

# 学位論文

Title

提出年月日 令和 4 年 X 月 X 日

改定日 令和 年 月 日

指導教員 都築 伸二 教授

入学年度 令和 X 年

学科名 電子情報工学専攻

論文提出者 XX YY

# 内容梗概

本論文は、筆者が愛媛大学大学院理工学研究科電子情報工学専攻電気電子工学コースに在学中に行った、…についてまとめたものであり、以下の5章から構成されている。

## 第1章 緒論

本研究を行うに至った経緯及び、本研究の目的について述べている。

## 第2章 2章タイトル

ここでは…について述べる。

## 第3章 3章タイトル

ここでは、…について述べている。

## 第4章 4章タイトル

ここでは、…について述べている。

## 第5章 結論

本研究によって明らかになった事項や今後の研究課題について簡単にまとめている。

# 目次

内容梗概	I
第 1 章 緒論	1
第 2 章 2 章のタイトル	2
2.1 節 緒言 . . . . .	2
2.2 節 ... . . . .	2
2.3 節 結言 . . . . .	2
第 3 章 3 章のタイトル	3
3.1 節 緒言 . . . . .	3
3.2 節 結言 . . . . .	3
第 4 章 4 章のタイトル	4
4.1 節 緒言 . . . . .	4
4.2 節 結言 . . . . .	4
第 5 章 結論と今後の課題	5
謝 辞	6
参考文献	7
付録 A 付録	8
A.1 水源監視システム (送信用 Python スクリプト) . . . . .	8
A.2 水源監視システム (受信用 Python スクリプト) . . . . .	14

# 第1章

## 緒論

… 第2章では…について述べる．第3章では……第5章では結論と課題を述べる。

## 第2章

## 2章のタイトル

### 2.1 節 緒言

本章では … について述べる .

### 2.2 節 …

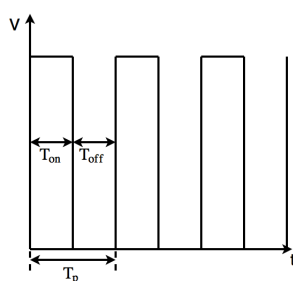


図 2.1 example

### 2.3 節 結言

本章 … について述べた。

次章では … について述べる。

## 第3章

### 3章のタイトル

#### 3.1 節 緒言

本章では、… について述べる。

#### 3.2 節 結言

本章では、… について述べた .

## 第4章

### 4章のタイトル

#### 4.1 節 緒言

本章では...について述べる .

#### 4.2 節 結言

本章では...について述べた .

## 第5章

### 結論と今後の課題

.....



## 謝 辞

本研究を行うにあたり、終始、懇切丁寧な御指導と適切な御助言を賜りました本学工学部電気電子工学科通信システム工学研究室の都築伸二教授に深甚なる感謝の意を表します。

最後に、有益な御助言を賜りました本大学大学院の〇〇に心より御礼申し上げます。

## 参考文献

- [3] 著者, タイトル, 引用日など.

## 付録 A

## 付録

### A.1 水源監視システム (送信用 Python スクリプト)

LoRa\_obs\_transmit.py のソースコードを A.1 に示す.

Listing A.1 LoRa\_obs\_transmit.py

---

```

1  ## *****coding:utf-8*****
2
3  import time
4  import os
5  from datetime import datetime
6  import serial
7
8  """ sleep()をいれて, 少し待たないとエラー落ちする """
9  time.sleep(60)
10
11 class Main() :
12
13     def __init__(self):
14         """ 初期値および対象ディレクトリの設定 """
15         self.s_num = 0
16
17         self.copy_dir = "C:/Users/taikimizukan/Dropbox/sumitomo
18         /obs_csv/"
19         self.target_dir = "C:/Users/taikimizukan/Desktop/
20         obs_data/"

```

```
19         self.temporary_log = "./temporary_log.txt"
20
21         """ 最終データを取得 """
22         with open(self.temporary_log,"r") as f :
23             self.old_line = f.readline()
24             print("前回のデータ:"+str(self.old_line))
25
26         """ 起動時 [$RFINF,ONコマンド送信***] """
27         INF = "$RFINF,ON***"
28         with serial.Serial("COM7",115200,timeout=2) as ser :
29             time.sleep(2)
30             while True :
31                 for i in INF :
32                     ser.write(i.encode("utf-8"))
33
34                     result = str(ser.readline())
35                     if result.find("RESULT,RFINF,ON,OK") > 0 :
36                         break
37                     else :
38                         time.sleep(2)
39
40         """ ループ関数実行 """
41         self.Loop()
42
43
44         """ 作成日時が最新ファイルのフルパスを取得し返す関数 """
45         def get_file_path(self, target_dir) :
46             """ 対象ディレクトリ下の . ファイルのパスを取得し
47                 dat, [target_filesに納める] """
48             target_files = []
49             for root, dir, files in os.walk(target_dir) :
50                 target_file = [os.path.join(root,f) for f in files
51                               if f.endswith(".dat")]# .txt -> .へ
52                 target_files.extend(target_file)
53             """ 取得した . ファイルのフルパスに作成時間を足してリストに納める
54                 dat """
55             file_ctime = []
56             for f in target_files :
57                 file_ctime.append((f,os.path.getctime(f)))
58             """ 取得時間でソートし最新の . ファイルのパスのみ返す dat """
59             sorted_file_ctime = sorted(file_ctime,key=lambda x :x
```

```
[1])

57
58         return sorted_file_ctime[len(sorted_file_ctime)-1][0]
59
60         """ 最終行を取得，シーケンス番号を加えてコピー """
61     def check_copy(self):
62         ##         name = self.target_file.replace(self.target_dir, "")
63         with open(self.target_file, "r") as f :
64             """ ファイルデータを全て読み込，最終行だけを取得 """
65             lines = f.readlines()
66             if len(lines) > 0 :
67                 line = lines[len(lines)-1]
68             else :
69                 line = self.old_line
70
71         """ 最終行が前回のものと異なるか? """
72         if line != self.old_line :
73
74             """ シーケンス番号を追加 """
75             self.s_num += 1
76
77             file_name = line.split(",")
78             file_name = "obs_" + "".join(file_name[0:3])
79             self.ymd = "".join(file_name[0:3])
80
81             self.old_line = line
82
83             """ にコピーDropbox """
84             with open(self.copy_dir+file_name+".csv", "a") as cf
85                 :
86                     data = line.strip() + "," + str(self.s_num)+"\n"
87                     "
88                     cf.write(str(data))
89
90             """ 最終行を保存 """
91             with open(self.temporary_log, "w") as f :
92                 f.write(line)
93
94             self.arduino_serial(data)
95             time.sleep(5)
96             self.TxMSG()
```

```

95         self.Ping()
96
97     else :
98         print("Not updated")
99         pass
100
101     """ を経由してにデータを送信する関数  arduinoLoRa """
102     def arduino_serial(self,d) :
103         print("---"*5 + "arduino_serial" + "---"*5)
104         buf = 0
105         with serial.Serial("COM7",115200,timeout=1) as ser :
106             """ ポートを開いて少し待機が必要 """
107             time.sleep(2)
108             """ごみの吸出し """
109             buf = ser.readlines()
110             d = d.strip()
111             """ 送信コマンドの形に """
112             d = "$RFSND,0004,"+d+"***"
113             print("To_arduino_Data-->" +d)
114
115             """ Python(PC) -> arduino -> LoRa だと文字ずつ送らないと
116                 いけない? 1 """
117             for i in d :
118                 ser.write(i.encode("utf-8"))
119
120             """ 0009 : 第二中継機にダミーをとばすMSG, 戻り値を保存 """
121     def TxMSG(self) :
122         target_add = "0009"
123         self.now = datetime.now().strftime("%Y,%m,%d,%H,%M,%S")
124         self.today = datetime.today().strftime("%Y%m%d")
125
126         msg = "$RFSND,{0},{1},{2},{2}***".format(target_add,
127             self.now,self.counter)
128
129         with serial.Serial("COM7",115200,timeout=15) as ser :
130             time.sleep(2)
131             for i in msg :
132                 ser.write(i.encode("utf-8"))
133                 time.sleep(0.05)
134             print(ser.readline().decode("utf-8"))
135             res = ser.readline().decode("utf-8")
```

```
134
135         if len(res) > 10 :
136             res = res.replace("□","").replace("*","").
                  replace(":","")
137             with open("C:/Users/taikimizukan/Dropbox/
                  sumitomo/RSSI_CHECK_TX/rssi_tx_obs_{}.csv".
                  format(str(self.today)), "a") as f :
138                 f.write(res+"\n")
139         else :
140             pass
141
142         """発電所のにを送って生存確認LoRaping"""
143     def Ping(self) :
144         PING = "$RPING,0004***"
145         with serial.Serial("COM7",115200,timeout=10) as ser :
146             time.sleep(2)
147             for i in PING :
148                 ser.write(i.encode("utf-8"))
149                 time.sleep(0.05)
150
151             print(ser.readline().decode("utf-8"))
152             res_ping = ser.readline().decode("utf-8")
153
154             if len(res_ping) > 10 :
155                 print(res_ping)
156                 now = datetime.now().strftime("%Y,%m,%d,%H,%M,%S")
157                 with open("C:/Users/taikimizukan/Dropbox/sumitomo/
                  PING/ping_{}.csv".format(str(self.today)), "a") as
                  f :
158                     f.write(str(now)+","+str(self.s_num)+"," +
                  str(res_ping))
159             else :
160                 pass
161
162         """ 繰り返し """
163     def Loop(self):
164         while True:
165             try :
166                 time.sleep(20)
167                 self.target_file = self.get_file_path(self.
                  target_dir)
```

```
168
169             self.check_copy()
170
171         except Exception as E :
172             with open("./Error_Log.txt","a") as ef :
173                 ef.write(str(E))
174
175
176 if __name__ == "__main__" :
177     Main()
```



## A.2 水源監視システム (受信用 Python スクリプト)

LoRa\_obs\_receive.py のソースコードを A.2 に示す .

Listing A.2 LoRa\_obs\_raceive.py

```
1  ## coding:utf-8
2
3  import paho.mqtt.client as mqtt
4  import serial
5  from datetime import datetime
6  import time
7
8  class Main() :
9      def __init__(self) :
10         self.INF_Input()
11         self.Loop()
12
13     def INF_Input(self) :
14         """ 起動時 [$RFINF,ONコマンド送信***] """
15
16         INF = "$RFINF,ON***"
17         with serial.Serial("COM3",115200,timeout=2) as ser :
18             time.sleep(2)
19             while True :
20                 for i in INF :
21                     ser.write(i.encode("utf-8"))
22
23                 result = str(ser.readline())
24                 if result.find("RESULT,RFINF,ON,OK") > 0 :
25                     print("RFINF,OK")
26                     break
27                 else :
28                     time.sleep(2)
29
30     def LoRa_Receive(self) :
31         try :
32             """ からデータを読み込むLoRa """
33             while True :
34                 with serial.Serial("COM3",115200,timeout=120)
35                     as ser :
```

```
35         res = ser.readline().decode("utf-8")
36         if len(res) > 15 and res.find("RFRX") > 0
           and res.find("RECEIVED") < 0 :
37             print("==="*25)
38             print("RXData_<_<=>_<_"+res)
39             break
40
41         else :
42             print("==="*25)
43             print("else_data=>"+res)
44
45         """ 必要なデータを取り出す """
46         res_list = res.split("*")[0].split(",")[1:]
47         data = ",".join(res_list)
48         add = res_list[0]
49
50         """ データの日付を確認(ymd) """
51         ymd = ",".join(res_list[1:4])
52
53
54         return = res, data, add, ymd
55
56     except Exception as E :
57         now = datetime.now().strftime("%y/%m/%d_%H:%M:%S")
58         with open("LoRa_Receive_Error_LOG.txt","a") as ef :
59             ef.write(now + "_:_"+ str(E)+"\n")
60         self.Loop()
61
62
63
64     def MQTT_Publish(self, res):
65         """ 情報MQTT(publish) """
66         host = "133.71.***.***"
67         port = 1883
68         topic = "*****"
69         """ MQTT-Publish """
70         try :
71             print("publish_<_<=>_<_"+str(res))
72             client = mqtt.Client(protocol=mqtt.MQTTv311)
73             client.connect(host,port=port,keepalive=10)
74             client.publish(topic,res)
```

```
75
76         except Exception as E :
77             now = datetime.now().strftime("%y/%m/%d_%H:%M:%S")
78             with open("Publish_Error_LOG.txt","a") as ef :
79                 ef.write(now + "_:" + str(E)+"\n")
80
81     def Storage(self,res,data,ymd):
82         try :
83             """ メタデータを保存 """
84             with open("./meta_data/meta_data_{}.txt".format(ymd)
85                     ),"a") as f :
86                 f.write(res+"\n")
87
88             """ データを保存 """
89             with open("./data/data_{}.txt".format(ymd),"a") as
90                 f :
91                 f.write(data+"\n")
92
93             """ に保存Dropbox """
94             with open("C:/Users/sumitomo02/Dropbox/test_folder/
95                     RX/DATA/RX_{}.txt".format(ymd),"a") as f :
96                 f.write(data+"\n")
97
98             with open("C:/Users/sumitomo02/Dropbox/test_folder/
99                     RX/META_DATA/RX_{}.txt".format(ymd),"a") as f :
100                 f.write(res+"\n")
101
102         except Exception as E :
103             now = datetime.now().strftime("%y/%m/%d_%H:%M:%S")
104             with open("Storage_Error_LOG.txt","a") as ef :
105                 ef.write(now + "_:" + str(E)+"\n")
106         print(E)
107
108     def Loop(self) :
109         while True :
110             res, data, add, ymd = self.LoRa_Receive()
111             self.Storage(res, data, ymd)
112             self.MQTT_Publish(res)
113             self.Ping(add)
```

```
112 main = Main()
```