

# In-kernel API for FPGA Programming

## Overview

The in-kernel API for FPGA programming is a combination of APIs from FPGA manager, bridge, and regions. The actual function used to trigger FPGA programming is `fpga_region_program_fpga()`.

`fpga_region_program_fpga()` uses functionality supplied by the FPGA manager and bridges. It will:

- lock the region's mutex
- lock the mutex of the region's FPGA manager
- build a list of FPGA bridges if a method has been specified to do so
- disable the bridges
- program the FPGA using info passed in `:c:expr:'fpga_region->info'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master)(Documentation) (driver-api) (fpga) fpga-programming.rst, line 18); [backlink](#)**

Unknown interpreted text role "c:expr".

- re-enable the bridges
- release the locks

The struct `fpga_image_info` specifies what FPGA image to program. It is allocated/freed by `fpga_image_info_alloc()` and freed with `fpga_image_info_free()`

## How to program an FPGA using a region

When the FPGA region driver probed, it was given a pointer to an FPGA manager driver so it knows which manager to use. The region also either has a list of bridges to control during programming or it has a pointer to a function that will generate that list. Here's some sample code of what to do next:

```
#include <linux/fpga/fpga-mgr.h>
#include <linux/fpga/fpga-region.h>

struct fpga_image_info *info;
int ret;

/*
 * First, alloc the struct with information about the FPGA image to
 * program.
 */
info = fpga_image_info_alloc(dev);
if (!info)
    return -ENOMEM;

/* Set flags as needed, such as: */
info->flags = FPGA_MGR_PARTIAL_RECONFIG;

/*
 * Indicate where the FPGA image is. This is pseudo-code; you're
 * going to use one of these three.
 */
if (image is in a scatter gather table) {

    info->sgt = [your scatter gather table]

} else if (image is in a buffer) {

    info->buf = [your image buffer]
    info->count = [image buffer size]

} else if (image is in a firmware file) {

    info->firmware_name = devm_kstrdup(dev, firmware_name,
                                      GFP_KERNEL);

}
```

```

/* Add info to region and do the programming */
region->info = info;
ret = fpga_region_program_fpga(region);

/* Deallocate the image info if you're done with it */
region->info = NULL;
fpga_image_info_free(info);

if (ret)
    return ret;

/* Now enumerate whatever hardware has appeared in the FPGA. */

```

## API for programming an FPGA

- `fpga_region_program_fpga()` - Program an FPGA
- `fpga_image_info()` - Specifies what FPGA image to program
- `fpga_image_info_alloc()` - Allocate an FPGA image info struct
- `fpga_image_info_free()` - Free an FPGA image info struct

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master) (Documentation) (driver-api) (fpga) fpga-programming.rst, line 92)**

Unknown directive type "kernel-doc".

```

.. kernel-doc:: drivers/fpga/fpga-region.c
   :functions: fpga_region_program_fpga

```

### FPGA Manager flags

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master) (Documentation) (driver-api) (fpga) fpga-programming.rst, line 97)**

Unknown directive type "kernel-doc".

```

.. kernel-doc:: include/linux/fpga/fpga-mgr.h
   :doc: FPGA Manager flags

```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master) (Documentation) (driver-api) (fpga) fpga-programming.rst, line 100)**

Unknown directive type "kernel-doc".

```

.. kernel-doc:: include/linux/fpga/fpga-mgr.h
   :functions: fpga_image_info

```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master) (Documentation) (driver-api) (fpga) fpga-programming.rst, line 103)**

Unknown directive type "kernel-doc".

```

.. kernel-doc:: drivers/fpga/fpga-mgr.c
   :functions: fpga_image_info_alloc

```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\driver-api\fpga\linux-master) (Documentation) (driver-api) (fpga) fpga-programming.rst, line 106)**

Unknown directive type "kernel-doc".

```

.. kernel-doc:: drivers/fpga/fpga-mgr.c
   :functions: fpga_image_info_free

```