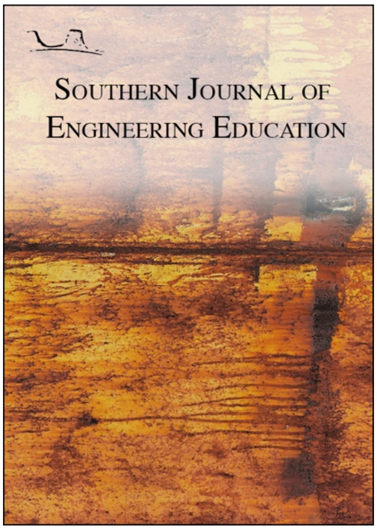


SOUTHERN JOURNAL OF
ENGINEERING EDUCATION

Publication details, including instructions for authors and other information can be found at https://sjee.org.za/

# Transforming embedded systems education: The potential of large language models

Christos Pietersen[a]
Reneé Smit[b]

[a] *Department of Electrical Engineering, Tshwane University of Technology, Pretoria, South Africa*
[b] *Department of Electrical Engineering, University of Cape Town, Cape Town, South Africa.*

Submission received: 11 June 2024 / Published: 14 November 2024

To cite this article: Pietersen, C., & Smit, R. (2024). Transforming embedded systems education: The potential of large language models, *Southern Journal of Engineering Education*, *3*, 62–83. DOI: 10.15641/sjee.v3i1.1572.

*Southern Journal of Engineering Education, 2024*
*Vol 3, 62–83*

# Transforming embedded systems education: The potential of large language models

Christos Pietersen*[a]
Reneé Smit[b]

[a] *Department of Electrical Engineering, Tshwane University of Technology, Pretoria, South Africa*
[b] *Department of Electrical Engineering, University of Cape Town, Cape Town, South Africa.*

This conceptual article delves into the potential benefits, challenges, and future directions of how educators might adapt practices to accommodate the use of artificial intelligence (AI) tools, in particular large language models (LLMs), with embedded systems education as a case study. Drawing on literature pertaining to embedded systems education and the associated challenges, a new way of approaching embedded systems education is suggested, where students and LLMs work together to solve problems. This article proposes that AI technologies have the potential to improve the productivity of students as they learn to programme and that LLMs can be leveraged as personal tutors, facilitating adaptive tuition. The role of educators remains crucial in this process, as students still require scaffolding and guidance on prompting LLMs. This article suggests that educators have different options when considering how to teach embedded systems with LLMs present, by changing the emphasis of teaching to focus on the process of learning and understanding and using constructive alignment of learning activities and assessment with the new goals. This promises to be an exciting avenue of research and practice going forward.

*Keywords*: Large language models; embedded systems education; conceptual article; constructive alignment

## Introduction

Embedded systems, which form the backbone of various technologies in the modern age, require a unique blend of hardware and software knowledge with cross-domain applications ranging from consumer electronics to industrial machines. The academic subject of embedded systems is regarded as a new and relatively undefined subject that incorporates areas such as computer science, automatic control and electrical engineering (Grimheden & Törngren, 2005a). Teaching embedded systems in higher education is challenging due to interdisciplinary relationships between high-level programming knowledge and low-level hardware interactions. Functional software development hinges on the ability of the student to write programmes to be deployed on hardware systems, with the objective of meeting the design

*Email: PietersenC@tut.ac.za

brief specifications. If the software has any semantical or syntactical errors, the hardware will not function as expected, if at all. If the hardware has any connection problems, then the software will fail to execute as purposed. Literature abounds with pedagogical strategies suggesting how to better teach this complex cross-domain discipline to deliver quality graduates to industry (Grimheden & Törngren, 2005b; Ibrahim et al., 2014; Nakutis & Saunoris, 2010; Sangiovanni-Vincentelli & Pinto, 2005). As sensors and microcontrollers advance in complexity and capabilities, so too do the demands that industry place on new graduates, requiring workers with more complex knowledge and skills (Ibrahim et al., 2014). The modern era of artificial intelligence (AI) technologies brings with it new opportunities and challenges. There are promising aspects that this technology has to offer education, but uncertainty resides in the method. It might be hard for teachers to work out how to use AI in the classroom and still meet a course's intended learning outcomes (ILOs), and how to assess students to determine if the outcomes are met.

The advent of large language models (LLMs) has ushered a new era filled with possibilities in the field of education, specifically embedded systems education. AI has been present in education for some time but LLMs, such as ChatGPT, have made AI technology readily available to the public. LLMs have the potential to generate both syntactically correct and semantically meaningful code, which is relevant as software development plays a significant role in the development of an embedded system. This raises concern for educators who fear degrading the competency of their students, as LLMs can do the 'heavy lifting' for students as far as code generation goes, resulting in over-reliance on technology (Mahapatra, 2024; Shabunina et al., 2023).

This article proposes that educators should use the opportunity of the potentially disruptive influence of LLMs to reconsider curriculum objectives for embedded systems education. One approach to consider is constructive alignment, a concept that has been used in educational research for many decades. In more recent work, Biggs (2014) emphasised the importance of the behaviour of students to be developed as well as the context in which this behaviour will operate. Embedded systems education in the context of AI requires different student behaviours than those previously taught. At first glance, AI might be seen as posing a threat to the development of proper learning behaviour in students, as they rely on LLMs to perform tasks for them. Issues arise such as academic dishonesty, students not learning to develop code that successfully integrates the software with the hardware and performing debugging procedures.

Even though AI technology comes with its own challenges in terms of integrating LLMs in education at large, and the alignment of LLM output with educational goals should be considered carefully. LLMs certainly have the potential to serve as powerful tools for teaching and learning embedded systems as they can provide real-time feedback, interactive guidance, and debugging assistance (Englhardt et al., 2023). Using constructive alignment to reframe the ILOs, teaching and learning activities, and assessment methods (Biggs, 2014) could benefit educators and students alike, and lead to possible transformation with regards to how embedded systems subjects are presented to students.

This conceptual article aims to delve into the potential benefits, challenges, and future directions of how educators might adapt practices to accommodate the use of AI tools, using LLMs in embedded systems education as a case study. As a conceptual contribution it draws on a framework proposed by Jaakkola (2020), having identified the focal phenomenon and the various concepts that relate to the focal phenomenon as they assist in the conceptualisation of the use of LLMs in embedded systems education. An argument is built using constructive alignment as the backbone, drawing on the concepts covered in literature.

## Theoretical considerations for conceptual work in a new field

As mentioned, the focal phenomenon studied in this paper is the use of LLMs in embedded systems education and how educators within this field could adapt their approaches to integrate this technology in the classroom. The *model*-type research design, as proposed by Jaakkola (2020), is used which provides a roadmap for understanding the new possibilities of LLMs in embedded systems education. Literature selected for review is based on the key variables and the association they have with the main idea. These concepts are AI in education, LLMs in education, embedded systems education, and the ability of LLMs to develop embedded systems solutions. This article starts with a review of relevant literature which forms the elements of the conceptual framework. The specific goal of this article is to outline the focal concept, how it is changing, the mechanisms employed and conditions that may affect it (MacInnis, 2011), and an outline of the pitfalls and potentials of LLMs in embedded systems education.

## Embedded systems and AI

The literature reviewed in this paper seeks to understand the relationships between the different aspects that play a role in embedded systems education. Before discussing the potential and challenges of large language models in embedded systems, one must understand the concepts involved in embedded systems education. These include, but are not limited to, programming education challenges, and artificial intelligence in education. Understanding the elements that influence the development of an embedded systems engineering graduate will be of use when considering how LLMs could play a role in the process.

### *AI in education – possibilities and concerns*

The application of AI in education (AIED) can be discussed under three overlapping categories: Student-focused, teacher-focused, and institution-focused AIED (Holmes & Tuomi, 2022, p. 550). Table 1 provides a taxonomy of AIED systems that are commonly available and researched. The focus of this article is on student-focused AIED and how LLMs and chatbots support learning.

Table 1: *AIED taxonomy (Holmes & Tuomi, 2022, p. 550)*

| **Student-focused AIED** | **Teacher-focused AIED** | **Institution-focused AIED** |
|---|---|---|
| Intelligent tutoring systems | Plagiarism detection | Admissions |
| Chatbots | Curation of learning materials | Course planning and scheduling |
| Automatic formative assessment | Automatic summative assessment | Identifying students at risk of dropping out |
| AI-assisted apps (mathematics, text-to-speech, language learning) | AI teaching assistant | |

Benjamin Bloom (1968) argued that all students engage learning activities with varying levels of prior knowledge, with different capabilities to meaningfully engage with the learning activity at hand, and therefore require varied support (Guskey, 2007; Holmes & Tuomi, 2022) to attain the same level of mastery of a topic or aspects of a topic. According to Holmes & Tuomi (2022), Bloom showed that individual tutoring can lead to two standard deviations in learning gain as compared to traditional whole-class teaching. This is perhaps the greatest opportunity for AI in education as it can facilitate individualised learning and tutorage for students. Embedded systems education can be considered fertile soil for the application of AI to provide students with much-needed individual support, depending on their own level of existing knowledge.

Although the potential benefits are promising, researchers warn that educators should approach AIED with caution. Dramatic claims regarding the capabilities of AI have later been found to be inflated and false (Selwyn, 2022). This prompts us to enter the world of AIED with realistic expectations around the limitation of the technology and then to foster discussions with fellow practitioners regarding how to implement it. Not all aspects of education are quantifiable and not all facets of students and the learning process can be captured and represented by data points (Selwyn, 2022). This means that while AIED has great potential, it should be implemented with care. Selwyn (2022) offers the following broad areas that remain points of contention and discussion amongst academics:

1. Hyperbole: exaggerated claims regarding the potential of AI in education.
2. Limitations: the limitation of AI in simulating real world issues within social context. Understanding the difference between 'technologically smart' but 'socially stupid' systems remain important.
3. Ideology in debates around AIED: acknowledging competing values, interests and agendas that underpin values of only one party (such as computer scientists) in contrast with broader interests that may offer counter arguments, like social justice concerns around privacy of the individual.
4. Environmental and ecological costs of AIED: production, consumption and disposal of digital technologies and acknowledging the impact it could have on the planet.

Selwyn (2022) concludes that the future of AI in education should be approached as contested and subjected to scrutiny prior to its integration in education, and not merely accepted as a neutral agent that will automatically bring about good. This point holds value and should be taken seriously while being realistic about how society is adopting this technology. According to Mahapatra (2024), accuracy and reliability is of concern as students potentially can be exposed to biased data, out-of-date knowledge, and misinformation, all depending on the data sets used for training the LLM.

Another concern is the prevention of plagiarism. The issues here are, firstly, students using LLMs to plagiarise and submit unoriginal work. Secondly, plagiarism detectors are easily bypassed as to produce similarity scores of 20% or less (Mahapatra, 2024). Working around the issue of plagiarism is challenging and contributes to the workload of educators as it requires that each student submission should be checked for plagiarism as well as AI detection, although AI detection also has a low success rate at this stage. As a sidenote, one might also argue the imperfections of LLMs could potentially be leveraged in learning activities where students must critically evaluate LLM outputs to determine their accuracy. This is not to suggest a default modus operandi, but as an example of how to put even a flawed LLM output to good use.

Students are making use of this technology, whether educators approve of it or not. The era prior to the launch of LLMs was perhaps different in the sense that accessibility to AI was limited, but open-source models such as ChatGPT disrupted the discussion and now require further investigation in terms of, for example, their social impact.

The end of 2022 was a turning point in the discussion around AI in education when generative AI hit the market. Seemingly, the educational sector felt the shockwaves of this disruptive technology the most, with concerns around intellectual property and academic dishonesty in the teaching and learning environment. To date, many still strategize on how to deal with LLMs such as ChatGPT within the context of education. What has become clearer with the passing of time is that their elimination from academic activity seems impossible. So, if they cannot be eliminated, can they be leveraged to promote learning? This remains a discussion for which there are no definitive answers yet.

### *Large language models in education*

ChatGPT, an LLM, is a natural language processing model that was launched by OpenAI in November 2022 (Qadir, 2023). Its architecture is based on that of GPT (Generative Pre-Trained Transformer), originally purposed for language generation and summarisation, and it can generate new content, in a conversational style, in real time. Large data sets from the internet were used to train ChatGPT which is what makes it so fluent in human-like conversation with a vast knowledge base (Qadir, 2023). But where does the term 'ChatGPT' come from? Chat refers to the conversation-like nature of the chatbot while GPT refers to the following: **Generative**: ability to generate novel text; **pre-trained**: trained on large datasets from the internet; and **Transformer**: GPT uses the transformer architecture which is a sequence-to-sequence neural network specially adopted for general purpose language modelling (Kamalov et al., 2023).

Preparing students for modern life, one where AI features and where AI literacy could be an advantage to graduates, requires that higher education systems seek the active implementation of approaches that may foster this preparation (Shabunina et al., 2023). If educators wish students to discover their own initiative and creative potential then it is necessary that conditions conducive to these expectations are created, which is possible through the integration of AI technologies in educational programmes (Shabunina et al., 2023). As educators, we must be aware of both the positive and negative aspects of this technology. In Table 2, Shabunina et al. (2023) offer a SWOT analysis as compiled through a review of literature exploring the potential/pitfalls of LLMs in education:

Table 2: *SWOT analysis of ChatGPT in the context of its current (or potential) application in education (Shabunina et al., 2023)*

**SWOT Analysis**

| | |
|---|---|
| *Strengths*<br><br>Enhanced learning experience through meaningful interaction with the LLM 24/7 availability.<br>Adaptive learning, meaning the LLM can respond to the level of knowledge of each individual student.<br>Generates plausible and credible responses. | *Weaknesses*<br><br>No human element.<br>Limited domain experience due to available training data.<br>Inability to scrutinise credibility of data post 2021.<br>Decline in higher order thinking skills of students. |
| *Opportunities*<br><br>Supplemental learning tool.<br>Provide challenging learning.<br>Individual learning paths.<br>Quick access to knowledge.<br>Decreasing educator workload by automating various teaching tasks. | *Threats*<br><br>Over-dependence on technology.<br>Plagiarism in education.<br>Privacy and security risk. |

Table 2 offers an overview of the current position of LLMs in education. This draws attention to the need for realistic expectations and sober vigilance regarding the integration of LLMs in the classroom. Both educators and students can benefit from LLMs (Mahapatra, 2024). For educators, LLMs can assist in creating course outlines, presentations, setting of assessments and formative quizzes, while for students it can be useful for solving questions, writing reports, generating code and obtaining feedback on work they have done (Mahapatra, 2024; Qadir, 2023). LLMs undoubtedly have great potential for students as they have been shown to have utility in both the learning of new material and preparing for assessment (Mahapatra, 2024).

The integration of LLMs in education is clearly a double-edged sword, but there are strategies educators can employ to mitigate the bad while incorporating the good. Table 3 summarises the aspects and strategies suggested to educators (Mahapatra, 2024).

Table 3: *Strategies in response to plagiarism concerns (Mahapatra, 2024)*

| **Aspect** | **Strategies** |
|---|---|
| Task design | Include questions that require diagrams in the answer as it is difficult for LLMs to generate these diagrams with accuracy. |
| | Use questions that require analysis. |
| Identification of AI writing | Plagiarism detectors could not detect AI originated text, but AI detectors did. |
| | Checking references as many references are merely fabricated. |
| Institutional policy | Establish anti-plagiarism guidelines |
| | Provide students with education on academic integrity. |

The workaround suggestions are not unrealistic as the strategies are implementable with little extra effort. Changing learning tasks to be more analytic and diagram-based will allow students to incorporate LLMs but make the copy-paste approach more difficult, as LLMs do not perform these tasks well. Although digital-free assessment formats might feel like a step backwards, this is a strategy that can be employed for certain assessments, such as final summative written assessments. Where students submit reports, AI detectors such as Quillbot can be used, as this platform has the capability to detect AI writing (Quillbot, 2024). Spot checking some of the references provided by students in their text can also be done to confirm if the references are real or fabricated, as ChatGPT tends to generate false references. Educating students by providing them with guidelines how to use LLMs and how to present their work can also help reduce plagiarism.

Chatbots can offer valuable support as they provide personalised assistance outside of formal class meetings, providing feedback and formative assessment for each student (Baskara, 2023). An attractive feature of an LLM is its ability to provide feedback in complex areas such as argumentation and critical thinking (Pendy, 2023), two very important skills to any engineering graduate. This kind of interaction will have to be facilitated through well-designed prompting, otherwise the LLM will simply provide answers to student questions, detracting from higher order thinking skills development as mentioned in Table 2.

LLMs and other AI applications can be useful in embedded systems education but not without well thought-out instructional design by the educator. To better understand the potential of LLMs in embedded systems education, the challenges faced by educators and students within the embedded systems education space is explored in the next section.

### *Embedded systems education*

Embedded systems can be described as a subject domain that includes computer science, automatic control and electrical engineering (Grimheden & Törngren, 2005a). A Swedish study found that industry is concerned with functional legitimacy, meaning that engineers working in the field need to be capable of designing and implementing embedded systems (Grimheden & Törngren, 2005a). Industry wants engineers who can solve problems with implementable solutions. Embedded systems design problems are complex, requiring the student to integrate software and hardware to develop a solution for the given design problem (Ibrahim et al., 2014). Figure 1 illustrates the methodology used for embedded system design. So how should embedded systems engineering students be educated to meet the requirements of industry?

Various pedagogical approaches have been developed by educators to facilitate the teaching of embedded systems in higher education institutions to close the divide between the skills taught at academic level and the skills required by industry (Ibrahim et al., 2015). Some of these approaches include software-oriented, hardware-oriented and codesign-oriented approaches, as described by Ibrahim et al. (2015). A large component of embedded systems design is the software development portion, which in itself is a challenge as learning to programme is complex and perceived by students as difficult, fraught with barriers (Becker et al., 2023). Learning to programme requires understanding of various concepts that build on one another. For a student to progress in programming, there are multiple threshold concepts that influence their overall understanding of the subject. These concepts are crucial as they contribute to how students move forward in their learning (Kallia & Sentance, 2017). The consequence of this is that students often tend to avoid programming fields as possible career paths (Suliman & Nazeri, 2024).

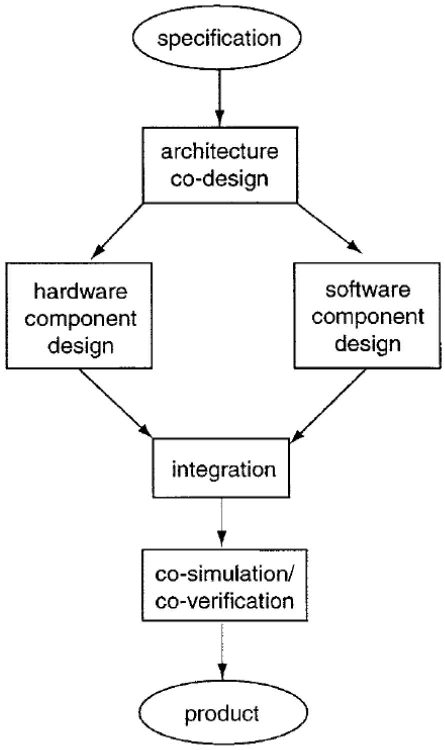


Figure 1: *Methodology for embedded systems design (Wolf & Madsen, 2000)*

The reality is that embedded systems design is far more complex than merely executing software on small computers. At its core, an embedded system has to interact with the physical world which is the main contributor to the complexities of embedded systems design (Fernandes & Machado, 2007). The challenges faced in embedded systems education can be categorised as student-related, lecturer-related and course-related (El-Abd, 2017). Figure 2 (overleaf) provides an overview of the challenges faced in teaching and learning embedded systems.

Embedded systems education has many challenges to overcome. The approach most employed by educators is the do-it-yourself, bottom-up approach. This gradually introduces the concepts of embedded systems in stages, while giving students the opportunity to implement these concepts practically for themselves (Ibrahim et al., 2014). Grimheden & Torngren (2005b) argue for teaching embedded systems courses through in-depth focus on topics. Also, didactic analysis finds that embedded systems have a thematic identity (themes specific to the domain of embedded systems). Grimheden et al. (2005) hold that educators should focus on teaching practical, real-world scenarios, as opposed to focusing on theoretical aspects alone. In other words, embedded systems education should emphasise the unique characteristics of the larger themes (hardware and software selection, real-time computing, specific functionality and real-world applications) that it forms a part of, and students should learn how to design systems that work according to these larger themes. The task for educators and students alike when teaching and learning embedded systems design is no small feat, and the advancement of AI in the current era requires that educators rethink classroom practices.

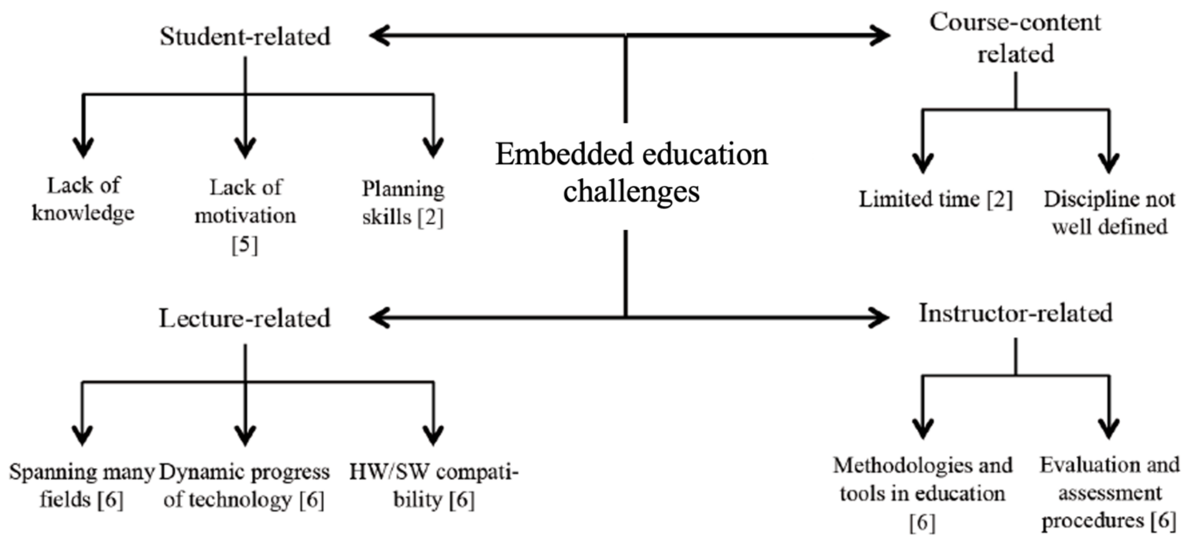


Figure 2: *Embedded education challenges (El-Abd, 2017)*

As educators, it is our responsibility to explore all possible means to better equip future graduates. To refer to the fertile soil metaphor used earlier, Figure 2 indicates the potential for growth. Addressing student related issues, LLMs can be useful in the 'lack of knowledge' area as they provide rapid access to knowledge of hardware and software related to embedded systems applications and the ability to provide in depth explanations on embedded systems-related content (Englhardt et al., 2023). Course-content challenges such as limited time to work through the syllabus and the undefined nature of the discipline can also benefit from the use of LLMs. The time constraints created by an academic semester place pressure on students to work through the course syllabus. Needing to cover complex topics and gain understanding through practical experiments adds to the cognitive load of students. Here LLMs can be of assistance as they have been shown to improve productivity in novice programmers and embedded systems designers (Englhardt et al., 2023). Although the time constraints are not removed, LLMs might assist students in being more productive and perhaps orientating them more quickly in the discipline.

Perhaps it is time to ask a new set of questions regarding what the code development process should look like. Should developers in the new AI era solely rely on their own skills or is AI-augmented development an acceptable way to learn?

### *The utility of AI in programming and embedded systems education*

Of the strategies proposed in Table 3 for implementing LLMs in the classroom, the most productive seems to be task design. To design learning activities that account for the use of LLMs, while being aligned with the goals, assessment methods and outcomes of the subject, can lead to the development of optimal learning environments (Loughlin et al., 2021).

An article published by the University of Reading discusses that when learning environments align learning outcomes, teaching and learning methods and the assessment tasks, students' learning experiences are bolstered, deep learning is promoted, motivation is boosted and mental health improved (Centre for Quality Support & Development, 2024). Thoughtful design of the learning environment will have to be done by educators, if the integration of LLMs in the classroom is to succeed. Teaching embedded systems in conjunction with LLM use by students is hardly known, and when educators rely on the trusted principles of constructive alignment, their new classroom approaches can be implemented with confidence. It need not be a mere knee-jerk reaction to the presence of LLMs, but a thoughtful, deliberate response that assists students to overcome learning barriers through various AI-enabled technologies.

AI code generation tools are fairly accessible to the general public through applications such as OpenAI Codex, DeepMind, AlphaCode and Amazon CodeWhisperer (Becker et al., 2023). These systems have the potential to improve productivity in programming, but they do not come without challenges. Becker et al. (2023) illustrate some of these by asking ChatGPT to elaborate on the opportunities and challenges presented in using automated code generation in education. The response of ChatGPT was:

> There are both educational opportunities and challenges presented by automated code generation tools. On the one hand, these tools can help students learn to code more quickly and efficiently. On the other hand, they can also lead to a false sense of understanding and proficiency, as students may become reliant on the tools to do the heavy lifting for them. Additionally, automated code generation tools can sometimes produce code that is difficult to read and understand, making it more challenging for students to debug and troubleshoot their programs. (Becker et al., 2023, p. 501).

Becker et al. (2023) report that integrating AI code generation tools in programming education simplifies the adoption of new codebases, reduces difficulties with context switching for experienced users, and makes programming more accessible to novices. Specific to learning, AI technologies might provide exemplar solutions to help students assess their own work, provide a variety of solutions to assist students to identify multiple ways to solve a problem, and review code to help find errors in it (Becker et al., 2023). For anyone in programming, regardless of the level of experience, these are helpful resources. The fact that these technologies can generate exercises and explain code while providing examples can greatly assist a new programming student in becoming comfortable with and learning new concepts of coding by reducing the cognitive load (Becker et al., 2023).

To turn to the challenges specific to embedded systems: the key challenge in an embedded system design is the requirement for interaction between the physical and digital environments (Englhardt et al., 2023). Code generated for an embedded system needs to be repeatedly verified through trials. The question is whether LLMs can generate code able to integrate with actual hardware required to interact with a real environment. Hardware knowledge and an understanding of various components are crucial when writing code in embedded systems, as different devices subscribe to different protocols. For example, although the I2C communication protocol has set standards, different devices such as port expanders, real time clocks and RAMs all have very specific procedural requirements to exchange data with the microcontroller. The implication here is that even if LLMs are able to generate code, they should also understand the functional requirements for the specific devices used in the design, or the code becomes meaningless. A study by Englhardt et al. (2023) set out to test the ability of LLMs to develop embedded systems through 450 experiments testing multiple LLMs (GPT-3.5, PaLM2 and GPT-4). Their findings can be summarised as follows:

1. LLMs are able to generate syntactically correct and semantically meaningful code from high level task descriptions.

2. Hardware specifications: they can generate register-level drivers, I2C interfaces and LoRa communication code, showing that they can successfully navigate hardware device requirements.
3. They provide context-specific debugging advice for hardware by providing clarity on wiring.
4. Human-AI co-development works best as GPT-4 could only provide perfect end-to-end code 14% of the time. It is worth noting that the partially correct programmes still contained functional code with detailed comments and explanations on how to design the system.
5. User success rate for complex tasks was improved from 25% to 100%. Users with zero hardware or C/C++ experience could build a fully functional LoRa sensor transmitter and receiver in 40 minutes.
6. LLMs could provide useful suggestions to designers working on building a system (hardware, communication protocols, and coding techniques).
7. Prompting is crucial: prompts should be clear and include key system information to enable the LLM to develop appropriate solutions, taking cognisance of the technical nuances and the intended behavior. This aspect of the findings has clear implications for teaching embedded systems with the use of LLMs, as educators will need to provide scaffolding for students to learn how to prompt the AI effectively to perform learning tasks alongside LLMs.

What Englhardt et al. (2023) found is impressive. However, they highlighted some limitations and concerns: the LLMs can misunderstand tasks due to ambiguous prompts, making assumptions that are incorrect; they can 'hallucinate' and, as a result, produce incorrect details in their responses. They were also found to make unprompted modifications to code, which becomes an issue in systems where system resources are limited, and any unnecessary code consumes the limited capacity of the microcontroller. Potentials and pitfalls of LLMs in education need to be well understood if the novel connections made between embedded systems education and LLMs are to be meaningful. The following section discusses how educators might think differently about embedded systems educations as far as LLMs are concerned.

## Rethinking embedded systems education

This article aimed to explore some of the potential benefits and challenges of AI technologies in education, and future directions of how educators could respond to the integration of LLMs in embedded systems education. It did this by reviewing literature relating to the focal phenomenon. This section of the article will explore how the different features of these principles can potentially be utilised effectively by educators specializing in embedded systems.

The overarching goal of educators in embedded systems is to produce graduates who are equipped for the complex demands of a growing industry. Using constructively aligned teaching can contribute to this overall goal of embedded systems educators by producing quality learning outcomes (LOs) and student satisfaction (Biggs, 2014). Following the principles of constructive alignment (CA), embedded systems educators need to reassess the goals, learning activities, outcomes and assessment of a given module.

This article does not propose that an entire syllabus should be redesigned, but rather that the focus should be selected learning activities to gradually integrate LLM. While it may still be too early to know exactly how AI will impact knowledge and skills requirement for embedded systems graduates, it seems expedient for educators to pay attention to the possibilities that may open up for graduates with an ability to utilise these new tools. Thoughtful inclusion of LLMs in the curriculum will give students the opportunity to engage responsibly with this technology in carefully designed learning activities. These specific activities can be assessed to measure the degree to which the outcomes are met.

If we are to rethink the teaching and learning process of embedded systems, we must start by being clear about the goals we want to achieve to ensure the teaching and learning of the subject are in alignment. One of the main goals of embedded systems education is to get students to write code that will allow hardware to interact with the external world. It is here where many students struggle, because the process of developing code is complex and presents a formidable barrier (Becker et al., 2023; Ibrahim et al., 2015; Kallia & Sentance, 2017; Suliman & Nazeri, 2024). Englhardt et al. (2023) demonstrate that LLMs are capable of producing code that is matched to the intended hardware of an embedded systems, in the context where it is to be deployed. They can also provide recommendations regarding the connections of the hardware used. This functionality is of great use to students working on an embedded systems design problem.

The main goal of embedded systems education is to teach students how to solve a given problem through the development of software code that will allow hardware platforms to interact with the world in a specific way. The first challenge for embedded systems students is gaining a clear grasp of the problem at hand, and thereafter to break the problem down into sub-components to develop the code for hardware. This requires that students have a thorough grasp on the exact requirements of the hardware and how the sensor interacts with the external world, as the developed code must be specific to the given requirements and specifications of the hardware components used. This is the second challenge for students. They know the basics of programming but fail to cross the bridge where they must apply the coding principles to the specific context of the hardware, addressing the functional requirements of the task at hand. They fail in facilitating interaction between the microcontroller and the peripheral hardware, such as sensors, etc.

The potential application of AI technologies in this regard is notable. LLMs, when applied in the correct way, can contribute to both helping the student to solve the problem at hand and learn coding in the process. This is where thoughtful learning design is crucial.

Using formative learning activities that include LLMs to practice threshold concepts, as defined by Kallia and Sentance (2017) above, allows students to engage with the LLM as with a tutor. The LLM can generate exemplar pieces of code, provide explanations, and generate small tests for the student based on the concept and hardware-specific platform used. This will consider the specific level of knowledge for each individual student and contributes toward one of the main outcomes of embedded systems education: the ability to write syntactically and semantically correct code. The potential lies in that LLMs can support student learning in the specific hardware environment that the student is working in, be it AVR, PIC, etc. This is relevant because through experience of working with students for 20 years in higher education, I have found that this is where they struggle: developing functional, hardware-specific code.

Traditionally, students were given learning activities where the goal was to produce code that, when implemented, would deploy correctly, causing the hardware to function as intended. The artefact would then be assessed through a demonstration where the student presented the work. The focus would be the functionality of the overall system. Due to the adoption and accessibility of LLMs by students, this traditional approach needs to be revisited. Students use LLMs, and the traditional assessment model might not be adequate in measuring the contribution of the student in the learning activity. This article proposes an alternative view by suggesting a more agile process, moving beyond a simplistic view on plagiarism detection and punitive actions during assessment, towards reformulating assessment activities that focus on the entire learning process.

This reformulation starts with the designing of LLM-enhanced embedded systems learning activities. Designing learning activities that include the use of LLMs but shifting the cognitive learning to the higher levels in Bloom's taxonomy, can mitigate some of the concerns around LLMs in the classroom. When the objective of the learning activity is not just an answer, but an exploration, plagiarism and AI generation concerns become less worrisome. The focus needs to shift from assessment alone towards the entire learning process, one that is tailored to respond to the varied learning requirements of individual students. This article proposes alternative approaches to overcome the challenges within embedded systems education in the AI era.

### *Suggestions of alternative approaches for the embedded systems classroom where LLMs feature*

The following examples serve as suggestions regarding how educators can adapt classroom activities.

1. After introducing a topic on microcontroller programming – for example, the use of internal timers – the educator gives students a written piece of code that draws together the elements taught in the completed section of work. The task of each student is then to firstly, identify the portions of the programme that they do not understand and, secondly, to start engaging with an LLM. The focus of the chat with the LLM will be the problematic elements for the individual student. This will be different for each student as each has varying levels of understanding and knowledge. The intended outcome of this learning activity, which is assessed, is for students to reflect and report on how they managed develop a better understanding of the elements that troubled them. The final portion of the learning activity is for the student to implement the given code in hardware.

This recommendation draws on the accepted approaches for teaching and learning embedded systems but revises it through the lens of AI augmentation. In this activity, the student is not assessed for just producing a practical demonstration, but attention is given to the process, which will be different for each student. The student has the unique opportunity to engage in a learning activity that is truly adaptive and matched to him/her, but the goal (practical demonstration) will look the same for the entire class.

2. In a project scenario where students, for instance, are required to develop a system where the microcontroller needs to read data using the I2C protocol and then display the data on a 16x2 liquid crystal display, the same principles can be applied without compromising the ILO. Here the educator can request that students write a detailed explanation of their code, and the role of the LLM in the practical experiment or learning task. Students still need to demonstrate the system in action, but the educator now asks questions regarding challenges they faced and how they overcame these challenges. The educator could require a student to implement a modification in the experiment as this would demonstrate their ability to adapt, indicating the level of their understanding. The assessment rubric could be designed with criteria for both the process and the result, allocating weight to how the student co-created a solution with AI, and demonstrating understanding and the ability to deliver a functional embedded system.

These two short examples are the first steps towards developing a roadmap that educators can use to incorporate LLM-enhanced learning activities, but with emphasis on the process, not only the outcome. It should be noted that the activities should still be constructively-aligned with the outcomes.

### *Focus assessment on the entire process, rather than just the outcome*

Typically, embedded systems students will perform smaller experiments, such as provided in the first example above. They will also design and build a project, as in the second example, and take formative and summative tests through the semester. Triangulation of all three – experiments, project and written tests – has the potential to provide the educator with a portfolio overview of the competence of each student, and if the ILOs have been met, the threat of over-reliance on AI on the part of students will be minimised.

As educators spread the focus across the entire process, a more flexible view is necessary to account for the presence of LLMs while still measuring the understanding of the students. It is part of the responsibility of the embedded systems educator to prepare students for a life where AI features, giving them the tools to improve their productivity by exposing them to AI in the embedded systems domain.

Implementing AI-enhanced learning activities such as these can lean into the strengths of and opportunities offered by AI, while mitigating some of the weaknesses and threats, such as plagiarism and over-reliance on LLMs, as discussed by Shabunina et al. (2023). This confirms the findings of Mahapatra (2024), offering task design as a strategy to work around the concerns regarding LLMs in education. Furthermore, this strategy is accessible to educators where resources, funding and capacity is limited: students merely require access to computers and an internet connection for interacting with open source LLMs such as ChatGPT, Copilot and Gemini, to name a few that have been found capable of embedded systems development and coding.

Embedded systems applications have become more advanced over the years. Exposing students to these increased complexities in technology is challenging due to time constraints in an academic semester. Using exemplifying selection, meaning to identify fewer important topics, allowing for more comprehensive exploration, as suggested by Grimheden and Törngren (2005a), is a good pedagogical approach to teach embedded systems. Using LLMs to assist in this task of teaching a topic in-depth can be beneficial, as the LLM can provide the support to students in times of private study. LLMs have demonstrated excellent ability to navigate the requirements of complex hardware components such as real time calendar clocks, I2C technologies, etc., making them well suited to be study partners and tutors.

The integration of AI technologies, particularly LLMs, in embedded systems education presents a promising avenue for enhancing the learning experience. The potential of these technologies to support students in overcoming complex challenges in teaching and learning in embedded systems is significant. However, it is crucial that educators approach this integration with a clear understanding of the goals they aim to achieve and the student outcomes they seek to measure. By constructively aligning learning activities with these goals and outcomes, educators can ensure that the use of LLMs enhance, rather than detract from, the learning process. Furthermore, the shift in assessment methods, from evaluating the final product to assessing the learning process, can provide a more accurate measure of student understanding and skills development. This approach not only prepares students for a future where AI is ubiquitous, but also equips them with the skills necessary to navigate and contribute to this future effectively. As we move forward, it will be essential to continue exploring and refining these strategies to ensure that the integration of AI technologies in education is done in a way that truly benefits students and educators alike.

## Conclusion

This conceptual article drew on multiple strands of literature pertaining to embedded systems education and its associated challenges. The intention was to provoke the reader to reconsider teaching and learning approaches of embedded systems in the presence of AI technologies such as LLMs. It offered practical examples of how educators in embedded systems education can use LLMs instead of resisting them or constantly policing students for unoriginal work.

Novel connections were identified between LLMs and embedded systems education. Suggestions were developed that could potentially guide educators, offering an alternative view on how to incorporate or manage the disruption caused by LLMs.

The contribution of this article is to challenge educators to apply some flexibility to their teaching and learning approaches, drawing on LLMs as an ally, an instrument that could potentially change the way in which students learn and become competent in embedded systems design. Based on how rapidly AI technologies are evolving, future embedded systems students could simulate complex embedded systems scenarios, allowing for risk-free experimentation in a virtual environment. Future LLMs could potentially analyse vast amounts of industry data to predict emerging trends and technologies, keeping the curriculum relevant. This is indeed an exciting frontier that could transform classrooms and make education more personalised, engaged and effective.

## References

Baskara, F. X. R. (2023). Chatbots and flipped learning: Enhancing student engagement and learning outcomes through personalised support and collaboration. *IJORER: International Journal of Recent Educational Research, 4*(2), 223-238. https://doi.org/10.46245/ijorer.v4i2.331

Becker, B. A., Denny, P., Finnie-Ansley, J., Luxton-Reilly, A., Prather, J., & Santos, E. A. (2023). *Programming is hard - or at least it used to be* [Conference presentation]. Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1. https://doi.org/10.1145/3545945.3569759

Biggs, J. (2014). Constructive alignment in university teaching. *HERDSA Review Of Higher Education*, *1*(3). https://www.tru.ca/__shared/assets/Constructive_Alignment36087.pdf

Bloom, B. S. (1968). Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment*, *1*(2), n2. https://eric.ed.gov/?id=eD053419

Centre for Quality Support & Development. (2024). Aligning teaching, learning, and assessment to learning outcomes. University of Reading.

El-Abd, M. (2017). A review of embedded systems education in the Arduino age: Lessons learned and future directions. *International Journal of Engineering Pedagogy, 7*(2), 79-93. https://www.learntechlib.org/p/207404/

Englhardt, Z., Li, R., Nissanka, D., Zhang, Z., Narayanswamy, G., Breda, J., Liu, X., Patel, S., & Iyer, V. (2023). Exploring and characterizing large language models for embedded system development and debugging [Conference presentation]. CHI EA '24 Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. Article 150, 1-9. https://doi.org/10.1145/3613905.3650764

Fernandes, J. M., & Machado, R. J. (2007). Teaching embedded systems engineering in a software-oriented computing degree [Conference presentation]. 2007 37th Annual Frontiers in Education Conference – Global Engineering: Knowledge Without Borders, Opportunities Without Passports. https://doi.org/10.1109/FIE.2007.4417949

Grimheden, M., & Törngren, M. (2005a). How should embedded systems be taught? *ACM SIGBED Review*, *2*(4), 34-39. https://doi.org/10.1145/1121812.1121820

Grimheden, M., & Törngren, M. (2005b). What is embedded systems and how should it be taught?—results from a didactic analysis. *ACM Transactions on Embedded Computing Systems*, *4*(3), 633-651. https://doi.org/10.1145/1086519.1086528

Guskey, T. R. (2007). Closing achievement gaps: Revisiting Benjamin S. Bloom's "Learning for Mastery". *Journal of Advanced Academics*, *19*(1), 8-31. https://doi.org/10.4219/jaa-2007-704

Holmes, W., & Tuomi, I. (2022). State of the art and practice in AI in education. *European Journal of Education*, *57*(4), 542-570. https://doi.org/10.1111/ejed.12533

Ibrahim, I., Ali, R., Adam, M. Z., & Elfidel, N. (2014, 2014-12-01). Embedded systems teaching approaches & challenges [Conference presentation]. 2014 IEEE 6th Conference on Engineering Education (ICEED). https://doi.org/10.1109/ICEED.2014.7194684

Ibrahim, I., Ali, R., Zulkefli, M., & Elfadil, N. (2015). Embedded systems pedagogical issue: Teaching approaches, students readiness, and design challenges. *American Journal of Embedded Systems and Applications*, *3*(1), 1-10. https://doi.org/10.11648/j.ajesa.20150301.11

Jaakkola, E. (2020). Designing conceptual articles: Four approaches. *AMS Review*, *10*(1), 18-26. https://doi.org/10.1007/s13162-020-00161-0

Kallia, M., & Sentance, S. (2017). Computing teachers' perspectives on threshold concepts [Conference presentation]. Proceedings of the 12th Workshop on Primary and Secondary Computing Education. https://doi.org/10.1145/3137065.3137085

Kamalov, F., Calong D.S., & Gurrib, I. (2023). New era of artificial intelligence in education: towards a sustainable multifaceted revolution. *Sustainability, 15*(6), 12451. https://doi.org/10.3390/su151612451

Loughlin, C., Lygo-Baker, S., & Lindberg-Sand, Å. (2021). Reclaiming constructive alignment. *European Journal of Higher Education*, *11*(2), 119-136. https://doi.org/10.1080/21568235.2020.1816197

MacInnis, D. J. (2011). A framework for conceptual contributions in marketing. *Journal of Marketing*, *75*(4), 136-154. https://doi.org/10.1509/jmkg.75.4.136

Mahapatra, S. (2024). Impact of ChatGPT on ESL students' academic writing skills: A mixed methods intervention study. *Smart Learning Environment*, *11*(9). https://doi.org/10.1186/s40561-024-00295-9

Nakutis, Ž., & Saunoris, M. (2010). Challenges of embedded systems teaching in electronic engineering studies. *Elektronika Ir Elektrotechnika*, *102*(6), 83-86. https://eejournal.ktu.lt/index.php/elt/article/view/9360

Pendy, B. (2023). Artificial intelligence: The future of education. *Jurnal Indonesia Sosial Sains,4*(3), 2003-2013.

Qadir, J. (2023). Engineering education in the era of ChatGPT: Promise and pitfalls of Generative AI for education [Conference presentation]. 2023 IEEE Global Engineering Education Conference (EDUCON), Kuwait, Kuwait. https://doi.org/10.1109/EDUCON54358.2023.10125121

Quillbot. (2024). *AI Detector*. https://quillbot.com/ai-content-detector

Sangiovanni-Vincentelli, A. L., & Pinto, A. (2005). Embedded system education. *ACM SIGBED Review*, *2*(4), 5-14. https://doi.org/10.1145/1121812.1121815

Selwyn, N. (2022). The future of AI and education: Some cautionary notes. *European Journal of Education*, *57*(4), 620-631. https://doi.org/10.1111/ejed.12532

Shabunina, V., Sarancha, V., Maslak, V., Shevchenko, O., & Tur, O. (2023). Educational potential of ChatGPT: Teaching tool for students' competencies development [Conference presentation]. 2023 IEEE 5th International Conference on Modern Electrical and Energy System (MEES), Kremenchuk, Ukraine. https://doi.org/10.1109/MEES61502.2023.10402380

Suliman, A., & Nazeri, S. (2024). The effectiveness of teaching and learning programming using embedded system [Conference presentation]. 2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012)

Wolf, W., & Madsen, J. (2000). Embedded systems education for the future [Conference presentation]. *Proceedings of the IEEE*, *88*(1), 23-30. https://doi.org/10.1109/5.811598