

```

/*
 * Copyright (c) 2013-2015, NVIDIA CORPORATION. All rights reserved.
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms and conditions of the GNU General Public License,
 * version 2, as published by the Free Software Foundation.
 *
 * This program is distributed in the hope it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
 * more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

#ifndef __OV5647_H__
#define __OV5647_H__

#include <media/nvc.h>
#include <media/nvc_image.h>

#define OV5647_IOCTL_SET_MODE _IOW('o', 1, struct ov5647_mode)
#define OV5647_IOCTL_SET_FRAME_LENGTH _IOW('o', 2, __u32)
#define OV5647_IOCTL_SET_COARSE_TIME _IOW('o', 3, __u32)
#define OV5647_IOCTL_SET_GAIN _IOW('o', 4, __u16)
#define OV5647_IOCTL_GET_STATUS _IOR('o', 5, __u8)
#define OV5647_IOCTL_SET_BINNING _IOW('o', 6, __u8)
#define OV5647_IOCTL_TEST_PATTERN _IOW('o', 7, \
    enum ov5647_test_pattern)
#define OV5647_IOCTL_SET_GROUP_HOLD _IOW('o', 8, struct ov5647_ae)
/* IOCTL to set the operating mode of camera.
 * This can be either stereo , leftOnly or rightOnly */
#define OV5647_IOCTL_SET_CAMERA_MODE _IOW('o', 10, __u32)
#define OV5647_IOCTL_SYNC_SENSORS _IOW('o', 11, __u32)
#define OV5647_IOCTL_GET_FUSEID _IOR('o', 12, struct nvc_fuseid)
#define OV5647_IOCTL_SET_HDR_COARSE_TIME _IOW('o', 13, struct ov5647_hdr)
#define OV5647_IOCTL_READ_OTP_BANK _IOWR('o', 14, \
    struct ov5647_otp_bank)
#define OV5647_IOCTL_SET_CAL_DATA _IOW('o', 15, \
    struct ov5647_cal_data)
#define OV5647_IOCTL_GET_EEPROM_DATA _IOR('o', 20, __u8 *)
#define OV5647_IOCTL_SET_EEPROM_DATA _IOW('o', 21, __u8 *)
#define OV5647_IOCTL_GET_CAPS _IOR('o', 22, struct nvc_imager_cap)
#define OV5647_IOCTL_SET_POWER _IOW('o', 23, __u32)

#define OV5647_INVALID_COARSE_TIME -1

/*
//NO EEPROM!
#define OV5647_EEPROM_ADDRESS 0x50
#define OV5647_EEPROM_SIZE 1024
#define OV5647_EEPROM_STR_SIZE (OV5647_EEPROM_SIZE * 2)
#define OV5647_EEPROM_BLOCK_SIZE (1 << 8)
#define OV5647_EEPROM_NUM_BLOCKS \
    (OV5647_EEPROM_SIZE / OV5647_EEPROM_BLOCK_SIZE)
*/

#define OV5647_OTP_LOAD_CTRL_ADDR 0x3D20
#define OV5647_OTP_BANK_SELECT_ADDR 0x3D84

#define OV5647_OTP_BANK_START_ADDR 0x3D00
#define OV5647_OTP_BANK_END_ADDR 0x3D1F
#define OV5647_OTP_NUM_BANKS (1)
#define OV5647_OTP_BANK_SIZE \
    (OV5647_OTP_BANK_END_ADDR - OV5647_OTP_BANK_START_ADDR + 1)
#define OV5647_OTP_SIZE \
    (OV5647_OTP_BANK_SIZE * OV5647_OTP_NUM_BANKS)
#define OV5647_OTP_STR_SIZE (OV5647_OTP_SIZE * 2)

```

```
#define OV5647_FUSE_ID_OTP_START_ADDR    0x3D00
#define OV5647_FUSE_ID_OTP_BANK 0
#define OV5647_FUSE_ID_SIZE      8
#define OV5647_FUSE_ID_STR_SIZE (OV5647_FUSE_ID_SIZE * 2)
```

```
#define OV5647_FRAME_LENGTH_ADDR_MSB      0x380E
#define OV5647_FRAME_LENGTH_ADDR_LSB      0x380F
#define OV5647_COARSE_TIME_ADDR_1         0x3500
#define OV5647_COARSE_TIME_ADDR_2         0x3501
#define OV5647_COARSE_TIME_ADDR_3         0x3502
#define OV5647_COARSE_TIME_SHORT_ADDR_1 0x3506
#define OV5647_COARSE_TIME_SHORT_ADDR_2 0x3507
#define OV5647_COARSE_TIME_SHORT_ADDR_3 0x3508
#define OV5647_GAIN_ADDR_MSB              0x350A
#define OV5647_GAIN_ADDR_LSB              0x350B
#define OV5647_GROUP_HOLD_ADDR            0x3208
```

```
struct ov5647_mode {
    int res_x;
    int res_y;
    int fps;
    __u32 frame_length;
    __u32 coarse_time;
    __u32 coarse_time_short;
    __u16 gain;
    __u8 hdr_en;
};
```

```
struct ov5647_ae {
    __u32 frame_length;
    __u8 frame_length_enable;
    __u32 coarse_time;
    __u32 coarse_time_short;
    __u8 coarse_time_enable;
    __s32 gain;
    __u8 gain_enable;
};
```

```
struct ov5647_fuseid {
    __u32 size;
    __u8 id[16];
};
```

```
struct ov5647_hdr {
    __u32 coarse_time_long;
    __u32 coarse_time_short;
};
```

```
struct ov5647_otp_bank {
    __u32 id;
    __u8 buf[16];
};
```

```
struct ov5647_cal_data {
    int loaded;
    int rg_ratio;
    int bg_ratio;
    int rg_ratio_typical;
    int bg_ratio_typical;
    __u8 lenc[62];
};
```

```
/* See notes in the nvc.h file on the GPIO usage */
```

```
enum ov5647_gpio_type {
    OV5647_GPIO_TYPE_PWRDN = 0,
    OV5647_GPIO_TYPE_RESET,
};
```

```
struct ov5647_eeprom_data {
    struct i2c_client *i2c_client;
```

```
    struct i2c_adapter *adap;
    struct i2c_board_info brd;
    struct regmap *regmap;
};

struct ov5647_power_rail {
    struct regulator *dvdd;
    struct regulator *avdd;
    struct regulator *dovdd;
};

struct ov5647_regulators {
    const char *avdd;
    const char *dvdd;
    const char *dovdd;
};

struct ov5647_platform_data {
    unsigned cfg;
    unsigned num;
    const char *dev_name;
    unsigned gpio_count; /* see nvc.h GPIO notes */
    struct nvc_gpio_pdata *gpio; /* see nvc.h GPIO notes */
    struct nvc_imager_static_nvc *static_info;
    bool use_vcm_vdd;
    int (*probe_clock)(unsigned long);
    int (*power_on)(struct ov5647_power_rail *);
    int (*power_off)(struct ov5647_power_rail *);
    const char *mclk_name;
    struct nvc_imager_cap *cap;
    struct ov5647_regulators regulators;
    bool has_eeprom;
    bool use_cam_gpio;
};

#endif /* __OV5647_H__ */
```