JB Gun 4 IR

4 IR led points light gun system

User Guide

JB Game Lab

Made by Jean-Baptiste Bongrand

Table of contents

Introduction

I started this project in 2019 with one goal in mind; making an affordable home LightGun solution that would offer an experience as close as possible from the arcade, while not sacrificing any feature. I wanted it to be customizable for any gamer, any setup.

After hundreds of hours of work, and a lot of testing and help from the passionated members of this community, we finally reached this goal, together.

Of course the story doesn't stop here, I will keep this project alive, improve it, make it bigger, better.

All of that wouldn't be possible without you, who use and support my project.

Thank you. And have a great lightgun experience!

***Before starting, a small obligatory disclaimer:** like with any DIY project, I am not responsible for any damage you might do to your hardware/yourself. Be sure to read everything carefully before using my firmware. I am not a professional in electronics. I am giving all the schematics and pics as examples, use them at your own risks.*
And of course, this firmware cannot be sold alone nor in a package/hardware, it is completely free. If some people sold it to you, you were scammed.

Features List

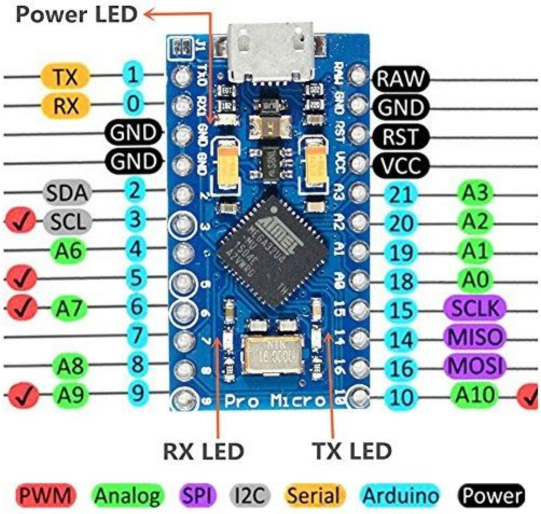
- **Perfect line of sight accuracy:** thanks to the 4 leds system, a ton of advanced math and pseudo 3D space calculation that does auto calibration and tracking in real time, you get a perfect line of sight accuracy all the time.
- **One time calibration:** calibrate the camera sensor and leds once, and then forget it, the aiming will still work perfectly no matter the angle, position, or if you disconnect your gun.
- **Ultra low latency:** this system uses a fast IR camera, and the firmware is heavily optimized, reducing the total processing latency to an average of 4ms (2ms ~ 7ms). Lowest latency of all modern lightgun systems.
- **No special software needed:** everything is handled by the arduino, making it plug and play with any system that supports a mouse/keyboard or a controller input. No extra software needed (the GUI is optional).
- **No external processing:** no processing needed on the host platform, no cpu overhead, no overlay added to the game screen. You can use your games as usual.
- **USB and Bluetooth Mouse and Controller compatible:** since it's using standard HID mouse, gamepad and bluetooth, it's compatible with everything that supports a usb/bt mouse & gamepad.
- **Reduced minimum distance:** thanks to the powerful tracking, this system allows you to play closer to the screen than most other modern systems, and even more if you add a wide lens or fisheye lens to the camera (any smartphone lens should work, even the cheap ones).
- **Support every kind of screen:** you can use this system pretty much on any screen type/ratio/size, it will just work.
- **Full offscreen tracking/reload:** it keeps track of your aiming even outside of the screen, and supports various offscreen options like offscreen reload.
- **Full feedback support:** you can add a solenoid, rumble motor and RGB led to your gun, and fully configure and control how each of them behaves and reacts to your games. It supports various functions like full auto and synchronisation with ingame feedback (for supported games).
- **Auto reload support:** you can activate it at any time to automatically reload your gun after 6 shoots.
- **Nunchuck support:** you can plug a nunchuck controller if you need more buttons.
- **Setting saved inside the gun:** each guns has its own memory, to save independent settings and customize each one to your liking.
- **Quick mode switching with mode button:** you can add a “Mode” button to your gun to triggers calibration when holding, or to quickly switch various modes:
    - Content mode (fullscreen/4:3).
    - Offscreen mode.
    - Input mode.
    - Pedal mode.
    - Auto reload mode.
- **Full serial command support:** you can use serial commands to set every gun mode, and sync the game feedback with the gun feedback.
- **Fully featured GUI for supporters:** for the supporters of this project, there is a full GUI with a lot of options and tools to configure every aspect of the gun.

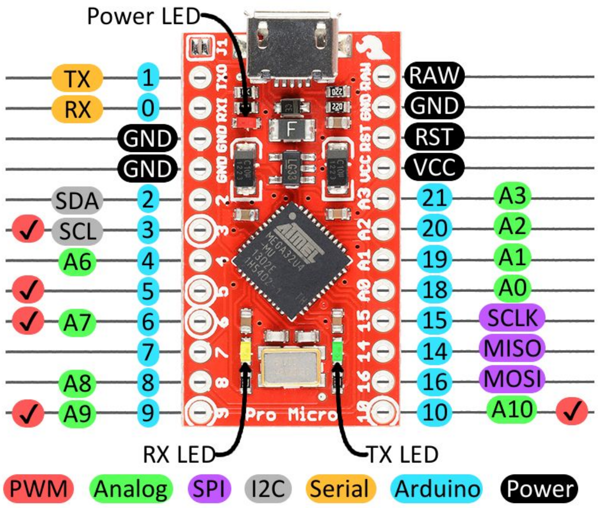
Building the hardware

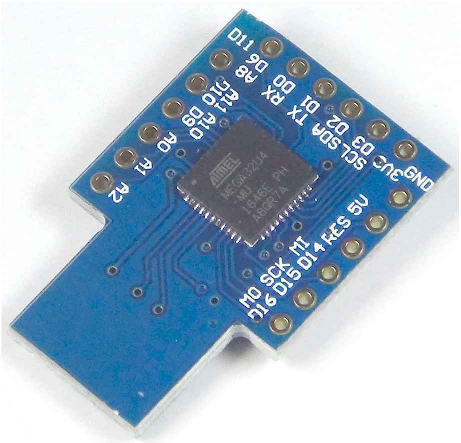
1. Needed components

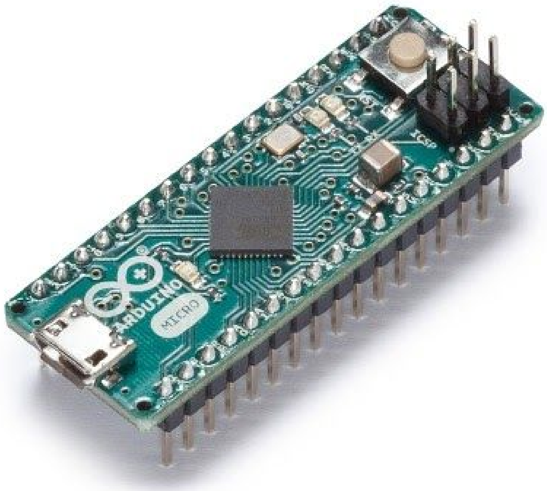
For the gun:

- Any old gun shell (lighgun of some airsoft) that has enough space to fit the camera and other components inside. It also needs to have buttons/switches of course.
- Arduino Pro Micro ATMega32U4 5V/16MHz (3.3V or 8MHz won't work), the smaller Pro Micro beetle (has less buttons/functions but is way smaller, be careful to get the model with the D6 and 3v3) or any other board with the same ATMega32U4 chipset. Some examples of boards that will work:









- Micro USB cable long enough for you to play comfortably (e.g. 3m)
- DFRobot IR Camera Here or a home made IR camera (with Wiimote sensor)
- Electronic cables to connect everything (e.g. kynar cables)
- Soldering iron & solder
- Hot glue or other methods to prevent anything from moving inside the gun
- (Optional) A push button with long shaft for the special "Mode" button

For the screen LEDs:

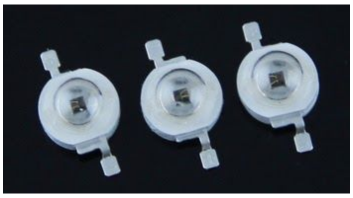
For the LEDs system, there is a wide range of options for any price and setup, the main 3 are as followed:

- **Cheapest solution -** works very well for smaller screens and basic setup, but isn't as reliable as the other option on bigger screens and from higher distance/angle;
  - 940nm IR LEDs (at least 3 leds per point, high power and angle around 20~40 degrees recommended)

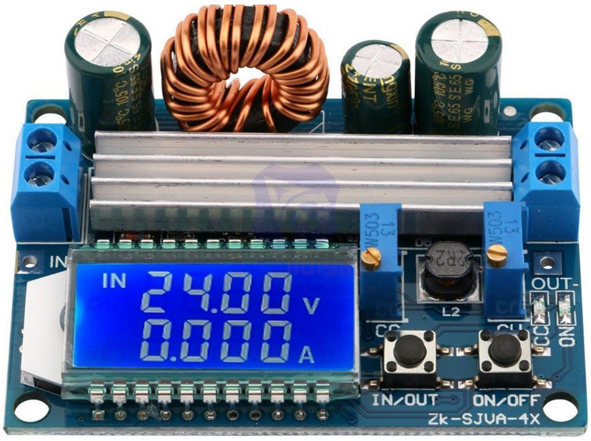
  - Resistors fitting the setup (see more about that in the next chapter)

  - USB cable

- **High power solution -** great for big screens and playing from distance, but not cheap, and requires a lot of precision and method when setting it up;
  - High angle 940nm smd LEDs (1 led/point, 1W 60~90 degrees leds recommended)

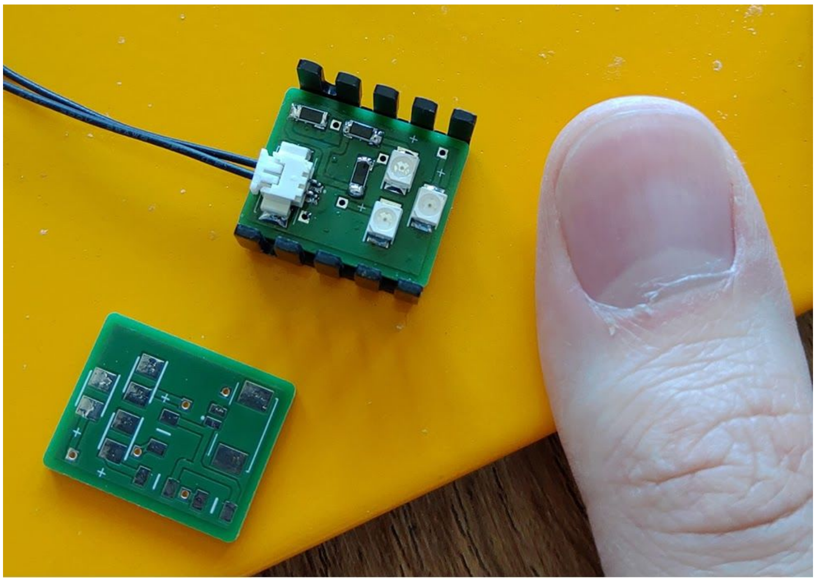


  - Voltage AND current regulator



  - Power supply strong enough to power it.

- **SMD pcb solution -** the most balanced and clean solution, but also a bit on the expensive side, and requires more advanced soldering skills (more info on that in the discord server, design and image by *Boojakascha*).



(Optional) For recoil solenoid:

- N-Channel logic gate mosfet (e.g. IRL540 or similar)
- Kickback diode (e.g. 1N4001 or similar)
- 1kOhms and 100kOhms Resistors
- Small pcb
- Solenoid that fits inside the shell without hitting anything when activated (or gun integrated solenoid)
- AC power supply matching the solenoid specs
- Power cable long enough to supply the solenoid

(Optional) For rumble feedback (or motor driven recoil):

- NPN transistor (e.g. PN2222 or similar)
- kickback diode (e.g. 1N4001 or similar)
- 270ohms~1Kohms resistor
- 0.1 uF ceramic capacitor
- Small pcb
- Rumble motor
- Power supply matching the specs of the motor (can be the arduino VCC if the rumble is 5V and doesn't need too much current)

(Optional) For RGB Led:

- RGB LED with common cathode (common ground)
- 3 resistors that match the LED specs

(Optional) For the Bluetooth:

There are 2 options for the bluetooth:

- The more expensive RN-42 bluetooth module
- The cheaper HC-05 bluetooth module, that needs to be hacked with the RN-42 firmware (tutorial for that is found in the discord server)
- 5v battery
- Voltage regulator and charger for the 5v battery

(Optional) Extra hardware:

- Temperature sensor for solenoid speed control (TMP36)
- A level shifter and/or voltage converter depending on what you want to build
- Corrugated Tube to hide the cables and prevent them from bending too much or get damaged.
- Trigger button microswitch mod

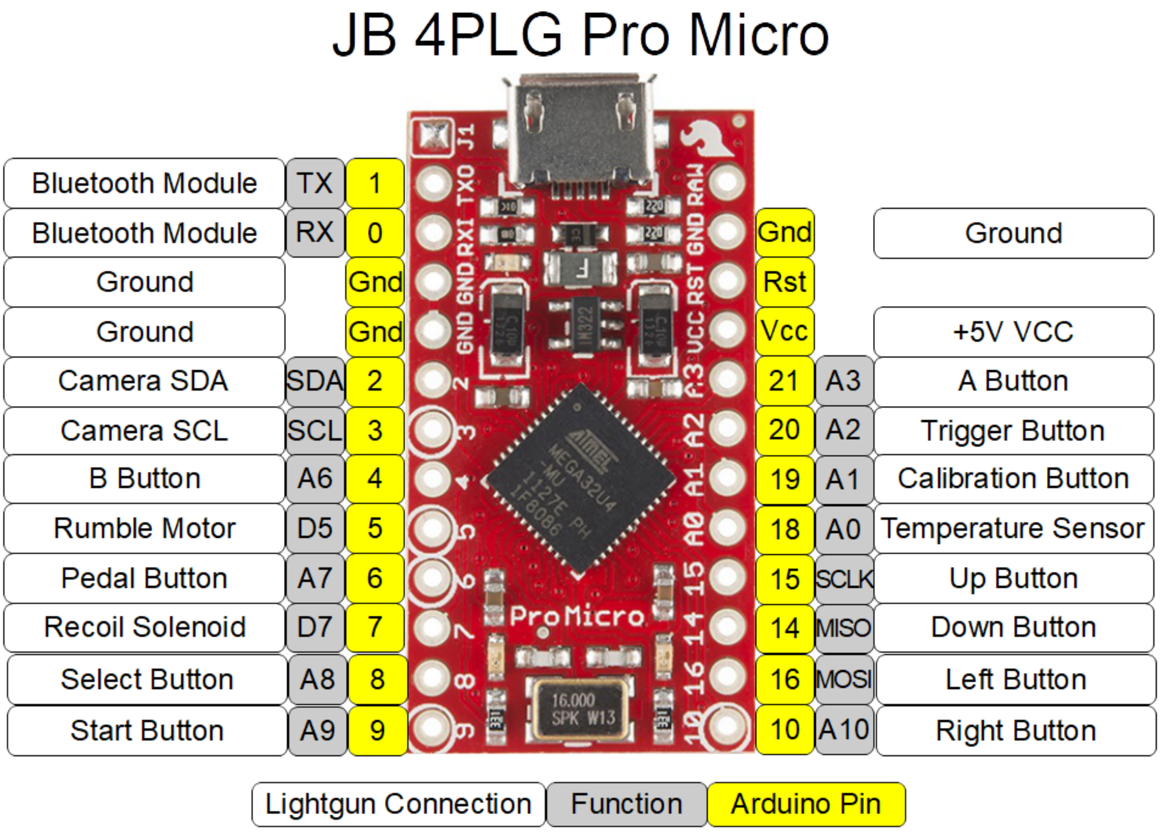
2. Connection to the arduino

First recommended thing to do is connecting the main buttons and the camera to the arduino, and test if everything works as expected. Then once everything is done, you can connect the optional things, like the feedbacks, temperature sensor, bluetooth...

Here is the pins listing, with each pin connection and function: (pins in red are the ones changing between RGB LED and DPad mode)

	Pro Micro with DPad	Pro Micro with RGB Led	Arduino Micro	Functionality	Mouse/Keyboard button	Gamepad button
Mode button	19 (A1)	19 (A1)	A1	calibration and modes	-	-
Trigger button	20 (A2)	20 (A2)	A2	gun trigger	L mouse button	Button 0
A button	21 (A3)	21 (A3)	14	reload/cover	R mouse button	button 1
B button	4 (A6)	4 (A6)	15	extra button	M mouse button	button 2
Pedal button	6 (A7)	15	-	Pedal	R mouse button	button 1
Select button	8 (A8)	8 (A8)	-	Select	Kb 5 key	button 3
Start button	9 (A9)	14	-	Start	Kb 1 key	button 4
Down button	14 (MISO)	not used	-	dpad Down	Kb Down arrow	dpad Down
Up button	15 (SCLK)	not used	-	dapd Up	Kb Up arrow	dpad Up
Left button	16 (MOSI)	not used	-	dpad Left	Kb Left arrow	dpad Left
Right button	10 (A10)	not used	-	dpad Right	Kb Right arrow	dpad Right
RGB LED Red	not used	6 (A7)	6 (A8)	RGB LED red pin	-	-
RGB LED Green	not used	9 (A9)	9 (A10)	RGB LED green pin	-	-
RGB LED Blue	not used	10 (A10)	10 (A11)	RGB LED blue pin	-	-
Recoil output	7	7	16	Recoil output pin	-	-
Rumble output	5	5	5	Rumble output pin	-	-
Temp sensor	18 (A0)	19 (A0)	A0	Solenoid temperature sensor	-	
BT module output	TX & RX	TX & RX	TX & RX	Bluetooth module connection	-	-

Here are the references for the corresponding pins for the main board models:



JB 4PLG Pro Micro

Lightgun Connection	Function	Arduino Pin	Arduino Pin	Function	Lightgun Connection
Bluetooth Module	TX	1			
Bluetooth Module	RX	0	Gnd		Ground
Ground		Gnd	Rst		
Ground		Gnd	Vcc		+5V VCC
Camera SDA	SDA	2	21	A3	A Button
Camera SCL	SCL	3	20	A2	Trigger Button
B Button	A6	4	19	A1	Calibration Button
Rumble Motor	D5	5	18	A0	Temperature Sensor
Pedal Button	A7	6	15	SCLK	Up Button
Recoil Solenoid	D7	7	14	MISO	Down Button
Select Button	A8	8	16	MOSI	Left Button
Start Button	A9	9	10	A10	Right Button

Lightgun Connection | Function | Arduino Pin

ProMicro
TXO RXI GND GND 2 3 4 5 6 7 8 9
RAW GND RST VCC A3 A2 A1 A0 15 14 16 10
MEGA32U4
16.000

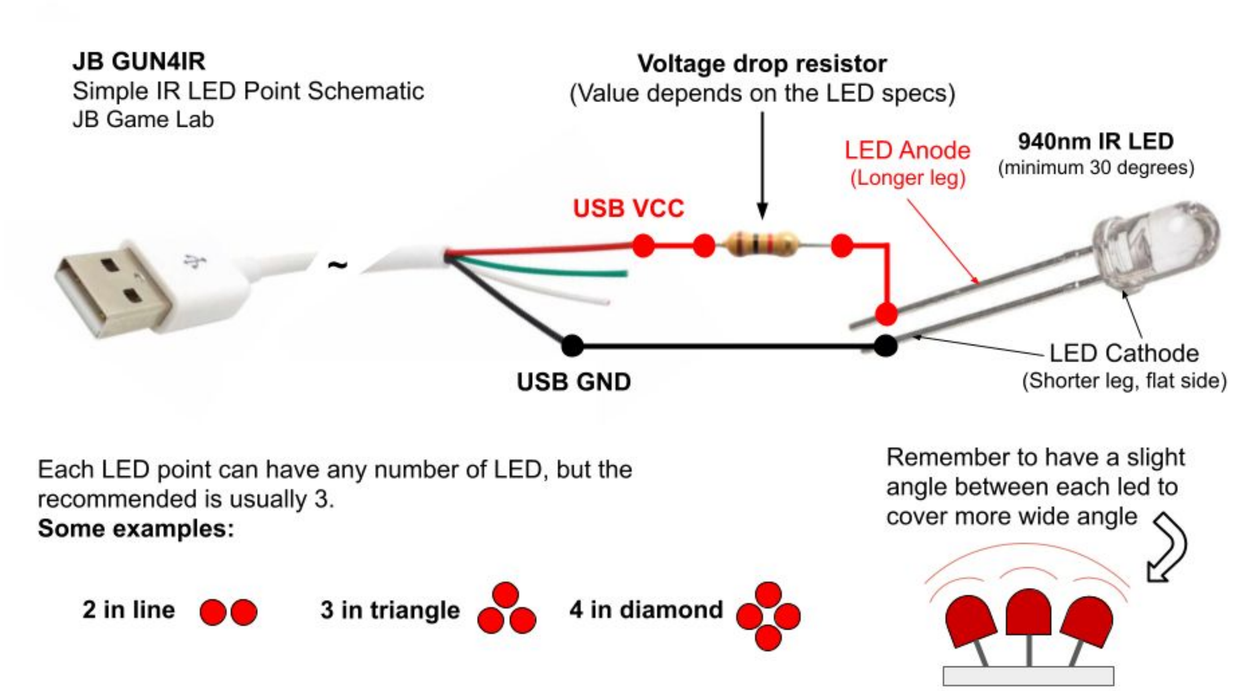
IMPORTANT NOTES:

- Each button must be connected between GND and the corresponding pin.
- Be careful of the pinout difference if you use the rgb led or dpad (it’s detected automatically by the firmware)
- You don't have to use every pin, just use the ones you want and leave the other pins free.
- Connect the camera to pin GND, VCC, 2 (SDA), and 3 (SCL)
- The DFRobot IR Cam often has the sensor tilted, so you might have to tilt the camera on left or right to make it work properly. Please adjust it with the test tool in the GUI before closing your gun.
- The pedal and A button have the same function by default, but it can be changed in the GUI or by buttons combo
- When firing offscreen, the gun trigger will use the secondary button (A button) instead. It is used for offscreen reload. It can be disabled with a button combo

3. IR LED points

Low cost IR system

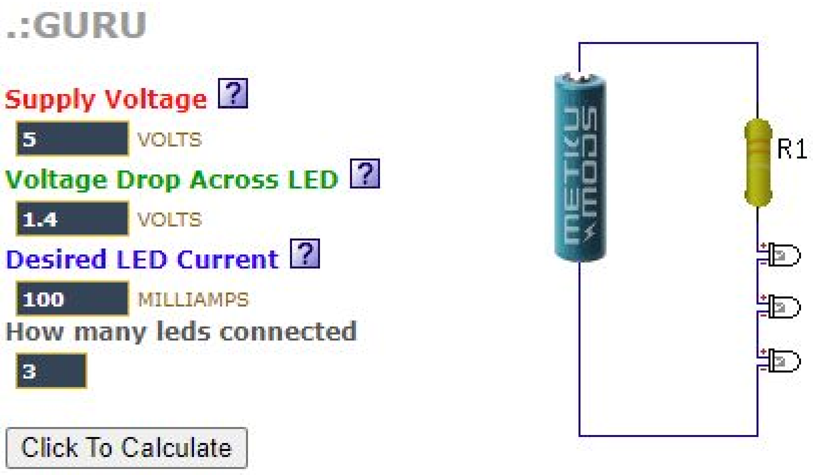
For this setup, I recommend putting them in a triangle with a slight angle on each side, to cover more angle with the IR light.
You can calculate the resistor needed and the power supply here:
http://ledcalc.com/#calc



Each LED point can have any number of LED, but the recommended is usually 3.
Some examples:

Remember to have a slight angle between each led to cover more wide angle

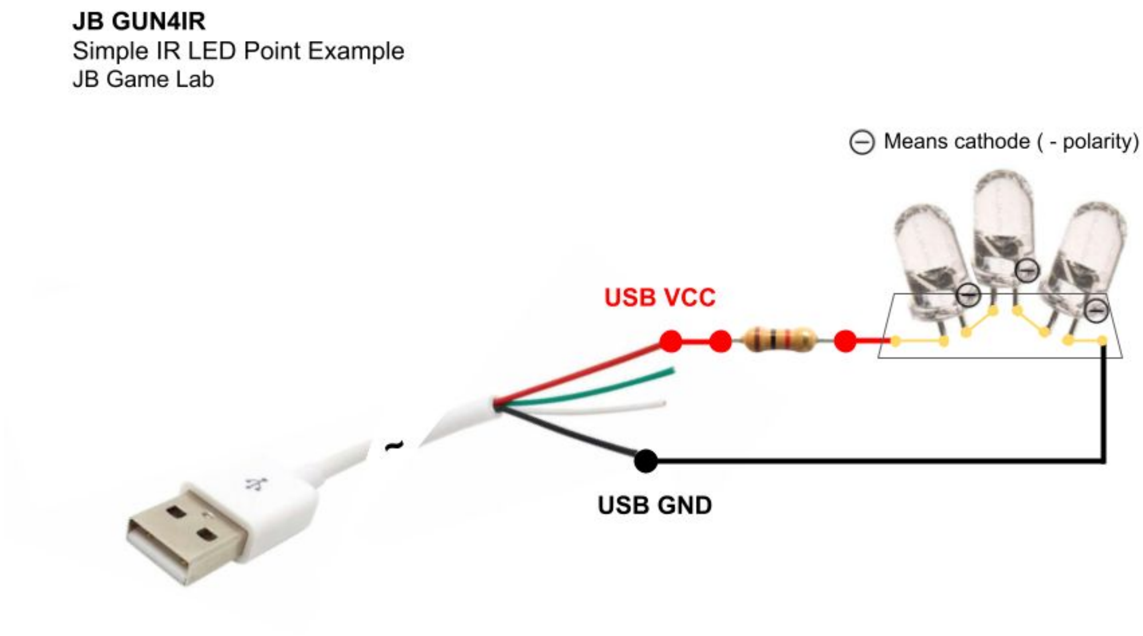
Note that for multiple leds per point, you can put them in series if the voltage drop isn't too high.
So for a triangle setup you can have something like this (it's just an example with made up values):



.:GURU

Supply Voltage ?
5 VOLTS
Voltage Drop Across LED ?
1.4 VOLTS
Desired LED Current ?
100 MILLIAMPS
How many leds connected
3

Which for our example setup becomes this:



Note that this image is made to be very basic and easy to understand, in reality you would need to put the leds as close as possible from each other.

A small 2 point led system I did back in the beginning (not recommended anymore, I just show as example);

High powered led

This setup is for experienced users only, as it can be dangerous for you and your hardware if setup/used incorrectly.

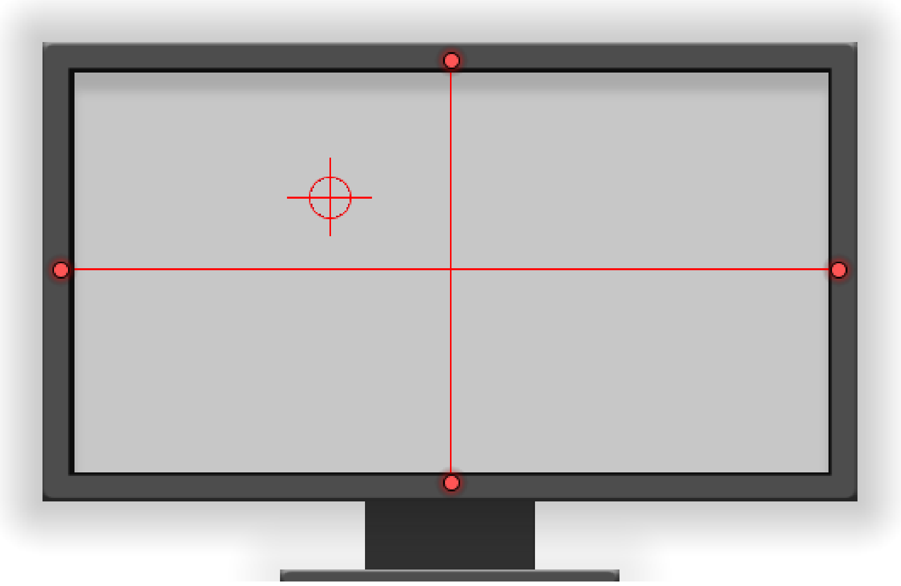
Here is the steps you need to follow:

- decide how you want to connect your leds (all in series, 2 by 2 in parallel or all in parallel), and calculate the voltage and current you need for the power supply to be set to with this: http://ledcalc.com/#calc
- connect the power converter to the power supply
- turn the potentiometers to get the correct voltage OUT, and minimum current, to be sure you don't blow the led when plugged
- either turn off the OUT of the power converter, or unplug it

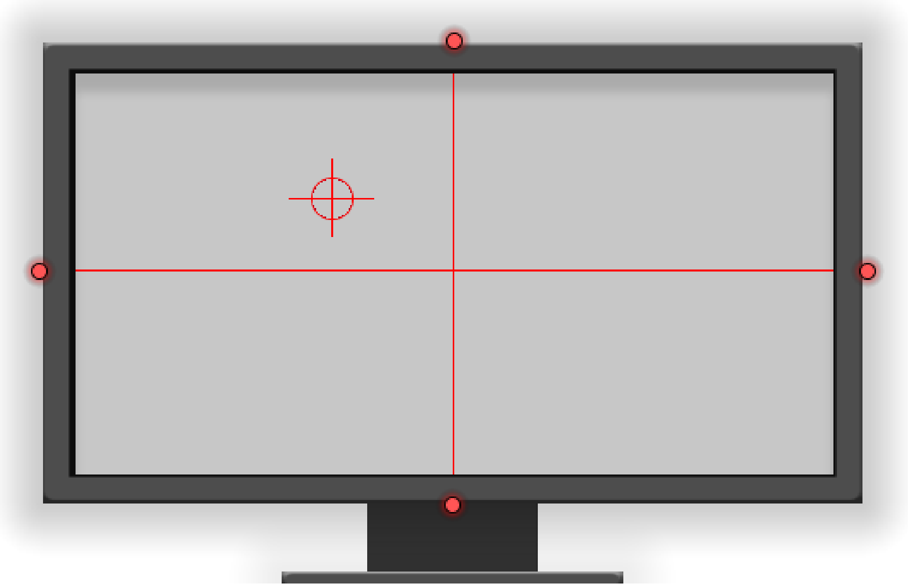
- connect the leds as you first decided (all of them, is some aren't connected the specs will change
- switch it on
- set correct the current
- now if the leds get too hot, reduce the current a bit. If the leds are not powerful enough, you can play with the voltage and current. But be extra careful, if not doing correctly you can easily burn the leds, and put fire/melt everything touching them, including yourself.

LEDs placement

Place the LEDs at the exact center of each side (up, down, left, right), aligned as well as possible, to have a good accuracy. You can use the GUI test tool, like that:

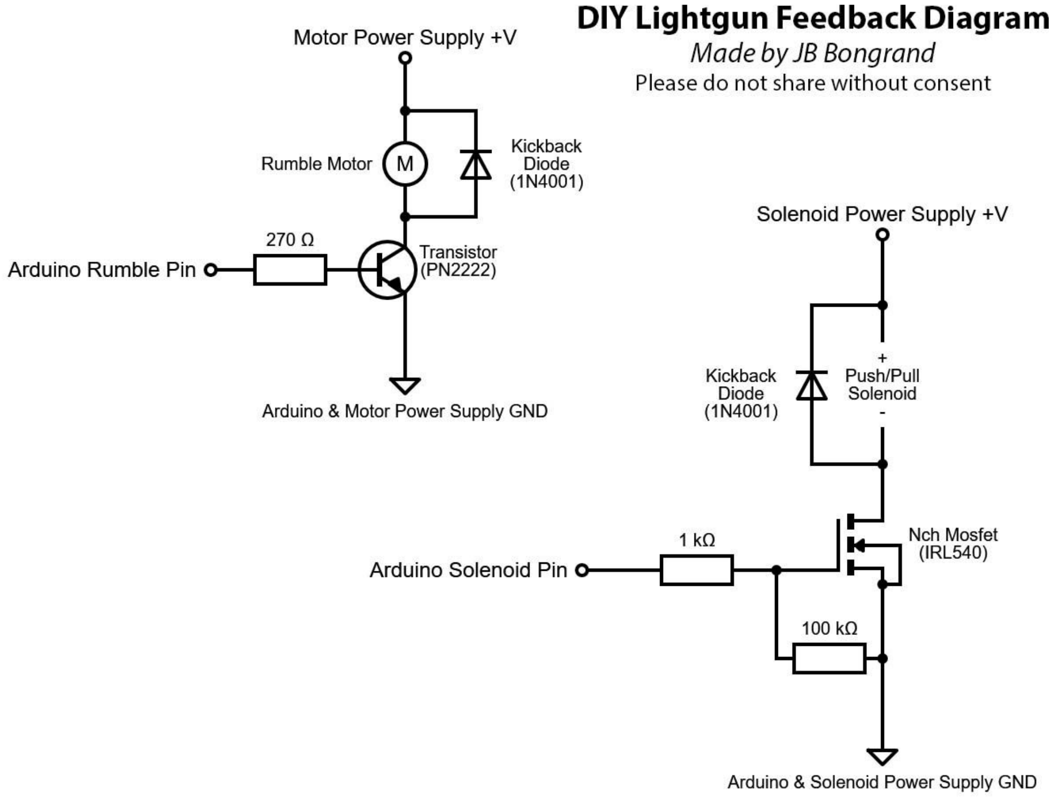


You can also attach the leds on the sides on the screen, and calibrate the screen borders afterward, this way;

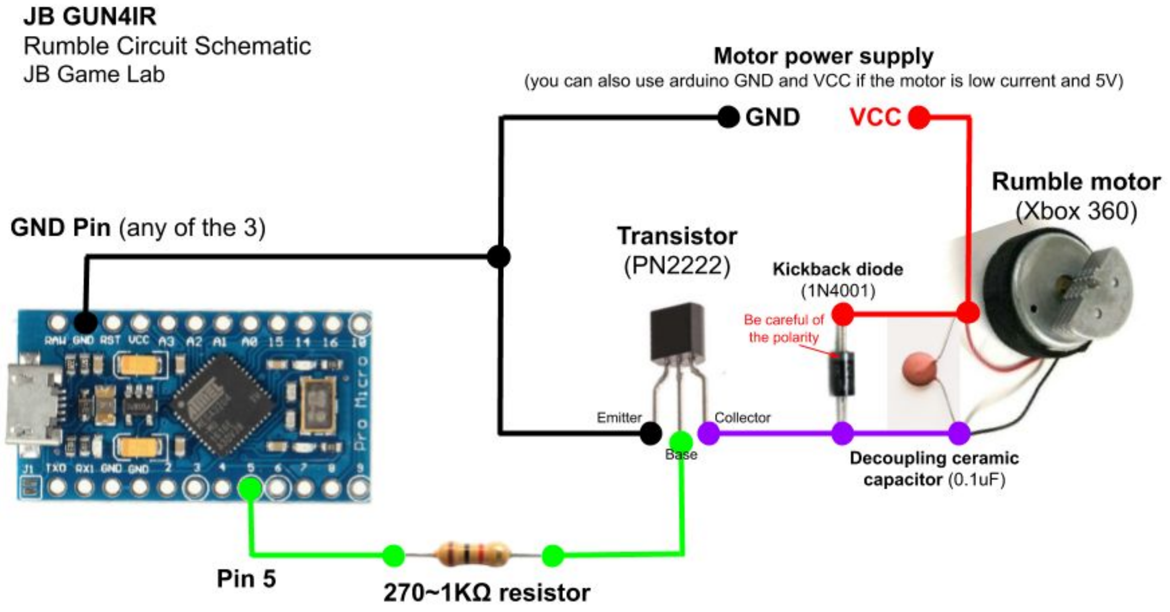


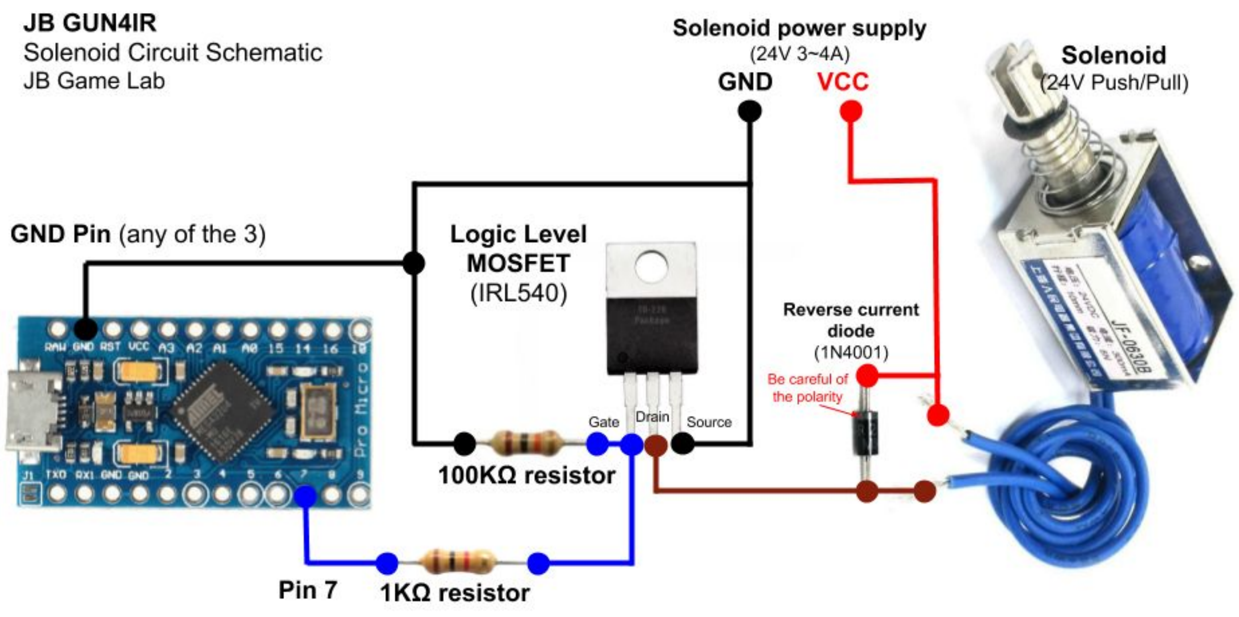
4. Feedbacks circuits

Here is a simple schematics of the motors and solenoids control circuits (motor on the right, solenoid on the left):



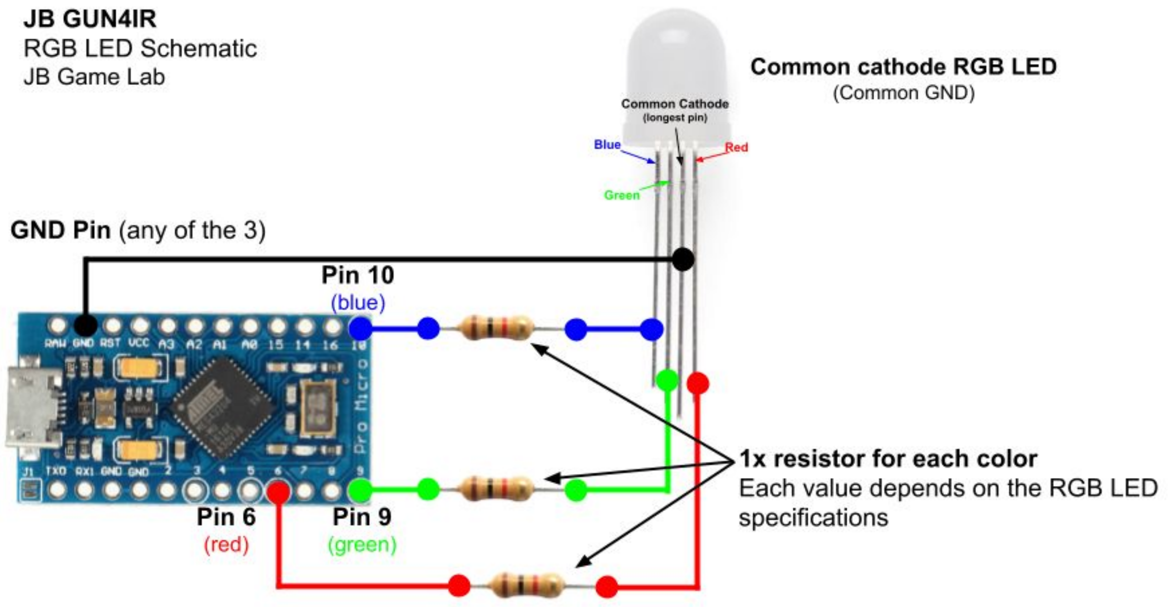
I also did some easier to understand schematics:





Of course you might need to adapt each schematics to the specs of your components. Pinout for instance differs between transistor and MOSFETs.

The RGB LED is more simple to use; first connect the common cathode to the GND, then each pin color to a resistance matching the led specs (check the links in the led point system chapter), and each of those resistors to the corresponding arduino pin.



5. Bluetooth

THIS PART IS FOR ADVANCED USERS ONLY
Notes: the bluetooth module is very limited compared to USB HID connection, it has only gamepad mode support, and has no serial connection or connection to the GUI. It also has higher input latency than the USB mode.

Now if you still want to do it, here is a small guide to make it work:

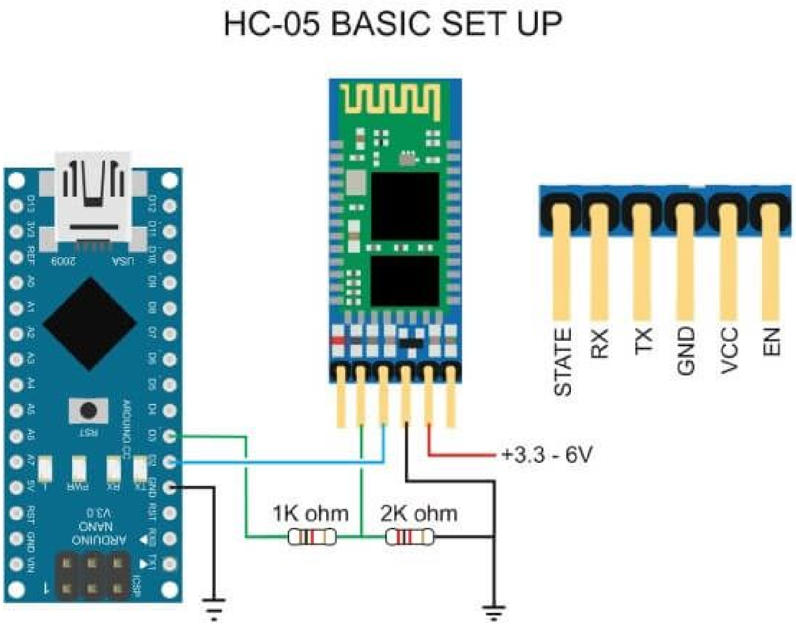
You will either need a RN-42 board, or a HC-05 hacked with the RN-42 firmware (they both have the same chipset).

Follow this guide for the HC-05, from the beginning to the end;
https://www.instructables.com/Upgrade-Your-3-Bluetooth-Module-to-Have-HID-Firmwa/

For both RN-42 and hacked HC-05, you need to connect the BT board to any Arduino **(don't forget it's 3v3 logic, so if your arduino is 5v you will need a voltage divider or a level shifter)**, use the test sketch available in the guide and configure the board the same way as written, with one difference; send SH,0210 instead of SH,0230 at the end to activate the gamepad mode instead of mouse/keyboard.

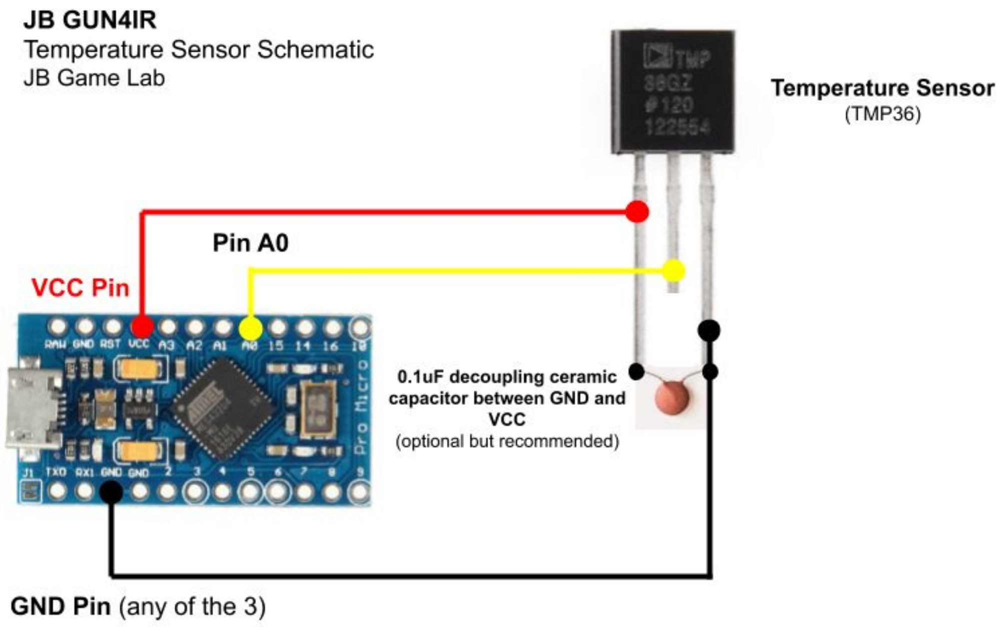
Once the board is configured correctly, you need to connect it to the gun's arduino as followed **(don't forget it's 3v3 logic, so you will need a voltage divider or a level shifter);**
Arduino TX (pin 0) => BT board RX
Arduino RX (pin 1) => BT board TX
And connect the VCC and GND.

Of course if you want to use it wirelessly, you need to add a 5v battery(3v3 won't work with the pro micro 16MHz), and a regulator.



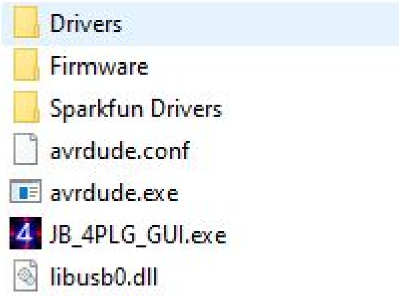
6. Extra hardware

The temperature sensor needs to be connected this way;


JB GUN4IR
Temperature Sensor Schematic
JB Game Lab
Temperature Sensor
(TMP36)
TMP
36GZ
#120
122554
Pin A0
VCC Pin
RAW GND RST VCC A3 A2 A1 A0 15 14 16 10
J1 TXO RXI GND GND 2 3 4 5 6 7 8 9
Pro Micro
0.1uF decoupling ceramic capacitor between GND and VCC
(optional but recommended)
GND Pin (any of the 3)

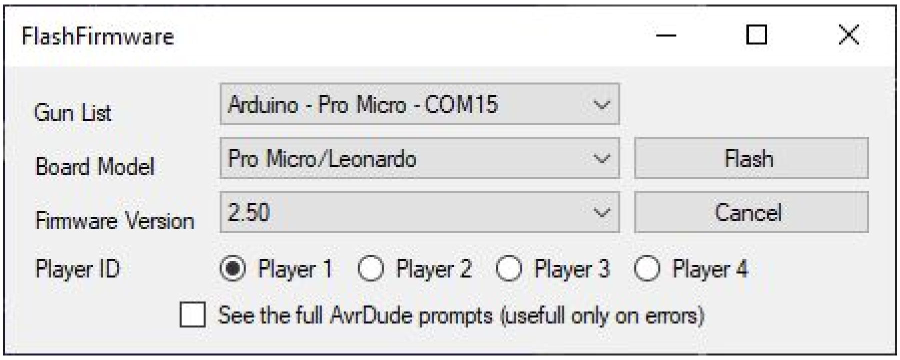
Installation

The initial installation can be done on a Windows PC only!
Once installed you can then use the gun on any machine/OS.



Drivers
Firmware
Sparkfun Drivers
avrdude.conf
avrdude.exe
JB_4PLG_GUI.exe
libusb0.dll

1. Download and unzipped the GUI files somewhere in your computer
2. Connect the arduino to the computer
3. If it asks you a drivers, install the drivers from the drivers folder
4. Open the GUI (you will need a license)
5. Go to Tool => Flash Firmware:



FlashFirmware

Gun List: Arduino - Pro Micro - COM15
Board Model: Pro Micro/Leonardo
Firmware Version: 2.50
Player ID: ● Player 1 ○ Player 2 ○ Player 3 ○ Player 4
□ See the full AvrDude prompts (usefull only on errors)

6. Select your board, the arduino model (it should be automatic), the firmware version you want to flash, and the player you want to assign the gun to (note that this last parameter is only useful for Mame to have a fixed gun assignment)
7. Flash the arduino
8. Now that the arduino is flashed, I recommend using the test functions to check if everything works correctly
9. Once everything is setup and working, you can now change each gun configuration to your liking

Serial commands usage (Mamehooker, launchers…)

Each of those commands can be sent by serial to the arduino through any application that can handle serial. *Note that in my examples I used the character ‘.’ to separate every number, but you can actually use other characters not used by my firmware, like ‘x’ for instance.*

Commands to change modes:

Feedback management:

Start serial mode (to force the fb to be only triggered by serial):

Sx (x = feeback number; 0 - solenoid, 1 - rumble, 2 - LED red, 3 - LED green, 4 - LED blue, 6 - all feedbacks)

End serial mode for all feedback: E

Various modes: Mx.y (x = mode number, y = mode option)

Mode	Mode number	Option 0	Option 1	Option 2	Default option
Input	0	Mouse/Keyboard	Gamepad	Hybrid	0
Offscreen shot	1	Disabled	Lower left corner shot	Reload button	1
Pedal	2	Separate button	Reload button		1
Game aspect ratio	3	Fullscreen	4:3		0
Temperature sensor	4	Disabled	Enabled		0
Autoreload	5	Disabled	Enabled		0
Rumble only mode	6	Disabled	Enabled		0
Fullauto	8	Disabled	Auto	Always On	1

Ingame feedback trigger commands:

Feedback command: Fx.y.zzz (x = feedback number, y = state, 0 - off, 1 - on, 2 - pulses, zzz = feedback strength 0~255 or number of pulses)

Feedback	Number
Solenoid	0
Rumble	1
RGB Led Red	2
RGB Led Green	3
RGB Led Blue	4

Various things to note;

- The strength is only used with rumble and led feedbacks
- For safety, the solenoid is always in pulse mode, no matter the command used
- The pulse mode has 2 ways of working, you can use 0 for the number to trigger a pulse only if there are no pulses triggered at the moment, or 1~255 to add pulses to the buffer that will be triggered as soon as the feedback is ready.

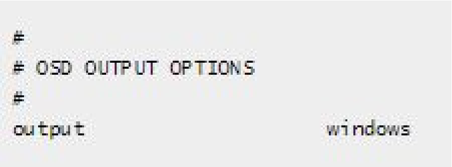
So for instance to add 2 pulses to the the rumble motor buffer, write "F1.2.2"

You can send those commands with any serial communication apps, but the most useful so far is mamehooker

MameHooker usage:

To use Mamehooker with this system, first follow those steps:

- In your Mame.ini file (in Mame folder), change "output" option to "windows";



```
#
# OSD OUTPUT OPTIONS
#
output                    windows
```

- Change the COM port of your gun 1 and 2 to be respectively COM1 and COM2 (you can do that in devices manager).
- Make sure Mamehooker is correctly configured for your system and detects the emulators you want to use (follow the instructions on its website).
- To send commands from mamehooker, you need to use the com communication function (Com write)

Gun usage

Aiming

After plugging in your lightgun, simply aim the screen from moderate distance, you should see your cursor starting to follow your motion right away.
If nothing happens, there might be something wrong with your LEDs, check them with your phone digital cam if you can see them bright enough from where you are aiming.

The tracking works down to one visible LED, and will track your aiming even offscreen, but keep in mind that the more LED points the camera can see at the same time, the better the precision of tracking will be.

Initial IR cam offset and leds placement calibration

On the first use, your aim might be off centered.
It is normal, each camera sensor has a slight offset.
Also if your LEDs are not close from the screen, you might have inaccuracies on the sides.

To correct both of those issues, you need to perform a simple 5 steps calibration.
You can start the calibration process by either holding the “mode” button until the mouse cursor snaps to the center of the screen, or using the GUI calibration function.

There are 5 steps in the calibration, for each of them you have to aim at the cursor and fire.
To avoid any erroneous calibration, calibration will advance to the next step only if there are no misdetections with the leds. Rumble and RGB LED will notify you if the step succeeds or fail.

Your sensor should now be calibrated, and since it's saved to the Arduino EEPROM, you won't have to do it ever again, even when changing your screen setup.

Offscreen shooting

There are 2 options available for how the gun behaves when shooting offscreen; shoot at the lower left corner of the screen, or trigger the A button (Right mouse click) instead of the trigger, useful for games that need offscreen shoot to reload.

Button combo mode changing

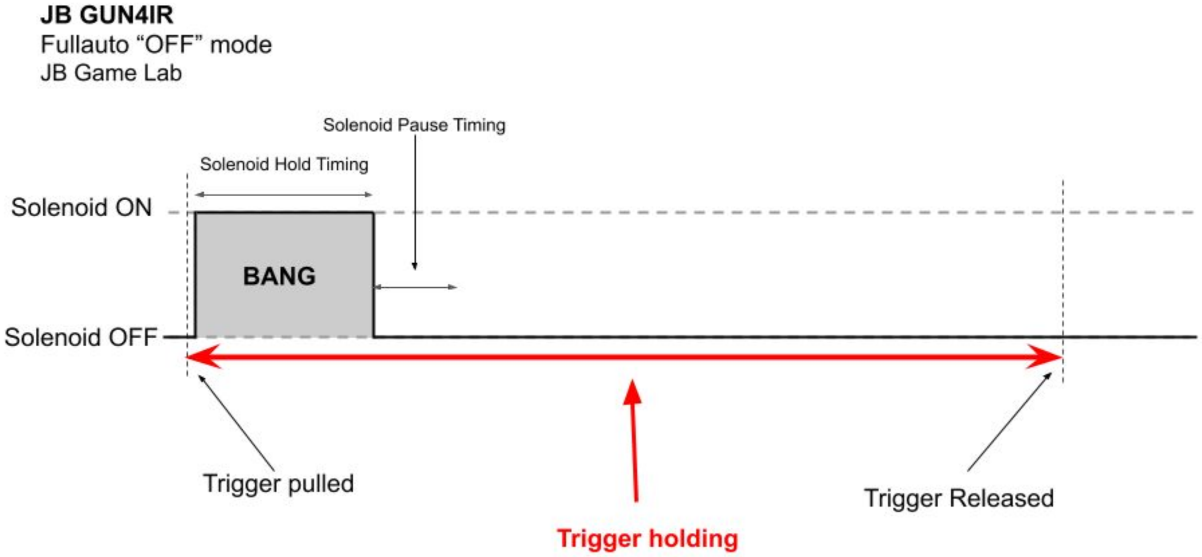
There are different modes you can change with buttons combo, **you must first hold one of the following buttons and press the “mode” button briefly for some to switch mode**. If you have a rumble motor, it will rumble briefly to tell you which mode it activated.

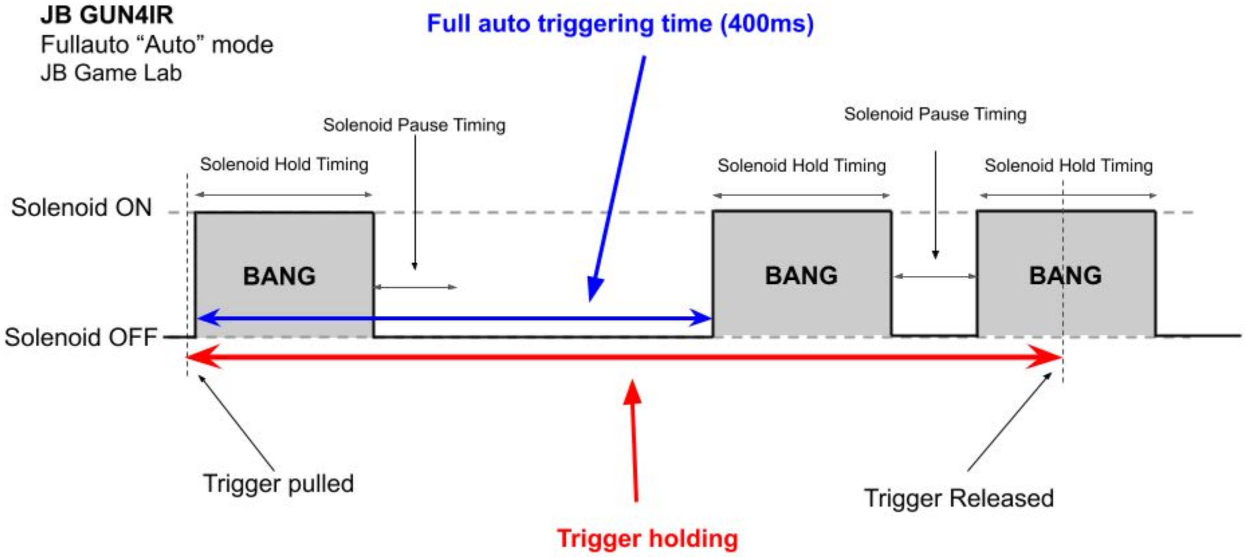
- Mode button alone: enable/disable 4:3 mode
- Trigger+Mode button: offscreen shoot mode (disable/aim at the corner/reload button)
- A+Mode button: input mode
- Pedal+Mode button: pedal button mode (unique/reload)
- Trigger+A+Mode button: enable/disable auto reload (reloads after 6 shots)

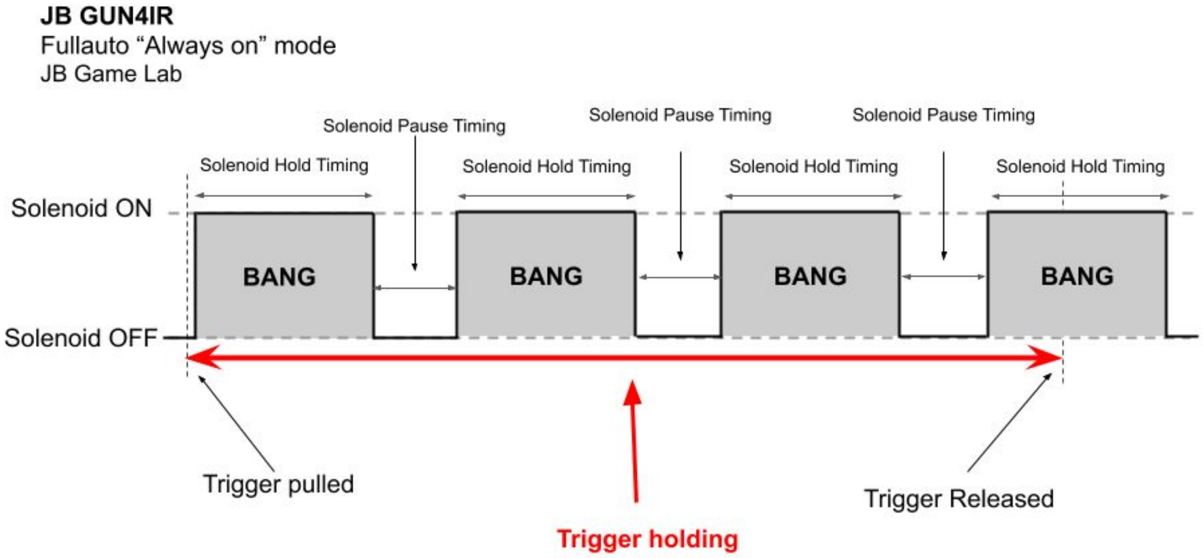
Fullauto Modes

There are 3 available fullauto modes for feedbacks, Off, Auto or Always On.
Here are 3 graphics showing how each mode works.

Note that each feedback has its own timings for better feeling.






JB GUN4IR
Fullauto "Always on" mode
JB Game Lab
Solenoid Pause Timing
Solenoid Pause Timing
Solenoid Pause Timing
Solenoid Hold Timing
Solenoid Hold Timing
Solenoid Hold Timing
Solenoid Hold Timing
Solenoid ON
BANG
BANG
BANG
BANG
Solenoid OFF
Trigger pulled
Trigger Released
Trigger holding

Troubleshooting

Problem: The movements of my cursor are reversed or messed up.
Solution: Your camera is either in a wrong orientation or your calibration settings are erroneous.

Problem: The aim is not working at all.
Solution: Pushing buttons work? Yes => issue with your LEDs. No => wrong gun settings or connection problem with your Arduino.

Problem: The aim is suddenly messed up.
Solution: The cam has trouble seeing your LEDs correctly, or the gun settings are wrong.

Problem: The aim is working in the center of the screen, but on the sides it's not aiming properly.
Solution: You are not in the right screen mode (fullscreen or 4:3). Don't forget to set up your aspect ratio in the GUI.

Problem: The gun doesn't work correctly in multi screen configuration.
Solution: It's not a limitation of my system, but a limitation of how mouse input works. You can fix the issue by either using an application that uses exclusive fullscreen, or using the gamepad mode instead of mouse mode.

Problem: Some or all feedbacks don't work.
Solution: Does it work when using the GUI testing button? Yes => check your gun settings. No => check your circuit and the arduino pins.

Problem: My arduino isn't detected by the GUI and/or fails to be flashed with the firmware.
Solution: Check that the correct drivers are installed for the arduino. They are all included in the package I provide. If needed, install Arduino IDE and flash the arduino as a "Arduino Leonardo".

Problem: I got some file errors when trying to flash the firmware.
Solution: do you have a 32 bits OS? If yes, please replace the needed files in the main folder by the ones in the drivers folder. If it's an avrdude issue, you can replace the avrdude.exe by the one you get when installing arduino IDE.