DOI: 10.23977/curtm.2025.080419
ISSN 2616-2261 Vol. 8 Num. 4

The Application of 'Ability Map + AIGC' in Programming Course Teaching: A Case Study of 'Principle and Application of Microcontroller'

Lu Lijun

School of Mechanical and Automation, Wuchang Shouyi College, Wuhan, 430064, Hubei, China

Keywords: Ability Map; AIGC; Programming; Course Teaching

Abstract: With the powerful application capabilities of generative artificial intelligence in programming, new requirements for teaching implementation in university programming courses have emerged. This paper first constructs a capability map that transforms enterprise needs into quantifiable abilities through structured analysis based on job competencies. Then, it proposes a three-tier AIGC teaching method based on the automatic code generation application of AIGC. Finally, it innovatively combines the capability map with AIGC to design a dual-driven teaching model of "capability map + AIGC," and verifies the effectiveness of this developed model for programming courses using the programming course "Principles and Applications of Microcontrollers" as an example.

1. Introduction

In the context of digital transformation and the development of new productive forces, the powerful general capabilities of AIGC have sparked fears of an unemployment wave, particularly in programming roles. The International Labor Organization believes that "whether workers are eliminated or empowered in the AI era largely depends on whether they have received professional training to use AI technology" [1]. This places higher demands on the comprehensive abilities of professionals. In traditional teaching models, programming courses suffer from fragmented knowledge points, practical scenarios that do not align with corporate needs, and low efficiency in code debugging [2]. With breakthroughs in generative artificial intelligence (AIGC) technology, its advantages in automated code generation, logic validation, and cross-modal resource integration offer new solutions to these challenges. However, existing research often focuses on theoretical discussions of AIGC in general education settings [2-3] or the construction of single competency maps [5], lacking systematic teaching practices that deeply integrate corporate job requirements with intelligent generation tools.

This study, with the core concept of "from enterprise to enterprise," innovatively combines job competency maps with AI-generated content. It aims to reconstruct the teaching content and practical teaching process of programming courses through the collaborative innovation of competency maps and AI-generated content. This effort seeks to deepen the alignment between higher education and industry needs, exploring new paradigms for cultivating new productive forces.

2. Construction of capability map based on job data

A competency map is a systematic and structured tool used to describe and analyze the knowledge graph of required competency elements and their relationships in specific fields or positions [6]. A competency map based on job data achieves precise alignment between the education chain and the industry chain by deeply analyzing core competency elements of jobs and reconstructing the curriculum capability system. It aims to address the pain point of "disconnection between learning and application" in traditional higher education, enhancing the adaptability and effectiveness of talent cultivation.

The process of constructing a job competency map is shown in Figure 1. First, job data preparation involves obtaining job data through recruitment platform data mining, company job descriptions, and skill requirement frequency analysis. Next, job data cleaning is performed, where the acquired job data is parsed, primarily through text recognition, paragraph segmentation, and in-depth information source exploration. Then, core skill items are extracted, with data elements (text, images, tables) being transformed into skill points. Finally, industry experts analyze these skill points, organizing them comprehensively in a modular manner to form a three-tier capability map: foundational level, advanced level, and innovative level [7].

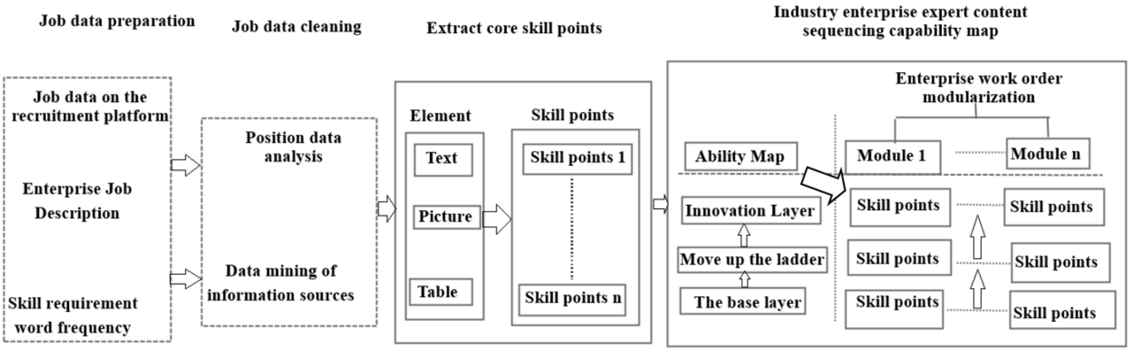

Figure 1: Job competency map construction process

3. Three-level automatic code generation based on AIGC technology

AIGC (Artificial Intelligence Generated Content) is a generative method based on artificial intelligence technology. By learning and training from large amounts of data, it can discover patterns and automatically generate content with certain quality and creativity, such as text, images, videos, and code . The application of AIGC in program code generation is mainly reflected in the following aspects: (1) Code auto-generation, generating complete code snippets or functions based on natural language descriptions or simple comments; (2) Code completion and suggestions, providing real-time code completion suggestions in IDEs to reduce manual input; (3) Test case generation, automatically generating unit test and integration test code to increase test coverage; (4) Code optimization and refactoring, analyzing existing code and proposing performance optimization or code structure improvement suggestions; (5) Intelligent debugging and error fixing, analyzing code errors and providing repair suggestions, or automatically fixing common vulnerabilities [8]. Generating code through AIGC helps programmers improve development efficiency, optimize code quality, reduce error and debugging time, enhance learning and self-iteration capabilities, and adapt to various programming languages and scenarios.

Based on the AIGC code generation application and its advantages, a code generation model for programming courses empowered by AIGC has been constructed. According to the difficulty of debugging automatically generated code by AIGC, it is divided into three levels: basic, intermediate, and advanced. All applications at these three levels require a certain amount of professional knowledge to construct prompt words for program code generation. The large model generates code automatically based on the input prompt words, with time complexity, space complexity, correctness, etc., needing to be verified and optimized through corresponding software.

The foundational layer involves generating code snippets or functions from natural language descriptions or annotations. Students input prompt words specific to their professional knowledge domain, and AIGC automatically generates corresponding code snippets or functions. Two or more code snippets can form a debuggable program segment, which is then verified for correctness using keil5+proteus. If issues arise, they identify the problems based on debugging phenomena and either correct them themselves or further engage AIGC for optimization. This method gradually generates a smaller amount of code, making it easier to pinpoint errors and reducing mechanical coding time, allowing students to focus on logical design. Multiple program segments are logically combined into complete code, which is finally verified and optimized again.

The advanced layer generates complete code from multimodal inputs. Students provide multimodal inputs (text, images, voice), and AIGC automatically generates all the code based on this rich information. However, due to the complexity of real-world situations and the limitations of the system, the generated code does not always pass. This requires students to carefully examine the error causes based on the Keil5+proteus verification results, further engage in dialogue with AIGC, and debug and optimize the code. The resulting programs are often large in volume, with high error rates and complexity, and require more precise prompts. In this process, students continuously enhance their ability to read and debug code;

The innovation layer is designed to build a personal code repository that generates code that meets creative needs and has higher accuracy. Students input multimodal data, and AIGC prioritizes generating complete code from the knowledge base when creating code. Afterward, it undergoes verification and optimization using keil5+proteus. Once optimized, students add their successfully debugged code and collected enterprise standard code to their personal code knowledge base. Through reinforcement learning mechanisms, this enhances the one-time success rate of the code. When students encounter similar problems in subsequent learning and practice, AIGC can generate more targeted and highly successful code based on the existing knowledge base and reinforcement learning outcomes. This also helps students achieve their creative goals more efficiently during innovative practices, driving them to continuously reach new heights in their studies.

4. Construction of teaching model driven by "ability map +AIGC"

The teaching model adopts a closed-loop implementation of "three processes, ten stages, and three levels." The "three processes" form the main thread of quality education, divided into pre-class guidance to awaken patriotic sentiments, in-class exploration to internalize scientific and craftsmanship spirits, and post-class expansion to practice core values, as shown in Figure 2. The "ten stages" form the main thread of knowledge education, with pre-class "searching and testing," combining online and offline learning, using task-driven methods to introduce relevant theoretical knowledge through case studies, solidifying students 'technical foundations. In-class "guidance, clarification, analysis, determination, solution, testing, and promotion," offline learning, integrates standards, norms, safety, and a pursuit of excellence throughout the process, cultivating professional skills. Post-class "expansion," offline personalized learning, uses intrinsic motivation to stimulate innovative spirit. The "three levels" represent teaching methods and means, progressively dividing knowledge from simple to complex into AIGC foundational layer, AIGC advanced layer, and AIGC innovative layer. The AIGC foundational layer is mainly used for pre-class guidance, the AIGC advanced layer for in-class exploration, and the AIGC innovative layer for post-class expansion.

The teaching content is skill points on the ability map, breaking down these points into multiple atomicized pieces of knowledge. Typical work task sheets from enterprises are selected to carry these atomicized pieces of knowledge, and according to students' cognitive patterns, enterprise work task sheets are transformed into learning tasks with progressively increasing difficulty. Learning Task 1 is a simple task, suitable for pre-class self-study. It uses the AIGC foundation layer to automatically generate basic program snippets for debugging and optimization, reducing the difficulty of pre-class learning; Learning Task 2 is a moderate task, suitable for in-class discussion and learning. It uses the AIGC advanced layer to automatically generate complete task code, with teachers assisting in verifying and optimizing the code; Learning Task 3 is a challenging task, suitable for post-class self-expansion. It uses the AIGC innovation layer to automatically generate code, which students can then verify and build their personal knowledge base on. Throughout the implementation process, teachers collect data on students' learning processes through the teaching platform, evaluate their task completion according to enterprise acceptance standards, and dynamically update the difficulty level of learning tasks based on the evaluation results. The enterprise work order system is used to capture real-time new technology requirements, updating the skill points in the knowledge graph. This forms a full-chain implementation process: "enterprise demand orientation → precise capability mapping → intelligent tool empowerment → dynamic feedback optimization."

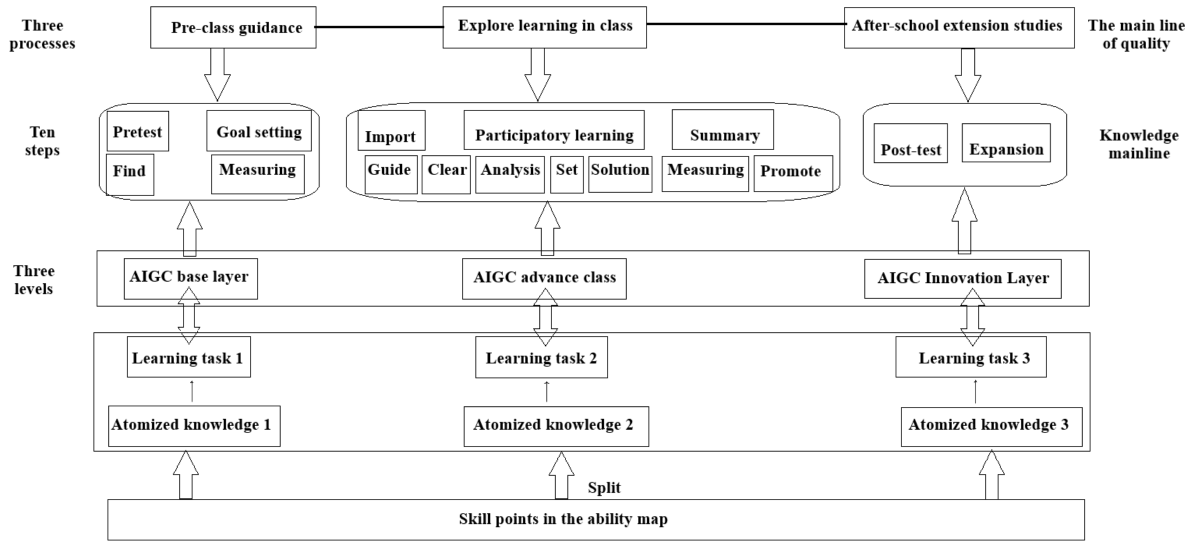

Figure 2: Ability map +AIGC "dual-wheel drive teaching model

5. Implementation of "Ability map +AIGC" teaching cases

5.1 Build a capability map to transform enterprise work orders into learning tasks

Taking the position of a microcontroller software engineer as an example, as shown in Table 1, the microcontroller course system is constructed based on job competency requirements, transforming linear textbooks into a combination of competency units. The process follows the competency map, and industry experts analyze the background, connotations, objectives, and practical applications of the competency map [9]. The core competencies for the microcontroller software engineer position are divided into seven modules: I/O ports, interrupt system, display output, key input, analog-to-digital conversion, communication protocols, and system debugging. Each module corresponds to three levels of skill points—basic, intermediate, and advanced—based on students' cognitive development, with a total of 21 skill points.

Taking the skill point "Timer Interrupt" as an example, as shown in Figure 3, the "Timer Interrupt" skill point is broken down into three atomic knowledge points: Knowledge Point 1-Timer Structure, Knowledge Point 2-Timer Program Writing, and Knowledge Point 3-Soft and Hard Integration of Timer Programs. Its prerequisite capability module is "IO Port." Therefore, the enterprise project task sheet for "Mass Production Development of Intelligent Floodlight Controller" is chosen as the carrier, which is then broken down into three learning tasks. Learning Task 1 carries Knowledge Point 1, Learning Task 2 carries Knowledge Point 2, and Learning Task 3 carries Knowledge Point 3. Each learning task includes its own sub-tasks, such as the software design of the floodlight in Learning Task 2, which includes Sub-task 1-Timer Interrupt Program Structure, Sub-task 2-Floodlight Mode Implementation, and Sub-task 3-PWM Breathing Light Implementation. The completion of each sub-task can be approached using different levels of AIGC based on difficulty, or a single sub-task can use three levels of AIGC.

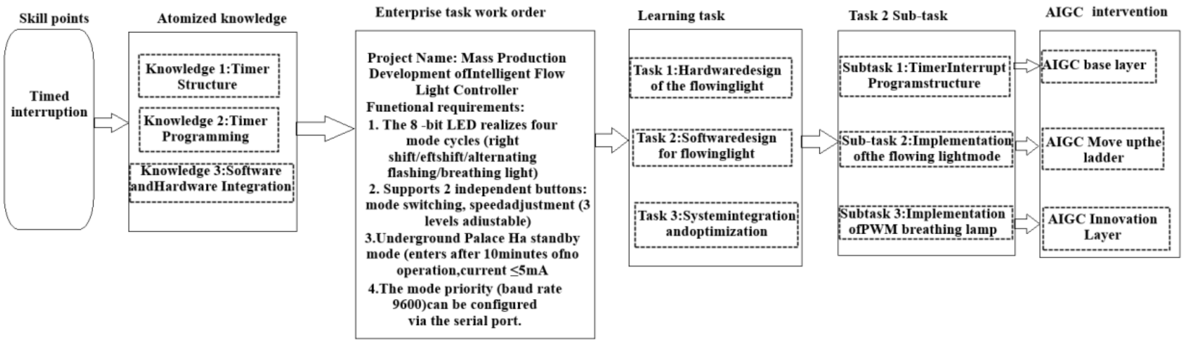

Figure 3: The skill points of the competency map are transformed into learning tasks

5.2 AIGC enables the implementation of learning task teaching

The teaching implementation is based on the design of the water lamp software in learning task 2.Sub-task 1 timer interrupt program structure is the pre-class learning task, sub-task 2 water lamp mode realization is the in-class learning task, and sub-task 3 PWM breathing lamp realization is the after-class learning task.

5.2.1 Pre-class AIGC basic layer: human-machine collaborative exploration and introduction

The pre-class phase is the stage for students to prepare their knowledge, primarily aimed at fostering their ability to learn independently. During this period, teachers issue pre-class learning sheets based on students 'cognitive levels and learning requirements, as shown in Table 2. The first sub-task of pre-class independent learning is to complete the timer interrupt program structure, following the pre-class steps and procedures, to sequentially complete the pre-class learning content and AIGC intervention. For the pre-class selection of a popular large model deepseek, students are introduced to the technological breakthroughs achieved through the independent innovation of the deepseek team, awakening their patriotic sentiments. Step one: The teacher posts various stream light videos from daily life on the Learning Hub platform to provoke students' thinking about "Why do LED lights move in a regular pattern?" Step two: Students ask deepseek with their questions, while the teacher provides key terms for deepseek's answers to guide further student thinking. Step three: For unfamiliar professional terms, the teacher records micro-lessons in advance and publishes them on the Learning Hub, guiding students to complete the micro-lesson learning according to the checklist. Step four: Based on the learning content, the teacher asks students to calculate the TMOD value and initial value for a 50ms timer interrupt. Step five: With tasks in mind, students start interacting with deepseek, comparing deepseek's responses with their calculated TMOD values and initial values through different prompts, to understand the accuracy of the prompt words and build their own prompt word bank. Step six: The teacher requires students to work in groups to prepare a report PPT on the 50ms timer interrupt program, reinforcing the knowledge learned independently before class and preparing for in-class learning.

Table 2: Pre-class learning sheet

Subtask 1: Timer interrupt program structure			
Pre-class session	step	Pre-class learning content	AIGC intervene
look for	one	Watch the video of the running light moving right and left, alternating flashing and breathing light in your life.	not have
	two	AI exploration	Q deepseek: Why does the running light move in a regular way? Answer deepseek: Timer control and the sequence of LED lights. Q deepseek: What knowledge should I master for timer control of 51 microcontroller? Answer deepseek: mode setting, initial value calculation, interrupt setting, timing interrupt function.
	three	Learn about the timing interrupt method setting, initial value calculation micro course, interrupt setting (pre-learning skills), and timing interrupt function on the Learning platform	not have
survey	four	Calculate the TMOD value and initial value of the 50ms timer interrupt	not have
	five	1. AI test; 2. Think: What is the difference between question 1 and question 2? Answer your thoughts on the Learning platform.	Question 1deepseek: Complete the 50ms timer interrupt initialization setting and interrupt function; Question 2deepseek: Using 51 microcontroller, the timing is 50ms. Please help me generate the initialization program and interrupt function of timer T0. The C language is required.
	six	Prepare a report PPT on the 50 ms interrupt program in groups.	not have

5.2.2 AIGC level of class: human-machine interaction to deepen knowledge

The in-class phase is the stage for deepening knowledge, primarily aimed at cultivating students 'professional skills to advance from theory to practice. During this phase, teachers guide while students explore independently, as shown in Table 3. Following the in-class segments and steps, teachers lead and students take the initiative to complete sub-task 2 in a stream-of-consciousness manner. The first step is "guidance," where teachers summarize the pre-class learning situation, and students collaboratively present their 50ms timer interrupt program report PPT, deepening their understanding of the structure of the timer interrupt program and the differences in the precision of prompt words, deepseek the accuracy of output results. Meanwhile, on the Learning Hub platform, teachers provide feedback based on the presentations, and groups evaluate each other's pre-class learning. The second step is "clarification," where teachers pose questions such as, "If the timer is set to 200ms, how do you calculate the initial value for the timer interrupt? How should the prompt word be written, and what will be the result?" Students discuss in groups using online resources and AIGC to explore scientifically. The third step is "analysis," where students use their research to engage in human-computer dialogue with deepseek. Based on the students' performance, teachers select groups with different conclusions to present and exchange ideas on stage. Through the clash of ideas, they understand the maximum basic timing for timer interrupts and the error codes in the initial value calculation by large models. The fourth step is "determination," where teachers further ask about the maximum and minimum basic timings for the four working modes based on the previous steps. Student calculation summary, teacher evaluation; Step five "solve," teacher guidance, students solve the 200ms timing; Step six "test," teacher guides students to interact with deepseek, completing the automatic code generation for Example (one) through Keil5 compilation and Proteus simulation to verify the correctness of the code. For Example (two), students independently interact with deepseek, practicing and exploring to complete the task, uploading the data from the practical process to an online platform. Step seven "promote," based on the 50ms and 200ms timing, students independently explore the 1s timing, completing the left and right shifts of the 2s cycle for the running light, promoting the application and transfer of knowledge. Teachers evaluate the completion according to enterprise acceptance standards, groups review each other's work, and upload the results to Learning Hub, collecting data for subsequent learning tasks, internalizing the spirit of scientific inquiry throughout the process.

5.2.3 Post-class AIGC innovation layer: human-machine symbiosis and integration of thought and action

After class is the stage for consolidating knowledge, primarily aimed at fostering students' ability to transfer and innovate their knowledge. During this phase, teachers assign sub-task 3: PWM breath light implementation, as shown in Table 4 of the post-class extension sheet. Students further complete tasks such as AIGC intervention, validation, evaluation, and feedback according to enterprise acceptance standards, uploading the results to the online learning platform Xuetong.

Table 4: Post-class learning sheet

Post-class session	Post-class tasks	AIGC intervene	Verify the task	Evaluation feedback
make rubbings from inscriptions	Sub-task 3: PWM breathing lamp implementation	1) Input: Generate PWM breathing light code for 51 microcontroller, frequency 100Hz, and brightness changes linearly 2) Output: Code framework with gamma correction and duty cycle lookup table tool	1) Call AIGC to generate code, debug and optimize; 2) Verify the PWM waveform through oscilloscope; 3) Modify the table lookup parameters and observe whether the brightness change conforms to human eye perception.	1) Automatic test to check whether the PWM error is within 2% positive or negative; 2) AIGC generates an evaluation report and marks the items that do not meet the standards.

5.3 Evaluation of teaching effectiveness

The research subjects come from an applied undergraduate institution in Province H, which aims to cultivate talent through a model of industry-academia collaboration and demand-driven education, thereby meeting the actual needs of regional economic and industrial development, aligning with the quasi-experimental requirements of this study. The experiment selects second-year students majoring in Robotics Engineering from the school-enterprise cooperative program as research subjects, setting up experimental groups (30 students) and control groups (30 students) at the class level. Before the start of the study, neither group of students had any experience using large language models for learning. All participating students signed informed consent forms and could withdraw from the experiment at any time during the research process.

The experimental group adopted a dual-driven approach of "Capability Map + AIGC" for course instruction, while the control group used traditional information-based teaching methods. Based on the course objectives of "Principles and Applications of Microcontrollers" and corporate acceptance standards, test questions were designed in two dimensions: foundation and capability. The foundation dimension included 40 multiple-choice questions worth 40 points and 12 fill-in-the-blank questions worth 60 points for program understanding. The capability dimension selected three enterprise work order tasks from simple to complex, assessing students 'understanding of problems, debugging programs, and designing programs, each worth 20 points, 40 points, and 40 points respectively. After testing, the highest and lowest scores in each part of both classes were removed, and the average scores were taken for each dimension, as shown in Table 5. The test results indicate that, in terms of foundational knowledge, the scores of both groups were evenly matched. However, in program understanding, problem understanding, program debugging, and program design, the experimental group scored significantly higher than the control group. This suggests that the dual-driven teaching approach of "Capability Map + AIGC" effectively enhances students' comprehensive application skills and strengthens their AI interaction skills, greatly meeting the demand for talent from enterprises. The effective implementation of this teaching model promotes a deeper alignment between engineering education and industry needs, exploring new paradigms for cultivating new productive forces. Additionally, during the teaching process, I found that students in the experimental group were very active and engaged in learning, with high levels of interaction and cooperation, indicating an internally driven learning process.

Table 5: Teaching effect of "ability map +AIGC"

group	foundation		ability		
	abc (40 points)	program comprehension (60 points)	Problem understanding (20 points)	program debug (40 points)	programming (40 points)
experimental group	38.2	56.5	18.6	35.5	38.7
control group	38.8	40.7	10.5	28.6	27.8

6. Conclusion

Despite initial practical tests showing that the "Capability Map + AIGC" dual-driven teaching model has achieved relatively satisfactory results, it is still necessary to continuously update the version of this model based on changes in the needs of enterprises, students, and teachers, as well as advancements in programming and AIGC technology. In summary, in the accelerated evolution of a new round of technological revolution and industrial transformation, the construction of a capability map through industry-education integration and the application of AIGC technology are key approaches for breaking through the boundaries of future education, representing effective pathways for human-machine collaboration, innovative talent cultivation, and trustworthy learning ecosystems[4].

Acknowledgement

Foundation Items: (1)Provincial Teaching Reform and Research Projects of Hubei

Undergraduate Universities in 2024"Ability Map+AIGC" enabling the teaching reform of control courses of robot engineering specialty(Project No:2024538); (2)Collaborative Education Program with Industry “Research on the reform of teaching methods under the meta universe Education” (Project No:220603231214852)

References

[1] China Economic Net. "ILO report shows - AI may exacerbate the social development gap" [EB/OL].(2024-08-12)http://www.ce.cn/xwzx/gnsz/gdxw/202408/12/t20240812_39100206.shtml

[2] Guo Feng, Yang Xiangqing, Zheng Chunhui, et al. A Preliminary Exploration of New Forms of Engineering Experiment Teaching Empowered by AIGC [J]. Research on Electro-Education, 2025,46(01):72-78+85. DOI:10. 13811/j.cnki.eer.2025.01.010.

[3] Zhu Sha, Li Jiayuan, Kuang Xiulin, et al. How Generative Artificial Intelligence Empowers the Cultivation of Students' Digital Literacy - An Empirical Study Based on Information Technology Courses [J]. China Educational Technology, 2025, (02):75-83.

[4] Ye Junmin, Yin Xinghan, Yu Shuang, et al. Generative AI Empowering Learning Analytics: Value Connotations, Practical Framework, and Development Directions [J]. Research on Electro-Education, 2025,46(01):86-92.DOI:10.13811/j.cnki.eer.2025.01.012.

[5] Wu Yang, Lu Yuqi, Du Jun, et al. The Endogenous Logic of Knowledge Graph-Driven Intelligent Learning [J]. China Educational Technology, 2025, (02):122-130.

[6] Zhang Meiling, Tan Hongyan. Application and Practical Reflection of Competency Mapping in Microbiology and Experimental Teaching Effect Evaluation [J/OL]. Biology Journal, 1-5[2025-04-24]. http://kns.cnki. net/kcms/detail/34. 1081.q.20241223.1639.002.html.

[7] Lu Dianan, Dang Yang, Wang Hongning, et al. Generative AI Empowering University Course Teaching: A Case Study of "Chemical Engineering Thermodynamics" [J]. Tsinghua University Education Research, 2024,45(05):89-98.DOI:10.14138/j.1001-4519.2024.05.008910.

[8] Xu Huifen, Pang Chang, Zheng Ruxin, et al. Generative AI Empowering Agile Course Development and Practice Research [J]. Distance Education Journal, 2024,42(05):95-101.DOI:10.15881/j.cnki.cn33-1304/g4.2024.05.010.

[9] Wang Youmei, Zhao Wenzhu, Wan Ping, et al. Digital IQ and Its Competency Map: International Progress and Future Education Framework [J]. China Educational Technology, 2020, (01):46-55.

Table 1: Capability map of microcontroller software engineer position

module / administrative levels	I/O mouth	interrupt system	Display output	Key input	ADC	communicating protocol	system debug
Innovation layer	I/O port expansion	Interrupt expansion	LCD show	Matrix keyboard	integrated application	UART/CAN protocol	Proteus Debugging
Move up the ladder	I/O port drive	timer interruption	REG show	Individual buttons	DAC apply	SPI protocol	Keil shakedown test
The base layer	memorizer	external interrupt	LED show	Key characteristics	ADC apply	I2C protocol	STC make use of

Table 3: Implementation of in-class tasks

Subtask 2: Implementation of running lamp mode				
Class segments	step	Teacher guidance	Student-led	
			student activities	AIGC intervene
lead	one	1) The group reports the 50ms timing interrupt system structure program or timing mode control TMOD or timing initial value calculation or interrupt function in one minute; 2) Group discussion and mutual evaluation (evaluation form released on Learning platform)	1) Group report; 2) Understand the three parts of the timer interrupt initialization program and the timer interrupt function.	not have
bright	two	Q: How do we design it if we want to time 200ms?	Students discuss and explore science	AIGC looked up information
divide	three	1) Please use large models (deepseek/Beanbag/Wenxin Yiyang/Xunfei Xinghuo, etc.) to complete the program writing with a time interval of 200ms. 2) Look at the initial values of the large models. Are their timing correct?	1) Use of large models; 2) Calculate the timing initial value generated by the large model? Think about the correctness of the program; 3) It is understood that the timing initial value of all large models is basic timing, and the 200ms timing is not completed, so the program is written incorrectly; 4) Scientifically verify the errors in the large model generation program.	Question 1: Select the timer T0 of 51 microcontroller, and help me generate the program with a timing of 200ms. The C language is required. Answer 1 (Big Model): The program was written in a serious manner.
fix	four	Q: What are the maximum and minimum timing of the four working modes?	1) Student answer: Method 1: Min = 1us, Max=8192us Method 2: Min = 1us, Max=65536us Mode 3: Min=1us, Max=256us Method 4: Min = 1us, Max=256us 2) It is impossible to understand the direct way of selecting timing 200ms.	not have
separate	five	Q: The maximum timing of mode 2 is 65ms, which is the longest among the four working modes, but it cannot meet the timing of 200ms. If one interrupt is 65ms, what about two interrupts? Three interrupts?	Student answer 1: If the basic timing is 65ms, the interruption of 3 times is 195ms; Student answer 2: you can choose the basic timing of 50ms, and 4 interrupts are 200ms; Student understanding: Accurate timing is a professional scientific	not have

			spirit.	
survey	six	Example (1): Please use a large model (deepseek/bean bag/Wenxin Yiyang/Xunfei Xinghuo, etc.) to complete a LED light flashing control program with a cycle of 400ms. Example (2): Please use a large model (deepseek/bean bag/Wenxin Yiyang/Xunfei Xinghuo, etc.) to complete the control program of eight LED lights alternating flashing with a cycle of 400ms.	1) Large model usage; 2) Analyze the correctness of AIGC program; 3) Use Keil5 and Proteus to debug and verify the correctness of AIGC program; 4) Upload programs and simulation circuits on the Learning platform, and encounter problems.	Example (1) Question 1: Select the timer T0 of 51 microcontroller to generate a flashing program for LED light on for 200ms and off for 200ms, requiring C language; Example (1) Answer 1 (Big Model): The program was written in a serious manner. Example (1) question 2: you give me the wrong timing, I need to time 200ms, please help me generate the program again; Example (1) Answer 2 (large model): Accept the timing error and regenerate the program. Example (2) question 1:... Example (2) Answer 1 (large model):... Example (II) question 2:... Example (2) Answer 2 (large model):...
short	seven	Please use large models (deepseek/bean bag/Wenxin Yiyang/Xunfei Xinghuo, etc.) to complete the right and left shift of the 2s cycle.	1) Large model usage; 2) Analyze the correctness of AIGC program; 3) Use Keil5 and Proteus to debug and verify the correctness of AIGC program; 4) Upload programs and simulation circuits on the Learning platform, and encounter problems.	Question 1: Select the timer T0 of 51 microcontroller, and help me generate the program of moving the running light right and left for 1s, which requires C language; Answer 1 (Big Model): The program was written in a serious manner. Question 2: You gave me the wrong timing, I need to time 1s, please help me generate the program again; Answer 2 (large model): Accept the timing error and regenerate the program.