

Inhalt

1. Einführung
2. Vorstellung des Programms
3. Modifizierung+
Einsatzmöglichkeiten
4. Fototransistor
5. Quelltext
6. Vorführen des Programms
7. Aufgetretene Probleme
8. Quellen

Einführung

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme



Quelle: http://www.alotec.de/bilder/331_dscn5609-1_roboter_monsano.jpg

Einführung

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen



Quelle: <http://www.bmlv.gv.at/pool/img/teodor.jpg>

Einführung

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen



Quelle: http://img.stern.de/_content/52/98/529811/roboter_250.jpg

Einführung

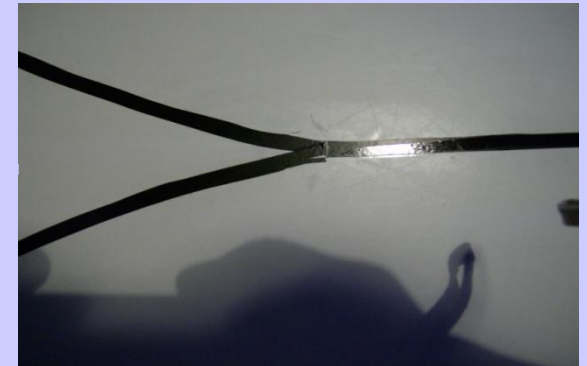
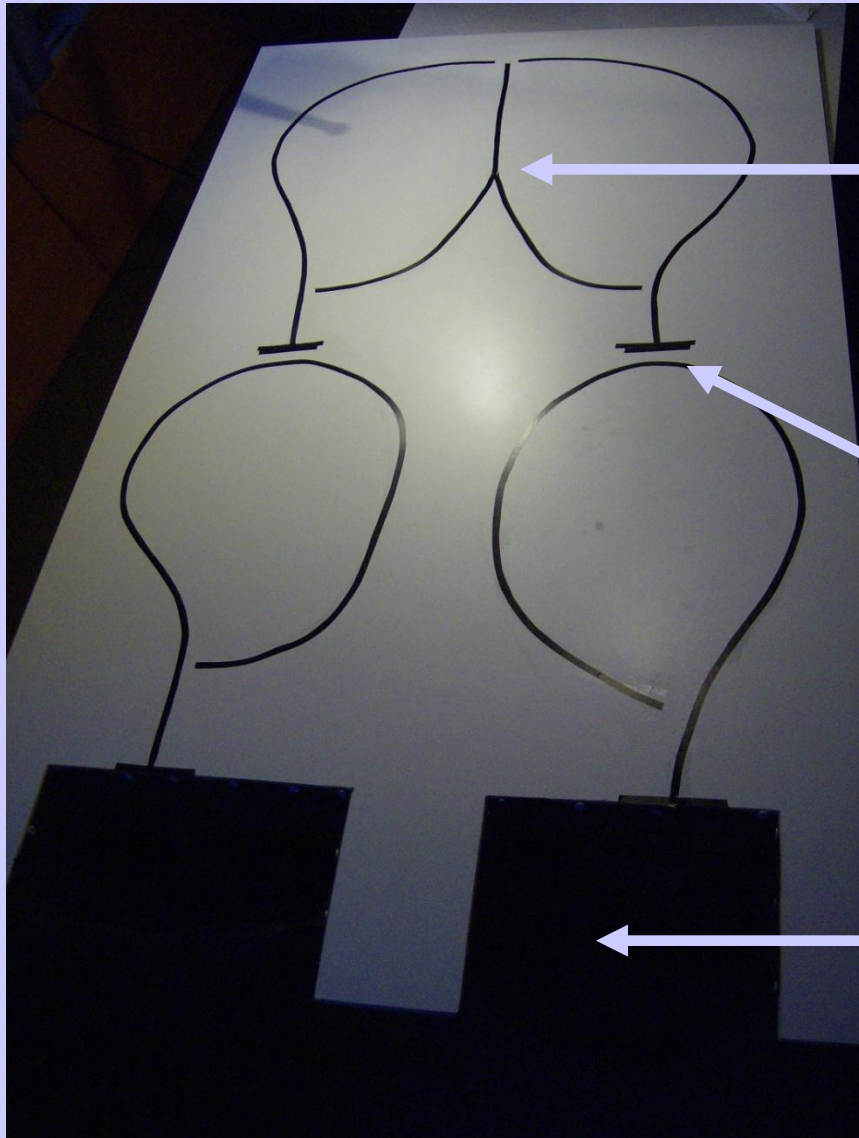
1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen



Quelle: <http://www.srlingolstadt.de/Indoor%20DM%202006%20Muell%20vom%20Fahrerlager.JPG>

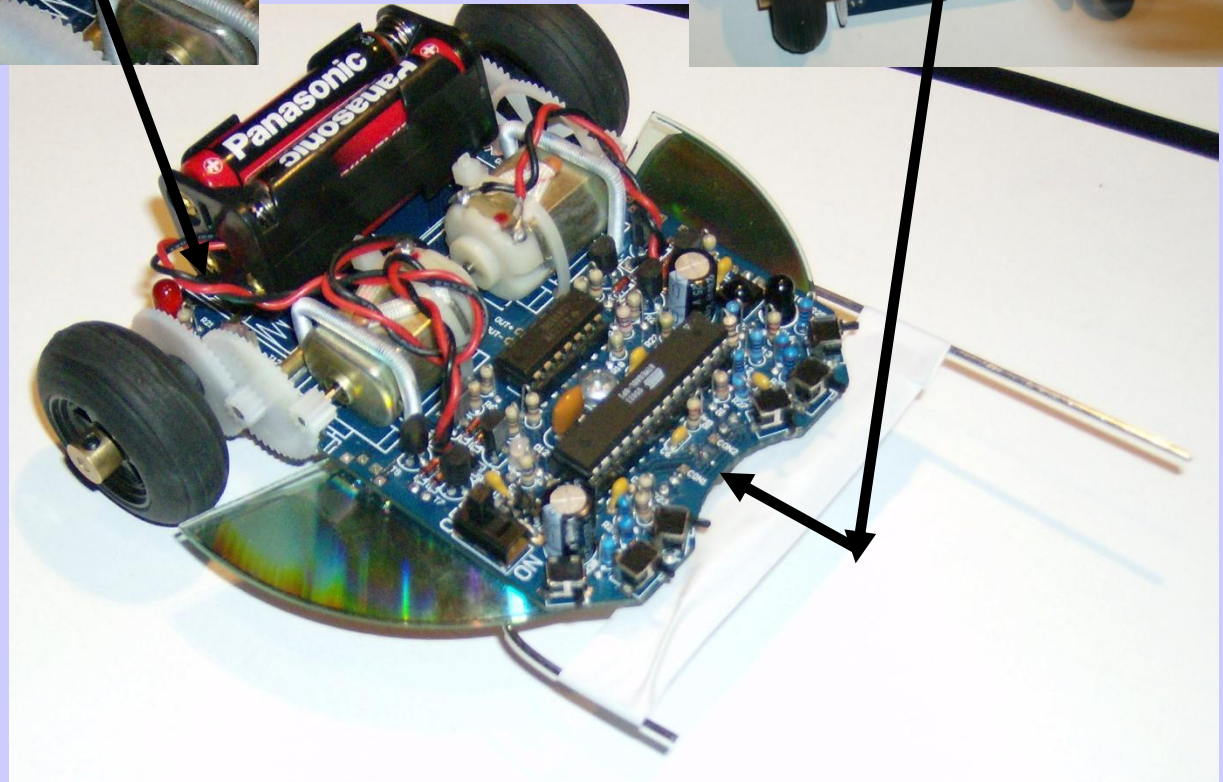
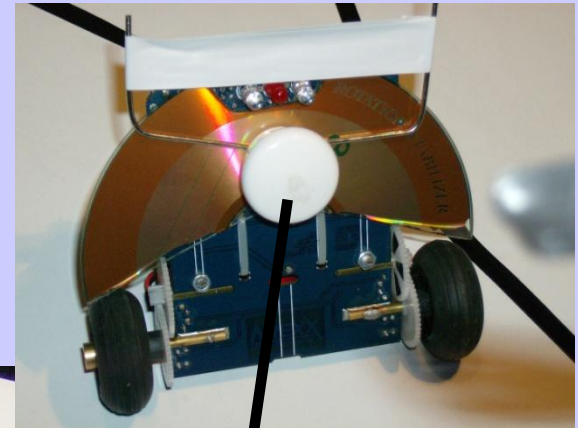
2. Vorstellung des Programms

1. Einführung
- 2. Vorstellung des Programms**
3. Modifizierung + Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene Probleme
8. Quellen



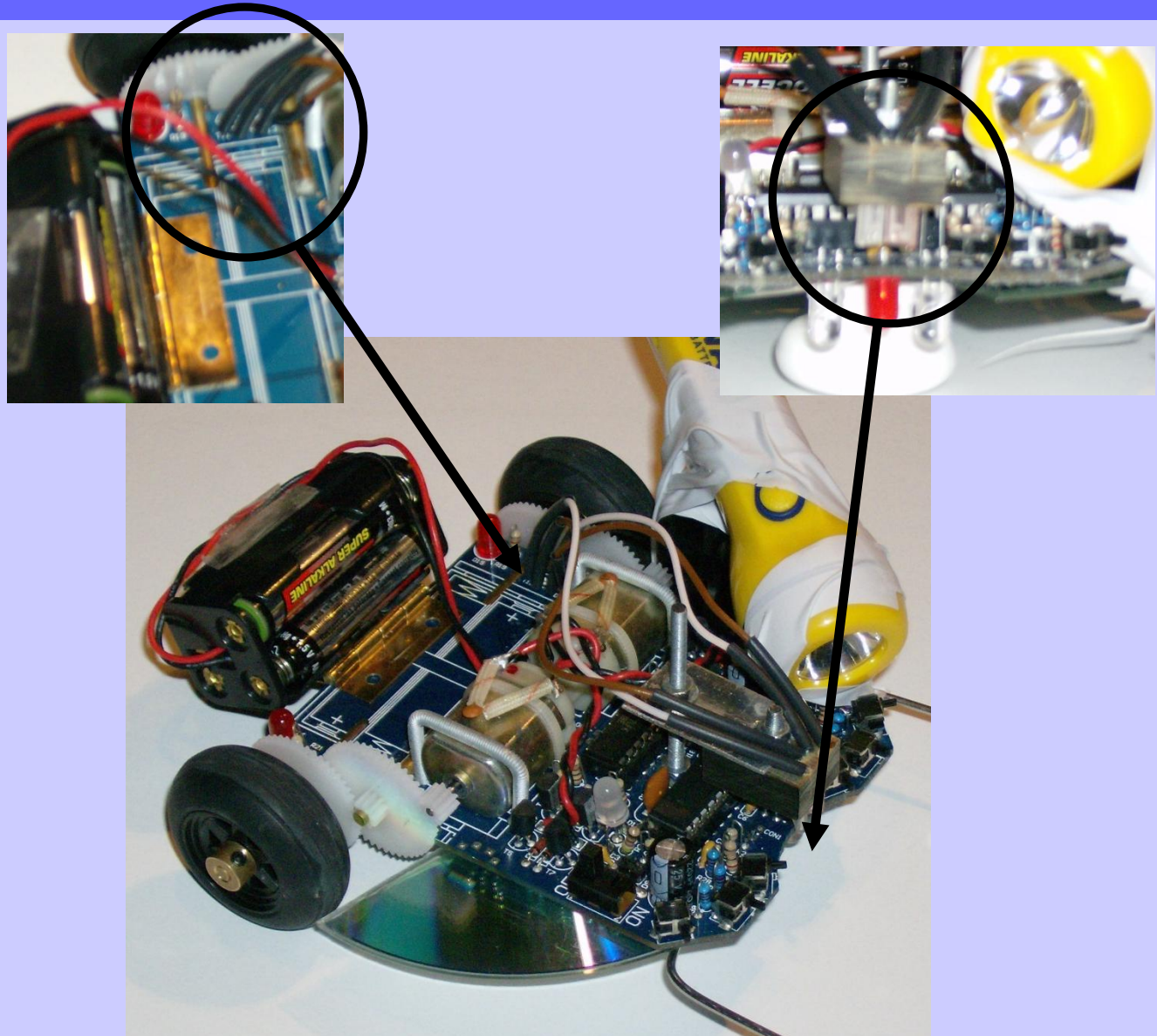
3. Modifizierung + Zukunft

1. Einführung
2. Vorstellung
des Programms
- 3. Modifizierung +
Zukunft**
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen



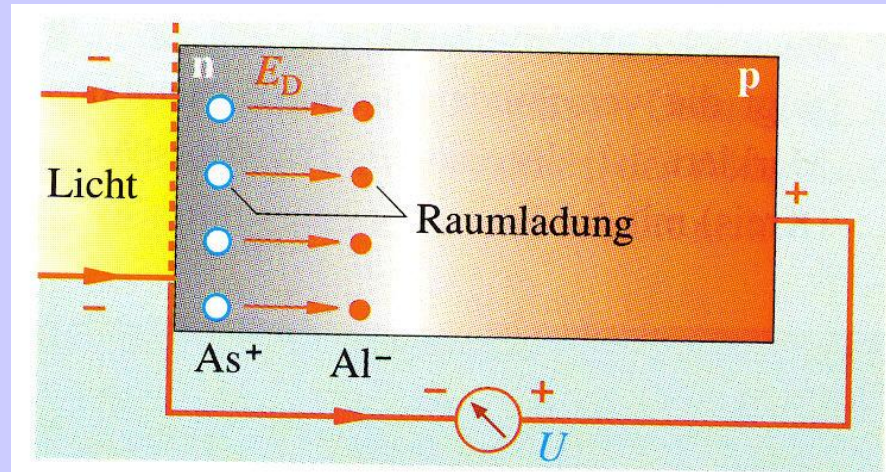
3. Modifizierung + Zukunft

1. Einführung
2. Vorstellung
des Programms
- 3. Modifizierung +
Zukunft**
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen

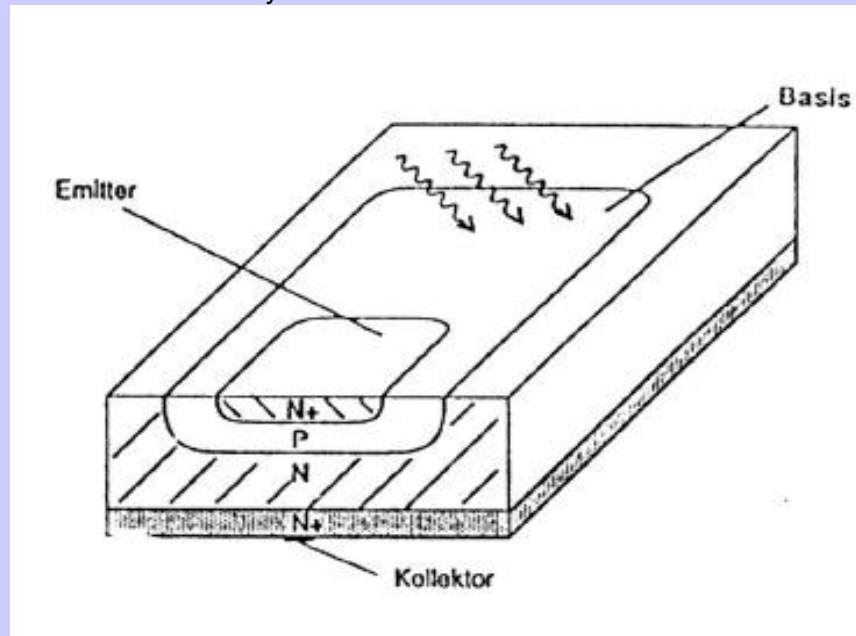


4. Fototransistor

1. Einführung
2. Vorstellung des Programms
3. Modifizierung + Zukunft
- 4. Fototransistor**
5. Quelltext
6. Vorführung
7. Aufgetretene Probleme
8. Quellen



Quelle: Dorn-Bader Physik 12/13



5. Quelltext

-ASURO 1/2-

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
- 5. Quelltext**
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen

```
#include "asuro.h"
#define HALT 80
//LINKER ASURO

int main(void)
{
    Init();

    unsigned int Ftrans[2];
    unsigned char abgrund = FALSE;

    FrontLED(ON);
    MotorDir(FWD, FWD);
    while(1)
    {
        MotorDir(FWD, FWD);
        if(! (abgrund))
        {
            LineData(Ftrans);
            if (Ftrans[1] > Ftrans[0])
            {
                MotorSpeed(90, 230);
                BackLED(OFF, ON);
                StatusLED(GREEN);
                abgrund = FALSE;
            }
            else if((Ftrans[0] < HALT) && (Ftrans[1] < HALT))
            {
                MotorDir(BREAK, BREAK);
                MotorSpeed(0, 0);
                StatusLED(RED);
                abgrund=TRUE;
            }
        }
    }
}
```

```
else
{
    MotorSpeed(230, 90);
    BackLED(ON, OFF);
    StatusLED(YELLOW);
    abgrund = FALSE;
}
else if(abgrund)
{
    MotorDir(RWD, RWD);
    MotorSpeed(0, 140);
    Msleep(1200);
    MotorDir(FWD, FWD);
    MotorSpeed(150, 150);
    Msleep(100);
    abgrund = FALSE;
}

return 0;
}
```


5. Quelltext

-ASURO 3-

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
- 5. Quelltext**
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen

```
#include "asuro.h"
#define HALT 100
#define ODOMETRIE 995

#define HELL 1
#define DUNKEL 0

#define LINKS 0
#define RECHTS 1

void Kurve(int LinksRechts, int VorZurueck) { ... }
void Kurve2(int LinksRechts, int VorZurueck) { ... }
int main(void)
{
    Init();

    unsigned int Ftrans[2];
    unsigned int Otrans[2];

    int Richtung = LINKS;
    int Status = 0;

    int i;

    FrontLED(ON);

    while(1){
        switch(Status)
        {
            //Ablauf des Programms
        }

    }

    return 0;
}
```

5. Quelltext

-ASURO 3-

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
- 5. Quelltext**
6. Vorführung
7. Aufgetretene
Probleme
8. Quellen

```
switch (Status)
{
    case 0:
        LineData(Ftrans);
        StatusLED(RED);
        if(Ftrans[LINKS] < Ftrans[RECHTS])
        {
            MotorSpeed(80,190);
        }
        else if(Ftrans[LINKS] > Ftrans[RECHTS])
        {
            MotorSpeed(190,80);
        }
        if((PollSwitch() == 2) || (PollSwitch() == 16))
        {
            Status = 1;
        }
        break;
    case 1:
        MotorSpeed(0,0);
        for(i=1;i<=10;i++)
        {
            OdometrieData(Otrans);
            if(Otrans[LINKS] > ODOMETRIE)
            {
                Richtung = RECHTS;
                BackLED(OFF,ON);
            }
            else if(Otrans[LINKS] < ODOMETRIE)
            {
                Richtung = LINKS;
                BackLED(ON,OFF);
            }
        }
        Msleep(1000);
        Status = 2;
        break;
```

```
case 2:
    Kurve(Richtung,FWD);
    MotorDir(FWD,FWD);
    Status = 3;
    break;
case 3:
    LineData(Ftrans);
    StatusLED(RED);
    if(Ftrans[LINKS] < Ftrans[RECHTS])
    {
        MotorSpeed(80,190);
    }
    else if(Ftrans[LINKS] > Ftrans[RECHTS])
    {
        MotorSpeed(190,80);
    }
    else if((Ftrans[RECHTS] < HALT) && (Ftrans[LINKS] < HALT))
    {
        StatusLED(YELLOW);
        MotorDir(BREAK,BREAK);
        MotorSpeed(0,0);
        Status = 4;
    }
    break;
case 4:
    MotorDir(RWD,RWD);
    MotorSpeed(200,200);
    Msleep(550);
    Kurve2(Richtung,FWD);
    Status = 0;
    break;
```


7. Aufgetretene Probleme

-ASURO-

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
- 7. Aufgetretene
Probleme**
8. Quellen

- Odometrie → Linienverfolgung
→ Msleep()
- Abgrundproblem → keine Kreuzungen
- Tasterproblem → Tieferlegung
- Gegenstandserkennung → Odometrie

7. Aufgetretene Probleme -Programmierung-

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
- 7. Aufgetretene
Probleme**
8. Quellen

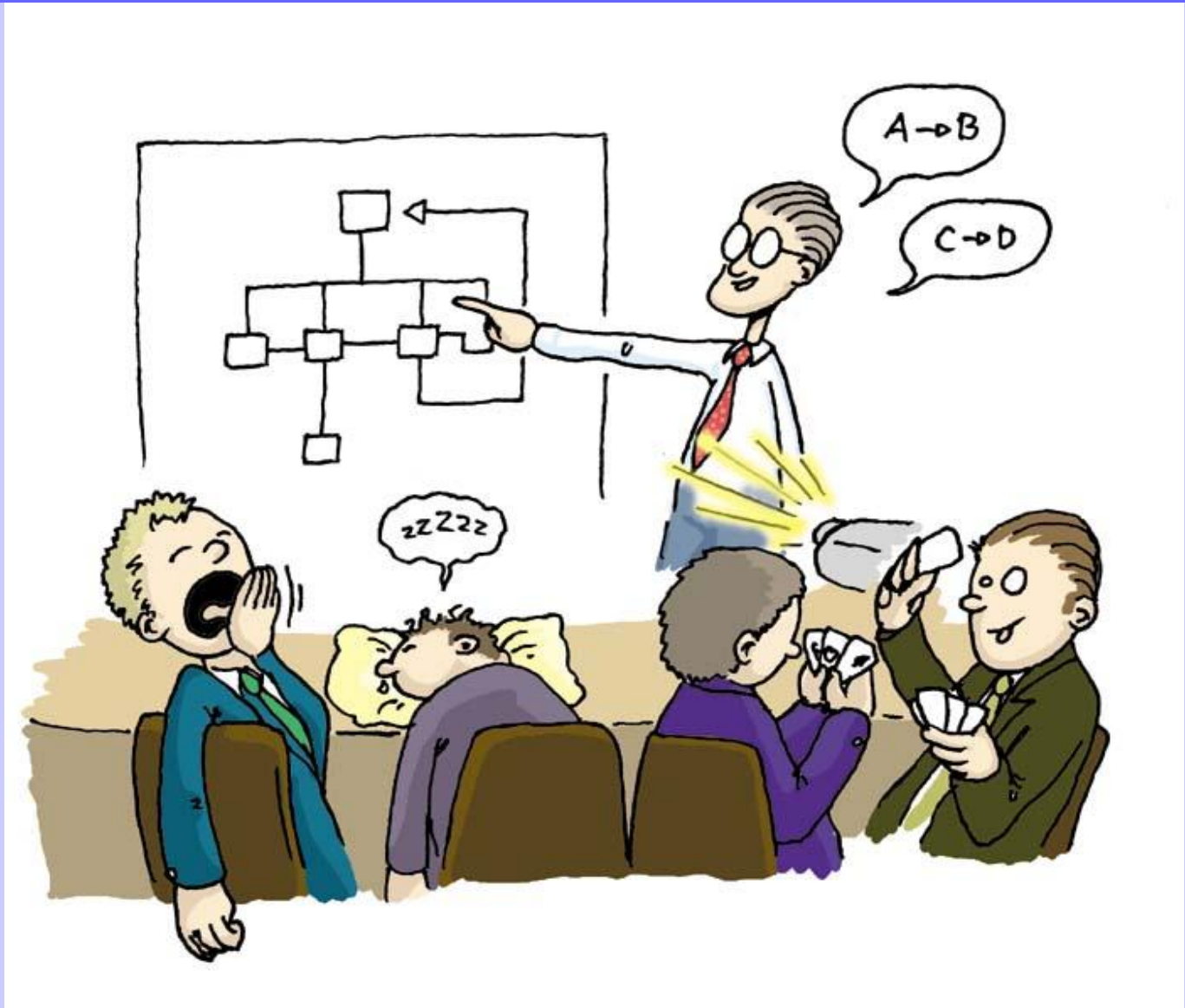
- Odometrie → Zählschleife
- Taster → einzelne Abfrage
- Anpassung an Lichtverhältnisse
→ Sensorabfrage

8. Quellen

1. Einführung
2. Vorstellung
des Programms
3. Modifizierung +
Zukunft
4. Fototransistor
5. Quelltext
6. Vorführung
7. Aufgetretene
Probleme
- 8. Quellen**

- www.alotec.de
- www.bmlv.gv.at
- www.stern.de
- www.srlringolstadt.de
- www.asurowiki.de
- www.arexx.com
- Mehr Spaß mit dem ASURO Band 1

Noch Fragen???



```
#include "asuro.h"
#define HALT 80
//LINKER ASURO

int main(void)
{
    Init();

    unsigned int Ftrans[2];
    unsigned char abgrund = FALSE;

    FrontLED(ON);
    MotorDir(FWD.FWD);
```



```
while (1)
{
    MotorDir (FWD, FWD) ;
    if (! (abgrund) )
    {
        LineData (Ftrans) ;
        if (Ftrans[1] > Ftrans[0])
        {
            MotorSpeed (90, 230) ;
            BackLED (OFF, ON) ;
            StatusLED (GREEN) ;
            abgrund = FALSE ;
        }
        else if ((Ftrans[0] < HALT) && (Ftrans[1] < HALT))
        {
            MotorDir (BREAK, BREAK) ;
            MotorSpeed (0, 0) ;
            StatusLED (RED) ;
            abgrund=TRUE ;
        }
    }
}
```

```
else
{
    MotorSpeed(230, 90);
    BackLED(ON, OFF);
    StatusLED(YELLOW);
    abgrund = FALSE;
}
}
else if(abgrund)
{
    MotorDir(RWD, RWD);
    MotorSpeed(0, 140);
    Msleep(1200);
    MotorDir(FWD, FWD);
    MotorSpeed(150, 150);
    Msleep(100);
    abgrund = FALSE;
}
}

return 0;
}
```

```
#include "asuro.h"  
#define HALT 100  
#define ODOMETRIE 995
```

```
#define HELL 1  
#define DUNKEL 0
```

```
#define LINKS 0  
#define RECHTS 1
```

```
void Kurve(int LinksRechts, int VorZurueck) { ... }  
void Kurve2(int LinksRechts, int VorZurueck) { ... }
```

```
int main(void)
{
    Init();

    unsigned int Ftrans[2];
    unsigned int Otrans[2];

    int Richtung = LINKS;
    int Status = 0;

    int i;

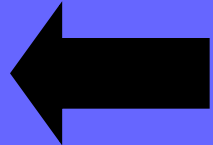
    FrontLED(ON);

    while(1){
        switch(Status)
        {
            //Ablauf des Programms
        }

    }
    return 0;
}
```



```
switch(Status)
{
    case 0:
        LineData(Ftrans);
        StatusLED(RED);
        if(Ftrans[LINKS] < Ftrans[RECHTS])
        {
            MotorSpeed(80,190);
        }
        else if(Ftrans[LINKS] > Ftrans[RECHTS])
        {
            MotorSpeed(190,80);
        }
        if((PollSwitch() == 2) || (PollSwitch() == 16))
        {
            Status = 1;
        }
        break;
}
```



```
case 1:
    MotorSpeed(0,0);
    for(i=1;i<=10;i++)
    {
        OdometrieData(Otrans);
        if(Otrans[LINKS] > ODOMETRIE)
        {
            Richtung = RECHTS;
            BackLED(OFF,ON);
        }
        else if(Otrans[LINKS] < ODOMETRIE)
        {
            Richtung = LINKS;
            BackLED(ON,OFF);
        }
    }
    Msleep(1000);
    Status = 2;
    break;
```

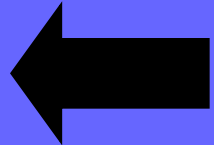
case 2:

```
Kurve(Richtung, FWD) ;
```

```
MotorDir(FWD, FWD) ;
```

```
Status = 3;
```

```
break;
```



```
case 3:
    LineData(Ftrans);
    StatusLED(RED);
    if(Ftrans[LINKS] < Ftrans[RECHTS])
    {
        MotorSpeed(80,190);
    }
    else if(Ftrans[LINKS] > Ftrans[RECHTS])
    {
        MotorSpeed(190,80);
    }
    else if((Ftrans[RECHTS] < HALT) && (Ftrans[LINKS] < HALT))
    {
        StatusLED(YELLOW);
        MotorDir(BREAK,BREAK);
        MotorSpeed(0,0);
        Status = 4;
    }
    break;
```


case 4:

```
MotorDir(RWD,RWD);  
MotorSpeed(200,200);  
Msleep(550);  
Kurve2(Richtung,FWD);  
Status = 0;  
break;
```

