For the attention of all staff responsible for the delivery of National Qualifications in Technical Education

Action by Recipient	
	Response required
✓	Note and pass on
	None — update/information only

Contact Name: Derek Middleton at Glasgow
Direct Line: 0845 213 5479
E-mail: derek.middleton@sqa.org.uk

Dear Colleague

Revised data booklets for Technological Studies

The data booklets have been revised and new versions posted on SQA's website (**www.sqa.org.uk**). Copies will also be provided to centres who are currently presenting classes in Technological Studies. These booklets should be used for the 2007 examinations.

The three data booklets are now organised as follows:

- Standard Grade and Intermediate 2 data booklet
- Higher data booklet
- Advanced Higher data booklet

Notes on the content of the new booklets are contained in Appendix 1 and Appendix 2 attached to this letter.

It should be noted that, although there are additions to the contents of the some of the data booklets, these additions are included in order to enable centres/candidates to optimise their teaching/learning; there are no additions that relate to new content not previously included in the Course.

Please do not hesitate to contact me if you require any more information.

Yours sincerely

D Middleton

Derek Middleton
Qualifications Manager
NQ Computing and Technical Education

Appendix 1 — Notes on changes to data booklets at each level

Technological Studies — Standard Grade and Intermediate 2 level: Data booklet 2006 version

This data booklet has been produced to better enable centres to prepare candidates for the external examinations in Technological Studies at Standard Grade and Intermediate 2 levels.

A single booklet has been produced to be used with either the Standard Grade or Intermediate 2 Courses. The layout has been changed and the following changes made in comparison to the previous Standard Grade and Intermediate 2 booklets:

Changes from previous Standard Grade booklet

Cosmetic changes to the layout and some descriptions.
Formula triangles are included to aid candidates in transposition of formula.

Changes from previous Intermediate 2 booklet

Pinout diagram for 7400 series ICs have been added.
Resistor Colour Coding has been added.
Formula triangles are included to aid candidates in transposition of formula.
Formula for pressure, force and area are now included.

Technological Studies — Higher level: Data booklet 2006 version

This data booklet has been produced to better enable centres to prepare candidates for the external examination in Technological Studies at Higher level.

The Intermediate 2 content has been deleted, and is now combined with Standard Grade content in a new combined Standard Grade and Intermediate 2 data booklet.

The content of the data booklet has been revised, and the presentation has been rearranged, in order to facilitate its use both in the classroom and in examinations. This version should be used by centres with immediate effect, and is intended for use in the 2007 Technological Studies Higher level external examination.

Summary of changes to content

The order of presentation of data has been reorganised, to follow the structure of the Higher Course. There are now three sections, as follows:

- Applied Electronics
- Systems and Control
- Structures and Materials

The added content for each of these sections is detailed below:

1. Applied Electronics
   - Kirchhoff's First Law
   - Kirchhoff's Second Law
   - Voltage Divider Rule
   - MOSFET Transconductance
   - Improved graphs for thermistors, thermocouple and LDR

2. Systems and Control
   - Flowchart symbols — revisions to symbols and descriptions
   - Simplification of Number Systems information
   - PBASIC instruction set — simplified presentation
   - Variables — added information on *byte* and *word* variables

The section headed 'PBASIC Instruction Set — additional instructions normally outwith the scope of Intermediate 2 and Higher' has been removed; all instructions needed for this section of the Course are included in the main table.

3. Structures and Materials
   - Simplified presentation of equations

Technological Studies — Advanced Higher level: Data booklet 2006 version

This data booklet has been produced to better enable centres to prepare candidates for the external examination in Technological Studies at Advanced Higher level.

The content of the data booklet has been revised, and the presentation has been rearranged, in order to facilitate its use both in the classroom and in examinations. Errors present in the 2001 version have been corrected. This version should be used by centres with immediate effect, and is intended for use in the 2007 Technological Studies Advanced Higher level external examination.

Summary of changes to content

The order of presentation of data has been reorganised, to follow the structure of the Advanced Higher Course. There are now three sections, as follows:

- Applied Electronics
- Systems and Control
- Structures and Materials

The added content for each of these sections is detailed below

1. Applied Electronics
   - Kirchhoff's First Law
   - Kirchhoff's Second Law
   - Voltage Divider Rule
   - Electrical Power
   - 555 Timer
   - 7493 binary counter
   - 7447 BCD — 7 segment display decoder
   - Integrator
   - Scmitt Trigger
   - Wien Bridge oscillator
   - 4-bit D – A converter
   - 4-bit A – D converter
   - Improved graphs for thermistors, thermocouple and LDR

2. Systems and Control
   - Flowchart symbols — revisions to symbols and descriptions
   - Special function registers — clarified description
   - Instruction syntax — clarified
   - Assembler language instruction set — improved description of functions, more logical order of presentation
   - Time delay sub-procedures — standardised delays specified (assume pre-written for examination purposes)

The section headed 'Assembler Code Instructions — additional instruction normally outwith the scope of Advanced Higher" has been removed; all instructions needed for delivery of this section of the Course are included within the main tables.

Appendix 1 — Changes to data booklets at each level

(Note: a letter of guidance has been published that provides extra support for teachers in the use of instructions required to perform proportional control by programmable systems, in order to further clarify the published LTS course notes for this section of the Course).

3. Structures and Materials
    - ♦ Simplified presentation of equations
    - ♦ Bending of Beams table corrected and improved presentation
    - ♦ Moments of Area simplified
    - ♦ Tables for Dimensions and properties of common sections — simplified

Appendix 2 — Notes on Assembler Code for Advanced Higher Technological Studies

The purpose of these notes is to describe and exemplify the use of instructions which are required in Outcome 3 for proportional control, and are intended to clarify the assignments given in the LTS course notes. The notes below expand on the descriptions given in the new data booklet for Advanced Higher (2006 Edition)

Mnemonic	Operand	Instructions
xorlw	k	(exclusive OR literal with W) — perform logical XOR on corresponding bits of constant and working register (store result in W). If the literal is equal to the value in W the result is zero.
xorwf	f,d	(exclusive Or working register with file) — performlogical XOR on corresponding bits of W and a file. If value in file is equal to value in W result is zero.
andlw	k	(AND working register with file) — perform Logical AND on corresponding bits of literal and W (store result in W).
andwf	f,d	(AND working register with file) — perform Logical AND on corresponding bits of W and a file.
rlf	d	(rotate left file) — move all bits of a file one place to the left, move bit 7 to carry bit; move carry bit to bit 0.
rrf	f,d	(rotate right file) — move all bits of a file one place to the right, move bit 0 to carry bit; move carry bit to bit 7.

Exclusive OR

This instruction compares two values using logical XOR where 0,0 or 1,1 = 0. If the values are equal the value 00000000 is the result. If the result in the Working Register after a comparison is zero then the Zero flag in the STATUS register is set.

xorlw k

This instruction allows a number (literal value k) to be compared with the value in the Working Register. If the two values are equal the **Zero** bit in the STATUS register is set; if they are not equal it is clear.

Working register	11000110	11000110
Number	11000110	11000111
Result in W	00000000	00000001
Status Zero bit	Set	Clear

An example is shown below.

```
init:   bsf     STATUS,RP0
        movlw   b'00001111'     ;pins 7-4 outputs, pins 3-0 outputs
        movwf   TRISB
        bcf     STATUS,RP0

main:   movlw   b'10010010'     ; a value is placed in the working register
        movwf   B3              ; W value stored into register file B3
```

```
        movfw   B3              ; put B3 contents into working register
        xorlw   b'10010010'     ; the W value is compared with the  binary value 10010010
        btfss   STATUS,Z        ; if the values are the same the next line is skipped
        goto    stop            ;if the values are not equal jump program ends
        movlw   b'11110000'     ;PORTB bits 7-4 are set
        movwf   PORTB
        movlw   d'30'
        call    wait            ; for 3 seconds
stop:   clrf    PORTB           ; all bits of PORTB are cleared
        end
```

This program compares the value put into the Working Register with the value in B3; if they are equal PORTB bits 7-4 will go high for 3 seconds. If the B3 value is modified so that it does not equal the literal value pins 7-4 will stay low.

xorwf f,d

This instruction allows the value in the register file (f) to be compared with the value in the Working Register. The result of the comparison (0 or 1) is saved either in the Working Register or back into the file. If the two values are equal, the Zero bit in the STATUS register will be set, but if they are not equal it will be clear. An example is shown below.

```
init:   bsf     STATUS,RP0
        movlw   b'00001111'     ;pins 7-4 outputs, pins 3-0 outputs
        movwf   TRISB
        bcf     STATUS,RP0


main:   movlw   b'00001011'     ; a value is placed in the working register
        xorwf   PORTB,W         ; the W value is compared with the value input to PORTB
        btfss   STATUS,Z        ; if the values are the same the next line is skipped
        goto    stop            ;if the values are not equal jump program ends
        movlw   b'11110000'     ;PORTB bits 7-4 are set
        movwf   PORTB
        movlw   d'30'
        call    wait            ; for 3 seconds
stop:   clrf    PORTB           ; all bits of PORTB are cleared
        end
```

This program compares the value input into pins 3-0 of PORTB with the literal value stored in the Working Register. If pins 0, 1 and 3 are set high and pin 2 low the two values are equal and PORTB bits 7-4 will go high for 3 seconds. If this input value is changed, and the program rerun, the values will not be equal and pins 7-4 will stay low.

The next example saves the result of the comparison back into the file, in this case PORTB. During the first loop PORTB bits are all 0 and the Working Register bits are all 1. This will set all output

bits of PORTB to 1 when the result of the XOR operation is saved back to PORTB. On the next loop PORTB and the Working Register are equal (both all 1) so the comparison produces a 0 in each of the bits, which causes PORTB outputs to go to 0. This cycle repeats at 4Hz.

```
init:   bsf     STATUS,RP0
        movlw   b'00001111'     ;pins 7-4 outputs, pins 3-0 outputs
        movwf   TRISB
        bcf     STATUS,RP0

main:   clrf    PORTB           ;all PORTB bits to 0
flash:  movlw   b'11111111'     ;set all Working Register bits to 1.
        xorwf   PORTB,F         ;XOR W with PORTB save result in PORTB
        movlw   d'250'
        call    pause           ;0.25 second delay
        goto    flash           ;repeat forever
```

These are very useful instructions as they allow a value obtained from another source (e.g. from the register file DATA after the sub-procedure adcread is called) to be compared with a set value and a decision taken whether these values are equal. They are used along with the sublw and subwf commands and the carry flag in the STATUS register. These instructions allow an input to be analysed to determine whether it is greater than, less than, or equal to, a set value.

AND

This instruction compares two values using logical AND where 0,0 or 1,0 or 0,1 = 0 and 1,1 = 1. If any pair of bits in the compared values are *both* zero the result for that bit in the Working Register will be 1.

andlw

AND number (literal) with W and save the result in W; if a pair of the bits in both bytes is equal to 1 the result in W will be 1 for that pair. Otherwise the result will be 0.

Working register	11000110
Number	00001111
Result in W	00000110

The example shown below uses the number to 'mask' off parts of PORTB which are not needed leaving only the status of the bits which are of interest. In this case input pins 0, 1 and 2 are being monitored; if these all go high (1) simultaneously pins 6 and 7 are set high. However the condition of the other input and output pins is not important so these are 'masked' so that they can be ignored.

```
init:   bsf     STATUS,RP0
        movlw   b'00011111'     ;pins 7-5 outputs, pins 4-0 outputs
```

```
        movwf   TRISB
        bcf     STATUS,RP0

main:   movfw   PORTB       ;Put the contents of PORTB into W
        andlw   b'00000111' ;mask of all but bits 0,1 and 2
        xorlw   b'00000111' ;XOR W with PORTB, save result in PORTB
        btfss   STATUS,Z
        goto    main
        movlw   b'11000000'
        movwf   PORTB
        end
```

andwf

AND W with a register file and save the result in the file or W; if a pair of the bits in both bytes are equal to 1 the result will be 1 for that pair. Otherwise the result will be 0.

For example: ANDing the contents of the Working Register (W) with a Register file (F)

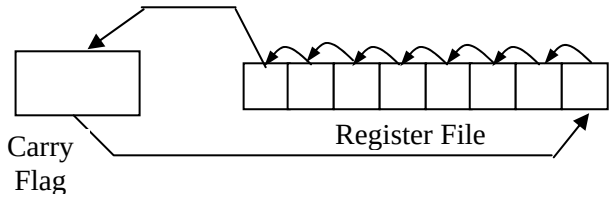
W contents	11000110
F contents	00001111
Result in W or F	00000110

Rotate file

This command allows the value in a file to be doubled or halved. If used in a loop it also allows bits to be moved in or out of a register file serially.

rlf

The contents of a register file are moved one place to the left, bit 7 is moved into the carry bit, and bit 0 is replaced using the carry bit as shown below. If the value in the register file is less than 128 (bit 7 = 0) then the file value is doubled



Before rlf; file value = 01100110 = 102
After rlf; file value = 11001100 = 204

The example shown below uses the rlf instruction to double the value in the file register COUNTER. As the carry flag bit is used to replace bit 7 this must be clear when the operation is

started.

```
init:    bsf     STATUS,RP0
         movlw   b'00001111'     ;pins 7-4 outputs, pins 3-0 outputs
         movwf   TRISB
         bcf     STATUS,RP0

main:    bcf     STATUS,C       ;clear the carry flag
         movlw   d'81'          ; put a decimal value of 81 in W
         movwf   COUNTER        ;put this number into COUNTER
         rlf     COUNTER,F      ; double COUNTER value to 162
         end                    ;stop program
```

rrf

The contents of a register file are moved one place to the right, bit 0 is moved into the carry bit and bit 7 is replaced using the carry bit as shown below.

Before rrf; file value = 01100110 = 102
After rrf; file value = 00110011 = 51

```
init:    bsf     STATUS,RP0
         movlw   b'00001111'     ;pins 7-4 outputs, pins 3-0 outputs
         movwf   TRISB
         bcf     STATUS,RP0

main:    bcf     STATUS,C       ;clear the carry flag
         movlw   b'011000000'   ; put a number into W
         movwf   PORTB          ;put this number into PORTB
         movlw   d'20'          ;PORTB pins 6 and 5 high
         call    wait           ;for two seconds
         rrf     PORTB,F        ;half PORTB so pins 5 and 4 are now high
         end                    ;stop program
```

comf

The complement file instruction is used to toggle the value of each bit in a Register File (0 becomes 1 and 1 becomes 0) as shown below.

```
File contents before comf      10110011
File contents after  comf      01001100
```

Working with a single 8 bit file, after a subtraction which results in a value < 0, the file becomes corrupted. This is because if 3 is subtracted from 0 the result is not -3 but 256-3 which is 253. The comf instruction corrects this although the result is 1 less than the true value as an 8 bit register can only show up to 255 and not 256; to correct this the value 1 is added to the result. The simplest way

to do this is using the incf instruction.

File contents after 0-3 11111101 = 253
File contents after comf 00000010 = 2
After incf instruction 00000011 = 3

The example shown below uses a number of sub-procedures to subtract a number from the Register File COUNTER and display the result. It checks to see if the value is less than zero and if so it corrects the answer.

```
subtract:  bcf      STATUS,C      ;clear the carry flag
           movlw    b'011000000'   ; put a number into W
           subwf    COUNTER,W     ;subtract the W value from the value of COUNTER
           btfsc    STATUS,C      ;PORTB pins 6 and 5 high
           call     display       ;call sub-procedure to show value in COUNTER
           comf     COUNTER,F     ;complement the corrupted file
           incf     COUNTER,F     ; add one to correct the file
           call     display       ;call sub-procedure to show value in COUNTER
           return
```