

MICROPYTHON DEVELOPMENT ON ESP32

DR.VARODOM TOOCHINDA

DEPT. OF MECHANICAL ENGINEERING

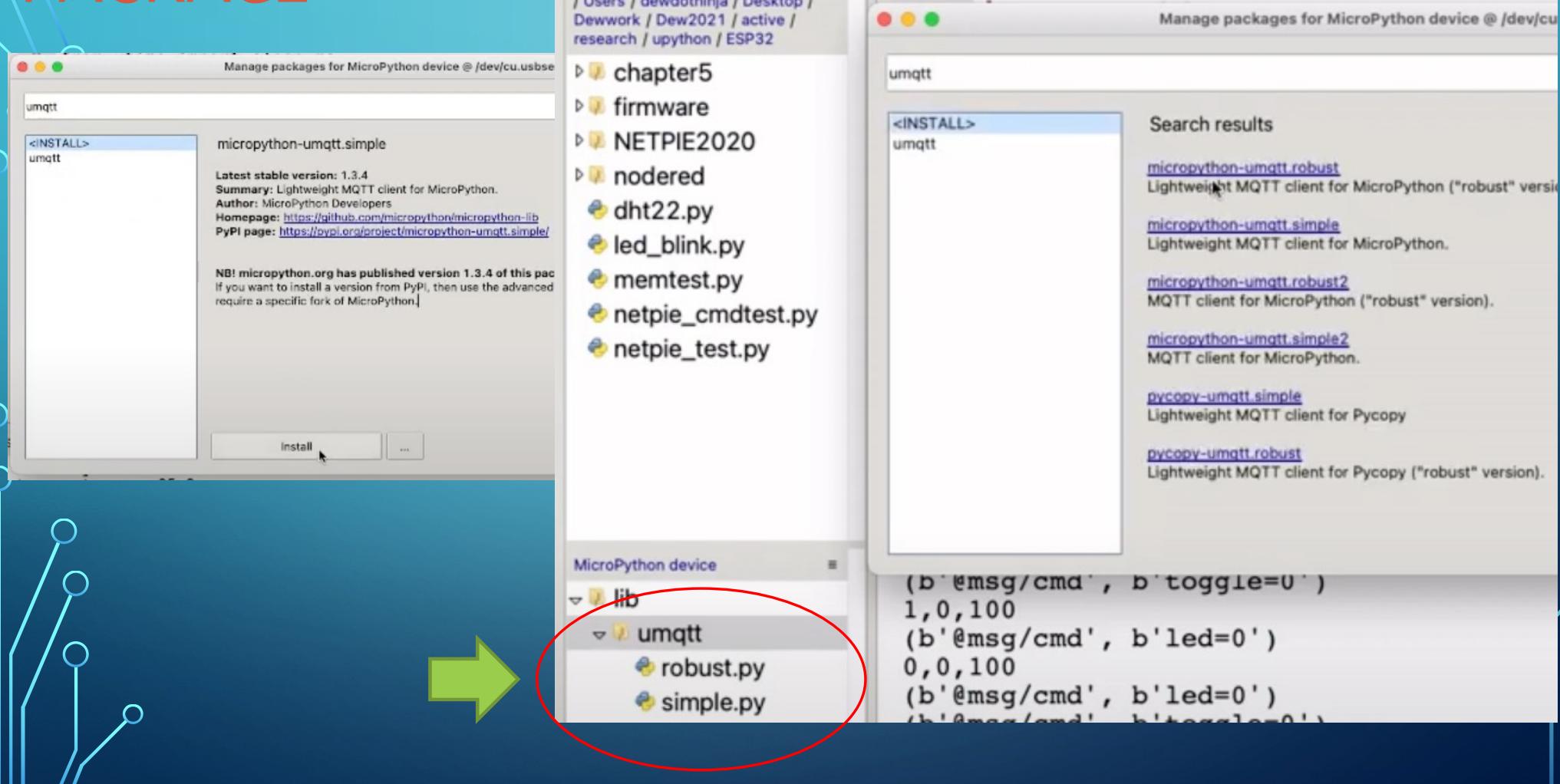
KASETSART UNIVERSITY

01205479 IoT for Electrical Engineering

TOPICS

- NETPIE 2020 connection using umqtt package
- Writing data on shadow and display on dashboard
- Sending command from NETPIE 2020
- Sending message to dashboard for updating

INSTALL MQTT PACKAGE



Ex 1: cmdtest2_iot.py

```
from umqtt.robust import MQTTClient
import ujson
import network

wifi_ssid = "mywifissid" # Fill in your wifi info
wifi_pwd = "123456"
MQTT_BROKER = "broker.netpie.io"
# and your NETPIE2020 info
MQTT_CLIENT = "d24e-c39d-4284-9203-a303fde05af7"
MQTT_USER = "NwPtR2g31R HFYYsZdefN6ECtfF"
MQTT_PWD = "aP*VAV2 sl~AIzLn8q#Zqov5_elMe"
shadow_data = {'led': 0, 'toggle': 0, 'period': 0}
PUBLISH_PERIOD = 2000 # milliseconds
time_previous = 0 # variables to control publishing period
```

Woksi link : <https://wokwi.com/projects/411973148509800449>

```
def wifi_connect():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        wlan.connect(wifi_ssid, wifi_pwd)
    while not wlan.isconnected():
        pass
```

```
def init_client():
    global client
    print("Trying to connect to mqtt broker.")
    try:
        client = MQTTClient(MQTT_CLIENT, MQTT_BROKER,
                            port=1883, user=MQTT_USER,
                            password=MQTT_PWD)
        client.connect()
        print("Connected to ",MQTT_BROKER)
        topic_sub = b"@msg/cmd"
        print("Subscribed to ",topic_sub)
        client.set_callback(sub_cb)
        client.subscribe(topic_sub)
    except:
        print("Trouble to init mqtt.")
```

Ex 1: cmdtest2_iot.py

```
online = True
if online:
    wifi_connect()
    init_client()
```

Ex 1: cmdtest2_iot.py

```
while True:
    if online:
        time_current = time.ticks_ms()
        publish_delta = time_current - time_prev
        if publish_delta>PUBLISH_PERIOD:
            time_prev = time_current
            client.check_msg()
            shadow_data['led'] = led.value()
            shadow_data['toggle'] = toggle_mode
            shadow_data['period'] = led_period
            publish_str = ujson.dumps({"data": shadow_data})
            print(publish_str)
            client.publish("@shadow/data/update", publish_str)
    else:
        user_input()
```

Output from cmdtest2_ IoT.py

data written to shadow

The screenshot shows the NETPIE platform interface. On the left, the sidebar includes sections for PROJECT (myproject), WORKSPACE (Overview, Device, Group, Event Hook, Console), and SETTING (Setting). The main area displays a device configuration for 'esp32' under 'myproject / device / esp32'. The device was created on 2023-07-05. A 'Detail' panel is open, and below it, a 'Shadow' tab is selected in a navigation bar. The shadow tree shows an object with three properties: led (value 0), period (value 1000), and toggle (value 1). To the right, a 'Key' section provides client ID (d206894e-c39d-428), token (NwPtR2g31R5GiJGK), secret (aP*VAV2avBjsl~AlzLr), status (Online), and an enable switch.

NETPIE

PROJECT + Add Project

myproject

WORKSPACE

- Overview
- Device
- Group
- Event Hook
- Console

SETTING

Setting

myproject / device / esp32

esp32

created date: 2023-07-05

Detail

Shadow Schema Trigger Feed i

Select a node...

- object {3}
 - led : 0
 - period : 1000
 - toggle : 1

Client ID: d206894e-c39d-428

Token: NwPtR2g31R5GiJGK

Secret: aP*VAV2avBjsl~AlzLr

Status: Online

Enable:

Last Update : 06-07-23 14:17

EXERCISE 1 : (optional)

- develop micropython program for 3 on-board LED commands
 - led=0/1 :
 - toggle=0/1 : blink LED with period 200 ms
 - brightness=x : adjust LED brightness from 0 (off) to 1 (maximum)
- Use ex1_netpie_cmptest.py and fill in the missing code
- Create dashboard to command ESP32

HCSR04 example

Homework # 5

WOKwi SAVE SHARE netpie_ultrasonic_oled Docs User

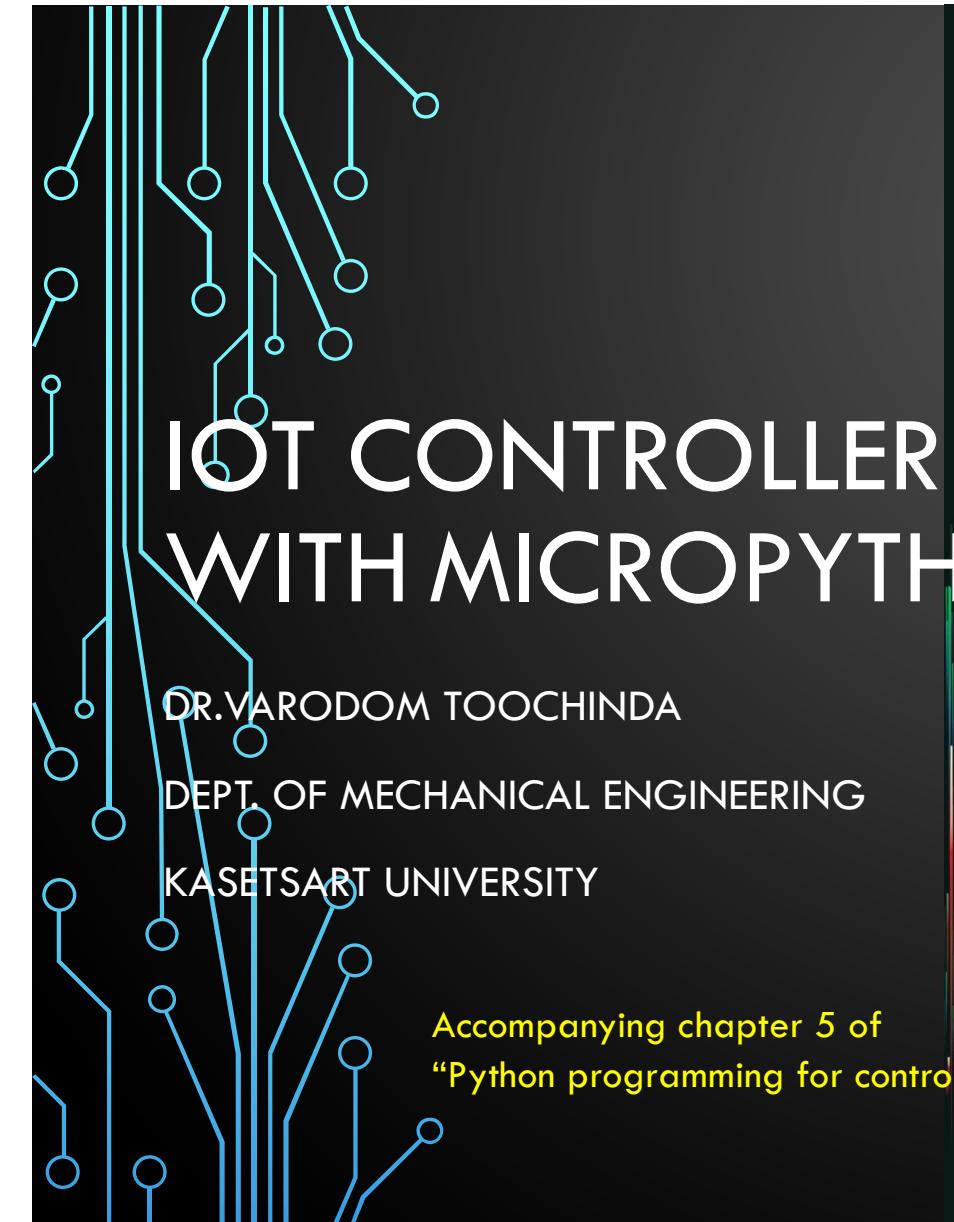
main.py diagram.json hcsr04.py ssd1306.py

```
169     while True:
170         if distance < mindist:
171             alert_status = 1
172             led.on()
173         else:
174             alert_status = 0
175             led.off()
176         oled_show(oled,mindist,distance,alert_status)
177         if online:
178             time_current = time.ticks_ms()
179             publish_delta = time_current - time_prev
180             if publish_delta>PUBLISH_PERIOD: # publish interval must be longer than 1 second
181                 time_prev = time_current
182                 client.check_msg()
183
184                 # -- this is code for writing data to shadow ----
185                 # --- change to direct message method if you want to see it in dashboard or node-red
186                 sensor_data['min_distance'] = mindist
187                 sensor_data['distance'] = distance
188                 sensor_data['alert'] = alert_status
189                 publish_str = ujson.dumps({"data": sensor_data})
190                 print(publish_str)
191                 client.publish("@shadow/data/update", publish_str)
192
193                 #time.sleep_ms(led_period)
194
195
196
197
198
199
```

Simulation 00:10.032 99%

HC-SR04

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00
,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:4728
load:0x40078000,len:14876
ho 0 tail 12 room 4
load:0x40080400,len:3368
entry 0x400805cc
```

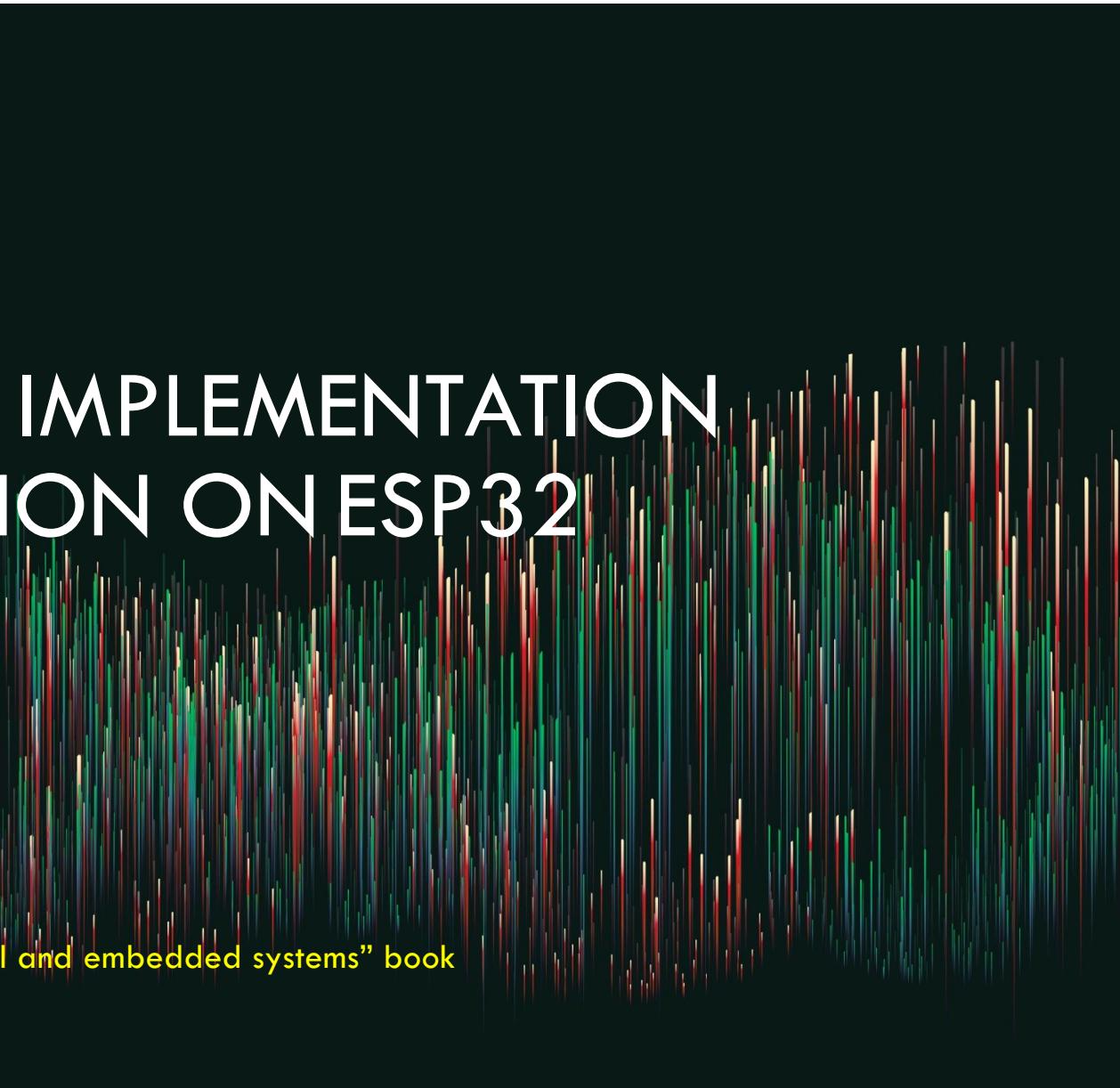


IOT CONTROLLER IMPLEMENTATION WITH MICROPYTHON ON ESP32

DR.VARODOM TOOCHINDA

DEPT. OF MECHANICAL ENGINEERING

KASETSART UNIVERSITY

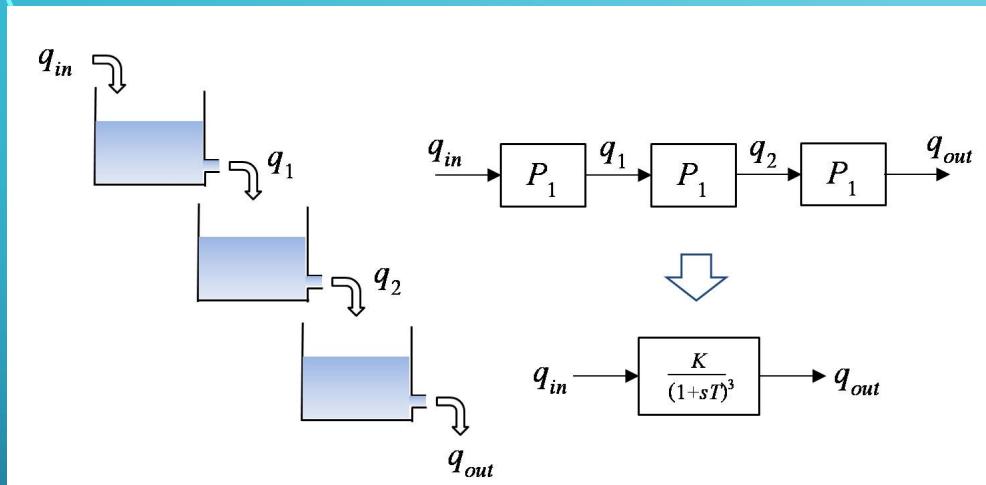


Accompanying chapter 5 of
“Python programming for control and embedded systems” book

TOPICS

- add IoT functionality to LAG3 controllers
- create Dashboard for control

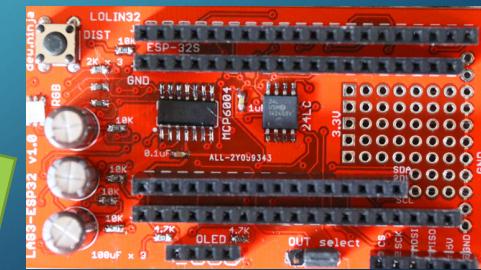
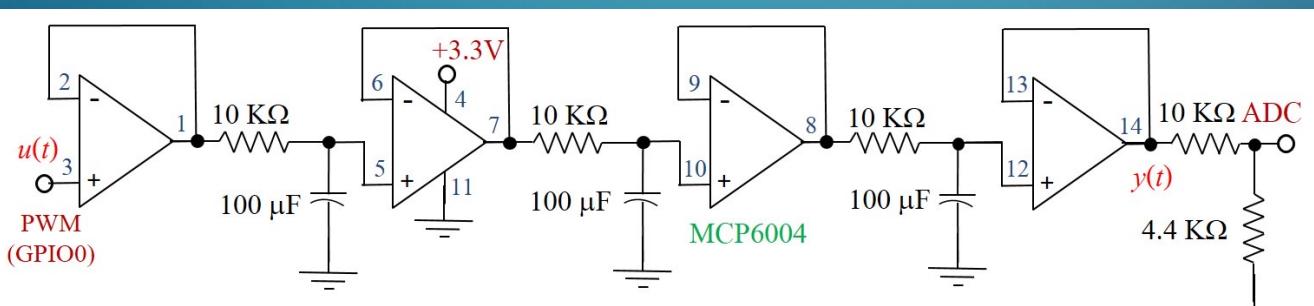
REVIEW OF LAG3 SETUP



normalized transfer
function

$$\frac{1}{(s + 1)^3}$$

Simulated by RC and op-amp circuit



LAG3 SIMULATION ON WOKWI

WOKWI SAVE SHARE lag3_iot_controllers Docs

main.py diagram.json lag3.chip.json lag3.chip.c filter.chip.json filter.chip.c
ssd1306.py

```
78     def init_client():
98         except:
100
101
102     def sub_cb(topic, msg):
103         global private_msg, cmdtime_prev
104         print((topic, msg))
105         if topic == b'@msg/cmd':
106             rcvdstrs = str(msg).split("") # get rid of b'
107             rcvdstr = rcvdstrs[1]
108             cmdInt(rcvdstr)
109             # this delay is needed for nodered implementation
110             # cmdtime_current = time.ticks_ms()
111             # delta_cmdtime = cmdtime_current - cmdtime_prev
112             # if delta_cmdtime > CMD_DELAY:
113             #     cmdtime_prev = cmdtime_current
114             #     cmdInt(rcvdstr)
115             # added in iot_controller3.py to retrieve shadow data
116             elif topic == b'@private/shadow/data/get/response':
117                 print(msg)
118                 private_msg = str(msg)
119
120     online = False # change to True to connect NETPIE
121     if online:
122         wifi_connect() # connect to WiFi network
123         init_client()
124
125     button = Pin(0,Pin.IN, Pin.PULL_UP) # on-board switch
126     button_state = True
127
128     def update_dashboard():
129         updatestr = ("{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}","{}")
130             .format(T, plantsim, datasize, capture, feedback,
131             controller.autotune,lsid,kp,ki,kd,kt,wp,wd,N)
```

Simulation + ⋮

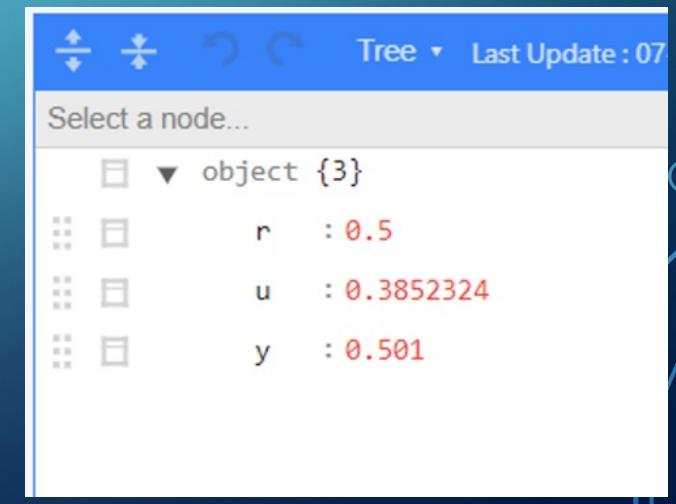
On line 141, set initparm = 0

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0
mode:DIO, clock div:2
load:0x3fff0000, len:1656
load:0x400780 ho 0 tail 12 https://wokwi.com/projects/406624428008336385
load:0x40080400, len:3712
entry 0x4008064c
0.05,0,200,0,1,0,0,0,4.8,2.74,2.1,0,1,1,50
Enter command :
```

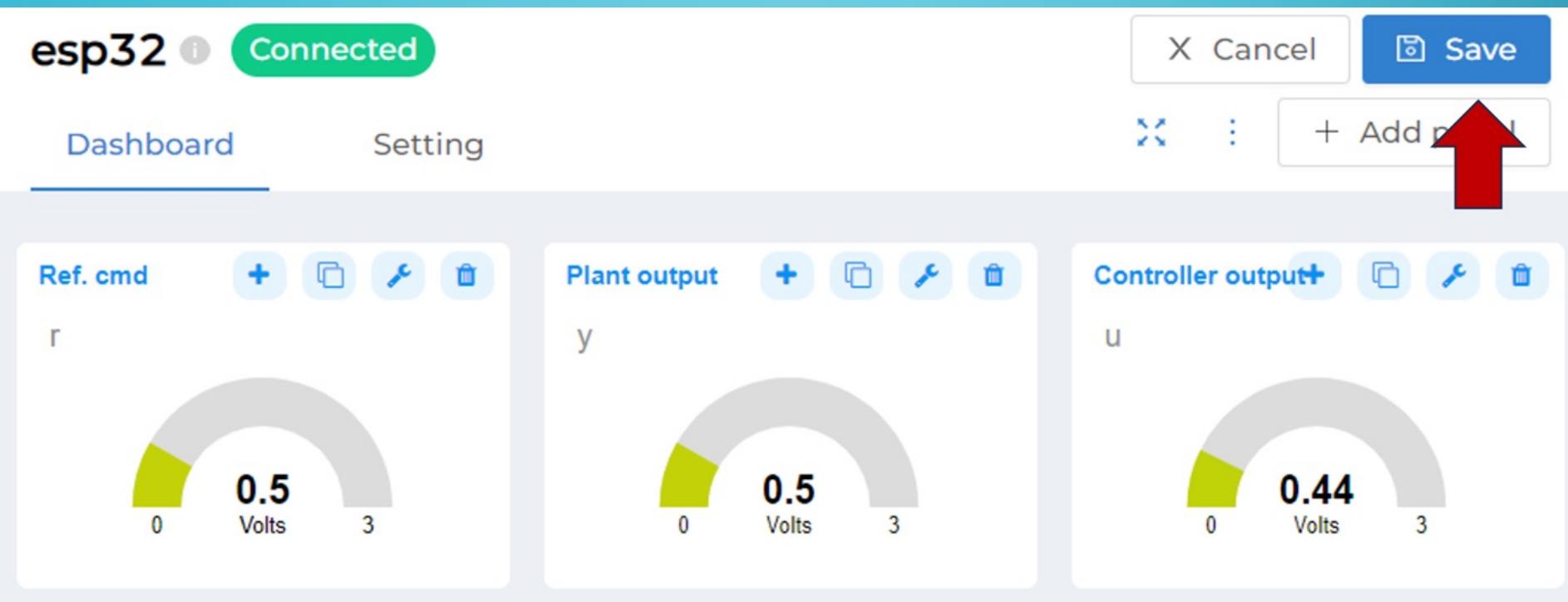
Ex 2: iot_controller1.py

```
shadow_data = {'r': 0, 'y': 0, 'u': 0}
```

```
shadow_data['r'] = r
shadow_data['y'] = y
shadow_data['u'] = ulim
publish_str = ujson.dumps({"data": shadow_data})
client.publish("@shadow/data/update", publish_str)
```



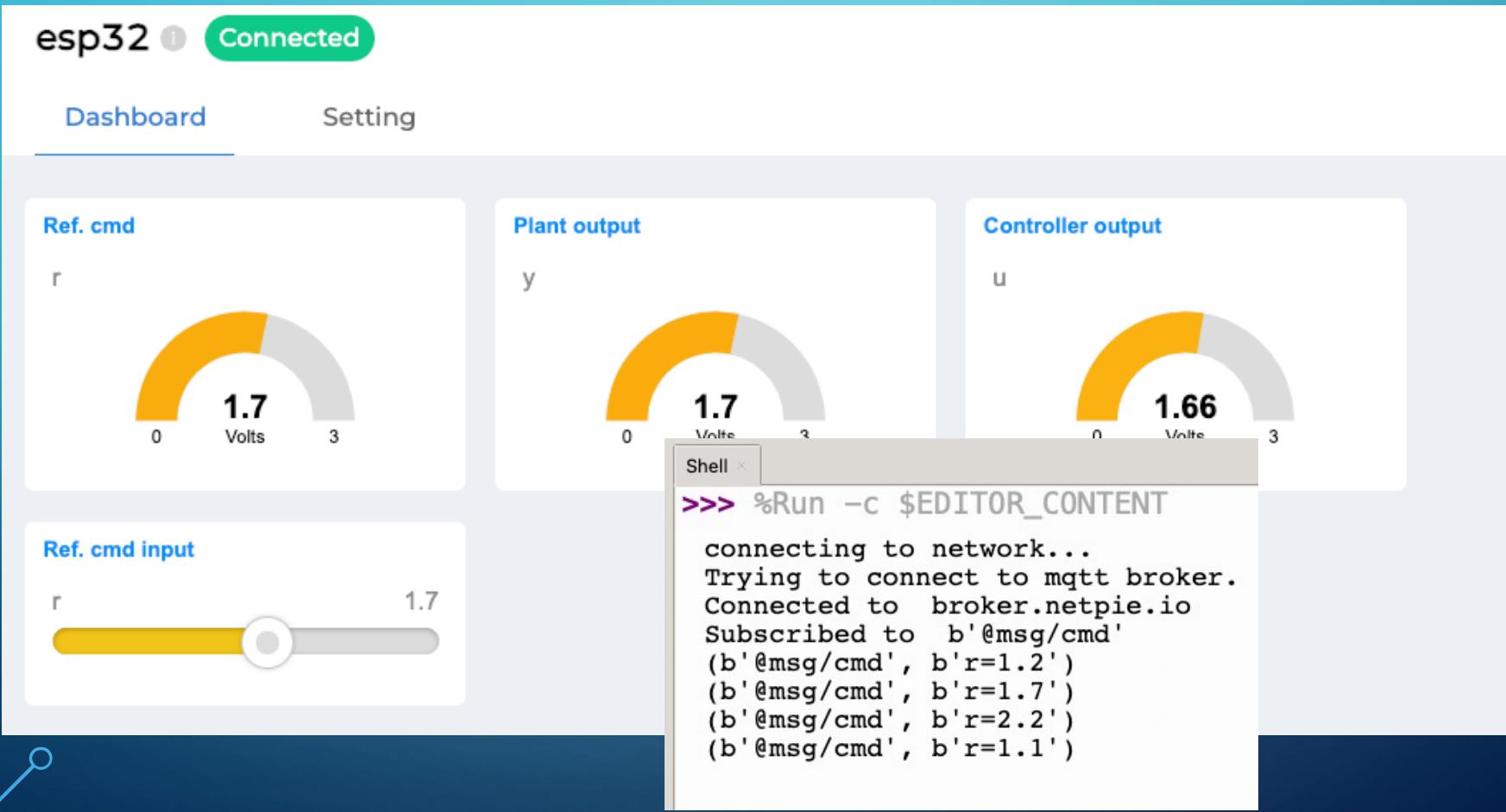
Gauge widgets for r,y,u data



```
def init_client():
    global client
    print("Trying to connect to mqtt broker.")
    try:
        client = MQTTClient(MQTT_CLIENT, MQTT_BROKER,
                            port=1883, user=MQTT_USER,
                            password=MQTT_PWD)
        client.connect()
        print("Connected to ",MQTT_BROKER)
        topic_sub = b"@msg/cmd"
        print("Subscribed to ",topic_sub)
        client.set_callback(sub_cb)
        client.subscribe(topic_sub)
    except:
        print("Trouble to init mqtt.")
```

```
def sub_cb(topic, msg):
    print((topic, msg))
    if topic == b'@msg/cmd':
        rcvdstrs = str(msg).split("") # get rid of b'
        rcvdstr = rcvdstrs[1]
        cmdInt(rcvdstr)
```

slider to send r command



Send message to dashboard directly

```
elif cmdstr.lower() == "kp":  
    if noparm==1:  
        print(kp)  
    else:  
        kp = float(parmstr)  
        if kp > 100: # set maximum kp  
            kp = 100  
        elif kp < 0: # minimum kp  
            kp = 0  
        PID_update()  
        update_dashboard() # <= add this line
```

iot_controller2.py

Send message to dashboard directly

Device

Show 2 devices

Check all

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



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

Note : for latest NETPIE update, only one device is required.

Kp slider setup

A slider widget that can perform Javascript action.

TYPE: Slider

SLIDER CAPTION: Kp

FILLED COLOR: Blue

DISPLAY VALUE: YES

MIN VALUE: 1

MAX VALUE: 20

STEP: 0.1

SENSITIVITY (MS): 200

INITIAL VALUE: 0

The default value set only the first time the widget is loaded.

AUTO UPDATED VALUE: `(#["aux_device"]["msg"]["update"]).split(",")[8]` ←

Slider will be updated upon the change of variables (e.g. other data sources)

← ←

ONSLIDE ACTION: Add some Javascript here. You can access to a slider attribute using variables 'value' and 'percent'. ←

ONSTOP ACTION: Add some Javascript here. You can access to a slider attribute using variables 'value' and 'percent'. `#["esp32"].publishMsg("cmd","kp="+value)` ←

← ←

Kp slider test

Proportional gain

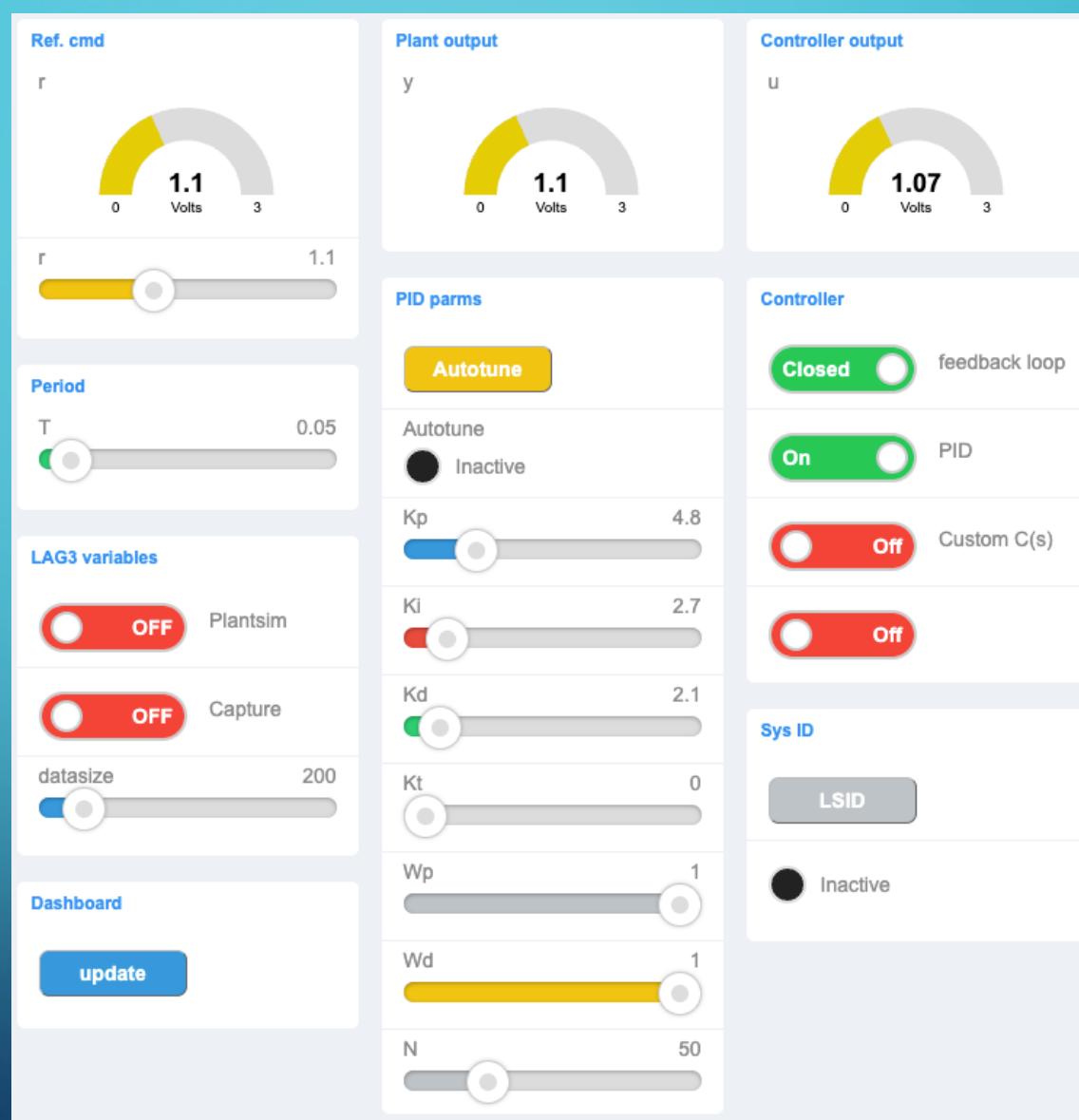
Kp

Redundant input

r

```
>>> %Run -c $EDITOR_CONTENT
connecting to network...
Trying to connect to mqtt broker.
Connected to broker.netpie.io
Subscribed to b'@msg/cmd'
0.05,0,200,0,1,0,0,0,4.8,2.74,2.1,0,1,1,50
(b'@msg/cmd', b'kp=8.9')
0.05,0,200,0,1,0,0,0,8.9,2.74,2.1,0,1,1,50
(b'@msg/cmd', b'kp=16')
0.05,0,200,0,1,0,0,0,16.0,2.74,2.1,0,1,1,50
(b'@msg/cmd', b'kp=1.6')
0.05,0,200,0,1,0,0,0,1.6,2.74,2.1,0,1,1,50
```

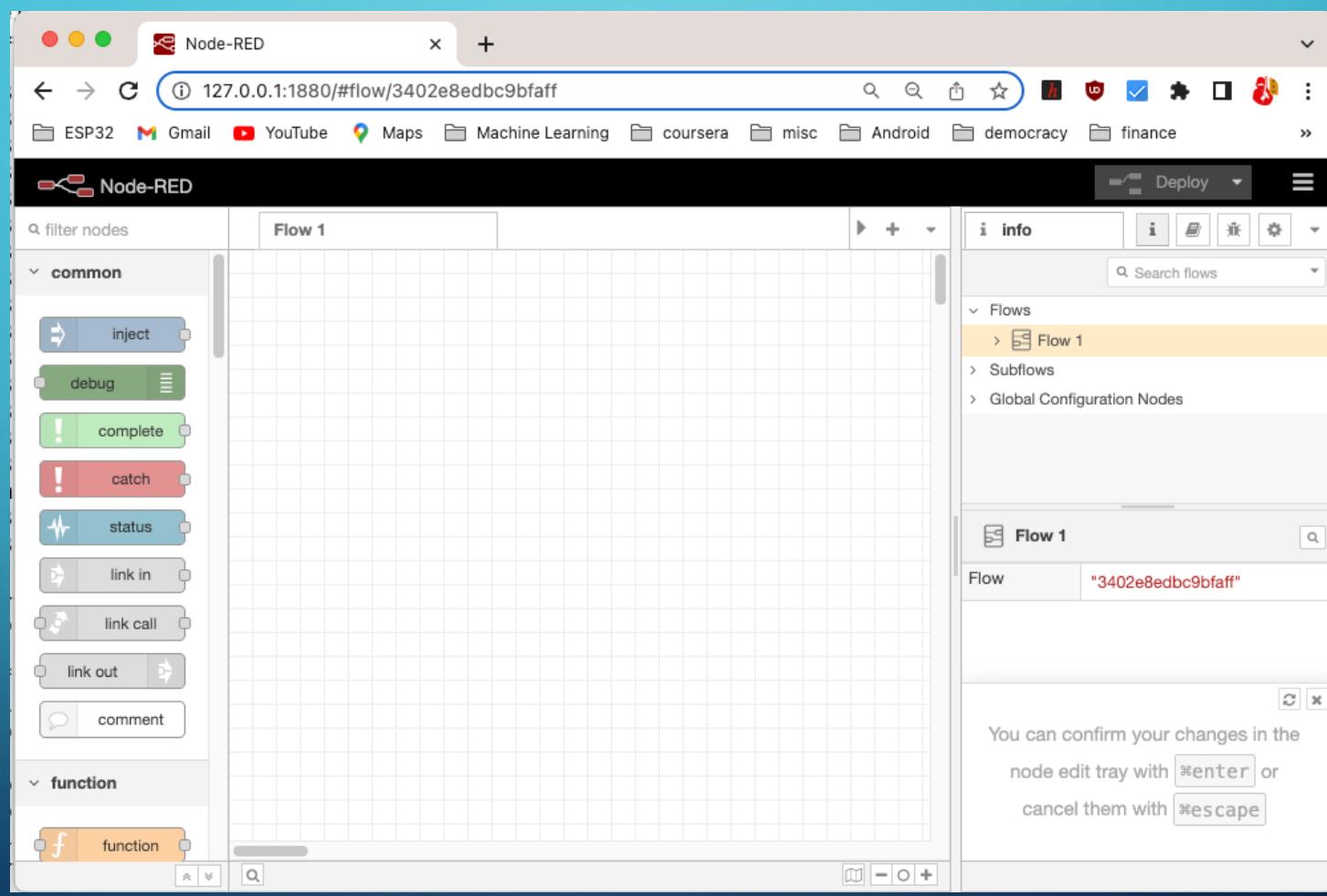
final dashboard for LAG3 control



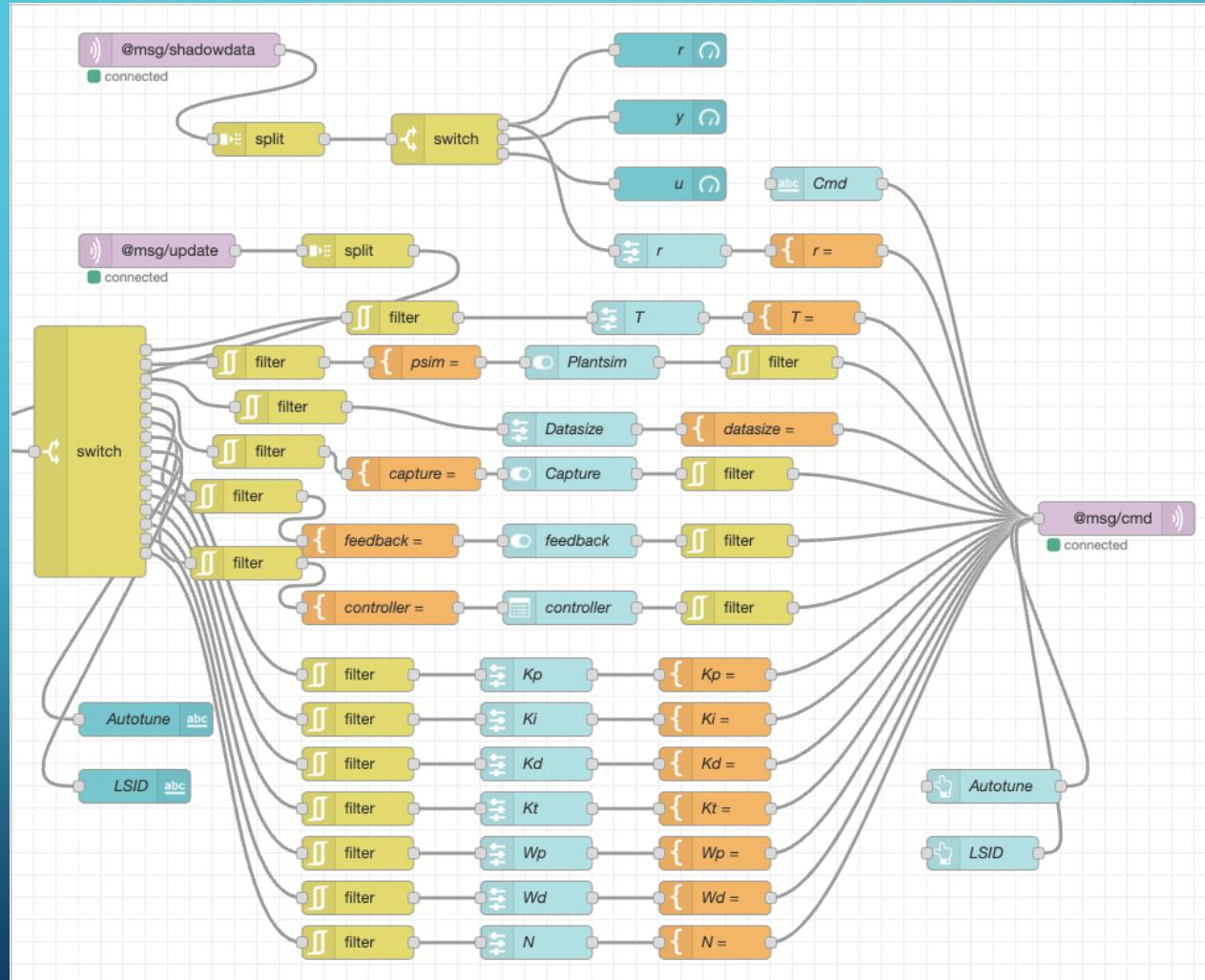
Development on node-red

- first install node.js from <https://nodejs.org>
- Open terminal and type command
 - `sudo npm install -g --unsafe-perm node-red`
- type **node-red** in terminal
- Look for message
 - Server now running at `http://127.0.0.1:1880/`
- copy and paste URL to browser

Development on node-red



Complete flow for LAG3



node-red dashboard for LAG3

LAG3

RYU display

Ref. Cmd
1 Volts

Plant output
1.001 Volts

Controller output
0.8270065 Volts

Controller

controller PID

feedback

Period 0.05

General

Cmd

Plant simulation

datasize 200

Capture

PID

AUTOTUNE

Autotune 0

Kp 5.5

Ki 2.9

Kd 2.6

Kt 0

LSID

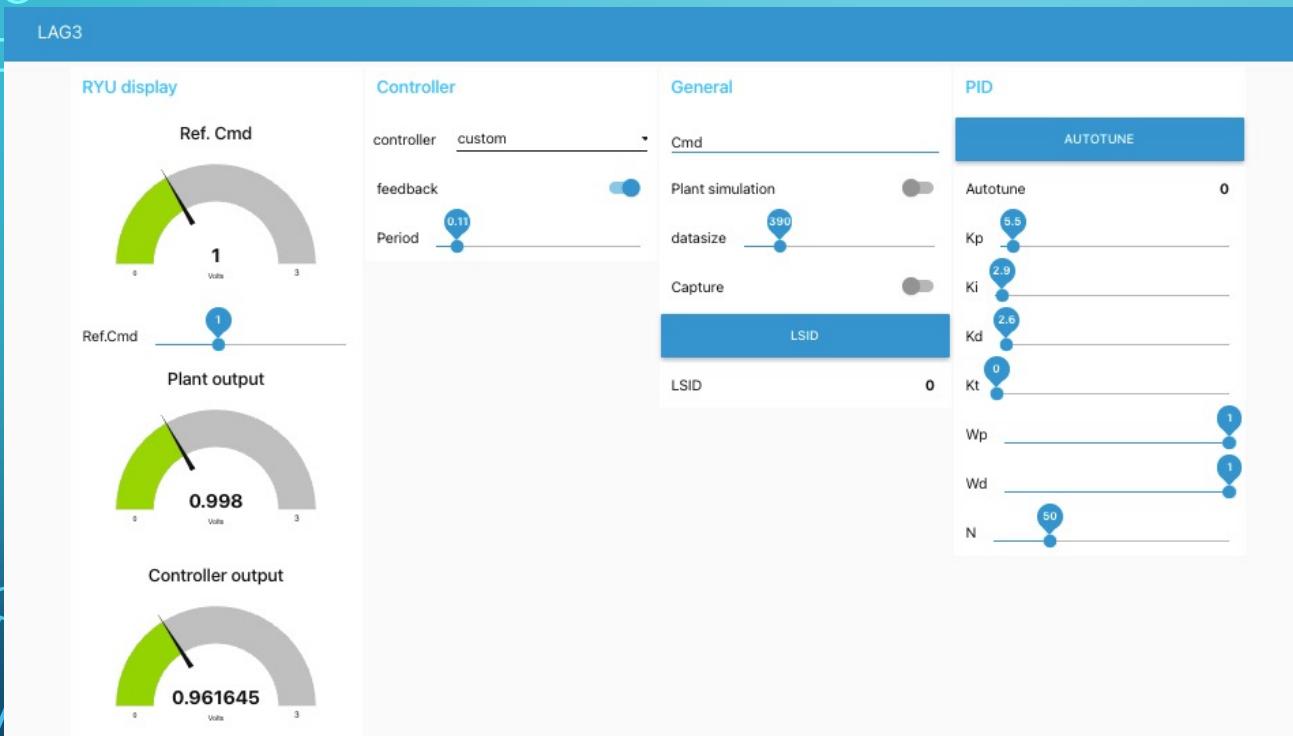
LSID 0

Wp 1

Wd 1

N 50

test node-red dashboard for LAG3



15 import time

Shell

```
(b'@msg/cmd', b'psim=0')  
0.05,0,200,0,1,0,0,0,5.493784,2.8914  
(b'@msg/cmd', b'capture=0')  
0.05,0,200,0,1,0,0,0,5.493784,2.8914  
(b'@msg/cmd', b'feedback=1')  
0.05,0,200,0,1,0,0,0,5.493784,2.8914  
(b'@msg/cmd', b'controller=0')  
0.05,0,200,0,1,0,0,0,5.493784,2.8914  
(b'@msg/cmd', b'r=0.5')  
(b'@msg/cmd', b'kp=5.5')  
0.05,0,200,0,1,0,0,0,5.5,2.891455,2.  
(b'@msg/cmd', b'ki=2.9')  
0.05,0,200,0,1,0,0,0,5.5,2.9,2.60955  
(b'@msg/cmd', b'kd=2.6')  
0.05,0,200,0,1,0,0,0,5.5,2.9,2.6,0.0  
(b'@msg/cmd', b'r=1')  
(b'@msg/cmd', b'controller=1')  
0.05,0,200,0,1,1,0,0,5.5,2.9,2.6,0.0  
(b'@msg/cmd', b't=0.11')  
0.11,0,200,0,1,1,0,0,5.5,2.9,2.6,0.0  
(b'@msg/cmd', b'datasize=390')  
0.11,0,390,0,1,1,0,0,5.5,2.9,2.6,0.0
```

Q&A SESSION

Thank You

