# A complete description of a digital microfluidic platform and its state in JSON format

## Introduction

In general, a physical digital microfluidic platform can be described at different level of abstraction depending on what is the application in which the representation is used.

For example, if the representation is used to draw a the DMF platform, information regarding the topology and the shape of the electrodes is needed. This can be represented as electrode shape and origin coordinates in a Cartesian space (XY). On the contrary, if the representation is used to compute the routes of droplets, information regarding electrodes connections (i.e., directly connected electrodes where a droplet can move) is needed. This is better represented as a connected graph where the nodes represent the electrodes and the edges represent the connection between those. In addition, state can be included in the representation. For example, the state of an electrode (ON/OFF). The same considerations also apply for the other elements of a DMF platform such as sensors, heaters, pumps, droplets, etc.

The following representation attempts to include all the possible information related to the DMF platform and its state in JSON format.

Please note that this is work in progress, so changes/improvements can be expected.

## Representation description

### Main structure

The JSON representation is divided into the following main classes:

- **information** : contains meta-information about the platform (for example, the platform name, platform type etc.)
- **electrodes** : contains the description of all the electrodes
- **actuators** : contains the description of all the actuators (heaters, pumps, magnets, etc.)
- **sensors** : contains the description of all the sensors (temperature, droplet color, droplet size, etc.)
- **inputs** : contains the description of the platform fluids inputs (reservoirs, tubes, etc.)
- **outputs** : contains the description of the platform fluids outputs (collection reservoirs, tubes, waste pads, etc.)
- **droplets** : contains the description of the active droplets
- **bubbles** : contains the description of gas bubbles (unwanted effect due to evaporation)

- **unclassified** : contains all the elements that do not belong to the previous classes (for now, empty)

Each class contains a list of JSON formatted elements. For each element a description is provided in the following section. Please note that the syntax `<string>`, `<int>`, `<float>`, `<json>` etc. are used as placeholder for the real value.

Here is a snipped to the main JSON structure with empty lists.

```
{
  "information": [<json>, <json>, ...],
  "electrodes": [<json>, <json>, ...],
  "actuators": [<json>, <json>, ...],
  "sensors": [<json>, <json>, ...],
  "inputs": [<json>, <json>, ...],
  "outputs": [<json>, <json>, ...],
  "droplets": [<json>, <json>, ...],
  "bubbles": [<json>, <json>, ...],
  "unclassified": [<json>, <json>, ...]
}
```

### Class information

The information is described as follows:

```
{
  "platform_name": <string>,
  "platform_type": <string>,
  "platform_ID": <int>,
  "sizeX": <int>,
  "sizeY": <int>
}
```

The parameters have the following meaning:

- **platform_name** : a string identifier for the platform
- **platform_type** : a string identifier
- **platform_ID** : a unique number for each DMF platform produced
- **sizeX** : an integer that identifies the size of the platform - it defines the maximum X size on which the parameter `positionX` of other elements refers to
- **sizeY** : an integer that identifies the size of the platform - it defines the maximum Y size on which the parameter `positionY` of other elements refers to

### Class electrodes

An electrode is described as follow:

```
{
  "name": <string>,
  "ID": <int>,
  "electrodeID": <int>,
  "driverID": <int>,
  "shape": <int>,
  "positionX": <int>,
  "positionY": <int>,
  "sizeX": <int>,
  "sizeY": <int>,
  "status": <int>,
  "corners": [[<int>, <int>],[<int>, <int>],[<int>, <int>], ...]
}
```

The parameters have the following meaning:

- **name**: a string identifier for the electrode
- **ID**: a unique integer ID identifying the element
- **electrodeID**: a unique electrode ID
- **driverID**: the ID of the electrode in the real platform_name (used by SETEL/CLREL)
- **shape**: the shape of the electrode - 0=rectangle, 1=custom_polygon
- **positionX** : the X coordinate of the top-left corner of the electrode for rectangular electrodes, or the origin for custom_polygon electrodes
- **positionY** : the Y coordinate of the top-left corner of the electrode for rectangular electrodes, or the origin for custom_polygon electrodes
- **sizeX** : the X size of the electrode for rectangle shape
- **sizeY** : the Y size of the electrode for rectangle shape
- **corners**: a list of corner points
- **status**: the current status of the electrode - 0=off , 1=on - (in the future a number from 0 to 100 shall be supported)

Here is an example of a rectangular electrode:

```
{
  "name": "res9el711",
  "ID": 711,
  "electrodeID": 2,
  "driverID": 25,
  "shape": 0,
  "positionX": 750,
  "positionY": 260,
  "sizeX": 20,
  "sizeY": 20,
  "status": 0,
  "corners": null
}
```

Here is an example of a custom_polygon electrode for a triangular electrode:

```
{
  "name": "res9el712",
  "ID": 712,
  "electrodeID": 10,
  "driverID": 300,
  "shape": 1,
  "positionX": 800,
  "positionY": 240,
  "sizeX": 0,
  "sizeY": 0,
  "status": 0,
  "corners": [
    [0, 0],
    [0, 20],
    [20, 20]
  ]
}
```

### Class actuators

An actuator is described as follow:

```
{
  "name": <string>,
  "ID": <int>,
  "actuatorID": -1,
  "type": <string>,
  "positionX": <int>,
  "positionY": <int>,
  "sizeX": <int>,
  "sizeY": <int>
  ...other custom parameters...
}
```

The parameters have the following meaning (for now, actuators are assumed to be rectangular):

- **name**: a string identifier for the actuator
- **ID**: a unique integer ID identifying the element
- **actuatorID**: a unique actuator ID
- **type**: the type of the actuator, for example `heater` or `magnet`
- **positionX** : the X coordinate of the top-left corner of the actuator
- **positionY** : the Y coordinate of the top-left corner of the actuator
- **sizeX** : the X size of the actuator
- **sizeY** : the Y size of the actuator

Other custom parameters can be present do describe state of a specific actuator. For example here is the JSON for a heater: Other custom paramenters can be present do describe state of a specific actuator. For example here is the JSON for a heater:

```
{
  "name": "heat2",
  "ID": 2,
  "actuatorID": 30,
  "type": "heater",
  "positionX": 270,
  "positionY": 0,
  "sizeX": 80,
  "sizeY": 400,
  "actualTemperature": 0,
  "desiredTemperature": 0,
  "status": false,
  "nextDesiredTemperature": 0,
  "nextStatus": false,
  "corners": null
}
```

For now, actuators are assumed to be rectangular.

### Class sensors

A sensor is described as follows:

```
{
  "name": <string>,
  "ID": <int>,
  "sensorID": -1,
  "type": <string>,
  "positionX": <int>,
  "positionY": <int>,
  "sizeX": <int>,
  "sizeY": <int>
  ...other custom parameters...
}
```

The parameters have the following meaning (for now, sensors are assumed to be rectangular):

- **name**: a string identifier for the sensor
- **ID**: a unique integer ID identifying the element
- **sensorID**: a unique sensor ID
- **type**: the type of the sensor, for example `temperature` or `RGB_color`
- **positionX** : the X coordinate of the top-left corner of the sensor
- **positionY** : the Y coordinate of the top-left corner of the sensor
- **sizeX** : the X size of the sensor
- **sizeY** : the Y size of the sensor

Other custom parameters can be present do describe state of a specific actuator. For example here is the JSON for a RGB_color sensor:

```
{
  "name": "rgb1",
  "ID": 3,
  "actuatorID": 35,
  "type": "RGB_color",
  "positionX": 270,
  "positionY": 0,
  "sizeX": 80,
  "sizeY": 400,
  "valueRed": 0,
  "valueGreen": 0,
  "valueBlue": 0
}
```

### Class inputs

An input specifies a point where a droplet appears after an input operation. An input is described as follows:

```
{
  "name": <string>,
  "ID": <int>,
  "inputID": -1,
  "positionX": <int>,
  "positionY": <int>,
  ...other custom parameters...
}
```

The parameters have the following meaning (for now, sensors are assumed to be rectangular):

- **name**: a string identifier for the input
- **ID**: a unique integer ID identifying the element
- **inputID**: a unique input ID
- **positionX** : the X coordinate of the input point
- **positionY** : the Y coordinate of the input point

### Class outputs

An output specifies a point where a droplet should be delivered after an output operation. An output is described as follows:

```
{
  "name": <string>,
  "ID": <int>,
  "outputID": -1,
  "positionX": <int>,
  "positionY": <int>,
  ...other custom parameters...
}
```

The parameters have the following meaning (for now, sensors are assumed to be rectangular):

- **name**: a string identifier for the output
- **ID**: a unique integer ID identifying the element
- **outputID**: a unique output ID
- **positionX** : the X coordinate of the output point
- **positionY** : the Y coordinate of the output point

### Class droplets

Droplets are assumed to have elliptical/circular shape. A droplet is described as follows:

```
{
  "name": <string>,
  "ID": <int>,
  "substance_name": <string>
  "positionX": <int>,
  "positionY": <int>,
  "sizeX": <int>,
  "sizeY": <int>,
  "color": <string>,
  "temperature": <float>
  ...other custom parameters...
}
```

The parameters have the following meaning:

- **name** : a string identifier for the droplet
- **ID** : a unique integer ID identifying the element
- **positionX** : the X coordinate of the top-left corner of the droplet
- **positionY** : the Y coordinate of the top-left corner of the droplet
- **sizeX** : the X size of the droplet
- **sizeY** : the Y size of the droplet
- **color**: the color of the droplet as HEX RGB
- **temperature**: the current temperature of the droplet

### Class bubbles

A gas bubble is an unwanted effect happening in closed-system DMF platforms due to the evaporation of the liquids. A bubble is trapped in the system and sometimes makes the movement of droplets more difficult. Bubble are assumed to have an elliptical/circular shape. A bubble is described as follows:

```
{
  "name": <string>,
  "ID": <int>,
  "positionX": <int>,
  "positionY": <int>,
  "sizeX": <int>,
  "sizeY": <int>
}
```

The parameters have the following meaning:

- **name** : a string identifier for the bubble
- **ID** : a unique integer ID identifying the element
- **positionX** : the X coordinate of the top-left corner of the bubble
- **positionY** : the Y coordinate of the top-left corner of the bubble
- **sizeX** : the X size of the bubble
- **sizeY** : the Y size of the bubble

### Class unclassified

For now this class does not contain any element. It is here to account for possible future elements that do not belong to any other class.

## An example of a 4 by 3 electrodes board

Here is a diagram of the board with the 12 electrodes numbered from 0 to 11. Electrode 4 coincides with input 0, electrode 7 coincides with output 0. The platform size is defined to be 80 by 60. The origin of the platform is in the top left corner

```
    x -->
    0                               79
y 0 ---------------------------------
|   |       |       |       |       |
|   |   0   |   1   |   2   |   3   |
    |       |       |       |       |
    ---------------------------------
    |       |       |       |       |
    |   4   |   5   |   6   |   7   |
    |  in0  |       |       | out0  |
    ---------------------------------
    |       |       |       |       |
    |   8   |   9   |  10   |  11   |
    |       |       |       |       |
 59 ---------------------------------
```

Here is the related JSON description

```
{
  "information": [
    {
      "platform_name": "4by3example",
      "platform_type": "example",
      "platform_ID": 012345678,
      "sizeX": 80,
      "sizeY": 60
    }
  ],
  "electrodes": [
    {
      "name": el0,
      "ID": 0,
      "electrodeID": 0,
      "driverID": -1,
      "shape": 0,
      "positionX": 0,
      "positionY": 0,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el1,
      "ID": 1,
      "electrodeID": 1,
      "driverID": -1,
      "shape": 0,
      "positionX": 20,
      "positionY": 0,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    }

    {
      "name": el2,
      "ID": 2,
      "electrodeID": 2,
      "driverID": -1,
      "shape": 0,
      "positionX": 40,
      "positionY": 0,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },
    {
      "name": el3,
      "ID": 3,
      "electrodeID": 3,
      "driverID": -1,
      "shape": 0,
      "positionX": 60,
      "positionY": 0,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el4,
      "ID": 4,
      "electrodeID": 4,
      "driverID": -1,
      "shape": 0,
      "positionX": 0,
      "positionY": 20,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el5,
      "ID": 5,
      "electrodeID": 5,
      "driverID": -1,
      "shape": 0,
      "positionX": 20,
      "positionY": 20,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el6,
      "ID": 6,
      "electrodeID": 6,
      "driverID": -1,
      "shape": 0,
      "positionX": 40,
      "positionY": 20,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el7,
      "ID": 7,
      "electrodeID": 7,
      "driverID": -1,
      "shape": 0,
      "positionX": 60,
      "positionY": 20,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el8,
      "ID": 8,
      "electrodeID": 8,
      "driverID": -1,
      "shape": 0,
      "positionX": 0,
      "positionY": 40,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el9,
      "ID": 9,
      "electrodeID": 9,
      "driverID": -1,
      "shape": 0,
      "positionX": 20,
      "positionY": 40,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el10,
      "ID": 10,
      "electrodeID": 10,
      "driverID": -1,
      "shape": 0,
      "positionX": 40,
      "positionY": 40,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    },

    {
      "name": el11,
      "ID": 11,
      "electrodeID": 11,
      "driverID": -1,
      "shape": 0,
      "positionX": 60,
      "positionY": 40,
      "sizeX": 20,
      "sizeY": 20,
      "status": 0,
      "corners": null
    }

  ],
  "actuators": null,
  "sensors": null,
  "inputs": [
    {
      "name": in0,
      "ID": ,
      "inputID": 0,
      "positionX": 10,
      "positionY": 30,
    }
  ],
  "outputs": [
    {
      "name": out0,
      "ID": ,
      "outputID": 0,
      "positionX": 70,
      "positionY": 30,
    }
  ],
  "droplets": null,
  "bubbles": null,
  "unclassified": null
}
```