

```
/*
 * ov5647_mode_tbls.h - ov5647 sensor mode tables
 *
 * Copyright (c) 2015-2016, 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_TABLES__
#define __OV5647_TABLES__

#include <media/camera_common.h>

#define OV5647_TABLE_WAIT_MS 0
#define OV5647_TABLE_END 1
#define OV5647_MAX_RETRIES 3

#define ov5647_reg struct reg_8

static const ov5647_reg ov5647_reset[] = {
// {0x4202, 0x0F}, /* streaming on */
    {0x0103, 0x01},
    {OV5647_TABLE_WAIT_MS, 10},
    {OV5647_TABLE_END, 0x00}
};

static const ov5647_reg ov5647_start[] = {
    {0x0100, 0x01},
    {OV5647_TABLE_END, 0x00}
};

static const ov5647_reg ov5647_stop[] = {
    {0x0100, 0x00},
    {OV5647_TABLE_END, 0x00}
};

static const ov5647_reg tp_colorbars[] = {
    {0x503D, 0x80},
    {0x0601, 0x01},

    {OV5647_TABLE_END, 0x00}
};

static const ov5647_reg mode_1920x1080[] = {

    {0x0103, 0x01}, //Software Reset
    {OV5647_TABLE_WAIT_MS, 10},
    {0x0100, 0x00}, /* Power Down Mode, Keep all registers, just don't output anything*/

    //Start Test Pattern
/*
    {0x503D, 0x80},
    {0x503E, 0x80},
    {0x5046, 0x80},
*/

    /* Clock Configuration */
    {0x3034, 0x1A}, /* PLL Control 0: PLL Charge Pump, MIPI Bit Mode: 10 */
    {0x3035, 0x21}, /* PLL Control 1: System Clock Divider 2, Scale Divider (1)*/
    /* Sys clock divider: 2 */
}
```

```
{0x3036, 0x62}, /* PLL Multiplier: 98 */
//{0x3036, 0x31}, /* PLL Multiplier: 49 */ //XXX: This didn't work (?? TOO SLOW ??)
{0x303C, 0x11}, /* PLL Control 2 PLL CP, PLL SYS DIV = 1*/
/* Clock = 24000 / 2 / 1 * 98 = 1.176 GHz */

{0x3106, 0xF5}, /* SRB Control: PLL Clock Divider / 4, Reset Arbiter: 0, SCLK Arb: 0 */

// //Undocumented
// {0x3827, 0xEC}, /* ??? */
// {0x370C, 0x03}, /* ??? */
// {0x3612, 0x5B}, /* ??? */
// {0x3618, 0x04}, /* ??? */

//ISP Control
{0x5000, 0x06}, /*ISP Control: Lens Correction: False, Defective Pixel Cancel BC_EN, WC_EN: True, True
*/
{0x5002, 0x40}, /*ISP Control: Windowing: True, OTP EN: False, AWB_GAIN_EN: False */
{0x5003, 0x08}, /*ISP Control: Buffer Enable: True, Bin Auto Mode: True*/

//Pad Control
{0x3000, 0x00}, /* Disable Parallel Pads */
{0x3001, 0x00}, /* Disable Parallel Pads */
{0x3002, 0x00}, /* PAD Enables... all disabled */
{0x3016, 0x08}, /* MIPI Pad Enable: LPH: 0x00, MIPI_PAD_ENABLE: True, Bypass HS Enable Latch: False,
ICTL: Bias Current Adj: 0x00 */
//{0x3017, 0xE0}, /* MIPI Common PHY: VCM: 0x03, LPTX (strength): 0x02, IHALF: False, VICD: CD Output
Low 0 */
//{0x3017, 0x90}, /* MIPI Common PHY: VCM: 0x03, LPTX (strength): 0x02, IHALF: False, VICD: CD Output
Low 0 */
{0x3017, 0x10}, /* MIPI Common PHY: VCM: 0x03, LPTX (strength): 0x02, IHALF: False, VICD: CD Output Low
0 */
{0x3018, 0x44}, /* MIPI Lane Mode: 2 Lanes, HS Power Down: False, LS RX Power Down: True, MIPI Enable:
True, MIPI Suspend: False */

//Probably from older driver
//{0x301C, 0xF8}, /* ??? */
//{0x301D, 0xF0}, /* ??? */

{0x3A18, 0x00}, {0x3A19, 0xF8},/* Auto Exposure Ceiling 0x00F8 */

{0x3C01, 0x80}, /* 50/60 Hertz Detection: Manual Mode */

//External Srobe
{0x3B07, 0x0C}, /* FREX MODE SEL: FX1_FM_EN: True, FREX_INV: False, FREX_STROBE_MODE: Rolling Strobe */

// //Timing
// {0x380C, 0x09}, /* Horizontal Size: 2322 */
// {0x380D, 0x70},
// //{0x380C, 0x0A}, /* Horizontal Size: 2416 */
// //{0x380D, 0x8C},
//
// {0x380E, 0x04}, /* Vertical Size: 1104 */
// {0x380F, 0x50},
//
// //{0x380E, 0x04}, /* Vertical Size: 1080 */
// //{0x380F, 0x38},
//
// //Windowing the Sensor
// {0x3800, 0x01}, {0x3801, 0x5C}, /* X Start Address: 348 */
// {0x3802, 0x01}, {0x3803, 0xB2}, /* Y Start Address: 434 */
// {0x3804, 0x08}, {0x3805, 0x09}, /* X End Address: 2275 */
// {0x3806, 0x05}, {0x3807, 0xF1}, /* Y End Address: 1521 */
// /* X Width: 2275 - 348: 1927 */
// /* Y Width: 1521 - 434: 1087 */
//
// /* DVP: Digital Video Port (Parallel Output), is this even used??? */
// {0x3808, 0x07}, {0x3809, 0x80},/* DVP Output Horizontal Width: 1920 */
```

```
// {0x380A, 0x04}, {0x380B, 0x38},/* DVP Output Vertical Height: 1080 */
//
//
// //ISP Offsets
// {0x3811, 0x04}, /* ISP Horizontal Offset: 4 */
// {0x3813, 0x02}, /* ISP Vertical Offset: 2 */
//
// /* Calculated Size:
//      Width: (1927 + 1) - (4 * 2)      = 1920
//      Height: (1087 + 1) - (2 * 2 * 2) = 1080
// */
//
// {0x3814, 0x11}, /* X Increment: H_ODD_INC: 1, H_EVEN_INC: 1 */
// {0x3815, 0x11}, /* X Increment: H_ODD_INC: 1, H_EVEN_INC: 1 */

{0x3800, 0x00}, {0x3801, 0x00}, //H Start:          0x0000: 0
{0x3802, 0x00}, {0x3803, 0xf8}, //V Start:          0x00F8: 248
{0x3804, 0x0a}, {0x3805, 0x3f}, //H End:           0x0A3F: 2623
{0x3806, 0x06}, {0x3807, 0xab}, //V End:           0x06AB: 1707
{0x3808, 0x07}, {0x3809, 0x80}, //Width:           0x0780: 1920
{0x380a, 0x04}, {0x380b, 0x38}, //Height:          0x0438: 1080
{0x380c, 0x0a}, {0x380d, 0x80}, //Horizontal Timing Size: 0x0A80: 2688
{0x380e, 0x07}, {0x380f, 0xc0}, //Vertical Timing Size:  0x07C0: 1984

{0x3810, 0x00}, {0x3811, 0x02}, //ISP X Window Offset:  0x0002: 2
{0x3812, 0x00}, {0x3813, 0x02}, //ISP Y Window Offset:  0x0002: 2

{0x3814, 0x11},          //X_ODD_INC, X_EVEN_INC:  1,1
{0x3815, 0x11},          //Y_ODD_INC, Y_EVEN_INC:  1,1

//The Following is undocumented in the datasheet
/*
{0x3630, 0x2E},
{0x3632, 0xE2},
{0x3633, 0x23},
{0x3634, 0x44},
{0x3636, 0x06},
{0x3620, 0x64},
{0x3621, 0xE0},
{0x3600, 0x37},
{0x3704, 0xA0},
{0x3703, 0x5A},
{0x3715, 0x78},
{0x3717, 0x01},
{0x3731, 0x02},
{0x370B, 0x60},
{0x3705, 0x1A},
{0x3F05, 0x02},
{0x3F06, 0x10},
{0x3F01, 0x0A},
*/

//50/60Hz Filter
{0x3A08, 0x01}, {0x3A09, 0x4B},// B50 Step
{0x3A0A, 0x01}, {0x3A0B, 0x13},/* B60 Step */
{0x3A0D, 0x04}, /* B60 Max: 4 */
{0x3A0E, 0x03}, /* B50 Max: 3 */

{0x3A0F, 0x58}, /* WPT: Stable Range High Limit (Enter) */
{0x3A10, 0x50}, /* BPT: Stable Range Low Limit (Enter) */
{0x3A1B, 0x58}, /* WPT: Stable Range High Limit (Go Out) */
{0x3A1E, 0x50}, /* BPT: Stable Range Low Limit (Go Out) */
{0x3A11, 0x60}, /* High VPT: 96 */
{0x3A1F, 0x28}, /* Low VPT: 40 */

//Black Level Control
{0x4001, 0x02}, /* BLC_CTRL01: Start Line: 2*/
```

```

{0x4004, 0x04}, /* BLC Line Number*/
{0x4000, 0x09}, /* BLC Control:*/
/* BIT[7]: BLC_MEDIAN_FILTER_ENABLE: False*/
/* BIT[3]: ADC_11BIT_MODE: False*/
/* BIT[2]: APPLY2BLACKLINE: False: False*/
/* BIT[1]: Black Line Average Frame: True*/
/* BIT[0]: BLC Enable: False*/

//{0x4837, 0x19}, /* Period of PCLK2X, PCLK_DIV = 1 and 1-bit Decimal: 25 */ //XXX: DIDN'T MAKE MUCH
// DIFFERENCE
{0x4837, 0x0A}, /* Period of PCLK2X, PCLK_DIV = 1 and 1-bit Decimal: 10 */
//{0x4837, 0x15}, /* Period of PCLK2X, PCLK_DIV = 1 and 1-bit Decimal: 25 */

//{0x4800, 0x34}, /* MIPI Control: */ //??? //NORMAL
{0x4800, 0x20}, /* MIPI Control: */ //??? //NORMAL
/* BIT[7]: MIPI HS Only: False */
/* MIPI Supports CD and Escape Mode */
/* BIT[6]: CK_MARK1_EN: False */
/* ENABLE Clock Lane Mark1 When Resume: */
/* BIT[5]: Clock Lane Gate Enable: Clock Gating */
/*0: Clock is free Running */
/*1: Clock is not free running (only active when sending) */
/* BIT[4]: Linx Sync Enable: Send Line Short Packet for Each Line: True */
/*0: Do not send line short packet for each line */
/*1: Send line short packet for each line */
/* BIT[3]: Lane Select: Lane 0 */
/*0: Use 0 as default lane select */
/*1: Use 1 as default lane select */
/* BIT[2]: Idle Status: LP11 */
/*0: MIPI Bus will be LP00 when no packet to transmit */
/*1: MIPI Bus will be LP11 when no packet to transmit */
/* BIT[1]: Clock Lane First bits: 0x55 */
/*0: First bits: 0x55 (Clock is low first) */
/*1: First bits: 0xAA (Clock is high first) */
/* BIT[0]: Manually set clock lanes to low power mode */

{0x4801, 0x0C}, //LSB First
//{0x4801, 0x04},
//{0x4801, 0x13},
//{0x4801, 0x34}, /* MIPI Control: */ //??? //NORMAL
//{0x4801, 0x04}, /* MIPI Control: */ //??? //NORMAL
//{0x4801, 0x0B}, /* MIPI Control 1:*/

{0x3503, 0x03}, /* Manual Control: Gain Latch Timing Delay: 0, VTS: Auto, AGC: Manual, AEC: Manual */
{0x3022, 0x00},

// Binning
{0x3820, 0x00}, /* Vertical Binning: False */
//{0x3821, 0x02}, /* Horizontal Binning: False, Mirror SNR */
{0x3821, 0x00}, /* Horizontal Binning: False, Don't Mirror SNR */
{OV5647_TABLE_END, 0x0000}
};

static const ov5647_reg mode_1280x720[] = {
{0x0103, 0x01}, //Software Reset
{OV5647_TABLE_WAIT_MS, 10 }, //wait 10ms
{0x0100, 0x00}, /* 0xIncluding, 0xsw, 0xreset, 0x*/

/* Clock Configuration */
{0x3034, 0x1A}, /* PLL Control 0: PLL Charge Pump, MIPI Bit Mode: 10 */
{0x3035, 0x21}, /* PLL Control 1: System Clock Divider 2, Scale Divider (1)*/
{0x3036, 0x69}, /* PLL Multiplier: 105 */
//{0x3036, 0x34}, /* PLL Multiplier: 52 */
{0x303C, 0x11}, /* PLL Control 2 PLL CP, PLL SYS DIV = 1*/
/* Clock = 24000 / 2 /1 * 105 = 1.26 GHz */

//Is this for I2C Clock??
{0x3106, 0xF5}, /* SRB Control: PLL Clock Divider / 4, Reset Arbiter: 0, SCLK Arb: 0 */
/* PCLK = SCLK / 4 */

```

```
//Undocumented
/*
{0x3827, 0xEC}, // ???
{0x370C, 0x03}, // ???
{0x3612, 0x5B}, // ???
{0x3618, 0x04}, // ???
*/

//ISP Control
{0x5000, 0x06}, /*ISP Control: Lense Correction: Disable, Defective Pixel Cancel BC_EN, WC_EN: True,
True */
{0x5002, 0x40}, /*ISP Control: WIN_EN: True, OTP EN Fale, AWB GAIN EN: False */
{0x5003, 0x08}, /*ISP Control: Digital Compensation Bypass: True, MAN_OPT: False, MAN_EN: False */

//Pad Control
{0x3000, 0x00}, /*Pad Output Enable */
{0x3001, 0x00},
{0x3002, 0x00}, /* PAD Enables... all disabled */
{0x3016, 0x08}, /* MIPI Pad Enable: LPH: 0x00, PIP_PAD_ENABLE: True, PGM_BP_HS_EN_LAT: False, ICTL: Bias
Current Adj: 0x00 */
//{0x3017, 0xE0}, /* MIPI Common PHY: VCM: 0x03, LPTX (strength): 0x02, IHALF: False, VICD: CD Output
Low 0 */
{0x3017, 0x10}, /* MIPI Common PHY: VCM: 0x03, LPTX (strength): 0x02, IHALF: False, VICD: CD Output Low
0 */
{0x3018, 0x44}, /* MIPI Lane Mode: 2 Lanes, HS Power Down: False, LS RX Power Down: True, MIPI Enable:
True, MIPI Suspend: False */

//Undocumented, probaby from older driver
//{0x301C, 0xF8}, /* ??? */
//{0x301D, 0xF0}, /* ??? */

{0x3A18, 0x00}, /* Auto Gain Ceiling 0x00F8 */
{0x3A19, 0xF8},

{0x3C01, 0x80}, /* 50/60 Hertz Detection: Manual Mode */

//External Srobe
{0x3B07, 0x0C}, /* FREX MODE SEL: FX1_FM_EN: True, FREX_INV: False, FREX_STROBE_MODE: Rolling Strobe */

//Timing
{0x380C, 0x07}, /* Horizontal Size: 1896 */
{0x380D, 0x68},

{0x380E, 0x04}, /* Vertical Size: 1104 */
{0x380F, 0x50},

{0x3814, 0x11}, /* X Increment: H_ODD_INC: 1, H_EVEN_INC: 1 */
{0x3815, 0x11}, /* X Increment: H_ODD_INC: 1, H_EVEN_INC: 1 */
//{0x3708, 0x64}, /* ??? */
//{0x3709, 0x12}, /* ??? */

{0x3808, 0x05}, /* DVP Output Horizintal Width: 1280 */
{0x3809, 0x00},

{0x380A, 0x02}, /* DVP Output Vertical Height: 720 */
{0x380B, 0xD0},

//Windowing the Sensor
{0x3800, 0x00}, /* X Start Address: 12 */
{0x3801, 0x0C},
{0x3802, 0x00}, /* Y Start Address: 4 */
{0x3803, 0x04},
{0x3804, 0x05}, /* X End Address: 1299 */
{0x3805, 0x13},
{0x3806, 0x02}, /* Y End Address: 731 */
{0x3807, 0xBD},

/* X Width: (1299 - 12) + 1: 1288 */
/* Y Width: (731 - 4) + 1: 728 */
```

```
//ISP Offsets
{0x3811, 0x04}, /* ISP Horizontal Offset: 16 */
{0x3813, 0x02}, /* ISP Vertical Offset: 6 */

/* Image Width: 1288 - (4 * 2) = 1280 */
/* Image Height: 728 - (2 * 2 * 2) = 720 */

//The Following is undocumented in the datasheet
/*
{0x3630, 0x2E},
{0x3632, 0xE2},
{0x3633, 0x23},
{0x3634, 0x44},
{0x3636, 0x06},
{0x3620, 0x64},
{0x3621, 0xE0},
{0x3600, 0x37},
{0x3704, 0xA0},
{0x3703, 0x5A},
{0x3715, 0x78},
{0x3717, 0x01},
{0x3731, 0x02},
{0x370B, 0x60},
{0x3705, 0x1A},
{0x3F05, 0x02},
{0x3F06, 0x10},
{0x3F01, 0x0A},
*/

//50/60Hz Filter
{0x3A08, 0x01}, /* B50 Step */
{0x3A09, 0x28},
{0x3A0A, 0x00}, /* B60 Step */
{0x3A0B, 0xF6},
{0x3A0D, 0x08}, /* B60 Max: 4 */
{0x3A0E, 0x06}, /* B50 Max: 3 */

{0x3A0F, 0x58}, /* WPT: Stable Range High Limit (Enter) */
{0x3A10, 0x50}, /* BPT: Stable Range Low Limit (Enter) */
{0x3A1B, 0x58}, /* WPT: Stable Range High Limit (Go Out) */
{0x3A1E, 0x50}, /* BPT: Stable Range Low Limit (Go Out) */
{0x3A11, 0x60}, /* High VPT: 96 */
{0x3A1F, 0x28}, /* Low VPT: 40 */

//Black Level Control
{0x4001, 0x02}, /* BLC_CTRL01: Start Line: 2*/
{0x4004, 0x04}, /* BLC_Line Number*/
{0x4000, 0x09}, /* BLC Control:*/
/* BIT[7]: BLC_MEDIAN_FILTER_ENABLE: False*/
/* BIT[3]: ADC_11BIT_MODE: False*/
/* BIT[2]: APPLY2BLACKLINE: False: False*/
/* BIT[1]: Black Line Average Frame: True*/
/* BIT[0]: BLC Enable: False*/

//{0x4837, 0x10}, /* Period of PCLK2X, PCLK_DIV = 1 and 1-bit Decimal: 22 */
{0x4837, 0x19}, /* Period of PCLK2X, PCLK_DIV = 1 and 1-bit Decimal: 22 */

//{0x4800, 0x24}, /* MIPI Control: */
{0x4800, 0x34}, /* MIPI Control: */
/* BIT[7]: MIPI HS Only: False */
/* MIPI Supports CD and Escape Mode */
/* BIT[6]: CK_MARK1_EN: False */
/* ENABLE_Clock Lane Mark1 When Resume: */
/* BIT[5]: Clock Lane Gate Enable: Clock Gating */
/*0: Clock is free Running */
/*1: Clock is not free running (only active when sending) */
/* BIT[4]: Linx Sync Enable: Send Line Short Packet for Each Line: True */
/*0: Do not send line short packet for each line */
```

```
        /*1: Send line short packet for each line */
/* BIT[3]: Lane Select: Lane 0 */
        /*0: Use 0 as default lane select */
        /*1: Use 1 as default lane select */
/* BIT[2]: Idle Status: LP11 */
        /*0: MIPI Bus will be LP00 when no packet to transmit */
        /*1: MIPI Bus will be LP11 when no packet to transmit */
/* BIT[1]: Clock Lane First bits: 0x55 */
        /*0: First bits: 0x55 (Clock is low first) */
        /*1: First bits: 0xAA (Clock is high first) */
/* BIT[0]: Manually set clock lanes to low power mode */

//TEST:
//{0x4801, 0x08}, /* MIPI Control 1:*/

//{0x3503, 0x03}, /* Manual Control: Gain Latch Timing Delay: 0, VTS: Auto, AGC: Manual, AEC: Manual */
{0x3503, 0x07}, /* Manual Control: Gain Latch Timing Delay: 0, VTS: Auto, AGC: Manual, AEC: Manual */
{0x350C, 0x00}, /* Manual Control: Gain Latch Timing Delay: Datasheet says "set to 0" */
{0x350D, 0x00}, /* Manual Control: Gain Latch Timing Delay: Datasheet says "set to 0" */

// Binning
{0x3820, 0x00}, /* Vertical Binning: False */
{0x3821, 0x02}, /* Horizontal Binning: False, Mirror SNR */

{OV5647_TABLE_END, 0x0000}
};

enum {
    OV5647_MODE_1920X1080,
    OV5647_MODE_1280X720,

    OV5647_MODE_START_STREAM,
    OV5647_MODE_STOP_STREAM,
    OV5647_MODE_TEST_PATTERN
};

static const ov5647_reg *mode_table[] = {
    [OV5647_MODE_1920X1080] = mode_1920x1080,
    [OV5647_MODE_1280X720] = mode_1280x720,

    [OV5647_MODE_START_STREAM] = ov5647_start,
    [OV5647_MODE_STOP_STREAM] = ov5647_stop,
    [OV5647_MODE_TEST_PATTERN] = tp_colorbars,
};

static const int ov5647_30fps[] = {
    30,
};

static const int ov5647_60fps[] = {
    60,
};

static const struct camera_common_frmfmt ov5647_frmfmt[] = {
    {{1920, 1080}, ov5647_30fps, 1, 0, OV5647_MODE_1920X1080},
    {{1280, 720}, ov5647_60fps, 1, 0, OV5647_MODE_1280X720},
};
#endif /* __OV5647_TABLES__ */
```