International Journal of Smart Home
Vol. 9, No. 8 (2015), pp. 29-36
http://dx.doi.org/10.14257/ijsh.2015.9.8.04

SPWM Implementation Using DSP Model Programming Based on Simulink

ChangyongY in, Xiaoyu Sun, Shuo Huang and Hongxia Yu

Shenyang Institute of Engineering, Shenyang 110136, China
** Corresponding author: culcrum@outlook.com*

Abstract

SPWM is an essential modulation algorithm in power electronic devices, but it is not easy for unskilled programmer or junior engineers to build the code effectively. Model programming with embedded target supporting package in Matlab/Simulink is a new method to replace the traditional code writing without complicacy. Comparing to the manual method, it is more effective with lower cost, and the most importantly, it is more reliable. This paper offers a popular asymmetric sampling SPWM prototype with algorithm deduction, codes and models from Matlab RTW-EC. Finally, a practical result which can prove the method to be functional is also provided. This may accelerate the power electronics developing project with reliability in projects.

Keywords*: SPWM, Model programming, Matlab, DSP*

1. Introduction

SPWM is an essential modulation algorithm to make AC power output in power electronic devices, which are widely used in generators [1], power converters [2] and flexible alternative current transmission systems (FACTS) [3]. However, it is not easy for unskilled programmer or junior engineers to implement the algorithm on embedded platforms (such as DSP and several MCUs). Obviously, engineers in power electronics have uneven programming techniques, and this may lead to the unreliable problems to the developing system. In other words, the traditional manual code building method has expectable risk to the developing system more or less. If some method can make the engineers focus on the developing ideas but not on the programming skills, it will speed up the project and reduce the total cost.

Model programming is an efficient method to this problem that it combined the reusable standard code modules with embedded system integrated developing environments (IDE). The structure of this idea in the recent Matlab/Simulink version is in Figure 1. And some characteristic comparison between the new method and manual programming is in Table 1. It is believed that the model programming can offer auto-built codes which can be at least the same efficient as the manual method by skilled programmer in execute efficiency, and the testing data from Visteon Corp. showed that the auto-built code has 5% less requirement in both RAM and ROM space while it saved more than 50% time. [4] And this idea is proven to be more economic and maintainable by various interactive reports in large industrial developing projects with a famous illustration such as F35 fighter.

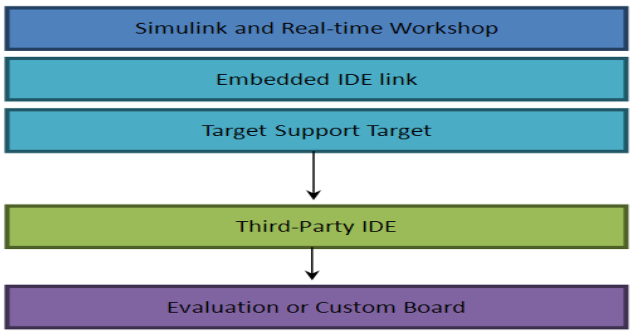


Figure 1.The Structure of Embedded IDE Link to 3rd-part IDE and Hardware

Table 1.Character Comparison between Manual Programming and Model Programming

	Reliability	Code Efficiency	Developing Period	Testing Methods	Developing Tools
Manual Programming	Uneven	Uneven	days	Manual and messy steps	Single IDE
Model Programming	High	High	Hours	Hardware in Loop Test with real-time response	Various 3rd-part IDEs

2. SPWM Algorithm

SPWM is to use the slope wave to sample the sine wave in order to calculate every intersection, so that the switches can output the pulse width which is the variation of sine wave. The frequency of the slope wave is N (N is 3 or several times of 3) times of that of sine wave, and the sample value controls the switches to turn on and off. The common sample methods of SPWM are nature sampling, symmetry regular sampling, and asymmetry regular sampling. Among them, asymmetry sampling is the most popular due to its simple calculation, which is suitable for real-time signal processing and the low harmonic component. The main idea of asymmetry regular sampling is shown in Figure 2.

According to the figure, sine wave need to be sampled twice in a slope wave period, one is on the top and the other is on the bottom. And the pulse width calculations depend on the two sample values is asymmetry. That is where its name comes. In Figure 2, t_{on} and t_{on}' are the switch on time, while t_{off} and t_{off}' are the switch off time; T_s is the sample interval, it's also the half of the slope wave period T_c

$$T_s = \frac{T_c}{2} = \frac{2\pi}{\omega N} \tag{1}$$

According to the similar triangle,

$$\begin{cases} t_{on} = \frac{T_s}{2}(1 + m \sin \omega t_1) \\ t_{off} = \frac{T_s}{2}(1 - m \sin \omega t_1) \\ t_{on}' = \frac{T_s}{2}(1 + m \sin \omega t_2) \\ t_{off}' = \frac{T_s}{2}(1 - m \sin \omega t_2) \end{cases} \tag{2}$$

Where, m is the modulation rate, which is the amplitude rate of sine wave and slope wave, and 0<m<1.

$$m = \frac{U_{\sin\max}}{U_{c\max}} \tag{3}$$

According to deduce above, it's necessary to make the t_1 and $t_2$discrete to get the pulse width suitable for DSP calculation. The width of k period is t_w, while k is from 1 to N.

$$\begin{aligned} t_w = t_{on} + t_{on}' &= \frac{T_s}{2}\left\{2 + m\sin[\frac{\pi}{2N}(4k-3)] + m\sin[\frac{\pi}{2N}(4k-1)]\right\} \\ &= \frac{T_c}{4}\left\{2 + m\sin[\frac{\pi}{2N}(4k-3)] + m\sin[\frac{\pi}{2N}(4k-1)]\right\} \end{aligned} \tag{4}$$

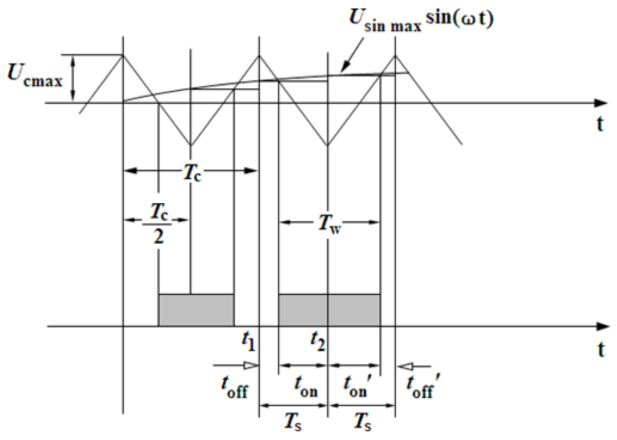


Figure 2. SPWM Asymmetry Sampling Operational Principle

3. SPWM Model Designing

TI's DSP TMS320F2812 is a popular chip widely used in power electronics and robots. The main reason is that its performance is good when running at the speed of 150MHz, and its peripheral resources are abundant. As a result, Matlab/Simulink offered a target support package of this chip. An engineer without any high level programming technique can build a complex prototype by the models in the package, and then he can modify the settings in RTW-EC tool to build the codes automatically. The main process of the model programming is in Figure.3. Obviously, the hardware and IDE and the software are linked together in all steps, hence the verification and test could be hardware in loop (HIL). It can be easily deduced that the codes reliability can be reasonable.

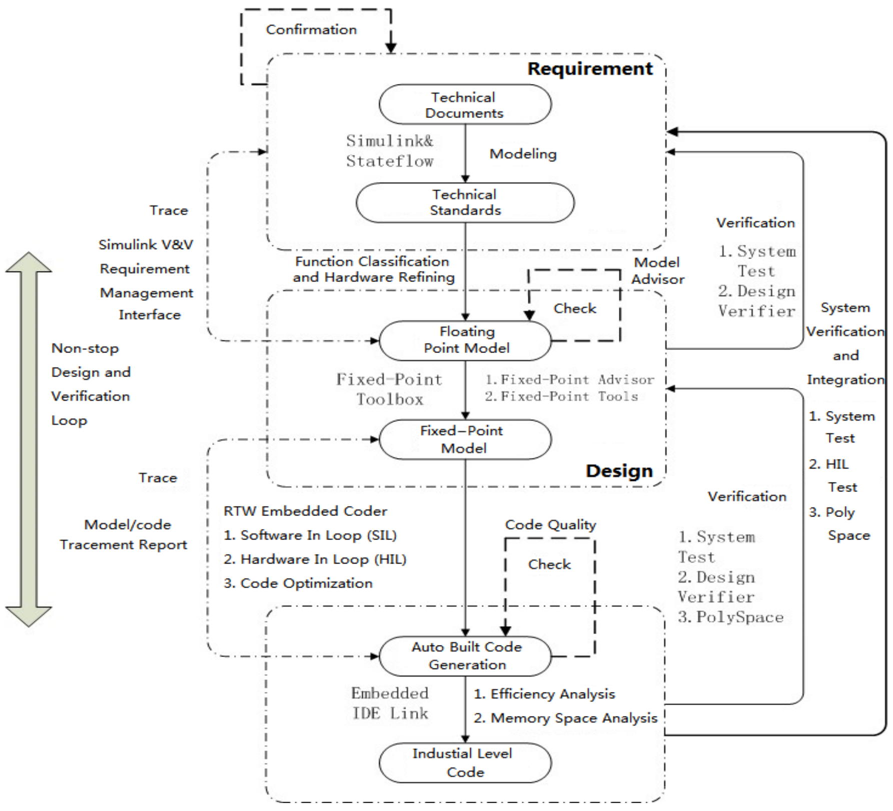


Figure 3. The Model Programming and Testing Flow Diagram [6]

According to the principle of SPWM, the quick prototype can be built as Figure.4. The sample slope wave is generated by timer 1 of Event A module (EVA), which is configured in the PWM model. The interrupt model configures the two hardware interrupts: one calculates the CMPR of EVA, which refers to the pulse width of a carrier wave period, at the time when the slope reaches the bottom; the other is the sample point which happens on the top or bottom of the slope wave, which is generated by timer 2 of EVA. And the timer 1 period (register PR) is twice of timer 2 PR because of the two samples in a period. As the period is set to 1ms, the PR can set as 37500. So EV Timer is used to configure timer2's interrupt, and it happens at the underflow of timer2. The two Data store Memory block are two global variations, "k" counts the steps in a sine period, and "Flg" signs the sample whether is on the top or bottom of the slope wave.

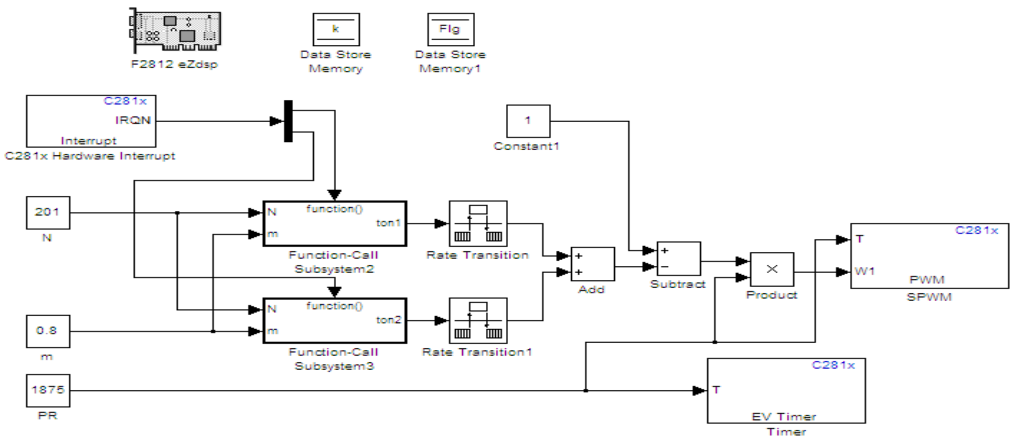


Figure 4. The SPWM Model Prototype

Function-Call subsystem is set to calculate t_{on} and t_{on}' in a period. The inner structure of it is shown in Figure.5. Function-Call subsystem 1 is to calculate the CMPR of EVA to generate PWM. The CMPR can be present as the equality (5), where the T_{EVA} is the clock period of EVA.

$$CMPR = PR - \frac{t_{on} + t_{on}'}{2T_{EVA}} \tag{5}$$

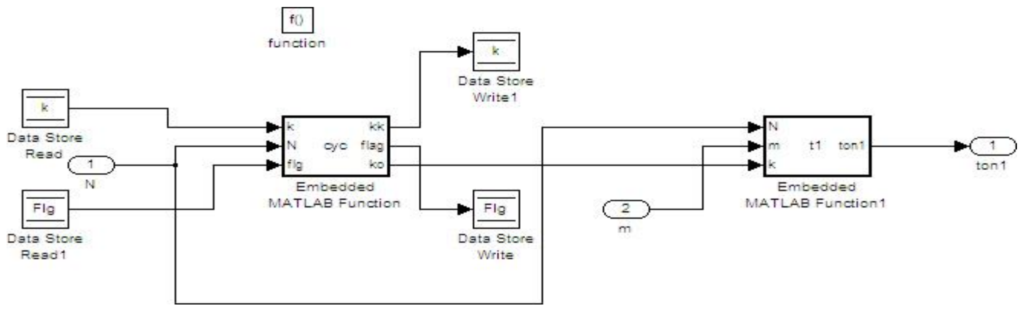


Figure 5. The Subsystem 2 Structure

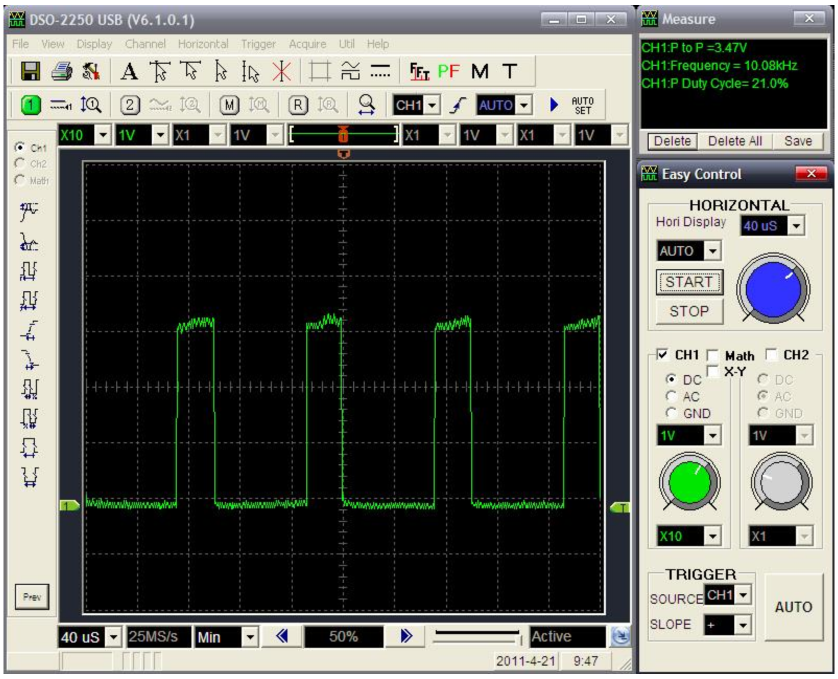
The two embedded Matlab functions in Figure 5 are used to calculate the samples. Because the structures of them are similar, one module which was written in m language of Matlab in order to adapt the steps parameter k is given as below:

```
if k<=N
  ko=k;
ifflg==0
  kk=k;
  flag=1;
else
  kk=k+1;
  flag=0;
  end
else
  ko=1;
  kk=1;
  flag=0;
end
```

4. Debugging and Implementation

After linking the hardware and the PC, the HIL debugging could be available. To generate the code automatically, the parameters configuration is essential. In “solver” option, stop time is set to be “inf” due to the fix point calculation DSP TMS320F2812[7], and solver type is “Fix-step”, while the “discrete” solver has to be selected. In “Real-Time Workshop” option, system target file is set to be “idelink_ert.tlc” to drive the TI CCS IDE, or other targets can be chosen to adapt to various CPU type. In “Embedded IDE Link” option, build action is selected to be “build”, and compiler options string can “get from IDE”, and system stack size (MAUs) is selected to be “512”. The other options can be default, if the space is not abundant, the optimization option can be adjusted to “o2” level. But to reach the top level of “o3” may have the message “cmdoverload” displayed in the status bar, and then some problems such as single step debug fail may come as a result. After just several seconds, the auto built code can be running in the DSP chip, the SPWM waveform can be seen on the oscilloscope as Figure.6, where the SPWM carrier wave is 10kHz and the sinusoidal wave is 50Hz. Hence, the model programming code can be proved to be functional.

During the debug process, model advisor report tool is helpful to deal with the possible mistakes. If the compiling result is not acceptable, the tlc report (*.html) can be helpful to adjust the tool chain (Matlab/Simulink – RTW – EC – CCS) configurations. Also the model advisor can monitor all the variations and their executing time in the models, so that all models can be set breakpoints to improve the codes according to the reports. Figure.7 shows the executing times distribution of all subsystems running in 2ms, while the DSP is running at the full speed of 150MHz. And Table.2 shows the measurement of the executing time. The algorithm can be further optimized by IQmath library due to the most time consumed by the trigonometric function calculation, but the code is fast enough in this application.



CH1:P to P =3.47V
CH1:Frequency = 10.08kHz
CH1:P Duty Cycle= 21.0%

Hori Display 40 uS
AUTO
☑ CH1 ☐ Math ☐ CH2 ☐ X-Y
CH1: DC; 1V; X10
CH2: AC; 1V; X1
SOURCE CH1
SLOPE +
40 uS 25MS/s Min 50% Active
2011-4-21 9:47

Figure 6. SPWM Waveform

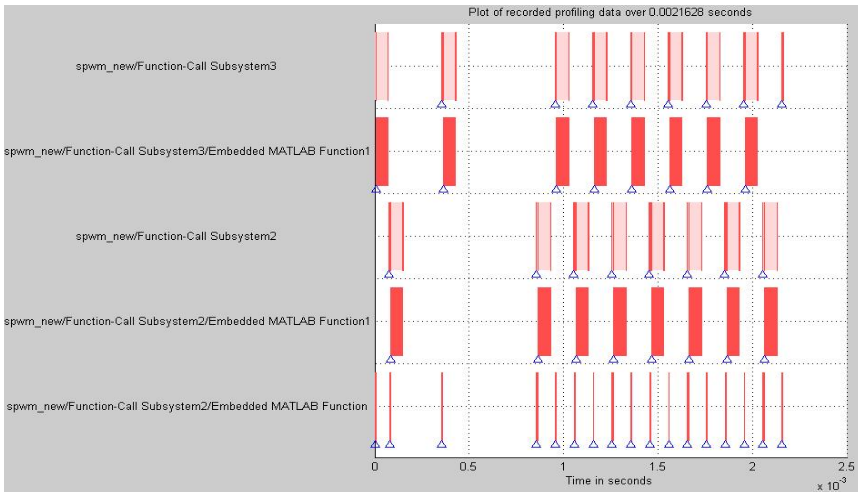
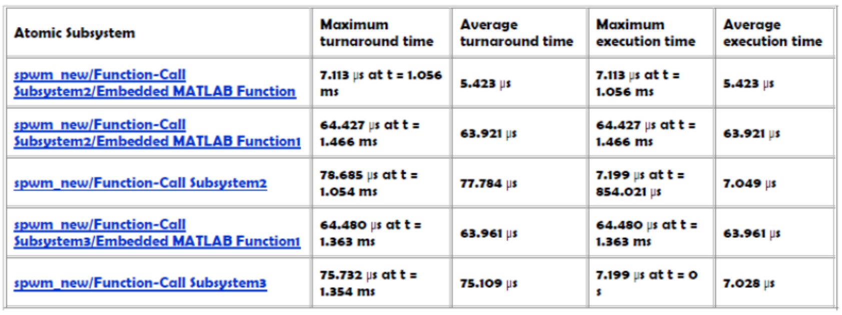


Figure 7. Subsystem Executing Time Distribution

Table 2. Subsystem Executing Time Measurement in SPWM Model



Atomic Subsystem	Maximum turnaround time	Average turnaround time	Maximum execution time	Average execution time
spwm_new/Function-Call Subsystem2/Embedded MATLAB Function	7.113 µs at t = 1.056 ms	5.423 µs	7.113 µs at t = 1.056 ms	5.423 µs
spwm_new/Function-Call Subsystem2/Embedded MATLAB Function1	64.427 µs at t = 1.466 ms	63.921 µs	64.427 µs at t = 1.466 ms	63.921 µs
spwm_new/Function-Call Subsystem2	78.685 µs at t = 1.054 ms	77.784 µs	7.199 µs at t = 854.021 µs	7.049 µs
spwm_new/Function-Call Subsystem3/Embedded MATLAB Function1	64.480 µs at t = 1.363 ms	63.961 µs	64.480 µs at t = 1.363 ms	63.961 µs
spwm_new/Function-Call Subsystem3	75.732 µs at t = 1.354 ms	75.109 µs	7.199 µs at t = 0 s	7.028 µs

5. Conclusion

This paper offers a SPWM algorithm quick prototype by Matlab/SimulinkModel programming, and an implementation with DSP has been tested, and the result has been analyzed by the helpful tools. As a result, the method is proved to be more efficient and reliable that it may accelerate the developing project of power electronics with lower cost but higher maintainability.

Acknowledgements

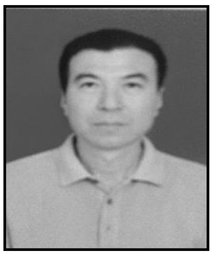
This work was supported by the Education Department of Liaoning Province, China, under the normal project L2014526.

References

[1] J. G. Li and X. D. Wang,“Two-phase Hybrid Stepping Motor SPWM Mini-step Drives Based on DSP”, S&N Electric Machines, vol. 29, no. 31,**(2002)**.
[2] L. Liu and X. Z. Qian,“Modeling and Simulation of Three Phase Full Bridge SPWM Inverter based on Matlab”, Electronic Design Engineering, vol. 22, no. 139,**(2014)**.
[3] X. F. Yang and Z. Q. Lin,“A Review of Modular Multilevel Converters”, Proceedings of the CSEE, vol. 33, no. 1,**(2013)**.
[4] W. Stuart and T. Erkkinen,“Reducing CUE Software Cost by Multi-object Modeling Technology”, EDN, **(2006)**.
[5] “The MathWorks Inc. Embedded IDE Link 4 User’s Guide for Use with Texas Instruments' Code Composer Studio”, **(2013)**.
[6] J. Liu,“Module Designing and Embedded System Implementation” Beijing University of Aviation and Aerospace Press,**(2010)**.
[7] K. F. Su,“TMS320X281X DSP Application System Design”, Beijing University of Aviation and Aerospace Press,**(2008)**.
[8] Q. Zhang, D. P. Shan, Z. D. Cheng and Z. G. Li,“Software Design of Asymmetry Rule Algorithm of Multitasking based on TMS320F2812”, Power Electronics, vol. 43, no. 78,**(2009)**.
[9] S. M. Wan,“Principles of TMS320F281X DSP and Application Examples”, Beijing University of Aviation and Aerospace Press,**(2007)**.
[10] X. Y. Xia and W. Tang,“Design of Single Phase Photovoltaic Grid-connected Inverter Based on DSP Control”, Journal of Electric Power Science and Technology, vol. 26, no. 52, **(2011)**.

Authors

ChangyongYin, received his master degree from Liaoning University in intelligent electrical load switch technique, and now he is a professor in College of International Education, Shenyang Institute of Engineering. His research interests include power electronics, FPGA embedded system, intelligent automation instruments, and smart grid.



XiaoyuSun, received his MBA degree from Liaoning University, mainly engaged in electrical device with intelligent controlling based on FPGA and soft core CPU, and now he is an associate professor in in College of International Education, Shenyang Institute of Engineering. His research interests include automation instruments, electronics, embedded system, and electrical systems.

ShuoHuang, received his master degree from Northeastern University in 2009, and now he is a lecturer and also a program coordinator in College of International Education, Shenyang Institute of Engineering. His research interests include power converters, smart grids, embedded systems, and power electronic simulations.

HongxiaYu, received her master degree from Northeastern University in 2008, and now she is a lecturer and also a program coordinator in College of International Education, Shenyang Institute of Engineering. Her research interests include PLCs, intelligent control methods, and automation theories.