

Development of Virtual Production Camera Control System based on Arduino

Li Penghui¹, Ki-Hong Kim², David-Junesok Lee³, Hou Zhengdong⁴, Zhang Gaohe⁵

¹ Master D. student, Department of Visual Contents, Dongseo University, Korea,
lphuihui@163.com

² Professor, Department of Visual Contents, Dongseo University, Korea, khkim@g.dongseo.ac.kr

³ Assistant Professor, Department of Visual Contents, Dongseo University, Korea,
jsdavid88@g.dongseo.ac.kr

⁴ Ph.D. Student, Department of Visual Contents, Dongseo University, Korea, 1207711532@qq.com

⁵ Ph.D. Student, Department of Visual Contents, Dongseo University, Korea,
zhanggaohe1005@gmail.com

Corresponding author: Ki-Hong Kim

Abstract: Virtual production technology is becoming more mature and is widely used in the production of video content such as movies, drama series, and advertisements. In the two solutions of virtual production (Live LED Wall Cinematography and Live Compositing), the virtual camera and the real camera are required to be identical, that is, after the synchronization of the intrinsic hardware parameters of the real camera and the virtual camera, the exposure parameters of the real camera are changed during shooting, the corresponding parameters of the virtual camera must also be changed to ensure the consistency of the picture. At present, most virtual production studios use camera control systems that are transformed from broadcast camera consoles usually require cumbersome accessories and high prices. During the actual production process, a camera control system using the Arduino development board is found, which inherits the characteristics of easy-to-operate and strong adaptability of the Arduino development board. The serial digital interface camera control protocol provided by Blackmagic Design (BMD SDI CCP), allows the virtual camera (CineCamera) in the virtual production engine Unreal Engine 5 (UE5) and the real camera to communicate in real-time through the use of serial communication. This paper uses literature analysis to explore the necessity and functional requirements of virtual production camera control systems and the feasibility of the virtual production camera system proposed in this study through analysis of academic journals, industry white papers, professional practitioners, and organization presentations. Describe the hardware requirement and design code for the camera control system, verify the feasibility of the system through empirical research, and provide a camera control system that can be customized for individuals or teams who want to make video content through virtual production.

Keywords: Virtual Production, Arduino, Camera control system, Internet of Things, Unreal Engine 5

1. Introduction

Three years after the release of the Star Wars episode The Mandalorian, virtual production has become a key deployment link for many video content production companies, and new virtual production-related

Received: October 26, 2022; 1st Review Result: December 05, 2022; 2nd Review Result: January 10, 2023
Accepted: January 31, 2023

video content production companies are emerging[1]. The advantages of virtual production technology are gradually being seen by more people, and an increasing number of video content production companies choose to use virtual production technology for production[2]. But since the computer-aided visualize virtual production pipeline has only been commercially viable since 2019, there hasn't been much research done on it. The majority of it focuses on the idea of virtual production and pipeline innovation; it is rare to find research on the specific technical details of the virtual production pipeline[1].

In virtual production, how to fully merge the real object and the virtual scene during shooting is a crucial challenge in the In-Camera VFX stage, after projecting the virtual scene onto the LED Wall or using the green screen real-time compositing for visualization. This requires that in the virtual shooting of WYSIWYG, the camera shooting the actual object and LED Wall or green screen background, in reality, needs to be consistent with the parameters of the camera shooting the virtual background in the virtual scene in all aspects. After the calibration and synchronization before the shooting, the photographer or photography assistant can synchronize the parameter of the real camera and the virtual camera in shooting[3]. This requires the existence of a control system that links the real camera and the virtual camera.

Most companies will use the camera control system transformed from the traditional broadcast camera console. This camera control system is expensive and needs to connect the real camera to the computer and controller through a variety of wires, which limits the freedom of the camera to a certain extent [5]. Some large-scale virtual production studios will develop camera control systems for their own cameras. Industrial Light and Magic mentioned in "The Virtual Production of The Mandalorian Season Two" published on YouTube that the camera control systems they use are integrated and developed software for iPad[4]. Chinese virtual production company Versatile Media mentioned in UnrealCircle, in July 2022 that the camera control system they use consists of a physical button panel developed by themselves[5]. These system patents are usually held within these companies and will not be disclosed to the public. The virtual production camera control system described in this paper will break down barriers and construct an easy-to-operate and strong adaptability system for virtual production practitioners to use.

The concept of the IoT has recently become increasingly popular across a variety of industries. This study, which is motivated by the idea of the IoT, develops a full wireless control system for virtual production cameras using Arduino, an open-source electronic design platform widely used in the IoT, and BMD SDI CCP, describes the production process of the virtual production camera control system and uses pseudocode and UE5 blueprint for a better explanation. The virtual production camera system proposed in this paper will enrich the research on the details of virtual production technology and provide technical ideas for relevant practitioners. The complete Arduino sketch and UE5 blueprints used in this paper can be viewed in the reference link[6].

2. Related Research

Due to the lack of direct research on the virtual production camera control system, this paper selects and analyzes a number of sources, including academic journals, industry white papers, professional practitioners, and organizational presentations, to demonstrate the validity and objectivity of its claims. It then uses these analyses to demonstrate the necessity of the virtual production camera control system it proposes as well as its viability.

In recent years, with the rise of the concept of the IoT, Arduino has become a very popular open-source electronic design platform. Not only can Arduino be used to create basic everyday necessities, but it can also be utilized to create some automated Internet of Things tools that can increase productivity. At present, Arduino-based toys, temperature, and humidity monitoring, smart home, and

even more complex electronic production are gradually becoming known to the public. More people use Arduino to connect things around them, or make tools to improve their efficiency in work or life.

Patnaik P. and Dinkar R.[7] mentioned in their article that can cope with various communication protocols and supports a variety of sensors, low Power consumption Bluetooth and WI-FI module, with strong scalability. Majid K. and Gabrielle V.[8] create a portable continuous measurement toolbox that offers a powerful solution for indoor environmental quality (IEQ) monitoring and performance assessment, easy to expand, and low-cost setup, using the ZigBee communication protocol to connect the XBee device connected to each Arduino and the XBee receiver connected to the receiver board. Muhammad A. and Khalil U.[9] proposed a low-cost and user-friendly remote-controlled home automation system using an Arduino board, Bluetooth module, smartphone, ultrasonic sensor, and humidity sensor.

The Arduino Bluetooth module enables the Arduino board to communicate with the computer via UART (Universal Asynchronous Receiver Transmitter). When the Arduino Bluetooth module is connected to the computer, the computer will automatically assign a serial port to the Arduino, and the potential changes generated by the Arduino sensor module will be transmitted to the computer in the form of "0&1", and the computer will compile these numbers through the serial communication protocol[10].

BMD SDI CCP is a set of camera control protocols provided by Blackmagic for the camera control system developed by them. This control system mainly consists of ATEM switches, Blackmagic 3G-SDI Shield for Arduino, and the Blackmagic Camera Control app. Developers can take advantage of the expandability of Arduino and make a console of their own with the help of BMD SDI CCP[11].

3. Methods of the Study

This paper uses literature analysis to explore the necessity and functional requirements of virtual production camera control systems and the feasibility of the virtual production camera system proposed in this study through the analysis of academic journals, industry white papers, professional practitioners, and organization presentations. Through the empirical research, design an experiment to demonstrate the flow of controlling a real camera and a virtual camera simultaneously using Arduino, and explain the code as well as the blueprint logic in the research process.

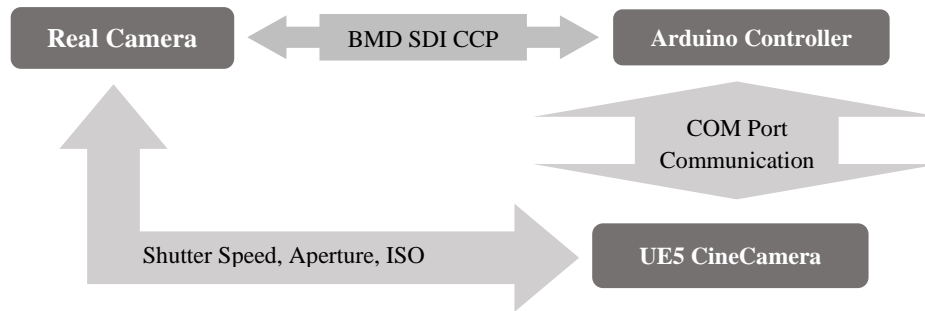
4. Construction of a Camera Control System based on BMD SDI CCP and Arduino

This part will discuss the construction of a virtual production camera control system based on Arduino and BMD SDI CCP through empirical research. After clarifying the requirements that the virtual production camera control system needs to meet, this section will be divided into two stages "construction conditions" and "construction process" to explain the construction details of this system. To better explain the construction logic, this paper uses LaTeX to build the relevant pseudocode, quotes some UE5 blueprints for an auxiliary explanation, and lists and explains the data in the experiment.

In virtual production, the pictures taken by the real camera must be exposed synchronously with the pictures taken by the virtual camera. This requires that after the synchronization between the virtual scene and the actual lighting is done in the early stage of shooting, the camera control must have bi-directionality, that is, the camera control system must be able to control the real camera and make corresponding controls on the virtual camera to ensure that the actual object consistent with the exposure of the virtual space.

4.1 Construction Conditions of Virtual Production Camera Control System

This study uses the BMD SDI CCP and the serial communication protocol between the Arduino development board and the computer to make a virtual production camera controller that enables the real camera to communicate with the CineCamera in UE5 in real-time. The controller is mainly composed of Blackmagic 3G-SDI Shield for Arduino, JARDUINO-UNO-BT Board, and YwRobot button board. [Fig. 1] shows the basic workflow of this controller.



[Fig. 1] Virtual Production Camera Control System Workflow

The camera used in this study is the URSA Mini Pro 12K launched by BMD (it can also be replaced with any camera that supports BMD SDI CCP). This development board is a Bluetooth-integrated Arduino UNO board, and the communication mode with the computer may be swapped between USB and Bluetooth. It works in conjunction with the Blackmagic 3G-SDI Shield for Arduino to realize wireless communication and Multiple parameter adjustment. (It can also be controlled by any Arduino UNO development board instead, but to realize wireless, it is important to use an Arduino Bluetooth module) and an YwRobot button board integrating eight buttons. In terms of power supply, a large-capacity Power Bank and a DC5.5*2.5 to USB, 5V To 12V boost line are used to supply power to the Arduino development board and Blackmagic 3G-SDI Shield.

To enable real-time communication between the Aduino development board and UE5, this study uses the "Unreal_Engine_SerialCOM" plug-in developed by Videofeedback to build the system. This plug-in is based on the computer serial communication protocol, and can transmit the data generated by any hardware device capable of serial communication with the computer to UE5 through the serial communication port[12].

4.2 Construction Process of Virtual Production Camera Control System

The process of ordinary video shooting usually uses many parameters, such as controlling the exposure level of the picture by adjusting the shutter speed, aperture, and ISO of the camera, and calibrating the color of the picture by adjusting the white balance of the camera[13]. Like ordinary image shooting, in virtual production, it is also necessary to adjust the parameters of various cameras to ensure that the picture can be shot correctly. It can be known from BMD SDI CCP that most of the common parameters of cameras supporting this protocol can be programmed and adjusted through the development board. This paper mainly explains the adjustment and programming of shutter speed, aperture, and ISO. The adjustment of other parameters can be set according to the sample code provided above and the official Developer Information provided by BMD.

The construction of the virtual production camera controller system based on BMD SDI CCP and Serial Communication Protocol is mainly divided into two steps. The first step is to program the controller in the Arduino IDE, so that the controller can communicate with the camera to achieve the

purpose of controlling the camera. The second step is to use the Unreal_Engine_SerialCOM Plugin to synchronize the actual camera adjustment information with the virtual camera in UE5 by upgrading the Arduino script and writing the UE5 blueprint.

When programming and controlling the camera, first call the BMD SDI CCP library in the Arduino IDE, set the communication address between the Arduino development board and the Blackmagic 3G-SDI Shield for Arduino, and define the variables to be used and the buttons ID of the button board. Second, the control right of the camera is invoked through the BMD SDI CCP. When the potential of the button on the button board changes, define the value of the camera to change as a variable and change it, so as to adjust the value of the camera.

Taking the shutter speed as an example, define the SW2 button on the button board as the shutter speed increase button and SW6 as the shutter speed decrease button, as shown in [Fig. 2]. Define the state of the button as "currentState" and "lastState", where the initial "currentState" is the digitalRead obtained from the Arduino Uno digital pin (presented in the form of high and low potentials), and the initial "lastState" is directly defined for high potential. When the button is pressed, "currentState" becomes high potential, "lastState" becomes low potential, "lastState = currentState", and a cycle is completed. Use "if(lastState == LOW && currentState == HIGH)" to detect button state changes.



[Fig. 2] Define SW2&SW6 Buttons as Shutter Speed Adjustment Buttons

In the process of adjusting the shutter speed, define two integer variables "currentShutterspeed" and "currentShutterspeedAdjust", where "currentShutterspeed" represents the reciprocal of the current shutter speed and "currentShutterspeedAdjust" represents the altered shutter speed value. In actual shooting, there will also be certain specifications for the use of shutter speed values, which will not be arbitrary values, and the aperture and ISO are also the same. Therefore, to restore the shooting more realistically, the shutter speed and other parameters of the virtual production camera can only be the same as those of the actual camera, and can be adjusted between several specific values. At this time, set "currentShutterspeedAdjust" as an array "currentShutterspeedAdjust[] = {}", where the value in {} is the reciprocal of the adjustable shutter speed value of the actual camera, here arrange the value in {} as: 24, 30, 50, 60, 100, 125, 200, 250, 500, 1000, 2000. The integer variable "s" is introduced as the serial number of the array.

[Fig. 3] shows that the shutter speed increases after the button SW2 is pressed, the shutter speed slows down after the button SW6 is pressed, and the control signal is transmitted to the camera process through the BMD SDI CCP. When the output port of the development board is connected to the input port of the camera through an SDI cable, the camera can be controlled. Then use the same way to let SW3 and SW7 on the button board control the increase and decrease of the camera aperture, and SW4 and SW8 control the increase and decrease of the camera ISO, realizing the basic parameter control of the camera exposure. When the output port of the development board is connected to the input port of the camera

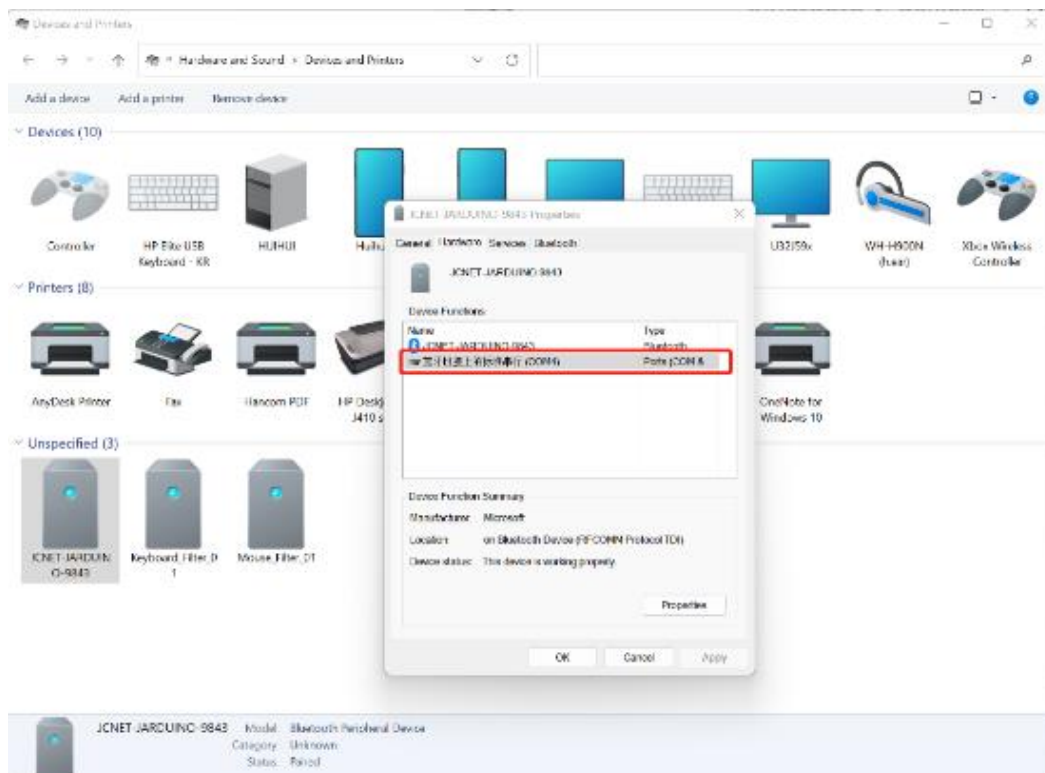
through the SDI cable, the control of the camera can be realized.

Code 1: Use BMD SDI CCP to change camera parameters

```

Data: currentState2, lastState2, currentState6, lastState6,
        currentShutterspeed, currentShutterspeedAdjust,
        ShutterspeedRange s
Result: Change the shutter speed of the camera through the Arduino
            button board
1 while the button potential on the button board changes do
2   send signal to camera via BMD SDI CCP;
3   if lastState2 = low and currentState2 = high then
4     | s = s + 1;
5     | currentShutterspeed = currentShutterspeedAdjust[s];
6     | 0 ≤ s ≤ 10;
7   else
8     | currentShutterspeed = currentShutterspeed;
9   end
10  /* increase shutter speed when button 2 is pressed */
11  if lastState6 = low and currentState6 = high then
12    | s = s - 1;
13    | currentShutterspeed = currentShutterspeedAdjust[s];
14    | 0 ≤ s ≤ 10;
15  else
16    | currentShutterspeed = currentShutterspeed;
17  end
    
```

[Fig. 3] After the Button Potential Changes, the Change Process of the "currentShutterspeed" Variable



[Fig. 4] Confirm the Communication Port ID between the Development Board and the Computer

To check the controller's control of the camera, use the tool Serial Monitor in the Arduino IDE to view the code running results. First, need to connect the development board to the computer with Bluetooth, and obtain the communication port ID for the communication between the development board and the computer from the "devices and printers" setting, as shown in [Fig. 4]. Different devices have a specific baud rate when communicating with the computer. Generally, the baud rate between the Arduino Bluetooth module and the computer is 9600, so it is necessary to add the "Serial.begin(115200)" statement to the Arduino Script to realize communication between the Arduino development board and the computer. Additionally, use the "Serial.print" and "Serial.println" statements to transfer the modified shutter, aperture, and ISO value systems. The ISO parameters are passed to the computer in the format of "currentShutterspeed=**", "currentAperture=**", "currentISO=**".

[Table 1] shows the output of the serial port when the controller is used to adjust the real camera after the above code is built and the hardware and software are set up.

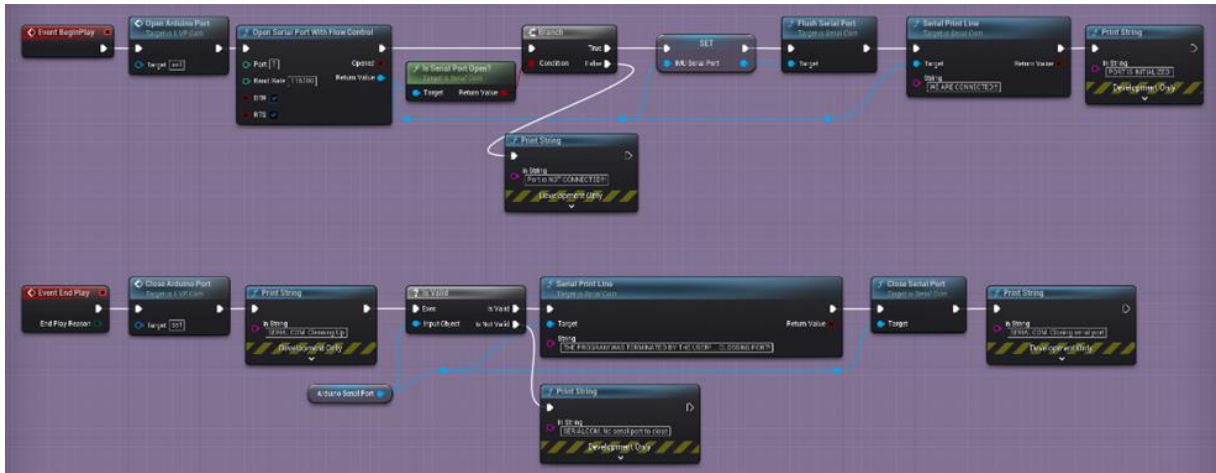
[Table 1] After Pressing the Controller Button, the Parameters of the Camera Change Simultaneously with the Code Output

	Camera parameters	Serial Monitor
Oringina	ShtterSpeed = 1/50 Apperture = 2.8 ISO = 400	ShtterSpeed = * Apperture = * ISO = *
Press SW6 button once	ShtterSpeed = 1/50 Apperture = 4.0 ISO = 400	ShtterSpeed = 1/50 Apperture = 4.0 ISO = 400
Press SW7 button once	ShtterSpeed = 1/30 Apperture = 4.0 ISO = 400	ShtterSpeed = 1/30 Apperture = 4.0 ISO = 400
Press SW8 button once	ShtterSpeed = 1/30 Apperture = 4.0 ISO = 320	ShtterSpeed = 1/30 Apperture = 4.0 ISO = 320

After completing the controller's one-way control of the real camera, the Unreal_Engine_SerialCOM Plugin is used to transfer the data from the Arduino development board to UE5 through the computer serial communication port and create blueprints to extract data from the development board and apply it to the virtual camera to achieve control of the virtual camera and achieve direct communication between the virtual camera and the real camera.

Make UE5 loads the Unreal_Engine_SerialCOM Plugin, and use the sample blueprint in the plugin to open the serial communication port between the controller and the computer, as shown in [Fig. 5]. Input the COM PORT ID and baud rate obtained in [Fig. 4] into the "Open Serial Port With Flow Control" node.

Through the above process, the communication port between the controller and the computer can be opened. Then use the "Event tick" event node to obtain the camera adjustment information transmitted by the controller in the "serial read line" in real-time. At this time, the format of the information input by the controller to UE5 is the same as that in [Table 1], and needs to be processed again in UE5. The pseudocode in [Fig. 6] shows the processing flow of redefining the data after the string signal of Arduino is transmitted to UE5 through the serial communication protocol and applied to CineCamera in UE5.



[Fig. 5] Use Plugin's Example Blueprint to Manage the Communication Port of the Controller

Code 2: Apply the adjusted parameters in UE5's CineCamera

```

Input: "currentShutterspeed = ss", "currentAperture = ap",
         "currentISO = iso".
/* "ss", "ap", "iso" are float variable */
Result: Use Arduino to control the exposure of CineCamera
1 if serial port open and Arduino data input successfully then
2   if input string = "currentShutterspeed = ss" then
3     | set the ss value as CineCamera's shutterspeed
4   else
5     if input string = "currentAperture = ap" then
6       | set the ap value as CineCamera's aperture
7     else
8       if input string = currentISO = iso then
9         | set the iso value as CineCamera's ISO
10      else
11        | Prompt error
12      end
13    end
14  end
15 else
16   | Prompt error;
17 end

```

[Fig. 6] Apply the Redefined Controller Data to CineCamera

The controller has realized the two-way control of the real camera (URSA Mini Pro 12K) and the virtual camera (UE5 CineCamera). After the hardware parameters are unified in the early stage, the requirements for the unified exposure of the virtual production camera can be realized. The same method can also be used to encode other adjustable parameters of the camera, so as to achieve more efficient and more suitable control of the virtual production camera.

4. Conclusions

Inspired by the application of Arduino in the field of the IoT, this paper makes a low-cost camera control system suitable for virtual production based on BMD SDI CCP and serial communication protocol. This system has most of the functions required by the virtual production camera control system. It can link the real camera and the virtual camera in the virtual production tool UE5. After the exposure and distortion calibration of the camera in the early stage, the parameters of the two cameras can be adjusted synchronously through the Arduino controller during the virtual shooting, so as to achieve the unified effect of real foreground and virtual background exposure. If it is necessary to increase the parameter control of other parameters of the camera, use the method mentioned above to refer to the BMD SDI CCP and program it. The Arduino development board communicates with the computer through the Bluetooth module, which reduces the physical connection between the camera and the computer, increasing the freedom of the camera.

The concept of using the serial communication between the Arduino development board and the computer to make a virtual production camera control system proposed in this paper applies to the production of most virtual production controllers. For example, the brightness and position of lights in the virtual scene can be physically mapped through the Arduino platform and serial port communication. Practitioners of virtual production, whether they are related technicians, cinematographers, lighting engineers, etc., can use the Arduino development board to improve the workflow and improve production efficiency according to their working habits.

Limited by the communication protocol between the real camera and the development board, the camera control system proposed in this study is not suitable for cameras that do not support BMD SDI CCP. Researchers can use the method used in this study to find the communication protocols of different camera manufacturers to build their own virtual production camera control systems.

References

- [1] M. Kavakli and C. Cremona, The Virtual Production Studio Concept – An Emerging Game Changer in Filmmaking, 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), IEEE, pp.29-37, (2022)
DOI: <https://doi.org/10.1109/VR51125.2022.00020>
- [2] Kim Mi Ra, Post Coronavirus, a shift of Video Contents Production Technology, Journal of the moving image technology association of korea, (2021), Vol.1, No.35, pp.27-44.
DOI: <https://doi.org/10.34269/mitak.2021.1.35.002>
- [3] N. Kadner, The Virtual Production Field Guide. Volume 1, Epic Games, (2019)
Available from: <https://cdn2.unrealengine.com/Unreal+Engine%2Fvpfieldguide%2FVP-Field-Guide-V1.2.02-5d28ccec9909ff626e42c619bcbe8ed2bf83138d.pdf>
- [4] https://www.youtube.com/watch?v=-gX4N5rDYeQ&ab_channel=IndustrialLight%26Magic, Nov 1 (2022)
- [5] https://www.bilibili.com/video/BV15r4y1E7ip/?spm_id_from=333.880, Nov 1 (2022)
- [6] https://drive.google.com/drive/folders/1fRMyWr88IZG6xfje3CQ7kqlmYKBI_puf?usp=sharing, Nov 1(2022)
- [7] Patnaik Patnaikuni, R. Dinkar, A Comparative Study of Arduino, Raspberry Pi and ESP8266 as IoT Development Board, International Journal of Advanced Research in Computer Science, (2017), Vol.8, No.5, pp.2350-2352.
DOI: <https://doi.org/10.26483/ijarcs.v8i5.3959>
- [8] M. Karami, G. V. McMorro, L. Wang, Continuous monitoring of indoor environmental quality using an Arduino-based data acquisition system, Journal of Building Engineering, (2018), Vol.19, pp.412-419.
DOI: <https://doi.org/10.1016/j.jobee.2018.05.014>

- [9] Asadullah, Muhammad, Khalil Ullah, Smart home automation system using Bluetooth technology, International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), IEEE, (2017)
Available from: <https://ieeexplore.ieee.org/abstract/document/7916544>
- [10] Agus Kurniawan, Arduino Uno: A Hands-On Guide for Beginner, PE Press, (2015)
- [11] <https://www.blackmagicdesign.com/developer/product/arduino>, Nov 1 (2022)
- [12] https://github.com/videofeedback/Unreal_Engine_SerialCOM_Plugin, Nov 1 (2022)
- [13] N. Nourani-Vatani, J. Roberts, Automatic camera exposure control, Proceedings of the Australasian Conference on Robotics and Automation, Australian Robotics and Automation Association (ARAA), (2007)
Available from: <https://eprints.qut.edu.au/82536/>