ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

Development of a Two Wheeler IOT Product

[1] Clament John, [2]Abhiram Dronavalli, [3] Gerardine Immaculate Mary
[1] [2] [3] School of Electronics and Communication Engineering, VIT University, Vellore

***Abstract:*-- The automotive industry is going through a seismic shift towards electric and smart vehicles. But most of the development is being worked on the car platform, and the two wheelers (bike or scooter) are being left out. Two wheelers are a different field altogether when it come to making them smart. Most low-cost two wheelers have mechanical than electrical sensors. In this paper we attempt to address this issue with the minimal electrical signals we have (such as accelerometer, battery voltage, etc.) This data can be used to monitor the bikes conditions, like battery drain/strength, capacity, speed, acceleration, GPS location data etc. A prototype hardware for the two wheeler market which can do the above checks has been presented.**

***Index Terms*— Internet of Things(IOT), Smart Bike Device, Two Wheeler Tracking, Real Time Operating System(RTOS), Bluetooth Low Energy (BLE), MCU**

I. INTRODUCTION

The Internet of Things (IOT) is a growing sector in the industry. As we get accustomed to connectivity, through our portable devices such as mobile phones and laptops and most recently smart watches, we the consumers are asking for more data and automation with our everyday objects and utilities. The automotive is slowly moving towards a connected environment where the user can know much more about his/her vehicle, interact with the vehicle like never before and share his data to his/her family and friends.

Inspired by this we are designing and prototyping a connected bike / two-wheeler device that can track a bike and provide necessary critical event notifications such as tow, crash etc. to the user.

II. SURVEY

Since the device is developed to be used as a tracking and safety module for the two wheelers, the basic modules inside the device include:

1. Power Module
2. GPS Module
3. GSM/GPRS Module
4. Memory Module
5. Axial Measurement Module
6. Timing Module
7. Computation Modules

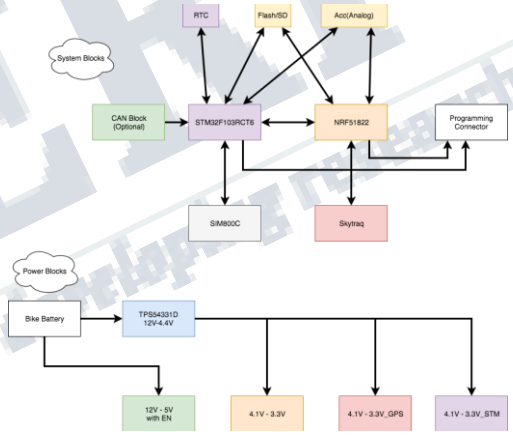

Fig 1: Block Diagram

III. MODULAR DETAILS

The device powers up through the two wheeler's battery. The power module then converts this voltage into usable voltage levels to power up the remaining modules and meet their current requirements. All modules except axial measurement and memory module work on 5V while others work on 3.3V logic. A buck converter is used to generate 5V and a voltage regulator generates 3.3V.

The GPS module helps keep track of the vehicle location in real time. The strings generated by this receiver is parsed into readable format and pushed to GPRS module.

Sensors such accelerometer are used to detect tow or crash incidents on the vehicle. These critical incidents are instantly notified to the user via SMS and through an android application.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

A memory module is used to keep track of the device's progress using data such as battery voltage, previous location, last timestamp and often due to poor network connectivity few packets are missed, this also stores those data packets so as to re-send it when network is stable.

Timing module is generating time stamp which is then appended with the data being sent to server so as to keep track of events in real time.

All these modules are linked to the microcontrollers which runs FreeRTOS to handle both data reception and transmission simultaneously. Out of the 2 microcontrollers STM32 is used for processing and analyzing data, and also in communication with the server while the other MCU uses Bluetooth Low Energy (BLE) to send real time updates to the user.

IV. DESIGN AND DEVELOPMENT

This device has been prototyped using multilevel modular design [1]. Through this approach each module can be developed individually and can later be integrated. This reducing the time and cost of development. Keeping modularity in mind each of the blocks in the system was developed individually and tested.

At the very core of the STM32 the freely implemented FreeRTOS by Real Time Engineers was used. The OS is scheduled between three threads to achieve a real time data capture and transfer. Thread 1 is responsible for gathering data such as speed and acceleration. This data is gathered from the GPS module. Thread 2 is responsible for managing the gathered data, error checking it and preprocesses it to make it into a packet that can be sent to the server. And thread 3 is the connectivity thread that takes the gathered packets and pushes the data to a remote server through a web socket connection. This is achieved through a set of AT commands the GSM module manufacturer has specified. The RTOS and FAT file system was developed using ST Microelectronics' STM32CubeMx which is a code generation software by STM to ease the development process in its ecosystem.

The second MCU gathers GPS data which is passed on to the STM32 through a SPI connection. At this time the STM32 passes on all the real time data it has gathered along with the current device parameters to the NRF so that it may be broadcasted over BLE.

The GPS receiver is connected to the NRF51822. The GPS module gathers data by communicating with GLONAS satellites which are global positioning satellites which provide location data, data time etc. This data is formatted into a string by the module and sent over UART. This string is in NMEA format and so the NRF51822 has a defined format to parse out the necessary data from. This data is stored in a data structure which is passed on to the STM32 over SPI, in which the STM32 is the master and NRF is the slave.

The data flow is as described in figure below.

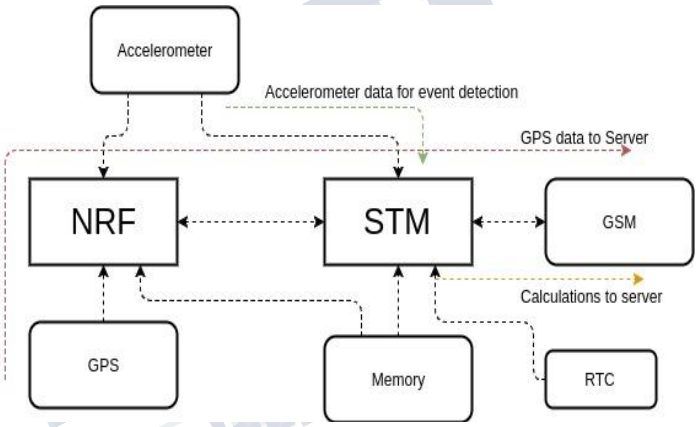

Fig 2: Flow of Data across the system

Every module interacts with the MCU's through RTOS. Explanation of every module's individual interaction with the RTOS threads is as follows:

The GSM module is connected to the STM32 through a UART peripheral. This module is responsible for the device's communication with the server. We initialize it using a set of AT commands as described in its application note [4]. After the device is initialized and it is connected to a network, we initiate a TCP connection. By default, when a TCP connection is initiated by this GSM module it is set as HTTP. But for our application we upgrade this HTTP connection into a web socket connection using a HTTP upgrade request. A web socket connection is superior in this case as our device needs to post at a very high frequency and the HTTP overheads for every PUSH or GET frame will slow down the process.

Predicting tow and crash of the vehicle is vital, using accelerometer module continuous axial orientation of the vehicle is monitored. Using ADC increases latency of the ALU, thus DMA was incorporated to process

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 5, May 2017

the ADC conversion in parallel with the RTOS threads. Through continuous monitoring of these axial values and measuring the rate of change along each axis we can predict the condition of the vehicle.

Memory module runs on a File Allocation Table- File System (FAT-FS). This file system being a universally followed logic to store and access data, hardware changes does not affect its functionality [3]. Thus keeping the modular nature of the memory module intact. Also, by using open sourced FAT-FS continuous updates and support provide better performance. Since SD card was used as the memory module, Flash Translation Layer and garbage collector would already have been implemented in the SD card, thus reducing development time.

Real time clock was used as a source of timestamp for all timing specific data packets such as emergency alerts to the user. This timestamp is appended with the data packets being sent to the server.

Before integrating all functions of these blocks each peripheral connected to the MCU is individually tested so as to reduce debugging efforts.

Finally, a printed circuit board was manufactured which included all these modules explained above. Several evaluation and development kits by various vendors such as Texas Instruments, microElectronika, Waveshare Electronics, Nordic Semiconductor, Adafruit Industries and Sparkfun Electronics were used.

Fig 3: Prototype Smart Two Wheeler IOT Device

V. TESTING

The prototype design has been integrated after individual modular level checks [2]. This device was tested along the roads of Mumbai for over 45km. The initial setup of the device has been shown below.

Fig 4: Device connection with Bike

Also, Fig 5 shows the tracked path of the vehicle on one such small trip.

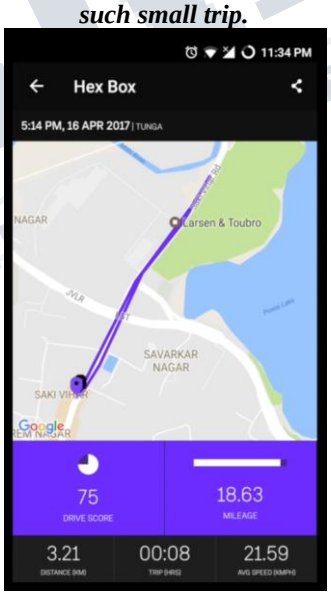

Fig 5: Carnot Dashboard

Through the complete period of this device testing we found out that the vehicle's battery discharged quickly.

VI. OPTIMIZATION

The device was set at full functionality throughout its running period. This was seen as the major reason for the power drain of the bike's battery. And so a battery management logic had to be incorporated into the firmware.

In this logic, the battery voltage of the device is monitored at regular basis. And if the vehicle is in OFF state, that is the vehicle is parked, only then the power management logic has to be implemented. This is so, because when the vehicle is running the alternator recharges the battery at a rate higher than the discharging rate of the device. Also being a bike tracking device the GPS coordinates do not change when the vehicle is parked, only the critical events must be monitored. Thus to consume lesser power the firmware will switch off some of its modules in case it detects the vehicle is parked and not running.

Power management is achieved by switching OFF the GSM/GPRS and the GPS. If the battery voltage drops below a certain threshold, power management is initiated. By this a fully functional GSM module which would consume 14mA will only consume 0.8mA in sleep mode. This optization is expected to increase the life time of the bike battery by almost a day.

VII. SUMMARY

This prototype, the bike product was developed using the modular approach described above. Each module of hardware and firmware was designed individually and tested so that we come across its problems beforehand, thus we are able to face the design issues in a modular level. Thus it enables us to debug problems easily. And only after a satisfactory result has been seen in the modular approach we moved on to develop the prototype bike tracking device.

Also we extend our gratitude to thank Carnot Technologies for providing us with the necessary platform to test this prototype.

REFERENCES

[1] S. Schulz, K. J. Buchenrieder and J. W. Rozenblit, Multilevel testing for design verification of embedded systems, in IEEE Design & Test of Computers, vol. 19, no. 2, pp. 60-69, March-April 2002.

[2] H.P.E. Vranken, M.F. Witteman, and R.C. Van Wuijtswinkel, Design for Testability in Hardware-Software Systems, in IEEE Design & Test of Computers, vol. 13, no. 3, Fall 1996, pp. 79-87.

[3] Elm Chan FAT-FS Front page for the open source implementation of FAT- FS. Retrieved from http://elm-chan.org/fsw/ff/00index_e.html

[4] SIM800 TCP/IP application note. Retrieved from https://cdn-shop.adafruit.com/datasheets/sim800_series_ip_application_note_v1.00.pdf