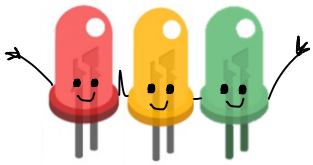
Team Name	Performance	Stylish Appearance	Complexity Bonus	Total Points	Award
Kamiller	10	1.5	2	13.5	Cinema Tickets
ARMless	10	2	1	13	Coffee
R2D2	10	1.5		11.5	Coffee
Catalyzer	10	0.5	0.5	11	Ice Cream
pepega	10	0.5		10.5	Ice Cream
Group++	6	1		7	Chocolate
SYS HACETTEPE	6			6	Chocolate
Power On	6			6	Chocolate
Power off	6		-1	6	Chocolate
gigadeth	6			5	Chocolate
kendini marul sanan marul	0		3	3	Appropriate punishment

LETTUCE CELEBRATE



EMBEDDED SYSTEMS LAB.

LAB. REPORT IV

TEAM NAME:

KENDINI MARUL SANAN MARUL

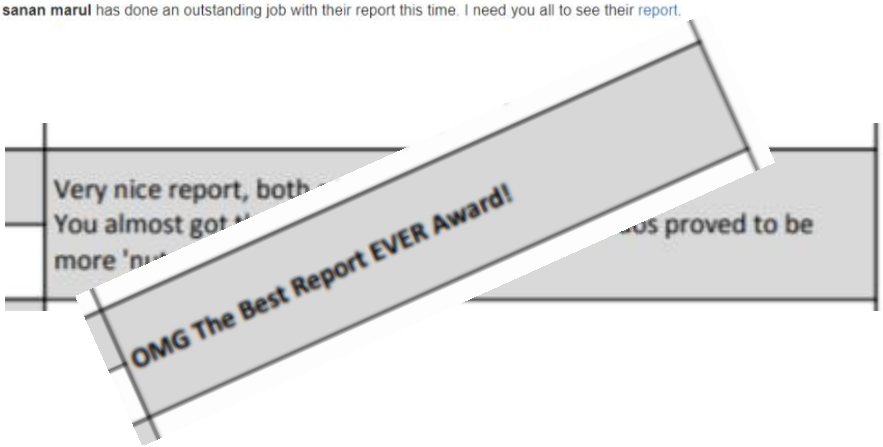
MEMBERS:

KUTAY BARCIN 21526715
DEFNE TUNCER 21627686

HACETTEPE UNIVERSITY

COMPUTER SCIENCE DEPARTMENT

I just have to make one other honorable mention this time: group **kendini marul sanan marul** has done an outstanding job with their report this time. I need you all to see their report.

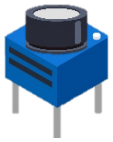


APPROACH LIGHTING SYSTEM

Three LEDs are lightened up with given order for 0.5 seconds each. With every switch press loop is reversed. Two ISR routines are implemented one for Systick timer and one for switch.
LOOP: GREEN, YELLOW, RED

STEP1: EQUIPMENT

LEDS: Three 1.8V 2mA LEDs
$R = (V_{OH} – V_d)/I_d$
$= (3.0 – 1.8)/0.002$
$= 600\Omega$



SWITCH: One B3F Tactile Switch
Switch is pressed: -0.1 mΩ
Switch is not pressed: +100 mΩ

RESISTORS: Three 470Ω resistors

RESISTORS: One 10kΩ resistor

Using 470Ω resistors:
$470\Omega = (3.0 – 1.8)/I_d$ $I_d = 0.0026$
Brighter light from LEDs is expected.

STEP1: I/O

One input for the switch and three outputs for the LEDs are decided.

		ON	OFF
INPUTS:	PA5		
OUTPUTS:	PA4, PA6, PA7	PA4: 0x10	PA6: 0x00
		PA6: 0x40	PA7: 0x00
		PA7: 0x80	PA4: 0x00

PORTA initialization is done: For the switch interrupts, **falling edge** event is used.

```
#define PA4 (*((volatile unsigned long *)0x40004040))
#define PA5 (*((volatile unsigned long *)0x40004080))
#define PA6 (*((volatile unsigned long *)0x40004100))
#define PA7 (*((volatile unsigned long *)0x40004200))

void PORTA_Init_4567(void){volatile unsigned long delay;
  SYSCTL_RCGCGPIO_R |= 0x00000001;      // activate clock for Port A
  delay = SYSCTL_RCGCGPIO_R;
  GPIO_PORTA_AMSEL_R &= ~0xF0;          // disable analog on PA4-7
  GPIO_PORTA_PCTL_R &= ~0xFFFF0000;     // PCTL GPIO on PA4-7
  GPIO_PORTA_DIR_R = 0xD0;              // direction PA5 as input PA4, PA7, PA6 as output
  GPIO_PORTA_AFSEL_R &= ~0xF0;          // PA4-7 regular port function
  GPIO_PORTA_DEN_R |= 0xF0;             // enable PA4-7 digital port
  GPIO_PORTA_IS_R &= ~0x20;             // PA5 is edge-sensitive
  GPIO_PORTA_IBE_R &= ~0x20;            // PA5 is not both edges
  GPIO_PORTA_IEV_R &= ~0x20;            // PA5 falling edge event
  GPIO_PORTA_ICR_R = 0x20;              // clear flag5
  GPIO_PORTA_IM_R |= 0x20;              // arm interrupt on PA5
  NVIC_PRI0_R = (NVIC_PRI0_R&0xFFFFFF00)|0x00000040; // priority 2
  NVIC_EN0_R = 0x00000001;              // enable interrupt 30 in NVIC
}
```

STEP2: SYSTICK TIMER

Loops are measured using SysTick Interrupts.

SYSTICK TIMER initialization is done:

```
void SysTick_Init(unsigned long period){
  NVIC_ST_CTRL_R = 0;          // disable SysTick during setup
  NVIC_ST_RELOAD_R = period-1; // reload value
  NVIC_ST_CURRENT_R = 0;       // any write to current clears it
  NVIC_SYS_PRI3_R = (NVIC_SYS_PRI3_R&0x00FFFFFF)|0xA0000000; // priority 5
  NVIC_ST_CTRL_R = 0x07;       // enable SysTick with core clock and interrupts
}
```

SYSTICK_HANDLER ISR ROUTINE

External circuit of the switch is implemented as **negative logic** and LEDs are as **positive logic**. Every 0.5 second only one LED is on and the color is decided from the value of **pressed**.

```
int pressed;

void SysTick_Handler(void){
  if (pressed){
    if (PA4&0x10){PA4 = 0x00;PA6 = 0x40;}        //RED OFF YELLOW ON
    else if (PA6&0x40){PA6 = 0x00; PA7 = 0x80;} //YELLOW OFF GREEN ON
    else{PA7 = 0x00;  PA4 = 0x10;}               //GREEN OFF RED ON
  }
  else{
    if (PA7&0x80){ PA7 = 0x00; PA6 = 0x40;}       //GREEN OFF YELLOW ON
    else if (PA6&0x40){ PA6 = 0x00; PA4 = 0x10;} //YELLOW OFF RED ON
    else{ PA4 = 0x00; PA7 = 0x80;}                //RED OFF GREEN ON
  }
}
```

STEP3: SWITCH

From observations bounce of switch is detected once it is pressed. In order to prevent this, value of **pressed** is compared with itself after 10ms delay. No change in value means no bounce, so **pressed** can be toggled.

GPIOPORTA_HANDLER ISR ROUTINE

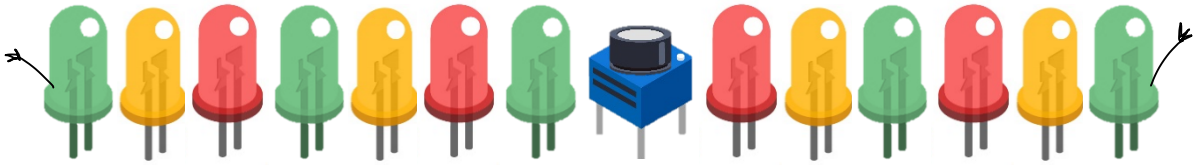
```
int bounce;

void GPIOPortA_Handler(void){
  GPIO_PORTA_ICR_R = 0x20;      // acknowledge flag5
  bounce = pressed;
  Delay(0.001);
  if(pressed==bounce)
    pressed ^= 1;
}
```

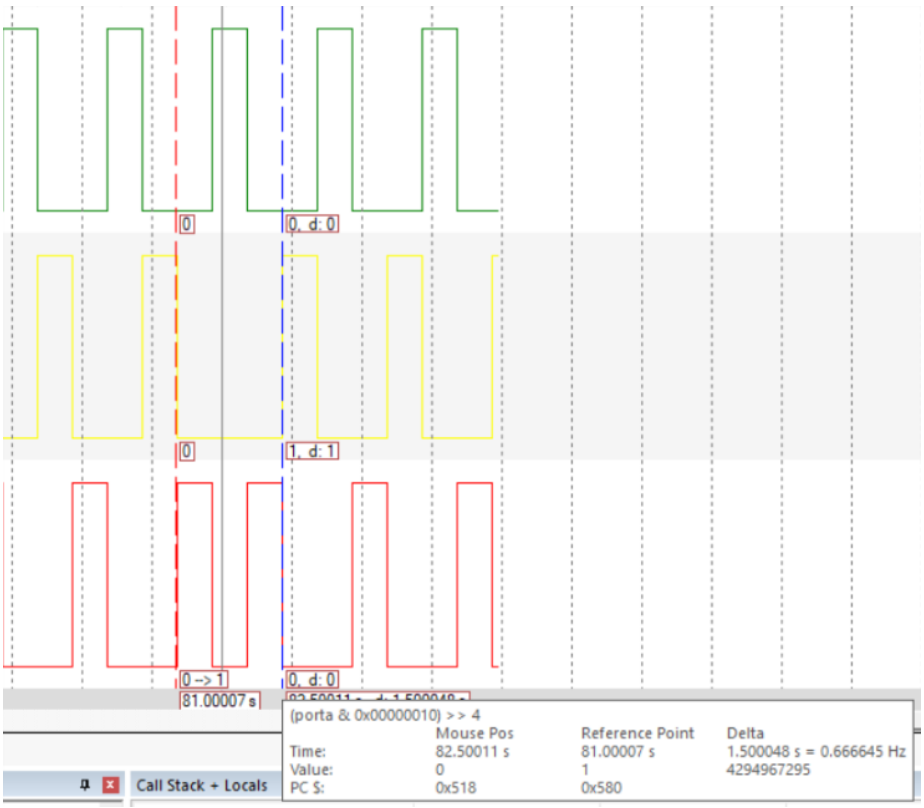
STEP4: PROGRAM

```
int main(void){volatile unsigned long delay;
  pressed = 0;
  PORTA_Init_4567();
  SysTick_Init(8000000);        // initialize SysTick timer, every 500ms
  EnableInterrupts();              // enable after everything initialized
  while(1){WaitForInterrupt();}
```

STEP5: SIMULATION

WITH EVERY SWITCH PRESS, LOOP IS REVERSED.

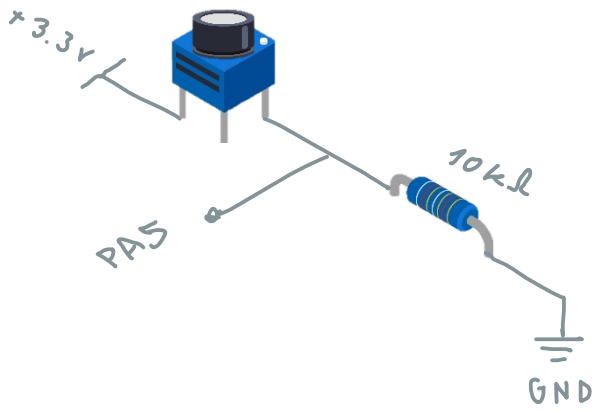


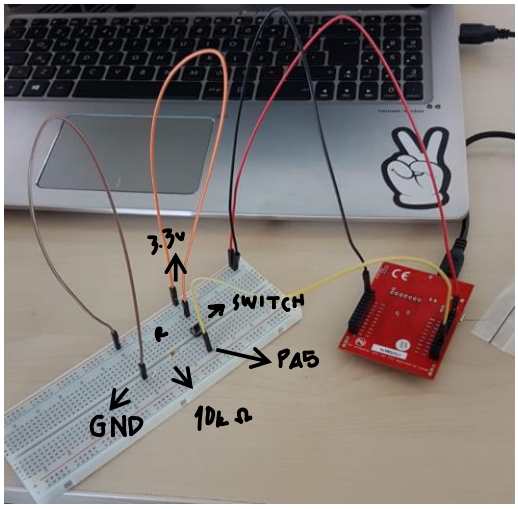


YELLOW LED STAYS OFF FOR 1.5 SECONDS.

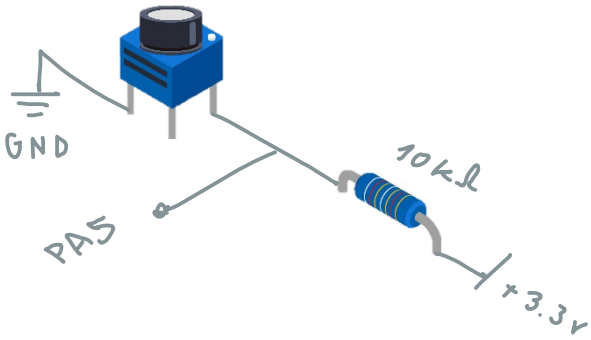
STEP6: SWITCHES

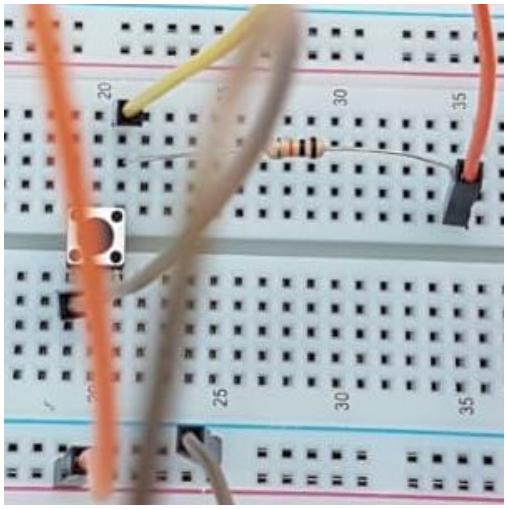
1. BUILD POSITIVE LOGIC SWITCH





2. BUILD NEGATIVE LOGIC SWITCH

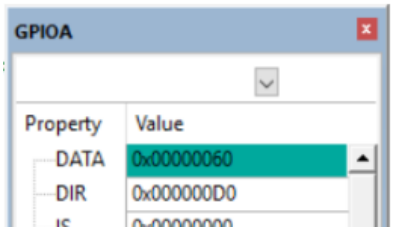


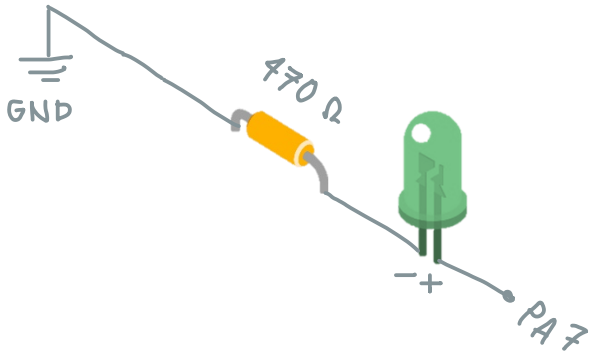


STEP7: LEDS

1. BUILD POSITIVE LOGIC LEDS

While switch is not pressed (PA5 is 1) and yellow LED (PA6) is lightened up GPIOA DATA register is read 0x00000060.





STEP8: CIRCUIT DESIGN