SOFTWARE FUNCTIONAL SPECIFICATION
=================================

Joystick Module for Risc OS Black

        Issue: 1/3P95
       Author: W.Turner
         Date: 30th January 1995
Drawing Number: 1303,002/FS/3P95

   Last Issue: C
_____

1. History

Version	Date	Notes
0.01	30-03-94	Initial draft
0.02	12-05-94	SWI interface changed after FS Review
0.03	08-06-94	Changed to reflect support of ADVAL & OSBYTEs
0.04	03-11-94	Added analogue calibration support
0.05	30-01-95	Added description of !Calibrate support application

2. Outstanding Issues
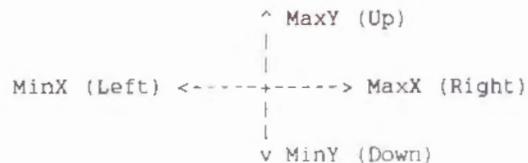None.

3. Product Overview

The joystick module for Risc OS Black is to provide support for up to two
joysticks on a machine fitted with an appropriate interface board. It will
support both Atari-style (digital, one fire button) and PC-style (analogue,
2 fire buttons) joysticks.
   To enable a wide variety of analogue joysticks to be used, an application
will be supplied that allows the joysticks to be calibrated from the
desktop. This prevents the need for each program to include its own
calibration routines.

4. Concepts & Definitions
As per the PC-style joystick standard, the voltage minima & maxima are as
defined in the diagram below, for analogue joysticks. In the software, the
default range will be -127 to +127, in order to match the existing SWI.
Alternatively, 16-bit values can be returned, depending on a SWI flag (see
section 6).
   Note that not all 'PC-style' analogue joysticks adhere to the standard,
but these are corrected in the joystick module so that they appear to comply
to the standard.

```
                      ^ MaxY (Up)
                      |
                      |
     MinX (Left) <------+------> MaxX (Right)
                      |
                      |
                      v MinY (Down)
```

For digi   (atari-type) joysticks, Left = -64, Right   64, Down = -64 and
Up = 64. Any other value is assumed to be the 'centred' position.

5. User Interface
There are no *commands supported by this module. Note that the module
determines the interface type at initialisation, so dynamic configuration of
the interface is not possible.

An application called !Calibrate is supplied with the joystick module.
This allows the user to calibrate analogue joysticks before use (this is
necessary because of the variations in joystick design). Once this
application has been run by the user, the returned analogue values from the
Joystick_Read SWI will cover the full range (0-65535), rather than an
arbitrary subset.

6. Programmer Interface
The new module is intended to use a very similar (and
backwards-compatible) calling interface to the module developed for the
A3010. It also defines two new SWIs to enable calibration of analogue
joysticks.

SWI: Joystick_Read (&43F40)

Returns the state of a joystick

On entry
     R0   bits 0-7:   joystick number
          bits 8-15:   reason code: 0 -> return 8-bit values
                                     1 -> return 16-bit values (only applicable
                                                                to analogue)
          bits 16-31: reserved (0)

On exit
   If Reason code = 0 (return 8-bit values)
     R0

          Byte   Value
           0   Signed Y axis value, range -127 to +127
           1   Signed X axis value, range -127 to +127
           2   Switch statuses, starting in bit 0.
           3   Reserved

   If Reason code = 1 (return 16-bit values)
     R0   bits 0-15: Signed Y value in the range 0 to 65535
          bits 16-31: Signed X value in the range 0 to 65535

     R1   bits 0-7: Switches (eg fire buttons) as specified in RO3 PRMs
          bits 8-31: Reserved

Interrupts
     Interrupt status is unaltered .
     Fast interrupts are enabled

Processor mode
     Processor is in SVC mode

Re-entrancy
     Not defined

Use
   This SWI is used to obtain the state of the requested joystick.

   If the hardware interface is configured for an analogue joystick, then the initial call to the SWI will always return 'centered', ie X=0, Y=0, no switches closed. The analogue routine makes use of a VSync-driven periodic routine that updates a 'magic location' with the latest readings. This allows the SWI to quickly read these locations & return, without the unacceptable overhead of having to wait for a conversion to complete. The digital routine simply looks directly at the hardware for the latest readings.

If the interface is configured to analogue, then the standard ADC OSBYTEs are also available (16, 17, 128, 188 - 190), as is the ADVAL command in BASIC. This is to retain backwards compatibility with the I/O Podule. Note that the 15-pin joystick connector is differently wired to the I/O Podule's ADC port, and so some kind of adaptor cable will be required to allow ADC port devices to be connected to the joystick port. Also, the ADC port only supports 2 switches, whereas the joystick port provides four (2 per joystick). For the ADVAL and OSBYTE commands, the ADC port device switch should be mapped to fire button 0.

Also, note that if both a joystick interface and an I/O podule are present in the machine, that the OSBYTEs and ADVAL command will be linked to the ADC port on the I/O podule. The SWI will still access the joystick port though.

   SWI: Joystick_CalibrateTopRight (&43F41)

   One of the pair of SWIs to calibrate an analogue joystick so that the Read SWI returns the full range of values (0 to 65535 or -128 to +128)

   On Entry
    No entry parameters

   On Exit
    All registers preserved

   On calling this SWI, the joystick should be in the top-right position.

   SWI: Joystick_CalibrateBottomLeft (&43F42)

   One of the pair of SWIs to calibrate an analogue joystick so that the Read SWI returns the full range of values (0 to 65535 or -128 to +128)

   On Entry
    No entry parameters

   On Exit
    All registers preserved

   On calling this SWI, the joystick should be in the bottom-left position.

NOTE: After one of these SWIs has been called, the ADVAL and Read SWI return errors until the other calibration SWI is called.

7. Standards
   The code is assembled with !ObjAsm.

8. Data Interchange
   There are no data interchange formats defined by this project.

9. Data Formats
   There are no new data formats defined by this project.

10.External Dependencies
    Digital Joystick board fitted to an appropriate test system.

11.Acceptance Test
    Must function on Medusa hardware. Also, in analogue mode, the Vsync-attached routine should not consume more than 1% of the CPU time in mode 13 (320x256x256, a common games mode).
    Should require less than 2Kb of memory for the module and its workspace.

12.Development Test Strategy
    A program will be written to graphically relay the values being read from the joystick to the screen. This will allow the linearity of the analogue response to be tested, as well as to ensure the joystick module allows equal range for both axes, and an acceptable response time.
    note that it may be wise to try a selection of joysticks, as resistances (and thus, overall range) will vary between them.