RESEARCH OF COMMUNICATION SCHEDULING BETWEEN COMPUTER I/O AND S/W FOR ACQUISITION OF SATELLITE SENSORED DATA

Cheol-hea Koo, Su-hyun Park, Soo-yeon Kang, Koon-ho Yang, Sung-bong Choi

COMS Department, Korea Aerospace Research Institute, chkoo@kari.re.kr

ABSTRACT: Various communication mechanisms have been developed to acquire a meaningful data from sensors. The key requirement during the sensor data acquisition is determinism and reduction of time dependency. It is a fundermental level of satellite data management for controlling satellite operation software data acquisition from sensors or subsystem. Satellite operation software has various software modules to be operated in addition to data acquisition. Therefore, unnecessary time delay shall be minimized to perform the data acquisition. As the result, interrupt method might be prefered than polling method because the former can decrease the restriction of design during implementation of data acquisition logic. The possible problems while using interrupt method like as interrupt latency caused by delaying of interrupt processing time are analyzed. In this paper, communication mechanism which can be used to interface with satellite computer and subsidary subsystem by using interrupt is presented. As well, time dependency between software scheduling and data acquisition is analyzed.

KEY WORDS: Data Aquisition, Satellite, Sensor, Computer I/O

1. INTRODUCTION

Communication mechanism between computer and sensor has been developed to achieve meaningful data from sensors. Various methods are available to clarify the enhanced Determinism and minimize Time Dependency. It is the most basic level of satellite data management for satellite operation software that acquires data from sensors or subsystem on satellite.

Non beneficial time delay from acquiring data should be minimized because satellite operating software has many modules to be executed. So, interrupt method is more effective than polling method to reduce the limit of design when data acquisition logic is implemented.

It is possible to use polling method as effective method for data acquisition on satellite system from time to time. But it has complex time limitation comparatively. On the other hands, interrupt method has lower time limitation than polling method, but it should be analyzed with detail because it could be under problematic situation according to interrupt latency coming from delay of interrupt processing time [1].

This paper presents communication mechanism between satellite computer and subsystem under interrupt method and analysis of time dependency between software scheduling and data acquisition. The COMS contract to develop the COMS satellite and to provide support for system activities has been awarded by KARI to ASTRIUM France. The COMS joint project group is composed of KARI and ASTRIUM engineers.

2. ABSTRACTION OF DATA ACQUISITION

There is clear boundary of interface between hardware and software, but by abstraction, can be seen in Figure 1, satellite operation software can handle hardware as a software module. Kernel and its utility is the exact boundary between hardware and software.

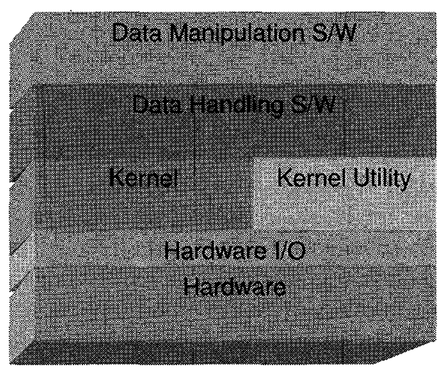

Figure 1. Structure of Architecture

The advantage of abstraction is that interface between hardware and software can be handled as interface between software modules. More important thing is that in real implementation, many assumptions necessary to interface with hardware can be skipped. Beacause, these assumptions are being processed on the abstraction level of interface already.

Generally, kernel can provide the tools to utilize hardware. But developer should take account of hardware operating software, so called driver. And this software category is the area of developer's charge.

3. BASIC MECHANISM OF DATA ACQUISITION

The purpose of abstraction of data acquisition is to divide the mechanism of generation, movement and store of the target data into item levels. After achieving this, each software item should be called exclusively by upper software module like as "Data Manipulation S/W" in Figure 1.

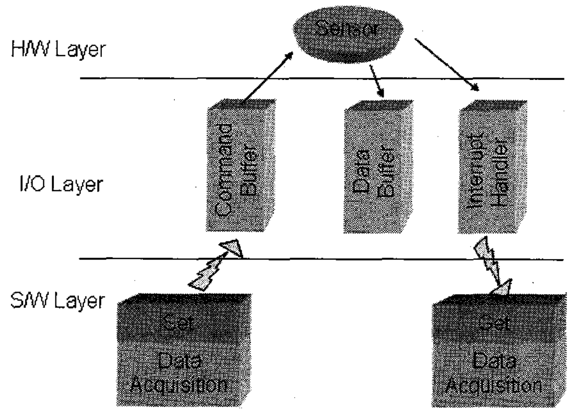

Figure 2. Basic mechanism of data acquisition

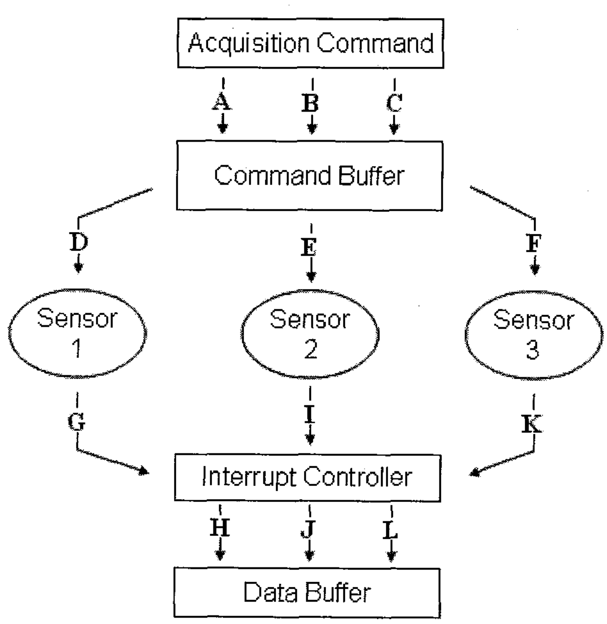

Figure 3. Diagram of data acquisition from sensors

Normal data acquisition mechanism is similar to Figure 2. The effect of latency between data acquisition ignition and data acquisition completion should be analyzed to guarantee the determinism of real time system like as satellite system and to minimize the unnecessary dead time happened when computer hardware I/O is received the command of data transmission and sensor transmit data and it's acquiring.

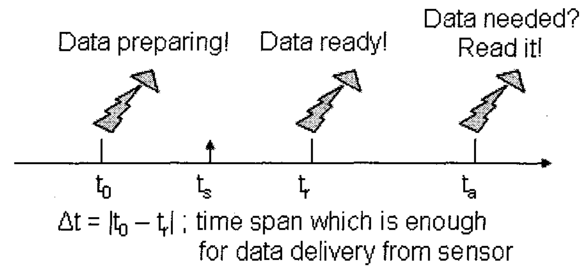

Figure 4. Timing diagram of data acquisition

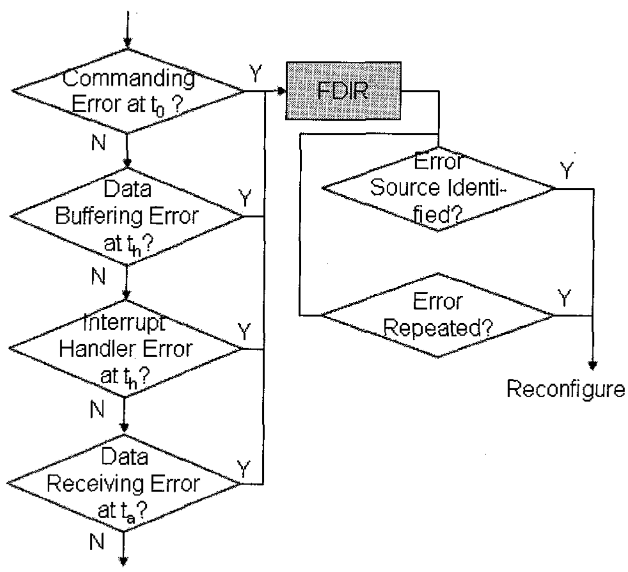

Figure 5. Diagram of FDIR

In Figure 3, data acquisition command interface and processing timing diagram from general sensor devices. The processing is executed as the order of 'A' to 'L' and it is an assumption. Because the time span between data acquisition ignition and data acquisition completion is variable according to the circumstances of sensor operation and computer hardware I/O, data acquisition task should not depend on the timing behavior of acquisition mechanism.

To solve this latency problem, it is recommended to allow for time span between data acquisition ignition and data acquisition completion using predetermined latency time as describing Figure 4. This software processing logic could not be sequential because another task has to wait long time for end of the processing if this processing is working sequentially. So, it is very reasonable to allocate the exact analyzed time span for achieving good timing performance during data acquisition.

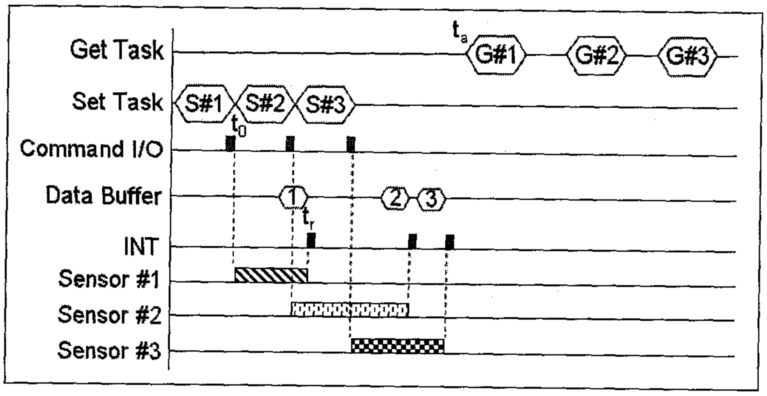

Figure 6. Timing diagram of data acquisition

Time span $\triangle t$ can be extracted from timing diagram of real hardware operation. And $\triangle t$ can be regarded as static(not dynamic) value.

As described in Figure 4, a task acquiring data is possible to read new data through computer I/O after t_r. So, to read the data, preparation of data before $\triangle t$ is performed and waiting for interrupt from computer I/O notifying the completion of data acquisition should be performed. This method is useful to diagnose the anormal state of data acquisition from sensors and intimately associated in FDIR mechanism. Because, if signal is not achieved within pre-specified time limit, there should be anormal situations in remote acquisition side.

4. FDIR

Error handling should be accompanied with the anormal data acquisition coming from sensor hardware problem when data acquisition is performed.

In the architecture described in this paper, FDIR can be considered like as Figure 5. It can be considered as sensor hardware error or related interface error if data is not arrived within predetermined time. Then reconfiguration of sensor system should be performed to keep the designed function.

If redundancy is existed, first of all the function should be recovered on redundancy. After then a diagnosis will be performed on erroneous hardware. If redundancy is not existed, erroneous hardware should be reset and be checked if function is restored after initialization. If malfunction is not cured, this erroneous hardware should be removed on mission configuration hardware to block the error spreading [2].

The software resource and time to take the fresh function when error happens should be checked because the errors during data acquisition give terrible impact on logic which needed to the data.

5. MANAGEMENT OF ACQUIRED DATA

The next step of data acquisition if data is acquired within specified time without problems is data manipulation of data.

It is the most important and last part of data acquisition to distribute the acquired data to the customers of the data. If data can not be delivered within specified time, every effort so far is labour lost [2, 3, 4].

If various data can be distributed to each customer, these data should be buffered because it is not efficient to perform the transaction every time data is arrived. If the number of N is the value to perform the transaction efficiently on a system, data should be stored until N_{th} data is arrived. Therefore a concept which processes the data as block unit level as described in Figure 6 is required and it will considerably contribute to raise the performance of data acquisition logic.

In Figure 6, an example is shown as detailed timing diagram during block unit data acquisition. Data can be read from the buffer(t_a) by data process task after receiving interrupt signal(t_r) which notifies the save finish of sensor data to data buffer form sensor #1.

So, "Get Task" can be invoked after or before $\triangle t$ of "Set Task". "Get Task" is just waiting the signal which notifies the event that sensor save the data to buffer successfully [1].

Therefore, "Get Task" is independent with "Set Task" with regard to time behavior and the focus of function. Actually, "Get Task" can neglect the existence of "Set Task" during the operation. The more the interrelationship between "Get Task" and "Set Task" is decreased, the more the operability and maintainability of data acquisition become simple.

Through this concept, the ambiguity between the time of data ready of sensors and the time of data acquisition is removed. And time dependency between "Set Task" and "Get Task" is lowered than the other mechanism [5, 6].

Actually, to implement this concept, a specific interrupt controller is needed to process the interrupt signals from multi sensors. So, special controller is mandatory for nice processing of multiple data acquisition and distribution and shall support main micro-controller because this kind of activity can be significant burden in the general purpose micro-controller even if it is present state of the art processor.

6. CONCLUSION

The boundary between hardware and software can be removed by abstraction concept. And data acquisition can be programmed to deal with various situations by modularity of function and considerations of time.

But, abstraction can be realized when rigor conditions is met. So, when data acquisition is not performing well due to hardware error, reconfiguration should be performed through FDIR as soon as possible. It means abstraction is fragile and can't be trustable at whole time. But abnormal situation can be cured more easily because the consisted modules are divided into small pieces of module.

Exactitude and determinism is the most valuable item to perform the correct function on real time system like as satellite system. Data acquisition through sensor hardware should be treated same manner.

Abstraction can cover earth/sun/thermal sensor, gyro, and wheel attached on satellite and even data acquisition itself. By using this abstraction, task which performs overall data acquisition can be divided into "Set Task" and "Get Task" with lower inter-dependency.

This paper presents interface between computer I/O and hardware for acquiring data from sensor hardware with guarantee of determinism and minimized time dependency, and FDIR logic.

REFERENCES

References from Journals:

[1]C. H. Koo, 2005. Method of data processing through polling and interrupt driven I/O on device data. *KSAS*, Vol. 33, No. 9, 2005

[2]A. Burns, A. J. Wellings, 1995. *Engineering a Hard Real-time System: From Theory to Practice*, SOFTWARE-PRACTICE AND EXPERIENCE, VOL. 25(7), pp. 705-726, 1995.

References from Books:

[3]Giorgio Buttazzo, *Real-Time Issues in Advanced Robotics Applications*, Pisa, Italy.

[4]Ken Tindell, *Analysis of Hard Real-Time Communications*, University of York, England.

[5]M. Ben-Ari, 1990. *Principles of Concurrent and Distributed Programming*, Prentice Hall International Series in Computer Science, pp 164~175.

[6]Narain Gehani, Andrew D. McGettrick, 1988. *Concurrent Programming*, International Computer Science Series, pp. 312~318.

ACKNOWLEDGEMENTS

This paper is a part of "COMS 1 System and Bus Development Project" which is sponsored by MOST.