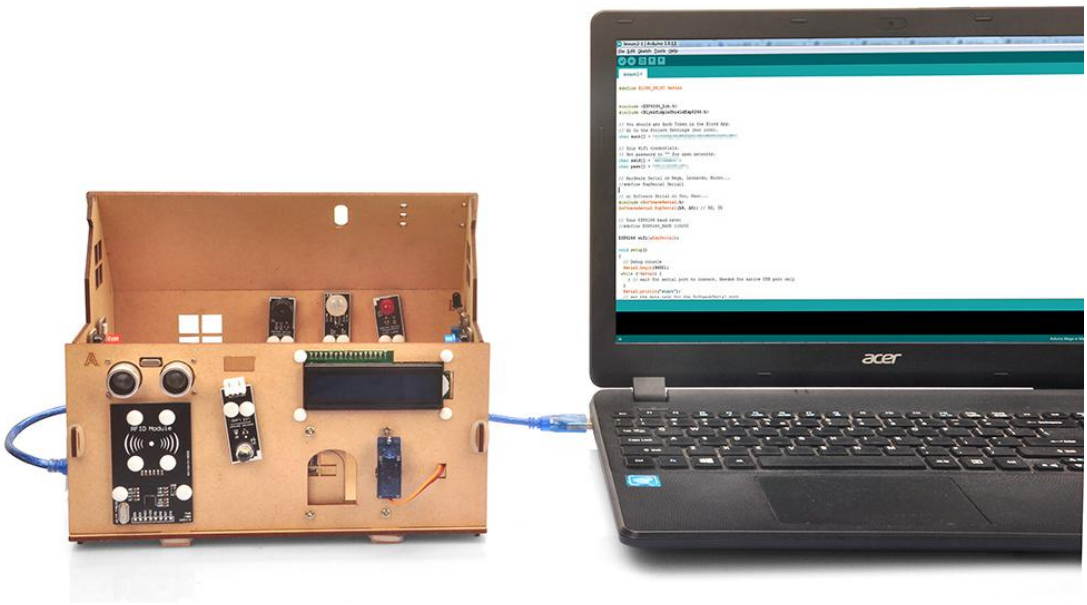
You also need to install the I2C library as above operations.



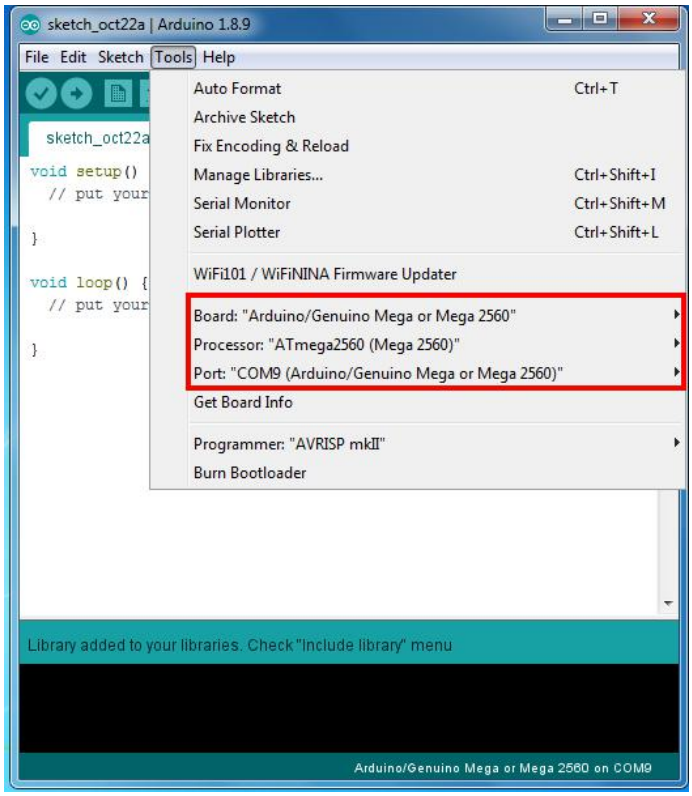
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-6.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

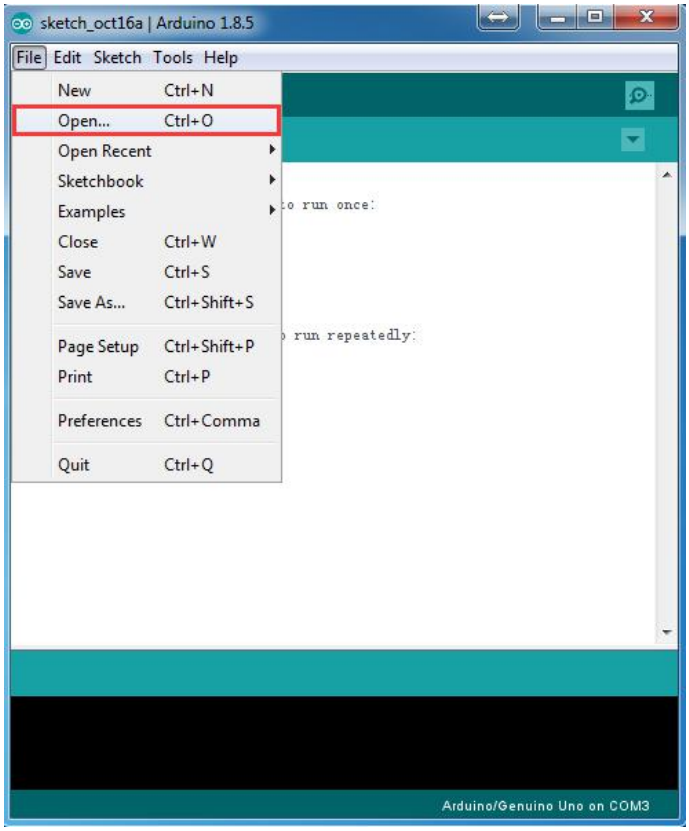
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project.



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



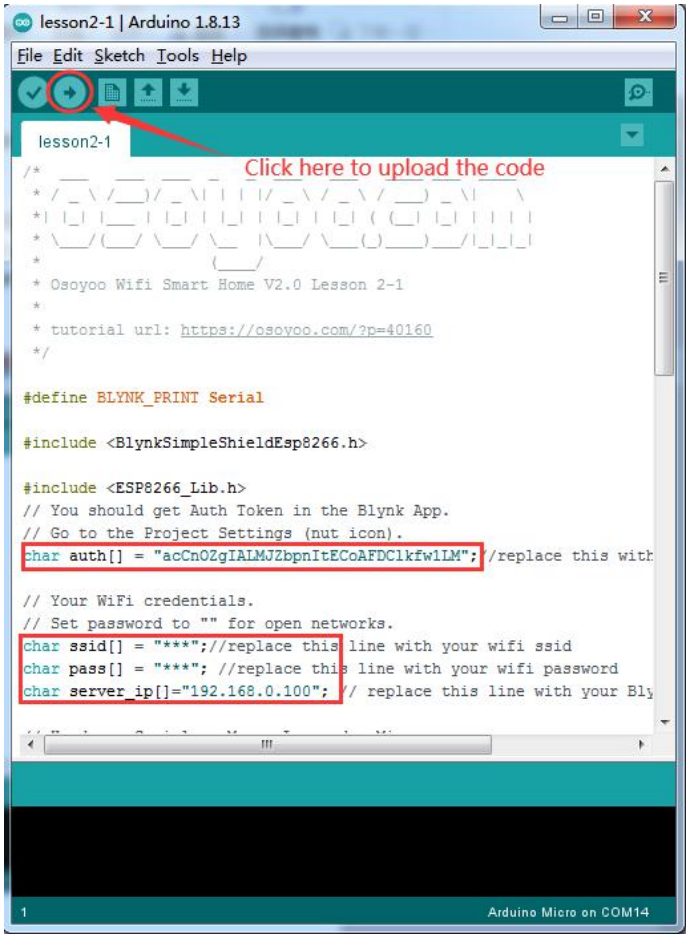
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

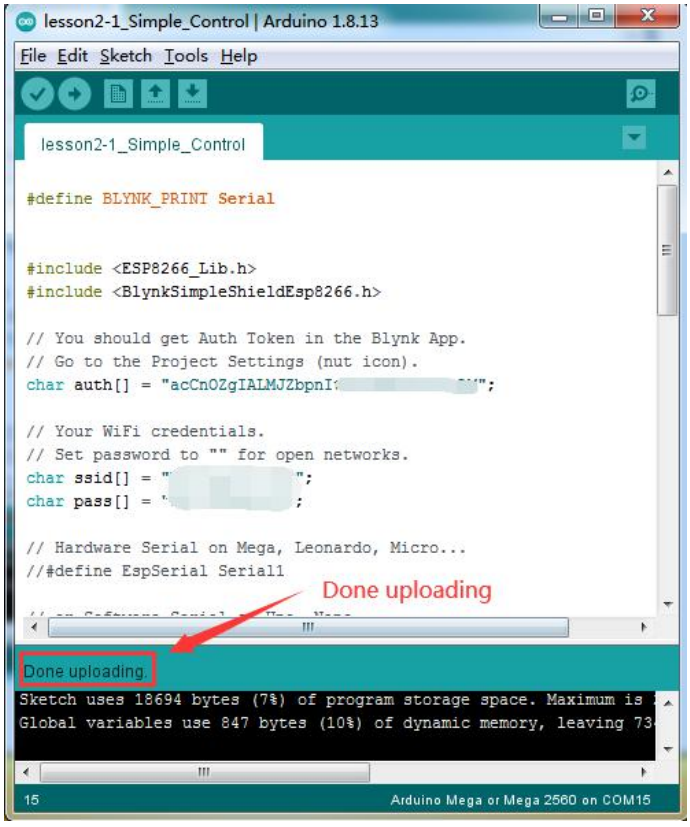
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14


lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

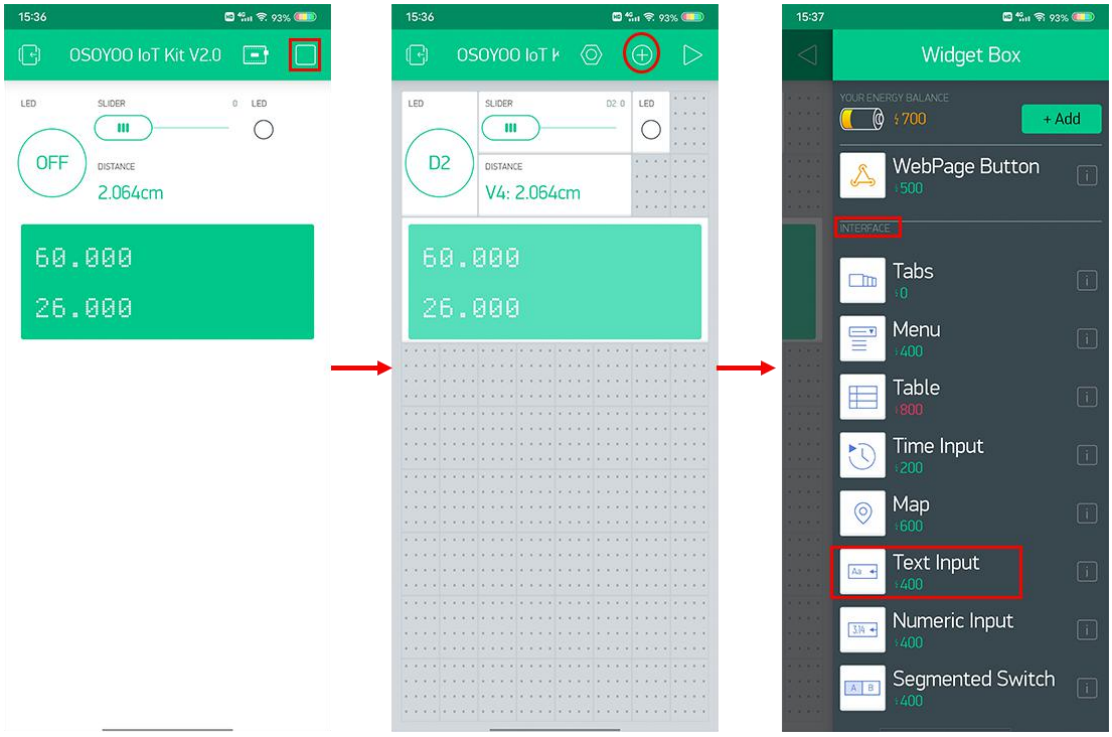
Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

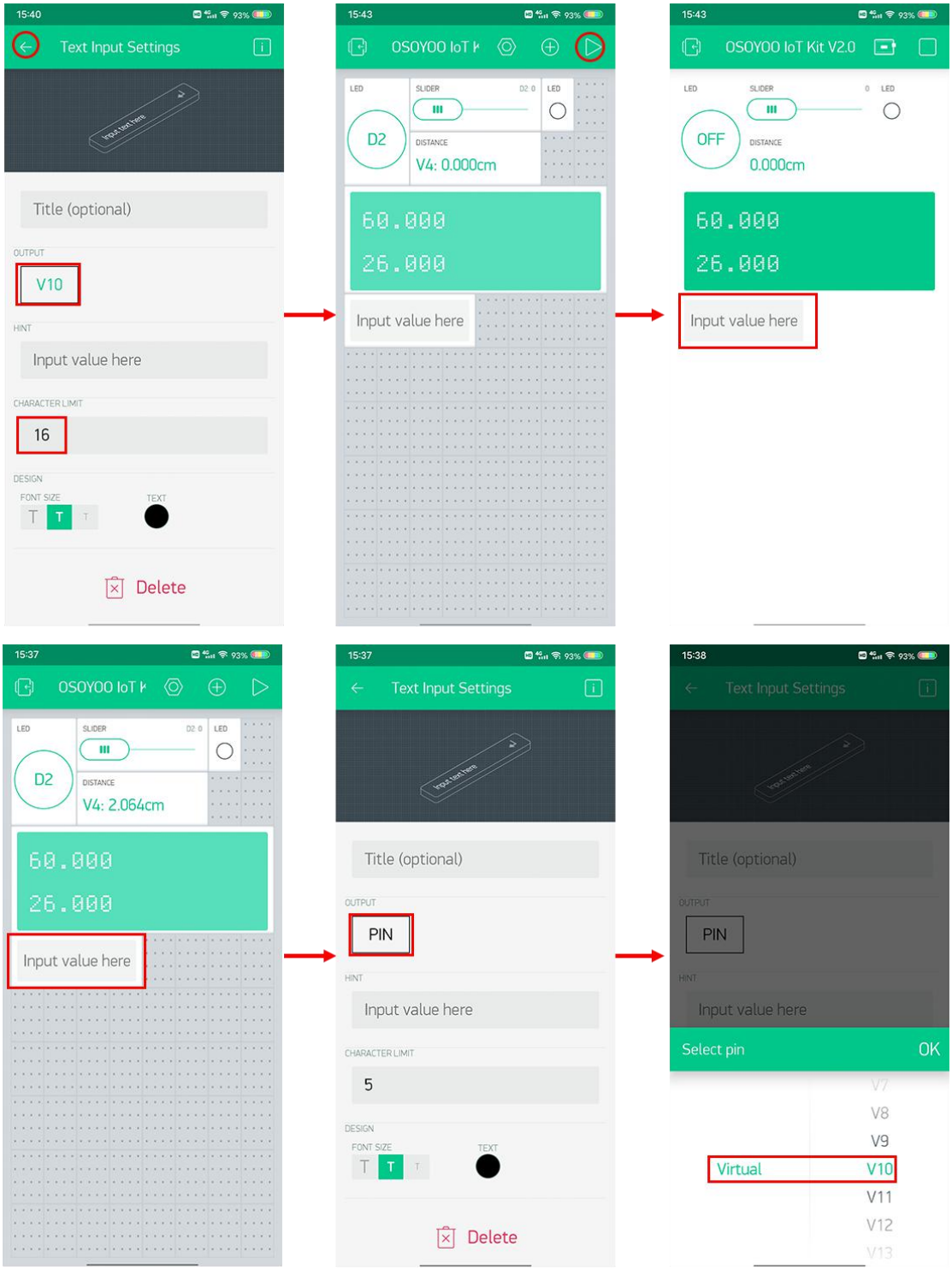
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

Overhere we need to add a Text Input wiget to get the real time distance value. Follow the next operations:

Widget Setting
Open your project page and press the "+" button to add the "Text Input" Widget.
Name Column: You can name your widget, whatever you like.
PIN: V10
CHARACTERLIMIT: 16



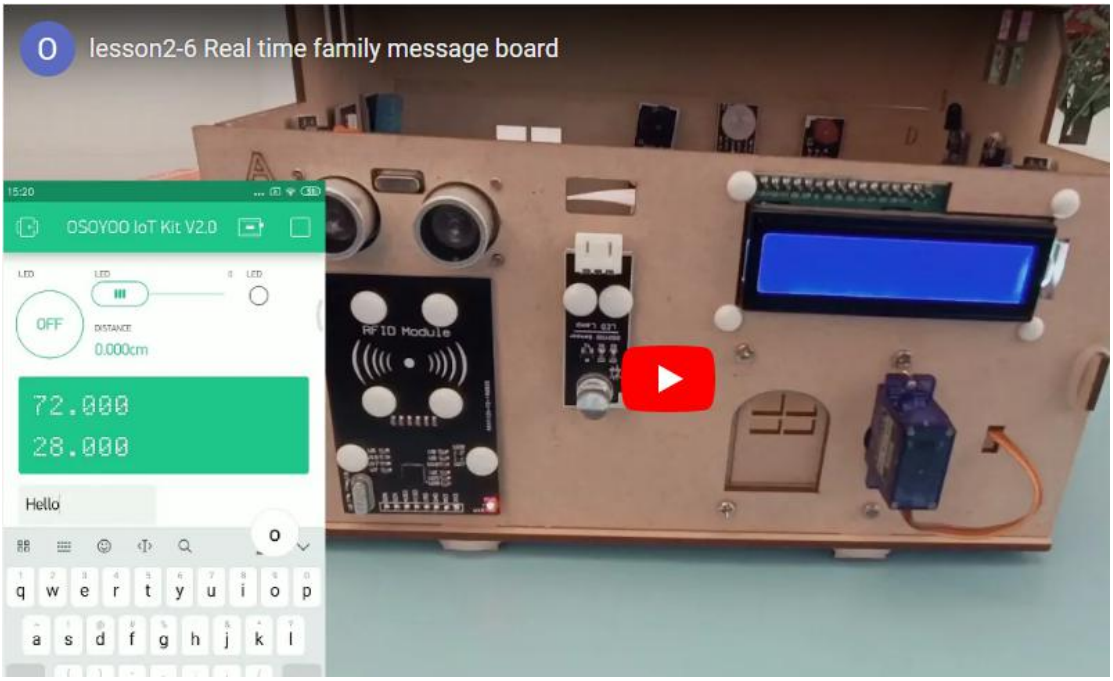


HOW TO PLAY

Click the text input field, enter what you want to say (up to 16 characters), click

the send button to send, and then you can see your message is displayed on the 1602 LCD screen.

Now we enter “Hello OSOYOO!” here, and leave a message telling the family that they need to do housework. Does it feel convenient and fun?



https://youtu.be/9kByMfgPRw0

You can also write code to scroll the words on the LCD screen, and you can also try to display your message on another line on the screen, as long as you are willing to use your brain, these are not difficult.

If you find the device is offline, please check your code, wiring and app settings, and try again.

2-7 Open the Door Remotely

OBJECTIVE

Here we will use OSOYOO IoT kit and Blynk APP to remotely control the door.

In this lesson, we actually simulate opening and closing the door by remotely controlling the rotation of the servo.

By using the Blynk button widget, we press the button on the phone, the door will be opened, and press the button again, the door will be closed. By sliding the Slider widget on the Blynk APP, we can control the opening and closing range of the door

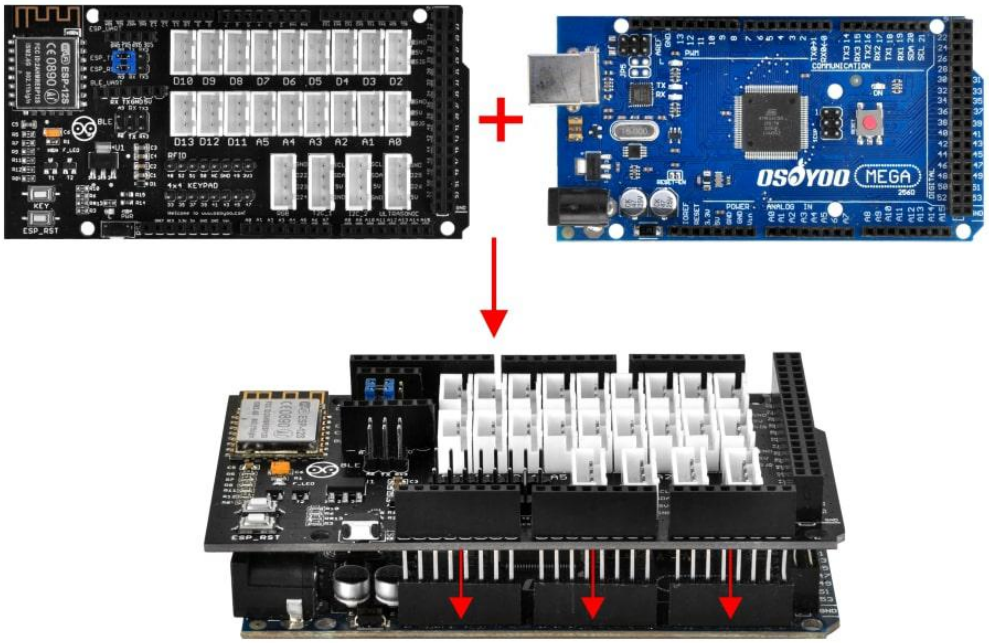
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
Servo x 1
USB Cable x 1

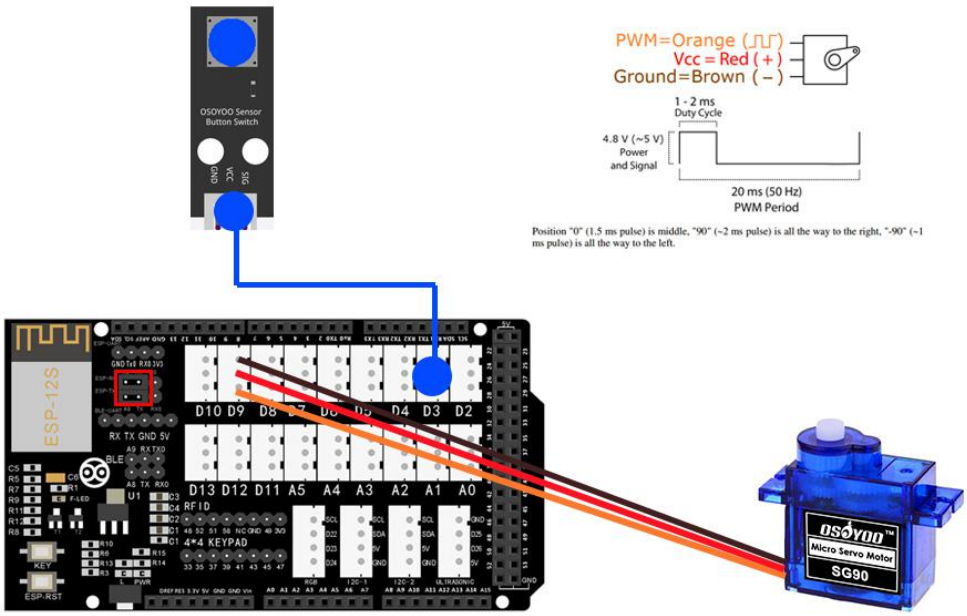
SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
Servo library

HOW TO MAKE

First, please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board:



Build the circuit as below:



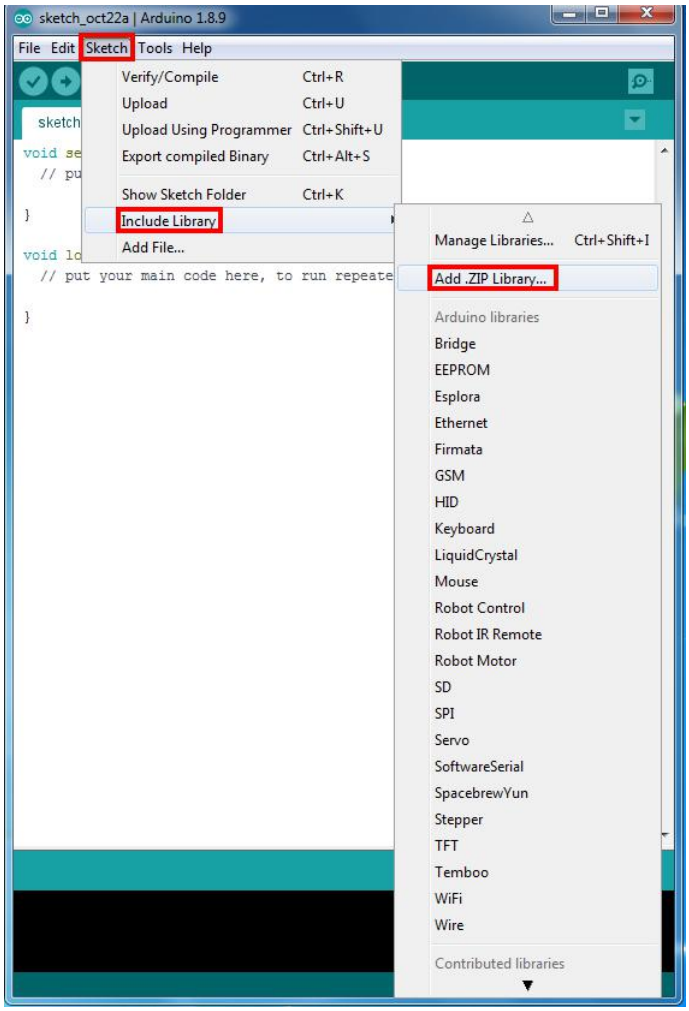
HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

Step 3 Library Installation

You also need to install the Servo library as above operations.



Step 4 Servo Test (Step 4 -Step 9) If you adjust the servo before, please skip this step.

Step 5 After installing above library, please download the code from following link, unzip it:

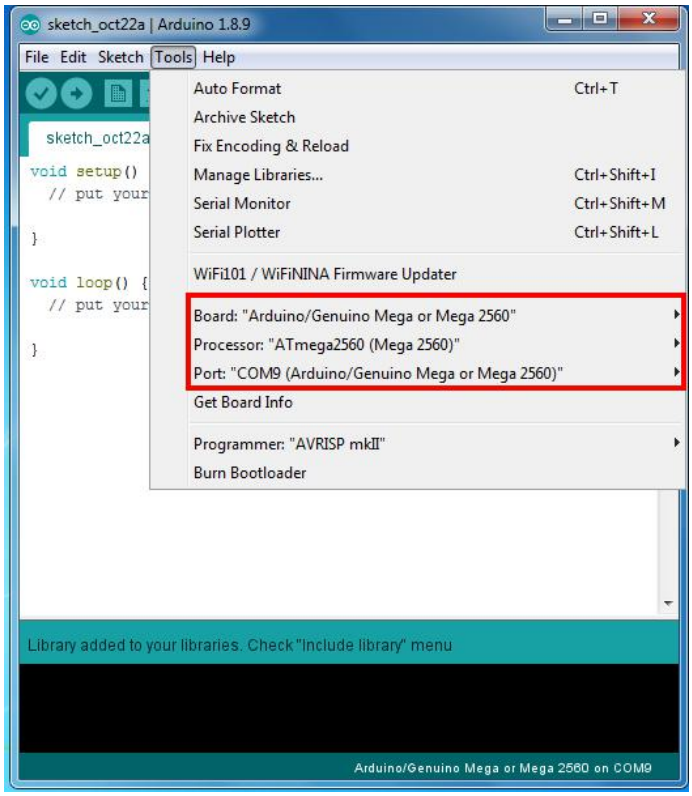
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-7.zip

Step 6 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

Notice: Shut off your battery or unplug your power adapter when upload

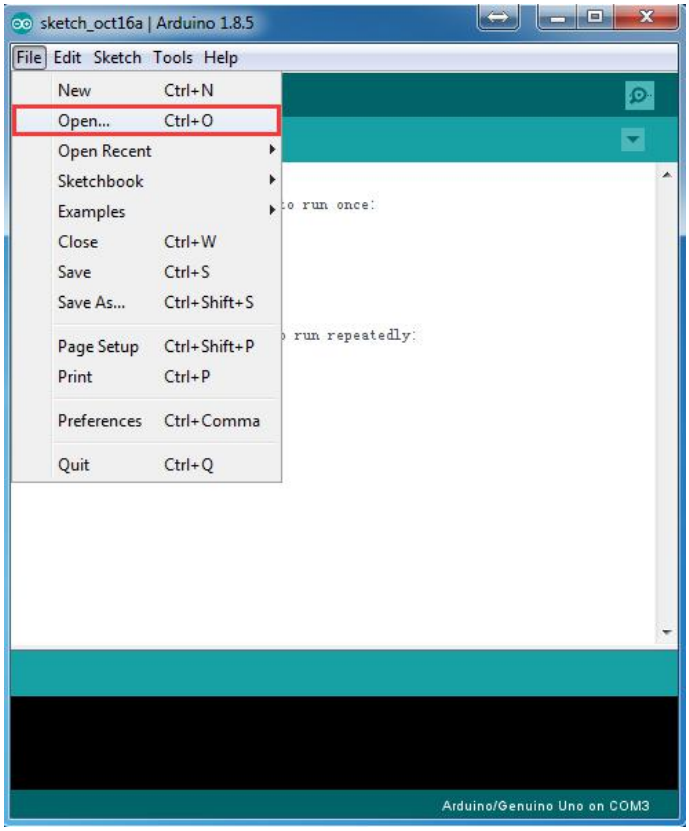
sketch code to Arduino.

Step 7 Open Arduino IDE: Choose corresponding board type and port type for you project.



Step 8 Arduino IDE: Click file – Open, then choose code in the folder, load up

the sketch onto your Arduino.



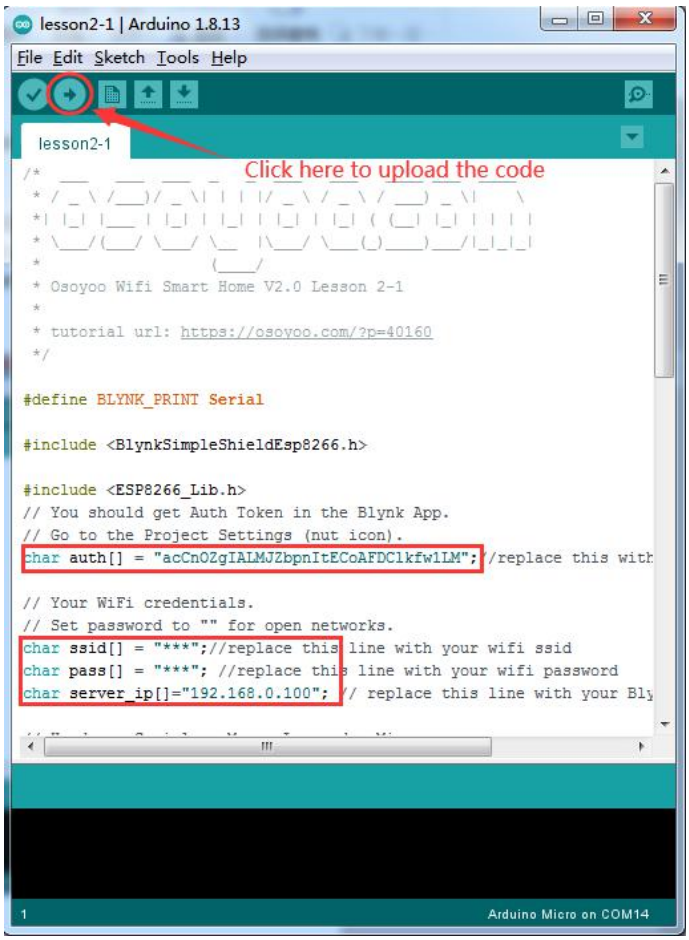
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

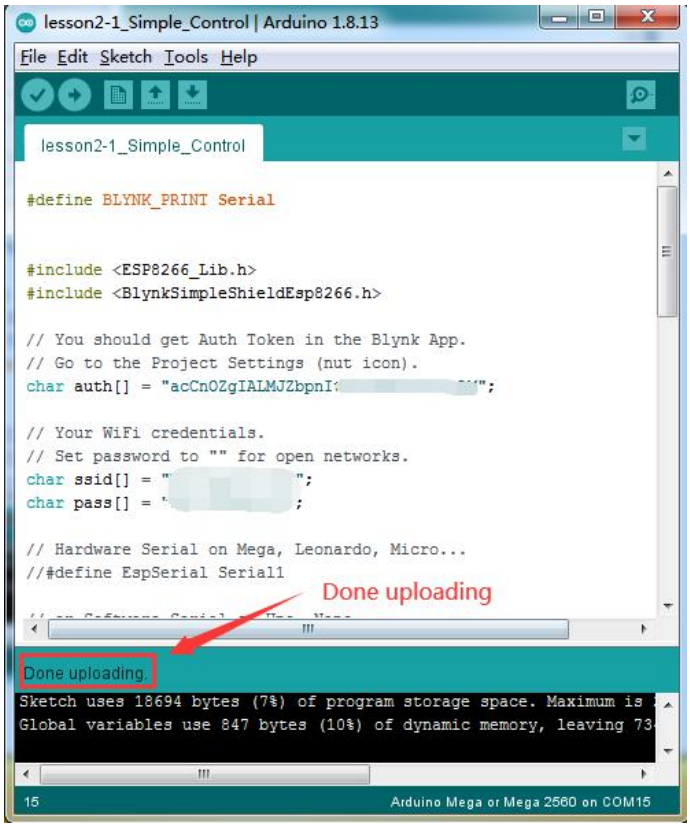
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14



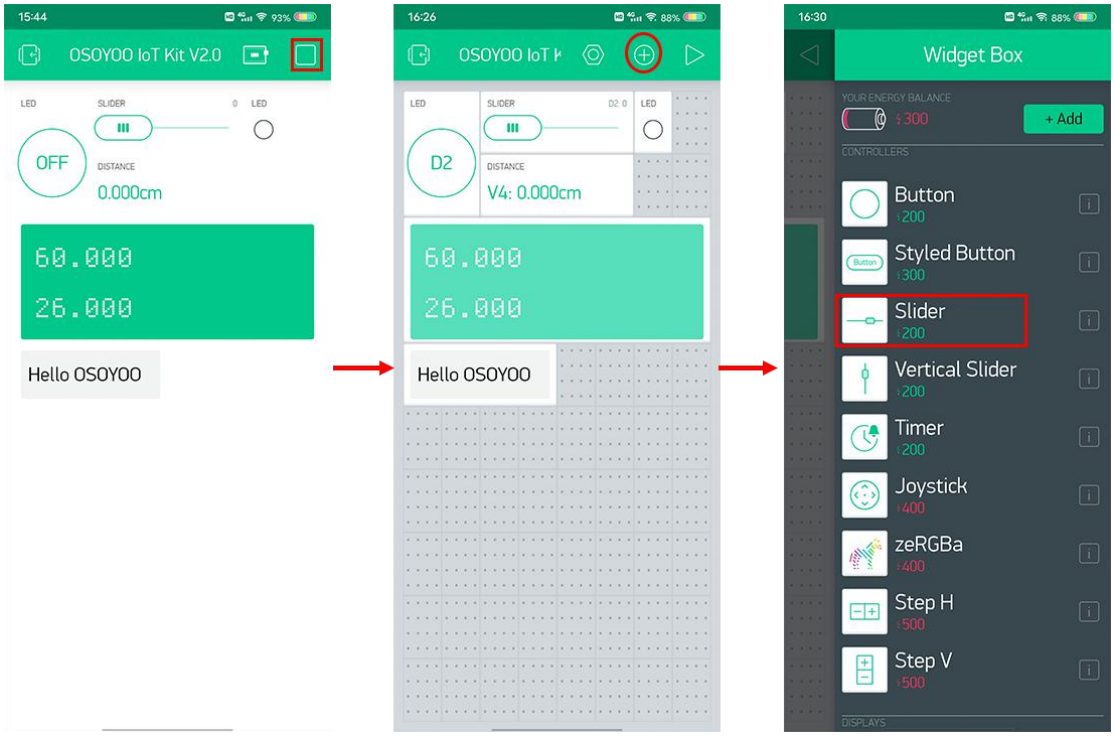
Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

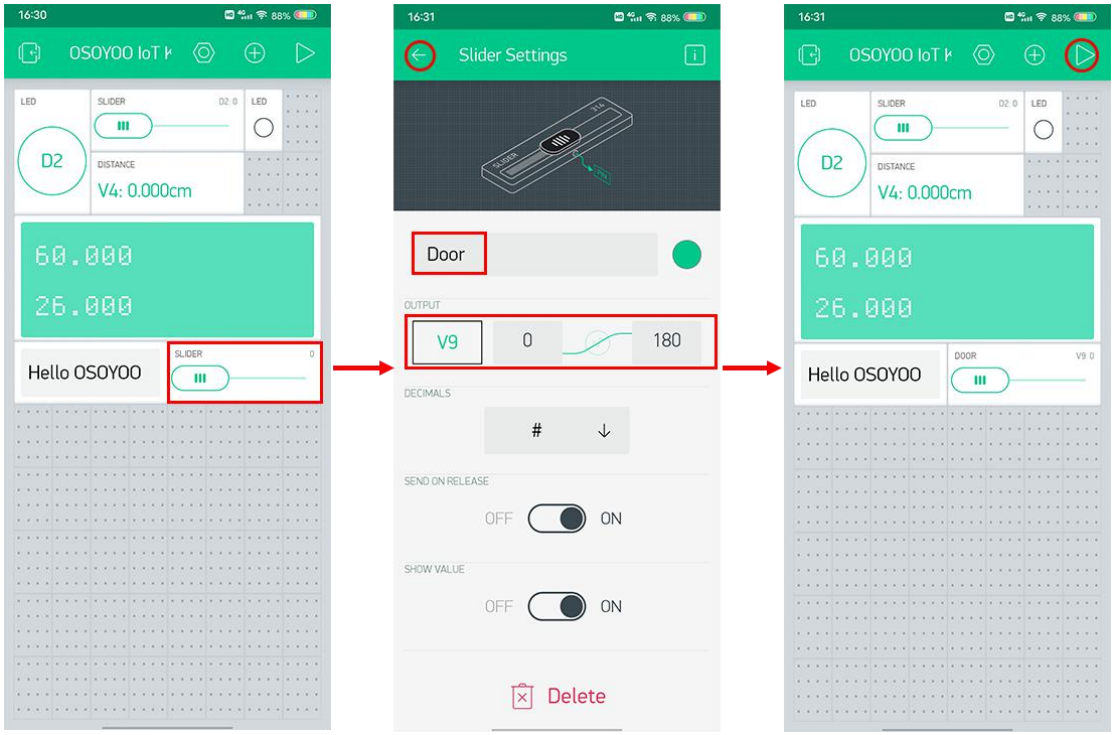
Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

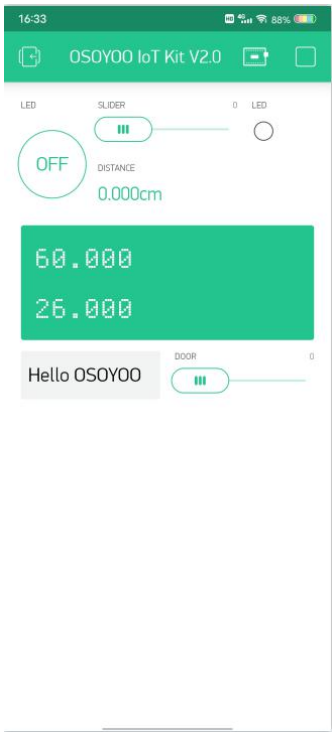
Add a Widget in Blynk (legacy) APP

Open your project page and press the "+" button to add the "Slider" Widget.

Slider Widget Settings
Name Column: Door
OUTPUT: V9 | 0~180


OSOYOO IoT Kit V2.0
OFF
0.000cm
60.000
26.000
Hello OSOYOO
D2
V4: 0.000cm
Widget Box
+ Add
Button
Styled Button
Slider
Vertical Slider
Timer
Joystick
zeRGBa
Step H
Step V


Hello OSOYOO
Slider Settings
Door
V9
0
180
Delete

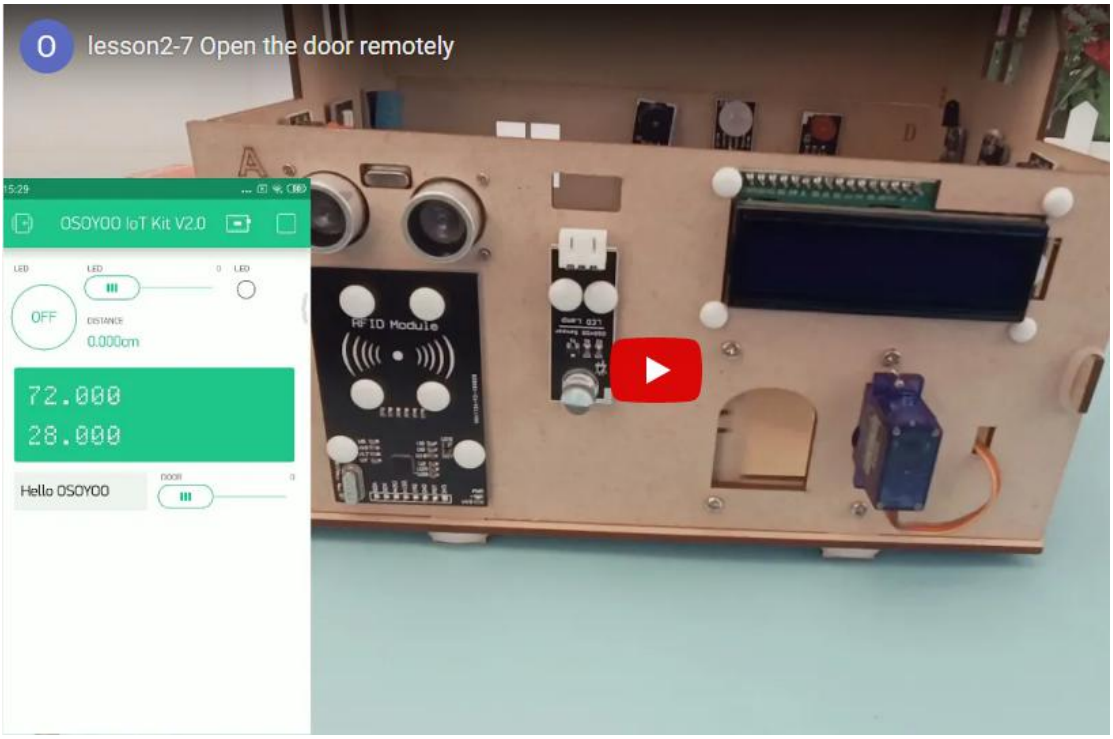


HOW TO PLAY

After you finished all above operations, open the serial monitor, then open the Blynk APP, press the PLAY button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

While in PLAY mode, you won't be able to drag or set up new widgets, press STOP and get back to EDIT mode.

Now, you'll see the servo turning with the slider!



https://youtu.be/w3TlIKLFMNc

While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

If you find the device is offline, please check your code, wiring and app settings, and try again.

Button widget to open the door remotely

Download the code from this link:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-7-button.zip

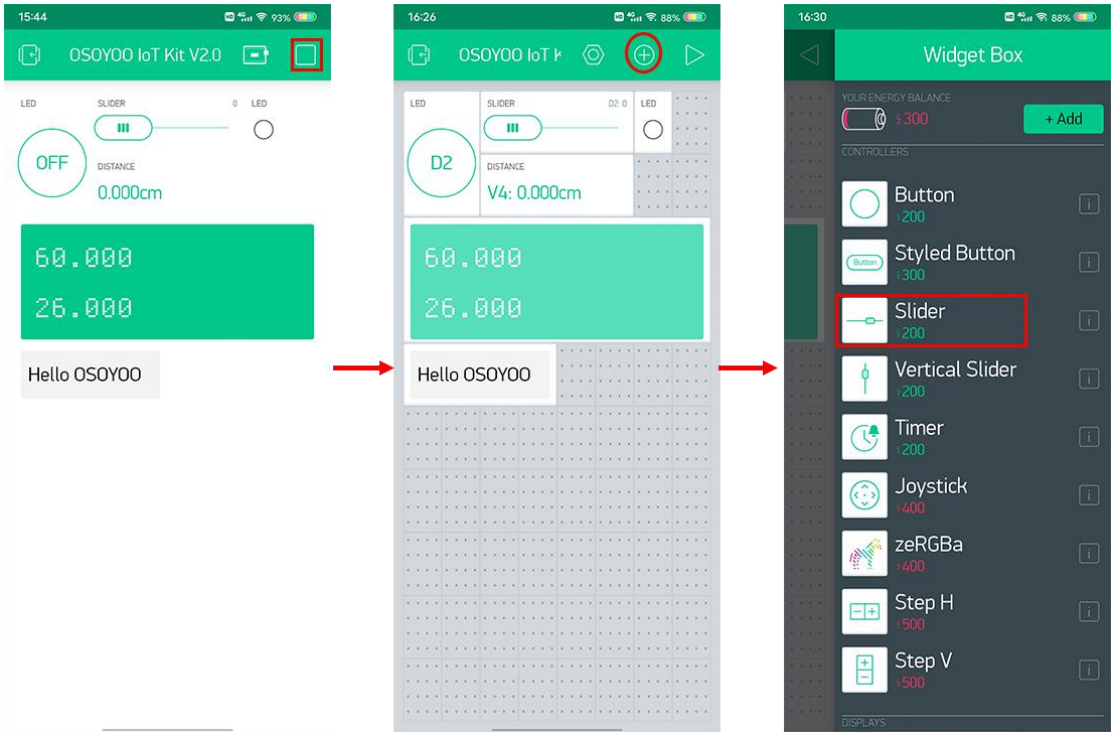
Then follow above Step 5 – Step 8 to edit and upload the code.

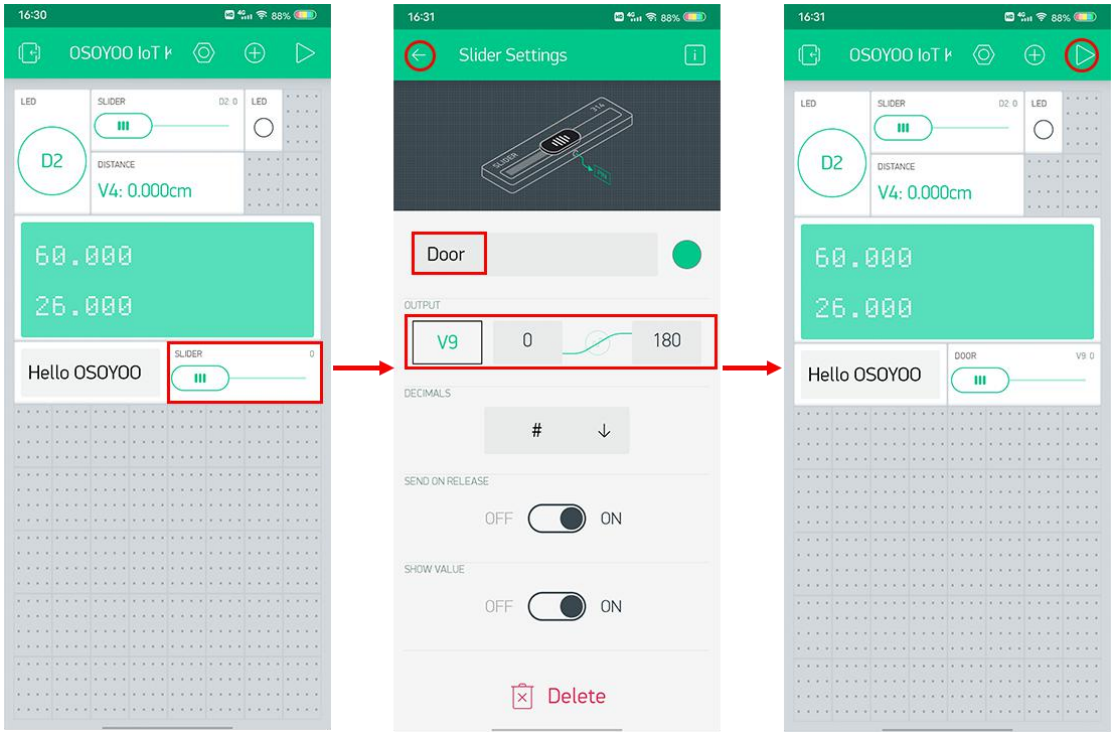
Here, we will show how to use the button widget on Blynk to open and close the door remotely. The preparation and hardware wiring are the same as the previous experiments.

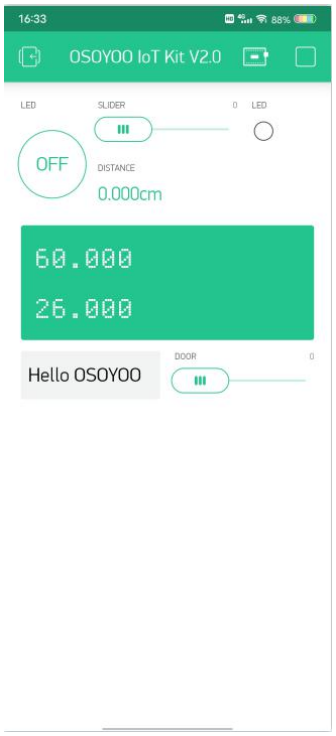
Open your project page and press the "+" button to add the "Button" Widget.
Button Widget settings:

Name Column: Door
OUTPUT: V8 | 1-0 (V stands for virtual)
MODE: We choose the Switch mode here.

You can modify other options according to your own habits or keep them as default.





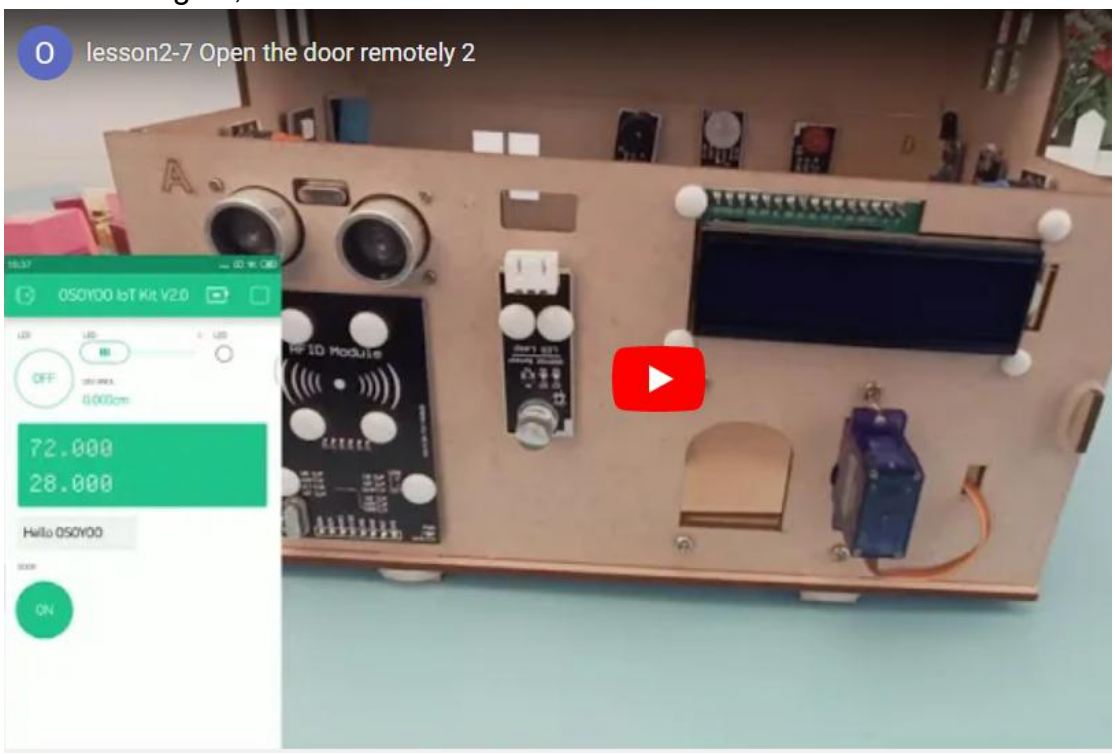


After you finished all above operations, press the "▷ " button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

While in PLAY mode, you won't be able to drag or set up new widgets, press "□" and get back to EDIT mode.

Running Result

Now, press the "Door" button on the APP, the door will be opened, and press the button again, the door will be closed.



https://youtu.be/yk2oax01pjQ

2-8 Smart atmosphere light

OBJECTIVE

In this lesson, we will show how to use the OSOYOO IoT kit and Blynk APP to make a remote-controllable ambient light. We can change colors of the light at home through the mobile APP to make more colorful moments.

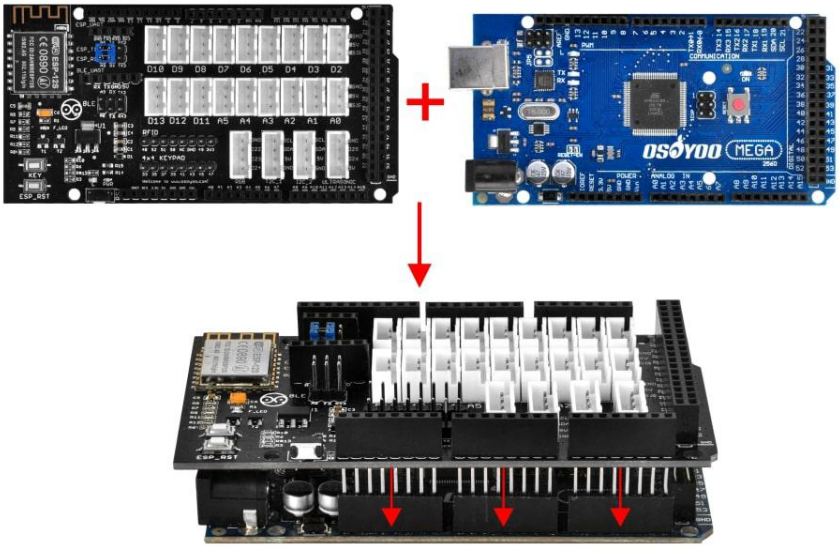
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x 1
RGB Module x 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

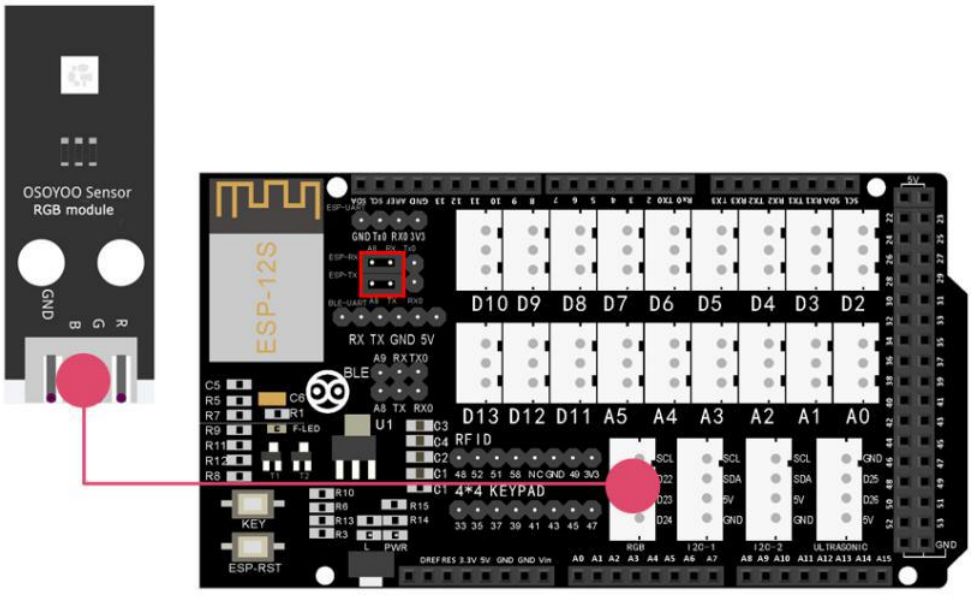
HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:

RGB module --- RGB port



HOW TO CODE

Step 1 Prerequisite

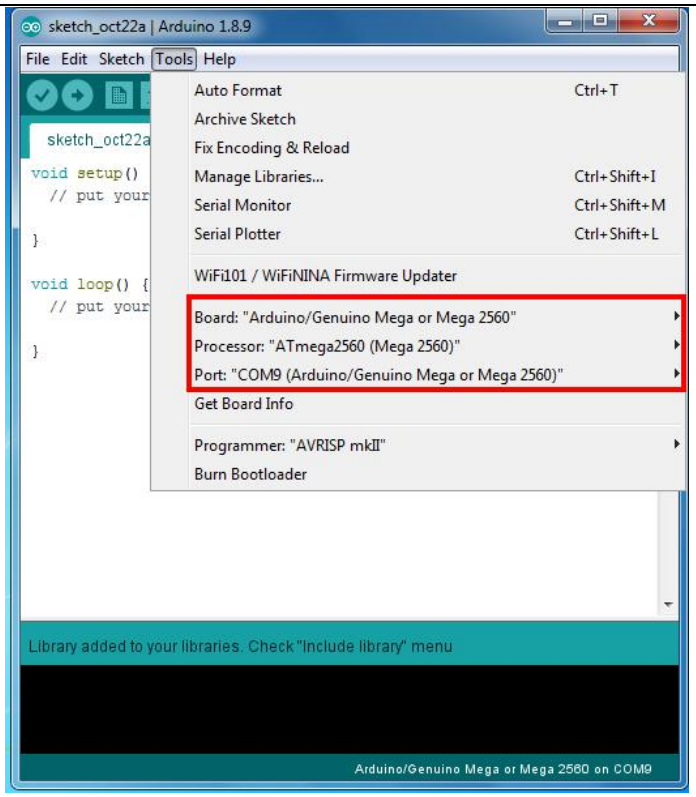
Step 2 Install latest Arduino IDE

Step 3 Library Installation

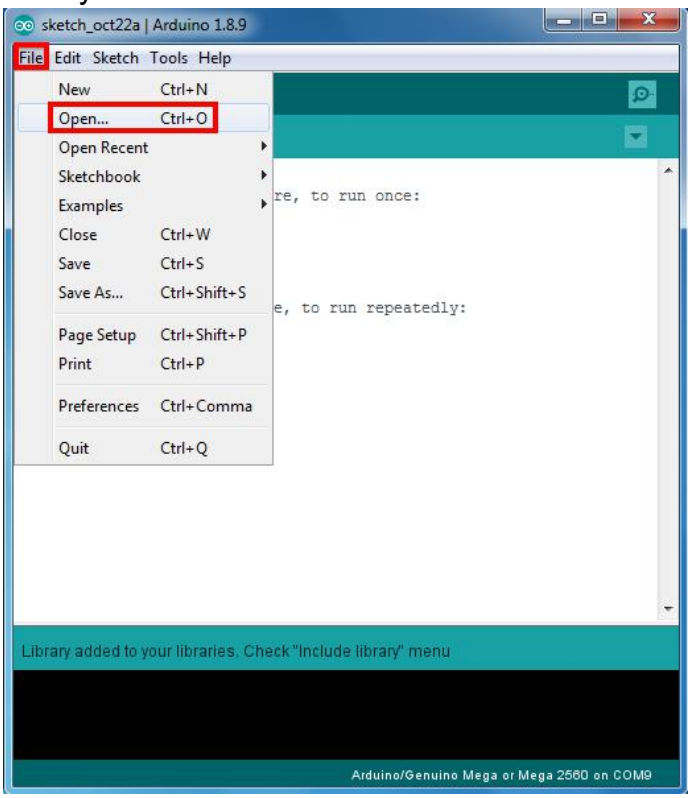
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-8.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



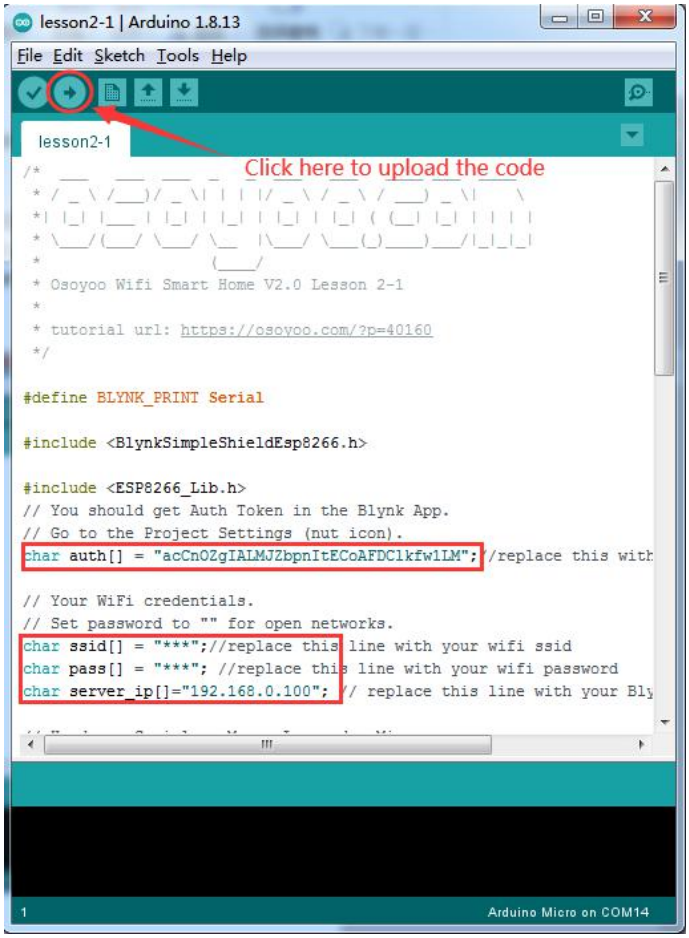
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

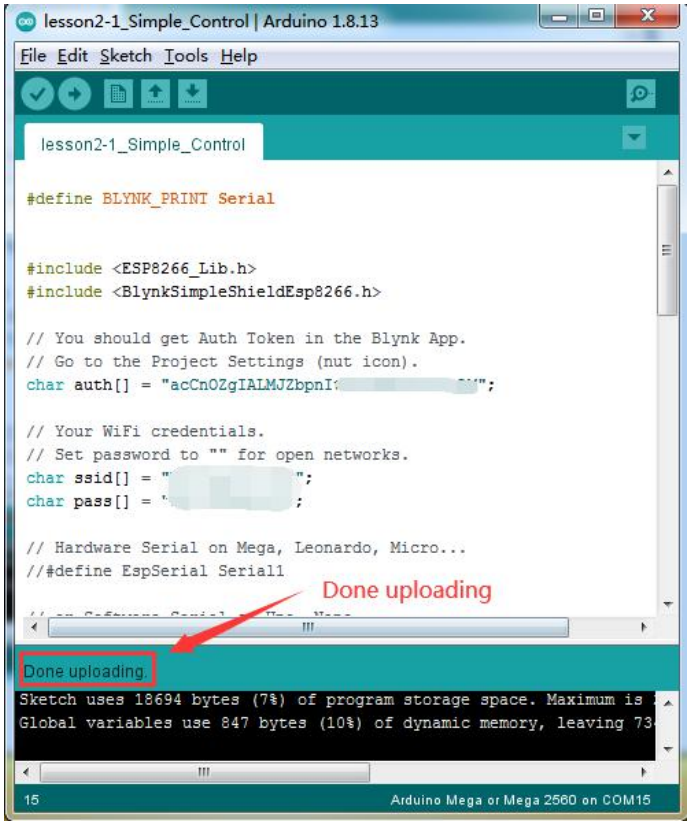
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14

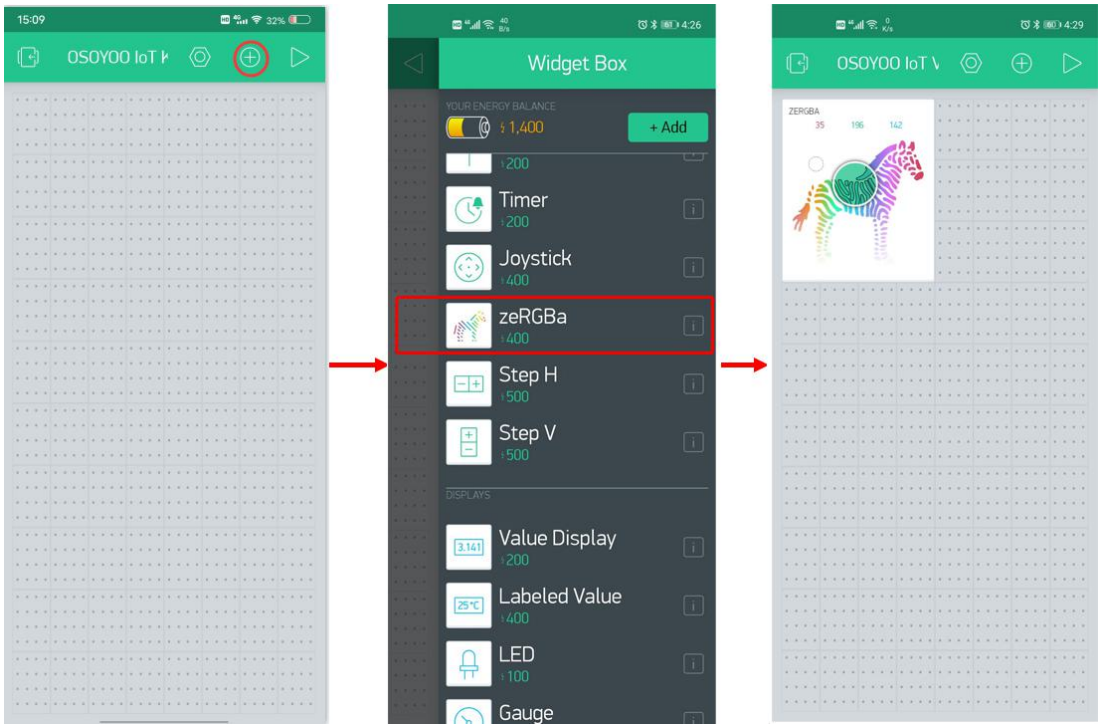

lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

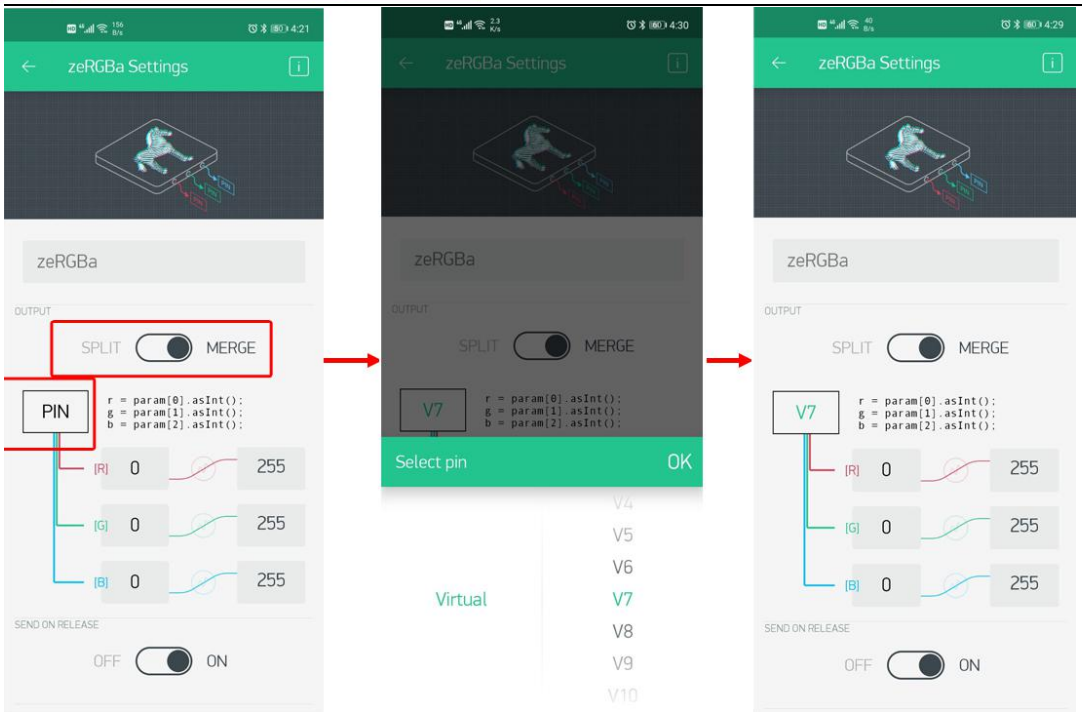
Add a Widget in Blynk (legacy) APP

Open the Blynk APP , Follow the next operations:
Open your project page and press the "+" button to add the "zeRGBa" Widget.



zeRGBa Widget settings:

Name Column: You can name your widget, whatever you like.
INPUT: V7
Select the "MERGE" output mode



Then back to the project page. and press the "▷ " button to start your project.

HOW TO PLAY

Slide the floating button on the zeRGBA widget, you can see that the lighting at home changes with your settings, so that you can control the color and brightness of the lighting at home at will.

Look the video on Youtube :

https://www.youtube.com/embed/tVexZRoUJXI

2-9 Christmas light show

OBJECTIVE

Before Christmas, we always like to decorate the house with colorful lights. Here, we will learn how to use this OSOYOO Kit and Blynk APP to control a fancy LED strip lights. Then, when Xmas is coming, you will have an eye-catching lovely home in your neighbourhood.

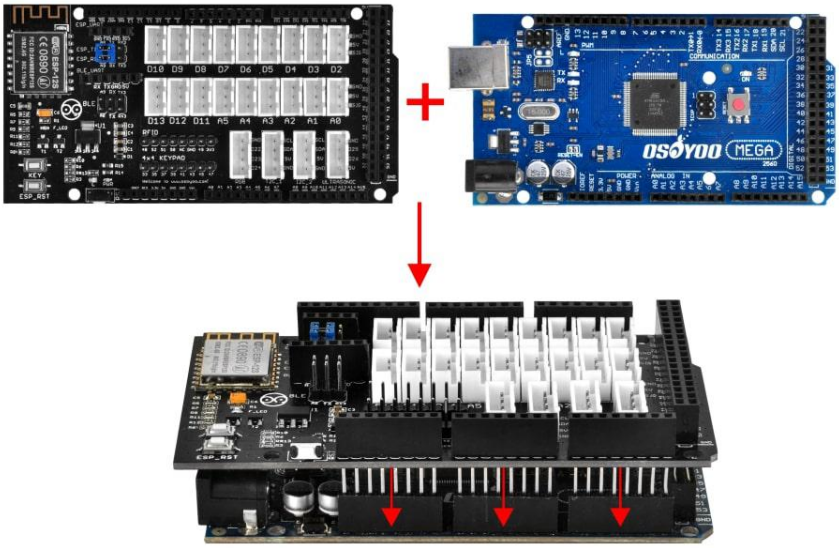
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
LED strip Lights x 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

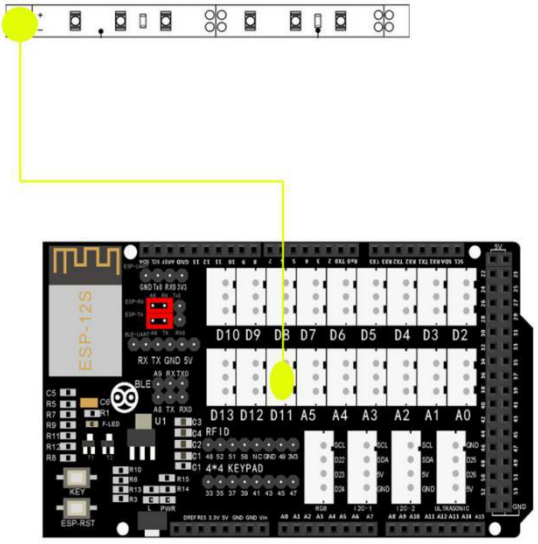
HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:

LED strip lights----D11



HOW TO CODE

Step 1 Prerequisite

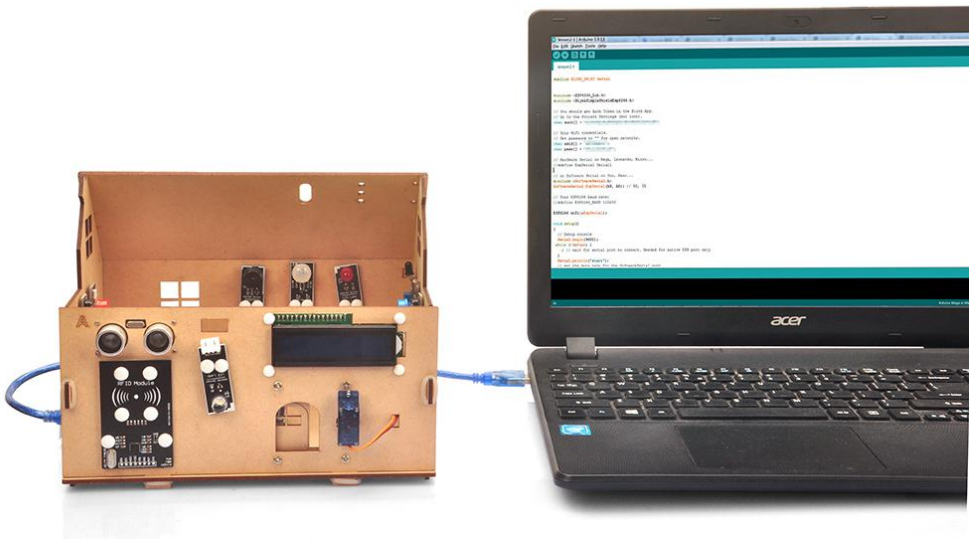
Step 2 Install latest Arduino IDE

Step 3 Library Installation

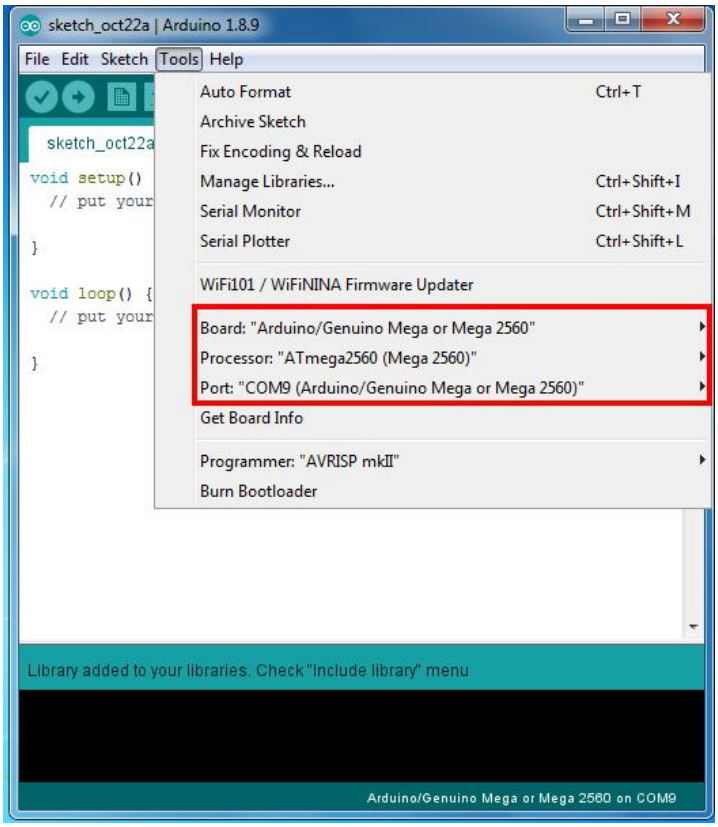
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson2-9.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

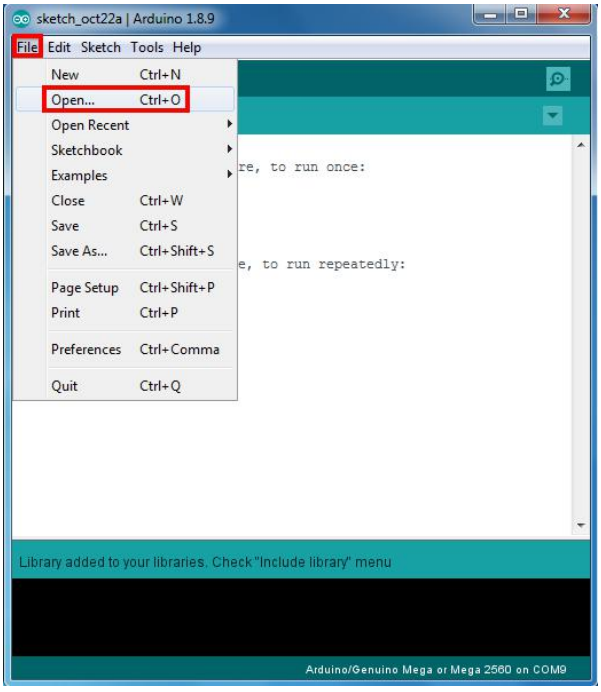
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

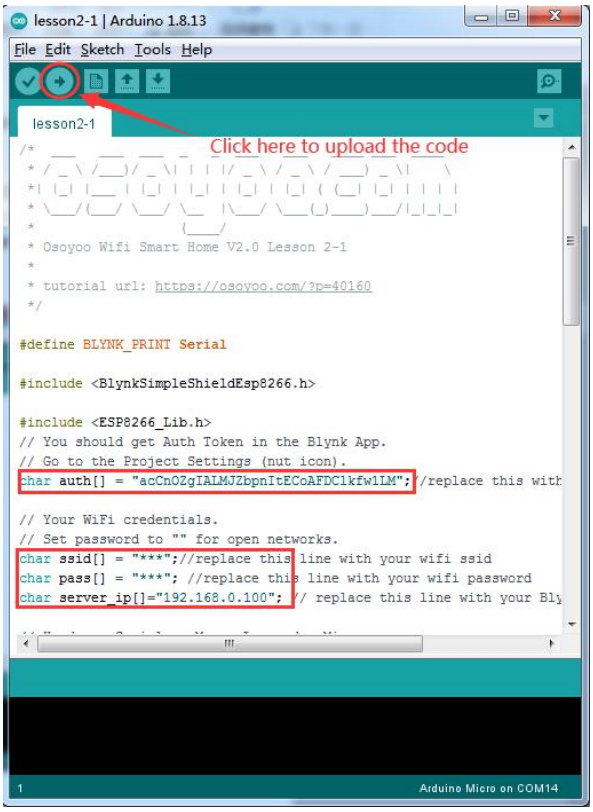
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

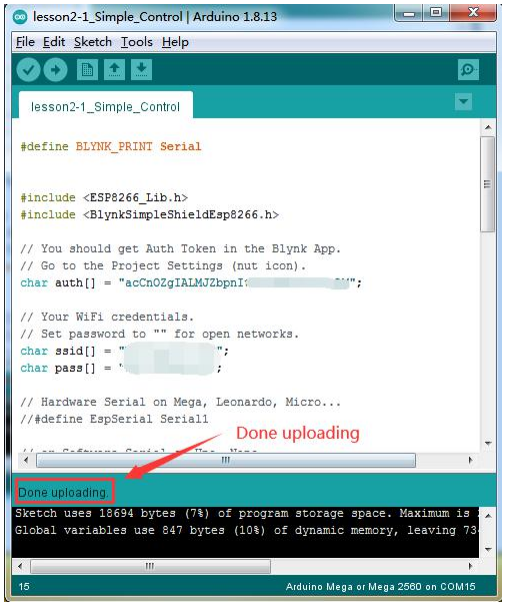
Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading

Congratulations! You are all ready! Your hardware is now connected to the Blynk cloud!



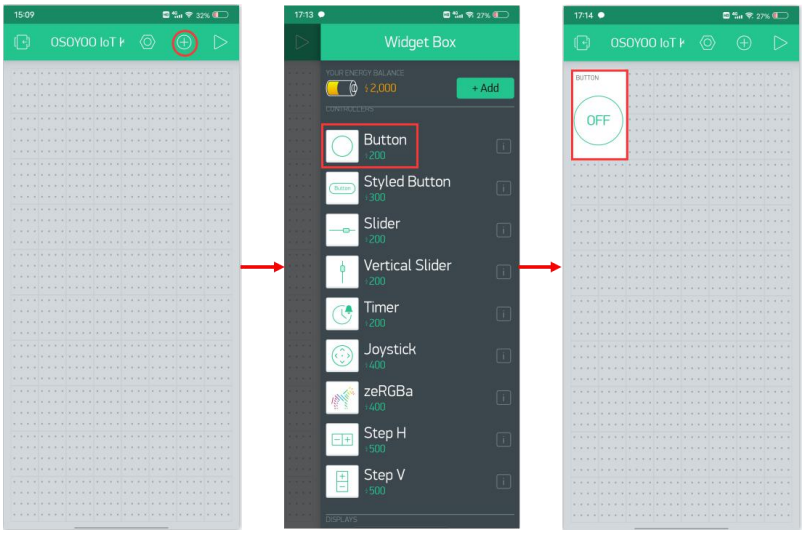


Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

Open the Blynk APP , Follow the next operations:
Open your project page and press the "+" button to add the ″Button" Widget.



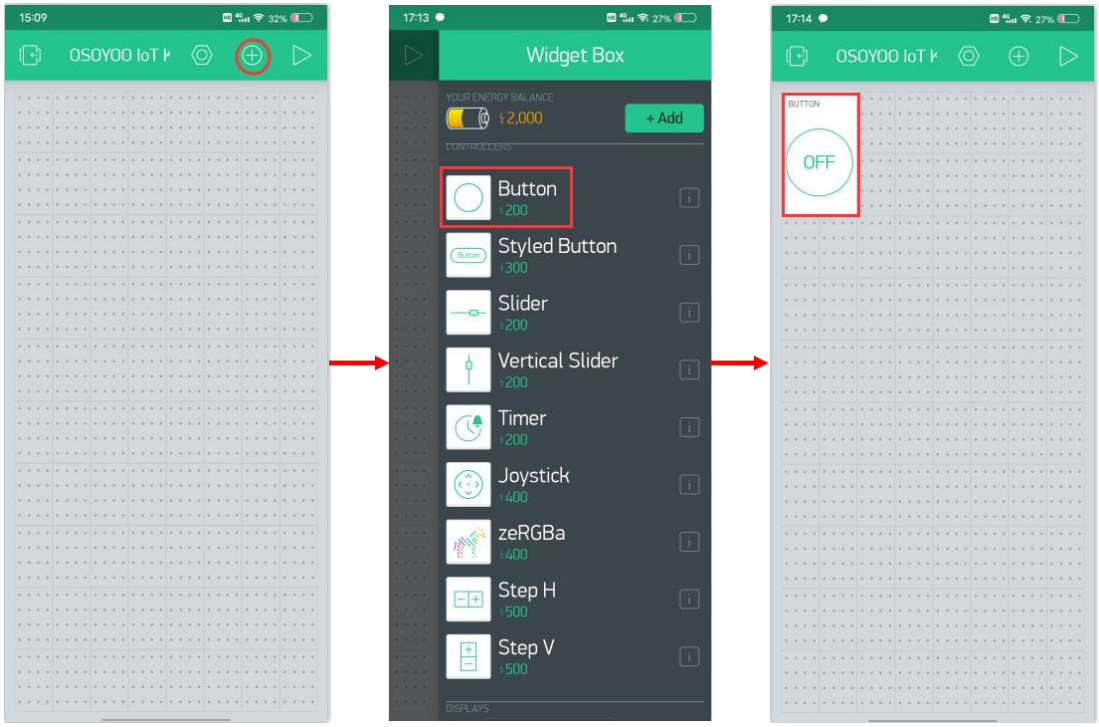
Button Widget settings:

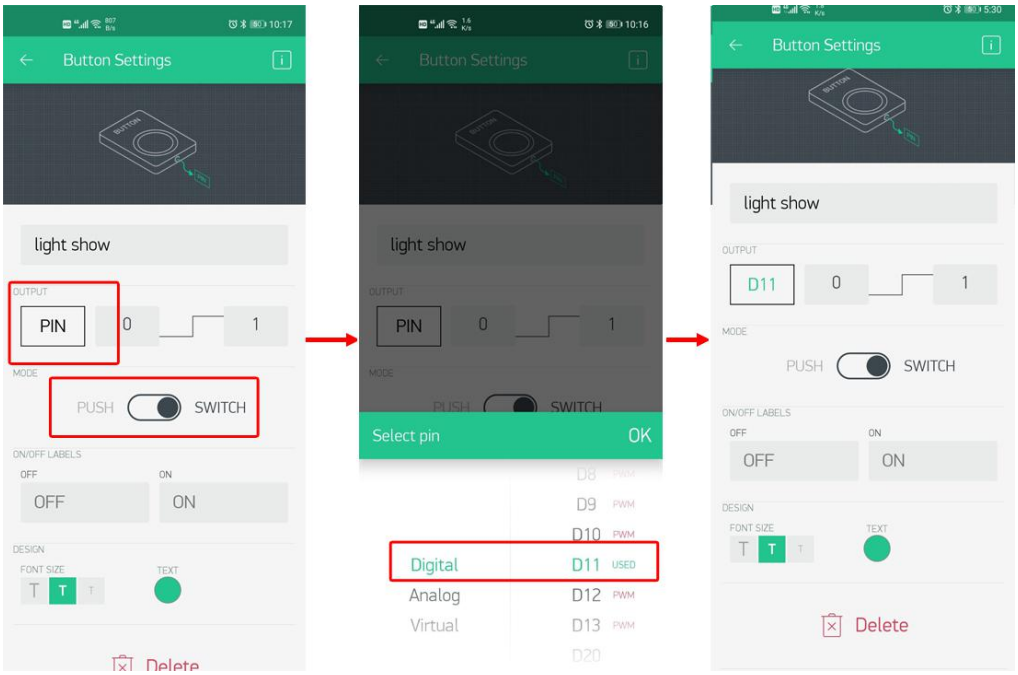
Name Column: You can name your widget, whatever you like.

OUTPUT: D11| 0-1

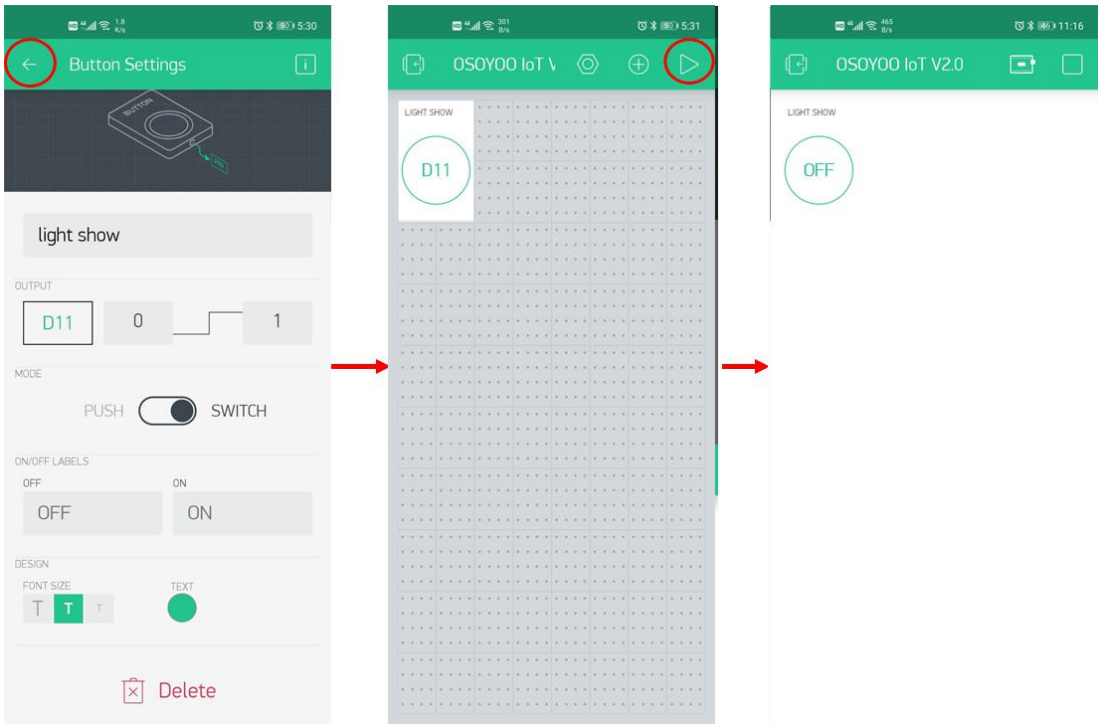
Select the "Switch"

You can modify other options according to your own habits or keep them as default.





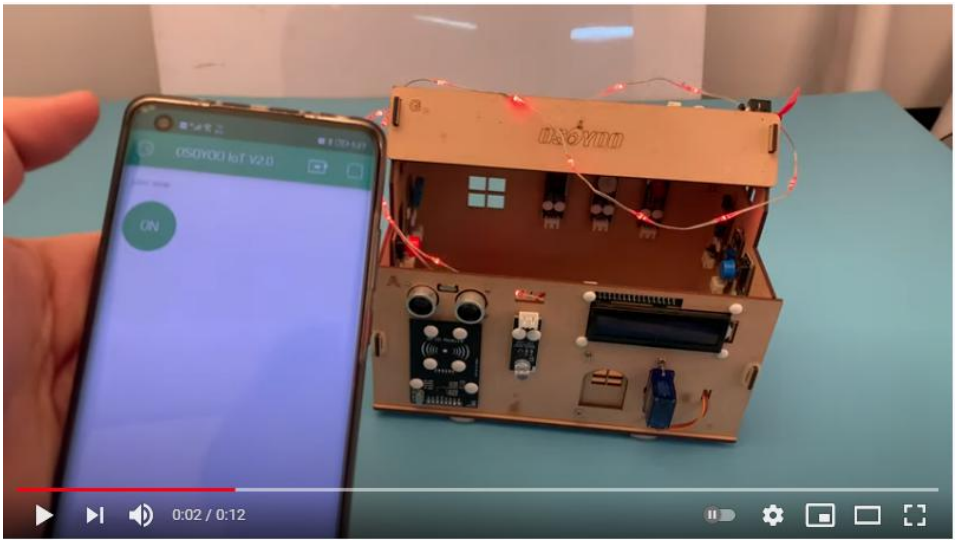
Then back to the project page. and press the "▷ " button to start your project.



HOW TO PLAY

Now, press the "LIGHTS" button on the APP, the light strip will be lit and flashing colorful lights, and then press the button again, the light strip will go out.

Look the video on Youtube :



https://www.youtube.com/embed/j8y650Xw03c

3-1 Breaking into the home alarm

OBJECTIVE

In this tutorial, we will learn how to use the Blynk push email API to send emails from this OSOYOO IoT device. For this demonstration, we will build a simple human infrared detection project, and then set the infrared detection alarm point. Whenever an alarm is triggered, an email will be sent.

This experiment can be applied to various security projects, such as breaking into the home alarm.

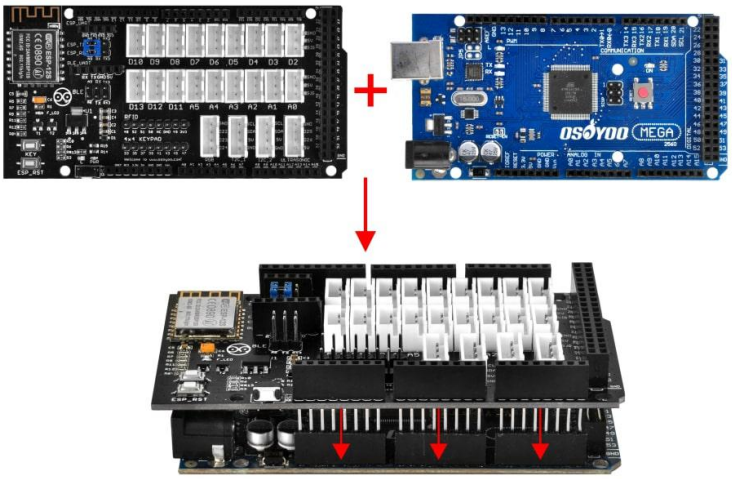
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
PIR Motion Detect Module x 1
Positive Buzzer x 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

HOW TO MAKE

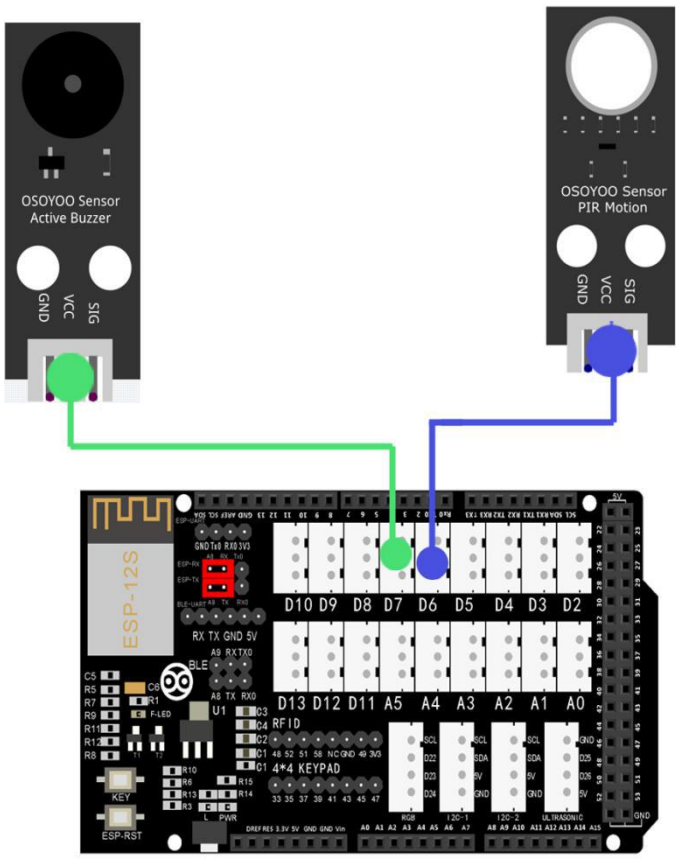
First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:

PIR motion sensor --- D6

buzzer -----D7.



HOW TO CODE

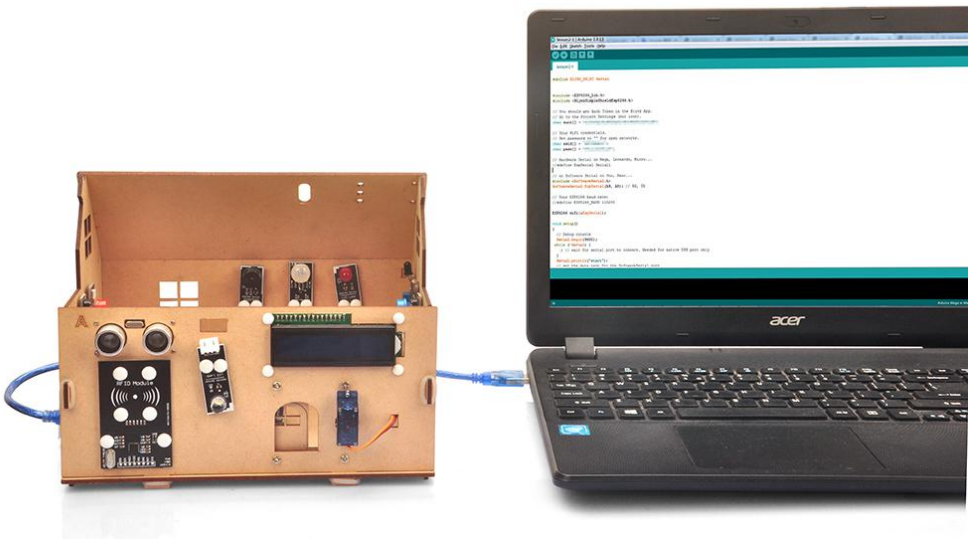
Step 1 Prerequisite

Step 2 Install latest Arduino IDE

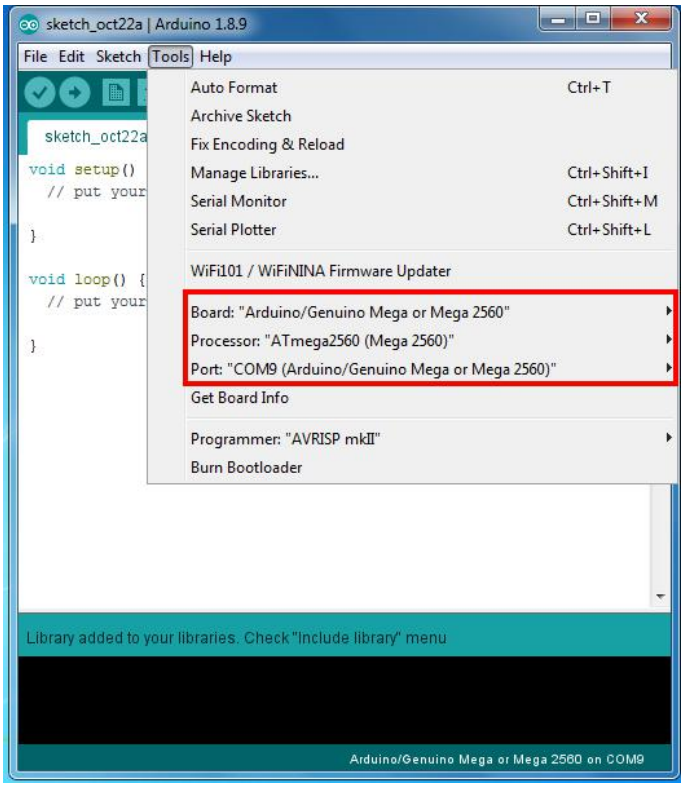
Step 3 Library Installation

Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-1.zip

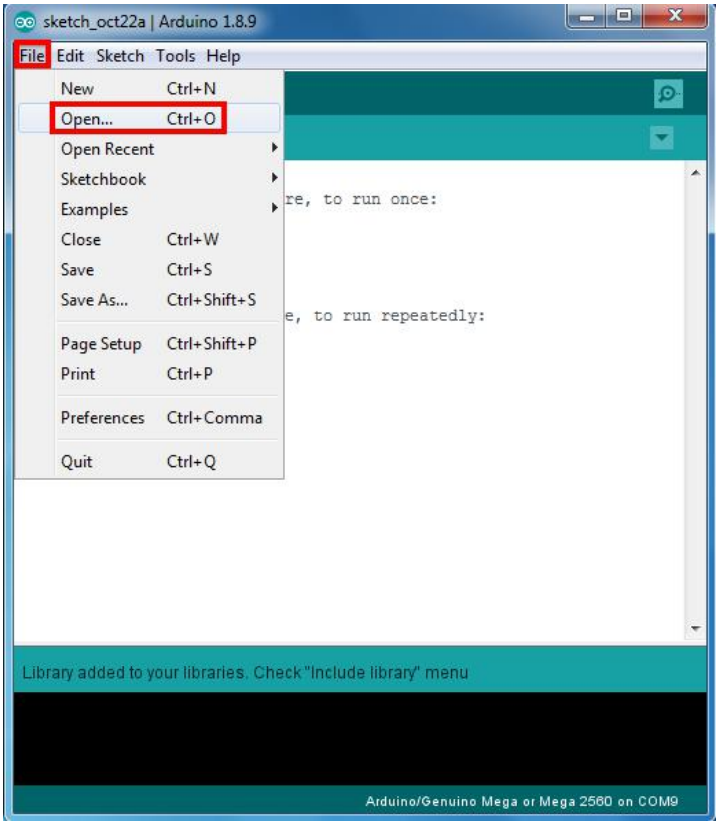
Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.

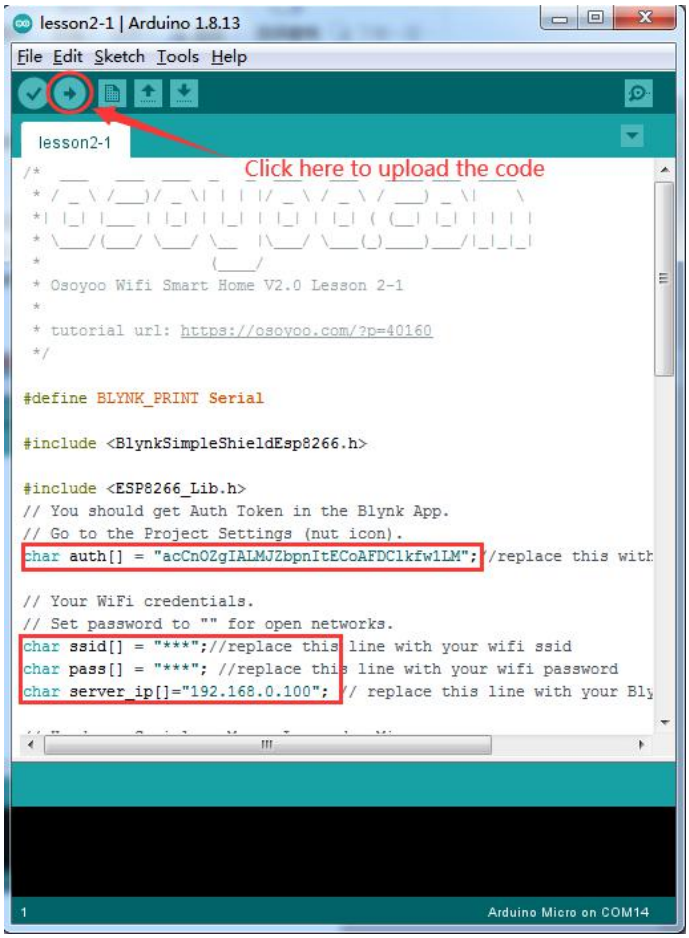


Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

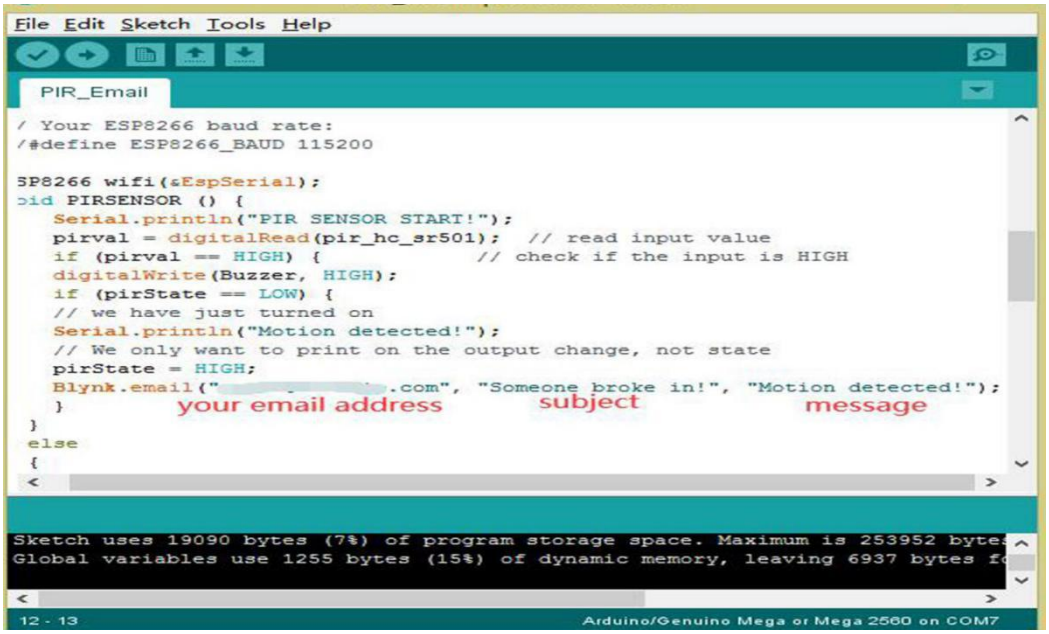
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

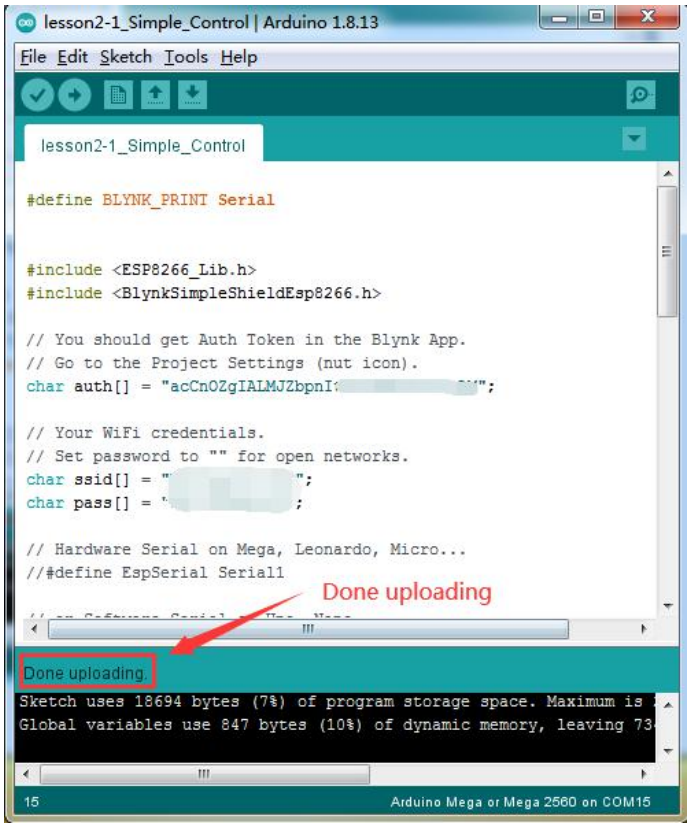


Then edit the email:



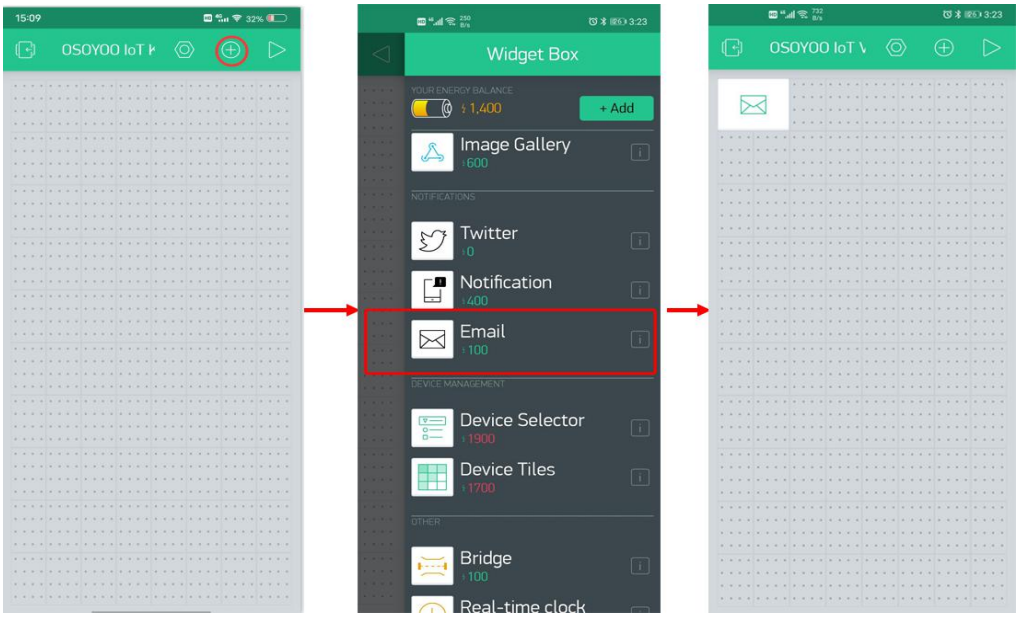
Finally Upload the sketch to the board. Wait until you see something like this: *Done uploading*

Congratulations! You are all ready! Your hardware is now connected to the Blynk cloud!

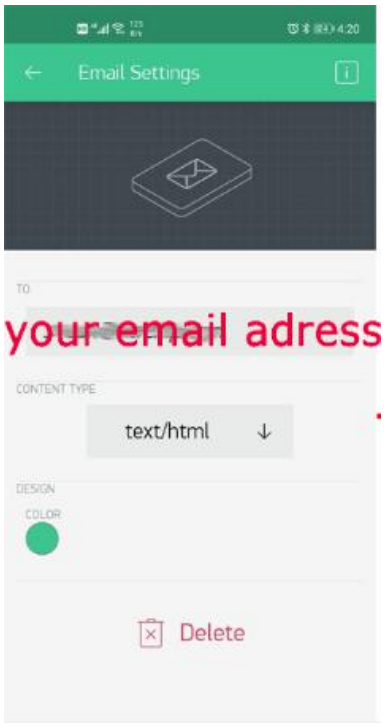


Add a Widget in Blynk (legacy) APP

Open the Blynk APP , Follow the next operations:
Open your project page and press the “+” button to add the "Email" Widget.



Email Widget settings:



Some information about Email (you can skip this part)

Email widget allows you to send email from your hardware to any address.

Example code:

```
Blynk.email("my_email@example.com", "Subject", "Your message goes here");
```

It also contains `to` field. With this field you may define receiver of email in the app. You may skip `to` field when you want to send email to your Blynk app login email:

```
Blynk.email("Subject", "Your message goes here");
```

You can send either `text/html` or `text/plain` (some clients don't support `text/html`) email. You can change this content type of email in the Mail widget settings.

Additionally you may use `{DEVICE_NAME}`, `{DEVICE_OWNER_EMAIL}` and `{VENDOR_EMAIL}` (for the local server) placeholders in the mail for the `to`, `subject` and `body` fields:

```
Blynk.email("{DEVICE_OWNER_EMAIL}", "{DEVICE_NAME} : Alarm", "Your {DEVICE_NAME} has critical error!");
```

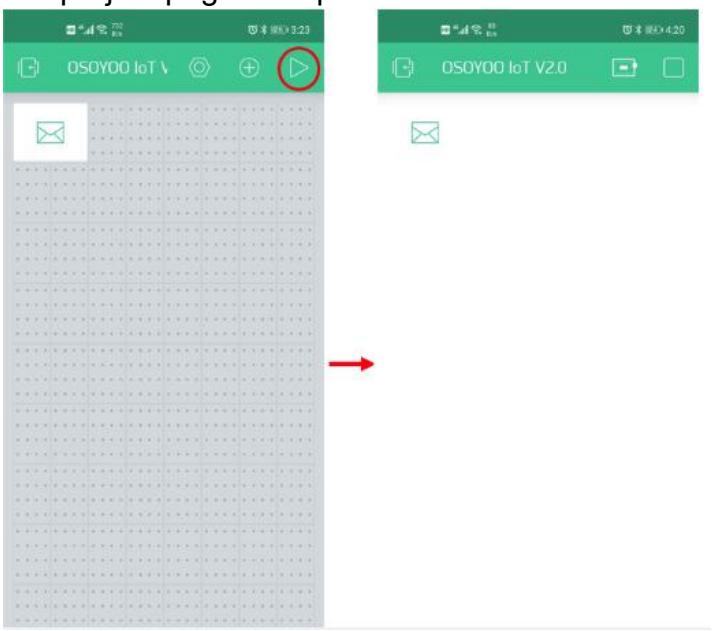
Limitations:

- Maximum allowed email + subject + message length is 120 symbols. However you can increase this limit if necessary by adding `#define BLYNK_MAX_SENDBYTES XXX` to you sketch. Where `XXX` is desired max length of your email. For example for ESP you can set this to 1200 max length `#define BLYNK_MAX_SENDBYTES 1200`. The `#define BLYNK_MAX_SENDBYTES 1200` must be included before any of the Blynk includes.
- Only 1 email per 5 seconds is allowed
- In case you are using gmail on the Local Server you are limited with 500 mails per day (by google). Other providers may have similar limitations, so please be careful.

- User is limited with 100 messages per day in the Blynk Cloud;

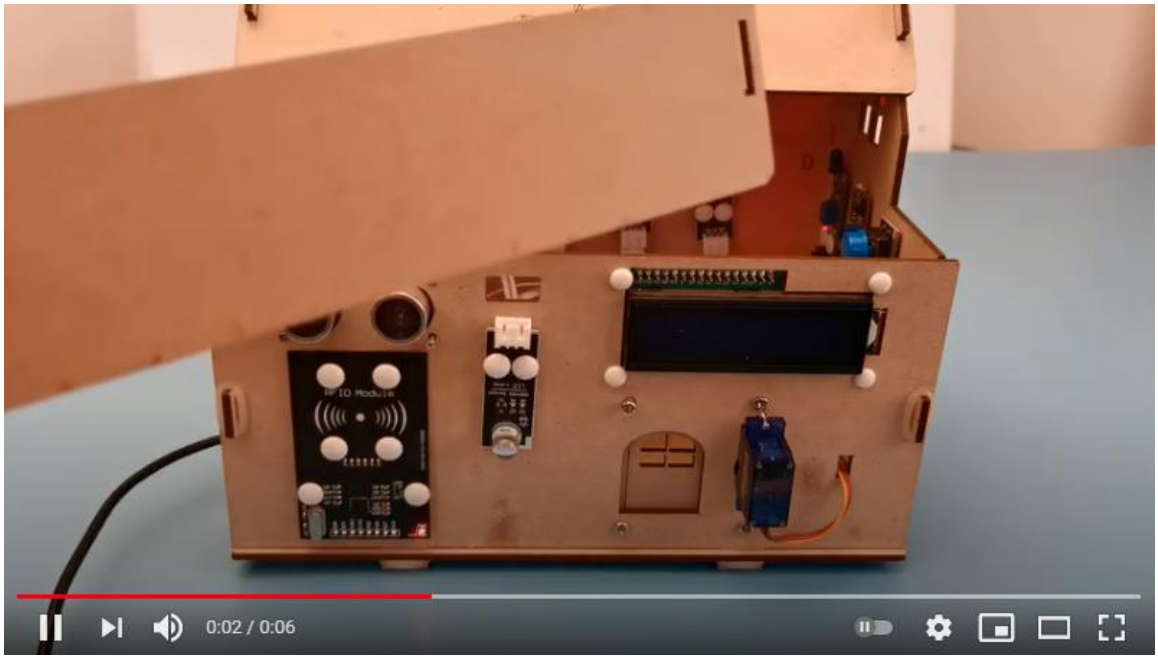
Sketch: Email

Then back to the project page. and press the "▷ " button to start your project.

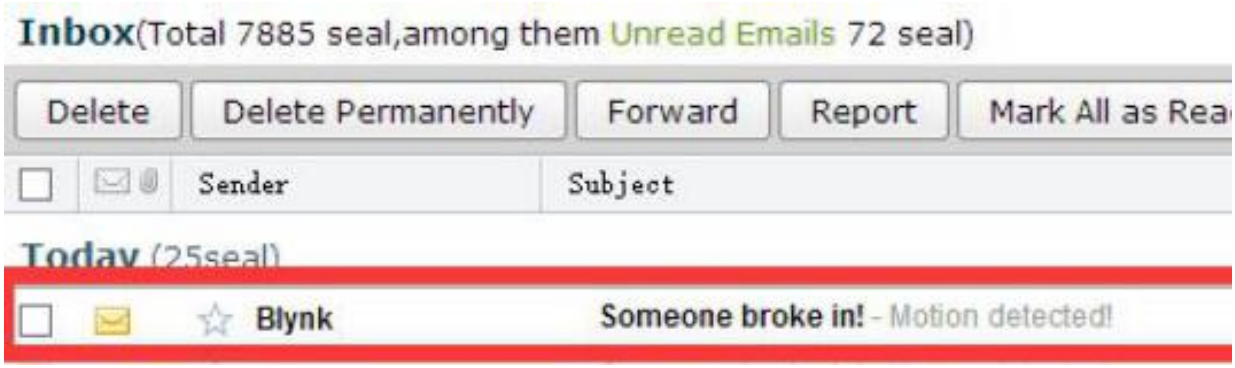


HOW TO PLAY

When you pass by the PIR sensor module, if the sensor detects human movement, the blue light on the sensor will be lit, the local buzzer will beep, and you will receive a warning email in your mailbox later. "Someone broke in!"
Look the video on Youtube :



https://www.youtube.com/embed/baHsi_5-EkM



3-2 Smart light

OBJECTIVE

In this lesson we will learn how to use this kit to make a smart light. In the dark night, the smart light will turn on when it detects human movement. During the day or no human activity is detected at night, the light will automatically turn off. Of course, you can also use your mobile phone to control the smart light.

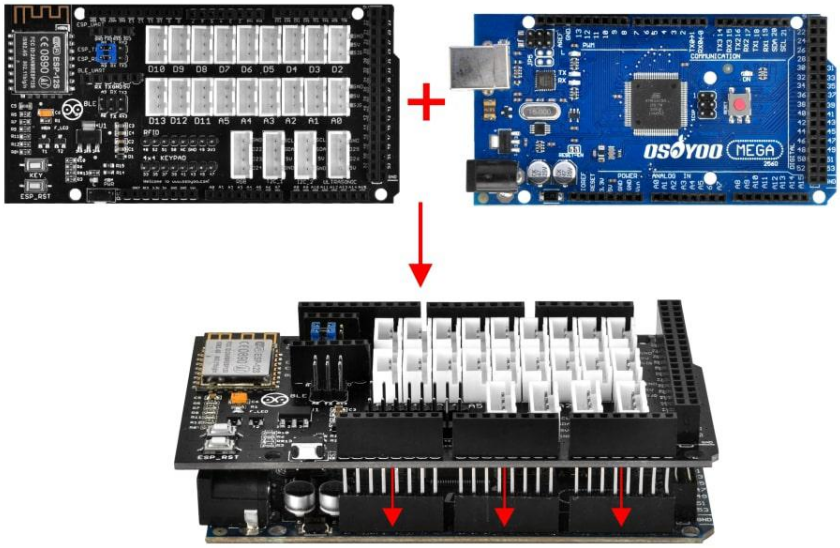
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
PIR Module x 1
Light Sensorx 1
LED Modulex 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



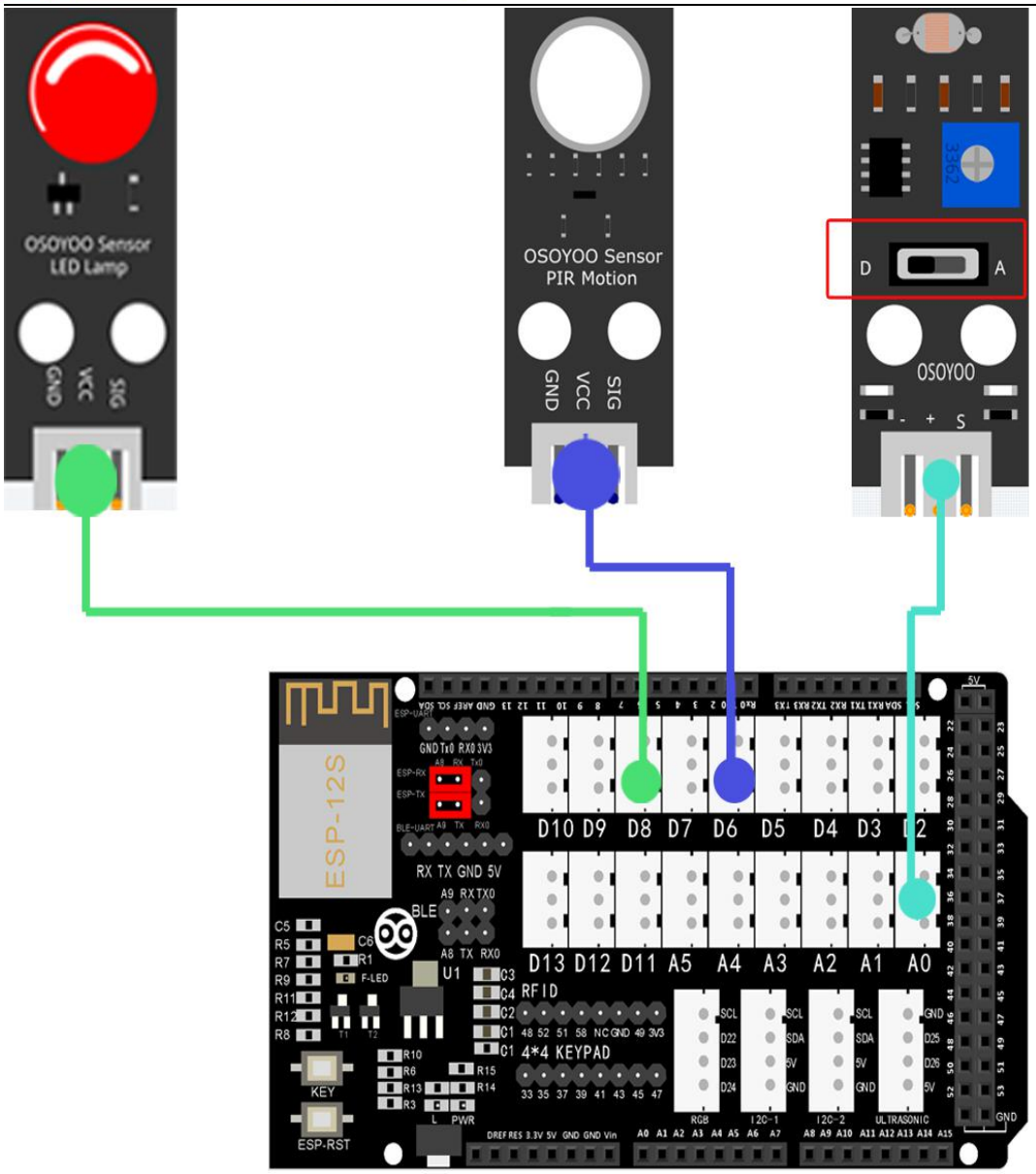
Build the circuit as below:

PIR sensor --- D6 ,

light detect sensor ---A0 port,

LED Module -- D8 port.

Note: Make sure the light sensor is on "A" mode(A for Analog, flip the switch on the sensor module to the letter "A" side).



HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

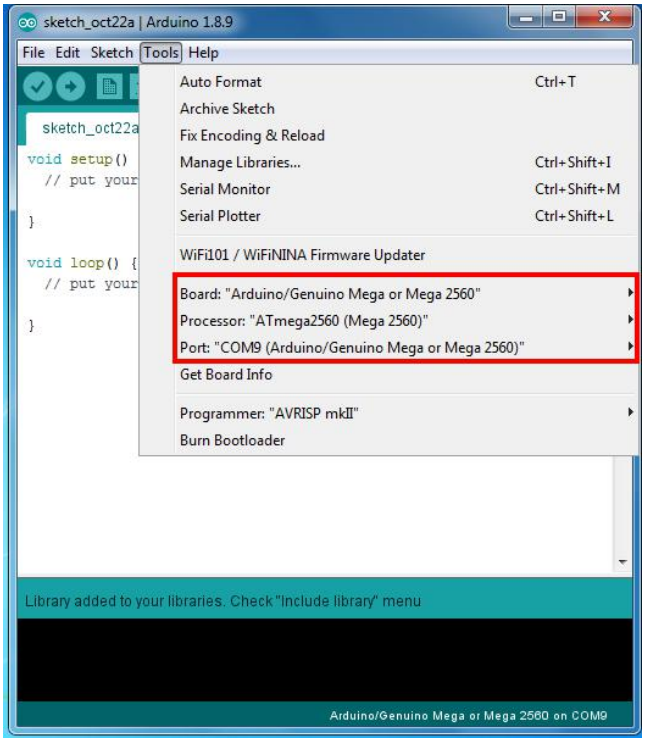
Step 3 Library Installation

Step 4 After installing above library, please download the code from following

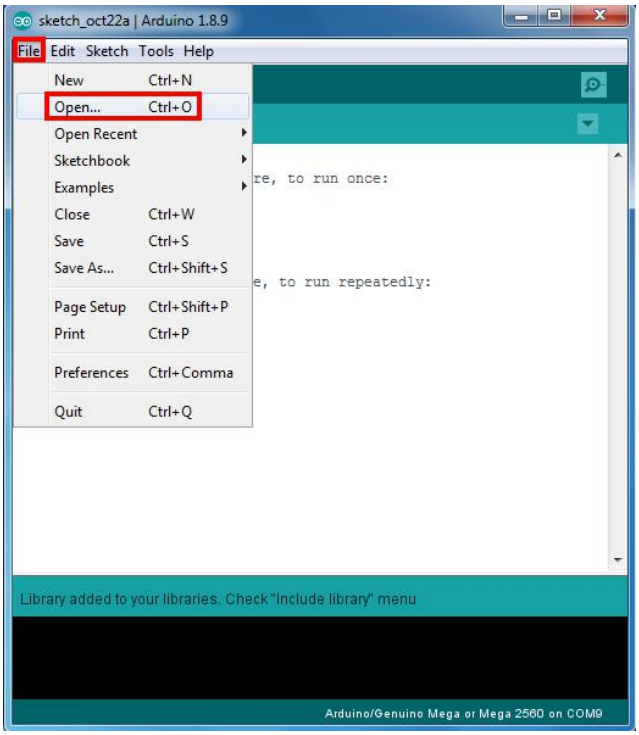
link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-2.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



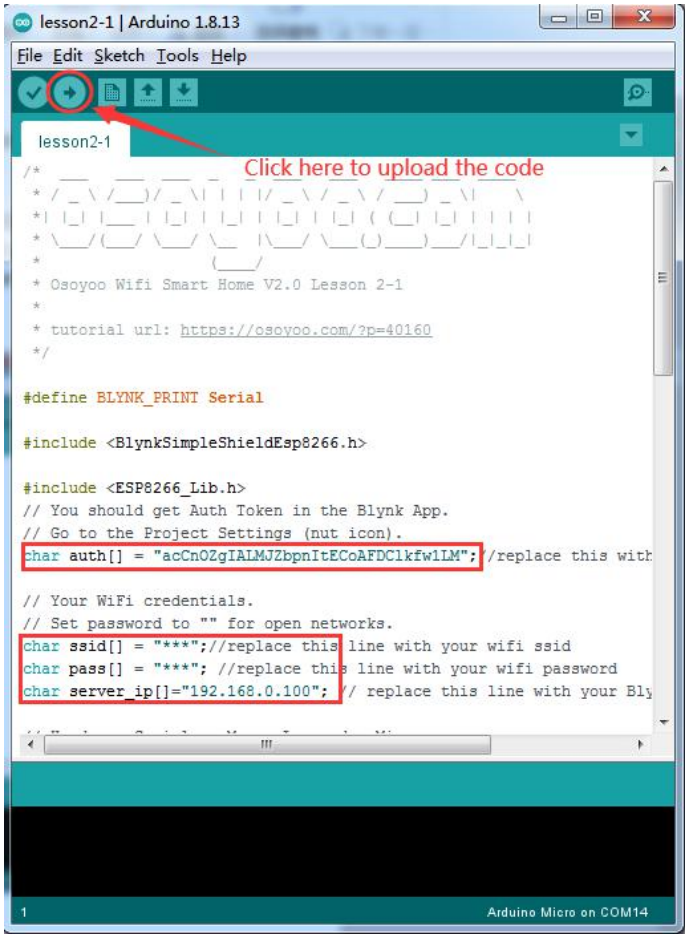
Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

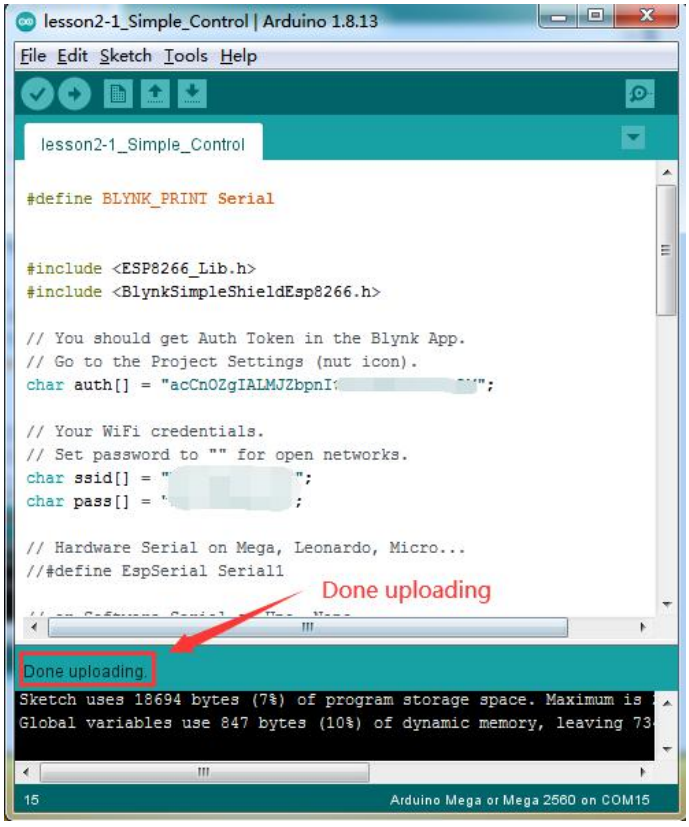
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:
Done uploading


lesson2-1 | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1
Click here to upload the code
* Osoyoo Wifi Smart Home V2.0 Lesson 2-1
* tutorial url: https://osoyoo.com/?p=40160
#define BLYNK_PRINT Serial
#include <BlynkSimpleShieldEsp8266.h>
#include <ESP8266_Lib.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "acCnOZgIALMJZbpnItECoAFDC1kfw1LM";//replace this with
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "***";//replace this line with your wifi ssid
char pass[] = "***"; //replace this line with your wifi password
char server_ip[]="192.168.0.100"; // replace this line with your Bly
1
Arduino Micro on COM14

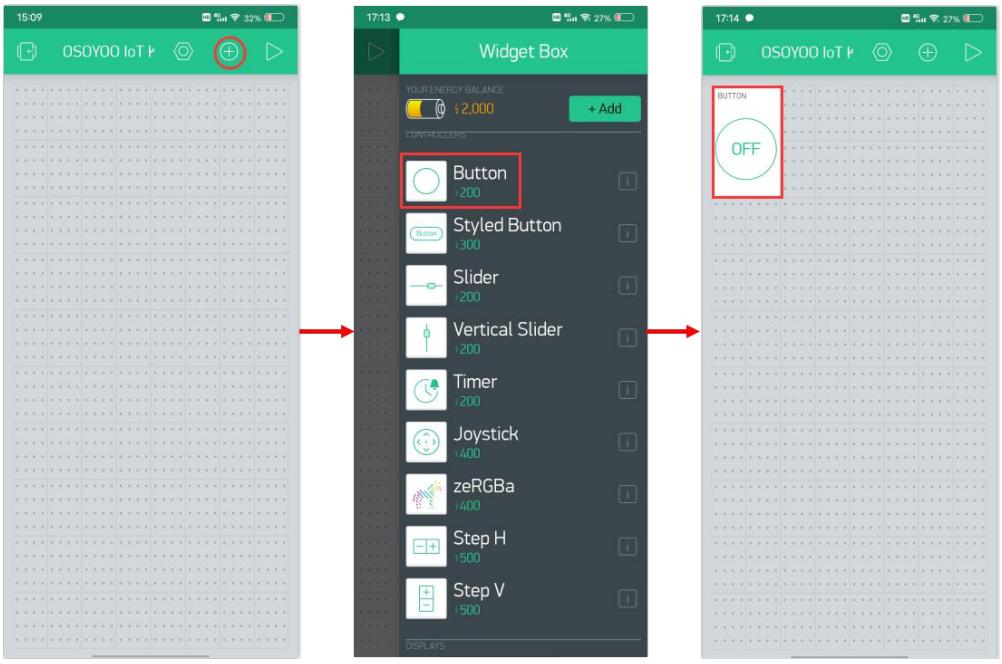

lesson2-1_Simple_Control | Arduino 1.8.13
File Edit Sketch Tools Help
lesson2-1_Simple_Control
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h>
#include <BlynkSimpleShieldEsp8266.h>
// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
// Your WiFi credentials.
// Set password to "" for open networks.
// Hardware Serial on Mega, Leonardo, Micro...
//#define EspSerial Serial1
Done uploading
Done uploading.
Sketch uses 18694 bytes (7%) of program storage space. Maximum is
Global variables use 847 bytes (10%) of dynamic memory, leaving 73
15
Arduino Mega or Mega 2560 on COM15

Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

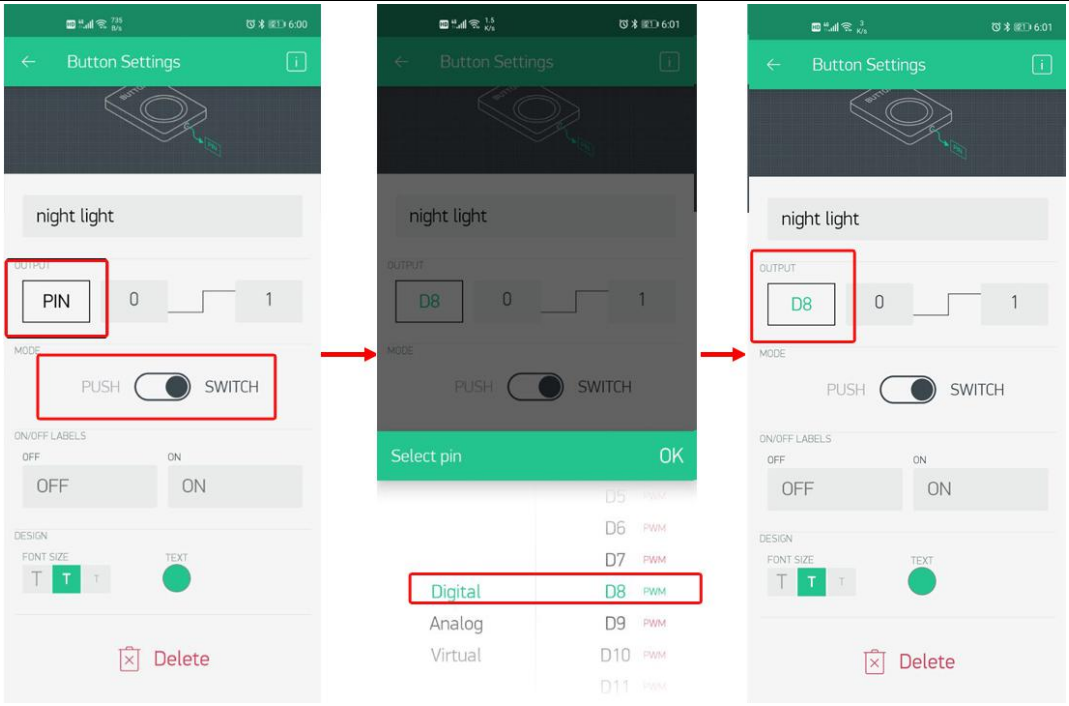
Add a Widget in Blynk (legacy) APP

Open the Blynk APP , Follow the next operations:
Open your project page and press the "+" button to add the ″Button" Widget.

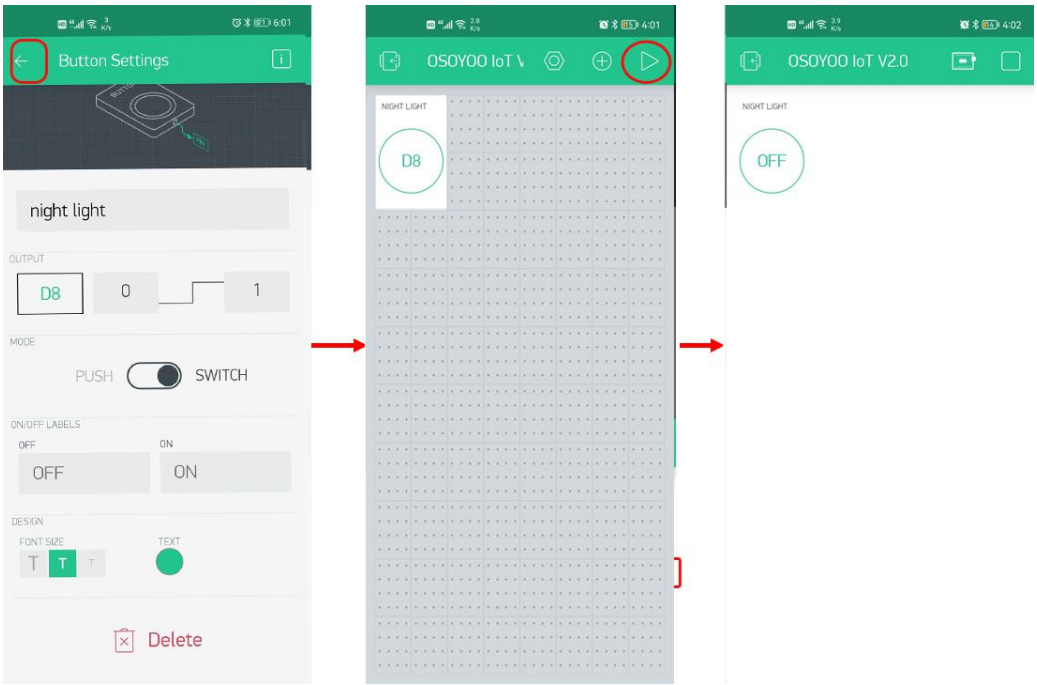


Button Widget settings:

Name Column: You can name your widget, whatever you like.
OUTPUT: D8| 0-1
Select the "Switch"



Then back to the project page. and press the "▷ " button to start your project.

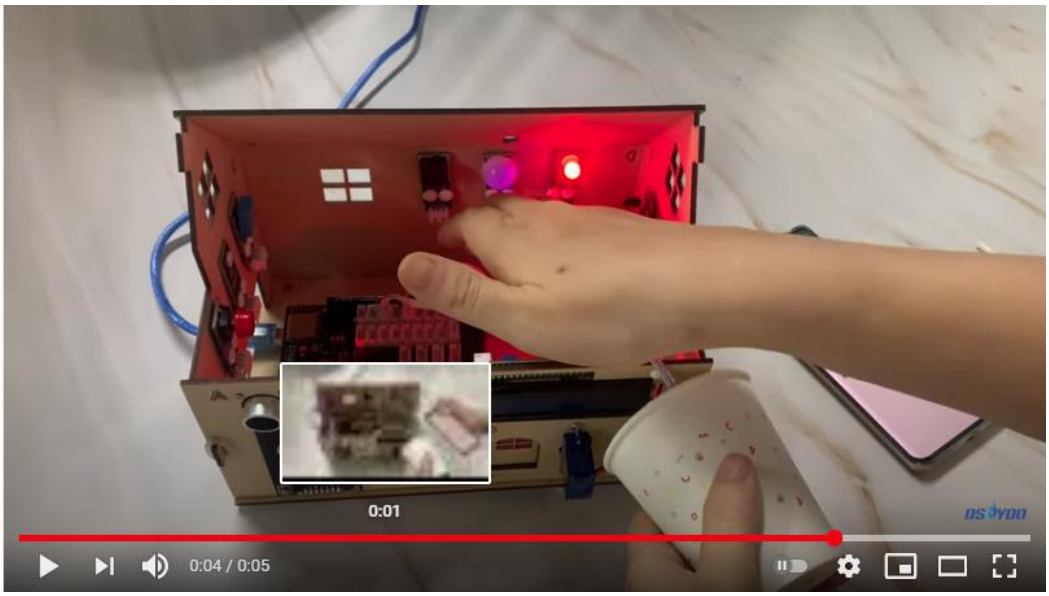


HOW TO PLAY

When the value from light sensor is low, the smart light will turn on when it detects human movement. Otherwise, the smart light will automatically turn

off.

Look the video on Youtube :



https://www.youtube.com/embed/yk2oax01pjQ

3-3 Remote fire alarm

OBJECTIVE

In this tutorial, we will learn how to use OSOYOO IoT kit with Blynk APP to make a remote fire alarm. When the fire alarm detects a flame, it will trig alarm and the white warning light will turn on. At the same time, the Blynk APP will also push fire alarm notifications .

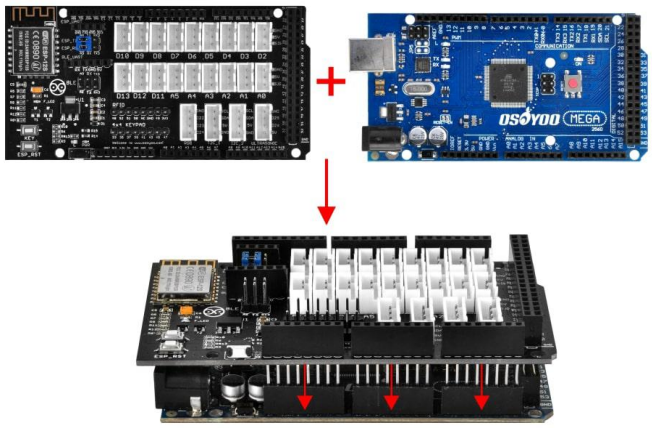
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
Fire Module x 1
Active Buzzerx 1
White LED Modulex 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library

HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board

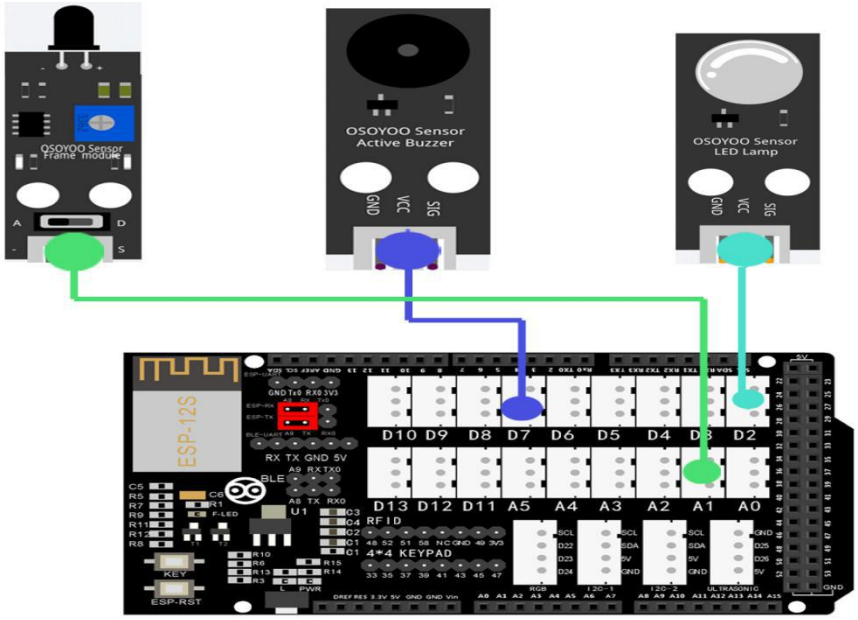


Build the circuit as below:

Flame sensor- A1 Make sure the Fire sensor is on "A" side.

Active buzzer - D7

White LED- D2



HOW TO CODE

Step 1 Prerequisite

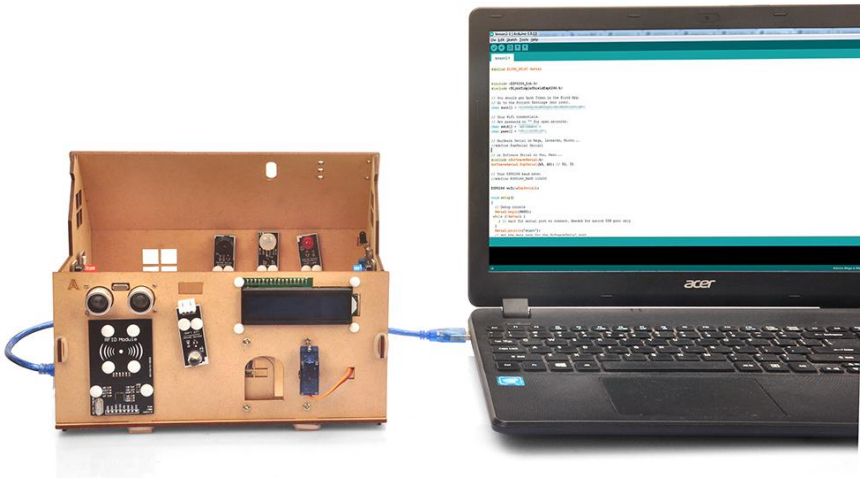
Step 2 Install latest Arduino IDE

Step 3 Library Installation

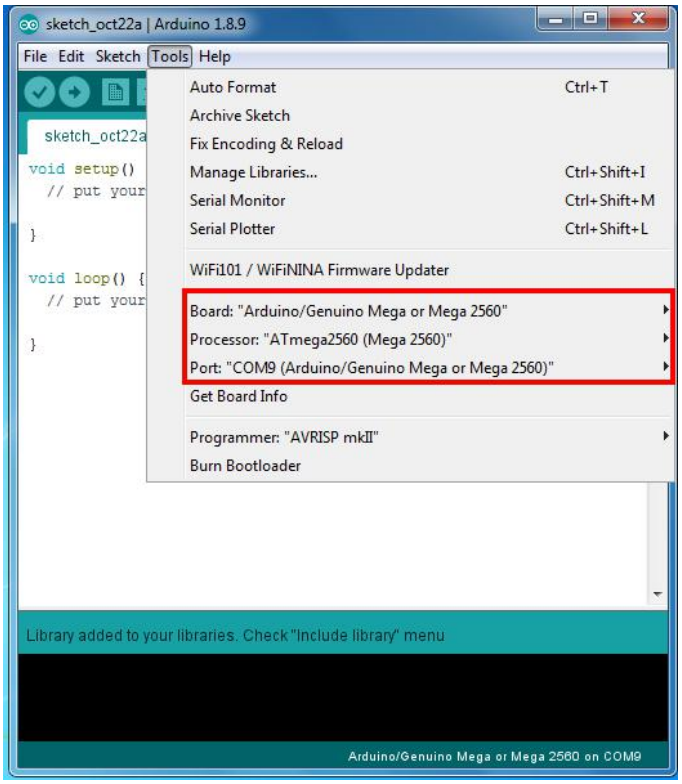
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-3.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

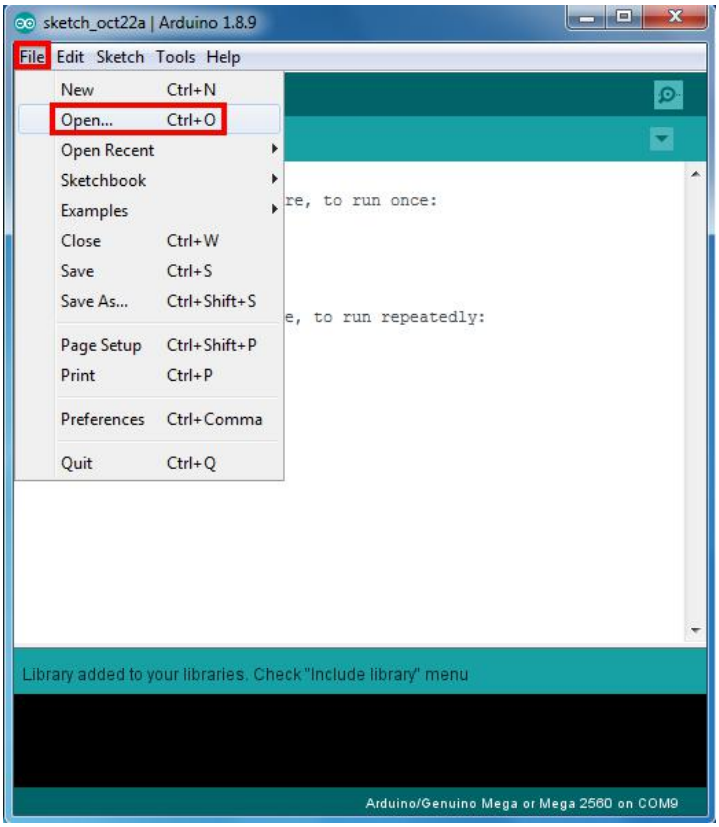
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.

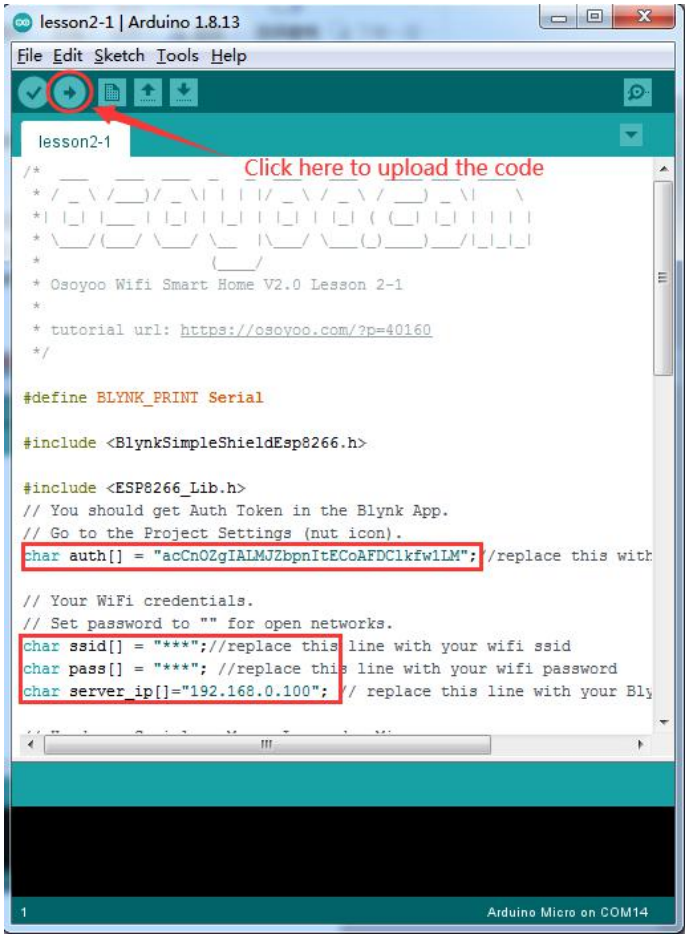


Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

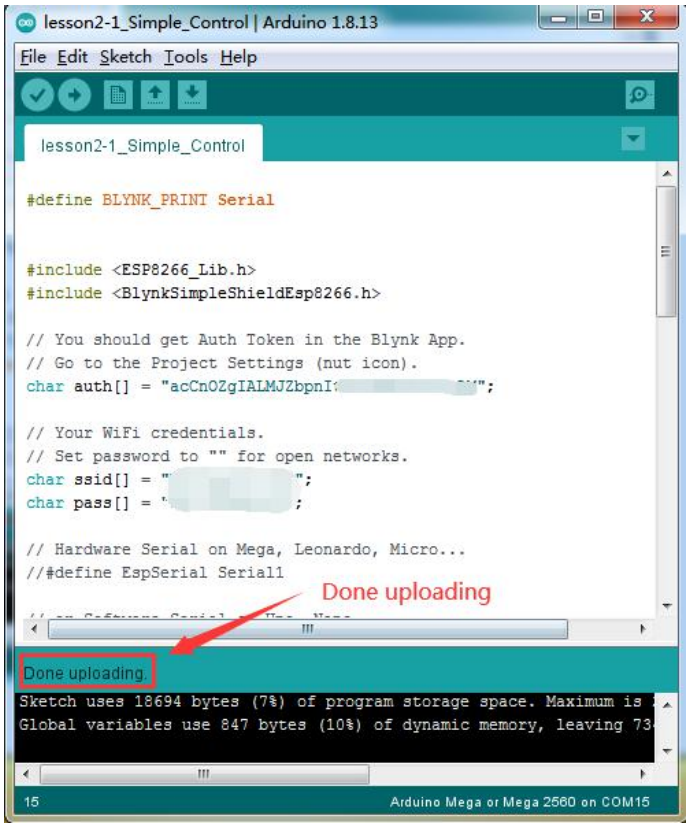
Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.



Then edit the email:

Finally Upload the sketch to the board. Wait until you see something like this:
Done uploading

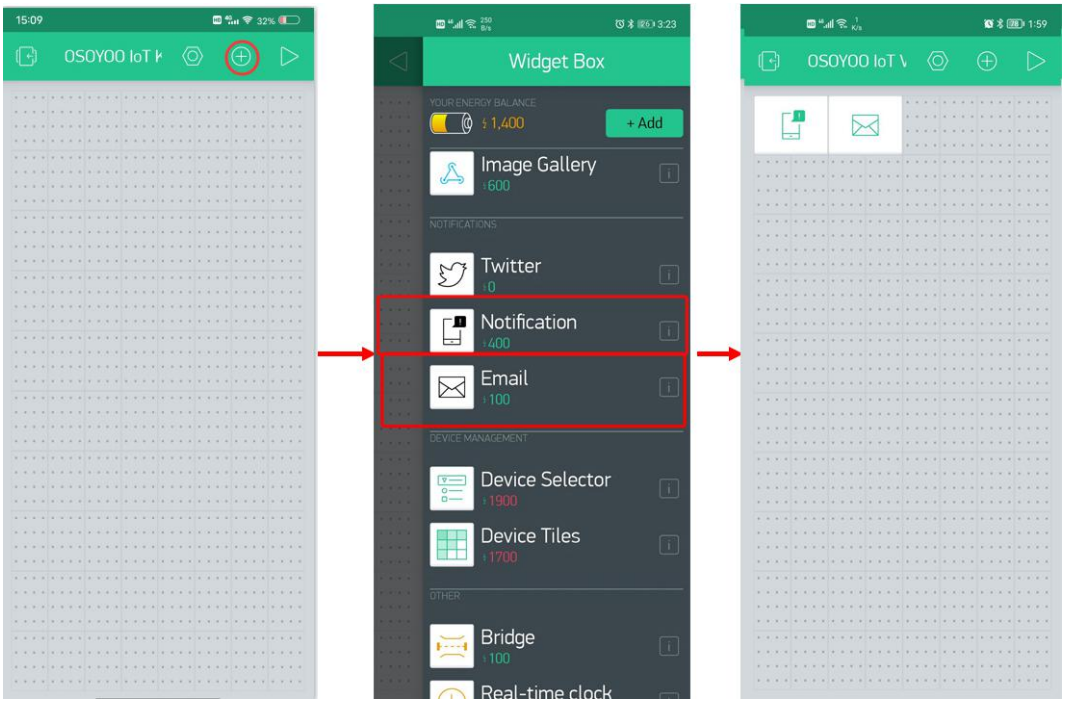


Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

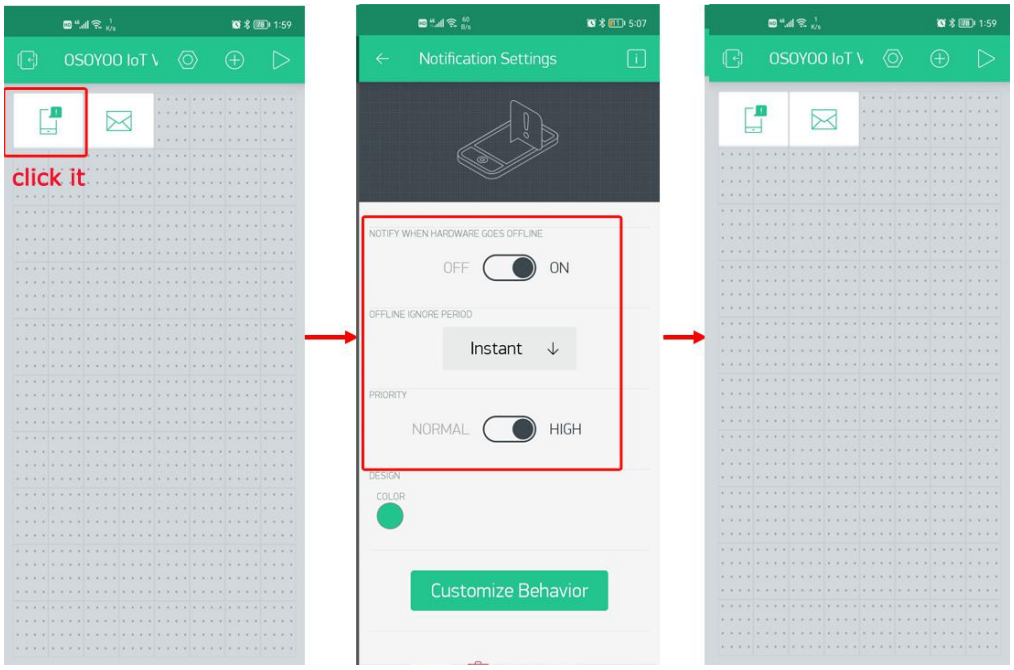
Open the Blynk APP , Follow the next operations:
Press the "+" button to add "Notification ″ "Button" Widget.



Notification Widget settings:

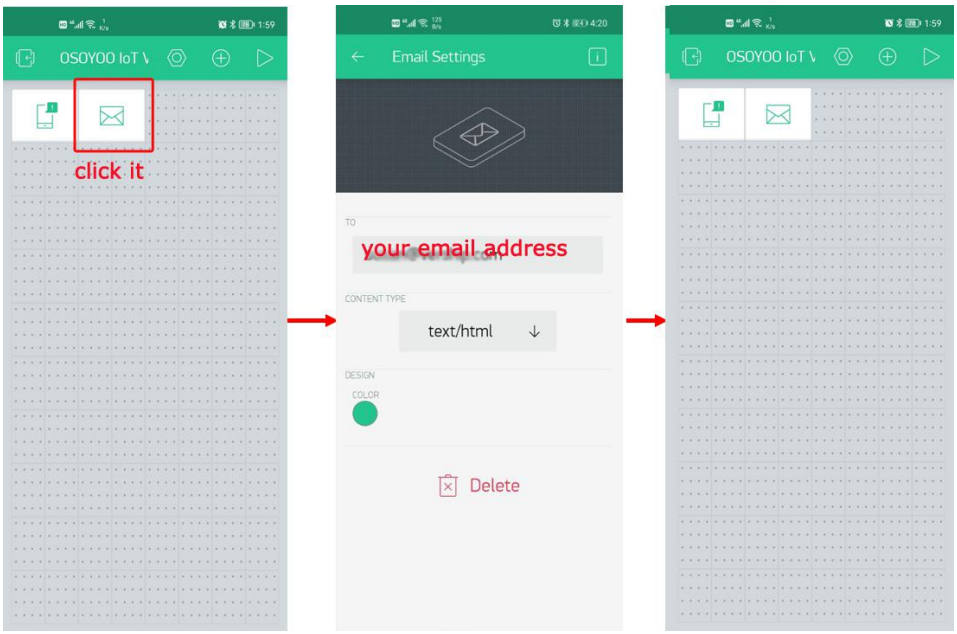
- Notify when hardware goes offline: ON
- Offline ignore period: Instant
- Priority: HIGH

Then set the Blynk notification to the highest authority in the phone settings.

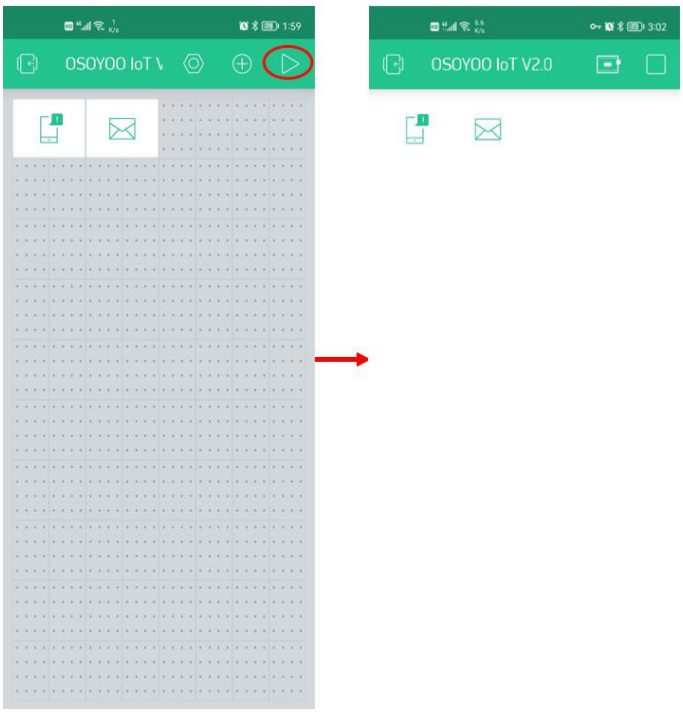


Email Widget settings:

Type your email address there and keep other settings as default.



Then back to the project page. and press the "▷ " button to start your project.



Push Notifications (you can skip it)

Push Notification widget allows you to send push notification from your hardware to your device. Currently it also contains 2 additional options:

- **Notify when hardware offline** - you will get push notification in case your hardware went offline.
- **Offline Ignore Period** - defines how long hardware could be offline (after it went offline) before sending notification. In case period is exceeded - “hardware offline” notification will be send. You will get no notification in case hardware was reconnected within specified period.
- **Priority** high priority gives more chances that your message will be delivered without any delays. See detailed explanation here.

WARNING: high priority contributes more to battery drain compared to normal priority messages.

Example code:

```
Blynk.notify("Hey, Blynkers! My hardware can push now!");
```

You can also use placeholder for device name, that will be replaced on the server with your device name:

```
Blynk.notify("Hey, Blynkers! My {DEVICE_NAME} can push now!");
```

Limitations:

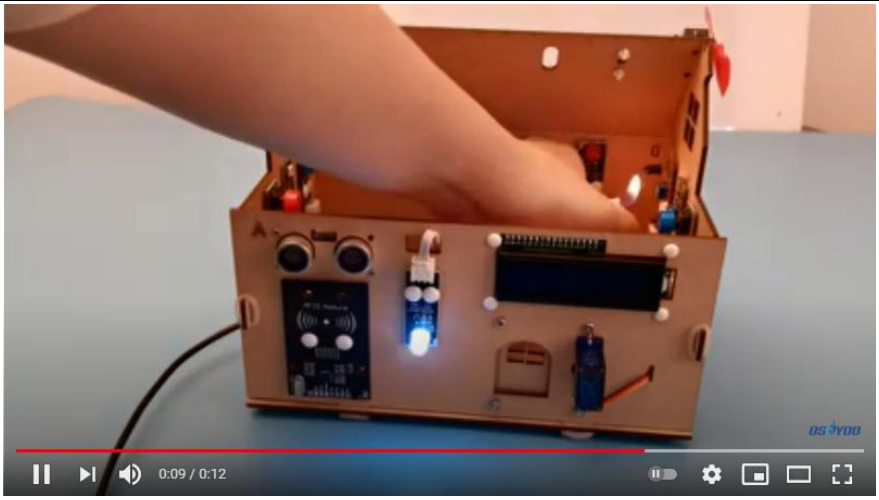
- Maximum allowed body length is 120 symbols;
- Every device can send only 1 notification every 5 seconds;

Sketch: PushNotification

HOW TO PLAY

When the fire alarm detects a flame, it will sound an alarm, and the white warning light will be light. At the same time, the Blynk APP will also push fire alarm noticications and emails to your phone.

Look the video on Youtube :



https://www.youtube.com/embed/ZyNpEVzOjwE

Note:

If you find the device is offline, please check your code, wiring and app settings, and try again.

If the mobile phone does not receive the push notification of Blynk APP or does not receive the email, please click the below settings:

- **Blynk server or local server.** The local server may not push notifications
- **Blynk Library version.** Please make sure it is the latest version but not a Beta version

- **Blynk APP version.** Please make sure it is the latest version but not a Beta version
- **Priority.** High priority gives more chances that your message will be delivered without any delays. See detailed explanation here.
- **Mobile network settings.** Blynk uses the messaging service provided by Google. If your phone cannot access Google servers normally, you will not receive push notifications, but you can still use email notifications normally.

If the above does not solve your problem, please search for similar topics in the Blynk forum https://community.blynk.cc/, or check http://docs.blynk.cc and http://help.blynk.cc.

3-4 Multifunctional intelligent access control system

OBJECTIVE

In this tutorial, we will learn how to use OSOYOO IoT kit in combination with Blynk APP to make a multifunctional access control system. This system can open the door remotely through the mobile phone APP or use the access card to open the door, after opening the door, the mobile phone will receive the APP notifications, and the local LCD screen will also display the door opening information.

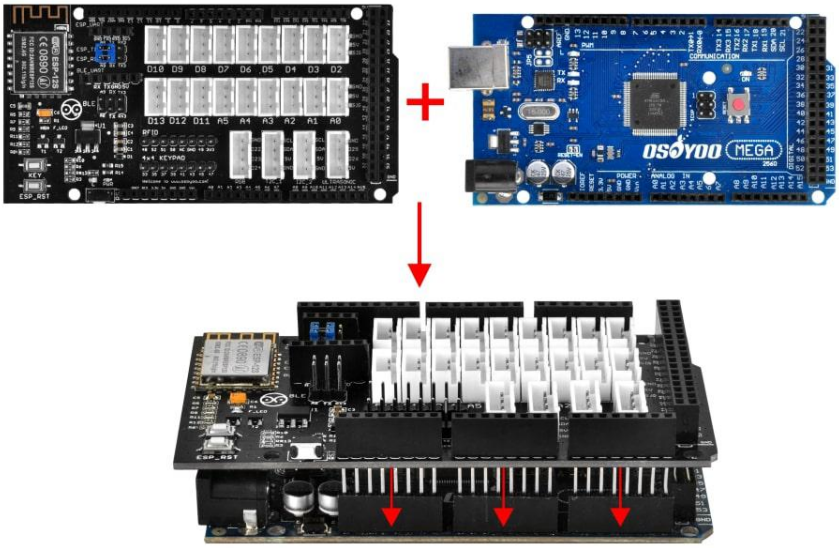
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
RFID Module x 1
RFID Card x 1
Servo x 1
I2C 1602 LCD Screen x 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
I2C library
RFID library

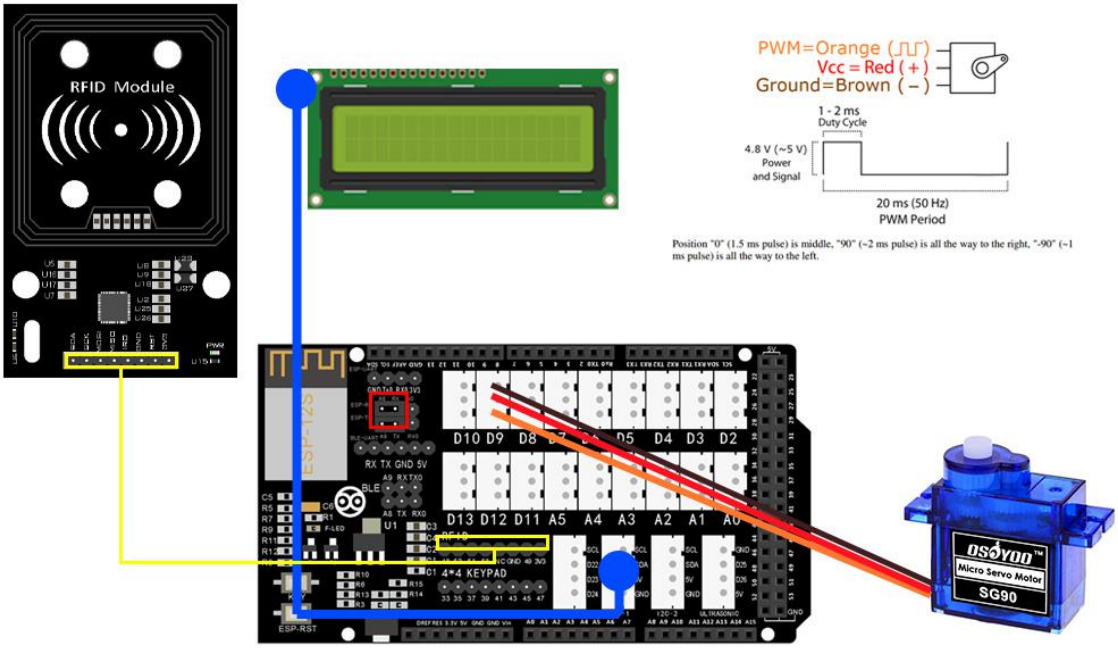
HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:

I2C 1602 LCD Display----I2c
RFID Module ---- RFID
servo ------- D9



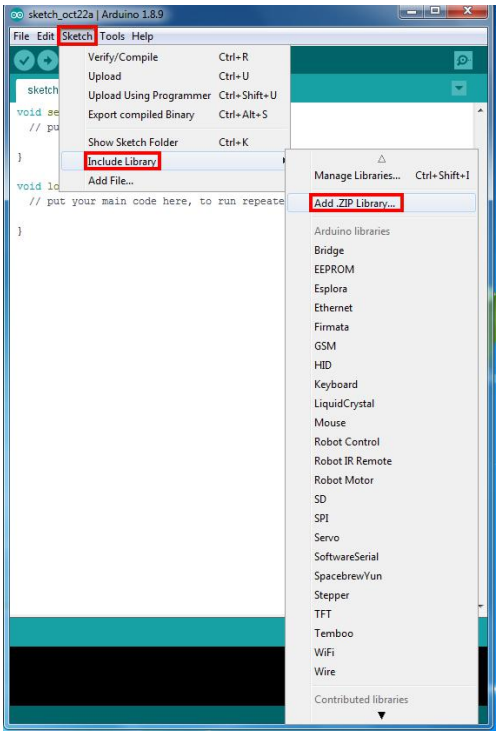
HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

Step 3 Library Installation

install the RFID and I2C library



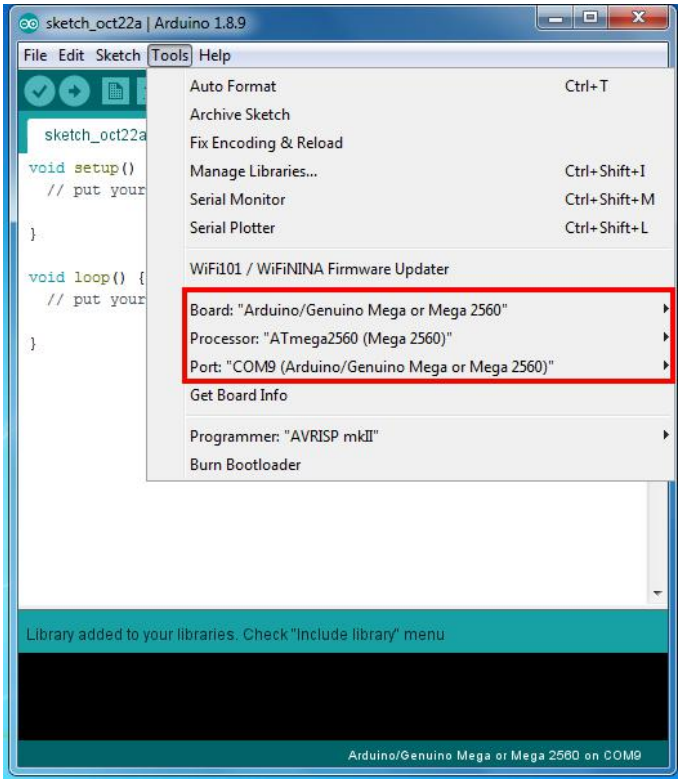
Step 4 Servo Test (Step 4 -Step 9) If you adjust the servo before, please skip this step.

Step 5 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-4.zip

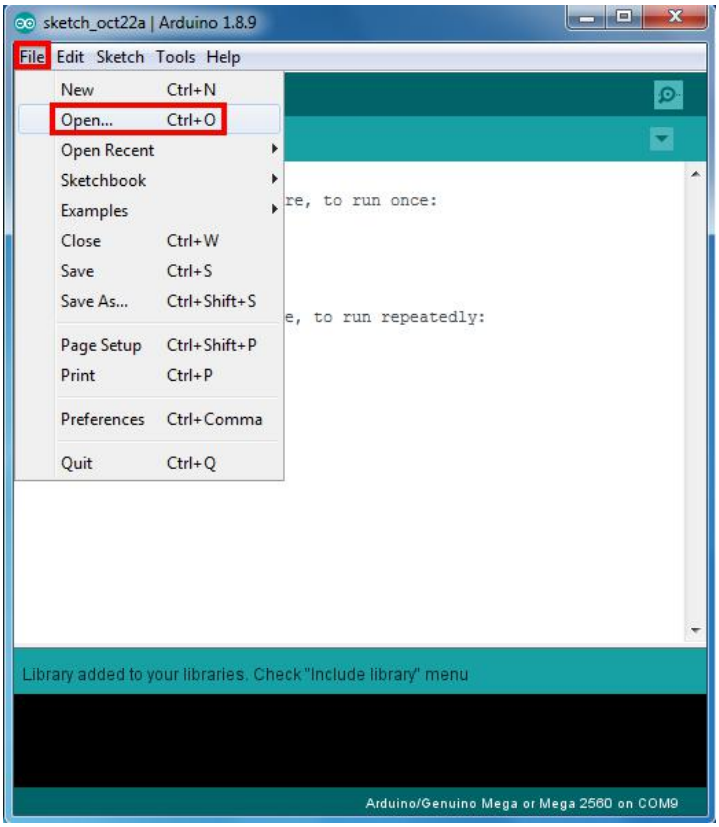
Step 6 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.

Step 7 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 8 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.

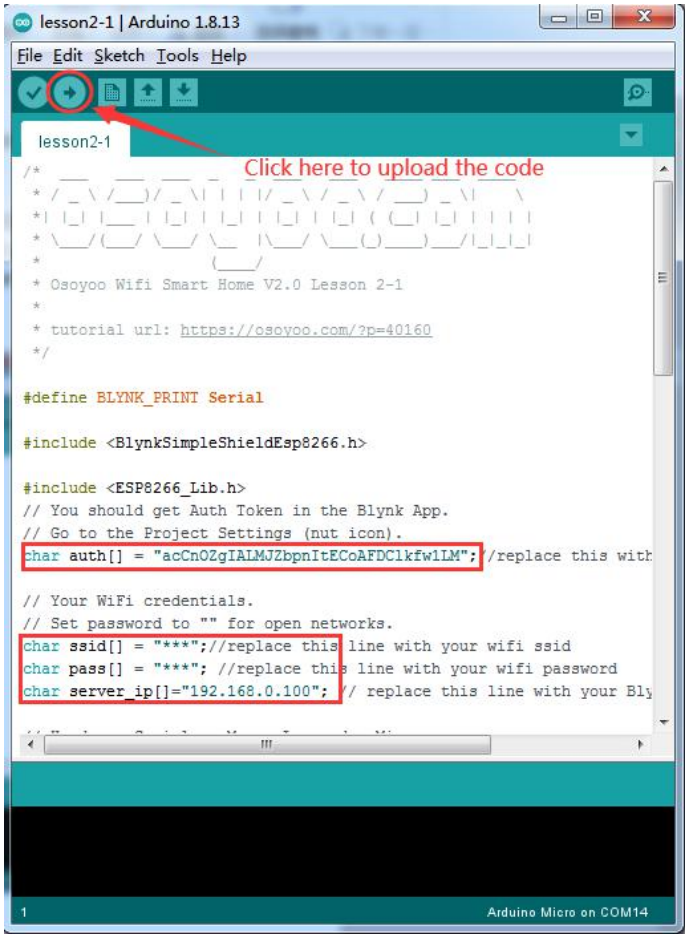


Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "*******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

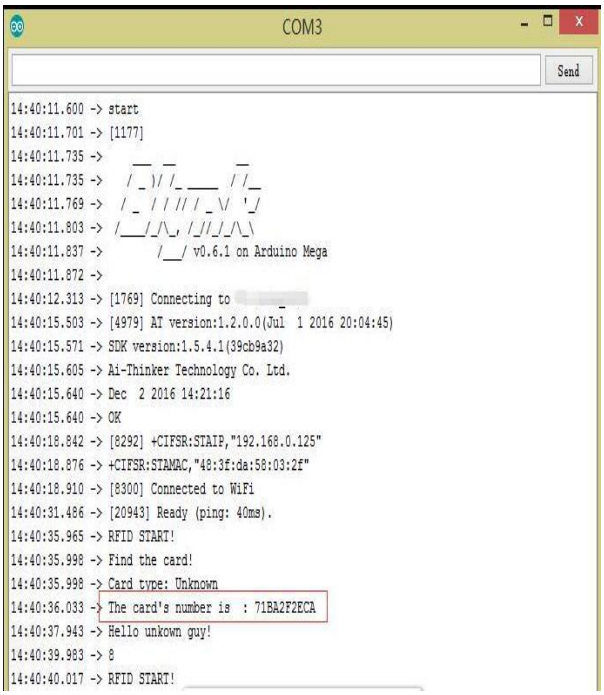
please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.



Then add your own RFID Card id:

After completing the above steps and uploading the code, open the IDE serial monitor, you can see that the system is initialized successfully, and then put your RFID card in the RDID module recognition area, you will see your serial monitor will display your The card information is as follows:

RFID START!
Find the card!
Card type: Unknown
The card's number is : 71BA2F2ECA



Then find these lines:

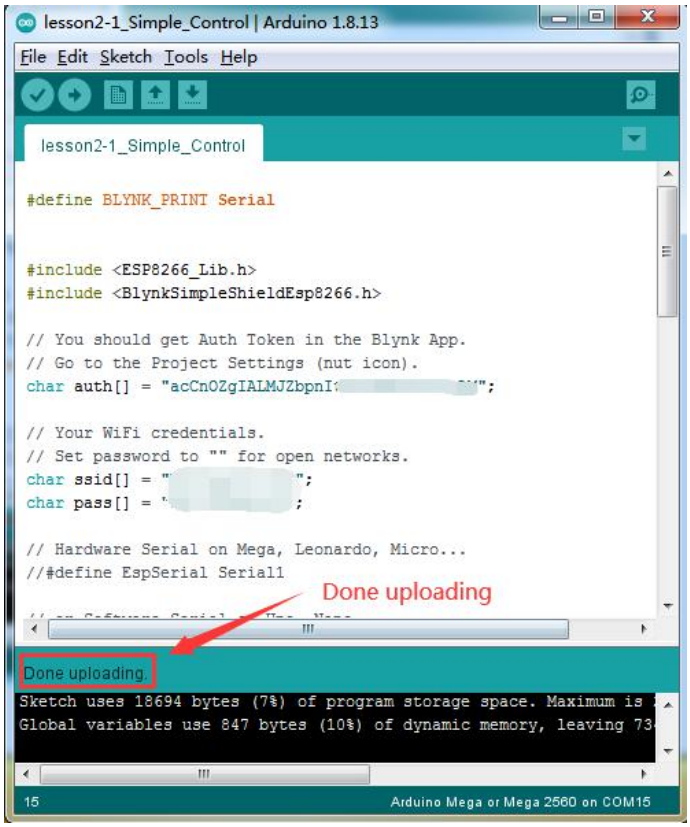
```
if( id[0]==0x32 && id[1]==0xDA && id[2]==0x94 && id[3]==0x10 )
```

And change the id data to your own card number, so the changed code is:

```
if( id[0]==0x71 && id[1]==0xBA && id[2]==0x2F && id[3]==0x2E )
```

Upload the sketch to the board again. Wait until you see something like this:

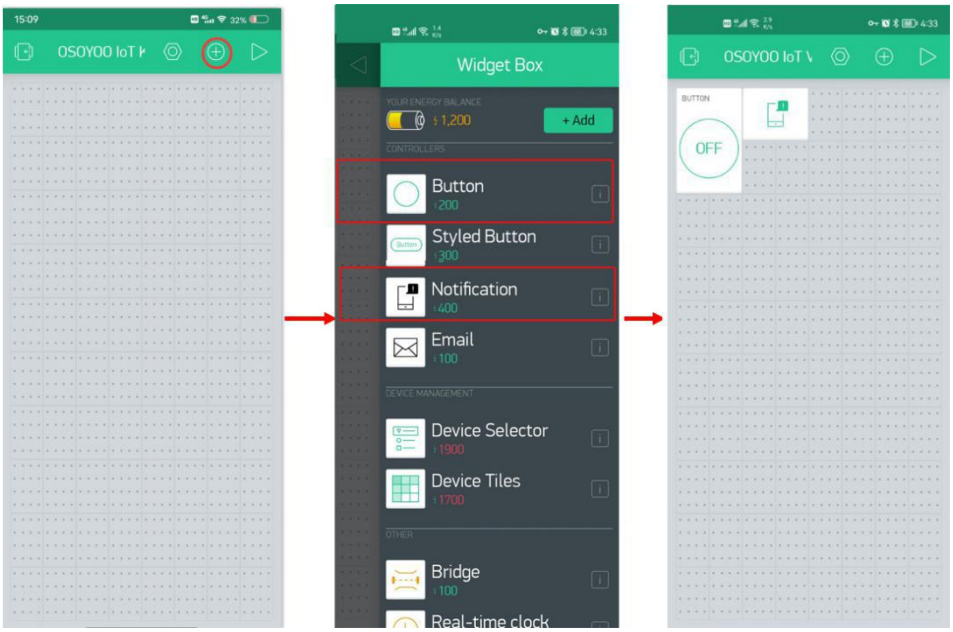
Done uploading



Add a Widget

Open the Blynk APP , Follow the next operations:

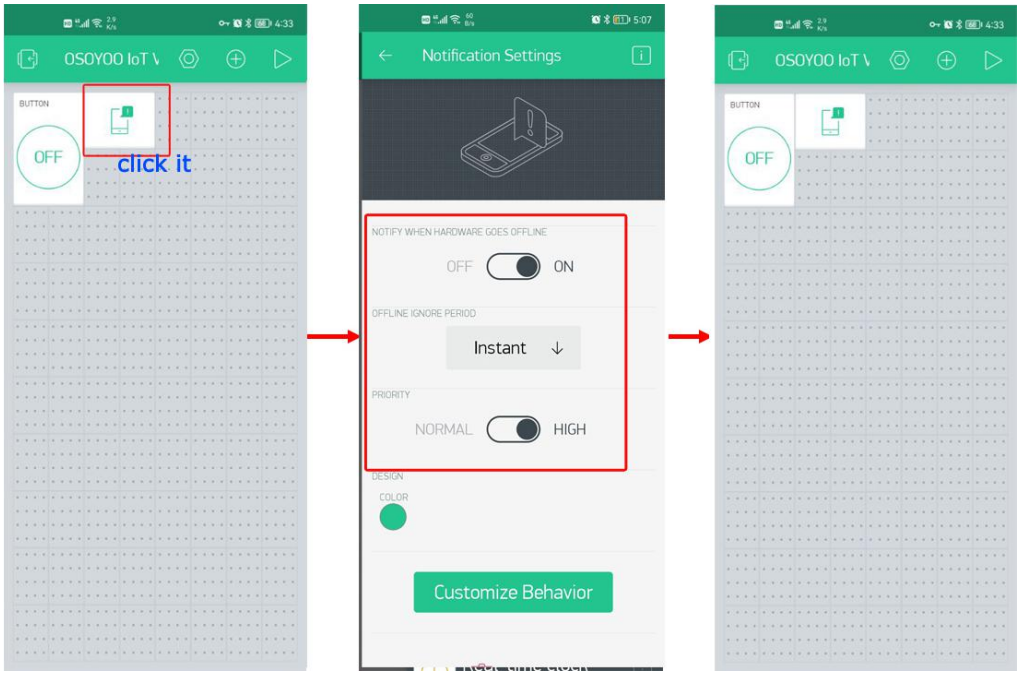
Open your project page and press the “+” button to add the “Notification” ″Button" Widget.



Notification widget:

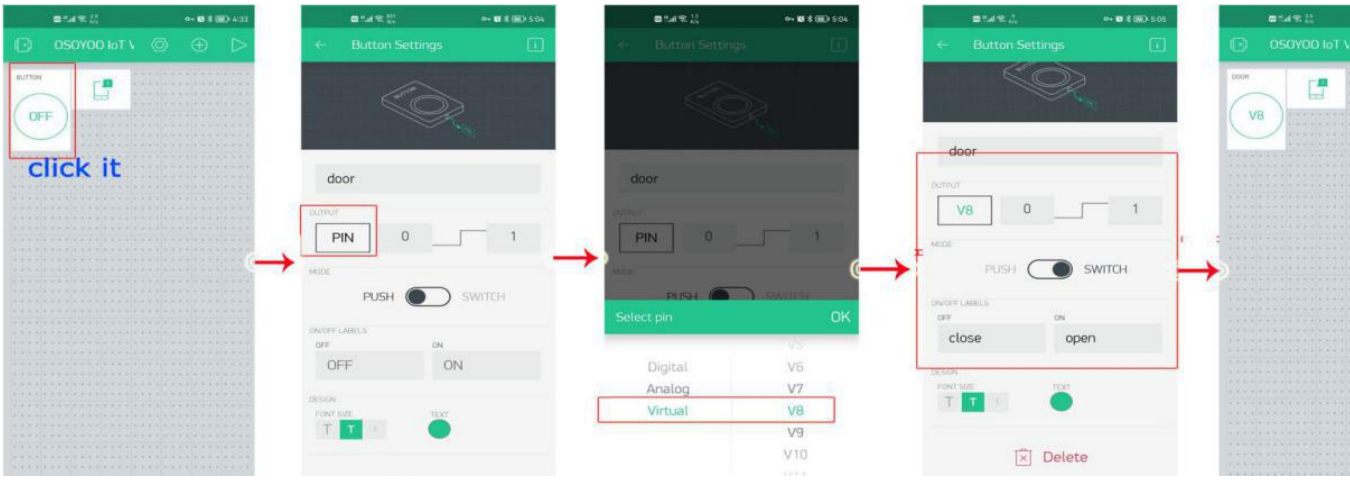
- Notify when hardware goes offline: ON
- Offline ignore period: Instant
- Priority: HIGH

Then set the Blynk notification to the highest authority in the phone settings.



Button Widget settings:

Name Door
OUTPUT: V8| 0-1
Select the "Switch"
ON/OFF LABELS: OFF-CLOSE, ON-OPEN



Then back to the project page. and press the "▷ " button to start your

project.

HOW TO PLAY

As the creator of this system, when you stand at the door of your house, you can use the RFID card in your hand to open the door. When you open the door, you can see the screen prompt at the door: "Hello, David! Welcome home!" , At the same time, Blynk on your mobile phone will also receive a push reminder: "David is home!". When other family members open the door with an RFID card, a similar prompt will appear on the door screen. The same is true for your mobile phone, so you can Know who has gone home.

If there is no one at home when a friend visits, you can use the Blynk APP to open the door for your friend remotely, and the screen at the door will display "Hello, guys!".

When someone wants to open the door with an unverified RFID card, the screen at the door will show an "Unknow user!" warning, and your phone will also receive a push notification "Unidentified user is ready to open the door!"

Look the video on Youtube :

https://www.youtube.com/embed/FazVGwMUnXk

3-5 Environmental Data Monitoring & Regulation System

OBJECTIVE

In this tutorial, we will learn how to use OSOYOO IoT kit and Blynk APP to build an intelligent system for environmental detection and regulation. Through this system, we can remotely obtain the temperature and humidity of the environment in real time, and display the data to the local display. When the temperature is higher than the warning value, the mobile phone will receive a push notification, the air circulation fan will be automatically turned on, and the temperature will be adjusted actively. , When the temperature is lower than the warning value, the fan will automatically shut down.

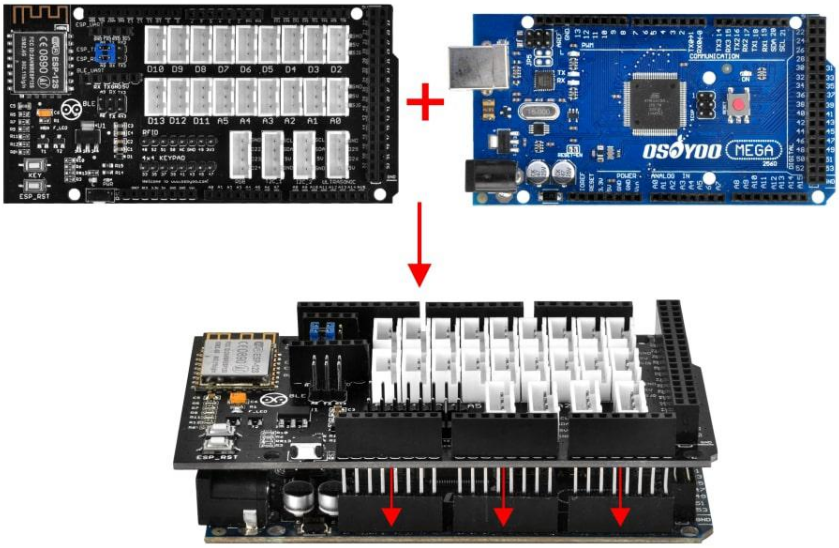
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
DHT11 Module x 1
Fan Module x 1
I2C 1602 LCD Screenx 1
USB Cable
Jumpers

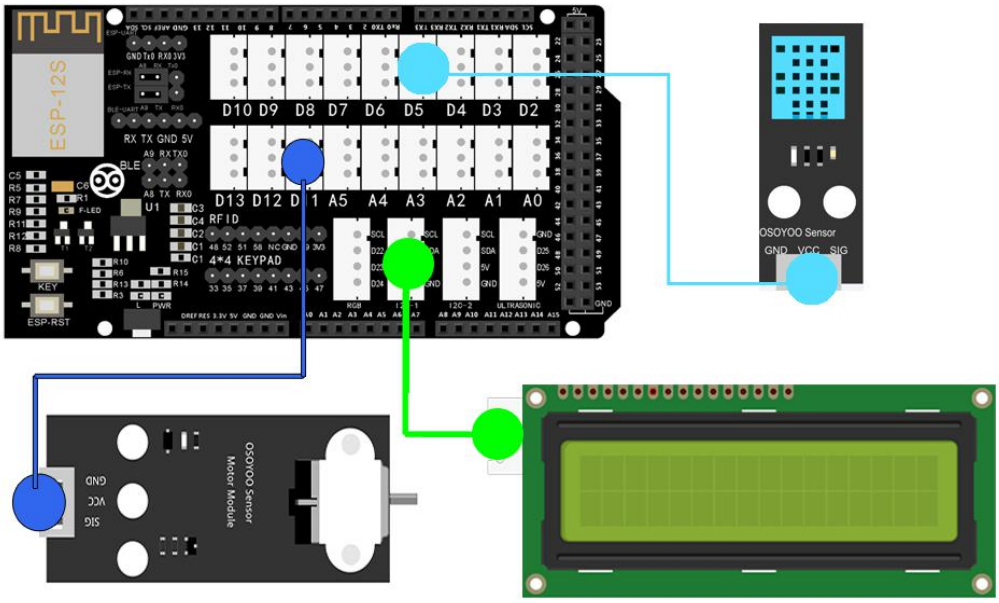
SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
I2C library

HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:
1602 display - I2C
DHT11 Module - D5
Fan Module - D11



HOW TO CODE

Step 1 Prerequisite

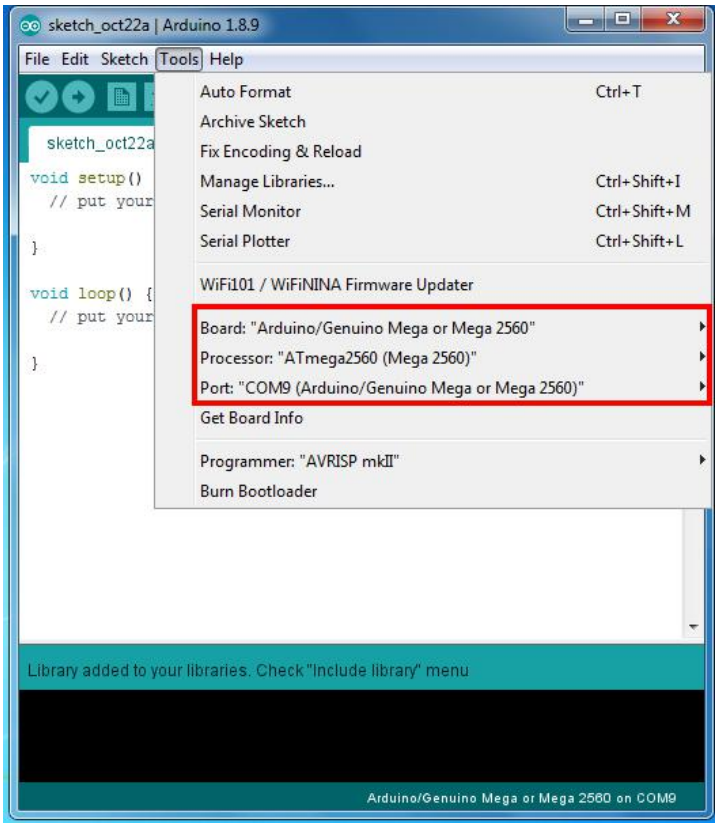
Step 2 Install latest Arduino IDE

Step 3 Library Installation

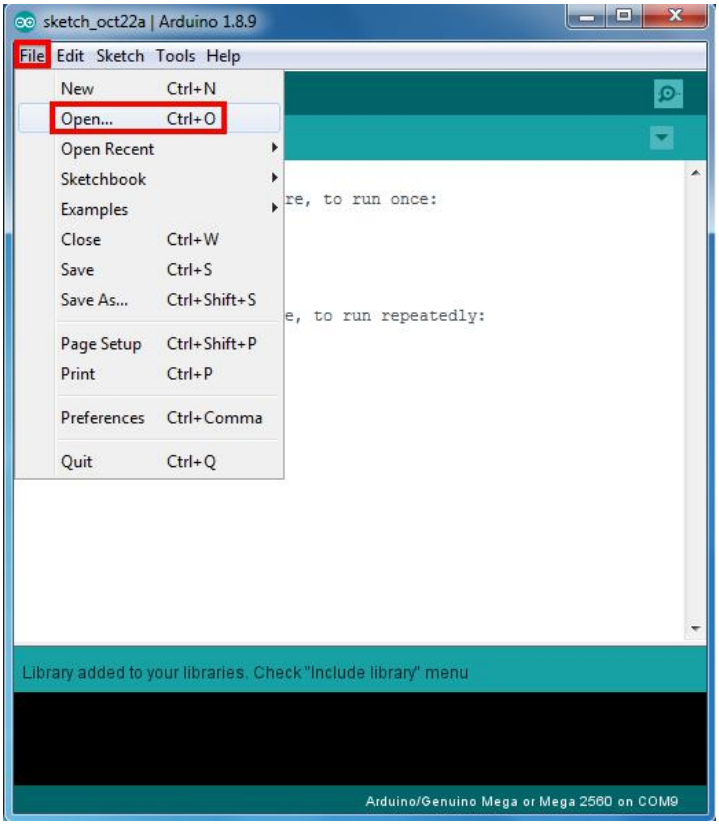
Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-5.zip

Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.

Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



Note: In the sketch, find this line as following:

char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:

*char ssid[] = "******"; // your network SSID (name)*

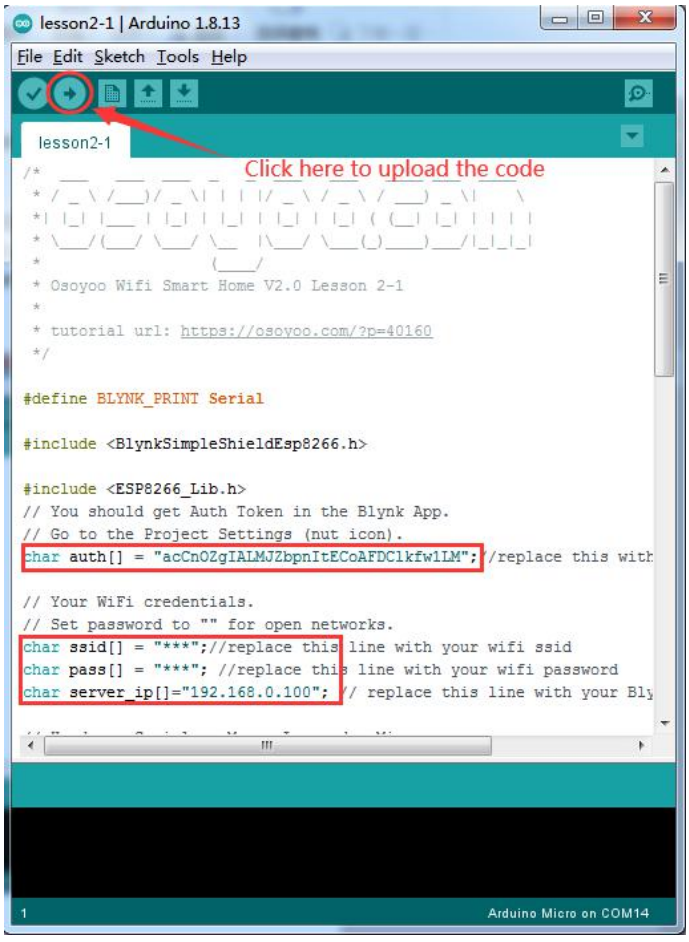
*char pass[] = "******"; // your network password*

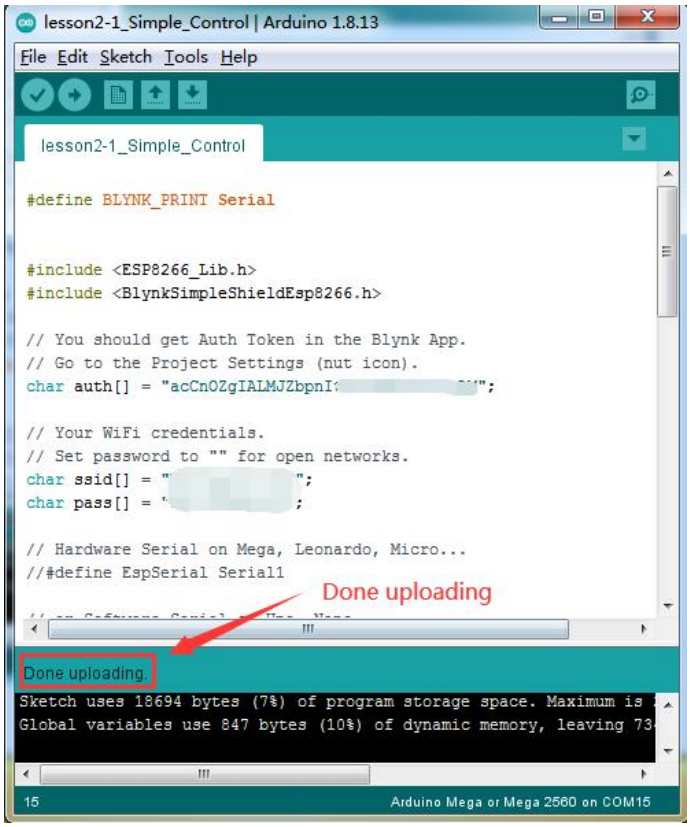
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.

Upload the sketch to the board. Wait until you see something like this:

Done uploading



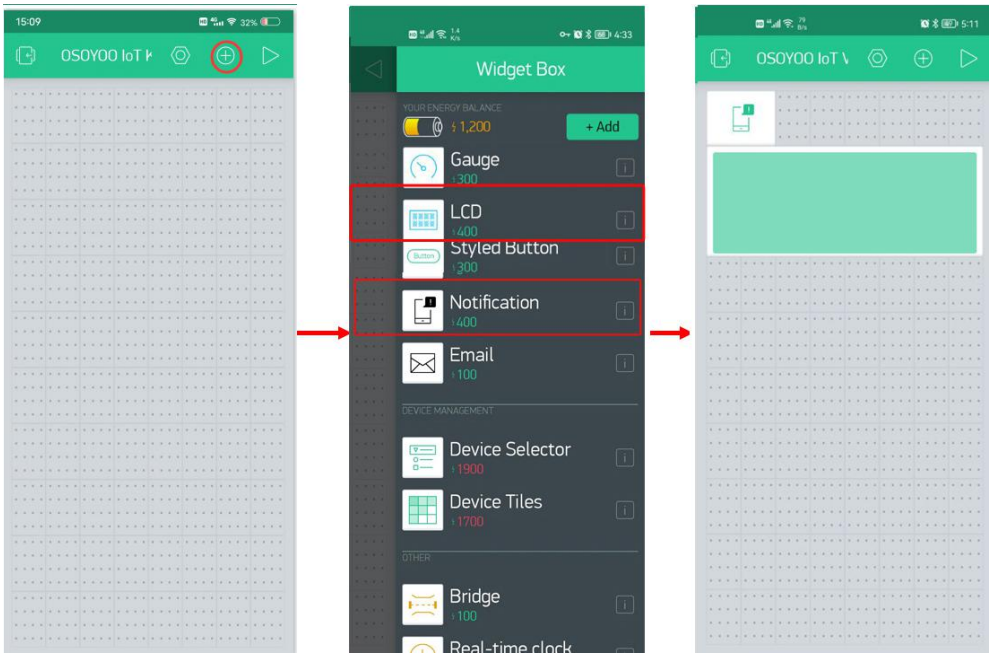


Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

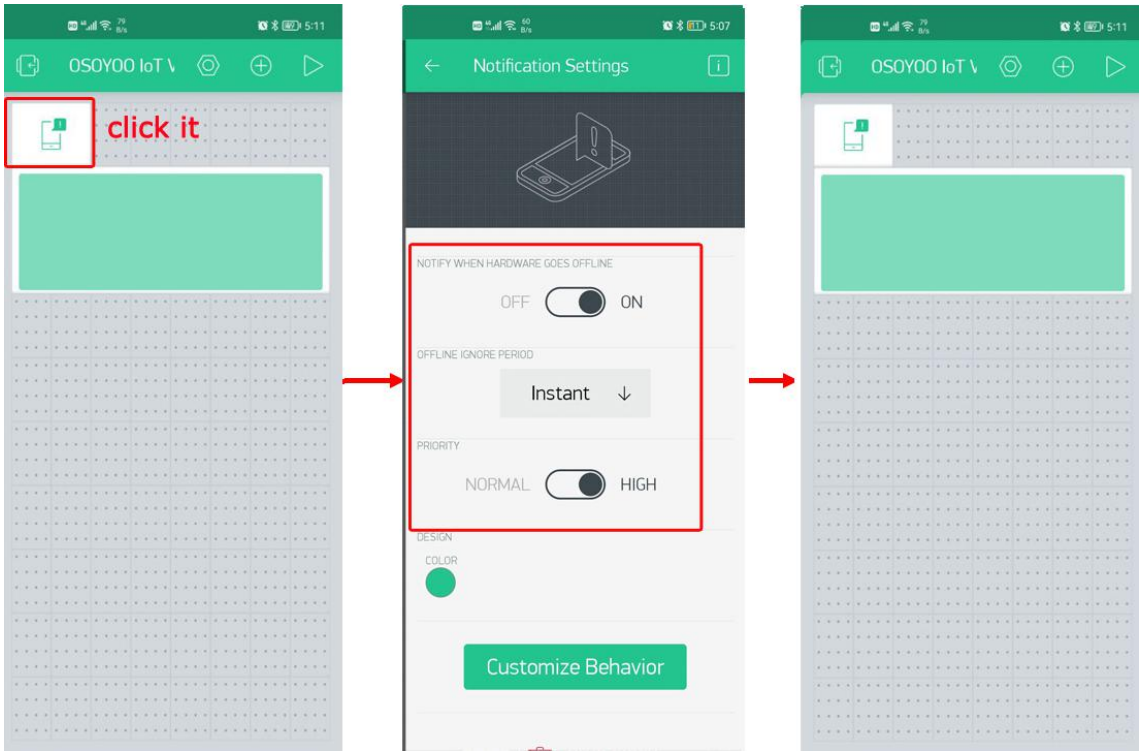
Open the Blynk APP , Follow the next operations:
Open your project page and press the "+" button to add the **Notification** and **LCD** Widget.



Notification widget:

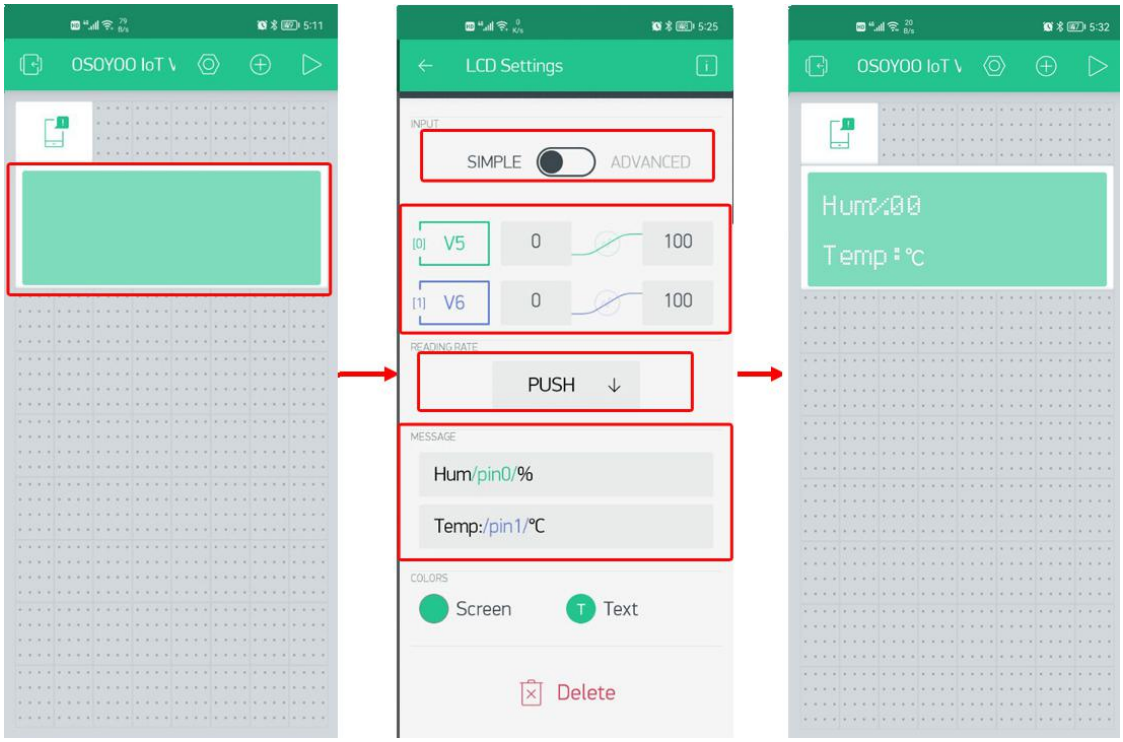
- Notify when hardware goes offline: ON
- Offline ignore period: Instant
- Priority: HIGH

Then set the Blynk notification to the highest authority in the phone settings.



LCD widget:

- INPUT: SIMPLE
- [0]-V5: 0~100
- [1]-V6: 0~100
- READING RATE: PUSH
- MESSAGE:[Hum: /pin0/%], [Temp: /pin1/℃]
- And you can keep other settings as default



After you finished all above operations, press the "▷ " button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.

HOW TO PLAY

With this system, we can know the temperature and humidity of the environment in real time through the Blynk APP, and the temperature and humidity of the environment will be displayed on the local display.

When the temperature is higher than 30 degrees Celsius, the phone will receive a push notification "The temperature is higher than 30°C! The fan is on!", the air circulation fan will automatically turn on and actively adjust the temperature. When the temperature is lower than the warning value, the fan will automatically shut down.

Look the video on Youtube :

https://www.youtube.com/embed/i-Px-_YA_jU

If your DHT11 sensor is disconnected or stops working for some reason, your local screen and your mobile phone will still receive a prompt message: "DHT11 Failed!" or "NO DHT11!"

If you find the device is offline, please check your code, wiring and app settings, and try again.

Tip: If someone finds that their wiring and code are correct, but the fan module does not rotate, please turn the fan blades by hand, and the fan will rotate slowly! To prevent users from being injured, we have designed the fan module for safety.

3-6 Outdoor security alarm

OBJECTIVE

In this tutorial, we will learn how to use OSOYOO IoT kit and Blynk APP to build an Outdoor security alarm device. An outdoor security alarm is a powerful deterrent to mischief makers, at the same time, you can also know the safety situation outside the door in time when you leave home.

After leaving home, we can turn on the warning device through Blynk. We can set the warning range through the software. If someone or something enters the monitoring area, the local device will sound an alarm, the red light will turn on, and our mobile phone will Receive push notification. Similarly, we can turn off this device with our mobile phone before going home.

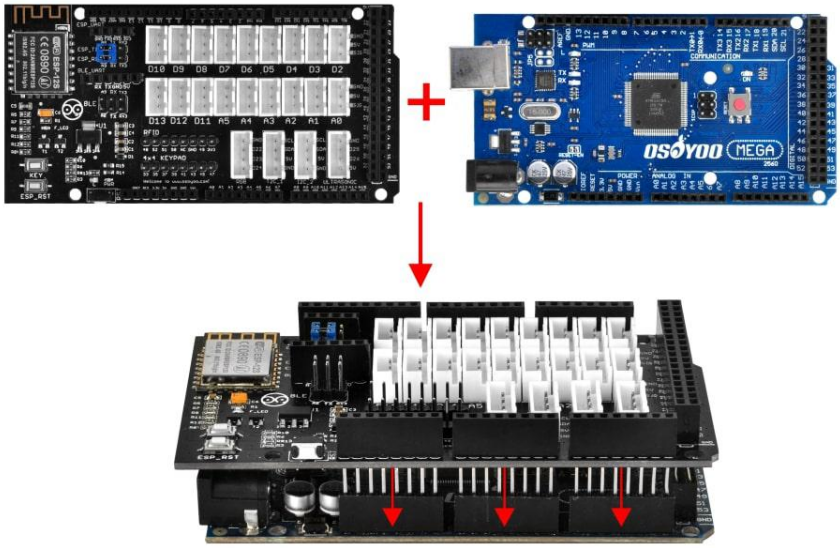
PARTS & DEVICES

HARDWARE
OSOYOO Mega2560 board x 1
OSOYOO IoT Shield x1
Active Buzzer Module x 1
Red LED Modulex 1
I2C 1602 LCD Display x 1
Ultrasonic Module x 1
USB Cable
Jumpers

SOFTWARE
Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
I2C library

HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



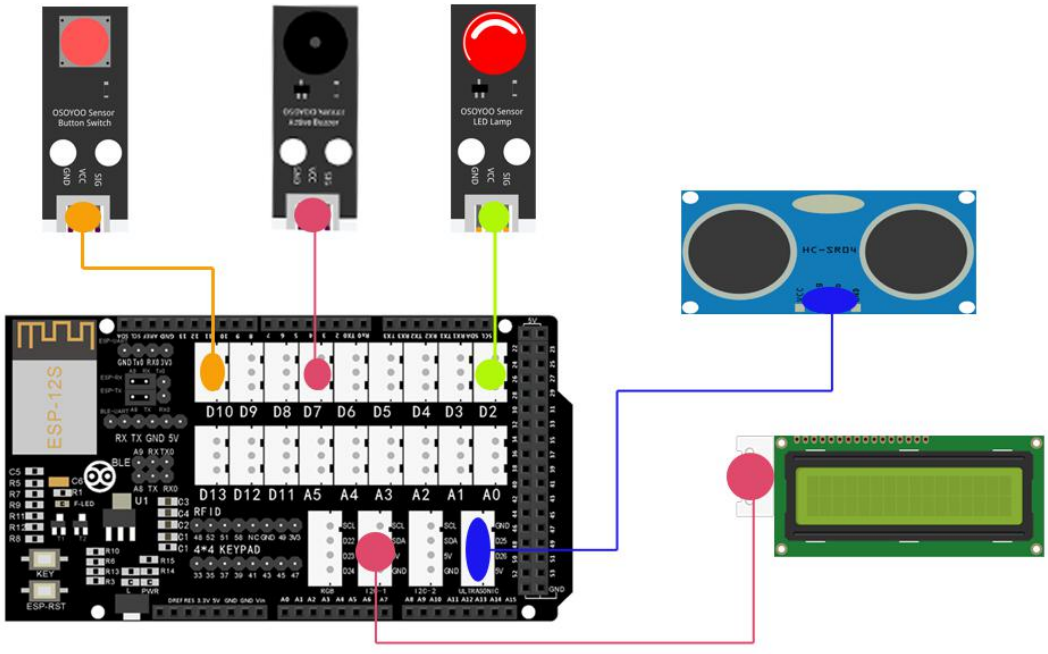
Build the circuit as below:

ultrasonic sensor-----Ultrasonic port

red LED-------D2

buzzer------D7

I2C 1602 LCD Display----I2C



HOW TO CODE

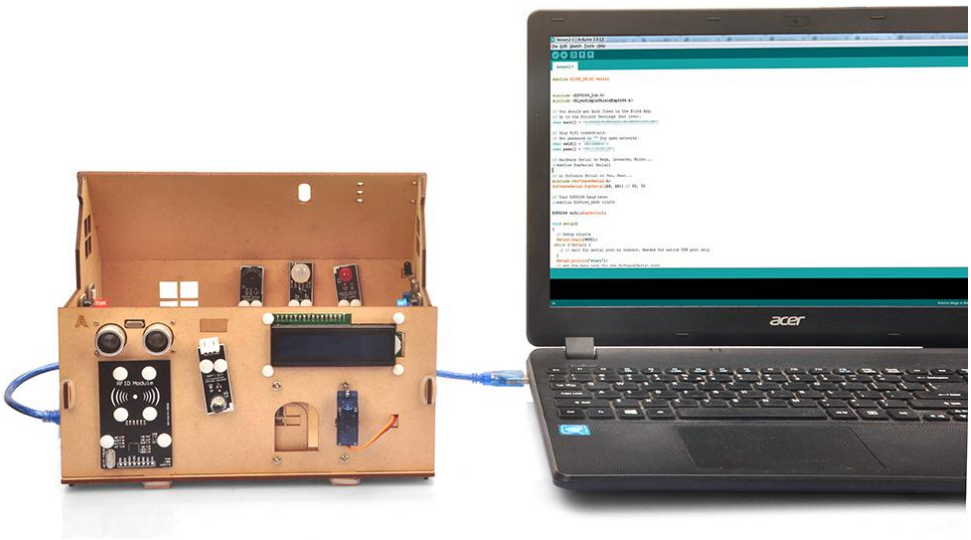
Step 1 Prerequisite

Step 2 Install latest Arduino IDE

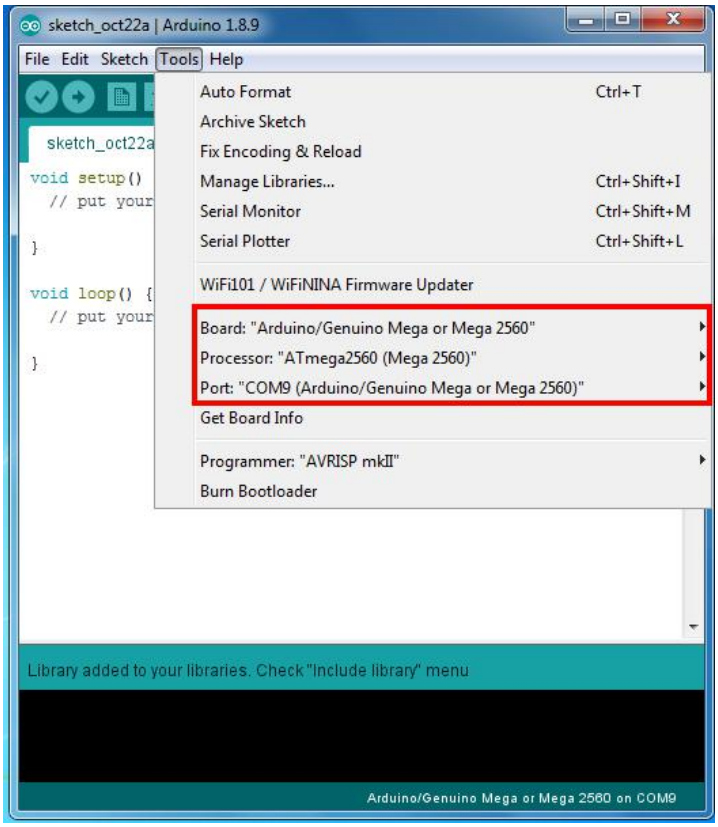
Step 3 Library Installation

Step 4 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson3-6.zip

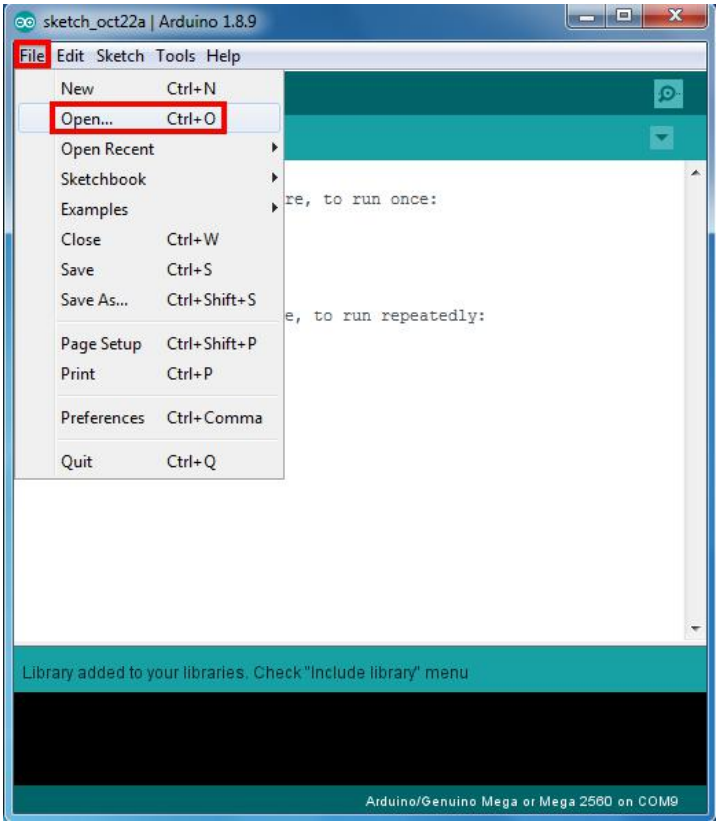
Step 5 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 6 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 7 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.



Note: In the sketch, find this line as following:

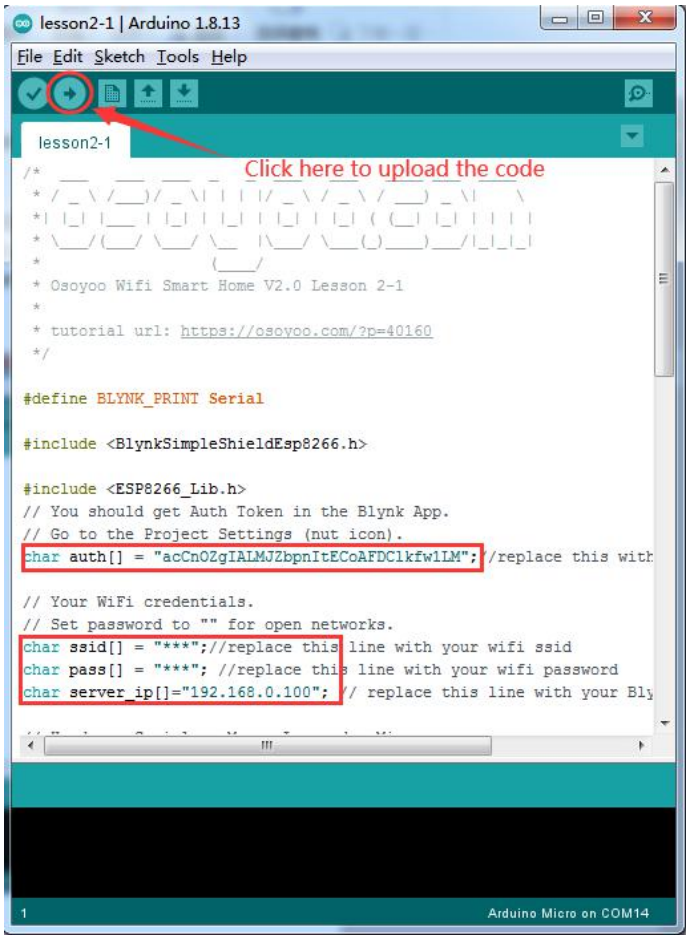
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

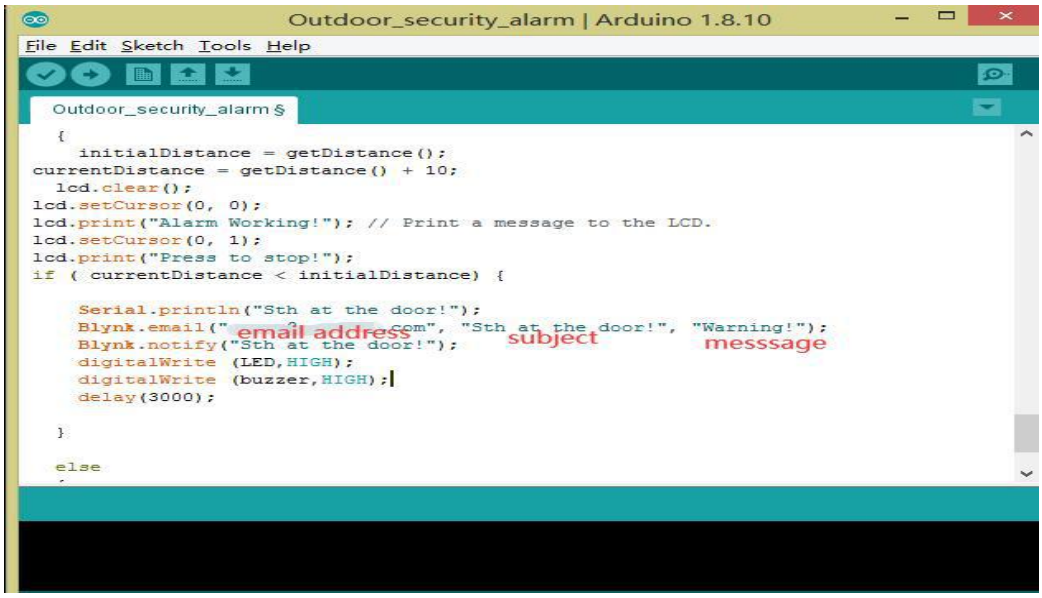
Then find these lines:

*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

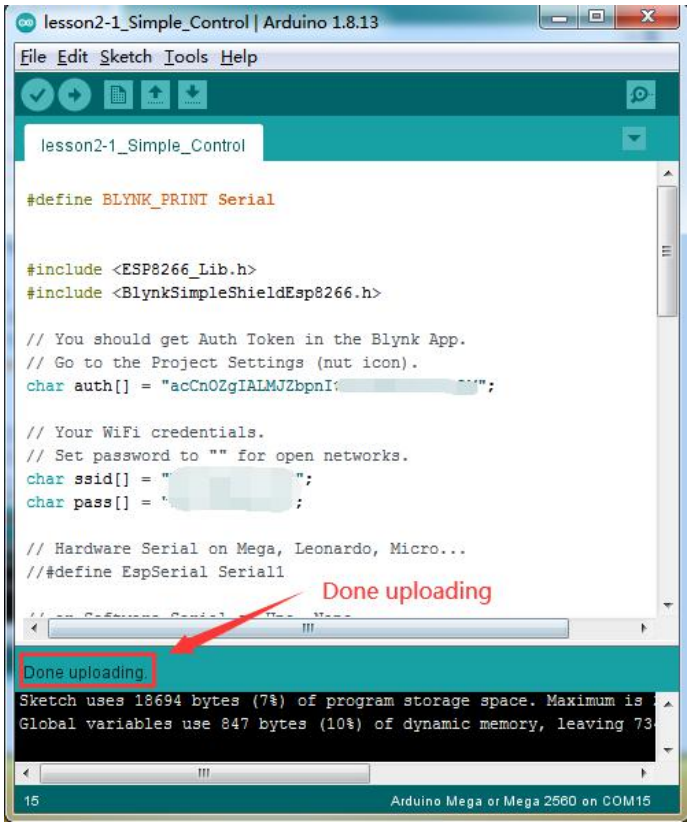
please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.



Then edit the email:



Finally Upload the sketch to the board. Wait until you see something like this: *Done uploading*

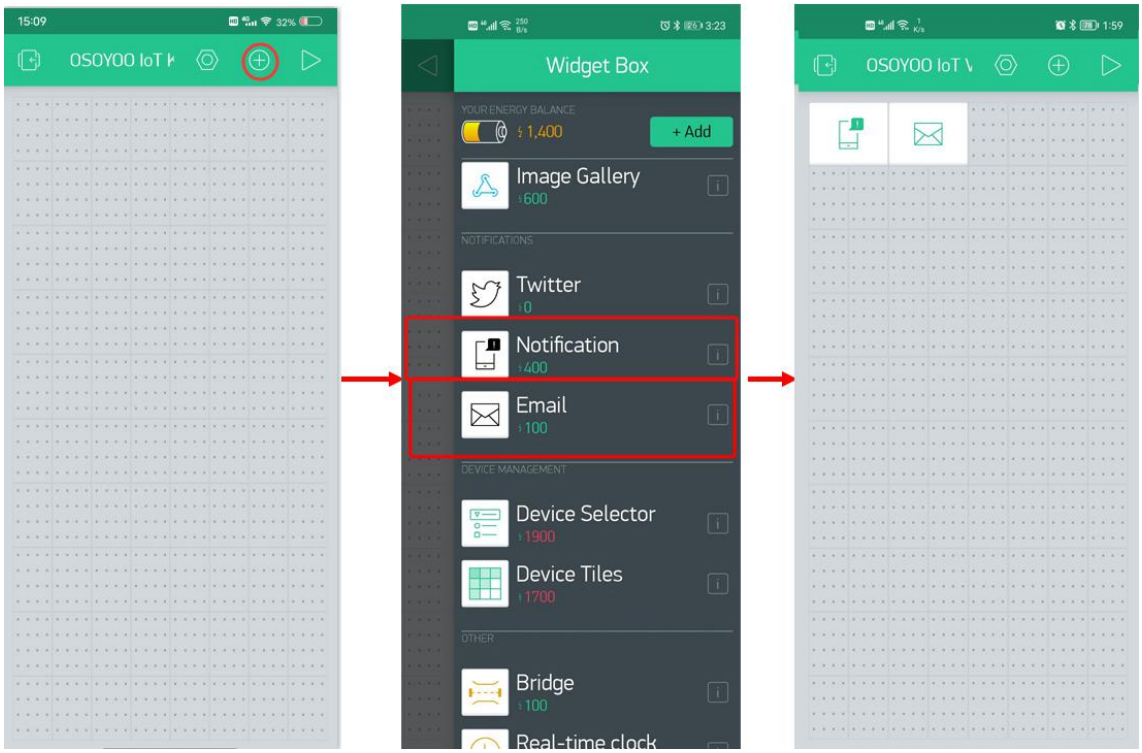


Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

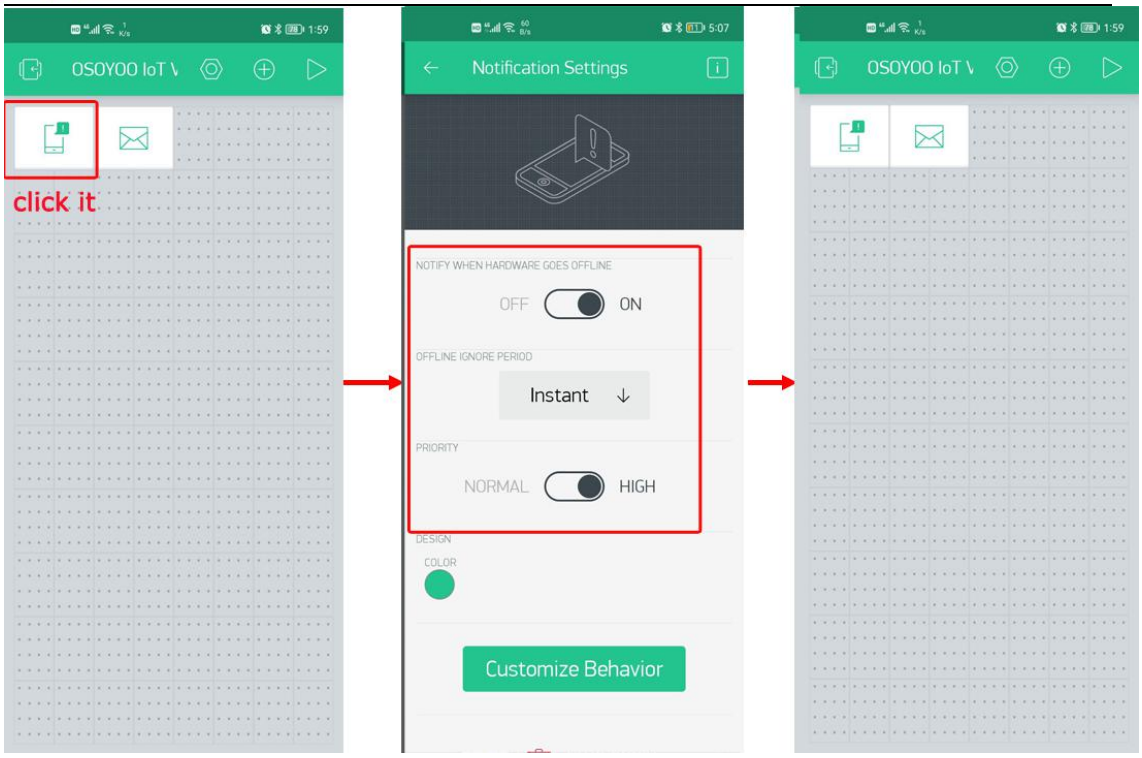
Open the Blynk APP , Follow the next operations:
Open your project page and press the “+” button to add the **Notification** and **email** Widget.



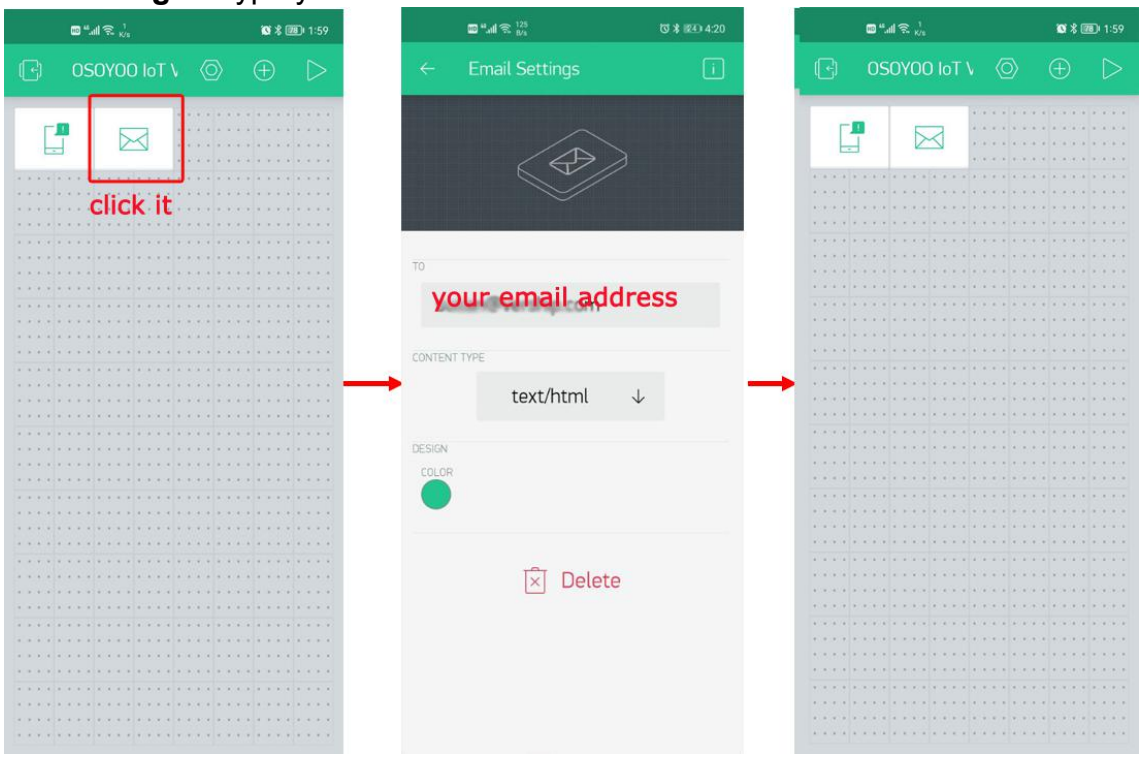
Notification widget:

- Notify when hardware goes offline: ON
- Offline ignore period: Instant
- Priority: HIGH

Then set the Blynk notification to the highest authority in the phone settings.



Email widget: Type your own email address there.



Then back to the project page. and press the "▷ " button to start your project.

HOW TO PLAY

First, you will see a prompt on the LCD screen to press the button to start the alarm. When your project is successfully connected to the network, press the button, and the LCD screen will display "Alarm Working!" and "Press to stop!". Or when an object passes in front of the ultrasonic distance sensor, both the mobile phone and the email will receive a prompt: "Sth at the door!", then the alarm will sound and the red light will be lit.If you want to turn off the alarm system, please press the button again and you will see "Alarm Stop! Press to start!" on the LCD screen.

Look the video on Youtube :

https://www.youtube.com/embed/ouCg17EKkMI

4 Smart Home IoT system

OBJECTIVE

Smart home is an intelligent place which can assist the residents to live independently and comfortably with the help of modern technology. In a smart home, all the mechanical and digital devices are connected with each other and can be controlled by computer through the internet.

In this capstone project, we will build the most complicated smart home system in this tutorial using all the parts inside OSOYOO Smart Home IoT kit. Through this system, we can complete remote data collection and remote control of multiple electronic devices through Blynk mobile APP.

PARTS & DEVICES

HARDWARE

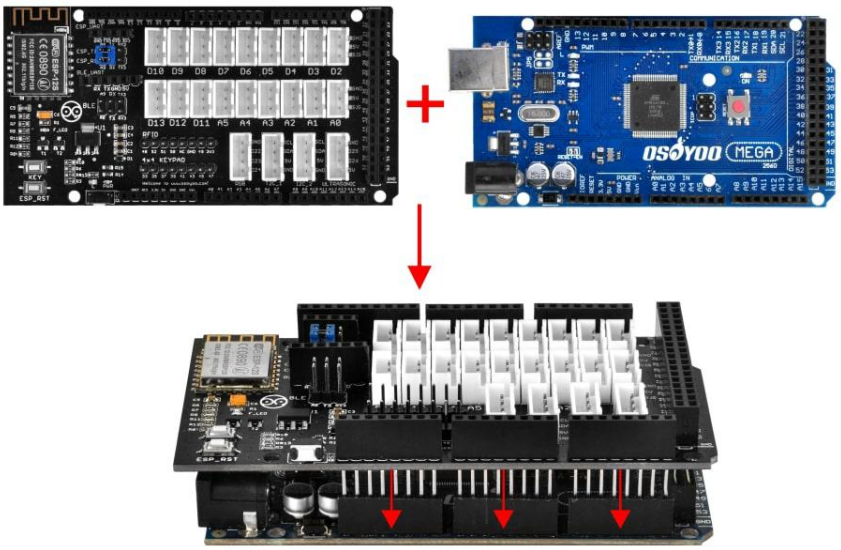
- Osoyoo Mega2560 board x 1
- OSOYOO IoT Shield x 1
- Active Buzzer Module x 1
- I2C 1602 LCD Display x 1
- USB Cable
- Flame sensor x 1
- HC-SR501 PIR sensor x 1
- DHT11 x 1
- Photosensitive sensor x 1
- SG90 180 degree Servo x 1
- RC522 RFID Module x 1
- HC-SR04 Ultrasonic Module x 1
- RGB Module x 1
- LED Module x 2
- LED strip light x 1
- Button Module x 2
- Fan Module x 1
- Jumpers
- PC x 1

SOFTWARE

Arduino IDE (version 1.6.4+)
Blynk Library
BlynkESP8266 Library
I2C library
RFID library
DHT11 library
Servo library

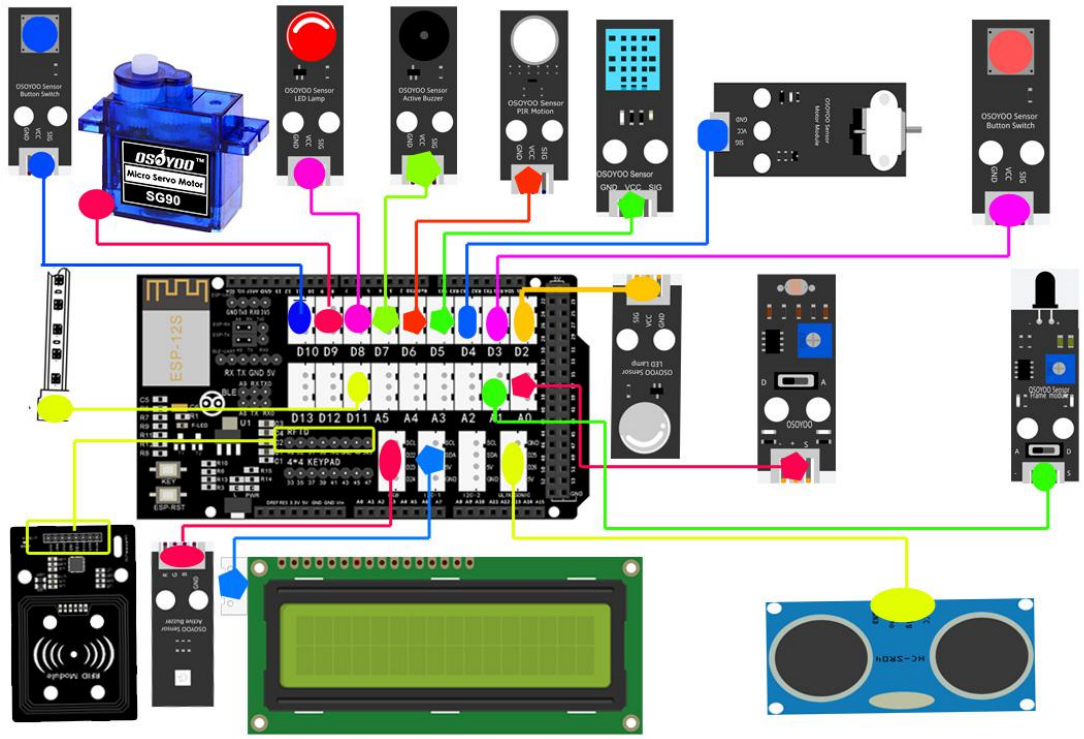
HOW TO MAKE

First please plug OSOYOO MEGA-IoT Extension Board into MEGA2560 board



Build the circuit as below:

Modules	Osoyoo Mega 2560
White LED Module	D2
Red Button Module	D3
Fan Module	D4
DHT11 Module	D5
PIR Motion Sensor	D6
Active Buzzer	D7
RED LED Module	D8
SG90 Servo	D9
Blue Button Module	D10
Led strip light	D11
Photosensitive sensor	A0
Flame sensor	A1
I2C 1602 LCD Display	I2C_1 Port
RGB Module	RGB Port
HC-SR04 Ultrasonic Module	Ultrasonic Port
RC522 RFID Module	RFID Port



Note：
Before you start this tutorial, make sure you know how to use these devices, or you need to learn the previous tutorial.

Because of the variety of sensors used in this tutorial, be careful, you can also use the code in the previous tutorial to detect whether the connection is correct.

HOW TO CODE

Step 1 Prerequisite

Step 2 Install latest Arduino IDE

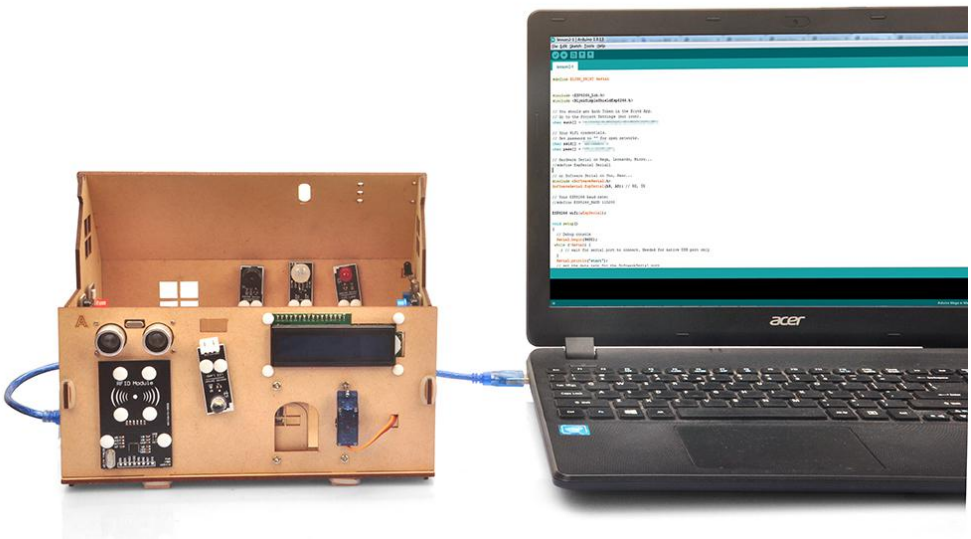
Step 3 Library Installation

Step 4 Servo Test (Step 4 -Step 9) If you adjust the servo before, please skip this step.

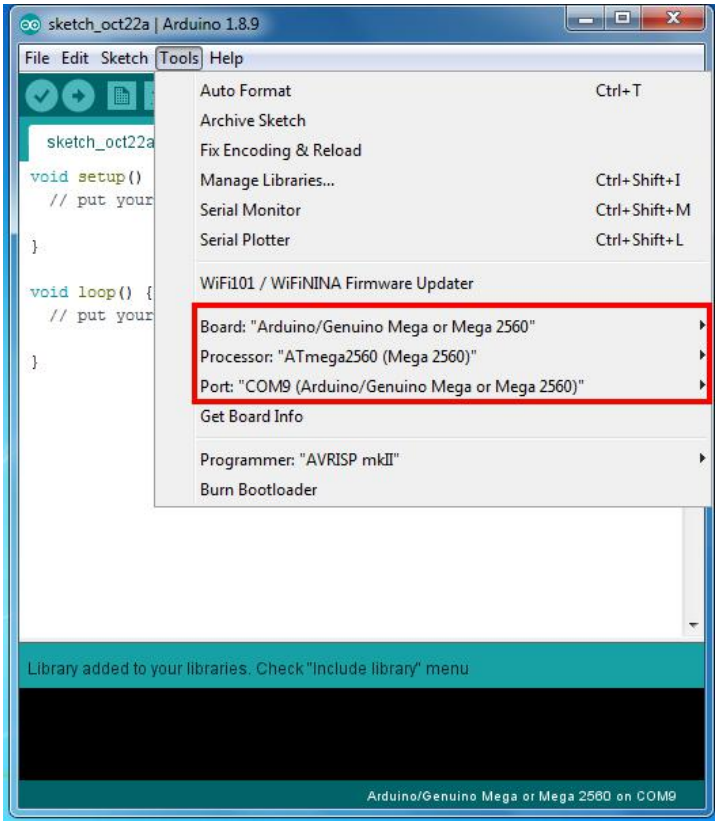
Step 5 After installing above library, please download the code from following link, unzip it:
https://osoyoo.com/driver/Smart_House_IoT_Learning_Kit_V2.0_code/lesson4.zip

Step 6 After above operations are completed, connect OSOYOO MEGA2560 Board to PC with USB cable.

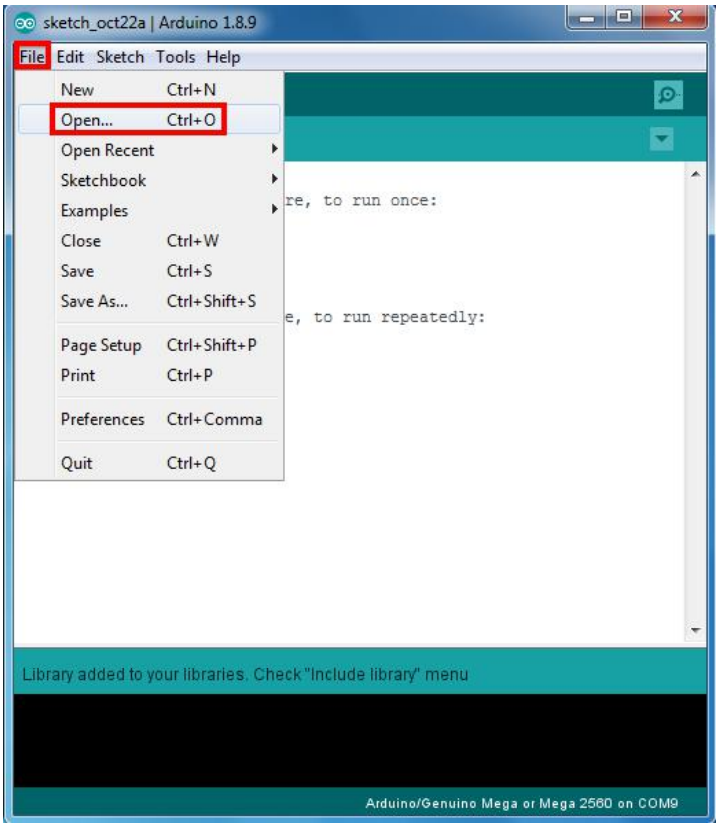
Notice: Shut off your battery or unplug your power adapter when upload sketch code to Arduino.



Step 7 Open Arduino IDE: Choose corresponding board type and port type for you project .



Step 8 Arduino IDE: Click file – Open, then choose code in the folder, load up the sketch onto your Arduino.

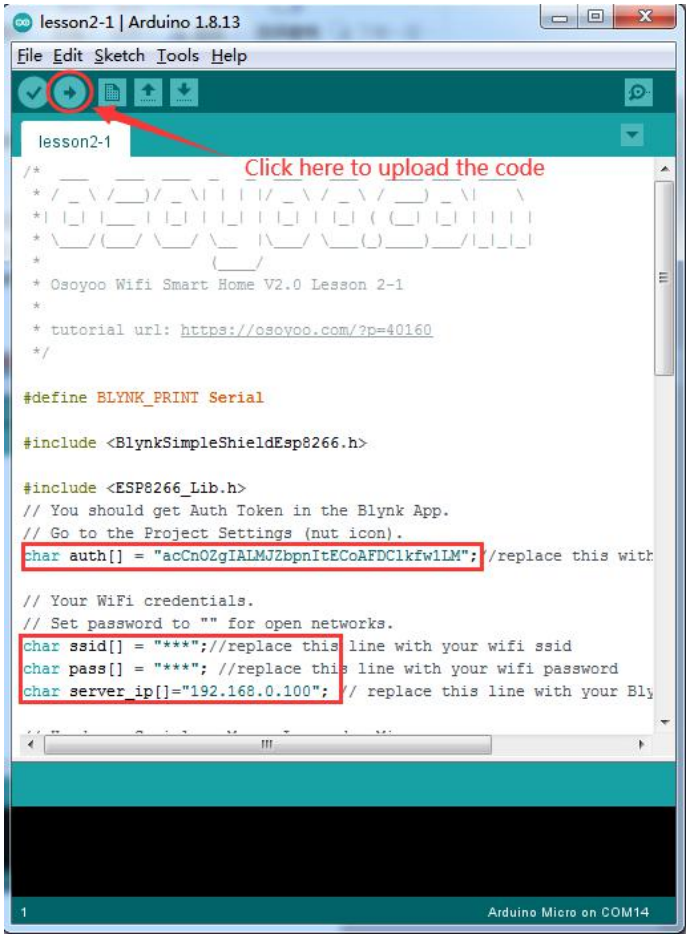


Note: In the sketch, find this line as following:
char auth[] =”0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-”;

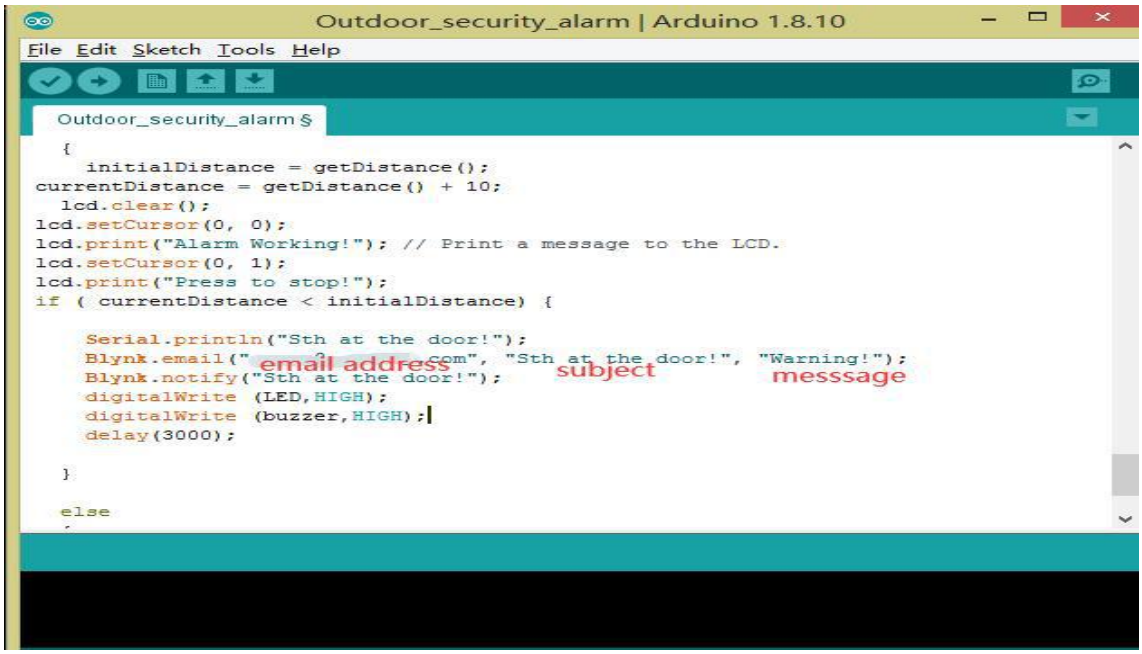
Replace **0ByWpc50cmOJ5O6OyHMaK4OSHkDvUeI-** with your local Blynk Token. If you don’t know how to get token, read Lesson 1-6.

Then find these lines:
*char ssid[] = "******"; // your network SSID (name)*
*char pass[] = "******"; // your network password*
char server_ip[]="192.168.1.81"; // replace this line with your Blynk Server IP address

please replace the ****** with your correct wifi SSID and password and replace your Blynk Server IP, otherwise your project can not connect to Internet.



Then edit the email:



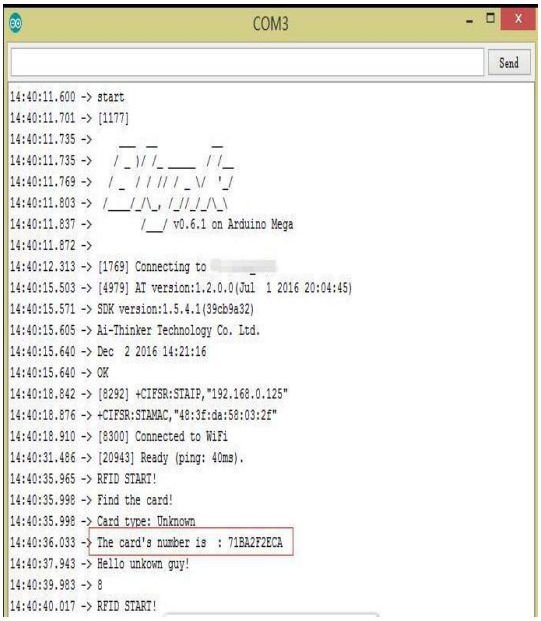
After completing the above steps , uploade the code, then open the IDE serial monitor, you can see that the system is initialized successfully, and then put

your RFID card in the RDID module recognition area, you will see your serial monitor will display your The card information is as follows:

Find the card!

Card type: Unknown

The card's number is : 71BA2F2ECA



Then find these lines:

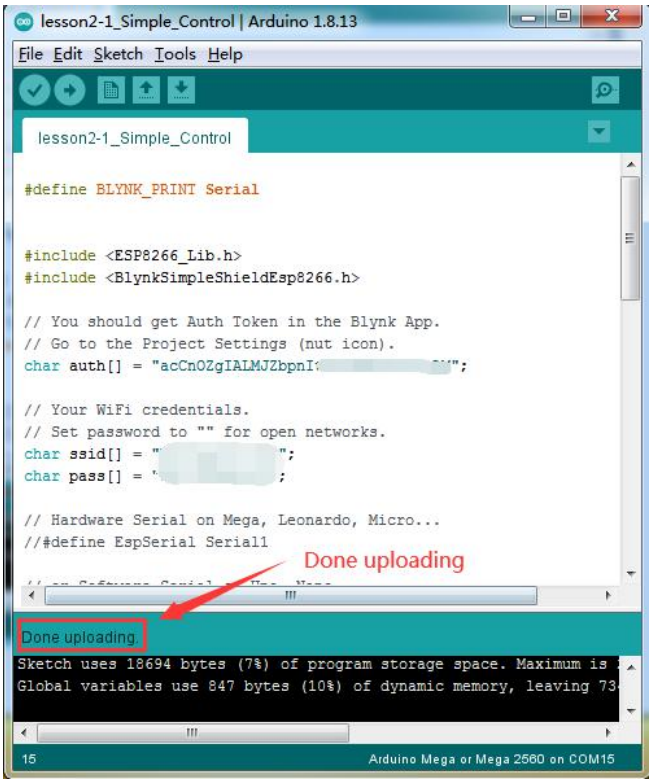
```
if( id[0]==0x32 && id[1]==0xDA && id[2]==0x94 && id[3]==0x10 )
```

And change the id data to your own card number, so the changed code is:

```
if( id[0]==0x71 && id[1]==0xBA && id[2]==0x2F && id[3]==0x2E )
```

Finally upload the sketch to the board again. Wait until you see something like this:

Done uploading



Now open your Serial Monitor, you will see Arduino is trying to talk to Blynk server. After 20 to 30 seconds, your Serial monitor will finally shows Arduino's IP address and a "**Ready (ping: 25ms)**" ending message.

Congratulations! You are all ready! Your hardware is now connected to the Blynk server.

Add a Widget in Blynk (legacy) APP

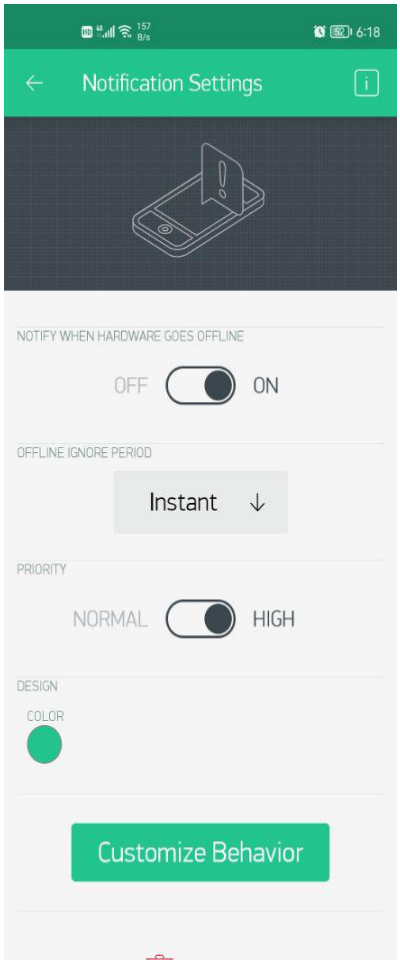
Open the Blynk APP , Follow the next operations:
Here we need to add the **Notification widget**, **LCD** widget, **Text Input** widget, **zeRGBa** widget, **Button** widgets, and the settings are as follows:

Notification widget:

- Notify when hardware goes offline: ON
- Offline ignore period: Instant

- Priority: HIGH

Then set the Blynk notification to the highest authority in the phone settings.



Note:

If you find the device is offline, please check your code, wiring and app settings, and try again.

If the mobile phone does not receive the push notification of Blynk APP or does not receive the email, please click the below settings:

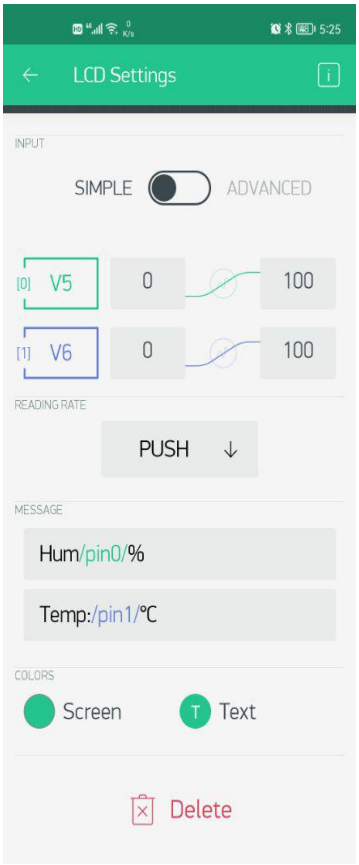
- **Blynk server or local server.** The local server may not push notifications
- **Blynk Library version.** Please make sure it is the latest version but not a Beta version
- **Blynk APP version.** Please make sure it is the latest version but not a Beta version
- **Priority.** High priority gives more chances that your message will be delivered without any delays. See detailed explanation here.

- **Mobile network settings.** Blynk uses the messaging service provided by Google. If your phone cannot access Google servers normally, you will not receive push notifications, but you can still use email notifications normally.

If the above does not solve your problem, please search for similar topics in the Blynk forum https://community.blynk.cc/, or
check http://docs.blynk.cc and http://help.blynk.cc.

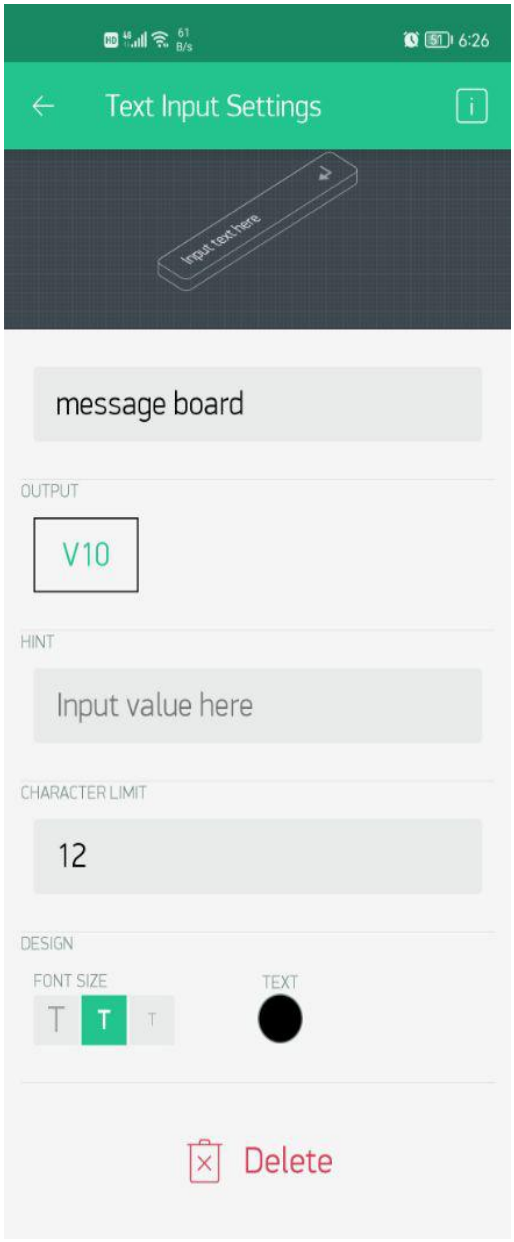
LCD widget:

- INPUT: SIMPLE
- [0] V5 (0~100)
- [1] V6 (0~100)
- READING RATE: PUSH
- MESSAGE [Hum: /pin0/%] [Temp:/pin1/℃]



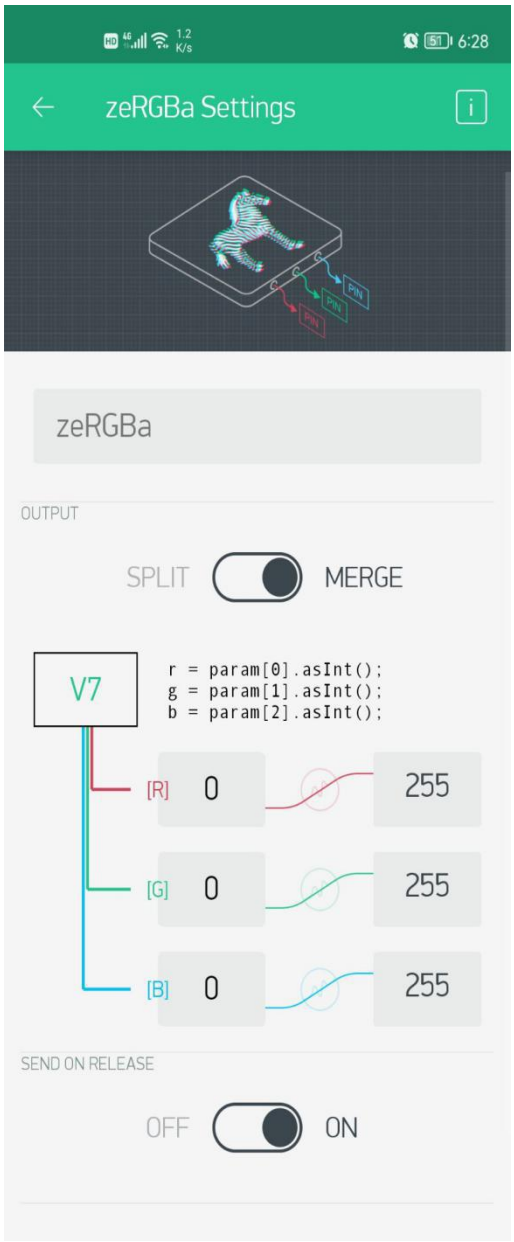
Text Input widget:

- Name： Message board
- OUTPUT: V10
- CHARACTER LIMIT: 12



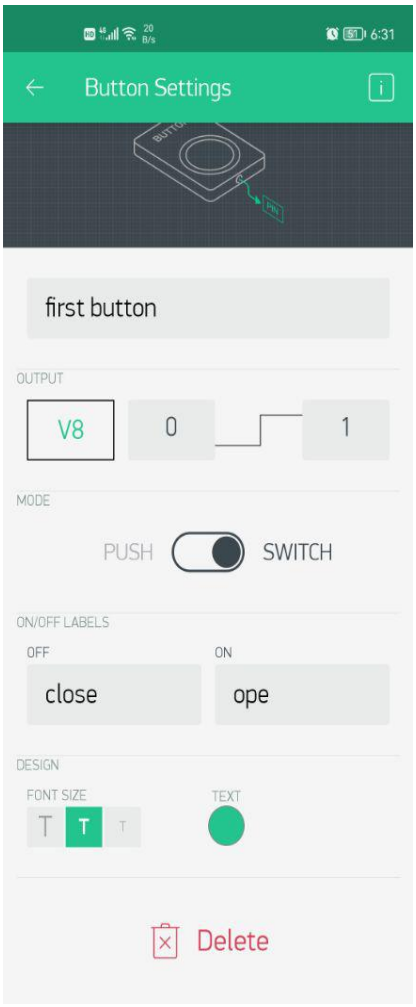
zeRGBa widget:

- OUTPUT: MERGE V7
- [R]: 0~255
- [G]: 0~255
- [B]: 0~255
- SENS ON RELEASE: ON



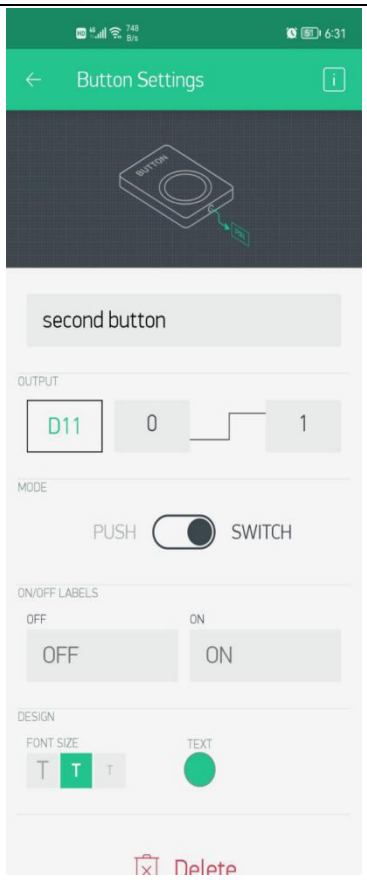
First Button widget:

- NAME: DOOR
- OUTPUT: V8 0~1
- MODE: SWITCH
- ON/OFF LABELS: OFF/CLOSE ON/OPEN

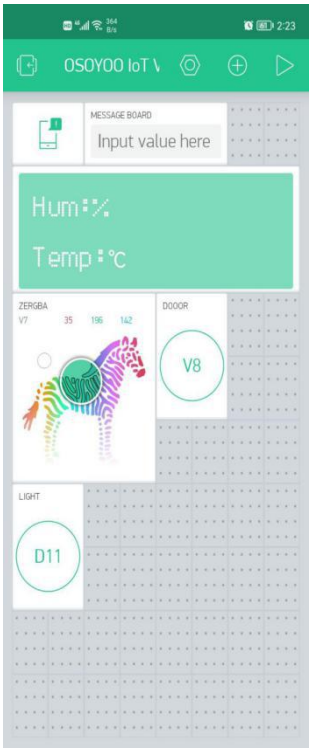


Second Button widget:

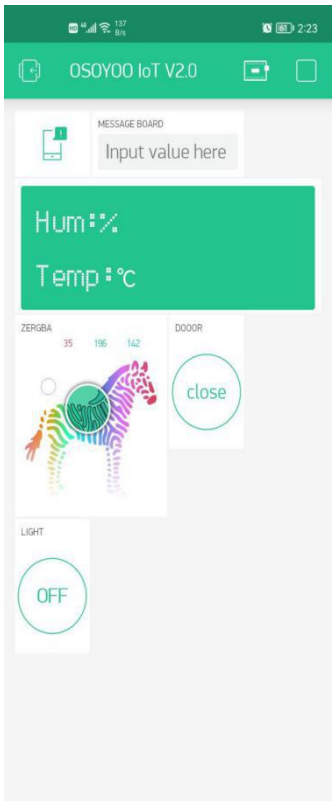
- NAME: Lights
- OUTPUT: D11 0~1
- MODE: SWITCH



After completed above operations, open the Blynk app and you will see as below:



After you finished all above operations, press the “▷ “ button. This will switch you from EDIT mode to PLAY mode where you can interact with the hardware.



HOW TO PLAY

Next, turn on the phone, we will see the real-time temperature and humidity data, and we can also see it on the I2C LCD display. If DHT11 is disconnected or fails, you will get a "DHT11 Failed!" prompt.

Moreover, when the room temperature exceeds 30 degrees Celsius, the fan module on the roof will automatically start to adjust the temperature, and the mobile phone will also receive a warning that the temperature is too high. When the temperature is lower than 30 degrees Celsius, the fan will stop working.

When we forget to take the key to go out, we can open the door through the BLYNK APP, click the DOOR button to open the door, the screen displays "Hello, guys!", click the button again to close the door, the screen displays "Close the door!"

If we bring a card or key fob, we can directly put the card in the sensing area of the RFID module. The system will recognize the user through the pre-set data. After the recognition is passed, the door will be opened and the screen will display something like "Hello Mary!" ""Welcome home!" welcome message, at the same time, our mobile phone will also receive the corresponding message push. After the door is opened, the door will automatically close after a while.

If someone wants to open the door with an unknown card, the door will not be opened, the screen will display "Unknow user!", and the phone will receive an alert push: "Unidentified user is ready to open the door!"

We often need different colors in different life scenes. We can easily adjust the light color through the RGB widget on the BLYNK APP. Press the Lights button, and the gorgeous light strip will be lit. We will often use it during festivals or celebrations. To turn off the light strip, we only need to press the button again. We always look forward to something different on special days.

Leave a small light for the family at night. The Red LED connected to port D8 will not light up when the light is sufficient during the day, but when the light is dim, the LED will be lighted. You can also adjust the sensitivity of the night light by modifying the code.

We can use the display screen as a remote electronic message board, open the Blynk APP, click the Text Input widget, edit what you want to say, click the send button, and then your message will be displayed on the screen at home.

Usually we attach great importance to home security. Here we have built a simple home smart security system, including environmental monitoring, human movement monitoring, safety distance monitoring outside the door, emergency help alarms, etc.

Ordinary fire alarms will only sound or emit lights when a dangerous situation is detected. As a more intelligent version, when our smart home system detects a fire, in addition to the most basic sound and light warning, it will also push the alarm information to the owner's mobile phone or the corresponding mailbox.

When we are alone at home, we need to pay more attention to protect ourselves. Press the button connected to D3 and we can turn on the security monitoring system outside the door. When a stranger or something approaches the door, an audible and visual alarm will be issued. The phone will also receive a prompt: "Sth at the door!". When it is unnecessary, we can turn off the system by pressing the D3 button again.

A motion detection system allows us to go out on vacation with greater peace of mind. If someone is active within the monitoring range, the system will issue an alarm and push the alarm information to the owner. To a large extent, it can deter those who have no good intentions.

When we encounter an emergency, we can press the emergency button connected to D10. When the button is pressed, the system will remotely send the help signal to the person being asked or send it to the corresponding mailbox. At a critical moment, a small button may help you out of trouble.

If you find the device is offline, please check your code, wiring and app settings, and try again.

Look the video on Youtube :

https://www.youtube.com/embed/3nSCMzKrf60

5.How to create new user account in private local Blynk legacy server

If you have completed our Blynk Smart Home IoT projects, you might have noticed that we only use one email ID **admin@blynk.cc** to do all the projects in Blynk Legacy APP and private Blynk Server.

The reason is that Blynk company will no longer to support Blynk Legacy APP(free version) to register new user account. Blynk company wants users to get their new Blynk IoT APP and use their Cloud server(monthly fee apply).

Fortunately Blynk legacy APP will still be available in Apple store and Google Play Store, we can use this legacy APP to do IoT projects on private local Blynk Server which is open source and free.

Blynk Legacy APP now only have one default user **admin@blynk.cc**.

However, in many cases , such as school class, we do need multiple blynk user accounts to allow students do individual projects separately.

Is there a solution to create multiple user accounts in same local Private Blynk Server?

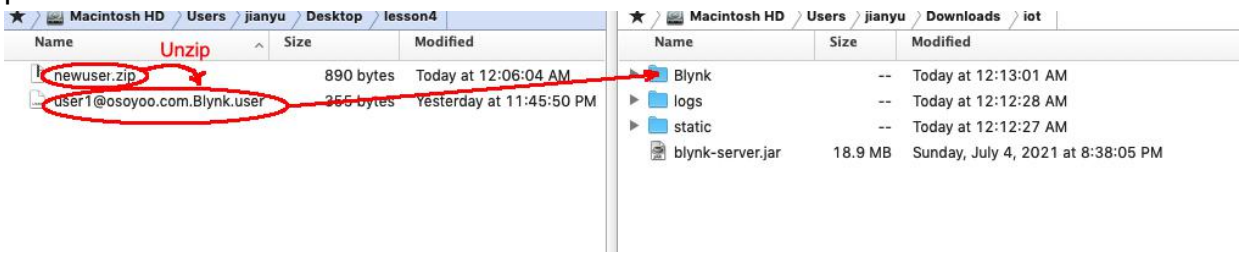
In this tutorial, we will tell you how to create new user accounts for Blynk legacy APP and private local Blynk Server.

Prerequisite:
If you don't know how to install local private Blynk server and how to use Blynk legacy APP, please read following article first:
https://osoyoo.com/2021/01/15/how-to-install-a-local-blynk-server-in-your-pc-and-get-a-local-blynk-token/

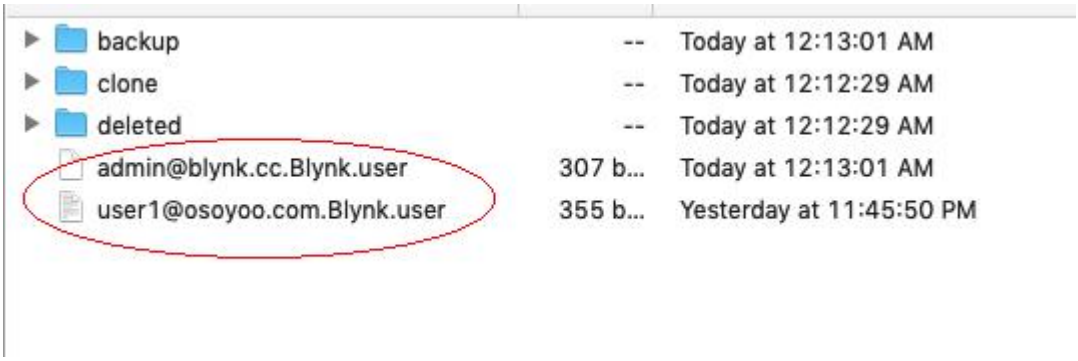
If you have already installed local private Blynk server in a PC and Blynk legacy APP in your cell phone, you can now start to create new users now.

Step 1)
Download a file from https://osoyoo.com/driver/blynk/newuser.zip
Unzip the file, you will see a file **user1@osoyoo.com.Blynk.user** , this file containers all data for a new user id **user1@osoyoo.com** , please copy this file to Blynk folder which is under the same parent directory of blynk-server.jar

file from which you start local Blynk server with **java** command. See following picture:



Now enter Blynk folder, you will see two user files : **admin@blynk.cc.Blynk.user** and **user1@osoyoo.com.Blynk.user** .



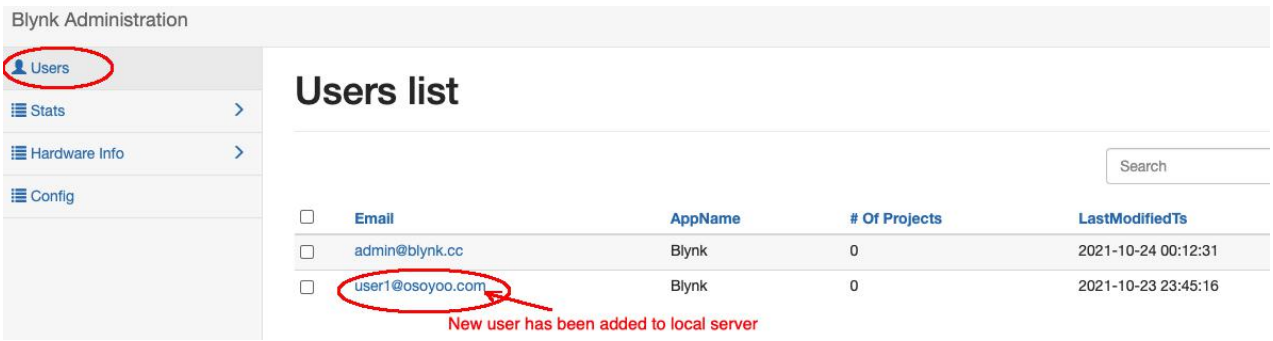
If you use a text editor to open above two files, you will find they are actually some json data files.

Now in cmd terminal , please start the local Blynk server by running following command:

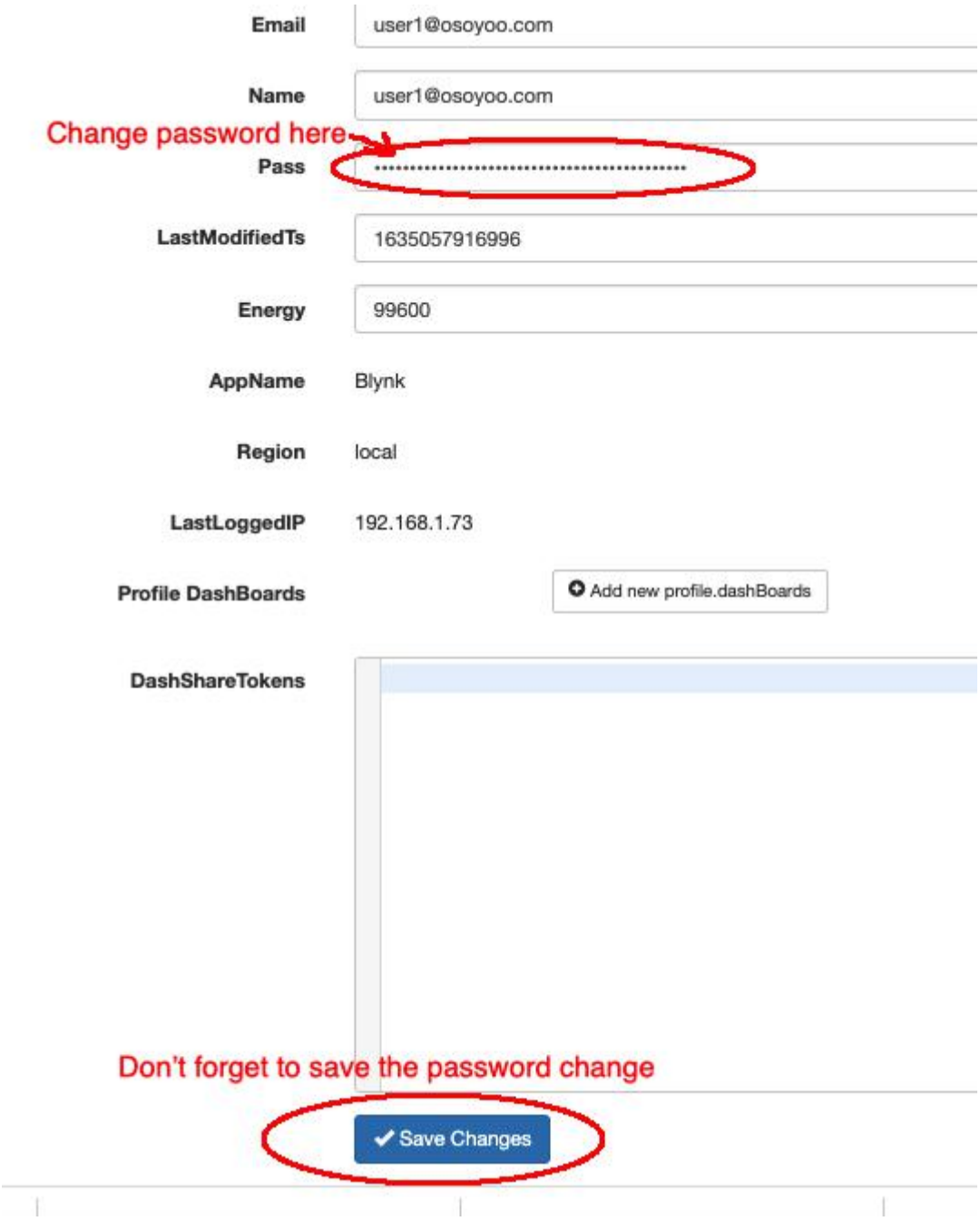
```
java -jar ./blynk-server.jar -dataFolder ./Blynk
```

then visit the local Blynk server admin dashboard url:
https://your_server_ip:9443/admin
login email is **admin@blynk.cc** , login password is **admin**

Now click Users, you will see a new Email id **user1@osoyoo.com** has been added into local Blynk dashboard. See following picture:



Click the Email ID **user1@osoyoo.com**, you can change its default password (default password is 123456).

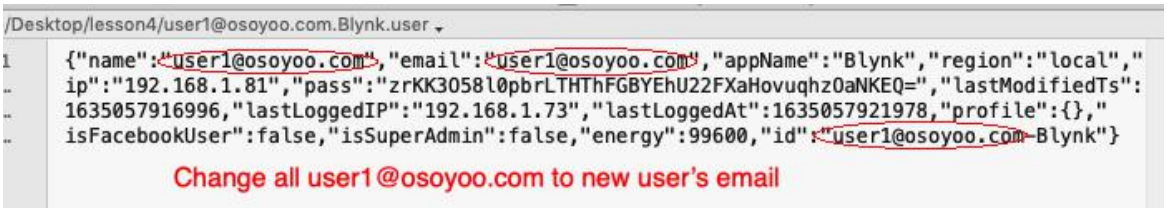


You have already successfully added one new email id in local Blynk!
Now you can use new email ID **user1@osoyoo.com** to login to Blynk legacy APP and Blynk local server admin dashboard.

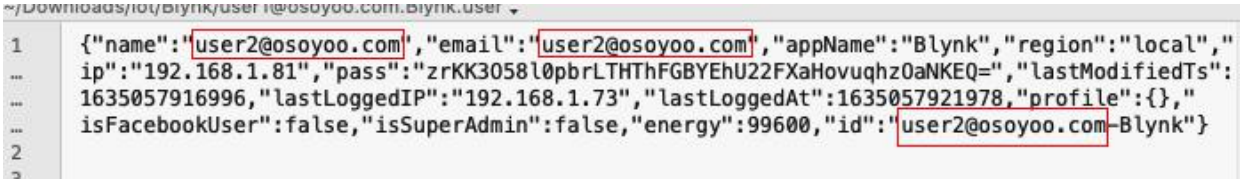
What if I want to add another user (email) to my local Blynk Server?
It is very easy! For example , you want to add another user email called **user2@osoyoo.com**

Step 1) Copy the file **user1@osoyoo.com.Blynk.user** to a new file and rename new file to **user2@osoyoo.com.Blynk.user**

Step 2) Use a text editor to open **user2@osoyoo.com.Blynk.user** json text file and replace all the string of **user1@osoyoo.com** to **user2@osoyoo.com** then save the file.
Old file is following:



After changing, new file is as following:



Move this **user2@osoyoo.com.Blynk.user** file to Blynk folder,

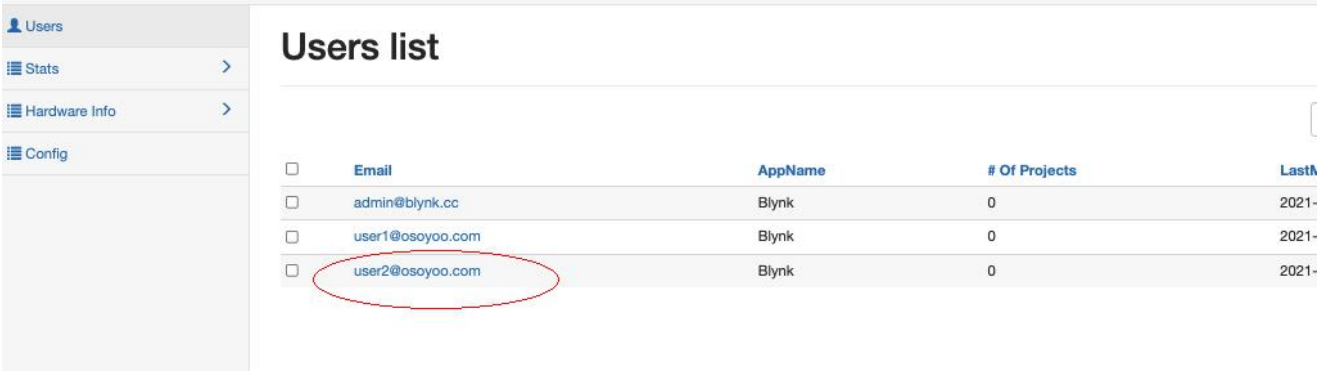
Now user ctrl C to stop Blynk and restart it with following command:

```
java -jar ./blynk-server.jar -dataFolder ./Blynk
```

In your browser, visit admin dashboard url: https://your_server_ip:9443/admin

login email is **admin@blynk.cc** , login password is **admin**

Now click Users, you will see a new Email id **user2@osoyoo.com** has been added into local Blynk dashboard. See following picture:



You can click to Email **user2@osoyoo.com and** change its password.
Now you can use Email id **user2@osoyoo.com** to login to Blynk legacy APP and make new project!