



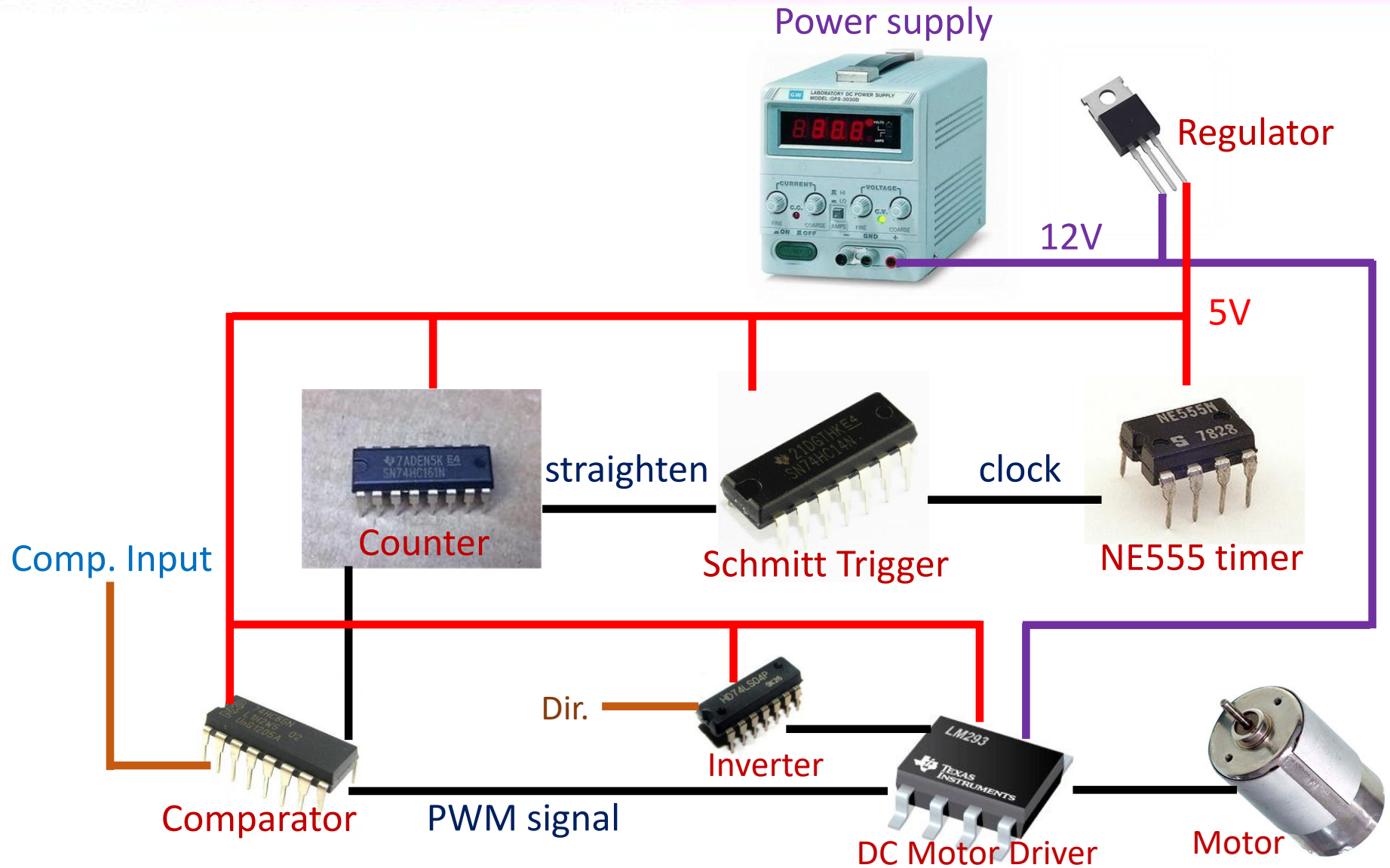
USER ACCESS
NAME ATTACHED TO HIS ID
ACCESS APPROVED
DNA CONTAINS THE GENETIC INFORMATION
ALLOWING ALL MODERN LIVING ...
1 BILLION YEARS SCAN DONE
ACCESS DENIED
RETURN ACCESS SCAN
ACCESS APPROVED, WELCOME
ENTER YOUR PASS CODE ...
XXXXXXXXXX

ELEC1100 - Tutorial 6

Sensors & MCU

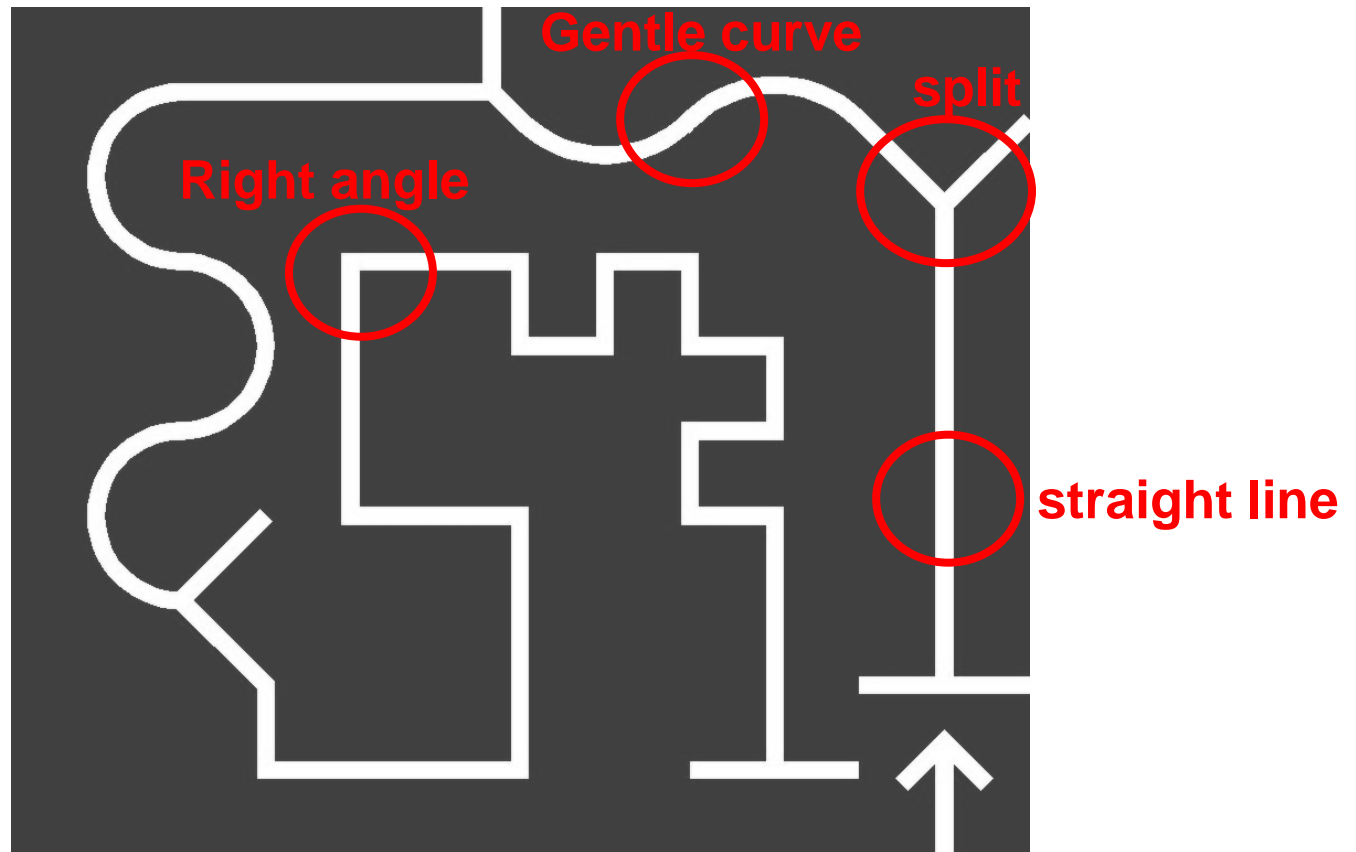
Introduction to Lab#05

Review

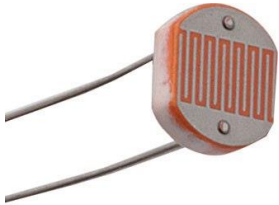


Your Demo Track

- ❖ Use **sensors** to detect changes in the track, then change the speed/DIR signals accordingly.



Photoresistor in Tinkercad



LDR

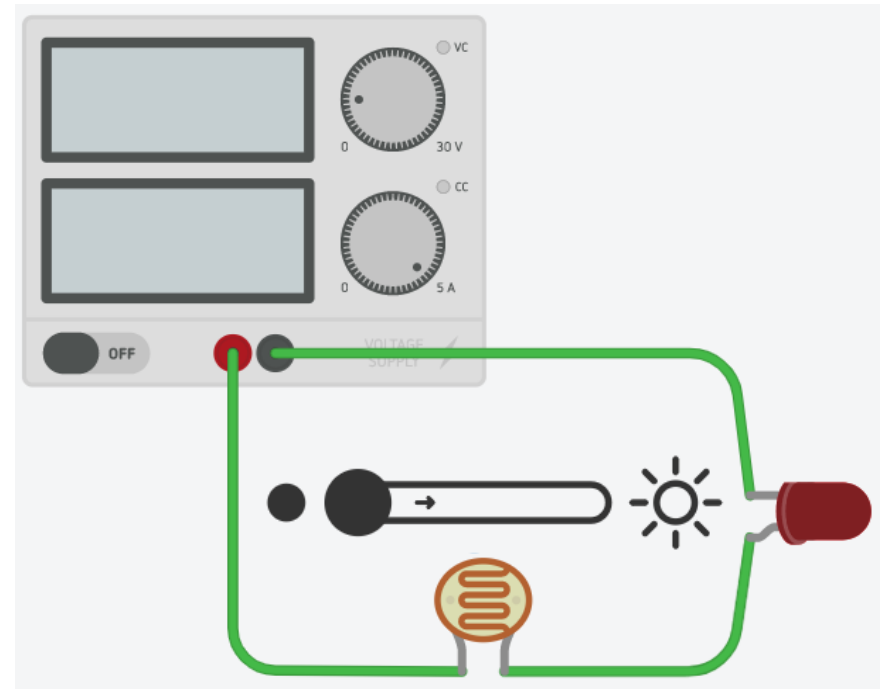
The real component



In Tinkercad

- ❖ Observe the LED brightness changes according to the changes on lightening condition.

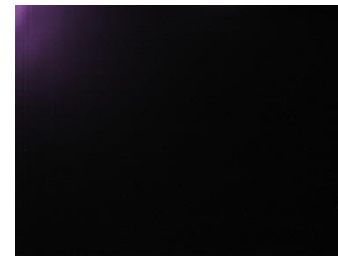
Lab#05: Simulation 1



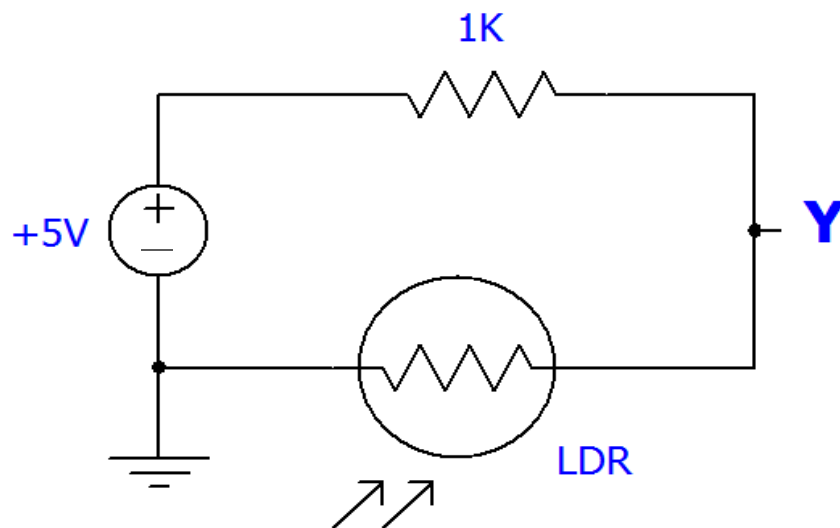
- “Light Dependent Resistor”



R low



R high



Example:

$R = 1k$ when light

$R = 2k$ when dark

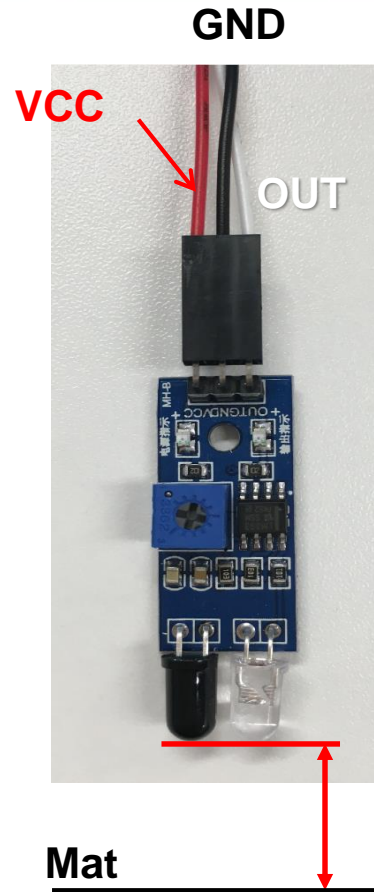
$Y = ?$

Sensor Connection

❖ At Physical lab

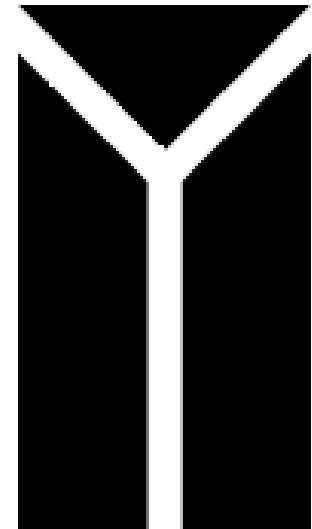
Light sensor connection

- **Red: 5V**
- **Black: GND**
- **White: Sensor Output Voltage**



Sensor output voltage

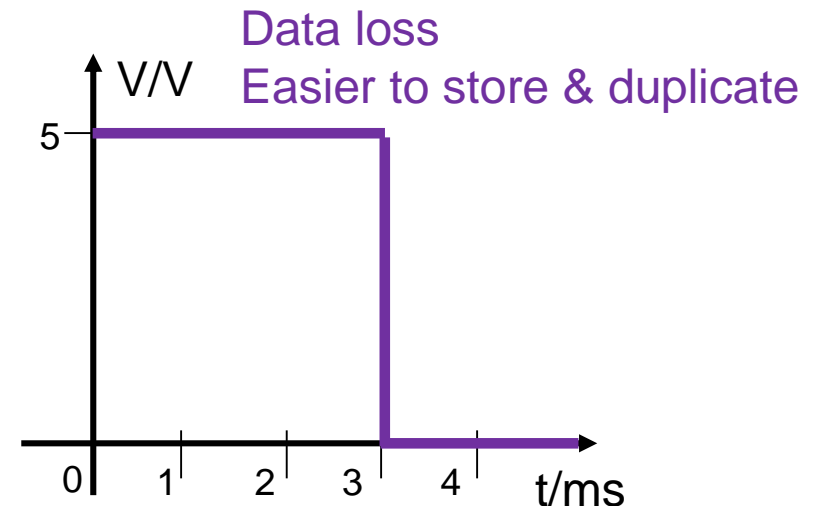
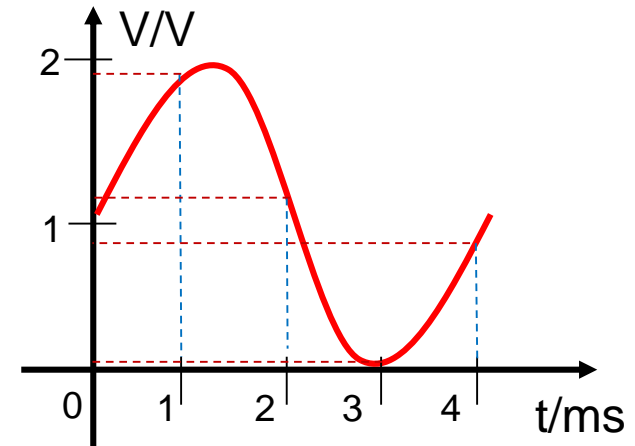
- White: Low (0V)
- Black: High (5V)



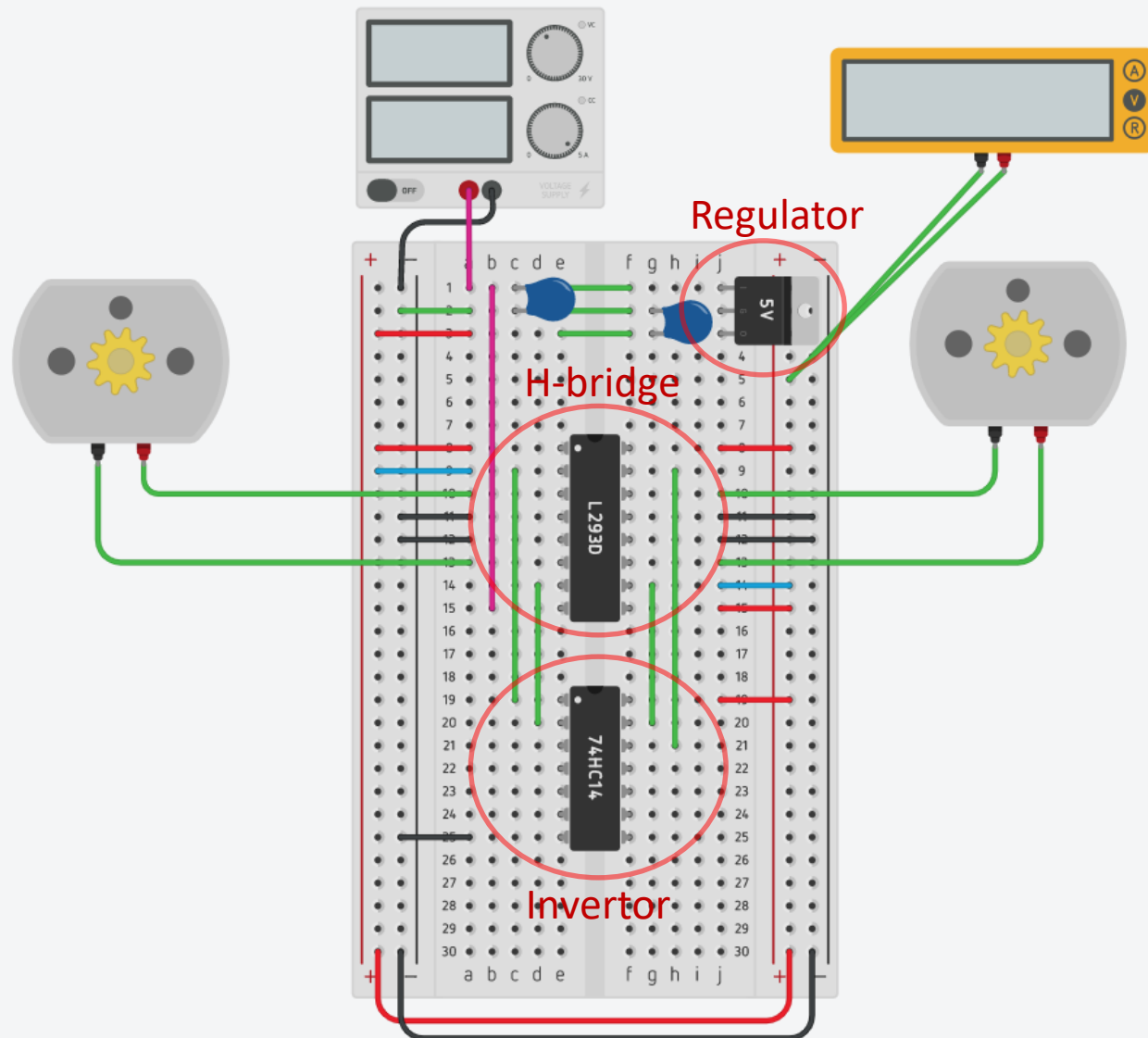
A black Mat with white lines on it

Data Transmission

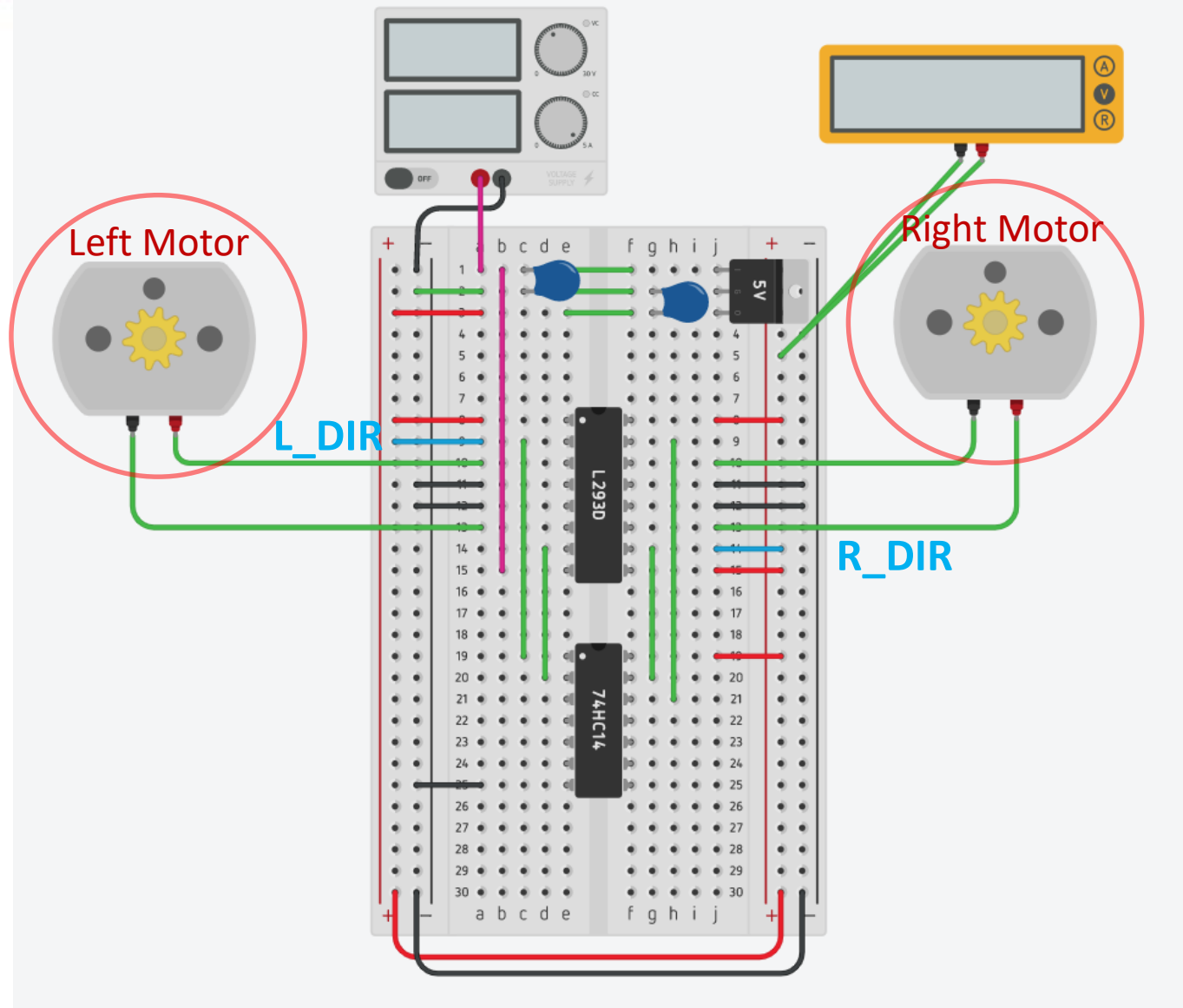
- ❖ Transmit this signal:
 - Convert to binary first
 - E.g. Sample every 1ms
 - E.g. only 2 levels:
 - $V > 1V \Rightarrow \text{"1"} (5V)$
 - $V < 1V \Rightarrow \text{"0"} (0V)$
 - Classify each sample into 1/0



Review: Circuits on your breadboard

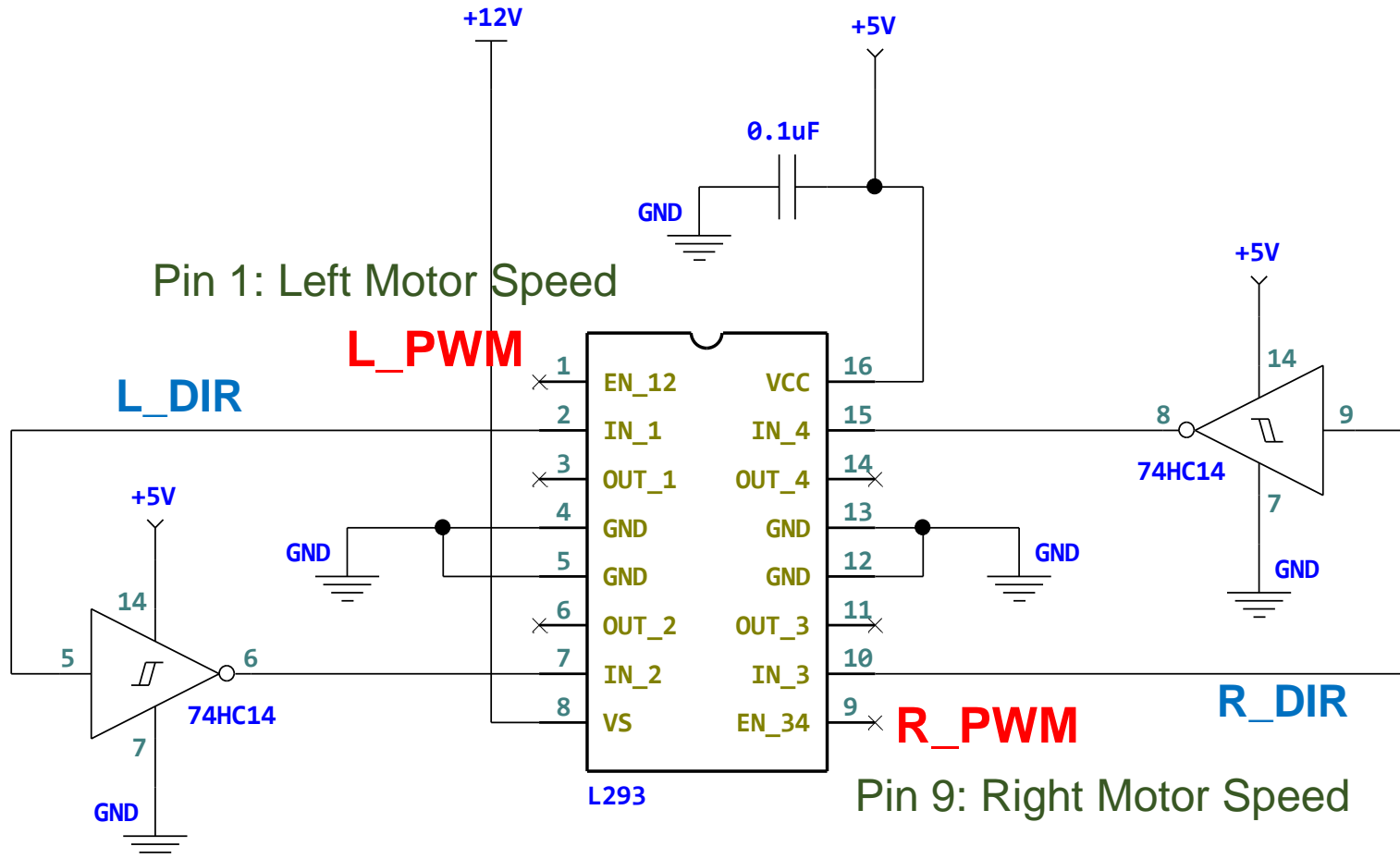


Review: Motor Rotation Control



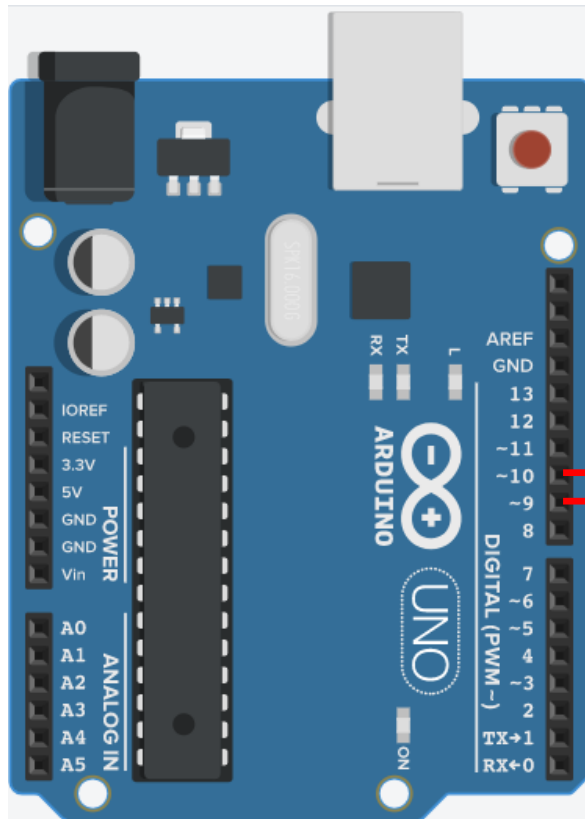
Motor Speed Control

❖ PWM: Speed control signal to your motors



PWM Control using Uno-board

BOARD	PWM PINS	PWM FREQUENCY
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)



L_PWM

R_PWM

❖ In your Lab#05:

Use Pin10 & 9 of your Uno-board to generate PWM signals to control the speed of your left & right motors.

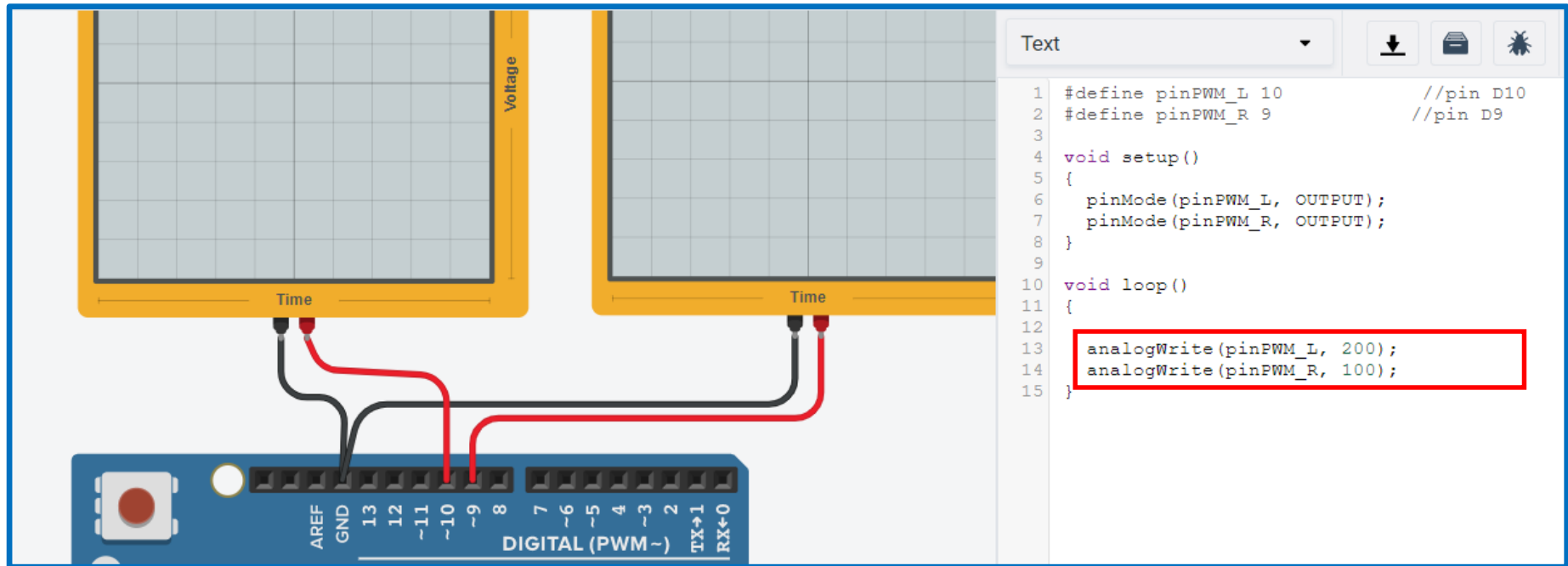
Arduino Coding

❖ **analogWrite():** Write an analog value (PWM wave) to a pin.

After a call to `analogWrite()`, the pin will generate a steady rectangular wave of the specific duty cycle.

```
analogWrite(pin, value)
```

Value: the duty cycle, between 0 (always off) and 255 (always on)



The image shows a screenshot of an Arduino IDE interface. On the left, a wiring diagram illustrates an Arduino Uno connected to two oscilloscopes. The oscilloscope on the left is labeled 'Voltage' and 'Time', and the one on the right is labeled 'Time'. Wires connect the Arduino's digital pins to the oscilloscopes: a black wire from GND to the common ground of both scopes, a red wire from pin 10 to the input of the left scope, and another red wire from pin 9 to the input of the right scope. On the right side of the IDE, the code editor shows the following code:

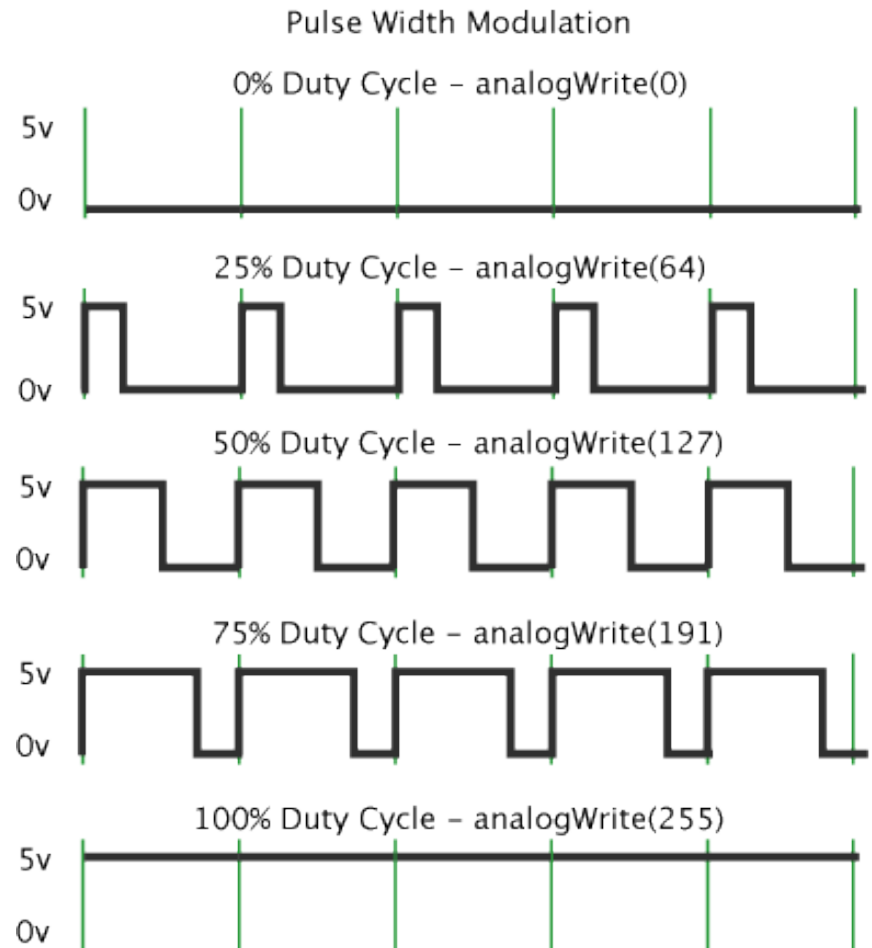
```
1 #define pinPWM_L 10           //pin D10
2 #define pinPWM_R 9           //pin D9
3
4 void setup()
5 {
6   pinMode(pinPWM_L, OUTPUT);
7   pinMode(pinPWM_R, OUTPUT);
8 }
9
10 void loop()
11 {
12   analogWrite(pinPWM_L, 200);
13   analogWrite(pinPWM_R, 100);
14 }
15
```

The lines `analogWrite(pinPWM_L, 200);` and `analogWrite(pinPWM_R, 100);` are highlighted with a red rectangular box.

Duty Cycle Value

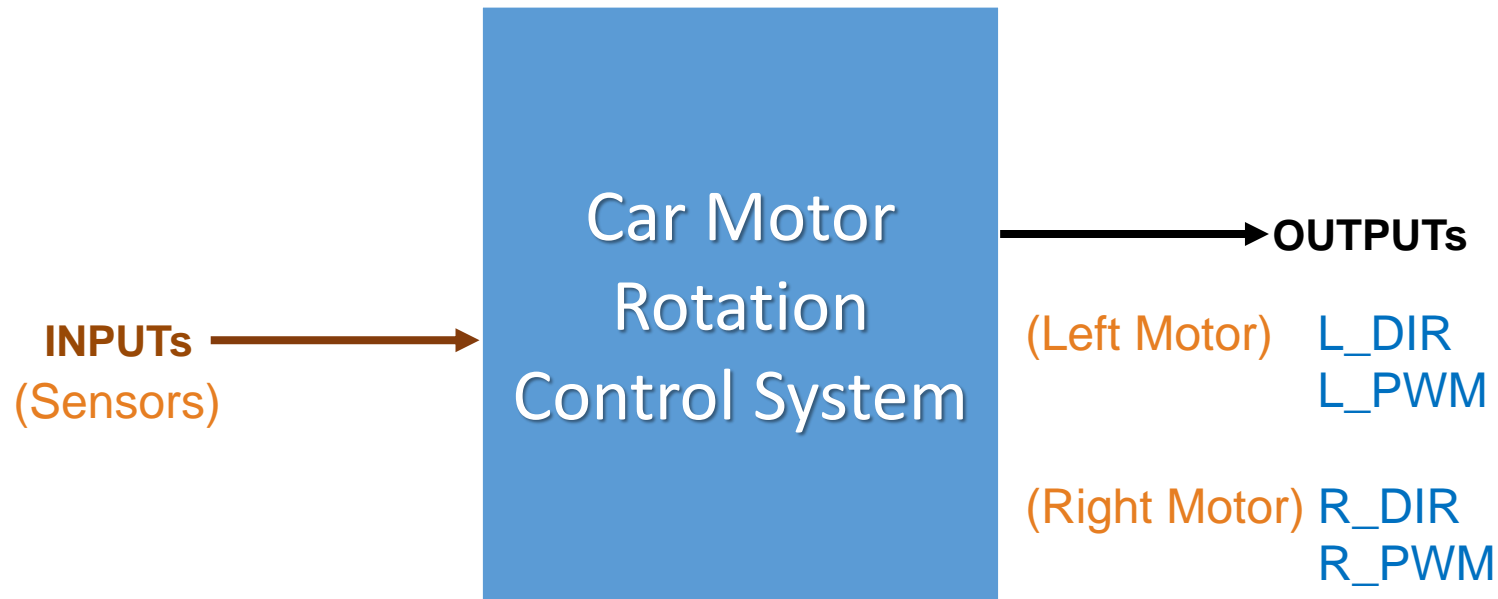
❖ A call to **analogWrite()** is on a scale of 0-255.

- Green lines: a regular time period
- `analogWrite(255)`: 100% duty cycle (always on)
- `analogWrite(127)`: 50% duty cycle (on half the time)



Motor Rotation Control

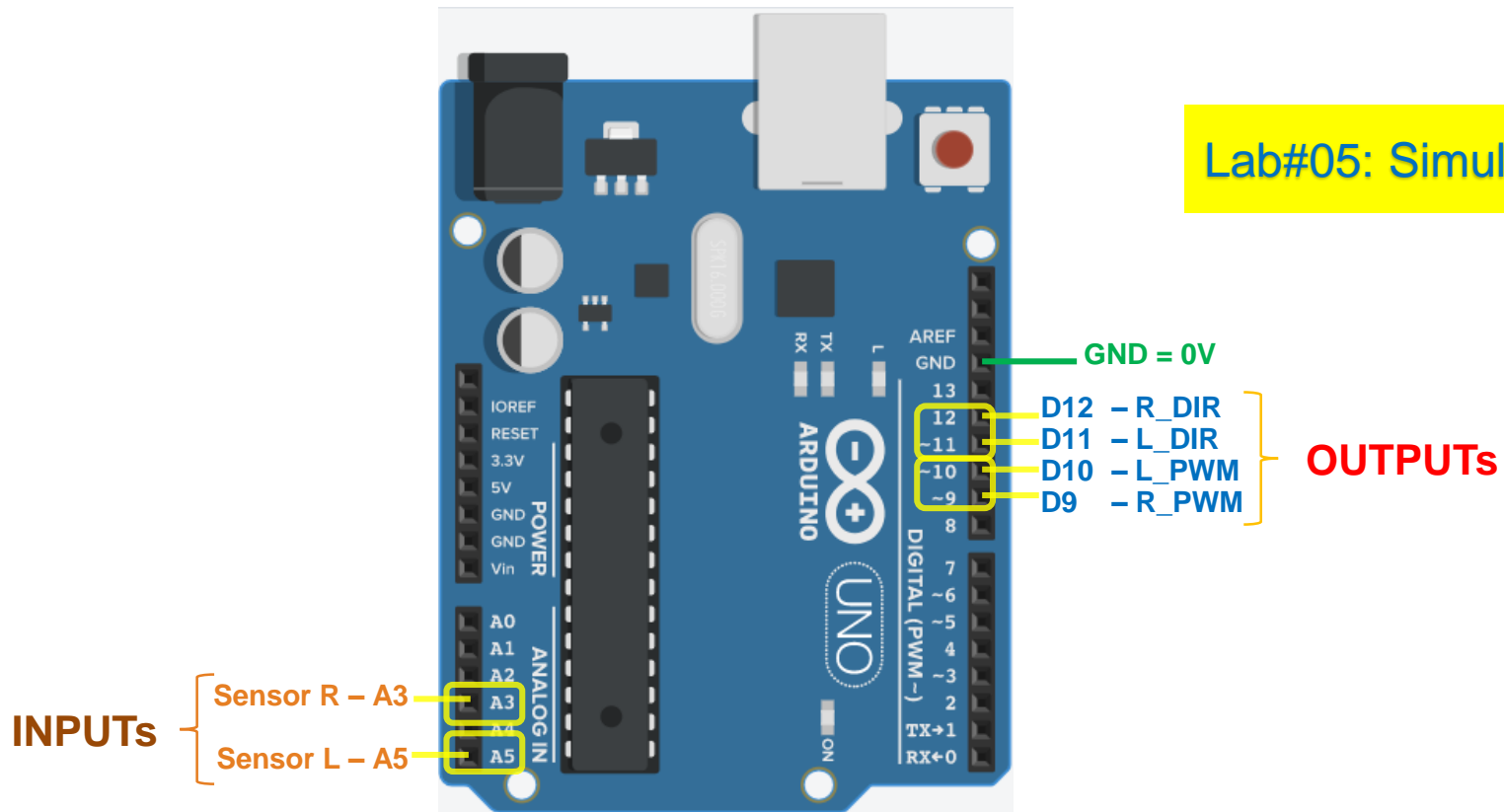
❖ Use sensors to control motor rotation



Motor Rotation Control

❖ Use sensors to control motor rotation

Connect your Uno-board to the H-bridge circuit on your breadboard.

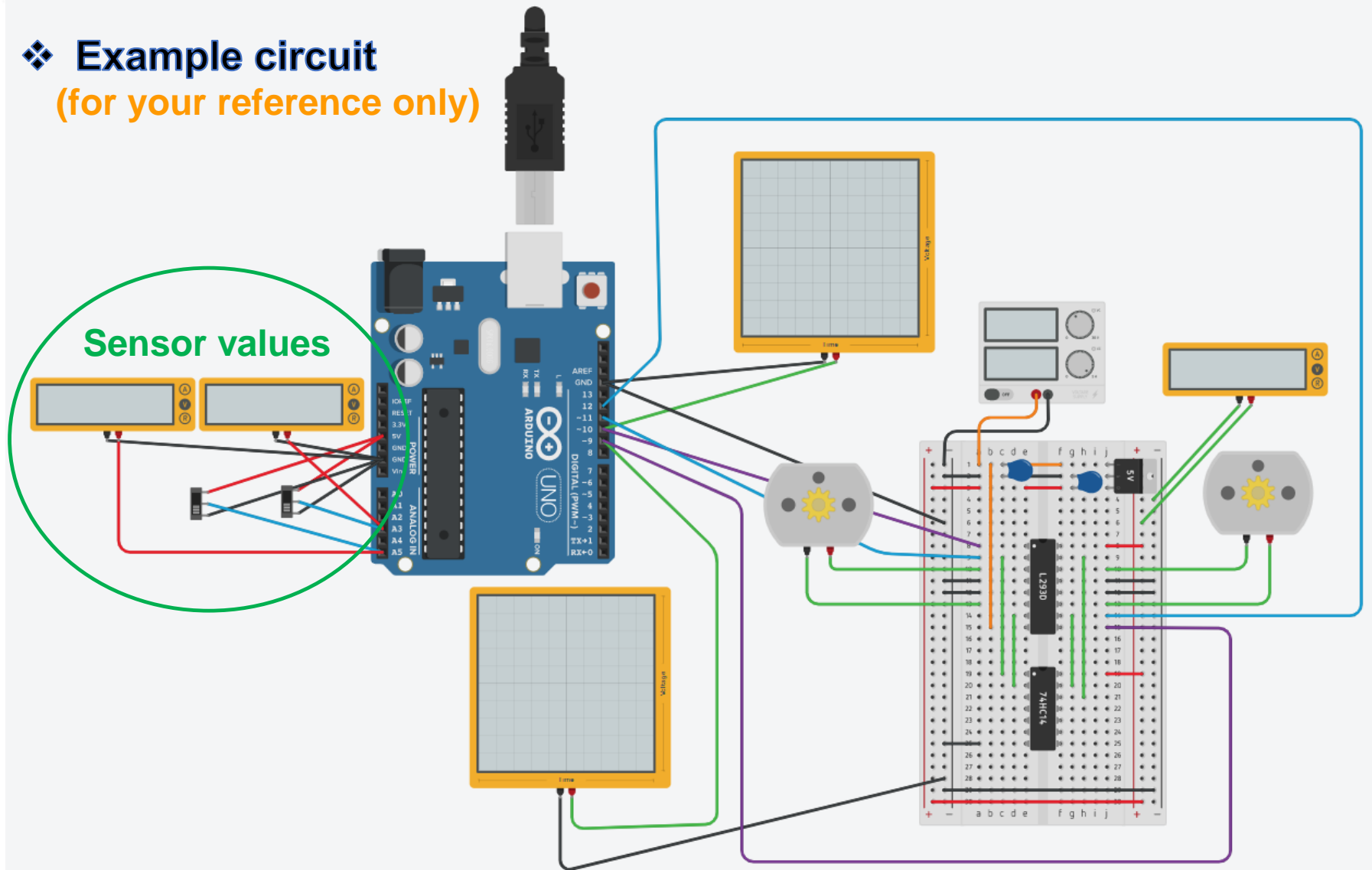


Lab#05: Simulation 4

Download [Lab#05 Arduino sketch](#), copy & paste into your Tinkercad coding text.
(for [Lab#05](#) only, we will do *logic design* to control DIR signals at [Lab#06](#))

Motor Rotation Control

❖ Example circuit (for your reference only)

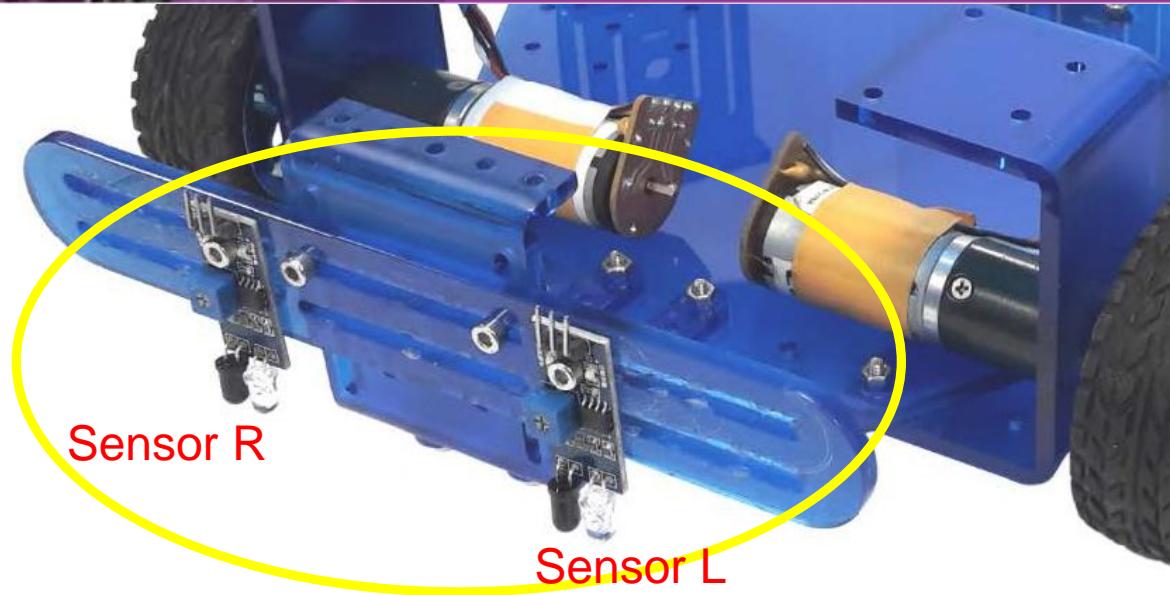


Sensor Value Simulation

❖ Sensors on your car

Measure the surface:

- White: Low (0V)
- Black: High (5V)



❖ Tinkercad Simulation

Two switches:

- Sensor R value (A3)
- Sensor L value (A5)

