

Aufgabenstellung:

Aufgabe war es, in 2er-Teams eine Platine zur Floppy-Disk-Motorsteuerung inklusive Programmierung zu erstellen.

Das Programm sollte folgende Schritte ausführen:

- Nach dem Einschalten sollte die Arbeits-LED am Floppy-Laufwerk blinken, bis eine Floppy-Disk eingelegt wurde (über Lichtschranke prüfbar)
- Anschließend sollte die LED dauerhaft leuchten und der Floppy-Motor sollte sich bewegen.
- Wenn nach einer bestimmten Zeit das Index-Loch der Floppy nicht gefunden wurde, sollte die LED schnell blinken.
- Wenn ein Index-Loch gefunden worden war, sollte der Motor stoppen und der Lesekopf sollte in Startposition fahren.
- Wenn der Lesekopf in die Startposition gefahren war, sollte er sich dauerhaft 20 Schritte nach links und 20 Schritte nach rechts bewegen, andernfalls sollte die LED noch schneller blinken, als in den Fehlerfällen davor.

Zudem sollte die Platine so angefertigt werden, sodass ein ?????-Stecker vom DIL und ein ???-Stecker zur Verbindung mit dem Floppy-Laufwerk angeschlossen werden konnten.

Lösung der Aufgabe:

Nachdem wir uns in 2er Gruppen aufgeteilt hatten, wurden die Aufgaben aufgeteilt.

Es wurde zuerst ein Programm geschrieben, welches Die LED und den Schrittmotor ansteuert, um diese Funktionen in Verbindung mit der Platine testen zu können:

```
//Floppy.c
#include "std_stm32_f103rb.h"

#define INTERVALL_SCHRITTMOTOR 100

int main() {
    init_leds_switches();
    do {
        set_leds(0x15);           //0x15: 1-> LED = EIN, 5-> Schrittmotor Bitstellung 0101
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x16);           //0x16: 1-> LED = EIN, 6-> Schrittmotor Bitstellung 0110
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x1A);           //0x1A: 1-> LED = EIN, A-> Schrittmotor Bitstellung 1010
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x19);           //0x19: 1-> LED = EIN, 9-> Schrittmotor Bitstellung 1001
        wait_ms(INTERVALL_SCHRITTMOTOR);

        set_leds(0x05);           //0x05: 0-> LED = EIN, 5-> Schrittmotor Bitstellung 0101
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x06);           //0x06: 0-> LED = EIN, 6-> Schrittmotor Bitstellung 0110
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x0A);           //0x0A: 0-> LED = EIN, A-> Schrittmotor Bitstellung 1010
        wait_ms(INTERVALL_SCHRITTMOTOR);
        set_leds(0x09);           //0x09: 0-> LED = EIN, 9-> Schrittmotor Bitstellung 1001
        wait_ms(INTERVALL_SCHRITTMOTOR);
    } while(1);
    return 0;
}
```