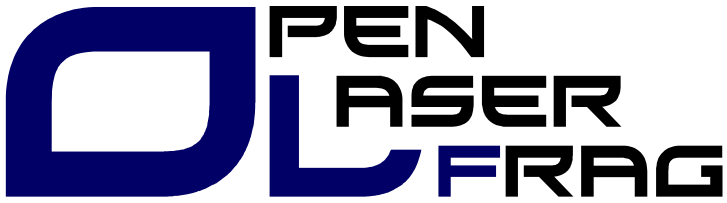


OpenLaserFrag
An Open-source laser pursuit gaming system

Specifications
Revision 0.9-draft3

Date: November 10, 2008

Author: Robin van Steenbergen (Stoney3K)
stoney3k@openlaserfrag.org

Contents

Change Log

August 13, 2008: 0.9-draft1

- Baseline specifications.

November 10, 2008: 0.9-draft3

- Added contents.
- Reference Implementation and Costume and Loadout Guidelines moved to Implementation Guide.
- General language and formatting.

Notes

Introduction

Thank you for downloading this specification and your interest in OpenLaserFrag. OpenLaserFrag (or OLF for short) is a community-driven project with the purpose of creating an extensible, modular, open-source laser gaming system. Similar in gameplay to Airsoft and Paintball, laser pursuit games have some significant advantages over the both. In this document you will find a brief overview of what the OLF system will consist of, followed by an extensive functional specification of every component. The eventual goal is that everyone can create their own implementation of this specification, while still remaining interoperable with everyone else. For those of you who want to get started quickly, a reference design and gear guidelines are provided in the appendices.
During the development track of OpenLaserFrag, a lot of commercial and DIY systems have been developed. A very large user base is in Australia, for the same reason OLF started development in the Netherlands, which is the strict ban on replica weapons which rules out Airsoft gameplay. Furthermore, laser tag has some significant advantages over projectile-based tactical and pursuit games and it allows for more extensive gameplay opportunities and cheat protection.
Of course, we are far from the only or the first organisation which attempted (or succeeded) in developing such a system. Starting with the early Worlds of Wonder sets in the 1980s, laser tag has gained a significant following and the advent of cheap and easy to program microcontrollers in the 2000s made the laser tag market take flight. Nowadays, hobbyists are building advanced systems like MilesTag and FragTag, which are on par with earlier military training systems, and commercial developers like Shoot The Moon have developed systems that are easy to use and get started with but still offer advanced gameplay possibilities.

OpenLaserFrag System Overview

While OpenLaserFrag is based on the existing features of the systems mentioned in the introduction, and may borrow design elements from other (open and unpatented) projects, it will offer a more extended feature set in the long run. The main goals for OLF are simple: **Modularity** and **extensibility**. This means that all OLF components are designed as self-contained modules, which are fully interchangeable. Regardless of implementation of a specific module, if the builder of that particular module designed it by this specification, it should work flawlessly with other modules, whether they were built by the same person or other hobbyists.

The modular nature of OLF does dictate a somewhat higher initial investment when building the first units, but this is marginal considering the cost of small microcontrollers these days. Instead of having the entire functionality put into one large, expensive chip with a long line of code, each module has a specific task contained in a small microcontroller. This offers the additional advantage of design simplicity and ease of testing.

Basically, when designing such a system, one starts out with functional requirements (features) in hand. For the 0.8 draft specification leading up to 1.0, the only requirement we put in is that we create a system that works in the very basics. This, functionally, means that the ability to land and register tags is required, along with a logic mechanism that allows interaction between the registering and the ability to land tags. Breaking this down, this comes down to the following components:

The **Personal Unit**, which is the core of the OLF game logic, and has the sole task of exchanging messages between other OLF modules, realizing factual gameplay. In the future, the Personal Unit will be extended to form a virtual "backpack", which can contain the player's ammunition, health, and other items.
The PU itself is a very basic unit, and can be built as a belt-pack to be worn on the player's uniform. Other implementations are, of course, also possible.

The **Sensor Controller** with attached sensors, which has the task of registering tags from other players, and translates them into messages for the PU to be processed. One can choose to integrate this with the PU into one casing or board, however merging the two functions into one chip would compromise the modular concept.

The **Gun** (or IR transmitter for political correctness) is responsible for transmitting a narrow infrared beam which is translated as a tag on reception by one of the sensors. The gun is an intelligent, self-contained unit – it will contain a trigger assembly so the player can control when to send tags. Future gun assemblies will be able to keep track of ammunition and rate-of-fire by themselves.

Specifications

The modular nature of OLF and the freedom of implementation of the modules demands a strict specification set, so module implementers know how their modules should behave to be interoperable with other players.

Section 1: FragWire Inter-module Communication

One of the most important considerations when designing a modular system is the way the modules communicate and interact. The logical choice for a starting point would be the Open Systems Interconnect (OSI) model, but this is too complex for microcontroller applications.
For OLF, we tried to keep to the OSI philosophy as much as possible, but it was cut down for simplicity. Basically, the OLF Inter-Module Communication protocol (fancifully named FragWire from now on) for the 1.0 draft consists of three layers:

Message Layer: This is the top level which all modules of a specific version need to understand, and the data contained in the Message Layer packets are actual gameplay events.
Data Link Layer: Facilitates packet encoding and rudimentary error checking.
Physical Layer: specifies the electrical signal, connector and cable requirements to communicate with OLF modules correctly on the lowest level.

The actual specifications for each layer (bottom up) will be outlined in this chapter. Note that there is a distinguishment between required and recommended specifications – without satisfying the former, interoperability cannot be guaranteed, where the latter will facilitate optimal communication and robustness.

Physical and Electrical

Currently, the physical requirements are different from the 1.0 milestone revision; this is mainly for ease of testing and keeping prototypes simple. This has obvious consequences for the 0.9-draft1 reference hardware design (see Appendix A).

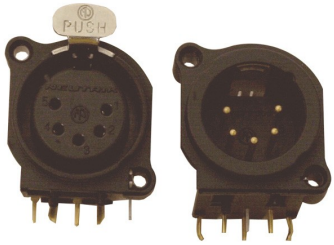
Electrical Specifications (Required)

Property	Min.	Typ.	Max.	Unit
Output Voltage (Logic High)	4.2			V
Output Voltage (Logic Low)			0.7	V
Output Source Current			20	mA
Input Voltage (Logic Low)	-0.5		1.5	V
Input Voltage (Logic High)	3.0		5.5	V
Input Leakage Current			1	uA
Input Impedance	10		25	kOhm

Electrical & Physical Specifications (Recommended)

Physical Interconnect

For interconnecting the modules physically, a somewhat ruggedized connector is required to prevent modules from becoming disconnected during play. For future implementations, the connector should have the possibility to transmit differential signals and supply power over the same cable, and lock into its receptacle to prevent accidental disconnects. Following this, the recommended connector is the **5-pole XLR connector series** by Neutrik and other manufacturers. Suggested is the following pin description:



5-pole XLR connector for FragWire		
Pin #	**Signal**	**Description**
1	GND	Signal common
2	NC	No Connect
3	D+	Serial data
4	VCC	Power live
5	VEE	Power neutral

Take note that this corresponds with the USITT DMX512 pinout, which allows devices that do not require power to be connected via a 3-pole XLR connector via an off-the-shelf adapter plug. Pin 2 is not connected for the 0.9 revision, since differential data transmission is not yet specified. The female connector should be used at the module end, with the male connector at the cable end.

Cabling

The cable to be used for connecting the modules is **120-ohm impedance twisted pair cable**, again in accordance with DMX512 for future upgrade to RS485. For shorter cable segments, other cable types are allowed. For example, the recommended cable to connect the Personal Unit to a gun is 3-pole stranded, wound headphones or microphone cable (curly cord) to make moving around with the gun without getting tangled up in cables.

Data Link and Packet Structure

Once the physical characteristics of the FragWire bus are met, for a module to be interoperable with other OLF modules, the data link requirements are the second layer which needs to match the rest of OLF.

Byte format

On the lowest level possible above the electrical characteristics, FragWire transmits bytes as **9600-baud serial data, with 8 data bits, 1 stop bit and no parity.** This facilitates the use of internal UARTs which are built into virtually every microcontroller, and it is possible to use the RS232 port on a PC for testing.

Packet Format

The FragWire packets describe the encapsulation of FragWire messages into bytes which can be transmitted over the physical bus. The packet format is fairly straightforward, and as follows:

Preamble	Destination	Source	Length	Data	CRC8

Preamble: A simple magic-number style code which makes it easy to determine that this is actually an OLF packet. This may not make as much sense internally, but may prove useful if an OLF device is inadvertedly connected to another similar bus (such as DMX512).

Destination: The address of the destination device for the packet. See Addressing Scheme below.

Source: The address of the device generating the packet.

Length: The length of the packet payload in bytes.

Data: The data payload of the packet.

CRC8: The packet's CRC8 checksum.

FragWire: Message Structure

The FragWire message exchange is what turns the separate OLF modules into a working laser tag system. They represent functional game events, and most of them end up at or originate from the Personal Unit as the result of processing of game logic.
Leading up to the 1.0 milestone revision, the message structure is relatively simple and facilitates the basic functionality for laser tag games while still remaining flexible for future improvements. The message command code list is specified here. (Please note, all command codes are in hexadecimal C-style format)

Command Code	Category	Description
0x00	General	NOP
0x01	General	Reset
0x02	General	Heartbeat
0x10	Sensors	Player Hit
0x20	Weapon	Weapon Enable
0x21	Weapon	Weapon Disable

Obviously, more command codes will follow in future revisions, when more specific features are being implemented. All of the functional descriptions for the above mentioned commands follow.

General: NOP

A general no-operation command, designed to disable functionality if a microcontroller happens to responds to bus silence (hence only receiving zeroes). This is not generally meant to be transmitted by any unit, but units should be prepared to handle it.

General: Reset

A simple, straight-forward command to trigger a unit for (soft) reset.

General: Heartbeat

This packet is meant to keep track of which modules are still in operation, which is also a mechanism for cheat protection. After all, a player who unplugs the sensor controller would be invulnerable if the “failure” of the sensors went unnoticed. To counter this, all devices should send out this heartbeat packet once every 60 seconds, to be governed by the PU. The PU should send this signal to the guns as well, to counter usability of guns when the PU is in a failed condition. The recommended heartbeat message frequency is **once every two seconds** for each module. More than twice per second is not recommended because of limited bus capacity.

Sensors: Player Hit

Transmitted to the Personal Unit by the sensor controller when a valid hit packet is received by one of the infrared sensors.

Weapon: Weapon en/disable
Transmitted to the weapon(s) connected by the Personal Unit when a weapon should be disabled or enabled.

Section 2: Infrared Inter-Player Communication
Probably the most fundamental concept in laser tag is the ability to tag someone. In nearly all systems, this involves transmitting a harmless, narrow infrared beam towards the target containing a specific code. This is almost identical to the way a TV remote sends commands to a TV set, and in some cases laser tag protocols are even compatible with remote protocols, which allows learning remotes to be used for some functions like transmitting reload and medic packets.

Infrared packets
In OpenLaserFrag, the infrared data packet format is based partly on MilesTag, and uses a time-based encoding system on top of a 40 kHz carrier.
For the 1.0 milestone revision, the MilesTag protocol is used, albeit in a simplified manner. For reference, see:
http://www.lasertagparts.com/mtdesign.htm

The simplification we're making for 1.0 is that the TeamID is set to 1 and the player ID is set to 0, resulting in every player being represented as number 32 (0x20 hex). Data byte 2 is set to zero for OpenLaserFrag.

Optics
The IR transmission is the most determining factor when it comes to weapon range and accuracy, the sensors used being the second. The current recommended optics configuration is an infrared LED used for transmission, collimated by a single lens. For further detail, see Figure 1.

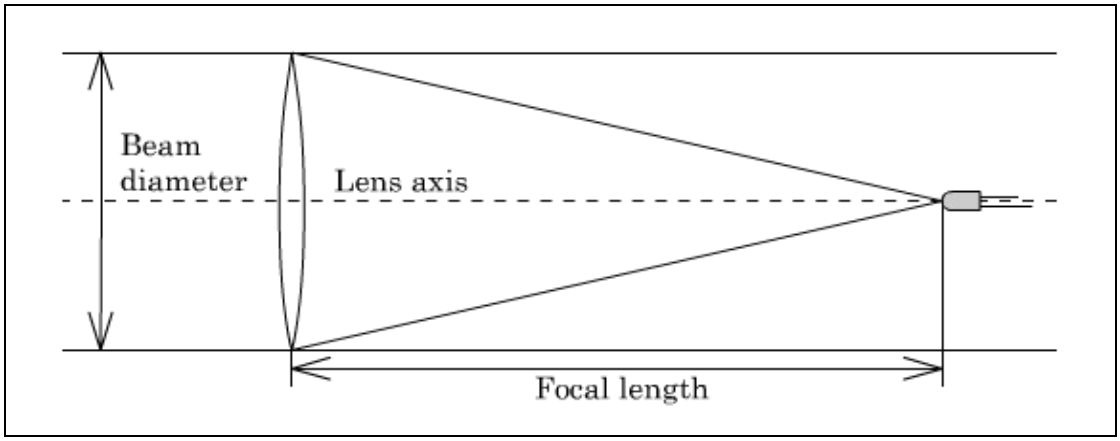


Figure 1: Recommended optics configuration for OpenLaserFrag.

As shown, the LED is centered along the lens axis in the focal point. A longer lens (with larger focal distance) will provide for more range and less beam spread, but it will make the overall beam diameter larger.
The recommended LED is the TSAL6100 from Vishay Semiconductor. It offers the best power-to-current ratio and the least (unfocused) beam spread.

Section 3: Module Functional Description

TODO: Add images

A very important part of the OLF specification is specifying how modules should behave and react to specific message exchanges. The simple, straightforward description of each module makes the overall design easy to understand, and also facilitates easier design changes when they are needed. Each of the individual modules is described below.

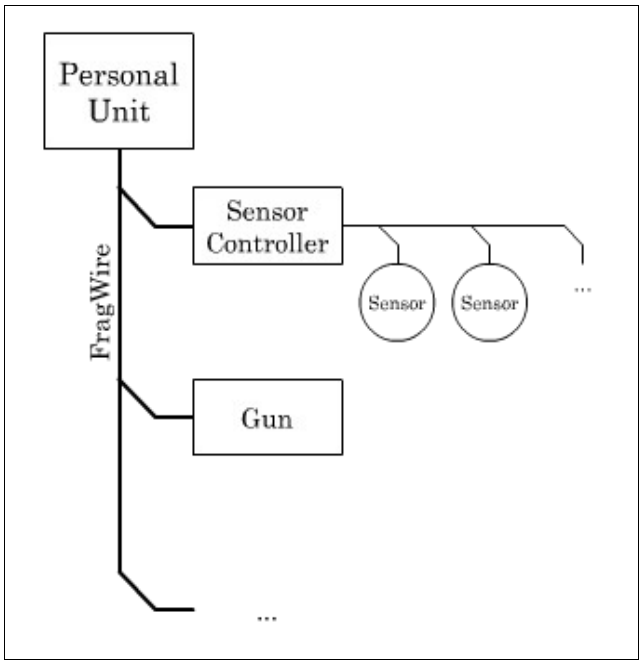


Figure 2: Overview of the OpenLaserFrag module network.

The OpenLaserFrag philosophy is that the player experience is defined by the combination of modules and that the player can determine his level of experience by adding, removing or changing the module configuration. In future versions, more modules can be added which provide other functionality, like human interface components (LCD, buttons, speaker, rumble), other weapons and sensors, statistics tracking, communications and other game elements. Since the FragWire protocol has been defined in a strict manner, all of the modules will interact flawlessly and provide the player with a game experience regardless of the type of modules or combination that is connected.

In OpenLaserFrag 1.0, the three basic module types will be defined. These are the **Personal Unit**, the **Sensor Controller** and the **Gun** module. The combination of these three makes a basic but versatile laser tag game possible.

The Personal Unit

The Personal Unit is the core of the OpenLaserFrag module network, distinguishing it from other laser tag systems which provide full, monolithical units. Although those systems are highly functional and in most situations very useful, the use of a Personal Unit allows for more flexibility in composing a loadout and playing different game characters. For example, one can play an unarmed character (such as a hostage or VIP) while still able to interact in the game. Think of the Personal Unit as your digital back-pack, where you carry your ammo, health and possible other items. It handles the game logic and is the bus master, and most of the message exchanges go via the Personal Unit.

The Personal Unit is a **required** module for correct operation of OpenLaserFrag, as it also masters and terminates FragWire. Without a Personal Unit connected, all OLF modules will remain inert and will not function. Additionally, the Personal Unit provides power to smaller modules powered through FragWire.

For the 1.0 milestone revision, the game logic inside the Personal Unit is simple and straightforward, but it can be used for a variety of game types. The logic is as follows:

- In normal game operation, the Personal Unit has no task other than checking for heartbeat signals.
- Upon reception of a *[Sensor: Player Hit]* message from the sensor controller, the PU responds with the transmission of a [Weapon: Disable] message towards the gun, temporarily disabling the hit player.
- Ten seconds later a *[Weapon: Enable]* message will be transmitted, allowing the hit player to continue play.
- If the player is hit within the ten-second time interval, the timer is reset, so a player will have to take cover if he or she wants to return fire.

This game logic is similar to the gameplay found in older indoor arenas, albeit without the extensive statistics tracking done there. It allows for quick free-for-all play and simple, but effective scenario play.

Sensor Controller

The sensor controller is one of the other key components in the OLF device chain. It takes the input from a set of infrared sensors and translates that into a *[Sensor: Player Hit]* message, notifying the Personal Unit of a confirmed tag. For the 0.9 version, standard MilesTag specification sensors are used (although the hit LED is not being used yet), and the sensor controller is powered from the FragWire directly. It can be integrated into the same casing as the Personal Unit if one wishes to do so. The sensors themselves are connected through RJ-11 connectors and 4-wire telephone cable like in MilesTag. Cheat-proof sensors are being designed for 1.0 which immediately disable the player when the sensors are tampered with.

Gun

The gun is, in principle, the most essential piece of a laser tag system. In OLF. The gun itself is an intelligent infrared transmission device capable of handling packet format, rate of fire and ammunition by itself, and will, for most users, be built into a rifle-like casing. Care must be taken to respect local weapon legislations: In the United Kingdom, the Netherlands and possible other countries, possession of replica weapons is illegal or virtually impossible for civilian players. The gun should contain a method of infrared transmission, although lasers are not allowed for eye safety reasons. When a LED is used, it is often focused or collimated through a lens.

The gun should provide for a trigger to control the transmission of tags. In OpenLaserFrag revision 1.0, all guns are required to provide for semi-automatic operation. No requirements on range are specified, and other functions like keeping track of ammo, sound and LCD screens are completely optional.