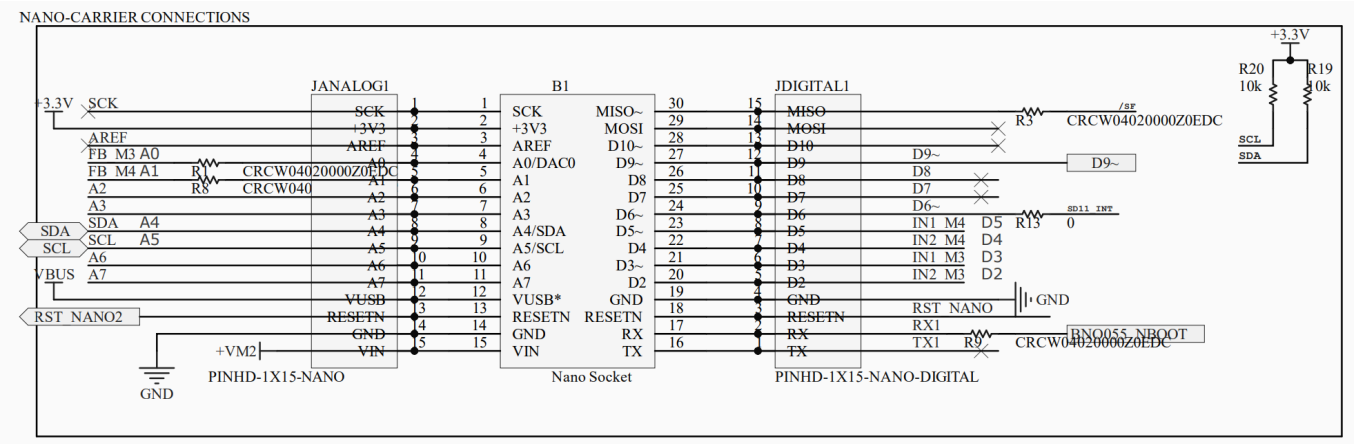
Reading Schematics and Datasheets

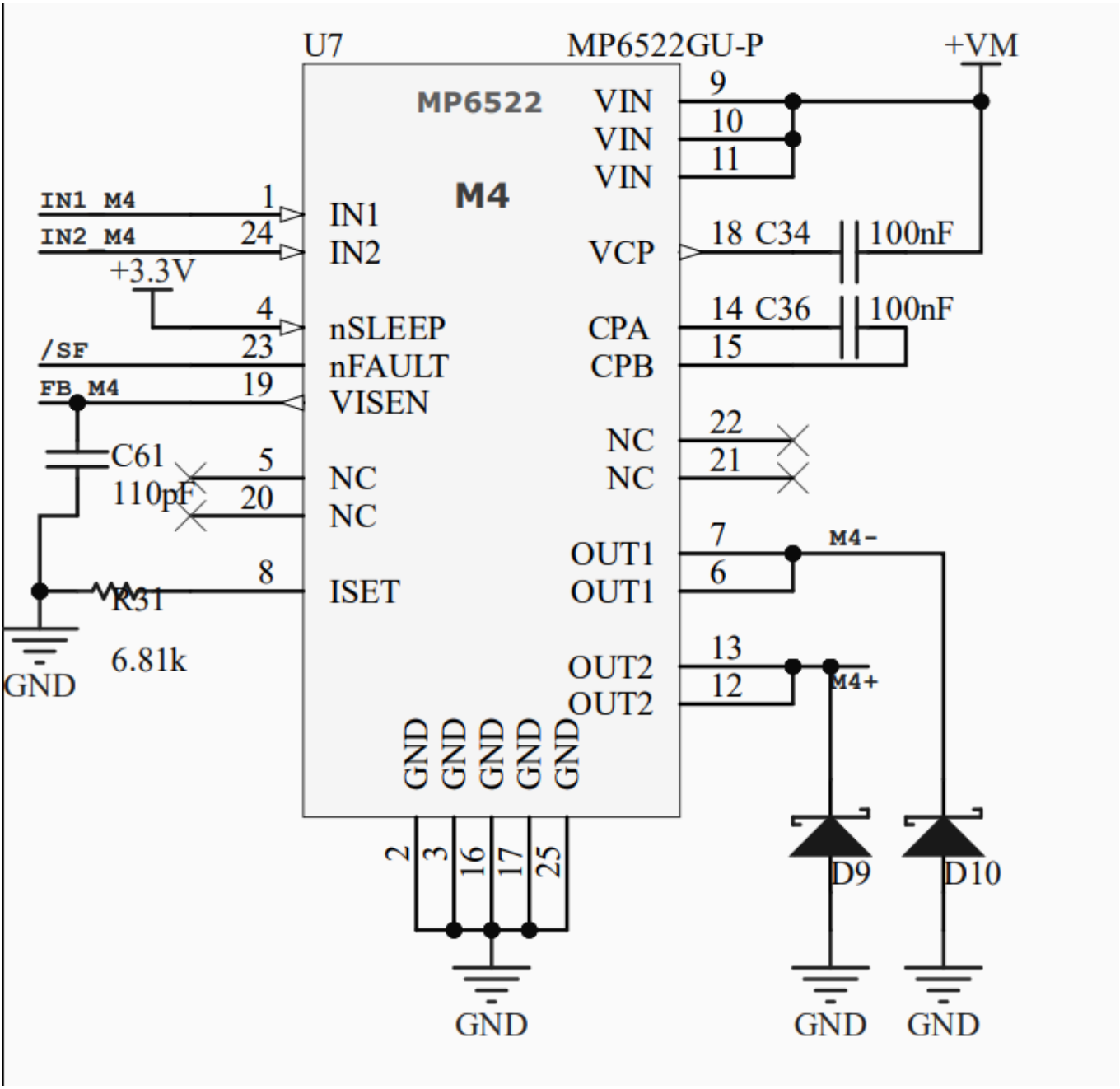
When reading schematics, it's important to read in order, From -> To.

When looking at a microcontrollers I/O schematic like this:



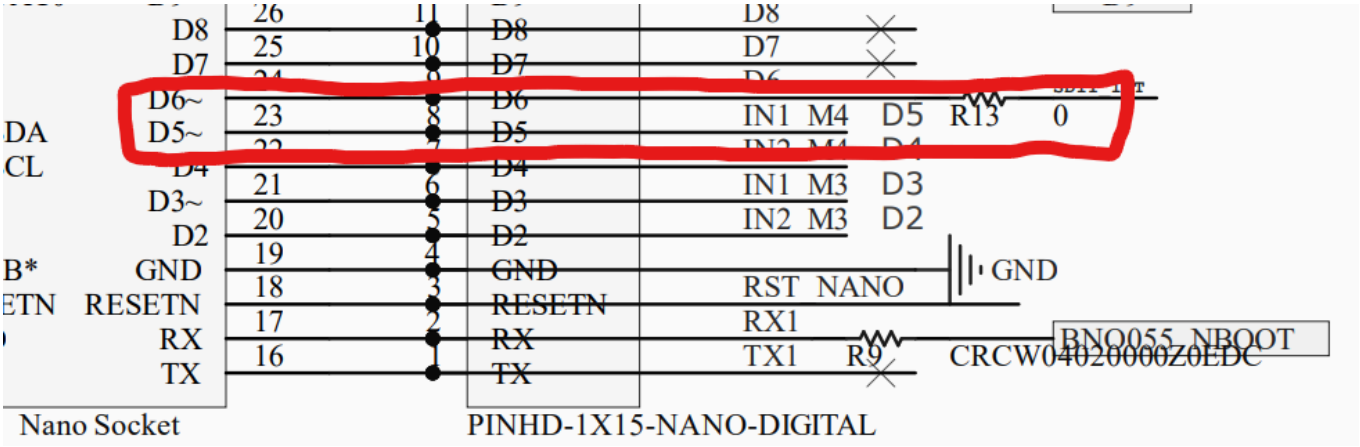
(Image 1: Arduino Nano IoT 33 I/O)

I.E. we're looking at a Nano Motor Driver, and specifically at the Nano Socket. And in this example, we want to know how to make a wheel move. Therefore, we look at the Motor Drivers M1, M2, M3, M4. When looking at the MP6522 M4, we see that Pin IN1 has a input IN1 M4.



(Image 2: MP6522 M4 schematics)

If we go back to the previous picture, we see that at D5 has the similar name IN1 M4



This means that for when we write code for the Arduino, we can write something like this, to control the high or low signals for the motor. BUT, this is just an assumption, we have to search up the datasheet of the MP6522, to see what IN1 is.

PIN FUNCTIONS

Pin #	Name	Description
1	IN1	**Control inputs.** IN1 and IN2 have internal pull-down resistors.
24	IN2	

As we've confirmed, IN1 and IN2 are Control inputs, conveniently lets us control the PWM signal, and HIGH, LOW signal if we needed.

We also need to figure out the OUT1, and OUT2 signals, from which when we read the datasheet from the MP6522, which we know from looking at the schematics at image2, is D10(OUT1), and D9(OUT2)

Input Logic

For the MP6522, control of the outputs is accomplished through IN1 and IN2 (see Table 1).

Table 1: Output Control Pins

IN1	IN2	OUT1	OUT2	Function
0	0	Z	Z	Coast
0	1	L	H	Forward
1	0	H	L	Reverse
1	1	L	L	Brake (low)

At the same datasheet, we can confirm and see how the H-bridge looks like

BLOCK DIAGRAM

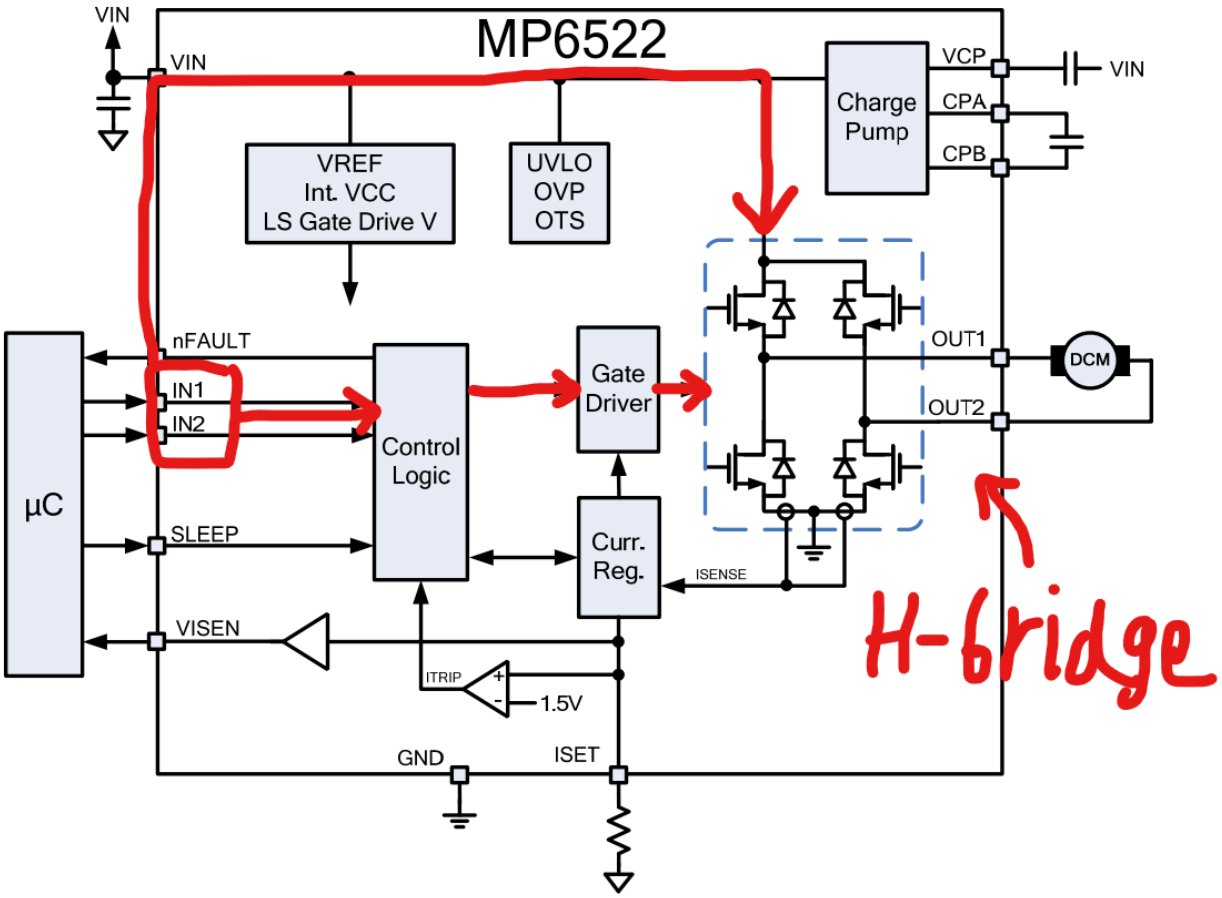
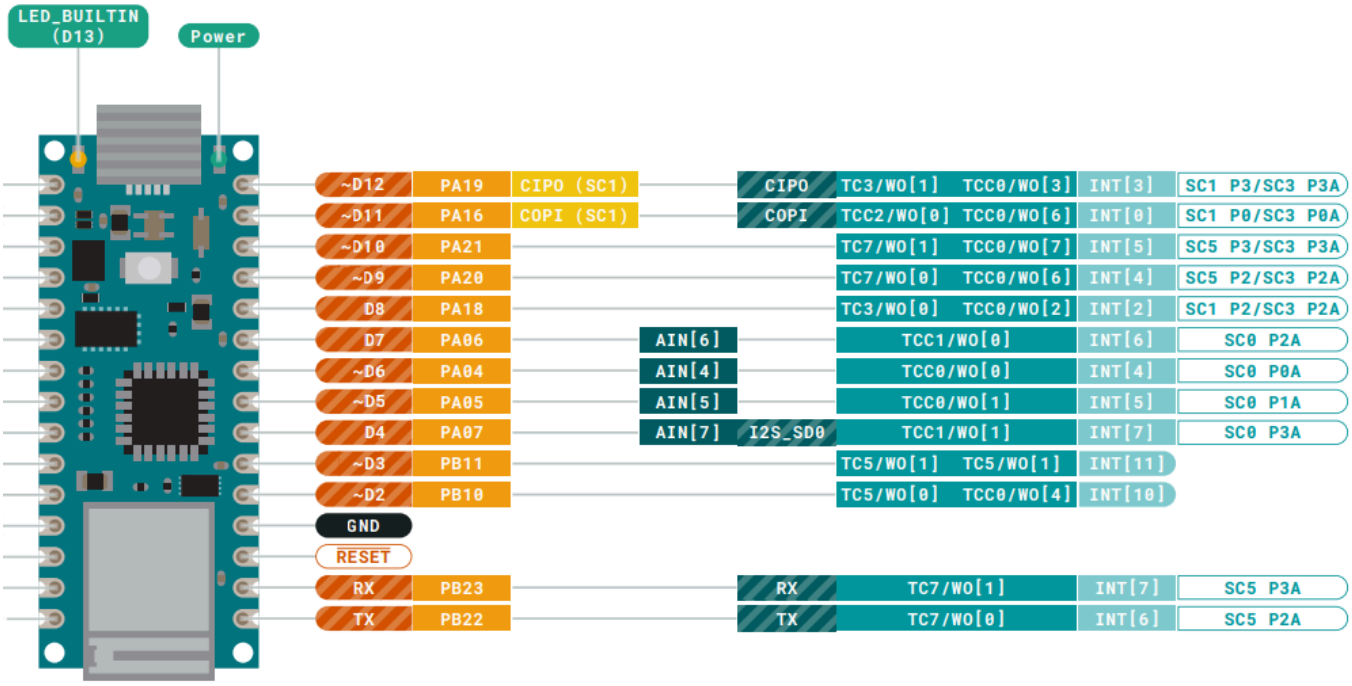


Figure 1: Functional Block Diagram

Now that the last remainder, is; when we're going to code, we need to know how/what we can code, analog or digital. Then we look at the schematics for the Arduino Nano itself and see that

~D5 as a Digital Pin and can be used for PWM, and ~D10 is only Digital Pin.



So, with all of this, we can now write code with the aruino;
Example code, for demonstration:

```ino
#include <Arduino.h>
// Calling the fuctions to be used
void forward(int speed);
void backwards(int speed);
void brake();
void coast();

const int M4_PWM1 = 5;  //  D5 -> MP6522 IN1
const int M4_PWM2 = 4;  //  D4 -> MP6522 IN2

void setup(){
  // To use PWM signal, we have set pinMode to OUTPUT
  pinMode(M4_PWM1, OUTPUT);
  pinMode(M4_PWM2, OUTPUT);
}
void loop(){
  forward(128);   // 50% PWM
  backwards(128); // 50% PWM
  brake();        // Immediate stop
  coast();        // Relaxed stop
}

// Custom functions
void forward(int speed) {
  speed = constrain(speed, 0, 255);
  digitalWrite(IN1, LOW);  // OUT1 Low
  analogWrite(IN2, speed); // OUT2 High
}

void backwards(int speed) {
  speed = constrain(speed, 0, 255);
  digitalWrite(IN2, LOW);  // OUT1 LOW
  analogWrite(IN1, speed); // OUT2 HIGH
}

void brake() {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
}

void coast() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
}
```