Multiple Protocol Converter

D.Sivaraj and G.Ragavi

1Asst. professorDept. of ECE, PSG College of Technology, Coimbatore, India
dsr.ece@psgtech.ac.in
2Communication Systems, Dept. of ECE, PSG College of Technology, Coimbatore, India
19ml10@psgtech.ac.in

dsr.ece@psgtech.ac.in

Abstract. In the modern world, IoT plays a vital role. In general, each device works on a different protocol. The problem of achieving communication between two processes across a network or an internetwork is considered. The notion of logical connectivity between processes in a protocol is formalized. The problem of constructing a protocol converter to achieve interoperability between processes that implement different protocols is addressed. A formal model is presented, based on the theory of protocol projection, for reasoning about the semantics of different protocols and conversions between them. This paper ‘Multiple Protocol Converter’ aims in converting from one protocol to any other protocol as per the requirement which supports different devices to interconnect with IoT seamlessly.

1. Introduction

To achieve the interoperability between devices of different protocols, a protocol converter is used. The goal of this proposed method is to enable one device to emulate the communication functions of another device. The main objective is to develop a module through which multiple protocol conversions are possible. To establish communication fabric that enables you to collect data from various devices and different protocols and translate those in a centralized device so that you can collect and compile data from all over a facility. Mainly for Inter – Compatibility, Better communication between devices.

Some applications such as a car has a CAN protocol and mobile which supports a bluetooth protocol can be communicated using a ELM which is used for PC to OBD interface. Here the protocol conversion plays a major role in communicating between devices. Also some other real time application like Anti-lock brake system and Electronic Stability Program also require a protocol conversion which helps in interfacing WiFi and CAN protocol for communication. In today world, conversion of WiFi and USB to transfer files like image, video, audio, etc., is a very basic needs for communication which is done by some protocol converters like Digisol. Many other such applications helps us feel comfortable to use such as Bluetooth headsets, Bluetooth speakers, Audio systems, WiFi hotspot systems, etc., Thus usage of bluetooth, WiFi communications became more necessary. Not all devices supports all protocols, thus protocol conversion is much required for all kind of IoT based applications. Home automation is the best example for such applications.

The proposed methodology of this project has a single common box which converts five different protocols with five different protocols.
Figure 1. Protocol Converter.

The figure 1 shows that any type of a protocol when given to protocol converter can be converted to any other protocols.

2. Communication Protocol

There are multiple communication protocols in which some protocols are been considered in this proposed method.

- Wired protocols
  - UART
  - USB
  - I2C

- Wireless protocols
  - Bluetooth
  - WiFi
3. **Proposed Methodology**

| Simulations                                                                                                                                                                                                 | Protocol Conversion                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| DS1307 RTC with microcontroller                                                                                                                                                                          | I2C To Parallel Data Conversion      |
| LPC 2148 EEPROM with microcontroller                                                                                                                                                                      | UART To I2C                          |
|                                                                enery between two microcontrollers                                                                                                                                                                     | I2C To UART                          |
| Communication between two microcontrollers                                                                                                                                                                | UART To Parallel Data Conversion     |
| Communication between 2 WiFi module                                                                                                                                                                       | USB To UART                          |
| Controlling devices via serial port using a WiFi Module                                                                                                                                                   | UART To USB                          |
| Controlling devices via Web Server using WiFi module                                                                                                                                                     | UART To USB                          |
| Simulating respirating of 2 led using WiFi module                                                                                                                                                         | UART To USB                          |
| Sending and Receiving data in a serial monitor using Bluetooth module                                                                                                                                     | UART To USB                          |
| Controlling a device using a bluetooth module                                                                                                                                                             | UART To USB                          |
| Controlling a device using both bluetooth and WiFi module                                                                                                                                                 | UART To USB                          |

4. **Simulation Output**

4.1.  **Interfacing DS1307 RTC with microcontroller**

DS1307 RTC and microcontroller are been interfaced because there is no built in I2C modules in 8051.
* DS1307 RTC- slave device and microcontroller- master. LCD is also connected with microcontroller, just to display the values received from the RTC.

**Figure 2.** Circuit diagram for DS1307 RTC with microcontroller

* The start a command is to set time to 6:58:48 AM and date to 13/10/20.

* Further on the time gets incremented every second and it can be viewed in the LCD.

**Figure 3.** LCD display for I2C based DS1307 RTC with microcontroller

**Figure 4.** I2C debug window
All the I2C messages are received and displayed in the ‘I2C Debug window’.

4.2. Interfacing the eeprom with microcontrollers using i2c:

I2C is a master-slave protocol, that includes the data along with clock pulse.

- Micro-controller - master.
- EEPROM works as a slave.

![Figure 5. Circuit diagram for EEPROM with microcontroller](image)

- Required data will be stored in the external memory. These data can store in memory by using the I2C, SPI or other communication protocol.
- Single byte will be stored in an EEPROM and later we can read back from EEPROM.
4.3. Establishing Communication between two microcontrollers

Here, one microcontroller which is going to transmit the message is the MASTER and one who receives is SLAVE.

- At, Master side, there will be 2 buttons. button1- up counter, button2- down counter
- The data that is received from microcontroller(1) is displayed on 7 segment display through parallel mode of data transfer.
- The 7 segment display is interfaced with both transmitter and receiver.
4.4. **Communication between two WiFi module**

One ESP8266 goes about as a sender. Other ESP8266 go about as beneficiary. Testing of this arrangement is finished with two ESP8266 sheets all the while. Additionally, can have the option to add more sheets to the setup. The ESP8266 sender gets a recognize message if the messages are effectively delivered.

In Arduino IDE Serial Monitor for the COM port the sender is associated to, it begin getting "Delivery Success" messages with the comparing recipient's MAC address in the Serial Monitor. On the off chance that you eliminate power from one of the sheets, you'll get a "Delivery Fail" message for that particular board. Thus, can rapidly distinguish which board didn't get the message.

4.5. **Controlling devices via serial port using a WiFi Module**

Considering devices working with a different protocols. Controlling their logical output using an WiFi module which can be controlled in the serial monitor of Android IDE. Here, we can vary the value to control the logical output of the device which is done with the help of protocol conversion.

Com port in the figure 10 gives the connectivity state and the execution state of the device. Figure 18 shows the logical output of the device. Left images shows the “N” for off state and the other image states the device is powered ON (“Y” state).
4.6. **Controlling devices via Web Server using WiFi module**

The updated version of the previous output is the controlling of devices via web server. Here, an IP address will be generated in the serial monitor once the code is been uploaded. Using that IP address those devices configured can be controlled via server.

Here in figure 12, we get the generated IP address. The obtained IP address- 192.168.249.70
Figure 13. Implementation to control device via Web server

Figure 13 shows the logical output of the device which is controlled via web server. Powered ON and power OFF state has be mentioned.

4.7. Simulating respirating of 2 led using WiFi module

Even the linear output of the devices can also be simulated using a WiFi module. The rate of the linear output can be controlled using the program.

The output shown in figure 14 gives the brief explanation about controlling of a linear output of a device. The three images shown states the output from high to low state without power OFF.

Figure 14. Implementation to control linear output of the device

4.8. Sending and Receiving data in a serial monitor using Bluetooth module

In this circuit the Arduino IDE just forwards the UART communication to the BT module. Once paired. Open Serial Bluetooth Terminal. Connect to the device.

Figure 15. Data transferred and received in serial monitor
In figure 15, Com port shows the communicated output commands. Commands with > are the commands given from the serial and the other commands are the received commands.

![Figure 15](image1.png)

**Figure 15.** Data transferred and received in Bluetooth terminal

Figure 16 shows the transparent communication between the Com and the terminal. The green commands are the received commands and the blue commands are the sent ones.

![Figure 16](image2.png)

**Figure 16.** Data transferred and received in Bluetooth terminal

Figure 16 shows the transparent communication between the Com and the terminal. The green commands are the received commands and the blue commands are the sent ones.

![Figure 17](image3.png)

**Figure 17.** Output displayed in serial monitor

4.9. **Controlling a device using a bluetooth module**

The device can be controlled by a bluetooth module which helps in home automations. Here, figure 17 shows the output statement when the blink command is sent via bluetooth terminal(“B”). The output is shown in the figure 18.

![Figure 18](image4.png)

**Figure 18.** Implementation to control device via bluetooth module
4.10. **Controlling a device using both bluetooth module and WiFi module**

A single device can be controlled by both bluetooth and WiFi module depending up to the parameters.

![Figure 19. Implementation to control device via bluetooth module](image1)

Here the obtained IP address- 192.168.109.223

LED status can be controlled in web server which controls the device.

![Figure 20. Implementation to control device via WiFi module](image2)

5. **Conclusion**

At the point when two gadgets utilize various conventions, an incongruence issue happens. As such, the two gadgets can't "see" one another and the information probably won't arrive at its objective. That is the place where a third convention change apparatus comes in. Though the device not supports a protocol, device with some other protocol can be interfaced with that device via software module by having a deep knowledge on that protocol.

The Protocol Converter programming intends to encourage interoperability between two applications that utilization distinctive correspondence conventions, however need to trade information. Its preferred position is that it doesn't need any equipment to work, which transforms it into a suitable option in contrast to conventions converters available. In this manner a Protocol Converter is to smooth out information trade between applications without having change the convention interface or the order sets of any of them.
The expansion of the task is to compose programming modules which convert the information designs, information rate and conventions of one organization into the conventions of other and to empower correspondence between two gadgets having various conventions.

References

[1] Trivedi, D., Khade, A., Jain, K., & Jadhav, R. 2018 SPI to I2C Protocol Conversion Using Verilog. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* doi:10.1109/iccubea.2018.8697415

[2] Yi, Q., Shi, M., & Li, S. 2017 Design of USB-UART interface converter and its FPGA implementation *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* doi:10.1109/iaeac.8054244

[3] Blessington, T. P., Murthy, B. B., Ganesh, G. V., & Prasad, T. S. R. 2012 Optimal implementation of UART-SPI Interface in SoC *International Conference on Devices, Circuits and Systems (ICDCS)* doi:10.1109/icdcsyst.2012.6188657

[4] Mr. P. Rajendra Prasad, Mr. S. P. Venu Madhava Rao 2016 Data Transactions from UART to SPI Slave devices through UART-SPI Controller *Advance in Electronic and Electric Engineering*. ISSN 2231-1297, 3(4), pp. 413-420

[5] Jia LIU, Guangmin SUN*, Dequn ZHAO, Xu YAO, Yihang ZHANG 2019 MCU-Controlling Based Bluetooth Data Transferring System *2019 International Workshop on Information and Electronics Engineering (IWIEE)*. Procedia Engineering 29 2109 – 2115

[6] Cong Yanqi, Liu Shijian, Liu Yingli, On UART Transport Layer, the Bluetooth HCI Protocol can be used between the two UART serial ports which are on the same PCB board *Electronic Application*. Beijing, 11, pp. 43-46, 2018.

[7] Jin Chun, Lin Jinzhao, Wan Baohong, 2010 The Bluetooth Protocol And Analysis Of Source Code, 1st edition, *National Defense Industry Press: Beijing*, pp.126-127.

[8] Umakanta Nanda, Sushant Pattnaik 2016 Universal Asynchronous Receiver and Transmitter (UART) DOI:10.1109/ICACCS.2016.7586376. *Conference: 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*

Acknowledgments

We wish to express our sincere gratitude to our beloved Principal In-charge Dr. K. Prakasan for providing an opportunity and necessary facilities in carrying out this project work.

We also wish to express our sincere thanks to Dr. V. Krishnaveni, Associate Professor and Head In-charge, Department of Electronics and Communication Engineering, for the encouragement and support that she extended towards this project work.