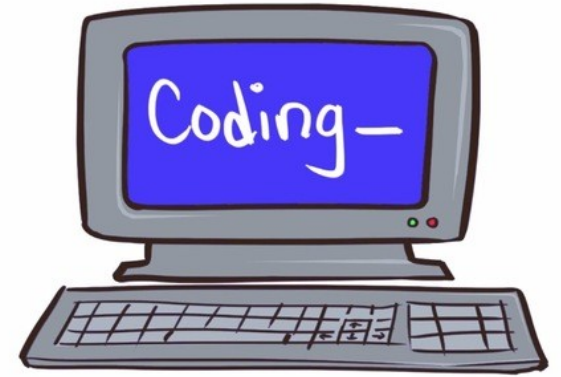


# Class 40



hbridge & infrared pickup

<https://www.arduino.cc/en/Hacking/LibraryTutorial>

<http://playground.arduino.cc/Code/Library>

# Delete libraries

Adafruit\_Sensor

We will use diff IMU

Adafruit\_10DOF

Adafruit\_BMP085\_Unified

Adafruit\_LGD20\_U

Adafruit\_LSM303DLHC

QuadEncoder

Needs mods

# Ternary operator:

= (x ? AAA : BBB)

Logical: If x is true, use AAA else use BBB

```
int x = true;
```

```
Serial.print( x ? "It's True" : "Not True!" );
```



# Infrared “linefollower”

Vcc → 3.3V

Out → GPIO30 (“BUTTON1”)

Gnd – Gnd



One IR device emits, reflected IR detected on other.  
“Conditioned” to be digital output.

For today, can simply use ERC32 to watch the response.

Or try `watch1input.ino` as following:

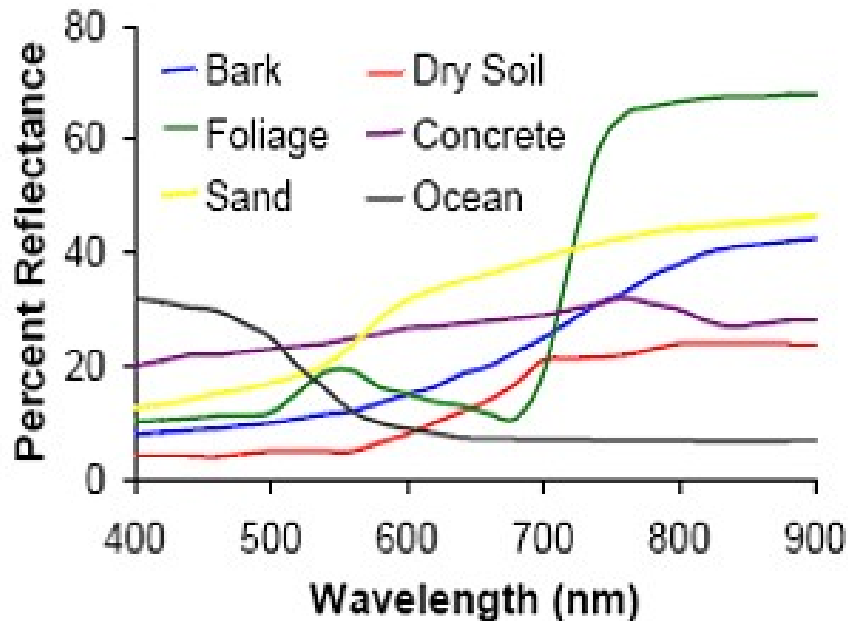
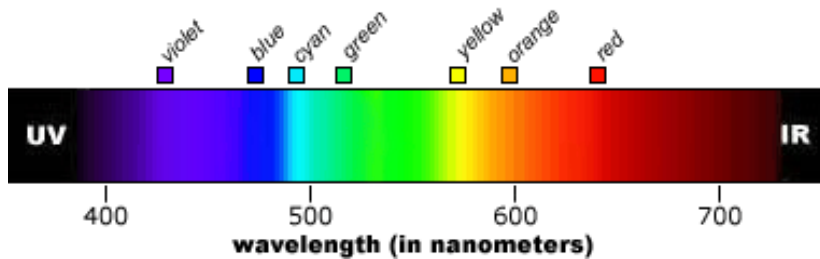
```
#include "myconfig.h"  
#include <Arduino.h>
```

```
void setup() {  
  Serial.begin(SERIAL_BAUD);  
  pinMode (BUTTON1, INPUT);  
}
```

Can you spot the MACROs ?  
Can you see the ternary ?  
Can you understand the printf ?

```
void loop() {  
  Serial.printf("Currently %s\n", digitalRead(BUTTON1) ? "ON" : "OFF");  
  delay(300);  
}
```

# IR reflectance not nec similar to visible light reflectance



Wavelengths:

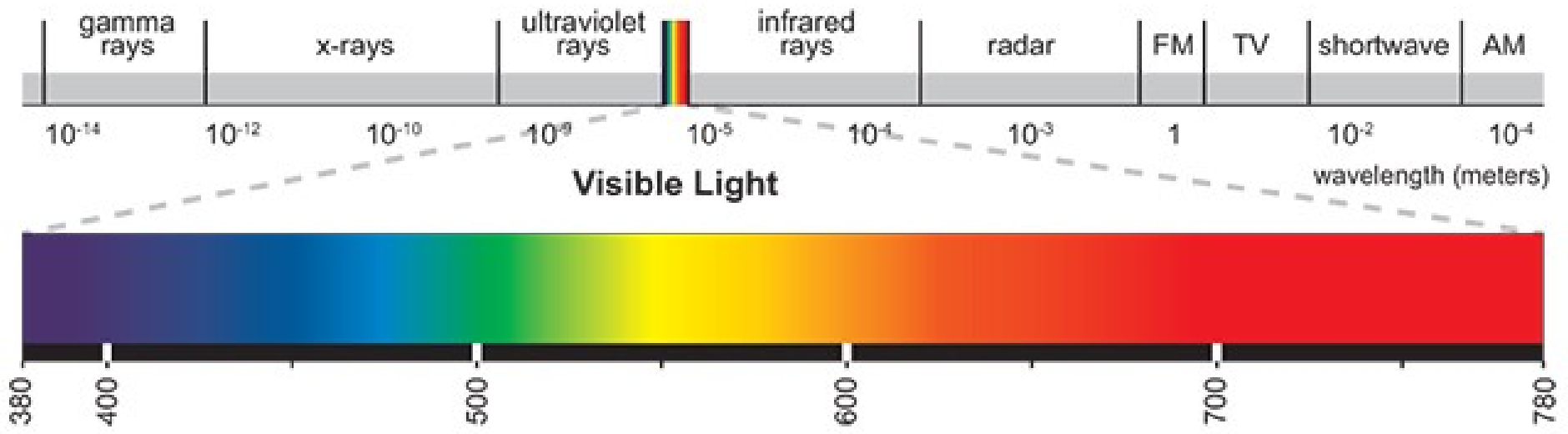
Visible light 380nm – 800nm

IR 1,000nm – 100,000nm !

The LEDs are often “near IR”  
often 850nm or 940nm

Even so, reflectance may not  
be as obvious as our  
visible “black” vs “white”





# Roles:

1) Line follower

- adjust steering to track along a ground line

2) Wheel rotation counter

- count the passing spokes

3) Wheel rotation “quad” counter

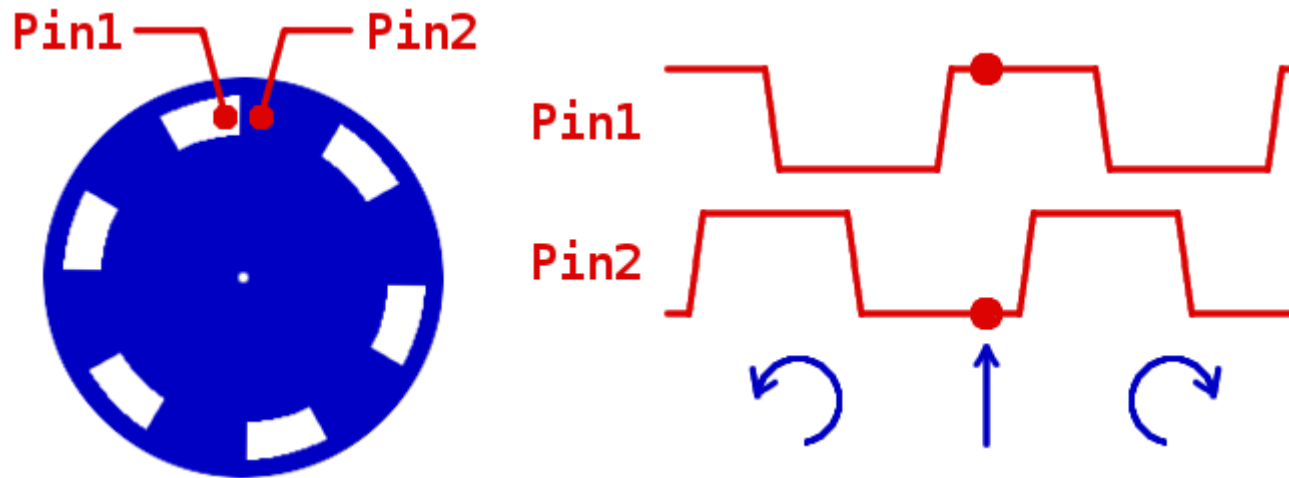
- count the passing spokes, + or – directions

4) Close proximity detector

- short range “radar”

# Quad Encoder

[https://www.pjrc.com/teensy/td\\_libs\\_Encoder.html](https://www.pjrc.com/teensy/td_libs_Encoder.html)



Or <http://www.creative-robotics.com/quadrature-intro>

# Counter / QuadEncoder Library

Next term

# Experiment !

What surfaces are good or bad reflectors?

What range for a front radar?

How accurately can we “focus” for using slotted wheel, and what paint or surface could be best?



Back to our H Bridge ...

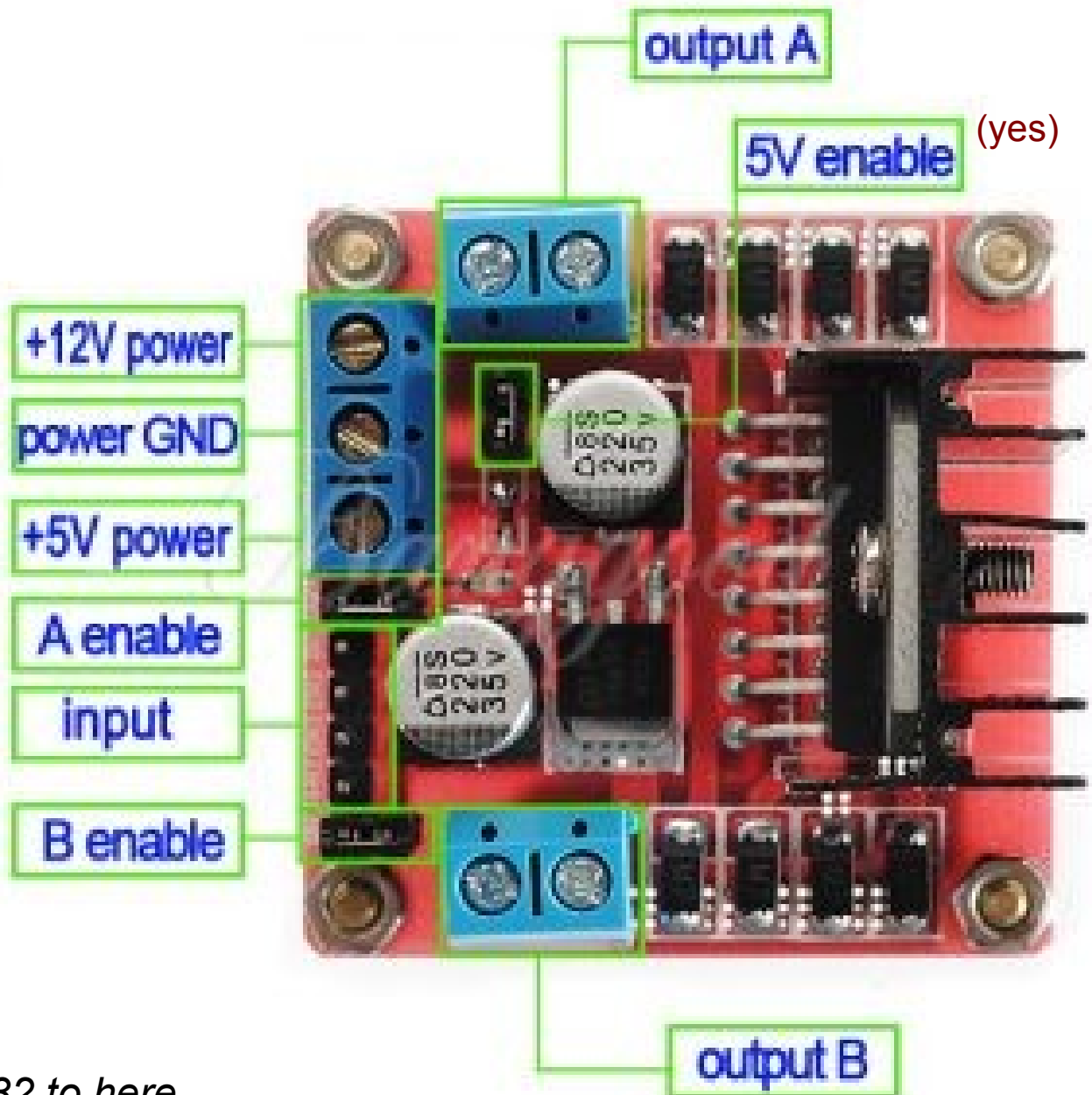
6volt or 7.5volt battery pack  
(4 or 5 AA cells)

“+12V” = Battery Pack positive  
“power GND” = Battery Gnd  
and GND from ESP32  
“+5V” = not connected

A/B “enable” = PWM  
for speed control.  
(Remove these 2 jumpers)

In1/2 and In3/4 = run/direction  
for motors A/B

Note there *is* a GND from ESP32 to here,  
but **no** 5V or 3.3V from ESP32 to here.



Your 2 motors A/B  
(at 6 or 7.5V)



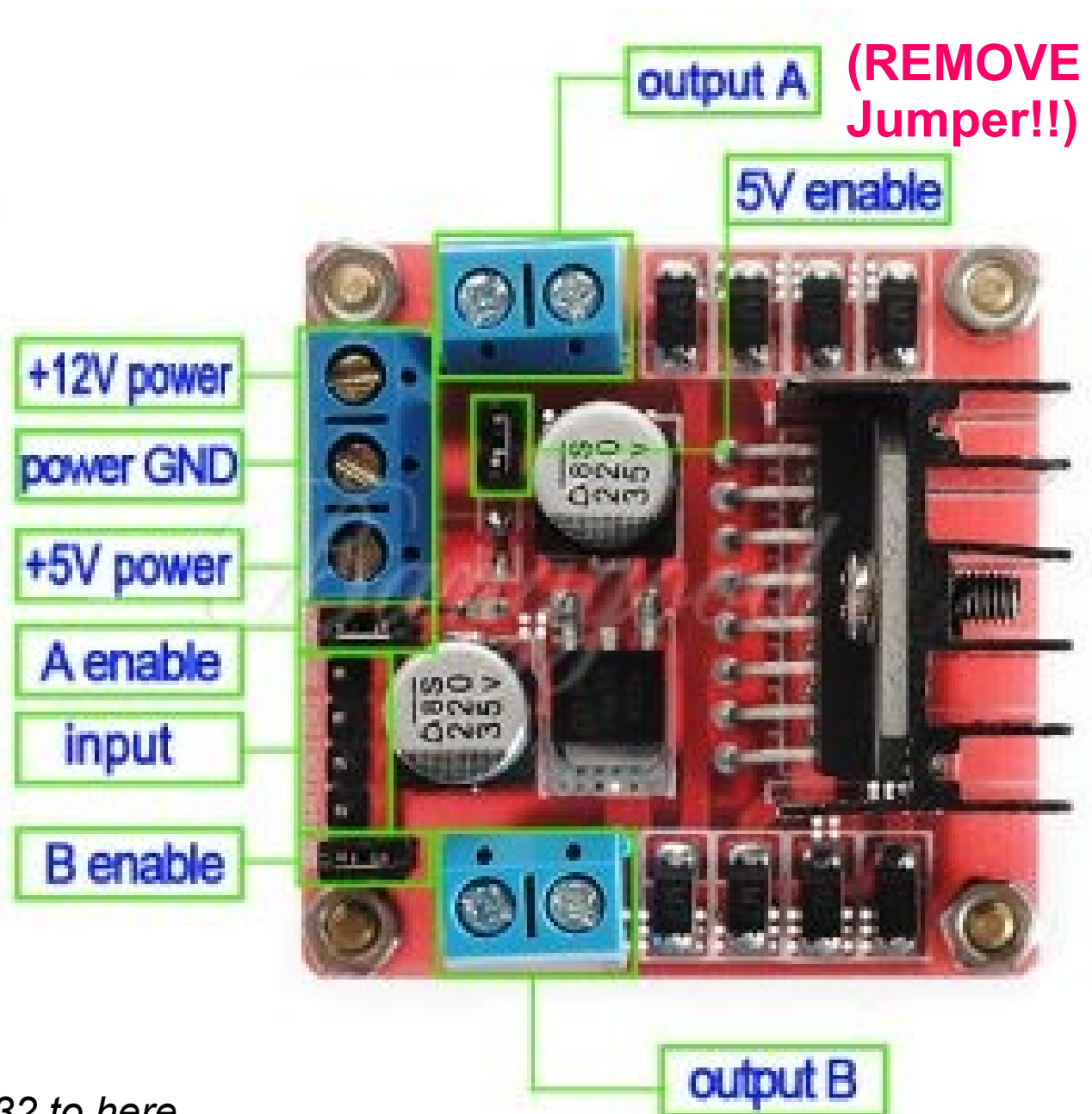
4.5volt battery pack  
(3 AA cells)  
Different connections:

“+12V” = Battery Pack positive  
“power GND” = Battery Gnd  
and GND from ESP32  
“+5V” = 5V from ESP32

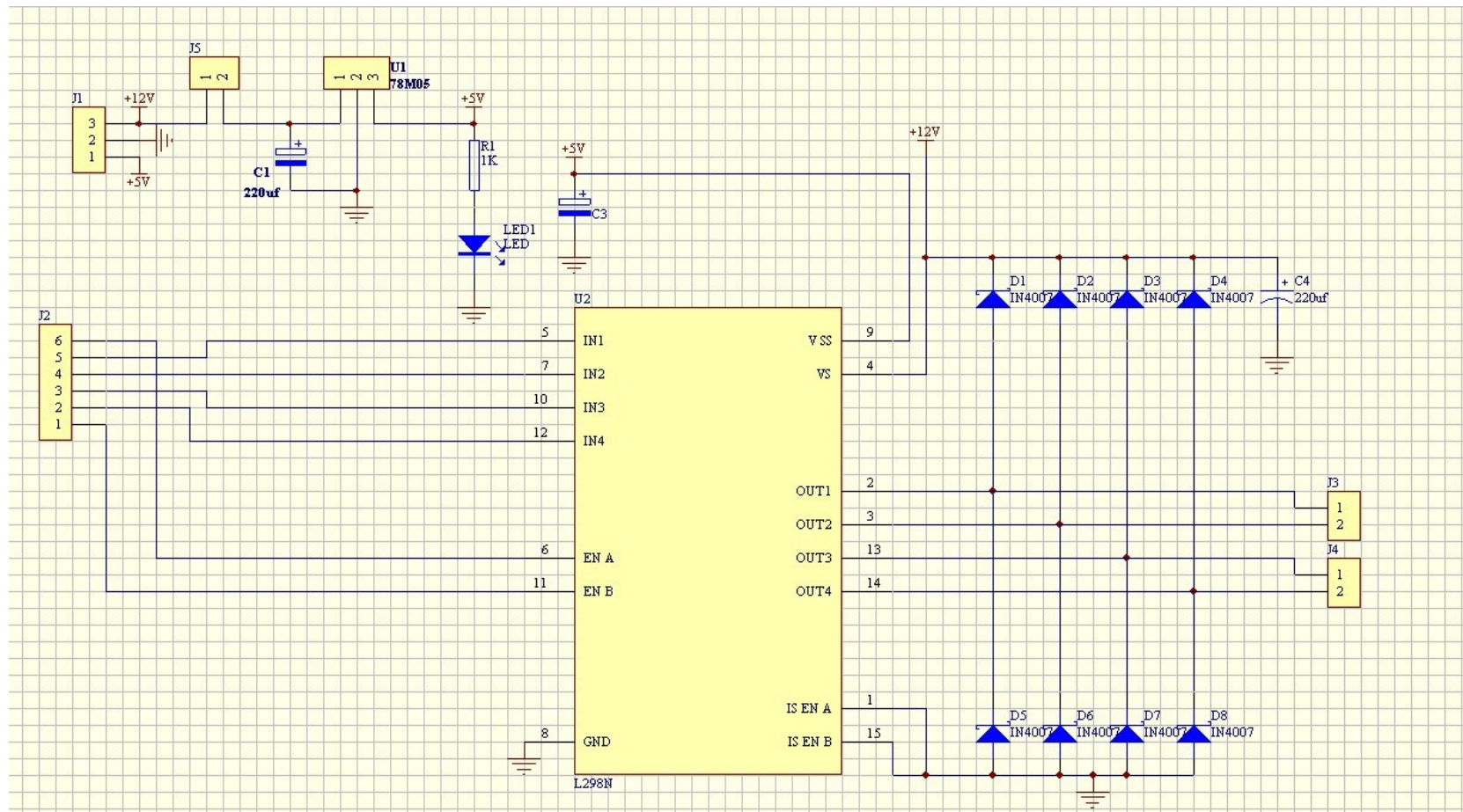
A/B “enable” = PWM  
for speed control.  
(Remove these 2 jumpers)

In1/2 and In3/4 = run/direction  
for motors A/B

Note there *is* a GND from ESP32 to here,  
but **AND** 5V from ESP32 to here.



Your 2 motors A/B  
(at 4.5V)



# main Library calls:

(See L298N.h)

Constructor: L298N motor1(Aen, Ain1, Ain2);

motor1.setSpeed(pwmval);

motor1.getSpeed();

motor1.run(direction);

motor1.forward(); or .backward()

motor1.forwardFor(time); backwardFor(...)

motor1.stop();

motor1.isMoving();

# exercise

Start with Hbridge\_blynk\_0.ino

and build it up to

Hbridge\_blynk.ino

Add in the L298N  
Hbridge calls