SERIAL COMMUNICATION ENCRYPTION IN EMBEDDED SYSTEM

[1]RHITVIK KUMAWAT, [2]GAURAV JAIN

[1,2]Electrical Engineering Department, Global Institute of Technology, Jaipur, India
E-mail: [1]rhitvik17@gmail.com, [2]gaurav.jain@gitjaipur.com

Abstract— With the increase in technology, the embedded systems are being employed in the daily as well as in the industrial level control systems. With the help of these not only the system becomes more cost effective but also the parameters of the system can be monitored for achieving an optimized and peak performance. One of the means to generate an embedded system apart from computer controlled directives is by using microcontrollers. These microcontrollers can be independently used or can be controlled by other microcontrollers. The data acquired by a single microcontroller unit that is measured by a transducer and is transmitted to the other microcontroller for further instructional cycles. So these communications are carried by the microcontrollers and these data are transmitted in a serial fashion after it is read from a buffer in which the data is held in a parallel order in the buffer. This data sent in a serial fashion and this type of communication is aggregately coined as serial communication. In this research paper, few approaches are simulated to encrypt the data sent in serial communication mode in an embedded system to retain the confidentiality of the transmitted data (as if one manages to sample the data with the help of a logic analyzer or any other means, the security of the system can be compromised) and a cascaded communication approach to solve the problem of pin count deficiency and independent parallel monitoring in an open loop control system to achieve a better level of observability and controllability in a digital control system.

Keywords— Embedded System, Control System, Serial Communication.

I. INTRODUCTION

Embedded systems are nowadaysgaining popularity in the modern world as these systems have the ability to digitalize the system with which these systems are integrated. So that the system can be optimized to its peak performance which makes the whole system more efficient and reliable just not on the basis of its performance but also makes the system more energy efficient and user friendly.

With the increased competitions in the field of business and technology, only the products that will be in demand are the ones that will give the most to the users at least cost possible. So in order to achieve this aim, the systems and products manufactured are digitalized and these then emerge out as a member of embedded family. As the systems are integrated with microprocessor technology so that the internal operational parameters can be monitored too and can be driven accordingly in order to render the system optimized.

1.1. Embedded System

An embedded system is formed by integrating the given system (whether the system is electrical, mechanical, or electromechanical) with a microprocessor based digital system, so that external as well as the internal parameters of the systems can be monitored continuously in real time. Hence the resulting system can be controlled to its very foundation by varying the base parameters to achieve the desired operational characteristics.

Nowaday's peripheral integration of microprocessors is obsolete due to their complexity and cost but with the advancement of technology, VLSI integration has made possible to integrate the whole system into a single chip known as microcontroller.The most popular in the field of embedded systems are microcontrollers based and computational developmental platforms like pine, Intel Edison etc.[1]

1.2. Control System

Control system in simple words is a system that has the ability to generate and maintain a desired level of output by comparing the output or its part as a feedback to the preset values and alter the parameters accordingly within itself to get the desired result.

Here open loop control system is used that implies that no feedback system is present in the system to ensure if the function entered is executed properly or not. In a closed loop control system the part of the output is sent back to the processing unit so as the system can compare the output generated by it to the desired output and can take corrective measures in order to obtain the desired results. These closed loop control systems if integrated with knowledge based and learning algorithms can give rise to intelligent systems. Another high end developmental platform that is being the center of attention to some of the best brains in the world.

1.3. Serial Communication

Serial communication is often used either to control or to receive data from an embedded microprocessor or microcontroller. For the purpose of explanationSerial communication formsan input-output in which the bits of a byte begin transferred appears one after another in a timed sequence on a single wire. These bits are then sampled on the other side of the wire and assembled back to the original byte and the data is thus said to be transmitted.

Nowadays serial communication has become the standard platform for digital communication.

There are many types of serial communication that are currently used according to the requirement like USART, SPI, TWI/I2C, RS232etc. The methodology of transmission of data is nearly identical but different platform are generated to suit the requirements of the system in a way that the communication taking place is precise and is reaching all the devices that are required to receive the data with least amount of noise or disturbances. The communication can also be synchronized in within two devices sharing a common clock line. Such a communication is known as synchronized serial communication.[11]

In this paper UART communication is taken to show how we can encrypt these transmitted signals. Also cascaded serial communication is demonstrated for unidirectional communication actuated sub systems.

II. USART METHODOLOGY

USART stands for Universal Synchronous Asynchronous Data Receive and Transmission.

This method can be further divided into two sub parts, i.e. synchronous communication and asynchronous communication. In this communication what actually happens is that one microcontroller sends the data to the second microcontroller device by changing the logic values on the communication channel in a certainly timed fashion. These logic levels are then detected on the receiving side of the line by the second device and are assembled in the receive buffer to obtain the original byte. In case of asynchronous communication mode, both the microcontrollers work on their respective clocks. Hence the communication parameters are preset in both of these microcontrollers that will share a common ground, baud rate, bit length, start stop bit values, parity bit, etc.[2,3,4,5]

Now if case happens that the two devices are in proximity to each other, they can both share a single clock line(generated by any one of the devices) apart from the common ground. This speeds up the communication speed as the communication is now synchronized and there is no requirement for a synchronization byte to be sent in a beginning of a communication, nor there is any requirement of start-stop or parity bits. But here clock polarity can be chosen for the purpose of reference to ease the user to choose whether the data transmission and sampling has to be done at rising clock pulse or falling clock pulse.

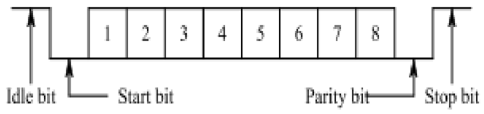

Fig. 1.Figure of USART Communication

2.1. Baud Rate

Baud rate is the rate at which data is transmitted in serial communication. The baud rate generator output is used directly by the Receiver's clock and data recovery units. In order to set the baud rate in microcontroller, UBRR (USART Baud Rate Register) is set with a value chosen according to the requirements of the hardware provided (includes oscillator or speed mode). These values are initialized in both the microcontrollers in same order.

2.2. Data Transfer Methods

The transmission can be done by two main methods. First is state cycle implementation where the microcontroller waits for the USART Data Register (UDR) to be empty and then loads the byte into it. This byte is then transferred in a framed fashion to the second microcontroller.

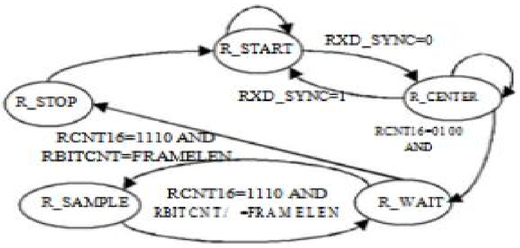

Fig. 2. State Cycle in Serial Communication

Second method is by using interrupts for transmission purposes, i.e. just as soon the UDR register is empty, an interrupt is generated due to which the current execution of the program halts and the values are loaded into UDR register and are transmitted. After the transmission is complete, another interrupt occurs that pulls the received values from the receive buffer. The later process is more efficient because of the fact that the microcontroller is not stalled due to polling which allows it to perform other assigned tasks.[8]

2.3. Need for Encryption

Suppose in an embedded system, data is being transferred in a serial mode. The data that is being transferred is a magnitude of a certain quantity measured by the transducer sensor for monitoring purposes. There can be a situation that if one manages to sample the data with the help some device, the security of the system can be compromised as of now the confidential information about the process that managed to keep the firm competitive in the market is exposed. So in order to retain the confidentiality of the data, the bits transmitted can be encrypted so that the data sampled by any third party will not be understandable until and unless the same approach is applied to decode it.

Following are some approaches to do so:

1-Variable baud rate based communication.

2- By rotating the transferred byte within the microcontroller itself before writing it to UDR.

3-Crypto-graphical data redundancy can be used, i.e. using cryptographic means to mix garbage values along with the information to hide it byte wise so that the sampled data by the third party will not be able to obtain the correct one.

III. ENCRYPTION ALGORITHM

3.1. Cascaded UART Communication

In this method if case happens that in a microcontroller in all the communication ports are occupied like I2C/TWI, SPI and only USART ports are available to communicate with the microcontrollers that need to communicate with each other, and we know that only limited ports are available in a microcontroller, so instead of using a high end configuration microcontroller with multiple UART channels, a closed chain of cascaded microcontroller can be generated in which one transmitter is connected to the receiver of the second microcontroller. Then the transmitter pin of this second microcontroller is connected to the receiver bit of the third microcontroller and so on. And finally the transmitter pin of the last microcontroller is connected to the receiver pin of the first microcontroller.[7]

Now each microcontroller unit can be assigned a unique id so that the microcontroller that is meant to receive data can be accessed on a first byte hand shake basis and the processing resources of the microcontrollers are not overused, as the microcontrollers will limit their communication to the unit the data is intended to be sent.

The communication can be done between any two or more microcontrollers in a cascaded fashion and the best part of this assembly is that the whole communication can be duplex in nature as a result of which closed loop control systems can be generated using this method. Generally the methodology of parallelcommunication is used by the devices for communication by the devices as an accepted convention. This causes two major problems during the communication process

1. There is a possibility that communication channel will become more noisy as the probability of the introduction of noise increases with the increase in the number of devices.
2. Second thing is that in case the algorithms entered in the system are intelligent then only the microcontroller that is assigned the id will be able to receive the data whereas in case of cascaded chain, the data will be received by all the preceding microcontrollers and will be able to learn at the same time or if the algorithms allows so, these microcontrollers will be able to draw conclusions from the data to learn accordingly to the task that these are employed for.

But due to communication lag it is advised to use this method in an open looped control system in order to maintain the response time of the system to a minimum value as well as making the communication more secured and noise free. If needed the cascaded system itself can encrypt the data received by it at each step accordingly so that sequential encryption and decryption will totally hide the information content and if sampled, it will show anonymous garbage data to the sampler put over the communication channel.

3.2. Variable Baud Rate

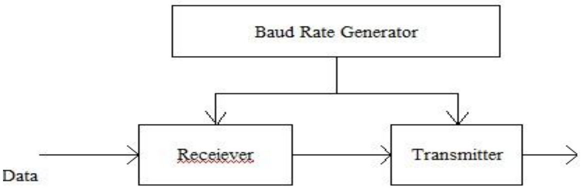

Fig. 3. Baud Rate Generation

This means that after a certain number of bits are sent, UBRR (USART Baud Rate Register) is updated a new value with the help of interrupt. In the simulation generated the baud rate is lowered down by a factor of two after every third byte is sent. Parity flags, data overflow flags and frame error flags in the interrupt register keep a precise observation on the data transmission and reception, i.e. if case happens that an error is generated during the transmission, not only the data will be resent but the counter for checking the number of bytes will halt too. In other words the UBRR (USART baud rate register) will not be updated unless 3 successful bytes or otherwise programmed are transferred.

This is how the baud rate variation can be successfully achieved-
Initial baud rate = 9600
TX will send out three bytes of data. This will be monitored by the transmitter as well as the receiver itself.
After the third byte is sent, the microcontroller will send a preset byte which will act as an inter system feedback.
After the byte is sent and successfully received, respective Interrupt Service Routines will refresh the value of baud rate by altering the
UBRR values given by the formulas
UBRR=Fosc/ (16*baud) -1
For U2X=0 i.e. double speed asynchronous mode is off.
UBRR=Fosc/ (8*baud) -1
For U2X=1 i.e. double speed asynchronous mode.
UBRR=Fosc/(2*baud) -1
For a synchronous transfer mode:
The baud rate changed from 9600 to 4800 to 2400 to 4800 to 9600 in a continuous communication mode.
(Fosc – frequency of oscillator)

This continuously alters the baud rate in a communication system and every fourth byte sent is a garbage byte as it is required by the received buffer to synchronize itself again at a redefined baud rate. This method can be used when continuous data transfer is being done.[9,10]

It is advised that the state cycle implementation is avoided in this form of variable rated communication as it might toll the microprocessor resources in the process and the other functionalities of the other computational systems can be rendered low.

This method is only applicable for the system in which only two devices are communicating only with each other.

3.3. Byte Rotation

In this, every byte that has to be written into UDR is firstly written into anotherregister and the MSB (Most Significant Bit) and LSB (Least Significant Bit) in the new register are closed to form a closed loop.

In this byte, each and every bit is shifted either clockwise or anti clockwise by a certain number like a plane is rotated in a Rubik's cube. For this the code can be generated that will shift the bits accordingly by a given amount in a certain direction. Now the loop is opened again from the same point from where it was joined originally and the obtained byte will be said to be encrypted and will not be understood in the context. This byte is then sent by the transmitter and unless the receiver carries out the samedeciphering algorithm, any third party that is sampling the aforementioned data will obtain a garbage value hence making sure that the confidentiality of the data is maintained.

This is then updated to the UDR register after USART Data Register Empty Interrupt is executed. The data is then transferred serially on the other side. The read data is again closed to form a closed loop and the byte is now rotated in opposite direction with respect to the transmitter side of the data by the same shift as done on the transmission side. This will get the original byte that was to be transferred.[15,16]

This can be used with the state cycle implemented analogy with some alterations in the code if required as the interrupts are critically engaged for some another priority based interrupt service. It is advised to use this method when the operational frequency of the system is high.

3.4. Cryptographicdata Redundancy

In this method, each transmission byte is sent in a packet with some garbage values and the original byte is hidden inside it. Only the microcontroller will know and set the original byte number and the number of bytes to be sent with the original one. Now the second microcontroller will also be programmed to screen the same exact byte received from a multiple number of bytes. For example if the microcontroller is sending 9 garbage bytes along with the original byte, this original byte can be sequentially be placed at different locations but in a predefined generated pattern the same will be registered in the second microcontroller itself. As a result of this the entire data train will be transmitted over the channel but only the byte filtered from the packet will be the predefined one from the algorithm fed inside and the position of the filtered byte will vary with time or as decided by the code written to the microcontroller.[10,16]

Also microcontrollers can be set with unique code generators that can alter the byte position and the number of garbage bytes in order to vary the communication speed according to the requirements. For example communication that will require continuous data transmission at high baud rate can be set with low number of garbage bytes. But this method is only applicable with the non state implemented schemes for data transmission as polling will not be able to filter in case one of the microcontroller units fails to receive the data. But in case of continuous communication using interrupts the counter will itself be programmed in the interrupt service routines; hence the count for data retrieval will be registered accurately by the processing unit.[18]

CONCLUSION

The main motive for serial communication encryption is that the data that is being transferred from one system to another in a control system serially is venerable to the third parties as one can easily access itif onemanages to get the hardware details. The data in the line can be easily sampled with the help of a logic analyzer and then can be easily read. As this might put the security of the system in a venerable position, serial communication if used in an encrypted methodology can prove to be a good alternative for AES to generate a secured channel for communication between the devices.

In the simulations generated the data was successfully transferred from one microcontroller to anothermicrocontroller with a closed loop control feedback and was successfully able to vary the baud rate as programmed accordingly.Also the other means of communication encryption algorithms successfully altered the transmission bytes and the later decoding successfully obtained the respective bytes.

This technique can be implemented in a serial communication where the confidentiality of the data being communicated is necessary as it might be sensitive information and its exposure to others might render operating firm in a zone of venerability as the processes in an industries are kept confidential to remain competitive in the market. Not only will the communication processes become more secured, but also the additional secured channel or hardware can be eliminated.

ACKNOWLEDGEMENT

RhitvikKumawat:B.Tech. student in Electrical Engineering Department of Global Institute Of Technology Jaipur, India. His research interest includes Embedded systems, robotics and automotive engineering.

Gaurav Jain: Presently he is working asAssistant professor in Electrical Engineering Department in Global institute of Technology, Jaipur, India. He Received the M. Tech. Degree from Guru Nanak Dev Engineering College, Ludhiana, India. His research interest includes in Power Transmission, Power Plant Engineering and Embedded systems.

REFERENCES

[1] Muhammad Ali Mazidi, SarmadNiami, SepherNaiami. “The AVR microcontroller and embedded systems” section 11.3 – 11.4, pp.405 – 408, 2011.

[2] Shouqian Yu,Lili Yi,Weihai Chen,Zhaojin Wen, “Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA” International Journal of Innovative Research in Science & Engineering, pp.1-7, 2007.

[3] Hazim Kamal Ansari, AsadSuhailFarooqi, ‘‘Design of High Speed UART For Programming FPGA” International Journal Of Engineering And Computer Science, vol.1, pp.28-36, 2012.

[4] O.A.Petlin, and S.B.Furber, ‘‘Built-In-Self-Testing of Micropipelines” IEEEAdvanced Research in Asynchronous Circuits and Systems, pp 22-29, 1997.

[5] AmanpreetKaur, AmandeepKaur, “An Approach for Designing A Universal Asynchronous Receiver Transmitter (UART)”, IJERA Vol. 2, Issue 3, May-Jun, pp. 2305-2311, 2012.

[6] BhavnaManure and Rahul Tanwar, “UART with automatic baud rate generator and frequency divider” Journal of Information Systems and Communication, vol.3, pp.265-268, 2012.

[7] Fang Yi-yuanChenXue, “Design and Simulation of UART Serial Communication Module Based on VHDL”, IEEE Conference, pp.1-4, 2011.

[8] Huimei Yuan, Junyou Yang,PeipeiPan(2010), ‘‘Optimized Design of UART IP Soft Core based on DMA Mode” IEEEIEEE Conference on Industrial Electronics and Applications, pp.1907-1910, 2010.

[9] Dr. GarimaBandhawarkarWakhle, ItiAggarwal and ShwetaGaba(2012), ‘‘Synthesis and Implementation of UART using VHDL Codes”, IEEE International Symposium on Computer, Consumer and Control, pp.1-3, 2012.

[10] Datasheet of“PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs”, National Semiconductor Application Note.

[11] M.S.Harvey, ‘‘Generic UART Manual’’, Silicon Valley, 1999.

[12] P.Bertin,D.Roncin and J.Vuillemin, ‘‘Programmable active memories: Performance measurements’’, ACM First International Workshop on Field-Programmable Gate Arrays, 1992.

[13] Datasheet of ‘‘Atmega32 microcontroller’’.

[14] Karalis, Edward, “Digital Design Principles and Computer Architecture.” Prentice-Hall, United States of America, 1997.

[15] Mohd Yamani IdnaIdris, MashkuriYaacob, “A Vhdl Implementation Of Uart Design With Bist Capability’’Malaysian Journal of Computer Science, vol.19, pp.73-86, 2006.

[16] S.Brown, R.Francis, J.Rose,Z.Vranesic, “Field-Programmable Gate Arrays”Kluwer Academic Publishers, 1992.

[17] J.Rose, A.ElGamal, A.Sangiovanni-Vincentelli, “Architecture of Field-Programmable Gate Arrays” in Proceedings of the IEEE, vol.13, pp. 1013-1029, 1993.

[18] M. Barr and A. Massa ‘‘Programming Embedded Systems: with C and GNU Development Tools’’, 2nd Edition.

★★★