

# E2 Counter

From GMod Wiki

## Contents

- [1 A Simple Counter Made Using Expression 2](#)
- [2 What the code means](#)
- [3 Wiring instructions](#)
- [4 Optimizations](#)

## A Simple Counter Made Using Expression 2

In this tutorial We'll be teaching you how to make a simple counter, using expression 2, that will display the count on a wire screen.

- **Function** - An Expression 2 Counter.
- **What it Does** - Displays a count on a wire screen.
- **Notes** - This tutorial is essential to follow the next tutorial [E2 Colour Changer](#)

### Introduction -

What this contraption does is using two wire buttons (or other wire inputs), counts up or down, and display this output on a wire screen. Counters are very useful and can be used for a variety of things, changing light colours, counting players, displaying game scores, the list is endless.

### The Expression 2 Code

```
@name Counter
@inputs Add Subtract
@outputs Screen
@persist
if(Add == 1 & Screen >= 0 & Screen <1000)
{
    Screen+= 1
}
if(Subtract == 1 & Screen > 0 & Screen <=1000)
{
    Screen-= 1
}
```

## What the code means

The first line of code defines the Expression 2 chips name. This can be anything you want but if you plan to use them on public servers give them a meaningful name!

Line 2 Defines the Chips inputs ; In this case I've defined two inputs, add, which will be used to add to the count, and subtract, which will be used to subtract from the count.

Line 3 Defines outputs ; Only one output is needed for the counter, I've named it Screen, as this is what will be wired to it. Naming inputs and outputs in this way makes them easier to wire up!

Line 4 Defines persists ; Persists are basically internal inputs and outputs that are not wired up, for the purposes of this we don't need any, so don't worry about them.

The previous 4 lines are required in every expression, otherwise you get error messages. The next lines are the actual unique code for your expression.

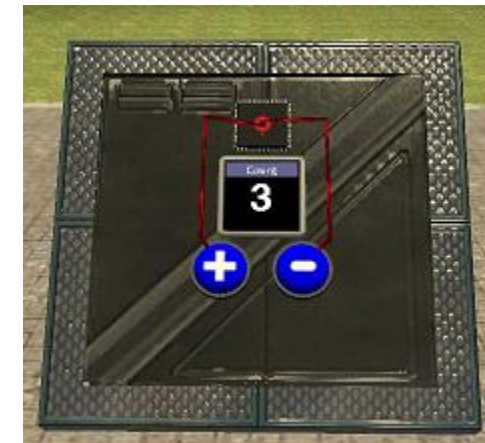
Line 5 and 6 at a glance look almost the same but there are three key differences I will explain in greater detail shortly, first let me explain "if" statements. An "if" statement is very similar to an "if then else" gate. [Wired Gate/Selection](#). What they does is look to see if the first value is true. This is the section in the curved bracket (). If it's true it will output the value contained within the first set of pointed brackets {} and if it is not true output the value in the second set of pointed brackets {}. A handy tip, if you want the third value to be 0, you don't need to include the second set of pointed brackets {}.

So in line 5 we are asking, if ADD is 1 (the add button is pressed)and the value displayed on the screen is GREATER or equal to zero to add one to the count. If not it won't add to the count.

In line 6 we are asking if SUBTRACT is 1 (the subtract button is pressed) and the value displayed on the screen is GREATER than 0, and less than or equal to 1000, to subtract one from the count.

## Wiring instructions

- 1 - Open your wire menu and select your expression 2 tool.
- 2 - Select "new expression".
- 3 - Copy and Paste the above code into the box that appears.
- 4 - Click save and exit, don't forget to give it a meaningful name
- 5 - Click udate then highlight the name you added in step 4.
- 6 - Close your menu.
- 7 - Left click on a prop. (A PHX 1x1 is good.)
- 8 - Spawn two wire buttons, non toggle not checked, on value 1, off value 0.
- 9 - Take out your wiring tool.
- 10 - Wire Add of the expression 2 to the add button.
- 11 - Wire Subtract of the expression 2 to the subtract button.
- 12 - Spawn a wire screen.
- 13 - Wire a of the screen to the expression 2.
- 14 - Play around with your counter and gloat to your friends who can't use expression 2. XD



## Optimizations

To keep server pings low, it is highly recommended to optimize any code you create, especially in heavily scripted environments like wiremod. Using some binary math the above code can be simplified to this:

### The Optimized Expression 2 Code

```
@name Counter
@inputs Add Subtract
@outputs Screen
@persist
Screen += Add-Subtract
```

Besides exchanging two branches and six comparisons for a simple subtraction, it also remedies the anomalous situation in which both inputs are active. There is a small change with the handling of numbers outside of [0,1000]. This can be remedied by adding "clamp(Screen, 0, 1000)" at the end.

I edited to its lined up, // xelander Thanks for that, // Headshot119 Went ahead and added an Optimization section. there should be an article on this kind of stuff.. // Ghostwo I changed the code sections so it's in the wiki code format. // Marabunta2048

Retrieved from "[http://wiki.garrysmod.com/?title=E2\\_Counter](http://wiki.garrysmod.com/?title=E2_Counter)"

Category: [Wire Addon Tutorials](#)

- This page was last modified on 13 July 2011, at 12:58.
- This page has been accessed 15,346 times.

- [Privacy policy](#)
- [About GMod Wiki](#)
- [Disclaimers](#)