STEM_9

自　學　教　材

編著：王俊文 (Aman Wong C.M.)

專案一：組裝 MeArm

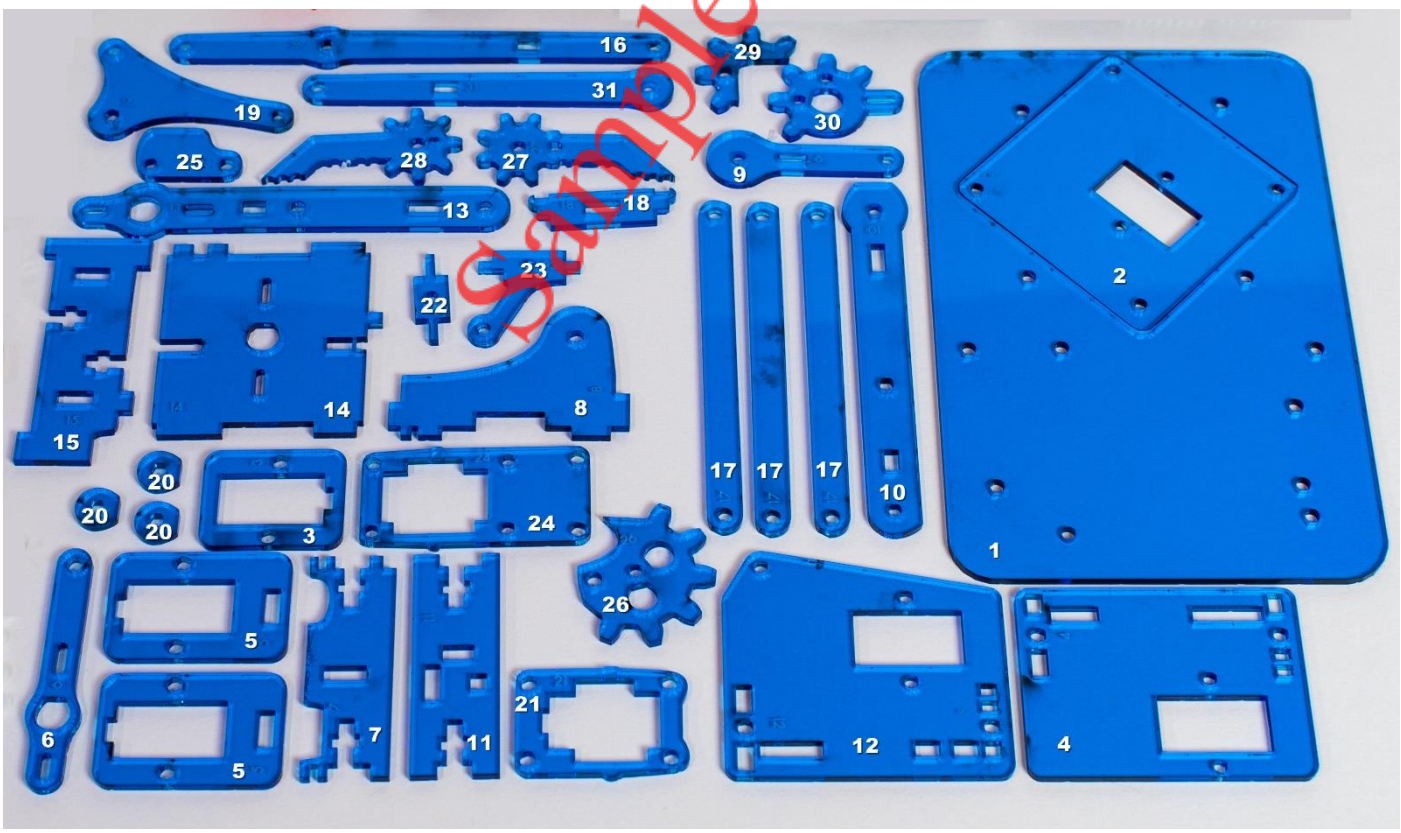
目錄

(本專案所含圖片只供參考)

1. 組件清單與硬件介紹

- Arduino Uno R3 兼容主板
- MeArm 搖桿擴展板
- SG90 伺服電動機連配件
- USB 線
- M3 螺母 × 10
- M3 4mm 螺絲（平頭）× 8
- M3 6mm 螺絲 × 6
- M3 8mm 螺絲 × 15
- M3 10mm 螺絲 × 3
- M3 12mm 螺絲 × 8
- M3 30mm 螺絲（平頭）× 4
- 橡膠腳墊 × 4
- 六角銅柱 × 4
- MeArm 鐳射切割 3mm 亞加力膠片組件（共 36 件*，每件均有編號 1-31，見下圖）

  *套件提供 37 件組件，但編號 20 組件有一件是多餘的，所提供的 4 件中只需 3 件就足夠。



(本套件另提供一把 2.5mm 六角匙和一把十字螺絲批作組裝過程工具。)

Arduino Uno R3 兼容主版

Arduino Uno R3 是 Arduino Uno 的第三代改進版，是一款易用型開放源碼微控制器開發板。其運作原理主要是運用按鈕、感應器、手機等訊息輸入 Uno 板，透過執行所燒載的程式而作出反應，輸出的零部件可以是直流電動機、伺服電動機、LED 等。

MeArm 搖桿擴展板

MeArm 搖桿擴展板是為 MeArm 機械臂設計的一款 Arduino Uno 擴展板，可以直接疊插到 Arduino Uno 板上，提供 4 個伺服電動機接口、一個藍牙模組的接口和一個內置的紅外線接收器。搖桿擴展板上的兩個搖桿根據 X/Y 位置輸出四個二軸類比訊號，用來控制 MeArm 四個伺服電動機轉動，從而操縱 MeArm 的上下移動、左右轉動、前後移動和夾爪開關。

SG90 伺服電動機

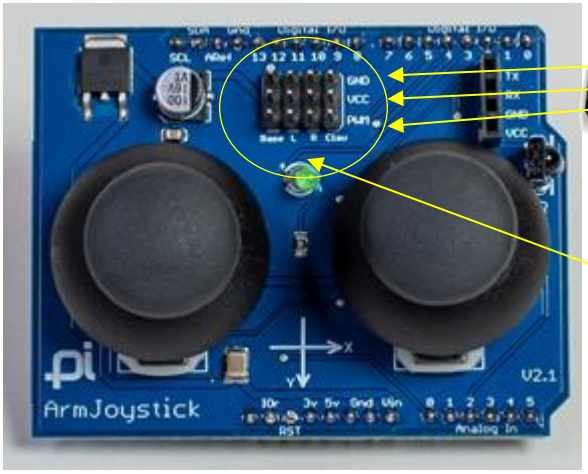
伺服電動機是一種旋轉驅動器，提供準確的角度位置控制。SG90 伺服電動機是一個可提供高輸出功率、可安裝在窄小位置的小型輕巧 (9 克) 的伺服電動機，能大約旋轉 180 度 (左右各 90 度)。它包含一個反饋式電動機控制器和齒輪箱。每個伺服電動機都備有伺服臂和螺絲配件。

2. Arduino 板連接與程式上載

在組裝 MeArm 主機體之前，先把各硬件與 Arduino 板連接，並從電腦上載程式到 Arduino 板，步驟如下：

1. 把 MeArm 搖桿擴展板疊插在 Arduino 板上。

2. 把四個伺服電動機接駁到擴展板上如下圖：



每條連接線都分三種顏色（褐色、紅色、橙色），分別針入每個接口的 GND、VCC 和 PWM 針腳。

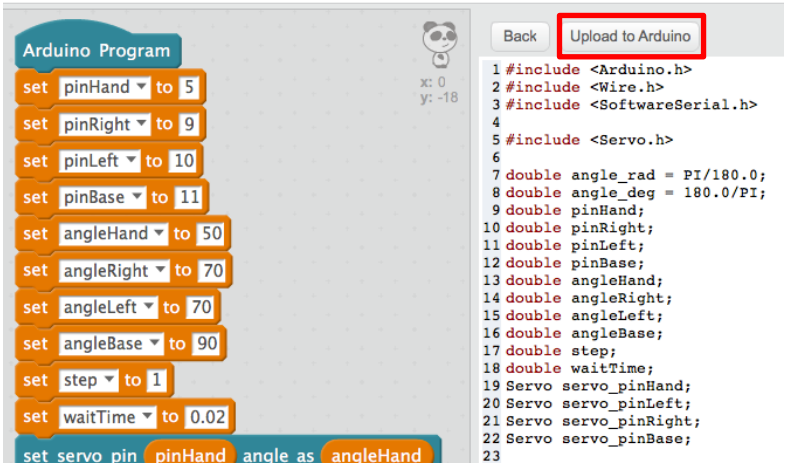
根據四個接口的標記分別插入對應的伺服電動機連接線：Base（BASE 伺服電動機）、L（LEFT 伺服電動機）、R（RIGHT 伺服電動機）和 Claw（CLAW servo）。

3. 用 USB 線把 Arduino 板與電腦連接起來。

4. 在電腦啟動 mBlock 3 程式（最新的 3.4.12 版本，安裝程式可以從 https://dl.makeblock.com/mblock3/mBlock_win_V3.4.12.exe 下載或從本套件光碟中獲取），在程式界面上方的**連接（Connect）**選單中點選**序列埠（Serial Port）**，再點選已連接了 Arduino 板的 COM 埠。

5. 在程式中開啟伺服電動機校正程式（本套件光碟已提供，檔名：meArm-Project1_CAL.sb2），在**編輯（Edit）**選單中點選 **Arduino 模式（Arduino mode）**，然後在右方出現的 Arduino 程式語言區的上方按**上傳到 Arduino（Upload to Arduino）**鈕上載程式到 Arduino 板。

6. 上載程序完畢時，可聽見機械的聲音，代表所有伺服電動機的角度位置已重設。

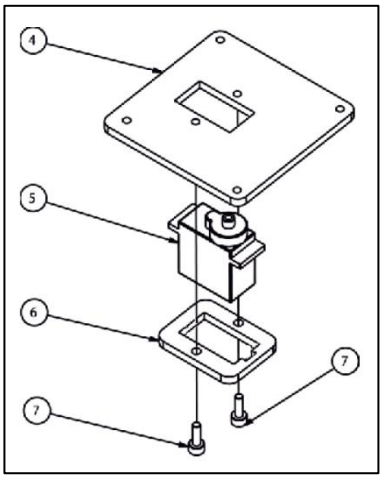


3. 組裝 MeArm 主體

以下是組裝 MeArm 主體（由 36 件亞加力組件、4 個伺服電動機及配件、M3 螺母和不同長度的螺絲等構成）的步驟：

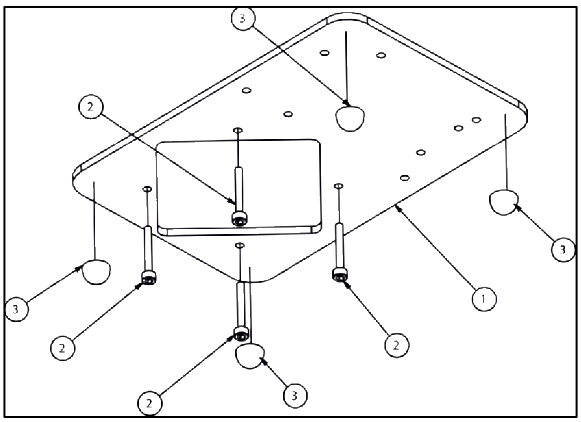
步驟 1：組裝底座

編號	組件名稱（亞加力組件編號）	數量
1.	底板 (1)	1
2.	M3 30mm 螺絲	4
3.	橡膠腳墊	4
4.	伺服電動機轉盤 (2)	1
5.	伺服電動機	1
6.	底座伺服電動機套環 (3)	1
7.	M3 8mm 螺絲	2
8.	M3 螺母	4



步驟 1A：組裝組件 4 至 7

將伺服電動機(5，MeArm 主體中作為 BASE[底座]伺服電動機)底部插入底座伺服電動機套環(6)使套環觸及電動機的凸緣，然後把電動機的頂部插入正方形伺服電動機轉盤(4)的方孔內。把兩顆 8mm 螺絲(7)由電動機套環(6)下方插入並鎖入電動機轉盤(4)，把組裝的組件(4-6)扣緊。



步驟 1B：組裝組件 1 至 3

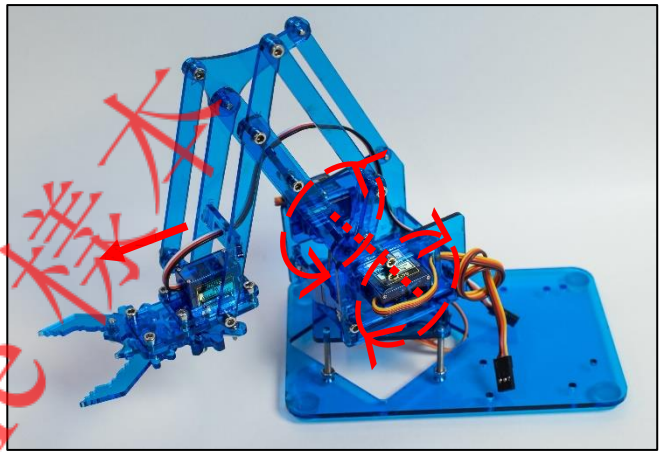
將四枚橡膠腳墊(3)黏貼於底板(1)下方的四個角，然後把四顆 30mm 螺絲(2)從底板下方插入四個小孔，使它們與即將連結底板(1)的電動機轉盤(4)上的四個小孔對齊。

專案 1 作業建議答案

1. 根據 4C，我們知道當平行連桿向前或向後移動時，夾爪組合會同時向上或向下移動。是否有可能令平行連桿前後移動時不改變夾爪組合的垂直位置呢？
2. 根據 4D，我們知道當夾爪組合向上或向下移動時，雙臂基底會同時向前或向後移動。是否有可能令夾爪組合向上或向下移動時不改變雙臂基底的水平位置呢？

問題 1：

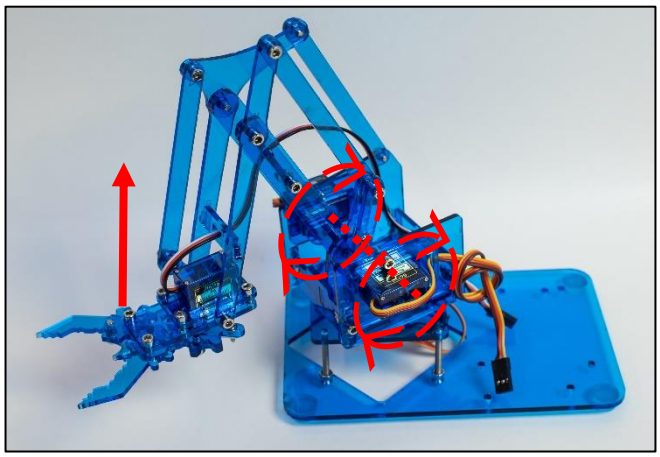
當 LEFT 和 RIGHT 伺服電動機同速地逆時針旋轉時，平行連桿(連同夾爪組合)會向前移動，夾爪組合的垂直位置卻不會改變。



相反，當這兩個伺服電動機同速地順時針旋轉時，平行連桿會向後移動，同樣地夾爪組合的垂直位置也不會改變。

問題 2：

當 LEFT 伺服電動機和 RIGHT 伺服電動機分別逆時針和順時針旋轉時，夾爪組合會向上移動，其水平位置卻不會改變（即不會前後移動）。



相反，當 LEFT 伺服電動機和 RIGHT 伺服電動機分別順時針和逆時針旋轉時，夾爪組合會向下移動，同樣地其水平位置也不會改變。

STEM_9

自　學　教　材

編著：王俊文 (Aman Wong C.M.)

專案二：MeArm 自動操作編程

目錄

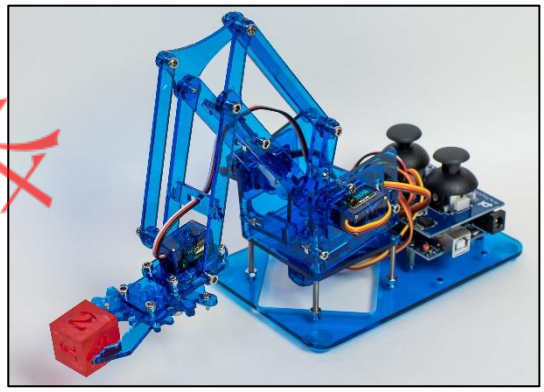
(本專案所含圖片只供參考)

1. 機械臂編程簡介

工廠在安裝一台機械臂時，目的是使某個程序得以自動化，那可能是一個過往用人手處理的程序，或是一個全新的程序。操作一台機械臂涉及兩個重要的元素，就是控制機械臂移動的微控制器 (在本套件中為 Arduino UNO)，以及製作指示微控制器如何進行控制的程式的軟件(在本套件中為 mBlock 3)。

為一個機械人執行某個任務編程之前，我們需要搞清楚機械人與任務之間的物理或幾何關係。為着這個目的，我們必須首先手動操控機械人去確立它執行該任務時涉及之範圍內的坐標位置。

2. MeArm 的 Scratch 編程



以下我們將會用 mBlock 3 編寫一個簡單的 Scratch 程式指示 MeArm 針對一個邊長為 2cm 立方積木 (或一個利用 3D 打印技術製作的 DIY 塑膠積木，見套件提供的"Dice for MeArm.stl" 檔案) 執行以下的任務：在同一個位置把同一粒積木不斷重複地撿起和放下。(示範影片：“Project2_Demo.mp4”)

A. 主程式：數值設定

在**機器人模組**類別中，把「Arduino 主程式」積木拖曳到腳本區。

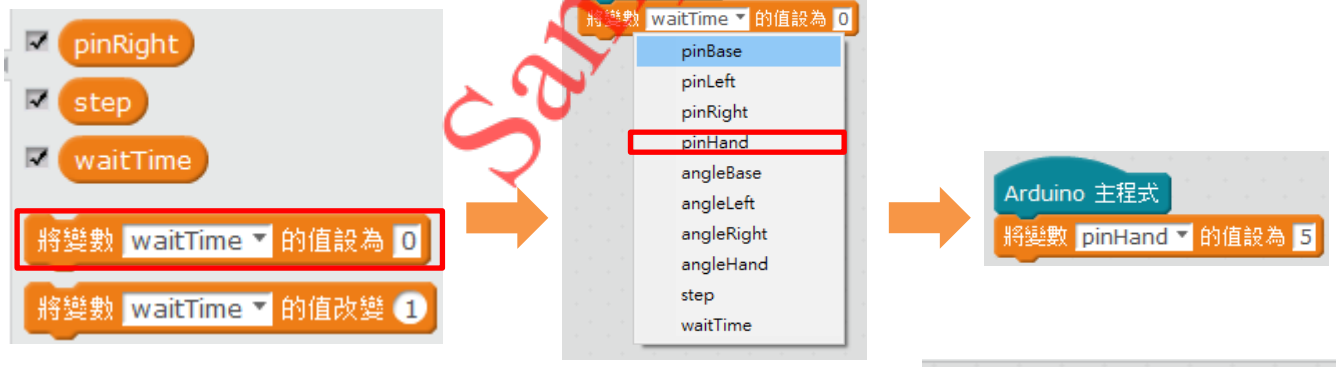
點按**資料和指令➜做一個變數**。在出現的「新變數」對話方塊中，輸入「pinHand」為變數名稱，然後按確認，用來儲存 CLAW 伺服電動機的數位埠數字。

重複以上的步驟去建立下表列出的其他變數：

變數名稱	用途
angleBase	儲存 BASE 伺服電動機的角度值
angleHand	儲存 CLAW 伺服電動機的角度值
angleLeft	儲存 LEFT 伺服電動機的角度值
angleRight	儲存 RIGHT 伺服電動機的角度值
pinBase	儲存 BASE 伺服電動機的數位埠數字
pinLeft	儲存 LEFT 伺服電動機的數位埠數字
pinRight	儲存 RIGHT 伺服電動機的數位埠數字
step	儲存逐級的角度調整值
waitTime	儲存逐級角度調整的時間長度

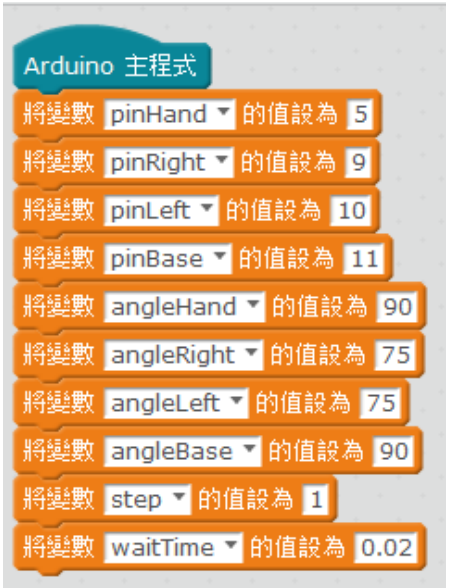
合共有十個新變數積木會出現如右。

在新的變數積木之下，把「將變數( )的值設( )」積木拖曳到腳本區，並嵌在「Arduino 主程式」積木下。點按「waitTime」旁的向下箭咀以開啟一個下拉式選單，當中包含全部變數的名稱。選「pinHand」並輸入數值「5」取代「0」，藉此創造了「將變量(pinhand)的值設為(5)」積本。



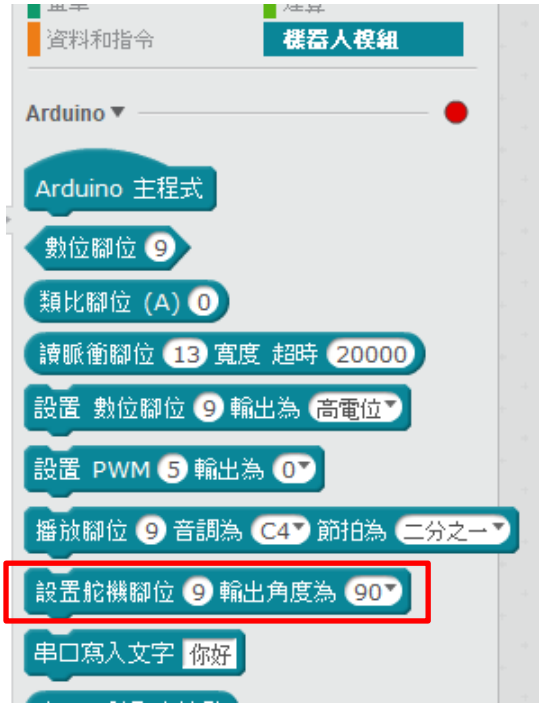
利用「將變數( )的值設為( )」積木重複以上的步驟為其他變數賦值如下，然後把他們一同嵌在「Arduino 主程式」之下。

- pinBase: 11
- pinLeft: 9
- pinRight: 10
- angleBase: 90
- angleLeft: 75
- angleRight: 75
- angleHand: 90
- step: 1
- waitTime: 0.02



現在為每個伺服電動機賦與一個初始角度值。

把一個「設置舵機腳位(9)輸出角度為(90)」積木從**機器人模組**類別拖曳出來，嵌在主程式（「Arduino 主程式」為首的模組）的最下方。



接着，把「pinHand」和「angleHand」變數積木從**資料和指令**類別拖曳出來，分別嵌入第一和第二個參數洞，藉此把「angleHand」變數儲存的初始值賦與 CLAW 伺服電動機(伺服腳位數字 :「pinHand」)。

Project 2 Assignment Answers

1. Basic task: Write a Scratch program to control the MeArm to pick up three vertically stacked blocks of the same colour one by one and then to stack them again up at another location.

2. Extended task: Write a Scratch program to control the MeArm to pick up three vertically stacked blocks in the different colours one by one and then to stack them again up in the original sequence at another location.

Question 1:

Apart from the coding in the paragraph for program testing, all coding remains unchanged. The lower part of the main program is modified as follows:

MeArm needs to grab the cubic blocks located at different heights. The suggested values for grabbing the upper block are U = 115 & F = 115, the middle one U = 100 & F = 130, and the lowest one are U = 90 & F = 140. Actually, you can adjust all these values according to your size and position of the blocks. Besides, you can also change the value of “BaseAngle” for pick-up and put-down location of the blocks.

Question 2:

Apart from the coding in the paragraph for program testing, all coding remains unchanged. The lower part of the main program is modified as follows:

STEM_9

Self-Learning Guide

Author：Aman Wong C.M. (王俊文)

Project 3：Programming for MeArm Manual Operations

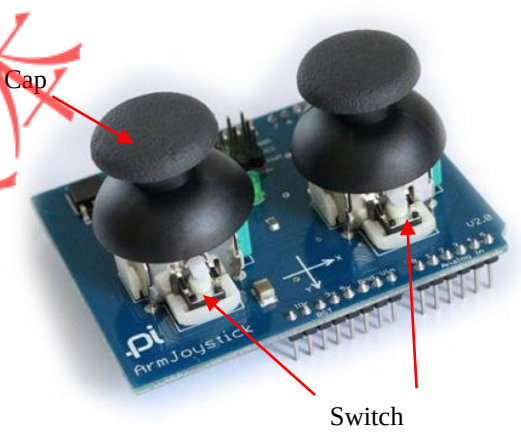
Contents

(Pictures shown in this document are for reference only.)

1. MeArm Joystick and MeArm Manual Operations

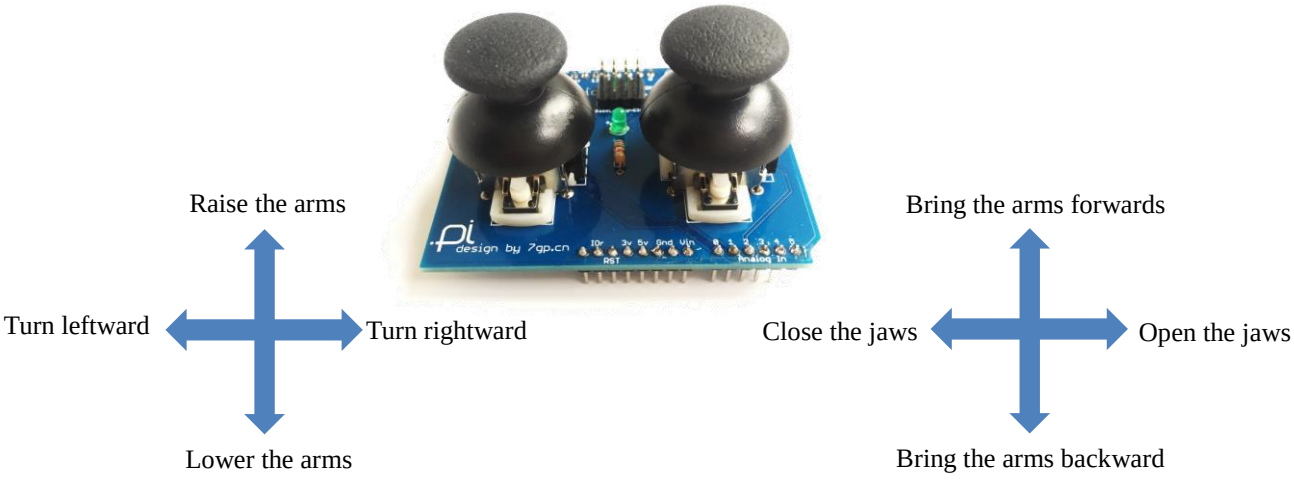
Very often robotic arms can help us handle difficult or dangerous tasks such as conducting experiments with biochemical hazards, dismantling unknown explosives and holding up heavy objects. In this project, we will learn to use the two joysticks on the MeArm Joystick Shield to control MeArm and to do coding for that purpose.

Each joystick on the shield outputs two-axis potentiometer analog signals according to the X/Y position. Two joysticks together serve as motion controllers for the four MeArm servo motors by manipulating MeArm's upward/downward, leftward/rightward and forward/backward movements as well as the open/close actions of the jaws. Each joystick is also a push button that contains a switch for sending Boolean on/off signals by pressing the cap.



Below we are going to create a Scratch program to read MeArm Joystick Shield data from the potentiometers during manual operations of MeArm through the two joysticks on the shield. Based on these data, the program will instruct the four servos of the robotic arm to turn to certain angles according to the following predefined relationships between the X/Y axial locations of the two joysticks and MeArm movements and:

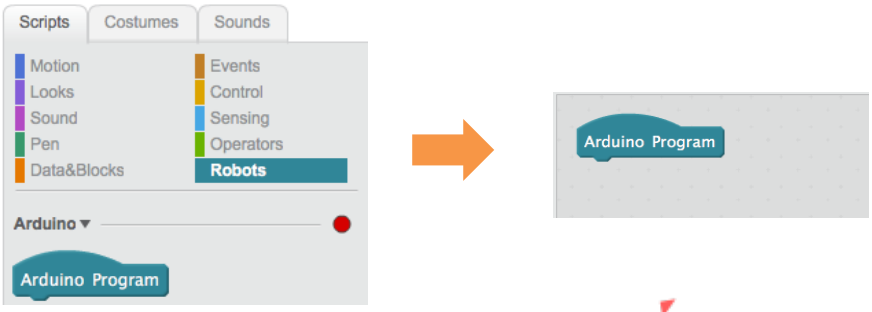
Relationship Between Joysticks' X/Y Axial Locations and MeArm Movements



2. Scratch Programming for MeArm Manual Operations

A. Main Program: Values Setup

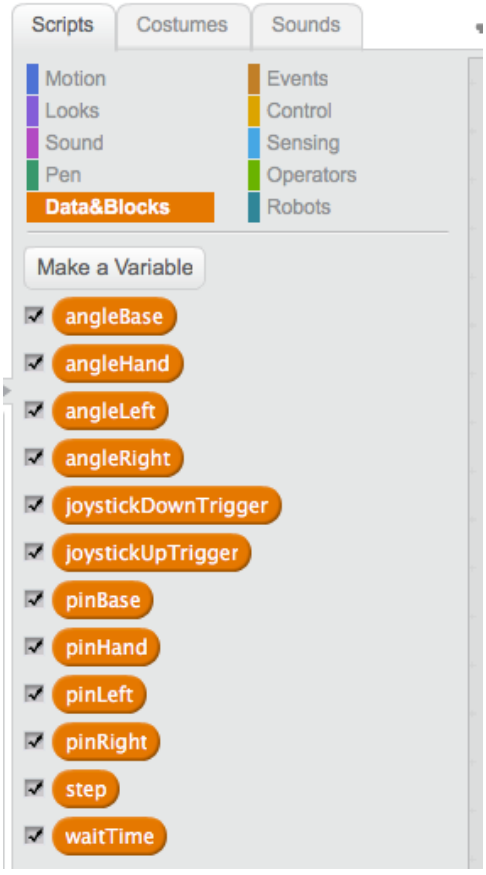
In **Robots** category, drag an “Ardunio Program” block to the script zone.



Click **Data&Blocks** ➜ **Make a Variable**. In the “New Variable” dialog box that pops up, input “pinHand” for the variable name and click **OK** to create the variable “pinHand” to store digital port number of the CLAW servo.

Repeat the above step to create other variables as listed in the following table:

Variable name	Purpose
angleBase	To store the angle value for BASE servo
angleHand	To store the angle value for CLAW servo
angleLeft	To store the angle value for LEFT servo
angleRight	To store the angle value for RIGHT servo
pinBase	To store the digital port number for BASE servo
pinLeft	To store the digital port number for LEFT servo
pinRight	To store the digital port number for RIGHT servo
step	To store the value for angular step adjustment
waitTime	To store the value for step adjustment duration
joystickUpTrigger	To store an up value for triggering movement
joystickDownTrigger	To store an down value for triggering movement

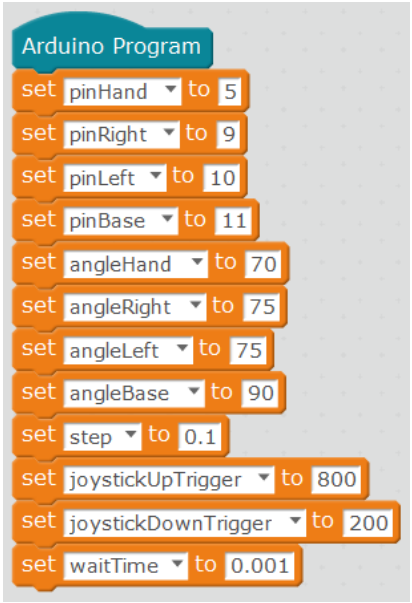


A total of twelve new variable blocks will appear as shown on the right.

Under the list of new variable blocks, drag the “set (waitTime) to (0)” block to the script zone and snap it below to the “Arduino Program” block. Click the down arrow next to “waitTime” to open a dropdown menu containing the full list of variable names. Select “pinHand” and enter the value “5” to replace “0” to create the “set (pinHand) to (5)” block.

Repeat the above step for value assignment of other variables as follows using the “set ( ) to ( )” block before snapping them together under “Arduino Program”.

- pinLeft: 10
- pinRight: 9
- pinBase: 11
- angleHand: 70
- angleLeft: 75
- angleRight: 75
- angleBase: 90
- step: 0.1
- joystickUpTrigger: 800
- joystickDownTrigger: 200
- waitTime: 0.001



Now an initial angle value will be assigned to each servo. Drag a “set servo pin(9) as (90)” block from the **Robots** category to snap it to the bottom of the main program (with the “Arduino Program” hat).