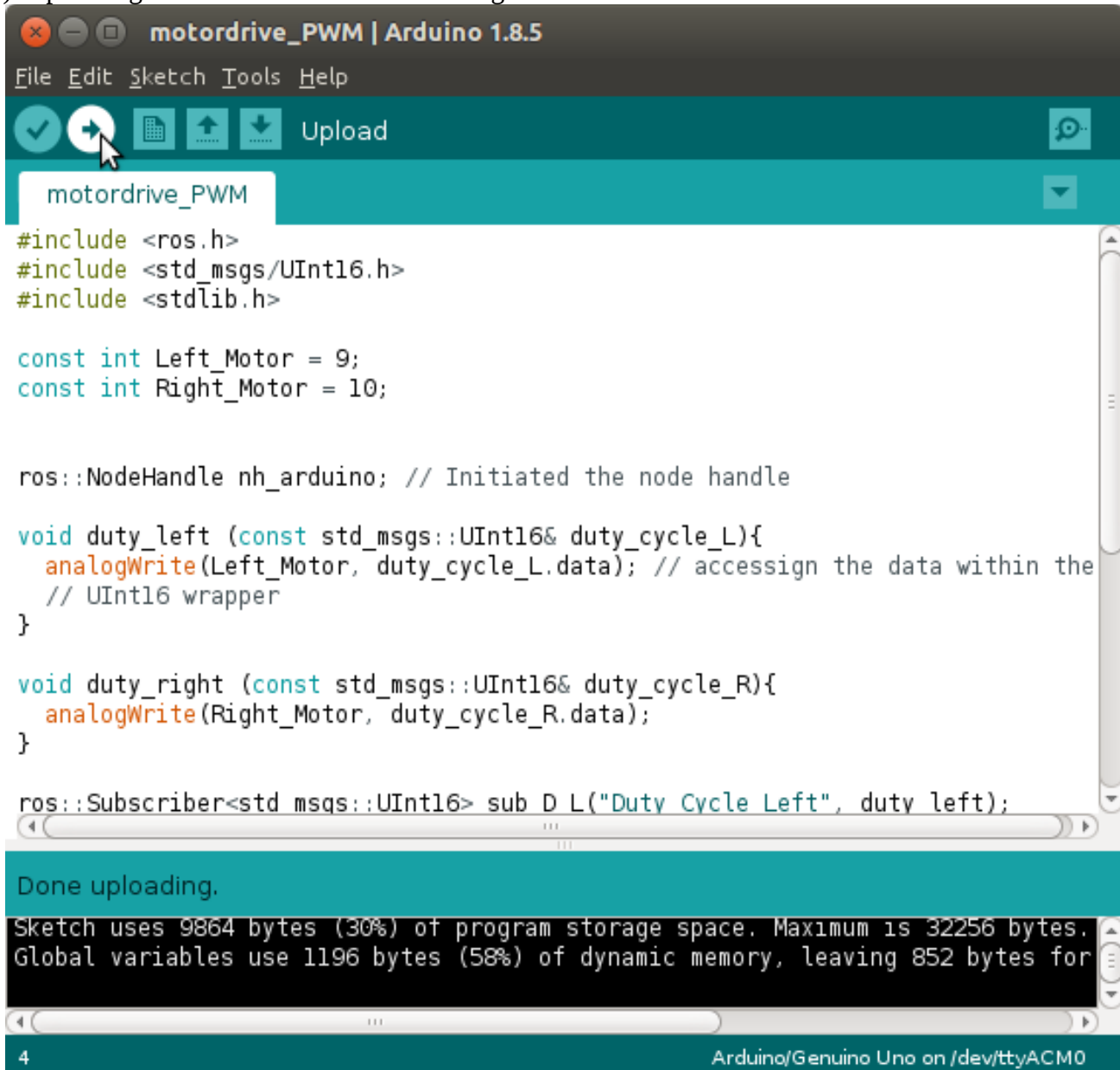


Steps:

1. Load Arduino Code onto board
2. Start ROS nodes

Step 1:

a). Uploading the Arduino code is done through the IDE



The screenshot shows the Arduino IDE interface. The title bar reads 'motordrive_PWM | Arduino 1.8.5'. The menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for opening, saving, and uploading, with the 'Upload' button (a right-pointing arrow) highlighted by a mouse cursor. Below the toolbar, the sketch name 'motordrive_PWM' is displayed. The main text area contains the following C++ code:

```
#include <ros.h>
#include <std_msgs/UInt16.h>
#include <stdlib.h>

const int Left_Motor = 9;
const int Right_Motor = 10;

ros::NodeHandle nh_arduino; // Initiated the node handle

void duty_left (const std_msgs::UInt16& duty_cycle_L){
  analogWrite(Left_Motor, duty_cycle_L.data); // accessign the data within the
  // UInt16 wrapper
}

void duty_right (const std_msgs::UInt16& duty_cycle_R){
  analogWrite(Right_Motor, duty_cycle_R.data);
}

ros::Subscriber<std msqs::UInt16> sub D L("Duty Cycle Left", duty left);
```

Below the code editor, a status bar shows '4' on the left and 'Arduino/Genuino Uno on /dev/ttyACM0' on the right. A console window at the bottom displays the upload results:

```
Done uploading.
Sketch uses 9864 bytes (30%) of program storage space. Maximum is 32256 bytes.
Global variables use 1196 bytes (58%) of dynamic memory, leaving 852 bytes for
```

b). Make sure that Arduino is connected to the computer through a USB port

Step 2.:

a) Execute the following commands in this order

```
utcr2018@utcr2018-OptiPlex-3010:~$ cd catkin_ws/
utcr2018@utcr2018-OptiPlex-3010:~/catkin_ws$ source devel/setup.bash
utcr2018@utcr2018-OptiPlex-3010:~/catkin_ws$ roslaunch motordrive Joystick_Drive.launch http://localhost:11311/
ex-3010-875.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://utcr2018-OptiPlex-3010:35951/

SUMMARY
=====

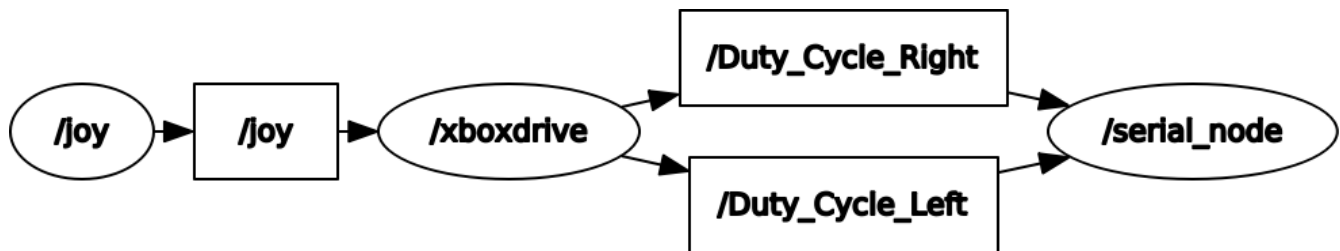
PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.13
* /serial_node/baud: 57600
* /serial_node/port: /dev/ttyACM0

NODES
/
  joy (joy/joy_node)
  serial_node (rosserial_python/serial_node.py)
  xboxdrive (motordrive/xboxdrive)

auto-starting new master
process[master]: started with pid [889]
ROS_MASTER_URI=http://localhost:11311
```

b) your rqt_graph should look like this:

Execute the command 'rqt_graph' in a separate terminal window



Driving the robot:

- 1) pushing A starts the robot – green light will turn on
- 2) left stick will move forwards and backwards
- 3) right stick will steer like a jet ski – must be moving under power to steer

If the Arduino does not connect, try `roslaunch motordrive Joystick_Drive.launch port:=/dev/tty/ACMX` where X is 1,2,3,4....