LINZ CENTER OF MECHATRONICS GMBH

Project Documentation foc_closed_loop_pos

January 21, 2016

Contents

Part I
X2C Model

1 Model Structure

1.1 Xcos Model

NOTE: Automated generation of model images is not implemented yet. You can manually include the model image by following these steps:

1. Go to Xcos model window
2. Select *File->Export all diagrams*
3. Save image(s) as *.png file(s) in *X2CCode* directory
4. Generate project documentation again by double-clicking the block *create Documentation*

1.2 Subsystems

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

Average: Average	
n	256
ts_fact	1.0
Used Implementation	FiP16

Average: Average1	
n	256
ts_fact	1.0
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

Constant: Constant7	
Value	0.0
Used Implementation	FiP16

Negation: Negation1	
Used Implementation	FiP16

Negation: Negation2	
Used Implementation	FiP16

PI: PI1	
Kp	4.0
Ki	1.0
ts_fact	1.0
Used Implementation	FiP16

PI: PI2	
Kp	0.8
Ki	1.0
ts_fact	1.0
Used Implementation	FiP16

PI: PI3	
Kp	5.0
Ki	5.0
ts_fact	1.0
Used Implementation	FiP16

PI: PI4	
Kp	0.8
Ki	0.0
ts_fact	1.0
Used Implementation	FiP16

Add: SVM__Add	
Used Implementation	FiP16

Gain: SVM__Gain	
Gain	-1.1
Used Implementation	FiP16

Gain: SVM__Gain1	
Gain	-1.1
Used Implementation	FiP16

Gain: SVM__Gain2	
Gain	-1.1
Used Implementation	FiP16

Gain: SVM__Gain3	
Gain	0.5
Used Implementation	FiP16

Maximum: SVM__Maximum	
Used Implementation	FiP16

Maximum: SVM__Maximum1	
Used Implementation	FiP16

Minimum: SVM__Minimum	
Used Implementation	FiP16

Minimum: SVM__Minimum1	
Used Implementation	FiP16

Sub: SVM__Sub	
Used Implementation	FiP16

Sub: SVM__Sub1	
Used Implementation	FiP16

Sub: SVM__Sub2	
Used Implementation	FiP16

Sub: Sub1	
Used Implementation	FiP16

Sub: Sub2	
Used Implementation	FiP16

Sub: Sub3	
Used Implementation	FiP16

Sub: Sub4	
Used Implementation	FiP16

TypeConv: TypeConv	
Used Implementation	FiP16_8

clark: clark	
Used Implementation	INT16_ASM

clark_inv: clark_inv	
Used Implementation	INT16_ASM

park: park	
Used Implementation	INT16_ASM

park_inv: park_inv	
Used Implementation	INT16_ASM

reset_qei: reset_qei	
Used Implementation	INT16

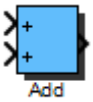
timer: timer	
ts_fact	1.0
time	1.0
Used Implementation	INT16

Part II
Used X2C-Blocks

4 Project Specific Blocks

5 Internal Library Blocks

Block: Add



Inports	
In1	Addend 1
In2	Addend 2

Outports	
Out	Sum

Description:

Addition of input 1 and input 2.

Calculation:

$$Out \quad = \quad In_1 + In_2$$

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	4960
Revision	0.3
C filename	Add_FiP8.c
H filename	Add_FiP8.h

8 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP16

Name	FiP16
ID	4961
Revision	0.3
C filename	Add_FiP16.c
H filename	Add_FiP16.h

16 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP32

Name	FiP32
ID	4962
Revision	0.3
C filename	Add_FiP32.c
H filename	Add_FiP32.h

32 Bit Fixed Point Implementation

Data Structure:

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

Implementation: Float32

Name	Float32
ID	4963
Revision	0.1
C filename	Add_Float32.c
H filename	Add_Float32.h

32 Bit Floating Point Implementation

Data Structure:

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

Implementation: Float64

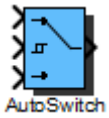
Name	Float64
ID	4964
Revision	0.1
C filename	Add_Float64.c
H filename	Add_Float64.h

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

Outports	
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: Switch >= Threshold up –> Out = In1
Switch signal falling: Switch < Threshold down –> Out = In3

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	128
Revision	0.1
C filename	AutoSwitch_FiP8.c
H filename	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: Average

Inports	
In	Input value

Outports	
Out	Averaged value

Mask Parameters	
n	Number of points to be averaged over
ts_fact	Multiplication factor of base sampling time (in integer format)

Description:

Calculation of moving average value over n numbers.

Calculation:

$$Out_k = \frac{1}{n} \sum_{i=k-n}^{k} In_i$$

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	5024
Revision	1
C filename	Average_FiP8.c
H filename	Average_FiP8.h

8 Bit Fixed Point Implementation

Controller Parameters	
n	Average window size
sfrn	Shift factor for computation of average value sfrn = ld(n)
sum	Temporary sum
count	Index counter
avg	Array with data values

Data Structure:

```
typedef struct {
      uint16          ID;
      int8            *In;
      int8            Out;
      uint16          n;
      uint8           sfrn;
      int16           sum;
      uint16          count;
      int8            *avg;
} AVERAGE_FIP8;
```

Implementation: FiP16

Name	FiP16
ID	5025
Revision	1
C filename	Average_FiP16.c
H filename	Average_FiP16.h

16 Bit Fixed Point Implementation

Controller Parameters	
n	Average window size
sfrn	Shift factor for computation of average value sfrn = ld(n)
sum	Temporary sum
count	Index counter
avg	Array with data values

Data Structure:

```
typedef struct {
      uint16          ID;
      int16           *In;
      int16           Out;
      uint16          n;
      uint8           sfrn;
      int32           sum;
      uint16          count;
      int16           *avg;
} AVERAGE_FIP16;
```

Implementation: FiP32

Name FiP32
ID 5026
Revision 1
C filename Average_FiP32.c
H filename Average_FiP32.h

32 Bit Fixed Point Implementation

Controller Parameters	
n	Average window size
sfrn	Shift factor for computation of average value sfrn = ld(n)
sum	Temporary sum
count	Index counter
avg	Array with data values

Data Structure:

```
typedef struct {
      uint16          ID;
      int32           *In;
      int32           Out;
      uint16          n;
      uint8           sfrn;
      int64           sum;
      uint16          count;
      int32           *avg;
} AVERAGE_FIP32;
```

Implementation: Float32

Name Float32
ID 5027
Revision 0.1
C filename Average_Float32.c
H filename Average_Float32.h

32 Bit Floating Point Implementation

Controller Parameters	
n	Average window size
sum	Temporary sum
count	Index counter
avg	Array with data values

Data Structure:

```
typedef struct {
```

```
    uint16          ID;
    float32         *In;
    float32         Out;
    uint16          n;
    float32         sum;
    uint16          count;
    float32         *avg;
} AVERAGE_FLOAT32;
```

Implementation: Float64

Name	Float64
ID	5028
Revision	0.1
C filename	Average_Float64.c
H filename	Average_Float64.h

64 Bit Floating Point Implementation

Controller Parameters	
n	Average window size
sum	Temporary sum
count	Index counter
avg	Array with data values

Data Structure:

```
typedef struct {
    uint16          ID;
    float64         *In;
    float64         Out;
    uint16          n;
    float64         sum;
    uint16          count;
    float64         *avg;
} AVERAGE_FLOAT64;
```

Block: Constant

Outports	
Out	Constant output

Mask Parameters	
Value	Constant factor

Description:

Constant value.

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	48
Revision	0.3
C filename	Constant_FiP8.c
H filename	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

Outports	
Out	Amplified input

Mask Parameters	
Gain	Gain factor in floating point format

Description:

Amplification of input by gain factor.

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	16
Revision	1.0
C filename	Gain_FiP8.c
H filename	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

Outports	
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

Name FiP8
ID 368
Revision 0.1
C filename Maximum_FiP8.c
H filename Maximum_FiP8.h

8 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP16

Name	FiP16
ID	369
Revision	0.1
C filename	Maximum_FiP16.c
H filename	Maximum_FiP16.h

16 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP32

Name	FiP32
ID	370
Revision	0.1
C filename	Maximum_FiP32.c
H filename	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

Outports	
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

Name FiP8
ID 352
Revision 0.1
C filename Minimum_FiP8.c
H filename Minimum_FiP8.h

8 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP16

Name	FiP16
ID	353
Revision	0.1
C filename	Minimum_FiP16.c
H filename	Minimum_FiP16.h

16 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP32

Name	FiP32
ID	354
Revision	0.1
C filename	Minimum_FiP32.c
H filename	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

Outports	
Out	Negated input value

Description:

Negation of input signal.

Calculation:

$$Out \quad = \quad -In$$

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	5040
Revision	0.1
C filename	Negation_FiP8.c
H filename	Negation_FiP8.h

8 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP16

Name	FiP16
ID	5041
Revision	0.1
C filename	Negation_FiP16.c
H filename	Negation_FiP16.h

16 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP32

Name	FiP32
ID	5042
Revision	0.1
C filename	Negation_FiP32.c
H filename	Negation_FiP32.h

32 Bit Fixed Point Implementation

Data Structure:

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

Implementation: Float32

Name	Float32
ID	5043
Revision	0.1
C filename	Negation_Float32.c
H filename	Negation_Float32.h

32 Bit Floating Point Implementation

Data Structure:

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

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

Implementation: Float64

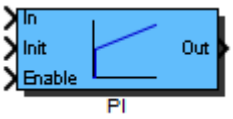
Name	Float64
ID	5044
Revision	0.1
C filename	Negation_Float64.c
H filename	Negation_Float64.h

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

Outports	
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) = Kp + Ki/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):

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

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

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	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

Outports	
Out	Difference

Description:

Subtraction of input Minus from input Plus.

Calculation:

$$Out \quad = \quad Plus - Minus$$

Implementations:

FiP8	8 Bit Fixed Point Implementation
FiP16	16 Bit Fixed Point Implementation
FiP32	32 Bit Fixed Point Implementation
Float32	32 Bit Floating Point Implementation
Float64	64 Bit Floating Point Implementation

Implementation: FiP8

Name	FiP8
ID	4992
Revision	0.1
C filename	Sub_FiP8.c
H filename	Sub_FiP8.h

8 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP16

Name	FiP16
ID	4993
Revision	0.1
C filename	Sub_FiP16.c
H filename	Sub_FiP16.h

16 Bit Fixed Point Implementation

Data Structure:

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

Implementation: FiP32

Name	FiP32
ID	4994
Revision	0.1
C filename	Sub_FiP32.c
H filename	Sub_FiP32.h

32 Bit Fixed Point Implementation

Data Structure:

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

Implementation: Float32

Name	Float32
ID	4995
Revision	0.1
C filename	Sub_Float32.c
H filename	Sub_Float32.h

32 Bit Floating Point Implementation

Data Structure:

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

Implementation: Float64

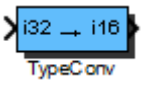
Name	Float64
ID	4996
Revision	0.1
C filename	Sub_Float64.c
H filename	Sub_Float64.h

64 Bit Floating Point Implementation

Data Structure:

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

Block: TypeConv



Inports	
In	

Outports	
Out	

Description:

Data Type Conversion

Implementations:

FiP8_16	8 to 16 Bit Fixed Point Implementation
FiP8_32	8 to 32 Bit Fixed Point Implementation
FiP16_8	16 to 8 Bit Fixed Point Implementation
FiP16_32	16 to 32 Bit Fixed Point Implementation
FiP32_8	32 to 8 Bit Fixed Point Implementation
FiP32_16	32 to 16 Bit Fixed Point Implementation

Implementation: FiP8_16

Name	FiP8_16
ID	176
Revision	0.1
C filename	TypeConv_FiP8_16.c
H filename	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

Name	FiP8_32
ID	177
Revision	0.1
C filename	TypeConv_FiP8_32.c
H filename	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

Name	FiP16_8
ID	178
Revision	0.1
C filename	TypeConv_FiP16_8.c
H filename	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

Name	FiP16_32
ID	179
Revision	0.1
C filename	TypeConv_FiP16_32.c
H filename	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;
```