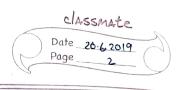
switch (event > event_ia) &

- 1	MQTTTCP
	# include (statio.h)
	# include (string.h)
	# include <stdint-h></stdint-h>
	#include <stddef h<="" th=""></stddef>
	Hindude "esparifich"
	# include "esp system."
	# include "nvs-blash.h"
	#include "esp-event-h"
	# include "topip adapter-h"
	# include "protocol_examples_common.h"
	# include "freertos/FreeRTOS.h"
	# include "freertos/task.n"
_	# include "freertos/semphr.h"
	#include "preentos/queve.n"
_	# include "larip (sockets.h"
	#include "lwip/dns.h"
_	# include "lwip/netalb.h"
	#include "esp log.h"
	# include "matt-client.h"
	Carried San
	the state of the s
_	Static const char *TACI = "MOTT_EXAMPLE";
_	
_	Static espernt matt event-handler ch Cosp matt event handle t event
_	esp matt client handle t client = event -> client;
	int msq id;



event > msg_id);

COUSE MOTT EVENT CONNECTED: ESP_LOGI CTAG, "MOTT_EVENT_CONNECTED"). msg id = esp - matt client publish (client, "Appie/gosi", "data 3", 0, 1, 0); ESP_LOGIL (TAG, "sent publish successful, magid="d", mag-id). mesensg_id = esp_matt_client_subscribe Eclient, "/topic/qoso", o). ESPLOGAL (TAG, "sent subscribe successful, mg-id= y.d", msg id); megid = esp_mg++_ client_subscribe (client, "/topic/gosi"). ESP_LOGI (TAG, "cent subscribe successful, meg id = v.d", meg-id); meg_id = esp_matt_client_unsubscribe (client, "topic/gasi"). ESP_LOGAL CTACA, "sent unsubscribe successful, msg id= xd" msg id); case MOTT EVENT DISCONNECTED! ESP. LOGI (TAG, "MOTT EVENT - DISCONNECTED"). Case MOTT EVENT SUBSCRIBED! event > msg_id). ESP LOCAL (TAG, "DRAMOTT-EVENT-SUBSCRIBED, Mig-i'd="xd" my i'd= esp matt dient publish (client, "Hopic/goso", "data",000) ESPLOGI (TAG, "sent publish successful, magid= xd, magid); break;

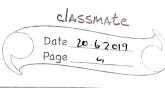
Case MOTT_EVENT-PUBLISHEDI

E SP. LOGIL (TAG. "MOTT-EVENT-UNSUBSCRIBED,

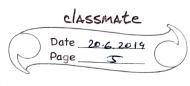
COSE MOTT-EVENT-UNSUBSCRIBED!

ESPLOGI (TAG, "MOTT EVENT PUBLISHED, msg. id=v.d", events msg.a)

	case MOTT-EVENT-DATA:
	ESP_LOGI (TAGI, "MOTT-EVENT_DATA"),
	print ("TOPIC = x. * clin" event -> topic len, event -> topic).
	printf ("DATA = x. *ser \n" event > data len, event > data).
	break;
	Case MOTT_EVENT-ERROR!
	ESP_LOCAL ("TACT, "MOTT_EVENT_ERROR").
	break;
	del
-	default:
	ESP_LOGICTAG, "other event i'd: Y.d", event -> event_id).
	break,
	· }
	return ESP-OK;
	3
-	
	static void matt event handler (void *handler args,
	esp event base t base, into 2 + event id, void * event data) &
	ESPLOGD (TAG) " event dispatched from event loop base = Xs
	event id = y, d, base, event_id).
	matt_event_handler_cb (event_data);
	3



	Date 20.62019 Page 4
BUT WHY:	ns x s
	Static void matt_app_start(void) &
	esp matt-client-config t matt-cfg = &
4	. Uni = CONFIG BROKER - URL
	3.
	# ONFIG BROKER URL FROM STDIN
	Chan line [128];
	of (stromp (matt-cfg. uri, "FROM_STDIN") == 0) &
	int count = 0;
	print ("please enter une of matt broken \n").
-	while (count 128) {
	int c= fget c (stdin),
	if (c== 1 h1) &
	line [count] = 'o';
	break;
	3 else if (c> 0 && c < 127) &
	line[count] = c;
	t+ count;
	3
	VTOSK Delay (10 / portTICK-MPERIOD_MS);
	printf (4 broker un1: " youn" line).
	3 else &
	ESP LOGE (TAG, "configuration mismatch: wrong broken and").
	abort ().
	3
	# endif
	U



esp_matt_client_handle + client = esp_matt_client_init(&matt_cfg). esp_matt_client_register_event Colient, ESP_EVENT_ANY IP matterent handler, client); esp chant matt client start (Syclient). 4 Void app main () & ESP_LOGI (TAG, "CAPPT Startup..."). ESP_LOGI (TAG, "TAPP] free memory: Yd bytes", esp get free heap vizel ESP_LOGI (TAG, "EAPPT IDF version: Y.s", esp-get idf-version ()) esplog level set ("*", ESP LOS INFO). esp log level set ("MOTT_CLIENT" ESP_LOG_VERBOSE). esp log level set ("MOTT EXAMPLE", ESP LOG VERBOSE). esp log level set ("TRANSPORT_TCP", ESP_LOG_VERBOSE); esp log-level cet ("TRAMLPORT-CSL", ESP LOG VERBOLE). esp log level set ("TRANSPORT", ESP 1000 VERBOSE). esp_log_level_set ("OUTBOX" ESP_LOG_VERBOLE). ECP-ERROR CHECK (nvc. flash ihit()). topip-adapter init(); ESP- FRROR CHECK (esp-event-loop-create default ()). ESP_ERROR_CHECK (example_connect()). matt-app-start ().