```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                      %%
%%                #######    #####      #####    #######                %%
%%                #         #     #    ##           #                   %%
%%                ####      #######     ####        #                   %%
%%                #         #     #        ##       #                   %%
%%                #         #     #    #####        #                   %%
%%                                                                      %%
%%                      FAlconSioc-dataTransfer                         %%
%%                                                                      %%
%%                      (c) michi "mihi" hirczy                         %%
%%  Based on code by Rich "Flareless" Sherkin and Erwin "Eagle9" Neyt   %%
%%                                                                      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
*****************
*  Disclaimer   *
*****************
```

FAST is provided "as is" and actually is still a Beta-version. Use it on your own risk! ;-)

```
*****************
* Installation  *
*****************
```

To install FAST, simply start the *setup.exe* and tell the installer where to put the FAST files.

Since it is completely written in VisualBasic6, some files of the VB6Runtime are required to run FAST. These files are included in the setup and will be installed automatically (for more details see install.txt)

```
*****************
* Configuration *
*****************
```

FAST is either configured through the config file *fast.cfg* or through the GUI (but changes in the GUI are not automatically written to the config file)

A typical config file looks like this:

```
SIOCServer=192.168.1.1
SIOCPort=8092
DataModel=AF
F4Timer=30
```

Here's a short description of the fields in the config file (and GUI):

SIOCServer:	the IP address of the machine where the SIOC server is running
SIOCPort:	the port SIOC is listening (usually 8092)
F4Timer:	intervall to poll Falcon data in milliseconds
DataModel:	version of Falcon to connect to: AF for AlliedFore, BMS for BMS

```
*****************
*   Overview    *
*****************
```

FAST is a Tool to send Data from Falcon (either AlliedForce or any BMS derivate) to a SIOC server. Data is send over TCP/IP through the IOCProtocol.

```
*****************
* Starting FAST *
*****************
```

When starting up, FAST reads the config file *fast.cfg* to get the needed parameters and then tries to connect to Falcon and SIOC according to the values set in the config file. When the SIOC server is not running, FAST does not automatically try to reconnect - you will have to connect by pressing the "connect" button.

When Falcon is not running, FAST will poll for Falcon in the interval given in the "F4Timer" field.

So the best starting order for FAST to work best is:

1.) start SIOC server software
2.) start FAST
3.) start Falcon

```
**********************
* SIOC configuration *
**********************
```

When you start SIOC, you can go to IOCPCONSOLE and open the file *iocp_config.con* - all SIOC variables used are labeled now. Since the IOCProtocol can only transport integer values, you will have to modify the values in SIOC (i.e. the mach value 86 would have to be divided by 100 to get the correct value of 0.86)

Once you look at the values in IOCPCONSOLE you will see, how to manipulate the values. (As soon as I have the time, a proper .ssi-file will be added to the installation which will do these conversions automatically)

FAST uses the following SIOC variables (comments in brackets):

VAR 0001 - internal bits (not implemented yet)

VAR 0010 - Caution Panel Lights

The following bits are used:

```
  FLCSFAULT = 0
  ENGINEFAULT = 1
  AVIONICSFAULT = 2
  SEATNOTARMED = 3
  ELECSYS = 4
  SEC = 5
  EQUIPHOT = 6          (is not in Memory, using CautionPanel OVERHEAT also)
  NWSFAIL = 7
  PROBEHEAT = 8
  FUELOILHOT = 9
  RADARALT = 10
  ANTISKID = 11
  CADC = 12              (is not in Memory, will always be OFF)
  INLETICING = 13        (is not in Memory, will always be OFF)
  IFF = 14
  HOOK = 15
  STORESCONFIG = 16      (is not in Memory, will always be OFF)
  OVERHEAT = 17
  NUCLEAR = 18           (is not in Memory, will always be OFF)
  OXY_LOW = 19
  ATFNOTENGAGED = 20     (is not in Memory, will always be OFF)
  EEC = 21               (is not in Memory, using ECM[not on real CP])
  CABINPRESS = 22
  FWDFUELLOW = 23
  AFTFUELLOW = 24
  BUC = 25
```

VAR 0011 - Front Panel Lights

The following bits are used:

```
 ' lefteyebrowlights
  MASTERCAUTION = 0
  TFFAIL = 1
  ' left indexer
  AOAHIGH = 2
  AOAMIDDLE = 3
  AOALOW = 4
  ' right indexer
  RDY = 5
  ARNWS = 6
  DISC = 7
  ' right eyebrowlights
  ENGFIRE = 8
  ENGINE = 9            (is not in Memory, using CautionPanel EngineFault)
  HYDOILPRESS = 10
  FLCS = 11             (is not in Memory, using CautionPanel FltCtrlSys)
  DBUON = 12            (is not in Memory, using DUAL->in AF on CP)
  TOLDGCONFIG = 13
  CANOPY = 14
  OXYLOW = 15
  ' tws prime
  HANDOFF = 16
  LAUNCH = 17
  PRIMODE = 18
  NAVAL = 19
  UNKNOWN = 20
  TGTSEP = 21
  ' misc panel
  ECMON = 22
  TFRACTIVE = 23
```

VAR 0012 - Console Lights

The following bits are used:

```
  JFSON = 0
  AVTRON = 1
  ' epu
  EPUON = 2
  HYDRAZIN = 3
  AIR = 4
  ' elec
  FLCSPMG = 5
  MAINGEN = 6
  STBYGEN = 7
  EPUGEN = 8
  EPUPMG = 9
  TOFLCS = 10
  FLCSRLY = 11
  BATFAIL = 12
  ' flight control
  FLTCTRLFAIL = 13
  ' marcer beacon light(s)
  MARKER_OUTER = 14
  MARKER_MIDDLE = 15
  MARKER_INNER = 16

  NOSEWHEEL = 17
  LEFTWHEEL = 18
  RIGHTWHEEL = 19
```

```
GEARHANDLE = 20
' tws aux
AUXSRCH = 21
AUXACT = 22
AUXLOW = 23
AUXPWR = 24
```

VAR 0013 - Instrument Flags (not implemented yet)

The following bits are used:

```
HSI_To = 0              ' HSI_FLAG_TO_TRUE
HSI_Ils = 1             ' HSI_FLAG_ILS_WARN
HSI_Course = 2          ' HSI_FLAG_CRS_WARN
HSI_Init = 3            ' HSI_FLAG_INIT
TotalFlags = 4          ' HSI_FLAG_TOTAL_FLAGS
ADI_OFF = 5             ' ADI OFF Flag
ADI_AUX = 6             ' ADI AUX Flag
ADI_GS = 7              ' ADI GS FLAG
ADI_LOC = 8             ' ADI LOC FLAG
HSI_OFF = 9             ' HSI OFF Flag
BUP_ADI_OFF = 10        ' Backup ADI Off Flag
VVI_OFF = 11            ' VVI OFF Flag
AOA_OFF = 12            ' AOA OFF Flag
```

To check any of these bits, use the "TESTBIT" function of SIOC. If you want to check, if the NWS light is on in Falcon, use:

```
C0 = TESTBIT V0011, 6
```

ATTENTION! Bits start with "0"!

so, to check the MasterCaution, you have to use

```
C0 = TESTBIT V0011, 0
```

VAR 0015-0018

are the original Falcon lightbit values. You can use these or the variables I implemented above.

To get more information about these values, please check the file *IOCP.pdf* which comes with FAST, it's from "MachoMan's" ***supertron*** program. There are some differences between the AF lightbits and the BMS lightbits (shown in *IOCP.pdf*):

- in lightbits the bits "WOW", "AutoPilotOn" and "TFR_STBY" are not available
- in lightbits2 the last bit "ENGINE" is not available
- in lightbits3 the bit "Power_Off" is not there, instead it is called "OnGround" - sadly, this bit doesn't really indicate, if the plane is on the ground, just the time from rampstart to main generator running
- in lightbits3 all the bits from "Eng2_Fire" to "RightGearDown" are not in AF

VAR 0020-0099

Data values out of Falcon - to know, which value is what, please check the IOCPConsole with the included *iocp_config.con* file. At the moment, only values from 20-68 are used, so there are some free spots for future data...

DED and PFL

Since SIOC does not support text to be transported, DED and PFL data is stored as 1 character ASCII-value in 1 sioc variable, starting at variable 100.

So VARS 0100 - 0124 hold the characters for the first DED line, 0125-0149 for the 2nd DED line, 0150-0174 the 3rd DED line, 0175-0199 the 4th DED line and 0200-0224 the 5th DED line.

The PFL starts at VAR 0300 and ends at 0424.

The inverted character positions are not implemented yet

```
*****************
*  Known Bugs   *
*****************
```

- Some values are not read correctly from Falcon: fwd and aft fuel
- Inverted characters of DED and PFL are not implemented yet

```
*****************
*   Changelog   *
*****************
```

V0.92
*) own light values added

V0.91
First release