AmigaFlight

Andrew Duffy Morris

COLLABORATORS

	TITLE : AmigaFlight		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Andrew Duffy Morris	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AmigaFlight

1.1 AmigaFlight® Help

Documentation for

AmigaFlight
Version 1.2

A freely distributable 68000 learning utility that provides many pre-programmed subroutines to eliminate the need for learning the internal workings of the Amiga® Operating System.

Written by XCNT :

MIDI   - Andrew Duffy
EXEC   - Alan James
NEMESIS - Chris Morris
Contacting XCNT

1.2 AmigaFlight® Help: Latest additions

Latest Additions
================

Thankyou for aquiring XCNT AmigaFlight version 1.2.

This is the latest version of AmigaFlight, and although it was completed in June 1994 (a mere 16 months ago), nothing was ever done about distributing it (except a few failed attempts at FTP'ing it to AmiNet).

Finally, after a year in industry, we are back and are distributing it now for real.

Dated : 20th November 1995.

Note : See section Usage for assembling your own files !

1.3 AmigaFlight® Help: Introduction to XCNT AmigaFlight

Introduction
============

During our first year of study at Staffordshire University, England we were introduced~ to the 68000 machine~language. Using an IBM-PC, we were expected to write programs (in a not too sophisticated text editor), cross assemble our code, and then download~the assembled programs~ to a 68000 circuit board.

It became apparent that a lot of students were having some difficulties in writing, assembling,~downloading and then running the programs that they had written. It was also evident that a lot of students had Amiga® computers ~at~home, and as the Amiga® range is based upon the Motorola 680x0 micro-processor, some students~tried to~assemble their programs on their computers - without much success.

The main reason why these programs won't work is that the EPROM routines that the Flight Board uses (such as INCH, OUTCH, INSTHEX) do not exist in the same form on the ~Amiga. ~There ~are ~routines on~ the Amiga that are similar, but~they~require a good working knowledge of~the operating system to be able to use them.

This is where AmigaFlight® ~comes in. AmigaFlight® is ~designed to ~be~a full, or as full as we can make it, emulation of the Flight Boards used by Staffordshire University. AmigaFlight® has, we hope, an easy to use and pleasant graphical ~interface ~that will make the task of writing these programs easier, and more convenient.

The original idea was to write the program specifically for Staffordshire University students, and sell licensed copies to anyone who was interested, however, it became apparent that this program could be of use to anybody who had a interest in learning or developing 680*0 programs.

1.4 AmigaFlight® Help: Backgroud Information

Background Information
======================

Version 1.0

XCNT AmigaFlight® started out life as a very small include file used only by the programmers.
It was when ~using this that it became apparent that other people would also like to ~use ~our program ~as ~well, so, after a little more work, a small prototype version was produced in the form of an object file that could be linked with the users code.

Version 1.1

After many more months of development, and research into the routines that curently exist ~on ~the FlightBoard, ~together ~with ~development of the Applications Board emulation, a version was completed that was acceptable for public release.
The xasm pre-processor was created in HiSoft BASIC.

Version 1.2

During ~the ~summer ~of ~1993, ~MIDI ~spent ~even ~more ~time ~developing AmigaFlight® with ~a ~major ~rewrite to ~make ~AmigaFlight a 'stand-alone' product that did not require linking with each and every single program, thus dramatically reducing the disk space required to comfortably use and develop with AmigaFlight®.
Over the months of autumn 1993 to spring 1994 AmigaFlight® was refined to provide a more configurable and far friendlier user interface.
An Installer script was also added.

Changes to the AmigaFlight® executable :

- The users assembled code is now simply loaded into AmigaFlight® and linking with the routines is done internally.

- An AppIcon and an AppMenu are now available to the user, and object files can be loaded using these two facilities.

- AmigaFlight® can now be loaded as a project to an icon, with the calling ~icon being ~initially loaded ~and ~executed. A filename can also be given when running from the Command Line Interface.

- Addition of the AmigaFlight® routines library.

- The AmigaGuide® ~library ~and ~program ~support is now also included with ~AmigaFlight®, ~together with ~Commodore’s ~standard ~installer software, both under a license from Commodore-Amiga Inc.

- The ~~user ~~interface ~~is ~now ~far ~friendlier ~and ~is ~slightly configurable - the window positions gadget positions, slider value, and options can now be saved to a preference file and are reloaded whenever Amigaflight® is run.

Changes to xasm :

- Re-written in C using standard ANSI calls. Compiled to approximately 13Kb.

- Re-written again to use specific Amiga system calls, and DOS library command line handling. Now compiles to just under 2Kb.

- Added ~multiple ~file ~name ~assembling, listing output, and a error listing window.

Changes to the AmigaGuide® Help and Documentation :

- The ~68000 ~instruction ~set ~help ~file ~was ~rewritten ~and ~made significantly more detailed.

- It now contains ~information on ~the internal structure of the 68000 chip, ~information ~on all 14 types of addressing modes, a brief comparison ~~between ~~the ~~Motorola ~68000 ~and ~the ~Intel ~8086, information on interrupt handling, the stack, and status flags.

- Each ~instruction ~description ~is ~as ~detailed as we could make it - size, data size, types of addressing modes available, number of clock cycles to execute, assembler ~syntax, ~and ~the status ~flags affected.

- It ~is ~now ~over ~8800 ~lines ~long. ~Approximately ~220 ~Kbytes of information.

- Further research was done ~in order to provide users with EVEN more on-line information, and the file was split up into several, more managable files.

- The files are now arranged into different instruction types and are quicker when loading.

- The help is now approximately 310 Kbytes of information.

Changes to Installer Script :

- Created initially to install specifically from "Flight Install:" to either a hard drive or floppy drive.

- Script modified to allow other languages (in the future).

- Changed to detect where abouts it is being installed from.
- Modified floppy drive installation to install a full bootable disk or just the AmigaFlight® program.

1.5 AmigaFlight® Help: Features

Features
========

Below is a run-down of the most important features of AmigaFlight®.

- Simple to use environment
- Emulates all Staffordshire University 68000 Flight Board routines
- Emulates all Staffordshire University Application Board routines
- Applications board consists of many control and output windows
- Works with all CPU’s, including 68040
- Uses Workbench 2 ‘newlook’ screen (clones Workbench mode)
- Uses AppIcons and AppMenuItems
- Conditional WB3 code to take advantage of new functions
- Comes with documentation in AmigaGuide® format
- Substantial on-line AmigaGuide® Help system
- Installation scripts for Installer
- Fully configurable environment

XCNT AmigaFlight® is a combination of XCNTs control program and assembling routines together a quality public domain software.

This release features the following piece of PD software :

```
       A68k    The Assembler
```

And contains the following routines from the FlightBoard System and the Applications Board :

```
      CONHEX    CRLF
      INADDR    INCH
      INCHEX    INKEY
      INSTHEX   OUTCH
      OUTHEX    OUT2HEX
      OUT4HEX   OUT8HEX
      OUTSTR    SPACE
      SPACES

      APPS_INIT CHECK_TIMER
      START_TIMER STOP_TIMER
```

1.6 AmigaFlight® Help: How to use AmigaFlight

Usage
=====

```
XCNT AmigaFlight® consists of two main programs.  These are :

  XASM    - The Pre-Processor / Assembler
  AmigaFlight - The Application


To use these programs, you must create a source file, assemble it using
the xasm pre-processor, and then load the created object file into the main
AmigaFlight executable.
```

1.7 AmigaFlight® Help: XASM - XCNT Assembler

```
XASM
====

XASM is the AmigaFlight® pre-processor for your source files.  This program
prepares your source files for assembly and then, using A68k, assembles it
into a standard object file.

The parameters for this program are :

    FILE/A/M  1 or more filenames
    LIST/S    Create an assembly list file

Any errors that occur during assembly will be output to a window for you to
look at and optionally printed to a listing file (.lst).

To invoke the AmigaFlight XCNT Assembler, create a source file with a `.asm'
extension and type "XASM [filename.asm]".

i.e.  If you create a source file called MileStone2.asm, you can assemble
      it by typing :

      XASM MileStone2.asm list

This will create an object file called "MileStone2.o" which can be loaded
into the main AmigaFlight control program and a source listing called
"MileStone2.lst".
```

1.8 AmigaFlight® Help : The AmigaFlight executable

```
The AmigaFlight Executable
==========================

The AmigaFlight executable is the main program into which created object
files are loaded and executed.

It can be run from Workbench by double-clicking on it's icon or by setting
the default tool of a created object file to AmigaFlight:AmigaFlight and
double-clicking on the object file.  You may also single-click on
AmigaFlight's icon, and then double click on an object file to initially
```

load that object file.
AmigaFlight can also be run from the Command Line Interface, giving an optional file name to load and execute.

The parameters for this program are :

```
    FILE/A    Initial file to load and execute
```

When the program has loaded, you will see a screen (in the same mode as your Workbench screen) with one full size window and, depending on any saved preferences, none to four application windows.

XCNT AmigaFlight is now under your control, via the following menu options :

```
Menu  Menu Item       Action
----  ---------       ------
Project :

  Open          Load an object file
  About         Display about box
  Quit          Quit AmigaFlight

Program :

  Run         Execute loaded object file
  Pause         Pause executing program
  Stop          Stop execting program

Windows :

  Show->Control Panel     Display Control Panel Window
  Show->Potentiometer     Display Potentiometer Window
  Show->Motor & Heater      Display Motor & Heater Window
  Show->LEDs        Display LEDs Window
  Show->All       Display All Windows

  Hide->Control Panel     Close Control Panel Window
  Hide->Potentiometer     Close Potentiometer Window
  Hide->Motor & Heater      Close Motor & Heater Window
  Hide->LEDs        Close LEDs Window
  Hide->All       Close All Windows

  Restore Positions->Control Panel  Restore Control Panel Window to
           position it was last opened
  Restore Positions->Potentiometer  Restore Potentiometer Window to
           position it was last opened
  Restore Positions->Motor & Heater Restore Motor & Heater Window to
           position it was last opened
  Restore Positions->LEDs     Restore LEDs Window to position
           it was last opened
  Restore Positions->All      Restore All Windows to positions
           they were last opened

  Arrange Positions->Control Panel  Arrange Control Panel Window to
           position at bottom right.
  Arrange Positions->Potentiometer  Arrange Potentiometer Window to
```

```
             position at bottom right.
  Arrange Positions->Motor & Heater Arrange Motor & Heaterl Window to
             position at bottom right.
  Arrange Positions->LEDs     Arrange LEDs Window to position
             at bottom right.
  Arrange Positions->All      Arrange All Windows to their
             positions at bottom right.
```

Preferences :

```
  Auto Execute        Set Auto Execute on/off
  Clever Windows        Set Clever Window on/off
  Reset Controls        Reset Control switches to default
             position
  Reset DIP Switches      Reset DIP switched to default
             position
  Reset Potentiometer     Reset Potentiometer to default
             value
  Reset To Defaults     Reset all to default positions
  Last Saved        Restore positions from preferences
             file
  Save          Save positions to preferences file
```

Help :

```
  AmigaFlight       View AmigaFlight.guide
  Assembly Language     View Assembly.guide
  Data Movement       View Data.guide
  Arithmetic Instructions     View Arithmetic.guide
  Logical Instructions      View Logical.guide
  Shifts And Rotates      View Shifts.guide
  Bit Test and Manipulation   View Bits.guide
  Binary Coded Decimal      View Binary.guide
  Flow Control        View Flow.guide
  System Control        View SysControl.guide
  A68K Assembler        View A68k.doc
```

1.9 AmigaFlight® Help : Source files

Source Files
============

The source code files used by this system can be any standard 680*0 assembly language file.

The only difference is that certain AmigaFlight routines for outputting text and results and reading from the keyboard can be called from the source program.

```
eg.   move.b      #'A',d0
    jsr         OUTCH Call OUTCH routine
    rts
```

All that this program does is output the letter 'A' on the AmigaFlight screen.

For students at Staffordshire University who are familiar with using the cross assembler on the PC's, there is no need to convert all of your source files as XASM will do this automatically for you. These conversions include :

```
    ORG <location>  Removed
    include "subrts.h"  Removed, included by default
    include "apps.h"  Removed, not required
    include "apps.asm"  Removed, not required
    trap  #0    Converted to an rts
```

1.10 AmigaFlight® Help: Object files

Object Files
============

All object files are standard Motorola 680*0 linkable objects and can be loaded into AmigaFlight and then executed.

Project icons can be added to these files and the Default Tool set to "AmigaFlight:AmigaFlight" so that they are loaded and executed automatically when double clicked on. A standard AmigaFlight project icon is included with this version.

1.11 AmigaFlight® Help: AmigaFlight's system requirements

System Requirements
===================

What do you need to run the AmigaFlight emulator?

At the moment, XCNT AmigaFlight has been written specifically for Workbench 2 and above, and WILL NOT work on older machines. Perhaps we will write a version to run on older machines but then there are many strong cases that are persuading us not to - the heavy use of Workbench 2 & 3 library routines for starters.

i) Quote from Micheal Berg (Programmer of PowerData) :

Writing good software that works ONLY on 2.04 and above will encourage Amiga owners all over the world to upgrade their hardware, which is both in their own interest - and in the interest of software developers all over the world.

ii) Taken from Amiga RKM - Libraries (Third Edition) :

Depending on your application and your market, you may choose to require the Release 2 operating system as a minimum platform. This can be a reasonable requirement for vertical markets and professional applications.

Also, AmigaFlight® requires the amigaflight library version 1 and lowlevel library version 40 (both supplied with this distribution).

1.12 AmigaFlight® Help: AmigaFlight emulated routines

Emulated Routines
=================

The emulated routines from the 68000 Flight Boards are :

```
          CONHEX       CRLF
          INADDR       INCH
          INCHEX       INKEY
          INSTHEX      OUTCH
          OUTHEX       OUT2HEX
          OUT4HEX      OUT8HEX
          OUTSTR       SPACE
          SPACES
```

With the following routines from the Applications Board :

```
          APPS_INIT      CHECK_TIMER
          START_TIMER    STOP_TIMER
```

Notes on using the timer subroutines :

To use the countdown timer, call START_TIMER to load the required time delay and start the timer. Then use CHECK_TIMER to see when the timer has reached zero (that is, the required time period has elapsed).

Elapsed time measurement can be achieved by calling START_TIMER with, say, the highest value allowed ($FFFFFF) at the start of the event to be timed and then calling STOP_TIMER at the end of event to be timed. Taking the value STOP_TIMER returned and subtracting it from the value you used to start the timer gives the total elapsed time in clock ticks. (One clock tick equals four microseconds). Note that, once started, the clock cycles round continually so that after it reaches zero it resets to the initial time period and continues counting down. (You could make sure that the clock has not wrapped round by calling CHECK_TIMER straight after).

1.13 AmigaFlight® Help: Compatibility with XCNT AmigaFlight

Compatability
=============

XCNT AmigaFlight has been written and tested on the following machines :

```
    Commodore AMIGA 4000/040, Kickstart 3.0
    Commodore AMIGA 4000/030, Kickstart 3.0
    Commodore AMIGA 1200,   Kickstart 3.0
    Commodore AMIGA 500+,   Kickstart 2.04
```

It should also be compatible with the rest of the AMIGA range with KickStart 2.04 upwards. However, should you experience any problems then let us know so we can fix it !

1.14 AmigaFlight® Help: Installing AmigaFlight®

Hard Drive Installation
=======================

It is very easy to install AmigaFlight to a hard disk, all you have to do is double click on the provided installation procedure, select hard disk installation, and supply a few simple pieces of information about where to install AmigaFlight.

Floppy Drive Installation
=========================

To install AmigaFlight to a floppy drive, you will need a blank disk, your original Workbench disks, and some patience.
Click on the installer icon, select floppy disk install, and then follow the on-line instructions.

1.15 AmigaFlight® Help: Future releases of XCNT AmigaFlight

Future
======

AmigaFlight will be updated in the future to include any new routines that we or users feel appropriate.

We will also try to improve the AmigaFlight enviroment by a possible merge of XASM into the AmigaFlight program, so you could simply load source files into AmigaFlight and it would assemble them and load the object file automatically.

A hardware AmigaFlight Applications Board is currently under consideration. This project will be available as a kit or fully constructed. Possible items on the board will be :

- The 'traffic light' LEDs
- Potentiometer
- Fan
- Heater
- DA converter
- AD converter
- 7 segment numerical display for 'traffic lights'

1.16 AmigaFlight® Help: Copyright

Copyright
=========

The AmigaFlight executable, library, and documentation are all Copyright ©1992-1994 by XCNT Productions, All Rights Reserved.

AmigaFlight is NOT public domain. The authors retain all rights to the program.

See Distribution for more details about distributing AmigaFlight.

1.17 AmigaFlight® Help: Distribution

Distribution
============

AmigaFlight may be freely distributed through the normal channels on the condition that NO charge be made for the AmigaFlight software and accompanying documents.
An appropriate charge MAY be made for any disk/packaging/postage that the AmigaFlight software is supplied on ONLY.

1.18 AmigaFlight® Help: Program Disclaimer & Liability

Disclaimer & Liability
======================

This program is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of this program is assumed by you.

Should the program prove defective, you alone assume the entire cost of all necessary servicing, repair, or correction. Further more, the authors of AmigaFlight neither warrant, guarantee, or make any representations regarding the use of, or the results of the use of, the program in terms of corrections, accuracy, reliability, currentness, or otherwise; and you rely on the program and results solely at your own risk.

The authors of AmigaFlight can in no event be held responsible for any data or information which may be lost or rendered inaccurate by AmigaFlight.

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE; NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY OR RESPONSIBILITY IS ASSUMED.

AMIGAGUIDE SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE; NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY

OR RESPONSIBILITY IS ASSUMED.

1.19 AmigaFlight® Help: About Andrew Duffy (MIDI)

Midi
====

Andrew Duffy is from Watford Village in Northamptonshire and studied for his A-levels at Nene College in Nothampton before going to Staffordshire University in September 1992 to study on a two year HND in Computing, but transferred to the BSc Computer Science course after the first year. Before university he had done no assembly language at all.

Joined XCNT as an official member in November 1992.

Spent industrial placement at Eurocopter, Munich.

Programmed most of the code, and worked on the help files.

1.20 AmigaFlight® Help: About Alan James (EXEC)

Exec
====

Alan James is from Northampton and studied for his ‘A’ levels at Nene College in Northampton before going to Staffordshire University in September 1992 to study on a four year BSc in Computing.

Alan had a little experience of assembly language before this course, mainly from a work experience period at Bullfrog.

Joined XCNT as an official member in November 1992.

Failed second year degree and, following this, also failed the second year HND. We have recently heard that he is currently a C programmer for Gallop.

Didn’t do much really.

1.21 AmigaFlight® Help: About Chris Morris (NEMESIS)

Nemesis
=======

Chris Morris, from near Farnborough, is one of the founder members of XCNT, set up by himself and a friend in October 1990. Other ’divisions’ of XCNT are ’XCNT Construction - Oxford Division’ and ’XCNT Electronics - Farnborough Division’

Chris has studied BTEC Computing & Mathematics and BTEC Electronic &

Electrical Engineering.

Chris went to Stafforshire University in September 1992 to study on a four year BSc in Computer Science.

Spent industiral placement at Advanced Recognition Ltd, Windsor.

Developed the help files and researched 68000 for them.

1.22 AmigaFlight® Help: Contacting XCNT

```
Contacting XCNT
===============

XCNT can be contacted at the following email addresses until July 1996 :

      a.duffy@staffs.ac.uk
      c.morris@staffs.ac.uk

From July onwards you should send us Snail-Mail at the following addresses :

MIDI :    Andrew Duffy,
      Redridge,
      Main Street,
      Watford Village,
      Northants.
      NN6 7UY.
      England.


CHRIS :   Chris Morris,
      7 Shelley Walk,
      Yateley,
      Camberley.
      Surrey.
      GU17 7YG.
      England.


Reporting Bugs
==============
```

When reporting bugs, it would help us a great deal if you could try to be as specific as possible about the conditions under which the program failed.
For example, your configuration, what exactly you were doing, what options you were using, what else might have been running, etc. Usually whatever file caused the problems will help us out too, if it’s relatively small you can email the source to us, or uuencode the object file, or whatever.

```
Suggestions
===========
```

We are always open to suggestions, although we are more likely to implement simpler suggestions than complex suggestions.

1.23 AmigaFlight® Help: Known Bugs

Known Bugs (Arrrgggggghhhhhh!)
=============================

Well, this is the section where I would like to put "There are no known bugs in this release" but I'm afraid I would be lying.

Program Menu : Sometimes it gets confused as to whether the program is running or stopped, so menu items may become incorrectly ghosted out.

Stopping Programs : This is a particularly risky operation by it's nature and can sometimes lead to AmigaFlight 'hanging' or causing a software failure. Since the users programs are launched as a new task AmigaFlight has next to no control over them and they can only be stopped by a call to RemTask(). RemTask() can only be done safely when the task is in a state of not owning any resources or and not waiting for any I/O, AmigaFlight will warn you of this condition when you try to stop your code, but it is up to you whether you go ahead or not. If you do stop your code during this dangerous situation then it is likely that the next program you run will crash.

Application Routines : These are still VERY unstable, sometimes they work lovely, other times they crash horribly.

Closing Apps Windows : Sometimes when closing an output window it will cause the user program to stop. This is only an inconvenience as you should be able to stop your program and restart it with the window now closed.

KingCON : KingCON seems to handle "RAW:" windows slightly differently to the normal built in console handler. This can cause strange effects when typing into the AmigaFlight window, so either disable KingCON, or load it only for "CON:".

I'm sure there are plenty more...

1.24 AmigaFlight® Help: Acknowledgements

Acknowledgements
================

- Charlie Gibbs for A68k
- Pete Cheetham, Staffordshire University, for initial assistance
- CAIN, Staffordshire University, for preliminary testing
- EXEC for doing nothing for the past 16 months

1.25 AmigaFlight® Help: CONHEX - Convert to Hex

```
CONHEX
======
  Converts an upper case ASCII characater value to a hex value.


Inputs
------
  Upper case ASCII value to be converted in lower byte of register
  d0.


Outputs
-------
  Hex value returned in lower byte of d0.
  C bit set if a valid hex number.


Affected Status Flags
---------------------
  C  Set if valid Hex number


Affected Registers
------------------
    BEFORE        AFTER
  D0 ASCII character value  D0 Hex value / Unchanged
```

1.26 AmigaFlight® Help: CRLF - Carriage Return and Line Feed

```
CRLF
====
  Outputs  a  carriage  return  and  a line feed, thus returning the
  cursor to the  next line  on the display, scrolling the page up if
  the current position of the cursor is at the bottom.


Inputs
------
  None.


Outputs
-------
  ASCII characters 10(d) and 13(d)
```

1.27 AmigaFlight® Help: INADDR - In Address

```
INADDR
======
  Asks for, and validates, a memory address from the user.
```

```
Inputs
------
  Hex memory address from keyboard.


Outputs
-------
  Returns  the  entered address  in  register  a0, and the number of
  characters entered from the keyboard in register d2.


Affected Registers
------------------
    BEFORE        AFTER
  D2        D2 Number of characters pressed
  A0        A0 Address entered
```

1.28 AmigaFlight® Help: INCH - In Character

```
INCH
====
  Waits for a key  to  be  pressed, then returns  the ASCII value of
  that key in register d0.


Inputs
------
  Character from keyboard.


Outputs
-------
  ASCII code of  key pressed  in lower byte of register d0.
  Rest of d0 remains unchanged.


Affected Registers
------------------
    BEFORE        AFTER
  D0        D0 ASCII code of character
```

1.29 AmigaFlight® Help: INCHEX - In Character Hex

```
INCHEX
======
  Waits for a key to be pressed, the displays it and returns the hex
  value (0-9, A-F) of that key.


Inputs
------
```

```
  Character from keyboard.


Outputs
-------
  Returns hex value in lower byte of d0 or ASCII code if non hex.
  C bit set if not a hex number.


Affected Status Flags
---------------------
  C  Set if not a valid hex number


Registers
---------
    BEFORE         AFTER
  D0        D0 Hex value / ASCII Character code
```

1.30 AmigaFlight® Help: INKEY - In Key

```
INKEY
=====
  Checks to see if a key on the keyboard has been pressed but not
  read using INCH.  i.e. A character is waiting in the buffer.


Inputs
------
  Keyboard press.


Outputs
-------
  None.


Affected Status Flags
---------------------
  Z  Cleared when key is pressed
```

1.31 AmigaFlight® Help: INSTHEX - In String Hex

```
INSTHEX
=======
  Reads characters  from the ~keyboard until ~a non-hex  charater is
  entered.


Inputs
------
  ASCII characters 0 to 9 and A to F
```

Outputs

Returns the hex number in register d1, non-hex terminator in register d0, and the number of characters entered in register d2.

Status Flags

N	Not affected
Z	Not affected
V	Not affected
C	Not affected
X	Not affected

Registers

BEFORE	AFTER
D0	D0 Non-hex terminator character
D1	D1 Hex value
D2	D2 Number of characters
D3	D3 Unchanged
D4	D4 Unchanged
D5	D5 Unchanged
D6	D6 Unchanged
D7	D7 Unchanged
A0	A0 Unchanged
A1	A1 Unchanged
A2	A2 Unchanged
A3	A3 Unchanged
A4	A4 Unchanged
A5	A5 Unchanged
A6	A6 Unchanged
A7	A7 Unchanged

1.32 AmigaFlight® Help: OUTCH - Out Character

OUTCH
=====

Ouputs a single charater to the screen.

Inputs

ASCII value of character to be printed in lower byte of register d0.

Outputs

ASCII character on the screen.

```
Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0 ASCII val of char to print D0 Unchanged
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.33 AmigaFlight® Help: OUTHEX - Out Hex

```
OUTHEX
======
  Outputs the  value of the  least significant nibble of register d0
  as a one digit hex number.


Inputs
------
  Value to be output in least significant nibble of register d0.


Outputs
-------
  Prints least significant  nibble (4 bits) of  register d0 as a one
  digit hex number.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Register Contents
-----------------
    BEFORE        AFTER
  D0 LS nibble (4 bits) to print  D0 Corrupted
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.34 AmigaFlight® Help: OUT2HEX - Out 2 Hex

```
OUT2HEX
=======
  Outputs the value of the least significant byte of  register d0 as
  a two digit hex number.


Inputs
------
  Value to be output in least significant byte of register d0.


Outputs
-------
  Prints least significant byte of  register d0  as a  two digit hex
  number.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0 LS byte (8 bits) to print  D0 Corrupted
  D1         D1 Unchanged
  D2         D2 Unchanged
  D3         D3 Unchanged
  D4         D4 Unchanged
  D5         D5 Unchanged
  D6         D6 Unchanged
  D7         D7 Unchanged

  A0         A0 Unchanged
  A1         A1 Unchanged
  A2         A2 Unchanged
  A3         A3 Unchanged
  A4         A4 Unchanged
  A5         A5 Unchanged
  A6         A6 Unchanged
  A7         A7 Unchanged
```

1.35 AmigaFlight® Help: OUT4HEX - Out 4 Hex

OUT4HEX
=======

Outputs the value of the least significant word of register d0 as a four digit hex number.

Inputs

Value to be output in least significant word of register d0.

Outputs

Prints least significant word of register d0 as a four digit hex number.

Status Flags

```
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected
```

Registers

```
    BEFORE        AFTER
  D0 LS word (16 bits) to print D0 Corrupted
  D1         D1 Unchanged
  D2         D2 Unchanged
  D3         D3 Unchanged
  D4         D4 Unchanged
  D5         D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.36 AmigaFlight® Help: OUT8HEX - Out 8 Hex

```
OUT8HEX
=======
  Outputs the value  of the long  word in  register d0  as an  eight
  digit hex number.


Inputs
------
  Value to be output in long word of register d0.


Ouputs
------
  Prints out the  four digit hex  value of the  contents of register
  d0.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0 32 bit number to print D0 Corrupted
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.37 AmigaFlight® Help: OUTSTR - Out String

```
OUTSTR
======
  Outputs an ASCII string terminated by a null character (Hex 0).


Inputs
------
  Address of the start of the ASCII string to print in register a6.


Outputs
-------
  Prints out characters until ASCII character 0 reached.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Corrupted
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6 Start address of string  A6 Corrupted
  A7        A7 Unchanged
```

1.38 AmigaFlight® Help: SPACE - Space

```
SPACE
=====
  Outputs a single space character (ASCII 32).


Inputs
------
  None


Outputs
-------
  ASCII character 32(d)


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Corrupted
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.39 AmigaFlight® Help: SPACES - Spaces

```
SPACES
======
  Outputs a specified number of space characters (ASCII 32).
```

```
Inputs
------
  Number of spaces to print in lower word of register d0.


Outputs
-------
  Prints to the screen specified number of character ASCII 32(d)


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0 Number of spaces   D0 Corrupted
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.40 AmigaFlight® Help: APPS_INIT - Applications Initialisation

```
APPS_INIT
=========
  Initialises AmigaFlight / Flight 68000  board ready  for using the
  Applications Board.

  Each  of the  eight bits  of Port A  are configured  for  input or
  output according to the value in the lower byte of register d0.

  Port B is configured so that all pins are set to output.


Inputs
------
```

```
  Bits 7  to 0 of d0 are sent to bits 7 to 0,  respectively, of  the
  data direction register  for  Port A,  thus selecting  each of the
  pins of Port A for input or output.

  Note: That a one sent to  a  bit in the  data direction register
    means that the corresponding pin is set to output, sending
    a zero means that it is set up to input.


Outputs
-------
  Sends input/output information to flight board.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Unchanged
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.41 AmigaFlight® Help: CHECK_TIMER - Check Timer

```
CHECK_TIMER
===========
  Checks to see if the countdown timer has reached zero yet. It does
  this by checking a zero detect status bit which is set to one when
  the countdown timer reaches zero and remains set to  one until  it
  is reset (by calling START_TIMER, for example).


Inputs
```

```
------
  None.


Outputs
-------
  Z flag is set to  0 if the  timer has reached zero (at least once)
  or set to 1 otherwise.


Status Flags
------------
  N  Not affected
  Z  Cleared when timer reaches zero
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Unchanged
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.42 AmigaFlight® Help: START_TIMER - Start Timer

```
START_TIMER
===========
  Initialises  the countdown  timer to  the required time period and
  starts the clock. Once started, the clock cycles round continually
  so  that after  it reaches zero  it resets  to the  required  time
  period and continues counting down.


Inputs
------
  Bits 23 to 0 of  register d0 contain the  required time period (in
  clock ticks). One  clock tick  equals  four  microseconds (that is, four
  millionths of a second).
```

```
Outputs
-------
  None.


Status Flags
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Unchanged
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.43 AmigaFlight® Help: STOP_TIMER - Stop Timer

```
STOP_TIMER
==========
  Stops the countdown timer and returns the clock value reached.


Inputs
------
  None.


Outputs
-------
  Returns timer value in bits 23 to 0 of register d0.


Status Flags
```

```
------------
  N  Not affected
  Z  Not affected
  V  Not affected
  C  Not affected
  X  Not affected


Registers
---------
    BEFORE        AFTER
  D0        D0 Timer value in bits 23 to 0
  D1        D1 Unchanged
  D2        D2 Unchanged
  D3        D3 Unchanged
  D4        D4 Unchanged
  D5        D5 Unchanged
  D6        D6 Unchanged
  D7        D7 Unchanged

  A0        A0 Unchanged
  A1        A1 Unchanged
  A2        A2 Unchanged
  A3        A3 Unchanged
  A4        A4 Unchanged
  A5        A5 Unchanged
  A6        A6 Unchanged
  A7        A7 Unchanged
```

1.44 AmigaFlight® Help: Index

INDEX
=====

```
AmigaFlight Routines :

   APPS_INIT
   CHECK_TIMER
   CONHEX
   CRLF
   INADDR
   INCH
   INCHEX
   INKEY
   INSTHEX
   OUTCH
   OUTHEX
   OUT2HEX
   OUT4HEX
   OUT8HEX
   OUTSTR
   SPACE
   SPACES
   START_TIMER
   STOP_TIMER

Programmers :

   EXEC    - Alan James
   MIDI    - Andrew Duffy
   NEMESIS - Chris Morris
   Contacting XCNT
```