

銘 傳 大 學  
電 腦 與 通 訊 工 程 學 系

專 題 研 究 文 件

組員：

蘇峻緯

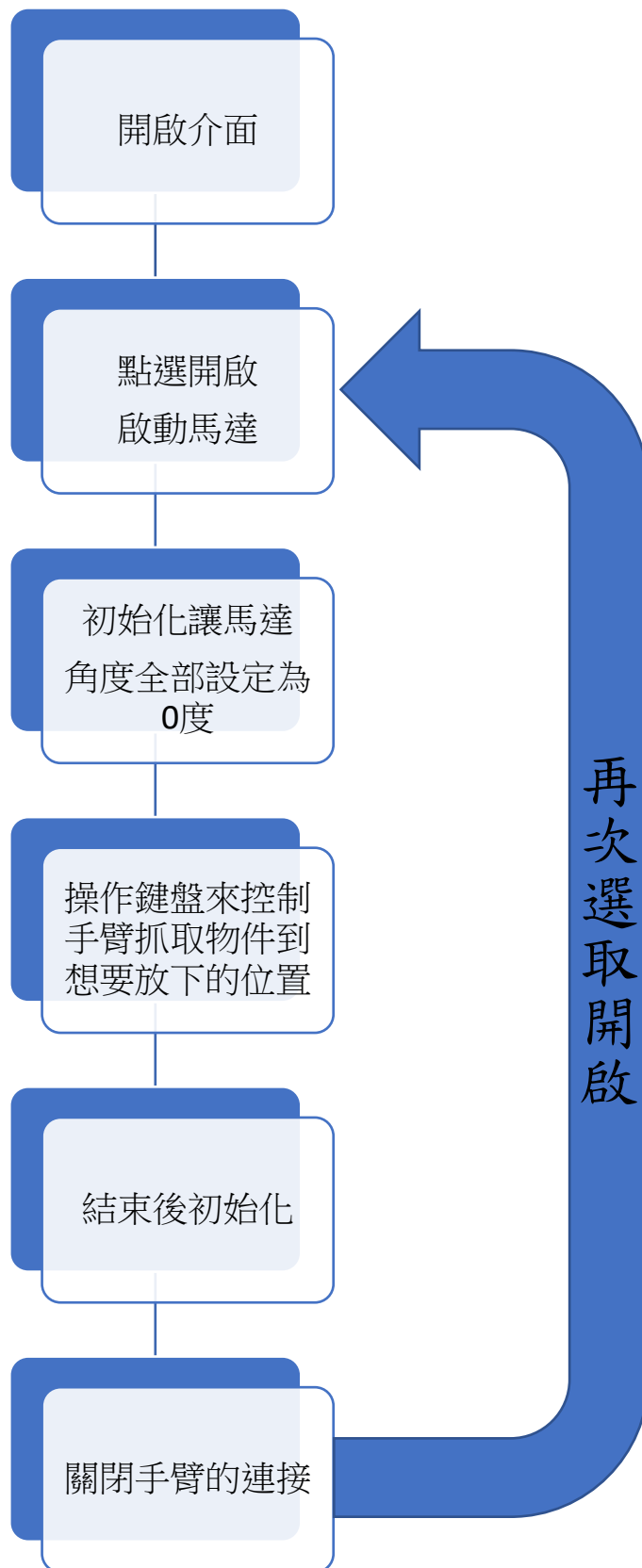
專題研究題目：遠端控制機器手臂

## 動機與目的

以機器取代人力，機器可以做出許多超出人力範圍的限制，例如：能搬運更大更重的物品、細微的過程能更精確等功能，也能夠減少人力的成本。

下面的實驗是以遙控的方式操作手臂拿取東西，到你想要放的位置。

## 動作流程圖含說明



## 程式含註解

```
pip install --pre pyax12 #安裝 pyax12 函式庫
pip install --upgrade pyax12 #更新 pyax12 函式庫
pip install pyautogui #安裝 GUI 介面

import tkinter as tk
from tkinter import *
from pyax12.connection import Connection
import time
#Connect to the serial port
serial_connection = Connection(port="COM4", baudrate=1000000)
#馬達編號
dynamixel_id1 = 1
dynamixel_id2 = 2
dynamixel_id3 = 3
dynamixel_id4 = 4
window = tk.Tk()
var = tk.StringVar()
i1,i2,i3,i4=0,0,0,0 #角度初始設 0
ov=0
#把馬達全部角度設為 0 度
def Initialization():
    print('初始化開始')
    global i1,i2,i3,i4
    i1,i2,i3,i4=0,0,0,0
    serial_connection.goto(dynamixel_id1, i1, speed=100,
degrees=True)
    time.sleep(1)    # Wait 1 second

    serial_connection.goto(dynamixel_id2, i2, speed=100,
degrees=True)
    time.sleep(1)    # Wait 1 second

    serial_connection.goto(dynamixel_id3, i3, speed=100,
degrees=True)
    time.sleep(1)    # Wait 1 second
```

```

    serial_connection.goto(dynamixel_id4, i4, speed=100,
degrees=True)

    time.sleep(1)    # Wait 1 second

    print('初始化結束')

    result_label.config(text='初始化結束')

    m1_label.config(text='馬達 ID1 的角度：{}度'.format(i1))
    m2_label.config(text='馬達 ID2 的角度：{}度'.format(i2))
    m3_label.config(text='馬達 ID3 的角度：{}度'.format(i3))
    m4_label.config(text='馬達 ID4 的角度：{}度'.format(i4))

#鍵盤移動馬達
def key(event):

    global i1,i2,i3,i4

    print ("pressed", event.keysym)

    #馬達 1

    if(event.keysym=='Left'):

        if(i1!=150):

            i1=int(i1+10)

            serial_connection.goto(dynamixel_id1, i1, speed=100,
degrees=True)

            print('馬達 ID1 的角度：{}度'.format(i1))

            m1_label.config(text='馬達 ID1 的角度：{}度'.format(i1))

            time.sleep(0.1)

        else:

            print('馬達 ID1 的角度已到最大臨界值')

            m1_label.config(text='馬達 ID1 的角度：{}度,已到最大臨界值'.format(i1))

    elif(event.keysym=='Right'):

        if(i1!=-150):

            i1=int(i1-10)

            serial_connection.goto(dynamixel_id1, i1, speed=100,
degrees=True)

            print('馬達 ID1 的角度：{}度'.format(i1))

            m1_label.config(text='馬達 ID1 的角度：{}度'.format(i1))

            time.sleep(0.1)

        else:

            print('馬達 ID1 的角度已到最小臨界值')

            m1_label.config(text='馬達 ID1 的角度：{}度,已到最小臨界值'.format(i1))

```

```

#馬達 2
if(event.keysym=='Up'):
    if(i2!=60):
        i2=int(i2+10)
        serial_connection.goto(dynamixel_id2, i2, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID2 的角度：{}度'.format(i2))
        m2_label.config(text='馬達 ID1 的角度：{}度'.format(i2))
    else:
        print('馬達 ID2 的角度已到最大臨界值')
        m2_label.config(text='馬達 ID2 的角度：{}度,已到最大臨界值'.format(i2))
elif(event.keysym=='Down'):
    if(i2!=-90):
        i2=int(i2-10)
        serial_connection.goto(dynamixel_id2, i2, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID2 的角度：{}度'.format(i2))
        m2_label.config(text='馬達 ID2 的角度：{}度'.format(i2))
    else:
        print('馬達 ID2 的角度已到最小臨界值')
        m2_label.config(text='馬達 ID2 的角度：{}度,已到最小臨界值'.format(i2))

#馬達 3
if(event.keysym=='w'):
    if(i3!=30):
        i3=int(i3+10)
        serial_connection.goto(dynamixel_id3, i3, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID3 的角度：{}度'.format(i3))
        m3_label.config(text='馬達 ID3 的角度：{}度'.format(i3))
    else:
        print('馬達 ID3 的角度已到最大臨界值')
        m3_label.config(text='馬達 ID3 的角度：{}度,已到最大臨界值'.format(i3))

```

```

elif(event.keysym=='s'):
    if(i3!=-90):
        i3=int(i3-10)
        serial_connection.goto(dynamixel_id3, i3, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID3 的角度：{}度'.format(i3))
        m3_label.config(text='馬達 ID3 的角度：{}度'.format(i3))
    else:
        print('馬達 ID3 的角度已到最小臨界值')
        m3_label.config(text='馬達 ID3 的角度：{}度,已到最小臨界值'.format(i3))
#馬達 4
if(event.keysym=='a'):
    if(i4!=30):
        i4=int(i4+10)
        serial_connection.goto(dynamixel_id4, i4, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID4 的角度：{}度'.format(i4))
        m4_label.config(text='馬達 ID4 的角度：{}度'.format(i4))
    else:
        print('馬達 ID4 的角度已到最大臨界值')
        m4_label.config(text='馬達 ID4 的角度：{}度,已到最大臨界值'.format(i4))
elif(event.keysym=='d'):
    if(i4!=-30):
        i4=int(i4-10)
        serial_connection.goto(dynamixel_id4, i4, speed=100,
degrees=True)
        time.sleep(0.1)
        print('馬達 ID4 的角度：{}度'.format(i4))
        m4_label.config(text='馬達 ID4 的角度：{}度'.format(i4))
    else:
        print('馬達 ID4 的角度已到最小臨界值')
        m4_label.config(text='馬達 ID4 的角度：{}度,已到最小臨界值'.format(i4))
#勾選馬達狀態

```

```

def tf():
    r1.focus_set()
    global
ov,serial_connection,dynamixel_id2,dynamixel_id3,dynamixel_id4
    print(var.get())
    result_label.config(text='馬達狀態:{}'.format(var.get()))
    if(var.get()=='開啟'):
        r1.bind("<Key>", key)
        if(ov==1):
            serial_connection = Connection(port="COM4",
baudrate=1000000)
            dynamixel_id1 = 1
            dynamixel_id2 = 2
            dynamixel_id3 = 3
            dynamixel_id4 = 4
            ov=0
        elif(var.get()=='關閉'):
            frame.unbind("<Key>")
            ov=1
            serial_connection.close()#關閉手臂
#介面
frame = Frame(window)
frame_l = Frame(frame)
frame_r = Frame(frame)
frame_e = Frame(window , bg='#C0C0C0')
frame.pack()
frame_l.pack(side=LEFT)
frame_r.pack(side=RIGHT)
frame_e.pack(side=BOTTOM)

m1_label = tk.Label(frame_r, text='馬達 ID1 的角度:',width=30)
m2_label = tk.Label(frame_r, text='馬達 ID2 的角度:',width=30)
m3_label = tk.Label(frame_r, text='馬達 ID3 的角度:',width=30)
m4_label = tk.Label(frame_r, text='馬達 ID4 的角度:',width=30)
button1 = tk.Button(window, text='初始化', fg='red',
command=Initialization)
r1 = tk.Radiobutton(frame_l, text='開啟', variable=var, value='開啟',command=tf)

```



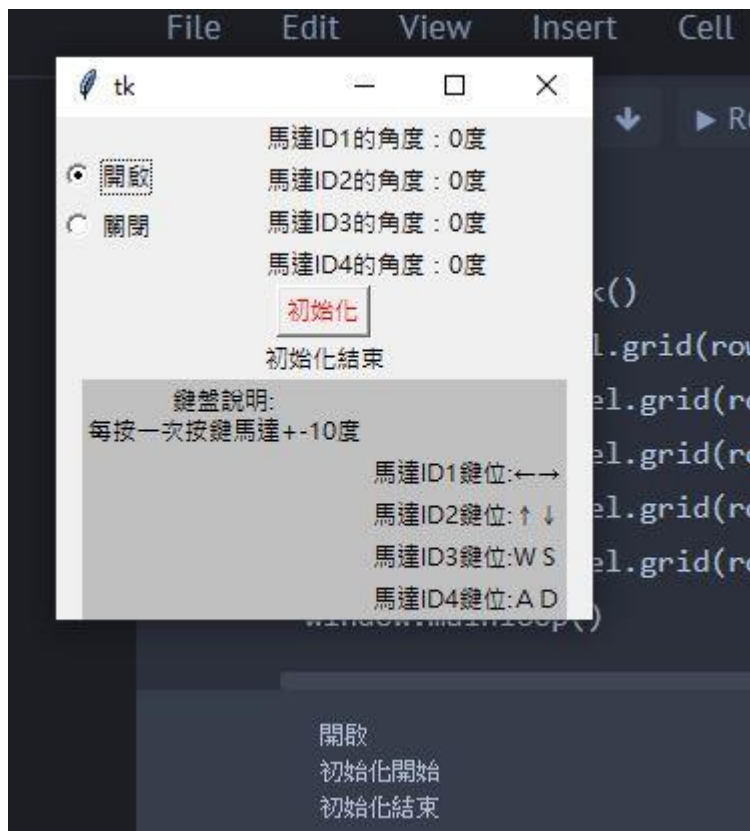
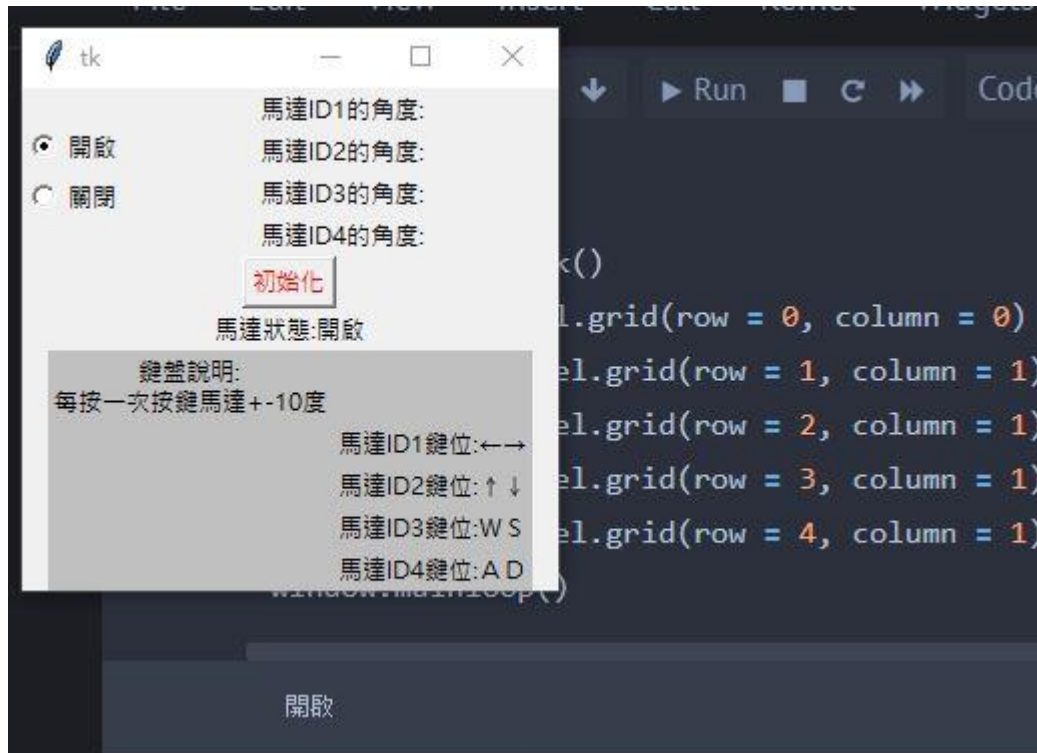
```

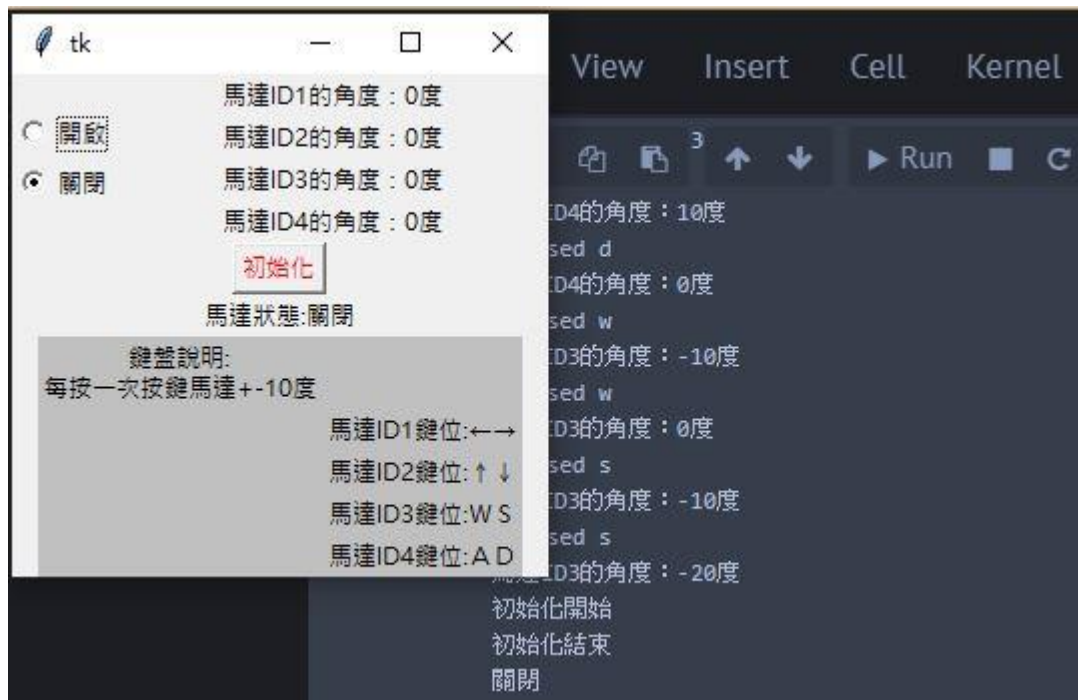
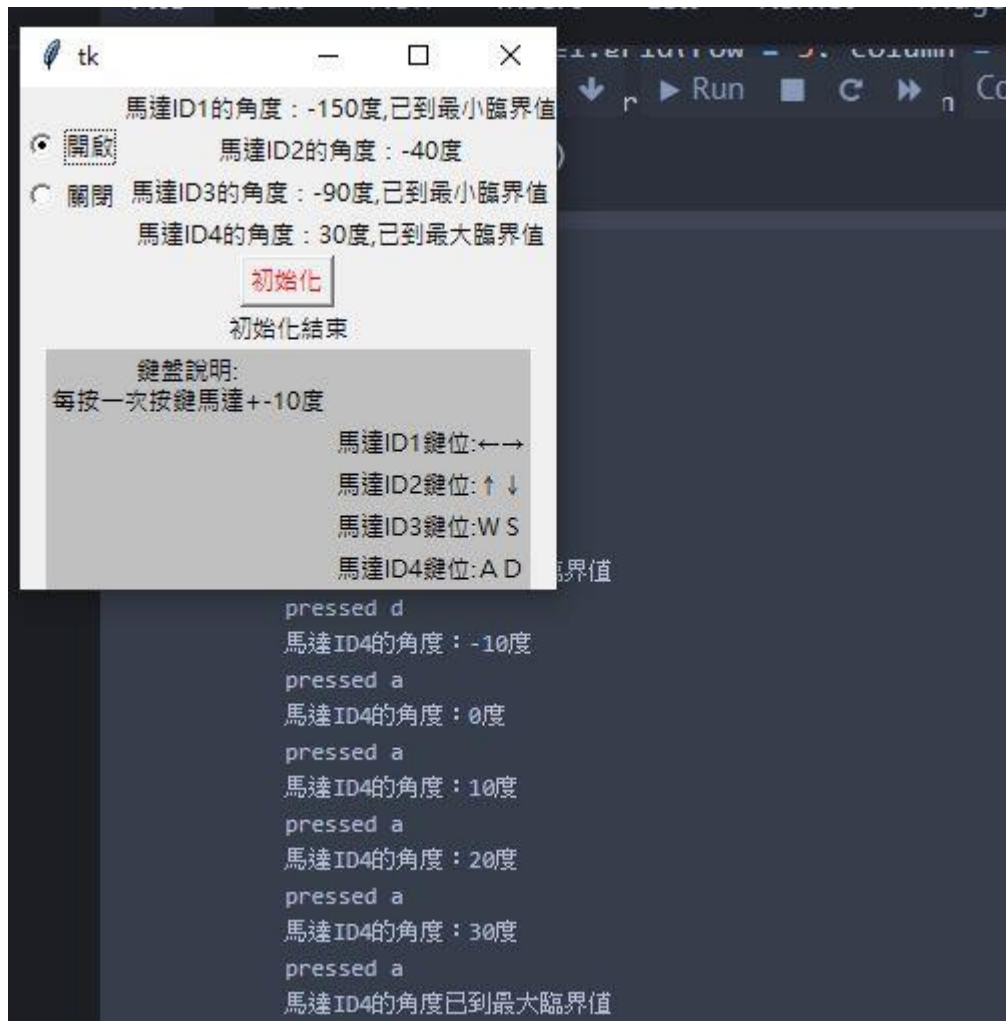
r2 = tk.Radiobutton(frame_1, text='關閉', variable=var, value='關閉',command=tf)
result_label = tk.Label(window, text='馬達狀態:未啟動')
Explanation_label = tk.Label(frame_e, text='{}'.format('鍵盤說明:\n每按一次按鍵馬達+-10 度'), bg='#C0C0C0')
Explanation1_label = tk.Label(frame_e, text='{}'.format('馬達 ID1 鍵位:←'), bg='#C0C0C0')
Explanation2_label = tk.Label(frame_e, text='{}'.format('馬達 ID2 鍵位:↑↓'), bg='#C0C0C0')
Explanation3_label = tk.Label(frame_e, text='{}'.format('馬達 ID3 鍵位:WS'), bg='#C0C0C0')
Explanation4_label = tk.Label(frame_e, text='{}'.format('馬達 ID4 鍵位:AD'), bg='#C0C0C0')
m1_label.pack()
m2_label.pack()
m3_label.pack()
m4_label.pack()
button1.pack()
r1.pack()
r2.pack()
result_label.pack()
Explanation_label.grid(row = 0, column = 0)
Explanation1_label.grid(row = 1, column = 1)
Explanation2_label.grid(row = 2, column = 1)
Explanation3_label.grid(row = 3, column = 1)
Explanation4_label.grid(row = 4, column = 1)
window.mainloop()

```

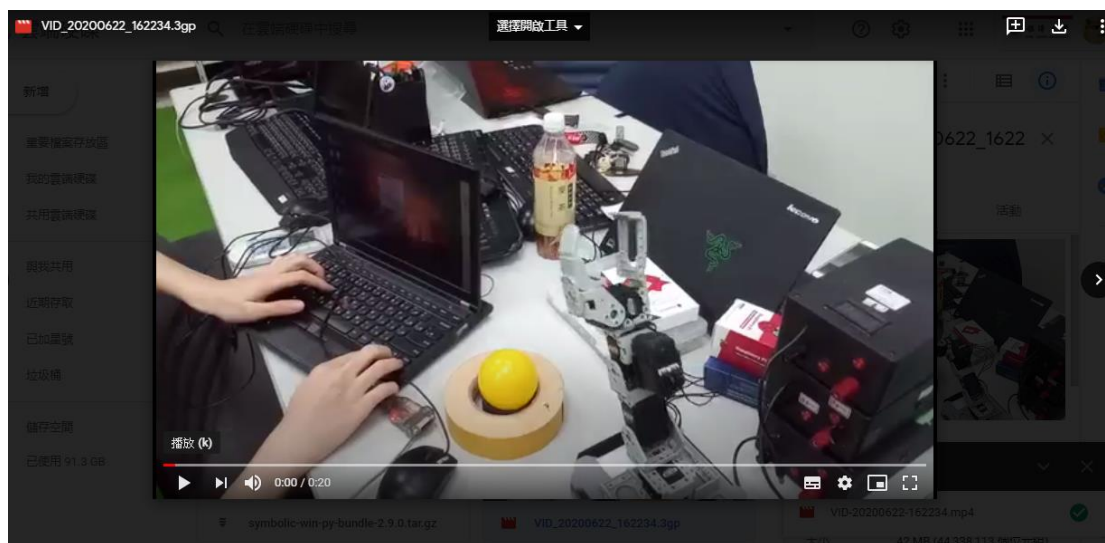
## 實驗結果(可使用連環圖呈現)

介面操作▼





## 實際操作影片



## 結論

搬運物品，並不是每個物品的重量都是人可以單獨承受的起，使用機械手臂搬取物品，不需要消耗大量的人力成本，只需要讓操作人員熟悉操作介面就可以少去掉一堆人在搬一件很重的物品，在這門課程學會了如何控制手臂以及結合來介面操手臂對於未來不論升學還是求職上對於技術都很有幫助。