Real Time Speed Monitoring of the Automobile using Open Source Alljoyn Frame Wok

Parthasarathy P
Research Scholar
VIT University, India

Abstract— In the last decade, one of the real time problems we are facing is the automobile accidents that occur due to the fatigue driving of drivers. We see such incidents daily which raise the question of our safety and security of the people. Therefore, there is a need of real time monitoring of the speed of the automobile as a precaution to avoid such accidents and ensure safety. The internet of things is used to exchange the data obtained by the embedded devices such as sensors, actuators in machines and many other physical objects that are intelligently connected to each other. Internet of Things has the capability to produce solutions to improve efficiency, education, energy and a lot of other aspects with minimal human intervention. In the proposed system, the system provides the monitoring of the speed of the automobile on the real time basis and transferring the data through an open source framework called Alljoyn. To monitor the speed, an accelerometer ADXL345 is used along with the Raspberry Pi 2. The sensor data obtained from the accelerometer ADXL345 is then transferred to other devices.

Key words: Alljoyn Framework, Linux, Raspberry Pi, Speed Monitoring, Wi-Fi

I. INTRODUCTION

Speed monitoring and analysis has been an active research and engineering topic for more than a decade. Main information acquired is the speed of the vehicle and the vehicle type identification. Speed data can be used for a variety of purposes including safety, driving pattern of individuals and roadway pavement use.

A. Acceleration

The rate at which an object changes its velocity is acceleration. If an object is changing its velocity, then its accelerating. Acceleration is not defined by moving fast, a person can be moving very fast but still not accelerating. Acceleration depends on changing the speed of the object.

B. Constant acceleration

If the velocity of the object changes by a constant amount every second, then it is called constant acceleration. An object with a constant acceleration and constant velocity are entirely different.

C. Changing acceleration

If the velocity of the object changes randomly every second, then it is changing acceleration.

D. Free fall motion

The distance travelled or the time taken is not a constant value for the accelerating because of their changing velocity. A free falling object constantly accelerates as it falls at a velocity approximately 4m/s in the first second. Approximately 14m/s in the second, 24m/s in the third second and so on. A free falling object which accelerates at a constant rate covers different distances in each consecutive second.

E. Average acceleration

It is calculated as the change in the velocity of an object in a given interval of time.

Avg. acceleration = Δvelocity/time taken =Vf-Vt/t

Acceleration is a vector quantity and the direction of the vector depends on two things:

- The object is speeding up or slowing down.
- The object is moving in the + or – direction.

The general principle is if an object is slowing down, hen its acceleration is in the opposite direction of its motion.

F. Acceleration of gravity

The acceleration for any moving object under the sole influence of gravity is known as acceleration of gravity. It is denoted by a symbol -g. The numerical value for the acceleration of gravity is accurately 9.8m/s/s. This value primarily depends upon the altitude. [4]

$$a=\Delta v/t = -9.8m/s/1s$$

where a= acceleration
Δv=change in the velocity
T= time taken

It offers various kinds of connectivity from services and devices. The Raspberry pi is a low cost, small sized computer that can be used to plug in to computer monitor or television, keyboard. Programmers can develop the script in python and its main operating system is Raspian. The heterogeneous networks are inter connected using IOT. The embedded devices with specific functions occupy a large proportion in IOT. Alljoyn is an open source framework that provides set of services that connects products, devices and software applications. Alljoyn is divided into two parts like Alljoyn Standard Client (AJSC) and Alljoyn Thin Client(AJTC). The messaging layer uses User Datagram Protocol (UDP) and TCP. To abstract the necessary native system functions, there is a very thin layer. Only the minimum required API to the bus attachment is exposed. The first step for AJTC is to discover and connect with the daemon. Once the connection is set up, the data of any kind ca be transferred using Wi-Fi.

II. METHODOLOGY

The ADXL345 accelerometer is connected to the Raspberry Pi. The communication between the accelerometer and the Raspberry Pi takes place through the I2C Protocol. The ADXL345 Accelerometer is programmed in python to send the data i.e. Speed in the form of g to the raspberry pi. Raspberry Pi and Linux machine are connected to the same Wi-Fi network. The accelerometer data from the Raspberry Pi is transferred to the Linux machine through Alljoyn Framework. Alljoyn Framework installed in the Raspberry Pi acts as a server device. The Linux machine with the Alljoyn serves as a client. Both the devices can be either server or

client or both. The data is transferred using Software Bus. When the server has to send the sensor data, it advertises using About Announcements. The client joins the session if it wishes to and obtains the data.

A. Accelerometer ADXL345 (Fig.1)

The ADXL345 is chosen because it is capable of sensing the static acceleration of gravity in tilt sensing applications and dynamic acceleration caused due to motion or shock. The presence or lack of motion is detected by activity or inactivity sensing that is obtained by comparing the acceleration on any axis with the user-set thresholds. Single and double taps are detected by tap sensing mechanisms. Similarly, the free fall sensing is activated when the device s free falling. The ADXL345is a thin wire, ultra-small, three axis accelerometer which has a high resolution up to 13 bits and is capable of measurement up to + or -16g. The output of the accelerometer is obtained in the form of g and digital output data is formatted as16 bit twos either SPI (Serial Peripheral Interface) interface or I2C interface.

1) SPI Interface:

SPI is a single master communication protocol which has one central device that communicates with the slaves. When the master wishes to send or receive data, it sends a low signal to the SS pin.

2) I2C Interface:

The ADXL345 Breakout has an I2C address of 0x53. It can share the I2C bus with other I2C devices as long as they have their own unique address. The Theory of operation is as follows: Differential capacitors are embedded in the sensor which consists of independent fixed plates and plates attached to the moving mass using which the deflection of the structure can be measured. The differential capacitor is unbalanced and the proof mass is deflected by the acceleration due to which the sensor output is obtained whose amplitude is proportional to the acceleration. The magnitude and polarity of the acceleration is determined by phase-sensitive demodulation.

3) SINGLE TAP:

When a single acceleration event occurs, and the value is greater than that of the threshold value present in the THRESH_TAP register then, SINGLE_TAP bit is set.

4) DOUBLE TAP:

When double acceleration event occurs, and the value is greater than the threshold value, then the DOUBLE_TAP bit is set.

5) ACTIVITY:

When the acceleration is greater han the value present in the THRESH_ACT register that is experienced on any of the axis set by ACT_INACT_REG then, the activity bit is set.

6) INACTIVITY:

It is set when the acceleration value is less than that of the threshold in act register for more time than that is present in the TIME_INACT register on any axes that is participating.

7) FREE FALL:

When the acceleration value is less than of the threshold value of the free fall register is experienced for more time than that is specified in the free fall register, on all the participating axes, then the free fall bit is set. During the free fall interrupt, all axes participate and are logically AND for the period that is smaller than that of the inactivity interrupt. DC-coupled is the mode of operation. The operating range of the sensor is set by the set Range () function. Lower values will have more sensitivity whereas higher values will have wider measurement range.

Valid range constants are:

- ADXL345_RANGE_16_G
- ADXL345_RANGE_8_G
- ADXL345_RANGE_4_G
- ADXL345_RANGE_2_G

B. Raspberry pi

It is a small sized computer which connects al inputs and output devices such as monitor, keyboard and mouse and can be used to perform operations as performed on a normal PC. It runs on programming languages such as C, Java, Python and many more basic programming languages.

- Single Board Computer running at 900 MHz powered by a Broadcom BCM2836 Arm7Quad Core Processor.
- 1GB RAM
- 40 pin extended GPIO
- 4 x USB ports
- 4 pole Stereo output and composite video port
- Full size HDMI
- Raspberry pi camera can be connected at the CSI camera port
- DSI Display Port for connecting the Raspberry Pi touche screen display.
- Micro SD Port for loading the operating system. Micro USB source.

C. Wi-Fi module

Wi-Fi Provides connectivity to the electronic devices through wireless LAN (WLAN) network. Devices which use this technology are personal computers, smart phones, digital laptops, video game consoles, tablet computers and so on. The devices that are compatible with Wi-Fi can connect to the internet via WLAN network and an access point which is also wireless. Such an access point is called a hotspot which has a range of around 20 meters indoors. Security in Wi-Fi is less compared to the wired connections like Ethernet since the intruder does not need any physical connection.

III. ALLJOYN FRAMEWORK

Alljoyn enables ad hoc, proximity based, peer to peer networking between devices and applications. Alljoyn makes peer to peer easy and frictionless. It discovers devices and applications around us. It adapts to devices that are not stationary. It manages transports like Bluetooth and Wi-Fi and message routing across them. Alljoyn interoperate across disparate operating systems and bearers. It is capable of exchanging information in a secured manner. The framework does not require cloud function, where as it is capable of running on the local network. If there is a need of the cloud, Gateway Agent is supported by the Alljoyn. One more advantage of this framework is that it reduces the risk of hacking by reduction of the attack surface. Hence the gateway agent is directly connected to the internet avoiding the more number of devices connecting to the internet. The communication between the apps and devices is very efficient, fast and secure. Alljoyn is designed to be easily compatible to all kinds of hardware. It is a distributed software bus in which

each bus runs as a bus daemon. All applications communicate directly with the daemon and it handles all cross device communication. Applications use a client direct to interact with the daemon. Bus Formation is ad hoc. It is based on discovery of applications or services proximally. It also contains link specific discovery mechanisms. Alljoyn framework is responsible for establishing connections between various Alljoyn applications. Any application, if it is willing to offer its service has to advertise itself using About Announcements. When the other remote connections discover this application with its unique name then, it can create a session called Join Session. The application which offers the service has the option of either accepting or rejecting the join session. Sessions are created on various ports. Different ports allow variety of topologies. The sessions can be of two types. They are as follows:

- Point to Point: This allows only one to one connection.
- Multi-point: It allows a group of applications or devices in a single session.

These are the common services required by devices to provide a set of interfaces for all kinds of devices to communicate and interoperate with one another. Few among the supported base services are mentioned below:

A. Control Panel:

It permits the devices to advertise a virtual control panel and control it remotely.

B. Audio Streaming:

Synchronized audio playback is possible in either one or many sinks. This provides a direct access to Alljoyn core APIs if it is programmed to Alljoyn Core Library or it can be programmed to Alljoyn Service Frameworks Libraries to provide higher functionality.

C. Configuration:

It helps to configure certain characteristics of an application or device, such as a friendly name.

D. Thin and standard:

Alljoyn Framework provides two variants i.e. Standard for the non-embedded devices like Android, iOS and Linux. Thin for all resource constrained embedded devices like Arduino, ThreadX, and Linux with limited memory.

E. On boarding

It provides a steady way to bring a new device into the Wi-Fi network.

F. Notification:

Real time notifications like text, audio or image URLs can be sent an received by devices in the same Alljoyn network.

G. Programming models:

Since the applications have to be compatible with different devices, the app code will be written using Alljoyn Framework APIs. By programming directly to the Alljoyn core APIs, the application can implement to its own service. While doing that it has to enable the ad hoc interactions between the other Alljoyn devices to follow the events and actions.

The framework runs on the local network enabling the devices and apps to advertise and discover each other. The framework comprises of apps and routers and they can communicate with each other whereas apps can communicate with other apps only through the router. If an app uses its own router, then it is called "bundled router" Since it is bundled with the app. It is usually found in mobile operating systems like android and windows. The router is called a standalone router when multiple apps on one device use one router. This is usually common on Linux systems where Alljoyn router runs as daemon process and other Alljoyn apps connect to the Standalone router. If an app uses a router on a different device, then they have to use the thin variant of the Alljoyn framework like embedded devices.

IV. ANDROID STUDIO

Android Studio, an IntelliJ's powerful code editor, is the official IDE for Android app development. It has extra-ordinary developer tools and many more features to enhance the productivity as shown below.

- It is capable of multiple APK file generation and building variants.
- Support for drag and drop editing is possible due to a user-friendly layout editor.
- Shrinking of the resources is possible by using Gradle and code shrinking by ProGuard.
- Common app features can be built easily with the help of code templates.
- Easy to integrate Google Cloud Messaging and App Engine with the built in support for Google Cloud Platform.
- Gradle based build system defines flexibility together with the capability of defining common standards for Android builds.
- Gradle supports multi-project builds from both IDE and the command line. It also supports the new. aar binary bundle format for library project.

V. RESULTS

A. Communication between raspberry pi and Linux(Fig.1)

It has the ability to send the Alljoyn Signals on a connected session. The application advertises a well-known name with a prefix of "org.alljoyn.bus. samples. Chat" It will then be followed by the name of the chat which depends on the user and can be displayed by the UI. Each chat room will accept sessions on the session port 27. The chat application in the Alljoyn runs different on each platform depending on the platform and the devices connected. Each application should have a role of either a client or a service or both and may need to join its own session if necessary, depending on the type of the platform.

B. Communication between server and client in ubuntu (Fig.2)

Alljoyn is compiled in Ubuntu as shown in the previous chapter. To communicate between the server and the client while both present in the Ubuntu, there has to be two files for server and client which consists of certain commands to make it happen. The server starts a session by sending a

signal with a -s flag. If the client wants to join the session, then it sends a signal with a -j flag and joins the session.

C. Data transfer between raspberry pi and ubuntu (Fig.3)

Similar to the chart application, data obtained from the accelerometer connected to raspberry pi can transfer to the Linux device wirelessly through alljoyn

D. Communication between Linux machine and android device

The simple server listens for any connections, the simple client joins the simple service sessions and a text message is sent from the simple client to the server via Bus method invoked on a Simple Service Bus Object is displayed on both Linux and android.

VI. CONCLUSION AND FUTURE SCOPE

The objective of the project was to establish communication between raspberry pi, Ubuntu and AllJoyn. This report shows the establishment of communication between the raspberry pi and Ubuntu, communication between Ubuntu and android application wirelessly using AllJoyn Framework. The Communication is not capable using the LAN, hence we connect all the devices to the same Wi-Fi network. Currently AllJoyn Framework supports Wi-Fi, Bluetooth which minimizes the range. Gateway Agent is being developed to add the cloud services to the framework. AllJoyn Framework is more valuable when it it is capable of working over the internet regardless the network. . AllJoyn Framework can be used in the houses, school buildings, parking lots and so on for the safety and ease of life.

VII. FIGURES

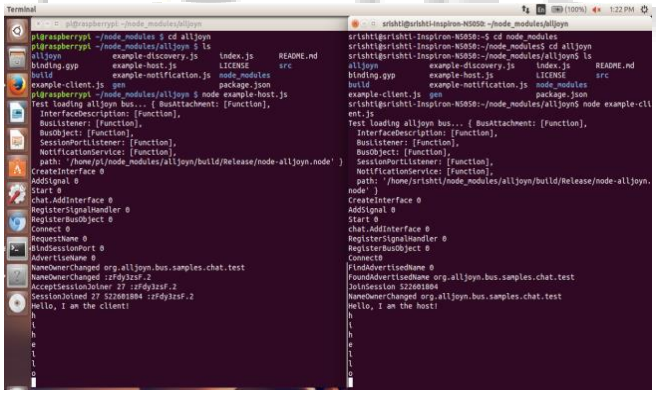

Fig.1 Communication between R-pi and Linux

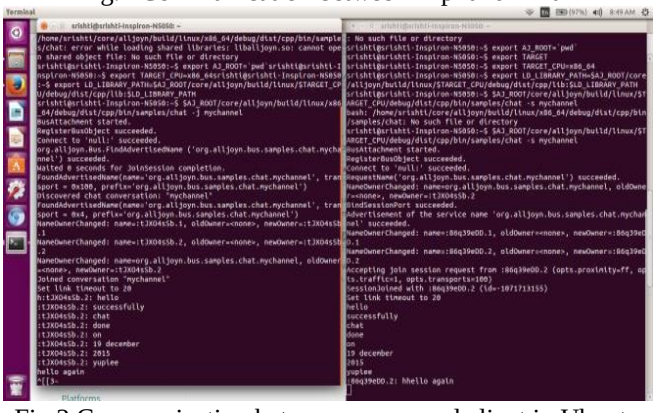

Fig.2 Communication between server and client in Ubuntu

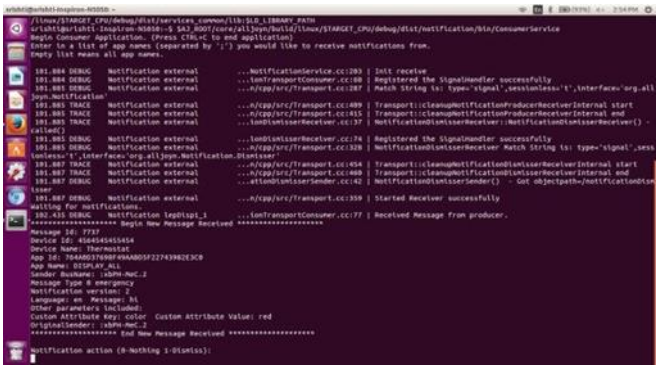

Fig.3 Data transfer between R-pi and ubuntu

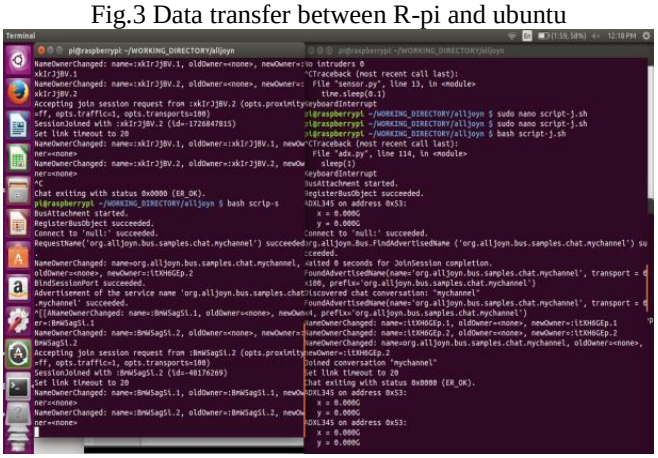

Fig.4 Communication between Linux and android

REFERENCES

[1] Kochlan, M., M. Hodon, L.Cechovic, J. Kapitulik and M.Jurecka (2014, September). WSN for traffic monitoring using Raspberry Pi board. In*Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on* (pp. 1023-1026). IEEE.

[2] Lamine, H., and H. Abid, (2014, December). Remote control of a domestic equipment from an [3].Android application based on Raspberry pi card. In *Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2014 15th International Conference on* (pp. 903-908). IEEE.

[3] Zhao, C. W., J. Jegatheesan and S. C. Loon (2015). Exploring IOT Application Using Raspberry Pi. *International Journal of Computer Networks and Applications*, *2*(1), 27-34.

[4] Zheng, K., T. Lv, Y. Li and Y. Lu (2014, November). The analysis and implementation of AllJoyn based thin client communication system with heartbeat function. In *Cyberspace Technology (CCT 2014), International Conference on* (pp. 1-4). IET.

[5] https://allseenalliance.org/framework/documentation

[6] https://allseenalliance.org/framework/documentation/learn/architecture

[7] https://allseenalliance.org/framework/documentation/learn/architecture

[8] http://cdn.mos.techradar.com/art/desktop_pcs_and_macs/Raspberry%20Pi/Raspberry%20Pi%202/raspberry-pi-2-hero-970-80.JPG

[9] http://www.emartee.com/Images/websites/emartee.com/ADXL345.jpg

[10] http://image.slidesharecdn.com/ethiotmsg20120905-130702081240-phpapp02/95/the-multi-service-gateway-concept-in-m2m-internet-of-things-iot-solutions-12-638.jpg?cb=1372753252
[11] http://image.slidesharecdn.com/2014ciscowimelfrinkiotworldforumfinalpdf1copy-141016082814-conversion-gate02/95/wim-elfrink-keynote-slides-10-638.jpg?cb=1413448218
[12] http://physics.info/acceleration/
[13] http://www.physicsclassroom.com/class/1DKin/Lesson-5/Acceleration-of-Gravity

