

MCU: F28P65x

Booster Board: DRV8316R

Motor: LVBLDC MTR (dn4240S24 026 24v)

Steps Followed:

1. We have taken the f28p65x example in the **universal motor control lab**.
2. Added the drv8316 configurations to **universal_motorcontrol_lab_p65x.projectspeg**

<!-- Launchxl-f28p65x + boostxl-drv8316rt, FAST Software Library, run out of FLASH !!!-Not Available -->

```
<configuration name="Flash_lib_DRV8316RT_3SC"
```

```
compilerBuildOptions="-v28 -ml -mt --float_support=fpu32 --idiv_support=idiv0 --  
tmu_support=tmu1 -O3 --opt_for_speed=3 --fp_mode=relaxed
```

```
-I${PROJECT_ROOT}/
```

```
-I${PROJECT_ROOT}/libraries/control/pi
```

```
-I${PROJECT_ROOT}/libraries/control/fwc
```

```
-I${PROJECT_ROOT}/libraries/control/mtpa
```

```
-I${PROJECT_ROOT}/libraries/control/vs_freq
```

```
-I${PROJECT_ROOT}/libraries/control/dclink_ss
```

```
-I${PROJECT_ROOT}/libraries/filter/filter_fo
```

```
-I${PROJECT_ROOT}/libraries/observers/est_lib
```

```
-I${PROJECT_ROOT}/libraries/observers/esmo
```

```
-I${PROJECT_ROOT}/libraries/observers/speedfr
```

```
-I${PROJECT_ROOT}/libraries/observers/speedcalc
```

```
-I${PROJECT_ROOT}/libraries/observers/ssipd
```

```
-I${PROJECT_ROOT}/libraries/observers/isbldc
```

```
-I${PROJECT_ROOT}/libraries/observers/encoder
```

```
-I${PROJECT_ROOT}/libraries/observers/hall
```

```
-I${PROJECT_ROOT}/libraries/transforms/clarke
-I${PROJECT_ROOT}/libraries/transforms/ipark
-I${PROJECT_ROOT}/libraries/transforms/park
-I${PROJECT_ROOT}/libraries/transforms/svgen
-I${PROJECT_ROOT}/libraries/transforms/volts
-I${PROJECT_ROOT}/libraries/utilities/angle_gen
-I${PROJECT_ROOT}/libraries/utilities/cpu_time
-I${PROJECT_ROOT}/libraries/utilities/datalog
-I${PROJECT_ROOT}/libraries/utilities/step_response
-I${PROJECT_ROOT}/libraries/utilities/traj/
-I${PROJECT_ROOT}/libraries/utilities/rimpulse
-I${PROJECT_ROOT}/libraries/utilities/mod6cnt
-I${PROJECT_ROOT}/libraries/utilities/types/include
-I${PROJECT_ROOT}/libraries/sfra
-I${PROJECT_ROOT}/libraries/dacs/dac128s085
-I${PROJECT_ROOT}/src_control
-I${PROJECT_ROOT}/src_control/common/include
-I${PROJECT_ROOT}/src_control/drivers/include
-I${PROJECT_ROOT}/src_board
-I${PROJECT_ROOT}/src_board/drvic
-I${PROJECT_ROOT}/src_device
-I${PROJECT_ROOT}/src_device/device_support/headers/include
-I${PROJECT_ROOT}/src_device/driverlib
--define=_INLINE --define=_FLASH --define=_F28P65x --define=CPU1
--define=_FULL_FAST_LIB
```

```
--define=BSXL8316RT_REVA

--define=MOTOR1_FAST
--define=MOTOR1_ESMO_N
--define=MOTOR1_ENC_N
--define=MOTOR1_HALL_N --define=HALL_CAL_N

--define=MOTOR1_OVM_N
--define=MOTOR1_FWC_N
--define=MOTOR1_MTPA_N
--define=MOTOR1_SSIPD_N
--define=MOTOR1_VOLRECT_N
--define=DATALOGF2_EN_N
--define=DAC128S_ENABLE --define=DAC128S_SPIB
--define=CPUTIME_ENABLE_N
--define=TEST_ENABLE_N
--define=SFRA_ENABLE_N
--define=STEP_RP_EN_N
--define=CMD_POT_EN_N
--define=CMD_CAP_EN_N
--define=CMD_SWITCH_EN_N
--define=CMD_CAN_EN_N

--gen_func_subsections=on --abi=eabi
--display_error_number --diag_warning=225 --diag_suppress=10063"
linkerBuildOptions="--entry_point code_start --stack_size=0x100
-I${PROJECT_ROOT}/libraries/sfra
```

```
-I${PROJECT_ROOT}/libraries/observers/est_lib
```

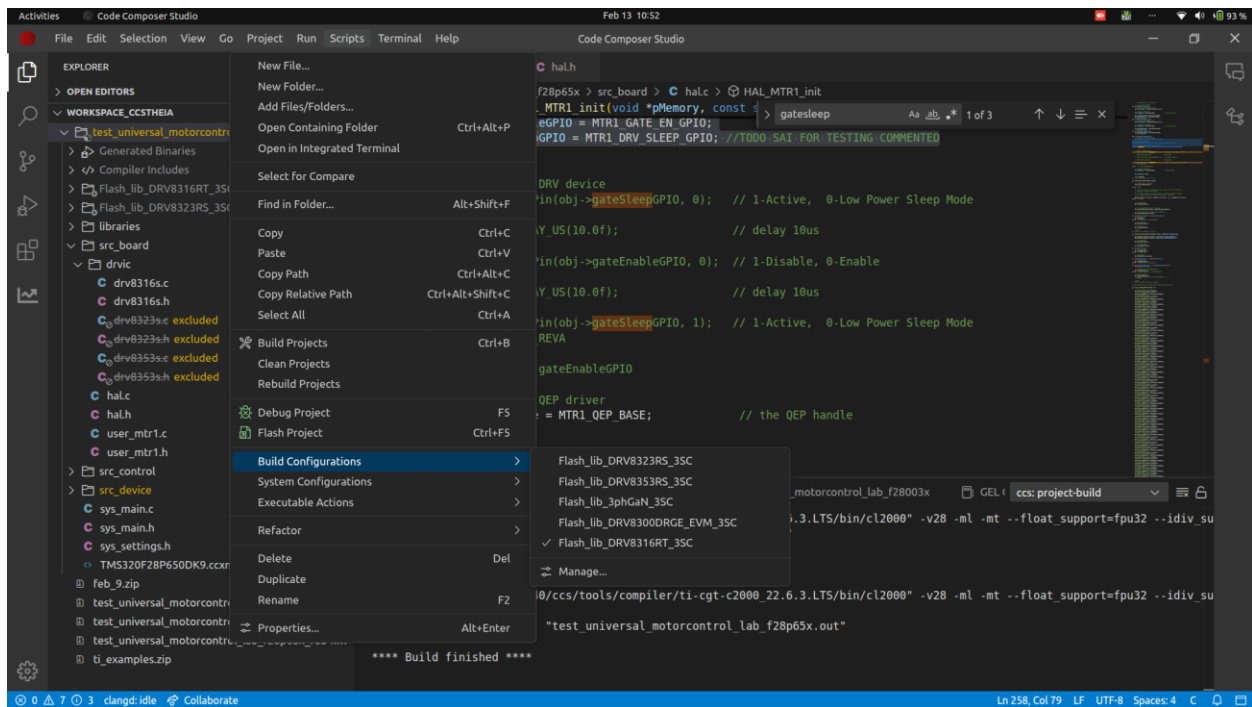
```
-l fast_full_lib.lib
```

```
-l sfra_f32_tmu_eabi.lib
```

```
--define=SFRA_ENABLE_N
```

```
-llibc.a -w" />
```

3. By Importing the project to CCS IDE, then we can see the flash_lib_drv8316RT_3sc in the build configurations



4.1.2.1 Setting Hardware Board Parameters

In the lab manual mentioned to use the **Motor Control Parameters Calculation.xlsx** file to change the hardware configuration, but in that the configuration for our drv board is missing (Not mentioned).

4.1.2.2 Modifying Motor Control Parameters

We changed the parameters as per the motor datasheet.

```
// the number of pole pairs of the motor  
#define USER_MOTOR1_NUM_POLE_PAIRS (4)  
  
// the rotor resistance value of the motor, in Ohm
```

```

#define USER_MOTOR1_Rr_Ohm (NULL)

// the stator resistance value of the motor, in Ohm
#define USER_MOTOR1_Rs_Ohm (0.73f) //(0.393955578f)

// the stator inductance value of the motor in the direct direction, in H
#define USER_MOTOR1_Ls_d_H (0.00109f) //(0.00190442806f)

// the stator inductance value of the motor in the quadrature direction, in H
#define USER_MOTOR1_Ls_q_H USER_MOTOR1_Ls_d_H //(0.000190442806f)

// the rated flux value of the motor, in V/Hz
#define USER_MOTOR1_RATED_FLUX_VpHz (0.0315f) //(0.0399353318f)

// the Id rated current value of the motor, in A. Induction motors only
#define USER_MOTOR1_MAGNETIZING_CURRENT_A (NULL)

// the maximum current value for stator resistance (R_s) identification, in A
#define USER_MOTOR1_RES_EST_CURRENT_A (0.5f) //(1.5f)

// the maximum current value to use for stator inductance identification, in A
#define USER_MOTOR1_IND_EST_CURRENT_A (-0.5f) //(-1.0f)

// the maximum current value of the motor, in A
#define USER_MOTOR1_MAX_CURRENT_A (3.0f) //(6.6f)

// the R/L excitation frequency for motor parameters identification, in Hz
#define USER_MOTOR1_FLUX_EXC_FREQ_Hz (30.0f) //(60.0f)

```

4.1.2.3 Changing Pin Assignment

Have changed the ADC, SPI, PWM, Other GPIO's as per the data sheet

	J1/J3	J3			J4/J2	J4
PIN NUMBER	F28P65x	Drv8316R		PIN NUMBER	F28P65x	Drv8316R
	1 3.3v MCU	3.3V			1 GPIO 8	INHA
	2 5V MCU	NULL			2 GND	gnd
	3 ADCINB0/VDAC	NULL			3 GPIO 9	INLA
	4 GND	GND			4 GPIO 57	nscs

5	GPIO 43 - BP			5	GPIO 6	INH B
6	ADCINA14/B14/C14	VSENA		6	GPIO 84	GPIO 84
7	GPIO 42 - BP			7	GPIO 7	INLB
8	ADCIN4	VSENB		8	GPIO 21_BP	GPIO 21_BP
9	GPIO 42- BP	NSLEEP		9	GPIO 10	INH C
10	ADCINB4	VSENC		10	XRSn	XRSn
11	ADCINC0			11	GPIO 11	INLC
12	ADCINC4	VSENV M		12	GPIO 16	SDI
13	GPIO18	SCLK		13	GPIO 14	HALLA
14	ADCINA7	ISENA		14	GPIO 17	SDO
15	GPIO72	FAULT		15	GPIO 15	HALLB
16	ADCINB7	ISENB		16	GPIO 12	DRVOFF
17	GPIO105	N SLEEP		17	GPIO 4_BP	HALLC
18	ADCINC7	ISENC		18	GPIO 23_BP	
19	GPIO104	N Fault		19	GPIO 5_Bp	VREF
20	ADCINA0/DACA_OUT	C TAP		20	GPIO 133	MCU LED

ADD THE TABLE OF THE GPIO OF F28P65X AND DRV316.

4.1.2.4 Configuring the PWM Module

```
#define MTR1_PWM_U_BASE EPWM5_BASE
#define MTR1_PWM_V_BASE EPWM4_BASE
#define MTR1_PWM_W_BASE EPWM6_BASE
```

```
#define MTR1_ADC_TRIGGER_SOC ADC_TRIGGER_EPWM5_SOC // EPWM5_SOC
```

We haven't changed the base address registers for PWM as we have taken the base board as f28p65x.

**4.1.2.5 Configuring the ADC Module (Suspecting)

We have compared the pin configuration of drv8316 and drv8323. We got to know that the ADC pins are mismatching with the drv8323. As f28p65x is configured with the drv8323, we have changed the mismatched pins and configured the ADC base addresses.

```
#define MTR1_IU_ADC_BASE ADCA_BASE //ADCB_BASE TODO SAI CHANGED // ADCB-B7* -SOC0-PPB1
#define MTR1_IV_ADC_BASE ADCB_BASE //ADCA_BASE TODO SAI CHANGED // ADCA-A7* -SOC0-PPB1
#define MTR1_IW_ADC_BASE ADCC_BASE // ADCC-C4* -SOC0-PPB1

#define MTR1_IU_ADCRES_BASE ADCARESULT_BASE //ADCBRESULT_BASE TODO SAI CHANGED
#define MTR1_IV_ADCRES_BASE ADCBRESULT_BASE //ADCARESULT_BASE TODO SAI CHANGED
#define MTR1_IW_ADCRES_BASE ADCCRESULT_BASE

#define MTR1_IU_ADC_CH_NUM ADC_CH_ADCIN7
#define MTR1_IV_ADC_CH_NUM ADC_CH_ADCIN7
#define MTR1_IW_ADC_CH_NUM ADC_CH_ADCIN7 //ADC_CH_ADCIN4 TODO SAI CHANGED

#define MTR1_IU_ADC_SOC_NUM ADC_SOC_NUMBER0
#define MTR1_IV_ADC_SOC_NUM ADC_SOC_NUMBER0
#define MTR1_IW_ADC_SOC_NUM ADC_SOC_NUMBER0

#define MTR1_IU_ADC_PPB_NUM ADC_PPB_NUMBER1
#define MTR1_IV_ADC_PPB_NUM ADC_PPB_NUMBER1
#define MTR1_IW_ADC_PPB_NUM ADC_PPB_NUMBER1

// Analog Subsystem Mux select
#define MTR1_IU_CMPHP_MUX 2 // ADCB-B7* - HP - CMPSS7 sel2
#define MTR1_IU_CMPLP_MUX 2 // ADCB-B7* - LP - CMPSS7 sel2

#define MTR1_IV_CMPHP_MUX 2 // ADCA-A7* - HP - CMPSS9 sel2
#define MTR1_IV_CMPLP_MUX 3 // ADCA-A7* - LP - CMPSS4 sel3

#define MTR1_IW_CMPHP_MUX 0 // ADCC-C4* - HP - CMPSS5 sel0
#define MTR1_IW_CMPLP_MUX 0 // ADCC-C4* - LP - CMPSS5 sel0

// ePWM XBAR MUX select
#define MTR1_IU_XBAR_EPWM_MUX_H XBAR_EPWM_MUX12_CMPSS7_CTRIPH // CMPSS7-HP
#define MTR1_IU_XBAR_EPWM_MUX_L XBAR_EPWM_MUX13_CMPSS7_CTRIPL // CMPSS7-LP
#define MTR1_IV_XBAR_EPWM_MUX_H XBAR_EPWM_MUX58_CMPSS9_CTRIPH // CMPSS9-HP
#define MTR1_IV_XBAR_EPWM_MUX_L XBAR_EPWM_MUX07_CMPSS4_CTRIPL // CMPSS4-LP
#define MTR1_IW_XBAR_EPWM_MUX_H XBAR_EPWM_MUX08_CMPSS5_CTRIPH // CMPSS5-HP
#define MTR1_IW_XBAR_EPWM_MUX_L XBAR_EPWM_MUX09_CMPSS5_CTRIPL // CMPSS5-LP
```

```

#define MTR1_IU_XBAR_MUX_H XBAR_MUX12 // CMPSS7-HP
#define MTR1_IU_XBAR_MUX_L XBAR_MUX13 // CMPSS7-LP
#define MTR1_IV_XBAR_MUX_H XBAR_MUX58 // CMPSS9-HP
#define MTR1_IV_XBAR_MUX_L XBAR_MUX07 // CMPSS4-LP
#define MTR1_IW_XBAR_MUX_H XBAR_MUX08 // CMPSS5-HP
#define MTR1_IW_XBAR_MUX_L XBAR_MUX09 // CMPSS5-LP

//V U, V, W
#define MTR1_VU_ADC_BASE ADCC_BASE // ADCA-A14/B14/*C14 -SOC1
#define MTR1_VV_ADC_BASE ADCA_BASE // ADCA-A4* -SOC1
#define MTR1_VW_ADC_BASE ADCB_BASE // ADCB-B4* -SOC1

#define MTR1_VU_ADCRES_BASE ADCCRESULT_BASE
#define MTR1_VV_ADCRES_BASE ADCARESLUT_BASE
#define MTR1_VW_ADCRES_BASE ADCBRESULT_BASE

#define MTR1_VU_ADC_CH_NUM ADC_CH_ADCIN14
#define MTR1_VV_ADC_CH_NUM ADC_CH_ADCIN4
#define MTR1_VW_ADC_CH_NUM ADC_CH_ADCIN4

#define MTR1_VU_ADC_SOC_NUM ADC_SOC_NUMBER1
#define MTR1_VV_ADC_SOC_NUM ADC_SOC_NUMBER1
#define MTR1_VW_ADC_SOC_NUM ADC_SOC_NUMBER1

// VDC, POT
#define MTR1_POT_ADC_BASE ADCC_BASE // ADCC-C0* -SOC2
#define MTR1_VDC_ADC_BASE ADCB_BASE // ADCB-B0* -SOC2-INT1

#define MTR1_POT_ADCRES_BASE ADCCRESULT_BASE
#define MTR1_VDC_ADCRES_BASE ADCBRESULT_BASE

#define MTR1_POT_ADC_CH_NUM ADC_CH_ADCIN0
#define MTR1_VDC_ADC_CH_NUM ADC_CH_ADCIN4 //ADC_CH_ADCIN0 TODO SAI CHANGED

#define MTR1_POT_ADC_SOC_NUM ADC_SOC_NUMBER2
#define MTR1_VDC_ADC_SOC_NUM ADC_SOC_NUMBER2

//-----
// interrupt
// VDC
#define MTR1_ADC_INT_BASE MTR1_VDC_ADC_BASE // ADCB-B0*-SOC2-INT1
#define MTR1_ADC_INT_NUM ADC_INT_NUMBER1
#define MTR1_ADC_INT_SOC MTR1_VDC_ADC_SOC_NUM

#define MTR1_PIE_INT_NUM INT_ADCB1
#define MTR1_INT_ACK_GROUP INTERRUPT_ACK_GROUP1 // ADCB_B0*-SOC2-INT1-CPU_INT1

```

4.1.2.6 Configuring the CMPSS Module

```
// CMPSS
#define MTR1_CMPSS_U_BASE_H CMPSS7_BASE // ADCB-B7* - HP - CMPSS7 sel2
#define MTR1_CMPSS_U_BASE_L CMPSS7_BASE // ADCB-B7* - LP - CMPSS7 sel2

#define MTR1_CMPSS_V_BASE_H CMPSS9_BASE // ADCA-A7* - HP - CMPSS9 sel2
#define MTR1_CMPSS_V_BASE_L CMPSS4_BASE // ADCA-A7* - LP - CMPSS4 sel3

#define MTR1_CMPSS_W_BASE_H CMPSS5_BASE // ADCC-C4* - HP - CMPSS5 sel0
#define MTR1_CMPSS_W_BASE_L CMPSS5_BASE // ADCC-C4* - LP - CMPSS5 sel0

// CMPSS selection
#define MTR1_IU_CMPHP_SEL ASYSCTL_CMPHPMUX_SELECT_7 // ADCB-B7* - HP - CMPSS7 sel2
#define MTR1_IU_CMPLP_SEL ASYSCTL_CMPLPMUX_SELECT_7 // ADCB-B7* - LP - CMPSS7 sel2

#define MTR1_IV_CMPHP_SEL ASYSCTL_CMPHPMUX_SELECT_9 // ADCA-A7* - HP - CMPSS9 sel2
#define MTR1_IV_CMPLP_SEL ASYSCTL_CMPLPMUX_SELECT_4 // ADCA-A7* - LP - CMPSS4 sel3

#define MTR1_IW_CMPHP_SEL ASYSCTL_CMPHPMUX_SELECT_5 // ADCC-C4* - HP - CMPSS5 sel0
#define MTR1_IW_CMPLP_SEL ASYSCTL_CMPLPMUX_SELECT_5 // ADCC-C4* - LP - CMPSS5 sel0
```

4.1.2.7 Configuring Fault Protection Function

In this function, we have just changed the nfault gpio pin to 72

```
#define MTR1_PM_nFAULT_GPIO 72
```

4.1.3 Adding Additional Functionality to Motor Control Project

We haven't added any other additional functionality to the motor control project like potentiometer, push buttons ...

Hardware parameters

```
//for BSXL8316RT_REVA

#define USER_M1_NOMINAL_DC_BUS_VOLTAGE_V 20.0f //(48.0f)
#define USER_M1_ADC_FULL_SCALE_VOLTAGE_V (44.28f)
#define USER_M1_VOLTAGE_FILTER_POLE_Hz (284.722f) // 100nF
#define USER_M1_ADC_FULL_SCALE_CURRENT_A (22.0f) // GAIN = 0.15V/A
#define USER_M1_IDC_OFFSET_AD (516.717f) // ~ = 4096.0/8.0
```

```
//! \brief ADC current offset for CMPSS
#define USER_M1_IDC_OFFSET_CMPSS (uint16_t)(USER_M1_IDC_OFFSET_AD)

#define USER_M1_IDC_OFFSET_AD_MAX (USER_M1_IDC_OFFSET_AD + 100.0f)
#define USER_M1_IDC_OFFSET_AD_MIN (USER_M1_IDC_OFFSET_AD - 100.0f)
#define USER_M1_SIGN_CURRENT_SF (1.0f)

//! \brief ADC current offsets for A, B, and C phases
#define USER_M1_IA_OFFSET_AD (1869.6f)
#define USER_M1_IB_OFFSET_AD (1877.3f)
#define USER_M1_IC_OFFSET_AD (1876.9f)

//! \brief ADC current offset for CMPSS
#define USER_M1_IS_OFFSET_CMPSS (uint16_t)((USER_M1_IA_OFFSET_AD + USER_M1_IB_OFFSET_AD +
USER_M1_IC_OFFSET_AD) / 3.0f)

//! \brief ADC voltage offsets for A, B, and C phases
#define USER_M1_VA_OFFSET_SF (0.477463514f)
#define USER_M1_VB_OFFSET_SF (0.479326069f)
#define USER_M1_VC_OFFSET_SF (0.477675498f)

//! \brief DC bus over voltage threshold
#define USER_M1_OVER_VOLTAGE_FAULT_V (40.0f)

//! \brief DC bus over voltage threshold
#define USER_M1_OVER_VOLTAGE_NORM_V (36.0f)

//! \brief DC bus under voltage threshold
#define USER_M1_UNDER_VOLTAGE_FAULT_V (10.0f)

//! \brief DC bus under voltage threshold
#define USER_M1_UNDER_VOLTAGE_NORM_V (12.0f)

//! \brief motor lost phase current threshold
#define USER_M1_LOST_PHASE_CURRENT_A (0.2f)

//! \brief motor unbalance ratio percent threshold
#define USER_M1_UNBALANCE_RATIO (0.2f)

//! \brief motor over load power threshold
#define USER_M1_OVER_LOAD_POWER_W (50.0f)

//! \brief motor stall current threshold
#define USER_M1_STALL_CURRENT_A (10.0f)

//! \brief motor fault check current threshold
#define USER_M1_FAULT_CHECK_CURRENT_A (0.2f)
```

```
//! \brief motor failed maximum speed threshold
#define USER_M1_FAIL_SPEED_MAX_HZ (500.0f)

//! \brief motor failed minimum speed threshold
#define USER_M1_FAIL_SPEED_MIN_HZ (5.0f)

//! \brief Defines the number of failed torque
#define USER_M1_TORQUE_FAILED_SET (0.000001f)
```