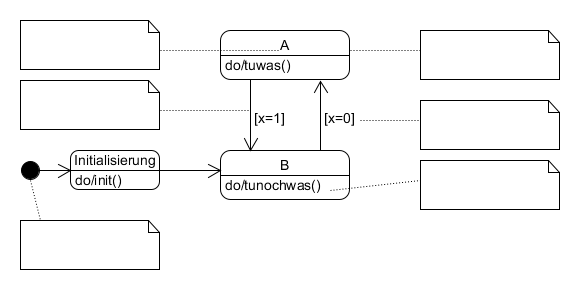
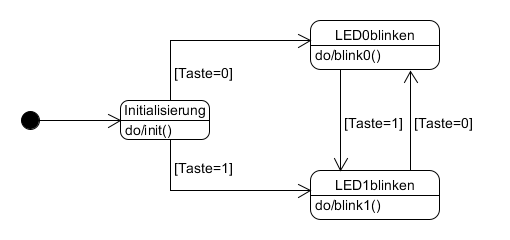
UML-Zustandsdiagramm auf dem Mikrocontroller

1. Zeichnen Sie ein Zustandsdiagramm mit folgenden Anforderungen:
   1. Nach dem Startzustand wechselt das Programm in den Zustand Initialisierung
   2. Im Zustand Initialisierung wird das Unterprogramm init() aufgerufen
   3. Wenn Taste=1 soll der Mikrocontroller led0 blinken (blink0())
   4. Wenn Taste=0 soll der Mikrocontroller led1 blinken (blink1())
2. Beschriften Sie das Zustandsdiagramm
3. Schreiben Sie den C-Code zum Zustandsdiagramm aus Aufgabe 2.
4. Eine Effektbeleuchtung kann: Blinken, Lauflicht, Muster ausgeben. Die Auswahl erfolgt mit dem Analogeingang „Sensor“. Wenn Sensor < 0,3 dann blinkt die Anzeige, bei 0,3 .. 0,7 wird das Lauflicht angezeigt. Bei Sensor>0,7 erfolgt die Ausgabe der Anzeigemuster.
   1. Ermitteln Sie die Zustände mit den zugehörigen Aktivitäten
   2. Ergänzen Sie die Transitionen mit den erforderlichen Wächterbedingungen
   3. Schreiben Sie das Programm

Lösungen:



// Blinking rate in milliseconds

#define BLINKING\_RATE\_MS 500

DigitalIn Taste(PA\_1);

DigitalOut led0(PC\_0);

DigitalOut led1(PC\_1);

void init()

{

Taste.mode(PullDown);

}

void blink0()

{

led0=!led0;

}

void blink1()

{

led1=!led1;

}

int main()

{

init();

while (true) {

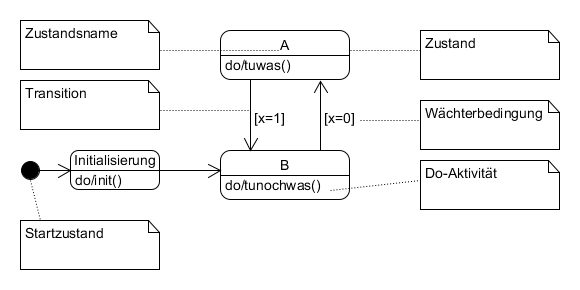
if (Taste==0) blink0();

else blink1();

thread\_sleep\_for(BLINKING\_RATE\_MS);

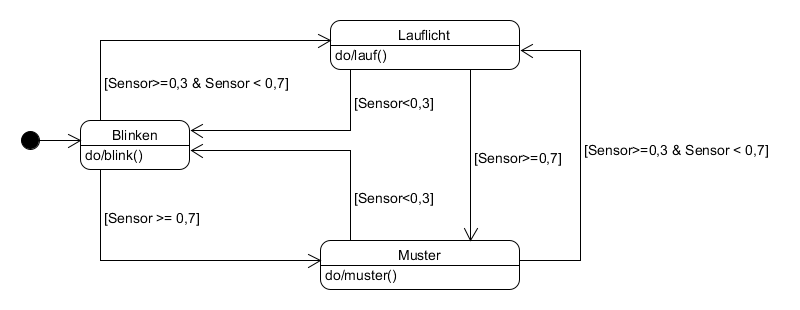
}

}

1. 
2. C-Code:  
   int main(void)  
   { init();  
    while(true){

If (x==1) tunochwas();  
else tuwas();

}}

1. 

#define BLINKING\_RATE\_MS 500

PortOut anzeige(PortC,0xFF);

AnalogIn Sensor(PA\_0);

int z=0;

int m[6]={0b10000001,0b01000010,0b00100100,0b00011000,0b00100100,0b01000010};

void blink()

{ anzeige=~anzeige;

}

void lauf()

{ anzeige=anzeige<<1;

if (anzeige==0) anzeige=1;

}

void muster()

{ z=(z+1)%6;

anzeige=m[z];

}

int main()

{ while (true) {

if (Sensor<0.3) blink();

if (Sensor>=0.3 && Sensor<0.7) lauf();

if (Sensor>=0.7) muster();

thread\_sleep\_for(BLINKING\_RATE\_MS);

}

}