Unit 6: Microcontroller Systems

Delivery guidance

Approaching the unit

This unit will inspire and challenge learners to use the knowledge and skills they develop and create an inventive microcontroller system solution to a given problem. Learners have the opportunity to find out about the important part that microcontroller systems play in the world around them and in the 'Internet of Things' (IoT). This unit cultivates learners' understanding of the types of microcontroller hardware and the diverse range of peripherals available that can be connected to this hardware. It also teaches learners how to control peripherals through programming. There is an opportunity within the delivery of this unit to develop a varied range of practical activities – selecting a microcontroller for various common problems, identifying pertinent peripherals and physically programming the systems to achieve their own desired solutions.

Learners will develop incremental programming skills to enable microcontrollers to respond to several different types of inputs and produce relevant and expected, sometimes imaginative, output/s through an assortment of interconnected devices. They will systematically plan and devise their own coding solutions, methodically testing them under simulation and upon individually constructed microcontroller systems. These skills, an essential part of the growing research and development field of the IoT, will inspire learners to design and create tailored solutions.

This unit, like the other mandatory units, could be delivered in a specialist context such as aeronautical, manufacturing or electrical and electronic engineering. For example, a centre wanting to deliver the mandatory units in an electrical/electronic context could explore a range of basic electrical products such as an iron, or a food blender to explore design and manufacturing processes applied. However, care must be taken to ensure learners are prepared for the task-based internal assessment that is set by Pearson.

You can involve local employers in the delivery of this unit if there are local opportunities to do so.

Assessment model

Learning aim	Key content areas	Assessment approach
A Produce a technical specification and a design for a microcontroller system to solve a problem	**A1** Control hardware **A2** Input devices **A3** Output devices **A4** Selecting hardware devices and system design	This unit will be assessed through a Pearson Set Assignment. Learners will be required to complete a set task in a limited time period. Learners' work will be submitted in the form of an electronic task booklet and an audio-visual file (the audio-visual file will be no longer than three minutes), which will be assessed by centre staff using the assessment criteria in this unit.
B Develop and test a software and hardware solution for a microcontroller system to solve a problem	**B1** Assembling and operating a microcontroller system **B2** Programming techniques **B3** Coding constructs **B4** Structured program design B5 Number systems	
C Project manage the development lifecycle and present the operation of a microcontroller system to solve a problem	**C1** Development processes **C2** Documentation **C3** Project log	

Assessment guidance

This unit is internally assessed with learners completing tasks for a Pearson Set Assignment Brief. The assessment will require learners to follow the processes as developed during delivery. This highlights the need for several practical examples for learners to refer to and work through. The assessment will place the learner in the role of a microcontroller design engineer with a client requiring the design and construction of a real-life product to meet a need. Learners will need to give evidence of understanding each topic and associated topic in order to realise the finished product.

Preparation of learners is essential, and it is important that they have access to all tools integrated components and components that they will need to construct the final product. Familiarity with the integrated development environment is crucial, as is understanding downloading and coding errors, to ensure that learners can develop a prototype which may not meet the full client brief but gives evidence of each topic.

During the delivery of this unit, one of the key aspects is the ability of learners to construct a prototype microcontroller system. As assessors, you are encouraged to develop the assessment of this unit in conjunction with *Unit 19: Electronic Devices and Circuits* to reinforce learners' skills in this area. Another key aspect is the use of the chosen integrated development environment, coding, downloading and error checking programs. Learners develop a good grounding in this area by developing and expanding simple programs to deliver gradually more complex outcomes.

Delivering the topics

Before beginning this unit, it is important that learners are issued with a suitable toolbox that contains the relevant tools, integrated circuits (ICs) and components that they will use when responding to briefs and completing practical activities.

Topic A1 covers the underpinning theory and knowledge of the internal make up of a microcontroller IC.

First, deliver this aspect by selecting a specific microcontroller that you will issue to each learner, together with a breadboard. Following learner research for the datasheet, learners could identify each pin and its primary/secondary function. You could give practical examples for each type of possible input and output, e.g. analogue input (temperature) and output via a single port to an LED array. This could be followed by the internal hardware specification and functions available within the chosen microcontroller, memory types, method of programming and how this can affect the cost and type of microcontroller required. This is an opportunity to begin a discussion on selecting the correct microcontroller for a given problem. Introduce practical tasks by using the selected microcontroller, building a simple power supply for the required operating voltage and powering up the microcontroller.

- This is also the ideal topic to introduce the integrated development environment (IDE) that you intend to use and how to obtain the software, whether it is open source or shareware.
- This gives an opportunity at this stage to map across with skills and knowledge from *Unit 19: Electronic Devices and Circuits*.

Topic A2 identifies many of the input peripherals, their operation and the characteristics that you can use with microcontrollers.

At this stage, it is important to have samples of the different possible input devices that can be used with your chosen microcontroller, including datasheets, so learners can get a feel for the size, shape and pin outs for the device. As you progress through this topic, have working examples of each input device, a simple program and an output through a simple array of eight lights emitting diodes (LEDs). This will enable you to hold several question and answer sessions about existing real-life applications for the related input device, which learners can also investigate. This can be a physical or simulated example – but, a physical example is preferred, as it gives learners an input device they can physically adjust and see instant results.

Using each device leads into discussions about types of interface and practical examples of each type of interface and real-life applications.

- This gives an opportunity at this stage to map across with skills and knowledge from *Unit 19: Electronic Devices and Circuits* by wiring eight LEDs directly to a port.
- This is the ideal topic to reflect on the simplicity of some of the programs that control the input devices and give the output to the light emitting diode (LED) array.
  By downloading and running a simple program to light, the LEDs give learners examples of the structure and layout of any program that they will develop later in the unit.

Topic A3 identifies many of the output peripherals, their operation and the characteristics that you can use with microcontrollers.

Again, at this stage, it is important to have samples of the different possible output devices that can be used with your chosen microcontroller, including datasheets, so learners can get a feel for the size, shape and pin outs for each of the devices. As you progress through this topic, have a working example of each output device, a simple program and an input through an analogue device – a potentiometer is typically the most common. This is also an opportunity to have input devices connected through different interfaces, which links directly back to topic A2. This will enable you to hold several question and answer sessions with learners about existing real-life applications for the related output device, which learners can also investigate. They can also develop imaginative ideas for potential uses. The example you use can be physical or simulated, but a physical example is preferred as it gives learners an output from an adjustable input that they can observe physically.

- This gives an opportunity at this stage to map across with skills and knowledge from *Unit 19: Electronic Devices and Circuits*. Connect a potentiometer to analogue input.
- Again, this is the ideal topic to reflect on the simplicity of some of the programs that control the input devices and give the output to the light emitting diode (LED) array. This gives learners examples of the structure and layout of any program that they will develop later in the unit.

Topic A4 enables learners to investigate different problems that require a microcontroller system to enable application. Learners will need to select and justify the selection for each of the different problems chosen.

It is important that learners understand the importance of working systematically through any problem that may have a microcontroller system solution by:

- investigating and identifying potential suitable options for inputs
- investigating and identifying potential suitable options for outputs
- researching aesthetics, cost and usability and
- reviewing the environment parameters of use.

It is important that you have a minimum of three complete examples to work through at this stage. Some groups of learners can grasp the concepts quickly and therefore, examples of varied complexity are required. Detailed discussions can develop about the advantages and disadvantages of each type of potential input and output for the desired solution. The example problems also allow learners to review and research different families of microcontroller that can be used as a potential solution and the reduction of costs and usability.

Topic B1 builds on the teaching of the previous learning aim and links each of the programming environments to the previous topics. Achieve this by building the microcontroller system from the ground up – creating the power supply, building an output of eight light emitting diodes, having input from a potentiometer (this could be already completed in another unit), the connection of an integrated development environment to the system, downloading a simple program to light the row of light emitting diodes, or manipulating these through the potentiometer. Ensure learners understand how to use typical electronic tools safely and appropriately and that they are able to test electronic equipment.

Topic B2 looks at programming techniques and should initially be a reflection of the programs downloaded to visually identify types of inputs and outputs and how they can be easily manipulated. Learners look at how to use a programming environment and coding practices. Revisit how learners connected to the integrated development environment (IDE) and compiled, downloaded and ran programs on the microcontroller. At this stage, identify the rules, folder structure, libraries, declarations, chip setup and syntax for any of the programs issued. It can also be worthwhile, depending on the ability of learners, by getting them to write the code into the integrated development environment directly during topic A so they encounter the common pitfalls early in the unit. Many of the compilation and downloading issues will then have been encountered during the delivery of topic A and typically peer learning will develop as learners share problems encountered and share solutions. This enables more time to be spent on creating the program itself. These developed and given programs will also enable you to instruct learners on how to step through a program and to simulate the full microcontroller system, if software is available.

Topic B3 looks at coding constructs and covers the use of embedded C as the most common programming language used for microcontroller systems. It is important at this stage to identify any books and/or have several examples that will help and give examples of coding for the majority of the aspects required for the unit. At this stage as you introduce each new coding construct, it can be related directly back to topic A and the peripherals examined.

This an opportunity to revise topic A and create discussions on how each peripheral can be controlled in order to get learners to research and investigate types of communication, control mechanisms and modes of operation for peripherals, such as light emitting diodes (LEDs) and the coding necessary to operate these arrays. To structure the learning more clearly from this point, it is best to select one input and a single defined output to deliver content on program flow, logic and arithmetic variables. This will enable the refinement of the initial program, developing libraries, calling subroutines, using delays and other subroutines to develop logic and arithmetic structures to aid learner understanding. The continual development of the program enables learners' understanding and the reinforcement of tools available within the integrated development environment which is being used.

In order to cover topic B4, you should supply the pseudo code, flow chart and/or decision chart for some of the initial basic programs and create a class discussion on the comparisons and links between the finished code and the program design. It is useful at this stage to get learners to write the design for some of the other basic programs already developed to understand the structure.

At this point, set a standard programming task such as incrementing a count that is outputted to a light emitting diode array. Learners develop the solution using the structure taught, creating the code, including libraries and calling suitable subroutines. Learners should again consistently use peer learning to ensure that they each get a complete running program.

In topic B5 (number systems), it can be difficult for some learners to understand unfamiliar mathematical concepts. One of the best methods for this topic is to incorporate it with B2 or teach in advance. The best way to begin is by breaking down decimal numbers into Base 10 and then identifying how Binary (Base 2) is structured and how a decimal can be written in binary. Even though Octal (Base 8) is not required, it is useful to introduce in order to explain to learners that a decimal number can be represented in any Base number.

Delivery of the methods of conversion can be enhanced by a number of apps and games available online for learners to develop a quick understanding of the conversion between binary and decimal. As learners develop their understanding of binary, it can be useful to incorporate hexadecimal for ease of programming if using embedded C.

In order to cover topic C1, you need to have a minimum of three complete projects for learners to work through and have a reserve for those learners who enjoy the additional challenge. The first should be a straightforward project that you guide learners through. The aim of this is to give them confidence that following the development process is the best way to ensure that the solution meets the client's needs. Ensure that they carry out an analysis of the brief, system design and program planning, system assembly and coding, and testing and operation.

If possible, it is important to link this learning outcome with *Unit 19: Electronic Devices and Circuits* so that learners develop the self-assurance required for the detailed construction of circuits for microcontroller systems being created from individual components and integrated components (ICs). It is still important if developing the finished system using an off-the-shelf development board, as learners still need to be able to identify the necessary components and peripherals to visualise the final solution.

Testing the system can be one of the most critical features of the process. It gives justification and analysis for parts of the project that the learner cannot complete or ensure full functionality for and is therefore a critical part of the delivery of this topic.

To deliver topic C2 it is best to have a template for learners to follow when creating the supporting documentation for any chosen solution and a completed example of the depth required for the assessment task. Make sure at this point that learners have a method to record their finished solution to include in the finished documentation.

In order to complete the project log for topic C3, learners will require a template to record their progress during the problem-solving task. Learners will need to practise recording their progress during and after each session in order to produce a project log that reflects on the progression and developments in each session and includes issues and solutions to any problems they encounter. They should also prepare for the following session with prioritised action points showing what they plan to do.

Getting started

This gives you a starting place for one way of delivering the unit. Activities are suggested to assist in preparation for the internal assessment task.

Unit 6: Microcontroller Systems
Introduction The delivery of this unit allows for input from local electronic and research and development engineering employers. As a result, there is an opportunity for employers to set real-life problems for learners to work through. Following the development process enables learners to cultivate transferable skills in problem- solving, creative development, independent learning and teamwork. The assessment will require learners to be able to plan a project and, more importantly, to schedule that project to ensure completion by the due date. The logical format of the essential content allows for a systematic understanding of the necessary skills to ensure learners can undertake the assessment task.
Topic A – Investigate typical microcontroller system hardware
● Topic A is broken up into several topics that should be delivered individually, building upon each aspect as you progress. Before beginning, it is important that learners have a suitable toolbox that includes a breadboard and basic tools. The contents can be expanded upon as the unit develops. ● Choose a microcontroller that suits the integrated development environment (IDE) used within your department and school. This can be either an externally purchased development board or an in-house developed solution. ● First, issue learners with the selected microcontroller, e.g. PIC 16FXXX, relevant basic datasheet and connect to a breadboard. If possible, use a component constructed 5V power supply developed through an associated unit to power the microcontroller and construct the IDE programming input. ● You can now direct learners in a research task on the internal architecture of the chosen microcontroller that will create an opportunity for class discussions and a Question and answer session. This allows learners to develop an understanding of the requirements for a microcontroller system, e.g. the number of ports needed, random access memory (RAM), hardware features and input/output capabilities for use in different applications. ● This topic requires extensive theory and discussions, therefore follow with an introduction to the features of the integrated development environment and the method of connecting to the microcontroller. Develop learner understanding by repeatedly opening the software and connecting to the microcontroller, identifying and correcting the common connection issues that exist. This is an opportunity to introduce some peer learning and guidance that will become essential as the unit develops. ● Topics A2, A3 and A4 allow group tasks and the teaching to overlap. Each group are given details of a device that may have one or multiple microcontroller systems within it. Learners can then take one microcontroller system and try to identify the potential inputs that there could be within the system, what the input is measuring and how it is being displayed (output). This allows learners to review real-life microcontroller systems, their component parts and how they have been constructed. ● It is important to define the inputs into either digital or analogue and create a discussion on the differences and what potentially can be measured by each type and how any output/s could be monitored. For example, why a fridge door light illuminates (type of input and output), how a

washing machine knows when to stop heating water, why the lights in corridors turn off if no one is there, burglar alarms, reversing sensors, etc.

- This allows learners to delve into topic A4 considering when and why you would use certain input or output devices, taking into account factors such as cost, style, accuracy and power requirements. For example, why have a liquid crystal display (LCD) output if the door of a washing machine is locked when a red-light emitting diode (LED) would suffice?
- This comfortably leads into interface requirements, how to convert an analogue input that can be understood by a microcontroller, signal conditioning and serial and parallel communications.
- It is essential that there are several examples of different types of input and output devices. Make sure that you can relate each device to a real-world application, both seen and unseen. Learners should be encouraged to conduct research for the relevant datasheet for each device with time taken in class to review the operational characteristics of the main input and output peripherals that can be used.
- Once learners have identified the main input and output peripherals, this will be the opportunity to further develop the integrated development environment (IDE), with circuit construction and by downloading simple programs to manipulate outputs based on digital or analogue input. For example, push button input to increment a count displayed (output) on light emitting diodes, 7-segment display or liquid crystal display (LCD). The LCD requires more structured programming if connected by learners and not part of a development board.
- Other examples are:
  - temperature sensor input that sounds a buzzer (output) at a certain temperature
  - light dependent resistor (LDR) input switches on a relay (output) that switches on a bulb, imitating a streetlight sensor
  - tilt switch input that switches off a relay (output) to cut power, imitating a forklift emergency power shut off
  - micro switch input that changes the speed of a motor using pulse width modulation (output).
- The output interfaces can be connected in various ways using serial or parallel communications or interfacing using I^2C.
- It is advisable to have a basic task sheet for each example that includes how to interface the input and output with the microcontroller, a schematic and basic code that can be developed. Small tasks to make minor iterations to the code will build learners' confidence as they develop through the next topics.
- The more examples that learners work through, the more confident they will become with the integrated development environment, resolving errors connected to the microcontroller, testing the code and preparing them for the assessment task.
- Some assessors give the code as .txt file to incorporate into the integrated development environment, others as a sheet of text for learners to type directly into the integrated development environment. The latter is recommended, as learners will encounter common typing mistakes and declaration and syntax errors as a result.
- Separate sessions should be set aside for 'output interface requirements' as this is typically a difficult aspect of the theory for learners to grasp.

Topic B – Programming techniques and coding

- This is the most vital part of the unit as most mistakes and failures to realise a microcontroller system are a result of poor coding and an inability to resolve issues by adhering to a structured plan for the code and a detailed testing schedule.
- Learners extend the previous development work depending on the equipment and development platform used. For example, this could include connecting input and output components on a breadboard, connecting input/output modules or identifying appropriate devices on a pre-assembled prototyping board. For this delivery guide, it is assumed that embedded C using the MPLAB® IDE (MPLAB® C) is used. A good way to start is with topic B5 (Number systems), so that learners can identify directly with the coding that they have previously written.
- Topic B5 will be straightforward for any learner who is undertaking an electronic or computing-themed final qualification, as they should have an understanding of binary at this stage in their studies.
- There are many apps and online tools to convert binary to decimal and vice versa but learners do not grasp the concept without taking it back to basics and completing the process on paper. It is important that learners see the connection between the decimal number output through a light emitting diode (LED) array, liquid crystal display (LCD) or 7-segment display and the binary or hexadecimal number in the code they develop. Hexadecimal is not part of the unit, but it is used widely within embedded programs, therefore, it is advisable to cover binary to hexadecimal conversions.
- By this stage, learners will have already been using the integrated development environment (IDE), connecting and reconnecting the microcontroller and will be familiar with compiling code, common syntax errors, debugging and safe practices. Learners will have identified the main aspects from the basic code example, such as declaring libraries, declarations, commenting and the coding structure. This allows you to reflect back over the code/construction tasks that they have completed. Discussion of each of the aspects can then follow in more detail and give the justification for each segment of code supplied.
- This is now the time to develop the basic code from the previous topic with learners. The code will allow you to identify the input and output constructs, what they are doing and why for learners.
- Learners can manipulate the original code to include comments that will help them develop an understanding and mean they have revision notes when they return. You can get learners to: output to different ports and use other analogue inputs, create and include different libraries, call subroutines, create their own delays and develop different control structures that have the same ultimate function (if, if else, switch, case, etc.).
- Again, good planning and having several coding examples from learning aim A will allow the individual aspects of logic and arithmetic operators to be discussed and explained to learners, with examples that they have already seen functioning on a microcontroller system. This will not cover each aspect, but if there are small, iteration tasks built into the examples, learners can see the final output by including new lines of code that use different constructs to give either the same output or an extension of the initial output. For example, a button that increments an output on a 7-segment display by one, a second input by two, a third that if held for two seconds will reset. Expand the code further so that it counts from 0 to 9 with one input, if left will stop and decrement when it reaches 8 after flashing for three seconds. Expand even further to display 0 to 9 depending on the value of an analogue input.

- This method of delivery encourages peer learning and practice-sharing to overcome coding errors, supporting any weaker learners that do not initially grasp understanding the construct and how it is written into code. Learners need to continually practise and develop code to achieve a more mature solution than that initially issued.
- Now, learners will have typed various example programs and developed these examples to create solutions that are more complex. Take one of the already completed programming tasks and issue the associated program design documentation. Learners review the documentation and discuss the similarities of the plan to the finished product in groups. Discuss the structure of the pseudo code, flow chart and any related decision tables and the importance of a plan in creating any piece of code to ensure that it functions the way it needs to. To develop this topic, a new fresh task is required that you can work through with learners, using all the skills and knowledge that they have developed to date regarding coding. This needs to be a programming task that will incorporate several of the coding constructs and should be a program that has the option to be further developed for those learners who want to stretch themselves.
- Initially, you should walk through the coding problem with learners brainstorming the solution, the decisions and actions the code needs to carry out and then reflecting how this can be notarised into a document that can be interpreted by any embedded programmer.

Topic 3 – Project manage the development lifecycle

- Select one of the minimum three complete projects suggested in the initial part of this specification, which you prepare in advance. This will follow the format of the internal assessment task. Working through each stage of those in small class groups will allow you to identify any weak areas in learner knowledge, giving you the opportunity to address areas of weakness if identified. This sample allows learners to see and understand the requirements of the assessment task, and to visualise a potential solution. It is important to review each learner's design solution, as they can often overthink the solution and create problems in coding and construction that is not immediately evident to them.
- The completion of an in-class project will improve learners' confidence in taking a design challenge and realising the solution in a structured methodical way that will ensure a variance of success depending on the abilities of the learner.
- If the topics above have been covered in sufficient detail, learners will only need to be instructed on how to collate their portfolio of evidence for any given process, including the amount of detail and annotation required at each stage.
- Finally, a second project should be introduced with learners spending several weeks under supervision and instruction completing a second sample project as though they were under examination conditions. There is one exception, you can give advice and guidance on direction, structure and errors identified that may prove crucial in realising the final product and/or solution.

Details of links to other BTEC units and qualifications, and to other relevant units/qualifications

This unit links to:

- Unit 17: Power and Energy Electronics
- Unit 19: Electronic Devices and Circuits
- Unit 20: Analogue Electronic Circuits
- Unit 21: Electronic Measurement and Testing of Circuits
- Unit 23: Digital and Analogue Electronic Systems
- Unit 32: Computer System Principles and Practice
- Unit 33: Computer Systems Security
- Unit 34: Computer Systems Support and Performance
- Unit 35: Computer Programming
- Unit 36: Programmable Logic Controllers
- Unit 37: Computer Networks
- Unit 38: Website Production to Control Devices

As a mandatory unit on all Extended Diploma in Engineering qualifications, this unit will give a suitable foundation for learners to progress onto higher level courses or an apprenticeship.

Resources

In addition to the resources listed below, publishers are likely to produce Pearson-endorsed textbooks that support this unit of the BTEC International qualifications in Engineering. Check the Pearson website (qualifications.pearson.com/endorsed-resources) for more information as titles achieve endorsement.

Textbooks

Blum R – Arduino Programming in 24 Hours – Sams Teach Yourself, 1st Edition (Sams, 2014) ISBN 9780672337123.
A good detailed guide to programming using the Arduino IDE.

Ibrahim D – *PIC Microcontroller Projects in C: Basic to Advanced, 2nd Edition* (Newnes Elsevier, 2014) ISBN 9780080999241.
This book demonstrates how to develop a range of microcontroller applications using a project-based approach.

Van Dam B – *PIC Microcontrollers: 50 Projects for Beginners & Experts* (Elektor Electronics, 2008) ISBN 9780905705705.
This book has a hands-on approach to projects and gives several useful examples for assessor and learner.

Siegesmund M – *Embedded C Programming: Techniques and Applications of C and PIC MCUS* (Newnes Elsevier, 2014) ISBN 9780128013144.
A hands-on introductory course on concepts of C programming using a PIC® microcontroller and CCS Compiler.

Websites

Go to the 'microchip' website. This is a site for learners to download the free MPLAB® IDE and conduct research.

Visit the 'picaxe' website. This is a site for learners to download the free PIXAXE® and conduct research.

Videos

Search for the following videos on YouTube:

- Tutorial to support using the MPLAB® IDE.
- Tutorial to support creating a project in MPLAB® IDE.
- Tutorial on soldering basics.
- Pulse Width Modulation (PWM) video.

Pearson is not responsible for the content of any external internet sites. It is essential for tutors to preview each website before using it in class so as to ensure that the URL is still accurate, relevant and appropriate. We suggest that tutors bookmark useful websites and consider enabling students to access them through the school/college intranet.