International Journal of Research in Advent Technology, Vol.7, No.4S, April 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Automatic Classification Of Moving Object in Wireless Multimedia Sensor Networks Using IoT

[1]B.Hemalatha, [2]Dr.Hari Shanker Srivastava,[3]V.Vinay Kumar
[1,2,3]Dept of ECE, Asst.Professor
Anurag Group of Institutions

Abstract-The Internet of Things (IoT) offers the user seamless interoperability and connectivity between devices, systems, services, disparate networks, and in particular control systems. End users expect to connect quickly and transparently via any endpoint device, be it a phone, tablet, wearable, TV, car, or other system that is Internet enabled. Smart IoT devices can serve as versatile control system interfaces, allowing for rapid response and potentially ubiquitous access. Using these technologies for user access control at IoT endpoints. There is no security to access the endusers.The proposed method represents a model that combines the capabilities of smart IoT devices with control system gateways using real-time challenge response for secure control operations. The proposed solution using both endpoint and gateway devices. End-to-End (E2E) IoT solutions comprise a combination of two basic platforms. One is the Edge Application Platform (EAP) consisting of sensors and actuators which are connected to an IoT gateway. The other is a Cloud RealTime Identity Monitor Service Platform (CSP) comprising hosted services for connectivity, manageability, and security. The EAP and CSP are designed to be futurecompatible with three primary objectives. First, they must be functional over large distances. Second, they must be able to be managed remotely and securely.

1. INTRODUCTION

The evolution of the industrial Internet of Things (IoT) [1] and Industry 4.0 [13], [14] creates the possibility of connecting computer automated control systems for remote monitoring and rapid response to events requiring real-time handling. In the past, a facilities manager had to physically attend to a control system often creating a delay in action leading to damages. In the IoT world, control commands can be issued instantly from a client device,suchasPC,tablet,orcellphone,toanywhereina given facility, such as an oil rig or power station. This ability to respond immediately from any location can prevent costly damages or possibly disasters such as the 2015 Oil Spill in Santa Barbara, CA, USA [5], which heavily cost both the company and the environment. With this new ability, however, comes an increased security risk, notably by an attacker obtaining unauthorized access and changing a system, issuing inappropriate commands, causing sabotage or system outages.

Today, control systems are increasingly being connected to the IoT world via gateways [6]. These control system gateways allow for simultaneous control of multiple industry environments, including factories, automated facilities, medical, and transportation systems. Gateway devices also simplify and unify the management interfaces of complex control devices [7]. Control system gateways with sufficient compute power can also be used with modern IoT client devices to help manage the security challenges of remote access and control. Using cloud services with a control system gateway for managing security is an increasingly used approach. Due to the growth in gateway computational power and increased storage/memory space as well as improvements in client device security capabilities advanced gateways are now able to manage the remote access security requirements directly from IoT client devices. Having both a cloud option and a directly connected option opens the control systems and automation arena to many new capabilities, modalities, and also introduces new challenges.

This paper has three main technical sections. Section II reviews potential security challenges and identifies key problems with user authentication and authorization. Sections III and IV present a model of how control system gateways can be used to meet these security challenges, focusing particularly on a strategy that connects IoT devices directly with control system gateways. The security is strengthened with multifactor authentication and authorization over an encrypted communication channel, and the addition of “Real-Time Identity Monitoring” to continuously assure a valid user identity. Section IV also presents both a strategy and a challenge to the research community to refine the overall techniques of using a combined IoT client device and control system gateway for fast and secure operations. Section V shows an example solution by presenting Intel’s End-to-End (E2E) IoT architecture. E2E supports complex gateways as well as smaller simpler gateways and is based on Intel processor technologies.

With embedded systems fast expanding its reach, subject matter related to this field is available in abundance. While working on this project we have studied matter from various sources such as books, online articles and reference manuals. The knowledge gained from this activity has been of great help to us in understanding the basic concepts related to our project and has ignited further interest in this topic.

“Linux for Embedded and Real time Applications”, by Doug Abbott has been of great help in providing an introduction to the process of building embedded systems in Linux. It has helped us understand the process of configuring and building the Linux kernel and installing toolchains.

We understood the preponderance of the ARM processors in the field of embedded systems and the features of ARM processors from the document “The ARM Architecture” by Leonid Ryzhyk. The ARM architecture is a confluence of many useful features that makes it better than other peer processors. Being small in size and requiring less power, they prove useful in providing an efficient performance in embedded applications.

2. PROPOSED ARCHITECTURE

The aim of this project is to get a fully functional single board computer (SBC) working with custom built monitoring software that communicates with all modern devices. It should be capable of extracting the necessary data from the control unit in order to use it in a meaningful and useful way. Communication to and from the CU will be done using the Onboard Diagnostics.

The objectives of this project are as follows:

To get the QT for Embedded Linux C++ framework successfully cross compiled for the ARM architecture so that it will run on the single board computer.

To configure the embedded Linux distribution, Debian Etch that comes with the SBC in such a way that X11 GUI service will be removed and the bare minimum services started.

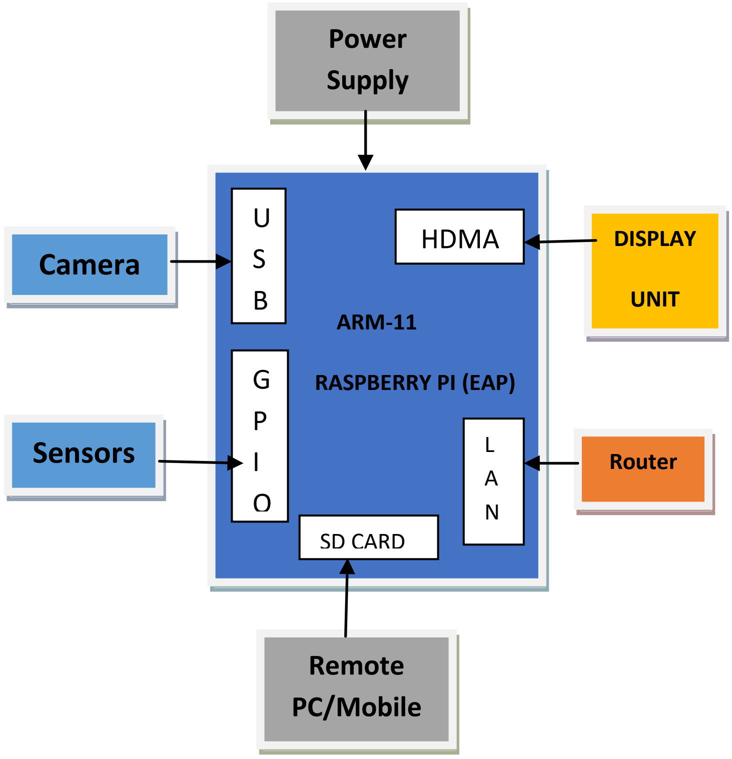
The following are the minimum requirements for this project:

- Software requirements
  - Linux OS
  - QT creator(4.8)
- Hardware requirements
  - Raspberry pi
  - Cables
  - PCF sensor
  - Accelerometer
  - USB Camera

The proposed method represents a model that combines the capabilities of smart IoT devices with control system gateways using real-time challenge response for secure control operations. The proposed solution using both endpoint and gateway devices.

End-to-End (E2E) IoT solutions comprise a combination of two basic platforms. One is the Edge Application Platform (EAP) consisting of sensors and actuators which are connected to an IoT gateway. The other is a Cloud RealTime Identity Monitor Service Platform (CSP) comprising hosted services for connectivity, manageability, and security. The EAP and CSP are designed to be futurecompatible with three primary objectives. First, they must be functional over large distances. Second, they must be able to be managed remotely and securely.

3. BLOCK DIAGRAM



A. ARM ARCHITECTURE: AN OVERVIEW

ARM is a 32-bit RISC processor architecture developed by the ARM Corporation. ARM processors possess a unique combination of features that makes ARM the most popular embedded architecture today. First, ARM cores are very simple compared to most other general-purpose processors, which means that they can be manufactured using a comparatively small number of transistors, leaving plenty of space on the chip for application specific macro cells. A typical ARM chip can contain several peripheral controllers, a digital signal processor, and some amount of on-chip memory, along with an ARM core. Second, both ARM ISA and pipeline design are aimed at minimising energy consumption — a critical requirement in mobile embedded systems. Third, the ARM architecture is highly modular: the only mandatory component of an ARM processor is the integer pipeline; all other components, including caches, MMU, floating point and other co-processors are optional, which gives a lot of flexibility in building application-specific ARM-based processors. Finally, while being small and low-power, ARM processors provide high performance for embedded applications.

A. RASPBERRY PI BOARD

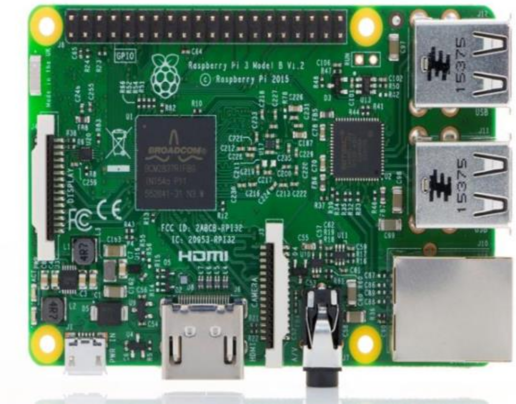


Fig.2.Raspberry pi MODB-1GB

The Raspberry Pi is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools.The Raspberry Pi is manufactured in two board configurations through licensed manufacturing deals with Newark element14 (Premier Farnell), RS Components and Egoman. These companies sell the Raspberry Pi online. Egoman produces a version for distribution solely in China and Taiwan, which can be distinguished from other Pis by their red coloring and lack of FCC/CE marks. The hardware is the same across all manufacturers.

This powerful credit-card sized single board computer can be used for many applications and supersedes the original Raspberry Pi Model B+ and Raspberry Pi 2 Model B. Whilst maintaining the popular board format the Raspberry Pi 3 Model B brings you a more powerful processor, 10x faster than the first generation Raspberry Pi. It has improved power management to support more powerful external USB devices and further to customer feedback based development the new Raspberry Pi 3 now comes with built-in wireless LAN and Bluetooth connectivity making it the ideal solution for powerful connected designs.

The Raspberry pi has a Broadcom BCM2387 chipset 1.2GHz Quad-Core ARM Cortex-A53 802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE) Core IV GPU, and was originally shipped with 1 Gb of RAM. It does not include a built-in hard disk or solid-state drive, but uses an SD card for booting and persistent storage.

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux),C, Java and Perl. Technical specifications of the board are:

- Broadcom BCM2837 64bit ARMv7 Quad Core Processor powered Single Board Computer running at
- 1.2GHz
- 1GB RAM
- BCM43143 WiFi on board
- Bluetooth Low Energy (BLE) on board
- 40pin extended GPIO
- 4 x USB 2 ports
- 4 pole Stereo output and Composite video port
- Full size HDMI
- CSI camera port for connecting the Raspberry Pi camera
- DSI display port for connecting the Raspberry Pi touch screen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source (now supports up to 2.4 Amps)
- Expected to have the same form factor has the Pi 2 Model B, however the LEDs will change position

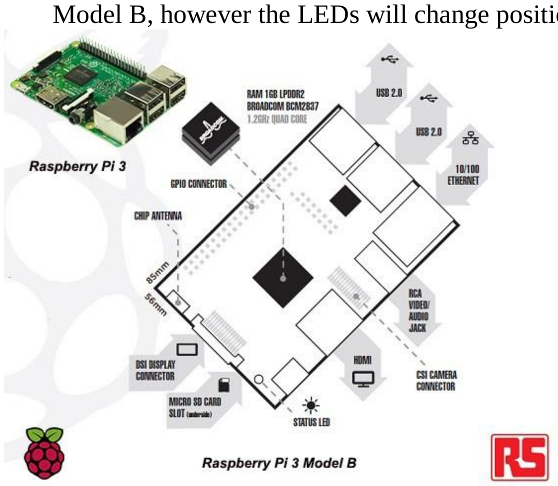


Figure.3 Raspberry pi Layout

B. UVC- DRIVER

A UVC (or Universal Video Class) driver is a USB-category driver. A driver enables a device, such as your webcam, to communicate with your computer's operating system. And USB (or Universal Serial Bus) is a common type of connection that allows for high-speed data transfer. Devices that are equipped with a UVC driver, such as the Logitech® QuickCam® Pro 9000 for Business, are capable of streaming video.In other words, with a UVC driver, you can simply plug your webcam into your computer and it'll be ready to use.

C. UNIVERSAL SERIAL BUS

Universal Serial Bus (USB) is an industry standard developed in the mid-1990s that defines the cables, connectors and communications protocols used in a bus for connection, communication, and power supply between computers and electronic devices.

USB was designed to standardize the connection of computer peripherals (including keyboards, pointing devices, digital cameras, printers, portable media players, disk drives and network adapters) to personal computers, both to communicate and to supply electric power. It has become commonplace on other devices, such as smartphones, PDAs and video game consoles. USB has effectively replaced a variety of earlier interfaces, such as serial and parallel ports, as well as separate power chargers for portable devices.

D. POWER SUPPLY

The device is powered by a 5V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. We have found that purchasing a 1.2A (1200mA) power supply from a reputable retailer will provide you with ample power to run your Raspberry Pi. Typically, the model B uses between 700-1000mA depending on what peripherals are connected; the model A can use as little as 500mA with no peripherals attached. The maximum power the Raspberry Pi can use is 1 Amp. If you need to connect a USB device that will take the power requirements above 1 Amp, then you must connect it to an externally-powered USB hub.

The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000m.

E. AUDIO CONFIGURATION

The Raspberry Pi has two audio output modes: HDMI and headphone jack. You can switch between these modes at any time.

If your HDMI monitor or TV has built-in speakers, the audio can be played over the HDMI cable, but you can switch it to a set of headphones or other speakers plugged into the headphone jack. If your display claims to have speakers, sound is output via HDMI by default; if not, it is output via the headphone jack. This may not be the desired output setup, or the auto-detection is inaccurate, in which case you can manually switch the output.

4. IMPLEMENTATION AND DEPLOYMENT

In this section, we are giving the complete description on the proposed system architecture. Here we are using Raspberry Pi board as our platform. It has an ARM-11 SOC with integrated peripherals like USB, Ethernet and serial etc. On this board we are installing Linux operating system with necessary drivers for all peripheral devices and user level software stack which includes a light weight GUI based on XServer, V4L2 API for interacting with video devices like cameras, TCP/IP stack to communicate with network devices and some standard system libraries for system level general IO operations. The Raspberry Pi board equipped with the above software stack is connected to the outside network and a camera is connected to the Raspberry Pi through USB bus. On the other side we have to host a web server with cloud facility.

The system uses webcam which is connected to Raspberry PI board through USB device. Install CMOS camera device driver in Raspberry pi board.

Now the camera will capture the images. The board will take the images from camera and display it on monitor.

After connecting all the devices then power up the device. When the device starts booting from flash, it first load the linux to the device and initialize all the drivers and the core kernel. After initialization of the kernel it first check whether all the devices are working properly or not. After that it loads the file system and start the startup scripts for running necessary processes and daemons. Finally it starts the main application.

When our application starts running it first check all the devices and resources which it needs are available or not. After that it checks the connection with the devices and gives control to the user.

A. C++ & QT FOR EMBEDDED LINUX

QT for Embedded Linux was used for this project. QT for Embedded Linux is very similar to the regular QT version. The differences are its performance, small foot print design and the support for its own windowing system. This means that it can write directly to the Linux frame buffer device removing the need for any X11 window manager.

The QT framework was originally written for C++ but now has several bindings to other languages. QT for Embedded Linux however h as to use C++ but this suits due to C++'s efficiency. Trolltech, the makers of QT do not release binary versions of their QT Embedded solution. This makes sense as it is impossible to have cross compiled binaries ready for every platform.

B. DEPLOYMENT ENVIRONMENT

The development environment was important to set up properly. first of all i had to install linux on my development machine since it is not possible to build qt embedded in any other operating system. i installed ubuntu since this flavour of ubuntu is very quick with more resources.

5. CONCLUSION

Automatic Classification Of Moving Object in Wireless Multimedia Sensor Networks Using IoT has been successfully designed and tested. It has been developed by integrating features of all the hardware components and software used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, usin highly advanced Raspberry Pi board and with th help of growing technology the project has been successfully implemented.

REFERENCES

[1]. A. Sharif, V. Potdar, E. Chang, "Wireless multimedia sensor network technology: A survey", Proc. 7th IEEE Int. Conf. Ind. Inf. (INDIN), pp. 606-613, Jun. 2009.

[2]. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: A survey", Comput. Netw., vol. 38, no. 4, pp. 393-422, Mar. 2002.

[3]. S. Hengstler, D. Prashanth, S. Fong, H. Aghajan, "MeshEye: A hybrid-resolution smart camera mote for applications in distributed intelligent surveillance", Proc. 6th Int. Symp. Inf. Process. Sensor Netw., pp. 360-369, Apr. 2007.

[4]. A. Kandhalu, A. Rowe, R. Rajkumar, "DSPcam: A camera sensor system for surveillance networks", Proc. 3rd ACM/IEEE Int. Conf. Distrib. Smart Cameras, pp. 1-7, Aug. 2009.

[5]. R. Pon et al., "Networked info mechanical systems: A mobile embedded networked sensor platform", Proc. 4th Int. Symp. Inf. Process. Sensor Netw., pp. 376-381, Apr. 2005.

[6]. D. Lymberopoulos, A. Savvides, "XYZ: A motion-enabled power aware sensor node platform for distributed sensor

network applications", Proc. 45th Int. Symp. Inf. Process. Sensor Netw. (IPSN), pp. 449-454, Apr. 2005.

[7]. V. Jeličić, M. Magno, D. Brunelli, V. Bilas, L. Benini, "An energy efficient multimodal wireless video sensor network with eZ430-RF2500 modules", Proc. 5th Int. Conf. Pervasive Comput. Appl., pp. 161-166, Dec. 2010.

[8]. T. Fan, L. Xu, X. Zhang, H. Wang, "Research and design of a node of wireless multimedia sensor network", Proc. 5th Int. Conf. Wireless Commun. Netw. Mobile Comput., pp. 1-5, Sep. 2009.

[9]. W. Chen, P. Y. Chen, W. S. Lee, C. F. Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks", Proc. Veh. Technol. Conf., pp. 218-222, May 2008.

[10]. Y. A. U. Rehman, M. Tariq, T. Sato, "A novel energy efficient object detection and image transmission approach for wireless multimedia sensor networks", IEEE Sensors J., vol. 16, no. 15, pp. 5942-5949, Aug. 2016.

[11]. Z. Sun et al., "BorderSense: Border patrol through advanced wireless sensor networks", Ad Hoc Netw., vol. 9, no. 3, pp. 468-477, May 2011.

[12]. H. Oztarak, K. Akkaya, A. Yazici, "Lightweight object localization with a single camera in wireless multimedia sensor networks", Proc. Global Telecommun. Conf., pp. 1-6, Nov. 2009.

[13]. S. Boragno, B. Boghossian, D. Makris, S. Velastin, "Object classification for real-time video-surveillance applications", Proc. 5th Int. Conf. Vis. Inf. Eng., pp. 192-197, Jul. 2008.

[14]. G. Bagus, B. Nugraha, S. Weng, H. Morita, "Multiple object tracking on static surveillance video using field-based prediction information in MPEG-2 video", Proc. 17th IEEE Int. Conf. Image Process., pp. 4625-4628, Sep. 2010.

[15]. M. T. Razali, B. J. Adznan, "Detection and classification of moving object for smart vision sensor", Proc. 2nd Int. Conf. Inf. Commun. Technol., vol. 1, pp. 733-737, Apr. 2006.

[16]. S. Zhang, C. Wang, S. C. Chan, X. Wei, C. H. Ho, "New object detection tracking and recognition approaches for video surveillance over camera network", IEEE Sensors J., vol. 15, no. 5, pp. 2679-2691, May 2015.