

Αναφορά 3^{ης} Εργαστηριακής Άσκησης

Στοιχεία Ομάδας:

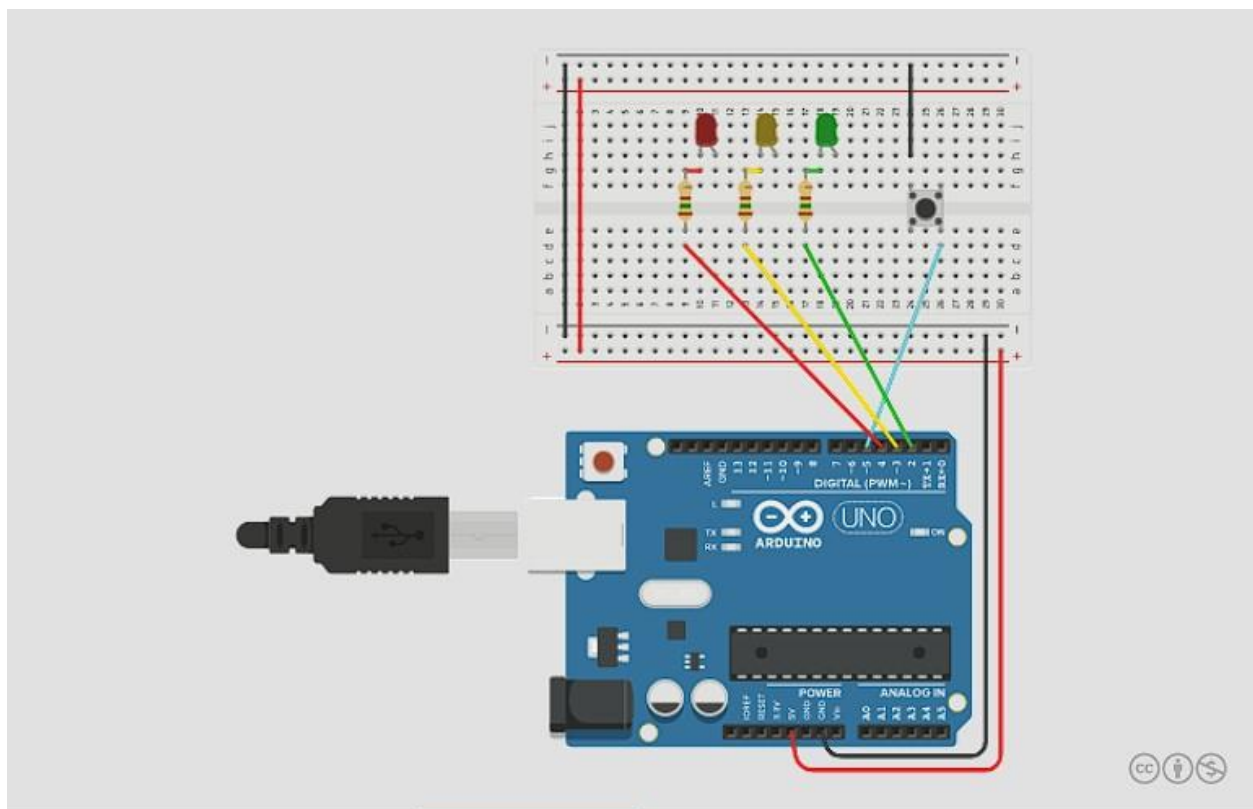
Δημήτριος Κωστορρίζος, ur1054419@upnet.gr

Λάμπρος Παπαδόπουλος, ur1054433@upnet.gr

Ομάδα Β

Τμήμα: 1

Tindercad Link: <https://www.tinkercad.com/things/51Y7Cqd3ZhF-copy-of-at91codeemuneask3template/editel?tenant=circuits>



Το κύκλωμα με την συγκεκριμένη άσκηση είναι υλοποιημένο όπως παραπάνω. Το κόκκινο led δείχνει τον αριθμό των λεπτών, το κίτρινο led δείχνει τον αριθμό των δευτερολέπτων και πράσινο led την

κατάσταση λειτουργίας του κυκλώματος. Η συνάρτηση FIQ Handler εκτελεί την διαδικασία μέτρησης των δευτερολέπτων. Στην συνάρτηση main, γίνεται η αρχικοποίηση των μεταβλητών.

Κώδικας Υλοποίησης σε Arduino

```
void FIQ_handler(void);
```

```
#define HIGH 0x1
```

```
#define LOW 0x0
```

```
#define BUT_IDLE 0
```

```
#define BUT_PRESSED 1
```

```
#define continue_count 0x01
```

```
#define stop_count 0x02
```

```
#define reset_and_start_count 0x05
```

```
#define idle 0
```

```
#define hold 1
```

```
unsigned int PIOA_int, TC_int;
```

```
unsigned long clockPulseCounter;
```

```
unsigned int Channel_0_CCR;
```

```
unsigned int previousButtonState;
```

```
unsigned long Channel_0_RC;
```

```
unsigned int buttonState;
```

```
unsigned int ledState;
```

```
unsigned int counter;
```

```
unsigned int decimalCounter;
```

```
unsigned int clockCounter;
```

```
void setup()
```

```
{
```

```
    // Green Led
```

```
    pinMode(2, OUTPUT);
```

```
    digitalWrite(2, LOW);
```

```
    // Yellow Led
```

```
    pinMode(3, OUTPUT);
```

```
    digitalWrite(3, LOW);
```

```
    // Red Led
```

```
    pinMode(4, OUTPUT);
```

```
    digitalWrite(4, LOW);
```

```
    // Button
```

```
    pinMode(5, INPUT_PULLUP);
```

```
    // Set the counters
```

```
    counter = 0;
```

```
    decimalCounter = 0;
```

```
    ledState = idle;
```

```
    clockCounter = 0;
```

```
    // Set the flags
```

```
    buttonState = BUT_IDLE;
```

```
    previousButtonState = BUT_PRESSED;
```

```
    clockPulseCounter = 0;
```

```

Channel_0_CCR = reset_and_start_count;
Channel_0_RC = 8192;

    // Set interrupt flag to 0
    PIOA_int = LOW;
    TC_int = LOW;

    // Set arduino serial input
    Serial.begin(9600);
}

void loop()
{
    // Reset and start counter
    if (Channel_0_CCR == reset_and_start_count)
    {
        clockPulseCounter = 0;
        Channel_0_CCR = continue_count;
    }

    // Continue to count the clock pulse
    if (Channel_0_CCR == continue_count)
    {
        clockPulseCounter++;
        if (clockPulseCounter == Channel_0_RC)
        {
            clockPulseCounter = 0;
            TC_int = HIGH;
        }
    }
}

```

```

}

// Read the button
buttonState = digitalRead(5);

// If the button is pressed
if (buttonState != previousButtonState)
{
    previousButtonState = buttonState;
    PIOA_int = HIGH;
}

// If an interrupt was flagged
if (PIOA_int | TC_int)
{
    // Call the fiq handler
    FIQ_handler();
}
}

void FIQ_handler()
{
    // If the interrupt was from the parallel input
    if (PIOA_int == HIGH)
    {
        PIOA_int = LOW;

        // If the button isn't pressed
        if(buttonState == BUT_IDLE)

```

```

    {
        buttonState = BUT_PRESSED;

        // If the led is idle
        if(ledState == idle)
        {
            // Turn on the led
            ledState = hold;
            Channel_0_RC = 4096;
        }
        else
        {
            // Turn off the led
            ledState = idle;
            Channel_0_RC = 8192;
        }
        Channel_0_CCR = 0x05;
    }

    else
    {
        // Turn off the led
        buttonState = BUT_IDLE;
    }
}

// If the interrupt was from the clock counter
if (TC_int == HIGH)
{

```

```
TC_int = LOW;
```

```
// If the led was idle
```

```
if(ledState == idle)
```

```
{
```

```
    // Set the leds
```

```
    digitalWrite(3, HIGH);
```

```
    digitalWrite(3, LOW);
```

```
    // Count the seconds
```

```
    counter++;
```

```
    if(counter > 59)
```

```
    {
```

```
        counter = 0;
```

```
    }
```

```
    // Print the seconds
```

```
    Serial.print("Time: ");
```

```
    Serial.println(counter);
```

```
    if(counter % 10 == 0)
```

```
    {
```

```
        decimalCounter = 0;
```

```
    }
```

```
    // If the decimal counter is lower than the seconds counter
```

```
    if(decimalCounter < (counter / 10))
```

```
    {
```

```
        digitalWrite(4, HIGH);
```

```
        digitalWrite(4, LOW);
```

```

        decimalCounter++;
    }
}
else
{
    digitalWrite(1, HIGH);
    digitalWrite(1, LOW);
}

// If the button isn't pressed
if(buttonState == BUT_IDLE)
{
    // Count the next clock second
    clockCounter++;

    // If the led is idle
    if(ledState == idle)
    {
        // Reset the clock counter
        if(clockCounter >= 1)
        {
            counter = 0;
            decimalCounter = 0;
            ledState = idle;
            clockCounter = 0;
            Channel_0_RC = 8192;
            Channel_0_CCR = 0x05;
        }
    }
}

```



```

else
{
    // Reset the clock counter is the led is lit
    if(clockCounter >= 2)
    {
        counter = 0;
        decimalCounter = 0;
        ledState = idle;
        clockCounter = 0;
        Channel_0_RC = 8192;
        Channel_0_CCR = 0x05;
    }
}
}
else
{
    // Reset the clock counter
    clockCounter = 0;
}
}
}

```

Κώδικας Υλοποίησης σε AT91

// Define the FIQ_handler

void FIQ_handler(void);

#define HIGH 0x1

#define LOW 0x0

#define BUT_IDLE 0

```
#define BUT_PRESSED 1

#define continue_count 0x01
#define stop_count 0x02
#define reset_and_start_count 0x05
#define idle 0
#define hold 1
```

```
unsigned int PIOA_int, TC_int;
unsigned long clockPulseCounter;
unsigned int Channel_0_CCR;
unsigned int previousButtonState;
unsigned long Channel_0_RC;
```

```
unsigned int buttonState;
unsigned int ledState;
unsigned int counter;
unsigned int decimalCounter;
unsigned int clockCounter;
```

```
void main()
{
    // Green Led
    pinMode(2, OUTPUT);
    digitalWrite(2, LOW);

    // Yellow Led
    pinMode(3, OUTPUT);
    digitalWrite(3, LOW);
```

```

// Red Led
pinMode(4, OUTPUT);
digitalWrite(4, LOW);

// Button
pinMode(5, INPUT_PULLUP);

// Set the counters
counter = 0;
decimalCounter = 0;
ledState = idle;
clockCounter = 0;

// Set the flags
buttonState = BUT_IDLE;
previousButtonState = BUT_PRESSED;
clockPulseCounter = 0;
Channel_0_CCR = reset_and_start_count;
Channel_0_RC = 8192;

// Set interrupt flag to 0
PIOA_int = LOW;
TC_int = LOW;

// Read until the 'e' character is inserted
while(getchar() != 'e')
{
    // Reset and start counter

```

```

if (Channel_0_CCR == reset_and_start_count)
{
    clockPulseCounter = 0;
    Channel_0_CCR = continue_count;
}

// Continue to count the clock pulse
if (Channel_0_CCR == continue_count)
{
    clockPulseCounter++;
    if (clockPulseCounter == Channel_0_RC)
    {
        clockPulseCounter = 0;
        TC_int = HIGH;
    }
}

// Read the button
buttonState = pioa->ISR;

// If the button is pressed
if (buttonState != previousButtonState)
{
    previousButtonState = buttonState;
    PIOA_int = HIGH;
}

// If an interrupt was flagged
if (PIOA_int | TC_int)

```

```

        {

            // Call the fiq handler
            FIQ_handler();

        }

    }

}

void FIQ_handler()
{

    // If the interrupt was from the parallel input
    if (PIOA_int == HIGH)
    {

        PIOA_int = LOW;

        // If the button isn't pressed
        if(buttonState == BUT_IDLE)
        {

            buttonState = BUT_PRESSED;

            // If the led is idle
            if(ledState == idle)
            {

                // Turn on the led
                ledState = hold;
                Channel_0_RC = 4096;

            }
            else
            {

                // Turn off the led

```

```

        ledState = idle;

        Channel_0_RC = 8192;

    }

    Channel_0_CCR = 0x05;

}

else
{

    // Turn off the led

    buttonState = BUT_IDLE;

}

}

// If the interrupt was from the clock counter
if (TC_int == HIGH)
{

    TC_int = LOW;

    // If the led was idle
    if(ledState == idle)
    {

        // Set the leds
        tc->Channel_0.CCR(0x04);

        // Count the seconds
        counter++;

        if(counter > 59)
        {

            counter = 0;

```

```

    }

    // Print the seconds
    printf("Time: %d.\n",counter);
    if(counter % 10 == 0)
    {
        decimalCounter = 0;
    }

    // If the decimal counter is lower than the seconds counter
    if(decimalCounter < (counter / 10))
    {
        tc->Channel_0.CCR(0x08);
        decimalCounter++;
    }
}
else
{
    tc->Channel_0.CCR(0x01);
}

// If the button isn't pressed
if(buttonState == BUT_IDLE)
{
    // Count the next clock second
    clockCounter++;

    // If the led is idle
    if(ledState == idle)

```

```

{
    // Reset the clock counter
    if(clockCounter >= 1)
    {
        counter = 0;
        decimalCounter = 0;
        ledState = idle;
        clockCounter = 0;
        Channel_0_RC = 8192;
        Channel_0_CCR = 0x05;
    }
}
else
{
    // Reset the clock counter is the led is lit
    if(clockCounter >= 2)
    {
        counter = 0;
        decimalCounter = 0;
        ledState = idle;
        clockCounter = 0;
        Channel_0_RC = 8192;
        Channel_0_CCR = 0x05;
    }
}
}
else
{
    // Reset the clock counter

```



```
clockCounter = 0;
```

```
}
```

```
}
```

```
}
```