3-PHASE POWER INVERTER

Guide to operate the 3-phase inverter experimental setup  
([v.morais@fe.up.pt](mailto:v.morais@fe.up.pt))

# Features

* The 3-phase 30A@450V power inverter for academic/research purposes;
* This inverter has built-in measurements of AC currents, AC voltages and DC bus voltage;
* The inverter is based on infineon IPM IKCM30F60GA module (a low-cost module, with integrated drivers);
* The control board is based on XMC4500 Relax (lite) kit, a well-known board in UP Laboratory of Power Electronics;

# Safety Issues

This module and all electric equipment’s are under **dangerous voltages**. The operation of this setup may require supervision to avoid injuries as well as destruction of equipment’s.

Next, some advices are listed:

* Keep the “shutdown” button of the DC BUS voltage source easily accessible;
* Always ensure the measurement of the DC bus voltage; A continuous current measurement with an oscilloscope current clamp is recommended.
* Avoid short-circuits with the scope’s ground wire: differential measurements are recommended;
* The isolation transformer is mandatory;
* The heatsink is not grounded. Despite being safe to touch the heatsink it is not recommended. A temperature measurement of the IGBT is available.
* Avoid plug/unplug loads under operation.

# How to use

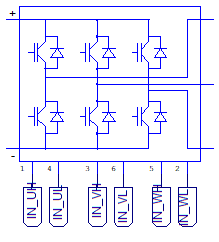
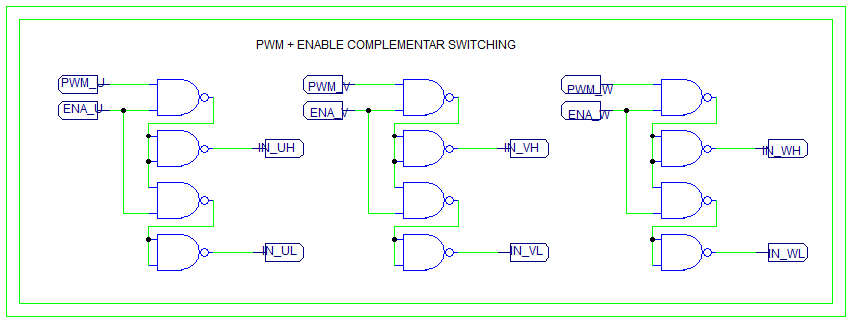
* SETUP:
  + Download/read IKCM30F60GA datasheets
    - <https://www.infineon.com/dgdl/Infineon-IKCM30F60GA-DS-v02_06-EN.pdf?fileId=5546d4624fb7fef2014fcb83cab8790a>
  + Download and install last version of DAVE and micrium
    - <https://www.infineon.com/cms/en/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-cortex-m/>
    - <https://infineoncommunity.com/dave-download_ID645>
    - <https://infineoncommunity.com/uC-Probe-XMC-software-download_ID712>
  + Clone github repository for this 3-phase inverter
    - <https://github.com/vitormorais/3phase_inverter>
* FIRST RUN (software)
  + Open Dave, import repository:
    - File->Import->General->Existing Projects into Workspace->Select archive file
    - *“\3phase\_inverter\4. Software\_WS\SELE\_initialProject.zip”*
    - Open main.c;
    - On C/C++ Projects tab, and on SELE\_initialProject, right click and “Set active Project”; Then right click and “Build Configurations->Set active->DEBUG”
  + Connect to XMC4500 via USB; click on DEBUG 
    - It might ask to create a debug configuration: Double click on “GBD SEGGER JLink Debugging”, then press “Search Project…”, then select the debug binary “SELE\_initialProject.elf” and then press DEBUG
    - It should compile and flash the microcontroller; press run button on debug environment.
  + Open micrium file “SELE\_micrium.wspx” via DAVE CE workspace
  + If requested, select the SELE\_initialProject.elf binary
  + Put in oscilloscope window and press “play” button
  + Now you can control the modulation index, the frequency, etc and monitor certain waveforms
* SETUP HARDWARE
  + Connect to the DC bus a voltage source
    - First use the LAB voltage source (max 30V)
    - Check with the oscilloscope probe the ac voltage of each of the 3phase outputs (with the scope ground connected to the negative side of –VDC\_BUS)
      * You should see the PWM at 10 kHz
  + Now, if all PWM are OK, connect a DC power source (isolation transformer + VARIAC + Rectifier)
  + Connect a 3phase load (RL) to the respective inverter outputs
  + Turn on the DC voltage source (with 30V); Check the LOAD current with a current probe: it should be a 50Hz current;
  + Now test the micrium interface with different frequencies, modulation indexes, etc

# SCHEMATIC



# PWM Generation waveforms

The PWM generation implemented in this power inverter module uses the following logic circuit:



This arrangement allows the generation of two complementary PWM for each inverter arm. In addition, there is an enable signal for each inverter arm.

On the XMC4500 side, the PWM pins are the following: **P0.5, P0.6** and **P0.4**; Respectively, the **P0.8**, **P0.12** and **P0.2**

# Acquisition of electric waveforms

This setup has built-in measurements of electric waveforms, specifically three currents (for each of the phases), the DC voltage and three AC isolated voltage measurements.

## Current Measurements (three measurements)

The current measurement is performed with an isolated hall effect based current sensor, specifically the [ACHS-7123-000E](https://www.broadcom.com/products/optocouplers/industrial-plastic/other/hall-effect-sensors/achs-7123-000e). The maximum value for the AC current sensor is +/- 30A peak.



In the provided code, the acquisition is performed at the beginning of the 10 kHz interruption, in the “Update\_PWM(void)” function:

**/// ADC acquisition - AC Currents**

I\_abc\_adc[0]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_2\_V2\_2);

I\_abc\_adc[1]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_1\_V1\_4);

I\_abc\_adc[2]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_1\_V1\_3);

I\_abc[0]=(I\_abc\_adc[0]- (1800.0f + 74.8f)) \* 0.022222222f;

I\_abc[1]=(I\_abc\_adc[1]- (1800.0f + 77.0f)) \* 0.022179506f;

I\_abc[2]=(I\_abc\_adc[2]- (1800.0f + 70.8f)) \* 0.021897810f;

Due to the precision of the resistors in the resistor divider, an offset and gain error occurs. A practical calibration of each sensor is recommended, with the calibration procedures presented in section 5.4.

## DC\_BUS Voltage Measurement

The DC bus voltage measurement is performed with an isolation amplifier, namely the [ACM1200](http://www.ti.com/product/AMC1200), as illustrated in the following figure:



Similarly, in the provided code, the acquisition is performed at the beginning of the 10 kHz interruption, in the “Update\_PWM(void)” function:

**/// ADC acquisition – DC\_BUS Voltage**

V\_DC\_adc = ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_1\_V1\_2);

V\_DC = (V\_DC\_adc-(1800.0f + 22.4f)) \* -0.58f;

Due to the precision of the resistors, an offset and gain error occurs. A practical calibration of each sensor is recommended, with the calibration procedures presented in section 5.4.

## AC Voltage Measurements (Three Measurements)

Similarly to the DC Bus Voltage measurement, AC voltage measurement is performed with an isolation amplifier, as illustrated in the following figure:



In the provided code, the acquisition is performed at the beginning of the 10 kHz interruption, in the “Update\_PWM(void)” function:

**/// ADC acquisition - AC Voltages**

V\_abc\_adc[0]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_1\_V1\_1);

V\_abc\_adc[1]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_2\_V2\_1);

V\_abc\_adc[2]=ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_2\_V2\_3);

V\_abc[0]=(V\_abc\_adc[0]- (1800.0f + 23.92f)) \* 0.720f;

V\_abc[1]=(V\_abc\_adc[1]- (1800.0f + 16.50f)) \* 0.720f;

V\_abc[2]=(V\_abc\_adc[2]- (1800.0f + 2.930f)) \* 0.720f;

Due to the precision of the resistors in the resistor divider, an offset and gain error occurs. A practical calibration of each sensor is recommended, with the calibration procedures presented in section 5.4.

## Calibration procedures for the measurements

As visible in the code, the calibration was performed, having the form:

I\_abc[k]=(I\_abc\_adc[k]-OFFSET) \* GAIN;

**STEP1**: set GAIN =1 and OFFSET = 0; program the microcontroller and open the micrium code;

**STEP2**: obtain the OFFSET, based on the measurement of the RMS value displayed in the oscilloscope interface of the micrium;

**STEP3**: adjust the program to include the offset. Download and run again the code.

**STEP4.1**: For the DC voltage, increase the DC bus voltage, and obtain the gain with the formula: Gain = V(real)/V(micrium)

**STEP4.2:** For the AC voltages, connect an AC voltage with a VARIAC and obtain the gain by comparing the RMS value of micrium and the multimeter;

**STEP4.3**: For the AC Currents, connect a RL load, enable the PWM, supply the DC bus and measure the current with the oscilloscope; compare the measured value with the displayed value in the micrium; adjust the gain accordingly.

## Temperature measurement

Based on the built-in thermistor of IKCM30F60GA IGBT module, the temperature is acquired, with the following code:

float conv\_temperature = ADC\_MEASUREMENT\_ADV\_GetResult(&ADC\_MEASUREMENT\_3\_V3\_1);

float conv\_temp\_C =(conv\_temperature\*conv\_temperature\*-0.00016973) + //x^2

(conv\_temperature\*0.69969095) + //x^1

-653.10731030; //x^0

conv\_temp\_C += 7.0f; //offset compensation (junction temperature)

# Version control

V1.0 – preliminary – [v.morais@fe.up.pt](mailto:v.morais@fe.up.pt)

V1.1 – added PWM generation details, ADC details and minor updates in the How to use section – [v.morais@fe.up.pt](mailto:v.morais@fe.up.pt)

V1.2 -