

Task. 1. Implement the following code. Use *Serial Monitor* to check if it works.

```
#define BAUDRATE 115200

char serial_data;

void setup() {
  Serial.begin(BAUDRATE); }

void loop() {
  if(Serial.available()>0) {
    serial_data = Serial.read();
    Serial.print(serial_data); }
}
```

What's new:

Serial.available method, ***Serial.read*** method

Task. 2. How to use *Serial Plotter*.

Arduino comes with a cool tool called the *Serial Plotter*. It can give you visualizations of variables in real-time. Choose *Tools* → *Serial Plotter* to run this tool.

```
#define BAUDRATE 115200

unsigned long p_millis = 0;

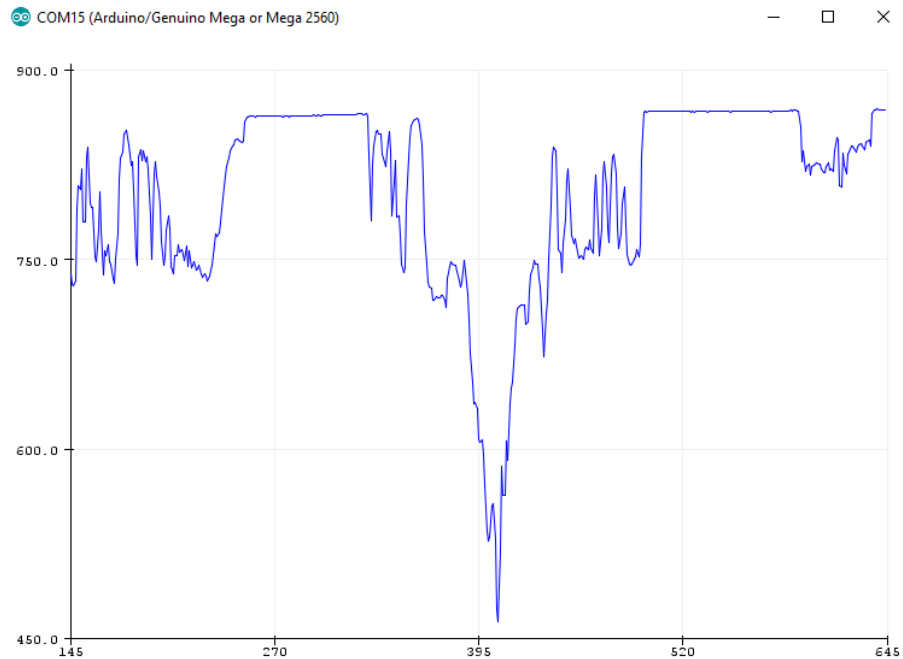
void setup() {
  Serial.begin(BAUDRATE); }

#define SERIAL_DELAY 1000

void loop() {
```

Exercise no 5: Arduino serial ports

```
if(millis() - p_millis > SERIAL_DELAY) {  
    Serial.println(int(random(1,1000)));  
    p_millis = millis(); }  
}
```



Task. 3. Implement the following code. Use *Serial Monitor* to check if it works.

```
#define BAUDRATE 115200  
  
char serial_data;  
  
void setup() {  
    Serial.begin(BAUDRATE); }  
  
void loop() {  
    if(Serial.available()>0) {  
        switch(Serial.read()) {  
            case 'A':
```

Exercise no 5: Arduino serial ports

```
        Serial.println("A was sent");
        break;

    case 'P':
        Serial.println("P was sent");
        break;

    case '1':
        Serial.println("1 was sent");
        break;

    default:
        Serial.println("Unrecognized value");
        break;
}
}
}
```

Task. 4. Implement the following code. Use *Serial Monitor* to check how it works.

```
#define BAUDRATE 115200

void setup() {
    Serial.begin(BAUDRATE); }

void loop() {
    Serial.write(0x48); // H
    Serial.write(0x65); // e
    Serial.write(0x6C); // l
    Serial.write(0x6C); // l
    Serial.write(0x6F); // o
    Serial.write(0x21); // !
}
```

Exercise no 5: Arduino serial ports

```
Serial.write(0x0A); // \n
delay(500); }
```

What's new:

Serial.write method

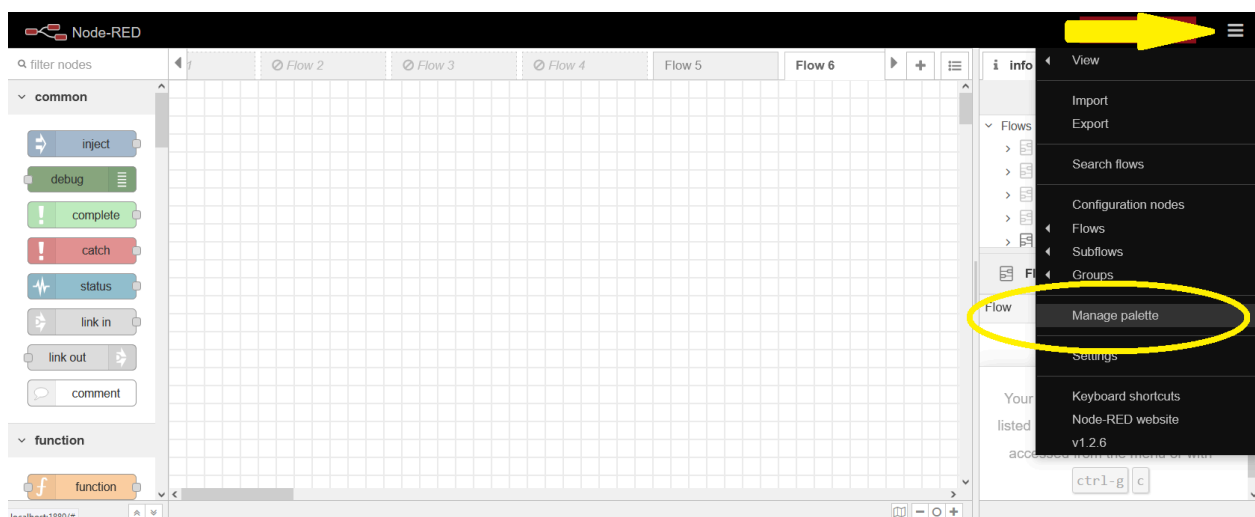
Task. 5. Node-RED installation.

Node-RED is a programming tool for wiring together hardware devices, APIs, and online services. To start working with Node-RED follow this manual:

<https://nodered.org/docs/getting-started/local>

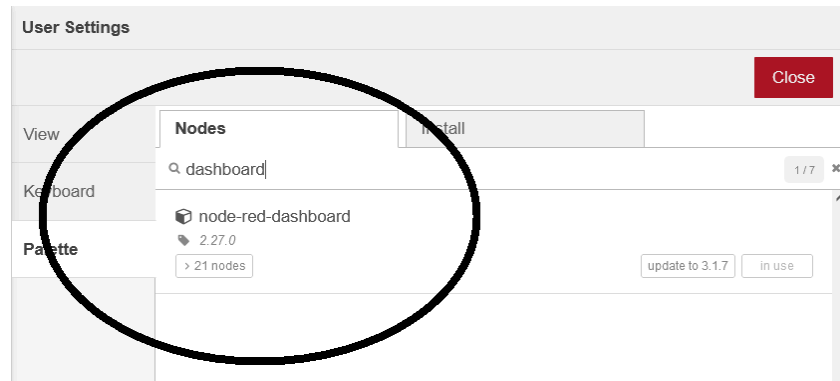
Run Node-RED after installation, open any web browser, and type *localhost:1880* in the address bar. To see UI type *localhost:1880/ui* in the address bar.

To install additional packages You have to open the Manage palette window:



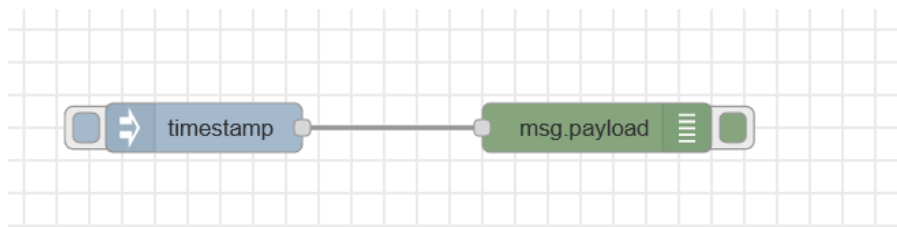
Install *dashboard* and *node-red-node-serialport* packages:

Exercise no 5: Arduino serial ports

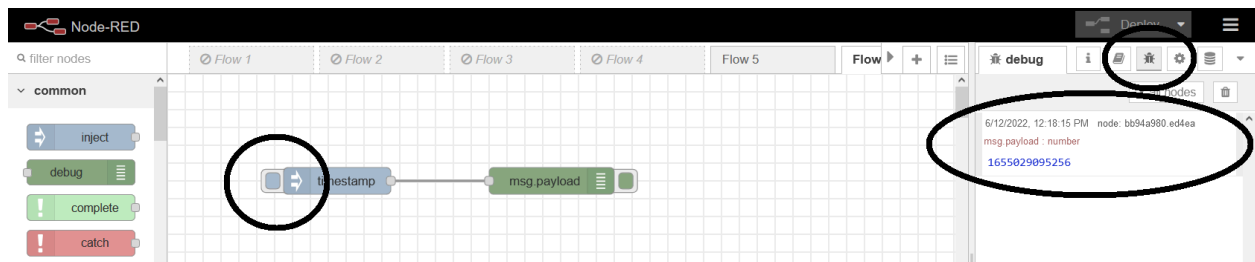


Task. 6. Hello Node-RED.

Create the following flow - all the required nodes (*Inject*, *Debug*) are available in the *Common* palette.

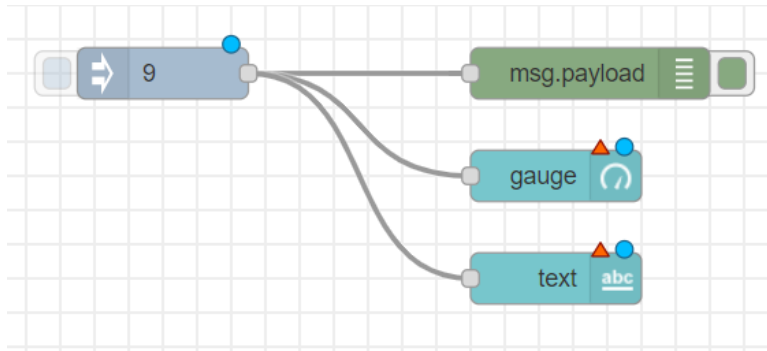


Test this flow using *Debug messages*.

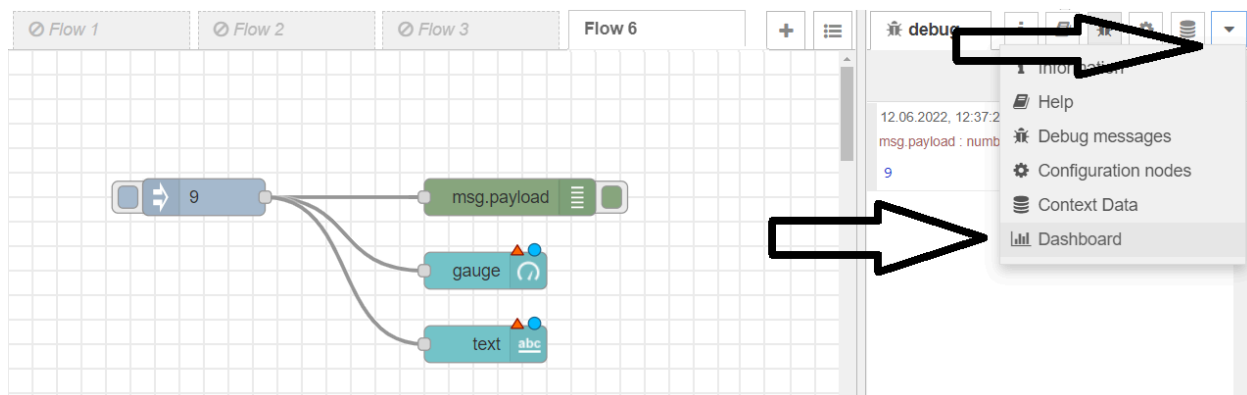


Task. 7. Introduction to a dashboard.

Double left-click on the *Inject* node. Change the *msg.payload* from *timestamp* to *number*. Set this number to the value of Your choice. Add *gauge* and *text* nodes from the *dashboard* palette.

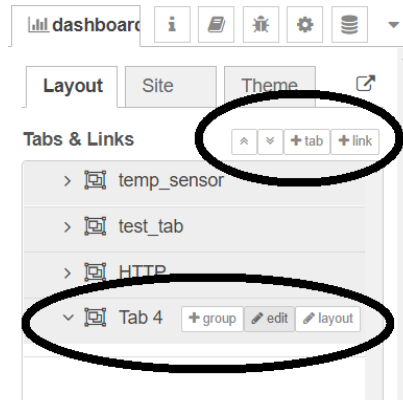


Hit the *Down* button and select *Dashboard* on the drop-down list.

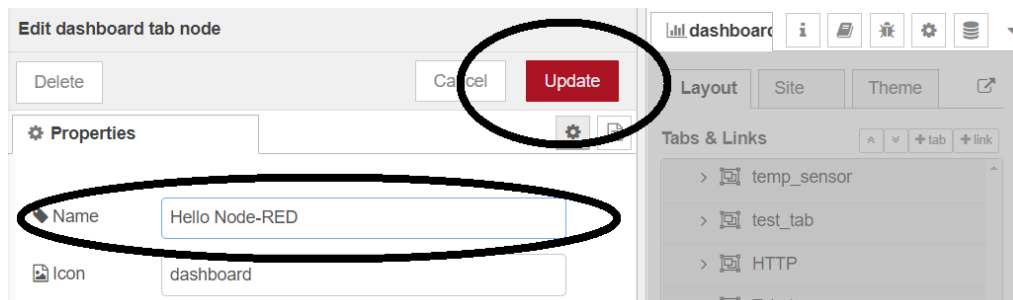


Select the *Layout* tab. Press the *+tab* button to add a new tab. Press the *edit* button to change the new tab name.

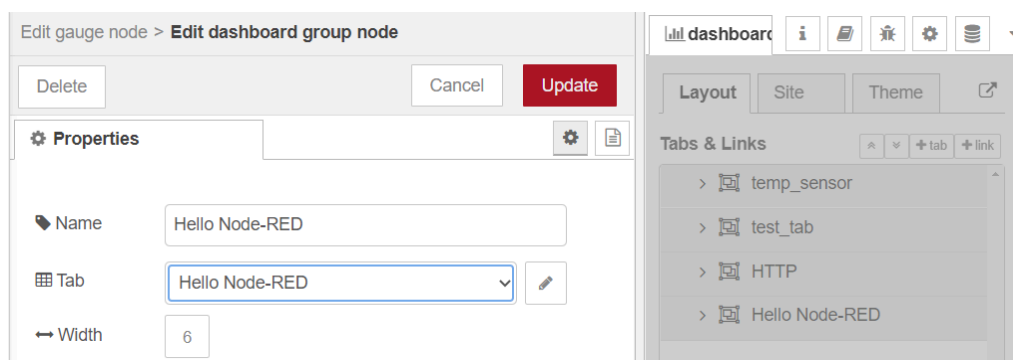
Exercise no 5: Arduino serial ports



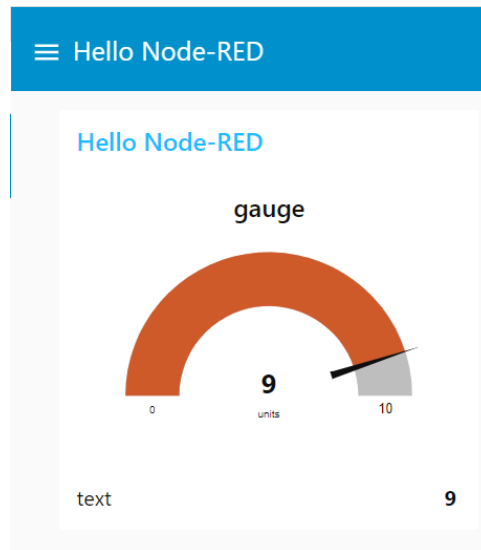
Type the name of Your choice and press the *Update* button.



Double left-click *gauge* node. Edit *Group* and select Your tab from the *Tab* dropdown list. Approve changes with the *Update* button.



Repeat the previous operation for the text node. Press *Deploy* to make the changes in Your flows. Inject the number into all nodes. Check the result in *Debug messages*. Open a new web browser window and type *localhost:1880/ui* to see the dashboard.



Task. 8. Arduino meets Node-RED.

Program the Arduino board with the following code.

```
#define BAUDRATE 115200

unsigned long p_millis = 0;

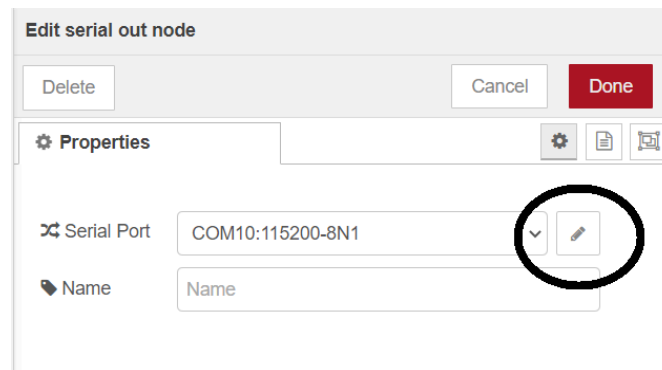
void setup() {
  Serial.begin(BAUDRATE); }

#define SERIAL_DELAY 1000

void loop() {
  if(millis() - p_millis > SERIAL_DELAY) {
    Serial.println(int(random(1,9)));
    p_millis = millis(); }
}
```

Replace *inject* node with *serial-in node* (from the *network* palette). Configure communication parameters by double left-clicking this node.

Exercise no 5: Arduino serial ports



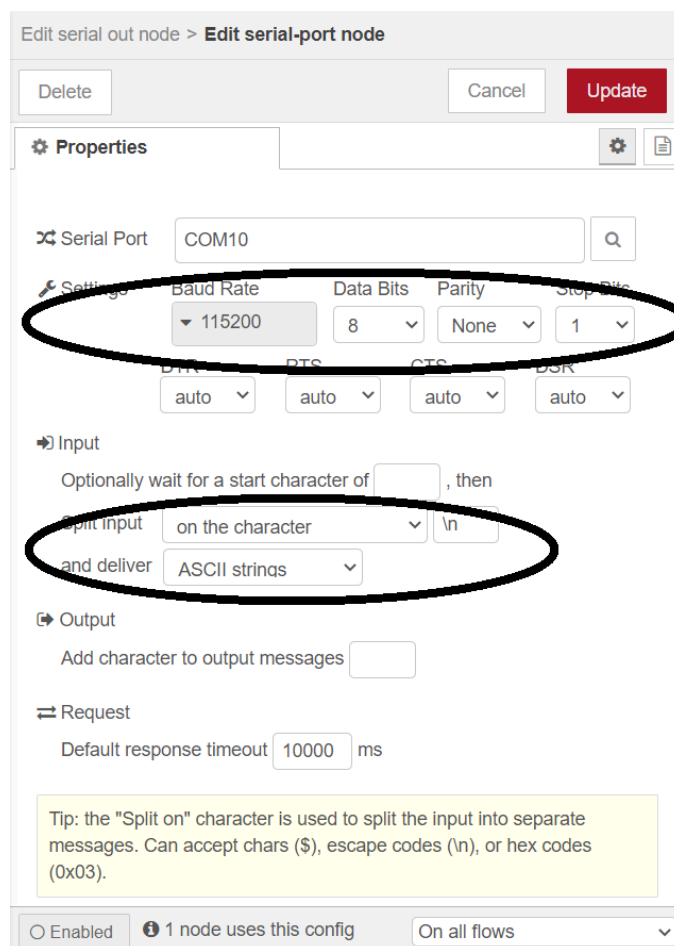
Edit serial out node

Delete Cancel Done

Properties

Serial Port COM10:115200-8N1

Name Name



Edit serial out node > Edit serial-port node

Delete Cancel Update

Properties

Serial Port COM10

Settings

Baud Rate 115200 Data Bits 8 Parity None Stop Bits 1

Input

Optionally wait for a start character of , then

Split input on the character \n

and deliver ASCII strings

Output

Add character to output messages

Request

Default response timeout 10000 ms

Tip: the "Split on" character is used to split the input into separate messages. Can accept chars (\$), escape codes (\n), or hex codes (0x03).

Enabled 1 node uses this config On all flows

Check the result in the *Debug messages*. Open a new web browser window and type *localhost:1880/ui* to see the dashboard.

Task. 9. Connect 3 LEDs to the Arduino board. Prepare a solution to control those LEDs from Your computer. The user should be able to control all the LEDs either separately or using 1 command.

Task. 10. Prepare a dashboard to control 3 LEDs connected to the Arduino board. It should also be possible to control those LEDs using buttons connected to the same Arduino module.

Task. 11. Prepare a solution to control a servomechanism connected to the Arduino board. The user should be able to set the servo position at least. LED indicators should tell the user that either maximal or minimal position was reached.

For those interested:

1. Node-RED web page:

nodered.org

2. Arduino reference on serial communication:

[www.arduino.cc/reference/en/language/functions/communication/
serial/](https://www.arduino.cc/reference/en/language/functions/communication/serial/)

3. Arduino reference on serialEvent():

[www.arduino.cc/reference/en/language/functions/communication/
serial/serialevent/](https://www.arduino.cc/reference/en/language/functions/communication/serial/serialevent/)