A Case Report on the Development of a Stage-Based Artificial Intelligence Science Education Curriculum and Practice Platform for Adolescents

Jianmin Guo, Qihang Guo, Yanyu Lan, Rongxing Yang, Bin Zhu

Abstract— **Artificial intelligence (AI) literacy has become an important component of science, technology and engineering education, but many K-12 implementations still face fragmented content, limited hands-on resources and insufficient alignment across school stages. This case report presents the design and early implementation plan of an adolescent AI science education project in Tianjin, China. The project integrates a stage-based curriculum for upper primary, junior secondary and senior secondary learners with a locally developed AI practice kit, sensor-based embedded experiments, an online course resource system and school outreach mechanisms. The curriculum begins with Scratch and introductory Python for programming thinking, extends to Python data structures and sensor control in junior secondary school, and culminates in embedded AI projects, automatic obstacle-avoidance vehicles, speech recognition, natural language processing and voice dialogue systems in senior secondary school. The case is analyzed against computational thinking, AI literacy and the AI4K12 five-big-ideas framework. The results of the design process suggest that a curriculum-plus-practice architecture can reduce the distance between abstract AI concepts and student experience, while supporting progressive development of logical thinking, problem solving, innovation awareness and responsible technology use. The paper contributes a replicable implementation model for AI science popularization projects that combine curriculum resources, hardware products, teacher support and online-offline dissemination.**

Index Terms— **AI literacy, artificial intelligence education, K-12, science popularization, embedded systems, Python, Scratch, curriculum design.**
Sub Area: Computer Science Education
Broad Area: Artificial Intelligence in Education

I. INTRODUCTION

Artificial intelligence is no longer only a professional engineering topic; it is increasingly a general literacy issue for young learners. UNESCO argues that AI and education policies should both use AI to improve learning and prepare citizens to live and work with AI [1]. In China, the New Generation Artificial Intelligence Development Plan called for AI-related courses in primary and secondary schools and gradual promotion of programming education [2]. These policy signals create a practical demand for curriculum resources that are age-appropriate, technically credible and feasible for schools.

However, adolescent AI education often encounters three obstacles. First, learning content is sometimes limited to isolated programming lessons or demonstrations of commercial products, making it difficult for students to form a coherent understanding of AI systems. Second, many schools lack teachers, hardware kits and experimental platforms that connect AI concepts with tangible practice. Third, abstract topics such as machine learning, perception and natural interaction are often introduced too late or too theoretically, while younger students need concrete experiences that match their cognitive development.

This paper reports a case of developing an adolescent AI science education curriculum system and practice platform. The project is based on local materials including an implementation plan, an AI development practice tutorial, curriculum diagrams, prototype devices and a planned online learning system. Rather than presenting a controlled experiment, the paper documents the design rationale, resource architecture, curriculum progression and implementation pathway. The purpose is to provide a directly usable case report for engineering education, science popularization and K-12 AI curriculum developers.

II. RELATED WORK AND DESIGN BASIS

The project draws on three strands of literature. The first is computational thinking. Wing described computational thinking as a broadly applicable skill set, not merely a technique for computer scientists [3]. For adolescents, this implies that programming courses should cultivate decomposition, abstraction, algorithmic expression and debugging habits rather than only syntax. The project's Scratch and Python modules are therefore positioned as thinking tools before they become engineering tools.

The second strand is creative programming and constructionist learning. Scratch was designed to make programming more meaningful, tinkerable and social for young learners [4]. Brennan and Resnick further proposed that computational thinking can be understood through concepts, practices and perspectives [5]. These ideas support a learning sequence in which upper primary students create games, animations and simple interactions before entering text-based programming and hardware control.

The third strand is AI literacy. Long and Magerko define AI literacy through competencies that enable people to critically understand, use and evaluate AI technologies [6]. The AI4K12 initiative organizes K-12 AI education around perception, representation and reasoning, learning, natural interaction and societal impact [7]. The present case uses these frameworks as design references: perception is

Manuscript received June 14, 2026
Jianmin Guo, Tiangong University, Tianjin, China
Qihang Guo, Tiangong University, Tianjin, China
Yanyu Lan, Tiangong University, Tianjin, China
Rongxing Yang, Tiangong University, Tianjin, China
Bin Zhu, Tiangong University, Tianjin, China
Project support: This work was supported by the 2024 Tianjin Science and Technology Popularization Project, "Development of an Artificial Intelligence Science Popularization Curriculum System for Adolescents"

represented through sensors and speech input; representation and reasoning through algorithms and Python logic; learning through iterative projects; natural interaction through speech recognition and dialogue systems; and societal impact through privacy, safety and responsible use.

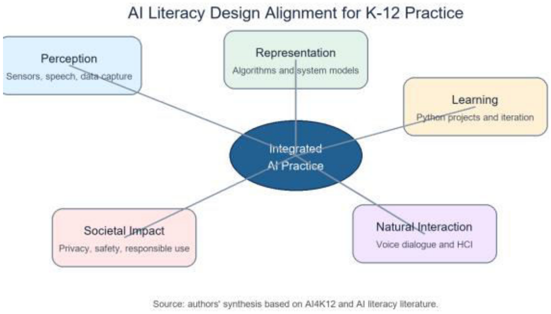

Fig. 1. AI literacy design alignment used in this case report.

III. CASE CONTEXT AND DEVELOPMENT GOALS

The case project aims to develop and promote an AI science education curriculum system for adolescents through simple, understandable and practice-oriented resources. The target learners include upper primary school students, junior secondary students and senior secondary students. The project is carried out by a university team with experience in computer application, AI, big data, sensors, signal processing and educational information systems, in collaboration with local educational resources.

The development goals are fourfold. First, the project constructs a stage-based curriculum resource package that introduces AI concepts, programming thinking and engineering practice in a progressive manner. Second, it develops AI science popularization products such as embedded development boards, sensor modules, automatic obstacle-avoidance vehicles and voice-control devices. Third, it prepares supporting tutorials and video lessons so that learning resources are not dependent on individual teachers. Fourth, it builds an online course resource system that supports PC, mobile phone and tablet access, and can be connected with public science popularization platforms.

The expected educational value is not short-term programming training, but long-term AI literacy. Students are expected to understand what AI is, where it appears in daily life, how AI systems perceive and respond to the environment, and how programming and sensors can be combined to solve practical problems. The project also emphasizes teamwork, creativity, safety awareness and ethical reflection.

IV. CURRICULUM AND PRODUCT ARCHITECTURE

A. Stage-Based Curriculum Design

The curriculum uses a progressive architecture across three learning stages. In the upper primary stage, the main task is to cultivate interest and basic programming thinking. Learners begin with Scratch programming, movement commands, appearance commands, sound commands, events, control structures, sensing and operations. They then enter introductory Python through simple programs, naming rules, input and output, GPIO library installation and basic LED or touch-switch experiments.

In the junior secondary stage, the curriculum deepens Python learning. Students study data types, variables, strings, lists, tuples, dictionaries, conditional control, loops, functions, modules, file operations, object-oriented programming and exceptions. These programming topics are connected with sensor experiments, including atmospheric pressure, sound, vibration, flame, light, buzzer, infrared, temperature and humidity, servo, ultrasonic, microphone and display modules. The pedagogical intention is to make syntax meaningful by attaching it to visible physical effects.

In the senior secondary stage, the curriculum shifts toward comprehensive AI and embedded practice. Students use analog-digital conversion boards, water level sensors, soil humidity sensors, light intensity sensors, smoke sensors, relays, collision detection modules and automatic obstacle-avoidance vehicles. They then explore voice wake-up, speech recognition, audio preprocessing, feature extraction, acoustic models, language models, natural language processing, speech synthesis and multimodal dialogue management. This stage links AI concepts with mathematics, physics, information technology and English.

Fig. 2. Stage-based science education resource architecture from project materials

B. Practice Kit and AI Science Products

The practice platform is designed as a curriculum-plus-hardware system. The hardware includes self-developed AI development boards, embedded expansion boards and sensor modules. Typical experiments include lighting an LED, controlling a touch switch, reading environmental data, using ultrasonic distance measurement, operating buzzers and controlling servos. These experiments create a bridge from programming to perception and action.

The project also develops AI science popularization products for student projects. Automatic obstacle-avoidance vehicles and voice dialogue obstacle-avoidance vehicles are used as comprehensive prototypes. In these prototypes, learners can connect perception, decision and action: sensors collect environmental information, Python scripts process data or control logic, and the vehicle performs navigation or avoidance. Voice interaction modules further allow students to experience natural language input, speech recognition and speech synthesis in a concrete device.

This architecture addresses a common weakness in AI popularization: students may hear about intelligent systems without understanding their components. By opening the system into sensors, boards, code and behavior, the project gives learners a visible structure of AI applications.

Fig. 3. Self-developed Raspberry Pi GPIO sensor expansion board used in the practice kit.

C. Online Resource System

The online resource system is planned with Java, Spring Boot and MyBatis technologies. It includes science popularization videos, programming courses, experimental courses and frontier AI content. It is designed to support multiple terminals and provide learning resources before and after offline activities. In addition, the system can collect learning data and provide differentiated learning suggestions, which is consistent with the idea of using technology to support personalized learning while maintaining teacher guidance.

Online resources are not treated as a replacement for practice. Instead, they operate as a dissemination layer for tutorials, videos, experiment instructions and project cases. This is important for schools that have limited AI teachers because repeatable resources can reduce preparation burden and improve consistency.

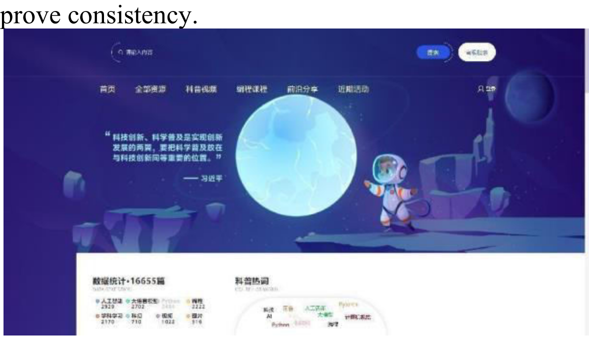

Fig. 4. Prototype interface of the online AI science popularization resource system

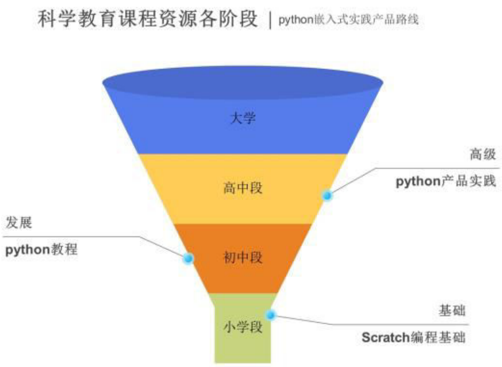

Fig. 5. Progressive route from Scratch and Python basics to embedded AI practice.

V. REPRESENTATIVE LEARNING ACTIVITIES

A. Primary Stage Example: From Python Syntax to LED Control

A representative upper-primary transition activity is the LED lighting experiment. After students learn basic Python syntax, output statements and program structure, they install or use the GPIO library and control a red-yellow-green LED module. The activity is intentionally simple: students map a pin number to a physical component, execute a loop, observe the LED state and modify the waiting time. In this process, abstract ideas such as variables, loops and Boolean output become visible through hardware behavior.

The teaching sequence can be organized as four steps: identifying the LED module and pins, reading the sample Python program, predicting the output sequence, and changing the code to create a new light pattern. The learning evidence includes whether students can explain the relation between code and pin output, whether they can debug a wrong pin assignment, and whether they can design a different timing rule.

```
import RPi.GPIO as GPIO
import time

led_pin = 5
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)
GPIO.setwarnings(False)
try:
    while True:
        GPIO.output(led_pin,GPIO.HIGH)
        time.sleep(1)
        GPIO.output(led_pin,GPIO.LOW)
        time.sleep(1)
finally:
    GPIO.cleanup()
```

Fig. 6. Python GPIO code example for blinking an LED in the implementation plan.

Fig. 7. LED module result used to connect code with observable output.

B. Junior Secondary Example: Multi-Sensor Inquiry

At the junior secondary level, students move from single-output experiments to sensor-based inquiry. The implementation plan lists modules such as atmospheric pressure, sound, vibration, flame, light, buzzer, infrared, temperature-humidity, servo, ultrasonic, microphone and display sensors. A typical lesson can ask students to compare environmental signals, collect readings, identify thresholds and trigger feedback such as a buzzer or display message.

This activity is useful because it links Python data structures and control flow with empirical measurement. For example, students may store repeated sensor readings in a list, calculate simple averages, and use an if-else structure to trigger a warning. The case therefore supports both computational thinking and scientific inquiry: students must define a problem, collect data, design rules and test whether the system behaves as expected.

Fig. 8. Sensor modules used for junior secondary embedded-system inquiry activities.

C. Senior Secondary Example: Collision Detection and Obstacle Avoidance

At the senior secondary level, the project uses robot collision detection and automatic obstacle-avoidance vehicles as integrated AI practice tasks. Students connect ultrasonic or collision sensors, interpret the detection signal, and control the vehicle response. The programming task is no longer only a syntax exercise; it includes event monitoring, threaded execution, interface feedback and safety-related decision rules.

The collision detection interface in the implementation plan records repeated collision events with timestamps. This interface provides a concrete way to discuss AI system monitoring: perception produces a signal, software records and interprets the signal, and the device changes behavior. Students can then discuss how detection frequency, sensor placement and response delay influence the reliability of the system.

Fig. 9. Collision detection interface recording sensor events in the robot practice task.

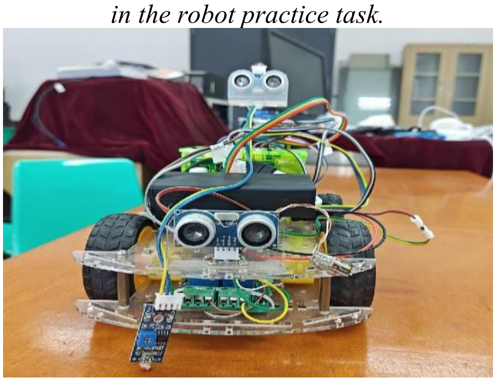

Fig. 10. Automatic obstacle-avoidance vehicle prototype from the implementation plan.

D. Senior Secondary Extension: Voice Dialogue Vehicle

The implementation plan further extends the vehicle task toward voice interaction. Learners can explore voice wake-up, speech recognition, text understanding, speech synthesis and multimodal dialogue management. This extension is closely aligned with AI4K12's natural interaction idea: students experience how an intelligent system accepts human language input, converts it into machine-processable form and produces an action or response.

In a classroom setting, the voice dialogue vehicle can be used as a capstone project. Teams may divide roles into hardware connection, Python control logic, speech module configuration, dialogue design and presentation. Compared with a purely software chatbot, the embodied vehicle gives students a stronger sense of the relationship among language, perception and physical action.

VI. IMPLEMENTATION PATHWAY

The implementation plan follows a development-to-promotion pathway. The first phase conducts needs analysis, project planning and personnel division. The second phase designs experimental equipment and curriculum materials for each school stage. The third phase develops the online system and compiles stage-based textbooks and tutorials. The fourth phase develops embedded system functions and validates whether they satisfy teaching purposes. The final phase checks the curriculum and equipment, debugs online resources and prepares tutorial publication.

For school promotion, the project proposes cooperation with schools in Heping District, Tianjin. The promotion model combines online and offline activities: classroom courses, science lectures, programming challenges, device demonstrations and public online resource release. This mixed mode is suitable for science popularization because students need repeated exposure, visible outcomes and opportunities to display their work.

Teacher support is a necessary part of implementation. The curriculum package should include lesson objectives, prerequisites, experiment procedures, troubleshooting notes, safety reminders and extension tasks. For hardware-based AI education, teachers also need guidance on circuit connection, sensor calibration, classroom management and student data protection.

VII. DISCUSSION

A. Innovation of the Case

The first innovation is continuity across school stages. Instead of offering separate activities, the project builds a pathway from visual programming to Python, from Python to sensors, and from sensors to embedded AI products. This design supports gradual accumulation of concepts and avoids both premature abstraction and delayed exposure.

The second innovation is the integration of AI literacy with engineering practice. AI topics such as perception, speech recognition and dialogue systems are not presented only as definitions. They are linked with microphones, sensors, development boards, control programs and physical prototypes. This helps students understand AI as a system of data, models, interfaces and consequences.

The third innovation is online-offline resource integration. The online platform provides repeatable learning materials, while offline practice provides embodied experience and teamwork. This combination is especially useful in regions where specialized AI teachers and laboratories are unevenly distributed.

B. Educational Implications

For curriculum designers, the case suggests that AI education should not begin with advanced models alone. A more accessible route is to begin with programming thinking, move to physical perception and control, and then introduce AI methods that explain intelligent behavior. For schools, the case suggests that low-cost sensors and embedded devices can make AI visible and testable. For students, the main learning outcome is not only coding skill but also the ability to ask how a system senses, represents, learns, interacts and affects society.

The project also shows that AI science popularization can connect multiple subjects. Mathematics appears in variables, functions and data processing; physics appears in circuits, sensors and motion; information technology appears in programming and system design; English appears in technical vocabulary and human-computer dialogue. Such integration supports a broader STEM learning ecology.

C. Limitations and Future Work

This paper is a case report based on development materials and implementation planning rather than a controlled learning-effect experiment. Therefore, claims about student outcomes should be interpreted as design expectations. Future work should collect classroom data, including participation records, student artifacts, teacher feedback, pre- and post-course AI literacy questionnaires, and rubrics for computational thinking and engineering design.

The project should also strengthen risk control. Hardware experiments require safe circuit connection, age-appropriate supervision and clear emergency procedures. Online learning data require privacy protection and transparent data use rules. AI content should include responsible technology use, bias awareness and limitations of AI systems.

VIII. CONCLUSION

This case report presents a stage-based adolescent AI science education curriculum and practice platform. The project integrates Scratch, Python, embedded sensors, AI practice products, tutorial resources and an online learning system. Its main contribution is a curriculum-plus-practice model that connects AI concepts with tangible experiments and progressive school-stage development. For low-threshold but practical AI education projects, this model provides a replicable approach: begin with programming thinking, deepen through Python and sensors, culminate in AI-enabled products, and support dissemination through online-offline resources.

As AI becomes part of everyday learning and life, adolescent AI education should help students move beyond passive use toward understanding, creation and responsible judgment. The proposed curriculum system is one practical step toward that goal.

ACKNOWLEDGMENT

The authors acknowledge the project team members and cooperating educational institutions involved in the development of the adolescent AI science education curriculum resources and practice platform.

REFERENCES

[1] F. Miao, W. Holmes, R. Huang, and H. Zhang, AI and Education: Guidance for Policy-makers. Paris: UNESCO, 2021.
[2] State Council of the People's Republic of China, New Generation Artificial Intelligence Development Plan, 2017.
[3] J. M. Wing, "Computational thinking," Communications of the ACM, vol. 49, no. 3, pp. 33-35, 2006.
[4] M. Resnick et al., "Scratch: Programming for all," Communications of the ACM, vol. 52, no. 11, pp. 60-67, 2009.
[5] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," Proc. AERA Annual Meeting, Vancouver, 2012.
[6] D. Long and B. Magerko, "What is AI literacy? Competencies and design considerations," Proc. CHI Conference on Human Factors in Computing Systems, pp. 1-16, 2020.
[7] AI4K12 Initiative, "Five big ideas in AI and K-12 AI guidelines," AAAI and CSTA, 2019-2024.
[8] S. Touretzky, C. Gardner-McCune, F. Martin, and D. Seehorn, "Envisioning AI for K-12: What should every child know about AI?" Proc. AAAI Conference on Artificial Intelligence, vol. 33, no. 1, pp. 9795-9799, 2019.
[9] K. Brennan, A. Monroy-Hernandez, and M. Resnick, "Scratch: Creating and sharing interactive media," Proc. 9th International Conference on Interaction Design and Children, pp. 75-78, 2010.

BIOGRAPHY

Jianmin Guo is an associate professor at Tianjin Polytechnic University. His work includes computer applications, artificial intelligence, big data technology and educational information systems. Author biographies and final affiliations should be revised according to the submission author's official information.

APPENDIX: CURRICULUM SUMMARY

Stage	Core Tools	Practice Focus	Expected Competence
Upper primary	Scratch; introductory Python	Games, basic commands, LED and switch experiments	Interest, programming thinking, basic logic
Junior secondary	Python; sensor modules	Data types, control, functions, files, object-oriented basics and sensor control	Problem solving, debugging, embedded practice
Senior secondary	Python AI practice; speech and NLP modules	Obstacle-avoidance vehicle, speech recognition, synthesis and dialogue system	Integrated AI design, teamwork and responsible use