

Protocol Laboratory Digital Engineering #3

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

Study questions	3
Challenge #1	4
Challenge #2	11
Challenge #3	17
Challenge #4	23
Challenge #5	28
Challenge #6	33
Challenge #7	36
Challenge #8	42
Challenge #9	48
Challenge #10	53
Challenge #11	59

Study questions

Question #1 – What is the difference between a Seven-Segment Display with common anode and common cathode?

All anodes/cathodes of the LED are connected to each other and then connected to Vcc/Ground respectively.

Question #2 – Why is it necessary to debounce the switch for the BCD counter?

If we don't debounce, it will mess up the counting sequence by registering many small inputs.

Question #3 – Describe a typical application for a 3-Line to 8-Line decoder used in a microprocessor system?

Memory chip select decoding. Example being Challenge #4 in this module: using 1 74LS138 IC, 3 switches for inputs and 8 LEDs for output, we are able to indicate the output line that is selected. Using this setup saves material by reducing the number of buttons and ICs needed instead of using multiple AND and OR gates to replicate the 3-Line to 8-Line decoder or use 8 different switches for 8 output LEDs.

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.

When we first started, we struggled to get the 7-segment display to work, after a while we noticed that the schematic given in the Description in Challenge #1 is incorrect. So, we switched to the schematic in **Figure 1** below and finished the challenge.

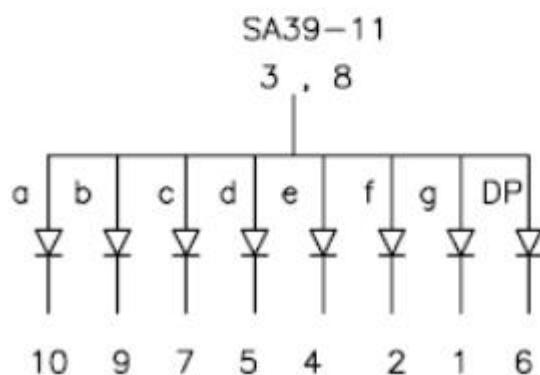
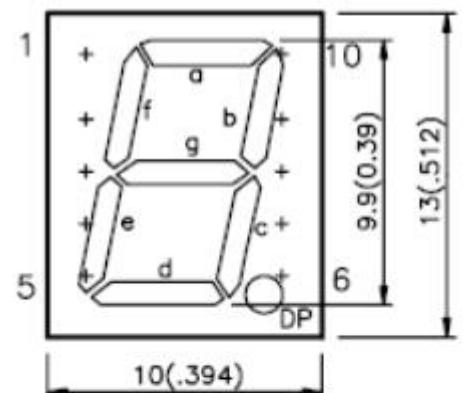
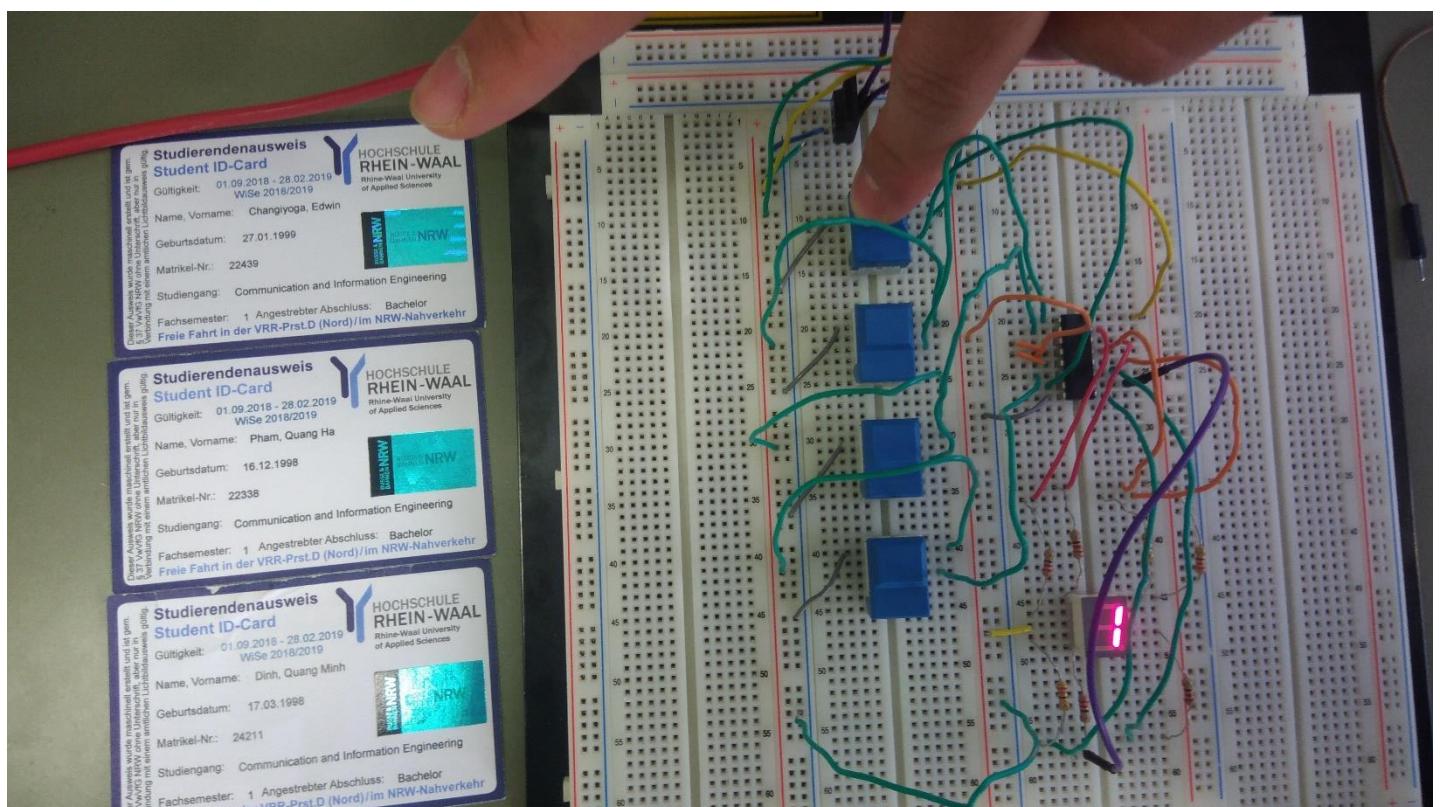
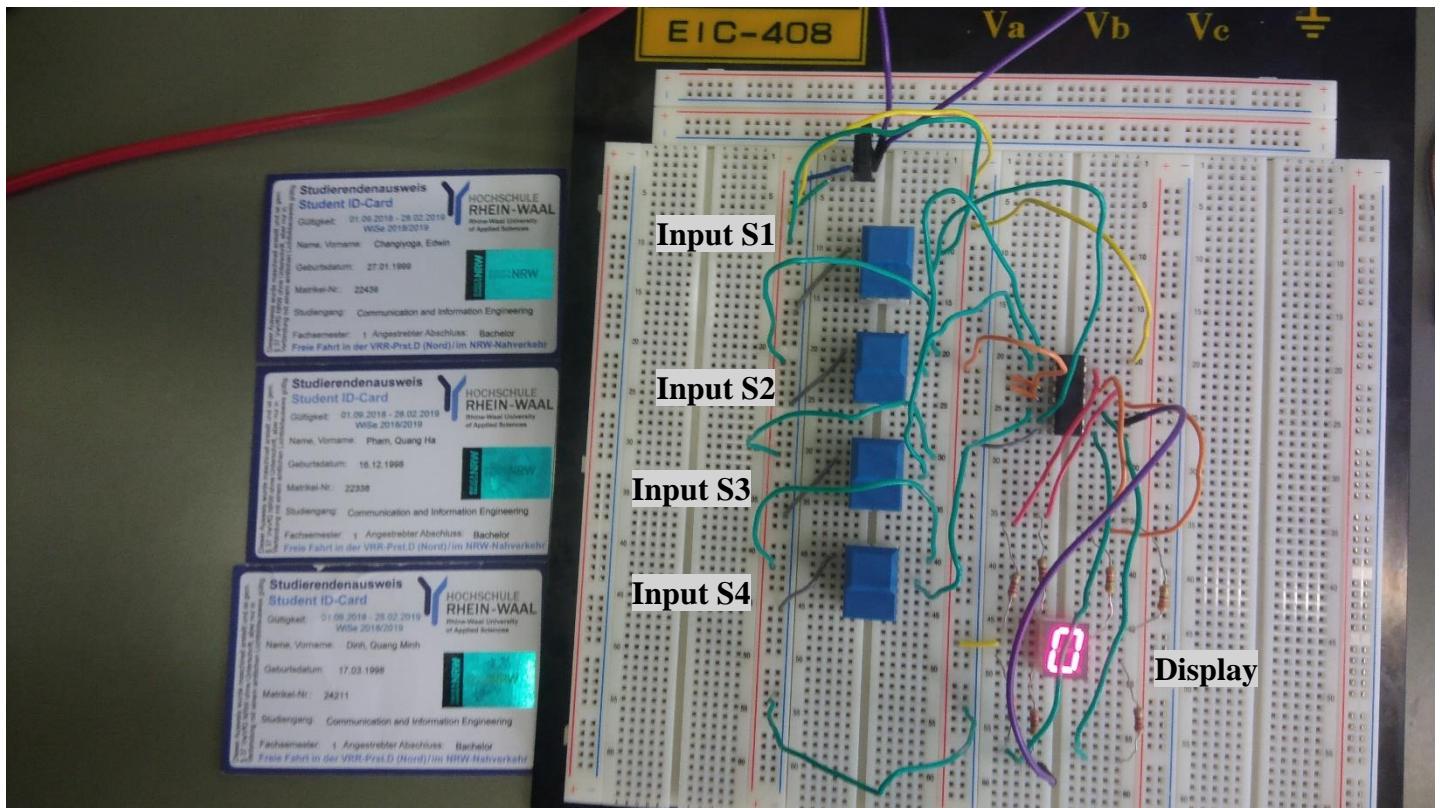
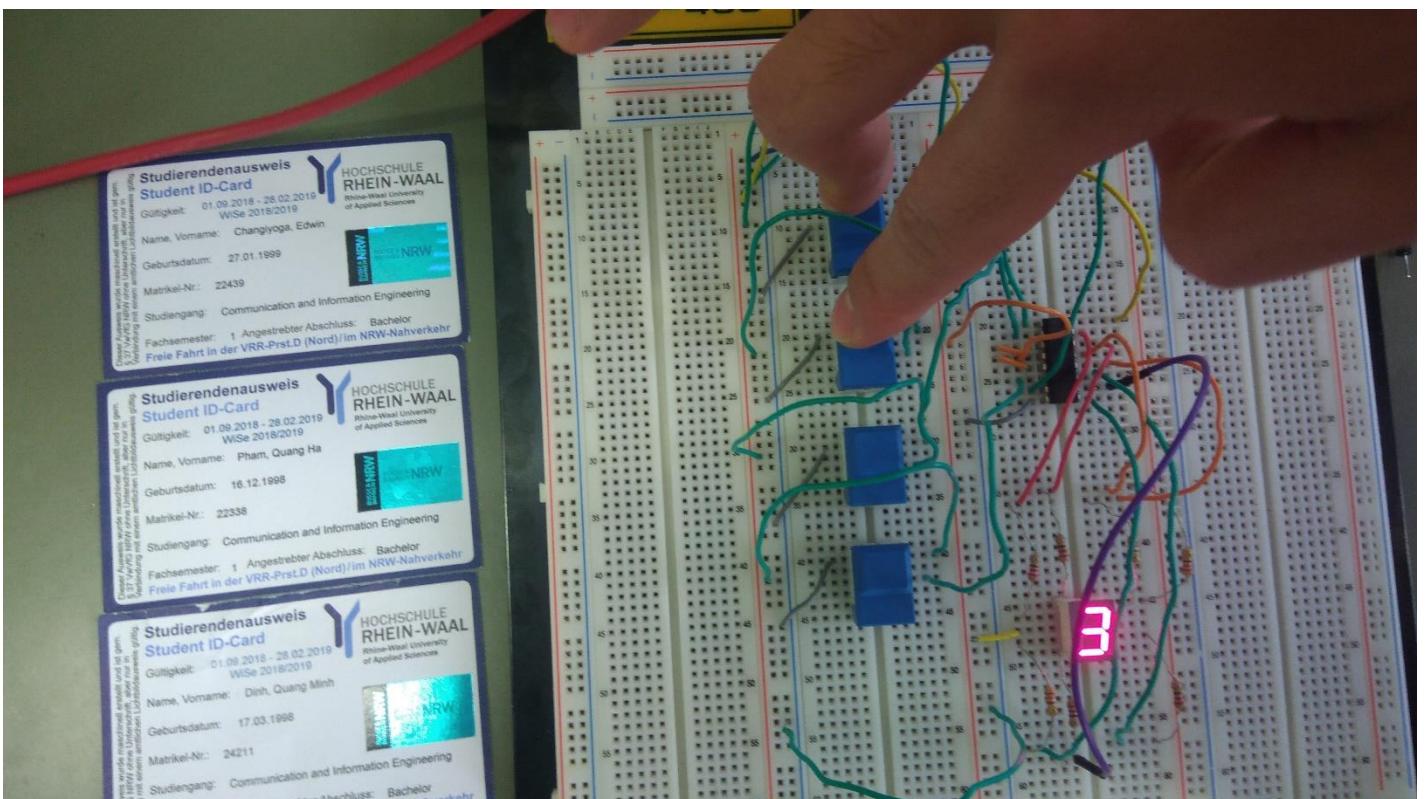
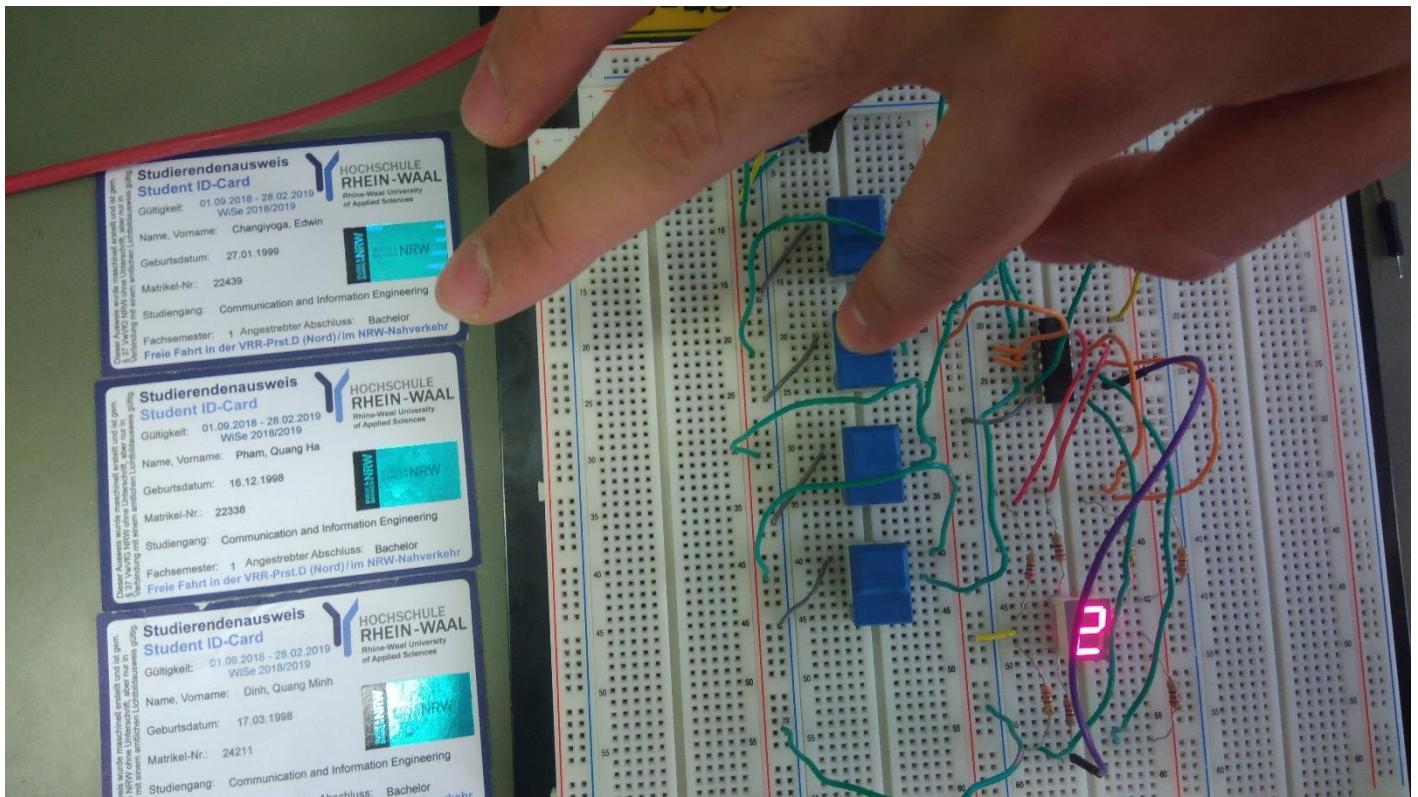
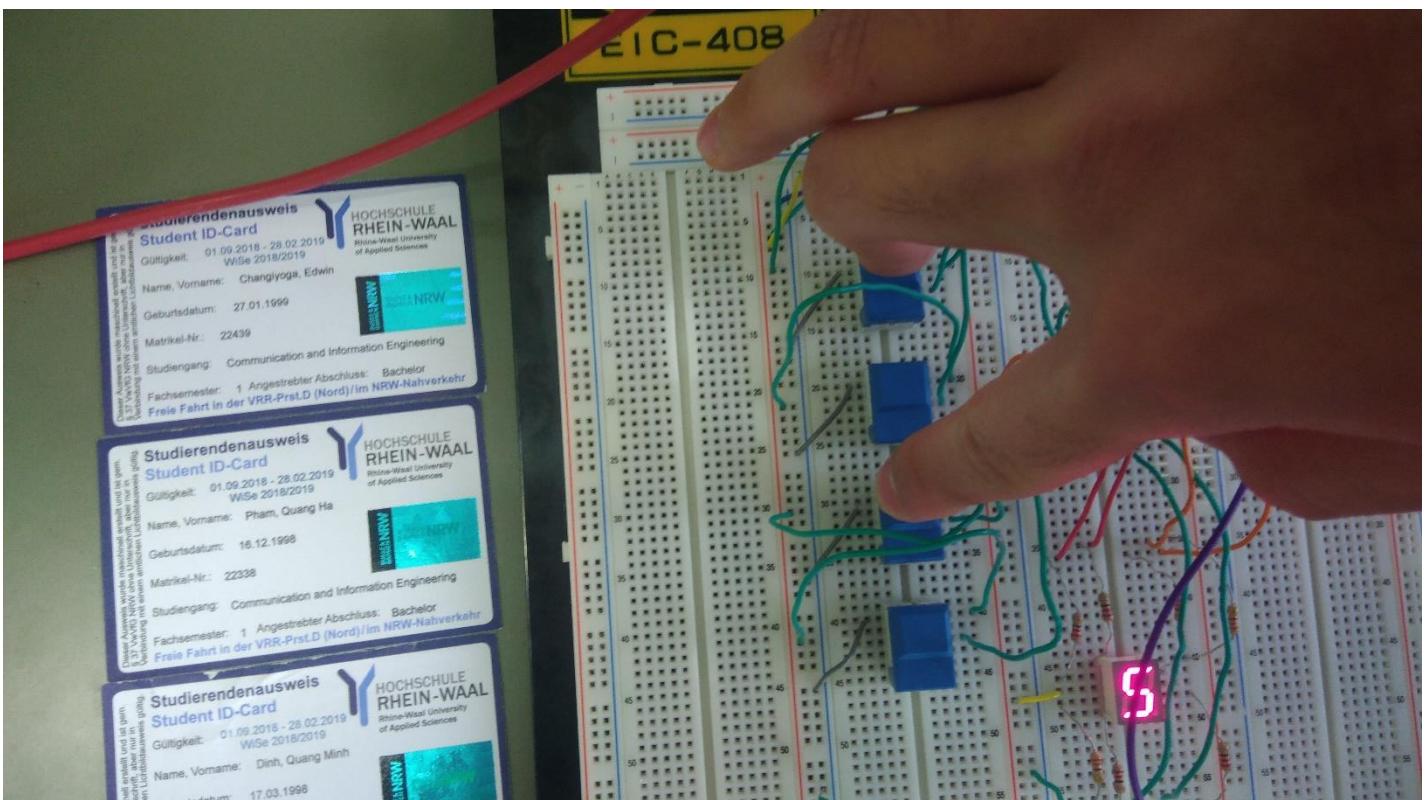
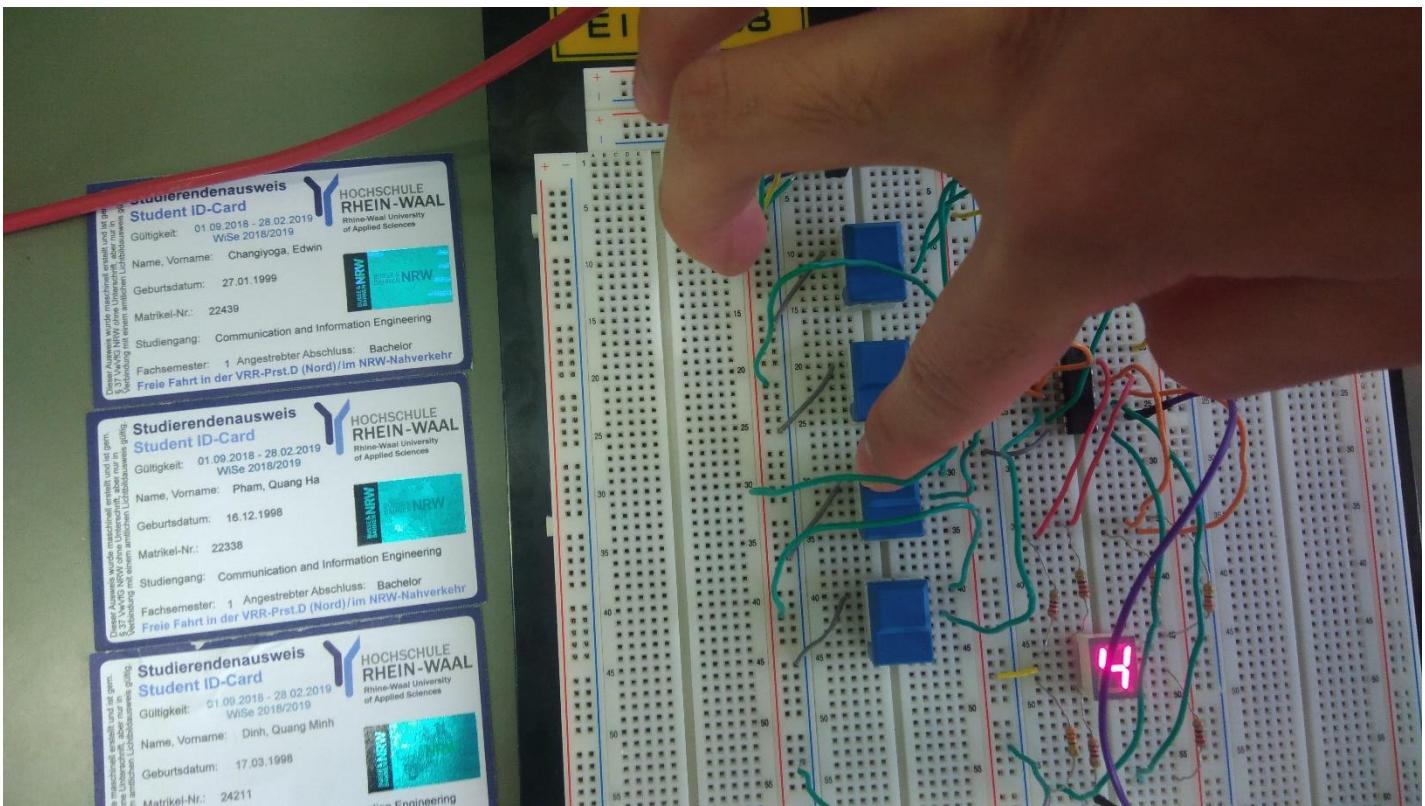


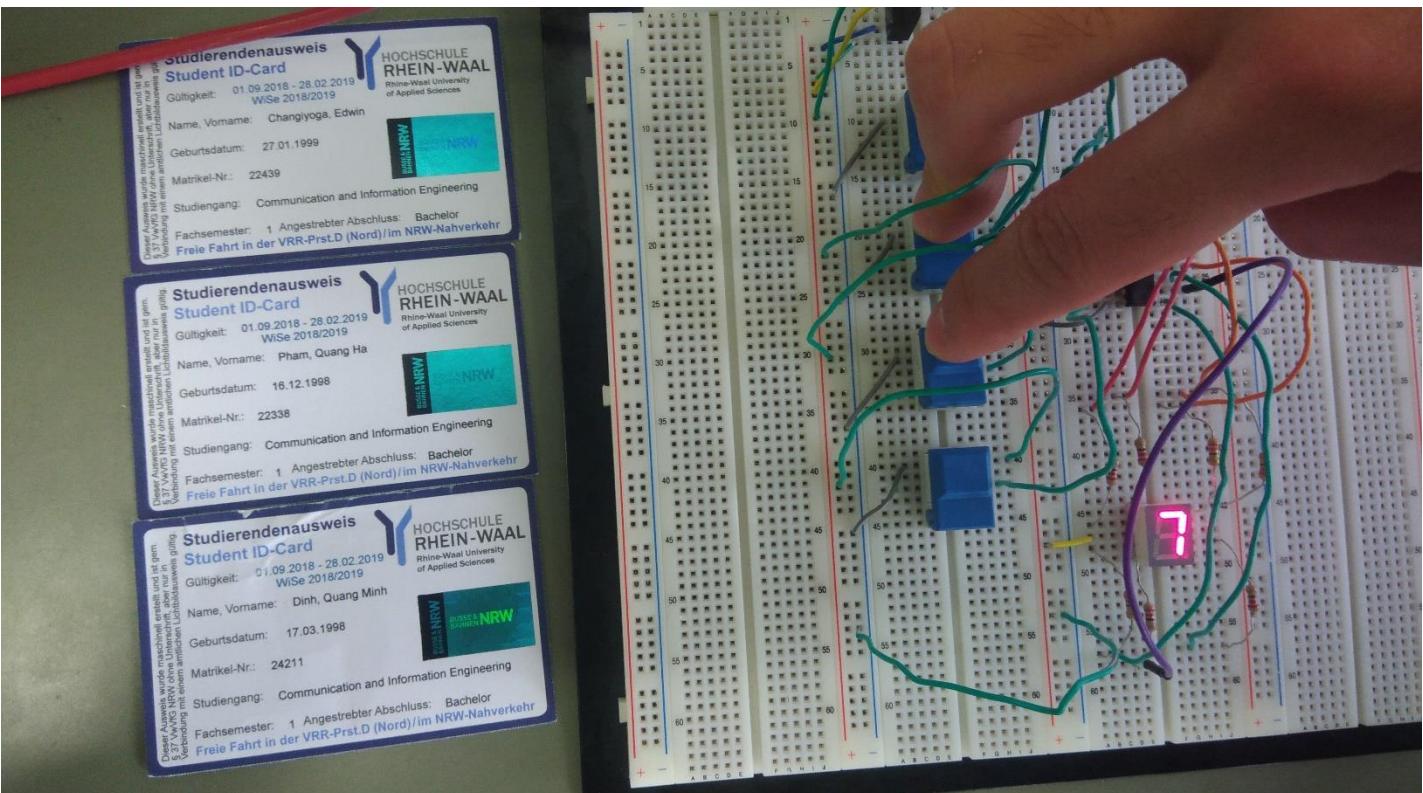
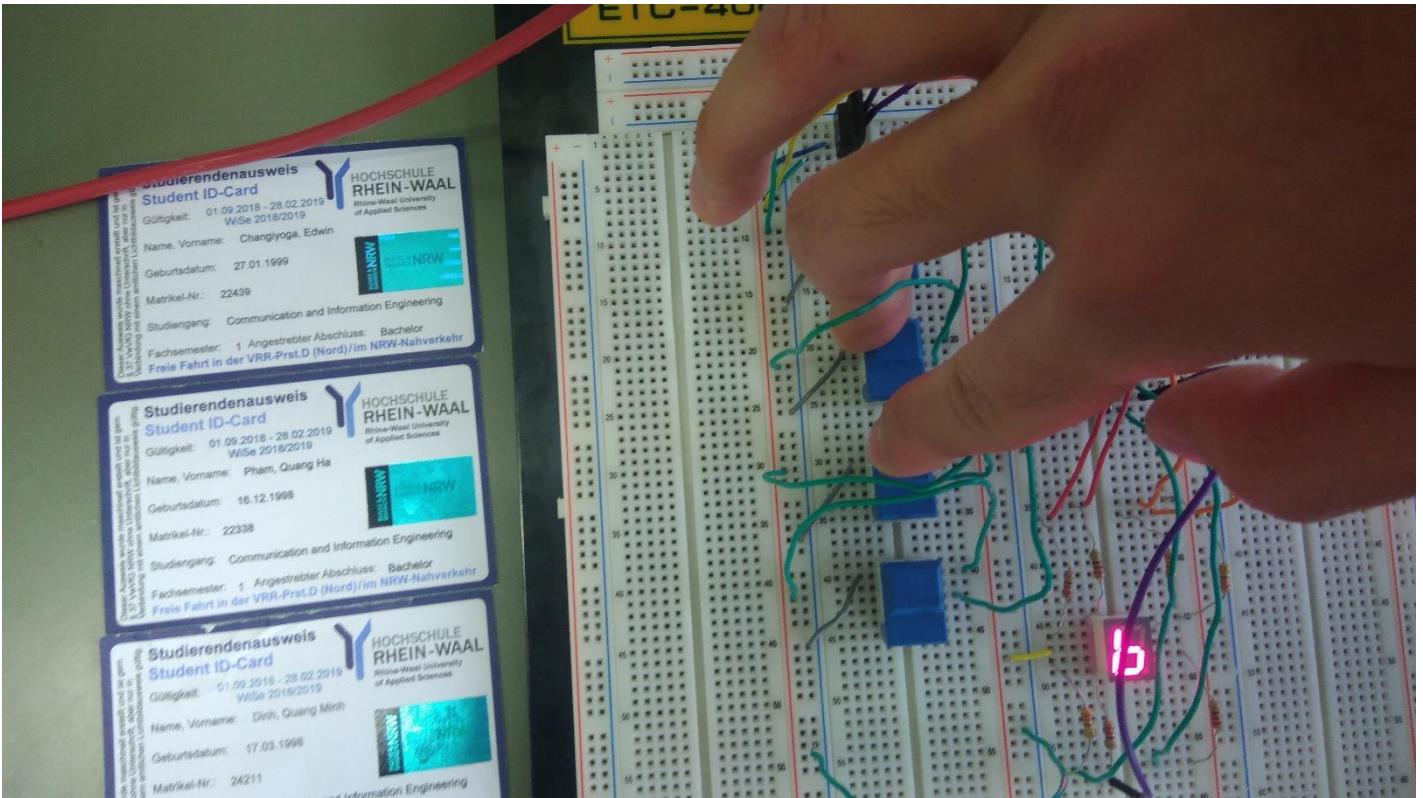
Figure 1: Correct 7-segment schematic

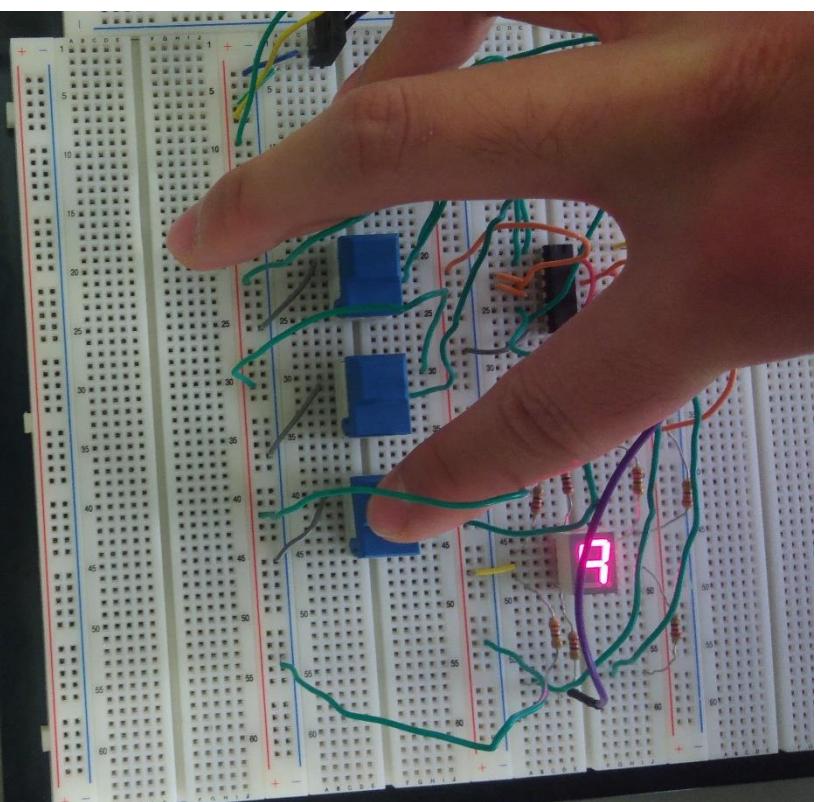
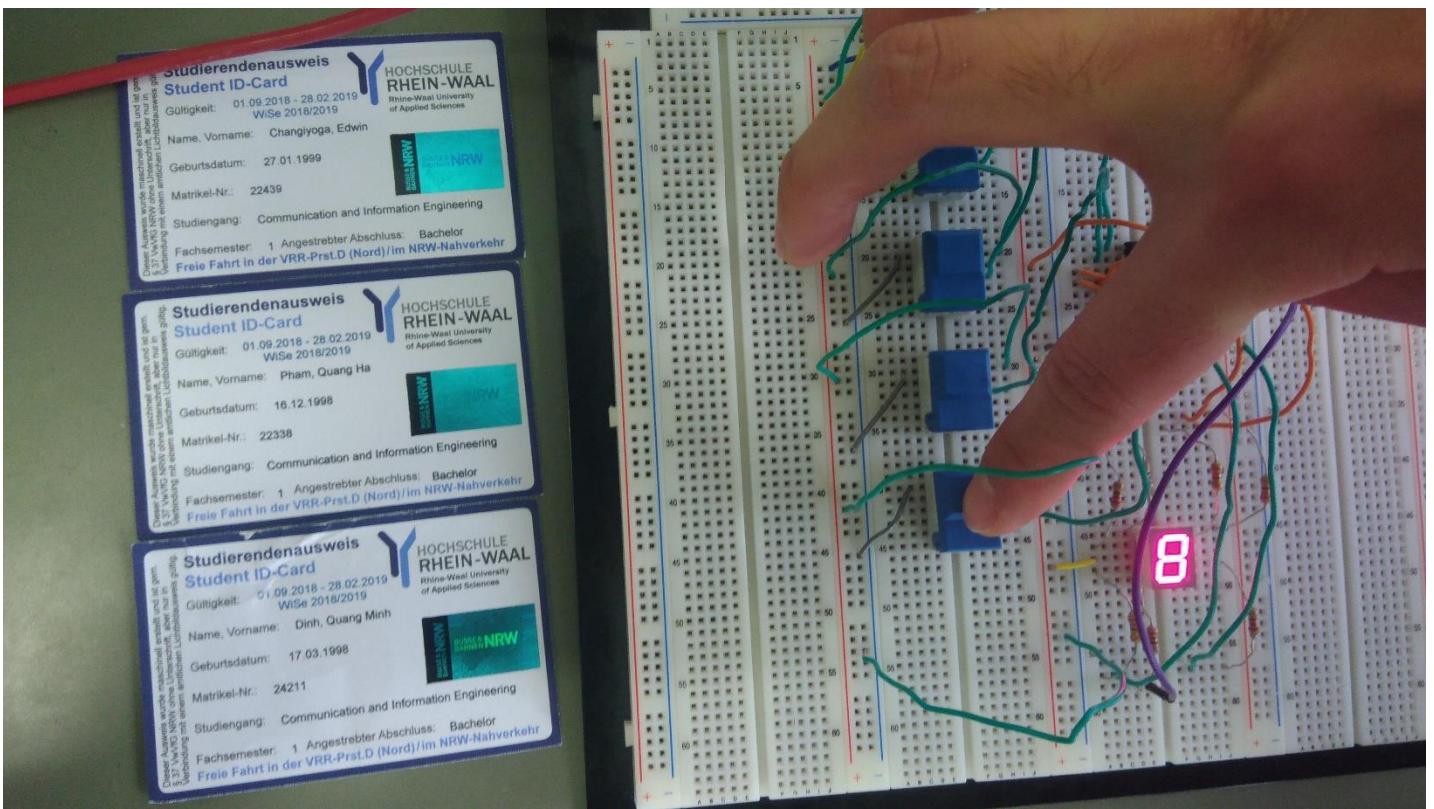
Pictures:











Result:

0: Switch not pressed

1: Switch pressed

S4	S3	S2	S1	Display
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

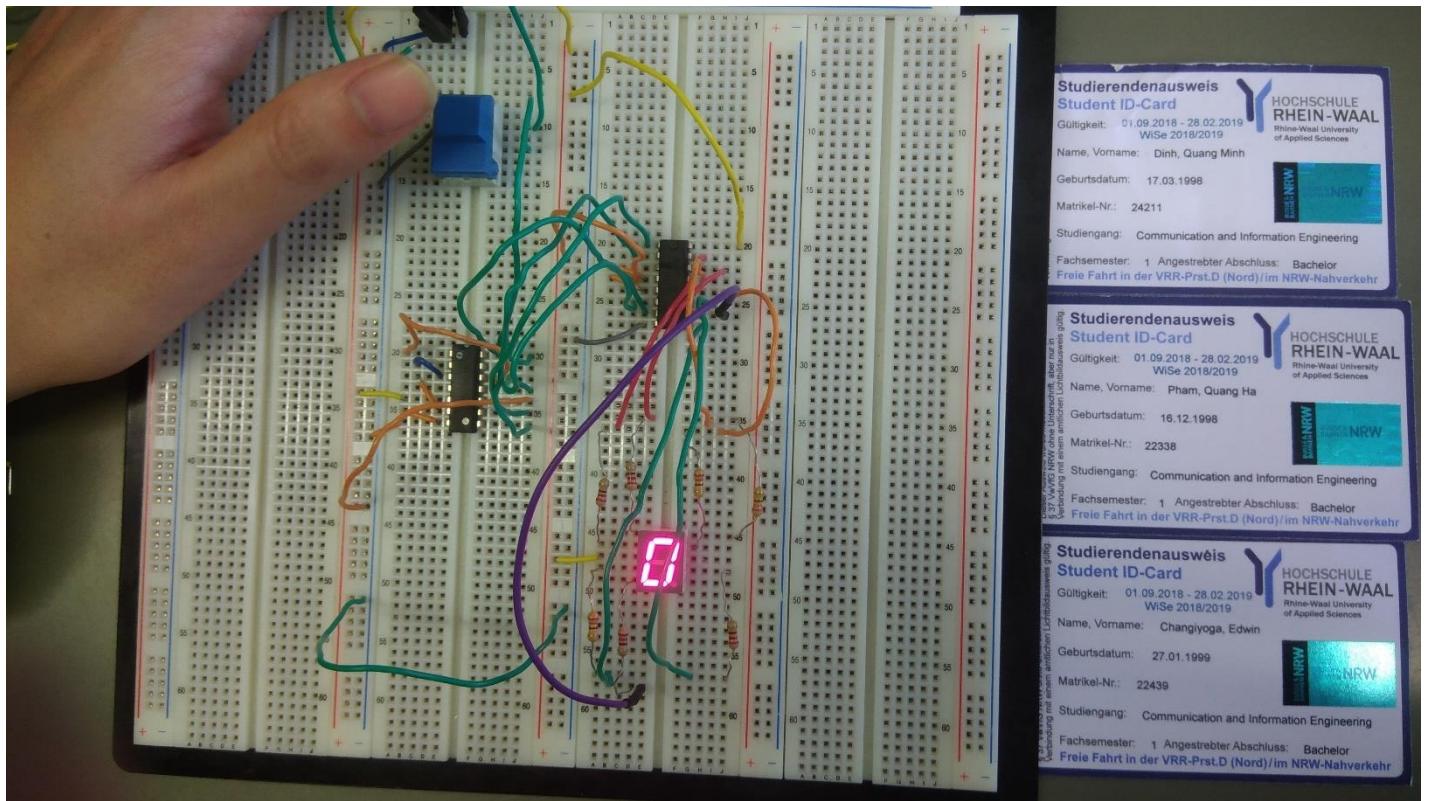
Challenge #2

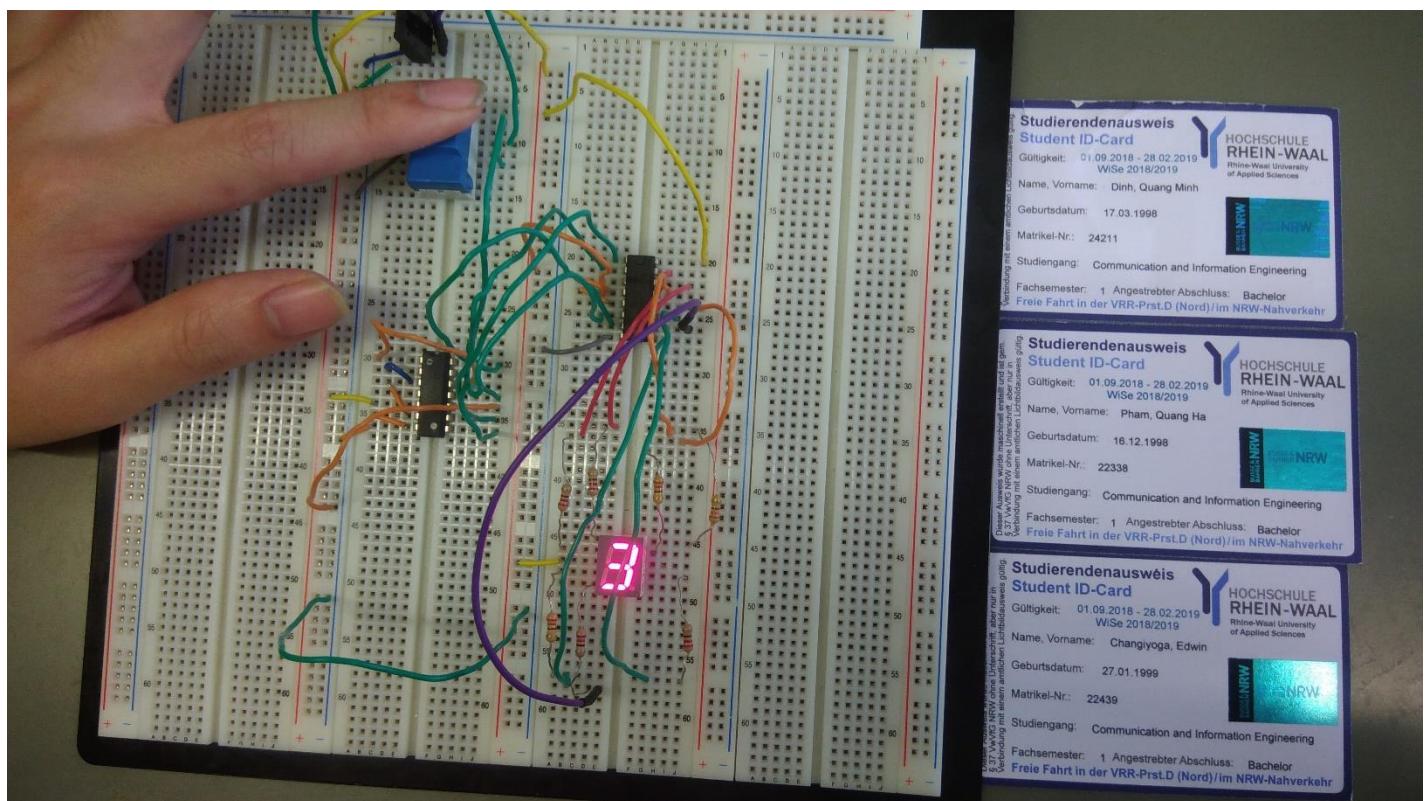
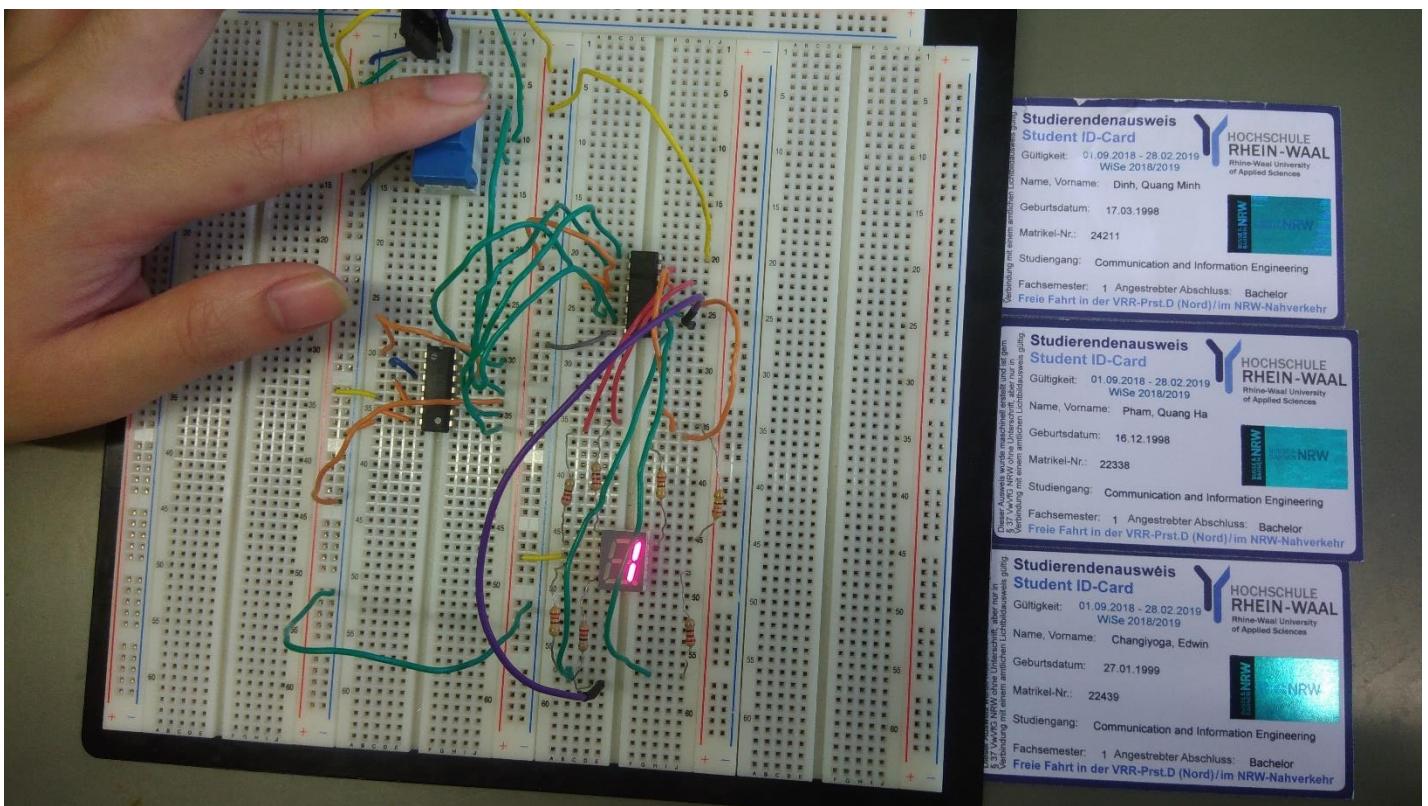
Abstract:

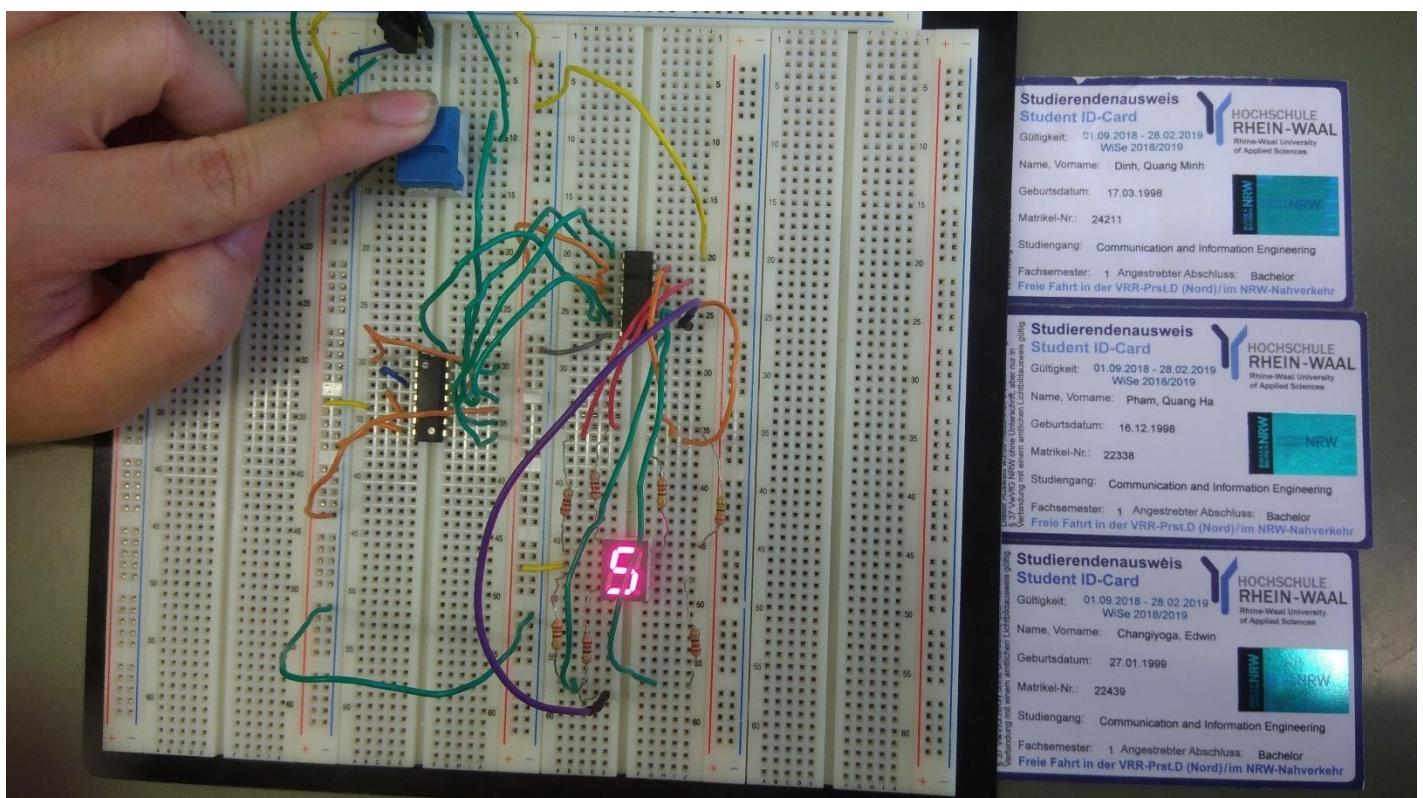
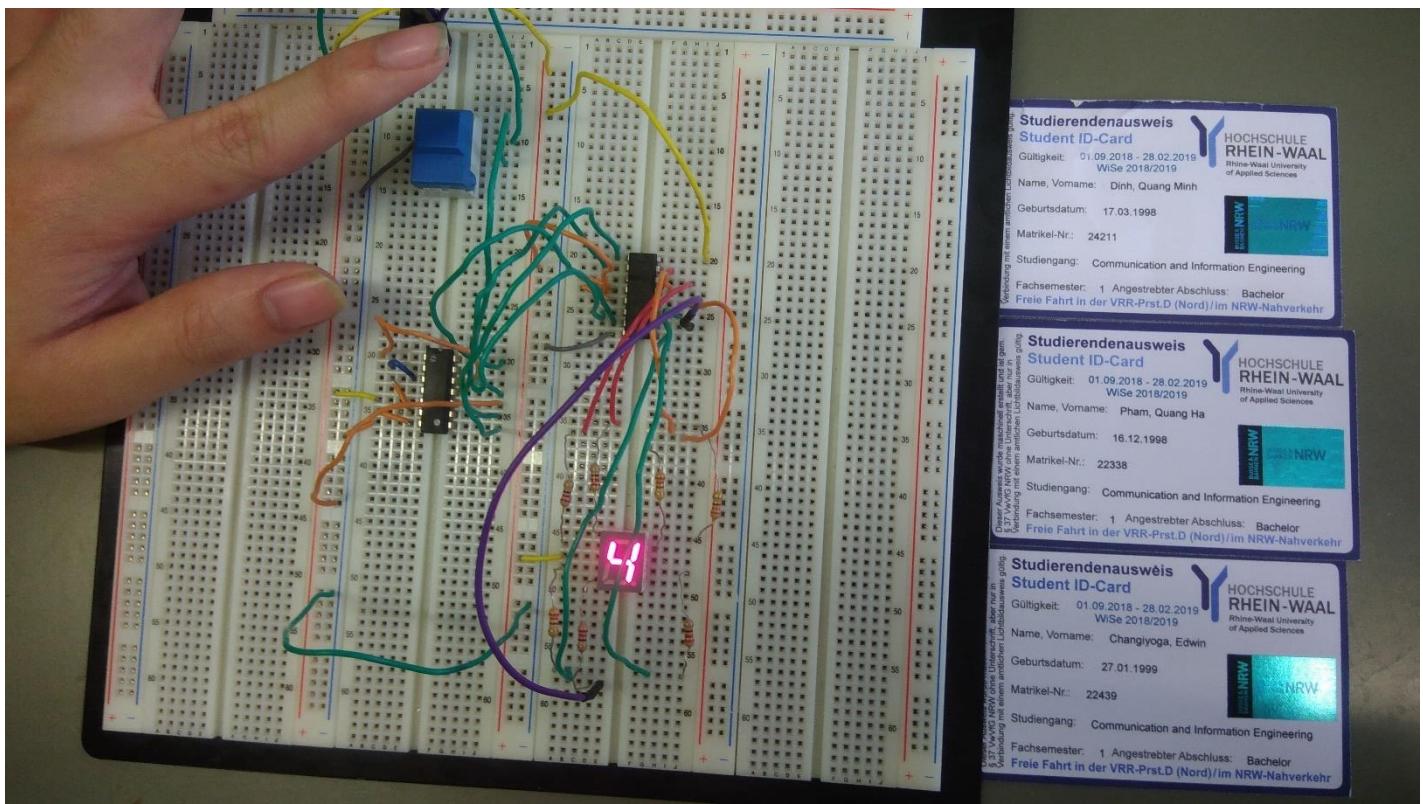
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

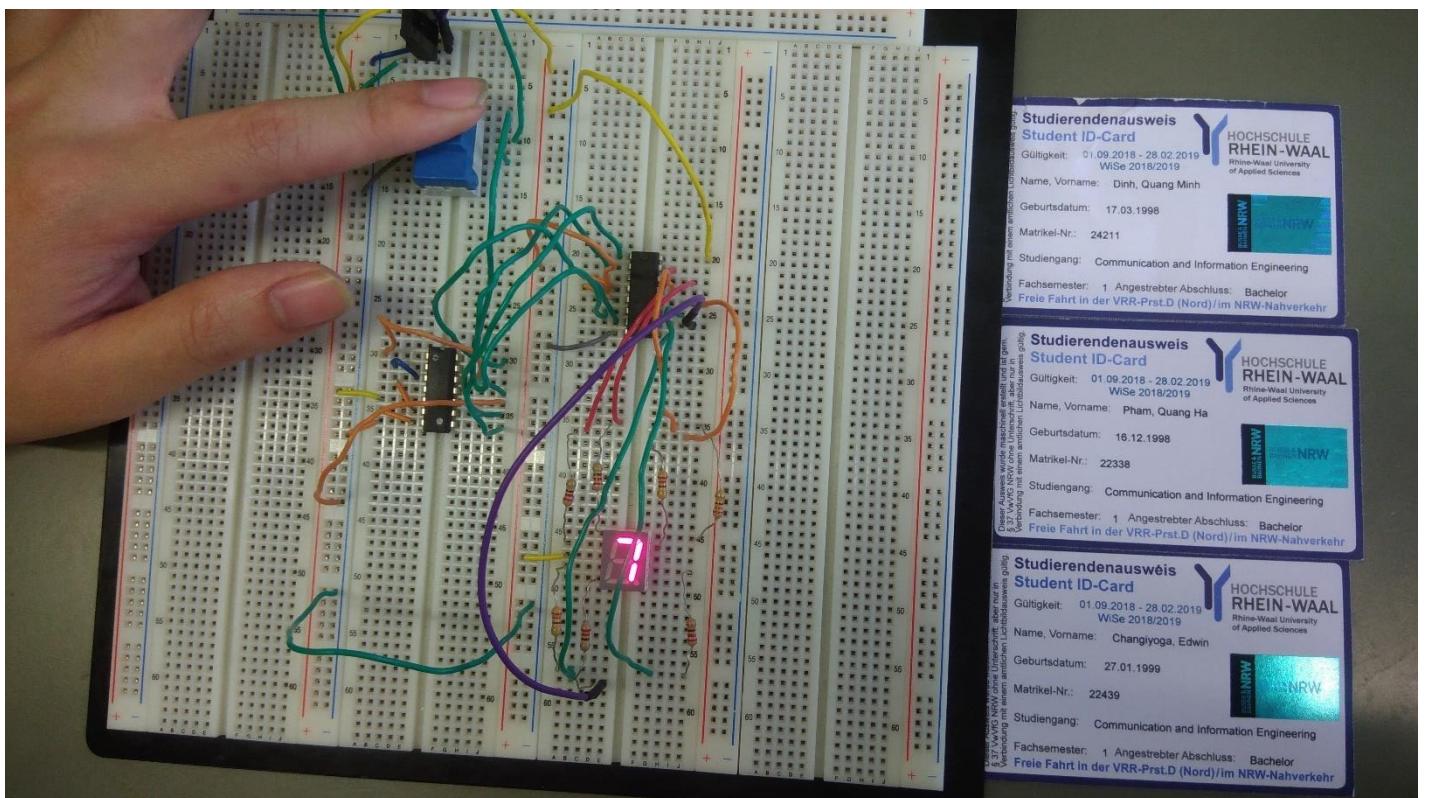
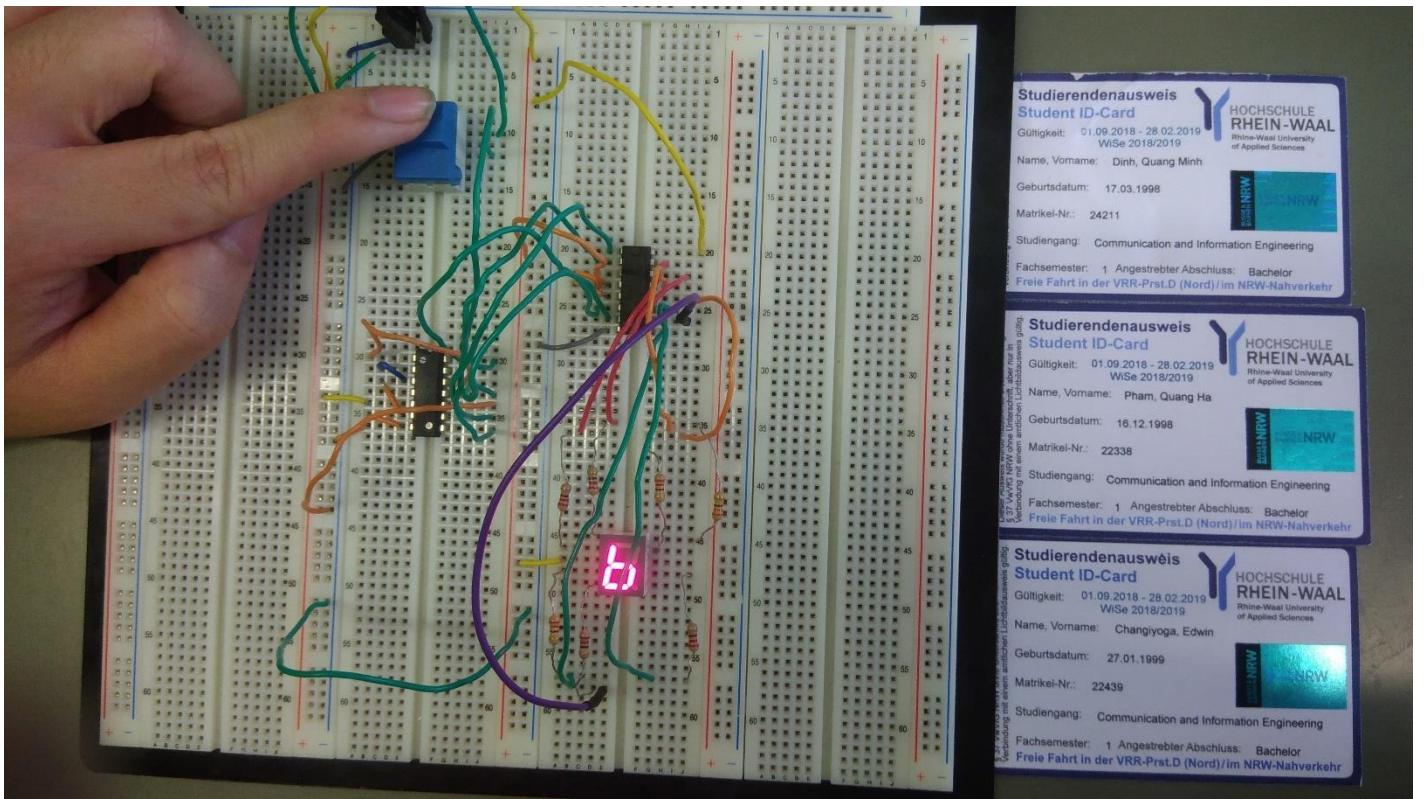
Since the button is not properly debounced, each time we press the button results in the display counter jumps erratically instead of the intended function: counter goes up by only one unit for each press. We only managed to take pictures of all numbers from 0 to 8 except for 2 and two unintended results.

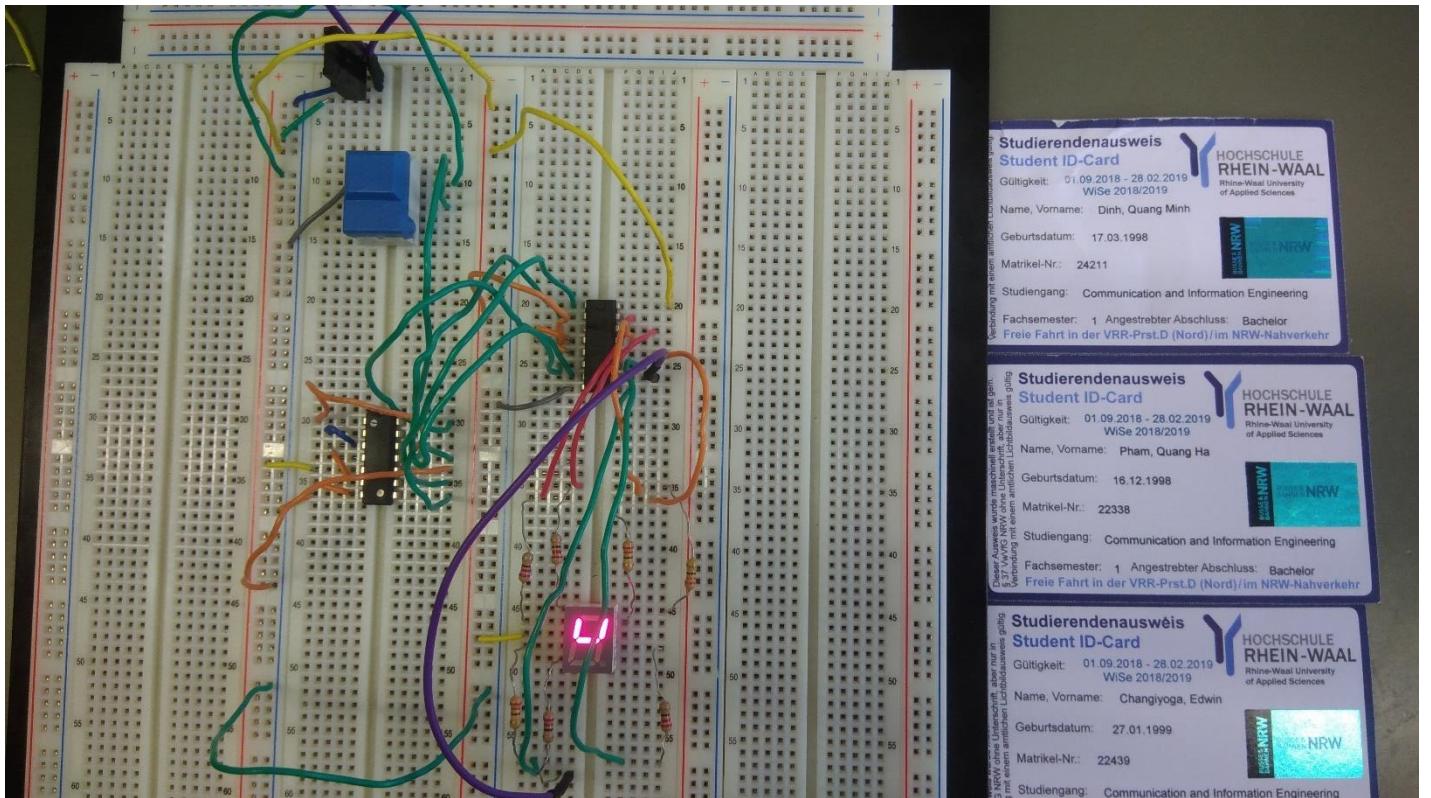
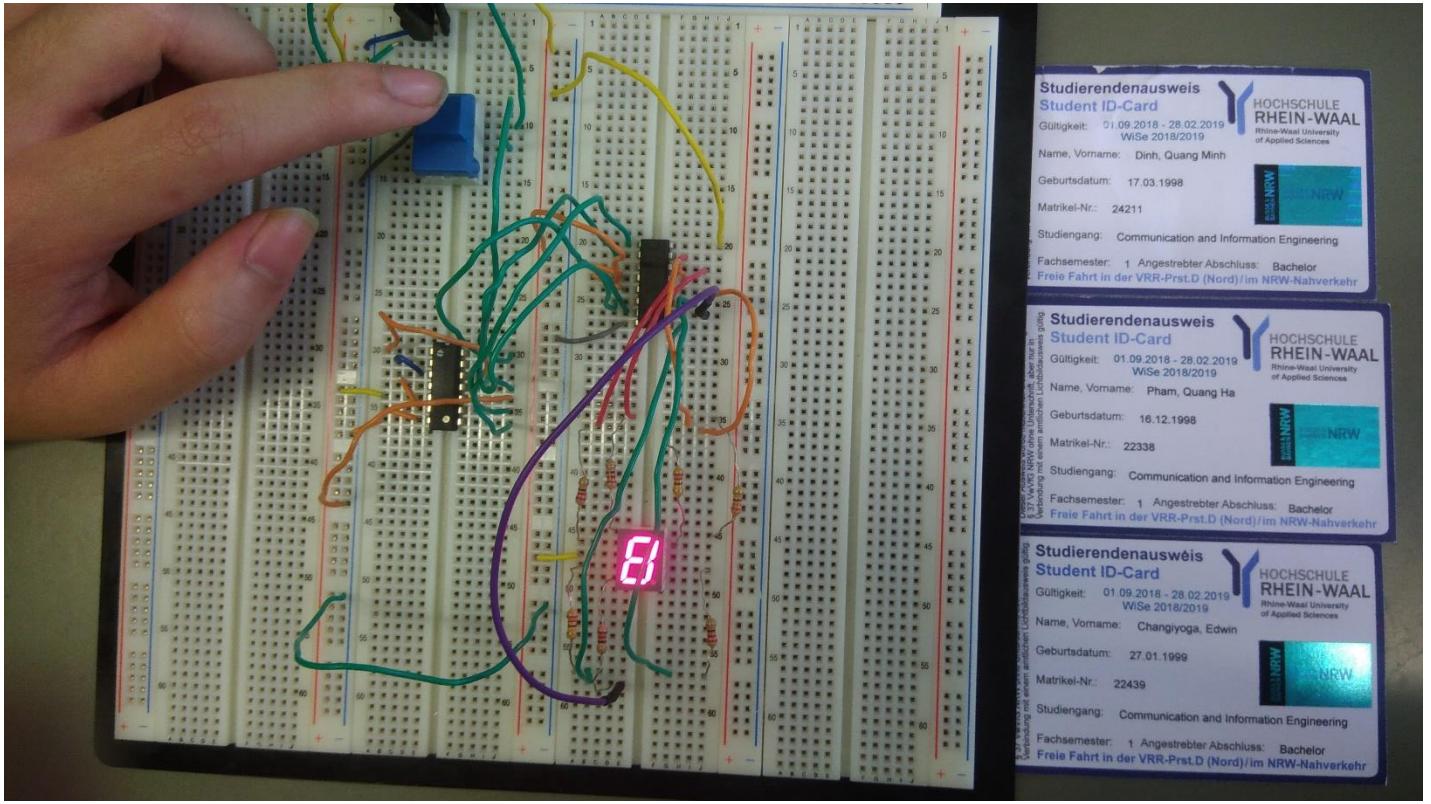
Pictures:

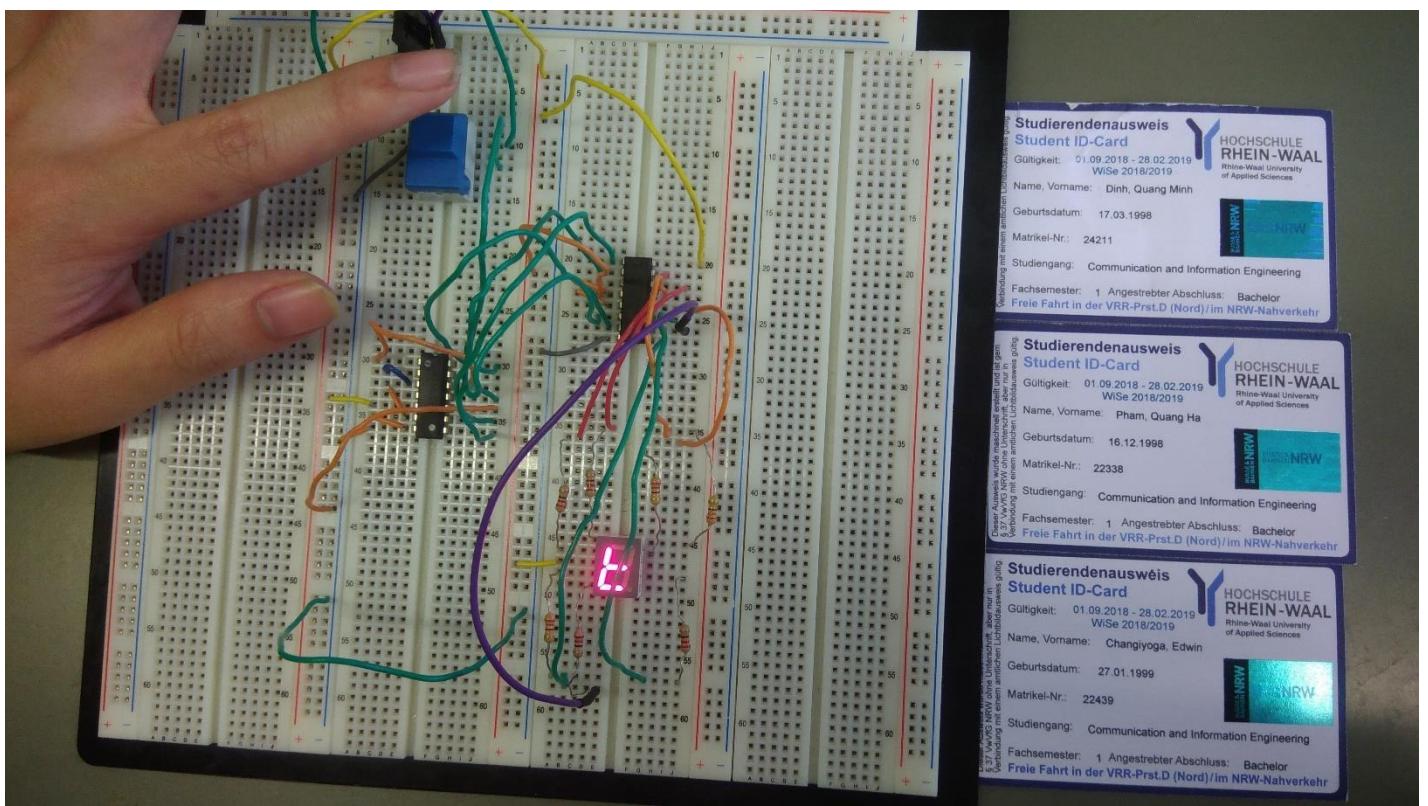












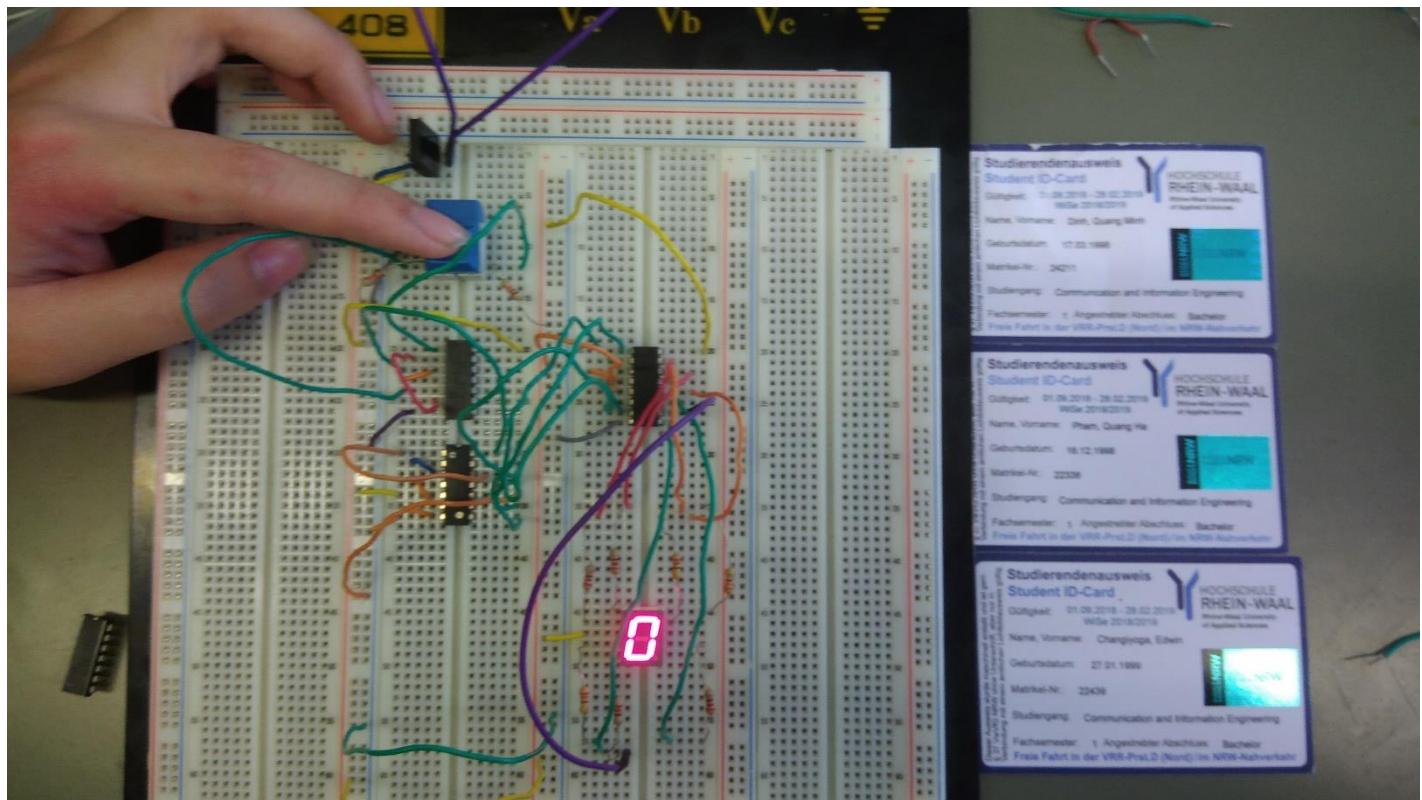
Challenge #3

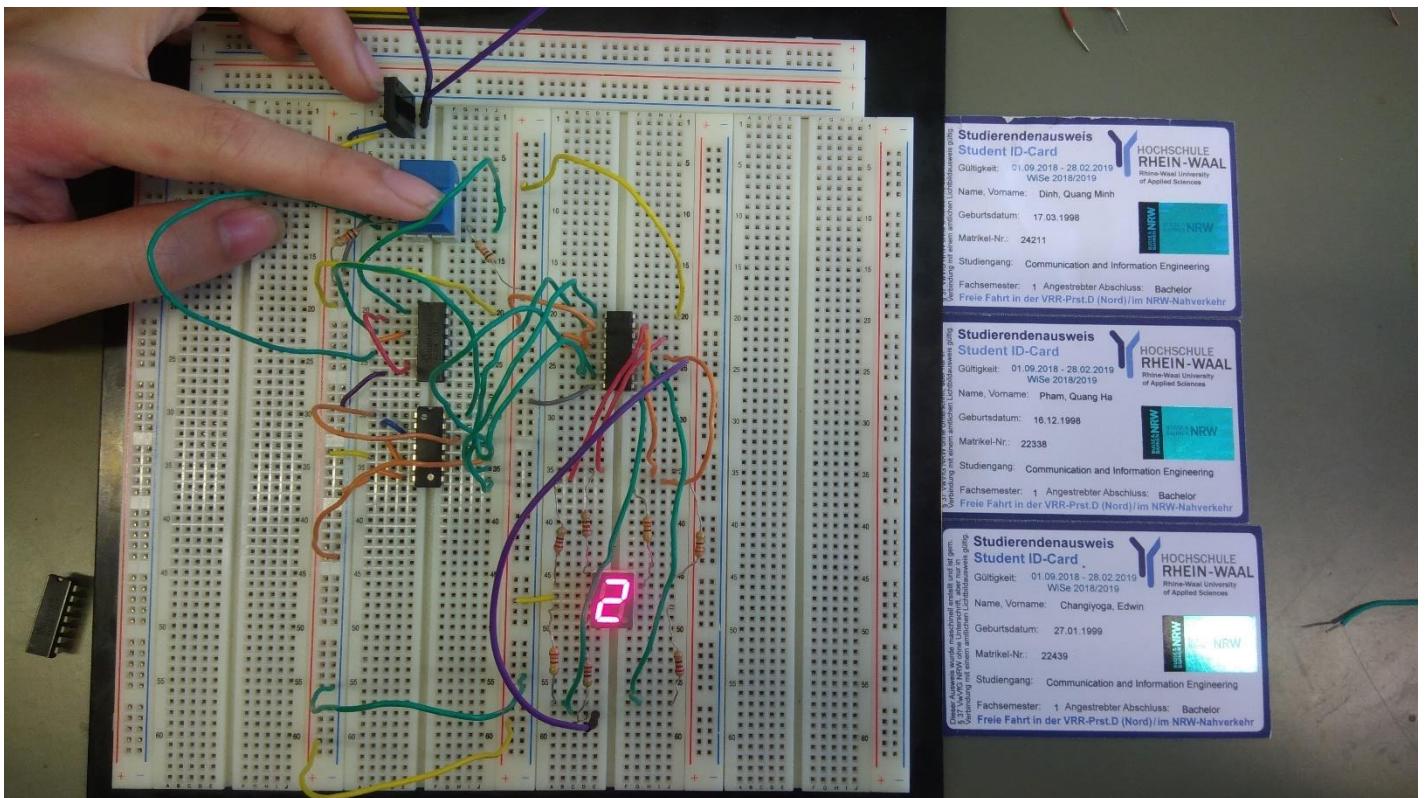
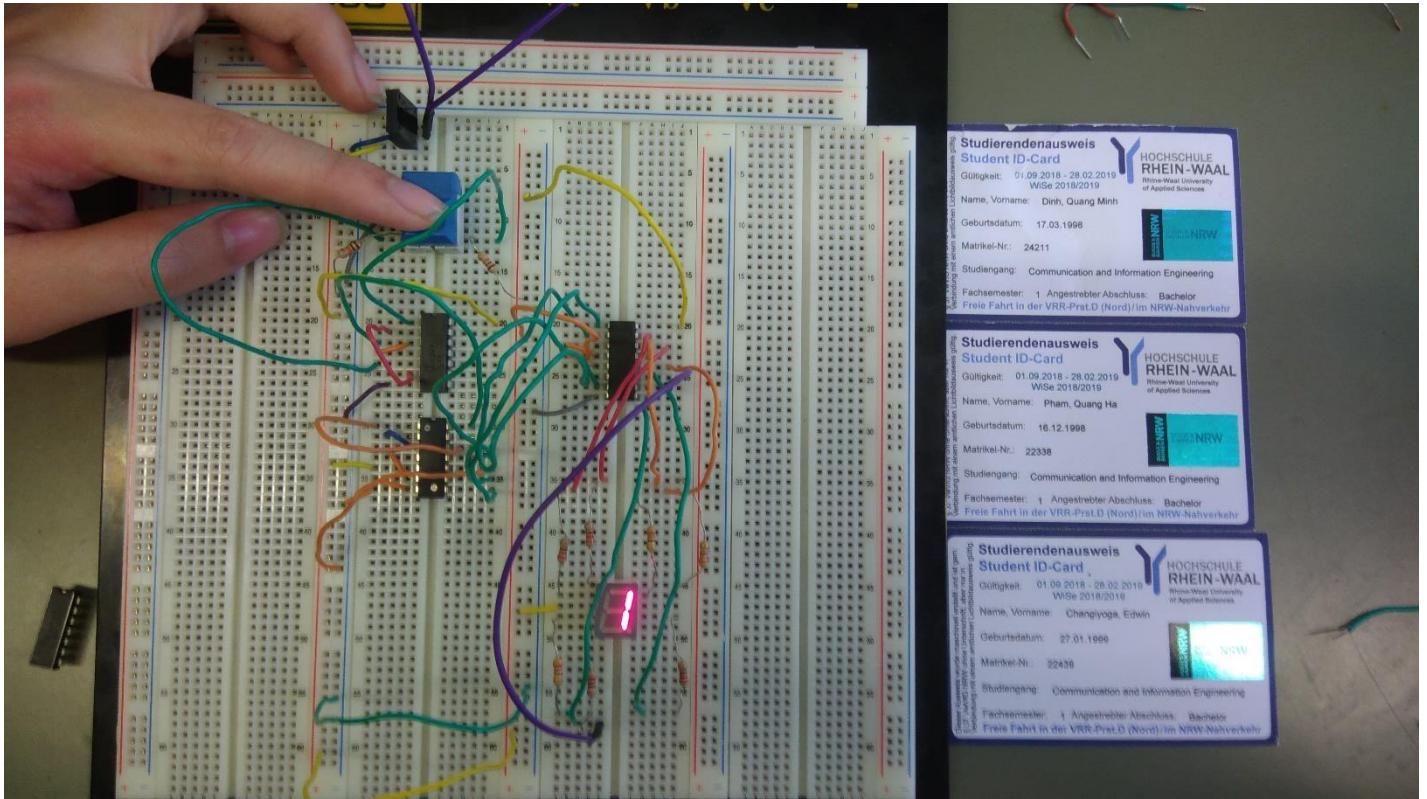
Abstract:

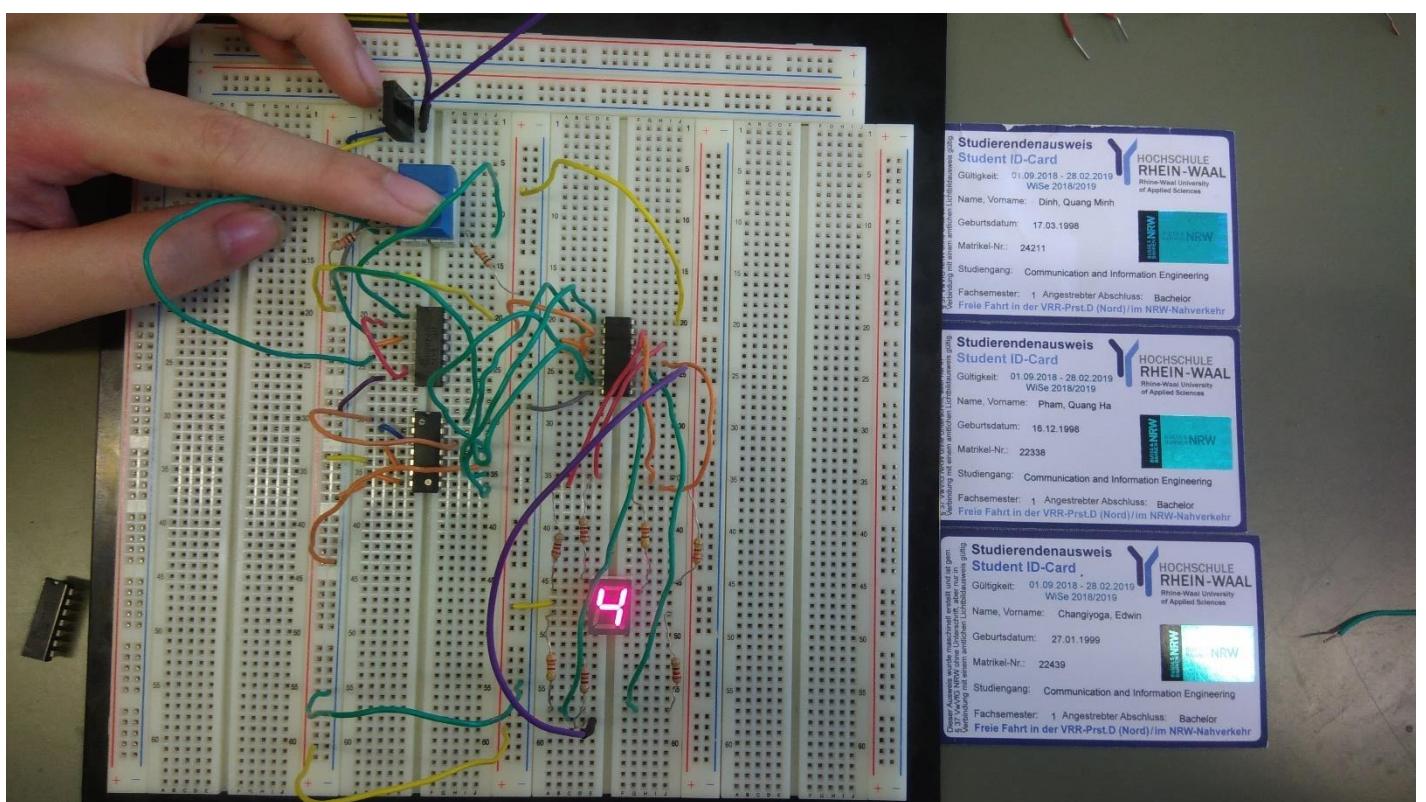
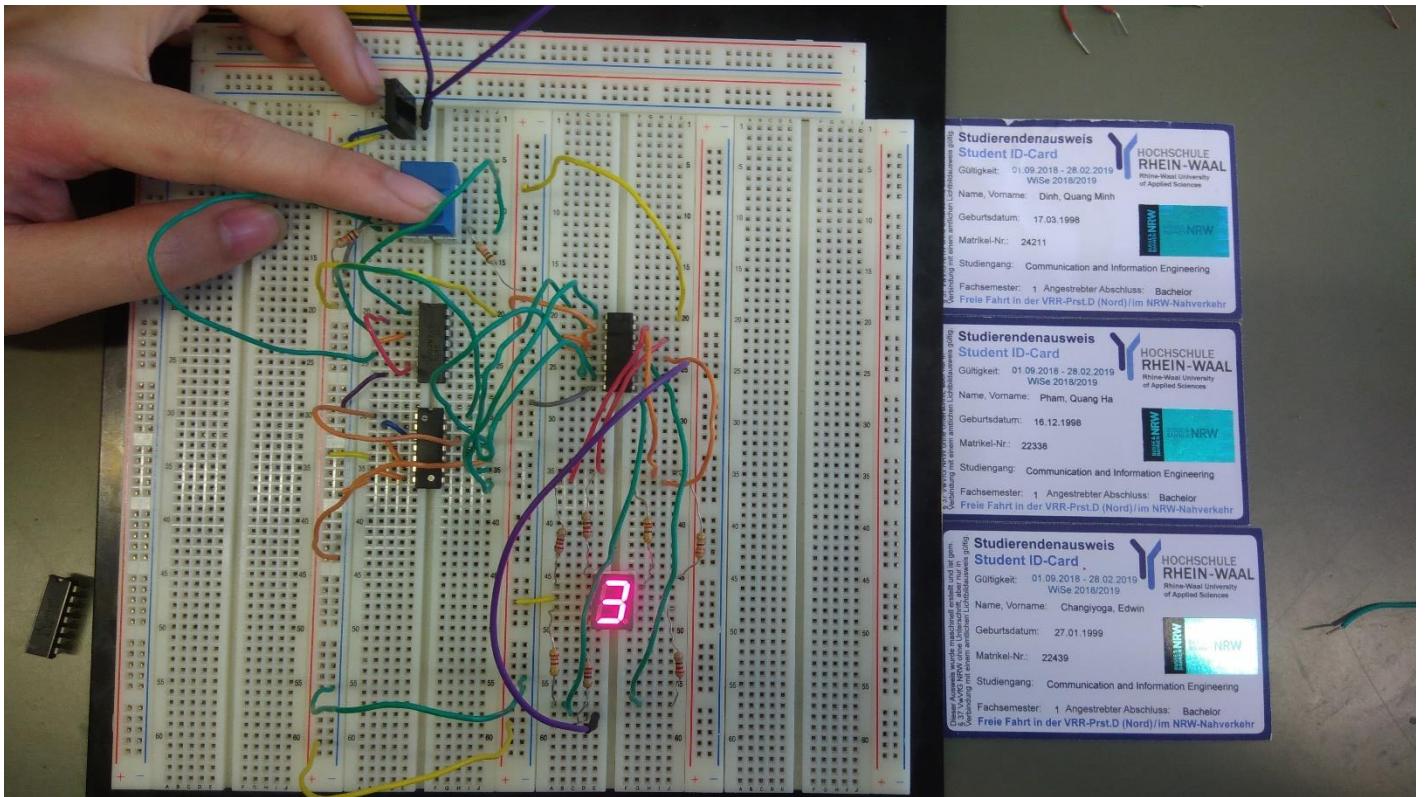
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

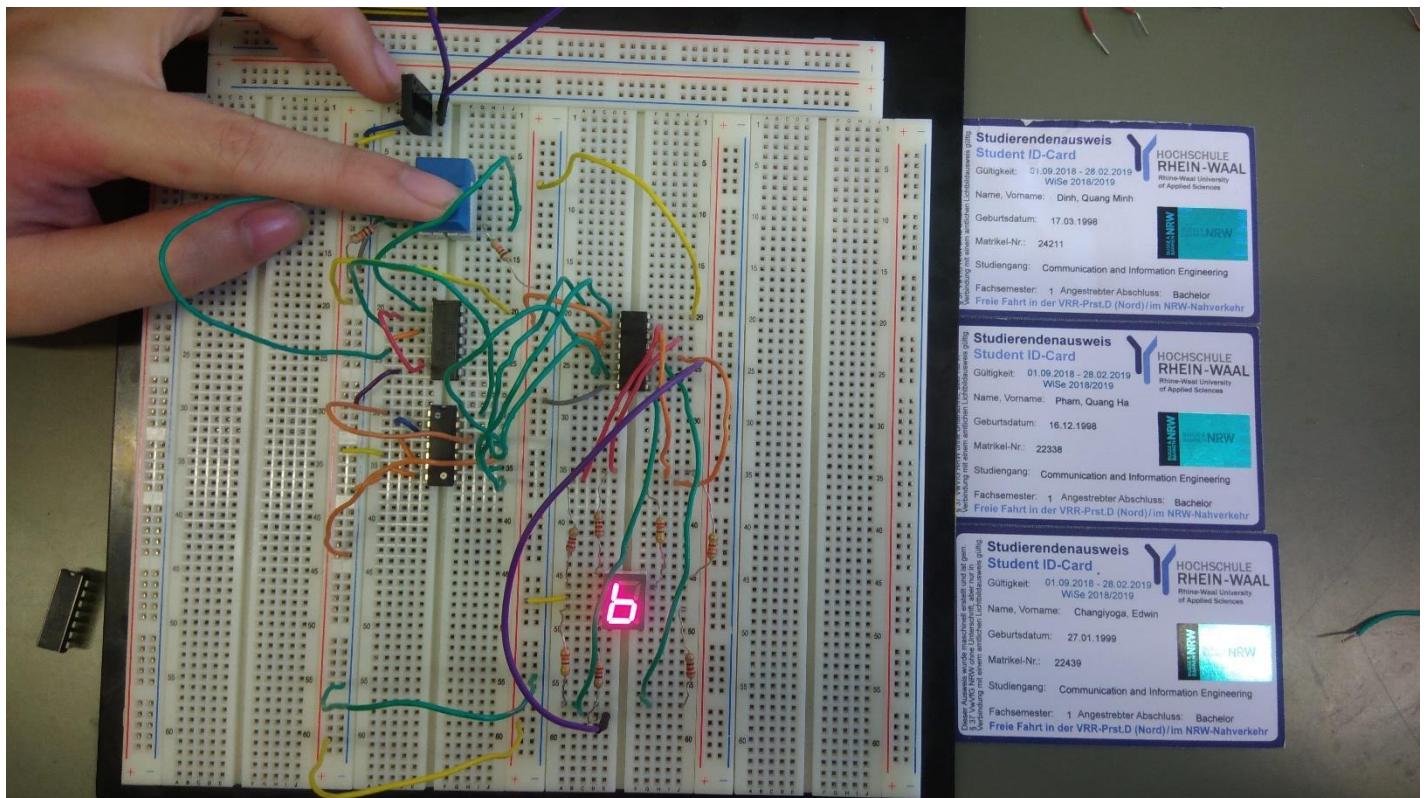
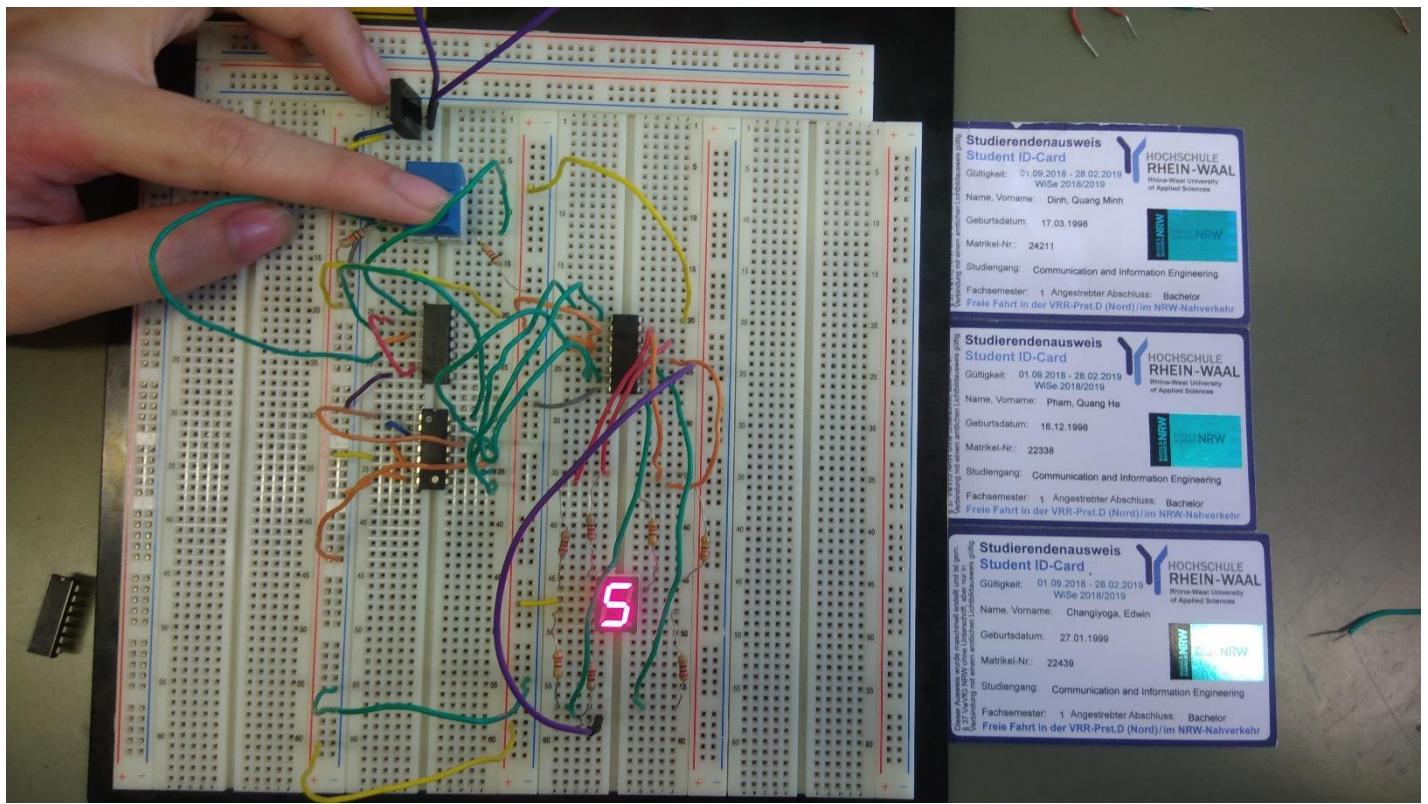
Compared to challenge #2, now for each button press the counter will only go up by 1 unit.

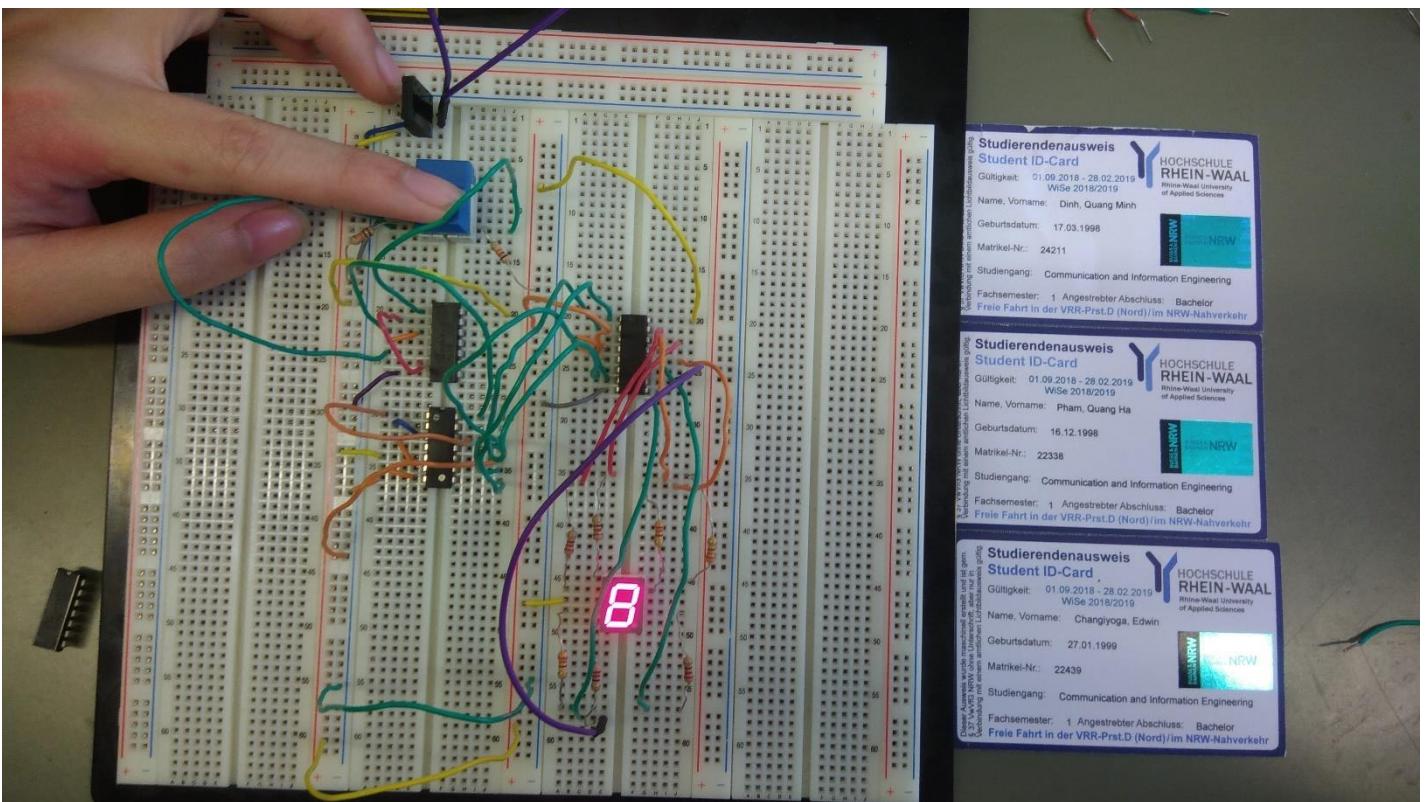
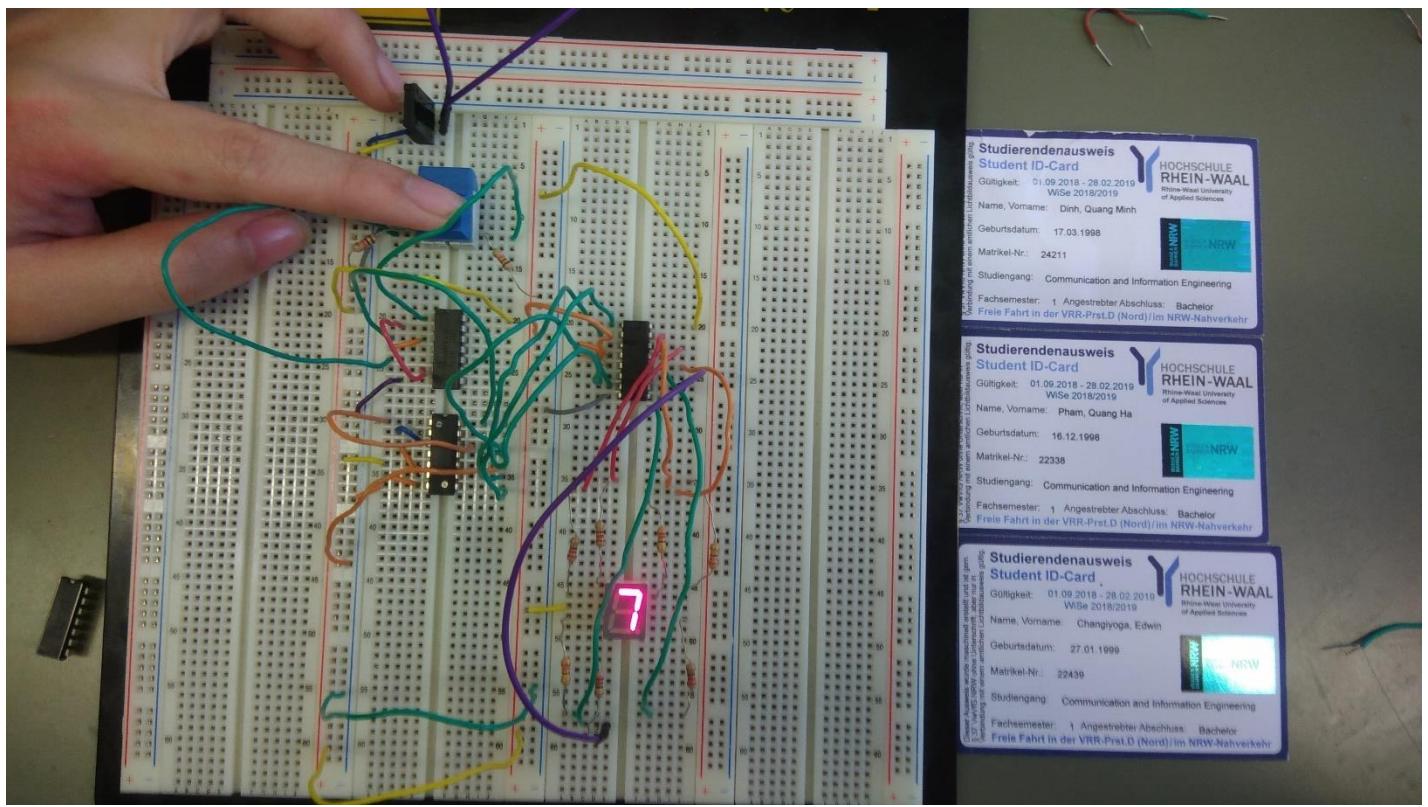
Pictures:

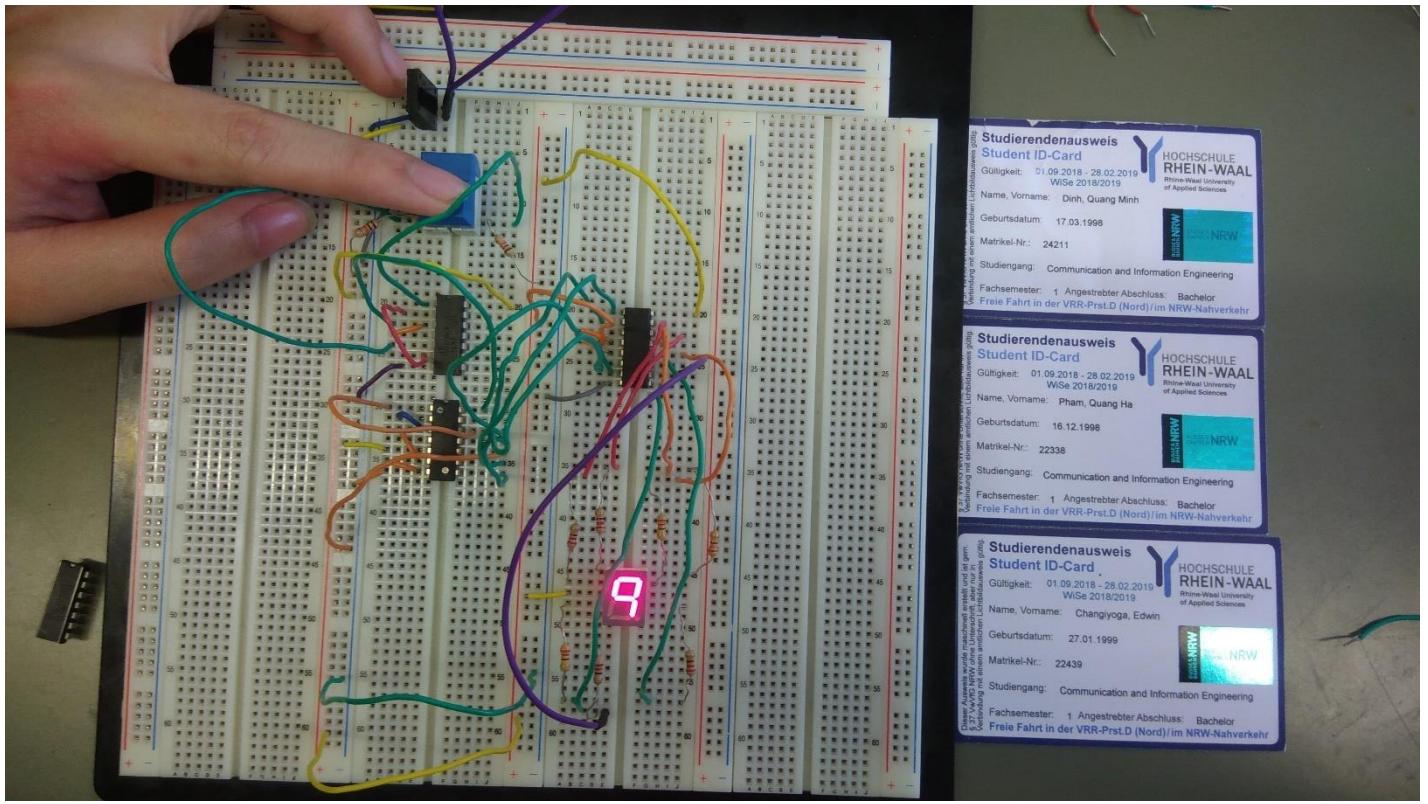










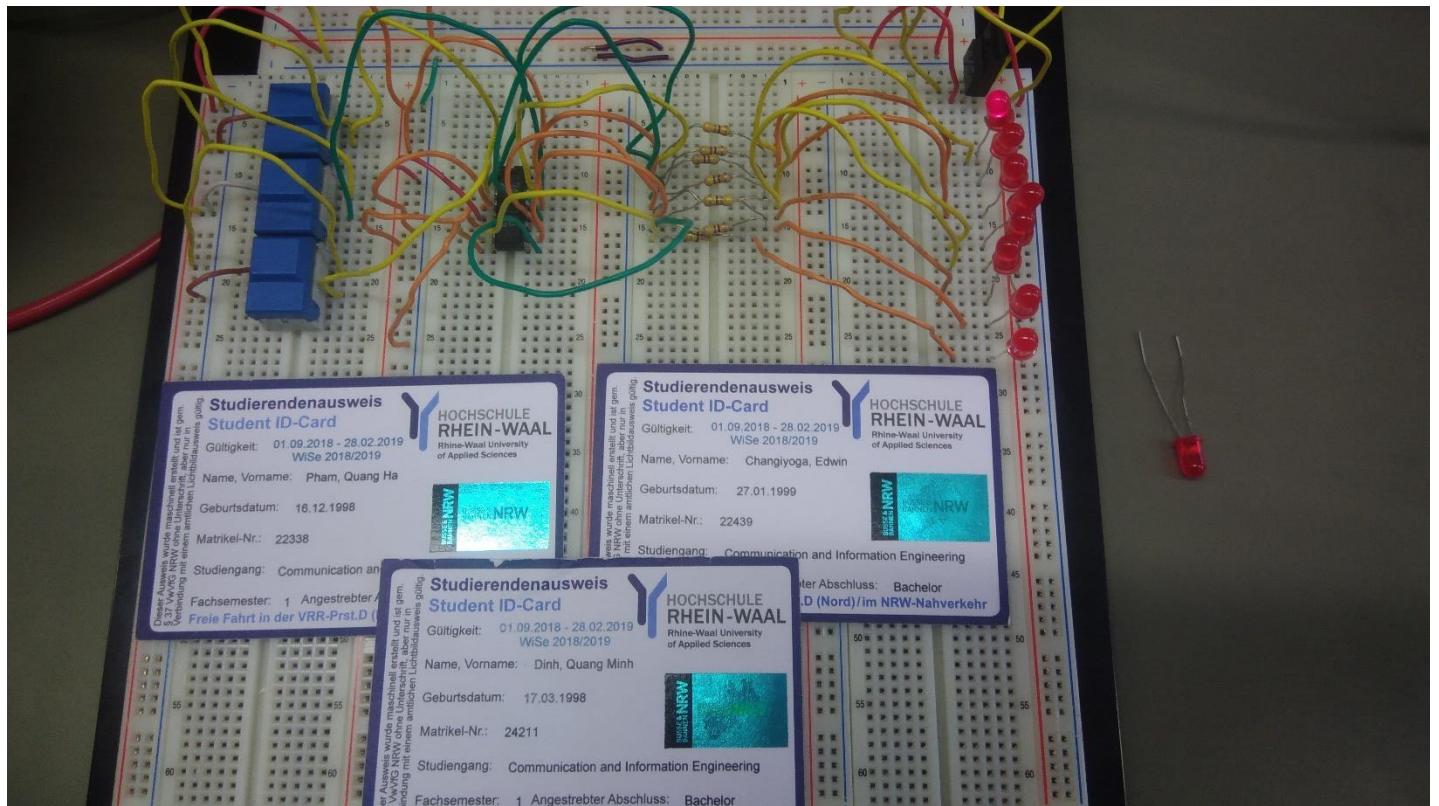


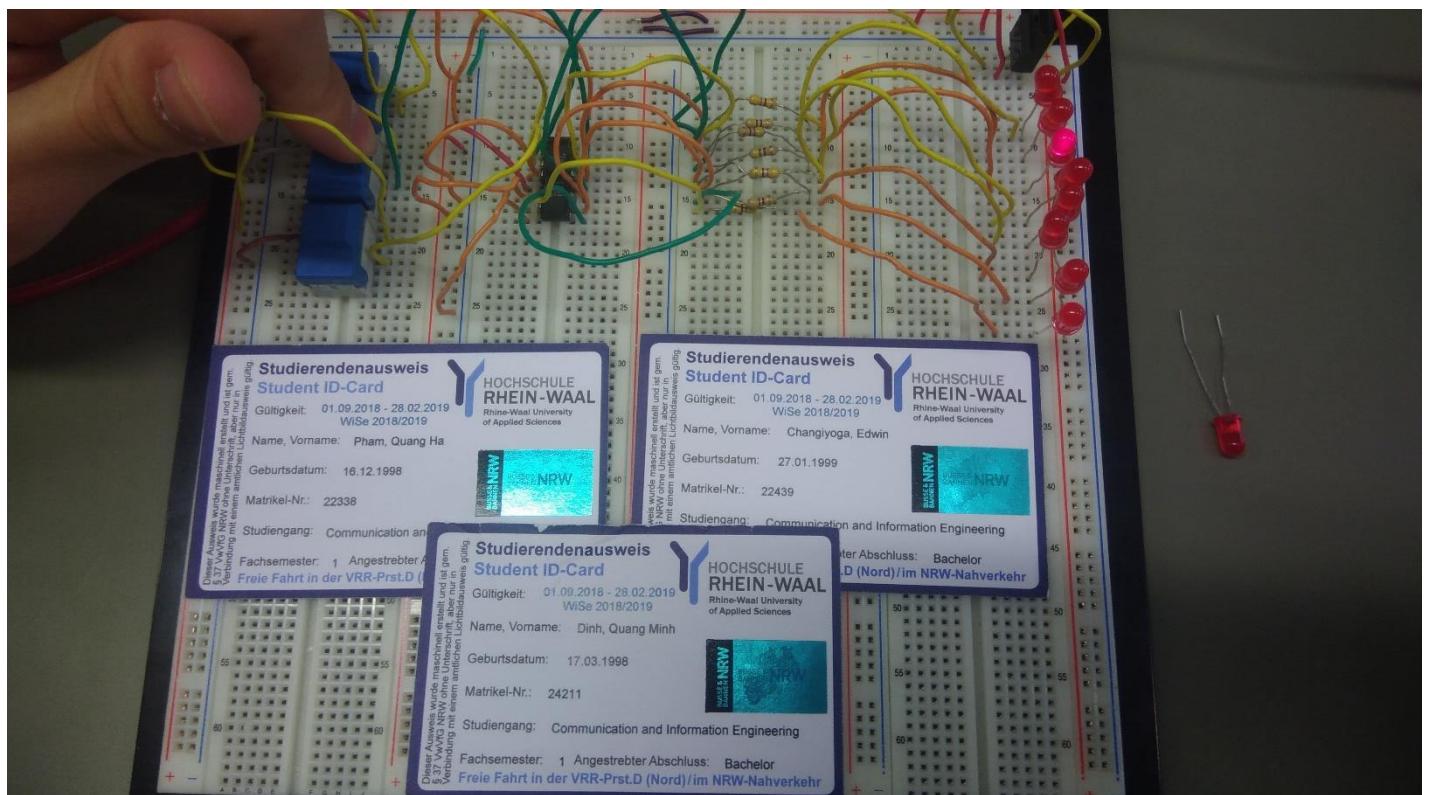
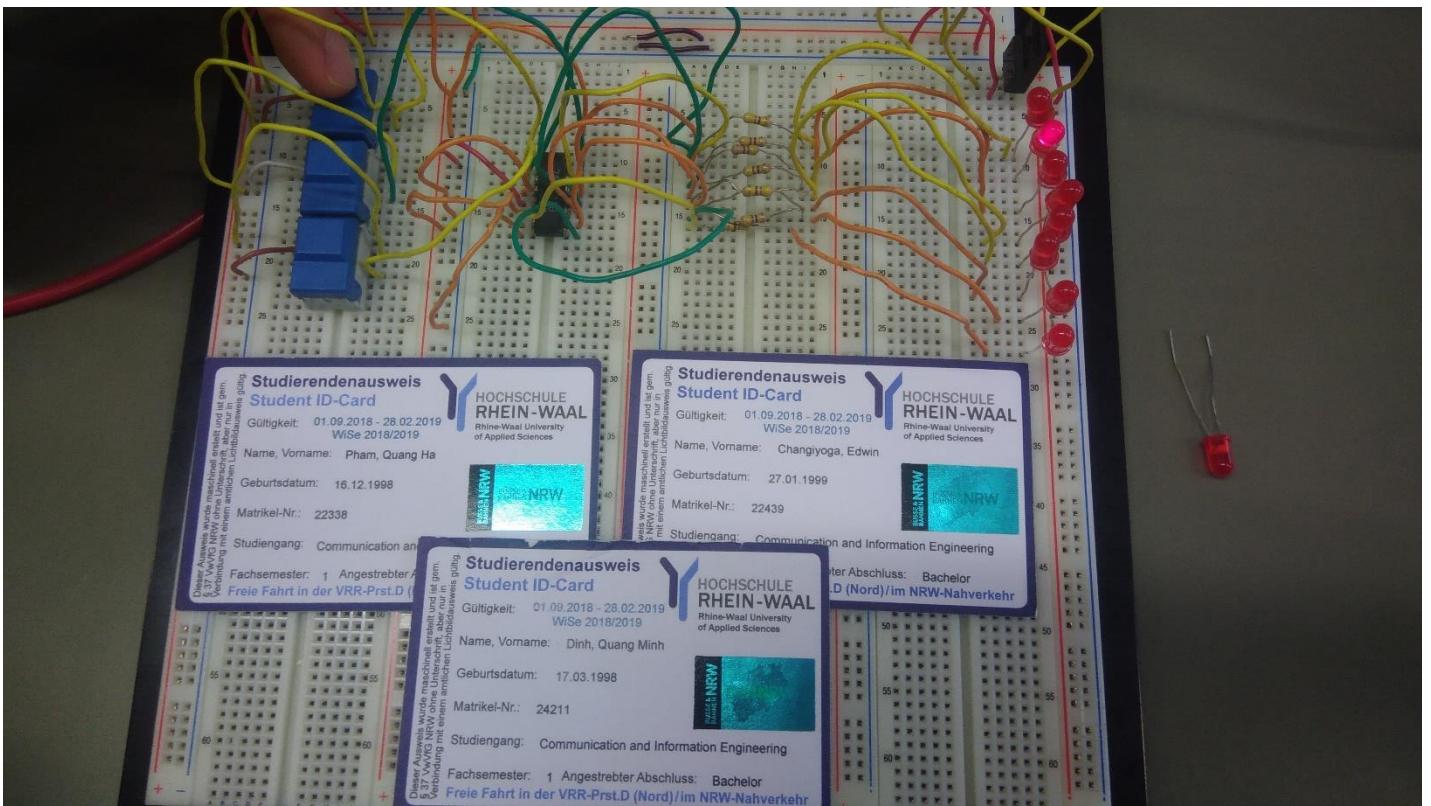
Challenge #4

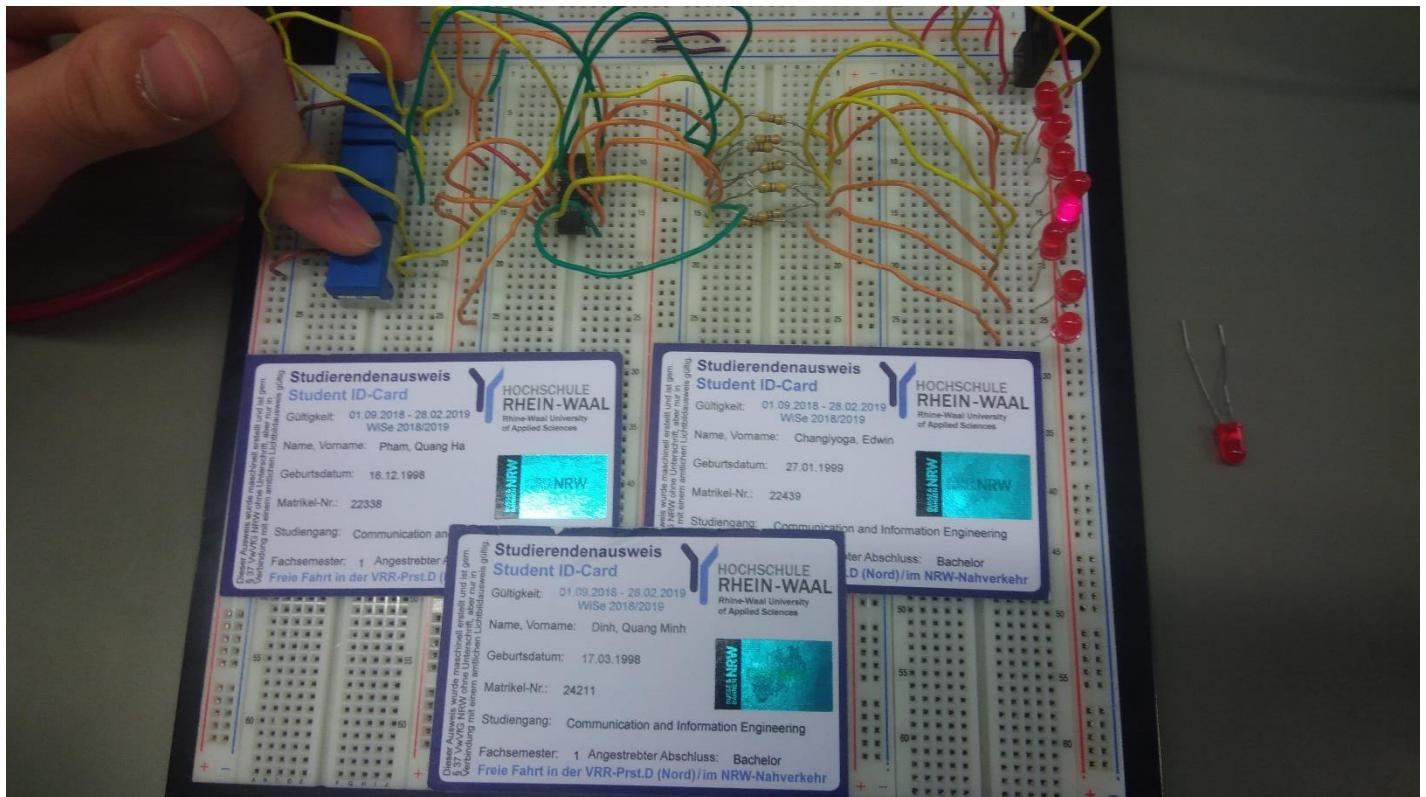
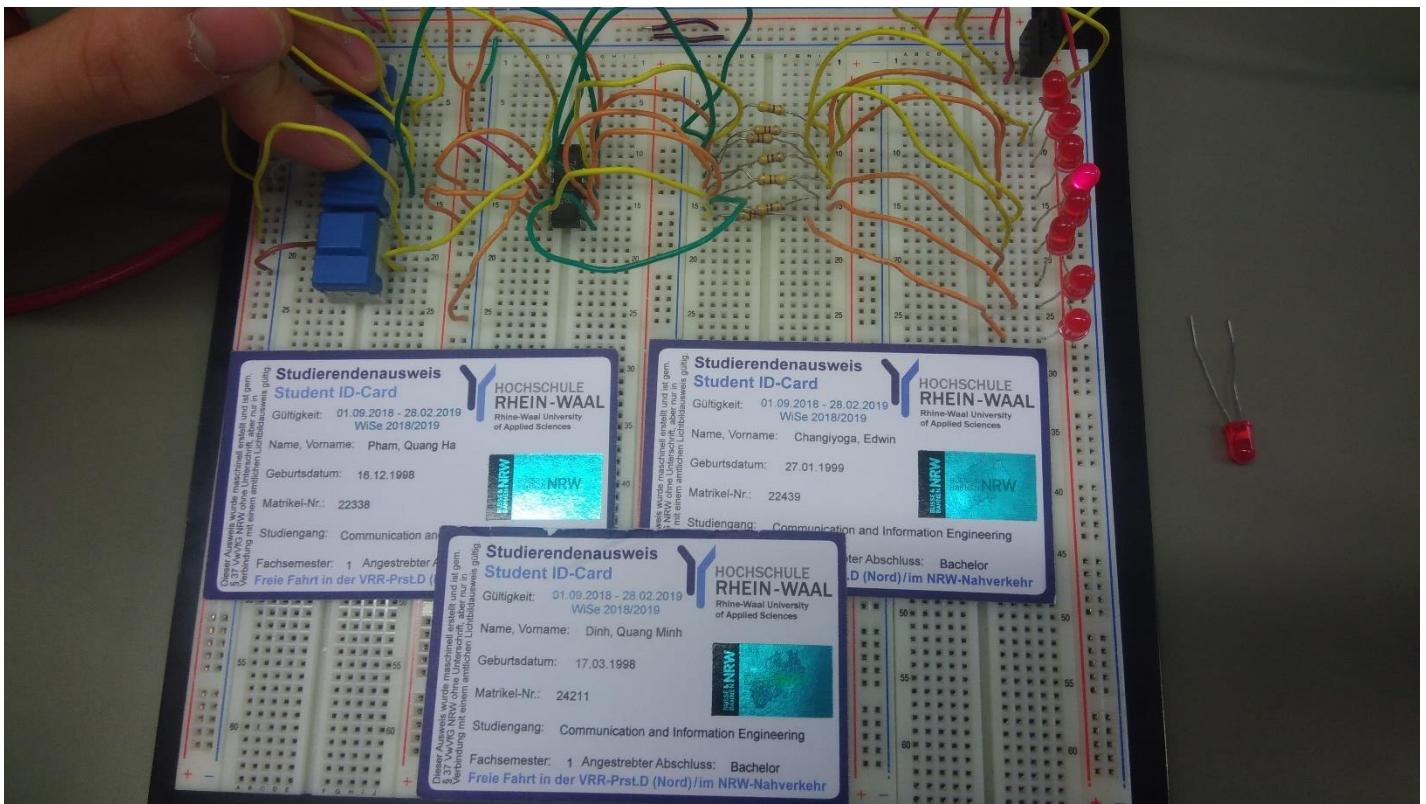
Abstract:

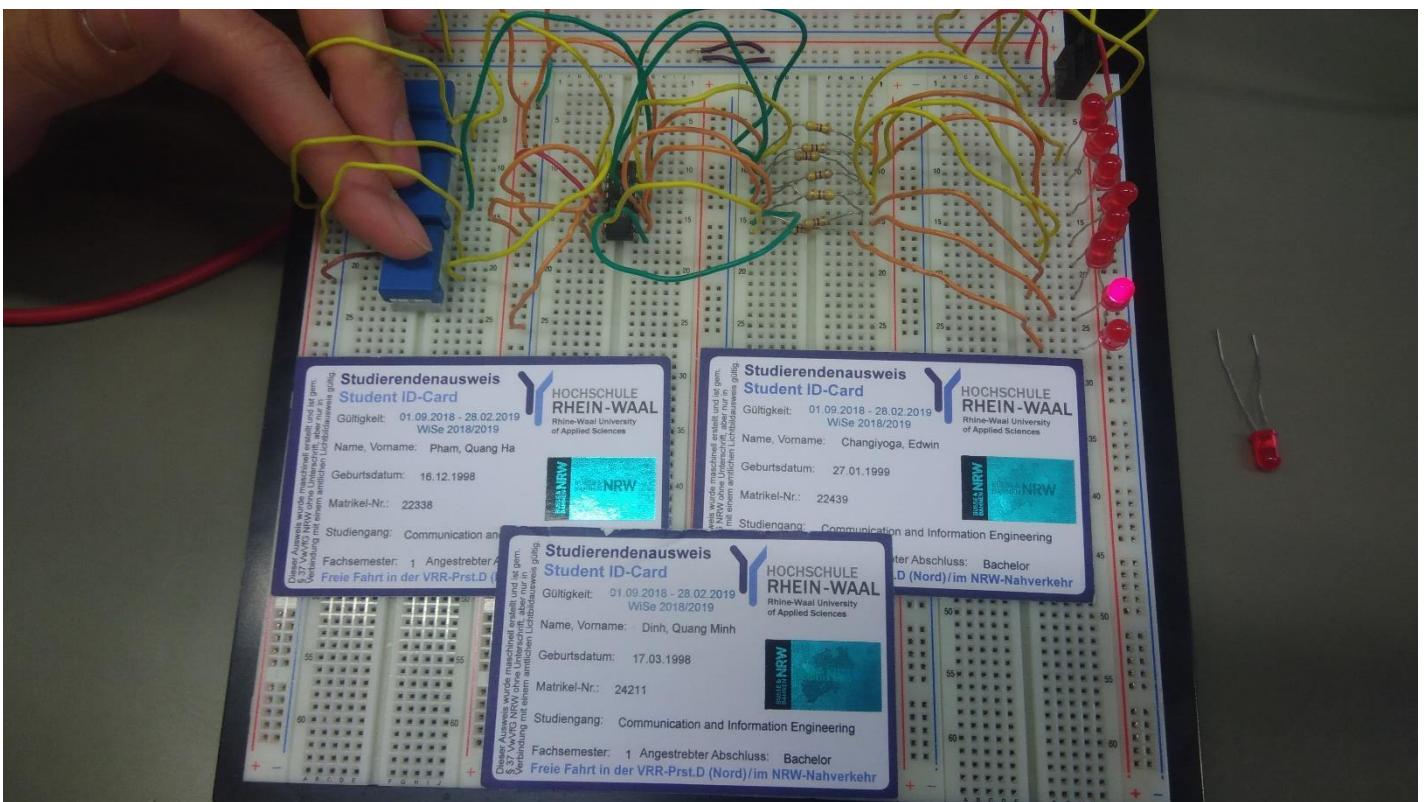
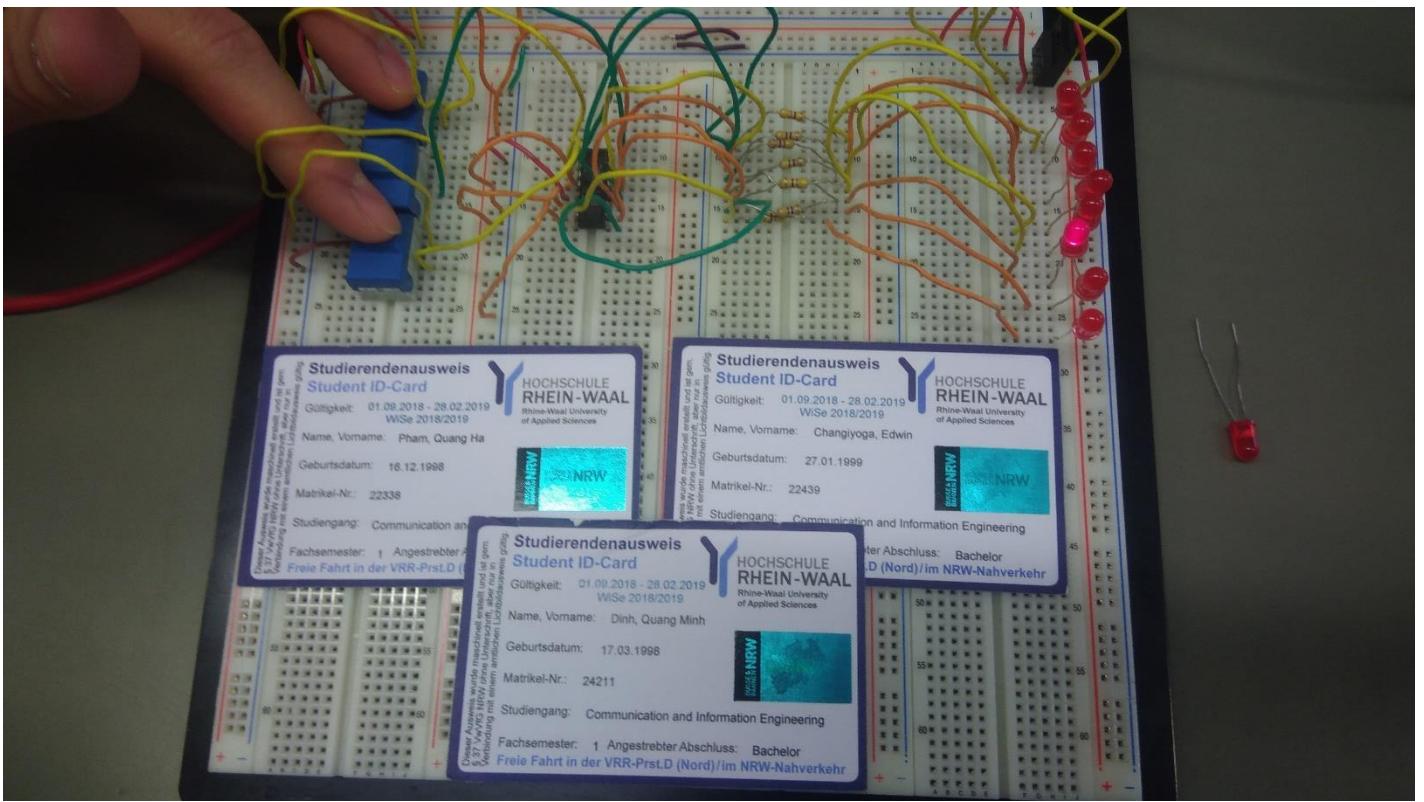
Our group managed to follow the steps given in the Description and completed the challenge without anything special to report.

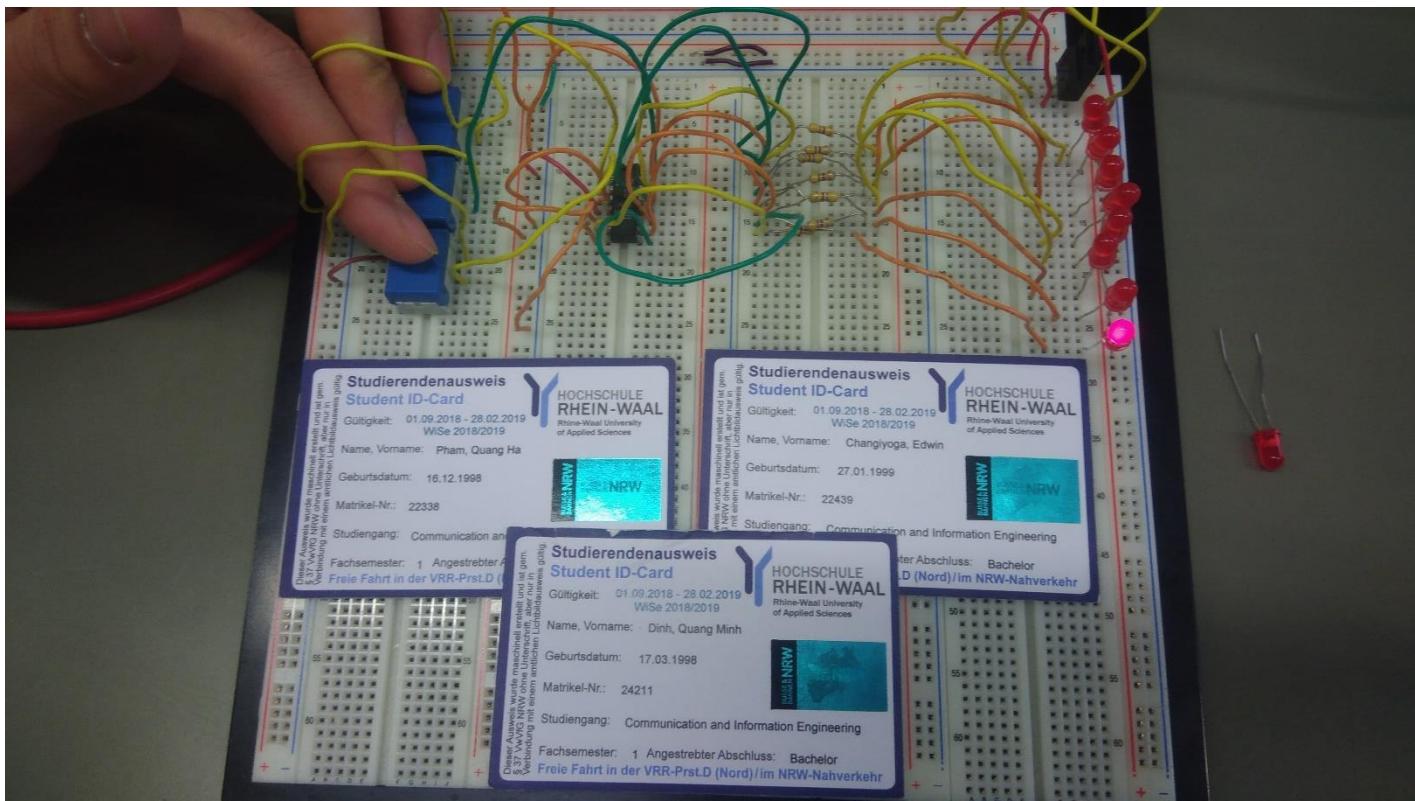
Pictures:









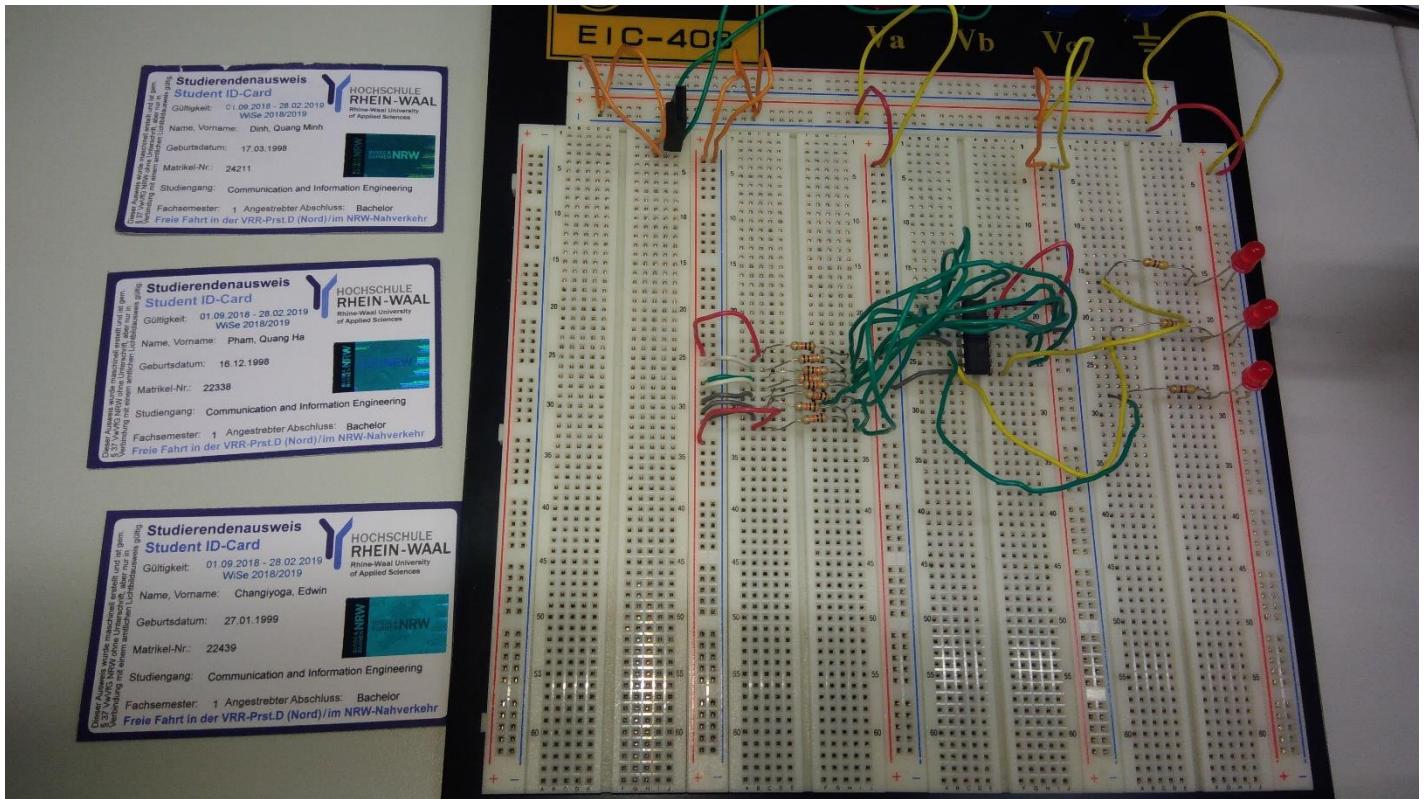


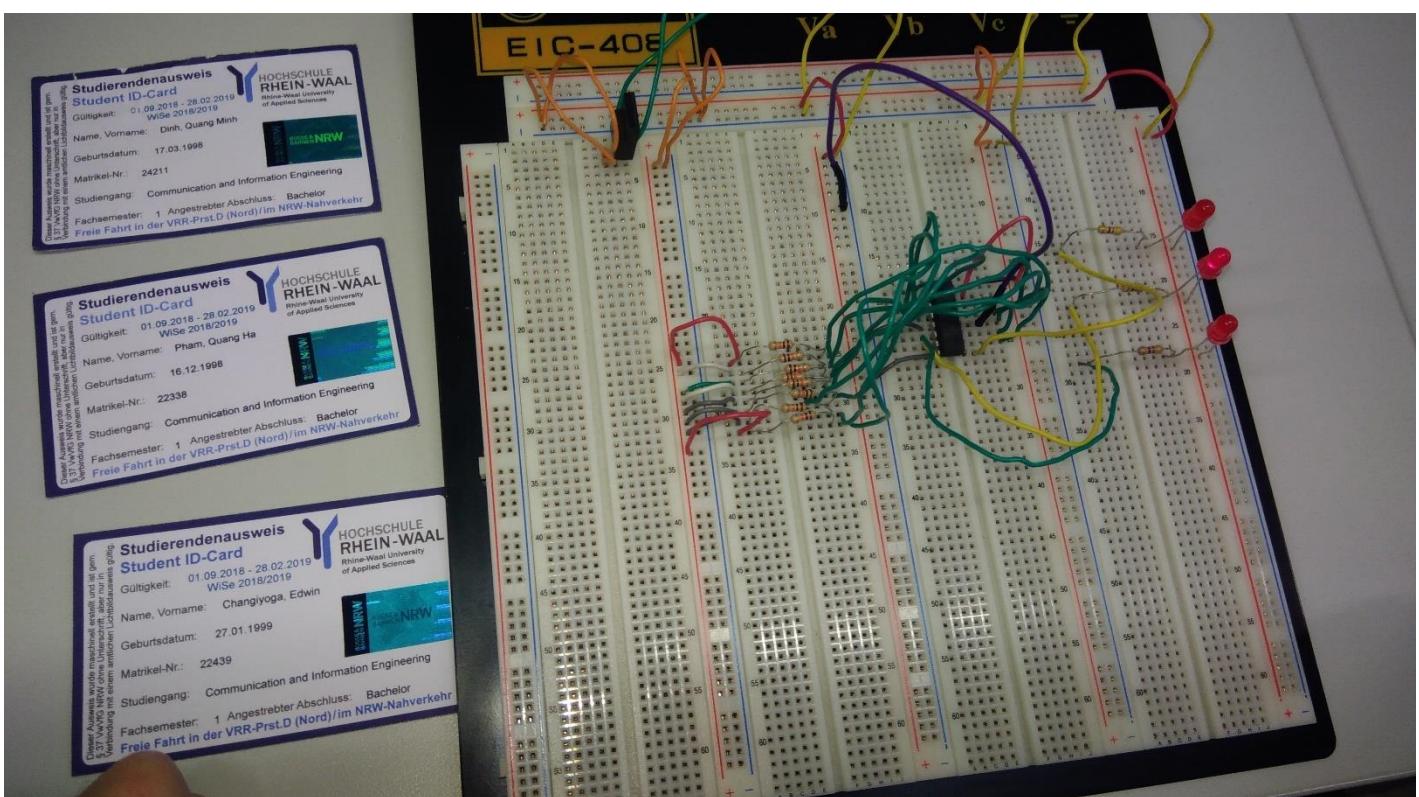
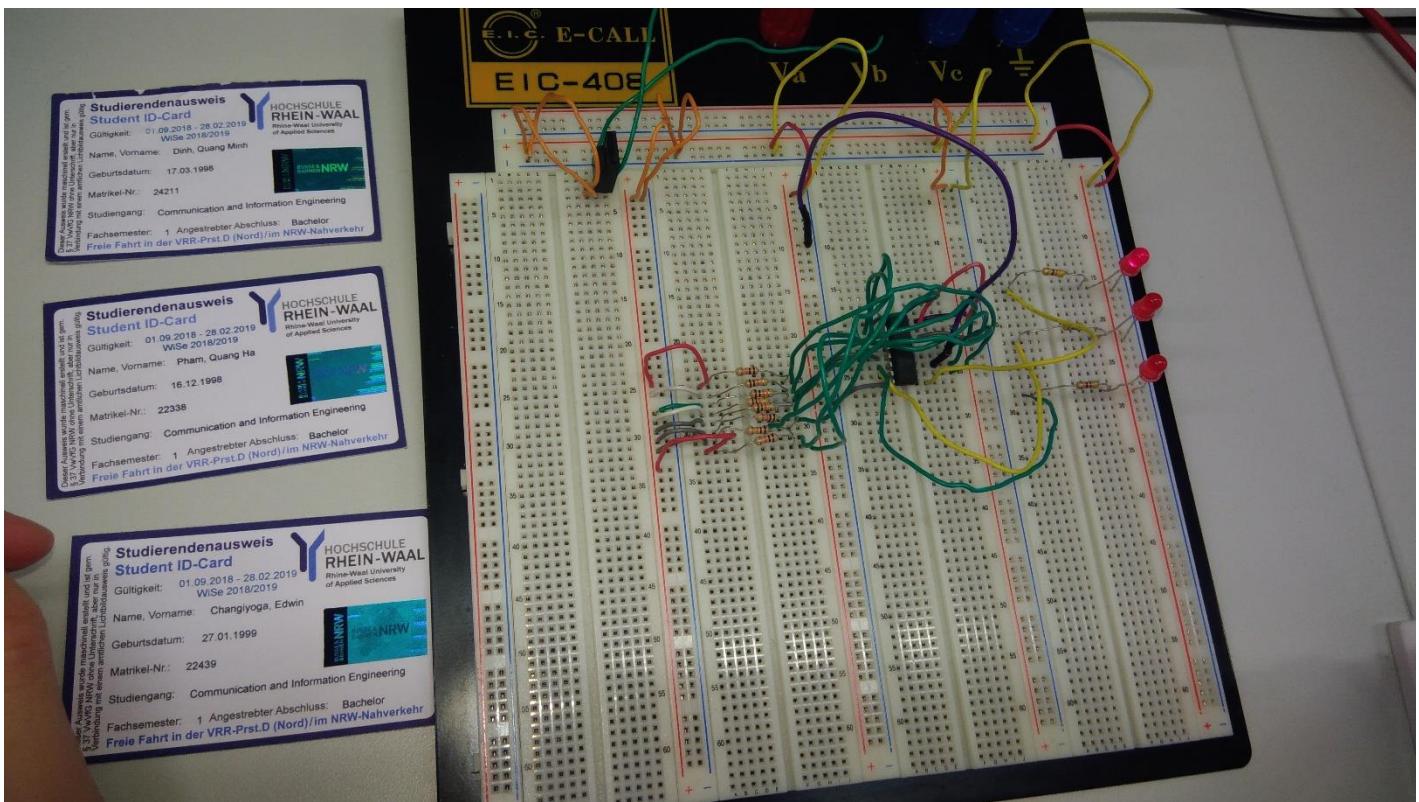
Challenge #5

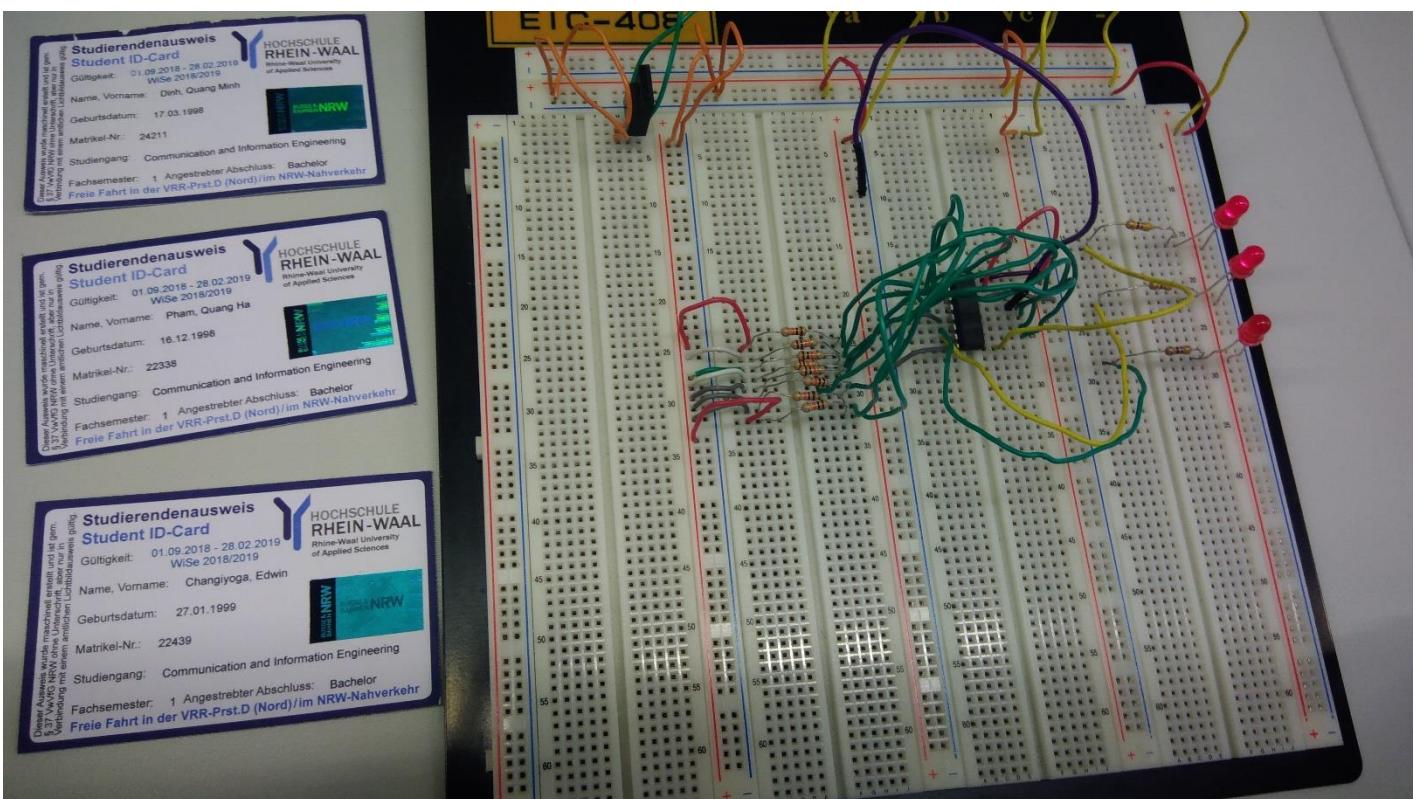
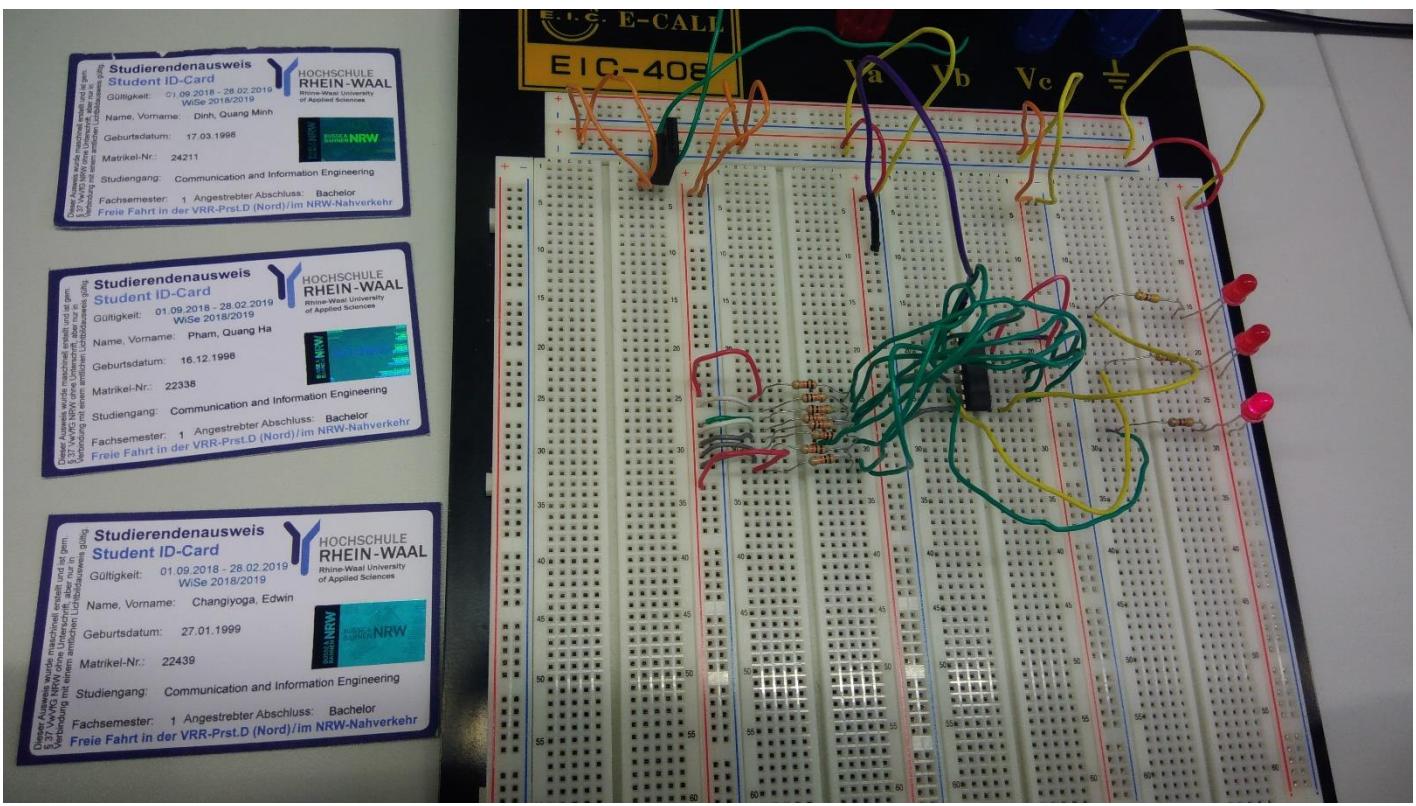
Abstract:

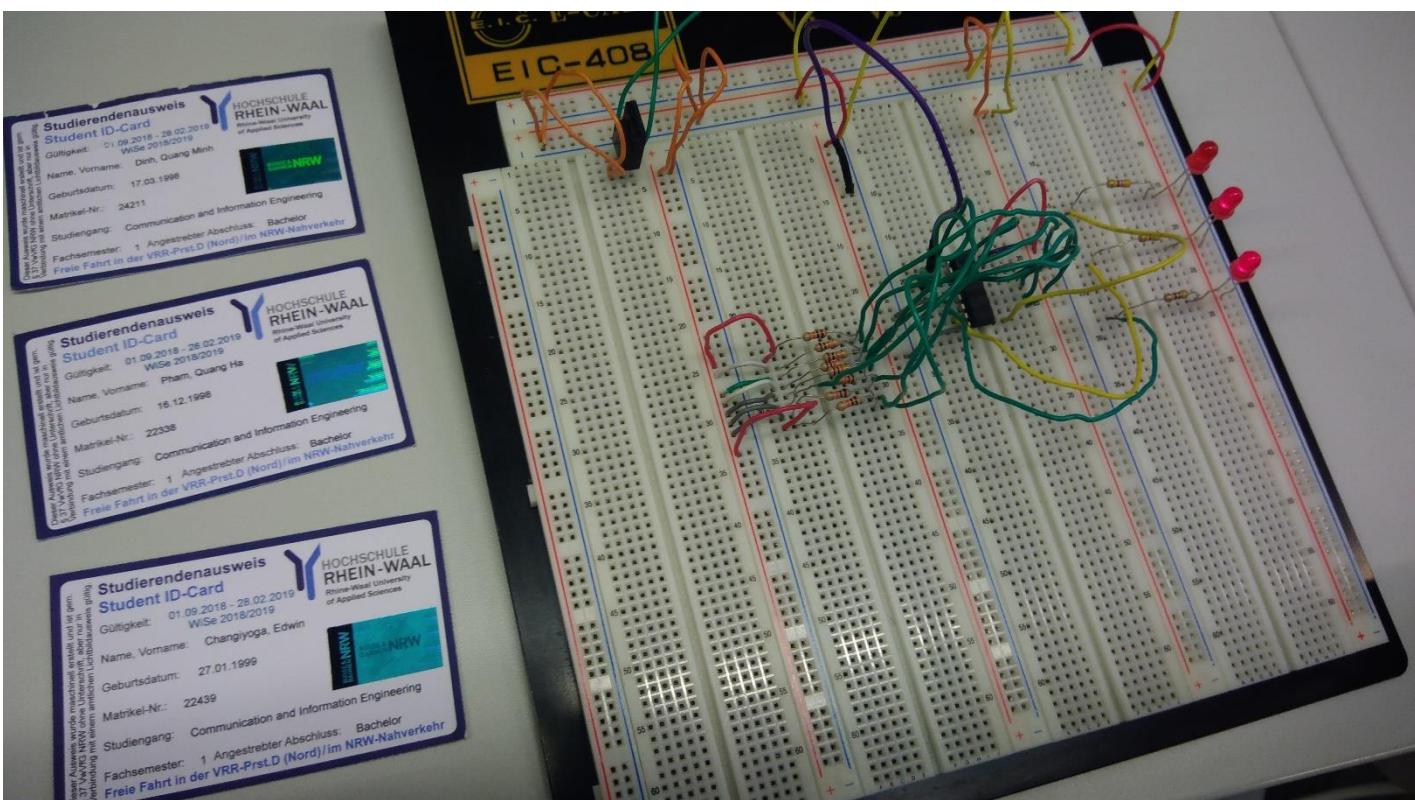
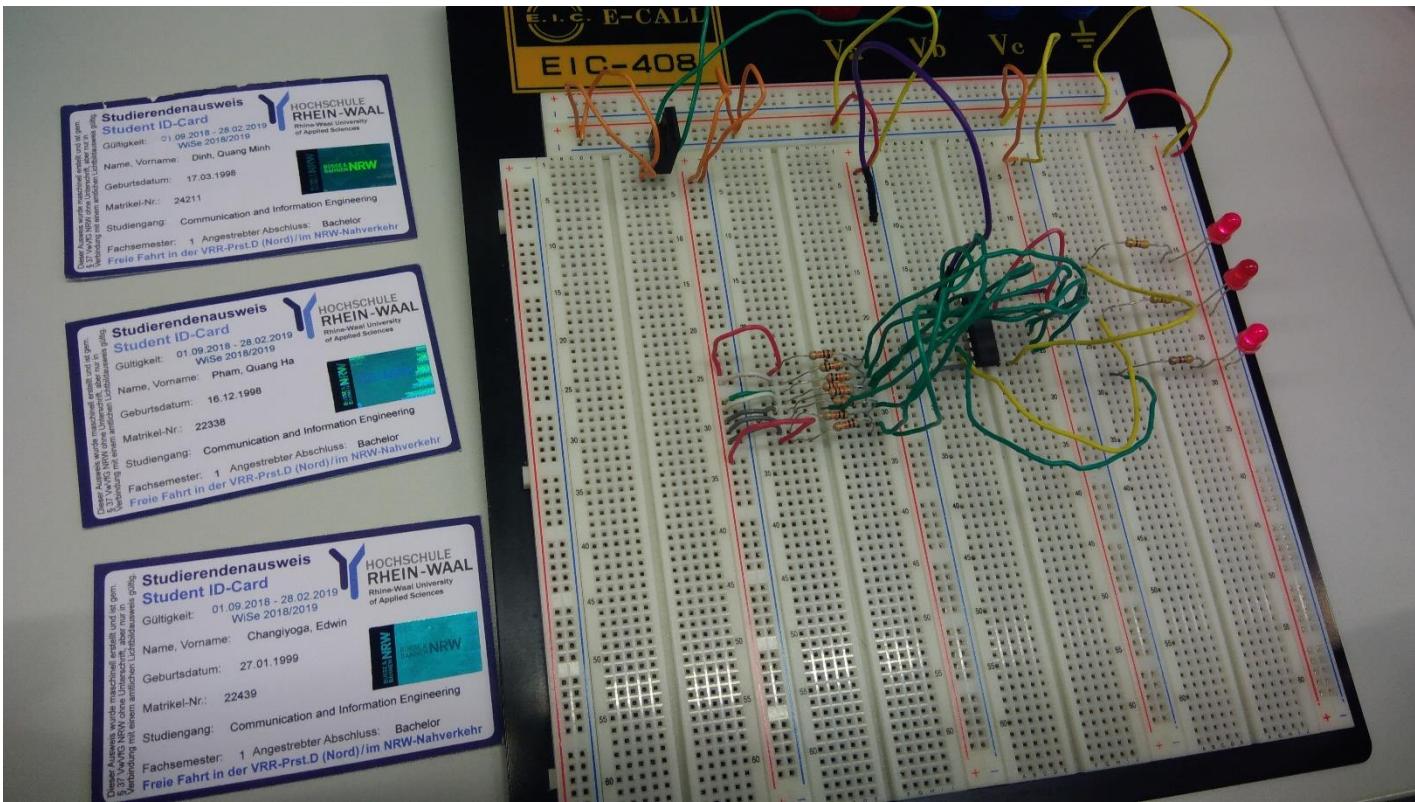
Our group managed to follow the steps given in the Description and completed the challenge without anything special to report.

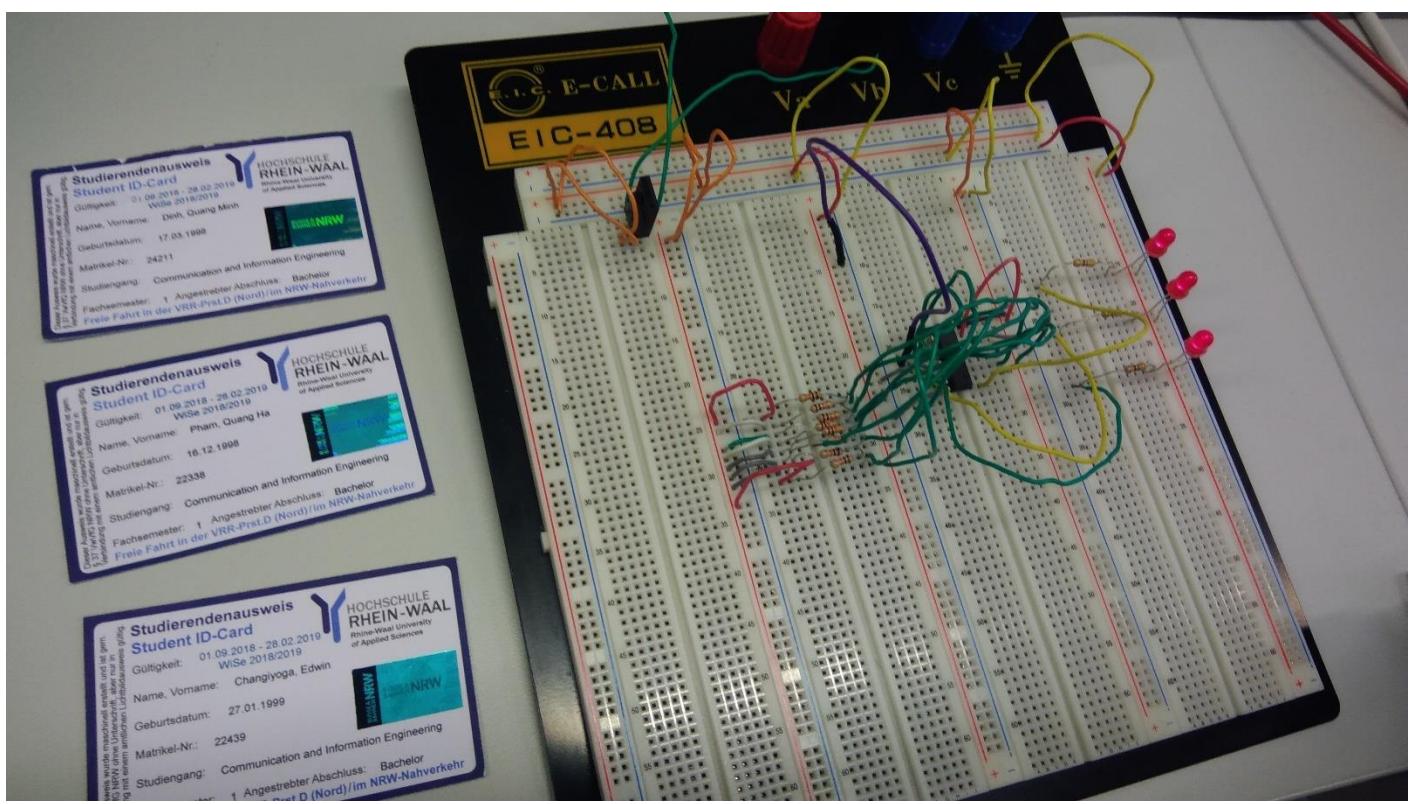
Pictures:











Challenge #6

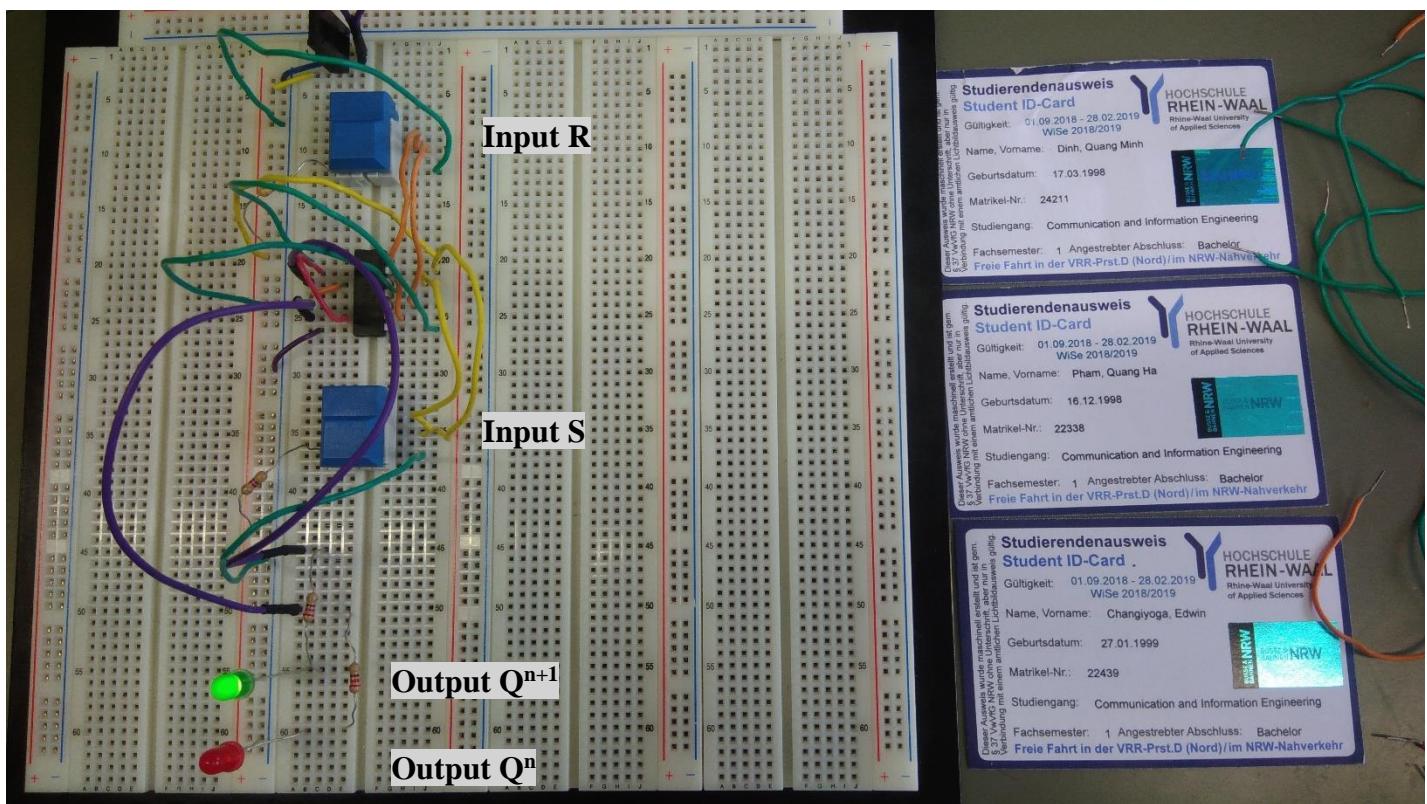
Abstract:

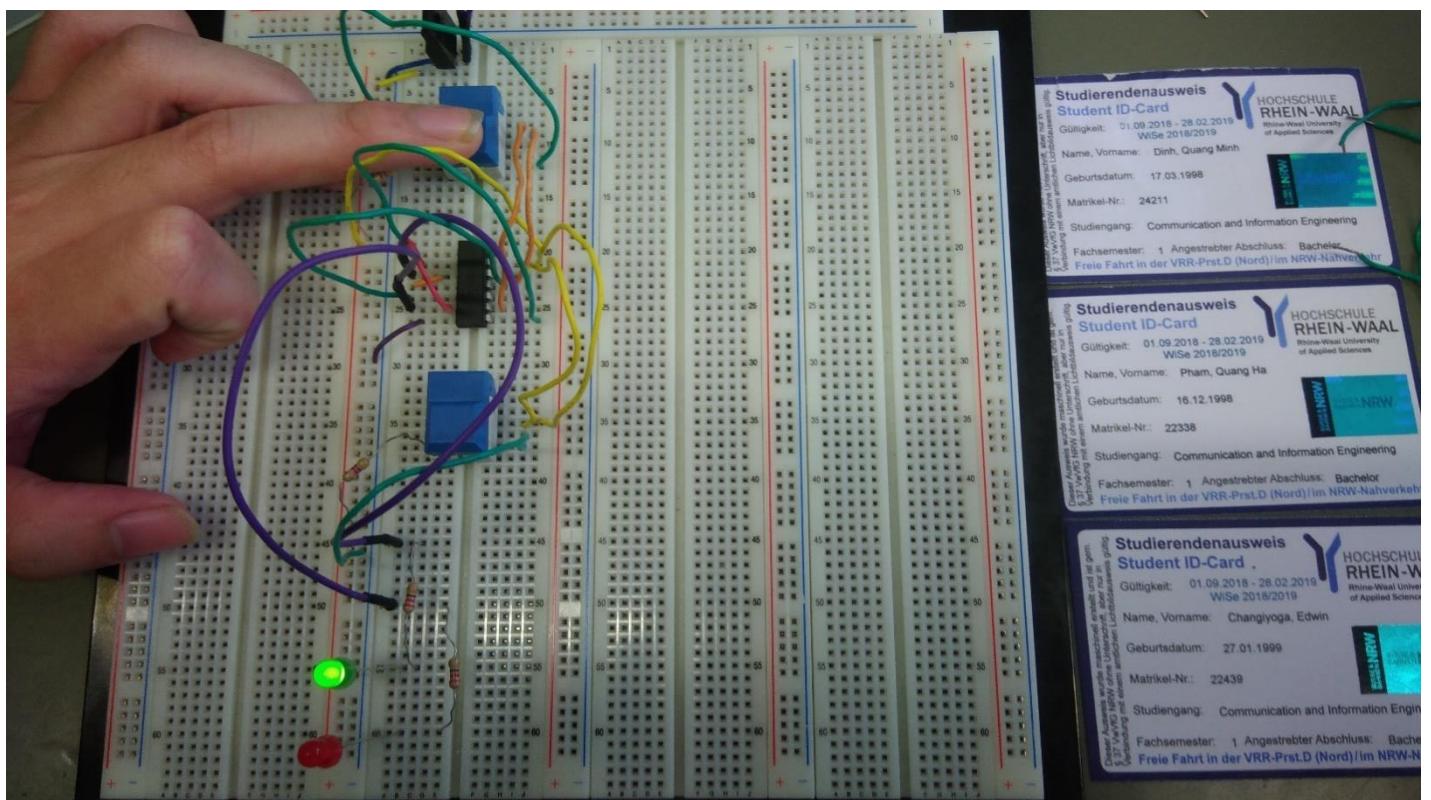
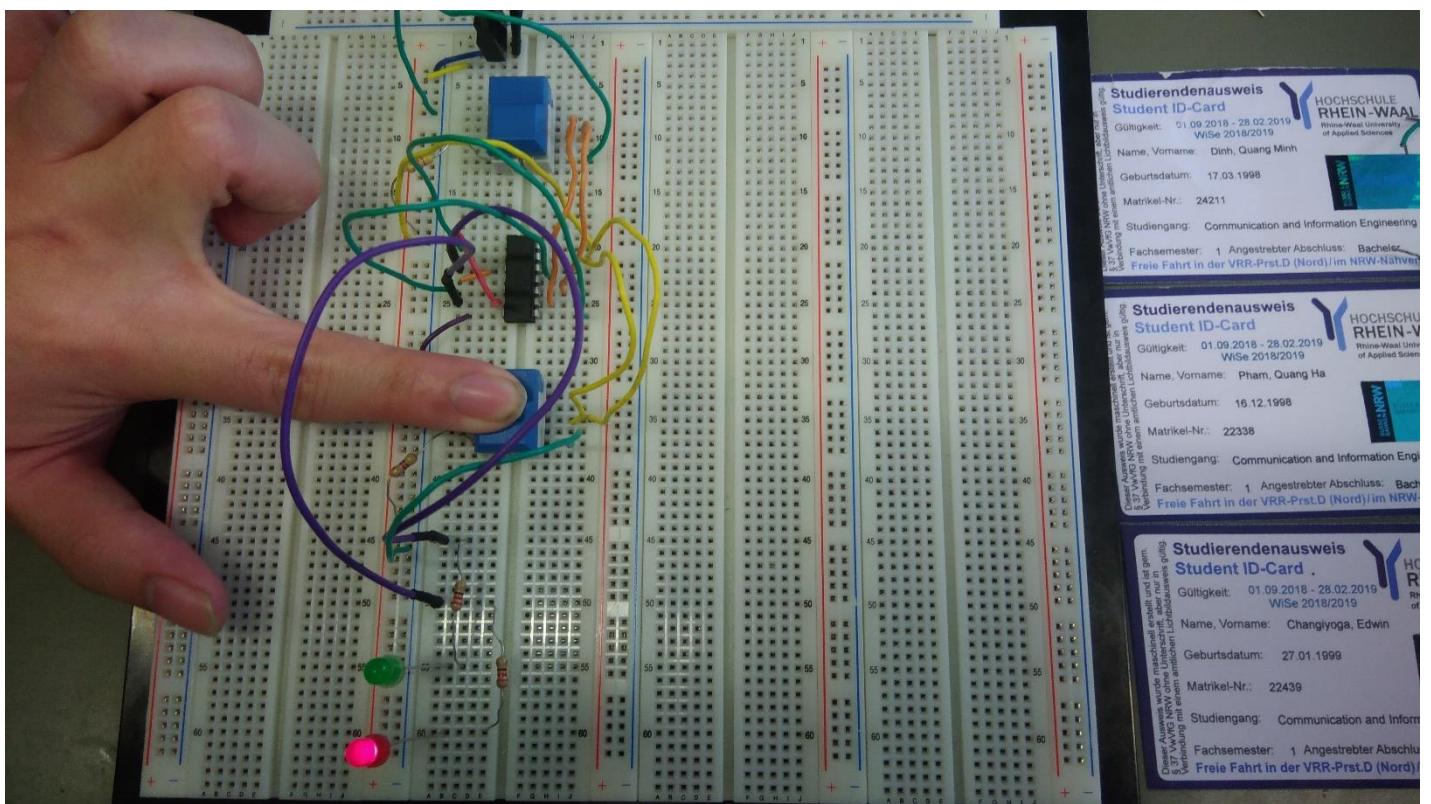
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

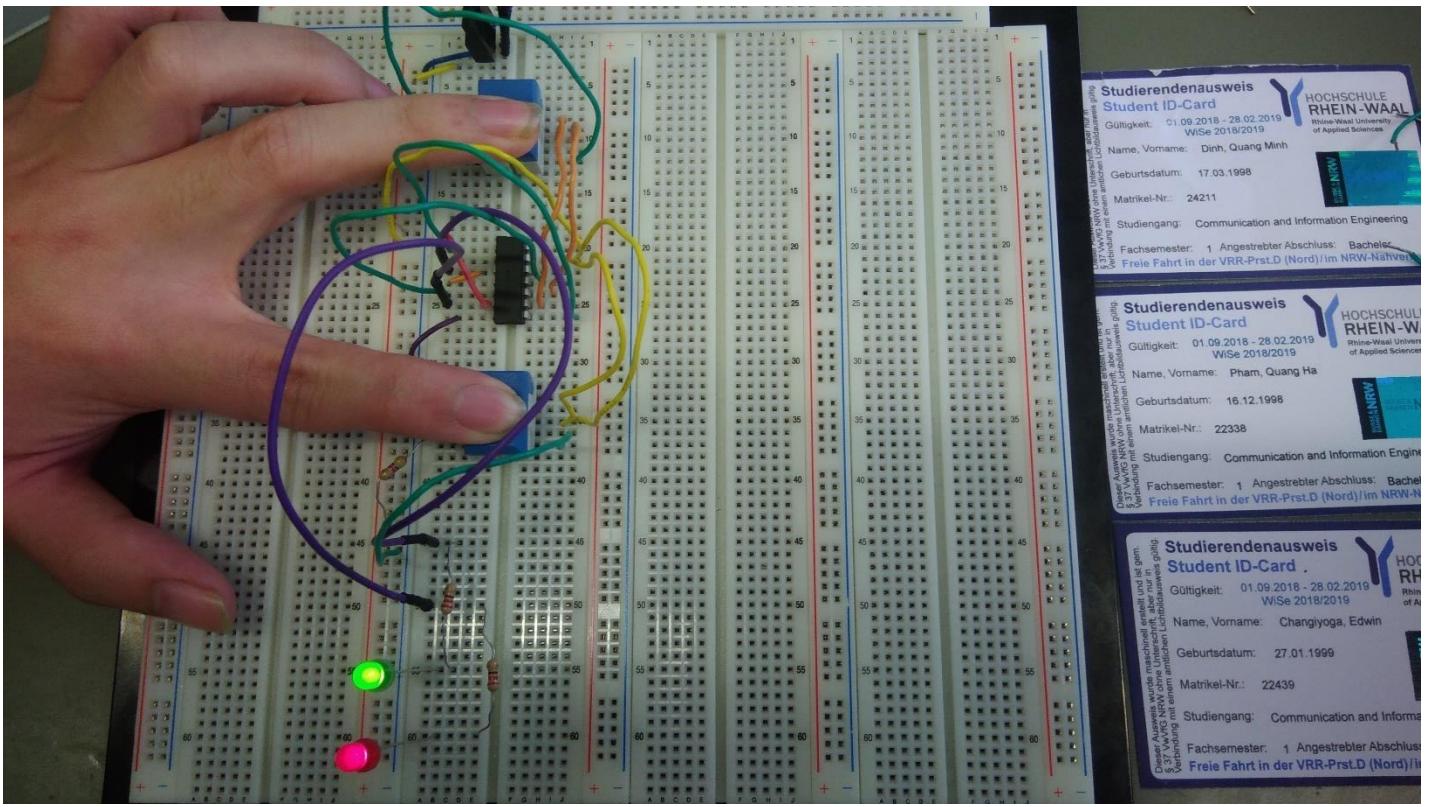
The function of this circuit is rather unique as it can “store” or “save” 1 bit of data. Unlike other challenges before where the output will turn off when we let go of the input button, for this circuit even if we press and let go of the input button the corresponding output will continue to activate. Take this challenge for example: it is able to retain the Q^n output even when $S=0$ and also able to retain the Q^{n+1} output even when $R=0$

One more thing to consider is when both inputs are on: $S=R=1$, both outputs are on. This is called Set dominant.

Pictures:







Result:

X: Does not happen.

S	R	Q^n	Q^{n+1}
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	X
1	1	1	1

Challenge #7

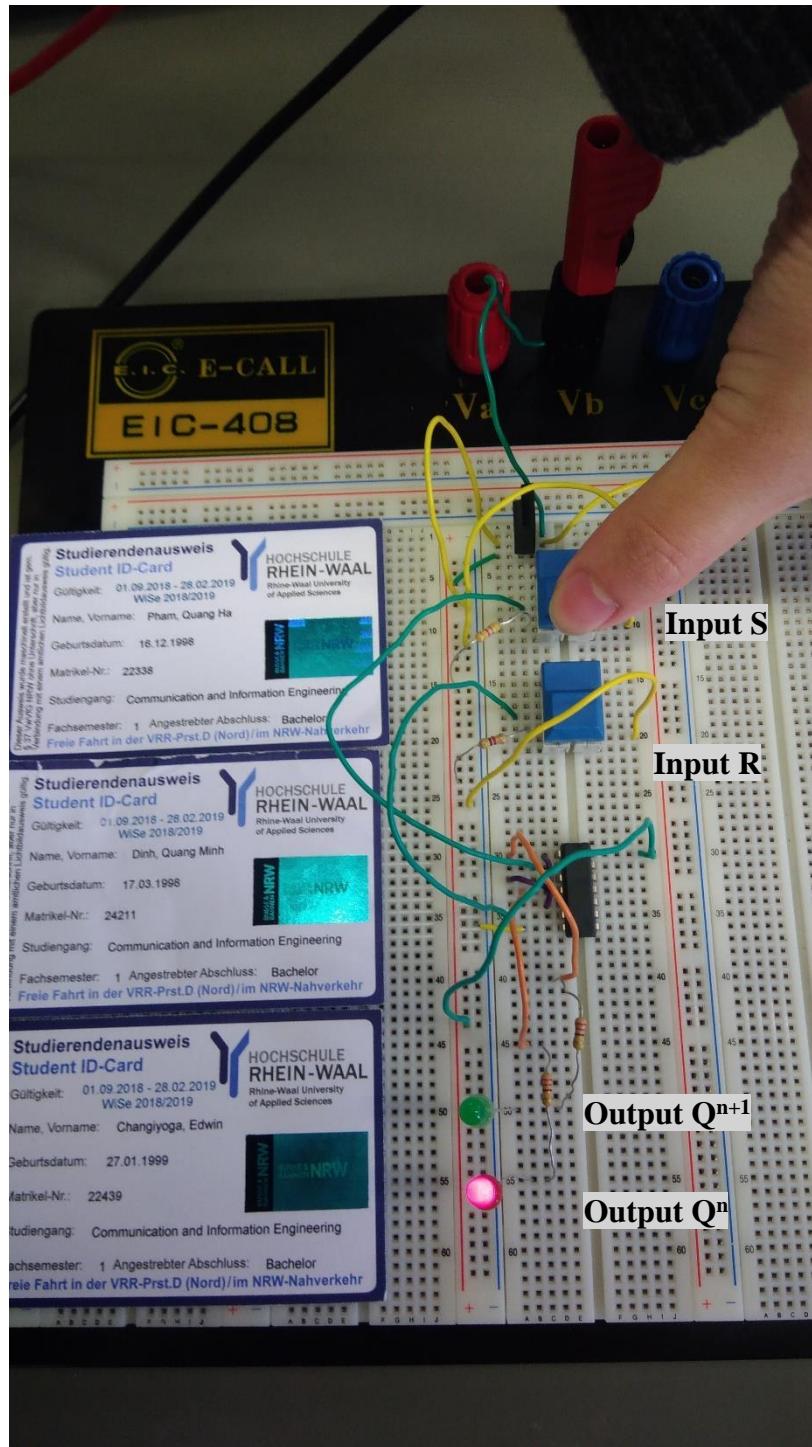
Abstract:

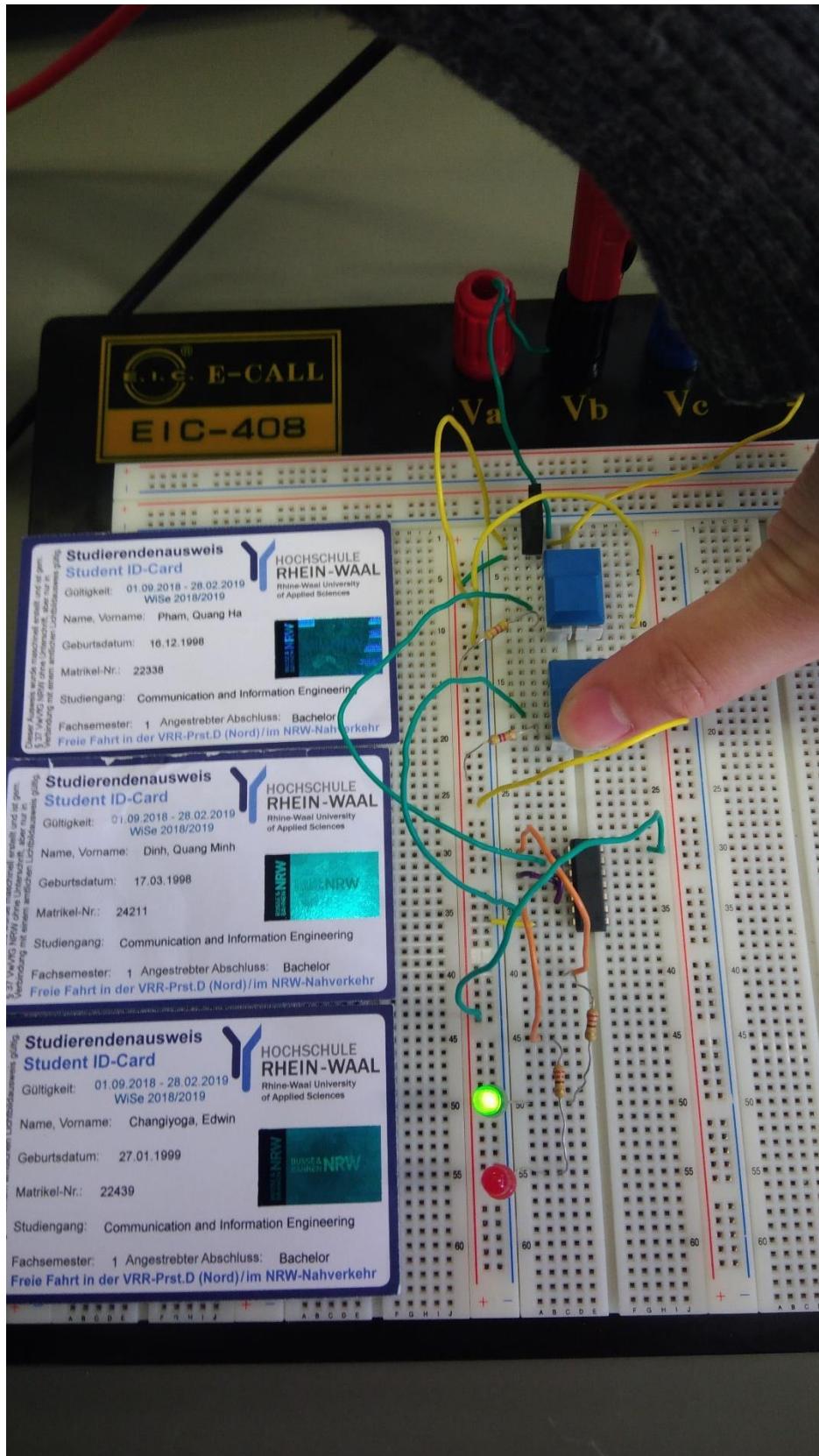
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

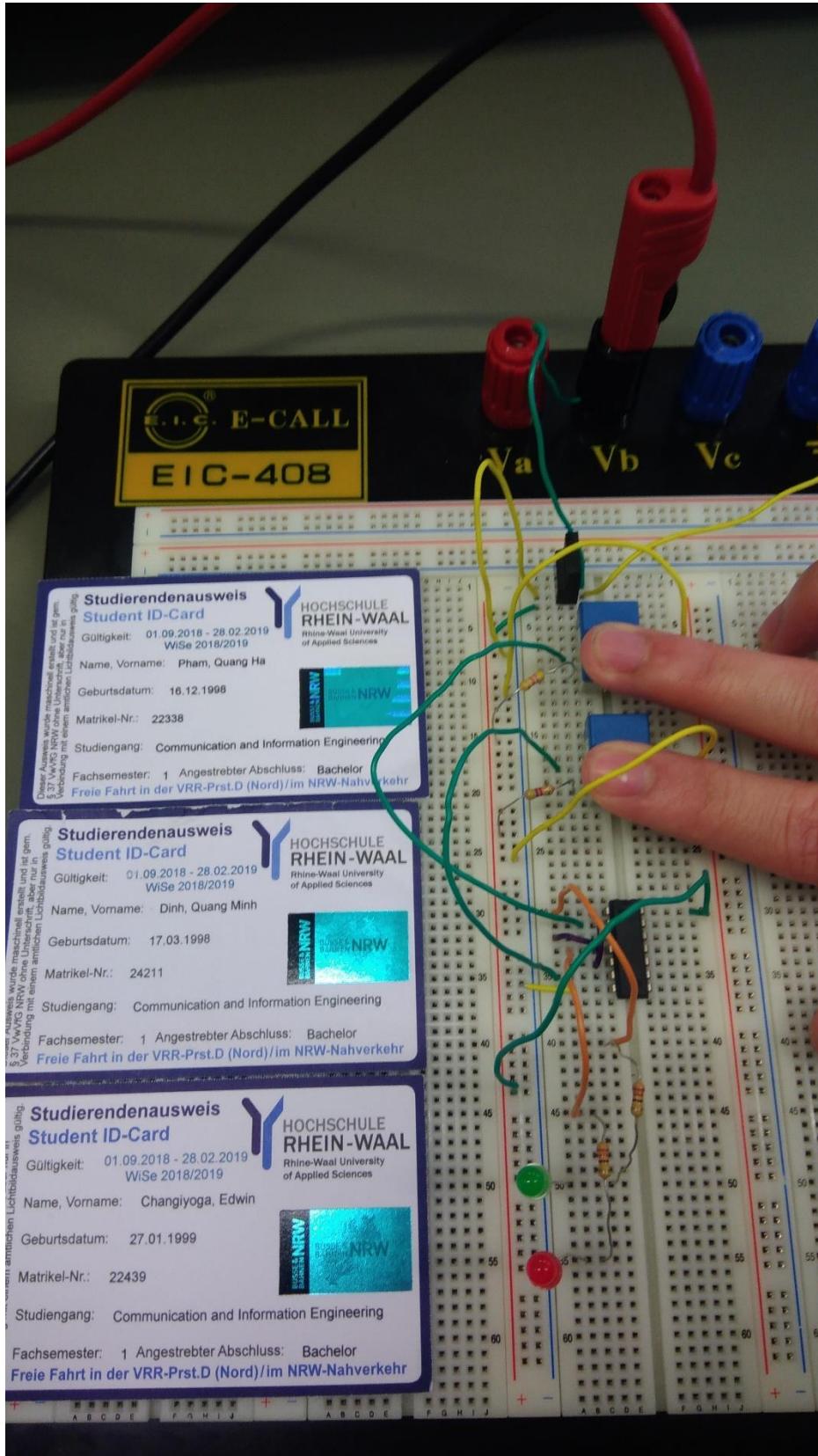
The function of this circuit is the same as described above in challenge #6.

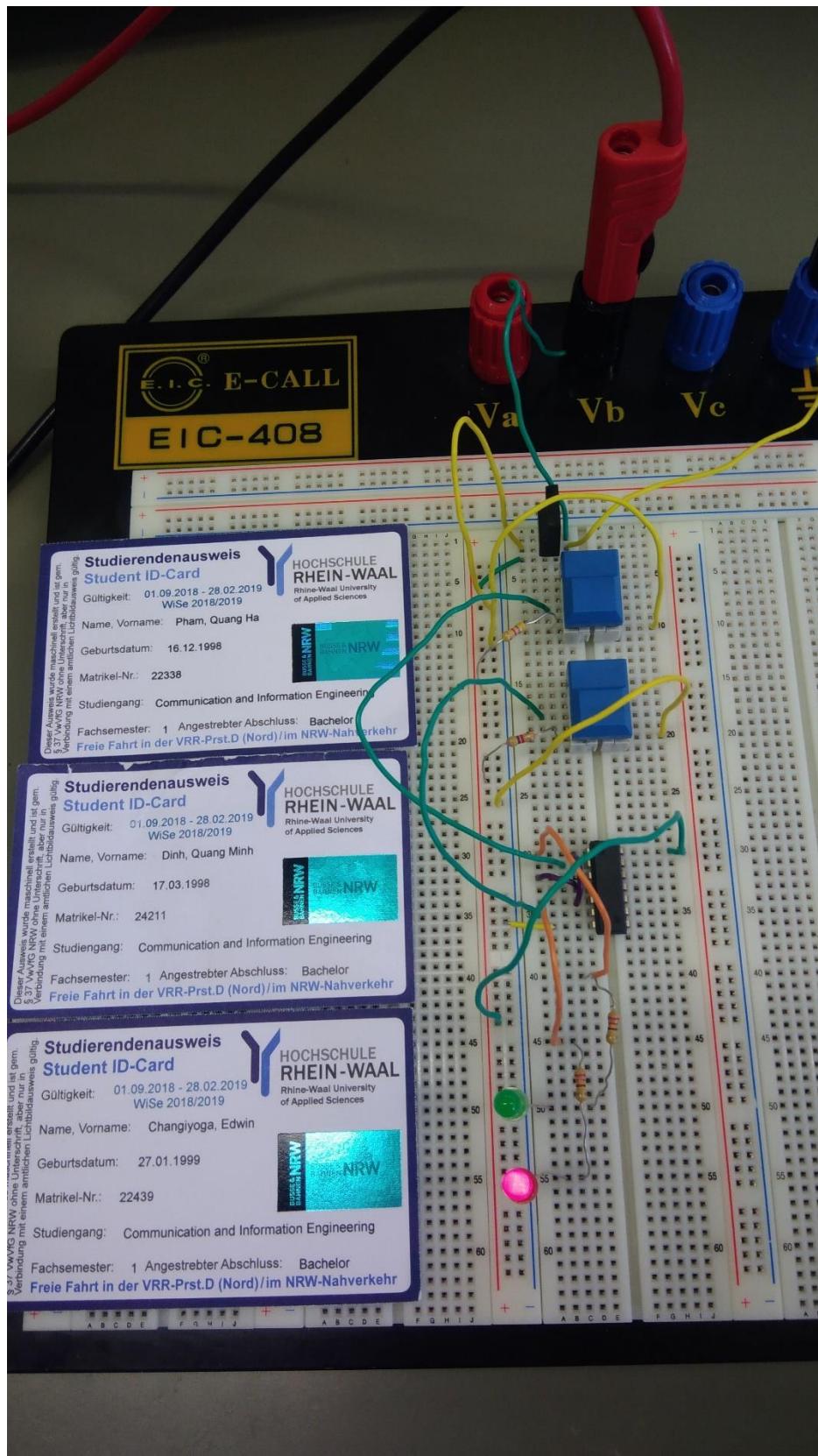
The only difference is when both inputs are on: $S=R=1$, both outputs are off. This is called Reset dominant.

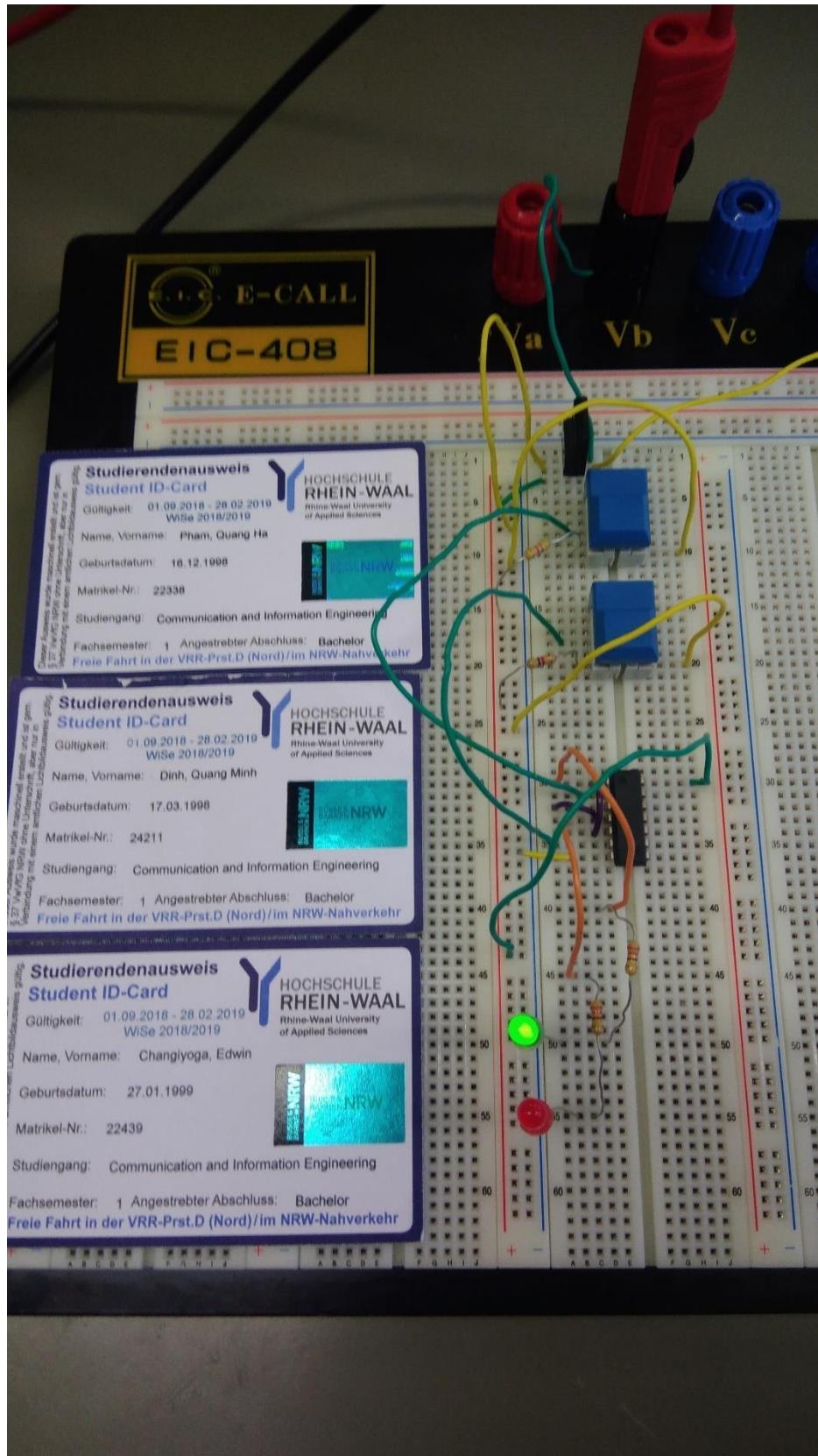
Pictures:











Result:

S	R	Qⁿ	Qⁿ⁺¹
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	X
1	0	1	0
1	1	0	0
1	1	1	X

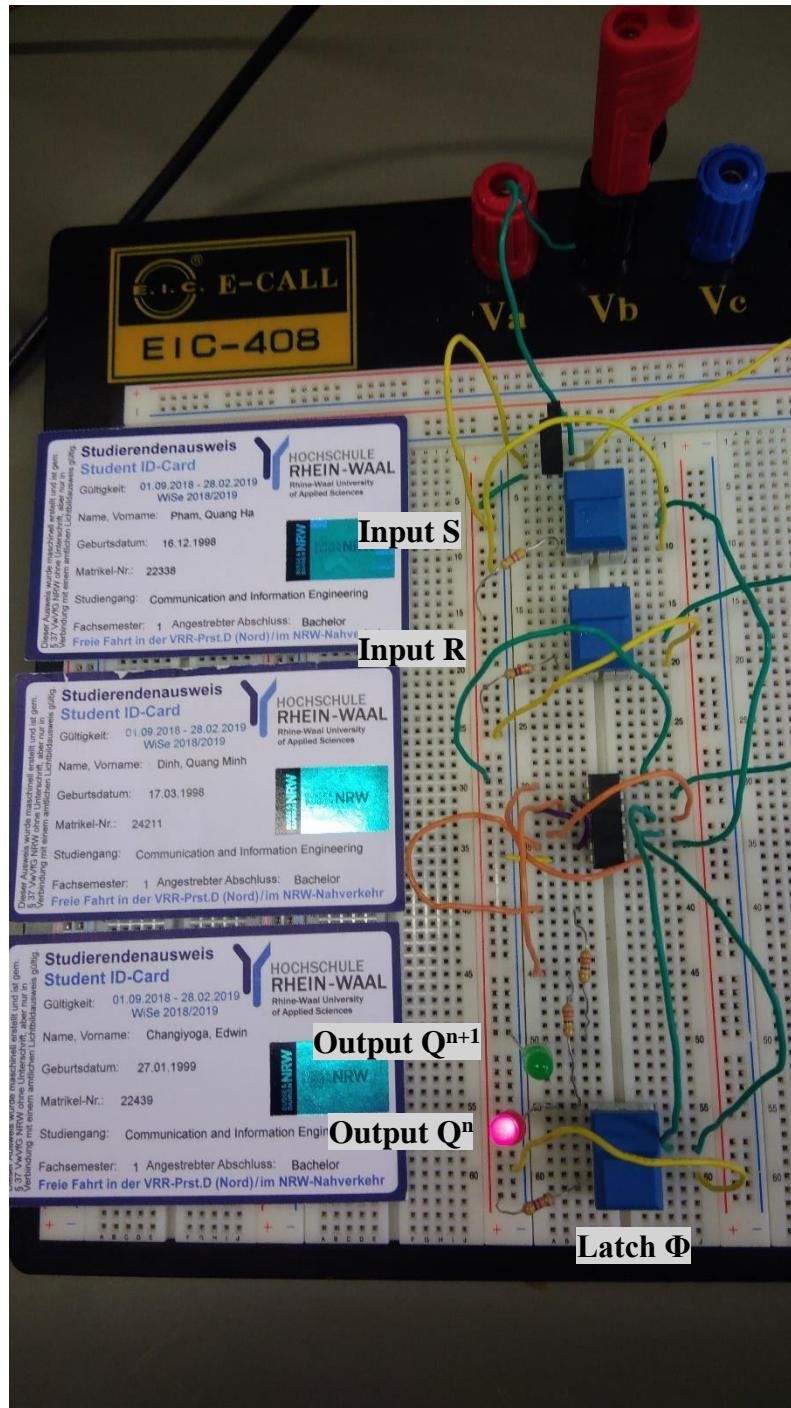
Challenge #8

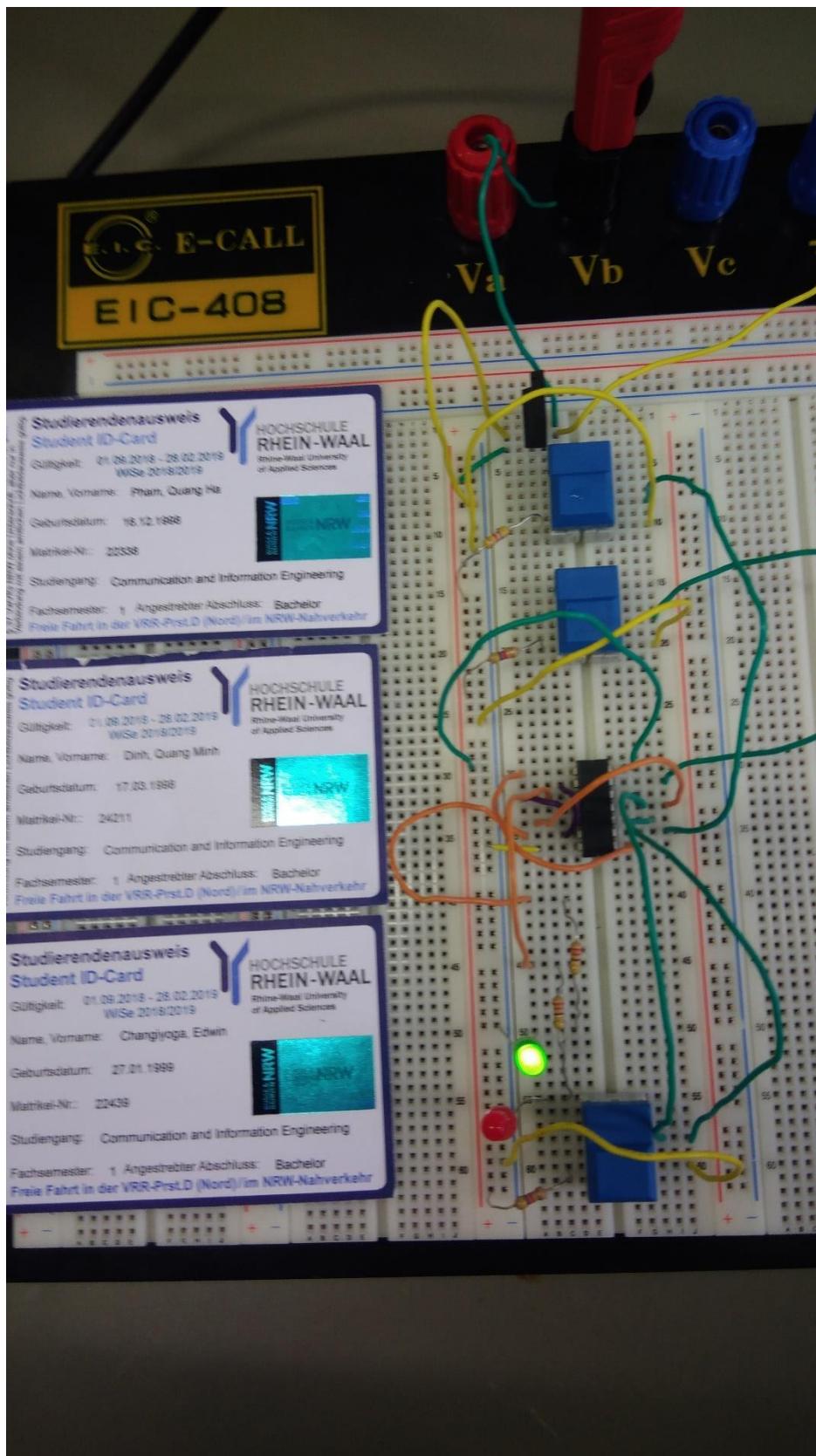
Abstract:

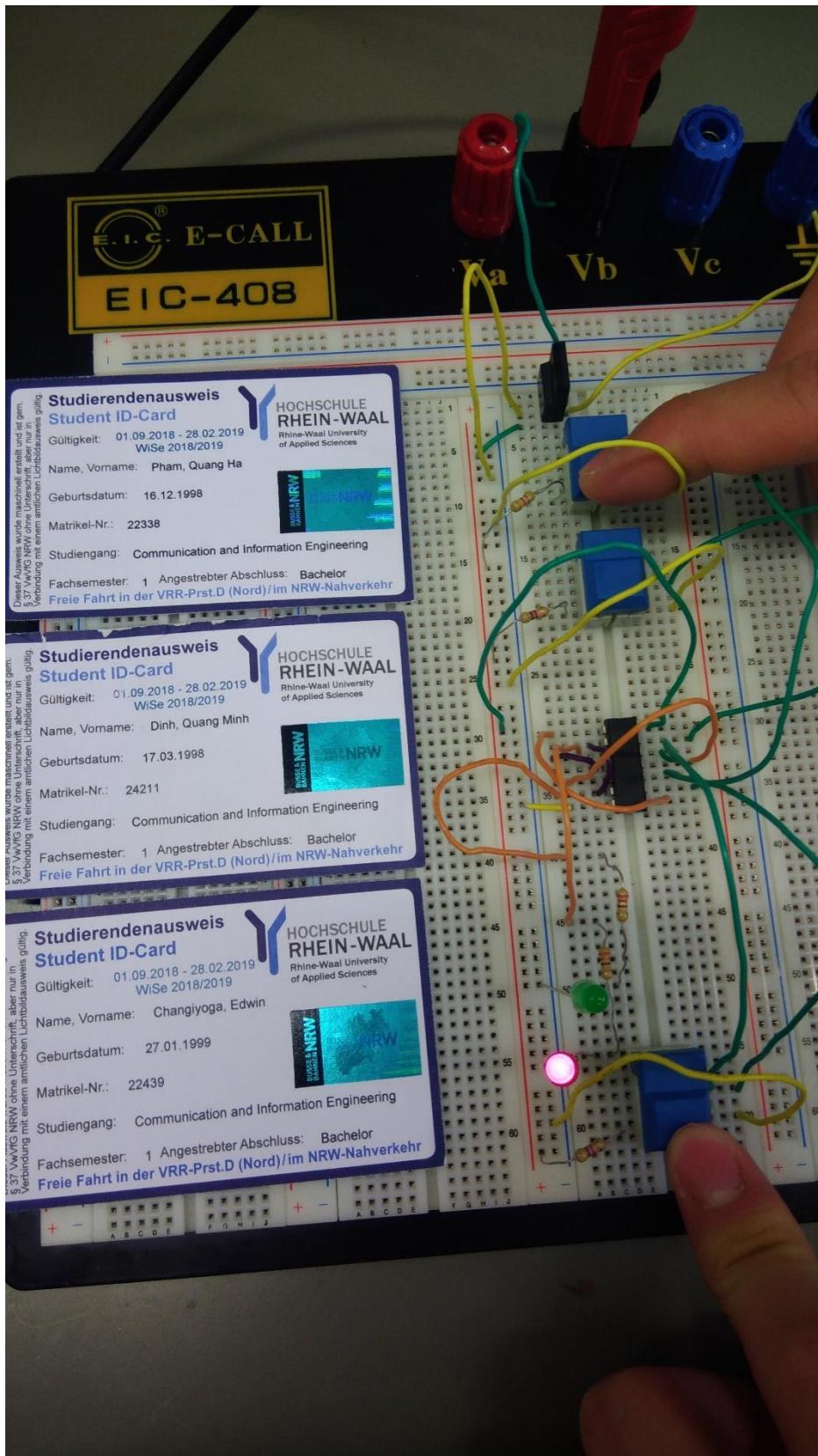
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

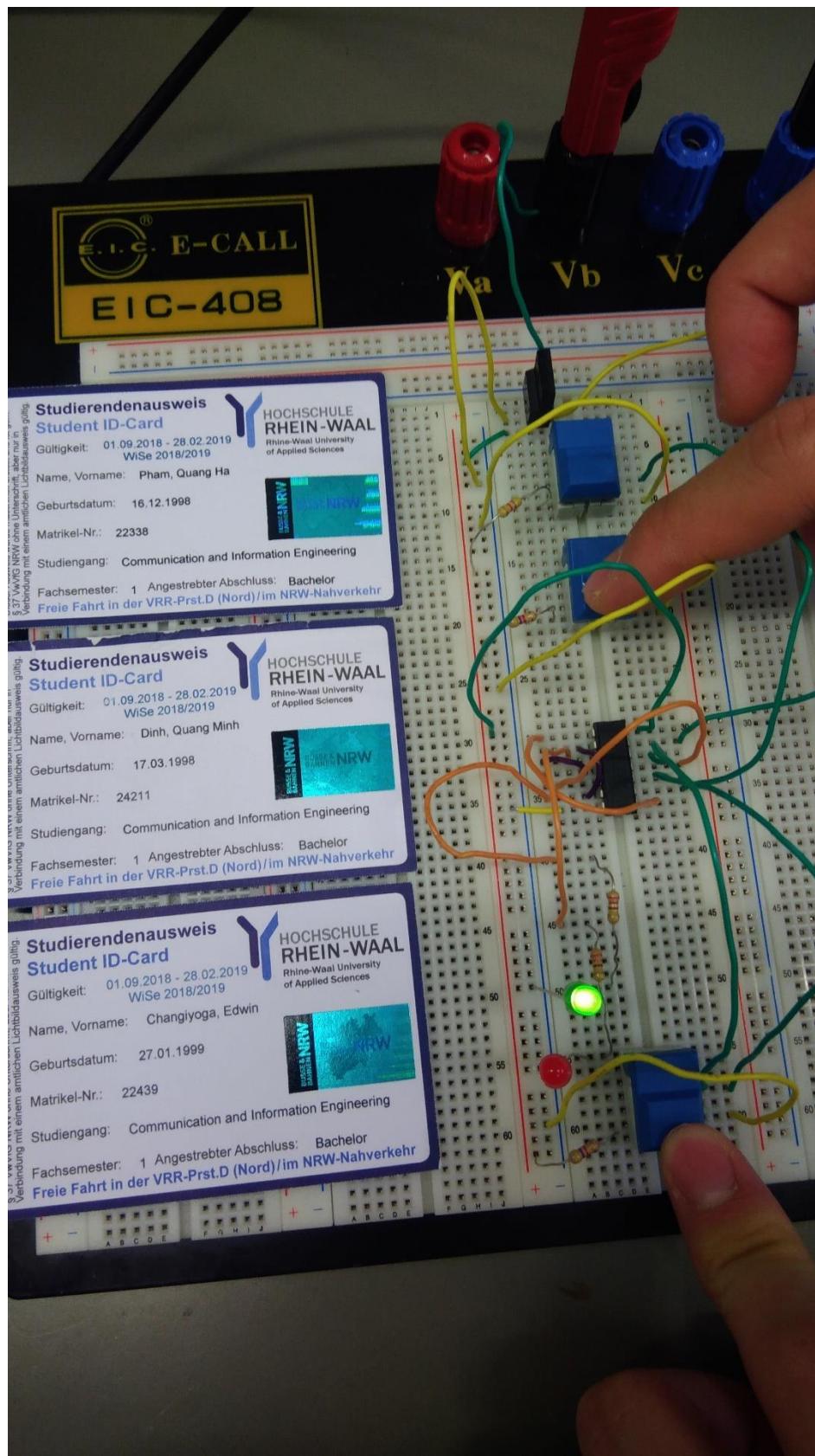
Everything is similar to challenge #6 but there is a twist: this circuit is “state-controlled” which means we can only change the input and output of this circuit only when the Latch Φ is on (input $\Phi=1$).

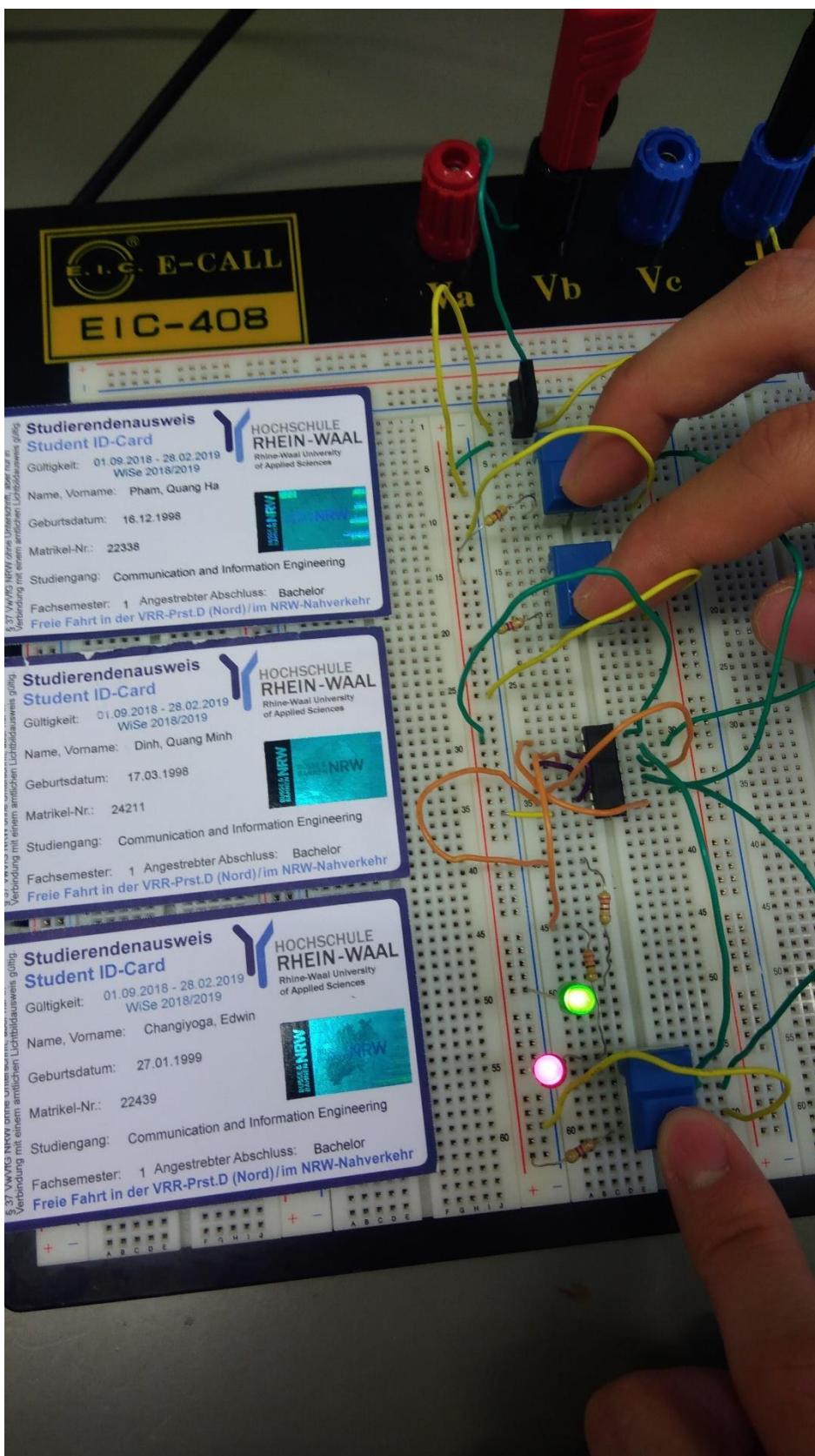
Pictures:











Result:

Φ : Latch.

Last state: The output remains their last state before Latch=0.

Φ	S	R	Q^n	Q^{n+1}
0	0	0	0	Last state
0	0	0	1	Last state
0	0	1	0	Last state
0	0	1	1	Last state
0	1	0	0	Last state
0	1	0	1	Last state
0	1	1	0	Last state
0	1	1	1	Last state
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	0
1	1	1	0	X
1	1	1	1	1

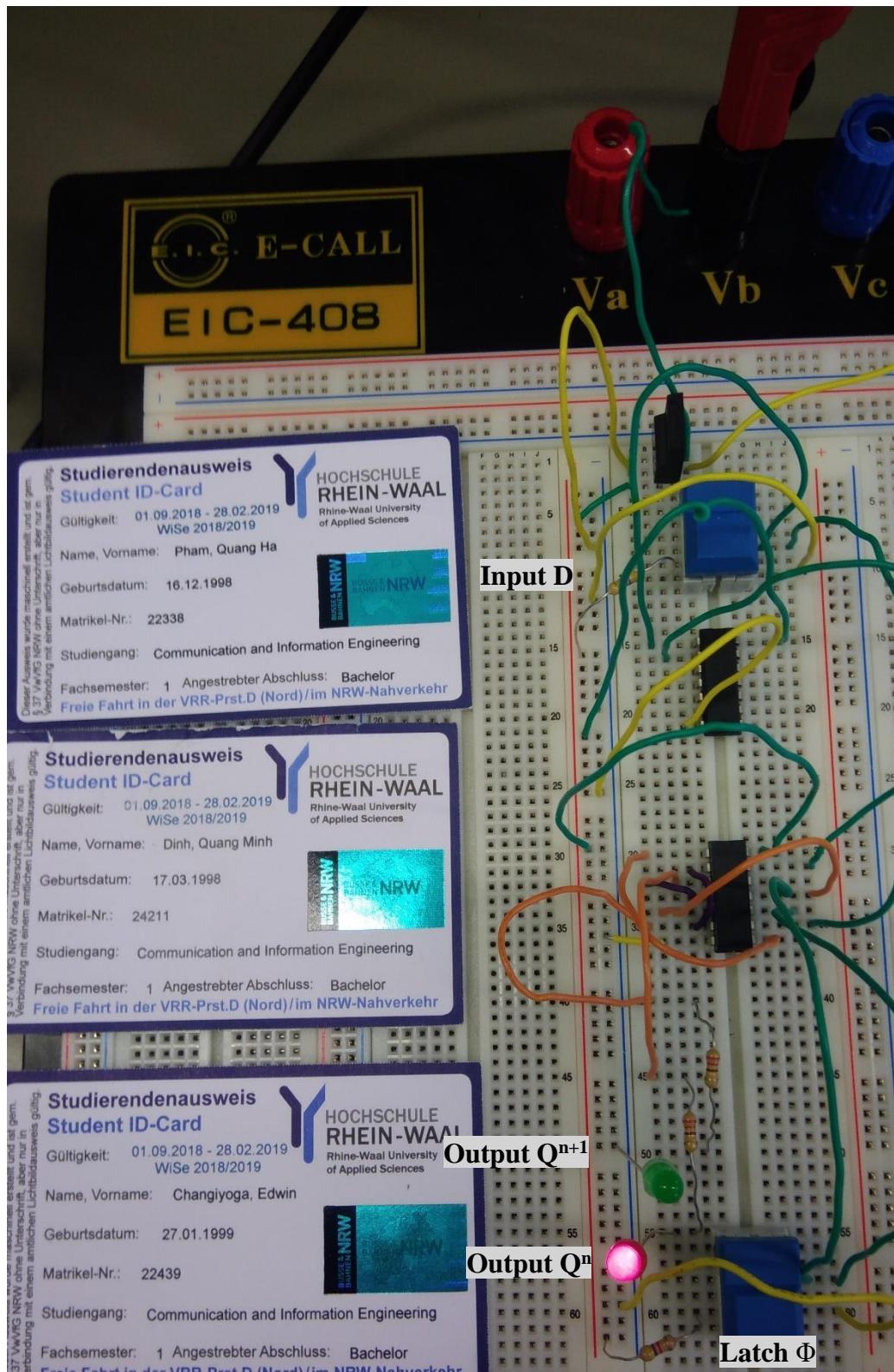
Challenge #9

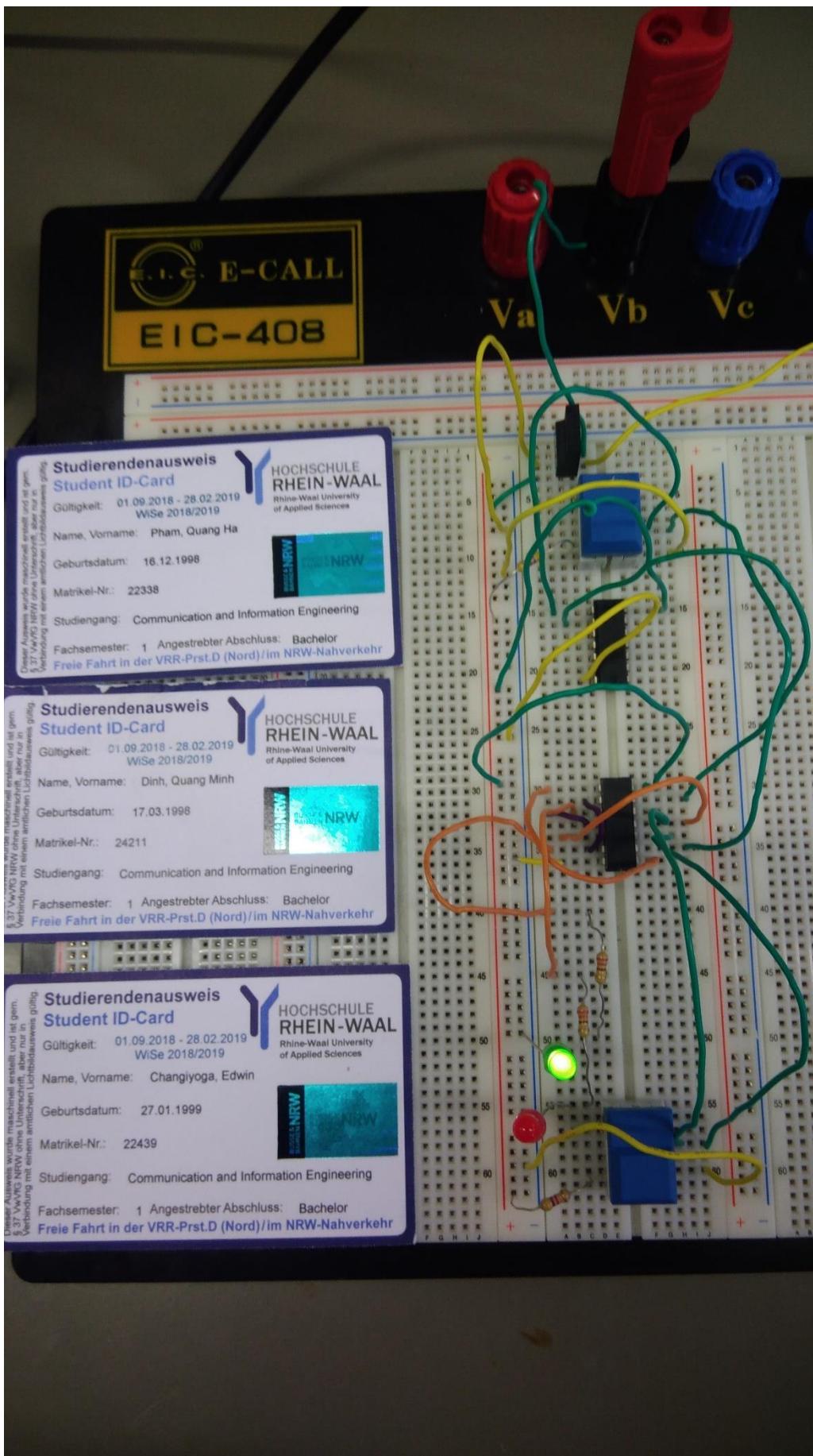
Abstract:

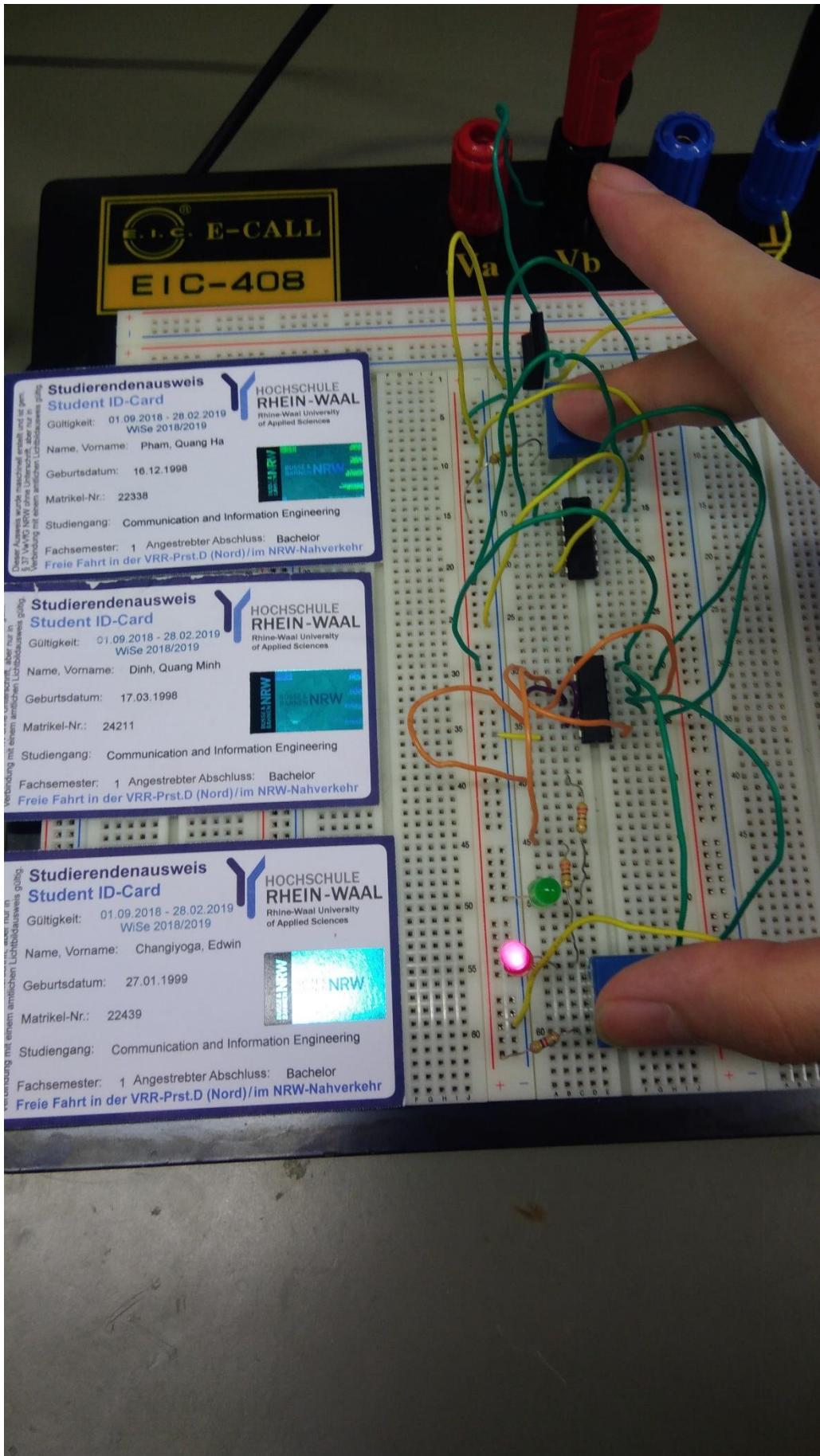
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

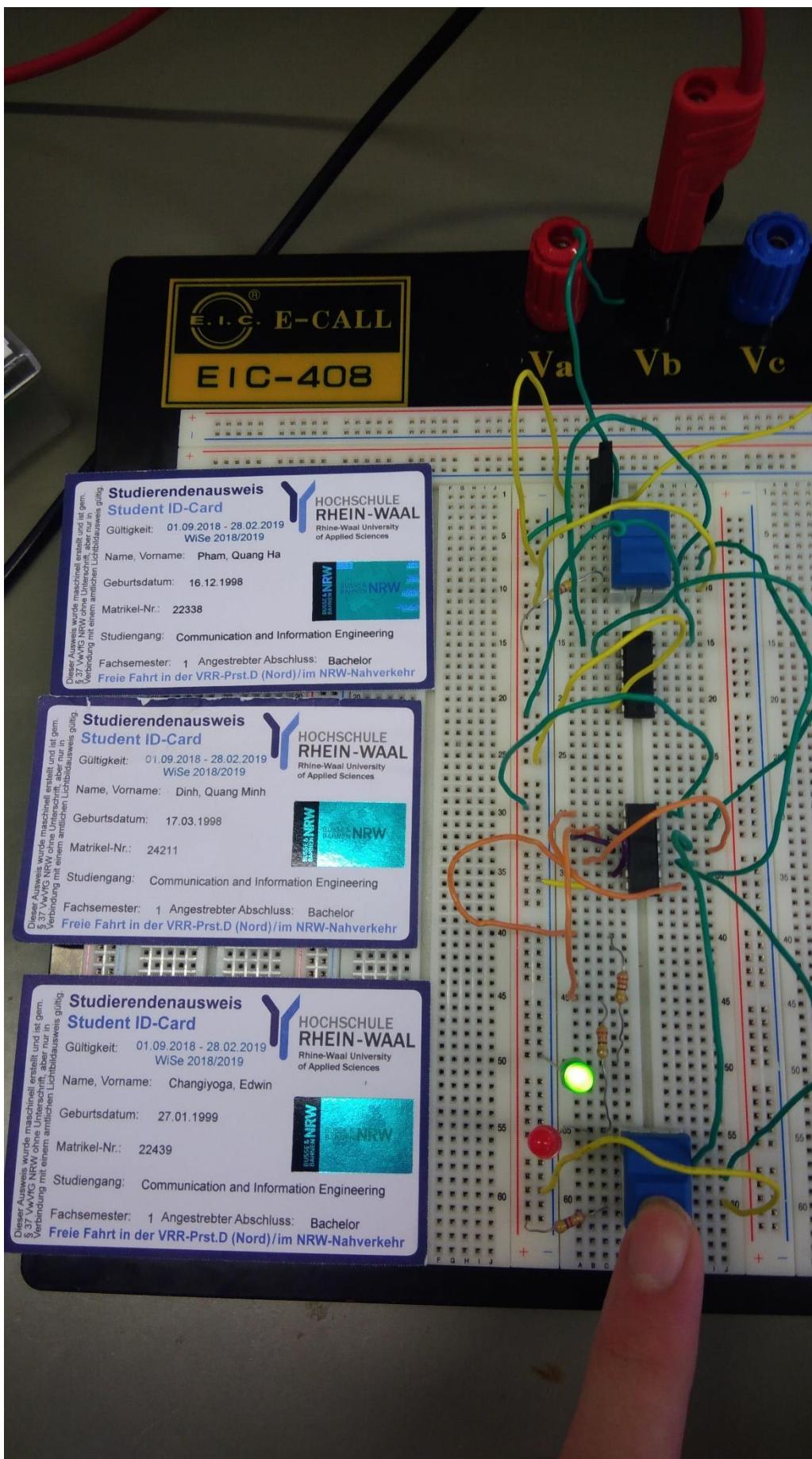
This circuit is a standard RS flipflop with a Latch included, the difference is there is an Inverter before the Reset input instead of a button. The Inverter is there to prevent the Input Set and Input Reset having the same state.

Pictures:









Result:

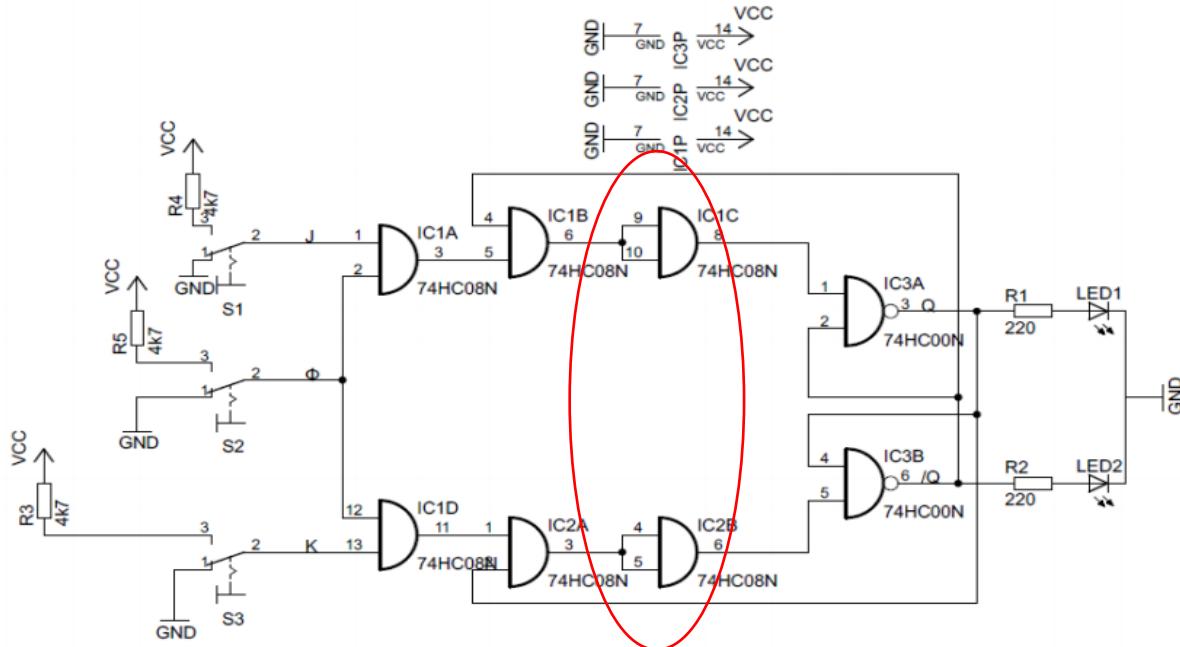
Φ	D	Q^n	Q^{n+1}
0	0	0	Last state
0	0	1	Last state
0	1	0	Last state
0	1	1	Last state
1	0	0	1
1	0	1	X
1	1	0	X
1	1	1	0

Challenge #10

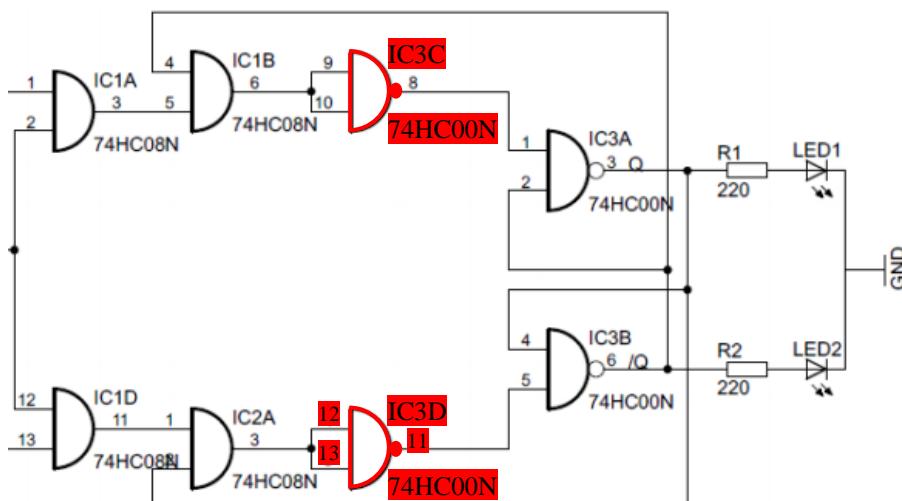
Abstract:

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

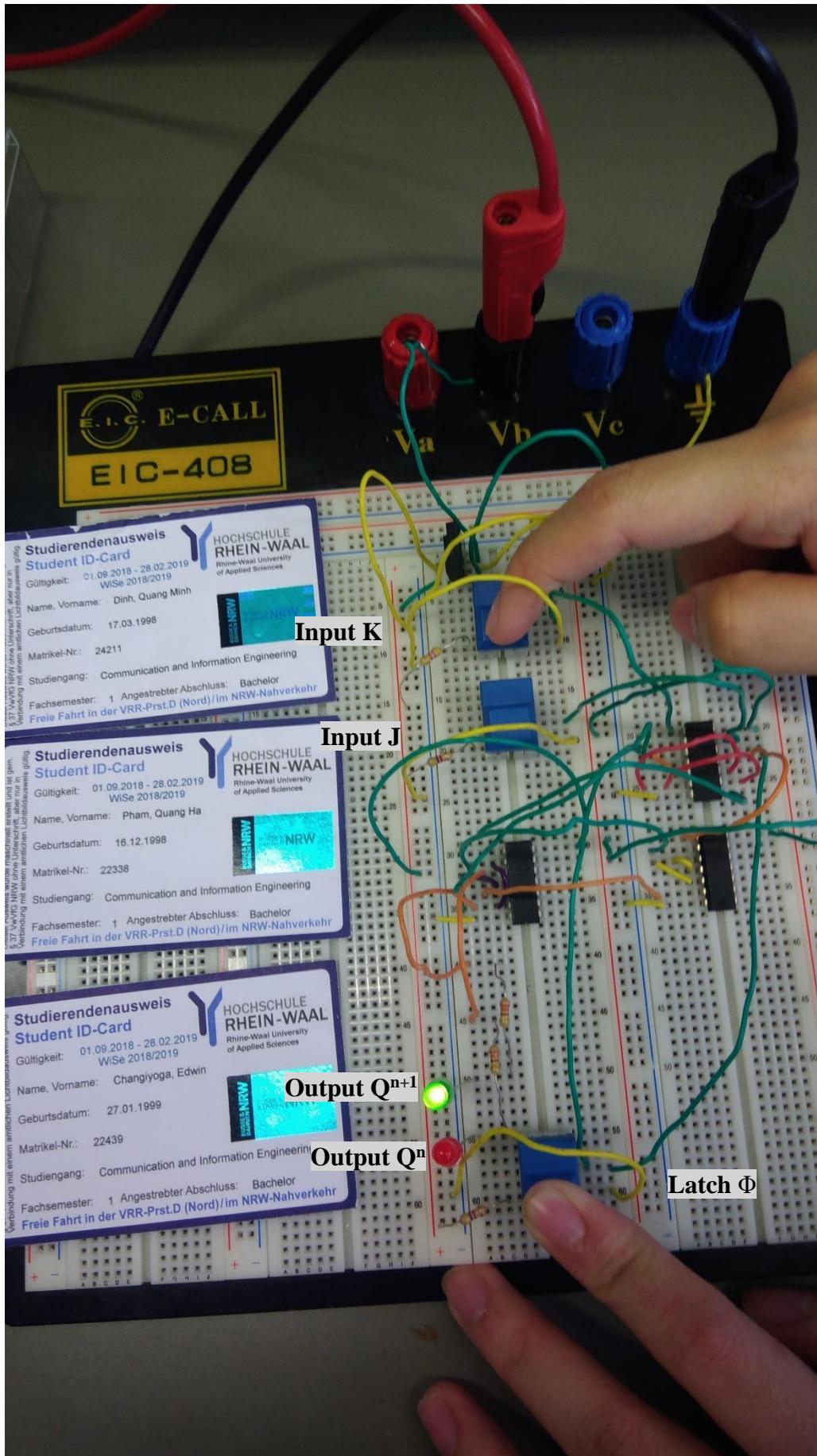
This circuit configuration is basically the same as Challenge #9 but with 2 inputs J and K instead of only input D and an Inverter.

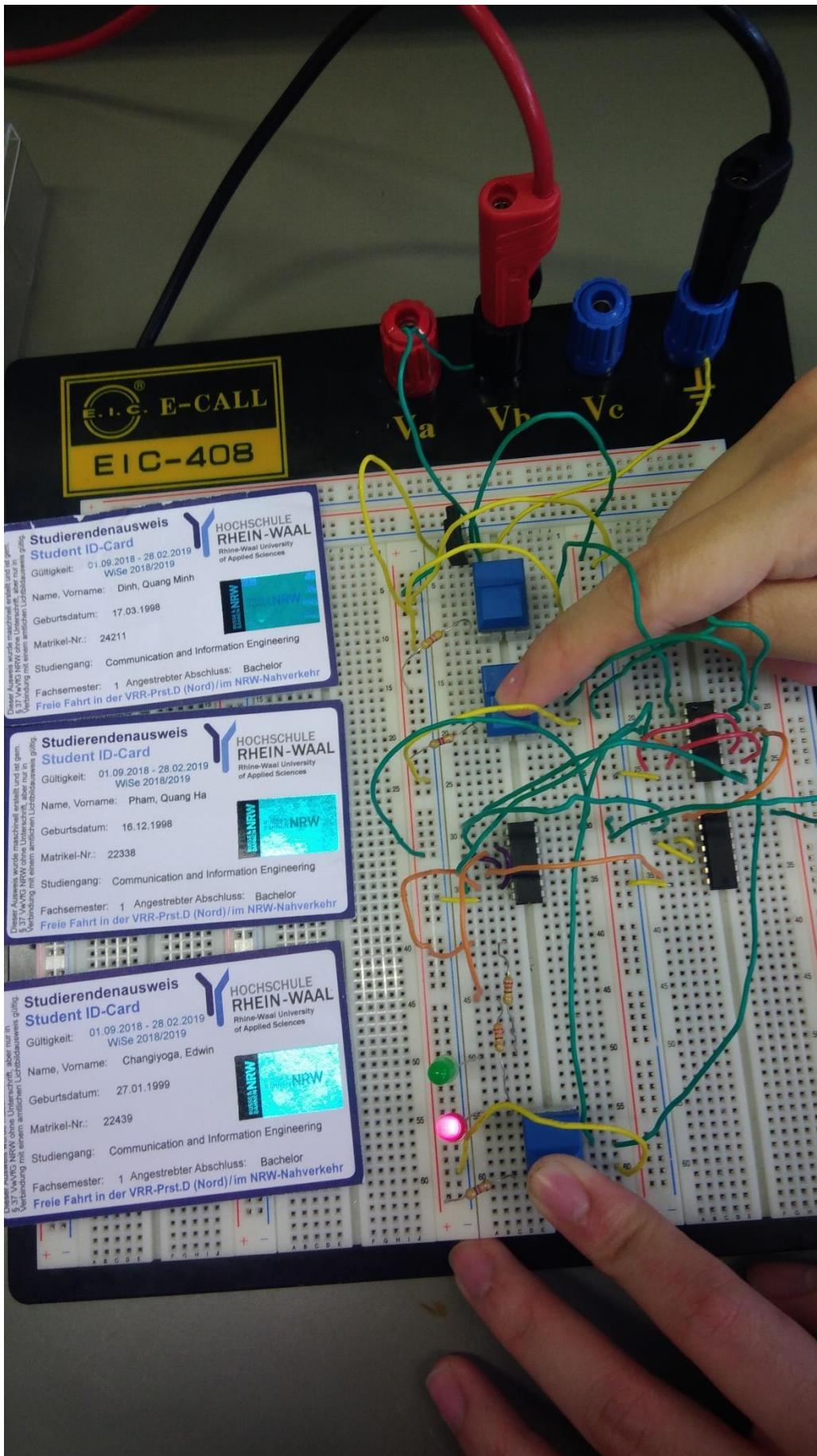


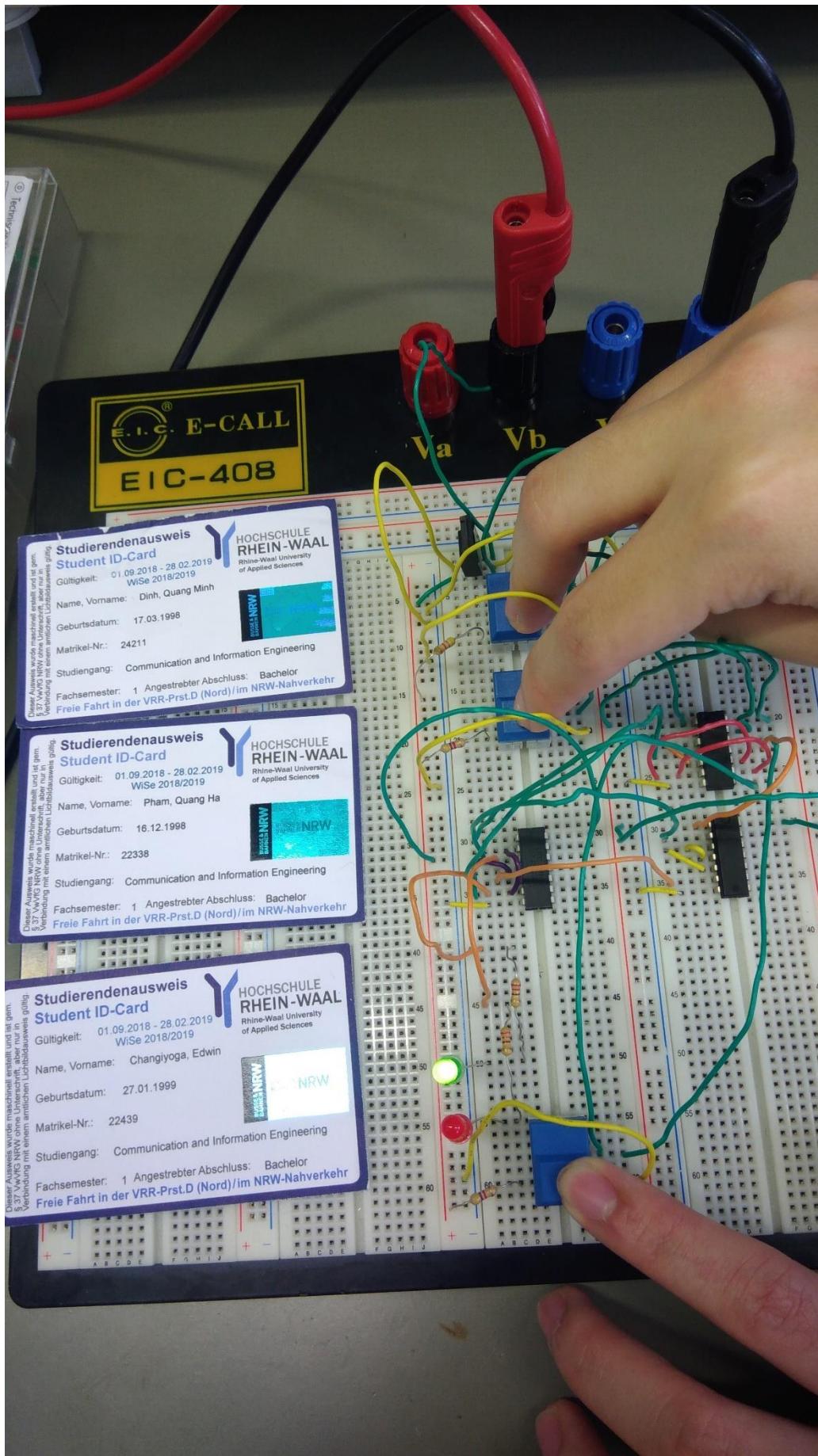
There is a strange connection in the red circle, we saw that it doesn't make sense to connect 2 AND inputs through an AND gate to get 1 output because it would be redundant, also to our knowledge that the only time that this connection makes sense is when it is connected to a NAND gate. So, we assume that this is a mistake and decided to make our own adjustments (labelled in red) to the schematics and got the correct truth table in the end according to the Internet.

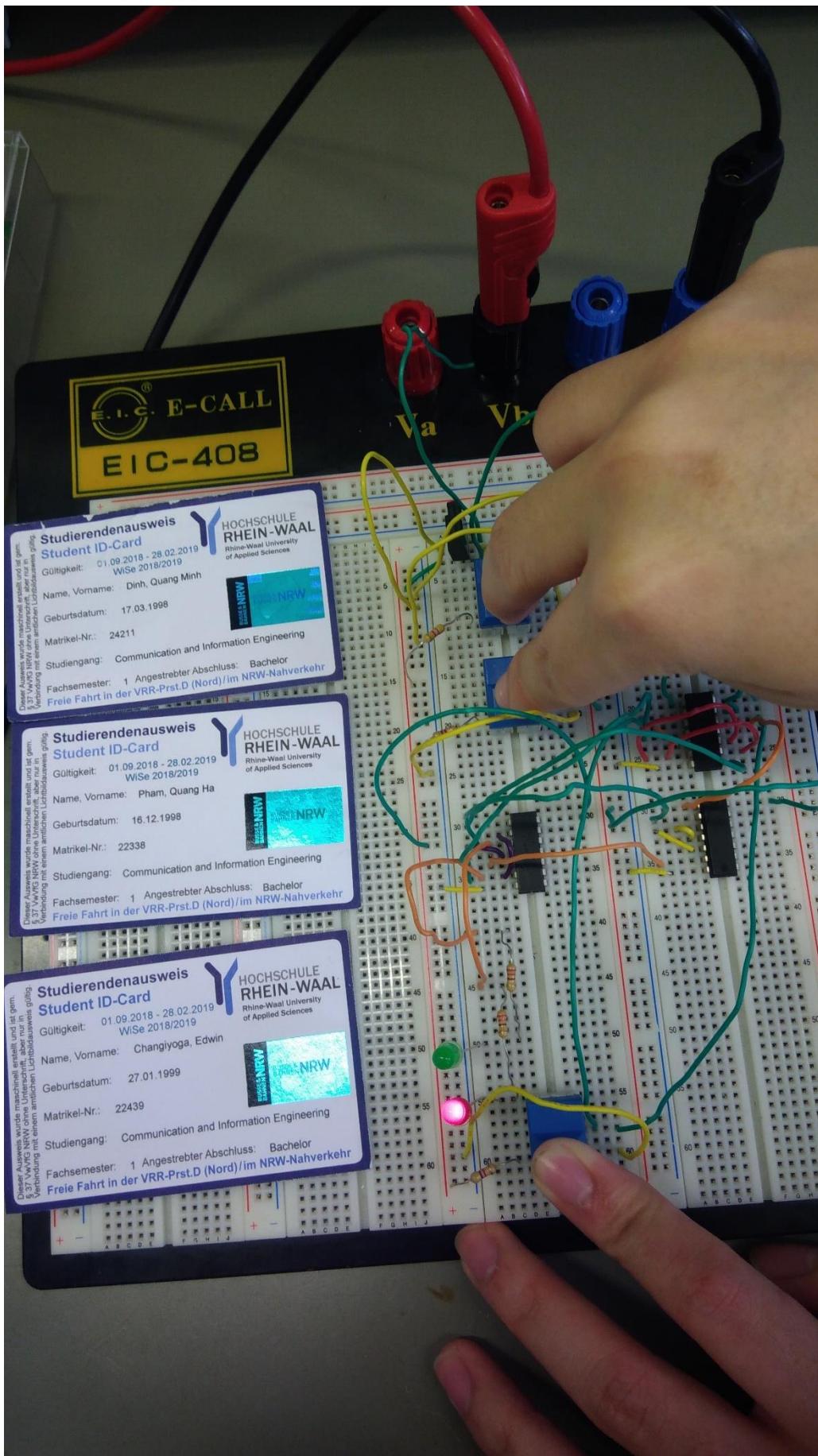


Pictures:









Result:

*: when $J=K=1$, the state changes after toggling the Latch Φ on and off.

Φ	J	K	Q^n	Q^{n+1}
0	0	0	0	Last state
0	0	0	1	Last state
0	0	1	0	Last state
0	0	1	1	Last state
0	1	0	0	Last state
0	1	0	1	Last state
0	1	1	0	Last state
0	1	1	1	Last state
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	X
1	1	0	0	X
1	1	0	1	0
1	1	1	0	1*
1	1	1	1	0*

Challenge #11

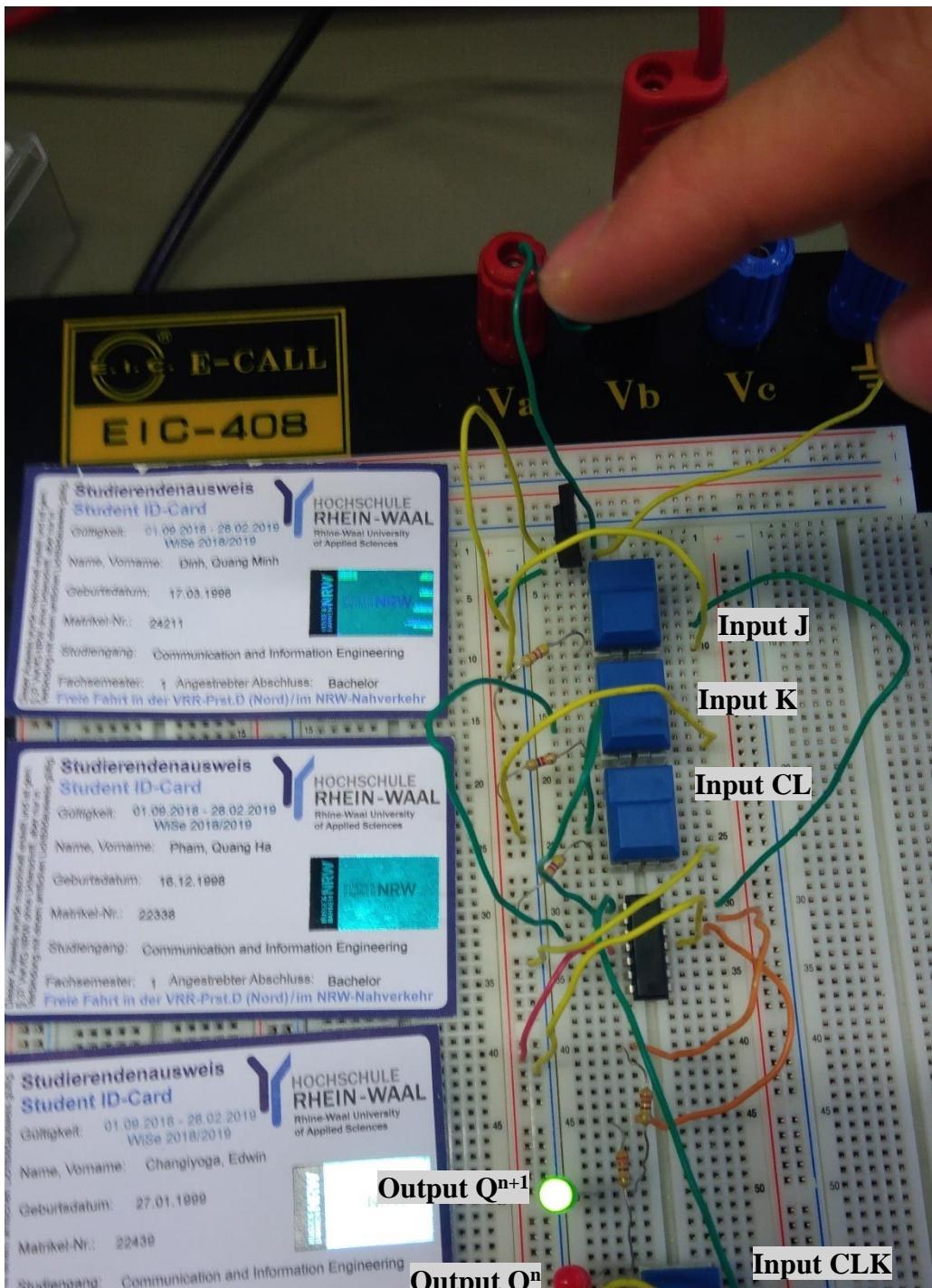
Abstract:

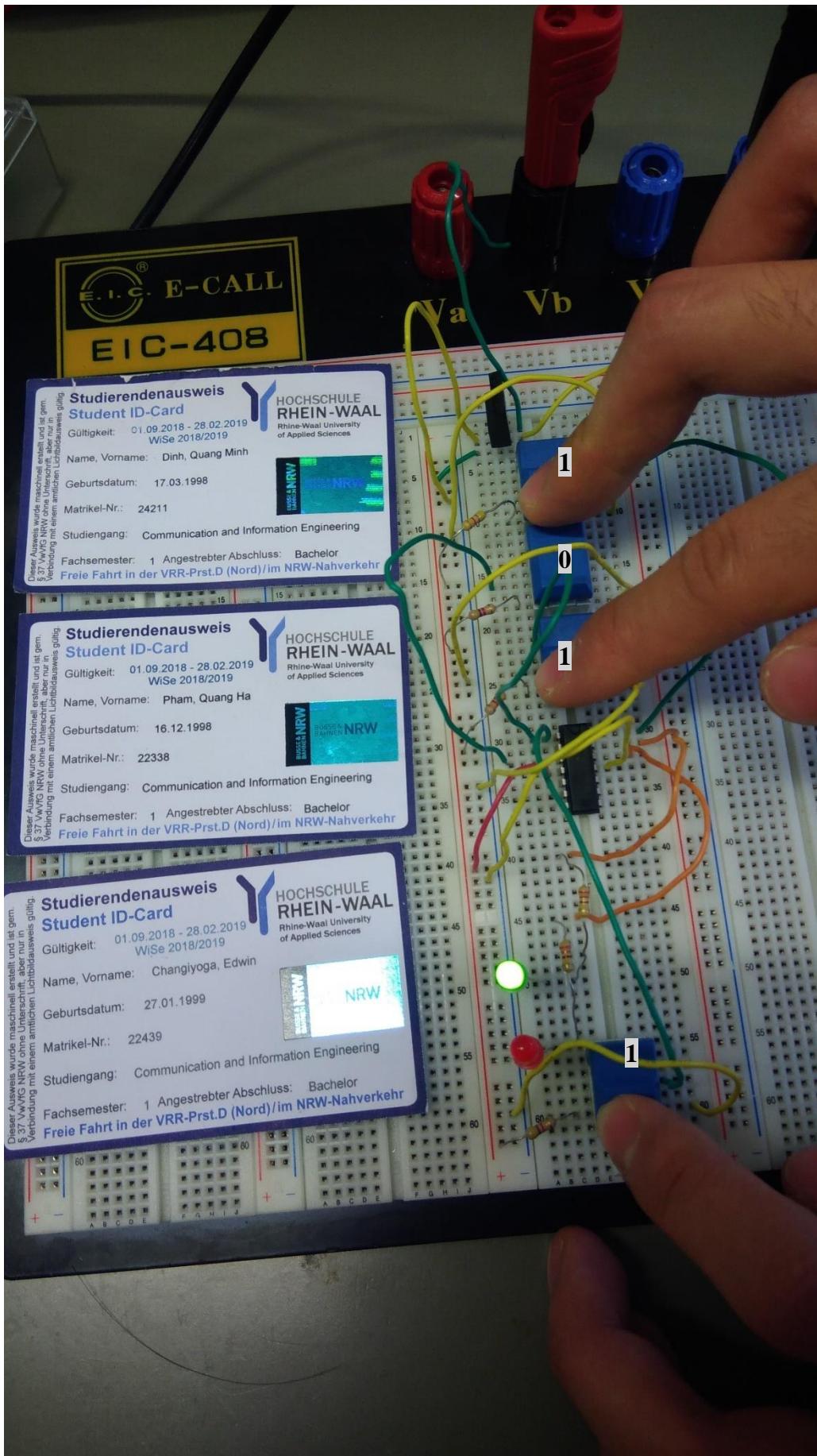
Our group managed to follow the steps given in the Description and completed the challenge. But there are some things to note.

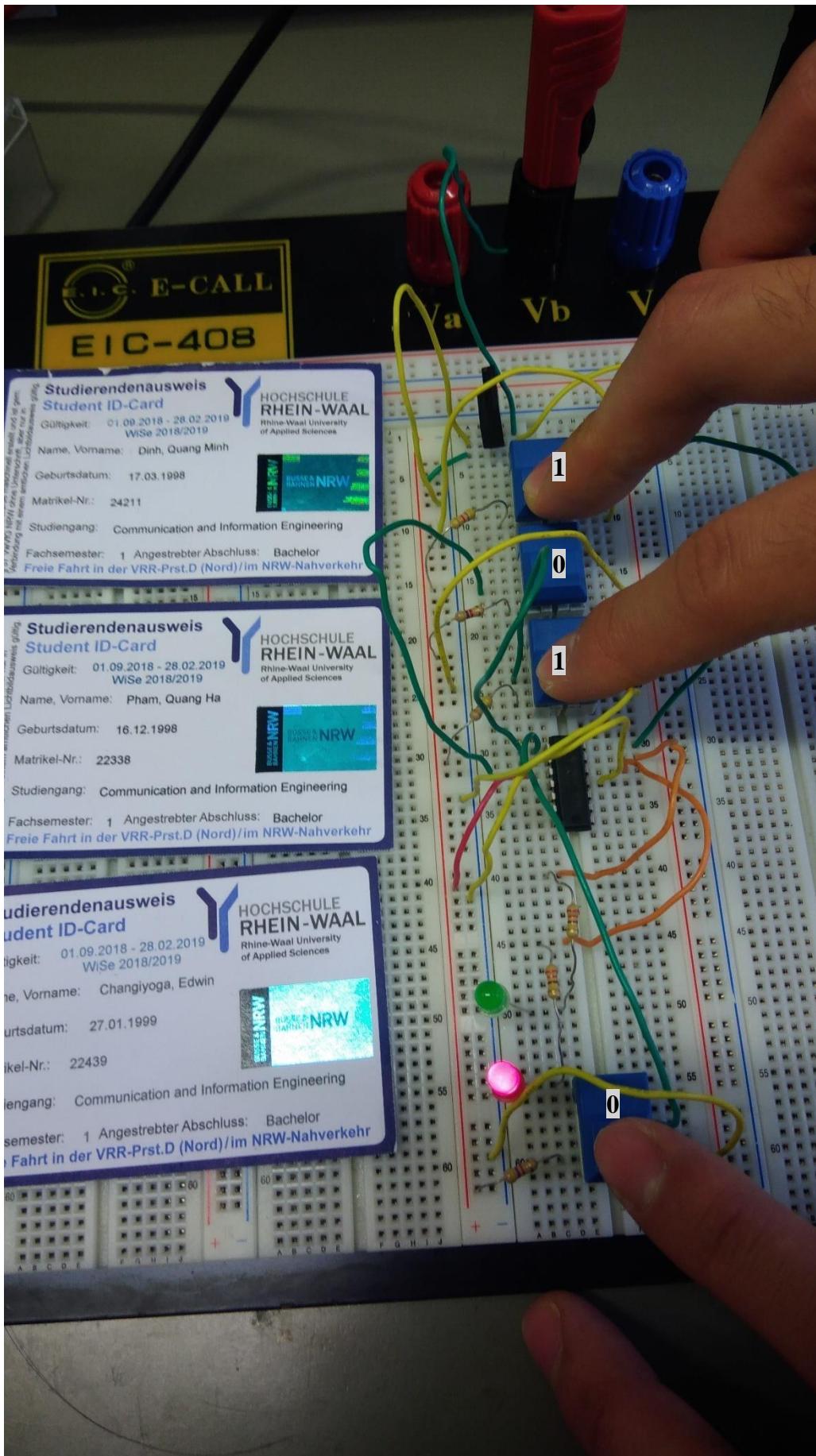
The difference between a JK-Flipflop and a JK-Master-Slave-Flipflop is that it takes 1 on and off cycle to change the output with CLK (Clock) and reset the output with CL (Reset), this is happening because of the way that a JK-Master-Slave-Flipflop is made.

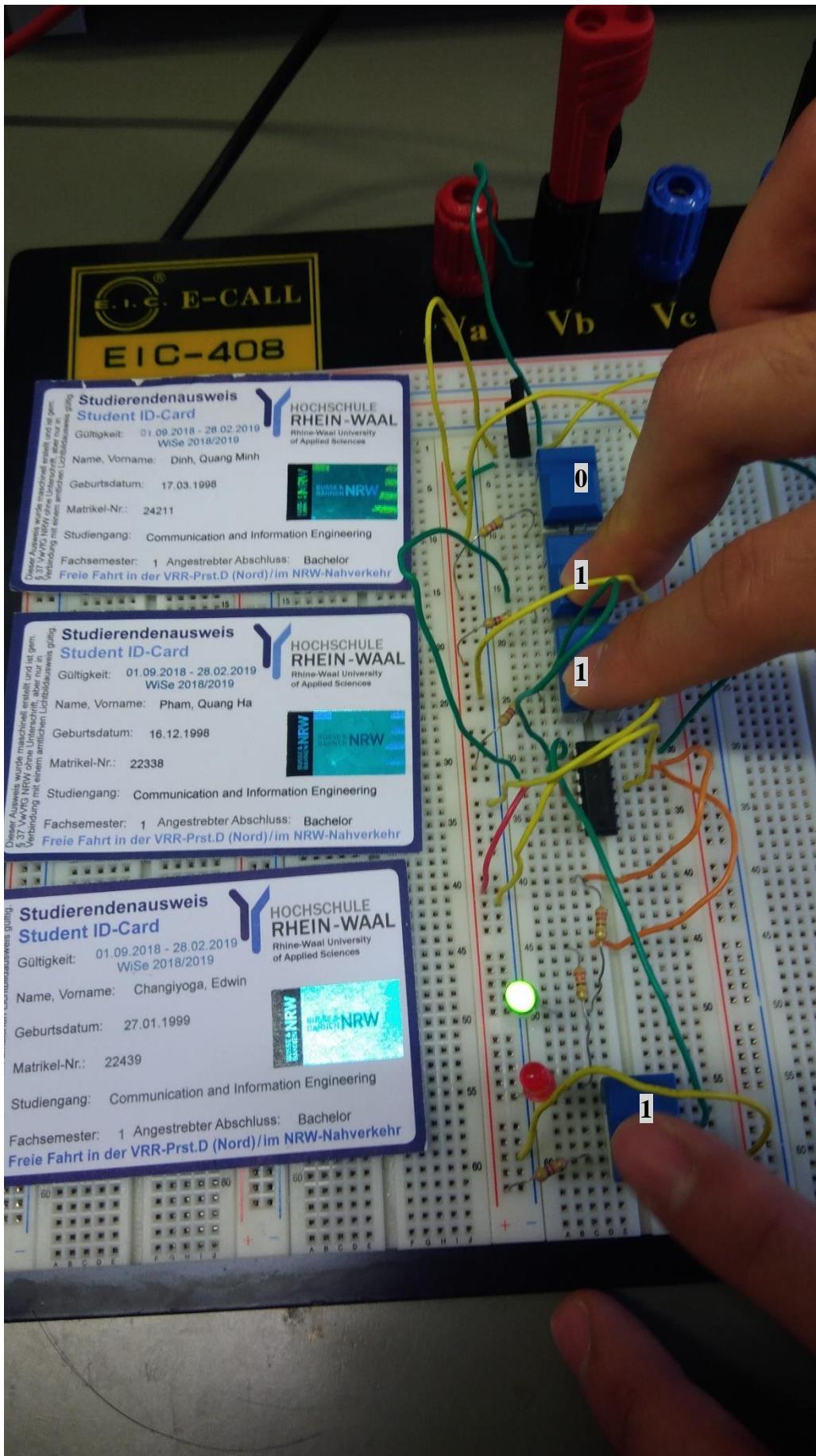
When the CLK or CL is at positive trigger (1) the “master” is learning the information of the input (becomes active) but the “slave” is still taking a rest (becomes inactive) so no output is produced, only when the CLK/CL is transitioning from 1 to 0 that the “master” will take a rest (becomes inactive) and give the information to the “slave” so that the “slave” can work (becomes active) and produce the output.

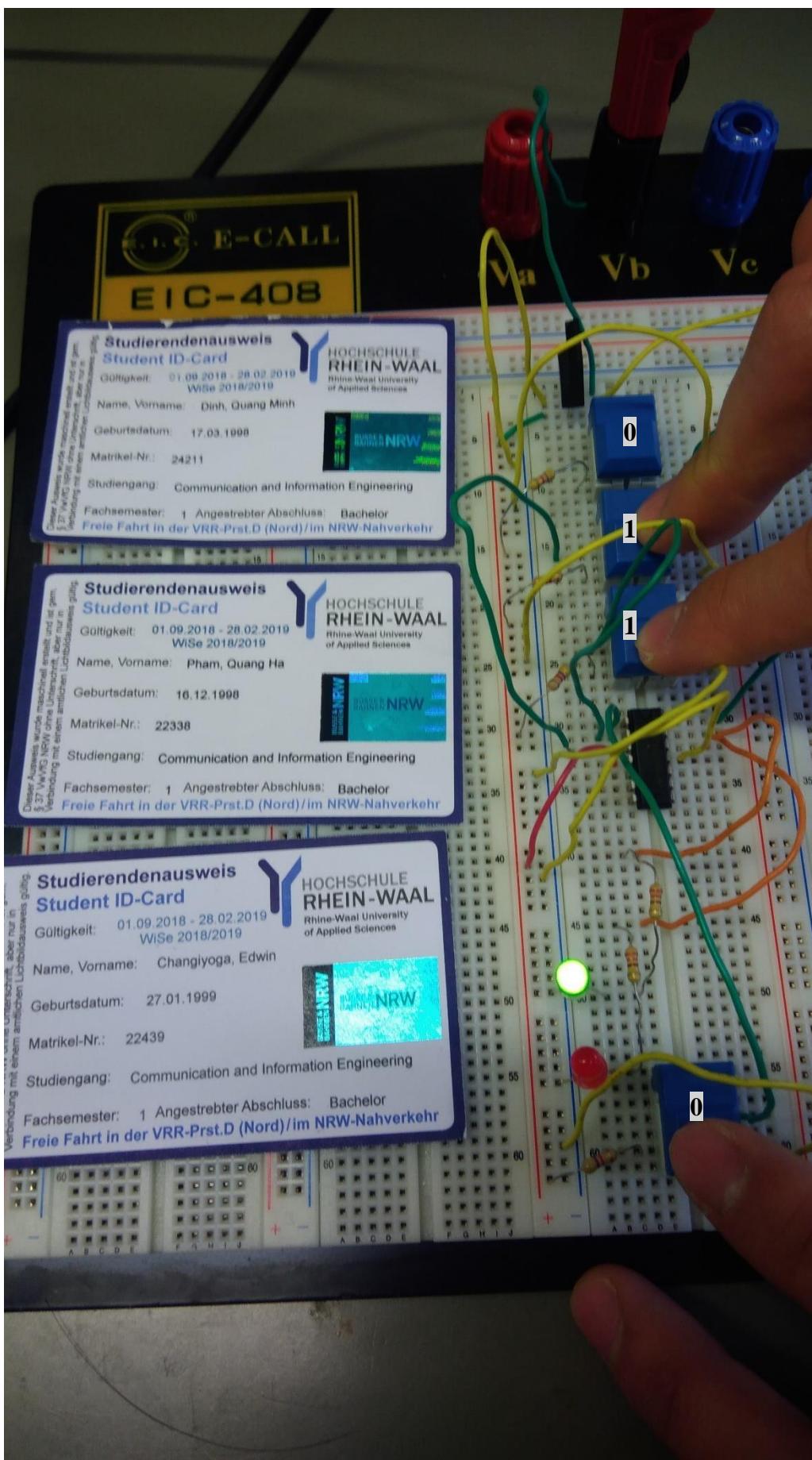
Pictures:

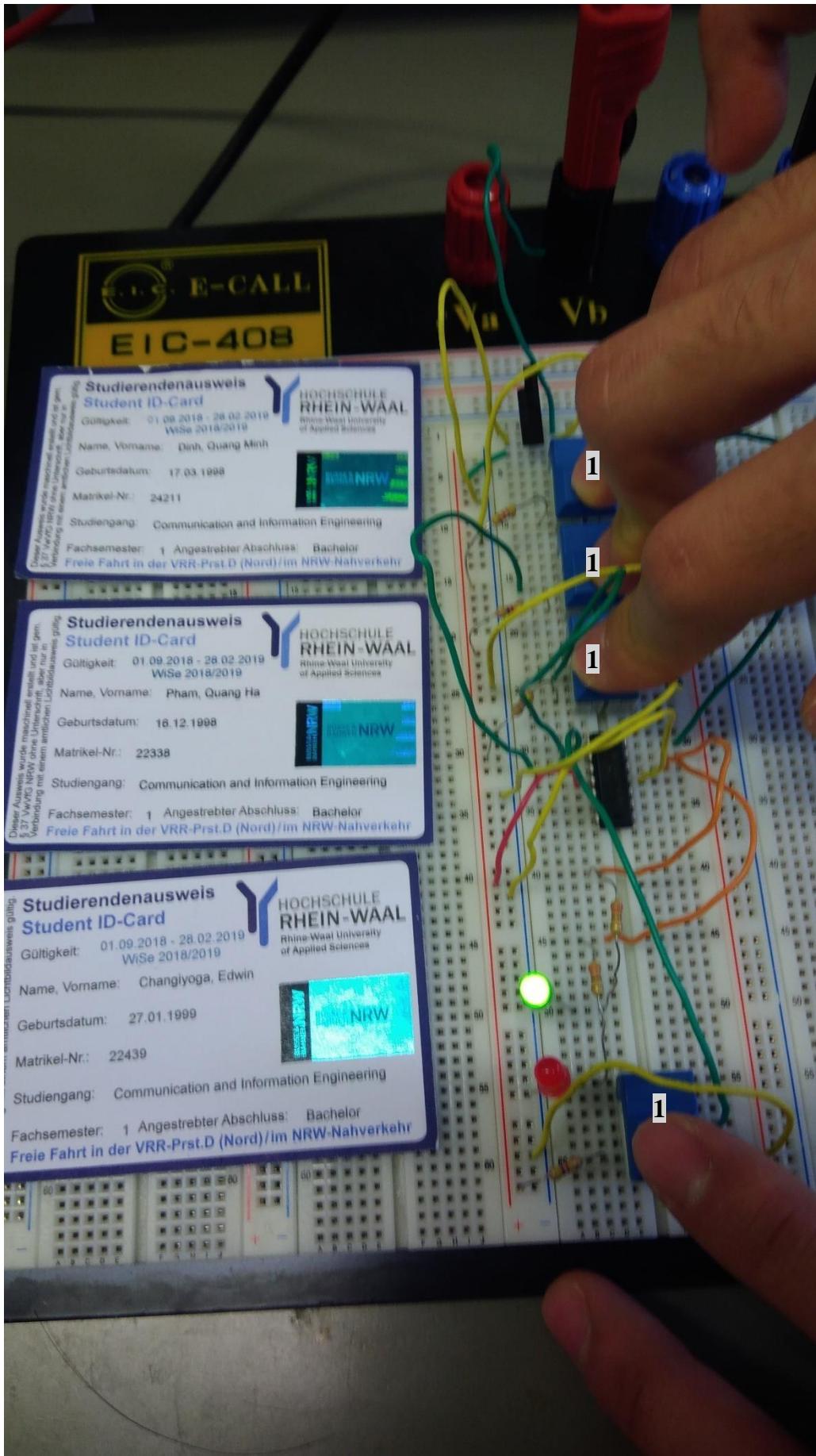




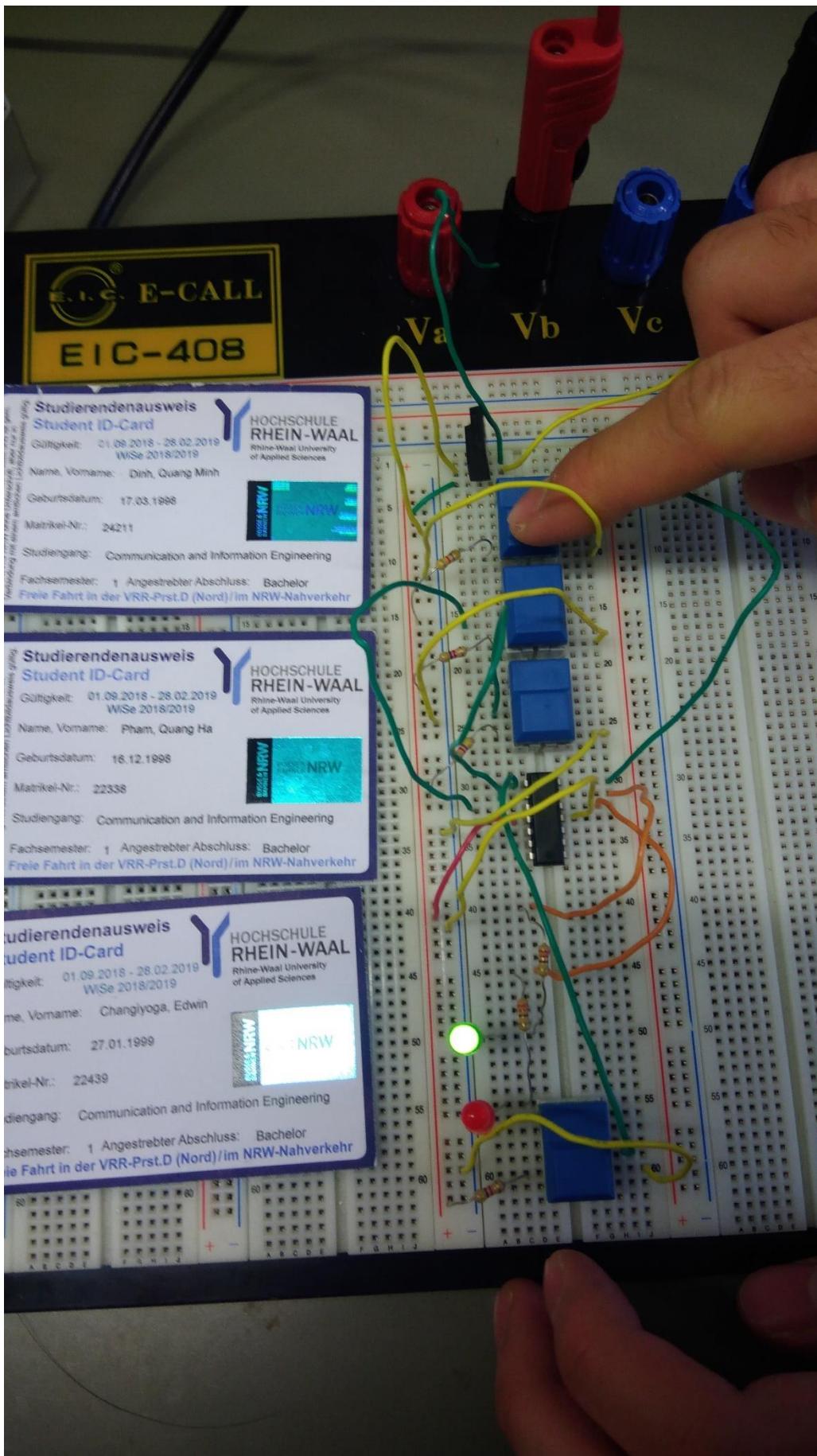


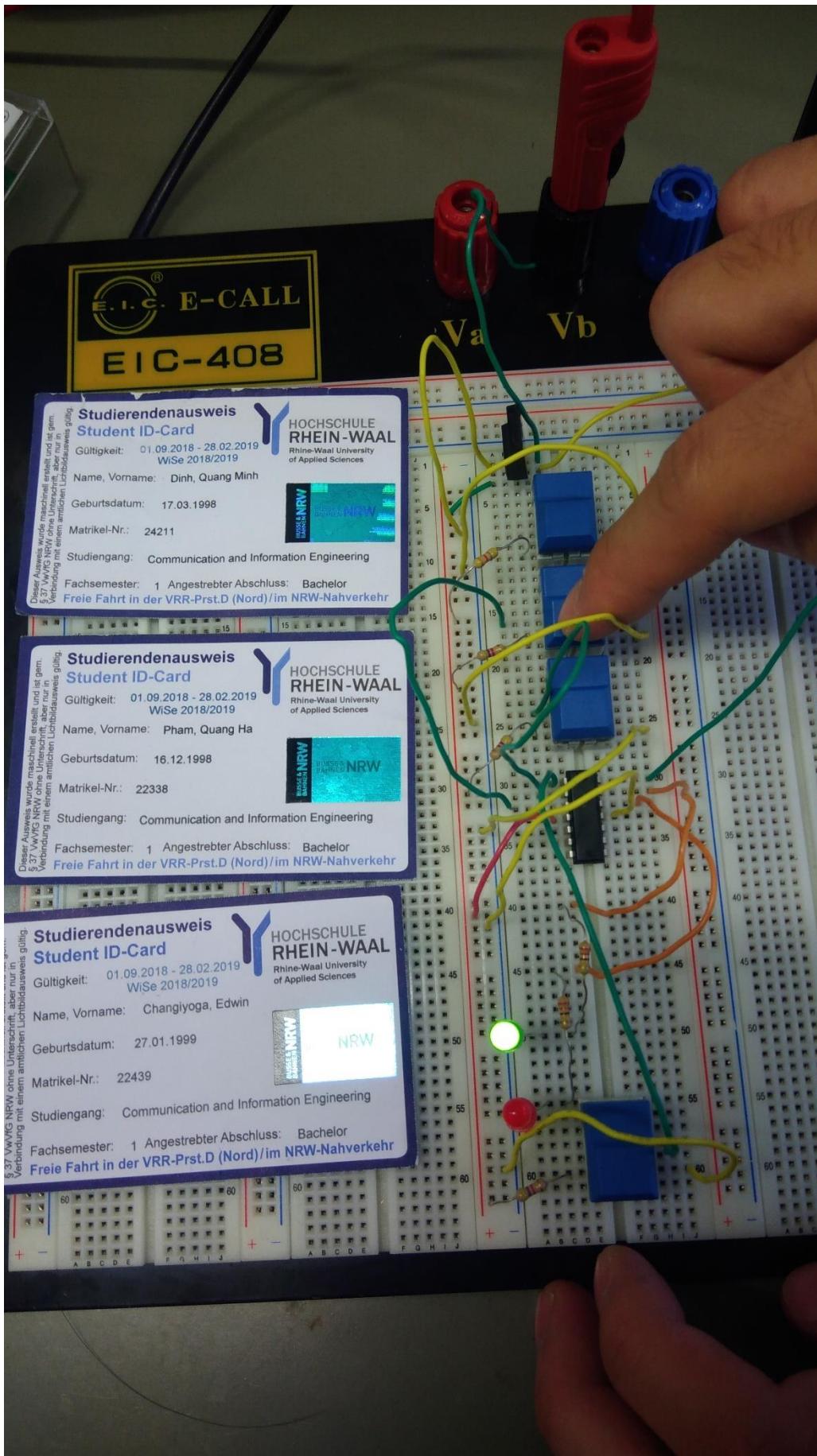


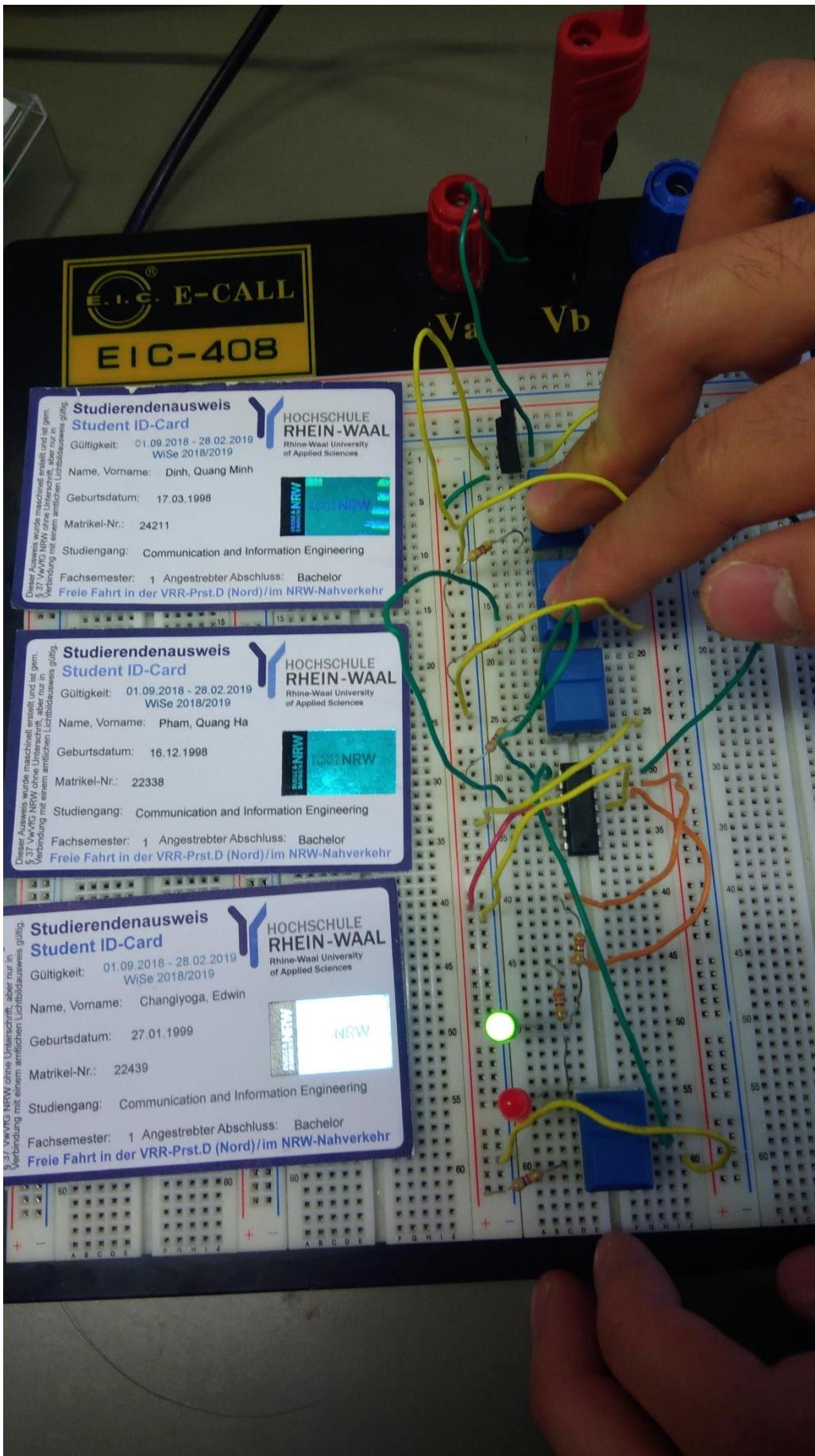


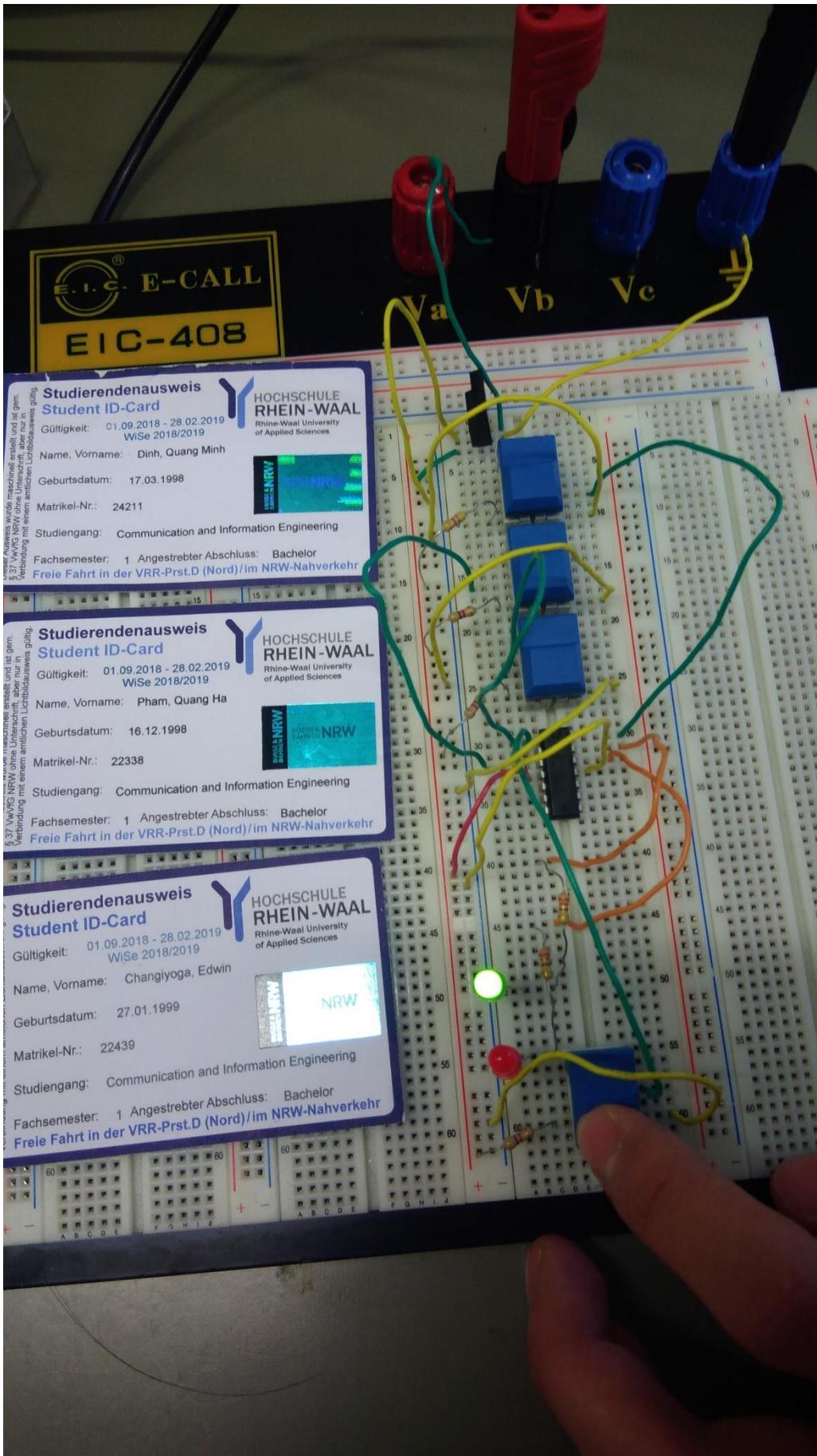


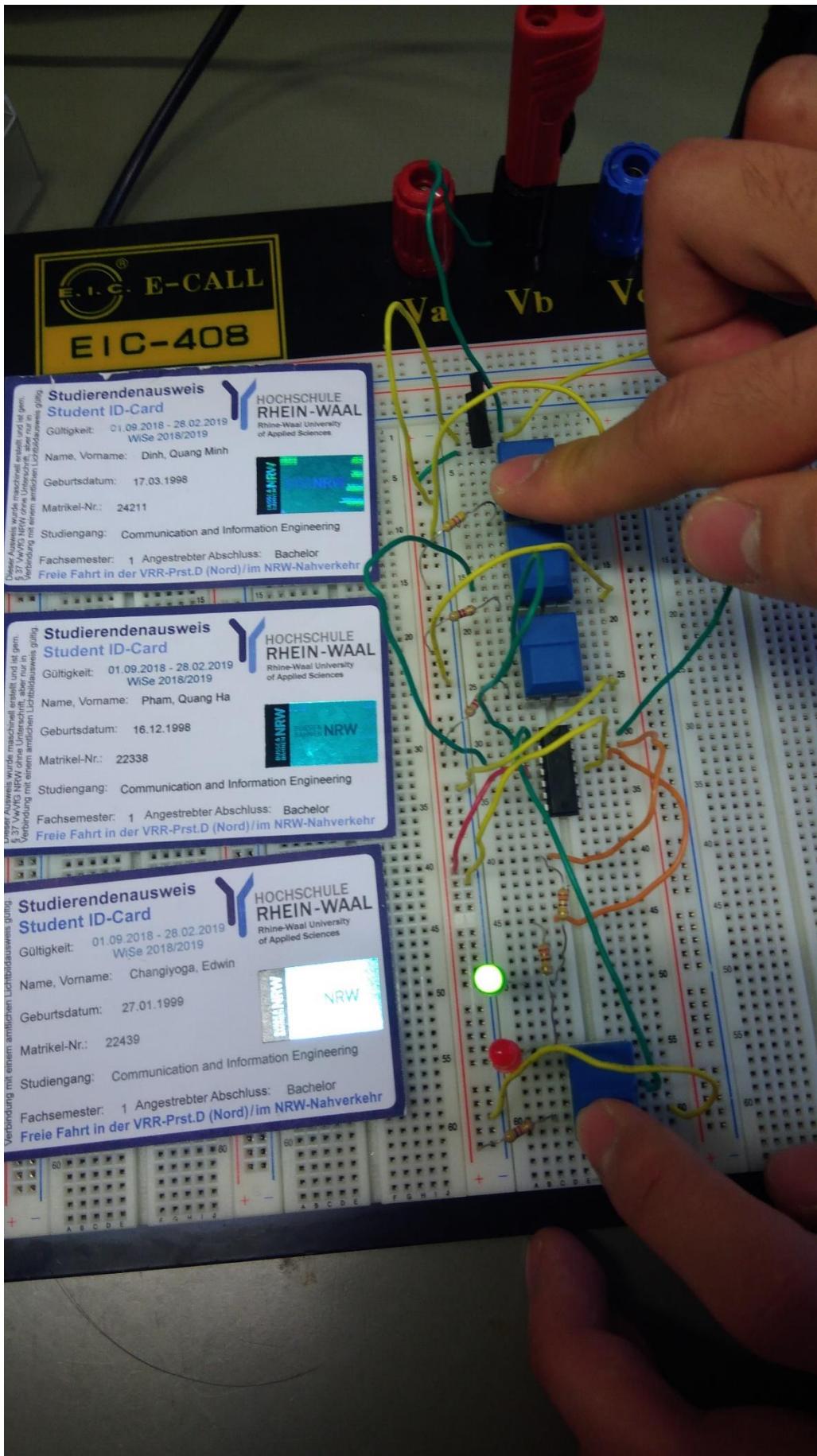


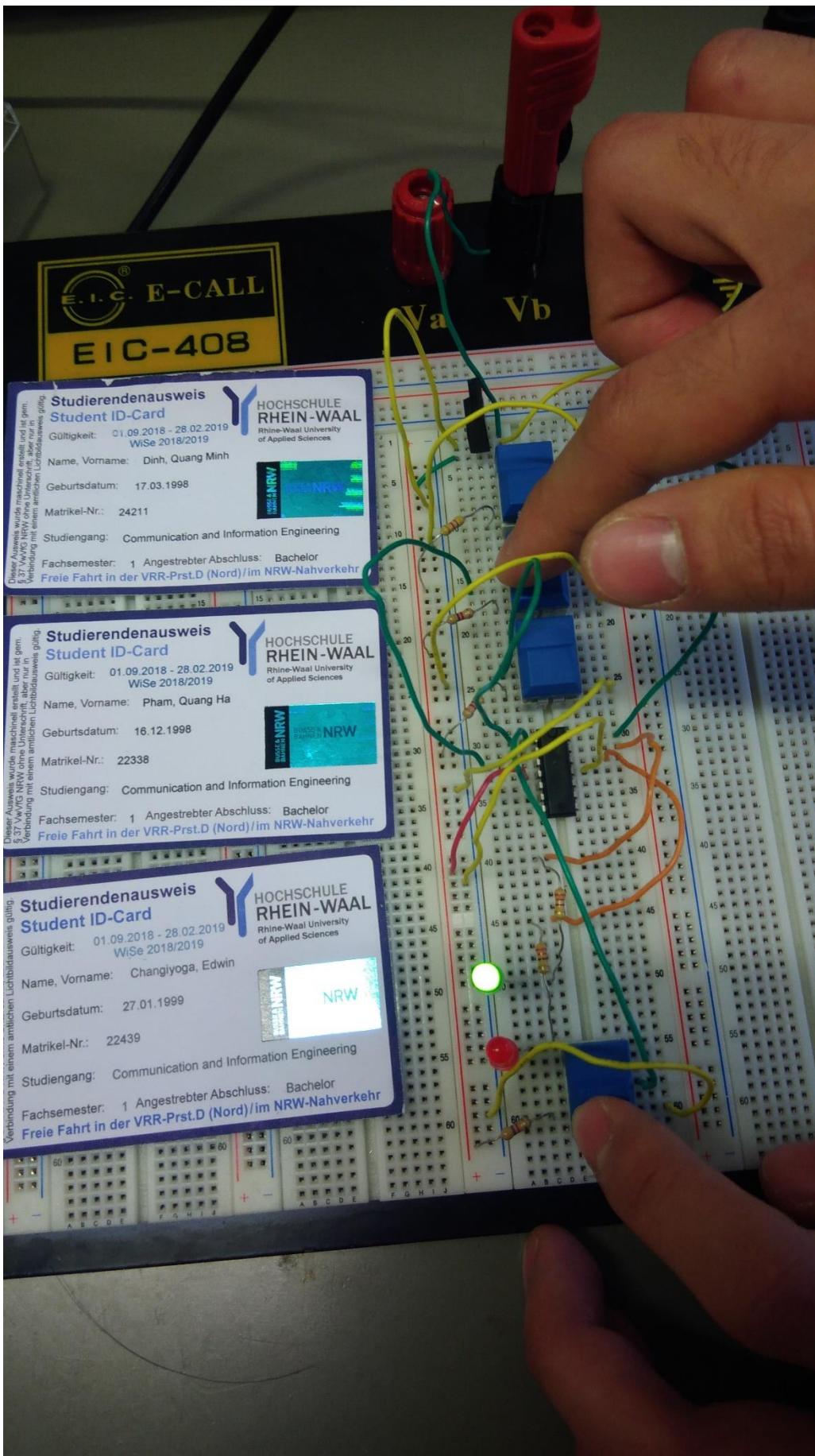


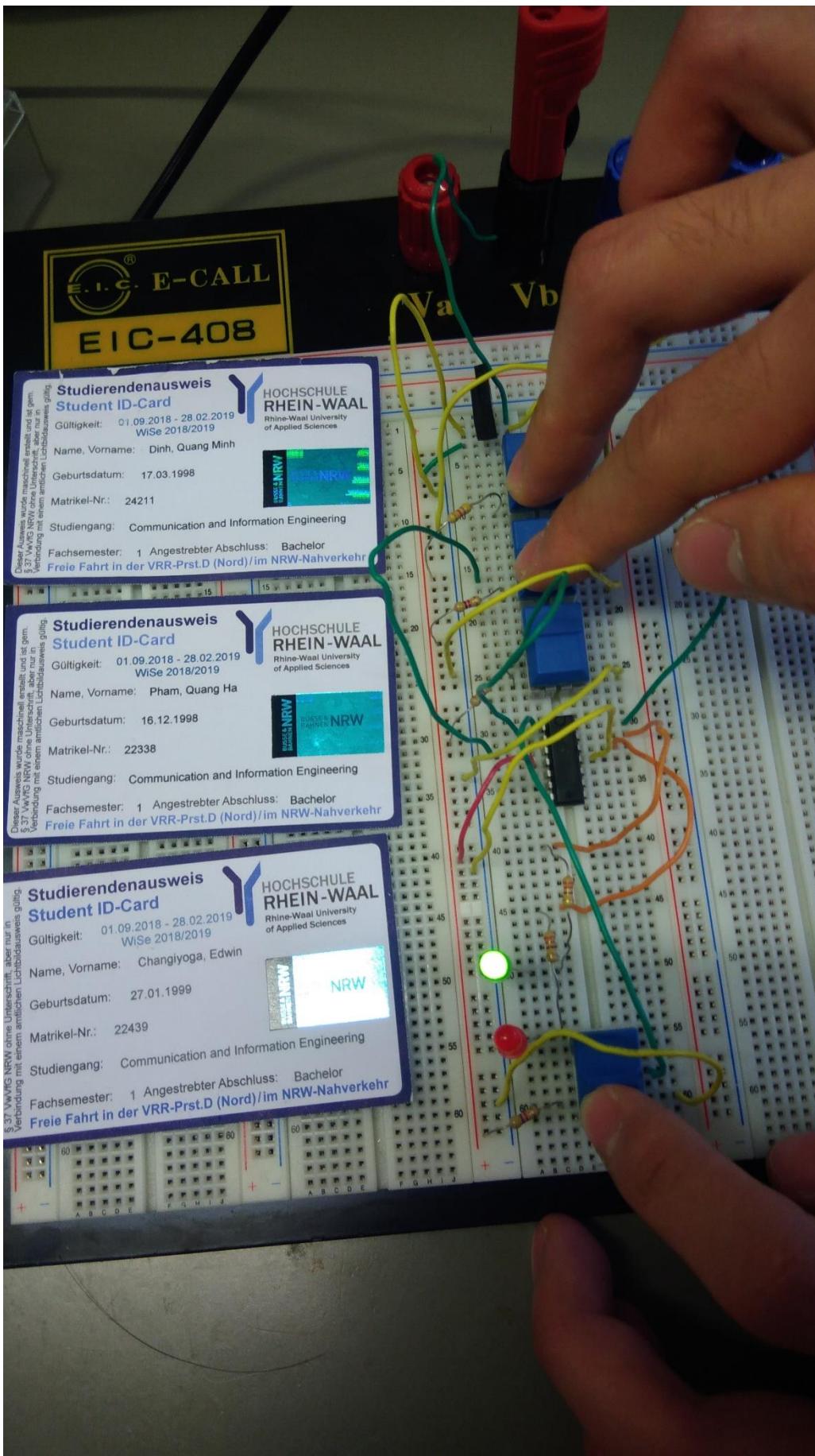












Result:

CL	J	K	CLK	Qⁿ	Qⁿ⁺¹
0	0	0	0	0	1
0	0	0	0	1	X
0	0	0	1	0	1
0	0	0	1	1	X
0	0	1	0	0	1
0	0	1	0	1	X
0	0	1	1	0	1
0	0	1	1	1	X
0	1	0	0	0	1
0	1	0	0	1	X
0	1	0	1	0	1
0	1	0	1	1	X
0	1	1	0	0	1
0	1	1	0	1	X
0	1	1	1	0	1
0	1	1	1	1	X
1	0	0	0	0	Last state
1	0	0	0	1	Last state
1	0	0	1	0	Last state
1	0	0	1	1	Last state
1	0	1	0	0	1
1	0	1	0	1	X
1	0	1	1	0	Last state
1	0	1	1	1	Last state
1	1	0	0	0	X
1	1	0	0	1	0
1	1	0	1	0	Last state
1	1	0	1	1	Last state
1	1	1	0	0	1*
1	1	1	0	1	0*
1	1	1	1	0	Last state
1	1	1	1	1	Last state