ERTS Final Assessment – Task 1 Presentation & Demo

Group 15

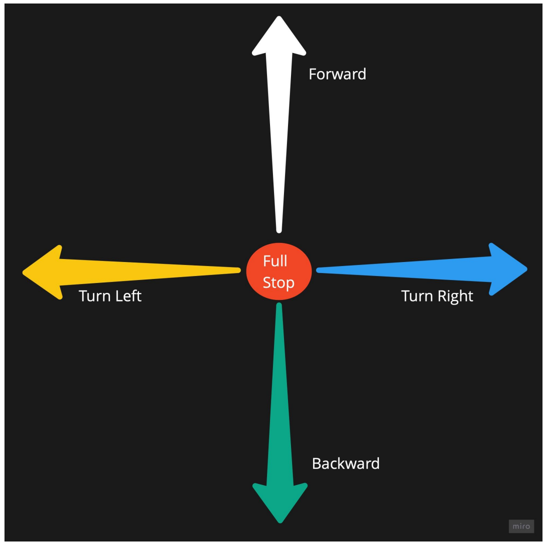
Gytis Budrevicius	km21822
Smita Yashwant Nangare	fc23356
Hongbing Qiu	mk20661
Yumu Xie	po21744

Practical elements

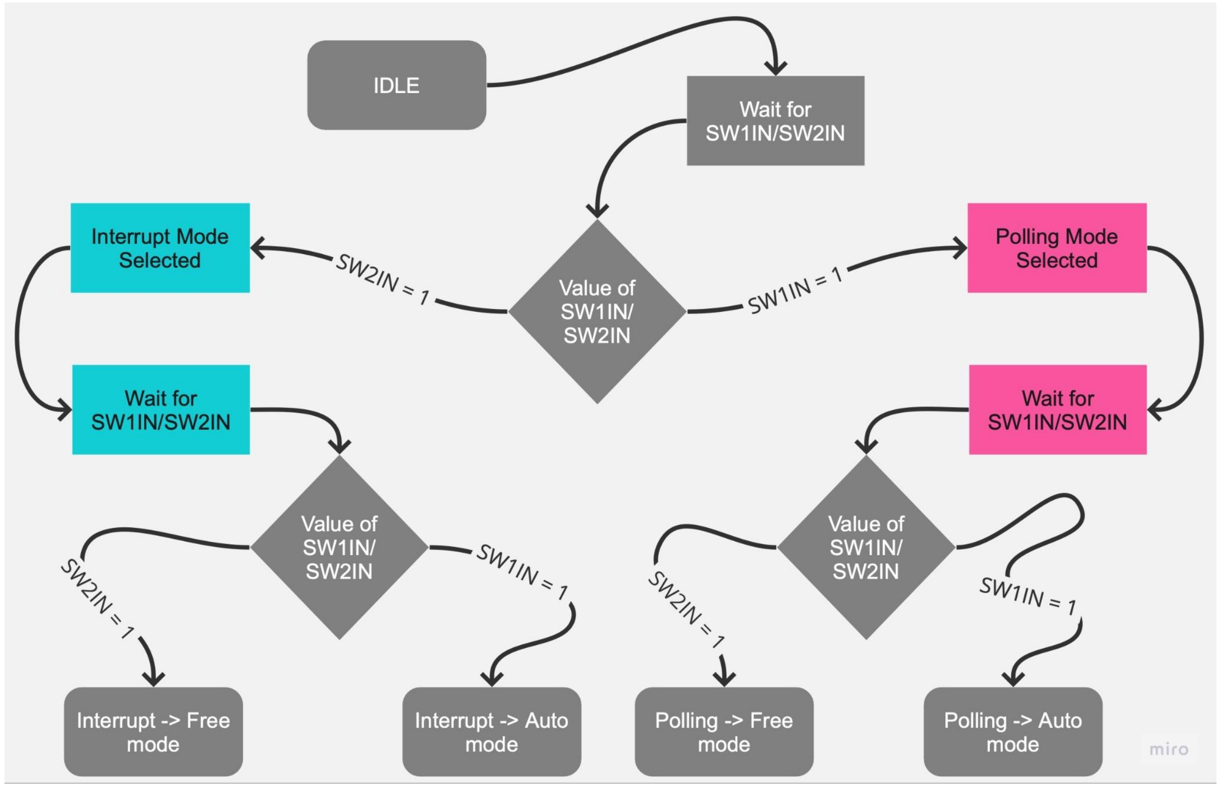
- Show Operation and Show how they different

Usage of LEDs:
Pink: Polling mode
Sky blue: Interrupt mode

Movement LED indicators:

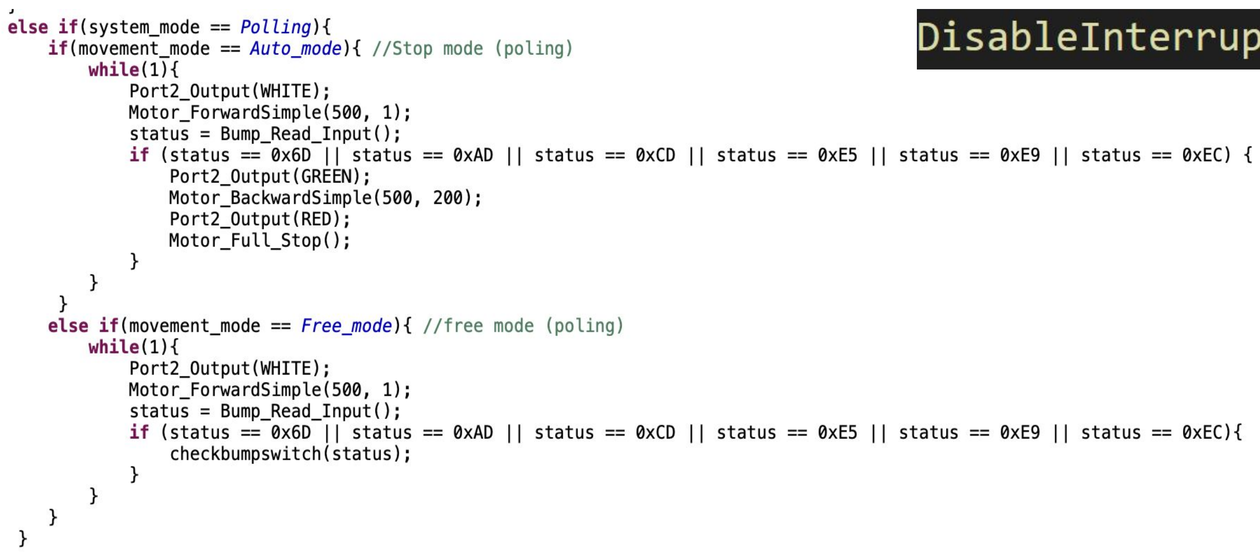


Operation Mode Selection:

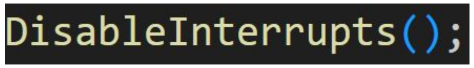


Interrupts

- How do you implement operation of robot with polling? Why?

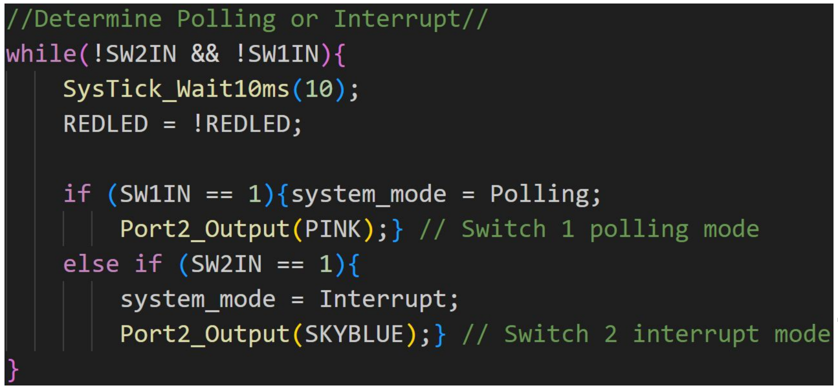


```
else if(system_mode == Polling){
    if(movement_mode == Auto_mode){ //Stop mode (poling)
        while(1){
            Port2_Output(WHITE);
            Motor_ForwardSimple(500, 1);
            status = Bump_Read_Input();
            if (status == 0x6D || status == 0xAD || status == 0xCD || status == 0xE5 || status == 0xE9 || status == 0xEC) {
                Port2_Output(GREEN);
                Motor_BackwardSimple(500, 200);
                Port2_Output(RED);
                Motor_Full_Stop();
            }
        }
     }
    else if(movement_mode == Free_mode){ //free mode (poling)
        while(1){
            Port2_Output(WHITE);
            Motor_ForwardSimple(500, 1);
            status = Bump_Read_Input();
            if (status == 0x6D || status == 0xAD || status == 0xCD || status == 0xE5 || status == 0xE9 || status == 0xEC){
                checkbumpswitch(status);
            }
        }
    }
 }
```

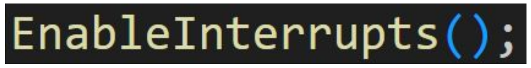


```
DisableInterrupts();
```

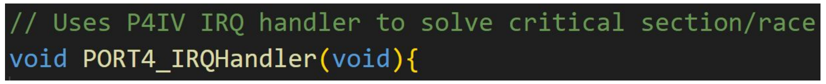
- How do you implement operation of robot with interrupts? Why?



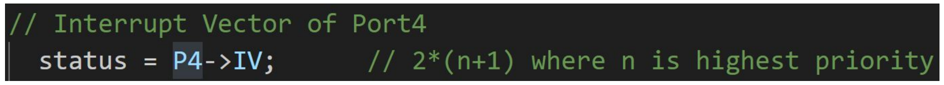
```
//Determine Polling or Interrupt//
while(!SW2IN && !SW1IN){
    SysTick_Wait10ms(10);
    REDLED = !REDLED;

    if (SW1IN == 1){system_mode = Polling;
        Port2_Output(PINK);} // Switch 1 polling mode
    else if (SW2IN == 1){
        system_mode = Interrupt;
        Port2_Output(SKYBLUE);} // Switch 2 interrupt mode
}
```



```
EnableInterrupts();
```



```
// Uses P4IV IRQ handler to solve critical section/race
void PORT4_IRQHandler(void){
```

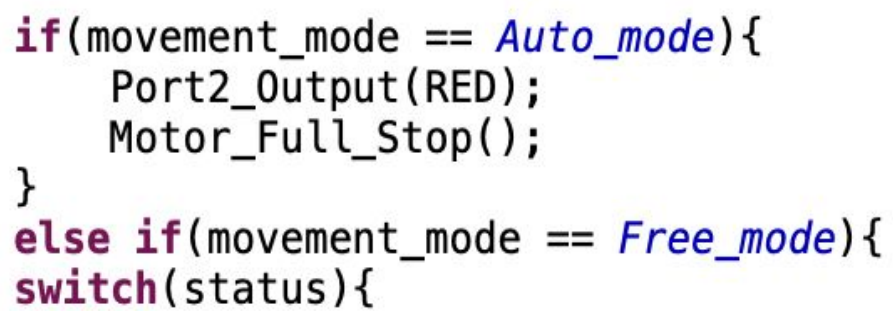


```
// Interrupt Vector of Port4
  status = P4->IV;      // 2*(n+1) where n is highest priority
```

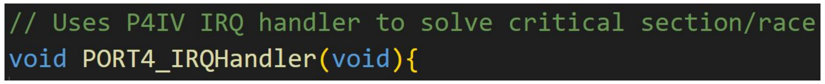
Interrupts

- How have you kept the ISR minimal?

- We have added an if statement which reduces cycle count in the main while loop (inside void PORT4_IRQHandler(void), not main)

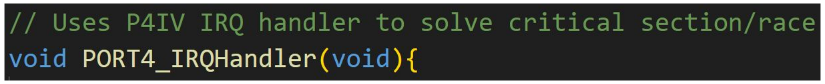


```
if(movement_mode == Auto_mode){
    Port2_Output(RED);
    Motor_Full_Stop();
}
else if(movement_mode == Free_mode){
switch(status){
```

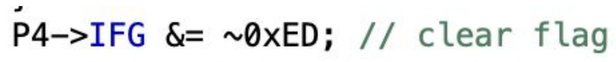


```
// Uses P4IV IRQ handler to solve critical section/race
void PORT4_IRQHandler(void){
```

- Cleared the interrupt flag to release resources & prevent interference



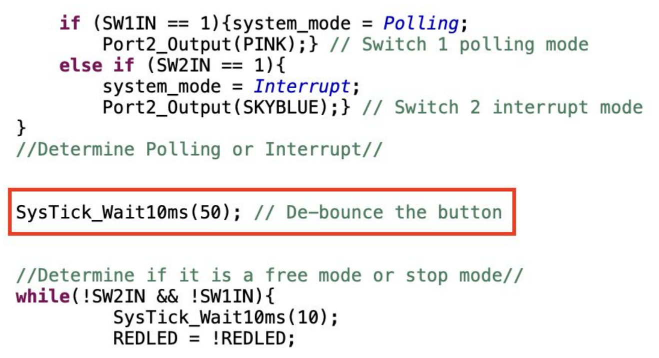
```
// Uses P4IV IRQ handler to solve critical section/race
void PORT4_IRQHandler(void){
```



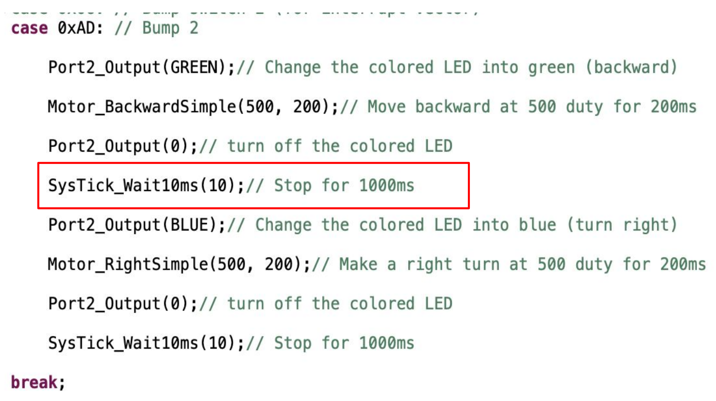
```
P4->IFG &= ~0xED; // clear flag
```

Delays

- Where have delays been used? Why?



```
    if (SW1IN == 1){system_mode = Polling;
        Port2_Output(PINK);} // Switch 1 polling mode
    else if (SW2IN == 1){
        system_mode = Interrupt;
        Port2_Output(SKYBLUE);} // Switch 2 interrupt mode
}
//Determine Polling or Interrupt//

SysTick_Wait10ms(50); // De-bounce the button

//Determine if it is a free mode or stop mode//
while(!SW2IN && !SW1IN){
         SysTick_Wait10ms(10);
         REDLED = !REDLED;
```



```
case 0xAD: // Bump 2

    Port2_Output(GREEN);// Change the colored LED into green (backward)

    Motor_BackwardSimple(500, 200);// Move backward at 500 duty for 200ms

    Port2_Output(0);// turn off the colored LED

    SysTick_Wait10ms(10);// Stop for 1000ms

    Port2_Output(BLUE);// Change the colored LED into blue (turn right)

    Motor_RightSimple(500, 200);// Make a right turn at 500 duty for 200ms

    Port2_Output(0);// turn off the colored LED

    SysTick_Wait10ms(10);// Stop for 1000ms

break;
```

- In main function , there is a delay(Systick_Wait10ms(50)) between two mode selection. This delay ensures that the user doesn't enters the error mode by pressing SW too fast or too long after selecting polling and interrupt modes.

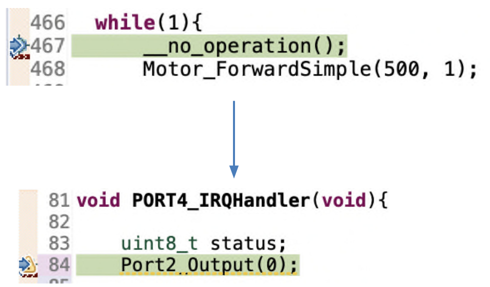
- There is a delay(SysTick_Wait10ms(10)) between changing directions of the robot. This delay can slow down the speed of the robot to make sure to turning process is successful. It also prevents the motor while fast turning direction operation.

CPU Usage

Two different approaches were done to compare the CPU usage in interrupt/polling modes:

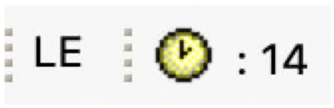
Detection of bump switches

Interrupt



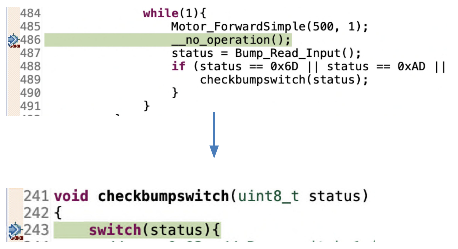
```
    while(1){
        __no_operation();
        Motor_ForwardSimple(500, 1);
```

```
void PORT4_IRQHandler(void){

    uint8_t status;
    Port2_Output(0);
```



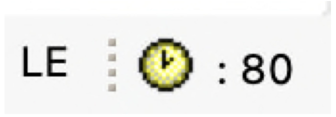
LE : 14

Polling

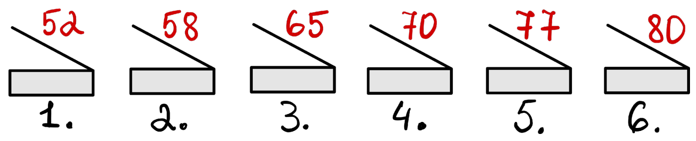


```
            while(1){
                Motor_ForwardSimple(500, 1);
                __no_operation();
                status = Bump_Read_Input();
                if (status == 0x6D || status == 0xAD ||
                    checkbumpswitch(status);
                }
            }
```

```
void checkbumpswitch(uint8_t status)
{
    switch(status){
```

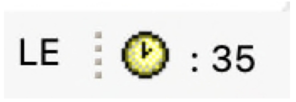


LE : 80

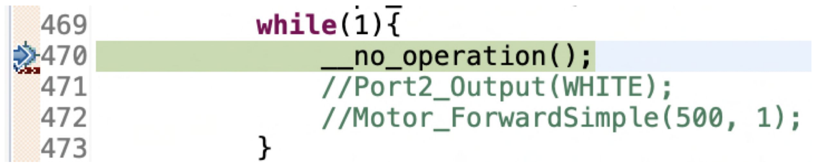


Waiting for bump switch trigger (5 iterations)

Interrupt

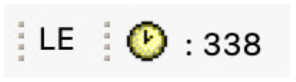


LE : 35

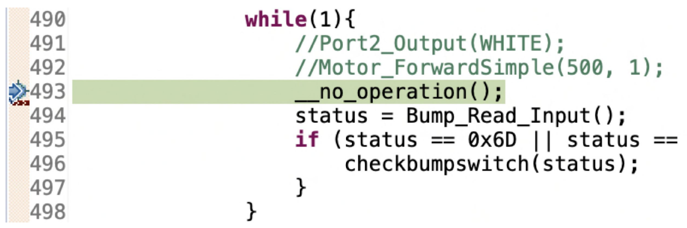


```
        while(1){
            __no_operation();
            //Port2_Output(WHITE);
            //Motor_ForwardSimple(500, 1);
        }
```

Polling



LE : 338



```
        while(1){
            //Port2_Output(WHITE);
            //Motor_ForwardSimple(500, 1);
            __no_operation();
            status = Bump_Read_Input();
            if (status == 0x6D  || status ==
                checkbumpswitch(status);
            }
        }
```