#ifndef \_TSL2591\_H\_

#define \_TSL2591\_H\_

#define TSL2591\_VISIBLE (2) // channel 0 - channel 1

#define TSL2591\_INFRARED (1) // channel 1

#define TSL2591\_FULLSPECTRUM (0) // channel 0

#define TSL2591\_ADDR (0x29)

#define TSL2591\_ID (0x50)

#define TSL2591\_READBIT (0x01)

#define TSL2591\_COMMAND\_BIT \

(0xA0) // 1010 0000: bits 7 and 5 for 'command normal'

#define TSL2591\_CLEAR\_INT (0xE7)

#define TSL2591\_TEST\_INT (0xE4)

#define TSL2591\_WORD\_BIT (0x20) // 1 = read/write word (rather than byte)

#define TSL2591\_BLOCK\_BIT (0x10) // 1 = using block read/write

#define TSL2591\_ENABLE\_POWEROFF (0x00)

#define TSL2591\_ENABLE\_POWERON (0x01)

#define TSL2591\_ENABLE\_AEN \

(0x02) // ALS Enable. This field activates ALS function.Writing a one

// activates the ALS.Writing a zero disables the ALS.

#define TSL2591\_ENABLE\_AIEN \

(0x10) // ALS Interrupt Enable. When asserted permits ALS interrupts to be

// generated, subject to the persist filter.

#define TSL2591\_ENABLE\_NPIEN \

(0x80) // No Persist Interrupt Enable. When asserted NP Threshold conditions

// will generate an interrupt, bypassing the persist filter

#define TSL2591\_LUX\_DF (408.0F)

#define TSL2591\_LUX\_COEFB (1.64F) // CH0 coefficient

#define TSL2591\_LUX\_COEFC (0.59F) // CH1 coefficient A

#define TSL2591\_LUX\_COEFD (0.86F) // CH2 coefficient B

enum {

TSL2591\_REGISTER\_ENABLE = 0x00,

TSL2591\_REGISTER\_CONTROL = 0x01,

TSL2591\_REGISTER\_THRESHOLD\_AILTL = 0x04, // ALS low threshold lower byte

TSL2591\_REGISTER\_THRESHOLD\_AILTH = 0x05, // ALS low threshold upper byte

TSL2591\_REGISTER\_THRESHOLD\_AIHTL = 0x06, // ALS high threshold lower byte

TSL2591\_REGISTER\_THRESHOLD\_AIHTH = 0x07, // ALS high threshold upper byte

TSL2591\_REGISTER\_THRESHOLD\_NPAILTL =

0x08, // No Persist ALS low threshold lower byte

// TSL2591\_REGISTER\_THRESHOLD\_NPAILTH = 0x09, // etc

TSL2591\_REGISTER\_THRESHOLD\_NPAIHTL = 0x0A,

TSL2591\_REGISTER\_THRESHOLD\_NPAIHTH = 0x0B,

TSL2591\_REGISTER\_PERSIST\_FILTER = 0x0C,

TSL2591\_REGISTER\_PACKAGE\_PID = 0x11,

TSL2591\_REGISTER\_DEVICE\_ID = 0x12,

TSL2591\_REGISTER\_DEVICE\_STATUS = 0x13,

TSL2591\_REGISTER\_CHAN0\_LOW = 0x14,

TSL2591\_REGISTER\_CHAN0\_HIGH = 0x15,

TSL2591\_REGISTER\_CHAN1\_LOW = 0x16,

TSL2591\_REGISTER\_CHAN1\_HIGH = 0x17

};

typedef enum {

TSL2591\_INTEGRATIONTIME\_100MS = 0x00,

TSL2591\_INTEGRATIONTIME\_200MS = 0x01,

TSL2591\_INTEGRATIONTIME\_300MS = 0x02,

TSL2591\_INTEGRATIONTIME\_400MS = 0x03,

TSL2591\_INTEGRATIONTIME\_500MS = 0x04,

TSL2591\_INTEGRATIONTIME\_600MS = 0x05,

} tsl2591IntegrationTime\_t;

typedef enum {

// bit 7:4: 0

TSL2591\_PERSIST\_EVERY = 0x00, // Every ALS cycle generates an interrupt

TSL2591\_PERSIST\_ANY = 0x01, // Any value outside of threshold range

TSL2591\_PERSIST\_2 = 0x02, // 2 consecutive values out of range

TSL2591\_PERSIST\_3 = 0x03, // 3 consecutive values out of range

TSL2591\_PERSIST\_5 = 0x04, // 5 consecutive values out of range

TSL2591\_PERSIST\_10 = 0x05, // 10 consecutive values out of range

TSL2591\_PERSIST\_15 = 0x06, // 15 consecutive values out of range

TSL2591\_PERSIST\_20 = 0x07, // 20 consecutive values out of range

TSL2591\_PERSIST\_25 = 0x08, // 25 consecutive values out of range

TSL2591\_PERSIST\_30 = 0x09, // 30 consecutive values out of range

TSL2591\_PERSIST\_35 = 0x0A, // 35 consecutive values out of range

TSL2591\_PERSIST\_40 = 0x0B, // 40 consecutive values out of range

TSL2591\_PERSIST\_45 = 0x0C, // 45 consecutive values out of range

TSL2591\_PERSIST\_50 = 0x0D, // 50 consecutive values out of range

TSL2591\_PERSIST\_55 = 0x0E, // 55 consecutive values out of range

TSL2591\_PERSIST\_60 = 0x0F, // 60 consecutive values out of range

} tsl2591Persist\_t;

typedef enum {

TSL2591\_GAIN\_LOW = 0x00, // low gain (1x)

TSL2591\_GAIN\_MED = 0x10, // medium gain (25x)

TSL2591\_GAIN\_HIGH = 0x20, // medium gain (428x)

TSL2591\_GAIN\_MAX = 0x30, // max gain (9876x)

} tsl2591Gain\_t;

#endif