Bind

f

Match

f1

Fold

f2

f3

f1

f1

Map

function

function

success

failure

fun. 1

success

failure

fun. 2

success

failure

Connect?

fun. 1

success

failure

fun. 2

success

failure

*on success*

*bypass*

Bind

f

Bind

fun. 2

fun. 1

success

failure

success

failure

Bind

fun. 2

fun. 1

success

failure

success

failure

Bind

fun. 3

success

failure

function

Failure

Success

function

Presentation Layer

Domain Layer

Persistence

Presentation Layer

Domain Layer

Persistence

Infrastructure

Infrastructure

Presentation functions

Domain functions

Persistence impl.

Infrastructure

Presentation types

Domain types

Persistence interfaces

Presentation functions

Domain functions

Persistence impl.

Infrastructure

Presentation types

Domain types

Persistence interfaces

Validation.fs

CustomerRepository.fs

STMP.fs

UITypes.fs

DomainTypes.fs

DbTypes.fs

SendEmail.fs

UserInterface.fs

UseCases.fs

Main.fs

Logger.fs

InfrastructureTypes.fs

Receive request

Validate request

Read existing user record

Update existing user record

Send verification email if changed

Show result

Receive request

Validate request

Read existing user record

Update existing user record

Send verification email if changed

Show result

Validation error

Db error

SMTP error

Not found

Request

Errors

Response

Response

Request

Validate

Update

Send

Errors

Output

Success

Input

A function representing   
the use case.

Output

A function

Input

The function representing   
the use case.

Input

Success

Failure

Output

The function representing   
the use case.

Input

Success

Failure

Output

Success

Errors

Output

Validate

Update

Send

The function representing   
the use case.

Input

Response

Success

Request

Failure

Validate

Output

Success

Failure

Input

Validate

Input

Success

Failure

Output

Validate

Update

Send

Errors

Output

Success

Input

The use case function

Validate

Update

Send

Success

Errors

Validate

Update

Send

Validate

Update

Send

Validate

Update

Send

Response

Success

Request

Failure

Validate

Output

Success

Failure

Input

Validate

Input

Success

Failure

Output

Validate

Success

Failure

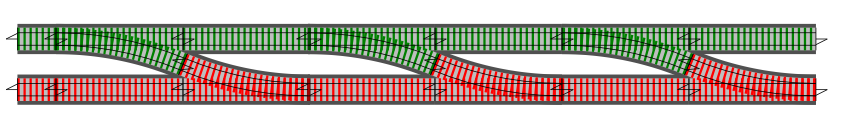
Update

Success

Failure

*on success*

*bypass*



Validate function

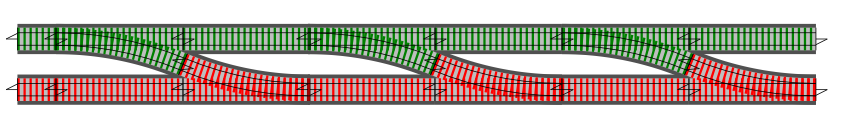
Send function

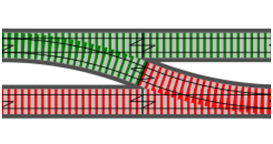
Update function

Validate function

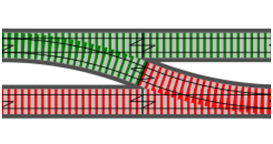
Send function

Update function

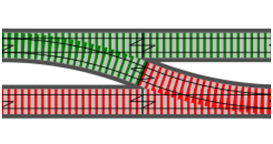




Validate function



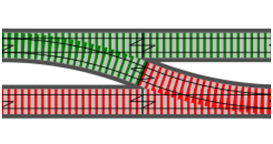
Update function



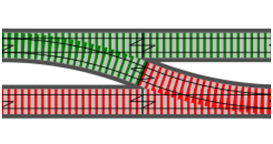
Send function

>>

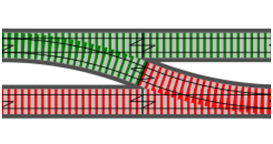
>>



Validate function



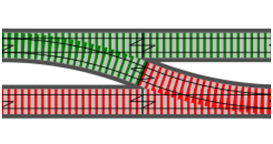
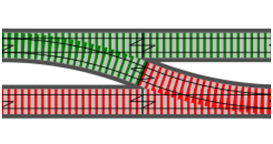
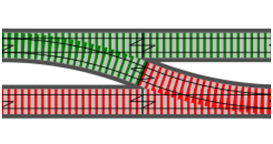
Update function



Send function

>>

>>



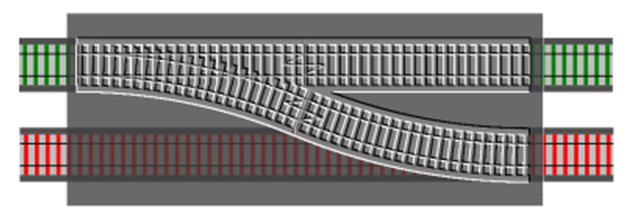
Send function

🗶

Validate function

Update function

🗶

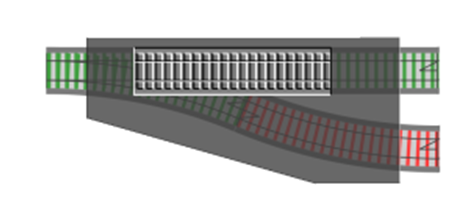


Two-track input

Slot for switch function

Adapter block

Two-track output

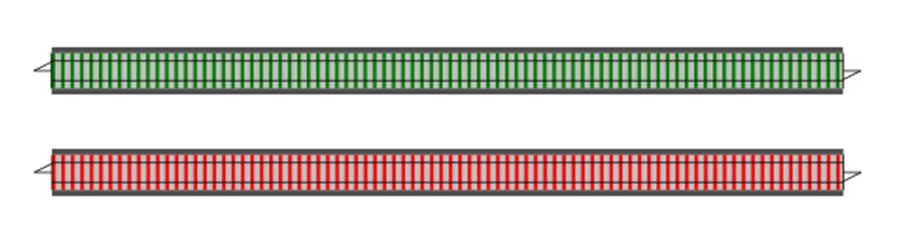


One-track input

Two-track output

Slot for one-track function

Adapter block



?



bind



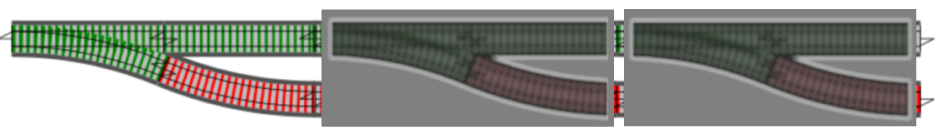
?

compose

Validate1 switch (no bind)

Validate3’ (after bind)

Validate2’ (after bind)



Validate2 switch

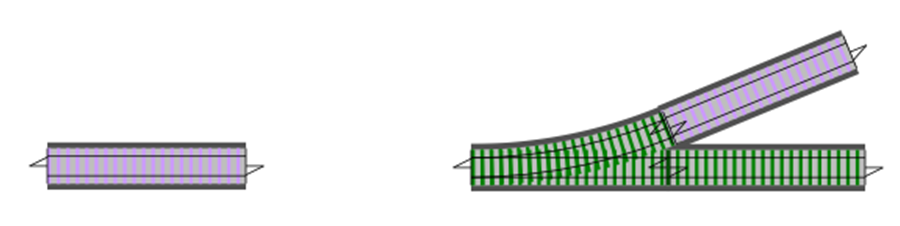
Validate3 switch

Switch function

Simple function



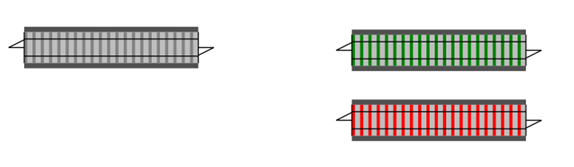
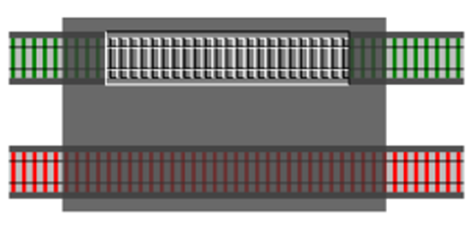
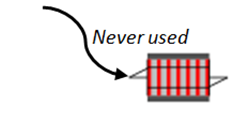
*Never used*



Dead end

Main flow

Dead end function



Two-track function

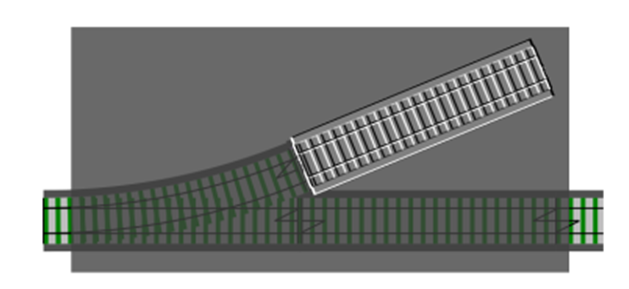
Simple function

Two-track input

Two-track output

Slot for one-track function

Adapter block

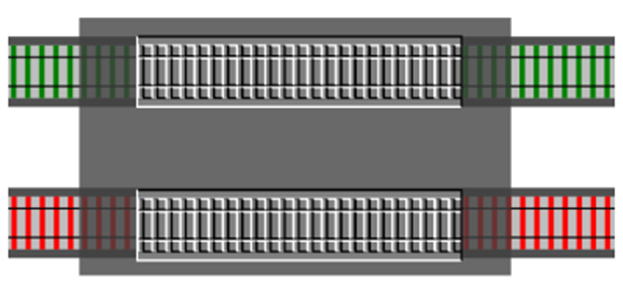


One-track input

One-track output

Slot for dead-end function

Adapter block



Two-track input

Two-track output

Slot for one-track success function

Adapter block

Slot for one-track failure function



then



Split input

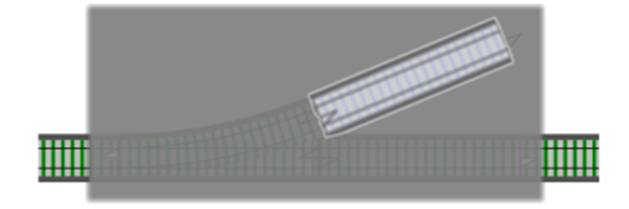
Combine results

Success

Failure

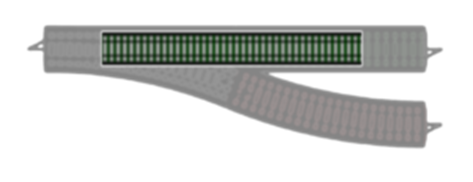


+



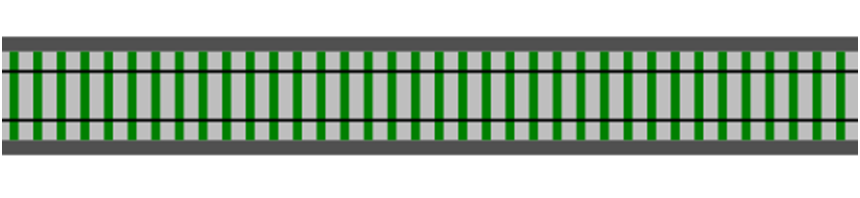
Slot for dead-end function

Pass through adapter block

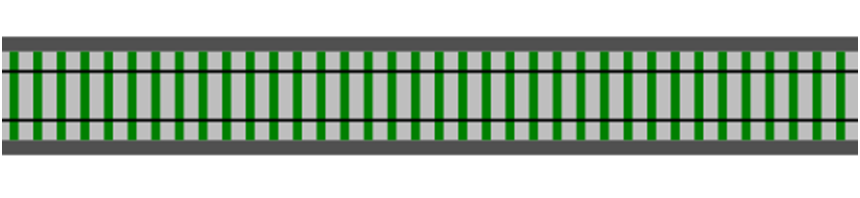


Slot for simple function

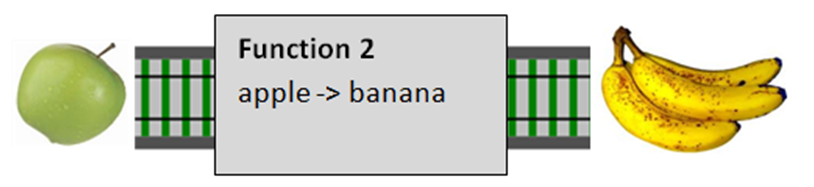
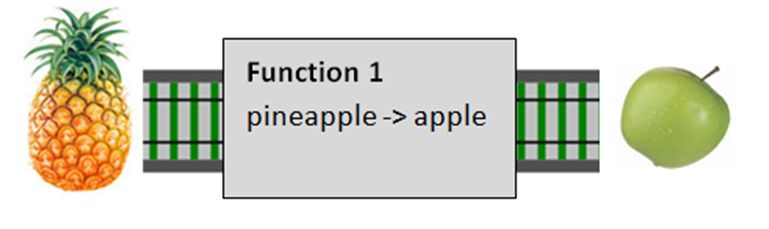
Switch adapter block



**Function 1**  
pineapple -> apple



**Function 2**  
apple -> banana



>>

High level Layer

Middle layer

Low Level layer

High level Layer

Middle layer

Low Level layer

High level Layer

Middle layer

Low Level layer

High level Layer

Middle layer

Low Level layer

High level Layer

Middle layer

Low Level layer

High level Layer

Middle layer

Low Level layer

New “super component”

Higher level components

Lower level components

Use case

Validation/transformation  
into domain objects

Ugly, untrusted world

Transformation from   
domain into outside world

Ugly, untrusted world

A state

C state

B state

*X event*

*X event*

*(ignore)*

*Y event*

*(ignore)*

*Z event*

*Z event*

*X event*

*Y event*

Empty

Accumulate  
Digits

*Digit*

*Clear*

*Digit*

Pending Op

*Op*

Accumulate  
Digits

*Digit*

Show Result

*Equalss*

*Digit*

*Op*

Show Error

Empty

Accumulate  
Digits

*NonZero  
Digit*

*Clear*

*Digit*

Pending Op

*Op*

Accumulate  
Digits

*Digit*

Show Result

*Equalss*

*Digit*

Accumulate  
Digits with decimal

*Separator*

*Digit*

*Op*

*Separator*

*(ignore)*

*Zero*

*(ignore)*

*Op*

Show Error

Zero

Accumulate  
Digits

*NonZero  
Digit*

*Clear*

*Digit*

Compute  
(w/pending)

*Op*

Accumulate  
Digits

*Digit*

Compute  
(no pending)

*Equalss*

*NonZero  
Digit*

Accumulate  
Digits with decimal

*Separator*

*Digit*

*Op*

*Separator*

*(ignore)*

*Zero*

*(ignore)*

*Op*

Show Error

*Zero*

Start

Start

Start

Start