MakerHouse: Empowering Makers

micro:bit

DC Motor Control

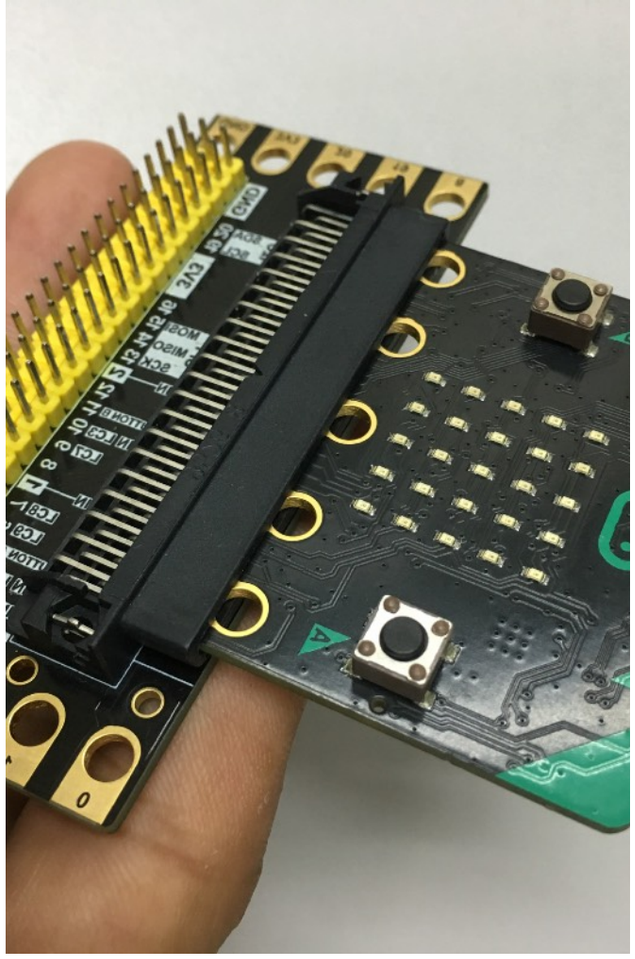
Things used in this project:

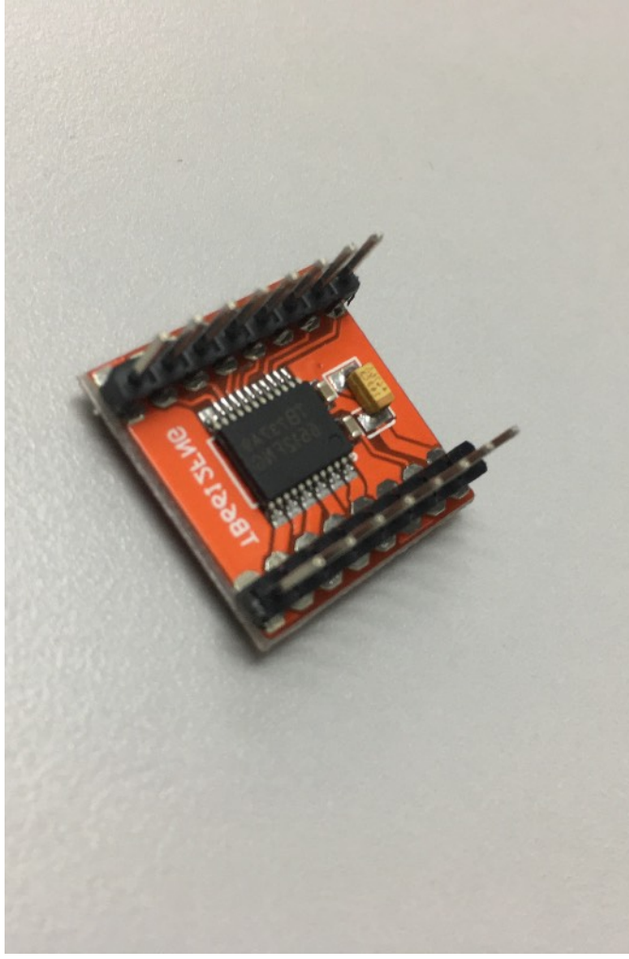
Hardware

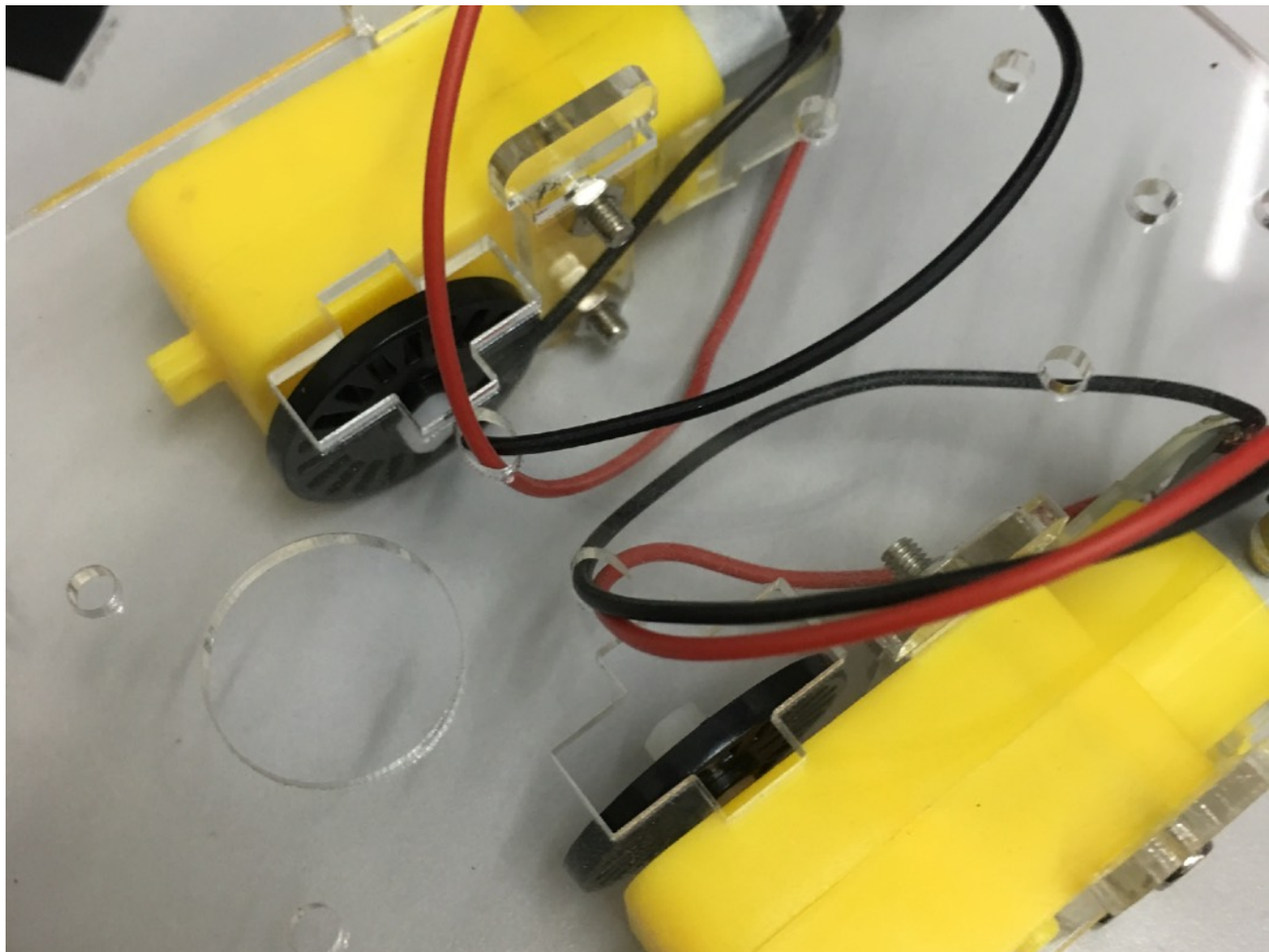
- micro:bit
- Edge breakout for micro:bit, I/O expansion
- Smart robot car chassis kit with DC motor set
- Dual motor driver controller module — TB6612FNG
- Breadboard
- Jumper wires

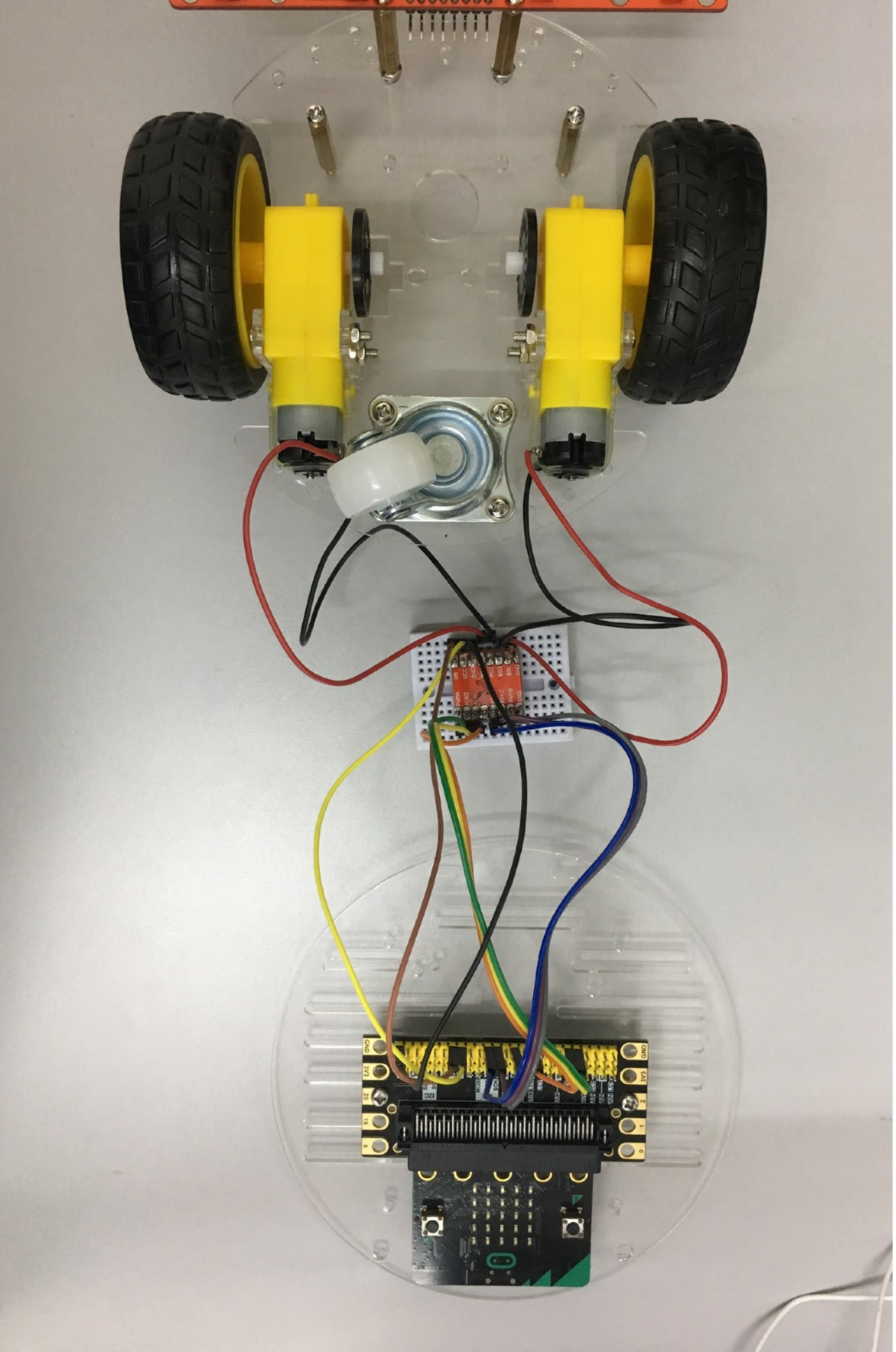
Software

- Microsoft MakeCode









Objectives

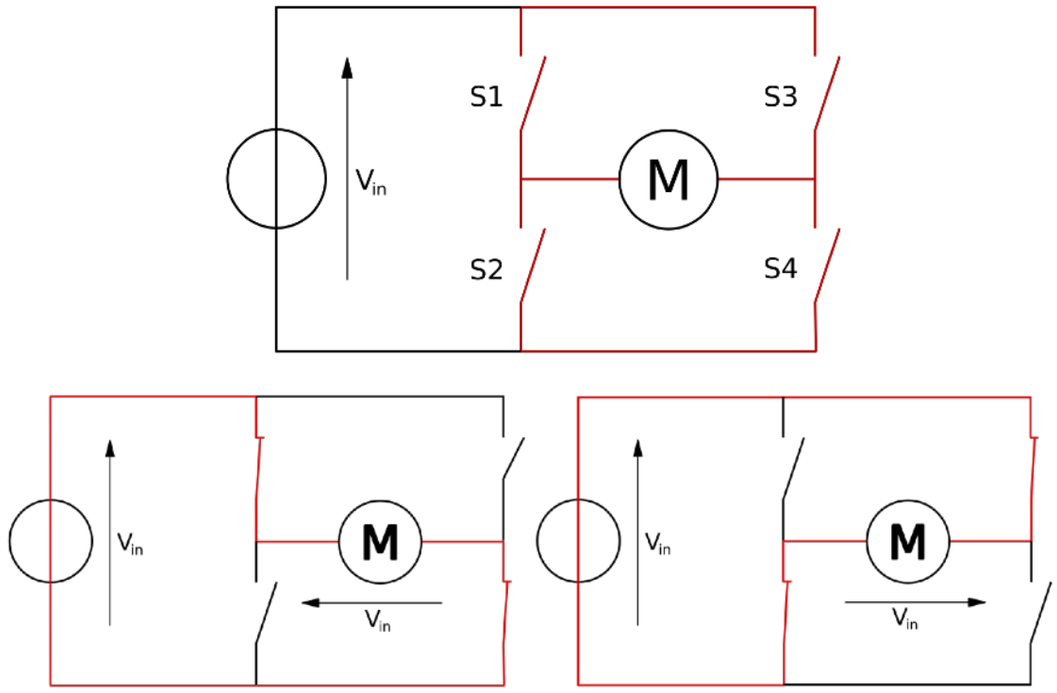
- Understand about DC motor control — speed and direction
- Using micro:bit to control the DC motor with the help of the motor driver

DC Motor Control

Motor is a kind of device, which can transfer electric energy into kinetic energy according to the law in electromagnetic induction. We will be using inexpensive hobby motors commonly used in educational robotics. These particular motors require about 3 volts to overcome the mechanical resistance and start turning. In this experiment, we are going to use a microbit and motor driver to control the direction and speed of a motor.

Direction

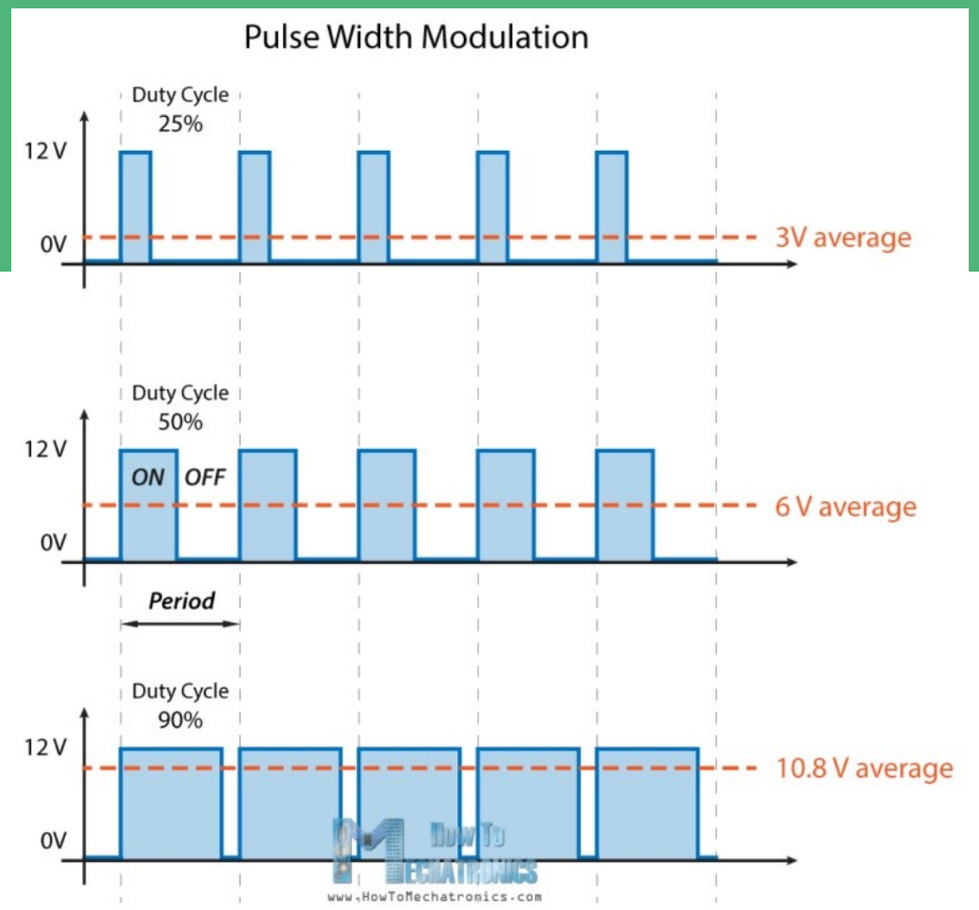
The H-bridge arrangement is generally used to reverse the polarity/direction of the motor, but can also be used to 'brake' the motor, where the motor comes to a sudden stop, as the motor's terminals are shorted, or to let the motor 'free run' to a stop, as the motor is effectively disconnected from the circuit. The following table summarises operation, with S1-S4 corresponding to the diagram above.



Operation				
S1	S2	S3	S4	Result
1	0	0	1	Motor moves right
0	1	1	0	Motor moves left
0	0	0	0	Motor coasts
1	0	0	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	
0	1	0	1	Motor brakes
1	0	1	0	
x	x	1	1	Short Circuit
1	1	x	x	

Speed

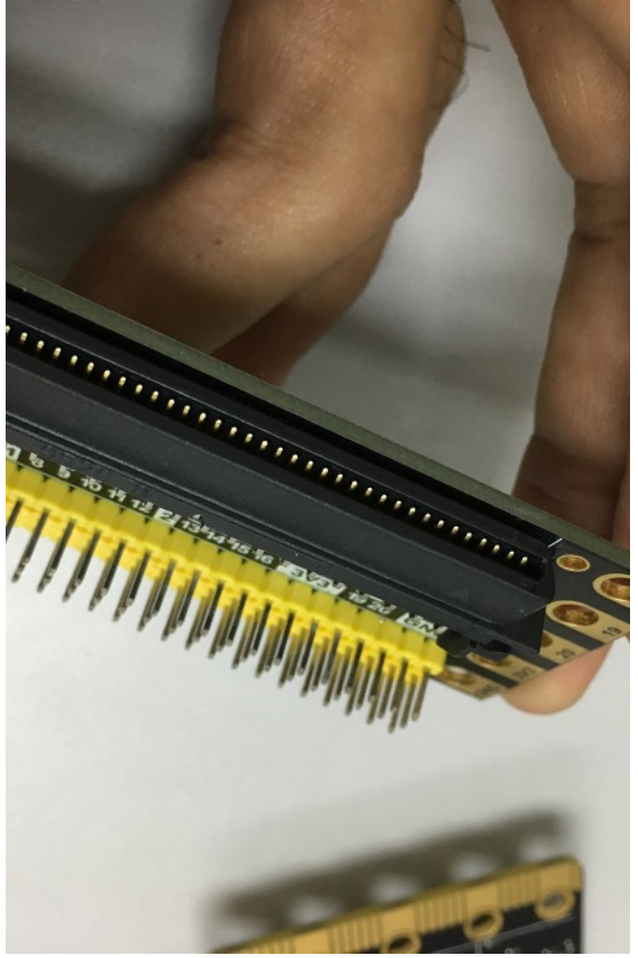
We can control the speed of the DC motor by simply controlling the input voltage to the motor and the most common method of doing that is by using PWM signal. The average value of current supplied to the load is controlled by the switch position and duration of its state.

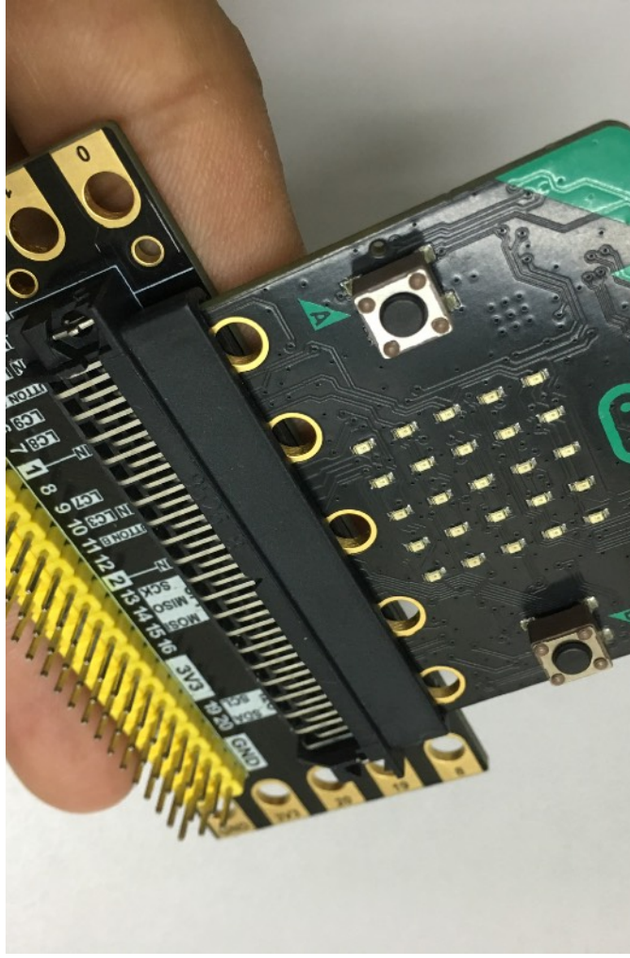


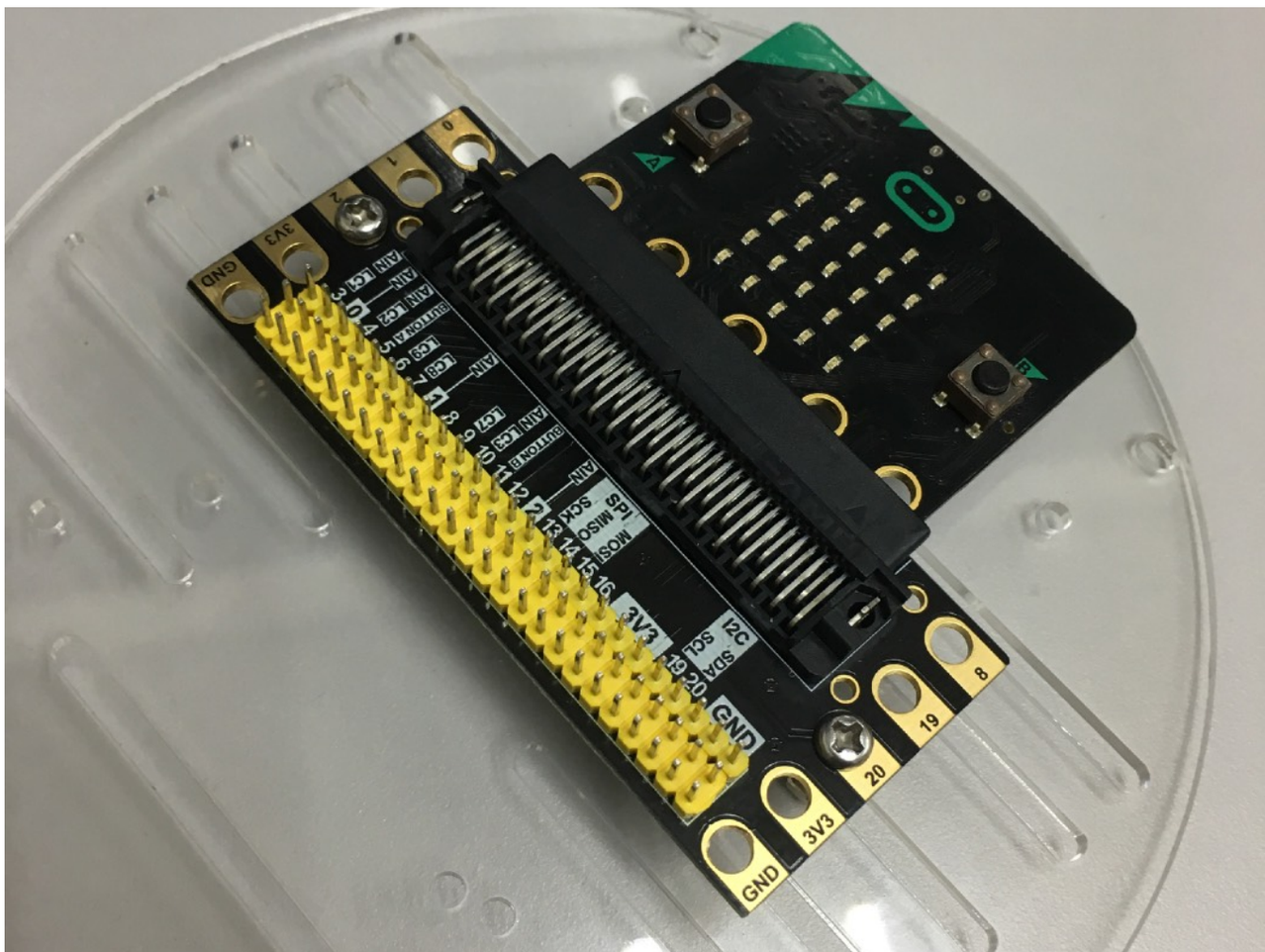
Assembly

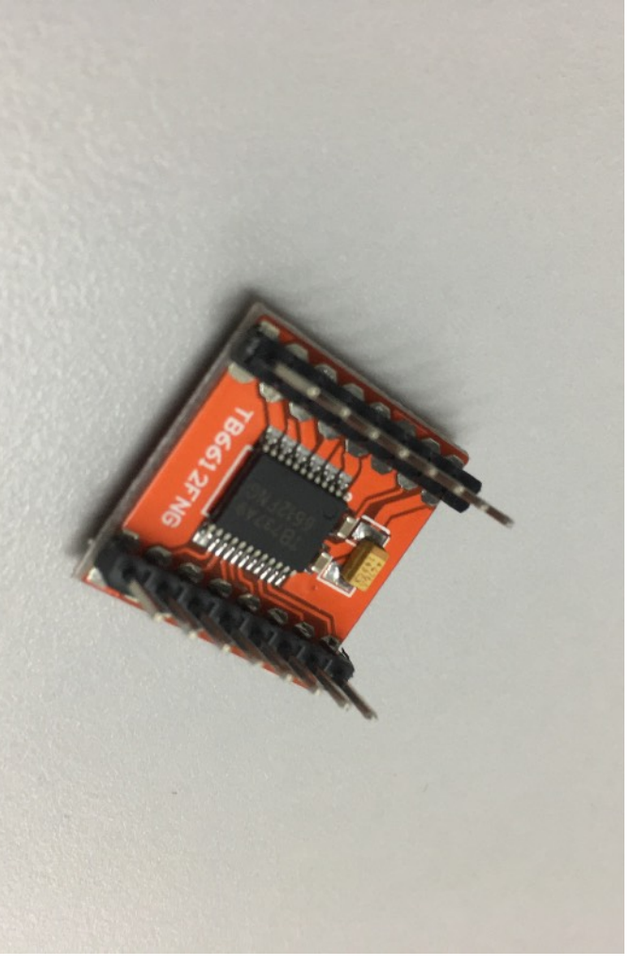
STEP 1: Slot in the micro:bit into the I/O expansion board.

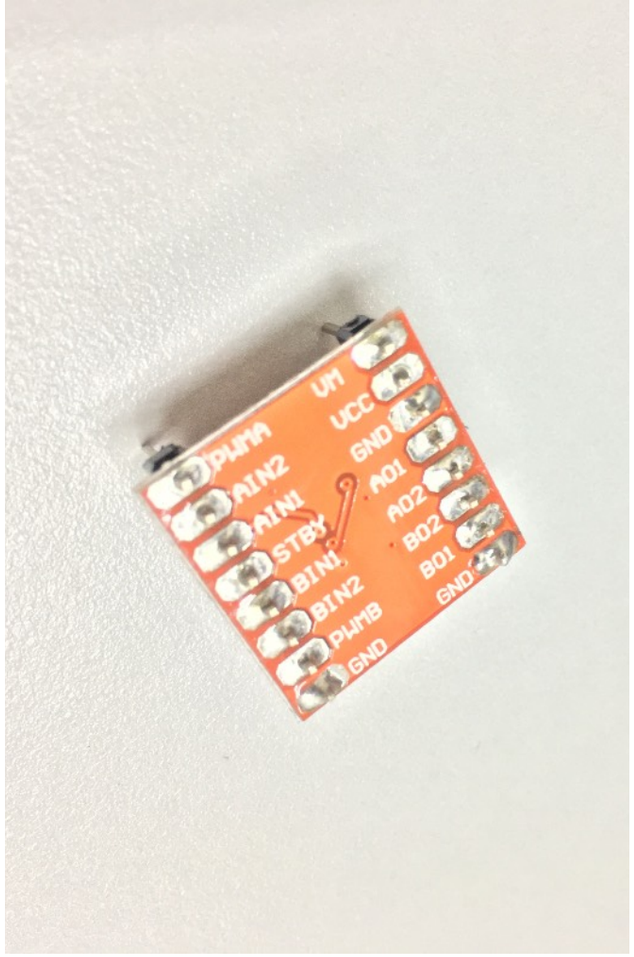
STEP 2: Attached and screw the expansion board onto the first layer of the robot chassis.

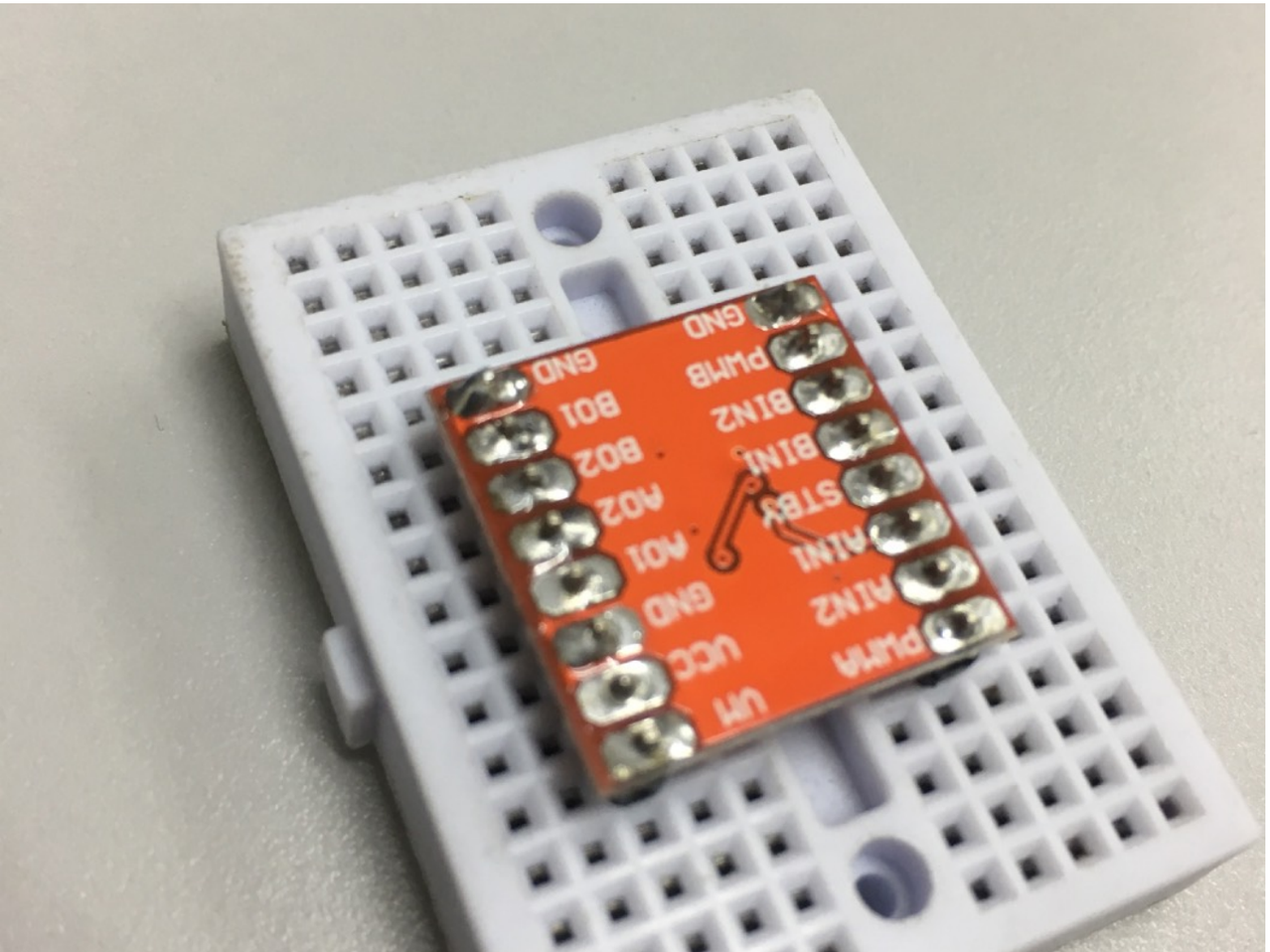








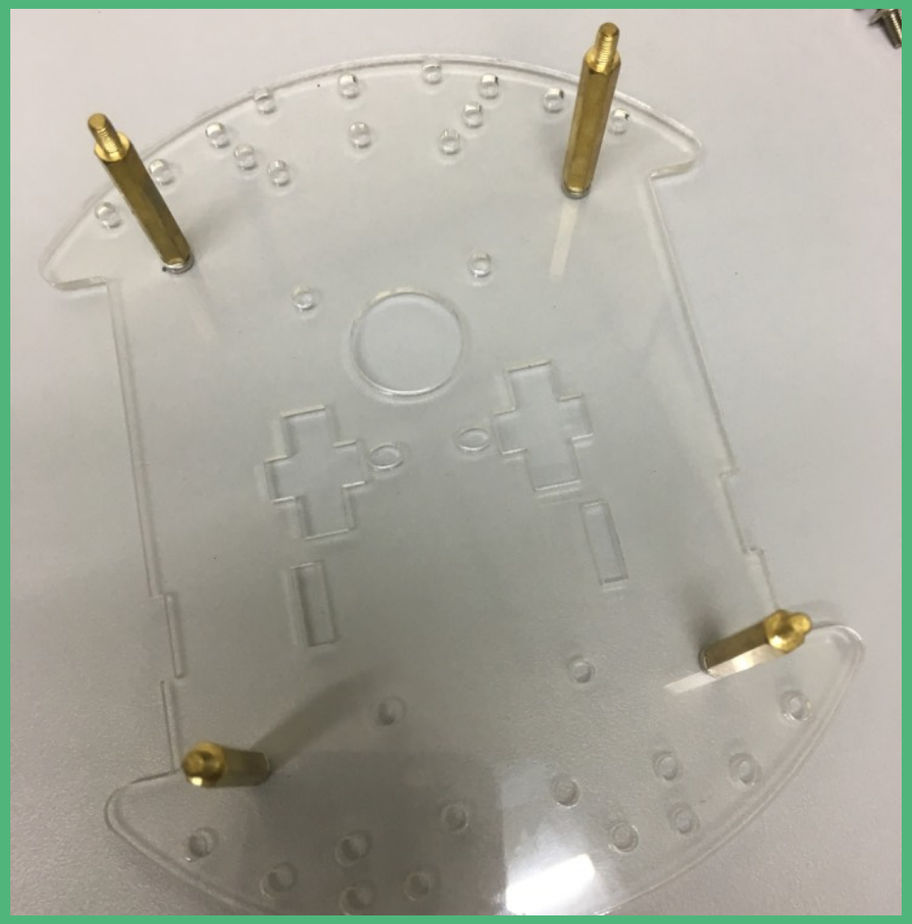




Assembly (cont.)

STEP 3: Soldering the male I/O pins to the motor driver.

STEP 4: Attached the completed motor driver onto the breadboard.



WARNING! ***The 30 mm PCB brass stand need to attached and screw first, otherwise problem may arise during dc motor assembly.***

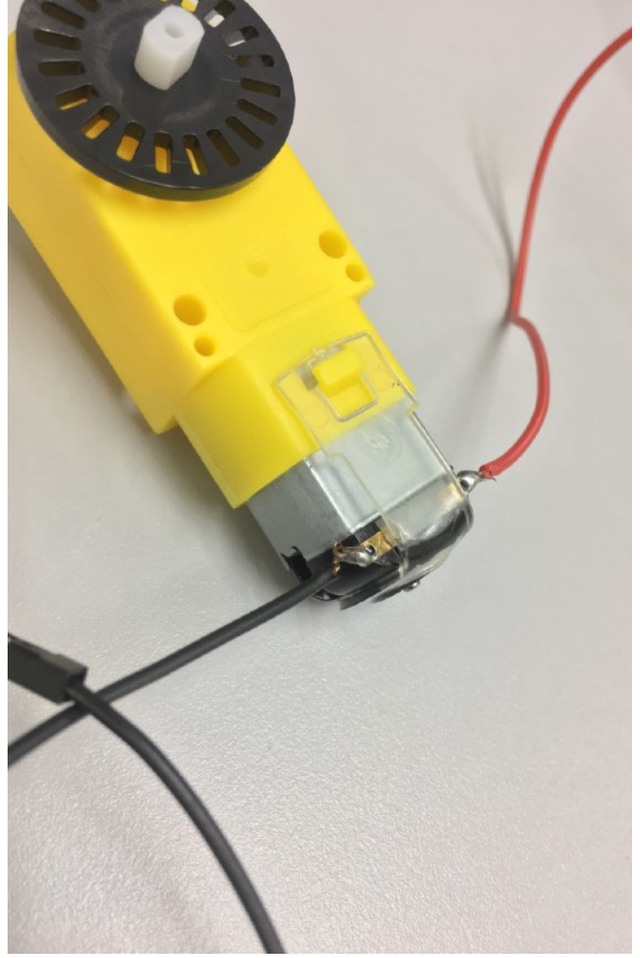
Assembly (cont.)

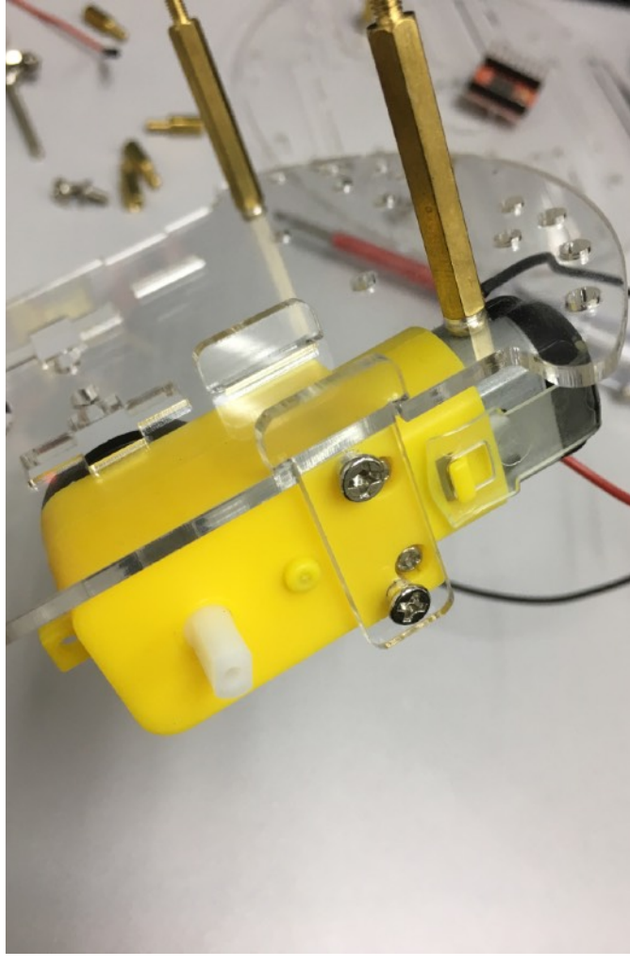
STEP 5: Prepare the bottom layer chassis, screw the 30mm PCB stand.

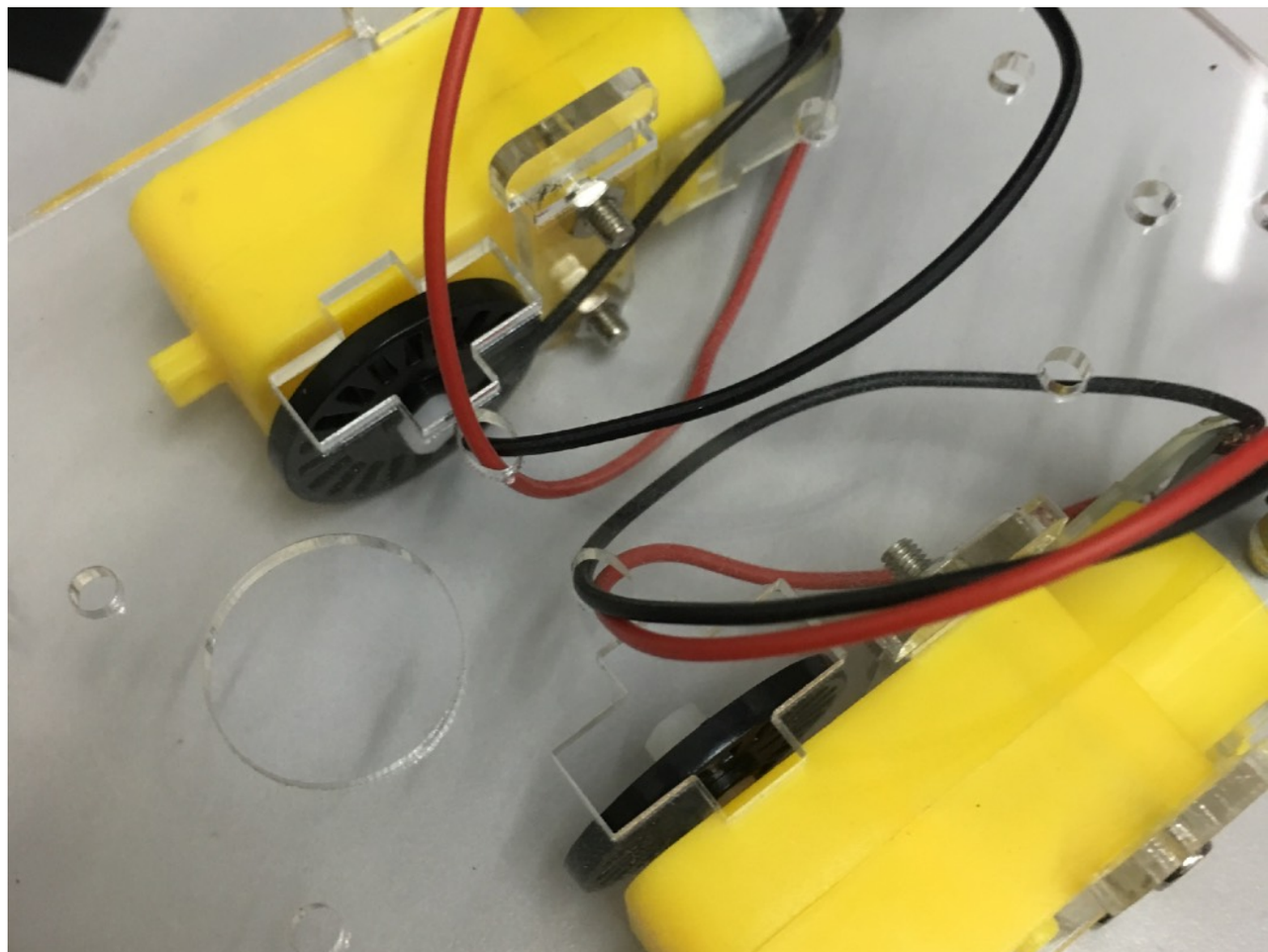
Assembly (cont.)

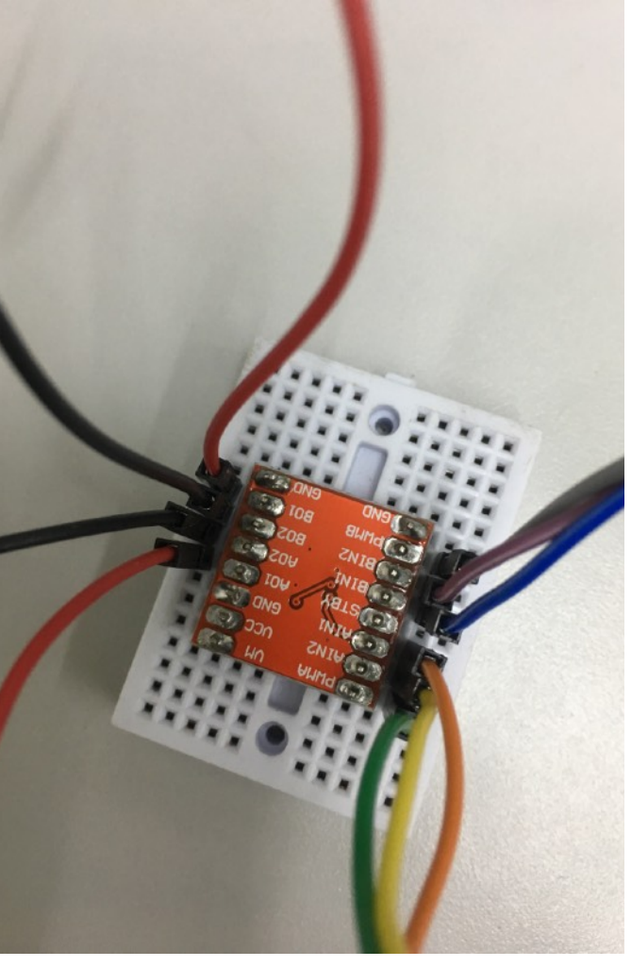
STEP 6: Solder the jumper wires on the dc motor.

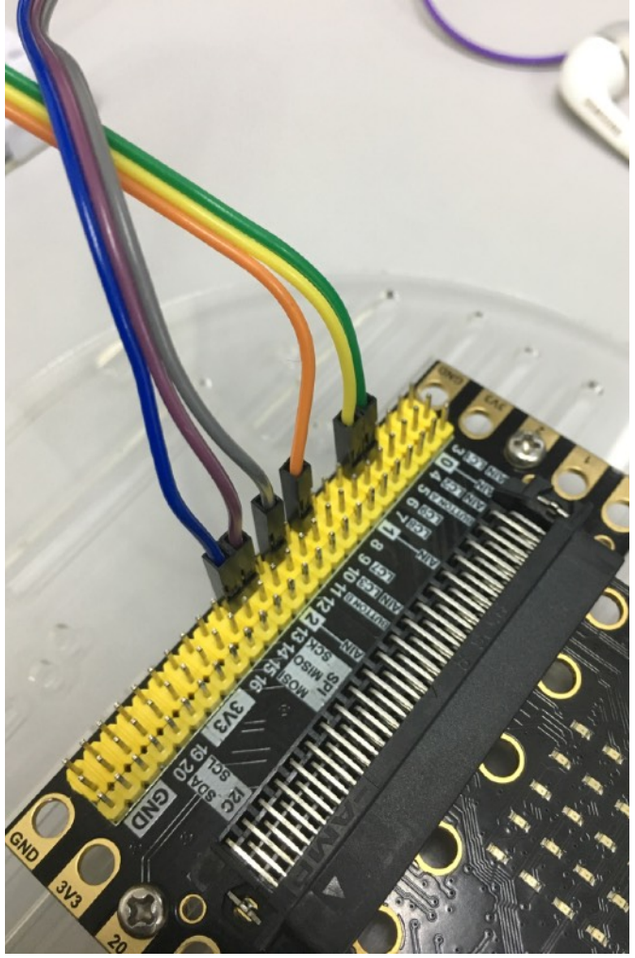
STEP 7: Attached and screw the dc motor onto the bottom layer of the robot chassis.

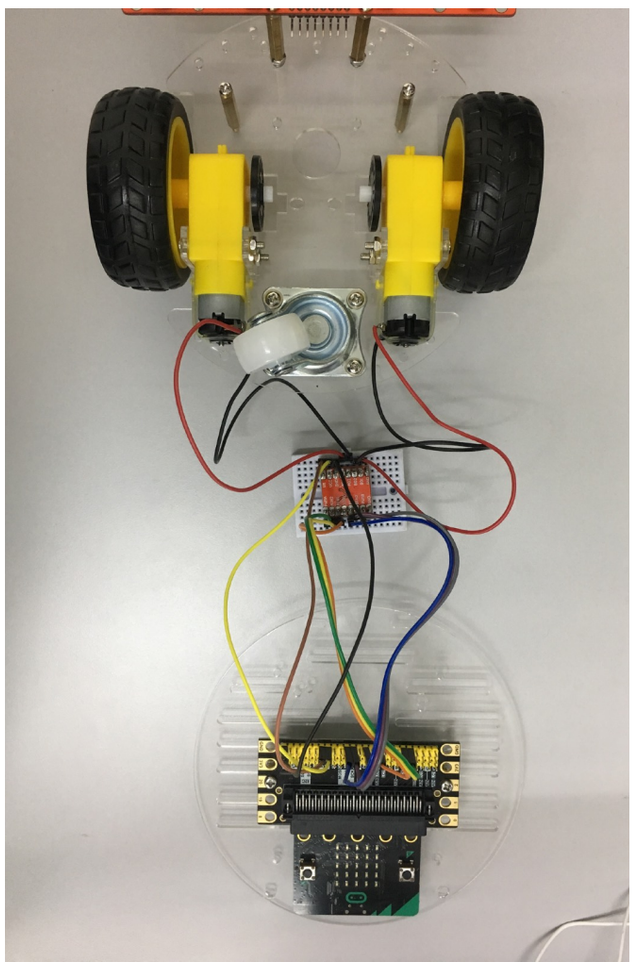












Assembly (cont.)

STEP 8: Finished the assembly by connecting the bottom and top layer of the chassis

STEP 9: Do the wiring, from the motor to the driver and from the driver to the micro:bit.

Pin Name	To/From
VM	3V3
VCC	3V3
GND	GND
A01	DC Motor
A02	DC Motor
B02	DC Motor
B01	DC Motor
GND	GND
PWMA	Pin 6
AIN2	Pin 7
AIN1	Pin 10
STBY	3V3
BIN1	Pin 14
BIN2	Pin 13
PWMB	Pin 12
GND	GND

Flashing Program

Initialization

Initialization — on start

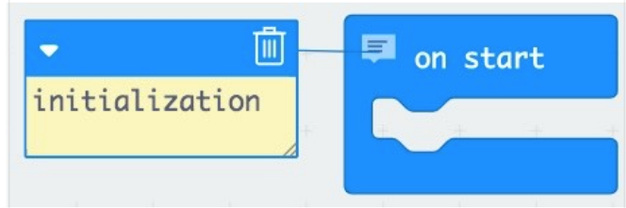
STEP 1: From **Basic** select **on start**.

STEP 2: Disable display (LED) — to free up some pins. Click **Led** > **...more**, select **led enable false**.

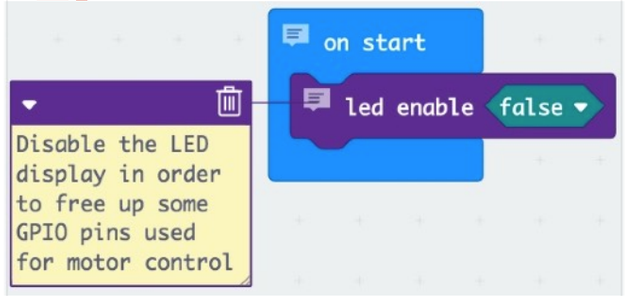
STEP 3: Set the motor condition as **STOP**. Table below as reference.

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

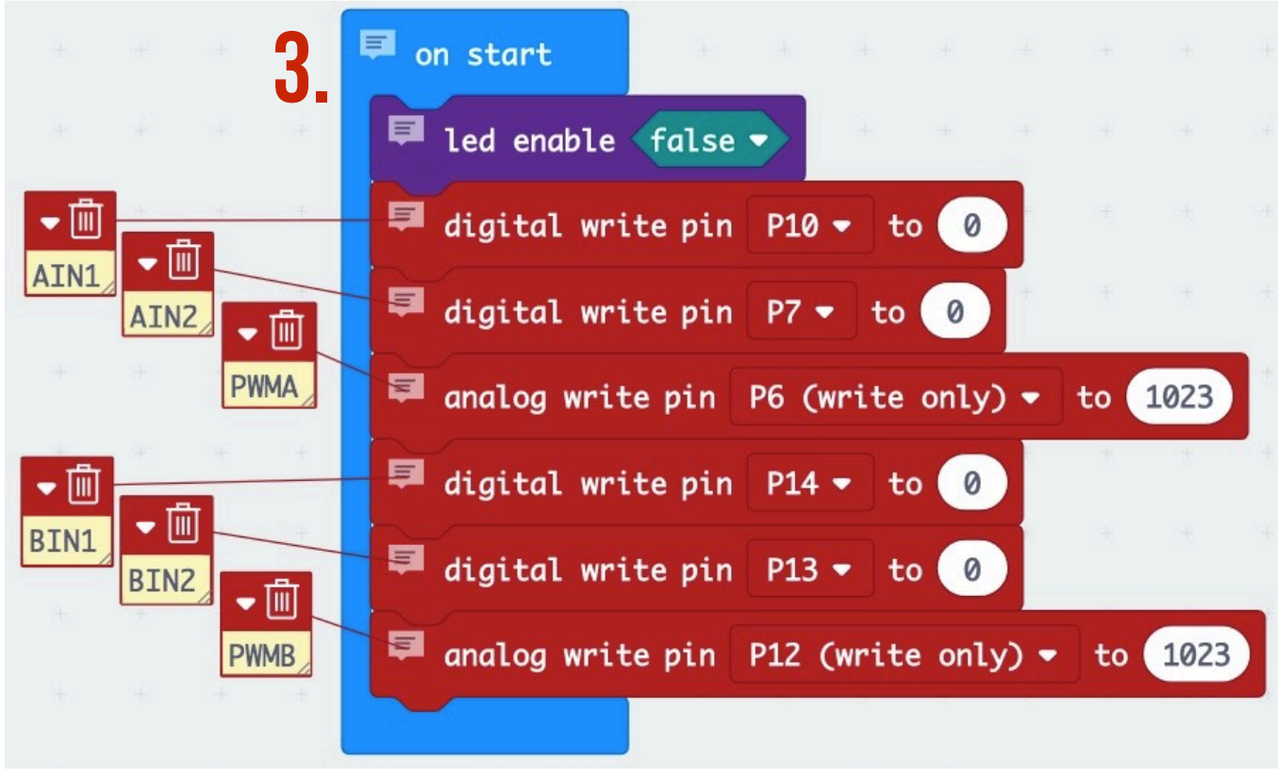
1.



2.



3.



Flashing Program

Repeating

loops — forever

STEP 4: From **Basic** select **forever**.

STEP 5: Inside **Logic** select **if…else**.

STEP 6: Make a condition **if** button A is pressed so that the motor move **else** stop.

STEP 7: Download!

STEP 8: Test the code — pressed button A.

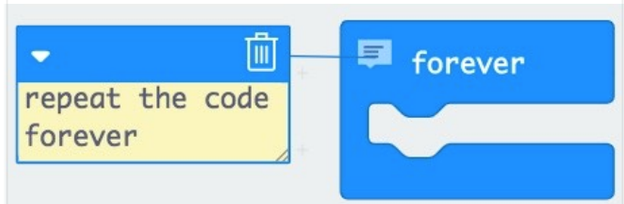
**** Observe the movement and the speed**

Challenge

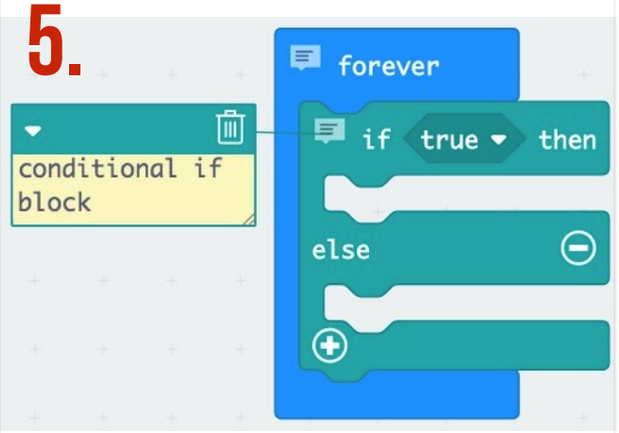
Different direction, Different speed

Make some changes — may add if button B pressed? change different movement? the rotation speed

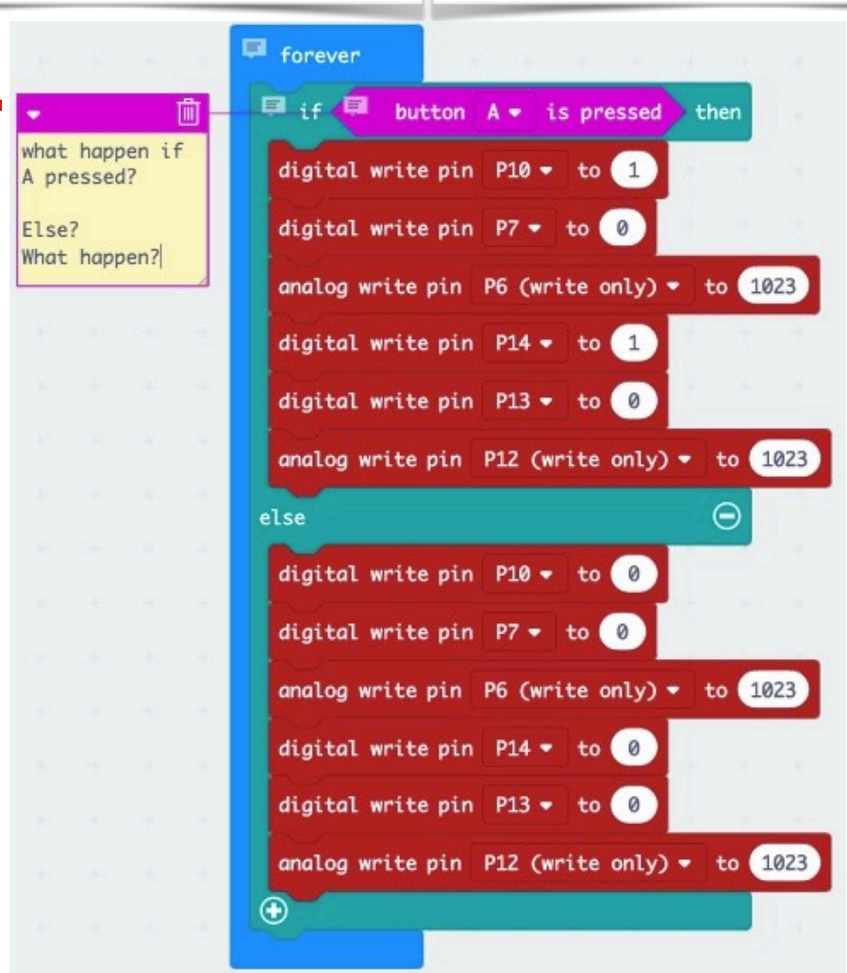
4.



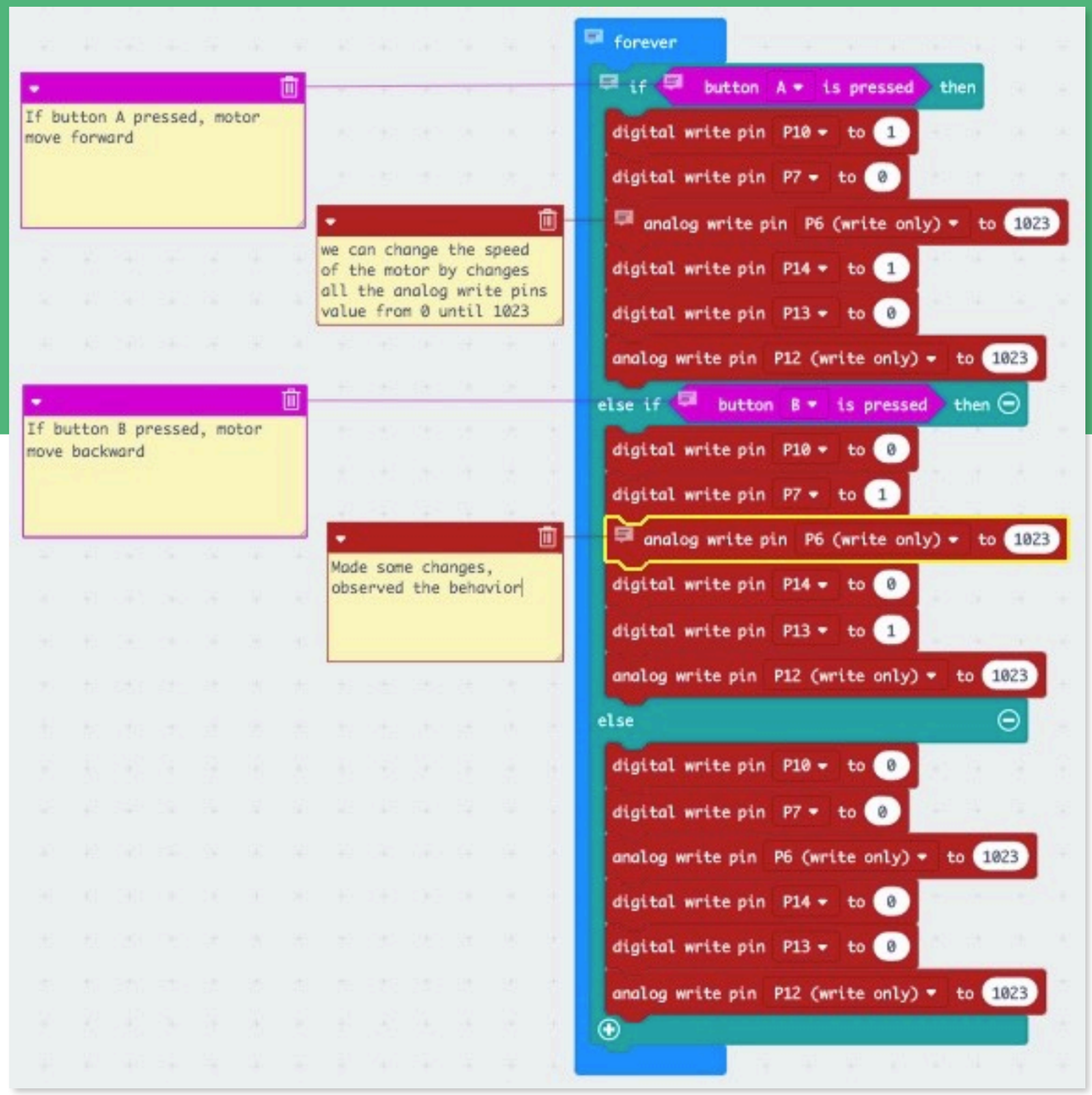
5.



6.



Hint!


If button A pressed, motor move forward
we can change the speed of the motor by changes all the analog write pins value from 0 until 1023
If button B pressed, motor move backward
Made some changes, observed the behavior
forever
if button A is pressed then
digital write pin P10 to 1
digital write pin P7 to 0
analog write pin P6 (write only) to 1023
digital write pin P14 to 1
digital write pin P13 to 0
analog write pin P12 (write only) to 1023
else if button B is pressed then
digital write pin P10 to 0
digital write pin P7 to 1
analog write pin P6 (write only) to 1023
digital write pin P14 to 0
digital write pin P13 to 1
analog write pin P12 (write only) to 1023
else
digital write pin P10 to 0
digital write pin P7 to 0
analog write pin P6 (write only) to 1023
digital write pin P14 to 0
digital write pin P13 to 0
analog write pin P12 (write only) to 1023