

Digitale Verarbeitung analoger Signale

Was tun mit den Daten?



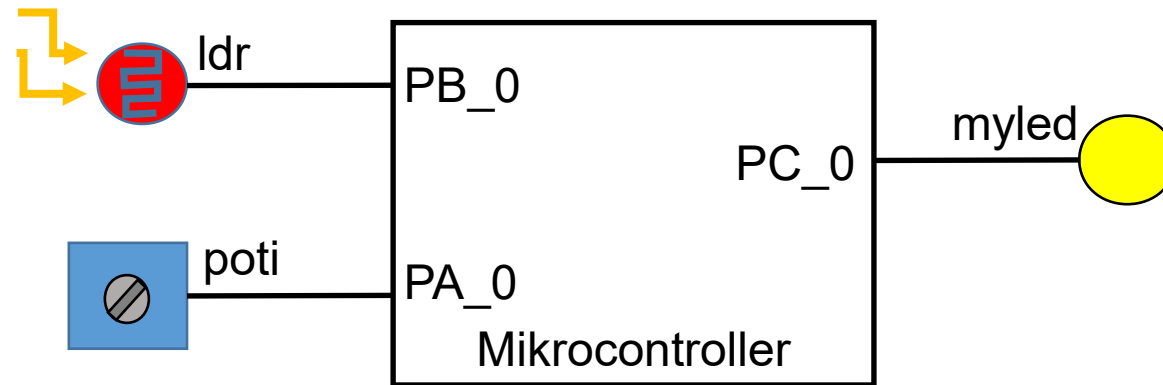
Hallo, ich bin Mik, Dein
Mikrocontroller



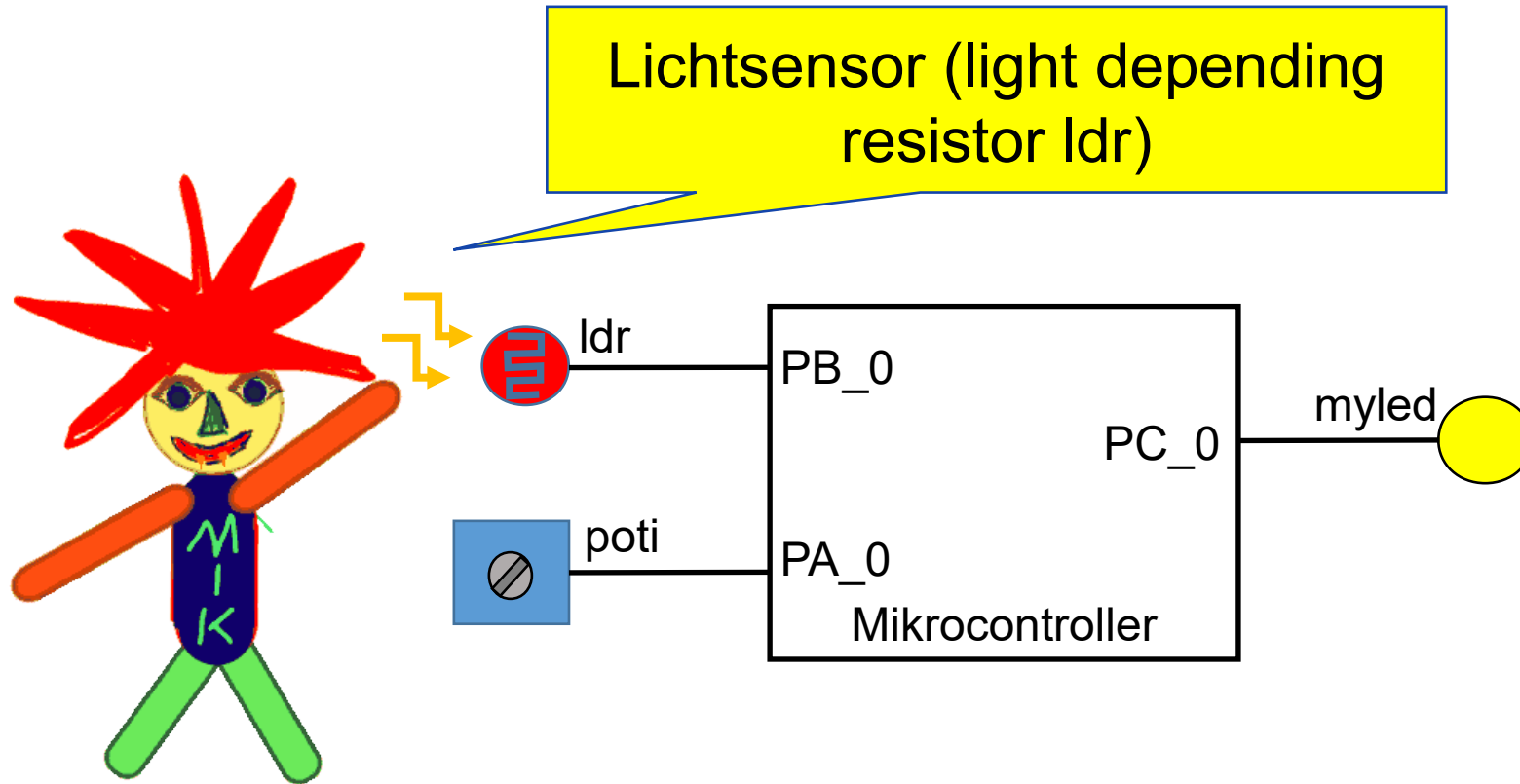
Digitale Verarbeitung analoger Signale



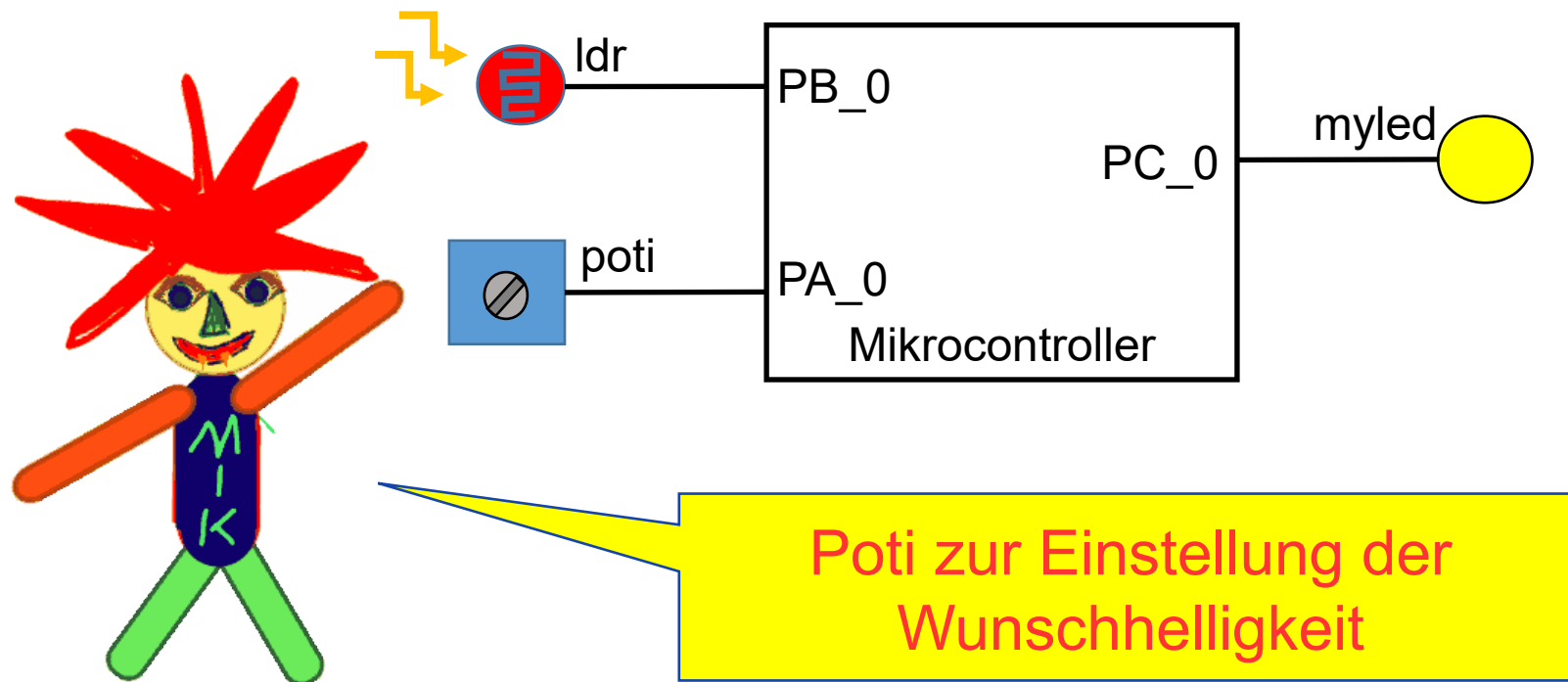
Ein Dämmerungsschalter soll
gebaut werden



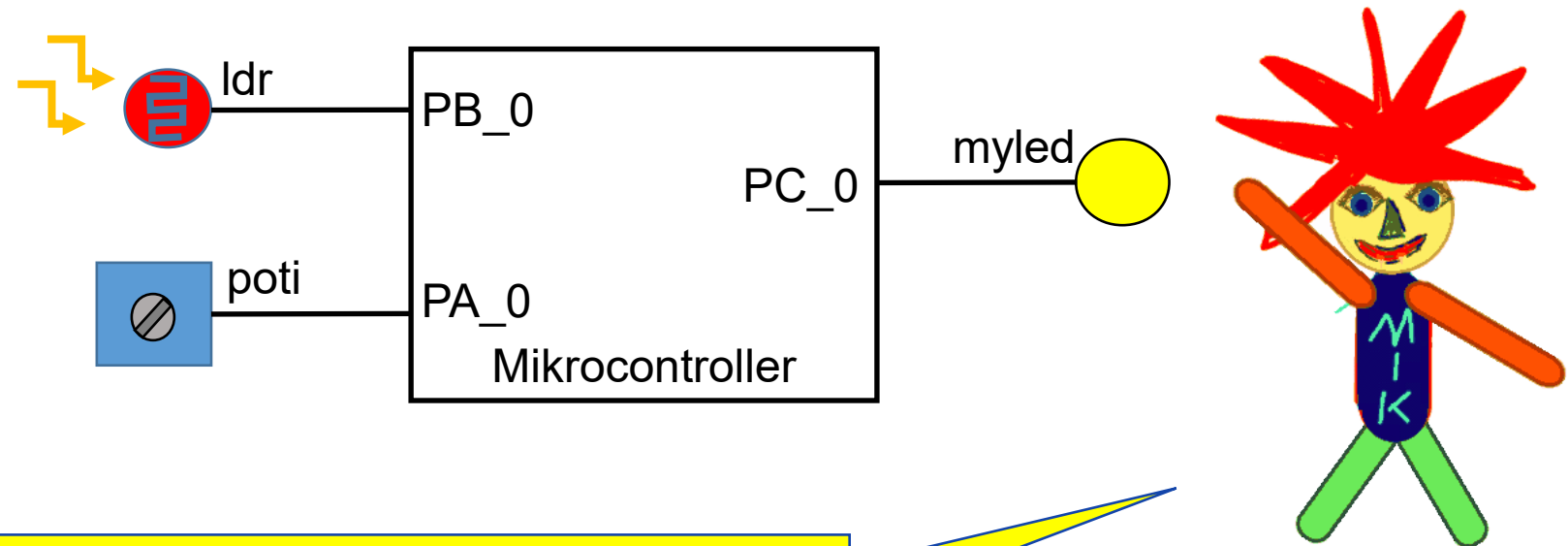
Digitale Verarbeitung analoger Signale



Digitale Verarbeitung analoger Signale



Digitale Verarbeitung analoger Signale



Wenn es dunkel wird soll die
Lampe angehen

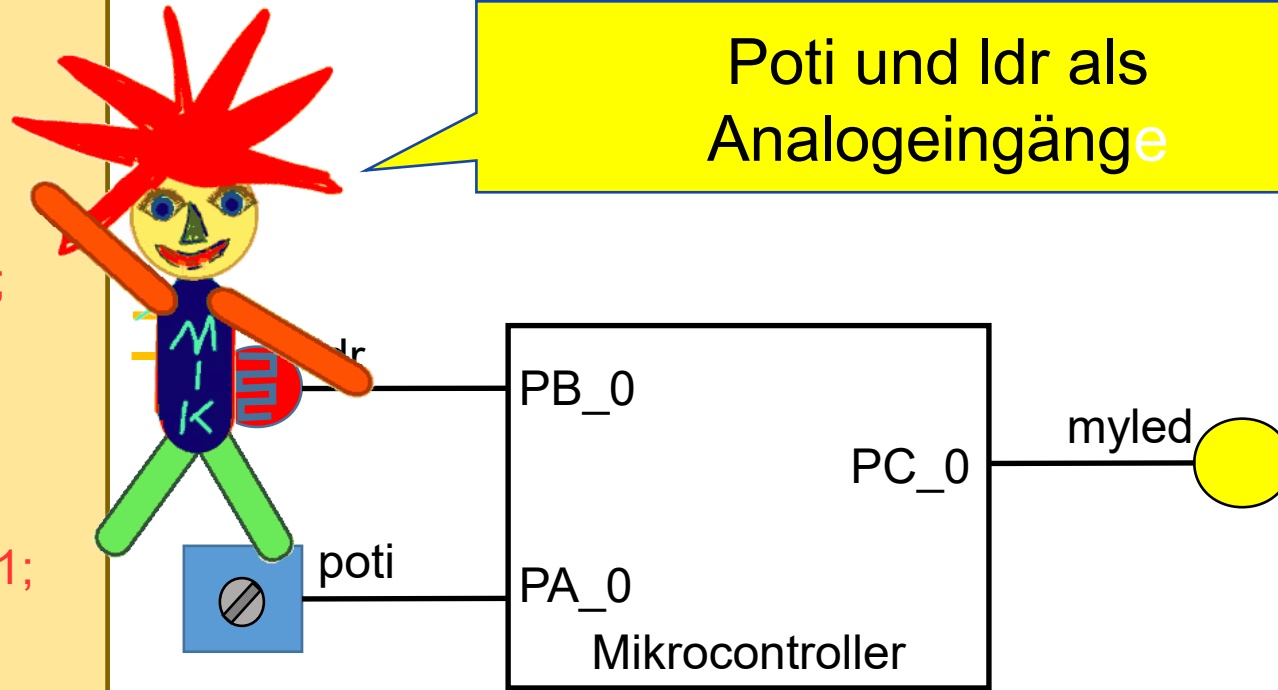


Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti<ldr) myled=1;  
        else myled=0;  
    }  
}
```

Poti und ldr als
Analogeingänge



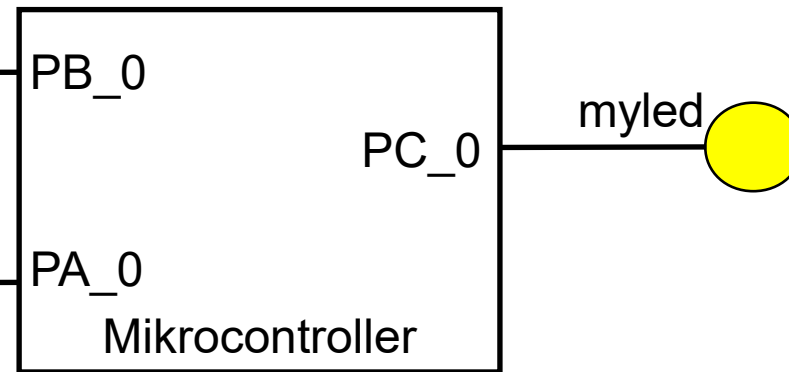
Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```



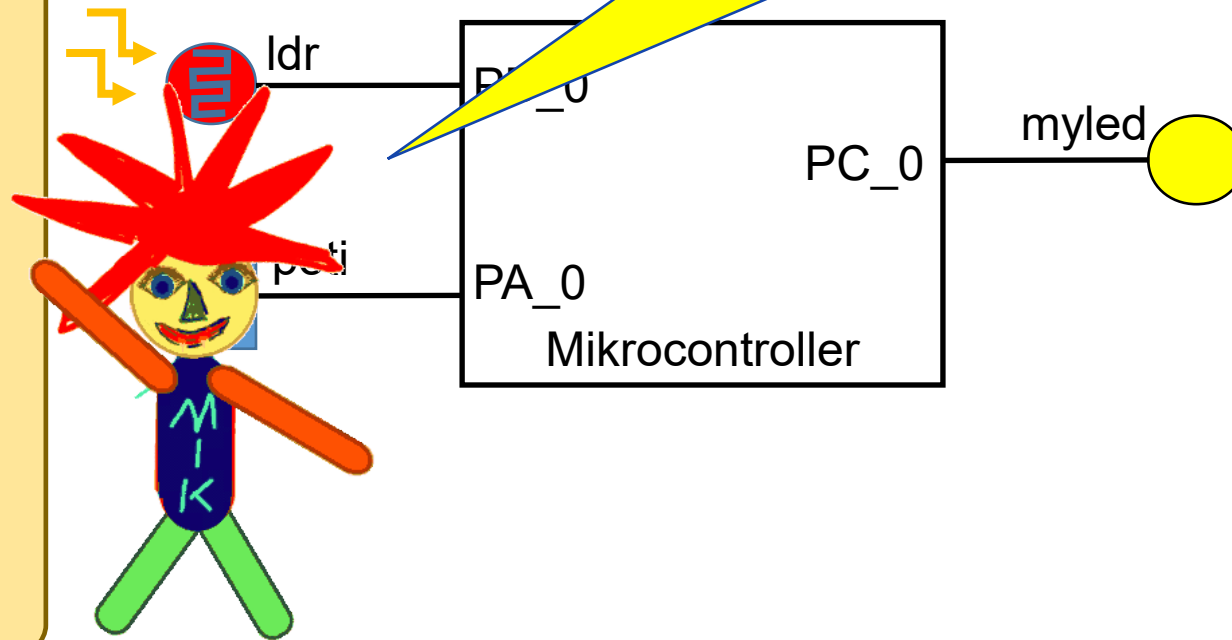
Beleuchtung myled als
Digitalausgang



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

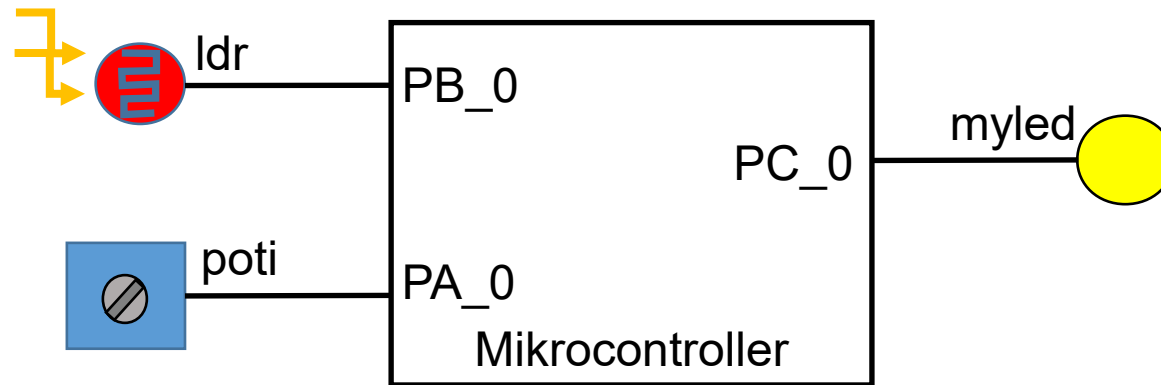
```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled = 1;  
        else myled = 0;  
    }  
}
```



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```



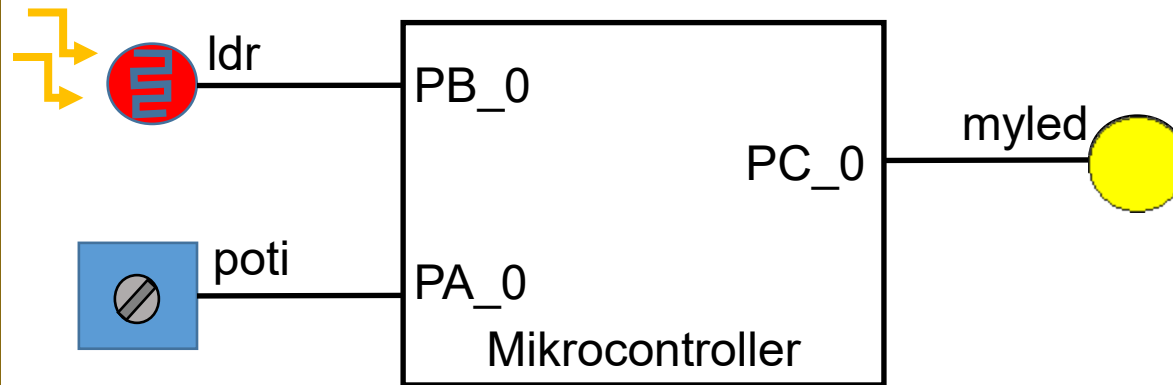
Funktioniert.
Aber...



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled = 1;  
        else myled = 0;  
    }  
}
```



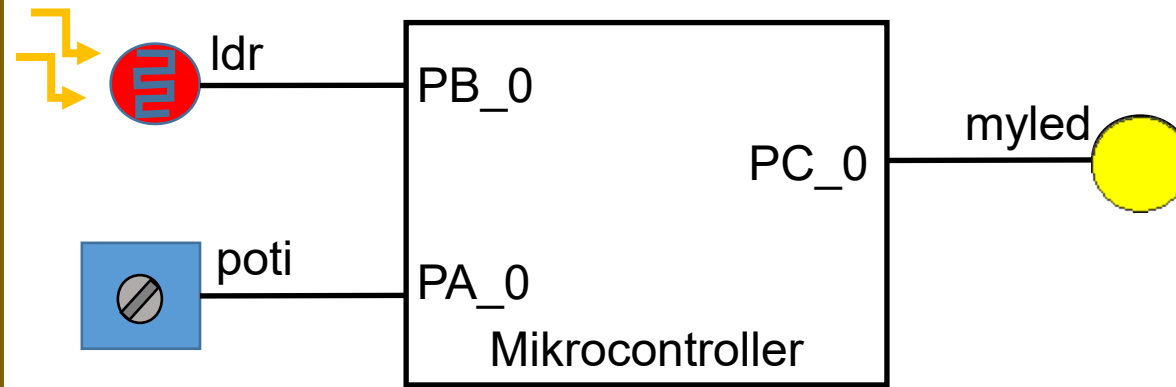
Aber...
Bei Dämmerung flackert
die Lampe



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```



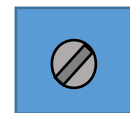
Aber...
Bei Dämmerung flackert
die Lampe



Digitale Verarbeitung analoger Signale

Aber...
Bei Dämmerung
flackert die Lampe:
myled wechselt
ständig zwischen 0
und 1

```
while (true) {  
    if (poti < ldr) myled = 1;  
    else myled = 0;  
}
```



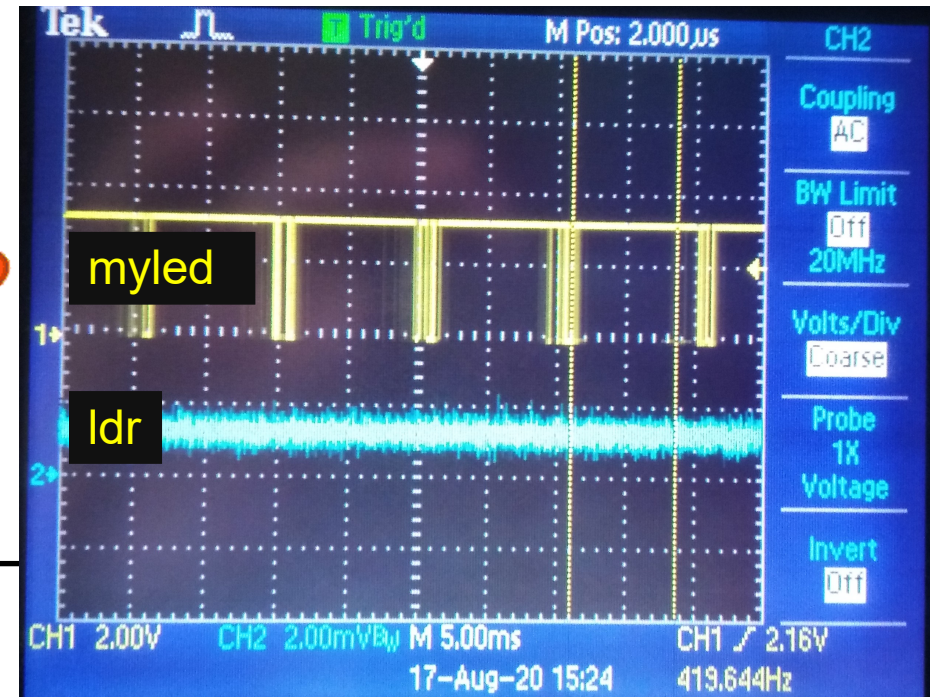
poti

PA_0

Mikrocontroller

ldr

PB_0



Digitale Verarbeitung analoger Signale

**Bei Dämmerung
schwankt das ldr-
Signal um das poti-
Signal**

An
An
Dig

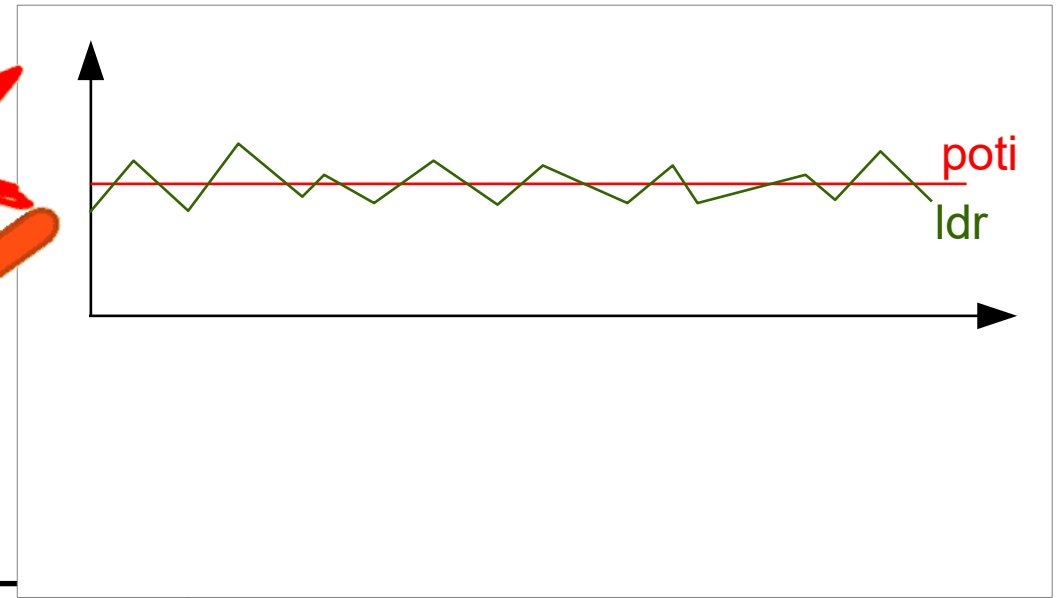
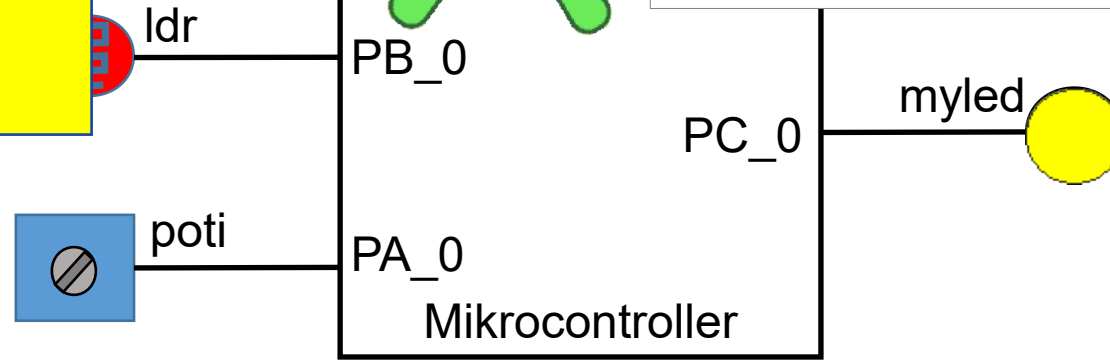
int

{

```
while (true) {  
    if (poti < ldr) myled=1;  
    else myled=0;  
}
```

}

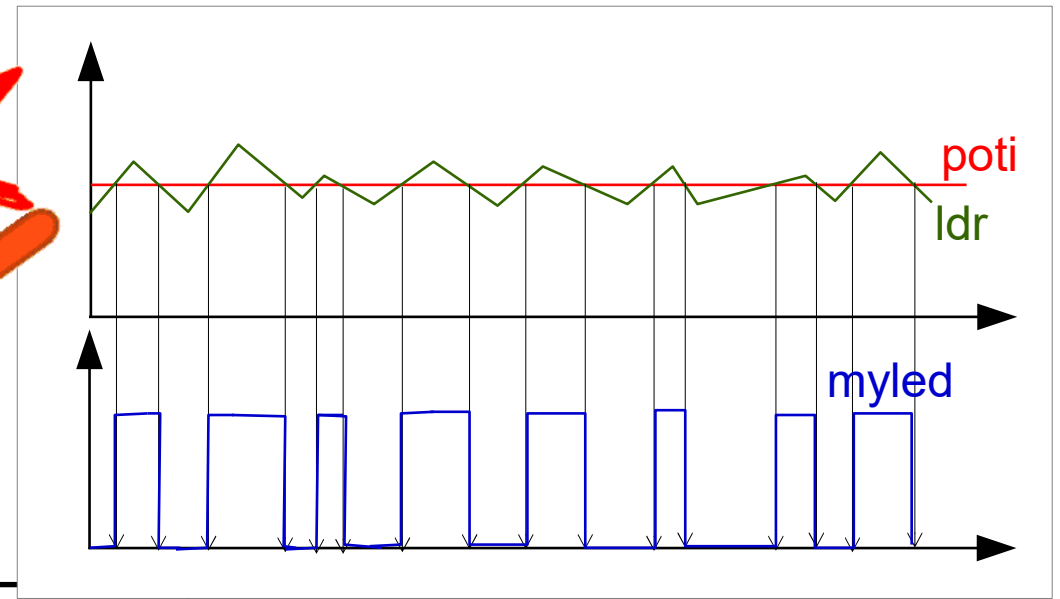
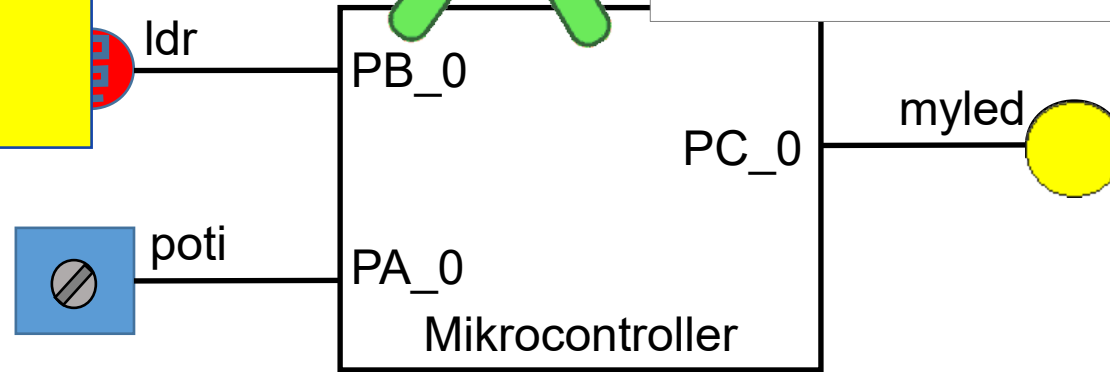
}



Digitale Verarbeitung analoger Signale

Immer wenn $ldr > poti$
wird myled
eingeschaltet, sonst
ausgeschaltet

```
while (true) {  
    if (poti < ldr) myled=1;  
    else myled=0;  
}
```

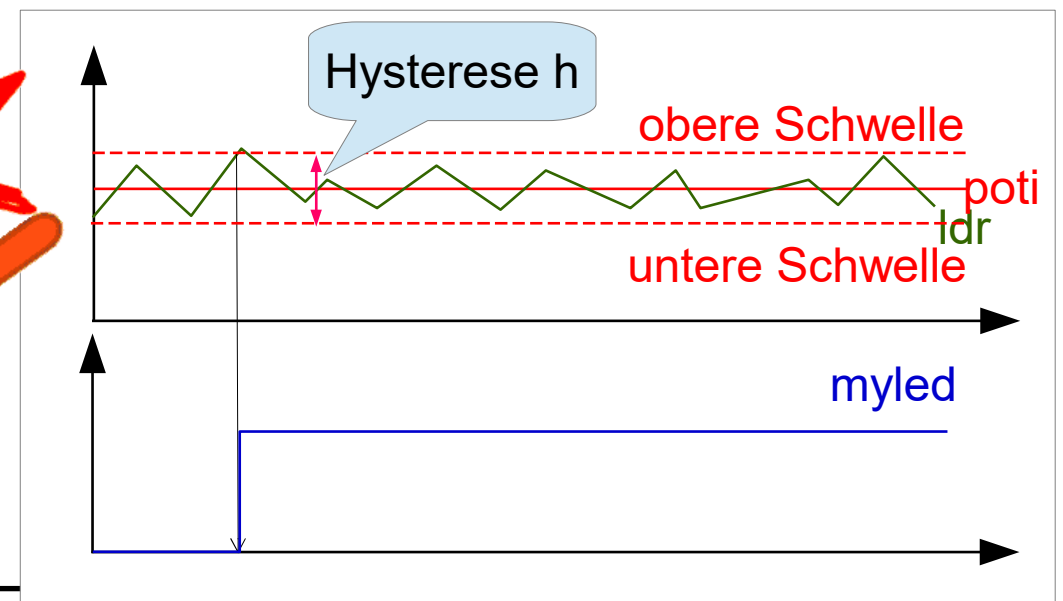
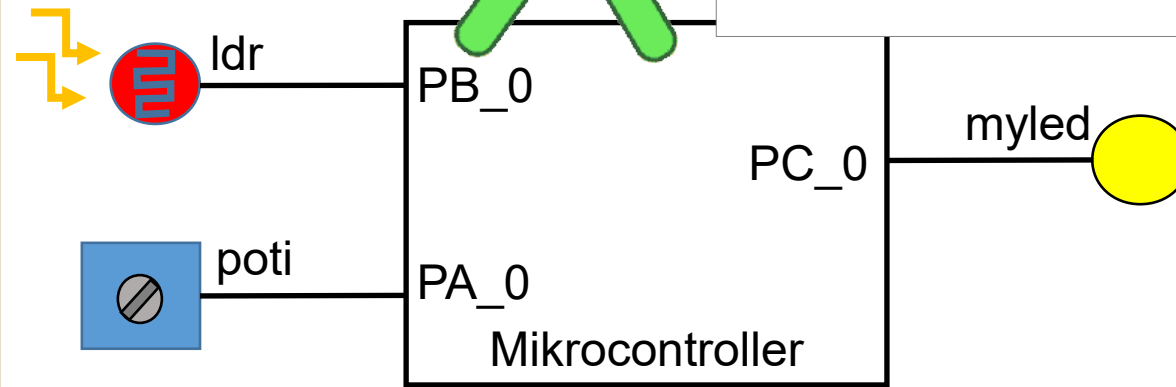


Digitale Verarbeitung analoger Signale

1. Lösungsmöglichkeit: Hysterese einbauen

```
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti<ldr) myled=1;  
        else myled=0;  
    }  
}
```

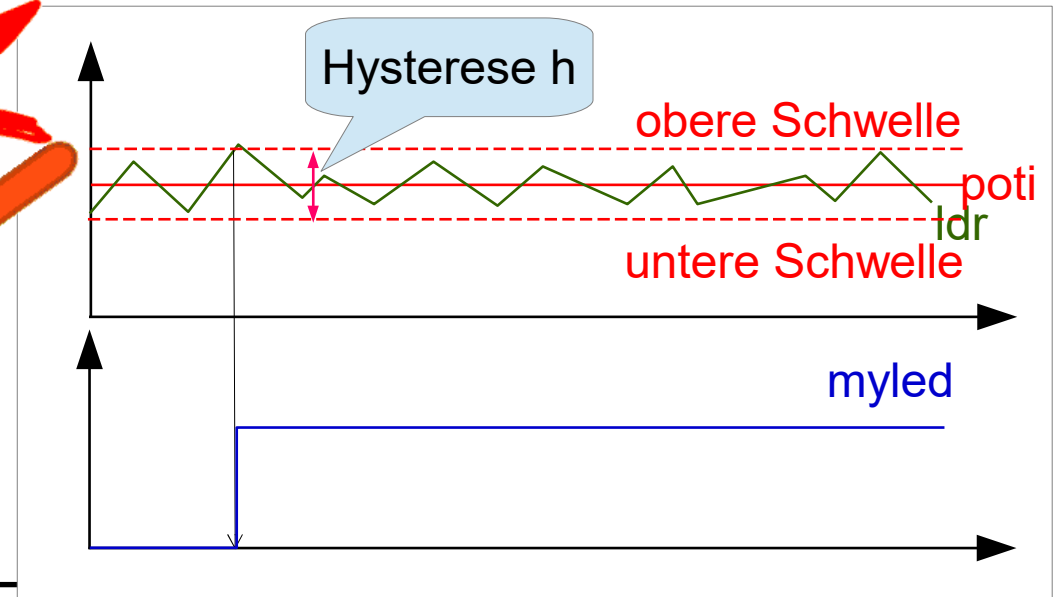
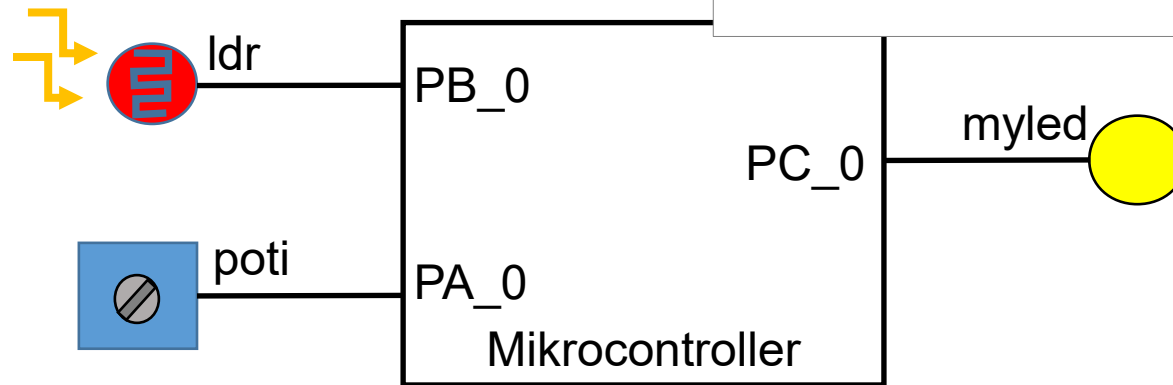


Digitale Verarbeitung analoger Signale

Obere Schwelle:
 $\text{poti} + h/2$

```
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```

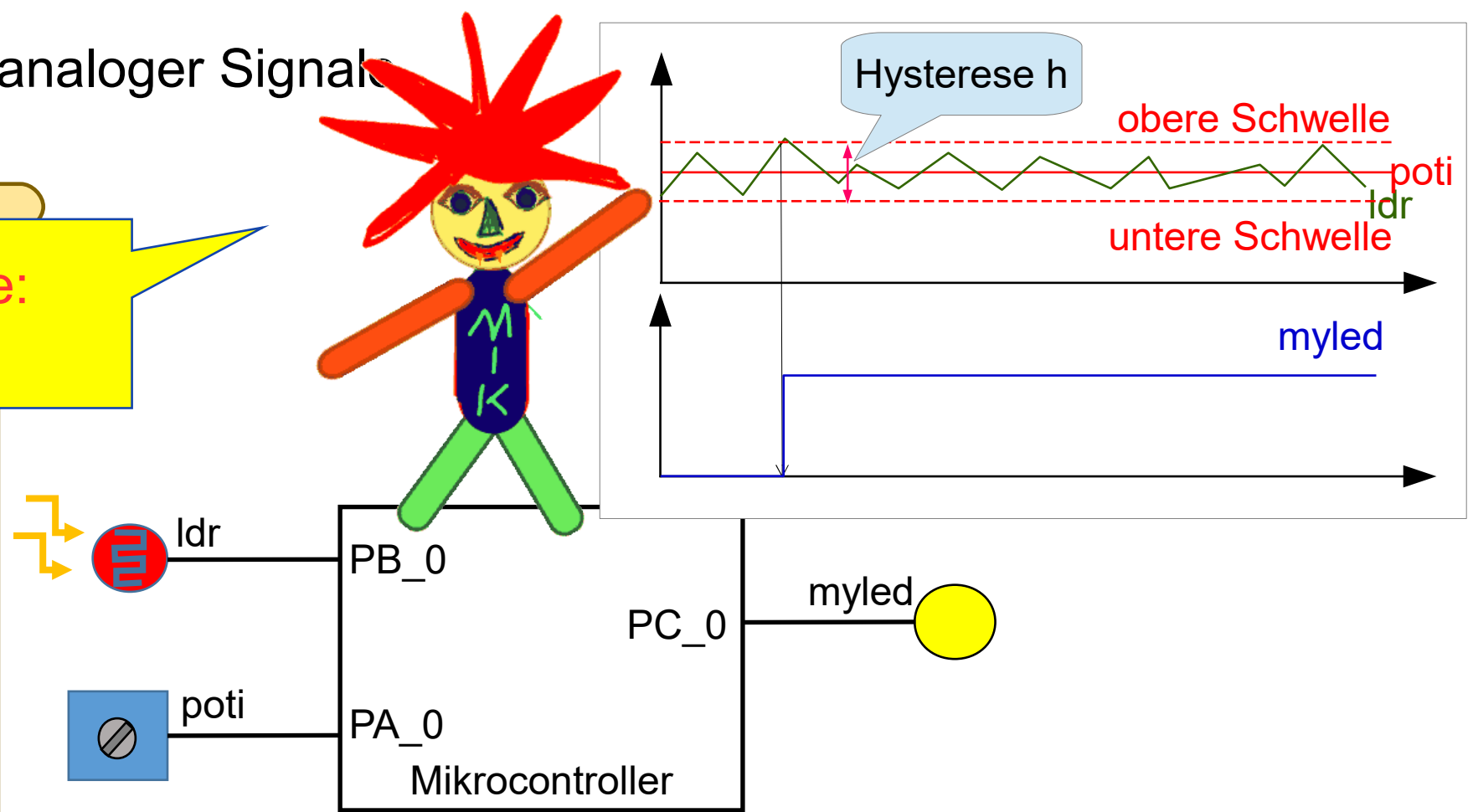


Digitale Verarbeitung analoger Signale

Untere Schwelle:
 $\text{poti} - h/2$

```
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```

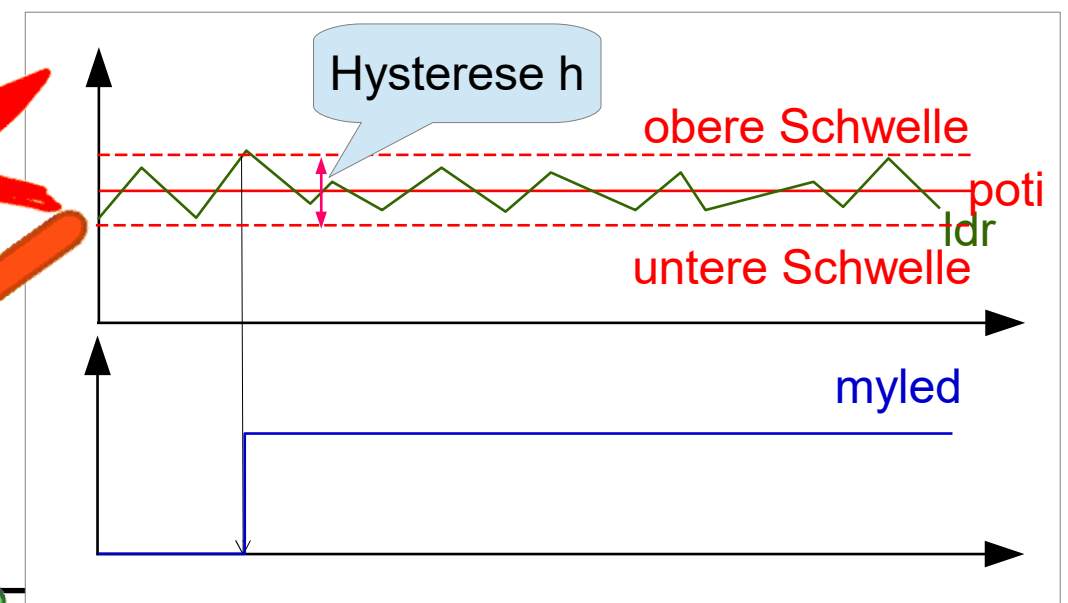
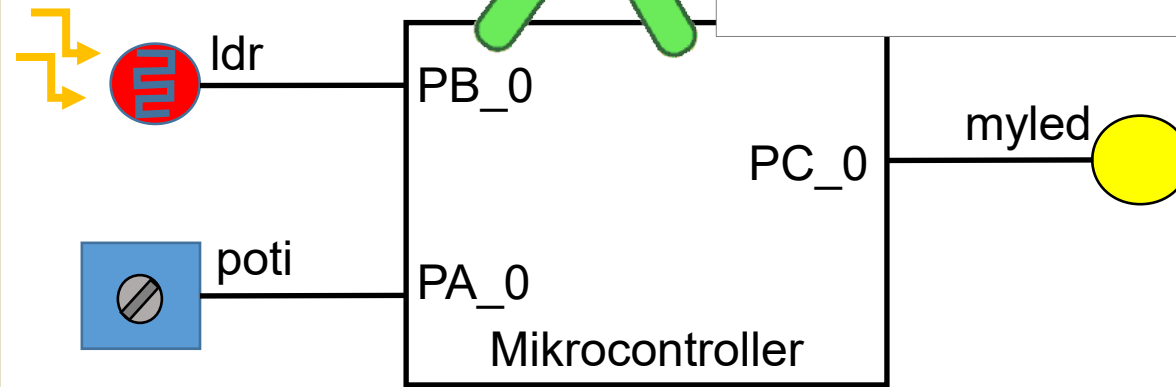


Digitale Verarbeitung analoger Signale

Einschalten wenn
 $ldr > poti + h/2$

```
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```

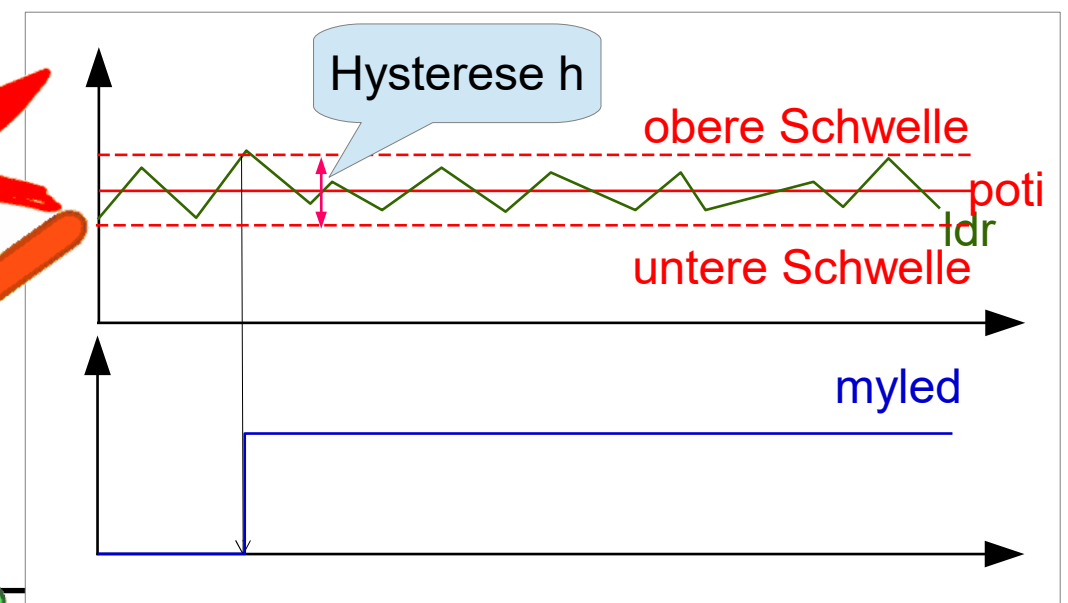
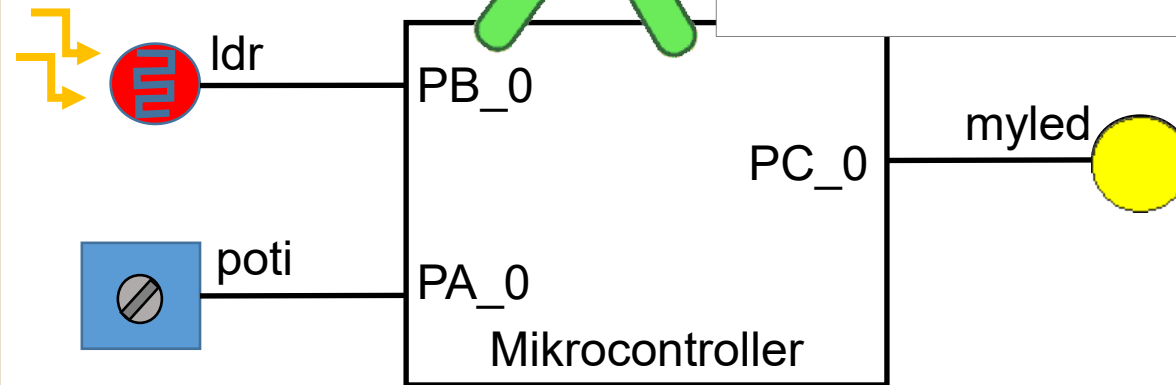


Digitale Verarbeitung analoger Signale

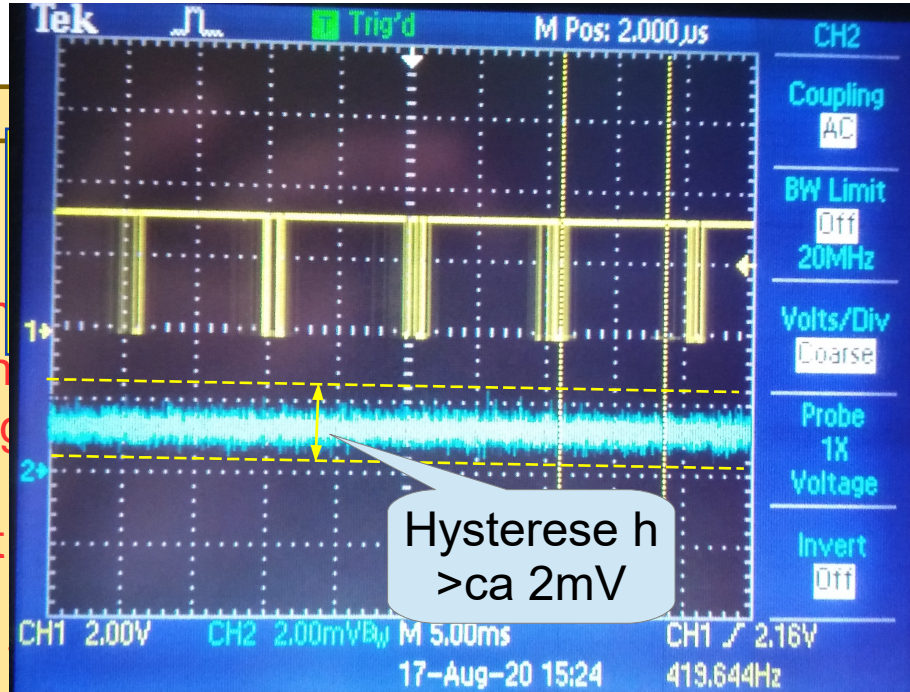
Ausschalten wenn
 $ldr < poti - h/2$

```
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

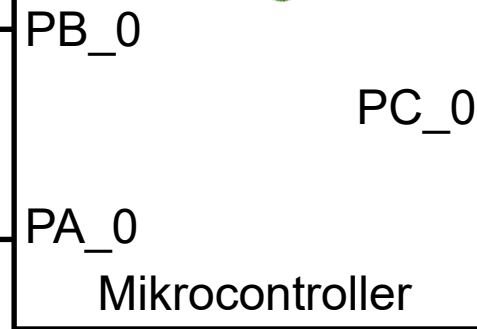
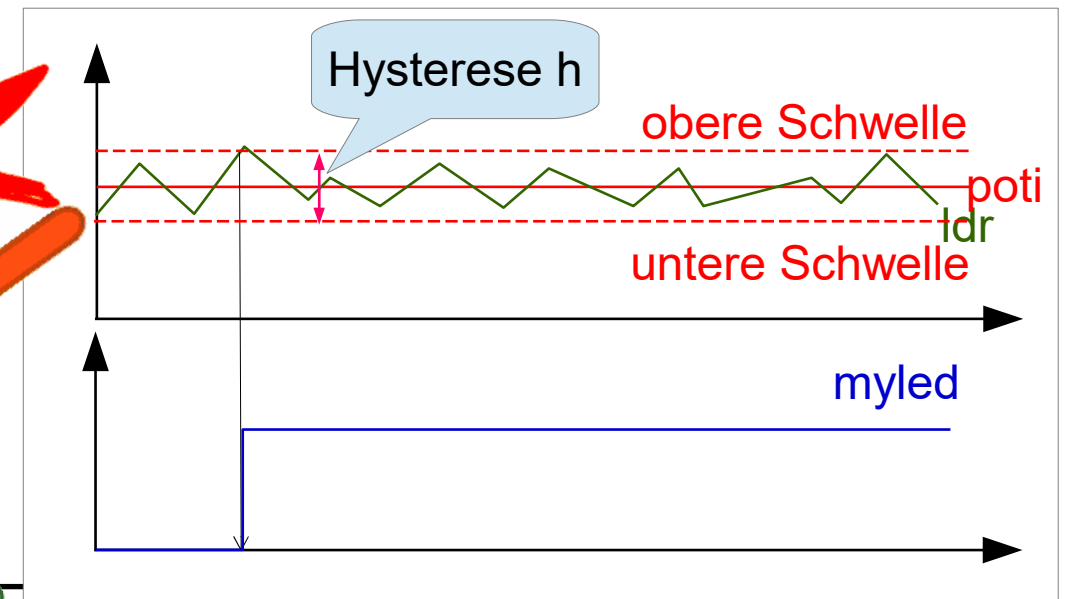
```
int main()  
{  
    while (true) {  
        if (poti < ldr) myled=1;  
        else myled=0;  
    }  
}
```



Digitale Verarbeitung analoger Signale



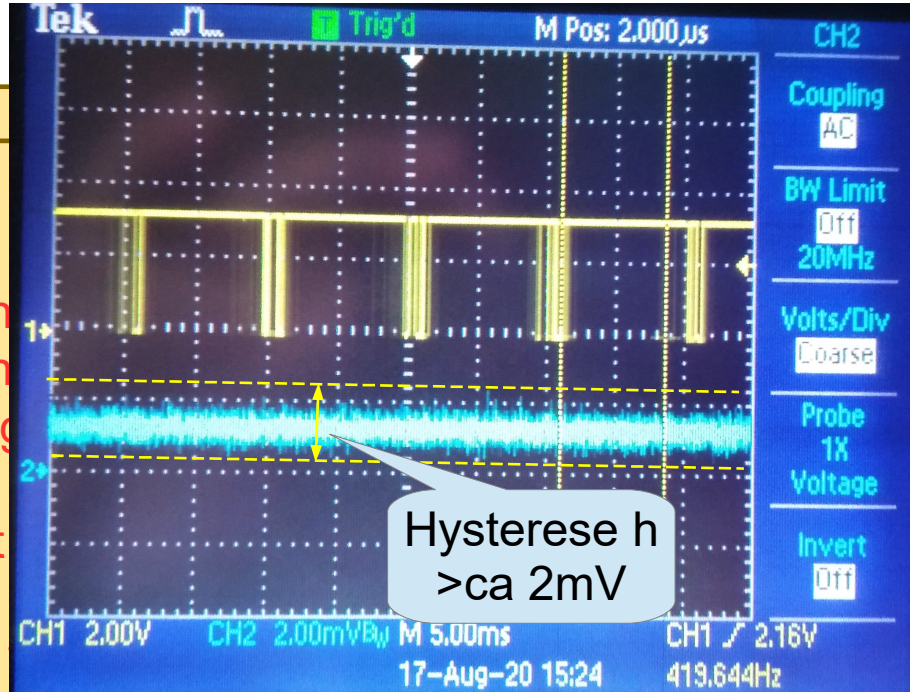
```
if (poti < ldr) myled = 1;  
else myled = 0;
```



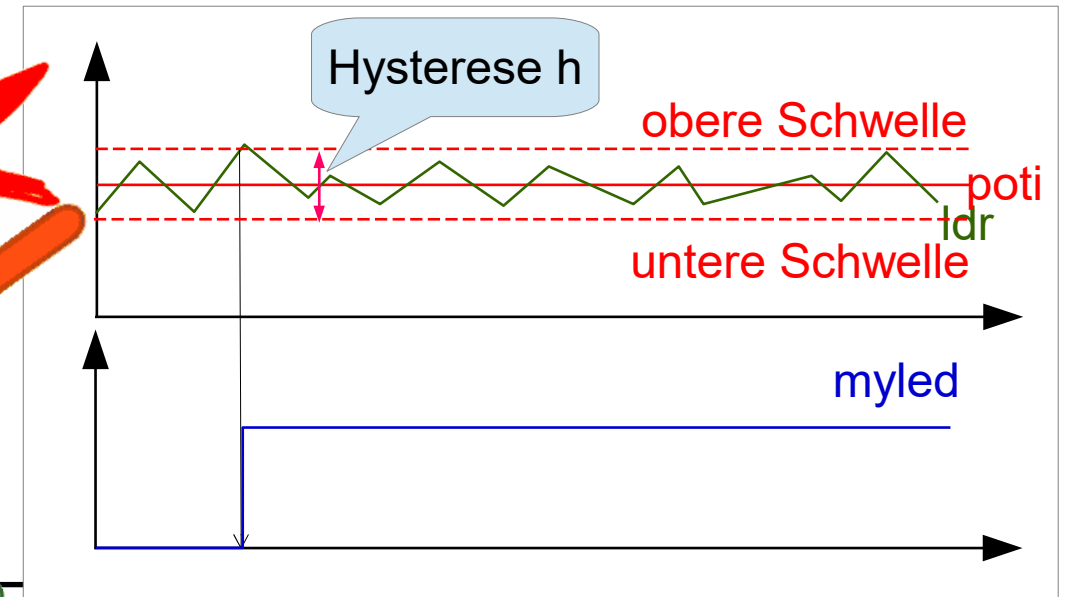
myled



Digitale Verarbeitung analoger Signale



```
if (poti < ldr) myled=1;  
else myled=0;
```



PB_0

PA_

MIKROCONTROLLER

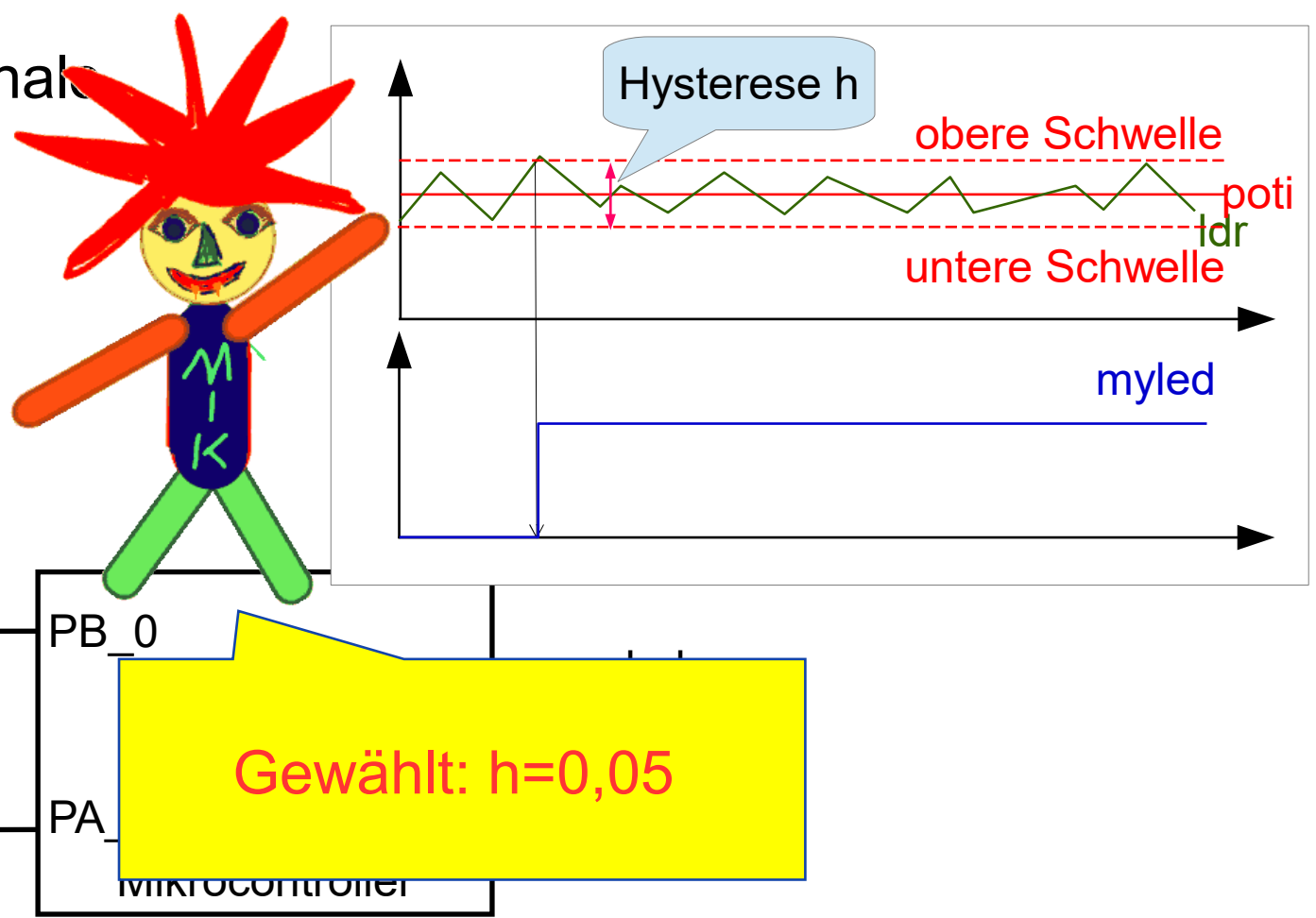
Gewählt: $h=0,005$



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

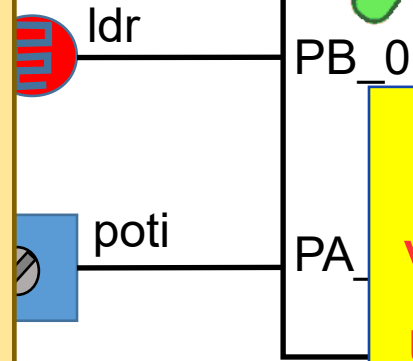
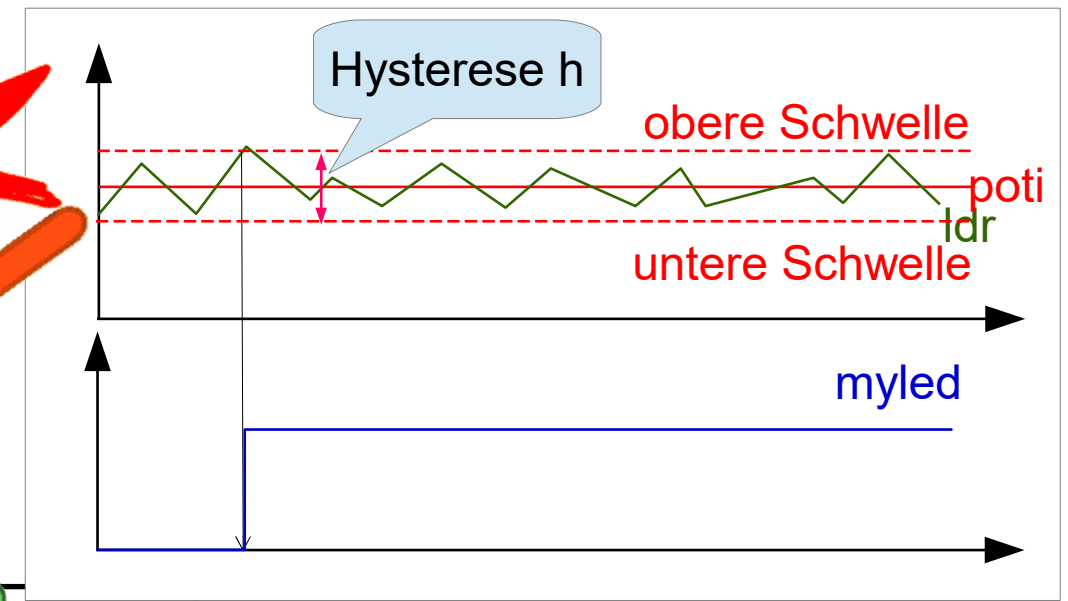
```
int main()  
{  
    float h=0.05;  
    while (true) {  
        if (ldr>poti+h/2) myled=1;  
        if (ldr<poti-h/2) myled=0;  
    }  
}
```



Digitale Verarbeitung analoger Signale

```
AnalogIn poti(PA_0);  
AnalogIn ldr(PB_0);  
DigitalOut myled(PC_0);
```

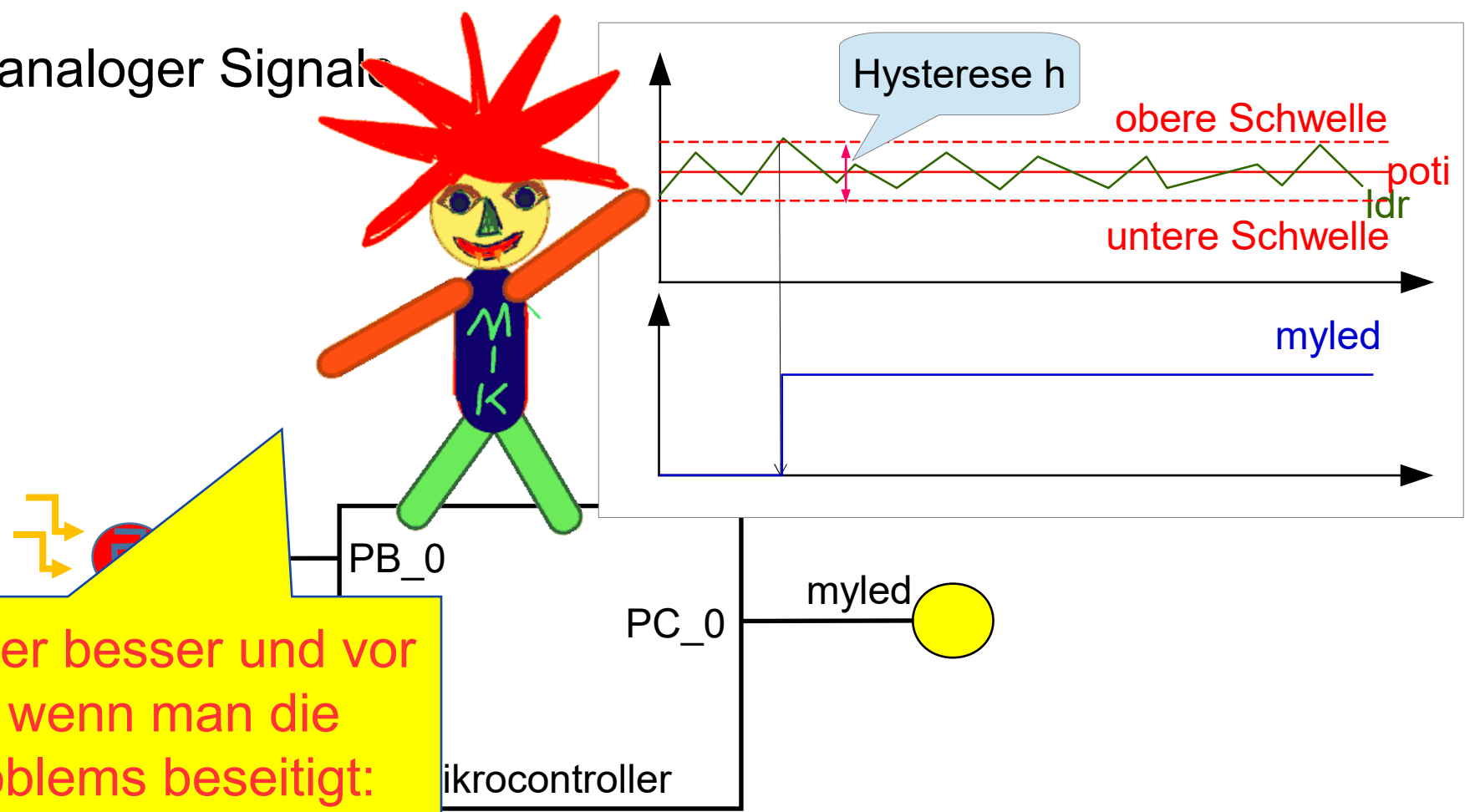
```
int main()  
{  
    float h=0.05;  
    while (true) {  
        if (ldr>poti+h/2) myled=1;  
        if (ldr<poti-h/2) myled=0;  
    }  
}
```



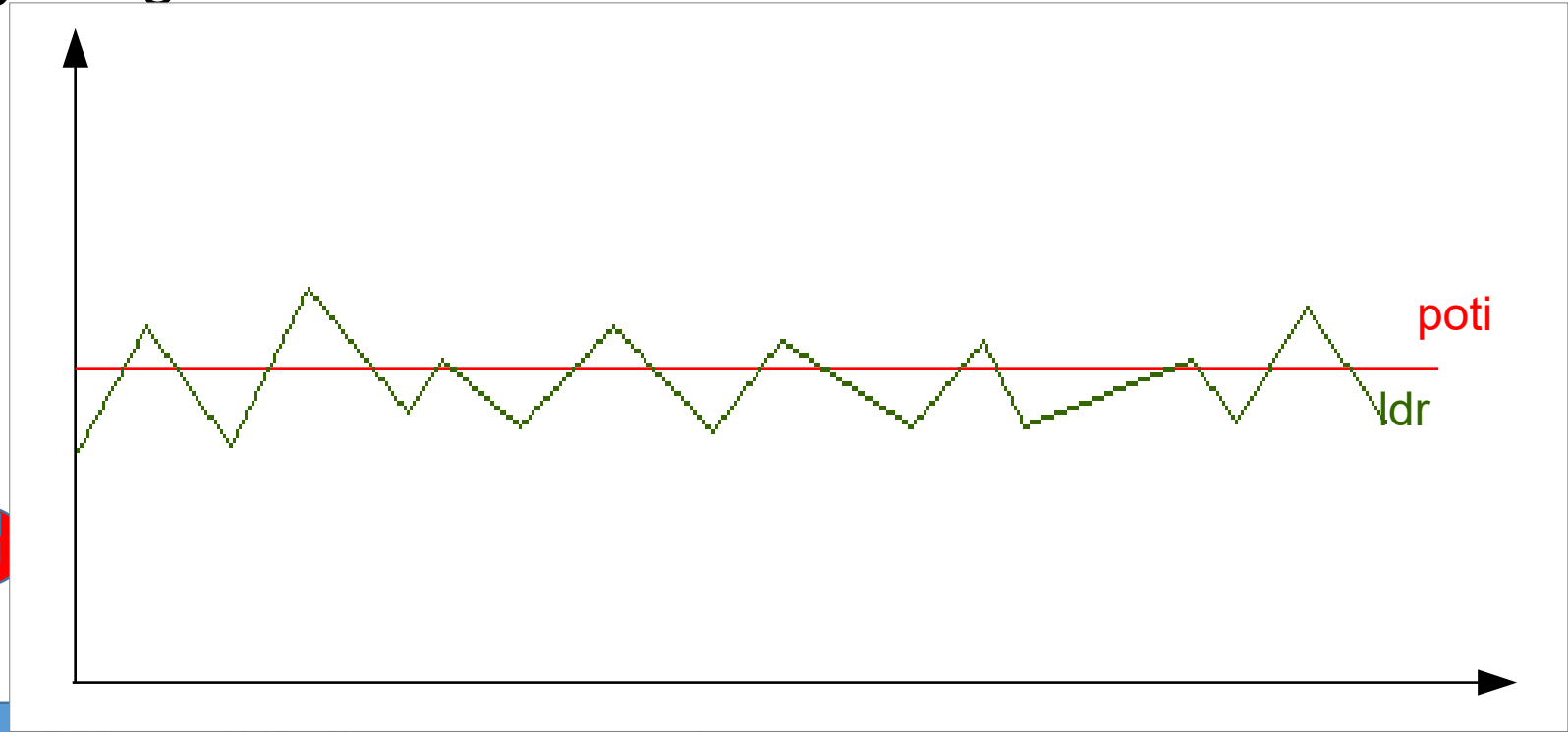
Hysterese bedeutet, dass man den Vergleichswert (poti) um eine obere und eine untere Schwelle erweitert. Der Abstand beider Schwellen heißt: „Hysterese“



Digitale Verarbeitung analoger Signale



Digitale Verarbeitung analoger Signale



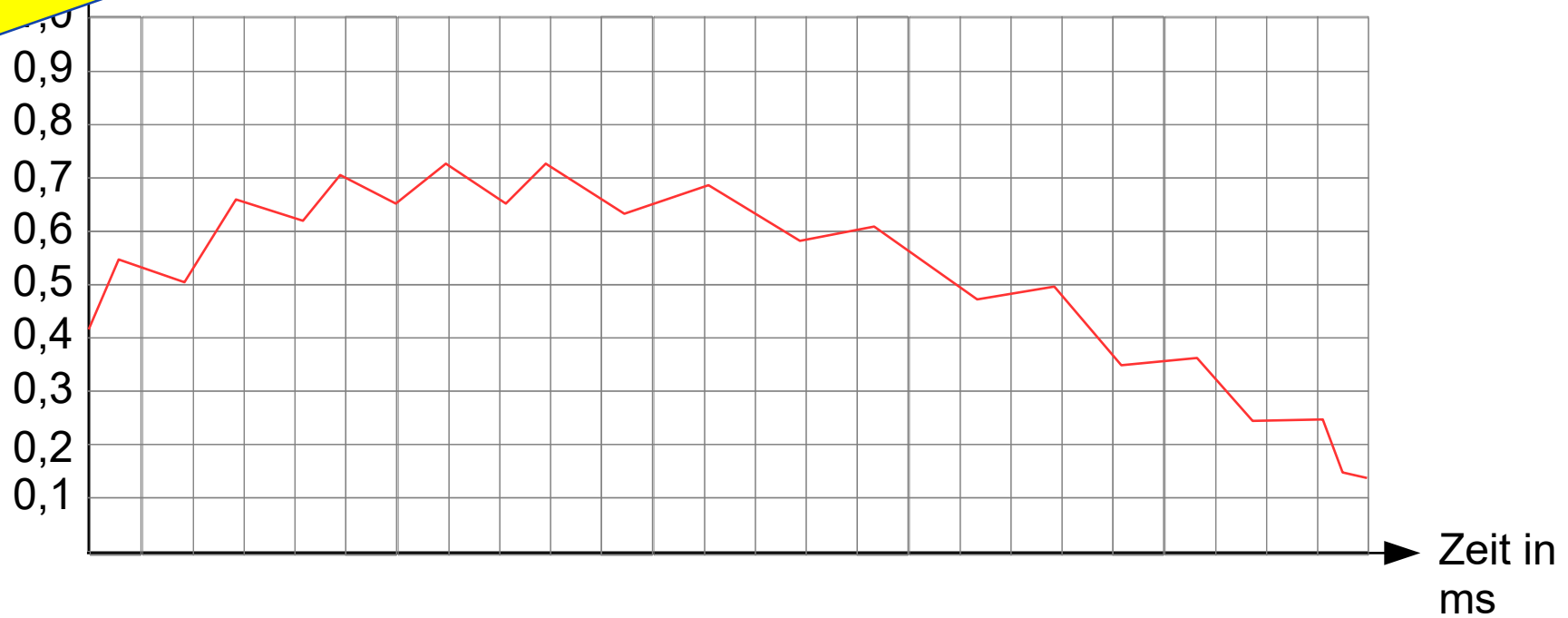
Eventuell ist es aber besser und vor allem
genauer, wenn man die Ursache des
Problems beseitigt:

**Die Störungen (Schwankungen) auf
dem Sensorsignal sollten entfernt
werden!!**



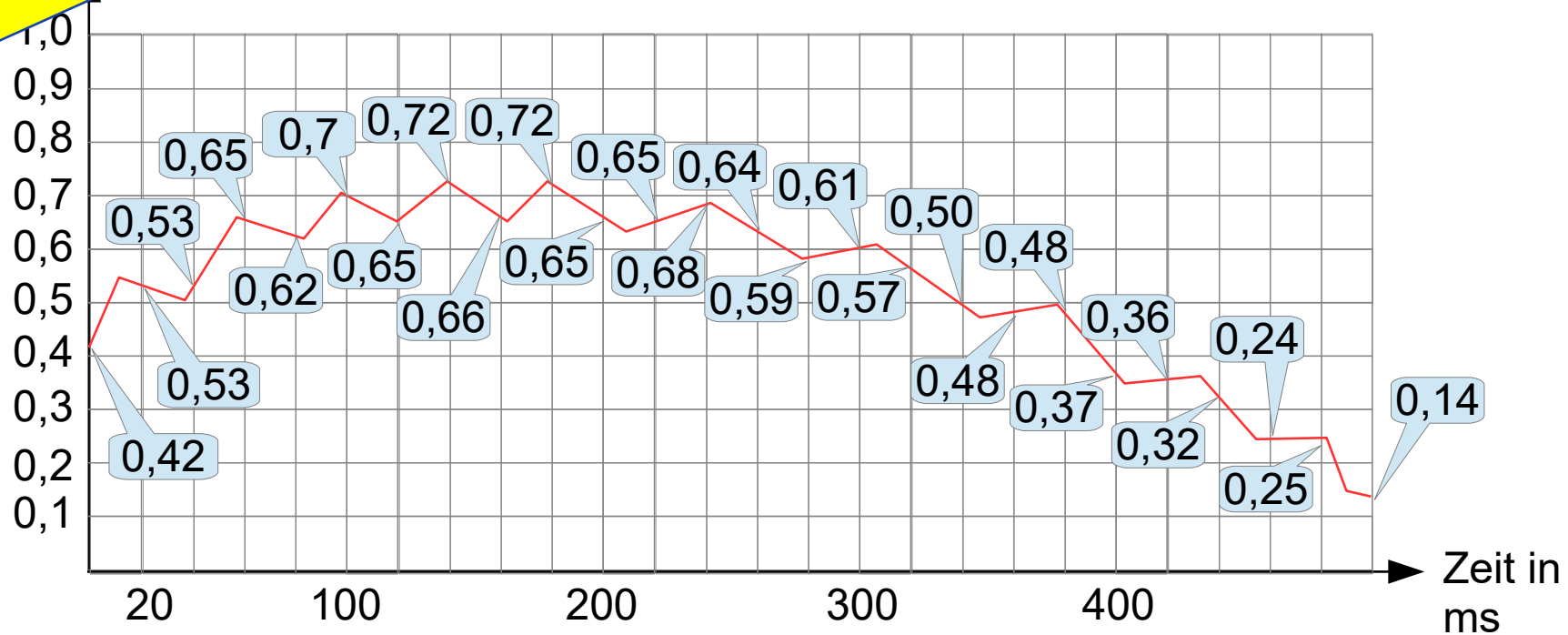
Digitale Verarbeitung analoger Signale

**Die Messwerte folgen einer Kurve und
sind gestört.**



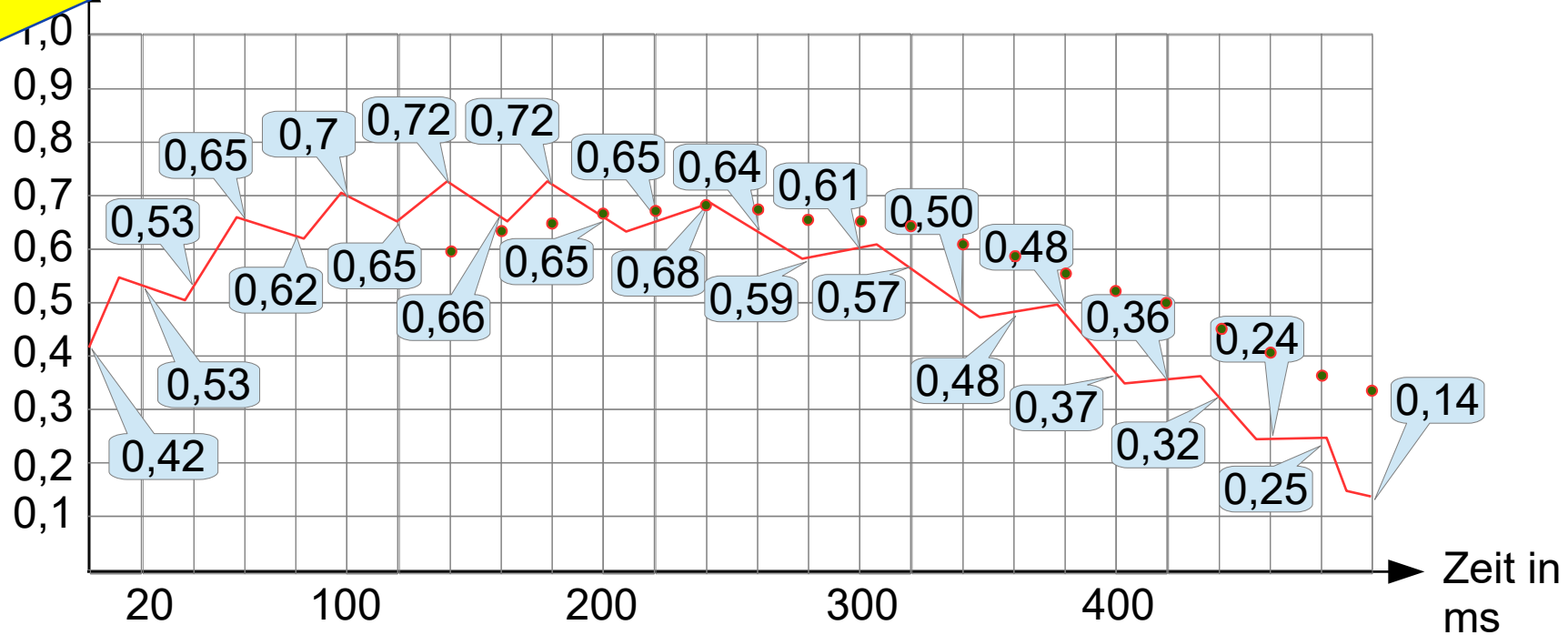
Digitale Verarbeitung analoger Signale

**Wir messen alle 20 ms.
Das nennt man Abtastung**



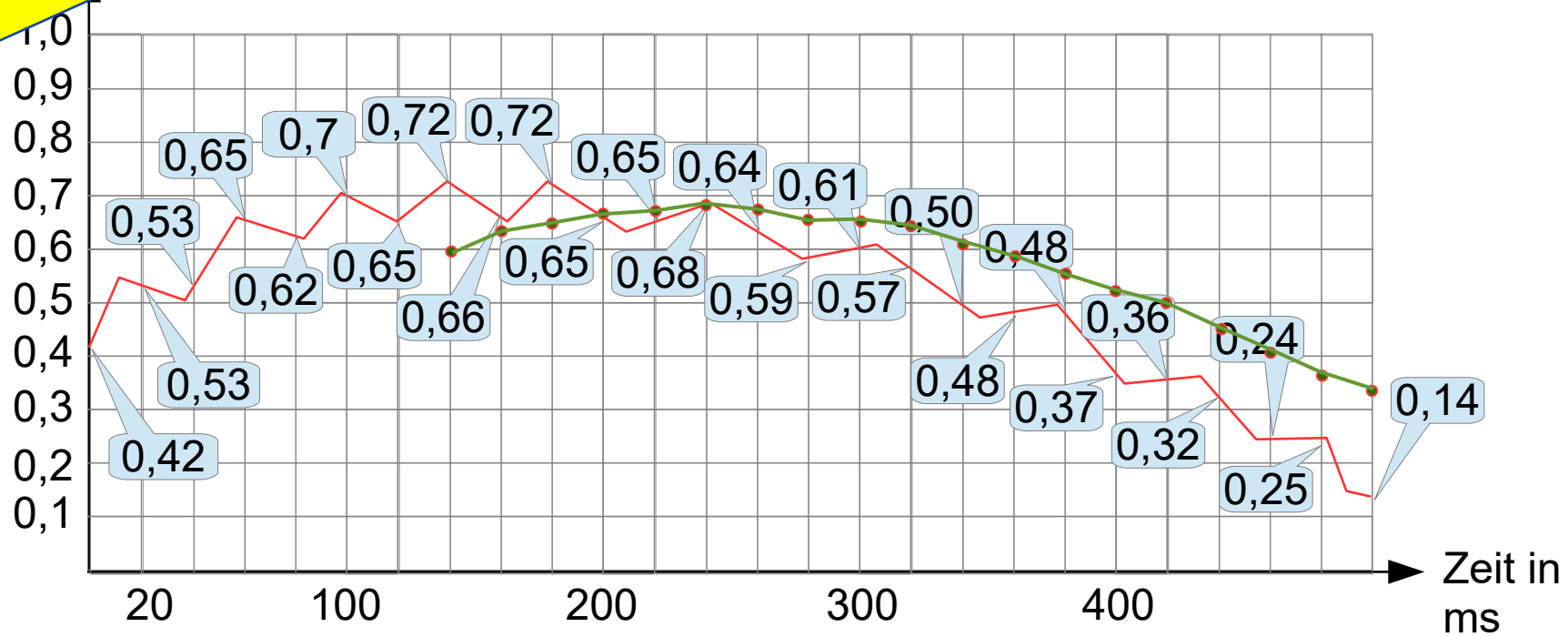
Digitale Verarbeitung analoger Signale

**Und bilden immer den
Mittelwert der letzten 8
Messwerte**



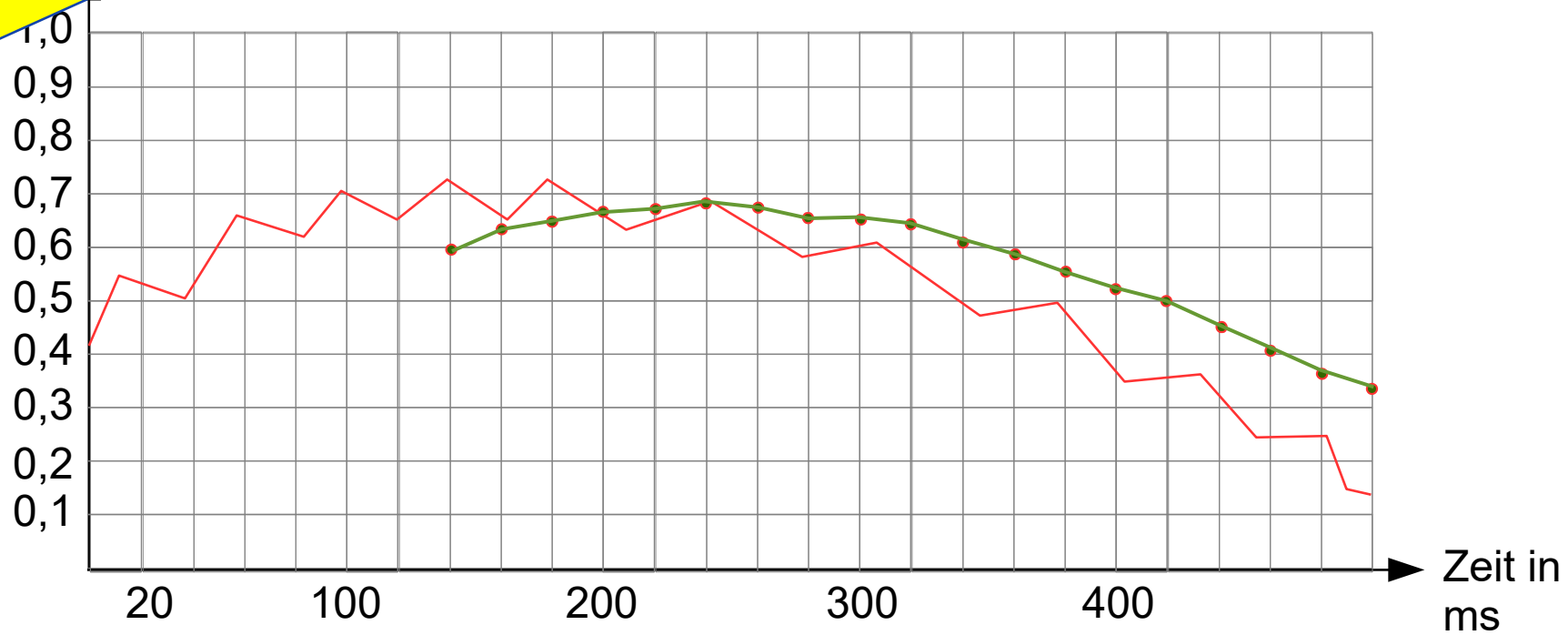
Digitale Verarbeitung analoger Signale

Wir verbinden die Punkte

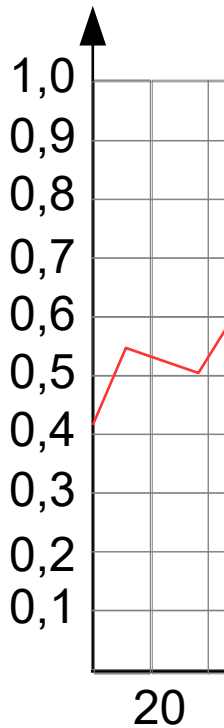


Digitale Verarbeitung analoger Signale

Die resultierende Kurve ist stark geglättet und nach rechts verschoben



Im Programm:



```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```



**Ticker t für
die 20ms**

```

AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```



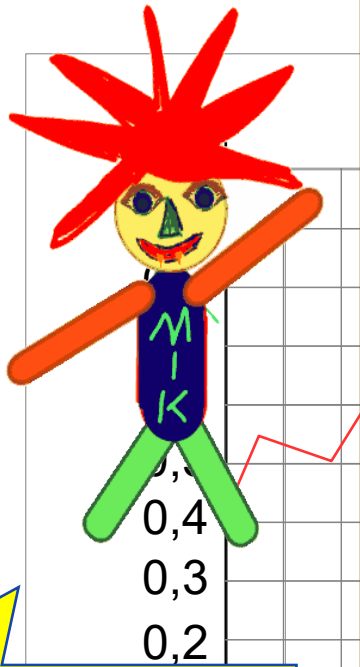

**Messwert-
Array**

```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```



**globale
Variablen**

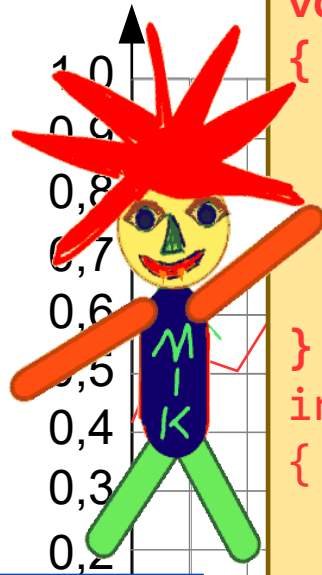
```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```



ISR

```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

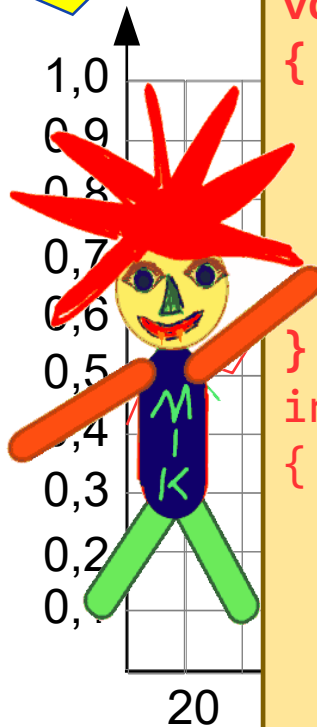


**Messwerte
aufsummieren**

```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

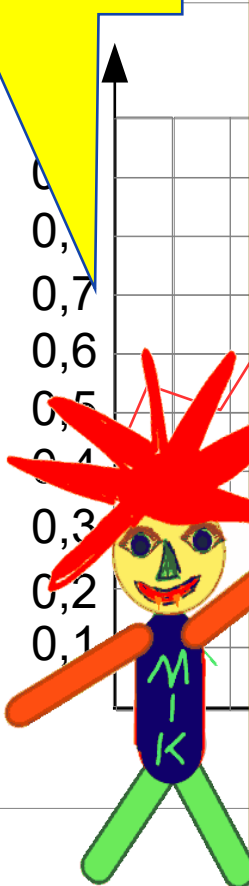
**Summe durch
8 teilen**



```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

**Ticker
initialisieren**



```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

**Messwert
vom AD-
Wandler holen**

0,7
0,6
0,5
0,4
0,3
0,2
0,1

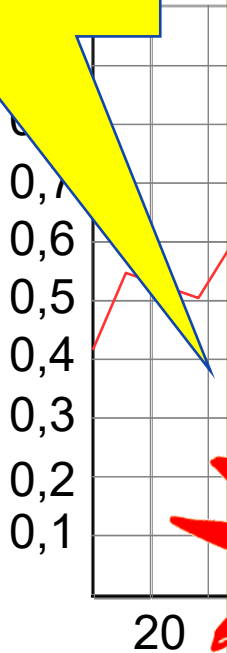
20



```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

**Messwertglatt
ermöglicht
eine deutlich
kleinere
Hysterese**



```
AnalogIn poti(PA_0);
AnalogIn ldr(PB_0);
DigitalOut myled(PC_0);
Ticker t;
float messwerte[8];
int i=0;
float messwert,messwertglatt;
void glaetten()
{
    messwerte[i%8]=messwert;
    messwertglatt=0;
    for (int j=0;j<8;j++) messwertglatt=messwertglatt+messwerte[j];
    messwertglatt=messwertglatt/8;
    i++;
}
int main()
{
    float h=0.01;
    t.attach(&glaetten,0.02);
    while (true) {
        messwert=ldr;
        if (messwertglatt>poti+h/2) myled=1;
        if (messwertglatt<poti-h/2) myled=0;
    }
}
```

Feldindex geht
von 0 bis 7

