中國文化大學教師教學創新暨教材研發獎勵成果報告書

壹、 計畫名稱

物理－資訊跨學科場域學習與教學之教材研發

貳、 實施課程、授課教師姓名

基礎科學跨學科教學之設計與實踐(五)、黃信健

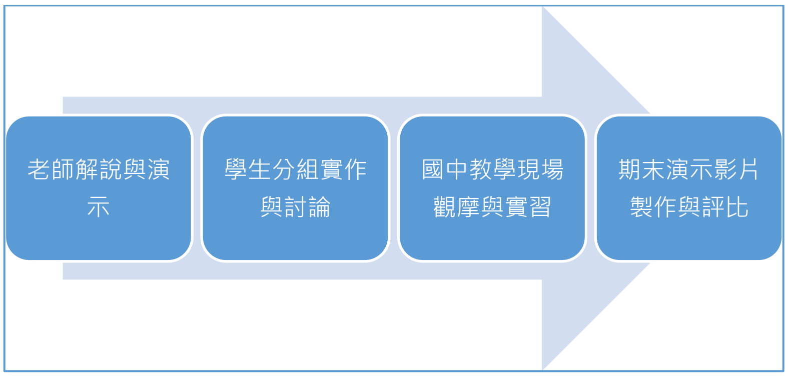
參、 前言

本計畫的精神是延續我們歷年來致力於協助提振國高中物理教育的成果，進一步帶領物理及其他合作系所的同學們進入場域，以提升其敘事能力與專業能力，並培養利他精神。我們的場域包含大學校園及都會與偏鄉的國高中，在大學校園方面，我們希望讓跨院系的學生接觸超酷物理遊戲、促成校園科普推展。而對都會國高中而言，有關自然與生活科技的跨領域教學對國高中老師實是沉重的負擔，在此方面，我們能夠提供以物理為基礎的跨領域支援；至於偏鄉則更需要各種教育資源之挹注。此外，面對日益嚴重的少子化浪潮，大學端莫不鼓勵各系所與高中建立密切的合作關係，而高中端也更加重視如何善用大學資源，本計畫恰可扮演支援與協助相關系所的角色，共同推動與高中之跨學科實質交流，並在開發跨領域議題、擴大計畫影響力等方面，獲得很好的回饋。事實上，我們的計畫已推動多年，但以往課程內容大多仍以物理為主體，直到近年來我們逐步建構了跨物理、資訊傳播、地理及化學領域的教師社群，足以利用本校系所完整多樣的優勢，開始著力於跨學科的基礎科學學習與教學模組之發展。在本期計畫中，我們以物理−資訊−工程跨學科學習與教學模組為目標，其內容在於採用樂高智慧型機器人作為基礎科學場域培養計算思維、提供人機互動環境之教學與學習工具，藉此提升學生基本及專業的程式設計及實務能力，並透過服務學習場域培養自主學習與創新創意能力。

肆、 計畫特色及具體內容

本計畫的主要創新處在於利用樂高智慧型機器人的趣味性與多變性，讓學生在課堂上先學會基本的操作概念、繼而發揮創意、開發新的操作模式，達成自主學習與創意學習，最終能在服務學習場域靈活發揮所學、指導國高中生，從中培養利他與服務精神，而整個培訓過程，也將為學生探索國內重點發展的智慧機器人產業奠定基礎。在

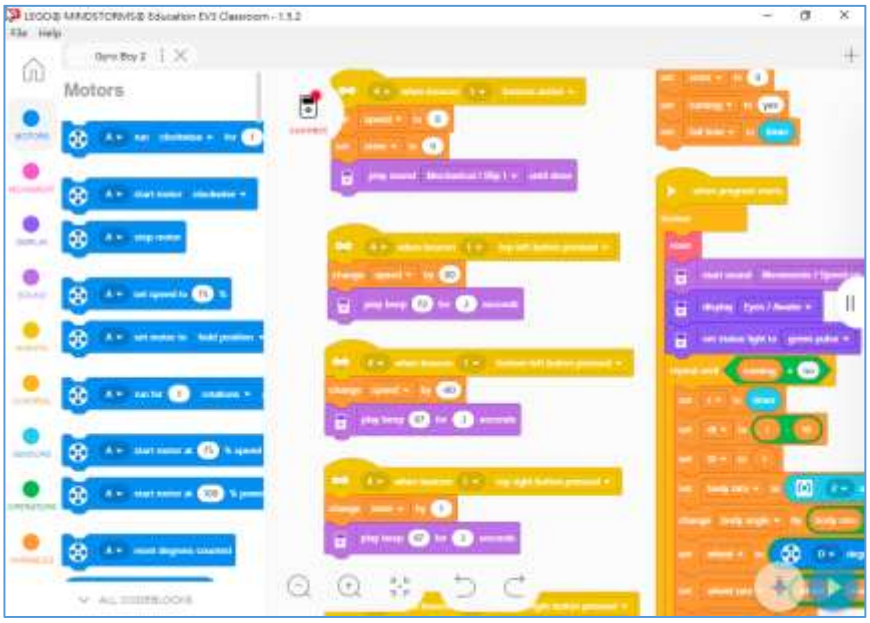
1091 的基礎科學跨學科教學之設計與實踐(三)計畫中，我們已開發了包含初階的可程式積木及進階的 LEGO MINDSTORMS Education EV3 Classroom App.圖形化程式教學，內容涵蓋系統與環境介紹、主機與程式軟體、連接阜、動作控制、訊號擷取、感測器及進階控制共十三單元；因此本計畫主要在指導學生學會基本的組裝及程序控制，再分組製作具有特殊功能的機器人，包含 Color Sorter、Gyro Boy、Puppy 及 Robot Arm，最後再讓學生進入服務場域，協助其發揮創意、進行跨領域學習與教學。其次，本計畫採行經驗證有效的國高中教學現場服務學習方案，使學生的學習過程包含老師解說與演示、學生分組實作與討論、國中教學現場觀摩與實習以及期末演示影片製作與評比四個階段(圖一)。在此流程下，學生被投進一個主動的動態學習迴圈，他們必須向老師、同學及國中小朋友證明自己四年所學，因此展現出與其他靜態課堂完全不一樣的主動學習態度。



圖一：「基礎科學跨學科教學之設計與實踐」課程的學習流程

本計畫主要在設計一套可供物理系四年級學生用於服務學習場域，協助國高中生活用樂高圖形化程式操控智慧型機器人的進階教材，操作方式包含初階的可程式積木(圖一)及進階的 SCRATCH 圖形化程式(圖二)，內容涵蓋 Color Sorter、Gyro Boy、Puppy 及 Robot Arm 四組具有特殊功能的機器人。我們採用的機型為 LEGO 樂高 MINDSTORMS EV3 45544，由物理系中程計畫支持；這是繼我們在 1091 與中山國中合作、1092 與本校機械系合作之後，首度以自己的設備來進行教學及服務學習，這也是我們在教學策略上的一大成功，更是為我們物理系的服務學習及專題教育開闢新方向。就服務學習而言，以往我們大部分與物理、自然學科合作，但這次則是資訊學科，對重視資訊知能的計算物理領域而言，深具意義。在專題教育面，物理系已有利用 arduino 及 8051 微處理器實現機電整合應用的兩個專題組，引入樂高教育機器人正

可提供另一個足資參照輝映的新方向。



圖二：樂高可程式積木　　　　　圖三：樂高 SCRATCH 圖形化程式

樂高 LEGO MINDSTORMS EV3 教育機器人風靡全球，也廣為各級學校採用，因此除樂高官網之外，各類教材甚多，在此我們參酌 LEGO® MINDSTORMS® Education EV3 的共享教材進行適用於我們場域的編譯，內容包含 Color Sorter、Gyro Boy、Puppy 及 Robot Arm 的組裝與圖形化程式操控共四單元(表一)；初學者可由此獲得具體的概念、奠定堅實的基礎，繼而藉由樂高靈活的多變性與擴充性，發揮創意，製作出各式各樣的新創作，而這也是我們希望藉以激發學生學習興趣與熱忱的一大考量。

表一：樂高智慧型機器人組裝與圖形化程式操控教材目錄

序號	章節英文名稱	章節中文名稱
1	Color Sorter	分色機器人
2	Gyro Boy	平衡機器人
3	Puppy	寵物機器人
4	Robot Arm	機械手臂

在課程實施上，我們先以六週的時間，讓學生學會、進而熟悉樂高機器人感測器的操

作及程式設定，繼而安排四週的時間進行特殊功能機器人的組裝及操作。最後。我們安排了為期四週華岡校園策展，並採機動調課方式於三天內密集進行，除作為教學成果期末策展，給予學生發揮所學的舞台，也希望為校園內跨科系的同學引入不同的視角與體驗(表二)。

表二：基礎科學跨學科教學之設計與實踐(三)課程表

週次	內容
1	基礎科學跨學科教學教材教法－樂高機器人 brick ultrasound
2	基礎科學跨學科教學教材教法－樂高機器人 brick color
3	基礎科學跨學科教學教材教法－樂高機器人 brick gyro
4	基礎科學跨學科教學教材教法－樂高機器人 notebook ultrasound
5	基礎科學跨學科教學教材教法－樂高機器人 notebook color
6	基礎科學跨學科教學教材教法－樂高機器人 notebook gyro
7	基礎科學跨學科教學教材教法－樂高機器人 Color Sorter
8	基礎科學跨學科教學教材教法－樂高機器人 Gyro Boy
9	期中成果報告
10	基礎科學跨學科教學教材教法－樂高機器人 Puppy
11	基礎科學跨學科教學教材教法－樂高機器人 Robot Arm
12	基礎科學跨學科教學教材教法－場域一: 華岡校園策展
13	基礎科學跨學科教學教材教法－場域二: 華岡校園策展
14	基礎科學跨學科教學教材教法－場域三: 華岡校園策展
15	基礎科學跨學科教學教材教法－場域四: 華岡校園策展
16	期末成果報告 I
17	期末成果報告 II
18	期末考週

圖五：1101 華岡校園策展　　　　圖六：各系逾千位同學踴躍參加

圖七：各組同學都能流利解說　　　　圖八：機器人的多樣應用吸引關注

伍、實施成效及影響（量化及質化，且說明是否達到申請時所期之學習目標與預期成效）

1. 藉由具備高度趣味性與啟發性的樂高機器人教材教具以提升學生學習興趣與學習成效。
2. 提升學生基本與專業的程式設計及實務能力。
3. 在設計不同的控制模式與解迷宮的過程中，落實學生創新創意能力之培養。
4. 強化學生物理敘事與演示的能力。
5. 透過華岡校園策展發表學習成果，讓學生藉此增進多元敘事力，培養利他精神，並向跨領域學生傳達計算思維與自動控制的概念。

綜合上述六項，本計畫已圓滿達成預期之學習目標與預期成效。

陸、 結論

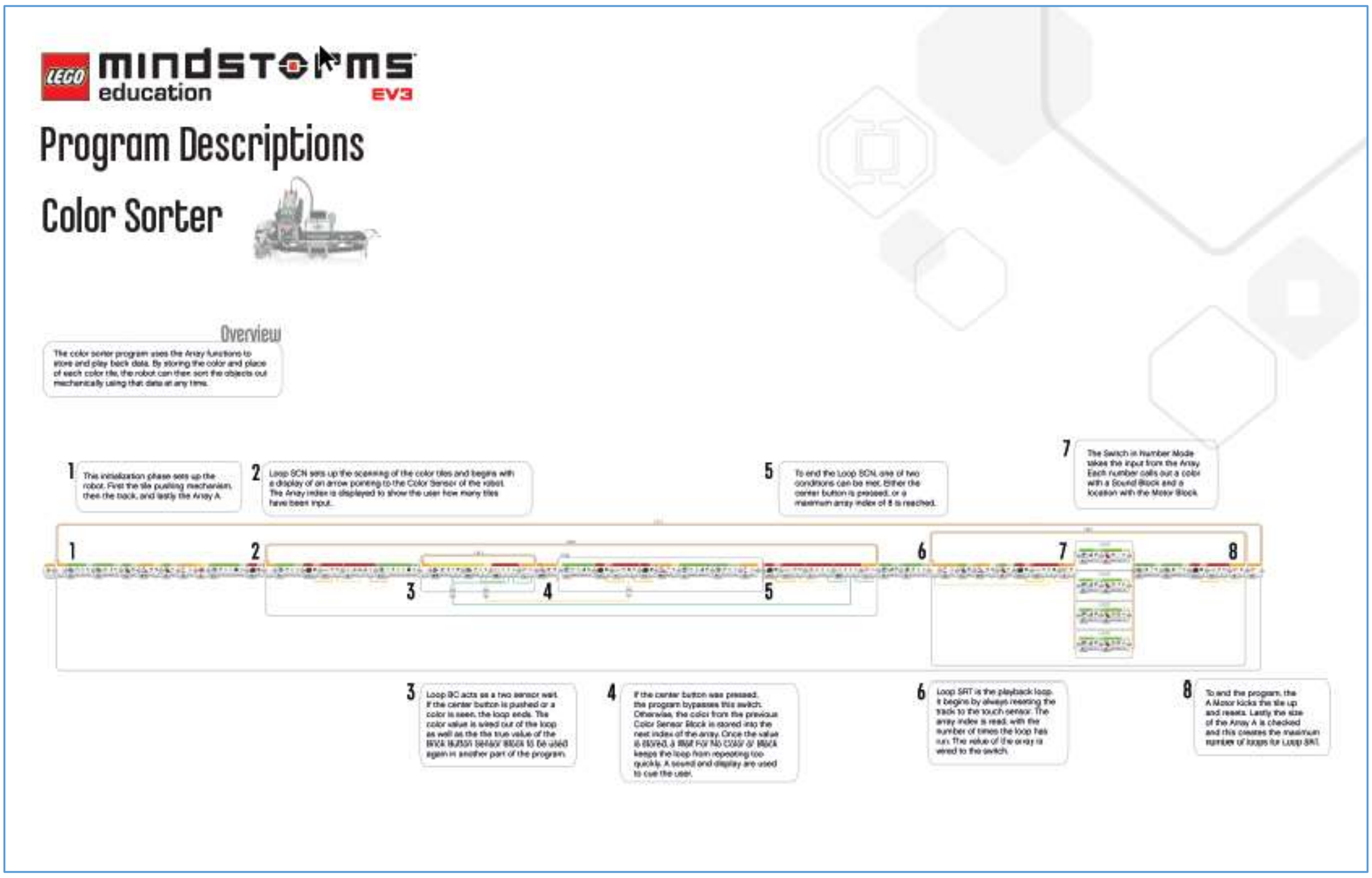
本計畫藉由設計四篇樂高圖形化程式操控智慧型機器人的教材，帶領學生透過大學跨學門全人學習場域的服務學習，培養其敘事力與利他心，在提升教學成效之同時，也落實跨域學習；而在華岡校園場域中面對跨學門學生時，我們也發現這套集計算思維與自動控制於一身的教材教法的吸引力，及其在計算思維融入學習活動中，所能發揮的巨大影響力。

柒、 附件：

講義一：「Color Sorter: 分色機器人」

講義二：「Gyro Boy：平衡機器人」
講義三：「Puppy: 寵物機器人」
講義四：「Robot Arm: 機械手臂」

1. Color Sorter



Program Descriptions

Color Sorter

Overview

The color sorter program uses the Array functions to store and play back data. By storing the color and place of each color tile, the robot can then sort the objects out mechanically using that data at any time.

1 This initialization phase sets up the robot. First the tile pushing mechanism, then the track, and lastly the Array A.

2 Loop SCN sets up the scanning of the color tiles and begins with a display of an arrow pointing to the Color Sensor of the robot. The Array index is displayed to show the user how many tiles have been input.

3 Loop BC acts as a two sensor wait. If the center button is pushed or a color is seen, the loop ends. The color value is wired out of the loop as well as the the true value of the Brick Button Sensor Block to be used again in another part of the program.

4 If the center button was pressed, the program bypasses this switch. Otherwise, the color from the previous Color Sensor Block is stored into the next index of the array. Once the value is stored, a Wait For No Color or Black keeps the loop from repeating too quickly. A sound and display are used to cue the user.

5 To end the Loop SCN, one of two conditions can be met. Either the center button is pressed, or a maximum array index of 8 is reached.

6 Loop SRT is the playback loop. It begins by always resetting the track to the touch sensor. The array index is read, with the number of times the loop has run. The value of the array is wired to the switch.

7 The Switch in Number Mode takes the input from the Array. Each number calls out a color with a Sound Block and a location with the Motor Block.

8 To end the program, the A Motor kicks the tile up and resets. Lastly the size of the Array A is checked and this creates the maximum number of loops for Loop SRT.

2. Gyro Boy

Program Descriptions

GyroBoy

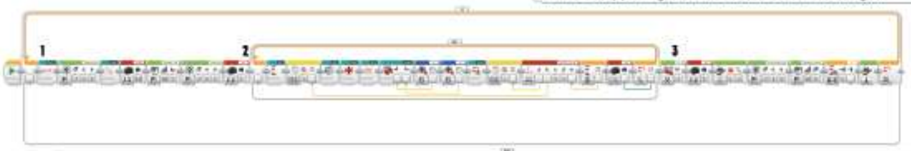
Overview

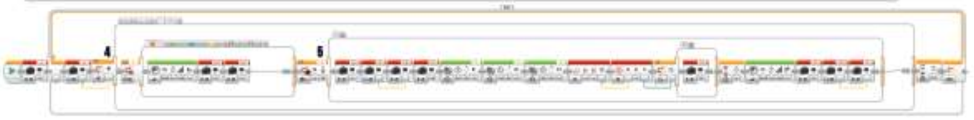
Gyroboy runs two parallel program strings. The first string, loop M, handles the data collection and balancing equations. These are all tuned to the robot and should not be changed without advanced knowledge. The second loop, BHV, handles the behavior of the robot. It allows basic control and sensor feedback. By changing the variables Cdrv and Cstr, you can make the robot do what you want.

1 Loop M begins with My Block RST, this will reset all motors and sensors. Myblock gOS checks to see how still the robot is. Once complete, the robot sets up its launch sequence. On Display graphics show an expression for the robot.

2 Loop BAL takes all of the data from the Gyro Sensor and Motor Sensors to process them to balance. Time is also used in the calculations. Timer 1 is used to calculate the time it takes to run the equation, and regulates the time between getting the data and running the balancing equation. The time needs to be regulated in order to have a more stable robot. The Unregulated Motor Blocks for Motor A and D are used to make sure that only the balancing equation regulates the movement, not any other internal calculations. Myblock CHK is there to check if the robot has fallen, and ends the loop with Variable ok if it does. The My Blocks are shown as they are to improve readability and possible customization.

3 Loop M ends by stopping the motors and behaviors. A change in graphics, sound and brick light display indicate the fallen status and is ready to reset. It should give time for the user to put the robot back on the stand, then by pressing the touch sensor, the robot can start again.





4 Loop BHV controls the behavior of the robot. State Variable S is wired to A switch in Numeric Mode with 3 options. If Variable S is 0, the Variables CDrv and CStr are set to 0 and this is the idle state of the robot. If S is 1, the robot does a launch sequence. Variable CDrv is set to 40 for 4 seconds, then back to zero then Variable S is changed to 2 in order to begin the movement state.

5 When Variable S is 2, it is the main operating and interactive state of the robot. The Color Sensor is checked and for each available color, there is a different value for variables Cstr and Cdrv. The next Switch in Ultrasonic Mode checks to see if an object is in front of the robot. If there is, the robot stops and saves its last driving condition, then prepares to turn away by moving slightly back, and waving its arms. The robot then turns randomly left or right for a few seconds and goes back to its previous driving state.

Program Descriptions

GyroBoy

RST

My Block RST resets all motors, sensors, timers and variables that are needed in this program.

Ctrl

My Block Ctrl uses two variables to control the robot. Cdrv is again used to create a target motor position. The Variable Cdrv is multiplied by tInt and subtracted from the previous mPos to create the target. Variable Cstr is multiplied by .1 and, when added to the power, Variable pwr will wire to the left drive motor. By subtracting the wire will go to the right drive motor.

GT

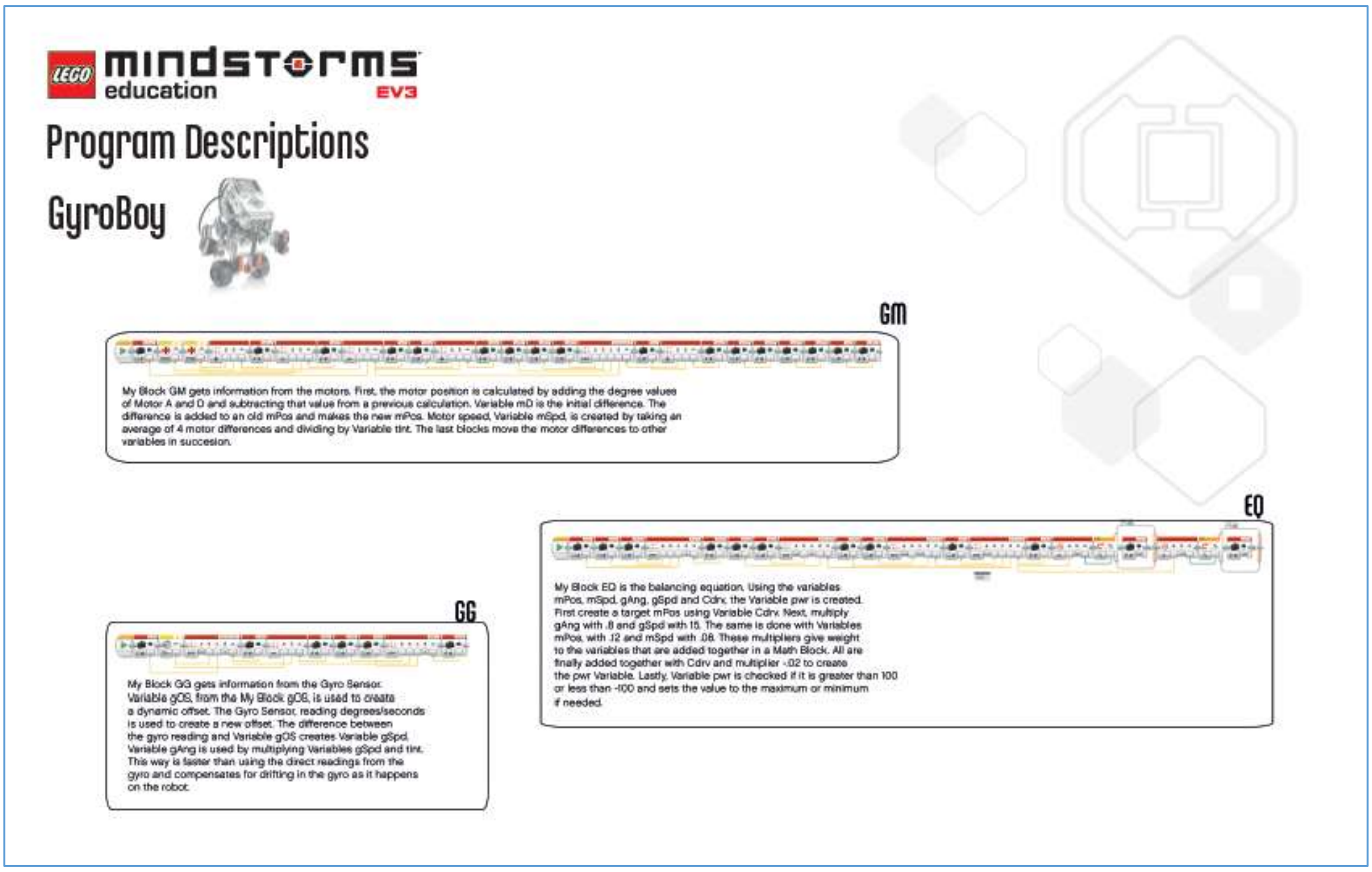
My Block GT calculates a time interval based on a timer and loop count. Timer 1 is divided by Variable cLo, which represents the loop count and wired to the Variable tInt. One is added to Variable cLo after the calculation.

Chk

My Block Chk checks if the robot has fallen. If the power is 100 for more than 2 seconds, the Variable ok will end the balancing loop and the robot can be placed on the stand again.

gOS

My Block gOS makes an offset value of the Gyro Sensor in its steady position. If the gyro is slightly drifting, this will keep the value regulated as the robot moves. The offset is also dynamically calculated while the robot is running. To calculate this value the gyro value is read and added together 250 times. In each addition, the value is checked at maximum and minimum values to make sure the robot is steady. After Loop gChk ends, the difference between gMax and gMin is checked, if the value is less than 2, the calculation will continue, otherwise the robot will check the gyro again. The average of the gSum is taken by the Math Block dividing by 250, and then this is wired to the Variable gOS.



LEGO mindstorms education EV3

Program Descriptions

GyroBoy

GM

My Block GM gets information from the motors. First, the motor position is calculated by adding the degree values of Motor A and D and subtracting that value from a previous calculation. Variable mD is the initial difference. The difference is added to an old mPos and makes the new mPos. Motor speed, Variable mSpd, is created by taking an average of 4 motor differences and dividing by Variable tInt. The last blocks move the motor differences to other variables in succesion.

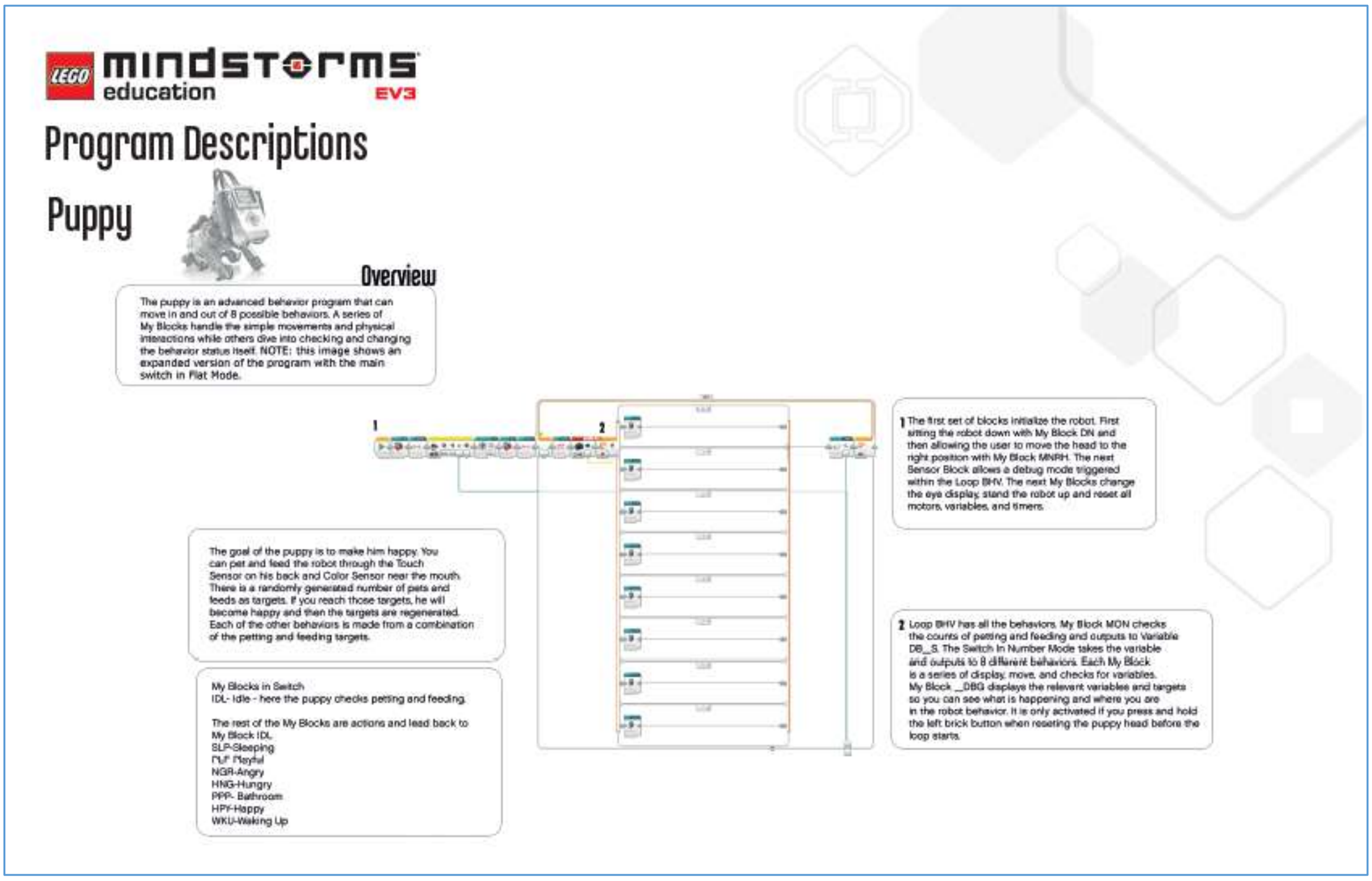
EQ

My Block EQ is the balancing equation. Using the variables mPos, mSpd, gAng, gSpd and Cdrv, the Variable pwr is created. First create a target mPos using Variable Cdrv. Next, multiply gAng with .8 and gSpd with 15. The same is done with Variables mPos, with .12 and mSpd with .08. These multipliers give weight to the variables that are added together in a Math Block. All are finally added together with Cdrv and multiplier -.02 to create the pwr Variable. Lastly, Variable pwr is checked if it is greater than 100 or less than -100 and sets the value to the maximum or minimum if needed.

GG

My Block GG gets information from the Gyro Sensor. Variable gOS, from the My Block gOS, is used to create a dynamic offset. The Gyro Sensor, reading degrees/seconds is used to create a new offset. The difference between the gyro reading and Variable gOS creates Variable gSpd. Variable gAng is used by multiplying Variables gSpd and tInt. This way is faster than using the direct readings from the gyro and compensates for drifting in the gyro as it happens on the robot.

3. Puppy



LEGO mindstorms education EV3

Program Descriptions

Puppy

Overview

The puppy is an advanced behavior program that can move in and out of 8 possible behaviors. A series of My Blocks handle the simple movements and physical interactions while others dive into checking and changing the behavior status itself. NOTE: this image shows an expanded version of the program with the main switch in Flat Mode.

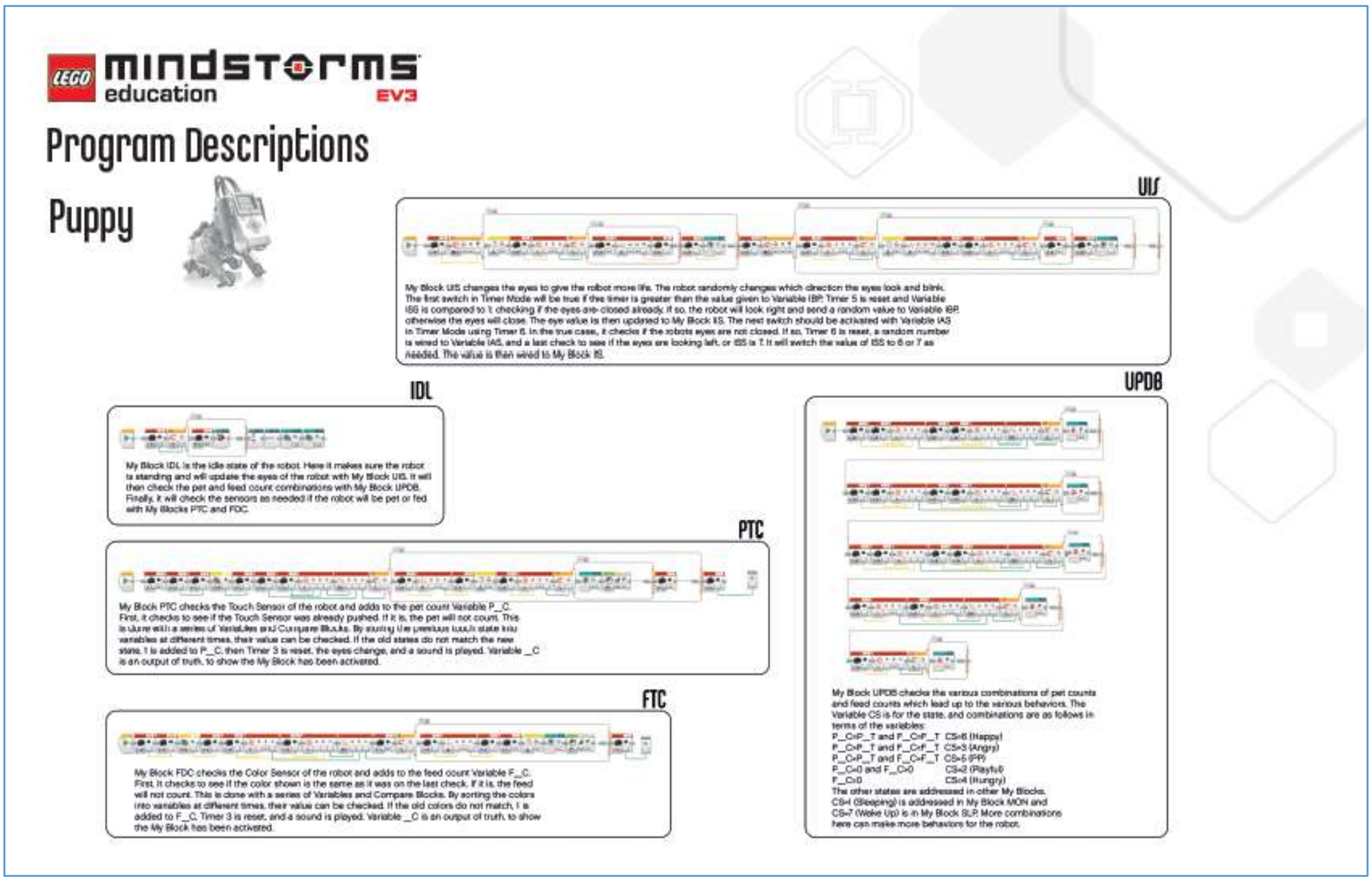
1 The first set of blocks initialize the robot. First sitting the robot down with My Block DN and then allowing the user to move the head to the right position with My Block MNRH. The next Sensor Block allows a debug mode triggered within the Loop BHV. The next My Blocks change the eye display, stand the robot up and reset all motors, variables, and timers.

The goal of the puppy is to make him happy. You can pet and feed the robot through the Touch Sensor on his back and Color Sensor near the mouth. There is a randomly generated number of pets and feeds as targets. If you reach those targets, he will become happy and then the targets are regenerated. Each of the other behaviors is made from a combination of the petting and feeding targets.

2 Loop BHV has all the behaviors. My Block MON checks the counts of petting and feeding and outputs to Variable DB_S. The Switch In Number Mode takes the variable and outputs to 8 different behaviors. Each My Block is a series of display, move, and checks for variables. My Block _DBG displays the relevant variables and targets so you can see what is happening and where you are in the robot behavior. It is only activated if you press and hold the left brick button when reseting the puppy head before the loop starts.

My Blocks in Switch
IDL- Idle - here the puppy checks petting and feeding.

The rest of the My Blocks are actions and lead back to My Block IDL
SLP-Sleeping
PLY-Playful
NGR-Angry
HNG-Hungry
PPP- Bathroom
HPY-Happy
WKU-Waking Up



mindstorms education EV3

Program Descriptions

Puppy

UIS

My Block UIS changes the eyes to give the robot more life. The robot randomly changes which direction the eyes look and blink. The first switch in Timer Mode will be true if thee timer is greater than the value given to Variable IBP. Timer 5 is reset and Variable ISS is compared to 1, checking if the eyes are closed already. If so, the robot will look right and send a random value to Variable IBP, otherwise the eyes will close. The eye value is then updated to My Block IS. The next switch should be activated with Variable IAS in Timer Mode using Timer 6. In the true case, it checks if the robots eyes are not closed. If so, Timer 6 is reset, a random number is wired to Variable IAS, and a last check to see if the eyes are looking left, or ISS is 7. It will switch the value of ISS to 6 or 7 as needed. The value is then wired to My Block IS.

IDL

My Block IDL is the idle state of the robot. Here it makes sure the robot is standing and will update the eyes of the robot with My Block UIS. It will then check the pet and feed count combinations with My Block UPDB. Finally, it will check the sensors as needed if the robot will be pet or fed with My Blocks PTC and FDC.

PTC

My Block PTC checks the Touch Sensor of the robot and adds to the pet count Variable P__C. First, it checks to see if the Touch Sensor was already pushed. If it is, the pet will not count. This is done with a series of Variables and Compare Blocks. By storing the previous touch state into variables at different times, their value can be checked. If the old states do not match the new state, 1 is added to P__C, then Timer 3 is reset, the eyes change, and a sound is played. Variable __C is an output of truth, to show the My Block has been activated.

FDC

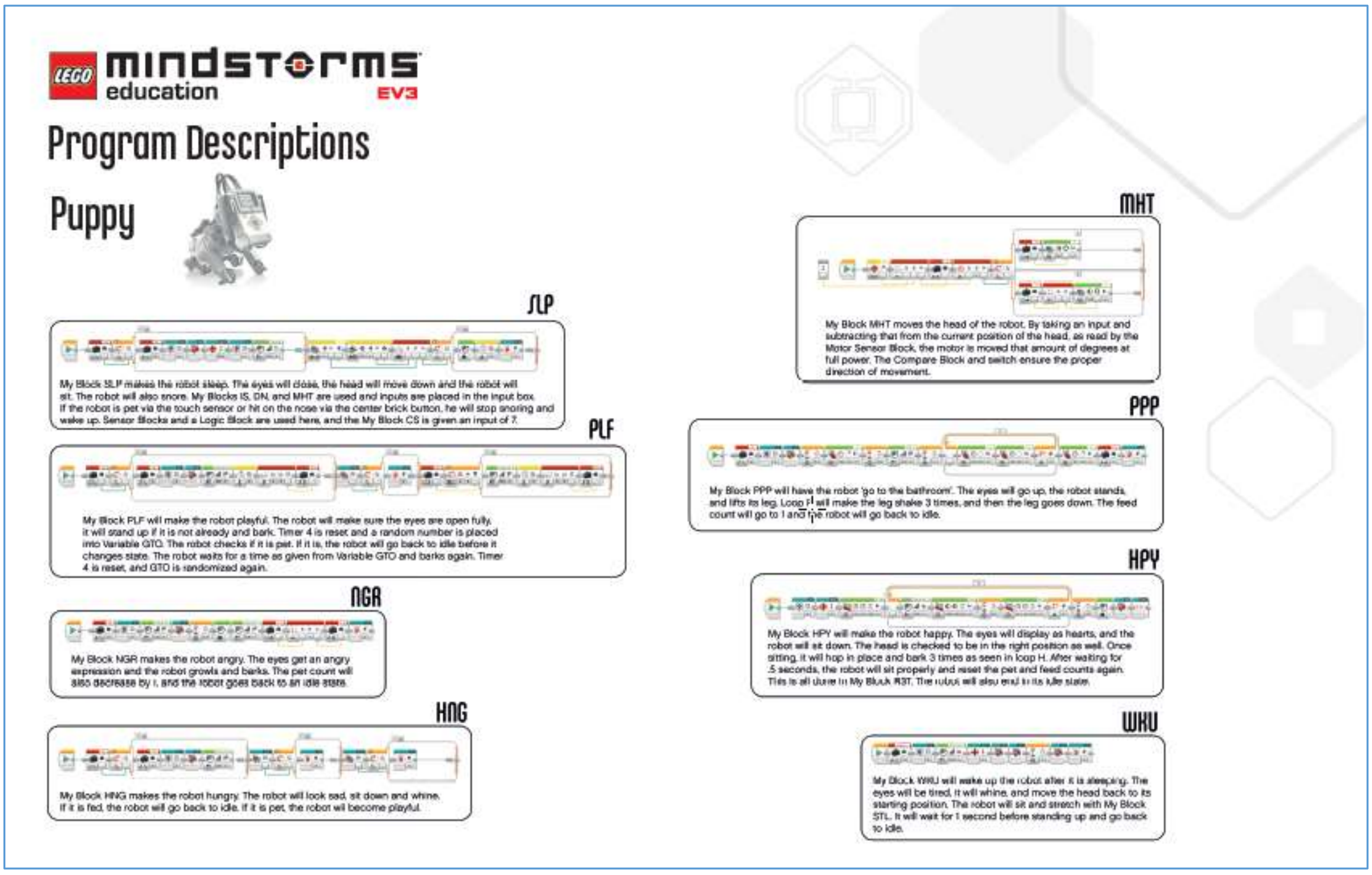
My Block FDC checks the Color Sensor of the robot and adds to the feed count Variable F__C. First, it checks to see if the color shown is the same as it was on the last check. If it is, the feed will not count. This is done with a series of Variables and Compare Blocks. By sorting the colors into variables at different times, their value can be checked. If the old colors do not match, 1 is added to F__C. Timer 3 is reset, and a sound is played. Variable __C is an output of truth, to show the My Block has been activated.

UPDB

My Block UPDB checks the various combinations of pet counts and feed counts which lead up to the various behaviors. The Variable CS is for the state, and combinations are as follows in terms of the variables:

P__C>P__T and F__C>F__T	CS=6 (Happy)
P__C>P__T and F__C<F__T	CS=3 (Angry)
P__C<P__T and F__C>F__T	CS=5 (PP)
P__C>0 and F__C>0	CS=2 (Playful)
F__C>0	CS=4 (Hungry)

The other states are addressed in other My Blocks. CS=1 (Sleeping) is addressed in My Block MON and CS=7 (Wake Up) is in My Block SLP. More combinations here can make more behaviors for the robot.



mindstorms education EV3

Program Descriptions

Puppy

SLP

My Block SLP makes the robot sleep. The eyes will close, the head will move down and the robot will sit. The robot will also snore. My Blocks IS, DN, and MHT are used and inputs are placed in the input box. If the robot is pet via the touch sensor or hit on the nose via the center brick button, he will stop snoring and wake up. Sensor Blocks and a Logic Block are used here, and the My Block CS is given an input of 7.

PLF

My Block PLF will make the robot playful. The robot will make sure the eyes are open fully, it will stand up if it is not already and bark. Timer 4 is reset and a random number is placed into Variable GTO. The robot checks if it is pet. If it is, the robot will go back to idle before it changes state. The robot waits for a time as given from Variable GTO and barks again. Timer 4 is reset, and GTO is randomized again.

NGR

My Block NGR makes the robot angry. The eyes get an angry expression and the robot growls and barks. The pet count will also decrease by 1, and the robot goes back to an idle state.

HNG

My Block HNG makes the robot hungry. The robot will look sad, sit down and whine. If it is fed, the robot will go back to idle. If it is pet, the robot will become playful.

MHT

My Block MHT moves the head of the robot. By taking an input and subtracting that from the current position of the head, as read by the Motor Sensor Block, the motor is moved that amount of degrees at full power. The Compare Block and switch ensure the proper direction of movement.

PPP

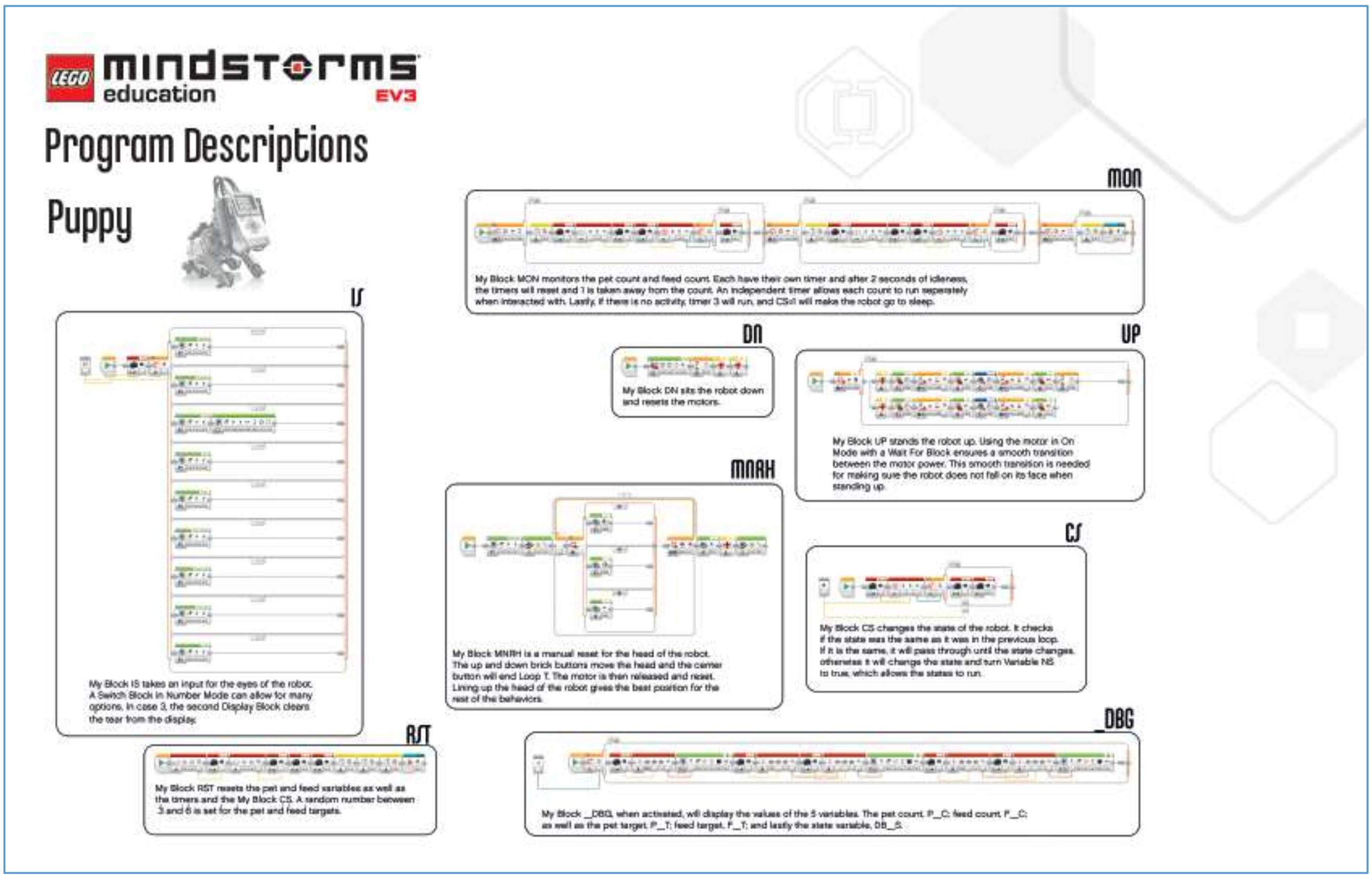
My Block PPP will have the robot 'go to the bathroom'. The eyes will go up, the robot stands, and lifts its leg. Loop F will make the leg shake 3 times, and then the leg goes down. The feed count will go to 1 and the robot will go back to idle.

HPY

My Block HPY will make the robot happy. The eyes will display as hearts, and the robot will sit down. The head is checked to be in the right position as well. Once sitting, it will hop in place and bark 3 times as seen in loop H. After waiting for .5 seconds, the robot will sit properly and reset the pet and feed counts again. This is all done in My Block RST. The robot will also end in its idle state.

WKU

My Block WKU will wake up the robot after it is sleeping. The eyes will be tired, it will whine, and move the head back to its starting position. The robot will sit and stretch with My Block STL. It will wait for 1 second before standing up and go back to idle.



LEGO mindstorms education EV3

Program Descriptions

Puppy

MON

My Block MON monitors the pet count and feed count. Each have their own timer and after 2 seconds of idleness, the timers will reset and 1 is taken away from the count. An independent timer allows each count to run seperately when interacted with. Lastly, if there is no activity, timer 3 will run, and CS=1 will make the robot go to sleep.

IS

My Block IS takes an input for the eyes of the robot. A Switch Block in Number Mode can allow for many options. In case 3, the second Display Block clears the tear from the display.

DN

My Block DN sits the robot down and resets the motors.

UP

My Block UP stands the robot up. Using the motor in On Mode with a Wait For Block ensures a smooth transition between the motor power. This smooth transition is needed for making sure the robot does not fall on its face when standing up.

MNRH

My Block MNRH is a manual reset for the head of the robot. The up and down brick buttons move the head and the center button will end Loop T. The motor is then released and reset. Lining up the head of the robot gives the best position for the rest of the behaviors.

CS

My Block CS changes the state of the robot. It checks if the state was the same as it was in the previous loop. If it is the same, it will pass through until the state changes, otherwise it will change the state and turn Variable NS to true, which allows the states to run.

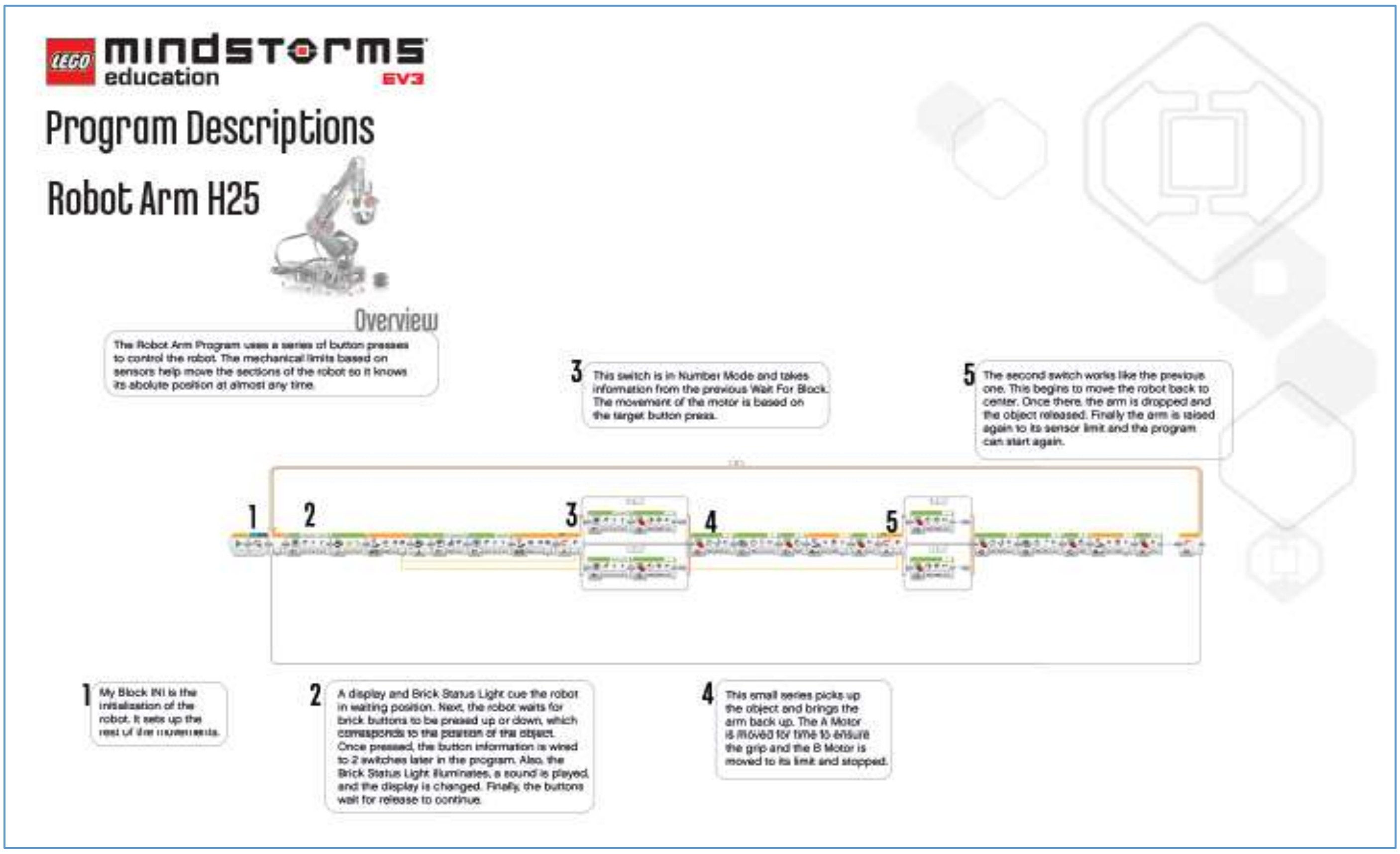
RST

My Block RST resets the pet and feed variables as well as the timers and the My Block CS. A random number between 3 and 6 is set for the pet and feed targets.

_DBG

My Block _DBG, when activated, will display the values of the 5 variables. The pet count, P_C; feed count, F_C; as well as the pet target, P_T; feed target, F_T; and lastly the state variable, DB_S.

4. Robot Arm



LEGO mindstorms education EV3

Program Descriptions

Robot Arm H25

Overview

The Robot Arm Program uses a series of button presses to control the robot. The mechanical limits based on sensors help move the sections of the robot so it knows its abolute position at almost any time.

1 My Block INI is the initialization of the robot. It sets up the rest of the movements.

2 A display and Brick Status Light cue the robot in waiting position. Next, the robot waits for brick buttons to be pressed up or down, which corresponds to the position of the object. Once pressed, the button information is wired to 2 switches later in the program. Also, the Brick Status Light illuminates, a sound is played, and the display is changed. Finally, the buttons wait for release to continue.

3 This switch is in Number Mode and takes information from the previous Wait For Block. The movement of the motor is based on the target button press.

4 This small series picks up the object and brings the arm back up. The A Motor is moved for time to ensure the grip and the B Motor is moved to its limit and stopped.

5 The second switch works like the previous one. This begins to move the robot back to center. Once there, the arm is dropped and the object released. Finally the arm is raised again to its sensor limit and the program can start again.