The low code development platform can be used for solar cell traceability

Gang Liu*, Meng Xie

Shanghai Glorysoft Software Co. ShangHai ShangHai 230031, China
Email: sdliugang136@163.com

Abstract. In the paper, innovative approaches to virtual wafer-code traceability for the production of solar cells are explored on a low-code platform for monolithic traceability. A brief overview of the background and significance of solar monolithic research and its original traceability is given first, followed by a detailed discussion of the advantages of using a low-code platform for a range of applications. Finally, the specific implementation steps and application scenarios are described in detail, followed by an analysis of the logic of the monolithic traceability design using low-code on the development platform, including the module algorithms used as well as its design scheme design and software design. Finally, the feasibility of the method is demonstrated through implementation and strong support is provided for its use in practice with an accuracy rate of over 98%.

Keywords: monolithic traceability, low code, virtual coding, solar cell production

1 Introduction

In traditional solar cell production, there are two ways of managing cell production. One is batch traceability and the other is single wafer traceability [10]. Batch refers to the size of the batch being defined prior to production using 10 carriers per batch, or traceability of the carriers, which contain RFID chips [11]. The traditional method of single wafer traceability involves engraving the solar wafers to ensure a unique wafer code over a period of time, which requires the wafers to be engraved prior to production and the addition of equipment for initial processing of the product. In addition, the use of manual production makes it inefficient and prone to error. In this paper, we explore the application of virtual coding of solar cells to production, which enables single-chip traceability of equipment through a low-code platform for logical calculations, thereby improving production efficiency and quality [1].

Low-code development platform has been in the limelight since 2018 when Siemens purchased the Medix low-code development platform[7], which has received increasing attention as it has evolved in recent years. IBM defined the low-code development platform in 2021 and stated that low-code is about faster delivery of applications through minimal manual coding, and that low-code platforms graphical and drag-and-drop

A. Rauf et al. (eds.), *Proceedings of the 3rd International Conference on Management Science and Software Engineering (ICMSSE 2023)*, Atlantis Highlights in Engineering 20,
https://doi.org/10.2991/978-94-6463-262-0_19

capabilities automate all aspects of the development process and eliminate the reliance on traditional computer programming methods [2], and in 2021 Bock Alexander C.; Frank Ulrich et al. Redefined what characteristics low-code development platforms have, what advantages low-code platforms have over the current state of research, what those platforms have achieved in terms of technological innovations, and what opportunities the current focus on low-code development issues offers for the future [3]. Now .there is growing awareness of low-code in this research community. For example, the workshop on low-code development at the Models conference in 2020 attracted the highest number of submiss of all workshops[8]. At the same time, low-code platforms are being actively applied in China, from the application development in the development of contract systems [4] to the practice in the digital transformation of Baiyun Airport [5] and the application development in the study of the application in the digital transformation of enterprises [6], low-code development platforms are favored by more and more enterprises. Financial industry applies low code to get services "in cloud" [9].The low-code development platforms described above are all applied in management. In this presentation the main focus is on manufacturing for product process management. Chapter 1 will describe the monolithic retrospective construction in low code; Chapter 2 describes the logic implementation of the low code development platform; Chapter 3 concludes.

2 Low-code development platform monolithic retrospective build

Shanghai Glorysoft Co., Ltd. (hereinafter referred to as Glory) has developed a low-code development platform based on its own digital products such as MES, EAP, ERP and APS, which have been successfully applied in the solar cell traceability project.

The architecture of the low-code development platform developed by Glory is shown in Figure 1.

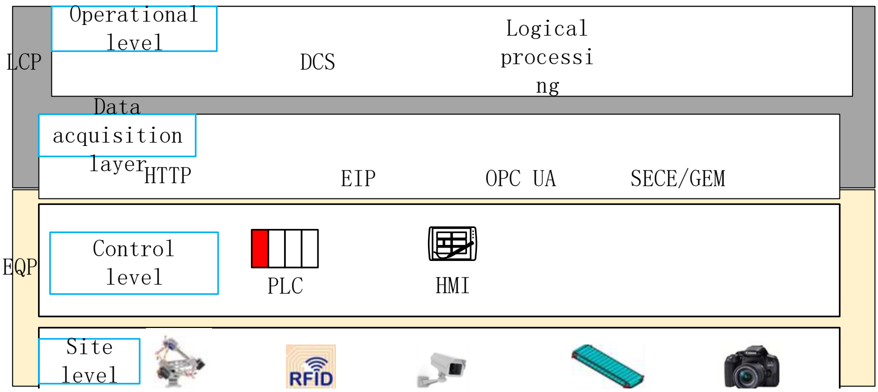


Fig. 1. Low-code development platform in the solar monolithic back-end application architecture

On the basis of the data acquisition, equipment communication, data collection, data processing and data entry are integrated; the quipment and the system use the PLC protocol to exchange data with the equipment; the data is collected according to the pre-defined PLC address; after the equipment data collection is completed, it is entered into the LCP platform and the logic processing work is performed on the LCP platform, which will be explained in later chapters; after the logic processing is completed The advantage of this design is that the rapid production of solar cells generates a large amount of data every day, which needs to be separated from production data and reporting data for better presentation to users in the future.

2.1 Cleaning and texture business logic

The logic for cleaning the fleece to generate the wafer code is shown in Figure 2

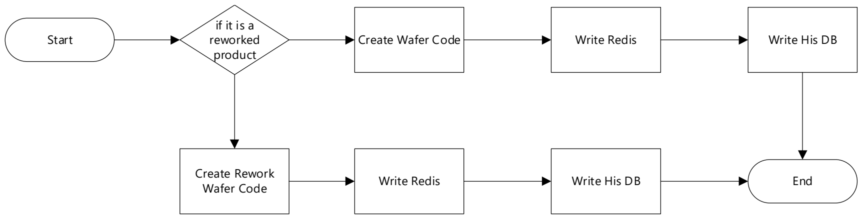


Fig. 2. Cleaning and texture the create wafer coding logic

The first stage of cell create requires the generation of a wafer virtual code, where the equipment reports to the system that the wafers are filled with carriers, the system receives the data reported by the equipment to determine the type of wafer to generate different wafer codes and records the positioning information of the wafers used on the carriers according to the generated wafer codes [1].

The most critical part of this process is the create wafer information, and depending on the incoming information will be created according to different rules, as shown in the Figure 3 diagram below, which are the rules for creating the wafer codes.

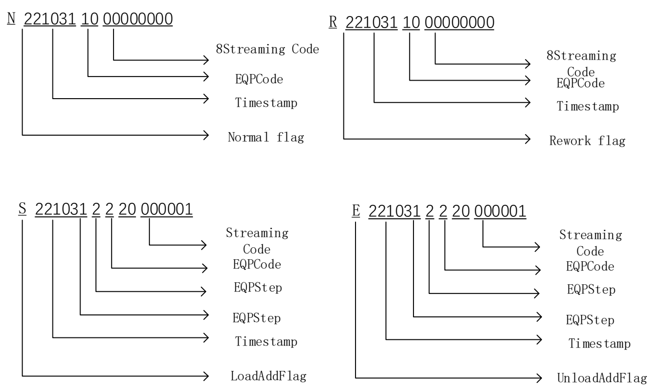


Fig. 3. Cleaning and texture to create wafer coding rules

The solar cell equipment uses a PLC controller to control the operation of the equipment. The memory space of the PLC is limited. In order to be able to transfer such a large amount of data, there can only be UDINT data in the PLC. Therefore, the code set here is a 10-bit UDINT code. The 10-bit code also takes into account the fact that a process section may need to make up wafers when there is no wafer ID in the production process, so the highest bit of the code generated is the normal code starting with 1 and the one starting with 2 is the complementary code.

2.2 Logic of loading and unloading

Single wafer traceability mainly refers to the process of loading and unloading can accurately get the virtual number of silicon wafers and can correspond to the corresponding physical, this paper will be a unified explanation of the loading and unloading logic, where the loading logic as shown in Figure 4 and the lower material as shown in Figure 5

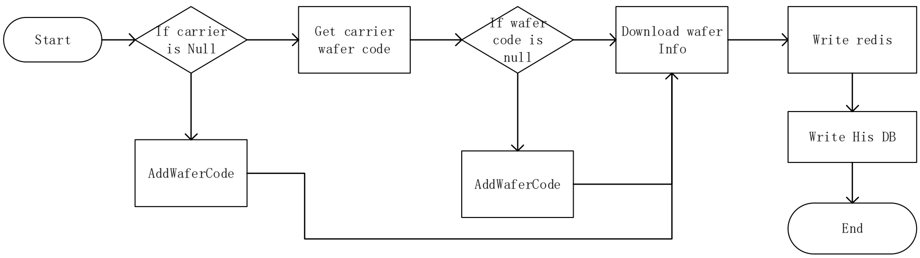


Fig. 4. Loading logic

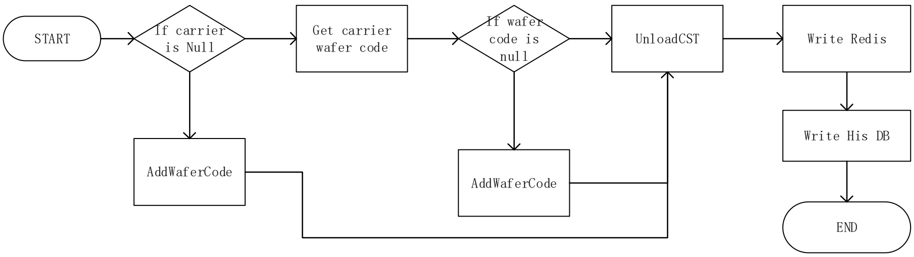


Fig. 5. Unloading logic

During the loading process, the device is first judged whether the carrier code is reported, based on the carrier code the wafer information in the carrier is searched, the wafer information is sent down to the device, the data is unbatched and bound to the device, the over station information in the history database is bound, if the carrier code does not exist then the wafer is generated again, the generation rules are described in the above rules for patching. When the carrier information is present but no carrier wafer information is found when searching the data in the lookup database, the wafer code is generated again, the rules for which have been described in the previous section, and then sent to the device after it has been generated and the device starts processing.

During the loading process, the carrier information and the wafer information within the carrier are reported by the equipment. The reported wafer information is matched with the corresponding system code through the database, and if there is a match, it is directly entered into the database; if the match is unsuccessful, the system will make up the wafer according to the number of missing wafer codes, and write the data to the database.

3 Low code development platform logic implementation

3.1 Low-code implementation of cleaning and texture

The low-code implementation of cleaning and texture is shown in Figure 6

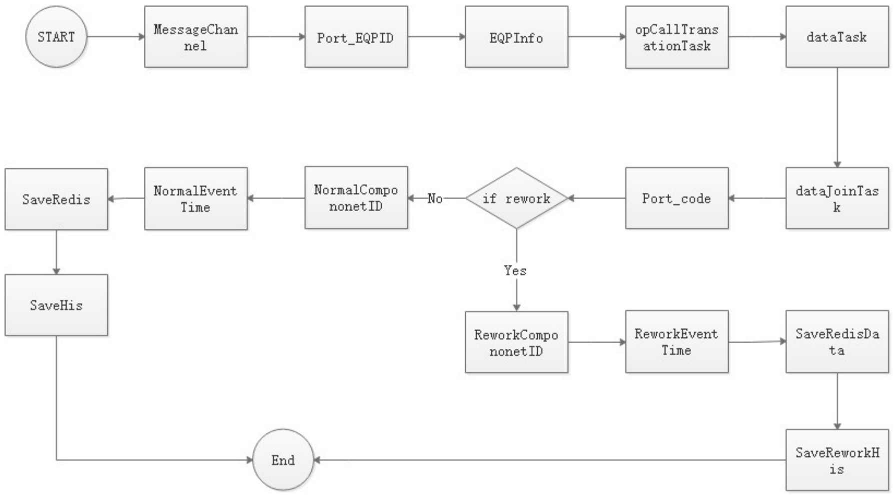


Fig. 6. Create the Wafer code for loading complete

The full process described in the previous section focuses on creating wafer codes, and the following section describes the low code to create wafer codes.

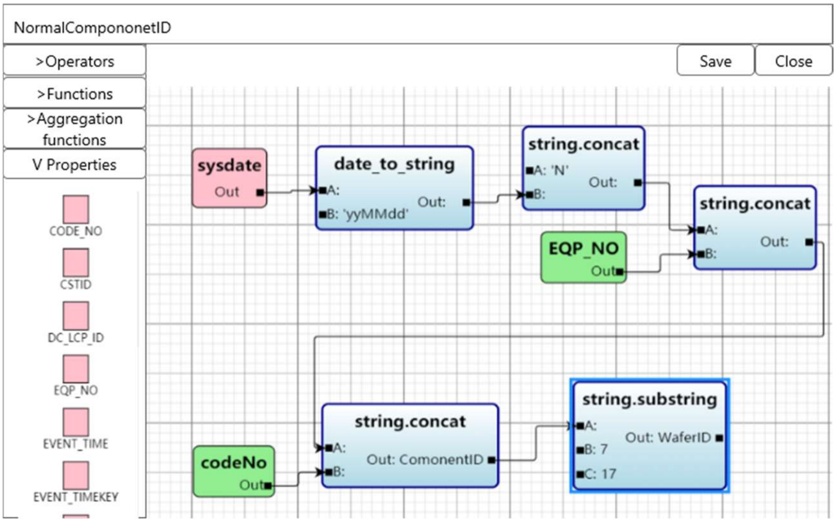
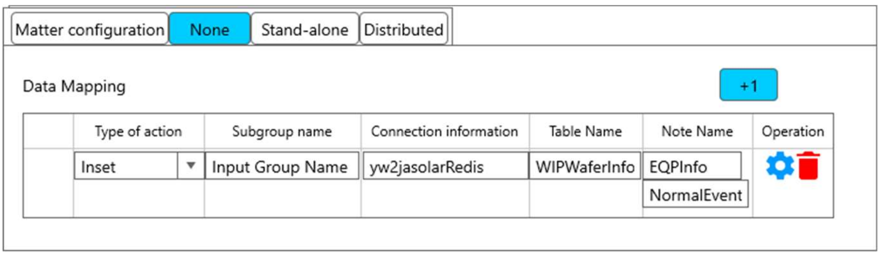


Fig. 7. Low-code platefrom create the wafer ID

Figure 7 represents the flow of in silico coding implemented in a low code platform. The input here is the device information and the serial number, the system time is generated by the low-code time module and the timestamp is converted to the corresponding format "yyMMdd" according to date_to_string. Here the normal silicon code is shown, so the first letter of the silicon is N. Here, the timestamp and the letter need to be combined together, so the low-code module string.concat is used to stitch the strings together. After splicing the device number with its splice it is easy to find the device in which device which off-loading port generated the wafer information, when the device number splice is complete the production line number with its splice here still use the string.concat module, so that the system is completed using the virtual number. Due to the storage of the device PLC, the number needs to be changed to a production number that the device can recognize, here it is a 10 bit code so the corresponding data needs to be intercepted by a module on a low code platform, here the string. substring module is used for processing.

After the wafer has been coded, it will be processed into a library (written to a database), where it will be written to either a redis library or a history library, depending on the business requirements. The history library is used for future data analysis, The detailed configuration process is shown in Figure 8.



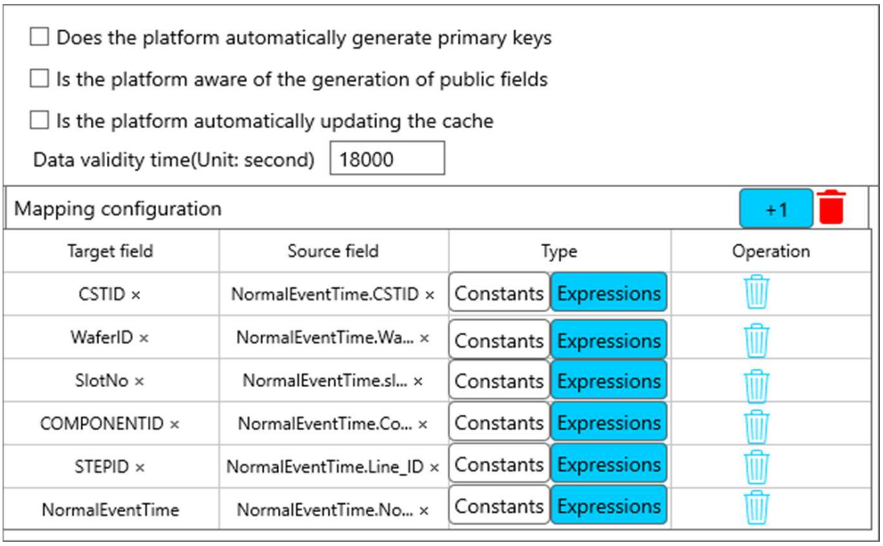


Fig. 8. Data insert into DB

The fields in the database table are selected from the left and synthesized from the data generated by the low-code platform on the right. The entry parameters are shown in the diagram above for selection, and all data is obtained from the nodes of the process, and then written to the database by matching the corresponding fields to the pattern selected on the right.

3.2 Low code development flatform loading business logic

If there is no carrier number, the wafer will be made up directly; if there is a carrier number, the wafer information in the carrier will be retrieved from the WIP library. If the wafer information in the carrier is not retrieved successfully, then it is necessary to generate the wafer information for replenishment, and if the wafer information exists, it will be directly sent to the equipment. The more important process nodes in the whole business process are the data filtering to get the latest wafer code and the wafer information distribution function.

To obtain the latest wafer code is to report the carrier number on the device and find the corresponding wafer code in the WIP, which is needed to avoid duplication of data and to extract the latest data to send to the device, as shown in Figure 9

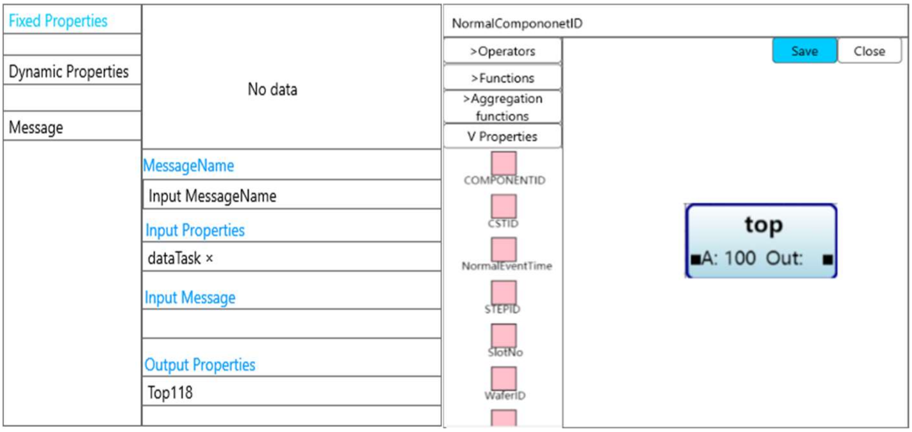


Fig. 9. Data filtering

The module is able to obtain the latest data based on the latest values entered. The module is developed using the Glory platform using "MessageTask" module and has the following main configurations

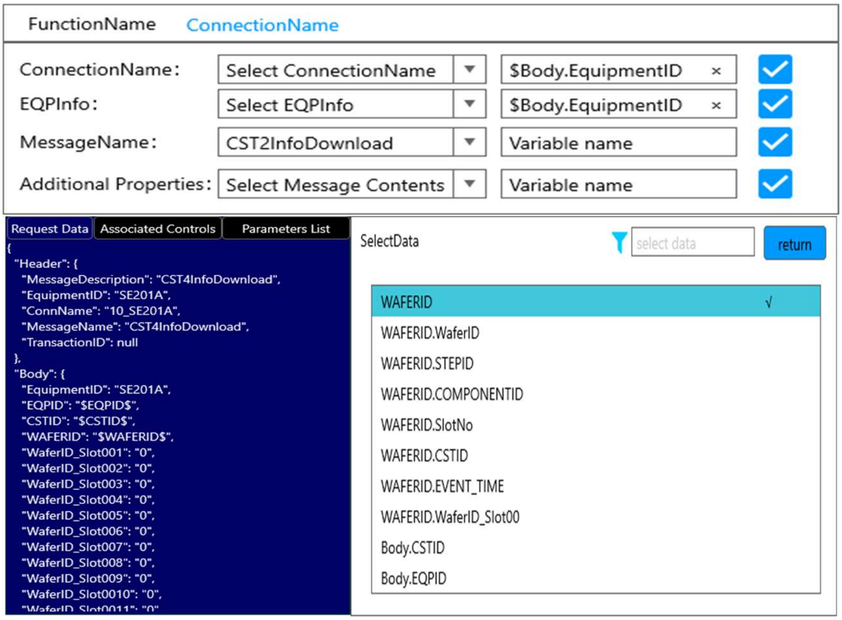


Fig. 10. Data Download

As can be seen through Figure 10 first select the configured equipment package, select the corresponding equipment number according to the equipment package. This equipment number can be selected through data processing. Once the device number is complete, the Key is selected according to the json of the data to be sent, where the data can be selected through a low-code development platform. Based on the above process and the whole process logic, the data obtained is passed to the MessageTask to send the data to the device. The entire business process is shown in Figure 11

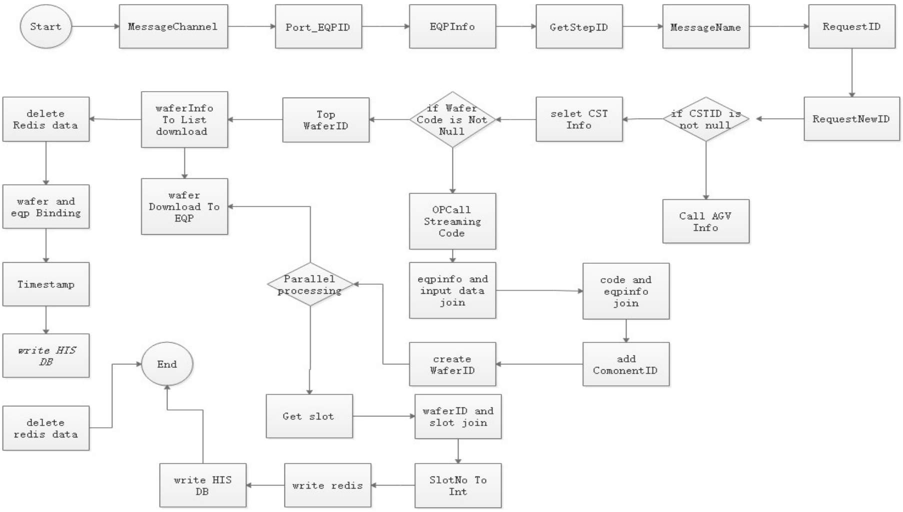


Fig. 11. Equipment loading process

3.3 Low-code platform Unloading operations logic

Once the wafer information is received, the system converts the reported wafer information into information that can be processed by the system. This information is used to search the database for the corresponding wafer code and to find the wafer code based on the wafer information data reported by the equipment. If the data found does not match the actual data of the carrier, the missing data will need to be compensated for. The exact process is shown in Figure 12.

In this process, the main process is to pre-process the information reported by the device, find out whether the reported data matches the system's data and the number of matches.

The pre-processing array is converted to JSON format and the device will be reported as value and name data. Here the value will be converted to WaferID and the name to SlotNo, so that during configuration the output name of the value is chosen to be WaferID and the output name to be SlotNo, and the output data will be written to a list, defined here as WaferInfoList, and the output will be set to List so that the data reported by the device will be converted to JSON for subsequent use. This will convert the data reported by the device to JSON for subsequent use.

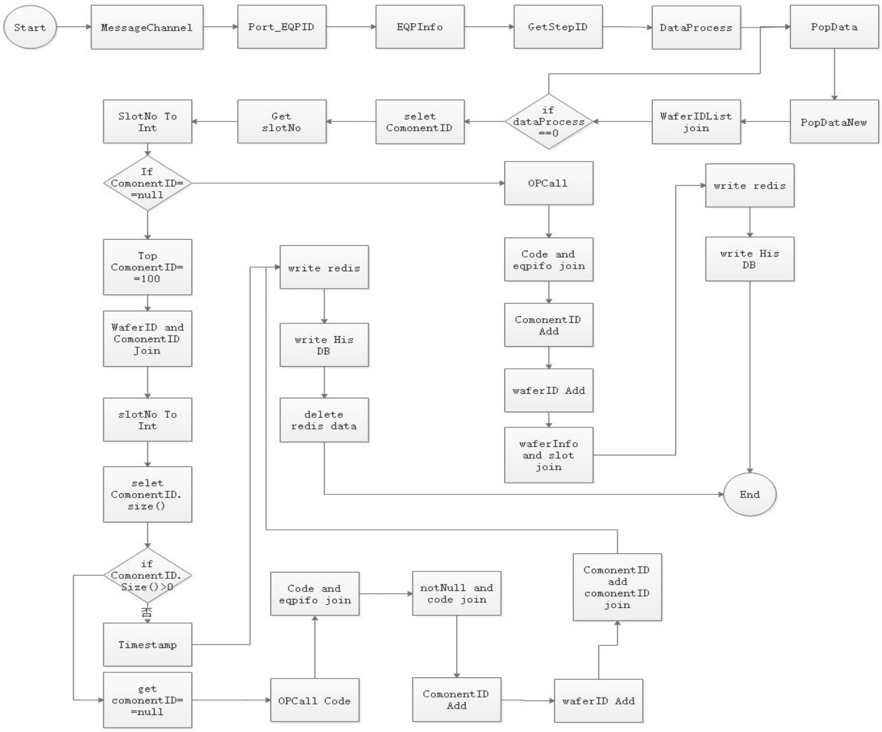


Fig. 12. Equipment UnLoading Process

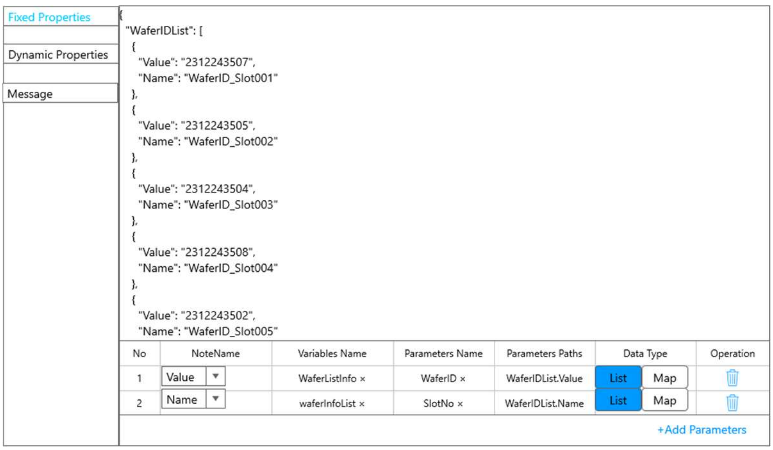


Fig. 13. Data pre-processing

Matching number processing of reported data and system data. The data processing is implemented in a low-code development platform as shown in Figure 13.

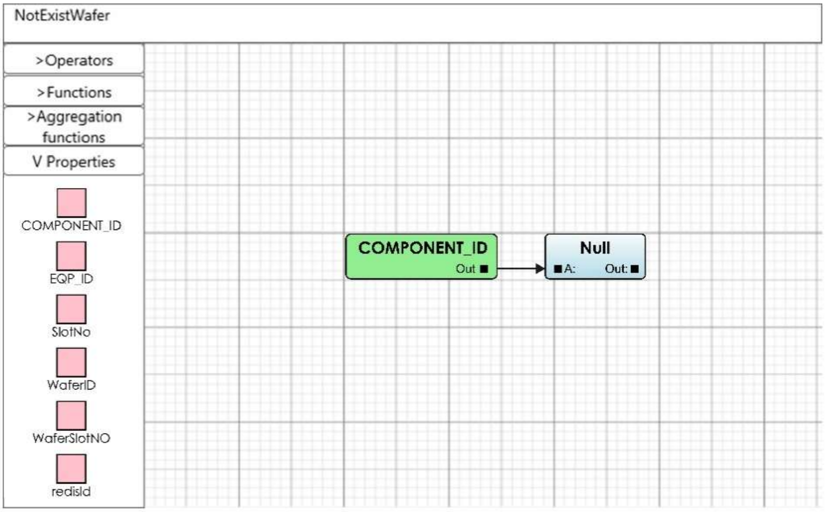


Fig. 14. Data filtering

Figure 14 represents the low code platform implementing data filtering to find the number of silicon wafers coded as null. Based on which the module can be used to generate a patching pipeline code and generate a patching wafer based on the patching rules, the specific generation rules and generation methods have been described previously and are not detailed here.

4 Summary

Compared to traditional software development tools, the use of low-code development platforms for single cell traceability of solar cells reveals that only a small amount of code or no code is required to achieve single cell traceability of solar cells at the same time. This greatly simplifies the application development timeline and does not impact on application performance. It also increases the efficiency of application development and significantly reduces development costs. The Glory low-code development platform also allows the establishment of an industrial ecology that enables the user departments or customers to achieve the information and digital transformation of the company.

Author's note:

Liu Gang (1986-), male, Shandong Jining, intermediate engineer, mas-ter's degree, research direction: enterprise digital construction, solar plant intelligence, artificial intelligence, email: sdliugang136 @163. com.

References:

1. Liu Gang, Lei Xueshuang, Sheng Zou, et al. Study on data acquisition system for solar energy (heterozygous) cell sheet production [J]. Intelligent Manufacturing, 2023 (2): 56-61.
2. https://www.ibm.com/uk-en/automation/low-code. Accessed 12 Sep 2021
3. Bock Alexander C., Frank Ulrich. Low-Code Platform[J]. Business & Information Systems Engineering,2021,63 (6).
4. Tianyuan Wang, Richard Zhang. Development of contract management system based on low-code platform [J]. Computer Knowledge and Technology, 2022, 18 (26): 38-41.
5. Musun, Ding Yan, Li Jun, et al. Practice of Low Code Rapid Development Platform in Digital Transformation of Baiyun Airport [J]. Software Development and Applications, 2022.
6. Feng Chul, Zhang Qiang. Application of low code development platform in enterprise digital transformation [J]. Internet Weekly, 2022.
7. Siemens launches Xcelerator portfolio [J]. Smart Manufacturing,2019(10):9.
8. https://lowcode-workshop.github.io/. Accessed Sep 12 2021.
9. Chen Xi, Yang Mian, Huang Yan. Exploration and practice of enterprise-level low-code platform of China Merchants Bank[J]. China Financial Computer,2023(04):53-57.
10. Wuxi Weinte Data Technology Co. Traceability method, device and storage medium for solar cells:CN202111297989.3[P]. 2022-01-21.
11. ifm efma electronics. Second bullet: Track and trace in solar cell production [EB/OL]. [2022-05-10]. https://sensor.ofweek.com/2022-05/ART-81000-8100-30560319.html.