AN ENERGY EFFICIENT GPS SENSING WITH CLOUD OFFLOADING

[1] KEERTHI SHEH ,[2] VIJAY GOPAL BAI
Dept of ECE,
Ajay Kumar Garg Engineering College, Ghaziabad.

ABSTRACT— The main aim of the research project is to design "a Cloud-Offloaded GPS (CO-GPS) solution that allows a sensing device to aggressively duty-cycle its GPS receiver and log just enough raw GPS signal for post-processing. It can collect temperature, humidity, light intensity, CO2 concentration and other air environmental information through sensors and get the current position (longitude, latitude and elevation) and timing (GMT) information through Global Positioning System (GPS) and map it to the cloud. Each node will then transmit the data to the monitoring station. The Global System for Mobile Communication (GSM)/ network will send the collected data to the data center server. The system uses a compact circuitry built around ARM Cortex, microcontroller programs are developed in Embedded C. Flash magic is used for loading programs into ARM Cortex. Location is a fundamental service for mobile computing. Typical GPS receivers, although widely available, consume too much energy to be useful for many applications. The location information can be post-processed when the data is uploaded to a server, we design a Cloud-Offloaded GPS (CO-GPS) solution that allows a sensing device to aggressively duty-cycle its GPS receiver and log just enough raw GPS signal for post- processing.

Keywords: ARM-Cortex, GPS, GSM, Cloud-offloading, IOT Webserver

I. INTRODUCTION

To sustain the Earth's environment while balancing human needs requires better decision making with more up-to-date information. Gathering accurate and timely information has been one of the greatest challenges facing by the researchers that must make these decisions. The Global Positioning System (GPS) helps to address that need. Data collection systems provide decision makers with descriptive information and accurate positional data about items that are spread across many kilometers of terrain. By connecting position information with other types of data, it is possible to analyze many environmental problems from a new perspective. Position data collected through GPS can be imported into IOT webserver, allowing spatial aspects to be analyzed with other information to create a far more complete understanding of a particular situation than might be possible through conventional means.

The research work provides a framework for designing a cost-effective and functional CO-GPS system, first discussing the general design considerations that should be evaluated before starting, followed by the implementation design using ARM Cortex, GPS and GSM.

II. LITERATURE SURVEY

There are four main reasons behind the high energy consumption of the GPS Module.

1) The time and satellite trajectory Information called Ephemeris are sent from the satellites at a data rate as low as 50 bps. A stand alone GPS receiver has to be turned on for up to 30 seconds to receive the full data packets from the satellites for computing its location.

2) The amount of signal processing required to acquire and track satellites is substantial due to weak signal strengths and Doppler frequency shifts. As a result, a GPS chip cannot easily be duty cycled for energy saving. In addition , it requires a powerful CPU for Post- processing and least – Square calculation.

3)The satellites move at a high speed. When a GPS chip is turned off completely for more than a few minutes, the previous code phases and Doppler Information are no longer useful, and the device must spend substantial energy to aquire the satellites.

4)Post –processing and least square calculation requires a powerful CPU.

III. GPS RECEIVING OVERVIEW AND GPS SIGNAL

A. GPS Receiving Overview

A GPS receiver computes its location by measuring the distance from the receiver to multiple GNSS satellites (also called space vehicles or SV for short). Ultimately, It needs to infer three pieces of information:

- A precise time T
- A set of visible SVs and their locations at time T
- The distance from the receiver to each SV at time T, often called the pseudoranges.

Typically, these are obtained from processing the signals and data packets sent from the satellites. With them, a receiver can use least-square (LS) minimization to estimate its location. To make this paper self-contained and to motivate our solution we give a brief (and much simplified) description of the GPS receiving process. We start with standalone GPS signals and then discuss GPS transmitted data and in particular a technique called coarse-time navigation.

B. GPS Signals

GPS satellites transmit signals that are received by the GPS receivers on the ground. These signals are then decoded and enable the receivers to provide the position information required. With limited power on the satellites, the signals transmitted are relatively low power, and in view of the bandwidth available, multiplexing techniques are used to provide access to all the signals that are available.The GPS satellites transmit a variety of signals that are picked up by the GPS receivers. These signals are relatively complicated but enable the system to operate in a very efficient fashion.

There are two primary frequencies that are used for the transmission of the GPS signal - both signals are in the UHF portion of the frequency spectrum. Additional GPS signals are used or being proposed as summarised below:

- ***L1 - 1575.42 MHz:*** This GPS signal is used to provide the course-acquisition (C/A) and encrypted precision P(Y) codes. It is also used for the L1 civilian (L1C) and military (M) codes on the Block III satellites
- ***L2 - 1227.60 MHz:*** This signal is used to carry the P(Y) code, as well as the L2C and military codes on the Block IIR-M and later satellites
- ***L3 - 1381.05 MHz:*** This frequency is used to carry information regarding any nuclear detonation (NUDET) event detected.
- ***L4 - 1379.913 MHz:*** This signal is being studied for use with additional ionospheric correction. This would considerably improve the accuracy.
- ***L5 - 1176.45 MHz:*** This GPS signal is being proposed for use as a civilian safety-of-life (SoL) signal.

I. GPS signal data

The data carried by the GPS signal contains three types of data:

- ***Pseudo-random code:*** This is an identification or ID code that identifies which satellite is transmitting the information.
- ***Ephemeris data:*** The almanac data on the GPS signal is used to carry information about the status of the satellite it also carries the current date and time which is used in the calculations for determining the position of the GPS receiver. This data is updated every two hours and is normally valid for four hours.
- ***Almanac data:*** The almanac data elements of the GPS signal provide information about the position of the satellite - orbit information about the satellite transmitting the information and all other satellites in constellation. This data is updated every 24 hours.

II. GPS transmitted data

Data transmitted in the GPS signals is split into frames to provide structure, and allow the receivers to be able to know where the beginning and end of messages are so that they can synchronise with the incoming signals and decode the data correctly.

A complete message is contained within a frame consisting of 15000 bit, transmitted at a rate of

Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

50 bits per second which takes 30 seconds to complete. It starts transmission exactly on the minute or half minute as determined by the atomic clock on each satellite.

Each frame is then further subdivided into five sub-frames, equal in length which takes six seconds to transmit and contain 300 bits.

Each sub-frame then contains ten words of 30 bits which take 0.6 seconds to transmit. Data required to be transmitted within the overall GPS signal is transmitted in set frames as detailed in the table below.

The GPS signal frames are split into sub-frames. In turn these have a specific structure to enable the system to be able to identify the data and keep the timing, etc.

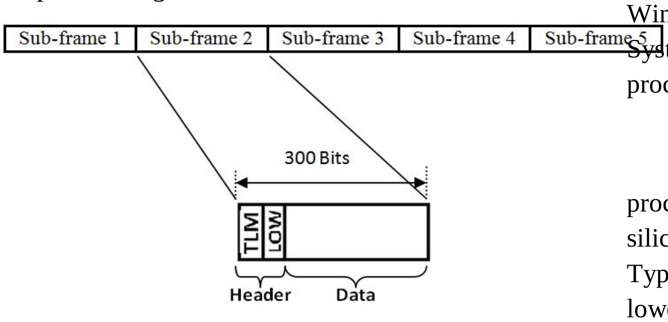


Fig 1: GPS Data Frame and Subframe Structure

SUB-FRAME	Information carried with in a SubFrame
1	Satellite clock, GPS time.
2-3	Ephemeris data
4-5	Almanac data

Table 1: Information carried with in SubFrame

IV. SELECTION OF A MICROCONTROLLER

ARM has developed quite a number of different processor products. The ARM processors are divided between the classic ARM processors and the newer Cortex processor product range. In addition, these processors are divided into three groups based on the application spaces:

Application Processors – High-end processors for mobile computing, smart phone, servers, etc. These processors run at higher clock frequency (over 1GHz), and support Memory Management Unit (MMU), which is required for full feature OS such as Linux, Android, MS Windows and mobile OSs. If you are planning to develop a product that requires one of these OSs, you need to use an application processor.

Real-time Processors – These are very high-performance processors for real-time applications such as hard disk controller, automotive power train and base band control in wireless communications. Most of these processors do not have MMU, and usually have Memory Protection Unit (MPU), cache, and other memory features designed for industrial applications. They can run at a fairly high clock frequency (e.g. 200MHz to >1GHz) and have very low response latency. Although these processors cannot run full versions of Linux or Windows, there are plenty of Real Time Operating Systems (RTOS) that can be used with these processors.

Microcontroller Processors – These processors are usually designed to have a much lower silicon area, and much high-energy efficiency. Typically, they have shorter pipeline, and usually lower maximum frequency running at over 200MHz. At the same time, the newer Cortex-M processor family is designed to be very easy to use; therefore, they are very popular in the microcontroller . The Arm Cortex-M processors utilize Arm Thumb-2 technology to offer superior code density compared to 8-bit,16-bit and fixed 32-bit architectures, reducing the memory requirements to maximize the usage of precious on-chip Flash memory. Thumb-2 technology supports a fundamental base of 16-bit Thumb instructions, extended to include more powerful 32-bit instructions.

Arm Cortex-M processors provide:

- High performance and efficiency
- Easy software development, as all Cortex-M processors are fully C programmable
- Compact data footprint, with support for 8-bit, 16-bit and 32-bit data transfers
- Fast and power-efficient algorithm processing with DSP extensions in Cortex-M4, Cortex-M7 and Cortex-M33

Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

- Extremely low power, the most energy efficient of all Arm processors. The Cortex-M0 achieves a power consumption below 4µW/MHz

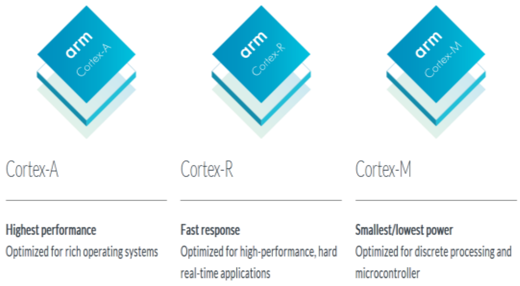
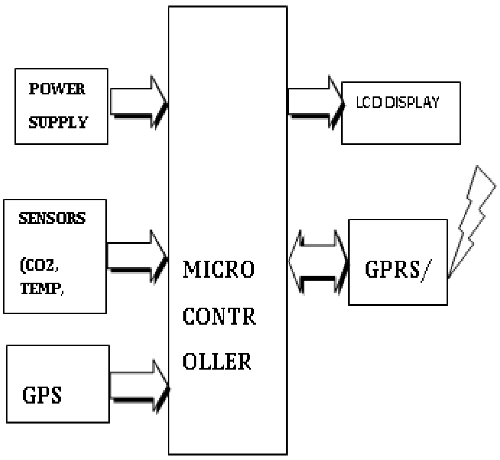


Fig 2: Classification of ARM Cortex processors

In our research project we selected the ARM Cortex-M series, STM32F030x4/x6/x8/xC microcontrollers Which incorporate the high-performance ARM Cortex-M0 32-bit RISC core operating at a 48 MHz frequency, high-speed embedded memories up to 256 Kbytes of Flash memory and up to 32 Kbytes of SRAM and an extensive range of enhanced peripherals and I/Os. ARM Cortex M0 offer standard communication interfaces two I2Cs, two SPIs and six USARTs, one 12-bit ADC, seven general-purpose 16-bit timers and an advanced-control PWM timer. The STM32F030x4/x6/x8/xC microcontrollers include devices in four different packages ranging from 20 pins to 64 pins. These features make the STM32F030x4/x6/x8/xC microcontrollers suitable for a wide range of applications such as application control and user interfaces, handheld equipment, A/V receivers and digital TV, PC peripherals, gaming and GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms, and HVACs.

V. BLOCK DIAGRAM & SCHEMATIC WORKING OPERATION



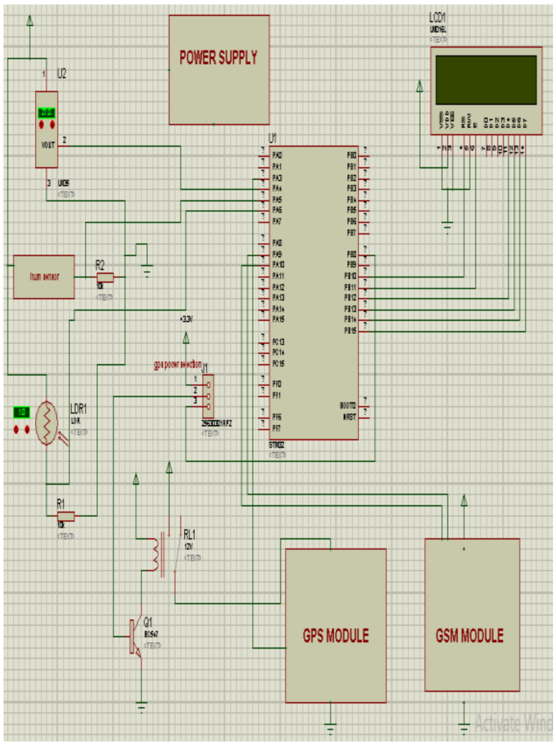


Fig 3 : Block Diagram and Schematic Implementation

In the proposed research work, an Energy Efficient CO-GPS system is designed with ARM Cortex, GPS and GSM. The design and implementation of sensor monitoring system is the model with the ability to perform data acquisition on temperature, humidity and LDR being interfaced with ARM Cortex . ARM can collect CO2 concentration, temperature, humidty, light intensity and other air environmental information through sensors and get the current position (longitude, latitude and elevation) and timing (GMT) information through Global positioning system (GPS) and map it to the cloud. Each node will then transmit the data to the monitoring station. The global system for mobile communication(GSM)/ network will send the collected data to the IOT webserver. When the GPS chip is operational, it captures the GPS signal through an antenna connected to either an active or passive antenna port. Depending on the application scenario, we selected to populate the board with various other sensors such as light, temperature, and humidity sensors and store sensor data together with the GPS signal. As the front-end of the CO-GPS solution, we developed a GPS sensing IOT platform that enables timeaccurate (millisecond granularity) GPS signal logging along with Temperature, humidity and LDR parameters. The front-end consists of a software

Journal of Engineering Sciences
Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

platform for GPS signal logging and accompanying PC-side software for performing parameter updates.

VI. RESULTS

Fig 4: Front End IOT WebServer

Cloud application is an application program that functions in the cloud, with some characteristics of a pure Web application. A web application operates entirely on a single device at the consumer's location it could be a mobile, laptop or tablet. A cloud application is stored on a server which is managed remotely and is accessed from the Internet through a browser interface. Application having security and every user having its own UserId and password

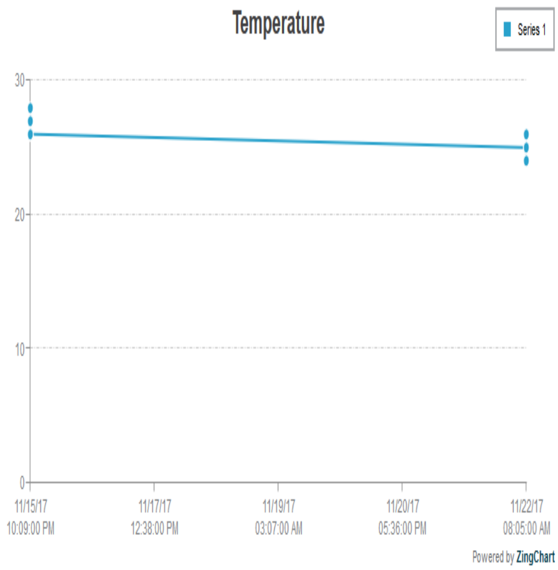


Fig5: Graphical Representation of Temperature

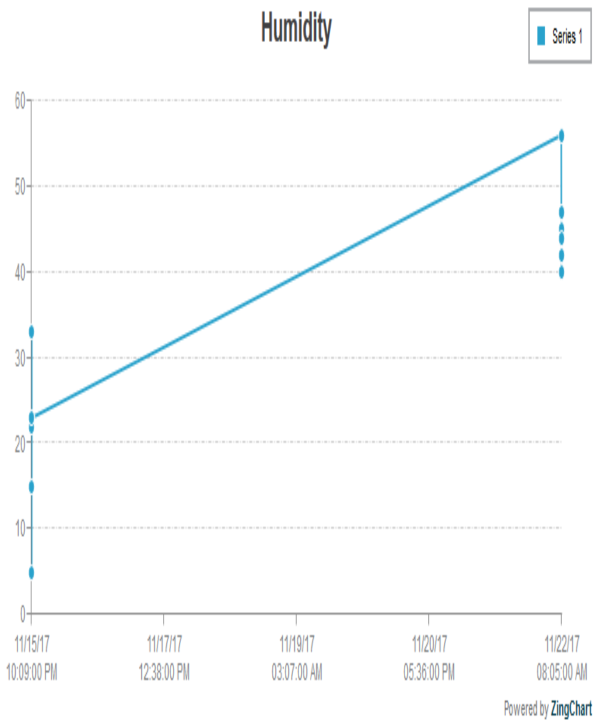


Fig6: Graphical Representaion of Humidity

Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

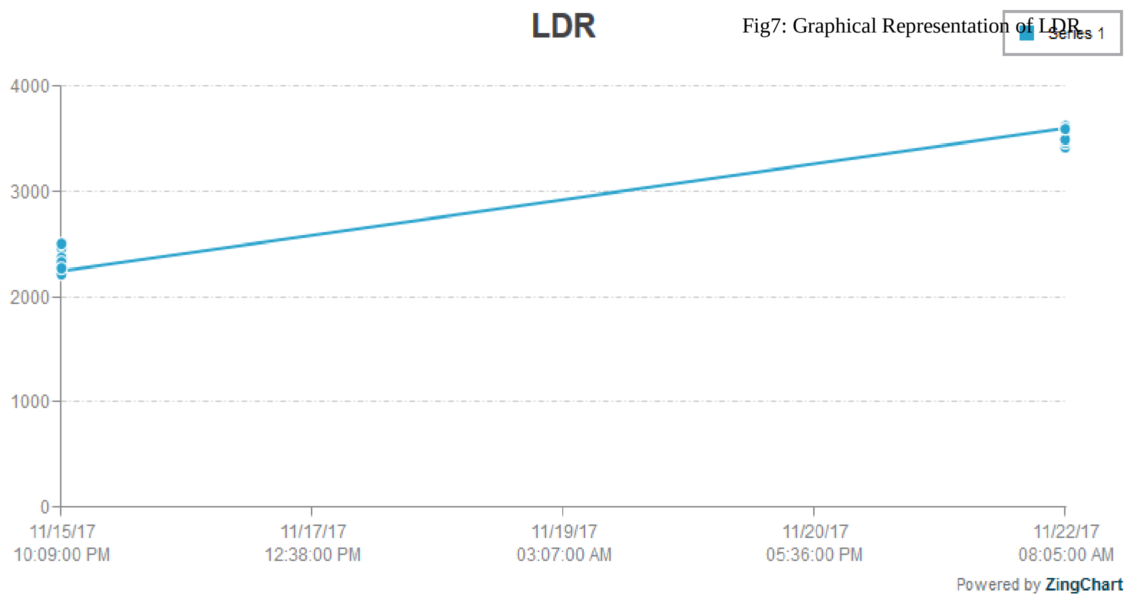


Fig7: Graphical Representation of LDR

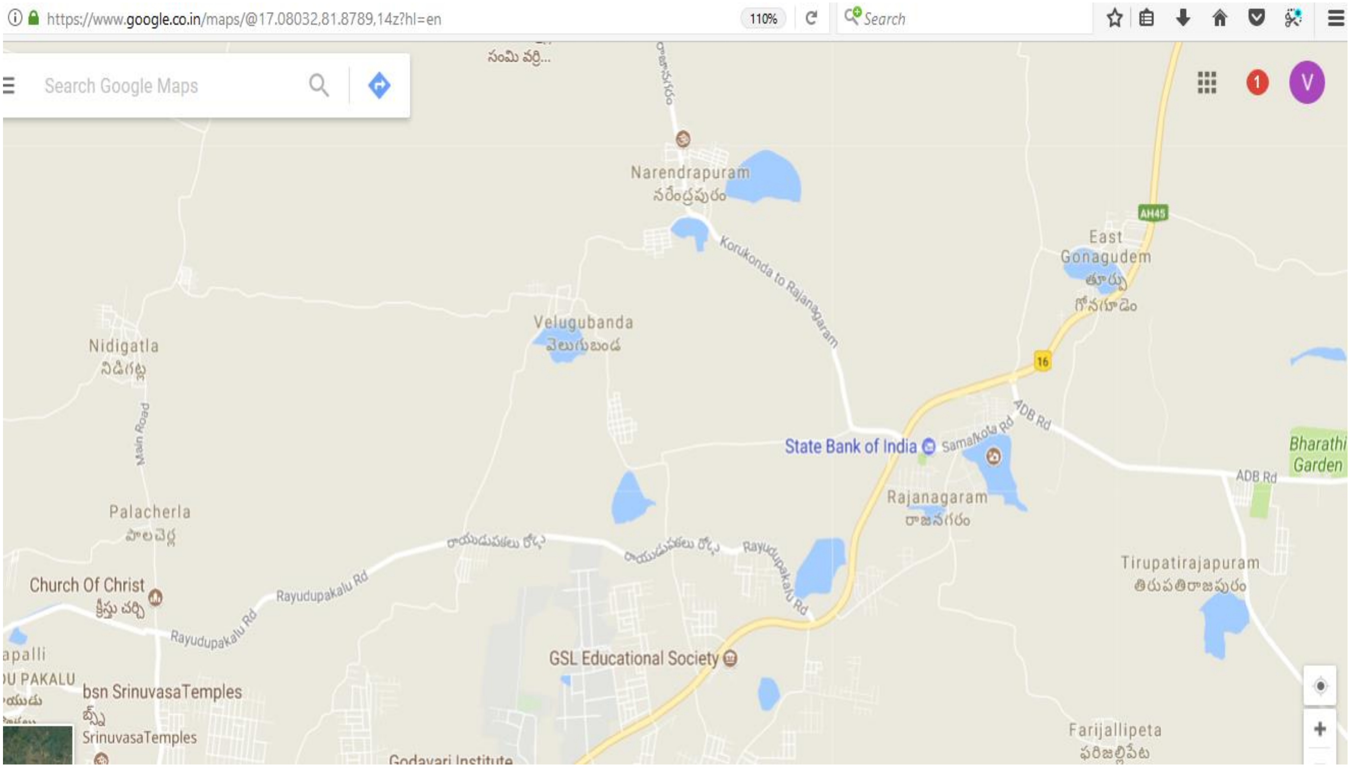


Fig8: GPS signal Logging and mapping

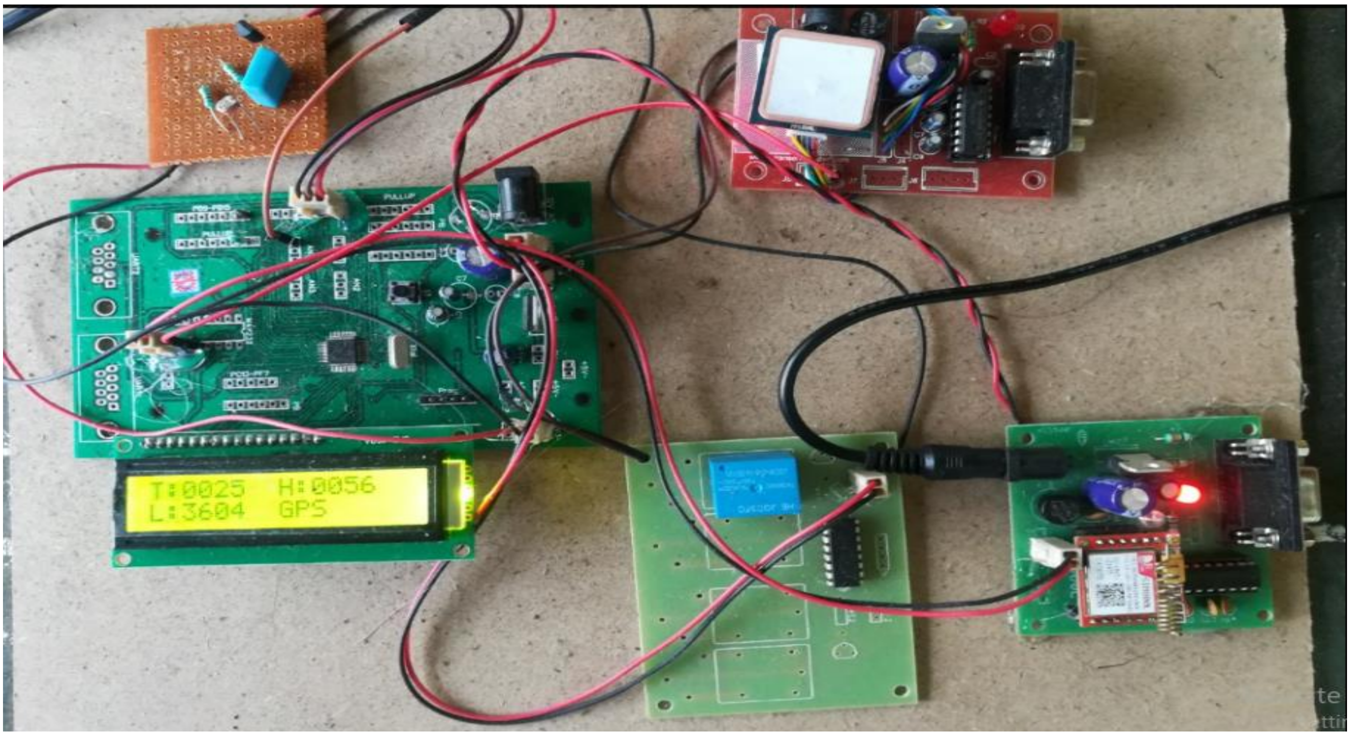


Fig9: RealTime Implementation of Hard ware kit

VII . CONCLUSION

GPS Cloud Offloading is proving beneficial to many real-life applications. We proposed an architecture using ARM Cortex and IOT web server as a cloud service, which smplifies automation system design. It is very suitable for real-time and effective requirements of the high-speed data acquisition and mapping in IoT environment. The ARM greatly simplifies the design of Interfacing circuitry, and makes the whole system more flexible and extensible.

REFERENCES

[1] Y. Fan, Y. Yin, L. Xu, Y. Zeng, and F. Wu, “IoT based smart rehabilitation system,” IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1568–1577, 2014.
[2] W. He, G. Yan, and L. Xu, “Developing vehicular data cloud services in the IoT environment,” IEEE Trans. Ind. Informat., vol. 10, no. 2, pp. 1587–1595, 2014.
[3] S. Li, L. Xu, X. Wang, and J. Wang, “Integration of hybrid wireless networks in cloud services oriented enterprise information systems,” Enterp. Inf. Syst., vol. 6, no. 2, pp. 165–187, 2012.
[4] Q. Li, Z. Wang, W. Li, J. Li, C. Wang, and R. Du, “Applications integration in a hybrid cloud computing environment: Modelling and platform,” Enterp. Inf. Syst., vol. 7, no. 3, pp. 237–271, 2013.
[5] L. Wang, L. D. Xu, Z. Bi, and Y. Xu, “Data cleaning for RFID and WSN integration,” IEEE Trans. Ind. Informat., vol. 10, no. 1, pp. 408– 418, Feb. 2014.
[6] M. T. Lazarescu, “Design of a WSN platform for long-term environmental monitoring for IoT applications,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 3, no. 1, pp. 45–54, Mar. 2013.
[7] L. Xu, “Introduction: Systems science in industrial sectors,” Syst. Res. Behav. Sci., vol. 30, no. 3, pp. 211–213, 2013.
[8] Z. Pang et al., “Ecosystem analysis in the design of open platformbased in-home healthcare terminals towards the internet-of-things,” in Proc. IEEE 15th Int. Conf. Adv. Commun. Technol. (ICACT), 2013, pp. 529–534.
[9] L. Benini, “Designing next-generation smart sensor hubs for the Internetof- Things,” in Proc. 5th IEEE Int. Workshop Adv. Sensors Interfaces (IWASI), 2013, p. 113.
[10] Y. Chen and V. Dinavahi, “Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems,” IET Gener. Transmiss. Distrib., vol. 7, no. 5, pp. 451–
463, 2013.
[11] A. Myaing and V. Dinavahi, “FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics,” IEEE Trans. Ind. Electron., vol. 58, no. 1, pp. 358–368, Jan. 2011.
[12] R. Dafali, J. Diguet, and J. Creput, “Self-adaptive network-on-chip interface,” IEEE Embedded Syst. Lett., vol. 5, no. 4, pp. 73–76, Dec. 2013.
[13] J. H.-L. Lu et al., “Alow-power, wide-dynamic-range semi-digital universal sensor readout circuit using pulsewidth modulation,” IEEE Sensors J., vol. 11, no. 5, pp. 1134–1144, May 2011.
[14] Z. Pang et al., “An RTOS-based architecture for industrial wireless sensor network stacks with multi-processor support,” in Proc. Ind. Technol. (ICIT), 2013, pp. 1216–1221.
[15] L. Xu, “Information architecture for supply chain quality management,” Int.J. Prod. Res., vol. 49, no. 1, pp. 183–198, 2011.
[16] W. Viriyasitavat, L. Xu, and W. Viriyasitavat, “A new approach for compliance checking in service workflows,” IEEE Trans. Ind. Informat. vol. 10, no. 2, pp. 1452–1460, 2014.
[17] S. Li, L. Da Xu, and X. Wang, “Compressed sensing signal and data acquisition in wireless sensor networks and internet of things,” IEEE Trans. Ind. Informat., vol. 9, no. 4, pp. 2177–2186, Nov. 2013.
[18] Z. Hanzalek and P. Jurcik, “Energy efficient scheduling for cluster-tree
wireless sensor networks with time-bounded data flows: Application to IEEE 802.15. 4/ZigBee,” IEEE Trans. Ind. Informat., vol. 6, no. 3,pp. 438–450, Aug. 2010.

Vol 10, Issue 8, Aug /2019
ISSN NO:0377-9254

[19] T. Wang, T. Shi, and B. Zhao, "The implementation of an intelligent IEEE1451.4 sensor data acquisition system," Microcomput. Inf., vol. 23,no. 4, pp. 1131–133, 2007.

[20] K. C. Lee et al., "IEEE-1451-based smart module for in-vehicle networking systems of intelligent vehicles," IEEE Trans. Ind. Electron.,
vol. 51, no. 6, pp. 1150–1158, Dec. 2004.

[21] V. Mattoli et al., "A universal intelligent system-on-chip based sensor interface," Sensors, vol. 10, no. 8, pp. 7716–7747, 2010.

[22] Smart Sensor Interface for IEEE 1451 [Online].
Available:http://www.jlminnovation.de/products/ieee1451,
accessed on Aug. 16, 2010.

[23] F. Ciancetta et al., "Plug-n-play smart sensor based on web service," IEEE Sensors J., vol. 7, no. 5, pp. 882–889, May 2007.

[24] L. Bissi et al., "Environmental monitoring system compliant with the IEEE 1451 standard and featuring a simplified transducer interface,"
Sens. Actuators A, Phys., vol. 137, no. 1, pp. 175–184, 2007.