

```
# FEB_report_out.py
```

```
# Author: Luca Ghislotti
```

```
# Version: 1.7.1
```

```
from __future__ import print_function
from audioop import bias
from pickle import FALSE, TRUE
from mailmerge import MailMerge
from docx2pdf import convert
```

```
from FEB_report_fun import (
    report_ENC,
    report_pedestal,
    report_temperature,
    report_thrdisp,
    text_to_pdf,
    get_bias_data,
    read_config_file,
    defect_notes,
)
```

```
from FEB_report_fun import ftxt_a, ftxt_w, ftxt_r, ftxt_w1, ftxt_w2
import re
import math
from matplotlib import lines
import os
from pathlib import Path
```

```
# read files from directory
```

```
def print_report(num_report):
```

```
    flag = False
    temp_data = []
    ENC_data = []
    thrdisp_data = []
    pedestal_data = ""
```

```
# file numbering based on predefined format
```

```
if num_report < 10:
    intermediate_path = "MODULE_00" + str(num_report)
    show = "F00" + str(i) + "I"
elif num_report < 100:
    intermediate_path = "MODULE_0" + str(num_report)
    show = "F0" + str(i) + "I"
else:
    intermediate_path = "MODULE_" + str(num_report)
    show = "F" + str(i) + "I"
```

```
intermediate_path = intermediate_path + "/1/"
```

```
# temperature file
```

```
dir_temp = os.path.dirname(__file__)
file_temp = os.path.join(
    dir_temp, "../modules/" + intermediate_path + "data/HK_Temperature.dat"
)
```

```
# noise ENC data
```

```
dir_ENC = os.path.dirname(__file__)
file_ENC = os.path.join(
    dir_ENC,
    "../modules/" + intermediate_path + "analysis_matlab/ENC/normal/ENC_normal.dat",
)
```

```
# threshold dispersion data
```

```
dir_thr = os.path.dirname(__file__)
file_thr = os.path.join(
    dir_thr,
    "../modules/"
```

```

+ intermediate_path
+ "analysis_matlab/ThresholdScan/Threshold_dispersion.dat",
)

# pedestal mean
dir_ped = os.path.dirname(__file__)
file_ped = os.path.join(
    dir_ped, "../modules/" + intermediate_path + "data/Pedestals.dat"
)

# write to log file
if Path(file_temp).is_file():
    ftxt_a.write("***** MODULE " + show + " *****\n")
    temp_data = report_temperature(file_temp)
    flag = True

# acquire ENC data
if Path(file_ENC).is_file():
    ENC_data = report_ENC(file_ENC)

# acquire threshold dispersion data
if Path(file_thr).is_file():
    thrdisp_data = report_thrdisp(file_thr)

# acquire pedestal data
if Path(file_ped).is_file():
    pedestal_data = report_pedestal(file_ped)

return [flag, temp_data, ENC_data, thrdisp_data, pedestal_data]

# main call
start = int(input("Range START: "))
stop = int(input(" Range STOP: "))

# script_values.txt
dir_txt3 = os.path.dirname(__file__)
file_txt3 = os.path.join(dir_txt3, "../output/script_values.csv")
ftxt_w3 = open(file_txt3, "a")

# write MATLAB script-obtained data as output
ftxt_w3.write(
    "ENC_0,ENC_7,ENC_15,ENC_16,ENC_23,ENC_31,thrdisp_bef,thrdisp_aft,ped_disp,\n"
)

# FEB cycling
for i in range(start, stop + 1):
    report_data = print_report(i) # report data
    config_data = read_config_file() # configuration data
    report_notes = defect_notes(i) # notes on defects

# select template
document = MailMerge("../report_template/test_report_FEB.docx")

# FEB identifier formatting
if report_data[0]:
    if i < 10:
        ID_number = "00" + str(i)
    elif i < 100:
        ID_number = "0" + str(i)
    else:
        ID_number = str(i)

# write to terminal window during execution
print("\nMODULE F" + str(ID_number) + config_data[0])

# acquire bias measurements
bias_data = get_bias_data(i)

```

```
# write to .docx file (same fields as in template)
```

```
document.merge(  
    board_ID_title=ID_number,  
    nation_letter=config_data[0],  
    board_ID="F" + str(ID_number) + str(config_data[0]),  
    doc_version=config_data[1],  
    date=config_data[2],  
    author=config_data[3],  
    asic_ID=ID_number,  
    nation_word=config_data[4],  
    AVDD=bias_data[1],  
    IVDD=bias_data[2],  
    DVDD=bias_data[3],  
    IDVDD=bias_data[4],  
    treVtre=bias_data[5],  
    ltreVtre=bias_data[6],  
    lbias=bias_data[7],  
    VCMSH=bias_data[8],  
    VCM=bias_data[9],  
    RVCM=bias_data[10],  
    temp_ADC=report_data[1][0],  
    temp_T=report_data[1][1],  
    no_resp_ch="0",  
    ENC_0=report_data[2][0],  
    ENC_7=report_data[2][1],  
    ENC_15=report_data[2][2],  
    ENC_16=report_data[2][3],  
    ENC_23=report_data[2][4],  
    ENC_31=report_data[2][5],  
    thr_disp_bef=report_data[3][0],  
    thr_disp_aft=report_data[3][1],  
    ped_disp=report_data[4],  
    notes=report_notes,  
)
```

```
# write to .csv file
```

```
ftxt_w3.write(  
    str(report_data[2][0])  
    + ","  
    + str(report_data[2][1])  
    + ","  
    + str(report_data[2][2])  
    + ","  
    + str(report_data[2][3])  
    + ","  
    + str(report_data[2][4])  
    + ","  
    + str(report_data[2][5])  
    + ","  
    + str(report_data[3][0])  
    + ","  
    + str(report_data[3][1])  
    + ","  
    + str(report_data[4] + ",\n")  
)
```

```
# write to log file
```

```
ftxt_a.write("*** NOTES ***\n")  
ftxt_a.write("\n" + report_notes + "\n")
```

```
ftxt_w1.write(report_data[1][0] + ",\n")
```

```
ftxt_w2.write(report_data[1][1] + ",\n")
```

```
for i in range(1, 39):
```

```
    ftxt_a.write("\n")
```

```
# save .docx file
```

```
document.write("../report_word/F" + str(ID_number) + config_data[0] + ".docx")
```

```
# convert .docx to .pdf
```

```
convert(  
    "../report_word/F" + str(ID_number) + config_data[0] + ".docx",  
    "../report_PDF/F" + str(ID_number) + config_data[0] + ".pdf",  
)
```

```
# close file handlers
```

```
ftxt_w.close()
```

```
ftxt_a.close()
```

```
ftxt_w1.close()
```

```
ftxt_w3.close()
```

```
# export log
```

```
text = ftxt_r.read()
```

```
ftxt_r.close()
```

```
text_to_pdf(text, "../output/FEB_report_log.pdf")
```