

Assignment 1

Embedded Programming

**Anne-Charlotte
Aske
Victor**

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

Agenda





```
1  /* *****  
2  #define RGB_COLOR_NONE      0x0  
3  #define RGB_COLOR_GREEN    0x8  
4  #define RGB_COLOR_BLUE     0x4  
5  #define RGB_COLOR_CYAN     0xC  
6  #define RGB_COLOR_RED      0x2  
7  #define RGB_COLOR_YELLOW   0xA  
8  #define RGB_COLOR_MEGANTA  0x6  
9  #define RGB_COLOR_WHITE    0xE
```

Defines

```
*****
```



```
1  /***** Function interfaces *****/
2  void rgb_setup(void);
3  /*****
4  *   Input      : -
5  *   Output     : -
6  *   Function   : Setup RGB colors
7  *****/
8
9  void rgb_set_color(uint32_t);
10 /*****
11 *   Input      : Color
12 *   Output     : -
13 *   Function   : Set color of RGB LEDs
14 *****/
```



```
1 void rgb_set_color(uint32_t color)
2 /* *****
3 *   Input      : Color
4 *   Output      :
5 *   Function    : Change color on RGB LEDs
6 ***** */
7 {
8     GPIO_PORTF_DATA_R &= ~RGB_PINS;
9     GPIO_PORTF_DATA_R |= color & RGB_PINS;
10 }
```

Switch 1

Function interfaces



```
1 void sw1_setup() ;
2 /* **** */
3 *   Input   : Nothing
4 *   Output  : Nothing
5 *   Function : To setup the necessary pins and values for sw1
6 **** */
7 bool sw1_read() ;
8 /* **** */
9 *   Input   : Nothing
10 *   Output  : State of switch
11 *   Function : To figure out rather or not the switch is on or off.
```

System Timer

Function interfaces



```
1  /** ***** Function interfaces ***** */
2  void timer_setup();
3  /** *****
4   *   Input      : Nothing
5   *   Output     : Nothing
6   *   Function   : To setup the necessary pins and values for the systick timer. The
   *                 period being 1ms.
7   ***** */
8
9  uint32_t timer_get();
10 /** *****
11 *   Input      : Nothing
12 *   Output     : Outputs the absolute time.
13 *   Function   : This function returns the absolute time.
14 ***** */
```

System Timer

Function interfaces



```
1
2 void timer_handler();
3 /* ****
4  *   Input      : Nothing
5  *   Output     : Nothing
6  *   Function   : The interrupt handler for systick, which increments a counter.
7  **** */
```


Main

Helper function



```
1 void color_change()
2 /* *****
3  *   Input   :
4  *   Output  :
5  *   Function : Changes the color based on direction variable in a predetermined
6  *               pattern
7  * *****/
8 {
9     if ( direction )
10    {
11        ++color;
12        color %= NUM_COLORS;
13    } else
14    {
15        color--;
16        if (color < 0)
17            color = NUM_COLORS - 1;
18    }
19    rgb_set_color( colors[color] );
20 }
```

Main

debounce



```
1      if( sw1_read() )
2      {
3          btn_cnt++;
4          valid_click = ( (btn_cnt >= BTN_DBOUNCE_COUNT) && (has_cleared == TRUE) )?
                        TRUE : FALSE;
5      } else
6      {
7          btn_cnt      = 0;
8          has_cleared = TRUE;
9      }
```



```
1  if ( valid_click )
2  {
3      has_cleared = FALSE;
4      valid_click = FALSE;
5      btn_time=timer_get();
6
7      do
8      {
9          btn_holdtime = timer_get() - btn_time;
10         } while( sw1_read() && (btn_holdtime <= THRESHHOLD_LONG) );
11
12         // Long press
13         if( btn_holdtime >= THRESHHOLD_LONG )
14         {
15             auto_mode = TRUE;
16         }
```



```
1      //Double click
2      else if( (timer_get() - btn_prev_time) < THRESHHOLD_DOUBLE_CLICK )
3      {
4          // Toggle Direction
5          direction ^= 0x1;
6      }
7      // Single press
8      else
9      {
10         auto_mode = FALSE;
11         btn_prev_time=timer_get();
12         color_change();
13     }
14 }
```

States

Auto



```
1     else if ( auto_mode && ((timer_get() % AUTO_PERIOD) == 0) )  
2     {  
3         color_change();  
4     }
```