

# **Network Security Camera System on an M2M Communication Platform**

Aparna.M1, Jesmy John2

*M.Tech Scholars, Embedded System, Sahrdaya College of Engineering &Technology, Thrissur*

---

**Abstract:** A network security camera system is proposed which can be applied to consumer electronic devices to provide visual information for an M2M communications platform. For this system, a network camera processor (NCP) chip and a network security board is considered. The network camera processor consists of a digital camera processor (DCP), a motion JPEG (MJPEG) encoder, an Ethernet controller, an ARM processor and many peripherals. These modules improve the raw images from a CCD image sensor to high quality images, compress the image data size, and transmit them to an M2M communications platform via Internet with no additional hardware. The network security board consists of the board driving the NCP chip and the CCD image sensor board. Its users can observe the scene of a CCD image sensor with no additional hardware on every device, such as a personal computer, a tablet or a mobile phone, transceiving data via Internet. An application framework known as VERTAF, that target at the automatic design of real-time embedded software for ARM-DSP based systems, can be used for automatic code generation of the proposed Network security camera system.

---

## **I. Introduction**

Recent developments in the fields of wireless technology have shown a strong prospective and affinity on improving human life by means of ubiquitous communications devices that enable smart and distributed services. In fact, traditional human to human (H2H) communications are gradually falling behind the scale of necessity. Therefore, machine to machine (M2M) communications have beaten H2H, thus drawing significant interest from industry and the research community in recent times. Advocated by Internet of Things (IoT) [1], a new kind of “networks” will become prevalent in future that connect not only networked terminals like mobile phones, computers, smart devices, but also daily life objects that until now have been to us just “un-networked things” or “inert objects”. Along with the increasing number of ubiquitous communication devices, the traditional human to human (H2H) communications with the inherent need of human operations are gradually falling behind the scale of necessity, which makes machine to machine (M2M) communications [3] obtain much thrust in the industry and research community recently. To approach the final goal of building an advanced society, M2M communications system over heterogeneous networks needs to be designed for contributing different services anytime and anywhere while nourishing the quality requirements of endusers’ experience in various smart services.

The second section presents a network security camera system [4] which can be applied to consumer electronic devices to provide visual information for an M2M communications platform. For this system, a network camera processor (NCP) chip and a network security board is developed. These modules improve the raw images from a CCD image sensor to high quality images, compress the image data size, and transmit them to an M2M communications platform via Internet with no additional hardware. The third section presents an application framework that can be used for real time embedded software development of the proposed network security camera system.

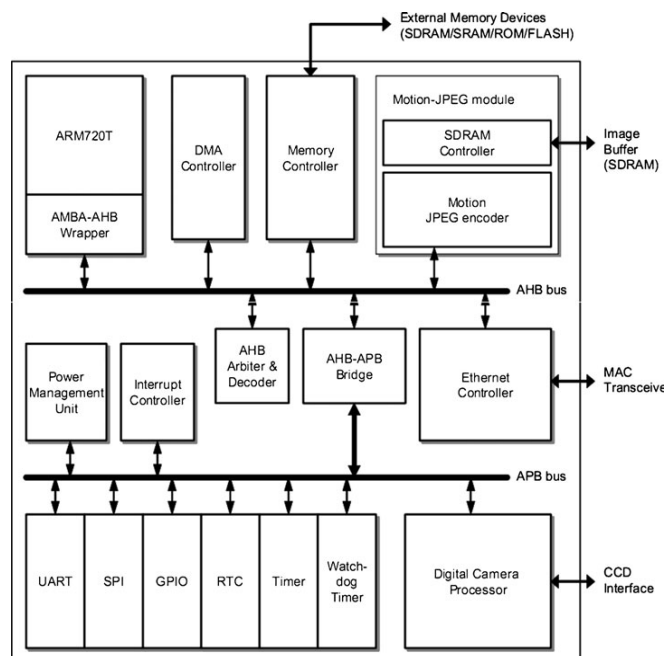
## **II. Network Security Camera System**

### **A. Network camera processor**

The network security camera system consists of two essential sub-boards: the network security main board driving the Network Camera Processor NCP chip and the CCD image sensor board. In this section, the details of NCP chip is summarised to understand the proposed network security camera system. The NCP includes an Ethernet controller and various image processing modules to enhance the quality of images from a

CCD image sensor. Therefore, people can easily observe the scene from a CCD image sensor via Internet, and the visualized information can be used usefully in a security M2M communications platform. Fig. 1 shows the block diagram of the NCP. The NCP consists of many modules: the DCP, which corrects and enhances the quality of the raw images from a CCD image sensor; the MJPEG encoder, which compresses the data; the Ethernet controller, which transmits the compressed data; the ARM processor; and various peripherals.

1) Digital camera processor: The DCP enhances raw images from a CCD sensor and compensates for errors in them. It is improved to obtain high quality image data from a CCD sensor. The DCP comprises of many modules such as a color matrix, an RGB controller, auto white balance, an edge enhancer, an edge compensator and so on.



**Figure 1**

2) Motion-JPEG encoder: To reduce the amount of data that is transmitted via Internet, the MJPEG encoder IP is embedded. The MJPEG format is fully attuned with the baseline JPEG standard. The MJPEG encoder has many functions, such as 4:2:2 & 4:2:0 video formats and 5-20X image compression. The MJPEG encoder compresses the images from the DCP module or external video channels. It also stores the compressed image in the external memory through the FIFO.

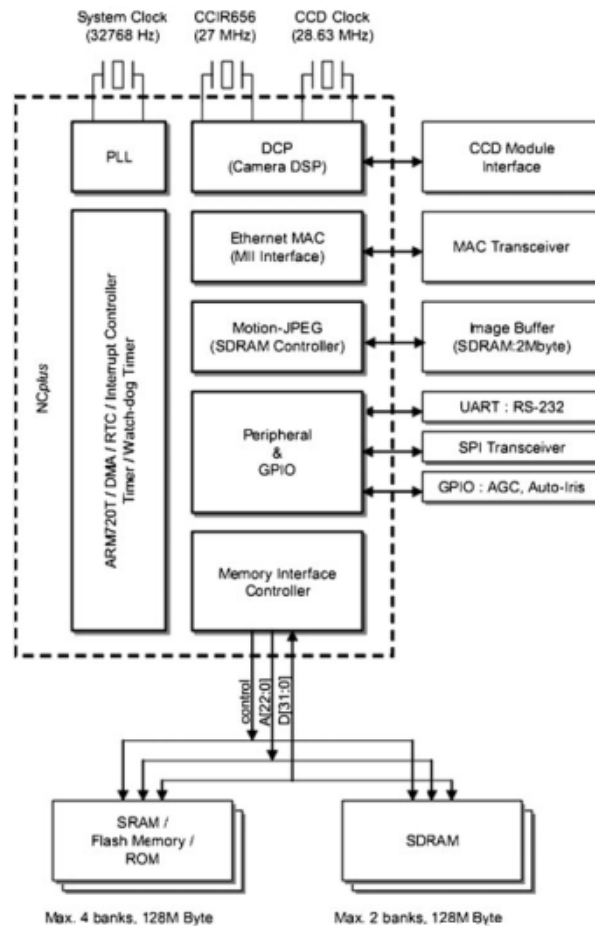
3) Ethernet controller: Ethernet MAC IP is embedded to the NCP chip. This MAC provides an IEEE802.3 compliant media access controller for 10Mbps Ethernet operation. The MAC transmits or receives the IEEE802.3-compliant Ethernet frames. It operates at full duplex or half-duplex. The MAC implements the carrier sense multiple access /collision detection (CSMA/CD) at half-duplex. It also supports the media independent interface (MII), and this allows easy access to the MII management registers.

4) ARM processor: The ARM720T processor is embedded. This low-power RISC processor contains an ARM7TDMI CPU core, an 8 KB unified cache, and a memory management unit (MMU). It also supports 16-bit instruction and also has the cost-effective characteristic of low-power consumption.

## **B. Network Security Main board**

The network security board is the final system to be applied to consumer electronics devices supporting an M2M communications platform. The board is consisting of two sub-boards: the network security main board driving the NCP chip and the CCD image sensor board. Therefore, its users can observe the scene of a CCD image sensor with no additional hardware on many of the devices, including a personal computer, a tablet and a mobile phone, transceiving data via Internet. Fig. 2 shows the block diagram of the network security

main board, and NCplus is the commercial name of the NCP chip. The main board consists of many modules, including the interface with the CCD image sensor board and the Ethernet ports; storage devices; power modules; and so on.



**Figure 2**

To determine the applicability of the network security system to consumer electronics devices supporting M2M communications platform, the proposed system can be applied to a vacuum cleaner. Vacuum cleaner is chosen because it is one of the most user-friendly consumer electronics devices. It is also a moving device, which meant that after applying the network security system to it, we could observe anywhere by moving the vacuum cleaner. To develop the network security vacuum cleaner, an automatic controller should be used that could move a vacuum cleaner automatically. It must be designed for providing two control modes: the normal mode and the security mode. In the normal mode, the vacuum cleaner moves by itself to the empty place, calculated with the information from two CCD image sensors, to clean the room. On the other hand, in the security mode, it travels to where movement is detected to observe and transmit the scene. Therefore, the vacuum cleaner can observe anywhere at home and transmit the visual information to an M2M communications platform via Internet. In a security M2M communications platform the visual information would be utilized very usefully.

### **III. VERTAF**

VERTAF(Verifiable Embedded Real-Time Application Framework)[5]is an application framework that target at the automatic design of real-time embedded software for mobile and ubiquitous systems. It incorporates three software engineering techniques, namely, software component-based reuse, formal synthesis and formal verification. The VERTAF has a component-based architecture. Reusable hardware and software design components can be added to the architecture and so it is easily extensible. VERTAF is highlighted with reduced relative design effort, high-level reuse of software components and highly increased design productivity. It is realized in the form of an integrated design environment focused at the acceleration of real-time embedded software construction. In order to integrate reuse, synthesis, and verification, and for having greater control on how the final generated application will be structured, VERTAF is implemented as an object-oriented application framework. It is also a “semi-complete” application, and the users have to fill in application specific objects and functionalities. A major feature is “inversion of control”, whereby the framework decides on the control flow of the generated application, rather than the designer.

Software Component Reuse and Integration is comprehended in VERTAF. For this, a subset of the Unified Modeling Language (UML)is used with restrictions for automatic design and analysis.Precise syntax and formal semantics are associated with each kind of UML diagram. Guidelines are provided and due to this requirement specifications are more error-free and synthesizable. A specific control flow is embedded within the framework, where scheduling is first performed and then verification because the complexity of verification can be greatly reduced after scheduling. For scheduling, variants of Petri Netsare used, which is automatically generated from user specified UML models that follow the given restrictions and guidelines. For synthesis, we employ quasi-static and quasi-dynamic scheduling methods that generate program schedules for a single processor. For verification, we employ symbolic model checkingthat generates a counterexample in the original user-specified UML models whenever verification fails for a system under design. The whole design process is automated through the automatic generation of respective input models, invocation of appropriate scheduling and verification kernels, and generating reports or useful diagnostics. For handling complexity, abstraction is inevitable, thus we apply model based, architecture-based, and function-based abstractions during verification.

In summary, VERTAF illustrates how an application framework may integrate all the above proposed design and verification solutions. Our implementation has resulted in a Verifiable Embedded Real-Time Application Framework whose features include formal modeling of real-time embedded systems through well-defined UML semantics, formal synthesis that guarantees satisfaction of temporal and spatial constraints, formal verification that checks if a system satisfies all properties and code generation that produces efficient portable code.

Software synthesis in VERTAF can be classified into a two-phase process which consists of a machine independent software construction phase and also a machine-dependent software implementation phase. This separation permits to plug-in different target languages, middleware, RTOS, and hardware device configurations. The two phases are also denoted to as front-end and back-end phases. The front-end phase is again divided into three sub-phases, namely UML modeling phase, real-time embedded software scheduling phase, and formal verification phase. There are two sub-phases in the backend phase, namely component mapping phase and code generation phase.

#### **A.UML modeling phase**

UMLis one of the most popular modeling and design languages in the industry. It standardizes the symbols and diagrams used to build a system model. Users of VERTAF haveto input three UML diagrams as system specification models, namely class diagram, sequence diagram, and statechart. These diagrams are chosen such that information redundancy in user specifications is minimized and at the same time adequate articulacy in user specifications is conserved. UML is a generic language and its specialisms are always required for targeting at any specific application domain. In VERTAF, the three UML diagrams are both restricted as well as enhanced along with guidelines for designers to follow in specifying synthesizable and verifiable system models. The three UML diagrams are extended for real-time embedded software specification as follows.

- **Class Diagrams with Deployment:** A deployment relation is used for specifying a hardware object on which a software object is deployed. Two types of methods, namely event-triggered and time-triggered are used for modeling real-time behaviour.
- **Timed Statechart:** UML statecharts are extended with real-time clocks that can be reset and values checked as state transition triggers.
- **Extended Sequence Diagrams:** UML sequence diagrams are extended with control structures which aid in formalizing their semantics and in mapping them to Petri net models for scheduling.

The set of UML diagrams input by a user, including a class diagram with deployments, a timed statechart corresponding to each class, and a set of extended sequence diagrams, constitutes the requirements for the real-time embedded software to be designed and verified by VERTAF.

### **B. Real-Time Embedded Software Scheduling**

There are two problems in real-time embedded software scheduling, namely how memory constraints are satisfied and how temporal specifications such as deadlines are satisfied. If the system under design does not have an RTOS, a scheduling algorithm known as Quasi-dynamic scheduling (QDS) [10] is applied to solve the above two issues. This requires Real-Time Petri Nets (RTPN) as system specification models. QDS prepares the system to be generated as a single real-time executive kernel with a scheduler. To apply this scheduling algorithm, we need to map the user-specified UML models into Real time Petri nets (RTPN). RTPN improves the standard Petri net with code execution characteristics associated with transitions. A message in a sequence diagram is mapped to a set of Petri net nodes, including an incoming arc, a transition, an outgoing arc, and a place. If it is an initial message, no incoming arc is generated. Different sequence diagrams are translated to different Petri-nets. If a Petri net has an ending transition which is the same as the initial transition of another Petri net, they are concatenated by merging the common transition. By applying the above mapping procedure, all user specified sequence diagrams are translated and combined into a compact set of Real time Petri nets. This set of RTPN is then input to QDS for scheduling.

### **C. Component Mapping**

This is the first phase in the back-end design of VERTAF and is more dependent on hardware. All hardware classes specified in the deployments of the class diagram are those supported by VERTAF and thus belong to some existing class libraries. The component mapping phase then becomes simply the configuration of the hardware system and operating system through the automatic generation of configuration files, make files, header files, and dependency files. The corresponding hardware class API will be linked in during compilation.

### **D. Code Generation**

A multitier approach is adopted for code generation which comprises of an operating system layer, a middleware layer, and an application layer. The ARM-DSP based systems are the supported underlying hardware platforms. This makes VERTAF the ideal framework for Network security camera system. For operating systems, VERTAF supports MontaVista Linux, MicroC/OS, Embedded Linux etc. For middleware, VERTAF is currently based on the Quantum Platform (QP), which is an infrastructure for developing event-driven, real-time embedded applications. QP is C/C++ based and is capable of model-driven development that is directly inherited from UML models. QP has a very small size of only 4 KB of code and data, which is the main reason why it is quite popular in embedded systems.

Utilizing the above mentioned features of VERTAF, the code for Network security camera system can be automatically generated. Inputs to the VERTAF must be UML models as specified in section A. When the device on which the network camera system is installed is operating in its security mode, the controller class notifies the DCP class to use the CCD Camera to capture image of the surroundings and then send the images to a master (the owner of the building or house). The control and dataflows of the media center are automatically generated by VERTAF and the user has to merely specify these sequence diagrams and deploy the related classes to hardware or software components in the class diagram. Hence, VERTAF can save a lot of coding and design efforts.

#### **IV. Conclusion**

The increase of devices with M2M communication capabilities is bringing closer the idea of an Internet of Things. The growth of the next generation consumer electronic devices will depend on the creativeness of the users in designing new applications. A Network security camera system which can contribute towards implementing an M2M communication platform is presented in this paper. An object oriented application framework that can be used for automatic design of real-time embedded software for the proposed system is also introduced. The consolidation of international initiatives is quite clearly accelerating progress towards IoT, providing an overarching view for the integration and functional elements that can deliver an operational IoT.

#### **References**

- [1]. L. Atzori, A. Iera, G. Morabito, "The internet of things: a survey", *Computer Networks* 54 (2010) 2787-2805
- [2]. K. Matthias, H. Paul, A. Schmidt, "Embedded Interaction-Interacting with the internet of things", *IEEE Internet Computings* (2010)
- [3]. M. Chen, "Towards smart city: M2M communications with software agent intelligence", *Multimed. Tools Appl.* Springer(2012)
- [4]. P. Jongsu, H. Seung, L. Y. Surk, "Network security camera system and its application for consumer electronics in ubiquitous environment", *Springer science+ Business Media*, April 2013.
- [5]. P. A. Hsiung, S. W. Lin, C. S. Lin, "Real time embedded software design for mobile and ubiquitous systems", *Springer science+ Business Media* (2008)
- [6]. N. Nakano, R. Nishimura, H. Sai, A. Nishizawa, H. Komatsu, "Digital still camera system for megapixel CCD" *IEEE Trans Consum Electron* 44(3):581-586
- [7]. H. Kim, J. Kim, S. Hwang, I. Park, C. Kyung, "Digital signal processor with efficient RGB interpolation and histogram accumulation", *IEEE Trans Consum Electron* 44(4):1389-1395
- [8]. H. Kami, T. Kurashige, M. Higashi, T. Imaide, T. Kinugasa, "A network MPEG camera", *IEEE Trans .Consum Electron* 45(4):1206-1212
- [9]. G. Wu, S. Talwar, K. Johnsson, N. Himayat, K. D. Johnson, Intel, "M2M: From Mobile to Embedded Internet", *IEEE Communications*, April 2011.
- [10]. P. A. Hsiung, T. Y. Lee, C. S. Lin, "Quasi dynamic scheduling for the synthesis of real-time embedded software with local and global deadlines", *RTCSA'2003* (February).