



# ***Project Documentation DemoApplication***

July 28, 2017

© Linz Center of Mechatronics GmbH

# Contents

|            |   |           |
|------------|---|-----------|
| <b>I</b>   | <b>X2C Model</b>                            | <b>2</b>  |
| <b>1</b>   | <b>Version Information</b>                  | <b>2</b>  |
| 1.1        | X2C . . . . .                               | 2         |
| 1.2        | Operating System . . . . .                  | 2         |
| 1.3        | Scilab . . . . .                            | 2         |
| <b>2</b>   | <b>Model Structure</b>                      | <b>2</b>  |
| 2.1        | Xcos Model . . . . .                        | 2         |
| 2.2        | Subsystems . . . . .                        | 3         |
| <b>3</b>   | <b>Model Parameter</b>                      | <b>4</b>  |
| 3.1        | Sample Time . . . . .                       | 4         |
| <b>4</b>   | <b>Mask Parameter</b>                       | <b>5</b>  |
| <br>       |   |           |
| <b>II</b>  | <b>Frame Program Documentation</b>          | <b>7</b>  |
| <br>       |   |           |
| <b>5</b>   | <b>File Index</b>                           | <b>7</b>  |
| 5.1        | File List . . . . .                         | 7         |
| <br>       |   |           |
| <b>6</b>   | <b>File Documentation</b>                   | <b>7</b>  |
| 6.1        | inc/Hardware.h File Reference . . . . .     | 7         |
| 6.1.1      | Detailed Description . . . . .              | 7         |
| 6.1.2      | Function Documentation . . . . .            | 8         |
| 6.2        | inc/Main.h File Reference . . . . .         | 8         |
| 6.2.1      | Detailed Description . . . . .              | 8         |
| 6.2.2      | Function Documentation . . . . .            | 8         |
| 6.3        | inc/X2cDataTypes.h File Reference . . . . . | 9         |
| 6.3.1      | Detailed Description . . . . .              | 10        |
| <br>       |   |           |
| <b>III</b> | <b>Used X2C-Blocks</b>                      | <b>11</b> |
| <br>       |   |           |
| <b>7</b>   | <b>Project Specific Blocks</b>              | <b>11</b> |
| <br>       |   |           |
| <b>8</b>   | <b>Internal Library Blocks</b>              | <b>11</b> |
|            | AutoSwitch . . . . .                        | 11        |
|            | Constant . . . . .                          | 12        |
|            | Delay . . . . .                             | 13        |
|            | I . . . . .                                 | 14        |
|            | Negation . . . . .                          | 15        |
|            | Sin3Gen . . . . .                           | 16        |
|            | SinGen . . . . .                            | 18        |

# Part I

## X2C Model

### 1 Version Information

#### 1.1 X2C

- X2Cfull: Version 1193

#### 1.2 Operating System

- OS: Windows 7 6.1

#### 1.3 Scilab

- Scilab: Version 5.5.2.1427793548
- Java: Version 1.6.0\_41

### 2 Model Structure

#### 2.1 Xcos Model

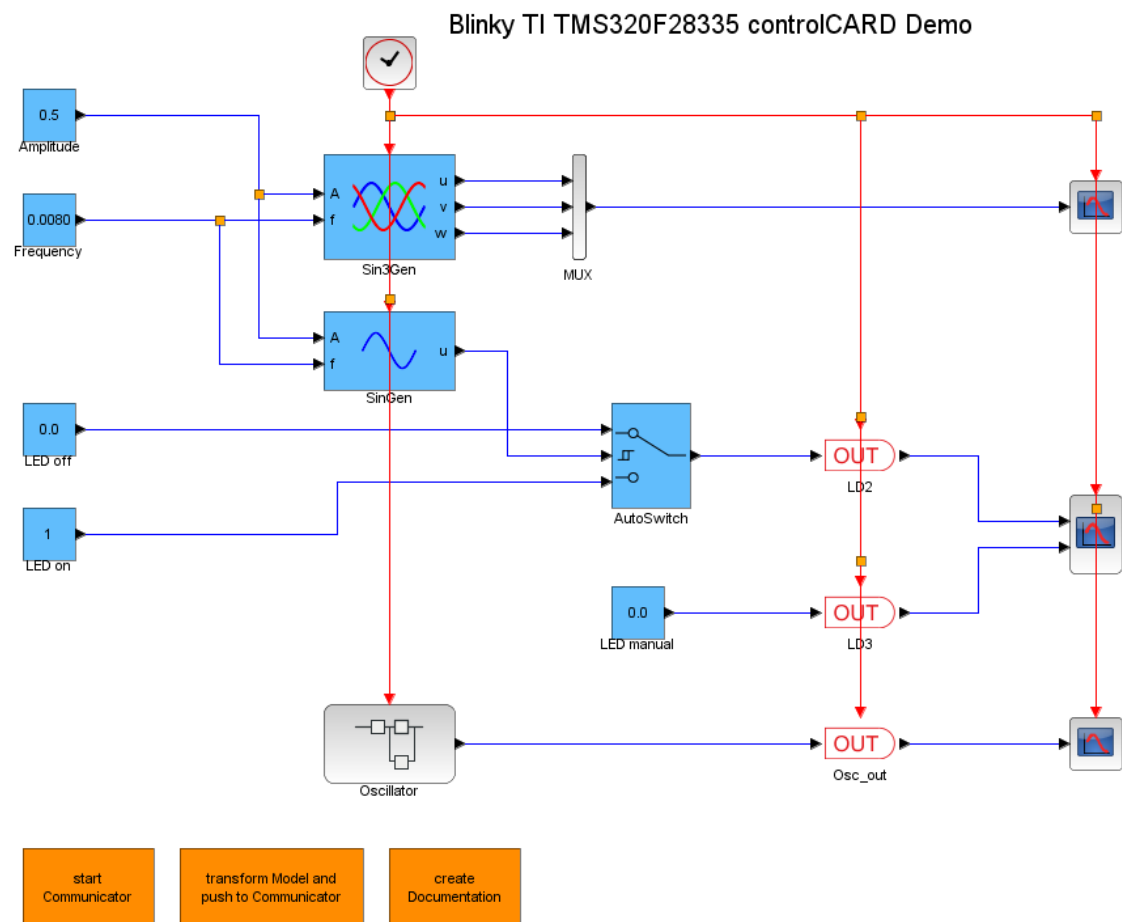


Figure 1: DemoApplication

## 2.2 Subsystems

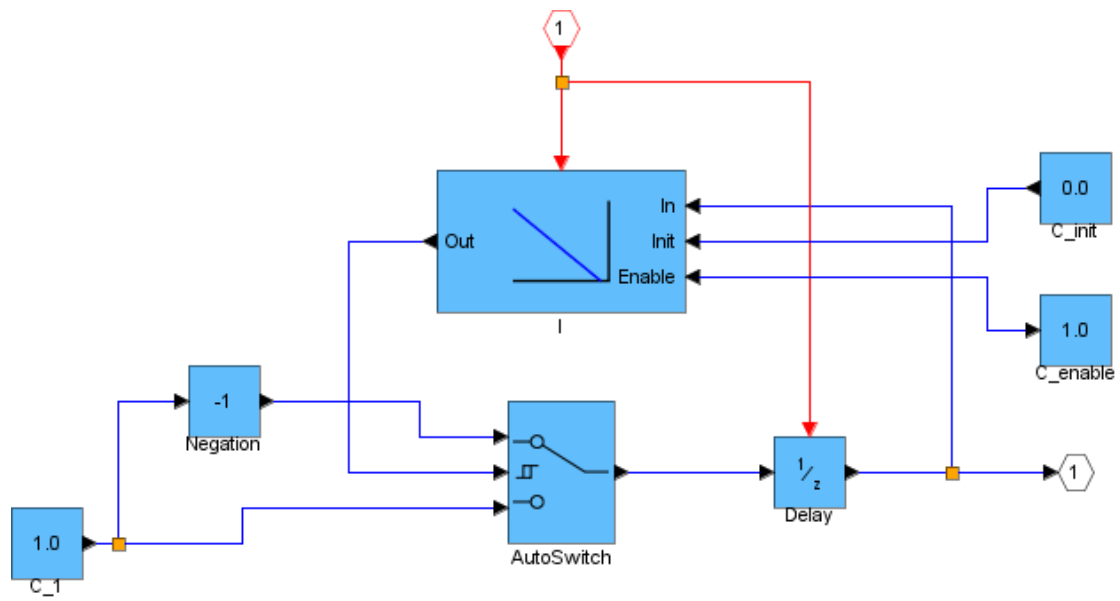


Figure 2: DemoApplication\_Oscillator

### 3 Model Parameter

#### 3.1 Sample Time

| Sample Time |            |
|-------------|------------|
| $T_S$       | $100\mu s$ |

## 4 Mask Parameter

| Constant: Amplitude |       |
|---------------------|-------|
| Value               | 0.5   |
| Used Implementation | FiP16 |

| AutoSwitch: AutoSwitch |       |
|------------------------|-------|
| Thresh_up              | 0.0   |
| Thresh_down            | 0.0   |
| Used Implementation    | FiP16 |

| Constant: Frequency |        |
|---------------------|--------|
| Value               | 0.0080 |
| Used Implementation | FiP16  |

| Constant: LED manual |       |
|----------------------|-------|
| Value                | 0.0   |
| Used Implementation  | FiP16 |

| Constant: LED off   |       |
|---------------------|-------|
| Value               | 0.0   |
| Used Implementation | FiP16 |

| Constant: LED on    |       |
|---------------------|-------|
| Value               | 1.0   |
| Used Implementation | FiP16 |

| AutoSwitch: Oscillator__AutoSwitch |       |
|------------------------------------|-------|
| Thresh_up                          | 0.5   |
| Thresh_down                        | -0.5  |
| Used Implementation                | FiP16 |

| Constant: Oscillator__C_1 |       |
|---------------------------|-------|
| Value                     | 1.0   |
| Used Implementation       | FiP16 |

| Constant: Oscillator__C_enable |      |
|--------------------------------|------|
| Value                          | 1.0  |
| Used Implementation            | Bool |

| <b>Constant: Oscillator__C_init</b> |       |
|-------------------------------------|-------|
| Value                               | 0.0   |
| Used Implementation                 | FiP16 |

| <b>Delay: Oscillator__Delay</b> |       |
|---------------------------------|-------|
| ts_fact                         | 1.0   |
| Used Implementation             | FiP16 |

| <b>I: Oscillator__I</b> |       |
|-------------------------|-------|
| Ki                      | 50.0  |
| ts_fact                 | 1.0   |
| Used Implementation     | FiP16 |

| <b>Negation: Oscillator__Negation</b> |       |
|---------------------------------------|-------|
| Used Implementation                   | FiP16 |

| <b>Sin3Gen: Sin3Gen</b> |        |
|-------------------------|--------|
| fmax                    | 1000.0 |
| Offset                  | 0.0    |
| ts_fact                 | 1.0    |
| Used Implementation     | FiP16  |

| <b>SinGen: SinGen</b> |        |
|-----------------------|--------|
| fmax                  | 1000.0 |
| Offset                | 0.0    |
| Phase                 | 0.0    |
| ts_fact               | 1.0    |
| Used Implementation   | FiP16  |

## Part II

# Frame Program Documentation

## 5 File Index

### 5.1 File List

Here is a list of all documented files with brief descriptions:

|                                    |                   |
|------------------------------------|-------------------|
| <a href="#">inc/Hardware.h</a>     |                   |
| Hardware configuration             | <a href="#">7</a> |
| <a href="#">inc/Main.h</a>         |                   |
| Main function                      | <a href="#">8</a> |
| <a href="#">inc/X2cDataTypes.h</a> |                   |
| X2C data type definitions          | <a href="#">9</a> |

## 6 File Documentation

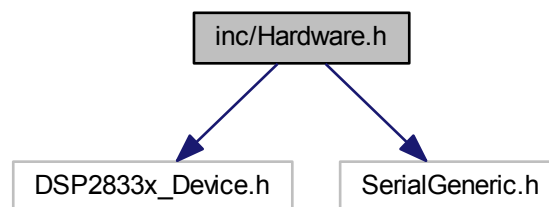
### 6.1 inc/Hardware.h File Reference

Hardware configuration.

```
#include <DSP2833x_Device.h>
```

```
#include "SerialGeneric.h"
```

Include dependency graph for Hardware.h:



### Functions

- void [initHardware](#) (void)  
*Hardware initialization.*
- void [initSerial](#) (tSerial \*serial)  
*Initialization of serial interface.*

#### 6.1.1 Detailed Description

Hardware configuration.



## 6.1.2 Function Documentation

### 6.1.2.1 void initHardware ( void )

Hardware initialization.

- Configuration of system clock and watchdog:
  - 30MHz external quartz
  - PLL
  - -> 150 MHz system clock
  - ~17 ms watchdog timeout
- Enable peripheral clocks
- Configuration of digital IOs
- Initialization of interrupts
- Configuration of ADC
- Configuration of PWM for interrupt generation

### 6.1.2.2 void initSerial ( tSerial \* *serial* )

Initialization of serial interface.

Parameters

|               |                          |
|---------------|--------------------------|
| <i>serial</i> | Serial interface object. |
|---------------|--------------------------|

## 6.2 inc/Main.h File Reference

Main function.

### Functions

- void [main](#) (void)  
*Main function.*
- void [mainTask](#) (void)  
*Main control task.*

### 6.2.1 Detailed Description

Main function.

### 6.2.2 Function Documentation

#### 6.2.2.1 void main ( void )

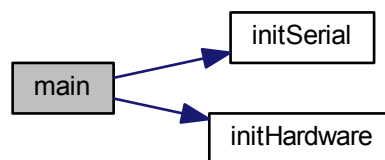
Main function.

## Returns

The main function will never return due to the never ending loop.

- Initialize "integrated monitor":
  - configuration of LNet protocol:
    - \* Node-ID: 1
    - \* Buffer size: 255
- Initialize serial interface
- Initialize hardware
- Initialize X2C
- Never ending loop -> interrupt driven algorithm

Here is the call graph for this function:



### 6.2.2.2 void mainTask ( void )

Main control task.

The main control task is called by the ADC interrupt service routine with a frequency of 10kHz.

- update X2C
- update outputs

## 6.3 inc/X2cDataTypes.h File Reference

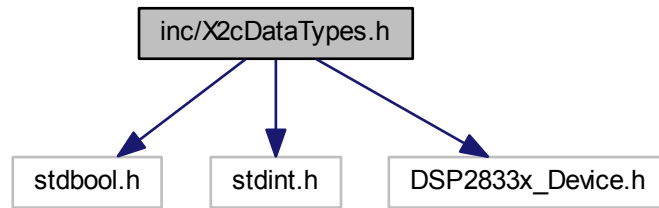
X2C data type definitions.

```
#include <stdbool.h>
```

```
#include <stdint.h>
```

```
#include <DSP2833x_Device.h>
```

Include dependency graph for X2cDataTypes.h:



### 6.3.1 Detailed Description

X2C data type definitions.

## Part III

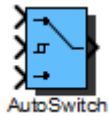
# Used X2C-Blocks

## 7 Project Specific Blocks

## 8 Internal Library Blocks

### 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  $\geq$  Threshold up  $\rightarrow$  Out = In1

Switch signal falling: Switch  $<$  Threshold down  $\rightarrow$  Out = 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 |

## Block: Constant

---



| Outputs |                 |
|---------|-----------------|
| Out     | Constant output |

| Mask Parameters |                 |
|-----------------|-----------------|
| Value           | Constant factor |

### Description:

Constant value.

### Implementations:

|                |                                      |
|----------------|--------------------------------------|
| <b>Bool</b>    | Boolean Integration                  |
| <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 |

## Block: Delay

---



| Inports |             |
|---------|-------------|
| In      | Input In(k) |

| Outputs |                       |
|---------|-----------------------|
| Out     | Output Out(k)=In(k-1) |

| Mask Parameters |   |
|-----------------|---|
| ts_fact         | Multiplication factor of base sampling time (in integer format) |

### Description:

Output delay by one sample time interval.

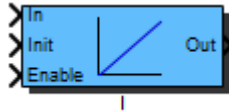
This block can be used to enable feedback loops in the model.

### Implementations:

|                |                                      |
|----------------|--------------------------------------|
| <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 |

## Block: I

---



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

| Outports |               |
|----------|---------------|
| Out      | Control value |

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

### Description:

I controller:

$$G(s) = K_i/s = 1/(T_i \cdot s)$$

Each fixed point implementation uses the next higher integer datatype for the integrational value storage variable.

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

Transfer function (zero-order hold discretization method):

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

### 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 |

## Block: Negation

---



| Inports |       |
|---------|-------|
| In      | Input |

| Outputs |                     |
|---------|---------------------|
| Out     | Negated input value |

### Description:

Negation of input signal.

Calculation:

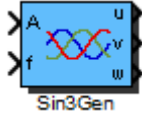
$$Out = -In$$

### 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 |



## Block: Sin3Gen



| Inports |           |
|---------|-----------|
| A       | Amplitude |
| f       | Frequency |

| Outputs |                          |
|---------|--------------------------|
| u       | Sine wave output phase u |
| v       | Sine wave output phase v |
| w       | Sine wave output phase w |

| Mask Parameters |   |
|-----------------|---|
| fmax            | Maximum Frequency in Hz   |
| Offset          | Offset  |
| ts_fact         | Multiplication factor of base sampling time (in integer format) |

### Description:

Generation of a 3 sine waves with amplitude (A) and frequency (f).

Calculation fixed point implementation:

$$\begin{aligned}
 u_k &= A_k \cdot \sin(2f_k \cdot f_{max} \cdot kT_S) + A_{Offset} \\
 v_k &= A_k \cdot \sin(2f_k \cdot f_{max} \cdot kT_S - \frac{2\pi}{3}) + A_{Offset} \\
 w_k &= A_k \cdot \sin(2f_k \cdot f_{max} \cdot kT_S + \frac{2\pi}{3}) + A_{Offset}
 \end{aligned}$$

For sine calculation a lookup table with 256 entries is used. This results in a short computation time but with the downside of reduced accuracy for the FiP32 implementation.

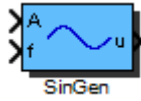
Calculation floating point implementation (parameter  $f_{max}$  is ignored):

$$\begin{aligned}
 u_k &= A_k \cdot \sin(2\pi f_k \cdot kT_S) + A_{Offset} \\
 v_k &= A_k \cdot \sin(2\pi f_k \cdot kT_S - \frac{2\pi}{3}) + A_{Offset} \\
 w_k &= A_k \cdot \sin(2\pi f_k \cdot kT_S + \frac{2\pi}{3}) + A_{Offset}
 \end{aligned}$$

**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 |

## Block: SinGen



| Inports |           |
|---------|-----------|
| A       | Amplitude |
| f       | Frequency |

| Outports |                  |
|----------|------------------|
| u        | Sine wave output |

| Mask Parameters |   |
|-----------------|---|
| fmax            | Maximum Frequency in Hz   |
| Offset          | Offset  |
| Phase           | Phase [-Pi..Pi]   |
| ts_fact         | Multiplication factor of base sampling time (in integer format) |

### Description:

Generation of a sine wave with amplitude (A) and frequency (f).

Calculation fixed point implementation:

$$u_k = A_k \cdot \sin(2f_k \cdot f_{max} \cdot kT_S + \phi_{Phase}) + A_{Offset}$$

For sine calculation a lookup table with 256 entries is used. This results in a short computation time but with the downside of reduced accuracy for the FiP32 implementation.

Calculation floating point implementation (parameter  $f_{max}$  is ignored):

$$u_k = A_k \cdot \sin(2\pi f_k \cdot kT_S + \phi_{Phase}) + A_{Offset}$$

### 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 |