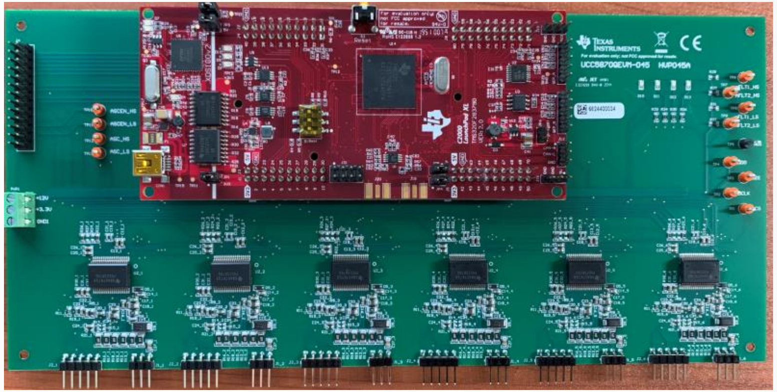
UCC5870QEVM-045 Quick Start Demo

Texas Instruments

Outline

- Introduction of UCC5870QEVM-045
- Test setup
- CCS software installation
- Open CCS and load example code into LaunchPad™ Development Kit
- Run code
- Scope measurements: output SPWMs

UCC5870QEVM-045 introduction:



Functional safety compliant 15-A isolated IGBT/SiC MOSFET gate driver three-phase EVM available now: EVM

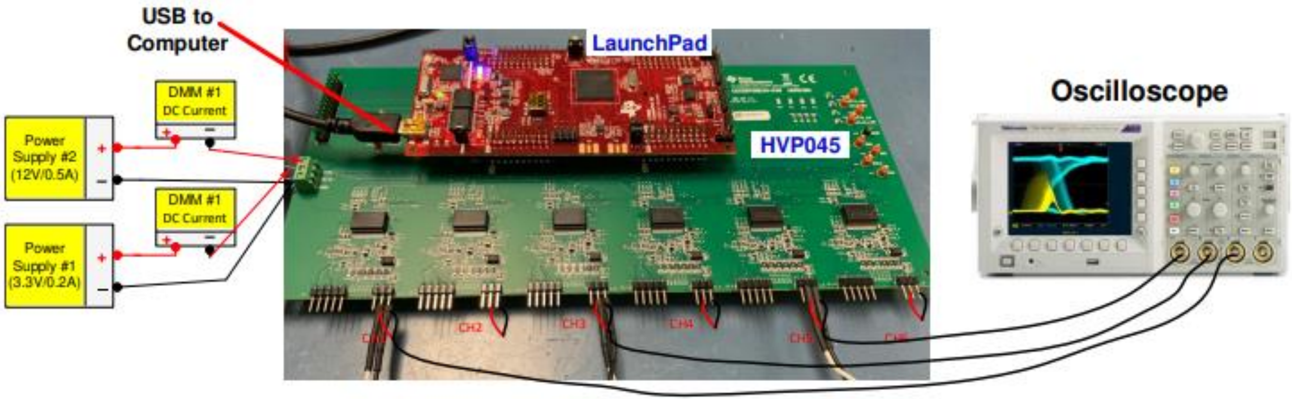
- Supports three-phase application software development
- Compatible with C2000™ MCU F28379D LaunchPad™ development kit (LAUNCHXL-F28379D)
- SPI-based device reconfiguration, verification, supervision and diagnosis
- Supports both address-mode SPI and daisy-chain mode SPI
- Power transistor protections: DESAT, OC/SC, Programmable soft turn off (STO) and two-level turn off (2LTOFF) during power transistor faults
- Split driver outputs guarantee 15-A peak source and 15-A peak sink currents

UCC5870-Q1 three phase EVM user guide

C2000™ MCU F28379D LaunchPad™ Development Kit

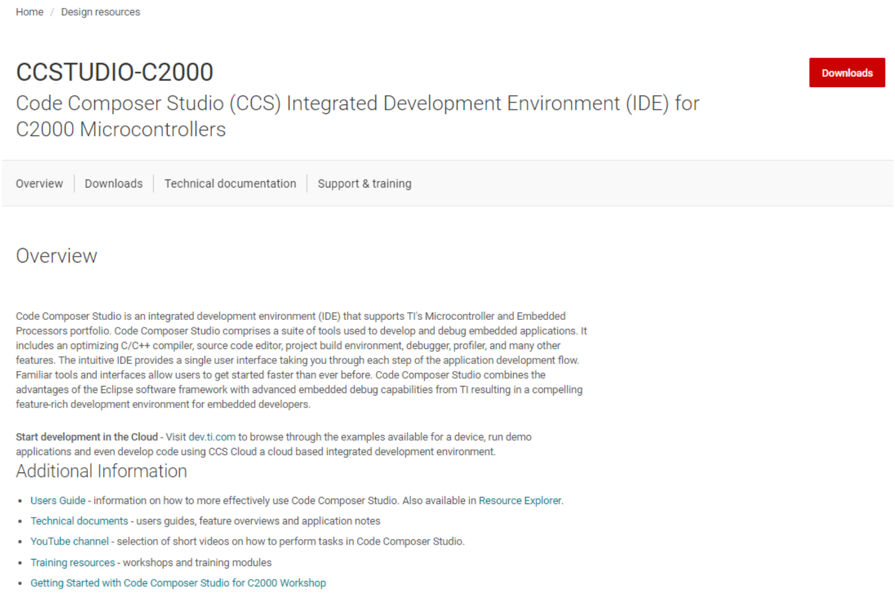
Test setup

1. Connect the Scope Channels as indicated. Ensure that the polarity on the channels are placed correctly
2. Connect the power supplies as indicated



Software installation

1. Visit https://www.ti.com/tool/LAUNCHXL-F28379D and download Code Composer Studio and C2000Ware.
2. Open the user's guide for Code Composer Studio and follow the Installation Process



CCSTUDIO-C2000

Code Composer Studio (CCS) Integrated Development Environment (IDE) for C2000 Microcontrollers

Overview

Code Composer Studio is an integrated development environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features. The intuitive IDE provides a single user interface taking you through each step of the application development flow. Familiar tools and interfaces allow users to get started faster than ever before. Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compelling feature-rich development environment for embedded developers.

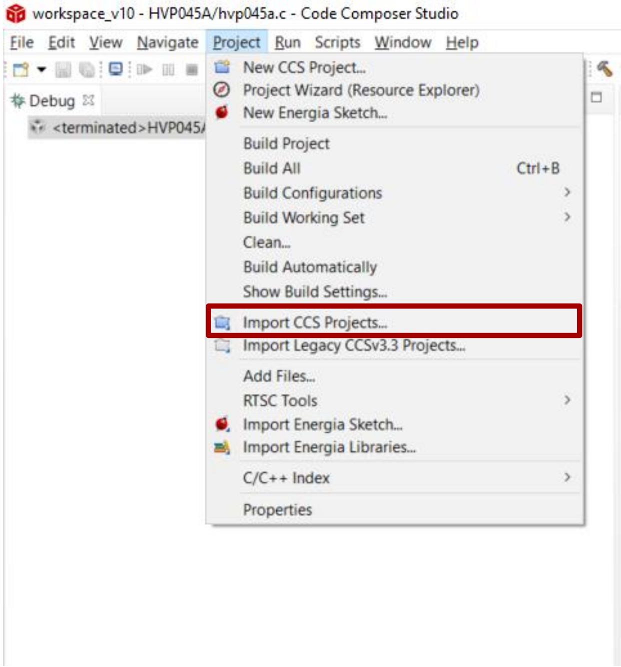
Start development in the Cloud - Visit dev.ti.com to browse through the examples available for a device, run demo applications and even develop code using CCS Cloud a cloud based integrated development environment.

Additional Information

- Users Guide - information on how to more effectively use Code Composer Studio. Also available in Resource Explorer.
- Technical documents - users guides, feature overviews and application notes
- YouTube channel - selection of short videos on how to perform tasks in Code Composer Studio.
- Training resources - workshops and training modules
- Getting Started with Code Composer Studio for C2000 Workshop

Open CCS and load example code into LaunchPad™ Development Kit

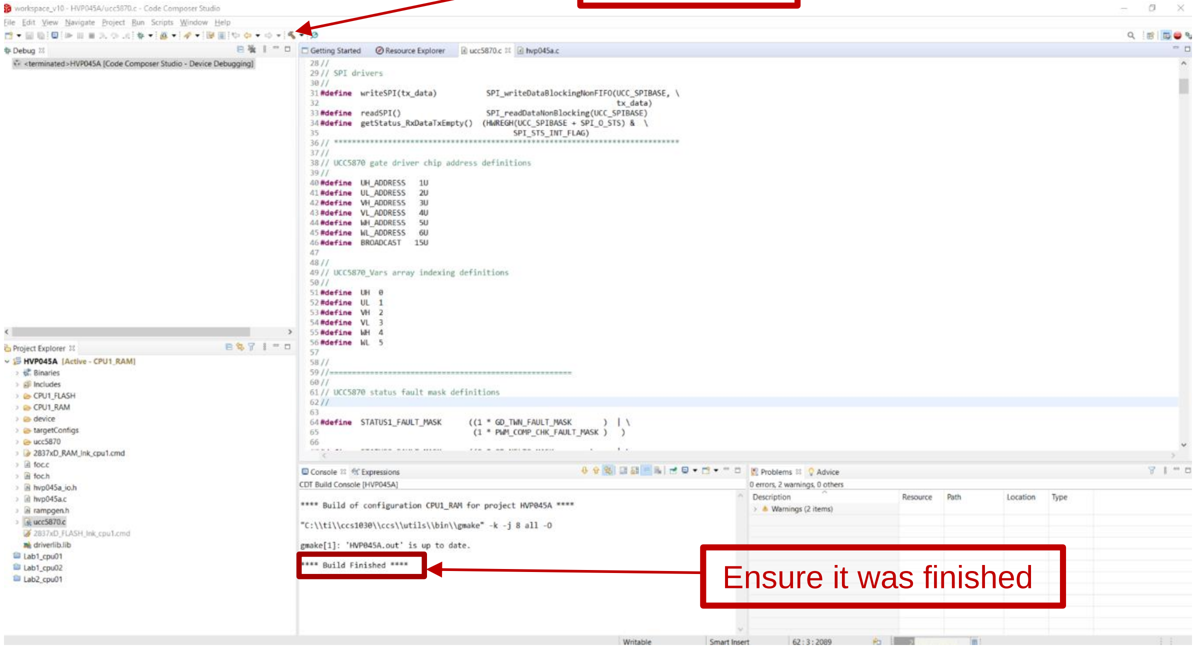
1. After Installation has been completed connect your C2000™ LaunchPad to your computer and import the project to Code Composer Studio



Build CCS project

1. After project has been imported, open ucc5870.c and build your project

Build project



```
//
// SPI drivers
//
#define  writeSPI(tx_data)          SPI_writeDataBlockingNonFIFO(UCC_SPIBASE, \
                                                                tx_data)
#define  readSPI()                  SPI_readDataNonBlocking(UCC_SPIBASE)
#define  getStatus_RxDataTxEmpty()  (HWREGH(UCC_SPIBASE + SPI_O_STS) &  \
                                          SPI_STS_INT_FLAG)
// ***************************************************************************
//
// UCC5870 gate driver chip address definitions
//
#define  UH_ADDRESS   1U
#define  UL_ADDRESS   2U
#define  VH_ADDRESS   3U
#define  VL_ADDRESS   4U
#define  WH_ADDRESS   5U
#define  WL_ADDRESS   6U
#define  BROADCAST   15U

//
// UCC5870_Vars array indexing definitions
//
#define  UH  0
#define  UL  1
#define  VH  2
#define  VL  3
#define  WH  4
#define  WL  5

//
//========================================================
//
// UCC5870 status fault mask definitions
//

#define  STATUS1_FAULT_MASK      ((1 * GD_TWN_FAULT_MASK       )  | \
                                  (1 * PWM_COMP_CHK_FAULT_MASK )   )

```

CDT Build Console [HVP045A]

```
**** Build of configuration CPU1_RAM for project HVP045A ****

"C:\\ti\\ccs1030\\ccs\\utils\\bin\\gmake" -k -j 8 all -O

gmake[1]: 'HVP045A.out' is up to date.

**** Build Finished ****
```

0 errors, 2 warnings, 0 others

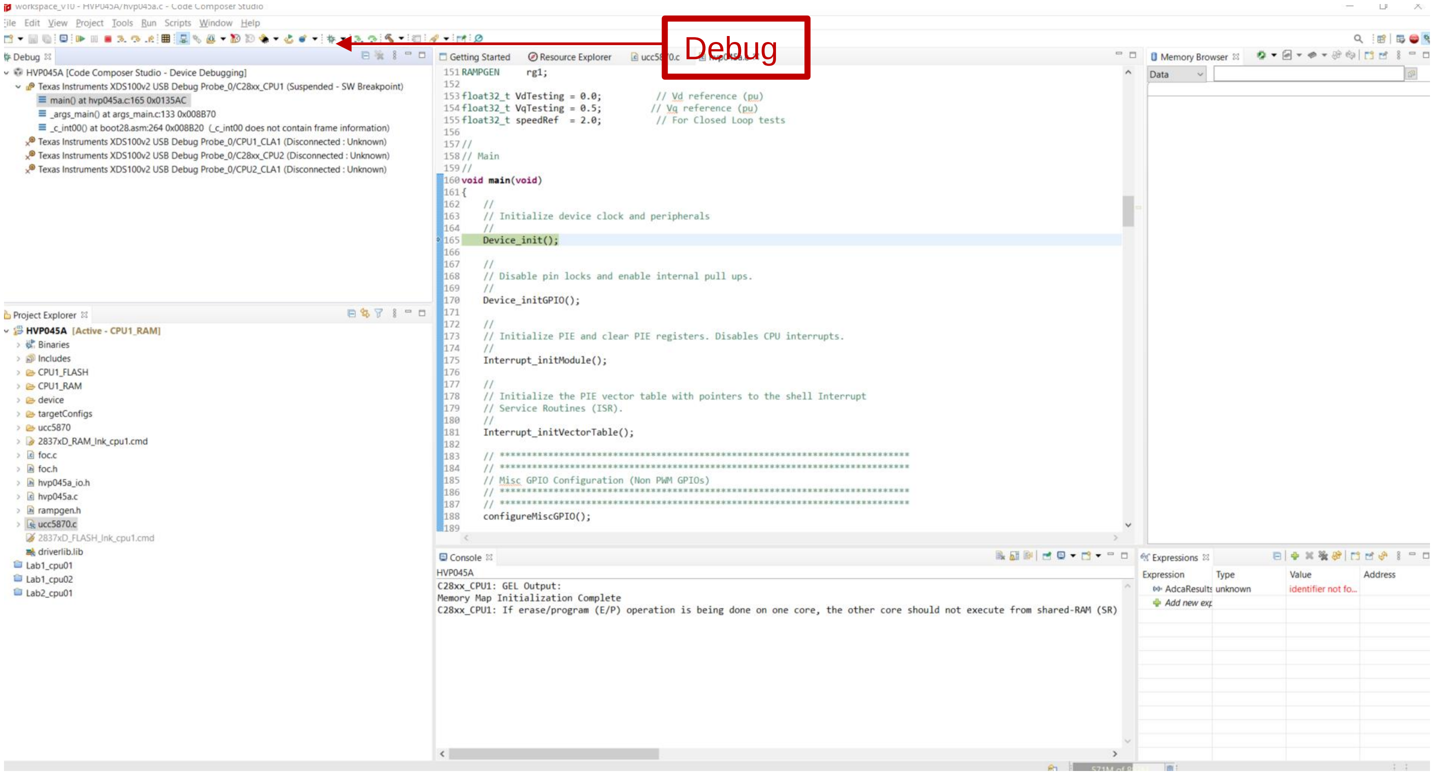
Description

Warnings (2 items)

Ensure it was finished

Load the project to the microcontroller

1. Upload the project into the microcontroller by selecting debug



HVP045A [Code Composer Studio - Device Debugging]
- Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU1 (Suspended - SW Breakpoint)
  - main() at hvp045a.c:165 0x0135AC
  - _args_main() at args_main.c:133 0x008B70
  - _c_int00() at boot28.asm:264 0x008B20 (_c_int00 does not contain frame information)
- Texas Instruments XDS100v2 USB Debug Probe_0/CPU1_CLA1 (Disconnected : Unknown)
- Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU2 (Disconnected : Unknown)
- Texas Instruments XDS100v2 USB Debug Probe_0/CPU2_CLA1 (Disconnected : Unknown)

HVP045A [Active - CPU1_RAM]
- Binaries
- Includes
- CPU1_FLASH
- CPU1_RAM
- device
- targetConfigs
- ucc5870
- 2837xD_RAM_lnk_cpu1.cmd
- foc.c
- foc.h
- hvp045a_io.h
- hvp045a.c
- rampgen.h
- ucc5870.c
- 2837xD_FLASH_lnk_cpu1.cmd
- driverlib.lib

Lab1_cpu01
Lab1_cpu02
Lab2_cpu01

```c
RAMPGEN     rg1;

float32_t VdTesting = 0.0;          // Vd reference (pu)
float32_t VqTesting = 0.5;         // Vq reference (pu)
float32_t speedRef  = 2.0;          // For Closed Loop tests

//
// Main
//
void main(void)
{
    //
    // Initialize device clock and peripherals
    //
    Device_init();

    //
    // Disable pin locks and enable internal pull ups.
    //
    Device_initGPIO();

    //
    // Initialize PIE and clear PIE registers. Disables CPU interrupts.
    //
    Interrupt_initModule();

    //
    // Initialize the PIE vector table with pointers to the shell Interrupt
    // Service Routines (ISR).
    //
    Interrupt_initVectorTable();

    // ****************************************************************************
    // ****************************************************************************
    // Misc GPIO Configuration (Non PWM GPIOs)
    // ****************************************************************************
    // ****************************************************************************
    configureMiscGPIO();
```

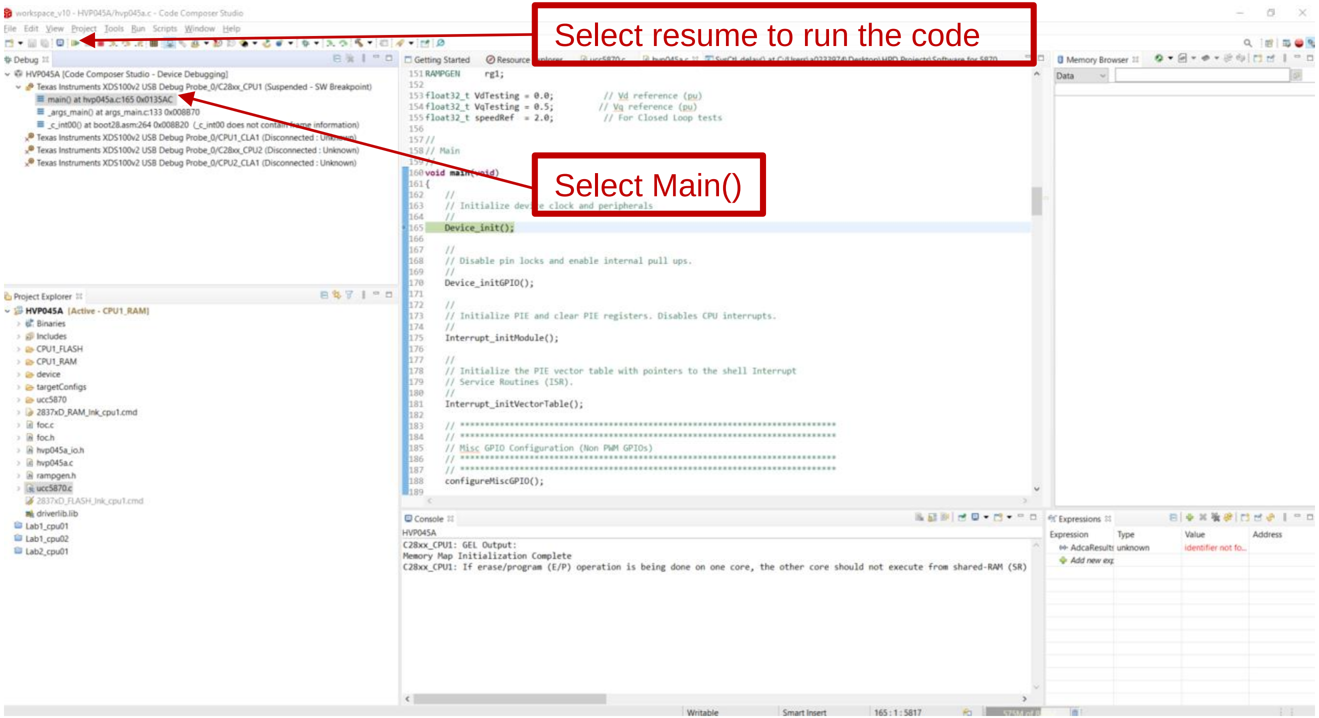
```
HVP045A
C28xx_CPU1: GEL Output:
Memory Map Initialization Complete
C28xx_CPU1: If erase/program (E/P) operation is being done on one core, the other core should not execute from shared-RAM (SR)
```

Expression	Type	Value	Address
AdcaResults	unknown	identifier not fo...	
Add new exp			

Debug

Run the code

1. Select main() and run the code.



workspace_v10 - HVP045A/hvp045a.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Debug
Getting Started
Resource Explorer
Memory Browser
Data

Select resume to run the code

Select Main()

```
HVP045A [Code Composer Studio - Device Debugging]
  Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU1 (Suspended - SW Breakpoint)
    main() at hvp045a.c:165 0x0135AC
    _args_main() at args_main.c:133 0x008B70
    _c_int00() at boot28.asm:264 0x008820  (_c_int00 does not contain frame information)
  Texas Instruments XDS100v2 USB Debug Probe_0/CPU1_CLA1 (Disconnected : Unknown)
  Texas Instruments XDS100v2 USB Debug Probe_0/C28xx_CPU2 (Disconnected : Unknown)
  Texas Instruments XDS100v2 USB Debug Probe_0/CPU2_CLA1 (Disconnected : Unknown)
```

```c
RAMPGEN      rg1;

float32_t VdTesting = 0.0;          // Vd reference (pu)
float32_t VqTesting = 0.5;         // Vq reference (pu)
float32_t speedRef  = 2.0;          // For Closed Loop tests

//
// Main
//
void main(void)
{
    //
    // Initialize device clock and peripherals
    //
    Device_init();

    //
    // Disable pin locks and enable internal pull ups.
    //
    Device_initGPIO();

    //
    // Initialize PIE and clear PIE registers. Disables CPU interrupts.
    //
    Interrupt_initModule();

    //
    // Initialize the PIE vector table with pointers to the shell Interrupt
    // Service Routines (ISR).
    //
    Interrupt_initVectorTable();

    // ****************************************************************************
    // ****************************************************************************
    // Misc GPIO Configuration (Non PWM GPIOs)
    // ****************************************************************************
    // ****************************************************************************
    configureMiscGPIO();
```

Project Explorer

```
HVP045A [Active - CPU1_RAM]
  Binaries
  Includes
  CPU1_FLASH
  CPU1_RAM
  device
  targetConfigs
  ucc5870
  2837xD_RAM_lnk_cpu1.cmd
  foc.c
  foc.h
  hvp045a_io.h
  hvp045a.c
  rampgen.h
  ucc5870.c
  2837xD_FLASH_lnk_cpu1.cmd
  driverlib.lib
Lab1_cpu01
Lab1_cpu02
Lab2_cpu01
```

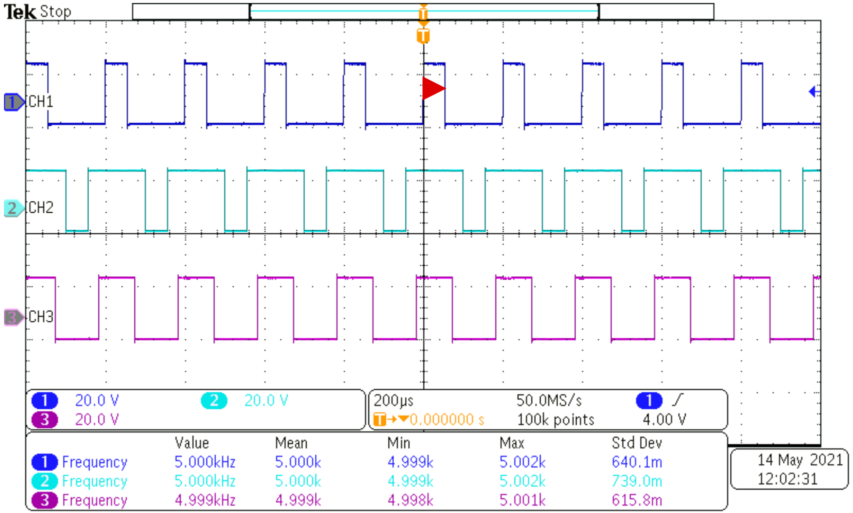
Console

```
HVP045A
C28xx_CPU1: GEL Output:
Memory Map Initialization Complete
C28xx_CPU1: If erase/program (E/P) operation is being done on one core, the other core should not execute from shared-RAM (SR)
```

Expressions

Expression	Type	Value	Address
AdcaResults	unknown	identifier not fo...	
Add new exp			

Scope measurements: output SPWMs

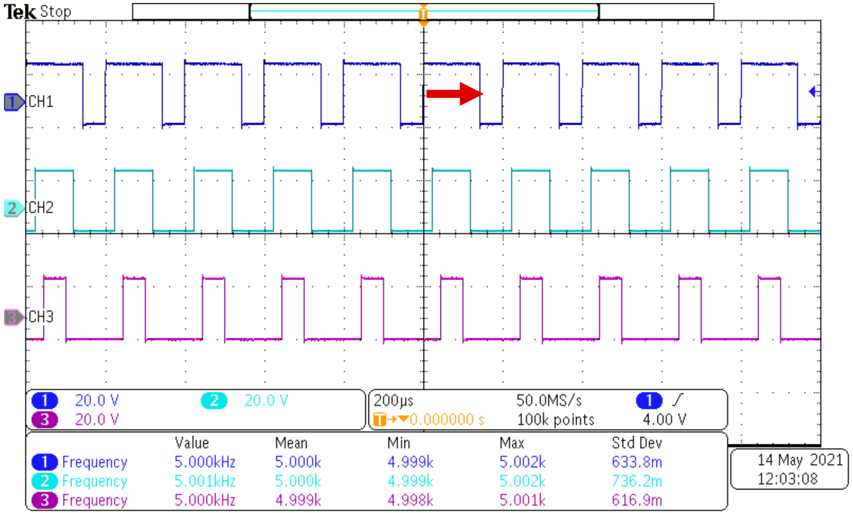


1 20.0 V 2 20.0 V
3 20.0 V

200µs 50.0MS/s 1 ⌠
T→▼0.000000 s 100k points 4.00 V

	Value	Mean	Min	Max	Std Dev
1 Frequency	5.000kHz	5.000k	4.999k	5.002k	640.1m
2 Frequency	5.000kHz	5.000k	4.999k	5.002k	739.0m
3 Frequency	4.999kHz	4.999k	4.998k	5.001k	615.8m

14 May 2021
12:02:31



1 20.0 V 2 20.0 V
3 20.0 V

200µs 50.0MS/s 1 ⌠
T→▼0.000000 s 100k points 4.00 V

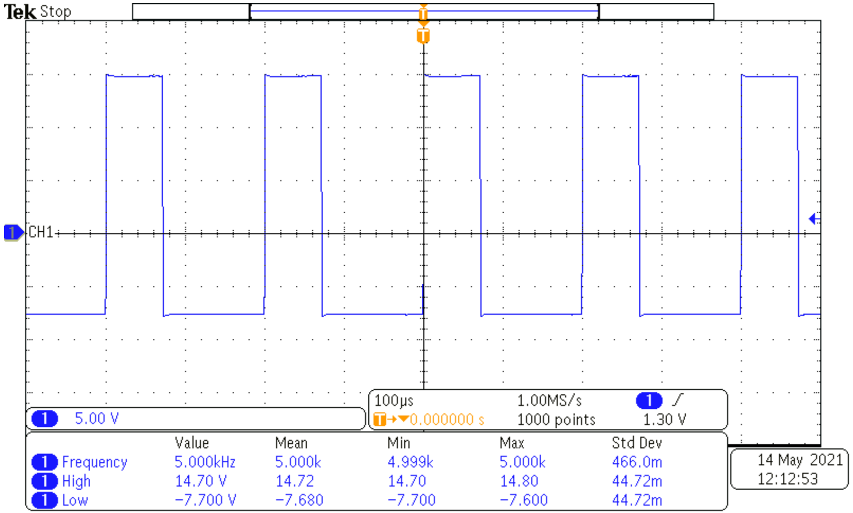
	Value	Mean	Min	Max	Std Dev
1 Frequency	5.000kHz	5.000k	4.999k	5.002k	633.8m
2 Frequency	5.001kHz	5.000k	4.999k	5.002k	736.2m
3 Frequency	5.000kHz	4.999k	4.998k	5.001k	616.9m

14 May 2021
12:03:08

Notice:

1. The output is open loop SPWM for all channels (varying duty cycle)
2. Frequency is 5 kHz

Scope measurements: output SPWMs

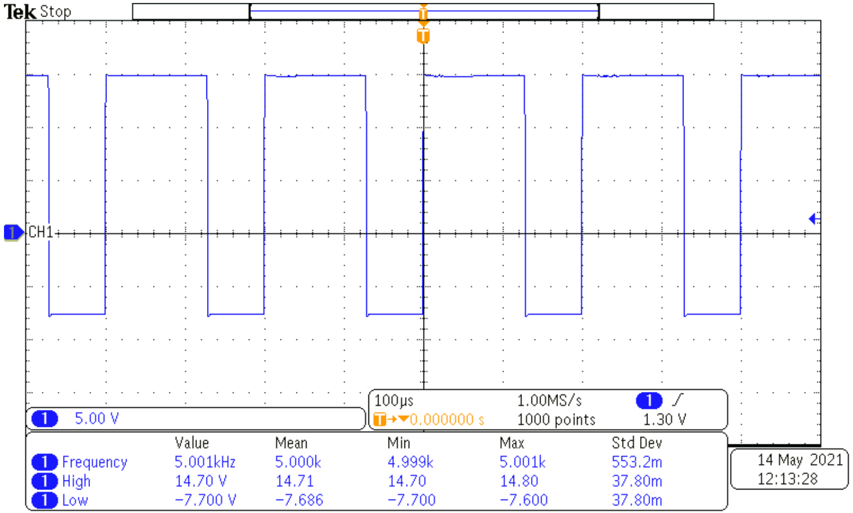


1 5.00 V

100µs 1.00MS/s 1 1.30 V
0.000000 s 1000 points

	Value	Mean	Min	Max	Std Dev
1 Frequency	5.000kHz	5.000k	4.999k	5.000k	466.0m
1 High	14.70 V	14.72	14.70	14.80	44.72m
1 Low	−7.700 V	−7.680	−7.700	−7.600	44.72m

14 May 2021
12:12:53



1 5.00 V

100µs 1.00MS/s 1 1.30 V
0.000000 s 1000 points

	Value	Mean	Min	Max	Std Dev
1 Frequency	5.001kHz	5.000k	4.999k	5.001k	553.2m
1 High	14.70 V	14.71	14.70	14.80	37.80m
1 Low	−7.700 V	−7.686	−7.700	−7.600	37.80m

14 May 2021
12:13:28

Notice:

1. The driver voltage is +15V/-7.7V
2. Frequency is 5 kHz