

**RAČUNARSKE ARHITEKTURE**  
**FIRST PRESS BAZZER GAME**



Student: **Amina Dacić**

Profesor: **prof.dr.Elmedin Selmanović**

Sarajevo, jun 2020

- Uvod i opis projekta

U kviz emisijama, posebno zabavnim aktivnostima (npr. takmičarskim aktivnostima odgovaranja) organizatori često primjenjuju buzzer sistem u cilju tačnog, pravičnog i vizuelnog određivanja broja mjesta odgovornika.

Sistem može da ilustruje tačnost procjene podataka, što ujedno i poboljšava zabavu. U isto vrijeme, to je poštenije i pravednije.

Dakle, kao rezultat ove igre dobijamo led rezultate ili rezultate koji će se ocrtavati na microbit-u i time dati i vizuelnu predstavu koji od učesnika je prvi, ko drugi, ko treći a ko četvrti. Za uspešnu realizaciju ovog projekta dozvoljena je igra za najviše četiri igrača zbog nedostatka pin-ova na microbit-u, pri čemu je peti taster rezervisan za ponovno pokretanje igre odnosno restartovanje sistema.

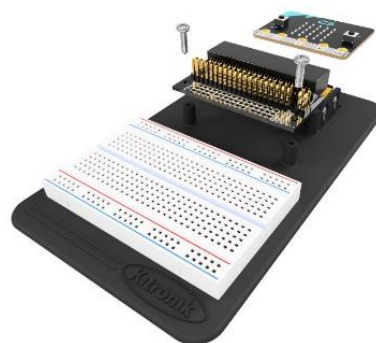
Varanje od strane bilo kojeg učesnika u ovom sistemu nije moguć jer ukoliko neki od takmičara ipak pokuša igrati prije dozvoljenog vremena, pojaviće se na ekranu microbit-a mali simbol “X” u njegovu stranu odnosno u stranu njegovog tastera.

Takođe se i prati vrijeme, izraženo u milisekundama, od početka takmičenja u jednom krugu do prvog pritiska jednog od tastera (odnosno učesnik koji je u tom trenutku pobijedio i ima pravo prvi odgovoriti) koji su u igri u cilju praćenja brzine reakcije odgovarajućeg učesnika.

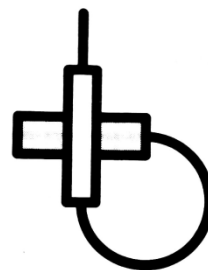
- Izrada projekta

Za implementaciju ovog projekta potrebne su sljedeće komponente:

- 1x BBC Micro:bit ploča

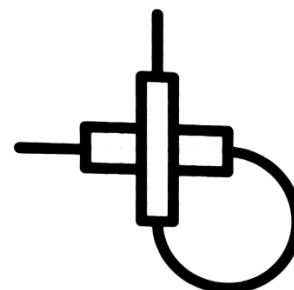


- 4x M/F poveznice



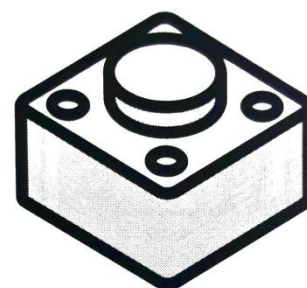
**MALE TO FEMALE  
JUMPER WIRES**

- 3x M/M poveznice



**MALE TO MALE  
JUMPER WIRES**

- 3x tastera

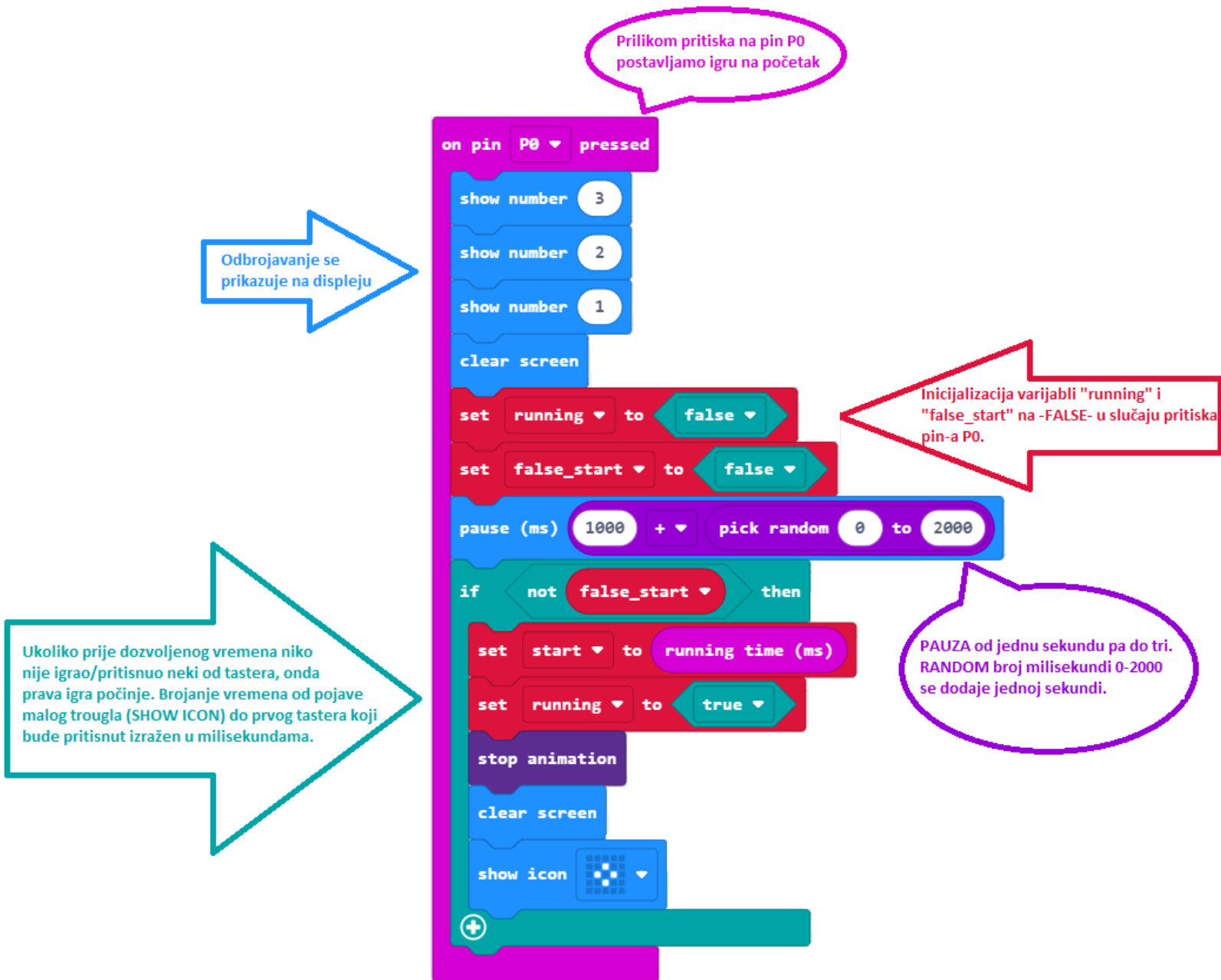


**PUSH SWITCH**

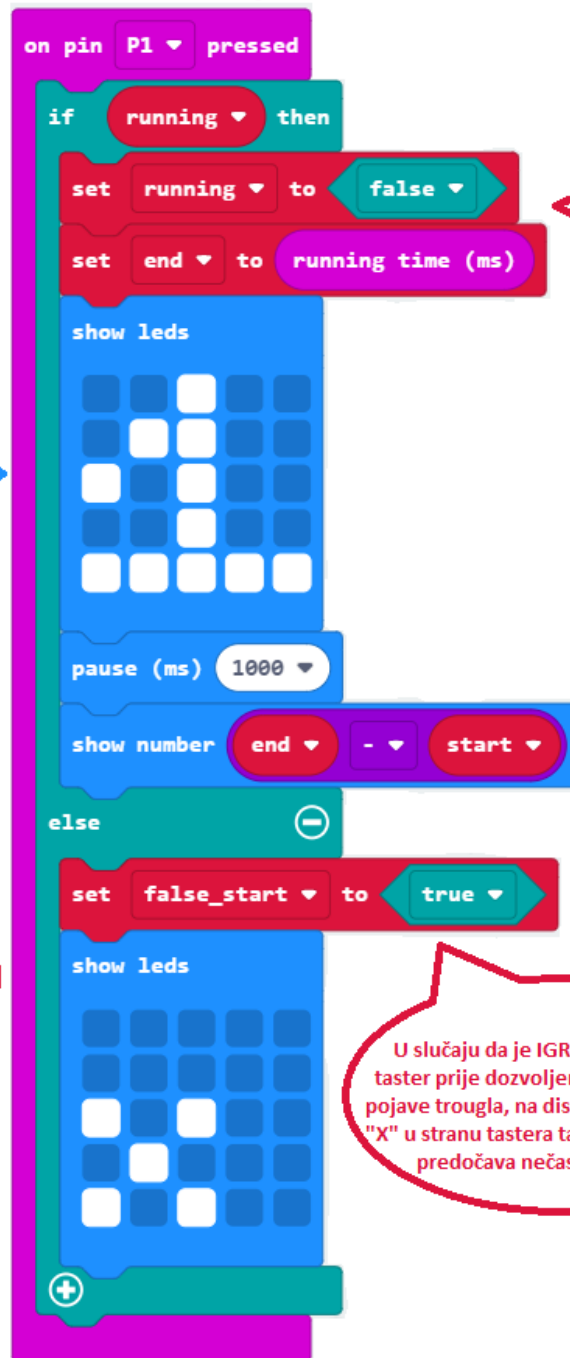
Takođe koristim i dva tastera A i B koji su ugrađeni u microbit. I naravno potreban je USB za komunikaciju microbit-a sa računarom tako da se kod, rađen u

programskom jeziku Python, može download-ovati kao HEX file u disk MICROBIT. Sada je sve spremno za pokretanje ovog sistema.

- Kod (Blocks)



Ovaj taster upotrijebit će IGRAČ 1 (pin 1)



U slučaju da IGRAČ 1 pobijedi odnosno prvi pritisne svoj taster u odnosu na druge igrače, vrijeme se zaustavlja i upisuje se u varijablu "end" (u ms).

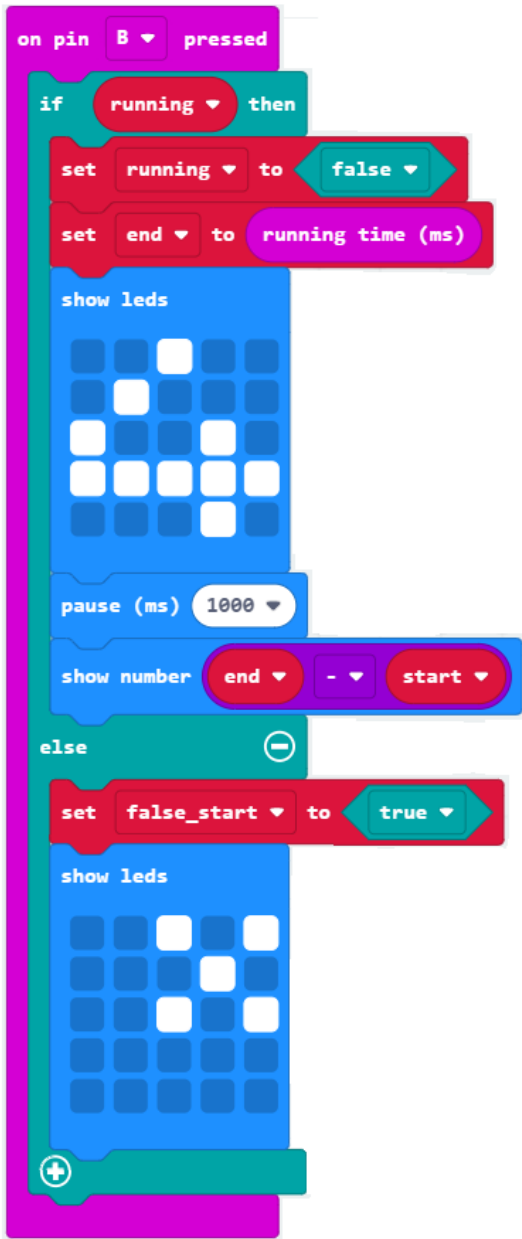
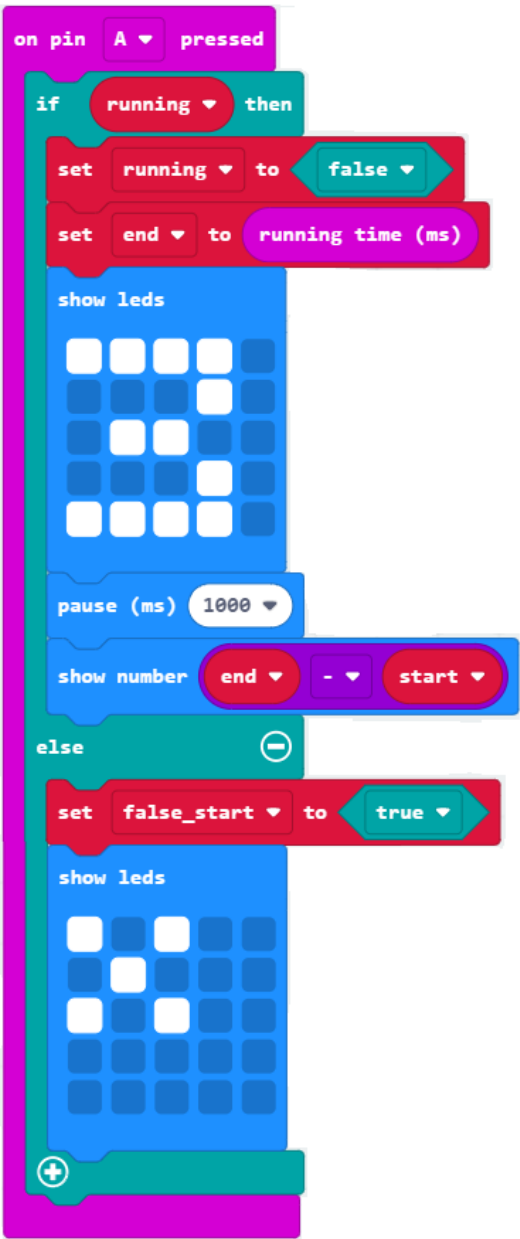
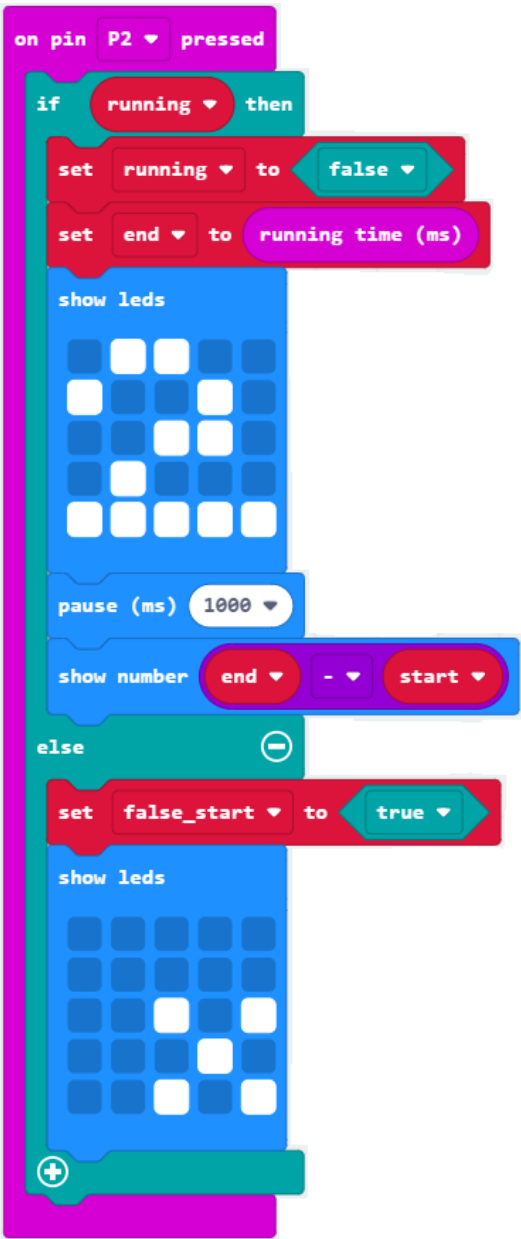
Ispis pobjedničkog broja na displeju

Pauza između prikaza broja "1" i vremena RAZLIKE početka "start" i završnog trenutka

Takođe se ovaj mali simbol "X" pojavljuje i nakon pobjednikovog broja očitano na displeju u pravom redosljedu. To znači da će se poslije pobjed. broja očitati svi ostali X-evi svih igrača u igri onako kako je koji pritisnut. (svaki X u svoju stranu na displeju)

U slučaju da je IGRAČ 1 pritisnuo svoj taster prije dozvoljenog vremena ili prije pojave trougla, na displeju se očitava simbol "X" u stranu tastera tako da vizuelno predočava nečasnog igrača :)

Slično objašnjenje i za sljedeće tastere: IGRAČ 2 (pin 2), IGRAČ 3 (pin A) i IGRAČ 4 (pin B).



- Kod (Python)

```
running = False
false_start = False
start = 0
end = 0
```

```
def on_pin_pressed_p0():
    global running, false_start, start
    basic.show_number(3)
    basic.show_number(2)
    basic.show_number(1)
    basic.clear_screen()
    running = False
    false_start = False
    basic.pause(1000 + randint(0, 2000))
    if not (false_start):
        start = input.running_time()
        running = True
        led.stop_animation()
        basic.clear_screen()
        basic.show_icon(IconNames.SMALL_DIAMOND)
input.on_pin_pressed(TouchPin.P0, on_pin_pressed_p0)
```

```
def on_button_pressed_a():
    global running, end, false_start
    if running:
        running = False
        end = input.running_time()
        basic.show_leds("""
            ###.
            ...#.
            .##..
            ...#.
            ###.
            """)
        basic.pause(1000)
        basic.show_number(end - start)
    else:
        false_start = True
        basic.show_leds("""
            #.##..
            .#...
            #.##..
            .....
            """)
```

```

        .....
        """)
input.on_button_pressed(Button.A, on_button_pressed_a)

```

```

def on_pin_pressed_p2():
    global running, end, false_start
    if running:
        running = False
        end = input.running_time()
        basic.show_leds("""
            . # # . .
            # . . # .
            . . # # .
            . # . . .
            # # # # #
            """)
        basic.pause(1000)
        basic.show_number(end - start)
    else:
        false_start = True
        basic.show_leds("""

```

```

            .....
            .....
            . . # . #
            . . . # .
            . . # . #
            """)

```

```

input.on_pin_pressed(TouchPin.P2, on_pin_pressed_p2)

```

```

def on_button_pressed_b():
    global running, end, false_start
    if running:
        running = False
        end = input.running_time()
        basic.show_leds("""
            . . # . .
            . # . . .
            # . . # .
            # # # # #
            . . . # .
            """)
        basic.pause(1000)
        basic.show_number(end - start)
    else:
        false_start = True
        basic.show_leds("""
            . . # . #

```



```

        ...# .
        ..# .#
        .....
        .....
        """)
input.on_button_pressed(Button.B, on_button_pressed_b)

def on_pin_pressed_p1():
    global running, end, false_start
    if running:
        running = False
        end = input.running_time()
        basic.show_leds("""
            ..#..
            .##..
            #.#..
            ..#..
            ####
            """)
        basic.pause(1000)
        basic.show_number(end - start)
    else:
        false_start = True
        basic.show_leds("""
            .....
            .....
            #.#..
            .#...
            #.#..
            """)
input.on_pin_pressed(TouchPin.P1, on_pin_pressed_p1)

```