TalentDetector
17th-19th June 2024
Gliwice-Szczyrk, Poland
DEPARTMENT OF ENGINEERING MATERIALS AND BIOMATERIALS
FACULTY OF MECHANICAL ENGINEERING
SILESIAN UNIVERSITY OF TECHNOLOGY
INTERNATIONAL STUDENTS SCIENTIFIC CONFERENCE

Wireless communication between a flight simulation and its controllers

Michał Lasak[a], Franciszek Borgosz[b], Piotr Bartosz[c], Dominik Bereta[d], Dariusz Myszor[e], Maria Ochman[f], Jakub Wieczorek[g]

[a]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN SpaceCoffee
email: mlasak@spacecoffee.net
[b]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN SpaceCoffee
email: franek.borgosz@gmail.com
[c]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN vFly
email: db308060@student.polsl.pl
[d]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN vFly
email: piotbar578@student.polsl.pl
[e]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, Department of Algorithmics and Software
email: dariusz.myszor@polsl.pl
[f]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN vFly
email: mo306378@student.polsl.pl
[g]Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science, SKN vFly
email: jw306100@student.polsl.pl

Abstract: This article presents the results of research conducted to evaluate potential methods for achieving error-free and wireless communication between a device running flight simulation software and its controllers. The issues that the research sought to address are not uncommon, yet the proposed solutions often entail significant costs or present implementation challenges. Therefore, an inexpensive and straightforward solution was sought.

Keywords: wireless, connection, flight simulation, yoke system, controller

1. INTRODUCTION

Figure 1. Flight simulation software

Flight simulation technology has witnessed remarkable advancements in recent years, offering unparalleled realism and immersion to enthusiasts and professionals alike. Central to the efficacy of these simulations is the seamless interaction between users and control interfaces. It is often difficult to imagine that completely wireless communication between flight simulation controls, such as pedals and yoke, and the simulation itself would be necessary. However, as our colleagues began to work on the simulator situated on the moving platform, the issue became evident.

Figure 2. The platform on which the simulator is situated.

The purpose of the research is to identify a straightforward approach to establishing a wireless connection between the two devices, utilizing pre-built, publicly available, plug-and-play versions of both the software and the controllers. The objective is to identify a solution that is both straightforward to implement and highly accurate.

2. INITIAL VERSION

Upon initial examination, it was hypothesized that a regular network connection could be established between a compact computing device, such as the Raspberry Pi, connected to the flight simulation controllers, and a computer running the simulation software.

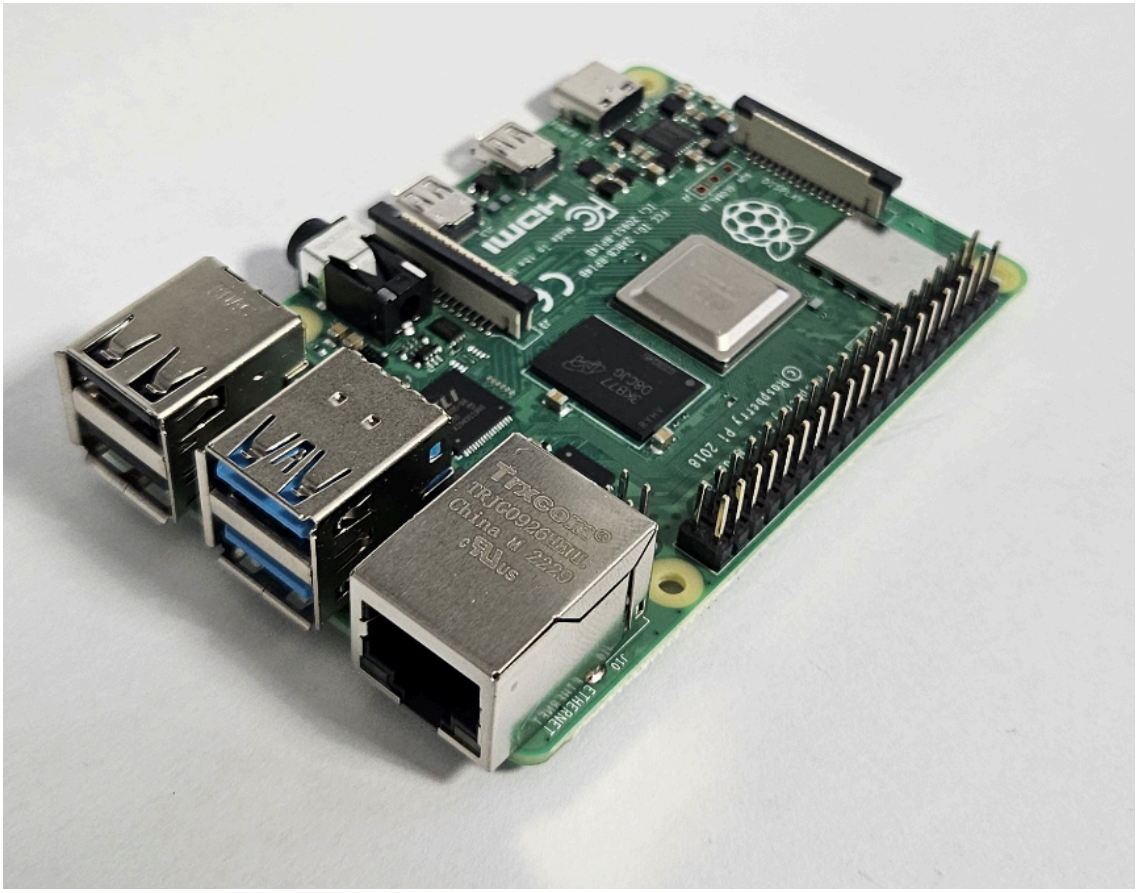

Figure 3. Raspberry Pi - a potential candidate for utilization as the transceiver in the aforementioned solution.

Although the solution appears straightforward, it is, in fact, quite complex [1, 2]. The majority of publicly available flight simulation controllers are connected via the Universal Serial Bus (USB). In fact there exists a simple method for wireless data transmission from the controllers. However, it is challenging to persuade a computer to interpret the received data as if it were acquired from the USB protocol [8]. Furthermore, we attempted to develop software that would be capable of receiving data from the aforementioned controllers.

The initial investigation involved the utilization of an external network adapter in conjunction with the Raspberry Pi Zero board. Our approach led to the conclusion that the development of self-designed software would be necessary, as existing pre-made software did not meet our expectations. The design and implementation of error-free, long-lasting, and robust software is a challenging endeavor. Furthermore, the challenge is compounded by the fact that the solution must be developed for a constantly evolving environment of flight simulation and virtual reality [9, 10].

3. ESTABLISHED SOLUTION

While still considering the implementation of the aforementioned software to correctly interpret controllers' data, we encounter another solution. Upon initial examination, the endeavor appeared to be a formidable challenge. However, upon closer examination, it became evident that the implementation process was relatively straightforward. The solution employs a flight simulation controller set, comprising a yoke and pedals, in conjunction with a regular wireless gaming console controller. The utilization of pre-built devices in this solution necessitates the disassembly of both components in order to reestablish the analog connections of the printed circuit boards (PCBs) of both devices, thereby enabling the gaming pad to function as a transceiver in a ready-made solution [7]. By affixing the console controller to the yoke system, a robust and secure connection can be established. Furthermore, this configuration obviates the issue of interpreting acquired data as the USB protocol data in subsequent stages of the simulation, as the simulation software can receive data directly from the gaming pad.

Figure 4. Xbox controller - a potential candidate for utilization as the transceiver in the aforementioned solution.

The implementation of this solution is relatively straightforward. The integration of a regular wireless gaming pad with the flight simulator controllers set is a key aspect of this solution. The internal wired outputs of the controllers set would be connected to the gamepad PCB, thereby enabling wireless connection. It is indeed possible due to the already existing remote connection integrated into

a gaming controller [12]. Subsequently, the gamepad's receiver can be directly connected to the simulation computer, allowing the utilization of data acquired from the pad's internal PCB within the simulation software. This is possible because these types of gamepads are already supported as input devices for the majority of available flight simulators. Moreover, the aforementioned solution is likely to be highly durable due to the limited number of alterations currently being implemented in flight simulation controllers. The prevailing focus of current developments is on enhancing the simulation itself.

The final design would function as a single device capable of wirelessly controlling any currently available simulation software. Given the utilization of professional flight simulation equipment, the solution can be used to train future pilots in a revolutionary and highly immersive manner [13].

Figure 5 Practical training conducted on the flight simulator

4. CONCLUSION

In conclusion, investigations conducted to establish error-free and wireless communication between flight simulation software and its controllers have yielded a solution that combines simplicity, durability, and effectiveness. Although the achieved results necessitate the utilization of a self-made

device for the remote control of the flight simulator, the solution is relatively simple in its implementation and can be readily reproduced if necessary.

BIBLIOGRAPHY

1. Sohn, J. M., Baek, S. H., & Huh, J. D. (2008, September). Design issues towards a high performance wireless USB device. In 2008 IEEE International Conference on Ultra-Wideband (Vol. 3, pp. 109-112). IEEE.
2. Bit, A., Orehek, M., & Zia, W. (2010, September). Comparative analysis of Bluetooth 3.0 with UWB and Certified Wireless-USB protocols. In 2010 IEEE International Conference on Ultra-Wideband (Vol. 2, pp. 1-4). IEEE.
3. Ng, T. S., & Ng, T. S. (2018). Flight Simulator Systems. Flight Systems and Control: A Practical Approach, 43-53.
4. Cameron, B., Rajaee, H., Jung, B., & Langlois, R. G. (2016, May). Development and implementation of cost-effective flight simulator technologies. In International Conference of Control, Dynamic Systems, and Robotics (No. 126, p. D0I).
5. Vujović, V., & Maksimović, M. (2014, May). Raspberry Pi as a Wireless Sensor node: Performances and constraints. In 2014 37th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1013-1018). IEEE.
6. Vitsas, P. A. (2016). Commercial simulator applications in flight test training. Journal of Aerospace Engineering, 29(4), 04016002.
7. Wagner, M., Avdic, D., & Heß, P. (2016, July). Gamepad Control for Industrial Robots. In Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (pp. 368-373).
8. Dawoud, D. S., & Dawoud, P. (2022). Serial communication protocols and standards. River Publishers.
9. Muñoz-Saavedra, L., Miró-Amarante, L., & Domínguez-Morales, M. (2020). Augmented and virtual reality evolution and future tendency. Applied sciences, 10(1), 322.
10. Kenwright, B. (2019). Virtual reality: Where have we been? where are we now? and where are we going?.
11. van der Linden, J., van der Heijde, F., & van Remundt, K. (2021). Bluetooth Ultrasimple Gamepad: Design of a wireless cooperative gaming device.
12. Thorpe, A., Ma, M., & Oikonomou, A. (2011, July). History and alternative game input methods. In 2011 16th International Conference on Computer Games (CGAMES) (pp. 76-93). IEEE.
13. Caro, P. W. (1973). Aircraft simulators and pilot training. Human Factors, 15(6), 502-509.