Programming Group assignment 4 (Week 5)

GROUP 8

```
//Week5-Assignment1
 #if 1
void zmain(void)
{
    RTC_Start(); // start real time clock
    int x,y;


    printf("enter the current hour:");
    scanf("%d",&x);
    printf("enter the current minute:");
    scanf("%d",&y);
    RTC_TIME_DATE now;
    // set current time
    now.Hour = x;
    now.Min = y;
   /* bool press = 0;*/
    RTC_WriteTime(&now); // write the time to real time clock

    for(;;)
    {
        if(SW1_Read() == 0) {
            // read the current time
            RTC_DisableInt(); /* Disable Interrupt of RTC Component */
            now = *RTC_ReadTime(); /* copy the current time to a local
variable */
            RTC_EnableInt(); /* Enable Interrupt of RTC Component */

            // print the current time

            printf("%2d:%02d\n", now.Hour, now.Min);
            print_mqtt("Zumo041/output", "%2d:%02d", now.Hour, now.Min);
            // wait until button is released
            while(SW1_Read() == 1);
        }
        vTaskDelay(50);
    }
 }
#endif

  //Week5-Assignment2
#if 1
//ultrasonic sensor//
void zmain(void)
{
    Ultra_Start();                          // Ultra Sonic Start function
    motor_start();
    uint8_t dir =0;



    while(1) {
    int d = Ultra_GetDistance();
    if (d<10){
        dir = rand() %2;
```

```
           motor_forward(0,0);
            motor_backward(100,500);
            if (dir ==0){
                motor_instant_right(150,150,300);
            print_mqtt("Zumo041/output", "robot turned right");
            }
            else{
                motor_instant_left(150,150,300);
            print_mqtt("Zumo041/output", "robot turned left");
            }

    motor_forward(100,0);



}
}
}
#endif
//Week5-Assignment3
#if 1
//reflectance
void zmain(void)
{

    struct sensors_ dig;
    struct sensors_ ref;
   /*uint8 button=1;*/
    TickType_t start=0;
    TickType_t end;
    int x;
    bool white = 0;
    reflectance_start();
    motor_start ();
    IR_Start();
    IR_flush();
    reflectance_set_threshold(9000, 9000, 11000, 11000, 9000, 9000); // set center sensor threshold to 11000 and others to 9000
    //button = SW1_Read(); // read SW1 on pSoC board
    // SW1_Read() returns zero when button is pressed
    // SW1_Read() returns one when button is not pressed



for (;;)
  {
    start = end;
    /*if ( SW1_Read()==0)
    {
    button= !button;
    vTaskDelay(200);
    }
    if (button==0)
    {
    */

    reflectance_digital(&dig);
    reflectance_read(&ref);

    motor_forward (100,0);
    if ( white ==1 && (dig.r3== 1 && dig.l3==1) )
    {
        white = 0;
        motor_forward(0,0);
        end = xTaskGetTickCount ();
        IR_wait();
        x=end-start;
        print_mqtt("Zumo041/output", "start %d", start);
        print_mqtt("Zumo041/output", "end number %d", end);
        print_mqtt("Zumo041/output", "lap number %d", x);
    }
    if ( dig.r3 == 0 && dig.l3 == 0)
    {
        white =1 ;

    }
     if (ref.l1 >=22000 && ref.r1 >=22000){
        motor_forward(100,1);}
    else if (ref.l1<=22000 && ref.r1 >22000){
        motor_turn(150,10,1);}
    else if (ref.l1 >=22000 && ref.r1 <22000){
        motor_turn(10,150,1);}

}
}
  #endif
```