

Freescale Embedded Software and Motor Control Libraries

1 Introduction

Advanced motor drives are an integral part of many systems in appliance, industrial, and automotive applications. Due to their complexity, the development of motor control applications is an intricate and challenging task for developers. In order to shorten the development time and so the product time to market, Freescale offers different motor control reference designs for various MCU devices, applications, and motor types.

One of the fundamentals on which the motor-control reference designs are built is the set of Embedded Software and Motor Control Libraries. This document focuses on the introduction and description of the Embedded Software and Motor Control Libraries and provides an example of their use in an application.

2 Understanding the Embedded Software and Motor Control Libraries

The Embedded Software and Motor Control Libraries (ESMCLIB) are a broad group of algorithms covering basic mathematics, logic, controllers, digital filters, modulations, and transformations, including advanced observer algorithms.

Contents

1	Introduction.....	1
2	Understanding the Embedded Software and Motor Control Libraries	1
2.1	Description, supported devices, and compilers.....	2
2.2	Testing.....	3
2.3	ESMCLIB installation and integration into the user's application.....	4
3	Example	5
4	Conclusion.....	6
5	References.....	7
6	Acronyms and abbreviated terms.....	7
7	Revision history.....	7

This tool is very important for beginners creating simple applications, as well as for professionals working on complex real-time dynamic systems (such as motor control, power conversion, UPS, etc.). The Embedded Software and Motor Control Libraries were designed to utilize available core features, optimized for both code size and execution speed. The ESMCLIB algorithms do not access MCU peripherals.

2.1 Description, supported devices, and compilers

The software libraries were originally developed for Freescale's 568000E 16-bit family of digital signal controllers. Later the group of supported devices was expanded to include the 32-bit ColdFire® V1 and Kinetis ARM® Cortex™-M4 MCUs. Therefore, as the libraries evolved there arose slight differences between the library sets for the supported cores.

The ESMCLIB consists of four sub-libraries, because the algorithms are divided into four groups:

- General Functions Library (GFLIB): Basic trigonometric and general math functions, such as sine, cosine, tangent, hysteresis, limitations, controllers, and so on.
- Motor Control Library (MCLIB, GMCLIB): Standard algorithms used for motor control, such as Clarke/Park transformations, Space Vector Modulation, and other functions related to motor control.
- General Digital Filter Library (GDFLIB): Digital IIR, FIR, and moving average filters, designed to be used in a motor control application.
- Advanced Control Library (ACLIB): Advanced algorithms for rotor position estimation.
 - Back-EMF Observer in d/q and alpha/beta coordinates
 - Tracking Observer
 - Angle Tracking Observer

The ESMCLIB forms the layer architecture, where only the GFLIB and GDFLIB libraries are completely independent and can be used stand-alone. The GMCLIB and ACLIB libraries depend on the GFLIB and GDFLIB, and cannot be used as stand-alone.

Here are detailed lists of the functions contained in each library set.

- GFLIB
 - Sine
 - Cosine
 - Tangent
 - Arcus sine
 - Arcus cosine
 - Arcus tangent
 - Arcus tangent YX
 - Shifted arcus tangent YX
 - Square root
 - Ramp
 - Dynamic ramp (DSC only)
 - Limiter (3)
 - Hysteresis (DSC and Kinetis only)
 - Signum (DSC and Kinetis only)
 - Lookup table
 - PI controller (2)
 - PID controller (DSC only)
- MCLIB
 - Clarke transformation
 - Inverse Clarke transformation
 - Park transformation
 - Inverse Park transformation
 - Space vector modulation¹

1. ESMCLIB for DSC contains six different SVM algorithms.

- Vector limiter
- PMSM decoupling
- DC bus ripple elimination
- GDFLIB
 - FIR filter (Kinetis only)
 - IIR filter
 - Moving average filter
- ACLIB
 - Tracking observer
 - Angle tracking observer (ColdFireV1 and DSC only)
 - PMSM BEMF observer in D/Q
 - PMSM BEMF observer in alpha/beta (ColdFireV1 and DSC only)
 - Integrator

Each library uses its own data types and structures defined for each function. The compound data types for some algorithms (for example controllers and filters) also contain different coefficients or state values that must be stored for ensuing calculations. To improve performance, only the addresses of input data structures and calculation result structures are passed with the library function call.

The libraries use 16-bit (DSC) and 32-bit (ColdFire V1, Kinetis) fractional arithmetic. The calculations are performed with the numbers in Q0.15 or Q0.31 format, which means that all numbers are within the $(-1, 1)$ interval.

Using fractional arithmetic brings several important factors to application development. For example, multiplication never overflows. Also, because there is only one numerical format within the application, scaling of the physical quantities to a numerical format only needs to be performed once. While there is direct hardware support for fractional arithmetic on the DSC, for the ColdFire V1 and Kinetis the mathematical operations in fractional arithmetic are emulated by software.

Table 1 summarizes the supported devices and compilers.

Table 1. Supported devices and compilers

—	MCU family	Supported compiler
DSC	MC56F80xx	CodeWarrior™ Development Studio for Freescale DSC56800/E
	MC56F81xx	
	MC56F82xx	Digital Signal Controllers, version 8.3
	MC56F83xx	
	MC56F84xx	
ColdFire V1	MCF51xx	CodeWarrior Development Studio V6.1 ColdFireV1 AC256 ALPHA Service Pack
—	Kinetis ARM Cortex-M4	IAR Embedded Workbench for ARM ver. 6.2 CodeWarrior Development Studio for Microcontrollers v10.2

More information can be found on the ESMCLIB home page on freescale.com, including detailed documentation and the embedded software.

2.2 Testing

In order to validate the implementation of the ESMCLIB, each function was tested for proper calculation results. The functions were tested using a target-in-the-loop method in an automated process.

The testing routine is developed in the MATLAB® / Simulink® environment. For each function there is a Simulink reference model created. The input values of a function are entered from a test vector. These are fed into the Simulink model of the function and also, via an RS-232 interface, to the target MCU. There the tested function is calculated.

Bidirectional data transmission between the MATLAB and the embedded MCU is supported by the SFIO toolbox. The results obtained from the reference model and from the MCU are then compared and a test report is generated. The size of the test vector and the values are chosen according to the character of the function and the number of elements in the input and output structures.

In order to perform testing of a function in a reasonable time, if the input data structure contains many elements then it is not feasible to test the function through the whole 16- or 32-bit range of the input values. Therefore the function is checked at least at the boundary conditions. It is also checked for values at which the function could possibly saturate or overflow, or where a branch is evaluated. In such a case the values for the input test vector are carefully selected.

In any case, the test vector contains tens of thousands of points. Figure 1 shows the process of the ESMCLIB testing.

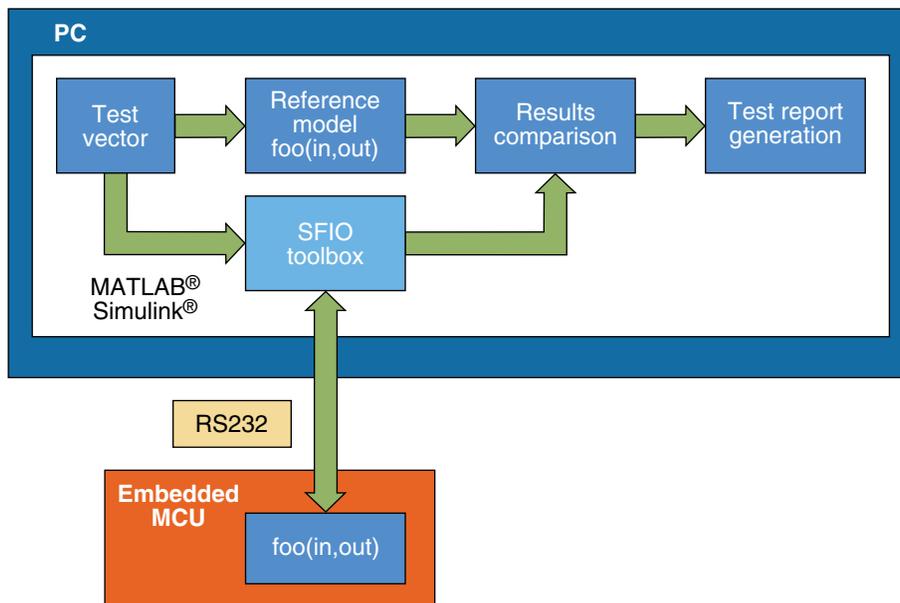


Figure 1. ESMCLIB testing process

2.3 ESMCLIB installation and integration into the user's application

The ESMCLIB is delivered as a single executable file. The inner structure of the ESMCLIB target directory differs between the individual MCU platforms, but generally can be divided into three parts:

1. All the header files, including the master header files of each library (gflib.h, gmclib.h, gdfplib.h, aclib.h) to be included in the user application.
2. Compiled binary library file to be included in the user application.
3. Documentation (user's manual).

All functions from the ESMCLIB are well documented. In addition to the API definition and a calculation description, the documentation also includes an example of how to use the library, as well as the size of the code and the execution clock cycles for each function.

As an example of how each function is documented, key points are picked from the section of the user reference manual describing the Clarke transformation:

Synopsis:

```
void MCLIB_ClarkTrf(MCLIB_2_COOR_SYST_ALPHA_BETA_T *pudtAlphaBeta,
MCLIB_3_COOR_SYST_T *pudtAbc)
```

Function arguments:

Name	Description
*pudtAlphaBeta	Pointer to a structure containing the data of a two-phase rotating orthogonal system, the MCLIB_2_COOR_SYST_ALPHA_BETA_T data type is defined in the header file MCLIB_types.h.
*pudtAbc	Pointer to a structure containing the data of a three-phase rotating system, the MCLIB_3_COOR_SYST_T data type is defined in the header file MCLIB_types.h.

User type definitions:

Typedef	Name of structure members	Format	Description
MCLIB_2_COOR_SYST_ALPHA_BETA_T	f16Alpha	Signed 16-bit fractional	Alpha component
	f16Beta		Beta component
MCLIB_3_COOR_SYST_T	f16A		A component
	f16B		B component
	f16C		C component

Performance:

Code size (bytes)	9	
Data size (bytes)	0	
Execution clock	Minimum	21 cycles (MC56F80xx), 22 cycles (MC56F83xx)
	Maximum	21 cycles (MC56F80xx), 22 cycles (MC56F83xx)

Integration of the library set into the user's application is different for each compiler. This is described in the documentation for the particular MCU family.

3 Example

A good example of using the Embedded Software and Motor Control Libraries is vector control of a permanent magnet synchronous motor (PMSM).

The motor control algorithm is divided into two periodically executed loops – a slow (speed) and a fast (current) control loop. Each loop consists of a chain of more or less standardized functions. Outputs of one function serve as inputs for the next function. The block diagram (Figure 2) highlights the parts of the algorithm that are contained in the ESMCLIB.

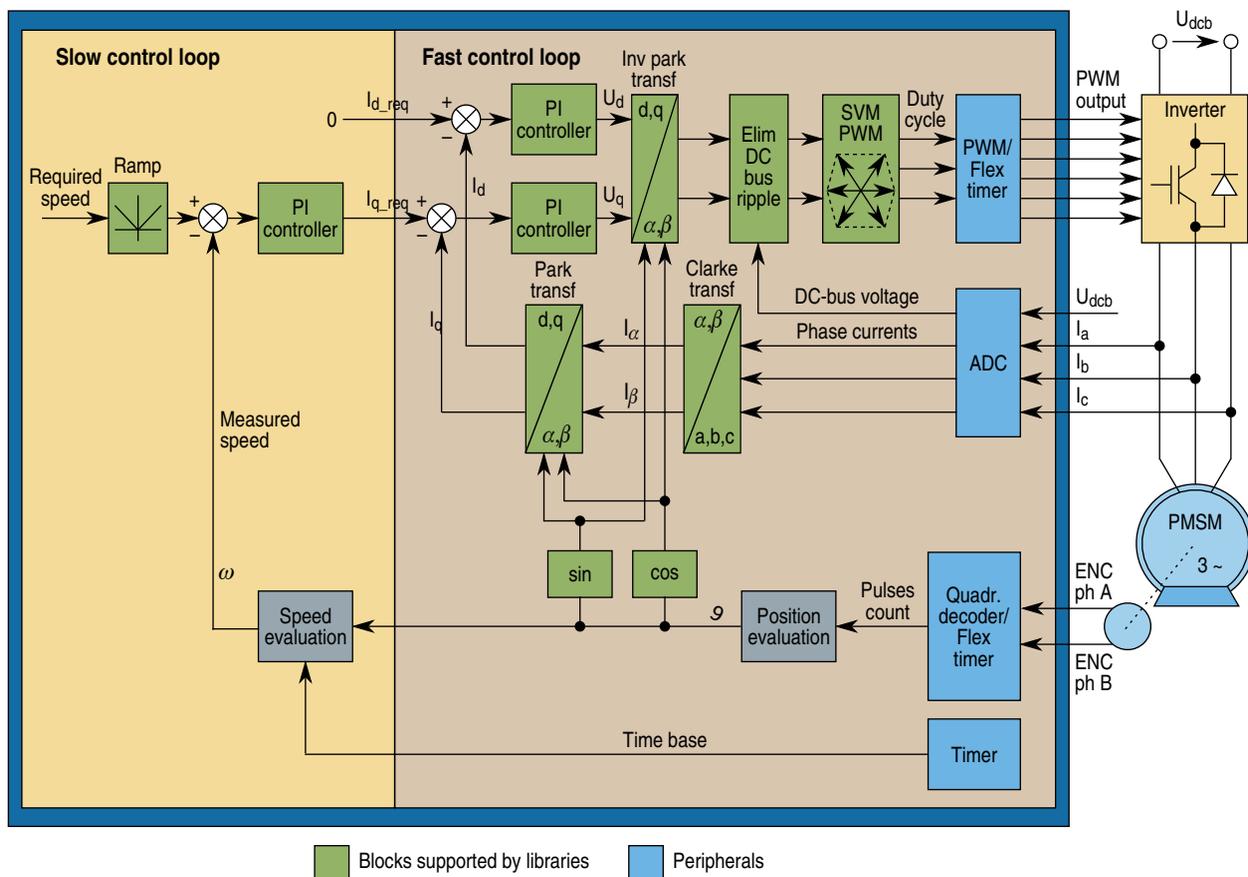


Figure 2. Block diagram of the PMSM vector control with encoder

As the section of code below demonstrates, in this particular motor control algorithm the coding of the fast control loop is, along with the necessary peripheral handling (reading the ADC results register, calculating the position and speed from the encoder signals, and updating the PWM duty cycle registers), reduced to only calling library functions and passing the addresses of the application structures.

```

....
// Iq current PI controllers
uDQReq.s32Arg2 = GFLIB_ControllerPIpAW(idQErr.s32Arg2, &qAxisPI);
// inverse Park trf for voltages
GMCLIB_ParkInv(&uAlBeReq, &thRotElSyst, &uDQReq);
// Elimination of DC bus ripple
elimDcbRip.s32ArgDcBusMsr = uDCBus;
GMCLIB_ElimDcbRip(&uAlBeReqDCB, &uAlBeReq, &elimDcbRip);
// Calculation of Standard space vector modulation
svmSector = GMCLIB_SvmStd(&pwm32, &uAlBeReqDCB);
....

```

4 Conclusion

The Embedded Software and Motor Control Libraries are a part of the enablement that help Freescale deliver a system solution to customers. Use of the software libraries, as well as other tools that Freescale offers, helps developers to accelerate application development and narrow the scope of their work to those technical challenges inherent in each application.

5 References

1. Freescale document MCLIBCORETXM4UG — *Set of General Math and Motor Control Functions for Cortex M4 Core, User Reference Manual*
2. Freescale document 56800E_MCLIB — *56800E Motor Control Library User's Reference Manual*
3. www.freescale.com/fslesl — *Embedded Software and Motor Control Libraries*

6 Acronyms and abbreviated terms

Table 2 contains abbreviated terms used in this document.

Table 2. Acronyms and abbreviated terms

Term	Meaning
ADC	Analog-to-digital converter
DSC	Digital signal controller
ESMCLIB	Embedded software and motor control library
PMSM	Permanent magnet synchronous motor
PWM	Pulse-width modulation

7 Revision history

Table 3. Revision history

Revision (Date)	Description
0 (Oct 2012)	This document was originally released in September 2012 as Freescale white paper UESMCLWP, with an identical title and content. It was determined that this document would be better classified as an application note and it has been released as AN4611.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2012 Freescale Semiconductor, Inc.