TI Designs
IoT Wi-Fi® Microstepping Stepper Motor Control

TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDM-TM4C123IOTSTEPPERMOTOR	TI Design Files
TM4C123GH6PM	Product Folder
DRV8833	Product Folder
CC3100	Product Folder
EK-TM4C123GXL	Tool Folder
DRV8833 EVM	Tool Folder
CC3100 BoosterPack	Tool Folder

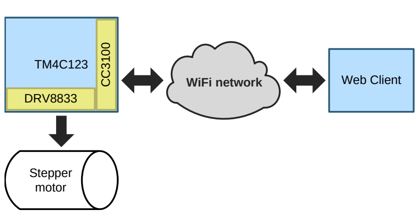
TI E2E™ Community — ASK Our E2E Experts

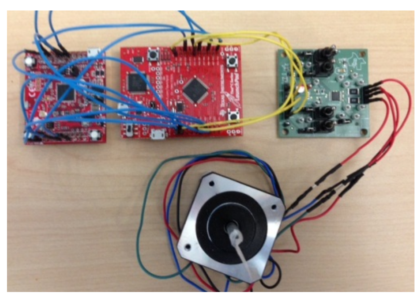
Design Features

- The TM4C123 Microcontroller Uses Four PWM Channels to Control DRV8833 to Rotate the Bipolar Stepper Motor in Full Step and Microstep Modes.
- The CC3100 BoosterPack™ Hosts a Webserver Interface to Control the TM4C123 LaunchPad™ and Stepper Motor Remotely From a Wi-Fi-Enabled Device Such as a PC or Mobile Device.
- The Software is Designed to Work With an EK-TM4C123GXL LaunchPad, CC3100 BoosterPack, and DRV8833 EVM.
- The HTML/MCU Code Also Lets the User Remotely Control the Operation of EK-TM4C123GXL LaunchPad, Including LED Toggling, Internal Temperature Reading, and Button Press Recording Through a Web Browser.
- This TI Design Supports Control of the Stepper Motor Using a UART Interface or a Webserver Over Wi-Fi®.

Featured Applications

- Industrial Application and Automation
- Speed Control Applications
- Precision Motion Control
- Textile Equipment and Medical Analyzers
- Consumer Applications Such as Metering Pumps, Printers, Antenna Positioning, and Security Cameras





1 System Description

This system shows how to control a bipolar stepper motor in full step and microstep modes with the TM4C123 high-performance microcontroller and DRV8833 motor driver. The CC3100 BoosterPack is integrated with TM4C123 LaunchPad to implement a Wi-Fi webserver interface to control the TM4C123 and stepper motor through a web browser. The functionalities for this project can be extended in real applications requiring the high-performance features of the TM4C device and connection to a wireless domain. The software accompanying this design is developed and tested on an EK-TM4C123GXL LaunchPad, DRV8833 EVM, and CC3100 BoosterPack.

1.1 TM4C123GH6PM

The TM4C123GH6PM microcontroller is targeted for industrial applications including the following:

- Remote monitoring
- Electronic point-of-sale machines
- Test equipment
- Measurement equipment
- Network appliances
- Switches
- Factory automation
- HVAC
- Building control
- Gaming equipment
- Motion control
- Transportation
- Security

The TM4C123GH6PM is an 80-MHz high-performance microcontroller with up to 256KB of on-chip flash and 32KB of on-chip SRAM. There are up to 43 GPIOs with programmable control for GPIO interrupts, pad configuration, and pin muxing. The MCU is integrated with the following:

- Six 32-bit general-purpose timers (up to twelve 16-bit)
- Eight UARTs
- Four synchronous serial interface (SSI) modules
- Four inter-integrated circuit (I2C) modules
- Two 12-bit analog-to-digital converters (ADCs) with 12 analog input channels and a sample rate of one million samples per second
- Eight pulse width modulation (PWM) generator blocks
- Two quadrature encoder interface (QEI) modules

The on-chip universal serial bus (USB) controller supports the USB OTG mode, host mode, and device mode. The ARM® PrimeCell 32-channel configurable µDMA controller is also integrated to provide a method to offload data transfer tasks from the ARM Cortex®-M4 processor and to efficiently use the processor and the bus bandwidth. See Figure 1 for more information.

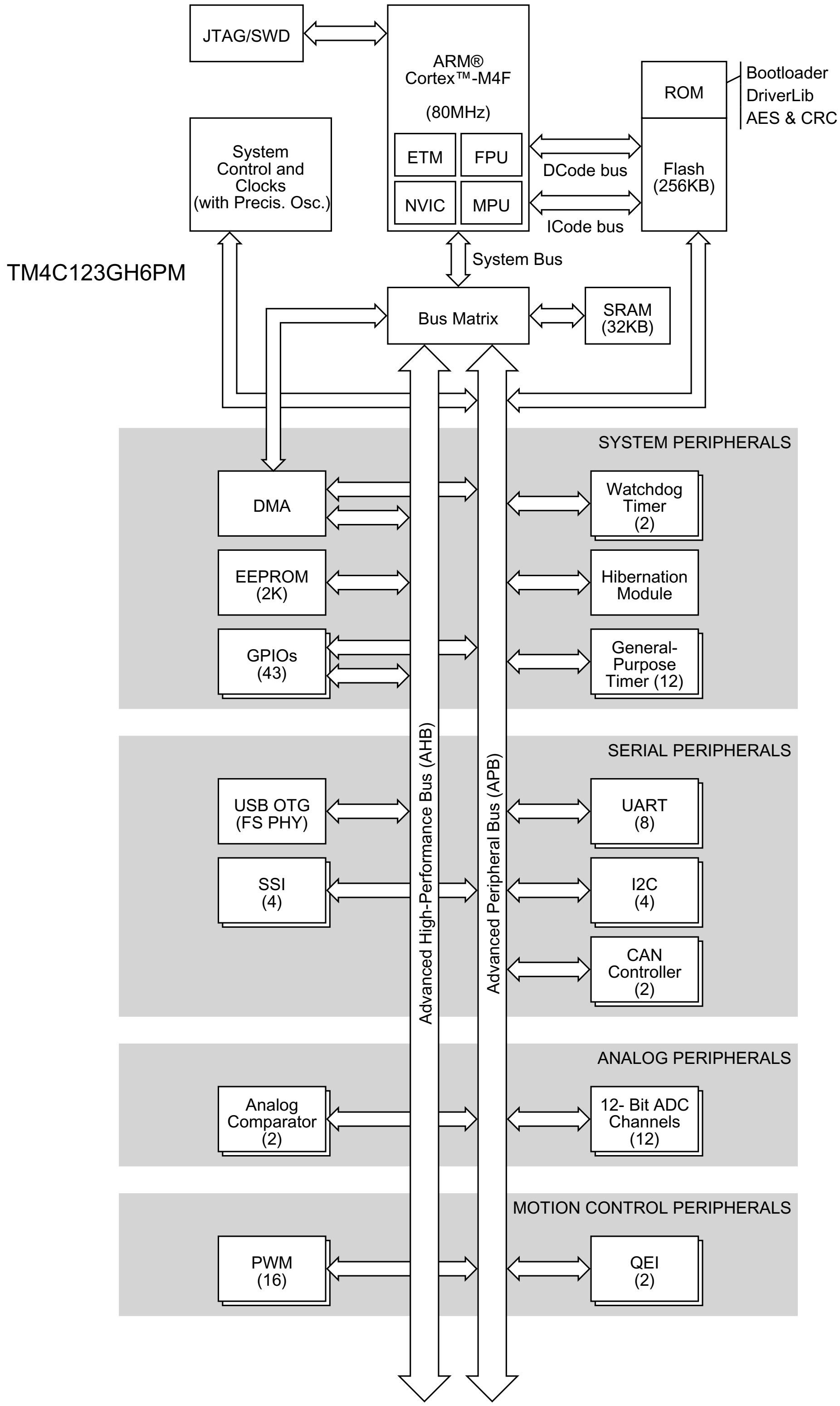


Figure 1. TM4C123GH6PM Microcontroller High-Level Block Diagram

1.2 DRV8833

The DRV8833 device has two H-bridge drivers to drive a bipolar stepper motor, two DC-brush motors, or other inductive loads. Aimed at driving 3.3-V and 5-V motors, this stepper driver with integrated FETs support up to 1.5 A (rms) with a low-power sleep mode to conserve power for battery-powered applications. Internal shutdown functions with a fault output pin are provided for overcurrent protection, short-circuit protection, undervoltage lockout, and overtemperature. See Figure 2 for more information.

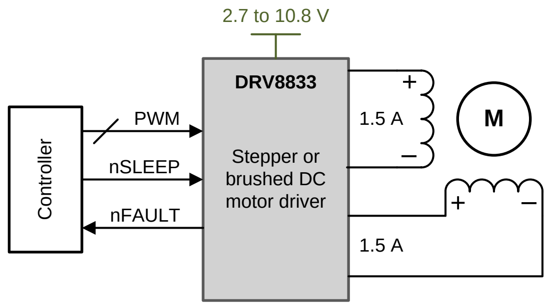


Figure 2. DRV833 Functional Block Diagram

1.3 CC3100

The CC3100 Wi-Fi network processor subsystem features a Wi-Fi Internet-on-a-chip™ integrated circuit and contains an additional dedicated ARM MCU that completely offloads the host MCU. This subsystem includes an 802.11 b/g/n radio, baseband, and MAC with a powerful crypto engine for fast, secure internet connections with 256-bit encryption. The CC3100 device supports station, access point, and Wi-Fi-direct modes. The device also supports WPA2 personal and enterprise security and WPS 2.0. This subsystem includes embedded TCP/IP and TLS/SSL stacks, HTTP server, and multiple internet protocols. See Figure 3 for more information.

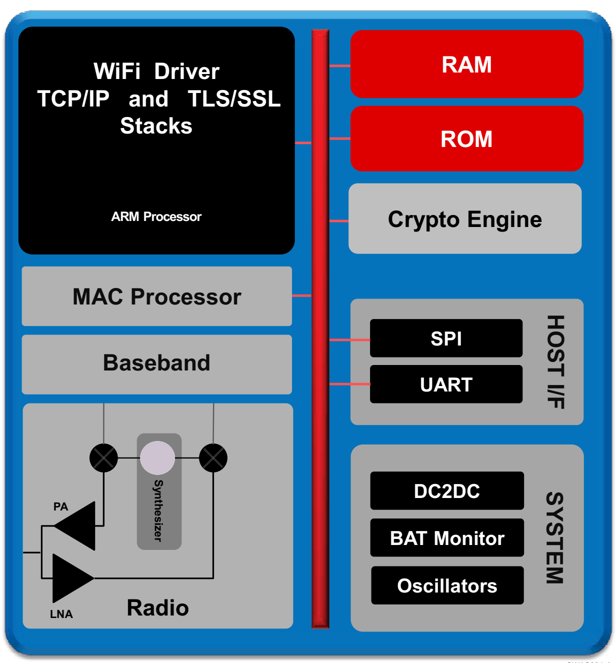


Figure 3. CC3100 Hardware Overview

1.4 Stepper Motor Control

A stepper motor is a brushless DC-electric motor that divides a full rotation into a number of equal steps. The position of the motor can be commanded to move and hold at one of these steps without feedback. The stepper motor is used in a wide range of applications involving precision motion control. Figure 4 shows the four PWM signals required to drive a bipolar stepper motor in full step mode with DRV8833.

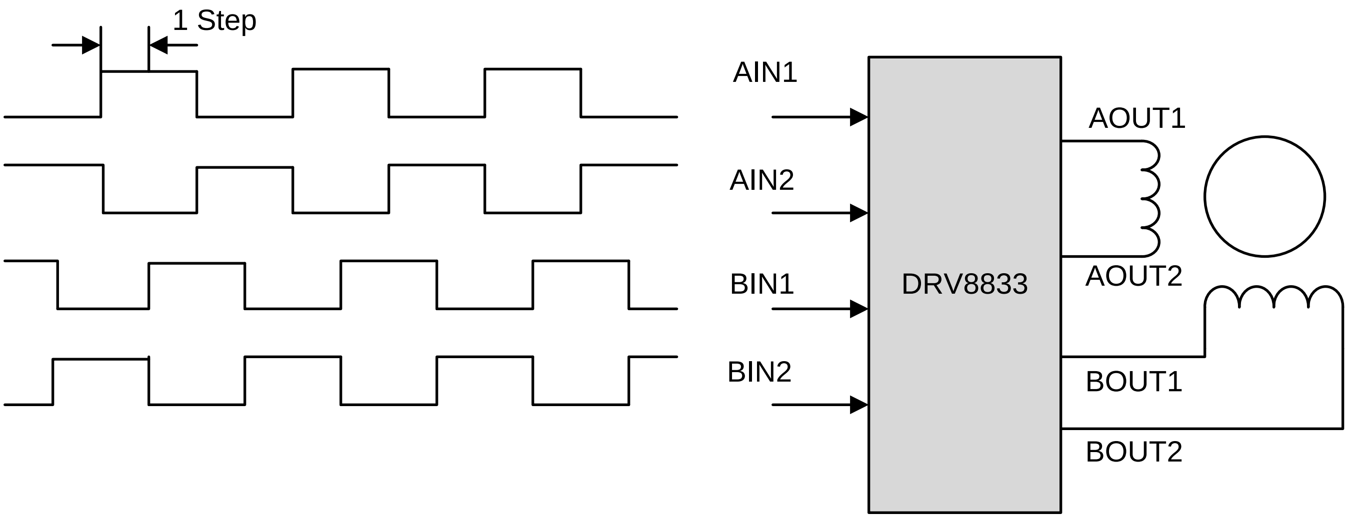


Figure 4. Driving the Stepper Motor in Full Step Mode

In full step operation, the stepper motor moves through its basic step angle. For a 1.8-degree step, the motor takes 200 steps per complete revolution. In the full step mode, the current is fully on during the driving phase. This mode provides the best torque and speed performance from the motor. Figure 5 shows the PWM signals driving DRV8833 in full step mode. In Figure 5, the signals are arranged from top to bottom as AIN1, AIN2, BIN2, and BIN1.

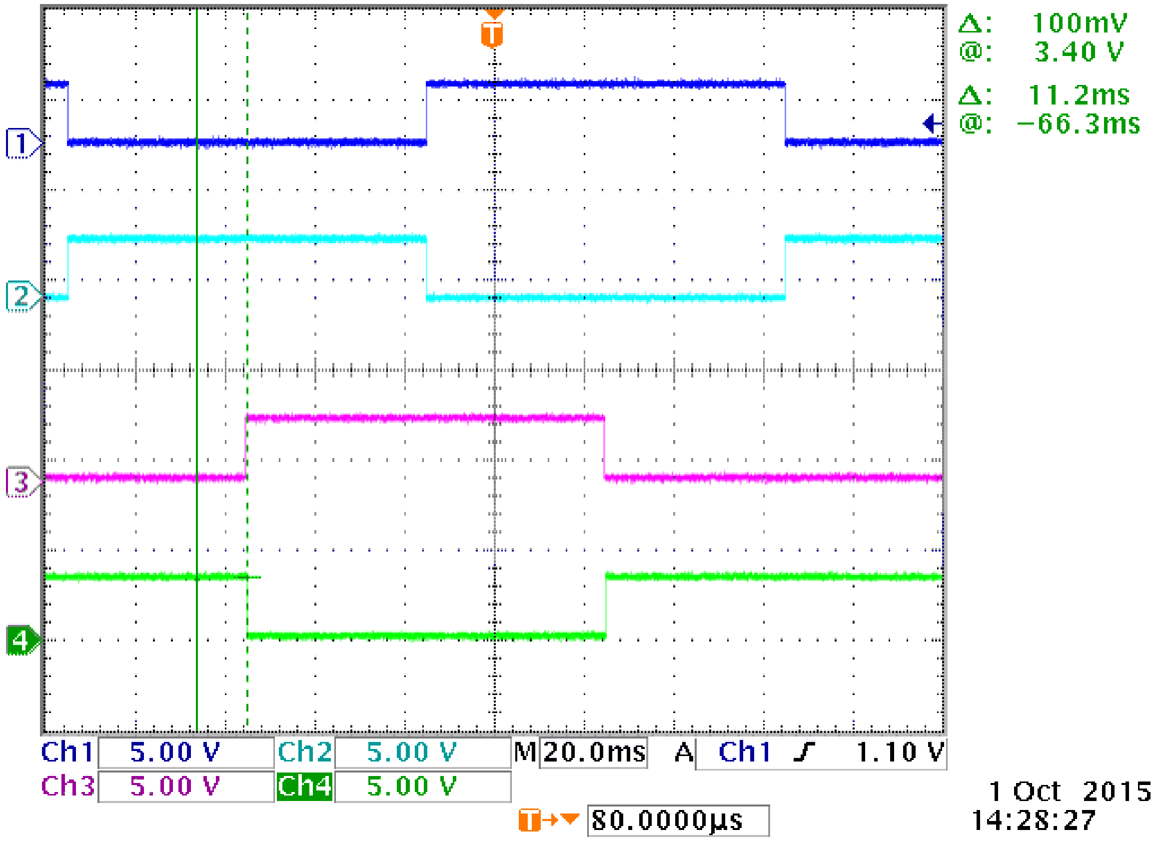


Figure 5. Driving Waveform to DRV8833: Full Step

Figure 6 shows the currents going through the motor windings in full step mode and two driving signals to DRV8833 for reference. In Figure 6, channel 1 is AIN1 and channel 2 is BIN1. There are two motor windings (A and B). Channel 3 is the current going through motor winding A and Channel 4 is the current going through motor winding B. The current flowing through the winding is FULL ON in one direction and FULL ON in the opposite direction.

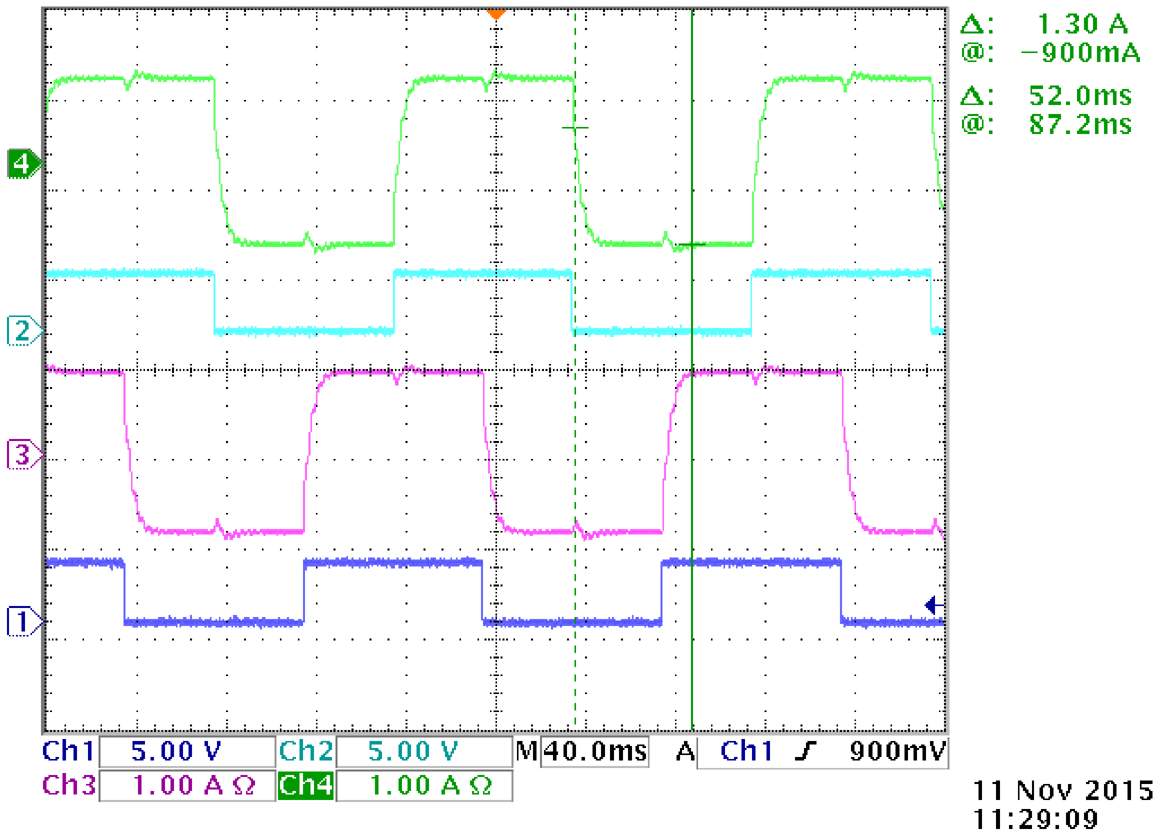


Figure 6. Motor Winding Currents: Full Step Mode

Although full step mode provides the best toque, the user might want to eliminate the mechanical vibration at low speeds. The microstepping technique eliminates the jerky and noisy character of low-speed stepping motor operation and reduces problems with resonance. Microsteps make the current flowing through the motor winding increase or decrease gradually. Microstepping allows for incremental pushing or pulling of the rotor rather than full force. In this way, the motor can move smoothly.

The most common driving waveform in microstepping is the sine wave. The microstepping technique approximates a sine wave by dividing a full step into multiple microsteps and regulating the current in each microstep. In this example, DRV8833 device can be only in an on or off state. To regulate current, one PWM pulse is generated for each microstep. The duty cycle of the PWM is dynamically modified to modulate the current in each microstep so that the integrated waveform is approximately a sine wave. Figure 7 shows the concept of dividing one full step into four microsteps.

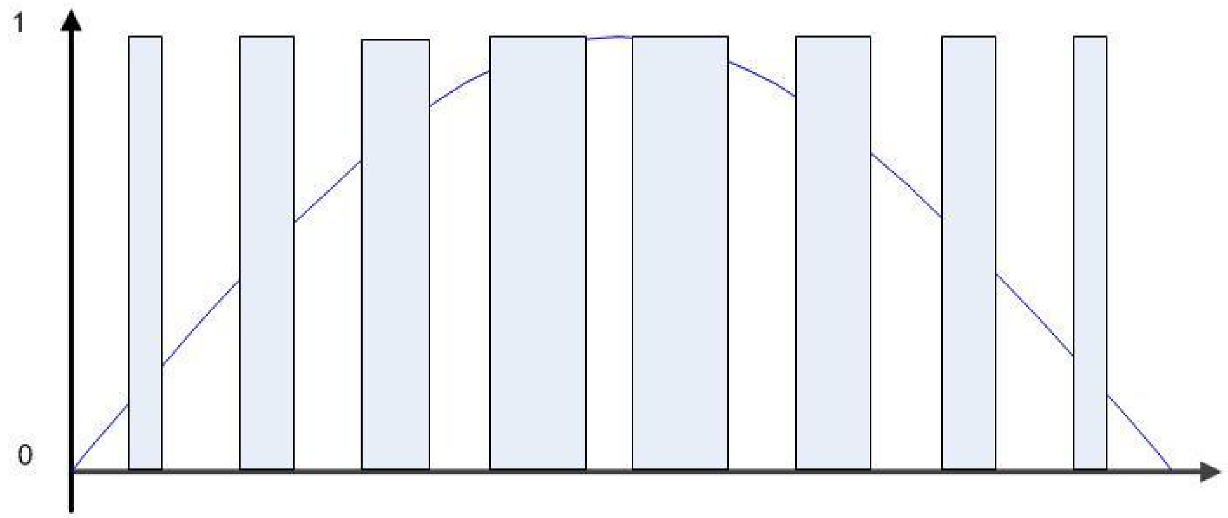


Figure 7. Waveform For Four Microsteps Per Full Step

As the microsteps become smaller (a full step is more divided), motor operation becomes smoother. The power of microcontroller computing and motor driver switching speed are the primary factors limiting the number of microsteps in an application. CPU needs to be interrupted at each microstep to change the duty cycle of the next PWM pulse to regulate current. At the same motor rotation speed, driving a motor with 256 microsteps requires 256 times more CPU power compared to the full step. The motor driver switching speed limits the minimum pulse width of the driving signal. In addition to the limitation of the controller processing speed and the motor driver switching speed, the mechanical friction and backlash limit the resolution. This system demonstrates how to generate the four channels of dynamically modulated PWM signals with the PWM module on TM4C123. Because the DRV8833 EVM does not support current sensing, only open-loop current regulation is used.

Figure 8 shows the PWM signals driving DRV8833 with 256 microsteps. The signals are arranged from top to bottom as AIN1, AIN2, BIN2, and BIN1.

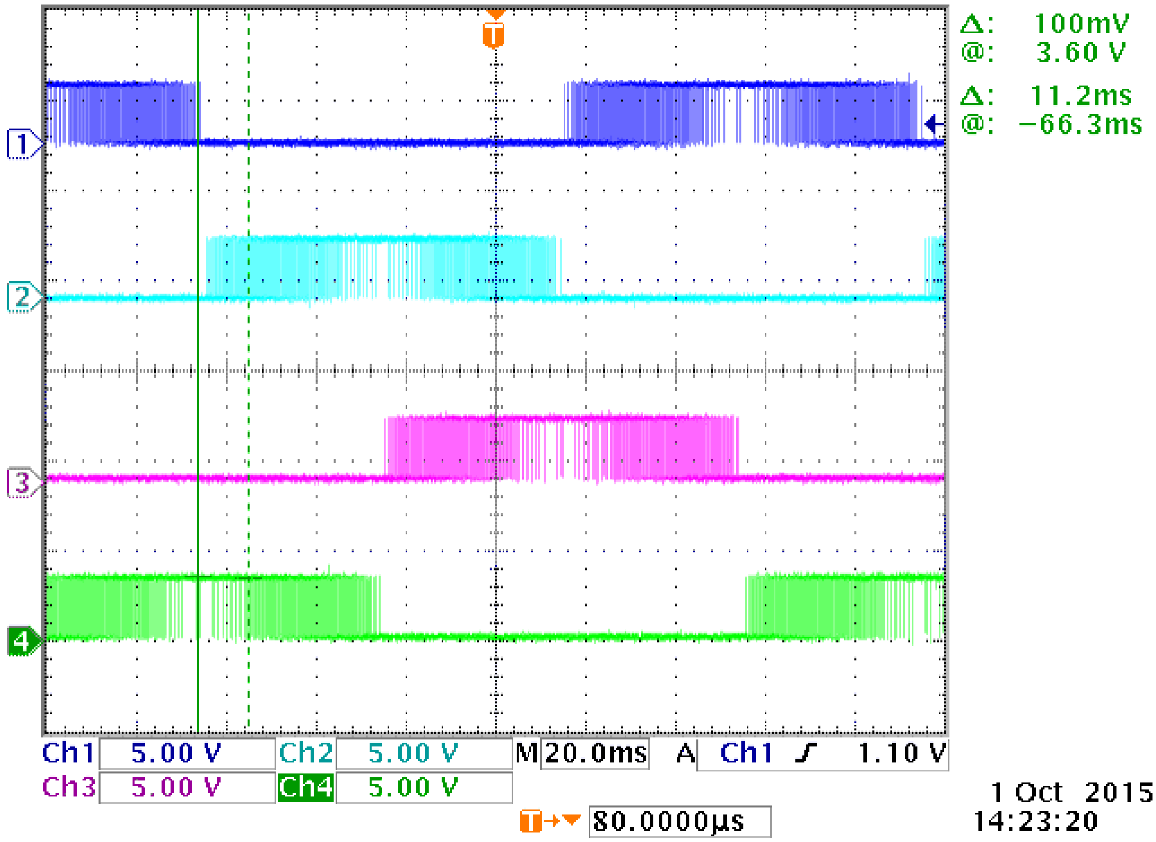


Figure 8. Driving Waveform to DRV8833: 256 Microsteps

Figure 9 shows the currents going through the motor windings with 256 microsteps and the two driving signals to DRV8833. In Figure 9, channel 1 is AIN1 and channel 2 is BIN1. There are two motor windings (A and B). Channel 3 is the current going through motor winding A and Channel 4 is the current going through motor winding B.

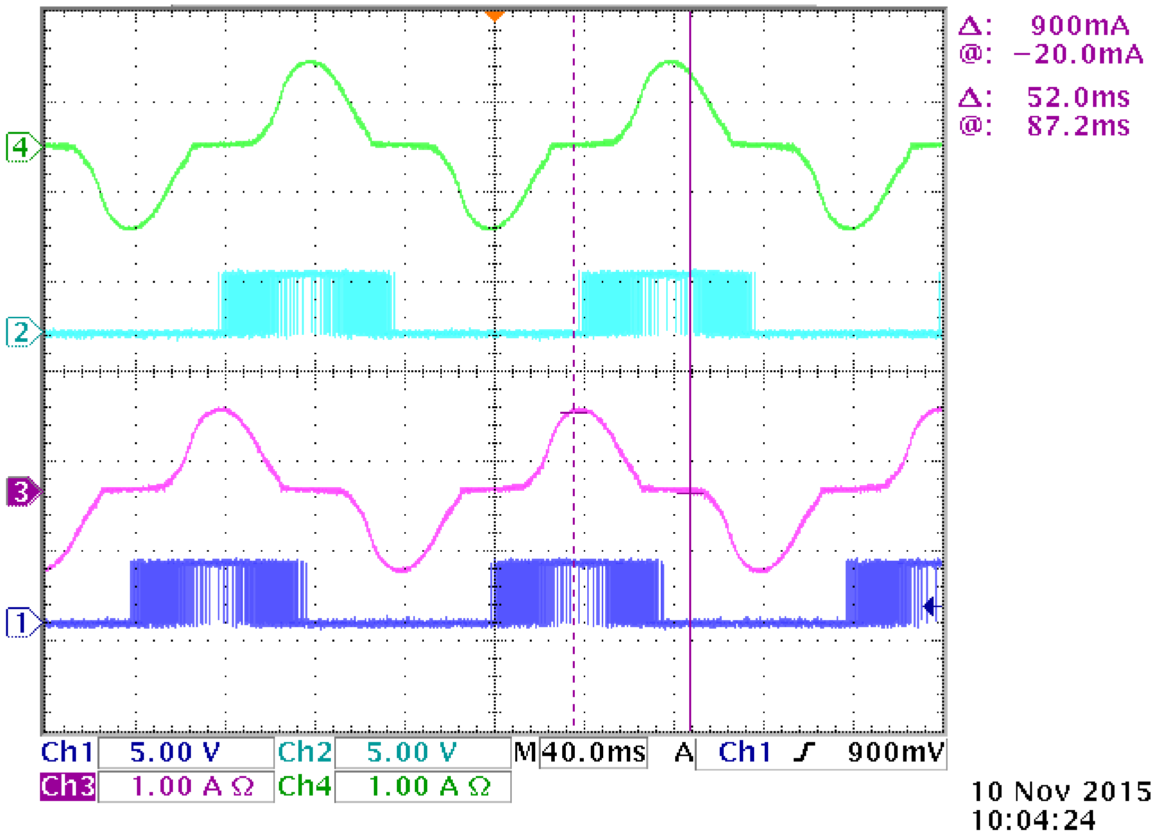


Figure 9. Motor Winding Currents: 256 Microsteps

2 Getting Started–Hardware

Figure 10 shows the connection signals of TM4C123, DRV8833, and CC3100. TM4C123 generates four channels of PWM signals to drive the stepper motor through the DRV8833 device. The period and duty cycle of the PWM signals are changed dynamically to drive the motor with the required mode (that is, full step and microstep) and speed.

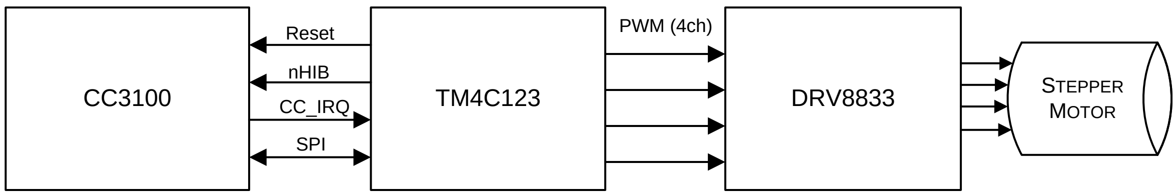


Figure 10. TM4C123, DRV8833, and CC3100 Integration

Two motor control user interfaces are provided in this system:

- The Wi-Fi function of CC3100
- A TM4C123 UART port

In addition to the standard SPI interface, there are three control and handshaking signals between TM4C123 and CC3100. The TM4C123 device uses the reset signal to reset CC3100. The TM4C123 uses the nHIB signal to enable and disable CC3100. The CC3100 uses the CC_IRQ signal to notify TM4C123 when it requires service. The SPI interface is used for the data communication between TM4C123 and CC3100. The TI SimpleLink™ library driver processes the message and controls the communication. The application running on TM4C123 can process the requests from CC3100 using http callback events without knowing the SPI messaging format.

The hardware in this example consist of the EK-TM4C123GXL LaunchPad, the DRV8833 motor driver EVM, and CC3100 BoosterPack. The EK-TM4C123GXL LaunchPad board connects to the DRV8833 board through the connectors on the EK-TM4C123GXL using jumper wires. Table 1 shows the signal mapping.

Table 1. TM4C123XL and DRV833 EVM Interface Signals

EK-TM4C123GXL LaunchPad	DRV8833 EVM
PB7 (M0PWM1)	AIN1
PB5 (M0PWM3)	AIN2
PE5 (M0PWM5)	BIN2
PC5 (M0PWM7)	BIN1
GND	GND

To use the PWM signals on the EK-TM4C123GXL LaunchPad connectors, the CC3100 BoosterPack cannot be plugged onto the EK-TM4C123GXL LaunchPad. Jumper wires connect the EKTM4C123GXL LaunchPad and the CC3100 BoosterPack. To more easily connect, make an interface board. For more information, see Table 2.

Table 2. TM4C123 and CC3100 Interface Signals

EK-TM4C123GXL LaunchPad	CC3100 BoosterPack
PE0	CC_SPI_CS
PA2 (SSI0_CLK)	CC_SPI_CLK
PA5 (SSI0_TX)	CC_SPI_DIN
PA4 (SSI0_RX)	CC_SPI_DOUT
PE1	CC_nHIB
PD3	CC_IRQ
RESET	MCU_RESET
3.3 V	3.3 V

Table 2. TM4C123 and CC3100 Interface Signals (continued)

EK-TM4C123GXL LaunchPad	CC3100 BoosterPack
5 V	5 V
GND	GND

The following TM4C123 peripherals are enabled in this system.

- All four PWM generators in the PWM0 modules are enabled to generate PWM signals on M0PWM1, M0PWM3, M0PWM5, and M0PWM7 pins to drive the motor. The load interrupt on the PWM generator 0 is enabled to dynamically load PWM parameters for all four PWM generators.
- SSI0 module for the communication with CC3100. PE0, PE1, and PD3 for handshaking.
- UART 0 as a separate interface for motor control
- Timer 2 interrupt to drive LED on EK-TM4C123GXL LaunchPad with GPIO pin PF1
- Timer 0 interrupt to sample button states on EK-TM4C123GXL LaunchPad from GPIO input pins PF0 and PF4, and the on-chip temperature sensor
- ADC0 for reading the on-chip temperature sensor.
- GPIO pin PC7 is configured as an output for measurement of critical processing time

3 Getting Started–Software

Figure 11 shows the architecture of the TM4C123 software. The TivaWare™ library configures the hardware on TM4C123. The load interrupt of the PWM generator 0 is enabled to occur at the interval of the PWM period. In the interrupt service routine, the load and compare A registers are updated for all four PWM generators to configure the period and duty cycle for the next PWM cycle. The updated value is loaded to PWM module when the PWM counters reach zero so that the four PWM generators are updated synchronously. The Internet data communication is accomplished by http callback function in the TI SimpleLink™ library through the http post and get events. A UART interface is also created to control the motor using the TivaWare utility library and the USB debug port on the EK-TM4C123GXL LaunchPad.

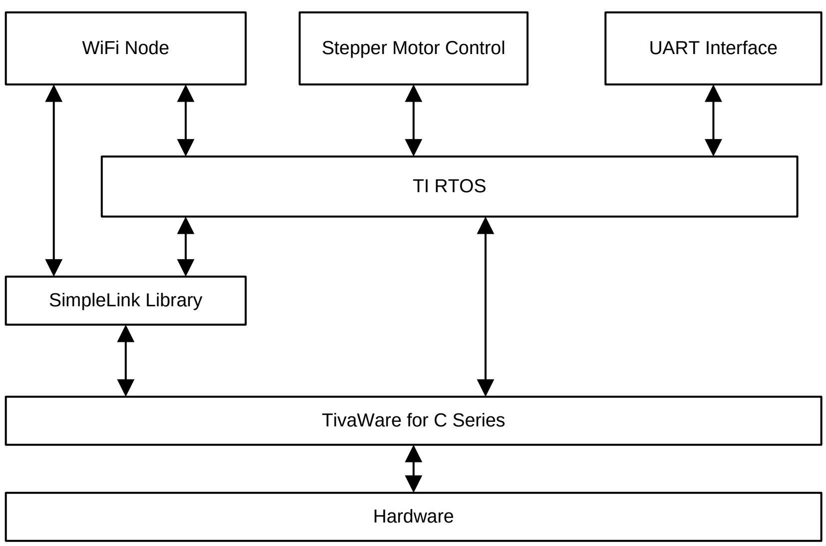


Figure 11. TM4C123 Software Architecture

TI RTOS schedules the processing. The following TI RTOS functions are statically configured in the TI RTOS configuration file.

- CLK0: 1-ms system tick generated by timer 3
- HWI_PWM0: PWM0 load interrupt to update the PWM duty cycle for each microstep
- HWI_timer0a: 10-ms timer 0 interrupt for sampling buttons and internal temperature sensor
- HWI_timer2a: Timer 2 interrupt with variable period for toggling LED

- HWI_UART0: UART0 interrupt for sending and receiving 1 byte on UART
- TI RTOS SPI and GPIO drivers handle the communication and handshaking between TM4C123 and CC3100
- Task_HttpServer: Process HTTP events using the SimpleLink library call back function
- Task_UART0: Process UART command for the stepper motor control

To minimize the CPU overhead in dynamically modifying the PWM duty cycle for approximating a sine wave, TI adopted a lookup table approach. The lookup table in this system has 512 integer entries created by the formula in Equation 1 to support a maximum resolution of 256 microsteps.

$$\text{Table}(i) = 32767 \times \sin\left(i \times \frac{\pi}{512}\right), \quad i = 0, \ldots 511 \tag{1}$$

At each microstep, the pointer to the lookup table is incremented. Table 3 lists that the increment size according to the resolution.

Table 3. Lookup Table Pointer Increment Size

Microstepping Resolution	Lookup Table Pointer Increment Size
256	1
128	2
64	4
32	8
16	16
8	32
4	64
2	128
1	256

The PWM counters are 16-bit down-counters. The period of the PWM is determined by the motor rotation speed and the microstepping resolution. For a given rotation speed and microstepping resolution, the duty cycle of the PWM is dynamically changed to approximate a sine wave for driving the motor. The duty cycle is determined by the value in the PWM compare A register and calculated as follows.

PWM Compare Register Value = (PWM period × Lookup Table Value) ÷ 32768

The PWM signal goes high when the PWM counter value matches the PWM compare register and returns to low when the PWM counter reaches zero. Tthe PWM period and compare register are then reloaded automatically. The load interrupt for PWM generator 0 is triggered for the CPU to update the PWM period and compare register for the next PWM cycle. See Figure 12 for more details.

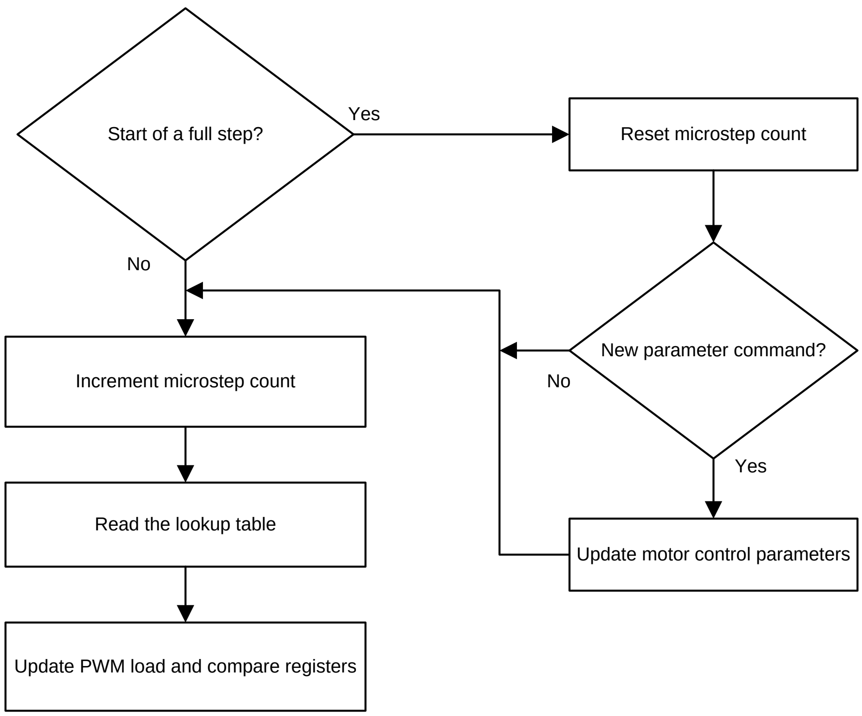


Figure 12. Processing in PWM Interrupt Service Routine

Figure 12 shows the processing in the PWM ISR. The command from the Wi-Fi and UART interface are processed at the beginning of the full step. The microstepping resolution can be changed only when the motor is stopped. The motor control interface supports the following commands.

- Configure microstepping resolution
- Configure motor rotation direction and speed
- Stop the motor
- Rotate the motor continuously
- Rotate the motor for predetermined full steps and stop
- Rotate the motor for predetermined full steps and reverse
- Display the motor control status

The command from the Wi-Fi interface is processed by the http callback function from the TI SimpleLink library. From the stepper motor control demo web page, you can send a post request to command the motor and a get request to acquire the motor control status. The text strings from those requests are processed in the http callback function and converted to a numerical value when necessary. When an http request is processed, a command flag is set to notify the motor control so that the control parameter can be updated at the beginning of the next full step. See Figure 13 from more information.

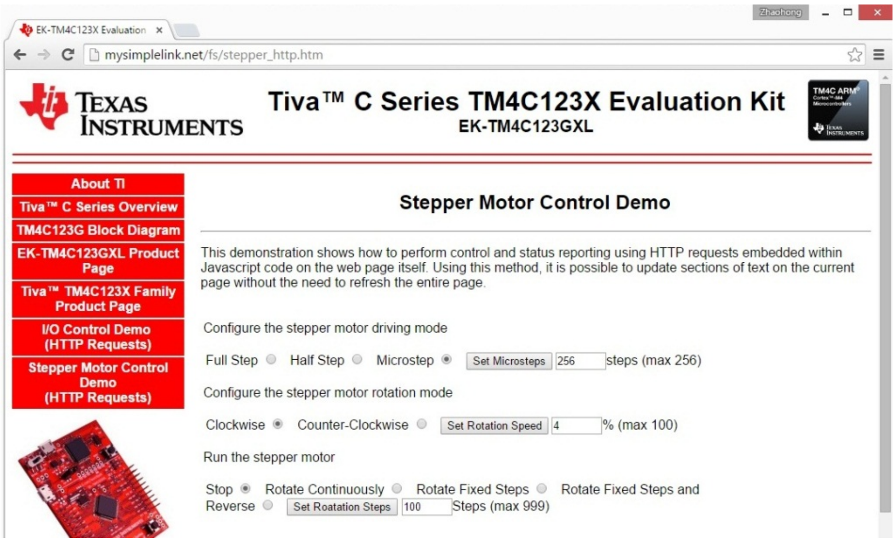


Figure 13. Stepper Motor Control Demo Web Page

The UART interface uses the UART0 module with the TivaWare utility library. The UART port is configured to a baud rate of 115200 with an 8-N-1 format. If the web interface and an UART interface are used in the same time, the web page must be manually refreshed to be synchronized with the UART command. Table 4 lists the command line input.

Table 4. Command Line

Name	Description
mode N	Configure the resolution to N (maximum 256) microsteps.
dir	Change the rotation direction.
speed N	Set the rotation speed to N (maximum 100) %.
stop	Stop the motor.
run	Rotate the motor continuously.
rfs N	Rotate the motor for N full steps and stop.
rfr N	Rotate the motor for N full steps and reverse.
help	Display available commands.
status	Display current motor control status.

The released MCU software and CC3100 html code also supports the I/O control demo described in detail in the TI Design (TIDM-TM4C129XWIFI).

4 Installing the Demonstration

To rebuild the demonstration, you must acquire the following TI development tools and software packages.

- EK-TM4C123GXL LaunchPad
- DRV8833 EVM
- CC3100 BoosterPack
- CC3100 EMU boost board (for programming of CC3100)
- Code Composer Studio™ (CCS) v6.0.1 or more
- TivaWare v2.1.0.12573 or above
- TI RTOS V2.14.0.10 or above
- CC3100 SDK 1.0.0 and above
- CCS Uniflash for CC3100/CC3200

The CC3100 html code must be programmed to CC3100 following the description in the TI DesignTIDM-TM4C129XWIFI.

Save the softwareanywhere in the PC from the zip file. To recompile the project correctly, configure the location of CCS, TivaWare, and CC3100 SDK to match their installation. The paths in Figure 14 are obtained with default locations for software installation. If you do not install the software to the default location, the linked paths must be modified to match the actual installation.

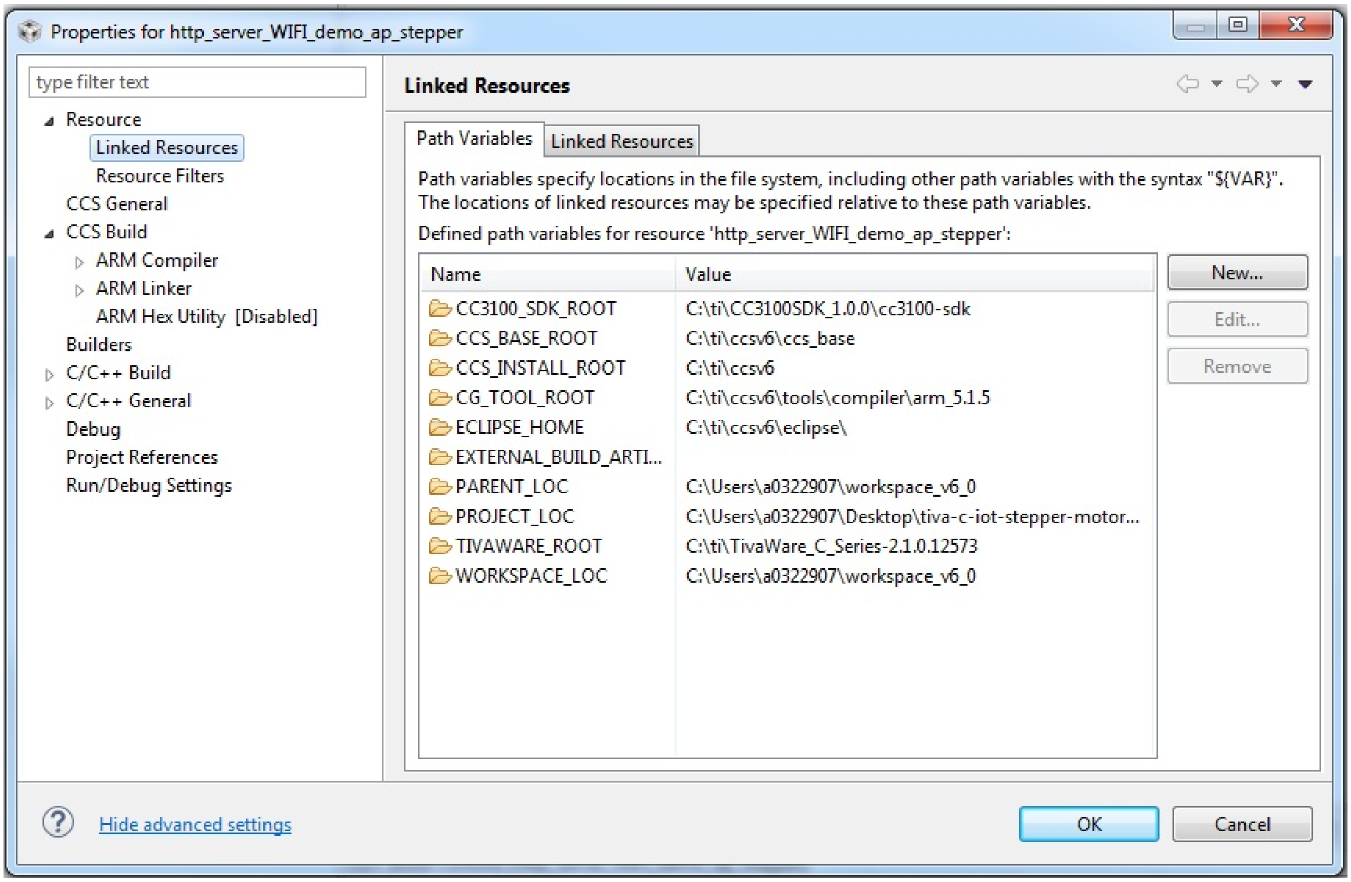


Figure 14. Resource Path for the CCS Project

5 Executing the Demonstration

In the software, the CC3100 is configured as a Wi-Fi access point. Connect to the Wi-Fi network hosted by CC3100 as follows.

1. Enter *cc3100* as the name of the network without a password.
2. Search for *mysimpleLink.net* from a web browser.
3. Enter *admin* as the login ID and password.

NOTE: Figure 15 shows the overview page of mysimplelink.net.

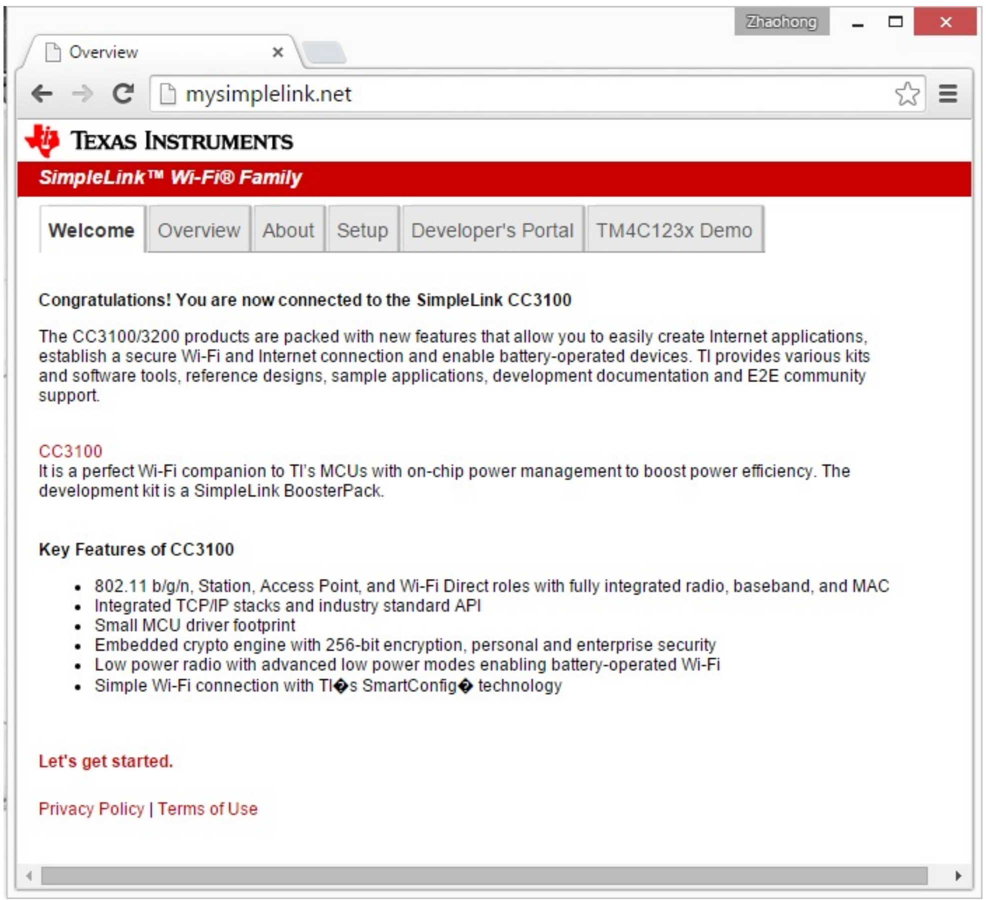


Figure 15. My SimpleLink Start Page

4. Select *TM4C123X Demo* from the top menu.

NOTE: Figure 16 shows the start page for the TM4X123 demonstration.

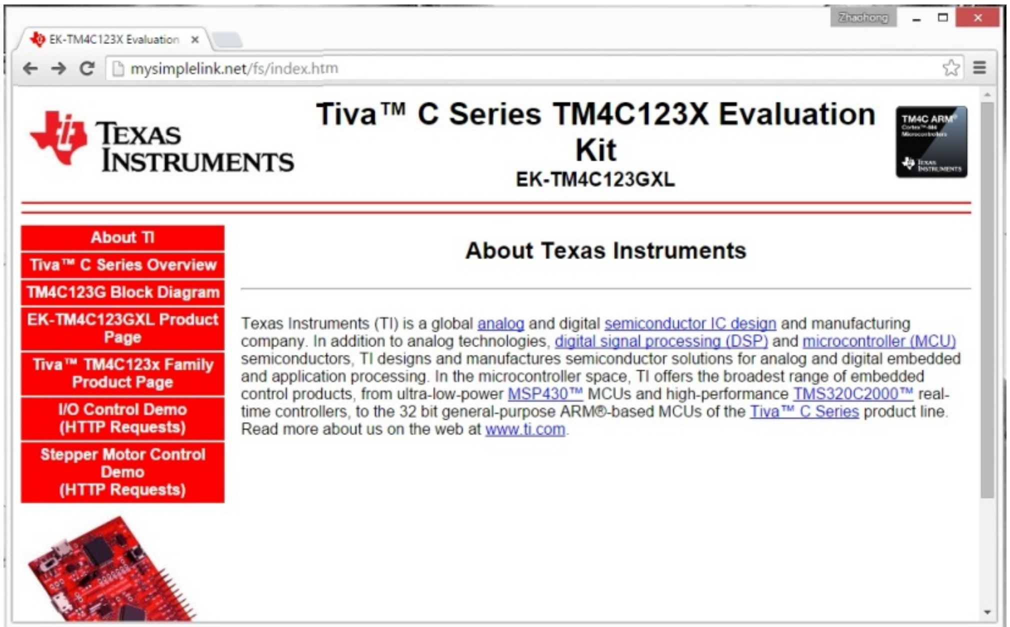


Figure 16. TM4C123 Demonstration Start Page

5. Click the *Stepper Motor Control* button from the menu to start the stepper motor control page as shown in Figure 16.
6. Click the radio buttons and input values in the text windows to control the motor.
7. Plug the debug USB cable into the EK-TM4C123GXL LaunchPad debug port.

NOTE: When the debug USB cable is plugged into the EK-TM4C123GXL LaunchPad debug port, a virtual COM port is created in the host PC.

8. Find the COM port number from the PC control panel.
9. Open the COM port with hyperterminal software.
10. Set the virtual COM port to 115200 baud and 8N1 data format.
11. Type *status* to display the motor control status.

6 Resources

To download the resource files for this reference design, see TIDM-TM4C123IOTSTEPPERMOTOR.

7 References

1. *TM4C123 MCU* (http://www.ti.com/product/tm4c123gh6pm)
2. *DRV8833 Stepper Motor Driver* (http://www.ti.com/product/drv8833)
3. *CC3100 Network Processor* (http://www.ti.com/product/cc3100)
4. *EK-TM4C123GXL LaunchPad* (http://www.ti.com/tool/EK-TM4C123GXL)
5. *DRV8833 EVM* (http://www.ti.com/tool/drv8833evm)
6. *SimpleLink CC3100 BoosterPack* (http://www.ti.com/tool/cc3100boost)
7. *TivaWare for C Series* (http://www.ti.com/tool/SW-TM4C)
8. *SimpleLink SDK* (http://www.ti.com/tool/cc3100sdk)
9. *Stellaris® In-Circuit Debug Interface (ICDI) and Virtual COM Port Driver Installation Instructions* (SPMU287)
10. *Uniflash for CC3100* (http://www.ti.com/tool/uniflash)
11. *High-Resolution Microstepping Driver With the DRV88xx Series* Application Report (SLVA416)
12. *Wi-Fi-Enabled IoT Node with High-Performance MCU Reference Design* (TIDM-TM4C129XWIFI)