# 如何開始使用:

## 加入檔案

URCoordinates.cs

UrSocketControl.cs

## using namespace

using UrRobot.Socket;

using UrRobot.Coordinates;

a

# 函式介紹

## 伺服器連線

void startServer(string ip, int port)

void stopServer()

## 手臂控制

### void Stop()

停止手臂(須等到目前動作結束)。

### bool getPosition(ref URCoordinates pos)

拿取手臂座標，手臂必須在mode.stop或是mode.pservoj

### void goGripper(int pos, int force = 0, int speed = 0, bool wait = true)

夾爪夾放，pos介於0~255，255為全部夾緊，在UR控制板上可以做夾放校正。Wait如果是false則直接進行下一個動作，不會等夾取完。

### void goPosition(URCoordinates pos)

移動到座標點[pos]，使用mode.pmovep，有時UR會算錯Rx Ry Rz，所以必須要用goPosition2才能解決。

### void goPosition2(URCoordinates pos)

移動到座標點[pos]，使用mode.pmovej，會使用內建逆向運動學[1]

備註:此一函式是利用UR指令的movej，所以移動速度比較快，也解決一些movep會說到不了的問題，但是因為經過內建的逆向運動學[1]，所以會出現問題。

### void goJoint(float j1, float j2, float j3, float j4, float j5, float j6)

# 附錄

1.內建逆向運動學

內建的逆向運動學有時會出錯，最常見的狀況是:希望計算的座標點與目前的座標點太過接近，就會出現error:找不到。