

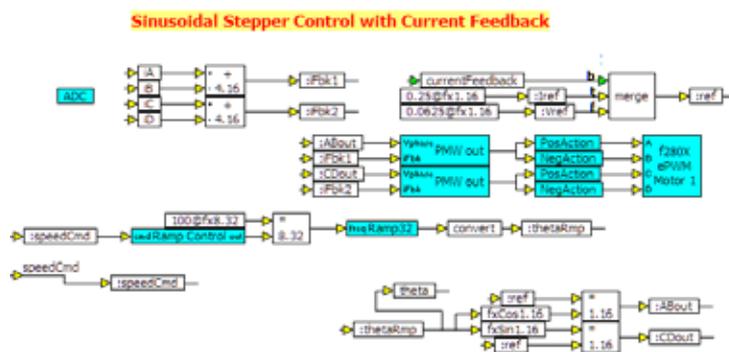
## Sinusoidal Stepper Motor Control with Current Feedback



TI Stepper Motor Control Kit

**Location:** Examples > Embedded > Digital Motor Control > Stepper  
**Name:** DRV8412EVM-stepper-28035

Embed is used to develop control systems for AC, BLDC, PMSM and stepper motors. This model generates C-code to control a Texas Instruments stepper motor control board. The diagram controls 1.8 deg 3A stepper motor. The board consists of a logic section with the F28035 Piccolo controlCARD (the socket will fit any of Texas Instruments C2000 controlCARD series), an on-board USB to XDS100 JTAG, and the [DRV8412 evaluation kit](#). This kit includes everything needed to spin two brushed DC or a single stepper motor out of the box: the DRV8412 motor driver (rated at 6A continuous / 12A peak at 50V w/98% efficiency), Piccolo F28035 controlCARD, 24V DC power supply, 2 DC brush motors and 1 stepper motor. In addition, there are high-precision op-amps tied to current sensors on the ground leg of each output phase, as well as the DC bus.



Embed source diagram to control TI Motor Control Kit

### Embed Diagram Controls the Hardware

The Altair Technical Services group has created the diagrams to drive the TI Stepper Motor Control Kit based on the F28035 Piccolo low-cost controlCARD. The controller modulates the 4 half H-bridges on the DRV8421 to drive the 4 phases of the stepper motor. Current sensors for each phase are used to provide a sinusoidal current drive. PID control is used to modulate the 4 PWM outputs to each half H-bridge. The diagram was assembled using standard Embed blocks such as configurable PWM blocks wired to pre-built subsystems like PIDs, ramp and ramp controllers, ADC inputs, GPIOs, and logic to control the operating mode of the inverter (open-loop ramp vs closed-loop sinusoidal). The closed-loop current mode gives more torque per pole commutation and allows for higher speed operation. Interactive inputs allow user control of speed and motor phase from the PC. Interactive vector outputs plot signals acquired on the target at the 10kHz on-chip sample rate and plot in digital scopes on the PC. This high-level interface allows rapid debugging at a high level. In this way, the designer can focus on the high-level motor control issues and not on problems due to C syntax, compiler flags, and missing libraries.

#### Simple Creation of Low-Speed Background Task

With a click on a subsystem, you can configure the subsystem to execute at any rate. You can also set it to run as a background task at that rate. The background task will run at the specified rate but is preempted by the main control task. Any blocks placed in the subsystem will run at the specified rate. Typically, supervisory commands on a CAN or UART bus are transmitted and received here. In this case, the motor control method selection logic (open-loop ramp speed vs closed-loop current mode) are placed in the subsystem, along with the DRV8412 reset logic. In the motor control diagram, the main control sample rate is 10kHz and one background task is set to run at 100Hz.

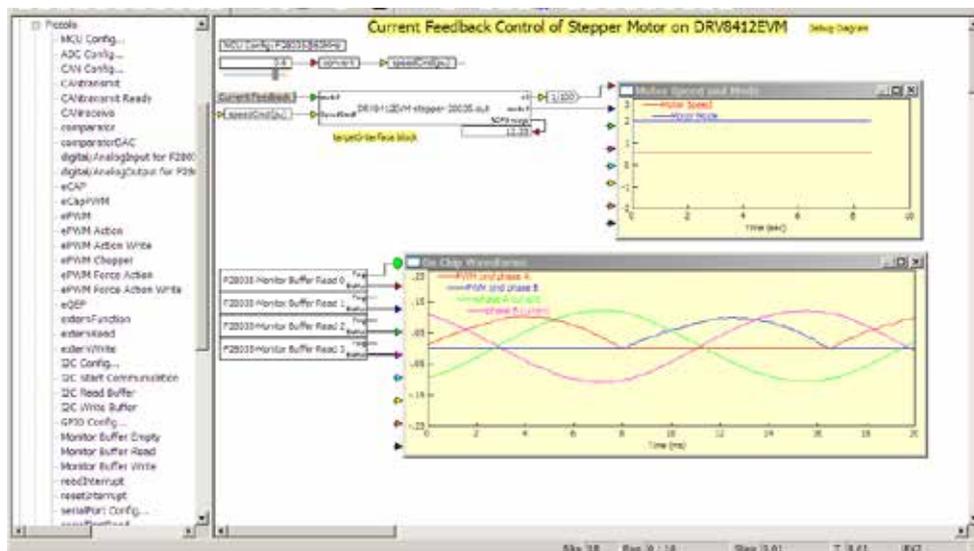
### From Diagram to Code with the Click of a Mouse

In Embed, the stepper motor control is constructed from standard fixed-point blocks coupled with blocks in the Digital Motor Control block set. This is seen in the diagram to the right. These blocks are placed inside a compound block with edge connectors for the values supplied from Embed running on the PC while the compiled control runs on the Piccolo target. In this case, a button input is supplied to select between three modes of operation: motor off, motor in open-loop mode, and motor in sensorless closed-velocity-loop mode. There are also input pins to vary the gain on the second harmonic space vector waveform generators feeding the PWMs and to vary settings of SMO gains.

To generate code and compile, click to the top level, select the encapsulated compound block, and choose **Tools > Codegen**. Clicking the **Compile** button generates the code and runs the linker to create an OUT file for the F28035 target.

### Interactive Download and Debug

Debugging takes place in the debug diagram. This diagram corresponds to the "source code" diagram from above; however, the compound block containing the motor control is substituted with a targetInterface block. The targetInterface block is initialized with the path to the .OUT file generated above and will automatically download the .OUT file and begin communication with the target on simulation start. This diagram is run in real time (using the System > System Properties command) at around 100Hz since the JTAG cannot support a communication rate much faster. By convention, the debug diagram shares the same name as the source code diagram, but has a "-d" suffix.



Embed Stepper Motor Control Debug Diagram

### Synchronize ADC with PWM Waveform

Another important consideration is to synchronize the ADC unit with the PWM waveform on the power device being measured. In this kit, the phase current sensors wired to ADCA0, ADCA1, ADCA2 and ADCA3. The F28035 Piccolo processor has an updated ADC architecture from the F280x series that allows each ADC channel to select any of the individual PWM events as a start of conversion signal. This is set up in the **Embedded > F280X > ADC Config** dialog. PWM units driving the DRV8402 are set to send a start-of-conversion signal (SOC) when TBCTR=PRD (period match). Since the PWM is set for up/down count, the measurement will be made in the center of the **off** portion of the PWM duty cycle. This works because the PWM is used to drive the high side of each phase leg, the current sensor is on the low side, and the DRV8412 automatically activates the low side when the PWM driving the high side is off. So we have set ADCA0 to trigger from PWM1-SOCA, and ADCA1 from PWM1-SOCB, ADCA2 from PWM2-SOCA, and ADCA3 from PWM2-SOCB.

### Waveform Capture for Debugging

For debugging, we are plotting PWM commands to phase A & B and phase currents  $i_a$  and  $i_b$  - is written to a 200-element buffer at the 10kHz control sample rate, giving a 50ms window. The buffer acquisition is triggered by a positive cross at .01 on the PWM A command. The waveform data is sent to Embed via the JTAG link and is updated in a digital scope in Embed Embedded on the PC at about 20Hz.

Authorized Distributors



**EMBEDDED SYSTEMS  
SOLUTIONS**  
www.embeddedindia.com

#5 606, World Trade Center, Bangalore  
altairsales@embeddedindia.com  
080-6764 8888/36, +91 98450 83528  
www.embeddedindia.com/contact.html