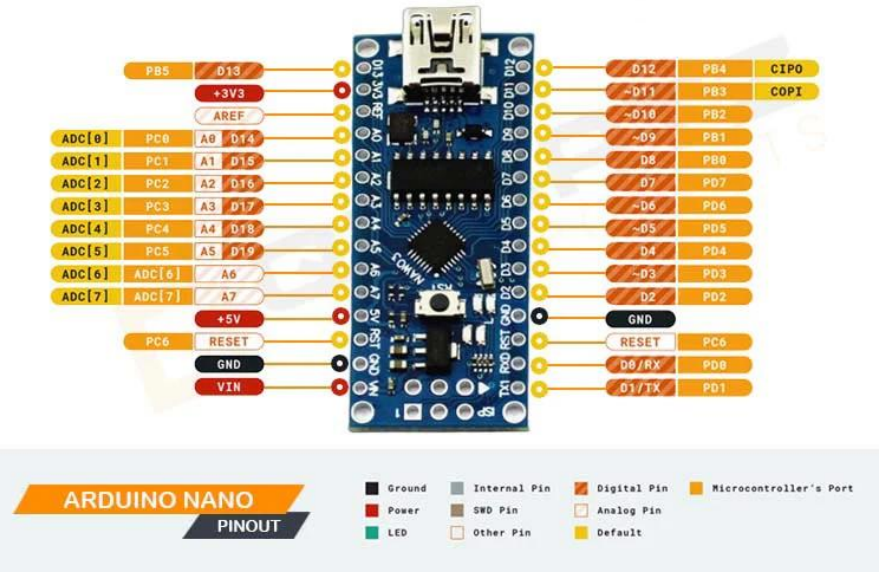
MXET 375 Final Project
Guide for Arduino Nano & DC Motor Control

Introduction

Basics of Arduino Nano

It's likely at this point most of you have heard of or have experience working with an Arduino microcontroller. Arduino is an open-source electronics platform for use in a wide range of DIY projects.



Most Arduino come with a wide range of GPIO or general-purpose input and output digital interfaces. The GPIO pins are best used to connect to a variety of digital-type sensors or actuators. A list of common digital sensors and actuators can be found here:

https://www.thegeekpub.com/wiki/list-of-arduino-sensors-and-modules/

There is also an available set of pins that can be used for analog inputs and outputs.

Given the wide range of available sensors and actuators, the Arduino has become one of the most popular platforms for robotics projects such as the one you will be working on for your final project. For those who have not worked with Arduino previously, watch the video provided here to understand how to read, write, compile, and execute code to your Arduino.

https://www.youtube.com/watch?v=9cxAjRHdMVY
https://www.arduino.cc/en/software

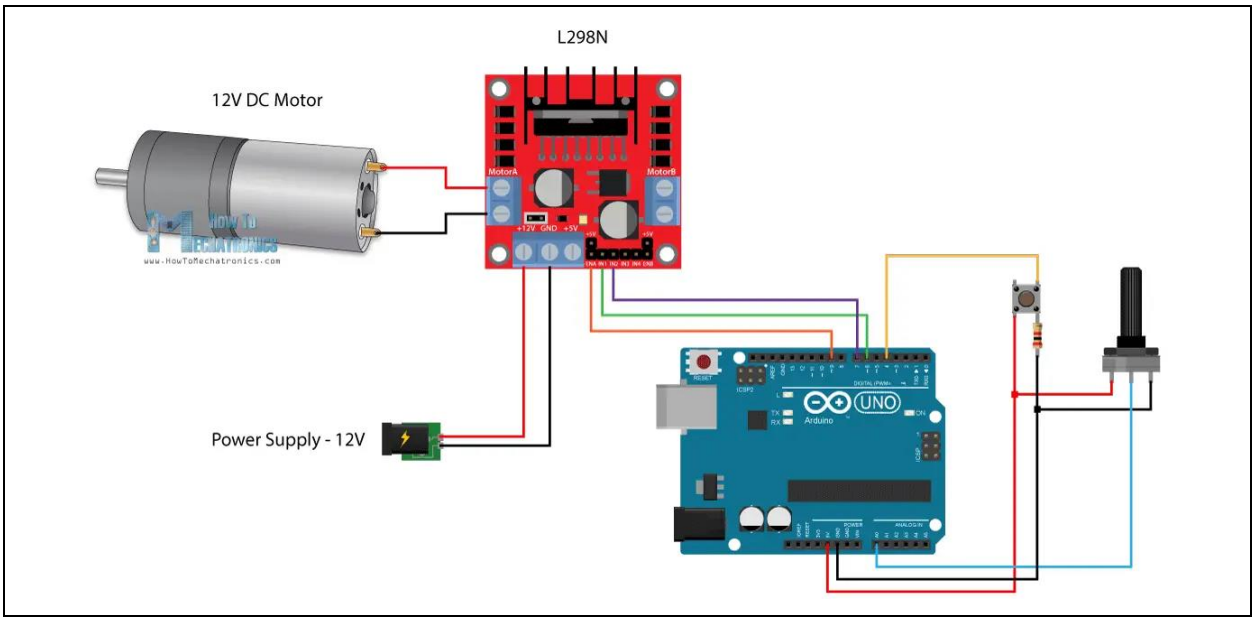
Arduino programs are written using a variation of C++ which provides a fast and efficient way to create complex programming sequences. It's highly recommended to use open-source projects to help write the code for your project. While an open-source project might not give you exactly what you need for it, they can help in understanding basic algorithmic structure and basic programming sequences that you can modify to meet your needs. Of course, whenever you use programs from an open-source project you should adequately **spend the time to read and understand how the code is written rather than just running it and "hoping it works".** In the past, students have had good luck with the following:

Instructables.com
Github.com

Feel free to use these resources and others to help you write, debug, and execute your programs***. Good Luck!***

DC Motor Control

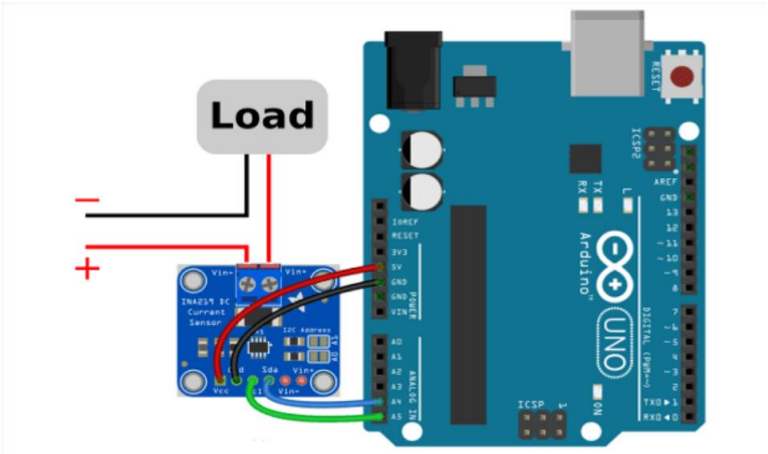
The following is an outline and source code for a project used in previous semesters. You may find this helpful in your project. There is a text file attached with this document that contains the source for the following project.



The goal of this project was to create a DC motor powertrain to estimate the torque and power consumption from a simple DC motor. Included in this project was a INA219, Hall Effect Rotary Encoder, L298N motor controller, Potentiometer, and Arduino Nano as shown in the figure above. To measure the amount of electrical power consumed by this setup the INA219 is placed in series with the high side (positive) input from the 12v Power supply to the L298N motor driver. To prevent damage to the digital pins on the Arduino a the L298N acts as a dedicated H-bridge that can handle the high current consumption for the 12V DC motor. The potential meter is used to create an adjustable voltage that can be used to change the duty cycle to adjust the RPM of the motor.

Current Sensing w/ the INA219 DC Current Sensor

- As you might recall from your basic electronics courses to measure current the positive voltage line of your load must be broken (i.e. the sensor must be placed within series. It must be connected as shown below.



- As shown in this picture the INA219 is reading the current for the load.
- ***** Important***** Do not place the INA219 in line before or after the DC motor. Instead, you should consider placing the INA219N before the L298n H-Bridge or the PWM Driver. It's possible only with smaller DC motors. Whilst this is not a 'perfect' measurement of current to through the motor the INA219 has a finite current limit and placing it next to the DC motor can cause the internal chip to become faulty or damaged because of the switching of the motors. The motor H-bridge /PWM is regulated to protect the voltage source lines so this guarantees some protection to the module.
- I highly recommend using the INA219 library supported by Adafruit. This is fairly user friendly considering you setup is correct.

ENCODER and PWM Control

- Below is a sample of code used to program a small hobby DC motor via PWM control. This DC motor includes a Hall Effect Type Rotary encoder which was used to record the speed of the motor. The INA219 Current sensing module was placed in series with the voltage input line to the DC motor h-bridge as mentioned before.
- When using PWM to control the motors speed the current sensing readings may become noisy and unstable. Its recommended to adjust your PWM frequency to 100% duty cycle at least for your final testing. Please ensure your circuitry/mechanical design is capable of handling this before doing so. If you find the motor is running too fast you can simple lower the input voltage to your system just account for this in your dynamics model.
- Below is the code utilized from a project a few years ago. Please utilize this as reference if you are still struggling with data acquisition. Furthermore, if you are using other sensors such as accelerometers, ToF, Ultrasonic, etc. you can modify this code block.

Acknowledgment

This guide was developed by Mr. Joshua Jalamo, an MXET Alumni and MSET graduate student.