



---

# ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ

---

ΟΜΑΔΑ 1  
2 ΕΓΡΑΣΙΑ/ΕΓΡΑΣΤΗΡΙΟ



20 ΜΑΙΟΥ 2022

Ανδρονίκου, Δημήτριος 9836 dimitriosa@ece.auth.gr  
Γιαννόπουλος, Νικόλαος 9629 ngiannop@ece.auth.gr

# Φιλοσοφία κώδικα

Η υλοποίηση ήταν απλή, με βάσει και τα παραδείγματα που παρουσιάστηκαν στο μάθημα.

Αρχικά φτιάχνουμε κάποιες συναρτήσεις για αρχικοποίηση του switch και σετάρισμα των led σε κάποιο χρώμα.

```
main.c
10 #define BUFF_SIZE 128
11
12 Queue rx_queue;
13
14 void uart_rx_isr(uint8_t rx) {
15     // Store the received character
16     queue_enqueue(&rx_queue, rx);
17 }
18
19 void switch_init(void) {
20     // #define P_SW_UP PA_1
21     gpio_set_mode(P_SW_UP, Input); // 1 switch
22 }
23
24 int switch_get(Pin pin) {
25     return !gpio_get(pin); // active low
26 }
27
28 void LED_RED_ON_ISR() {
29     leds_set(1, 0, 0);
30 }
31
32 void LED_GREEN_ON_ISR() {
33     leds_set(0, 1, 0);
34 }
35
36 void LED_BLUE_ON_ISR() {
37     leds_set(0, 0, 1);
38 }
39
40 void LEDS_OFF() {
41     leds_set(0, 0, 0);
42 }
43
```

Έπειτα στη main(), αφού αρχικοποιήσουμε κάποιες απαραίτητες μεταβλητές και ένα πίνακα για αποθήκευση των λαμβανόμενων χαρακτήρων από τη uart, αρχικοποιούμε τη uart, το διακόπτη, τα led και ενεργοποιούμε τη σειρά προτεραιότητας των interrupts. Θέτουμε τη προτεραιότητα της uart τελευταία, έτσι ώστε να είμαστε σίγουροι ότι το switch, θα έχει μεγαλύτερη προτεραιότητα.

```

45 int main() {
46     uint8_t rx_char = 0;
47     int EINT1_IRQn = 21;
48     int echo = 1, counter = 0;
49     char buff[BUFF_SIZE];
50     uint32_t buff_index;
51
52     queue_init(&rx_queue, 128);
53     uart_init(57600);           // Baud -> 57600
54     uart_set_rx_callback(uart_rx_isr); // set receiver ISR
55     uart_enable();             // enable to send char
56
57     switch_init();
58     leds_init();
59
60     __enable_irq();
61
62     NVIC_SetPriority(EINT1_IRQn, 192); // Set uart with lowest priority
63     uart_print("\r");
64

```

Στη συνέχεια ξεκινάει το κυρίως μέρος του προγράμματος στο infinity loop. Το πρώτο μέρος είναι παρόμοιο με τα παραδείγματα του μαθήματος, δηλαδή αναγνωρίζουμε αν λαμβάνουμε backspace και εκτυπώνουμε τους χαρακτήρες που πληκτρολογούνται.

```

64
65 while(1){
66     uart_print("Enter your second name: ");
67     buff_index = 0;
68     do{
69         while(!queue_dequeue(&rx_queue, &rx_char))
70             __WFI();
71         if(rx_char == 0x7F){ // Handle backspace character
72             if(buff_index > 0){
73                 buff_index--;
74                 if (echo) {
75                     uart_tx(rx_char);
76                 }
77             }
78         } else if ((rx_char >= 'a' && rx_char <= 'z') || (rx_char >= 'A' && rx_char <= 'Z') || rx_char == ' ' || rx_char == '\r'){
79             // Store character
80             buff[buff_index++] = (char) rx_char;
81             if (echo) {
82                 uart_tx(rx_char);
83             }
84         }
85     } while ( rx_char != '\r' && buff_index < BUFF_SIZE );
86     buff[buff_index - 1] = '\0';
87     if (echo) {
88         uart_print("\r\n");
89     }
90

```

Στη συνέχεια, ελέγχουμε αν λαμβάνουμε κάποιους συγκεκριμένους χαρακτήρες. Έπειτα ελέγχουμε αν το τελευταίο στοιχείο του buff (που αποθηκεύει τους χαρακτήρες), είναι φωνήεν ή σύμφωνο και πράττουμε ανάλογα, εκτυπώνοντας μήνυμα στην uart και ανάβοντας τα καταλληλά led.

```
main.c
92     if (buff_index < BUFF_SIZE)
93     {
94         if(buff[buff_index - 1] == 'e' || buff[buff_index - 1] == 'y' ||
95            buff[buff_index - 1] == 'u' || buff[buff_index - 1] == 'i' ||
96            buff[buff_index - 1] == 'o' || buff[buff_index - 1] == 'a'
97            || buff[buff_index - 1] == 'E' || buff[buff_index - 1] == 'Y' ||
98            buff[buff_index - 1] == 'U' || buff[buff_index - 1] == 'I' ||
99            buff[buff_index - 1] == 'O' || buff[buff_index - 1] == 'A' )
100        {
101
102            uart_print("Red Led ON!");
103            LED_RED_ON_ISR();
104        } else if( buff[buff_index - 1] == 'q' || buff[buff_index - 1] == 'w' ||
105            buff[buff_index - 1] == 'r' || buff[buff_index - 1] == 't' ||
106            buff[buff_index - 1] == 'p' || buff[buff_index - 1] == 's' ||
107            buff[buff_index - 1] == 'd' || buff[buff_index - 1] == 'f' ||
108            buff[buff_index - 1] == 'g' || buff[buff_index - 1] == 'h' ||
109            buff[buff_index - 1] == 'j' || buff[buff_index - 1] == 'k' ||
110            buff[buff_index - 1] == 'l' || buff[buff_index - 1] == 'z' ||
111            buff[buff_index - 1] == 'x' || buff[buff_index - 1] == 'c' ||
112            buff[buff_index - 1] == 'v' || buff[buff_index - 1] == 'b' ||
113            buff[buff_index - 1] == 'n' || buff[buff_index - 1] == 'm' ||
114            || buff[buff_index - 1] == 'Q' || buff[buff_index - 1] == 'W' ||
115            buff[buff_index - 1] == 'R' || buff[buff_index - 1] == 'T' ||
116            buff[buff_index - 1] == 'P' || buff[buff_index - 1] == 'S' ||
117            buff[buff_index - 1] == 'D' || buff[buff_index - 1] == 'F' ||
118            buff[buff_index - 1] == 'G' || buff[buff_index - 1] == 'H' ||
119            buff[buff_index - 1] == 'J' || buff[buff_index - 1] == 'K' ||
120            buff[buff_index - 1] == 'L' || buff[buff_index - 1] == 'Z' ||
121            buff[buff_index - 1] == 'X' || buff[buff_index - 1] == 'C' ||
122            buff[buff_index - 1] == 'V' || buff[buff_index - 1] == 'B' ||
123            buff[buff_index - 1] == 'N' || buff[buff_index - 1] == 'M')
124        {
125
```

```
125
126            uart_print("Green Led ON!");
127            LED_GREEN_ON_ISR();
128        } else if(switch_get(P_SW_UP) == 1)
129        {
130            if (counter < 1){
131                uart_print("BLUE Led ON!");
132                LED_BLUE_ON_ISR();
133                counter++;
134            } else{
135                uart_print("BLUE Led OFF!");
136                LEDS_OFF();
137                counter = 0;
138            }
139        }
140    }
141}
142}
143}
144
```

Τέλος, ελέγχουμε αν πατήθηκε ο διακόπτης καθώς και μετρούμε πόσες φορές έχει πατηθεί και πράττουμε ανάλογα, όπως και παραπάνω.

# Προβλήματα

Το κυρίως πρόβλημα ήταν, η έλλειψη γνώσεων γύρω από uart, led και switch, λόγω των περιορισμένων παραδειγμάτων.

## Testing

Τα τεστ έγιναν στο εργαστήριο πάνω στην πλακέτα καθώς δοκιμάσαμε για όλα τα φωνήεν και σύμφωνα με την σειρά για να βεβαιωθούμε ότι λειτουργεί σωστά. Στην συνέχεια δοκιμάσαμε το κουμπί άμα δουλεύει και εκτυπώνει στο arduino το ανάλογο μήνυμα και αλλάζοντας το χρώμα του led διαφορετικά να το σβήναμε.