# RecoilGun Firmware Configuration Guide

|  |  |  |  |
| --- | --- | --- | --- |
| ***Version 1.1*** | ***Matteo Scordino*** | ***24 May 2017*** | ***Initial Version*** |

Table of Contents

[RecoilGun Firmware Configuration Guide 1](#_gjdgxs)

[Introduction 2](#_1fob9te)

[Configuration parameters 3](#_3znysh7)

[Weapon Definition 3](#_2et92p0)

[ID 3](#_tyjcwt)

[Trigger Mode 3](#_3dy6vkm)

[Rate of Fire 4](#_1t3h5sf)

[PowerIR1 (long-range) 4](#_4d34og8)

[PowerIR2 (short-range) 4](#_2s8eyo1)

[PowerLED1 (muzzle) 4](#_17dp8vu)

[PowerLED2 (power LED) 4](#_3rdcrjn)

[PowerMotor 4](#_26in1rg)

[FlashLED1 (muzzle) 4](#_lnxbz9)

[FlashLED2 5](#_35nkun2)

[Flash Parameter 1 5](#_1ksv4uv)

[Flash Parameter2 5](#_44sinio)

[Global parameters 6](#_2jxsxqh)

[Shot Config 6](#_z337ya)

[IR Config 7](#_3j2qqm3)

[Configuration via the Test App 7](#_1y810tw)

# Introduction

The RecoilGun firmware system has been designed to allow configuration of its behaviour.

This can be achieved statically (by defining default parameters at build time) or dynamically during the game (by sending appropriate BLE commands to alter the configuration).

The purpose of this is to allow flexibility during the design, integration and testing phases of the development, according a simple 3-steps workflow:

1. Tentative default configuration parameters are set during the firmware design phase
2. During integration and testing, configuration parameters can be altered via BLE to achieve the desired in-game behaviour. This allows rapid testing iteration without the need to go through firmware rebuild -> release -> test for each iteration.
3. Once the configuration is deemed satisfactory, the parameters can be set as defaults in the firmware for the final build before manufacturing. Dynamic configuration will still be possible, but it will no longer be required.

# Configuration parameters

The available parameters can be widely separated in 2 groups:

1. Weapon definition parameters
2. Global parameters

## Weapon Definition

The firmware supports the definition of up to 12 Weapon Types. A Weapon Type defines the behaviour of RecoilGun when firing,

At any time, only one Weapon Type is active; the app selects the active Weapon Type with a BLE command sent to the Command characteristic (beyond the scope of this document, refer to the Recoil Protocol document)

Each Weapon Type is defined by the following parameters.

### ID

The ID of the Weapon Type. This is what the app will use to select this weapon for firing.

The valid range is 0-11

### Trigger Mode

How trigger behaves when this Weapon Type is selected.

#### Plasma mode (trigger mode 0)

In this mode, RecoilGun fires on trigger **release**. While the trigger is pressed, RecoilGun “charges up” a plasma shot, consuming 4 rounds from the clip for each “rate of fire” period. When the trigger is released, a single plasma shot is fired, but it carries the number of rounds that was accumulated while the trigger was pressed,

If the clip runs out of ammo, the shot is not automatically fired; rather, RecoilGun keeps waiting for release without accumulating any more rounds.

Example: the rate of fire is set to 1 shot per second. The player presses the trigger, and 4 rounds are immediately taken from the clip and added to the plasma shot. After one second, 4 more rounds are taken from the clip and added to the shot. A fraction of a second later, the player releases the trigger, causing a single plasma shot to be fired. That plasma shot is worth 8 rounds.

#### Single Shot mode (trigger mode 1)

In this mode, RecoilGun fires a single shot on trigger **press**. The shot is worth one round. Until the trigger is released, nothing else will happen.

#### N-Burst mode (trigger mode N, with N=2 to 253)

In this mode, RecoilGun fires a single shot on trigger **press**. The shot is worth one round. However, while the trigger is pressed, RecoilGun will fire one more round per “rate of fire period” until one of the following conditions occurs:

1. The trigger is released
2. N shots have been fired, where N is the burst length defined in the burst mode
3. The clip is empty

#### Full Auto mode (trigger mode 254)

In this mode, RecoilGun fires a single shot on trigger **press**. The shot is worth one round. However, while the trigger is pressed, RecoilGun will fire one more round per “rate of fire period” until one of the following conditions occurs:

1. The trigger is released
2. The clip is empty

### Rate of Fire

This parameter sets the rate of fire for N-Burst mode, Full Auto and Plasma mode. The value is in units of 50ms, so a value of 10 will fire twice a second (once every 10\*50ms)

### PowerIR1 (long-range)

This parameter sets the power for the long-range, narrow-angle IR emitter that transmits the “bullet” when firing RecoilGun. The values are 0 (off) to 255 (100% power).

**Note**: the 0-255 scale is only used for interface consistency. Internally, this gets scaled down to 0-18 (with 18=100%), rounding down. That means that a value of 25 will actually generate a 5.5% duty cycle (25\*18/255 with truncation)

### PowerIR2 (short-range)

This parameter sets the power for the short-range, wide-angle IR emitter that transmits the “bullet” when firing RecoilGun. The values are 0 (off) to 255 (100% power).

**Note**: the 0-255 scale is only used for interface consistency. Internally, this gets scaled down to 0-18 (with 18=100%), rounding down. That means that a value of 25 will actually generate a 5.5% duty cycle (25\*18/255 with truncation)

### PowerLED1 (muzzle)

This parameter sets the power for the visible LED on the RecoilGun muzzle. This LED can be used for firing feedback according to parameter FlashLED1. The values are 0 (off) to 255 (100% power).

### PowerLED2 (power LED)

This parameter sets the power for the visible LED that signals RecoilGun power on state. This LED can be used for feedback “when being shot”, according to parameter FlashLED2. The values are 0 (off) to 255 (100% power). **Usage of this parameter is only enabled in special builds, as LED2 is meant to be a reliable indication of power ON/OFF.**

### PowerMotor

This parameter sets the time the recoil motor will run for, in units of 5ms, assuming the battery is fully charged.

The values are 0 (off) to 255 (1275ms). The recoil mechanism on current pistol prototypes requires 90ms for the spring to be fully loaded when the battery is fully charged, therefore the default is 18 (90ms)

The spring loading time is internally corrected for, based on battery voltage (as the motor is directly connected to unregulated VBAT)

### FlashLED1 (muzzle)

This sets the behaviour of the muzzle LED when firing RecoilGun. For each of the modes available, the power is set with parameter PowerLED1.

#### None (flash mode 0)

The muzzle LED is off when firing.

#### Square Wave (flash mode 1)

The muzzle LED flashes one sequence per shot fired by RecoilGun. Each flash sequence is made of N flashes, each lasting T milliseconds. N is set with FlashParam1, T is set with FlashParam2.

#### Glow (flash mode 2)

The muzzle LED goes through a complex sequence, in lockstep with the trigger of the RecoilGun.

1. When the trigger is pressed, the LED starts glowing with a period of T milliseconds. T is set with FlashParam2.
2. While the trigger is pressed, for every “rate of fire” period the glowing frequency is doubled
3. When the trigger is released, a square wave sequence is generated. The sequence is made of N flashes, each lasting T milliseconds. N is set with FlashParam1, T is set with FlashParam2.

#### Solid ON

The muzzle LED is on while the trigger is pressed.

### FlashLED2

This parameter behaves exactly like FlashLED1, but controls the Power LED. **Usage of this parameter is only enabled in special builds, as LED2 is meant to be a reliable indication of power ON/OFF.**

### Flash Parameter 1

The meaning of this parameters depends on the setting of parameters FlashLED1 and FlashLED2.

For FlashLED modes None and Solid ON, this parameter is not used.

For FlashLED mode “Square Wave”, this parameter sets the number of flashes for each shot.

For FlashLED mode “Glow”, this parameter sets the number of flashes when the trigger is released.

### Flash Parameter2

The meaning of this parameters depends on the setting of parameters FlashLED1 and FlashLED2.

For FlashLED modes None and Solid ON, this parameter is not used.

For FlashLED mode “Square Wave”, this parameter sets the duration of each flash, in units of 100ms. E.g. setting this to 2 will make the LED flash 5 times a second.

For FlashLED mode “Glow”, this parameter actually sets 2 frequencies:

1. The initial Glowing frequency, that is doubled once very “Rate of fire” interval. For this, the parameter is used as 500ms units
2. The frequency of flashes on trigger release. For this the parameter is used as 100ms units.

e.g. setting this parameter to 4 with a Glow FlahLED1 setting will make the flash start glowing with a 2 seconds period, which then becomes 1 second after 1 “rate of fire” period, then 500ms and so on, until the trigger is released and the LED flashes a square wave with a 400 ms period.

## Global parameters

Currently, there is are two global configuration group for RecoilGun: ShotConfig and IR Config

### Shot Config

The firmware supports the definition of the way shooting behave **on top of** the weapon type that is currently selected.

The Shot Config is defined by the following parameters.

#### Auto Feedback

The AutoFeedback is a composite parameter that controls the feedback of RecoilGun on the occurring of specific events. This automatic feedback replicates behaviour that could be attained via the app by sending BLE commands, but has the advantage that, being implemented by the firmware, it carries almost zero lag with respect to the event occurring.

#### AutoRecoil

When this setting is enabled, RecoilGun automatically triggers the recoil mechanism when a shot is fired. This setting is enabled by default.

#### AutoFlash on shooting

When this setting is enabled, RecoilGun automatically drives the Muzzle LED according to the Weapon Type when a shot is fired. This setting is enabled by default.

#### AutoFlash on being shot

When this setting is enabled, RecoilGun automatically drives the Power LED when a shot is received. This setting is disabled by default and **is only implemented in special builds, as LED2 is meant to be a reliable indication of power ON/OFF.**

#### Trigger Mode Override

This parameter allows the app to override the trigger mode in the currently selected Weapon Type. This allows setting LED behaviour by weapon type, while having trigger control from the app.

Specifying a value for this makes RecoilGun use this parameter instead of the one specified in the currently selected Weapon Type.

The default is 255, which means “Use the trigger mode defined by the weapon type” (i.e. by default, overriding is disabled)

### IR Config

The firmware supports the definition of the way IR transmission and reception behave **on top of** the weapon type that is currently selected. These parameters have the potential of breaking the functionality of IR emitters and sensors, so deviations from the default settings should be thoroughly tested.

The IR Config is defined by the following parameters.

#### TX Repeats

The number of times each “bullet” packet is repeated over IR. This increases the chance of successful reception in case there is IR noise or the player is being shot simultaneously by more than one opponent. Default is 2.

Be aware that increasing this can disrupt shooting functionality: if the total transmission time for all repeats is longer that the rate of fire period, bullets will overlap and cancel each other out.

The current default of 2 is optimised for the scenario where the maximum rate of fire of any gun in the game is 10Hz and no more than 2 opponents are assumed to be shooting at a player from the same side.

#### TX Flags

Miscellaneous flags to control TX behaviour. At the moment, only one flag is implemented:

1. Randomise TX time. Set by default. Enables randomisation of the interval between repetitions of the same “bullet” over IR. Together with TXRepeats>1, this makes it less likely that 2 players shooting contemporaneously at the same target will cancel each other out.

#### RX Enable

Mask to enable or disable each individual IR sensor. Default 0xF (all sensors enabled)

1. 0x01 = enable sensor 0 (Gun)
2. 0x02 = enable sensor 1 (Gun)
3. 0x03 = enable sensor 2 (Gun)
4. 0x04 = enable sensor 3 (Clip-on)

#### Clip-On Sensor Check Interval

Period between tests of the presence the Clip-on sensor, in 500ms units.

Beware that:

1. Due to behaviour of the IR sensor, absence is only declared after 3 consecutive tests fail.
2. Presence is detected immediately
3. Each test disables the clip-on sensor for a few milliseconds. Setting this too low increases the probability that one shot is not received

# Configuration via the Test App

The Test App as committed on <https://hg-server-win-1.hotgen.uk-ltd:8443/svn/SkyRocketToys/Recoil> allows the configuration of the firmware via BLE since revision 11932.

The button “SEND CONFIG” will transmit a configuration table according to this doc.

The values in the configuration table are hardcoded in the TestApp and can be edited in the file **RecoilGunConfig.cs**, method **RecoilGunConfig::RecoilGunConfig()**

The following is an example configuration (mirroring the default config of the firmware), you can used this as a starting point for fine tuning of the gun behaviour.

public RecoilGunConfig()

{

weapons = new List<WeaponDefinition>();

WeaponDefinition auxWeapon;

// Single shot

auxWeapon = new WeaponDefinition();

auxWeapon.id = 0;

auxWeapon.triggerMode.mode = TriggerMode\_mode.SINGLE;

auxWeapon.triggerMode.burstLen = 1;

auxWeapon.rateOfFire = rateOfFire\_ms(1000);

auxWeapon.powerIR1 = 255;

auxWeapon.powerIR2 = 25;

auxWeapon.powerLED1 = 255;

auxWeapon.powerLED2 = 255;

auxWeapon.powerMotor = 255;

auxWeapon.flashLED1 = FlashMode.SQUARE\_WAVE;

auxWeapon.flashLED2 = FlashMode.NONE;

auxWeapon.flashParam1 = 1;

auxWeapon.flashParam2 = squareWavePeriod\_ms(300);

weapons.Add(auxWeapon);

// Burst

auxWeapon = new WeaponDefinition();

auxWeapon.id = 1;

auxWeapon.triggerMode.mode = TriggerMode\_mode.BURST;

auxWeapon.triggerMode.burstLen = 5;

auxWeapon.rateOfFire = rateOfFire\_ms(500);

auxWeapon.powerIR1 = 255;

auxWeapon.powerIR2 = 25;

auxWeapon.powerLED1 = 255;

auxWeapon.powerLED2 = 255;

auxWeapon.powerMotor = 255;

auxWeapon.flashLED1 = FlashMode.SQUARE\_WAVE;

auxWeapon.flashLED2 = FlashMode.NONE;

auxWeapon.flashParam1 = 5;

auxWeapon.flashParam2 = squareWavePeriod\_ms(300);

weapons.Add(auxWeapon);

// Full Auto

auxWeapon = new WeaponDefinition();

auxWeapon.id = 2;

auxWeapon.triggerMode.mode = TriggerMode\_mode.FULL\_AUTO;

auxWeapon.triggerMode.burstLen = 1;

auxWeapon.rateOfFire = rateOfFire\_ms(500);

auxWeapon.powerIR1 = 255;

auxWeapon.powerIR2 = 25;

auxWeapon.powerLED1 = 255;

auxWeapon.powerLED2 = 255;

auxWeapon.powerMotor = 255;

auxWeapon.flashLED1 = FlashMode.SQUARE\_WAVE;

auxWeapon.flashLED2 = FlashMode.NONE;

auxWeapon.flashParam1 = 4;

auxWeapon.flashParam2 = squareWavePeriod\_ms(300);

weapons.Add(auxWeapon);

// Plasma

auxWeapon = new WeaponDefinition();

auxWeapon.id = 3;

auxWeapon.triggerMode.mode = TriggerMode\_mode.PLASMA;

auxWeapon.triggerMode.burstLen = 1;

auxWeapon.rateOfFire = rateOfFire\_ms(2000);

auxWeapon.powerIR1 = 255;

auxWeapon.powerIR2 = 25;

auxWeapon.powerLED1 = 255;

auxWeapon.powerLED2 = 255;

auxWeapon.powerMotor = 255;

auxWeapon.flashLED1 = FlashMode.GLOW;

auxWeapon.flashLED2 = FlashMode.NONE;

auxWeapon.flashParam1 = 15;

auxWeapon.flashParam2 = glowPeriod\_ms(2000);

weapons.Add(auxWeapon);

// Shot config

shotConfig = new ShotConfig();

shotConfig.autoRecoil = true;

shotConfig.autoFlashOnFiring = true;

shotConfig.autoFlashOnBeingShot = false;

shotConfig.triggerModeOverride.mode = TriggerMode\_mode.NO\_OVERRIDE;

shotConfig.triggerModeOverride.burstLen = 1;

// IR Config

irConfig = new IRConfig();

irConfig.randomTXTimes = true;

irConfig.txRepeats = 2;

for (int i = 0; i < 4; i++)

{

irConfig.sensorEnable[i] = true;

}

irConfig.clipOnCheckInterval = cliponcheck\_ms(3000);

}