



# ***Project Documentation foc\_closed\_loop***

January 26, 2016

© Linz Center of Mechatronics GmbH

# Contents

<b>I</b>	<b>X2C Model</b>	<b>2</b>
<b>1</b>	<b>Model Structure</b>	<b>2</b>
1.1	Xcos Model . . . . .	2
1.2	Subsystems . . . . .	2
<b>2</b>	<b>Model Parameter</b>	<b>3</b>
2.1	Sample Time . . . . .	3
<b>3</b>	<b>Mask Parameter</b>	<b>4</b>
<b>II</b>	<b>Used X2C-Blocks</b>	<b>8</b>
<b>4</b>	<b>Project Specific Blocks</b>	<b>8</b>
<b>5</b>	<b>Internal Library Blocks</b>	<b>8</b>
	Add . . . . .	8
	AutoSwitch . . . . .	11
	Constant . . . . .	15
	Gain . . . . .	18
	Maximum . . . . .	21
	Minimum . . . . .	23
	Negation . . . . .	25
	PI . . . . .	28
	Sub . . . . .	33
	TypeConv . . . . .	36
	clark . . . . .	39
	clark_inv . . . . .	40
	park . . . . .	41
	park_inv . . . . .	43
	reset_qei . . . . .	44
	timer . . . . .	45

# Part I

## X2C Model

### 1 Model Structure

#### 1.1 Xcos Model

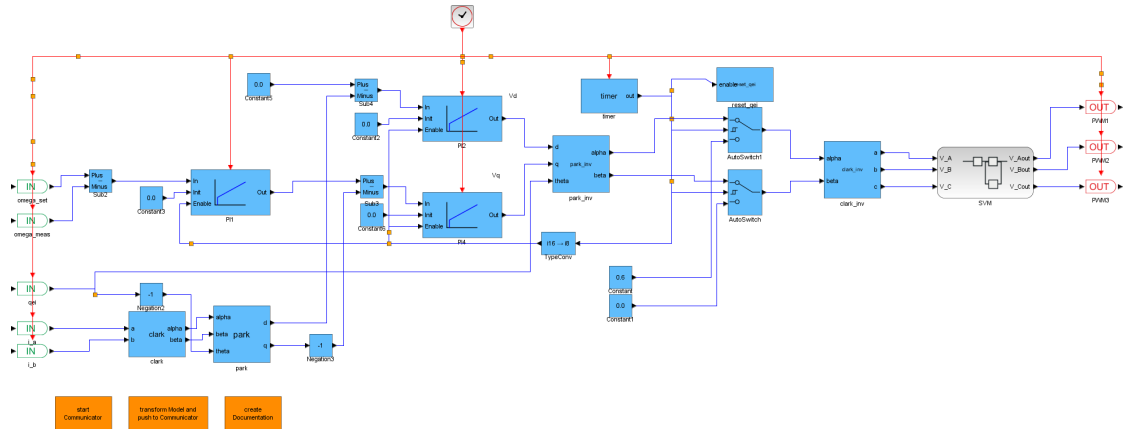


Figure 1: foc\_closed\_loop

#### 1.2 Subsystems

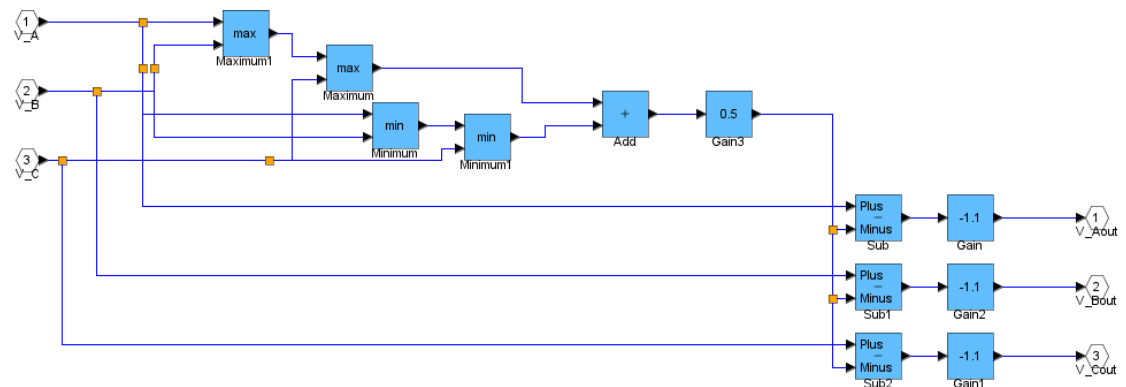


Figure 2: foc\_closed\_loop\_SVM

## 2 Model Parameter

### 2.1 Sample Time

Sample Time	
$T_S$	$50\mu s$

### 3 Mask Parameter

AutoSwitch: AutoSwitch	
Thresh_up	0.5
Thresh_down	0.0
Used Implementation	FiP16

AutoSwitch: AutoSwitch1	
Thresh_up	0.5
Thresh_down	0.0
Used Implementation	FiP16

Constant: Constant	
Value	0.6
Used Implementation	FiP16

Constant: Constant1	
Value	0.0
Used Implementation	FiP16

Constant: Constant2	
Value	0.0
Used Implementation	FiP16

Constant: Constant3	
Value	0.0
Used Implementation	FiP16

Constant: Constant5	
Value	0.0
Used Implementation	FiP16

Constant: Constant6	
Value	0.0
Used Implementation	FiP16

Negation: Negation2	
Used Implementation	FiP16

<b>Negation: Negation3</b>	
Used Implementation	FiP16

<b>PI: PI1</b>	
Kp	12.0
Ki	4.0
ts_fact	1.0
Used Implementation	FiP16

<b>PI: PI2</b>	
Kp	0.5
Ki	2.0
ts_fact	1.0
Used Implementation	FiP16

<b>PI: PI4</b>	
Kp	0.5
Ki	2.0
ts_fact	1.0
Used Implementation	FiP16

<b>Add: SVM__Add</b>	
Used Implementation	FiP16

<b>Gain: SVM__Gain</b>	
Gain	-1.1
Used Implementation	FiP16

<b>Gain: SVM__Gain1</b>	
Gain	-1.1
Used Implementation	FiP16

<b>Gain: SVM__Gain2</b>	
Gain	-1.1
Used Implementation	FiP16

<b>Gain: SVM__Gain3</b>	
Gain	0.5
Used Implementation	FiP16

<b>Maximum: SVM__Maximum</b>	
Used Implementation	FiP16

<b>Maximum: SVM__Maximum1</b>	
Used Implementation	FiP16

<b>Minimum: SVM__Minimum</b>	
Used Implementation	FiP16

<b>Minimum: SVM__Minimum1</b>	
Used Implementation	FiP16

<b>Sub: SVM__Sub</b>	
Used Implementation	FiP16

<b>Sub: SVM__Sub1</b>	
Used Implementation	FiP16

<b>Sub: SVM__Sub2</b>	
Used Implementation	FiP16

<b>Sub: Sub2</b>	
Used Implementation	FiP16

<b>Sub: Sub3</b>	
Used Implementation	FiP16

<b>Sub: Sub4</b>	
Used Implementation	FiP16

<b>TypeConv: TypeConv</b>	
Used Implementation	FiP16_8

<b>clark: clark</b>	
Used Implementation	INT16_ASM

<b>clark_inv: clark_inv</b>	
Used Implementation	INT16_ASM

<b>park: park</b>	
Used Implementation	INT16_ASM

<b>park_inv: park_inv</b>	
Used Implementation	INT16_ASM

<b>reset_qei: reset_qei</b>	
Used Implementation	INT16

<b>timer: timer</b>	
ts_fact	1.0
time	1.0
Used Implementation	INT16



## Part II

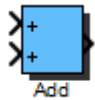
# Used X2C-Blocks

## 4 Project Specific Blocks

## 5 Internal Library Blocks

### Block: Add

---



Inports	
In1	Addend 1
In2	Addend 2

Outputs	
Out	Sum

#### Description:

Addition of input 1 and input 2.

Calculation:

$$Out = In_1 + In_2$$

#### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

#### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	4960
<b>Revision</b>	0.3
<b>C filename</b>	Add_FiP8.c
<b>H filename</b>	Add_FiP8.h

## 8 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int8        *In1;  
    int8        *In2;  
    int8        Out;  
} ADD_FIP8;
```

### Implementation: FiP16

<b>Name</b>	FiP16
<b>ID</b>	4961
<b>Revision</b>	0.3
<b>C filename</b>	Add_FiP16.c
<b>H filename</b>	Add_FiP16.h

## 16 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In1;  
    int16       *In2;  
    int16       Out;  
} ADD_FIP16;
```

### Implementation: FiP32

<b>Name</b>	FiP32
<b>ID</b>	4962
<b>Revision</b>	0.3
<b>C filename</b>	Add_FiP32.c
<b>H filename</b>	Add_FiP32.h

## 32 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       *In1;  
    int32       *In2;  
    int32       Out;  
} ADD_FIP32;
```

## Implementation: Float32

---

<b>Name</b>	Float32
<b>ID</b>	4963
<b>Revision</b>	0.1
<b>C filename</b>	Add_Float32.c
<b>H filename</b>	Add_Float32.h

32 Bit Floating Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    float32     *In1;  
    float32     *In2;  
    float32     Out;  
} ADD_FLOAT32;
```

## Implementation: Float64

---

<b>Name</b>	Float64
<b>ID</b>	4964
<b>Revision</b>	0.1
<b>C filename</b>	Add_Float64.c
<b>H filename</b>	Add_Float64.h

64 Bit Floating Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    float64     *In1;  
    float64     *In2;  
    float64     Out;  
} ADD_FLOAT64;
```

## Block: AutoSwitch

---



Inports	
In1	Input #1
Switch	Input #2: Threshold signal
In3	Input #3

Outputs	
Out	Either value of input #1 or input #3 dependent on value of input #2

Mask Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal

### Description:

Switch between In1 and In3 dependent on Switch signal:

Switch signal rising:  $\text{Switch} \geq \text{Threshold up} \rightarrow \text{Out} = \text{In1}$

Switch signal falling:  $\text{Switch} < \text{Threshold down} \rightarrow \text{Out} = \text{In3}$

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	128
<b>Revision</b>	0.1
<b>C filename</b>	AutoSwitch_FiP8.c
<b>H filename</b>	AutoSwitch_FiP8.h

8 Bit Fixed Point Implementation

Controller Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal
Status	Current hysteresis state

#### Data Structure:

```
typedef struct {
    uint16      ID;
    int8        *In1;
    int8        *Switch;
    int8        *In3;
    int8        Out;
    int8        Thresh_up;
    int8        Thresh_down;
    int8        Status;
} AUTOSWITCH_FIP8;
```

#### Implementation: FiP16

**Name**            FiP16  
**ID**                129  
**Revision**        0.1  
**C filename**      AutoSwitch\_FiP16.c  
**H filename**      AutoSwitch\_FiP16.h

16 Bit Fixed Point Implementation

Controller Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal
Status	Current hysteresis state

#### Data Structure:

```
typedef struct {
    uint16      ID;
    int16        *In1;
    int16        *Switch;
    int16        *In3;
    int16        Out;
    int16        Thresh_up;
    int16        Thresh_down;
    int8         Status;
} AUTOSWITCH_FIP16;
```

#### Implementation: FiP32

**Name** FiP32  
**ID** 130  
**Revision** 0.1  
**C filename** AutoSwitch\_FiP32.c  
**H filename** AutoSwitch\_FiP32.h

32 Bit Fixed Point Implementation

Controller Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal
Status	Current hysteresis state

**Data Structure:**

```

typedef struct {
    uint16      ID;
    int32       *In1;
    int32       *Switch;
    int32       *In3;
    int32       Out;
    int32       Thresh_up;
    int32       Thresh_down;
    int8        Status;
} AUTOSWITCH_FIP32;
  
```

**Implementation: Float32**

**Name** Float32  
**ID** 131  
**Revision** 0.1  
**C filename** AutoSwitch\_Float32.c  
**H filename** AutoSwitch\_Float32.h

32 Bit Floating Point Implementation

Controller Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal
Status	Current hysteresis state

**Data Structure:**

```

typedef struct {
    uint16      ID;
    float32     *In1;
    float32     *Switch;
    float32     *In3;
    float32     Out;
    float32     Thresh_up;
}
  
```

```

float32    Thresh_down;
int8       Status;
} AUTOSWITCH_FLOAT32;

```

## Implementation: Float64

**Name** Float64  
**ID** 132  
**Revision** 0.1  
**C filename** AutoSwitch\_Float64.c  
**H filename** AutoSwitch\_Float64.h

64 Bit Floating Point Implementation

Controller Parameters	
Thresh_up	Threshold level for rising switch signal
Thresh_down	Threshold level for falling switch signal
Status	Current hysteresis state

## Data Structure:

```

typedef struct {
    uint16    ID;
    float64    *In1;
    float64    *Switch;
    float64    *In3;
    float64    Out;
    float64    Thresh_up;
    float64    Thresh_down;
    int8       Status;
} AUTOSWITCH_FLOAT64;

```

## Block: Constant

---



Outputs	
Out	Constant output

Mask Parameters	
Value	Constant factor

### Description:

Constant value.

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	48
<b>Revision</b>	0.3
<b>C filename</b>	Constant_FiP8.c
<b>H filename</b>	Constant_FiP8.h

8 Bit Fixed Point Implementation

Controller Parameters	
K	Constant factor

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 Out;  
    int8 K;  
} CONSTANT_FIP8;
```



### Implementation: FiP16

---

**Name** FiP16  
**ID** 49  
**Revision** 0.3  
**C filename** Constant\_FiP16.c  
**H filename** Constant\_FiP16.h

16 Bit Fixed Point Implementation

Controller Parameters	
K	Constant factor

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       Out;  
    int16       K;  
} CONSTANT_FIP16;
```

### Implementation: FiP32

---

**Name** FiP32  
**ID** 50  
**Revision** 0.3  
**C filename** Constant\_FiP32.c  
**H filename** Constant\_FiP32.h

32 Bit Fixed Point Implementation

Controller Parameters	
K	Constant factor

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       Out;  
    int32       K;  
} CONSTANT_FIP32;
```

### Implementation: Float32

---

**Name** Float32  
**ID** 51  
**Revision** 0.1  
**C filename** Constant\_Float32.c  
**H filename** Constant\_Float32.h

32 Bit Floating Point Implementation

Controller Parameters	
K	Constant factor

#### Data Structure:

```
typedef struct {
    uint16      ID;
    float32     Out;
    float32     K;
} CONSTANT_FLOAT32;
```

#### Implementation: Float64

**Name** Float64  
**ID** 52  
**Revision** 0.1  
**C filename** Constant\_Float64.c  
**H filename** Constant\_Float64.h

64 Bit Floating Point Implementation

Controller Parameters	
K	Constant factor

#### Data Structure:

```
typedef struct {
    uint16      ID;
    float64     Out;
    float64     K;
} CONSTANT_FLOAT64;
```

## Block: Gain

---



Inports	
In	Input

Outputs	
Out	Amplified input

Mask Parameters	
Gain	Gain factor in floating point format

### Description:

Amplification of input by gain factor.

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	16
<b>Revision</b>	1.0
<b>C filename</b>	Gain_FiP8.c
<b>H filename</b>	Gain_FiP8.h

8 Bit Fixed Point Implementation

Controller Parameters	
V	Gain factor
sfr	Shift factor

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 *In;
```

```

    int8      Out;
    int8      V;
    int8      sfr;
} GAIN_FIP8;

```

### Implementation: FiP16

**Name**            FiP16  
**ID**                17  
**Revision**        1.0  
**C filename**      Gain\_FiP16.c  
**H filename**      Gain\_FiP16.h

16 Bit Fixed Point Implementation

Controller Parameters	
V	Gain factor
sfr	Shift factor

#### Data Structure:

```

typedef struct {
    uint16      ID;
    int16       *In;
    int16       Out;
    int16       V;
    int8        sfr;
} GAIN_FIP16;

```

### Implementation: FiP32

**Name**            FiP32  
**ID**                18  
**Revision**        1.0  
**C filename**      Gain\_FiP32.c  
**H filename**      Gain\_FiP32.h

32 Bit Fixed Point Implementation

Controller Parameters	
V	Gain factor
sfr	Shift factor

#### Data Structure:

```

typedef struct {
    uint16      ID;
    int32       *In;

```

```

    int32      Out;
    int32      V;
    int8       sfr;
} GAIN_FIP32;

```

## Implementation: Float32

**Name** Float32  
**ID** 19  
**Revision** 0.1  
**C filename** Gain\_Float32.c  
**H filename** Gain\_Float32.h

32 Bit Floating Point Implementation

Controller Parameters	
V	Gain factor

### Data Structure:

```

typedef struct {
    uint16      ID;
    float32     *In;
    float32     Out;
    float32     V;
} GAIN_FLOAT32;

```

## Implementation: Float64

**Name** Float64  
**ID** 20  
**Revision** 0.1  
**C filename** Gain\_Float64.c  
**H filename** Gain\_Float64.h

64 Bit Floating Point Implementation

Controller Parameters	
V	Gain factor

### Data Structure:

```

typedef struct {
    uint16      ID;
    float64     *In;
    float64     Out;
    float64     V;
} GAIN_FLOAT64;

```

## Block: Maximum

---



Inports	
In1	Input #1
In2	Input #2

Outputs	
Out	Maximum of Input #1 and Input #2

### Description:

Outputs the greater value of the two input signals.

Calculation:

$$Out = \max(In_1, In_2)$$

### Implementations:

- FiP8** 8 Bit Fixed Point Implementation
- FiP16** 16 Bit Fixed Point Implementation
- FiP32** 32 Bit Fixed Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	368
<b>Revision</b>	0.1
<b>C filename</b>	Maximum_FiP8.c
<b>H filename</b>	Maximum_FiP8.h

8 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 *In1;  
    int8 *In2;  
    int8 Out;  
} MAXIMUM_FIP8;
```

## Implementation: FiP16

---

<b>Name</b>	FiP16
<b>ID</b>	369
<b>Revision</b>	0.1
<b>C filename</b>	Maximum_FiP16.c
<b>H filename</b>	Maximum_FiP16.h

16 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In1;  
    int16       *In2;  
    int16       Out;  
} MAXIMUM_FIP16;
```

## Implementation: FiP32

---

<b>Name</b>	FiP32
<b>ID</b>	370
<b>Revision</b>	0.1
<b>C filename</b>	Maximum_FiP32.c
<b>H filename</b>	Maximum_FiP32.h

32 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       *In1;  
    int32       *In2;  
    int32       Out;  
} MAXIMUM_FIP32;
```

## Block: Minimum

---



Inports	
In1	Input #1
In2	Input #2

Outputs	
Out	Minimum of Input #1 and Input #2

### Description:

Outputs the lesser value of the two input signals.

Calculation:

$$Out = \min(In_1, In_2)$$

### Implementations:

- FiP8** 8 Bit Fixed Point Implementation
- FiP16** 16 Bit Fixed Point Implementation
- FiP32** 32 Bit Fixed Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	352
<b>Revision</b>	0.1
<b>C filename</b>	Minimum_FiP8.c
<b>H filename</b>	Minimum_FiP8.h

8 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 *In1;  
    int8 *In2;  
    int8 Out;  
} MINIMUM_FIP8;
```



## Implementation: FiP16

---

<b>Name</b>	FiP16
<b>ID</b>	353
<b>Revision</b>	0.1
<b>C filename</b>	Minimum_FiP16.c
<b>H filename</b>	Minimum_FiP16.h

16 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In1;  
    int16       *In2;  
    int16       Out;  
} MINIMUM_FIP16;
```

## Implementation: FiP32

---

<b>Name</b>	FiP32
<b>ID</b>	354
<b>Revision</b>	0.1
<b>C filename</b>	Minimum_FiP32.c
<b>H filename</b>	Minimum_FiP32.h

32 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       *In1;  
    int32       *In2;  
    int32       Out;  
} MINIMUM_FIP32;
```

## Block: Negation

---



Inports	
In	Input

Outputs	
Out	Negated input value

### Description:

Negation of input signal.

Calculation:

$$Out = -In$$

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	5040
<b>Revision</b>	0.1
<b>C filename</b>	Negation_FiP8.c
<b>H filename</b>	Negation_FiP8.h

8 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {
    uint16 ID;
    int8 *In;
    int8 Out;
} NEGATION_FIP8;
```

### Implementation: FiP16

---

<b>Name</b>	FiP16
<b>ID</b>	5041
<b>Revision</b>	0.1
<b>C filename</b>	Negation_FiP16.c
<b>H filename</b>	Negation_FiP16.h

16 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In;  
    int16       Out;  
} NEGATION_FIP16;
```

### Implementation: FiP32

---

<b>Name</b>	FiP32
<b>ID</b>	5042
<b>Revision</b>	0.1
<b>C filename</b>	Negation_FiP32.c
<b>H filename</b>	Negation_FiP32.h

32 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       *In;  
    int32       Out;  
} NEGATION_FIP32;
```

### Implementation: Float32

---

<b>Name</b>	Float32
<b>ID</b>	5043
<b>Revision</b>	0.1
<b>C filename</b>	Negation_Float32.c
<b>H filename</b>	Negation_Float32.h

32 Bit Floating Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    float32     *In;
```

```
    float32    Out;  
} NEGATION_FLOAT32;
```

---

## Implementation: Float64

---

<b>Name</b>	Float64
<b>ID</b>	5044
<b>Revision</b>	0.1
<b>C filename</b>	Negation_Float64.c
<b>H filename</b>	Negation_Float64.h

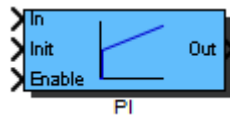
64 Bit Floating Point Implementation

### Data Structure:

```
typedef struct {  
    uint16    ID;  
    float64    *In;  
    float64    Out;  
} NEGATION_FLOAT64;
```

---

## Block: PI



Inports	
In	Control error input
Init	Value which is loaded at initialization function call
Enable	Enable == 0: Deactivation of block; Out set to 0 Enable 0->1: Preload of integral part Enable == 1: Activation of block

Outputs	
Out	

Mask Parameters	
Kp	Proportional Factor
Ki	Integral Factor
ts_fact	Multiplication factor of base sampling time (in integer format)

### Description:

PI controller:

$$G(s) = K_p + K_i/s$$

Each fixed point implementation uses the next higher integer data type for the integral value storage variable.

A rising flank at the *Enable* inport will preload the integral part with the value present on the *Init* inport.

Transfer function (zero-order hold discretization method):

$$G(z) = K_P + K_I T_s \frac{1}{z - 1}$$

**Developer note:** For the fixed point implementations the source code of block ?? is used.

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

**Name** FiP8  
**ID** 3216  
**Revision** 2.0  
**C filename** PI\_FiP8.c  
**H filename** PI\_FiP8.h

8 Bit Fixed Point Implementation

Controller Parameters	
b0	Integral coefficient
b1	Proportional coefficient
sfrb0	Shift factor for PI coefficient b0
sfrb1	Shift factor for PI coefficient b1
i_old	Integrator value from previous cycle
enable_old	Enable value of previous cycle

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 *In;  
    int8 *Init;  
    int8 *Enable;  
    int8 Out;  
    int8 b0;  
    int8 b1;  
    int8 sfrb0;  
    int8 sfrb1;  
    int16 i_old;  
    int8 enable_old;  
} PI_FIP8;
```

### Implementation: FiP16

---

**Name** FiP16  
**ID** 3217  
**Revision** 2.0  
**C filename** PI\_FiP16.c  
**H filename** PI\_FiP16.h

16 Bit Fixed Point Implementation

Controller Parameters	
b0	Integral coefficient
b1	Proportional coefficient
sfrb0	Shift factor for PI coefficient b0
sfrb1	Shift factor for PI coefficient b1
i_old	Integrator value from previous cycle
enable_old	Enable value of previous cycle

#### Data Structure:

```
typedef struct {
    uint16    ID;
    int16     *In;
    int16     *InIt;
    int8      *Enable;
    int16     Out;
    int16     b0;
    int16     b1;
    int8      sfrb0;
    int8      sfrb1;
    int32     i_old;
    int8      enable_old;
} PI_FIP16;
```

#### Implementation: FiP32

**Name**            FiP32  
**ID**                3218  
**Revision**        2.0  
**C filename**      PI\_FiP32.c  
**H filename**      PI\_FiP32.h

32 Bit Fixed Point Implementation

Controller Parameters	
b0	Integral coefficient
b1	Proportional coefficient
sfrb0	Shift factor for PI coefficient b0
sfrb1	Shift factor for PI coefficient b1
i_old	Integrator value from previous cycle
enable_old	Enable value of previous cycle

#### Data Structure:

```
typedef struct {
    uint16    ID;
    int32     *In;
    int32     *InIt;
}
```

```

    int8      *Enable;
    int32     Out;
    int32     b0;
    int32     b1;
    int8      sfrb0;
    int8      sfrb1;
    int64     i_old;
    int8      enable_old;
} PI_FIP32;

```

## Implementation: Float32

**Name** Float32  
**ID** 3219  
**Revision** 2.0  
**C filename** PI\_Float32.c  
**H filename** PI\_Float32.h

32 Bit Floating Point Implementation

Controller Parameters	
b0	Integral coefficient
b1	Proportional coefficient
i_old	Integrator value of previous cycle
enable_old	Enable value of previous cycle

## Data Structure:

```

typedef struct {
    uint16     ID;
    float32    *In;
    float32    *Init;
    int8       *Enable;
    float32    Out;
    float32    b0;
    float32    b1;
    float32    i_old;
    int8       enable_old;
} PI_FLOAT32;

```

## Implementation: Float64

**Name** Float64  
**ID** 3220  
**Revision** 2.0  
**C filename** PI\_Float64.c  
**H filename** PI\_Float64.h

64 Bit Floating Point Implementation



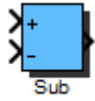
Controller Parameters	
b0	Integral coefficient
b1	Proportional coefficient
i_old	Integrator value of previous cycle
enable_old	Enable value of previous cycle

#### Data Structure:

```
typedef struct {
    uint16    ID;
    float64   *In;
    float64   *Init;
    int8      *Enable;
    float64    Out;
    float64    b0;
    float64    b1;
    float64    i_old;
    int8       enable_old;
} PI_FLOAT64;
```

## Block: Sub

---



Inports	
Plus	Minuend
Minus	Subtrahend

Outputs	
Out	Difference

### Description:

Subtraction of input Minus from input Plus.

Calculation:

$$Out = Plus - Minus$$

### Implementations:

<b>FiP8</b>	8 Bit Fixed Point Implementation
<b>FiP16</b>	16 Bit Fixed Point Implementation
<b>FiP32</b>	32 Bit Fixed Point Implementation
<b>Float32</b>	32 Bit Floating Point Implementation
<b>Float64</b>	64 Bit Floating Point Implementation

### Implementation: FiP8

---

<b>Name</b>	FiP8
<b>ID</b>	4992
<b>Revision</b>	0.1
<b>C filename</b>	Sub_FiP8.c
<b>H filename</b>	Sub_FiP8.h

8 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int8 *Plus;  
    int8 *Minus;  
    int8 Out;  
} SUB_FIP8;
```

### Implementation: FiP16

---

<b>Name</b>	FiP16
<b>ID</b>	4993
<b>Revision</b>	0.1
<b>C filename</b>	Sub_FiP16.c
<b>H filename</b>	Sub_FiP16.h

16 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *Plus;  
    int16       *Minus;  
    int16       Out;  
} SUB_FIP16;
```

### Implementation: FiP32

---

<b>Name</b>	FiP32
<b>ID</b>	4994
<b>Revision</b>	0.1
<b>C filename</b>	Sub_FiP32.c
<b>H filename</b>	Sub_FiP32.h

32 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int32       *Plus;  
    int32       *Minus;  
    int32       Out;  
} SUB_FIP32;
```

### Implementation: Float32

---

<b>Name</b>	Float32
<b>ID</b>	4995
<b>Revision</b>	0.1
<b>C filename</b>	Sub_Float32.c
<b>H filename</b>	Sub_Float32.h

32 Bit Floating Point Implementation

#### Data Structure:

```
typedef struct {
    uint16      ID;
    float32     *Plus;
    float32     *Minus;
    float32     Out;
} SUB_FLOAT32;
```

## Implementation: Float64

<b>Name</b>	Float64
<b>ID</b>	4996
<b>Revision</b>	0.1
<b>C filename</b>	Sub_Float64.c
<b>H filename</b>	Sub_Float64.h

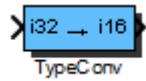
64 Bit Floating Point Implementation

### Data Structure:

```
typedef struct {
    uint16      ID;
    float64     *Plus;
    float64     *Minus;
    float64     Out;
} SUB_FLOAT64;
```

## Block: TypeConv

---



Inports	
In	

Outputs	
Out	

### Description:

Data Type Conversion

### Implementations:

<b>FiP8_16</b>	8 to 16 Bit Fixed Point Implementation
<b>FiP8_32</b>	8 to 32 Bit Fixed Point Implementation
<b>FiP16_8</b>	16 to 8 Bit Fixed Point Implementation
<b>FiP16_32</b>	16 to 32 Bit Fixed Point Implementation
<b>FiP32_8</b>	32 to 8 Bit Fixed Point Implementation
<b>FiP32_16</b>	32 to 16 Bit Fixed Point Implementation

### Implementation: FiP8\_16

---

<b>Name</b>	FiP8_16
<b>ID</b>	176
<b>Revision</b>	0.1
<b>C filename</b>	TypeConv_FiP8_16.c
<b>H filename</b>	TypeConv_FiP8_16.h

8 to 16 Bit Fixed Point Implementation

### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int8        *In;  
    int16       Out;  
} TYPECONV_FIP8_16;
```

### Implementation: FiP8\_32

---

<b>Name</b>	FiP8_32
<b>ID</b>	177
<b>Revision</b>	0.1
<b>C filename</b>	TypeConv_FiP8_32.c
<b>H filename</b>	TypeConv_FiP8_32.h

8 to 32 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int8        *In;  
    int32       Out;  
} TYPECONV_FIP8_32;
```

### Implementation: FiP16\_8

---

<b>Name</b>	FiP16_8
<b>ID</b>	178
<b>Revision</b>	0.1
<b>C filename</b>	TypeConv_FiP16_8.c
<b>H filename</b>	TypeConv_FiP16_8.h

16 to 8 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In;  
    int8        Out;  
} TYPECONV_FIP16_8;
```

### Implementation: FiP16\_32

---

<b>Name</b>	FiP16_32
<b>ID</b>	179
<b>Revision</b>	0.1
<b>C filename</b>	TypeConv_FiP16_32.c
<b>H filename</b>	TypeConv_FiP16_32.h

16 to 32 Bit Fixed Point Implementation

#### Data Structure:

```
typedef struct {  
    uint16      ID;  
    int16       *In;
```

```

        int32      Out;
    } TYPECONV_FIP16_32;

```

---

### Implementation: FiP32\_8

---

**Name**            FiP32\_8  
**ID**                180  
**Revision**        0.1  
**C filename**       TypeConv\_FiP32\_8.c  
**H filename**       TypeConv\_FiP32\_8.h

32 to 8 Bit Fixed Point Implementation

#### Data Structure:

```

typedef struct {
    uint16      ID;
    int32       *In;
    int8        Out;
} TYPECONV_FIP32_8;

```

---

### Implementation: FiP32\_16

---

**Name**            FiP32\_16  
**ID**                181  
**Revision**        0.1  
**C filename**       TypeConv\_FiP32\_16.c  
**H filename**       TypeConv\_FiP32\_16.h

32 to 16 Bit Fixed Point Implementation

#### Data Structure:

```

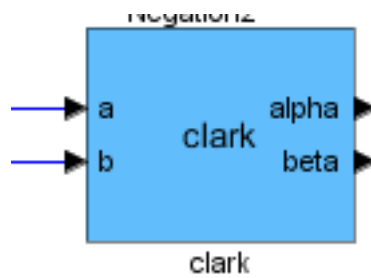
typedef struct {
    uint16      ID;
    int32       *In;
    int16       Out;
} TYPECONV_FIP32_16;

```

---

## Block: clark

---



Inports	
a	
b	

Outports	
alpha	
beta	

### Description:

Clark-Transformation

### Implementations:

**INT16\_ASM**

### Implementation: INT16\_ASM

---

**Name** INT16\_ASM  
**ID** 20864  
**Revision**  
**C filename** clark\_INT16\_ASM.c  
**H filename** clark\_INT16\_ASM.h

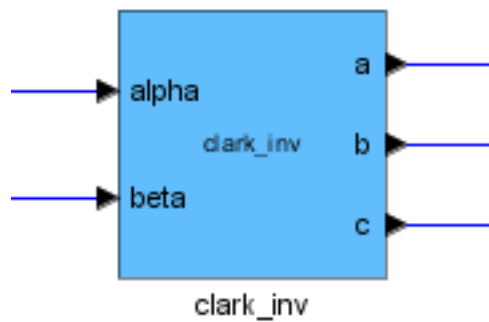
### Data Structure:

```
typedef struct {  
    uint16 ID;  
    int16 *a;  
    int16 *b;  
    int16 alpha;  
    int16 beta;  
} CLARK_INT16_ASM;
```



## Block: clark\_inv

---



Inports	
alpha	
beta	

Outports	
a	
b	
c	

**Description:**

**Implementations:**

**INT16\_ASM**

### Implementation: INT16\_ASM

---

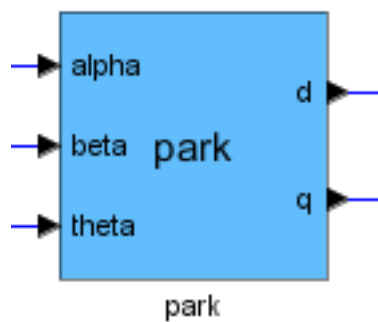
**Name** INT16\_ASM  
**ID** 20896  
**Revision**  
**C filename** clark\_inv\_INT16\_ASM.c  
**H filename** clark\_inv\_INT16\_ASM.h

**Data Structure:**

```
typedef struct {  
    uint16      ID;  
    int16       *alpha;  
    int16       *beta;  
    int16       a;  
    int16       b;  
    int16       c;  
} CLARK_INV_INT16_ASM;
```

## Block: park

---



Inports	
alpha	
beta	
theta	

Outports	
d	
q	

### Description:

Park-Transformation

### Implementations:

**INT16\_ASM**

### Implementation: INT16\_ASM

---

**Name** INT16\_ASM  
**ID** 20880  
**Revision**  
**C filename** park\_INT16\_ASM.c  
**H filename** park\_INT16\_ASM.h

### Data Structure:

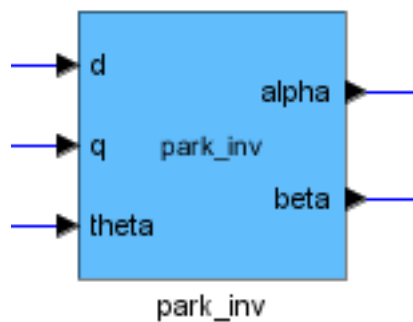
```
typedef struct {  
    uint16    ID;  
    int16     *alpha;  
    int16     *beta;  
    int16     *theta;  
    int16     d;  
    int16     q;  
}
```

```
} PARK_INT16_ASM;
```

---

## Block: park\_inv

---



Inports	
d	
q	
theta	

Outports	
alpha	
beta	

**Description:**

**Implementations:**

**INT16\_ASM**

### Implementation: INT16\_ASM

---

**Name** INT16\_ASM  
**ID** 20912  
**Revision**  
**C filename** park\_inv\_INT16\_ASM.c  
**H filename** park\_inv\_INT16\_ASM.h

**Data Structure:**

```
typedef struct {  
    uint16    ID;  
    int16     *d;  
    int16     *q;  
    int16     *theta;  
    int16     alpha;  
    int16     beta;  
} PARK_INV_INT16_ASM;
```

## Block: reset\_qei

---



Inports	
enable	

**Description:**

**Implementations:**

**INT16**

### Implementation: INT16

---

<b>Name</b>	INT16
<b>ID</b>	20832
<b>Revision</b>	
<b>C filename</b>	reset_qei_INT16.c
<b>H filename</b>	reset_qei_INT16.h

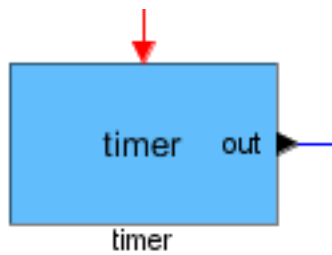
Controller Parameters	
is_reseted	

**Data Structure:**

```
typedef struct {  
    uint16      ID;  
    int16       *enable;  
    int16       is_reseted;  
} RESET_QEI_INT16;
```

## Block: timer

---



Outputs	
out	

Mask Parameters	
ts_fact	
time	

### Description:

outputs 1 after a configurable time

### Implementations:

**INT16**

### Implementation: INT16

---

**Name** INT16  
**ID** 20816  
**Revision**  
**C filename** timer\_INT16.c  
**H filename** timer\_INT16.h

Controller Parameters	
cnt_val	
cnt_lim	

### Data Structure:

```
typedef struct {
    uint16 ID;
    int16 out;
    int16 cnt_val;
    int16 cnt_lim;
} TIMER_INT16;
```