STAFF CORNER

SLQ Wiki Fabrication Lab 2026/04/24 09:56

STAFF CORNER

Staff Resources

- Online information on the Lego League: https://firstaustralia.org/programs/first-lego-league/
- Lego Robotics Newsletter
- Expectations on Parents and Participants
- Example Email to Mentors

2019 Staff Training

Library staff members received training on the 20th of February. What follows is an overview of discussion topics and material.

Have a look at the **First Lego League Webinar on EV3 and Mindstorms**. Feel free to utilize the videos for session preparation as well as throughout your sessions.

IMPORTANT: ENSURE THAT THE BATTERY OF YOUR EV3 IS FULLY CHARGED AT THE START OF EACH SESSION, AS THE BATTERY VOLTAGE WILL CHANGE THE CHARACTERISTICS OF THE MOTOR.

Control Panel

In the Lego Mindstorms EV3 software you will find a control panel divided into three sections.

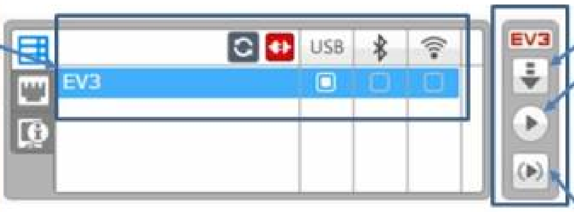
HINT: While your EV3 is connected to the computer, it won't turn off.

Files:

- Conection Tab
- Settings Tab

Connection Tab

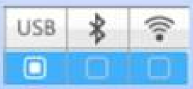
Used to control & monitor connections and code on the EV3 brick.



Connection Panel

- Disconnect Button
- Refresher Button

As well as the connection status



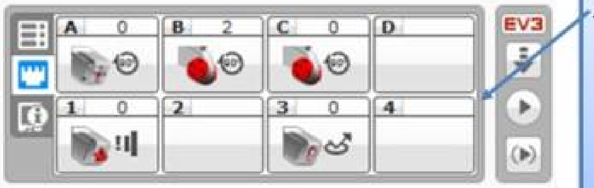
This EV3 is connected to the computer via USB.

Play Panel

- Download your Code
- Download and run your code

You can run several PLAY BUTTONS at the same time

- Run selected code – select one play button



Motor & Sensor Port Panel

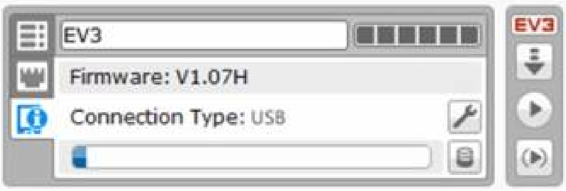
- **A-B**: Connected Motors
- **1-4**: Connected Sensors

Example:

A = Medium motor	1 = Touch sensor
B&C = Large motor	3 = Colour sensor

Settings Tab

Used to control & monitor connections and code on the EV3 brick.



EV3

Firmware: V1.07H

Connection Type: USB

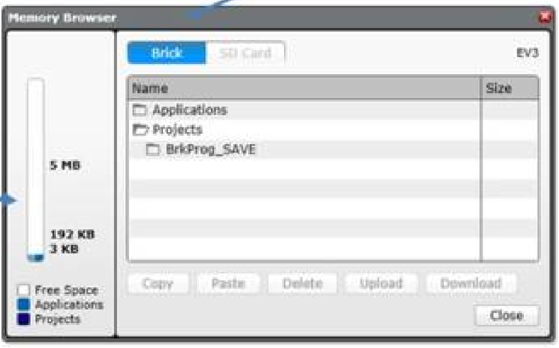
Settings & Database Panel

- You can change the name of the Brick EV3
- Check Battery level of the brick
- Settings which will be set by library staff members
- Database where you can delete and manage your code

Database > Memory Browser

When pressing the database symbol the below Memory Browser opens and in this instance shows an empty EV3 brick, only the factory settings are currently installed.

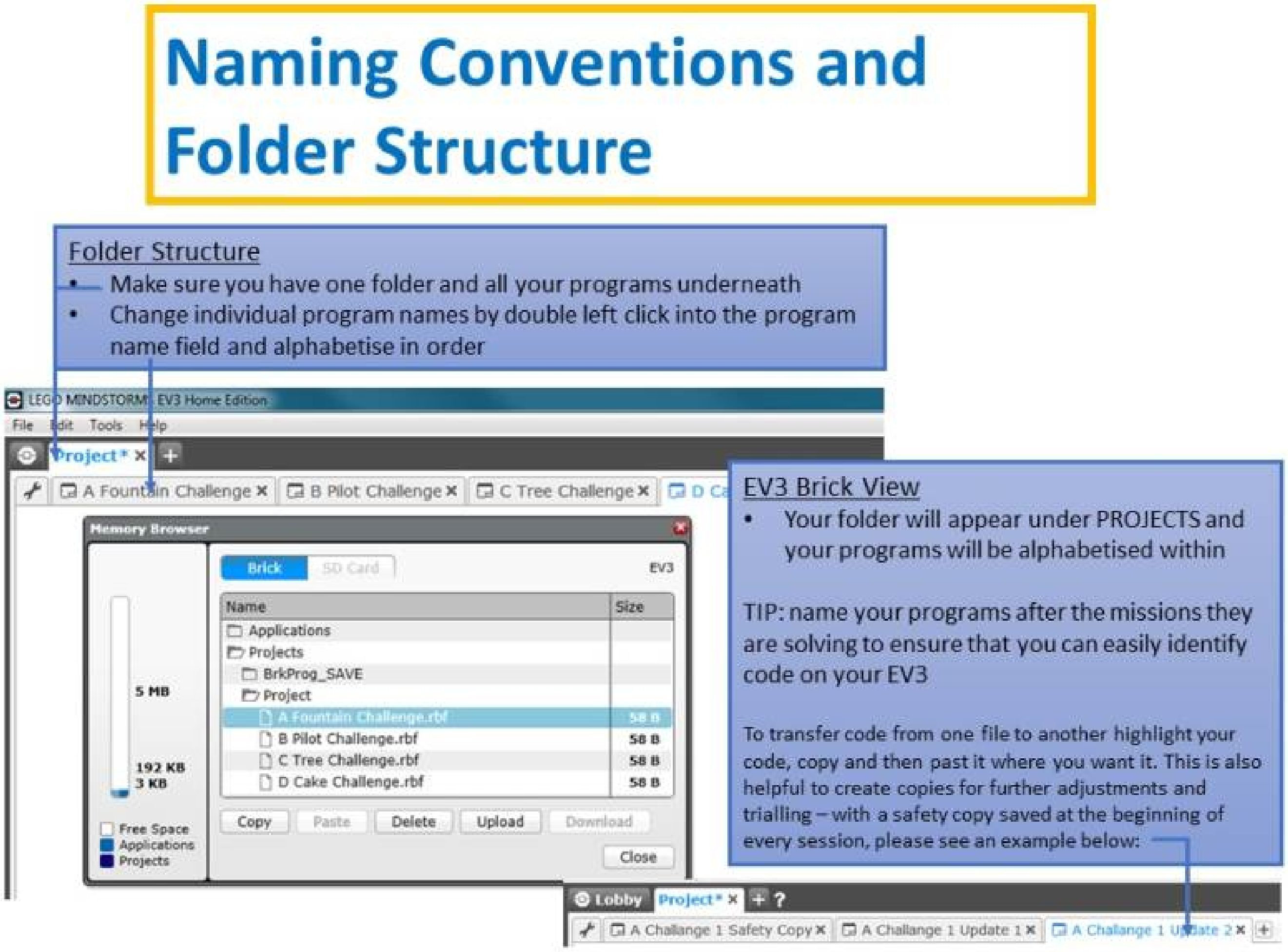
- Check the available space on your brick
- View the saved code under Projects
- Copy / Paste / Delete / Upload or Download code



Naming Conventions and Folder Structure

File:

Control Panel



Naming Conventions and Folder Structure

Folder Structure

- Make sure you have one folder and all your programs underneath
- Change individual program names by double left click into the program name field and alphabetise in order

EV3 Brick View

- Your folder will appear under PROJECTS and your programs will be alphabetised within

TIP: name your programs after the missions they are solving to ensure that you can easily identify code on your EV3

To transfer code from one file to another highlight your code, copy and then past it where you want it. This is also helpful to create copies for further adjustments and trialling – with a safety copy saved at the beginning of every session, please see an example below:

Lego Robotics Mat

- Set-up & Pack-up
    - Create a tub where each challenge piece has it's place and a picture of the correct pack-up is taped to the bottom and top of the tub.
    - Make sure each team member has a role to play in the set-up of the table.
    - Ensure that the table set up is clear to all team members and a printed copy of the correct table set up is available at all times.
- Lining up the robot
    - The lining up of the robot is one of the most crucial tasks when tackling a Lego robotics challenge, choose therefore one or two points on the mat as the designated start point.
    - Ensure that all team members know the designated start points and use them for their code
    - The challenge mat cannot be altered to show starting points but you can use Lego in the base to make it easier for the team.
    - Challenge mats will be cut at the beginning of the 10 week program so do not use the back of the table as an indicator, depending on the table you use during the Tech Fest

those line ups will not work.

Session Management

Time Management

Throughout the ten week program each library will hold one two hour session each week and if possible two further four hour sessions closer to the Tech Fest for concentrated work on individual challenge tasks.

It is important that the structure outlined in the 10 week overview is followed as closely as possible to ensure that all teams have the same chance at completing as many challenges as possible. Keep in mind that we try for 4 - 5 challenges for the Tech Fest, otherwise you might not be able to run all challenges you have solved within the 2.5 minutes allocated.

Break your sessions up into workstations and rotate team members as needed. In the 10 week overview you will see that each session is at least broken into two parts or workstations to allow for a change.

Behavior Management

Ensure that your team takes breaks. If your team members are fatigued they will not be able to perform or behave. This can be anything from a 10 minute snack time to a 30 minute outdoor break, depending on the length of your session.

Behavior management plays an important part and is the responsibility of the library staff and parent. Follow library policies and your own common sense when calling your team member to order. Below two approaches that worked well last year in Moranbah.

Timeout Cards:

- the Timeout Card is given to participants who do not follow the core values and thereby disrespect either the library space, resources, staff, mentors or team members
- The parent and participant will use that time to review the core values and team member responsibilities
    - cb_ev3_timeout-card.pptx

Leadership Points:

- One Leadership point is awarded by each mentor each session for exceptional demonstration of core values and teamwork
- The library staff keeps a running tally of the leadership points
- The team member with the most leadership points will run the EV3 on the competition day
- The next runner up can have also an equally important task and so on

Leadership Points are to reward behavior and to encourage the participants to review the core values regularly.

Teamwork

Establish little working groups that can be swapped regularly to encourage your team members to help and work with each other. Follow the core values and establish an atmosphere where the team members first consult with each other on a problem before asking an adult. Also make sure that problems are tackled and resolved as a team and not as an answer from an adult.

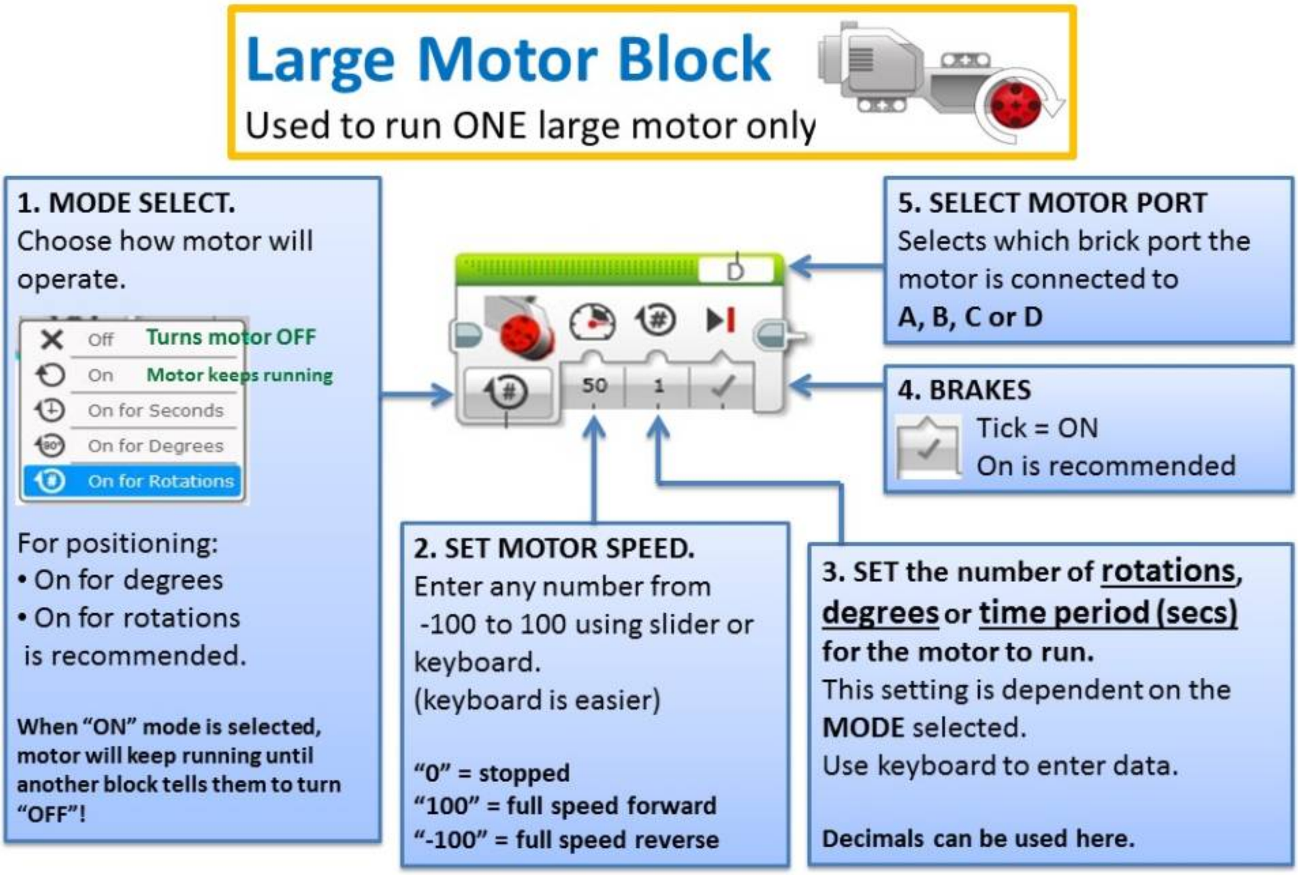
Tech Fest Preparation

- Ensure that you are starting with the run-through as outlined in the 10 week overview, these are crucial to ensure that the team understands what will happen on the Tech Fest day and can prepare.
- Clean the code with your team and take your teams code with you to the Tech Fest to ensure nothing is missing.
- Charge your EV3, the EV3 runs slower when the battery is nearly empty.
- Make sure everyone is clear about their role on the day and has prepared their team/individual introduction

Motor or Action Blocks

Files:

- Large Motor Block
- Medium Motor Block
- Tank Motor Block
- Steering Motor Block



Large Motor Block

Used to run ONE large motor only

1. MODE SELECT.
Choose how motor will operate.

For positioning:
- On for degrees
- On for rotations

is recommended.

When "ON" mode is selected, motor will keep running until another block tells them to turn "OFF"!

5. SELECT MOTOR PORT
Selects which brick port the motor is connected to
A, B, C or D

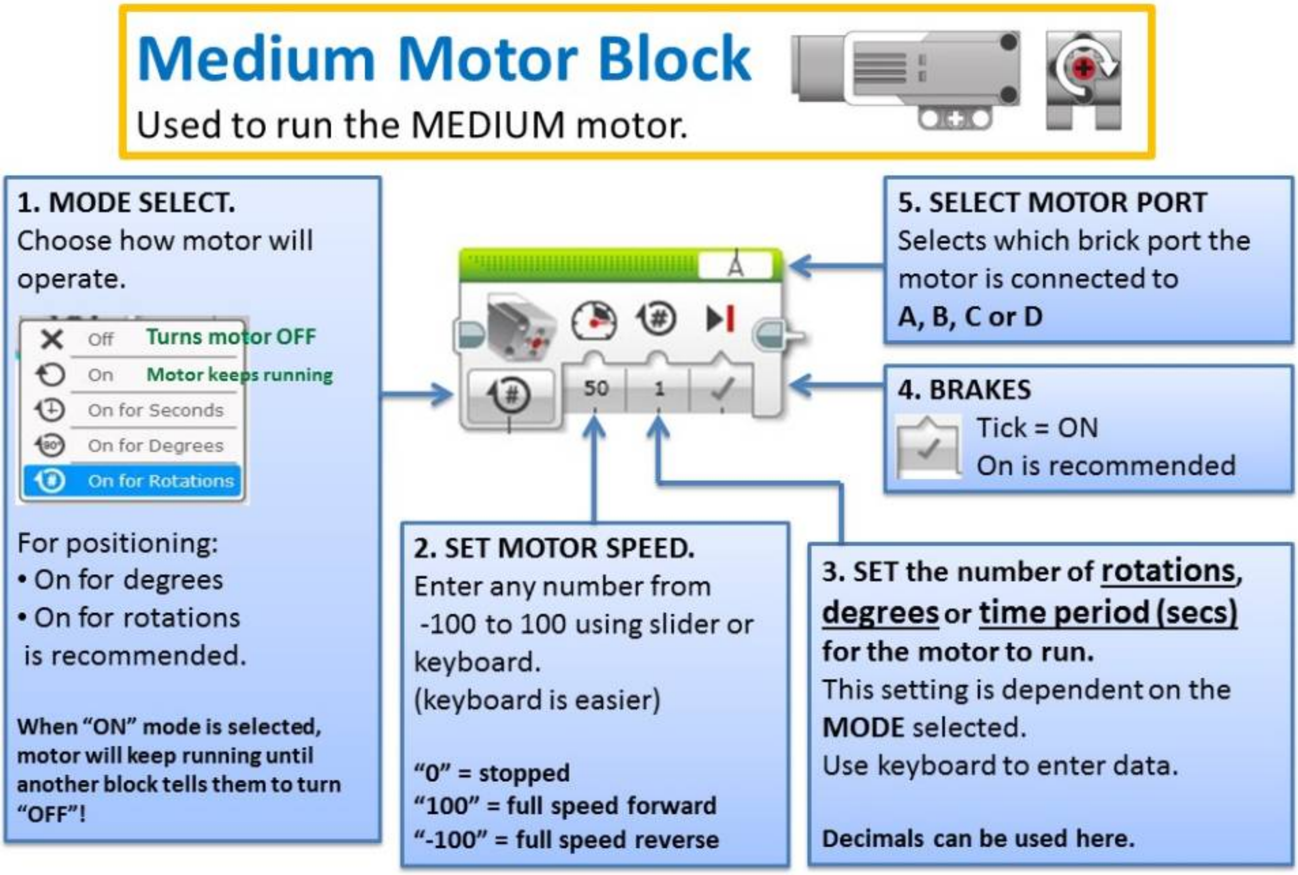
4. BRAKES
Tick = ON
On is recommended

2. SET MOTOR SPEED.
Enter any number from -100 to 100 using slider or keyboard.
(keyboard is easier)

"0" = stopped
"100" = full speed forward
"-100" = full speed reverse

3. SET the number of rotations, degrees or time period (secs) for the motor to run.
This setting is dependent on the **MODE** selected.
Use keyboard to enter data.

Decimals can be used here.



Medium Motor Block

Used to run the MEDIUM motor.

1. MODE SELECT.
Choose how motor will operate.

For positioning:
- On for degrees
- On for rotations

is recommended.

When "ON" mode is selected, motor will keep running until another block tells them to turn "OFF"!

5. SELECT MOTOR PORT
Selects which brick port the motor is connected to **A, B, C or D**

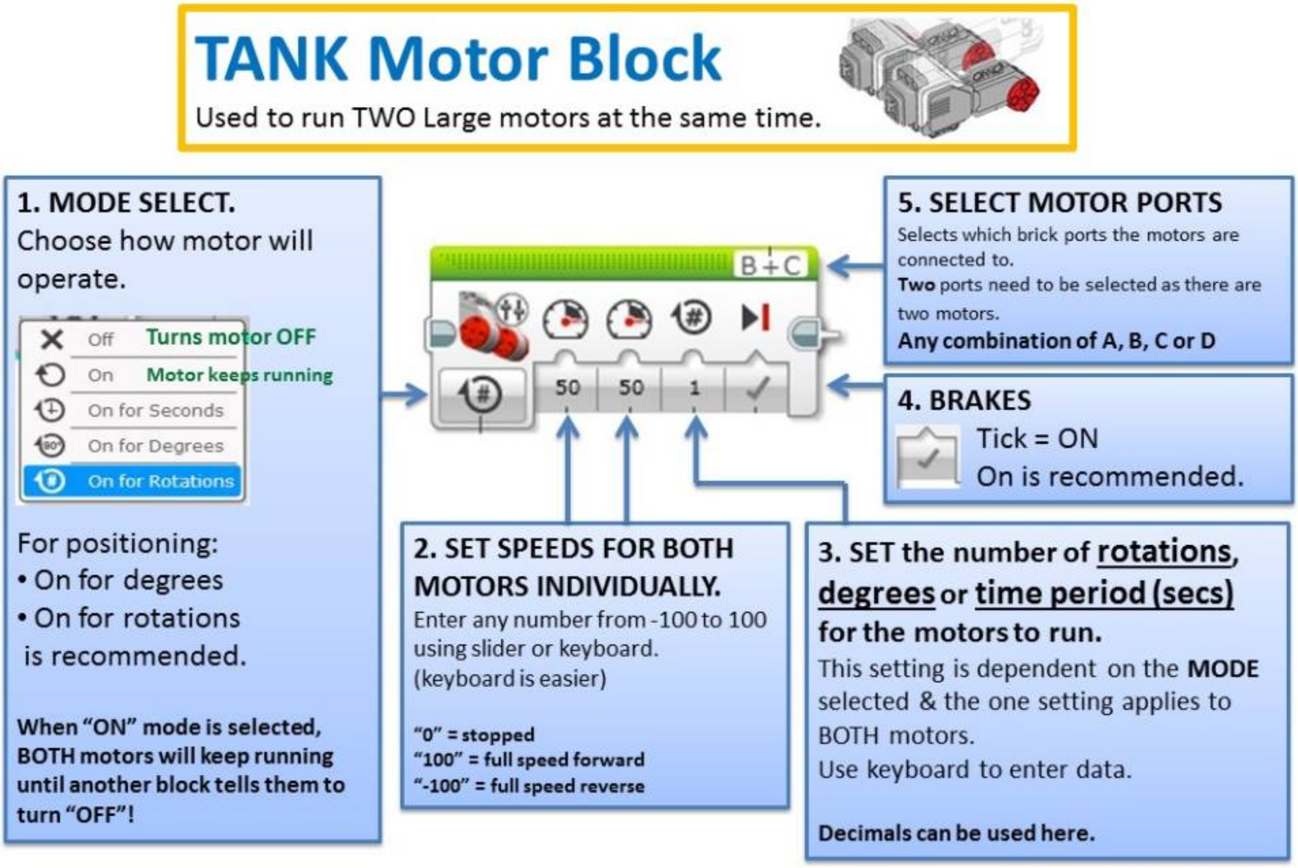
4. BRAKES
Tick = ON
On is recommended

2. SET MOTOR SPEED.
Enter any number from -100 to 100 using slider or keyboard.
(keyboard is easier)

"0" = stopped
"100" = full speed forward
"-100" = full speed reverse

3. SET the number of <u>rotations</u>, <u>degrees</u> or <u>time period (secs)</u> for the motor to run.
This setting is dependent on the **MODE** selected.
Use keyboard to enter data.

Decimals can be used here.



TANK Motor Block

Used to run TWO Large motors at the same time.

1. MODE SELECT.
Choose how motor will operate.

For positioning:
• On for degrees
• On for rotations
is recommended.

When "ON" mode is selected, BOTH motors will keep running until another block tells them to turn "OFF"!

2. SET SPEEDS FOR BOTH MOTORS INDIVIDUALLY.
Enter any number from -100 to 100 using slider or keyboard. (keyboard is easier)

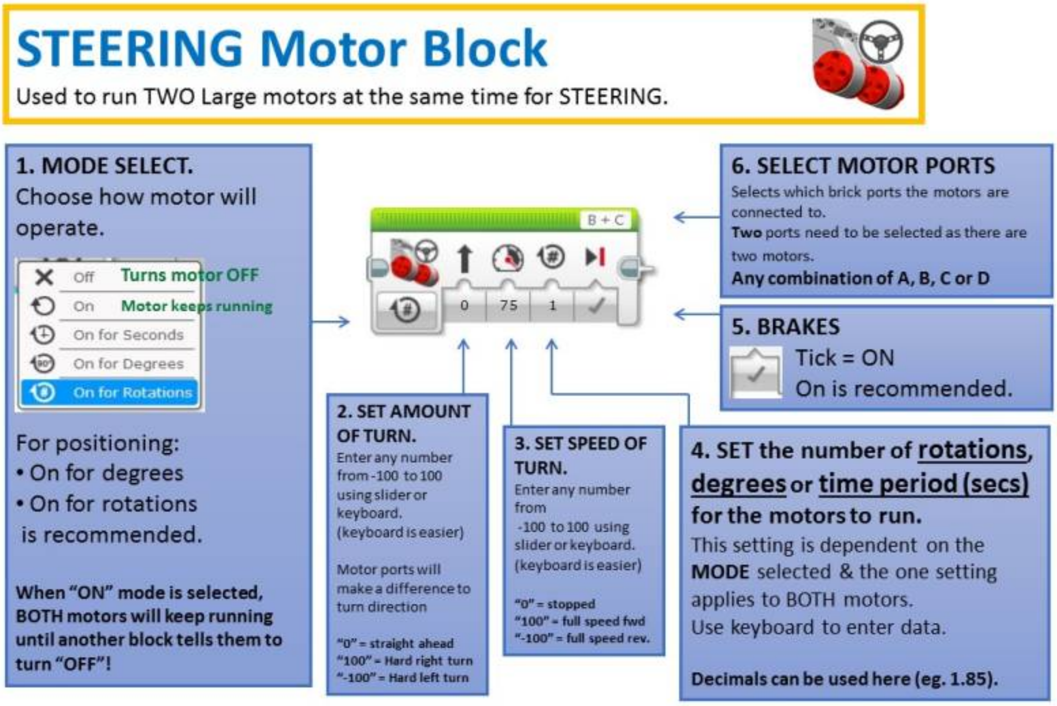
"0" = stopped
"100" = full speed forward
"-100" = full speed reverse

3. SET the number of rotations, degrees or time period (secs) for the motors to run.
This setting is dependent on the MODE selected & the one setting applies to BOTH motors.
Use keyboard to enter data.

Decimals can be used here.

4. BRAKES
Tick = ON
On is recommended.

5. SELECT MOTOR PORTS
Selects which brick ports the motors are connected to.
Two ports need to be selected as there are two motors.
Any combination of A, B, C or D



STEERING Motor Block

Used to run TWO Large motors at the same time for STEERING.

1. MODE SELECT.
Choose how motor will operate.

For positioning:
• On for degrees
• On for rotations
is recommended.

When "ON" mode is selected, BOTH motors will keep running until another block tells them to turn "OFF"!

2. SET AMOUNT OF TURN.
Enter any number from -100 to 100 using slider or keyboard. (keyboard is easier)

Motor ports will make a difference to turn direction

"0" = straight ahead
"100" = Hard right turn
"-100" = Hard left turn

3. SET SPEED OF TURN.
Enter any number from -100 to 100 using slider or keyboard. (keyboard is easier)

"0" = stopped
"100" = full speed fwd
"-100" = full speed rev.

4. SET the number of rotations, degrees or time period (secs) for the motors to run.
This setting is dependent on the MODE selected & the one setting applies to BOTH motors.
Use keyboard to enter data.

Decimals can be used here (eg. 1.85).

5. BRAKES
Tick = ON
On is recommended.

6. SELECT MOTOR PORTS
Selects which brick ports the motors are connected to.
Two ports need to be selected as there are two motors.
Any combination of A, B, C or D

Motor Blocks Handy Hints

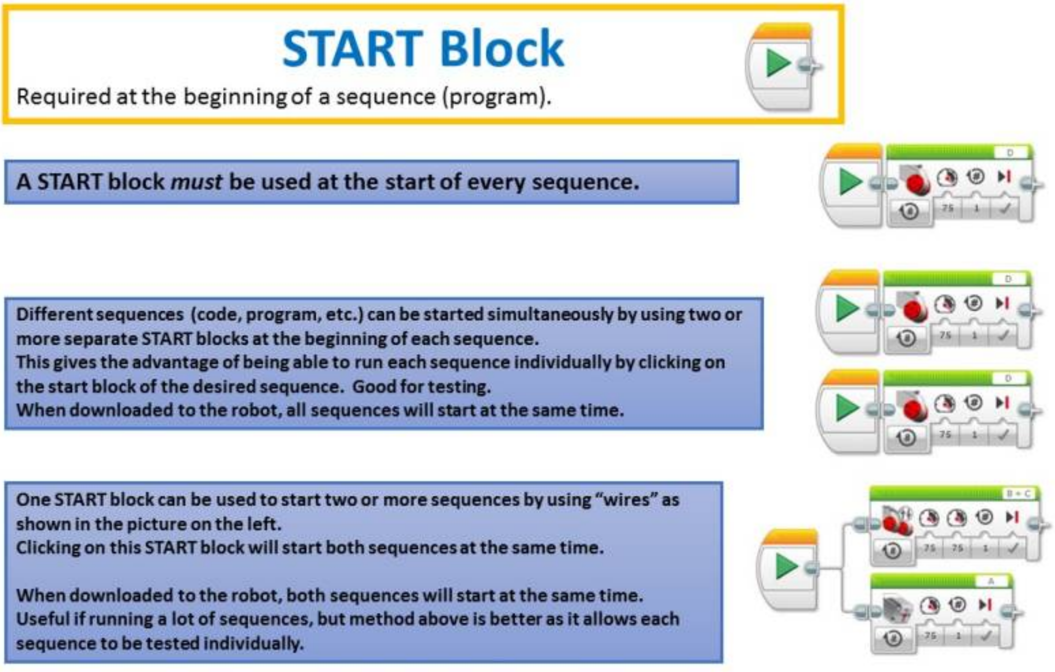
- ALWAYS ensure the battery is fully charged. Battery state WILL affect motor speed & power.

- ALWAYS ensure that the large motors have their gears “preloaded” to ensure “backlash” doesn’t cause the robot to skew on start up.
- Consider adding a small delay at the start of sequence (code, program, etc) to allow the robot operator to get their hand clear after pressing the “go” button.
- Moving slowly when completing a challenge will enhance accuracy. Once task is completed, return to base as fast as possible. Accuracy isn’t required for return.
- When using forks or other attachments, use code to “settle” the attachment on the ground before raising to required position. This will improve the “repeatability” of the action.
- Consider “gearing” the medium motor to increase torque & accuracy.
- ALWAYS ensure the battery is fully charged. Battery state WILL affect motor speed & power.

- ALWAYS ensure the battery is fully charged. Battery state WILL affect motor speed & power.

Flow Control/Sensor Blocks

- Start Block
- Wait Block:
  - Time Delay Block
  - Colour Sensor Block
  - Ultrasonic Sensor Block
  - Touch Sensor Block
- Single Loop Block
- Double Loop Block



START Block

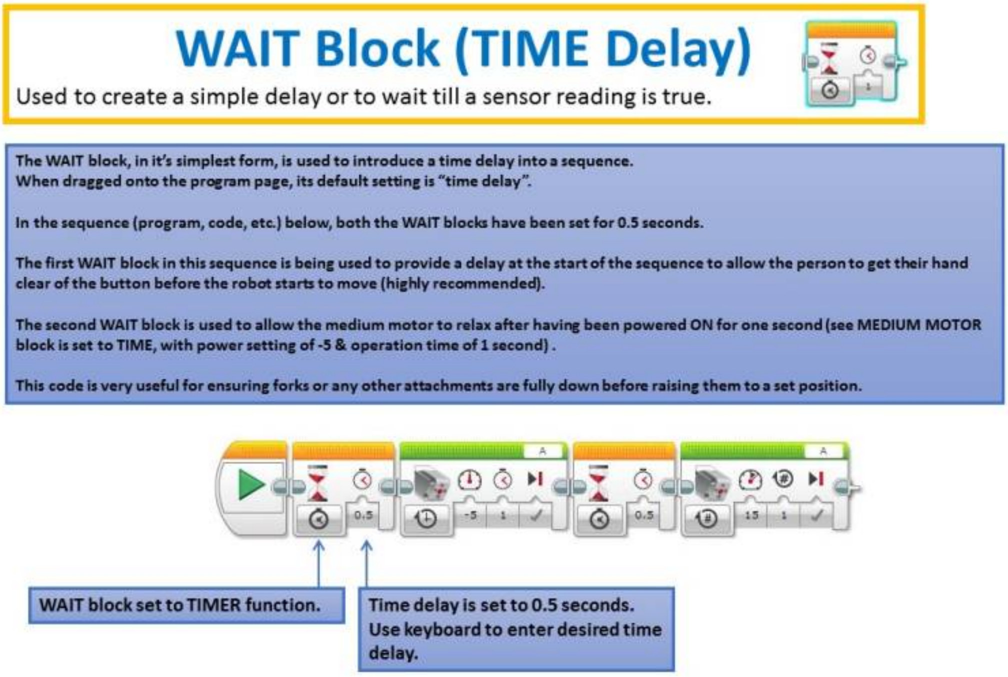
Required at the beginning of a sequence (program).

A START block *must* be used at the start of every sequence.

Different sequences (code, program, etc.) can be started simultaneously by using two or more separate START blocks at the beginning of each sequence.
This gives the advantage of being able to run each sequence individually by clicking on the start block of the desired sequence. Good for testing.
When downloaded to the robot, all sequences will start at the same time.

One START block can be used to start two or more sequences by using “wires” as shown in the picture on the left.
Clicking on this START block will start both sequences at the same time.

When downloaded to the robot, both sequences will start at the same time.
Useful if running a lot of sequences, but method above is better as it allows each sequence to be tested individually.



WAIT Block (TIME Delay)

Used to create a simple delay or to wait till a sensor reading is true.

The WAIT block, in it's simplest form, is used to introduce a time delay into a sequence.
When dragged onto the program page, its default setting is "time delay".

In the sequence (program, code, etc.) below, both the WAIT blocks have been set for 0.5 seconds.

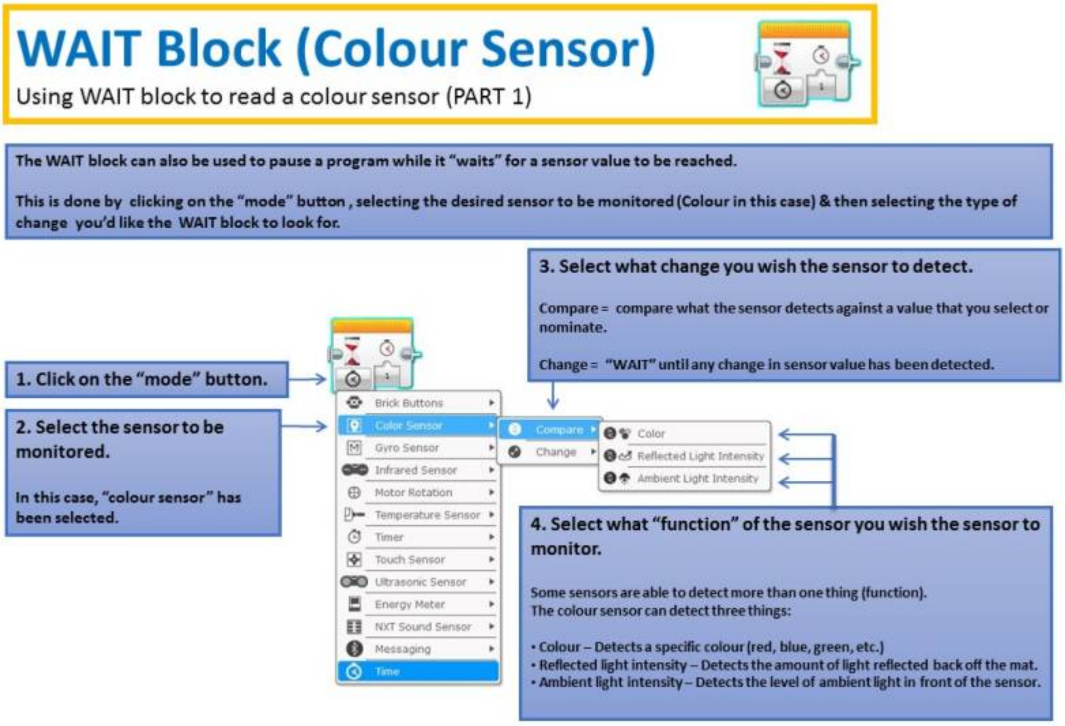
The first WAIT block in this sequence is being used to provide a delay at the start of the sequence to allow the person to get their hand clear of the button before the robot starts to move (highly recommended).

The second WAIT block is used to allow the medium motor to relax after having been powered ON for one second (see MEDIUM MOTOR block is set to TIME, with power setting of -5 & operation time of 1 second) .

This code is very useful for ensuring forks or any other attachments are fully down before raising them to a set position.

WAIT block set to TIMER function.

Time delay is set to 0.5 seconds. Use keyboard to enter desired time delay.



WAIT Block (Colour Sensor)

Using WAIT block to read a colour sensor (PART 1)

The WAIT block can also be used to pause a program while it "waits" for a sensor value to be reached.

This is done by clicking on the "mode" button , selecting the desired sensor to be monitored (Colour in this case) & then selecting the type of change you'd like the WAIT block to look for.

1. Click on the "mode" button.

2. Select the sensor to be monitored.

In this case, "colour sensor" has been selected.

3. Select what change you wish the sensor to detect.

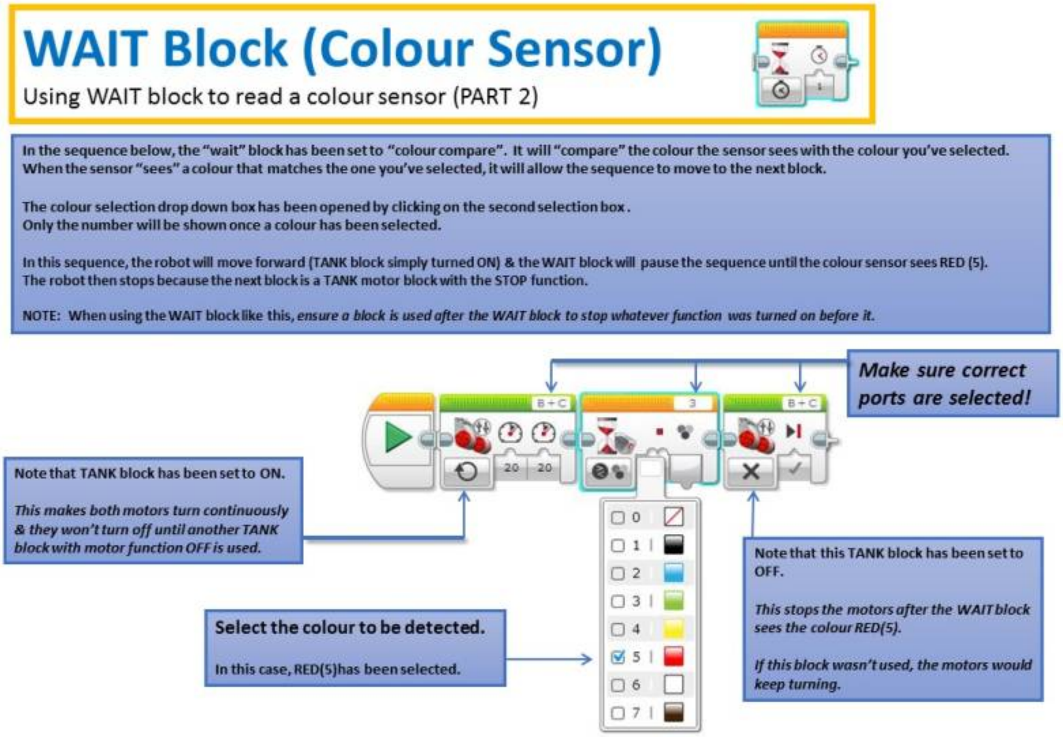
Compare = compare what the sensor detects against a value that you select or nominate.

Change = "WAIT" until any change in sensor value has been detected.

4. Select what "function" of the sensor you wish the sensor to monitor.

Some sensors are able to detect more than one thing (function).
The colour sensor can detect three things:

- Colour – Detects a specific colour (red, blue, green, etc.)
- Reflected light intensity – Detects the amount of light reflected back off the mat.
- Ambient light intensity – Detects the level of ambient light in front of the sensor.



WAIT Block (Colour Sensor)

Using WAIT block to read a colour sensor (PART 2)

In the sequence below, the "wait" block has been set to "colour compare". It will "compare" the colour the sensor sees with the colour you've selected. When the sensor "sees" a colour that matches the one you've selected, it will allow the sequence to move to the next block.

The colour selection drop down box has been opened by clicking on the second selection box . Only the number will be shown once a colour has been selected.

In this sequence, the robot will move forward (TANK block simply turned ON) & the WAIT block will pause the sequence until the colour sensor sees RED (5). The robot then stops because the next block is a TANK motor block with the STOP function.

NOTE: When using the WAIT block like this, *ensure a block is used after the WAIT block to stop whatever function was turned on before it.*

Make sure correct ports are selected!

Note that TANK block has been set to ON.

This makes both motors turn continuously & they won't turn off until another TANK block with motor function OFF is used.

Select the colour to be detected.

In this case, RED(5)has been selected.

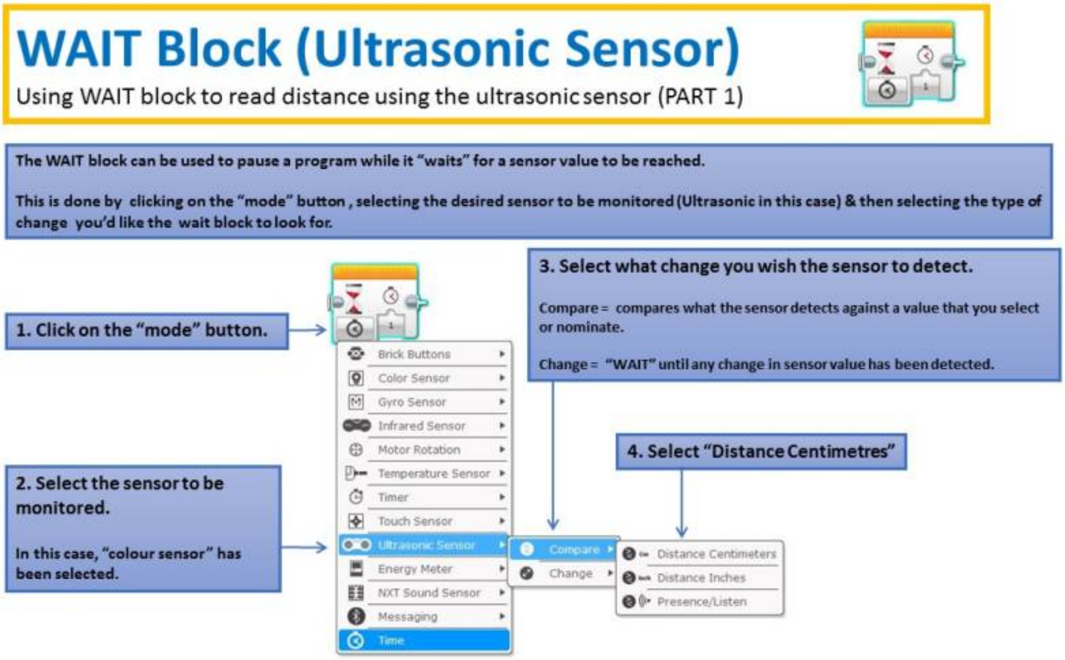
Note that this TANK block has been set to OFF.

This stops the motors after the WAIT block sees the colour RED(5).

If this block wasn't used, the motors would keep turning.

Colour Sensor Handy Hints

- Check sensor ports are correct by using the "PORT VIEW" in Mindstorms. This information will only be available or be updated when the robot is plugged into the computer.
- When the robot is connected to the computer, the "PORT VIEW" will show what the sensor is reading. This is handy to determine what colour the sensor is "seeing".
- No two COLOUR sensors will read exactly the same, so if using two at the same time, be prepared for discrepancies between the two.
- Colour sensors will detect colours on a "picture" or shaded background. This can cause false triggering when travelling over areas that are not distinct colours.



WAIT Block (Ultrasonic Sensor)

Using WAIT block to read distance using the ultrasonic sensor (PART 1)

The WAIT block can be used to pause a program while it "waits" for a sensor value to be reached.

This is done by clicking on the "mode" button , selecting the desired sensor to be monitored (Ultrasonic in this case) & then selecting the type of change you'd like the wait block to look for.

1. Click on the "mode" button.

2. Select the sensor to be monitored.

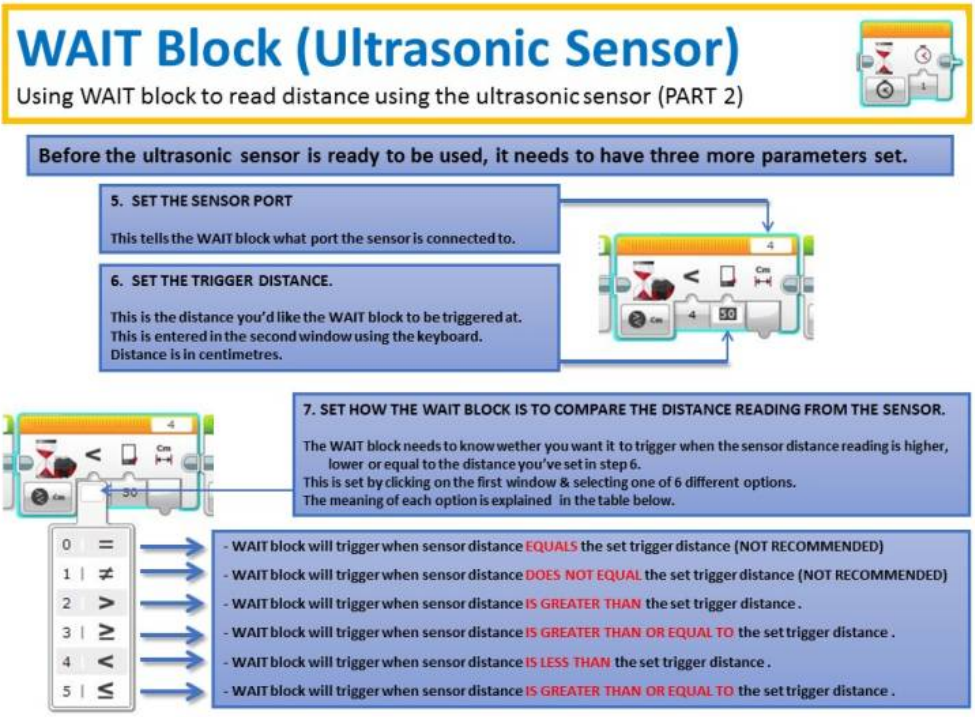
In this case, "colour sensor" has been selected.

3. Select what change you wish the sensor to detect.

Compare = compares what the sensor detects against a value that you select or nominate.

Change = "WAIT" until any change in sensor value has been detected.

4. Select "Distance Centimetres"



WAIT Block (Ultrasonic Sensor)

Using WAIT block to read distance using the ultrasonic sensor (PART 2)

Before the ultrasonic sensor is ready to be used, it needs to have three more parameters set.

5. SET THE SENSOR PORT

This tells the WAIT block what port the sensor is connected to.

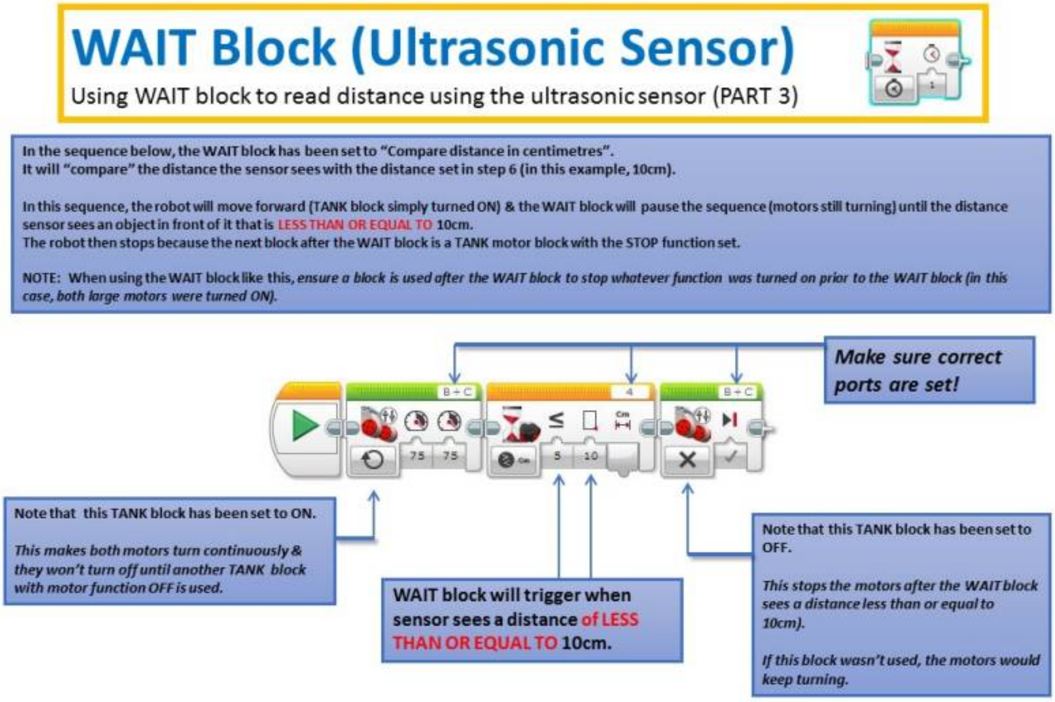
6. SET THE TRIGGER DISTANCE.

This is the distance you'd like the WAIT block to be triggered at. This is entered in the second window using the keyboard. Distance is in centimetres.

7. SET HOW THE WAIT BLOCK IS TO COMPARE THE DISTANCE READING FROM THE SENSOR.

The WAIT block needs to know wether you want it to trigger when the sensor distance reading is higher, lower or equal to the distance you've set in step 6.
This is set by clicking on the first window & selecting one of 6 different options.
The meaning of each option is explained in the table below.

0	=	- WAIT block will trigger when sensor distance EQUALS the set trigger distance (NOT RECOMMENDED)
1	≠	- WAIT block will trigger when sensor distance DOES NOT EQUAL the set trigger distance (NOT RECOMMENDED)
2	>	- WAIT block will trigger when sensor distance IS GREATER THAN the set trigger distance .
3	≥	- WAIT block will trigger when sensor distance IS GREATER THAN OR EQUAL TO the set trigger distance .
4	<	- WAIT block will trigger when sensor distance IS LESS THAN the set trigger distance .
5	≤	- WAIT block will trigger when sensor distance IS GREATER THAN OR EQUAL TO the set trigger distance .



WAIT Block (Ultrasonic Sensor)

Using WAIT block to read distance using the ultrasonic sensor (PART 3)

In the sequence below, the WAIT block has been set to "Compare distance in centimetres".
It will "compare" the distance the sensor sees with the distance set in step 6 (in this example, 10cm).

In this sequence, the robot will move forward (TANK block simply turned ON) & the WAIT block will pause the sequence (motors still turning) until the distance sensor sees an object in front of it that is LESS THAN OR EQUAL TO 10cm.
The robot then stops because the next block after the WAIT block is a TANK motor block with the STOP function set.

NOTE: When using the WAIT block like this, *ensure a block is used after the WAIT block to stop whatever function was turned on prior to the WAIT block (in this case, both large motors were turned ON).*

Make sure correct ports are set!

Note that this TANK block has been set to ON.

This makes both motors turn continuously & they won't turn off until another TANK block with motor function OFF is used.

WAIT block will trigger when sensor sees a distance of LESS THAN OR EQUAL TO 10cm.

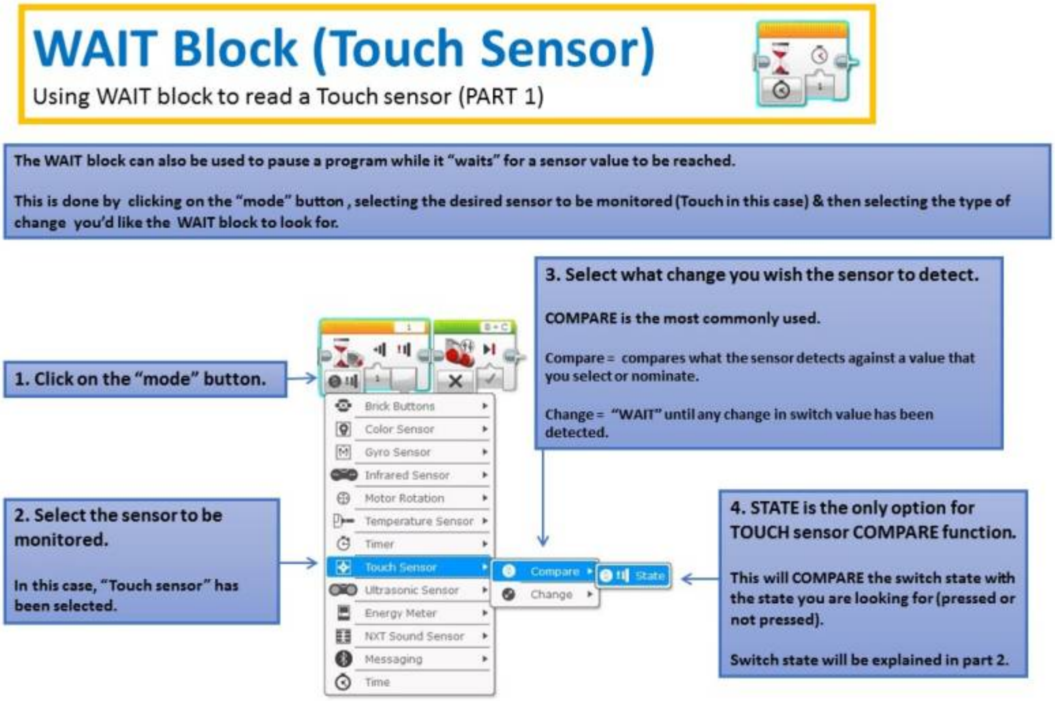
Note that this TANK block has been set to OFF.

This stops the motors after the WAIT block sees a distance less than or equal to 10cm).

If this block wasn't used, the motors would keep turning.

Ultrasonic Sensor Handy Hints

- Check sensor ports are correct by using the "PORT VIEW" in Mindstorms. This information will only be available or be updated when the robot is plugged into the computer.
- When the robot is connected to the computer, the "PORT VIEW" will also show what the sensor is reading. This is handy to determine what distance the sensor is "seeing".
- The Ultrasonic sensor will struggle to read reliably if the surface in front of it is too small or is sharply angled. This is because the ultrasonic sound wave will bounce off at an angle rather than being reflected back to the sensor.
- Using "greater than or equal to" or "less than or equal to" is highly recommended as it means the wait block will react even if a count of two or more is skipped between sensor readings.
- When using the ultrasonic sensor, moving the robot slowly will result in better accuracy.



WAIT Block (Touch Sensor)

Using WAIT block to read a Touch sensor (PART 1)

The WAIT block can also be used to pause a program while it "waits" for a sensor value to be reached.

This is done by clicking on the "mode" button , selecting the desired sensor to be monitored (Touch in this case) & then selecting the type of change you'd like the WAIT block to look for.

1. Click on the "mode" button.

2. Select the sensor to be monitored.

In this case, "Touch sensor" has been selected.

3. Select what change you wish the sensor to detect.

COMPARE is the most commonly used.

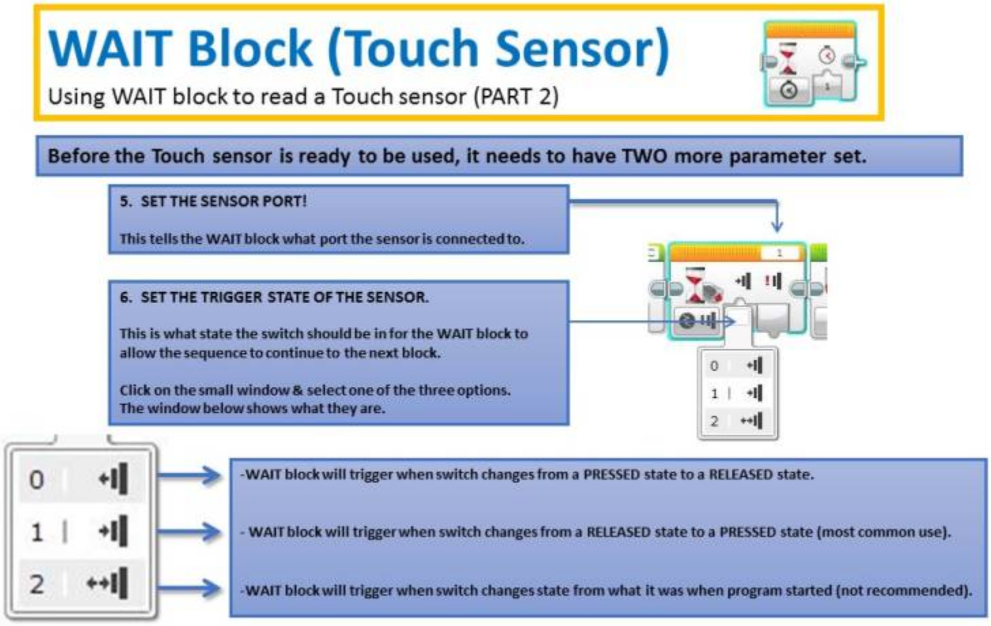
Compare = compares what the sensor detects against a value that you select or nominate.

Change = "WAIT" until any change in switch value has been detected.

4. STATE is the only option for TOUCH sensor COMPARE function.

This will COMPARE the switch state with the state you are looking for (pressed or not pressed).

Switch state will be explained in part 2.



WAIT Block (Touch Sensor)

Using WAIT block to read a Touch sensor (PART 2)

Before the Touch sensor is ready to be used, it needs to have TWO more parameter set.

5. SET THE SENSOR PORT!

This tells the WAIT block what port the sensor is connected to.

6. SET THE TRIGGER STATE OF THE SENSOR.

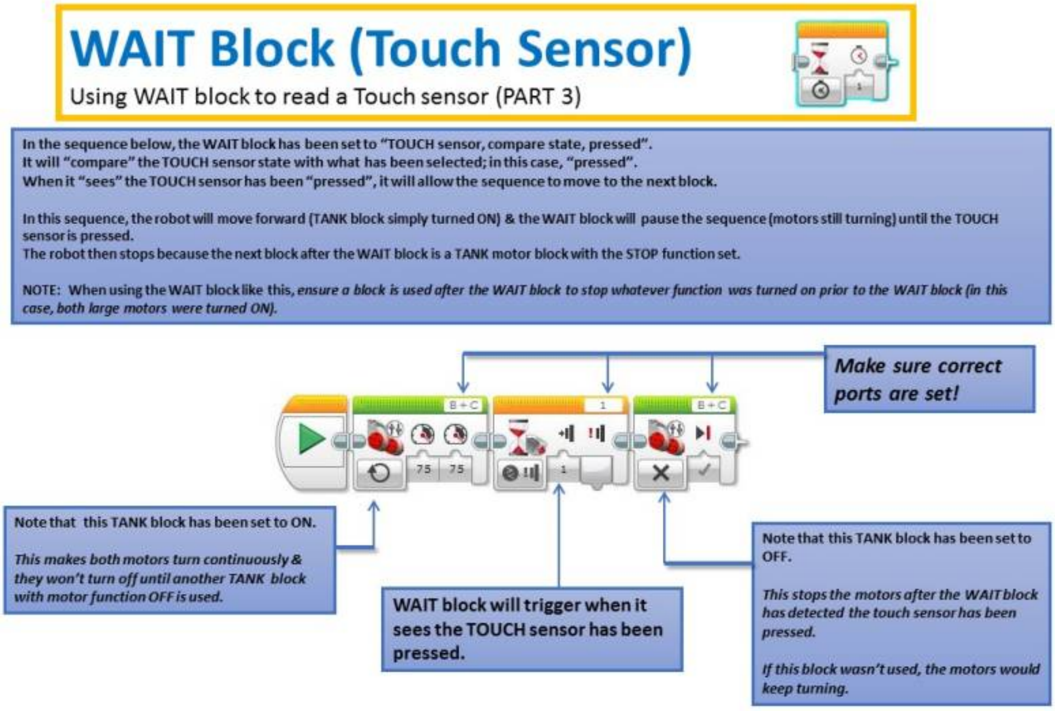
This is what state the switch should be in for the WAIT block to allow the sequence to continue to the next block.

Click on the small window & select one of the three options. The window below shows what they are.

-WAIT block will trigger when switch changes from a PRESSED state to a RELEASED state.

- WAIT block will trigger when switch changes from a RELEASED state to a PRESSED state (most common use).

-WAIT block will trigger when switch changes state from what it was when program started (not recommended).



WAIT Block (Touch Sensor)

Using WAIT block to read a Touch sensor (PART 3)

In the sequence below, the WAIT block has been set to "TOUCH sensor, compare state, pressed". It will "compare" the TOUCH sensor state with what has been selected; in this case, "pressed". When it "sees" the TOUCH sensor has been "pressed", it will allow the sequence to move to the next block.

In this sequence, the robot will move forward (TANK block simply turned ON) & the WAIT block will pause the sequence (motors still turning) until the TOUCH sensor is pressed. The robot then stops because the next block after the WAIT block is a TANK motor block with the STOP function set.

NOTE: When using the WAIT block like this, *ensure a block is used after the WAIT block to stop whatever function was turned on prior to the WAIT block (in this case, both large motors were turned ON).*

Make sure correct ports are set!

Note that this TANK block has been set to ON.

This makes both motors turn continuously & they won't turn off until another TANK block with motor function OFF is used.

WAIT block will trigger when it sees the TOUCH sensor has been pressed.

Note that this TANK block has been set to OFF.

This stops the motors after the WAIT block has detected the touch sensor has been pressed.

If this block wasn't used, the motors would keep turning.

Touch Sensor Handy Hints

- Check sensor ports are correct by using the "PORT VIEW" in Mindstorms. This information will only be available or be updated when the robot is plugged into the computer.
- When the robot is connected to the computer, the "PORT VIEW" will also show what the sensor is reading. This is handy to determine what state the sensor is in (pressed or released).
- The TOUCH sensors do require some pressure to activate. Make sure the frame being used to hold the sensor is sturdy enough to not "flex" when pressure is applied. Too much flex will provide unreliable actuation of the sensor.
- Because pressure needs to be applied to activate the sensor, ensure that the robot can apply enough pressure without spinning the wheels or being "skewed" off at an angle.

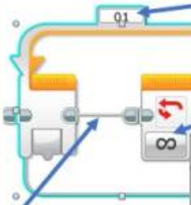
Single Loop Block

Used to perform a task until a condition is met: Until something happens do this – once condition is met end loop.

The loop block can condense your code by performing two tasks simultaneously. – If – Then
One, it will perform the tasks inside the loop until the loop condition is met.
Two, the EV3 will try to complete the set condition of the loop.

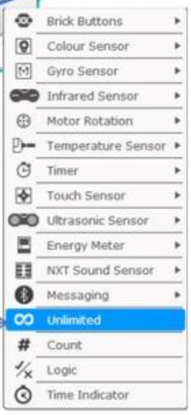
Conditions can include a simple count down or encompass a sensor.

1. Click on the "mode" button to open the loop condition options.

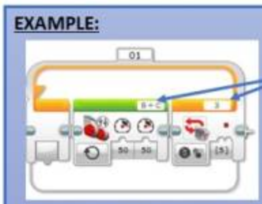


Loop Name: Make sure your loops are easily identifiable by numbering them in order.

"Mode" Button



EXAMPLE:



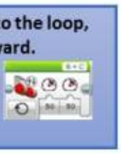
Make sure to check your motor and sensor inputs are correct!

In this example the loop will:

1. Drive straight forward until the loop condition is fulfilled;
2. The loop condition is: Colour sensor has to see the colour "RED", once this condition is fulfilled the EV3 will stop driving forward and the program will go to the next coding block.

3. Insert a task into the loop, such as drive forward.

The loop grows with Each block you put inside it.



2. Select the condition on which the loop will end, once it has been full filled.

In this case, "unlimited" has been selected, the default condition for the loop block.

Double Loop Block

Used to perform a task until a condition is met and once the condition is met perform a different task

The double loop block can condense your code by performing two tasks simultaneously and prepare a third. – If – Then – Else

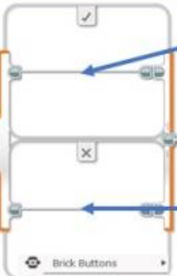
1. **It will perform the tasks inside the crossed out area of the loop, until the condition of the loop is fulfilled.** ☒
2. **The EV3 will try to complete the set condition of the loop, selected under "mode".**
3. **Once the loop condition has been fulfilled the EV3 will perform the task within the ticked area.** ✓

Conditions can include a simple count down or encompass a sensor.

Click on the "mode" button to open the loop condition options.

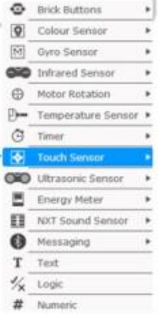
Selected Port

To select a loop condition select the "mode" button".



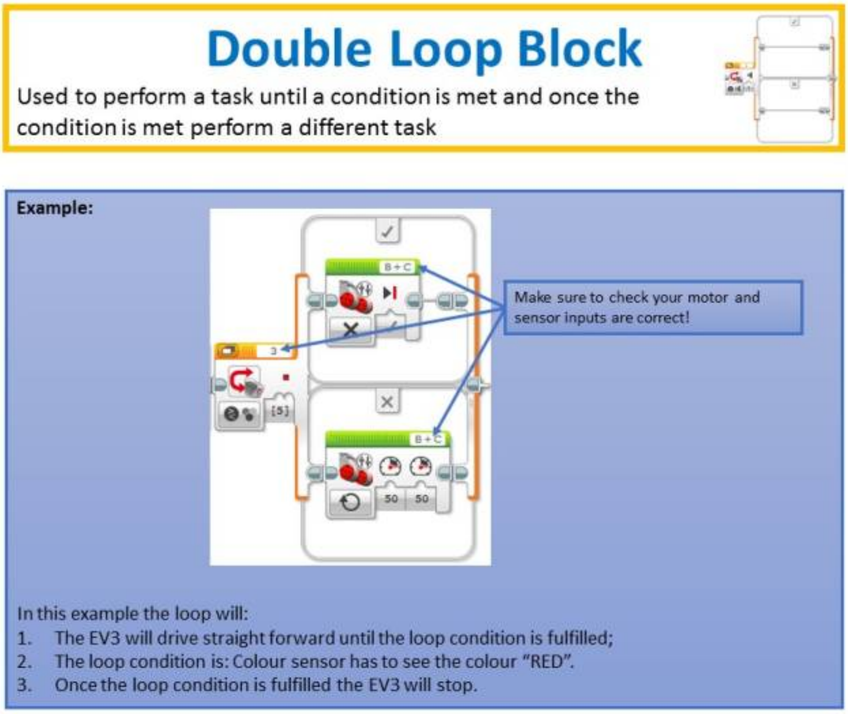
Loop condition is fulfilled: The task in this area will not be performed until the loop condition is fulfilled

Loop condition is not fulfilled: The task in this area will be performed while the loop condition is not fulfilled



Select the condition on which the loop will go from the non ticked area to the ticked area = the loop condition is fulfilled.

In this case, the "touch sensor" is selected, 1 means the touch sensor has to be pressed to fulfil the condition of the loop.



Double Loop Block

Used to perform a task until a condition is met and once the condition is met perform a different task

Example:

Make sure to check your motor and sensor inputs are correct!

In this example the loop will:

1. The EV3 will drive straight forward until the loop condition is fulfilled;
2. The loop condition is: Colour sensor has to see the colour "RED".
3. Once the loop condition is fulfilled the EV3 will stop.

Mentor Training

The following will be discussed during the mentor training.

1. Naming conventions and folder structure
2. Introduction to the coding blocks by the cheat sheets - with robot
    1. Coding with the cheat sheets and a EV3 robot
    2. Play Button
    3. Motor Blocks
        1. Gear train
        2. Medium motor gearing
    4. Time Delay
        1. Setting ground zero for lifting operations
        2. Half-second delay after start
    5. Touch Sensor
    6. Colour Sensor
    7. Ultrasonic Sensor
    8. Control Panel
    9. Loop Blocks
3. Connection Panel
    1. Connecting Robot
    2. Port view
    3. Memory & Settings view
    4. Don't Press Update on the robot during a session - it will take a very long time.
4. How to program the robot - USB only
5. Mat
    1. Positioning

    2. Cutting of the mat
6. Homework options (self paced introduction at home)
    1. knowing the blocks
    2. how to set ports correctly
    3. how to read a block
    4. first problem solving experience in Mindstorms
7. Core Values and how to get the participants to find solutions themselves:
    1. Have the team member read the code back to you
    2. Have the team member write out what the robot is supposed to achieve

Training Resources

- Mentor Training Print Out
- Challenge guide with point system
- Field set up guide
- Core Values
- 10 Week Program
- Score Sheet
- Challenge Updates
- Robot Game
- Rotations Overview

Back to Lego Robotics