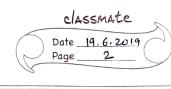


	12C SELF TEST
	10 read external BH1750 ambient light sensor
	© self 12C master — 12c slave
	12c master gpio18 gpio19
	12c slave gpio25 gpio26
	Cexternal pull-up resistors not required for SDA/SCL
	as driver will enable internal pull-up resistors
3	The second secon
	make menuconfig / idf.py menuconfig
	12c master-> pins, port, frequency
	12c slave > pins, port, address
	BH1750 sensor -> slave address (0x23 / 0x5c)
	operation mode Cone time L-resolution
	(4 lux, time = 16 ms)
	A seem of the seems of the seem
	make -jh flash monitor
	Surprise Contract Con
	The same of the sa
	tinclude (stdion)
世	include "esp-log.h"
#	include "driver/i2c.h"
**	include "sakconfig.h"
٠.	tatic const char +TAG = "izc-example";
*	de
(define DATA-LENGTH 512
	define RW_TEST_LENGTH 128
	Holing DELAY_TIME_BETWEEN_ITEMS_MS 1000



define 12c_SLAVE_SCL_10 CONFIGE_12C_SLAVE_SCL # define 12c_SLAVE_SDA_10 CONFIG_12c_SLAVE_SDA # define 12C_SLAVE_NUM 12C_NUMBER (CONFIG_ 12C-SLAVE PORT-NUM) # define 12C-SLAVE_TX-BUF-LEN (2* BATA-LENGTH) # define 12C-SLAVE_RX_BUF_LEN C2* DATA_LENGITH) # define 12C-MASTER-SCL-10 CONFIG-12C-MASTER-SCL # define 12C_MASTER_SDA_10 CONFIG_12C_MASTER_SDA # define 12C_MASTER_NUM 12C_NUMBER (CONFIG_12C_MASTER_PORT_NUM) #define 12C-MASTER_FREQ_HZ CONFIG-12C-MASTER_FREQUENCY #define DC_MASTER-TX-BUF_DISABLE O #define 12C_MASTER_RX_BUF_DISABLE D #deline BHITSD_SENSOR_ADDR CONFIG_BHITSD_ADDR # define BHI750-CMD_START CONFIG-BHI750-OPMODE #define ESP_SLAVE_ADDR CONFIG: 12C_SLAVE_ADDRESS # define WRITE-BIT 12C-MASTER-WRITE #define READ-BIT 12C_MASTER-READ #define ACK-CHECK-EN OXI # define ACK-CHECK-DLS OND # define ACK-VAL OXO #define NACK_VAL DXI Semaphore Handle - + print-mox = NULL;

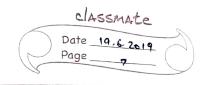
Static esperrt ize-master read-slave (ize-port-t izanum, vintet *datand, sizet size) & if (size == 0) return ESP_OK; ize-cmd-handle-t cmd = ize cmd link-create(). izc master start (cmd); izc_master_write_byte Comd, (ESP_SLAVE_ADDR <<1) | READ BIT, ACK_CHECK_EN); if (size >1) & man man man, 20 10 izc master read (cmd, data-vd, size-1, ACK-VAL ize master read byte (cmd, data rd + size -1, NACK-VAL). ise_master_stop(cmd); esp_err_t ret = i2c_master_begin (ise num, emd, 1000/portTICK RATE MS); ise- cmd-link delete (cmd); static esperret ize-master write-slave (ize port ize num, wints + *data wr, size + size } ize_cmd_handle_t cmd = ize_cmd_link_create(); izc_master_start (cmd). isc-master-write-byte (cmd, (ESP SLAVE ADPR (1) 1 WRITE BIT, ACK CHECK EN) ize-master conite (cmd, data-cor, size ACK_CHECK_EN) izc-master_ stop (cmd), esperant ret = izc master and tagin (ic num,

emd, 1000 / portTICK_RATE_MS)

izc_cmd_link_delete (cmd). return ret; static esperret izc-master senson test (izc-port t izc num vints-t *data h vints (*data 1) izc_cmd_handle_t cmd = izc_cmd_link_create(). izc-master start (and); 12 c. master write byte (cmd, (BH1750 SENSOR ADDR K) I WRITE BIT, ACK CHECK EN). ize-master corite byte (cmd, BHITCO CMD START, ACK CHECK FX) izc-master stop (cmd), int ret = ize - master and begin (ize num, and, 1000/portTICK) if (ret 1 = FSP OK) return ret, VTask Delay (20 / portTICK RATE MS). cmd = izc_ mound cmd_link_create(). izc-master-start (cma); isc moster write byte (cmd, (BHITTO SENSOR ADDRKL) IREAD BIT, ACK_CHECK_EN); ize_master_read_tyte (cmd, data-h, ACK-VAL) ize master read tyte (cond, data-1, NACK VAL isc-master Stop Cand). ret = izc_master cmd_begin (izc:num, and, 1000/portTick_ ize_cmd_link_delete (cmd); return ret; 2

Static espernt izc_master_init() & int i2c_master_port = 12c_MASTER_NUM; ize-config-t conf; Conf. mode = 12C-MODE-MASTER; conf. sdq-10-num = 12C-MASTER-SDA-10; conf. sda-pullup_en = GPID_PULLUP_ENABLE; Conf. scl-10-num = 12C_MASTER - SCL-101 confiscl-pullupen = GPIO-PULLUP- ENABLE; conf. master. clk-speed = 12C-MASTER-FREQ-HZ; 12c-param-config (isc master port, & conf); return izc-driver-install Cizc-master-port Conf. mode, IZC_MASTER-RX_BUF_DISABLE 12C MASTER TX-BUF DISABLE, 0) static esperrt izc_slave init() & int izc_slave-port = 12C_SLAVE_NUM; ize-config-t conf-slave; conf clave solatio- num = 12c SLAVE SDA 10; conf_slave .da_pullup_en = GP10_PULLUP_ENABLE; conf_slove. Scl_co_num = 12c_SLAVE_SCL_10; conf_slave scl-pollup en = GPIO- PULLUP_ENABLE; conf. slave mode = 12C_MODE_SLAVE, conf_slave slave, addr_16 bit_en = 0. Conf_Slave, slave slave - addn = ESP_SLAVE - ADDR. (2c - peram-config (icc slave port, &conf slave). return ize-driver_install (ize-slave port, conf-slave mode 12C-SLAVE_RX_BUF_LEN, 12C SLAVE_TX_BUF_LEN, 0);

Static void deep by (vinter + buy, int len) & for (i=0; i < len; i++) & print ("Y.02x", buf [i]); if ((i+1) x 16 == 0) print(("\n"), print ("\n"). Static void isc-test task (void *arg) & int 1=0; int reti vint32-t task_idx = (vint32-t)ang. vint 8 + * data = (vints +) malloc (DATA_LENGTH); wints + data wor = (wints + *) malloc (DATA-LENGITH); vinte + * dota rd = (vinte + +) malloc (DATA LENGTH). wints-t sensor data-h, sensor data-l; int cnt = 0; while (1) & ESP LOGI (TAG, "TACK [Y.d] test ent: xd", task-idx, ent++), ret = 12c-master sensor test (12c-MASTER-NUM, & sensor data - h, & sensor data - l). x Semaphore Take (print-mux, portMAX_DELAY). if (ret == ESP_ERR_TIMEOUT) & ESP_LOGE (TAG, "12C timeout"); 3 else if (ret == ESPOK) & print (" ****** ** ** * * * * *); print ("TASK [x.d] MASTER READ SENSOR (BHIZSO) IN", task idx).



print ("data-h: x.o.ze \n", sensor data-h); print ("data-1: youx in", sensor-data-1); print ("sensor val: x. ozf [LUX] \n") (sensor data-h KKB | sensor data-1) /12 ESP-LOGW (TAG, "Y.S: No ack, senson not connected ... skip ... , esp err to name (ret)). x Semaphore Give (grint morex). VTack Delay (CDELAY_TIME_BETWEEN_ITEMS_MS * (+ask-idx+1))/portTICK_RATE_MC). for (i=0; i'L DATA_LENGITH: itt) & data [i] = i, x Semaphore Take (print mux, portMAX DELAY). size + d size = 12c - slave write - buffer C12c - SLAVE NUM, data, RW-TEST-LENGITH, 1000 / partTICK RATE MS) if (d-size == 0) & ESP_LOGIO (TAG, "ize slove to buffer full" ret = 12C Black - read_slave (12C_MASTER_NUM, data_rd } else & RUD_ TEST_LENGTH) ret = 12c - master read slave (12c MASTER NOM, data rd ર if (ret = = ESP. FRR-TIMEOUT) & ESP_LOGE (TAG, "12c timeout");

& else if Kret == ECP_OK) &

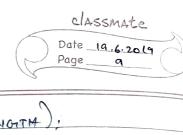
prints (" * * * * * * * * * * * * * * * * *):

print ("TASK EXIL MASTER READ FROM SLAVE IN" + LOSK.

print (" * * * * * * * * * * * * * "); printf ("==== TASK [Y.d] Slave buffer data ==== \n', rask_idx) disp-buf (data, d-size); Prints ("==== TASK[Yd] master read ==== \n", task_idx); diep-buf (data rd, disize). 3 else & ESP-LOGUE TAG, "TASK[r.d] Y.S: master read slave error, IO not connected ... \n", task_idx, esp_en_to_name(rot)). x Semaphore Give (print_mux); POTHTICK- RATE-MS); VTaskDelay((DELAY TIME-BETWEEN_ITEMS_MS * (task idx+ 1)) for (i=0; i < DATA_LENGTH; i++) & data_wr [i] = i+10; X. Semaphore Take (print mux, portMAX_DELAY);

RW-TEST-LENGITH);

ret = i2c-master write_slave (12c_MASTER-NUM, data-wr), if (ret == ESP-OK) S Size = izc-slave read buffer (12c-SLAVE NUM, data RW. TEST_LENGTH, 1000/portTICK RATE MS). if (ret == ESP ERR TIMEOUT) & ESP_LOGE (TAG, "12c timeout"). I else of (ret = = ESPOK) f print ("+ *** ** * * * * * * * * * * * *); print ("TASKEY. d] MASTER WRITE TO SLAVE ("), task idx), print (11 --- TASK[7,d] Moster write [7,d] bytes --- In" task-idx (dx)



disp-buf Conta-wr RW-TEST-LENGITH

print [" --- TASK [r.d] slave read: Drd] bytes --- In" task-idr, size);

disp by (data, size); Felse &

ESP LOGO ("TAG, "TACK [xd] YS: master write to slave error

10 not connected .. in " task-idx, esp err to name Cret)

* Semaphone Give (print mux);

V. Semaphone Delete (print mox);

print_mux = x Semaphore (reate Motor ().

ESP_ERROR_CHECK (isc_shoe_ini+()).

ESP_ERROR_CHECK (izc_master_ini+ c)).

*Tack Create Ciza test tack, "iza test tack o", 1024 x2

* Task Create (ize test task, "exc test task 1", 1024 x2, (void*)

VTaskDelete (NULL);

void app main () &

VTask Delay (CBELAY TIME BETWEEN TEMS - MS & (task idn+)

LO, MULL);