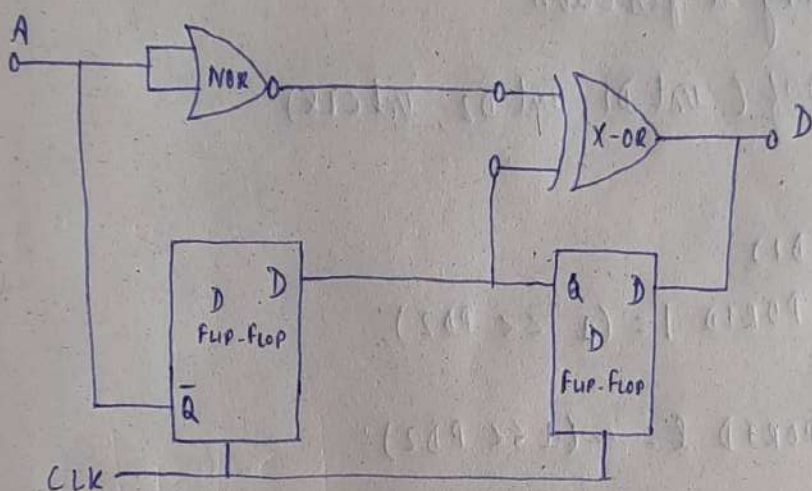


14-11-2023

AVR - GCC

GATE EE 2023COMPLETED

Q 56. Neglecting the delays due to the logic gates in the circuit shown in figure, the decimal equivalent of the binary sequence [ABCD] of initial logic states, which will not change with clock, is _____



SOLUTION:-

from circuit

$$D_1 = B \oplus C = A_2 \oplus A_1$$

$$D_2 = A_1$$

$$\text{Let, } A_2 A_1 = 00$$

$$\text{CLK} \quad D_2 \quad D_1 \quad A_2 \quad A_1$$

$$\begin{array}{ccccc} \text{CLK} & D_2 & D_1 & A_2 & A_1 \\ \text{0} & 0 & 0 & 0 & 0 \end{array} \left. \vphantom{\begin{array}{ccccc} \text{CLK} & D_2 & D_1 & A_2 & A_1 \\ \text{0} & 0 & 0 & 0 & 0 \end{array}} \right\} \text{No Change}$$

Sequence \Rightarrow A B C D

$$\bar{A}_2 \quad A_2 \quad A_1 \quad (A_1 \oplus A_2)$$

$$1 \quad 0 \quad 0 \quad 0$$

\therefore The decimal equivalent of 1000 is $(8)_{10} = 8$

CODE :-

```
-> #include <avr/io.h>
-> #include <util/delay.h>
-> // Declaring all variables as integers
-> int D1, D2, CLK;
-> int A1 = 0, A2 = 0;
-> // Creating a function
-> void ref ( int D1, int D2, int CLK)
-> {
->     if (D1)
        PORTD |= (1 << PD2);
    else
        PORTD &= ~(1 << PD2);
->     if (D2)
        PORTD |= (1 << PD3);
    else
        PORTD &= ~(1 << PD3);
->     if (CLK)
        PORTB |= (1 << PB3);
    else
        PORTB &= ~(1 << PB3);
-> }
-> int main (void) {
->     // Setting DDR registers for required pins
```

```

-> DDRD |= (1 << PD2) | (1 << PD3);
-> DDRB |= (1 << PB3);
-> // setting input pins
-> DDRB &= ~( (1 << PD6) | (1 << PD7) );
-> // the loop function runs over & over again
-> while (1) {
->   PORTB |= (1 << PB3);
->   - delay_ms (1000);
->   A1 = (PIND >> PD6) & 0x01;
->   A2 = (PIND >> PD7) & 0x01;
->   D2 = A1; // D2 = A1
->   D1 = A2 || A1; // D1 = A2 + A1
->   PORTB &= ~(1 << PB3);
->   ref (D1, D2, CLK);
-> }
-> }

```

COMPONENTS :-

- * Arduino UNO
- * 7-segment Display
- * 7447 IC

14-11-2023

AVR - GCC

PATH:-

14 Nov 12:35

/sdcard / digital-design / avr-gcc / setup / codes

ls

* Makefile

* avr.elf

* main.c

* main.hex

* avr.c

* avr.hex

* main.elf

* main.o

* avr.elf

* avr.o

* main.elf

To open file code = `mvim avr.c`

To run file code = `make`

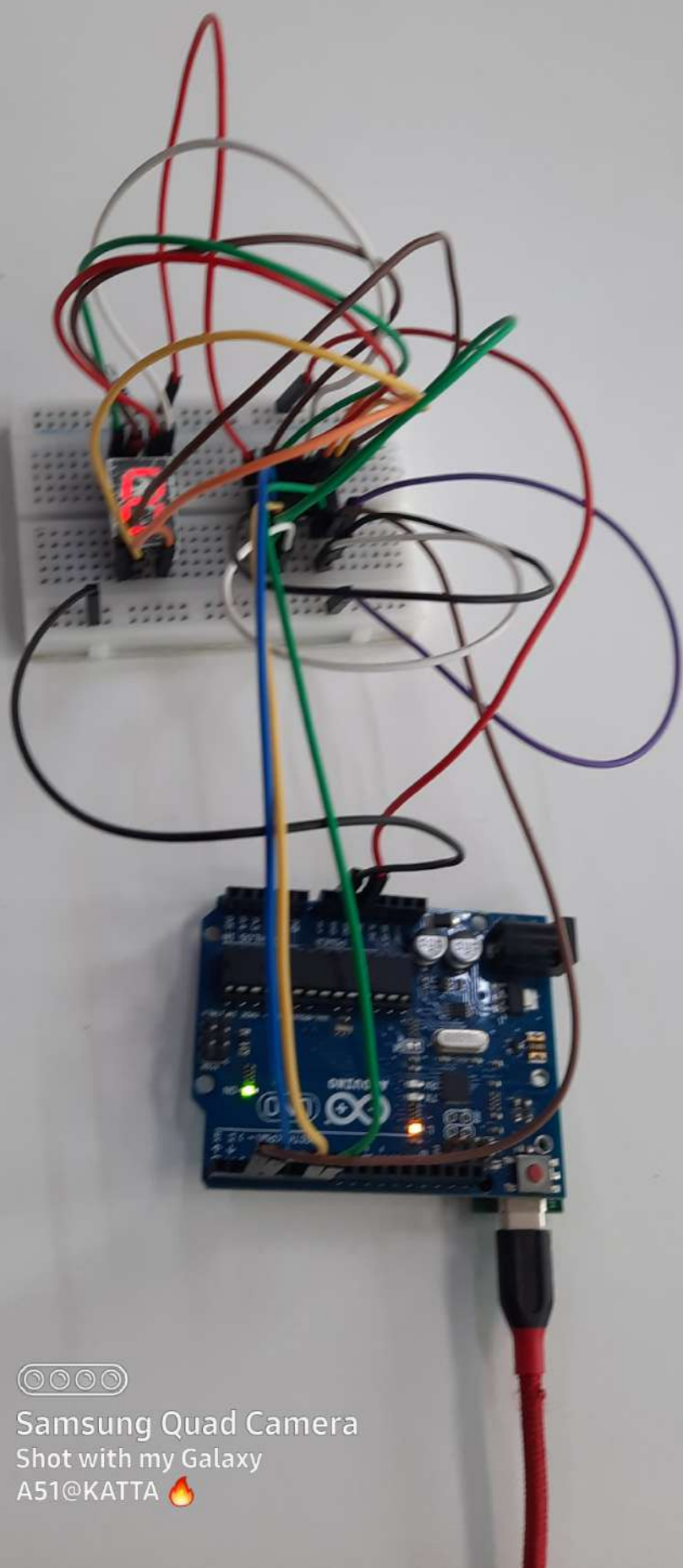
avr.hex

SKETCH UPLOAD IN ARDUINO DROID =>

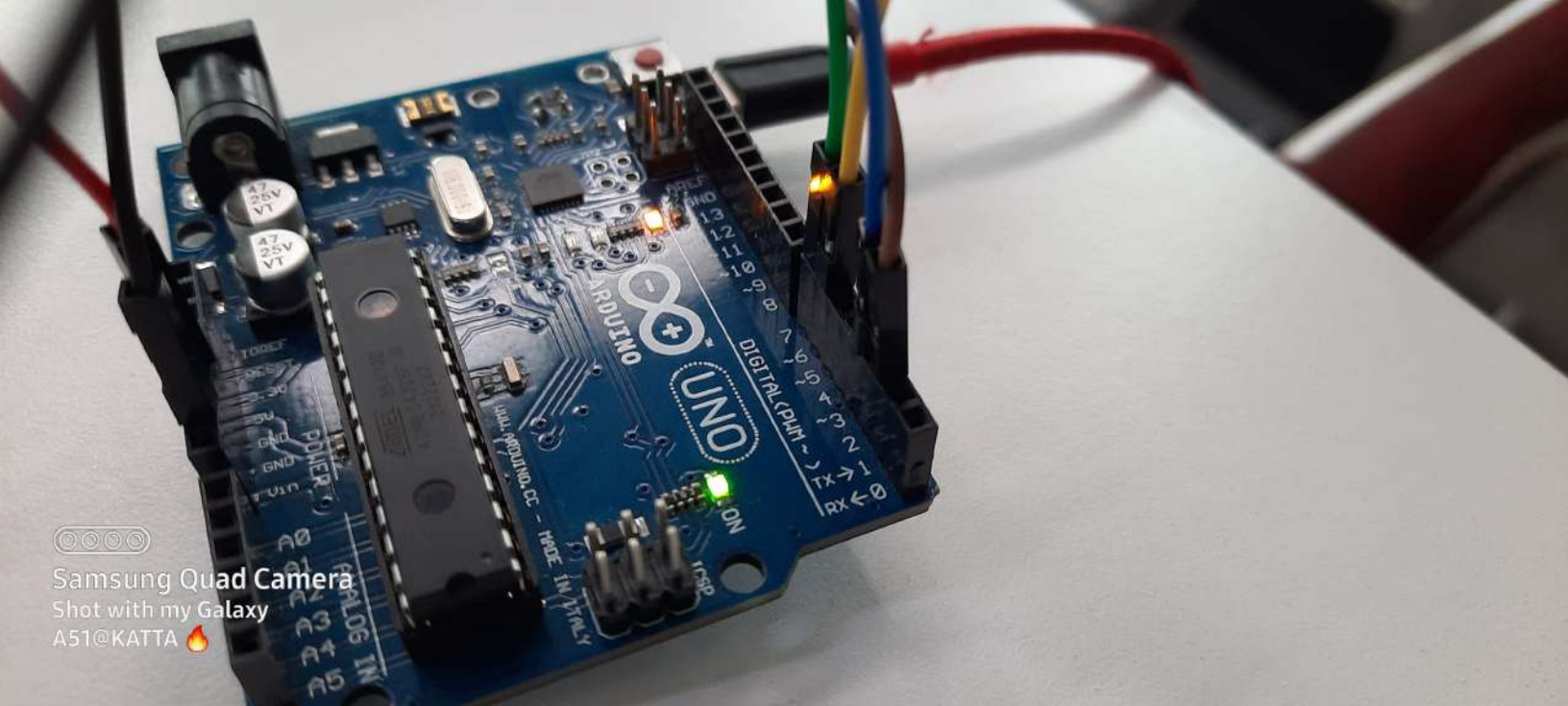
Actions -> Upload -> Upload precompiled ->

digital - design -> avr-gcc -> setup -> codes ->

* avr.hex * main.hex -> select * avr.hex



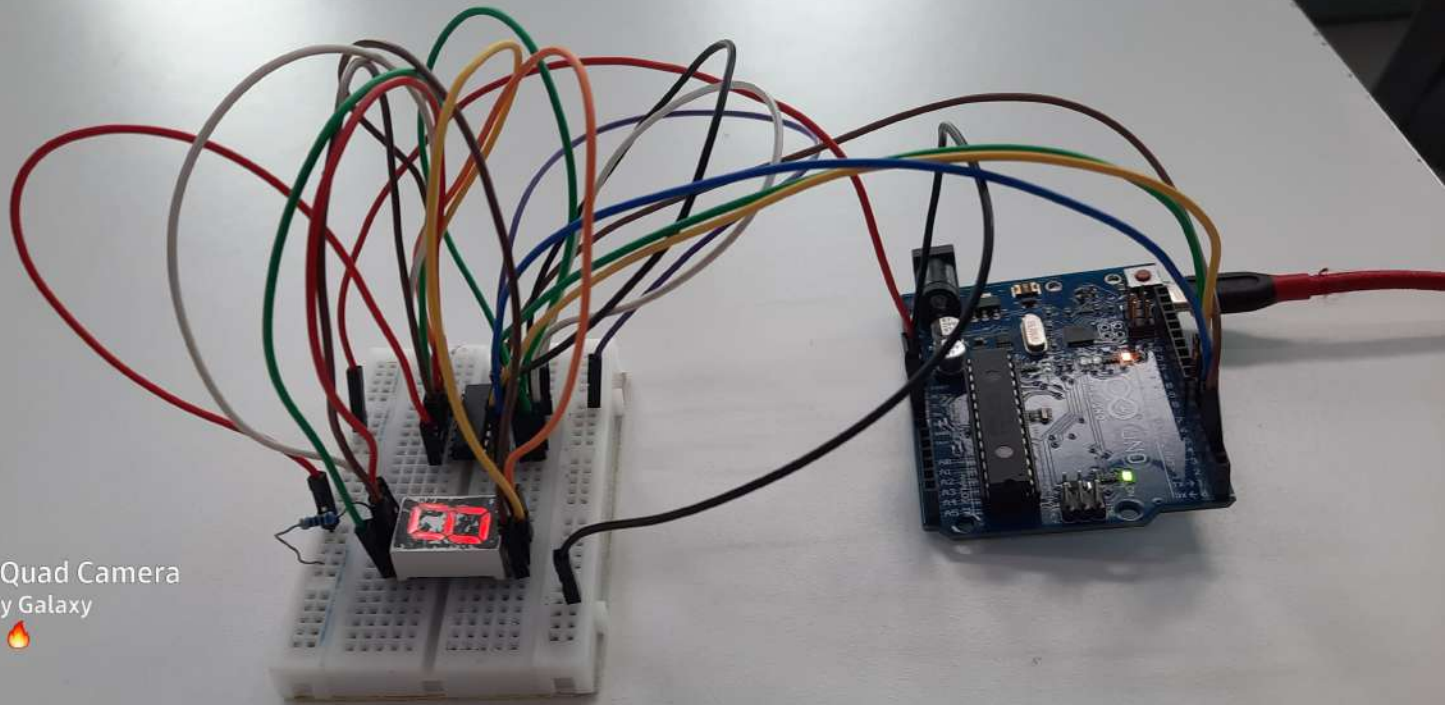
Samsung Quad Camera
Shot with my Galaxy
A51@KATTA 🔥



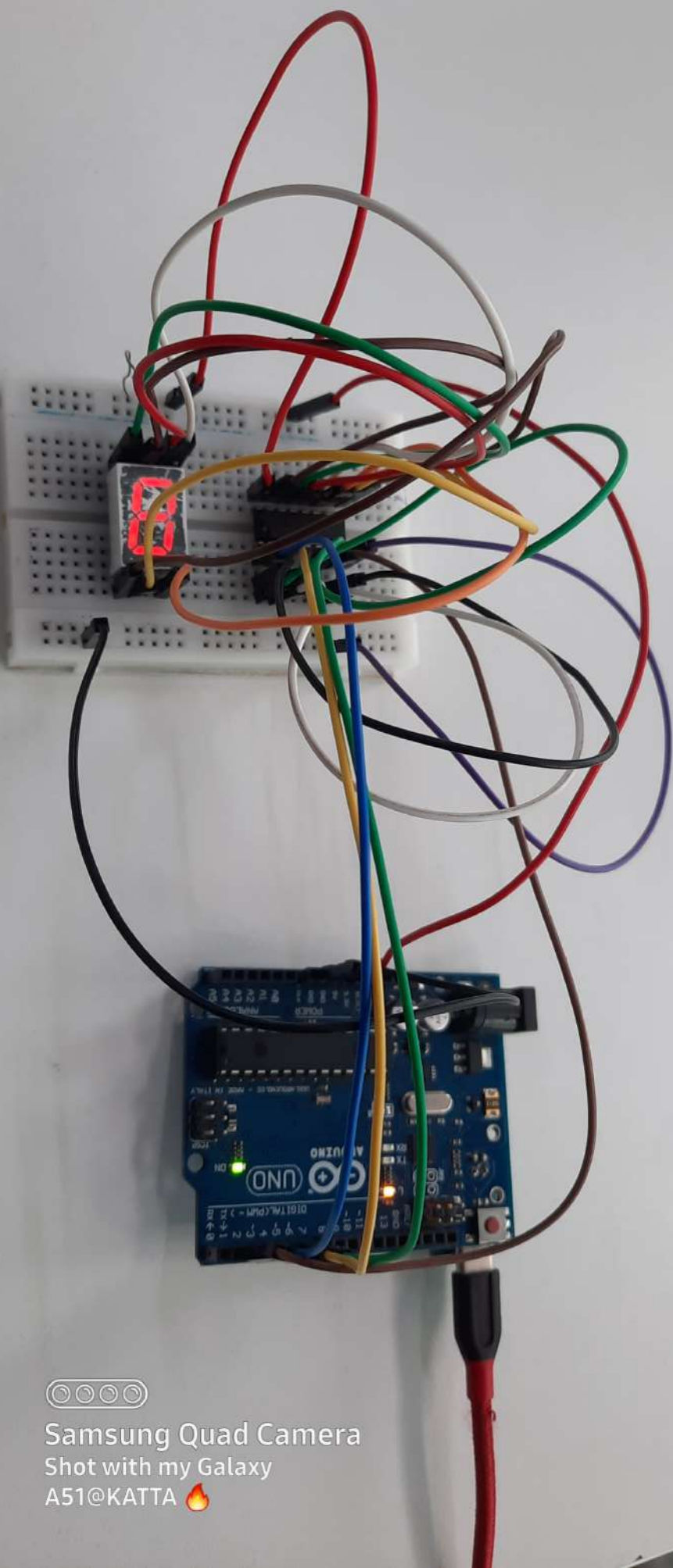
Samsung Quad Camera

Shot with my Galaxy

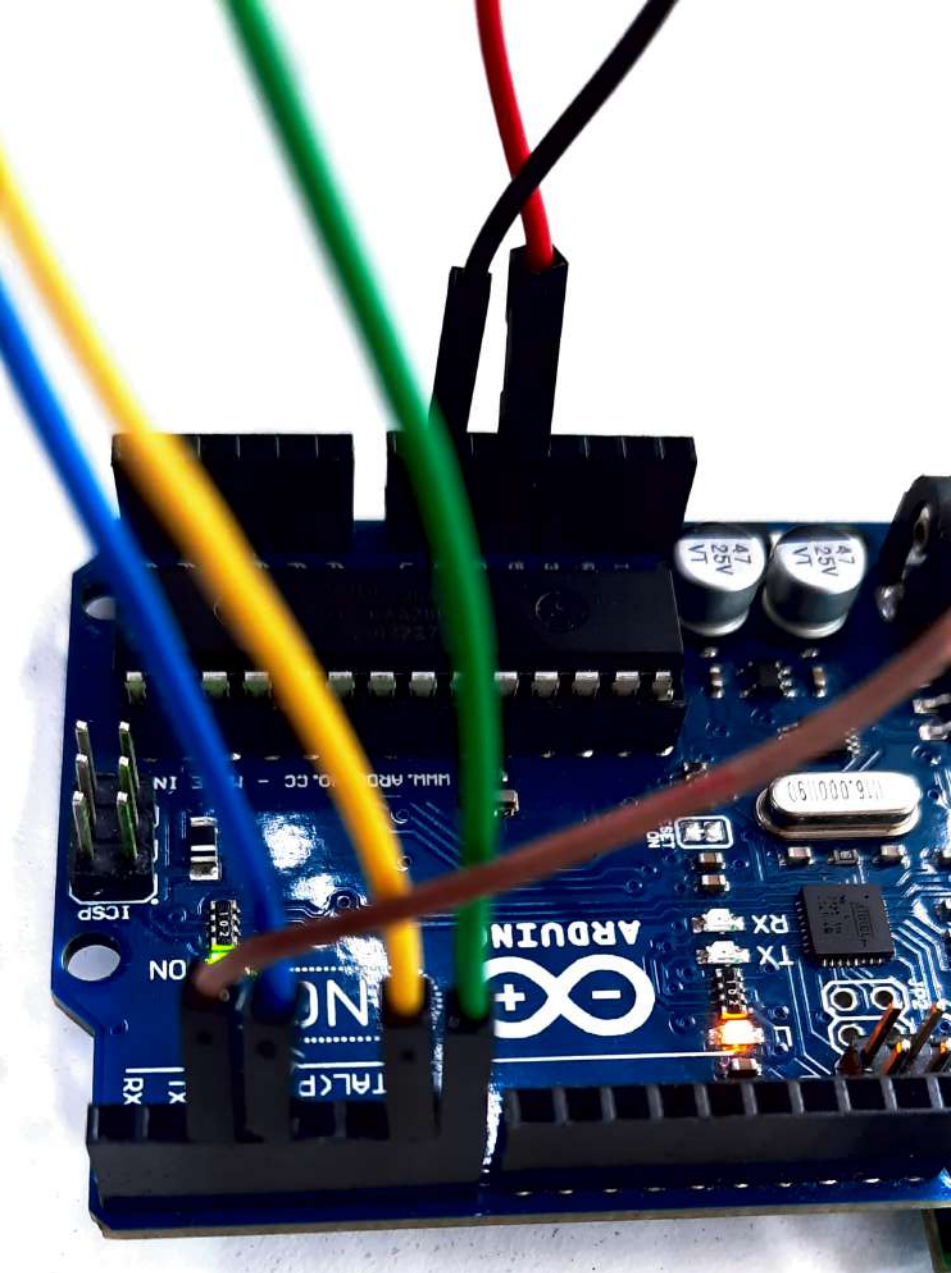
A51@KATTA 🔥



Samsung Quad Camera
Shot with my Galaxy
A51@KATTA 🔥



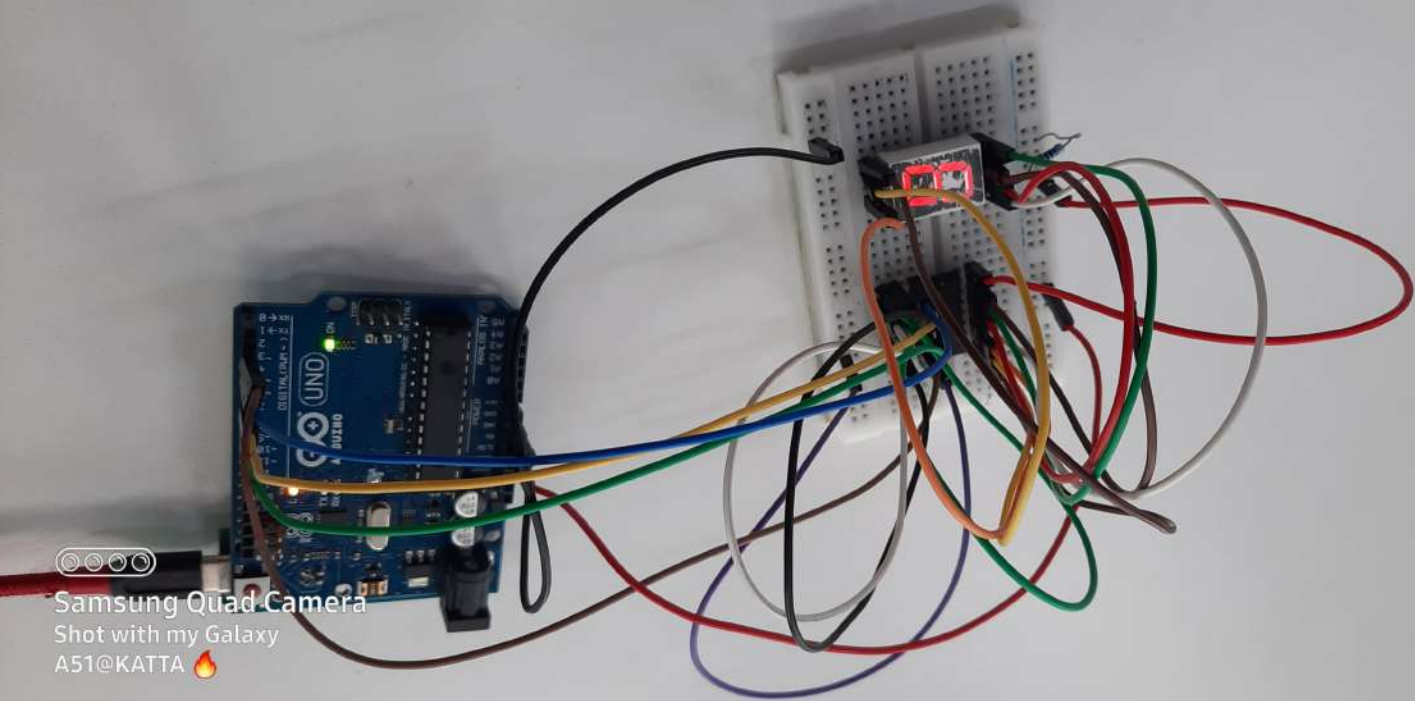
Samsung Quad Camera
Shot with my Galaxy
A51@KATTA 🔥



Samsung Quad Camera
Shot with my Galaxy
A51@KATTA 🔥



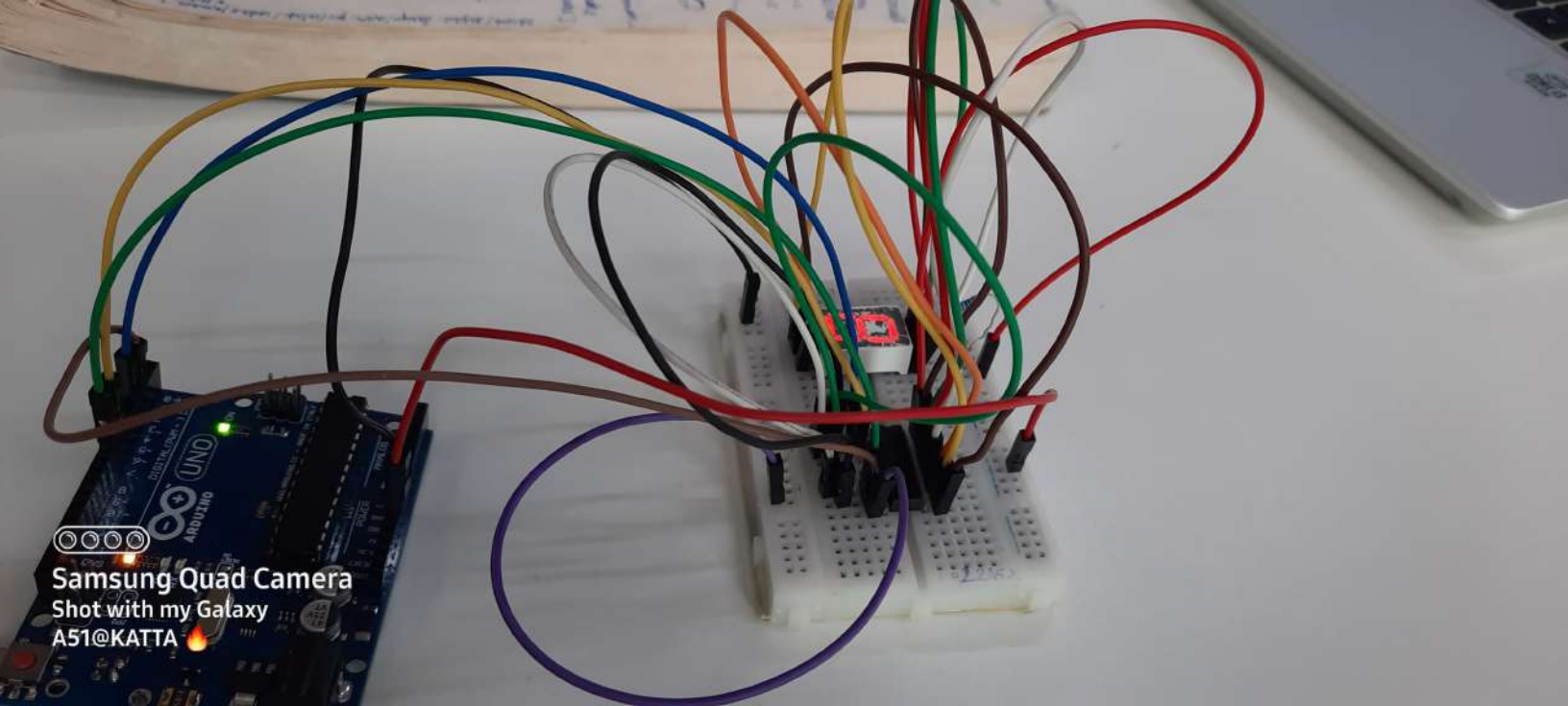
Samsung Quad Camera
Shot with my Galaxy
A51@KATTA 🔥



Samsung Quad Camera

Shot with my Galaxy

A51@KATTA 🔥



Samsung Quad Camera
Shot with my Galaxy
A51@KATTA