

Program source code(Python) / プログラム ソースコード(Python)

(As Of May 22, 2022)

Shu JA3GQJ

```
<br>#--Incorporating libraries for makeing control panels<br>
#--制御パネル作成用ライブラリーの取り込み<br>
import tkinter as tk<br>
import tkinter.ttk as ttk<br><br>

#--Incorporating libraries for motor control, point control, position sensor<br>
#--モーター速度制御、ポイント制御、位置センサー用ライブラリーの取り込み<br>
from gpiozero import * #Motor, Button<br>
from time import sleep<br>
#--GPIO pin allocation (motor driver TA8428K) ###<br>
#--GPIO ピン割り付け (モータドライバ TA8428K) ###<br>
motor = Motor(20, 21) # GPIO20(pin38)を IN1 へ, GPIO21(pin40)を IN2 へ繋ぐ<br>
#--GPIO pin allocation (for Button position sensor)<br>
#--GPIO ピン割り付け (Button 位置センサー用) <br>
button1 = Button(7) # 位置センサーを GPIO7(pin26)へ<br>
button2 = Button(8) # 位置センサーを GPIO8(pin24)へ<br>
#button3 = Button(13) # 位置センサーを GPIO13(pin33)へ<br>
button4 = Button(16) # 位置センサーを GPIO16(pin36)へ<br>
button5 = Button(25) # 位置センサーを GPIO25(pin22)へ<br>
button6 = Button(26) # 位置センサーを GPIO26(pin37)へ<br>
button7 = Button(27) # 位置センサーを GPIO27(pin13)へ<br>
#--GPIO pin allocation (Motor driver TA8428K is used to the point switching)<br>
#--GPIO ピン割り付け ( モータドライバ TA8428K をポイント切り替えに使用) <br>
point1 = Motor(22, 23) # pin15, 16( GPIO22, GPIO23)<br>
point2 = Motor(17, 18) # pin11, 12(GPIO17, GPIO18)<br>
point3 = Motor(5, 6) # pin29, 31( GPIO5, GPIO6)<br><br>

#-----Check the state of the GPIO pin allocated when the program starts-----#<br>
#-----プログラム起動時に割り付けた GPIO ピンの状態をチェック-----#<br>
print('*****MOTOR*****')<br>
print('motor.value(pin20, 21)= ',motor.value)<br><br>

print('*****SENCER*****')<br>
print('button1.value(pin26)= ',button1.value)<br>
```

```

print('button2.value(pin24)= ',button2.value)<br>
print('button3.value(pin33)= *****' )<br>
print('button4.value(pin36)= ',button4.value)<br>
print('button5.value(pin35)= ',button5.value)<br>
print('button6.value(pin37)= ',button6.value)<br>
print('button7.value(pin13)= ',button7.value)<br><br>

print('*****POINT*****')<br>
print('point1.value(pin22,23)= ',point1.value)<br>
print('point2.value(pin17,18)= ',point2.value)<br>
print('point3.value(pin29,31)= ',point3.value)<br>
#-----Acceleration----- #<br>
def l_acc():<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    i=0<br>
    print('Train start from Yard to Main Track')<br>
    for i in range(0, maxspeed, adj):<br>
        speed=i/100<br>
        motor.backward(speed)<br>
        sleep(0.3)<br><br>

    return()<br>
def r_acc():<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    for i in range(0, maxspeed, adj):<br>
        speed=i/100<br>
        motor.forward(speed)<br>
        sleep(0.3)<br>
    return()<br>
#-----Decelerate----- #####<br>
def decr_1(): # CW 減速 1 段目<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    reduce=int(maxspeed*0.7)<br>
    i=0<br>
    for i in range(maxspeed, reduce, -adj):<br>

```

```

        speed=i/100<br>
        motor.forward(speed)<br>
        sleep(0.1)<br>
    return()<br>
def decr_2(): # CW 減速 2 段目<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    reduce=int(maxspeed*0.7)<br>
    i=0<br>
    for i in range(reduce, 0, -adj):<br>
        speed=i/100<br>
        motor.forward(speed)<br>
        sleep(0.08)<br>
    return()<br>
def decl_1(): # CCW 減速 1 段目<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    reduce=int(maxspeed*0.7)<br>
    i=0<br>
    for i in range(maxspeed, reduce, -adj):<br>
        speed=i/100<br>
        motor.backward(speed)<br>
        sleep(0.1)<br>
    return()<br>
def decl_2(): # CCW 減速 2 段目<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    reduce=int(maxspeed*0.7)<br>
    i=0<br>
    for i in range(reduce, 0, -adj):<br>
        speed=i/100<br>
        motor.backward(speed)<br>
        sleep(0.08)<br>
    return()<br>
#-----P_All_1-----#<br>
def p_all_1():<br>
    p1() # step1<br>
    p2() # step2

```

```

    p3() # step3<br>
    p4() # step4<br>
    p7() # step7<br>
    p8() # step8<br>
    return()<br>
#-----P_All_2-----#<br>
def p_all_2():<br>
    p1() # step1<br>
    p2() # step2<br>
    p3() # step3<br>
    p4() # step4<br>
    p5() # step5<br>
    p6() # step6<br>
    p7() # step7<br>
    p8() # step8<br>
    return()<br>
#-----P_All_3-----#<br>
def p_all_3():<br>
    return()<br>
#-----P_All_4-----#<br>
def p_all_4():<br>
    return()<br>
#-----Plan1-----#<br>
def p1():<br>
    point3_2() # ポイントを操作場の B 線へ切替<br>
    print('Point is switched to Yard_B')<br>
    point2_2() # ポイントを操作場切替<br>
    print('Pont is switched to Yard')<br>
    sleep(1)<br>
    print('Train start(CW) from Yard to Main Track')<br>
    r_acc()<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('*****Sencer1 is detected for Plat-home_1 *****', button1.value)<br>
    point1_1()<br>
    sleep(1.5) # 回送電車がセンサー1 を検出して 1 番線に停車させる調整時間<br>
    r_stop_1() # 回送電車が 1 番線に入線して停車<br>
    return()<br>
# -----Plan2----- #####<br>

```

```

def p2():<br>
    point2_2() # ポイントを操作場へ切替<br>
    print('Pont is switched to 2_2')<br>
    sleep(1)<br>
    point1_2() # ポイントを 2 番ホームへ切替<br>
    print('Point is switched to Plat-home_2')<br>
    sleep(1)<br>
    point3_1() # ポイントを操作場の A 線へ切替<br>
    print('Point is switched to Yard_A')<br>
    sleep(1)<br>
    r_acc() # CW 加速<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('*****Sencer1 is detected for Plat-home_2 *****', button1.value)<br>
    sleep(1.5)<br>
    r_stop_2() # 回送電車が 2 番線に入線して停車<br>
    print('Then Train will stop at Station(Plat-home_1)')<br>
    return()<br>
#-----Plan3-----<br>
def p3():<br>
    point2_1() # ポイントを本線に切り替えて<br>
    print('Point is switched to Main-Track')<br>
    sleep(1)<br>
    point1_1() # ポイントを 1 番線に切り替えると<br>
    print('Point is switched to Plat-home_1')<br>
    point3_1() # ポイントを操作場 A 線に切り替える<br>
    print('Point is changed to Yard_A')<br>
    sleep(1)<br>
    sleep(2) # しばらくして<br>
    print('After while,Train will start(CCW) from Plat-Home_1')<br>
    l_acc() # CCW 加速<br>
    button2.wait_for_press() # 位置センサー(2)を検出して<br>
    print('*****Sencer2 is detected for Plat-home_1 *****', button1.value)<br>
    l_stop_1() # 先ほど発車した電車が再び 1 番線に停車する<br>
    sleep(1)<br>
    return()<br>
#-----Plan4-----<br>
def p4():<br>
    point1_2()<br>

```

```

print('Point is changed to Plat-home_2')<br>
sleep(1)<br>
print('Then Train start')<br>
r_acc() # CW 加速<br>
button1.wait_for_press() # 位置センサー(1)を検出したら<br>
print('***** Sencer1 is detected for Plathome_2 *****', button1.value)<br>
#point1_2()<br>
#print('Point is changed to Plathome_2')<br>
sleep(1)<br>
r_stop_2() # 電車が2番線に入線して停車<br>
sleep(1)<br>
return()<br>
#-----Plan5----- #####<br><br>
def p5():<br><br>
    point1_1()<br>
    print('Point is changed to Plat-home_2')<br>
    sleep(1)<br>
    print('Then Train start')<br>
    r_acc() # CW 加速<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('***** Sencer1 is detected for Plathome_2 *****', button1.value)<br>
    sleep(1)<br>
    r_stop_1() # 電車が2番線に入線して停車<br>
    sleep(1)<br>
    return()<br>
#-----Plan6----- #####<br>
def p6():<br>
    point1_2() # ポイントを1番線に切り替えると<br>
    print('Point is switched to Plat-home_1')<br>
    point3_1() # ポイントを操作場A線に切り替える<br>
    print('Point is changed to Yard_A')<br>
    sleep(1)<br>
    sleep(2) # しばらくして<br>
    print('After while,Train will start(CCW) from Plat-Home_1')<br>
    l_acc() # CCW 加速<br>
    button2.wait_for_press() # 位置センサー(2)を検出して<br>
    print('*****Sencer2 is detected for Plat-home_1 *****', button1.value)<br>
    l_stop_2() # 先ほど発車した電車が再び1番線に停車する<br>

```

```

    sleep(1)<br>
    return()<br>
#-----Plan7----- #####<br>
def p7():<br>
    point1_1()<br>
    print('Pont is switched to 1_1')<br>
    sleep(1)<br>
    point2_2() # ポイントを Yard へ切替<br>
    print('Pont is switched to Tard')<br>
    sleep(1)<br>
    point3_1() # ポイントを Yard の A 線へ切替<br>
    print('Point is changed to Yard_A')<br>
    sleep(1)<br>
)    l_acc() # CCW 加速<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('*****Sencer1 is detected for Yard_A *****', button1.value)<br>
    sleep(0.1) # センサー1 検出からセンサー6 検出までの時間調整<br>
    l_stop_3() # 減速して Yard の A 線停車<br>
    sleep(1)<br>
    return()<br>
#-----Plan8----- #####<br>
def p8():<br>
    point1_2()<br>
    print('Pont is switched to 1_1')<br>
    sleep(1)<br>
    point2_2() # ポイントを Yard へ切替<br>
    print('Pont is switched to Tard')<br>
    sleep(1)<br>
    point3_2() # ポイントを Yard の A 線へ切替<br>
    print('Point is changed to Yard_A')<br>
    sleep(1)<br>
    l_acc() # CCW 加速<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('*****Sencer1 is detected for Yard_A *****', button1.value)<br>
    sleep(0.1) # センサー1 検出からセンサー6 検出までの時間調整<br><br>
    l_stop_4() # 減速して Yard の A 線停車<br>
    sleep(1)<br>
    return()<br>

```

```

#-----Plan9----- #####<br>
def p9():<br>
    point3_2() # ポイントを操作場の B 線へ切替<br>
    print('Point is switched to Yard_B')<br>
    point2_2() # ポイントを操作場切替<br>
    print('Pont is switched to Yard')<br>
    sleep(1)<br>
    print('Train start(CW) from Yard to Main Track')<br>
    r_acc()<br>
    button1.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('*****Sencer1 is detected for Plat-home_1 *****', button1.value)<br>
    point1_1()<br>
    sleep(1.5) # 回送電車がセンサー1 を検出して 1 番線に停車させる調整時間<br>
    r_stop_1() # 回送電車が 1 番線に入線して停車<br>
    *****<br>
    point2_1() # ポイントを本線に切り替えて<br>
    print('Point is switched to Main-Track')<br>
    sleep(2) # しばらくして<br>
    print('After while,Train will start(CCW) from Plat-Home_1')<br>
    r_acc() # CCW 加速<br>
    button1.wait_for_press() # 位置センサー(2)を検出して<br>
    print('*****Sencer1 is detected for Plat-home_1 *****', button1.value)<br>
    point1_2()<br>
    r_stop_2() # 先ほど発車した電車が再び 1 番線に停車する<br>
    sleep(1)<br>
    p10()<br>
    return()<br>
#-----Plan10----- #####<br>
def p10():<br>
    point3_1() # ポイントを操作場の B 線へ切替<br>
    print('Point is switched to Yard_B')<br>
    point2_2() # ポイントを操作場切替<br>
    print('Pont is switched to Yard')<br>
    sleep(1)<br>
    point1_1()<br>
    sleep(1)<br>
    print('Train start(CW) from Yard to Main Track')<br>
    r_acc()<br>

```

```

button1.wait_for_press() # 位置センサー(1)を検出したら<br>
print('*****Sencer1 is detected for Plat-home_1 *****', button1.value)<br>
sleep(1.5) # 回送電車がセンサー1を検出して1番線に停車させる調整時間<br>
r_stop_1() # 回送電車が1番線に入線して停車<br>
#<br>
point1_2()<br>
print('Pont is switched to Plathome_2')<br>
sleep(1)<br>
point2_2() # ポイントをYardへ切替<br>
print('Pont is switched to Yard')<br>
sleep(1)<br>
point3_2() # ポイントをYardのB線へ切替<br>
print('Point is changed to Yard_B')<br>
sleep(1)<br>
l_acc() # CCW 加速<br>
button1.wait_for_press() # 位置センサー(1)を検出したら<br>
print('*****Sencer1 is detected for Yard_B *****', button1.value)<br>
sleep(0.3) # センサー1検出からセンサー7検出までの時間調整<br>
l_stop_4() # 減速してYardのB線停車<br>
sleep(1)<br>
p11()<br>
return()<br>
#-----Plan11-----<br>
def p11():<br>
    point2_1() # ポイントを本線に切り替えて<br>
    print('Point is switched to Main-Track')<br>
    sleep(1)<br>
    point1_1() # ポイントを1番線に切り替えると<br>
    print('Point is switched to Plat-home_1')<br>
    point3_1() # ポイントを操作場A線に切り替える<br>
    print('Point is changed to Yard_A')<br><br>
    sleep(1)<br>
    sleep(2) # しばらくして<br>
    print('After while,Train will start(CCW) from Plat-Home_1')<br>
    l_acc() # CCW 加速<br>
    button2.wait_for_press() # 位置センサー(2)を検出して<br>
    print('*****Sencer2 is detected for Plat-home_1 *****', button1.value)<br>

```

```

l_stop_1() # 先ほど発車した電車が再び1番線に停車する<br>
sleep(1)<br>
p12()<br>
return()<br>
#-----Plan12----- #####<br>
def p12():<br>
    point3_1() # ポイントを操作場のB線へ切替<br>
    print('Point is switched to Yard_B')<br>
    point2_2() # ポイントを操作場切替<br>
    print('Pont is switched to Yard')<br>
    sleep(1)<br>
    point1_1()# ポイントを1切替<br>
    sleep(1)<br>
    print('Train start(CW) from Platform1')<br>
    r_acc()<br>
    button2.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('***** Sencer2 is detected then train stop *****', button1.value)<br>
    sleep(1.5) # 回送電車がセンサー2を検出して停車させる調整時間<br><br>

    decr_1() # 減速1段目<br>
    sleep(0.05)<br>
    decr_2() # 減速2段目<br>
    sleep(0.05)<br>
    motor.stop()<br>
#<br>
    point1_2()<br>
    print('Pont is switched to Plathome_2')<br>
    sleep(1)<br>
    point2_1() # ポイントをYardへ切替<br>
    print('Pont is switched to Main track')<br>
    sleep(1)<br>
    point3_2() # ポイントをYardのB線へ切替<br>
    print('Point is changed to Yard_B')<br>
    sleep(1)<br>
    l_acc() # CCW 加速<br>
    button2.wait_for_press() # 位置センサー(1)を検出したら<br>
    print('***** Sencer2 is detected for Yard_B *****', button2.value)<br>
    sleep(0.3) # センサー1¥2 検出からセンサー5 検出までの時間調整<br>

```

```

l_stop_5() # 減速して Yard の B 線停車<br>
sleep(1)<br>
#p11<br>
return()<br>
#-----Plan13----- #####<br><br>
走行パターンを追加する予定<br>
#-----Plan14----- #####<br>
走行パターンを追加する予定<br>
#-----右回り 1 番線停止用 2 段減速-----#<br>
def r_stop_1():<br>
    decr_1() # 減速 1 段目<br>
    button4.wait_for_press()<br>
    print('Sencer4 detected', button4.value)<br>
    decr_2() # 減速 2 段目<br>
    sleep(0.05)<br>
    motor.stop()<br>
    return()<br>
#-----右回り 2 番線停止用段減速--- ----#<br>
def r_stop_2():<br>
    decr_1() # 減速 1 段目<br>
    button5.wait_for_press()<br>
    print('Sencer5 detected', button5.value)<br>
    decr_2() # 減速 2 段目<br>
    sleep(0.05)<br>
    motor.stop()<br>
    return()<br>
#-----左回り 1 番線停止用 2 段減速-----#<br>
def l_stop_1():<br>
    decl_1() # 減速 1 段目<br>
    button4.wait_for_press()<br>
    print('Sencer4 detected', button4.value)<br>
    decl_2() # 減速 2 段目<br>
    sleep(0.05)<br>
    motor.stop()<br>
    return()<br>
#-----左回り 2 番線停止用 2 段減速-----#<br>
def l_stop_2():<br>
    decl_1() # 減速 1 段目<br>

```

```

    button5.wait_for_press()<br>
    print('Sencer5 detected', button5.value)<br>
    decl_2() #減速 2 段目<br>
    sleep(0.1)<br>
    motor.stop()<br>
    return()<br>
#-- -----左回り操作場 A 線停車用 2 段減速--- ----#<br>
def l_stop_3():<br>
    decl_1() # 減速 1 段目<br>
    button6.wait_for_press()<br>
    print('***** Sencer6 detectedfor stop Yard_A *****')<br>
    sleep(0.05)<br>
    decl_2()<br>
    sleep(0.1)<br>
    motor.stop()<br>
    return()<br>
#-- -----左回り操作場 B 線停車用 2 段減速--- ----#<br>
def l_stop_4():<br>
    decl_1() # 減速 1 段目<br>
    button7.wait_for_press()<br>
    print('*****Sencer7 detected for stop Yard_B *****')<br>
    sleep(0.03)<br>
    decl_2()<br>
    sleep(0.1)<br>
    motor.stop()<br>
    return()<br>
#----- ---point1-1---1 番線用----- #<br>
def point1_1():<br>
    point1.backward(1.0) # 上りポイント切り替え<br>
    sleep(0.3)<br>
    point1.stop() # point.forward(0.0)<br>
    sleep(0.2)<br>
    return()<br>
#----- ---point1-2---2 番線用----- #
def point1_2():<br>
    point1.forward(1.0) # 2 番線へポイント切り替え<br>
    sleep(0.5)<br>
    point1.stop()<br>

```

```

    sleep(0.2)<br>
    return()<br>
#-----point2-1---操作場から本線へ-----#<br>
def point2_1():<br>
    point2.backward(1.0) # 本線へポイント切り替え<br>
    sleep(0.5)<br>
    point2.stop()<br>
    sleep(0.2)<br>
    return()<br>
#-----point2-2---本線から操作場へ----- #<br>
def point2_2():<br>
    point2.forward(1.0) # ヤードへポイント切り替え<br>
    sleep(0.5)<br>
    point2.stop() # point.backward(0.0)<br>
    sleep(0.2)<br>
    return()<br>
#----- ---point3-1---操作場 A 線用----- #<br>
def point3_1():<br>
    point3.forward(1.0) # 操作場 A 線へポイント切り替え<br>
    sleep(0.5)<br>
    point3.stop()<br>
    sleep(0.2)<br>
    return()<br>
#-----point3-2---操作場 B 線用----- #<br>
def point3_2():<br>
    point3.backward(1.0) # 操作場 B 線へポイント切り替え<br>
    sleep(0.5)<br>
    point3.stop()<br>
    sleep(0.2)<br>
    return()<br>
#-----右回り走行（手動）----- #<br>
def r_manual():<br>
    maxspeed = int(text_32.get())    print('Manual Inbound Train Start-CW')<br>
    adj = int(maxspeed//20)<br>
    i=0<br>
    for i in range(0, maxspeed, adj):<br>
        speed=i/100<br>
        motor.forward(speed)<br>

```

```

    sleep(0.01)<br>
# -----右回り走行停止（手動）----- #<br>
def r_stop_manual():<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    i=0<br>
    for i in range(maxspeed, 10, -adj):<br>
        speed=i/100<br>
        motor.forward(speed)<br>
        sleep(0.01)<br>
    motor.stop()<br>
#-----l_manual----- #<br>
def l_manual():<br>
    maxspeed = int(text_32.get())<br>
    print('Manual Inbound Train Start-CCW')<br>
    adj =int(maxspeed//20)<br>
    i=0<br>
    for i in range(0, maxspeed, adj):<br>
        speed=i/100<br>
        motor.backward(speed)<br>
        sleep(0.01)<br>
#-----l_stop_manual-----#<br>
def l_stop_manual():<br>
    maxspeed = int(text_32.get())<br>
    adj = int(maxspeed//20)<br>
    i=0<br>
    for i in range(maxspeed, 10, -adj):<br>
        speed=i/100<br>
        motor.backward(speed)<br>
        sleep(0.01)<br>
    motor.stop()<br>
#-----<br>
# root メインウィンドウの設定<br>
root = tk.Tk()<br>
root.title("Raspberery Pi による N ゲージ運転制御")<br>
root.geometry("500x360")<br>
# メインフレームの作成と設置<br>
frame = ttk.Frame(root)<br>

```

```
frame.grid(pady=20) # pack(fill = tk.BOTH, padx=20,pady=10)<br>
# StringVar のインスタンスを格納する変数 text の設定<br>
text_1 = tk.StringVar(frame)<br>
text_1.set("右回り始動")<br>
text_2= tk.StringVar(frame)<br>
text_2.set("右回り停止")<br>
text_3 = tk.StringVar(frame)<br>
text_3.set("左回り始動")<br>
text_4 = tk.StringVar(frame)<br>
text_4.set("左回り停止")<br>
text_5 = tk.StringVar(frame)<br>
text_5.set("Step_1")<br>
text_6= tk.StringVar(frame)<br>
text_6.set("Step_2")<br>
text_7 = tk.StringVar(frame)<br>
text_7.set("Step_3")<br>
text_8 = tk.StringVar(frame)<br>
text_8.set("Step_4")<br>
text_9 =tk.StringVar(frame)<br>
text_9.set("Step_5")<br>
text_10 =tk.StringVar(frame)<br>
text_10.set("Step_6")<br>
text_11 =tk.StringVar(frame)<br>
text_11.set("Step_7")<br>
text_12 =tk.StringVar(frame)<br>
text_12.set("Step_8")<br>
text_13 = tk.StringVar(frame)
text_13.set("Point1_1")<br>
text_14 = tk.StringVar(frame)
text_14.set("Point1_2")<br>
text_15 = tk.StringVar(frame)<br>
text_15.set("Point2_1")<br>
text_16 = tk.StringVar(frame)<br>
text_16.set("Point2-2")<br>
text_17 = tk.StringVar(frame)<br>
text_17.set("Point3-1")<br>
text_18 = tk.StringVar(frame)<br>
text_18.set("Point3-2")<br>
```

```
text_19 = tk.StringVar(frame)<br>
text_19.set("PATTERN_1")<br>
text_20 = tk.StringVar(frame)<br>
text_20.set("PATTERN_2")<br>
text_21 = tk.StringVar(frame)<br>
text_21.set("PATTERN_3")<br>
text_22 = tk.StringVar(frame)<br>
text_22.set("PATTERN_4")<br>
text_31 = tk.StringVar(frame)<br>
text_31.set("最高速度入力")<br>
text_32 = tk.StringVar(frame)<br>
text_32.set("")<br>
text_e = tk.StringVar(frame)<br>
text_e.set("閉じる")<br><br>
```

各種ウィジェットの作成

```
button_1 = tk.Button(frame, activeforeground='red', textvariable=text_1, command=r_manual)#右回り試  
運転<br>
button_2 = tk.Button(frame, activeforeground='red', textvariable=text_2, command=r_stop_manual)#右  
回り試運転終了<br>
button_3 = tk.Button(frame, activeforeground='red', textvariable=text_3, command=l_manual)#左回り試  
運転<br>
button_4 = tk.Button(frame, activeforeground='red', textvariable=text_4, command=l_stop_manual)#左  
回り試運転<br>
button_5 = tk.Button(frame, textvariable=text_5, fg='blue', activeforeground='red', command=p1) #運行  
プラン 1<br>
button_6 = tk.Button(frame, textvariable=text_6, fg='blue', activeforeground='red', command=p2) #運行  
プラン 2<br>
button_7 = tk.Button(frame, textvariable=text_7, fg='blue', activeforeground='red', command=p3) #運行  
プラン 3<br>
button_8 = tk.Button(frame, textvariable=text_8, fg='blue', activeforeground='red', command=p4) #運行  
プラン 4<br>
button_9 = tk.Button(frame, textvariable=text_9, fg='blue', activeforeground='red', command=p5) #運行  
プラン 5<br>
button_10 = tk.Button(frame, textvariable=text_10, fg='blue', activeforeground='red', command=p6) #運  
行プラン 6<br>
button_11 = tk.Button(frame, textvariable=text_11, fg='blue', activeforeground='red', command=p7) #運  
行プラン 7<br>
```

```

button_12 = tk.Button(frame, textvariable=text_12, fg='blue', activeforeground='red', command=p8) #運行プラン 8<br>
button_13 = tk.Button(frame, activeforeground='red', textvariable=text_13, command=point1_1)#1 番線用ポイント<br>
button_14 = tk.Button(frame, activeforeground='red', textvariable=text_14, command=point1_2)#2 番線用ポイント<br>
button_15 = tk.Button(frame, activeforeground='red', textvariable=text_15, command=point2_1)#本線用ポイント<br>
button_16 = tk.Button(frame, activeforeground='red', textvariable=text_16, command=point2_2)#操作場用ポイント<br>
button_17 = tk.Button(frame, activeforeground='red', textvariable=text_17, command=point3_1)#操作場 A 線用ポイント<br>
button_18 = tk.Button(frame, activeforeground='red', textvariable=text_18, command=point3_2)#操作場 B 線用ポイント<br>
button_19 = tk.Button(frame, textvariable=text_19, fg='blue', activeforeground='red', command=p_all_1)#<br>
button_20 = tk.Button(frame, textvariable=text_20, fg='blue', activeforeground='red', command=p_all_2)#<br>
button_21 = tk.Button(frame, textvariable=text_21, fg='blue', activeforeground='red', command=p9)#<br>
button_22 = tk.Button(frame, textvariable=text_22, fg='blue', activeforeground='red', command=p10)#<br>
button_e = tk.Button(frame, textvariable=text_e, fg='red', command=quit)# root.destroy)#プログラム終了<br>
label_1 = ttk.Label(frame, textvariable=text_31)<br>
entry_1 = ttk.Entry(frame, textvariable=text_32)<br><br>

# 各種ウィジェットの設置<br>
label_1.grid(column=1, row=0, padx=10, pady=10) #, pady=(10, 10))#inputmaxspeed<br>
entry_1.grid(column=2, row=0) # inputmaxspeed<br>
button_1.grid(column=0, row=3, padx=10, pady=10) # r_manual_start<br>
button_2.grid(column=1, row=3, padx=(10, 10)) # l_manual_stop<br>
button_3.grid(column=2, row=3, padx=(10, 10)) # r_manual_starat<br>
button_4.grid(column=3, row=3, padx=(10, 10)) # u_manual_stop<br>
button_5.grid(column=0, row=5, padx=(10, 10)) # plan1<br>
button_6.grid(column=1, row=5, padx=(10, 10)) # plan2<br>
button_7.grid(column=2, row=5, padx=(10, 10)) # plan3<br>
button_8.grid(column=3, row=5, padx=(10, 10)) # plan4<br>

```

```
button_9.grid(column=0, row=7, padx=(10, 10)) # plan5<br>
button_10.grid(column=1, row=7, padx=(10, 10)) # plan6<br>
button_11.grid(column=2, row=7, padx=(10, 10)) # plan7<br>
button_12.grid(column=3, row=7, pady=(10, 10)) # plan8<br>
button_13.grid(column=1, row=9, padx=(10, 10)) # point1_1<br>
button_14.grid(column=2, row=9, padx=(10, 10)) # point1_2<br>
button_15.grid(column=1, row=10, padx=(10, 10)) # point2_1<br>
button_16.grid(column=2, row=10, padx=(10, 10)) # point2_2<br>
button_17.grid(column=1, row=11, padx=(10, 10)) # point3_1<br>
button_18.grid(column=2, row=11, padx=(10, 10)) # point3_2<br>
button_19.grid(column=0, row=2, padx=(10, 10)) # ALL Plan1<br>
button_20.grid(column=1, row=2, padx=(10, 10)) # ALL Plan2<br>
button_21.grid(column=2, row=2, padx=(10, 10)) # ALL Plan3<br>
button_22.grid(column=3, row=2, padx=(10, 10)) # ALL Plan4<br><br><br>
```

```
button_e.grid(column=3, row=13, pady=(10, 10)) # close<br><br>
```

```
root.mainloop()<br>
```