

```
1 file: 5e-groupe/tp.02/buttons.c
2 /*
3  * buttons.c
4  *
5  * Created on: Oct 1, 2018
6  * Author: sven
7  * Modified: Marc
8  */
9
10 #include <am335x_gpio.h>
11
12 #define GPIO1      AM335X_GPIO1
13 #define S1         (15)
14 #define S2         (16)
15 #define S3         (17)
16
17 // macro to compute number of elements of an array
18 #define ARRAY_SIZE(x) (sizeof(x) / sizeof(x[0]))
19
20 static const struct gpio_init_in {
21     uint32_t pin_nr;
22 } gpio_init_in[] = { { S1 }, { S2 }, { S3 } };
23
24 void buttons_init() {
25     am335x_gpio_init(GPIO1);
26
27     //set buttons in
28     for (uint32_t i = 0; i < ARRAY_SIZE(gpio_init_in); ++i) {
29         am335x_gpio_setup_pin_in(GPIO1, gpio_init_in[i].pin_nr,
30                                 AM335X_GPIO_PULL_NONE, false);
31         am335x_gpio_change_state(GPIO1, gpio_init_in[i].pin_nr, 1);
32     }
33 }
34
35 uint32_t get_states_buttons() {
36
37     return am335x_gpio_get_states(GPIO1);
38 }
39
40
```

en tête !

pas utile !

*ok, mais ici il faut judicieusement
décommenter la sortie, ht 0:2
et non pas 15:1 !*

```
68 file: 5e-groupe/tp.02/leds.c
69 /*
70  * leds.c
71  *
72  * Created on: Oct 1, 2018
73  * Author: sven
74  * Modified: Marc
75  */
76
77 #include <am335x_gpio.h>
78 #include <stdio.h>
79
80 #define LED1 (12)
81 #define LED2 (13)
82 #define LED3 (14)
83 #define GPIO1 AM335X_GPIO1
84
85 #define ARRAY_SIZE(x) (sizeof(x) / sizeof(x[0]))
86
87 static const struct gpio_init_leds {
88     uint32_t pin_nr;
89 } gpio_init_leds[] = { {LED1}, {LED2}, {LED3} };
90
91 void leds_init() {
92     am335x_gpio_init(GPIO1);
93     for (uint32_t i = 0; i < ARRAY_SIZE(gpio_init_leds); ++i) {
94         am335x_gpio_setup_pin_out(GPIO1, gpio_init_leds[i].pin_nr, false);
95         am335x_gpio_change_state(GPIO1, gpio_init_leds[i].pin_nr, false);
96     }
97 }
98
99 void set_state_by_led(int pin_led, bool state) {
100     am335x_gpio_change_state(GPIO1, pin_led, state);
101 }
102
```

pin nr!

enum!

```
124 file: 5e-groupe/tp.02/main.c
125 /**
126  * Copyright 2018 University of Applied Sciences Western Switzerland / Fribourg
127  *
128  * Licensed under the Apache License, Version 2.0 (the "License");
129  * you may not use this file except in compliance with the License.
130  * You may obtain a copy of the License at
131  *
132  * http://www.apache.org/licenses/LICENSE-2.0
133  *
134  * Unless required by applicable law or agreed to in writing, software
135  * distributed under the License is distributed on an "AS IS" BASIS,
136  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
137  * See the License for the specific language governing permissions and
138  * limitations under the License.
139  *
140  * Project:      HEIA-FR / Embedded Systems 2 Laboratory
141  *
142  * Abstract:      On this project we will start using the different tools on our BeagleBone like the
143  * rotative encoder and the double seven digits and also the 3 button selector.
144  *
145  *
146  * Author:       Marc Roten / Sven Rouvinez
147  * Date:         2018-10-01
148  */
149
150 #include <stdio.h>
151 #include <stdint.h>
152 #include <stdbool.h>
153 #include "wheel.h"
154 #include "serpent.h"
155 #include "seg7.h"
156 #include "buttons.h"
157 #include "leds.h"
158
159 // -- constants & variable declaration -----
160
161
162 #define S1          (1<<15)
163 #define S2          (1<<16)
164 #define S3          (1<<17)
165 #define LED1        (12)
166 #define LED2        (13)
167 #define LED3        (14)
168
169
170
171 // macro to compute number of elements of an array
172 #define ARRAY_SIZE(x) (sizeof(x) / sizeof(x[0]))
173
174 static int actual_number = 0;
175 static int snakeState = 0;
176
177 void initialize() {
178     actual_number = 0;
179
180     buttons_init();
181     leds_init();
182     wheel_init();
183     seg_init();
184 }
185
186 void start_counter() {
187
188     enum wheel_direction wheel_dir = get_wheel_direction();
189
190     while (true) {
191
192     wheel_dir = get_wheel_direction();
193     set_number(actual_number);
194
195     uint32_t button_state = get_states_buttons();
196     if (wheel_dir == WHEEL_LEFT) {

```

```
272
273     if ((button_state & S1) == 0) {
274         set_state_by_led(LED1, true);
275         set_state_by_led(LED2, false);
276         set_state_by_led(LED3, false);
277
278         start_counter();
279     } else if ((button_state & S2) == 0) {
280         set_state_by_led(LED1, false);
281         set_state_by_led(LED2, true);
282         set_state_by_led(LED3, false);
283         start_snake();
284     } else if ((button_state & S3) == 0) {
285         set_state_by_led(LED1, false);
286         set_state_by_led(LED2, false);
287         reset_functions();
288
289     }
290 }
291
292 return 0;
293
294 }
295
296
```

```
371 am335x_gpio_init(GPI00);
372 am335x_gpio_init(GPI01);
373 am335x_gpio_init(GPI02);
374
375 //init 7 segments
376 for (uint32_t i = 0; i < ARRAY_SIZE(gpio_init_7seg); ++i) {
377     am335x_gpio_setup_pin_out(GPI00, gpio_init_7seg[i].pin_nr, false);
378 }
379
380 //init decimal point
381 am335x_gpio_setup_pin_out(GPI02, 4, false);
382 am335x_gpio_change_state(GPI02, DP1, false);
383 am335x_gpio_setup_pin_out(GPI02, 5, false);
384 am335x_gpio_change_state(GPI02, DP2, false);
385
386 //init selectors
387 for (uint32_t i = 0; i < ARRAY_SIZE(gpio_init_DIG); ++i) {
388     am335x_gpio_setup_pin_out(GPI02, gpio_init_DIG[i].pin_nr, false);
389     am335x_gpio_change_state(GPI02, gpio_init_DIG[i].pin_nr, false);
390 }
391 }
392
393 }
```

regrouper sup!

permette

```
394
395 static struct digit split_number(int number) {
396
397     int number_nABS = number;
398
399     number = abs(number);
400
401     struct digit digit_to_print = { number / 10, number % 10, false };
402
403     if (number_nABS < 0)
404         digit_to_print.negative = true;
405     else
406         digit_to_print.negative = false;
407
408     return digit_to_print;
409 }
410
```

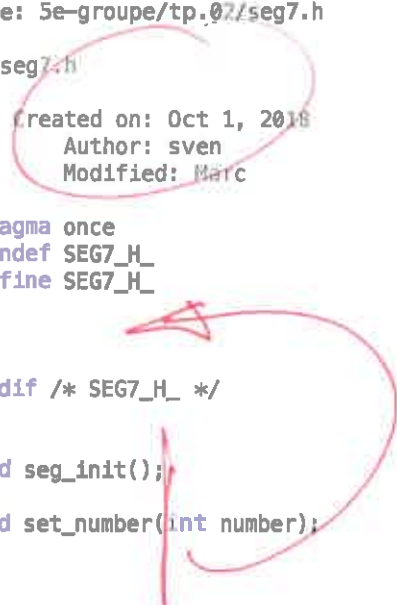
⇒ counter!

```
411 void refresh_display() {
412
413     static unsigned digit = 0;
414     //switch between dig
415     digit = (digit + 1) % 2;
416
417     am335x_gpio_change_states(GPI00, array_digits[0], false);
418     am335x_gpio_change_state(GPI02, DP1, false);
419
420     switch (digit) {
421         //turn on dig1
422         case 0:
423
424             am335x_gpio_change_state(GPI02, DIG2, false);
425             am335x_gpio_change_state(GPI02, DIG1, true);
426             am335x_gpio_change_states(GPI00, digit1, true);
427
428             if (dp)
429                 am335x_gpio_change_state(GPI02, DP1, true);
430
431             break;
432
433         //turn on dig2
434         case 1:
435
436             am335x_gpio_change_state(GPI02, DIG1, false);
437             am335x_gpio_change_state(GPI02, DIG2, true);
438             am335x_gpio_change_states(GPI00, digit2, true);
439             break;
440     }
441 }
442
```

utiliser un tableau
⇒ voir solution!

```
443 void set_number(int number) {
444
```

```
459 file: 5e-groupe/tp.02/seg7.h
460 /*
461  * seg7.h
462  *
463  * Created on: Oct 1, 2018
464  * Author: sven
465  * Modified: Marc
466  */
467 #pragma once
468 #ifndef SEG7_H_
469 #define SEG7_H_
470
471
472
473 #endif /* SEG7_H_ */
474
475
476 void seg_init();
477
478 void set_number(int number);
479
480
```



```
553 file: 5e-groupe/tp.02/serpent.h
554 /*
555  * serpent.h
556  *
557  * Created on: Oct 1, 2018
558  * Author: sven
559  * Modified: Marc
560  */
561 #pragma once
562 #ifndef SERPENT_H_
563 #define SERPENT_H_
564
565
566
567 #endif /* SERPENT_H_ */
568 void displaySnake(int snakeState, bool value_change);
569 uint32_t get_size_path();
570
571
```

```
635 file: 5e-groupe/tp.02/wheel.h
636 /*
637  * wheel.h
638  *
639  * Created on: Oct 1, 2018
640  * Author: sven
641  * Modified : Marc
642  *
643  */
644 #pragma once
645 #ifndef WHEEL_H_
646 #define WHEEL_H_
647
648
649
650 #endif /* WHEEL_H_ */
651
652 enum wheel_direction {
653     WHEEL_STILL, WHEEL_RIGHT, WHEEL_LEFT
654 };
655
656 void wheel_init();
657
658 enum wheel_direction get_wheel_direction();
659
```

