MathComplex

IEC 61131 Library for ACSELERATOR RTAC® Projects

SEL Automation Controllers

Table of Contents

RTAC LIBRARY

MathComplex

Introduction

The MathComplex library allows the storage of complex numbers as well as basic manipulations that can be performed on them.

All numbers in polar notation are expected to be in units of degrees and not radians.

Supported Firmware Versions

You can use this library on any device configured using ACSELERATOR RTAC® SEL-5033 Software with firmware version R143 or higher.

Versions 3.5.0.2 and older can be used on RTAC firmware version R132 and higher.

Structures

Structures provide a means to group together several memory locations (variables), making them easier to manage.

struct_ComplexRect

This structure represents a complex number in rectangular coordinates.

Name	IEC 61131 Type	Description
Re	LREAL	The real component of this complex number.
Im	LREAL	The imaginary component of this complex number.

Functions

fun_ComplexAbs (Function)

This function returns the absolute value, or magnitude, of the provided complex number.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The number from which the absolute value is calculated.

Return Value

IEC 61131 Type	Description
LREAL	The absolute value of *num*.

Processing

This function returns a value equal to $\sqrt{(num.Re)^2 + (num.Im)^2}$.

fun_ComplexAdd (Function)

This function returns the sum of two complex numbers.

Inputs

Name	IEC 61131 Type	Description
num1	struct_ComplexRect	The first addend.
num2	struct_ComplexRect	The second addend.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The result of *num1* + *num2*.

fun_ComplexCmp (Function)

This function compares two complex values based on magnitude.

Inputs

Name	IEC 61131 Type	Description
num1	struct_ComplexRect	The first value to compare.
num2	struct_ComplexRect	The second value to compare.

Return Value

IEC 61131 Type	Description
INT	Returns –1 if the magnitude of *num2* is larger than *num1*, 0 if the magnitudes are equal, and 1 if the magnitude of *num1* is larger than *num2*.

Processing

This function calculates the absolute value of both provided inputs and then provides its return based on which one is larger.

fun_ComplexConjugate (Function)

This function returns the conjugate of the provided complex number.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The number from which the conjugate is derived.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The complex conjugate of *num*.

Processing

This function returns a struct_ComplexRect with a negated imaginary component.

fun_ComplexDivide (Function)

This function returns the quotient of *num1* / *num2*.

Inputs

Name	IEC 61131 Type	Description
num1	struct_ComplexRect	The dividend.
num2	struct_ComplexRect	The divisor.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The quotient of *num1* / *num2*.

Processing

This function performs the equivalent of $\frac{num1 \times num2^*}{num2 \times num2^*}$ providing a result formatted as a single complex number.

fun_ComplexExp (Function)

Compute e^{num}.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The number e will be raised to.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The result of e^{num}.

Processing

Performs the calculation $e^{a+bi} = e^a(\cos b + i \sin b)$.

fun_ComplexLn (Function)

Compute the natural logarithm of *num*.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The number from which the natural log is calculated.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The natural log of *num*.

Processing

This function returns an *Re* component of the natural log of the magnitude of *num* and an *Im* component of the angle defined by the arctangent of *num.Im* and *num.Re*.

fun_ComplexMultiply (Function)

This function returns the product of multiplying two complex numbers.

Inputs

Name	IEC 61131 Type	Description
num1	struct_ComplexRect	The first factor.
num2	struct_ComplexRect	The second factor.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The product of *num1* and *num2*.

fun_ComplexScale (Function)

This function multiplies a complex number by a scalar.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The number to scale.
scalar	LREAL	The scalar.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The product of *num* and *scalar*.

fun_ComplexSubtract (Function)

This function returns the difference of two complex numbers.

Inputs

Name	IEC 61131 Type	Description
num1	struct_ComplexRect	The minuend.
num2	struct_ComplexRect	The subtrahend.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	The difference of *num1 – num2*.

fun_ComplexZero (Function)

This function zeros the provided struct_ComplexRect.

Inputs/Outputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The struct to be zeroed.

struct_ComplexRect_TO_vector_t (Function)

This function converts a complex number stored as rectangular coordinates to a complex number stored as polar coordinates. The angle returned by this function will be between –180 and 180 degrees.

Inputs

Name	IEC 61131 Type	Description
num	struct_ComplexRect	The rectangular coordinates to be converted.

Return Value

IEC 61131 Type	Description
vector_t	*num* represented as polar coordinates.

vector_t_TO_struct_ComplexRect (Function)

This function converts a complex number stored as polar coordinates to a complex number stored as rectangular coordinates.

Inputs

Name	IEC 61131 Type	Description
num	vector_t	The polar coordinates to be converted.

Return Value

IEC 61131 Type	Description
struct_ComplexRect	*num* represented as rectangular coordinates.

Benchmarks

Benchmark Platforms

The benchmarking tests recorded for this library are performed on the following platforms.

- SEL-3505
  - R135-V0 firmware
- SEL-3530
  - R135-V0 firmware
- SEL-3555
  - Dual-core Intel i7-3555LE processor
  - 4 GB ECC RAM
  - R135-V0 firmware

Benchmark Test Descriptions

BasicOps Performance

The times fun_ComplexAbs, fun_ComplexZero, and fun_ComplexConjugate require to run, averaged over 1000 calls.

AlgebraicOps Performance

The times fun_ComplexAdd, fun_ComplexSubtract, fun_ComplexDivide, fun_ComplexScale, and fun_ComplexMultiply require to run, averaged over 1000 calls.

Exponential Performance

The times fun_ComplexExp and fun_ComplexLn require to run, averaged over 1000 calls.

Conversion Performance

The times struct_ComplexRect_TO_vector_t and vector_t_TO_struct_ComplexRect require to run, averaged over 1000 calls.

Benchmark Results

Operation Tested	Platform (time in μs)		
	SEL-3505	SEL-3530	SEL-3555
fun_ComplexAbs	10	7	1
fun_ComplexConjugate	5	4	1
fun_ComplexZero	1	1	1
fun_ComplexAdd	2	2	1
fun_ComplexSubtract	2	2	1
fun_ComplexMultiply	2	2	1
fun_ComplexDivision	3	2	1
fun_ComplexScale	1	1	1
fun_ComplexExp	36	15	1
fun_ComplexLn	50	13	1
vector_t_TO_struct_ComplexRect	19	6	1
struct_ComplexRect_TO_vector_t	23	12	1

Examples

These examples demonstrate the capabilities of this library. Do not mistake them as suggestions or recommendations from SEL.

Implement the best practices of your organization when using these libraries. As the user of this library, you are responsible for ensuring correct implementation and verifying that the project using these libraries performs as expected.

Performing a Mathematical Operation on Two Complex Numbers

Objective

A user has a need to sum two complex numbers.

Solution

The usage shown in *Code Snippet 1* would be the same for any other operation as well.

Code Snippet 1 prg_ComplexAdd

```
PROGRAM prg_ComplexAdd
VAR
    ComplexNumOne : struct_ComplexRect := (Re := 4.0, Im := 3.0);
    ComplexNumTwo : struct_ComplexRect := (Re := 2.0, Im := 7.0);
    ComplexResult : struct_ComplexRect;
END_VAR
```

```
ComplexResult := fun_ComplexAdd(ComplexNumOne, ComplexNumTwo);
```

Manipulating vector_t Structures

Objective

A user has data points received through an Axion module containing complex numbers represented in polar notation as vector_t objects.

The user would like to perform some set of mathematical operations on them and then prepare them to send to another device.

Solution

Code Snippet 2 prg_ComplexConvert

```
PROGRAM prg_ComplexConvert
VAR
    DatumOne : vector_t := (ang := 36.87, mag := 35_614);
    DatumTwo : vector_t := (ang := 53.13, mag := 17_735);
    ComplexNumOne : struct_ComplexRect;
    ComplexNumTwo : struct_ComplexRect;
    ComplexResult : struct_ComplexRect;
    DatumResult : vector_t;
END_VAR
```

```
ComplexNumOne := vector_t_TO_struct_ComplexRect(DatumOne);
ComplexNumTwo := vector_t_TO_struct_ComplexRect(DatumTwo);
ComplexResult := fun_ComplexAdd(ComplexNumOne, ComplexNumTwo);
//Scale from Kilovolts to Volts.
ComplexResult := fun_ComplexScale(ComplexResult, 1000);
DatumResult := struct_ComplexRect_TO_vector_t(ComplexResult);
```

Release Notes

Version	Summary of Revisions	Date Code
3.5.1.0	➤ Allows new versions of ACSELERATOR RTAC to compile projects for previous firmware versions without SEL IEC types "Cannot convert" messages. ➤ Must be used with R143 firmware or later.	20180921
3.5.0.2	➤ Corrected bug in `struct_ComplexRect_TO_vector_t()` that returned a value of NaN due to rounding errors. ➤ Defined the range of the angle returned by `struct_ComplexRect_TO_vector_t()` to be [–180, 180] degrees. Previous to this release, this was undefined but returning [0, 360].	20150925
3.5.0.1	➤ Fixed negative return values in `fun_ComplexAbs()`. This also corrected issues in `fun_ComplexCmp()` and `fun_ComplexLn()`.	20150722
3.5.0.0	➤ Initial release.	20150511