**OMNI Wheel 4DW Robot**

Provided: Arsham Vosoughinia

Instructor: Dr. Abdelkader Abdessameud

**NEXUS ROBOT RB008\_4WD Omni wheel**

**Power**

the power cord is responsible for charging the batteries. The robot only operates based on the batteries power supply. If the batteries connection socket is disconnected the motors would not operate. There is also another connection socket that powers the red power button of the robot.

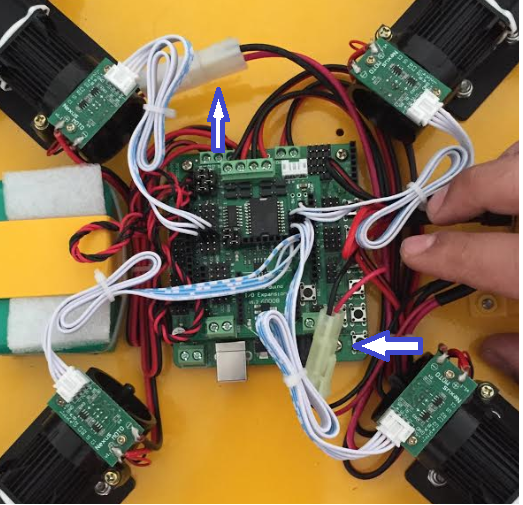
****

Figure 1

**Robot & Motors Orientation for programming**

The correct orientation of the motors can be seen below where the power switch is located on the left side and all the programing is with respect to the following positioning.

**\ /**

**wheel1 \ / wheel4**

**UpperLeft (UL) \ / UpperRight (UR)**

**\* power switch**

**/ \**

**wheel2 / \ wheel3**

**LowerRight (LR) / \ LowerLeft (LL)**

**I/O Expansion board**

The I/O expansion board is mounted on the main Arduino board as shown in figure 1. Noting that there are no male pins on the bottom of expansion board allocated for the POWER female pins of 6 and 7(Lower right side of the Arduino board) on the main Arduino board; Therefore, the expansion board should be installed with respect to the POWER pins of 6 and 7 of the main Arduino remaining empty and not inserted with the male pins of the bottom of the expansion board.

**Encoders**

Each of the motors in this robot contain an encoder placed on top of each motor. Each encoder has 4 pins. The pin description of an encoder can be seen in the images below.



Figure 2

|  |  |
| --- | --- |
| Pin Number | Description |
| 1 | 5 Volts |
| 2 | GROUND |
| 3 | Phase A |
| 4 | Phase B |

All of the four encoders are connected directly to the pins on the expansion board.

**Encoder** pin out on expansion board for **UpperLeft** Motor:

The pins inside the red rectangles in the image below can be found on the left side of the expansion board.

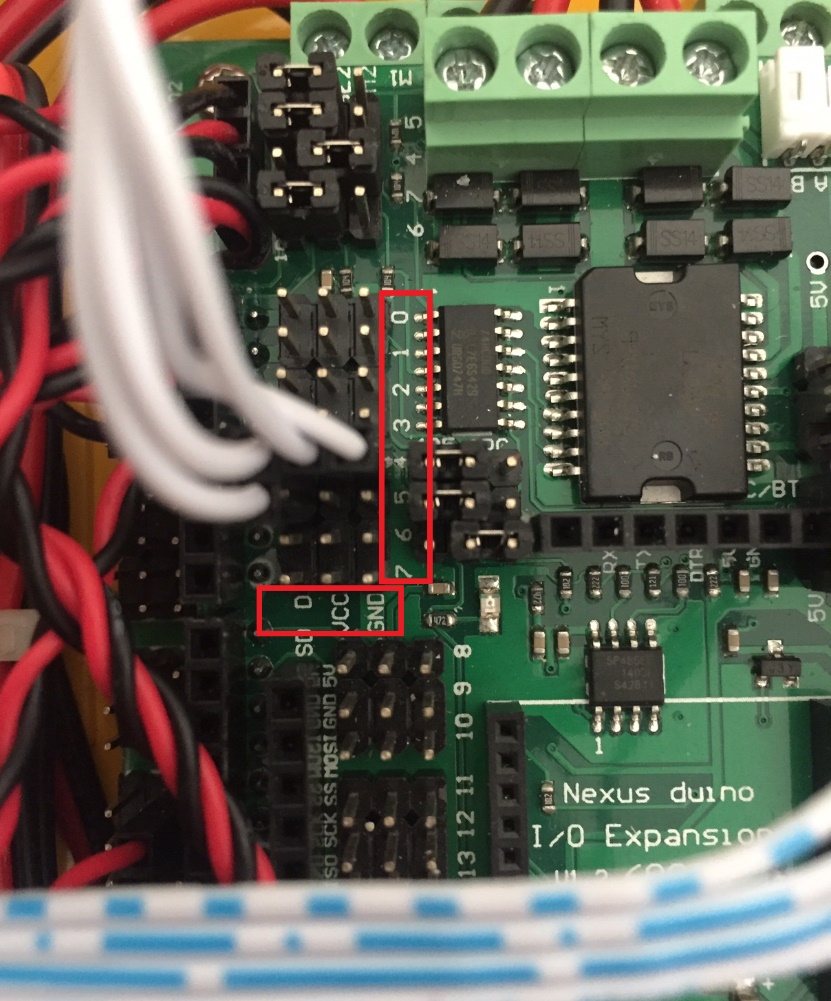


Figure 3

|  |  |
| --- | --- |
| **Pins Encoder for Motor Upper Left (UL)** | **Connection Pin on Expansion Board** |
| Pin 1 (+5V) | VCC on Pin 4 |
| Pin 2 (GND) | GND on Pin 4 |
| Pin 3 (Phase A) | D on pin 4 |
| Pin 4 (Phase B) | D on Pin 5 |

**Encoder** pin out on expansion board for **LowerLeft** Motor:

The pins inside the red rectangles in the image below can be found on the right side of the expansion board.

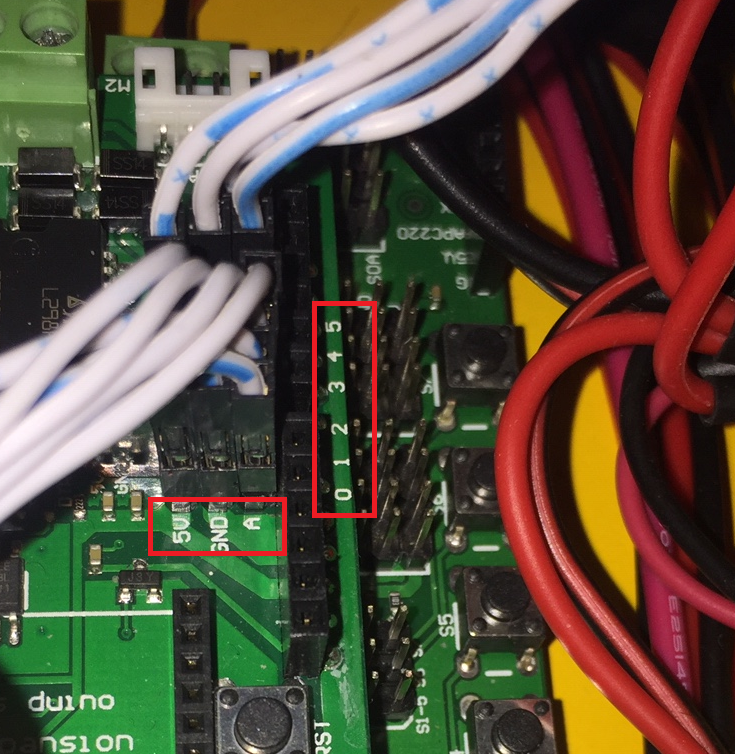


Figure 4

|  |  |
| --- | --- |
| **Pins Encoder for Motor Lower Left (LL)** | **Connection Pin on Expansion Board** |
| Pin 1 (+5V) | 5V on Pin 0 |
| Pin 2 (GND) | GND on Pin 0 |
| Pin 3 (Phase A) | A on pin 0 |
| Pin 4 (Phase B) | A on Pin 1 |

**Encoder** pin out on expansion board for **LowerRight** Motor:

(Refer to Figure 4)

|  |  |
| --- | --- |
| **Pins Encoder for Motor Lower Left (LR)** | **Connection Pin on Expansion Board** |
| Pin 1 (+5V) | 5V on Pin 2 |
| Pin 2 (GND) | GND on Pin 2 |
| Pin 3 (Phase A) | A on pin 2 |
| Pin 4 (Phase B) | A on Pin 3 |

**Encoder** pin out on expansion board for **UpperRight** Motor:

(Refer to Figure 4)

|  |  |
| --- | --- |
| **Pins Encoder for Motor Lower Left (LR)** | **Connection Pin on Expansion Board** |
| Pin 1 (+5V) | 5V on Pin 4 |
| Pin 2 (GND) | GND on Pin 4 |
| Pin 3 (Phase A) | A on pin 4 |
| Pin 4 (Phase B) | A on Pin 5 |

**Uploading a code sketch**

* Before uploading a code select your board as Arduino Duemilanove or Diecimila on the Tools tab. Then select the correct port.

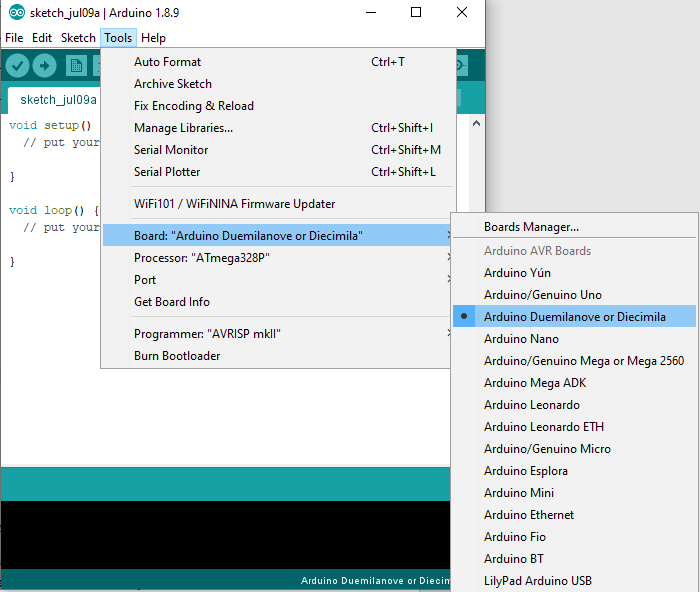


Figure 5

* Code sketches can not be uploaded to the board unless the shown JUMPER in the image below is removed before uploading the code and placed back after the upload is done.

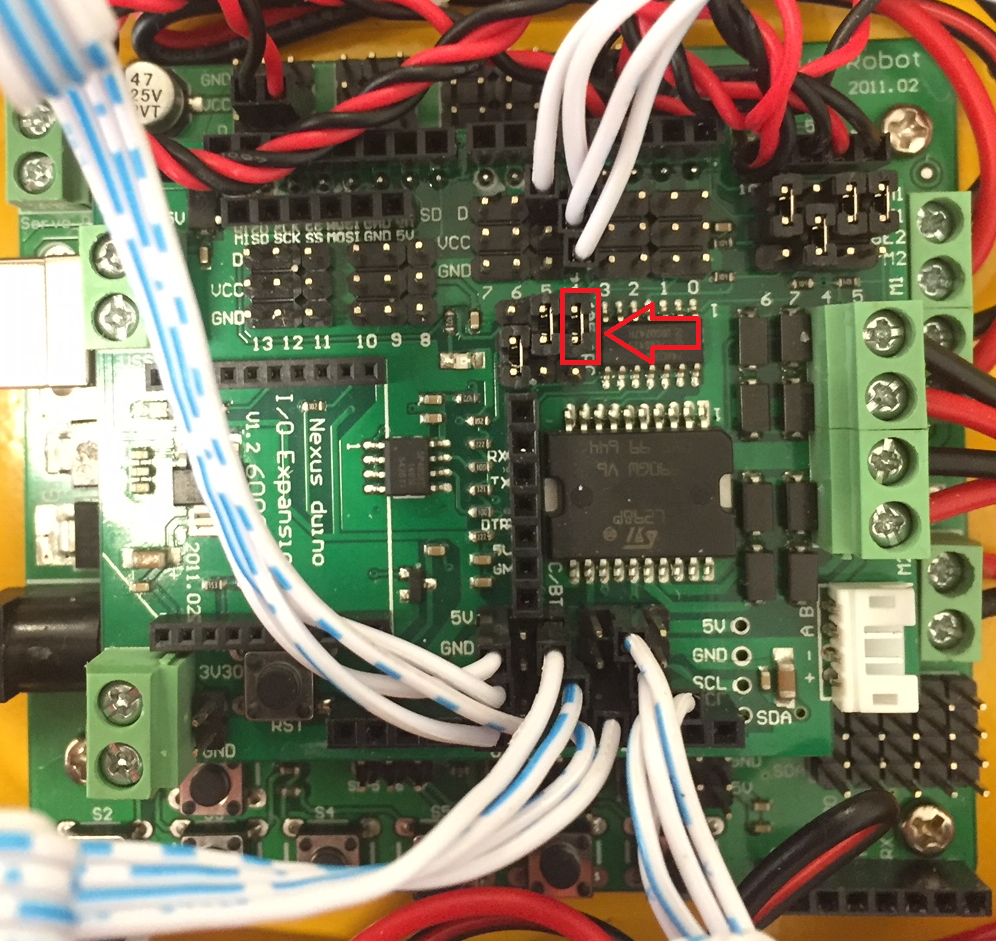


Figure 6

**Coding**

* Speed in this document used is Millimeter Per Second (MMPS). For other types of adjusting velocity for this robot refer to the user manual.
* Refer to **the program provided by the company for PWM and Pin set up configuration**.
* Libraries provide vital information to understand the code lines present in the companies provided program.
* Desirable code commands and functions can be written with respect to the functions and configurations defined in various libraries provided by the company.
* Refer to the user manual for code samples and demo actions.
* All of the commands in the void loop() section can be rewritten using FOR or WHILE loops.

**USABLE COMMANDS:**

**Moving Right**

*void loop() {*

*Omni.setCarRight(Speed in MMPS);*

*Omni.delayMS(MS);*}



Figure 7

Similarly;

**Moving Left:**

*void loop() {*

*Omni.setCarLeft(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Advancing:**

*void loop() {*

*Omni.setCarAdvance(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Backing Off:**

*Omni.setCarBackoff(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Moving Upper Right:**

*void loop() {*

*Omni.setCarUpperRight(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Moving Lower Right:**

*Omni.setCarLowerRight(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Moving Upper Left:**

*void loop() {*

*Omni.setCarUpperLeft(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Moving Lower Left:**

*Omni.setCarLowerLeft(Speed in MMPS);*

*Omni.delayMS(MS);*}

**Setting car slow to stop**

void loop() {

Omni.setCarRight(MMPS);

Omni.delayMS(MS);

Omni.setCarSlow2Stop(Stoppage in MS);}

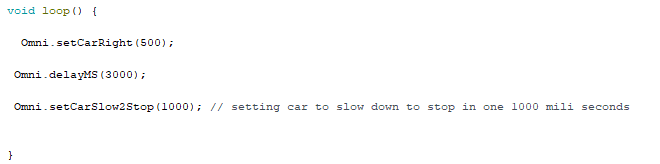


Figure 8

**Codes for Controlling each motor singularly**

Each motor’s Speed and Direction can be set using the command lines below. By applying different speeds to the motors, the car can move in a circular pattern. The Robot can move in different circular path patterns depending on the speed and direction of each motor.

**Upper Left Wheel:**

*Omni.wheelULSetSpeedMMPS(MMPS,DIR\_ADVANCE);*

*Or*

*Omni.wheelULSetSpeedMMPS(MMPS,DIR\_BACKOFF);*

**Lower Left Wheel:**

*Omni.wheelLLSetSpeedMMPS(MMPS,DIR\_ADVANCE);*

*Or*

*Omni.wheelLLSetSpeedMMPS(MMPS,DIR\_BACKOFF);*

**Lower Right Wheel:**

*Omni.wheelLRSetSpeedMMPS(MMPS,DIR\_ADVANCE);*

*Or*

*Omni.wheelLRSetSpeedMMPS(MMPS,DIR\_BACKOFF);*

**Upper Right Wheel:**

*Omni.wheelURSetSpeedMMPS(MMPS,DIR\_ADVANCE);*

*Or*

*Omni.wheelURSetSpeedMMPS(MMPS,DIR\_BACKOFF);*

**Sample code for moving in a CIRCLE using the top commands:**

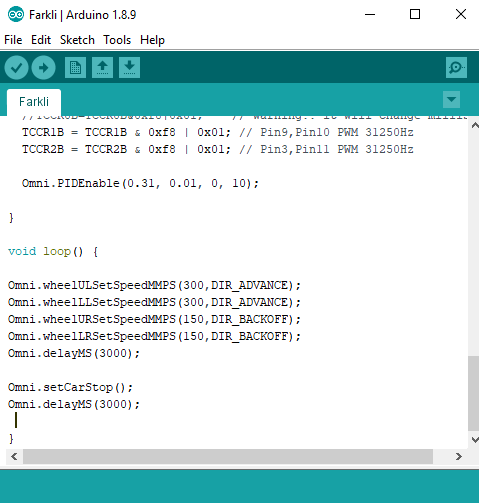
**

Figure 9