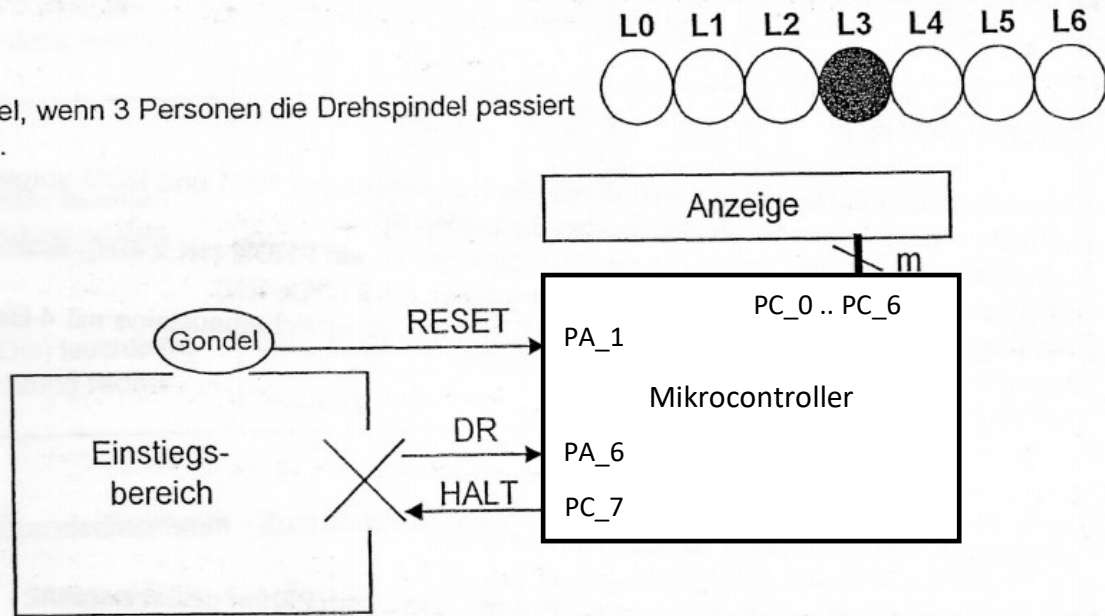


## Skigondel

Eine Skipiste ist über eine Gondelbahn zu erreichen. Jede Gondel hat ein Fassungsvermögen von maximal 6 Personen. Der Einstiegsbereich einer Gondel ist mit einer Drehschindel, die die Anzahl der zusteigenden Personen zählt, abgegrenzt. Über eine Lauflichtanzeige (siehe Abbildung 1) wird angezeigt, wie viele Personen bereits die Drehschindel zum Eingangsbereich passiert haben. Alle Personen (maximal 6), die sich im Einstiegsbereich zur Gondel befinden, müssen in die Gondel einsteigen, so dass nach dem Start der Gondel der Einstiegsbereich wieder leer und die Anzeige L0 leuchtet.

Abb1:

Beispiel, wenn 3 Personen die Drehschindel passiert haben.



Ist der Einstiegsbereich mit 6 Personen voll besetzt, soll die Drehschindel mit dem Signal  $\text{HALT} = 1$  blockieren, so dass keine weiteren Personen in den Einstiegsbereich gelangen können. Mit ablegen der Gondel, sendet diese ein Resetsignal ( $\text{RESET} = 1$ ) und die Sperre der Drehschindel wird wieder aufgehoben.

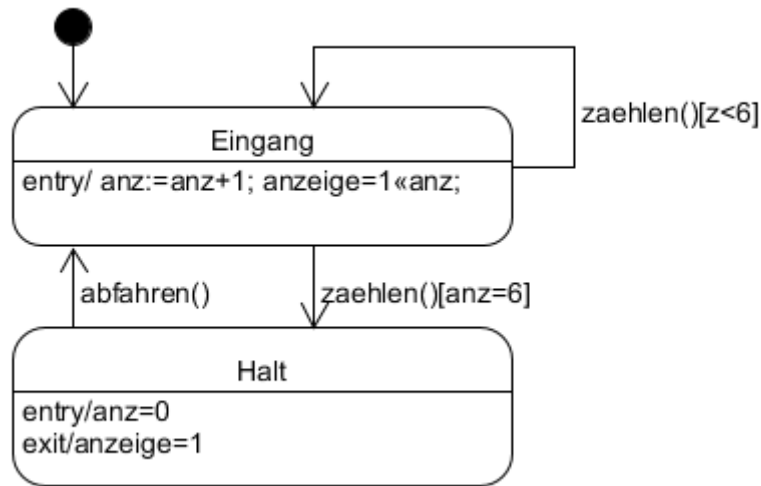
Die Steuerung der Drehschindel und der Anzeige, sowie das Zählen der Fahrgäste werden durch den Mikrocontroller. Realisiert. RESET und DR lösen beim Mikrocontroller Interrupts aus.

Fragen und Aufgaben:

1. In welchen 2 Zuständen kann sich der Mikrocontroller befinden?
2. Zeichnen Sie das Zustandsdiagramm
3. Schreiben Sie das Programm in C/CPP

Lösungen:

- ## 1. Die Zustände: Eingang und Halt



- 2.
3. Programm

```
#include "mbed.h"
```

```
PortOut zustand(PortC,0x80);
PortOut anzeige(PortC,0x7F);
#define Eingang 0x00
#define Halt 0x80
```

```
InterruptIn res(PA_1);
InterruptIn DR(PA_6);
```

```
int z=0;
```

```
void zaehlen()  
{  
    switch (zustand)  
    {  
        case Eingang: z++;  
            anzeige=1<<z;  
            if (z==6)  
            {  
                zustand=Halt;  
                z=0;  
            }  
            break;  
    }  
}
```

```
void abfahren()
{
    switch (zustand)
    {
        case Halt:
            anzeige=1;
            zustand=Eingang;
            break;
    }
}
```

```
int main()
{
    DR.mode(PullDown);
    DR.rise(&zaehlen);
    res.mode(PullDown);
    res.rise(&abfahren);
    zustand=Eingang;
    anzeige=1;
    while(true)
    {

    }
}
```