omegaio

Version 1.0.0 – 16 June 2016
Kit Bishop

Document History		
Version	**Date**	**Change Details**
Version 1.0.0	16 June 2016	First released version

Contents

1. Overview

omegaio is C++ program that provides command line access to the functionality provided by the **libnewgpio**, **libnewi2c** or **libarduino** libraries.

The program also provides basic scripting facilities with the input being provided on the command line and/or from standard input and/or from external files.

Notes:

- The **omegaio** program makes use of all of the following libraries:
  - **libnewgpio** library that is available and documented at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libnewgpio
  - **libnewi2c** library that is available and documented at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libnewi2c
  - **libarduino** library that is available and documented at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libarduino additionally, the Arduino library **arduino-omega** for use on the Arduino is available and documented at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/arduino_omega

omegaio consists of single program in static and dynamic forms.

This program is described in more detail in this document.

The program was developed on a KUbuntu-14.04 system running in a VirtualBox VM and uses the OpenWrt toolchain for building the code:

The toolchain used can be found at:

- https://s3-us-west-2.amazonaws.com/onion-cdn/community/openwrt/OpenWrt-Toolchain-ar71xx-generic_gcc-4.8-linaro_uClibc-0.9.33.2.Linux-x86_64.tar.bz2

and details of its setup and usage can be found at:

- https://community.onion.io/topic/9/how-to-install-gcc/22

omegaio comes with **NO GUARANTEES** ☺ but you are free to use it and do what you want with it.

2. Files Supplied

omegaio is supplied in files in a GitHub repository at https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/omegaio. This repository contains the following important directories and files:

- **omegaio.pdf** – this documentation as a PDF file
- **Makefile** – the Makefile for **omegaio** program
- **omegaio-example.txt** – an example script file that can be used as input to the **omegaio** program that demonstrates some of the scripting capabilities
- **hdr** – directory containing header (***.h**) files for **omegaio** program

- **src** – directory containing source (***.cpp**) files for **omegaio** program
- **bin** – directory containing the built program code:
    - **dynamic/omegaio** – the dynamically linked version of the program
    - **static/omegaio** – the statically linked version of the program

3. Usage and Installation

Installing and using the program is simple. It primarily consists of linking the program and if needed (see below) the library code.

3.1. Using omegaio statically linked program

To use **omegaio** statically linked program you simply need to copy the program to the Omega and run it.

3.2. Using and Installing omegaio dynamically linked program

To use **omegaio** dynamically linked program you need to copy it and all of the libraries **libnewgpio**, **libnewi2c** and **libarduino** used to your Omega and then run the program.

All of these libraries that the program uses of will need the **.so** versions of these libraries to be copied to the **/lib** directory on your Omega

Alternatively, you can copy the library to any location that may be set up in any **LD_LIBRARY_PATH** directory on your Omega. For example, I use the following for testing:

- Created directory **/root/lib**
- Copied the libraries to **/root/lib**
- Added the following lines to my **/etc/profile** file:

```
LD_LIBRARY_PATH=/root/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
```

4. Using Makefile

A **Makefile** is supplied that can be used to build the library.

4.1. Modify Makefile

The **Makefile** will need modifying:

- You **NEED** to and **MUST** change **TOOL_BIN_DIR** to the "bin" directory of your OpenWrt uClibc toolchain. E.G. make appropriate change to **<xxxx>** in:

  TOOL_BIN_DIR=<xxxx>/OpenWrt-Toolchain-ar71xx-generic_gcc-4.8-linaro_uClibc-0.9.33.2.Linux-x86_64/toolchain-mips_34kc_gcc-4.8-linaro_uClibc-0.9.33.2/bin

- You **MAY** need to change **LIBNEW-GPIO_DIR** to relative directory of libnewgpio if you are not using the sources as originally supplied.

The default if using the standard **source** directory structure as supplied is:

LIBNEW-GPIO_DIR=../libnewgpio

- You **MAY** need to change **LIBNEW-I2C_DIR** to relative directory of libnewi2c if you are not using the sources as originally supplied.
  The default if using the standard **source** directory structure as supplied is:

  LIBNEW-GPIO_DIR=../libnewi2c

- You **MAY** need to change **LIBARDUINO_DIR** to relative directory of libarduino if you are not using the sources as originally supplied.
  The default if using the standard **source** directory structure as supplied is:

  LIBARDUINO_DIR=../libarduino

4.2. Makefile targets

The **Makefile** implements the following set of targets:

- **make**
  The default target. Performs a complete build of both static and dynamic link versions of the program.
  This is directly equivalent to:
  make static dynamic

- **make static**
  Performs a complete build of just the static link version of the program.

- **make dynamic**
  Performs a complete build of just the dynamic link version of the program.

- **make clean**
  Removes all previous build files, both static and dynamic link versions.
  This is directly equivalent to:
  make clean-static clean-dynamic

- **make clean-static**
  Removes all previous build files for static link versions only
  .
- **make clean-dynamic**
  Removes all previous build files for dynamic link versions only

If the following is added to the **make** command line:

builddep=1

then **libnewgpio**, l**ibnewi2c** and **libarduino** libraries that the program depends on will also be built before building the program.

5. Usage of the omegaio program

5.1. Brief Usage Information

Brief usage of the program can be obtained by running the command:

```
./omegaio -?
```

This produces the following output on **stderr**:

```
Control of GPIO, I2C devices and Arduino access
Brief Usage:
===========
./omegaio [any length sequence of space separated <input-element>s]
An <input-element> is one of:
       <option>
       <file-input>
       <operation>

Help is available by using one of:
       ./omegaio -?  - for basic help
       ./omegaio -u  - for brief usage
       ./omegaio -h  - for full help

Sources available at: https://github.com/KitBishop/Omega-GPIO-I2C-Arduino
```

The program is generally self-documenting via usage of variations on the **-h** option. Most of the rest of this document catalogues such self-documentation with additional comments where relevant.

5.2. General Processing

In general, when the program is run, there are two phases to the processing:

1. Parse all the supplied input elements reporting on any errors
2. If there are no error, then execute the actions specified by the input elements

Output from the program is written as follows:

- All output other than that controlled by the **-o** option (see later) is written to **stderr** – this includes any help output
- Output of the specific results of operations as controlled by the **-o** option (see later) is written to **stdout**

5.3. Input Elements

The input elements fall into 3 main types:

- **Options** – used to control specific aspects of the operation of the program in general
- **File Input** – used to provide input to the program from separate file(s) rather than just the command line

- **Operations** – specify specific operations to be performed with associated parameters where relevant

The sequence of input elements supplied to the program (either directly on the command line or indirectly via File Input) constitute a script that controls the operation of the program when the sequence of elements is executed.

A summary of the Input Elements and their general usage can be obtained by using the command:

```
./omegaio -u
```

This produces the following output on **stderr**:

```
A C++ program to control and interact with Omega GPIO pins,
    I2C devices, and Arduino access via scripted operations.
       Program version:         1.0.0
       GPIO Library version:    1.4.1
       I2C Library version:     1.0.0
       Arduino Library version: 1.0.0

Basic Usage:
===========
./omegaio [any length sequence of space separated <input-element>s]
An <input-element> is one of:
       <option>
       <file-input>
       <operation>
An <option> is one of:
       -v - verbose output
       -q - quiet output
       -o - result output
       -r - report output
       -i - ignore errors
       -e - error output
       -s - automatic setting of pin direction
       -x - hex output
       -d - output debugging information after parse complete
       -h - various levels of help output
       -u - brief usage help output
       -? - basic help output
A <file-input> is of the form:
       @<file-name> - input commands from file
An <operation> is of the form:
       <operation-name> <operation-parameters>
           An <operation-name> is one of:
             GPIO Operations::
              info          set           read          set-input
              set-output    get-direction pwm           pwmstop
              irq           irq2          irqstop             tone
              tonestop      shiftout      shiftin             pulseout
              pulsein             frequency     expled        expledstop
             I2C Operations::
              i2cprobe      i2cread8      i2cread16     i2cread32
              i2creadbuf    i2cwrite8     i2cwrite16    i2cwrite32
              i2cwritebuf
             Arduino Operations::
              arduinosys    asreboot      asretries     aspinmode
              asdigitalread asdigitalwrite      asanalogref   asanalogread
              asanalogwrite astone        asnotone      asshiftin
              asshiftout    aspulsein     arduinoport   apreboot
              apretries     apsendcmd     apsend8             apsend16
              apsend32      apsendbuf     apgetstatus   apget8
              apget16             apget32             apgetbuf
             Flow Control Operations::
              while         endwhile      if            else
              endif         continue      break         exit
             File Operations::
```

```
             filein       fileout             filedelete
           Miscellaneous Operations::
            delay        exec         assign
          The <operation-parameters> depend on the specific operation

More information can be displayed by using one of:
       -h or -h:all - for all help
       -h:-<option-letter> - for help on the option
       -h:@ - for help on file input
       -h:<operation-name> - for help on the operation and its parameters
       -h:expression - for help on expressions

Sources available at: https://github.com/KitBishop/Omega-GPIO-I2C-Arduino
```

5.4. Options

Option elements are primarily used to set various settings on the execution of the program. Options:

- may appear at any place in the input sequence
- they may be used multiple times
- with the exception of the **-d** option, they take affect at the point they appear within the input sequence

Help may be obtained on any valid option using a command like:

```
./omegaio -h:-<option-letter>
```

5.4.1. Option -v : verbose output

Help on the option is given by:

```
./omegaio -h:-v
```

Gives:

```
-v and -v+ - sets verbose mode; equivalent to -o -r -e
-v- - resets verbose mode; equivalent to -q
       Defaults are: -o+ -r- -e+
```

5.4.2. Option -q : quiet output

Help on the option is given by:

```
./omegaio -h:-q
```

Gives:

```
-q and -q+ - sets quiet mode; equivalent to -o- -r- -e-
-q- - resets quiet mode; equivalent to -v
```

5.4.3. Option -o : result output

Help on the option is given by:

```
./omegaio -h:-o
```

Gives:

```
-o and -o+ - enables output to stdout of any results of operation
-o- - disables output to stdout of any results of operation
       Default is: -o+
```

5.4.4. Option -r : report output

Help on the option is given by:

```
./omegaio -h:-r
```

Gives:

```
-r and -r+ - enables output to stderr of report on actions taken
-r- - disables output to stderr of report on actions taken
      Default is: -r-
```

5.4.5. Option -i : ignore errors

Help on the option is given by:

```
./omegaio -h:-i
```

Gives:

```
-i and -i+ - enables ignoring of any errors during processing
-i- - disables ignoring of any errors during processing
      Default is: -i-
```

5.4.6. Option -e : error output

Help on the option is given by:

```
./omegaio -h:-e
```

Gives:

```
-e and -e+ - enables output to stderr of any errors during processing
-e- - disables output to stderr of any errors during processing
      Default is: -e+
```

5.4.7. Option -s : automatic setting of pin direction

Help on the option is given by:

```
./omegaio -h:-s
```

Gives:

```
-s and -s+ - causes the program to ensure that the pin direction
      is set appropriately for each operation
-s- - does not set the direction for each operation
      Default is: -s-
```

NOTE: This option affects operations both on GPIO and on Arduino

5.4.8. Option -x : hex output

Help on the option is given by:

```
./omegaio -h:-x
```

Gives:

```
-x and -x+ - sets output to be displayed in hex
-x- - disables hex output, output is in decimal
      Default is: -x-
```

5.4.9. Option -d : output debugging information after parse complete

Help on the option is given by:

```
./omegaio -h:-d
```

Gives:

```
-d - enables debugging output on entered data
       When used anywhere in the input causes debugging output
       to be displayed on scanned input and processed operation data
       prior to any execution.    By default, debugging output is disabled
```

5.4.10. Option -h : various levels of help output

Help on the option is given by:

```
./omegaio -h:-h
```

Gives:

```
-h or -h:all - displays all available help
-h:-<option-letter> - displays help for the given option letter
-h:@ - displays help for input from file
-h:<operation> - displays help for the given operation
-h:expression - displays help for expressions
```

5.4.11. Option -u : brief usage help output

Help on the option is given by:

```
./omegaio -h:-u
```

Gives:

```
-u - displays brief usage help
```

5.4.12. Option -? : basic help output

Help on the option is given by:

```
./omegaio -h:-?
```

Gives:

```
-? - displays basic help
```

5.5. File Input

A File Input elements is used to insert the text of the named file in the input element list at the point at which it is used.

Help on the file input is given by:

```
./omegaio -h:@
```

Gives:

```
@<file-name> - causes input to be taken from the given file
       Input is free form sequence of space separated standard
              input elements: <option> <file-input> <operation>
       Any line with first non-blank character of # is ignored
       NOTES:
       1. Input is taken from the file at parse time prior to execution
       2. In general, use of @<file-name> can be nested but not
          recursively. I.E. the same file can NOT be referred to in
           another file either directly or indirectly.
       3. If <file-name> is '-', then uses standard input
       4. Input from standard input can only be used once and then only
          on the command line, NOT from within another file
       5. Input from standard input is terminated by Ctrl-D at the
          start of a new input line.
```

5.6. Operations

Operations are the main input-elements used to perform the required actions.

Many of the operations take additional parameters specific to the operation. These parameters immediately follow the operation named separated by spaces.

Many of the parameters to operations can be specified in the form of a general expression.

For convenience of reference only, the operations are grouped functionally by the type of operation. These groups are:

- **GPIO Operations** – Operations relating to access to GPIO pins
  - info
  - set
  - read
  - set-input
  - set-output
  - get-direction
  - pwm
  - pwmstop
  - irq
  - irq2
  - irqstop
  - tone
  - tonestop
  - shiftout
  - shiftin
  - pulseout
  - pulsein
  - frequency
  - expled
  - expledstop

- **I2C Operations** – Operations relating to access to I2C devices
  - i2cprobe
  - i2cread8
  - i2cread16
  - i2cread32
  - i2creadbuf
  - i2cwrite8
  - i2cwrite16
  - i2cwrite32
  - i2cwritebuf

- **Arduino Operations** – Operations relating to access to Arduino devices connected via I2C
  - arduinosys

    - asreboot
    - asretries
    - aspinmode
    - asdigitalread
    - asdigitalwrite
    - asanalogref
    - asanalogread
    - asanalogwrite
    - astone
    - asnotone
    - asshiftin
    - asshiftout
    - aspulsein
    - arduinoport
    - apreboot
    - apretries
    - apsendcmd
    - apsend8
    - apsend16
    - apsend32
    - apsendbuf
    - apgetstatus
    - apget8
    - apget16
    - apget32
    - apgetbuf

- **Flow Control Operations** – Operations relating to controlling the flow of execution of operations
    - while
    - endwhile
    - if
    - else
    - endif
    - continue
    - break
    - exit

- **File Operations** – Operations relating to access to external files
    - filein
    - fileout
    - filedelete

- **Miscellaneous Operations** – Other operations not otherwise grouped
    - delay

- exec
- assign

Help may be obtained on any valid operation using a command like:

```
./omegaio -h:<operation-name>
```

Help may be obtained on expressions using the command:

```
./omegaio -h:expression
```

5.6.1. Expressions in Operations

Many of the parameters to operations can be represented by a general expression.

Help may be obtained on expressions using the command:

```
./omegaio -h:expression
```

Which gives:

```
An expression is represented by an <expression_string>
       An expression can be used in a variety of places.
       The value of an expression is the result of evaluation
       of the expression string at run time.
       <expression-string> must be enclosed in " characters
       if it contains spaces or other special character.
       Formulated as a standard integer expression using:
       - Parentheses: ( and )
       - Unary Operators: + - ! ~
       - Binary Operators:
           Multiplication: * / % >> <<
           Addition: + - & | ^ && ||
           Comparison: > >= < <= == !=
       - Integer constant values, decimal or
         hex (starting with '0x' or '0X')
       - <varref> which will be replaced by actual value at run time.
         A <varref> is one of:
         - <variable-name> = current value of variable
           as assigned in an 'assign' operation
           or 0 if variable has never been assigned
         - $n or $nn = current value of pin n or nn
         - $! = value of latest result set by earlier operations
           The operations that set the result are:
             read          get-direction pulsein   shiftin   frequency
             filein        filedelete
             i2cprobe      i2cread8      i2cread16 i2cread32
             asdigitalread asanalogread  asshiftin aspulsein
             apgetstatus   apget8        apget16   apget32
             exec          assign
         - $? = status of last executed operation: 0=error, 1=ok
         - $[<file-name>] = file <file-name> exists; 0=no, 1=yes

       When an expression is used as a condition (as in the 'while'
       and 'if' operations), a zero value represents 'false' and a
       non-zero value represents 'true'.
```

5.6.2. GPIO Operations

GPIO Operations relate to access to Omega GPIO pins.

For more details, see **libnewgpio** library that is available and documented at
https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libnewgpio

5.6.2.1. info Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:info
```

Which gives:

```
info <pin-expr>
       Displays information on given or all pins
       <pin-expr> is either the value: all
       or an expression that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.2. set Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:set
```

Which gives:

```
set <pin-expr> <value-expr>
       Sets given pin(s) to the given value
       <pin-expr> is either the value: all
       or an expression that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <value-expr> is an expression that evaluates to the
       value to output. A value of zero outputs a 0,
       non-zero outputs a 1
```

5.6.2.3. read Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:read
```

Which gives:

```
read <pin-expr>
       Reads, displays and returns value of given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.4. set-input Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:set-input
```

Which gives:

```
set-input <pin-expr>
       Sets given pin or all pins to be input pins
       <pin-expr> is either the value: all
       or an expression that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.5. set-output Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:set-output
```

Which gives:

```
set-output <pin-expr>
       Sets given or all pins to be output pins
       <pin-expr> is either the value: all
```

```
       or an expression that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.6. get-direction Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:get-direction
```

Which gives:

```
get-direction <pin-expr>
       Reads, displays and returns the current direction
       of given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.7. pwm Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:pwm
```

Which gives:

```
pwm <pin-expr> <freq-expr> <duty-expr> <duration-expr>
       Starts a separate process to perform PWM output
       on given pin with given information
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <freq-expr> is an expression that evaluates to the
       frequency of the PWM pulses. Must be greater than 0
       <duty-expr> is an expression that evaluates to the
       duty cycle % of the PWM pulses. Must be >= 0 and <= 100
       <duration-expr> is an expression that evaluates to the
       duration of the output in milliseconds
       Must be >= 0 When 0, duration is indefinite
       The separate process runs until the duration expires
       or the 'pwmstop' operation is performed on the same <pin-number>
```

5.6.2.8. pwmstop Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:pwmstop
```

Which gives:

```
pwmstop <pin-expr>
       Stops any separate process that is currently performing PWM
       output on given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.9. irq Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:irq
```

Which gives:

```
irq <pin-expr> <irq-type> <command> <debounce-expr>
       Starts a separate process to respond to interrupts
       on given pin and perform given <command> on interrupt
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <irq-type> is the type of interrupt to catch
       Must be one of:
```

```
              rising - interrupt occurs on rising edge
              falling - interrupt occurs on falling edge
              both - interrupt occurs on both rising and falling edges
       <command> is the command to be performed on interrupt
       Must be enclosed in " characters if it contains spaces
       or other special character
       The <command> may contain any number of <varsub>s which will be
       replaced by actual values at the time the irq operation is
       actually invoked.
       A <varsub> is any sequence like {<varref>}
       where '<varref>' is one of:
         - <variable-name> = current value of variable
           as assigned in an 'assign' operation
           or not substituted if variable has never been assigned
         - $n or $nn = current value of pin n or nn
         - $! = value of latest result set by earlier commands
         - $? = status of last executed command: 0=error, 1=ok
         - $[<file-name>] = file <file-name> exists; 0=no, 1=yes
       <debounce-expr> is an expression that evaluates to a
       debounce time in milliseconds.  Any interrupts that occur within
       this time of a previous interrupt will be ignored.
       Used to cater for noisy mechanical signals
       Must be greater than or equal to 0
       When 0, no debounce testing is applied
       The separate process runs until the 'irqstop' operation
       is performed on the same <pin-number>
```

5.6.2.10. irq2 Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:irq2
```

Which gives:

```
irq2 <pin-expr> <rising-command> <falling-command> <debounce-expr>
       Starts a separate process to respond to interrupts
       on given pin and perform given commands on interrupt
       both for rising and falling edges
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <rising-command> is the command to be performed on interrupt
              rising edge.
       <falling-command> is the command to be performed on interrupt
              falling edge.
       Each must be enclosed in " characters if it contains spaces
       or other special character
       Each command may contain any number of <varsub>s which will be
       replaced by actual values at the time the irq2 operation is
       actually invoked.
       A <varsub> is any sequence like {<varref>}
       where '<varref>' is one of:
         - <variable-name> = current value of variable
           as assigned in an 'assign' operation
           or not substituted if variable has never been assigned
         - $n or $nn = current value of pin n or nn
         - $! = value of latest result set by earlier commands
         - $? = status of last executed command: 0=error, 1=ok
         - $[<file-name>] = file <file-name> exists; 0=no, 1=yes
       <debounce-expr> is an expression that evaluates to a
       debounce time in milliseconds.  Any interrupts that occur within
       this time of a previous interrupt will be ignored.
       Used to cater for noisy mechanical signals
       Must be greater than or equal to 0
       When 0, no debounce testing is applied
       The separate process runs until the 'irqstop' operation
       is performed on the same <pin-number>
```

5.6.2.11. irqstop Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:irqstop
```

Which gives:

```
irqstop <pin-expr>
       Stops any separate process that is currently performing IRQ
       handling on given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.12. tone Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:tone
```

Which gives:

```
tone <pin-expr> <freq-expr> <duration-expr>
       Starts a separate process to output a continuous
       tone to given pin with given information
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <freq-expr> is an expression that evaluates to the
       frequency of the tone. Must be greater than 0
       <duration-expr> is an expression that evaluates to the
       duration of the output in milliseconds
       Must be >= 0 When 0, duration is indefinite
       The separate process runs until the duration expires
       or the 'tonestop' operation is performed on the same pin
       NOTE: equivalent to doing a 'pwm' operation with a <duty> of 50%
```

5.6.2.13. tonestop Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:tonestop
```

Which gives:

```
tonestop <pin-expr>
       Stops any separate process that is currently performing tone
       output on given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
```

5.6.2.14. shiftout Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:shiftout
```

Which gives:

```
shiftout <datapin-expr> <clockpin-expr> <clockperiod-expr>
              <bitord> <value-expr>
       Performs a serial output of a value on a data pin
       clocked by a clock pin
       <datapin-expr> specifies the data pin number
       <clockpin-expr> specifies the clock pin number
       They must be expressions that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       and evaluate to different pin numbers
       <clockperiod-expr> is an expression that evaluates to
       the period of the clock in nano-seconds
```

```
       Must not be less than 100
       <bitord> is the order the bits are transferred
       Must be one of:
              msb - most significant bit first
              lsb - least significant bit first
       <value-expr> is an expression that evaluates to
       the value sent
       Must be >= 0 and <= 255
```

5.6.2.15. shiftin Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:shiftin
```

Which gives:

```
shiftin <datapin-expr> <clockpin-expr> <clockperiod-expr> <bitorder>
       Performs a serial input of a value from a data pin
       clocked by a clock pin
       <datapin-expr> specifies the data pin number
       <clockpin-expr> specifies the clock pin number
       They must be expressions that evaluates to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       and evaluate to different pin numbers
       <clockperiod-expr> is an expression that evaluates to
       the period of the clock in nano-seconds
       Must not be less than 100
       <bitorder> is the order the bits are transferred
       Must be one of:
              msb - most significant bit first
              lsb - least significant bit first
```

5.6.2.16. pulseout Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:pulseout
```

Which gives:

```
pulseout <pin-expr> <duration-expr> <level-expr>
       Outputs a single pulse to given pin with supplied data
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <duration-expr> is an expression that evaluates to the
       length of pulse in micro-seconds. Must be >0
       <level-expr> is an expression that evaluates to the pulse level.
       A nonzero value represents a level of 1
```

5.6.2.17. pulsein Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:pulsein
```

Which gives:

```
pulsein <pin-expr> <level-expr> <timeout-expr>
       Waits for and displays and returns duration of pulse input on
       given pin
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <level-expr> is an expression that evaluates to the pulse level.
       A nonzero value represents a level of 1
       <timeout-expr> is an expression that evaluates to the time to
       wait for the pulse and its completion in micro-seconds.
       Must be >=0  If 0, timeout is indefinite
```

5.6.2.18. frequency Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:frequency
```

Which gives:

```
frequency <pin-expr> <sampletime-expr>
       Obtains, displays and returns frequency of input on
       given pin
       Operates by counting the number of pulses during the sample time
       <pin-expr> is an expression that must evaluate to one of:
              0,1,6,7,8,12,13,14,15,16,17,18,19,23,26
       <sampletime-expr> is an expression that evaluates to the time
       in milliseconds over which the frequency is taken. Must be >0
```

5.6.2.19. expled Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:expled
```

Which gives:

```
expled <value-expr>
Or:
expled rgb <red-expr> <green-expr> <blue-expr>
       Starts a separate process to output given colour to
       expansion dock LED.
       <value-expr> is an expression that evaluates to a value
       that specifies all the LED components as follows:
         red = (value >> 16) & 0xFF
         green = (value >> 8) & 0xFF
         blue = value & 0xFF
       where each is in the range '0x00' (off) to '0xFF' (fully on)
       <red-expr> <green-expr> <blue-expr> are expressions that
       specify the LED colours individually
       Each must be in the range 0 (off) to 100 (fully on)
       The separate process runs until the 'expledstop' operation
       is performed
       NOTE: expled uses pins:15 (blue), 16 (green), 17 (red)
```

5.6.2.20. expledstop Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:expledstop
```

Which gives:

```
expledstop
       Stops any separate process that is currently performing
       expled output
```

5.6.3. I2C Operations

I2C Operations relate to access to I2C devices connected to the Omega.

For more details, see **libnewi2c** library that is available and documented at
https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libnewi2c

5.6.3.1. i2cprobe Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:i2cprobe
```

Which gives:

```
i2cprobe <i2caddr-expr>
       Probes an I2C address and displays and returns result
       <i2caddr-expr> is an expression that evaluates to the
       I2C address.  Must be >= 0x03 and <= Ox7f
```

5.6.3.2. i2cread8, i2cread16 and i2cread32 Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:i2cread8
```

or:

```
./omegaio -h:i2cread16
```

or:

```
./omegaio -h:i2cread32
```

Which give:

```
i2cread8 <i2caddr-expr> <regaddr-expr>
i2cread16 <i2caddr-expr> <regaddr-expr>
i2cread32 <i2caddr-expr> <regaddr-expr>
       Reads, displays and returns an 8 bit or 16 bit or 32 bit
       value from I2C
       <i2caddr-expr> is an expression that evaluates to the
       I2C address.  Must be >= 0x03 and <= Ox7f
       <regaddr-expr> is an expression that evaluates to the
       I2C register address to be read from or
       'none' for no register address.
       If not 'none', must be >= 0 and <= Oxff
```

5.6.3.3. i2creadbuf Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:i2creadbuf
```

Which gives:

```
i2creadbuf <i2caddr-expr> <regaddr-expr> <maxbytes-expr>
       Reads and displays a buffer from I2C
       <i2caddr-expr> is an expression that evaluates to the
       I2C address.  Must be >= 0x03 and <= Ox7f
       <regaddr-expr> is an expression that evaluates to the
       I2C register address to be read from or
       'none' for no register address.
       If not 'none', must be >= 0 and <= Oxff
       <maxbytes-expr> is an expression that evaluates to the
       maximum number of bytes to be read. Must be > 0 and <= 32
```

5.6.3.4. i2cwrite8, i2cwrite16 and i2cwrite32 Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:i2cwrite8
```

or:

```
./omegaio -h:i2cwrite16
```

or:

```
./omegaio -h:i2cwrite32
```

Which give:

```
i2cwrite8 <i2caddr-expr> <regaddr-expr> <value-expr>
i2cwrite16 <i2caddr-expr> <regaddr-expr> <value-expr>
i2cwrite32 <i2caddr-expr> <regaddr-expr> <value-expr>
       Writes an 8 bit or 16 bit or 32 bit value to I2C
       <i2caddr-expr> is an expression that evaluates to the
       I2C address.  Must be >= 0x03 and <= Ox7f
       <regaddr-expr> is an expression that evaluates to the
       I2C register address to be written to or
       'none' for no register address.
       If not 'none', must be >= 0 and <= Oxff
       <value-expr> is an expression that evaluates to the
       value to be written.
```

5.6.3.5. *i2cwritebuf Operation*

Help on this operation may be obtained using the command:

```
./omegaio -h:i2cwritebuf
```

Which gives:

```
i2writebuf <i2caddr-expr> <regaddr-expr> <bufbytes-list>
       Writes a buffer of bytes to I2C
       <i2caddr-expr> is an expression that evaluates to the
       I2C address.  Must be >= 0x03 and <= Ox7f
       <regaddr-expr> is an expression that evaluates to the
       I2C register address to be written to or
       'none' for no register address.
       If not 'none', must be >= 0 and <= Oxff
       <bufbytes-list> is a comma separated list of expressions each
       representing the value of one byte of the buffer.  It must be
       enclosed in quotes if it contains spaces or other special
       characters.  The list is of the form:
          <val_expr>;<val_expr>;....
       where there may be from at least 1 to at most 32 <val_expr>s
       Each <val_expr> is an expression that evaluates to the
       value of one byte.  The least significant 8 bits of each
       value are used for the byte.
```

5.6.4. Arduino Operations

Arduino Operations relate to access to Arduino devices connected to the Omega via I2C.

For more details, see **libarduino** library that is available and documented at
https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/libarduino

Additionally, the Arduino library **arduino-omega** for use on the Arduino is available and documented at
https://github.com/KitBishop/Omega-GPIO-I2C-Arduino/tree/master/arduino_omega

The Arduino Operations effectively make up two sub-groups:

- **Arduino System Operations**
  These use an **ArduinoSystem** object from **libarduino** on the Omega side and an **OmegaArduinoSystemPort** object from **arduino_omega** on the Arduino side to provide access to Arduino I/O operations

  The operations involved are:
  - **arduinosys** – used to create the current instance of **ArduinoSystem** used
  - All operations with name like **asxxxxx** that are used to operate on the current instance of **ArduinoSystem**

- **Arduino Port Operations**
  These use an **ArduinoPort** object from **libarduino** on the Omega side and an object that is a descendent of **OmegaPort** from **arduino_omega** on the Arduino side to provide access to any such functionality provided by the **OmegaPort** object on the Arduino

  The operations involved are:
  - **arduinoport** – used to create the current instance of **ArduinoPort** used
  - All operations with name like **arxxxxx** that are used to operate on the current instance of **ArduinoPort**

5.6.4.1. Arduino System Operations

5.6.4.1.1. arduinosys Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:arduinosys
```

Which gives:

```
arduinosys <i2caddr-expr> <port-expr>
       Sets the Arduino System for usage by subsequent 'as*' operations
       By default, if no Arduino System has been set the default system
       on I2C address 0x08 and port 0 is used
       <i2caddr-expr> is an expression that evaluates to the
       I2C address to use for the Arduino System.
       Must be >= 0x03 and <= Ox7f
       <port-expr> is an expression that evaluates to the
       Arduino port number to be used for Arduino System access
       Must be >= 0 and <= 15
       NOTE: Usage of Arduino System access requires that an
       Arduino be attached to the Omega for access with I2C
       (e.g. Arduino Dock) and that the Arduino be running
       an instance of OmegaArduinoSystemPort from the Arduino
       library 'Arduino-Omega' with the corresponding I2C address
       and port number.
```

5.6.4.1.2. asreboot Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asreboot
```

Which gives:

```
asreboot
       Reboots the current Arduino System
```

5.6.4.1.3. asretries Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asretries
```

Which gives:

```
asretries <delay-expr> <count-expr>
       Sets the retry information for the current Arduino System
       <delay-expr> is an expression that evaluates to
       the delay in milliseconds between retries on error.
       Must be >= 0
       <count-expr> is an expression that evaluates to
       the maximum number of retries to perform on error.
       Less than 0 represents unlimited retries.
```

5.6.4.1.4. aspinmode Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:aspinmode
```

Which gives:

```
aspinmode <arduinopin-expr> <pinmode>
       Sets given Arduino System pin mode to the given value
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       <pinmode> is the mode to use for the pin and must be one of:
         - input
         - input_pullup
         - output
```

5.6.4.1.5. asdigitalread Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asdigitalread
```

Which gives:

```
asdigitalread <arduinopin-expr>
       Reads, displays and returns value of given Arduino System pin
       On the Arduino, uses the function 'digitalRead(pin)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
```

5.6.4.1.6. asdigitalwrite Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asdigitalwrite
```

Which gives:

```
asdigitalwrite <arduinopin-expr> <val-expr>
       Sets given Arduino System pin to the given value
       On the Arduino, uses the function 'digitalWrite(pin, value)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       <val-expr> is an expression that evaluates to give the
       value to set the pin to:
         == 0 - sets pin to LOW
         != 0 - sets pin to HIGH
```

5.6.4.1.7. asanalogref Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asanalogref
```

Which gives:

```
asanalogref <mode>
       Sets Arduino System analog reference mode to the given value
       <mode> is the mode to use for the analog reference and must be
       one of:
         - default
         - internal
         - external
```

5.6.4.1.8. asanalogread Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asanalogread
```

Which gives:

```
asanalogread <arduinopin-expr>
```

```
       Reads, displays and returns analog value of given Arduino System
       pin
       On the Arduino, uses the function 'analogRead(pin)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       NOTE: Arduino normally refers to analog pins as A0 to A5 which
       correspond to actual pin numbers 14 to 19.
       For convenience, this operation will convert pin numbers 0 to 5
       to use 14 to 19
       Any pin number other than 0 to 5 or 14 to 19 is invalid.
```

5.6.4.1.9. asanalogwrite Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asanalogwrite
```

Which gives:

```
asanalogwrite <arduinopin-expr> <val-expr>
       Writes given analog value to Arduino System pin
       On the Arduino, uses the function 'analogWrite(pin, value)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       NOTE: The Arduino can only perform analog write functionality
       on pins 3, 5, 6, 9, 10, and 11 any other pin values are invalid.
       <val-expr> is an expression that evaluates to give the
       value to set the pin to.  Only the least significant 8 bits
       of the value are used (i.e. 0 to 255).
       NOTE: The Arduino does NOT perform true analog output.
       Rather, it produces PWM output on the pin used with a value of
       0 producing 0% duty cycle and 255 producing 100% duty cycle.
```

5.6.4.1.10. astone Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:astone
```

Which gives:

```
astone <arduinopin-expr> <freq-expr> <duration-expr>
       Starts tone output to Arduino System pin
       On the Arduino, uses the function
       'tone(pin, frequency, duration)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       <freq-expr> is an expression that evaluates to give the
       frequency of the tone.  Must be > 0
       <duration-expr> is an expression that evaluates to
       the duration in milliseconds for which the tone is output.
       Must be >= 0  If 0, tone is continuous.
       NOTE: The tone continues until the duration expires or
       another 'astone' operation is used or the 'asnotone' operation
       is used on the same pin.
```

5.6.4.1.11. asnotone Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asnotone
```

Which gives:

```
asnotone <arduinopin-expr>
       Stops tone output on Arduino System pin
       On the Arduino, uses the function 'noTone(pin)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
```

5.6.4.1.12. asshiftin Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asshiftin
```

Which gives:

```
asshiftin <arduinodatapin-expr> <arduinoclockpin-expr> <bitorder>
       Reads, displays and returns result of performing shift in
       operation using given pins on Arduino System
       On the Arduino, uses the function
       'shiftIn(dataPin, clockPin, bitOrder)'
       <arduinodatapin-expr> is an expression that evaluates to
       the data pin number to use.
       <arduinoclockpin-expr> is an expression that evaluates to
       the clock pin number to use.
       <bitorder> specifies the order in which the data bits are
       transferred.  Must be one of:
         - lsb = least significant bit first
         - msb = most significant bit first
```

5.6.4.1.13. asshiftout Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:asshiftout
```

Which gives:

```
asshiftout <arduinodatapin-expr> <arduinoclockpin-expr> <bitorder>
           <value-expr>
       Performs shift out operation using given pins on Arduino System
       On the Arduino, uses the function
       'shiftOut(dataPin, clockPin, bitOrder, value)'
       <arduinodatapin-expr> is an expression that evaluates to
       the data pin number to use.
       <arduinoclockpin-expr> is an expression that evaluates to
       the clock pin number to use.
       <bitorder> specifies the order in which the data bits are
       transferred.  Must be one of:
         - lsb = least significant bit first
         - msb = most significant bit first
       <value-expr> is an expression that evaluates to
       the value to be sent.  Only the least significant 8 bits
       of the value are used (i.e. 0 to 255).
```

5.6.4.1.14. aspulsein Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:aspulsein
```

Which gives:

```
aspulsein <arduinopin-expr> <level-expr> <timeout-expr>
       Performs a pulse in operation on given Arduino System pin
       and displays and returns the value in microseconds
       On the Arduino, uses the function 'pulseIn(pin, level, timeout)'
       <arduinopin-expr> is an expression that evaluates to
       pin number to use.
       <level-expr> is an expression that evaluates to give the
       value of the pulse level to input:
         == 0 - for a LOW level pulse
         != 0 - for a HIGH level pulse
       <timeout-expr> is an expression that evaluates to
       the time in microseconds to wait for the pulse.
       Must be >= 0
```

5.6.4.2. Arduino Port Operations

5.6.4.2.1. arduinoport Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:arduinoport
```

Which gives:

```
arduinoport <i2caddr-expr> <port-expr>
       Sets the Arduino Port for usage by subsequent 'ap*' operations
       <i2caddr-expr> is an expression that evaluates to the
       I2C address to use for the Arduino Port.
       Must be >= 0x03 and <= 0x7f
       <port-expr> is an expression that evaluates to the
       Arduino port number to be used for Arduino Port access
       Must be >= 0 and <= 15
       NOTE: Usage of Arduino Port access requires that an Arduino
       be attached to the Omega for access with I2C
       (e.g. Arduino Dock) and that the Arduino be running
       an instance of OmegaPort from the Arduino library
       'Arduino-Omega' with the corresponding I2C address and port
       number and with the method 'processCommand' implemented.
```

5.6.4.2.2. apreboot Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:apreboot
```

Which gives:

```
apreboot
       Reboots the current Arduino Port
```

5.6.4.2.3. apretries Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:apretries
```

Which gives:

```
apretries <delay-expr> <count-expr>
       Sets the retry information for the current Arduino Port
       <delay-expr> is an expression that evaluates to
       the delay in milliseconds between retries on error.
       Must be >= 0
       <count-expr> is an expression that evaluates to
       the maximum number of retries to perform on error.
       Less than 0 represents unlimited retries.
```

5.6.4.2.4. apsendcmd, apsend8, apsend16 and apsend32 Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:apsendcmd
```

or:

```
./omegaio -h:apsend8
```

or:

```
./omegaio -h:apsend16
```

or:

```
./omegaio -h:apsend32
```

Which give:

```
apsendcmd <command-expr>
apsend8 <command-expr> <value-expr>
apsend16 <command-expr> <value-expr>
apsend32 <command-expr> <value-expr>
       Sends a command to the current Arduino Port with either
       no data or an 8 bit or 16 bit or 32 bit data value
       <command-expr> is an expression that evaluates to the
       command to send.  Only the least significant 8 bits are used.
```

```
       <value-expr> is an expression that evaluates to the
       value to be sent.
       NOTE: Depending upon the implementation of the Port on the
       Arduino, use of one of these operations will normally be
       followed by one of the operations 'apgetstatus', 'apget8',
       'apget16', 'apget32', 'apgetbuf' to retrieve the result of
       the operation.
```

5.6.4.2.5. apsendbuf Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:apsendbuf
```

Which gives:

```
apsendbuf <command-expr> <bufbytes-list>
       Sends a buffer of bytesto the current Arduino Port
       <command-expr> is an expression that evaluates to the
       command to send.  Only the least significant 8 bits are used.
       <bufbytes-list> is a comma separated list of expressions each
       representing the value of one byte of the buffer.  It must be
       enclosed in quotes if it contains spaces or other special
       characters.  The list is of the form:
          <val_expr>;<val_expr>;....
       where there may be from at least 1 to at most 32 <val_expr>s
       Each <val_expr> is an expression that evaluates to the
       value of one byte.  The least significant 8 bits of each
       value are used for the byte.
       NOTE: Depending upon the implementation of the Port on the
       Arduino, use of this operation will normally be followed by
       one of the operations 'apgetstatus', 'apget8','apget16',
       'apget32', 'apgetbuf' to retrieve the result of the operation.
```

5.6.4.2.6. apgetstatus, apget8, apget16 and apget32 Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:apgetstatus
```

or:

```
./omegaio -h:apget8
```

or:

```
./omegaio -h:apget16
```

or:

```
./omegaio -h:apget32
```

Which give:

```
apgetstatus
apget8
apget16
apget32
       Gets, displays and returns a status or an 8 bit or 16 bit or
       32 bit value from the current Arduino Port
       NOTE: Depending upon the implementation of the Port on the
       Arduino, use of one of these operations will normally be
       called after performing one of the operations 'apsendcmd',
       'apsend8', 'apsend16', 'apsend32', 'apsendbuf' to retrieve the
       result of that operation.
```

5.6.4.2.7. apgetbuf Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:apgetbuf
```

Which gives:

```
apgetbuf <maxbytes-expr>
       Reads and displays a buffer from the current Arduino Port
       <maxbytes-expr> is an expression that evaluates to the
       maximum number of bytes to be read. Must be > 0 and <= 32
       NOTE: Depending upon the implementation of the Port on the
       Arduino, use of this operation may need to be called after
       performing one of the operations 'apsendcmd', 'apsend8',
       'apsend16', 'apsend32', 'apsendbuf' to retrieve the result of
       that operation.
```

5.6.5. Flow Control Operations

Flow Control Operations relate to controlling the sequence of execution of operations.

5.6.5.1. while and endwhile Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:while
```

or:

```
./omegaio -h:endwhile
```

Which give:

```
while <conditional-expression>
endwhile
       Repeatedly execute all operations between 'while' and 'endwhile'
       so long as <conditional-expression> is true
       <conditional-expression> is true if the expression evaluates
       to non-zero.
       Use the help parameter -h:expression for help on expressions.
```

5.6.5.2. if, else and endif Operations

Help on these operations may be obtained using the commands:

```
./omegaio -h:if
```

or:

```
./omegaio -h:else
```

or:

```
./omegaio -h:endif
```

Which give:

```
if <conditional-expression>
endif
Or:
if <conditional-expression>
else
endif
       Executes all operations between 'if' and 'else' or
       'endif' if the <conditional-expression> is true
       If 'else' is used, executes all operations between 'else' and
       'endif' if the <conditional-expression> is false
       <conditional-expression> is true if the expression evaluates
       to non-zero.
       Use the help parameter -h:expression for help on expressions.
```

5.6.5.3. continue Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:continue
```

Which gives:

```
continue
       If within a 'while' loop, immediately continues next execution
       of the loop body skipping execution of rest of body.
       If not within a 'while' loop, ends all execution
```

5.6.5.4. break Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:break
```

Which gives:

```
break
       If within a 'while' loop, breaks execution of the loop
       If not within a 'while' loop, ends all execution
```

5.6.5.5. exit Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:exit
```

Which gives:

```
exit <result>
       Exits execution with given result
       <result> is the value to return on exit
```

5.6.6. File Operations

File Operations relate to access to external files

5.6.6.1. filein Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:filein
```

Which gives:

```
filein <file-name> <variable-name>
       Reads integer value from first line of file to variable
       and saves value read.
       Along with 'fileout', and 'filedelete' operations and
       $[<filename>] expression item, provides a rudimentary
       mechanism for communicating with other programs.
       <file-name> is the name of the file to read from.
       Must be enclosed in " characters if it contains spaces
       or other special character.
       <variable-name> is the user variable to which to assign
       the value read.
       Any sequence of letters and digits starting with a letter.
       Names are case sensitive
```

5.6.6.2. fileout Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:fileout
```

Which gives:

```
fileout <file-name> <expression>
       Writes the value of the expression as the first line
```

```
       of given file.
       Along with 'filein', and 'filedelete' operations and
       $[<filename>] expression item, provides a rudimentary
       mechanism for communicating with other programs.
       <file-name> is the name of the file to write to.
       Must be enclosed in " characters if it contains spaces
       or other special character.
       <expression> is the expression value to write.
       Use the help parameter -h:expression for help on expressions.
```

5.6.6.3. *filedelete Operation*

Help on this operation may be obtained using the command:

```
./omegaio -h:filedelete
```

Which gives:

```
filedelete <file-name>
       Deletes the file with given name.
       Along with 'filein', and 'fileout' operations and
       $[<filename>] expression item, provides a rudimentary
       mechanism for communicating with other programs.
       <file-name> is the name of the file to delete.
       Must be enclosed in " characters if it contains spaces
       or other special character.
```

5.6.7. Miscellaneous Operations

Miscellaneous Operations are operations that are not otherwise categorised.

5.6.7.1. *delay Operation*

Help on this operation may be obtained using the command:

```
./omegaio -h:delay
```

Which gives:

```
delay <delay-expr>
       Pauses execution for the given period
       <delay-expr> is an expression that evaluates to
       the delay period in milliseconds. Must be >=0
```

5.6.7.2. *exec Operation*

Help on this operation may be obtained using the command:

```
./omegaio -h:exec
```

Which gives:

```
exec <command>
       Executes the given command
       <command> is the command to execute.
       Must be enclosed in " characters if it contains spaces
       or other special character
       The <command> may contain any number of <varsub>s which will be
       replaced by actual values at run time.
       A <varsub> is any sequence like {<varref>}
       where '<varref>' is one of:
         - <variable-name> = current value of variable
           as assigned in an 'assign' operation
           or not substituted if variable has never been assigned
         - $n or $nn = current value of pin n or nn
         - $! = value of latest result set by earlier commands
         - $? = status of last executed command: 0=error, 1=ok
         - $[<file-name>] = file <file-name> exists; 0=no, 1=yes
```

5.6.7.3. assign Operation

Help on this operation may be obtained using the command:

```
./omegaio -h:assign
```

Which gives:

```
assign <variable-name> <expression>
       Assigns the given expression to the given variable
       <variable-name> is the user variable to assign to.
       Any sequence of letters and digits starting with a letter.
       Names are case sensitive
       <expression> is the expression value to assign.
       Use the help parameter -h:expression for help on expressions.
```

5.7. omegaio-example.txt – example script file

The file **omegaio-example.txt** is an example script file that can be used as input to the **omegaio** program that demonstrates some of the scripting capabilities.

The file contains extensive comments on its usage.

The script file is best run under **omegaio** as a background task using the command:

```
./omegaio @omegaio-example.txt &
```

Note that use of this script requires action by you. The script outputs prompts for the actions to be performed. Once running, the script cycles the expansion LED through red, green and blue repeatedly once a second until stopped according to the displayed instructions.

For reference, the contents of this file is:

```
# Ensure that pin directions set appropriately by using the -s option
-s

# Use the -q option to suppress extraneous output
-q

# Set 'conut' variable to 0
assign count 0

# Ensure expled is turned off
expled rgb 0 0 0

# Initial instruction to start main processing loop
exec "echo"
exec "echo"
exec "echo Ensure pin 0 is connected to a high to start main processing"

# Wait until pin 0 is high
while !$0
endwhile

# Clear file stop.txt to start with 0
fileout stop.txt 0

# Provide feedback on how to stop main processing loop
exec "echo"
exec "echo Main processing started by pin 0 being connected to a high"
exec "echo It will continue until pin 0 is low or the file \'stop.txt\'"
exec "echo ' starts with a line containing 99'"
exec "echo"
```

```
# Main processing loop runs so long as pin 0 is high
while $0

# Set expled to red when count is a multiple of 3
    if "(count % 3) == 0"
       expled rgb 100 0 0
    endif

# Set expled to green when count is a (multiple of 3) + 1
    if "(count % 3) == 1"
       expled rgb 0 100 0
    endif

# Set expled to blue when count is a (multiple of 3) + 2
    if "(count % 3) == 2"
       expled rgb 0 0 100
    endif

# Delay for 1 second
    delay 1000

# Display current count when count is a multiple of 10
    if "((count % 10) == 0) && (count != 0)"
       exec "echo count is now {count}"
    endif

# Increment count
    assign count "count+1"

# Check that file stop.txt exists
# Actually redundant since if it doesn't reading from it will return 0 anyway
    if $[stop.txt]

# Read first line from file stop.txt to variable stopval
       filein "stop.txt" stopval

# Test if stopval read is 99 and break main loop
       if "stopval == 99"
           break
       endif
    endif

# End of while loop
endwhile

# Report on reason that main loop ended
if "stopval == 99"
    exec "echo"
    exec "echo Processing ended by existence of \'stop.txt\' starting with line
with value 99"
else
    exec "echo"
    exec "echo Processing ended by pin 0 going low"
endif

# Turn expled off
expled rgb 0 0 0

# Stop process responsible for running expled
expledstop
```

6. Further Development

Development of **omegaio** is on-going. There will be changes and additions to the code in the future.