Test Report

Report No	TR/11/265/M1
Client	Ei Electronics Shannon Industrial Estate, Shannon, Co. Clare, Ireland
Authority & date	2/36639 (SMO 7708632) – 11/07/08
Items tested	Ei208 & Ei207 software evaluation
Specification	BS EN 50291: 2010 Clause 4.6 – *Electrical apparatus for the detection of carbon monoxide in domestic premises*
Results	The software has been examined to the relevant requirements of the above specification and has been found to comply with these requirements, subject to the implementation of any corrective actions detailed in this test report.
Prepared by	Anees Vaheed
Authorised by	PT Paul Turner
Issue Date	06/09/11
Conditions of issue	This Test Report is issued subject to the conditions stated in current issue of CP0322 'Conditions of Contract for Testing'. The results contained herein apply only to the particular sample/s tested and to the specific tests carried out, as detailed in this Test Report. The issuing of this Test Report does not indicate any measure of Approval, Certification, Supervision, Control or Surveillance by BSI of any product. No extract, abridgement or abstraction from a Test Report may be published or used to advertise a product without the written consent of the Managing Director, Testing Services, who reserves the absolute right to agree or reject all or any of the details of any items or publicity for which consent may be sought. **Opinions and interpretations expressed herein are outside the scope of UKAS accreditation**

UKAS TESTING 0351

Blank Page

Test Sample Designation

Sample No.	Description of Sample (Including Serial Number)
N/A	Software Evaluation – Paperwork only

Design Documentation

Document Description	Issue
Software Changes Review process	10-01-2011
Electrical Schematic	23-08-2010
Software Flow Chart	09-02-2010
Source Code	A16989 Rev 1a

Description of Test Sample

- Software documentation relating to Ei208

1. Introduction

The objective of the evaluation was to assess the Ei208 software documentation provided for compliance with the relevant requirements of BS EN 50291: 2010 Clause 4.6 – *Electrical apparatus for the detection of carbon monoxide in domestic premises*. The assessment was made by the review and analysis of the software documentation provided by the manufacturer.

Note: This report is issued to cover the Ei207 as both Ei207 and Ei208 are based on the same generic software design.

```
;                    Product Selector Section
;                    ------- -------- -------
;            (to be set by Test Engineering before production)
;                    Model Selection
;MODEL               equ    EI207
MODEL                equ    EI208D
```

The Ei207 and Ei208 use the same software but the test engineer selects whether the version used is the Ei207 or Ei208 by commenting the model that is not used (see the code example above EI207 model is commented out thereby selecting the EI208 version).

2. Test Plan

This report covers the evaluation of the system against the relevant requirements of BS EN 50291: 2010 Clause 4.6 – Electrical apparatus for the detection of carbon monoxide in domestic premises. No other clauses of the standard are covered by this report.

3. Observation and Results

3.1. Clause 4 – Design Principles

3.1.1. Clause 4.1 – Basic Requirements

3.1.1.1. General

The observations made regarding the Design requirements from the software evaluation are as follows.

3.1.1.2. Analogue/Digital interface

Analog module and Digital Port initialisation of the unit

```
;                    Port Initialisations
;
;             Port A
TRIS_A_INIT equ    B'00101111' ; I/O configuration of Port A
                               ; for normal operation
TRIS_A_ICT1 equ    B'00001100' ; I/O configuration of Port A
TRIS_A_ICT2 equ    B'00001000' ; for ICT tests
TRIS_A_ICT3 equ    B'00001000'
;
TRIS_A_IO_OUT      equ    B'00001111' ; I/O configuration of Port A
                               ; for IO output
TRIS_A_TX    equ   B'00001111' ; I/O configuration of Port A
                               ; for RF module data output
TRIS_A_LCD   equ   B'00101011' ; I/O configuration of Port A
                               ; for LCD clock output
;
PORT_A_INIT equ    B'00000000' ; Initial settings of Port A outputs
PORT_A_ICT1 equ    B'00100011' ; Port A outputs for ICT Phase 1
PORT_A_ICT2 equ    B'00010100' ; Port A outputs for ICT Phase 2
PORT_A_ICT3 equ    B'00110000' ; Port A outputs for ICT Phase 3
;
;
;             Port C
TRIS_C_T_BUT       equ    B'00111001' ; I/O configuration of Port C
                               ; for test button read
TRIS_C_NORM equ    B'00111000' ; I/O configuration of Port C
                               ; for normal operation
TRIS_C_ICT   equ   B'00000110' ; I/O configuration of Port C
                               ; for ICT test
TRIS_C_LED   equ   B'00000000' ; I/O configuration of Port C
                               ; for LED lighting
;
PORT_C_INIT equ    B'00111000' ; Initial settings of Port C outputs
PORT_C_ICT1 equ    B'00111001' ; Port C outputs for ICT Phase 1
PORT_C_ICT2 equ    B'00000000' ; Port C outputs for ICT Phase 2
PORT_C_ICT3 equ    B'00111001' ; Port C outputs for ICT Phase 3
;
```

All the ports are defined as digital Input/output and the digital conditions are predefined in the program.

```
;            Analog Setup
;
      IF     DEVICE == PIC16F684
;
;                  A/D Converter Settings
ADCON1_SET   equ    B'00010000' ; Sets A/D clock to Tosc/8
;
ANSEL_SET    equ    B'00000111' ; Sets RA0, RA1 and RA2
                                ; as analog inputs
;
ADCON0_OFF   equ    B'00000000' ; Sets analog channel to
                                ; smoke reference (AN0)
                                ; Result is left justified
                                ; Reference voltage is Vdd
                                ; Turns OFF A/D converter
ADCON0_HI_CO       equ    B'00000001' ; Sets analog channel to
                                ; high CO gain (AN0)
                                ; Result is left justified
                                ; Reference voltage is Vdd
                                ; Turns ON A/D converter
ADCON0_LO_CO       equ    B'00000101' ; Sets analog channel to
                                ; low CO gain (AN1)
                                ; Result is left justified
                                ; Reference voltage is Vdd
                                ; Turns ON A/D converter
ADCON0_BATT_REF    equ    B'00001001' ; Sets analog channel to
                                ; reference voltage (AN2)
                                ; Result is left justified
                                ; Reference voltage is Vdd
                                ; Turns ON A/D converter
;
;                  Comparator Settings
CMCON0_INIT equ    B'00000111' ; Initial comparator settings
;
      ENDIF ; DEVICE


      IF     DEVICE == PIC16LF1823
;
;                  A/D Converter Settings
ADCON1_SET   equ    B'00010000' ; Sets A/D clock to Tosc/8
                                ; Result is left justified
                                ; Reference voltage is Vdd
;
ANSEL_A_SET equ    B'00000111' ; Sets RA0, RA1 and RA2
ANSEL_C_SET equ    B'00000000' ; as analog inputs
;
ADCON0_OFF   equ    B'00000000' ; Sets analog channel to
                                ; high CO gain (AN0)
                                ; Turns OFF A/D converter
ADCON0_HI_CO       equ    B'00000001' ; Sets analog channel to
                                ; high CO gain (AN0)
                                ; Turns ON A/D converter
ADCON0_LO_CO       equ    B'00000101' ; Sets analog channel to
                                ; low CO gain (AN1)
                                ; Turns ON A/D converter
ADCON0_BATT_REF    equ    B'00001001' ; Sets analog channel to
                                ; reference voltage (AN2)
                                ; Turns ON A/D converter
;
;                  Comparator Settings
CMCON0_INIT equ    B'00000111' ; Initial comparator settings
;
      ENDIF ; DEVICE


AD_CNVRT_CODE      BCF    INTCON,GIE  ; Disable interrupts
```

```
;
            BANKSEL     ADCON0             ; Lines put here to allow longer
            MOVWF ADCON0               ; time for analog inputs to settle
;
            BANKSEL     ADCON1
            MOVLW ADCON1_SET
            MOVWF ADCON1
;
            DLY_X_uS_SUB      D'4'  ; Wait 12us (4us set time + 8us
                              ; overheads) for A/D channel to
                              ; settle
;
            BANKSEL     ADCON0
            BSF   ADCON0,1    ; Set A/D GO bit.
;
;                 Wait until conversion complete
TEST_AD_BIT_2     BR_IF_TRUE  AD_START_BIT,TEST_AD_BIT_2
;
;                 Turn off A/D converter
            BCF   AD_POWER_BIT
;
;                 Leave result MSB in W register
            MOVF  ADRESH,W
;
;                 Restore bank to base bank
            BANKSEL     BASE_BANK
;
;
      IF    RF_MODE_2 == ON
;
            BSF   INTCON,GIE  ; Enable interrupts again
;
      ENDIF ; RF_MODE_2
;
            GOTO  SUB_RETURN  ; Returns pointing to upper bank

;                 Select high gain CO channel
FIRST_CO    MOVLW ADCON0_HI_CO
;
;                 Carry out A/D conversion
            CALL  AD_CONVERT  ; Returns pointing to base bank
                              ; Result MSB in W register
                              ; Interrupts enabled by AD_CONVERT
;
;                 Read A/D value
;                 Save left justified 10-bit reading
;                 in MSB, LSB registers
            MOVWF CO_READ_MSB
            BANKSEL     ADRESL
            MOVF  ADRESL,W
            BANKSEL     BASE_BANK
            MOVWF CO_READ_LSB
```

Once the analog conversion is finished the MSB value is copied to the working register and the LSB is copied to the corresponding registers, which are both 8 Bits and are well within the ADC limits.

Note: This assessment does not include consideration of environmental interference to the A/D – or D/A – converters, e.g. temperature variation.

3.1.1.3. Numerical errors

```
;           PIC Flag Definitions
#DEFINE          CARRY_FLG    STATUS,C     ; Carry flag      (STATUS)
#DEFINE          ZERO_FLG     STATUS,Z     ; Zero flag (STATUS)
```

The quantisation, rounding and calculation errors are estimated by using the Carry and Zero flag.

NOTE: The deviation of measured values arising from the digital unit will be typically much lower than 50 % of the smallest deviation of indication. Deviations arising from other sources (e.g. sensor) are expected to be dominant.

3.1.1.4. Measuring operation

All the registers used in the measuring operations are 8 Bits and the range violations are controlled by the carry and Zero flag of the microcontroller. The successive measuring operations are done within milliseconds.

NOTE: This timing requirement may not be applied to output signals which are explicitly claimed by the manufacturer to be not safety-relevant.

3.1.1.5. Special state indication

3.1.1.5.1. Fixed and transportable apparatus

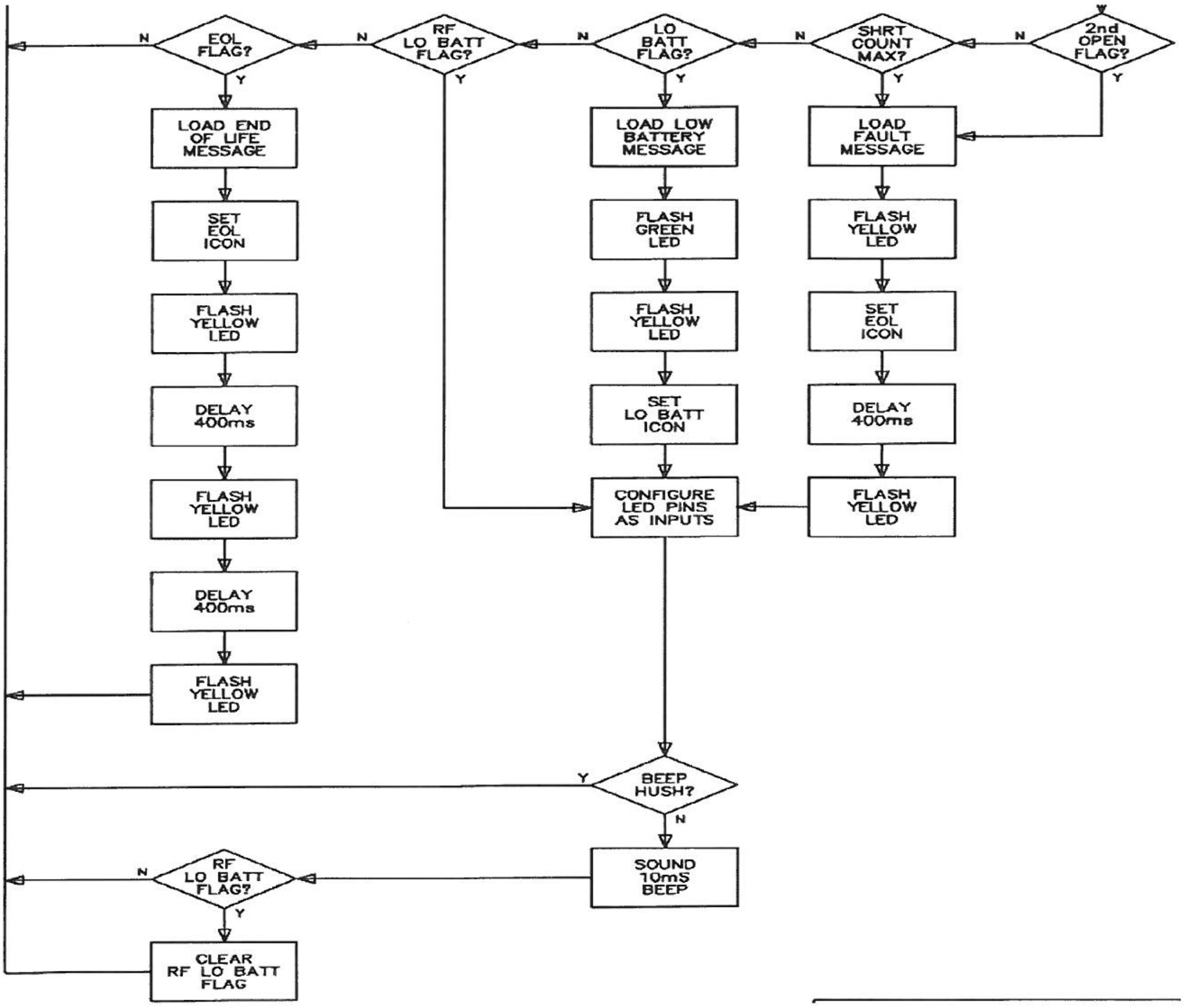
a) Control units
The signals are transmitted using an RF module and different modules are communicated to each other when the CO level goes higher than the set level.

b) Gas detection apparatus intended to be used with control units
Special state of the gas detection apparatus is transmitted to the control unit continuously.

c) Apparatus having self-contained sensors
The signals are transmitted using an RF module and different modules are communicated to each other when the units are in the fault state or have higher CO set level.

3.1.1.5.2. Portable apparatus

The fault signals are continuously indicated by the LEDs, Display and Sound Horn.

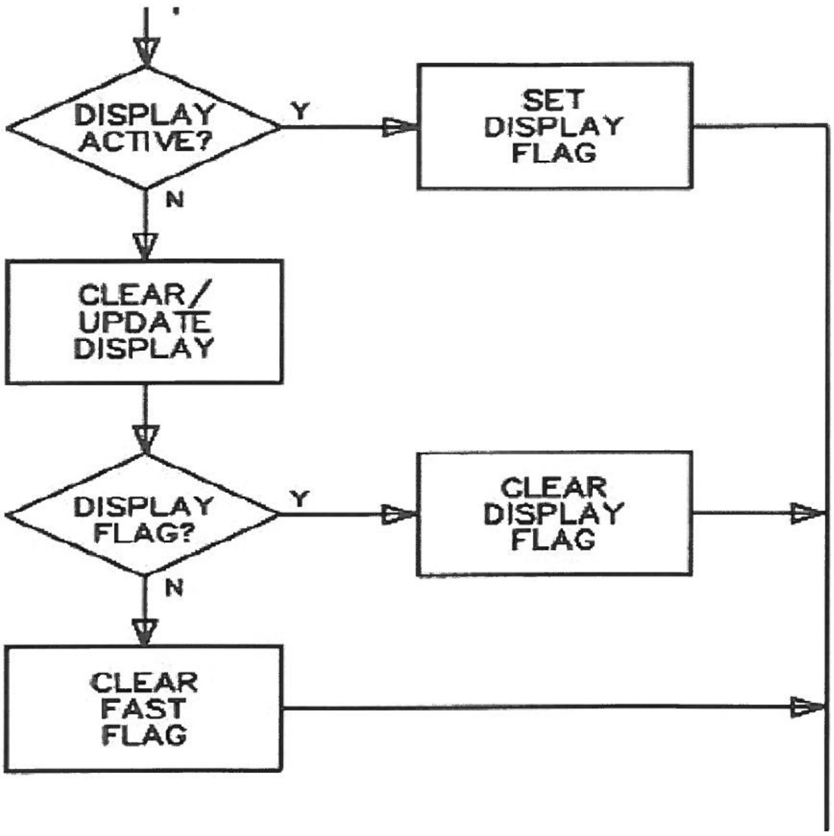


The special state "fault" signals are shown above in the flow chart.

3.1.2. Displays

3.1.2.2. Indication of messages

If it is intended to indicate messages on a display the flowchart below is followed:



a) To Display all active messages simultaneously or a consolidate signals has been generated by using Display flags signals.

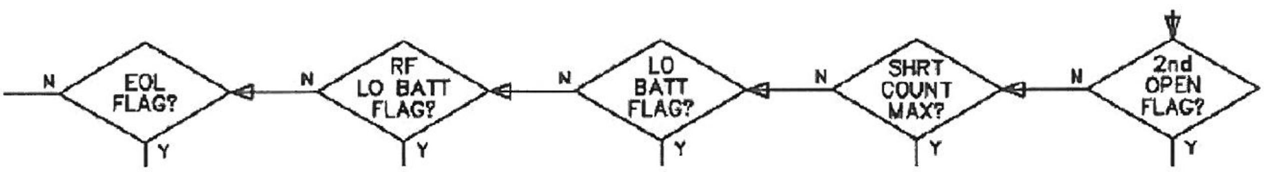
b) Different flag signals have been generated for display.

c) The display is updated regular basis.

NOTE: It is recommended that the manufacturer defines an appropriate set of messages in order to enable the user an easy identification of alarms, special states, etc.

3.1.2.3. Indication of measured values

Different indication signals are generated for different measured values.



3.1.3. Software

3.1.3.1. General

The documentation provided for the software evaluation gave the required information for the software development process in order to show compliance with EN 61508-3.

The software used for Ei208 is written in PIC assembly language, which is based on 35 RISC instructions and the MPLAB complier was used to compile and generate the hex codes for the Ei208. No operating systems were used and all code are written in assembly language which gives the software full control over the timing, peripherals in the microcontroller and memory access. The libraries and header files for the controller (PIC16F684 and PIC16LF1823) are written by microchip for the MPlab compiler.

Header files used in the software.

#include <C:\Program Files\Microchip\MPASM Suite\p16F684.inc>
#include <C:\Program Files\Microchip\MPASM Suite\p16LF1823.inc>

3.1.3.2. Re-used or commercial operating systems

3.1.3.2.1. Requirements

No operating system is used and the software has got full control of the peripherals of the microcontroller. All updates are done only by the manufacturer.

3.1.3.2.2. Use of operating systems without validation of functional safety

No operating system is used and the software has got full control of the peripherals of the microcontroller.

3.1.3.3. Software requirements

a) Revision level of the software is printed on the base label.

b) It shall not be possible for the user to modify the software function. It shall be impossible to change the program code under any operating conditions. Upgrades shall only be possible under the control of the manufacturer.

The program code can only be changed and read by the manufacturer. It is read protected in the configuration setting of the microcontroller.

```
CODE_PROTECT     equu     ON
```

Configuration setting of the PIC16F684 is stated below.

```
__CONFIG   _FCMEN_OFF & _IESO_OFF & _BOD_OFF & _CPD_OFF & _CP_ON  & _MCLRE_ON &
_PWRTE_ON & _WDT_ON & _INTOSCIO
```

_CP_ON - implies code protection ON, no one can read the code written in the controller. In order to alter the program the manufacturer has to re write the entire code to the microcontroller

The configuration setting of the PIC16LF1823 is stated below.

```
__CONFIG _CONFIG1, _FCMEN_OFF & _IESO_OFF & _CLKOUTEN_OFF & _BOREN_OFF &
_CPD_OFF & _CP_ON  & _MCLRE_ON & _PWRTE_ON & _WDTE_ON  & _FOSC_INTOSC
__CONFIG _CONFIG2, _LVP_OFF & _BORV_19 & _STVREN_ON &_PLLEN_OFF &_WRT_ALL
```

_CP_ON - implies code protection ON.

c) CO sensitivity settings are programmed into device ROM during production, using a specialist programmer, and are not accessible to the user. Also, these parameters are checked during the production process as part of an overall sensitivity check on the product.

d) Control flags are set or re-set in each program cycle.

3.1.3.4. Requirements for software documentation

The software documentation shall include:

Clause	Description	Pass
a.	Designation of the apparatus to which the software belongs;	Yes
b.	Unambiguous identification of program version	Yes
c.	If applicable, version the operating system	Yes
d.	If applicable, versions of libraries	N/A
e.	Any software modification provided with the date of change and new identification data;	Yes
f.	Documentation of the software development process (modification included, if applicable) according to 4.3.5;	Yes
g.	Source code;	Yes
h.	Functional description;	Yes
i.	Software structure (e.g. flow chart, Nassi-Schneidermann diagram).	Yes

```
;=============================================================================
;
;                              REVISIONS
;
; Rev   Description                                  Date Approved
;
; A     First Testing                                13 Aug 10
;
; B     Last flag check added to test mode. Delay     2 Sept 10
;       added after initial POR LED flash
;
; C     LED flash reduced to 10ms                    15 Sept 10
;
; D     RF cancel messages only sent in fast mode    15 Sept 10
;
; E     Beep hush only active if unit beeping        21 Sept 10
;
; F     Display function added. Beep hush bug fixed  22 Sept 10
;
; G     Extra zero check added to floating point 14 Oct 10
;       conversion routine. Soft start removed
;       from remote test function
;
; H     Fast flag set during memory LED flash set    18 Oct 10
;
; J     Display code modified for correct display    19 Oct 10
;
; K     Delays added to allow RF module to support   19 Nov 10
;       both RF communication modes
;
; L     Interrupt flag added to prevent clock         6 Dec 10
;       speed-up during module fast mode. Display
;       blanked out if CO less then limit.
;
; N     ICT test mode added                          20 Dec 10
;
; P     Code added for PIC16LF1823. Interrupt wakeup 22 Dec 10
;       handling changed
;
; Q     ICT output level 2 changed. Continuous horn  31 Jan 11
;       test mode was green LED now red LED. Sensor
;       categories added.
;
; R     EOL indicator suspend mode added             11 Feb 11
;
; S     Display constant adjusted. Low battery LED   17 Feb 11
;       flash modified. Display Iq reduced.
;
; T     EOL button test flash was green now yellow.  25 Feb 11
;       EOL disable was 30 days now 24 hrs.
;
; U     Port A 0/1 ICT outputs reversed               4 Mar 11
;
; V     Fine tuning of calibration constants         25 Mar 11
;
; V2    Fast test mode enabled. Memory less than      4 May 11
;       current level check added
;
; V3    Fine tuning of calibration constants         13 May 11
;
; V4    Moved ICT analog tests to ICT Phase 2        27 May 11
;
;
; 0     Released                   ECN 3888           7 June 11
;
;
; 1a    EOL beeps added. Beep silence period         15 June 11
;       increased to 24 hrs. CO measurement
;       suppressed during radio Tx.
;
```

The software documentation which contains the revision control and modifications to the software is shown above.

3.1.3.5. Requirements for the software development process

3.1.3.5.1. General

The software development process is carried out as stated below.

Scope: This procedure covers the revision and change control process on software programs for EI products. This does not cover production program revision control.

Procedure: When a new project is initialled a product specific folder will be opened on the "Eng folder Server" within this folder will reside the source code for the product while in development.

As well as source code each product family will have a series of flowcharts outlining the software architecture and software design flow.

Once the initial source code has been developed all changes and revision will be clearly outline within the first page of the source code. This will consist of the current revision, the previous revisions and a short description of the changes.

Revision Numbering: In the development stage "Pre-Release" all software will be either "Alpha" or "Alpha, Numeric" revision controlled.
Once the product software is production "Released" the revision will move to "Numeric" or "Numeric, Alpha". Status. In the case of Production released software the initial released revision will be Rev 0. This release will be processes via our internal ECN process and any and all subsequent releases will be control in the same manner.
Changes to released software will only be released after engineering validation, as such it is recognised that in some cases this may take one or two iterations so the following rules apply. Once a software change is initiated to a release program it is assigned the next incremental "Alpha" number in addition to its "Released" "Numerical" rev (i.e. Rev 0A) , If in the validation process a second iteration is found to be required the software revision will be incremented by the next "Alpha" number (i.e. Rev 0B), this process will continue until the software is ready to be released to production. At which stage it will be released under ECN to the next revision (i.e. Rev 1)

Software Location: While in development the software will be located in the source file folder located in the "product Design specific folder" (i.e. EI208 / Source code). Once the software reaches a "released" status the source code will move to the Drawings folder on Engineering Server.

Local copies of source code are prohibited, access to the source code must at all times be via the server. This is to avoid multiple parallel changes been made to software files.

3.1.3.5.2. Software requirements specification

The software requirements are clearly described in the Flow chart for each interface including: hardware devices, human interfaces, communications interfaces.

The requirements for the software are documented in sufficient detail in natural language. Practical graphical schemes such as flow charts are used for the sake of precision and each requirement of the software is identified and traced unambiguously.

All the hardware fault signals are clearly stated in the flow chart and also the error signals which are used to identify the hardware faults.

3.1.3.5.3. Software architecture

3.1.3.5.3.1. Architecture

The software architecture designed based on

a) Hardware architecture and Software requirements specification and is clearly stated in the Flow Chart.

3.1.3.5.3.2. Tools and coding standards

The language used is Assembly language for 16fXXX series PIC microcontroller. The complier used is Mplab IDE and Mplab assembler. The microcontroller and compiler are developed by microchip and are well established and are easily available. The Mplab IDE is free and can be download from the microchip website.

a) Range of functions and performance – Microchip microcontrollers are RISC microcontrollers and can be used with wide range of peripheral modules as required and also with a variable processor frequency.

b) Operating experience – PIC microcontrollers are used for a wide range of applications.

c) Updates, release notes – When updated the Mplab IDE and microcontroller the latest release notes are available readily.

d) Error lists – Every PIC microcontroller datasheet comes with an error list and revision history.

e) References - http://www.microchip.com/

f) Publications related to the tool – Publications and articles related to the product can be found on the Microchip website.

g) Experience with similar products of the manufacturer – Microchip has a wide range of microcontrollers which varies from 10F,12f, 16F,17F,18F,24F,32F series of microcontroller and other peripheral devices that can be easily interfaced to microcontrollers.

h) Market presence of the manufacturer – Microchip is well established and with has a vast number of product ranges.

The programming language used is a Low-level language, the assembly language used is written by microchip for the PIC microcontroller. PIC microcontrollers are RISC microcontrollers and have 35 assembly instructions, so the program written is platform oriented and is written for a specific microcontroller. The source code is well documented and can be read easily

3.1.3.5.4. Software design

The software design is carried out in a structured manner and demonstrates how the software requirements are implemented. The software design was found to adhere to the following rules which are included in the software developers programming standard.

a) The software components are decomposed into software modules.

b) Functionality of the software modules are clearly specified.

c) Specification of data structures and assignment to the software modules are carried based on the functional requirements of the apparatus, complete and free of contradictions;

d) Unambiguous interfaces between the software modules are specified.

e) Design of the software modules are done in a modular manner.

f) A test program has been developed to check all relevant functions of the product. The test program is modified to reflect the changes in the product (as they occur). The product software is checked and verified during execution of the test program.

The software design is carried out according to the following rules:

g) Keep the number of possible paths through a software module small;

h) Avoid complex branching;

i) Avoid complicated calculations as basis for branching or loop conditions;

j) Software modules shall usually communicate with other software modules via their interfaces - where global or common variables are used:

1) They shall be well structured;

2) They shall be declared in one central module;

3) Access shall be controlled;

4) Their use shall be justified in each instance;

5) Competing write- and read-access by parallel running processes shall be avoided;

All variables and constants are defined at the beginning of the program.

3.1.3.5.5. Coding

The source code met the following standards.

Clause	Description	Pass
a.	Readable, understandable, and testable	Yes
b.	Implement the design of the software modules	Yes
c.	Satisfy the requirements of the coding standards	Yes
d.	Implement each software requirement	Yes

3.1.3.5.6. Software test

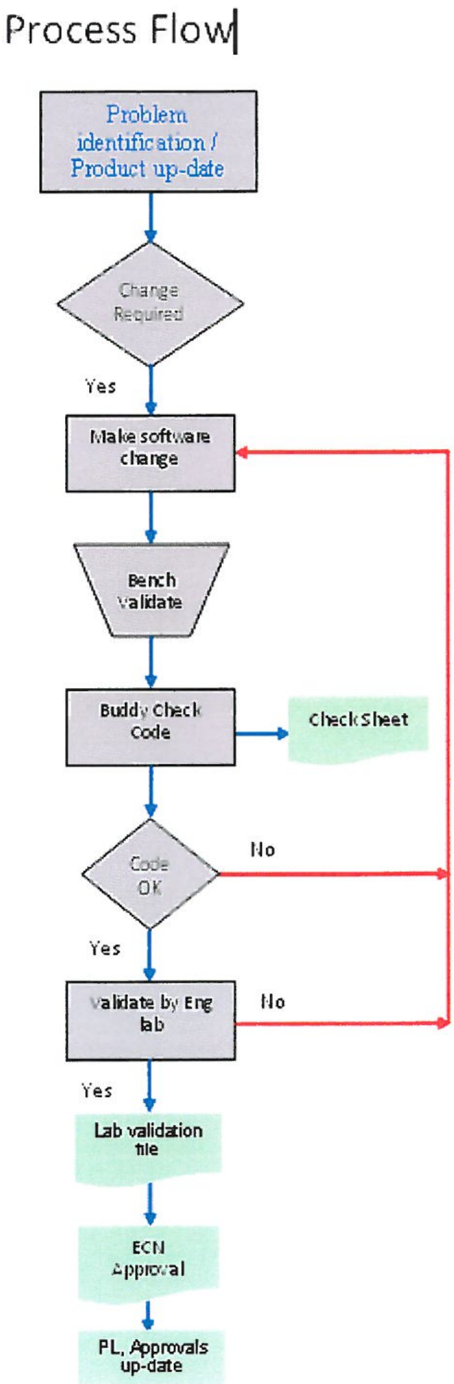
A test program has been developed to check all relevant functions of the product. The test program is modified to reflect the changes in the product (as they occur). The product software is checked and verified during execution of the test program.

3.1.3.5.7. Hardware-software integration test

The test program created by the manufacturer is designed to carry out the software hardware integration test. Different hardware units such as battery, Horn etc are tested during the normal operation of the unit. The product is checked and verified during execution of the test program.

3.1.3.5.8. Validation and Software modification

Software modification and validation is carried out as seen in the flow diagram below.



3.1.4. Hardware

The Parameters and variables are permanently stored in the EEPROM and the data stored in the EEPROM an so remain after the product is switched off or in the special state when the supply voltage is removed.

```
;               ==== READ_EEPROM Subroutine Code ====
;               Code for READ_EEPROM Subroutine
;
READ_EE_CODE        CALL  READ_EE_SUB
                GOTO  SUB_RETURN
;
;               ==== End READ_EEPROM Subroutine Code ====
;
;
;               ==== READ_EE_SUB Subroutine ====
;               Reads data from EEPROM, leaves in W register
;               Address must be supplied in W register
;
;
        IF      DEVICE == PIC16F684
;
READ_EE_SUB BCF     INTCON,GIE   ; Disable interrupts
;
                BANKSEL     EEADR
                MOVWF EEADR
                BSF   EECON1,RD
                MOVF  EEDATA,W
                BANKSEL     BASE_BANK
                MOVWF EE_RETURN
;
                IF    RF_MODE_2 == ON
;
                BSF   INTCON,GIE   ; Enable interrupts again
;
                ENDIF ; RF_MODE_2
;
                RETURN
;
        ENDIF ; DEVICE
;
        IF      DEVICE == PIC16LF1823
;
READ_EE_SUB BCF     INTCON,GIE   ; Disable interrupts
;
                BANKSEL     EEADRL
                MOVWF EEADRL
                BCF   EECON1,CFGS ; Deselect Config space
                BCF   EECON1,EEPGD      ; Point to DATA memory
                BSF   EECON1,RD
                MOVF  EEDATL,W
                BANKSEL     EE_RETURN
                MOVWF EE_RETURN
;
                IF    RF_MODE_2 == ON
;
                BSF   INTCON,GIE   ; Enable interrupts again
;
                ENDIF ; RF_MODE_2
;
                RETURN
;
        ENDIF ; DEVICE
;
;               ==== End READ_EE_SUB Subroutine ====
;
;
;
;               ==== WRITE_EE Subroutine Code ====
```

```
;            Code for DLY_mS Subroutine
;
      IF     DEVICE == PIC16LF1823
;
WRITE_EE_CODE     CALL   WRITE_EE_SUB
             GOTO   SUB_RETURN
;
      ENDIF ; DEVICE
;
;            ==== End WRITE_EE Subroutine Code ====
;
;
;
;            ==== WRITE_EE_SUB Subroutine ====
;            Writes data in W register to EEPROM
;            Address must be preloaded in EEPROM_ADDR register
;            Data is written to EEPROM_DATA register by subroutine
;            before write process is started
;
;
;
      IF     DEVICE == PIC16LF1823
;
WRITE_EE_SUB
;                   Check if write enable was selected
             SKP_IF_TRUE WRITE_EN_FLG
             RETURN
;
;                   Write enable was selected, deselect it
;
             BCF    INTCON,GIE  ; Disable interrupts
             BCF    WRITE_EN_FLG       ; Clear write enable flag
             MOVWF EEPROM_DATA ; Save write data
;
;                   Set up write loop count
             MOVLW EE_WRITE_SET
             MOVWF EE_WRITE_CNT
;
;                   Load address to be written to
EE_WRITE_LP MOVF   EEPROM_ADDR,W
             BANKSEL      EEADR
             MOVWF EEADR
;
;                   Load data to be written
             BANKSEL      BASE_BANK
             MOVF   EEPROM_DATA,W      ; Reload value to be written
             BANKSEL      EEDAT
             MOVWF EEDAT
;
;                   Make sure previous write completed
             BR_IF_TRUE   EECON1_WR,$-1
;
;                   Write EEPROM
             BCF    EECON1,EEPGD       ; Select EEPROM for write
             BSF    EECON1,WREN ; Enable writing.
             MOVLW 55H          ; Required to unlock write.
             MOVWF EECON2
             MOVLW 0AAH
             MOVWF EECON2
             BSF    EECON1,WR   ; Start write operation.
             BCF    EECON1_WREN ; Disable writing.
;
;                   Wait for write to finish
             BR_IF_TRUE   EECON1_WR,$-1
;
;                   Read back written value
             BANKSEL      BASE_BANK
             MOVF   EEPROM_ADDR,W      ; Read back written value
             CALL   READ_EE_SUB
```

```
            SUBWF EEPROM_DATA,W      ; Subtract from correct value
            SKPNZ                 ; Zero?
            GOTO  EE_BYTE_GOOD
;
;                Values do not match, repeat read/write cycle
            DECFSZ      EE_WRITE_CNT,F     ; If not, advance count
            GOTO  EE_WRITE_LP ; and rewrite EEPROM
;
;
;                Set EE address pointer to dummy location
EE_BYTE_GOOD     MOVLW EE_STDBY_ADDR
            MOVWF EEADR
;
            BANKSEL      BASE_BANK
            BSF   INTCON,GIE  ; Enable interrupts again
            RETURN
;
      ENDIF ; DEVICE
```

3.1.5. Digital data transmission between components of apparatus

Data transmission is carried between the microcontroller and the RF module. Transmission errors, repetitions, deletion, insertion, resequencing, corruption, delay and masquerade were considered in order to ensure reliable data transmition.

```
;           RF Module Comm Settings
NO_PIC_BITS equ   D'8'          ; Number of data bits transferred
PIN_CHNG_SET      equ   D'50'         ; Allowed delay for data bit length
RF_ALM_CNT_SET    equ   D'3'          ; Starts incoming alarm counter
RF_BT_CNT_SET     equ   D'3'          ; Starts incoming button test counter
NO_TX_TRIES equ   D'5'          ; Number of attempts to talk to module
RF_DLY_SET  equ   D'10'         ; Time CO measurement is suppressed

PIC_DATA_OUT      equ   40H   ; Registers for transferring data
PIC_DATA_IN equ   41H   ; between head and RF module
;
RF_CLK_CNT  equ   42H   ; Sets interval between module wakeups
;
RF_ALM_CNT  equ   43H   ; Set by incoming RF alarm message
;
RF_BT_CNT   equ   44H   ; Set by incoming RF alarm message
;
FAST_TEST_CNT     equ   45H   ; Sets duration of fast test mode
;           Messages sent from head to RF module
HD_ALM_CODE equ   038H  ; B'00111000'     ; Head is in alarm
HD_BT_CODE  equ   057H  ; B'01010111'     ; Head is being button tested
HD_CNCL_CODE      equ   094H  ; B'10010100'     ; Head has stopped alarming
HD_FLT_CODE equ   0E6H  ; B'11100110'     ; Head has a fault
HD_LB_CODE  equ   062H  ; B'01100010'     ; Head is in low battery
HD_EOL_CODE equ   0F4H  ; B'11110100'     ; Head has reached EOL
;
;
;           Commands sent from RF module to head
RF_FIRE_CODE      equ   038H  ; B'00111000'     ; Fire alarm command
RF_CO_CODE  equ   0C4H  ; B'11000100'     ; CO alarm command
RF_BT_CODE  equ   057H  ; B'01010111'     ; Button test command
RF_CNCL_CODE      equ   094H  ; B'10010100'     ; Alarm cancel command
RF_LB_CODE  equ   062H  ; B'01100010'     ; Remote low battery command
RF_RT_CODE  equ   023H  ; B'00100011'     ; Remote test message command
                                    ; (not sent - RF_BT_CODE sent
                                    ; instead)
RF_HUSH_CODE      equ   0BAH  ; B'10111010'     ; Remote hush message command
RF_SELF_TEST      equ   09DH  ; B'10011101'     ; Remote monitoring command
```

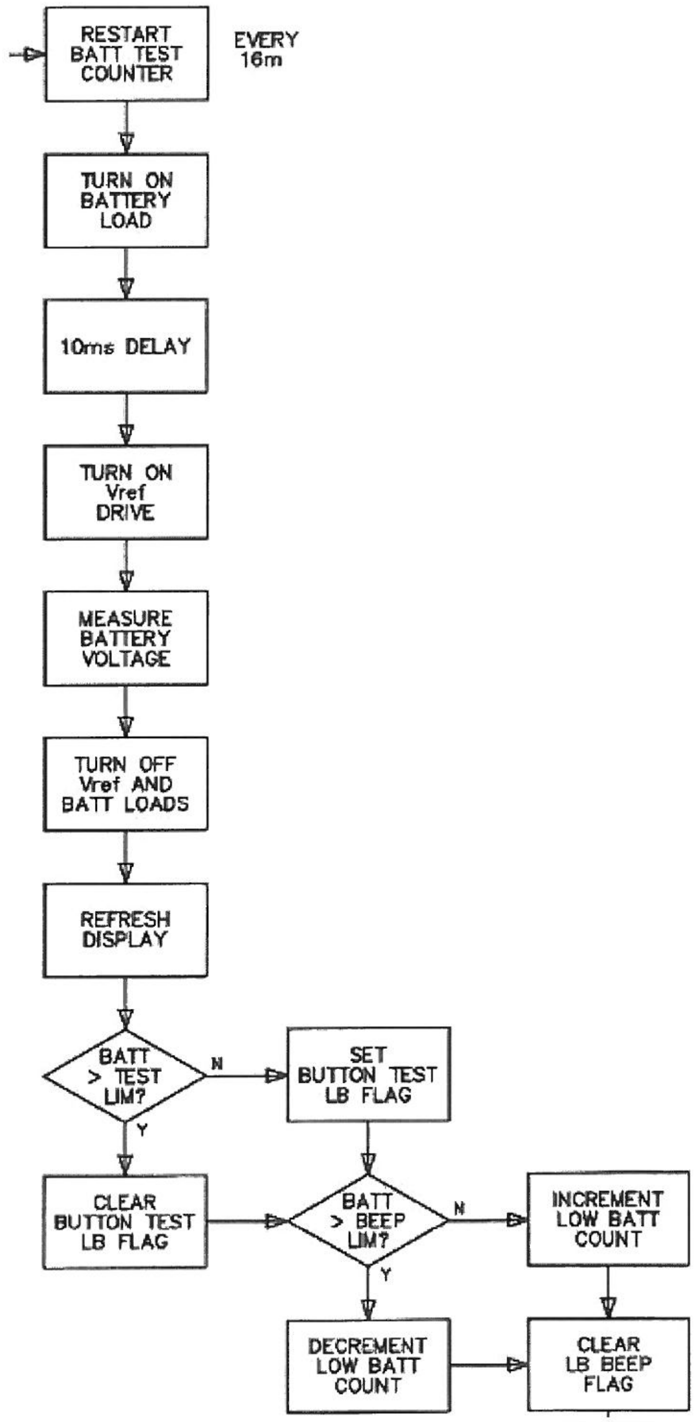
The data communication between the modules was checked and the communication channel works properly as stated in the software (ref TR/11/265).

3.1.6. Test routines

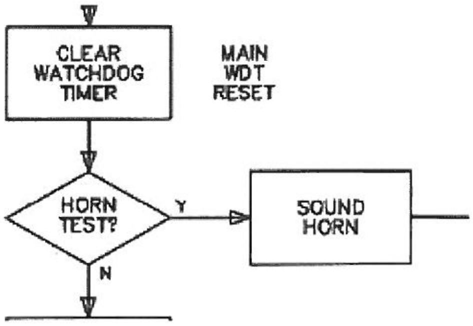
A test program has been developed to check all relevant functions of the product. The test program is modified to reflect any changes in the product (as they occur). The product software is checked and verified during execution of the test program.

The following minimum tests shall be performed by the apparatus:

a) Battery voltage is monitored every 16m and it is checked against the Vref voltage of the microcontroller.

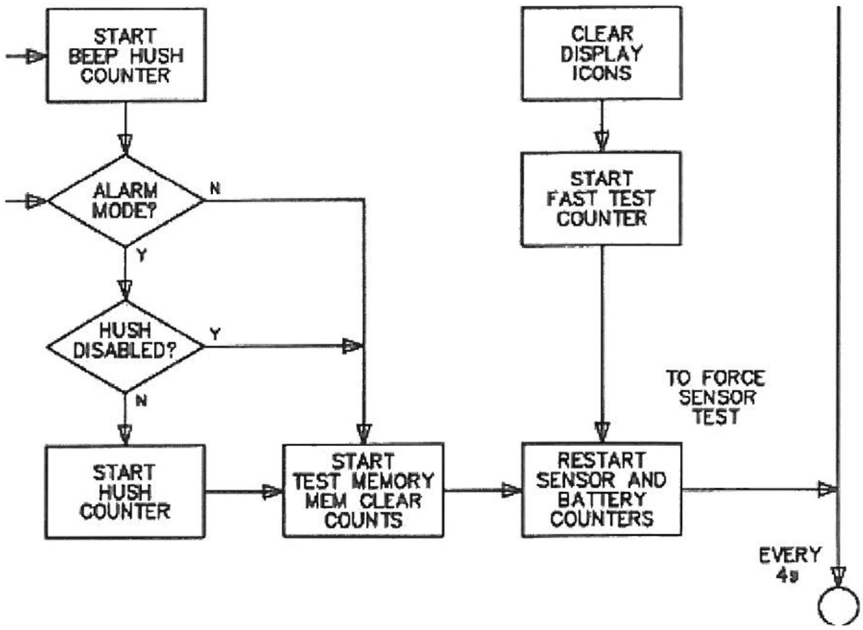


b) All visible and audible output functions are tested automatically after starting operation.



c) The sensor and battery are monitored using separate counter registers and are updated quite often in the program.

d) Memory counter is used to monitor the memory operations.



3.1.7. Instruction manual

The following information was included in the instruction manual

Clause	Description	Pass
a.	Instructions how fault and status outputs shall be wired and monitored for safe operation	Yes
b.	Relevance of the alive signal or confidence signal for safety	Yes
c.	Description of all special states including cause, signalling and termination	Yes
d.	Description of all messages available to the user and methods for interrogation	Yes
e.	Behaviour of displays, measuring outputs and all signal outputs at under scale or over scale	Yes
f.	Minimum refresh rates of all safety relevant output signal(s)	Yes
g.	All user changeable parameters and their valid ranges	Yes
h.	Life time of data storage if a back-up battery is used for preserving the data content of parameter memory when the supply voltage is removed	Yes
i.	Instruction for fixed or transportable apparatus that the apparatus shall be re-started after end of maintenance work if the power-on self test has been skipped for this maintenance work;	Yes
j.	Instruction for fixed or transportable apparatus that the apparatus shall be re-started after end of maintenance work if the power-on self test has been skipped for this maintenance work;	Yes
k.	Instruction that after changing parameters by using an external device (e.g. PC), the user shall check the correctness of the parameter settings 1) By checking the parameter settings at the gas detection apparatus; or alternatively 2) By reading back the parameters from the gas detection apparatus and manually verifying the received values;	Yes
l.	If compliance with SIL 1 is claimed	N/A

3.2. Clause 5 – Test of the digital unit

3.2.1. General

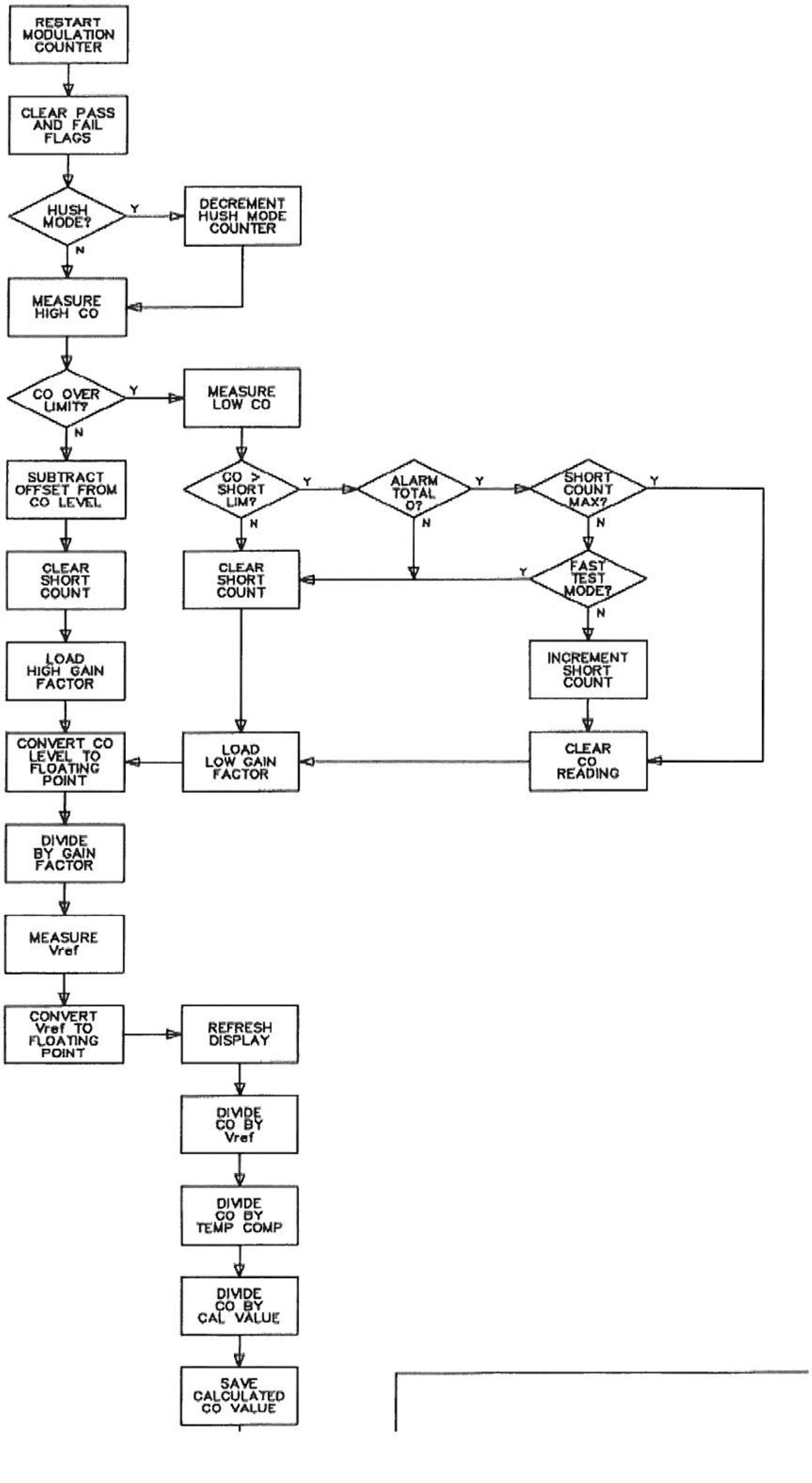
The testing of the digital units is part of the testing of the apparatus which is needed for compliance with the performance requirements. It is divided into two phases. In the first phase the functional concept of the digital unit is inspected with regard to meeting the requirements for the design and for the software development process (Clause 4) within the framework of the entire apparatus. The second phase comprises a performance test of the digital units. It shall detect errors that can occur when transferring the design concept into hard- and software.

3.2.2. Verification of functional concept

Functional concept analysis and evaluation depend on the documentation from the manufacturer. The verification shall be performed by using the following list.

a) Functional description of the digital unit which is preferably structured like Clause 4:

1) Measuring sequence
2)



2) Estimation of numerical errors – Various methods such as checking the carry flag & zero flags of the microcontroller.

3) Possible special states – All states are predefined and accesses to the states are done by setting corresponding flags.

4) Representation of measured values and messages – Measured values and messages are compared with the predefined values set at the beginning of the program.

5) Generation of alarms and signals – Alarm signals are generated by setting corresponding flag signals in the program.

6) Extent and realisation of remote data transmission - remote data transfer is done through the RF module.

7) Extent and realisation of test routines – A test program is created to check all relevant functions of the product.

b) Hardware description:

1) Design of the digital unit – Circuit diagram and relevant datasheet.

2) Block functional description of the digital unit – Flow chart and circuit diagram which describe the function of each Block/Module.

3) Resolution, errors and input/output ranges of A/D- or D/A-interfaces – Described in the data sheet of the microcontroller.

4) Specification of interfaces between functional parts – Described in the circuit diagram of the unit.

c) Software documentation – Flow chart and each instruction of the assembly language program is well commented and can be read easily.

3.2.3. Performance test

The performance test is done as stated below

1. CO Testing

Place two units in 30ppm CO for two hours. Units must not alarm. Increase CO to 60ppm. Red LED must immediately start flashing at the slow rate (once per 2 seconds), display must indicate CO level. Unit must alarm after 72 minutes. Remove CO. LED should stop flashing and display go blank below 30ppm. Allow to settle for 20 minutes.
Set CO to 100ppm. Red LED must immediately start flashing at the medium rate (once per second), display must indicate CO level. Unit must alarm after 18 minutes. Remove CO. LED should stop flashing and display go blank below 30ppm. Allow to settle for 20 minutes.
Set CO to 200ppm. Red LED must immediately start flashing at the fast rate (twice per second), display must indicate CO level. Unit must alarm after 40 seconds. Remove CO. LED should stop flashing and display go blank below 30ppm. Place unit in the small piston-operated box, and place this box in 5000ppm of CO. When the CO in the outer box has stabilised, open the inner box. Unit must alarm after 40 seconds and display should show approx. 900ppm.

2. Button Test

Press button on unit. Horn should sound at a low level for the first burst, then go to full power for the second burst. Display should indicate "000", check mark, and full battery. Green LED should flash at the medium rate. Remove the sensor from the unit and button test. Horn should <u>not</u> sound, yellow LED should flash at the medium rate, and display should indicate "REPLACE UNIT". Apply a low battery voltage and button test. Horn should sound, yellow LED should flash at the medium rate, and green LED should flash at the fast rate. Display should indicate"000" and low battery. Record voltage at which this point is reached. It should be 50-100mV above the low battery beep point. Program a unit to EOL and button test. Horn should sound, yellow LED should flash at the medium rate, and display should indicate "REPLACE UNIT".

3. Hush Modes
Place unit in 70ppm and allow to alarm. Button test unit. Alarm should hush for 5 minutes, then resume sounding. Button test unit again. Unit should not hush. Remove CO, allow to settle for 20 minutes. Place unit in 200ppm and allow to alarm. Button test unit. Unit should not hush. Remove sensor and wait for unit to start beeping. Button test unit. Beeps should stop for 12 hours. Apply a low battery voltage and wait for unit to start beeping. Button test unit. Beeps should stop for 12 hours. Program a unit to EOL and wait for the yellow LED flashes to appear. Button test unit. Flashes should stop for 12 hours.

4. Memory Function
Place unit in 70ppm and allow to alarm. Remove CO, allow to settle for 20 minutes. Red LED should flash for 4 seconds every 48 seconds at the maximum rate seen during alarm. Button test unit. Display should indicate maximum CO seen during alarm mode. Red LED should flash for 20 seconds at the maximum rate seen during alarm. Press and hold button for 20 seconds. Memory should clear.

5. Low Battery
Measure low battery trip point by powering up the unit at various battery voltages, and looking for a beep after power on. Power the unit with a good battery voltage for 2 days, then reduce the voltage to under the low battery level. Measure how long it takes to go into low battery, should be approx. 24 hours. Allow to beep overnight, then increase the voltage to a good battery level. Beeps should stop after 4 hours.

6. End of Life
Program units with standard software. Record the time they were powered up. Read the end of life EEPROM location over the course of several weeks. The count should decrease by one every 4hrs 46 minutes, it will also drop by one every time the unit is powered up. Program more units with modified software where the final timing loop before decrementing the EEPROM is skipped. This will cause the EEPROM to decrement every 67 seconds instead of 4hr 46 min. These units will reach end of life in 10 days instead of 7 years.

7. Fast Test Mode
Place unit into a bag and inject 50-100ppm of CO. Button test unit. 4 seconds after the button test, the horn should sound for 8 seconds.

8. Power Consumption/ Head Removed Indicaton
Power unit through a 1 ohm resistor with the fast logging volt meter across it. Leave powered up for several hours. Record the total current consumption. It should be less than 7 uA.

Conclusion

The software assessment of performed on the Ei208 software documentation provided is considered to show that the control complies with the relevant requirements of BS EN 50291: 2010 Clause 4.6 – Electrical apparatus for the detection of carbon monoxide in domestic premises.

KEY TO TERMS USED IN REPORT		
YES	- Complies. Tested by BSI engineers at BSI laboratories.	
YES1	- Complies. Witness tested by BSI engineers in manufacturer's laboratory.	**Not UKAS Accredited**
YES2	- Complies. Tests carried out by manufacturer; results accepted by BSI.	
N/A	- Not applicable to design under consideration.	
NTx	- Not tested due to similarity to previously tested item; reference earlier test report.	
NO	- Non compliance.	