

Protocol Laboratory Digital Engineering #5

Hochschule Rhein-Waal

Rhine-Waal University of Applied Sciences

Group members: Quang Ha, Pham / 22338

Quang Minh, Dinh / 24211

Edwin, Changiyoga / 22439

Course name: Laboratory Digital Engineering

Group: A

Faculty: Communication and Environment

Deadline: 24.02.2019

Semester: WS 2018/19

Supervisor: Prof. Dr.-Ing. Christian Ressel

Dipl.-Ing. (FH) Nicole Yazici

Table of Contents

Challenge #1	3
Challenge #2	6
Challenge #3	8
Challenge #4	11
Challenge #5	13
Challenge #6	24
Challenge #7	45

Challenge #1

Abstract:

Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

To start the challenge, our group first setup the initial parameters for the Arduino software, that is doing step 5 and 6 of the Description. Then we copy the given code into the Arduino software and compile the sketch, the sketch works without problem.

Next, we tried removing the /n from the code to see what will happen, turns out the function of /n is to make a space between each line of text.

Finally, we tried changing the baud rate and putting in our own text. Our team found out that the baud rate has to be the same both in the code and in the Serial Monitor for the text to appear normally in the Serial Monitor, else it will print out an unintelligible text.

Pictures:



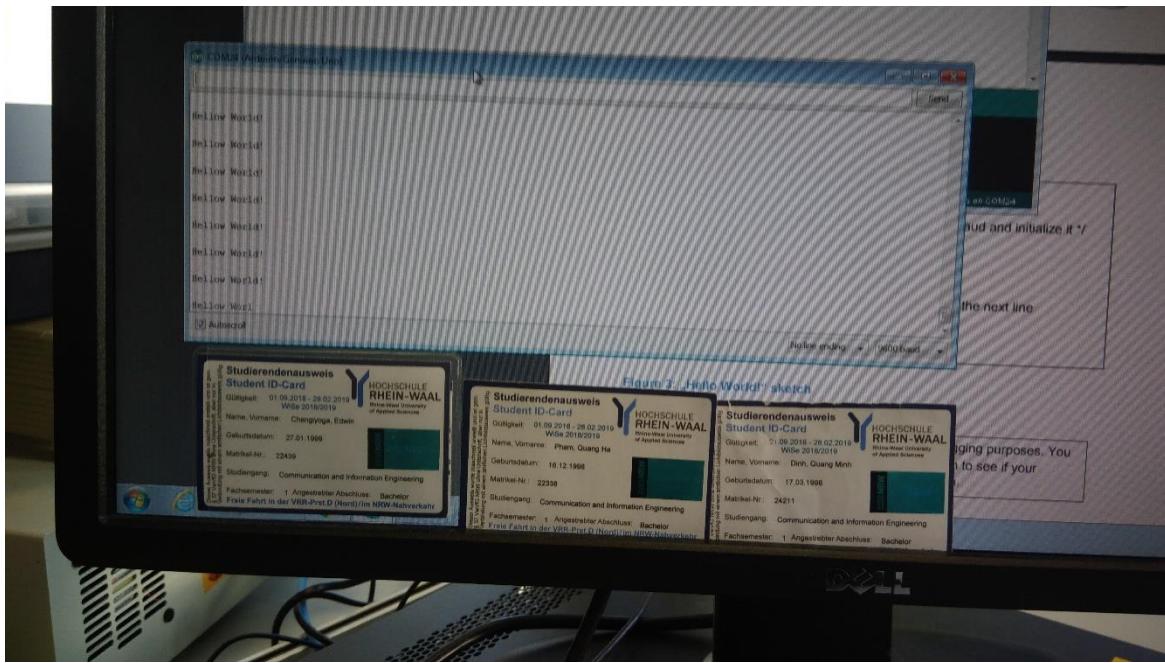


Figure 3... „Hello World!“ sketch

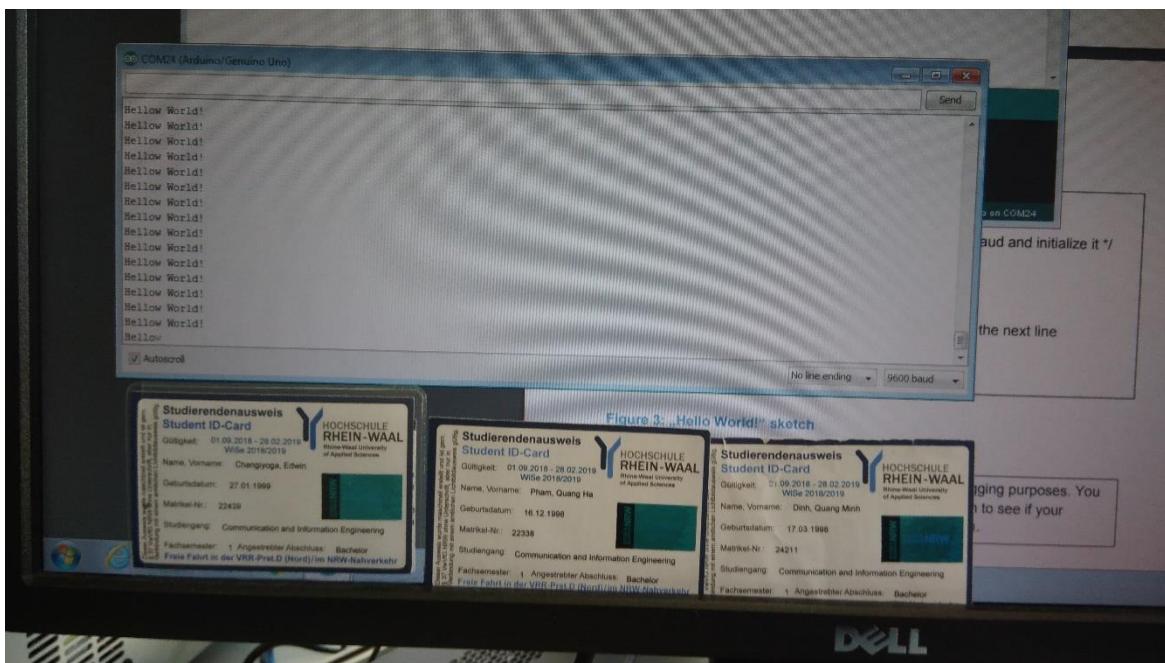
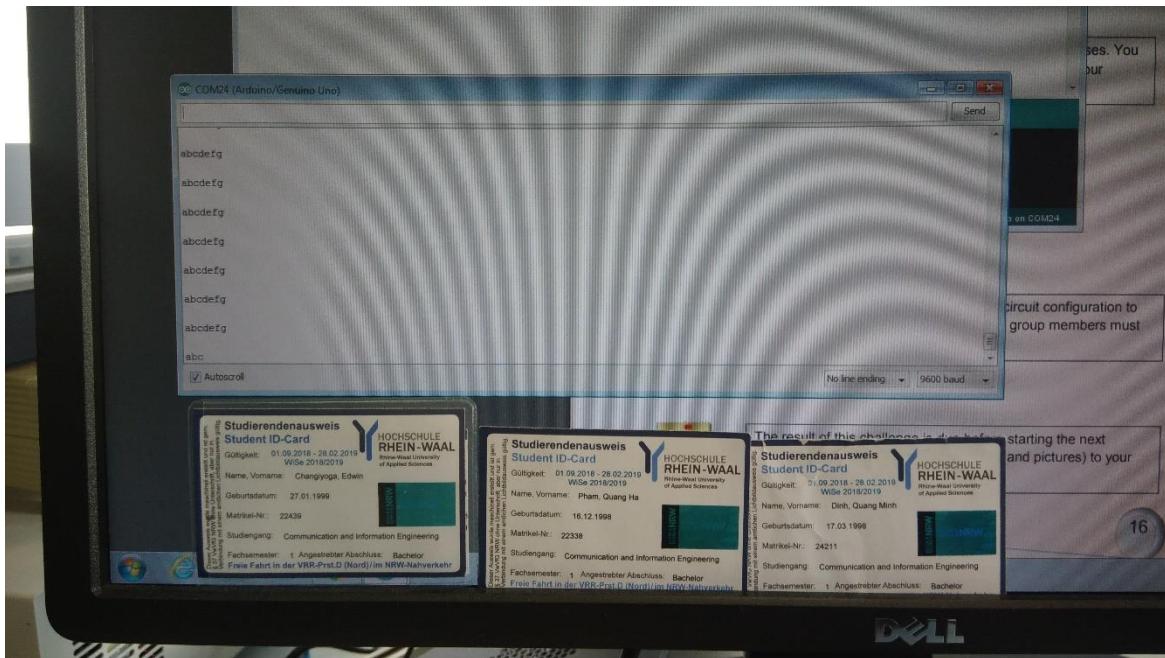


Figure 3... „Hello World!“ sketch



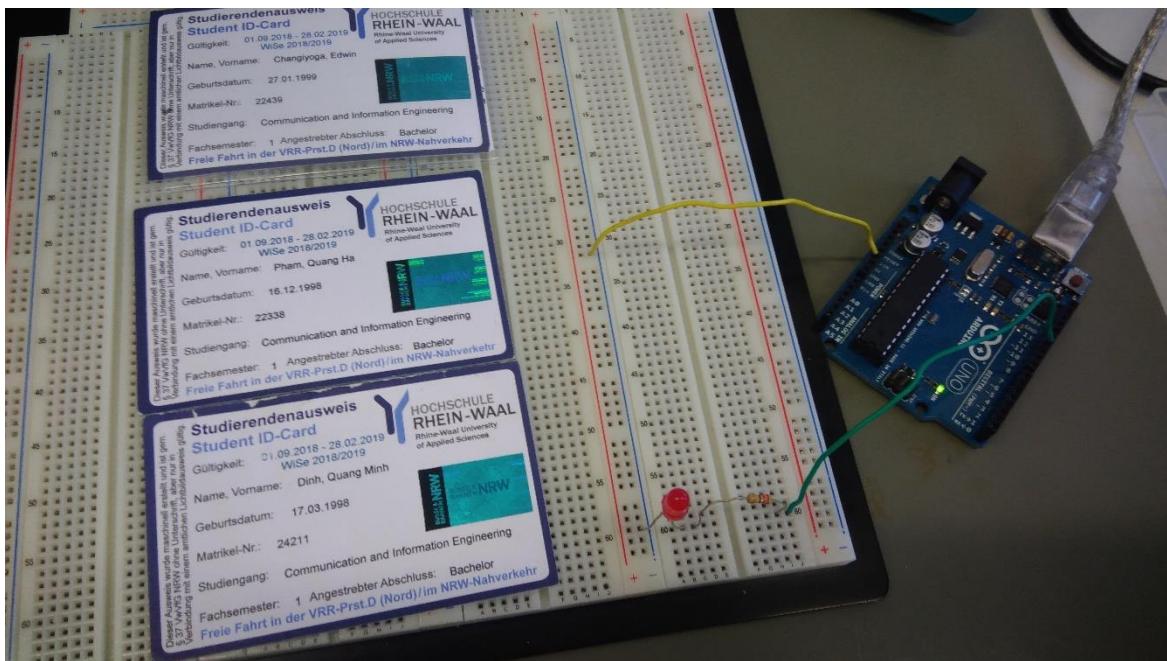
Challenge #2

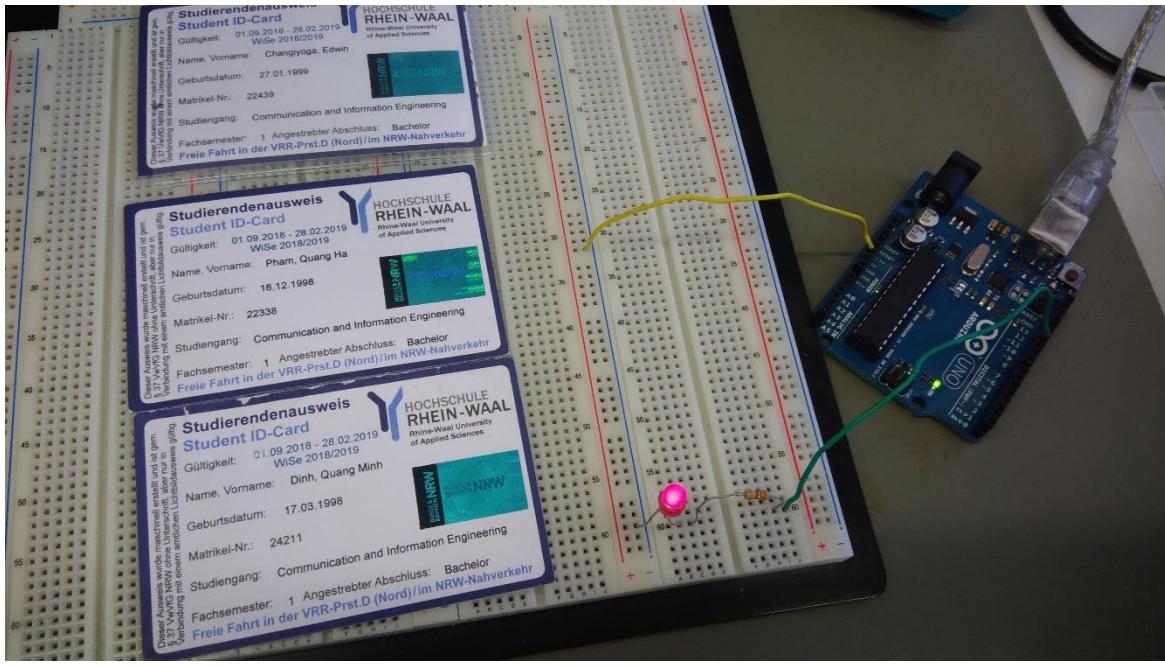
Abstract:

Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

We have informed ourselves about the function `delay()`. As the name suggests, this function serves the purpose of creating a delay before an input is executed, the higher the delay value, the less frequent the LED blinks. The unit for the function `delay()` by default is ms (millisecond).

Pictures:





Challenge #3

Abstract:

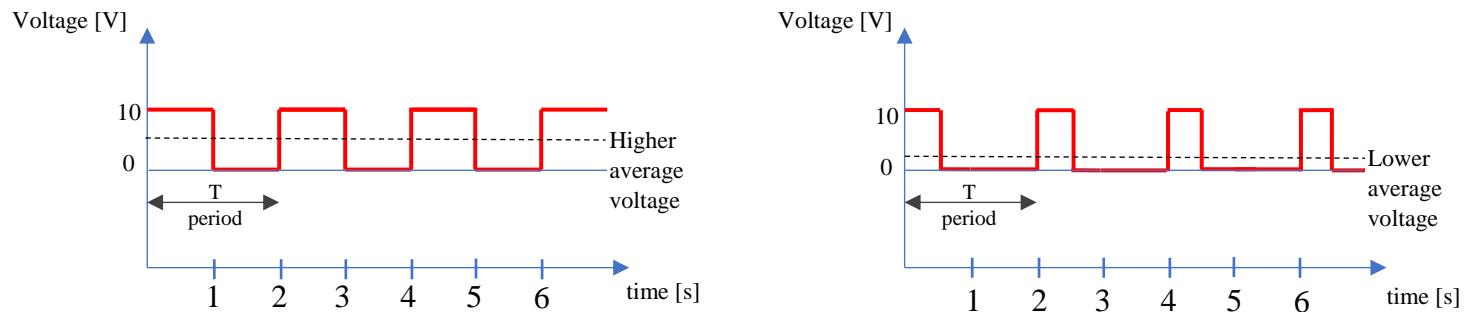
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

We have informed ourselves about the function of `analogWrite()`, the purpose of this function is to write an analog value (PWM wave) to a pin.

The higher the analog value of `analogWrite()`, the brighter the LED shines. First, our group put in an analog value of 100 to see how bright the LED becomes, then we lower the value down to 50 and took pictures to compare the two cases. Since we don't know how to use the Oscilloscope properly, we relied on the Auto Measure function of this Oscilloscope for measurements.

As for the meaning of PWM, it is short for Pulse Width Modulation. Changing the analog value to a pin changes the width of the pulse, this results in changing the average Voltage. There are many uses for this technology, for example dimming the lights or changing how fast a motor can spin.

We made the figure below to graphically represent the above-mentioned theory.

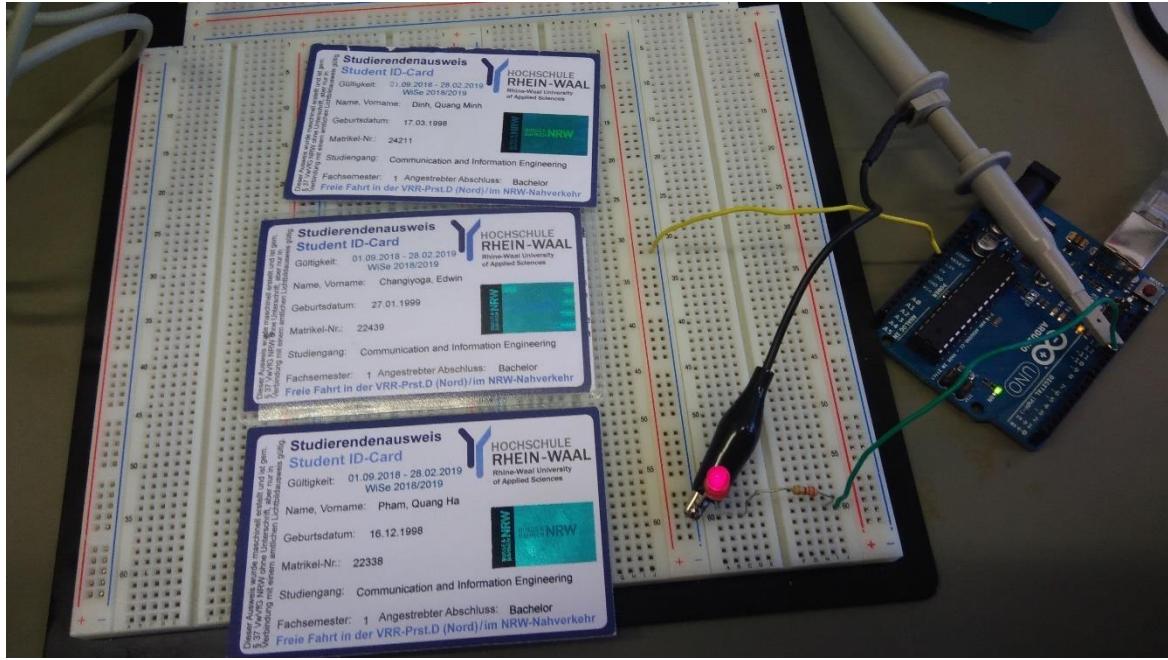


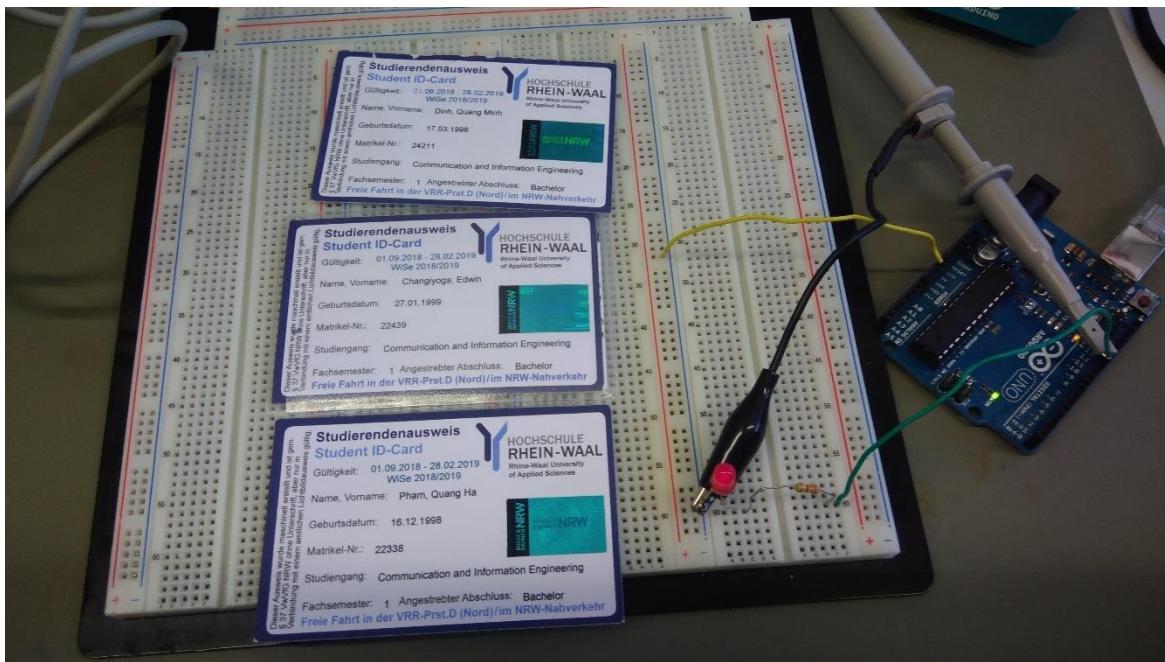
For this example, the time for a period is 2s.

The pulse on the left is active for 1s – half the time of a period, so the average voltage of the graph on the left will be half the maximum voltage of 10V, which is 5V.

The pulse on the right is active for 0.5s – a quarter of a period, so the average voltage of the graph on the right is will be a quarter of the maximum voltage of 10V, which is 2.5V.

Pictures:





Challenge #4

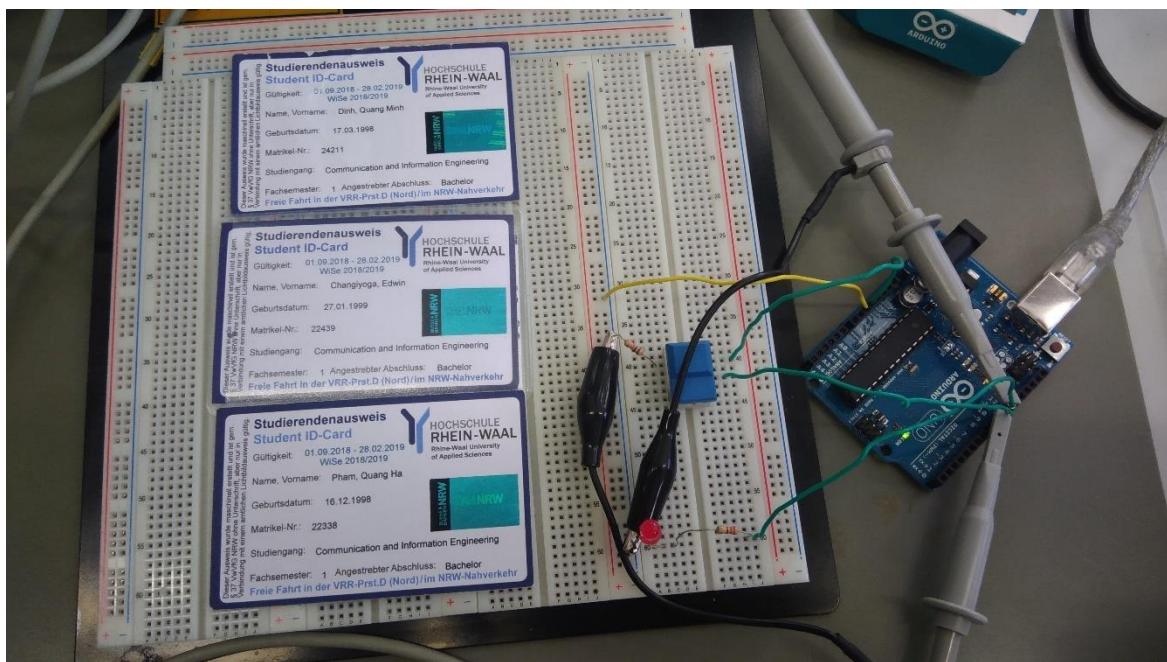
Abstract:

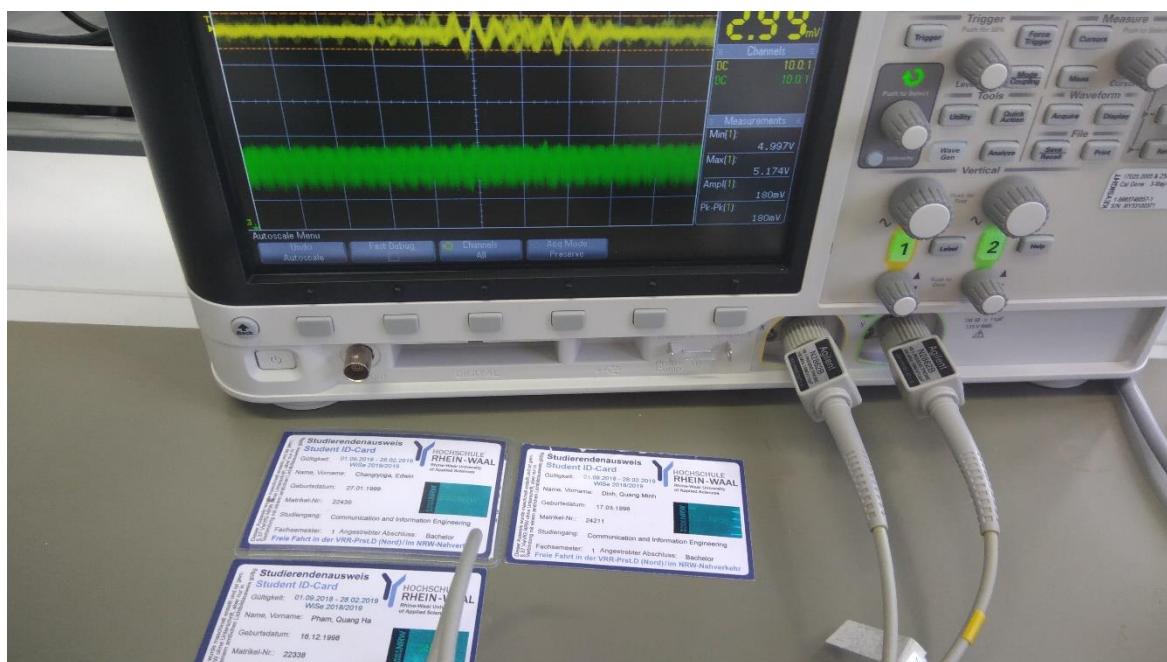
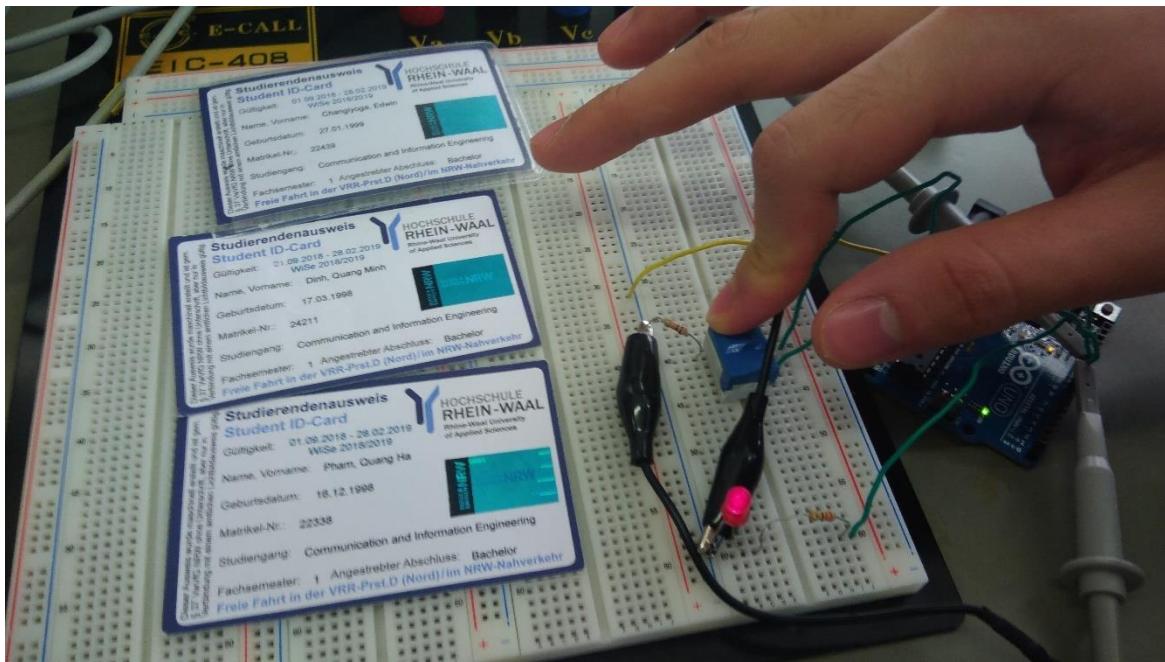
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

The function `digitalRead()` enables the Arduino to read the value of a specific digital pin, this can either be LOW or HIGH.

When we exchange to a switch with Pull-Up-Resistor, the LED shines by default, only when we press the button that the LED will turn off.

Pictures:





Challenge #5

Abstract:

Our group managed to follow the steps given in the Description and completed the challenge without anything special to report.

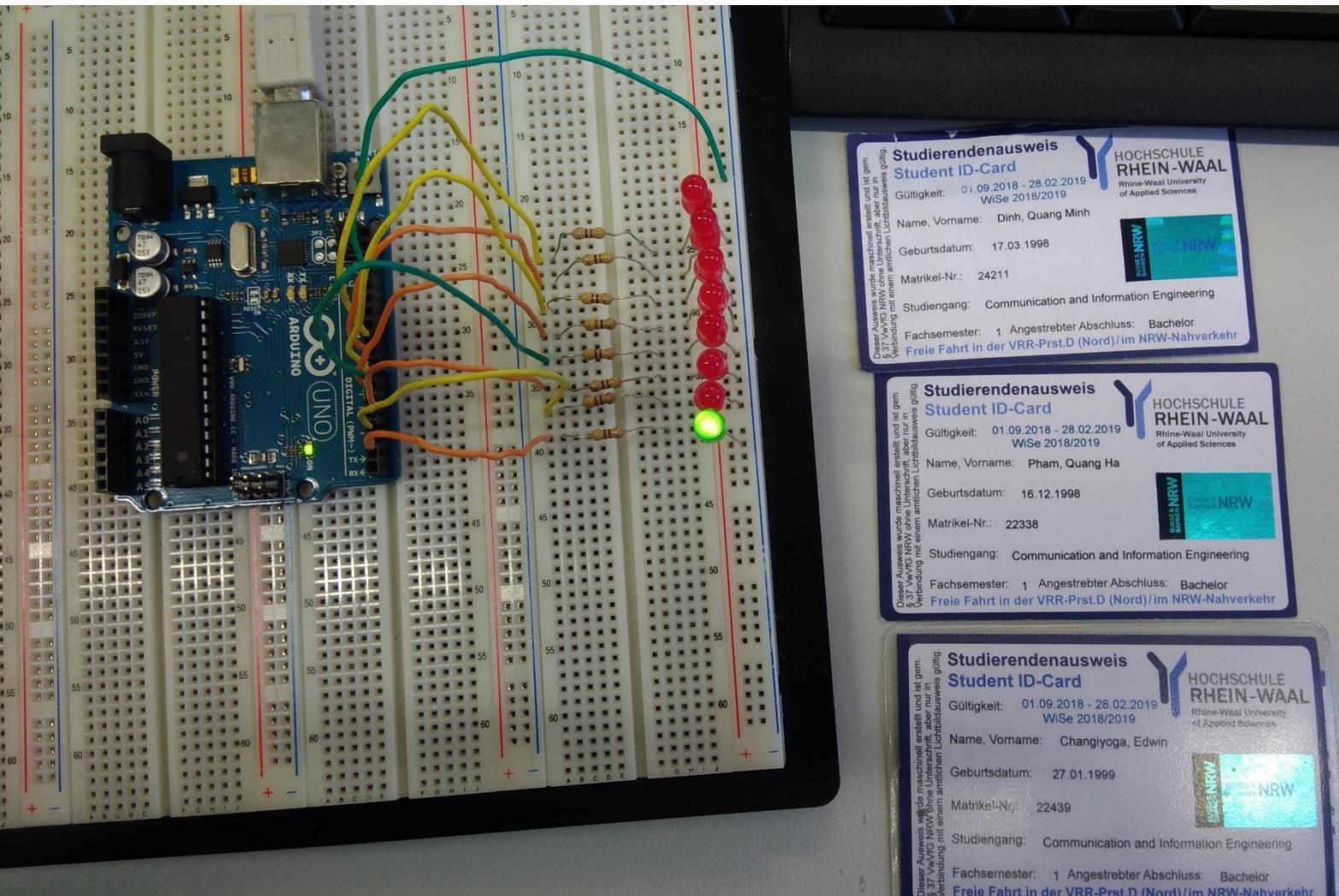
Our code used:

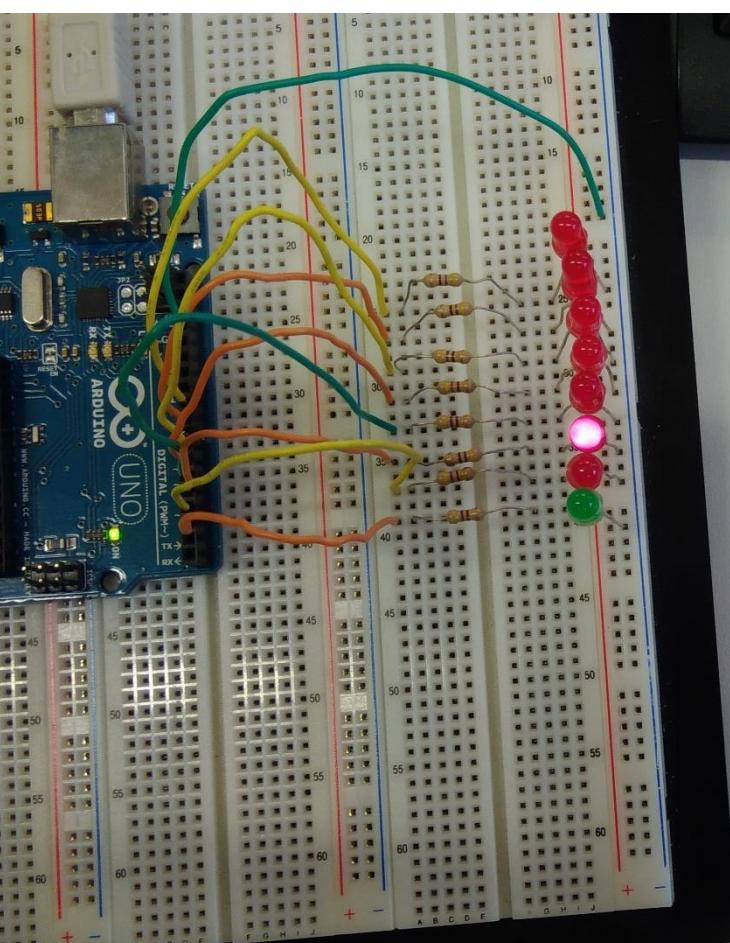
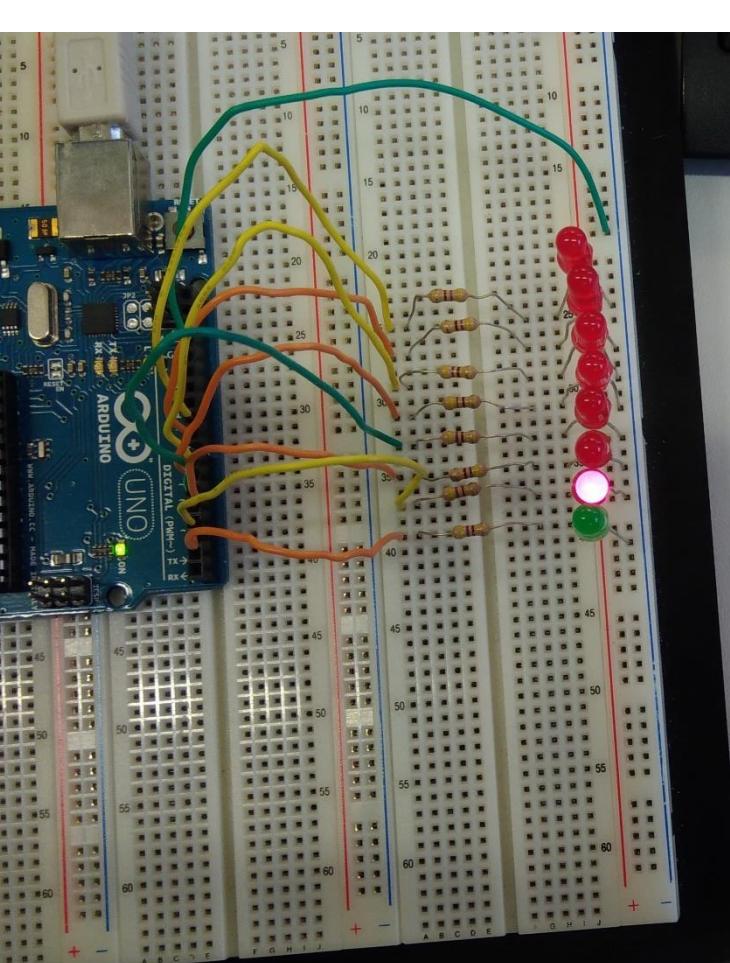
```
int LED_PIN_2 = 2;  
int LED_PIN_3 = 3;  
int LED_PIN_4 = 4;  
int LED_PIN_5 = 5;  
int LED_PIN_6 = 6;  
int LED_PIN_7 = 7;  
int LED_PIN_8 = 8;  
int LED_PIN_9 = 9;  
  
void setup (void)  
{  
    // set required digital pins to Output  
    pinMode(LED_PIN_2, OUTPUT);  
    pinMode(LED_PIN_3, OUTPUT);  
    pinMode(LED_PIN_4, OUTPUT);  
    pinMode(LED_PIN_5, OUTPUT);  
    pinMode(LED_PIN_6, OUTPUT);  
    pinMode(LED_PIN_7, OUTPUT);  
    pinMode(LED_PIN_8, OUTPUT);  
    pinMode(LED_PIN_9, OUTPUT);  
}  
void loop (void)  
{
```

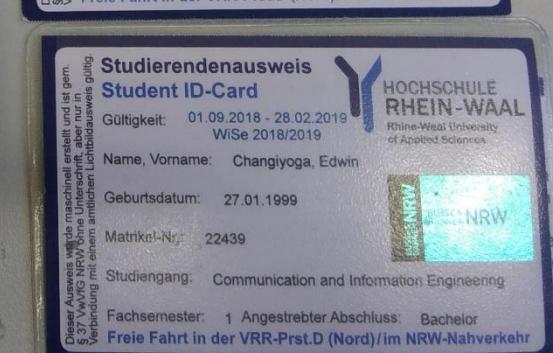
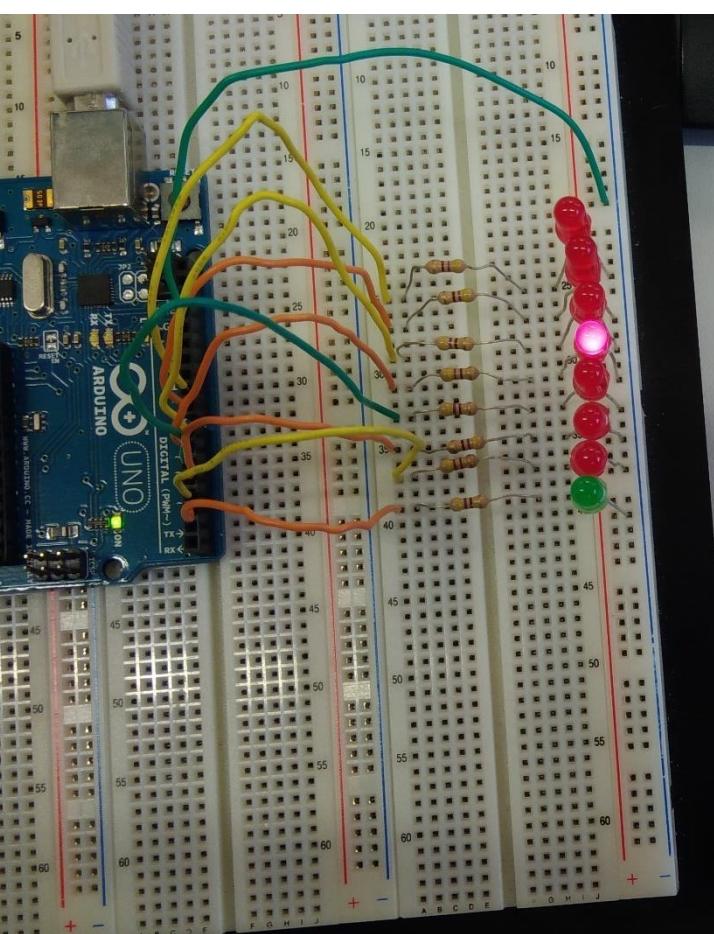
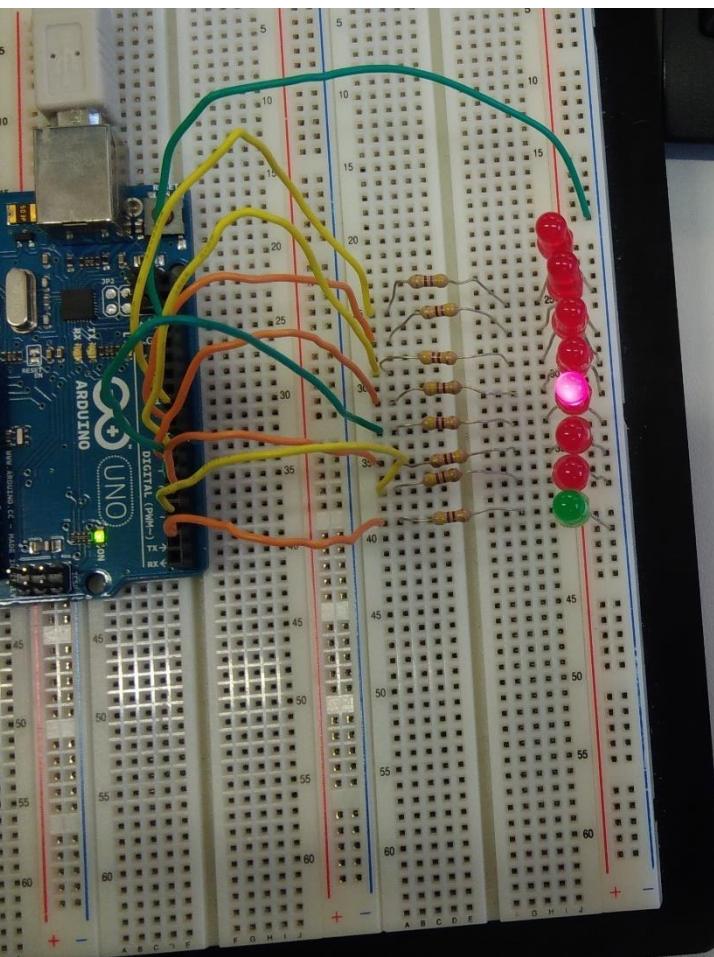
```
// Set Pin to high level (LED on). This is the beginning of the loop.  
digitalWrite(LED_PIN_2, HIGH);  
digitalWrite(LED_PIN_3, LOW);  
digitalWrite(LED_PIN_4, LOW);  
digitalWrite(LED_PIN_5, LOW);  
digitalWrite(LED_PIN_6, LOW);  
digitalWrite(LED_PIN_7, LOW);  
digitalWrite(LED_PIN_8, LOW);  
digitalWrite(LED_PIN_9, LOW);  
delay (1000);  
  
// Set Pin to low level (LED off)  
digitalWrite(LED_PIN_2, LOW);  
digitalWrite(LED_PIN_3, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_3, LOW);  
digitalWrite(LED_PIN_4, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_4, LOW);  
digitalWrite(LED_PIN_5, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_5, LOW);  
digitalWrite(LED_PIN_6, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_6, LOW);  
digitalWrite(LED_PIN_7, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_7, LOW);  
digitalWrite(LED_PIN_8, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_8, LOW);  
digitalWrite(LED_PIN_9, HIGH);
```

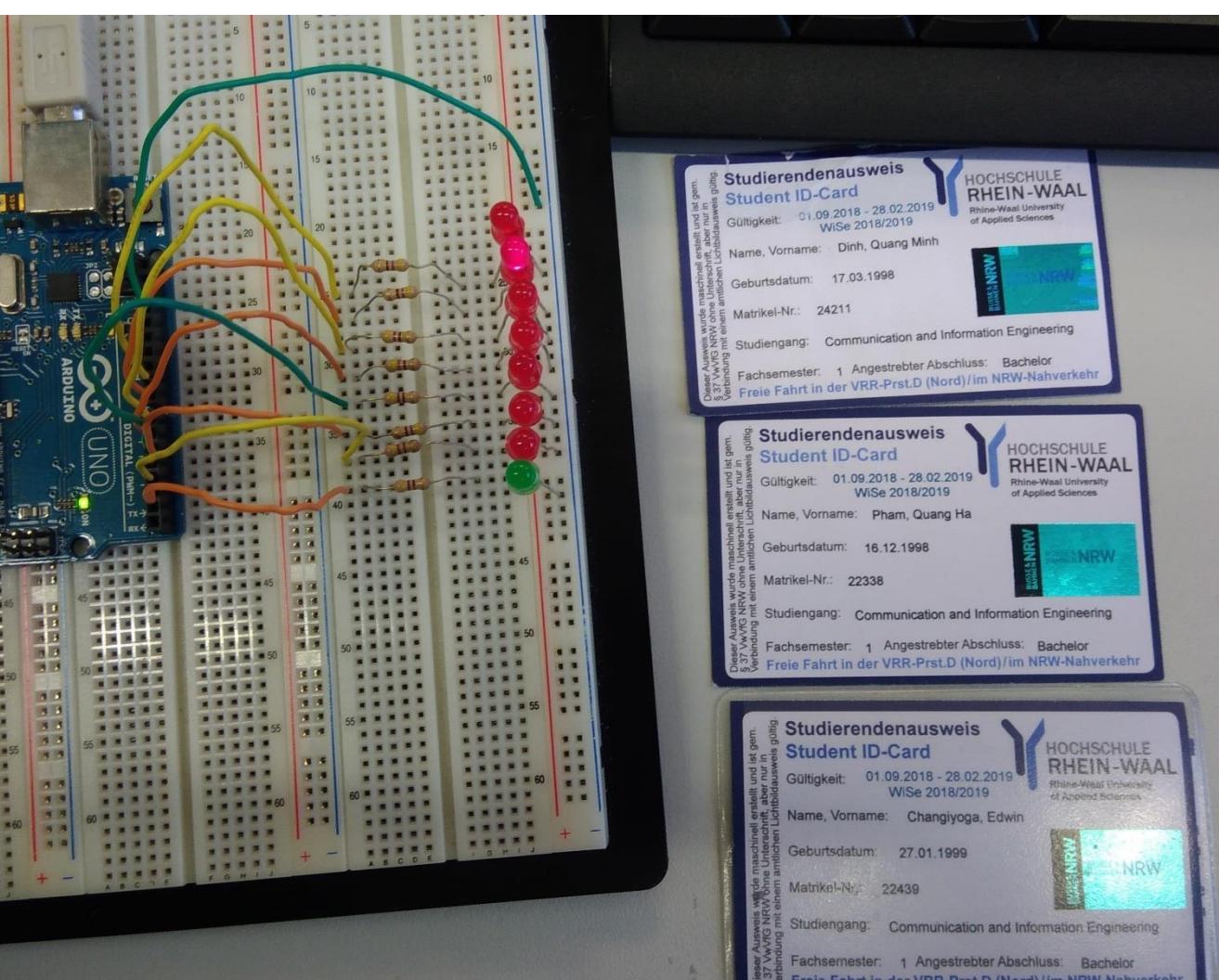
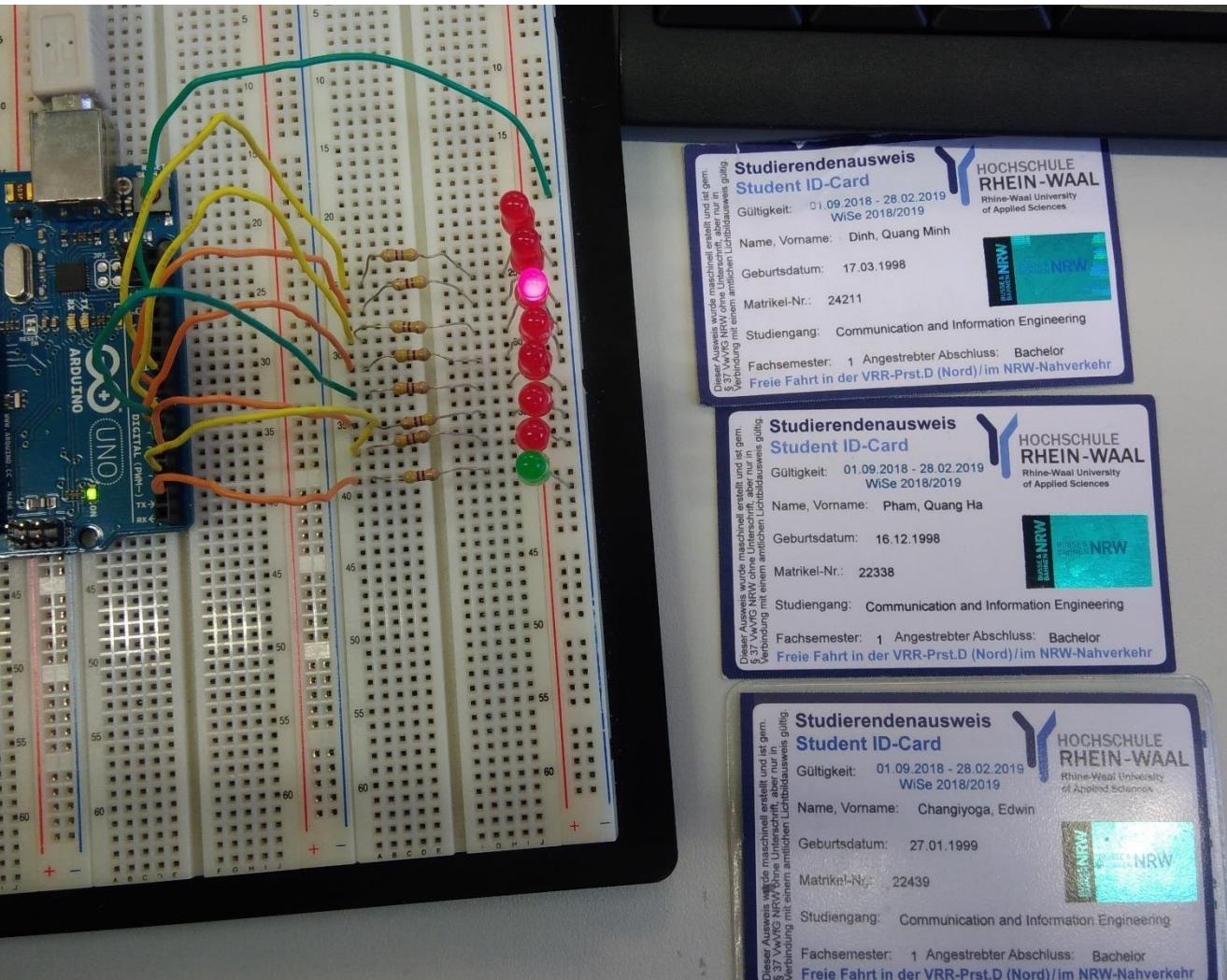
```
delay (1000);  
digitalWrite(LED_PIN_9, LOW);  
digitalWrite(LED_PIN_8, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_8, LOW);  
digitalWrite(LED_PIN_7, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_7, LOW);  
digitalWrite(LED_PIN_6, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_6, LOW);  
digitalWrite(LED_PIN_5, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_5, LOW);  
digitalWrite(LED_PIN_4, HIGH);  
delay (1000);  
digitalWrite(LED_PIN_4, LOW);  
digitalWrite(LED_PIN_3, HIGH);  
delay (1000);  
// Loop over. Back to the beginning of the loop.  
}
```

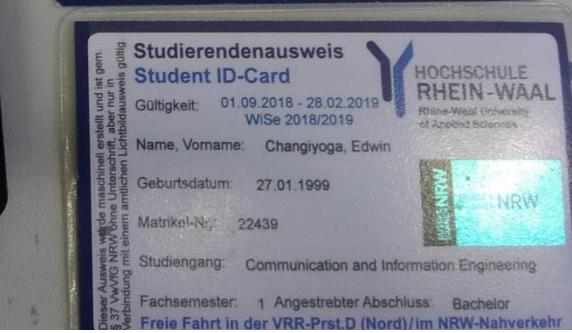
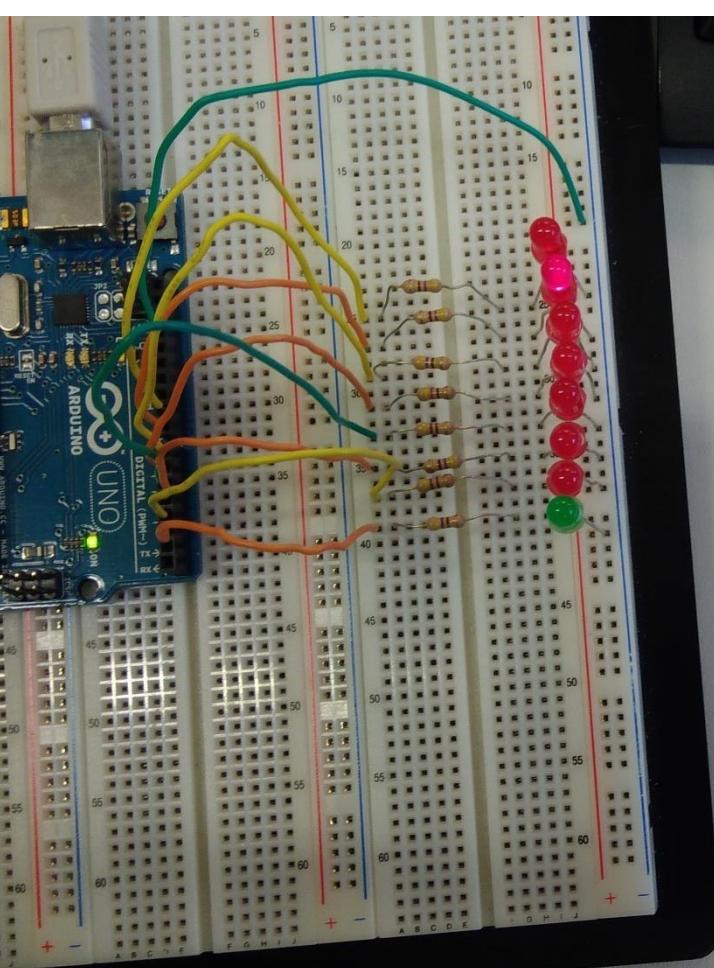
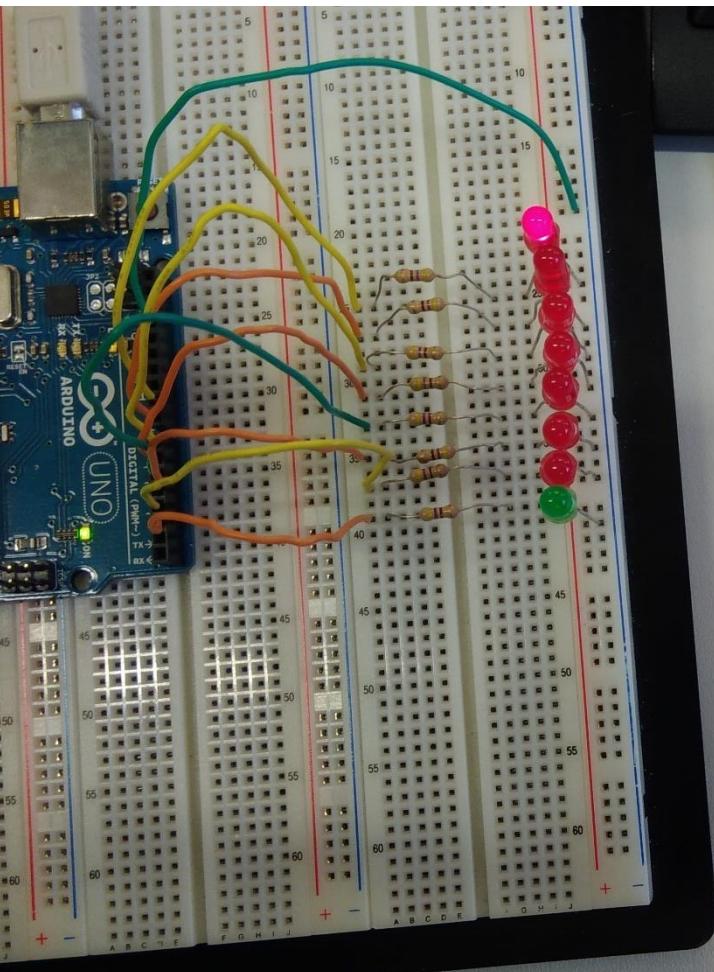
Pictures:

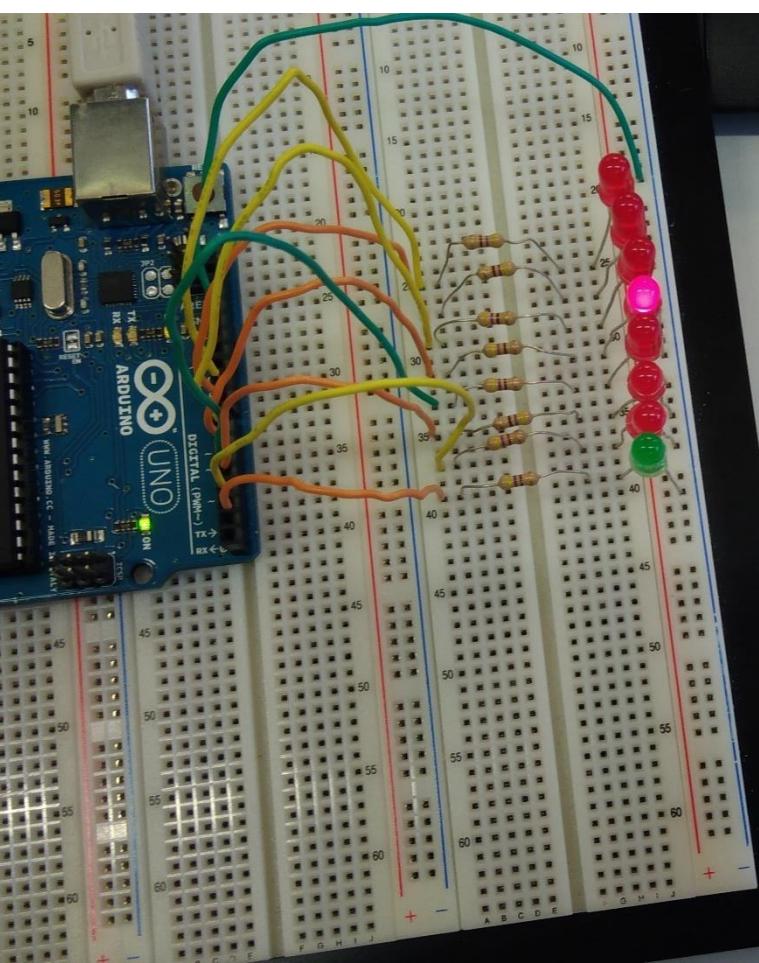
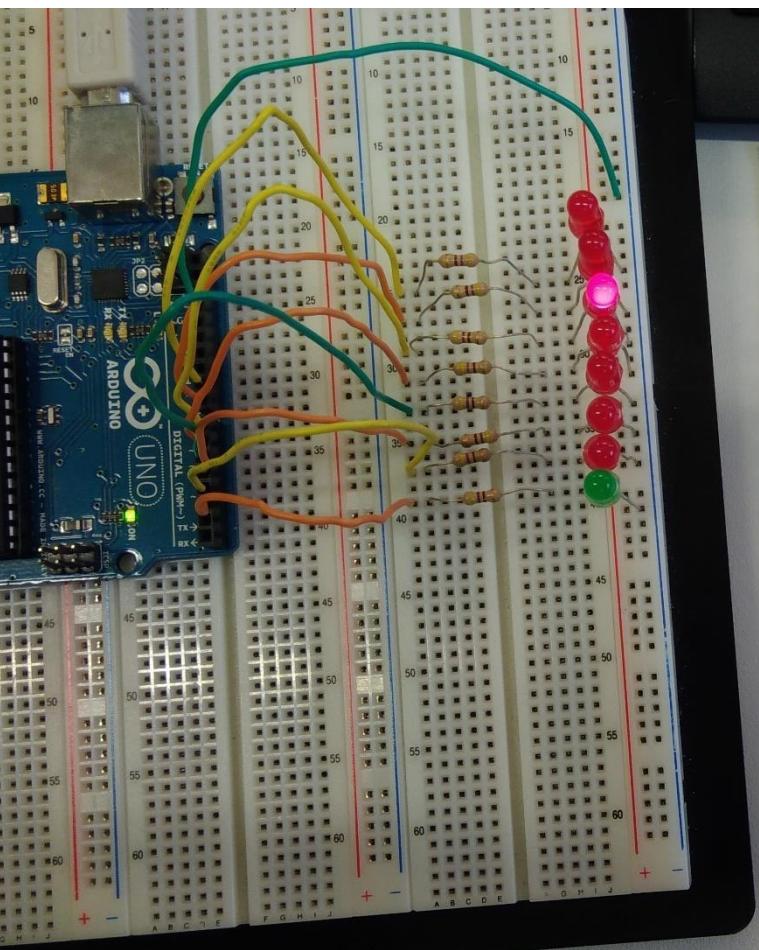


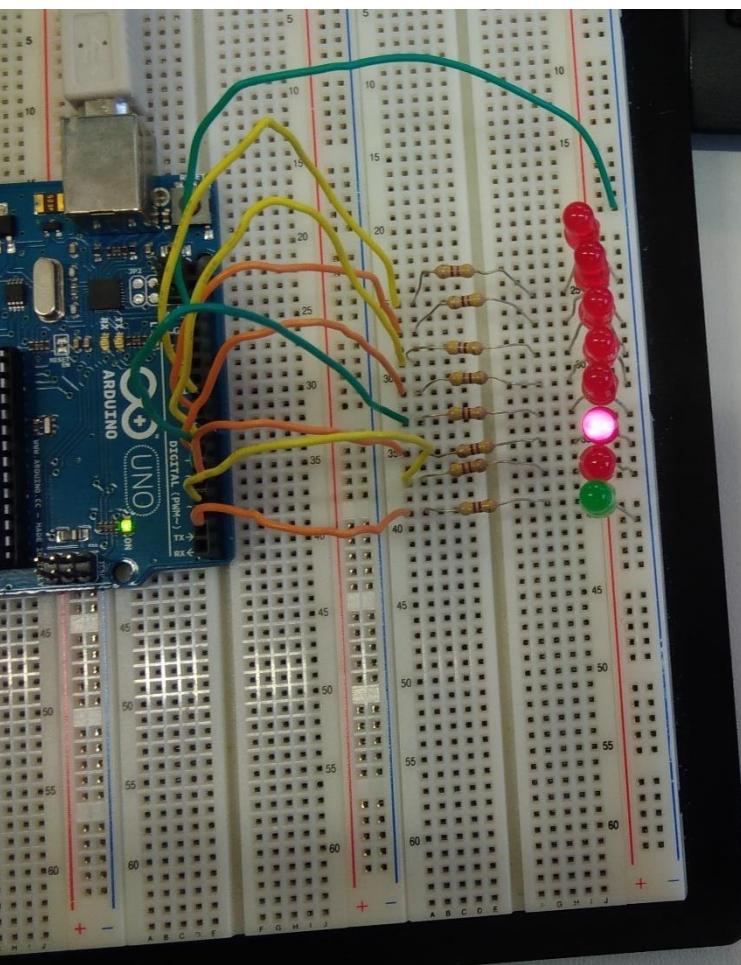
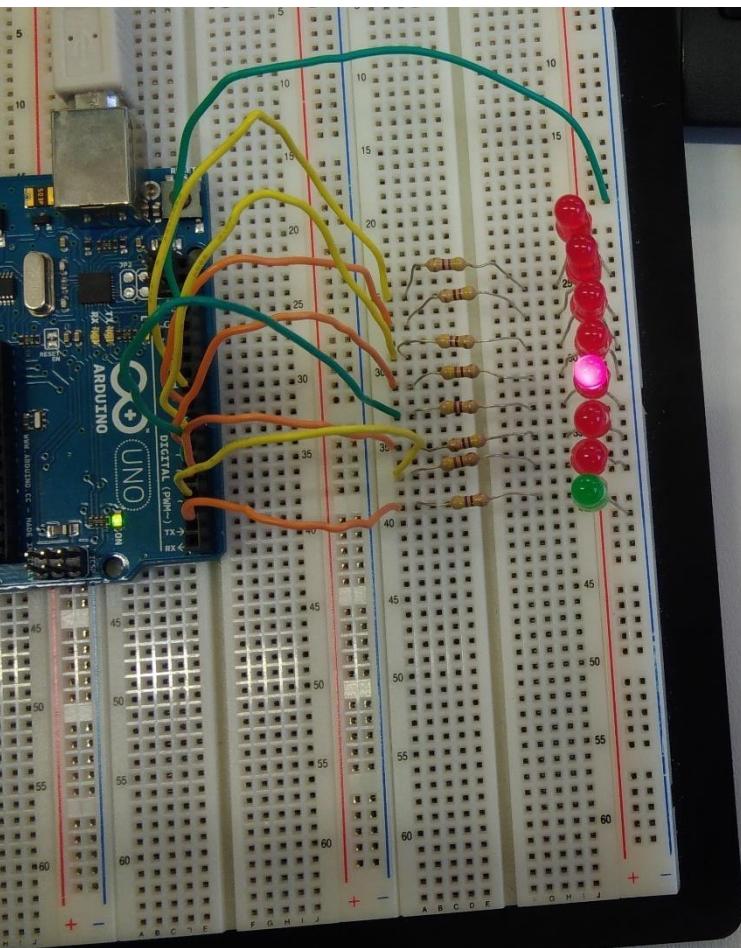


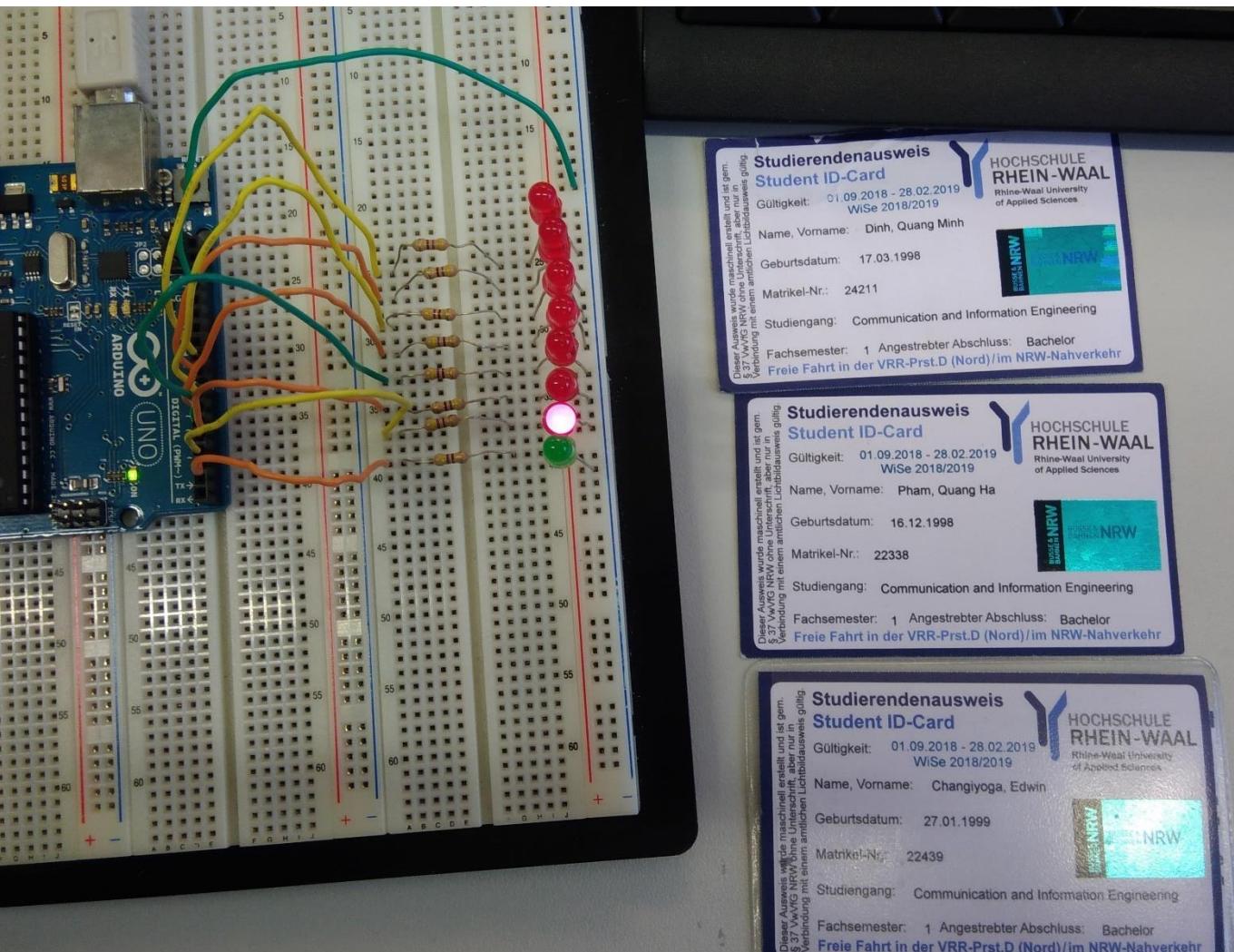












Challenge #6

Abstract:

Our group managed to follow the steps given in the Description and completed the challenge without anything special to report.

Our code used:

```
int LED_PIN_2 = 2;  
int LED_PIN_3 = 3;  
int LED_PIN_4 = 4;  
int LED_PIN_5 = 5;  
int LED_PIN_6 = 6;  
int LED_PIN_7 = 7;  
int LED_PIN_8 = 8;  
int LED_PIN_9 = 9;  
  
void setup ()  
{  
  
pinMode(LED_PIN_2, OUTPUT);  
pinMode(LED_PIN_3, OUTPUT);  
pinMode(LED_PIN_4, OUTPUT);  
pinMode(LED_PIN_5, OUTPUT);  
pinMode(LED_PIN_6, OUTPUT);  
pinMode(LED_PIN_7, OUTPUT);  
pinMode(LED_PIN_8, OUTPUT);  
pinMode(LED_PIN_9, OUTPUT);  
  
}  
void loop ()  
{
```

```

for (int i=0; i<=9; i++)
    // for (Initial value, Condition, Execute).
    // for (i=0 at the beginning, if i is smaller or equal to 9, value of i counts up by 1).
    {
        switch (i)
            // the Case used will depends on the value of "i"
            {
                case 0:
                    digitalWrite(LED_PIN_2, HIGH);
                    digitalWrite(LED_PIN_3, HIGH);
                    digitalWrite(LED_PIN_4, HIGH);
                    digitalWrite(LED_PIN_5, HIGH);
                    digitalWrite(LED_PIN_6, HIGH);
                    digitalWrite(LED_PIN_7, HIGH);
                    digitalWrite(LED_PIN_8, LOW);
                    digitalWrite(LED_PIN_9, HIGH);
                    delay(1000);
                    break;

                case 1:
                    digitalWrite(LED_PIN_2, LOW);
                    digitalWrite(LED_PIN_3, HIGH);
                    digitalWrite(LED_PIN_4, HIGH);
                    digitalWrite(LED_PIN_5, LOW);
                    digitalWrite(LED_PIN_6, LOW);
                    digitalWrite(LED_PIN_7, LOW);
                    digitalWrite(LED_PIN_8, LOW);
                    digitalWrite(LED_PIN_9, HIGH);
                    delay(1000);
                    break;
            }
    }
}

```

```
case 2:
```

```
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, LOW);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);
```

```
break;
```

```
case 3:
```

```
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);
```

```
break;
```

```
case 4:
```

```
    digitalWrite(LED_PIN_2, LOW);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, LOW);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);
```

```
break;
```

```
case 5:
```

```
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, LOW);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);
```

```
break;
```

```
case 6:
```

```
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, LOW);
    digitalWrite(LED_PIN_4, HIGH);
    digitalWrite(LED_PIN_5, HIGH);
    digitalWrite(LED_PIN_6, HIGH);
    digitalWrite(LED_PIN_7, HIGH);
    digitalWrite(LED_PIN_8, HIGH);
    digitalWrite(LED_PIN_9, HIGH);

    delay(1000);

    break;
```

```
case 7:
```

```
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, HIGH);
    digitalWrite(LED_PIN_4, HIGH);
    digitalWrite(LED_PIN_5, LOW);
    digitalWrite(LED_PIN_6, LOW);
    digitalWrite(LED_PIN_7, LOW);
    digitalWrite(LED_PIN_8, LOW);
    digitalWrite(LED_PIN_9, HIGH);

    delay(1000);

    break;
```

```
case 8:
```

```
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
  
    delay(1000);
```

```
break;
```

```
case 9:
```

```
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
  
    delay(1000);
```

```
break;
```

```

default:

// numbers not in the range 0 to 9 will switch off all segments
digitalWrite(LED_PIN_2, LOW);
digitalWrite(LED_PIN_3, LOW);
digitalWrite(LED_PIN_4, LOW);
digitalWrite(LED_PIN_5, LOW);
digitalWrite(LED_PIN_6, LOW);
digitalWrite(LED_PIN_7, LOW);
digitalWrite(LED_PIN_8, LOW);
digitalWrite(LED_PIN_9, LOW);

delay(1000);

break;

}

}

// First loop ends, starting second loop, starts with i=8.

for (int i=8; i>0; i--)

{
switch (i)

{
case 0:

digitalWrite(LED_PIN_2, HIGH);
digitalWrite(LED_PIN_3, HIGH);
digitalWrite(LED_PIN_4, HIGH);
digitalWrite(LED_PIN_5, HIGH);
digitalWrite(LED_PIN_6, HIGH);
digitalWrite(LED_PIN_7, HIGH);
digitalWrite(LED_PIN_8, LOW);
digitalWrite(LED_PIN_9, HIGH);

delay(1000);

break;
}

```

```
case 1:
```

```
digitalWrite(LED_PIN_2, LOW);  
digitalWrite(LED_PIN_3, HIGH);  
digitalWrite(LED_PIN_4, HIGH);  
digitalWrite(LED_PIN_5, LOW);  
digitalWrite(LED_PIN_6, LOW);  
digitalWrite(LED_PIN_7, LOW);  
digitalWrite(LED_PIN_8, LOW);  
digitalWrite(LED_PIN_9, HIGH);  
  
delay(1000);
```

```
break;
```

```
case 2:
```

```
digitalWrite(LED_PIN_2, HIGH);  
digitalWrite(LED_PIN_3, HIGH);  
digitalWrite(LED_PIN_4, LOW);  
digitalWrite(LED_PIN_5, HIGH);  
digitalWrite(LED_PIN_6, HIGH);  
digitalWrite(LED_PIN_7, LOW);  
digitalWrite(LED_PIN_8, HIGH);  
digitalWrite(LED_PIN_9, HIGH);  
  
delay(1000);
```

```
break;
```

case 3:

```
digitalWrite(LED_PIN_2, HIGH);
digitalWrite(LED_PIN_3, HIGH);
digitalWrite(LED_PIN_4, HIGH);
digitalWrite(LED_PIN_5, HIGH);
digitalWrite(LED_PIN_6, LOW);
digitalWrite(LED_PIN_7, LOW);
digitalWrite(LED_PIN_8, HIGH);
digitalWrite(LED_PIN_9, HIGH);

delay(1000);

break;
```

case 4:

```
digitalWrite(LED_PIN_2, LOW);
digitalWrite(LED_PIN_3, HIGH);
digitalWrite(LED_PIN_4, HIGH);
digitalWrite(LED_PIN_5, LOW);
digitalWrite(LED_PIN_6, LOW);
digitalWrite(LED_PIN_7, HIGH);
digitalWrite(LED_PIN_8, HIGH);
digitalWrite(LED_PIN_9, HIGH);

delay(1000);

break;
```

```
case 5:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, LOW);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);  
break;
```

```
case 6:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, LOW);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);  
break;
```

```
case 7:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, LOW);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, LOW);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);  
break;
```

```
case 8:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(1000);  
break;
```

```

default:

digitalWrite(LED_PIN_2, LOW);
digitalWrite(LED_PIN_3, LOW);
digitalWrite(LED_PIN_4, LOW);
digitalWrite(LED_PIN_5, LOW);
digitalWrite(LED_PIN_6, LOW);
digitalWrite(LED_PIN_7, LOW);
digitalWrite(LED_PIN_8, LOW);
digitalWrite(LED_PIN_9, LOW);
delay(1000);

break;

}

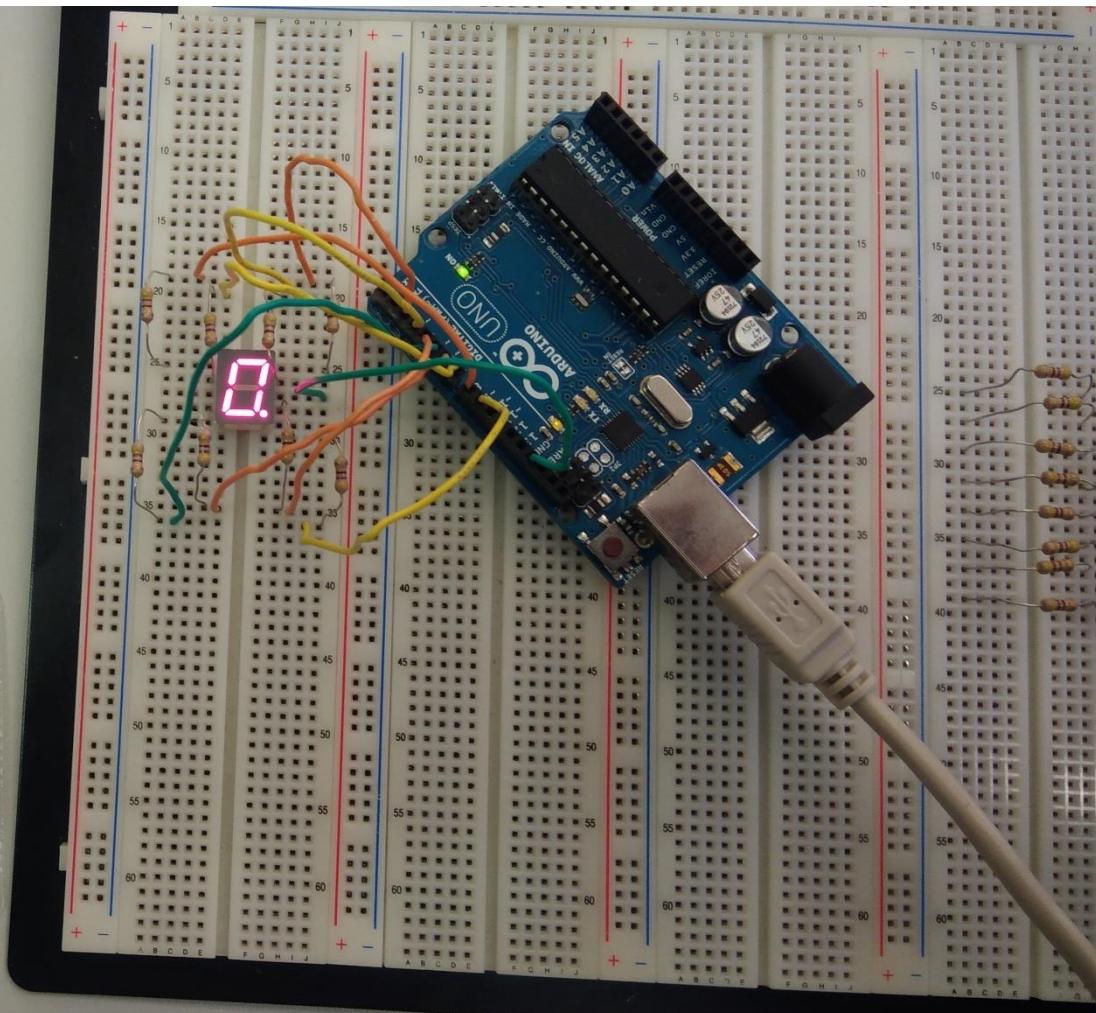
}

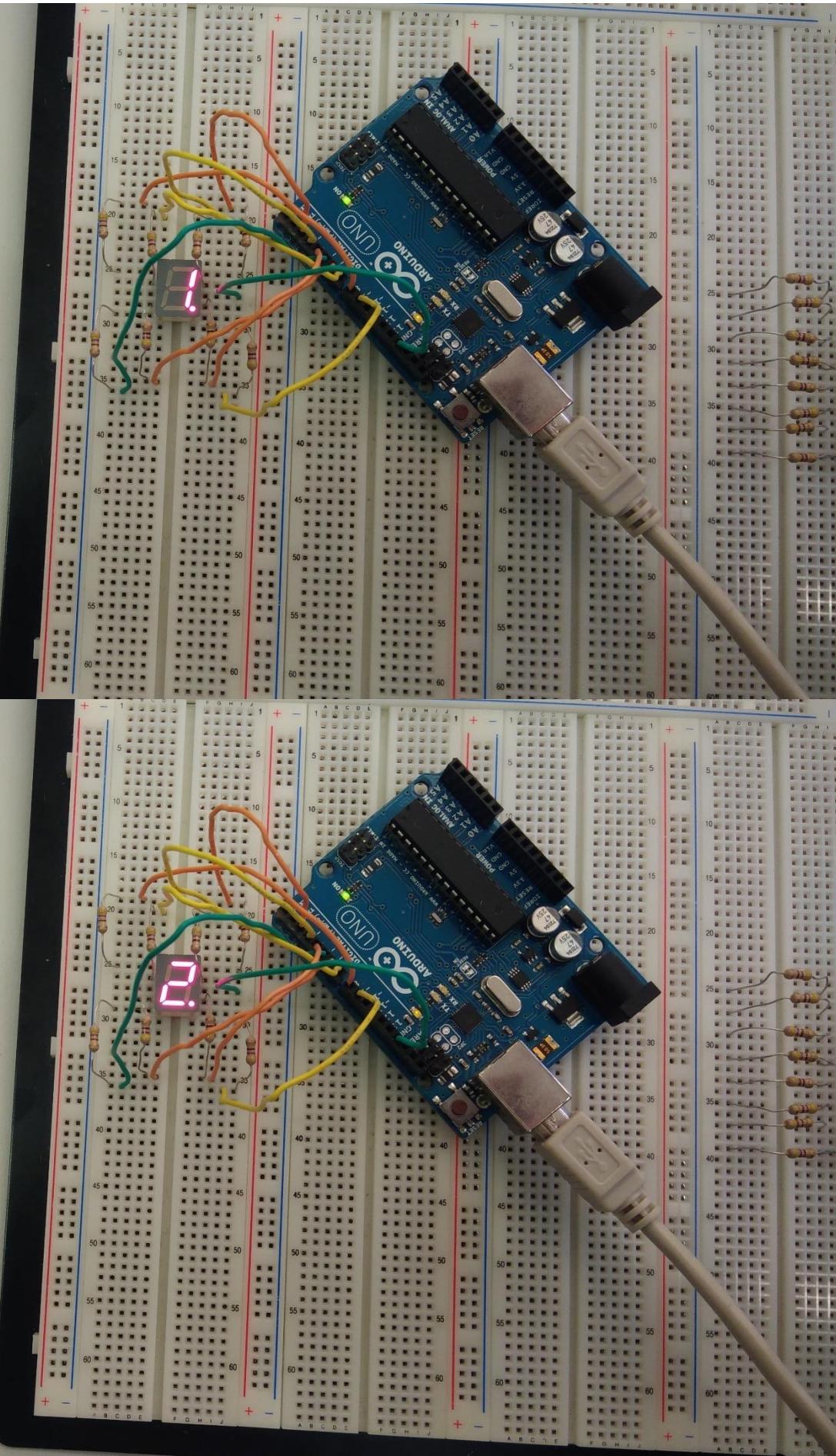
// Second loop complete, entire void (loop) completed. Starts again from 0.

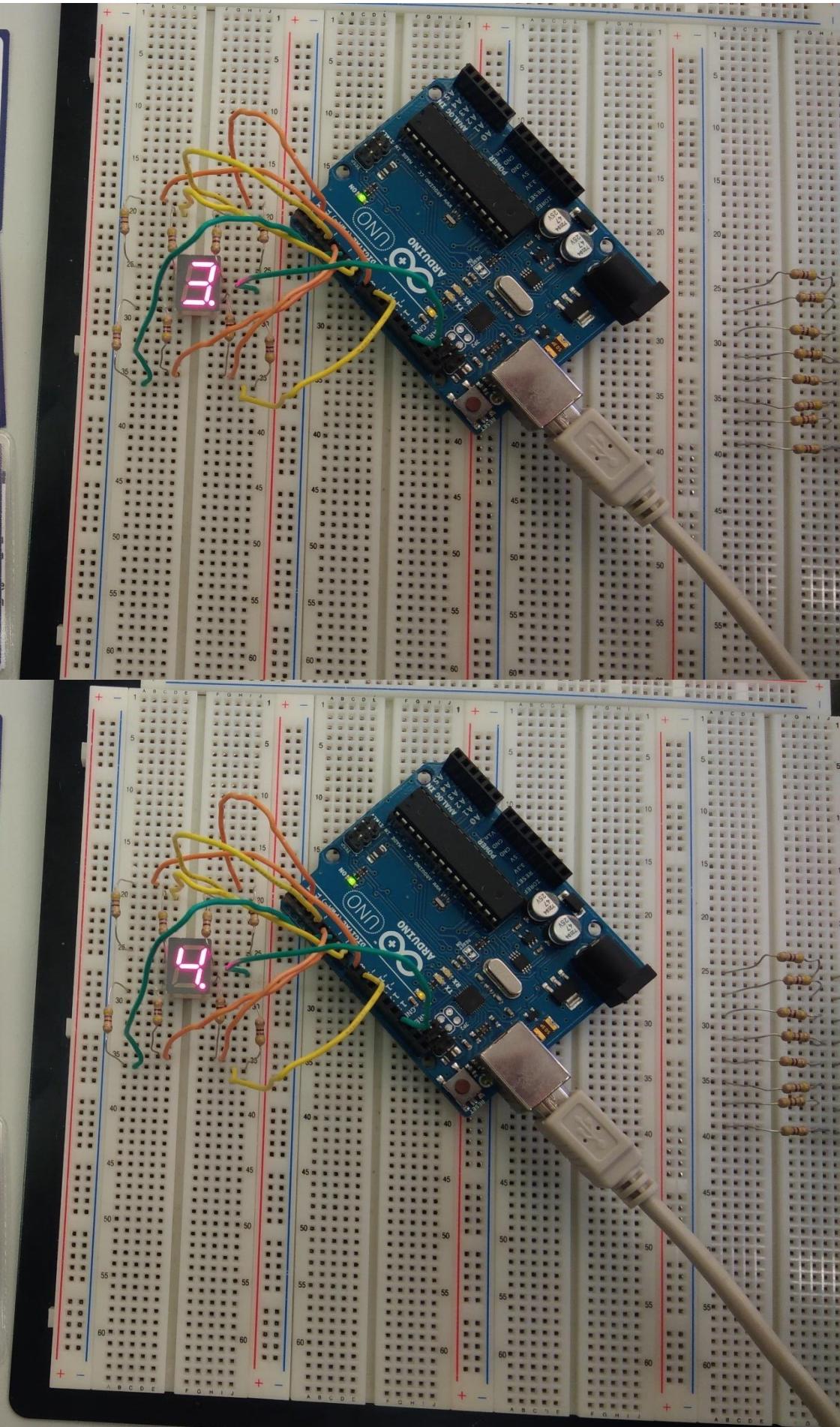
}

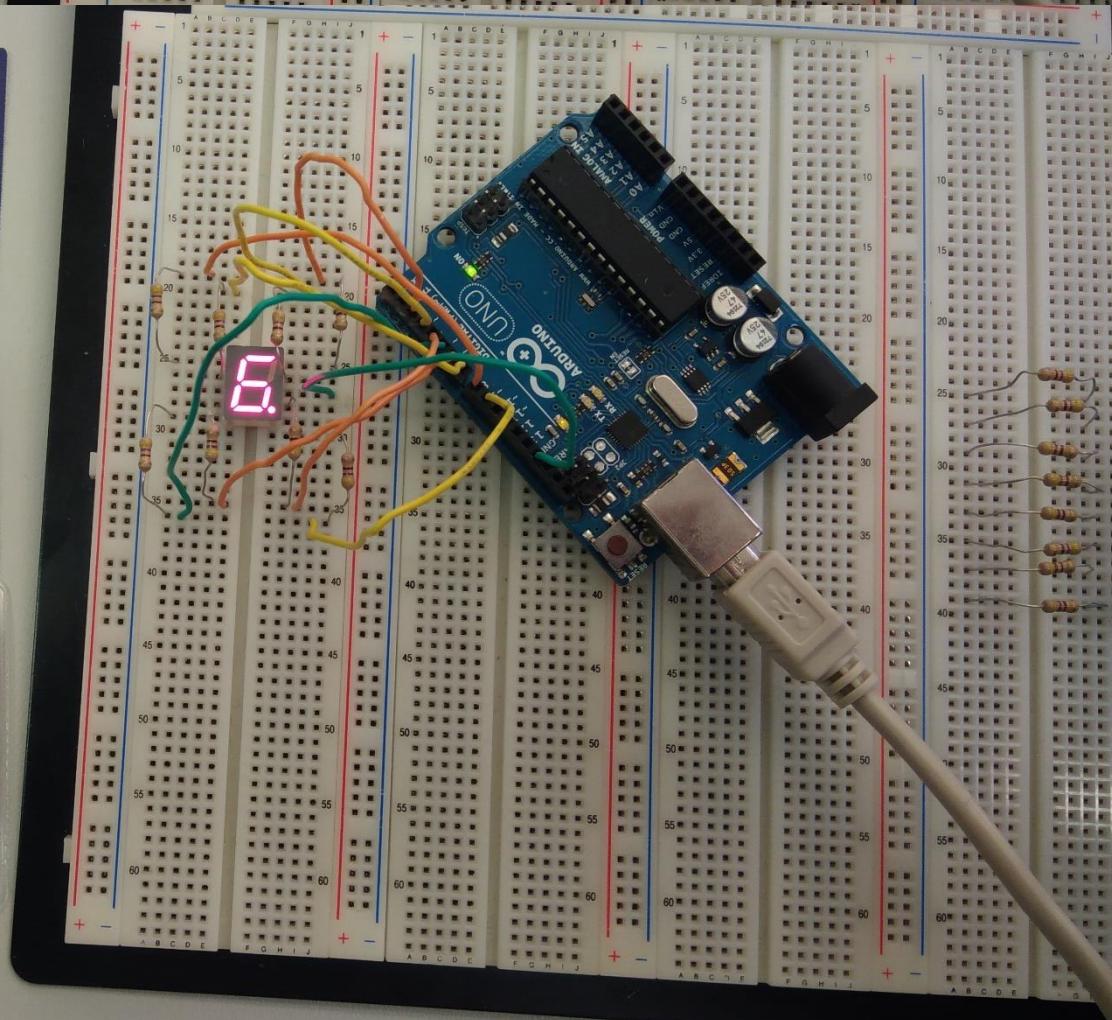
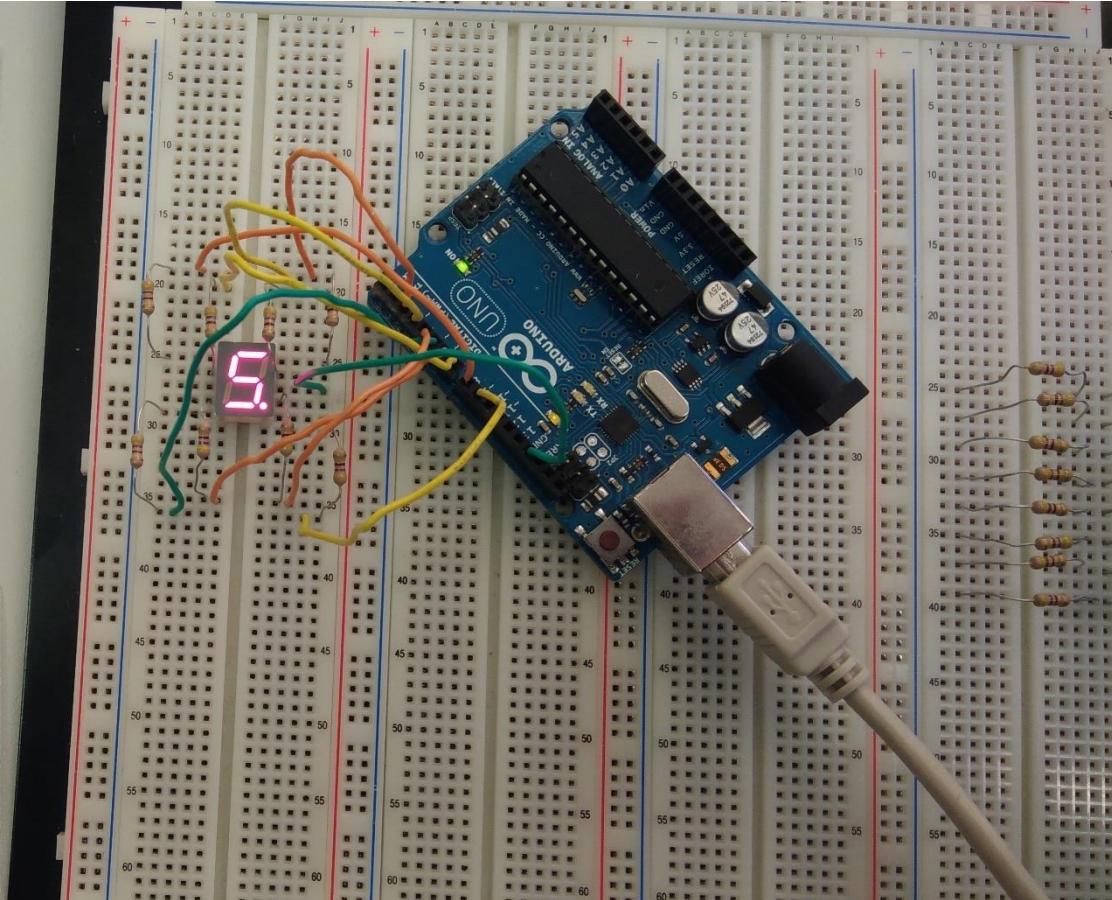
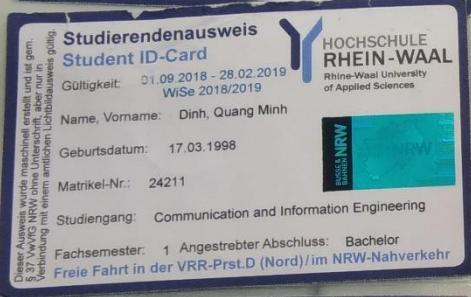
```

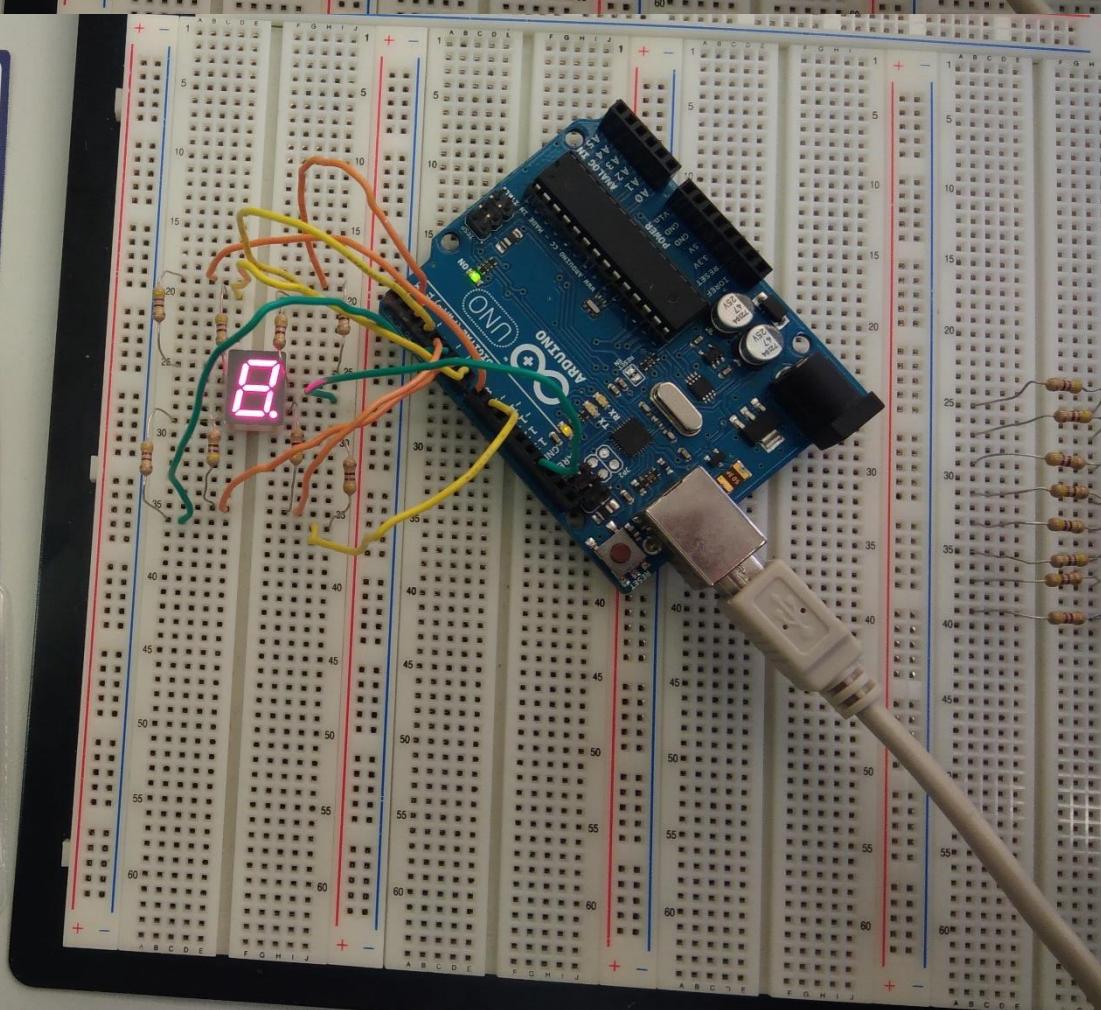
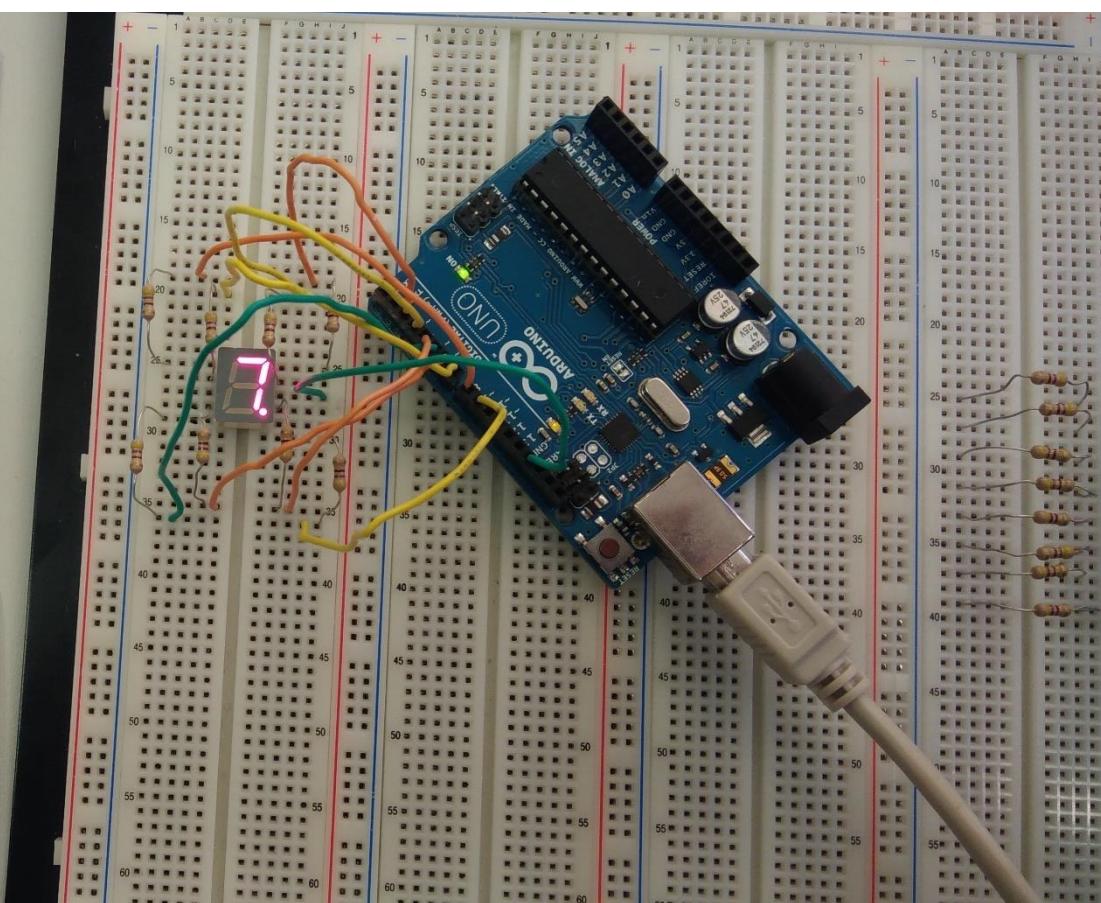
Pictures:

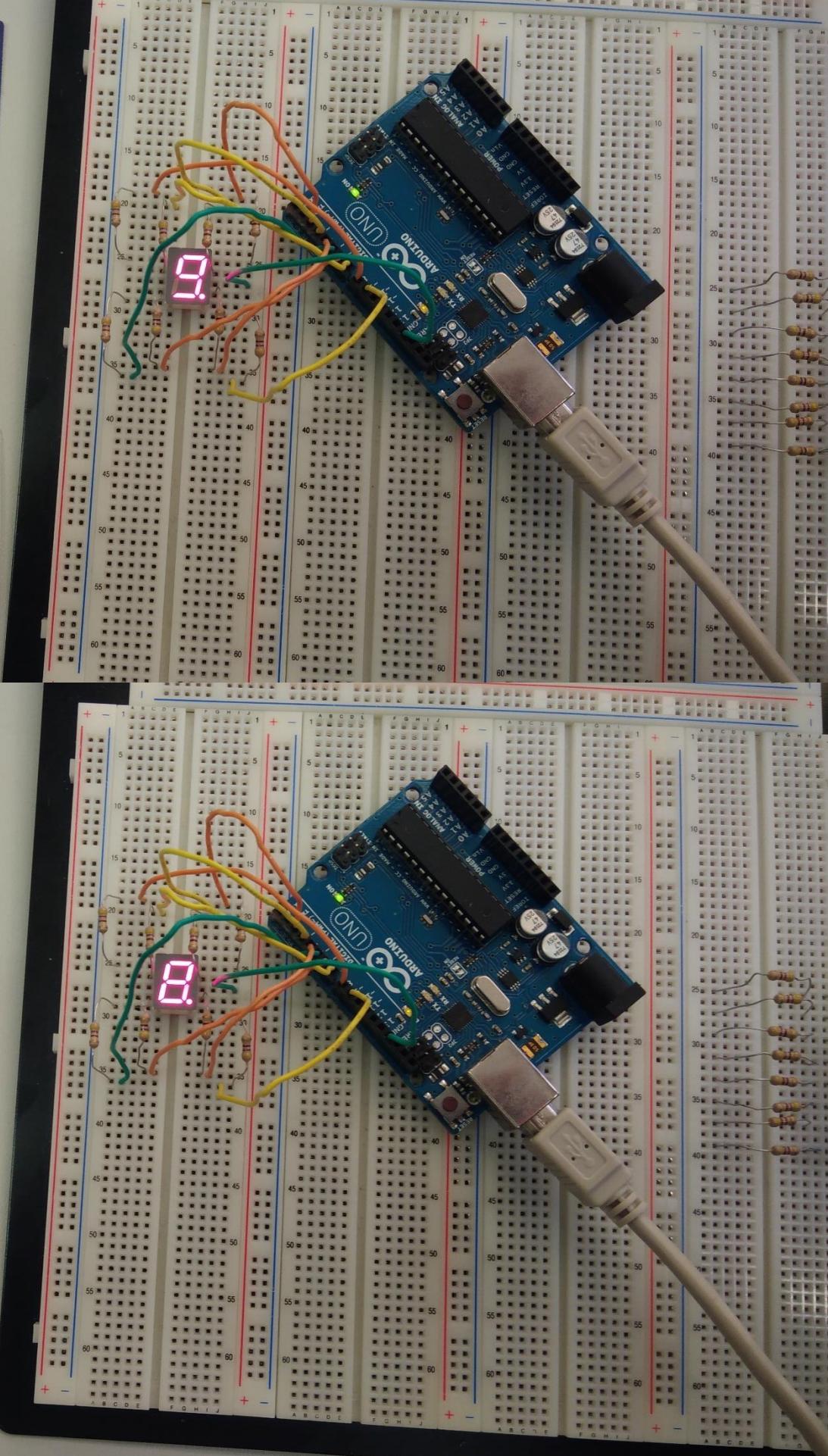


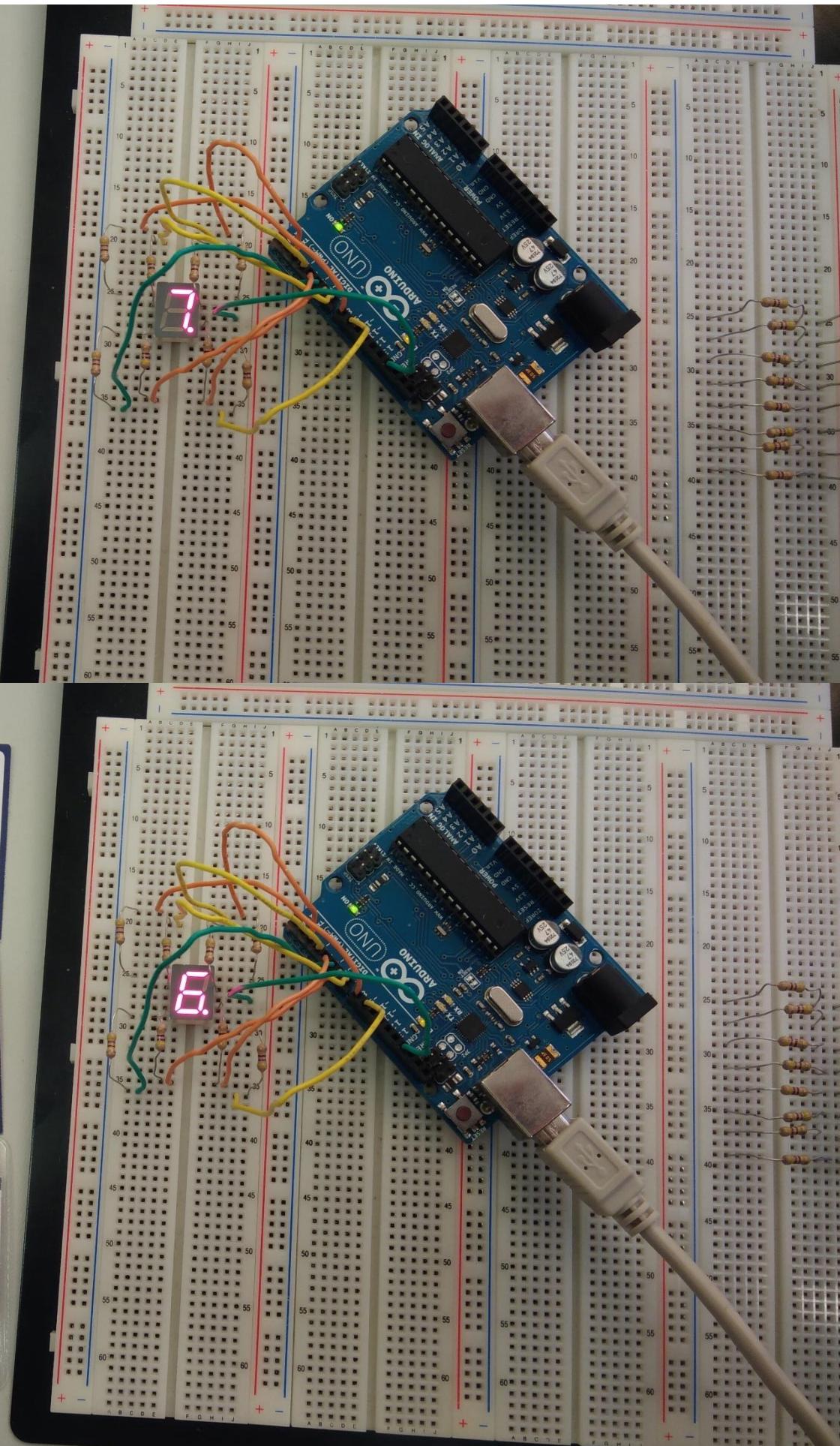
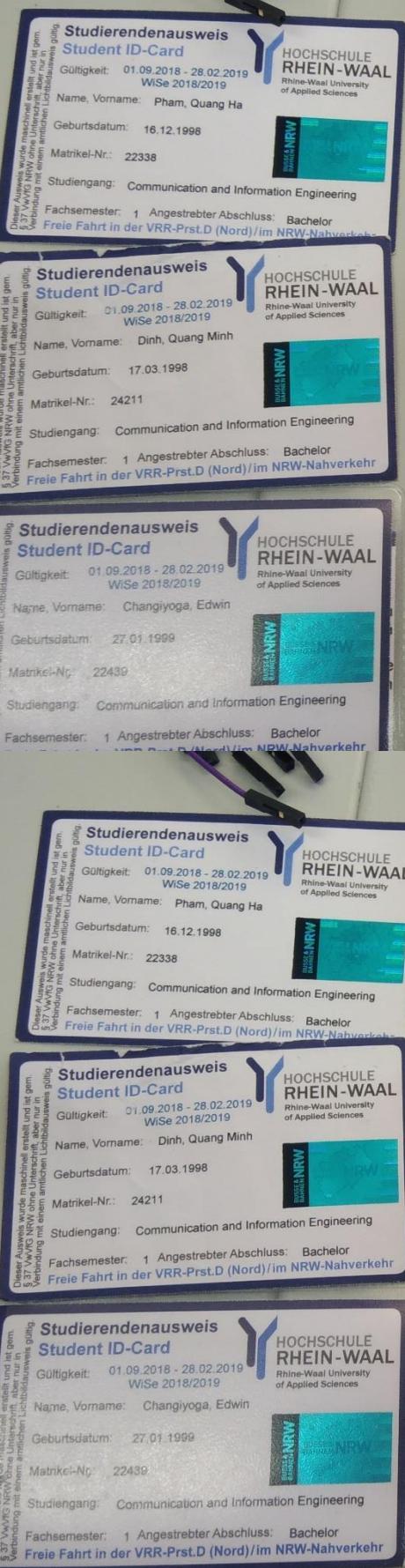


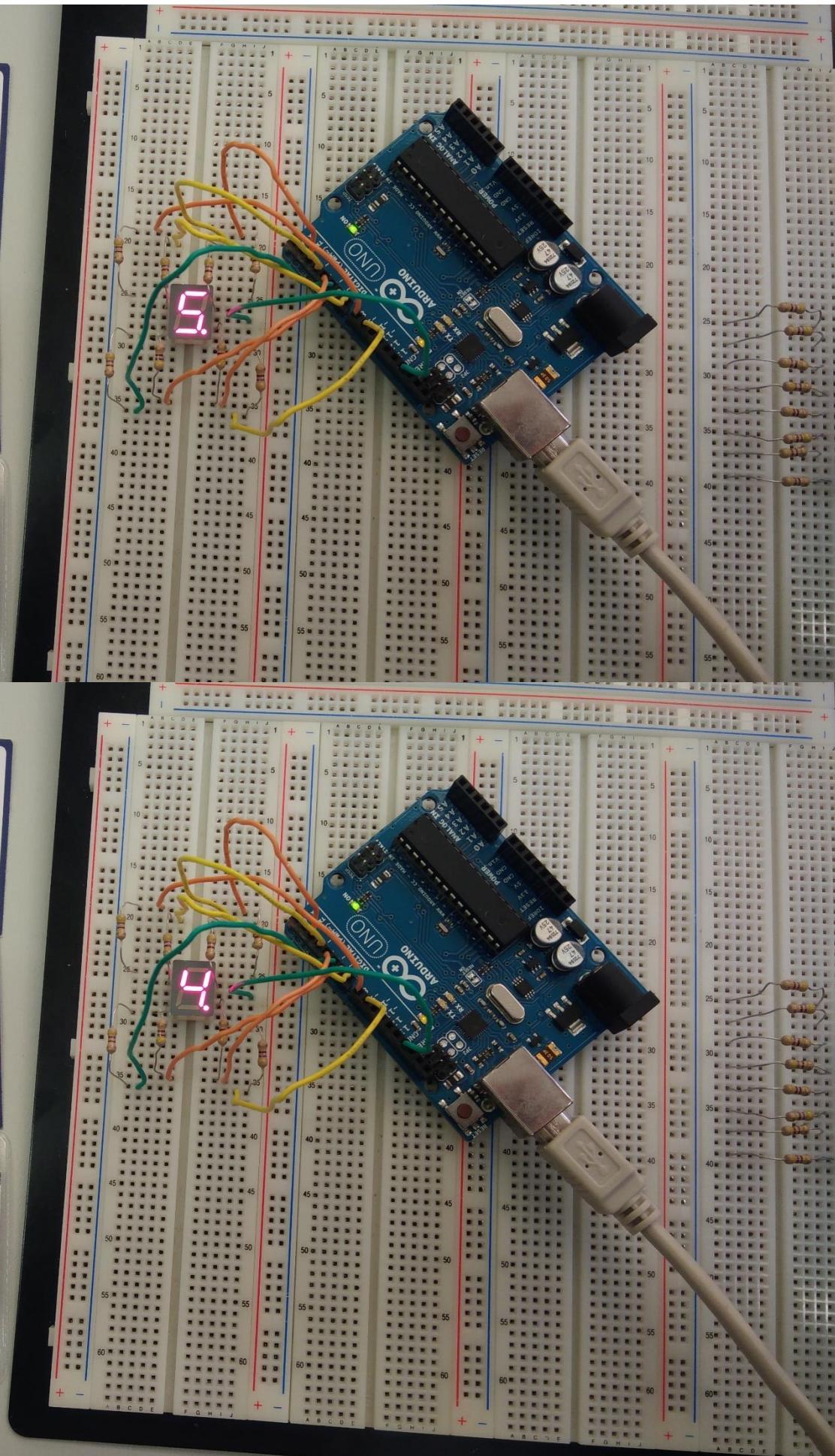


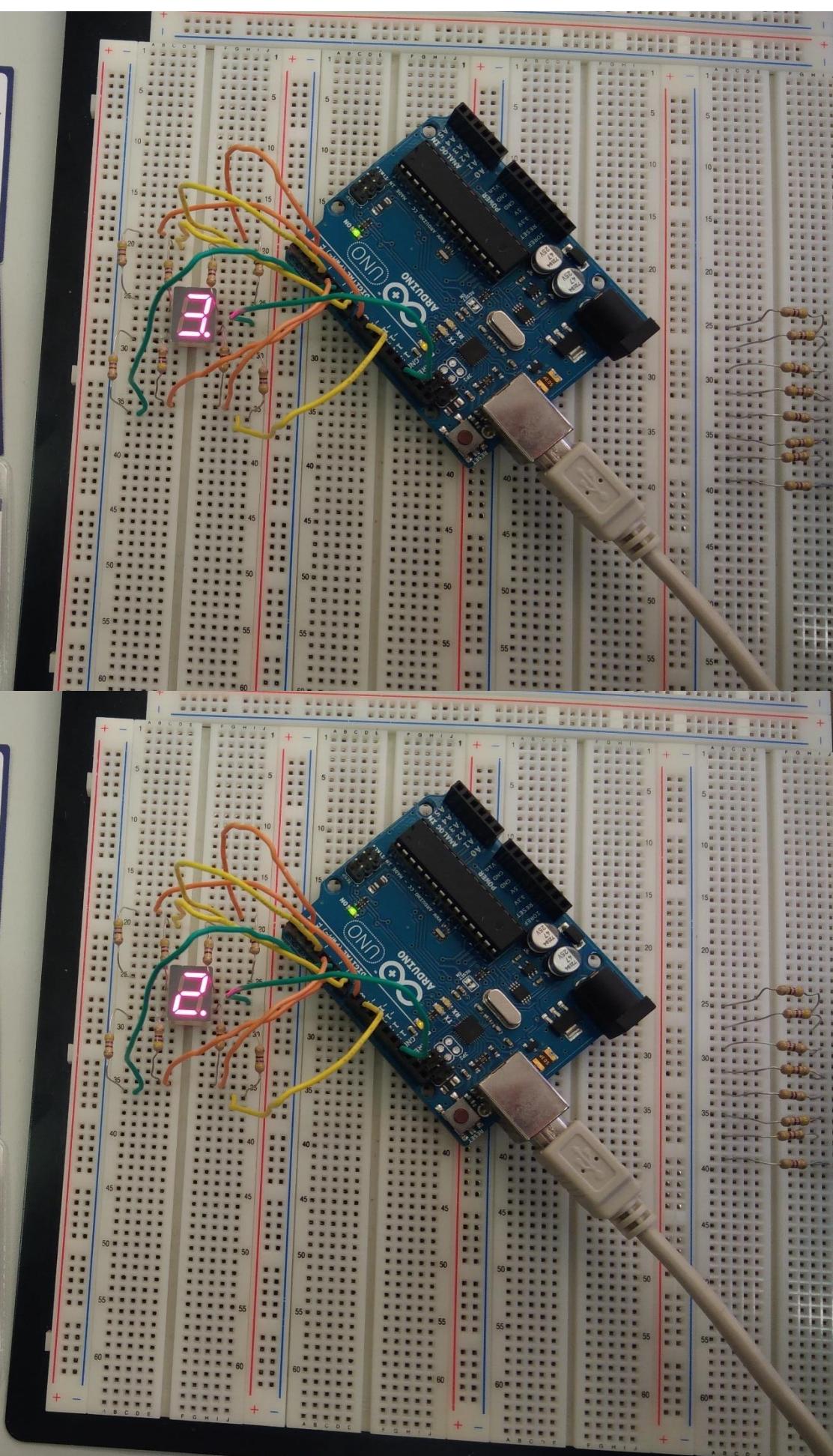


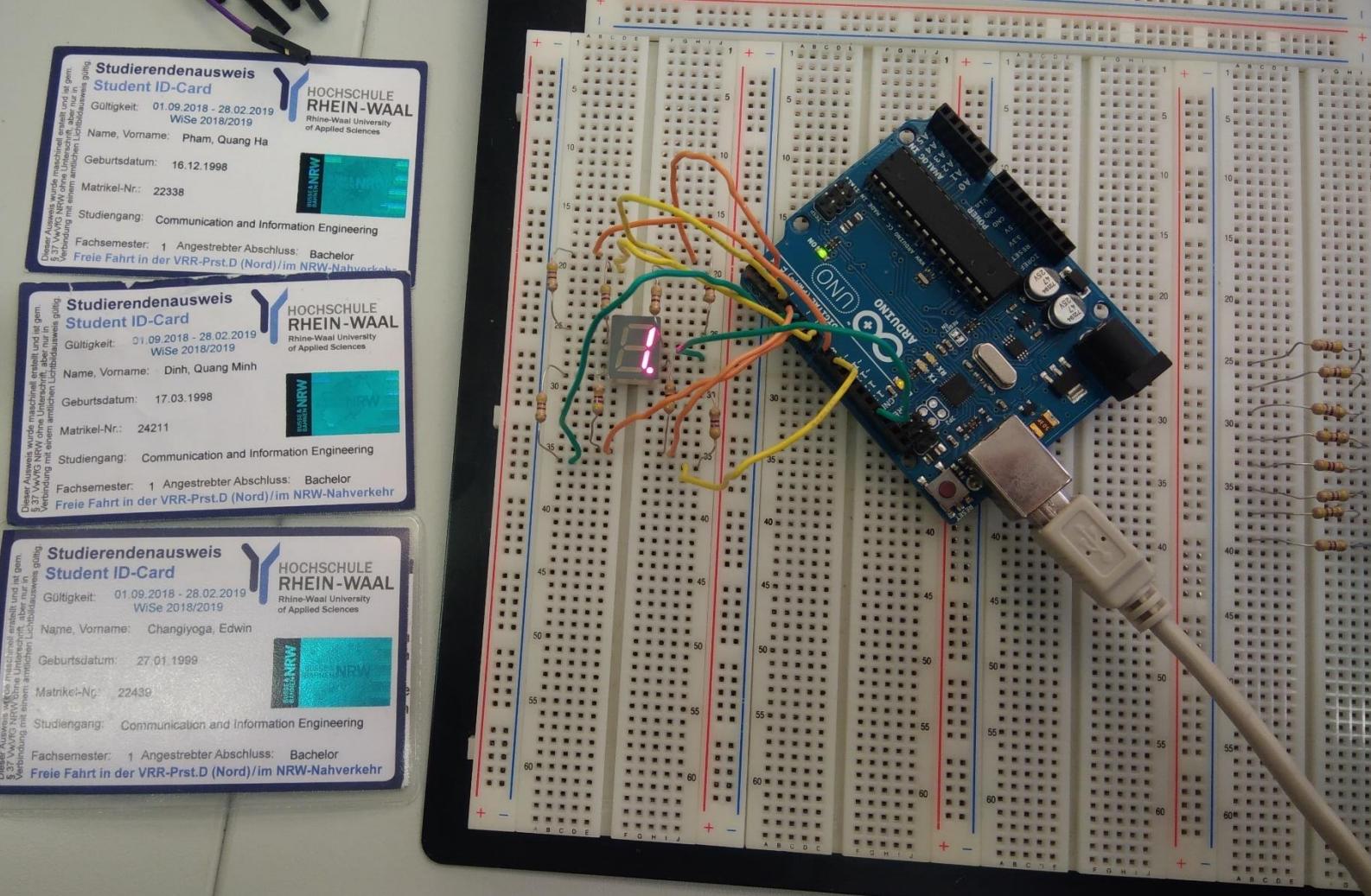












Challenge #7

Abstract:

Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

Each time we press the button before de-bouncing the switch, the display counter goes up a random amount with each press. After de-bouncing the switch, the display counter only goes up by 1 unit as normal.

Our code used:

```
int LED_PIN_2 = 2;  
int LED_PIN_3 = 3;  
int LED_PIN_4 = 4;  
int LED_PIN_5 = 5;  
int LED_PIN_6 = 6;  
int LED_PIN_7 = 7;  
int LED_PIN_8 = 8;  
int LED_PIN_9 = 9;  
int Switch = 10;  
int StateOfSwitch = 0;  
int LastStateOfSwitch = 0;  
int counter = 0;  
  
void setup ()  
{  
  
pinMode(LED_PIN_2, OUTPUT);  
pinMode(LED_PIN_3, OUTPUT);  
pinMode(LED_PIN_4, OUTPUT);  
pinMode(LED_PIN_5, OUTPUT);  
pinMode(LED_PIN_6, OUTPUT);  
pinMode(LED_PIN_7, OUTPUT);  
pinMode(LED_PIN_8, OUTPUT);  
pinMode(LED_PIN_9, OUTPUT);  
pinMode(Switch, INPUT);
```

```

}

void loop ()
{
    StateOfSwitch = digitalRead(Switch);

    if (StateOfSwitch != LastStateOfSwitch)
        // Procedure to check if button is pressed (State NOT EQUAL to LastState).

        {
            delay (20); // De-bounce algorithm.

            if (StateOfSwitch == LOW) // Checking if button is pressed.

                {} // Nothing happens

            else

                {
                    counter++;

                }

        // Make "LastState = State" to disable the button right after it's pressed. Do this so that
        // the numbers don't go up when HOLDING down the button.

        LastStateOfSwitch = StateOfSwitch;

        // Last State & State MUST be in this order (LastState go first because LastState is
        // applying status 0 (inactive) to State).

    }

    switch (counter)

    {
        case 0:

            digitalWrite(LED_PIN_2, HIGH);
            digitalWrite(LED_PIN_3, HIGH);
            digitalWrite(LED_PIN_4, HIGH);
            digitalWrite(LED_PIN_5, HIGH);
            digitalWrite(LED_PIN_6, HIGH);
            digitalWrite(LED_PIN_7, HIGH);
            digitalWrite(LED_PIN_8, LOW);
            digitalWrite(LED_PIN_9, HIGH);

            delay(50);

            break;
    }
}

```

```
case 1:  
    digitalWrite(LED_PIN_2, LOW);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, LOW);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, LOW);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
break;  
  
case 2:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, LOW);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
break;  
  
case 3:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, LOW);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
break;
```

```
case 4:  
    digitalWrite(LED_PIN_2, LOW);  
    digitalWrite(LED_PIN_3, HIGH);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, LOW);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
    break;  
  
case 5:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, LOW);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, LOW);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
    break;  
  
case 6:  
    digitalWrite(LED_PIN_2, HIGH);  
    digitalWrite(LED_PIN_3, LOW);  
    digitalWrite(LED_PIN_4, HIGH);  
    digitalWrite(LED_PIN_5, HIGH);  
    digitalWrite(LED_PIN_6, HIGH);  
    digitalWrite(LED_PIN_7, HIGH);  
    digitalWrite(LED_PIN_8, HIGH);  
    digitalWrite(LED_PIN_9, HIGH);  
    delay(50);  
    break;
```

```
case 7:
```

```
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, HIGH);
    digitalWrite(LED_PIN_4, HIGH);
    digitalWrite(LED_PIN_5, LOW);
    digitalWrite(LED_PIN_6, LOW);
    digitalWrite(LED_PIN_7, LOW);
    digitalWrite(LED_PIN_8, LOW);
    digitalWrite(LED_PIN_9, HIGH);

    delay(50);
```

```
break;
```

```
case 8:
```

```
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, HIGH);
    digitalWrite(LED_PIN_4, HIGH);
    digitalWrite(LED_PIN_5, HIGH);
    digitalWrite(LED_PIN_6, HIGH);
    digitalWrite(LED_PIN_7, HIGH);
    digitalWrite(LED_PIN_8, HIGH);
    digitalWrite(LED_PIN_9, HIGH);

    delay(50);
```

```
break;
```

```
case 9:
```

```
    digitalWrite(LED_PIN_2, HIGH);
    digitalWrite(LED_PIN_3, HIGH);
    digitalWrite(LED_PIN_4, HIGH);
    digitalWrite(LED_PIN_5, HIGH);
    digitalWrite(LED_PIN_6, LOW);
    digitalWrite(LED_PIN_7, HIGH);
    digitalWrite(LED_PIN_8, HIGH);
    digitalWrite(LED_PIN_9, HIGH);

    delay(50);
```

```
break;
```

```

default:

counter=0;

/// When counter is not 0-9, then counter goes back to 0

break;

}

}

```

Pictures:

