

สัมมนาเชิงปฏิบัติการเรื่อง ^{*} "ไอโอทีสำหรับระบบควบคุม และการเรียนรู้ของเครื่อง" ^{*}

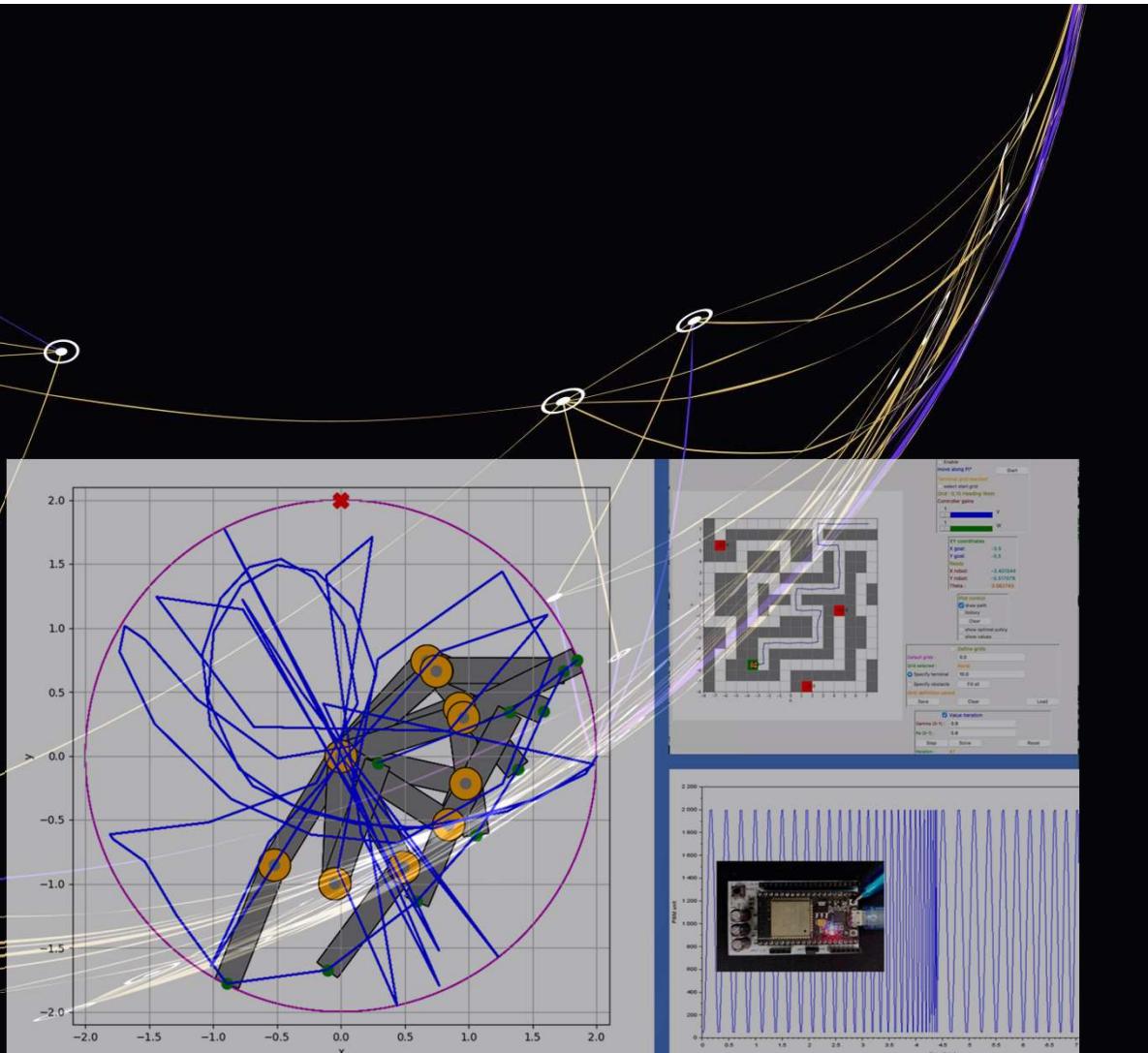
"IOT FOR CONTROL SYSTEMS AND MACHINE LEARNING" SEMINAR & WORKSHOP

ภาควิชาพิสิกส์ คณะวิทยาศาสตร์
มหาวิทยาลัยนเรศวร

17 กุมภาพันธ์ 2567 : 9 AM – 4 PM

ดร.วโรดม ตุ้มจินดา

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์



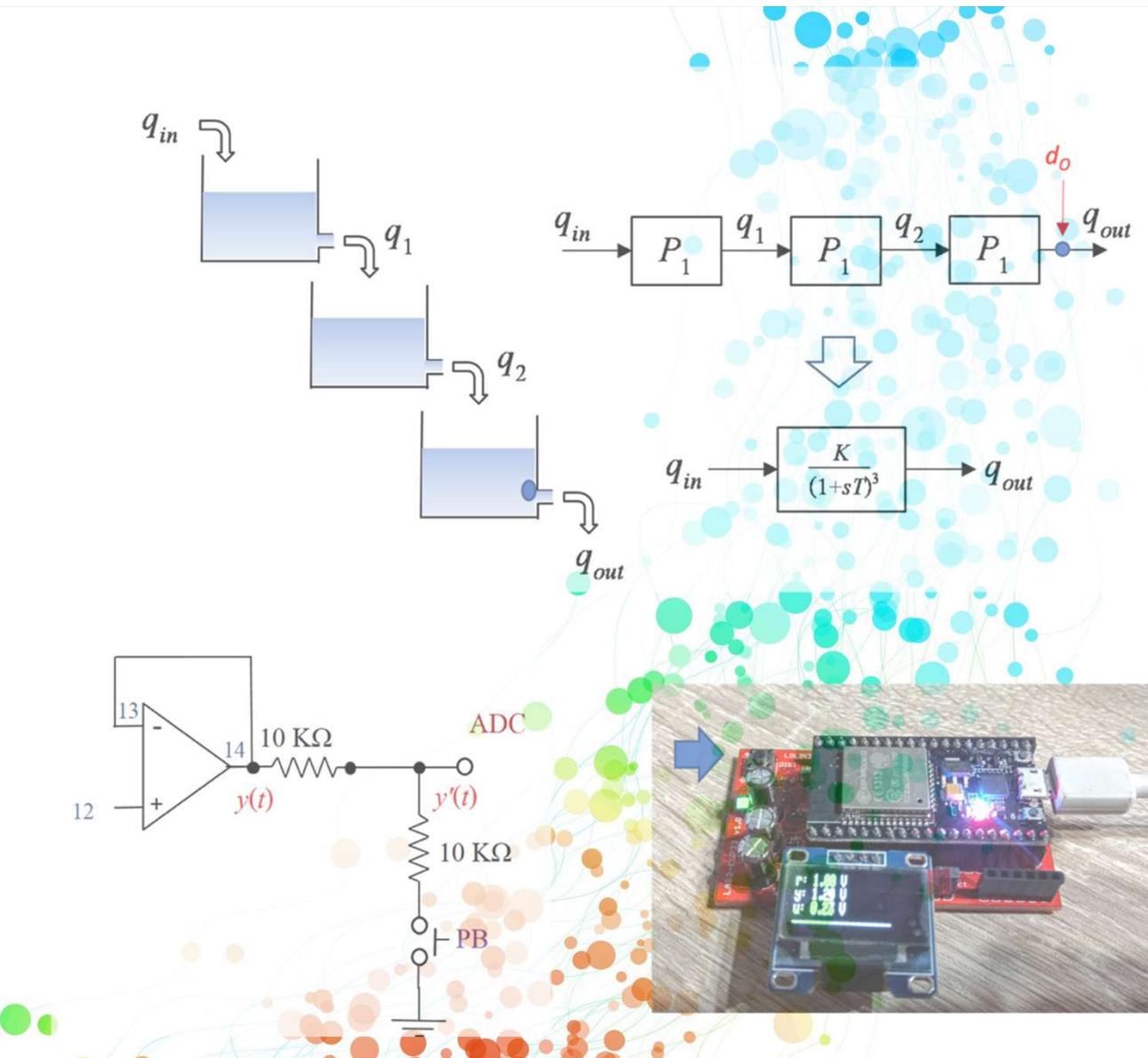
กำหนดการ

▶ 9 AM - 12 PM

- ติดตั้งซอฟต์แวร์ สาธิตการใช้ Jupyter notebook และ Thonny
- วงจรเชื่อมต่อ กับ MCU และการปรับแต่งสัญญาณ
- พื้นฐานของระบบควบคุม (เน้นตัวควบคุม PID)
- การอิมเพลเม้นต์ระบบควบคุมเชิงเส้น
- การจำลองระบบผังตัวบน Wokwi
- การพัฒนาไอโอทีบัน NETPIE 2020
- การแสดงผลและควบคุมบน NETPIE 2020 dashboard
- เขียนโปรแกรมไมโครไฟทอนเพื่อควบคุมอัตราการไหลของถังน้ำ 3 ระดับ

▶ 1 – 4 PM

- การใช้งาน paho-mqtt บน Jupyter notebook เพื่อนำข้อมูลจากระบบผังตัวมาแสดงผลในรูปแบบที่ต้องการ
- การอิมเพลเม้นต์ตัวควบคุมไม่เป็นเชิงเส้นสำหรับแขนกล 2 กำนันต่อ (2-link manipulator)
- การจำลองจลนศาสตร์หุ่นยนต์ (robot kinematics)
- ตัวควบคุมสำหรับหุ่นยนต์เคลื่อนที่ (mobile robot)
- การวางแผนเส้นทางเดินโดยอัลกอริทึมวนซ้ำมูลค่า (value iteration)
- สรุปการสัมมนา



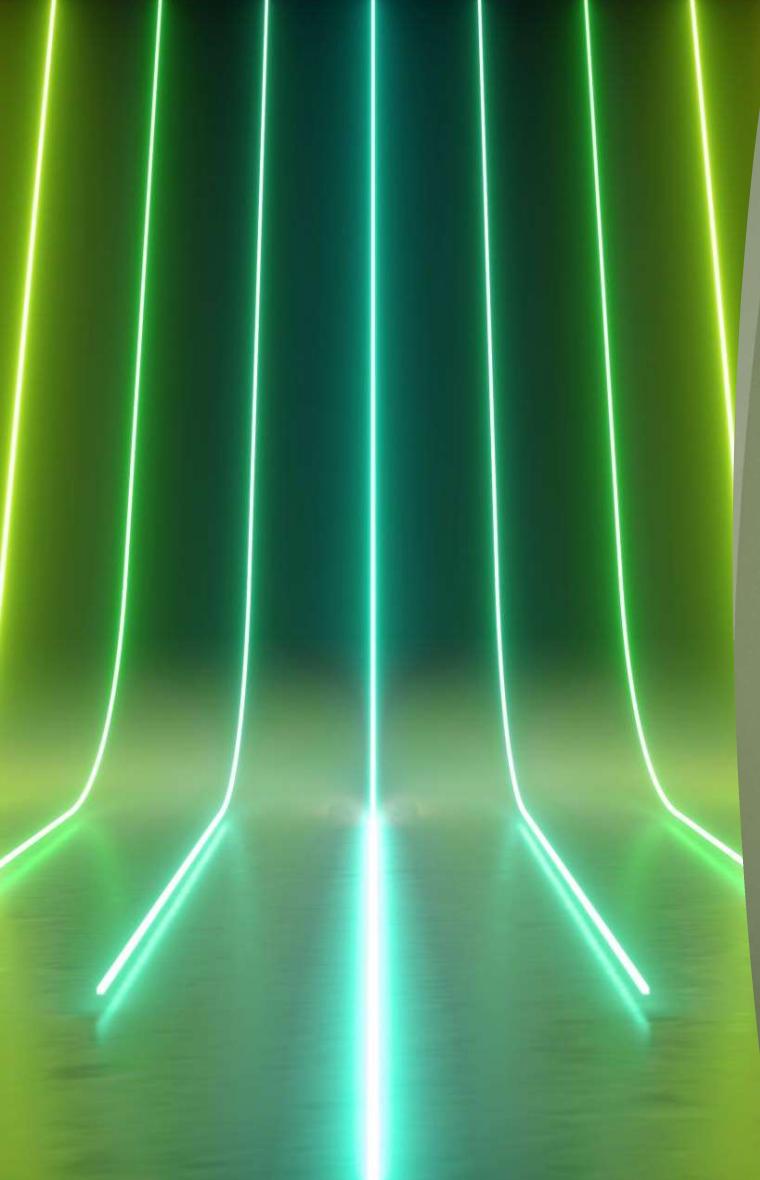
វាគម្រោង :
9:00 AM – 12:00 PM

อุปกรณ์และซอฟต์แวร์ที่ใช้ในการอบรม

- เครื่องคอมพิวเตอร์ระบบปฏิบัติการ Windows, Mac-OSX หรือ Linux
- สัญญาณ WiFi เพื่อเชื่อมต่ออินเทอร์เน็ต (สร้าง hotspot จากมือถือของผู้เรียนได้)
- ซอฟต์แวร์ (ทั้งหมดเป็น open-source หรือบริการที่ไม่มีค่าใช้จ่าย)
 - Jupyter notebook (สามารถใช้ Google colab)
 - Thonny <https://thonny.org/>
 - NETPIE 2020 <https://netpie.io/>
 - Wokwi <https://wokwi.com/>
 - Micropython <https://micropython.org/>

เอกสารประกอบ

- ▶ Slide, program , notebook ทั้งหมดรวมอยู่ในหน้า github
<https://github.com/dewdotninja/nuws24>
- ▶ การสัมมนานี้สอดคล้องกับเนื้อหาในหนังสือ "การโปรแกรมไฟฟอนสำหรับงานควบคุมและฝังตัว" ที่สามารถดาวน์โหลดได้ฟรีทั้งเล่มจากลิงก์
<https://github.com/dewdotninja/py4conemb>

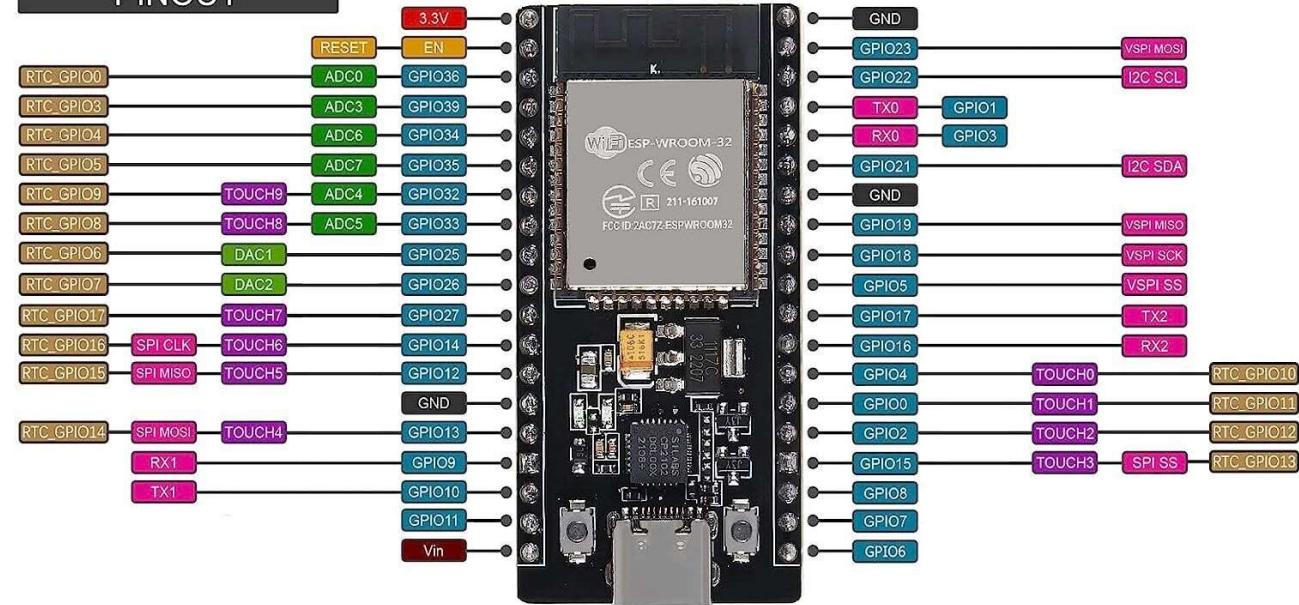


1. ส่วนต่อประสาน硬件และ
และการปรับแต่งสัญญาณ

Hardware interfacing
and signal conditioning

NodeMCU-32S

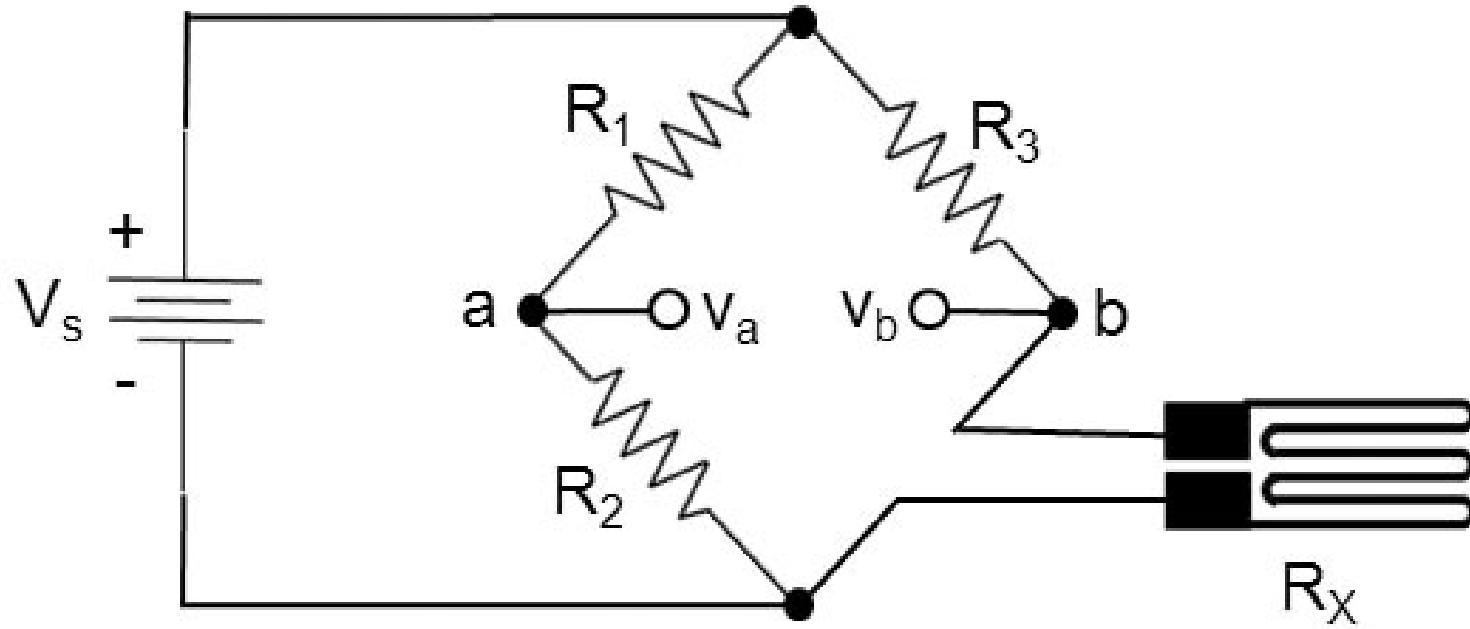
PINOUT



NodeMCU-32S has a total of 32 GPIO pins,
numbered from P0 - P39 corresponding to GPIO0 - GPIO39 in the pin diagram

Except for GPIO34, GPIO35, GPIO36, and GPIO39, which can only be input, but not output,
most of the other GPIOs can perform I/O bidirectional input and output control

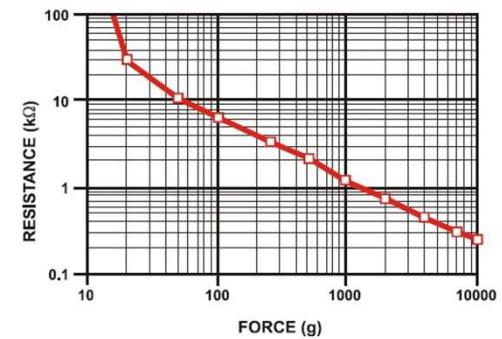
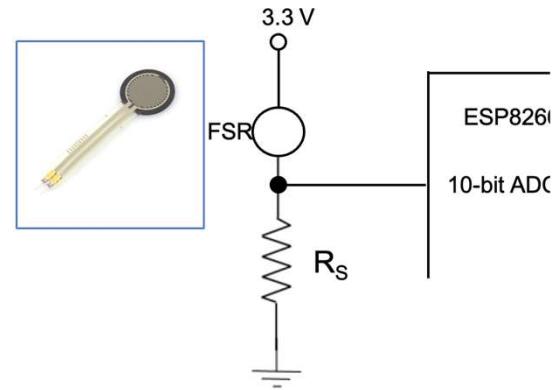
ESP32 pinout (nodemcu-32S)



การใช้งานจรรแบ่งแรงดัน (voltage divider)

ตัวอย่าง 1.1 : FSR (Force Sensitive Resistor)

- ▶ กำหนด $R_S = 10000$ โอห์ม
- ▶ สร้างสมการเส้นตรงที่แกน $X =$ แรงดัน ADC และแกน $Y =$ แรงที่กดบน FSR

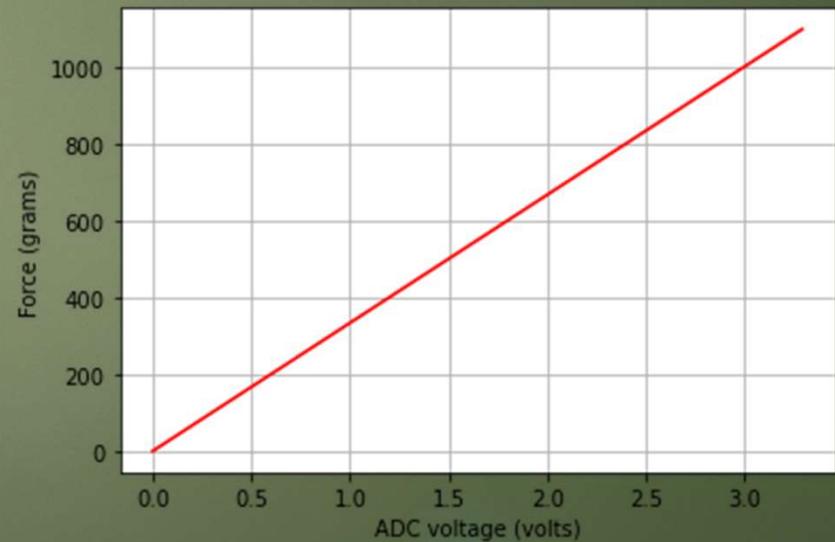


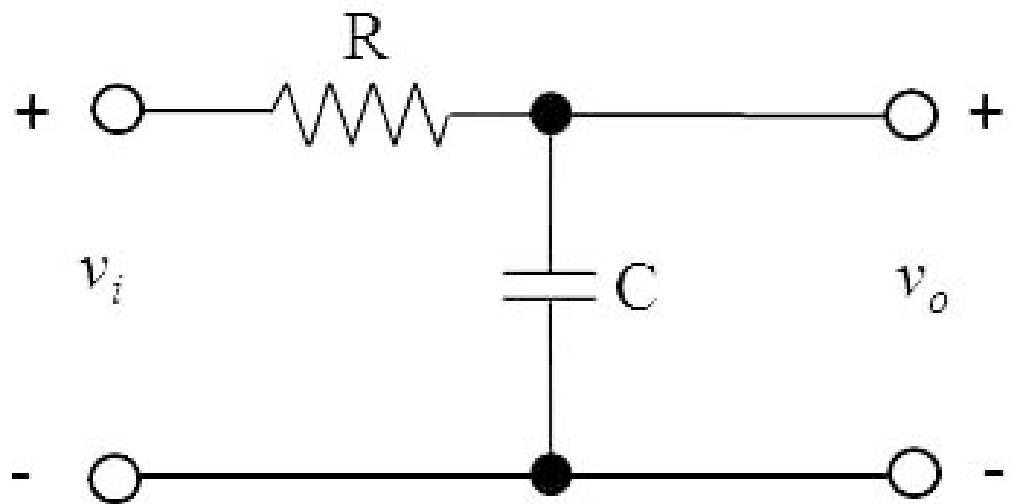
Solution

- ตั้งชื่อความต้านทาน $FSR = Rx$ และแรงดัน $ADC = Vs$
- จากกราฟ $Rx \rightarrow$ อนันต์ (เสมีอนเปิดวงจร) เมื่อ $F = 0 \rightarrow Vs = 0$
- เมื่อ $F = 1000$ g จากกราฟได้ $Rx = 1000$ โอห์ม คำนวณ Vs
- คำนวณ slope จาก $F = mVs$
- พล็อตกราฟเส้นตรง

```
vsvec = np.arange(0,3.3,0.01)
# ADC range
fvec = m*vsvec
plt.figure()
plt.plot(vsvec, fvec,'r-')
plt.xlabel('ADC voltage (volts)')
plt.ylabel('Force (grams)')
plt.grid(True)
plt.show()
```

$$\begin{aligned}Rs &= 10000 \\Rx &= 1000 \text{ # at } F = 1000 \\Vs &= 3.3*Rs/(Rx+Rs) \\m &= F/Vs\end{aligned}$$





$$H(s) = \frac{1}{RCs + 1}$$

$$\omega_o = \frac{1}{RC}$$

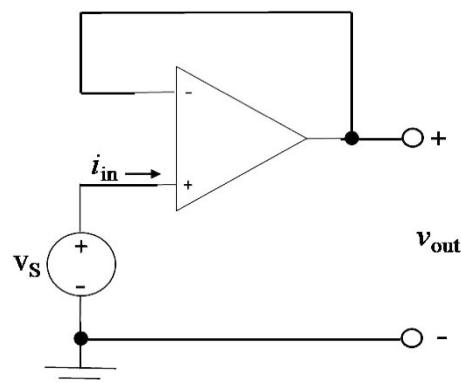
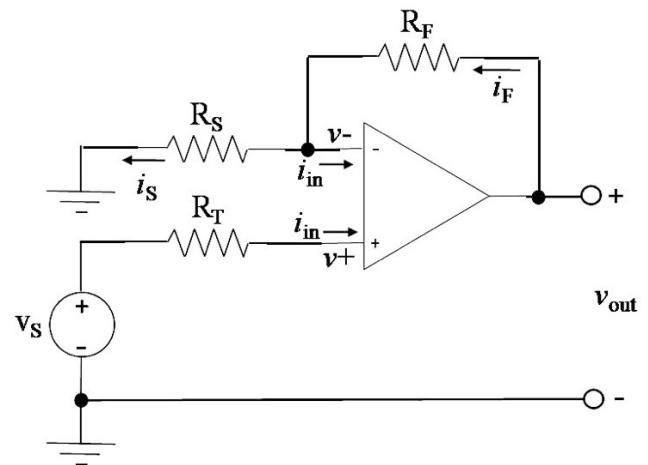
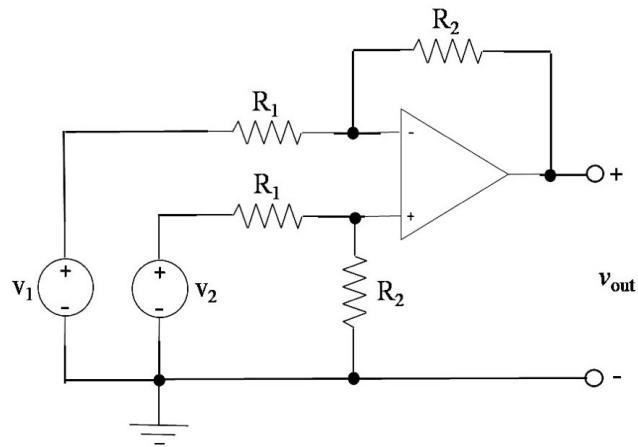
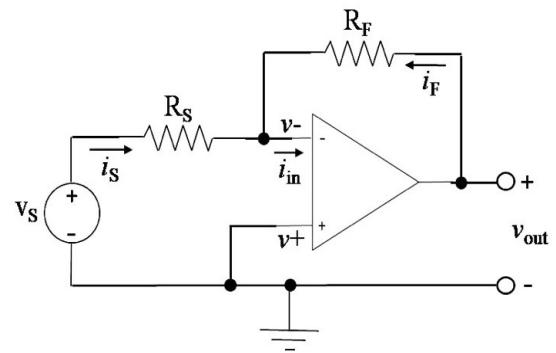
การกรองสัญญาณ

รบกวนโดยวงจร

RC

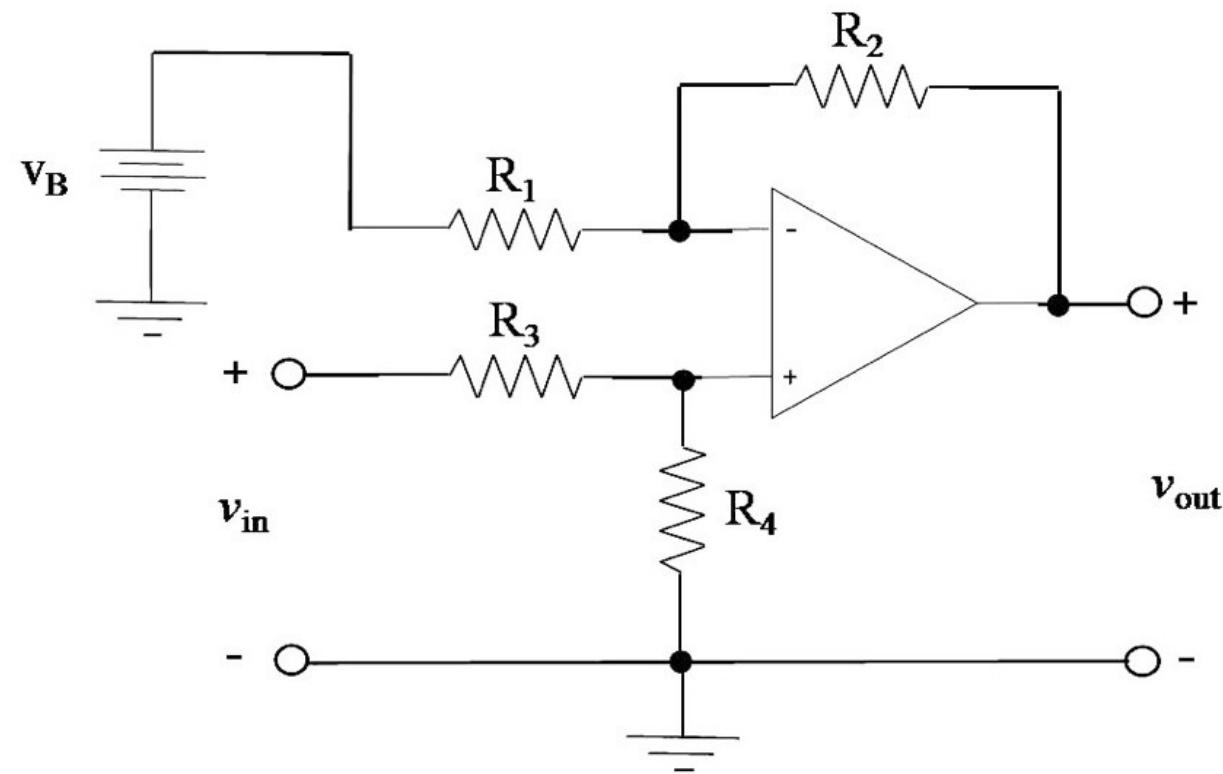
ดูตัวอย่าง 2.6 ในหนังสือ

วงจรอปปेनซ์ (operational amplifiers)



Quiz: จับคู่

1. วงจรขยายผลบวก
2. วงจรขยายไม่ผลบวก
3. วงจรตามแรงดัน
4. วงจรขยายผลต่าง

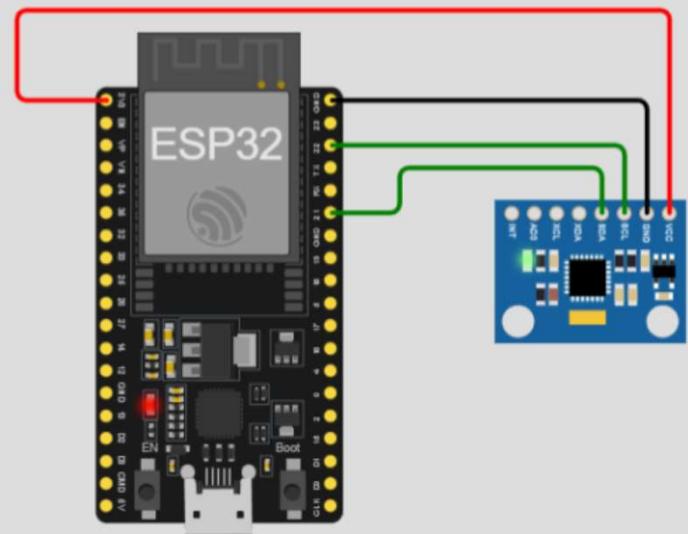


วงจรเลื่อนระดับ
แรงดัน
(level
shifter)

ดูตัวอย่าง 2.8-2.9 ในหนังสือ

mpu6050.py

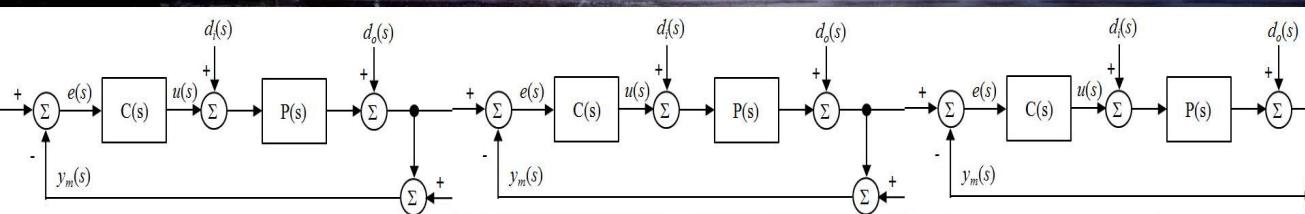
```
1 from machine import I2C  
2 from machine import Pin  
3 #from machine import sleep  
4 from time import sleep  
5 import mpu6050  
6 i2c = I2C(scl=Pin(22), sda=Pin(21))  
7 mpu= mpu6050.accel(i2c)  
8 while True:  
9     mpu.get_values()  
10    print(mpu.get_values())  
11    sleep(1)  
12
```



ຕົວຢ່າງການໃຊ້ໂມຄູລ i2C

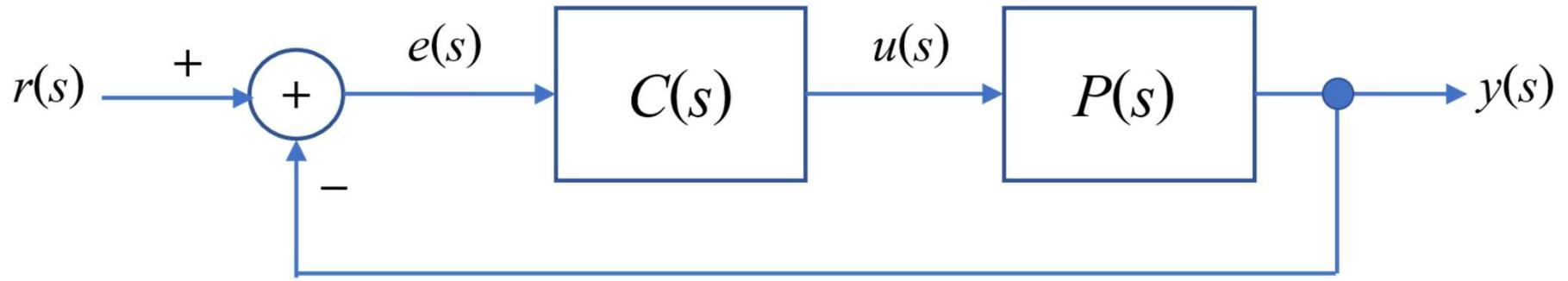
<https://wokwi.com/projects/388794293974202369>

$F = G \frac{m_1 m_2}{d^2}$
 $i\hbar \frac{\partial}{\partial t} \psi = \hat{H} \psi$
 $E = mc^2$
 $\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}$



พื้นฐานของ ระบบควบคุม

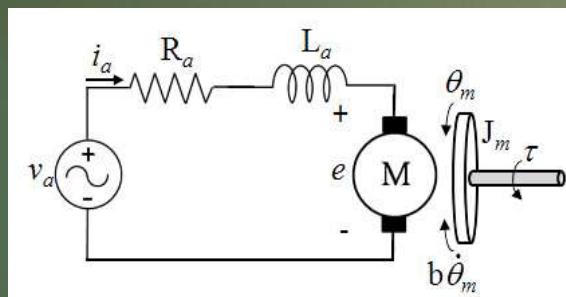
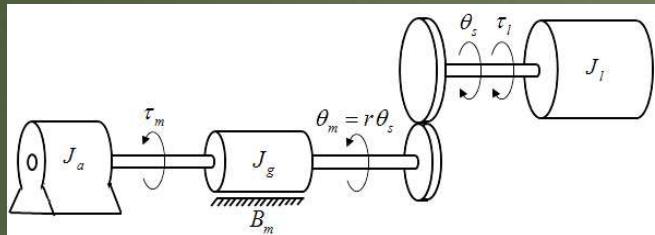
FUNDAMENTALS OF
CONTROL SYSTEMS



แผนภาพทั่วไปของระบบป้อนกลับ LTI-SISO

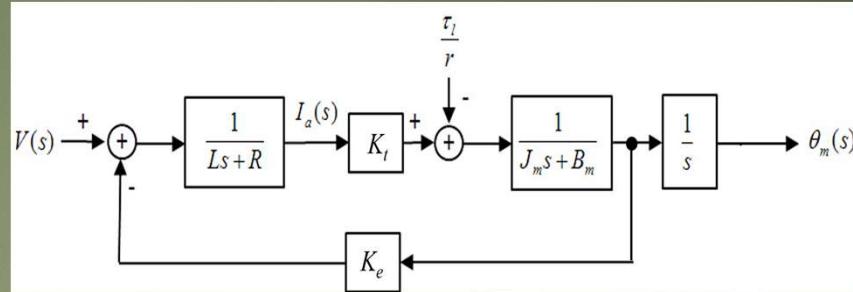
(Linear Time Invariant, Single-Input-Single Output)

ฟังก์ชันถ่ายโอน (transfer function)

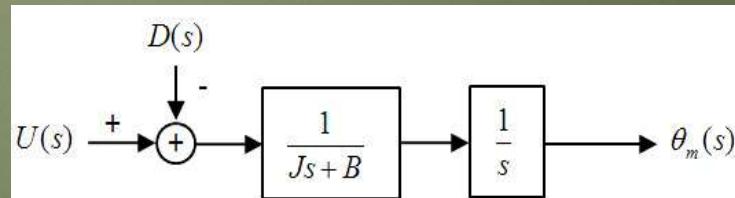


differential equation

Laplace transform

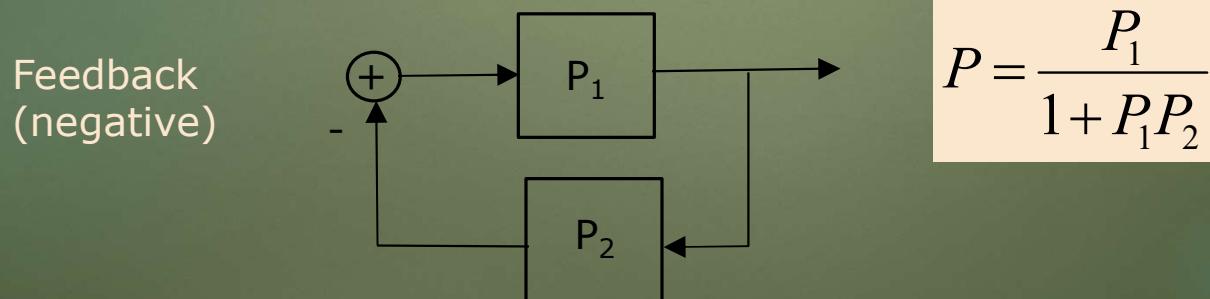
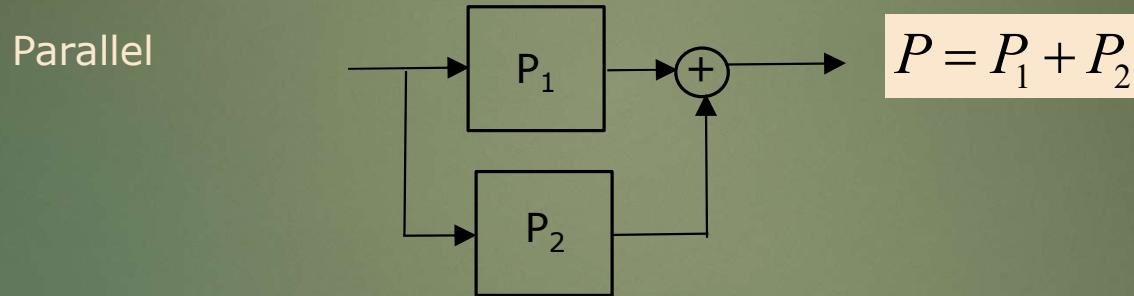
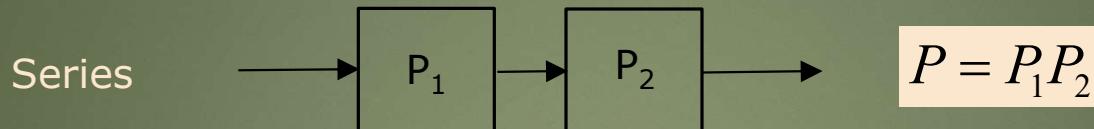


Block diagram reduction

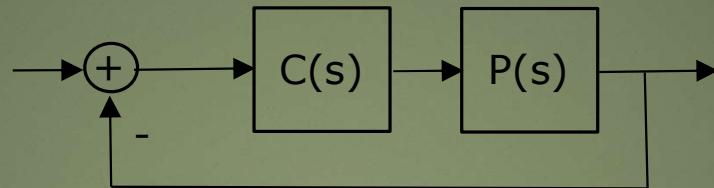


$$P(s) = \frac{1}{s(Js+B)}$$

การลดทอนแผนภาพบล็อก (block diagram reduction)



ຝຶກ໌ຂັ້ນຄ່າຍໂອນວເປີດແລະວົງປິດ



ຝຶກ໌ຂັ້ນວາງ :

Loop t.f.

$$L(s) = C(s)P(s)$$

ຝຶກ໌ຂັ້ນຄວາມໄວ :

Sensitivity t.f.

$$S(s) = \frac{1}{1 + L(s)}$$

ຝຶກ໌ຂັ້ນເຕີມເຕີມຄວາມໄວ : Complementary sensitivity t.f.

$$T(s) = \frac{L(s)}{1 + L(s)}$$

ໂປລແລະ ຊື່ໂຮງອັນພຶກສັນລ່າຍໂອນ

Ex.

$$P(s) = \frac{s+0.5}{10s^2 + 0.1s + 1}$$

← Roots ຂອງ numerator ອື່ອ zeros

← Roots ຂອງ denominator ອື່ອ poles

```
In [2]: s=ctl.tf('s')
P = (s+0.5)/(10*s**2+0.1*s+1)
P

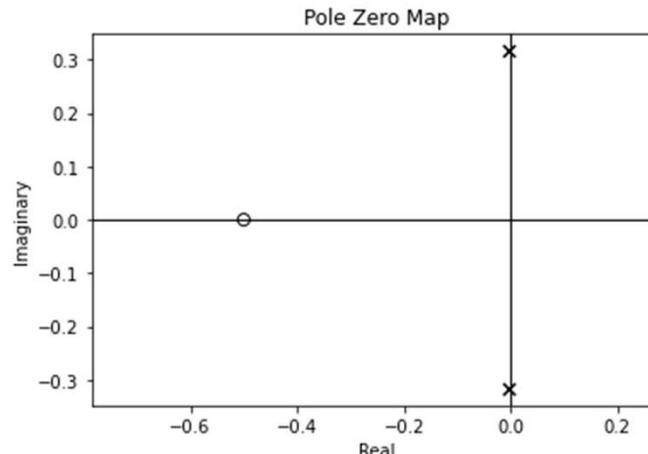
Out[2]: 
$$\frac{s + 0.5}{10s^2 + 0.1s + 1}$$

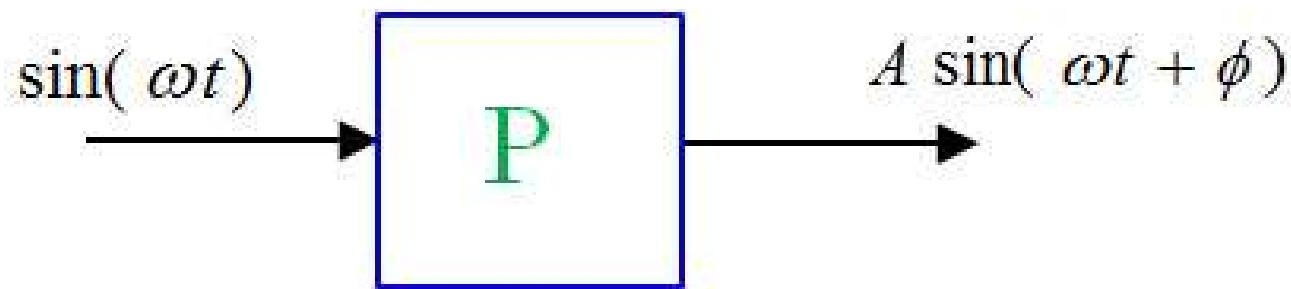

In [3]: num = [1, 0.5]
den = [10, 0.1, 1]
P = ctl.tf(num,den)
P

Out[3]: 
$$\frac{s + 0.5}{10s^2 + 0.1s + 1}$$

```

```
In [4]: ctl.pzmap(P)
Out[4]: (array([-0.005+0.31618824j, -0.005-0.31618824j]), array([-0.5]))
```





A = magnitude (gain)

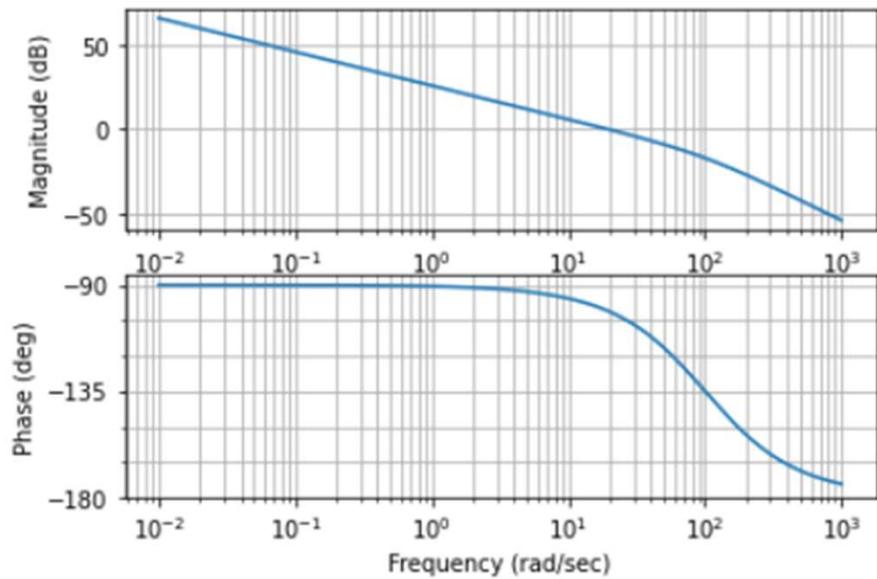
ϕ = phase

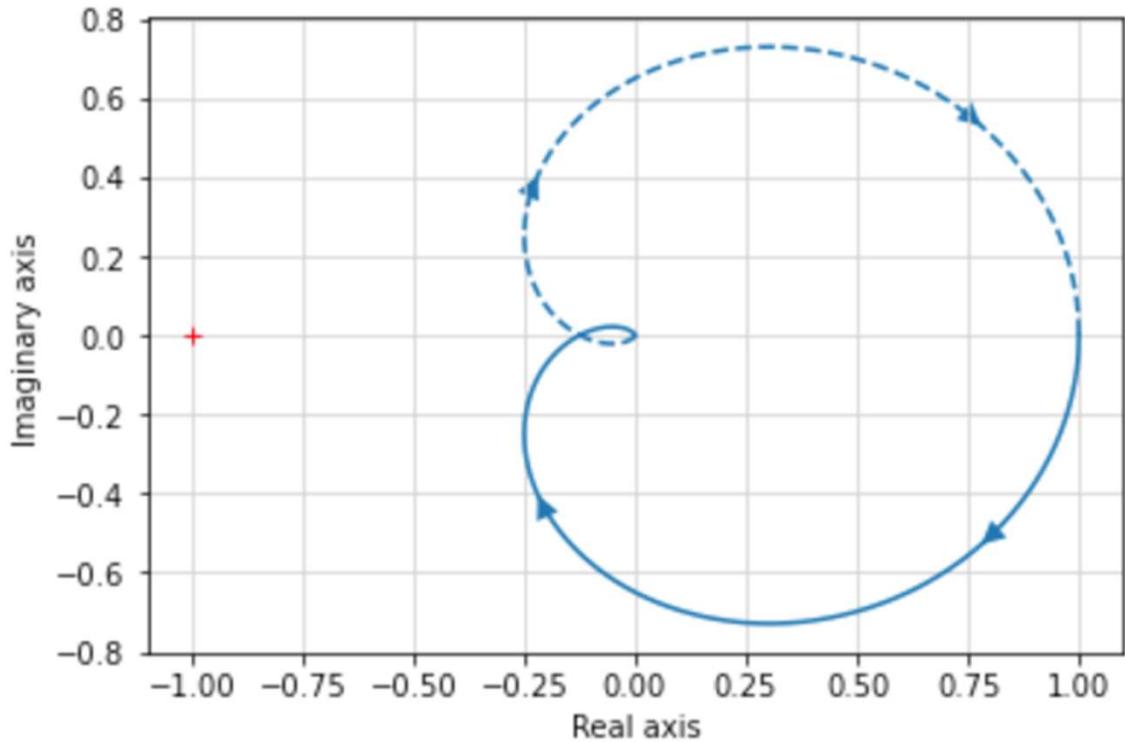
$$P(j\omega) = P(s)\Big|_{s=j\omega}$$

ผลตอบสนองความถี่ของระบบ LTI

แผนภาพบ่อเด (Bode plot)

```
In [21]: _, _, _ = ctl.bode_plot(L, dB=True, omega_limits=(0.01, 1000))
```





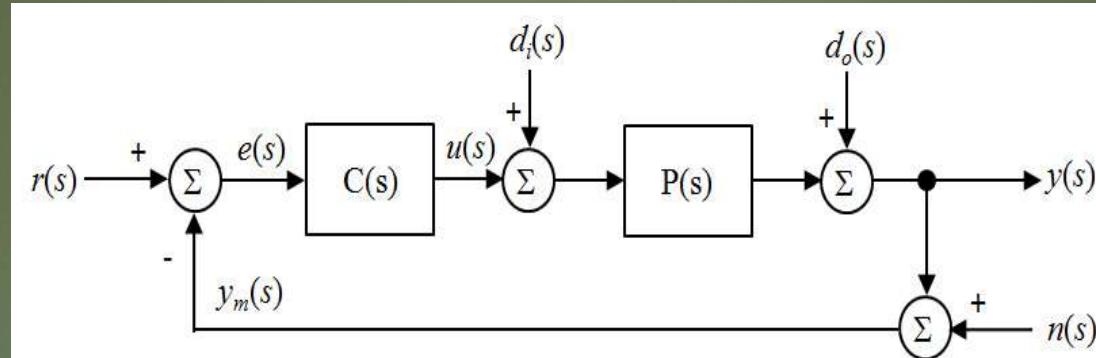
```
s = ctl.tf('s')
P = 1/(s+1)**3
_ = ctl.nyquist_plot(P)
```

แผนภาพในควิสต์
(Nyquist plot)

คุณสมบัติการป้อนกลับ (feedback properties)

- ▶ เสถียรภาพ (stability)
- ▶ สมรรถนะ (performance)
- ▶ การตามรอย (tracking)
- ▶ การลดทอนการรบกวน (disturbance & noise attenuation)
- ▶ ข้อกำหนดอื่นที่ผู้ใช้ต้องการ เช่น ความคงทนต่อการพิດพลาดของโมเดล
การจำกัดขนาดเอ่าต์พูตตัวควบคุม ฯลฯ

เสถียรภาพของระบบป้อนกลับ SISO

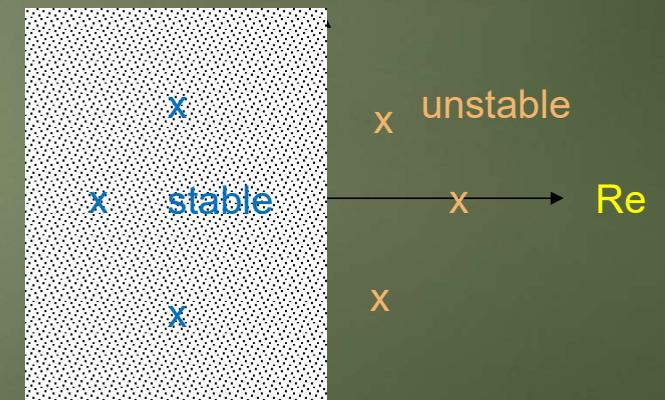


Im

ตรวจสอบ poles ของฟังก์ชันถ่ายโอนวงปิด S หรือ T
ว่าอยู่ท่า哪ด้านซ้ายของระนาบเชิงซ้อนหรือไม่?

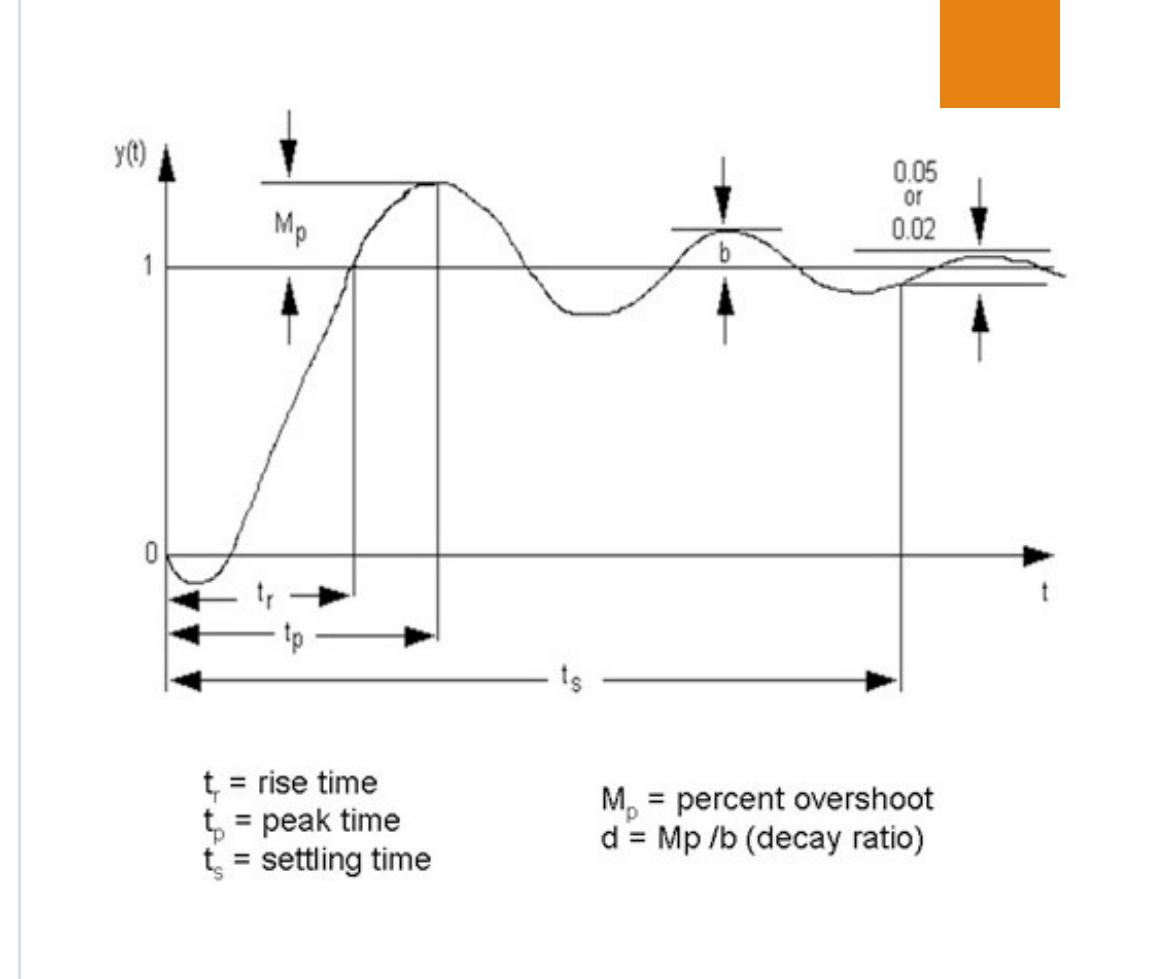
Stable: โพลทุกตัวต้องอยู่ด้านซ้ายของระนาบ

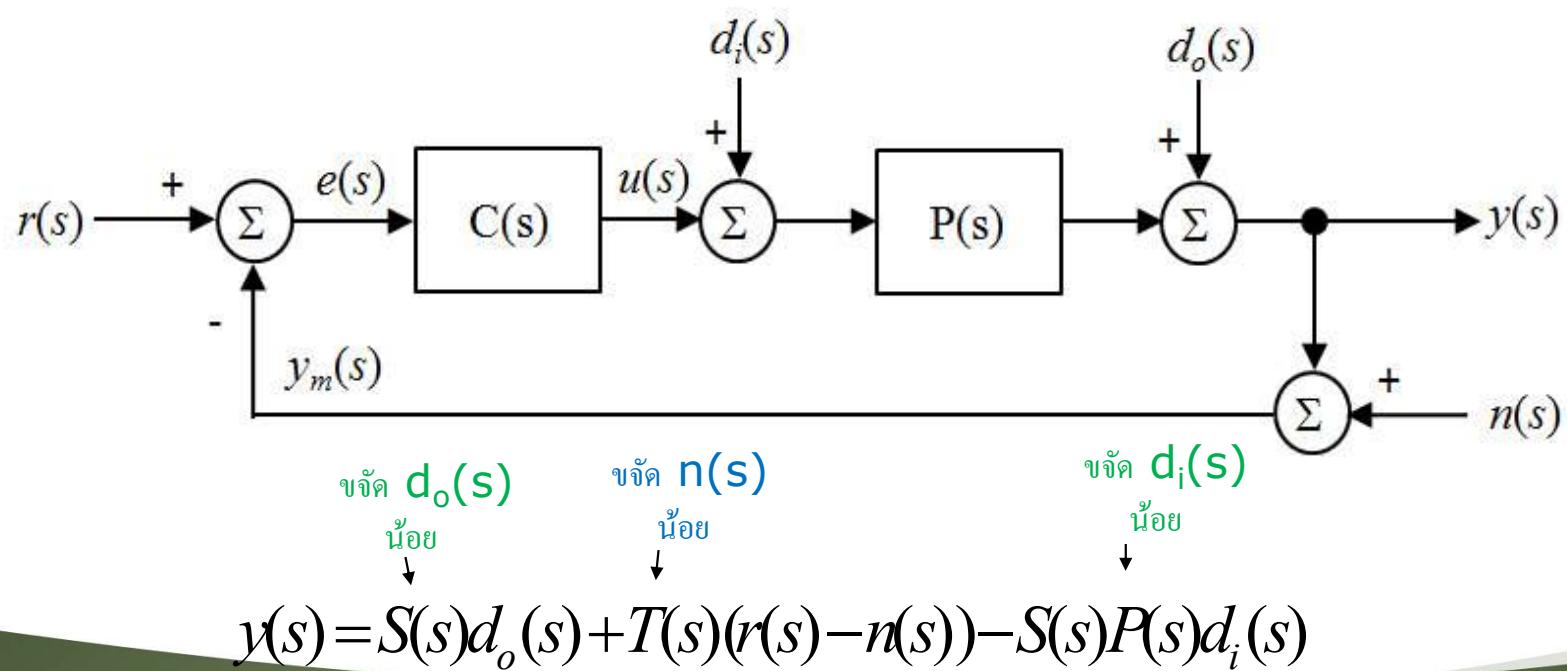
Unstable: มีโพลอย่างน้อยหนึ่งตัวอยู่ท่า哪ด้านขวาของระนาบ



สมรรถนะการตามรอย

- ▶ ผลตอบสนองต่อสัญญาณ
ขั้นบันได (step response)

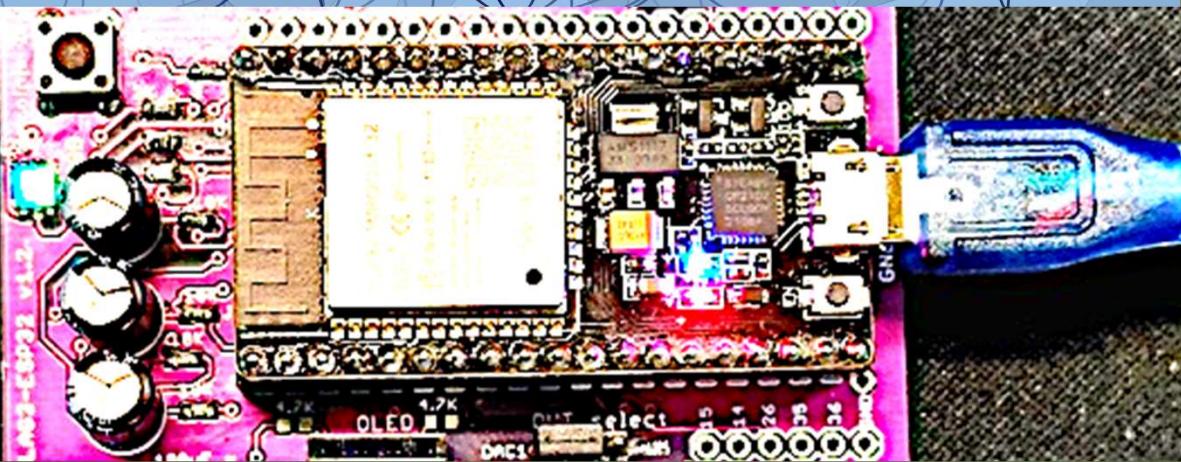




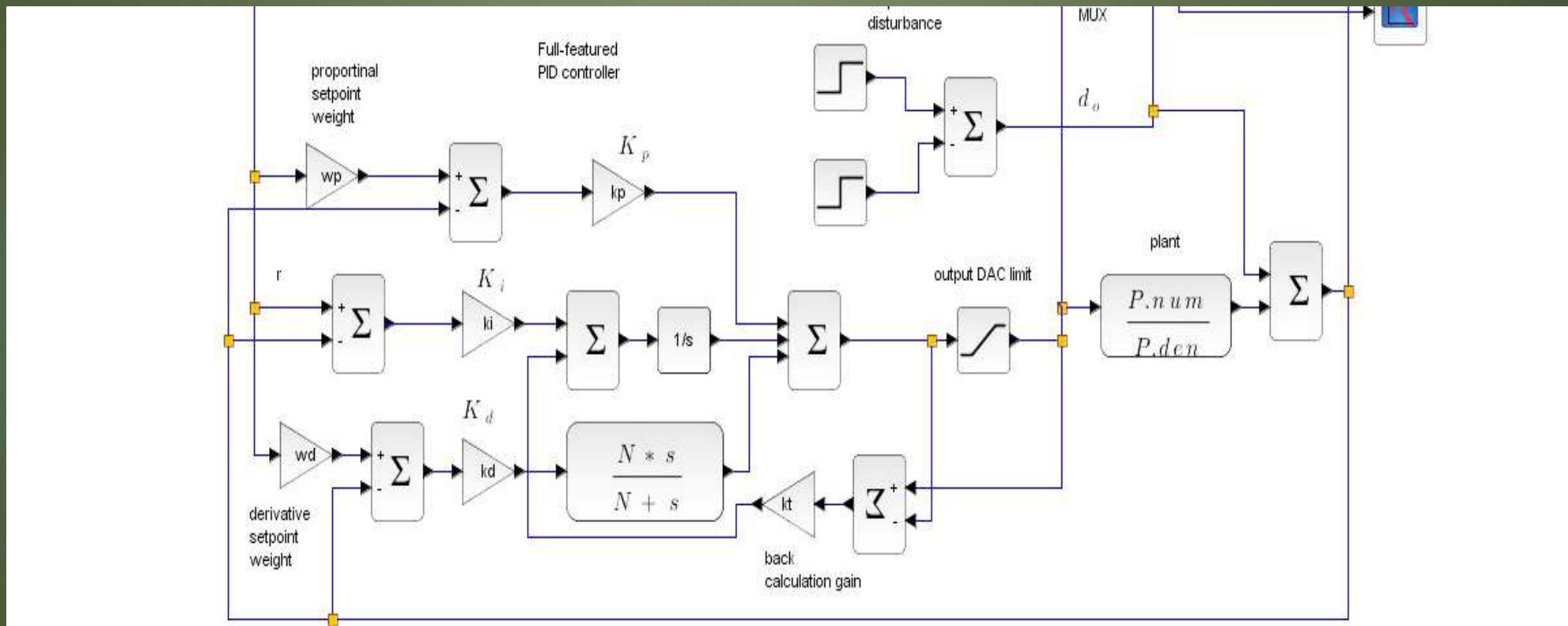
สมรรถนะการลดTHONการรบกวน



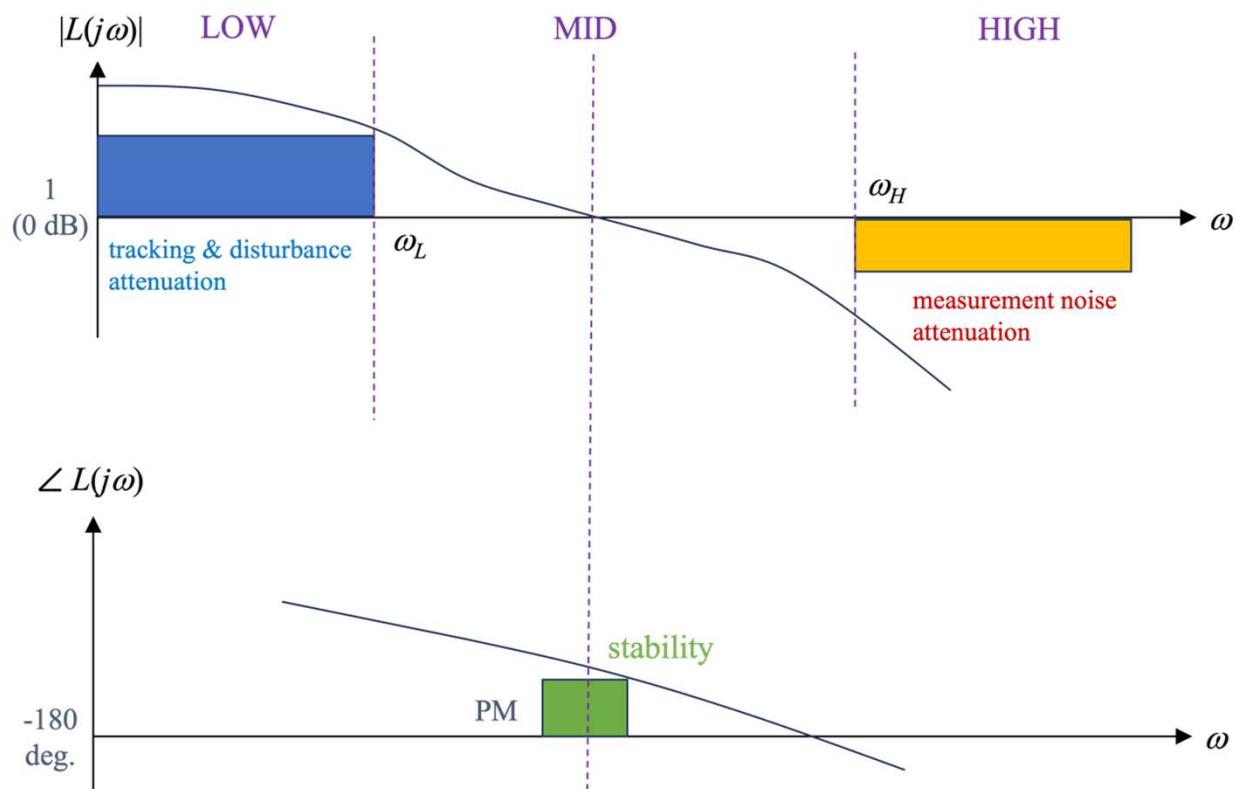
ตัวควบคุมเชิง
เส้นที่ใช้เป็น[†]
ตัวอย่างบนบอร์ด
LAG3



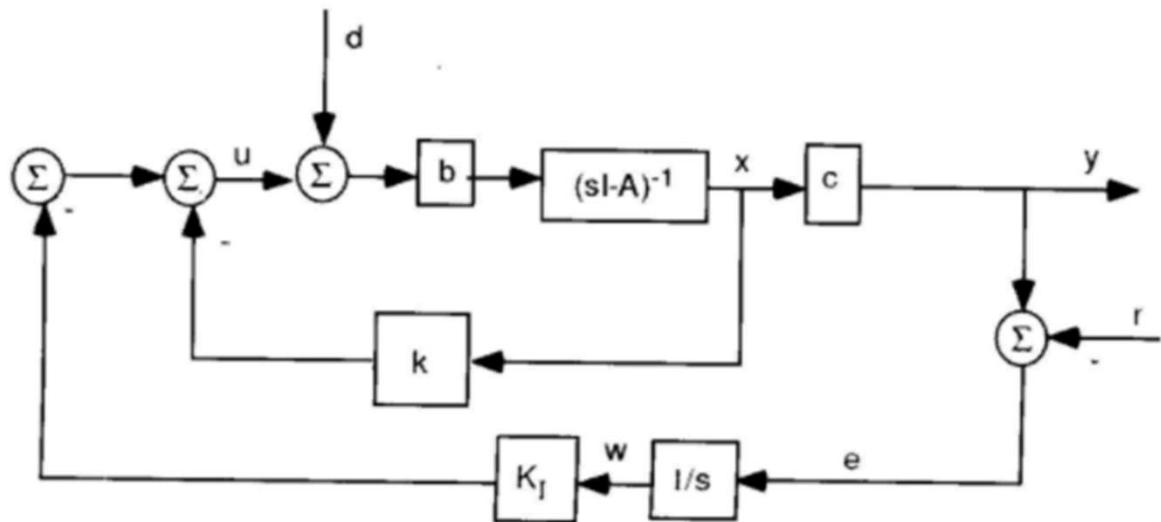
1. ตัวควบคุม PID พร้อมฟังก์ชันเสริม

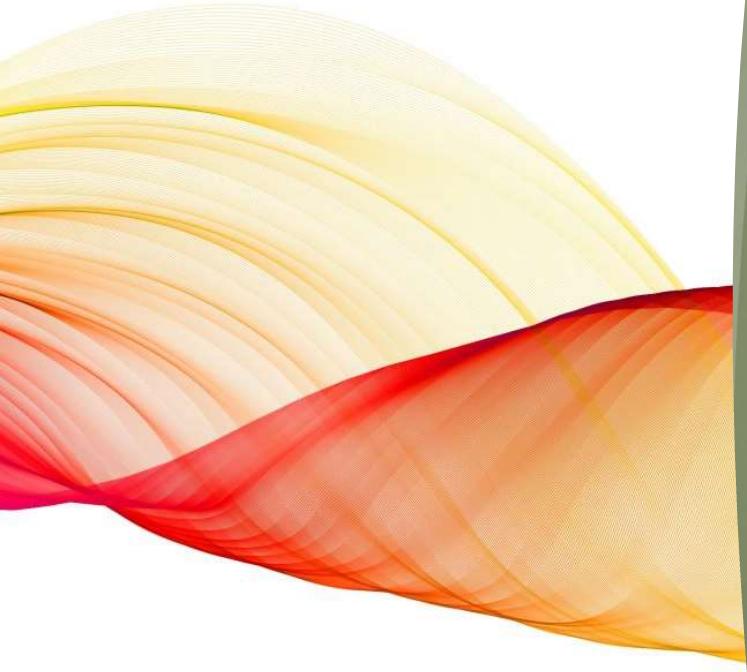


2. ตัวควบคุมที่ ออกแบบโดยวิธี loopshaping



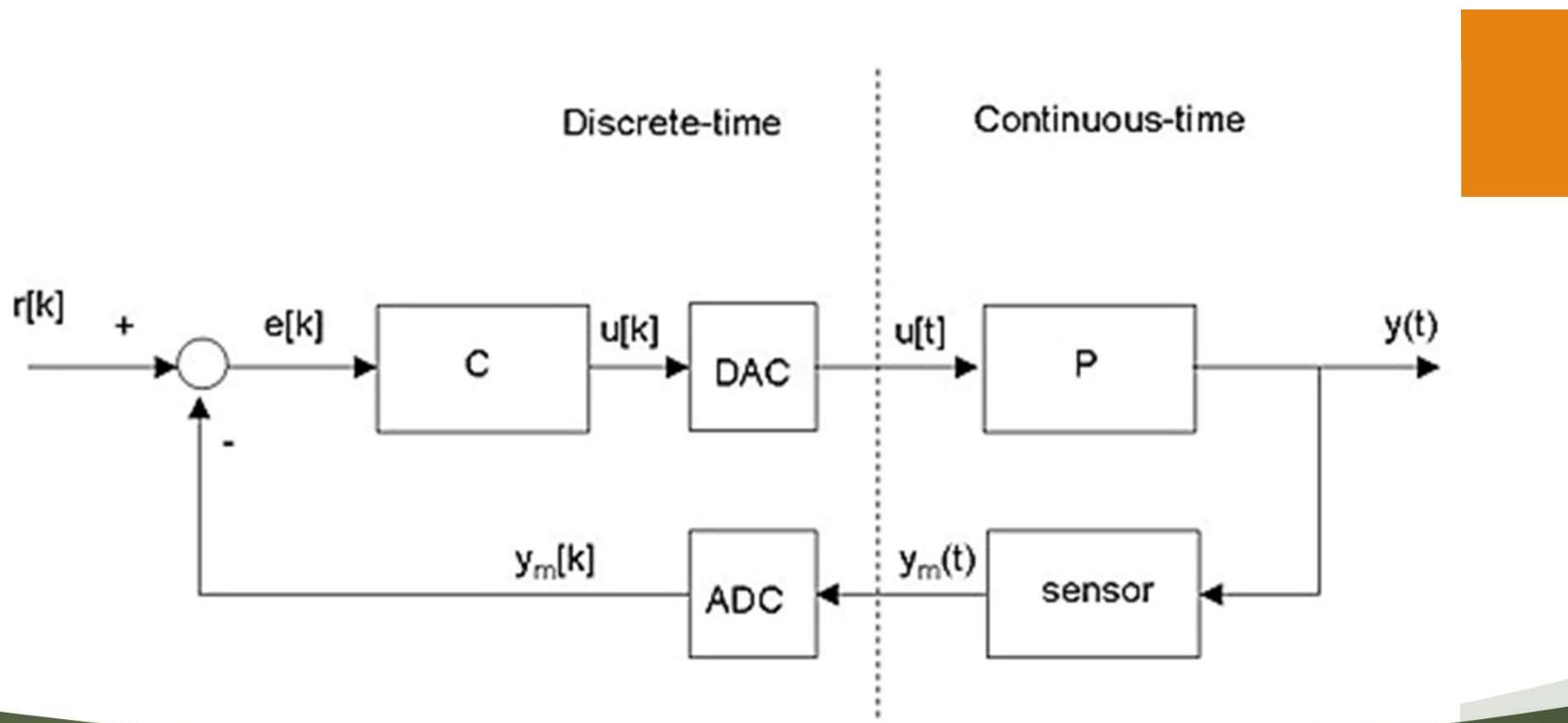
3. ตัวควบคุม ป้อนกลับสถานะ + ตัวปริพันธ์





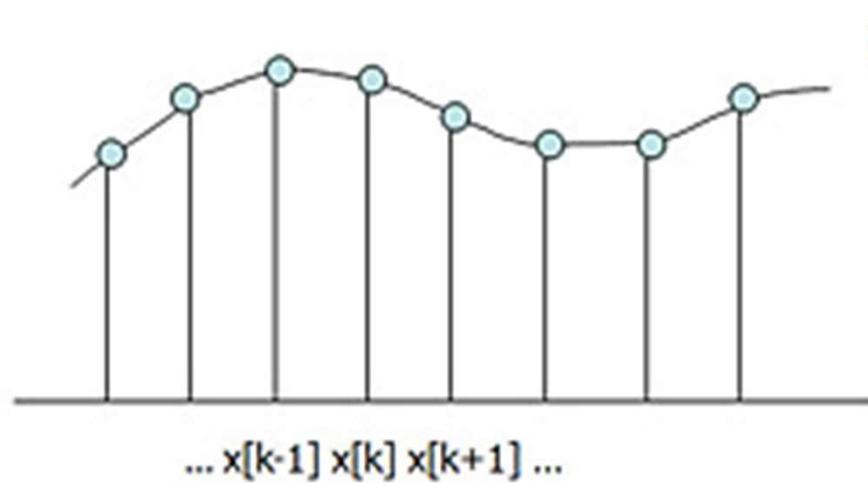
การอิมเพลเม้นต์ตัวควบคุม

Controller implementation



ระบบควบคุมแบบผสม

การสุ่มสัญญาณต่อเนื่อง



$x(t)$

$$x[k] = x(k\Delta)$$

Δ = sampling period

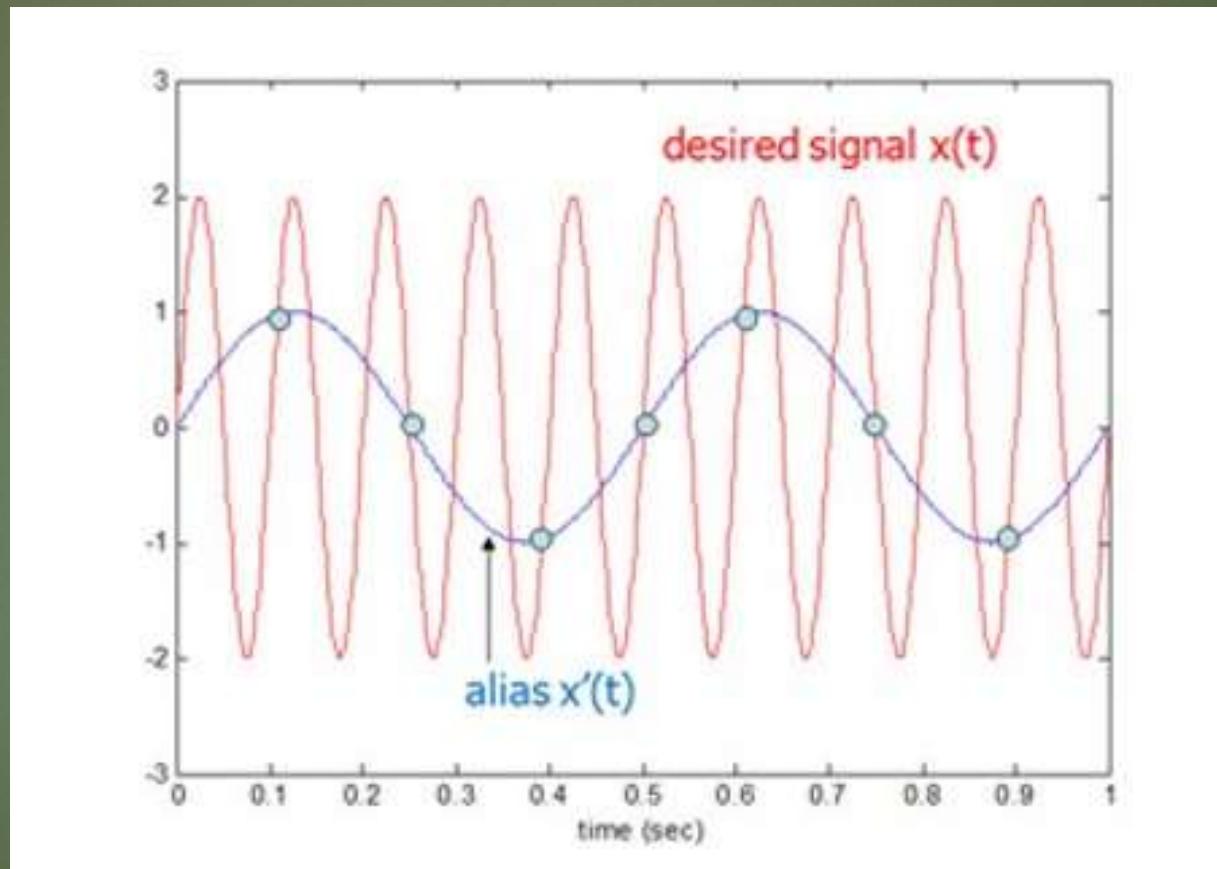


pointer

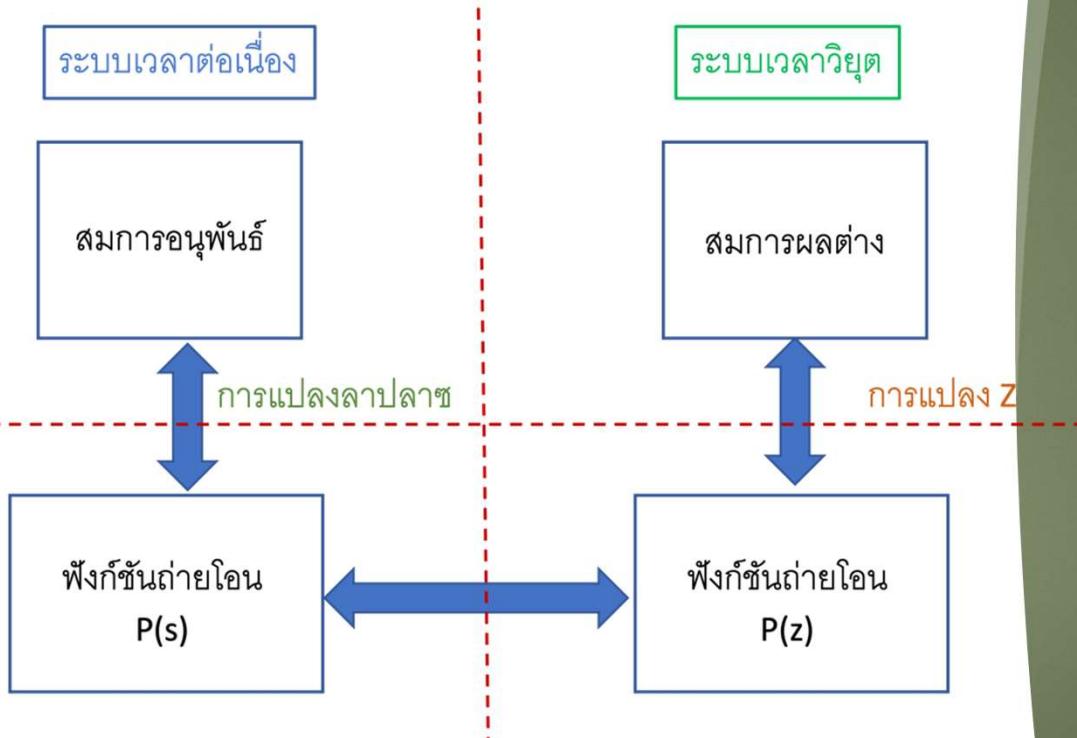
$x[0]$
$x[1]$
$x[2]$
$x[3]$
$x[4]$
:

memory

ปัญหาการแฝง



ความสัมพันธ์ของ ระบบเวลาต่อเนื่อง และระบบเวลาวิบูร্ধ



การประมาณค่าระบบเวลาต่อเนื่อง 3 วิธี

- ▶ วิธีผลต่างข้างหน้า (forward difference)
- ▶ วิธีผลต่างย้อนหลัง (backward difference)
- ▶ วิธีแปลงเชิงเส้นคู่ (bilinear transformation)



ชื่อเรียกอื่นคือ “การประมาณเชิงลี่เหลี่ยมคางหมู” (trapezoidal approximation)
หรือ Tustin

การประมาณค่าตัวปริพันธ์

$$\dot{u}(t) = e(t)$$

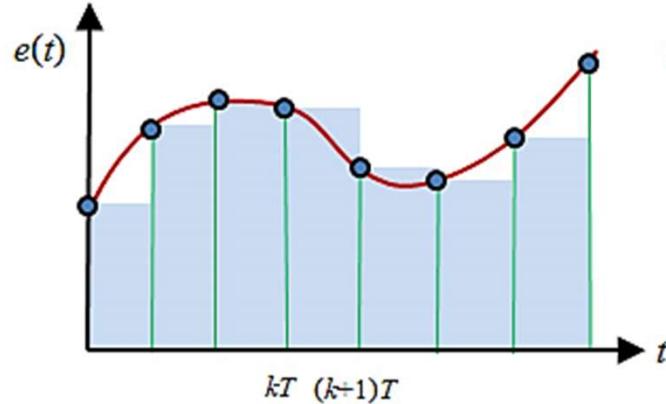
$$C(s) = \frac{U(s)}{E(s)} = \frac{1}{s}$$

$$u(t) = u(t_0) + \int_{t_0}^t e(\tau) d\tau$$

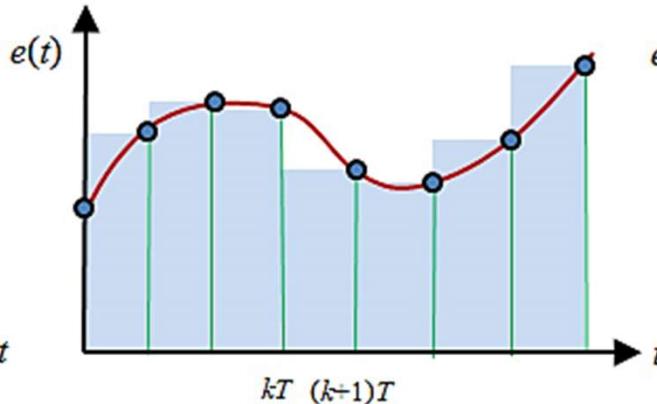
ต้องการประมาณค่า



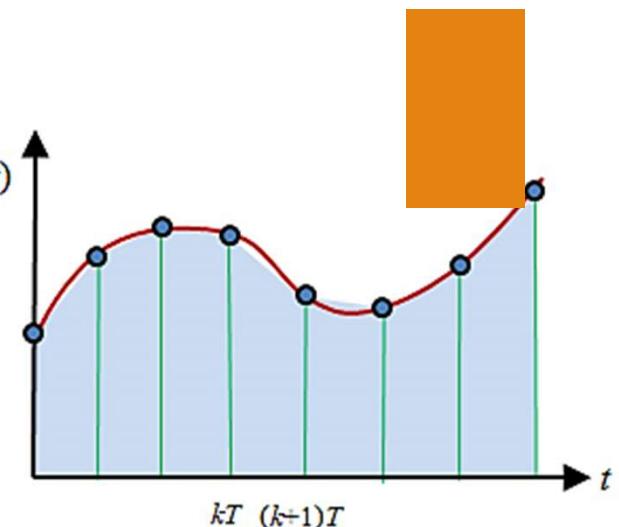
$$u((k+1)T) = u(kT) + \int_{kT}^{(k+1)T} e(\tau) d\tau$$



Forward Difference



Backward Difference



Bilinear Transform

FD $u(k+1) \approx u(k) + e(k)T$

การประมาณค่าปริพันธ์ 3 วิธี

BD $u(k+1) \approx u(k) + e(k+1)T$

BT $u(k+1) \approx u(k) + \left(\frac{e(k+1) + e(k)}{2} \right) T$

ความสัมพันธ์ในการประมาณค่า 3 วิธี

FD

$$s \rightarrow \frac{z - 1}{T}$$

BD

$$s \rightarrow \frac{z - 1}{Tz}$$

BT

$$s \rightarrow \frac{2}{T} \frac{(z-1)}{(z+1)}$$

ตัวอย่าง 3.13 พิจารณาตัวควบคุม PID เวลาต่อเนื่องที่ใช้ตัวกรองแทนพจน์อนุพันธ์ (3.35)

$$C(s) = K_p + \frac{K_i}{s} + \frac{NK_d}{1 + N/s}$$

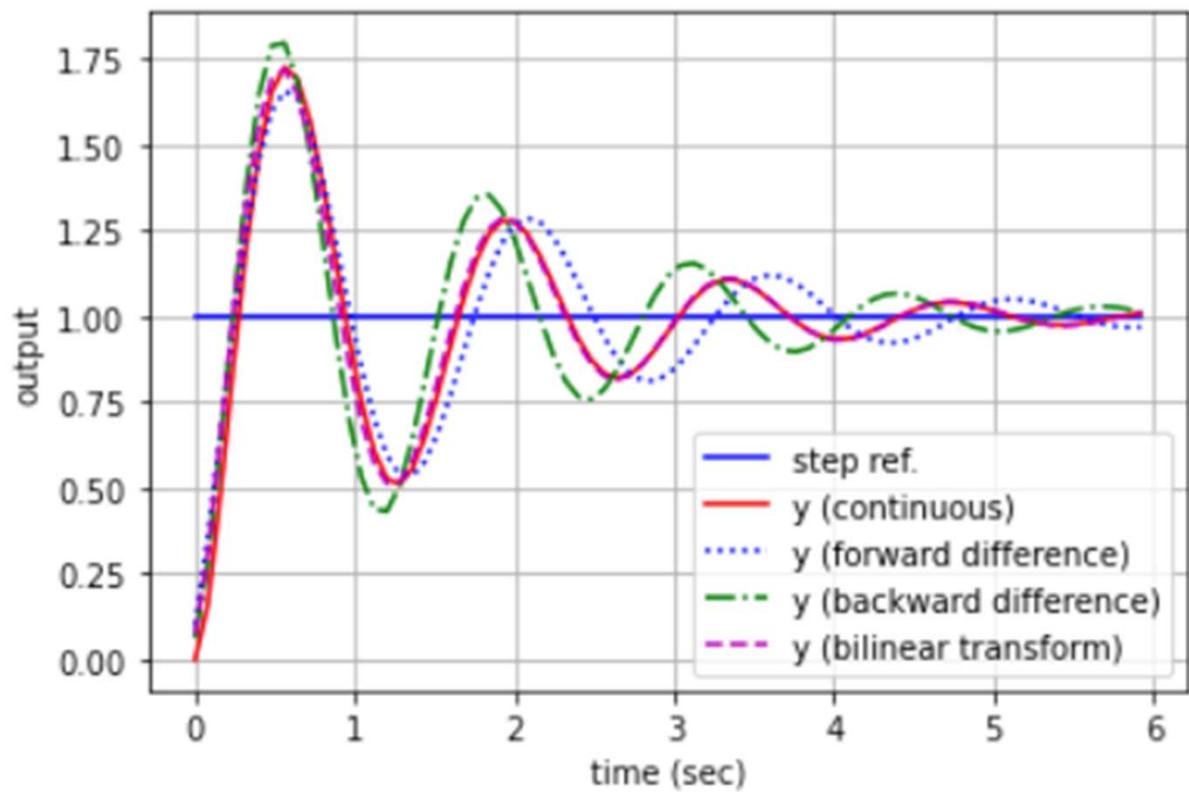
เมื่อกำหนดค่าบเวลาการสั่น T จะได้ตัวควบคุมเวลาวิถุตจากการประมาณค่า 3 วิธีดังนี้

FD: $C(z) = K_p + \frac{K_i T}{z-1} + \frac{K_d N(z-1)}{z-1 + NT} \quad (3.72)$

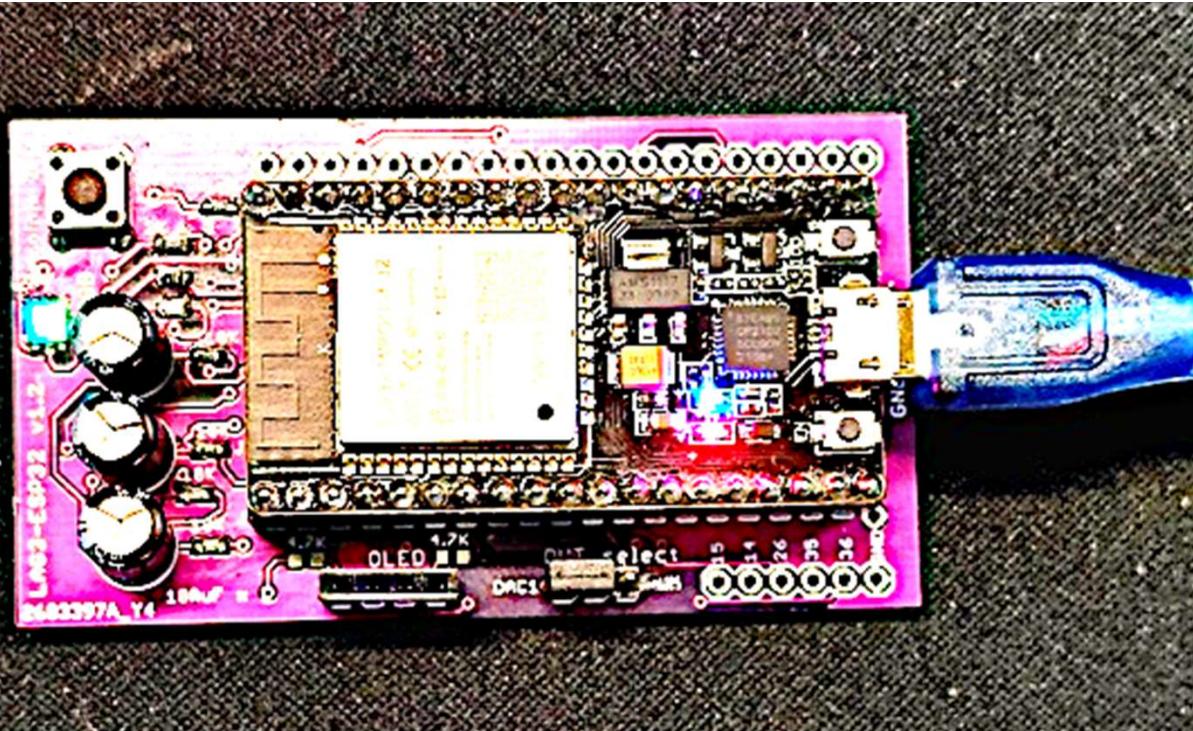
BD: $C(z) = K_p + \frac{K_i T z}{z-1} + \frac{K_d N(z-1)}{(1+NT)z-1} \quad (3.73)$

BT: $C(z) = K_p + \frac{0.5 K_i T (z+1)}{z-1} + \frac{K_d N(z-1)}{(1+0.5NT)z+0.5NT-1} \quad (3.74)$

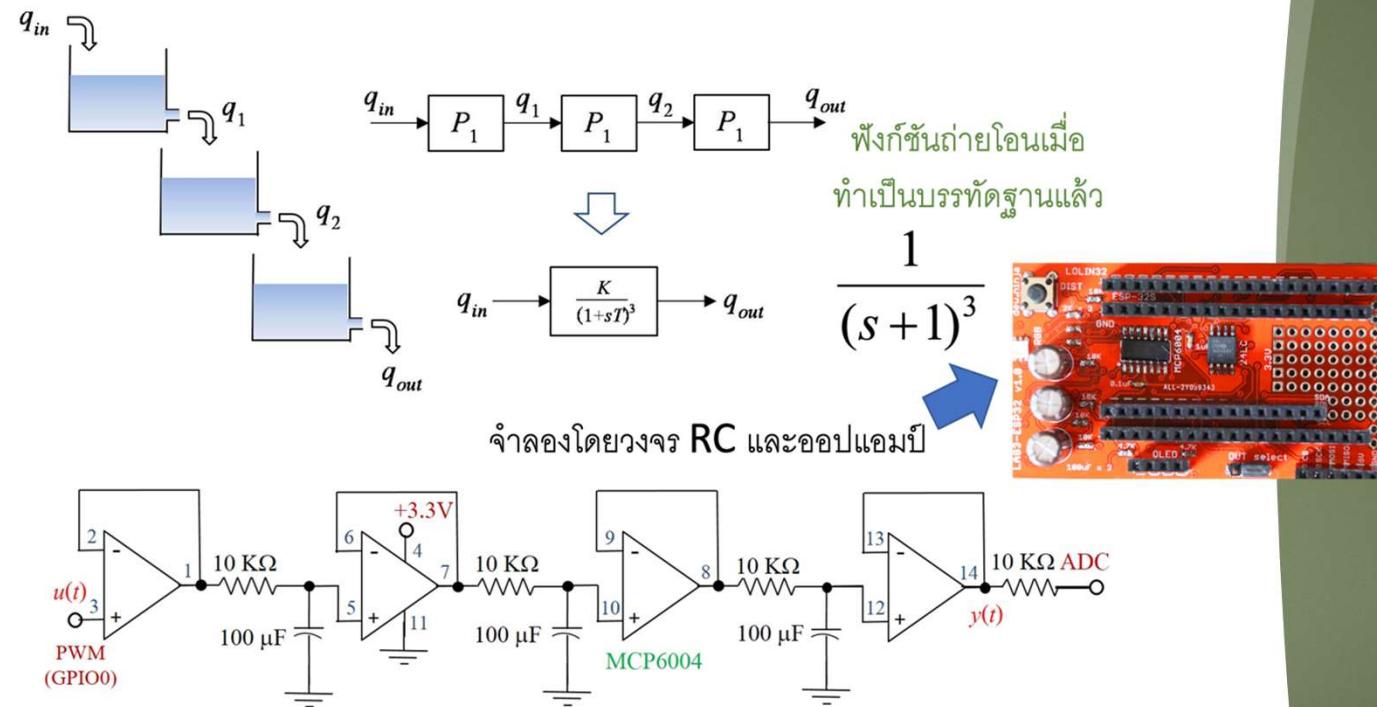
ปริยบเทียบ
ผลตอบสนอง
ขั้นบันได
($T_s = 0.08$ sec)

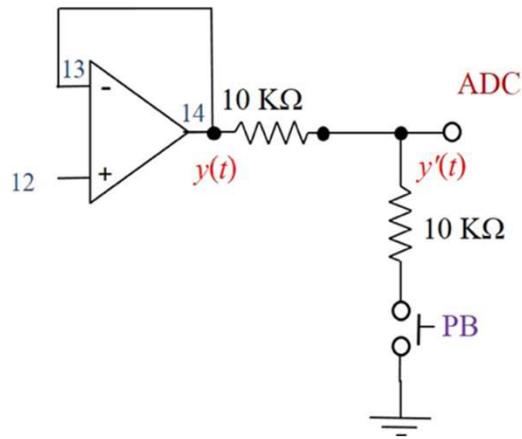
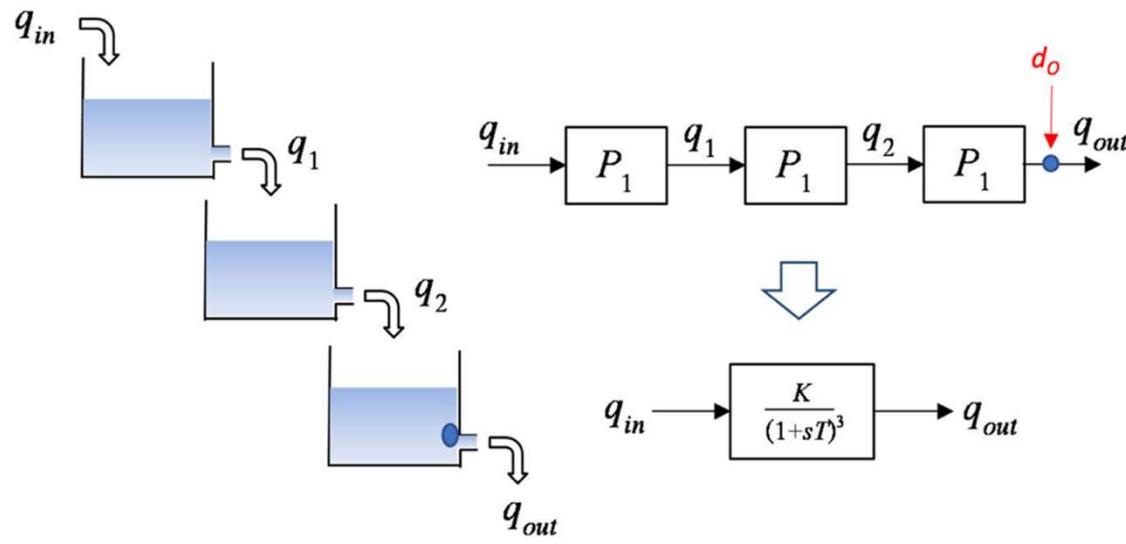


บอร์ดจำลองถัง
น้ำ 3 ระดับ
(LAG3-ESP32)



การจำลองระบบ ถึงน้ำ 3 ระดับ โดยวงจร อิเล็กทรอนิกส์

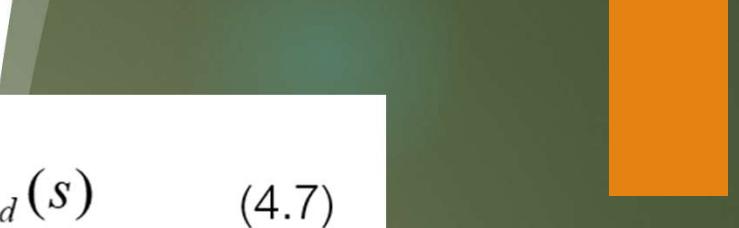




สวิตซ์จำลองการ
รับกวนที่เอกสารพุต

ตัวควบคุม 3 ประเภทที่อิมพลิเมนต์บน LAG3

- ▶ controller = 0: Full featured PID controller
- ▶ controller=1: custom lead-lag controller
- ▶ controller=2: state-feedback+integrator


$$u(s) = K_p e_p(s) + \frac{K_i}{s} e(s) + \frac{K_t}{s} e_{us}(s) + \frac{K_d N_s}{s + N} e_d(s) \quad (4.7)$$

$$e = r - y$$

$$e_p = W_p r - y$$

$$e_d = W_d r - y$$

$$e_{us} = u_{SAT} - u$$


$$u = u_p + u_i + u_d \quad (4.12)$$



Full-featured PID controller

$$u = u_p + u_i + u_d \quad (4.12)$$

$$u_p(k) = K_p e_p(k) \quad (4.17)$$

$$u_i(k) = u_i(k-1) + b_i(e(k) + e(k-1)) + b_t(e_{us}(k) + e_{us}(k-1)) \quad (4.18)$$

$$u_d(k) = a_d u_d(k-1) + b_d(e_d(k) - e_d(k-1)) \quad (4.19)$$

โดยมีค่าสัมประสิทธิ์

$$b_i = 0.5TK_i \quad (4.20)$$

$$b_t = 0.5TK_t \quad (4.21)$$

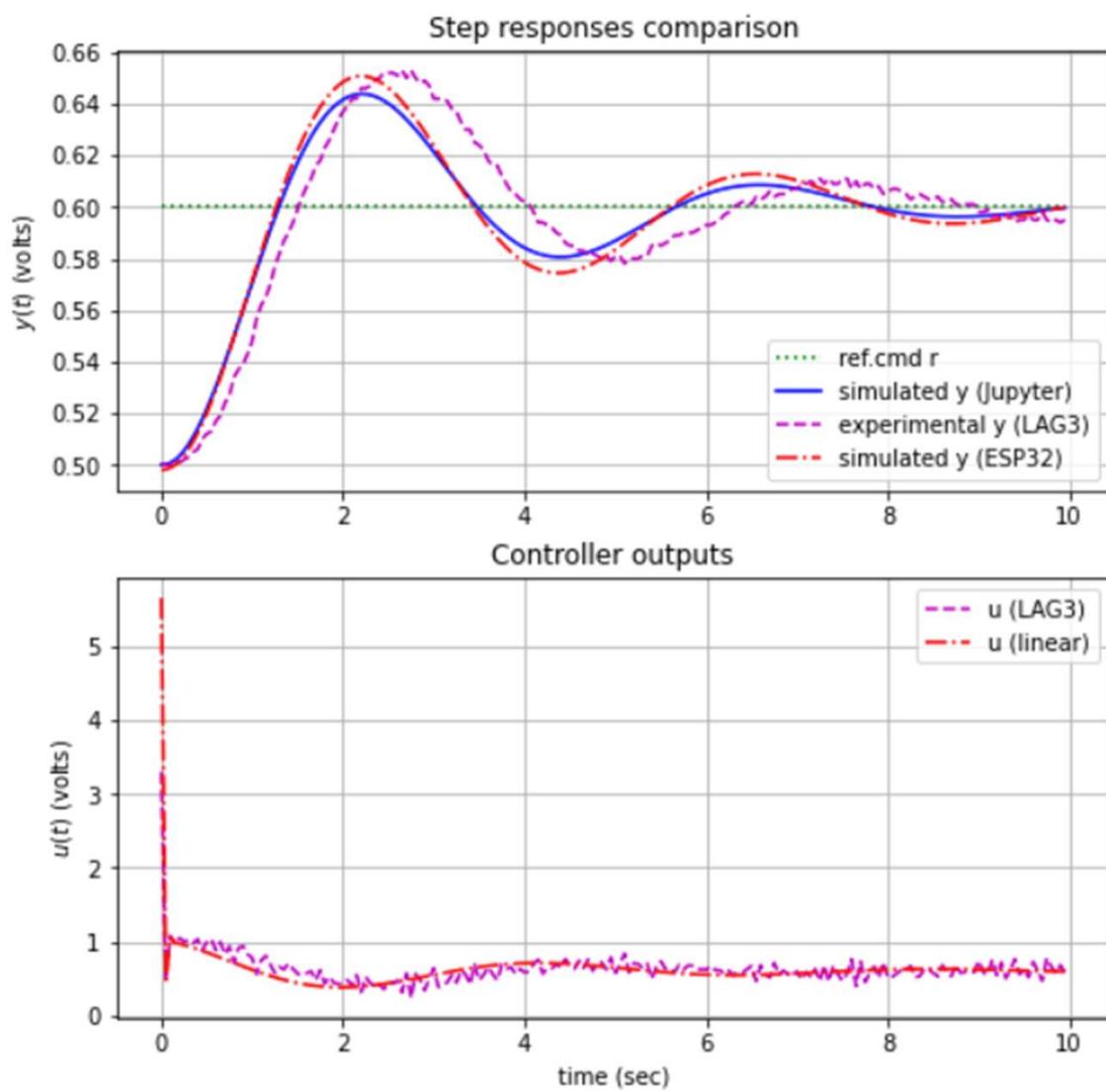
$$a_d = \frac{-(0.5NT - 1)}{(1 + 0.5NT)} \quad (4.22)$$

$$b_d = \frac{K_d N}{(1 + 0.5NT)} \quad (4.23)$$

Discrete-time full-featured PID

ใช้การแปลงเชิงเส้นคู่

$$s = \frac{2(z-1)}{T(z+1)}$$

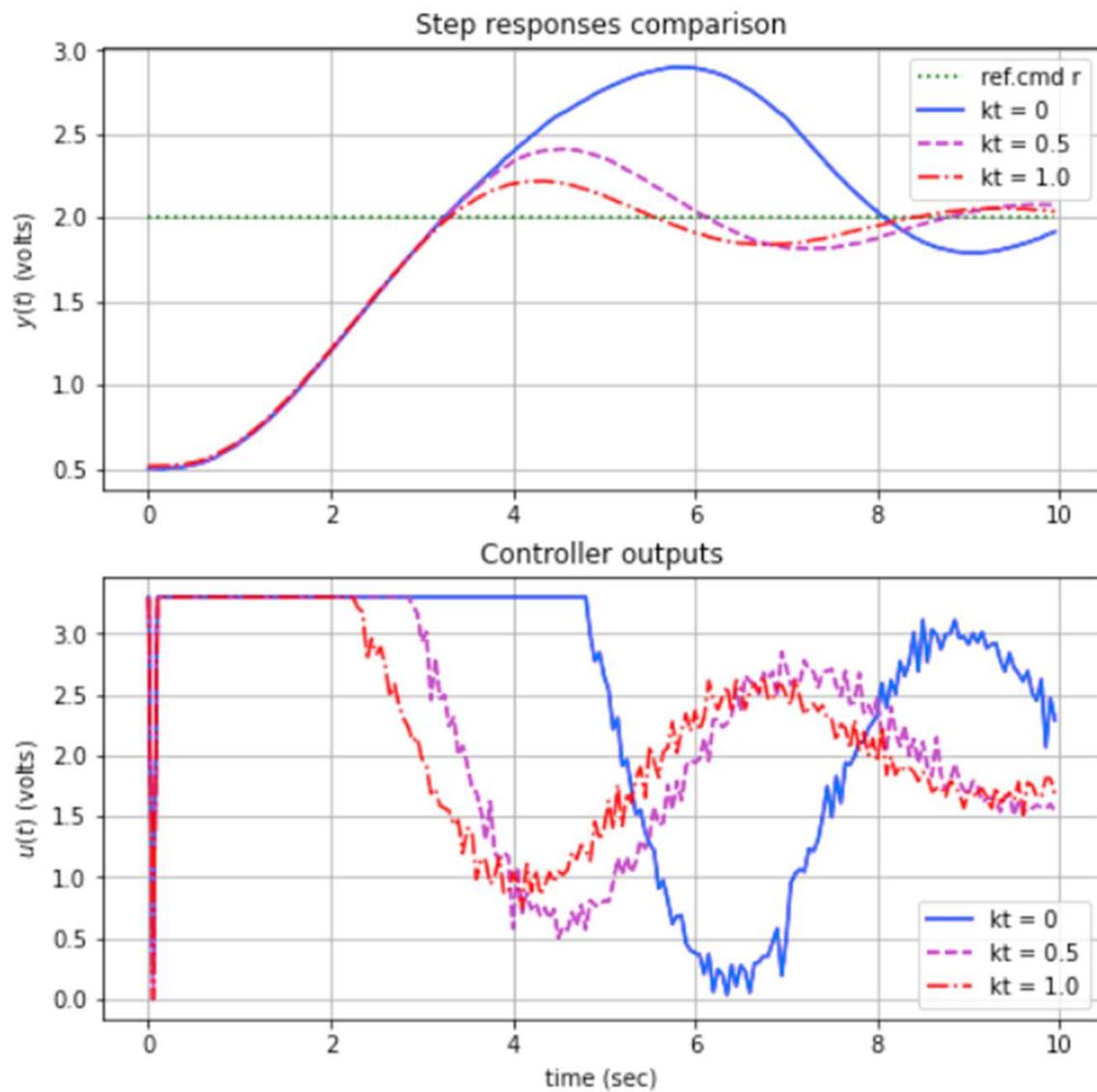


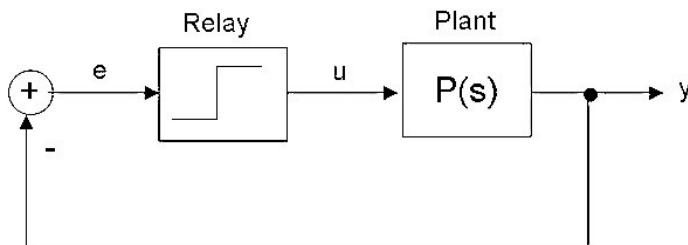
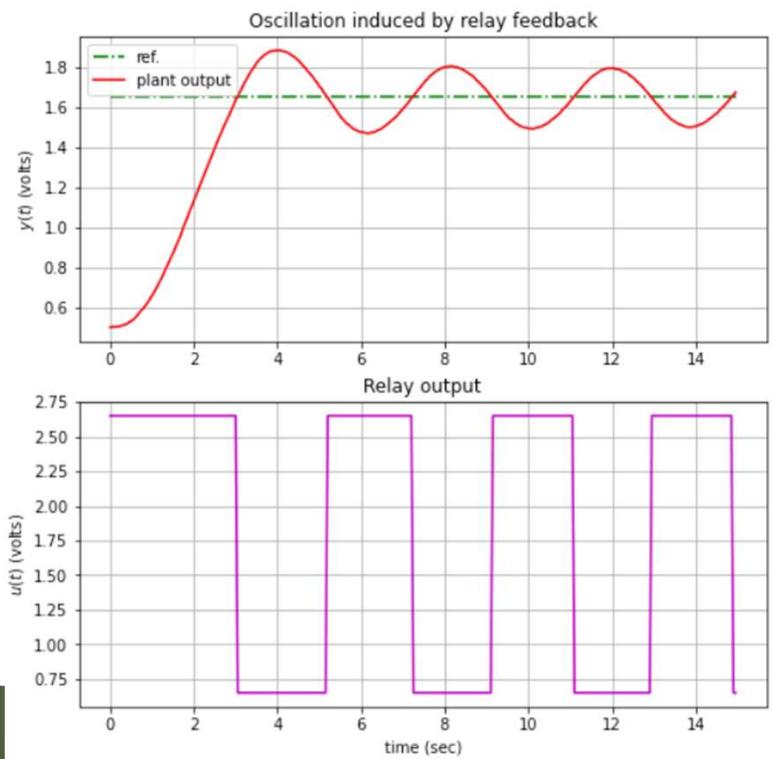
ເປົ້າຍບເທີຍບ

ຜລຕອບສນອງ

ຂຶ້ນບັນໄດ

การปรับค่า k_t เพื่อลด
ผลกระทบจากการสะสม
ค่าผิดพลาด
(antiwindup)





>>> %Run -c \$EDITOR_CONTENT

```

Enter command : autotune
Autotuning starts ...
Oscillation cycle : 1
Oscillation cycle : 2
Oscillation cycle : 3
Oscillation cycle : 4
Tu = 4.100011
Ku = 7.808143
Autotuning ends.
Updating PID parameters with Ku and Tu
kp = 4.684886
ki = 2.285304
kd = 2.401011
    
```

Enter command : |

คำสั่ง autotune สำหรับปรับแต่ง PID อัตโนมัติ

ตัวควบคุมป้อนกลับເອົາຕີພຸດທີ່ອອກແບບໂດຍ
loopshaping

$$C(s) = \frac{K(s+z)}{s(s+p)}$$

ตัวปริพันธ์

$$C(s) = \frac{1}{s}$$

$$C(z) = \frac{u(z)}{e(z)} = 0.5T \frac{(1+z^{-1})}{(1-z^{-1})} \quad (4.25)$$

เมื่อแปลงเป็นสมการผลต่างจะได้เป็น

$$u(k) = u(k-1) + 0.5T(e(k) + e(k-1)) \quad (4.26)$$

ตัวนำหน้า-ล้าหลัง

$$C(s) = \frac{s+a}{s+b}$$

รายละเอียดที่ใช้เป็นแบบฝึกหัดว่า เมื่อแทนค่า $s = \frac{2(z-1)}{T(z+1)}$ จะได้เป็น

$$C(z) = \frac{u(z)}{e(z)} = \frac{(\alpha_1 + \alpha_2 z^{-1})}{(\beta_1 + \beta_2 z^{-1})} \quad (4.28)$$

โดย

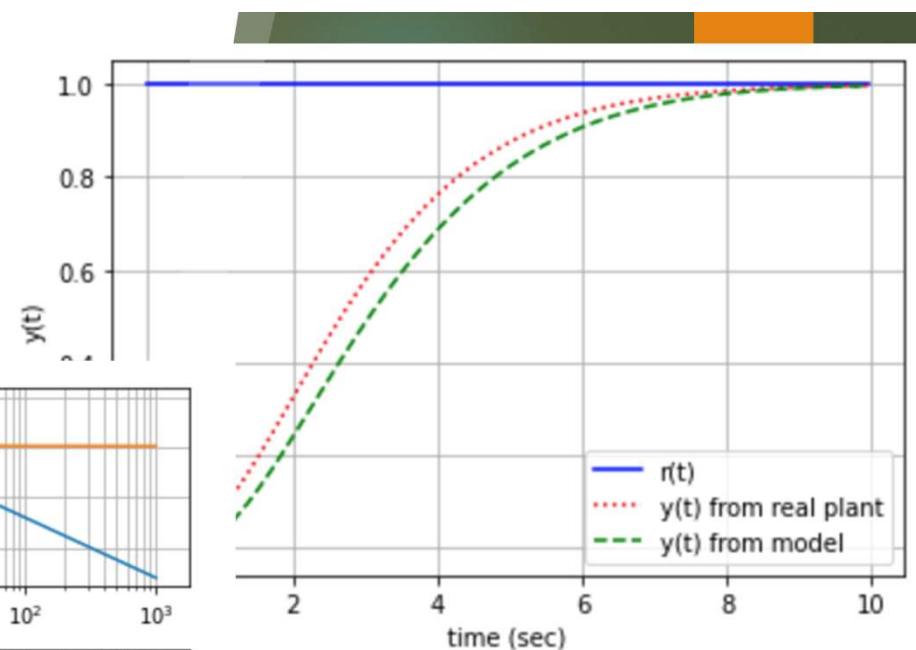
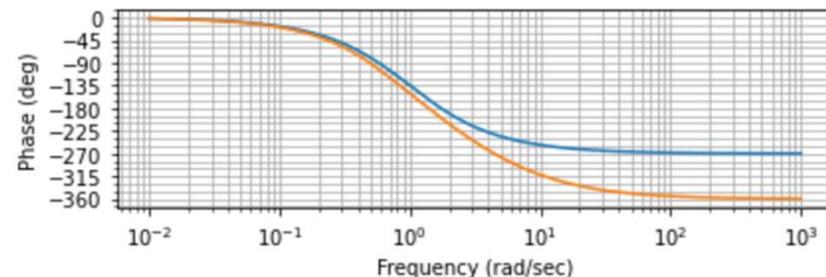
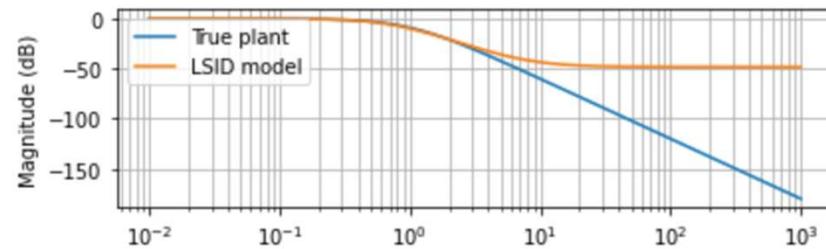
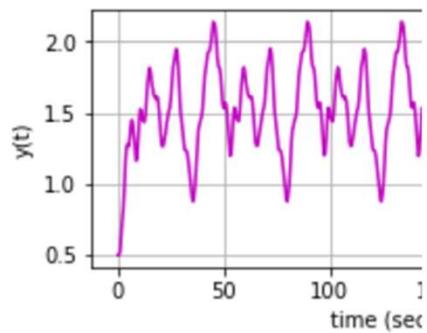
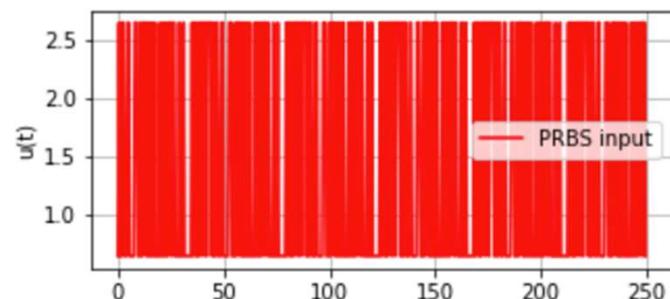
$$\alpha_1 = 2 + aT, \quad \alpha_2 = aT - 2, \quad \beta_1 = 2 + bT, \quad \beta_2 = bT - 2, \quad (4.29)$$

เมื่อแปลงเป็นสมการผลต่างจะได้เท่ากับ

$$u(k) = \gamma_1 u(k-1) + \gamma_2 e(k) + \gamma_3 e(k-1) \quad (4.30)$$

โดยมีสัมประสิทธิ์

$$\gamma_1 = \frac{-\beta_2}{\beta_1}, \quad \gamma_2 = \frac{\alpha_1}{\beta_1}, \quad \gamma_3 = \frac{\alpha_2}{\beta_1} \quad (4.31)$$



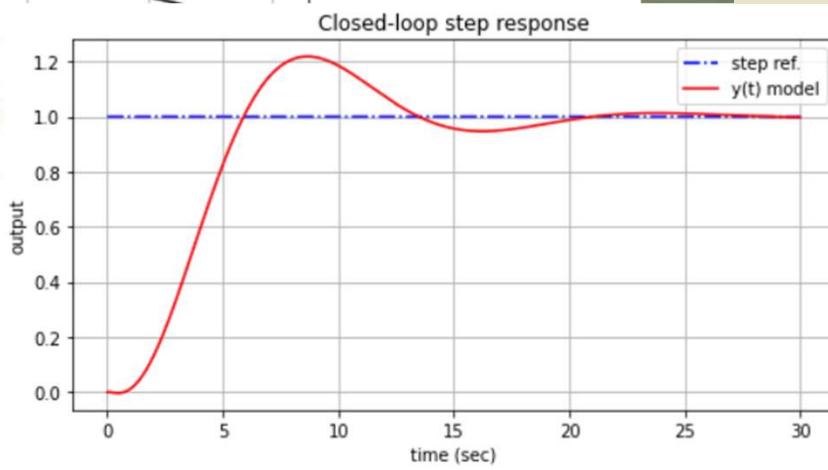
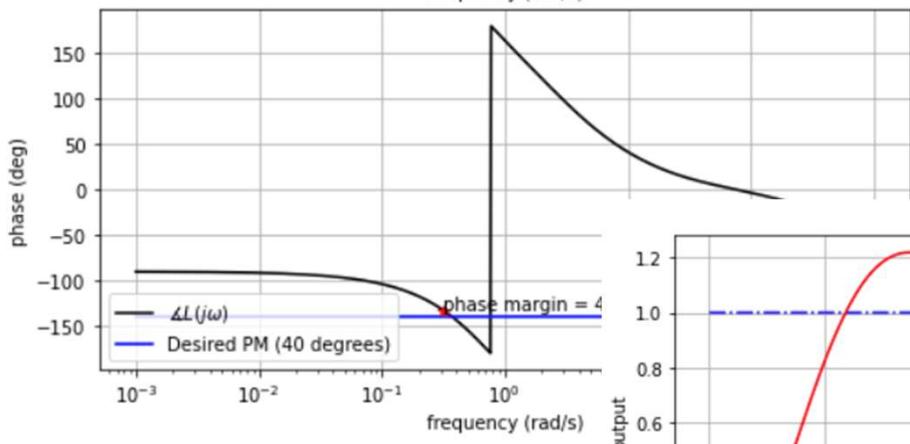
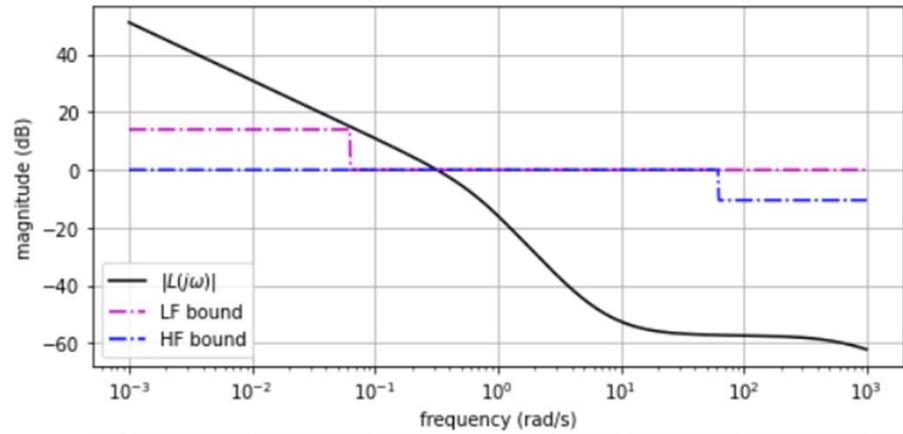
P_id

$$\begin{array}{r} -0.001856s^3 + 0.01114s^2 + 0.1043s - 0.4766 \\ \hline -0.4852s^3 - 1.651s^2 - 1.513s - 0.4773 \end{array}$$

ห้ามเดลโดยวิธี
Least-square
identification

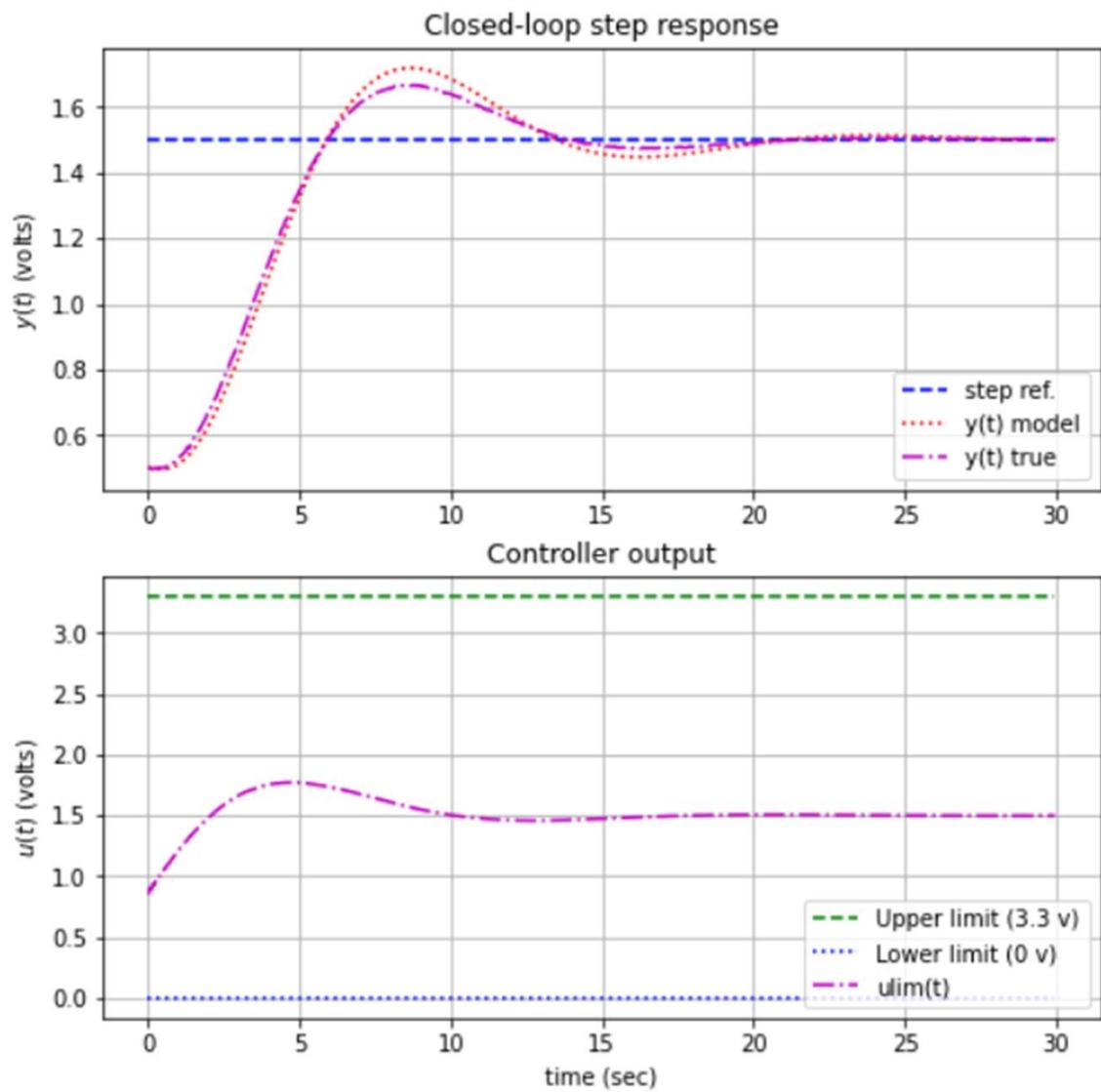
Loopshaping design

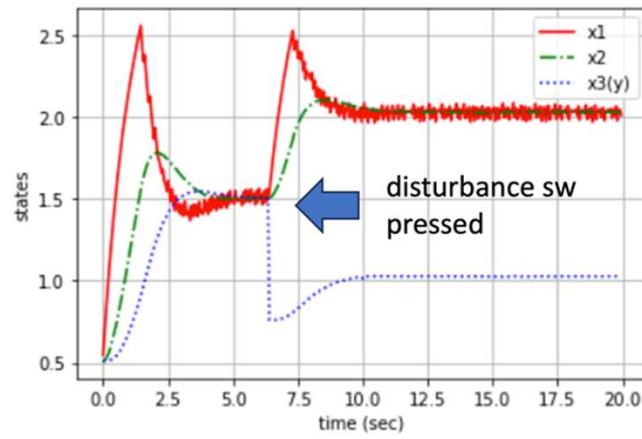
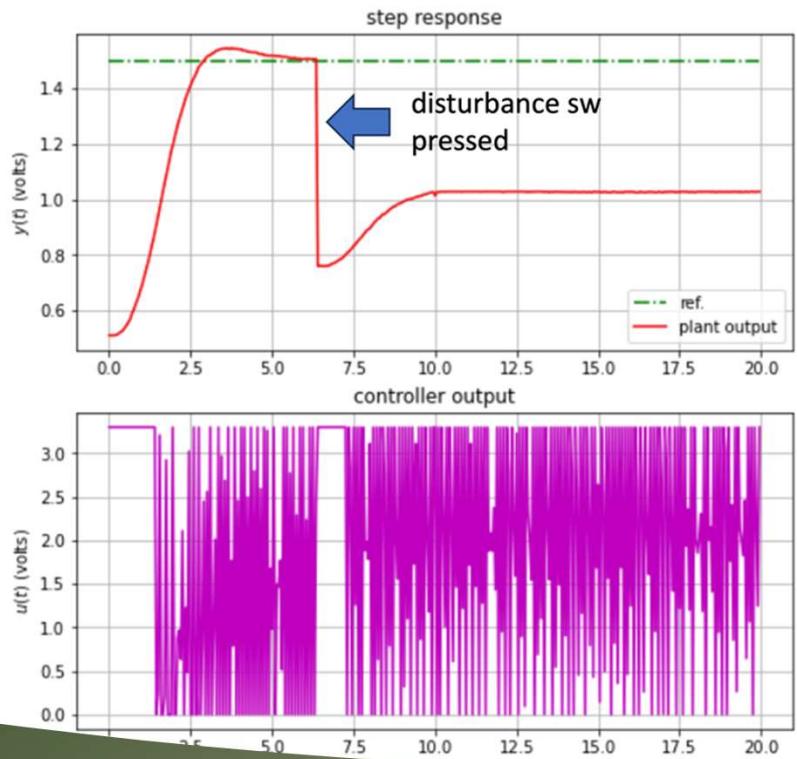
$$C(s) = 250 \frac{(s+1)}{s(s+700)}$$



ปริยบเทียบผลตอบสนอง
ขั้นบันได

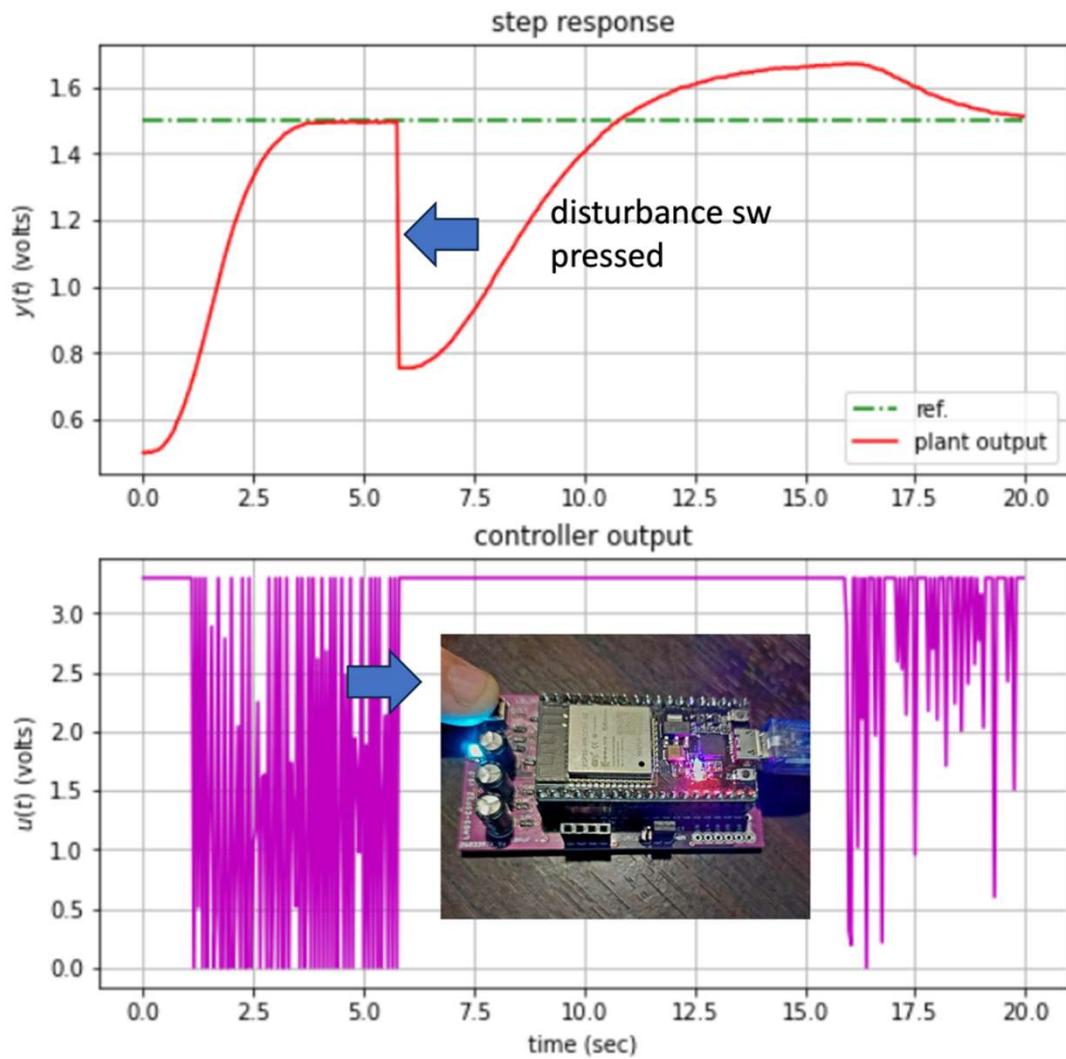
$$C(s) = 250 \frac{(s+1)}{s(s+700)}$$

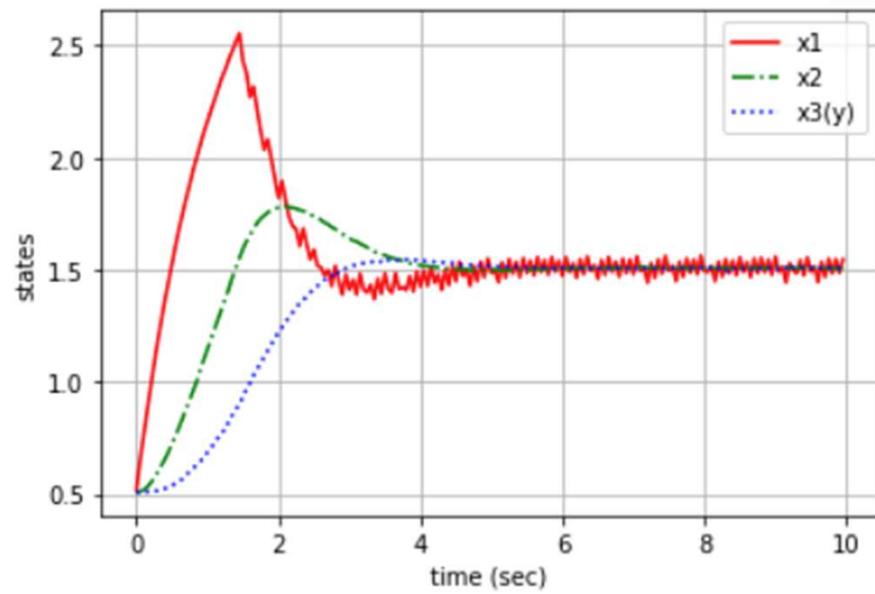
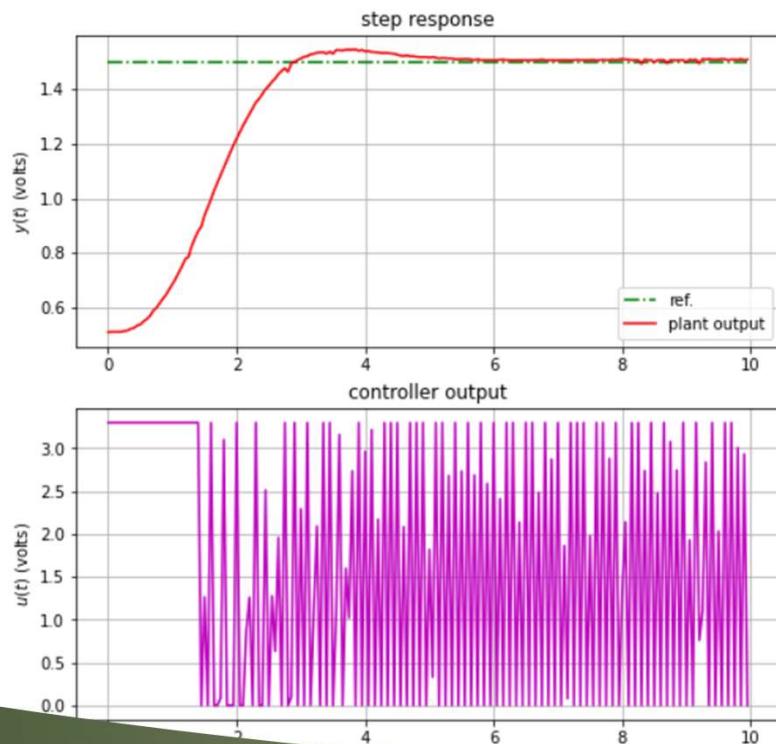




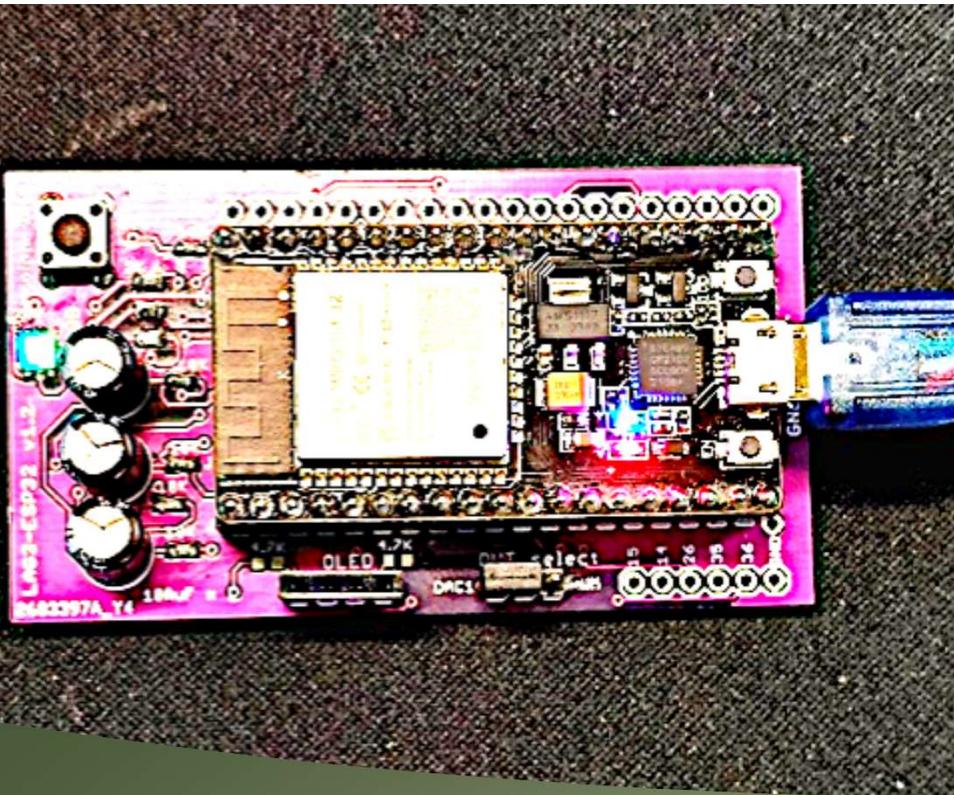
ผลตอบสนองขั้นบันไดเมื่อมีการรบกวนที่อาจพุ่ง

ตัวควบคุมป้อนกลับ สถานะ + ตัวปริพันธ์





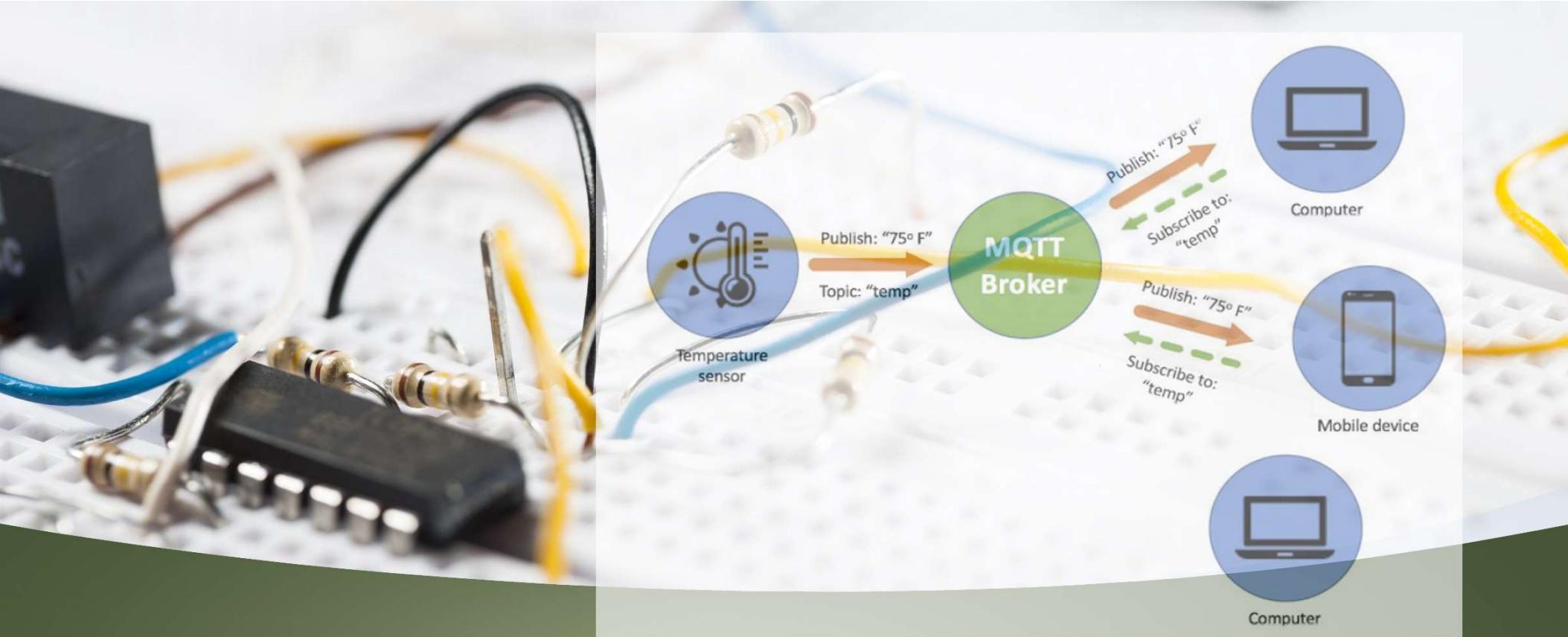
ตัวควบคุมป้อนกลับสถานะ



ทดลองโปรแกรม iot_controller4 .py

ตั้งค่าตัวแปร online=False และ initparm=0

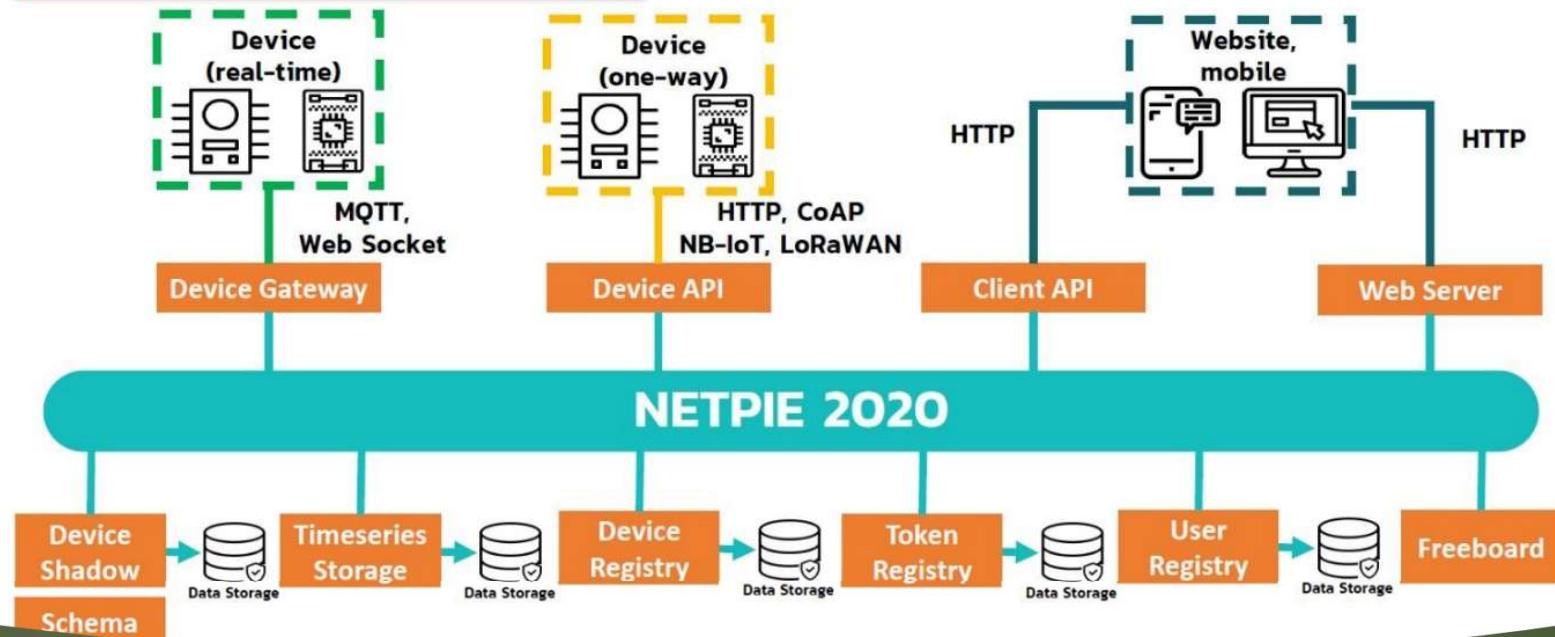
Wokwi link: <https://wokwi.com/projects/389146196723524609>



ตัวควบคุมไอโอที

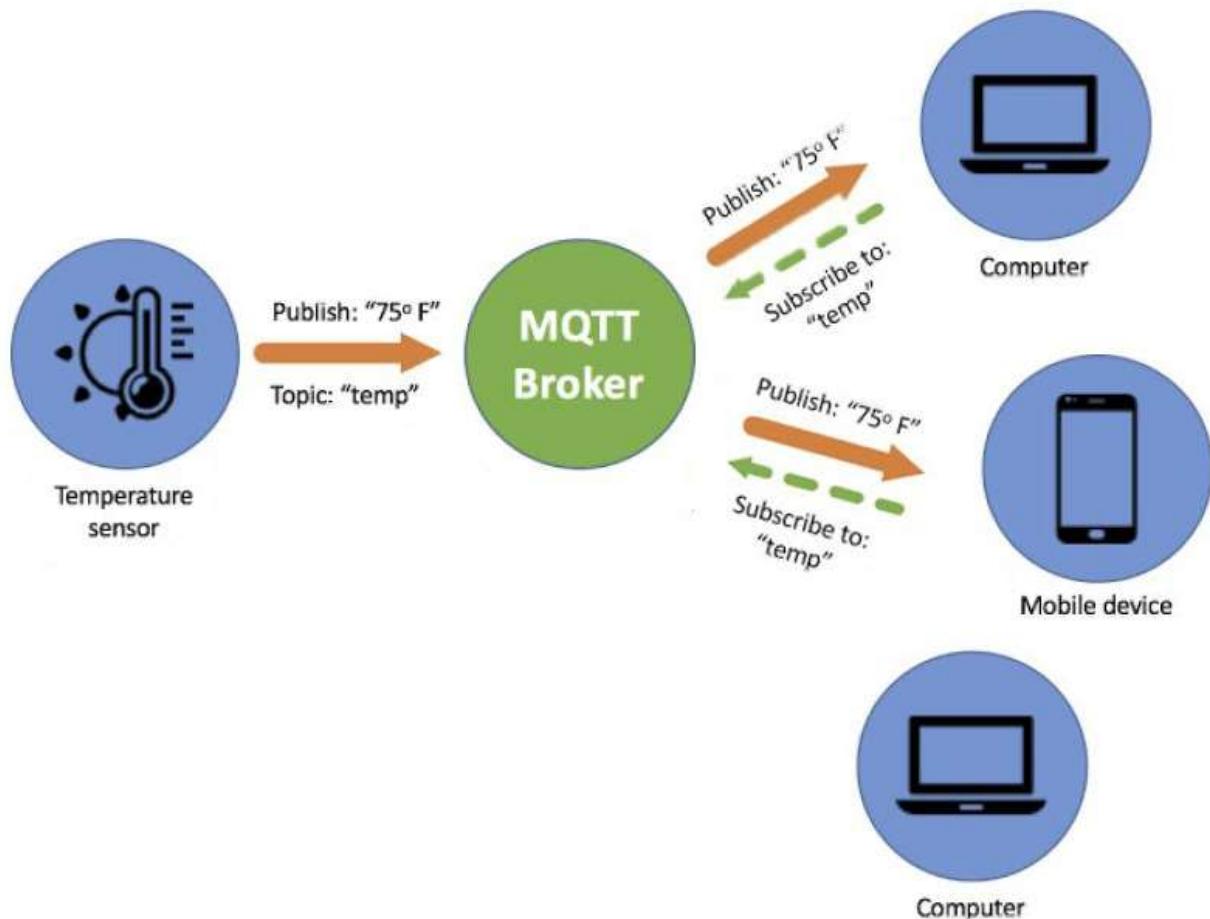
IOT CONTROLLERS

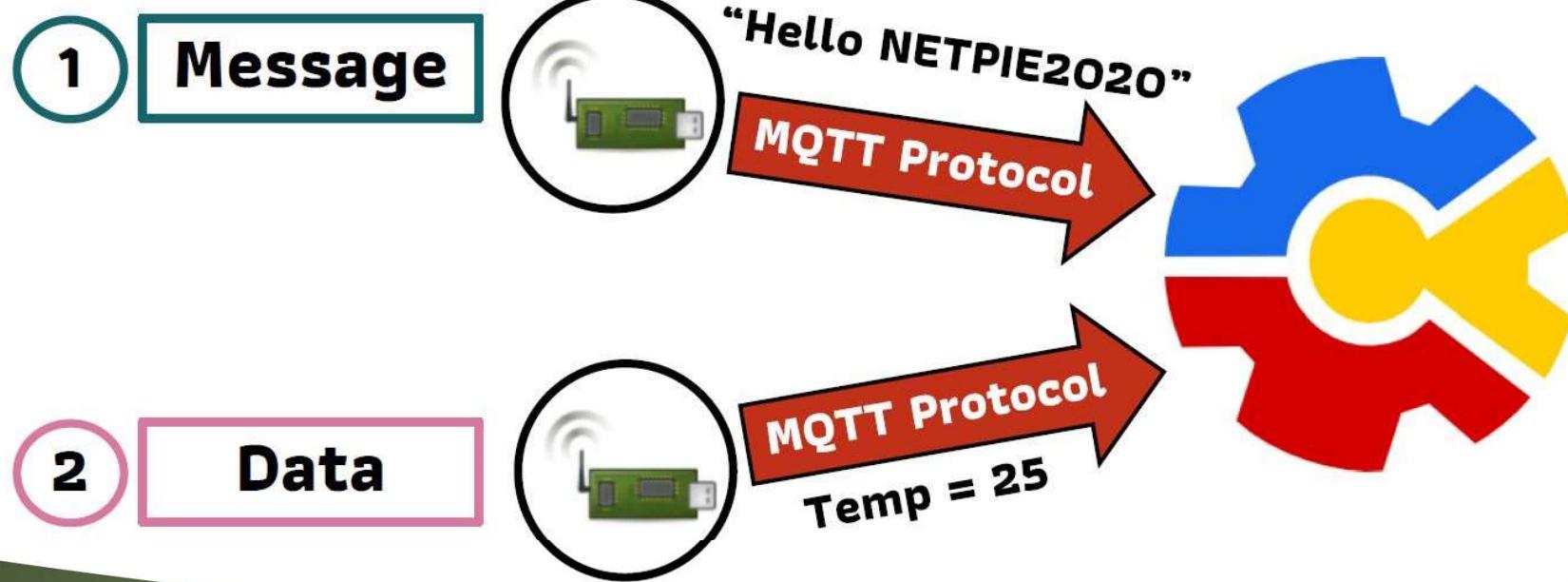
NETPIE 2020 Architecture



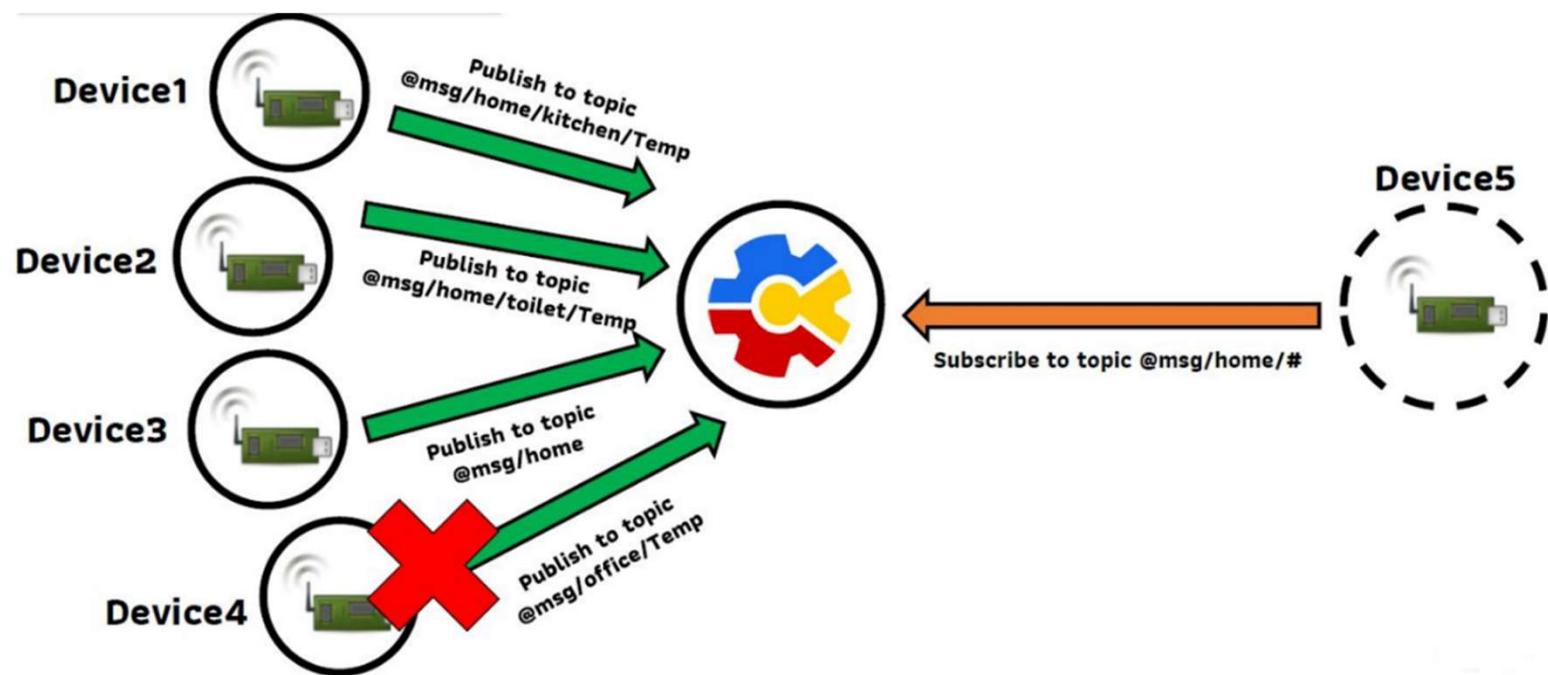
NETPIE 2020 <https://netpie.io/>

การสื่อสารข้อมูล โดยโปรโตคอล MQTT

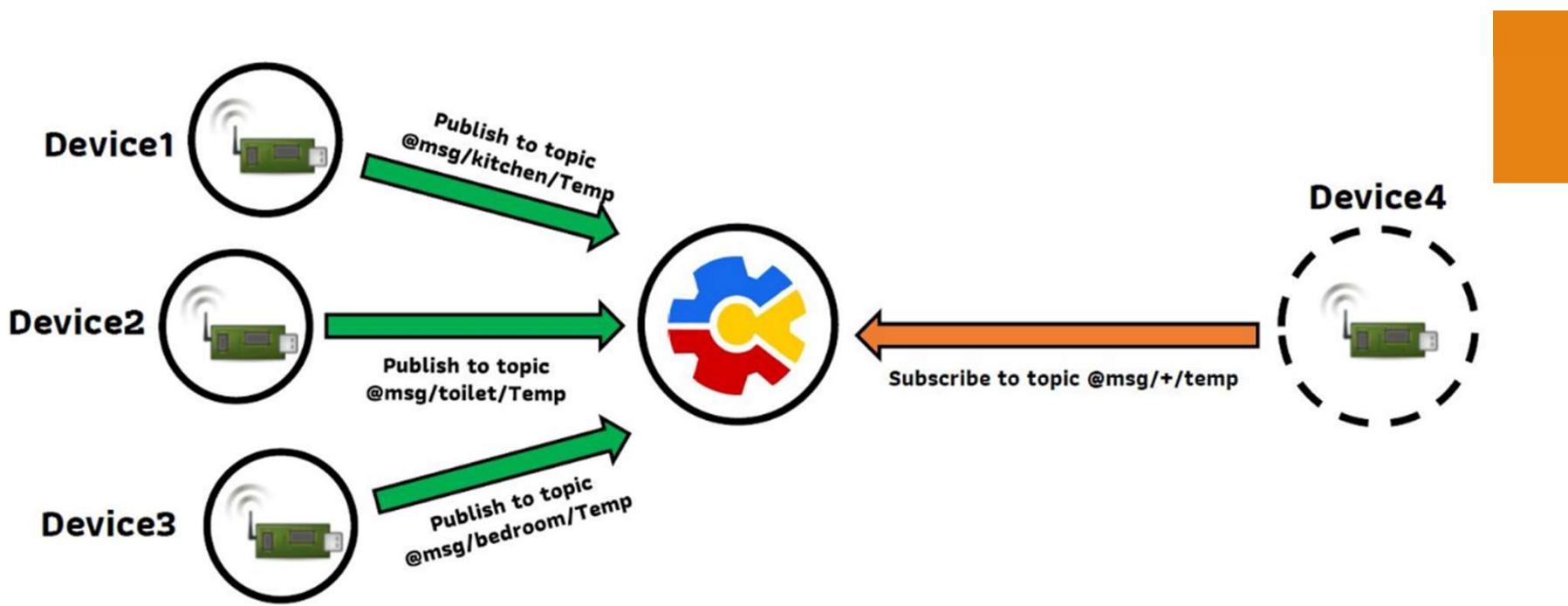




ประเกทข้อมูล MQTT บน NETPIE 2020



การสมัครสมาชิกโดยใช้ wildcard



การสมัครสมาชิกโดยใช้ wildcard +

Connection	Port
mqtt/tcp [broker.netpie.io]	1883
mqtts/tls [broker.netpie.io]	8883
ws [mqtt.netpie.io/mqtt]	80
wss [mqtt.netpie.io/mqtt]	443

หมายเลขพอร์ต MQTT

The image shows a Wokwi project interface. On the left is a code editor with a Python file named 'main.py'. The code includes imports for machine, sys, time, utime, and MQTTClient, along with WiFi setup and MQTT broker configuration. On the right are tabs for 'Simulation' and 'Description'. The 'Simulation' tab shows a digital circuit with an ESP32 microcontroller connected to a breadboard. The 'Description' tab displays a JSON array of three objects, each representing a message to be sent over MQTT. Each object has a 'data' field containing a dictionary with 'led', 'toggle', and 'period' keys.

```
1 # cmdtest2_iot.py
2 # dew.ninja
3 # June 2023
4 # extend to iot
5
6 from machine import Pin, Timer
7 import sys
8 import time
9 from utime import sleep_ms
10
11 # --- added for iot development-----
12 from umqtt.robust import MQTTClient
13 import ujson
14 import network
15
16 # --- WiFi SSID and pwd setup for wokwi
17 wifi_ssid = "Wokwi-GUEST"
18 wifi_pwd = ""
19 #
20 MQTT_BROKER = "broker.netpie.io"
21 MQTT_CLIENT = "924efa0f-b856-440f-9aee"
22 MQTT_USER = "hf7fiG96rw12DfLRJxNkYxYRT"
```

```
[{"data": {"led": 0, "toggle": 1, "period": 1000}}, {"data": {"led": 0, "toggle": 1, "period": 1000}}, {"data": {"led": 0, "toggle": 1, "period": 1000}}]
```

ช่วงสาธิตการใช้งาน NETPIE 2020

cmdtest2_iot.py

<https://wokwi.com/projects/389155007838242817>

เพิ่มนิยามฟังก์ชัน `update_dashboard()` สำหรับส่งข้อความให้กับแดชบอร์ดในหัวข้อ '`@msg/update`' โดยเรียงลำดับของค่าตัวแปรใน `updatestr` ตามโค้ดที่เขียนในฟังก์ชัน

การส่งข้อความให้เดชบอร์ด โดยตรง

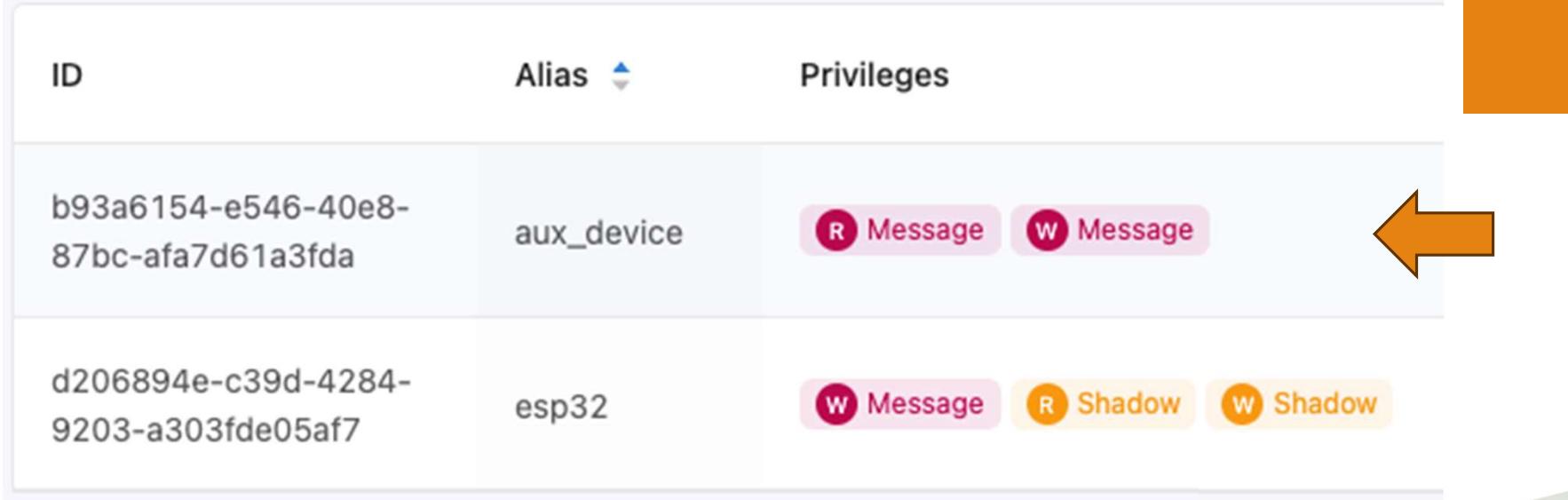
The screenshot shows a device management interface with the following details:

ID	Name	Group	Status
b93a6154-e546-40e8-87bc-af...	aux_device device to receive @msg/update	group1	Offline
d206894e-c39d-4284-9203-a...	esp32 device attach to hardware	group1	Online

Annotations in the image:

- An orange box labeled "เชื่อมต่อเดชบอร์ด" (Connect to Dev Board) has an orange arrow pointing to the first device (aux_device).
- A red box labeled "เชื่อมต่อ ESP32" (Connect to ESP32) has an orange arrow pointing to the second device (esp32).
- A large red arrow points from the bottom right towards the status column.

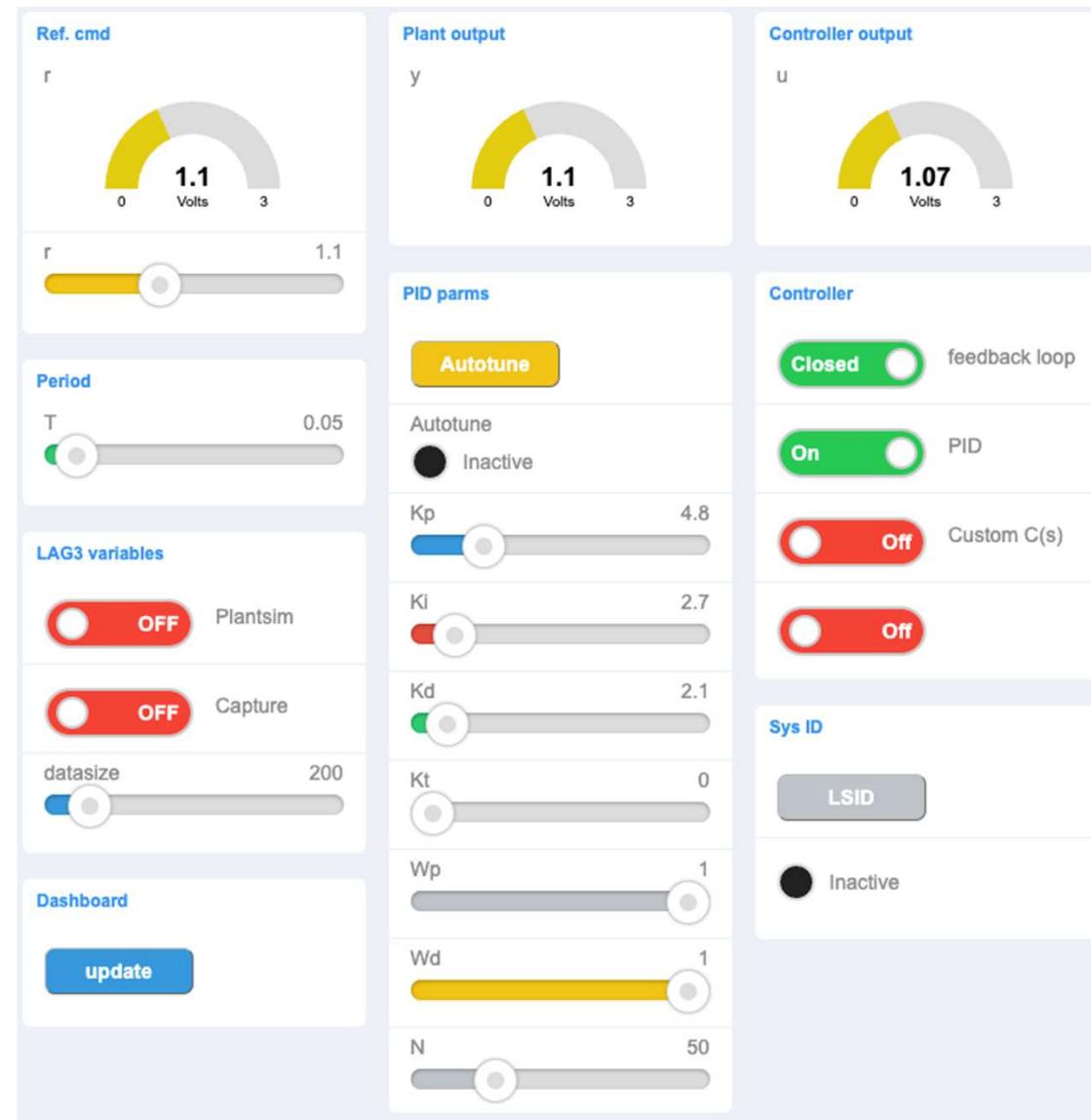
รวมกลุ่มอุปกรณ์ 2 ตัวเข้าด้วยกัน



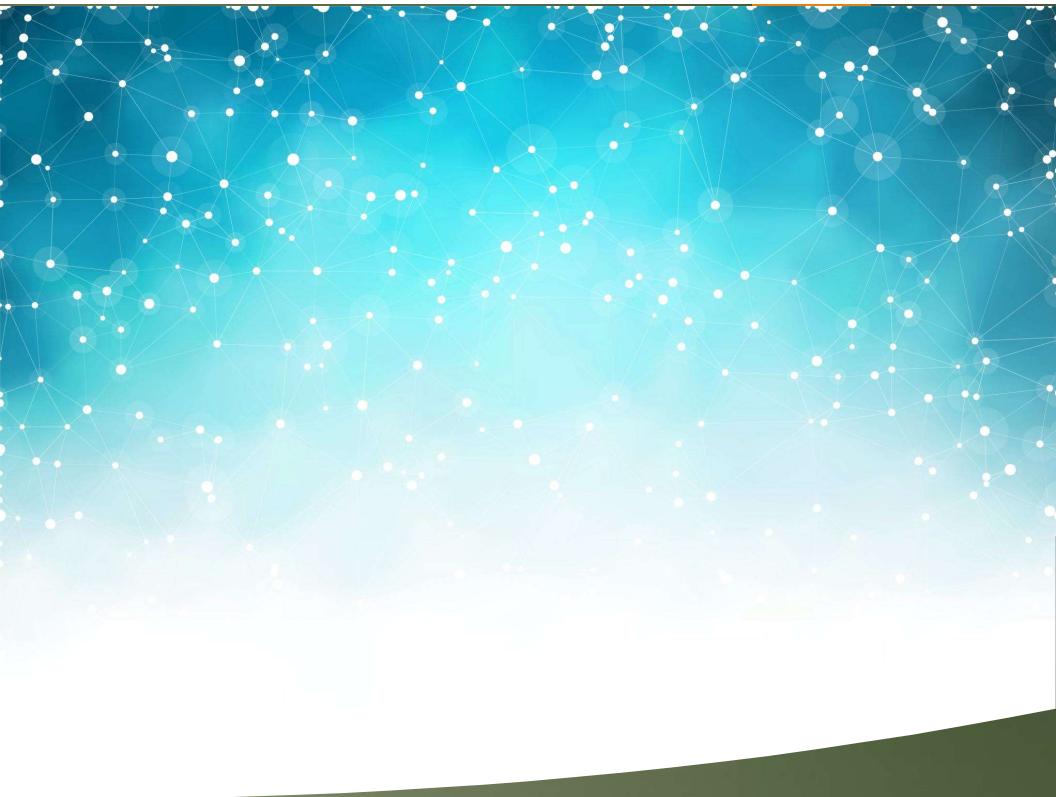
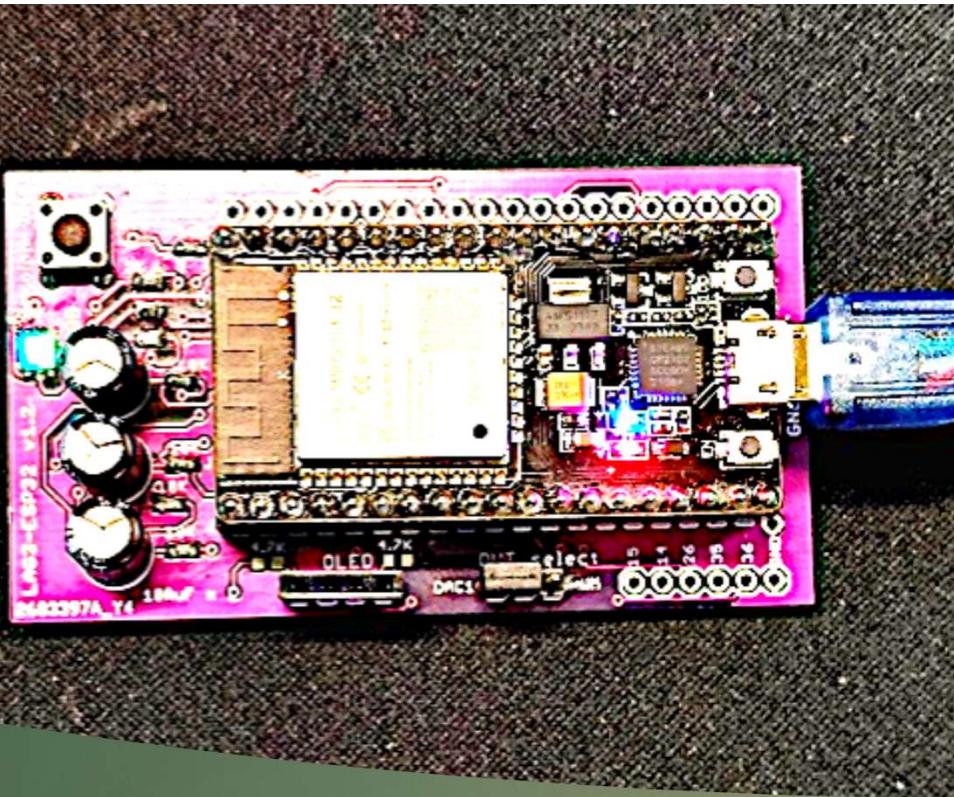
ID	Alias	Privileges
b93a6154-e546-40e8-87bc-afa7d61a3fda	aux_device	R Message W Message
d206894e-c39d-4284-9203-a303fde05af7	esp32	W Message R Shadow W Shadow

เพิ่มอุปกรณ์ตัวใหม่ให้กับเดชบอร์ด

ขั้นตอนหลังจากนี้อยู่ในหนังสือหน้า 237 - 241



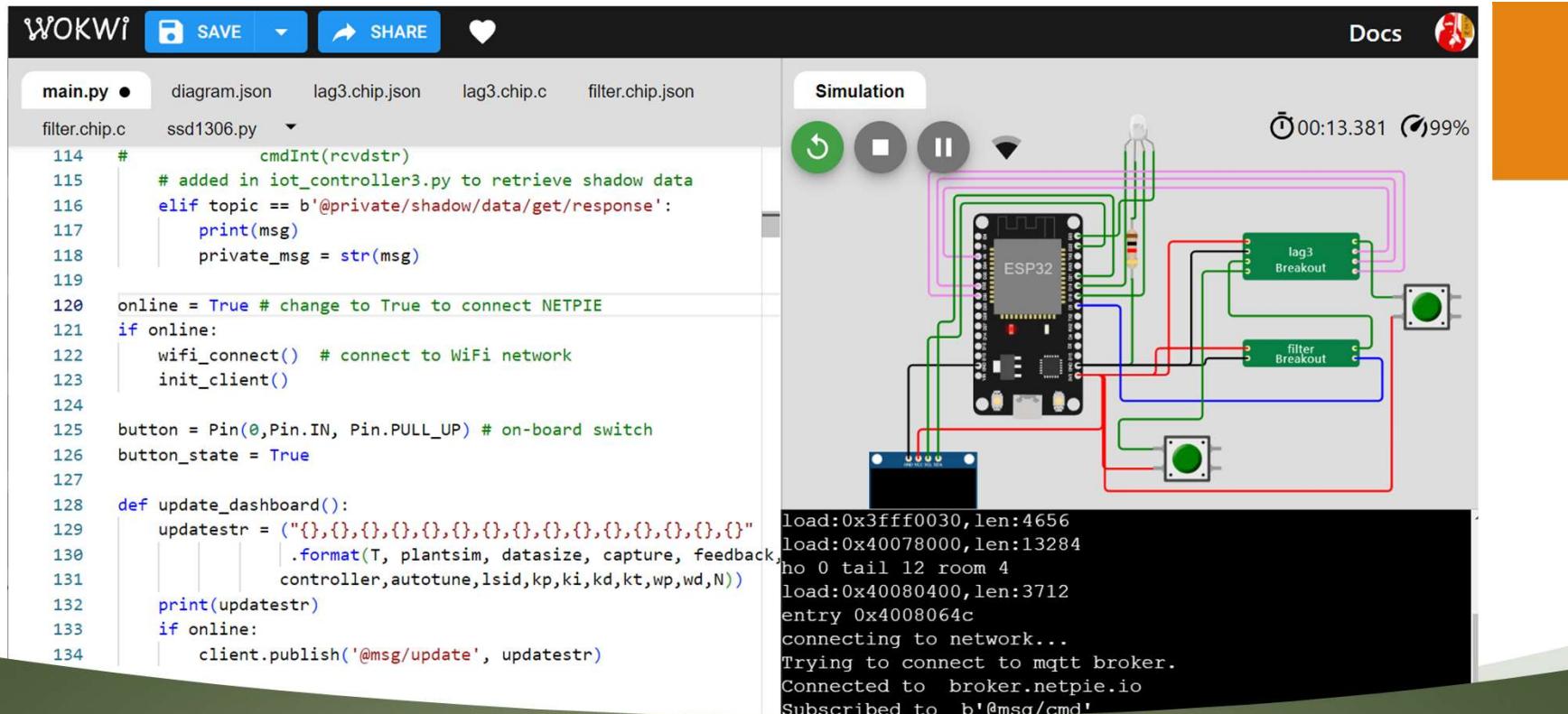
ແຜ່ນບອົງດສໍາຫຼັບ
LAG3-ESP32



ทดลองโปรแกรม **iot_controller4 .py**

ตั้งค่าตัวแปร `online=True` และ `initparm=0`

Wokwi link: <https://wokwi.com/projects/389146196723524609>



ໃຊ້ກາຣຈໍາລອງບນ Wokwi

Workshop A:

- ▶ ทดลองสร้างแดชบอร์ดสำหรับ LAG3 เพื่อนำไปติดตั้งหน้า 71 ให้ได้ widgets มากสุดเท่าที่ทำได้
- ▶ import แดชบอร์ดจากไฟล์ lag3_dashboard-config.json





Part A Summary & Questions

ภาคเช้า (9:00 AM – 12:00 PM)