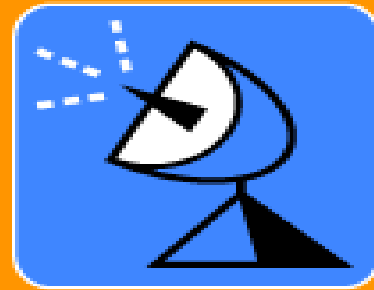


Wiki Bot in Python + ファイルのUpload機能 (IoTLT版)

IoT縛りの勉強会! IoTLT vol.119 2025 1/27

福山大学工学部情報工学科 山之上卓

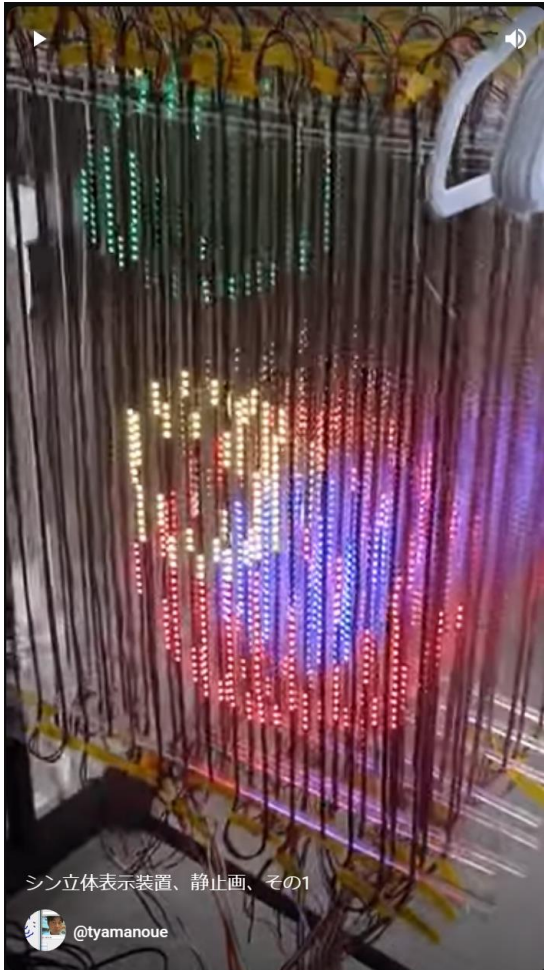


自己紹介





自己紹介



シン立体表示装置、静止画、その1

@tyamanoue

- 1980年 interface LKIT-16によるFORTRANシステム
- 1980年 bit ナノピコ教室じゃんけん大会優勝
- 1988年日本経済新聞「パソコン通信使い電算機言語開発」
- 2016年 ACM SIGUCCS Hall of Fame
- 「ギャル電の意識の低いプレゼンバトル」優勝
@Geek night 広島, 2017 12/9
- MF Tokyo 2018, MF Taipei 2018, NT広島2019
- NT金沢2022, OMMF2022,
- MFT2023, MFK2024, OMMF2024
- ヒーローズ・リーグ2023 トランジスタ技術賞受賞
- 情報処理学会フェロー
- 月刊高専 山之上 卓

何を作ったか？



- PukiWikiのページの記述でRaspberry Pi を制御
(昨年のIoTLT110, Raspberry Pi Pico Wを制御するシステム・・・Pico Bot) pico→Raspberry Pi
- 昨年作ったPico Botを実現するための、PukiWiki の
API、Puki Wiki Driver(MicroPython)をPythonに移植。
(Create, Read, Update, Delete うちの、R,U (D))

Wiki Bot panel

main script

sta	sto	save pr	clear
bot yama_bot_0000_0000_0000_0001		wait	600000

URI	name	pass
http://192.168.1.19/pukiwiki/?ex01		*****
http://192.168.1.19/pukiwiki/?ex01		*****

property	value
read_interval	1800000
exec_interval	0
write_interval	0
report_length	400

command enter clear_message

```

object_page http://192.168.1.19/pukiwiki/?ex01 or http://192.168.1.19/pukiwiki/?ex01
device yama_bot_0000_0000_0000_0001 or yama_bot_0000_0000_0000_0001 start after no write for 10 min.
command: set read_interval=1800000
command: set exec_interval=0
command: set report_length=400
command: py test
py: import time
py: import subprocess
py: subprocess.run("/home/pi/python/capture.sh")
py: date=self.Pico_Time.get_now()
py: #date=2024-12-31 14:25:16.61944
py: time_x=(date.split())[1]
py: print('time_x='+time_x)
py: time_xx=time_x.split(':')
py: hm=time_xx[0]+'_'+time_xx[1]
py: print('time='+hm)
py: attach_dir='/home/pi/python/'
py: attach_name=hm+'.jpg'
py: print('attach_name='+attach_name)
py: cmd_line="/usr/bin/libcamera-jpeg -o "+attach_dir+attach_name+" -t 100 --width 640 --height 480"
2025-01-01 09:44:14.574391: set read_interval=600000
2025-01-01 09:44:14.762779: set exec_interval=0
2025-01-01 09:44:14.937731: set report_length=400
2025-01-01 09:54:13.393681: start_write_script_lines
2025-01-01 09:54:14.665069: set read_interval=600000
2025-01-01 09:54:14.847620: set exec_interval=0
2025-01-01 09:54:15.011775: set report_length=400
2025-01-01 10:04:13.544265: start_write_script_lines
2025-01-01 10:04:14.909613: set read_interval=600000
2025-01-01 10:04:15.081381: set exec_interval=0
2025-01-01 10:04:15.253976: set report_length=400
2025-01-01 10:14:13.722705: start_write_script_lines
2025-01-01 10:14:14.893036: set read_interval=600000
2025-01-01 10:14:14.997480: set exec_interval=0
2025-01-01 10:14:15.136541: set report_length=400
2025-01-01 10:24:13.760008: start_write_script_lines
2025-01-01 10:24:14.741961: set read_interval=600000
2025-01-01 10:24:14.929094: set exec_interval=0
2025-01-01 10:24:15.122491: set report_length=400
2025-01-01 10:34:13.761560: start_write_script_lines
2025-01-01 10:34:15.084748: set read_interval=1800000
2025-01-01 10:34:15.252019: set exec_interval=0
2025-01-01 10:34:15.453258: set report_length=400

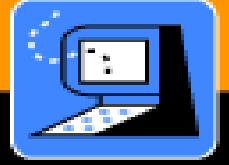
```



- GUI
 - 初期Wikiページ
 - PukiWiki Admin password
 - 初期 read interval
 - 初期 exec interval
 - 初期 report length
 - Script
 - 実行結果



- Wiki Botができること(1/2)。
 - BotはWikiに書かれたスクリプトを定期的に行
 - スクリプト内に、Wiki Botで実行されるPythonのプログラムを埋め込み可。これをWiki Pythonと呼ぶことにします
 - Wiki BotはWikiに実行結果を書き込み可
 - Wiki Bot を実行しているRaspberry Pi に格納された画像などのデータをUploadして、Wikiページの添付ファイルとして、Wiki ページに格納可能



- Wiki Botができること(2/2)
 - Wiki ページの添付ファイルのリストをWiki Python が入手可能
 - Wiki ページの添付ファイルをWiki Python が削除可能
 - Wiki Bot のGUIで、最初に行うWikiページを設定可能。そのときのWikiサーバのPukiWikiのadmin passを設定可能
 - Wiki Bot の起動時のオプションにより、保存された最初に行うWikiページなどの情報を元に、自動起動可能

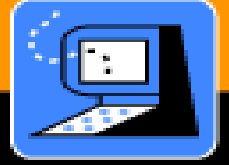
何を作ったか？



- PukiWikiへのファイルのUpload機能。
- Qiita Python のプログラムでPukiWiki にファイルをUpload:

<https://qiita.com/takashiyamanoue/items/3d8a33d53eae880a97b5>

(ハートマークのクリック、よろしくお願いいたします)



- Qiita Wiki Bot in Python 202501

<https://qiita.com/takashiyamanoue/items/eb202e3a5b43550d9b15>

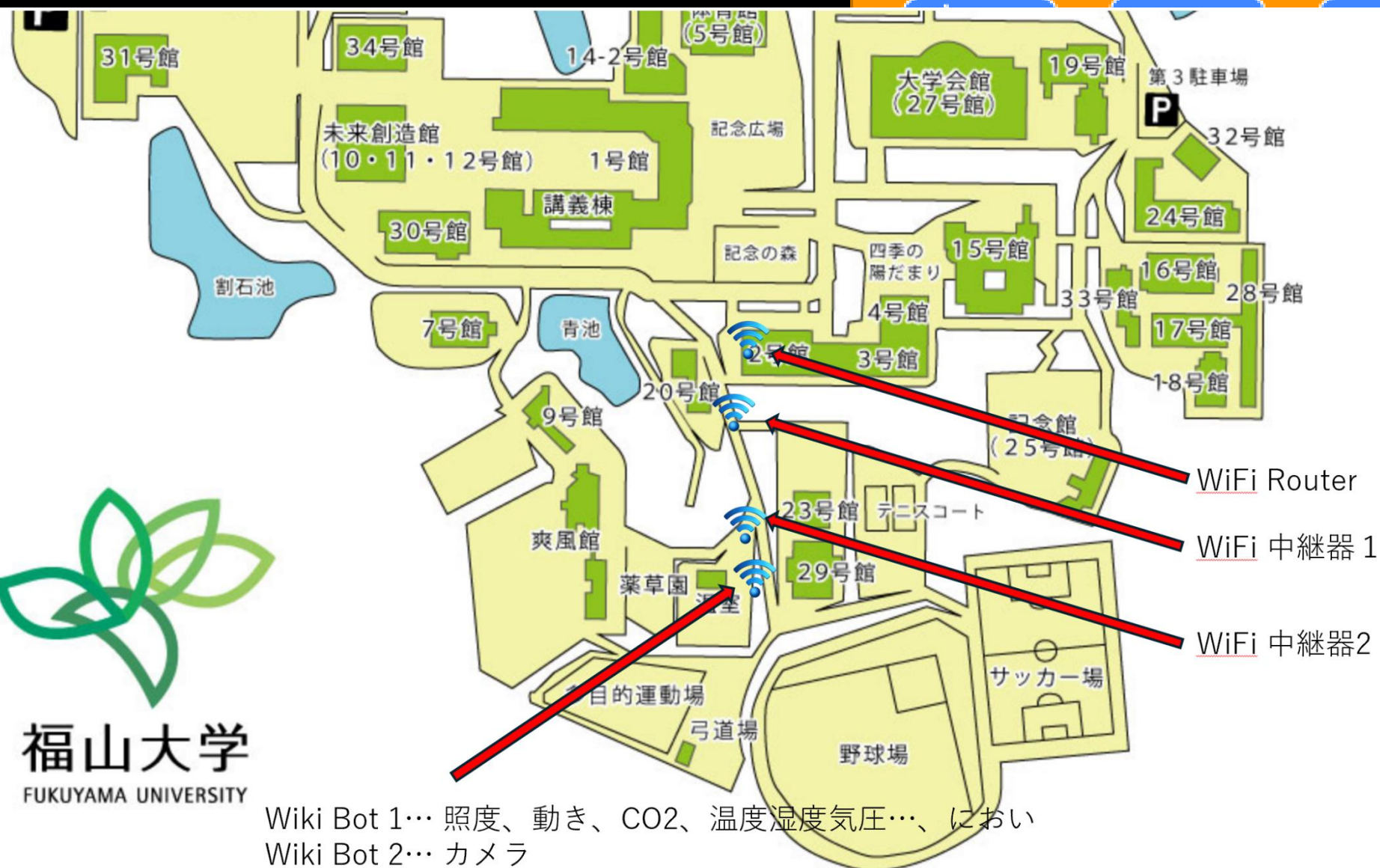
–（ハートマークのクリック、よろしくお願いいたします）



・ スマート
農業に
使える
かな？



福山大学
FUKUYAMA UNIVERSITY





index

FrontPage

simple example

music

NetDraw

basic

pen

最新の20件

2025-01-27

- sensor_01_20241007
- sensor_03_20241031
- sensor_02_20241007
- sensor_05_202412130001
- sensor_04_20241120
- capture_20250115

2025-01-25

- map01
- Value_insertion_01_20241007
- capture_20250125

2025-01-15

- tDressBot_ex1

2025-01-06

```
object_page http://www.██████████?capture_20250115 or http://www.██████████?capture_20250115
device yama_bot_0000_0000_0000_0001 or yama_bot_0000_0000_0000_0001 start after no write for 10 min.
command: set read_interval=300000
command: set exec_interval=5000
command: set report_length=200
command: py test
py: import subprocess
py: import time
py: date=self.Pico_Time.get_now()
py: #date=2024-12-31 14:25:16.61944
py: time_x=(date.split())[1]
py: #print('time_x='+time_x)
py: time_xx=time_x.split(':')
py: hm=time_xx[0]+'_'+time_xx[1]
py: ms=time_xx[1]
py: ss=time_xx[2]
py: dec_sec=ss[0]
py: #print('hm='+hm+' ms='+ms+' ss='+ss+' desec='+dec_sec)
py: if (ms[1]!='0' or ms[1]!='5') and dec_sec=='0': #when **:00:0*. or **:5:0*.
py:     print('time='+time_x)
py:     line="device=camera_time*, Date="+self.Pico_Time.get_now()
py:     self.write_line(line)
py:     if ms=='00': #when **:00
py:         print('time='+hm)
py:         attach_dir='/home/pi/python/'
py:         attach_name=hm+'.jpg'
py:         print('attach_name='+attach_name)
py:         cmd_line="/usr/bin/libcamera-jpeg -o "+attach_dir+attach_name+" -t 1000 --width 640 --height 480"
py:         print(cmd_line)
py:         cmd_line_list=cmd_line.split()
py:         subprocess.run(cmd_line_list)
py:         list=self.get_attachment_list()
py:         if attach_name in list:
py:             self.delete_attachment(attach_name)
py:         self.upload_file(attach_dir, attach_name)
py:         line="device=camera, Date="+self.Pico_Time.get_now()+"", v="+attach_name
py:         self.write_line(line)
py:         time.sleep(6)
py:         self.send_result()
py:         time.sleep(1)
command: end test
command: run test
result:
device=camera_time*, Date=2025-01-25 21:20:05.542581
device=camera_time*, Date=2025-01-25 21:25:06.596673
device=camera_time*, Date=2025-01-25 21:30:06.767833
device=camera_time*, Date=2025-01-25 21:35:07.096213
device=camera_time*, Date=2025-01-25 21:40:07.498304
```



Wikiのスク립ト
例: Wiki Bot 2...
カメラ



```
device=camera_time*, Date=2025-01-27 17:00:00.122969  
device=camera, Date=2025-01-27 17:00:09.245927, v=17_00.jpg  
device=camera_time*, Date=2025-01-27 17:05:09.879690  
device=camera_time*, Date=2025-01-27 17:10:05.923669  
device=camera_time*, Date=2025-01-27 17:15:06.053327  
device=camera_time*, Date=2025-01-27 17:20:05.647912  
device=camera_time*, Date=2025-01-27 17:25:00.483535  
device=camera_time*, Date=2025-01-27 17:30:05.164501  
device=camera_time*, Date=2025-01-27 17:35:01.391589  
device=camera_time*, Date=2025-01-27 17:40:05.940672  
device=camera_time*, Date=2025-01-27 17:45:05.898760  
device=camera_time*, Date=2025-01-27 17:50:07.995644  
current_device="yama_bot_0000_0000_0000_0001". Date=2025-01-27 17:50:08.603398
```

・ 先のページ
の下の方

添付ファイル: [17_00.jpg](#) [詳細] [16_00.jpg](#) [詳細] [15_00.jpg](#) 1件 [詳細] [14_00.jpg](#) [詳細] [13_00.jpg](#) 1件 [詳細] [09_00.jpg](#) 1件 [詳細] [03_00.jpg](#) [詳細] [02_00.jpg](#) [詳細] [01_00.jpg](#) [詳細] [00_00.jpg](#) 1件 [詳細] [21_00.jpg](#) [詳細] [20_00.jpg](#) [詳細] [19_00.jpg](#) [詳細] [18_00.jpg](#) 2件 [詳細] [08_00.jpg](#) 3件 [詳細] [05_00.jpg](#) 2件 [詳細] [04_00.jpg](#) 2件 [詳細] [19_20.jpg](#) 5件 [詳細] [19_10.jpg](#) 6件 [詳細]

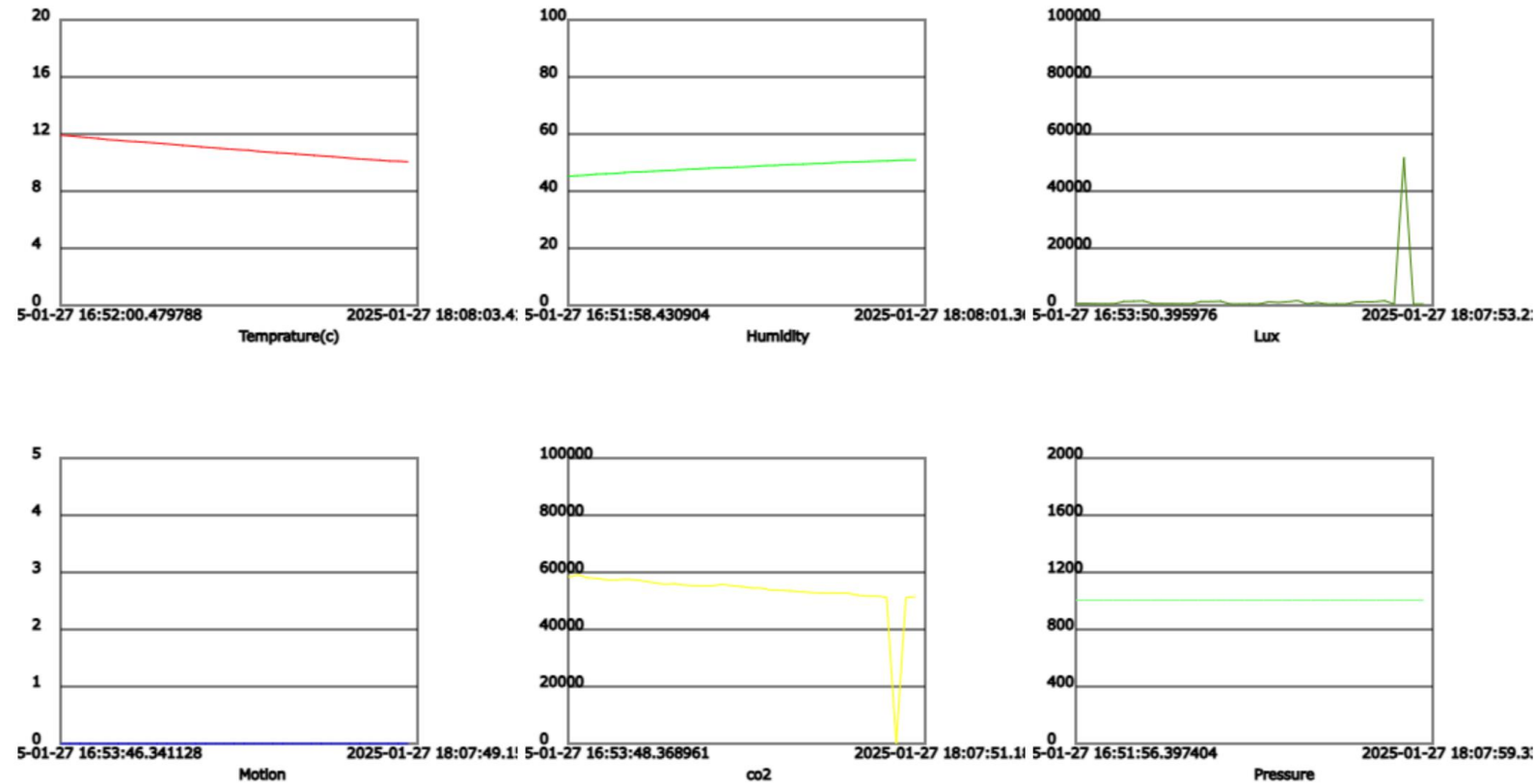




- ・ 17_00.jpgを
クリックすると



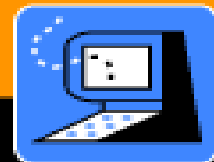
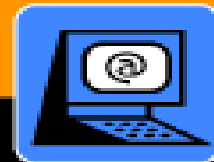
Graph Area



- Wiki Bot 1で
採取した
データ



謝辞



- ・ PukiWikiの関係者の皆様
- ・ 数多くの公開情報
- ・ 科研費 基盤研究(C)21K11858

