CEC Continuing Education Center

DesignNews

Hands-On with Industry 4.0 using the Raspberry Pi and the Arduino Platforms

DAY 3: A Conceptual Industry 4.0 Test Circuit Part 2 – Wireless Control

Webinar Logistics

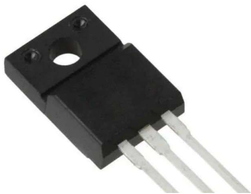
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Dr. Don Wilcher

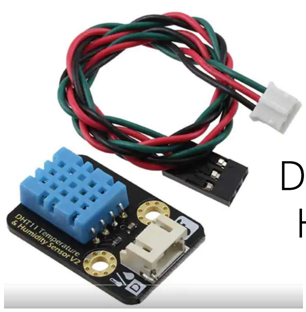
Visit 'Lecturer Profile' in your console for more details.

Course Kit and Materials

IRFS630A N-Channel PMOSFET

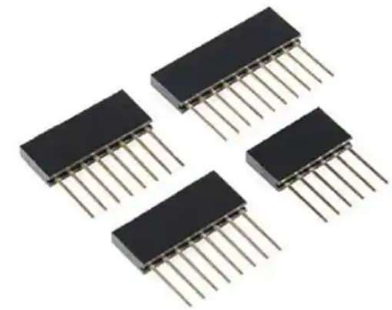


Axial DC Fan, 5VDC

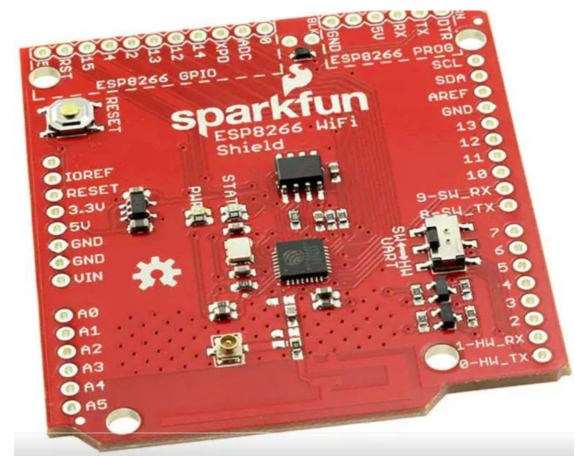


DHT Temperature-Humidity Sensor

Arduino Stackable Header Kit



Osoyoo WiFi Internet of Things Learning Kit



ESP8266 WiFi Shield

Agenda:

- Cyber-Physical Systems Model: A Research Perspective
- Modeling a DC Motor and PID Controller
- Basic Servo Motor Controllers and Circuits
- Lab: Wireless Servo Motor Controller

Industry 4.0:

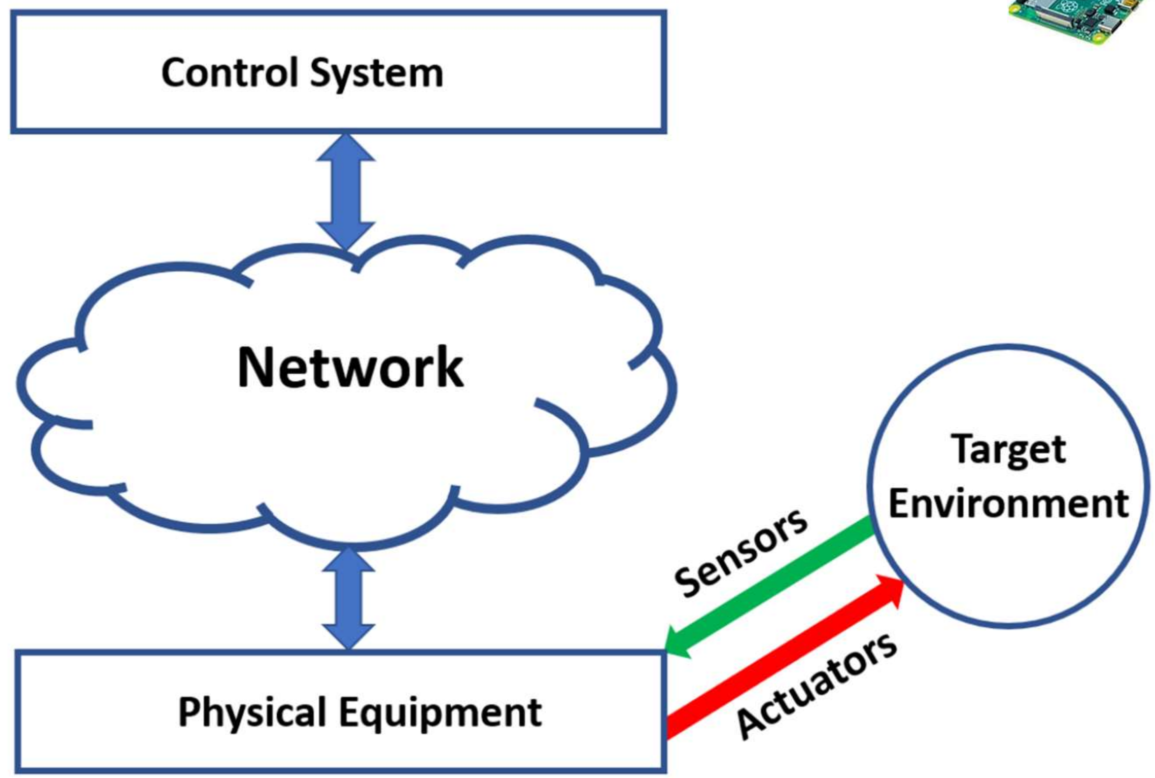
" Over the last two decades, there has been massive progress in the fields of information technology, automation, robotics, Big Data, sensor technology (Internet of Things) and Artificial Intelligence (Russmann et al. 2015). These technological advances may be interpreted as prerequisites for the increasingly popular concept of Industry 4.0, which refers to the current trend of digitalization, automation, and data exchange in manufacturing (Kagermann et al., 2013; Schwab, 2016)."

Continuing Education Center

Sponsored By

CPS System Model

"CPS is an engineering system designed to control and operate physical processes through computers" (Mughees, 2020).



The Model provides a hands-on framework for exploring CPS System Concepts using Physical Computing Techniques. Design Opportunities!!

Question 1

In slide 7, what two components interact with the Physical Equipment and Target Environment?

a) sensors, switches
b) sensors, motors
c) sensors, actuators
d) none of the above

CPS System Model: A Research Perspective

- CPS and the Internet of Things (IoT) are research fields with special interest to systems researchers.
- The special interest lies in education and experimentation.
- Off-the-shelf components are being used to educate the CPS specialist.
- The off-the-shelf components consist of:
  - a) An Arduino
  - b) Firmata,
  - c) single board computer (Raspberry Pi)
  - d) Software components (JavaScript, nodejs, Linux, and a cloud-based IDE

Skraba et al. (2019)

CPS System Model: A Research Perspective

- The CPS research perspective is to educate and train systems engineers on developing control algorithms for DC motor position controllers.
- One approach to this educational and training method is to allow system engineering learners to code the DC motor position controller models and control algorithms from "scratch."
- JavaScript would be the software to derive and implement the models and control algorithms.

Skraba et. al.(2019)

CPS System Model: A Research Perspective – A ChatGPT Lab Approach

A Generative Approach to Creating CPS Designs!

Create a CPS training system using an Arduino, a small DC motor, and C++ software

To create a CPS (Controlled Process System) training system using an Arduino, a small DC motor, and C++ software, you can follow these steps:

Step 1: Gather the required components

- Arduino board (e.g., Arduino Uno)
- Small DC motor (e.g., 3V-6V)
- Motor driver module (e.g., L298N)
- Potentiometer (e.g., 10K ohms)
- Breadboard and jumper wires

Step 2: Connect the components

- Connect the motor driver module to the Arduino board using jumper wires. The module should have four input pins (IN1, IN2, IN3, IN4) and two output pins (OUT1, OUT2).
- Connect the motor to the motor driver module's output pins (OUT1, OUT2).
- Connect the potentiometer to the Arduino board. Connect the outer pins to GND and 5V, and the middle pin to an analog input pin (e.g., A0).
- Connect the motor driver module's input pins to digital output pins on the Arduino board (e.g., IN1 to pin 2, IN2 to pin 3, IN3 to pin 4, and IN4 to pin 5).

CPS System Model: A Research Perspective – A ChatGPT Lab Approach

Step 3: Write the C++ code

- Include the necessary libraries (e.g., "AFMotor.h" for the motor driver module).
- Define the motor driver module and motor objects using the appropriate pins.
- Define a variable to store the potentiometer reading.
- Use the "analogRead" function to read the potentiometer value and map it to a speed value for the motor (e.g., between 0 and 255).
- Use the "setSpeed" function of the motor object to set the motor speed based on the potentiometer reading.
- Use the "run" function of the motor driver module to control the direction of the motor (e.g., clockwise or counterclockwise).
- Use a delay function (e.g., "delay(100)") to control the time interval between motor movements.

Here is an example code for the CPS training system:

CPS System Model: A Research Perspective – A ChatGPT Lab Approach

Here is an example code for the CPS training system:

scss   Copy code

```
#include <AFMotor.h>

AF_DCMotor motor(1); // create motor object on output 1 of motor driver module
int potPin = A0; // define potentiometer input pin

void setup() {
  pinMode(potPin, INPUT); // set potentiometer pin as input
  motor.setSpeed(0); // set initial motor speed to 0
}

void loop() {
  int potValue = analogRead(potPin); // read potentiometer value
  int speed = map(potValue, 0, 1023, 0, 255); // map potentiometer value to motor s
  motor.setSpeed(speed); // set motor speed
  if (speed > 0) {
    motor.run(FORWARD); // run motor clockwise
    delay(100); // delay between movements
  }
  else {
    motor.run(RELEASE); // stop motor
  }
}
```

CPS System Model: A Research Perspective – A ChatGPT Lab Approach

Step 4: Upload the code to the Arduino board

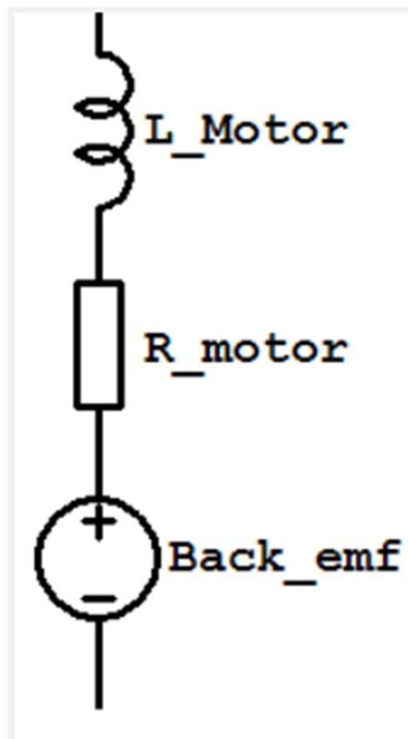
- Connect the Arduino board to your computer using a USB cable.
- Open the Arduino IDE software.
- Select the appropriate board and port in the "Tools" menu.
- Copy and paste the C++ code into a new sketch.
- Click the "Upload" button to upload the code to the Arduino board.

Step 5: Test the system

- Turn the potentiometer knob to adjust the motor speed.
- Observe the motor movements and direction based on the potentiometer reading.
- Adjust the code and circuit as needed to achieve the desired CPS training system

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller

Electrical Circuit Equivalent of a DC Motor



Source: https://www.precisionmicrodrives.com/ab-025#:~:text=The%20equivalent%20circuit%20for%20a,resistance%2C%20and%20back%20EMF%20respectively.

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller

Electrical Setup of a DC Motor

$$V_S = R\,i(t) + L\frac{d\,i(t)}{dt} + V_B \qquad (1)$$

where V_S is supplied voltage to the motor terminals, R resistance of the coils, $i(t)$ current, L inductance and V_B back Electro Motive Force (EMF) voltage which is proportional to the angular speed:

$$V_B = K_B\omega \qquad (2)$$

where ω is angular speed and K_B back EMF constant.

Skraba et. al.(2019)

Question 2

What term in Equation 1 on slide 16 represents the voltage drop across a resistor?

a) R di(t)/dt
b) L di(t)/dt
c) Ri(t)
d) none of the above

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller

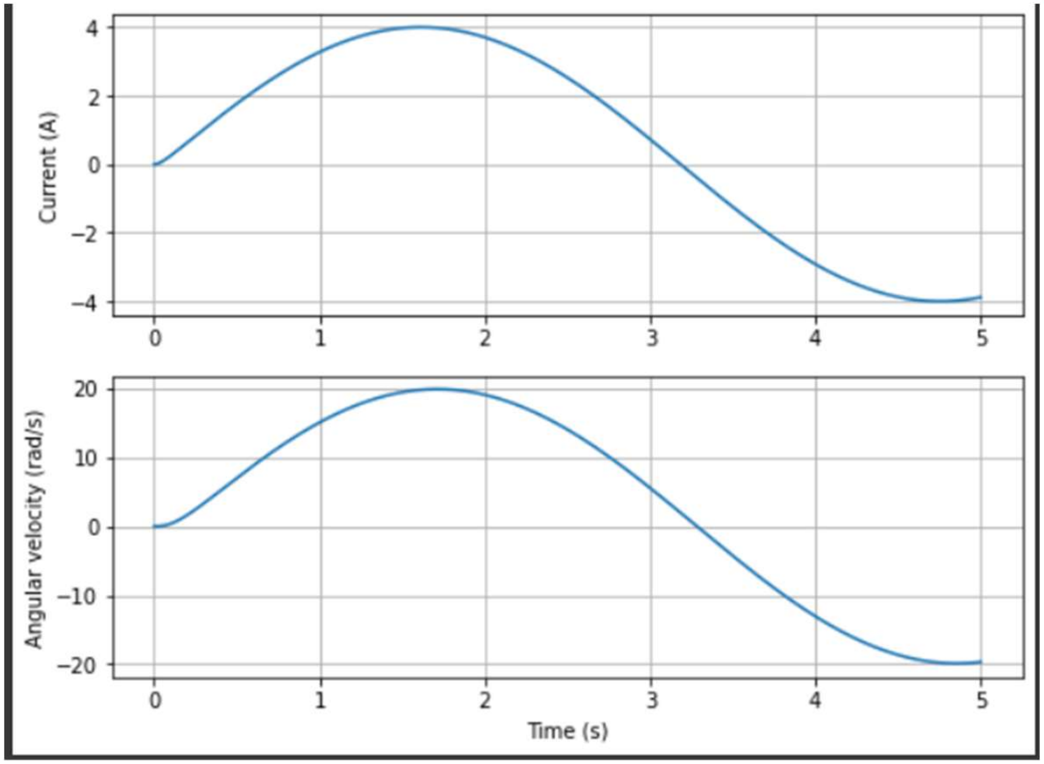
ChatGPT generated Python Code For a DC Motor: Partial Code

Note: 4 Prompt Engineering events occur for error-free results!

```python
import math

# define motor parameters
R = 2.5  # resistance (ohms)
L = 0.1  # inductance (henries)
k = 0.5  # torque constant (Nm/A)
b = 0.1  # friction constant (Nm/(rad/s))
J = 0.01  # moment of inertia (kg.m^2)

# define initial conditions
t0 = 0
I0 = 0
w0 = 0

# define simulation parameters
dt = 0.001  # time step
tf = 5      # final time

# define input voltage function
def V(t):
    return 10 * math.sin(t)

# define function for calculating motor dynamics
def motor_dynamics(t, y):
    I = y[0]
    w = y[1]
    V_in = V(t)
    tau = k * I - b * w
    dI_dt = (V_in - R * I - L * tau) / L
    dw_dt = tau / J
    return [dI_dt, dw_dt]
```

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller. . .

Colaboratory DC Motor generated Python Results



CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller

Setup of a PID Controller:

Industrial control systems typically use PID controllers for stability and self-error correction (Skraba et al., 2020).

$$u(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(\tau)\, \mathrm{d}\tau + K_d \frac{\mathrm{d}\varepsilon(t)}{\mathrm{d}t} \qquad (3)$$

where K_p, K_i and K_d are Proportional, Integral and Differential coefficients for the proportional, integral, and derivative terms respectively. $\varepsilon(t)$ is difference between the reference and output value representing error on which is the input to the PID controller.

Skraba et. al.(2020)

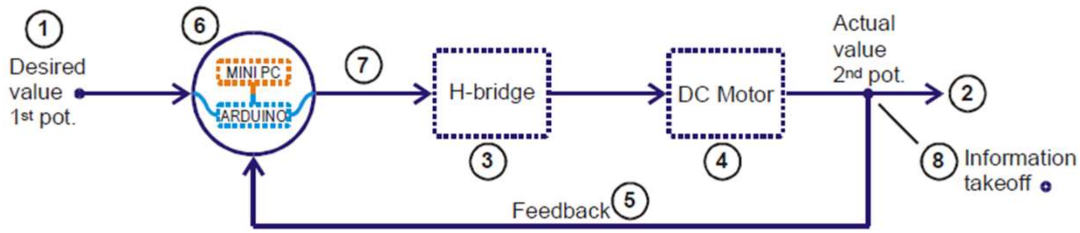
CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller. . .

ChatGPT generated Python Code For a PID Controller:

Note: One Prompt Engineering event occurred for error-free results!

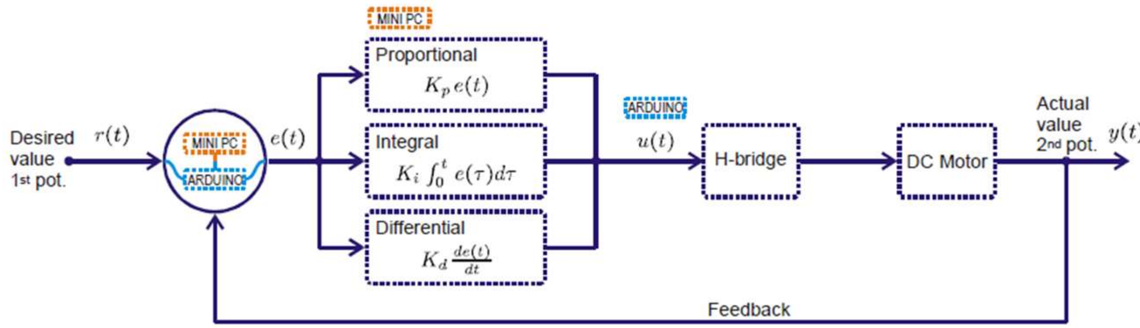
```python
import numpy as np
import matplotlib.pyplot as plt

class PIDController:
    def __init__(self, Kp, Ki, Kd, Ts):
        self.Kp = Kp
        self.Ki = Ki
        self.Kd = Kd
        self.Ts = Ts
        self.error_sum = 0
        self.last_error = 0

    def control(self, error):
        self.error_sum += error * self.Ts
        derivative = (error - self.last_error) / self.Ts
        self.last_error = error
        control = self.Kp * error + self.Ki * self.error_sum + self.Kd * derivative
        return control

# Define system parameters
K = 1.0     # process gain
tau = 2.0   # process time constant
theta = 0.5 # process time delay

# Define simulation parameters
dt = 0.01   # simulation time step
tf = 20.0   # simulation final time

# Define PID controller parameters
Kp = 1.0    # proportional gain
Ki = 0.5    # integral gain
Kd = 0.1    # derivative gain
Ts = dt     # controller sampling time
```

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller. . .



Skraba et. al.(2020)

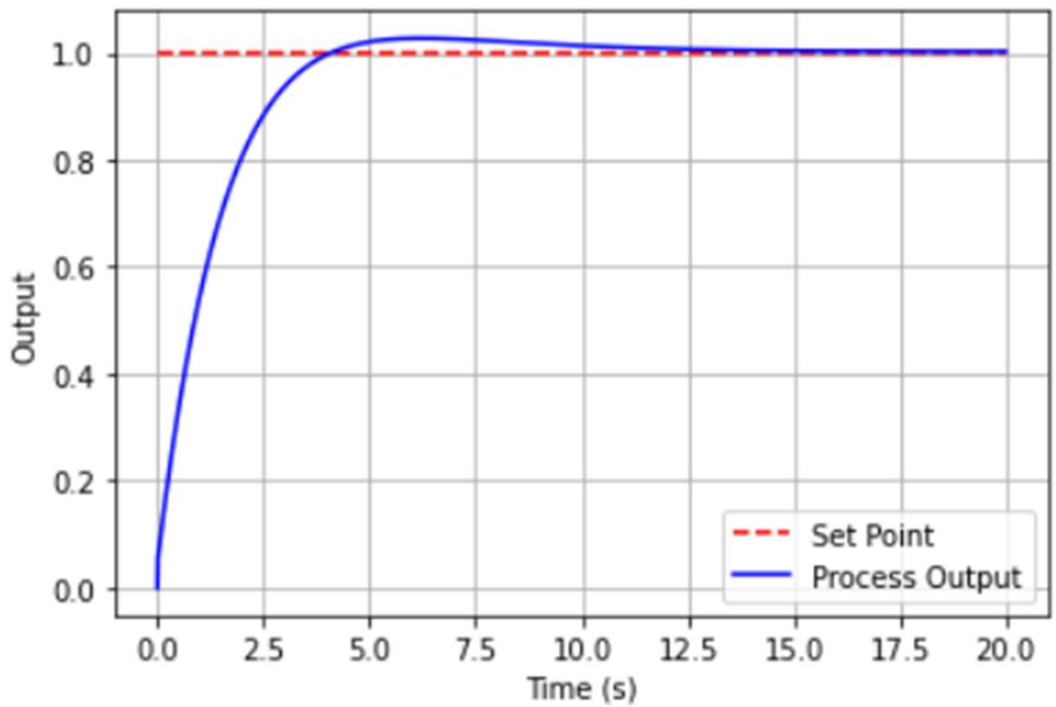
Typical Control System



Control System with PID Controller

CPS System Model: A Research Perspective – Modeling a DC Motor and a PID Controller. . .

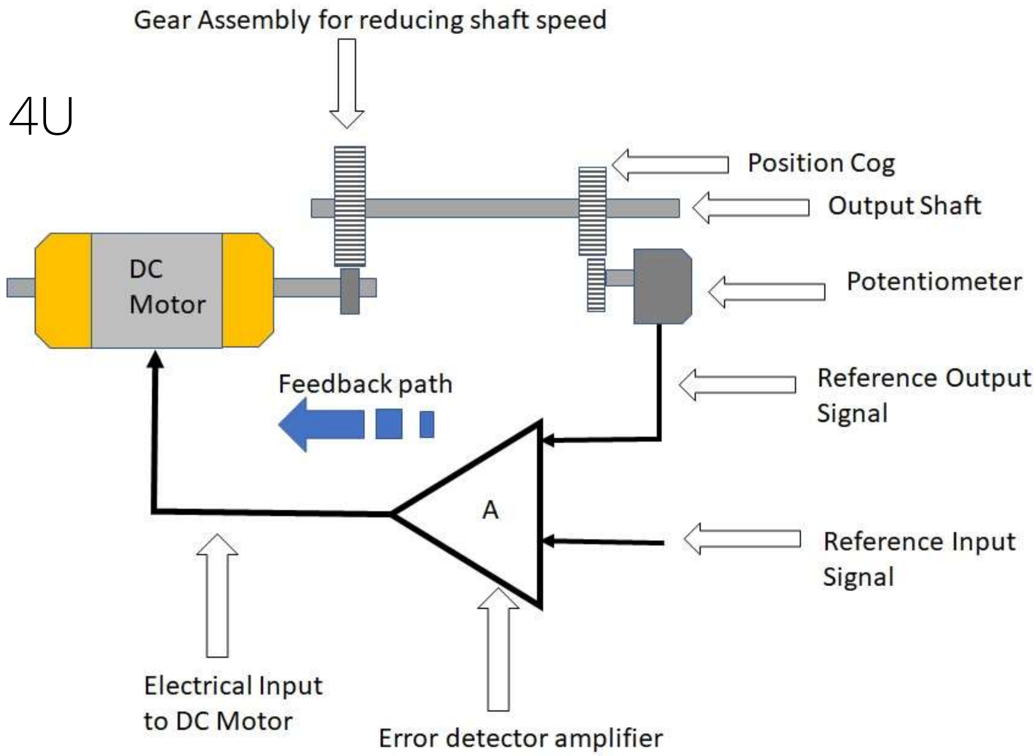
Colaboratory generated PID Controller Python Results



Basic Servo Motor Controllers and Circuits

What is a Servo Motor?

Electrical 4U



Servo Motor Functional Block Diagram

Note: A servo motor is the output device controlled by a PID

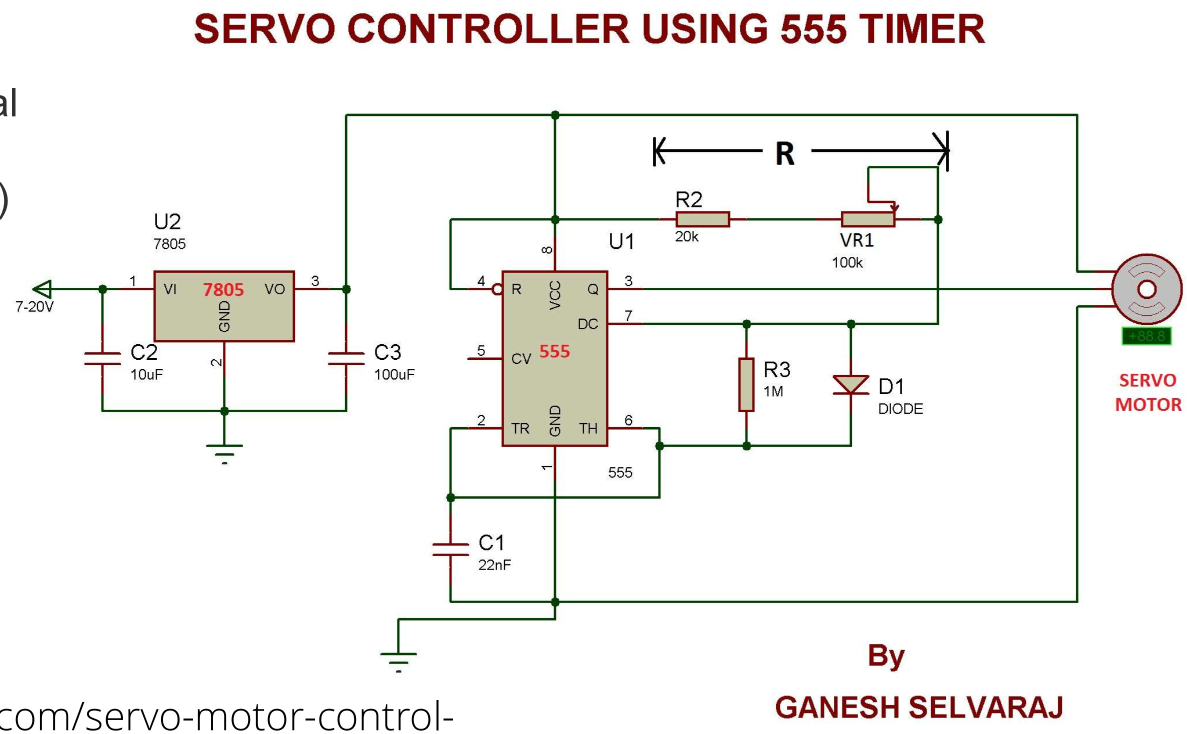
Question 3

A servo motor is the output device controlled by a PID.

a) True

b) False

Basic Servo Motor Controllers and Circuits

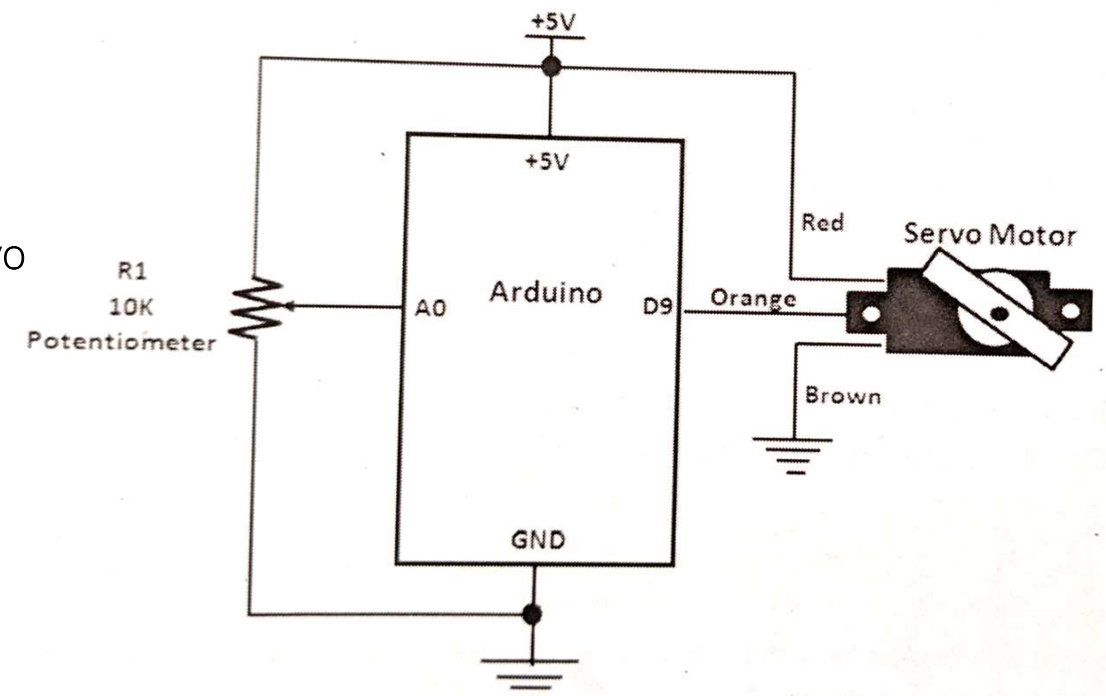


Pulse Width Modulated signal usually with a frequency of 50Hz (i.e., a period of 20msec) or 25Hz (Period of 40msec). The angle of the servo varies according to the ON period of the signal (also known as the pulse's duration or the pulse's width.)

Source:
https://www.engineersgarage.com/servo-motor-control-using-555-timer-ic/

Basic Servo Motor Controllers and Circuits. . .

A potentiometer-controlled servo motor

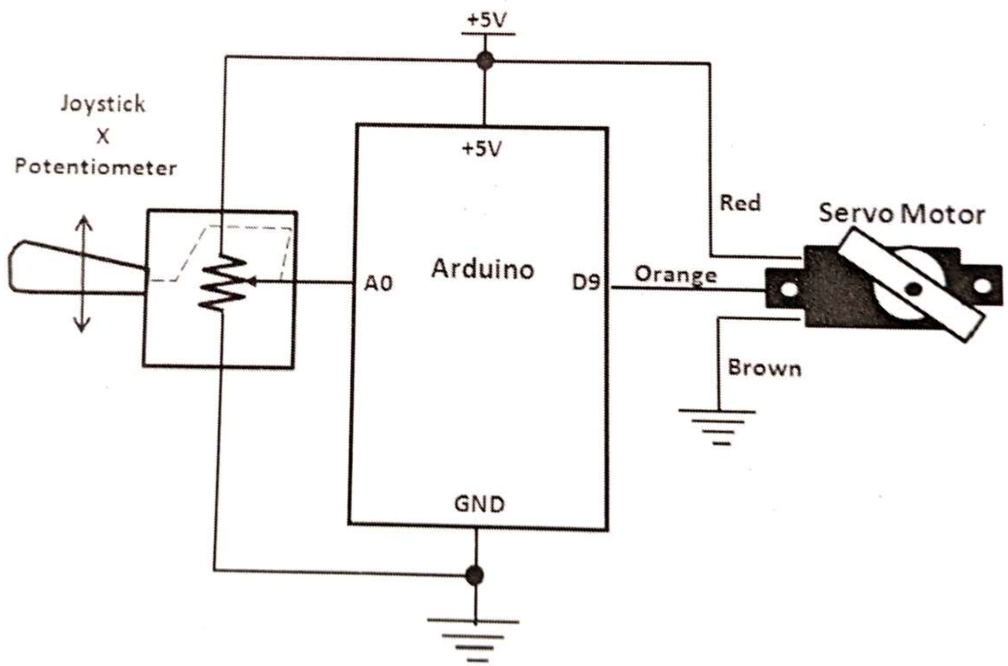


Source:

Wilcher, D. (2012). *Learn electronics with arduino*. Apress.

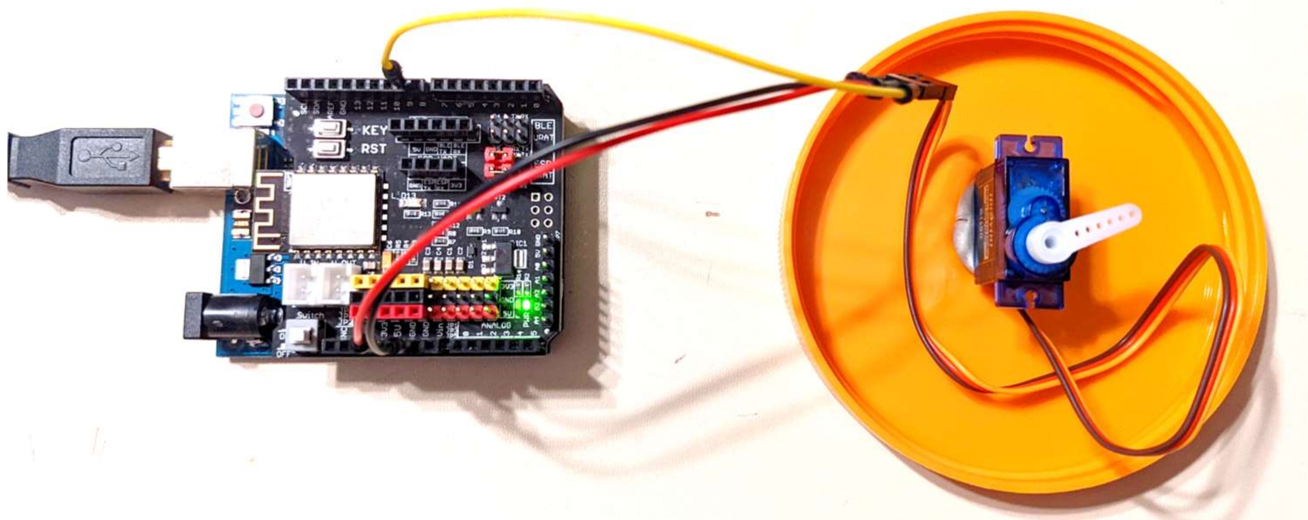
Basic Servo Motor Controllers and Circuits. . .

A joystick-controlled servo motor



Source:
Wilcher, D. (2012). *Learn electronics with arduino*. Apress.

Lab: Wireless Servo Motor Controller



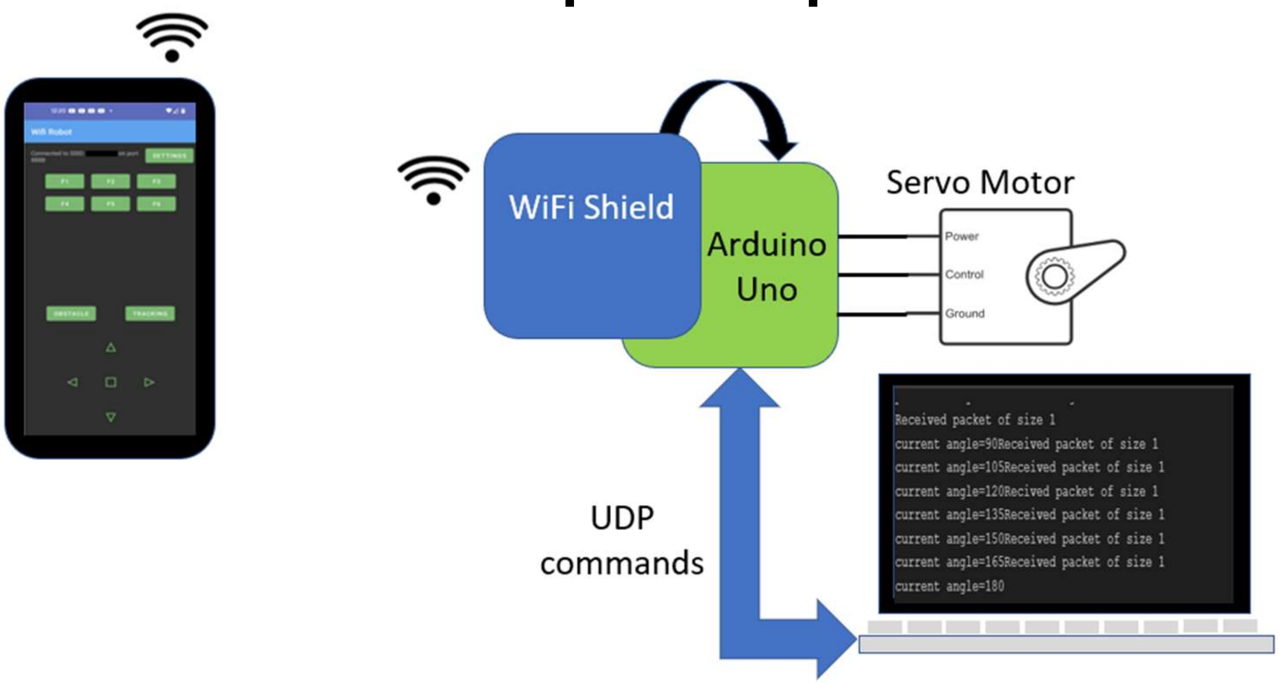
Lab: Wireless Servo Motor Controller. . .

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 light sensor.
- You will learn how to use a mobile app to communicate with WiFi to adjust a servo motor’s rotational angle.

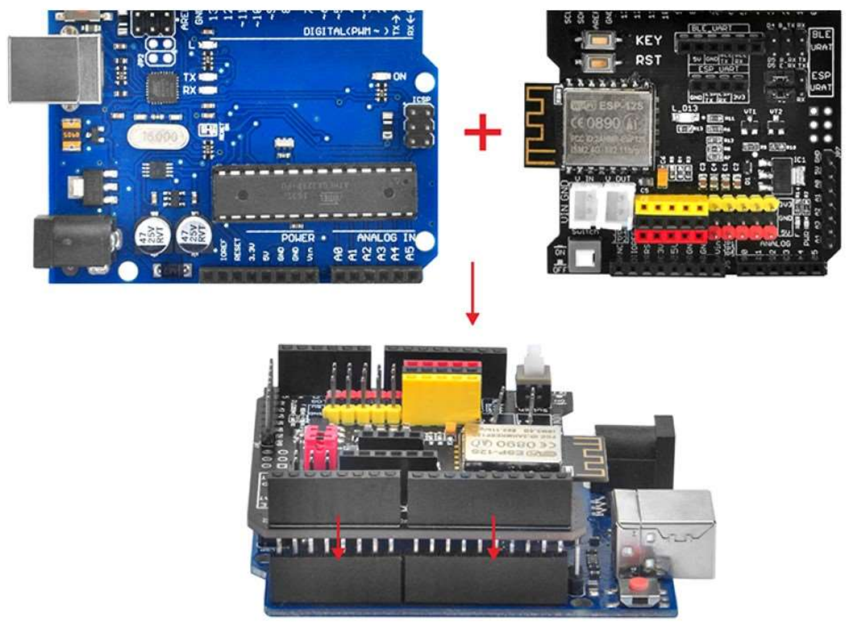
Lab: Wireless Servo Motor Controller. . .

Lab Setup Concept



Lab: Wireless Servo Motor Controller. . .

Lab Setup: Attaching WiFi Shield to the Arduino Compatible



Notes:

a) Attach IoT unit to your development machine

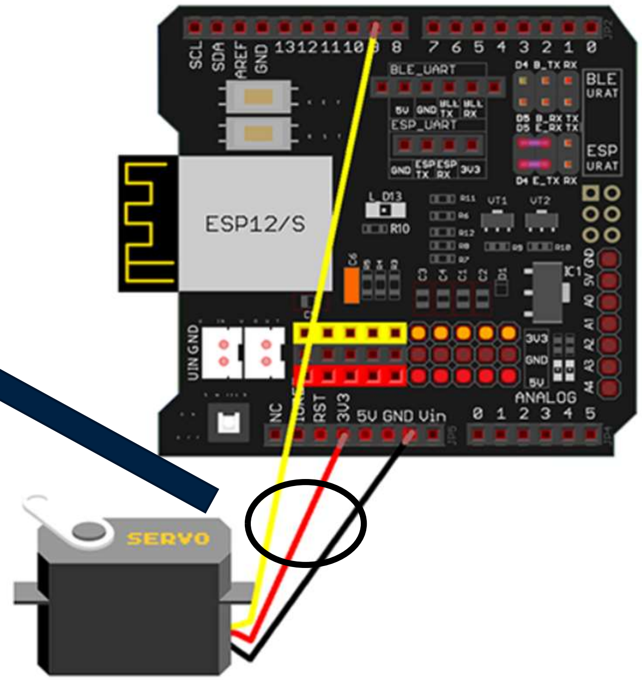
b) Connect your Arduino Compatible to the correct COM port

Lab: Wireless Servo Motor Controller. . .

Lab Setup: Wiring the Servo Motor to the IoT unit

Wiring Chart:

OSOYOO basic board	Servo Wire
GND	Brown
3.3V	RED
D9	Orange



Lab: Wireless Servo Motor Controller. . .

Lab Setup: Wiring the Servo Motor to the IoT unit

Jumper Harness Notes:

a) Insert individual jumper wires into servo motor's 3pin female connector

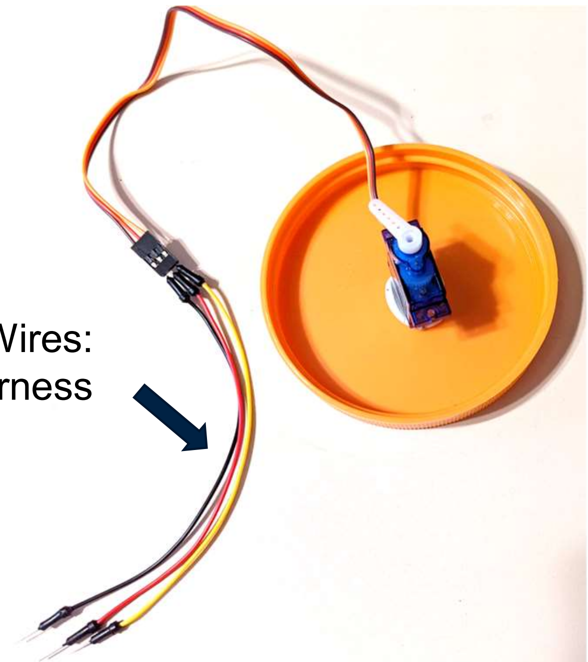
b) Individual jumper wire color scheme:

   i. Brown to Black(GND)

   ii. Red to Red(3.3V)

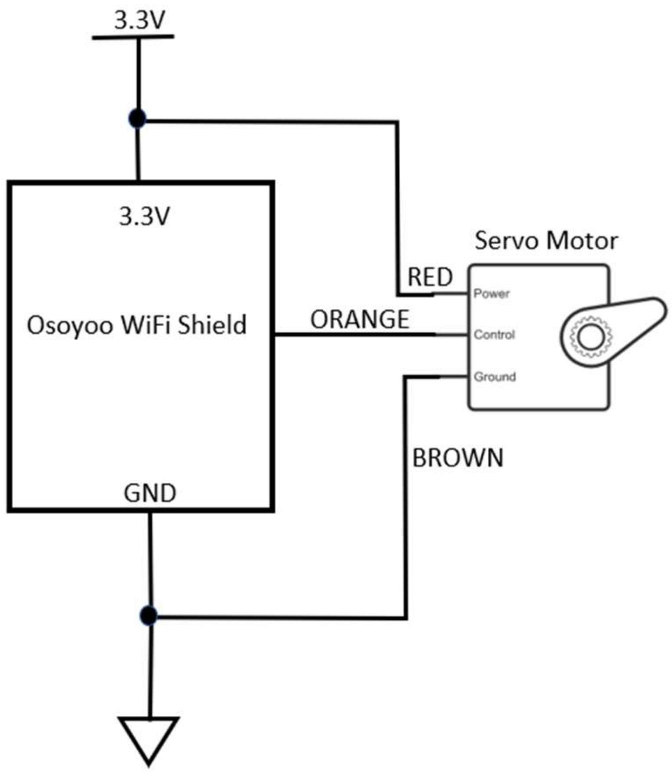
   iii. Orange to Yellow(D9)

Extended Wires: Jumper Harness Assembly



Lab: Wireless Servo Motor Controller. . .

Lab Setup: IoT Receiver - Servo Motor Controller Circuit Schematic Diagram



CEC Continuing Education Center

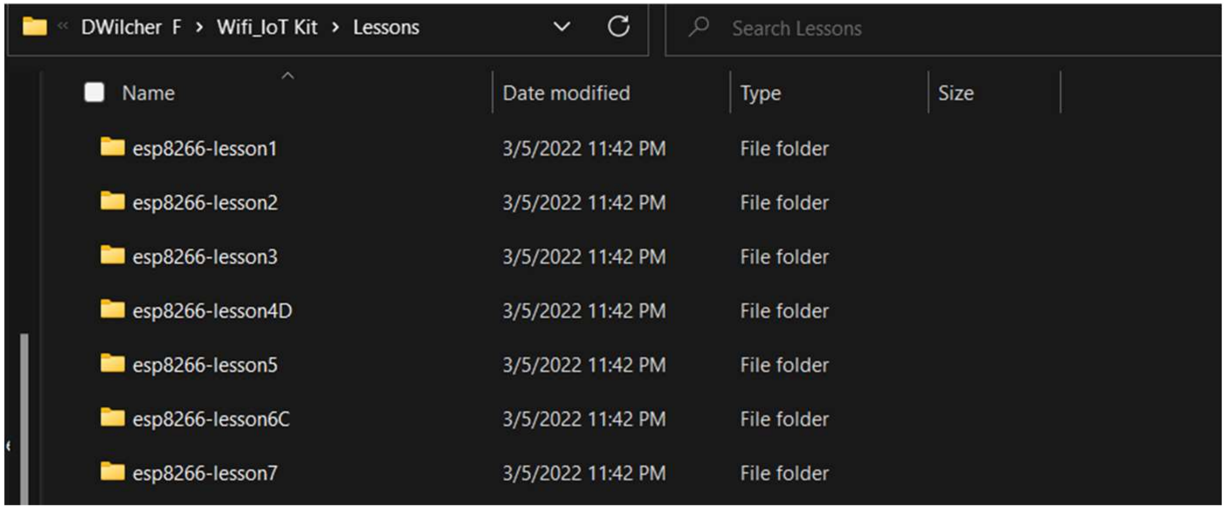
Sponsored By Digi-Key Corporation

Lab: Wireless Servo Motor Controller. . .

Lab Setup: Upload Lesson 6C code to Arduino Compatible

Download the code from here!

WiFi Internet of Things Learning Kit for Learn Coding with Arduino IDE 6: Servo motor « osoyoo.com



Lab: Wireless Servo Motor Controller. . .

Lab Setup: Mobile App Control

Mobile App:

OSOYOO WiFi UDP Robot Car APP

Note:

To get the best servo motor rotation response, the keys on the mobile device screen

Mobile App keys mapped to UDP commands

Button	UDP message
F1	F
F2	G
F3	H
F4	I
F5	J
F6	K
▲	A
▼	B
▶	R
◀	L
square	E

Servo motor response

- servo will rotate about 5 degree counterclockwise(left side)
- servo will rotate about 5 degree clockwise(right side)
- servo will rotate to right end (0 degree position)
- servo will rotate to left end (180 degree position)
- servo will rotate to central position (90 degree position)

Lab: Wireless Servo Motor Controller. . .

Partial C++ UDP Commands Code

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

  case 'A': angle=angle+5 ;break; //▲ button pressed, rotate 5 degree counterclockwise
  case 'B': angle=angle-5 ;break; //▼ button pressed, rotate 5 degree clockwise
  case 'L': angle=180 ;break; //< button pressed, rotate to 180 degree position case 'R': angle=0
  case 'E': angle=90 ;break; //SQUARE button pressed, rotate to 90 degree position
```

Question 4

On slide 37, case ‘B’ is mapped to what mobile app key?

- **a) the up arrow key**
- **b) the down arrow key**
- **c) the square key**

Lab: Wireless Servo Motor Controller. . .

Lab Setup: Arduino IDE Serial Monitor Response

```
Received packet of size 1
current angle=90Received packet of size 1
current angle=105Received packet of size 1
current angle=120Recived packet of size 1
current angle=135Received packet of size 1
current angle=150Received packet of size 1
current angle=165Received packet of size 1
current angle=180
```

CEC Continuing Education Center

Sponsored By Digi-Key Corporation

Lab: Wireless Servo Motor Controller. . .

Play with the Code!

Line 17:Change the servo pin and observe the servo motor response

```
#define servo_pin 9
```

Line 18: Change the angle initialize value and observe the servo motor response

```
int angle=90;
```

Question 5

Which code instruction statement is correct?

- a) define servo_pin 9
- b) servo_pin 9
- c) #define servo_pin 9
- d) none of the above

Thank you for attending

Please consider the resources below:

555 timer application: https://www.engineersgarage.com/servo-motor-control-using-555-timer-ic/

ElectronicsTutorial. (2021). Light sensors. *https://www.electronics-tutorials.ws/io/io_4.html#:~:text=The%20light%20sensor%20is%20a,)%20into%20electricity%20(electrons)*

ESP8266 Hardware Design Guidelines: https://www.espressif.com/en/support/documents/technical-documents

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for implementing the strategic initiative Industrie 4.0: Securing the future of German manufacturing industry*. https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf

Mughees, A. (Sept 05, 2020). *Discrete and process automation: From cyber-physical systems to pervasive intelligence*. https://electronics360.globalspec.com/article/15647/from-cyber-physical-systems-to-pervasive-intelligence

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

Russamann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engle, P., & Hanrisch, M. (2015). *Industry 4.0 – The future of productivity and growth in manufacturing industries*. The Boston Consulting Group.

Schwab, K.(2016). *The fourth industrial revolution.* Penguin Random House.

Thank you for attending

Please consider the resources below:

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

Skraba, A., Stanovov, V., & Semenkin, E. (2019). Modelling of dc motor and educational application in cyber-physical systems. *Materials Science and Engineering*, 537. https://doi:10.1088/1757-899X/537/4/042008

Skraba, A., Stanovov, V., & Semenkin, E. (2020). Development of control systems for study of PID controller in the framework of cyber-physical systems. *Materials Science and Engineering*, 537. https://doi:10.1088/1757-899X/537/4/042008

Wilcher, D. (2012). *Learn electronics with arduino*. Apress.

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key Electronics

informa markets