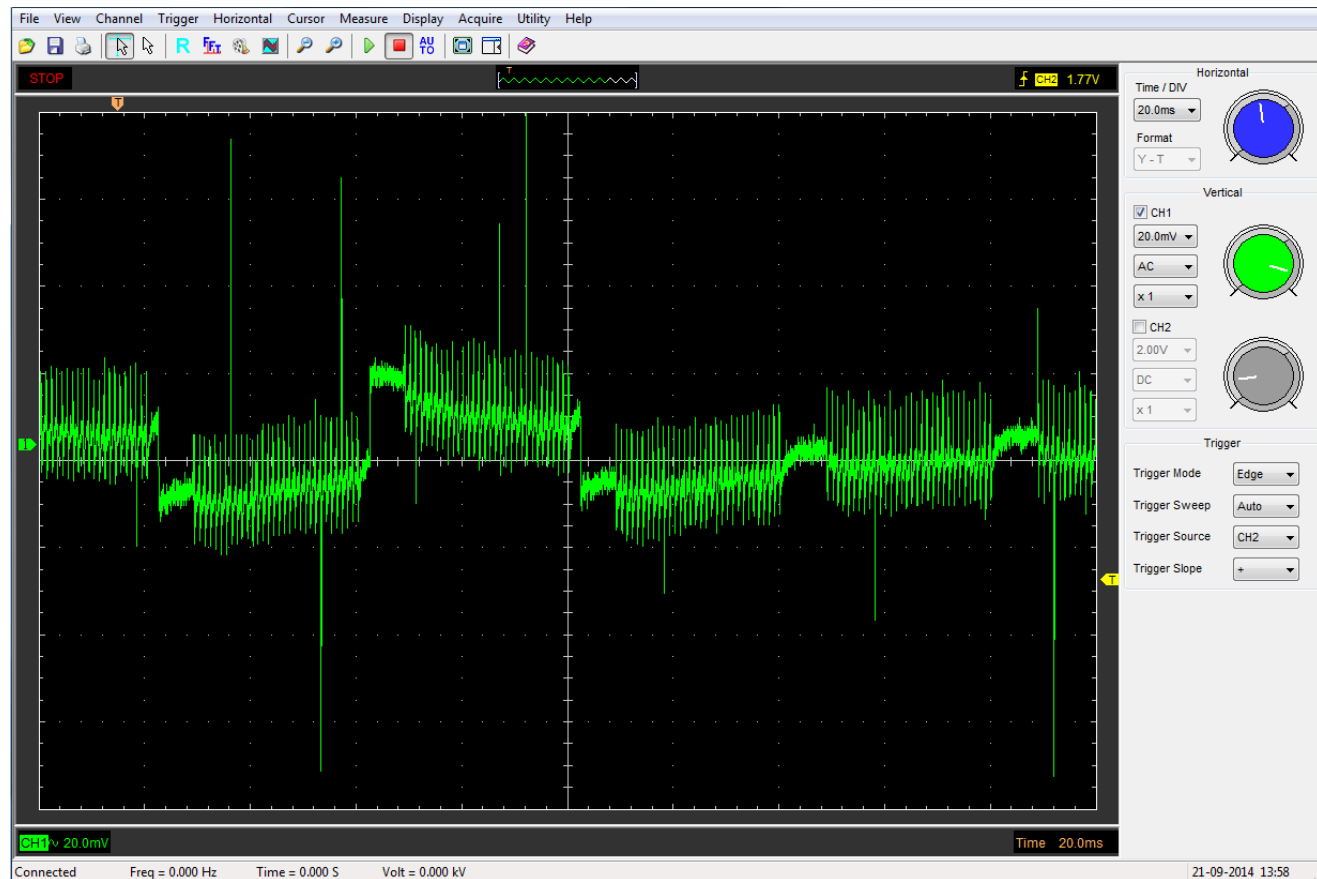


CAR – Dag 7

C, Arduino & Robots

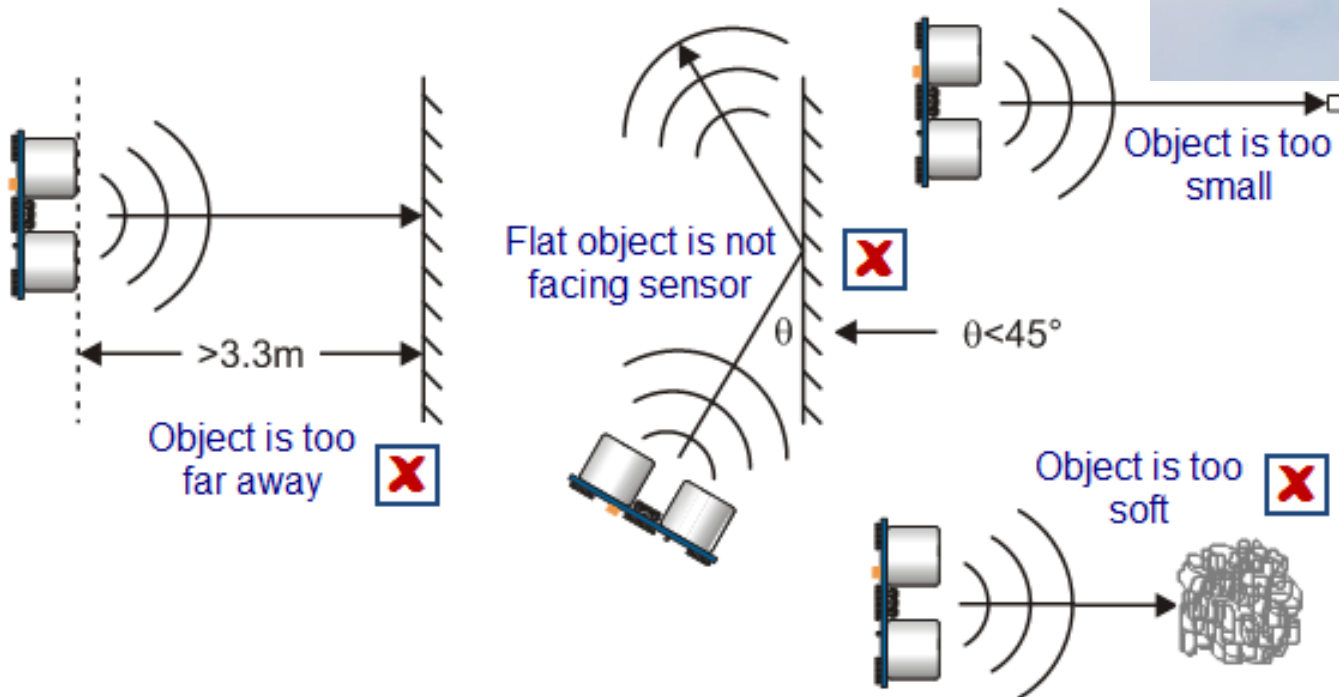
Sensor ruis

- Variatie in meetwaarde.



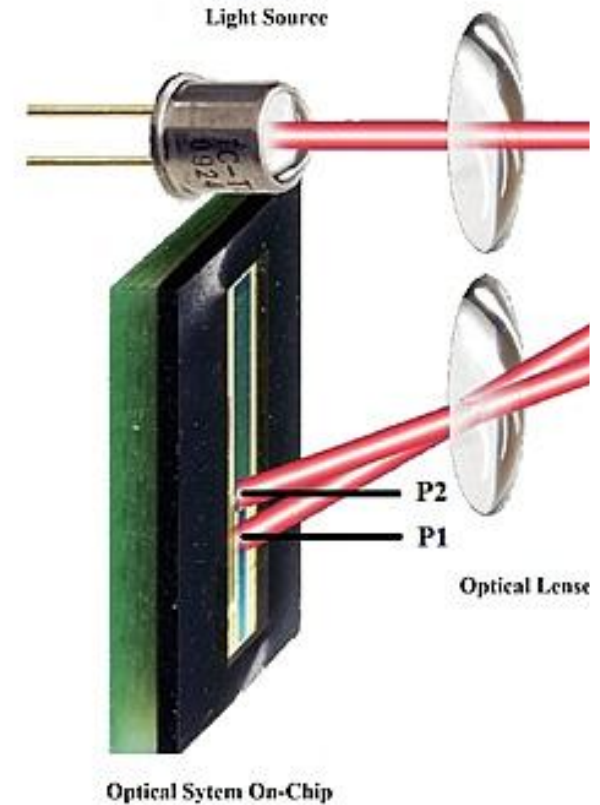
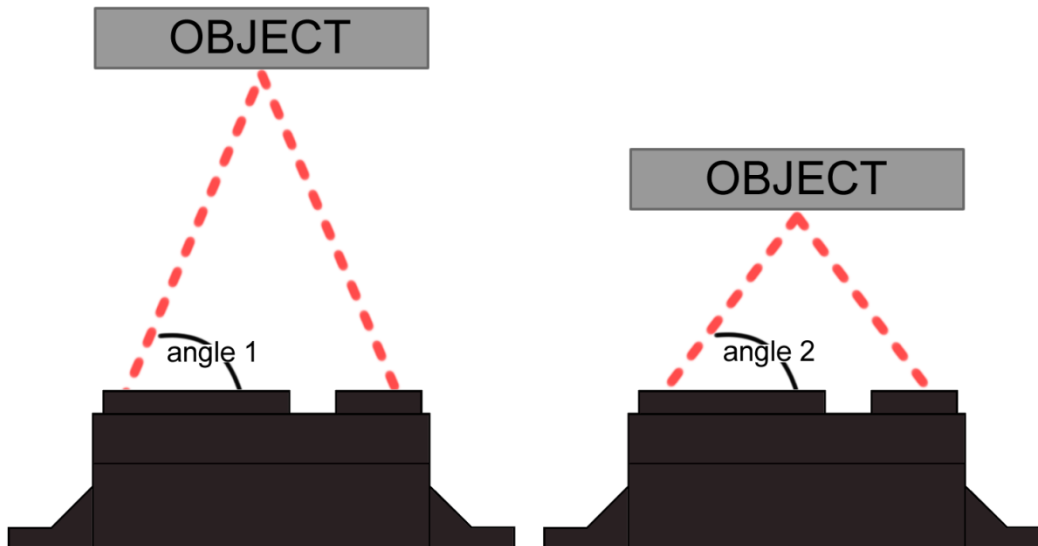
Sensor – stealth

- ▶ Niet alles zichtbaar.

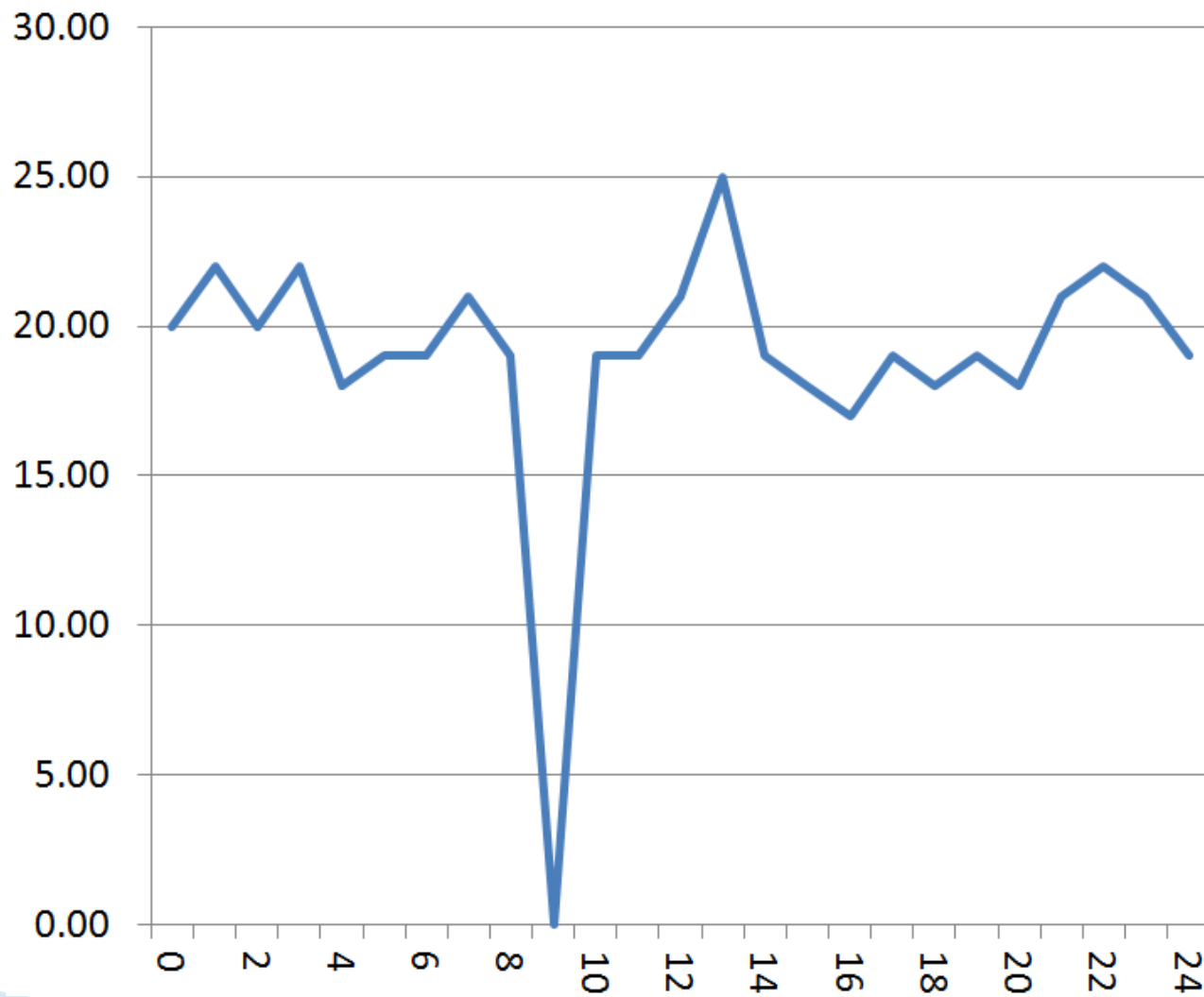


Sharp – Fantoom

- ▶ Driehoeksmeting



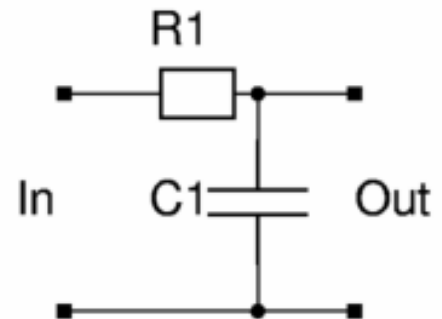
Sensor ruins



Middelen sensor (laagdoorlaat)

- ▶ Dempen (vertragen) van variaties in input
- ▶ Hoeveel dempen?
 - Te snel: toch nog last van variaties
 - Te langzaam: wijzigingen komen te laat door
- ▶ $\text{Nieuwe waarde} = 7/8 * \text{oude waarde} + 1/8 * \text{nieuwe waarde}.$

Low pass filter



Middelen- Oefening

les_7_p10_sharp_lpf

Sharp_lpf

```
int SharpAfstand(int Pin, int &Middelen)
{
    int SensorValue = analogRead(Pin);

    Middelen = Middelen - Middelen / 8 + SensorValue;

    int Gemiddelde = Middelen / 8;

    int Afstand = (40*148) / Gemiddelde;
    return Afstand;
}
```

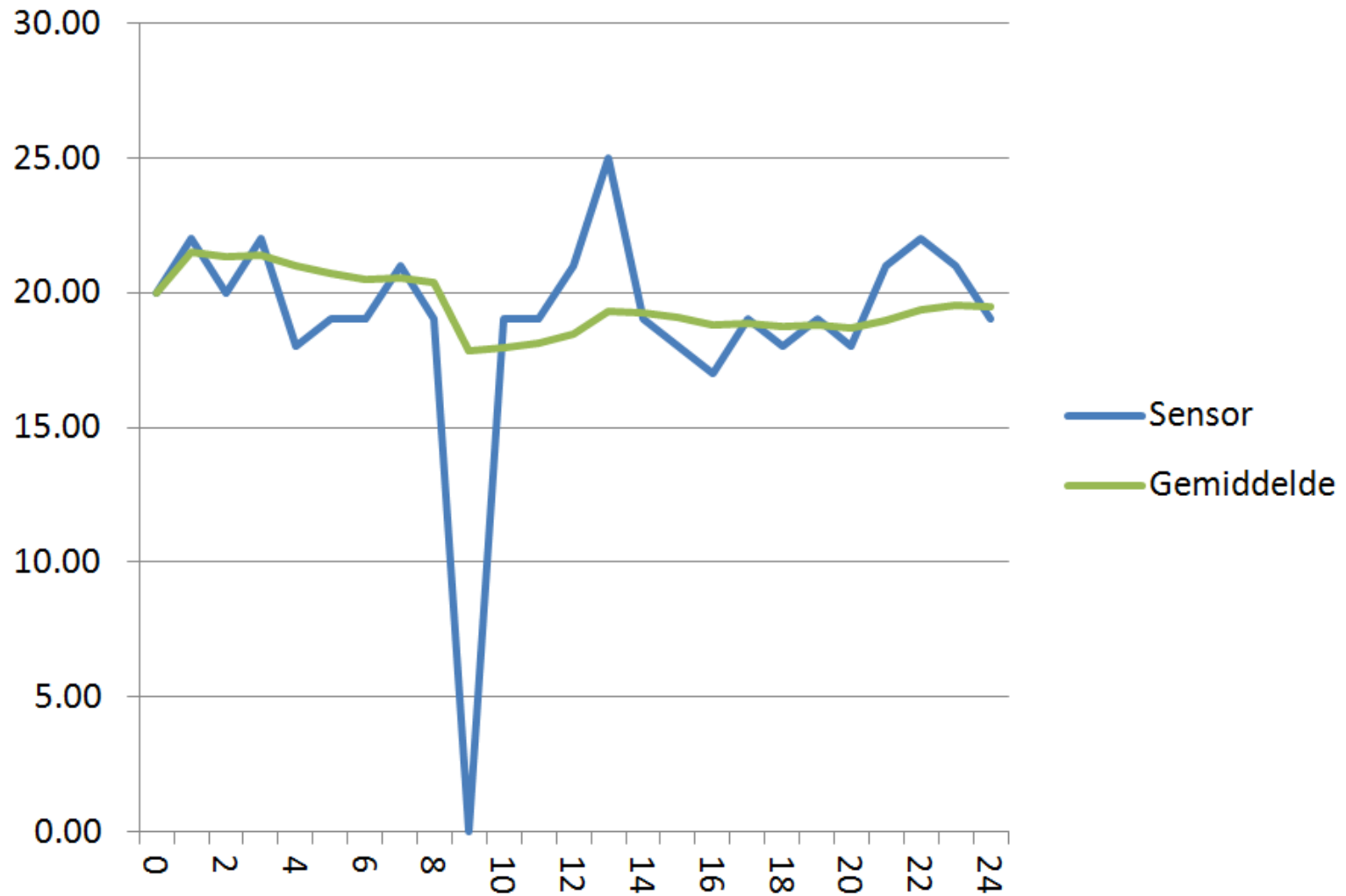
```
int Centimeters;
int LpfSharpR;

void loop() {
    static int SecTakt;

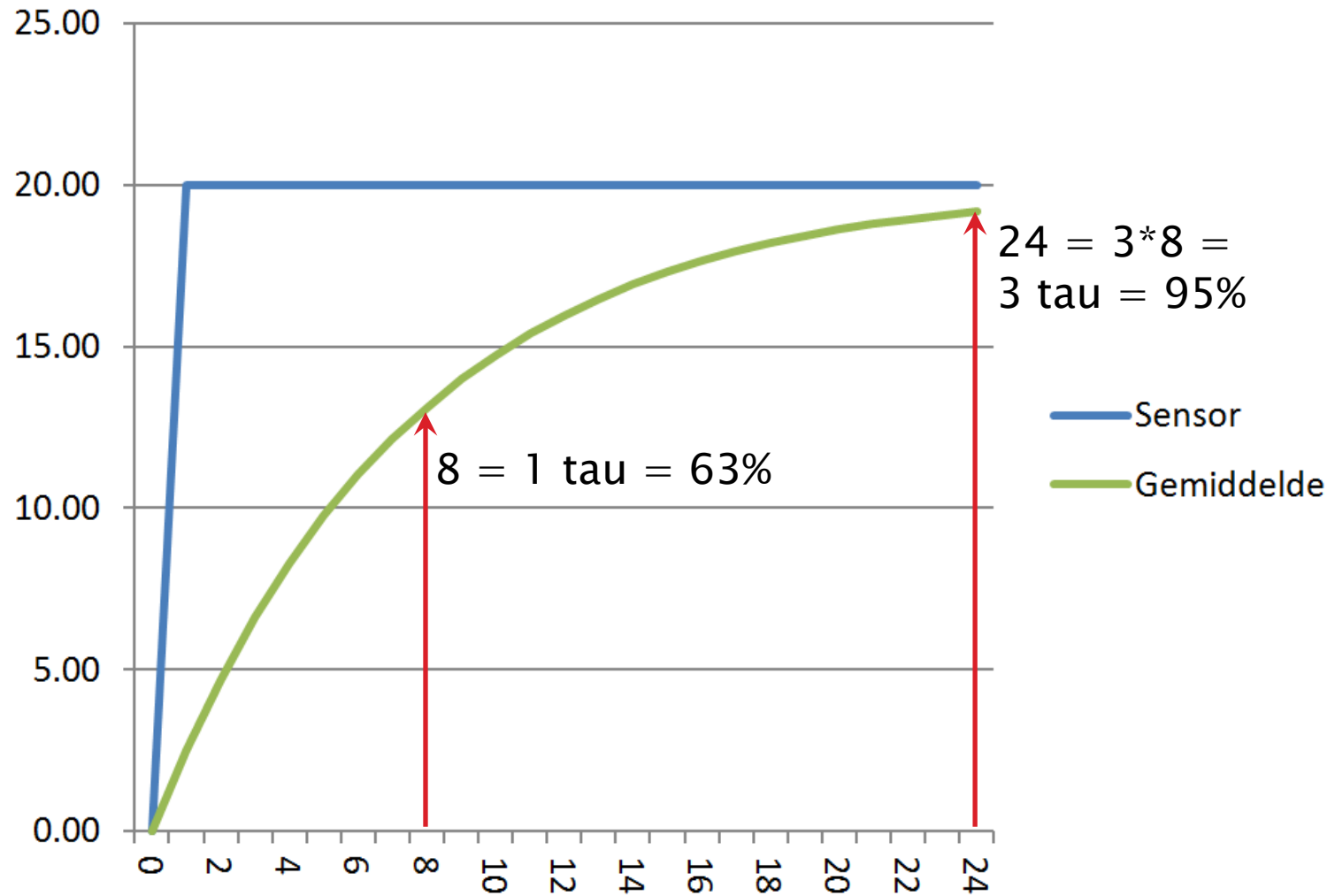
    // seconde lus
    int ms = millis();
    if ((ms - SecTakt) > 0) {
        SecTakt = ms + 1000; // zet tijd voor volgende interval

        Centimeters = SharpAfstand(A2, LpfSharpR);
        printf("Sharp: %d cm\n", Centimeters);
    }
}
```

Sensor ruis – filter output



Sensor ruis – stap response



Call by value – by Reference

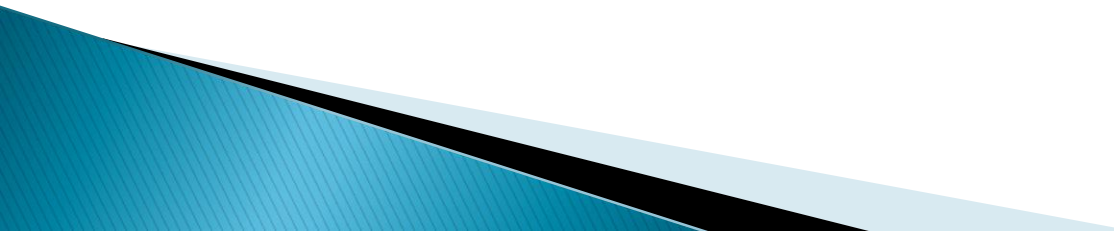
```
int SharpAfstand(int Pin, int &Middelen)
{
    int SensorValue = analogRead(Pin);

    Middelen = Middelen - Middelen / 8 + SensorValue;

    int Gemiddelde = Middelen / 8;

    int Afstand = (40*148) / Gemiddelde;
    return Afstand;
}
```

Quiz

- ▶ 5 vragen.
 - ▶ Per vraag:
 - 15 seconden bedenktijd.
 - 1 punt, als je het antwoord helemaal goed hebt.
 - ▶ Zelf je score bijhouden.
- 

Quiz V1 – Wat is de output?

```
int test = 5;
```

```
if (test == 0) printf("Nul\n");
```

```
if (test == 1) printf("Een\n");
```

```
if (test == 2) printf("Twee\n");
```

```
if (test == 3) printf("Drie\n");
```

```
if (test == 4) printf("Vier\n");
```

```
if (test == 5) printf("Vijf\n");
```

```
if (test == 6) printf("Zes\n");
```

Quiz V2 – waarom ‘oops!’ ?

Gevonden op het internet:

```
if ( ... )  
    LedAan ( ) ;  
else  
    printf ( "Calling bar ( ) " ) ;    // oops!  
    LedUit ( ) ;                      // oops!
```

Quiz V3 – Wat is de output ?

```
int Waarde = 0;
switch(Waarde) {
    case 0 : {
        printf("Nul ");
    }
    case 1 : {
        printf("Een ");
    }
    case 2 : {
        printf("Twee");
    }
    default : {
        printf("Ongeldige waarde");
    }
} // einde van switch
```

Quiz V4 – wat is het 4^e getal dat wordt geprint?

```
int Teller;  
  
void loop() {  
    TestFunctie(Teller);  
}  
  
void TestFunctie(int Teller)  
{  
    Teller++;  
    printf("Teller: %d\n", Teller);  
}
```

Quiz V5 A – Wat doet functie a()?

```
printf("Opstarten gereed.\n");

a(37, 7, 30);
}

void a(int b, int c, int d)
{ int e, f = 1; d += 2;

  while (d-->0) {
    e = b / c; b = (b - e * c) * 10;
    printf("%d", e); if (f) printf(".");
    f = 0;
  }
}
```


Quiz V5 (herkansing voor ½ punt)

```
void PrintBreuk(int Teller, int Noemer, int Decimalen)
{
    int Quotient;
    int Rest = Teller;
    bool First = true;

    Decimalen += 2;

    // Staartdeling
    while (Decimalen) {
        Quotient = Rest / Noemer;
        Rest = (Rest - Quotient * Noemer) * 10;
        printf("%d", Quotient);
        if (First) {
            // Na de eerste keer de decimale punt printen.
            printf(".");
            First = false;
        }
        Decimalen --;
    }
}
```

Quiz A1 – Wat is de output?

```
int test = 5;
```

```
if (test = 0)    printf("Nul\n");
```

```
if (test = 1)    printf("Een\n");
```

```
if (test = 2)    printf("Twee\n");
```

```
if (test = 3)    printf("Drie\n");
```

```
if (test = 4)    printf("Vier\n");
```

```
if (test = 5)    printf("Vijf\n");
```

```
if (test = 6)    printf("Zes\n");
```

Quiz A2 – waarom ‘oops!’ ?

Gevonden op het internet:

```
if( ... )  
    LedAan();  
else  
    printf( "Calling bar()" );    // oops! the else stops here  
    LedUIT();                    // oops! bar is always executed
```

Quiz A3 – Wat is de output ?

```
int Waarde = 0;
switch(Waarde) {
    case 0 : {
        printf("Nul ");
    }
    case 1 : {
        printf("Een ");
    }
    case 2 : {
        printf("Twee");
    }
    default : {
        printf("Ongeldige waarde");
    }
} // einde van switch
```

Quiz A4 – wat is het 4^e getal dat wordt geprint?

```
int Teller;

void loop() {
    TestFunctie(Teller);
}

void TestFunctie(int Teller)
{
    Teller++;
    printf("Teller: %d\n", Teller);
}
```

Quiz A5 A – Wat doet functie a()?

```
printf("Opstarten gereed.\n");

a(37, 7, 30);
}

void a(int b, int c, int d)
{ int e, f = 1; d += 2;

  while (d-->0) {
    e = b / c; b = (b - e * c) * 10;
    printf("%d", e); if (f) printf(".");
    f = 0;
  }
}
```

Quiz A5 (herkansing voor ½ punt)

```
void PrintBreuk(int Teller, int Noemer, int Decimalen)
{
    int Quotient;
    int Rest = Teller;
    bool First = true;

    Decimalen += 2;

    // Staartdeling
    while (Decimalen) {
        Quotient = Rest / Noemer;
        Rest = (Rest - Quotient * Noemer) * 10;
        printf("%d", Quotient);
        if (First) {
            // Na de eerste keer de decimale punt printen.
            printf(".");
            First = false;
        }
        Decimalen --;
    }
}
```


The image shows a close-up of a yellow warning sign with black diagonal stripes. The word "WARNING" is printed in a large, bold, black serif font in the center. The sign has a weathered, textured appearance with some visible scratches and discoloration.

WARNING

Bitwise & Logical operators

C Logical			&&	
C Bitwise			&	
A	B		AND	OR
0	0		0	0
0	1		0	1
1	0		0	1
1	1		1	1

Bits in een byte

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	128	64	32	16	8	4	2	1	

Bits in een byte – 41

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	128	64	32	16	8	4	2	1	
A	0	0	1	0	1	0	0	1	
	0	0	32	0	8	0	0	1	41

Bits in een byte – 41 en 18

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	128	64	32	16	8	4	2	1	
A	0	0	1	0	1	0	0	1	
	0	0	32	0	8	0	0	1	41
B	0	0	0	1	0	0	1	0	
	0	0	0	16	0	0	2	0	18

Bitwise AND – 41 en 27

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
	128	64	32	16	8	4	2	1	
A	0	0	1	0	1	0	0	1	
	0	0	32	0	8	0	0	1	41
C	0	0	0	1	1	0	1	1	
	0	0	0	16	8	0	2	1	27
A & C	0	0	0	0	1	0	0	1	
	0	0	0	0	8	0	0	1	9

Logical operators (&&, ||)

- ▶ Werken met
 - WAAR (true, variabele ongelijk 0) en
 - NIET_WAAR (false, variabele is 0)
- ▶ Resultaat is WAAR (1) of NIET_WAAR (0)
- ▶ Logical operators worden (bijvoorbeeld) gebruikt om een aantal voorwaarden (if's) samen te voegen.
- ▶ Bitwise operators worden gebruikt als je specifieke bit(s) wilt testen.

Logical operators (&&, ||)

```
if (A != 0) {  
    if (B > 55) {  
        // doe iets  
    }  
}
```

```
if (A != 0) {  
    // doe iets  
}  
if (B > 55) {  
    // doe iets  
}
```

```
if ((A != 0) && (B > 55)) {  
    // doe iets  
}
```

```
if ((A != 0) || (B > 55)) {  
    // doe iets  
}
```


Voorspel de output

► Les_7_p20_and.ino

```
int A = 41;
```

```
int B = 18;
```

```
int C = 27;
```

```
printf("%d & %d = %d\n", A, B, A & B);
```

```
printf("%d && %d = %d\n", A, B, A && B);
```

```
printf("%d & %d = %d\n", A, C, A & C);
```

```
printf("%d && %d = %d\n", A, C, A && C);
```

Resultaat –logical / bitwise

```
int A = 41;
```

```
int B = 18;
```

```
int C = 27;
```

```
printf("%d & %d = %d\n", A, B, A & B);
```

```
printf("%d && %d = %d\n", A, B, A && B);
```

```
printf("%d & %d = %d\n", A, C, A & C);
```

```
printf("%d && %d = %d\n", A, C, A && C);
```

Opstarten gereed.

41 & 18 = 0

41 && 18 = 1

41 & 27 = 9

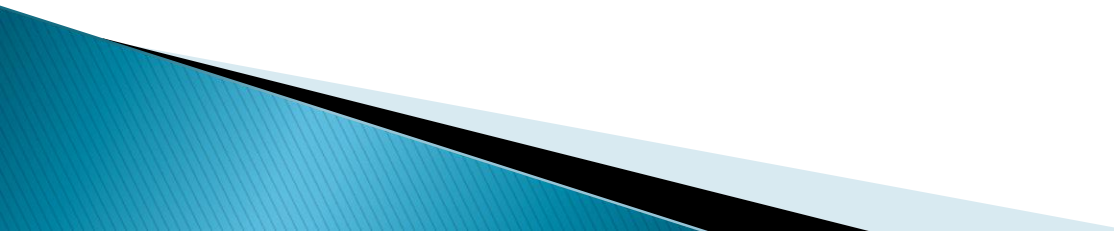
41 && 27 = 1

Logical operators samenvatting

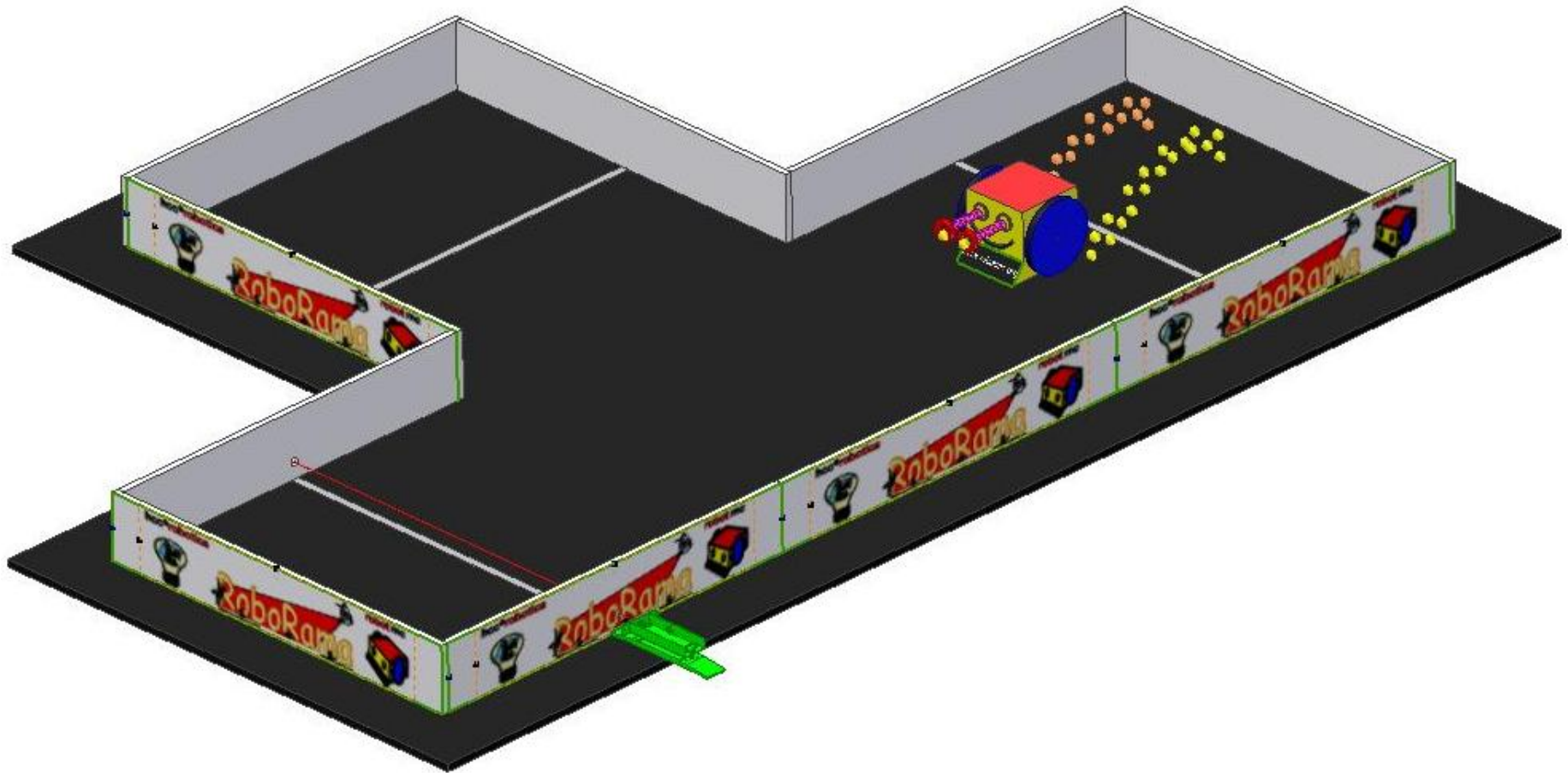
&, |

- ▶ Bitwise
- ▶ Als je bepaalde bit's nodig hebt.

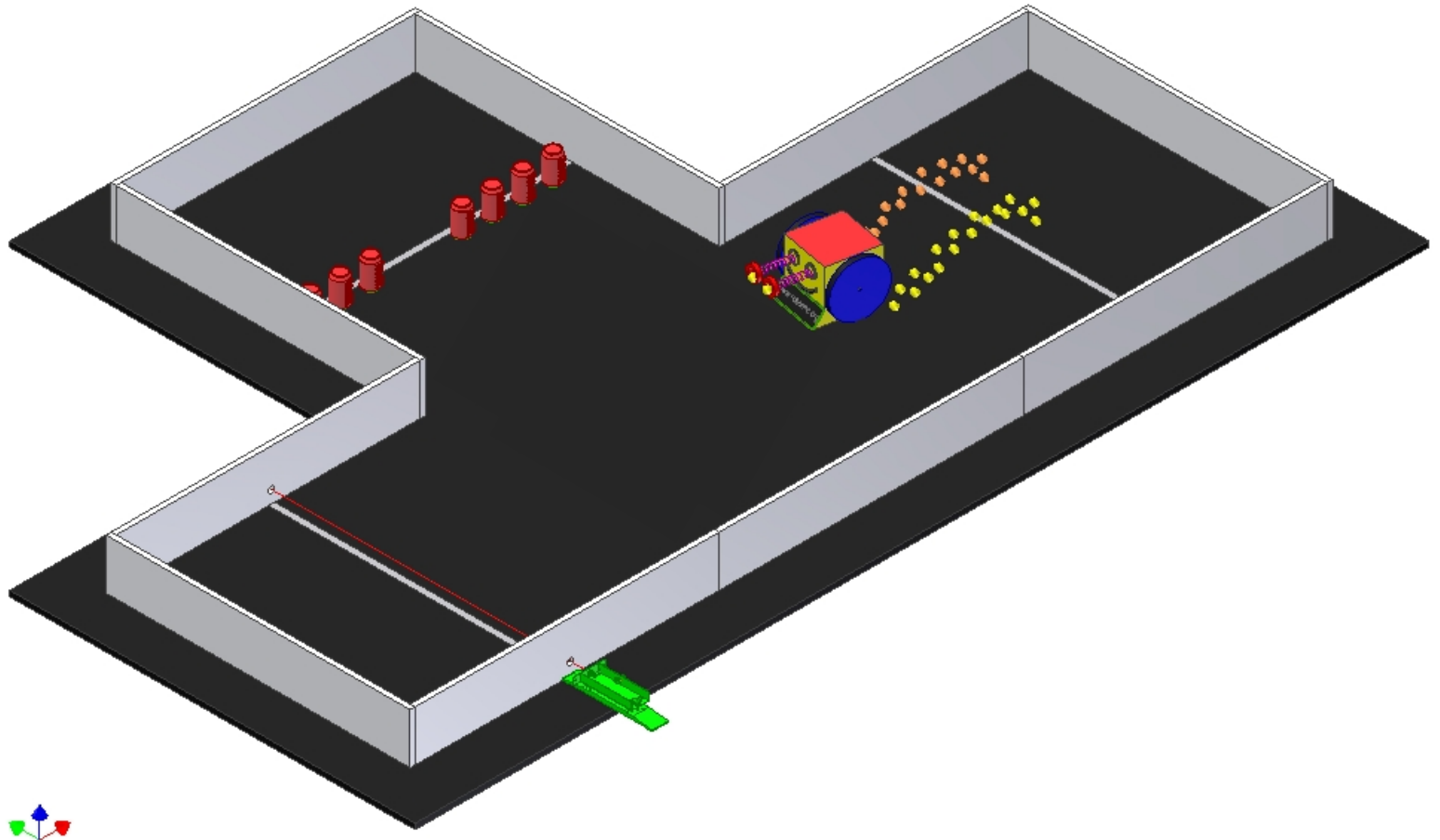
&&, ||

- ▶ Logical
 - ▶ O.a. bij 'combineren' if-statements.
 - ▶ Geeft true (1) of false (0)
- 

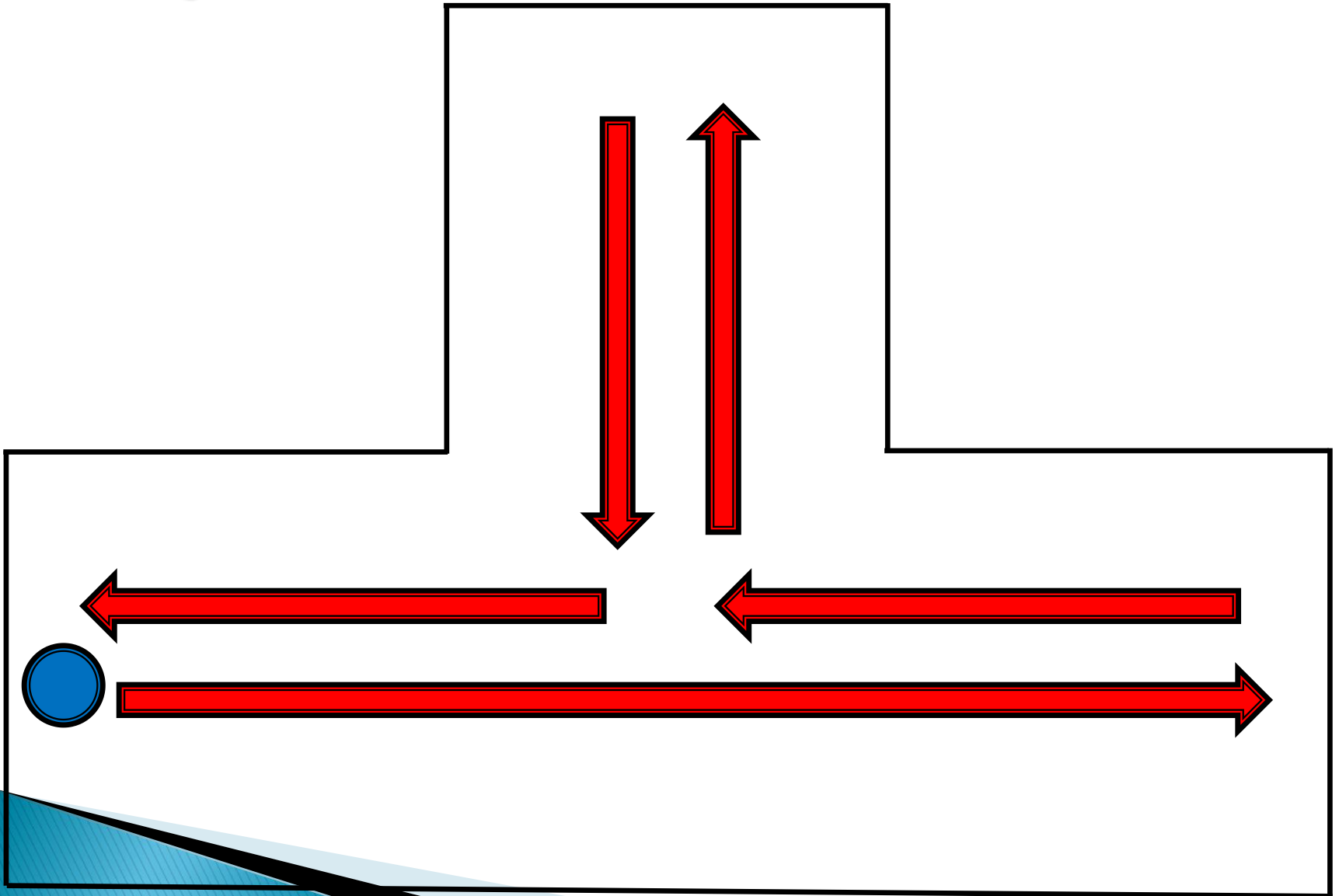
Roborama T-tijd



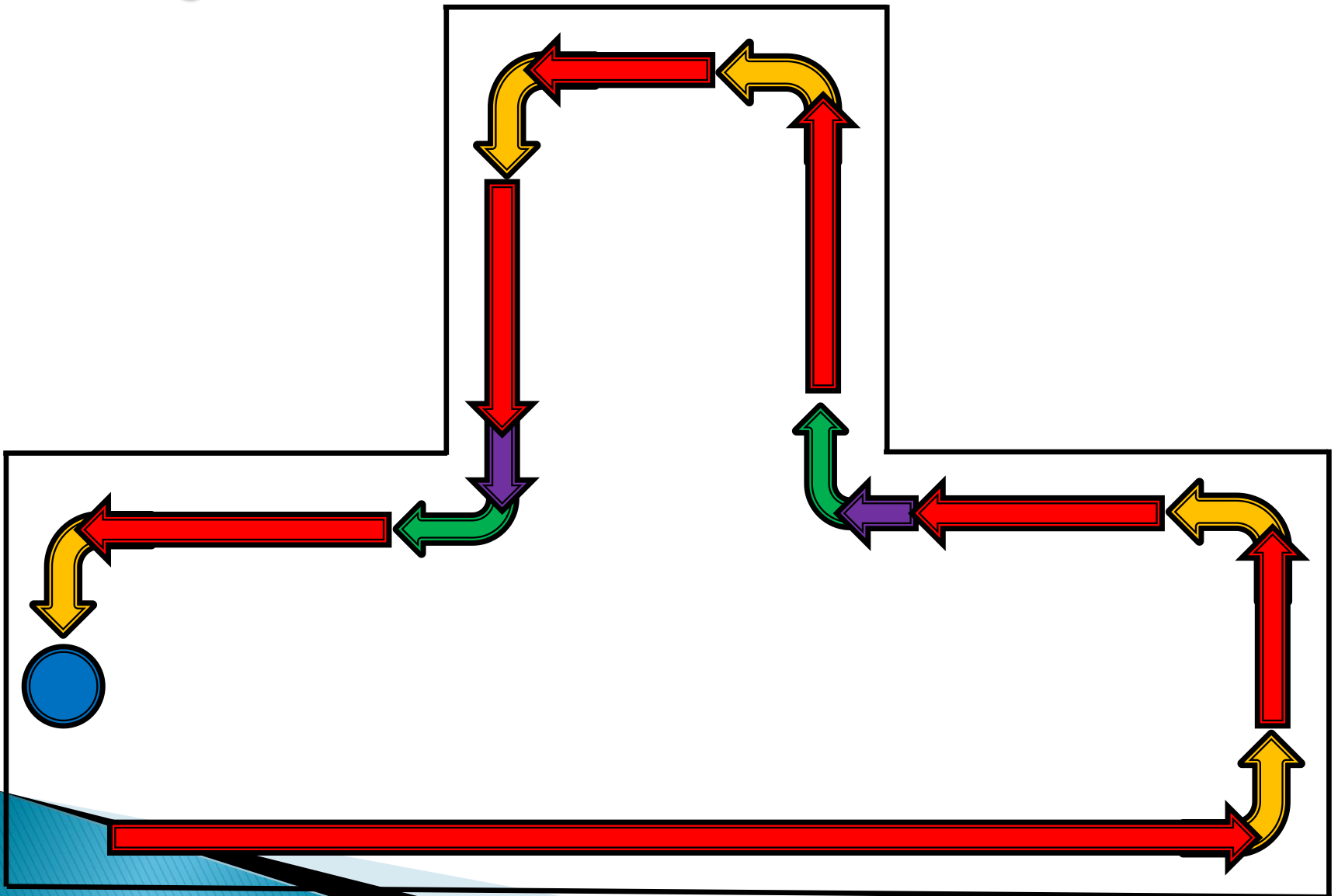
Roborama T-tijd Bonus 2



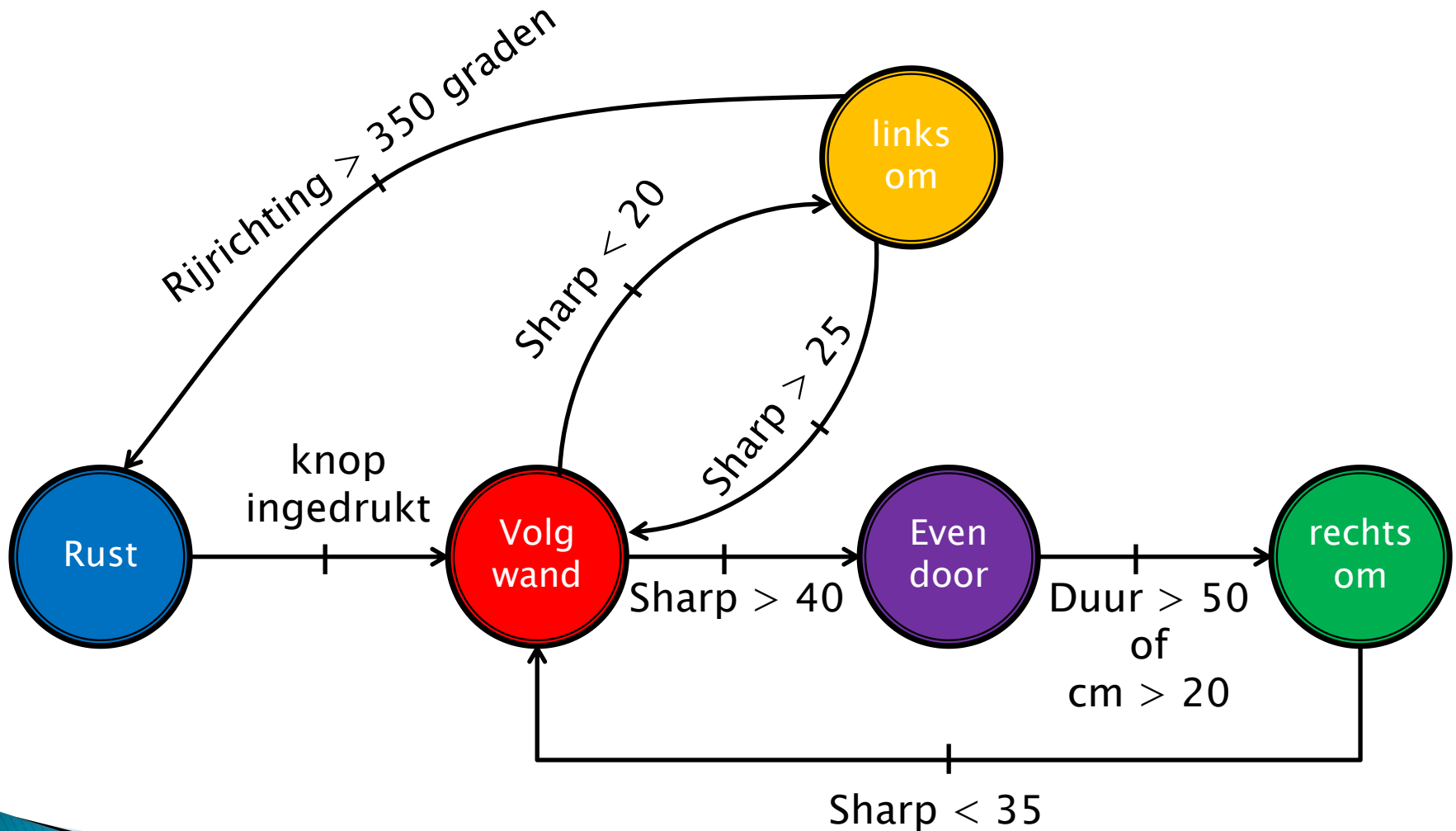
T-Tijd route



T-Tijd states

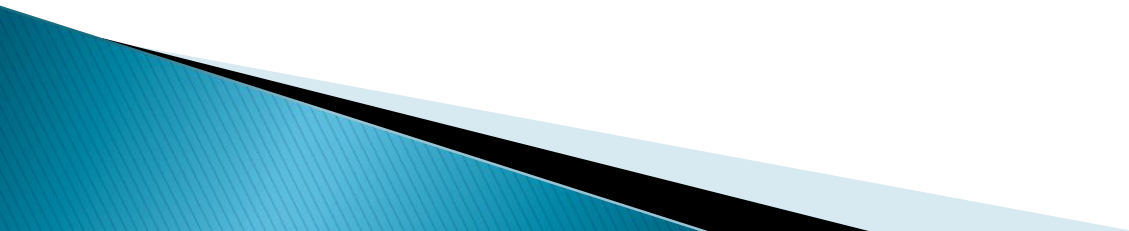


T-Tijd state machine



Ttijd Bonus 2 (Smalle doorgang)

Ideeën?



Huiswerk: roborama ttijd

1. Ttijd (zonder bonus)
2. Ttijd Bonus 2 (Smalle doorgang) – Ideeën?

Komende keer examen – **zaterdag 1 november.**