Carpathian Journal of Electronic and Computer Engineering 14/2 (2021) 11-14

DOI: 10.2478/cjece-2021-0008

Actuator control using TCP IP communication under LabVIEW USB6001 environment

Guo Zenan
Department of Mechatronics
University of Debrecen
Faculty of Engineering
Debrecen, Hungary
guozenan@eng.unideb.hu

Buchman Attila
Department of Mechatronics
University of Debrecen
Faculty of informatics
Debrecen, Hungary
buchman.attila@inf.unideb.hu

Péter Tamás Szemes
Department of Mechatronics
University of Debrecen
Faculty of Engineering
Debrecen, Hungary
szemespeter@eng.unideb.hu

***Abstract*—This article introduces a basic LED control using the USB6001 hardware, which is a LabVIEW product. It may be thought of as an extension device for real-time testing of simulation results. LabVIEW2014 is the software used. TCP is the communication technique, which has already been incorporated in LabVIEW via one of the communication modules. It might also be done on the same platform. There are two objects in this article, a server and a client, that may transport data or messages between these two applications. The Transmission Control Protocol governs the transmission process.**

***Keywords*—Internet of Thing (IoT), LabVIEW, USB6001, Transmission Control Protocol**

I. Introduction

The "Internet of Things Connected" is the Internet of Things (IoT, Internet of Things). It's an Internet-based network that's been extended and enlarged. It is a massive network created by connecting numerous information detecting devices. The interconnectedness of people, machines, and things may be achieved at any moment and in any location[6].

It refers to the collection of sound, light, and heat in real-time from any objects or processes that need to be monitored, connected, and interacted with using various information sensors, radio frequency identification technologies, global positioning systems, infrared sensors, laser scanners, and other devices and technologies. Various required information, such as electricity, mechanics, chemistry, biology, location, and so on, can be accessed through various possible networks in order to realize the ubiquitous connection between things and people, as well as to realize intelligent perception and recognition of objects and processes. The Internet of Things (IoT) is a data transport system that uses the Internet, traditional telecommunications networks, and other technologies. It enables all common physical things that may be addressed separately to be joined to form a network[5].

II. Architecture and Protocols

The Internet of Things (IoT) has the potential to connect trillions or perhaps billions of objects in a variety of configurations. As a result, in industrial applications, flexible design is crucial. The ever-growing number of suggested architectures has yet to settle on a standard model[4].

A. Objects Layer

The object layer, which is the physical sensor for acquiring and processing information in the Internet of Things, is the initial layer. Sensors and actuators are used to perform a variety of functions, such as location, temperature, motion, vibration, acceleration, humidity, and so on. Through a secure link, the perception layer will digitize and transfer data to the object abstraction layer. At this layer, the Internet of Things generates big data.

B. Object Abstraction layer

Object abstraction transmits the data generated by the object layer to the service management layer through a secure channel[9]. Data can be transferred through various technologies such as RFID, 3G, GSM, UMTS, WiFi, Bluetooth Low Energy, infrared, ZigBee, etc[1].

C. Service Management Layer

Based on address and name, the middleware or service management layer (peering) matches the service with its requester[1]. Through the network line protocol, this layer also analyses the received data, makes judgments, and offers the appropriate services[3].

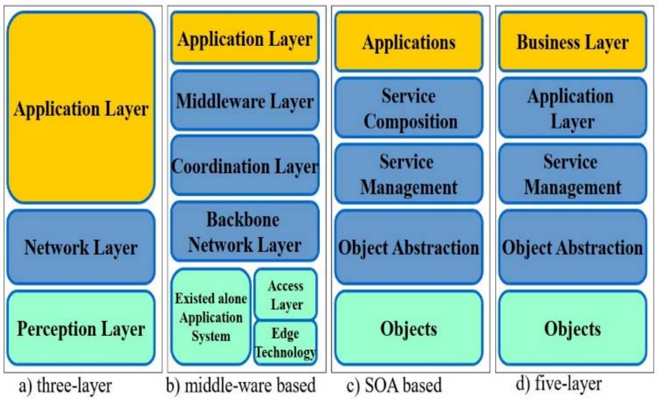

Fig. 1. IoT architecture[1]

D. Application Layer

Customers might receive services from the application layer. For example, the application layer may offer customers the measured temperature and humidity they require[1]. It can provide high-quality intelligence services to fulfill the needs of visitors. Intelligent family applications, intelligent buildings, traffic, industrial automation, and intelligent medical healthcare are all covered by a vertical expansion[7].

E. Business Layer

The IoT operations and services are managed by the Business layer. The task include creating a business model, graph, and block diagram, as well as getting data from the application layer. The business model allows for the processing of decisions based on big data analysis. It may also keep an eye on and regulate the other four levels at the bottom. It compares the output values to the expected values of each layer, thereby improving services and protecting customers' privacy.

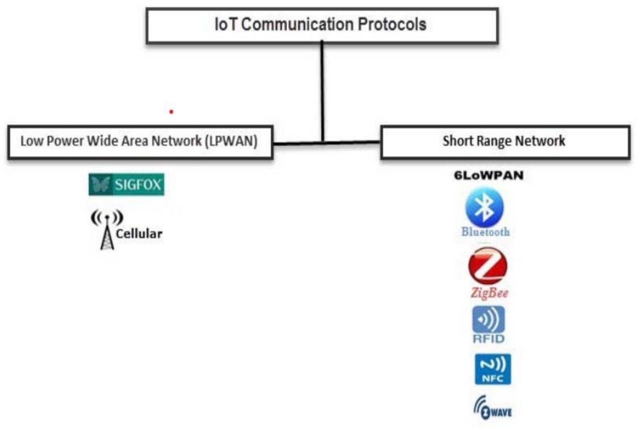

Fig. 2. IoT communication protocols[2]

The branches of IoT communication protocols are depicted in Fig. 2, which may be classified as (A): LPWAN and (B): Short-range network[2].

III. IOT COMMUNICATION MODELS

From an operational standpoint, it is vital to evaluate the Internet of Things connectivity and communication paradigm via technology. In the following, certain frames will be addressed, and the important points of each model will be provided in a simple manner[2]. The various models depict many functions or applications for various communications scenarios.

A. Device-to-Device Communications

Instead of requiring a middle software application to communicate, device-to-device models demonstrate the direct connection and communication between two or more devices. There are several forms of the web that people use to connect with one another, including IP networks and the Internet. In the meanwhile, these devices or applications would link directly to one another via protocols such as Bluetooth, Z-Wave, or ZigBee[2].

B. Device-to-Cloud Communications

In the Device-to-Cloud communication model, an Internet of Things device connects to the internet cloud. For example, an application program service provider could exchange data in order to govern the flow of information or messages. This approach utilizes a communication protocol such as Wifi or Ethernet to build a link between the network or Internet and the Cloud service, which is then connected to the Cloud service via the application and IP networks[2].

The gadget and cloud services are usually provided by the same company. If there are proprietary protection data between the device and the Cloud service, the user's or owner's device might be linked to it. Meanwhile, while utilizing a gadget that is integrated with a certain platform design, the user might feel relieved[2].

C. Device-to-Gateway Model

The device-to-gateway model, or the more traditional device-to-application-layer gateway paradigm, is used. The ALG services, which is the Cloud service's tunnel, link the Internet of Things devices to the gateway. It indicates that the software application program is operating on the local gateway device and that this software serves as a conduit between the device and the Cloud service, providing secure security and other services such as data translation or protocol conversion[2].

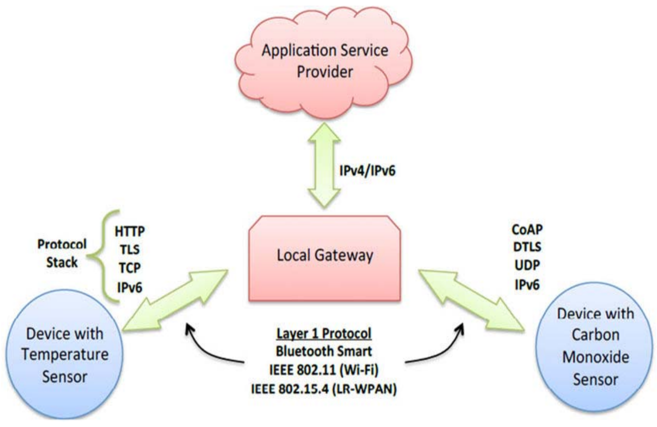

Fig. 3. device to gateway communication[2]

There are numerous stacks of protocols, such as HyperText Transfer Protocol (HTTP), Transport Layer Security (TLS), Transmission Control Protocol/Internet Protocol (TCP/IP), and IPv6, as seen in Fig. 3.

D. Back-End Data-Sharing Model

The back-end data-sharing model is a communication architecture that may assist users in combining data from many sources before extracting and analyzing object data from the Cloud service. Meanwhile, the effective back-end data sharing structure allows users to exchange data when IOT services are switched, allowing this architectural structure to speed up the requirement for data transmission. It dismantles the usual data-blocking barrier. To ensure interoperability of intelligent device data housed in the cloud, it is recommended that a federated Cloud service or cloud application programming interface be used[2].

IV. METHODOLOGY OF THE LED CONTROL USING USB6001

The USB6001 is a multifunction DAQ device with a modest price tag. It has analog and digital I/O, as well as a 32-bit counter. Basic capability is provided by the USB6001 for applications including simple data logging, portable measurements, and academic lab studies. The NI USB6001 Pinout is shown in the following Fig. 4.

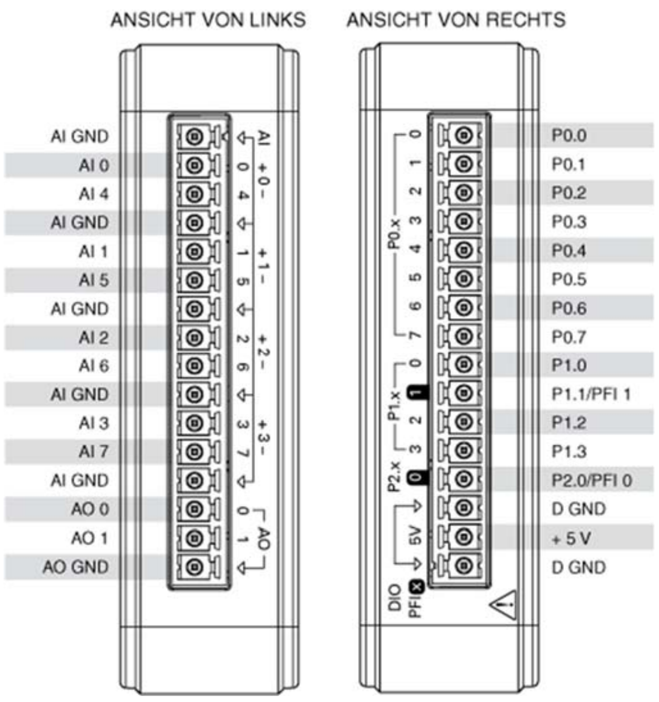

Fig. 4. NI USB6001 Pinout[10]

On the left side of the NI USB6001 board, the analog input port 0 and the ground are connected to the potential meter with the breadboard. The third pin of the potential meter is the 5V power supply. Then the potential meter could send the signals to the NI USB6001 device through the analog input port 0. LabVIEW program will receive the signal and the then send it to the analog output port on the USB6001 device. Then analog output port 1 and analog output port 0 provide the enough voltage to the LED.

When the voltage reaches the limit, the light-emitting diode (LED) will light up. The potential meter has three ports: two for positive and negative signals, and one for a 5V power source. The potential meter is then used to regulate the LED's input voltage, and the input voltage may convey the signal through the USB6001's Analog Input connection. The signal might be sent to the LED using the analog output connector.

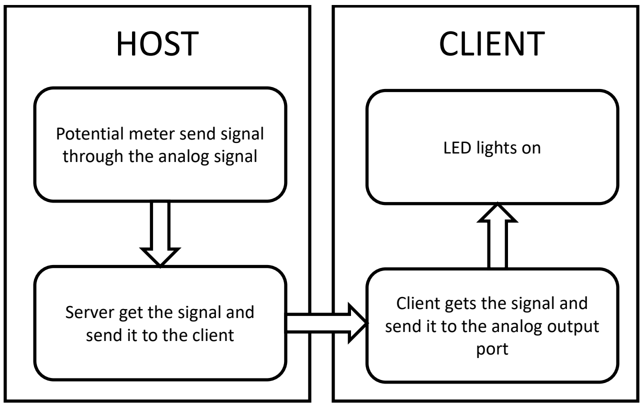
HOST
CLIENT
Potential meter send signal through the analog signal
Server get the signal and send it to the client
LED lights on
Client gets the signal and send it to the analog output port

Fig. 5. Block diagram of the whole system

The Analog Input and Analog Output loops can be used to split the function loop into two parts. Because this article is about the Internet of Things, TCP can offer transmission between two applications based on the TCP concept. In the overall system, there is a server and a client. The server is the host that allows one software to transmit a signal or message to another. The receiver, or client, is a program that can receive a message from another program. The software in this paper is a LabVIEW Virtual Instrument (VI).

The server, client, input, and output loops make up the entire system. Because the input value is obtained by the server Loop, the two VIs might be combined into one. The Client and Output Loops use the same technique. The system's stages are depicted in the block diagram above (Fig. *1*Fig. 5).

The Server Block Diagram in LabVIEW is shown in Fig. 6. in the following. The upper loop is the Analog Input loop, which gets the voltage from the potential meter, and the lower loop is the TCP communication module in LabVIEW, where the port address can be anything as long as it matches the Client address port.

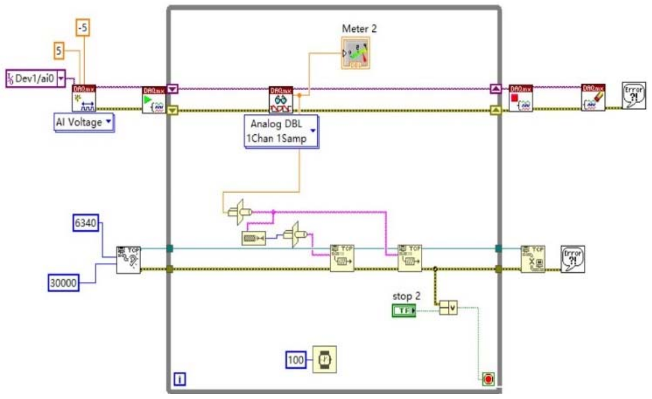

Fig. 6. The server of the system

In the following, the Client loop is shown in Fig. 7. The lower loop is the client loop, which receives the signal from the server and sends it to the top loop, which is the Analog Output loop. The LED could then function.

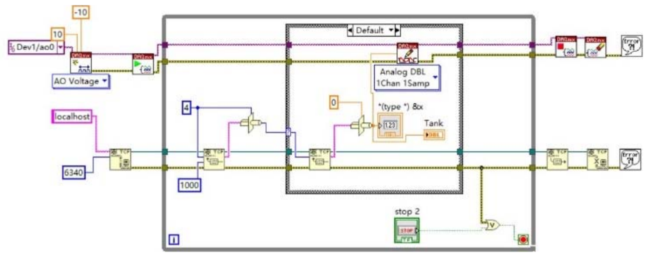

Fig. 7. The client of the system

The TCP uses a three-way handshake, which is the foundation of the LabVIEW Transmission Control Protocol. It is not contained in the above-mentioned software code. The USB6001 and the breadboard with the potential meter are shown in Fig. 8.

The voltage input is controlled by the potential meter, which then sends the signal to the USB60001. The message is received by the TCP server LabVIEW code, which then sends it to the Client. The Client receives the signal and shares it with the TCP analog output terminal, where the fixed voltage is then sent out through the USB6001 Analog Output port to light the LED. The USB6001 may provide the potential meter with a 5V power source.

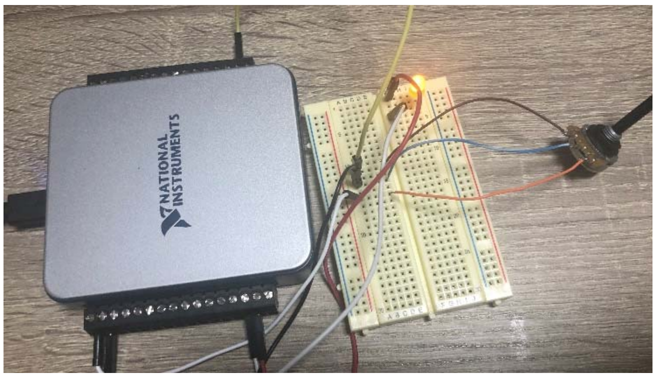

Fig. 8. LED control using TCP communication

ACKNOWLEDGMENT

A LED control approach was provided in this work utilizing LabVIEW and the Transmission Control Protocol. It's sandwiched between two Loops, but it's on the same platform. The software might be implemented on different platforms in the future. One is for the message to be sent by the server, and the other is for the message to be received by the client. To achieve communication, the environment might be WIFI or a cloud service.

REFERENCES

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4):2347–2376, 2015.

[2] Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. Internet of things (iot) communication protocols. In *2017 8th International conference on information technology (ICIT)*, pages 685–690. IEEE, 2017.

[3] Moumena A Chaqfeh and Nader Mohamed. Challenges in middleware solutions for the internet of things. In *2012 international conference on collaboration technologies and systems (CTS)*, pages 21–26. IEEE, 2012.

[4] Srdjan Krco, Boris Pokri ˇ c, and Francois Carrez. Designing iot architecture (s): A european perspective. In *´ 2014 IEEE world forum on internet of things (WF-IoT)*, pages 79–84. IEEE, 2014.

[5] Santosh Kulkarni and Sanjeev Kulkarni. Communication models in internet of things: a survey. *International Journal of Science Technology & Engineering*, 3:3, 2017.

[6] J Sathish Kumar and Dhiren R Patel. A survey on internet of things: Security and privacy issues. *International Journal of Computer Applications*, 90(11), 2014.

[7] Lu Tan and Neng Wang. Future internet: The internet of things. In *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, volume 5, pages V5–376. IEEE, 2010.

[8] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun, and Hui-Ying Du. Research on the architecture of internet of things. In *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, volume 5, pages V5–484. IEEE, 2010.

[9] Zhihong Yang, Yingzhao Yue, Yu Yang, Yufeng Peng, Xiaobo Wang, and Wenji Liu. Study and application on the architecture and key technologies for iot. In *2011 International Conference on Multimedia Technology*, pages 747–751. IEEE, 2011.

[10] Instruments, N. "USER GUIDE NI USB-6001/6002/6003 Low-Cost DAQ USB Device." (2014).