CEC Continuing Education Center

DesignNews

Developing WiFi IoT ESP8266-Arduino Based Devices

# DAY 2 : Remote Control LED

Sponsored by

Digi-Key ELECTRONICS

informa markets

# Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

# Don Wilcher

Visit ‘Lecturer Profile’ in your console for more details.

Course Kit:
Osoyoo ESP8266 Arduino IoT Kit

# Agenda:

- What is an Internet-based Control System (ICS)?
- Functional Modelling of an ICS
- User Datagram Protocol Basics
- Osoyoo ESP8266 Arduino Kit Overview
- Lab: Remote LED

# Internet of Things :

" The Internet of Things (IoT) is a concept in which the virtual world of information technology integrates seamlessly with the real world of thing." (Uckelman, Harnson & Michahelles, 2011, p.2).

# What is an Internet-based Control System?

- Internet-based Control System (ICS) are:
  a) networked control systems
  b) control systems that communicate with
    i. sensors
    ii. actuators
    iii. other smart technologies
  c) other control systems over a medium
    i. wired
    ii. wireless
- Unlike local control, a remote control allows an operator to control an object at another location.
- a) remote control was previously called teleoperators
- b) distance between the controller and the controlled object can vary widely

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# What is an Internet-based Control System?. . .

All remote control systems have the following features.

a) An operator interface that the operator uses to command the systems.

b) A local control device that performs the operator's commanded actions at the remote site.

c) A communication channel between the remote and local sites.

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# Question 1

**ICS is the abbreviation for what term?**

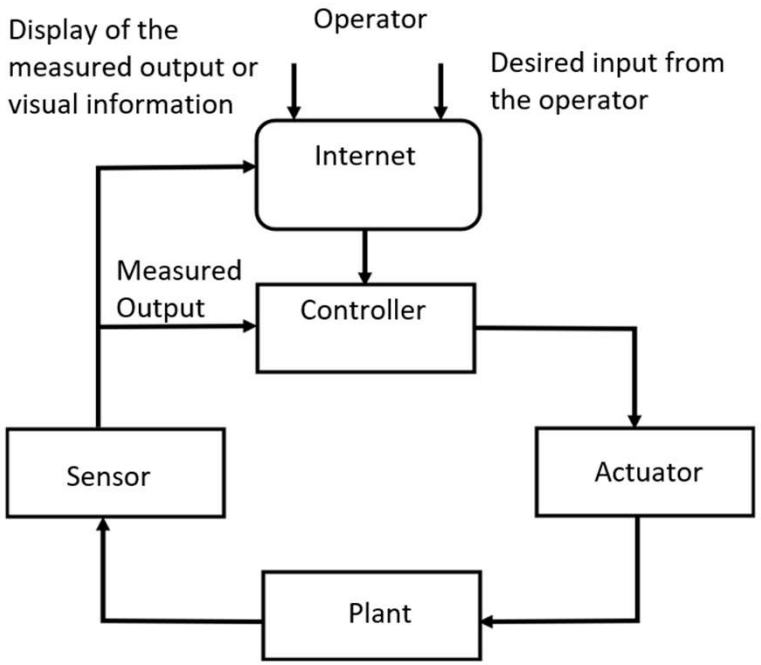
**a) integrated circuits**

**b) internet-based central system**

**c) internet-based control system**

# What is an Internet-based Control System?. . .

Control structure with the operator located remotely



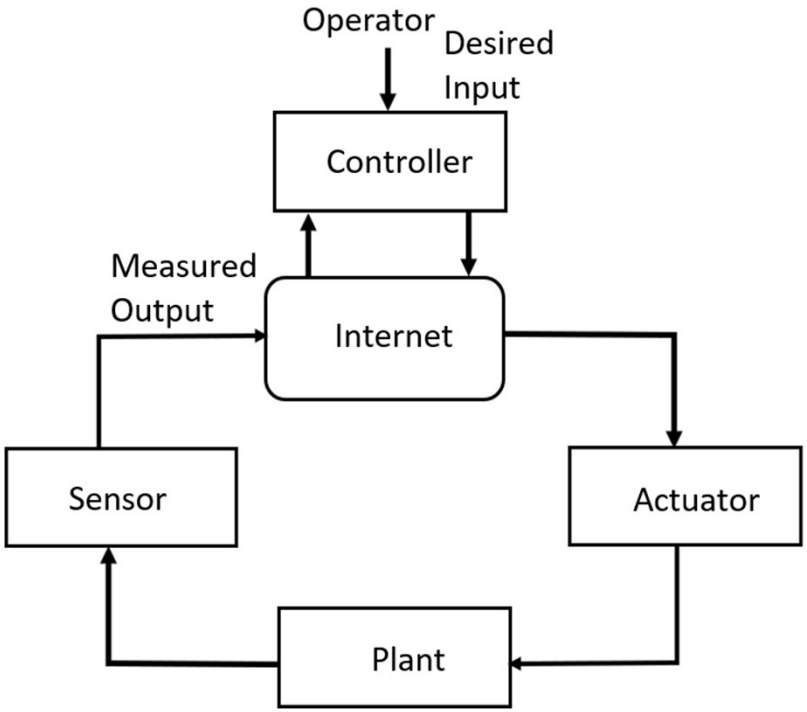

Notes:

a) Plant: The combination of an actuator and a process.
b) Inputs and output

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# What is an Internet-based Control System?. . .

Control structure with the controller located remotely




Notes:

a) Plant: The combination of an actuator and a process.
b) Inputs and output

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# Functional Modelling of an ICS

Functional Modelling consists of

a) making a specification of a control system as expected by the user.

b) expressing the control system's operation.

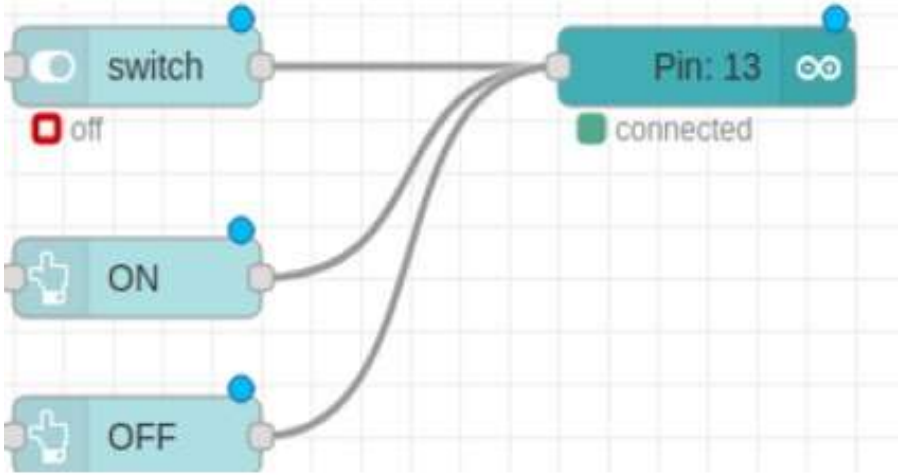
c) of a description of what the process will control.

d) defining the user needs using a Data Flow Diagram (DFD).

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# Functional Modelling of an ICS. . .

A Node RED Data Flow Diagram example:




button node: ON configuration

# Functional Modelling of an ICS. . .

Functional Modelling consists of

a) making a specification of a control system as expected by the user.

b) expressing the control system's operation.

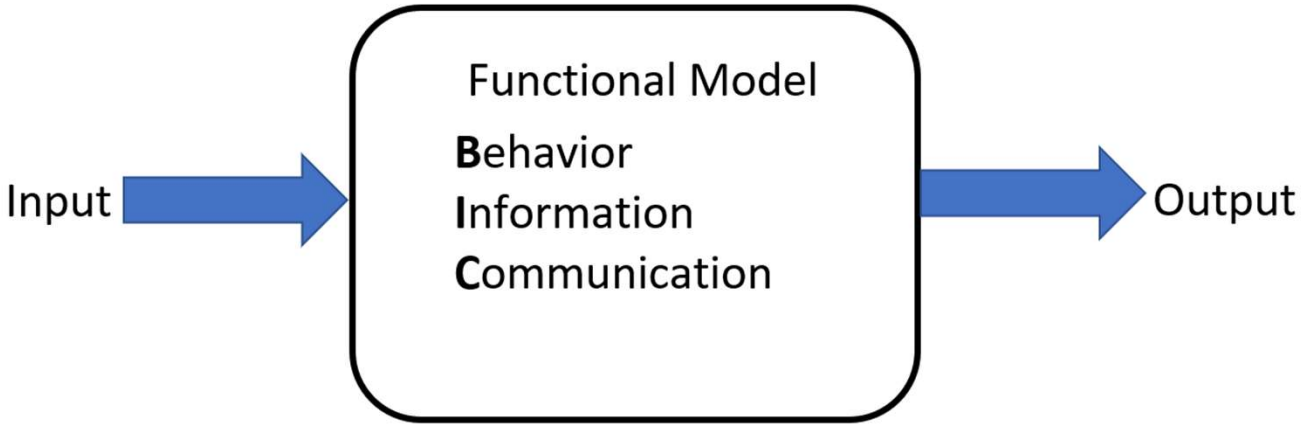
c) of a description of what the process will control.

d) defining the user needs using a Data Flow Diagram (DFD).

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# Functional Modelling of an ICS. . .

A DFD can be represented using a BIC systems functional model.




Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

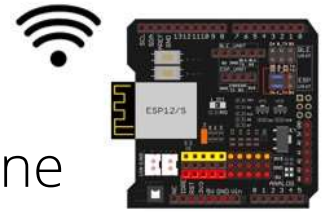
# Question 2

**Functional Modeling consists of defining the user need using a datagram.**

**a) True**

**b) False**

# Functional Modelling of an ICS. . .



B – defines the behavior of the function. Modeled by a state machine if-then or case statements.
I – defines the information produced by or used by the control system
C – defines the communication scheme. Modeled by data flows connecting the function with other system functions.

These triplets define the functional structure of the specific control system. Note: Functional Modelling may be considered the hardware and-or software requirements of the control systems.

Source: Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer.

# Functional Modelling of an ICS. . .

Functional Model of
Remote -Control LED
Lab project

**Input:Wireless Commands**

**Behavior: switch-case statement**

```
switch (c)    //serial control instructions
{

  case 'F': digitalWrite(ledPin, HIGH) ;break; //TURN ON LED
  case 'G':digitalWrite(ledPin, LOW) ;break; //TURN OFF LED

  default:break;
}
```

**Information:UDPpackets**

```
void loop() {
  int packetSize = Udp.parsePacket();
  if (packetSize) {                              // if you get a client,
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }
```

**Communication:WiFi**

```
#include "WiFiEsp.h"
int LED = 13;
```

**Output:LED**

# Functional Modelling of an ICS. . .

Functional Model for Remote-Control LED Lab project is C++ Software-based

## Behavior: switch-case statement

```
switch (c)    //serial control instructions
{

  case 'F': digitalWrite(ledPin, HIGH) ;break; //TURN ON LED
  case 'G':digitalWrite(ledPin, LOW) ;break; //TURN OFF LED

  default:break;
}
```

## Information:UDPpackets

```
void loop() {
  int packetSize = Udp.parsePacket();
  if (packetSize) {                              // if you get a client,
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }
```

## Communication:WiFi

```
#include "WiFiEsp.h"
int LED = 13;
```

# User Datagram Protocol Basics

- What is the User Datagram Protocol(UDP)?
  a) protocol used for communication throughout the internet.
  b) It is specifically chosen for time-sensitive applications like gaming playing videos, or wireless control.

- What is the benefit of using UDP?
  Results in speedier communication because it does not spend time forming a firm connection with the destination before transferring the data.

  Note: Because establishing the connection takes time, eliminating this step results in faster data transfer speeds.

# User Datagram Protocol Basics

What is the disadvantage of User Datagram Protocol(UDP)?

a) UDP can also cause data packets to get lost as they go from the source to the destination.

b) It can also make it relatively easy for a hacker to execute a distributed denial-of-service (DDoS) attack.

Note: A DDoS is a malicious attempt to disrupt the normal traffic of a targeted service or network. A DDoS can be achieved by overwhelming the target or its surrounding infrastructure with a flood of internet traffic.

# User Datagram Protocol Basics

The operation of a User Datagram Protocol

a) With no handshaking mode required, data packets or datagrams will continuously be sent from the transmitter to the receiver.
b) Nothing to identify the data is sent in order.
c) There is a checksum to check for data integrity.
d) Therefore, the data may be received or not.
e) Based on inconsistent sent/receive of data, the final signal may be glitchy.

Note: The straightforward request/response communication of relatively small amounts of data, eliminating concerns regarding controlling errors or the flow of the packets.

No concerns when used in remote control applications!!

# User Datagram Protocol Basics. . .

UDP being applied in the Remote-Control LED Functional Model:

Information:UDPpackets

```
void loop() {
  int packetSize = Udp.parsePacket();
  if (packetSize) {                              // if you get a client,
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    int len = Udp.read(packetBuffer, 255);
    if (len > 0) {
      packetBuffer[len] = 0;
    }
```

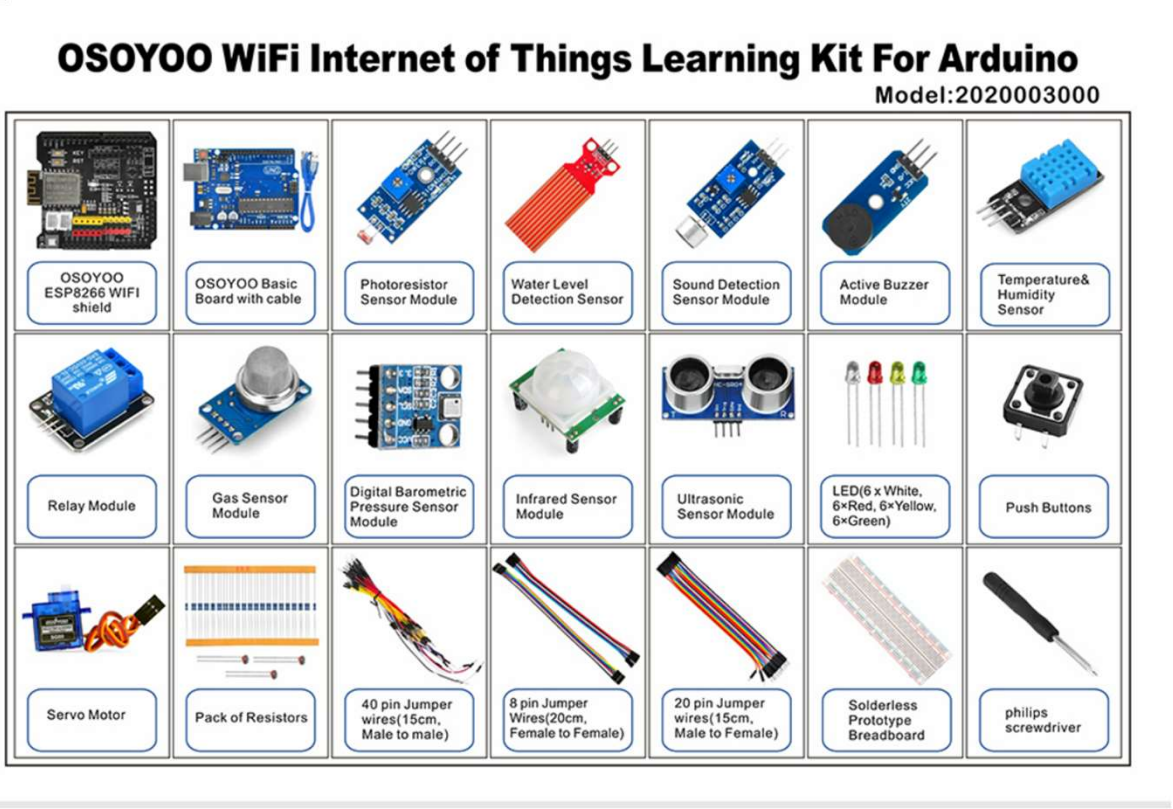
# Question 3

**In reviewing slide 23, what code of instruction is used to assign udp.parsePacket(); ?**

**a) int sizepacket = udp.parsePacket();**
**b) int sizepackets = udp.parsePacket();**
**c) int packetsize = udp.parsePacket();**

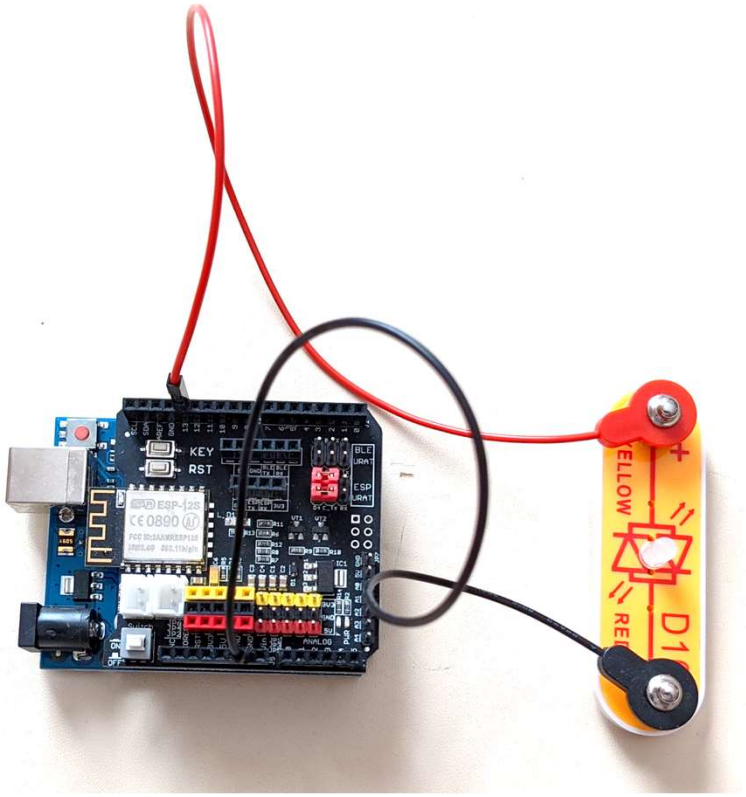
# Osoyoo ESP8266 Arduino Kit Overview

# Osoyoo ESP8266 Arduino Kit Overview




https://osoyoo.com/2020/05/30/wifi-iot-learning-kit-for-arduino/

# Lab: Remote Control LED

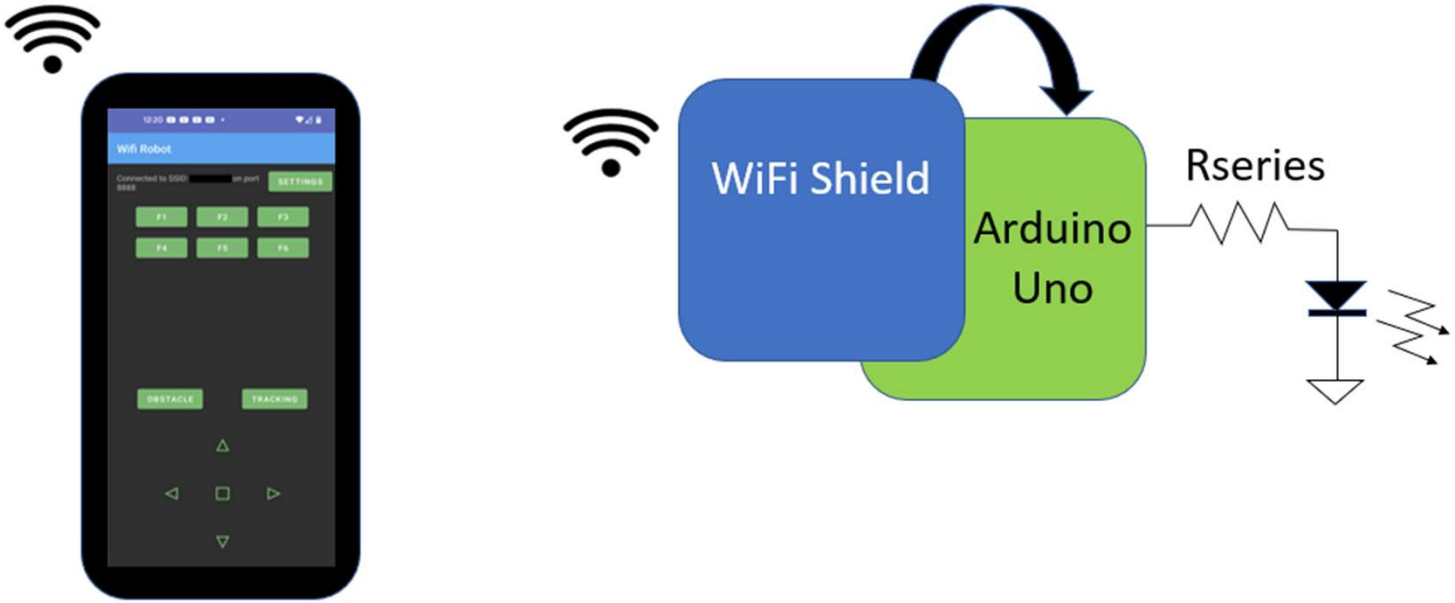
# Lab: Remote Control LED . . .

**Learning Objectives:**

- You will learn how to use a WiFi Shield with an Arduino Compatible.
- You will learn how to use an Arduino Compatible as a wireless controller.
- You will learn how to use an off-the-shelf mobile app to transform a smartphone into a remote-control unit.
- You will learn how to modify C++ code to remap ON/OFF keys of the mobile app.

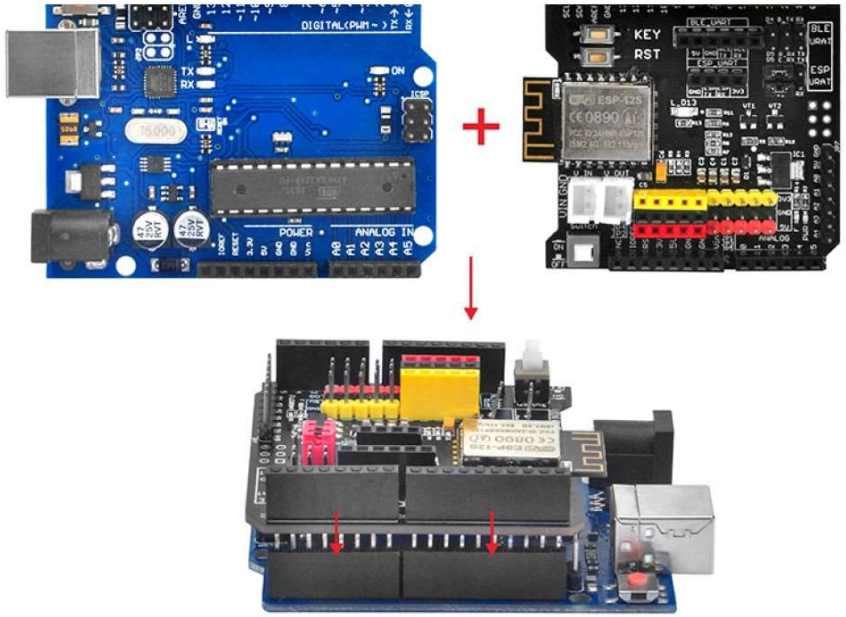
# Lab: Remote Control LED. . .

## Lab Setup Concept



# Lab: Remote Control LED. . .

## Lab Setup: Attaching WiFi Shield to the Arduino Compatible



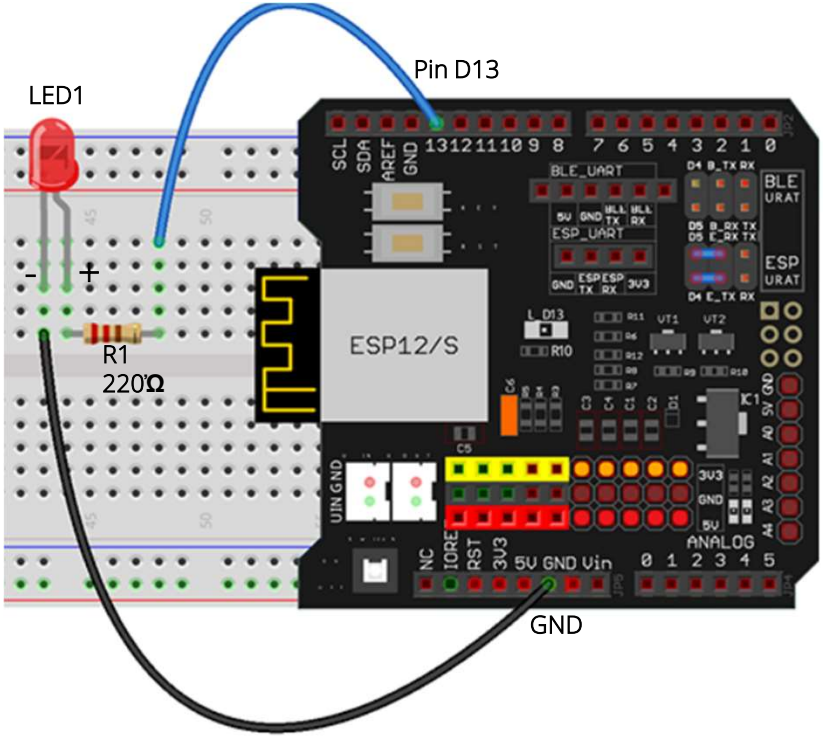
**Notes:**

a) Attach IoT unit to your development machine

b) Connect your Arduino Compatible to the correct COM port

# Lab: Remote Control LED . . .

## Lab Setup: Wiring the LED to the IoT unit




Note:

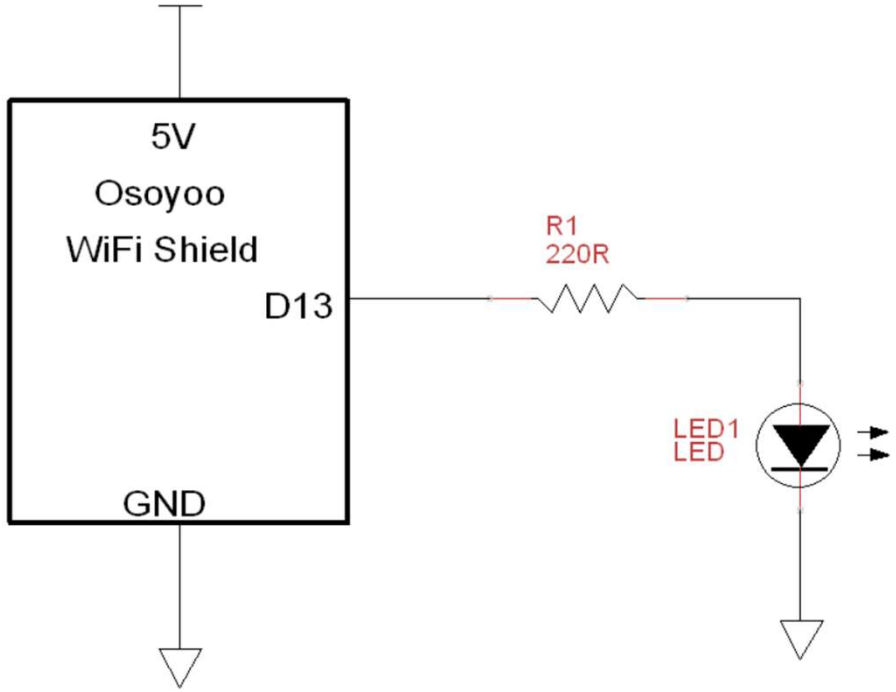
A 200ohm resistor may be used for R1.

# Lab: Remote Control LED . . .

## Lab Setup: IoT Receiver Electronic Circuit Schematic Diagram

Note:

A 200ohm resistor may be used for R1.



# Question 4

**What digital pin on the Arduino Compatible is used to drive the LED?**

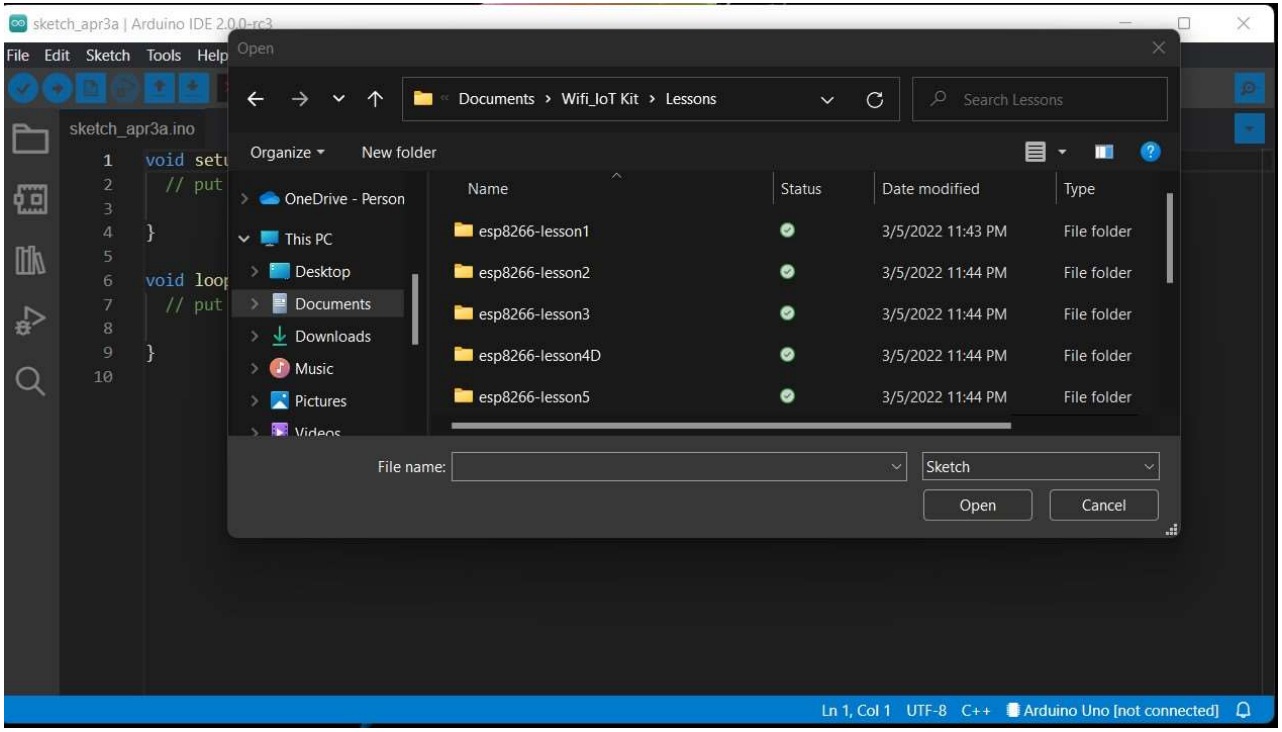
**a) D15**
**b) D13**
**c) D12**

# Lab: Remote Control LED . . .

## Lab Setup: Upload Lesson 2 code to Arduino Compatible

**Download the code from here!**

WiFi Internet of Things Learning Kit for Learn Coding with Arduino IDE 2: Remote control LED « osoyoo.com



# Lab: Remote Control LED . . .

## Lab Setup: Setup of Osoyoo WiFi Udp robot car mobile app

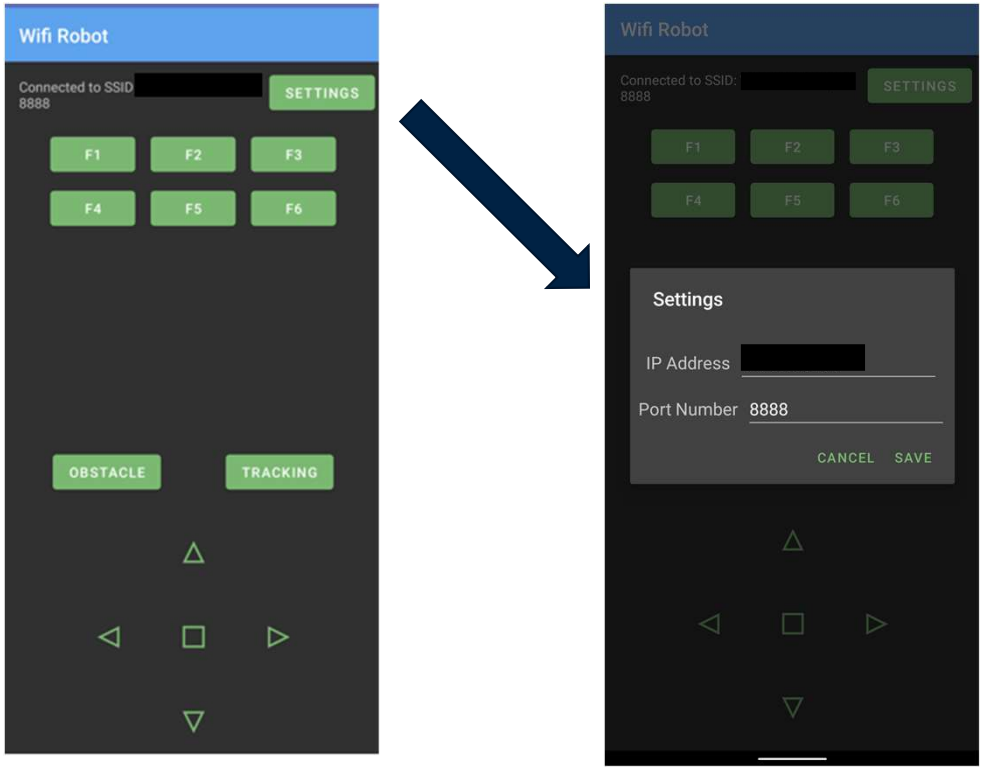
**Download the app from:**

a) Google Play
b) Apple Store

**Remote Control Operation:**

**F1 KeyPress:**
LED ON

**F2 KeyPress:**
LED OFF



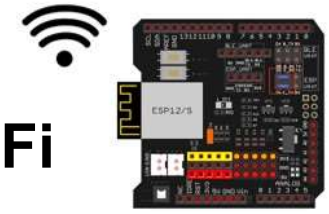
# Lab: Remote Control LED . . .

## Lab Setup: Single letter messages

| Button | UDP message |
|---|---|
| F1 | F |
| F2 | G |
| F3 | H |
| F4 | I |
| F5 | J |
| F6 | K |
| ▲ | A |
| ▼ | B |
| ► | R |
| ◄ | L |
| ⇪ | E |

When you press a button of the mobile app, the mobile app will send a single-letter message through UDP protocol to the target device (in this example, our Wifi Shield)

# Lab: Remote Control LED . . .



## Lab Setup: Lines of code (72 & 73) to remap keys on WiFi robot app

esp8266-lesson2.ino

```
68      char c=packetBuffer[0];
69      switch (c)    //serial control instructions
70      {
71
72        case 'F': digitalWrite(ledPin, HIGH) ;break; //TURN ON LED
73        case 'G':digitalWrite(ledPin, LOW) ;break; //TURN OFF LED
74
75        default:break;
76      }
77    }
```

# Lab: Remote Control LED . . .

## Lab Setup: Received packets

Output Serial Monitor ×

Message (Ctrl + Enter to send message to 'Arduino Uno' on 'COM6'

New Line 9600 baud

```
Received packet of size 1
Received packet of size 1
Received packet of size 1
Received packet of size 1
Received packet of size 1
Received packet of size 1
Received packet of size 1
Received packet of size 1
```

# Question 5

**What programming construct is used to map the mobile app buttons with the UDP message?**

**a) if – then - else**
**b) if – then – else if**
**c) switch - case**

# Thank you for attending

## Please consider the resources below:

ESP8266 Hardware Design Guidelines (www.expressif.com)

Osoyoo Website.(2022). WiFi iot learning kit. https://osoyoo.com/2020/05/30/wifi-iot-learning-kit-for-arduino/

Yang, S. (2011). *Internet-based control systems: Designs and applications*. Springer

CEC Continuing Education Center

DesignNews

# Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets