SQA Advanced Unit Specification

General information for centres

Unit title: MCU/MPU Assembly Language Programming

Unit code: HP42 47

Unit purpose: This Unit is designed to give candidates knowledge and understanding and apply a structured approach to the writing of assembly language programs in an engineering environment. The environment best suited to this Unit is a host/target arrangement where the target is a microcontroller/microprocessor Unit (MCU/MPU) development system. Outcome 1 aims to cover microprocessor/microcontroller operation and some relevant architecture. Outcome 2 introduces the idea of a linear program designed to read an input location, process it and output the result. Outcome 3 introduces the concept of branching and looping. Outcome 4 develops the concept of the creating and calling a sub program.

On completion of the Unit the candidate should be able to:

1. Describe the operation of an assembly language program in a microcontroller or microprocessor based system.
2. Generate software to read data, process it and output the correct result.
3. Generate software that exercises various program branching structures while accessing external devices.
4. Generate software that utilises sub programs in the control of external devices.

Credit value: 1 SQA Credit at SCQF level 7: (8 SCQF credit points at SCQF level 7*)

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from National 1 to Doctorates.*

Recommended prior knowledge and skills: The normal entry qualification for this Unit would be possession of a Higher in Electronics or National Qualification Units D134 11 Combinational Logic or E9LG 11 Computing in Engineering 1 or D980 11 Programmable Systems.

Core skills: There may be opportunities to gather evidence towards Core Skills in this Unit, although there is no automatic certification of Core Skills or Core Skills components.

Context for delivery: This Unit was developed for the SQA Advanced Certificate/ Diploma in Electronics awards. If this Unit is used in another group award(s) it is recommended

that it should be taught and assessed within the context of the particular group award(s) to which it contributes.

Assessment: It is recommended that this Unit be assessed via a logbook and programming projects that cover the knowledge and skills presented in each Outcome.

It is recommended that assessment takes the form of laboratory experiments, followed up by written reports. Practical exercises should be undertaken under controlled, supervised conditions. Centres should supply candidates with guidelines on the necessary standard of documentation. Candidates should submit written reports within 14 days of the practical work being completed.

Centres should take every reasonable effort to ensure that reports are the candidates' own work. It may be possible to issue each candidate with a slightly different specification of equal complexity, or alternatively where there is a suspicion of copying or plagiarism, an appropriate response may be to interview candidates. A checklist should be used to record oral evidence of the candidates' understanding.

Centres are recommended to develop appropriate checklists to support the assessment requirements for each of the knowledge and skills items. Centres are also recommended to produce a marking schedule based on the evidence requirements listed indicating clearly the required content of the report. Candidates who do not meet the standard should be obliged to correct and resubmit their work.

Unit specification: statement of standards

Unit title: MCU/MPU Assembly Language Programming

Unit code: HP42 47

The sections of the Unit stating the Outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Describe the operation of an assembly language program in a microcontroller or microprocessor based system

Knowledge and/or skills

- Elementary understanding of fetch/execute cycle
- Understand the difference between operation and operand
- Understands and is able to use Hexadecimal and Binary codes

Evidence requirements

All parts of the knowledge and skills listed above shall be assessed. Evidence will be provided in the form of a written report. The candidate's response will be judged to be satisfactory where evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- describe the operation of an assembly language program
  - including an elementary description of the fetch/execute cycle for simple instructions
  - including details of the program flow

- draw the block diagram of a Microprocessor/Microcontroller system
  - including CPU, Memory, Input, Output and bus structure
  - showing directions of transfer of data on each bus.
  - including control bus signals.

Assessment guidelines

Centres may wish to combine the assessment of this Outcome with the assessment of one or more of the other Outcomes.

Outcome 2

Generate software to read data, process it and output the correct result

Knowledge and/or skills

- Data transfer and arithmetic instructions
- Use of Editor and Assembler
- Use of tools to download to and debug software on a target system
- Use of tools to test and demonstrate software and to gather information showing correct operation

Evidence requirements

All parts of the knowledge and skills listed above shall be assessed. Evidence will be provided in the form of a written report. The candidate's response will be judged to be satisfactory where evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- develop assembly language programs to read data, process it and output the correct result utilising a variety of the available instructions
- record output of the program for appropriate test data.
- write a report which includes the following:
  - a correctly commented listing of the program used
  - a description of the operation of each instruction in the program
  - details of test data and output result
  - conclusions

Candidates may be permitted to modify programs that they have developed in tutorial exercises, or combine sections of code adapted from other programs.

Assessment guidelines

Centres may wish to combine the assessment of this Outcome with the assessment of one or more of the other Outcomes.

Outcome 3

Generate software that exercises various program branching structures while accessing external devices

Knowledge and/or skills

- Knowledge of how operations affect the Condition Code Register (CCR)
- Use of conditional and unconditional branch instructions to alter program flow and create the following structures:
  - If/Then, If/Then/Else, While/Do, Repeat/Until, For/Next
- Knowledge of short and long branch operation (if available)

Evidence requirements

Evidence for the knowledge and or skills in this Outcome will be provided on a sample basis. The evidence will be provided in the form of a written report. Each candidate will need to demonstrate that she/he can provide a written report based on a sample of the items shown above. In any assessment of this Outcome both knowledge and/or skill items **one and three**

will be assessed in full and knowledge and/or skills item **two will be assessed according to the sample** below:

If/Then/Else;
1 of either While/Do or Repeat/Until;
For/Next

In order to ensure that the candidates will not be able to foresee what items they will be questioned on, a different sample is required each time the Outcome is assessed. Candidates must provide a satisfactory response to all three knowledge and/or skills items.

The candidate's response will be judged to be satisfactory where evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- develop an assembly language program to demonstrate correct operation of each of the above structures using a selection of available branching instructions
- correctly utilise the branching instruction appropriate to the CCR flag
- write a report which includes the following:
  - a commented listing of the program used
  - a description of the operation of the program used
  - test data and evidence of correct program operation
  - conclusions

Candidates may be permitted to modify programs that they have developed in tutorial exercises, or combine sections of code adapted from other programs.

Assessment guidelines

Centres may wish to combine the assessment of this Outcome with the assessment of one or more of the other Outcomes.

Outcome 4

Generate software that utilises sub programs in the control of external devices

Knowledge and/or skills

- Processor stack and stack pointer operation
- Operation of Jump and Branch to Subroutine instructions **or** Operation of Call instructions to Subroutines
- Operation of Return from Subroutine instruction

Evidence requirements

All parts of the knowledge and skills listed above shall be assessed. Evidence will be provided in the form of a written report. The candidate's response will be judged to be satisfactory where evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- develop assembly language subroutines
- develop programs that correctly call and return from subroutine(s)

- test the program with suitable data
- record results of tests
- write a report which includes the following:
  - a listing of the program(s) used
  - a description of the operation of the call to the subroutine with details of where return addresses are stored
  - details of test data
  - evidence of successful operation of the program
  - conclusions

Candidates may be permitted to modify programs that they have developed in tutorial exercises, or combine sections of code adapted from other programs.

Assessment guidelines

Centres may wish to combine the assessment of this Outcome with the assessment of one or more of the other Outcomes.

Administrative information

Unit code: HP42 47

Unit title: MCU/MPU Assembly Language Programming

Superclass category: CB

Date of publication: August 2017

Version: 01

Source: SQA

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of SQA Advanced Qualifications.

FURTHER INFORMATION: Call SQA's Customer Contact Centre on 44 (0) 141 500 5030 or 0345 279 1000. Alternatively, complete our Centre Feedback Form

Unit specification: support notes

Unit title: MCU/MPU Assembly Language Programming

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 40 hours.

Guidance on the content and context for this Unit

This Unit has been written in order to allow candidates to develop the knowledge, understanding and skills in the following areas:

1. Describe the operation of an assembly language program in a microcontroller or microprocessor based system.
2. Generate software to read data, process it and output the correct result.
3. Generate software that exercises various program branching structures while accessing external devices.
4. Generate software that utilises sub programs in the control of external devices.

This Unit was designed to permit candidates to apply assembly language to the control of input/output hardware, and develop skill in the utilisation of an appropriate program development environment. It is intended that candidates will be given example programs to assist in the development of their ability to understand Microcontroller/Microprocessor operation. To access external hardware, set-up and read/write ports candidates should be supplied with program modules such as macros or subroutines. The external devices could include, but not be limited to, switch/LED Units, stepper motors, A to D and D to A converters.

This Unit does not include specifications for the type of target system - this is left up to the discretion of centres. It is appreciated that centres will have access to a wide variety of hardware, such as personal computers, microprocessor development boards, microcontrollers, and DSP devices, all of which may be utilised to support this Unit. Centres should endeavour to utilise current technology in delivering this Unit.

Guidance on the delivery and assessment of this Unit

This Unit was developed as one of a series of SQA Advanced programming Units and is at SCQF level 7. The Unit is designed to be taken as a stand-alone Unit or integrated with the Unit MCU/MPU I/O Hardware Control at SCQF level 8.

This Unit should be taught using a student centred approach. In designing this Unit it was envisaged that candidates would learn by doing rather that in a classroom context. The provision of subroutines or macros to provide access to external hardware is intended to permit the student to concentrate on program operation and structures via programs that will interact with the external world.

Outcome 1 depth and coverage should be sufficient to underpin the candidates' understanding of the software development expected in Outcomes 2, 3 and 4. A suggested approach might be to introduce elements of the hardware and architecture as they become necessary during the development of the candidates' assembly language skills. For example, when explaining how a simple program runs, a simplified explanation of the fetch-execute cycle could be developed with reference to the microprocessor program counter, instruction register, address and data registers, control signals and random access memory. Similarly when introducing the candidates' to branching programs the function of the condition code register could be introduced and explained.

A list of topics by Outcome is given below with approximate timings and advice on coverage. The timings are to be seen as a guide to how much weight to place in each area. They are not prescriptive.

Outcome 1
Describe the operation of an assembly language program in a microcontroller or microprocessor based system.

It is difficult to specify a time allocation here since it is anticipated that the fetch/execute cycle description, instruction format and hexadecimal notation will be incorporated into the teaching and practical examples used for Outcomes 2, 3 and 4. The idea of CPU, ROM, RAM, I/O and memory map should feature in discussions. As a rough guide spend no more than a 2 or 3 hours spread through the Unit time allocation.

Outcome 2
Generate software to read data, process it and output the correct result. (8hours)

- cover data move, add, subtract, multiply and divide instructions. Bring in a reference to the effect that these have on the Condition Code Register (CCR).
- while these rather simple programs are being exercised the students should be given the opportunity to gain familiarity with the editing/programming/debug environment. Many simple examples can be demonstrated within a debugging environment which show very clearly the program flow and effects on registers, memory and CCR.
- as the student's experience grows access to I/O ports can be introduced with examples of the use of (supplied) subroutines to set up and access external ports

Outcome 3
Generate software that exercises various program branching structures while accessing external devices (10 hours)

- Demonstrate using the debug tools the flow of a program that has an unconditional branch. Follow this with a simple conditional branch and show the branch being taken and NOT taken depending on the state of the corresponding CCR flag.
- Illustrate the following structures:
  — If/Then, If/Then/Else, While/Do, Repeat/Until, For/Next

Outcome 4
Generate software that utilises sub programs in the control of external devices (12 hrs)

- use the debug tool to illustrate what happens when a subroutine call is made and a subsequent Return from Subroutine. Introduce and show the operation of the Stack for storing return addresses.
- some microprocessors and microcontrollers have separate Jump and Branch to Subroutine instructions. Illustrate both of these to show the difference.

Most of the student learning should be through practical tutorials following demonstrations. Students should be encouraged to modify existing programs rather than starting from scratch. The tutorial exercises could be developed in the context of an overarching final assessment or smaller assignments which still yield evidence to support the candidates' competence as described in each Outcome.

The extra 8 hours not accounted for above can be counted as home study and preparation for laboratory exercises.

Open learning

This Unit is more suitable for laboratory delivery however it could be delivered by distance learning provided the candidate has access to a microprocessor development system and external hardware interfaces. This may require some degree of on-line support. However, with regard to assessment, planning would be required by the centre concerned to ensure the sufficiency and authenticity of candidate evidence. Arrangements could be required to be put in place to ensure that the assessment whether done at a single or at multiple events was conducted under controlled, supervised conditions.

To keep administrative arrangements to a minimum, it is recommended that a single assessment paper (taken by candidates at a single assessment event) be used for distance learning candidates.

For information on normal open learning arrangements, please refer to the SQA guide *Assessment and Quality Assurance of Open and Distance Learning (SQA 2000)*

Equality and inclusion

This unit specification has been designed to ensure that there are no unnecessary barriers to learning or assessment. The individual needs of learners should be taken into account when planning learning experiences, selecting assessment methods or considering alternative evidence.

Further advice can be found on our website www.sqa.org.uk/assessmentarrangements.

General information for candidates

Unit title: Assembly Language Programming

This Unit is intended to introduce the candidate to embedded systems software design. There are many devices in current use which utilise devices which contain a microprocessor, read-only and random access memory, EEprom memory, various peripheral interfaces all contained inside a single integrated circuit. These devices are called Microcontrollers. Common examples include car engine management, airbag systems, printers and mobile phones. They are usually programmed in C and/or Assembly language.

This Unit will introduce you to assembly language programs to read, process and output data to effect control of external hardware. This will give you the necessary knowledge and skills to apply assembly language programming expertise to engineering solutions which utilise embedded systems. You will also gain experience in the use of a software development environment the complexity of which will depend on the resources available to the centre.

You will have the opportunity to develop input/output programs and test them with suitable external hardware in the laboratory. Each part of the Unit will be introduced by experiment, after which there will be an assessed practical exercise. These will be assessed mostly by laboratory reports. The lecturer will also be completing checklists to ensure that you have completed all necessary parts of the exercises. You may also be required to submit to a brief oral examination to ensure that the work you have submitted is your own and that you fully understand the material.

The centre where you are studying may choose to combine parts of this Unit with the level 8 SQA Advanced Unit MCU/MCP I/O Hardware Control. This will have the benefit of reducing the amount of assessment you have to undertake. It will be the responsibility of the centre to ensure that all parts of the syllabus are still covered.