PRACTICE 2: MODERN IOT

1 Analog vs Digital

The first demonstration to compare analog and digital function is proposed in the circuit below, with 2 different LEDs, connected to pin number 2 and 3.

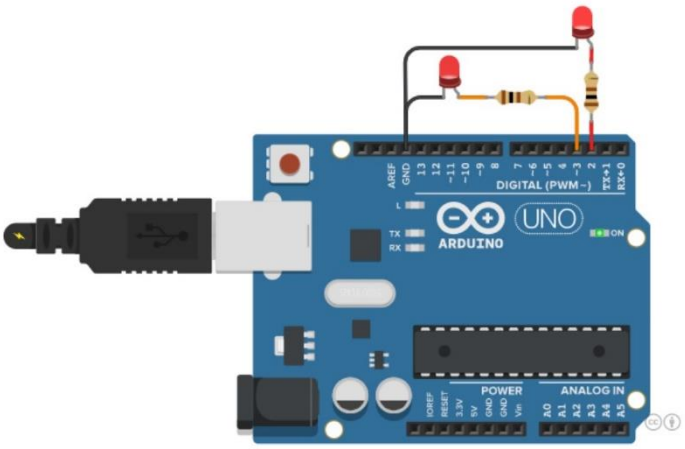


Figure 2.1: *Analog vs Digital Testing*

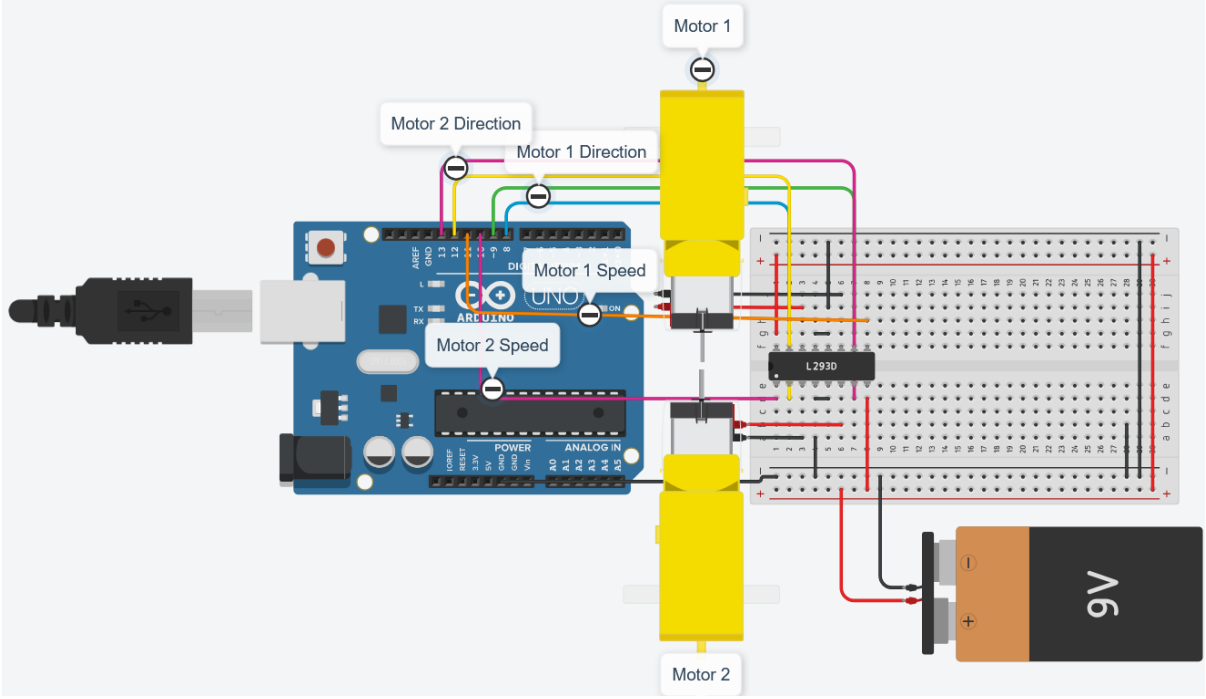
The brightness of the LED connected to pin number 3 is increased from the minimum to the maximum, which can be compared to the brightness of the LED connected to pin number 2. This LED is set to the maximum brightness by using the digitalWrite function. The source code of this demo is shown below:

```
void setup(){ pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    digitalWrite(2, HIGH);
}

void loop(){
    for(int i = 0; i < 255; i += 5){ analogWrite(3, i);
        delay(100);


    }
}
```

Although the animation in TinkerCad is not easy to observe, students can find the principle of the analogWrite. This can be used to control some devices such as a motor (speed) or a buzzer.

2 Motor Controller

From this section, students are proposed to construct their programs to control 2 motors, which are normally used for robot movement. The schematic of the simulation is proposed as follows.



2.1 Left Motor Controller

To control the left motor, following pins are required:

- Pin 8: Direction configuration pin 1 by **dititalWrite**
- Pin 9: Direction configuration pin 2 by **digitalWrite**
- Pin 11: Speed controller pin by **analogWrite**

Students are proposed to implement following program to test the first motor.

```
void    setup() {
        pinMode(8, OUTPUT); pinMode(11,
        OUTPUT); pinMode(9, OUTPUT);

}
void    loop() {
        digitalWrite(11, HIGH); digitalWrite(8, LOW);
        analogWrite(9, 100);
        }
```

Students can change the parameter of the **analogWrite** function. The parameter is in the range **from 0 to 255**.

Finally, students implement following function to control the left motor.

```
void    left_speed(int speed){
        //TODO: Implement your code                here
        if(speed > 0){

        }else{
                speed = 0 - speed;
        }
}
```

2.2 Left Motor Testing

Implement a short program to make the left motor move forward with speed 100 for 2 seconds, then stop for 2 seconds, then move backward with speed for 2 seconds before stopping for 2 seconds again. This behavior is repeated forever. The function **left_speed** is used in this demo.

```
void    loop() {
        //TODO: Implement
}                                   testing       script
```

2.3 Right Motor Testing

Like the left motor, a function to control the right motor is also required. The connection pins for this motor are described as follows:

- Pin 12: Direction configuration pin 1
- Pin 13: Direction configuration pin 2
- Pin 10: Speed controller pin

```
void   loop() {
        //TODO: Implement
}                                   testing       script
```

What does happen if the numer 300 is used in analogWrite function? Briefly provide your answer in the report.

2.4 Full Motor Control Functions

Implement four functions to control the movements of a Robot having 2 motors.

```
void    forward(int speed) {
        //TODO
} void

        backward(int speed) {
        //TODO
} void

        turnleft(int speed) {
        //TODO
} void
}

        turnright(int speed) {
        //TODO
```

2.5 Full Motor Testing

Implement a script in a loop, to test four functions implemented above.

```
void loop(){
        //TODO
}
```

3 Serial Monitor

In data transmission, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This contrasts with parallel communication, where several bits are sent on a link with several parallel channels. serial communication is available in most microprocessor and micro-controller systems, such as the PC, Smart Phone, or the Arduino board.

In this lab, serial communication is used for data transmission between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port (also known as a UART or USART) named Serial. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if you use these functions, you cannot also use pins 0 and 1 for digital input or output. To activate the serial communication, the following source code is required in the setup function.

```
void    setup(){
        Serial.begin(115200);
        Serial.println("Hello TinkerCad”)
}
```

From the TinkerCad terminal, you can send some data to the Arduino board by pressing any charater on your PC keyboard and the click button Send. However, some code needs to be implemented in the Arduino board as follows: the board keeps checking if there is a character sent to it. If there is a character, it will read this character and send it back to the PC terminal.

```
void loop() { if(Serial.available())
    {
        char temp = Serial.read();
        Serial.print("I received: ");
        Serial.println(temp);
    }
}
```

Students propose to run the program again and check the communications between the Arduino board and the TinkerCad terminal.

3.1 LED Controller 1

Implement a short program to make the LED connected to pin 13 turn on if a character **'O'** is received. Other characters, the LED is turned off.

Hint: Use the if statement: if(temp == 'O') .... else ...

3.2 LED Controller 2

Improve your program to make the LED connected to pin 13 turn on if a character **'O'** is received. However, the LED is turned off only when receiving character **'F'**.

3.3 Robot Controller

When characters W, S, A and D are sent, the robot moves forward, backward, left, and right, respectively. With all the other characters, the robot stops moving.