

How to use Microcontrollers for mobile robots

Exercise 1:

Code:

```
/*
 * Exercise 1: Interrupt
 * Requirements:
 *   • Use a push button as a hardware interrupt.
 *   • Use timer interrupt every 1 sec to count up number by 1
     which starts from 0. (Declare the variable named count.)
 *   • In loop function, display the number of the count on
     Serial monitor.
 *   • When the bush button is pushed, the number of count
     will be set as 0 at any time.
 * Constrains:
 *   • Do not use digitalWrite or analogRead commands.
 *   • Do not use delay command
 */
#include <MsTimer2.h>

#define BUTTON 2

volatile unsigned int count = 0;
unsigned int lastCount = 99999;

void setup()
{
    pinMode(BUTTON, INPUT_PULLUP);
    Serial.begin(115200);

    // define Timer Interrupt
    MsTimer2::set(1000, increaseCount);
    MsTimer2::start();

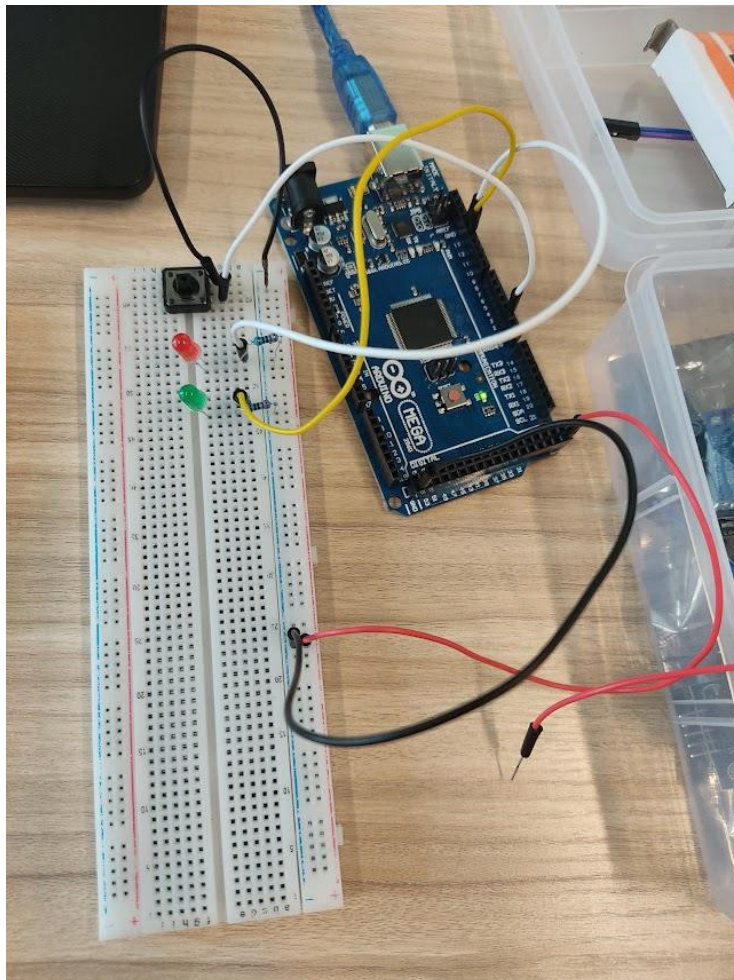
    // define Hardware Interrupt
    attachInterrupt(digitalPinToInterrupt(BUTTON), resetCount, RISING);
}
```

```
void loop()
{
    if (lastCount != count)
    {
        lastCount = count;
        Serial.println(count);
    }
}

void resetCount()
{
    count = 0;
}

void increaseCount()
{
    count++;
}
```

Circuit Image:



Exercise 2:**Code:**

```
/*
 * Exercise 2: Pseudo Multitasking without delay
 * Requirements:
 *   • 2 LEDs blink at the separate timing. A Red LED blinks
     every 100ms and a Green LED blinks every 300ms.
 *   • Each LED's turn-on time and turn-off time can be the
     same, but red LED and green led must blink separately
 * Constrains:
 *   • Do not use delay() command
 *   • Use millis() command to get current time.
 */

#define RED_BLINK_TIME_MS 100
#define GREEN_BLINK_TIME_MS 300

#define RED_LED_PIN 13
#define GREEN_LED_PIN 12

unsigned long lastRedTimeMS = 0;
unsigned long lastGreenTimeMS = 0;

void setup()
{
    pinMode(RED_LED_PIN, OUTPUT);
    pinMode(GREEN_LED_PIN, OUTPUT);
}

void loop()
{
    unsigned long currentTimeMS = millis();
    if (currentTimeMS - lastRedTimeMS >= RED_BLINK_TIME_MS)
    {
        lastRedTimeMS = currentTimeMS;
        digitalWrite(RED_LED_PIN, !digitalRead(RED_LED_PIN));
    }
    if (currentTimeMS - lastGreenTimeMS >= GREEN_BLINK_TIME_MS)
    {
        lastGreenTimeMS = currentTimeMS;
        digitalWrite(GREEN_LED_PIN, !digitalRead(GREEN_LED_PIN));
    }
}
```

Circuit Images:

