Reaction time Tester

(RTT)

15.06.2022

smart Robotics

Table of Contents

[Terms of reference 3](#_Toc105957610)

[Component list 3](#_Toc105957611)

[Circuit 4](#_Toc105957612)

[Structure 5](#_Toc105957613)

[Source code with comments 6](#_Toc105957614)

# Terms of reference

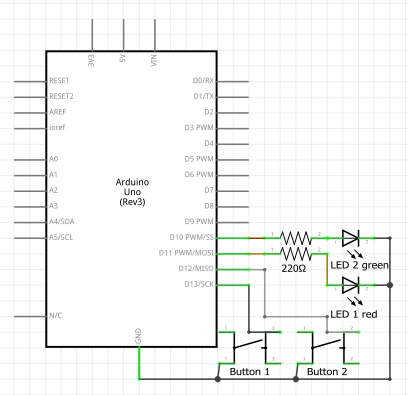
The task is to build and program a reaction time tester (RTT). The tester must consist out of two buttons and two LEDs. It should work like described below:

1. One LED is on and on the monitor, you can see that the RTT is ready, and you must press the first button.
2. When the first button gets pressed the LED starts blinking fast and, on the monitor, you see that you should wait for the red light.
3. As soon as the red LED lights up and the green LED goes out, the time is counted until the second button is pressed.
4. After the second button is pressed the red light goes out and the green one lights up again. Now you can read your reaction time on the screen.

# Component list

|  |  |  |  |
| --- | --- | --- | --- |
| Quantity | Part | Symbol | Picture |
| 2 | 220Ω Resistor |  | E-Projects 100EP512220R 220 Ohm Resistors, 1/2 W, 5% (Pack of 100):  Amazon.com: Industrial & Scientific |
| 1 | Arduino Uno | Ein Bild, das Text, Antenne enthält.  Automatisch generierte Beschreibung | Arduino UNO Rev3 [A000066] : Amazon.de: Computer & Zubehör |
| 2 | Button | Schalter — Grundwissen Elektronik | Pts645 Series 6 Mm Tact Switches |
| 1 | green LED 5mm |  | Led verte diffusante 5mm |
| 1 | red LED 5mm |  | L-7113SURCK | Kingbright LED 645nm Rot 5 mm T-1 3/4 | Distrelec Deutschland |

# Circuit



# Ein Bild, das Text, Schaltkreis, Elektronik enthält. Automatisch generierte BeschreibungStructure

# Source code with comments

#include <Bounce2.h> // including library Bounce2

#define btn1 11 // button 1

#define btn2 10 // button 2

class Led { // class Led

  int pin; // pin

  bool state; // state

  public:

  Led(int ledPin){ // constructor

    pin = ledPin; // pin

    pinMode(pin, OUTPUT); // pin mode

  }

  void toggle(){ // toggle led

    state = !state; // state

    digitalWrite(pin, state); // set state for pin

  }

  void speedToggle(int ms, int maxC){ // speed toggle

    int c; // counter

    while (c < maxC) // while counter is less than max counter

    {

      toggle(); // toggle led

      delay(ms); // delay

      c = c + random(ms); // counter

    }

  }

};

int gState = 1; // global state

long start; // global start time

long stop; // global stop time

double result; // global result

Bounce btn1Bouncer = Bounce(); // creating Bounce object for button 1

Bounce btn2Bouncer = Bounce(); // creating Bounce object for button 2

void setup(){ // setup function

  Serial.begin(9600); // starting serial communication

  btn1Bouncer.attach(btn1, INPUT\_PULLUP); // attaching button 1 to Bounce object

  btn2Bouncer.attach(btn2, INPUT\_PULLUP); // attaching button 2 to Bounce object

  btn2Bouncer.interval(25); // setting interval for button 2

  btn1Bouncer.interval(25); // setting interval for button 1

}

Led led1(12); // creating Led object for led 1

Led led2(13); // creating Led object for led 2

void loop(){ // loop function

  btn1Bouncer.update(); // updating Bounce object for button 1

  btn2Bouncer.update(); // updating Bounce object for button 2

  switch(gState){ // switch statement for global state

    case 1: // case 1

      Serial.println("------------------------------------------"); // printing message

      Serial.println("Reaction Tester ready!! Press button 1."); // printing message

      led1.toggle(); // toggle led 1

      gState = 2; // global state

      break; // break

    case 2: // case 2

      if (btn1Bouncer.fell()){ // if button 1 fell

        gState = 3; // global state

      }

      break; // break

    case 3:

      Serial.println("Waiting for the red light..."); // printing message

      gState = 4; // global state

      break; // break

    case 4: // case 4

      led1.speedToggle(300, random(5\*1000)); // speed toggle led 1

      led2.toggle(); // toggle led 2

      led1.toggle(); // toggle led 1

      start = millis(); // start time

      gState = 5; // global state

      break; // break

    case 5: // case 5

      if (btn2Bouncer.fell()){ // if button 2 fell

        stop = millis(); // stop time

        led2.toggle(); // toggle led 2

        led1.toggle(); // toggle led 1

        result = stop - start; // result

        Serial.print("Reaction time: "); // printing message

        Serial.print(result / 1000); // printing result

        Serial.println(" seconds"); // printing message

        Serial.println("------------------------------------------");

// printing message

        led1.toggle(); // toggle led 1

      }

      break; // break

    default: // default case

      Serial.println("Debug: Error!"); // printing message

      break; // break

  }

}