Data Logging on Google Drive Spreadsheet

Swapnil R. Katre, Prof. Mrs. Jyoti Giri

Abstract— the growing demand of intelligent embedded devices in day-to-life has also increase the demand of monitoring and managing data provided by it. So, a solution to connect and store data on server space, which is link to the user Gmail services, could prove to be a great user friendly step. This solution will address the challenge of logging, securing and accessing data from remote device for Analysis.

In this paper, an embedded sensor device will connect to the Google Drive. The sensor data will be uploaded to Google Spreadsheet using user credentials. The uploaded data can then be able to view, edit across the globe. Using different Google featured tools linked in Google Drive; data can be easily analyzed and represented. As Google interface are user friendly and used by many people it can be easier to adapt. Further the data can be linked with Cloud computing services, which uses technology like Big Data, Hadoop, etc for analyzing.

Index Terms — Data Logger, Google Drive, IoT, RTOS MQX, MK60FN1M0VLQ12

———————————— ◆ ————————————

1 INTRODUCTION

In the recent decades we have seen many types of Data logging systems. The recent technological developments in cloud computing and cloud storage have given new possibility in logging and viewing data for analysis and diagnostics.

As the cost charged by service provider of such services is very high, normal individual and small research associate find it difficult to avail such professional help. So, if our embedded system can connect to internet and could link to online storage space services like Google Drive, Dropbox, etc to log data which could further be easier to represented and analyzed globally. As these services are secure, the confidentiality of the data will be maintained.

2 OBJECTIVE

To Connect Embedded Device as Client to Google Server Services "Google Drive". Use Google Drive product "Spreadsheet" to display logged data.

3 SYSTEM ARCHITECTURE

In **Error! Reference source not found.**, complete block dia-

————————————————

- *Swapnil R Katre is currently pursuing masters degree program in Embedded systems and VLSI Design under electronics and telecommunication Department in Pune University, India, PH-917276658909.*
  E-mail: swapnilkatre@gmail.com

Figure 1 System Block Diagram
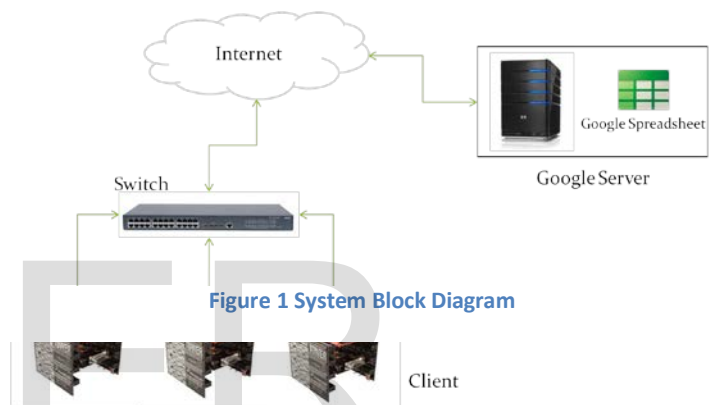
gram of the system is shown. The clients are the customized embedded system, which would connect to the local network. This device will create a link with the Google web service, this web service will then log the data in the spreadsheet file created under user credentials.

The Embedded device will have a RTOS running on it, to support the TCP/IP stack which will configure device as client to form a data link and as server to collect the data in the same network.

The Google webs service will be written in XML, JSON as per instructed by Google developer guide.

4 EMBEDDED DEVICE - DATA LOGGER

To connect the device as client, TCP/IP stack should run in firmware, which is dependent on RTOS. The RTOS used here is an open-source Freescale MQX RTOS. The benefit on this RTOS is the RTCS stack which can support more than 5 sockets as per availability of memory in controller. We can configure the same device as server simultaneously to make a gateway for collecting the data in network and by single link connect the device to Google web service.
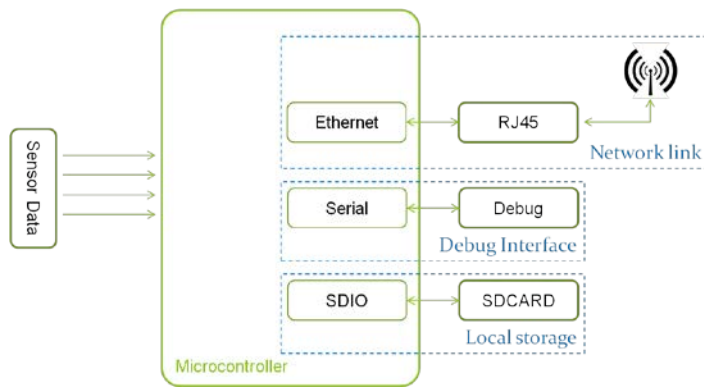
4.1 Block Diagram

Figure 2 Block Diagram

The Block diagram in the *Figure 2 Block Diagram* gives us the detailed about the hardware need by the embedded device. The Data logger should log data locally in case of no connectivity, we would require a micro SDCARD slot and related hardware. For TCP/IP connectivity, Ethernet interface would be required. In case, this device is installed in remote area where internet is not available, SDCARD will come in rescue. To run these entire interfaces a much faster controller is required. Freescale MK60 controller which has Cortex M4 core with 1MB Flash and 128 KB RAM which is sufficient to run a system in it.

4.1.1 Sensor

The system has all GPIO available over the connector for future expansion of sensors. For demonstration of the prototype, present system has temperature sensor for generating data.

Temperature sensor

The PT100 temperature sensor is used to collect ambient temperature of surrounding.

Sensor interface

The sensor interface is a provision to connect new sensor to measure different physical quantity. Appropriate signal condition circuit is required before connecting it to the interface.

4.1.2 Data Logging

The Data-logging is a critical operation, as all the data are of equal importance. As there may be case, where system may not be present in network, for such particular case, device will log data to local storage like SDCARD. When connected to network, it should upload the data to the server.

Local storage:

For local storage capacity, some calculation is performed, which will predict total number of years data can stored in card if the system never connects to the network.

One record will have detailed of the device like assigned name, location, time interval of sampling, timestamp on which the data was collected. The records will be written at user defined interval, from 1 minute to 60 minute. So, with one record

of around 84 byte and using Equation 1 and Equation 2 to calculated minimum and maximum records logged per day and per year with total memory size they consumed,

Equation 1 Memory consumed per day

$$Total\ number\ of\ Record = \frac{Total\ Minutes\ per\ day}{Total\ size\ of\ one\ record}$$

Table 1 Records per day

	Number	*Size*	Units
Min record in a Day	24	1.96875	KB
Max records in a Day	1440	118.125	KB

Equation 2 Records and size generated per years

$$Total\ size\ consumed\ per\ year = \frac{S \times R \times 365}{1024}$$

S = Size of one record
R = Number of records generated per day

Table 2 Total size consumed per year

	Number	*Size*	Units
Min record in a year	8760	718.59375	KB
Max records in a year	525600	43115.625	KB

With the above information, calculation can be done for how many years will it take to consume, memory over the SDCARD's. With Equation 3 come conclusion for standard size SDCARD can be observed in Table 3.

Equation 3 Total years to consume memory over SDCARD

$$Number\ of\ Years = \frac{Total\ writable\ space\ over\ SDCARD}{Total\ size\ consumed\ over\ year}$$

Table 3 Total years to consumed memory over SDCARD

Card size in GB	*Max*	*Min*	*Unit*
2	48.64	2918.41	Years
4	97.28	5836.82	Years
8	194.56	11673.64	Years

16	389.12	23347.29	Years
32	778.24	46694.58	Years
64	1556.49	93389.16	Years

Note: The above values are subjected to change as per record size which is dependent to number of sensor interface to it and amount of data generated to upload.

Google spreadsheet (online logging)

The device will connect to the Google spreadsheet and log data on it. Spreadsheet architecture is roughly analogous to that of a relational database: Spreadsheet file implies database, Worksheet implies table, Columns implies fields/properties and Rows implies records. *Figure 3 Google Spreadsheet based data logger* shows the sample spreadsheet with few log values.

Figure 3 Google Spreadsheet based data logger

Google spreadsheet integrated tools for representation

The data logged in Spreadsheet can easily be represented in graphical ways for analysis and generating reports. The tools are very much user friendly and can easily be adopted.
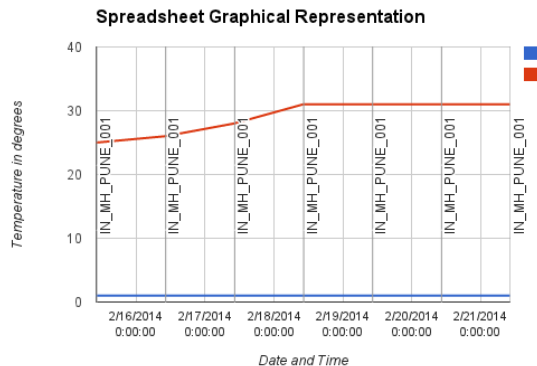

Figure 4 Google Spreadsheet values over Graph

Google Authorizing requests with OAuth 2.0

Google services are very secure services to provide user data. To access user data we need to follow some of the defined step by Google:

a) Before creating application, it should first register with Google. Google the provide information need later by application, such as client ID and a client secret.

b) Activate the Google Spreadsheets API in the Services pane of the Google APIs Console.

c) When application needs access to user data, it asks Google for a particular scope of access.

d) Google displays an OAuth dialog to the user, asking them to authorize application to request some of their data.

e) If the user approves, then Google gives application a short-lived access token.

f) Application requests user data, attaching the access token to the request.

g) If Google determines that your request and the token are valid, it returns the requested data.

4.1.3 Firmware

The Firmware will consist of RTOS and Application with Different TASK. The RTOS will have BSP library for the Board, RTCS library for TCP/IP related activity, MFS library for FAT32 File system to be installed in the SDCARD. The Application will have three tasks for Network, Files system initialization and last one for collecting sensor data, writing it on SDCARD and uploading it to online spreadsheet. The firmware general flow is as shown in *Figure 5 Flowchart*

```
┌─────────────────┐         ┌─────────────────┐
│ BSP initialization│         │  Create Socket  │
└────────┬────────┘         └────────┬────────┘
         │                            │
┌────────▼────────┐         ┌────────▼────────┐
│ RTOS Initiaization│        │      Bind       │
└────────┬────────┘         └────────┬────────┘
         │                            │
┌────────▼────────┐         ┌────────▼────────┐
│Initialize Networking│      │Collect Sensor's Data│
└────────┬────────┘         └────────┬────────┘
         │                            │
┌────────▼────────┐         ┌────────▼────────┐
│Acquire dynamic IP│         │Wirte on SDCARD File│
└─────────────────┘         └─────────────────┘
```

```
┌──────────────────────┐
│Connect to Google Server│
└───────────┬──────────┘
            │
┌───────────▼──────────┐
│  Create HTTP Request  │
│      Response         │
└───────────┬──────────┘
            │
┌───────────▼──────────┐
│     Send / Receive    │
└───────────┬──────────┘
            │
┌───────────▼──────────┐
│   Close Connection    │
└──────────────────────┘
```
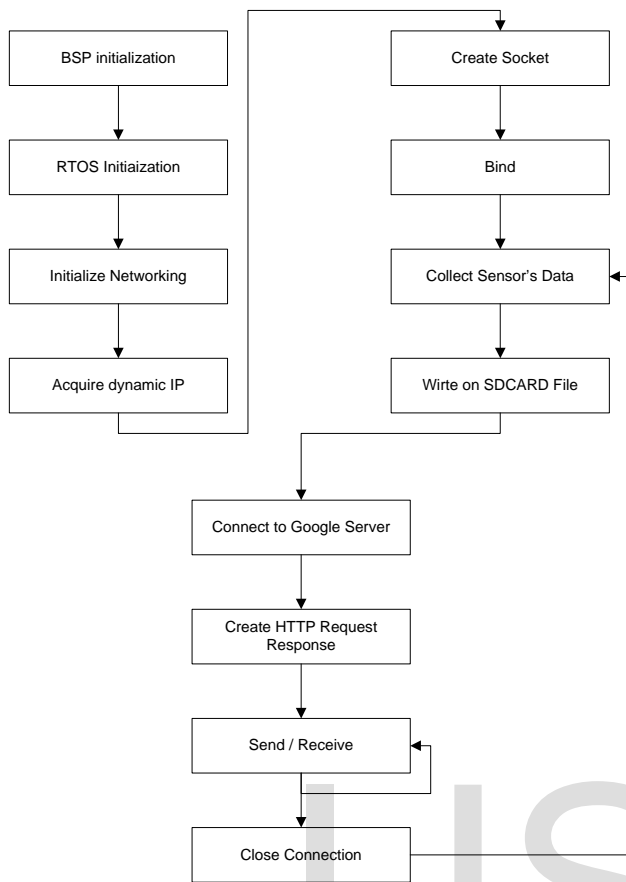
Figure 5 Flowchart

5 ADVANTAGE & DISADVANTAGE

Advantages

- Logging of Data at Global Level
- Data is secure with Google
- Google Services like fusion table, Draw, etc helps in representing information
- Sharing will become easy

Disadvantage

- Total of 400,000 cells & 256 columns per sheet
- Google changing security policy

6 APPLICATION

- Research associates can use this to application in Research Institute
- In Colleges for student attendance
- Home Monitoring like Gardening, Green House, Baby monitoring etc
- Application where data can be shown public

7 FUTURE SCOPE

1) Add support to Google product like Big Data.
2) Writing more Smart code in embedded system to understand the response from Server.
3) Making API at Embedded level for universal ac-

ceptance.
4) Proving the entire system on Power Line Communication.

Abbreviations and Acronyms

EVAL – Evaluation, IoT – Internet over things, RTOS – Real time operating systems, BSP – Board support package, RTCS – Real Time Communication Stack, BOM – Bill of Material.

8 CONCLUSION

The Data logger can successfully able to log sensor generated data over SDCARD and able to connect Google Spreadsheet to upload the same. With use of the Google Driver tools, it is easy to represent data in graphical format and with formulas analysis is also possible.

9 REFERENCES

[1] IEEE Paper – "**Real-Time Smart Meter with Embedded Web Server Capability**", Giuseppe Del Prete, Carmine Landi, Dipartimento di Ingegneria dell'Informazione, Seconda Universita di Napoli Via Roma, 29 - 81031- Aversa (CE) Italy.

[2] Pengzhi Xu , Weimin Zheng ; Yongwei Wu ; Xiaomeng Huang ; Chuncong Xu, "Enabling Cloud Storage to Support Traditional Applications", ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual, pp. 67 – 172

[3] Jiabin Tsai, Wei-Tek ; Tempe, AZ, U.S. ; Qi, Guanqiu ; Chen, Yinong ," Choosing cost-effective configuration in cloud storage", 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), 6-8 March 2013, pp. 1 - 8