

# Construction of Automated Experiment Software Platform Based on Hardware Integration

Yijing Chen<sup>1,2</sup>, Xinyu Tian<sup>1,2</sup>, Wenlong Zhu<sup>1,2</sup>, Yusheng Shao<sup>1,2</sup>, Junjie Zhong<sup>1,2</sup>, Yiming Lu<sup>1,2</sup>, Tianwei Tan<sup>1,2\*</sup>

<sup>1</sup> Beijing Key Lab of Bioprocess, National Energy R&D Center for Biorefinery, Beijing 100029, PR China

<sup>2</sup> College of Life Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

\* Corresponding author, Tianwei Tan: [twtan@mail.buct.edu.cn](mailto:twtan@mail.buct.edu.cn) (T.W.T.)

Manuscript submitted December 10, 2023; revised January 8, 2024; published May 15, 2024.

doi: 10.17706/jsw.19.2.52-63

---

**Abstract:** The combination of automation and biotechnology has led to the emergence of numerous automated biological experiment platforms. However, the utilization of such software necessitates the purchase of a comprehensive equipment package, which hinders secondary development and complicates the user experience. Moreover, the scalability of these platforms in later stages is limited. In this study, the hardware equipment necessary for biosynthesis automation is identified and organized based on the functional modules involved. Subsequently, the development platform is restructured accordingly. The platform incorporates custom communication protocols to facilitate control over individual devices, while the user graphical interface is implemented using C#. The SQL Server database is employed for efficient data management and storage within the platform. This platform enables convenient operation management, modular process design, and future scalability, thereby facilitating the automation of biosynthesis processes and diverse experimental applications. It offers enhanced convenience in managing operations, allows for the modularization of process design, and ensures scalability for future functional expansions. As a result, the platform has the potential to automate the entire biosynthesis process and support a wide range of experimental applications in the future. Finally, utilizing this platform, we successfully conducted a color reaction experiment, leveraging automated operation to minimize human intervention in the experimental process. This approach significantly enhanced experimental efficiency and effectively mitigated experimental errors. By reducing the reliance on manual intervention, the platform proved instrumental in streamlining the experimental workflow and improving overall experimental outcomes.

**Keywords:** Hardware Integration platform, robot, software systems, automated lab, automation

---

## 1. Introduction

As the basis of modern biotechnology [1], biomanufacturing is an industrial approach to the large-scale production of basic materials and chemicals needed by humans [2]. Biomanufacturing has a wide range of applications and plays an important role in the food, paper, pharmaceutical, mining, chemical, and energy fields [3]. However, due to the complexity and low success rate of biological systems [4], a trial-and-error approach is required to perform a large number of repetitive experiments when prototyping biological systems [5], a process that also adds a lot of burden to the experimentalists. Therefore, there is a strong need to develop high-throughput prototyping capabilities for biomanufacturing [6], in which automation

technologies will inevitably be introduced [7].

The application of automation technology in industry has had a significant impact since its early development [8]. The relative stability of underlying protocols in industrial production has facilitated the integration of automation. Moreover, industries that require testing have embraced automation technology, and laboratory automation has recently gained attention [9–12]. Automation technology has enabled the integration of preceding and following processes, allowing biological experiments to transition from mutually independent closed loops to partially interconnected functional networks. Automation technology has freed experimental staff from manual labor, reducing errors caused by human factors and shortening experimental time. As a result, research has become more efficient and precise, with implications for industries such as pharmaceuticals and biotechnology [13].

In recent years, several automated biomanufacturing platforms have emerged. Ginkgo Bioworks, established in 2008 [14], specializes in strain transformation and automation platforms, which integrate hardware and tools provided by upstream technology companies, and create platforms for customers to edit cells for downstream product applications [15–17]. In 2014, Professor Zhao Huimin's laboratory at the University of Illinois at Urbana-Champaign developed the iBioFAB automated synthetic biology system, which combines artificial intelligence and automation to accelerate bioengineering research [18–21]. Thermo Fisher collaborated with the University of Greifswald in 2016 to build the LARA system, a platform used in biocatalysis research that can achieve high-throughput screening of protein engineering mutant libraries [22]. Currently, core technologies such as synthetic biology automation equipment and robot integration are controlled by companies such as Thermo, Beckman, Tecan, HiRes, Hamilton, and Roche. These system integrators provide buyers with proprietary software to assist customers in customizing automated biological platforms to meet their specific needs.

Nevertheless, these bespoke platforms necessitate the procurement of both hardware and software components for effective utilization. Mere acquisition of software proves insufficient, thereby impeding secondary development endeavors and constraining overall flexibility. Moreover, expanding equipment infrastructure during subsequent stages entails protracted interactions with system integration developers to devise a viable solution, resulting in a cumbersome and costly process. The resultant lack of adaptability and scalability poses significant challenges for users striving to tailor and augment their experimental configurations.

Addressing this issue, this paper introduces a novel hardware layout concept and develops a corresponding software system based on drag-and-drop functionality aligned with the hardware arrangement. By identifying the requisite equipment for the experimental procedure, the approach involves the procurement of hardware components and meticulous layout planning. Subsequently, the c#.Net framework [23] is employed to create a user-friendly graphical interface [24], while SQL Server [25] is utilized for constructing the database. The software system seamlessly controls the hardware platform, effectively synchronizing its operations.

In the subsequent phases of development, a new concept called the "three positions" is proposed for the coordinated movement of the manipulator. This innovation enhances the safety and reliability of the manipulator's movement process. Through these advancements, a comprehensive automated biological experiment platform software system has been meticulously established, and tailored to our specific requirements. This system design liberates users from the need to delve into intricate program logic; their focus remains on elucidating the experimental process. By simply dragging the necessary instruments in the intended sequence and configuring relevant parameters, the process editing is accomplished (as depicted in Fig. 1). Subsequently, hardware manipulation instigates the automation of the experimental procedure. This user-centric approach underscores the platform's user-friendliness and convenience.

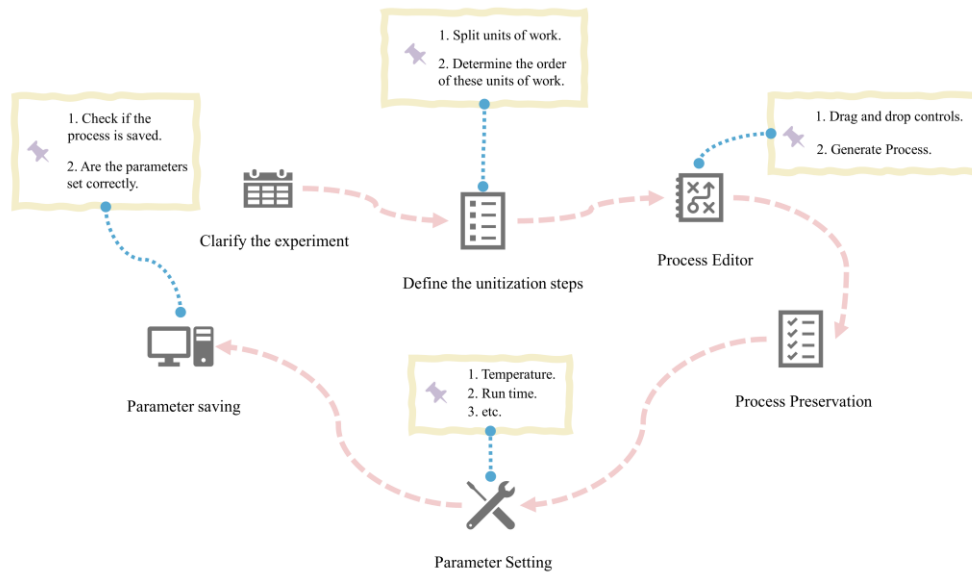


Fig. 1. Experimental flow editing process diagram.

## 2. Platform Overview

### 2.1. Hardware Overview

Within our platform, a meticulously assembled array of critical components constitutes its operational foundation. Notable among these elements is the central robot, specifically identified as the JAKA Zu7—an intelligent, lightweight, and collaborative robot equipped with six degrees of freedom. Complementing this is a 2.5-meter-long track supplied by AELTA, accompanied by a liquid handler known as the Tecan Freedom Evo, and a Biotek microplate reader. Augmenting these features is a cultivation module, an indispensable centrifuge sourced from Hettich, and a self-engineered spin stack that we have developed to meet our specific requirements. Of particular interest is the shaking culture module, a pivotal component of our setup. This module encompasses a shaking table, distinguished by its dual composition—comprising both self-developed fixed components and externally acquired parts. This shaking culture module also integrates a Small Robot (SCARA) and a self-constructed temporary storage transfer station, contributing to its multifaceted functionality.

The system architecture incorporates two pivotal robotic entities. The central robot orchestrates the harmonious arrangement and deployment of all equipment elements. Conversely, the SCARA robot assumes responsibility for the seamless movement and transportation of materials within the designated module. Each device fulfills distinct, autonomous roles within the realm of experimental unitized operations. Notable functions encompass pivotal tasks such as pipetting, dispensing, shaking, and culturing. An intriguing facet lies in the central robot's ability to interconnect these disparate unitized operations, thereby forming an integrated workflow module. A comprehensive aerial depiction of the platform's intricate hardware arrangement is meticulously presented in Fig. 2, encapsulating the essence of our experimental framework.

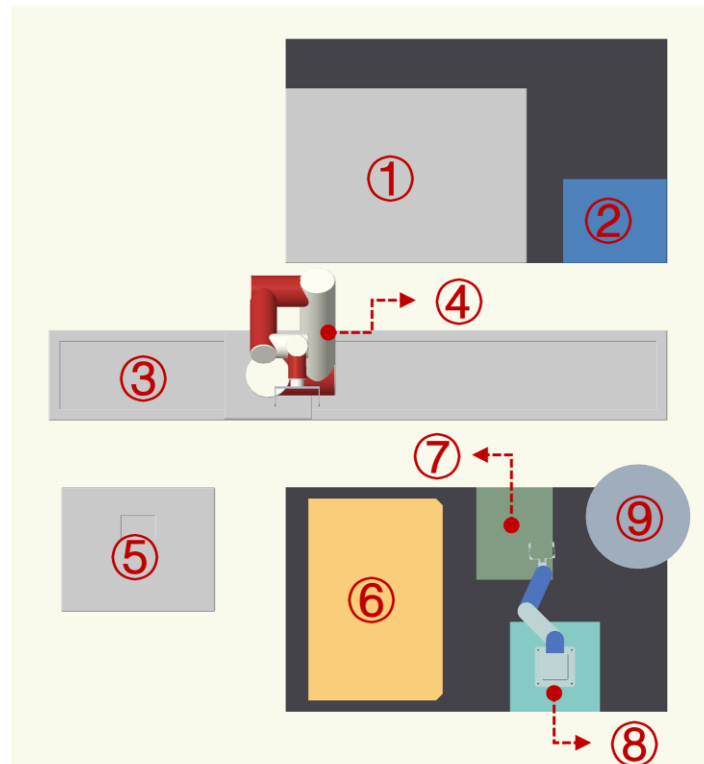


Fig. 2. Schematic diagram of the hardware layout of the platform. ①liquid handler, ②microplate reader, ③ track, ④JAKA robot, ⑤centrifuge, ⑥shaker, ⑦temporary secondary station, ⑧SCARA robot, ⑨stack.

## 2.2. Software Introduction

### 2.2.1. Overall system architecture

The system employs a robust three-tier architecture to achieve the principles of "high cohesion and low coupling"[26]. Within this architectural framework, each discrete module is meticulously designed to fulfill specific tasks with utmost autonomy, minimizing reliance on external code components. Simultaneously, the interconnections among modules are streamlined to ensure simplicity and efficiency. To manifest this architectural philosophy, the entire system is thoughtfully organized into distinct folders, thereby exemplifying the tenets of the three-tier structure. Notably, five key folders play pivotal roles in maintaining the structural integrity of the project: UI, BLL, DAL, Model, and Class. Within the UI folder, all system pages find their designated repository, while the BLL (Business Logic Layer) folder dutifully houses the components responsible for orchestrating and processing business logic submissions. In parallel, the DAL (Data Access Layer) folder stands as a repository for all SQL statements, encapsulating a comprehensive range of database operations encompassing additions, deletions, modifications, and queries. The Model folder diligently holds a repository of entity classes, each attribute intricately aligned with its corresponding database table, thereby maintaining an essential one-to-one correspondence. Finally, the Class file repository undertakes the responsibility of housing packaged or secondary-packaged classes, thereby consolidating the software architecture's cohesiveness.

This strategic division of project components into these well-defined folders substantiates the architectural design's underlying principles, optimizing both operational efficiency and code structure within the system.

### 2.2.2. Way of communication

The system integrates various devices and therefore requires an appropriate communication method based on the communication protocol of each device. Currently, the system employs various communication methods, including TCP/IP communication [27], Modbus communication [28], USB communication [29], UART communication [30], and custom communication methods.

Various communication methods are employed in the system, depending on the communication protocol of each device. For liquid handler and self-developed equipment, a custom communication protocol is utilized. The protocol is designed based on predetermined rules that guide the sending and receiving parties of the device to receive and analyze instructions. The protocol includes specifications for crucial aspects such as device name, command content, and check code.

In the process of engaging with the liquid handler, a master-slave host computer configuration is established to facilitate communication. This arrangement involves the utilization of a USB-to-serial cable and a serial cable to establish a communicative link between the two computers. Within this framework, the host computer assumes the role of an initiator, dispatching distinct commands in response to user input. On the other hand, the slave machine, having received these instructions, executes pre-configured scripts corresponding to the conveyed commands. These scripts, conceived through the proprietary software embedded within the liquid handler, allow for precise control over diverse experimental procedures. Concurrently, the host computer undertakes the responsibility of monitoring the receipt of messages from the device. This vigilance serves a dual purpose—capturing the device's status during task execution and ascertaining the successful completion of the assigned operation. It is pivotal to clarify that the "interface schematic diagram" showcased in Fig. 3 primarily functions as a tool for verifying the integrity of master-slave communication rather than constituting an essential component of the system's user interface. This delineation is important to prevent any misconceptions regarding the role and relevance of the depicted diagram.

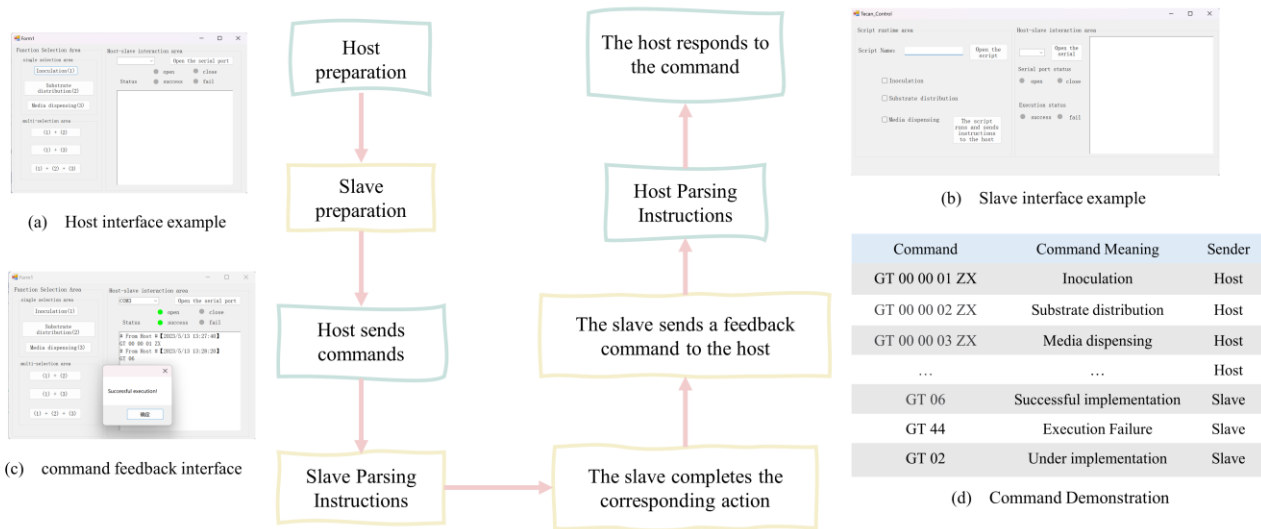


Fig. 3. Schematic diagram of master-slave upper computer communication workflow.

The system encompasses essential functional modules that cater to various critical aspects, including real-time monitoring, experiment process editing, experiment instrument parameter configuration, error alarm management, experiment log retrieval, and equipment single-step debugging, among others. Currently, the system has achieved effective communication with hardware devices, allowing for single-step debugging of said devices. Furthermore, it has successfully implemented precise plate transmission and proficiently undertaken the formulation and execution of experimental procedures. Part of the system page is shown in Fig. 4.

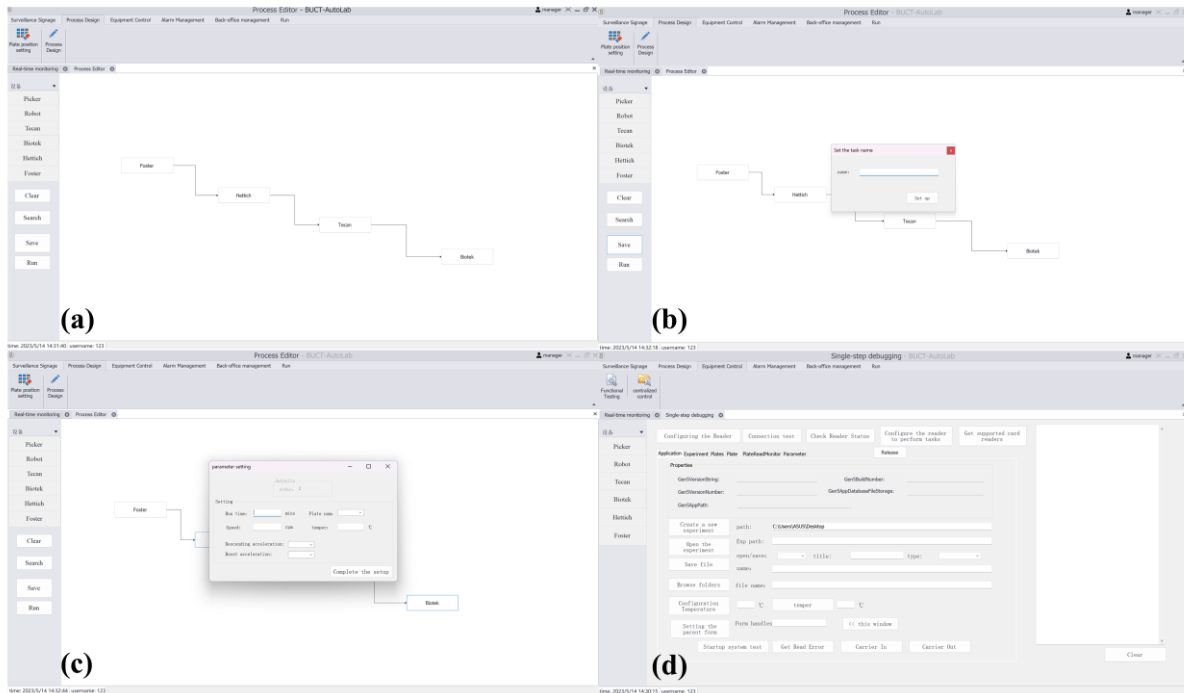


Fig. 4. Partial interface display of the system. (a) process editing interface, (b) experimental task naming interface, (c) parameter setting interface, (d) single-step debugging interface.

### 3. Experiment

Upon establishment of the system, the initial step involves the execution of a single-step debugging function test for the equipment. The process entails establishing communication channels using the prescribed communication method inherent to the hardware devices. Subsequently, each distinct function is transcribed into a code file, closely adhering to the communication protocol. These individualized functions are then amalgamated following successful singular execution, facilitating seamless integration within the program. This phase is pivotal in validating the discrete operations of each device. Preliminarily, the system device configures essential parameters, followed by meticulous observation of the device's performance to ascertain alignment with predetermined values. Upon successful completion of the single-step debugging function for the equipment, the subsequent phase involves initiating the debugging process for the central robot. The system orchestrates the precise positioning and grasping assessment of a singular instrument, achieved through the coordinated interplay of the central robot and the track. Initially, the robot is maneuvered along the track to approximate its position relative to the equipment's front. This position is then recorded within the database, subsequently serving as a reference point for further adjustments. After this rough positioning, meticulous alignment is achieved through the manipulation of the robot's six axes. This undertaking, though arduous, demands unwavering patience, necessitating repetitive calibration of each axis' angle to realize the desired precision in instrument placement.

Importantly, the robot's instrument positioning entails a three-fold categorization: the safe position, the lifted position, and the lowered position. Each of these stances mandates meticulous fine-tuning across all six axes, vital for averting any potential collisions. Upon satisfactory alignment, the refined six-axis angles are logged within the database for future reference.

Once the positioning sequence concludes, the system initiates the transmission of motion commands. This entails the robot mounted on the track navigating towards the designated target. Joint motion directives are transmitted, coordinating the robot's movements. The orchestrated sequence encompassing

safe positioning lifted, and lowered is executed in harmony with the grasping function, culminating in iterative debugging cycles. Ultimately, the outcome is the precise manipulation and placement of the plate—an exemplary portrayal of this process is illustrated in Fig. 5.

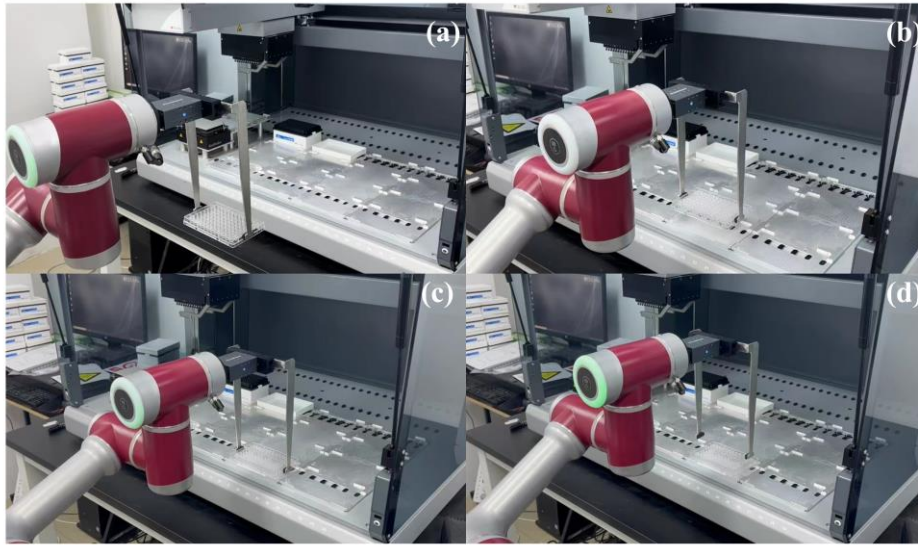


Fig. 5. Central robot positioning grab test diagram, (a) safe position, (b) lifted position, (c) lowered position, (d) gripper open and in lowered position.

Upon the successful completion of the single-step process test, the subsequent phase involves conducting a system process test designed to replicate authentic robot movements within the experimental context. Following the robot's successful material grasping action, it seamlessly transitions to the subsequent designated instrument point for plate transfer, thereby effectuating the pivotal positioning and transfer test—a transition from a single instrument to the seamless coordination of multiple instruments. Throughout this intricate process, the robot adeptly undertakes precise pick-and-place actions while sequentially transporting materials along a predefined program route. This dynamic orchestration is vividly illustrated in Fig. 6.

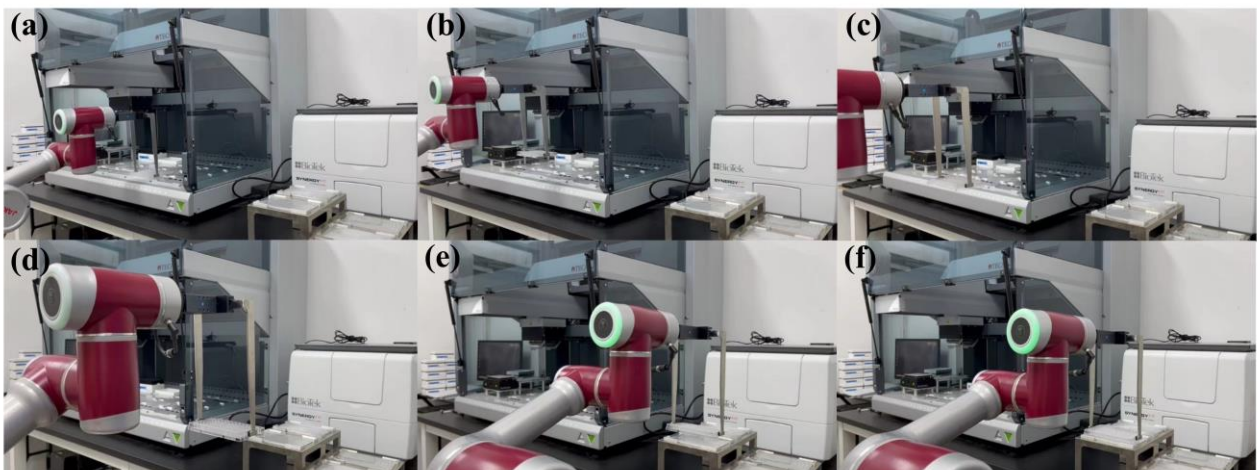


Fig. 6. Central robot process motion test. (a) take the plate from the liquid handler, (b) lifted position, (c) safe position, (d) move to next device, (e) lifted position, (f) lowered position.

To fortify the process's integrity, a program-wide closed-loop self-inspection mechanism is incorporated during the execution cycle. This intricate system scrutinizes each operational juncture against predefined

expectations. Any deviation from the expected outcomes triggers an immediate halt in progression to the subsequent step. This stringent self-regulation augments operational safety, mitigating the risk of undesirable outcomes—a concept effectively illustrated in Fig. 7.

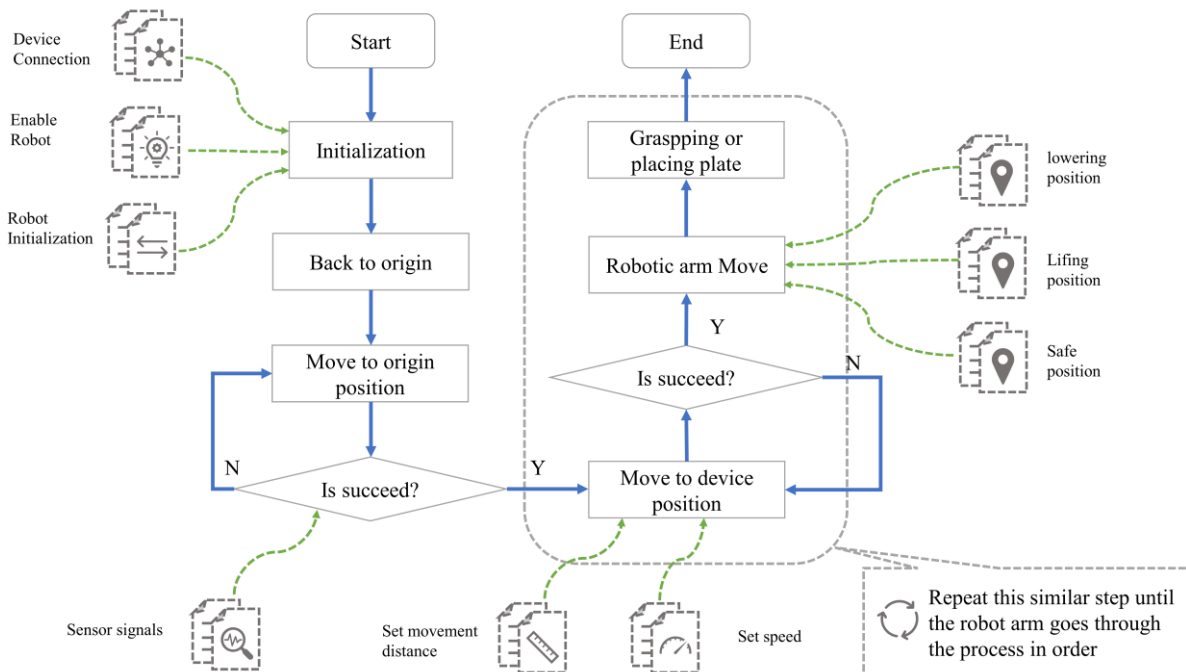


Fig. 7. Program logic of system flow control.

Furthermore, we conducted simulation experiments involving pipetting, substrate addition, and enzyme activity measurement using the included software of the liquid handler. To initiate the experiment, the initial step involved composing and saving the experiment script within the liquid handler's provided software. Subsequently, the system was employed to seamlessly arrange the essential equipment in a prescribed sequence within the user interface. Accompanying this, the relevant parameters were configured, and the experiment was ultimately initiated by selecting the "start" command, thereby effectuating the comprehensive automation of the entire process.

In this experimental setup, the liquid handler's robotic component undertakes a series of intricate actions. Initially, the robot deftly aspirates both tips and reagents, adeptly transferring them into the specified compartments within the 96-well plates. Upon the completion of these tasks, a prompt acknowledgment in the form of a "successful execution signal" is relayed to the system. Subsequently, the central robot embarks on a meticulously orchestrated journey along the designated track, drawing on a systematic sequence that adheres to the protocol of "three positions"—these encompass the safe position, the raised position, and the lowered position.

Upon reaching the designated location, the robot's gripper delicately retrieves the plate, demonstrating a seamless fusion of precision and controlled motion. Once the task is accomplished, the robot gracefully withdraws to a secure vantage point, thoughtfully maintaining an optimal distance from the liquid handler. This sequence of movements is then replicated to facilitate the strategic transition between the long and short sides of the plate within the confines of the temporary secondary station.

With meticulous execution, the orchestrated process culminates in the transportation of the plate to the microplate reader—a pivotal step that ushers in the analytical phase of the experiment. For a comprehensive visual depiction and understanding of this experimental intricacy, please refer to Fig. 8,



which encapsulates the key stages and nuances of this operation.

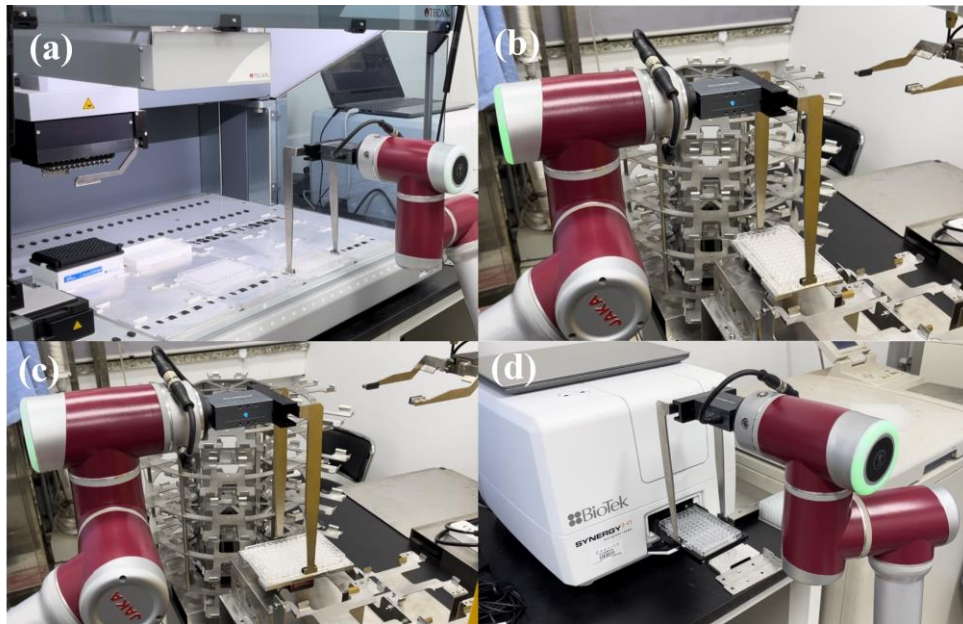


Fig. 8. Experimental process simulation. (a) take the plate from the liquid handler, (b) temporary storage transfer station put plate, (c) switch from long to short side, (d) plate to the microplate reader.

In the final phase, we conducted three sets of comparative experiments to assess the color reaction of glycolaldehyde and TTC. These experiments involved both automated and manual methodologies. The experimental timeline was carefully monitored, commencing from the initiation of pipetting and culminating in the precise placement of the plate onto the microplate reader.

Of particular interest was the noticeable discrepancy in the time required for the two approaches. The automated procedure demonstrated remarkable efficiency, completing an experiment within a mere 1 minute and 55 seconds. In contrast, the manual execution necessitated 2 minutes and 57 seconds—a discernible contrast highlighting the efficiency of automation.

Upon detailed examination of the microplate reader's results, a distinctive trend emerged. The data obtained from the automated system exhibited a significantly lower margin of error compared to the results generated through manual operation. This observation underscores the enhanced precision and data accuracy offered by the automated process.

For a visual representation of these findings, please refer to Fig. 9. These graphs provide a clear comparison, revealing the advantageous performance of the automated method in terms of data stability and reliability.

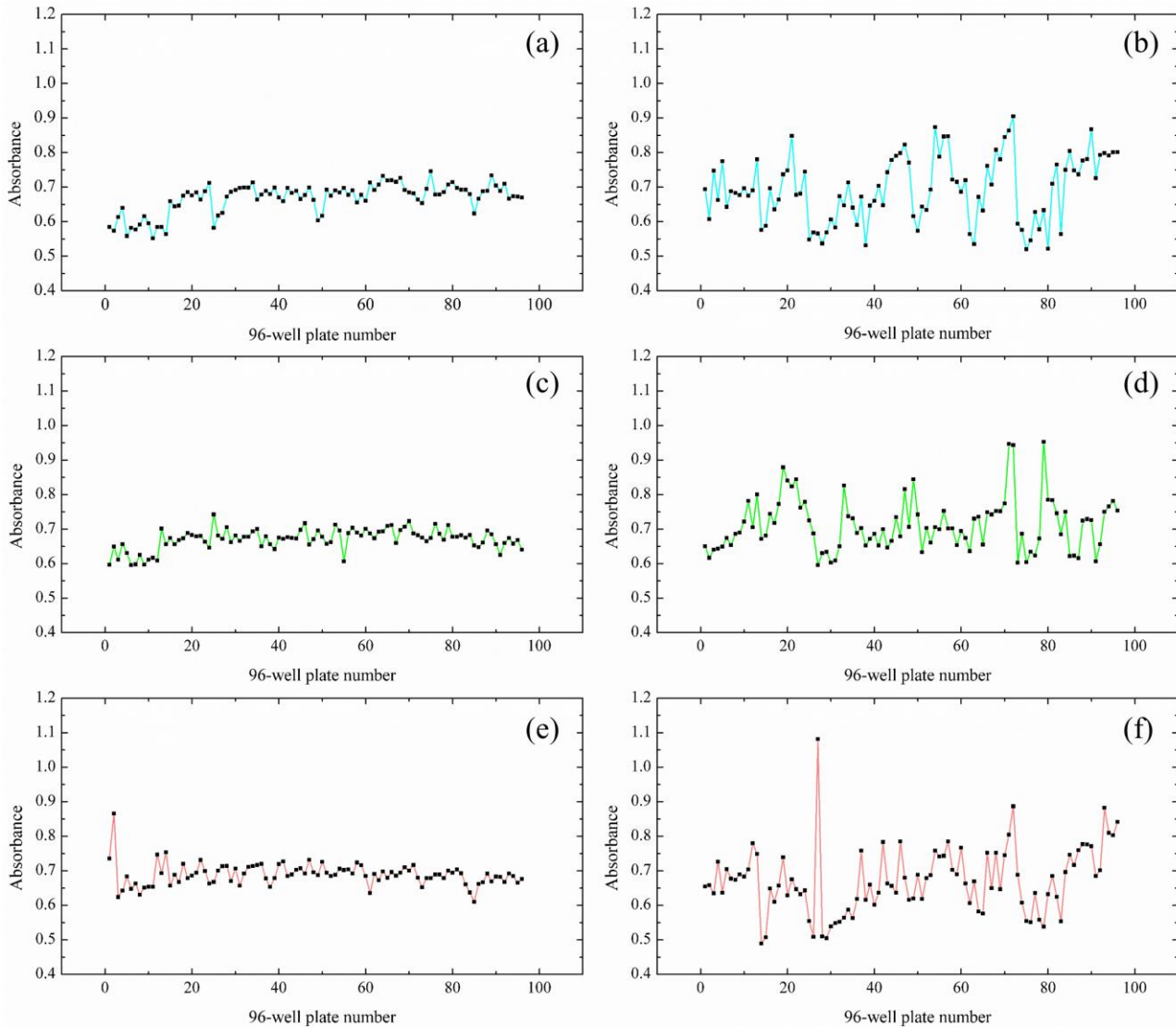


Fig. 9. Comparison of experimental results data. (a), (c), (e) are the measurement results of the automated platform, (b), (d), (f) are the measurement results of manual experiments under the same conditions.

#### 4. Conclusion

This platform has developed an automated integration system for specific experiments from scratch. The system uses a drag-and-drop interface to complete workflow editing. Users only need to drag and drop the required hardware during the experiment, and the system interface will automatically generate connections and sort out the driving route of the central robot. The communication protocol and signal processing complete the automation of the experimental process. The interface has developed different permissions for the two identities of experimenters and administrators, which ensures the security of the system. A testament to its efficiency, the amalgamation of hardware integration and autonomous control results in the significant reduction of experimental time and the concurrent enhancement of experimental efficiency. Simultaneously, the platform averts the inherent errors associated with manual intervention, thereby augmenting the overall reliability of experimental outcomes.

Anticipating future developments, the strategic optimization of hardware layout emerges as a priority to minimize superfluous movements of the robot arm and consequently economize time. Additionally, the consideration of intricate task scheduling within process-based experiments emerges as pivotal in

endowing the platform with heightened levels of intelligence and operational efficacy.

### Conflict of Interest

The authors declare no conflict of interest.

### Author Contributions

T.T.: Conceptualization, Methodology; Y.C.: Software, Formal analysis, Data Curation, Writing - Original Draft; X.T. and W.Z.: Supervision; Y.S. J.Z. and Y.L.: Investigation; all authors had approved the final version.

### Acknowledgment

This work was supported by National Key R & D Program of China (grant no. 2021YFC2103100).

### References

- [1] Bai, J., Cao, L., Mosbach, S., Akroyd, J., Lapkin, A. A., & Kraft, M. J. J. A. (2022). From platform to knowledge graph: Evolution of laboratory automation. *2(2)*, 292–309.
- [2] Boyd, J. J. S. (2002). Robotic laboratory automation. *295(5554)*, 517–518.
- [3] Boyle, P. J. (2019). Biology by Design: Organism engineering at ginkgo bioworks. 13–13.
- [4] Chao, R., Liang, J., Tasan, I., Si, T., Ju, L., & Zhao, H. (2017). Fully automated one-step synthesis of single-transcript talen Pairs using a biological foundry. *ACS Synth Biol*, *6(4)*, 678–685.
- [5] Chao, R., Mishra, S., Si, T., & Zhao, H. J. M. E. (2017). Engineering biological systems using automated biofoundries. *42*, 98–108.
- [6] Chao, R., Yuan, Y., & Zhao, H. (2015). Building biological foundries for next-generation synthetic biology. *Sci China Life Sci.*, *58(7)*, 658–665.
- [7] Coli, B. F. E. (2010). Ginkgo bioworks. Retrieved from: <https://www.osti.gov/biblio/1046606>
- [8] Di, L. A., Berchi, R., Pescarmona, G., & Sonnessa, M. J. I. J. o. i. t. (2005). analysis and prototyping of biological systems: The abstract biological process model. *2*, 216–224.
- [9] Dörr, M. *et al.*, (2016). Fully automatized high-throughput enzyme library screening using a robotic platform. *Biotechnology and Bioengineering*, *113(7)*, 1421–1432.
- [10] Edelman, G. M., & Gally, J. A. J. P. o. t. N. A. o. S. (2001). Degeneracy and complexity in biological systems. *98(24)*, 13763–13768.
- [11] Forouzan, B. A. (2002). TCP/IP protocol suite: McGraw-Hill Higher Education.
- [12] Genzen, J. R., Burnham, C. A. D., Felder, R. A., Hawker, C. D., Lippi, G., & Peck Palmer, O. M. J. C. C. (2018). Challenges and opportunities in implementing total laboratory automation. *64(2)*, 259–264.
- [13] Han, Y. (2011). Bioworks: A workflow system for automation of bioinformatics analysis processes. *Proceedings of the 2011 International Conference on Ubiquitous Computing and Multimedia Applications*.
- [14] Hayden, E. C. J. N. (2014). Synthetic-biology firms shift focus. *505(7485)*, 598–598.
- [15] Helal, S., Hammer, J., Zhang, J., & Khushraj, A. (2001). A three-tier architecture for ubiquitous data access. *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications*.
- [16] Hillson, N. *et al.*, (2019). Building a global alliance of biofoundries. *10(1)*, 2040.
- [17] Holland, I., & Davies, J. A. (2020). Automation in the life science research laboratory. *Front Bioeng Biotechnol*, *8*, 571777.
- [18] Łacki, K. M. J. C. O. I. C. E. (2014). High throughput process development in biomanufacturing. *6*, 25–32.
- [19] Laddha, N. R., Thakare, A. J. I. j. O. A. R. I. C. S., & engineering, S. (2013). A review on serial communication by UART. *3(1)*.
- [20] Lippi, G., Da Rin, G. J. C. C., & Medicine, L. (2019). Advantages and limitations of total laboratory

automation: A personal overview. *57(6)*, 802–811.

- [21] Lutz, M. H., & Laplante, P. A. J. I. S. (2003). C# and the .NET framework: Ready for real time? *20(1)*, 74–80.
- [22] MacDonald, M. (2008). *User Interfaces in C#: Windows forms and custom controls*: Apress.
- [23] Markin, R. S., & Whalen, S. A. J. C. C. (2000). Laboratory automation: trajectory, technology, and tactics. *46(5)*, 764–771.
- [24] Mistry, R., & Misner, S. (2014). *Introducing Microsoft SQL Server 2014*, Microsoft Press.
- [25] Nissim, N., Yahalom, R., Elovici, Y. J. C., & Security. (2017). USB-based attacks. *70*, 675–688.
- [26] Scown, C. D. (2022). Prospects for carbon-negative biomanufacturing. *Trends in Biotechnology*, *40(12)*, 1415–1424. doi:10.1016/j.tibtech.2022.09.004.
- [27] Sibi, G. (2022). *Environmental Biotechnology: Fundamentals to Modern Techniques*, CRC Press.
- [28] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, *14(11)*, 4724–4734.
- [29] Thomas, G. J. T. E. (2008). Introduction to the modbus protocol. *9(4)*, 1–4.
- [30] Xie, D. M. (2022). Continuous biomanufacturing with microbes – Upstream progresses and challenges. *Current Opinion in Biotechnology*, *78*, 11.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))