

LOGISCHE FUNCTIES: STRUCTUUR

Elke RLM heeft een And(En), Or (OF) en een Loop functie aan boord.

We proberen dit hier uit te leggen aan de hand van een schematische voorstelling:

Je moet deze module (RLM) bekijken als een blad.

Het bovenste gedeelte wordt gebruikt voor berekeningen die de standaard werking van de functie beheren.

Dat is dan bv het in- en uitschakelen, de timerfunctie, rolluikfunctie...

Je ziet dat er erg weinig ruimte van de totale pagina wordt gebruikt.

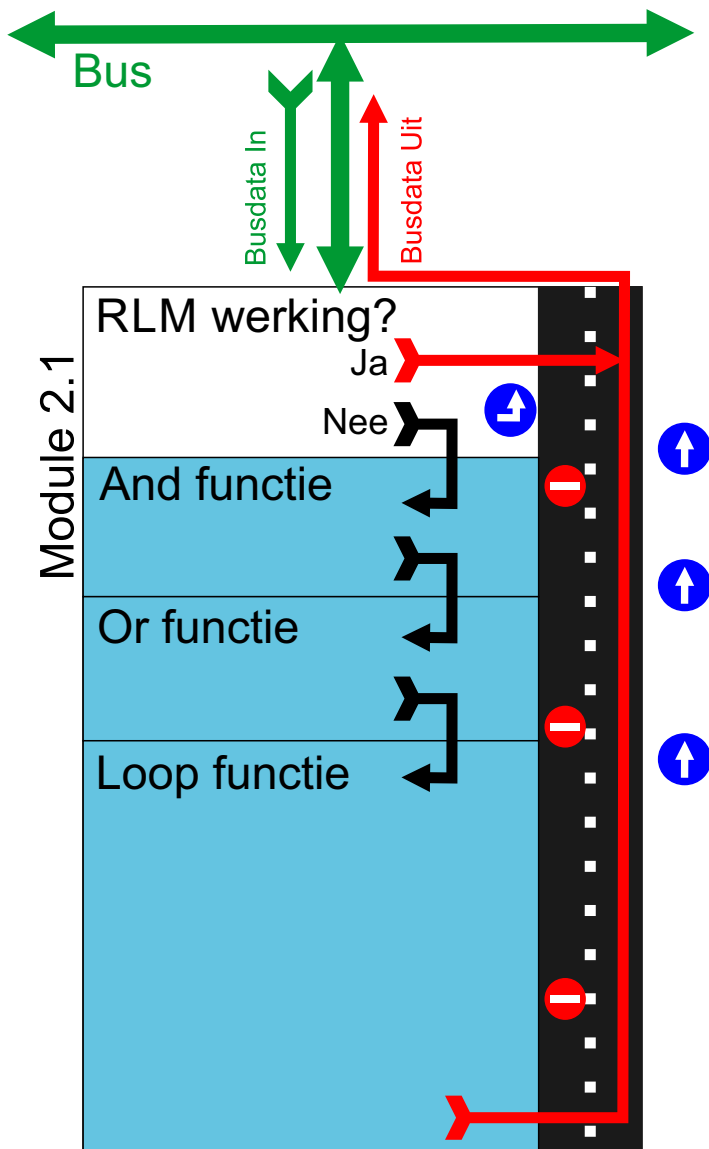
Daarom hebben we besloten om de rest van de pagina vrij te geven voor logische functies.

De RLM wordt dus de host (gastheer) van de logische functies.

Dit betekent:

- dat we op de vrije ruimte van die bewuste pagina schrijven, maar de standaard werking niet beïnvloeden.
- dat de logische functie onafhankelijk is van de toestand van de RLM. (Aan, uit, timer,...)

Merk op: Als de RLM rechtstreeks aangestuurd wordt (standaard werking) worden de logische functies overgeslagen. Als de logische functie berekend wordt, gaat de informatie rechtstreeks op de bus, en wordt niet meer gekeken naar het gedeelte van de standaardwerking.



Bij het gebruik van logische functies is er altijd 1 belangrijke beperking: je kan logische functies gebruiken naar eigen wens, maar een module die geschreven wordt in bv module 2.1, kan nooit deze module aansturen, of de toestand ervan bekijken.

Medio 2016 zal de software deze zin geautomatiseerd worden en hoef je hier zelf geen rekening meer mee te houden.

LOGISCHE FUNCTIES: AND (EN)

1. Wat is een logische And (en)?

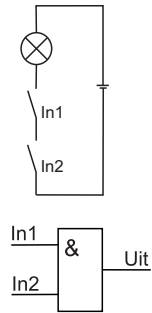
Als je een uitgang (lamp, rolluik,...) automatisch wil aansturen, kan je dit doen aan de hand van een aantal voorwaarden (licht, beweging,...)

Stel: Je wil je opritverlichting automatiseren. Als het donker is, mag ze gaan branden.

Ze moet niet de ganse nacht branden. Tussen 23u en 7u hoeft ze niet te branden.

In de waarheidstabel zie je dat alle ingangen waar moeten zijn om de uitgang ook op waar te zetten.

Elektrisch



Waarheidstabel

In1	In2	Uit
0	0	0
1	0	0
0	1	0
1	1	1

Waar/onwaar

In1	In2	Uitgang
Onwaar	Onwaar	Onwaar
Waar	Onwaar	Onwaar
Onwaar	Waar	Onwaar
Waar	Waar	Waar

Volgens voorbeeld

In1	In2	Uitgang
Licht	23u-7u	Licht uit
Donker	23u-7u	Licht uit
Licht	7u-23u	Licht uit
Donker	7u-23u	Licht aan

2. Het voorbeeld in detail

Ingang1: donker / niet donker

Ingang2: 7-23u : 23-7u

Uitgang: Opritverlichting

De verlichting brandt enkel als het tussen 7 en 23u is én als het donker is.

In alle andere gevallen brandt ze niet.

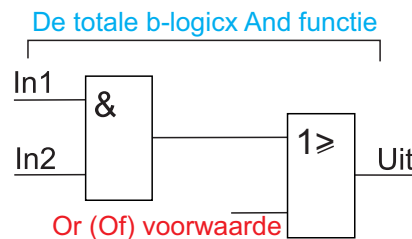
Kortom: En ingang 1 is waar EN ingang 2 is waar om de uitgang te activeren.

3. De B-Logicx And functie

Hier gaan we nog een stap verder en voegen we een Or (Of) voorwaarde toe.

Als je het zuiver logisch bekijkt, kan je zeggen dat een And functie gecombineerd wordt met een Or functie.

In1	In2	Inv	Uit
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1
0	0	1	1
1	0	1	1
0	1	1	1
1	1	1	1



Als de inverse ingang waar is, wordt de toestand van de andere ingangen niet meer bekeken en zal de and functie overbrugd worden. Op het moment dat de inverse voorwaarde onwaar wordt, zal rekening gehouden worden met de toestand van de andere ingangen.

Waarom doen we dit?

In bovenstaand voorbeeld is het niet ondenkbaar dat je de opritverlichting met een drukknop ook nog manueel wil aandoen. De inverse voorwaarde wordt dan bediend door de drukknop.

LOGISCHE FUNCTIES: OR (OF)

1. Wat is een logische Or (Of)?

Als je een uitgang (lamp, rolluik,...) automatisch wil aansturen, kan je dit doen aan de hand van een aantal voorwaarden (licht, beweging,...)

Stel: Je wil de nachthal automatiseren.

2 bewegingsmelders (of drukknoppen) schakelen de verlichting aan en uit.

In de waarheidstabel zie je dat alle ingangen waar moeten zijn om de uitgang ook op waar te zetten.

Elektrisch	Waarheidstabel	Waar/onwaar	Volgens voorbeeld																																													
	<table border="1"> <thead> <tr> <th>In1</th> <th>In2</th> <th>Uit</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	In1	In2	Uit	0	0	0	1	0	1	0	1	1	1	1	1	<table border="1"> <thead> <tr> <th>In1</th> <th>In2</th> <th>Uitgang</th> </tr> </thead> <tbody> <tr><td>Onwaar</td><td>Onwaar</td><td>Onwaar</td></tr> <tr><td>Waar</td><td>Onwaar</td><td>Waar</td></tr> <tr><td>Onwaar</td><td>Waar</td><td>Waar</td></tr> <tr><td>Waar</td><td>Waar</td><td>Waar</td></tr> </tbody> </table>	In1	In2	Uitgang	Onwaar	Onwaar	Onwaar	Waar	Onwaar	Waar	Onwaar	Waar	Waar	Waar	Waar	Waar	<table border="1"> <thead> <tr> <th>In1</th> <th>In2</th> <th>Uitgang</th> </tr> </thead> <tbody> <tr><td>BM1</td><td>BM2</td><td>Licht uit</td></tr> <tr><td>BM1</td><td>BM2</td><td>Licht aan</td></tr> <tr><td>BM1</td><td>BM2</td><td>Licht aan</td></tr> <tr><td>BM1</td><td>BM2</td><td>Licht aan</td></tr> </tbody> </table>	In1	In2	Uitgang	BM1	BM2	Licht uit	BM1	BM2	Licht aan	BM1	BM2	Licht aan	BM1	BM2	Licht aan
In1	In2	Uit																																														
0	0	0																																														
1	0	1																																														
0	1	1																																														
1	1	1																																														
In1	In2	Uitgang																																														
Onwaar	Onwaar	Onwaar																																														
Waar	Onwaar	Waar																																														
Onwaar	Waar	Waar																																														
Waar	Waar	Waar																																														
In1	In2	Uitgang																																														
BM1	BM2	Licht uit																																														
BM1	BM2	Licht aan																																														
BM1	BM2	Licht aan																																														
BM1	BM2	Licht aan																																														

2. Het voorbeeld in detail

Ingang1: Bewegingsmelder 1 actief

Ingang2: Bewegingsmelder 2 actief

Uitgang: Nachthal verlichting

De verlichting brandt als minstens 1 van de 2 bewegingsmelders iets ziet.

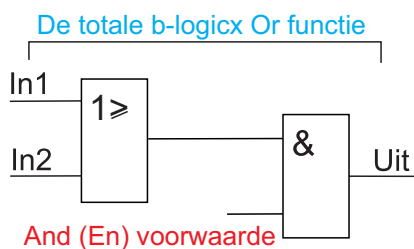
Pas als alle bewegingsmelders in rust zijn, gaat het licht terug uit.

Kortom: OF ingang 1 is waar OF ingang 2 is waar om de uitgang te activeren.

3. De B-Logicx And functie

Hier gaan we nog een stap verder en voegen we een And (En) voorwaarde toe.

Als je het zuiver logisch bekijkt, kan je zeggen dat een Or functie gecombineerd wordt met een And functie.



In1	In2	Inv	Uit
0	0	0	0
1	0	0	1
0	1	0	1
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

Als de And voorwaarde waar is, wordt de toestand van de andere ingangen niet meer bekeken en zal de totale Or functie geblokkeerd worden. De uitgang zal dan ook uitgestuurd worden.

Op het moment dat de inverse voorwaarde onwaar wordt, zal rekening gehouden worden met de toestand van de andere ingangen.

Waarom doen we dit?

In bovenstaand voorbeeld is het niet ondenkbaar dat je de nachthal verlichting niet wil laten branden als er al iemand slaapt.

Op dat moment an je de And voorwaarde activeren via een drukknop of via het alarm.

De verlichting zal dan pas terug branden als het alarm uitgeschakeld is, of als de drukknop terug bediend wordt.

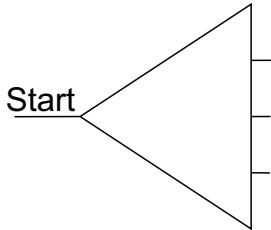
LOGISCHE FUNCTIES: LOOP

1. Wat is een Loop?

De loop is een veel gebruikt logische functie.

Het perfecte voorbeeld om dit te verduidelijken is een 'alles uit'.

Je drukt op een knop en verlichting, stopcontacten, ... worden uitgeschakeld.



De loop is als het ware een lijstje dat van boven naar onderen wordt afgewerkt.

Je gebruikt gewoon het aantal lijnen die je wil gebruiken, de rest laat je leeg.

80 adressen na elkaar verzenden is mogelijk.

Theoretisch is het mogelijk om die alle 80 direct na elkaar te verzenden,.

Elektrisch is dit onverantwoord, kwestie van pieken op het net te voorkomen.

Daarom kan je dus ook onbeperkt pauzes invoegen.

Dit kan in minuten of seconden zijn.

Als je 5 uitgangen uitschakelt, dan een seconde wacht, weer 5 uitgangen uitschakelt, zit je ruim goed.

2. De loop activeren

A. Standaard wordt de loop 'Enkel' uitgevoerd.

- Dat wil zeggen dat hij bovenaan start, alle lijnen afwerkt tot hij beneden is en dan stopt.
- Je hoeft hem niet zelf te stoppen.
- Bij ieder startcommando start hij altijd opnieuw bovenaan.
- Je kan hem stoppen met de functie 'Reset'.

B. De loop kan ook 'Continu' uitgevoerd worden.

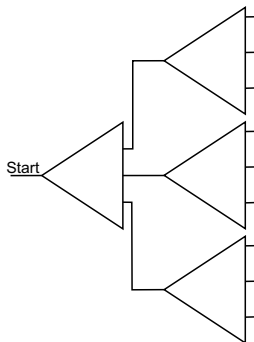
- Dat wil zeggen dat hij bovenaan start, alle lijnen afwerkt tot hij beneden is, en dan opnieuw begint.
- Bij ieder startcommando herbegint hij altijd opnieuw bovenaan.
- Je **moet** hem stoppen met de functie 'Reset'.

Na het aanmaken van een loop kan je zelf beslissen hoe je hem start.

Zo kan je de 'Alles uit' dagelijks meerdere keren manueel of automatisch starten.

Als je op vakantie vertrekt, kan je hem, met tussenpozen, quasi continu laten lopen.

Mocht een buur of familielid, tijdens een controle, de verwarming per ongeluk aanzetten of de verlichting laten branden, dan zal deze functie dit rechtzetten.



De ene loop kan ook een andere starten.

Het is daarom ook handig om de 'alles uit' te verdelen in kleinere stukken.

Je maakt er een voor het gelijkvloers, de verdieping, de kelder, buiten,...

Een aparte loop kan dan eerst de verdieping, dan het gelijkvloers uitschakelen, enz...

Selecteer 'Set' van 'alles uit verdieping', ipv rechtstreeks een Member aan te spreken.

Stel dat de 'alles uit verdieping' 10 sec loopt, zet dan een pauze van 15 seconden in de volgende lijn.

Selecteer 'Set' van 'alles uit gelijkvloers', Als deze 20 seconden loopt, zet dan een pauze van 25 seconden in de volgende lijn, enz...

Tip:

Bij het aanmaken van een alles uit functie raden we aan om de verlichting aan de toegangsdeuren als laatste te schakelen.

M.a.w: je schakelt eerst de verlichting, die centraler gelegen in de woning is, uit, dan zet je een 'Pauzetijd' van ongeveer 15 seconden, en pas dan schakel je de verlichting in de buurt van de toegangsdeuren uit.

Op deze manier kan je de 'Alles uit' starten, en heb je toch nog de tijd om de deuren te sluiten, en kan je veilig de woning verlaten zonder in het donker overal tegen aan te lopen.