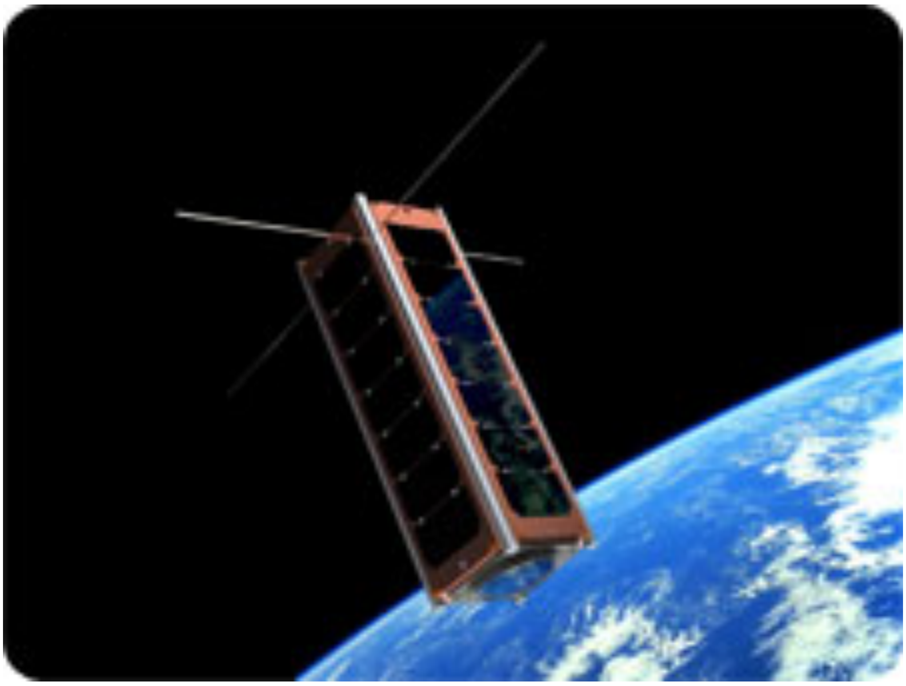
Anexo II
Manual de inicio rápido



Manual – Instalación del Software

Título:	Manual – Instalación del Software
Entidad:	Solar MEMS Technologies
Autores:	Plaza Gomariz, Pablo
Fecha:	15/7/2016

[Página intencionalmente en blanco]

1. Requisitos

Para comenzar a usar NanoMind es necesario:

- Distribución GNU/LINUX, por ejemplo Ubuntu o Debian.
- Placa NanoMind.
- Convertidor USB-Serial.
- Programa Minicom instalado.
- Amontech JtagKEY-Tiny.
- CEPHEUS Software folder.

No es necesario estar registrado en github para poder recibir código de gomspace, ya que actualmente los repositorios no se encuentran abiertos a los suscriptores.

Para el proyecto se ha hecho uso de la distribución de Ubuntu 14.04 LTS, descargada de la página oficial: https://www.ubuntu.com/

2. Toolchain

Para poder utilizar la placa es necesario instalar el último ARM toolchain. Está construido especialmente para la placa Nanomind y preparado para sistemas de 32 y 64 bits.

Para su instalación:

1. En CEPHEUS Software folder, se monta la imagen Nanomind.iso.
2. Dentro de la carpeta 4F012F70C81A4 se tienen los archivos *arm-none-eabi-gcc-4.6.4-x64-tar.gz* y *arm-none-eabi-gcc-4.6.4-x86-tar.gz* los cuales son los toolchain para los sistemas de 64 y 32 bits respectivamente.
3. Se selecciona el archivo adecuado para el equipo y se extrae en ‘usr/local/arm’.
4. Se edita el ~/.profile del sistema, añadiéndose la siguiente sentencia:

```
PATH=”usr/local/arm/bin:$PATH”
```

5. Se reinicia el equipo.

Este proceso es esencial para el correcto funcionamiento del Nanomind con el equipo, ya que este archivo contiene las instrucciones necesarias para la programación del procesador de la placa.

3. Instalación Eclipse

Para la visualización del Proyecto se ha hecho uso de Eclipse Mars en su distribución para Ubuntu. No es necesario instalar ningún compilador ya que para este fin se utilizará el script waf. La forma más rápida para instalar es seguir las instrucciones en la página oficial; https://eclipse.org/downloads/

4. Distribución de carpetas

Cada proyecto está estructurado de la siguiente forma:

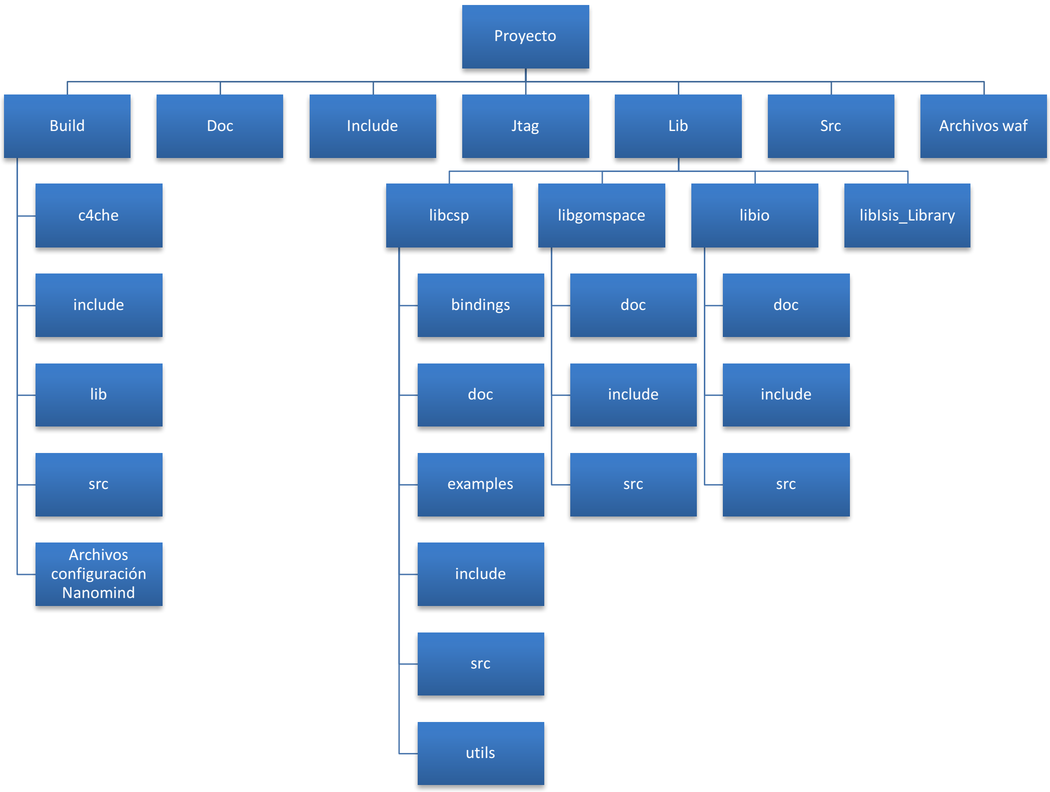
- En la carpeta **build**, según el esquema anterior, se tiene:
  - *C4che*: se incluyen archivos de configuración de la build del proyecto
  - *Include*: archivos de cabecera con la configuración del Cubesat Space Protocol, la placa Nanomind, los subsistemas de GomSpace y las entradas y salidas de dispositivos.
  - *Lib* y *src*: archivos de salida de la compilación del código.
- En **include** se encuentran todas las cabeceras de las bibliotecas de ISIS.
- En **Jtag** aparecen archivos de configuración para la programación del microprocesador de la placa.
- En **lib** se encuentran todas las librerías propias del Nanomind, las cuales fueron distribuidas junto con la placa..
- En **src** aparecen los archivos de código para el inicio de la placa; interrupciones, reloj interno en tiempo real, arranque del sistema freeRTOS, etc. En esta carpeta se irán añadiendo el código de control del satélite.
- **Waf** es el programa encargado de la compilación del código y de la programación del microcontrolador de la placa.

5. Librerías

GomSpace proporciona 6 librerías destinadas a distintos usos:

- Libgomspace: contiene drivers, el sistema freeRTOS y utilidades usadas en los productos GomSpace.
- Libcsp: contiene el Cubesat Space Protocol stak.
- Libio: librería para la integración de otros productos de GomSpace con Nanomind.
- Libstorage: sistemas de archivos UFFS y FAT.
- Libcdh: planificador de vuelo.
- Libadcs: control de actitud. Contiene: B-dot detumbling, filtro de Kalman, etc.

En el caso del satélite CEPHEUS, solo se encuentran disponibles las tres primeras librerías. Por lo tanto la gestión de la memoria, el cálculo de los tiempos de conexión en una órbita o el control de actitud, deben ser totalmente desarrollados.


Proyecto
Build
Doc
Include
Jtag
Lib
Src
Archivos waf
c4che
include
lib
src
Archivos configuración Nanomind
libcsp
libgomspace
libio
libIsis_Library
bindings
doc
examples
include
src
utils
doc
include
src
doc
include
src

6. Waf

Waf es un moderno sistema basado en python, el cual debe estar instalado en el sistema, que implementa el software de la placa Nanomind. Se encuentra, y debe hacerlo, en la carpeta raíz de cada proyecto que se quiera compilar y programar. Gracias a él se pueden configurar ciertos parámetros tanto de Nanomind como de las bibliotecas.

A continuación se adjunta un extracto de las distintas opciones de las que dispone:

```
waf [commands] [options]

Main commands (example: ./waf build -j4)
  build    : executes the build
  clean    : cleans the project
  configure: configures the project
  dist     :
  distcheck: checks if the project compiles (tarball from 'dist')
  distclean: removes the build directory
  eclipse  :
  install  : installs the targets on the system
  list     : lists the targets to execute
  program  :
  step    : executes tasks in a step-by-step fashion, for debugging
  uninstall: removes the targets installed
  update   : updates the plugins from the *waflib/extras* directory
  upload   :

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -j JOBS, --jobs=JOBS  amount of parallel jobs (4)
  -k, --keep               keep running happily even if errors are
found
  -v, --verbose         verbosity level -v -vv or -vvv [default: 0]
  --nocache             ignore the WAFCACHE (if set)
  --zones=ZONES        debugging zones (task_gen, deps, tasks, etc)
  --arch=ARCH           Set architechture [x86, avr32, arm, avr]
  --mcu=MCU             Set CPU type [atmega128 atmega1281 uc3a0512
uc3b0256
                        ap7000]
  --toolchain=TOOLCHAIN
                        Set toolchain prefix

  configure options:
    -o OUT, --out=OUT   build dir for the project
    -t TOP, --top=TOP   src dir for the project
    --prefix=PREFIX    installation prefix [default: '/usr/local/']
    --download          try to download the tools if missing
```

```
  build and install options:
    -p, --progress      -p: progress bar; -pp: ide output
    --targets=TARGETS   task generators, e.g. "target1,target2"

  step options:
    --files=FILES       files to process, by regexp, e.g.
                        "*/main.c,*/test/main.o"

  install/uninstall options:
    --destdir=DESTDIR   installation root [default: '']
    -f, --force         force file installation

  libgomspace options:
    --install-gomspace  Installs GOMSPACE headers and lib
    --with-log=WITH_LOG
                        Choose log backend [disable, printf, color,
cdh]
    --with-console=HANDLE
                        Choose USART[0,1,2,3] for console
    --console-disable-history
                        Disable console history completely
    --console-history-len=LEN
                        Choose length for history
    --console-input-len=LEN
                        Choose length for input
    --with-i2c=DRIVER   Set driver [twi, pca9665]
    --with-freertos-config=CONFIG
                        Set FreeRTOS configuration file
    --enable-supervisor
                        Compile supervisor
    --enable-sdram      Enable SDRAM for AVR32 UC3A
    --enable-spi        Enable SPI for AVR32 UC3A
    --enable-driver-debug
                        Enable driver debug
    --enable-gosh       Compile Gomspace Shell
    --gosh-const         Store GOSH commands in const (sorting will
be disabled)
    --enable-rpcx       Compile Gomspace RPCX

  libio options:
    --enable-ftp-server
                        Enable FTP server
    --enable-ftp-client
                        Enable FTP client (posix only)
    --install-io        Installs IO headers and lib
    --enable-nanomind-client
                        Enable client code for NanoMind
    --enable-nanohub-client
                        Enable client code for NanoHub
```

```
    --enable-nanopower-client
                        Enable client code for NanoPower
    --enable-nanocom-client
                        Enable client code for NanoCom
    --enable-nanocam-client
                        Enable client code for NanoCam
    --enable-csp-client
                        Enable client code for CSP
    --enable-if-sia     Enable Interface SIA

  libcsp options:
    --includes=INCLUDES
                        Add additional include paths. Separate with
comma
    --install-csp       Installs CSP headers and lib
    --disable-output    Disable CSP output
    --disable-verbose   Disable filename and lineno on debug
    --enable-rdp        Enable RDP support
    --enable-qos        Enable Quality of Service support
    --enable-promisc    Enable promiscuous mode support
    --enable-crc32      Enable CRC32 support
    --enable-hmac       Enable HMAC-SHA1 support
    --enable-xtea       Enable XTEA support
    --enable-bindings   Enable Python bindings
    --enable-examples   Enable examples
    --enable-if-i2c     Enable I2C interface
    --enable-if-kiss    Enable KISS/RS.232 interface
    --enable-if-can     Enable CAN interface
    --with-driver-can=CHIP
                        Build CAN driver. [socketcan, at91sam7a1,
at91sam7a3 or
                        at90can128]
    --with-driver-usart=DRIVER
                        Build USART driver. [windows, linux, None]
    --with-drivers=PATH
                        Set path to Driver header files
    --with-os=OS       Set operating system. Must be either 'posix',
'macosx',
                        'windows' or 'freertos'
    --with-freertos=PATH
                        Set path to FreeRTOS header files
    --with-rdp-max-window=SIZE
                        Set maximum window size for RDP
    --with-max-bind-port=PORT
                        Set maximum bindable port
    --with-max-connections=COUNT
                        Set maximum number of concurrent connections
    --with-conn-queue-length=SIZE
                        Set maximum number of packets in queue for
a connection
```

```
    --with-router-queue-length=SIZE
                        Set maximum number of packets to be queued
at the input
                        of the router
    --with-padding=BYTES
                        Set padding bytes before packet length field
    --with-loglevel=LEVEL
                        Set minimum compile time log level. Must be
one of
                        'error', 'warn', 'info' or 'debug'

  NanoMind options:
    --hostname=HOSTNAME
                        Set system hostname
    --model=MODEL       Set system model
    --rom               Compile ROM image
    --config-sd-cs=CONFIG_SD_CS
                        Chip select for SD-card
    --enable-sd         With SD-Card (requires FAT)
    --enable-df         With DataFlash (requires UFFS)
    --enable-flash-fs   With Flash FS (requires UFFS)
    --enable-can        With CAN
    --enable-cpp        Use C++
    --enable-task-connless
                        Start demo task: connless
    --enable-task-hk    Start demo task: housekeeping
    --enable-rtc        Enable NanoMind A712C RTC
    --enable-mpio       Enable NanoMind A712D MPIO
    --with-storage      Enable Storage module
    --with-adcs         Enable ADCS module
    --with-cdh          Enable CDH module
```

Todas estas opciones deben ser configuradas previamente. Para ello, se edita un .sh en la carpeta raíz del proyecto y se escriben de forma consecutiva las diferentes opciones de configuración para un proyecto. Véase el siguiente ejemplo:

```
#!/bin/sh
/usr/bin/python waf configure \
    --enable-driver-debug \
    --with-console=0 \
    --with-storage \
    --enable-sd \
    --enable-df \
    --enable-flash-fs \
    --enable-rtc \
    --rom \
    --hostname="nanomind" \
    --model="A712" \
     eclipse
```

Se observa la estructura que debe seguir el archivo de configuración para que sea correctamente leído por el código waf. Hay que tener en cuenta que hay ciertas características de la placa Nanomind que deben ser configuradas en estos archivos si se quiere disponer de ellas. Un ejemplo es el reloj en tiempo real `(--enable-rtc)`, el cual no se encuentra activado por defecto.

7. Instalación OpenOCD

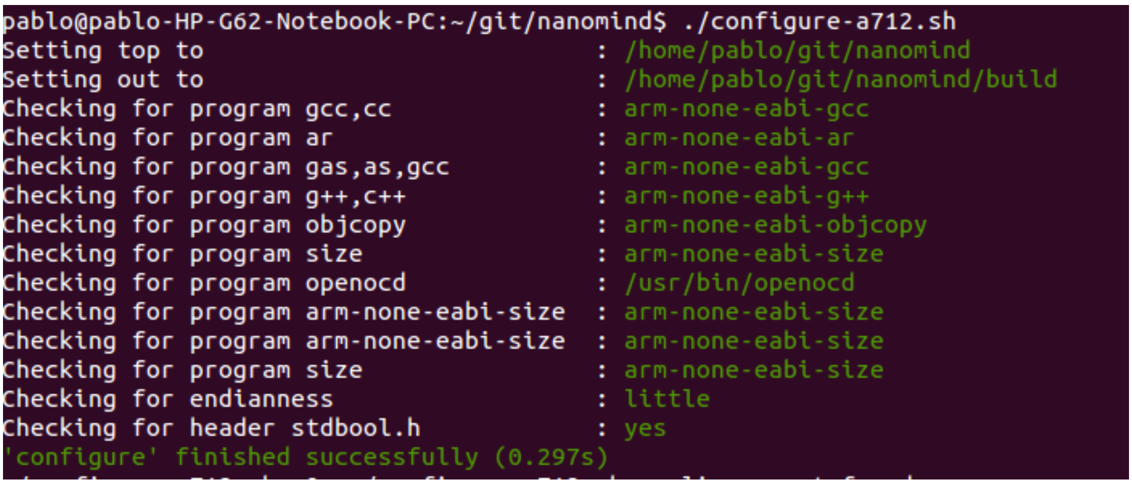
Para el correcto funcionamiento de la carga de los proyectos, es necesario la instalación de OpenOCD (Open On-Chip Debugger). Para esto solo es necesario escribir en el terminal: `sudo apt-get install openocd.`

8. Ejemplo de carga de proyecto

Una vez instalado todo el software anterior ya es posible cargar un proyecto a la placa Nanomind. Existen dos tipos de carga; escribiendo en la memoria flash o en la RAM. En este caso, y para todos los restantes, se empleará el primer método. Si se quiere tener más información sobre la carga de software directamente en la memoria RAM, se encuentra disponible el capítulo 1 de *GS-SW-DOC*.

Proceso de carga:

1. En la carpeta raíz, editar y guardar un archivo .sh con la configuración para la placa Nanomind. Todas las posibilidades se encuentran en el capítulo waf de este documento.
2. En el terminal del equipo, empleamos el comando `cd` hasta llegar a la carpeta en la que se aloja el proyecto. Una vez allí, se introducirá el nombre del archivo de configuración con su extensión y anteponiendo "./"; por ejemplo: `./configure-a712.sh`. Aparecerá la carga de la configuración como en la imagen inferior.



3. Cuando la configuración está cargada no es necesario volver a repetir el proceso a menos que se quiera cambiar la inicial. Para eliminar una configuración o una build en el terminal escribir: `./waf clean`. Para compilar el proyecto, escribir en el terminal: `./waf build`. Las diferentes bibliotecas se irán cargando y compilando, de forma que debe aparecer el resultado inferior:

```
pablo@pablo-HP-G62-Notebook-PC:~/git/nanomind$
pablo@pablo-HP-G62-Notebook-PC:~/git/nanomind$ ./waf build
Waf: Entering directory `/home/pablo/git/nanomind/build'
[  1/103] c: lib/libgomspace/src/dev/ants/cmd_ants.c -> build/lib/libgomspace/src/dev/ants/cmd_ants.c.1.o
[  2/103] c: lib/libgomspace/src/dev/ants/isis_ants.c -> build/lib/libgomspace/src/dev/ants/isis_ants.c.1.o
[  3/103] c: lib/libgomspace/src/dev/pca9665/pca9665.c -> build/lib/libgomspace/src/dev/pca9665/pca9665.c.1.o
[  4/103] c: lib/libgomspace/src/dev-arm/adc/adc.c -> build/lib/libgomspace/src/dev-arm/adc/adc.c.1.o
[  5/103] c: lib/libgomspace/src/dev-arm/aic/libAT91sam7a1.c -> build/lib/libgomspace/src/dev-arm/aic/libAT91sam7a1.c.1.o
[  6/103] c: lib/libgomspace/src/dev-arm/cpu/cpu.c -> build/lib/libgomspace/src/dev-arm/cpu/cpu.c.1.o
[  7/103] c: lib/libgomspace/src/dev-arm/flash/flash.c -> build/lib/libgomspace/src/dev-arm/flash/flash.c.1.o
[  8/103] c: lib/libgomspace/src/dev-arm/flash/flash_at49bv320dt.c -> build/lib/libgomspace/src/dev-arm/flash/flash_at49bv320dt.c.1.o
[  9/103] c: lib/libgomspace/src/dev-arm/gyro/gyro.c -> build/lib/libgomspace/src/dev-arm/gyro/gyro.c.1.o
[ 10/103] c: lib/libgomspace/src/dev-arm/i2c/pca9665_isr.c -> build/lib/libgomspace/src/dev-arm/i2c/pca9665_isr.c.1.o
[ 11/103] c: lib/libgomspace/src/dev-arm/lm70.c -> build/lib/libgomspace/src/dev-arm/lm70.c.1.o
[ 12/103] c: lib/libgomspace/src/dev-arm/magnetometer/hmc5843.c -> build/lib/libgomspace/src/dev-arm/magnetometer/hmc5843.c.1.o
[ 13/103] c: lib/libgomspace/src/dev-arm/max6675.c -> build/lib/libgomspace/src/dev-arm/max6675.c.1.o
[ 14/103] c: lib/libgomspace/src/dev-arm/port/idle.c -> build/lib/libgomspace/src/dev-arm/port/idle.c.1.o
[ 15/103] c: lib/libgomspace/src/dev-arm/port/port.c -> build/lib/libgomspace/src/dev-arm/port/port.c.1.o
[ 16/103] c: lib/libgomspace/src/dev-arm/port/portISR.c -> build/lib/libgomspace/src/dev-arm/port/portISR.c.1.o
[ 17/103] c: lib/libgomspace/src/dev-arm/pwm/pwm.c -> build/lib/libgomspace/src/dev-arm/pwm/pwm.c.1.o
[ 18/103] c: lib/libgomspace/src/dev-arm/rtc/cmd_ds1302.c -> build/lib/libgomspace/src/dev-arm/rtc/cmd_ds1302.c.1.o
../lib/libgomspace/src/dev-arm/pwm/pwm.c: In function 'pwm_set_dir_fromzero':
../lib/libgomspace/src/dev-arm/pwm/pwm.c:250:21: warning: unused variable 'pwm_tmp2' [-Wunused-variable]
../lib/libgomspace/src/dev-arm/pwm/pwm.c:250:11: warning: unused variable 'pwm_tmp1' [-Wunused-variable]
[ 19/103] c: lib/libgomspace/src/dev-arm/rtc/ds1302.c -> build/lib/libgomspace/src/dev-arm/rtc/ds1302.c.1.o
[ 20/103] c: lib/libgomspace/src/dev-arm/spi/spi.c -> build/lib/libgomspace/src/dev-arm/spi/spi.c.1.o
[ 21/103] c: lib/libgomspace/src/dev-arm/syscalls.c -> build/lib/libgomspace/src/dev-arm/syscalls.c.1.o
[ 22/103] c: lib/libgomspace/src/dev-arm/usart/usart.c -> build/lib/libgomspace/src/dev-arm/usart/usart.c.1.o
[ 23/103] c: lib/libgomspace/src/dev-arm/wdt/wdt_at91sam7a1.c -> build/lib/libgomspace/src/dev-arm/wdt/wdt_at91sam7a1.c.1.o
[ 24/103] c: lib/libgomspace/src/freertos/heap_3.c -> build/lib/libgomspace/src/freertos/heap_3.c.1.o
[ 25/103] c: lib/libgomspace/src/freertos/hook_impl.c -> build/lib/libgomspace/src/freertos/hook_impl.c.1.o
[ 26/103] c: lib/libgomspace/src/freertos/list.c -> build/lib/libgomspace/src/freertos/list.c.1.o
[ 27/103] c: lib/libgomspace/src/freertos/queue.c -> build/lib/libgomspace/src/freertos/queue.c.1.o
[ 28/103] c: lib/libgomspace/src/freertos/tasks.c -> build/lib/libgomspace/src/freertos/tasks.c.1.o
[ 29/103] c: lib/libgomspace/src/freertos/timers.c -> build/lib/libgomspace/src/freertos/timers.c.1.o
```

Al final de la compilación no debe aparecer ningún error (color rojo). Los warnings deben ser tenidos en cuenta, aunque generalmente no afectan a la carga del proyecto y algunos son inherentes a las bibliotecas. Véase el ejemplo inferior.

```
[ 99/103] cstlib: build/lib/libcsp/src/arch/freertos/csp_malloc.c.1.o build/lib/libcsp/src/arch/freertos/csp_queue.c.1.o build/lib/libcsp/src/ar
ch/freertos/csp_semaphore.c.1.o build/lib/libcsp/src/arch/freertos/csp_system.c.1.o build/lib/libcsp/src/arch/freertos/csp_thread.c.1.o build/li
b/libcsp/src/arch/freertos/csp_time.c.1.o build/lib/libcsp/src/crypto/csp_hmac.c.1.o build/lib/libcsp/src/crypto/csp_sha1.c.1.o build/lib/libcsp
/src/crypto/csp_xtea.c.1.o build/lib/libcsp/src/csp_buffer.c.1.o build/lib/libcsp/src/csp_conn.c.1.o build/lib/libcsp/src/csp_crc32.c.1.o build/
lib/libcsp/src/csp_debug.c.1.o build/lib/libcsp/src/csp_endian.c.1.o build/lib/libcsp/src/csp_io.c.1.o build/lib/libcsp/src/csp_port.c.1.o build
/lib/libcsp/src/csp_route.c.1.o build/lib/libcsp/src/csp_service_handler.c.1.o build/lib/libcsp/src/csp_services.c.1.o build/lib/libcsp/src/inte
rfaces/csp_if_i2c.c.1.o build/lib/libcsp/src/interfaces/csp_if_kiss.c.1.o build/lib/libcsp/src/interfaces/csp_if_lo.c.1.o build/lib/libcsp/src/t
ransport/csp_rdp.c.1.o build/lib/libcsp/src/transport/csp_udp.c.1.o -> build/lib/libcsp/libcsp.a
[100/103] cstlib: build/lib/libio/src/extras/csp_console.c.2.o build/lib/libio/src/ftp/backend_ram.c.2.o build/lib/libio/src/ftp/ftp_server.c.2.
o build/lib/libio/src/io/cmd_cam.c.2.o build/lib/libio/src/io/cmd_com.c.2.o build/lib/libio/src/io/cmd_eps.c.2.o build/lib/libio/src/io/cmd_obc.
c.2.o build/lib/libio/src/io/nanocam.c.2.o build/lib/libio/src/io/nanocom.c.2.o build/lib/libio/src/io/nanomind.c.2.o build/lib/libio/src/io/nan
opower.c.2.o -> build/lib/libio/libio.a
[101/103] cxxprogram: build/src/boot.c.2.o build/src/cmd_panels.c.2.o build/src/cmd_periph.c.2.o build/src/crt.s.2.o build/src/main.c.2.o build/
src/mpio.c.2.o build/src/task_hk_collector.c.2.o build/src/task_init.c.2.o build/src/task_server.c.2.o build/src/task_server_connless.c.2.o buil
d/src/test_cpp.cpp.2.o build/src/test_rpcx.c.2.o -> build/nanomind.elf
[102/103] objcopy: build/nanomind.elf -> build/nanomind.bin
[103/103] size: build/nanomind.elf
   text    data     bss     dec     hex filename
 211688   56748    4132  272568   428b8 nanomind.elf
Waf: Leaving directory `/home/pablo/git/nanomind/build'
'build' finished successfully (9.490s)
```

4. Por último, para cargar el proyecto compilado al Nanomind solo es necesario escribir en el terminal: `./waf program`. Para este paso hay que tener en cuenta que la configuración física de la placa con el equipo debe ser la correcta, tal y como se refleja en la siguiente figura:

```
pablo@pablo-HP-G62-Notebook-PC:~/git/flash_pl_2$ ./waf program
Waf: Entering directory `/home/pablo/git/flash_pl_2/build'
[1/1] openocd: build/nanomind.elf
Open On-Chip Debugger 0.7.0 (2013-10-22-08:31)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.sourceforge.net/doc/doxygen/bugs.html
Info : only one transport option; autoselect 'jtag'
srst_only separate srst_gates_jtag srst_open_drain connect_deassert_srst
adapter_nsrst_delay: 250
jtag_ntrst_delay: 250
DEPRECATED! use 'adapter_khz' not 'jtag_khz'
adapter speed: 6000 kHz
fast memory access is enabled
dcc downloads are enabled
Info : clock speed 6000 kHz
Info : JTAG tap: at91sam7a1.cpu tap/device found: 0x1f0f0f0f (mfg: 0x787, part:
0xf0f0, ver: 0x1)
Info : Embedded ICE version 1
Info : at91sam7a1.cpu: hardware has 2 breakpoint/watchpoint units
target state: halted
target halted in ARM state due to debug-request, current mode: System
cpsr: 0x6000001f pc: 0x40032178
Clear flash status
Unlocking flash
Disabling watchdog timer
Info : Flash Manufacturer/Device: 0x001f 0x90c4
Warn : expected one protection register field, but found 255
erased sectors 0 through 25 on flash bank 0 in 10.112150s
wrote 317048 bytes from file ../build/nanomind.bin to flash bank 0 at offset 0x
0000000 in 6.398227s (48.391 KiB/s)
Info : JTAG tap: at91sam7a1.cpu tap/device found: 0x1f0f0f0f (mfg: 0x787, part:
0xf0f0, ver: 0x1)
shutdown command invoked
Waf: Leaving directory `/home/pablo/git/flash_pl_2/build'
'program' finished successfully (18.340s)
pablo@pablo-HP-G62-Notebook-PC:~/git/flash_pl_2$
```

9. Instalación minicom

Para poder recibir información de la placa Nanomind a través del puerto USB es necesario la instalación del programa minicom. Para esto solo es necesario escribir en el terminal: `sudo apt-get install minicom`.

En cuanto a la configuración, se adjuntan varias capturas de pantalla con la configuración adecuada para que sea posible recibir y enviar información con la plataforma.

```
+----[Parámetros de Comunicación]----+
|                                    |
| SpeeCurrent: 500000 8N1      Data  |
| A: <next>        L: None     S: 5  |
| B: <prev>        M: Even     T: 6  |
| C:    9600       N: Odd      U: 7  |
| D:   38400       O: Mark     V: 8  |
| E: 115200        P: Space          |
|                                    |
| Stopbits                           |
| W: 1             Q: 8-N-1          |
| X: 2             R: 7-E-1          |
|                                    |
|                                    |
| Elija la opción  o (Enter) para sal|
|ir:                                 |
+------------------------------------+
```

```
+------------[Configuración]-------------+
| Nombres de archivos y rutas            |
| Protocolos de transferencia de archivos |
| Configuración de la puerta serial      |
| Modem y marcado de número              |
| Pantalla y teclado                     |
| Salvar configuración como dfl          |
| Salvar configuración como..            |
| Salir                                  |
+----------------------------------------+
```

```
+-----------------------------------------------------------------------+
| A - Dispositivo Serial            : /dev/ttyUSB0                      |
| B - Localización del Archivo de Bloqueo : /var/lock                   |
| C - Programa de Acceso           :                                    |
| D - Programa de Salida             :                                  |
| E - Bps/Paridad/Bits             : 500000 8N1                         |
| F - Control de Flujo por Hardware: No                                 |
| G - Control de Flujo por Software: No                                 |
|                                                                       |
|    ¿Qué configuración alterar?                                        |
+-----------------------------------------------------------------------+
```

CEPHEUS FM Platform - Quick Start Guide

ISIS.CEPHEUS.UM.001

Issue 1.0

Release information

Written by:	L. Monas	2014-07-24
Checked by:	R. Barnhoorn, J. Elstak	2014-07-24
Approved by:	R. Fernandez	2014-07-24

Distribution List:

Change log

Date	Issue	Modified by	Section / Pages Affected	Reason for Change
15-04-14	Draft 1.0	LMON	All	Document created
16-04-14	Draft 1.1	RBAR, JELS	All	Document reviewed
24-07-2014	1.0	RFER	All	First Release

Applicable Documents

#	Document Identifier	Document Title	Version
AD01	GS-DS-BP4-series-1.5	NanoPower BP4 series Datasheet	24 Oct 2011
AD02	gs-ds-nanopower-p31u-8.0	NanoPower P-series Datasheet P31uS V8.0	30 Sep 2013
AD03	GS-DS-NM712C-1.1	NanoMind A712C Datasheet	2 May 2013
AD04	ISIS.ANTS.UM.001	Antenna system User Manual	22 Feb 2012
AD05	ISIS.CGSE.UM.002	ISIS RF Checkout box	26 July 2012
AD06	ISIS.IGIS.UM.001	ISIS Generic Interface User Manual	20 Nov 2011
AD07	ISIS.IMTQ.UM.001	Magnetotorquer Board User Manual	01 Dec 2011
AD08	ISIS.iSPA_side.UM.001	ISIS Solar Panel-Side Mount User Manual	17 Dec 2012
AD09	ISIS.MGSE.0.1.022	MGSE 3U User Manual	22 July 2013
AD10	ISIS.STSX.03U.UM	STS-X 3U User Manual	29 July 2013
AD11	ISIS.TRXUV.UM.001	ISIS VHF/UHF Transceiver	16-03-2012

Reference Documents

#	Document Identifier	Document Title	Version
RD01			
RD02			
RD03			
RD04			
RD05			

TBD/TBC/TBW

TBD/TBC/TBW	Responsible	Action	Page

Disclaimer

The contents of this document are subject to the relevant provisions in the contract concluded between the parties. ISIS – Innovative Solutions In Space B.V. ("ISIS") shall not be liable, in full or in part, for any damage arising out from the application or use of any product or circuit described herein, in case such application or use are carried out in a manner not in line with the instructions and warranties provided in the User Manual, Safety Manual, product information sheets or any other document provided by ISIS upon the delivery of the product ("Documents"). Further, ISIS shall not be liable for any damage caused by any use which exceeds the function(s) of the product, or does not conform to such function(s) as described in the Documents. ISIS shall not be liable for any damage arising from a use which is not carried out in a manner conforming to acceptable practices in the aerospace industry.

ISIS warrants that the product is supplied after relevant tests had shown the product is in good order and functioning, as far as these tests may indicate and predict product functionality.

List of Acronyms

ABF	Apply Before Flight
AC/DC	Alternating Current/Direct Current
EPS-EGSE	Electronic Power Supply Electronic Ground Support Equipment
IGIS	ISIS Generic Interface System
JTAG	Joint Test Action Group
LED	Light Emitting Diode
OBC	OnBoard Computer
PSU	Power supply
SW	Switch (in document context)
USB	Universal Serial Bus

Table of Contents

Table of Figures

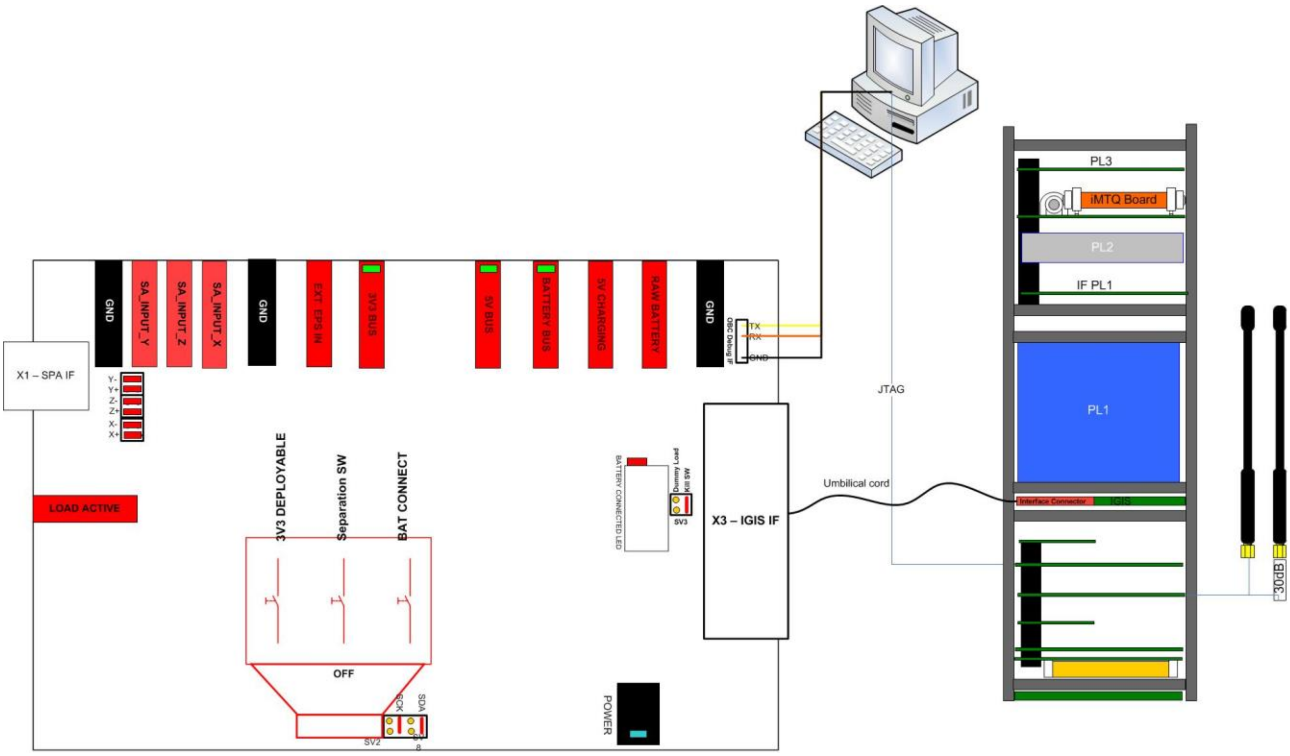


Table of Tables

No table of figures entries found.

1 Handling and Storage

1.1 Handling

ELECTROSTATIC SENSITIVE DEVICE

DO NOT HANDLE PLATFORM WITHOUT OBSERVING ESD HANDLING PRECAUTIONS

- Wear grounded wristband at all times
- Ensure platform is grounded
- Use anti-static mats for your workstation

1.2 Storage

Disconnect the platform from the EPS-EGSE (Umbilical) and store it a closed ESD Bag.

2 Quick start

Please use this document as the quick start reference on safely operating the ISIS platform. The following subjects will be discussed throughout this manual:

- Operating the platform from an External Power Source using the EPS-EGSE
- Operating the platform from the platform battery using the EPS-EGSE
- Charging the platform battery using the EPS-EGSE
- Operating the platform from the platform battery
- Connecting to the OBC using the EPS-EGSE
- Programming the OBC using JTAG connection

It is recommended to monitor the battery voltage at all times while operating the platform. The set-up to monitor the battery voltage is depicted in Figure 2-1. When the platform is operated as if it would be in flight it is advised to monitor the voltage prior to performing a test and after performing a test. In case the battery voltage reaches a voltage level of 14.0V it is recommended to charge the battery up to maximum 15.5V.

Note that when the battery voltage reaches a voltage level of 12.0V the EPS will automatically switch all of its outputs off (this could potentially be harmful to the system. Therefore it is strongly advised to charge the battery as soon as the voltage level reaches the level of 14.0V).

In order to be able to operate the platform it needs to be powered by either an external power source or the platform batteries and one of the separation switches must be activated. The platform has 3 separation switches which are redundant. This means that when one of the switches is activated the system is operational (assuming that the battery is connected). Two of the separation switches are integrated in the structure. The third separation switch is included in the umbilical harness, which allows the separation switch to be controlled from the EPS-EGSE.

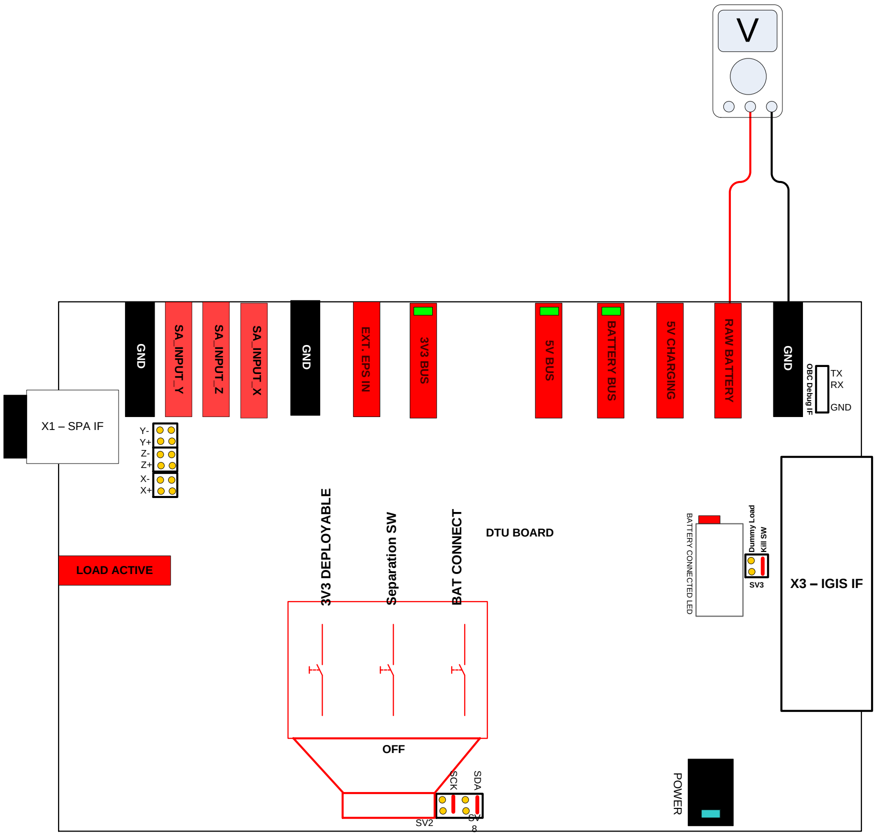


Figure 2-1 Platform battery voltage monitoring set-up

2.1 Operating the platform from an External Power Source using the EPS-EGSE

When operating the platform from an external power source using the EPS-EGSE, the steps described below need to be followed in the exact sequence described.

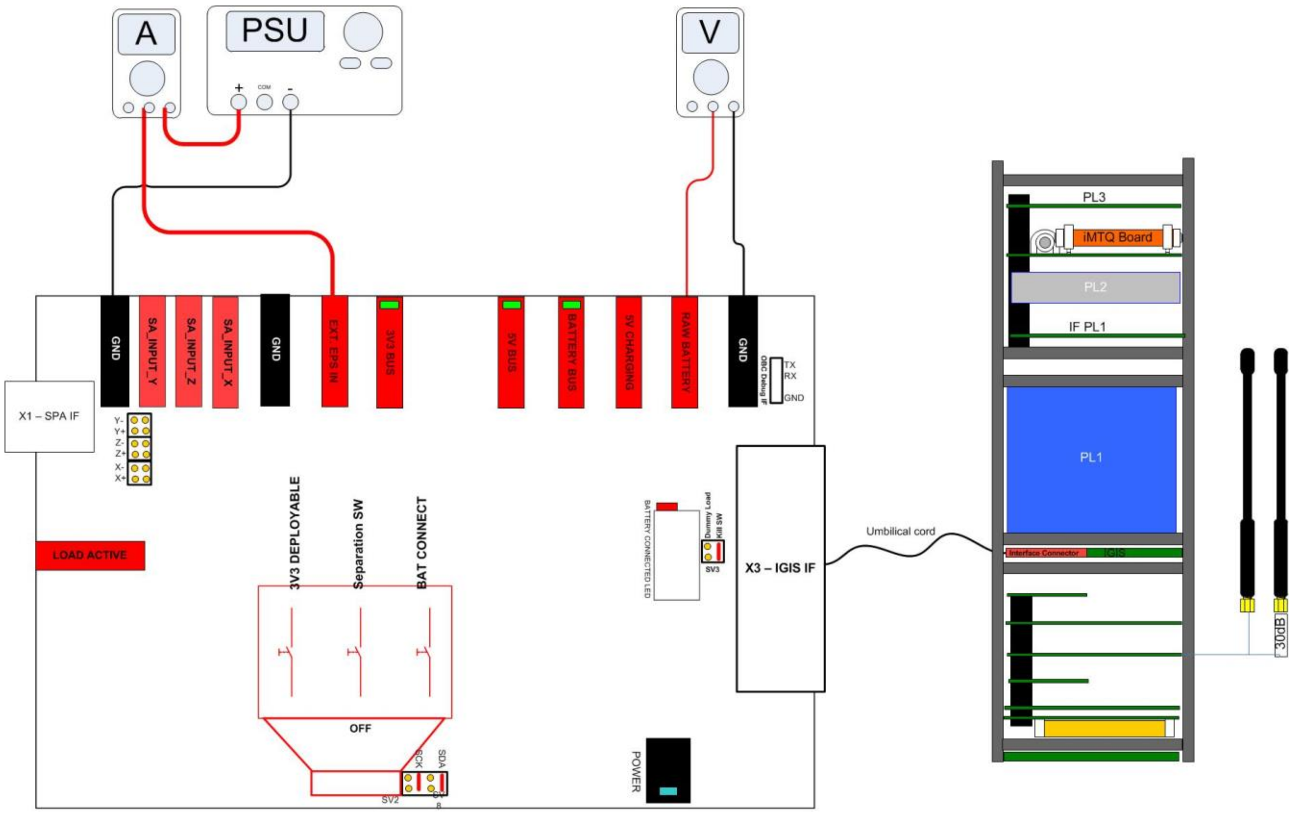


Figure 2-2: Operating the platform from an external power source using the EPS-EGSE

WARNING: When using an external power supply as the power source for the platform, the battery must be disconnected.

EPS-EGSE BATTERY CONNECT switch must be in OFF position.

2.1.1 Connecting the external power source

1) Ensure that the EPS-EGSE is powered using the supplied AC/DC adapter
2) Connect the umbilical cord. Connect the D37 connector on the EPS-EGSE and connect the Omnetics connector to the IGIS 37 Omnetics connector.
3) Ensure that the external power source is set to 14.9V with the current limit set to <1.0A.
4) Ensure prior to connecting the external power source to the EPS-EGSE that the output of the external power source is disabled.
5) Connect the external power source return/ground terminal to one of the GND connectors (black banana socket) of the EPS-EGSE.
6) Ensure that the platform battery is disconnected by checking the EPS-EGSE BATTERY CONNECT switch position (must be in OFF position) and the status of the red LED (should be OFF).
7) Connect external power source positive terminal (+14.9V) to the EPS-EGSE EPS IN connector.

8) Enable the output of the external power source.
9) Connect the Separation Switch using the switch available in the EPS-EGSE (SEPARATION SW to ON position) and verify that the status LEDs (green LEDs) for the 3.3V bus, 5V bus and Bat bus turn on.
10) Connect the 3.3V Deployable switch using the switch available in the EPS-EGSE (3V3 DEPLOYABLE SW to ON position).
NOTE: If this switch is not set to the ON position the antennas will not be powered.

2.1.2 Disconnecting the external power source

1) Disconnect the 3.3V Deployable switch using the switch available in the EPS-EGSE (3V3 DEPLOYABLE SW to OFF position).
2) Disconnect the Separation Switch using the switch available in the EPS-EGSE (SEPARATION SW to OFF position) and verify that all status LEDs for the 3.3V bus, 5V bus and bat bus come off.
3) Disable the output of the external power source.
4) Disconnect the external power source from the EPS-EGSE EPS IN connector.

2.2 Operating the platform from the platform battery using the EPS-EGSE

When operating the platform from platform battery using the EPS-EGSE, the steps described below need to be followed in the exact sequence described.

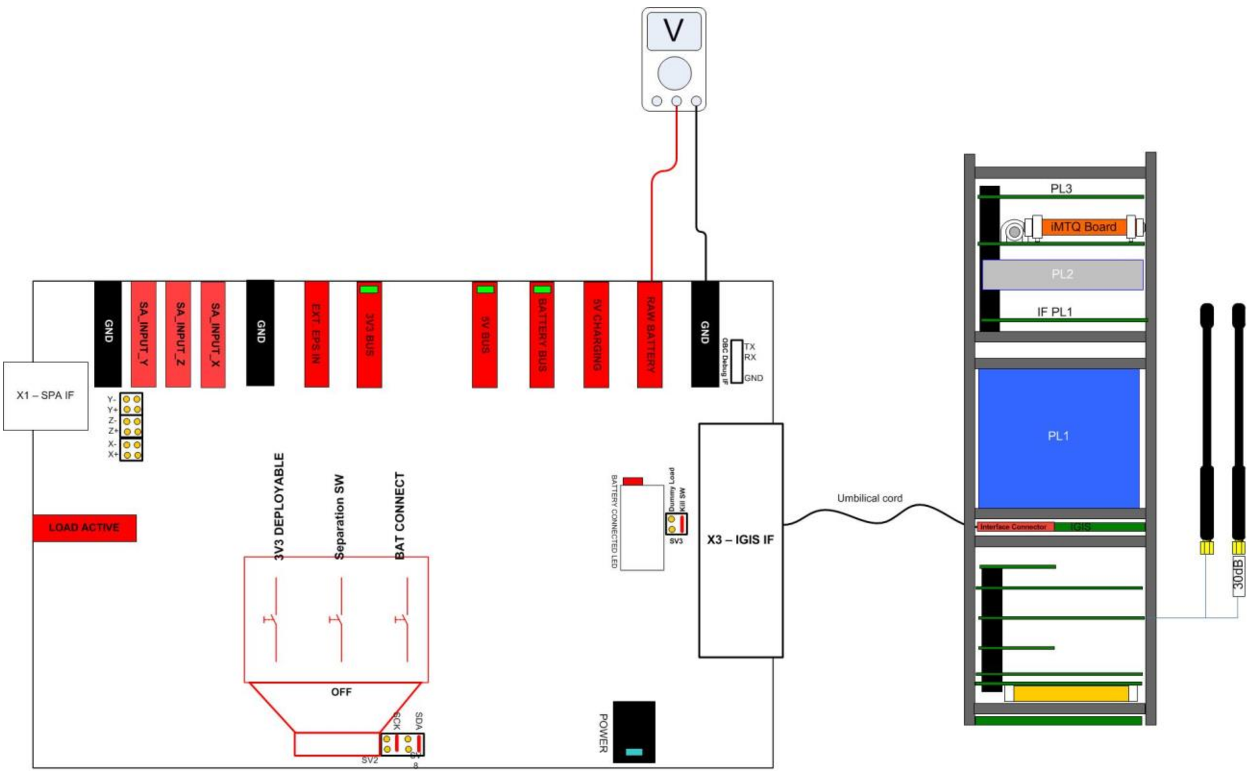


Figure 2-3 Operating the platform from the platform battery using the EPS-EGSE

2.2.1 Switching On

1) Ensure that the EPS-EGSE is powered using the supplied AC/DC adapter
2) Connect the umbilical cord. Connect the D37 connector on the EPSE-EGSE and connect the omnetics connector to the IGIS 37 omnetics connector
3) Connect the Battery Connect Switch using the switch available on the EPS-EGSE (BAT CONNECT SW to ON position) and verify that the status LED (red LED) turns on
4) Connect the Separation Switch using the switch available in the EPS-EGSE (SEPARATION SW to ON position) and verify that the status LEDs (green LEDs) for the 3.3V bus, 5V bus and Bat bus turn on

2.2.2 Switching Off

1) Disconnect the Separation Switch using the switch available in the EPS-EGSE (SEPARATION SW to OFF position) and verify that the status LEDs (green LEDs) for the 3.3V bus, 5V bus and Bat bus turn off
2) Disconnect the Battery Connect Switch using the switch available in the EPS-EGSE (BAT CONNECT SW to OFF position) and verify that status LED (red LED) comes off

2.3 Charging the platform battery using the EPS-EGSE

When the battery voltage drops below 14.0V it is recommended to charge the battery. This needs to be done using the EPS-EGSE together with an external power source to provide the charge voltage.

The steps described below need to be followed in the exact sequence described. It is recommended to monitor the battery voltage level during the charging of the platform battery.

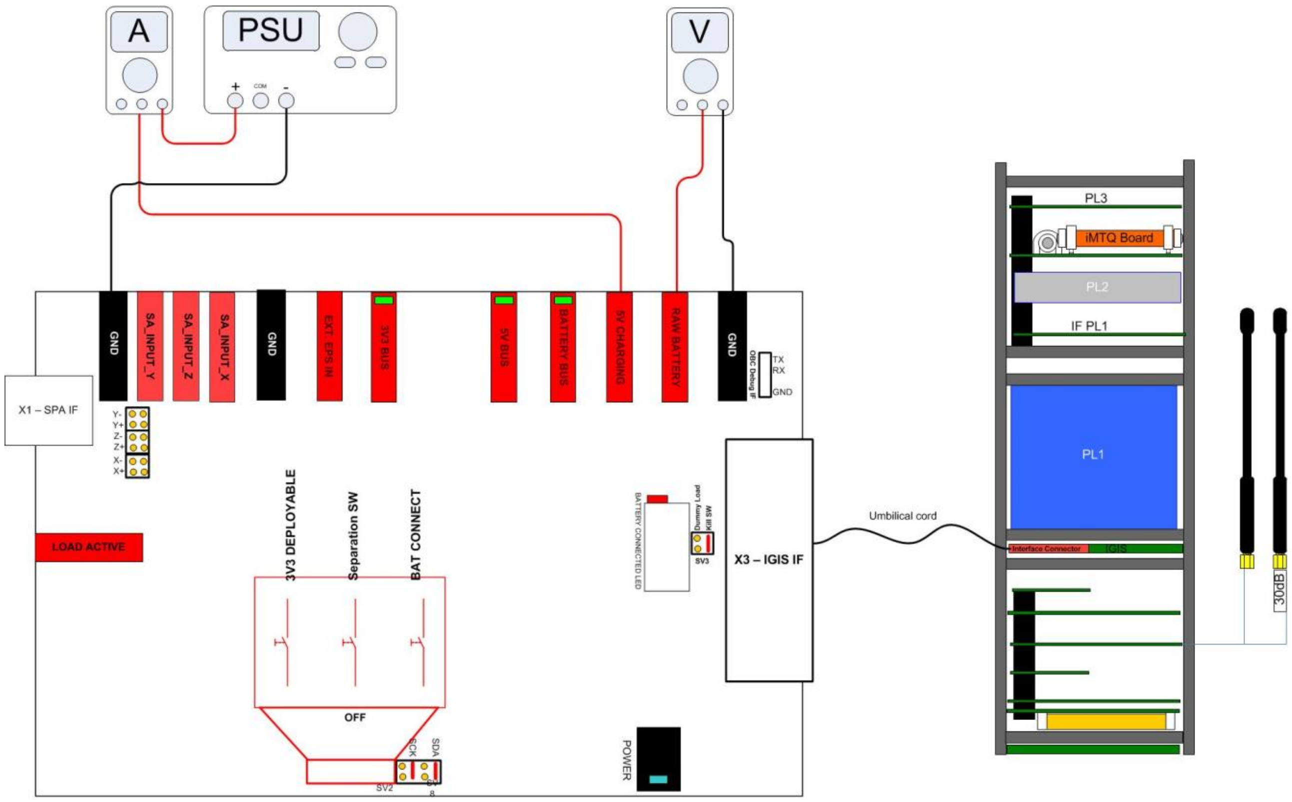


Figure 2-4 Charging the platform battery using the EPS-EGSE

WARNING: Do not connect a higher voltage than +5V to the 5V_CHARGING interface as it could incur in battery permanent damage or performance degradation. Also take special care not to connect the +5V to any of the outputs of the EPS-EGSE.

2.3.1 Connecting the external power source for charging

1) Ensure that the EPS-EGSE is powered using the supplied AC/DC adapter
2) Connect the umbilical cord. Connect the D37 connector on the EPSE-EGSE and connect the Omnetics connector to the IGIS 37 Omnetics connector
3) Ensure that the external power source is set to 5.0V with the current limit set to <1.0A
4) Ensure prior to connecting the external power source to the EPS-EGSE that the output of the external power source is disabled
5) Connect the external power source return/ground terminal to one of the GND connectors (black banana socket) of the EPS-EGSE
6) Connect the external power source positive terminal (+5.0V) to the EPS-EGSE 5V CHARGING connector
7) Connect the Battery Connect Switch using the switch available in the EPS-EGSE (BAT CONNECT SW to ON position) and verify that the status LED (red LED) comes on.

8) Enable the output of the external power source

When the battery voltage has reached a level of 15.5V the battery is sufficiently charged and the charging can be stopped.

2.3.2 Disconnecting the external power source after charging

1) Disable the output of the external power source
2) Disconnect the external power source positive terminal (+5.0V) from the EPS-EGSE 5V CHARGING connector
3) Disconnect the Battery Connect Switch using the switch available in the EPS-EGSE (BAT CONNECT SW to OFF position) and verify that the status LED (red LED) comes off

2.4 Operating the platform from the platform battery

The last option on powering the platform is when the platform is configured as if it is flight prepped. This means that the Apply Before Fly (ABF) connector is connected to the IGIS and that the separation switches in the platform structure are operational.

In the ABF connector the battery is connected. The platform will become operational as soon as one of the separation switches is activated.

2.5 Connecting to the OBC using the EPS-EGSE

In order to be able to program the OBC with user developed software the following connections are required and the following steps need to be followed.

Note, as two methods of powering the platform using the EPS-EGSE are described, it is up to the user to choose the method that will fit the needs.

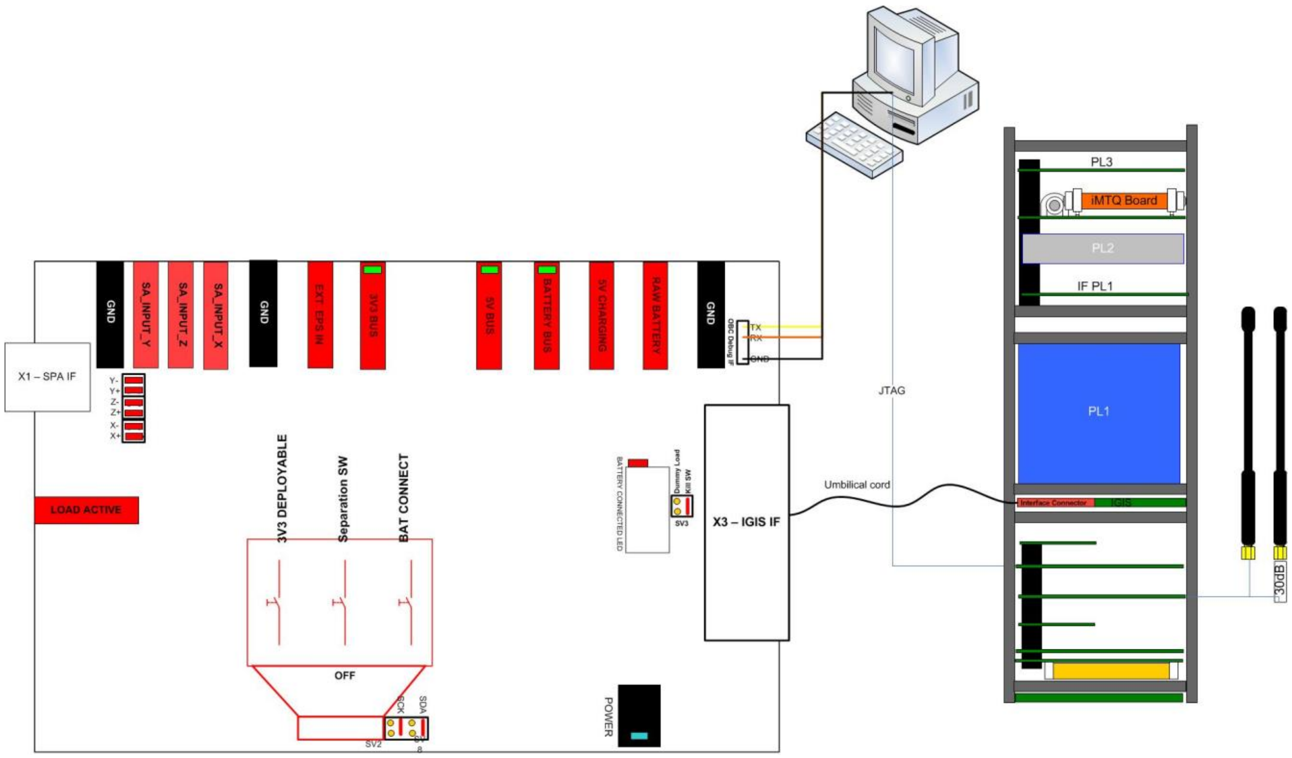


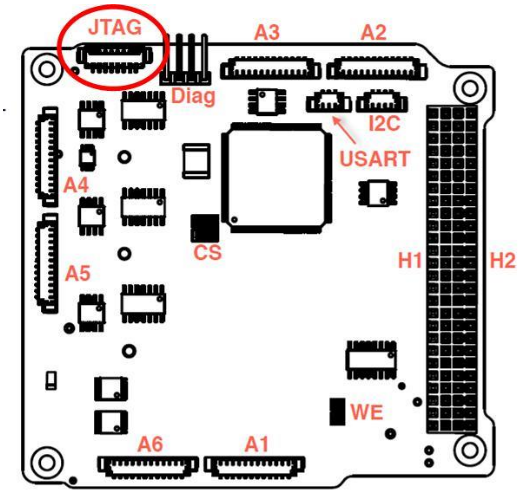
Figure 2-5 Connecting to the OBC using the EPS-EGSE

WARNING: Ensure prior to connecting the USB to Serial cable that the VCC line in the header is removed from the connector. If this is not done, the OBC will be powered from two different sources which will damage the OBC.

1) Connect the USB to serial cable to the test computer
2) Connect the USB to serial cable to the EPS-EGSE serial interface header.
3) On the test computer open a program that allows connecting through the serial port to the OBC software. The factory default baud rate setting is a speed of 500000 baud.

2.5.1 Programming the OBC using JTAG connection

1) Connect the JTAG hardware to the JTAG programming and debugging interface on the OBC (indicated in Figure 2-6)
2) Connect the USB to serial cable as described in the previous section and start serial terminal.
3) On the test computer use software that allows to connect through the serial port to the JTAG programming and debugging interface (not delivered by ISIS).
4) Power the platform as described previous sections.



Pinout for JTAG:

1. TDO
2. TCK
3. TMS
4. TDI
5. RST (not used)
6. RST (not used)
7. VCC
8. GND

Figure 2-6: JTAG programming and debugging interface location

Checkout Box User Manual

ISIS.CB.UM.001

Issue 2.0

Release information

Written by:	S. Speretta	2014-07-22
Checked by:	R. Fernandez	2014-07-23
Approved by:	E. Celton	2014-07-23

Distribution List:

Change log

Date	Issue	Modified by	Section / Pages Affected	Reason for Change
2014-07-22	2.0	S. Speretta	All	Manual general review
2012-12-20	1.8.3	S. Speretta	§2	Noise figure and output power specs updated
2012-11-18	1.8.2	S. Speretta	All	Minor layout changes
2012-09-17	1.8.1	S. Speretta	All	Updated receiver section
2012-06-07	1.8	S. Speretta	All	Added FFT window and Log
2012-05-22	1.7	S. Speretta	All	Updated layout
2011-06-17	1.6	S. Speretta	§6 - §10	Updated Linux installation guide
2011-05-30	1.5	S. Speretta	§5	Updated SDR description
2011-03-30	1.4	S. Speretta	§15 - §17	Added hardware description
2011-03-29	1.3	S. Speretta	§12 - §14	Added transmitter operations
2011-03-07	1.2	S. Speretta	§11	Modified receiver settings
2011-05-30	1.1	S. Speretta	All	Added comments on TCP interface
2011-02-09	1.0	S. Speretta	All	Initial version

Applicable Documents

AD01			
AD02			

Reference Documents

RD01			
RD02			
RD03			

TBD/TBC/TBW

TBD/TBC/TBW	Responsible	Action	Page

Disclaimer

The contents of this document are subject to the relevant provisions in the contract concluded between the parties. ISIS – Innovative Solutions In Space B.V. ("ISIS") shall not be liable, in full or in part, for any damage arising out from the application or use of any product or circuit described herein, in case such application or use are carried out in a manner not in line with the instructions and warranties provided in the User Manual, Safety Manual, product information sheets or any other document provided by ISIS upon the delivery of the product ("Documents"). Further, ISIS shall not be liable for any damage caused by any use which exceeds the function(s) of the product, or does not conform to such function(s) as described in the Documents. ISIS shall not be liable for any damage arising from a use which is not carried out in a manner conforming to acceptable practices in the aerospace industry.

ISIS warrants that the product is supplied after relevant tests had shown the product is in good order and functioning, as far as these tests may indicate and predict product functionality.

List of Acronyms

ADC	Analog to Digital Converter
AFSK	Audio Frequency-Shift Keying
AX.25	Amateur X.25
BPSK	Binary Phase Shift Keying
DAC	Digital to Analog Converter
DSP	Digital Signal Processing
ESD	Electro Static Discharge
FCS	Frame Check Sequence
FSK	Frequency-Shift Keying
GPS	Global Positioning System
GUI	Graphical User Interface
KISS	Keep It Simple and Stupid
OFDM	Orthogonal Frequency Division Multiplexing
OS	Operating System
PPS	Pulse Per Second
PSK	Phase Shift Keying
QOS	Quality Of Service
RH	Relative Humidity
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
SDR	Software Defined Radio
TNC	Terminal Node Controller
UHF	Ultra High frequency (400-470MHz within this context)
USB	Universal Serial Bus
VHF	Very High frequency (136-170MHz within this context)
WEEE	Waste Electrical and Electronic Equipment

Table of Contents

Table of Figures

Table of Tables

1 Introduction

This document contains the user manual for the VHF - UHF Transceiver Ground Support Equipment (GSE), also referred to as RF Checkout Box.

The VHF - UHF Transceiver GSE is designed specifically for the reception on ground of a satellite downlink. Radio signals demodulation is performed in software using DSP techniques: the receiver electrical interface was designed to comply with USB specifications and to allow a fast data transfer between the receiver and a PC, which is used to handle modulation and demodulation.

The receiver uses a Binary Phase Shift Keying (BPSK) modulation scheme with Raised Cosine bit shaping: this modulation was selected because, of all common available modulation schemes, it yields best performance in terms of Bit Error Rate vs. signal to noise ratio. It should be noted that, thanks to its noise level tolerance, BPSK can achieve the same performances as a synchronous FSK demodulator with a 3 dB lower noise margin. Furthermore, the modulation scheme and the bit shaping allow using a limited bandwidth for transmission, actually enabling a narrower bandwidth receiver (which has then a lower noise floor).

The transmitter supports several modulation schemes: AFSK and FSK with G3RUH scrambling were selected for their simplicity and for the implicit robustness of the satellite receiver while BPSK can be used to perform loopback tests to verify the functionality of the system.

The transmitter and receiver are compliant with the whole ISIS receiver / transceiver family of products.

1.1 Document Configuration Management

This document is valid until it is declared obsolete or replaced with a succeeding version. Changes with respect to the previous version will be clear from the revision. As this document may be updated without prior notice, it is advised to check the ISIS website “www.isispace.nl” or ask us for the latest version at “support@isispace.nl” before using this document as reference.

1.2 Shipment contents

Please check the contents of the shipment case for completeness and damage during shipment. The shipment should typically contain the following items:

Item	Quantity	Check
ISIS RF Checkout Box	1	
USB cable	1	
Monopole antenna	2	
90° SMA adapter	2	
Software package (digital copy)	1	
Unit documentation (digital copy)	1	

Table 1-1 Shipment Content Check-list

1.3 Specification

Parameter	Typical Value	Comments
Overall Specifications		
Operational temperature range	0°C to +60°C	
Storage temperature range	-45°C to +85°C (RH<60%)	
Mass	1 kg	
Size	160x170x30 mm (excluding antennas)	
Transmitter		
Frequency range	70 MHz – 6 GHz	
Output power	5 dBm (3 mW)	
Modulation scheme	AFSK, FSK-G3RUH, BPSK	
Data Rate	1200 bps 2400 bps 4800 bps 9600 bps	
Protocol	AX.25	
Receiver		
Frequency range	70 MHz – 6 GHz	
Noise Figure	15 dB	
Recommended minimum input signal level	-80 dBm	
Maximum input signal	0 dBm (1 mW)	
Modulation scheme	BPSK	
Data Rate	1200 bps 2400 bps 4800 bps 9600 bps	
Protocol	AX.25	
Antennas		
Gain	0 dBi	
Radiation pattern	λ / 4 monopole	

Table 1-2 RF Checkout Box Overall Specification

1.4 Requirements

Parameter	Typical Value	Comments
CPU	Intel/AMD Dual Core PC at least 1.6 GHz	
RAM	At least 512 MB	
Connectivity	1 x USB 2.0	
Operating system	Windows XP SP2 / SP3, 32- or 64-bits Windows 7, 32- or 64-bits Linux with at least kernel version 2.6.32, 32-bits (Preferred)	
Software	SUN Java JRE version 1.6 or higher 32- or 64-bits depending on OS SUN Java JDK version 1.6 or higher 32- or 64-bits depending on OS (Preferred)	

Table 1-3 RF Checkout Box system requirements

1.5 Radio Spectrum Licensing

The RF Checkout Box is capable of generating RF signals in the VHF and UHF amateur radio frequency bands. It is also capable of receiving in these bands. Please make sure that all applicable laws in the country are met, and that the user obtains the proper license(s) if needed. Also depending on the country, listening to certain stations may be restricted.

The customer is responsible for acquiring all necessary licenses and the license fees are at the customer's expense.

2 Handling, Storage and Disposal

2.1 Handling

2.1.1 Electrostatic discharge

CAUTION

Note that the Checkout Box is sensitive to Electro Static Discharge (ESD).

The printed circuit boards contained in the Checkout Box can be damaged by electrostatic discharge.

Do not touch any of the boards unless it is absolutely necessary.

If they must be handled, please wear a grounded wrist strap and take anti-static precautions.

Do not connect / disconnect the Antenna port without proper ESD control.

2.1.2 Receiver Input power

WARNING

An input signal higher than the maximum specified input level (**0 dBm**) on the receiver antenna port **will damage the device**. In situations in which the maximum power limitation may be exceeded, please add a protection to the receiver input port. An attenuator (like Minicircuits VAT-30W2+) or a power limiter (like Minicircuits ZFLM-252-1WL+) may help in solving the eventual issue.

Do **NOT** connect the TX and RX ports with a cable: this will damage the receiver input.

Do **NOT** connect the RF Checkout Box directly to the satellite receiver and transmitter ports because this will damage **both** the satellite and the RF Checkout Box.

2.1.3 Exposed Voltages

WARNING

Handling the system with an active power supply connection is not recommended.

The system itself could be damaged and there is a possibility of electric shock hazard. In the event of emergency, disconnect the power supply and proceed, if required, with first aid activities.

2.1.4 Operation Conditions

CAUTION

This system is intended for INDOOR use only and should not be operated outdoors as the electronics may be damaged. Ensure that the system is always operated within its qualification temperature range: 0° C to +60° C.

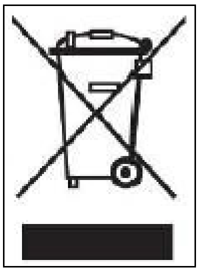
2.2 Storage

CAUTION

The absolute maximum ratings for storage temperature are from -45°C to +85°C with a Relative Humidity < 60%.

2.3 Disposal

WARNING



This product contains materials that can be harmful for the Environment and as such it should not be disposed of with conventional waste but treated according to WEEE regulations (EU Directives 2002/96/EC and further amendments) and brought to an appropriate recycling facility.

3 Functional Description

The RF Checkout Box is composed of two main parts: a radio transceiver (down- and up- converter) and a software application running on a PC. The software will require to be installed on the target machine to allow data reception and transmission. The next two sections will provide a detailed installation guide for users.

Please see Figure 3-1 for the RF Checkout Box block diagram.

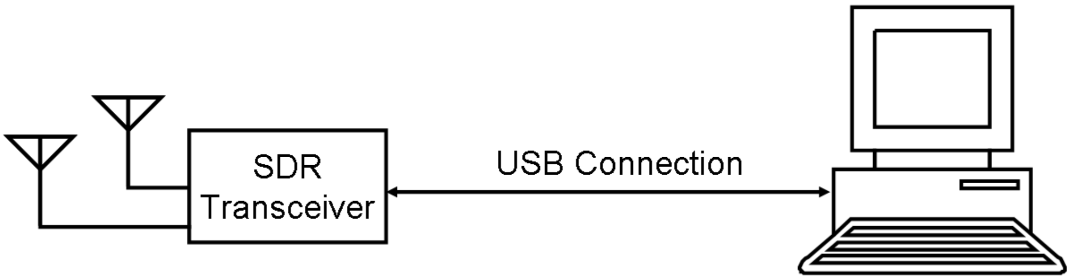


Figure 3-1 RF Checkout Box hardware block diagram

Figure 3-2 shows the RF Checkout Box assembled with the two provided antennas and 90° elbows connected to the RX and TX ports and the USB cable that should be connected to the computer.

Figure 3-2 RF Checkout Box connections

Power to the device is provided by the USB bus, but in case of problems an external power supply can be connected to the power jack (recommended voltage is 6V – 15V) as shown in Figure 3-2. On the back panel, together with the USB and power connections, there are connectors for an external frequency reference (GPS, 10 MHz and PPS in the previous figure) but they are **NOT** currently supported by the software. Please contact us for details.

3.1 Radio

The RF Checkout Box uses a software defined radio that allows simultaneous transmission and reception operations (receiving and transmitting frequency range 70MHz – 6000MHz). The output power is limited to 5 dBm to limit interferences with other systems. Two independent antennas connections are provided for full-duplex operations. The receiver noise figure is 15 dB, which is enough to perform checkout operations on the satellite from a short distance.

All modulation and demodulation operations are performed digitally on a computer, guaranteeing extreme flexibility to the system.

Typically the RF Checkout Box should be kept in short proximity with the satellite (typically 2 - 10 meters) to guarantee that the receiver port is not overloaded and that the signal is strong enough to be received.

It is strongly recommended **NOT** to connect the RF Checkout Box directly to the satellite receiver / transmitter ports with a cable because the power transmitted from the RF Checkout Box would easily damage the satellite receiver and the other way around.

4 Installation

The following sections explain in detail the RF Checkout Box software installation procedure.

The software can successfully run on Windows (XP and 7) or Debian-based Linux systems (other distributions can be supported upon request).

4.1 Windows

The RF Checkout Box requires a JAVA Virtual Machine version 1.7 or higher installed: the setup script will automatically detect its presence and notify the user if it was not found and the installation process will be aborted.

To install the software, first locate the RF Checkout Box setup application on the provided digital media and start it: the application will guide the user through the entire installation process, checking the required dependencies and installing all the necessary software, including the drivers.

Please follow the instructions on the screen and refer to Figure 4-1 and Figure 4-2 for further details. The installer will try to auto-detect the required version of the transport library for the system (32- or 64-bit).

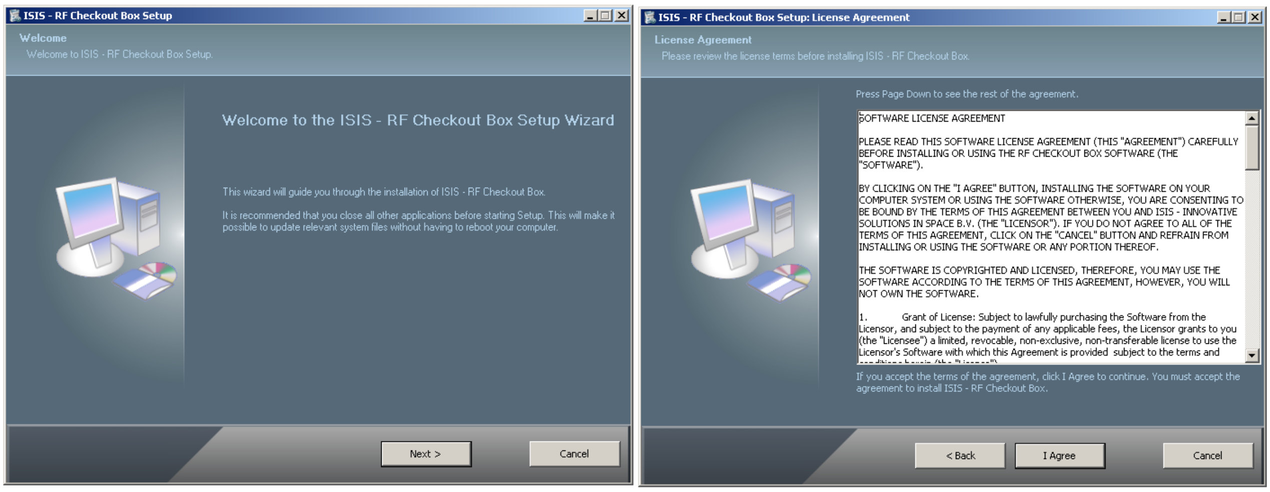


Figure 4-1 RF Installation program screenshot (1 and 2 of 4)

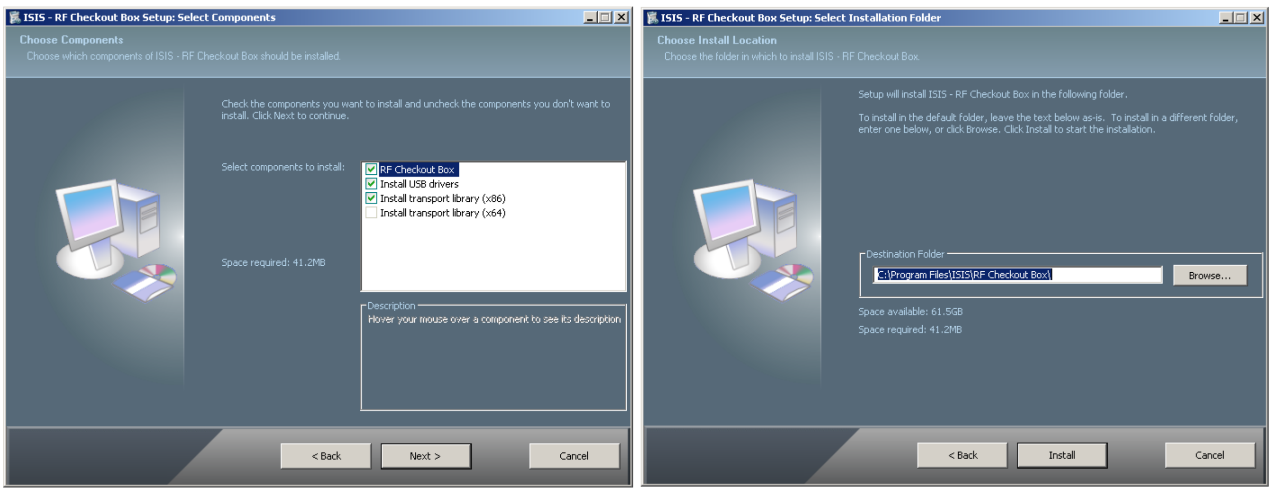


Figure 4-2 RF Installation program screenshot (3 and 4 of 4)

During installation, as depicted in Figure 4-2, please select "Install USB Drivers" (to make sure Windows recognizes the hardware peripheral) and "Install transport library" (to ensure the software required for communication over USB is installed). The installer will automatically detect, based on the computer architecture and JAVA virtual machine installed, the best software architecture to install (32- or 64-bit). Please make sure that the hardware is connected to the PC during this process.

During the software installation process, the installer may require the user to confirm the installation of an unsigned driver (it may happen 3 times): this procedure is absolutely safe and is required to install the driver. Please select **Install this driver anyway**. Not confirming the installation of the driver will result in a non functional setup.

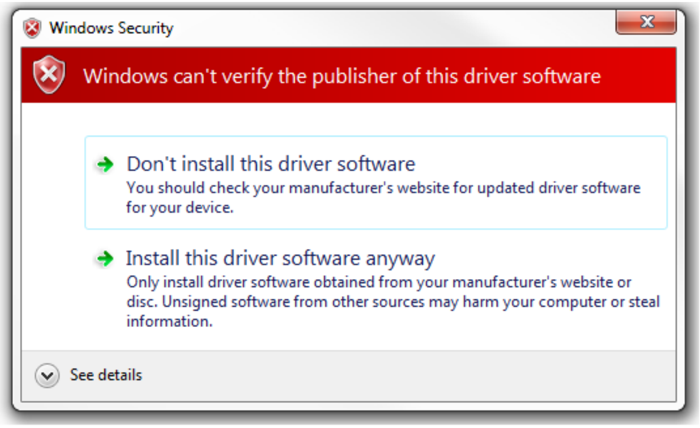


Figure 4-3 Unsigned driver installation confirmation

Figure 4-4 shows the transport library installation script first screen: please click next at every step and complete the installation. While installing the 64-bit version, please make sure to install it in the 64-bit program folders (for example C:\Proman Files and NOT C:\Program Files (x86)).

Figure 4-4 Transport library installation program

If, after connecting the hardware to the computer, Windows reports that no driver was found, it is always possible to perform a manual installation by going to the Windows **Control Panel** and selecting the **Device Manager**. Please follow Figure 4-5 to Figure 4-7 to manually install the drivers.

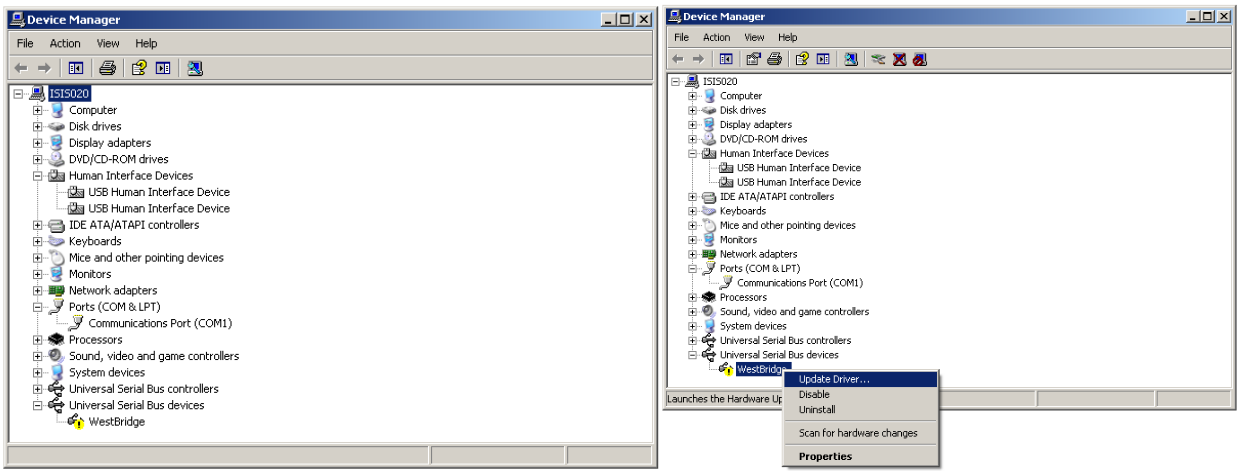


Figure 4-5 Manual driver installation (1 and 2 of 6)

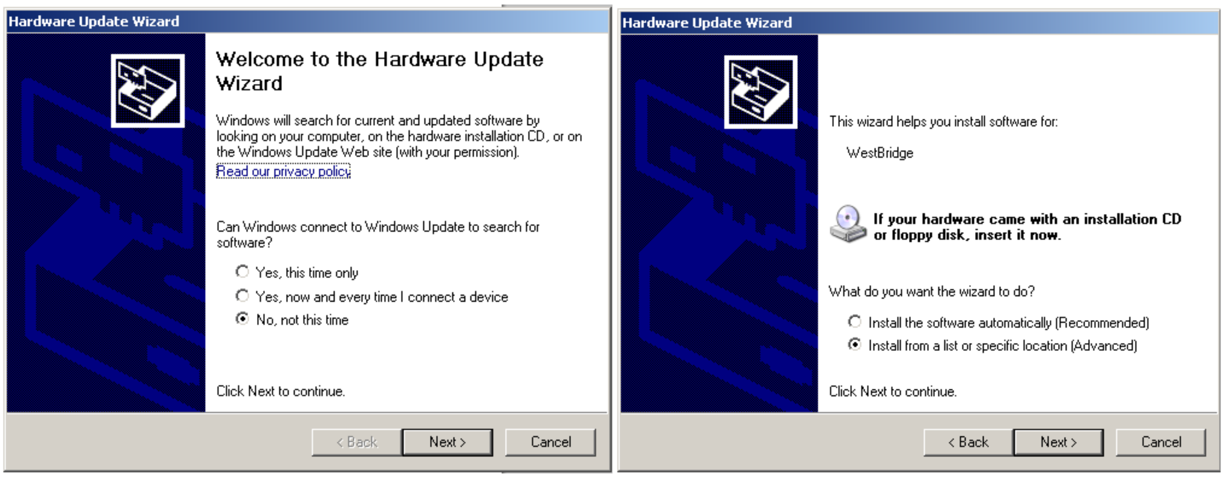


Figure 4-6 Manual driver installation (3 and 4 of 6)

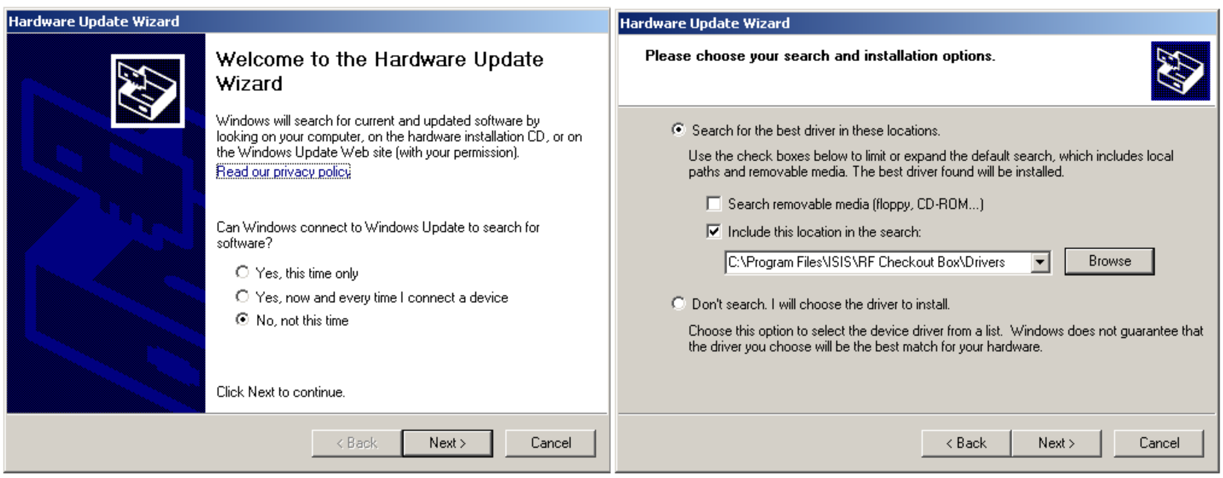


Figure 4-7 Manual driver installation (5 and 6 of 6)

4.2 Linux

The following installation guide was compiled using a clean Debian 7 distribution, but it can be used for other Debian based distributions. The software can also be installed on different Ubuntu distributions.

Before beginning with the software installation, please ensure that the Linux kernel version is 2.6.32 or later: this is required for fast USB operations using the provided driver (an older kernel version may not guarantee a high speed transfer between the SDR and the computer, actually limiting the maximum data rate achievable with the receiver).

A Java JRE version 1.7 is needed to use the software: the software dependency is handled through the Linux packaging system so, if the Java JRE is not present, the dependency can be automatically resolved.

First, locate the software folder in the provided media support: the software is provided in multiple ***.deb** Debian packages for simple installation.

The **isis-core** package (included in the distribution) is required and it should be installed first (as **root** user type **dpkg –i isis-core*** from the folder containing the package).

To properly communicate with the device, the low level driver needs to be installed: it can be found in a package called **libuhd003** and **uhd-host** in the distribution repository or alternatively, these packages are also provided for Debian. If the package cannot be found, please refer to

http://ettus-apps.sourcerepo.com/redmine/ettus/projects/uhd/wiki

for the source code.

To install the provided packages, please type as **root** user **dpkg –i package_name** where package_name is replaced with the file name of the package. Please make sure the package corresponding to the computer architecture is installed (i686 / x86_64). At last, the Checkout Box software can be installed by typing as **root** user **dpkg –i rfcheckoutbox***.

Eventual package dependencies which were not installed yet (such as, maybe, the Java JRE) can be installed automatically by typing as **root** user **apt-get –f install** as root user. This will force the installation of all the missing packages and complete the installation of the packages.

To allow non-root users to change the scheduling priority (which is done by the software to optimize system schedule), please add the following lines to the **/etc/security/limits.conf** file:

<my_group> - rtprio 99

Please replace <my_group> with a group to which the current user belongs. Settings will not take effect until the user has logged out and in again.

4.3 First Run

Once the software is successfully installed, the application can be tested.

Under Windows the application can be started from the Start menu: please select **ISIS → RF Checkout Box → RF Checkout Box**.

Under Linux please start the interface software by typing **rfcheckoutbox** at Linux prompt or look in the start menu under **Applications → ISIS → RF Checkout Box → RF Checkout Box**.

In case of errors, log messages are appended to a log file that can be accessed also from the Start menu. The file can be found under the following folder: \USER\HOME/.isis/rfcheckoutbox.log where USER\HOME is the user home directory (for example /home/username under Linux or C:\Users\username under Windows).

NOTE: If an issue with the software needs to be reported, please provide us with the log file together with the description of the problem.

Figure 4-8 shows the RF Checkout Box window, after connecting the hardware and selecting it from the **Radio** select box.

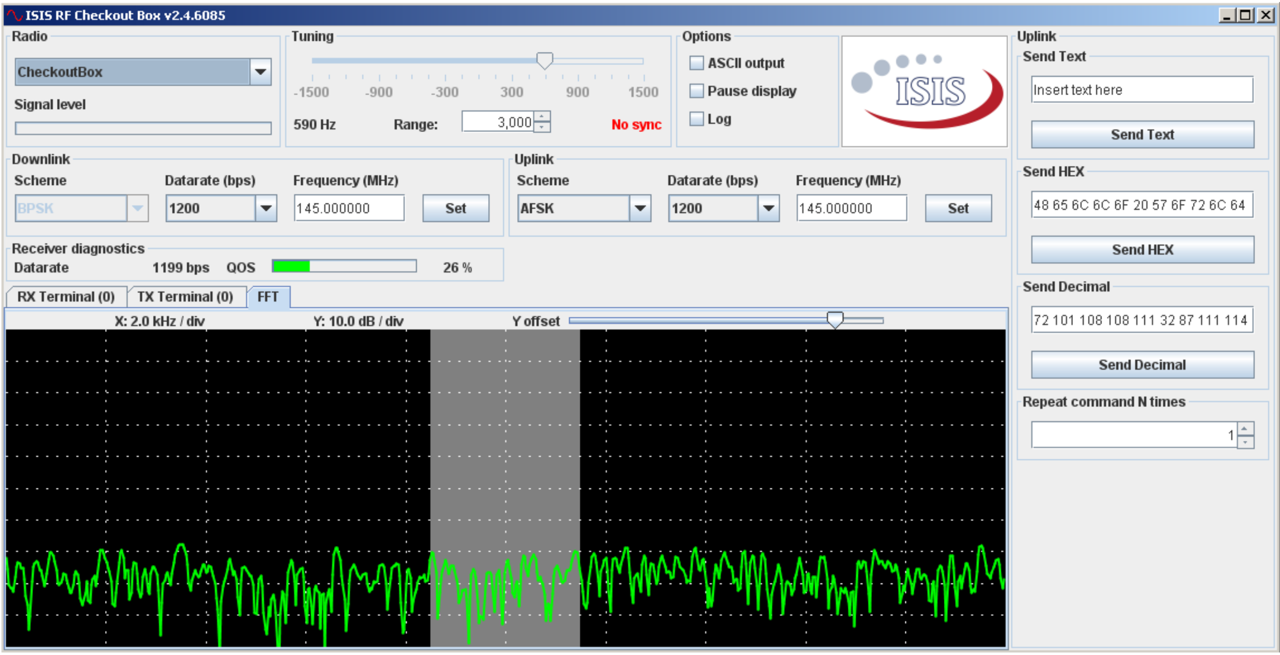


Figure 4-8 RF Checkout Box main window

5 Radio operations

The RF Checkout Box employs a software defined radio for digital data transmission and reception: baseband signals generated by a high speed DAC are up-converted to RF in the transmitter chain while RF signals are down-converted and sampled using an ADC in the receiving chain. Data modulation / demodulation are performed digitally on the computer with digital signal processing techniques.

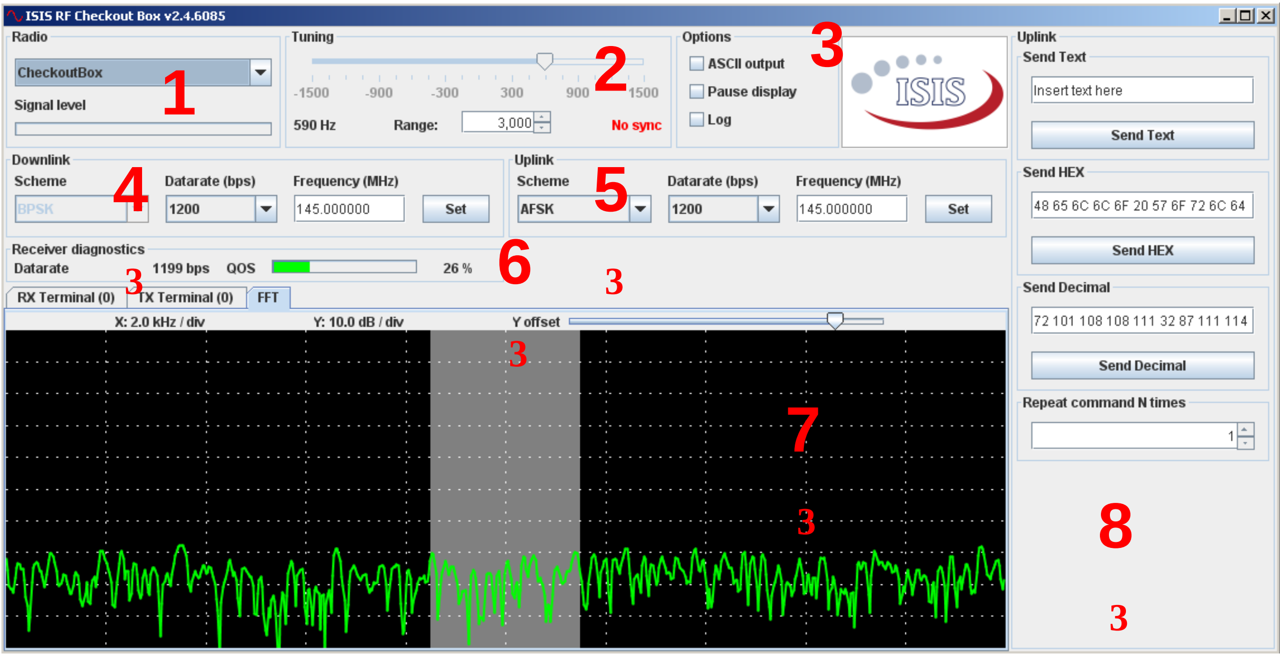


Figure 5-1 RF Checkout Box main window

The RF Checkout Box can be controlled through its GUI, which is depicted in Figure 5-1. The main available controls are highlighted in the figure and are better detailed in the next paragraphs.

- **Input Source Selection** (item 1 in Figure 5-1): the top select box is used to select the desired input source. When a new source is selected the demodulator and modulator are restarted. The signal level indicator displays the input signal strength: if the indicator bar is red, the input signal is too weak or too strong for optimal operations, while a green bar indicates a good input level (the bar will be constantly changing depending on the received input power). The power indicator increases from left to right, so a full bar indicates a strong signal.
- **Fine tuning** (item 2 in Figure 5-1): this control shows the actual reception frequency (relative to the radio centre frequency). The frequency information is computed by the demodulation loop and when the receiver locks on an input signal, a lock indicator is displayed (green indicator). A combo box is also available to narrow the receiver bandwidth in case reception is particularly difficult.

- **Options panel** (item 3 in Figure 5-1): the **ASCII Output** control is used to display packets in the Terminal window as ASCII characters instead of hexadecimal data. The **Pause Terminal** check-box can be used to stop incoming packets display in the Terminal window. The **Log** check-box instead enables packets logging (see Section 7 for details).
- **RX Channel settings** (item 4 in Figure 5-1): the **Modulation select** box allows selecting one among the available modulation schemes. The **Datarate select** box allows choosing the proper data rate. When a new modulation or data rate is selected the demodulator is restarted. The receiver centre frequency can be set by typing a new value (in MHz) in the text box and clicking the **Set** button.
- **TX Channel settings** (item 5 in Figure 5-1): the **Modulation select** box allows selecting one among the available modulation schemes. The **Datarate select** box allows choosing the proper data rate. When a new modulation or data rate is selected the modulator is restarted. The transmitter centre frequency can be set by typing a new value (in MHz) in the text box and clicking the **Set** button.
- **Link diagnostics** (item 6 in Figure 5-1): This section displays the measured data rate on the downlink channel and the estimated signal quality (see Section 5.3 for further details).
- **Display tab** (item 7 in Figure 5-1): This tab is a multi-function panel used to display a wide range of information regarding the received / transmitted packets and the input spectrum (see Section 5.2 for further details).
- **Uplink panel** (item 8 in Figure 5-1): this panel is used to send commands to the satellite. When a new command has been sent, it will also be displayed in the Terminal panel inside the Display tab.

5.1 Tuning

The tuning panel reports the tuning status of the demodulator: a lock indicator (as can be seen in Figure 5-2) is provided which shows if the receiver is locked on an incoming carrier. Furthermore, a frequency indicator is provided to show the actual receiving frequency (the frequency read-out should be considered as an offset to the radio frequency).

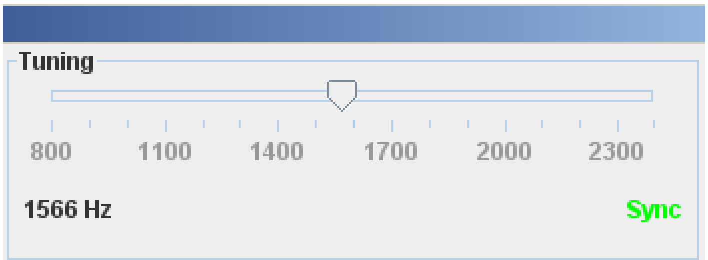


Figure 5-2 Tuning panel with receiver in LOCK

5.2 Display Tab

The display tab is a multi-function tab used to display a wide range of information regarding received packets, input spectrum and received signal statistics.

5.2.1 RX and TX Terminal Panel

The terminal panels are used to display the received and transmitted packets, as can be seen in Figure 5-3. This panel is used to display all the received and sent packets. The user can select to display the content of the packet in HEX or in ASCII on the **Option Panel** (item 3 in Figure 5-1). This panel keeps track of the most recent 100 received / transmitted packets.

Each time a packet is received, it will be displayed in the **RX Terminal** tab and each time a packet is transmitted, it will be displayed in the **TX Terminal** tab (unless **Pause display** is selected). The number of received and transmitted packets is also displayed in the tab title bar and it can be cleared by right-clicking on the panel title and selecting **Clear** (each panel counter can be reset separately for convenience).

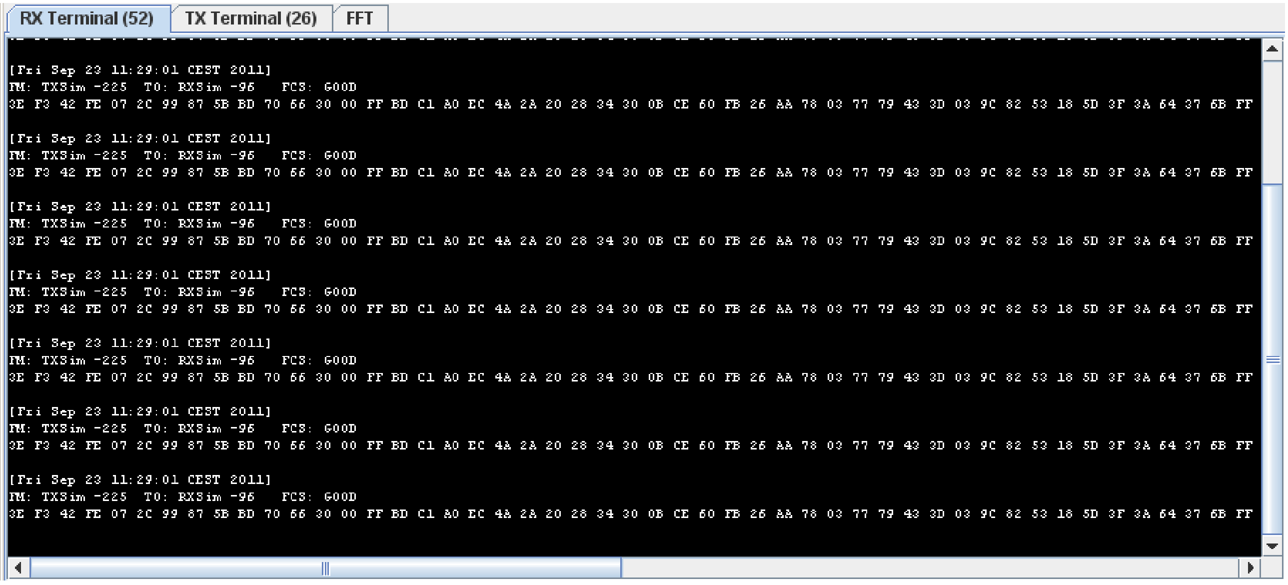


Figure 5-3 Terminal panel during data demodulation

5.2.2 Spectrum Monitoring

Beyond data demodulation, the RF Checkout Box can also be used to perform spectrum monitoring: this is achieved by performing a Fourier transform on the acquired samples. Figure 5-4 shows the spectrum analysis panel.

The vertical axis represents the received signal power (even if this is non calibrated in W or dBm), while the horizontal axis represents frequency, as in a common spectrum analyzer.

The vertical line in the centre of the tab corresponds to the centre downlink frequency, while the grey vertical bar represents the receiver reception bandwidth. To successfully receive a signal, it should be within this grey band.

The cursor can be used to estimate the frequency of a signal displayed in the spectrum tab: when the cursor is moved on this tab, a tooltip will be displayed showing the current frequency (as difference with respect to the centre frequency and as absolute frequency in Hz). Please see Figure 5-5 for details.

Furthermore, the screen has a grid to help identifying amplitude and frequency. The vertical offset of the signal can be adjusted using the **Y offset** slider on the screen.

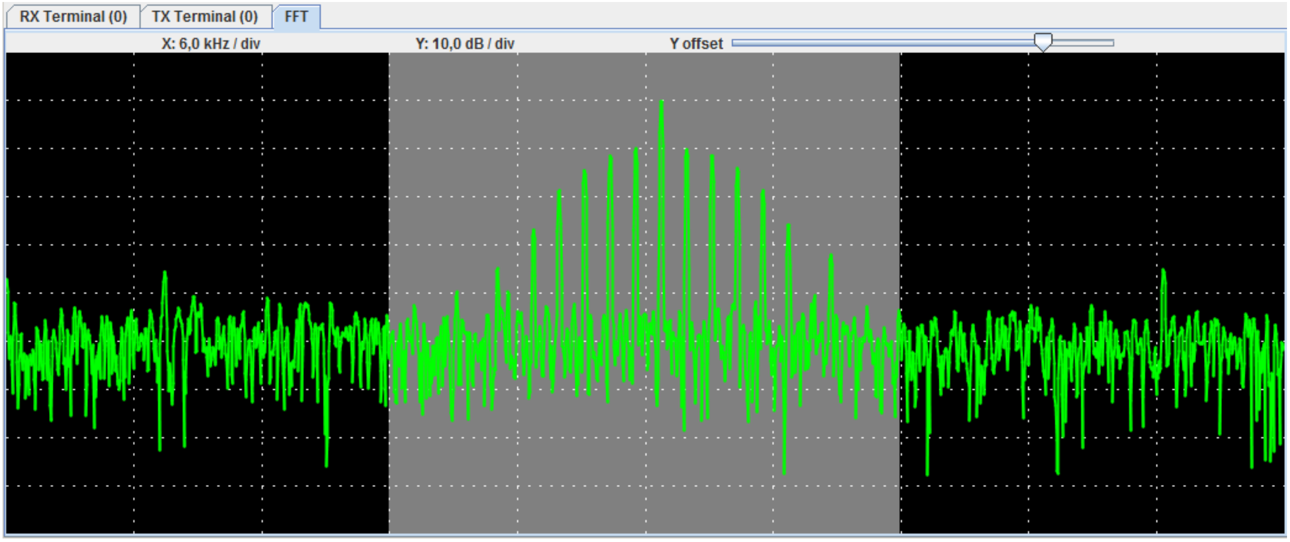


Figure 5-4 FFT panel

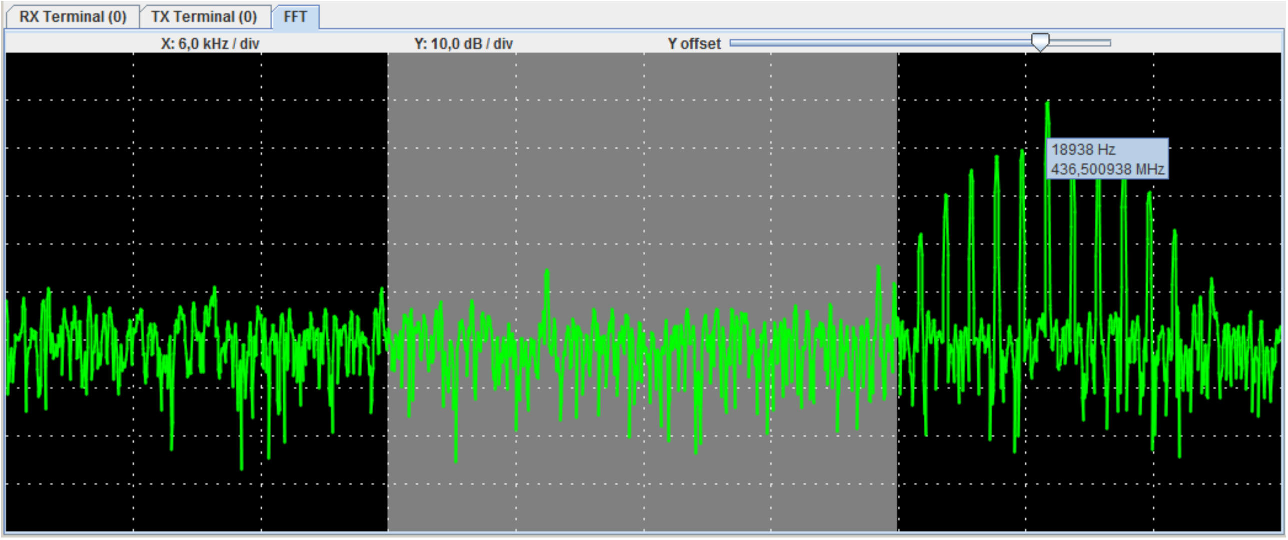


Figure 5-5 FFT panel with frequency indication

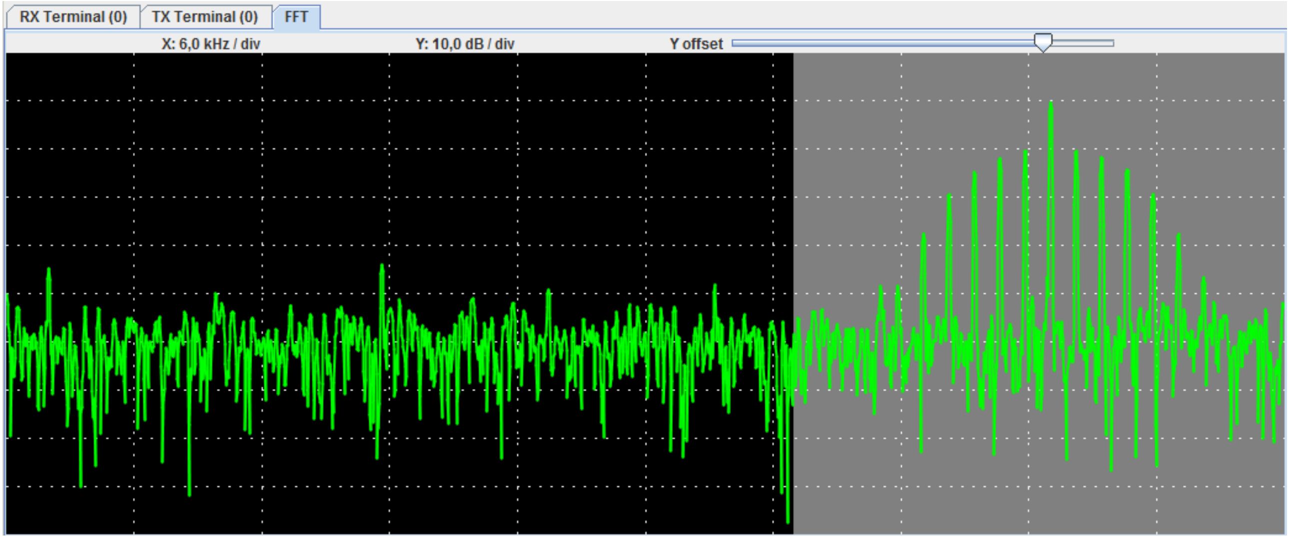


Figure 5-6 FFT centre frequency offset

If the signal to be received is outside the grey band, two things can be done:

- The centre frequency can be adjusted using the **Set** button
- The user can double click on the signal in the FFT tab: this will re-tune the receiver and move the grey band on the signal (see Figure 5-6 for details).

In case the received signal strength is too high or too low to properly visualize the signal, the user can drag the vertical scale up or down.

5.3 Link diagnostics

The diagnostic panel displays diagnostics on the receiver link, in particular the measured data rate and a quality indicator are displayed. The data rate is measured by the demodulator clock tracking loop and it is continuously updated during reception. The tracking loop can compensate a data rate error of approximately 5%.

The QOS indicator provides an estimation of the quality of the demodulated signal in relative terms: 100% corresponds to a perfect synchronization and a number lower than 50% represents a signal strongly corrupted by noise. For best results the QOS should be above 60% - 70%.

5.4 Uplink panel

The Uplink Panel is used to directly send commands to the satellite (see Figure 5-7 for further details) but before being able to transmit packets, a valid radio amateur callsign should be specified: the first time a command is sent, if no valid callsign has been provided yet, a pop-up window will show up (see Figure 5-8).

NOTE: Only radio amateur license holders are legally allowed to transmit packets, and their callsign should be put in every packet. Legislation may vary according to the country in which operations are performed.

Commands can be sent by using the buttons in the panel and the data to be sent can be specified in three different formats (as an ASCII text, as hexadecimal or as decimal values). The radio will transmit AX.25 packets with the INFO field specified by the content of the text field corresponding to the button that has been pressed.

Commands can also be repeated by using the **Repeat commands N times** field: whenever this value is set to something different than 1 and one of the command buttons is pressed, a continuous sequence of commands will be sent. To stop a long transmission it is possible to select 1 in the text field and press again one of the command buttons.

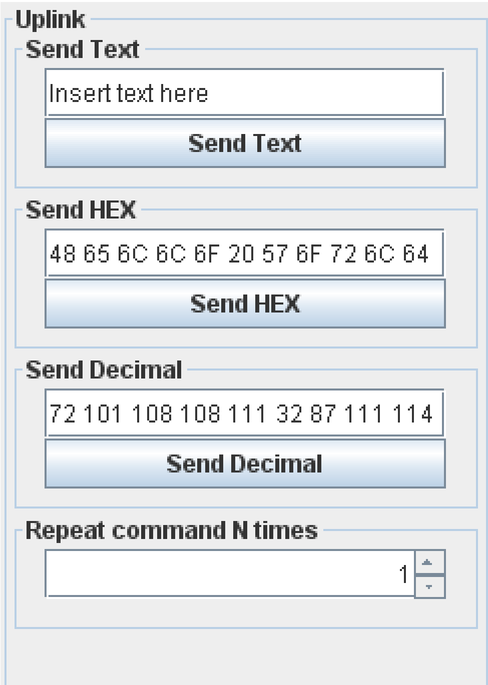


Figure 5-7 Uplink panel

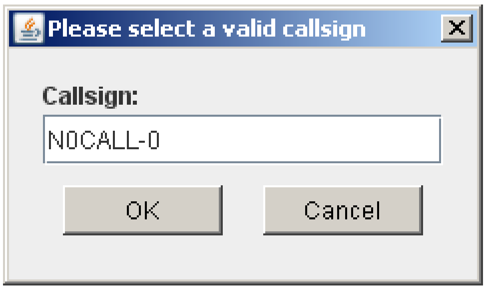


Figure 5-8 Callsign selection box

5.5 Quick start guide

This section shows the recommended procedure for successfully receiving data through the RF Checkout Box:

1. Connect the RF Checkout box to the PC with a USB cable.
2. Start the ISIS Checkout box application (please refer to section 4.3 for further details).
3. Select **CheckoutBox** from the Radio tab (please refer to section 5 and Figure 5-9 for further details). Please give time to the application to load the firmware on the radio (it generally takes about 30 seconds).

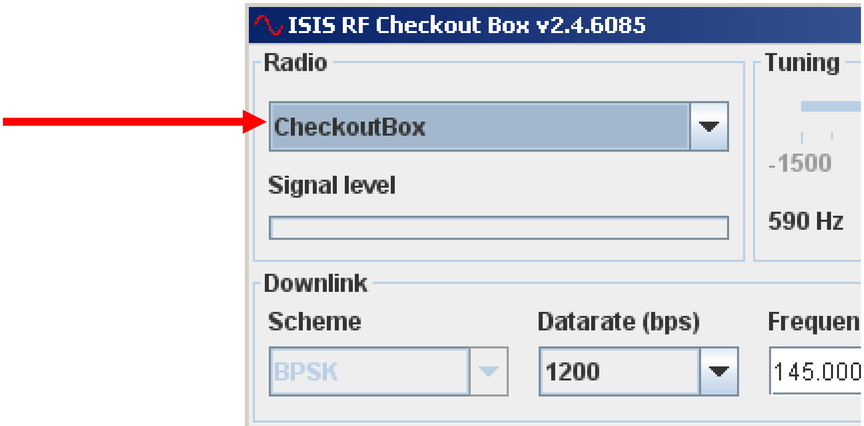


Figure 5-9 Select radio

4. Set appropriate reception frequency, modulation scheme and data rate (please refer to section 5 for further details. In the example below the reception frequency was set to 145 MHz, the modulation was set to BPSK and the data rate was set to 1200 bps).

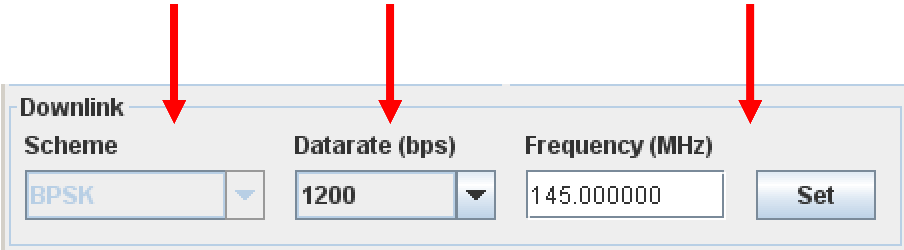


Figure 5-10 Set downlink parameters

5. Set appropriate transmission frequency, modulation scheme and data rate (please refer to section 5 for further details. In the example below the transmission frequency was set to 145 MHz, the modulation was set to AFSK and the data rate was set to 1200 bps)

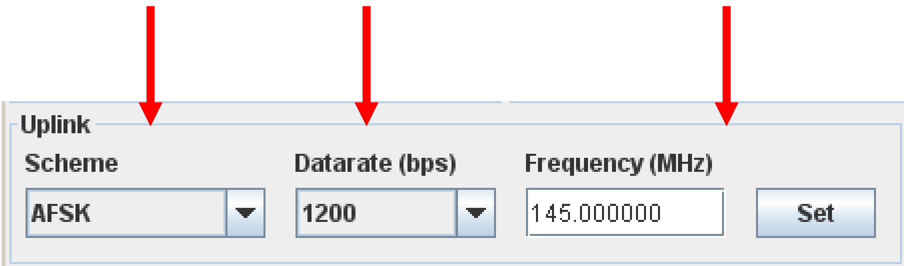


Figure 5-11 Set uplink parameters

6. In the **FFT** tab, please adjust the signal offset using the **Y** slider to fully display the received signal, as shown in Figure 5-12.

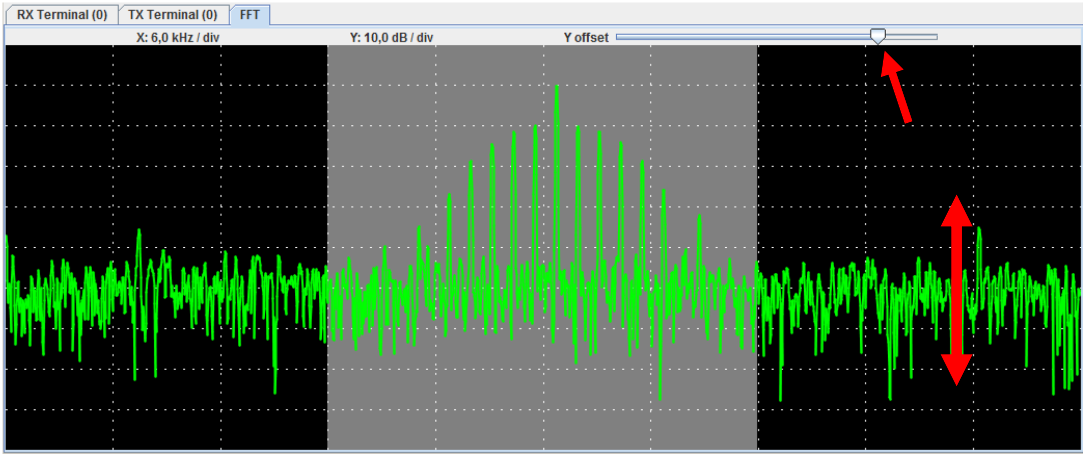


Figure 5-12 Signal offset adjustment

7. When the satellite is transmitting, the lock indicator will flash green and decoded packets will be displayed in the **RX Terminal** tab.
8. To send a command, please type the values to send in the text boxes in the **Uplink** panel and press the corresponding button (see section 5.4 for further details).

6 Data interface

Received data packets can be transferred to other applications by using a dedicated TCP/IP socket interface: this is the preferred way to access data for processing. When using it, please ensure the computer Firewall is not blocking the desired ports.

Two data interfaces are available, one based on KISS frames (see Section 6.1 for further details) and a binary one (see Section 6.2 for further details).

6.1 KISS interface

The full AX.25 received packet (beginning with callsigns and ending with the Info field, excluding the FCS) is transmitted on the socket enclosed in a KISS frame, to simplify the use of existing software for data decoding. This connection is available on the TCP port **3211**.

The socket connection is bi-directional, so user-defined packets can be sent to the satellite through this application, too. As in the downlink case, AX.25 frames (beginning with callsigns and ending with the Info field, thus excluding the FCS) should be enclosed in KISS frames, as for the receive case.

To retrieve the received packets it is necessary to connect a socket client on the receiver PC machine IP. A simple terminal application, like **Realterm** (http://realterm.sourceforge.net) can be used on a Windows machine to check the received packets. This application is not necessary for correct data reception.

Applications designed to operate using a serial port can be used if their output is redirected to a TCP socket. Linux applications can use **socat** for tunnelling (please check **man socat** on a console prompt for further info) while Windows users can use **com0com** and **com0tcp** (http://com0com.sourceforge.net/doc/UsingCom0com.pdf for further details).

Please see http://www.ka9q.net/papers/kiss.html for further details on the KISS TNC protocol. A typical received frame is shown below as an example:

C0 00 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19 C0

The previous frame can be interpreted as follows:

- C0 – KISS frame preamble
- 00 – KISS data frame command
- 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 - AX.25 header
- 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19 - AX.25 packet payload (NO FCS included, it was discarded by the software)
- C0 – KISS frame end

6.2 Binary interface

The binary interface is used to transfer the received packets or the packets to send, together with further information regarding the packet (RSSI or reception timestamp). This is useful to acquire statistics about the received data to characterize the link. This connection is available on the TCP port **3210**.

Table 6-1 summarizes the data structure that encapsulates every received packet. Multi-byte fields, like **double** or **int** are sent in **Little Endian** order (so, least significant byte first). This is the common byte order for Intel CPU and for C/C++ applications (this simplifies copy / conversion routines). The **Frame received command** is 0x55 for received packets. EPOCH (or UNIX timestamp) is defined as January 1st 1970, 00:00:00 UTC. The **Frame length** field expresses the length of the whole frame, taking into account every field listed in Table 6-1.

	Downlink Frame description	
	Field	**Data type**
HEADER	Frame received command	unsigned char
	Frame length [bytes]	unsigned int32
PAYLOAD	Data rate [bps]	unsigned int32
	Modulation name length [bytes]	unsigned char
	Modulation name	unsigned char []
	RSSI [dB]	IEEE 754 double
	Frequency [Hz]	IEEE 754 double
	Packet length [bytes]	unsigned short
	Packet	unsigned char[]
	UTC Time from EPOCH [ms]	unsigned long

Table 6-1 Received frame data structure

As an example, upon reception of an AX.25 frame, on the binary socket the following bytes will be transmitted (in HEX notation):

55 92 00 00 00 B0 04 00 00 04 42 50 53 4B 70 76 CA D6 13 B3 4C C0 6D FF 52 6F 88 63 A1 41 6A 00 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19

The previous frame can be interpreted as follows:

- 55 - Frame received command
- 92 00 00 00 - Frame length [bytes] (146 bytes)
- B0 04 00 00 – Data rate [bps] (1200 bps)

- 04 - Modulation name length [bytes]
- 42 50 53 4B - Modulation name (BPSK in ASCII)
- 70 76 CA D6 13 B3 4C C0 - RSSI [dB] (-57.3990429391 dB)
- 6D FF 52 6F 88 63 A1 41 - Frequency [Hz] (145867831.662 Hz)
- 6A 00 - Packet length [bytes] (106 bytes)
- 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 - AX.25 header
- 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19 - AX.25 packet payload (NO FCS included, it was discarded by the software)

The uplink frames interface works in a similar way: communication details can be found in Table 6-2. The **Send frame** command should be 0x56 for the command to be sent.

The **Frame length** field expresses the length of the whole frame, taking into account every field listed in Table 6-2.

	Uplink Frame description	
	Field	**Data type**
HEADER	Send frame command	unsigned char
	Frame length [bytes]	unsigned int32
PAYLOAD	Packet	unsigned char[]

Table 6-2 Transmit frame data structure

To transmit an AX.25 packet on the binary socket, the following bytes could be transmitted (in HEX notation):

56 6A 00 00 00 A8 A4 B0 AA AC 40 60 F0 F0 F0 F0 40 40 E1 03 F0 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19

The previous frame can be interpreted as follows:

- 56 - Send frame command
- 6A 00 00 00 - Frame length [bytes] (106 bytes)
- A8 A4 B0 AA AC 40 60 F0 F0 F0 F0 40 40 E1 03 F0 - AX.25 header
- 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19 - AX.25 packet payload (NO FCS included, it will be calculated by the software)

As soon as a frame has been sent to the satellite, an acknowledgement message is sent as confirmation: the **Frame sent acknowledge** is 0x57 and the data structure is listed in Table 6-3. As an example, upon successful transmission of an AX.25 frame, on the binary socket the following bytes will be transmitted (in HEX notation):

57 92 00 00 00 B0 04 00 00 04 42 50 53 4B 00 00 00 00 00 00 00 00 6D FF 52 6F 88 63 A1 41 6A 00 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19

	Send Acknowledge description	
	Field	**Data type**
HEADER	Frame sent acknowledge	unsigned char
	Frame length [bytes]	unsigned int32
PAYLOAD	Data rate [bps]	unsigned int32
	Modulation name length [bytes]	unsigned char
	Modulation name	unsigned char []
	Reserved	unsigned char[8]
	Frequency [Hz]	IEEE 754 double
	Packet length [bytes]	unsigned short
	Packet	unsigned char[]
	UTC Time from EPOCH [ms]	unsigned long

Table 6-3 Transmitted data acknowledge structure

The previous frame can be interpreted as follows:

- 57 - Frame sent acknowledge
- 92 00 00 00 - Frame length [bytes] (146 bytes)
- B0 04 00 00 – Data rate [bps] (1200 bps)
- 04 - Modulation name length [bytes]
- 42 50 53 4B - Modulation name (BPSK in ASCII)
- 00 00 00 00 00 00 00 00 - Reserved
- 6D FF 52 6F 88 63 A1 41 - Frequency [Hz] (145867831.662 Hz)
- 6A 00 - Packet length [bytes] (106 bytes)
- 8A 82 A4 A8 90 40 00 9C 86 84 A4 64 40 00 03 F0 01 04 00 5F 00 00 00 01 F2 00 F6 00 32 CC CC 35 02 87 01 B7 03 04 00 04 00 00 00 4D 00 F1 00 95 01 00 00 00 00 00 00 04 00 12 1A 65 00 19 00 18 00 19 00 19 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 05 73 01 00 00 00 27 00 00 00 00 00 00 04 00 05 00 04 00 80 19 - Frame sent (NO FCS included, it was discarded by the software)

7 Data logging

As described in the previous sections, all the received and transmitted packets can be logged to a text file.

NOTE: The logging engine only records a limited number of received and transmitted packets: this is intended for debug and verification only. It is not intended as the main logging system for the data exchanged with the satellite.

If a recording system for received and transmitted packets is needed, please use the data interface (see Section 6 for details) and an external application.

Log files are stored in the following folder: $USER_HOME/.isis/ where $USER_HOME is the user home directory (for example **/home/username** under Linux or **C:\Users\username** under Windows).

Received packets are logged in a file called **downlink.log** while transmitted ones are logged in **uplink.log**. Packets logging is OFF by default and it should be turned manually ON by ticking the Log check box on the user interface (see Section 5 for further details).

An example log file can be found below.

[Thu Jul 12 17:12:47 CEST 2012] A8 A4 B0 AA AC 40 60 A6 A6 A6 40 40 40 E1 03 F0 48 65 6C 6C 6F 20 57 6F 72 6C 64 A7 DC

[Thu Jul 12 17:13:11 CEST 2012] A8 A4 B0 AA AC 40 60 A6 A6 A6 40 40 40 E1 03 F0 49 6E 73 65 72 74 20 74 65 78 74 20 68 65 72 65 00 54

[Thu Jul 12 17:14:28 CEST 2012] A8 A4 B0 AA AC 40 60 A6 A6 A6 40 40 40 E1 03 F0 30 30 30 30 30 30 30 A1 DD

Reception time (or transmission time depending on the file) is at the beginning of each line and then the packet HEX representation follows. The full AX.25 packet is represented, also including the FCS at the end.

8 Troubleshooting

This section lists some common problems that can happen during operations and also lists suggestions on how to solve them. If the problem cannot be solved using this section, please refer to section 9 for support.

8.1 Device not found

When selecting the Checkout Box from the **Radio** tab, the message in Figure 8-1 may be displayed. This message notifies that the hardware could not be detected on the system. This may be caused by 3 reasons:

- The device was not connected to the computer USB port. To solve the issue, please connect the RF Checkout Box to the computer.
- The operating system is not recognizing the hardware. To solve the issue, please check section 4.1 and 4.2 and re-install the drivers on the system. In Windows systems this error may occur the first time an RF Checkout Box is connected to a new USB port (even if the device has already been used on the same machine but on a different port) because of the Windows device enumeration strategy.
- The user does not have permissions to connect to the device. This issue is common on Linux systems and to be solved requires the user to set the access rules to the device. Please remove and re-install the **libuhd003** package as described in section 4.2.

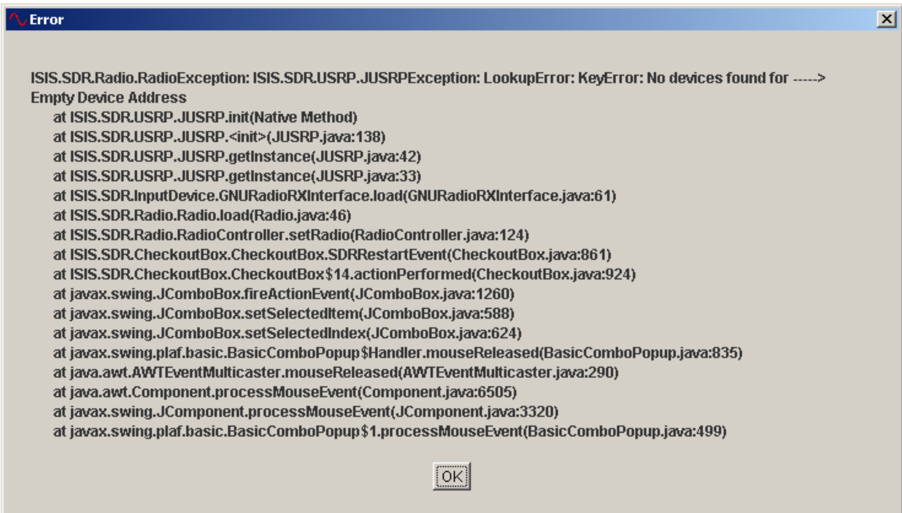


Figure 8-1 Device not found error message

8.2 Cannot load software on the platform

When selecting the Checkout Box from the **Radio** tab, the message in Figure 8-2 or Figure 8-3 may be displayed. This message notifies that the installed USB communication library is not compatible with the current Java virtual machine (32-bit library used with 64-bit virtual machine or the other way around). The issue can be solved by re-installing the software, as described in section 4.1 and selecting the proper transport library architecture (either x86 or x64).

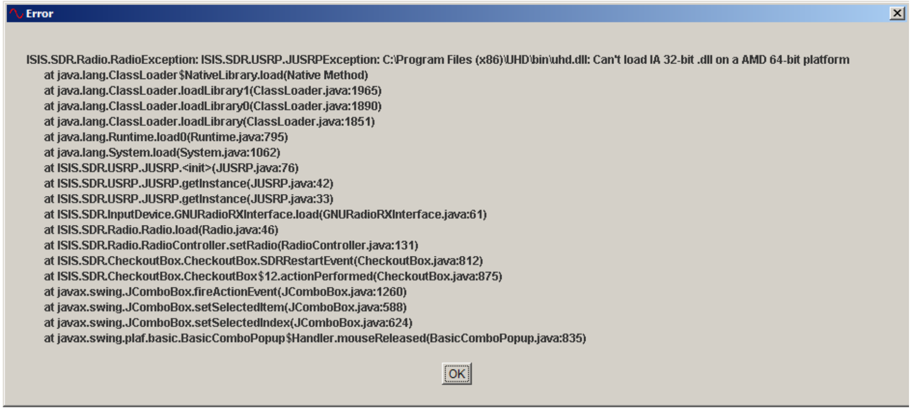


Figure 8-2 Software architecture mismatch (1 / 2)

Figure 8-3 Software architecture mismatch (2 / 2)

8.3 Address already in use

When starting the RF Checkout Box application the message shown in Figure 8-4 could be displayed. This could mean that an external application is already using the socket ports used for communication by the software (see section 6 for further details). This will prevent the socket ports from being used but will not prevent the application from receiving or transmitting data. To solve the issue, please check the documentation of the application that is using the socket ports required by this software.

The same error message could also be displayed in case multiple instances of the RF Checkout Box software were started at the same time. To solve this issue, please close all the running instances of the software and restart it.

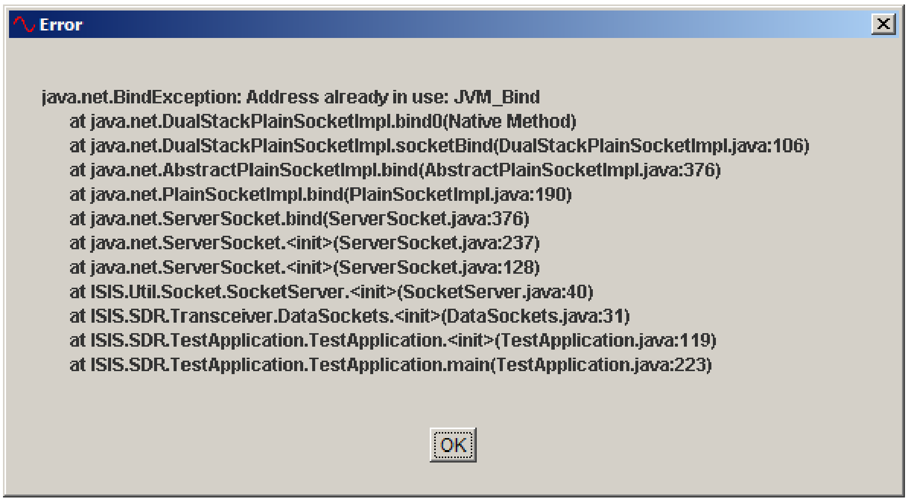


Figure 8-4 Application already started

8.4 Application seems to freeze when selecting a radio

When selecting a new radio from the **Radio** tab the application may seem to freeze: this behavior happens while the computer is loading the firmware on the radio for the first time. This may take up to about 30 seconds on most computers (but the loading time depends on the actual computer hardware): please wait for the load process to complete, then the radio will be fully operational or an error message will be displayed in case of failure.

8.5 No signal being displayed in the FFT tab

If, after starting the radio, no signal is displayed in the FFT tab but the tuning indicator (see section 5.1 for further details) is changing over time, the radio is working properly but the vertical offset of the display needs to be adjusted. Please adjust the **Y** slider to show the signal on the display. The value afterwards will be saved and kept for the next sessions.

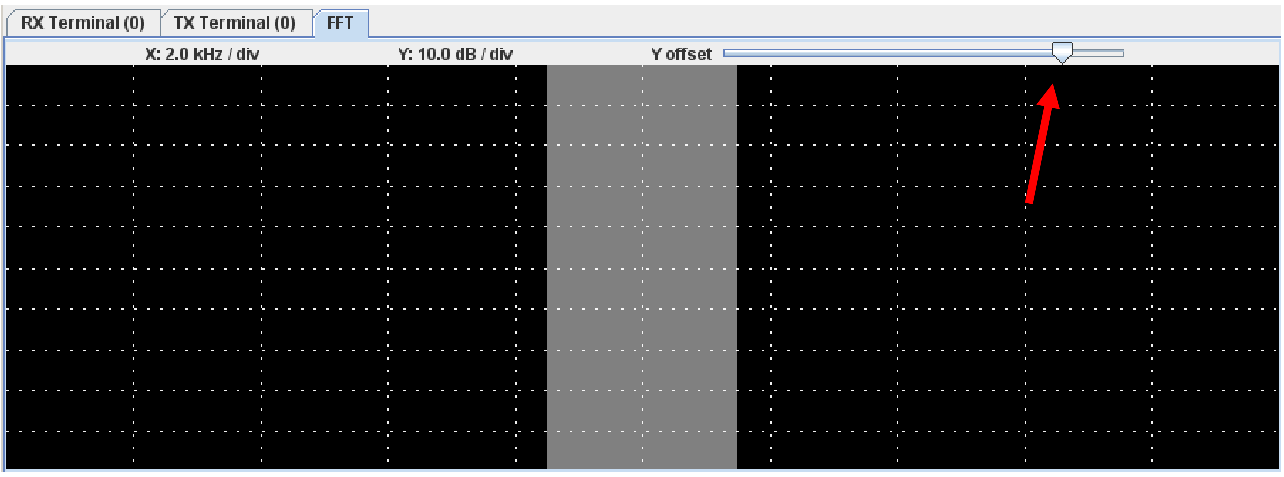


Figure 8-5 Adjusting the offset of the FFT tab

9 Support

For support, please contact:

ISIS - Innovative Solutions In Space
Molengraaffsingel 12-14
2629 JD Delft
The Netherlands
Telephone: +31 15 2569018
support@isispace.nl
www.isispace.nl