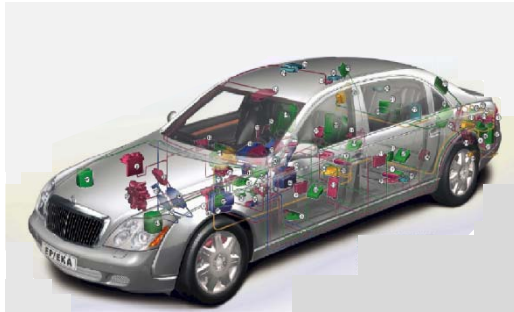


Automatic Code Generation for Embedded Control Systems

Tiffany Liang
Application Engineer
MathWorks

Embedded Control Systems



Automotive



Aero/Def



Construction
Machine



Power
Electronics

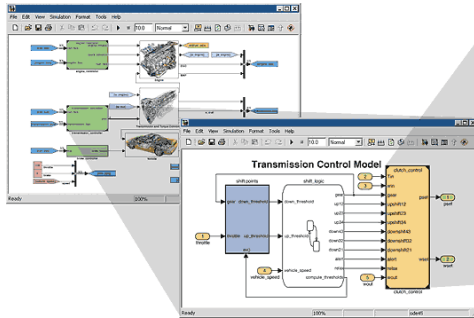


Industrial
Equipment

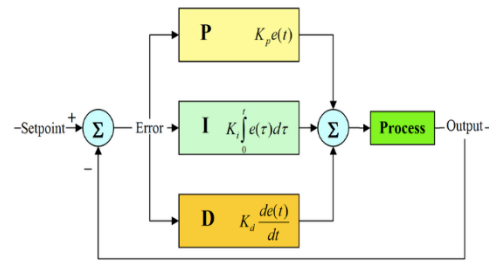


Robot

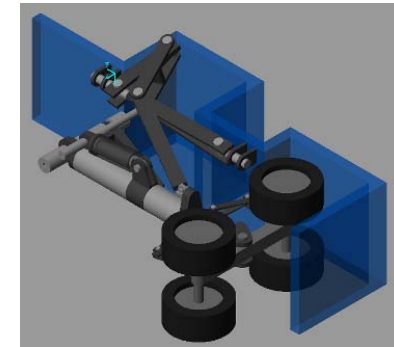
Model-Based Design “From Concept to Code”



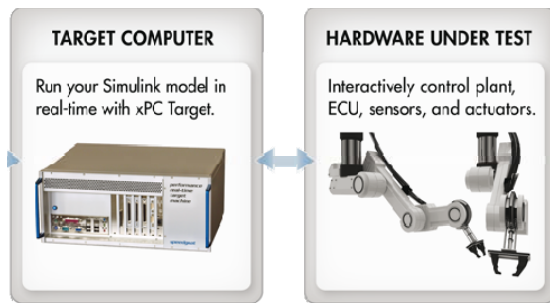
Modeling & Simulation



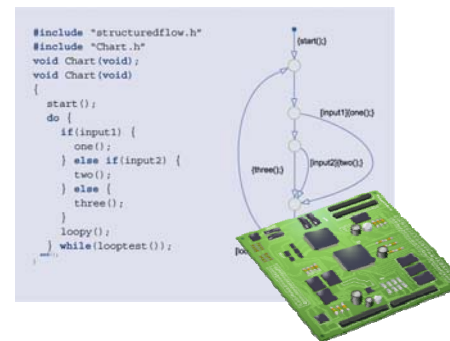
Control Design



Physical Modeling



Real-Time Test



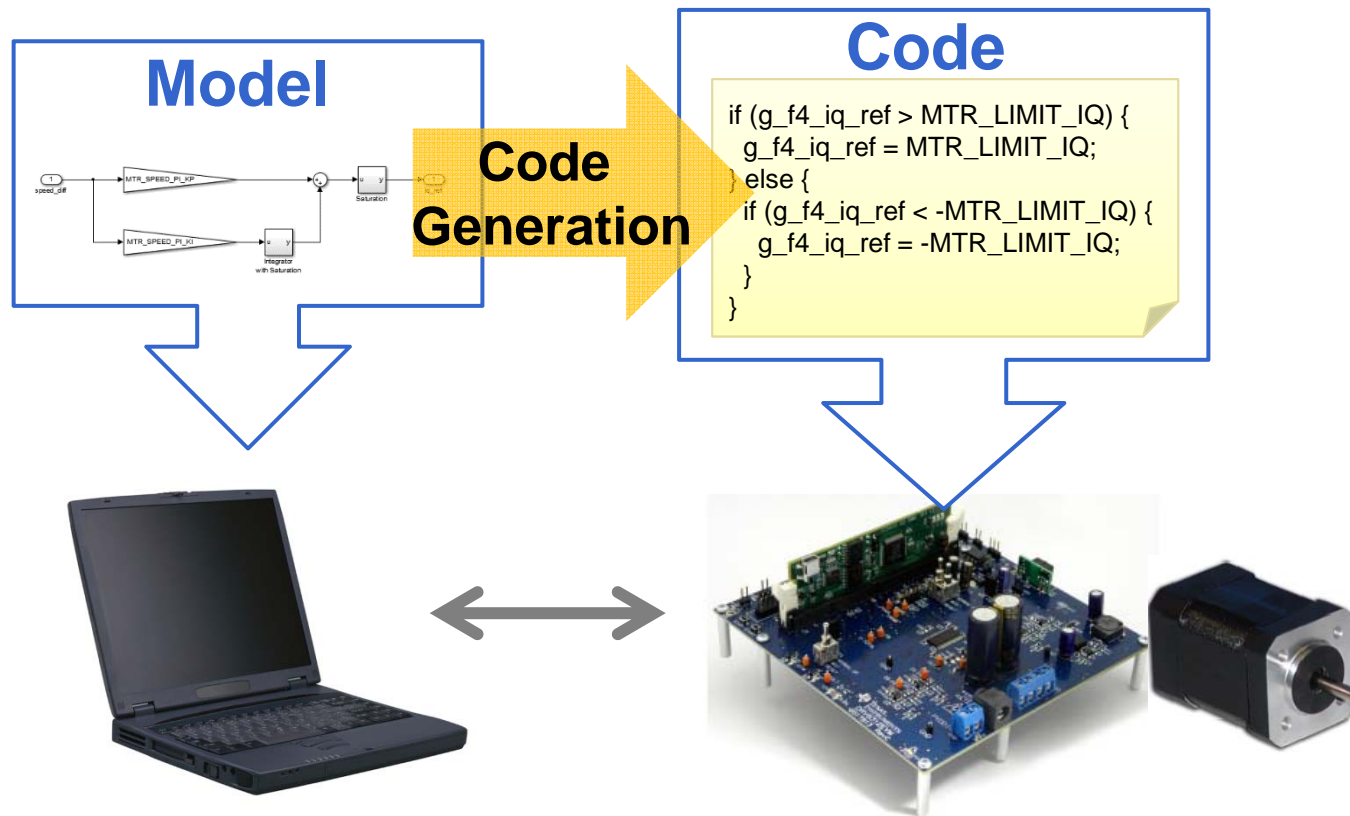
Code Generation



Verification & Validation

Quick MBD Overview

Brushless DC Motor Control



TI F28069ISO + DRV8312-C2-KIT used

モーター制御システム外観

Example on YouTube MATLAB Channel



The screenshot shows a YouTube video player with the following content:

- Video Title:** Programming TI C2000 Launchpad with Simulink
- Channel:** MATLAB (35,085 subscribers)
- Views:** 7,777
- Published:** 2014/03/14 to public
- Description:** Program TI C2000 LaunchPad using Simulink® for your real-time control programming applications. To Download the files used in this video: <http://www.mathworks.com/matlabcentra...>
- Next Video List:**
 - Matlab-C2000 LaunchPad (Lokman Gokdere, 4,850 views, 26:16)
 - Programming DSP TMS320F28335: Part1 - Creating a first project with (THN Channel, 2,211 views, 26:16)
 - OWON SDS7102 Complete Review (Mark Madel, 62,909 views, 33:46)
 - Simulink code generation from MATLAB for TI C2000 processors - (MATLAB, 17,471 views, 18:36)
 - Dummy load for BLDC controller testing (Oleg Mazurov, 7,445 views, 3:18)
 - Simple DC Motor Explained (Mutley5000, 178,409 views, 2:59)
 - STM32 Matlab PIL Simulation (John Shamu, 5,932 views, 8:11)

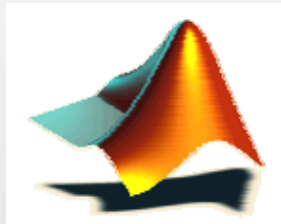
<https://www.youtube.com/watch?v=wxYTLbYfBP0>

How to Embed Your Model? Coder Solution!

MATLAB Coder[®]

MEX

C



MATLAB

Simulink Coder[®]

C



RCP/HILS

HDL Coder[™]

HDL



FPGA/ASIC

Embedded Coder[®]

C/C++



MCU/DSP

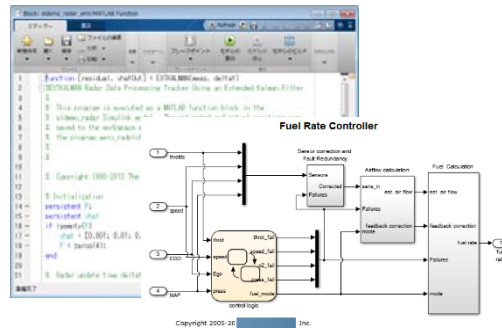
Simulink PLC Coder[™]

ST



PLC


C/C++ Code Development with Embedded Coder



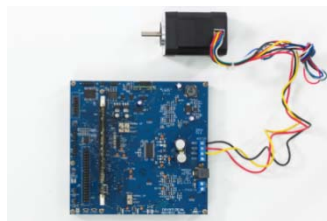
MATLAB Program
Simulink/Stateflow Model

```

if (reset) {
    y = 0;
} else {
    y += k * u;
}
    
```



Code Generation



- Used with existing code
- Used for implementation
- Standalone executable/dll

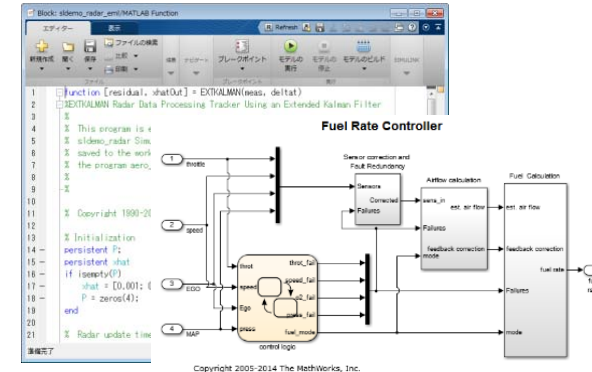
What is Your Merit?

Modeling & Simulation

- Graphical modeling
 - Readable, easy understanding
- Early verification & validation
 - Concept test without hardware

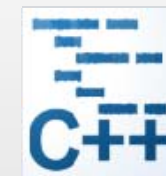
Automatic Code Generation

- Coding time reduction
- Error reduction from model to code conversion
- Automatic synchronization between model and code



```

if (reset) {
    y = 0;
} else {
    y += k * u;
}
    
```



User Story

Model-based design was a key enabler to Volt's rapid development.



Credit: General Motors LLC 2011

With its more than 100 electronic controllers and about 10 million lines of computer code, the Volt significantly raised the level of controls and systems-integration challenges for GM engineers.

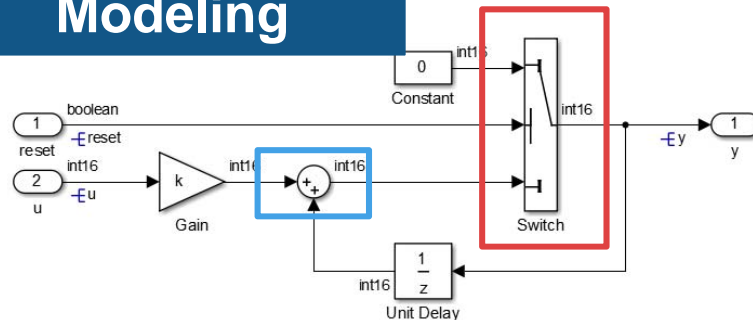
“We have a single source for how a particular function should behave. Automatic code generation using The MathWorks’ Real-Time Workshop Embedded Coder was vital to meeting Volt’s aggressive program timing.”

Greg Hubbard
Senior Manager

Nearly 100% of the software for many of Volt’s modules was generated automatically.

Embedded Coder Quick Demo

Modeling



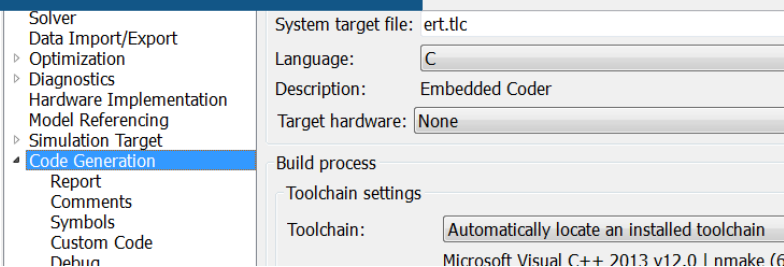
```

boolean_T reset;
const volatile int8_T k = 2;

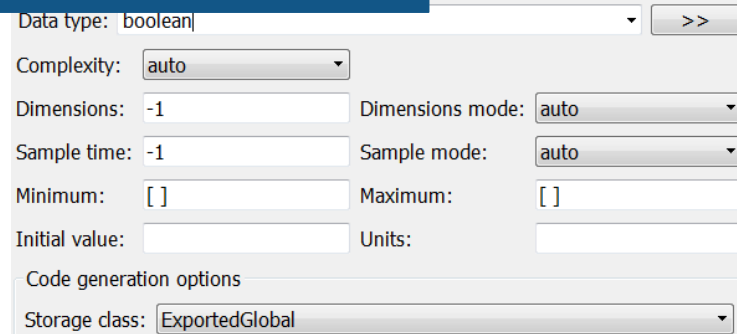
void rst_cntr_step(void)
{
    if (reset) {
        y = 0;
    } else {
        y += (int16_T)(k * u);
    }
}

```

Configuration

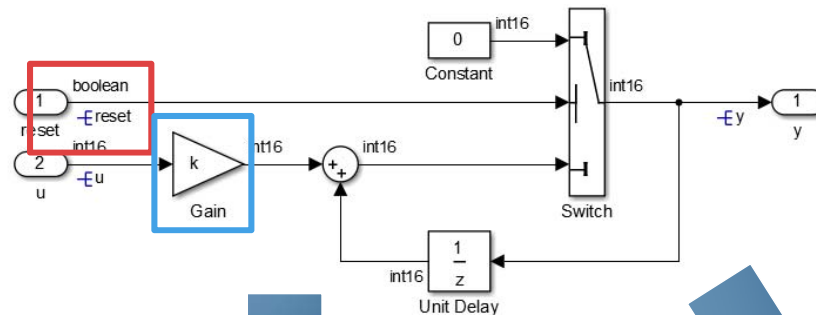


Data Attributes



- Equivalent Code as Model
- High Readability & Efficiency

Data Attributes



- You can define data attributes in your model
 - Signal line
 - States (Unit Delay etc.)
 - Block parameter
 - Stateflow data

```

boolean T reset;
const volatile int8_T k = 2;

void rst_cntr_step(void)
{
    if (reset) { Global Variable
        y = 0;
    } else {
        y += (int16_T)(k * u);
    }
}
    
```

```

void rst_cntr(
    boolean_T reset, Function Argument
    int16_T u,
    int16_T *y)
    
```

Code Generation Report

The screenshot shows the Code Generation Report interface with the following components:

- Code Metrics (lines, memory size, etc.):** A callout pointing to the 'Contents' sidebar on the left, which lists sections like 'Code Metrics Report', 'Traceability Report', 'Static Code Metrics Report', and 'Code Replacements Report'.
- Generated Code:** A callout pointing to the 'Generated Code' sidebar on the left, which lists files such as 'ert_main.c', 'rst_cntr.c (1)', 'rst_cntr.h', and 'rtwtypes.h'.
- HTML report:** A callout pointing to the main code editor window, which displays C code for a reset function. The code includes comments for traceability, such as '/* Switch: <Root>/Switch incorporates: * Constant: <Root>/Constant * Gain: <Root>/Gain * Inport: <Root>/reset * Inport: <Root>/u * Sum: <Root>/Sum1 * UnitDelay: <Root>/Unit Delay */'.
- Traceability between Model and Code:** A callout pointing to the Simulink model view at the bottom, which shows a block diagram with blocks for 'Gain', 'Unit Delay', and 'Switch'. A large blue double-headed arrow connects the code editor to the model view, indicating the traceability between the two.

Simulink Report Generator required to generate model view

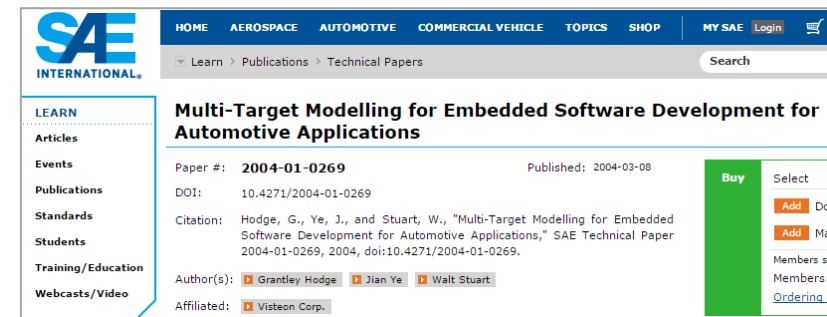
How about Efficiency of Generated Code?

Table 2 ROM and RAM comparison between a floating-point hand code and auto code.

	Hand Code	Auto Code
ROM	6408	6192
RAM	132	112

Table 1. Throughput Comparison between Model-Generated Code and Hand-Code

Task / Module	Throughput (uSec)	
	Model	Hand-Code
Current Magnitude and Phase Process (2.2)	1.42	1.31
ABC to QD0 Frame Transformation (2.3)	0.76	0.52
Resolver Harmonic Learn (1.12)	0.48	0.22
Angle Position Determination (2.1)	0.93	0.84
PI-Current Regulator (2.5)	7.62	7.51
Torque Mode (1.3)	4.82	4.72
DQ0 Rotating to Stationary Frame Transformation (2.7)	0.94	0.82
Complete 100 uSec Task	65.37	63.83



SAE INTERNATIONAL

HOME AEROSPACE AUTOMOTIVE COMMERCIAL VEHICLE TOPICS SHOP MY SAE Login

Learn > Publications > Technical Papers

Search

LEARN

Articles

Events

Publications

Standards

Students

Training/Education

Webcasts/Video

Multi-Target Modelling for Embedded Software Development for Automotive Applications

Paper #: **2004-01-0269** Published: 2004-03-08

DOI: 10.4271/2004-01-0269

Citation: Hodge, G., Ye, J., and Stuart, W., "Multi-Target Modelling for Embedded Software Development for Automotive Applications," SAE Technical Paper 2004-01-0269, 2004, doi:10.4271/2004-01-0269.

Author(s): [Grantley Hodge](#) [Jian Ye](#) [Walt Stuart](#)

Affiliated: [Visteon Corp.](#)

Buy Select

Add

Add

Members s

Members

Ordering



SAE INTERNATIONAL

HOME AEROSPACE AUTOMOTIVE COMMERCIAL VEHICLE TOPICS SHOP MY SAE Login

Learn > Publications > Technical Papers

Search

LEARN

Articles

Events

Publications

Standards

Students

Training/Education

Webcasts/Video

Application of Auto-Coding for Rapid and Efficient Motor Control Development

Paper #: **2014-01-0305** Published: 2014-04-01

DOI: 10.4271/2014-01-0305

Citation: Walters, J., Harianto, C., Kelly, E., and Sugiarto, T., "Application of Auto-Coding for Rapid and Efficient Motor Control Development," SAE Int. J. Passeng. Cars - Electron. Electr. Syst. 7(2):481-490, 2014, doi:10.4271/2014-01-0305.

Author(s): [James Walters](#) [Cahya Harianto](#) [Edward Kelly](#) [Tante Sugiarto](#)

Affiliated: [Delphi Automotive](#)

Buy Select

Add

Add

Members s

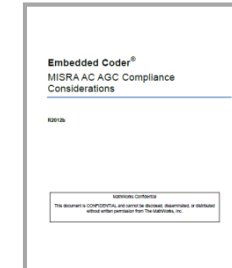
Members

Ordering

How about the Standard Compliance?

MISRA C

- Most rules satisfied
- You can download MISRA compliance document
<http://www.mathworks.com/matlabcentral/answers/102532-misra-c>



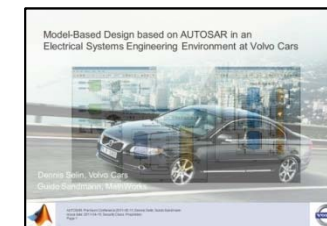
ISO 26262 / IEC 61508

- Tool Qualified by TÜV SÜD (ISO26262 ASIL A-D)



AUTOSAR

- You can download AUTOSAR Support Package via Simulink menu



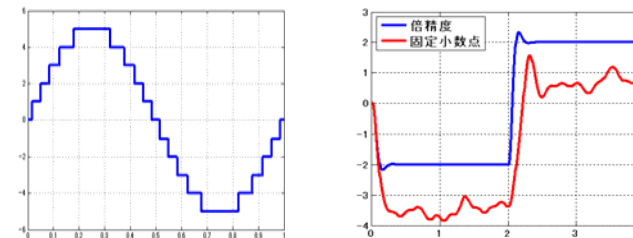
How to Design Fixed-Point Controller? Fixed-Point Designer

Fixed-Point Data Type Settings



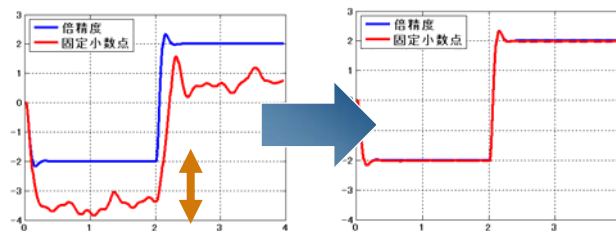
Conversion between Float and Fixed-Point

Useful for Quantization Error Test



Fixed-Point Simulation

Efficient Fixed-Point Scaling



Reduce Quantization Error

Fixed-Point Auto Scaling

Efficient Code Generation

```
int16_T k = 13U;
```

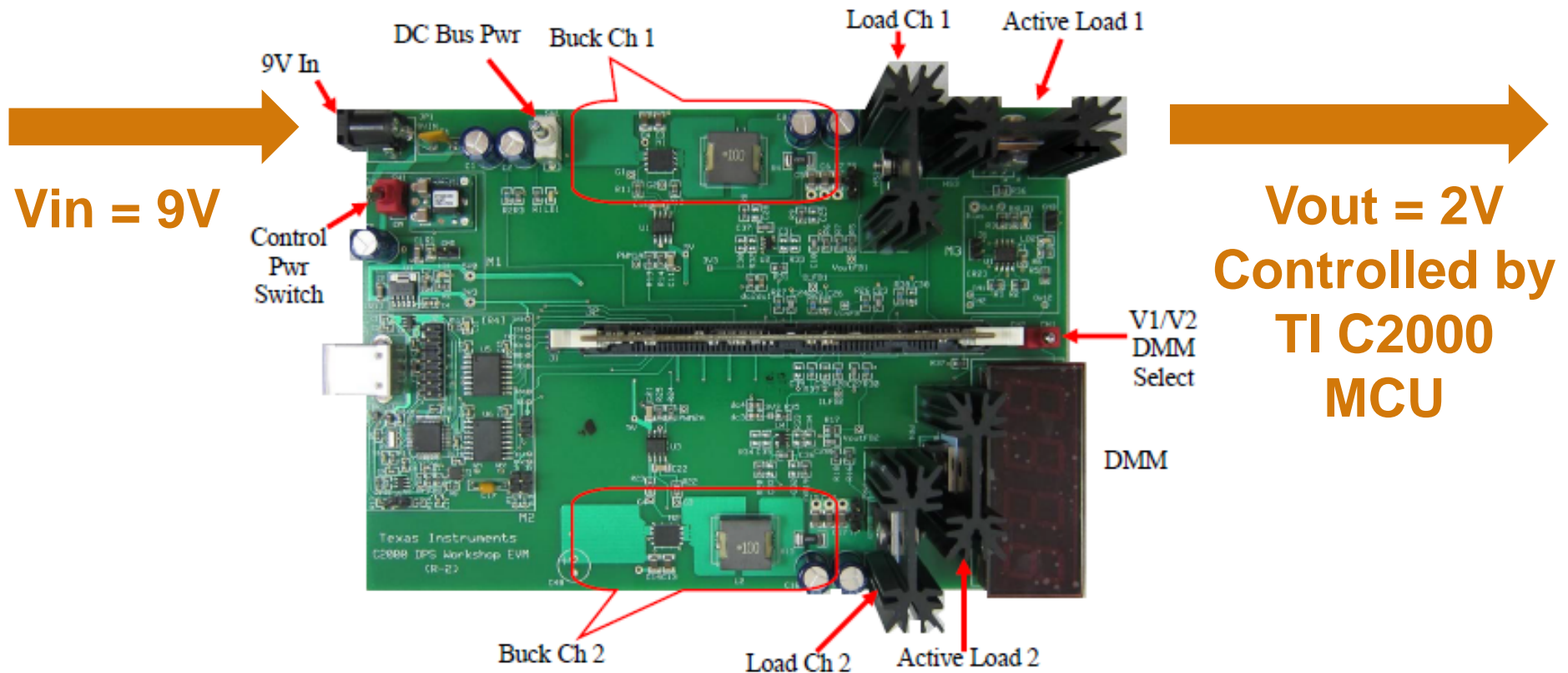
Fixed-Point Value

```
y = y + k * u >> 3;
```

Scaling Adjustment Calculation

Fixed-Point Code Generation

One More Example: Digital DC/DC Converter Control System



TI C2000 DPS Workshop Kit (CPU: F28035)

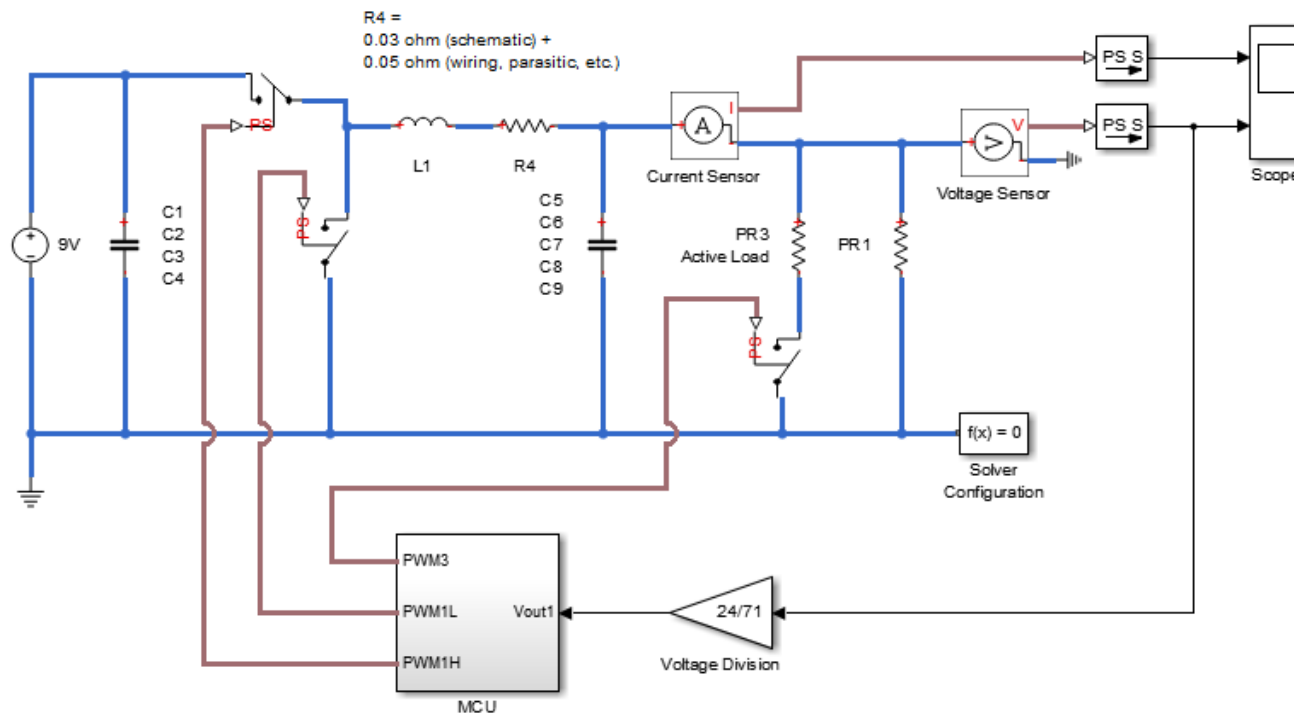
Simulation Model

Use for circuit/controller design, parameter tunings

- Making an analog circuit with Simscape components
- Fixed-point PI controller + Rate Limiter (Soft Start)

Digital DC/DC Synchronous Buck Converter Voltage Mode Control (VMC)

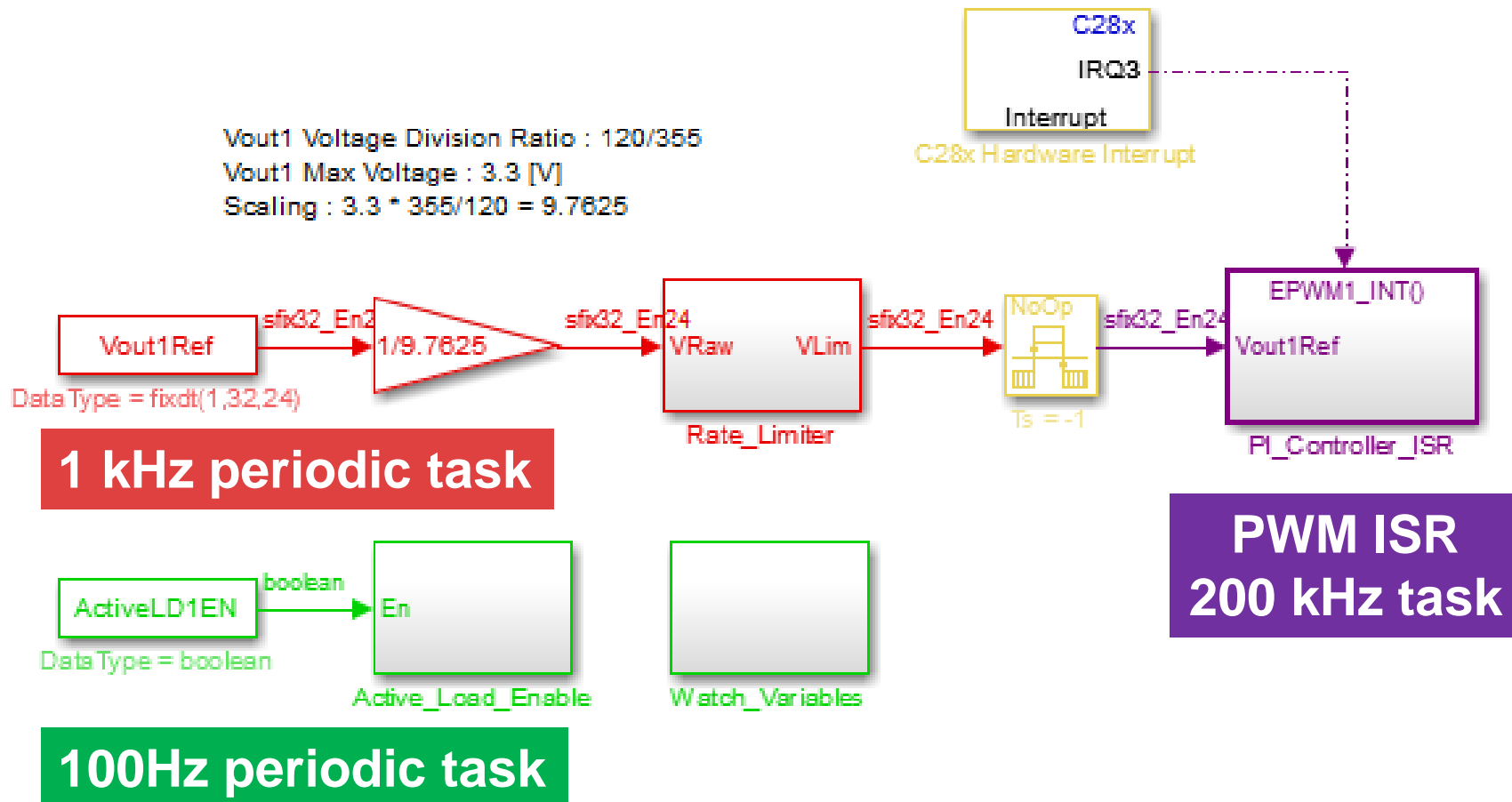
This example is a control system model to simulate TI C2000 DPS Workshop EVM (TMDSC2KWRKSHPKIT).



Code Generation Model to F28035

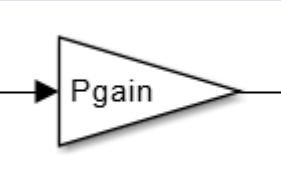
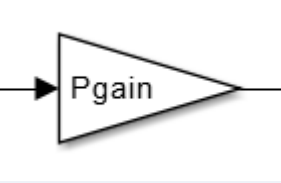
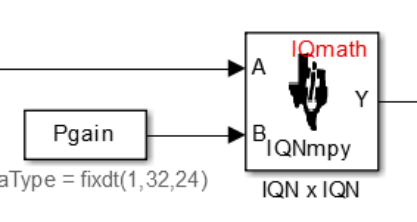
Turnkey to download the controller model into MCU

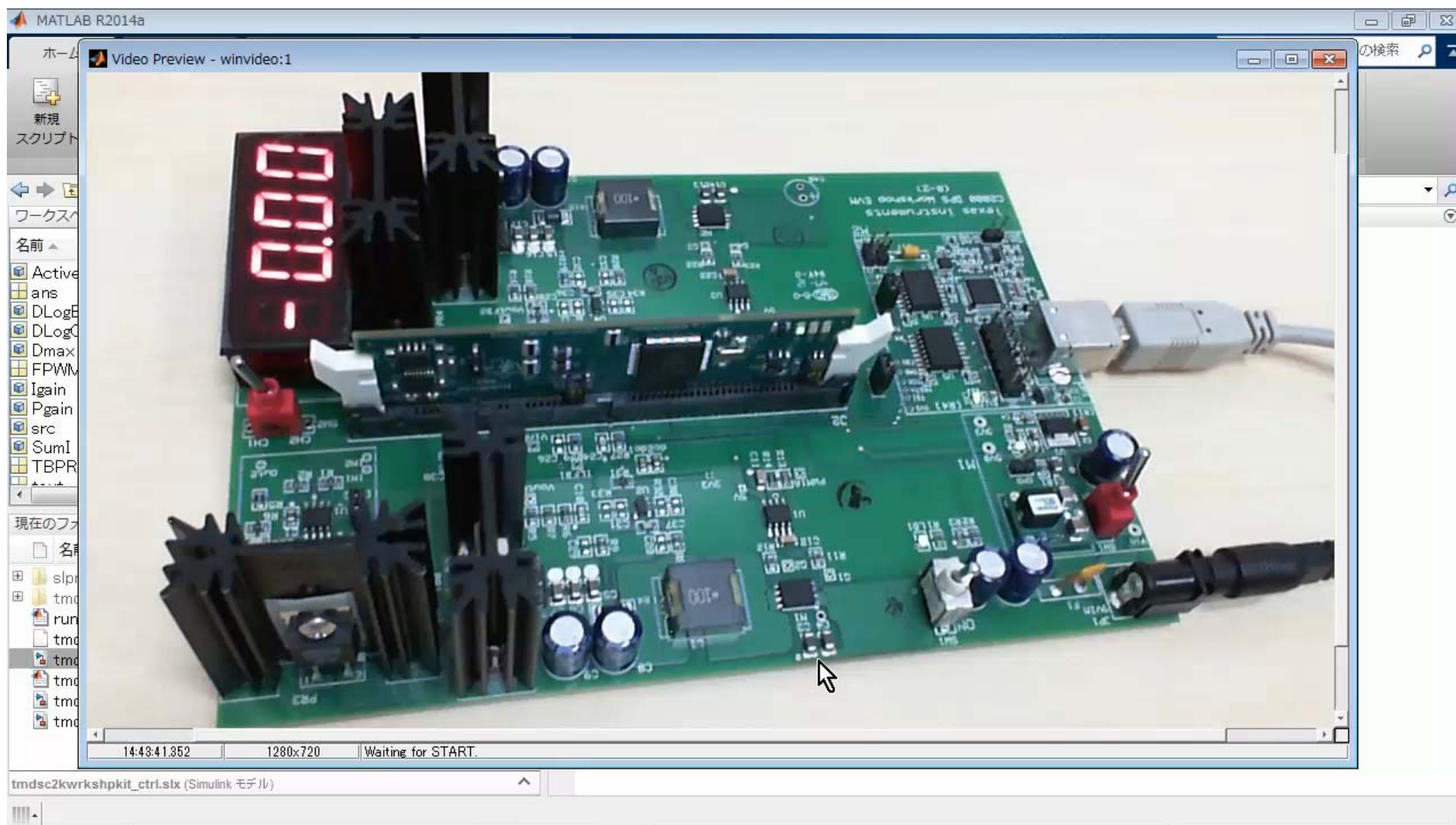
Using EC TI C2000 support package to implement I/O and ISR



CLR/IQmath Blocks and PIL Profiling

C2000 Support Package provides code optimization opportunity and execution time estimation

Model	Generated Code	PIL Execution
ANSI-C (No Optim) 	<code>rtb_Gain * Pgain</code>	6.15 us (cannot run at 200 kHz)
TI C28x CLR Only 	<code>c28x_mul_s32_s32_s32_sr(Pgain, rtb_Gain, 24L)</code>	3.23 us
TI C28x CLR + IQmath 	<code>_IQ24mpy(rt_b_Gain, Pgain)</code>	1.8 us



Summary

You can achieve highly efficient development cycle

