DOT MATRIX DISPLAY

What is our GOAL for this CLASS?

In this class, we learned about the **Led Dot Matrix** unit. We also learn how **LED Dot Matrix** units work. Using this knowledge, we created a project to display data on a display board and used the **MD_Parola** library to add animations.

What did we ACHIEVE in the class TODAY?

- Learned about the **LED Dot Matrix** unit.
- Understood how **LED Dot Matrix** unit works with **MAX7219 controller**.
- Learned to light up a single dot on the **LED Dot Matrix** display.
- Learned to display text on the LED display.

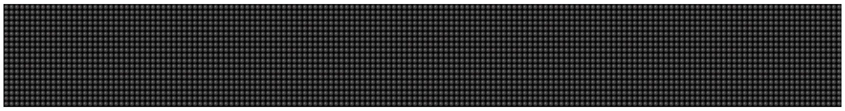
Which CONCEPTS/ CODING BLOCKS did we cover today?

- Concepts: conditional statements. **LED Dot Matrix** unit, **MAX7219 controller**.
- Coding blocks: **if** statements, functions from **MD_MAX72xx** and **MD_Parola** library.

How did we DO the activities?

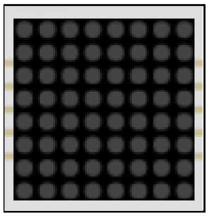
1. Learned about **8x8 LED Dot Matrix display**:

   8x8 LED Dot Matrix is used a lot in real life. The information boards on buses and metros are usually made up of LED Dot Matrix units. These are used to display scores during sports as well.



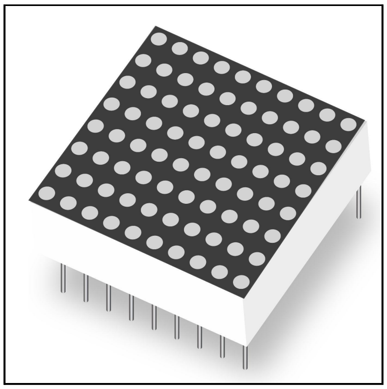
   You can view the example video here.

   An **8x8 LED Dot Matrix** display looks something like this-



   There are 8x8 LEDs i.e. 64 LEDs here. These LEDs can be lit up to form a character/ digit on the display.

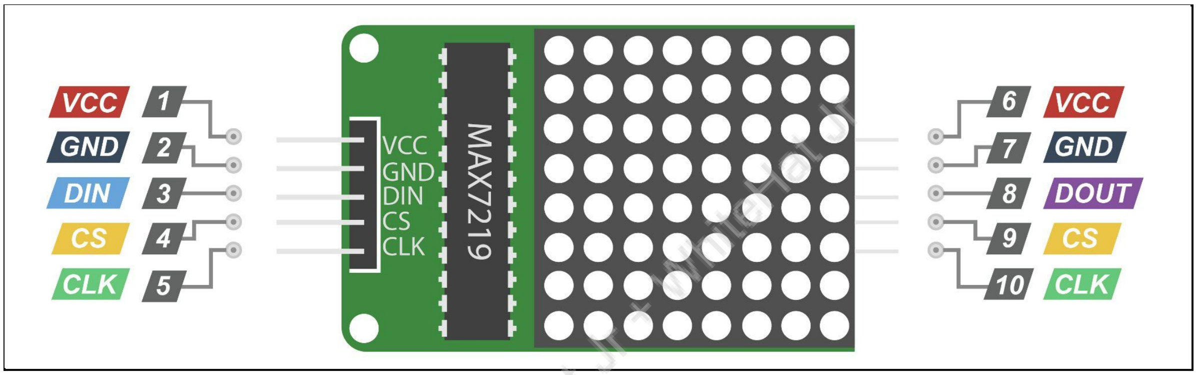
   An **8x8 Dot Matrix** unit has 16 pins. 8 for each row and 8 for each column. To reduce the number of pins, the LEDs in each row and each column are connected together.



2. Learned about **MAX7219** controller:

   But 16 pins are also a lot to connect with an Arduino Uno board. It requires a lot of connections. Also, it needs to constantly refresh the display to keep the image stable.

   To resolve this problem, the **LED Dot Matrix** display comes with a **MAX7219** controller. The **MAX7219** controller accepts data and can drive up to 64 LEDs.



3. Understood the pins of **LED Dot Matrix** display with **MAX7219** controller:

   Let's observe the **LED Dot Matrix** display with **MAX7219** controller. Observe the pins on the **right-hand side** first.

Pin Name	Description
VCC	Voltage supply
GND	Ground
DIN	Data input
CS	Chip Select
CLK	Clock input

   There are similar pins and a **DOUT** pin on the left-hand side to daisy-chain more LED Dot Matrix units to make the display larger. Daisy-chaining is a way of connecting electronic components with each other.

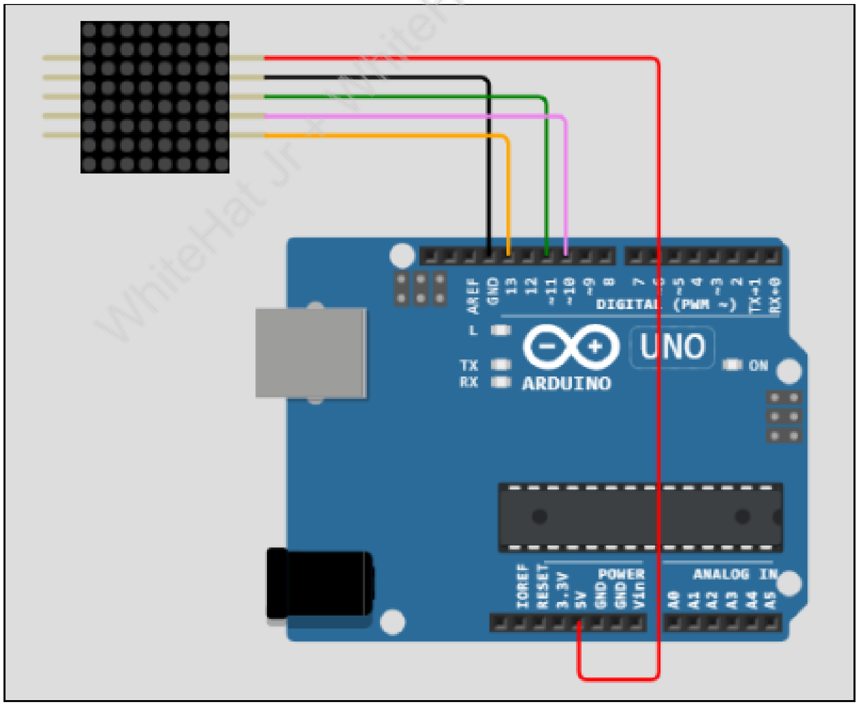
4. Create the circuit:

- Select the components
  - 1 x **Arduino Uno board**
  - 1 x **LED Dot Matrix with MAX7219 Controller**

- Let’s connect:

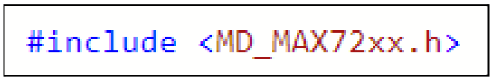
Arduino Uno board	LED Dot Matrix Pin Number
5V	VCC
Ground	GND
GPIO 11	DIN
GPIO 10	CS
GPIO 13	CLK

Reference circuit:



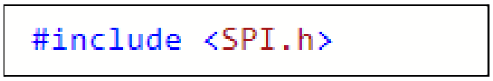
5. Code:

- First, go to **sketch.ino.** Add the header files there-

  - 

    ```
    #include <MD_MAX72xx.h>
    ```

    MD_MAX72xx library implements functions that allow the **MAX72xx** (eg, MAX7219) to be used for LED matrices (64 individual LEDs). It also allows the programmer to use the LED matrix as a display where each LED pixel can be addressed. Pixel stands for picture element.

  - 

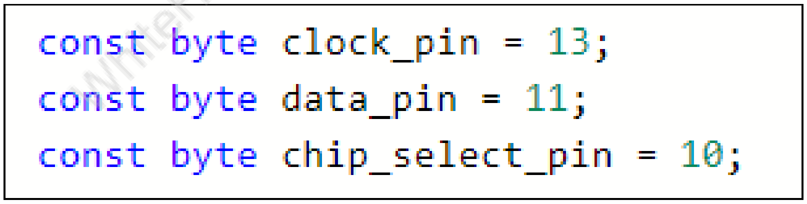
    ```
    #include <SPI.h>
    ```

    SPI library allows you to communicate with other devices, with the Arduino as the master device.

    When humans communicate, there are some rules which we refer to as grammar. This helps us to communicate meaningfully so that others can understand us.

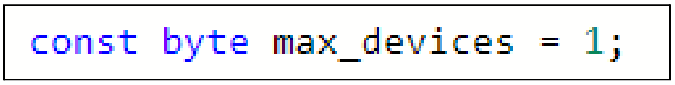
    Similarly, **Serial Peripheral Interface (SPI)** is a set of communication rules which the electronic components have to follow in order to communicate successfully.

- Now, initiate variables that hold the pin numbers that connect the Dot Matrix display with the **Arduino Uno** board.



```
const byte clock_pin = 13;
const byte data_pin = 11;
const byte chip_select_pin = 10;
```

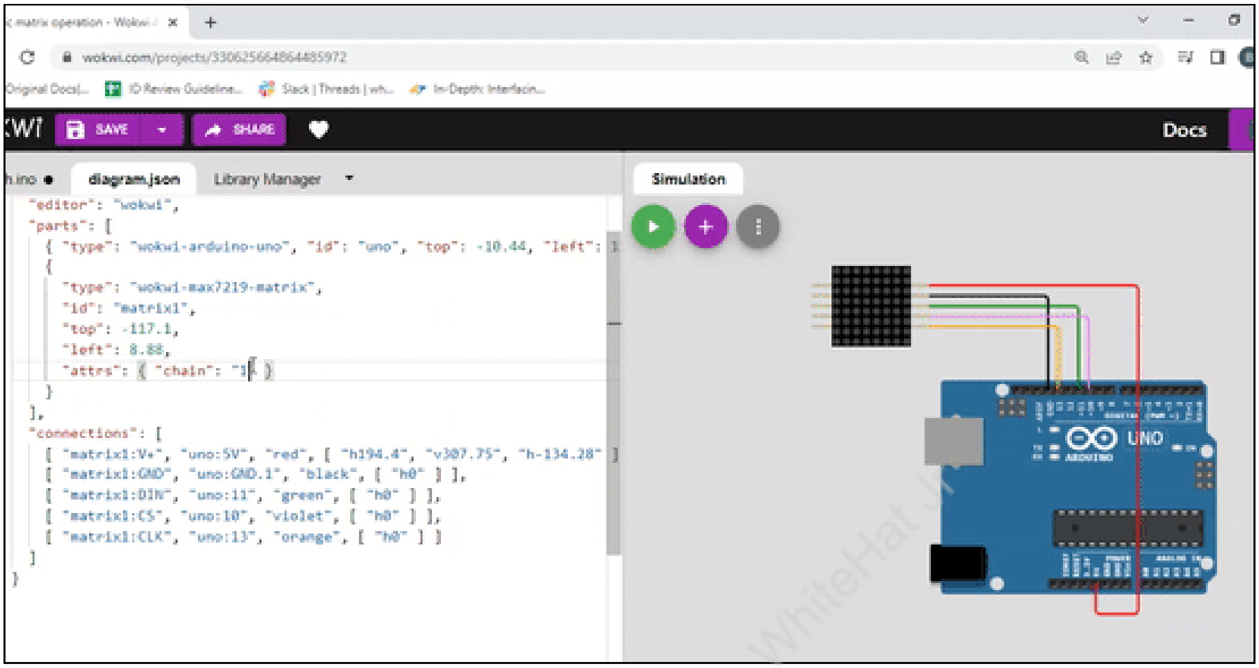
- Initiate a variable named **max_devices** which will hold the number of displays we will be using. We will keep it as 1 for now.



```
const byte max_devices = 1;
```

The number of devices can be increased in the following manner-

Go to **diagram.json** → find **wokwi-max7219-matrix** under "**parts**" → find the **chain** attribute → change the value. Keep the value as 1.



```
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-arduino-uno", "id": "uno", "top": -10.44, "left": 1
    {
      "type": "wokwi-max7219-matrix",
      "id": "matrix1",
      "top": -117.1,
      "left": 8.88,
      "attrs": { "chain": "1" }
    }
  ],
  "connections": [
    [ "matrix1:V+", "uno:5V", "red", [ "h194.4", "v307.75", "h-134.28" ]
    [ "matrix1:GND", "uno:GND.1", "black", [ "h0" ] ],
    [ "matrix1:DIN", "uno:11", "green", [ "h0" ] ],
    [ "matrix1:CS", "uno:10", "violet", [ "h0" ] ],
    [ "matrix1:CLK", "uno:13", "orange", [ "h0" ] ]
  ]
}
```

Click here to view the reference video.

- Now, let's create an instance-

```
MD_MAX72XX matrix = MD_MAX72XX(MD_MAX72XX::PAROLA_HW, chip_select_pin, max_devices);
```

- Now, let's write the following code inside the **setup()** function.
  - Initiate the module.

```
matrix.begin();
```

  - Clear the Dot Matrix display, if there is anything –

```
matrix.clear();
```

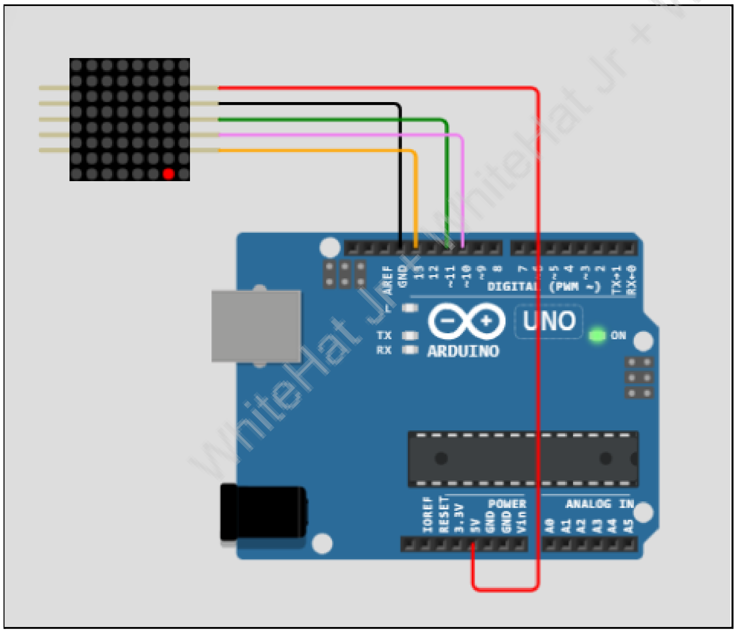
- Now, we have 64 LEDs which we can light up. Use the **setPoint()** function to light up individual LEDs on display.

  Syntax: **setPoint(row, column, status)**

The rows and columns are numbered in the following way for an **8x8 LED Dot Matrix** unit.

	c7	c6	c5	c4	c3	c2	c1	c0
r0	(0,7)	(0,6)	(0,5)	(0,4)	(0,3)	(0,2)	(0,1)	(0,0)
r1	(1,7)	(1,6)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)
r2	(2,7)	(2,6)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)
r3	(3,7)	(3,6)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)
r4	(4,7)	(4,6)	(4,5)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)
r5	(5,7)	(5,6)	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)
r6	(6,7)	(6,6)	(6,5)	(6,4)	(6,3)	(6,2)	(6,1)	(6,0)
r7	(7,7)	(7,6)	(7,5)	(7,4)	(7,3)	(7,2)	(7,1)	(7,0)

Let's observe the output-

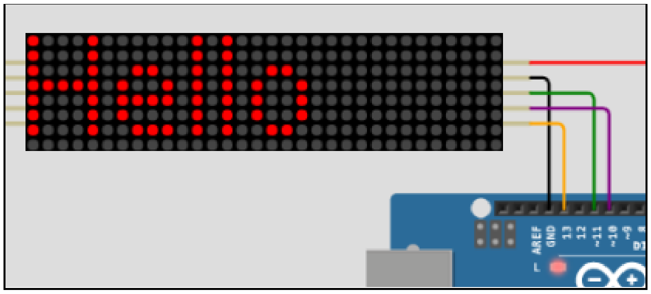


6. Let's try to print "**hello**" on the display now.

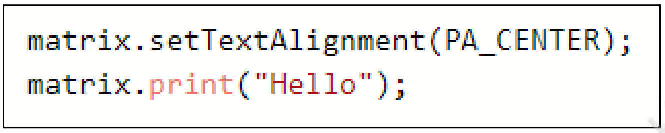
- Use the function **print()** for it. Write this function inside the **loop()** function.

```
matrix.print("Hello");
```

Run the code and observe the output.

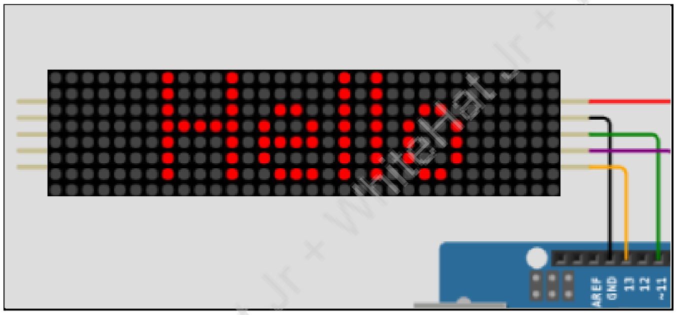


- Change the alignment as well by using the **setTextAlignment()** function.

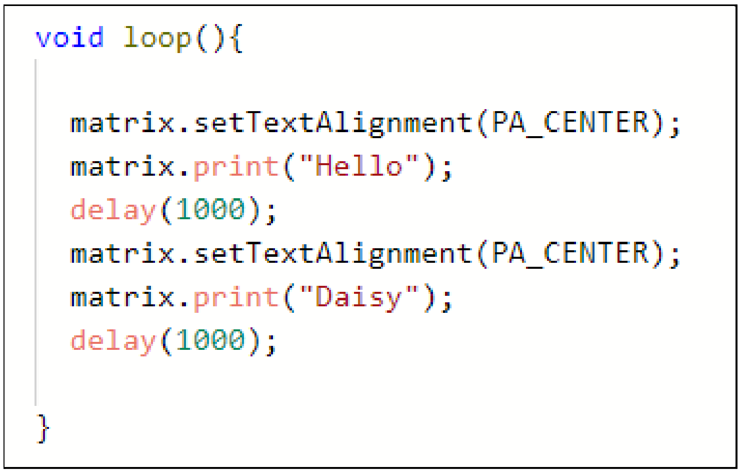


```
matrix.setTextAlignment(PA_CENTER);
matrix.print("Hello");
```

Run and check the output.

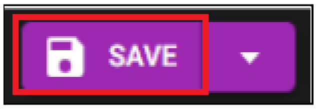


- Show hello for a 1 second and after that your name for 1 second. Use the **delay()** function in between and pass 1000 milliseconds (= 1 second) through the function.

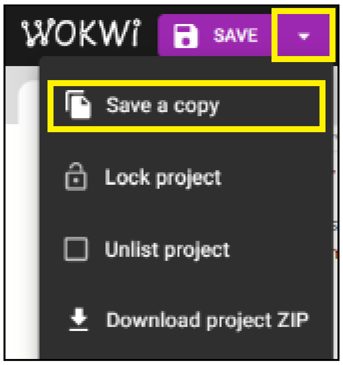


```
void loop(){

  matrix.setTextAlignment(PA_CENTER);
  matrix.print("Hello");
  delay(1000);
  matrix.setTextAlignment(PA_CENTER);
  matrix.print("Daisy");
  delay(1000);

}
```

7. Let's save the code till now. By clicking on the save button at the top.



Once, you have saved it. Make a copy of this file to do further experiments without losing the current code.



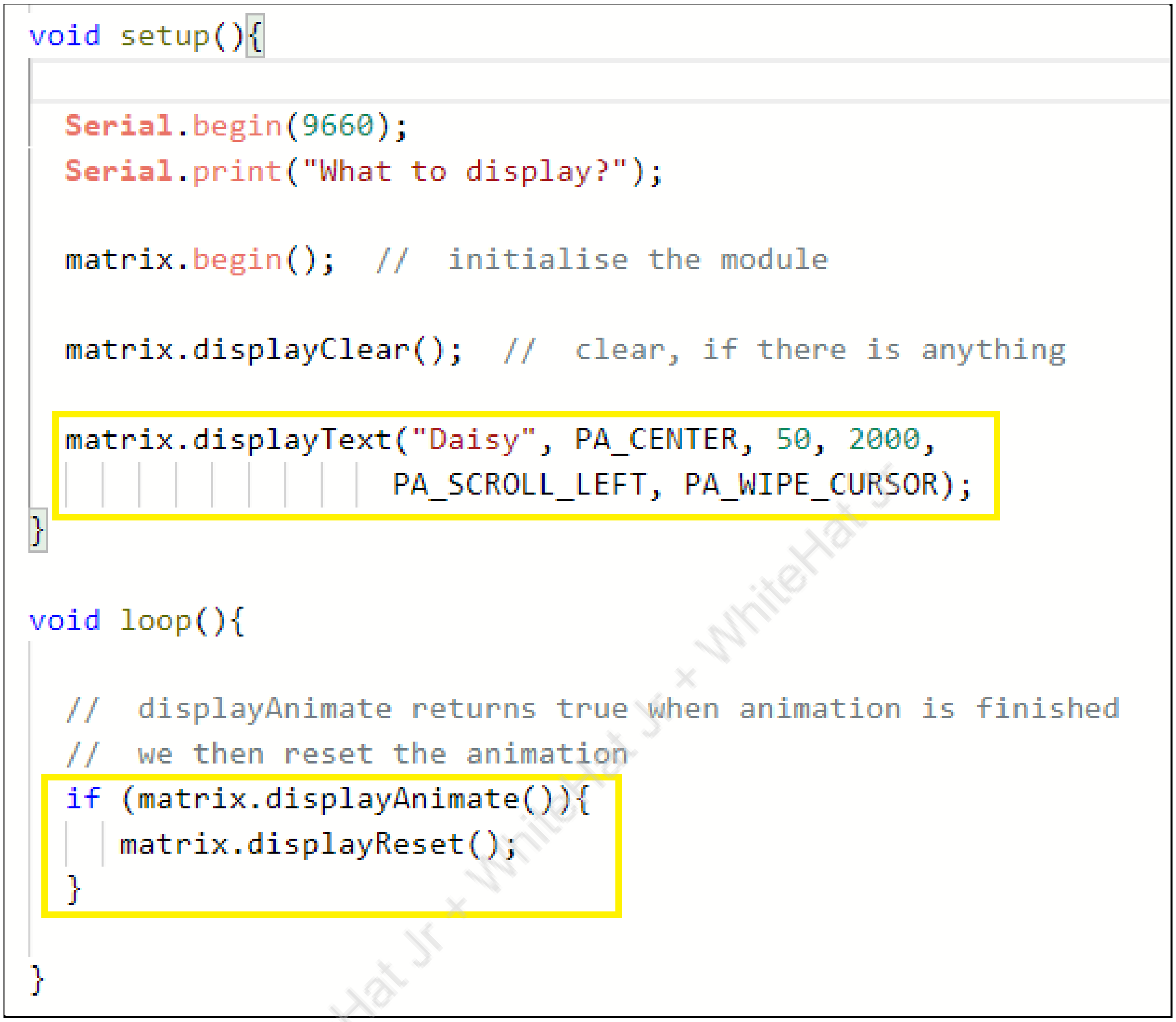
8. Now, use the **displayText()** method from the MD_Parola library.

   This method is a convenient way to set up a text display. All the data necessary for setup is passed through as parameters. Display animation can be added too.

   It has 6 parameters. The 6 parameters of the **displayText()** function are :

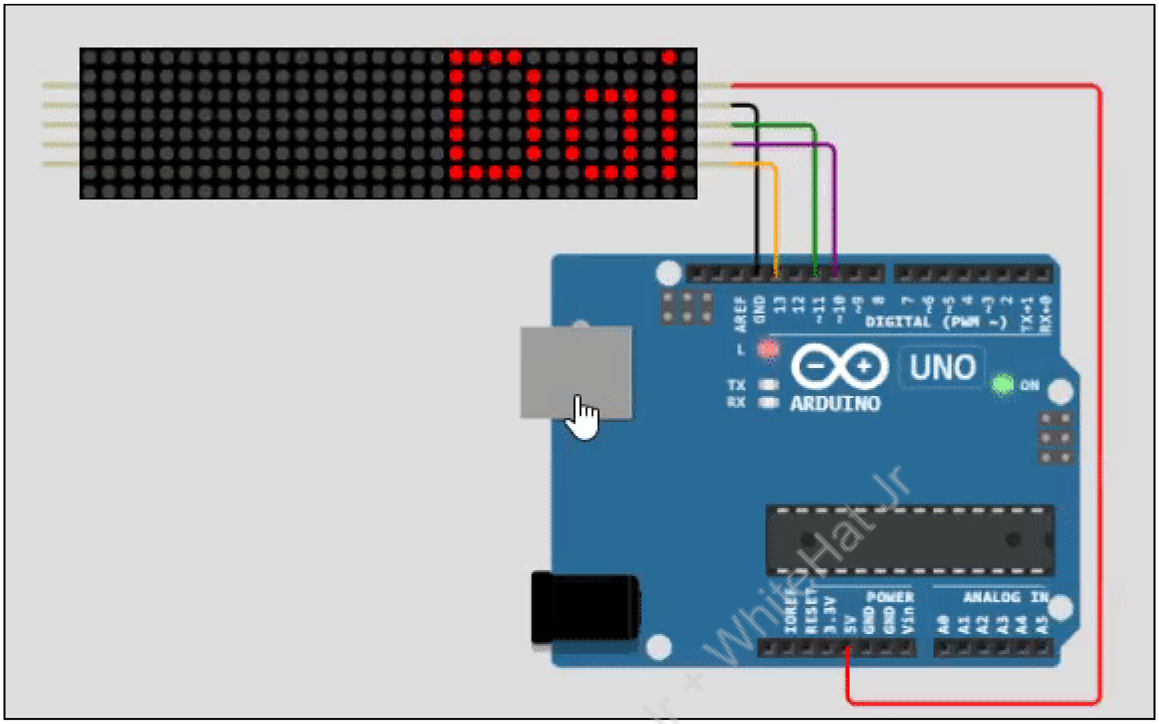
   - text to be displayed
   - alignment
   - time interval between frames (scrolling speed). If you increase this number the speed will decrease.
   - pause between animation in and out
   - effect in
   - effect out

     View the list of suitable effects for **effects in** and **effects out** here.



```
void setup(){

  Serial.begin(9660);
  Serial.print("What to display?");

  matrix.begin();  //  initialise the module

  matrix.displayClear();  //  clear, if there is anything

  matrix.displayText("Daisy", PA_CENTER, 50, 2000,
                  PA_SCROLL_LEFT, PA_WIPE_CURSOR);
}

void loop(){

  //  displayAnimate returns true when animation is finished
  //  we then reset the animation
  if (matrix.displayAnimate()){
     matrix.displayReset();
  }

}
```

Reference Output:



Click here to view the reference video.

What's NEXT?

In the **next class**, we will learn about the **joystick** component. We will learn how to use it along with a **Dot Matrix display**.

Expand Your Knowledge

To know more about the **LED Dot Matrix** unit on wokwi**,** click here.