

# Assignment 1

## Embedded Programming

**Anne-Charlotte  
Aske  
Victor**

SDU Robotics  
The Maersk Mc-Kinney Moller Institute  
University of Southern Denmark

# Agenda



## Modules

RGB

Switch 1

System Timer

## Main

Helper function

Debounce

States



```
25  /* *****  
26  #define RGB_COLOR_NONE      0x0  
27  #define RGB_COLOR_GREEN    0x8  
28  #define RGB_COLOR_BLUE     0x4  
29  #define RGB_COLOR_CYAN     0xC  
30  #define RGB_COLOR_RED       0x2  
31  #define RGB_COLOR_YELLOW    0xA  
32  #define RGB_COLOR_MEGANTA   0x6  
33  #define RGB_COLOR_WHITE     0xE
```

## *Defines*

```
*****
```



```
39  /***** Function interfaces *****/
40  void rgb_setup(void);
41  /*****
42  *   Input      : —
43  *   Output     : —
44  *   Function   : Setup RGB colors
45  *****/
46
47  void rgb_set_color(uint32_t);
48  /*****
49  *   Input      : Color
50  *   Output     : —
51  *   Function   : Set color of RGB LEDs
52  *****/
```



```
32 void rgb_set_color(uint32_t color)
33 /* ****
34 *   Input      : Color
35 *   Output     :
36 *   Function   : Change color on RGB LEDs
37 **** */
38 {
39     GPIO_PORTF_DATA_R &= ~RGB_PINS;
40     GPIO_PORTF_DATA_R |= color & RGB_PINS;
41 }
```

# Switch 1

## Function interfaces



```
31 void sw1_setup();
32 /* ****
33 *   Input   : Nothing
34 *   Output  : Nothing
35 *   Function : To setup the necessary pins and values for sw1
36 **** */
37 bool sw1_read();
38 /* ****
39 *   Input   : Nothing
40 *   Output  : State of switch
41 *   Function : To figure out rather or not the switch is on or off.
```

# System Timer

## Function interfaces



```
30  /***** Function interfaces *****/
31  void timer_setup();
32  /*****
33  *   Input      : Nothing
34  *   Output     : Nothing
35  *   Function   : To setup the necessary pins and values for the systick timer. The
                    period being 1ms.
36  *****/
37
38  uint32_t timer_get();
39  /*****
40  *   Input      : Nothing
41  *   Output     : Outputs the absolute time.
42  *   Function   : This function returns the absolute time.
43  *****/
```

# System Timer

## Function interfaces



```
44
45 void timer_handler();
46 /* ****
47 *   Input      : Nothing
48 *   Output     : Nothing
49 *   Function   : The interrupt handler for systick, which increments a counter.
50 **** */
```



# Main

## Defines



```
28 #define THRESHHOLD_LONG          2000    // ~2s
29 #define THRESHHOLD_DOUBLE_CLICK  300      // ~.3s
30 #define BTN_DBOUNCE_COUNT         3
31 #define NUM_COLORS                 8
32 #define AUTO_PERIOD                200     // ~.2s
33 #define FALSE                      0
34 #define TRUE                       1
35 #define DIRECTION_M                0x1
```

# Main

Helper function



```
55 void color_change()  
56 /* ****  
57 *   Input   :  
58 *   Output  :  
59 *   Function : Changes the color based on direction variable in a predetermined  
60 *               pattern  
61 **** */  
62 {  
63     if ( direction )  
64     {  
65         ++color;  
66         color %= NUM_COLORS;  
67     } else  
68     {  
69         color--;  
70         if ( color < 0)  
71             color = NUM_COLORS - 1;  
72     }  
73     rgb_set_color( colors[color] );  
74 }
```



```
99     if( sw1_read() )
100     {
101         btn_cnt++;
102         valid_click = ( (btn_cnt >= BTN_DBOUNCE_COUNT) && (has_cleared == TRUE) )?
                        TRUE : FALSE;
103     } else
104     {
105         btn_cnt      = 0;
106         has_cleared  = TRUE;
107     }
```

```
109  if ( valid_click )
110  {
111      has_cleared = FALSE;
112      valid_click = FALSE;
113      btn_time    = timer_get();
114
115      do
116      {
117          btn_holdtime = timer_get() - btn_time;
118      } while( sw1_read() && (btn_holdtime <= THRESHHOLD_LONG) );
119
120      // Long press
121      if( btn_holdtime >= THRESHHOLD_LONG )
122      {
123          auto_mode = TRUE;
124      }
```



```
125      //Double click
126      else if ( (timer_get() - btn_prev_time) < THRESHHOLD_DOUBLE_CLICK )
127      {
128          // Toggle Direction
129          direction ^= DIRECTION_M;
130      }
131      // Single press
132      else
133      {
134          auto_mode      = FALSE;
135          btn_prev_time  = timer_get();
136          color_change();
137      }
138  }
```



```
139     else if ( auto_mode && ((timer_get() % AUTO_PERIOD) == 0) )
140     {
141         color_change();
142     }
```