

Protocol Laboratory Digital Engineering #4

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	15
Challenge #3	25
Challenge #4	37
Challenge #5	47
Challenge #6	59
Challenge #7	67
Challenge #8	77
Challenge #9	86
Challenge #10	115

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.

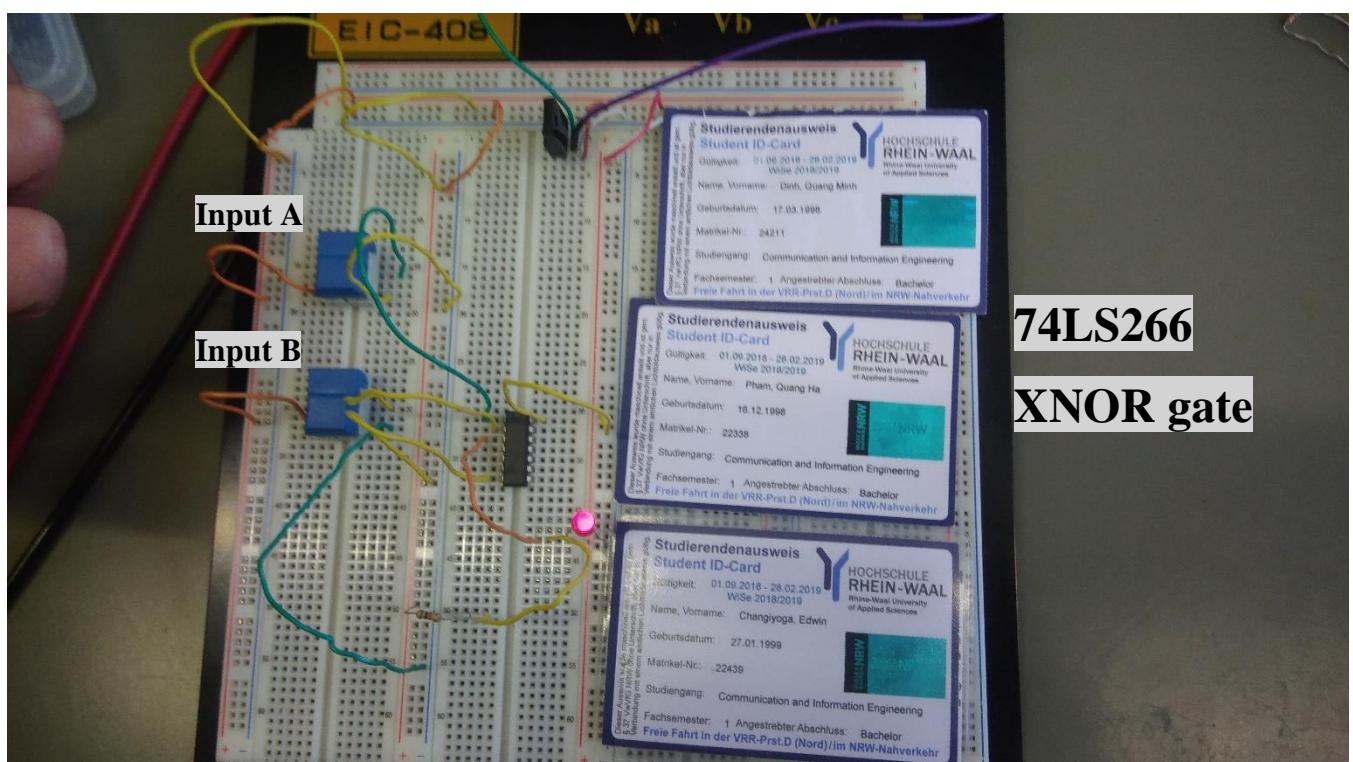
For this challenge we are required to choose 3 out of the 5 given ICs, we have chosen the following 3 ICs: 74LS266N, 74LS279 and 74HC74.

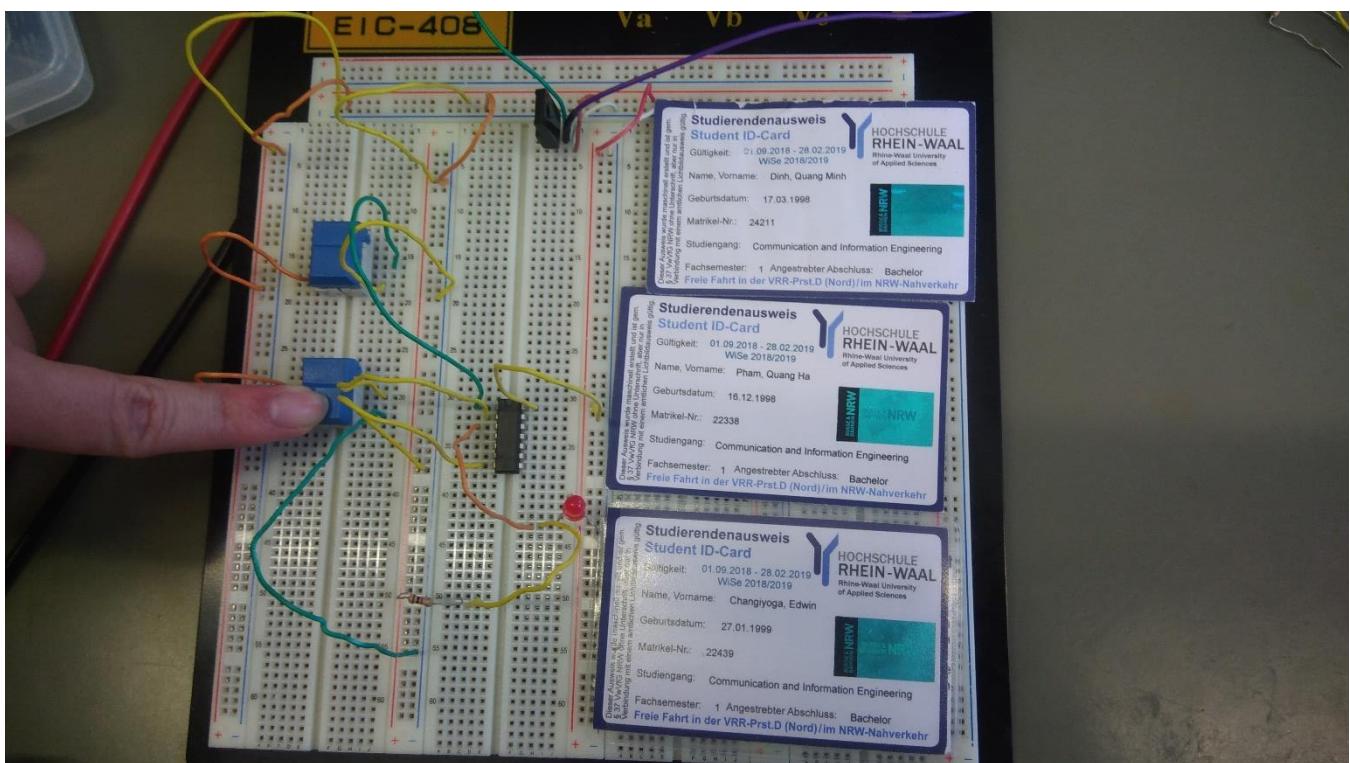
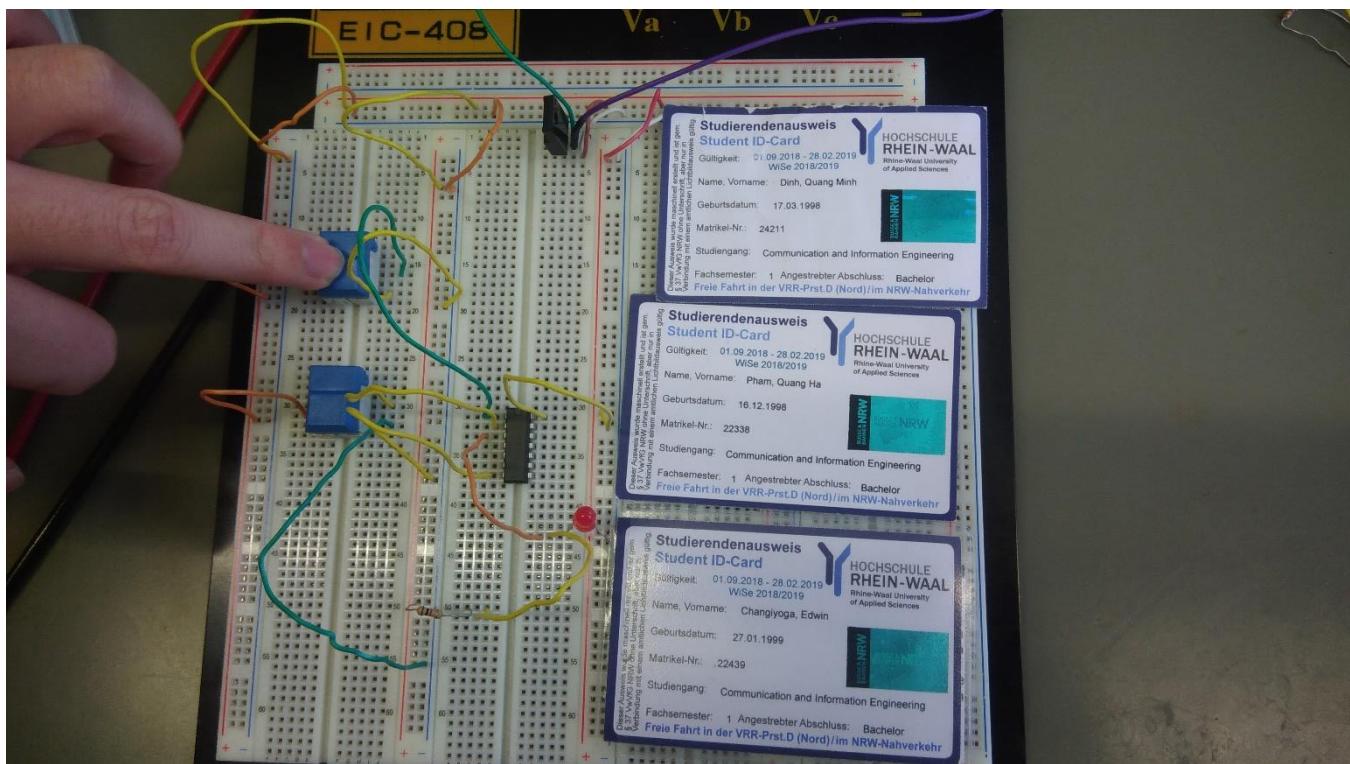
74LS266N is a quadruple 2 input XNOR gate, it is a standard XNOR but with a twist: the output for this IC is open-drain. This means when a connection from the output device for example an LED is made with the IC via the output pin, the LED will be connected to ground instead of Vcc, to us this is quite similar to how a pull-up resistor works. To make the circuit work like a normal XNOR gate, we need to connect both Vcc and the output pin parallel to the LED.

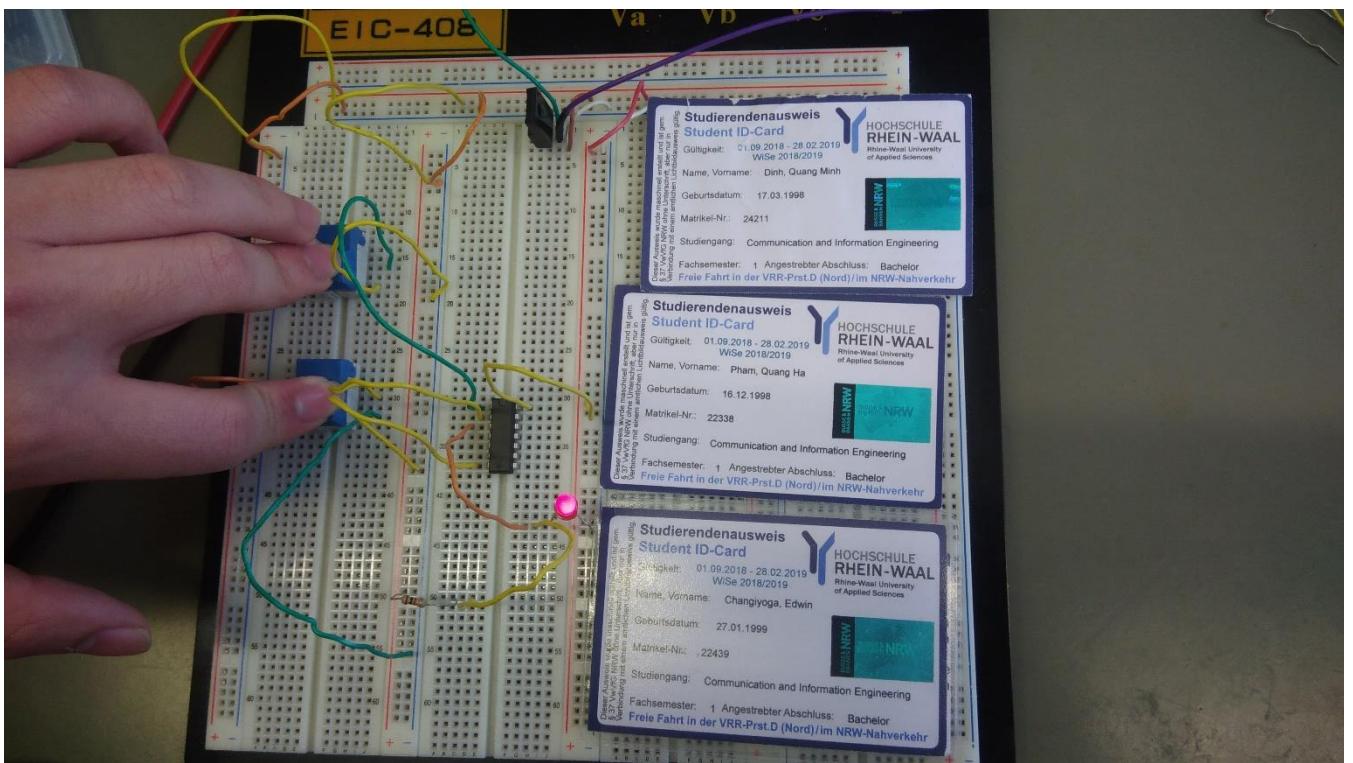
74LS279 is a quadruple S-R Latches, which means there are 4 S-R latches total inside this IC. In those 4 S-R latches includes two types of S-R latches, the first two are 2 regular S-R latches (pin 5-7, pin 13-15). The others are 2 “two-S-input” S-R latches (pin 1-4, pin 9-12), these two S inputs are AND together.

74HC74 is a dual D Flip-flop, which means there are 2 D Flip-flop in this IC. This is the same as challenge #9 in Lab 3 but with individual Set, Reset, and Clock inputs. Set and Reset are also normally closed.

Pictures:



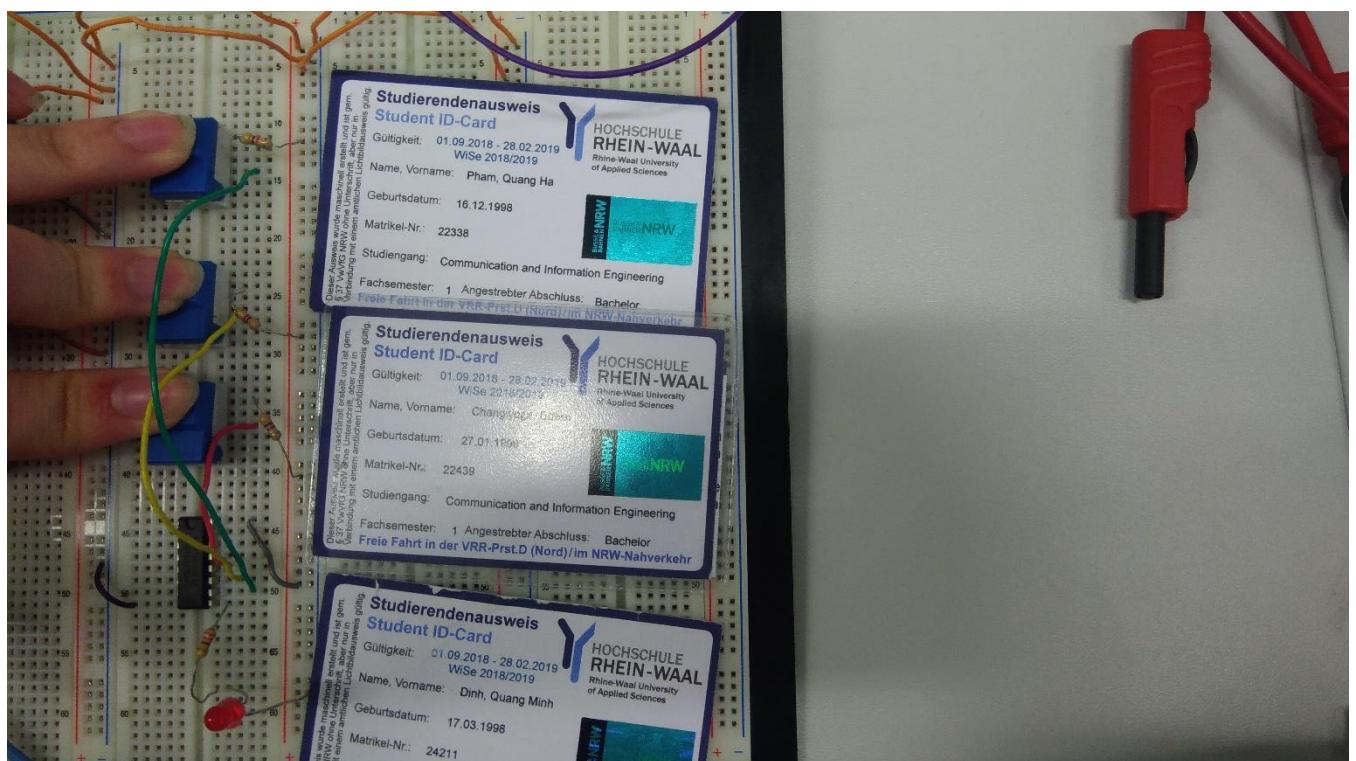
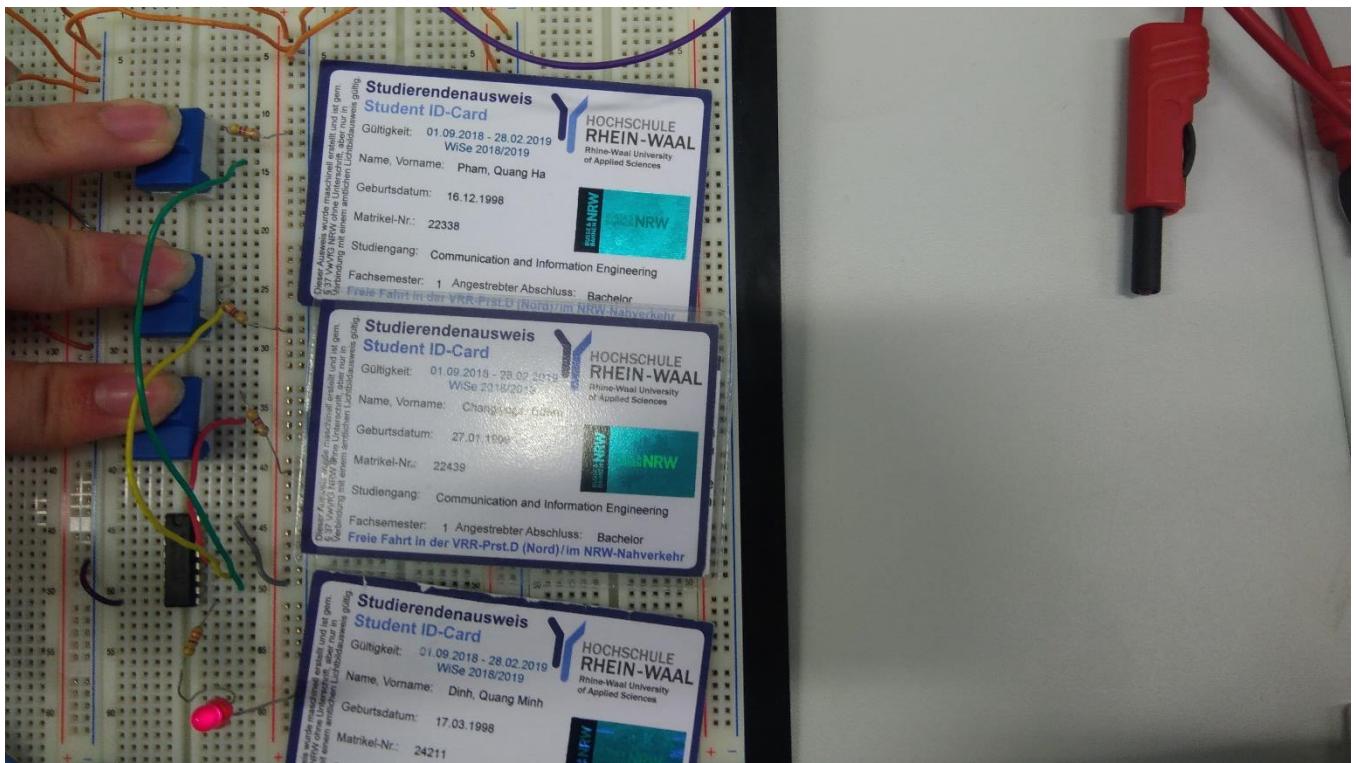


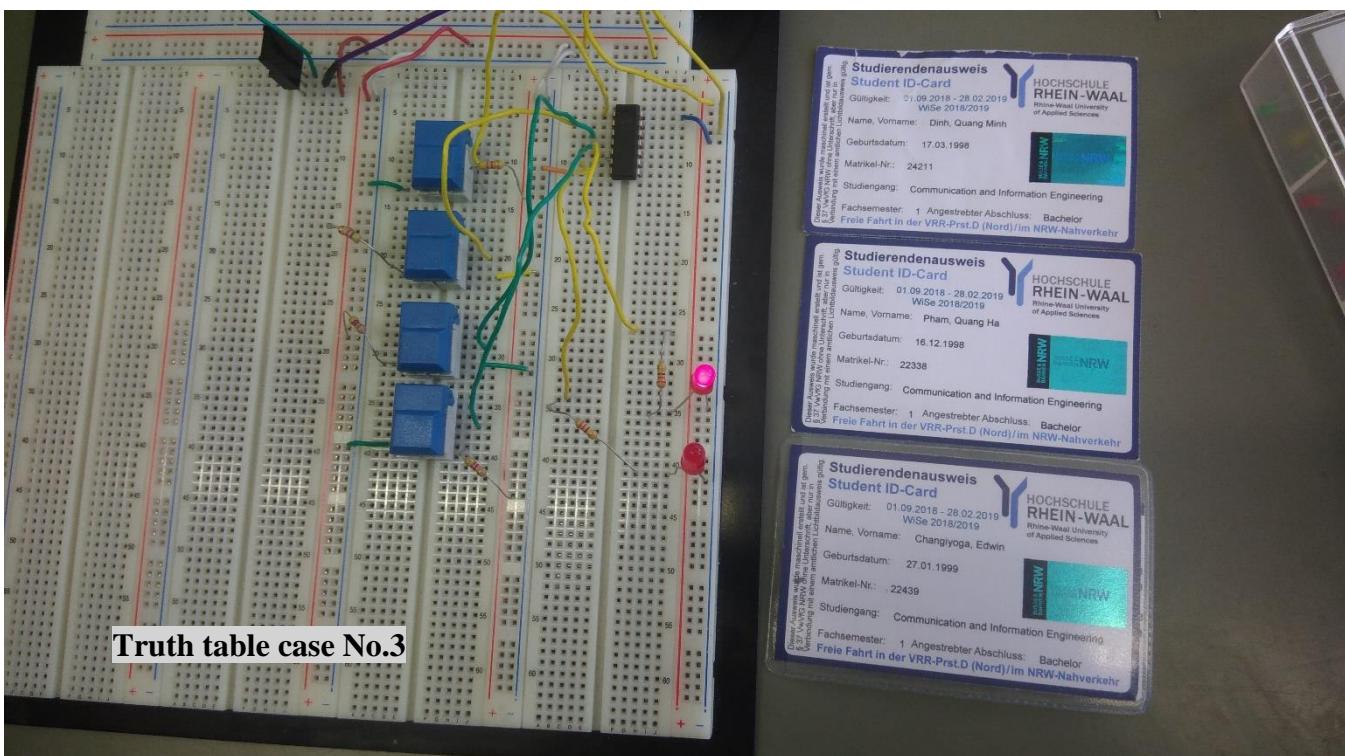
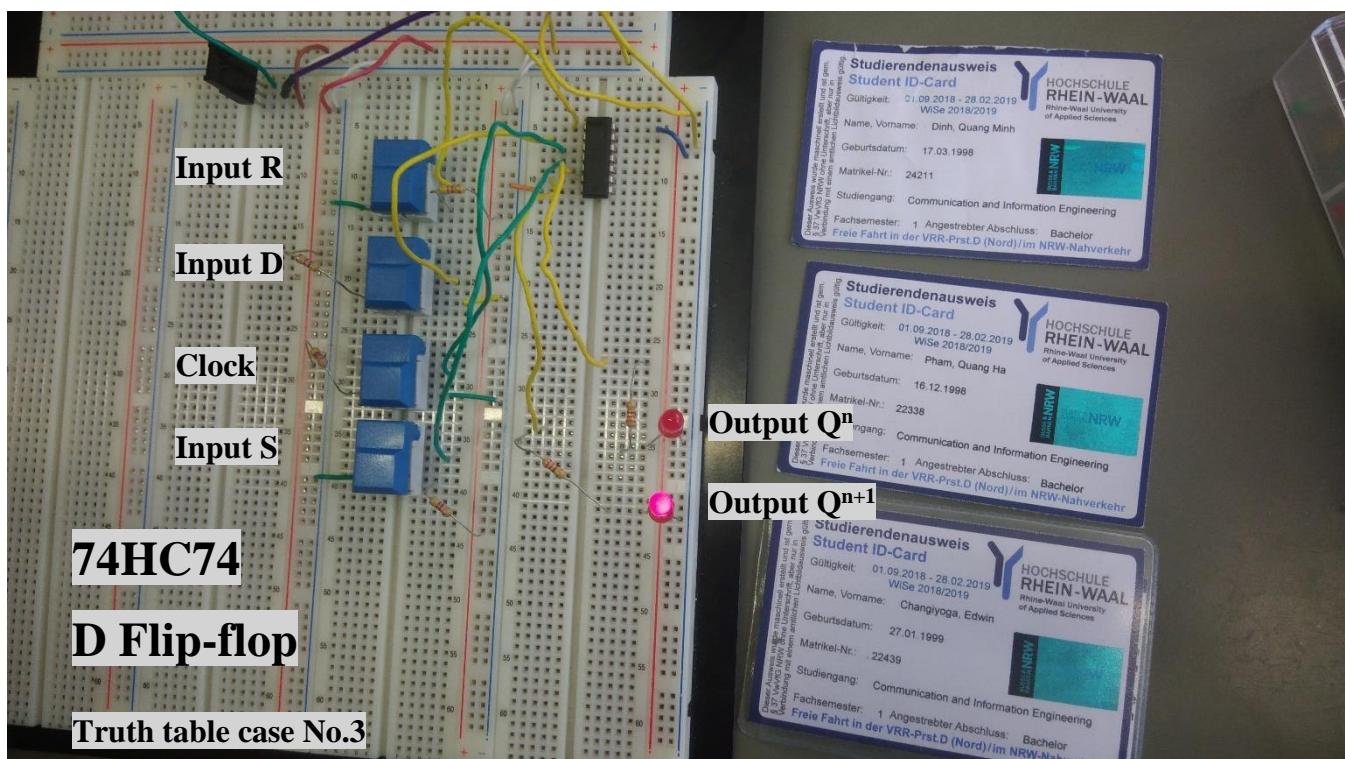


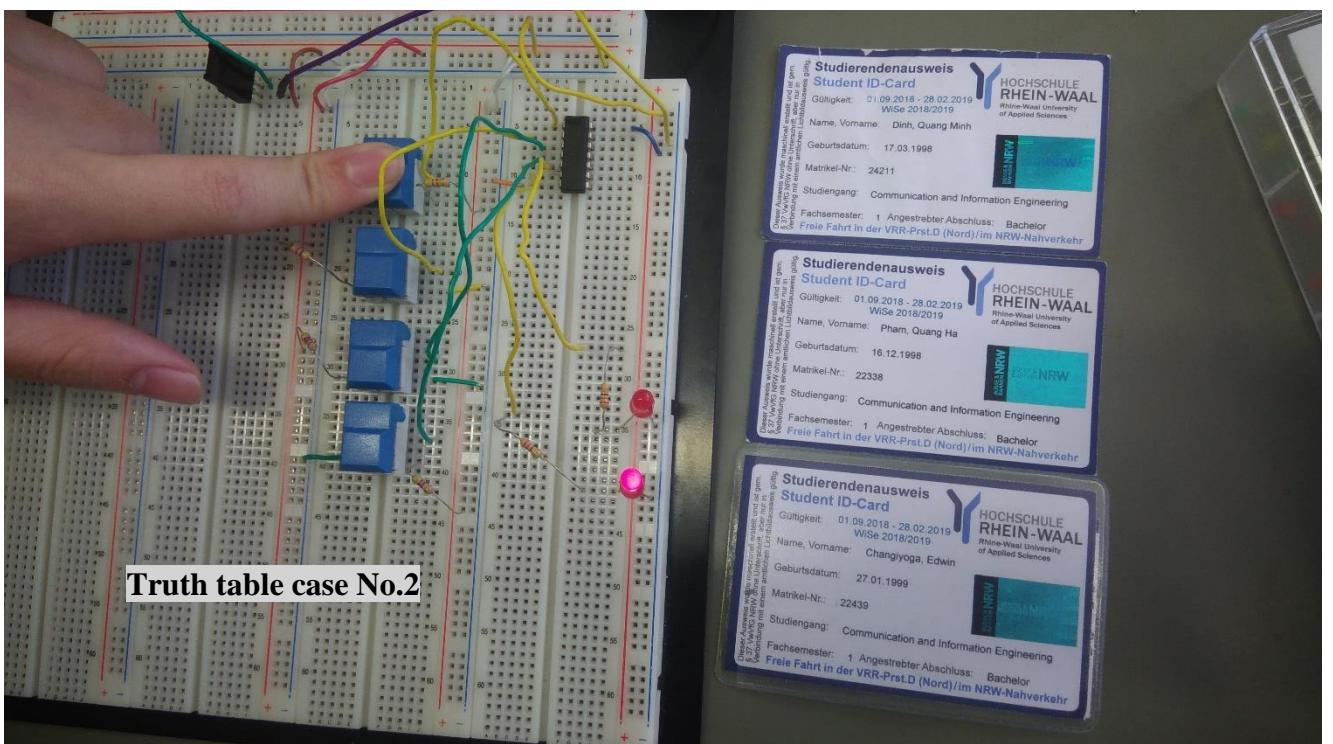
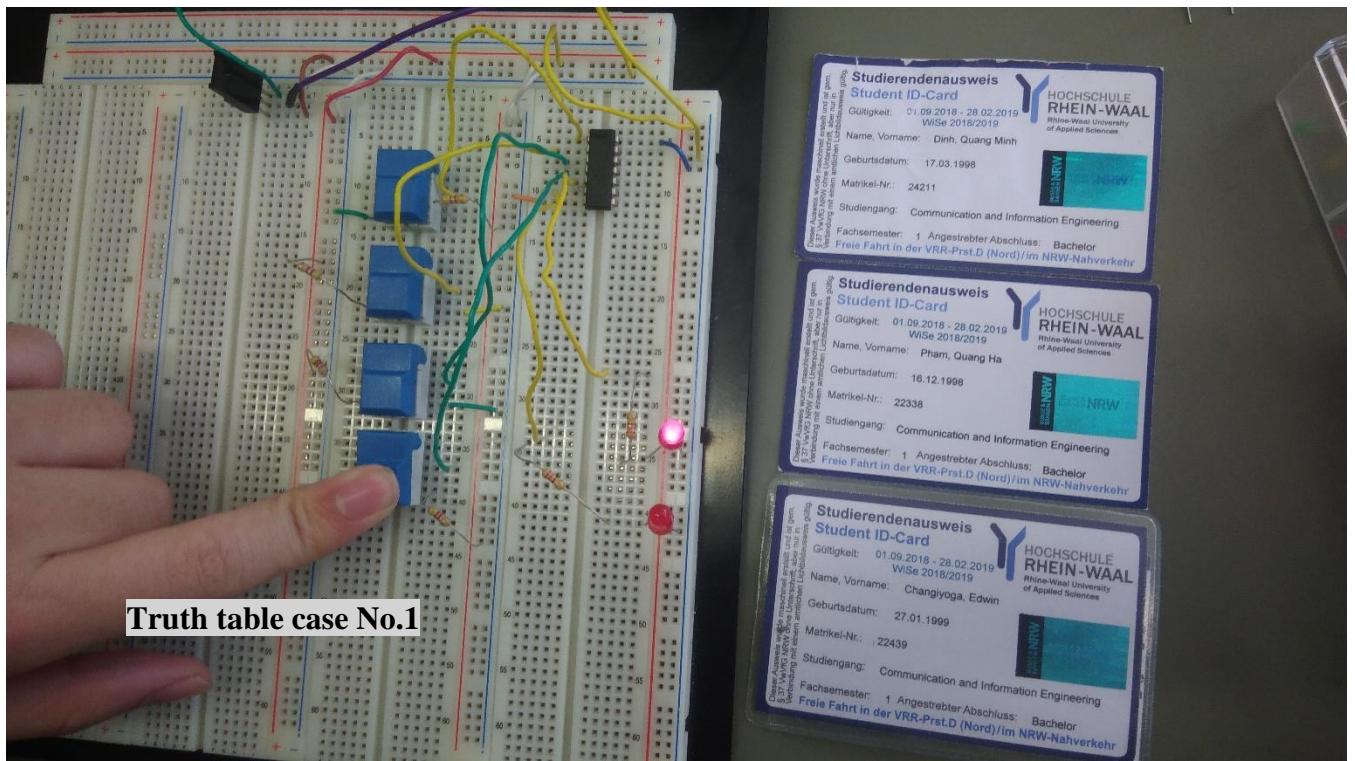


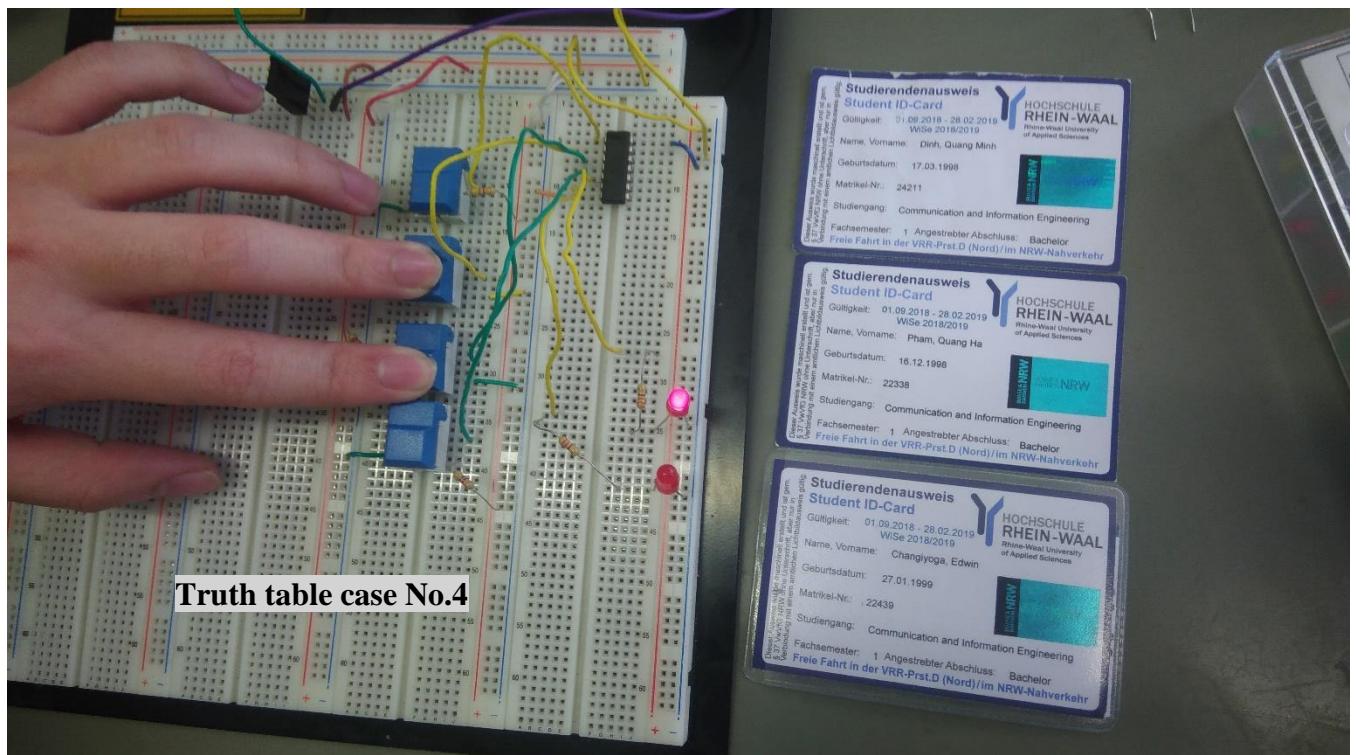
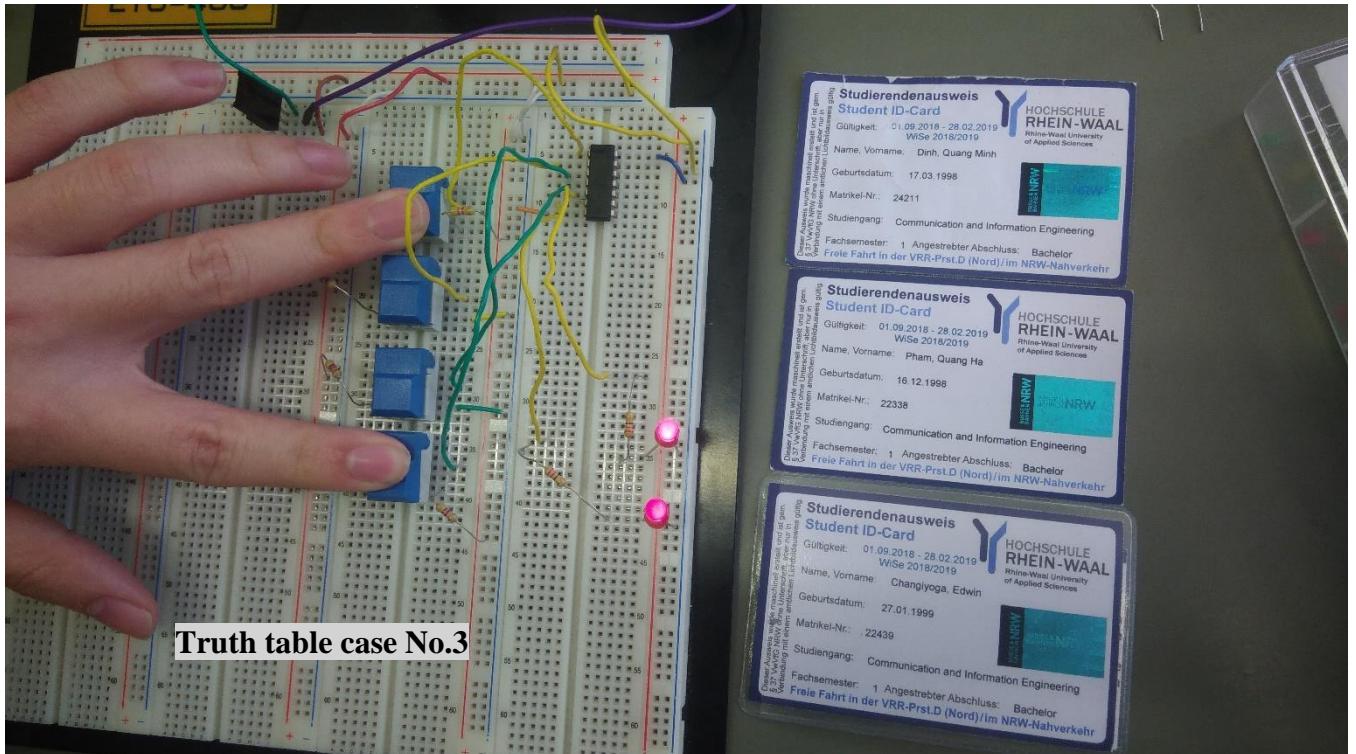


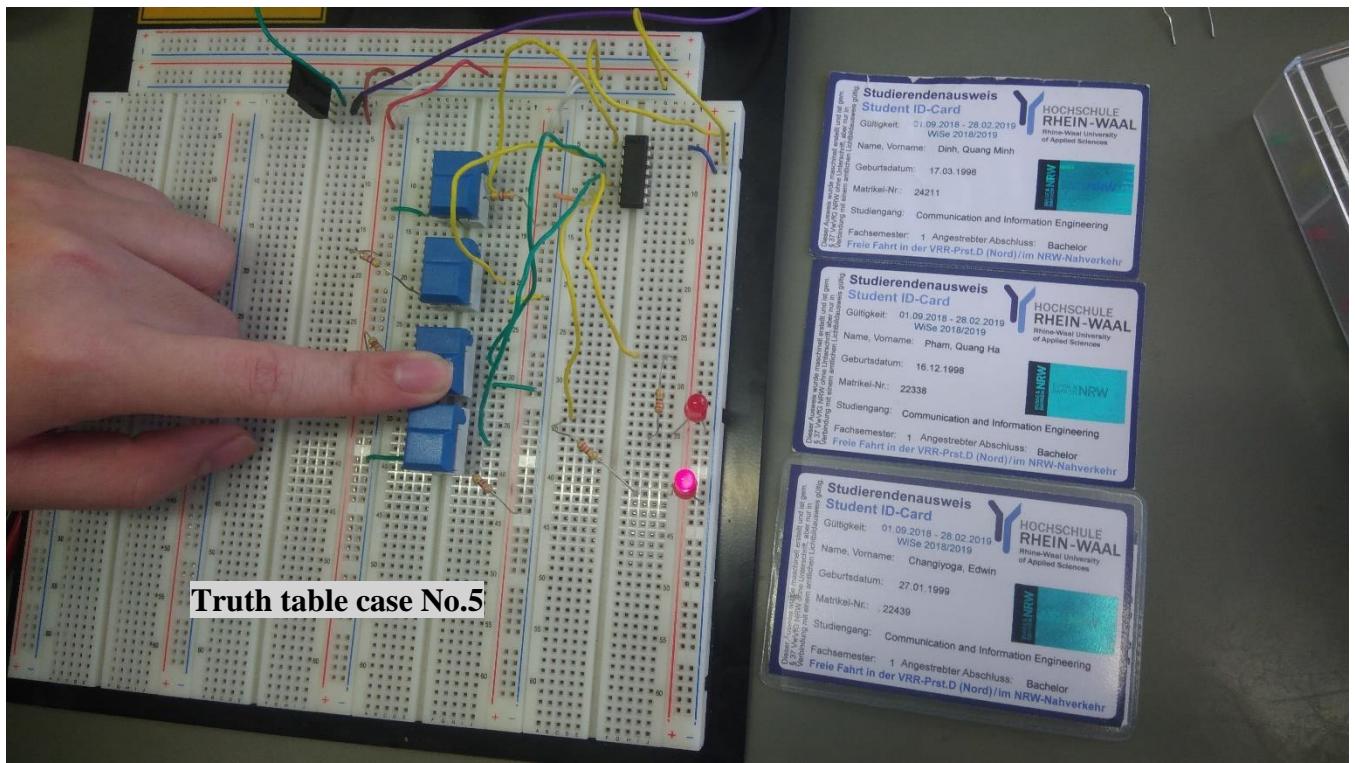












Result:

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Table 1: Truth table for 74LS226

S*	R	Q
0	0	1
1	0	0
0	1	1
1	1	Last state

Table 2: Truth table for 74LS279

S*: with two-S-input: S*=0 when either S1 or S2 = 0, S*=1 when S1=S2=1.

No.	S	Clock	D	R	Q ⁿ	Q ⁿ⁺¹
1	0	X	X	1	1	0
2	1	X	X	0	0	1
3	0	X	X	0	1*	1*
4	1	1	1	1	1	0
5	1	1	0	1	0	1
6	1	0	X	1	Q ⁿ	Q ⁿ⁺¹
7	1	1	X	1	Q ⁿ	Q ⁿ⁺¹
8	1	0	X	1	Q ⁿ	Q ⁿ⁺¹

Table 3: Truth table for 74HC74

*: If S=R=0 then Qⁿ=Qⁿ⁺¹=1. If we try S=R=1 at the same time, the Q results are unpredictable.

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.

After looking at the schematics, we have decided to categorize the connections as follows:

Parallel connection is an OR gate.

Series connection is an AND gate.

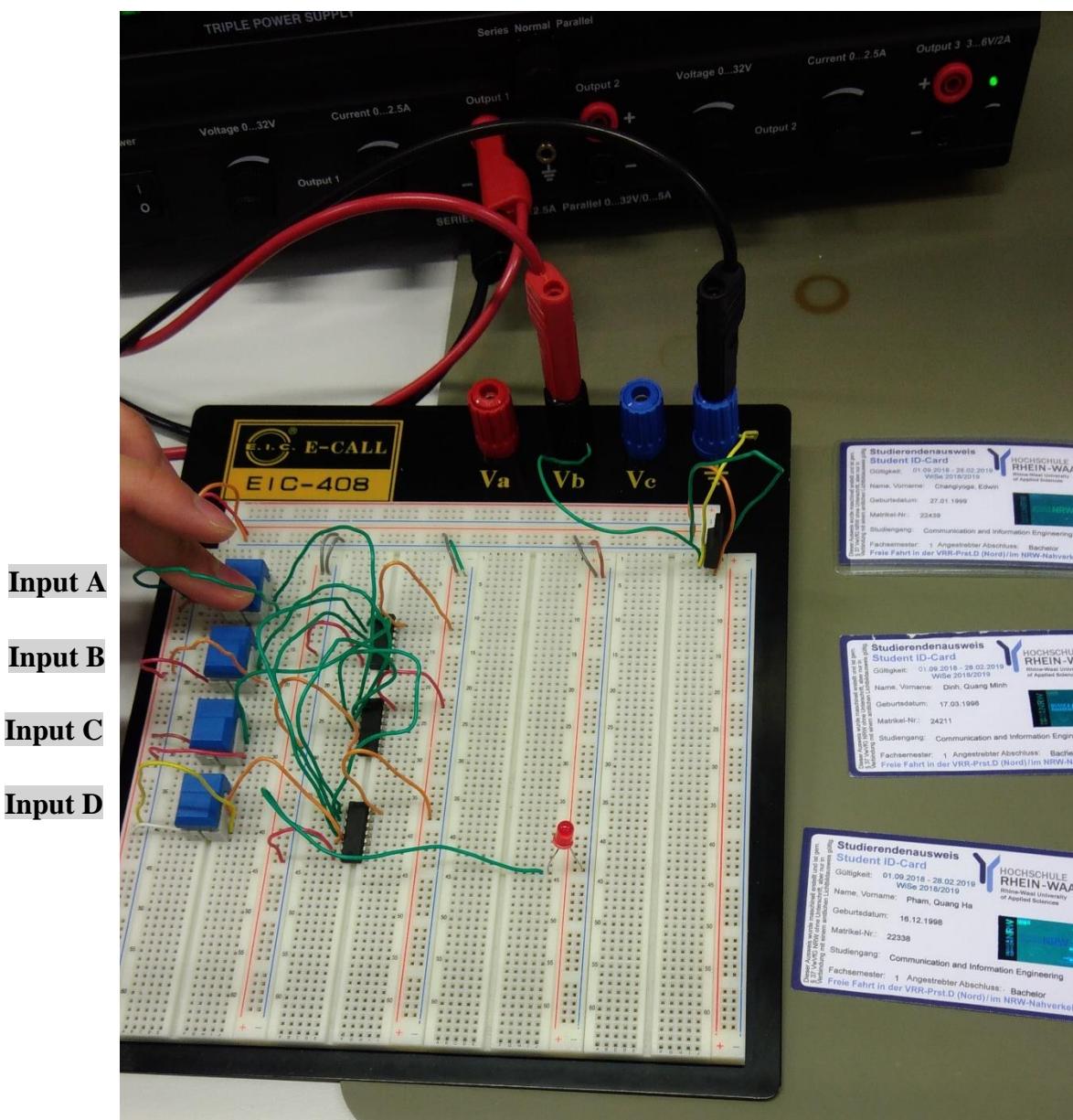
For a Normally open switch: Not pressed is in state 0, Pressed is in state 1.

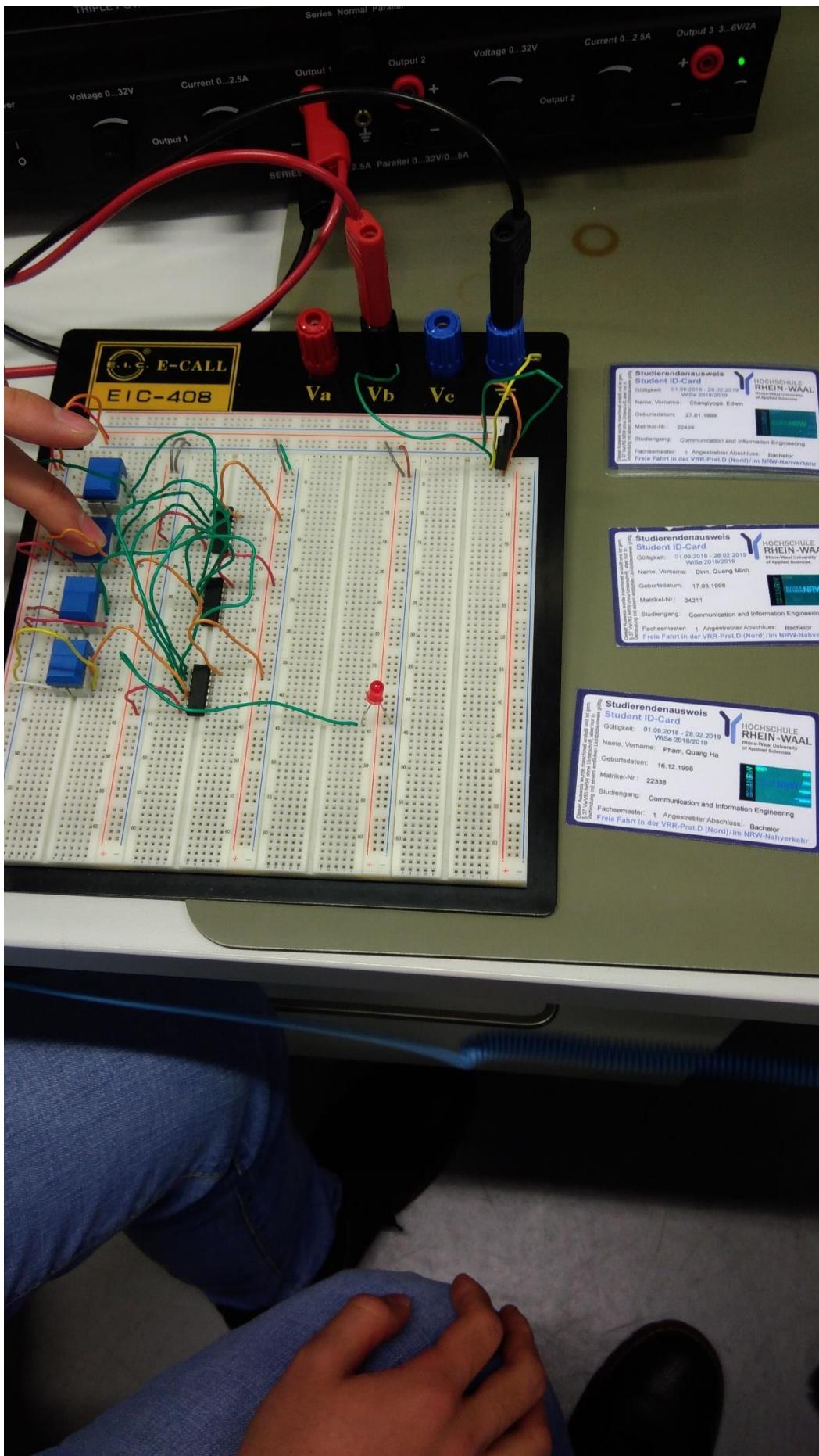
For a Normally closed switch: Not pressed is in state 1, Pressed is in state 0.

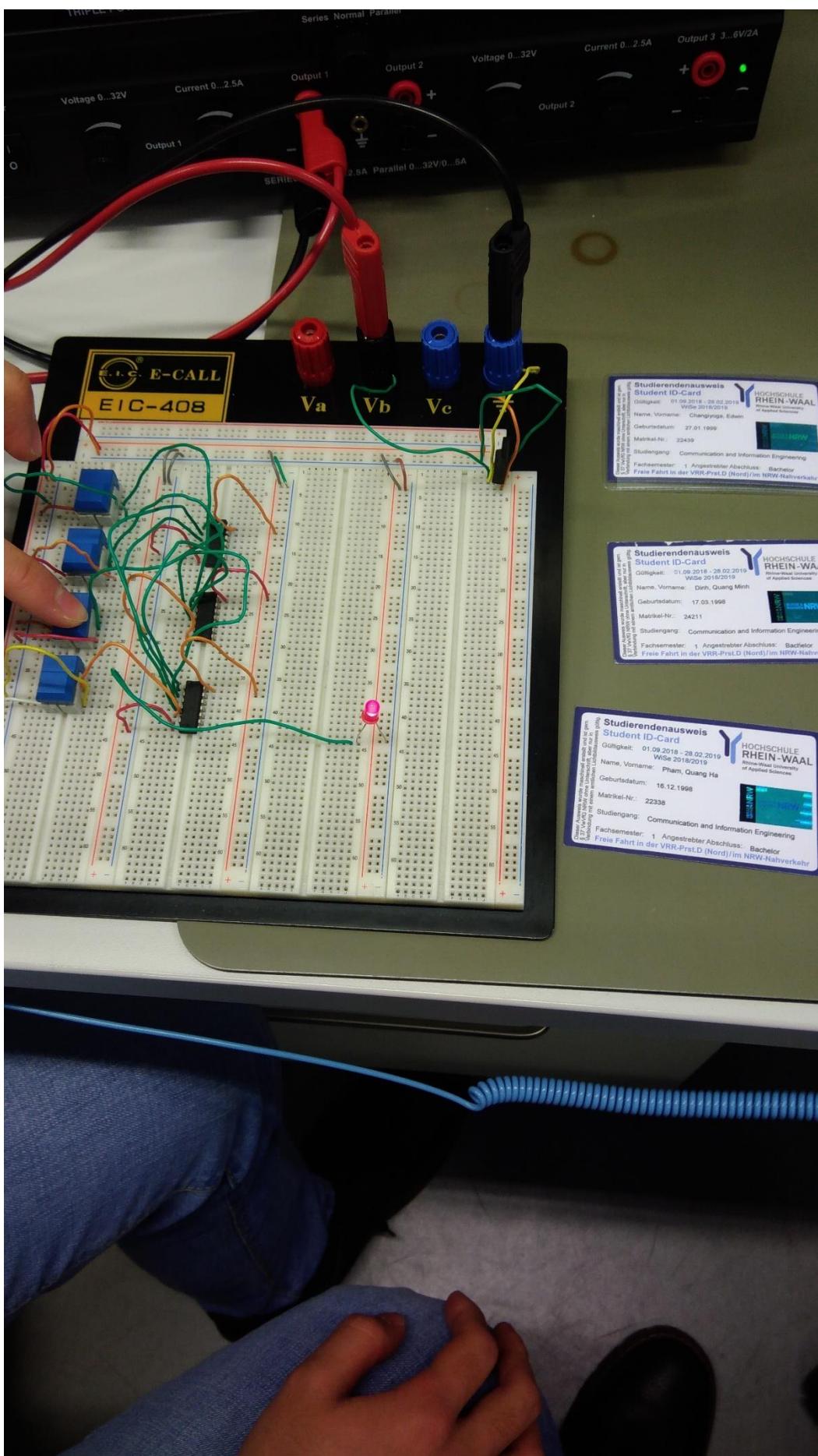
We conclude that a Normally closed switch is an Inverter.

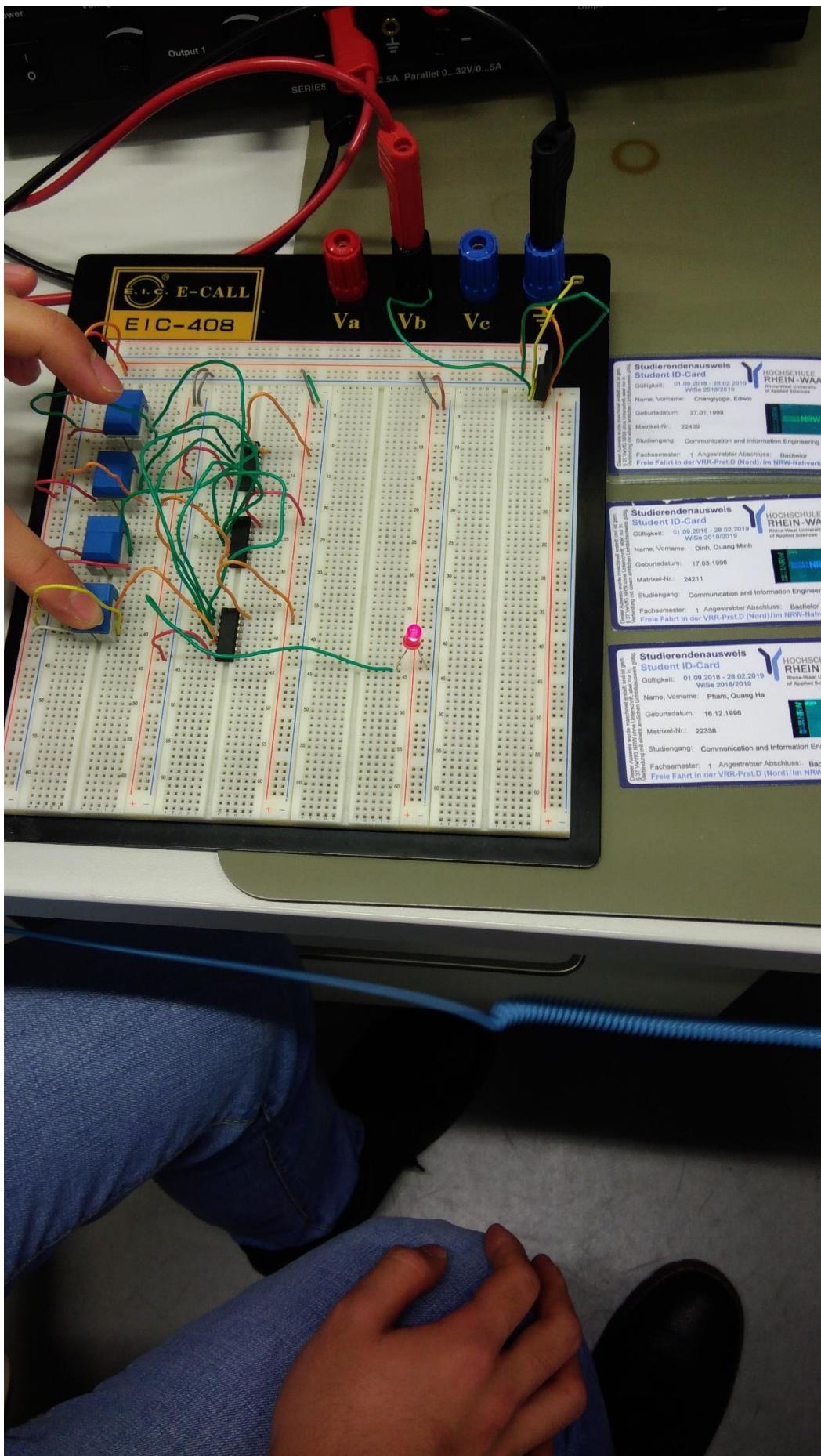
With the connections categorized, we got our final circuit as follows: $(C \wedge (\neg A \vee \neg B)) \vee D$.

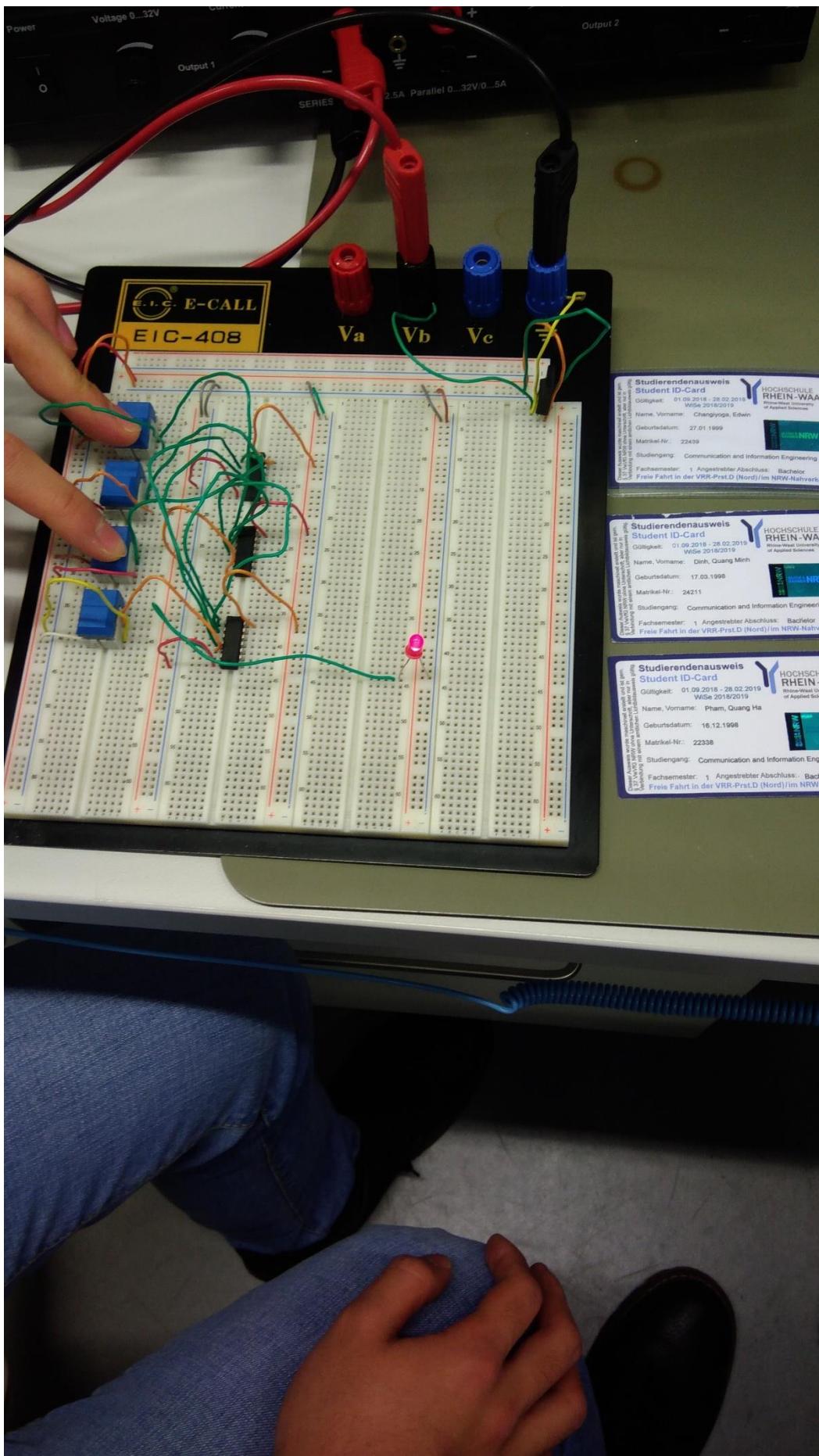
Pictures:

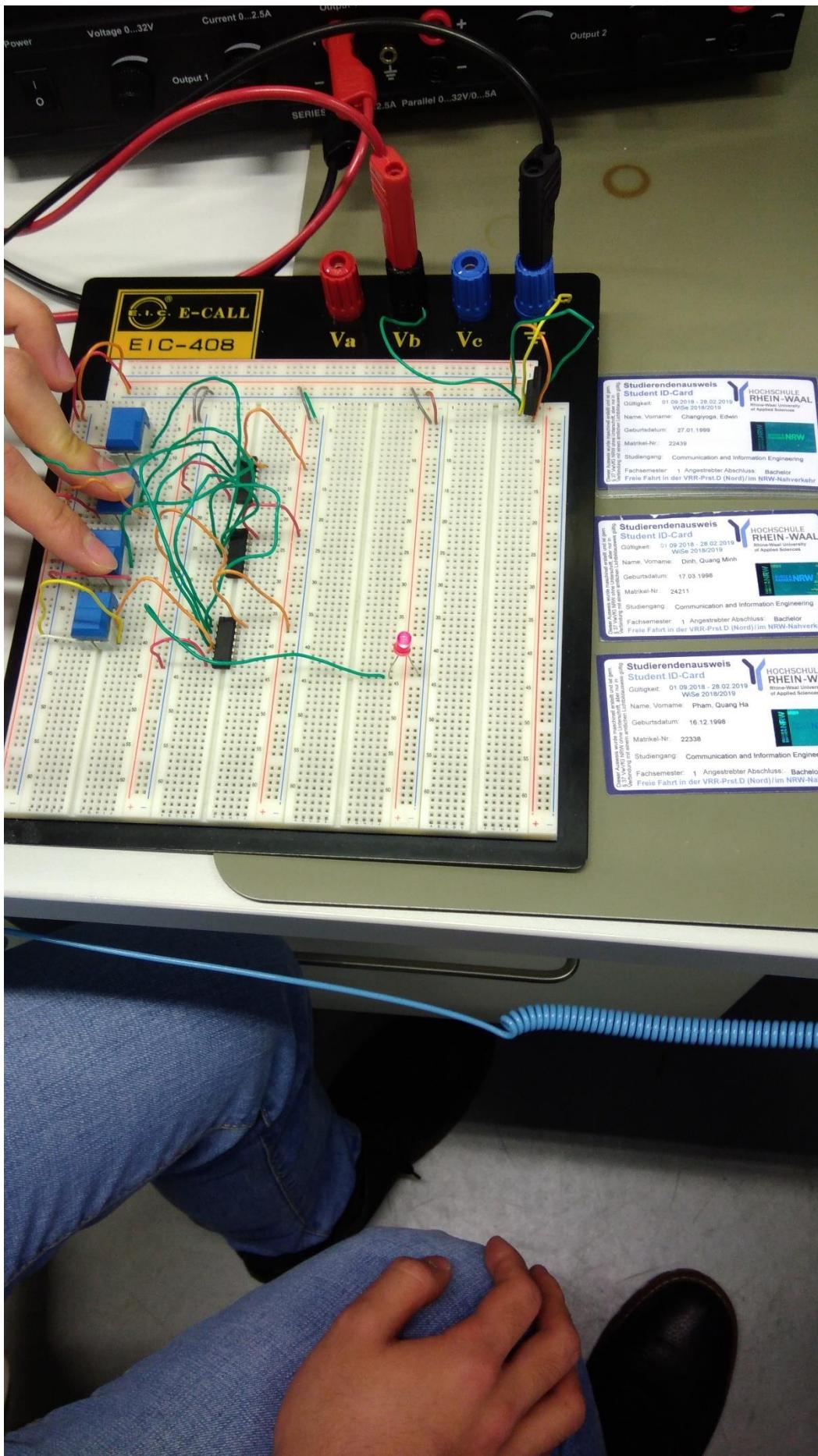


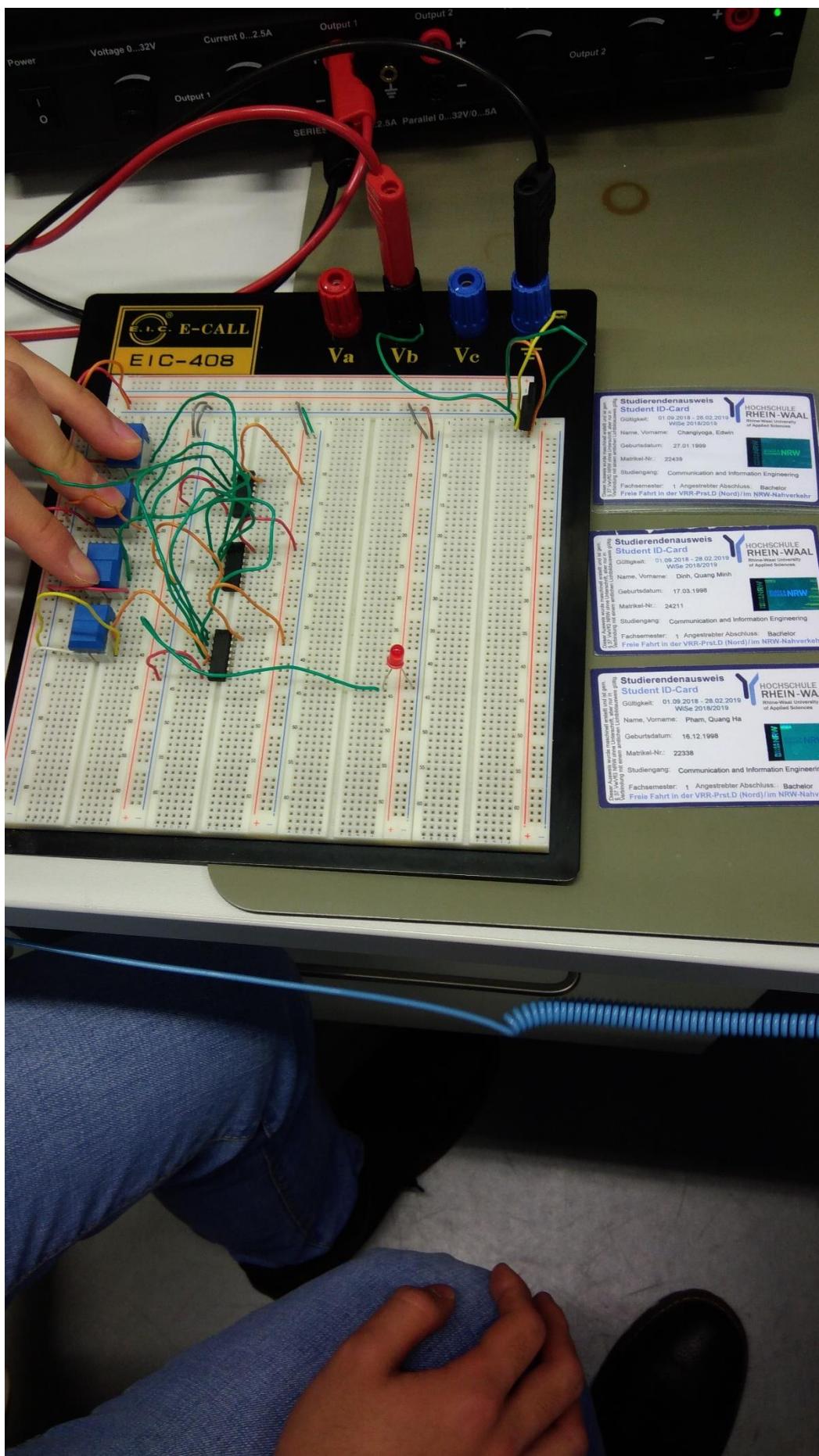


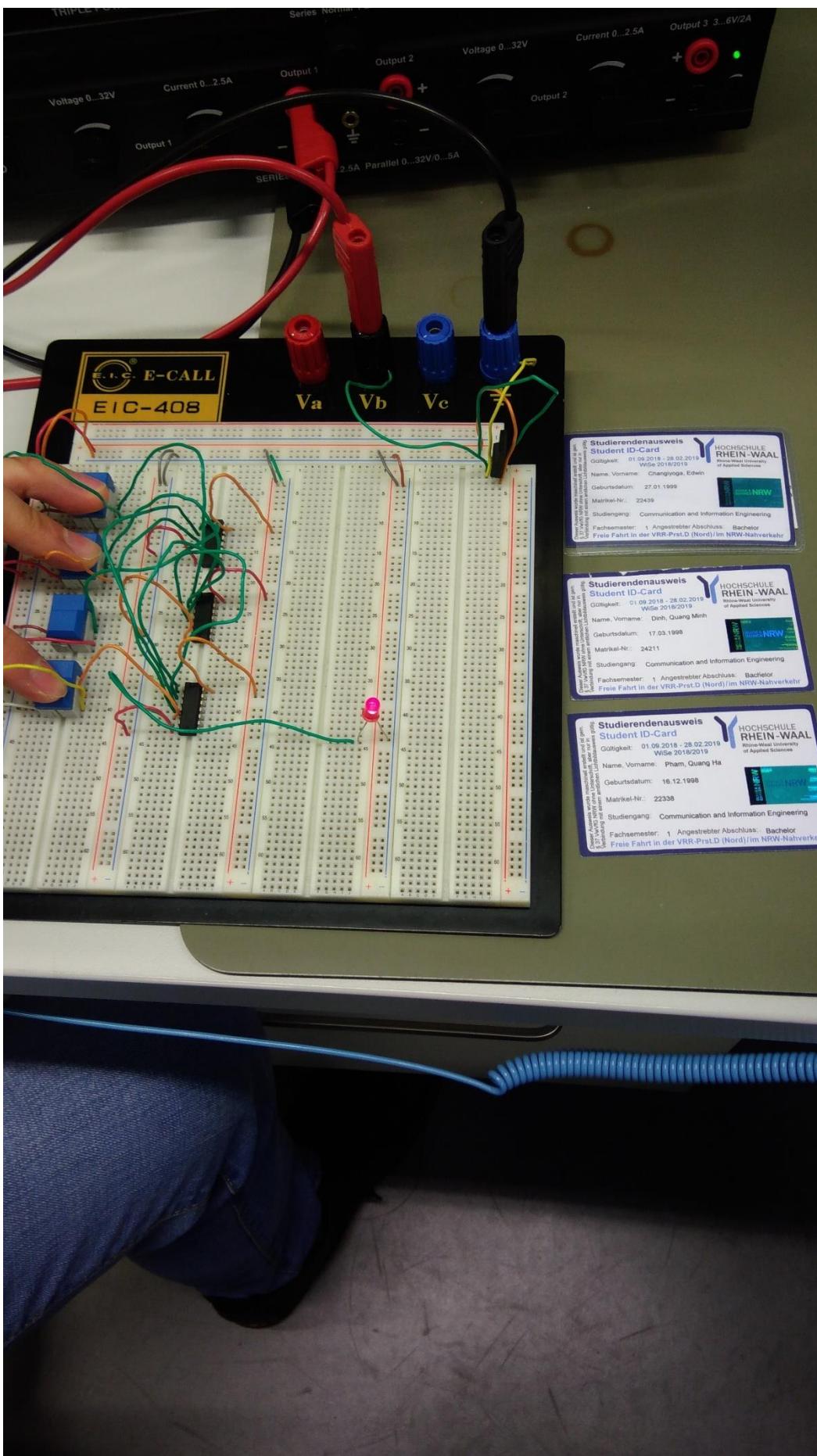


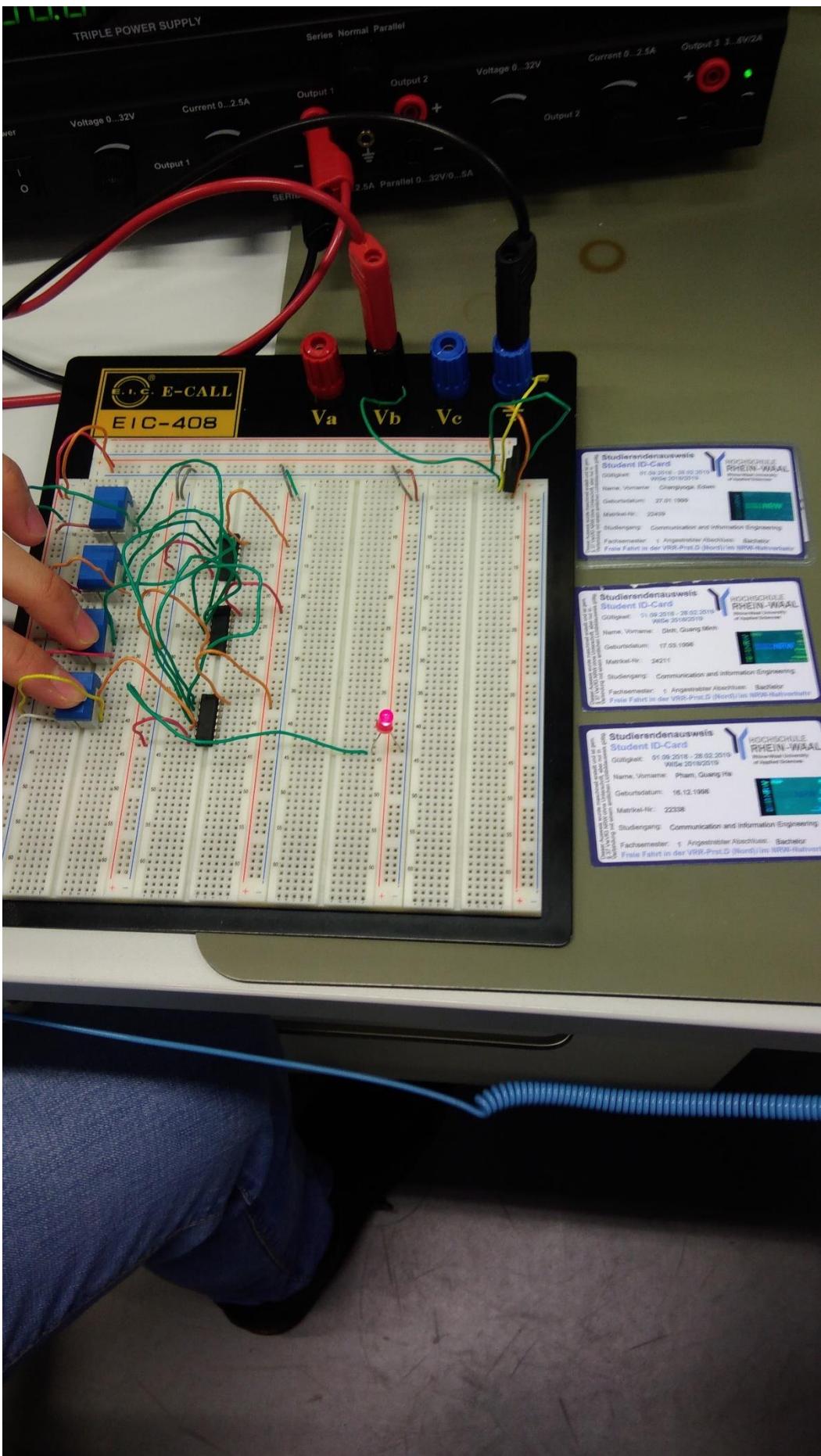


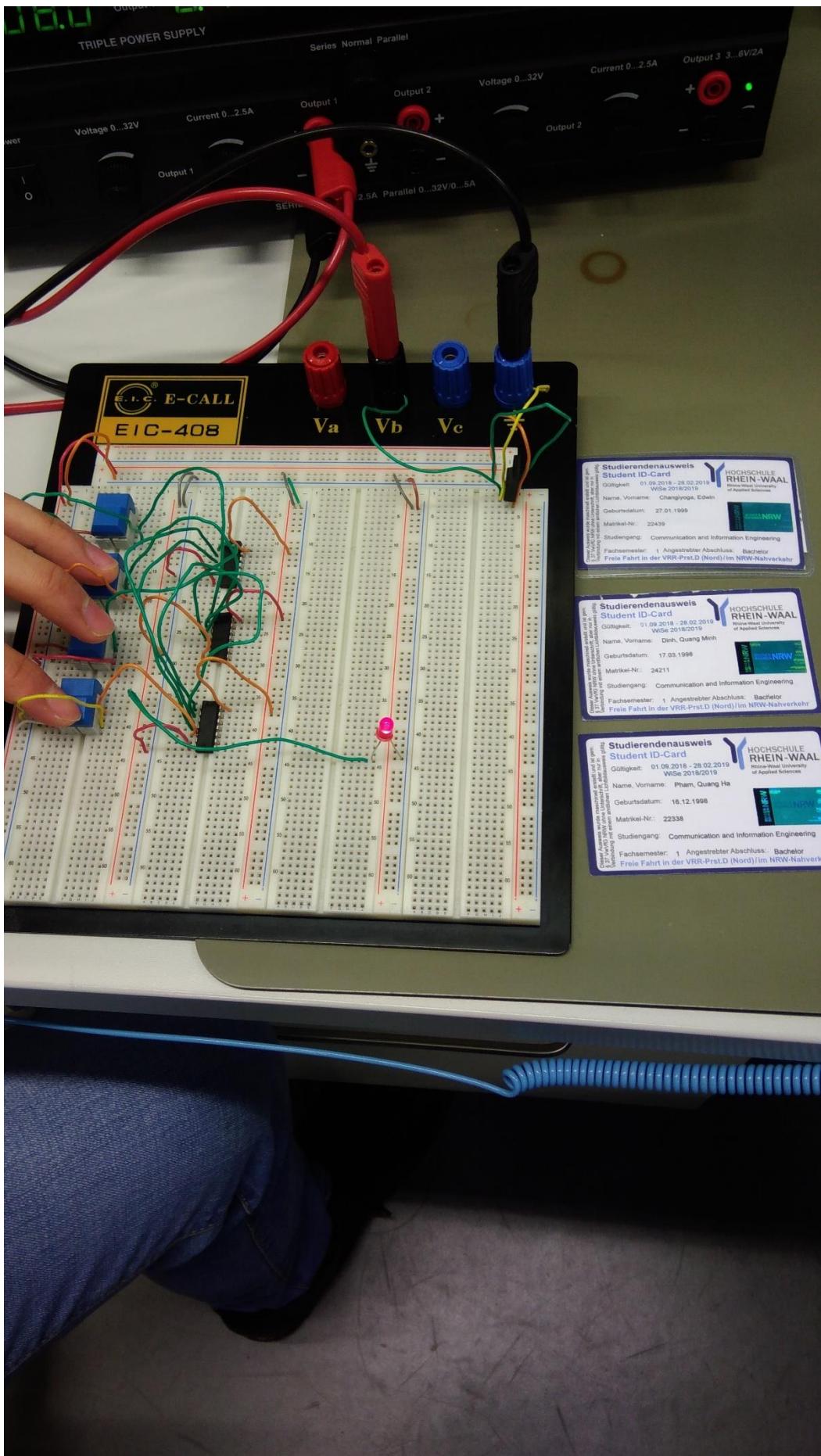












Challenge #3

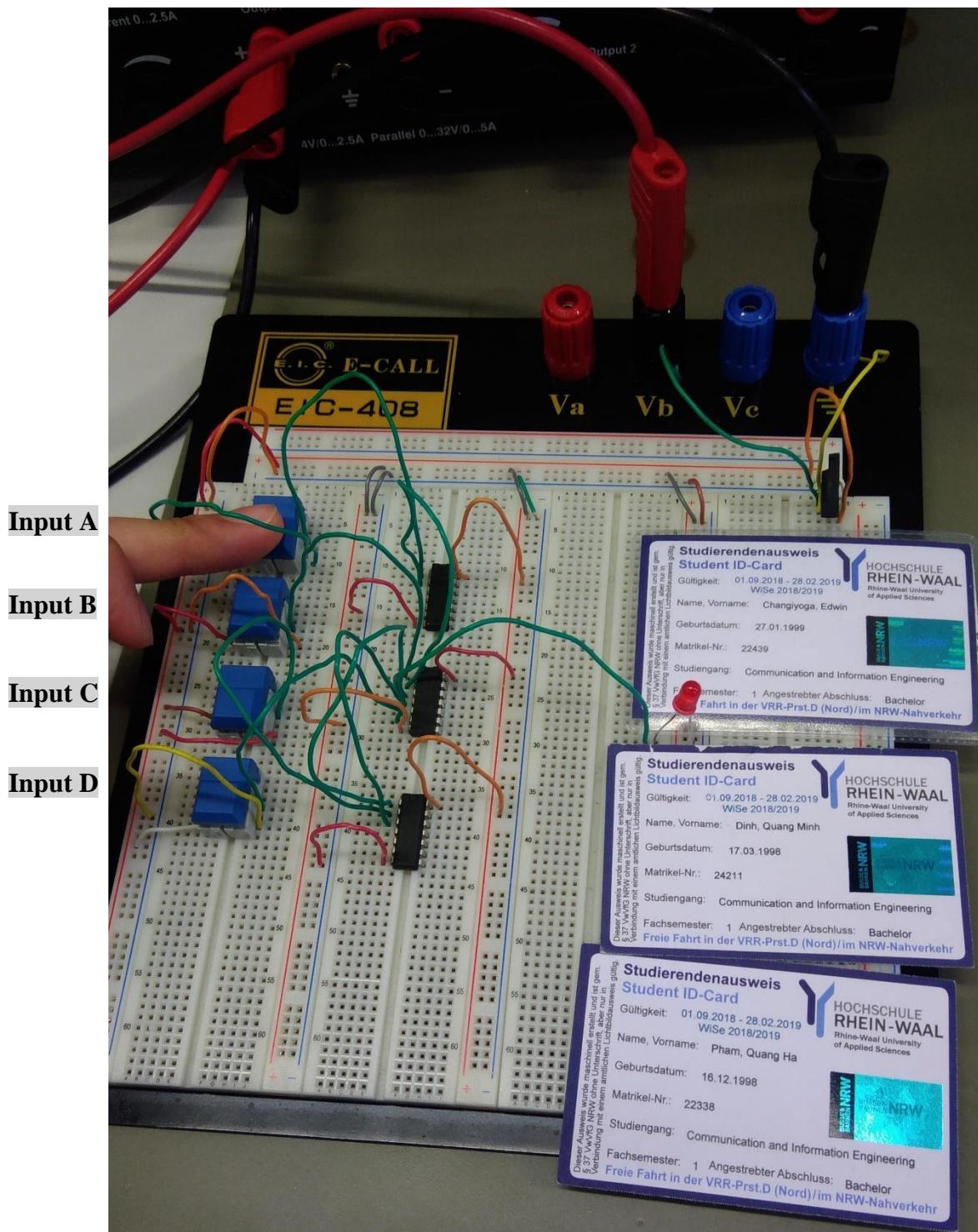
Abstract:

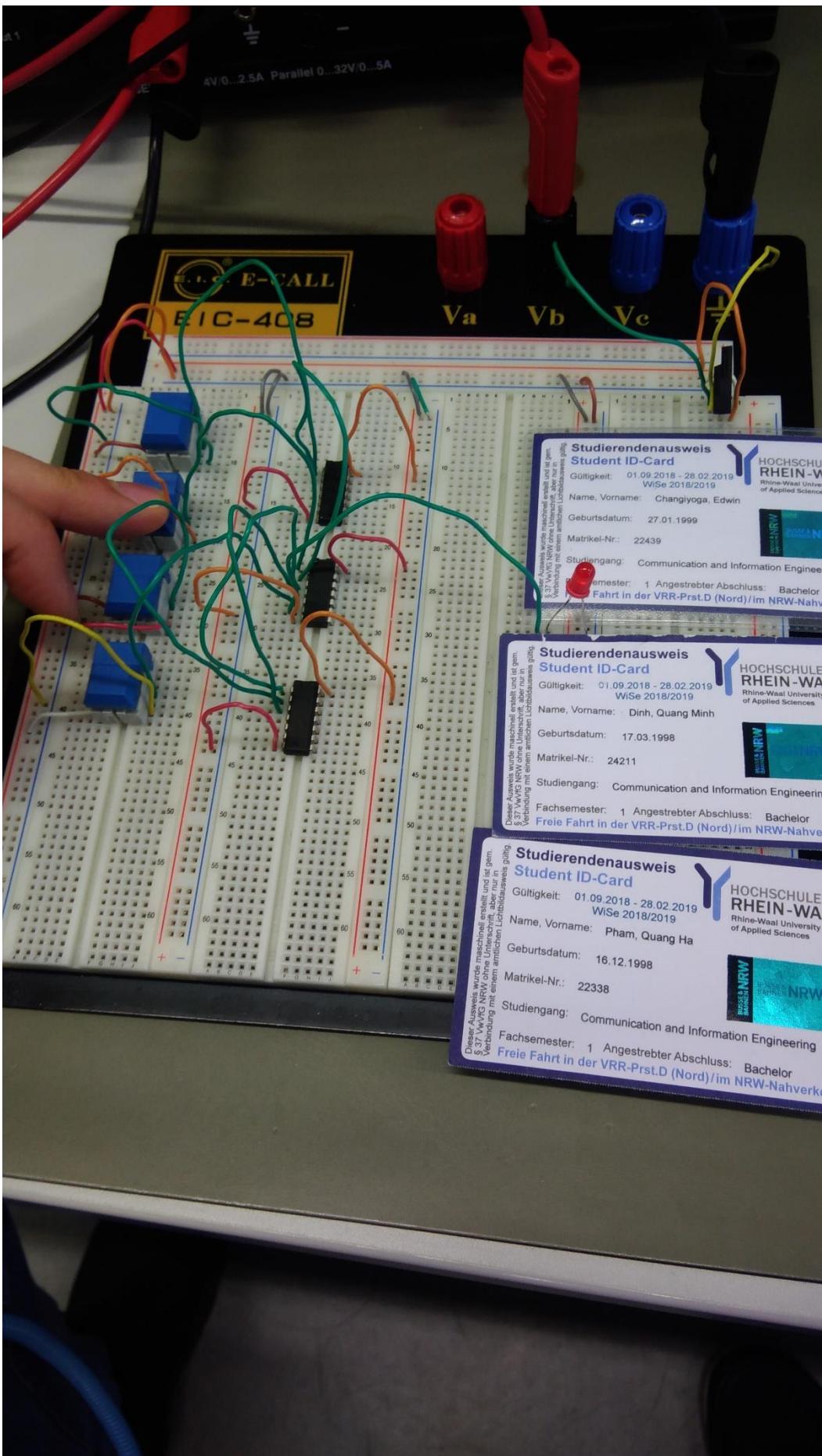
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

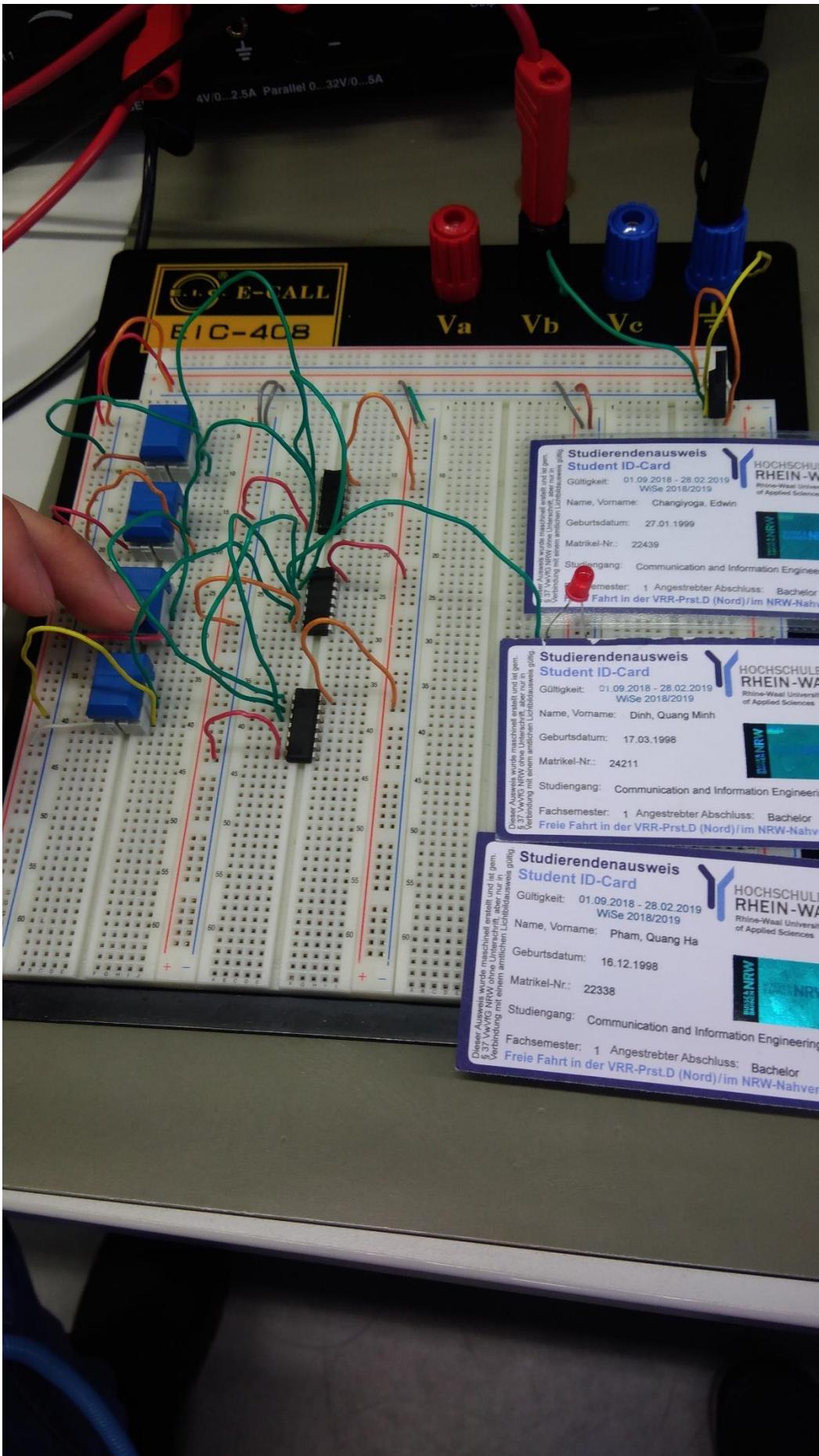
Input D in this case is both a Normally closed switch and an invert, so our way of thinking is: an invert of an invert is a Normally open switch.

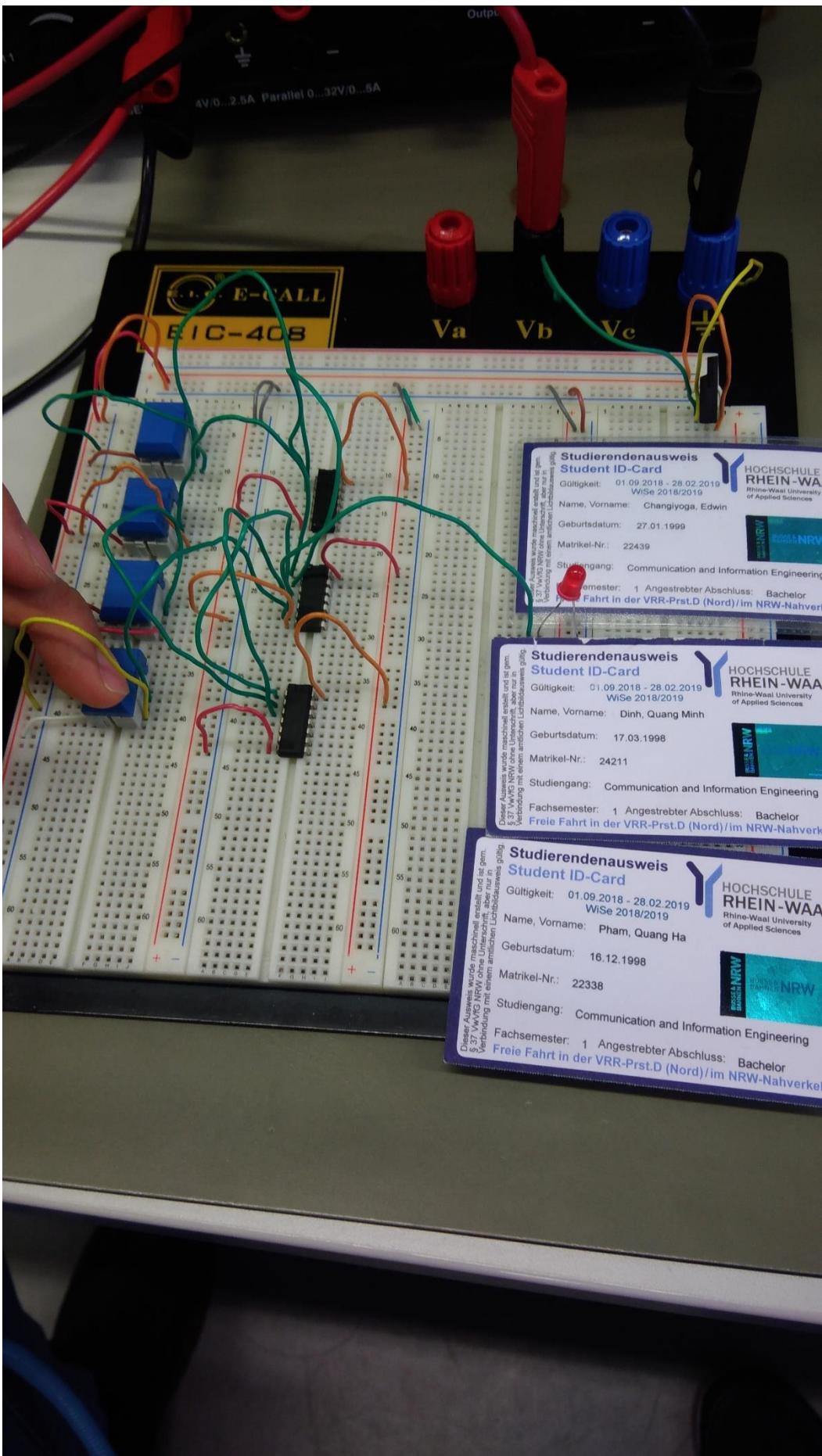
Our final circuit is: $((\neg A \wedge B) \vee C) \wedge D$

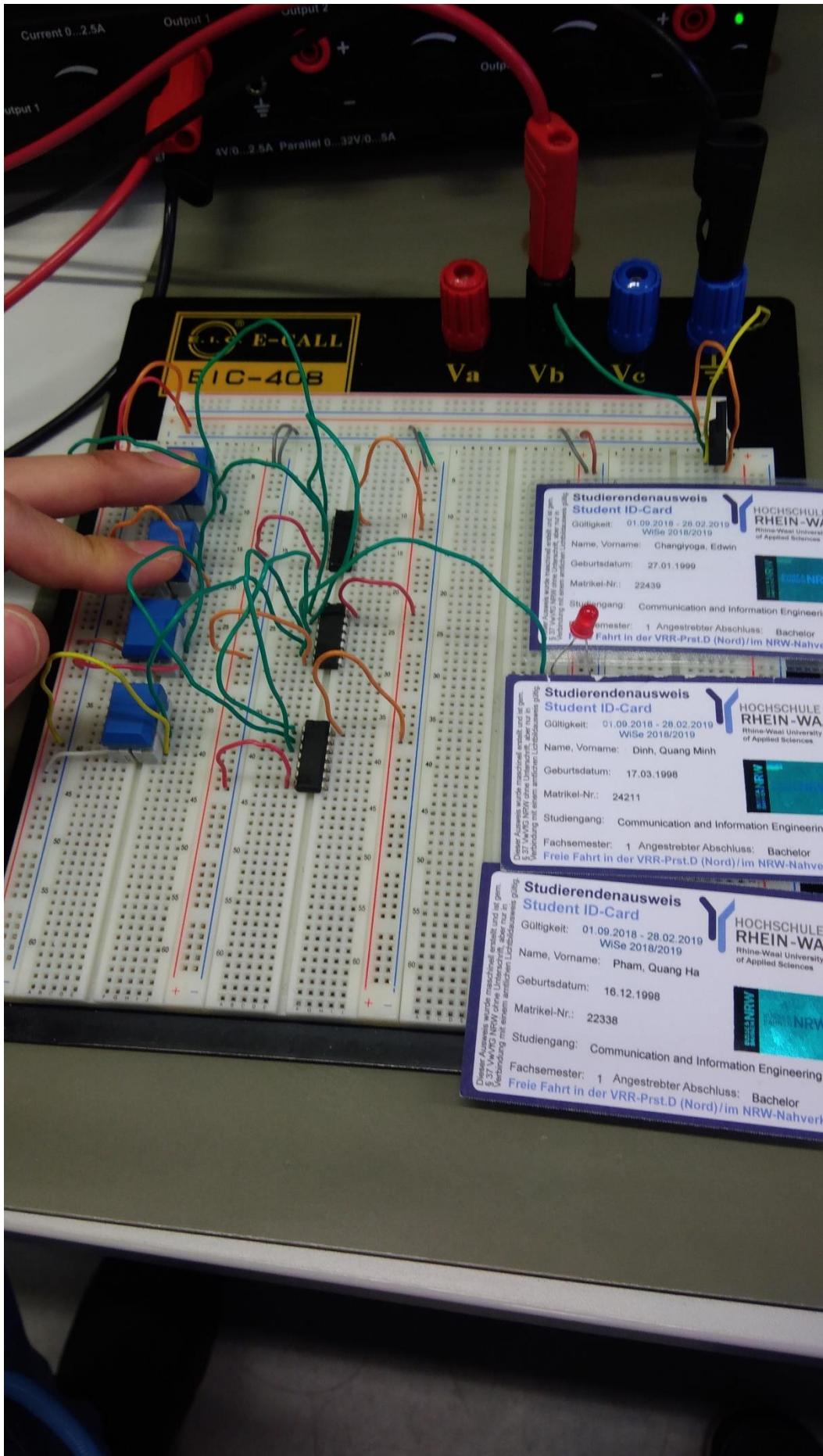
Pictures:

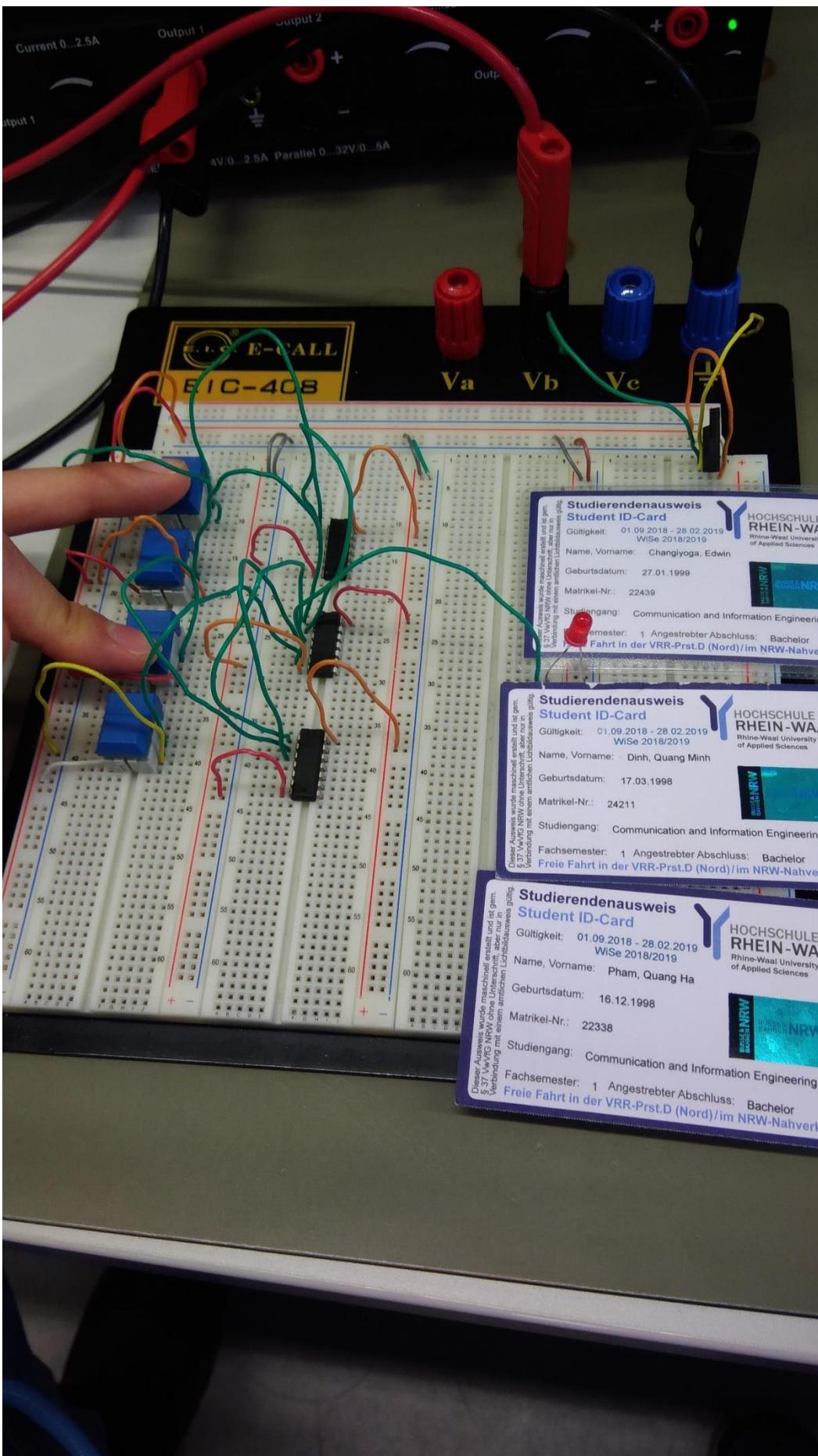


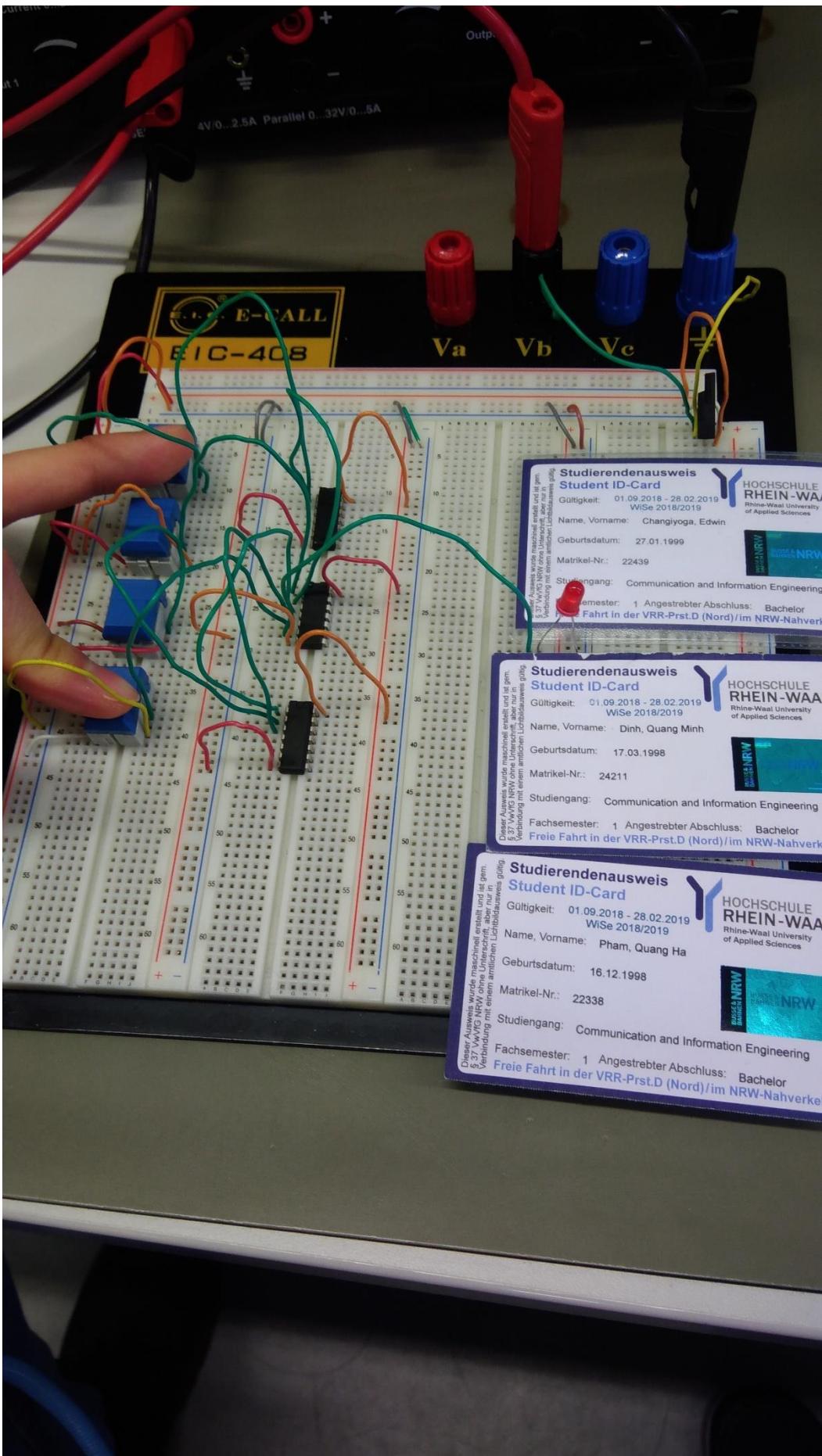


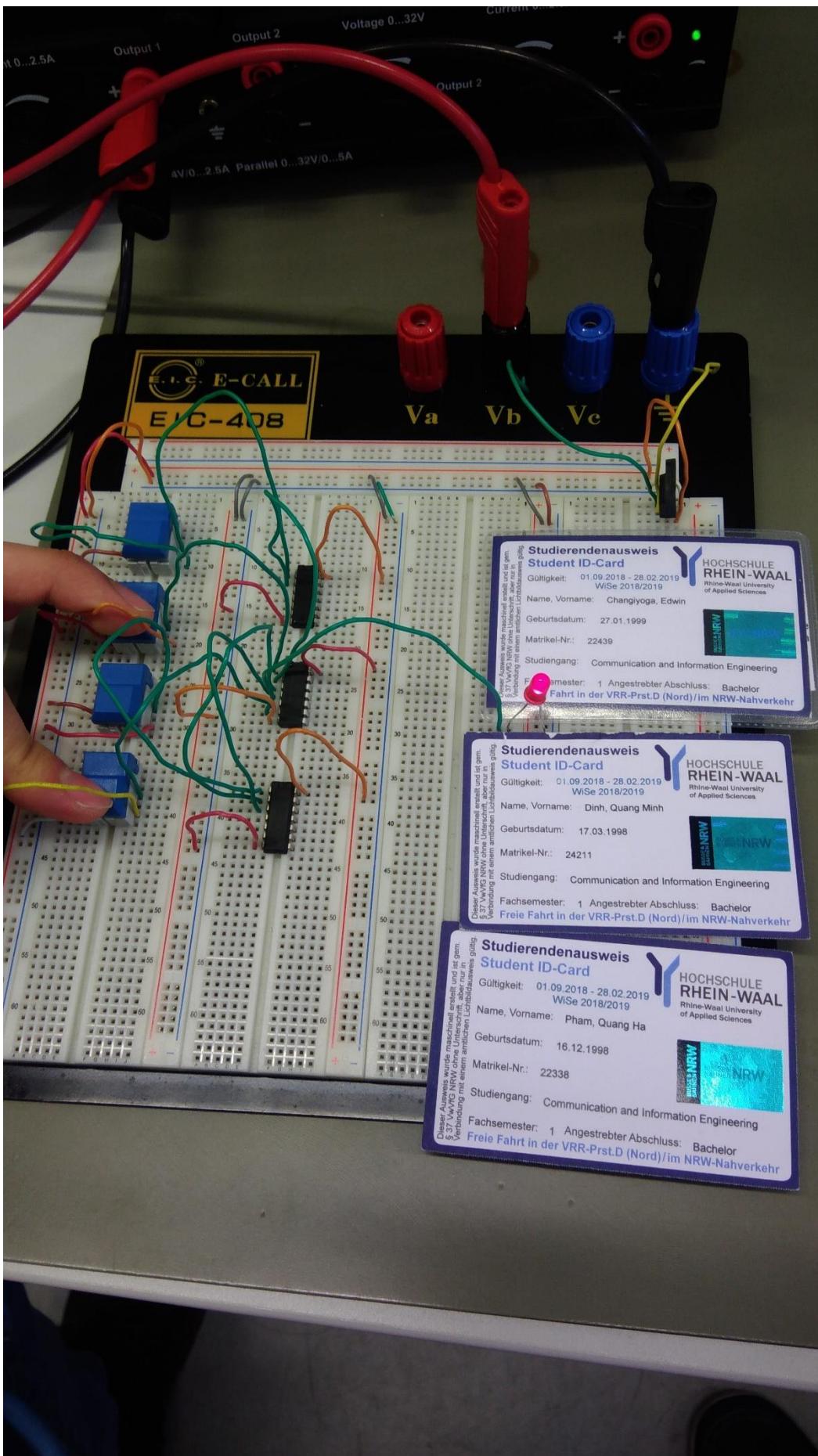


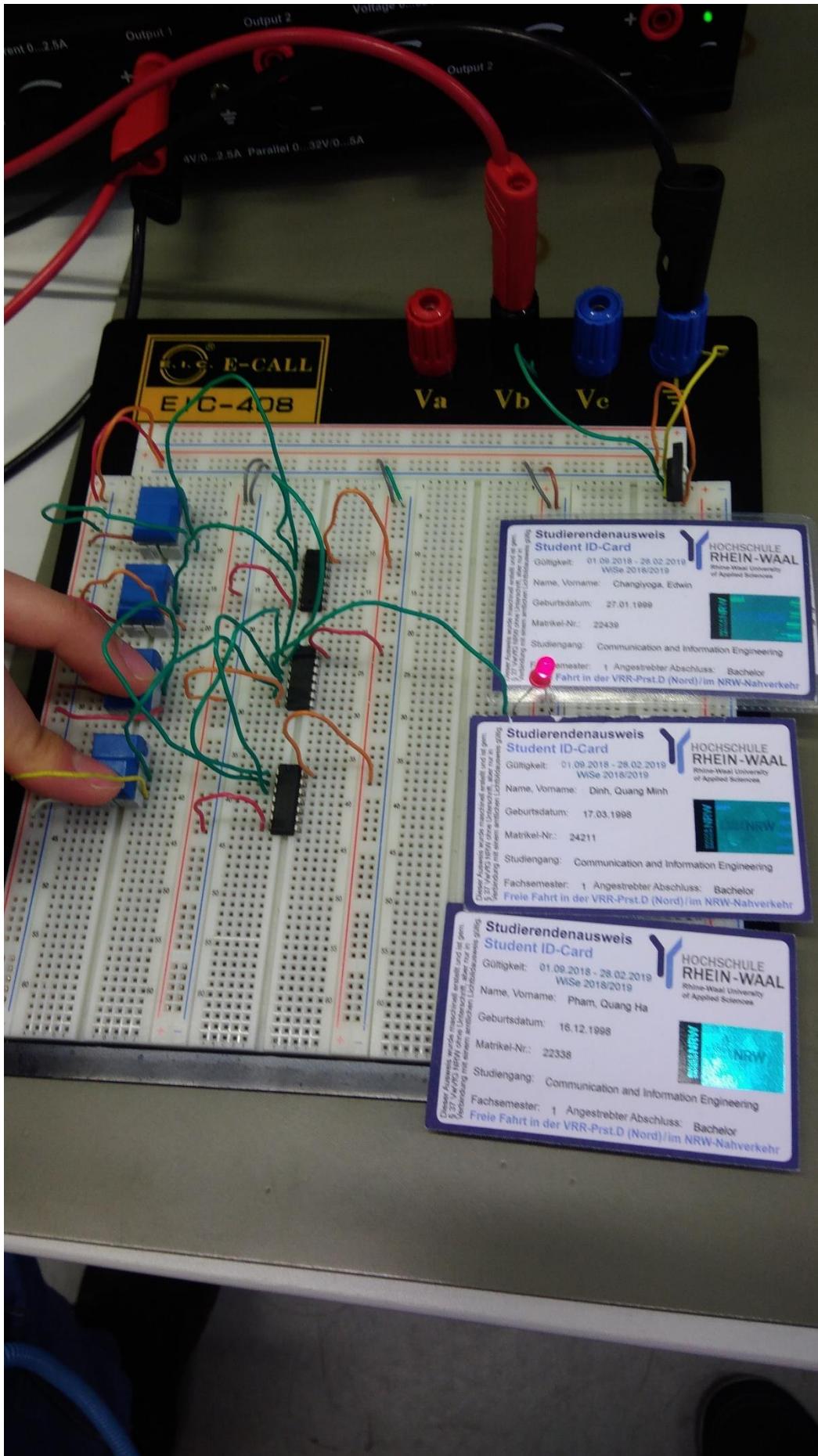


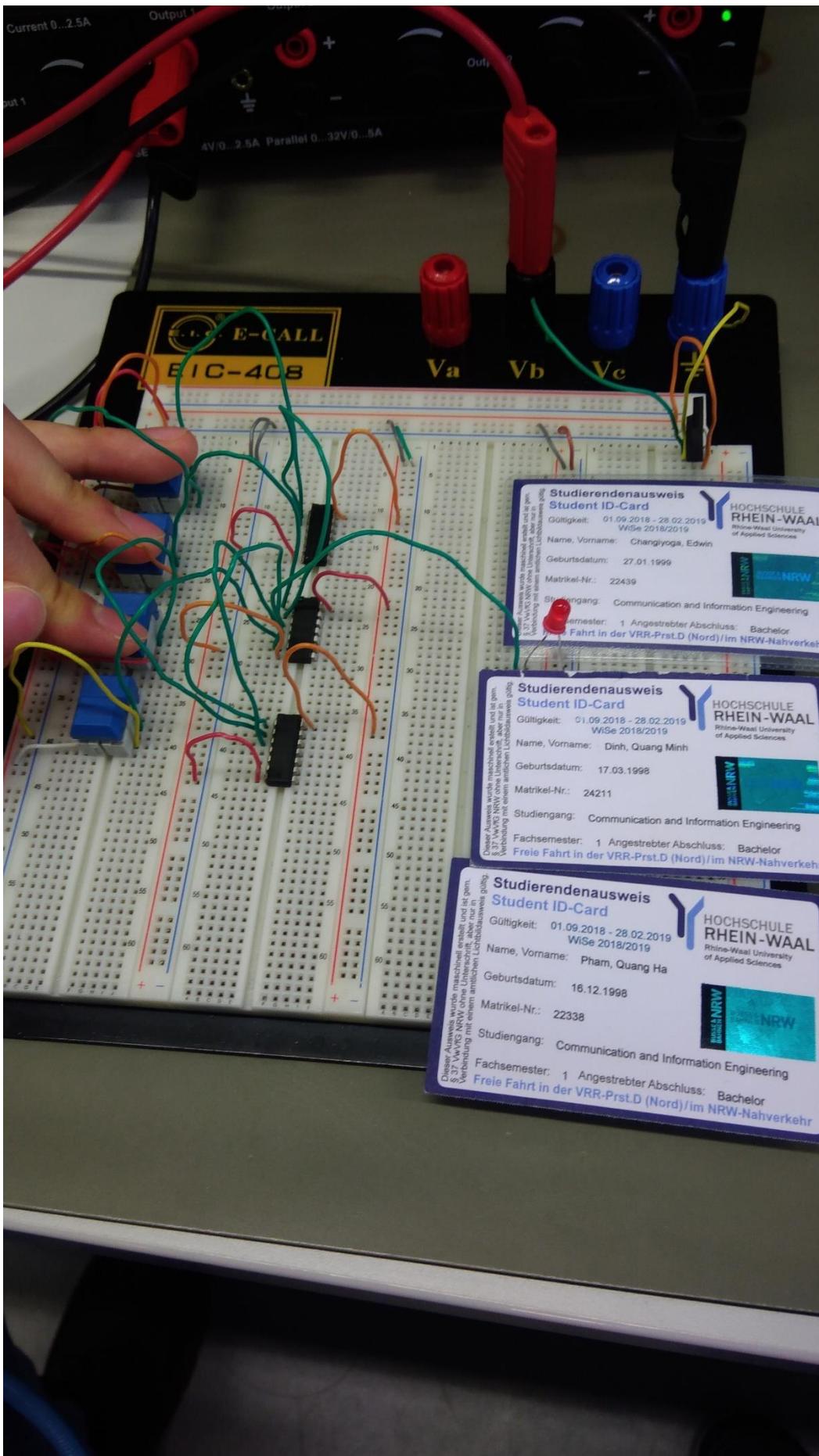


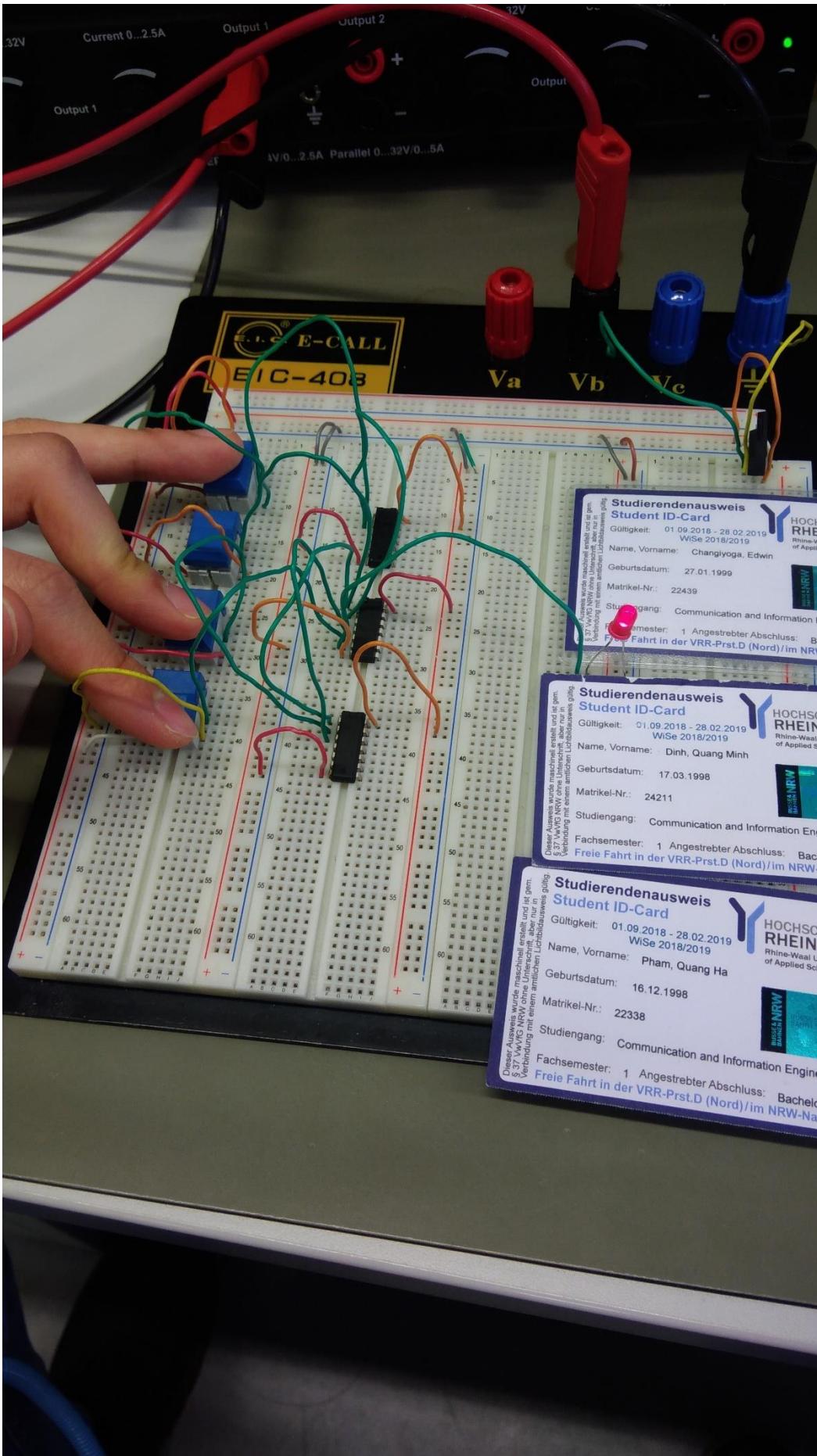


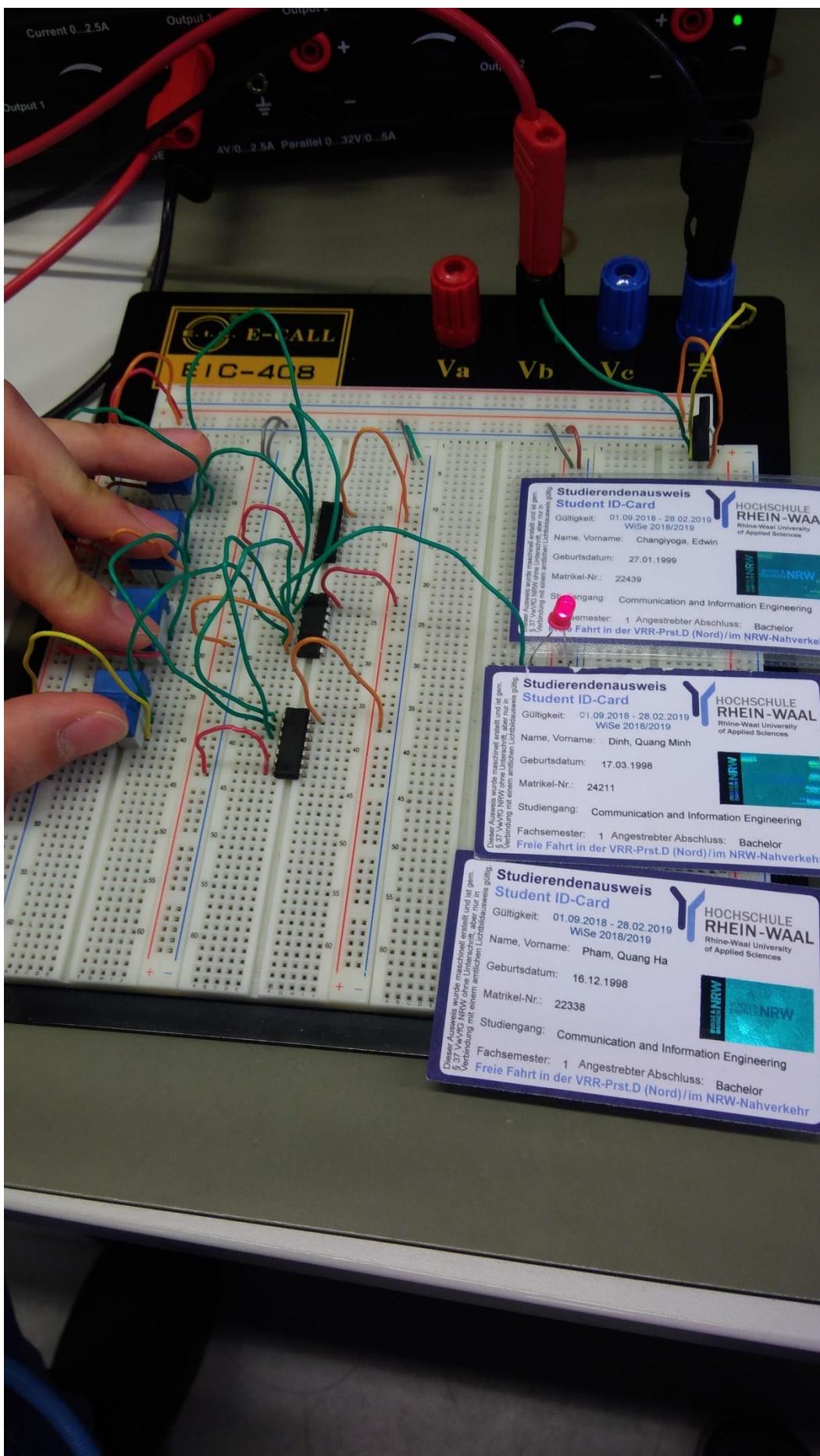










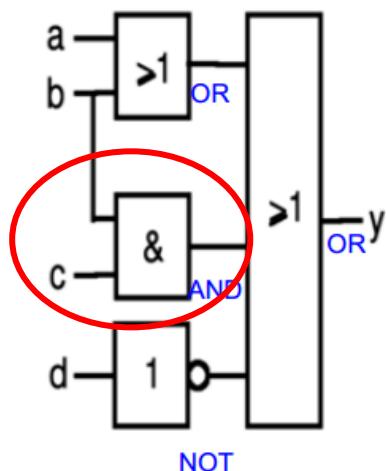


Challenge #4

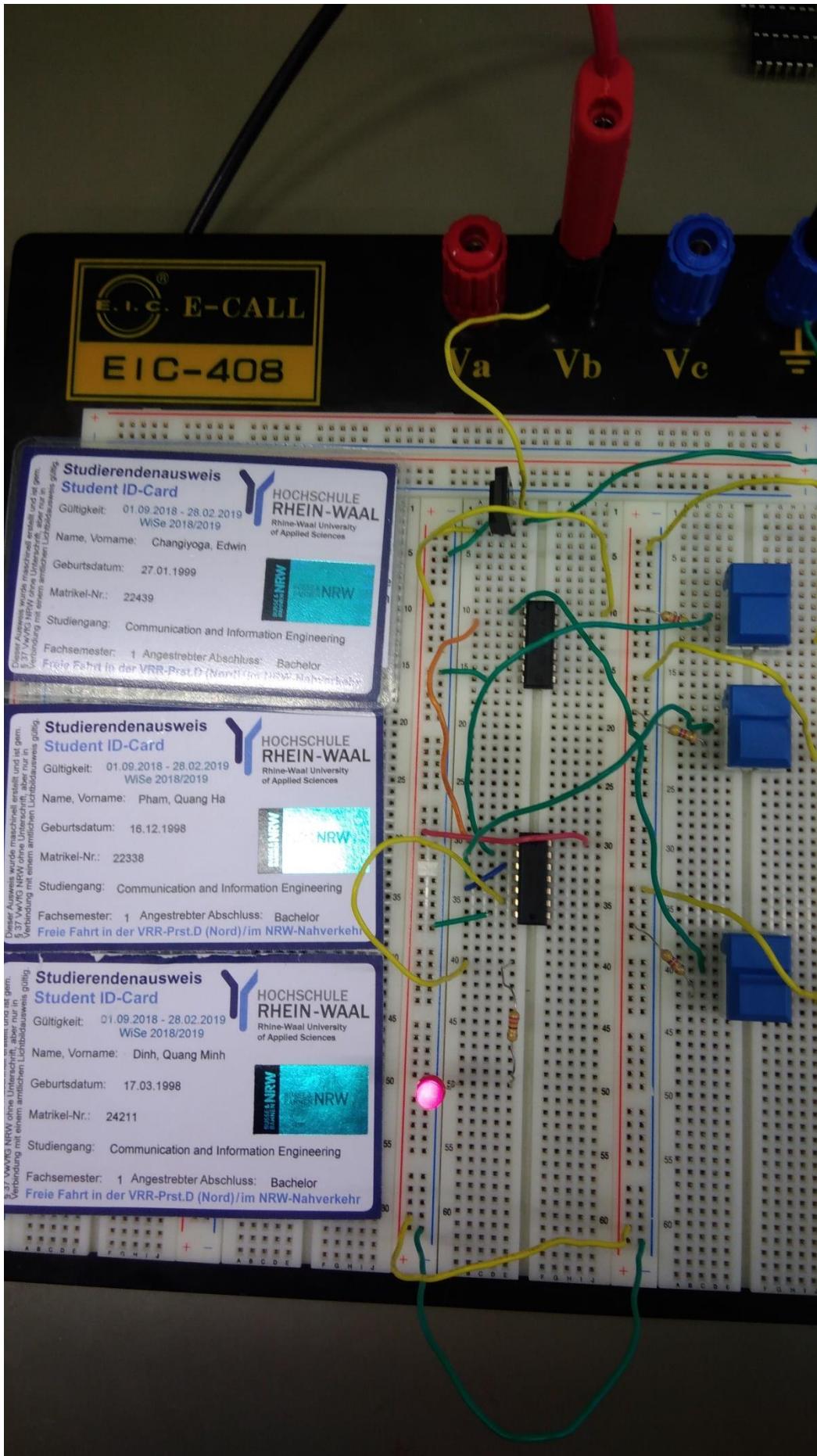
Abstract:

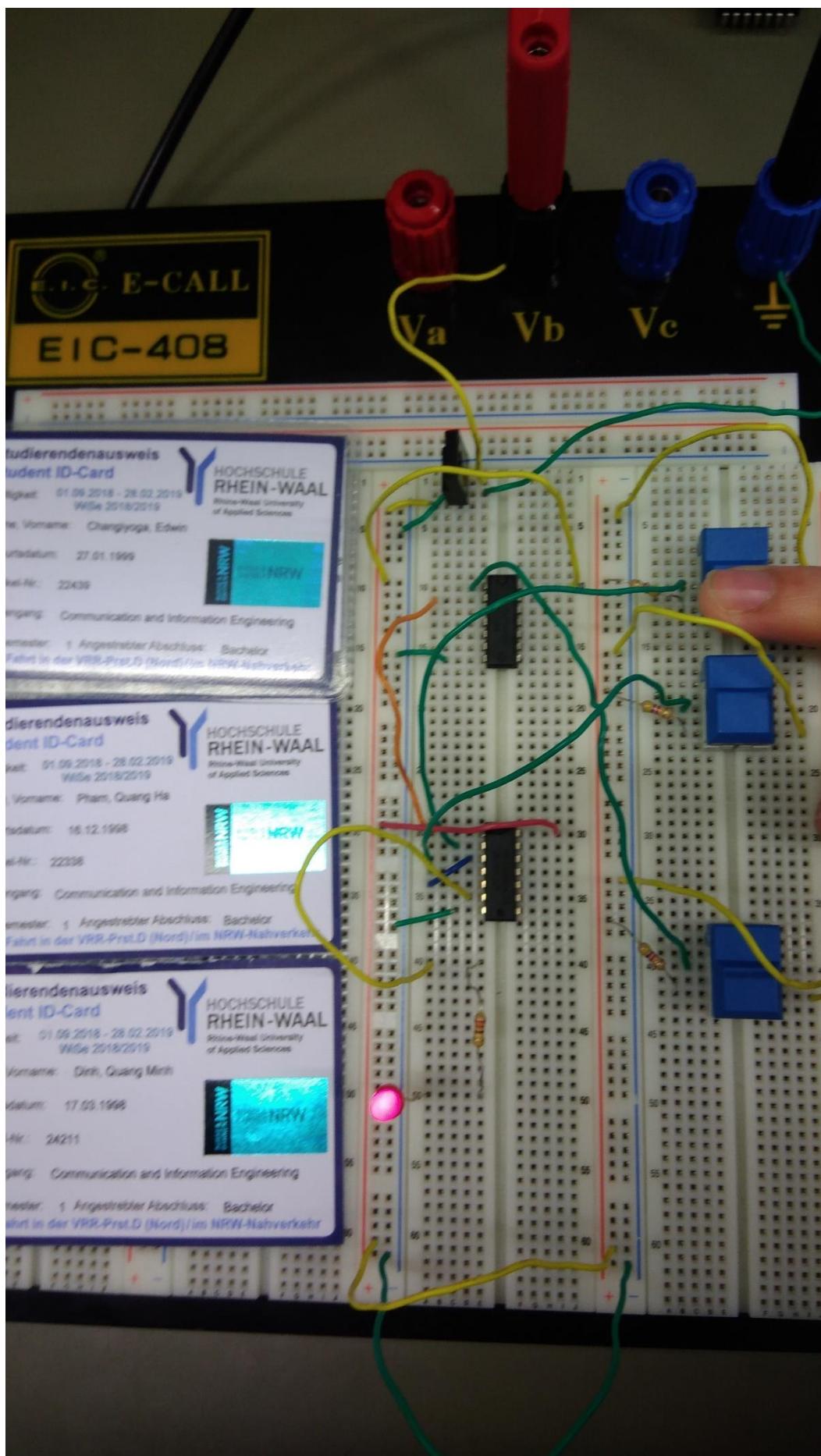
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

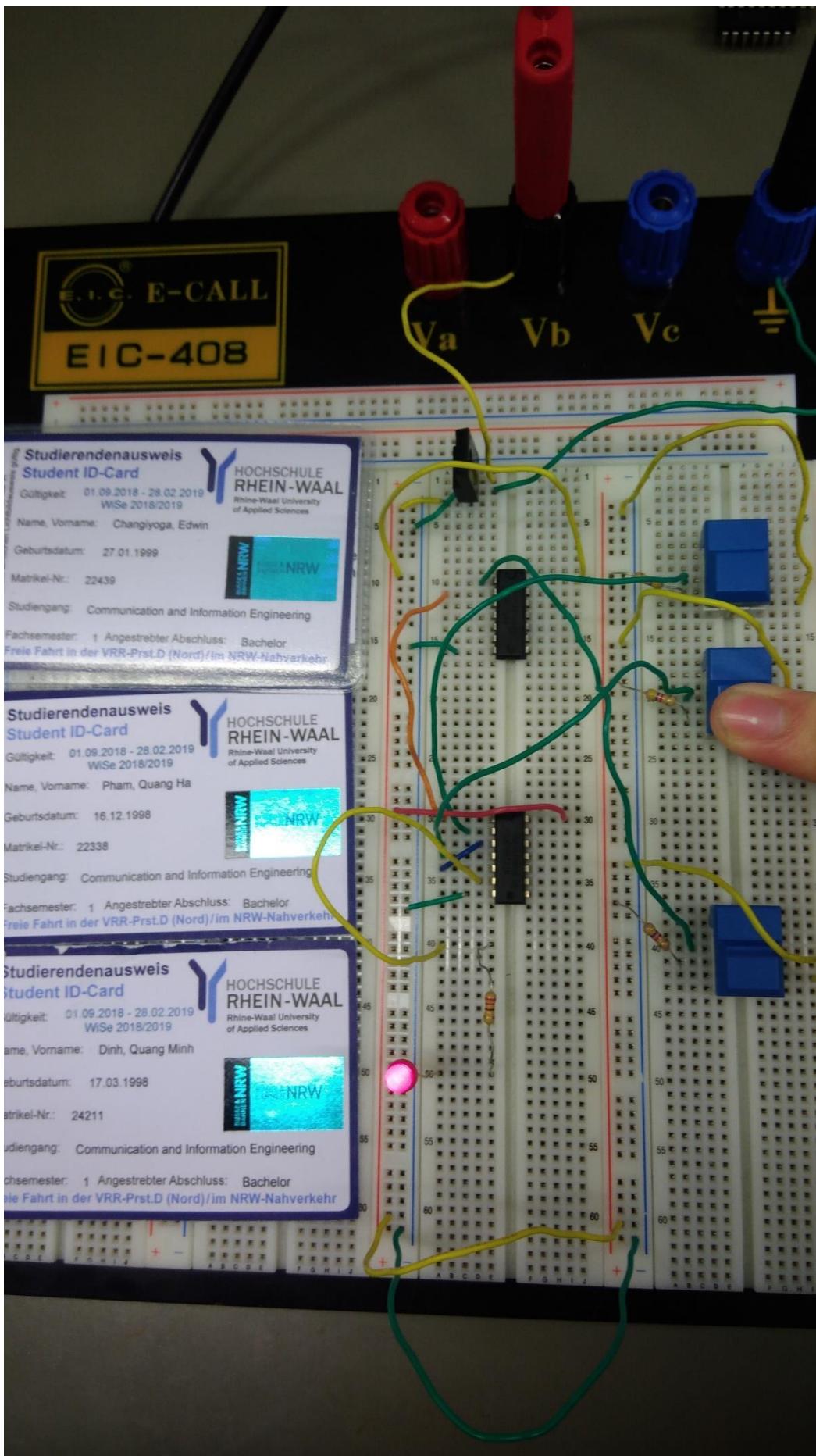
The area marked in the red circle is redundant and can be removed without changing the state of the output Y. Our final simplified circuit is: $A \vee B \vee (\neg D)$.

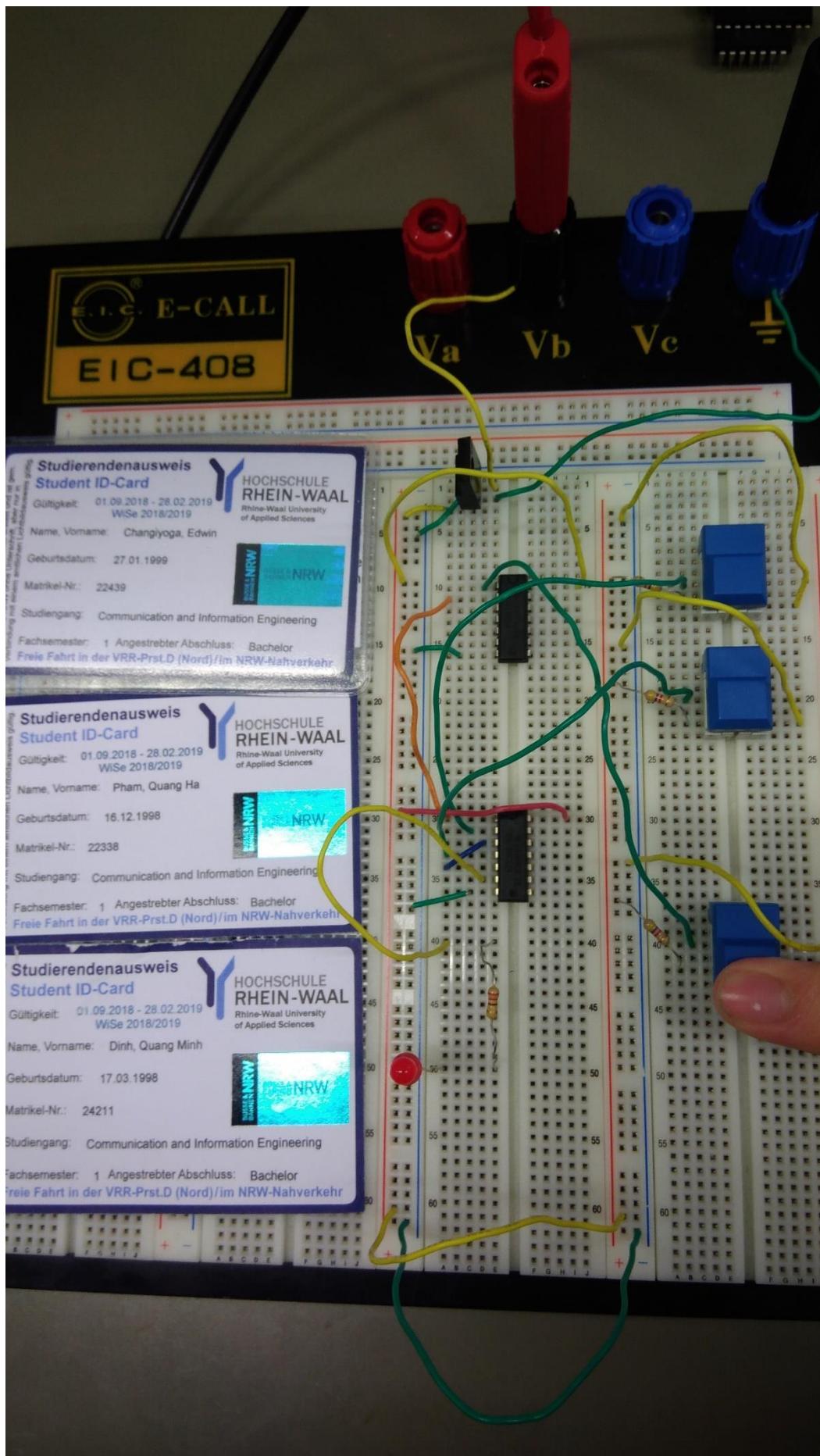


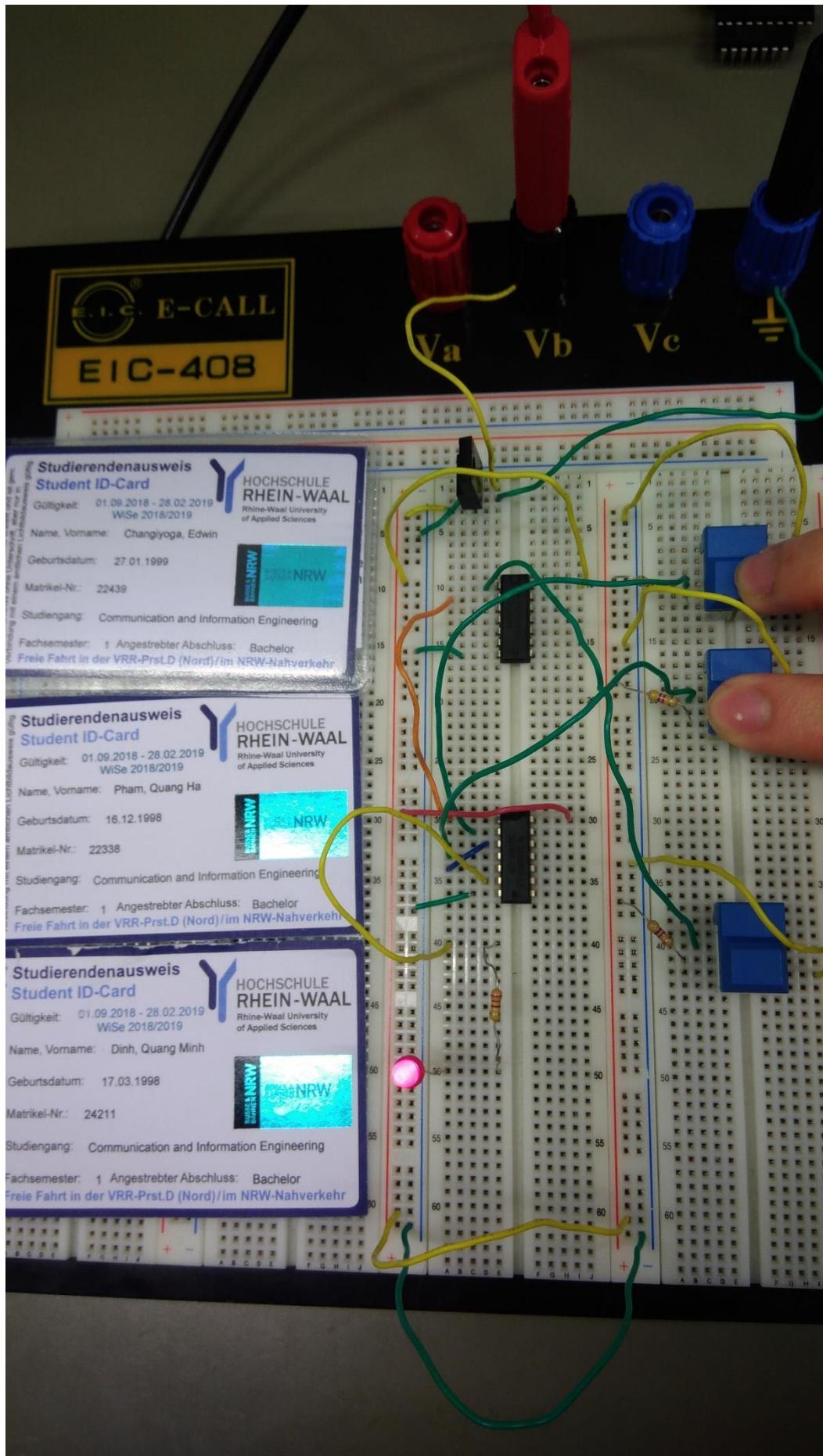
Pictures:

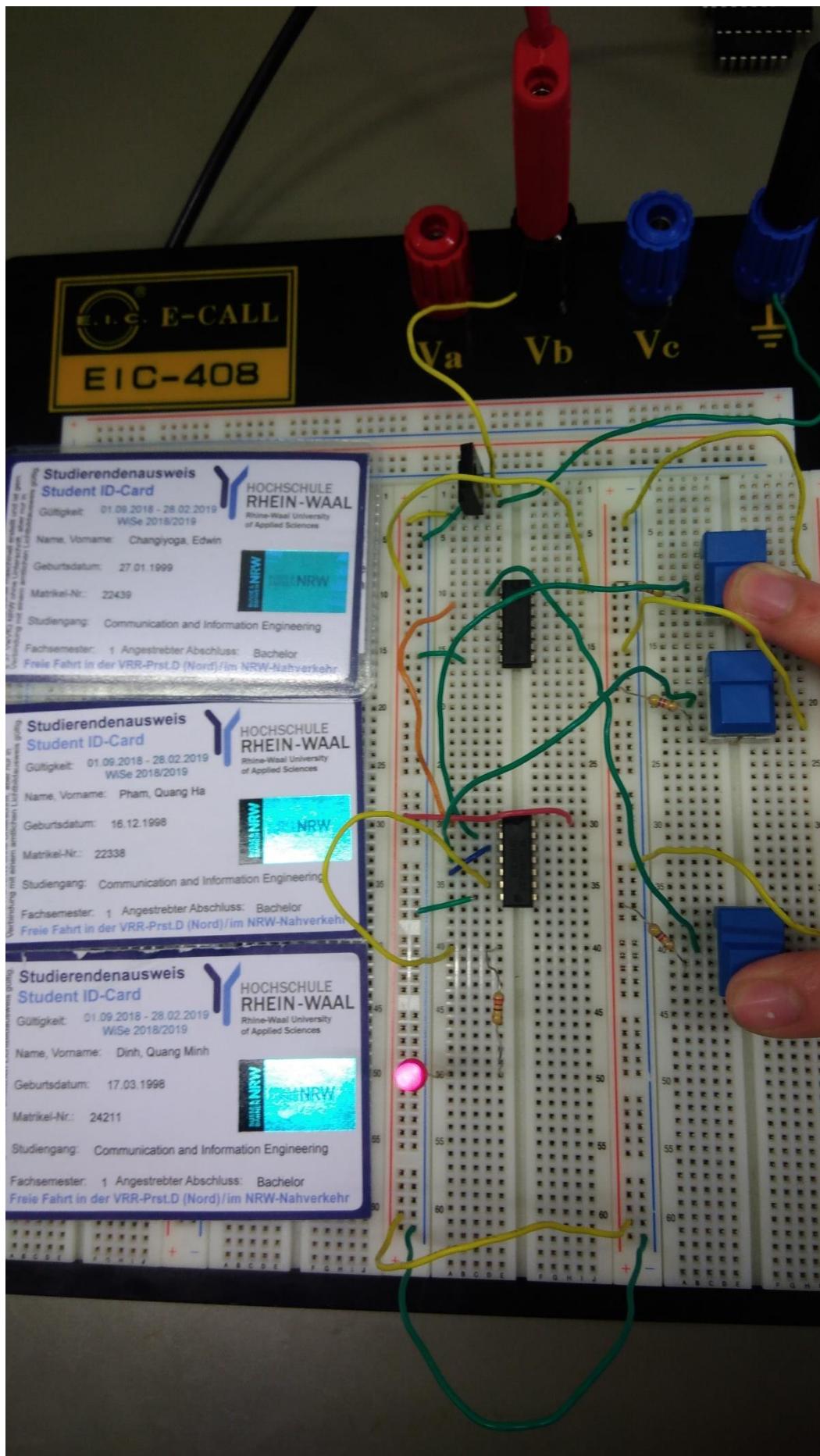


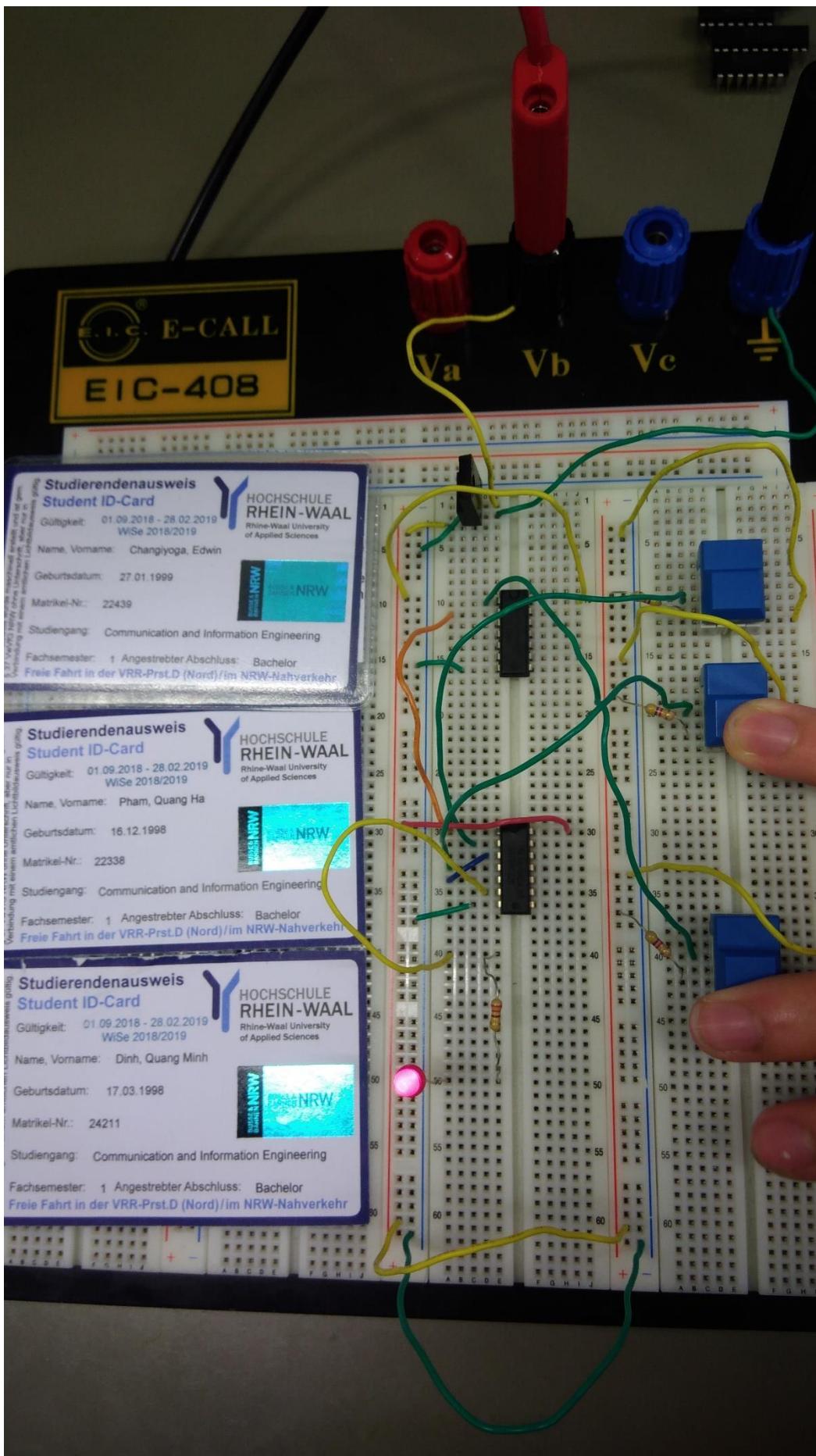


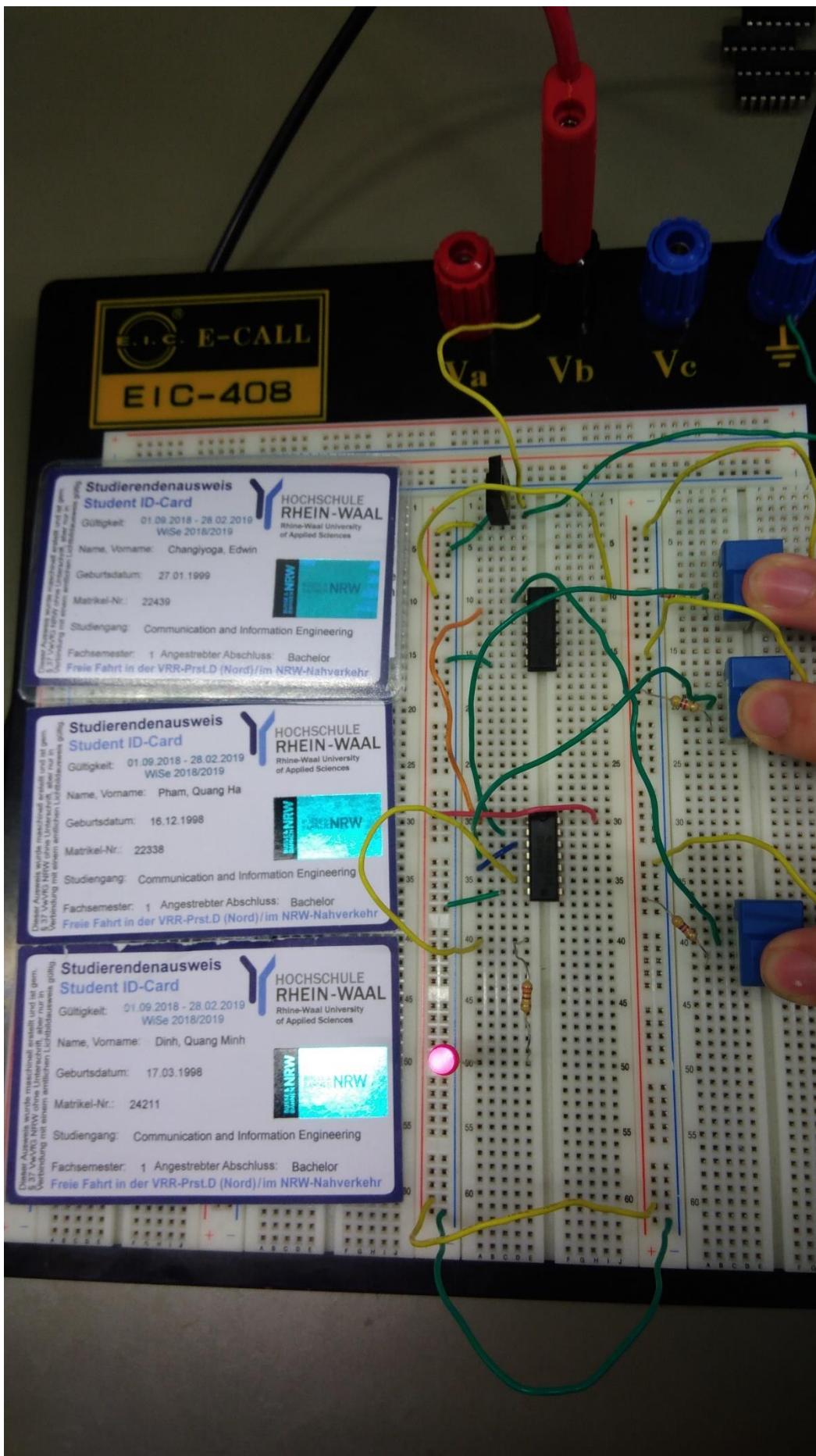












Result:

Comparison of truth table between before (left) and after (right) circuit simplification.

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1
1	1	1	1	1

A	B	D	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

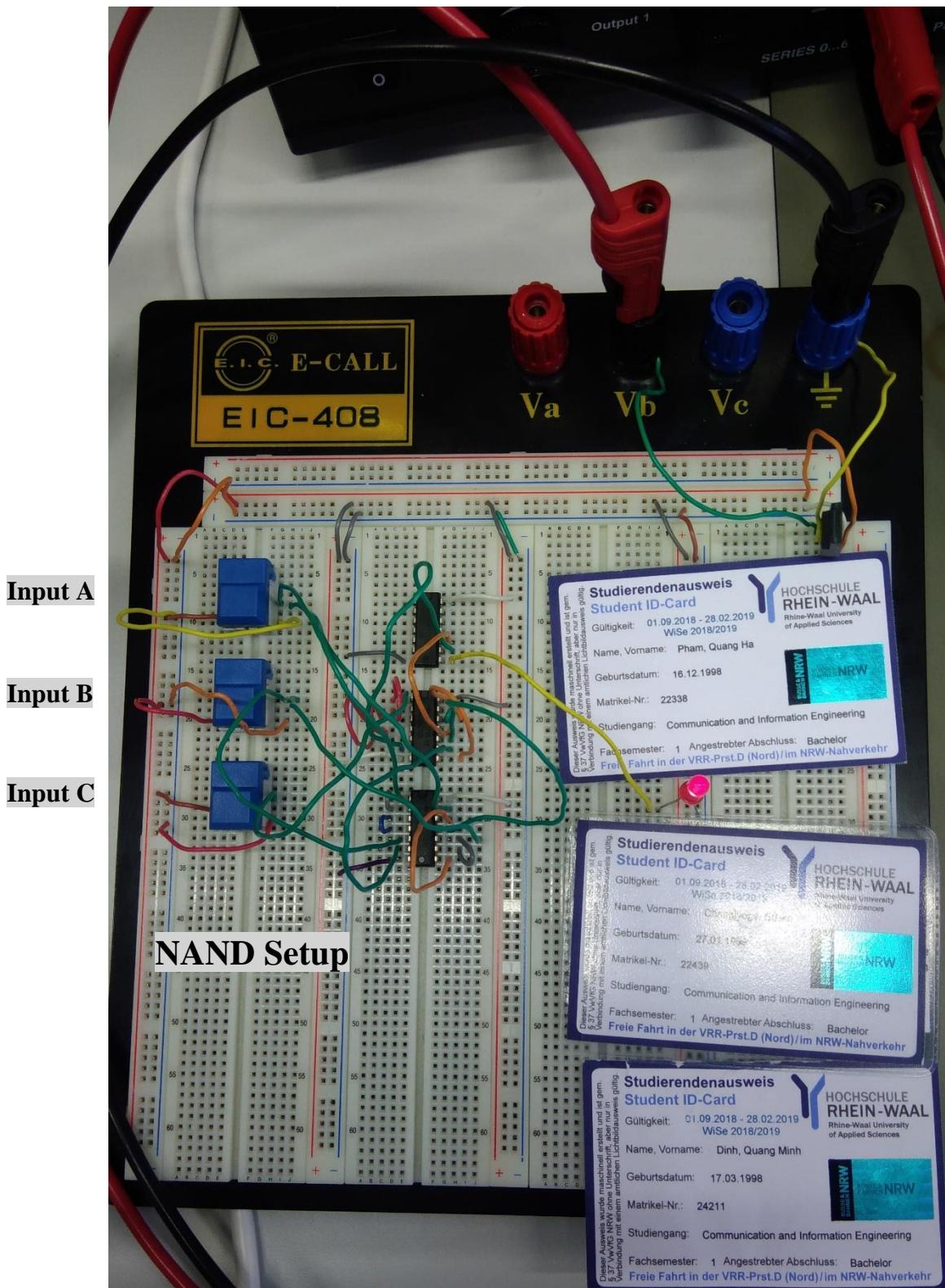
Challenge #5

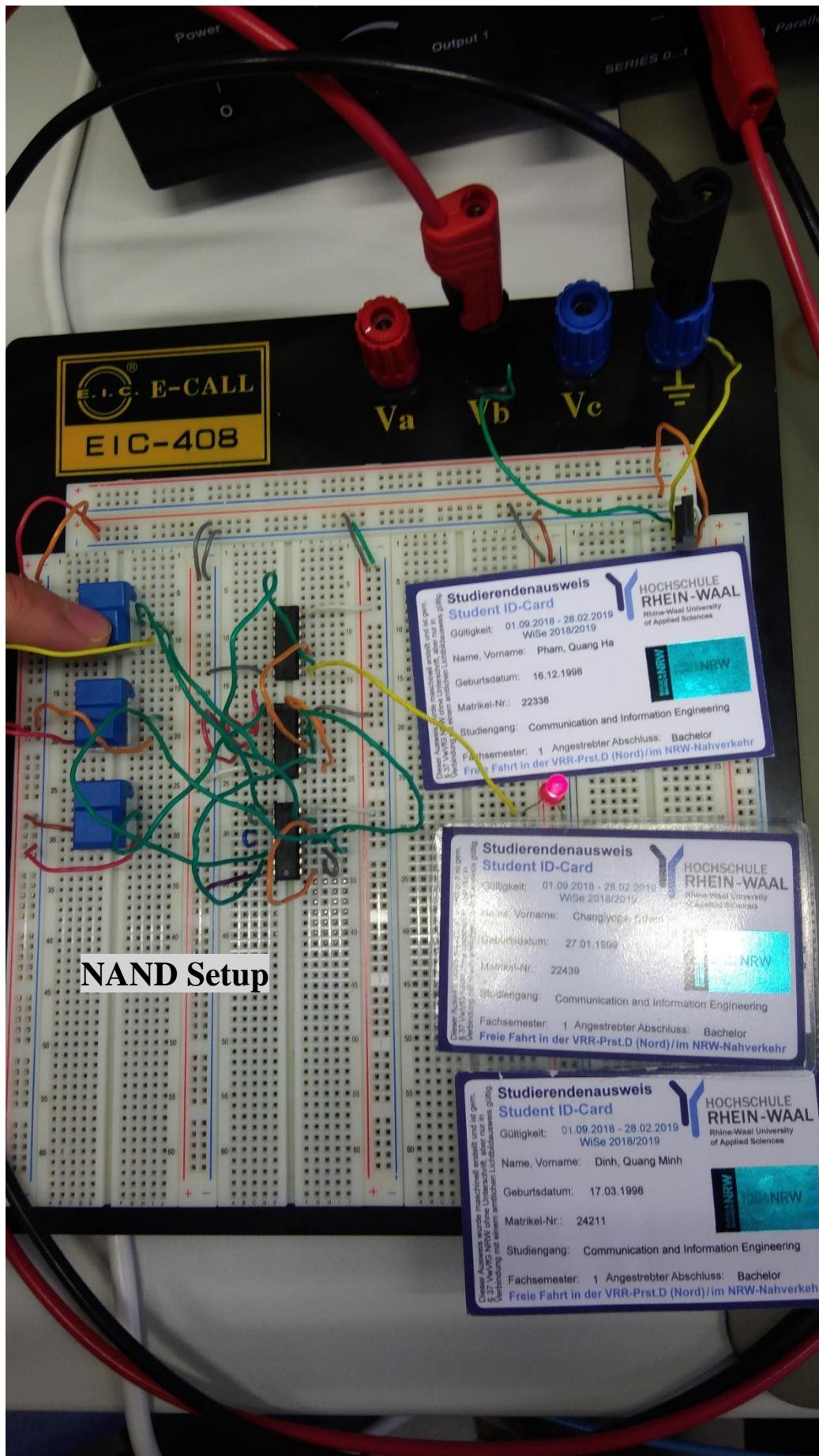
Abstract:

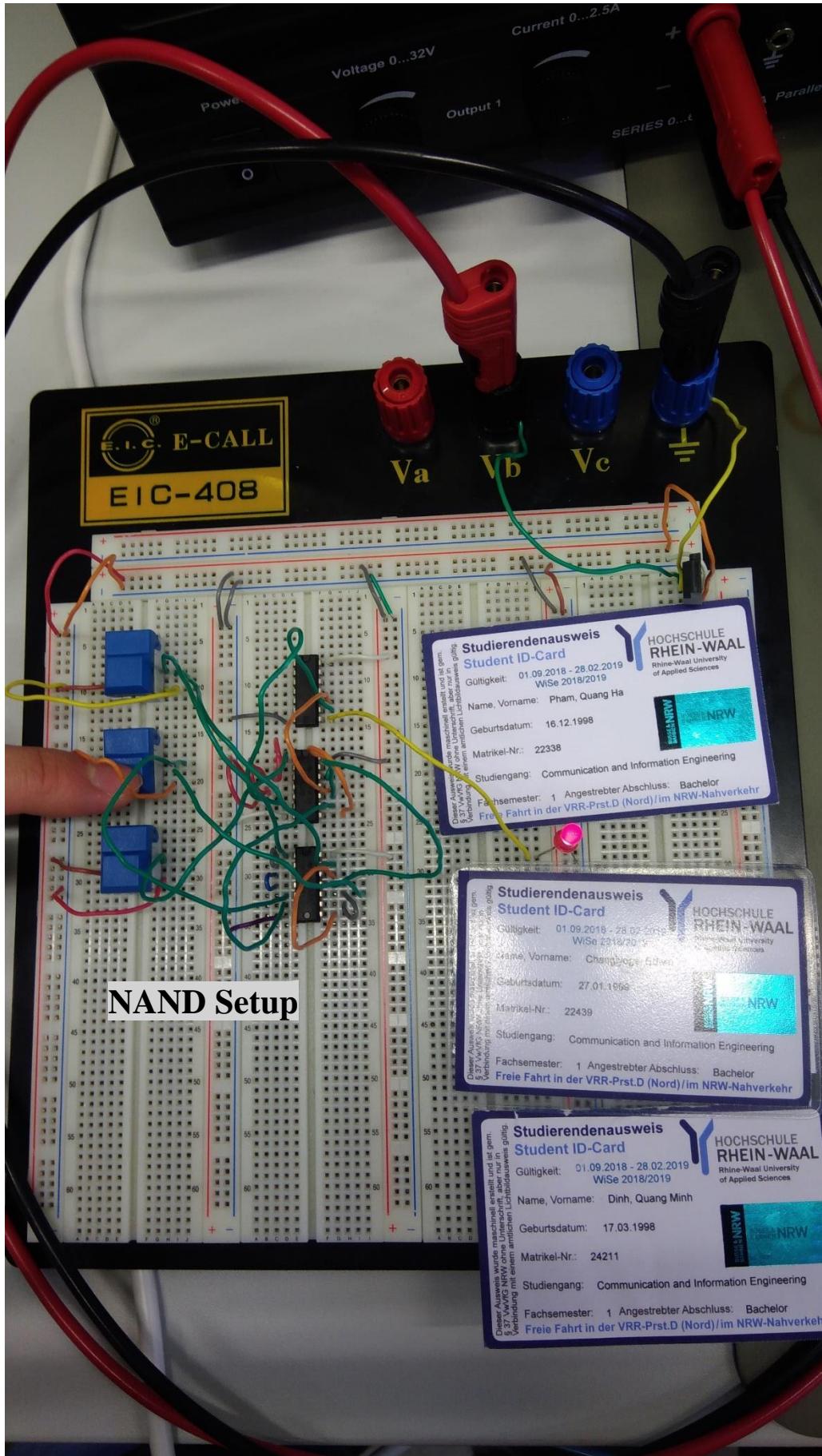
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

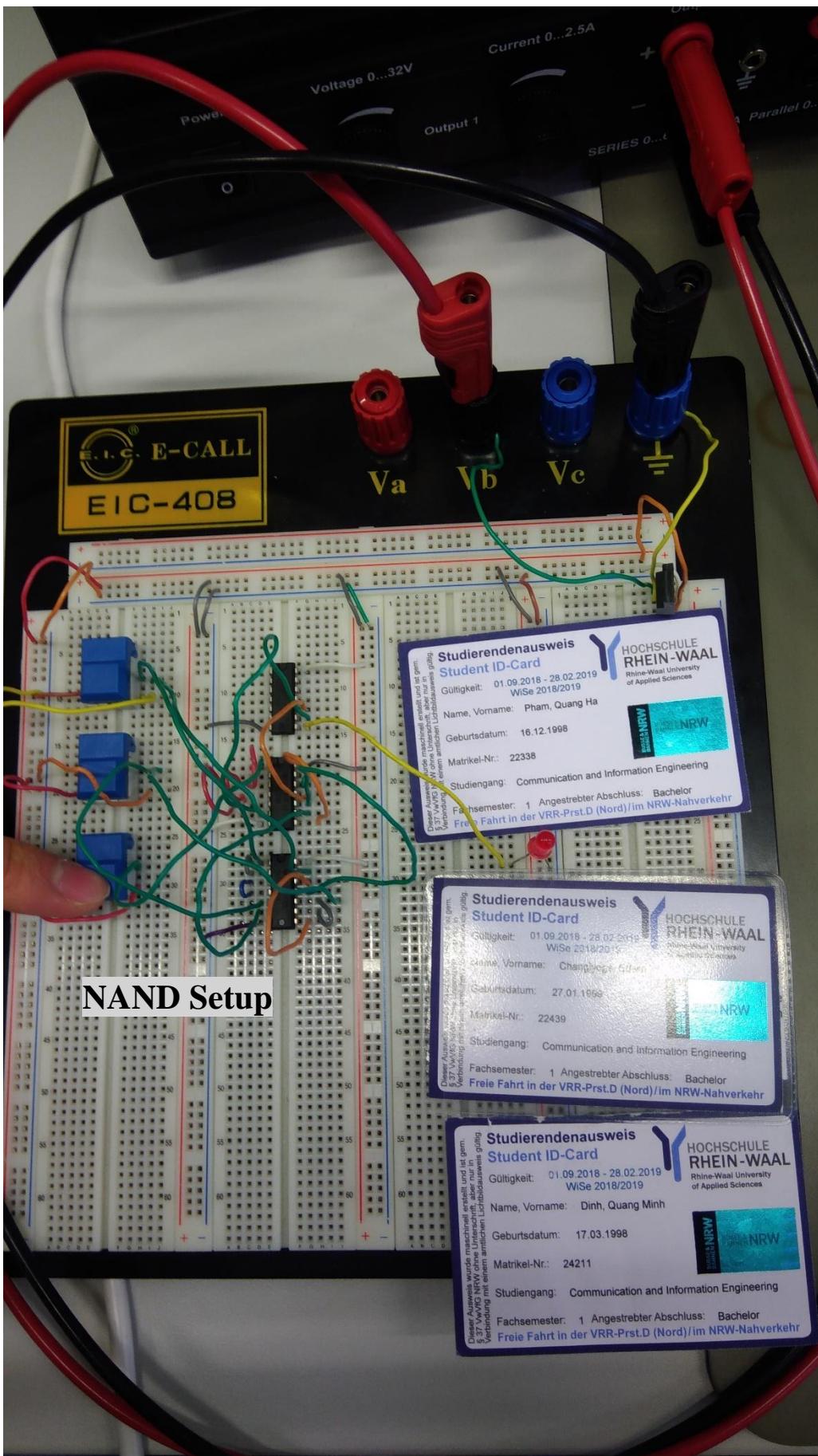
Using the truth table, we were able to simplify the switching function by using a KV diagram, our final result is: $(A \wedge \neg B) \vee (\neg A \wedge B) \vee (\neg C)$.

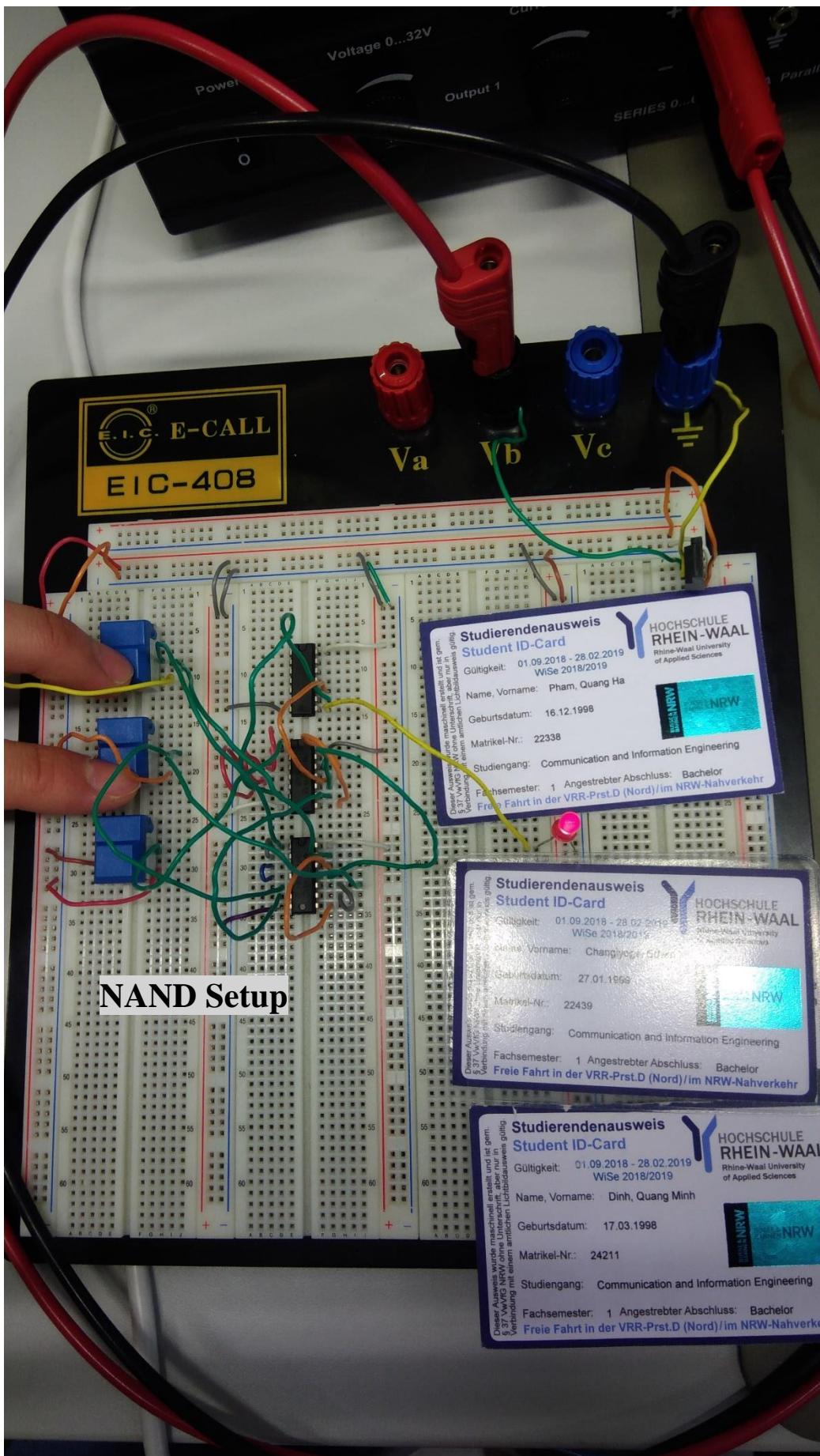
Pictures:

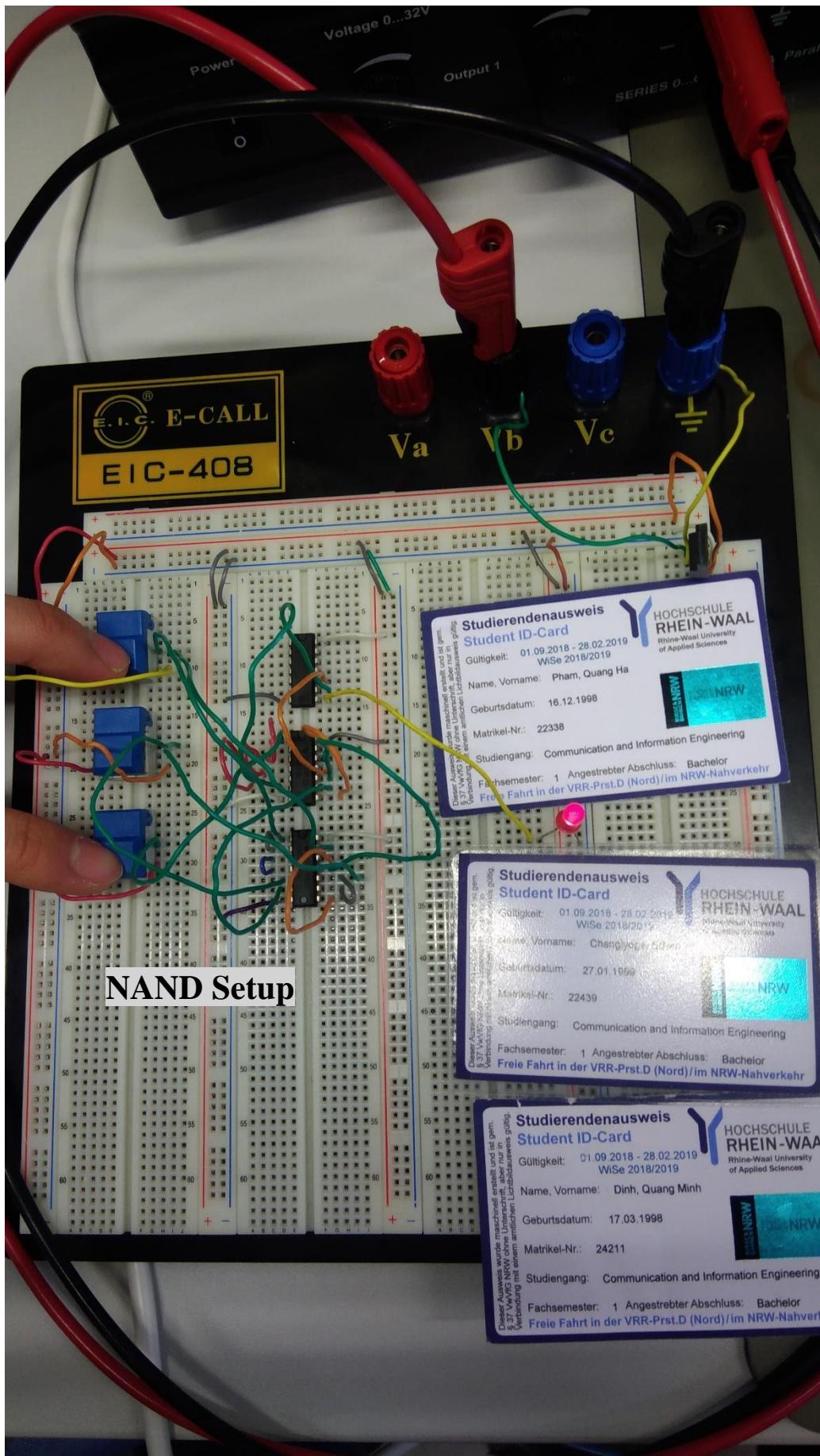


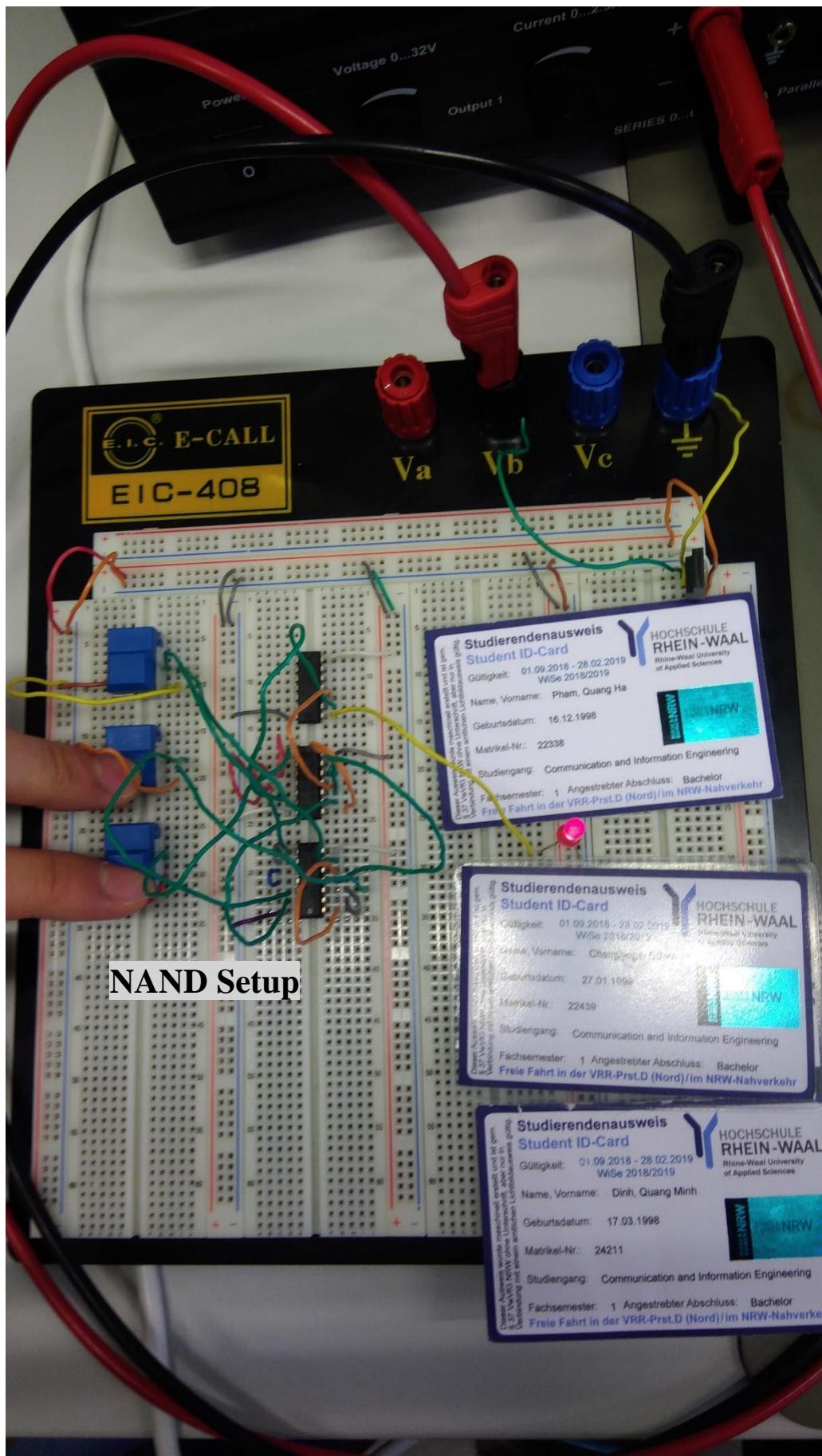


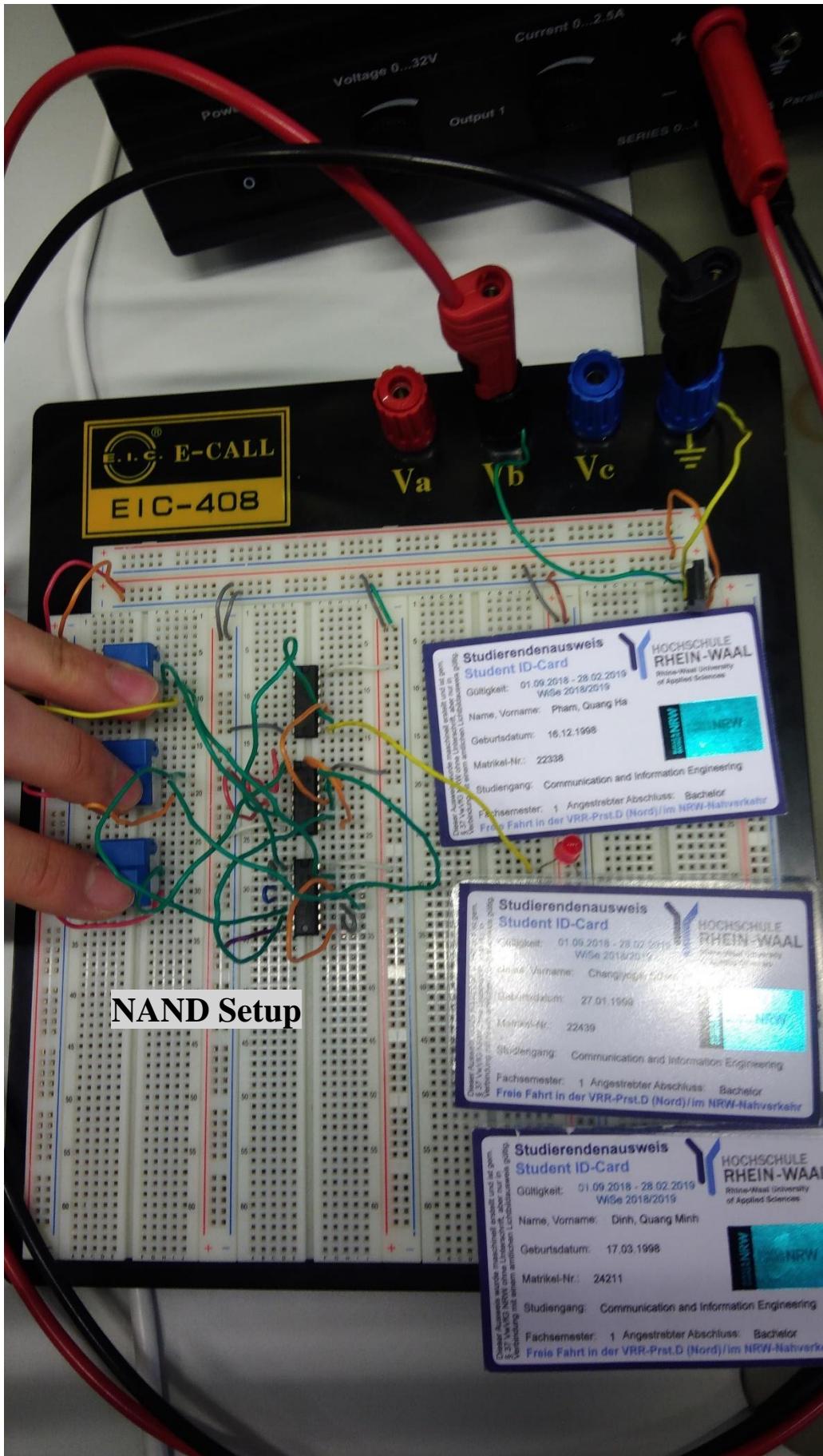


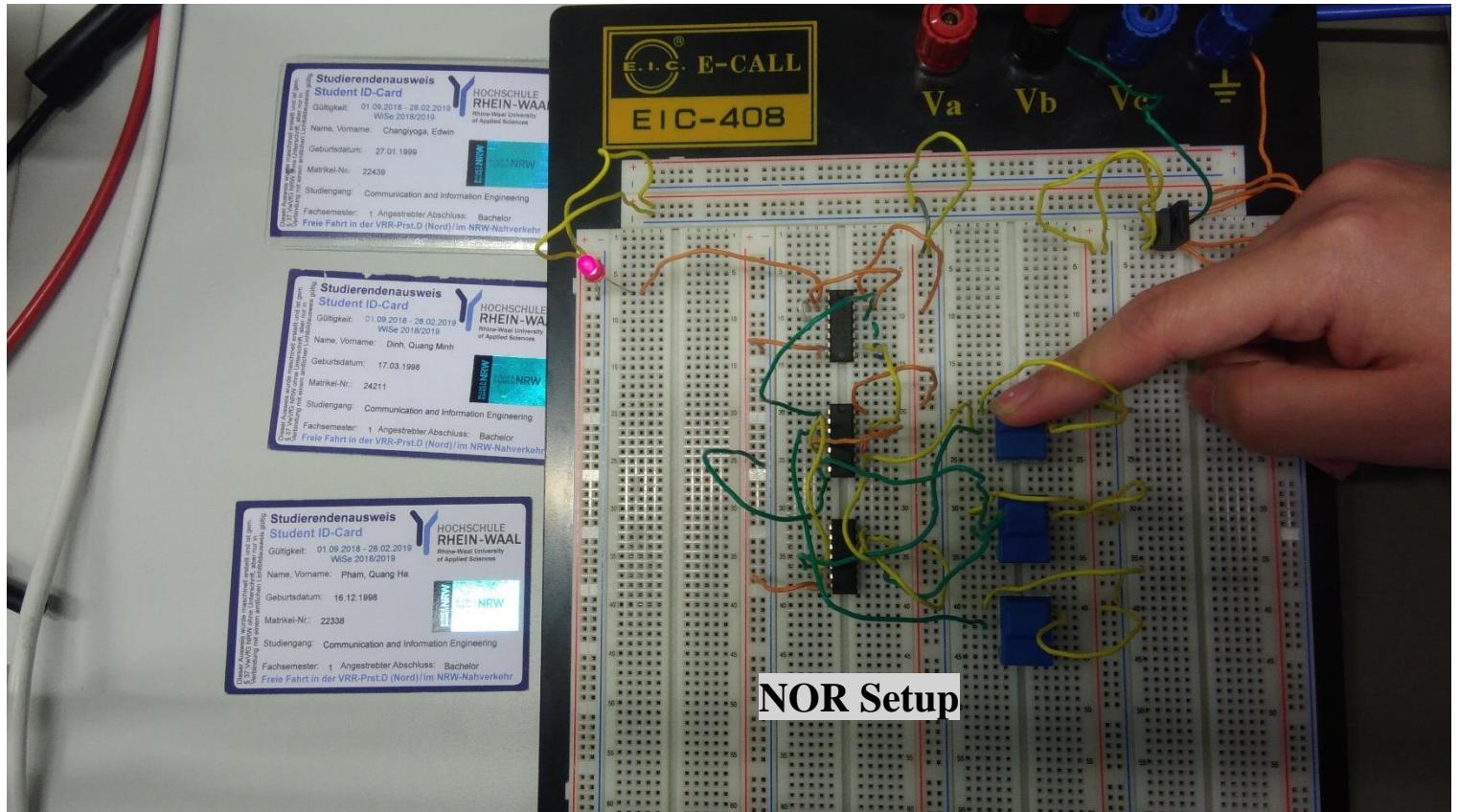
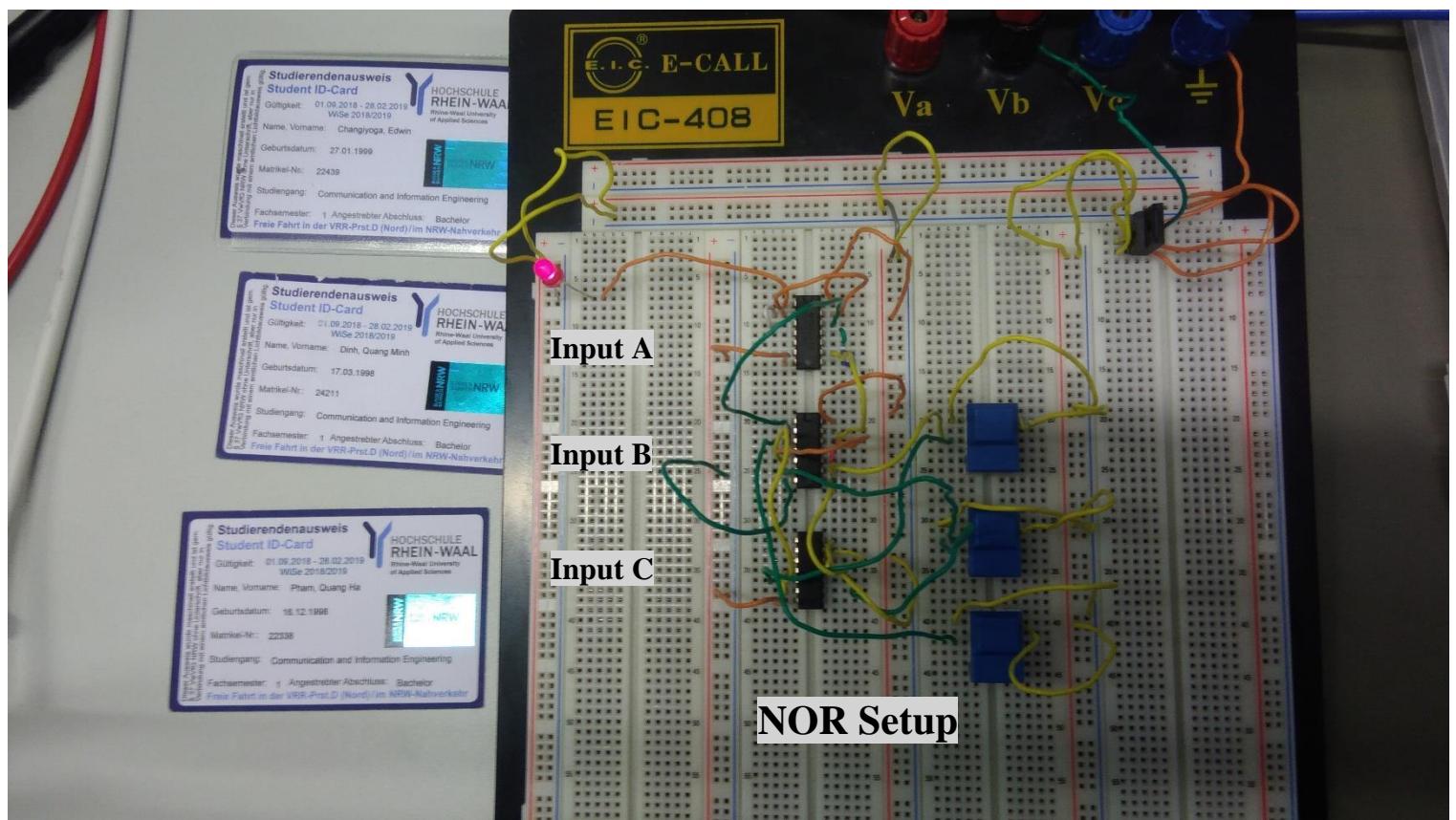


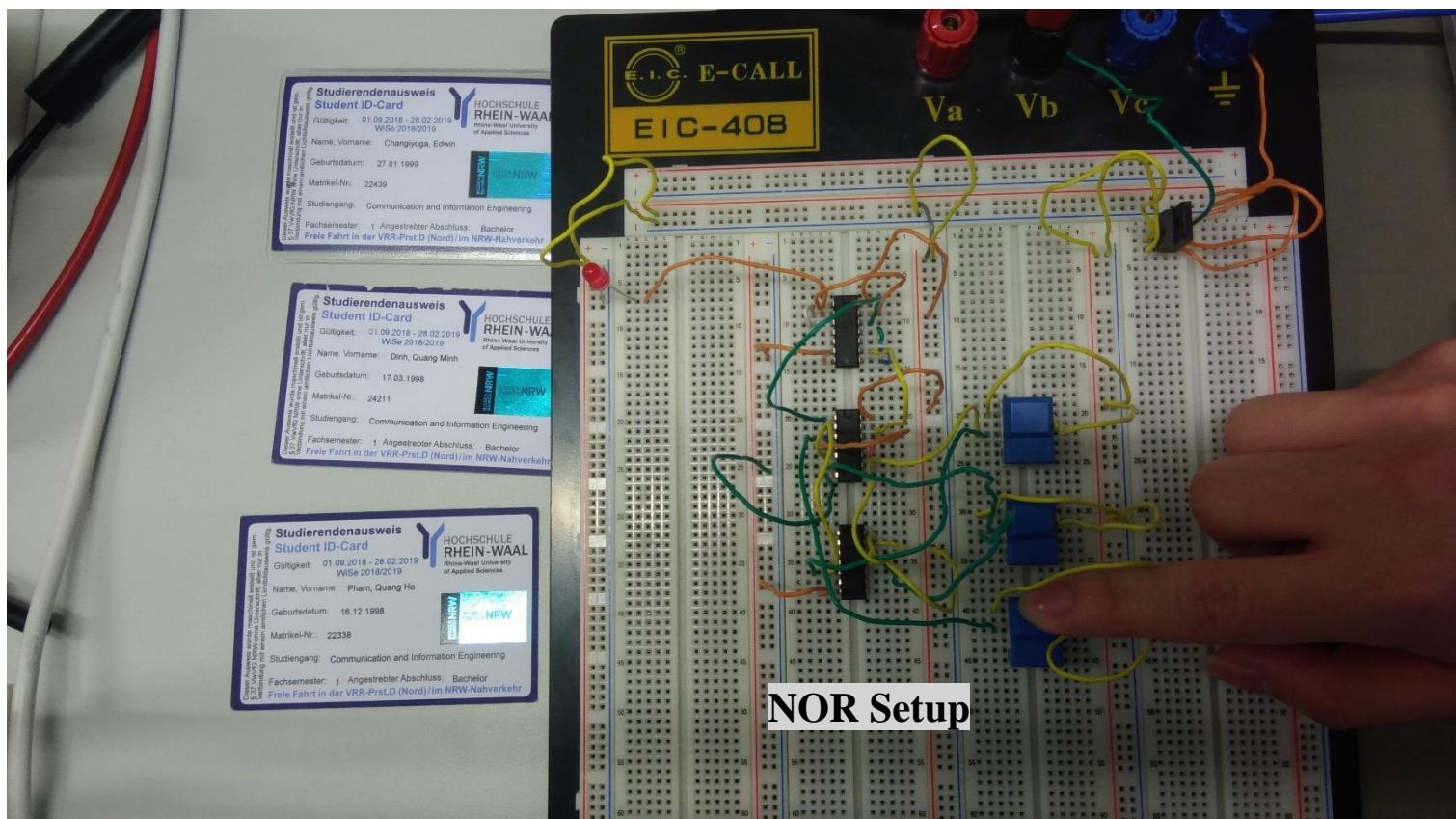
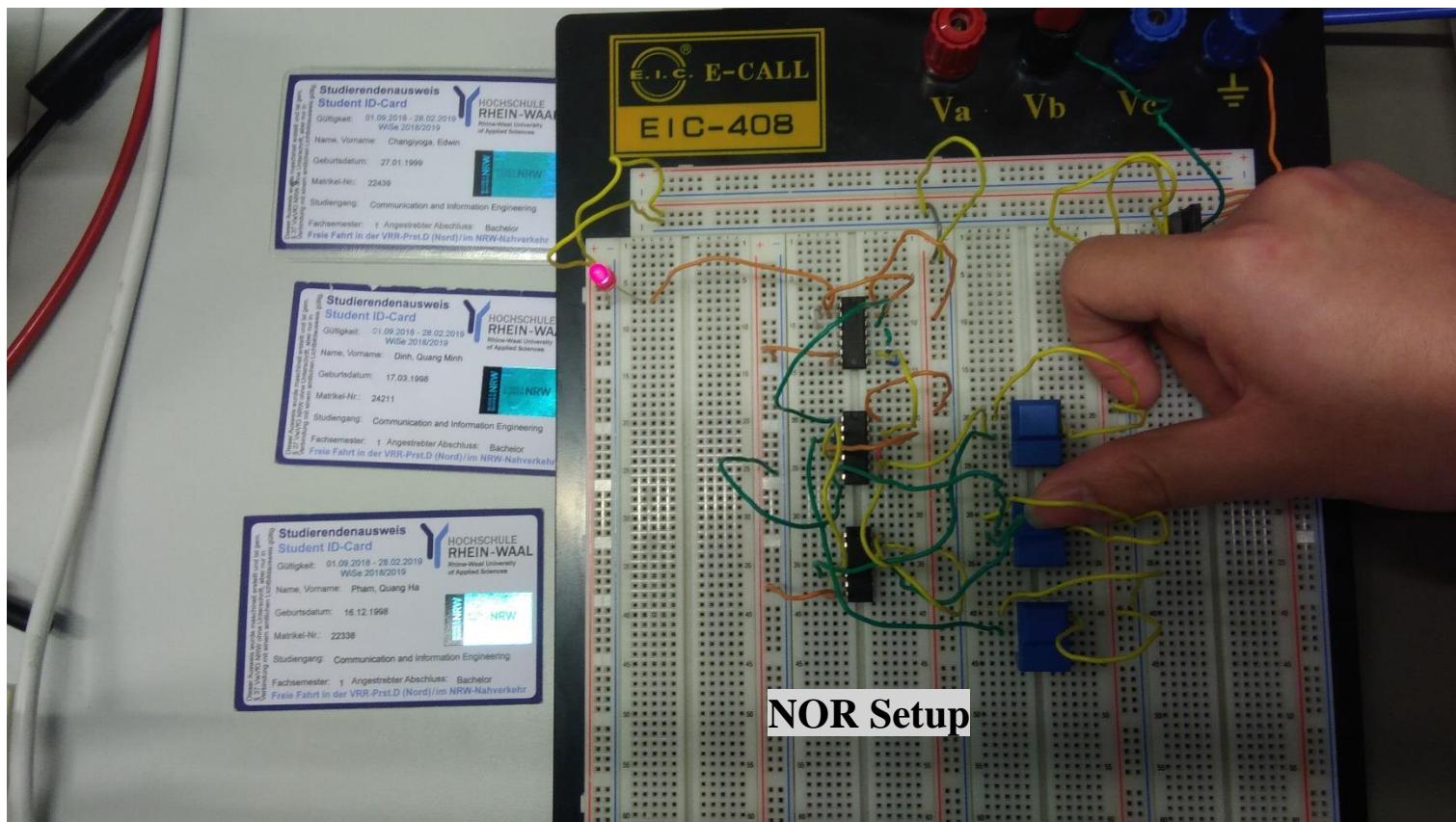


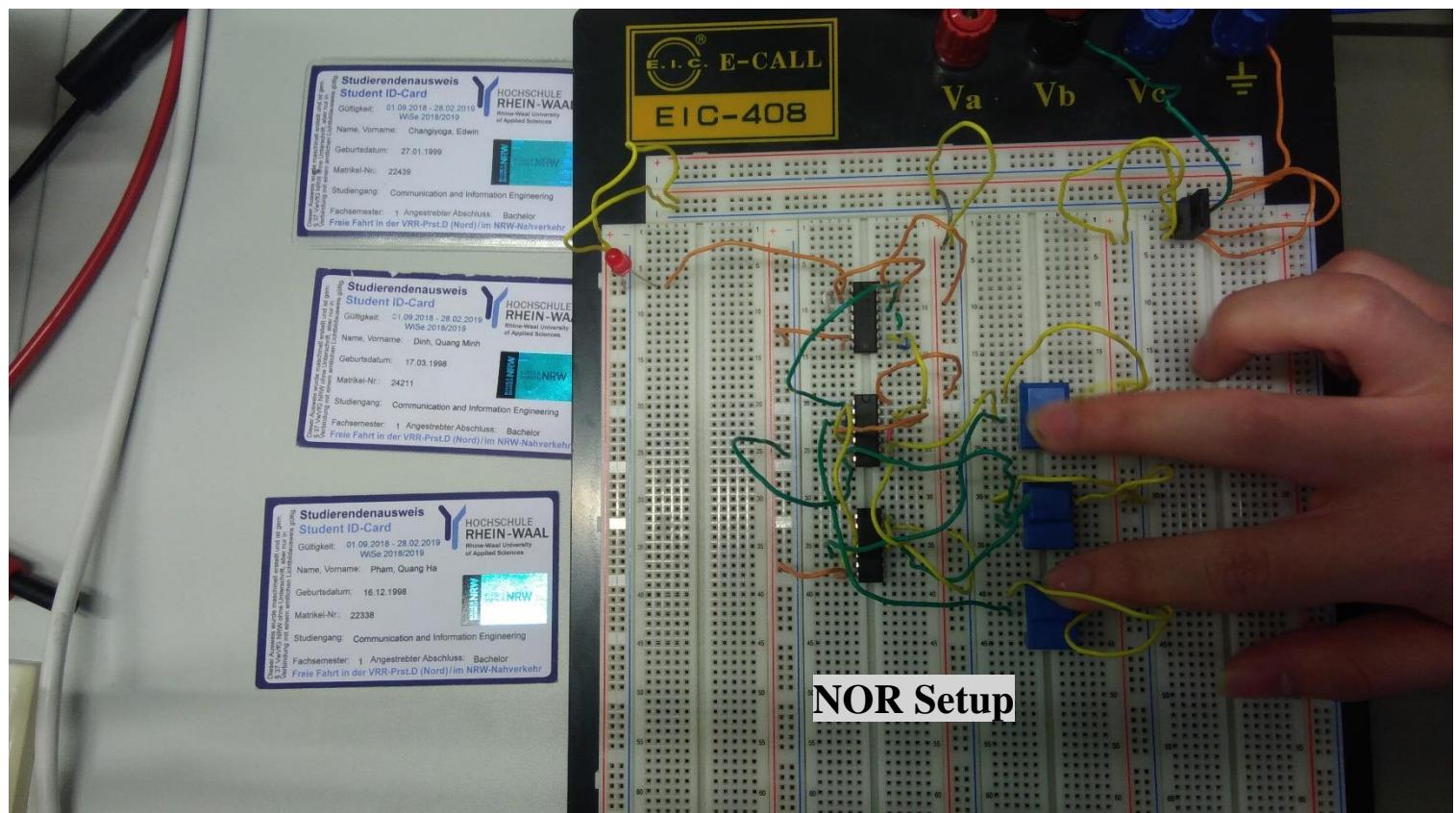
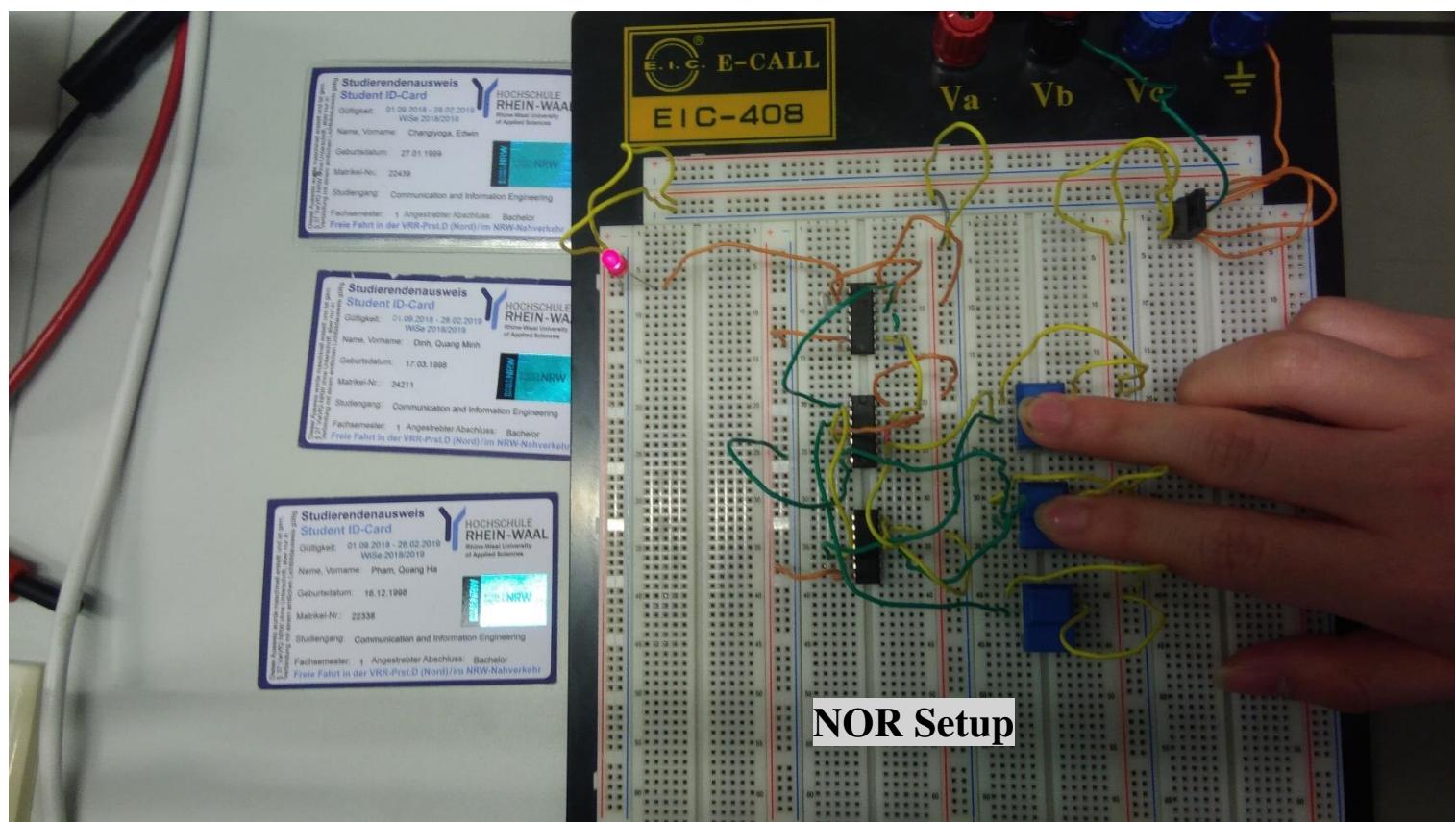


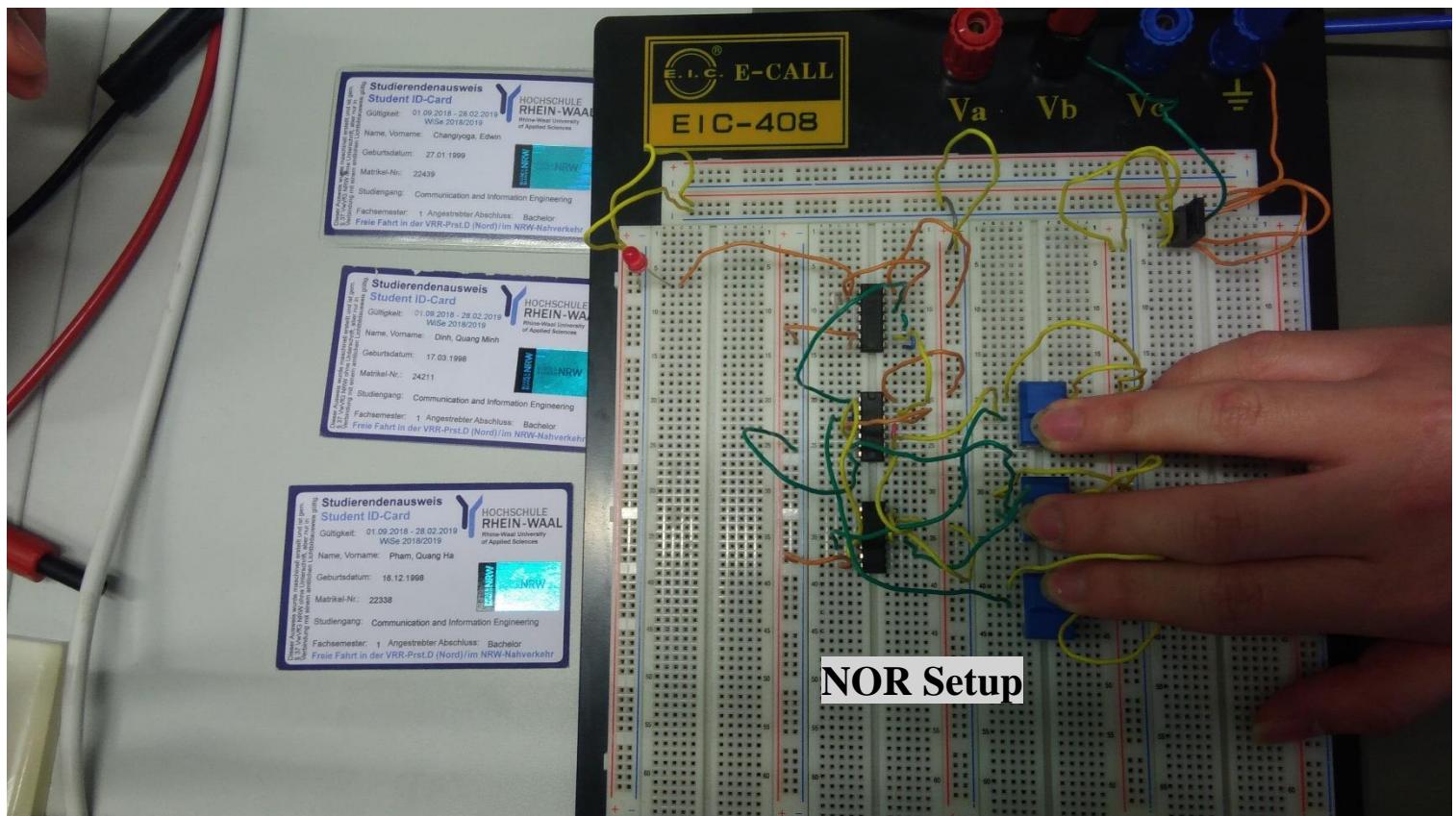
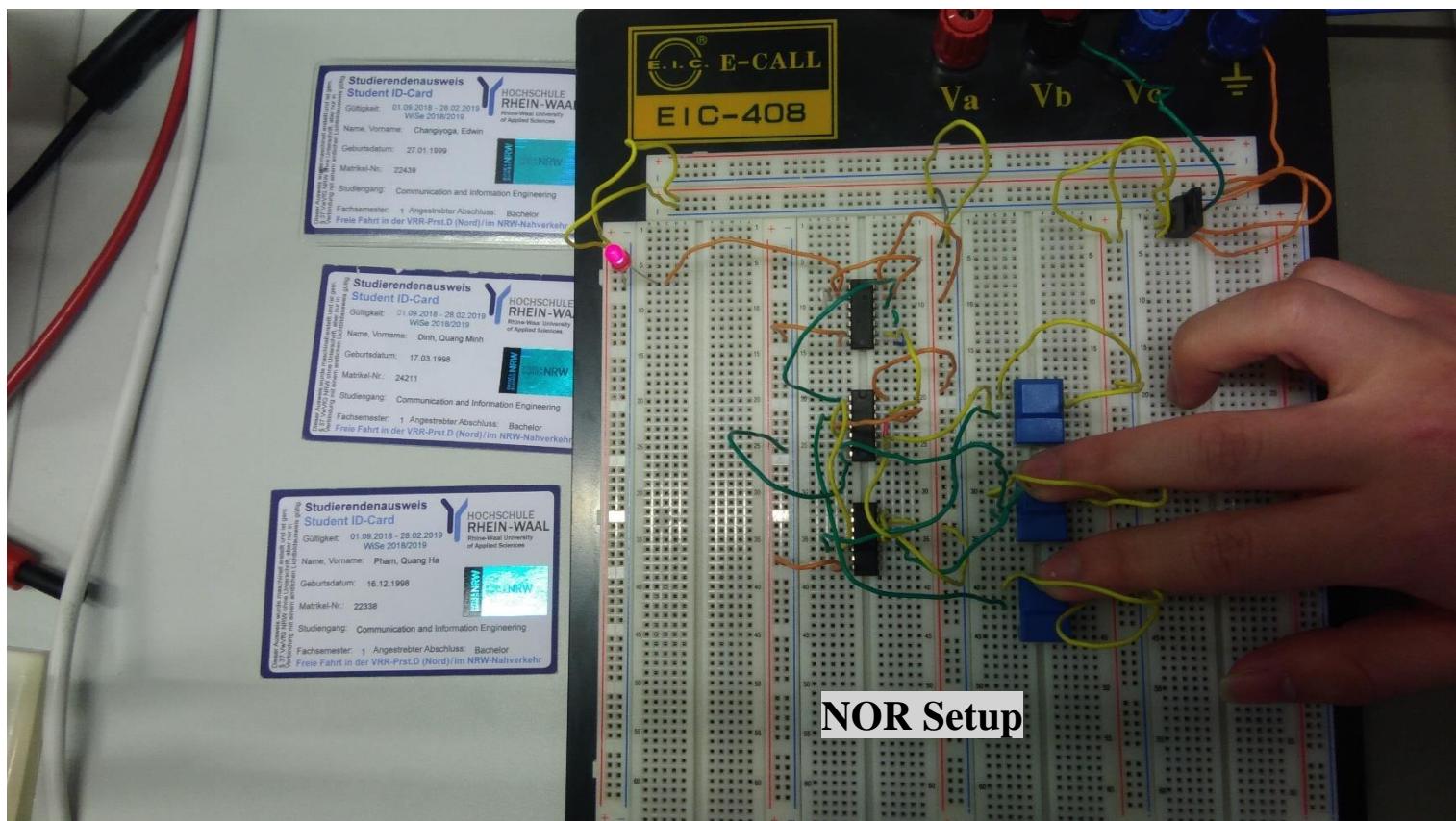












Challenge #6

Abstract:

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

We were able to determine the simplified circuit using the KV diagram:

$$(\neg A \wedge D) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg B \wedge \neg C \wedge D)$$

However, the most challenging problem to us is building the circuit entirely using NAND gates. What we did was replacing every OR, AND, NOT connection in the simplified circuit with the respective NAND construction, we got the information for construction only using NAND gates on Wikipedia.



Figure 1: NOT gate with NAND construction

[https://en.wikipedia.org/wiki/Inverter_\(logic_gate\)](https://en.wikipedia.org/wiki/Inverter_(logic_gate))

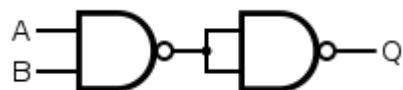


Figure 2: AND gate with NAND construction

https://en.wikipedia.org/wiki/AND_gate

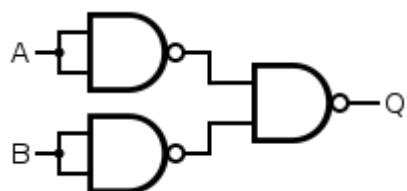
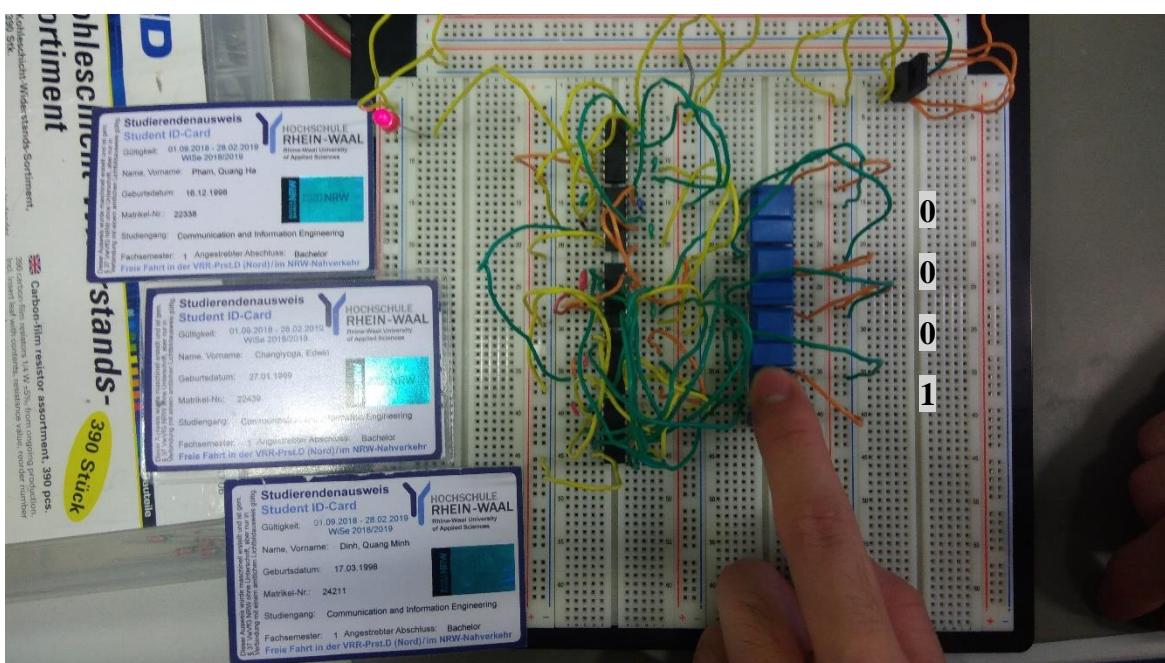
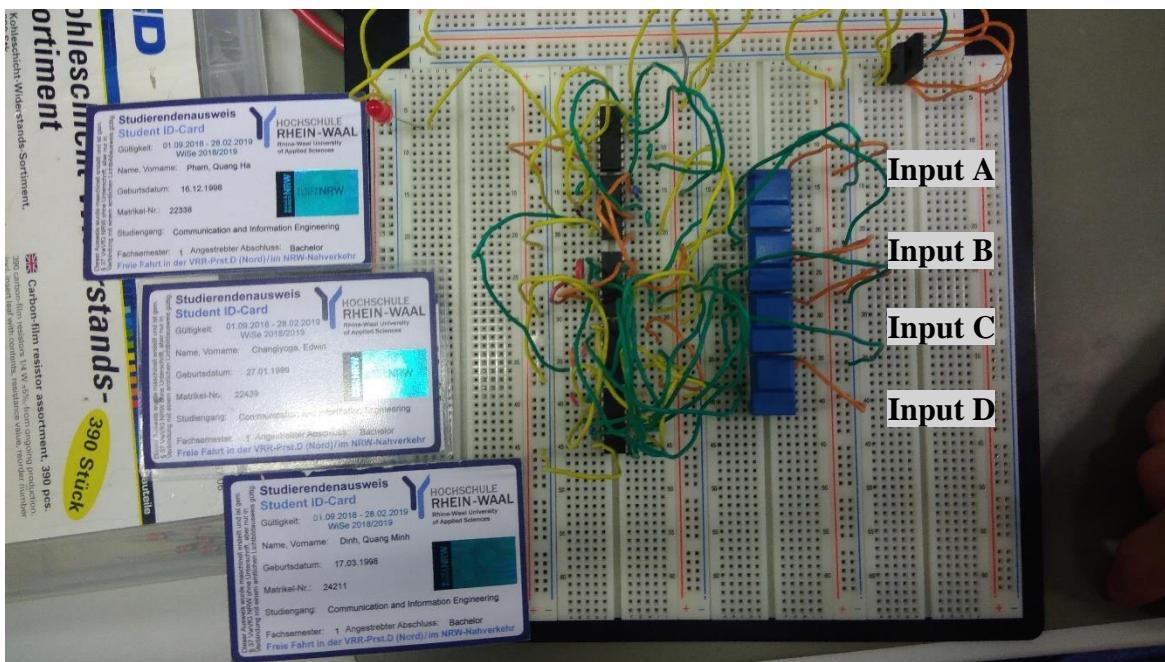
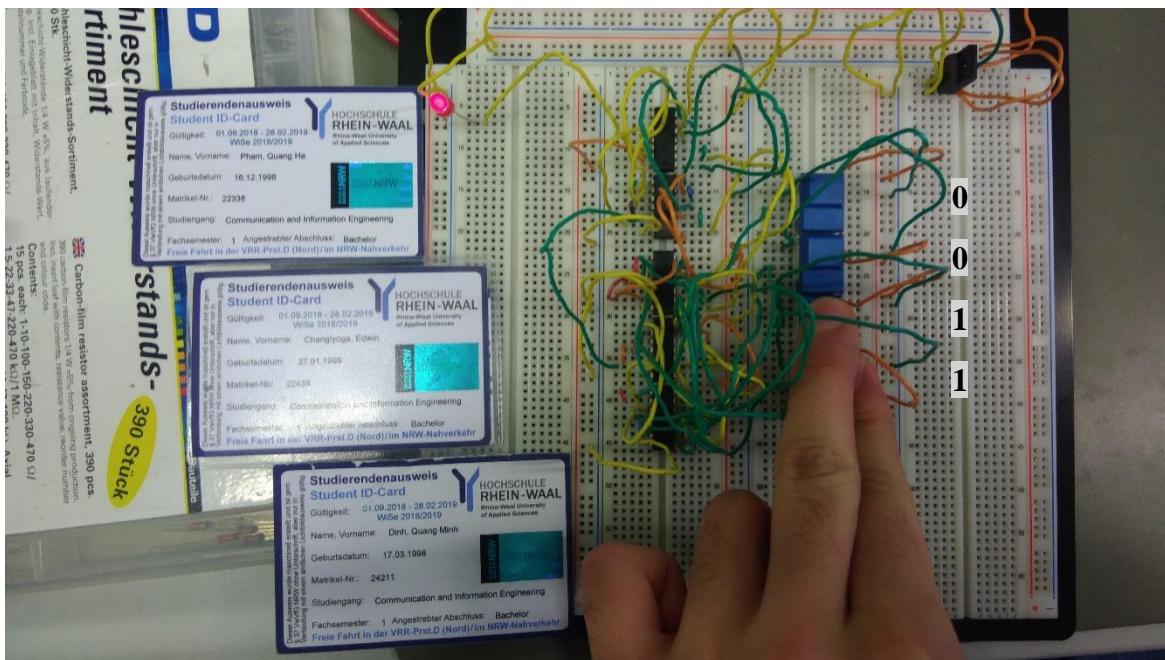
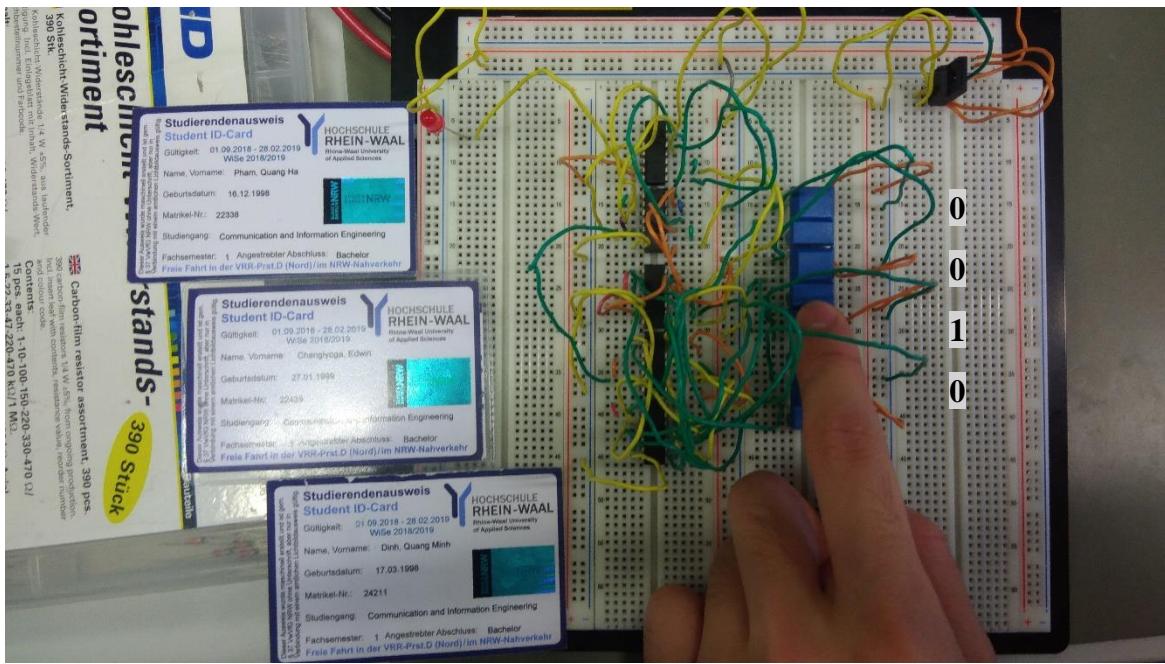


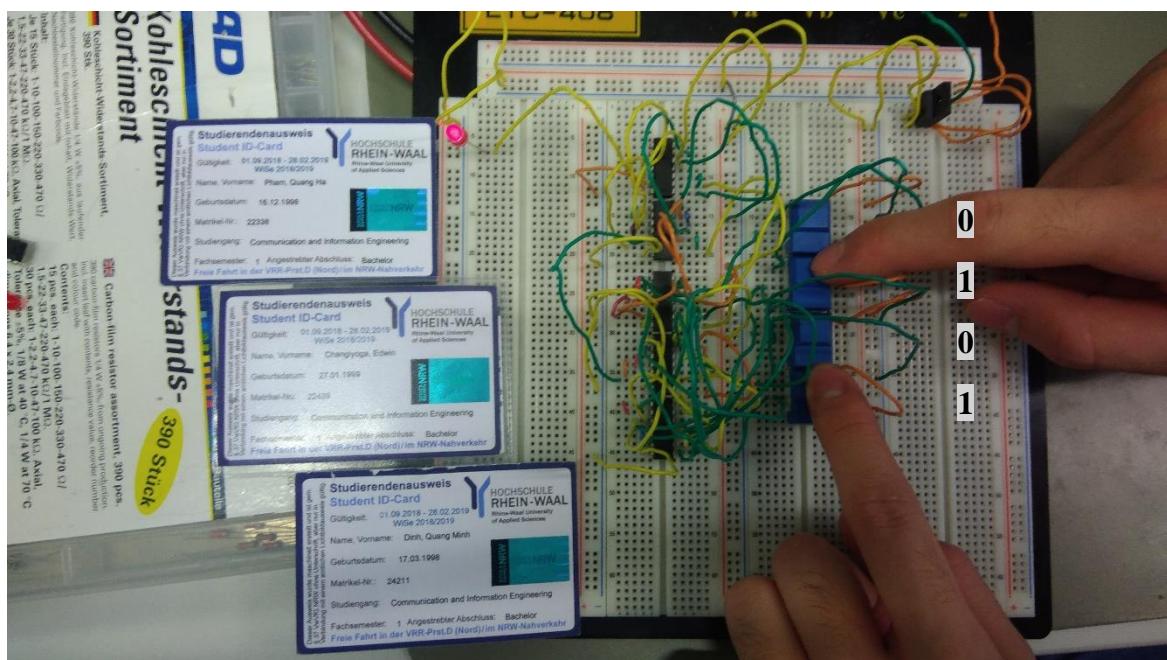
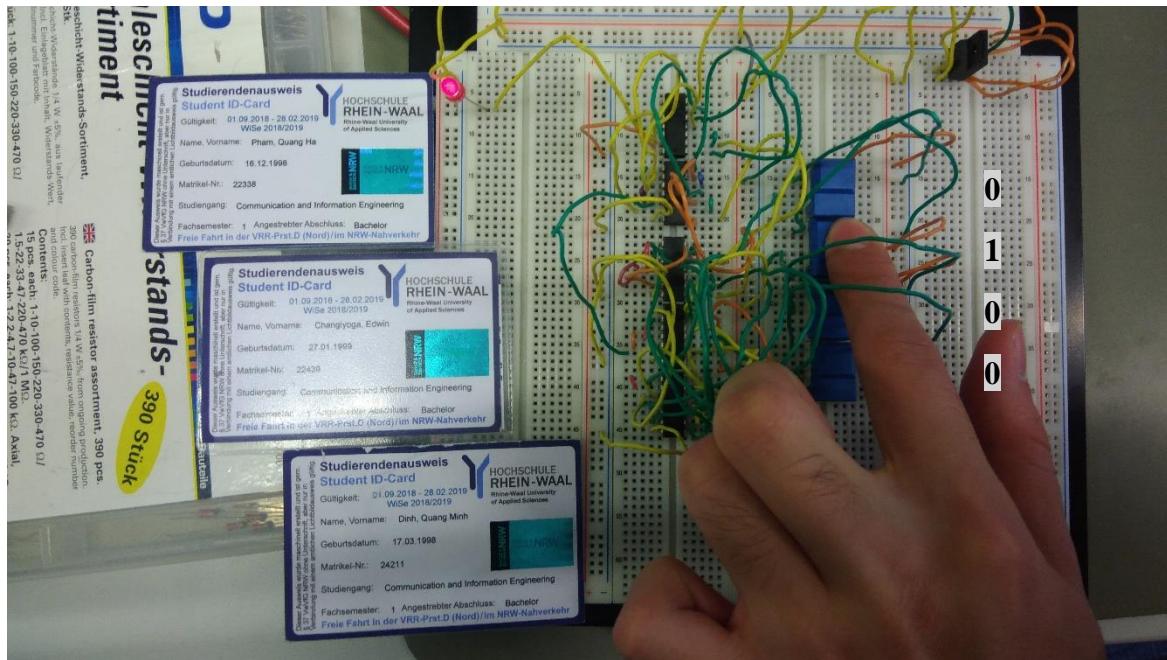
Figure 3: OR gate with NAND construction

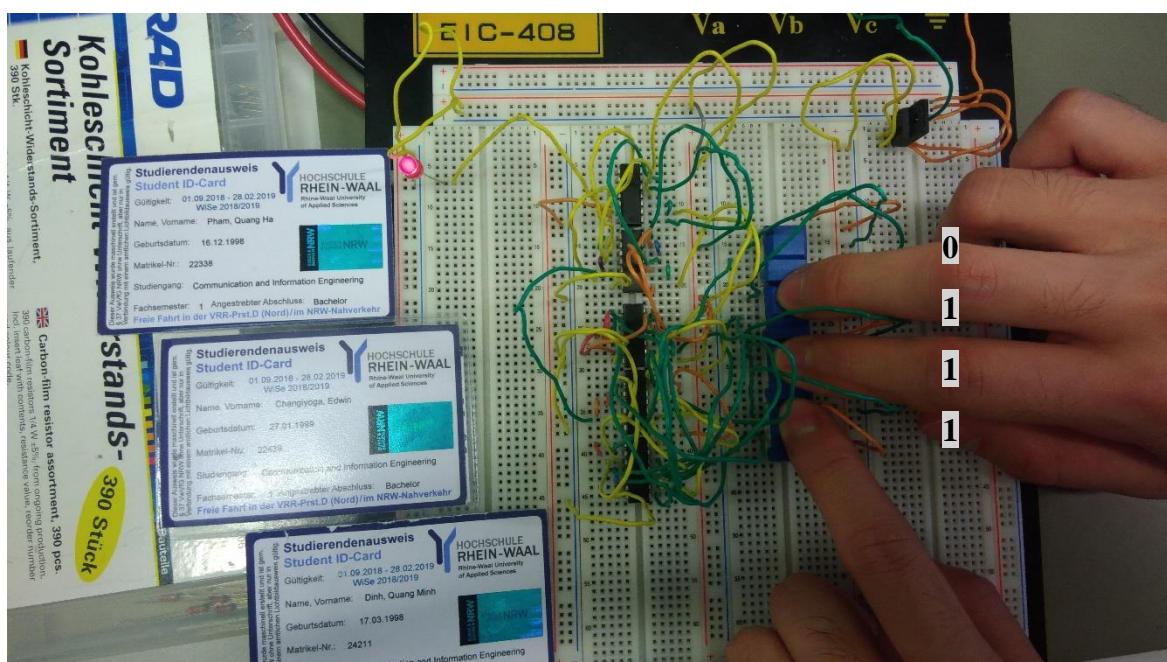
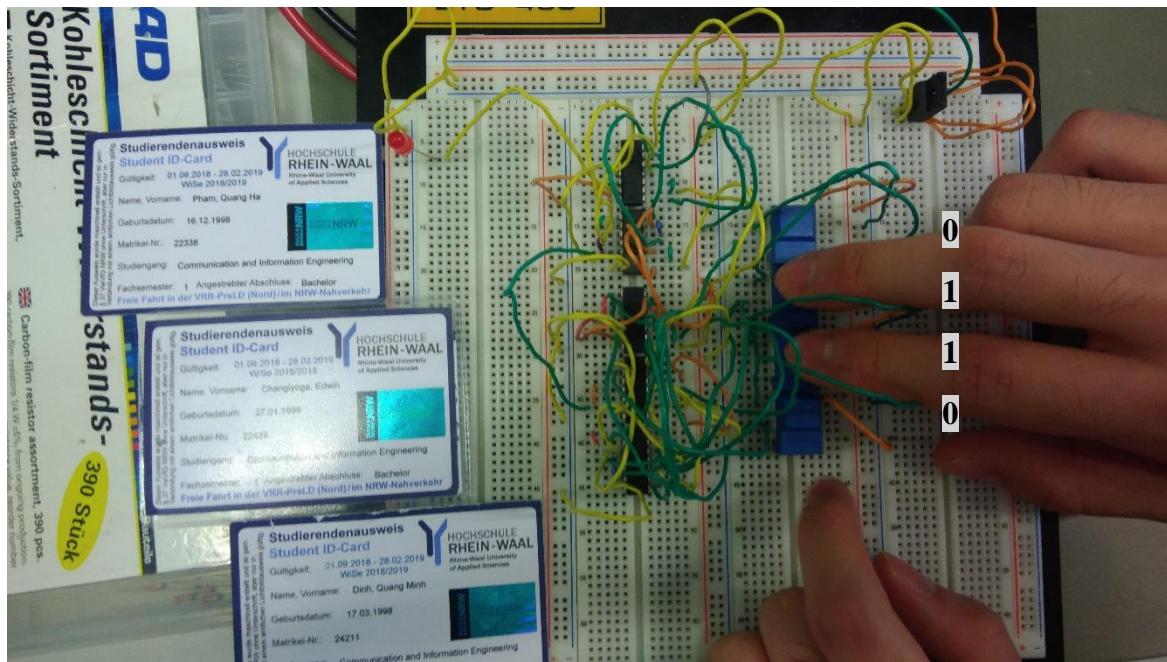
https://en.wikipedia.org/wiki/OR_gate

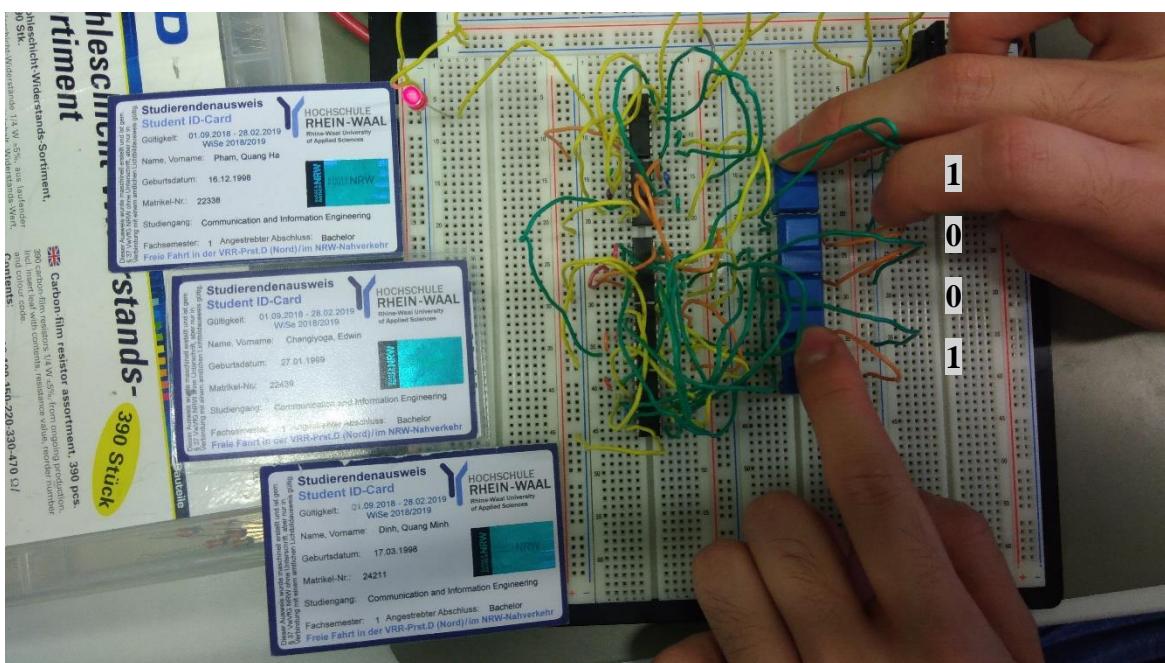
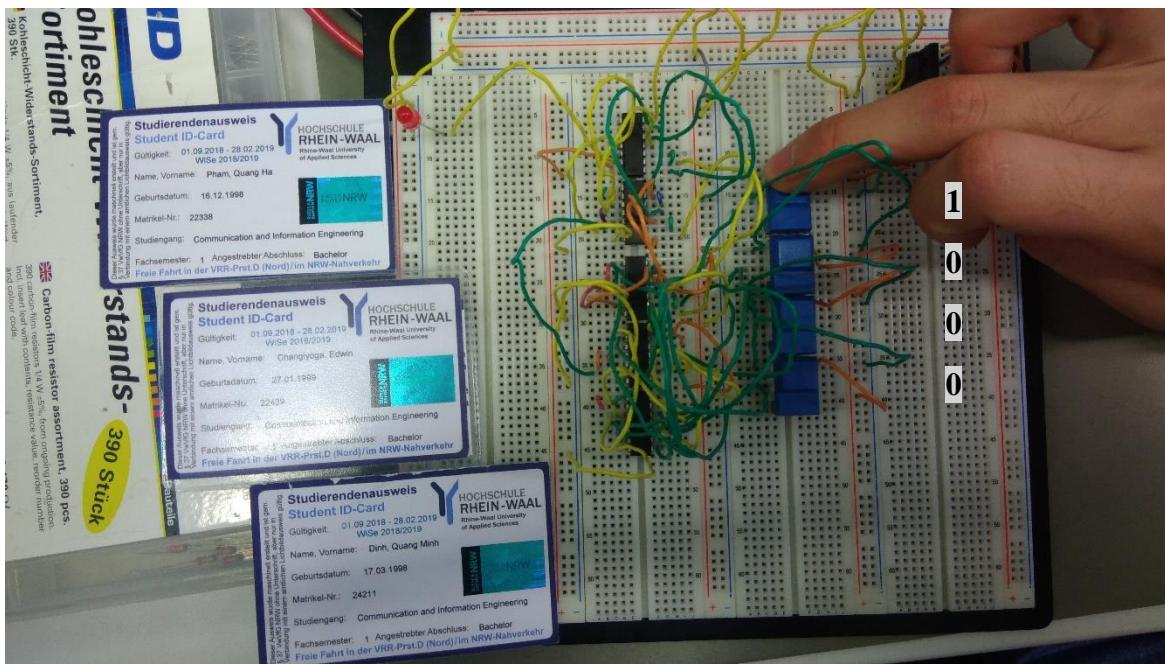
Pictures:

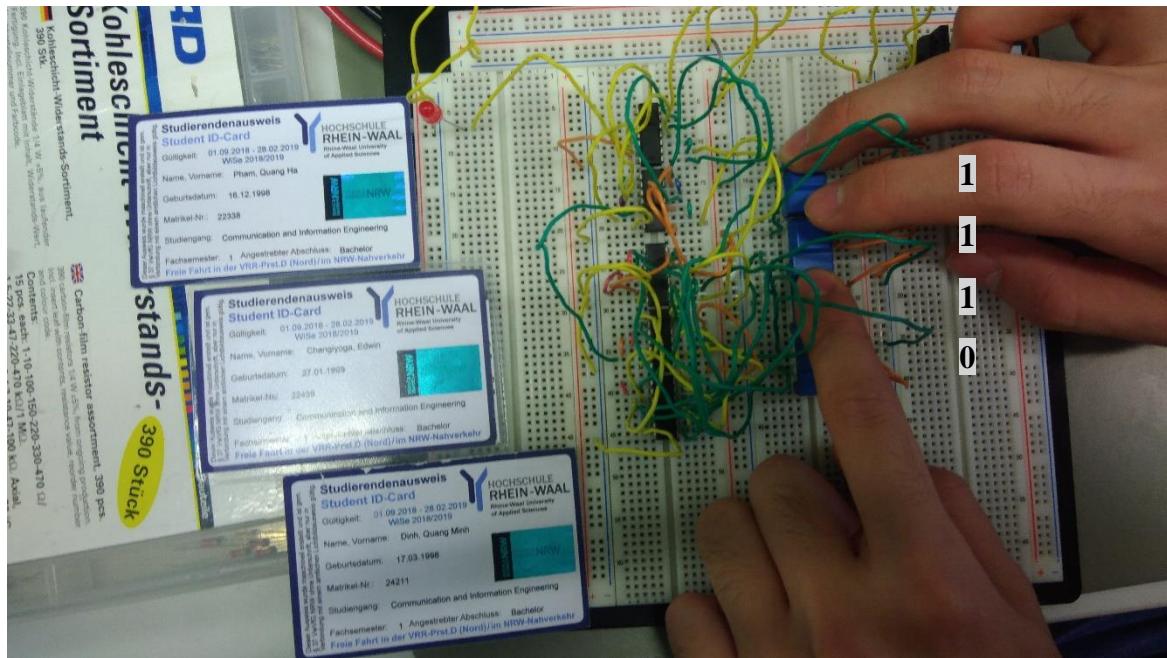












Result:

No.	A	B	C	D	Y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	X
12	1	1	0	0	X
13	1	1	0	1	X
14	1	1	1	0	X
15	1	1	1	1	X

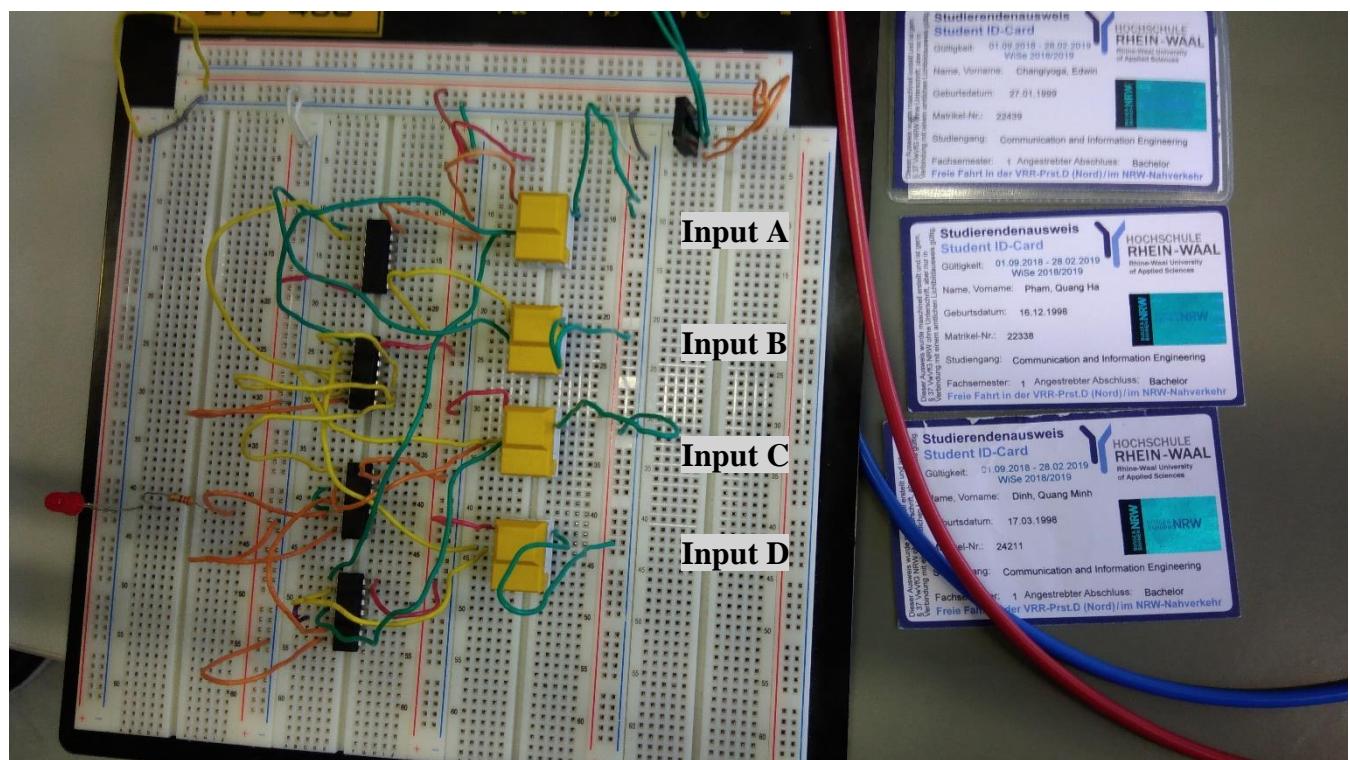
Challenge #7

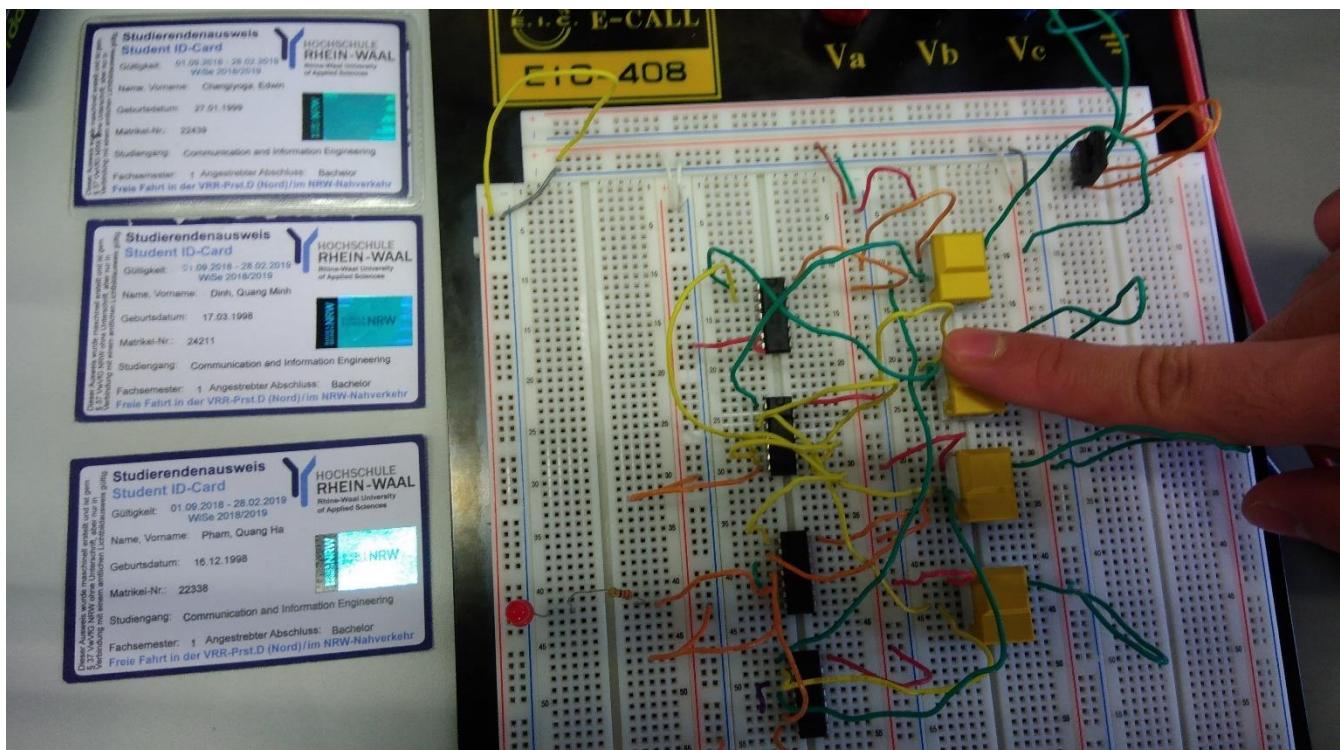
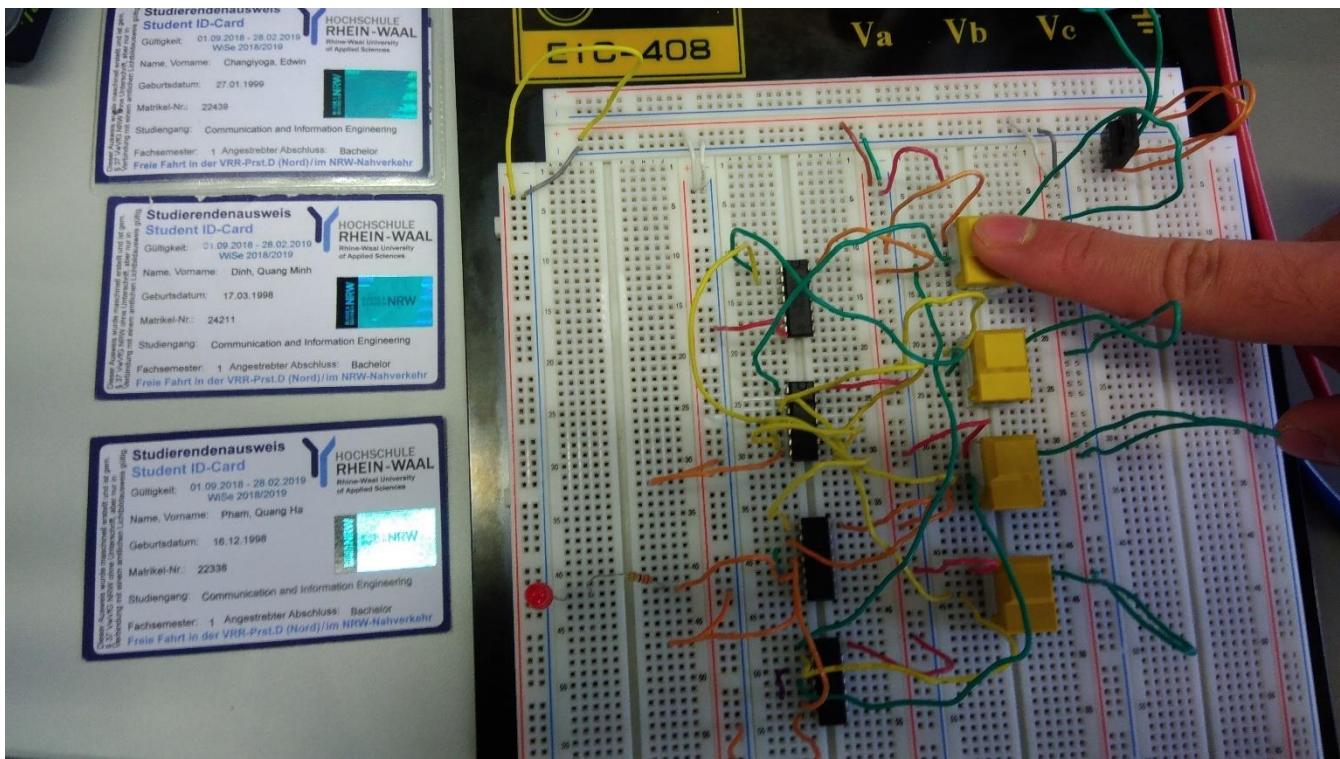
Abstract:

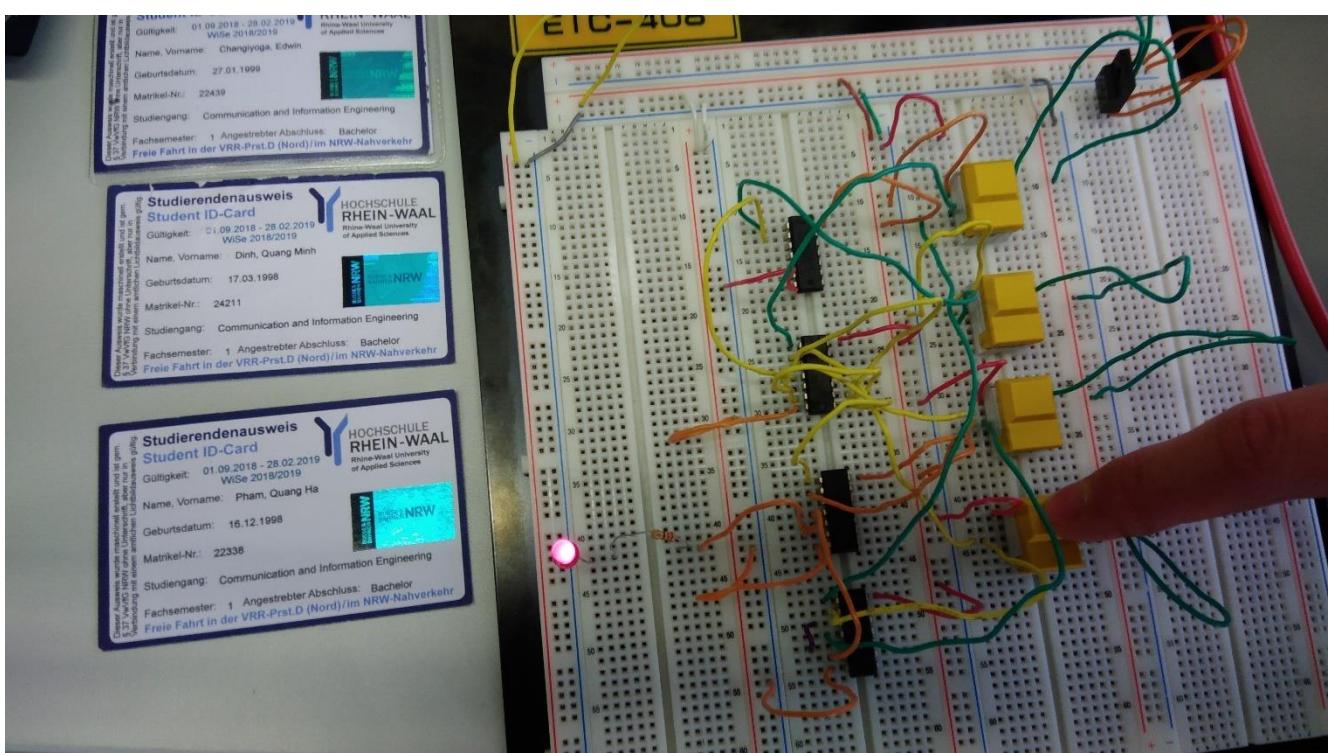
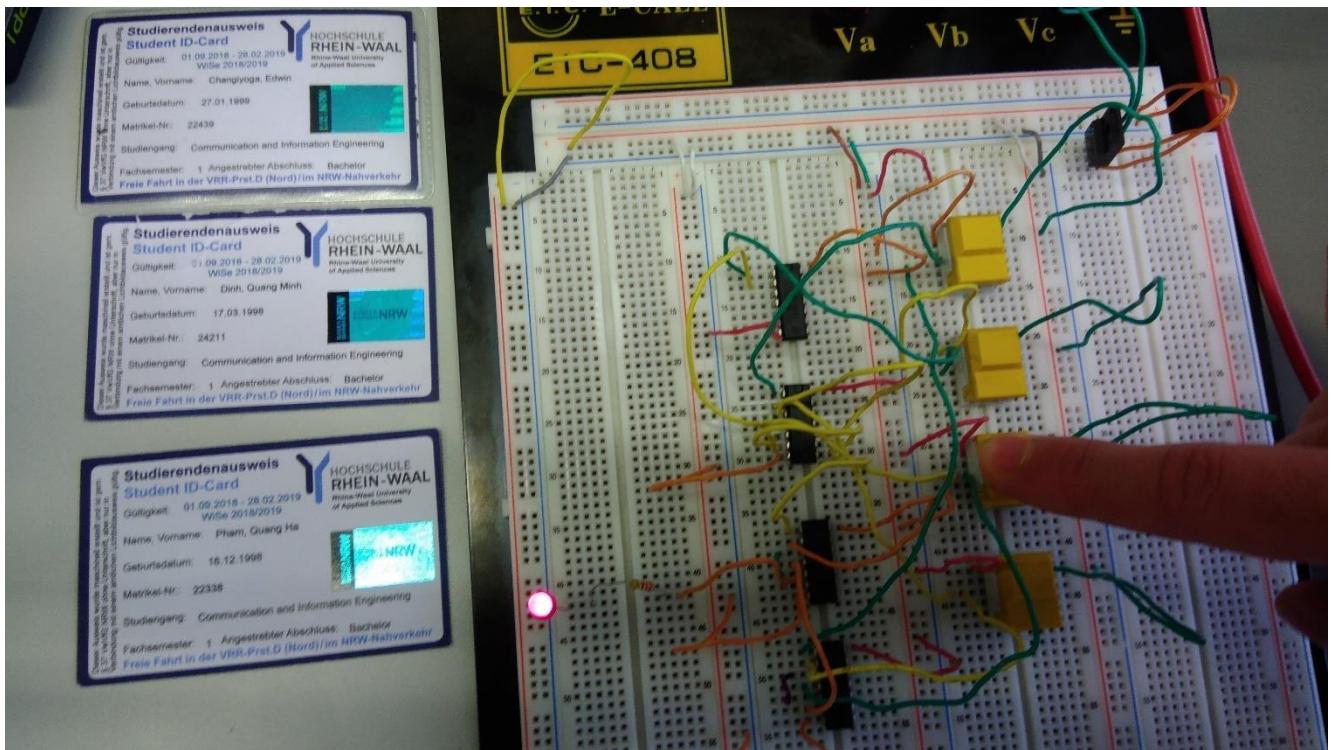
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

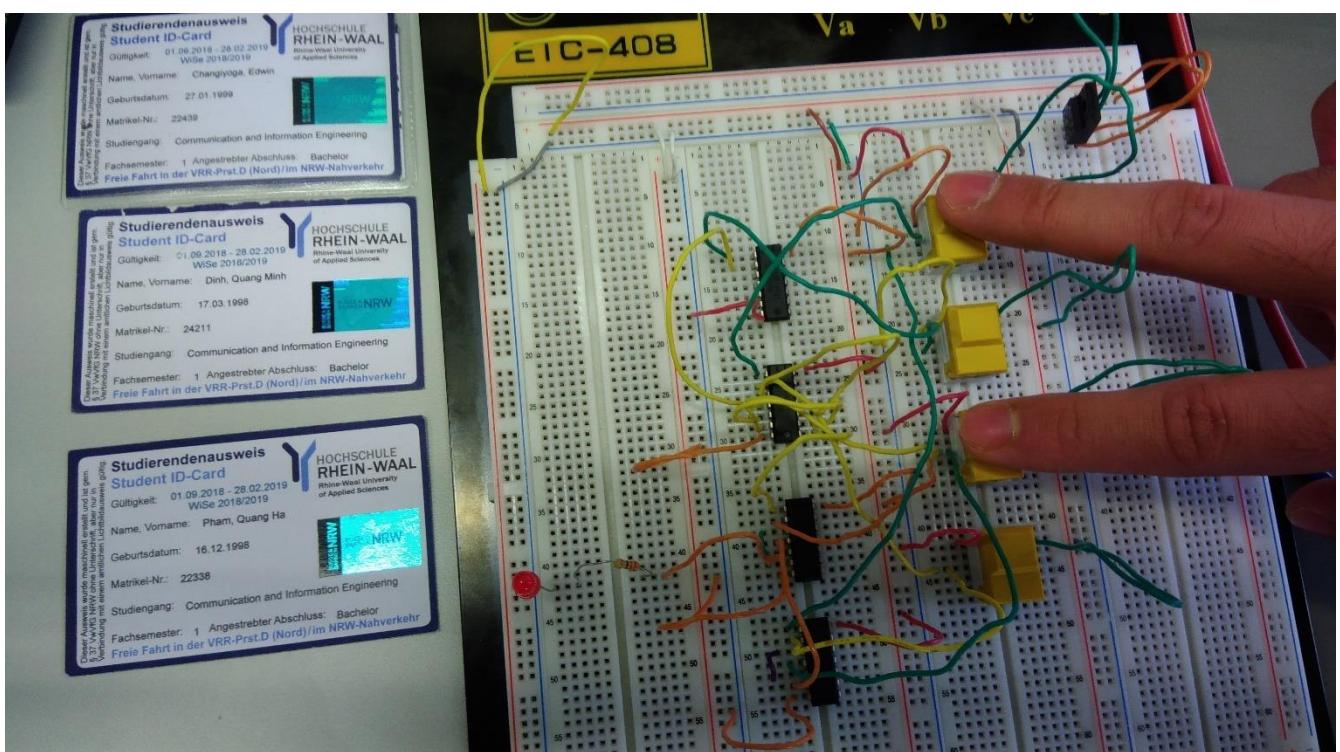
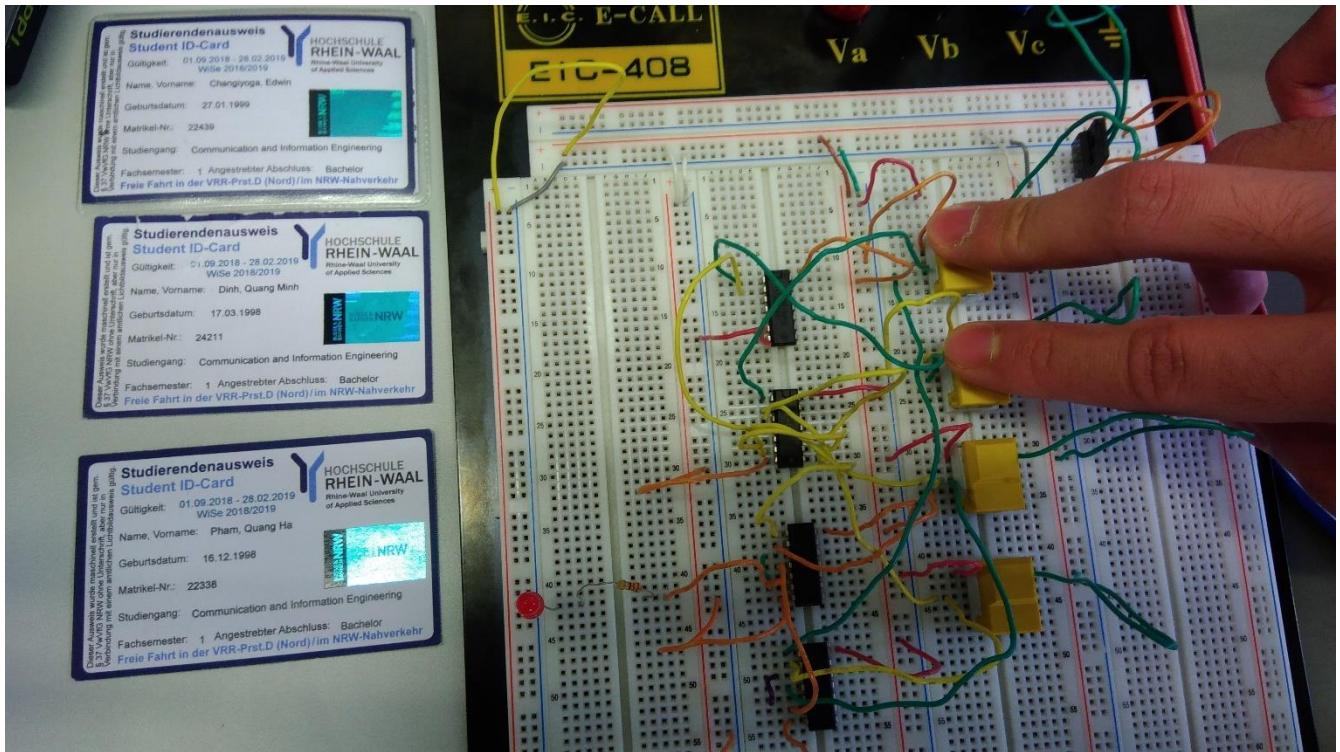
This circuit compares two binary numbers in order to find out whether one binary number is equal, less than or greater than the other binary number.

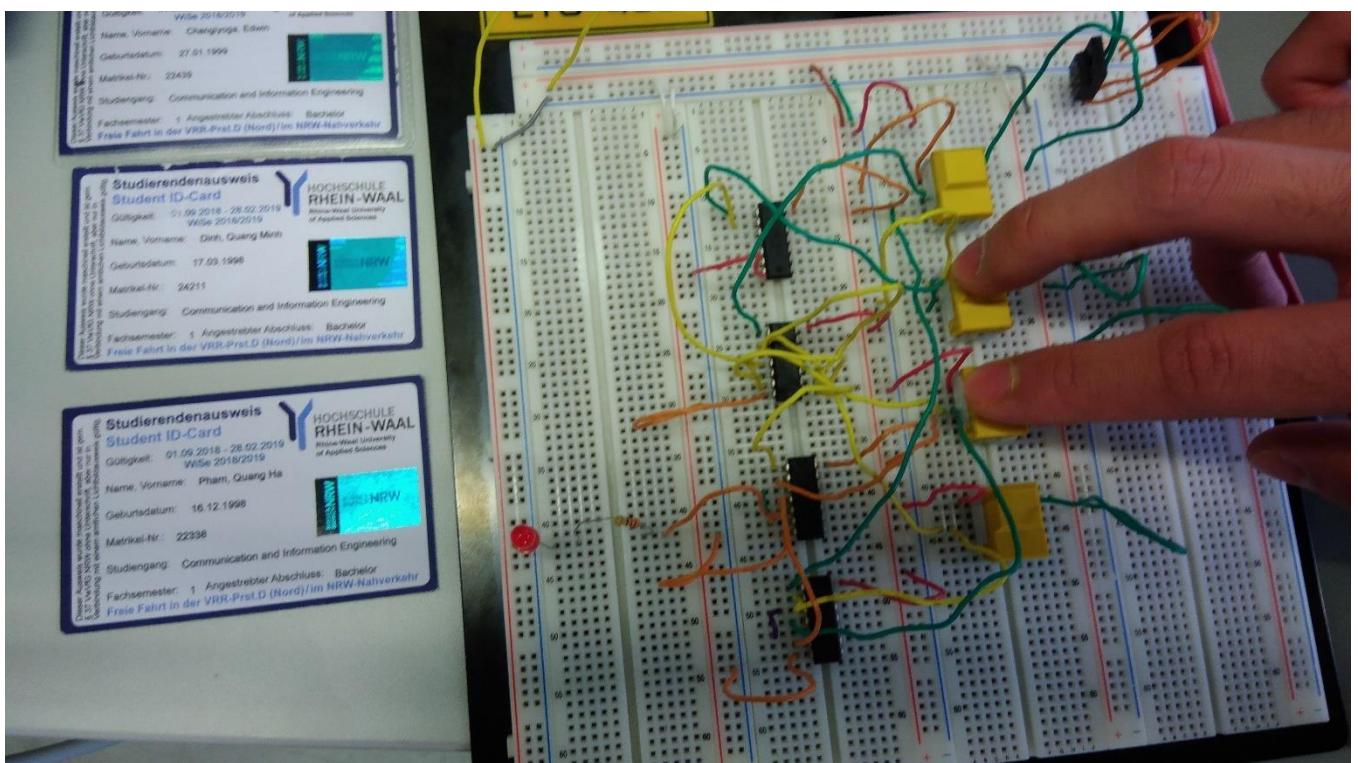
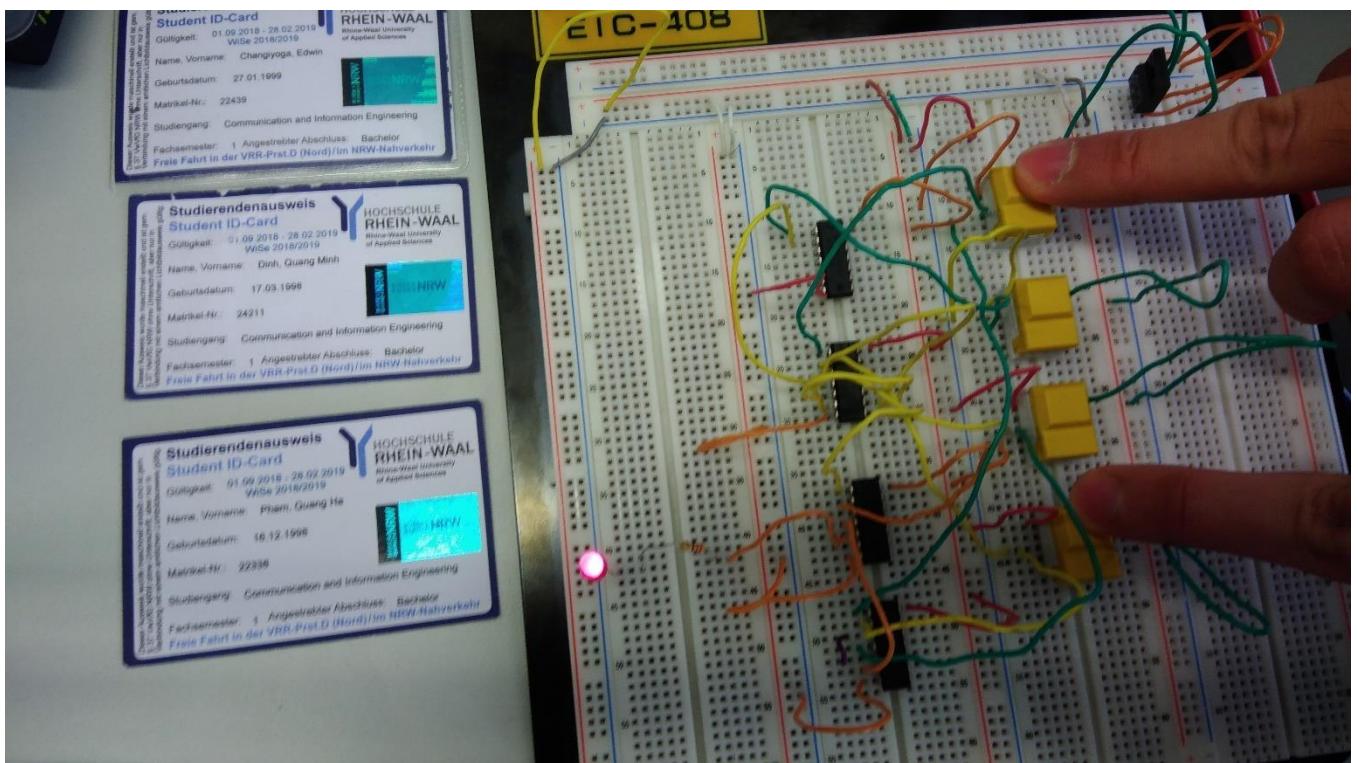
Picture:

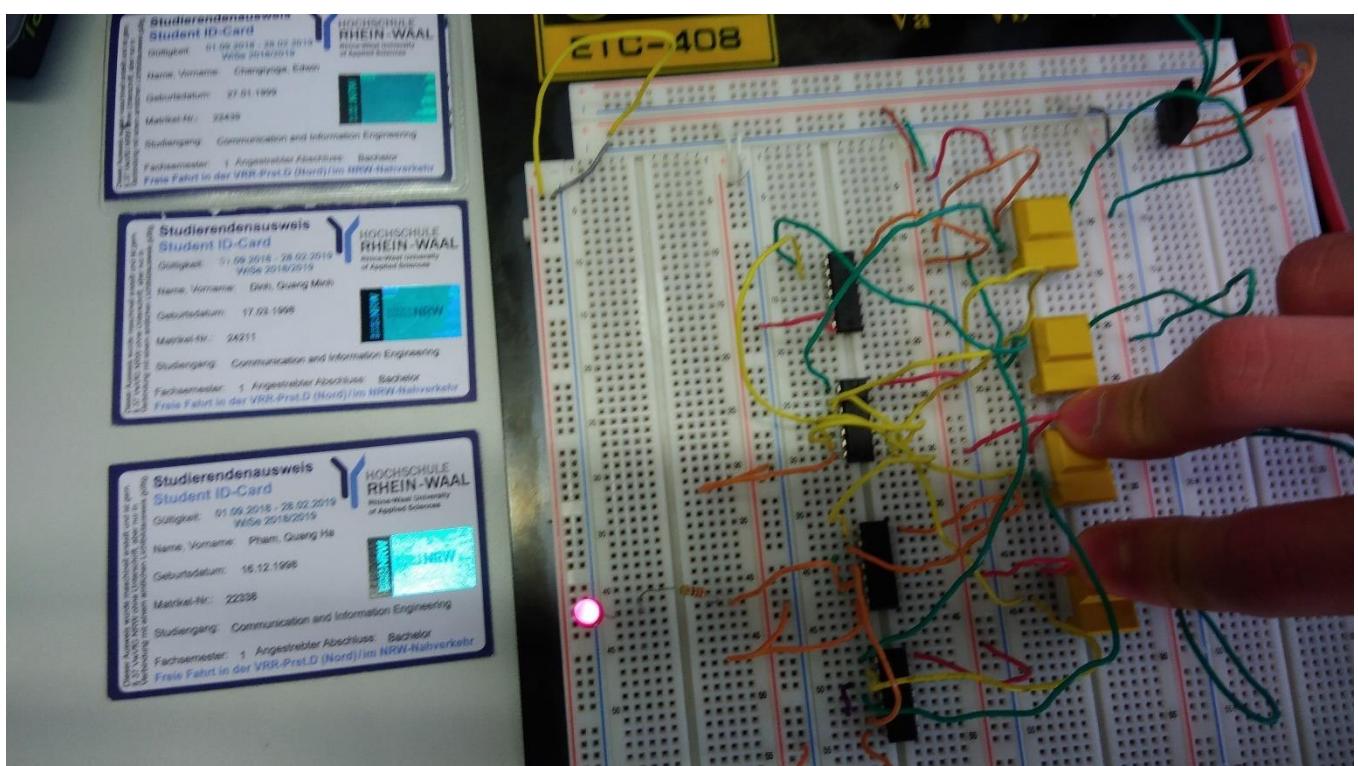
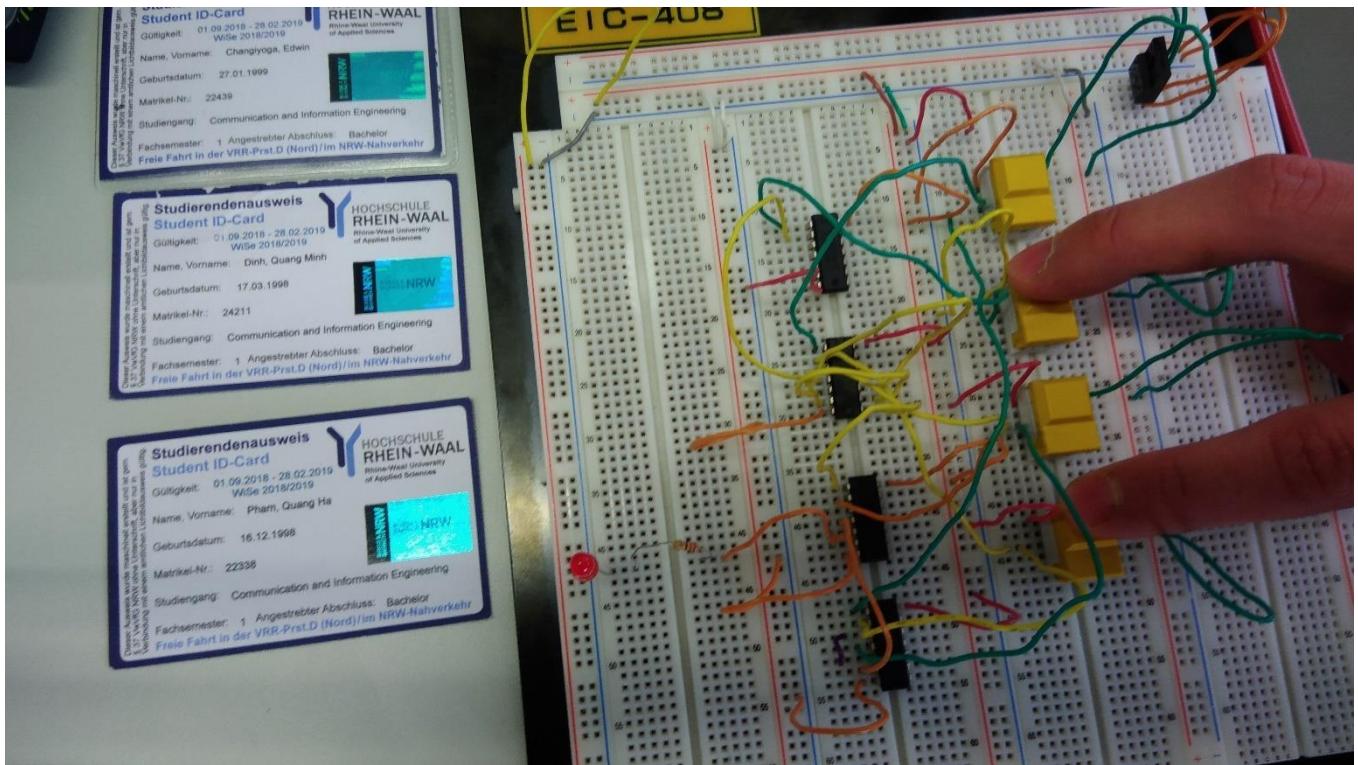


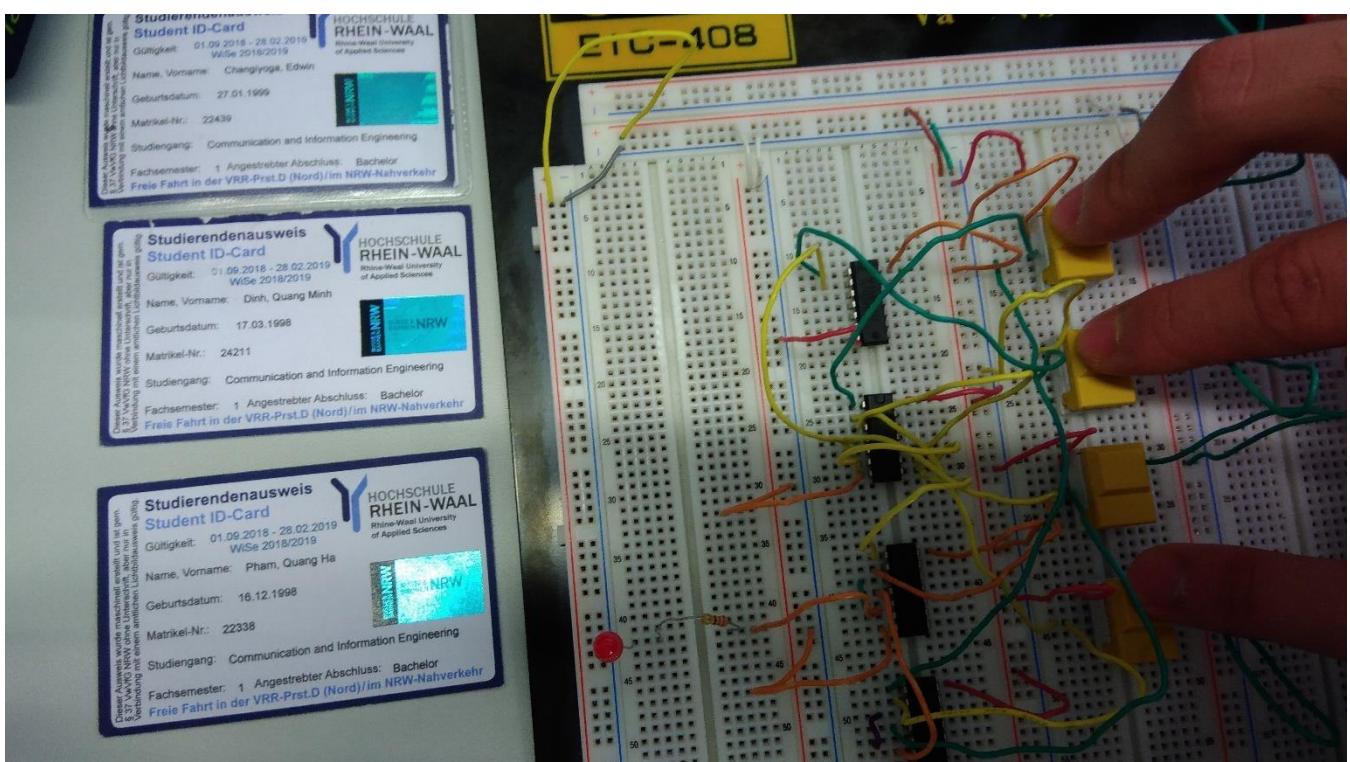
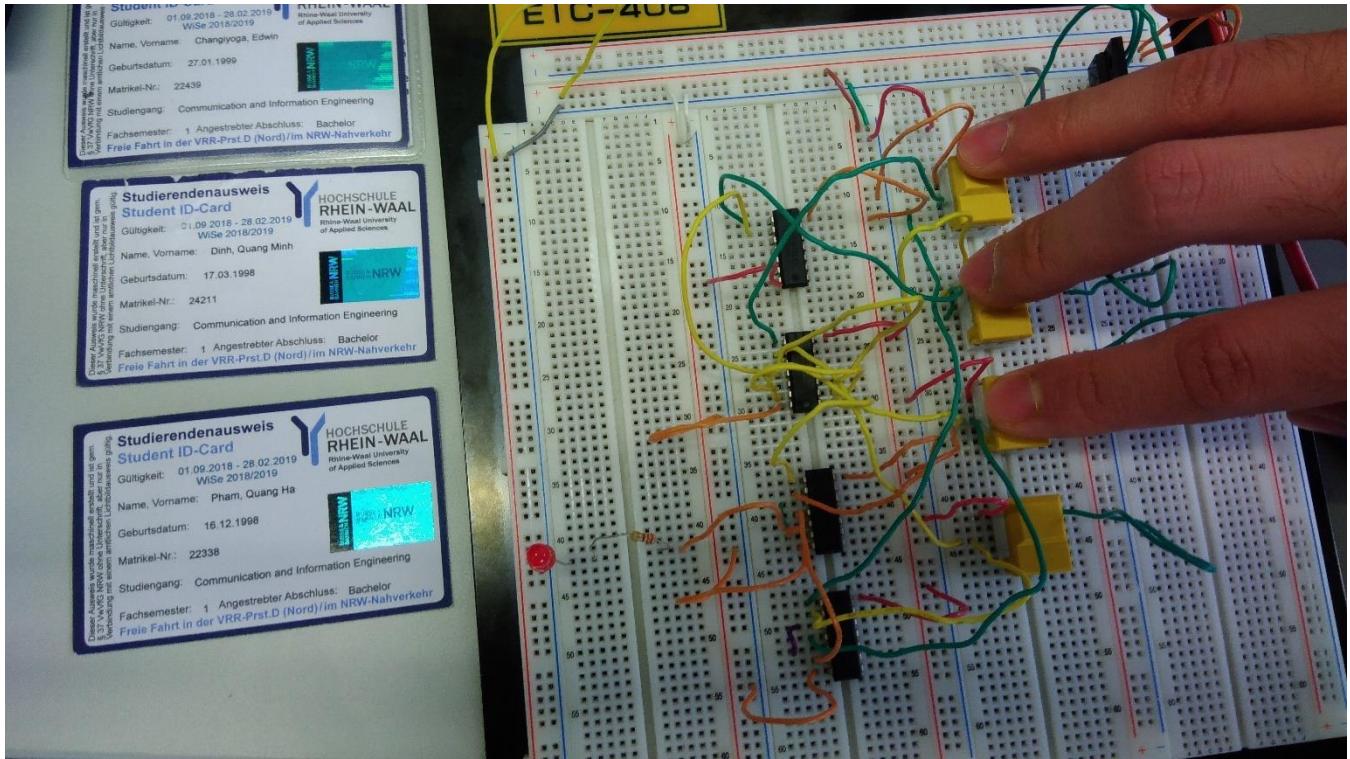


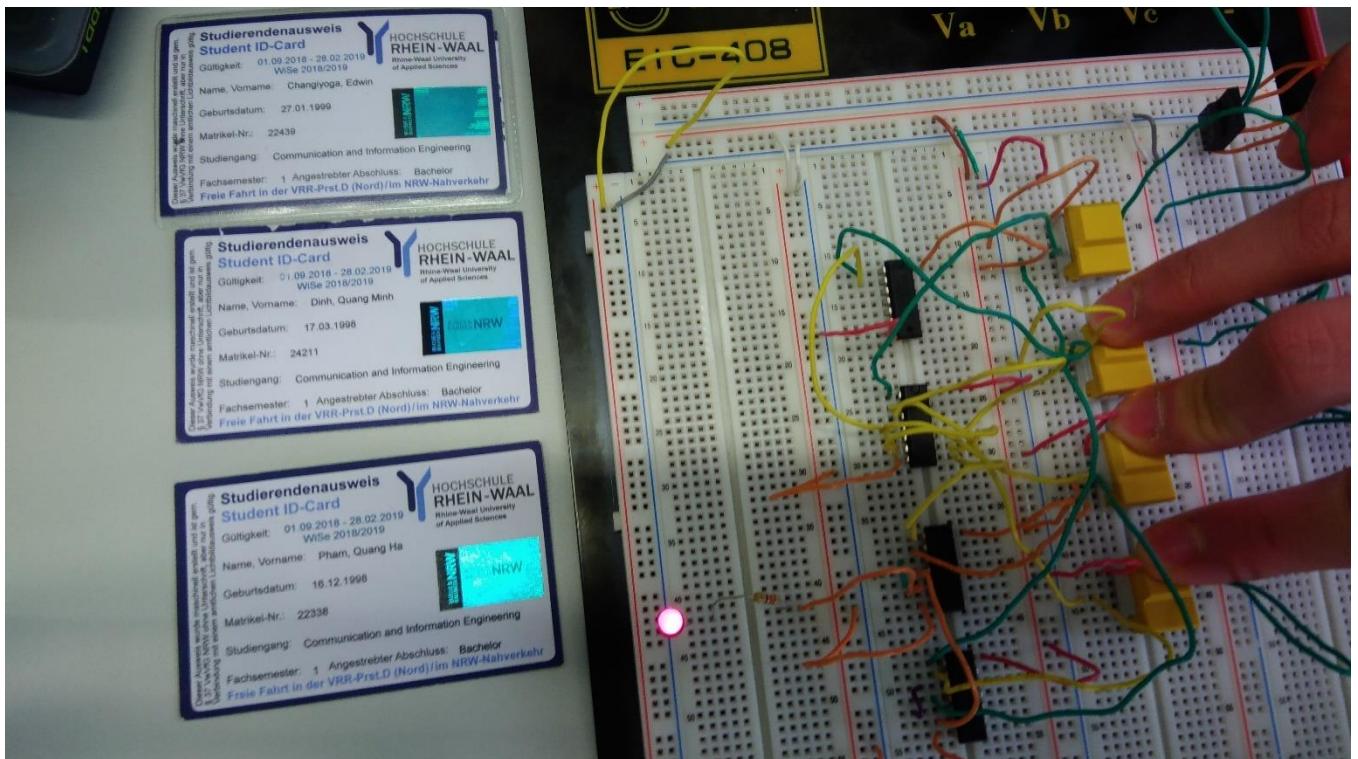
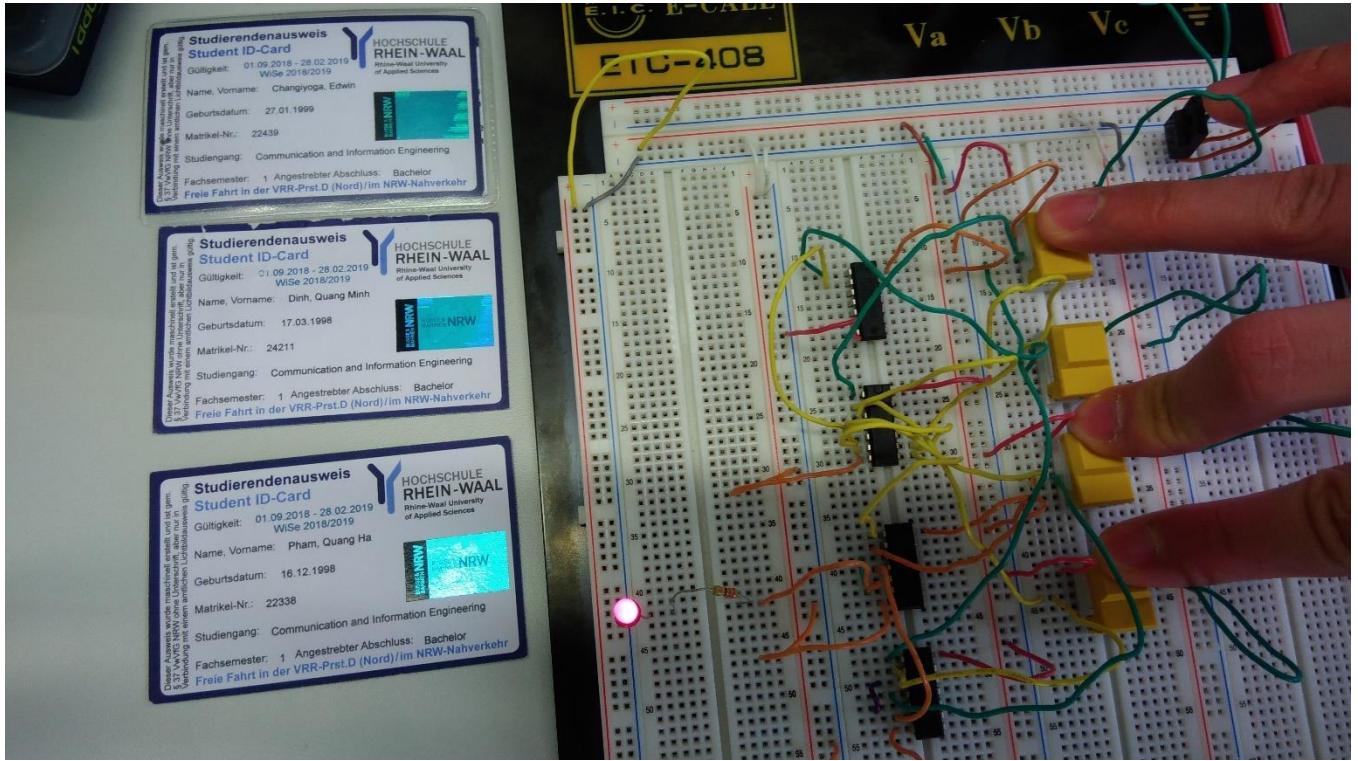


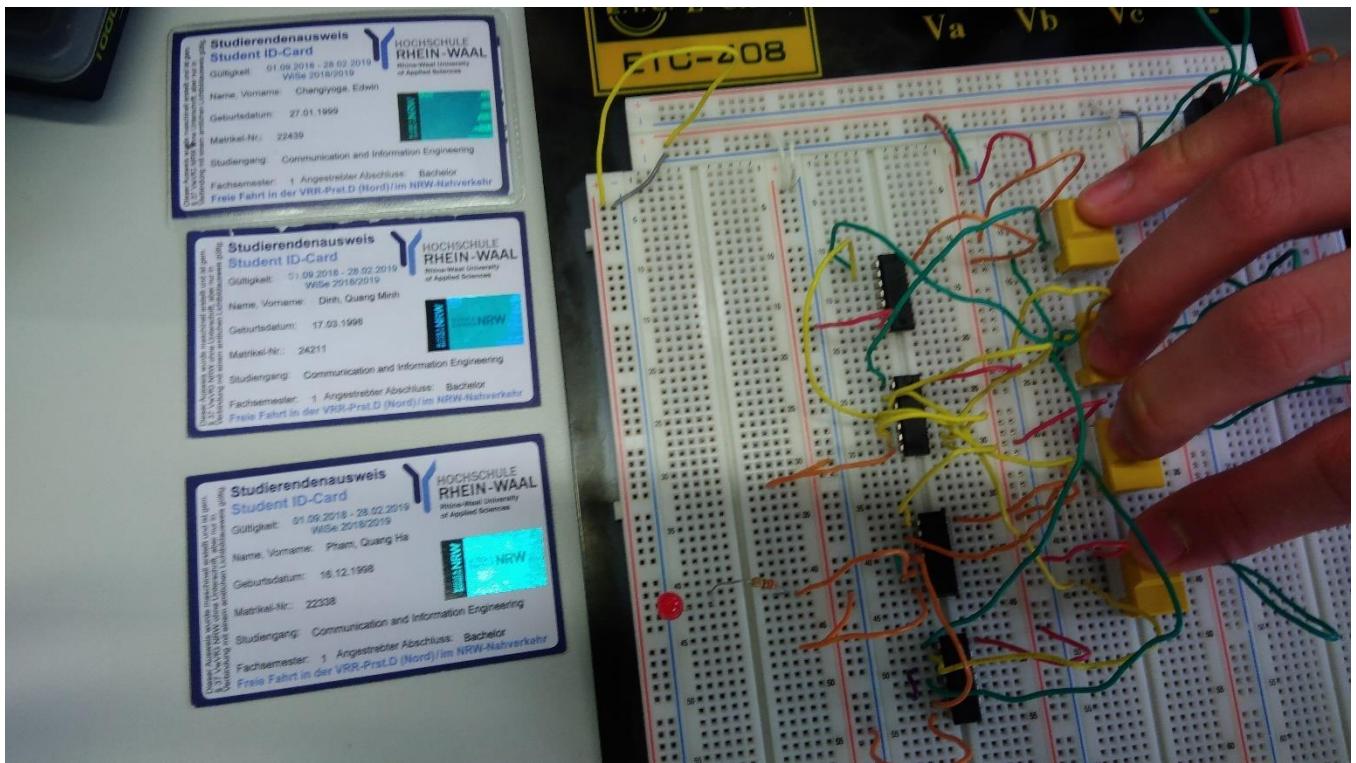












Result:

	A1 Second digit	A1 Unit digit	B2 Second digit	B2 Unit digit	
Decimal	D	C	B	A	Y
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Challenge #8

Abstract:

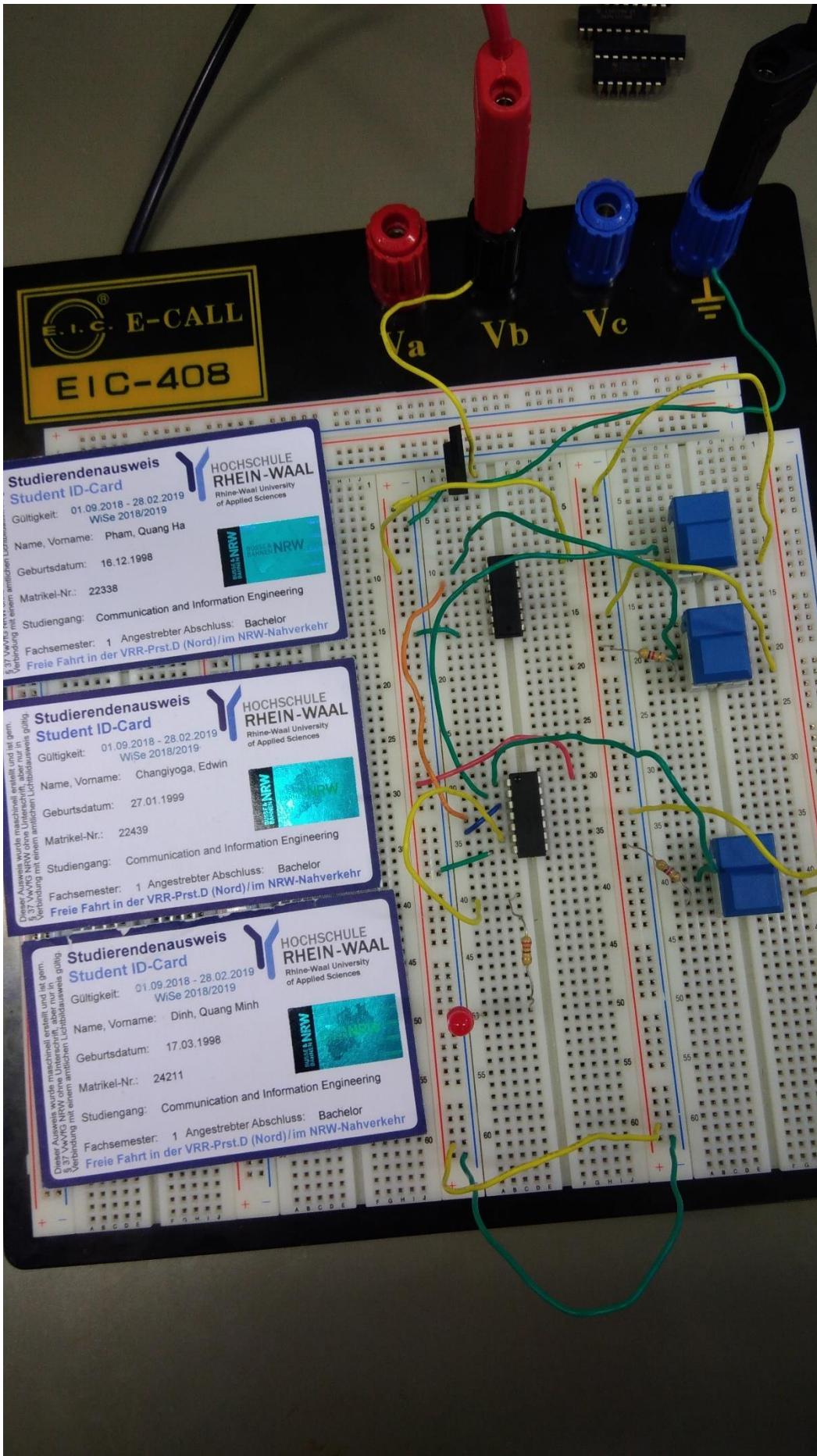
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

Our team first decided to categorize different inputs into ABCs.

A	B	C	Y
Headlight	Engine	Door	Signal
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Then we determined the simplified circuit according to the requirements: $A \wedge \neg B \wedge C$

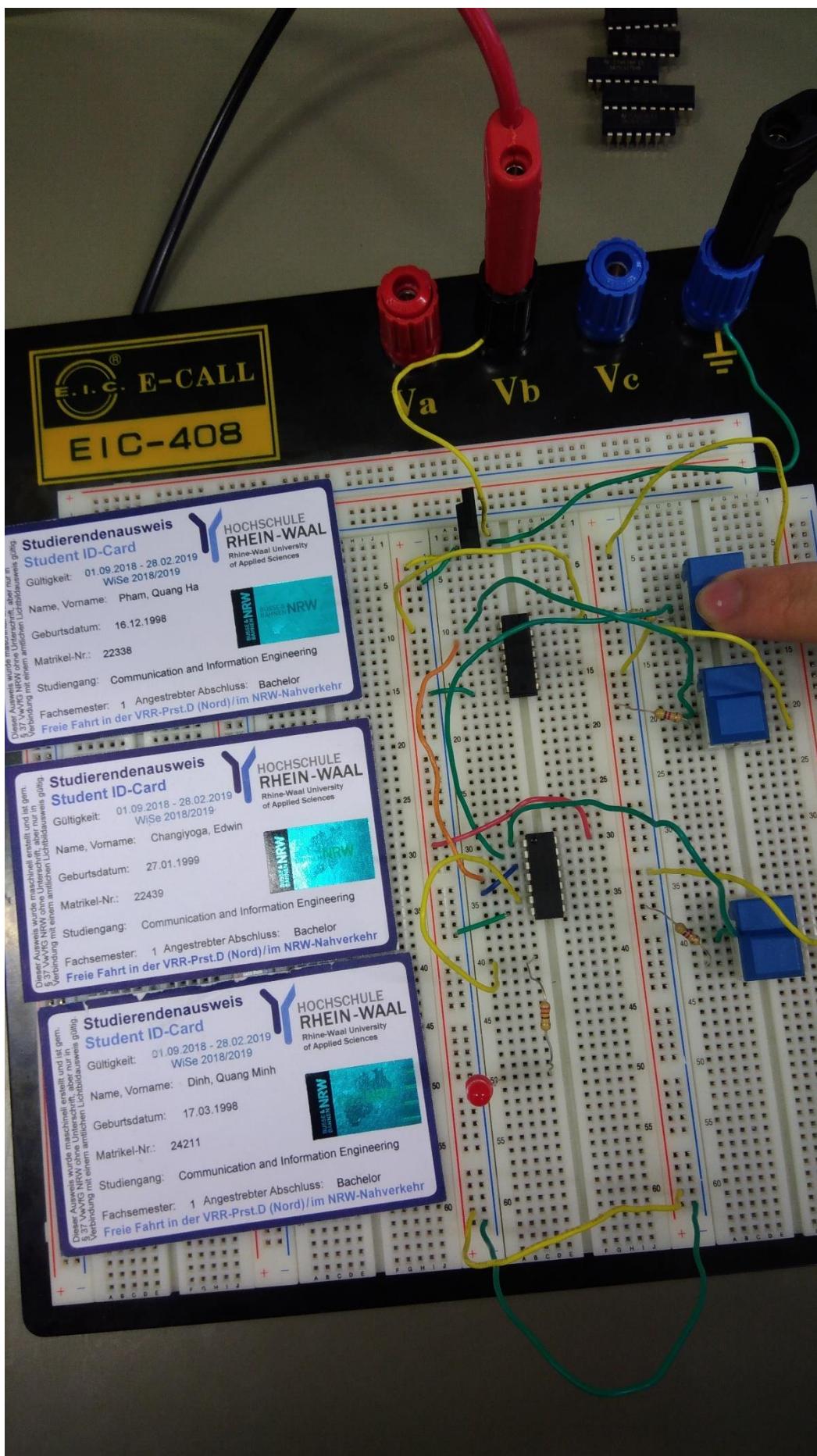
Pictures:

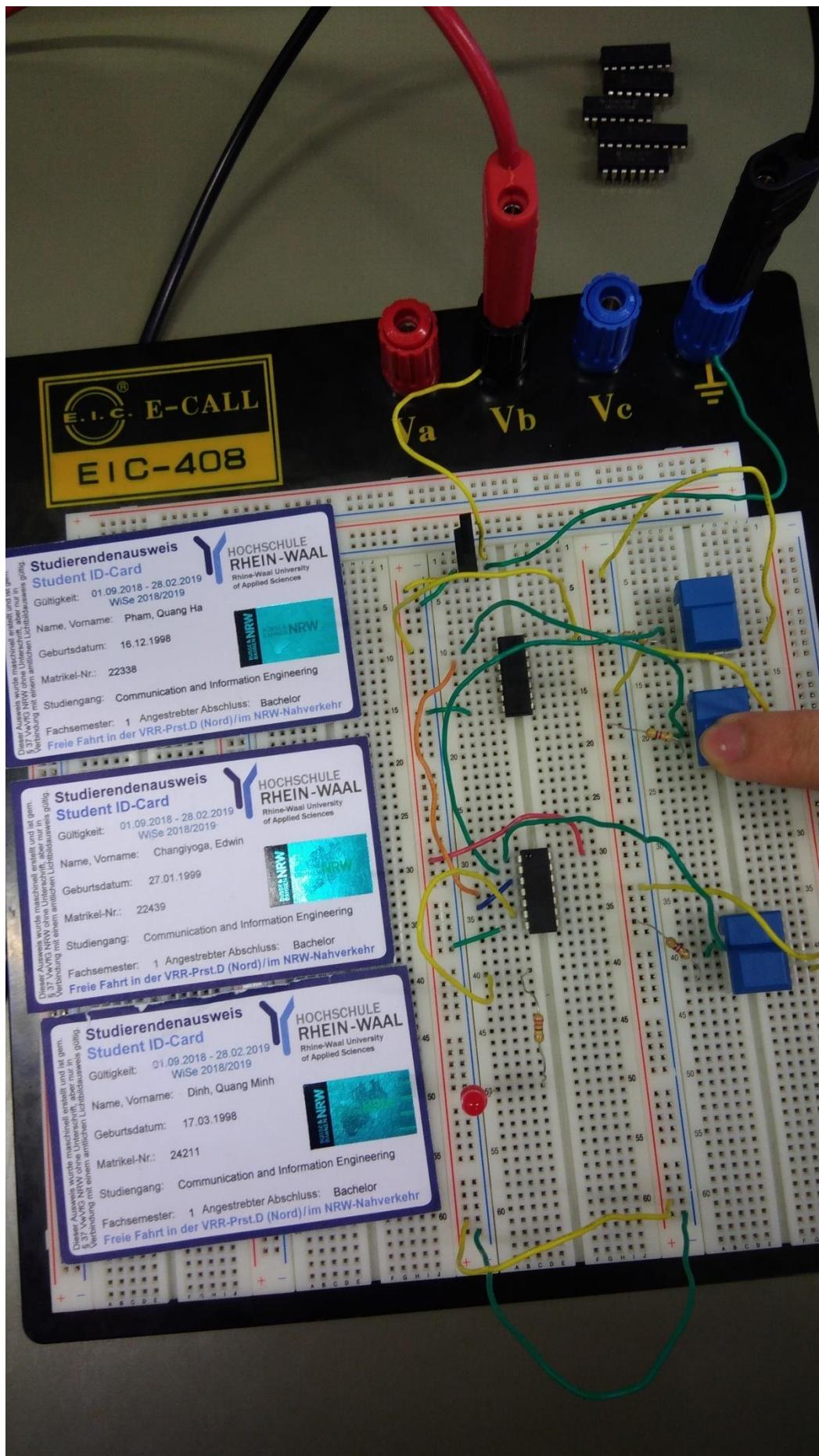


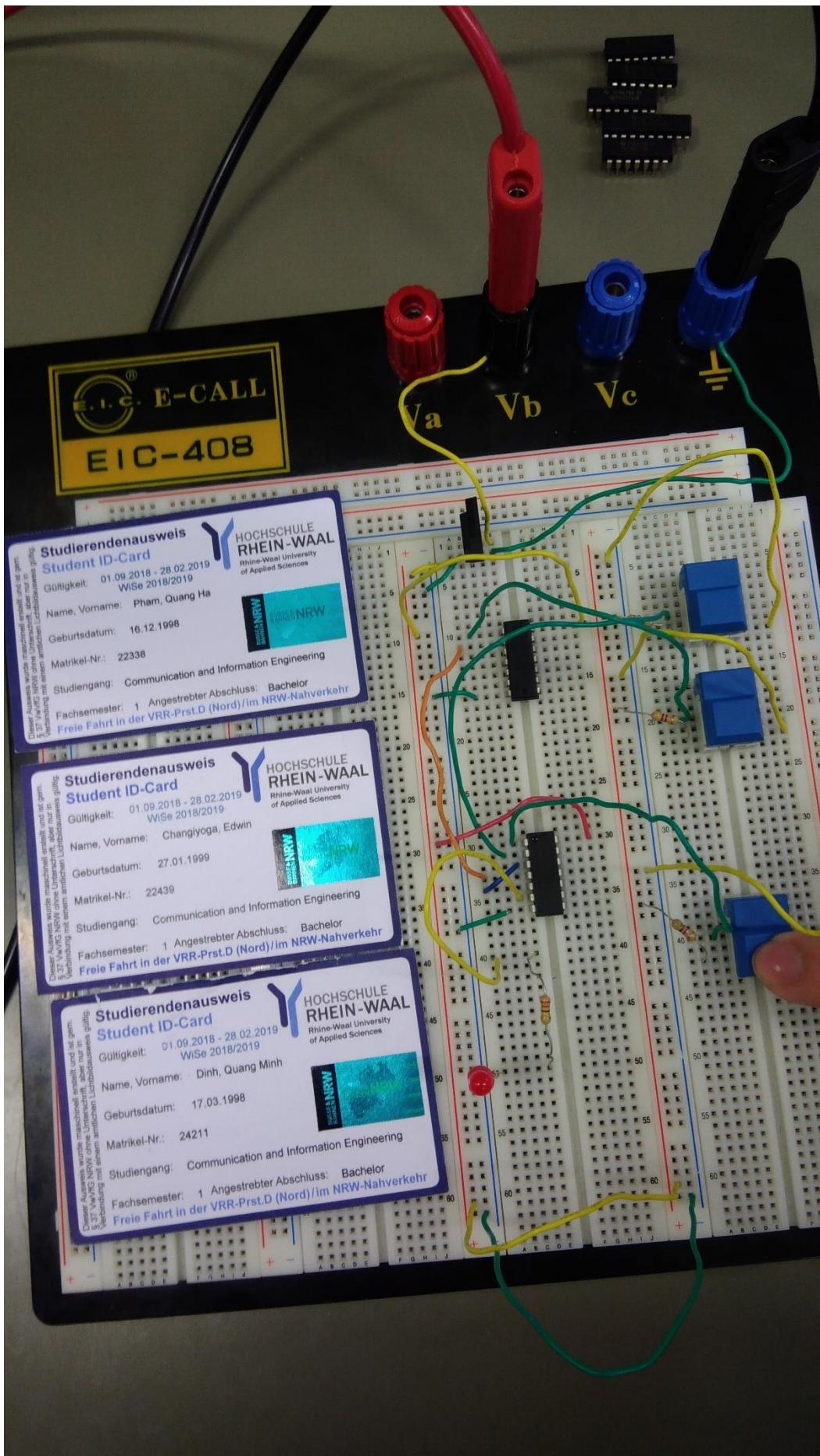
Input A

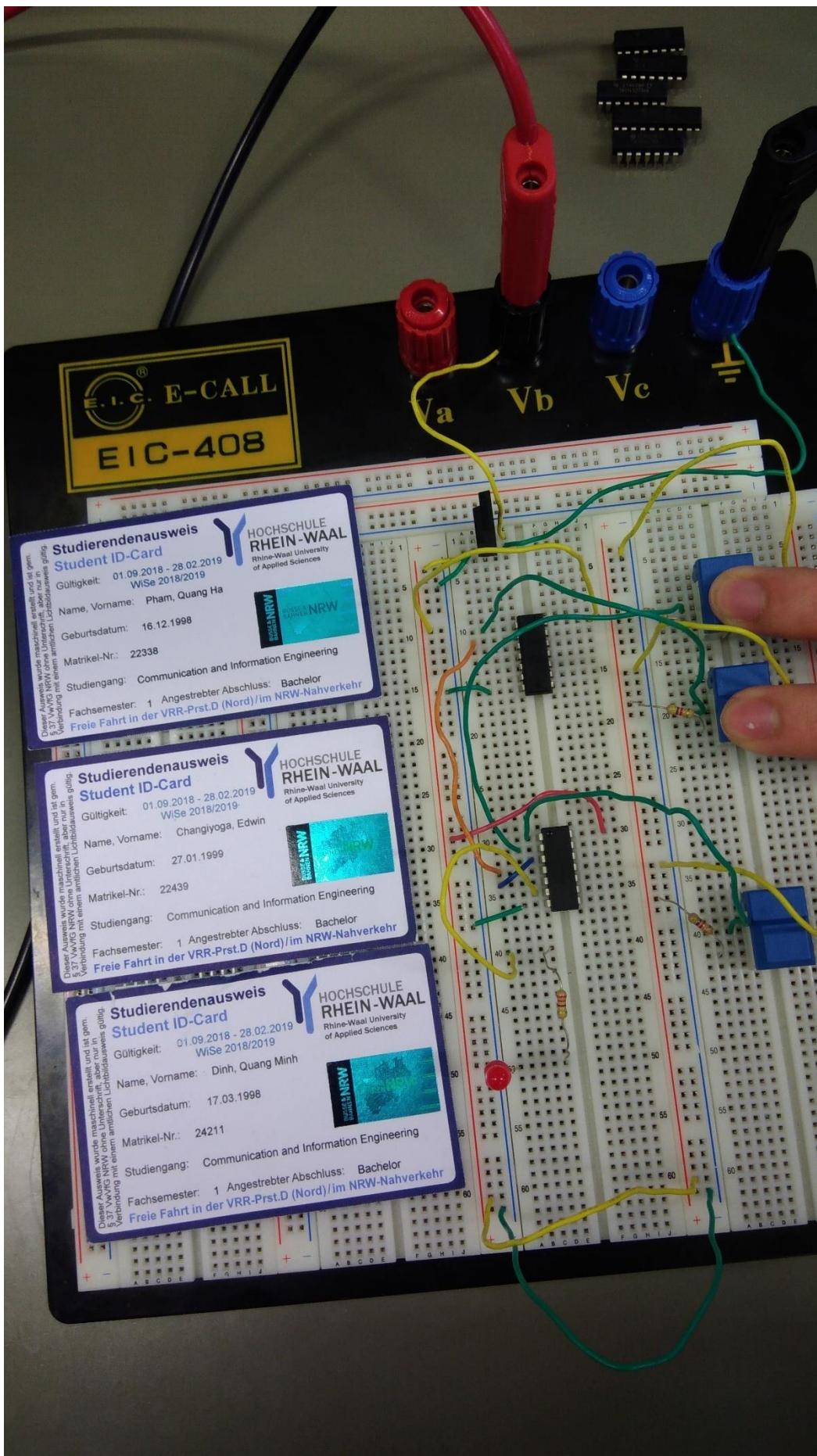
Input B

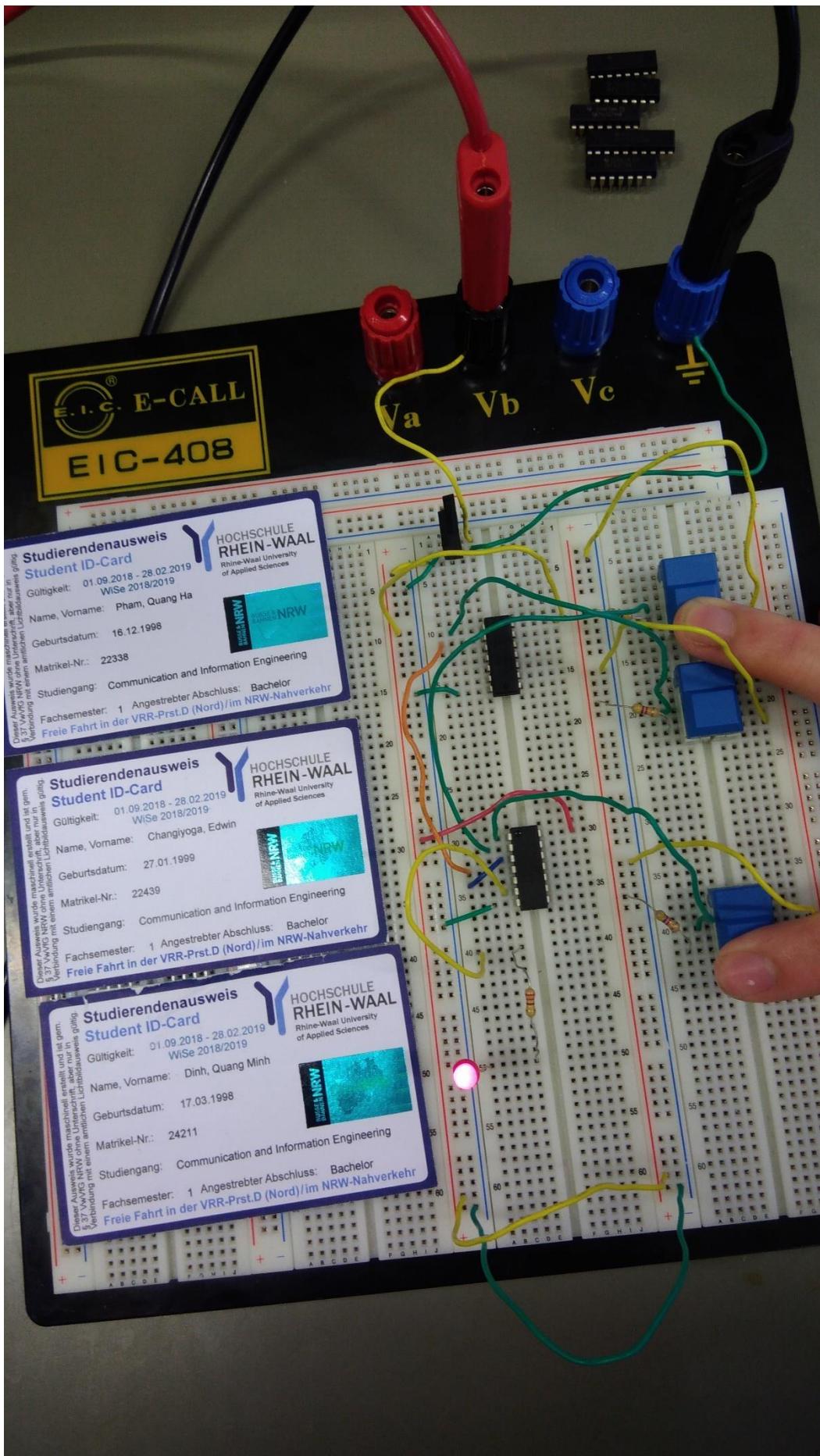
Input C

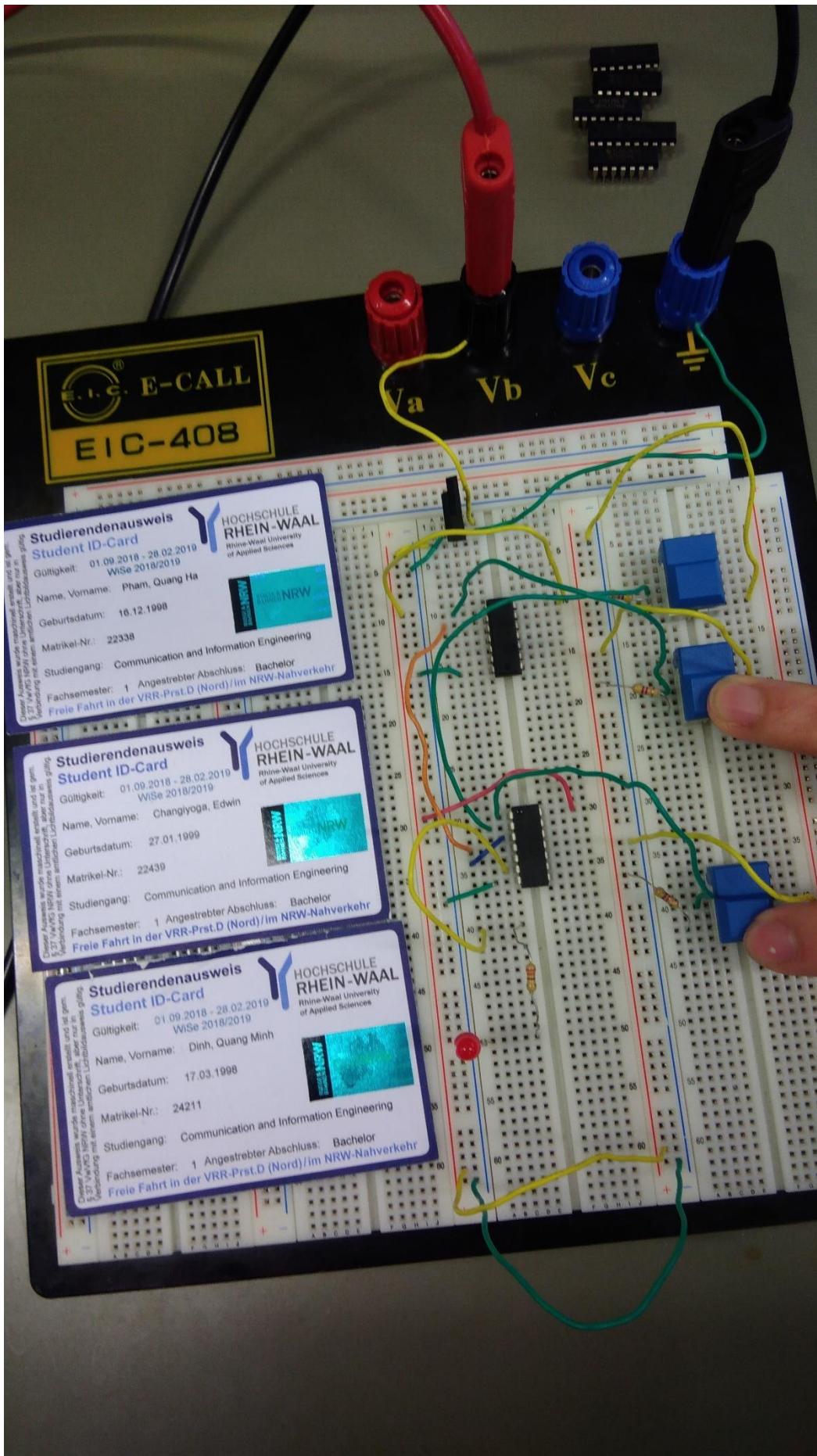


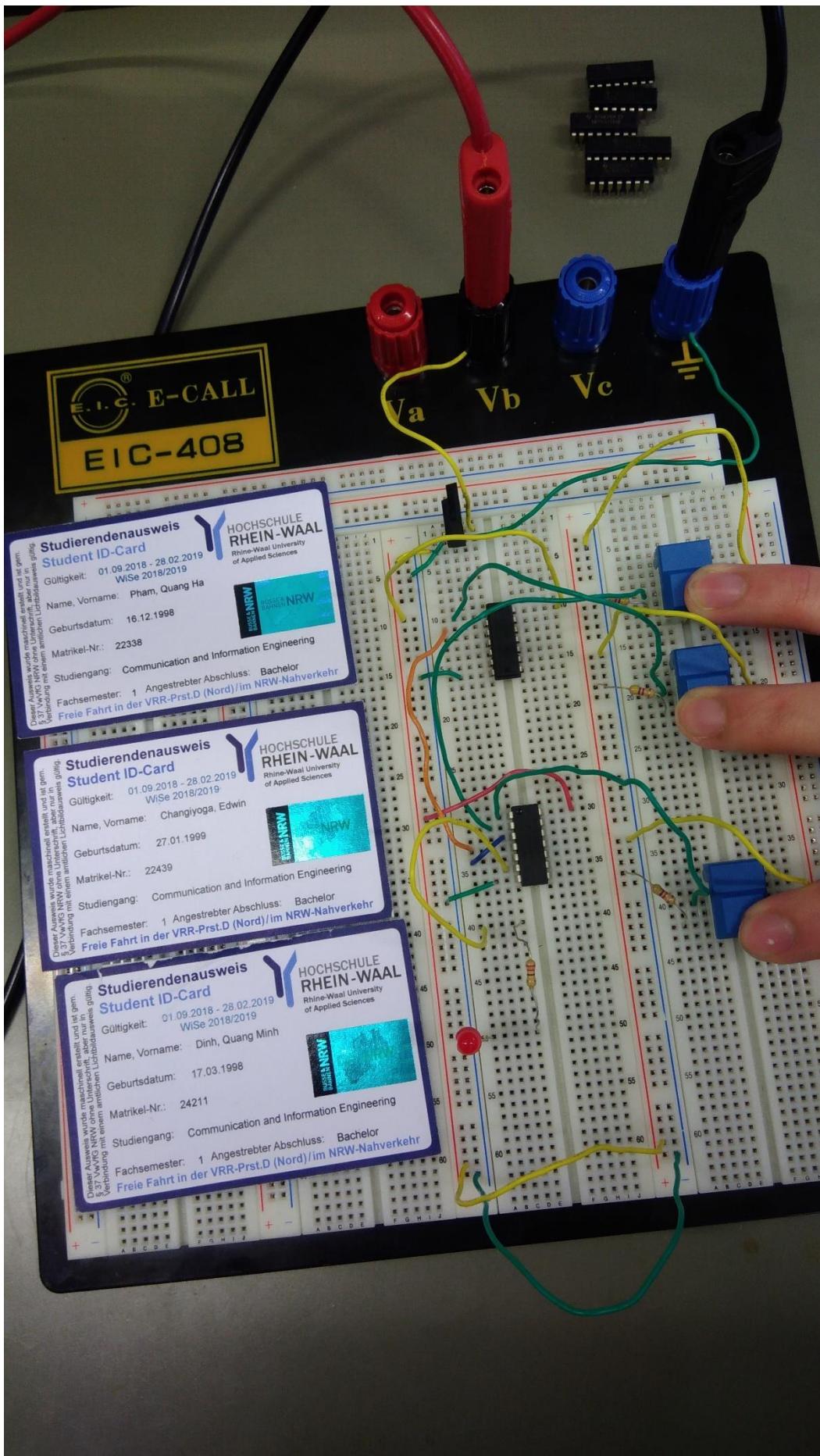












Challenge #9

Abstract:

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

Our team first read the requirements carefully and separate the inputs into two different categories: SET and RESET.

Inputs required for the Output to SET ($Q1=1$) are: S1, B1.

Inputs required for the Output to RESET ($Q1=0$) are: S0, S2, B2, F1.

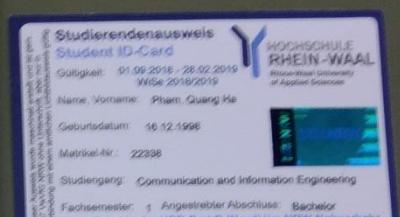
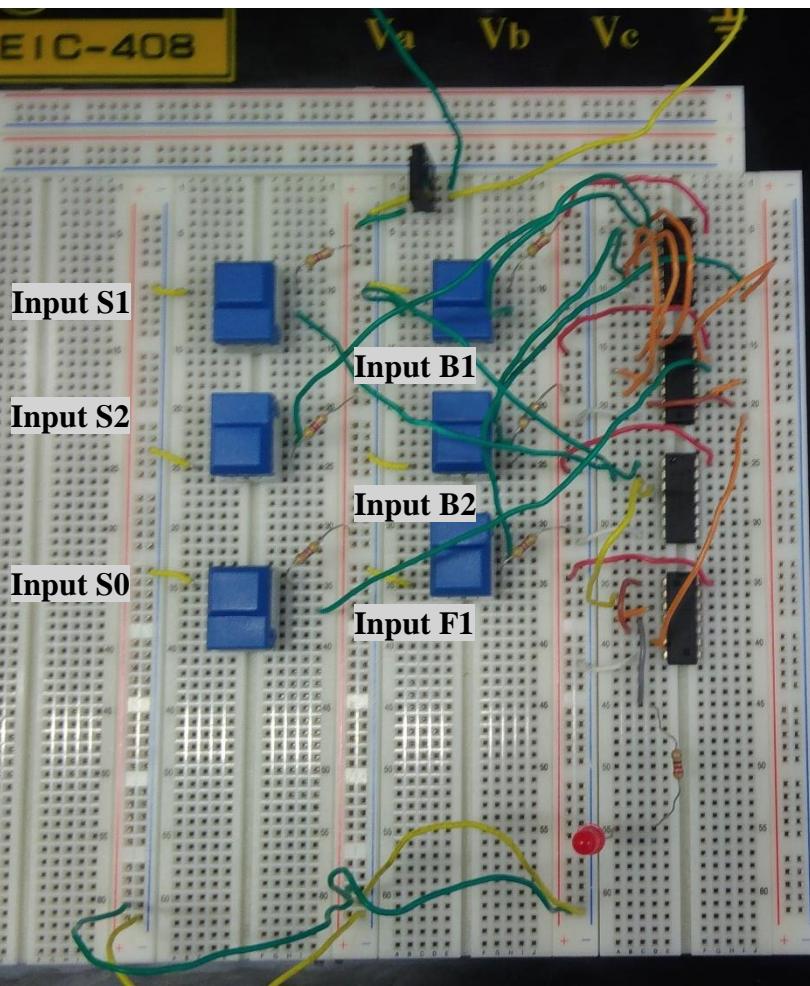
Since the requirement states that “switch-on interlock with open protective cover”, or in other words: S1 is interlock with B1, we decided to connect S1 and B1 together with an AND gate.

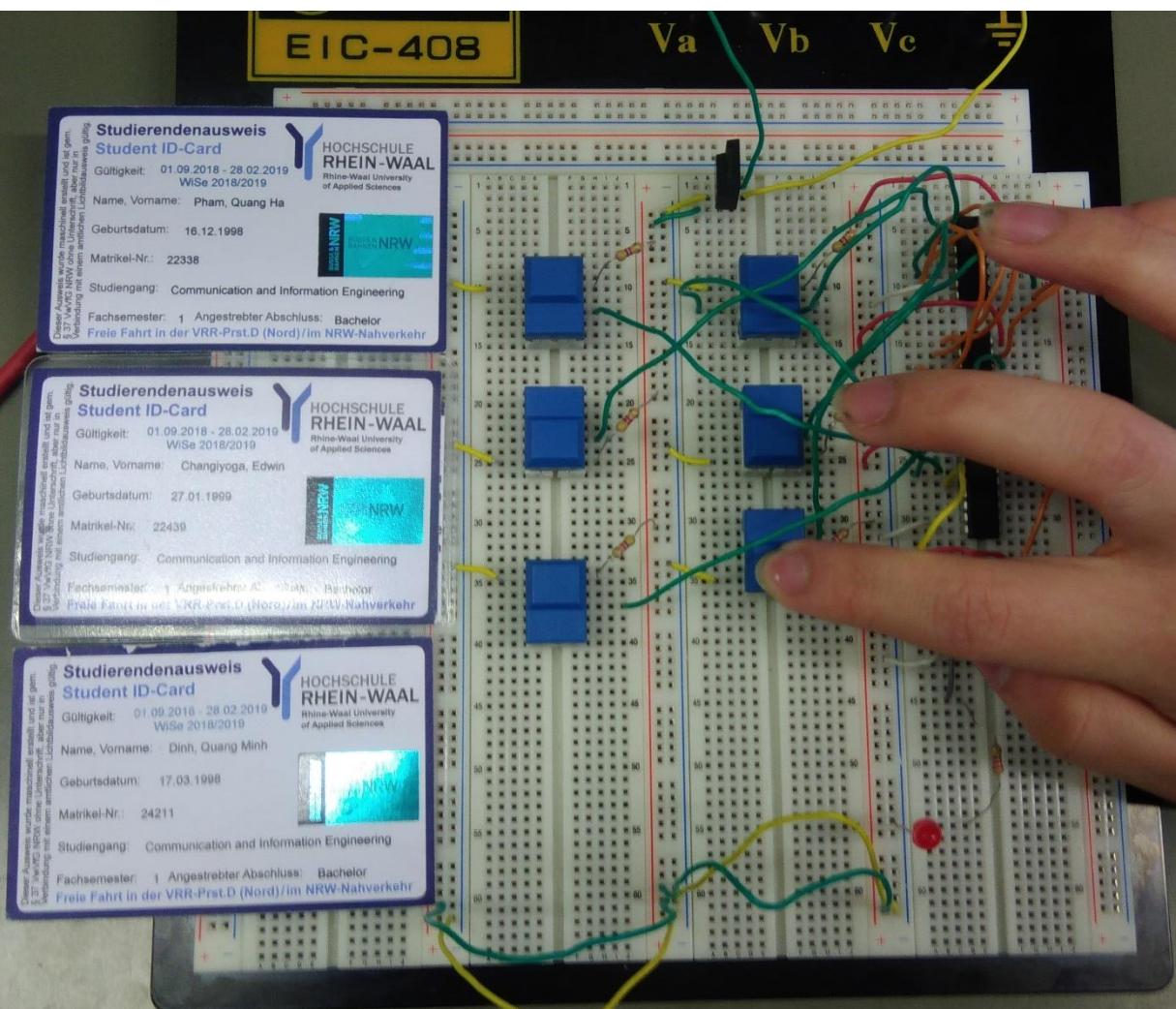
For the rest of the inputs, we didn't find anything that mentions that these inputs are interlocked with each other so we decided to connect S0, S2, B2, F1 together with an OR gate.

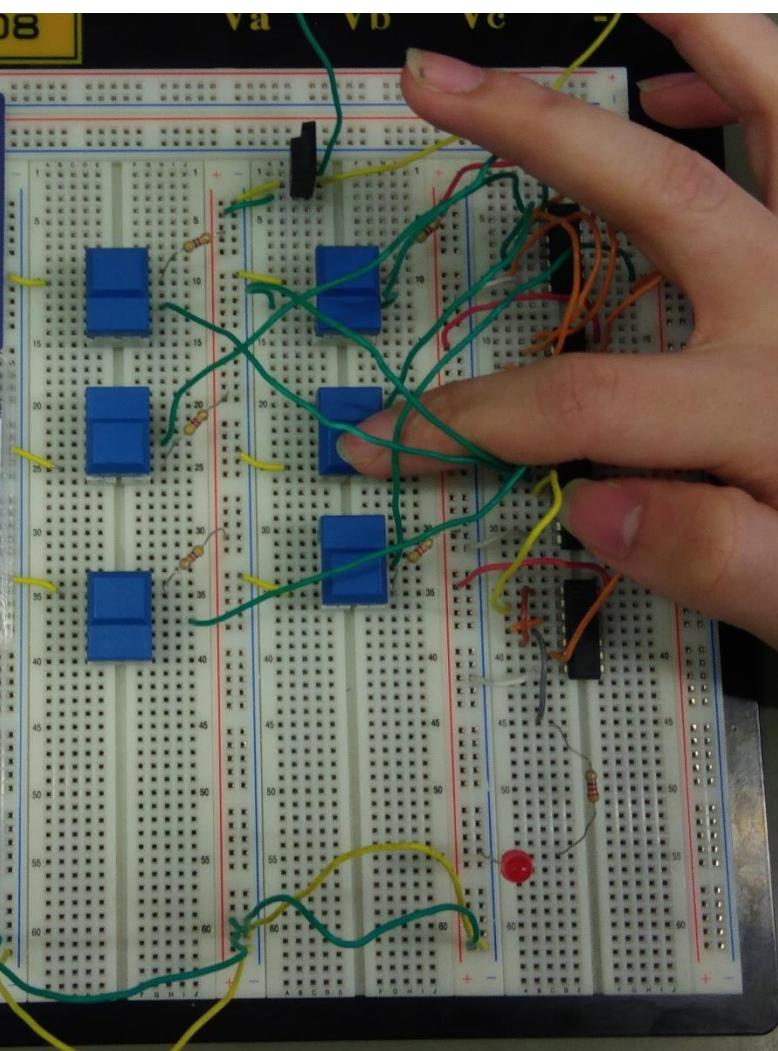
The requirement also needs a digital circuit for controlling the saw using a reset-dominant RS flip-flop so we used the circuit in Lab 3 challenge 7 as our base, but for this challenge we made the SET input: $(S1 \wedge B1)$ and RESET input: $(S0 \vee S2 \vee B2 \vee F1)$.

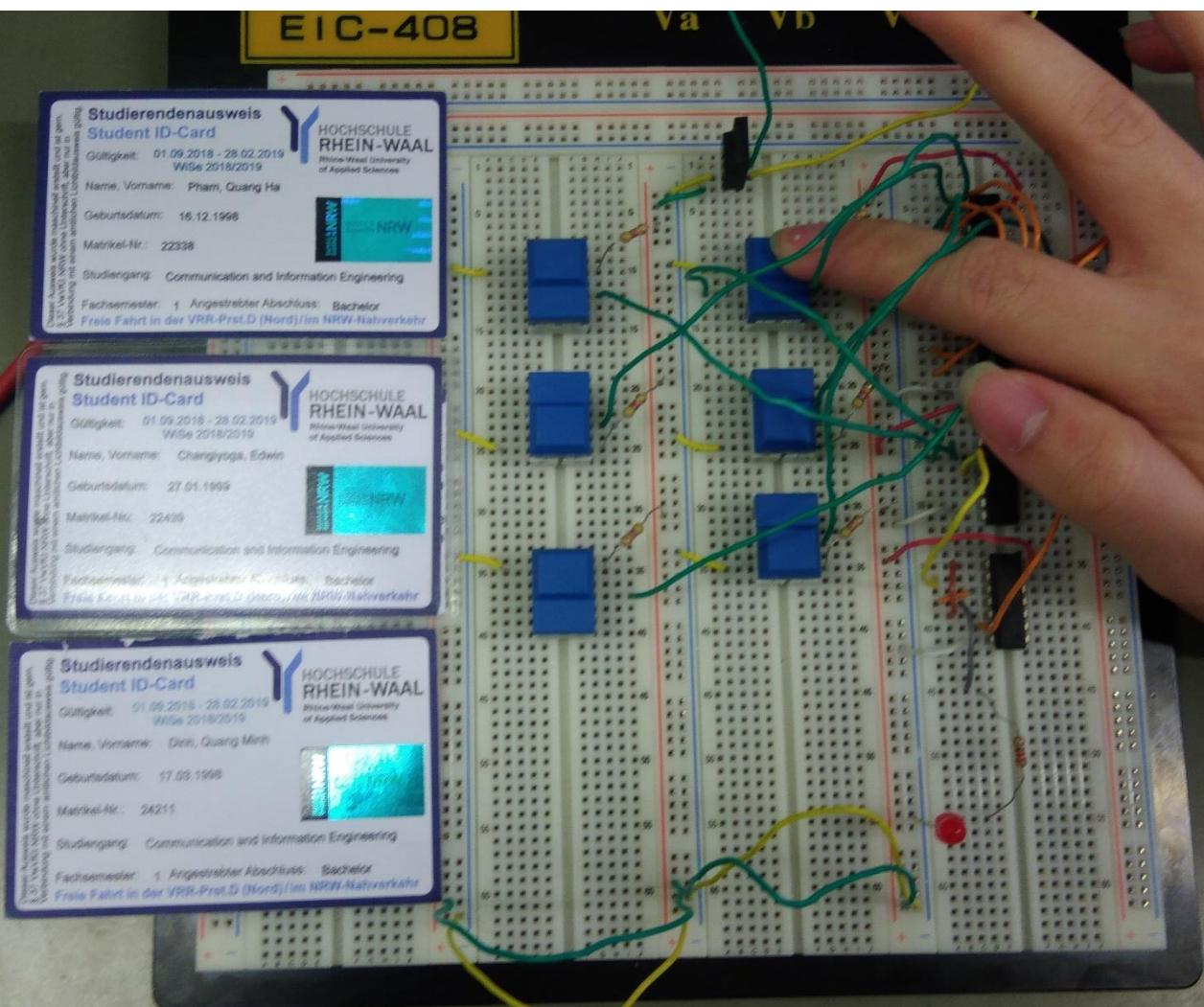
Our group determined that the only case where output $Q1=1$ is when: $S1=B1=1$ and $S0=S2=B2=F1=0$.

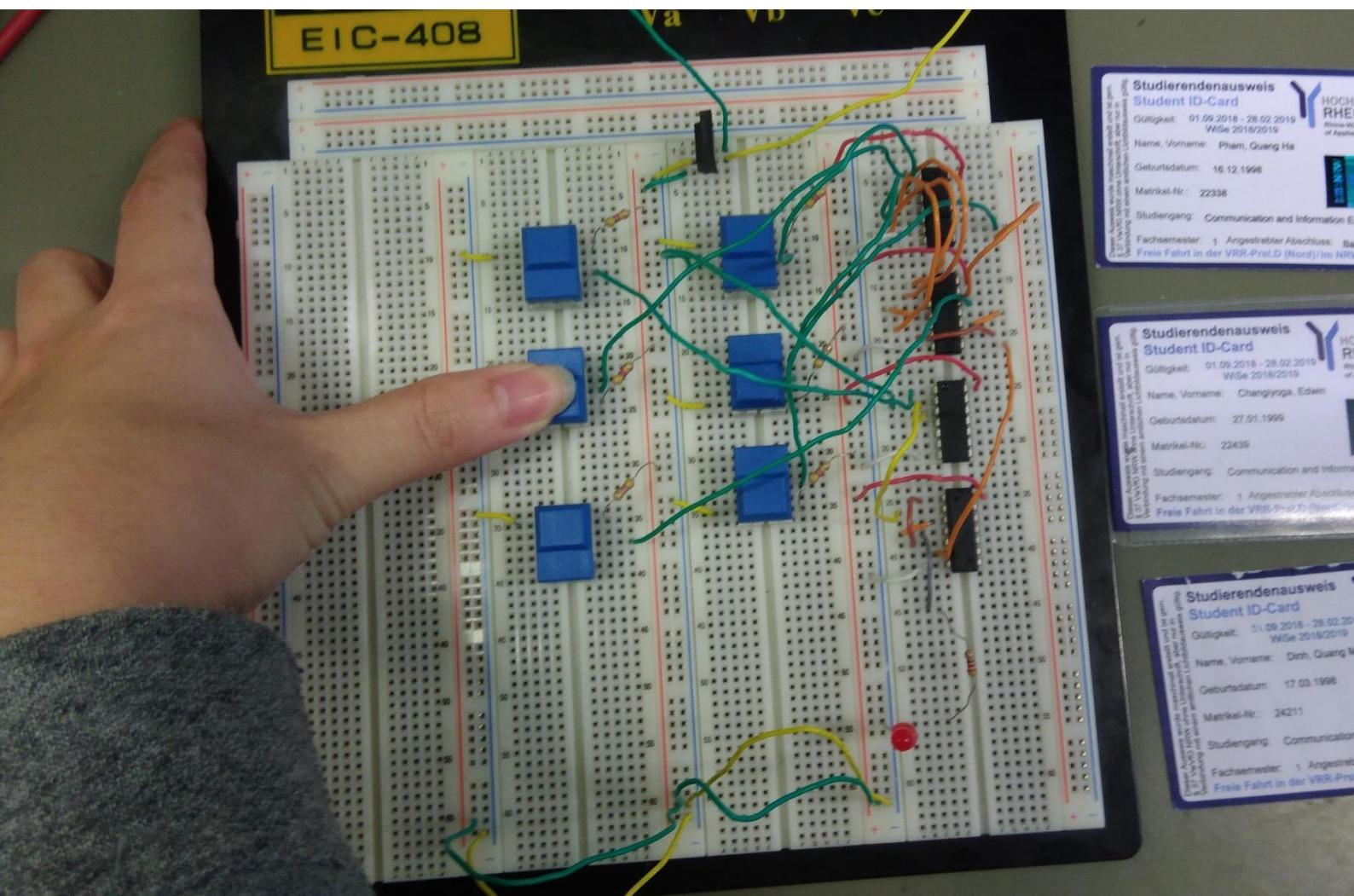
Pictures:

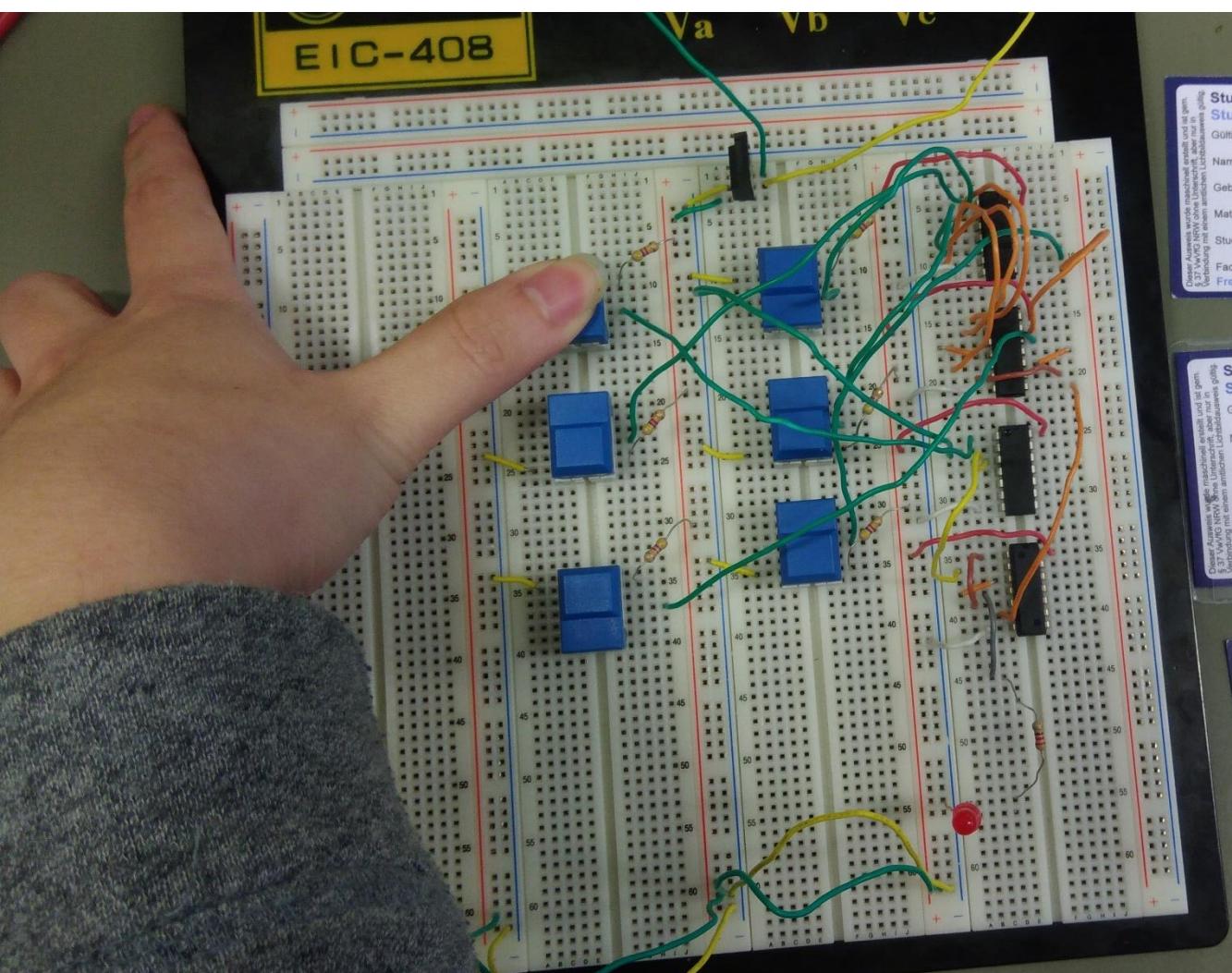


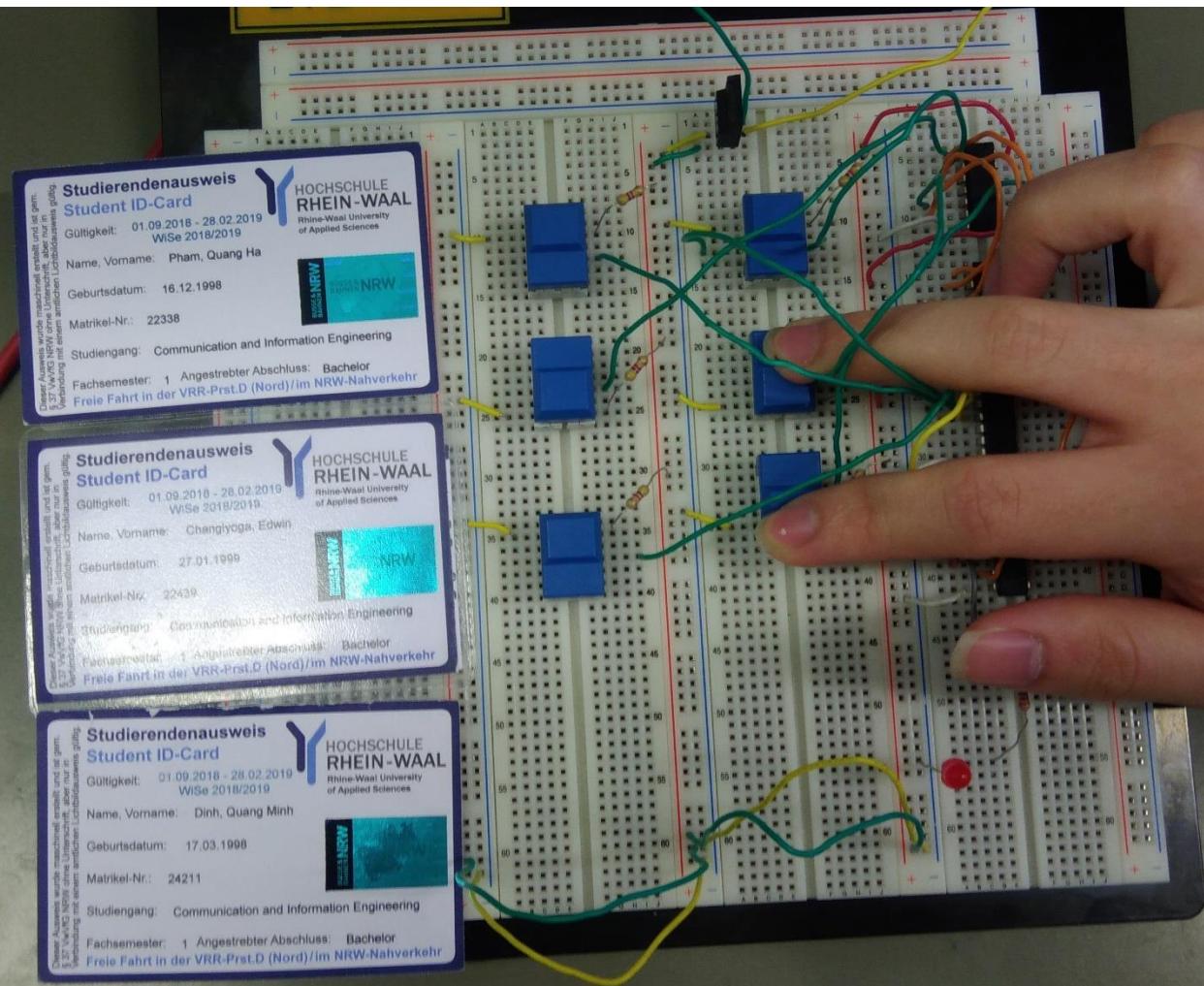


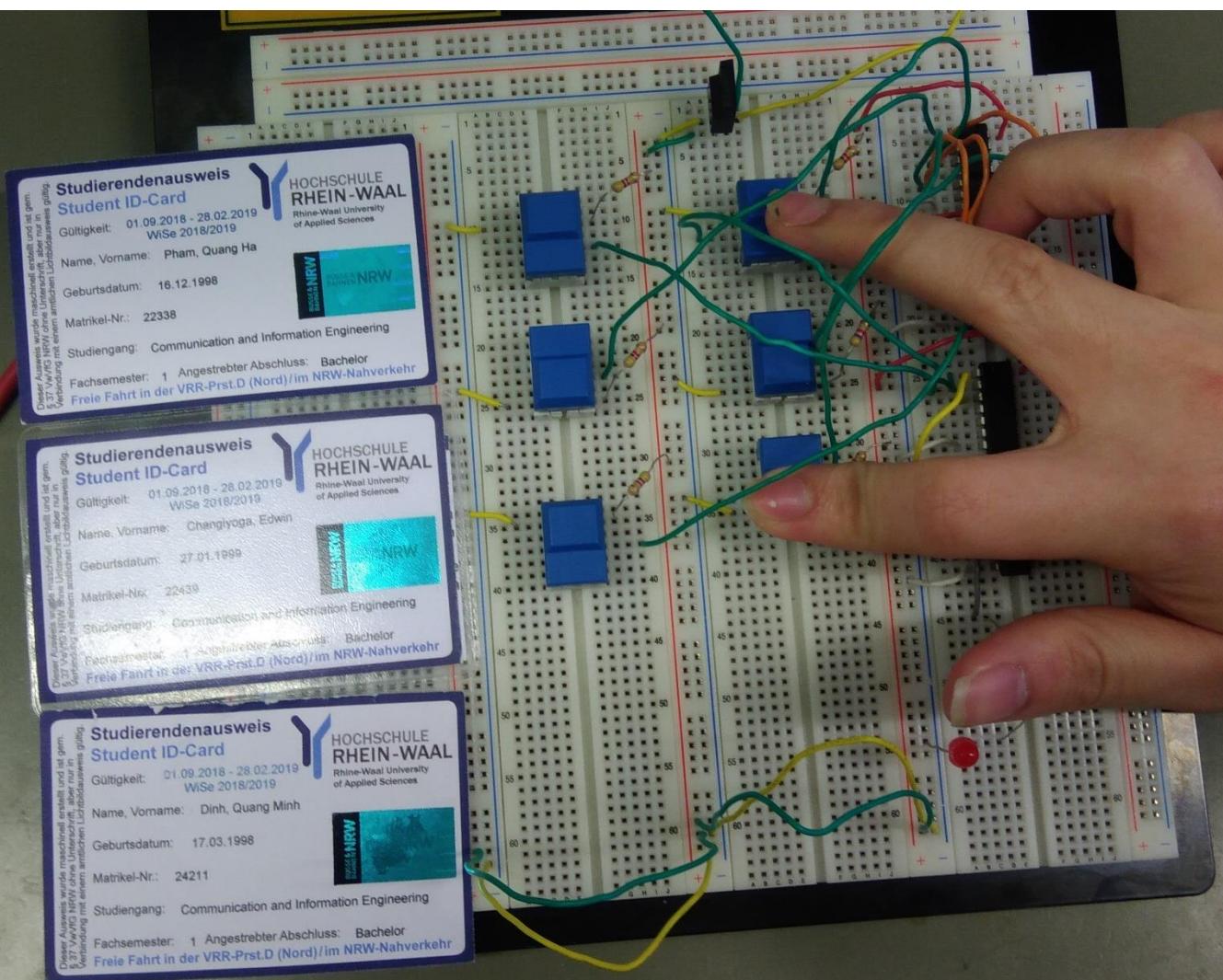


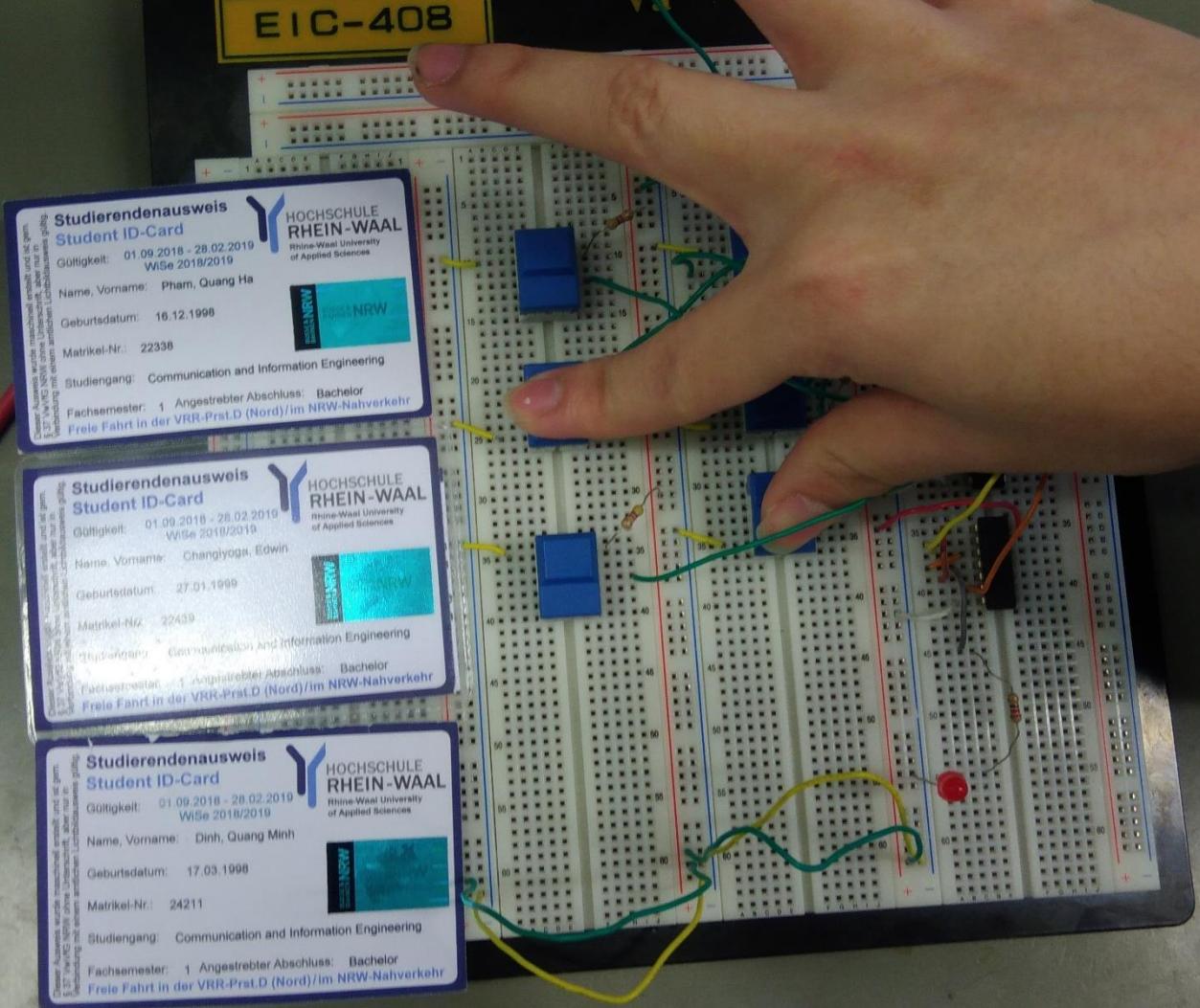


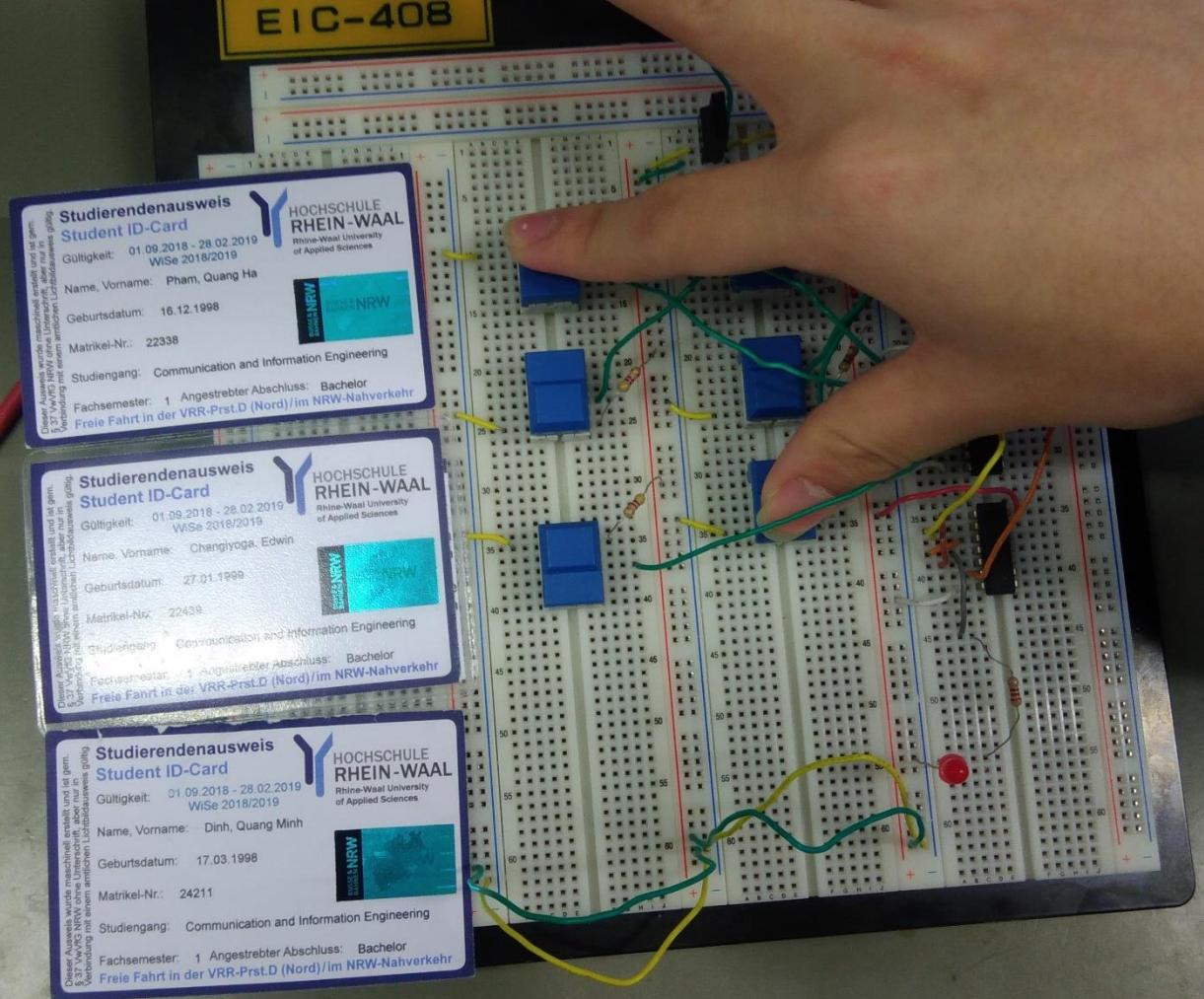




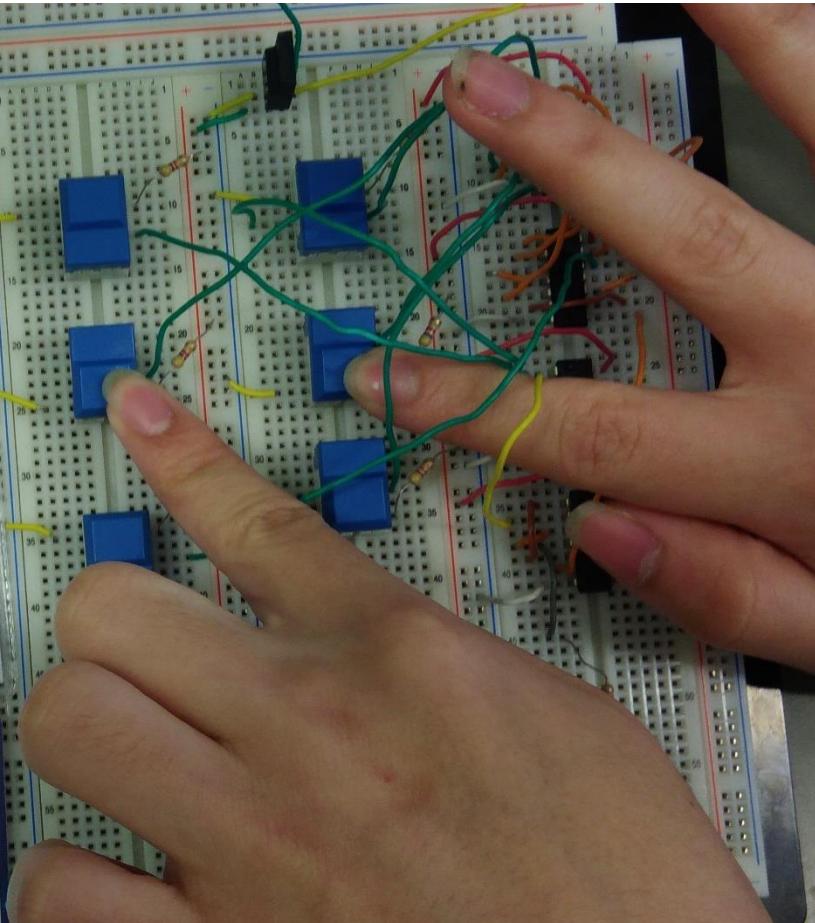
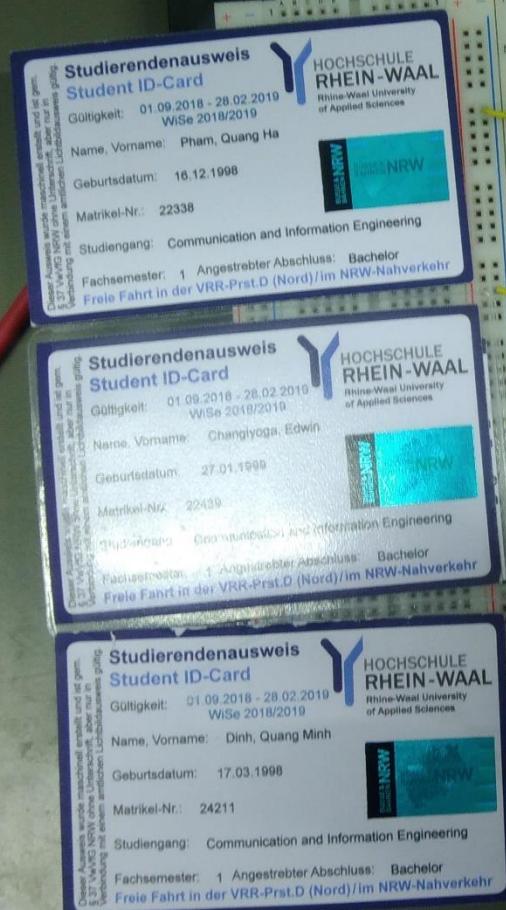


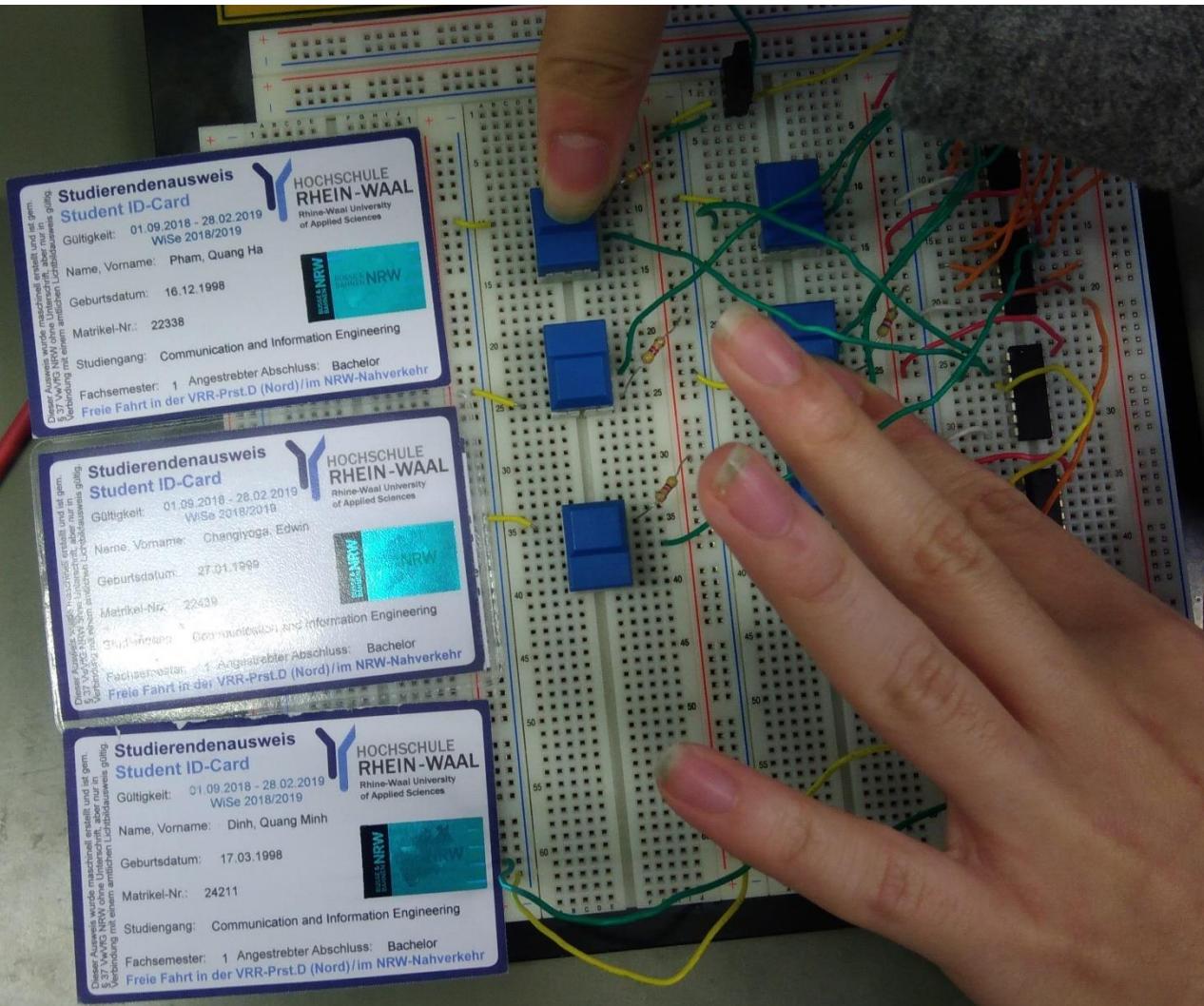




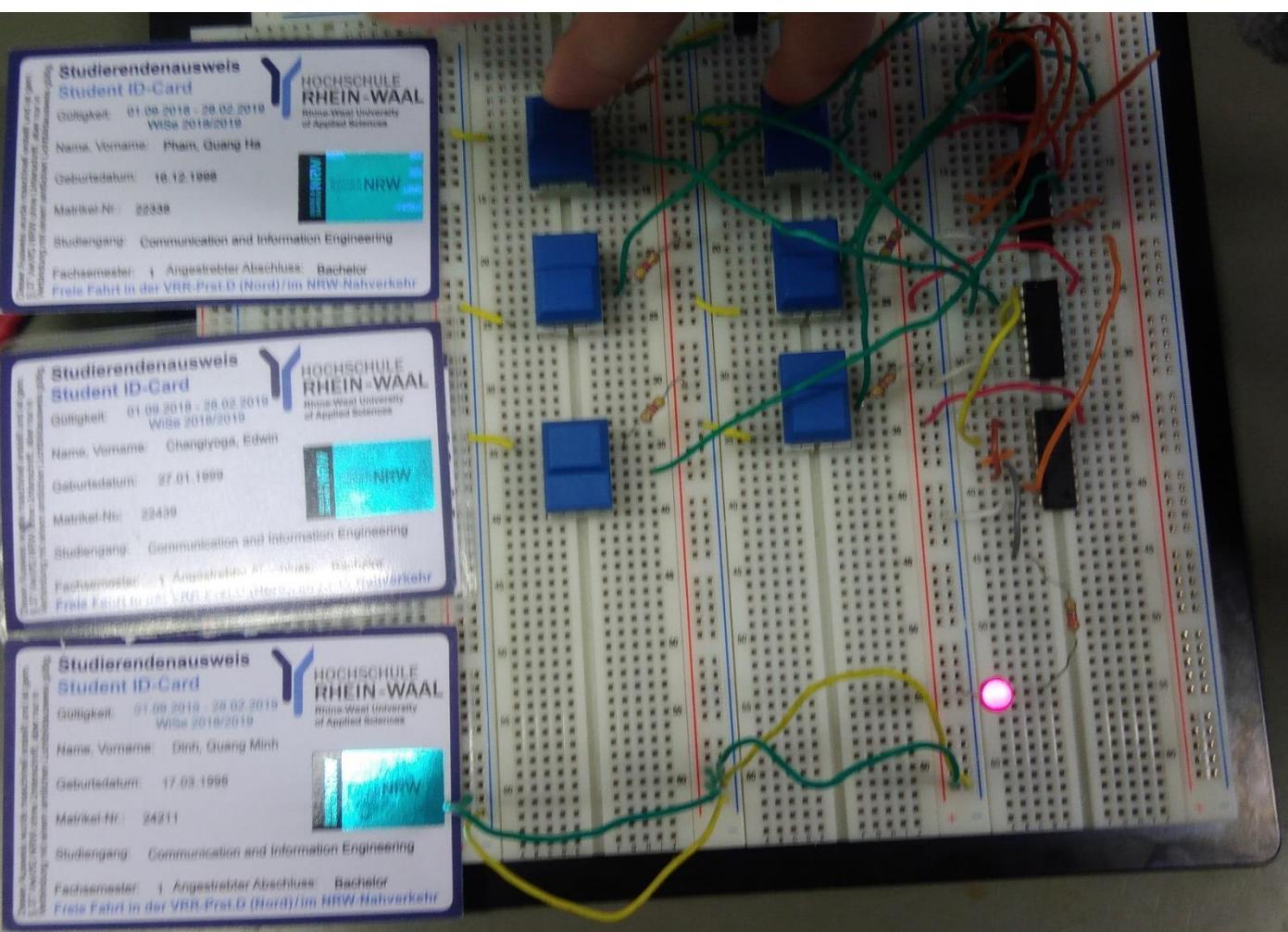


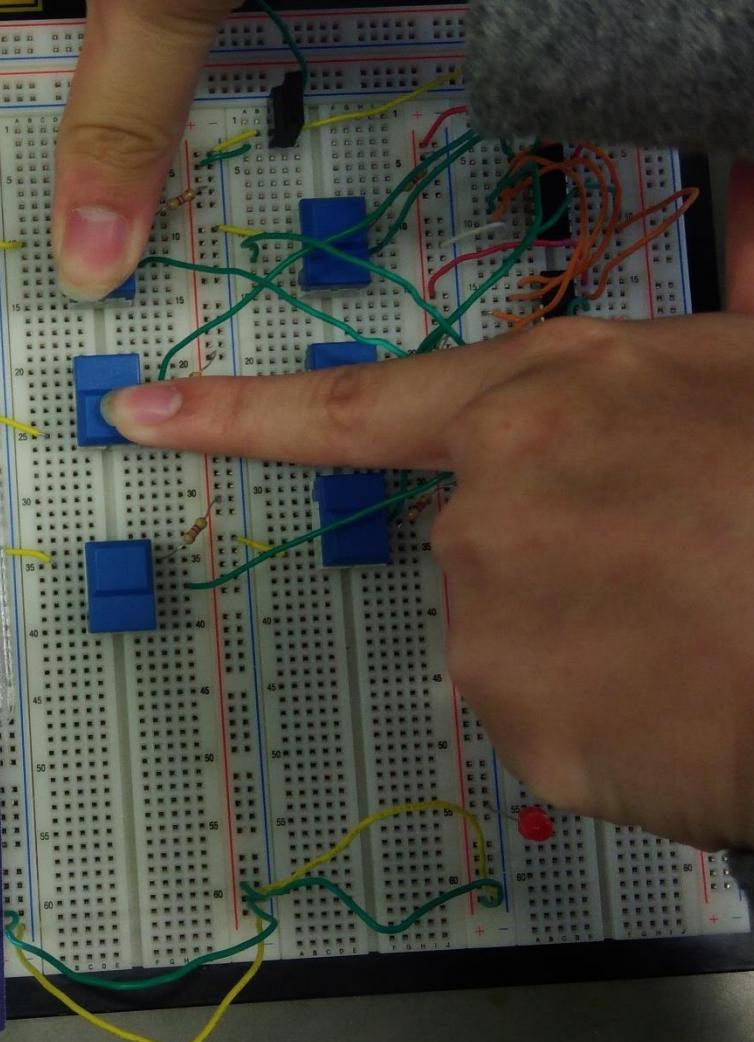


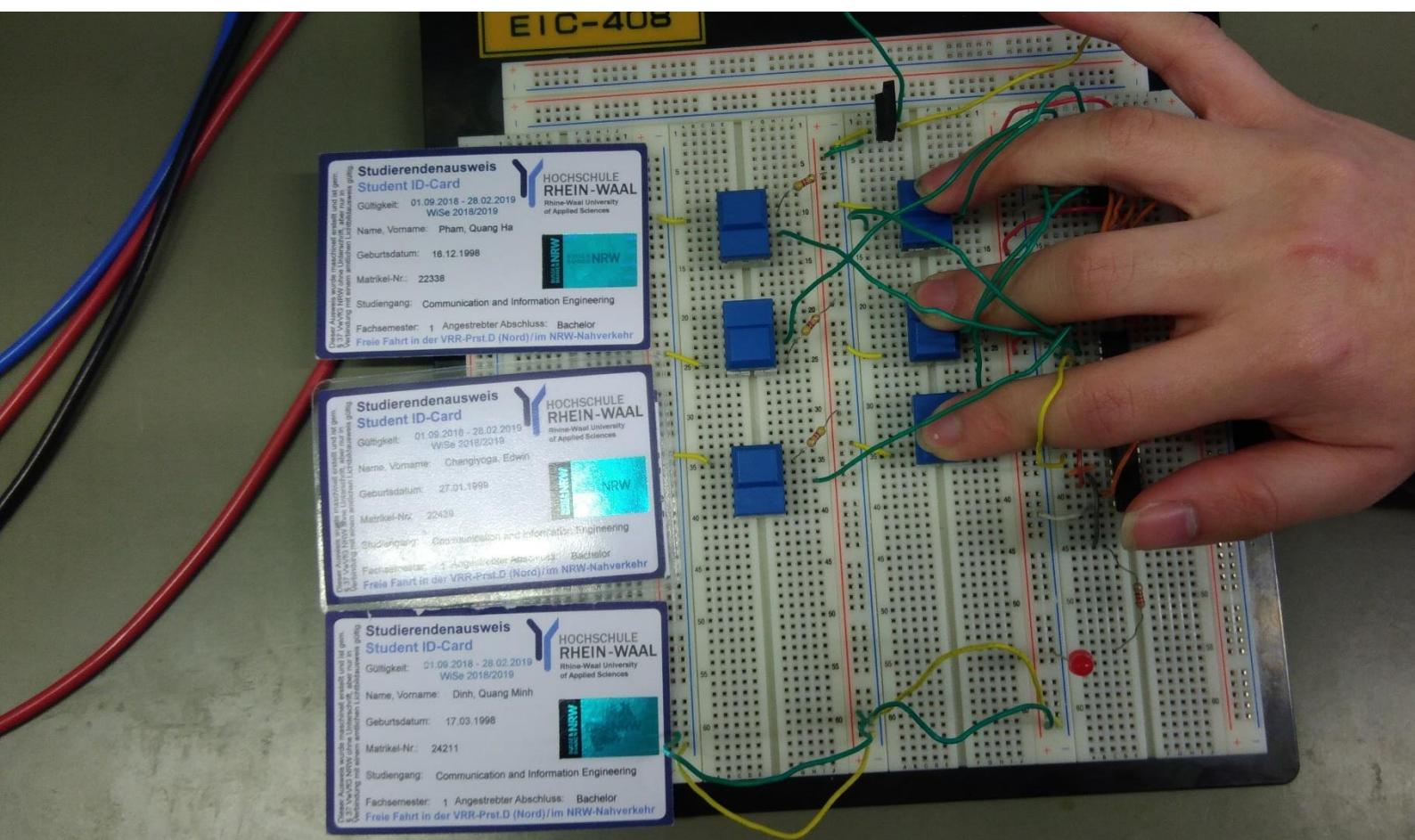




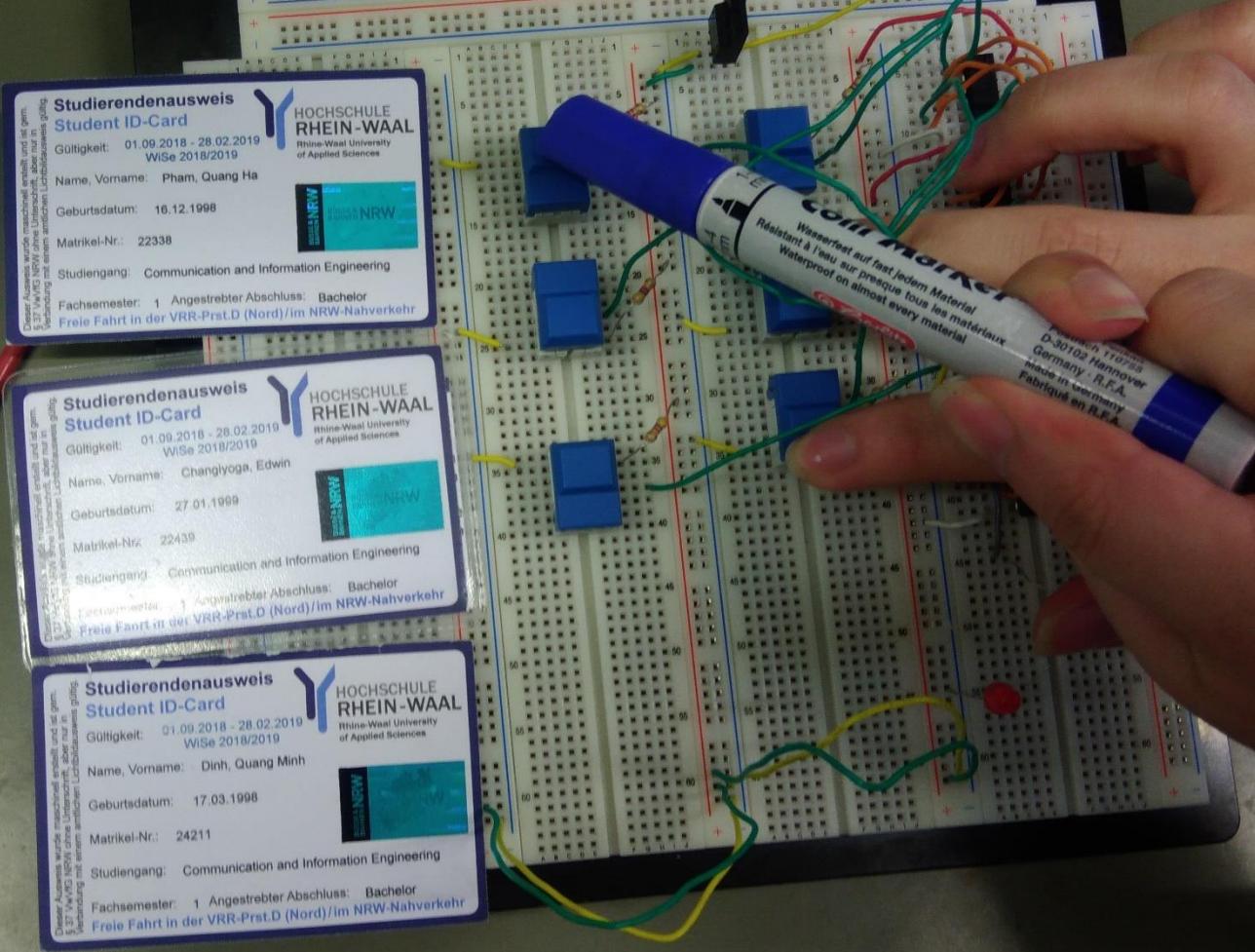




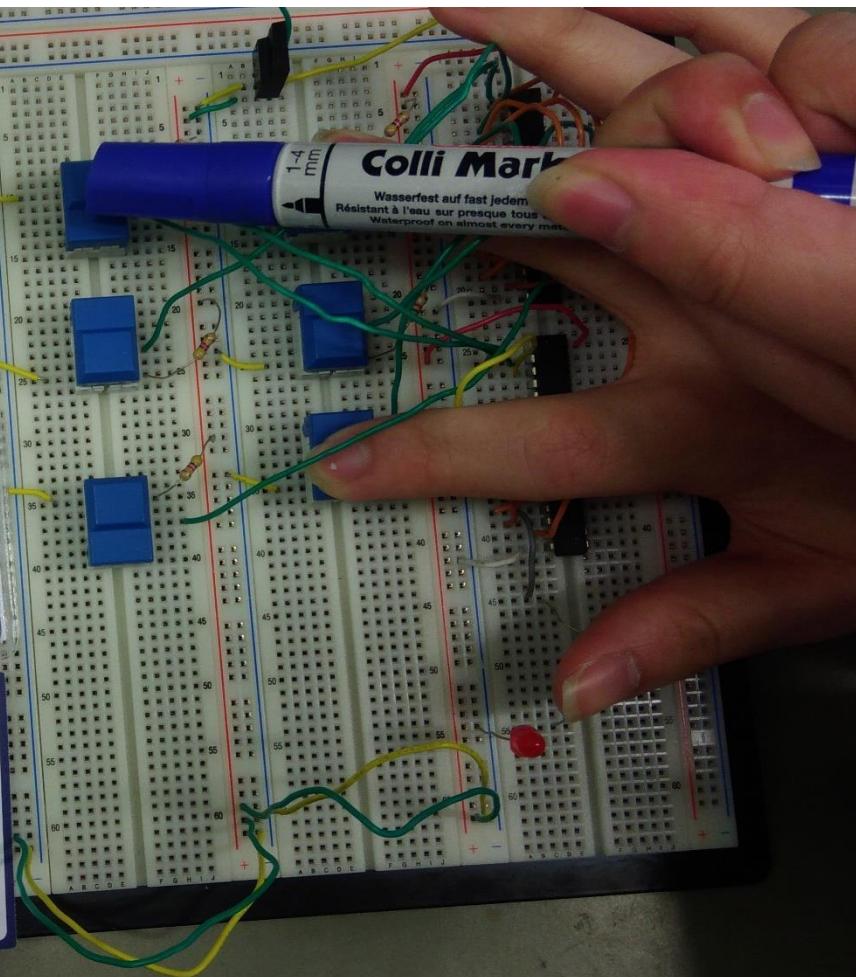


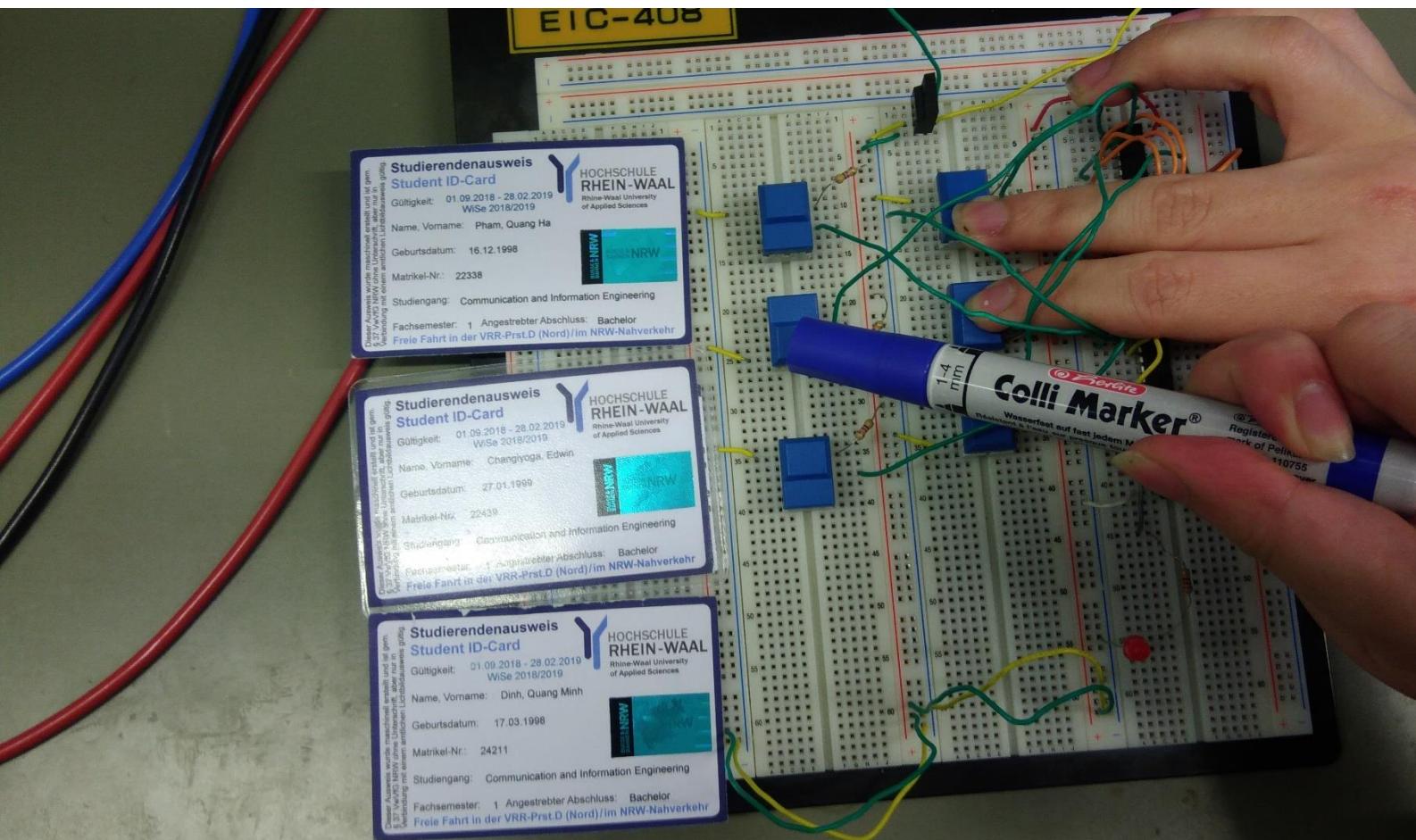


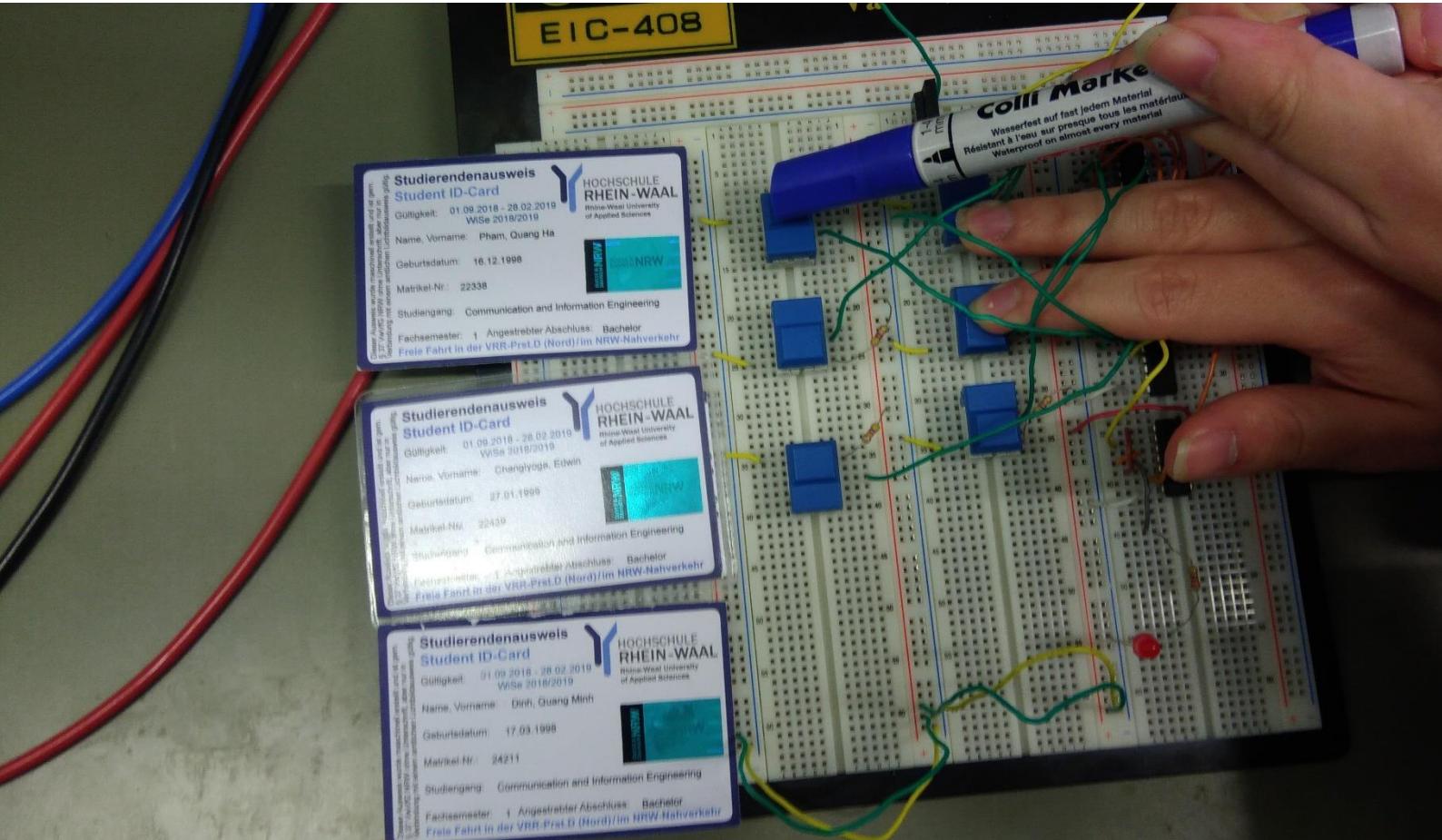


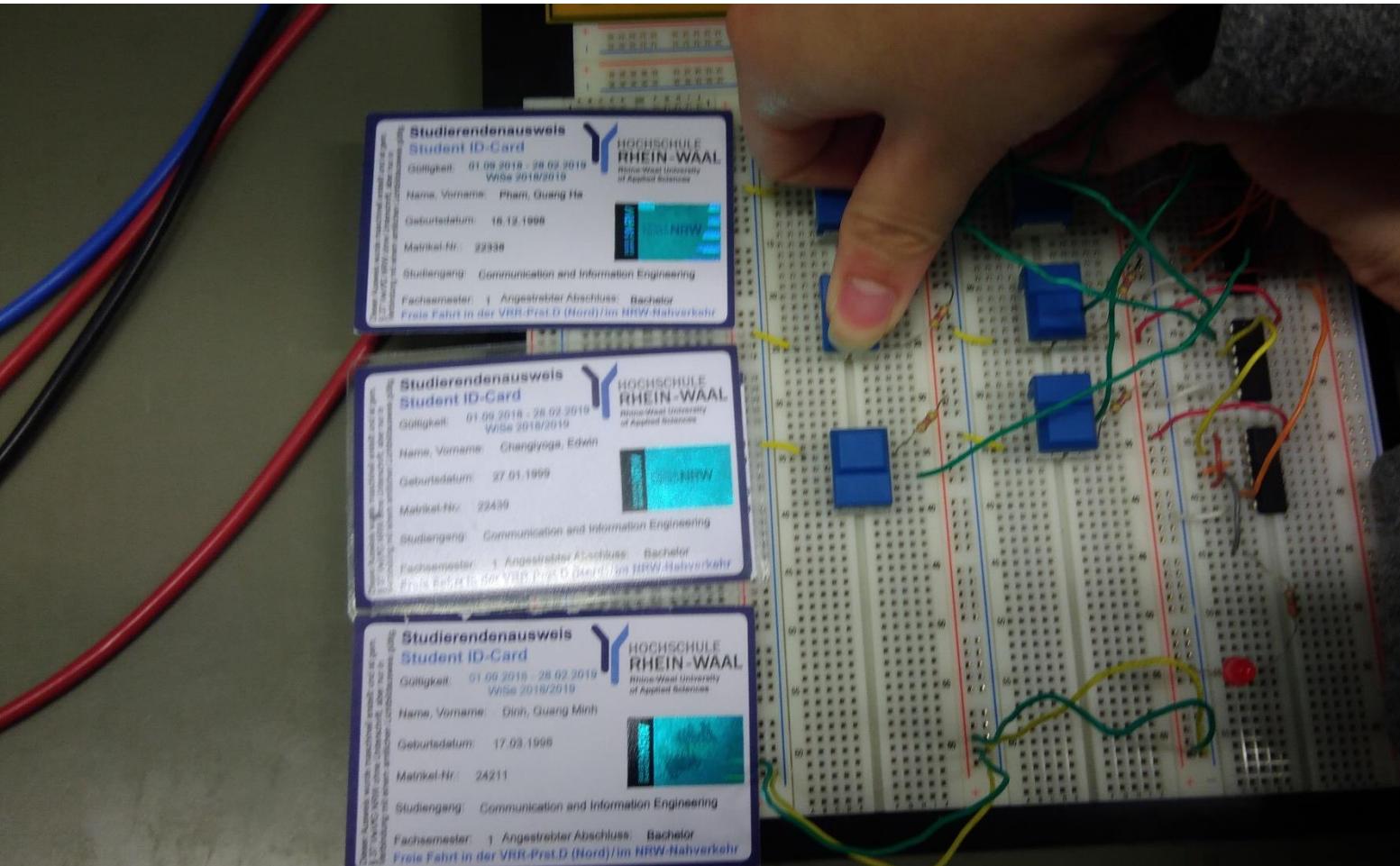


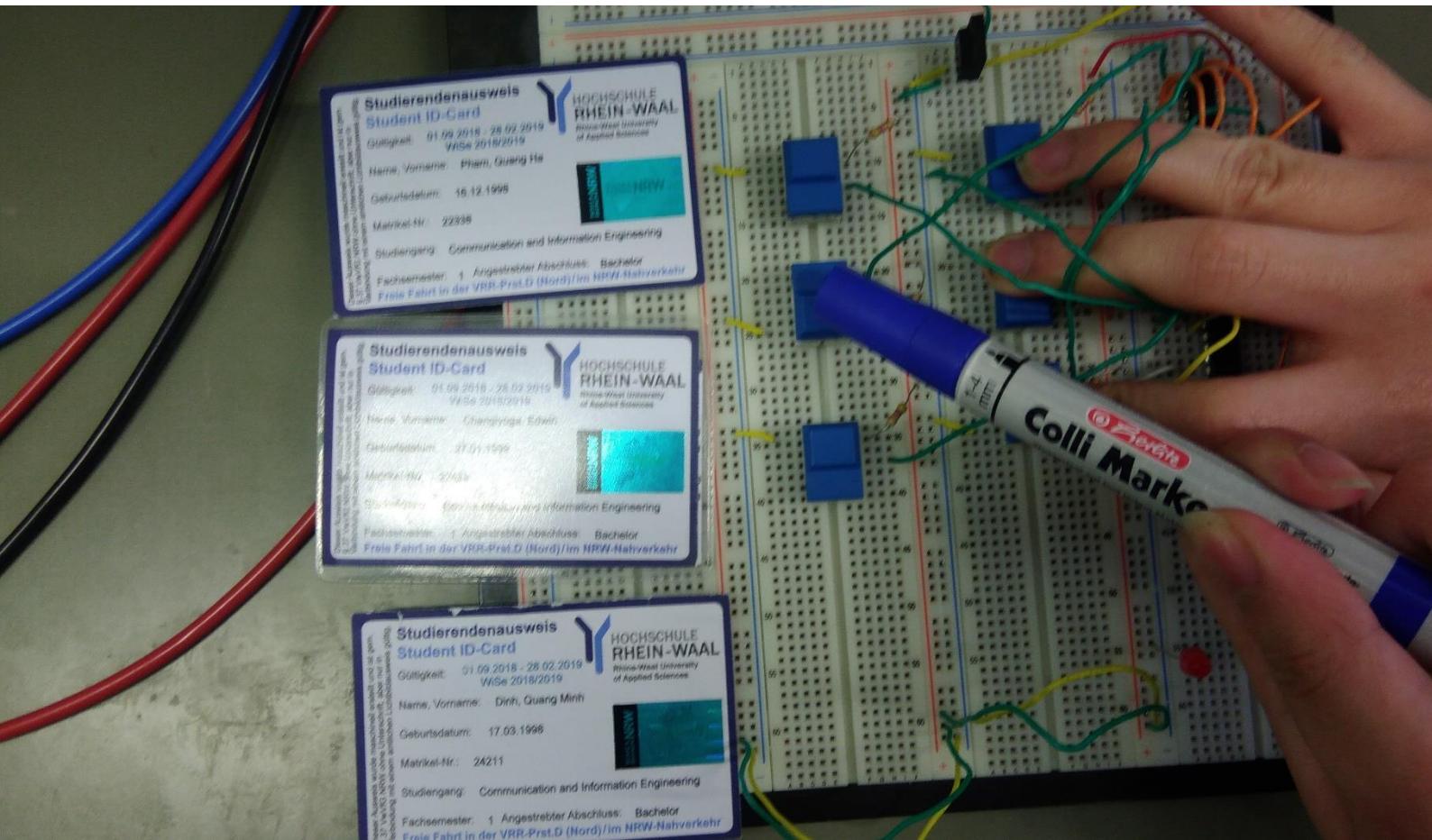




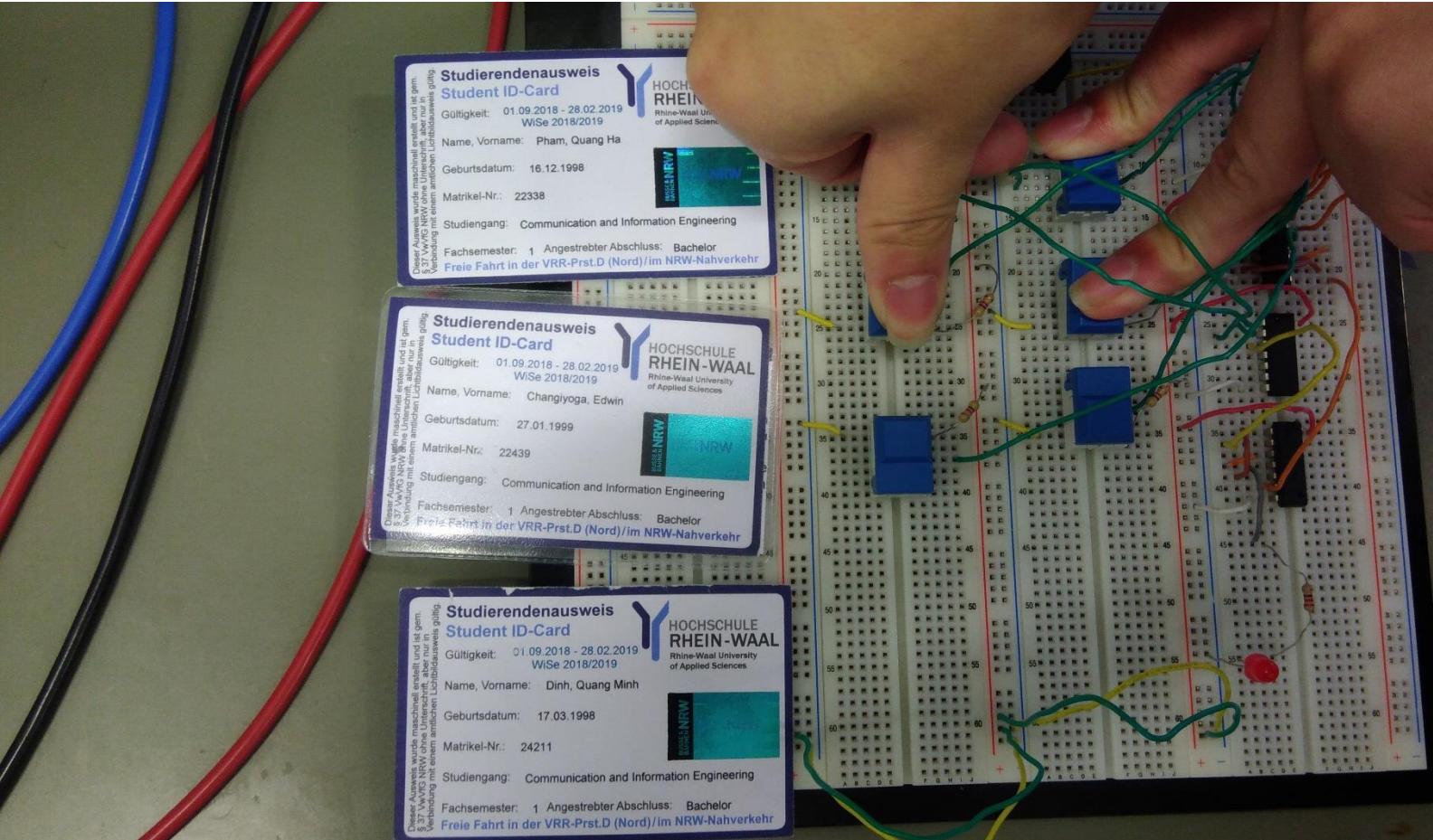


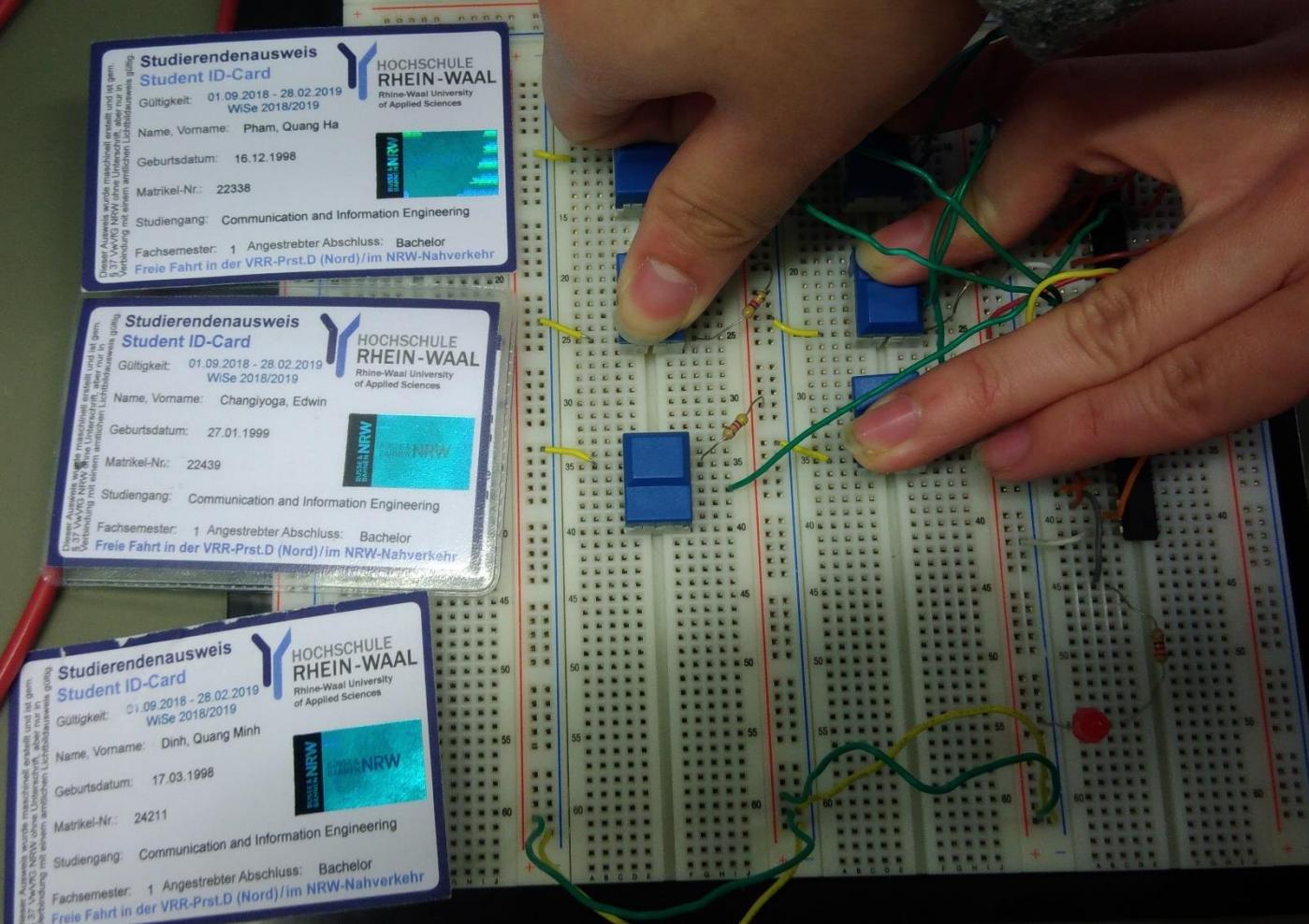












Result:

Since we have 6 inputs, in theory there should be 64 button combinations which will make our truth table very long. But according to the requirements, every time the Emergency button is actuated ($S_0=1$) the output will be $Q_1=0$, so we can ignore the 32 cases when $S_0=1$ and only focus on the 32 cases when $S_0=0$.

S0	S1	S2	B1	B2	F1	Q1
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	0
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	1	0	0	0	0
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	0
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	0	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	1	0	1	0	0	1
0	1	0	1	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	0	0
0	1	1	0	0	1	0
0	1	1	1	0	0	0
0	1	1	1	0	1	0
0	1	1	1	1	0	0
0	1	1	1	1	1	0

Challenge #10

Unfortunately, our group did not manage to come up with a solution for this challenge.