** Introduction

This document describes what I managed to discover about the protocol used to specify force effects to I-Force 2.0 devices. None of this information comes from Immerse. That's why you should not trust what is written in this document. This document is intended to help understanding the protocol.
This is not a reference. Comments and corrections are welcome. To contact me, send an email to: johann.deneux@gmail.com

** WARNING **

I shall not be held responsible for any damage or harm caused if you try to send data to your I-Force device based on what you read in this document.

** Preliminary Notes:

All values are hexadecimal with big-endian encoding (msb on the left). Beware, values inside packets are encoded using little-endian. Bytes whose roles are unknown are marked ??? Information that needs deeper inspection is marked (?)

** General form of a packet **

This is how packets look when the device uses the rs232 to communicate.
2B OP LEN DATA CS
CS is the checksum. It is equal to the exclusive or of all bytes.

When using USB:
OP DATA
The 2B, LEN and CS fields have disappeared, probably because USB handles frames and data corruption is handled or unsignificant.

First, I describe effects that are sent by the device to the computer

** Device input state

This packet is used to indicate the state of each button and the value of each axis
OP= 01 for a joystick, 03 for a wheel
LEN= Varies from device to device
00 X-Axis lsb
01 X-Axis msb
02 Y-Axis lsb, or gas pedal for a wheel
03 Y-Axis msb, or brake pedal for a wheel
04 Throttle
05 Buttons
06 Lower 4 bits: Buttons
   Upper 4 bits: Hat
07 Rudder

** Device effects states

OP= 02
LEN= Varies
00 ? Bit 1 (Value 2) is the value of the deadman switch
01 Bit 8 is set if the effect is playing. Bits 0 to 7 are the effect id.
02 ??
03 Address of parameter block changed (lsb)
04 Address of parameter block changed (msb)
05 Address of second parameter block changed (lsb)
... depending on the number of parameter blocks updated

```
** Force effect **
OP=  01
LEN= 0e
00 Channel (when playing several effects at the same time, each must be assigned a channel)
01 Wave form
        Val 00 Constant
        Val 20 Square
        Val 21 Triangle
        Val 22 Sine
        Val 23 Sawtooth up
        Val 24 Sawtooth down
        Val 40 Spring (Force = f(pos))
        Val 41 Friction (Force = f(velocity)) and Inertia (Force = f(acceleration))


02 Axes affected and trigger
        Bits 4-7: Val 2 = effect along one axis. Byte 05 indicates direction
                  Val 4 = X axis only. Byte 05 must contain 5a
                  Val 8 = Y axis only. Byte 05 must contain b4
                  Val c = X and Y axes. Bytes 05 must contain 60
        Bits 0-3: Val 0 = No trigger
                  Val x+1 = Button x triggers the effect
        When the whole byte is 0, cancel the previously set trigger

03-04 Duration of effect (little endian encoding, in ms)

05 Direction of effect, if applicable. Else, see 02 for value to assign.

06-07 Minimum time between triggering.

08-09 Address of periodicity or magnitude parameters
0a-0b Address of attack and fade parameters, or ffff if none.
*or*
08-09 Address of interactive parameters for X-axis, or ffff if not applicable
0a-0b Address of interactive parameters for Y-axis, or ffff if not applicable

0c-0d Delay before execution of effect (little endian encoding, in ms)


** Time based parameters **

*** Attack and fade ***
OP=  02
LEN= 08
00-01 Address where to store the parameteres
02-03 Duration of attack (little endian encoding, in ms)
04 Level at end of attack. Signed byte.
05-06 Duration of fade.
07 Level at end of fade.

*** Magnitude ***
OP=  03
LEN= 03
00-01 Address
```

```
02 Level. Signed byte.

*** Periodicity ***
OP=  04
LEN= 07
00-01 Address
02 Magnitude. Signed byte.
03 Offset. Signed byte.
04 Phase. Val 00 = 0 deg, Val 40 = 90 degs.
05-06 Period (little endian encoding, in ms)

** Interactive parameters **
OP=  05
LEN= 0a
00-01 Address
02 Positive Coeff
03 Negative Coeff
04+05 Offset (center)
06+07 Dead band (Val 01F4 = 5000 (decimal))
08 Positive saturation (Val 0a = 1000 (decimal) Val 64 = 10000 (decimal))
09 Negative saturation

The encoding is a bit funny here: For coeffs, these are signed values. The
maximum value is 64 (100 decimal), the min is 9c.
For the offset, the minimum value is FE0C, the maximum value is 01F4.
For the deadband, the minimum value is 0, the max is 03E8.

** Controls **
OP=  41
LEN= 03
00 Channel
01 Start/Stop
        Val 00: Stop
        Val 01: Start and play once.
        Val 41: Start and play n times (See byte 02 below)
02 Number of iterations n.

** Init **

*** Querying features ***
OP=  ff
Query command. Length varies according to the query type.
The general format of this packet is:
ff 01 QUERY [INDEX] CHECKSUM
responses are of the same form:
FF LEN QUERY VALUE_QUERIED CHECKSUM2
where LEN = 1 + length(VALUE_QUERIED)

**** Query ram size ****
QUERY = 42 ('B'uffer size)
The device should reply with the same packet plus two additional bytes
containing the size of the memory:
ff 03 42 03 e8 CS would mean that the device has 1000 bytes of ram available.

**** Query number of effects ****
QUERY = 4e ('N'umber of effects)
```

```
The device should respond by sending the number of effects that can be played
at the same time (one byte)
ff 02 4e 14 CS would stand for 20 effects.

**** Vendor’s id ****
QUERY = 4d (’M’anufacturer)
Query the vendors’id (2 bytes)

**** Product id *****
QUERY = 50 (’P’roduct)
Query the product id (2 bytes)

**** Open device ****
QUERY = 4f (’O’pen)
No data returned.

**** Close device *****
QUERY = 43 (’C’)lose
No data returned.

**** Query effect ****
QUERY = 45 (’E’)
Send effect type.
Returns nonzero if supported (2 bytes)

**** Firmware Version ****
QUERY = 56 (’V’ersion)
Sends back 3 bytes - major, minor, subminor

*** Initialisation of the device ***

**** Set Control ****
!!! Device dependent, can be different on different models !!!
OP=  40 <idx> <val> [<val>]
LEN= 2 or 3
00 Idx
   Idx 00 Set dead zone (0..2048)
   Idx 01 Ignore Deadman sensor (0..1)
   Idx 02 Enable comm watchdog (0..1)
   Idx 03 Set the strength of the spring (0..100)
   Idx 04 Enable or disable the spring (0/1)
   Idx 05 Set axis saturation threshold (0..2048)

**** Set Effect State ****
OP=  42 <val>
LEN= 1
00 State
   Bit 3 Pause force feedback
   Bit 2 Enable force feedback
   Bit 0 Stop all effects

**** Set overall gain ****
OP=  43 <val>
LEN= 1
00 Gain
   Val 00 = 0%
   Val 40 = 50%
   Val 80 = 100%
```

** Parameter memory **

Each device has a certain amount of memory to store parameters of effects.
The amount of RAM may vary, I encountered values from 200 to 1000 bytes. Below
is the amount of memory apparently needed for every set of parameters:
 - period : 0c
 - magnitude : 02
 - attack and fade : 0e
 - interactive : 08

** Appendix: How to study the protocol ? **

1. Generate effects using the force editor provided with the DirectX SDK, or
use Immersion Studio (freely available at their web site in the developer
section:
www.immersion.com)
2. Start a soft spying RS232 or USB (depending on where you connected your
joystick/wheel). I used ComPortSpy from fCoder (alpha version!)
3. Play the effect, and watch what happens on the spy screen.

A few words about ComPortSpy:
At first glance, this software seems, hum, well... buggy. In fact, data appear
with a
few seconds latency. Personally, I restart it every time I play an effect.
Remember it's free (as in free beer) and alpha!

** URLS **
Check www.immerse.com for Immersion Studio, and www.fcoder.com for ComPortSpy.

** Author of this document **
Johann Deneux <johann.deneux@gmail.com>
Home page at http://www.esil.univ-mrs.fr/~jdeneux/projects/ff/

Additions by Vojtech Pavlik.

I-Force is trademark of Immersion Corp.