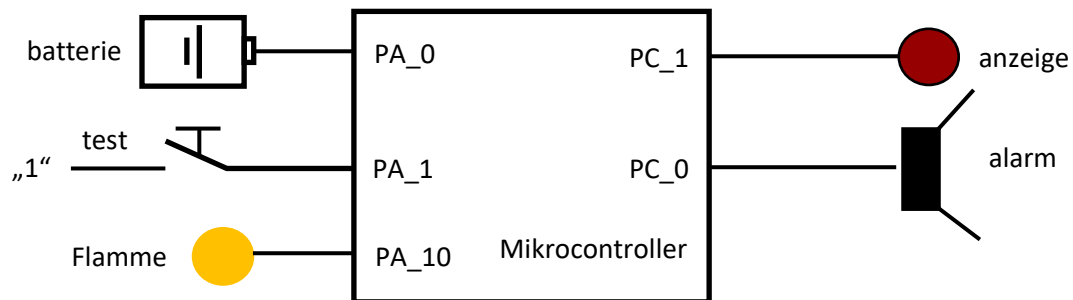
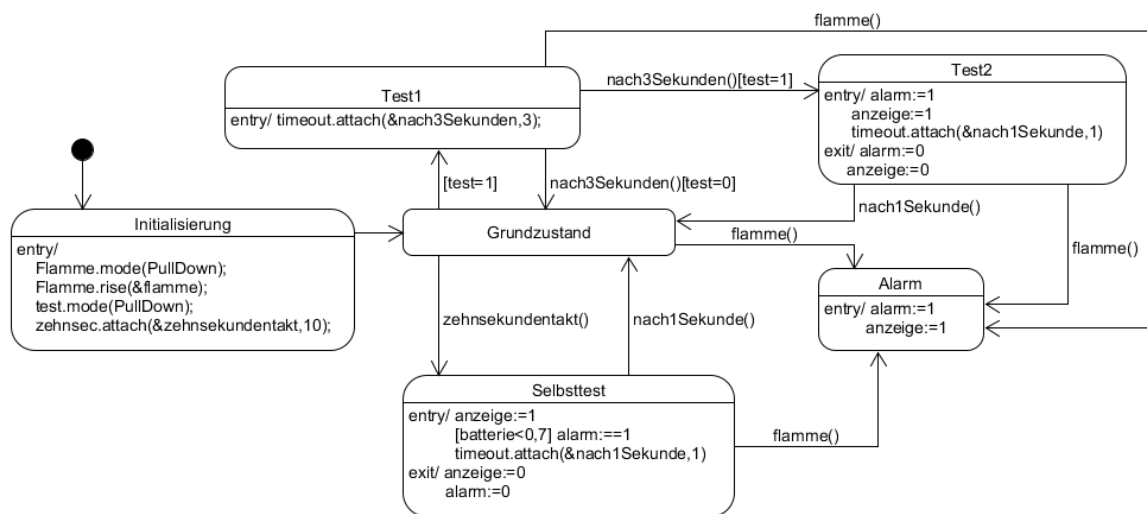


# Brandmelder

## Technologieschema:



## Zustandsdiagramm:



## Beschreibung:

### Definition der Zustände:

Grundzustand	0
Test1	1
Test2	2
Selbsttest	3
Alarm	4

### Deklaration der Ein- und Ausgänge:

PC\_0 als Digitalausgang alarm  
 PC\_1 als Digitalausgang anzeige  
 PA\_10 als Interrupteingang Flamme (Sensor wird durch Taster simuliert)  
 PA\_0 als Analogeingang batterie  
 PA\_1 als Digitaleingang test

### Globale Deklarationen:

zustand:int=Grundzustand  
 Timeout als Timeout  
 zehnsec als Ticker

Operationen: Hinweis: Alle Operationen, sowie die Endlosschleife beginnen mit switch(zustand){

- void nach1Sekunde(): Timeout-Callback (ISR) bewirkt Zustandswechsel von Selbsttest bzw. Test2 in den Grundzustand, führt die exit-Aktivität der Ausgangszustände aus.
- void nach3Sekunden(): Timeout-Callback (ISR) bewirkt Zustandswechsel, in Abhängigkeit von test-Taste, von Test1 in den Grundzustand oder in Test2. Führt bei Zustandswechsel in Test2 die entry-Aktivität von Test2 aus.
- void zehntsekundentakt(): Ticker-Callback (ISR) bewirkt Zustandswechsel von Grundzustand zu Selbsttest. Führt die entry-Aktivität von Selbsttest aus.
- void flamme(): Callback (ISR) von Interrupteingang Flamme. Bewirkt, unabhängig vom Ausgangszustand einen Zustandswechsel in den Zustand Alarm. Führt die entry-Aktivität dieses Zustands aus.

Aufgaben:

1. Beschreiben Sie in eigenen Worten die Funktion des Timeouts
2. Beschreiben Sie in eigenen Worten die Funktion des Tickers
3. Berechnen Sie: Ab Unterschreitung welcher Batteriespannung geht alarm für 1 Sekunde auf  
1. Hinweis: Die Referenzspannung des AD-Wandlers beträgt 3,3V
4. Erklären Sie, wie sich die Eingänge: test und batterie unterscheiden
5. Schreiben Sie das Mikrocontrollerprogramm und testen Sie es

Lösung:

```
DigitalOut alarm(PC_0);  
DigitalOut anzeige(PC_1);  
InterruptIn Flamme(PA_10);  
AnalogIn batterie(PA_0);  
DigitalIn test(PA_1);
```

```
#define Grundzustand 0  
#define Test1 1  
#define Test2 2  
#define Selbsttest 3  
#define Alarm 4
```

```
int zustand=Grundzustand;
```

```
Timeout timeout;  
Ticker zehnsec;
```

```
void nach1Sekunde()  
{  
    switch(zustand)  
    {  
        case Selbsttest:  
            anzeige=0;  
            alarm=0;  
            zustand=Grundzustand;  
            break;  
        case Test2:  
            alarm=0;  
            anzeige=0;  
            zustand=Grundzustand;  
    }  
}
```

```
void zehnsekudentakt()  
{  
    switch (zustand)  
    {  
        case Grundzustand:  
            zustand=Selbsttest;  
            anzeige=1;  
            if (batterie<0.7) alarm=1;  
            timeout.attach(&nach1Sekunde,1);  
            break;  
    }  
}
```

```
void nach3Sekunden()  
{  
    switch (zustand)
```

```

{
    case Test1:
        if (test==1)
        {
            zustand=Test2;
            alarm=1;
            anzeige=1;
            timeout.attach(&nach1Sekunde,1);
        }
        else
        {
            zustand=Grundzustand;
        }
        break;
    }
}

void flamme()
{
    zustand=Alarm;
    alarm=1;
    anzeige=1;
}

int main()
{
    Flamme.mode(PullDown);
    Flamme.rise(&flamme);
    test.mode(PullDown);
    zehnsec.attach(&zehnsekudentakt,10);
    while(true)
    {
        switch(zustand)
        {
            case Grundzustand:
                if (test==1)
                {
                    zustand=Test1;
                    timeout.attach(&nach3Sekunden,3);
                }
                break;
        }
    }
}

```