

Versuch V1

C752 Digitaltechnik

Version: .22

Stand: 30. Oktober 2023

Tom Mohr

Martin Ohmeyer

Inhaltsverzeichnis

1	Aufgabe 3	1
1.1	Aufgabe 3.1	1
2	Aufgabe 6	2
2.1	Aufgabe 6.1	2
2.2	Aufgabe 6.2	2
3	Aufgabe 7	3
3.1	Aufgabe 7.1	3
3.2	Aufgabe 7.2	3
3.3	Aufgabe 7.3	3
3.4	Aufgabe 7.4	3
4	Aufgabe 9	4
4.1	Aufgabe 9.1	4
4.2	Aufgabe 9.2	4
4.3	Aufgabe 9.3	4
4.4	Aufgabe 9.4	4
5	Aufgabe 10	5
5.1	Aufgabe 10.1	5
6	Aufgabe 11	6
6.1	Aufgabe 11.1	6
6.2	Aufgabe 11.2	7
6.3	Aufgabe 11.3	7

1 Aufgabe 3

1.1 Aufgabe 3.1

2 Aufgabe 6

2.1 Aufgabe 6.1

2.2 Aufgabe 6.2

3 Aufgabe 7

3.1 Aufgabe 7.1

3.2 Aufgabe 7.2

3.3 Aufgabe 7.3

3.4 Aufgabe 7.4

4 Aufgabe 9

4.1 Aufgabe 9.1

4.2 Aufgabe 9.2

4.3 Aufgabe 9.3

4.4 Aufgabe 9.4

5 Aufgabe 10

5.1 Aufgabe 10.1

```
#include <b15f/b15f.h>

int main()
{
    int mode = 0;
    B15F &drv = B15F::getInstance();

    while (true)
    {
        mode = drv.readDipSwitch();
        if (mode == 0)
        {
            drv.digitalWrite0(255 - (int)drv.digitalRead0());
        }
        else if (mode == 1)
        {
            for (int output = 64; output > 1; output /= 2)
            {
                drv.digitalWrite0(output);
                drv.delay_ms(150);
            }

            for (int output = 1; output <= 128; output *= 2)
            {
                drv.digitalWrite0(output);
                drv.delay_ms(150);
            }
        }
    }
}
```

6 Aufgabe 11

6.1 Aufgabe 11.1

6.2 Aufgabe 11.2

```
#include <iostream>
#include <b15f/b15f.h>

using namespace std;

string toBinary(int n)
{
    string r;
    while(n!=0) {r=(n%2==0 ?"0":"1")+r; n/=2;}
    return r;
}

int main()
{
    B15F &drv = B15F::getInstance();
    int max = 0;
    cout << "Max. Byte: ";
    cin >> max;

    for(int i = 0; i < max; i++)
    {
        drv.digitalWrite0(i);
        int input = (int)drv.digitalRead0();

        string bin = toBinary(i);
        while(bin.length() < (log(max) / log(2)))
        {
            bin = bin.insert(0, "0");
        }
        cout << bin << "\t" << input << endl;
    }

    return 0;
}
```

6.3 Aufgabe 11.3