STEPS TO OPERATE

1. Switch on power supply of robot.
2. Start controller button on control box of robot.
3. Connect Arduino’s and power them
4. Start keyboard_teleop_code (tal_brabo/keyboard_teleop.py)
5. Confirm all joints are moving as expected by manually moving them using keyboard.
6. Open tal_brabo_driver node (tal_brabo/tal_driver.py) and **comment out the homing part**.
7. Start tal_brabo_driver node (tal_brabo/tal_driver.py) (MAX_PULSES < 60)
8. Now you can start other nodes
9. Attach camera to end effector
10. Start usb_cam_node
11. Start object_tracker node (tal_brabo/object_tracker.py)
12. Start Rviz (brabo_description/launch/display.launch)
13. Start velocity controller node (tal_brabo/velocity_controller.py)

For Keyboard the key-presses are: X and Right Arrow Key (→), X and Left Arrow Key (←)

Y and Up Arrow Key (⬆) etc. (Read code to understand which keys to press to move specific joint)

SAFETY MEASURES

1. Always run code in parts (comment out and run one thing at a time while testing unless sure)
2. Always start controller button on the box for reducing vibration in motors (Green Button)
3. Streaming rate to Arduino is 100. Thus any message published on the topic '/joint_command' would result in publishing a command to Arduino. Arduino would try to achieve desired angle in 10ms. Tal_brabo_node keeps track of current joint angles of the robot. When any node publishes to '/joint_command' topic tal_brabo_node calculates velocity based on published angles and current angles and accordingly sends command to

Arduino. This velocity will be high if difference between current angles (tracked by tal_brabo_driver_node) and angles published on '/joint_command' is large as time step is only 10ms. Tal_brabo_driver node publishes current joint angles on the topic '/joint_states' so that other nodes can use that to publish desired angles. Therefore always use current angles published on topic '/joint states' to publish desired joint angles. Otherwise a high velocity would result in safety problems.

4. Do not start publishing in between task, always restart tal_brabo_driver node before starting to publish desired joint angles. Because failing to do so would result in same problem described in above point.
5. **When you write a new code, do all the above mentioned steps but don’t connect actual motors directly. Just connect Arduino Mega and always first run the code on URDF model in R-Viz and check if robot moves as expected by your code and only then implement on the actual physical robot.**

------------------------x--------------------------x------------------------x--------------------

Just in case homing sensor connection is required:

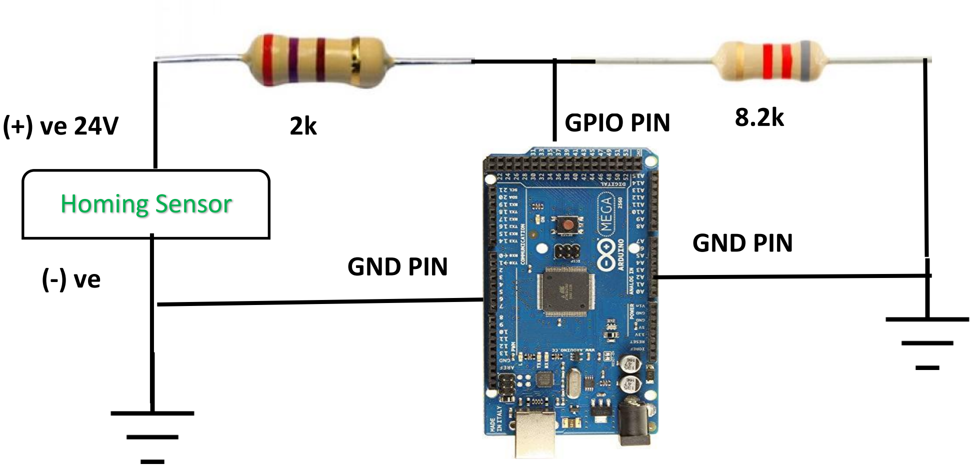


Fig. 3.1 Voltage Divider for Homing Sensor

Vout = Voltage at GPIO Pin of Arduino = 24V * [2k / (2k+8.2k)] = 4.8 V

#Only one ground is tapped from controller and connected to common ground point on breadboard

Connect Homing Sensor of **X Joint to pin 33, of Y Joint to pin 34, Z Joint to pin 35**

of **U Joint to pin 36, of V Joint to pin 37 on Arduino Mega**

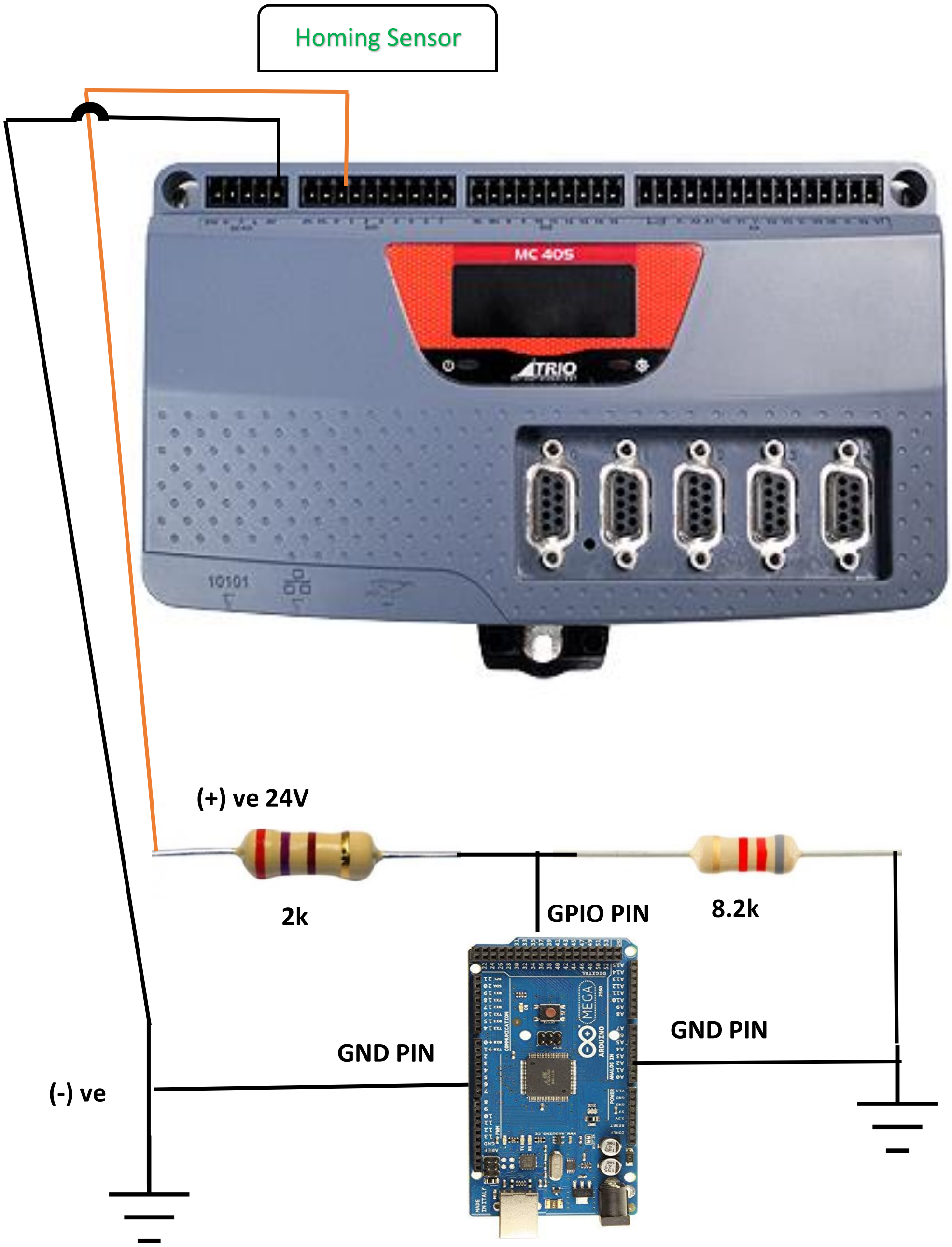
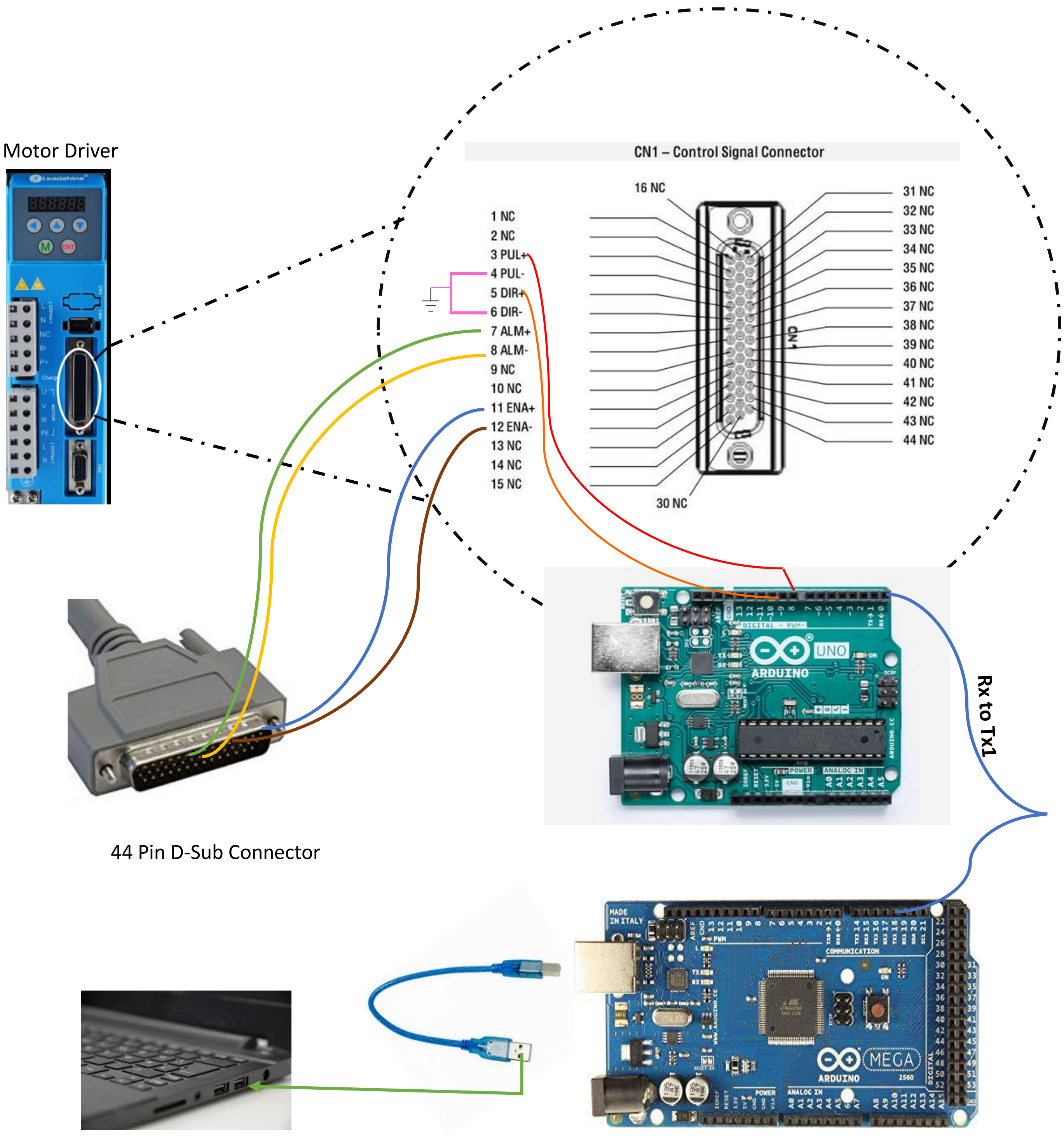
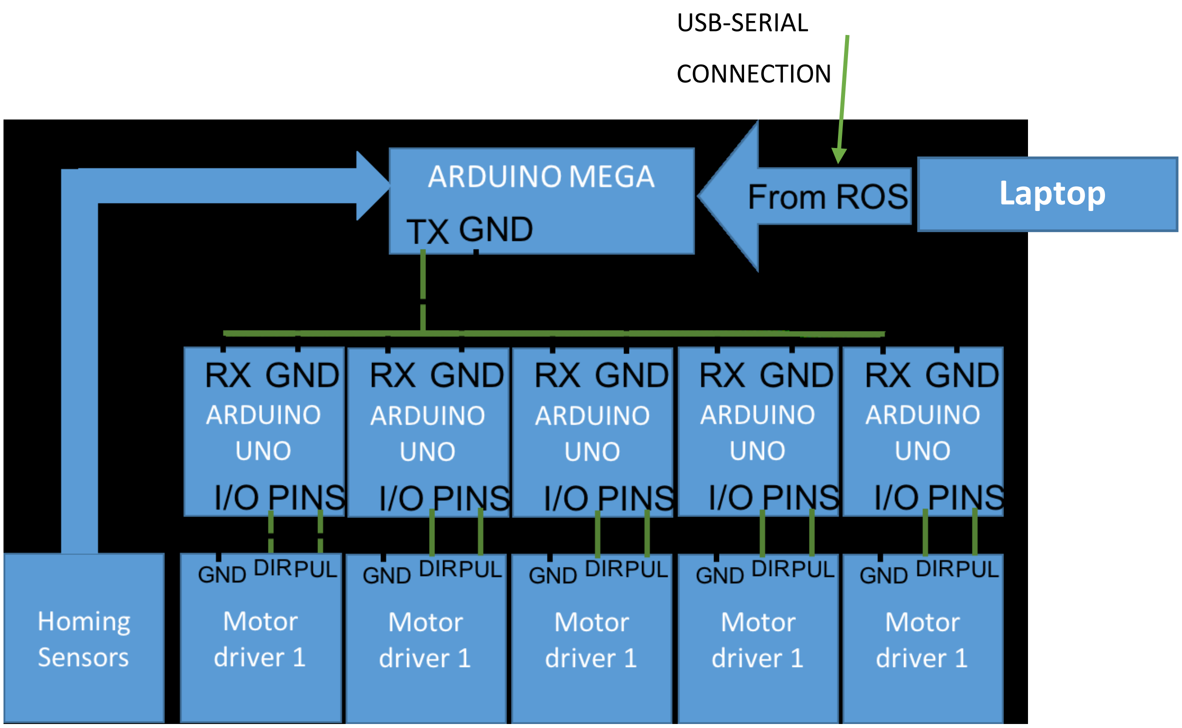


Fig. 3.1 Voltage Divider for Homing Sensor

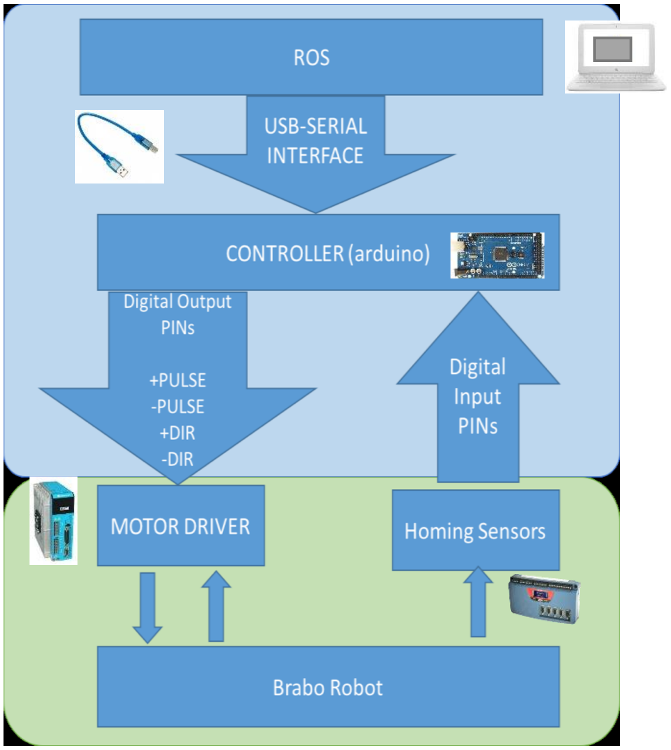
MOTOR CONNECTIONS



PIN 8 on Arduino UNO = PUL+ , PIN 9 on Arduino UNO = DIR+



Overall architecture of the controller



- **Please Note: ENABLE CONNECTION IS NOT REQUIRED FOR MOTOR 4 and MOTOR 5. Just Give Pul+ , Pul-, Dir+, Dir- as per the pin diagram of DC Servo Drive.**
- **You have to give power to every Arduino. But don't take Arduino UNO power from main laptop running the code. Only Arduino Mega should be connected to main laptop running the codes. Rest all Arduinos should draw power from other laptop using USB Hub.**
- **Upload Mega Code on Mega Arduino.**

  And motor1, motor2 etc. on Arduino UNO respectively.
- **Remember on which Arduino UNO you flashed motor1 code (X Joint) and only make connections of Motor Driver X to this Arduino. Don't confuse with Arduino Unos. You might by mistakenly upload motor1 code and connect it to motor Y or Z so be very carefully. Take time but make connections very carefully and verify, because one silly mistake and you will short the main driver. Then even manual teach-pendant control will not be possible.**
- **While uploading code Rx, Tx pin of the Arduino should not be connected**

mega.ino = Mega Code

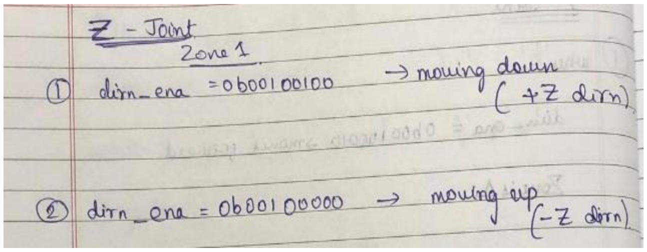
motor1.ino = Motor X code.

motor2.ino = Motor Y code.

motor3.ino = Motor Z code.

motor4.ino = Motor U code.

motor5.ino = Motor V code.

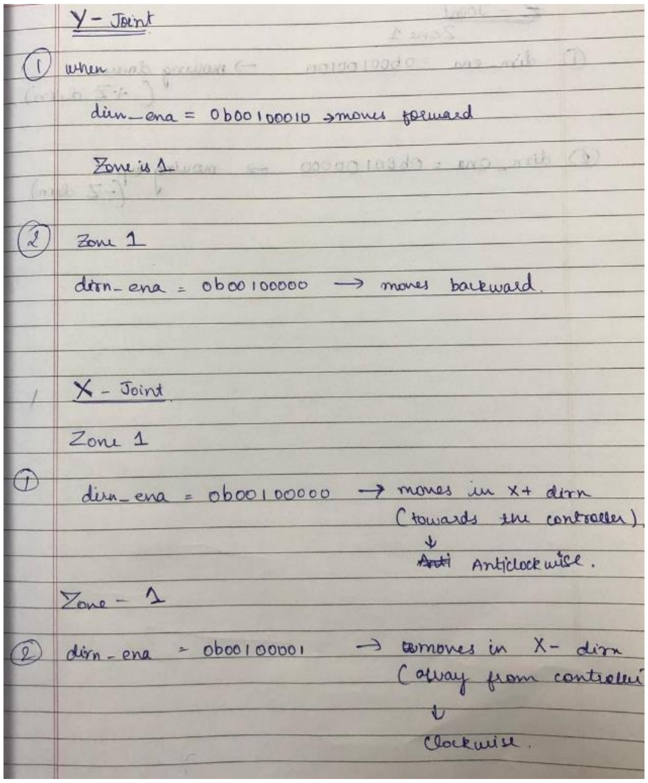


Z - Joint

Zone 1

① dirn_ena = 0b00100100 → moving down (+Z dirn)

② dirn_ena = 0b00100000 → moving up (-Z dirn)



Y - Joint

① when

dirn_ena = 0b00100010 → moves forward

Zone is 1

② Zone 1

dirn_ena = 0b00100000 → moves backward.

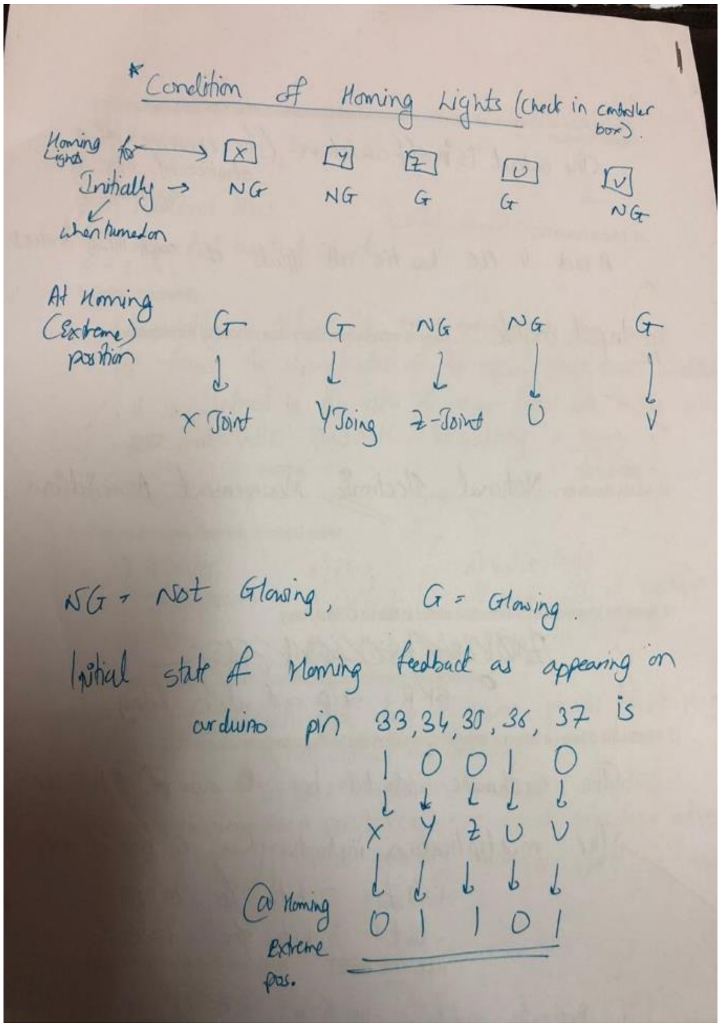
X - Joint

Zone 1

① dirn_ena = 0b00100000 → moves in X+ dirn (towards the controller).
↓
~~Anti~~ Anticlockwise.

Zone - 1

② dirn_ena = 0b00100001 → ~~to~~moves in X- dirn (away from controlle
↓
Clockwise.



*Condition of Homing Lights (Check in controller box).

Homing Lights for →	X	Y	Z	U	V
Initially → ↙ when turned on	NG	NG	G	G	NG
At Homing (Extreme) position	G	G	NG	NG	G
	↓	↓	↓	↓	↓
	X Joint	Y Joing	Z-Joint	U	V

NG = Not Glowing, G = Glowing.

Initial state of Homing feedback as appearing on arduino pin 33, 34, 35, 36, 37 is

	1	0	0	1	0
	↓	↓	↓	↓	↓
	X	Y	Z	U	V
	↓	↓	↓	↓	↓
@ Homing Extreme pos.	0	1	1	0	1

The instructions below are applicable only when you are using Trio Controller and not the Arduino Controller

- **While using normally from matlab make sure to note the mode in which you are moving the robot.**

  JOINT MODE = DEFAULT MODE = 756,0

  LINEAR MODE = 756,1

- **Give MoveAbs Co-ordinates very carefully depending on the Mode of Operation which is Linear or Joint.**
- **Learn the homing pattern.**
- **If you switch off controller in middle of a process, first home it (only for default trio controller) and only then restart the earlier process because the co-ordinate system of robot changes once you restart it.**

(Never run homing procedure in Arduino Control even if u restart the controller in middle because we don't have any reference co-ordinate system there.)

***PLEASE DON'T BE OVER CONFIDENT ***

*****RECHECK*****

…… BE SAFE AND TAKE CARE OF THE ROBOT…...

☺…..ALL THE BEST…… ☺