SENSACELL FOR PROCESSING

DOCUMENTATION

Class Sensacell

The Sensacell class contains a virtual array representing the Sensacell array which is connected to. Each cell of this array has one sensor value and one color value. The sensors values can only be changed by the Sensacell array. The colors values can be changed by the user and then displayed on the Sensacell array.

Constructor and Description

Constructor and Description
Sensacell(Serial sensaPort, PApplet parent) Initializes a newly created Sensacell object so that it represents the Sensacell array connected to the serial sensaPort.

Method Summary

Type	Method and Description
void	moduleDisplay(int moduleAddress) Display on sensacell the values contains in the module number moduleAddress.
void	fullDisplay() Display on sensacell all the values of the virtual array.
void	moduleListening(int moduleAddress) Set on the virtual array the sensors values of the module number moduleAddress.
void	fullListening() Set on the virtual array all the sensors values of the sensacell array.
void	Update() Intelligent listening and displaying
void	setSerial(Serial sensaPort) Set the serial which is connected to the sensacell array.
void	setProportionnalMode() Set the proportional read mode on sensacell.
void	setBinaryMode() Set the binary read mode on sensacell.
int	getAddress(int x, int y) Return the address of the module which contains the cell[x][y].
int	getSensorValue(int x, int y) Return the value of the sensor of the cell[x][y].
int	getColor(int x, int y) Return the color (hexadecimal value) of the cell[x][y].
void	setColor(int x, int y, int colorValue) Set the color (hexadecimal value) of the cell[x][y].
int	getHeight() Return the height of the virtual array.
int	getWidth()

	Return the width of the virtual array.
void	autoAddressing(String fileName) The virtual array is initialized with the sensacell initialization protocol and then the configuration is saved on a file named filename.
void	fileAddressing(String fileName) The virtual array is initialized with the file named filename.

Examples

Setup :

```
import processing.serial.*;

//New object sensacell
 Sensacell sensacell;

   public void setup() {
     //Looking for the availables serials ports
     int sIndex = 0;
      for(int i=0;i<Serial.list().length;i++){
         if(Serial.list()[i]!=null)
         {
           println(i+" "+Serial.list()[i]);
           sIndex=i;
         }
       }

       //Initializes a newly created Sensacell object connected to the only available serial
       sensacell = new Sensacell(new Serial(this,Serial.list()[0],230400));

       //The virtual array is initialized with the sensacell initialization protocol
       sensacell.autoAddressing("Config.txt");

       //Set the size og the processing windows
       size(sensacell.getWidth()*20,sensacell.getHeight()*20);

       //Display on sensacell the virtual array
       sensacell.fullDisplay();
   }
```

Class Blob

The Blob class contains allow blob detection and other methods with Sensacell.

Constructor and Description

Constructor and Description
Blob(Sensacell tab) Initializes a newly created Blob object.

Method Summary

Type	Method and Description
coord[]	getCentroids(Sensacell tab) Get the coordinates of every detected blob on Sensacell.

Examples

Coming

Class Utils

The Utils class contains different methods which can be used with Sensacell.

Constructor and Description

Constructor and Description
Utils(Sensacell Array) Initializes a newly created Sensacell object so that it represents the Sensacell array connected to the serial sensaPort.

Method Summary

Type	Method and Description
void	DrawFilledCircle(int x0, int y0, int radius, int Color) Set a filled circle on the virtual array of Sensacell.
void	DrawCircle(int x0, int y0, int radius, int Color) Set a circle on the virtual array of Sensacell.

Examples

Coming