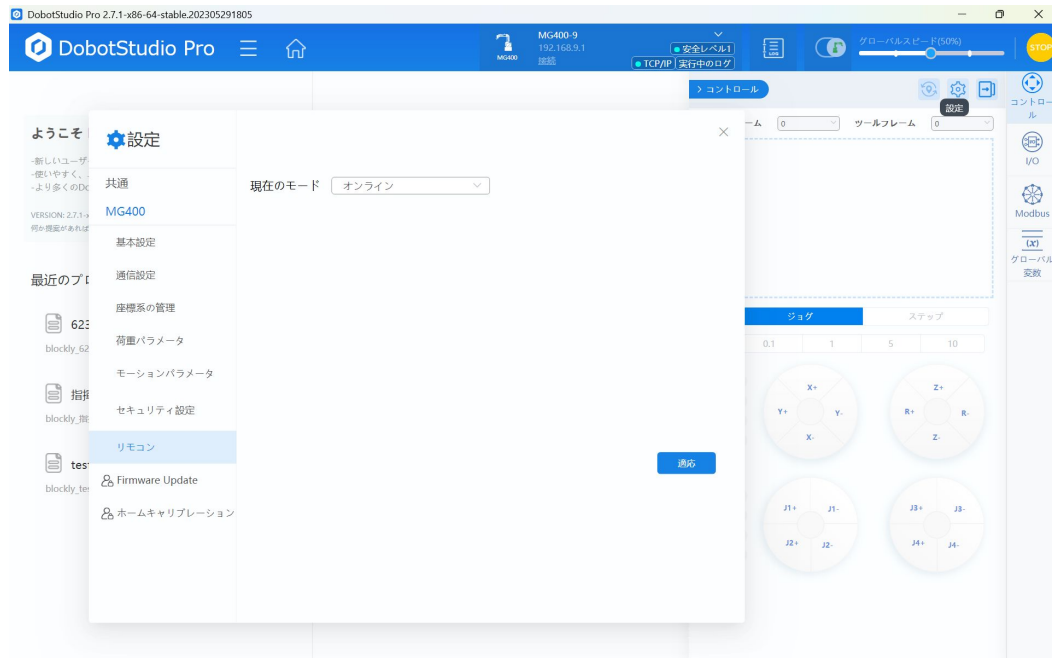


Dobot MG400をpythonで動かす

1. Dobot Studio Proでの操作



Dobot Studio ProでMG400に接続し歯車のボタンから設定を開く(イネーブルスイッチはオフにしたまま)

1. Dobot Studio Proでの操作



リモコンの項目でモードをオンラインからTCP/IP二回開発に変更し適応をクリック。

適応したらDobot Studio Proは閉じる

2. python言語での接続

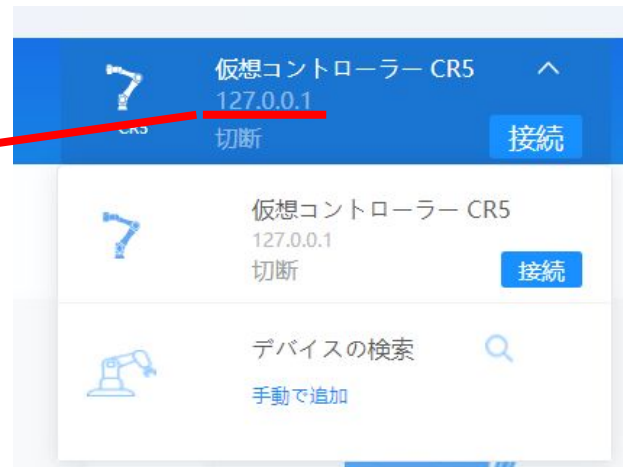
```
# MG400設定
MG400_IP = "192.168.9.1" # MG400のIPアドレス（環境に応じて変更）
SOCKET_TIMEOUT = 10 # Wi-Fi接続用タイムアウト（秒）
```

```
def connect_mg400_wifi(self):
    """MG400にWi-Fi接続"""
    try:
        print("MG400への接続を開始します...")

        # Dashboard接続（29999ポート）
        self.dashboard_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.dashboard_socket.settimeout(SOCKET_TIMEOUT)
        self.dashboard_socket.connect((MG400_IP, 29999))

        # Move接続（30003ポート）
        self.move_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.move_socket.settimeout(SOCKET_TIMEOUT)
        self.move_socket.connect((MG400_IP, 30003))

        print(f"MG400にWi-Fi接続しました")
        return True
    except Exception as e:
        print(f"MG400 Wi-Fi接続エラー: {e}")
        return False
```



自分が使うMG400のIPを書き、
socketというモジュールをインストール
して左のコードを実行

3. MG400をpythonで有効化

pythonでの接続が完了後右のコードでMG400を有効化(Dobot Studio Proで言うイネーブルスイッチをオン)

```
def enable_mg400(self):
    """MG400を有効化"""
    try:
        print("MG400初期化シーケンスを開始します...")

        # ロボット状態確認
        mode_response = self.send_command(self.dashboard_socket, "RobotMode()")
        print(f"RobotMode応答: {mode_response}")

        # 複数回エラークリアを試行
        for i in range(3):
            print(f"エラークリア試行 {i+1}/3...")
            clear_response = self.send_command(self.dashboard_socket, "ClearError()")
            if clear_response and "0" in clear_response:
                print("エラークリア成功")
                break
            time.sleep(1)

        # ロボット無効化+有効化
        print("ロボットを無効化...")
        self.send_command(self.dashboard_socket, "DisableRobot()")
        time.sleep(2)

        print("ロボットを有効化...")
        enable_response = self.send_command(self.dashboard_socket, "EnableRobot()")
        print(f"EnableRobot応答: {enable_response}")

        # 有効化の確認
        for i in range(5):
            time.sleep(2)
            mode_response2 = self.send_command(self.dashboard_socket, "RobotMode()")
            print(f"状態確認 {i+1}/5: {mode_response2}")

            if mode_response2 and "5" in mode_response2:
                print("✓ ロボットが有効化されました (ライトが緑色になっているはずです)")

                # 有効化時の座標を表示
                pos_response = self.send_command(self.dashboard_socket, "GetPose()")
                print(f"有効化時の座標: {pos_response}")

                return True
            elif mode_response2 and "4" in mode_response2:
                print("ロボットは無効状態です。再度有効化を試みます...")
                self.send_command(self.dashboard_socket, "EnableRobot()")

        print("△ ロボット有効化の確認ができませんでした")
        return False

    except Exception as e:
        print(f"MG400有効化エラー: {e}")
        return False
```

4. pythonで動かす

```
# z軸下降
print("z軸下降中...")
self.send_command(self.move_socket, "MovJ(250,0,30,0)")

# 動作完了を待つ間も停止フラグをチェック
for i in range(30): # 待機時間を3秒に延長（遅い速度に対応）
    if self.stop_loop:
        break
    time.sleep(0.1)

if self.stop_loop:
    break

# 下降後の位置確認
pos_response = self.send_command(self.dashboard_socket, "GetPose()")
print(f"z軸下降後の位置: {pos_response}")

# 停止フラグチェック
if self.stop_loop:
    break

# z軸上昇
print("z軸上昇中...")
self.send_command(self.move_socket, "MovJ(250,0,100,0)")
```

あとは動きをプログラミングするだけ