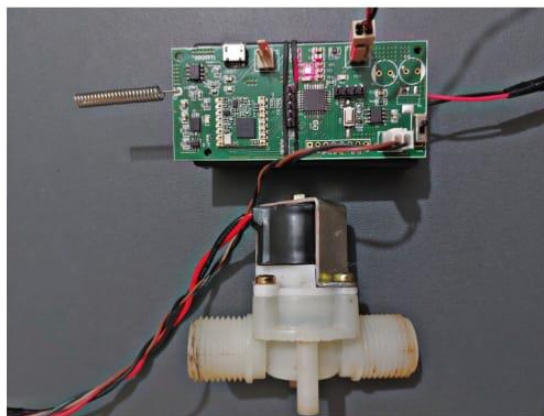
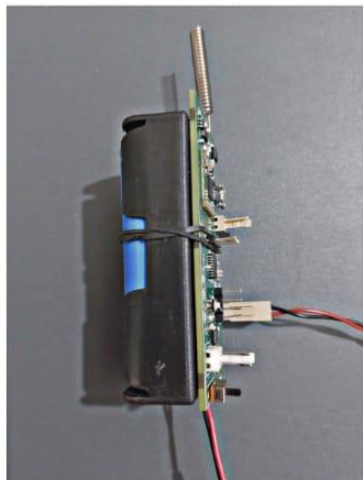
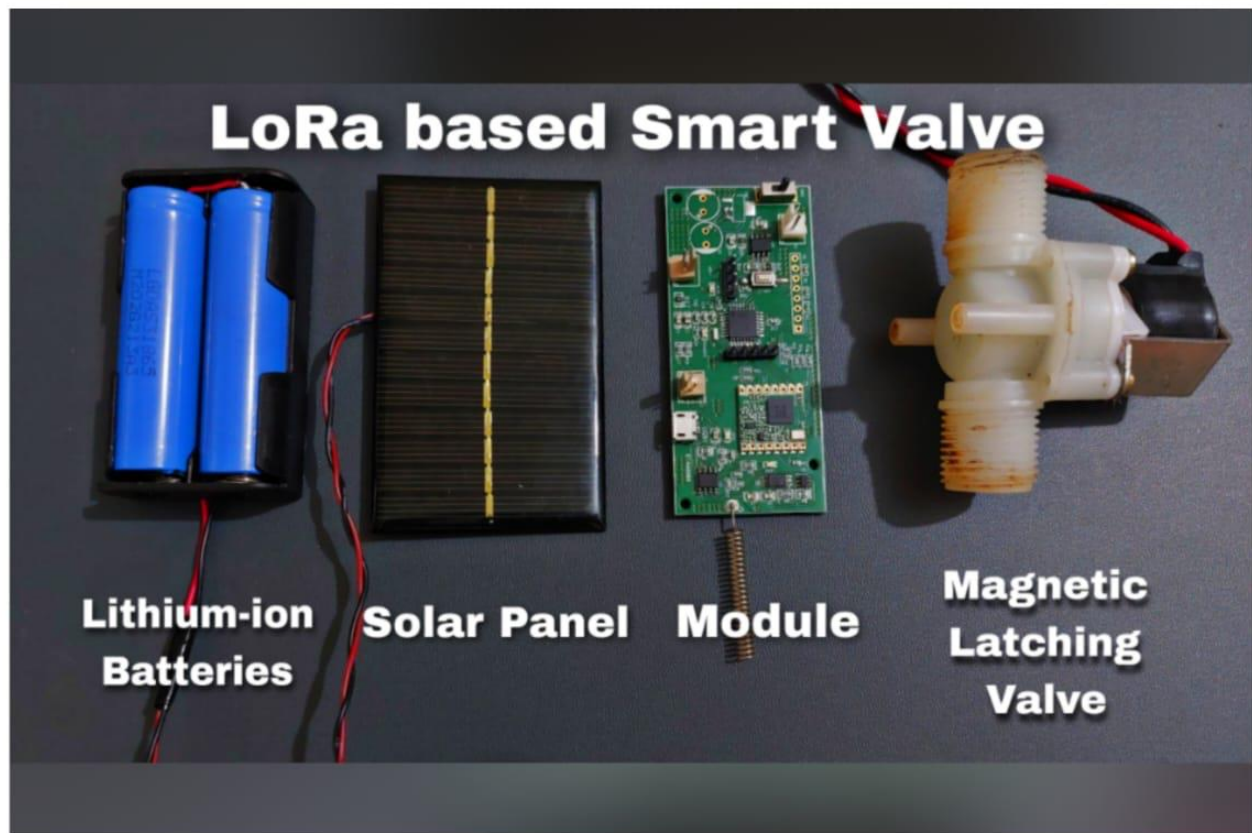
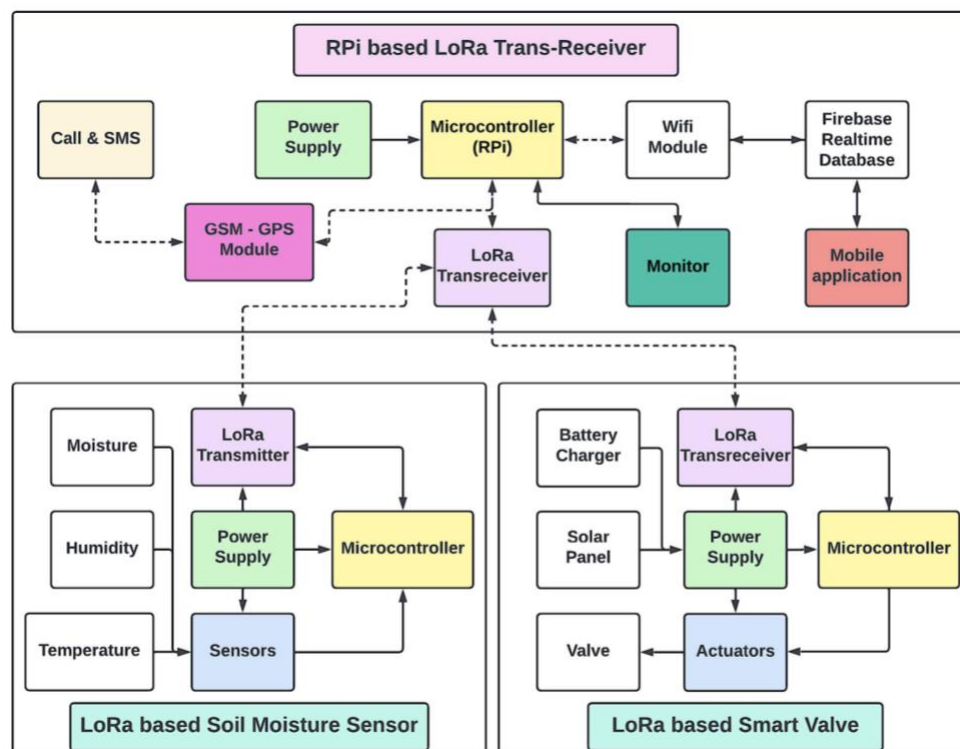


# ★ LoRa base Smart Valve ★

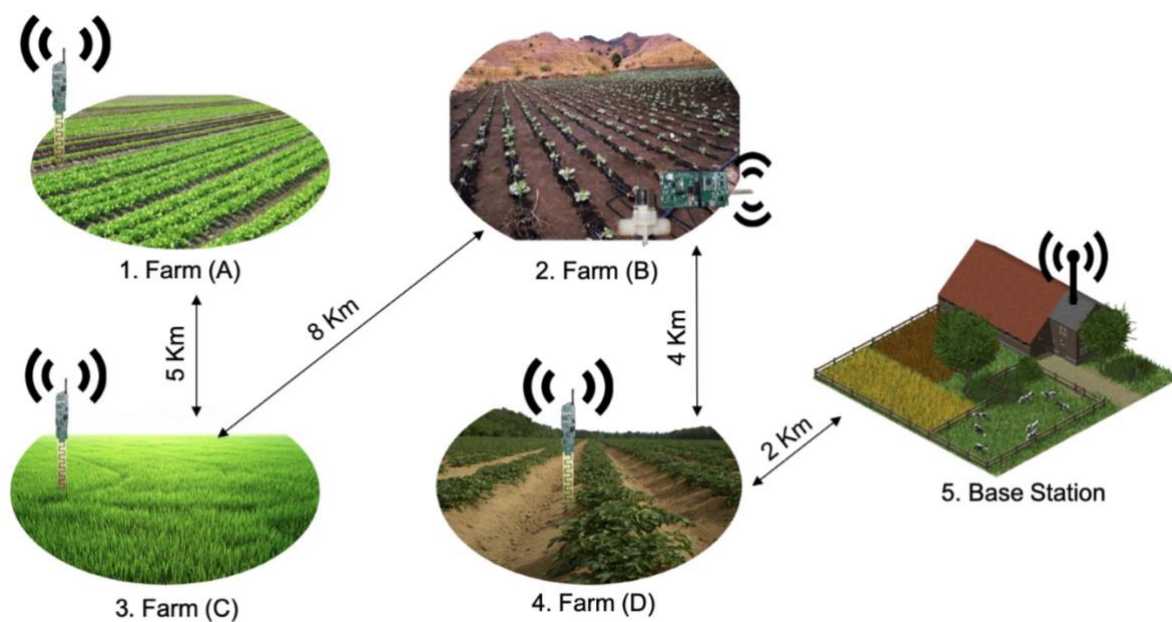
Project by (Product): Deepak Tukaram Hadkar



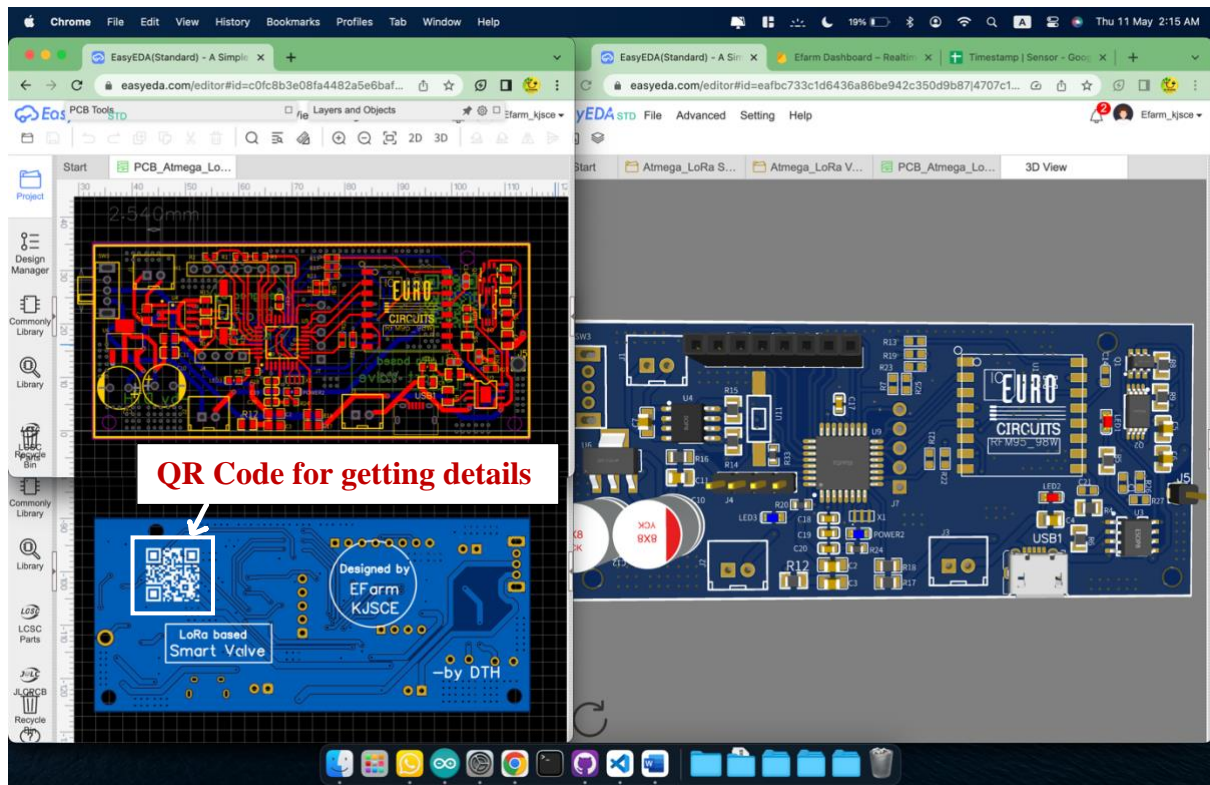
## 1. Block Diagram of the entire system :



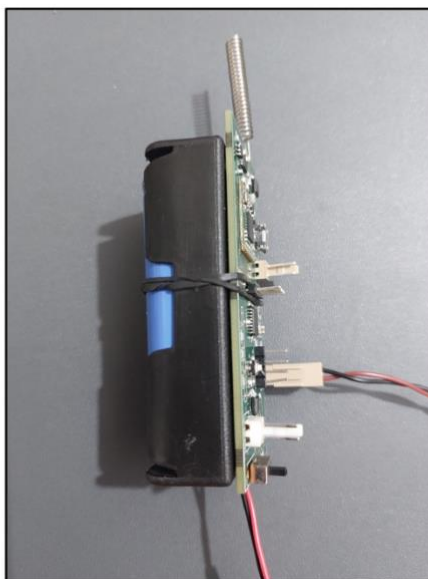
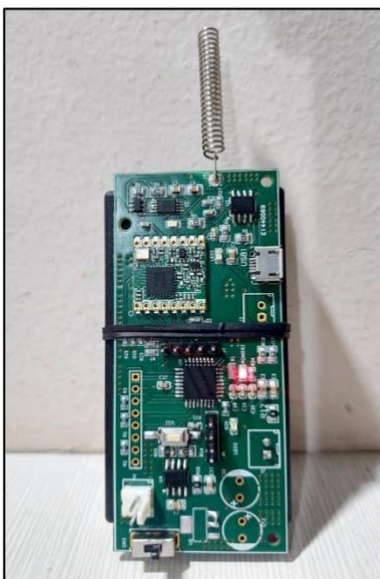
## 2. Conceptual Model :



### 3. Designing PCB for the Smart Valve:





### 4. Ready PCB with all the components onboard:





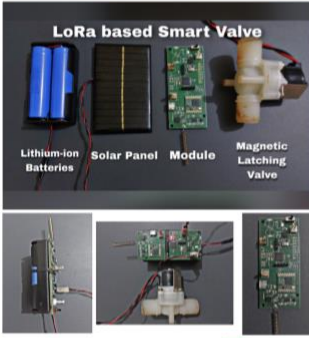
## 5. Prize received:

### Won 3<sup>rd</sup> Prize in the competition



EUROCIRCUITS PCB DESIGN CONTEST 2022


#### LoRa based Smart Valve



#### Smart Valve - Electric Tap


Made by [efarm-kjsce](#) / Communication / Environmental Sensing / Garden / Notifications / Sensors

About the project  
LoRa-based Smart Valve(Electric tap), which is connected through the water source to the drip irrigation system. It is operated according to the moisture readings of the soil received by him. It is a Lithium-ion battery operated with a micro-USB charging port in addition to Solar Panel to charge the battery.

 Eurocircuits PCB contest winner

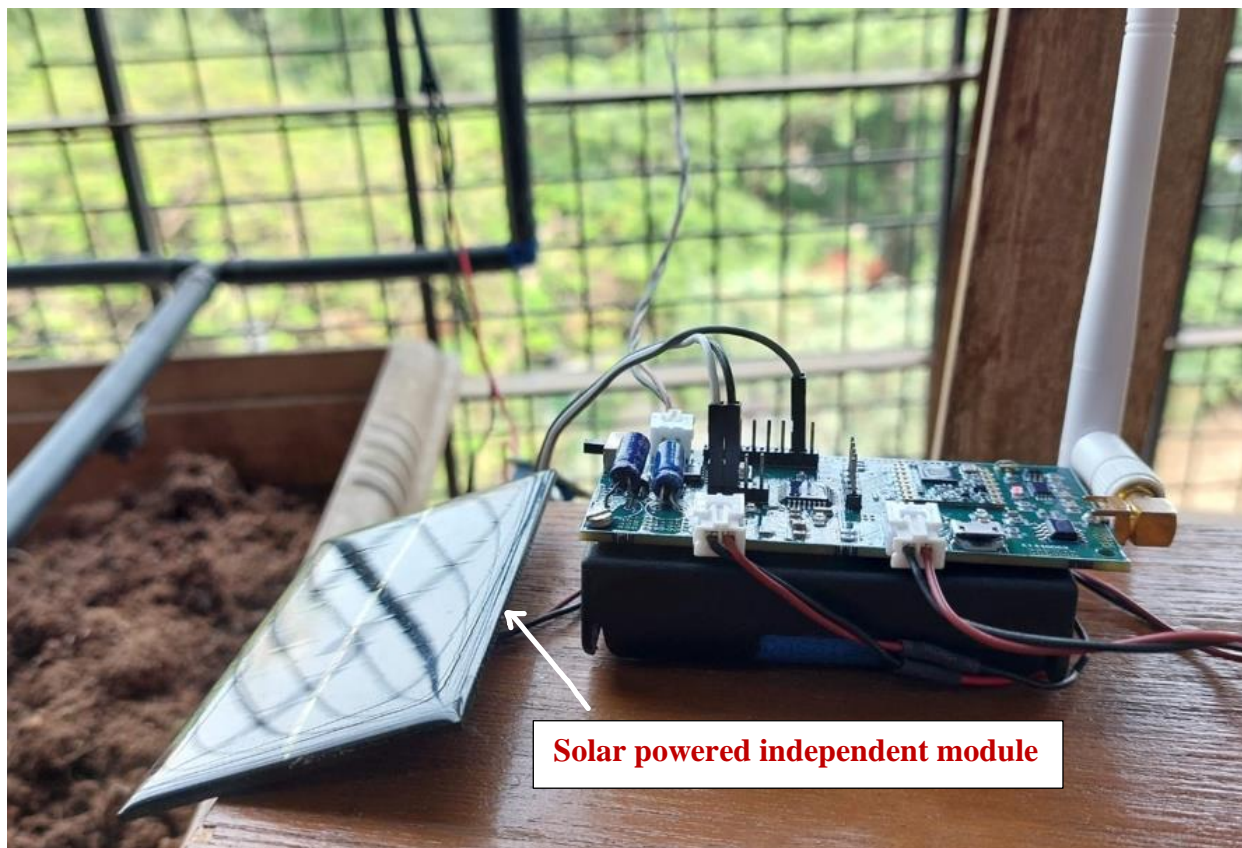
Project info  
Difficulty: Medium  
Platforms: Arduino  
Estimated time: 1 week  
License: [Copyright © All Rights Reserved - The intellectual property and copyright belong exclusively to the author of the work](#)

Third Place: **Smart Valve - Electric Tap**  
In third place, winning €400 is [efarm-kjsce](#) with their custom **Smart Valve** project. The PCB incorporates a low-power MCU for taking soil moisture readings, triggering a valve to irrigate dry land. Where things get really smart is the incorporated LoRa module for communicating soil moisture levels to the cloud and allowing for remote triggering by the user.



28th June 2022: Idea submission opens  
26th July 2022: Ideas submission deadline  
3rd August 2022: PCB's manufactured & shipped  
4th October 2022: Projects submissions deadline  
18th October 2022: Winners announced!

## 6. Implemented in Drip based Vertical Farming model (Prototype) :



## 7. Implemented in Drip based Vertical Farming model (Prototype) :



## 8. Output ( Serial Monitor ) :

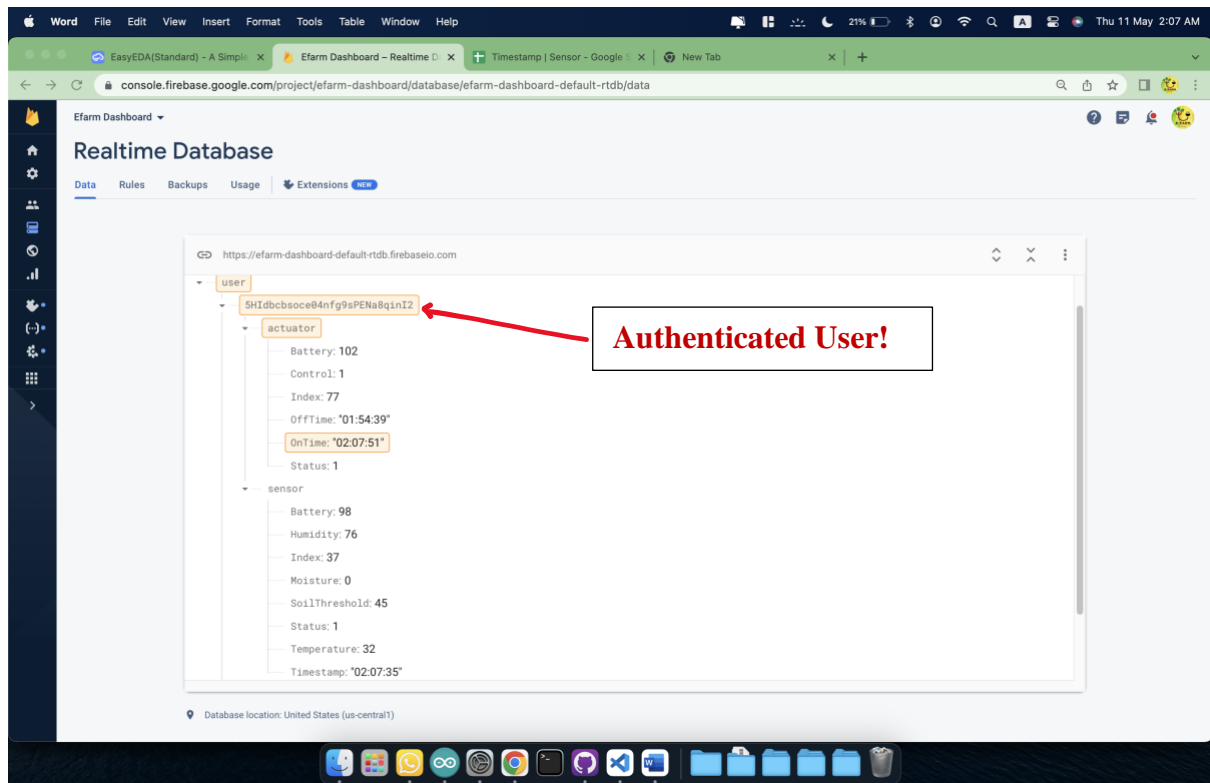
```
Arduino
/dev/cu.usbserial-10

02:24:34.122 -> YTA IP Address: 192.168.66.167
02:24:34.122 -> Ttt token), status = on request
02:24:35.334 -> Token info: type = id token (GITKit token), status = ready
02:24:36.353 -> User UID: SHIdcbsoce04nfg9sPENaQinI2Connecting to script.google.com
02:24:37.675 ->
02:24:37.675 -> Write into cell 'A1'
02:24:37.675 -> ----->
02:24:41.694 -> Successfully wrote: TimeStamp
02:24:41.694 -> into spreadsheet.
02:24:42.184 ->
02:24:42.184 -> GET: Fetch Google Calendar Data:
02:24:42.184 -> ----->
02:24:43.042 -> <!DOCTYPE html>-html>-head<-link rel="shortcut icon" href="//ssl.gstatic.com/docs/script/images/favicon.ico"><title>Error</title><style type="text/css" nonce="iGq132w#BducWzoAPyQaW">body {backgrou
02:24:44.045 ->
02:24:44.045 -> Start Sending Sensor Data to Google Spreadsheet
02:24:44.045 -> Do Something
02:24:44.045 -> Received message: 0,89,0,76.89,32.41,1,96,129,1,102,0
02:24:58.126 -> Updated Firebase!!!
02:25:05.353 -> Sent : Temp and Humid
02:25:05.353 -> POST or SEND Sensor data to Google Spreadsheet:
02:25:09.223 -> Success
02:25:14.249 -> Received message: 0,90,0,76.91,32.41,1,96,130,1,102,0
02:25:26.122 -> Updated Firebase!!!
02:25:28.141 -> POST or SEND Sensor data to Google Spreadsheet:
02:25:31.632 -> Success
02:25:36.650 -> Received message: 0,91,0,76.91,32.41,1,96,131,1,102,0
02:25:48.564 -> Updated Firebase!!!
02:25:50.644 -> POST or SEND Sensor data to Google Spreadsheet:
02:25:55.594 -> Success
02:26:00.633 -> Received message: 0,92,0,76.87,32.41,1,98,132,1,102,0
02:26:12.233 -> Updated Firebase!!!
02:26:14.215 -> POST or SEND Sensor data to Google Spreadsheet:
02:26:18.028 -> Success
02:26:23.037 -> Received message: 0,93,0,76.87,32.42,1,98,133,1,102,0
02:26:34.915 -> Updated Firebase!!!
02:26:36.954 -> POST or SEND Sensor data to Google Spreadsheet:
02:26:40.759 -> Success
02:26:45.778 -> Received message: 0,94,0,76.84,32.42,1,98,134,1,102,0
02:26:57.977 -> Updated Firebase!!!
02:26:59.973 -> POST or SEND Sensor data to Google Spreadsheet:
02:27:03.393 -> Success
02:27:08.449 -> Received message: 0,95,0,76.88,32.42,1,97,135,1,102,0
02:27:20.054 -> Updated Firebase!!!
02:27:22.074 -> POST or SEND Sensor data to Google Spreadsheet:
02:27:25.500 -> Success
02:27:30.524 -> Received message: 0,96,0,76.87,32.42,1,96,136,1,102,0
02:27:41.796 -> Updated Firebase!!!
02:27:43.831 -> POST or SEND Sensor data to Google Spreadsheet:

Autoscroll Show timestamp Newline 9600 baud Clear output
```



## 9. Output ( Realtime Firebase Database ) :



## 10. Output ( Google Sheets ) – Realtime Data Logging:

The screenshot shows a Google Sheet titled "Timestamp | Sensor" with the following data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	TimeStamp	Humidity	Temperature	Moisture										
1	01:17:32	76%	32°C	0%										
2	01:17:32	76%	32°C	0%										
3	01:17:32	76%	32°C	0%										
4	01:18:01	76%	32°C	0%										
5	01:18:25	76%	32°C	0%										
6	01:18:48	76%	32°C	0%										
7	01:19:11	76%	32°C	0%										
8	01:19:34	76%	32°C	0%										
9	01:21:01	76%	32°C	0%										
10	01:21:01	76%	32°C	0%										
11	01:21:29	76%	32°C	0%										
12	01:21:53	76%	32°C	0%										
13	01:22:15	76%	32°C	0%										
14	01:22:38	76%	32°C	0%										
15	01:23:00	76%	32°C	0%										
16	01:23:23	76%	32°C	0%										
17	01:23:45	76%	32°C	0%										
18	01:24:10	76%	32°C	0%										
19	01:24:33	76%	32°C	0%										
20	01:24:55	76%	32°C	0%										
21	01:25:20	76%	32°C	0%										
22	01:25:43	76%	32°C	0%										
23	01:26:05	76%	32°C	0%										
24	01:26:27	76%	32°C	0%										
25	01:26:50	76%	32°C	0%										
26	01:27:14	76%	32°C	0%										
27	01:27:37	76%	32°C	0%										
28	01:28:01	76%	32°C	0%										

## 11. Output ( Web Application ) :

