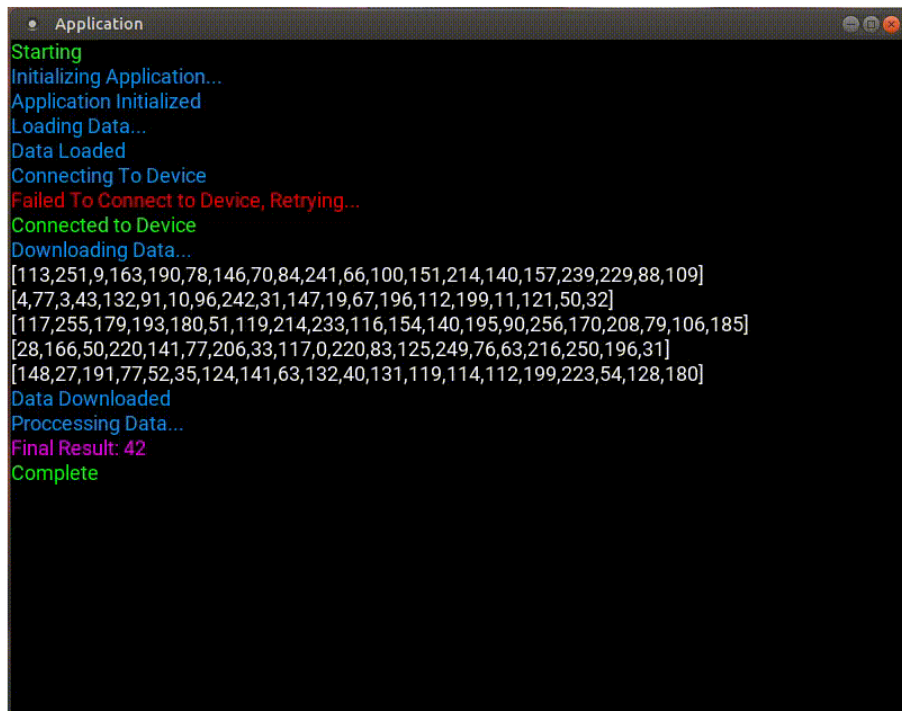


Kivy Console Widget



```
Application
Starting
Initializing Application...
Application Initialized
Loading Data...
Data Loaded
Connecting To Device
Failed To Connect to Device, Retrying...
Connected to Device
Downloading Data...
[113,251,9,163,190,78,146,70,84,241,66,100,151,214,140,157,239,229,88,109]
[4,77,3,43,132,91,10,96,242,31,147,19,67,196,112,199,11,121,50,32]
[117,255,179,193,180,51,119,214,233,116,154,140,195,90,256,170,208,79,106,185]
[28,166,50,220,141,77,206,33,117,0,220,83,125,249,76,63,216,250,196,31]
[148,27,191,77,52,35,124,141,63,132,40,131,119,114,112,199,223,54,128,180]
Data Downloaded
Processing Data...
Final Result: 42
Complete
```

This widget was designed to provide a terminal-like output running inside a Kivy GUI app. Aside from displaying data to the screen, it also has the ability to write data to a plain text log file. This tool has proven useful in several different industrial automation and manufacturing applications. While not shown in the included animation, the Console descends from scroll view and will show a scroll bar once the content becomes larger than the current screen.

The Console class is initialized like this:

This widget was designed to provide a terminal-like output running inside a Kivy GUI app. Aside from displaying data to the screen, it also has the ability to write data to a plain text log file. This tool has proven useful in several different industrial automation and manufacturing applications. While not shown in the included animation, the Console descends from scroll view and will show a scroll bar once the content becomes larger than the current screen.

The Console class is initialized like this:

console = Console(logDir,logFile)

where logDir is the path to the log file directory and logFile is the name of the log file.

The Console class exposes the following methods (look inside Demo.py for an example implementation).

message(text, color)

displays the supplied text to the console using the specified color (or white if not specified) and also writes that same message to the log file

message_temp(text, color)

displays the supplied text to the console using the specified color (or white if not specified) and also writes that same message to the log file. This line will be deleted and replaced the next time a line is written to the console.

note(text, color)

displays the supplied text to the console using the specified color (or white if not specified) but writes nothing to the log file.

note_temp(text, color)

displays the supplied text to the console using the specified color (or white if not specified) but writes nothing to the log file. This line will be deleted and replaced the next time a line is written to the console.

log(text)

writes the supplied text to the log file, but does not write it to the console.

colors:

These are the currently supported colors:

```
pallet = {  
    'yellow':'FFFF00',  
    'white':'FFFFFF',  
    'blue':'0099FF',  
    'green':'00FF00',  
    'red':'FF0000',  
    'purple':'FF00FF',  
    'orange':'FFBB00',  
    'gray':'999999'  
}
```

They can be extended, if desired, by editing Console.py

known issues:

in the file Console.py I have assigned an absolute path to the layout file (Console.kv). You must replace this with the absolute path on your own system or this will not work.