BUREAU D'ÉTUDE ÉLECTRONIQUE AUTOMOBILE - DETAILS OF PROGRAM PMSM_TGN002_CDE_UF

Alexandre Boyer
alexandre.boyer@insa-toulouse.fr

November 2020

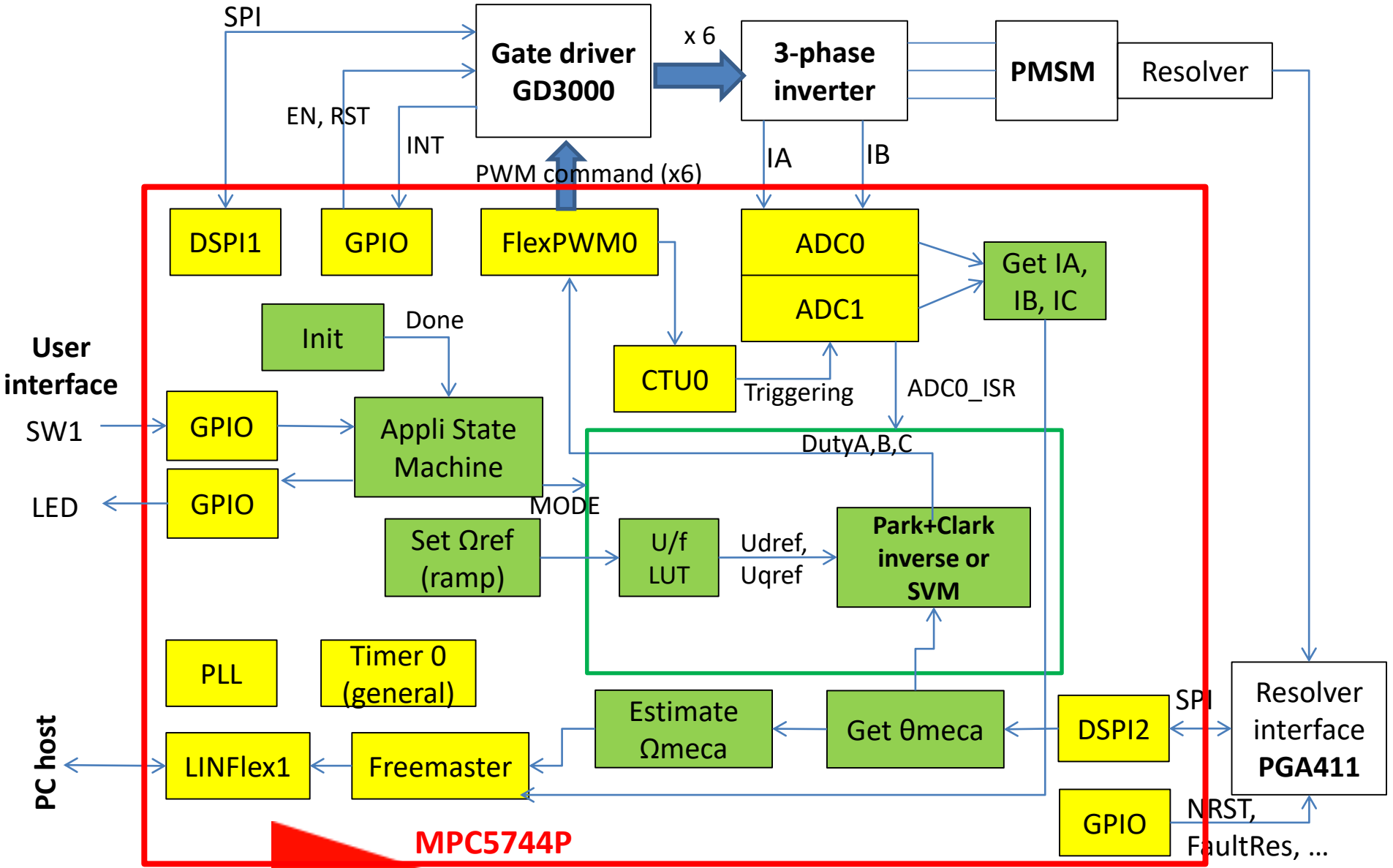
DEPARTEMENT OF ELECTRICAL AND COMPUTER ENGINEERING

INSA NATIONAL INSTITUTE OF APPLIED SCIENCES TOULOUSE
DEPARTMENT OF COMPUTER & ELECTRICAL ENGINEERING

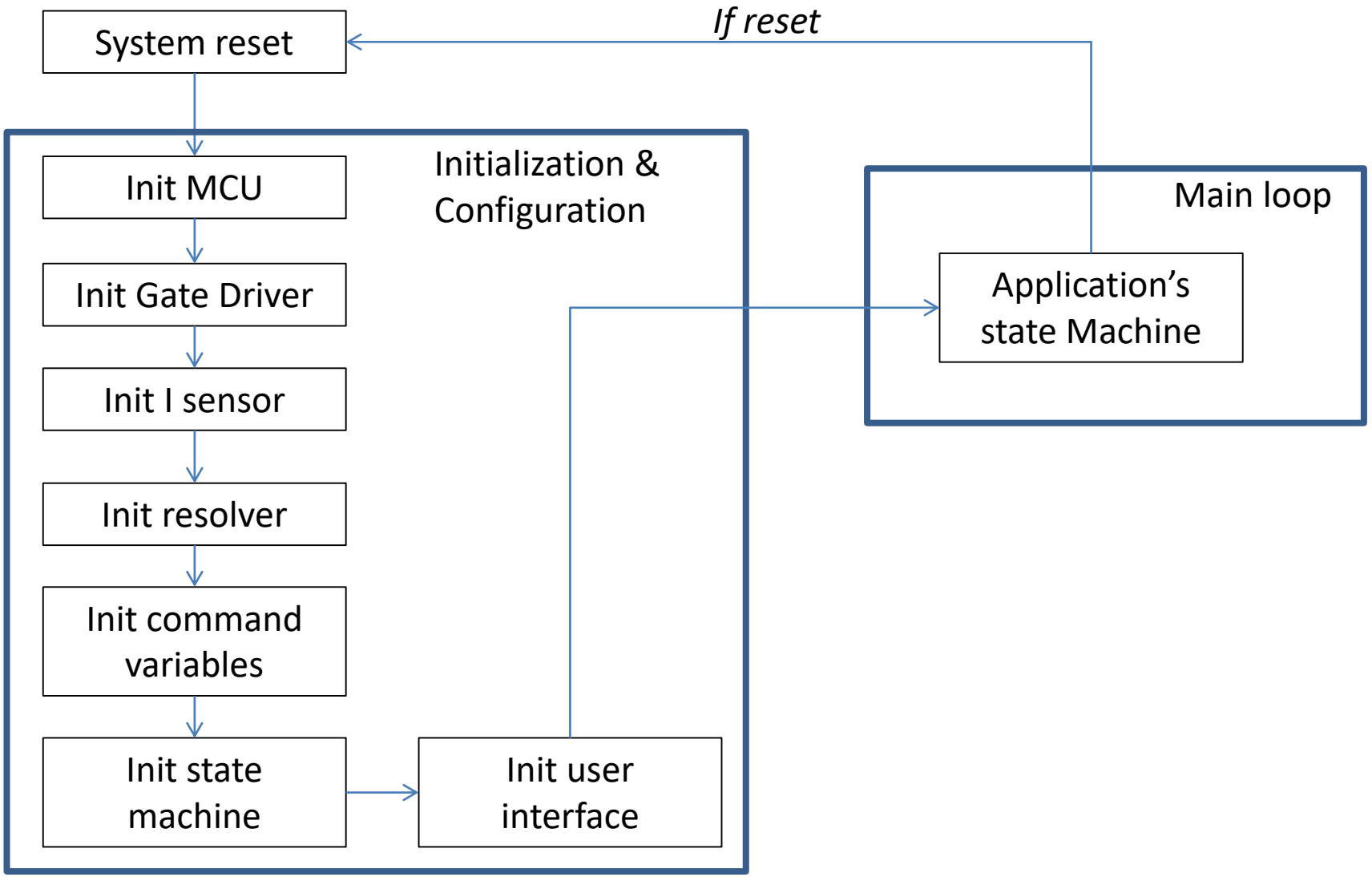
Brief description

- Purpose: demonstration of U/f command of a PMSM motor (TG Drive TGN2-0028-30-24), with MPC5744P MCU, GD3000 gate driver and PGA411 resolver interface.
- The program supports two modes: either the motor is simulated (for software debug purpose) or the motor is connected.
- Pushing switch1 activates or stops the motor rotation.
- The speed rotation increases according to a ramp profile (+0.25 RPM/ms).
- The command is based on a 120 µs PWM signal, which synchronizes the update of the next PWM duty cycle (fast loop) and speed command (slow loop).
- Current on phases A, B, C are acquired every 120 µs. Rotor speed and position are acquired every 1.20 ms. They can be transmitted to host PC by Freemaster for validation/instrumentation purpose.
- All the numbers are in 32-bit float format and mathematical operations are based on functions of AMMCLIB libraries.

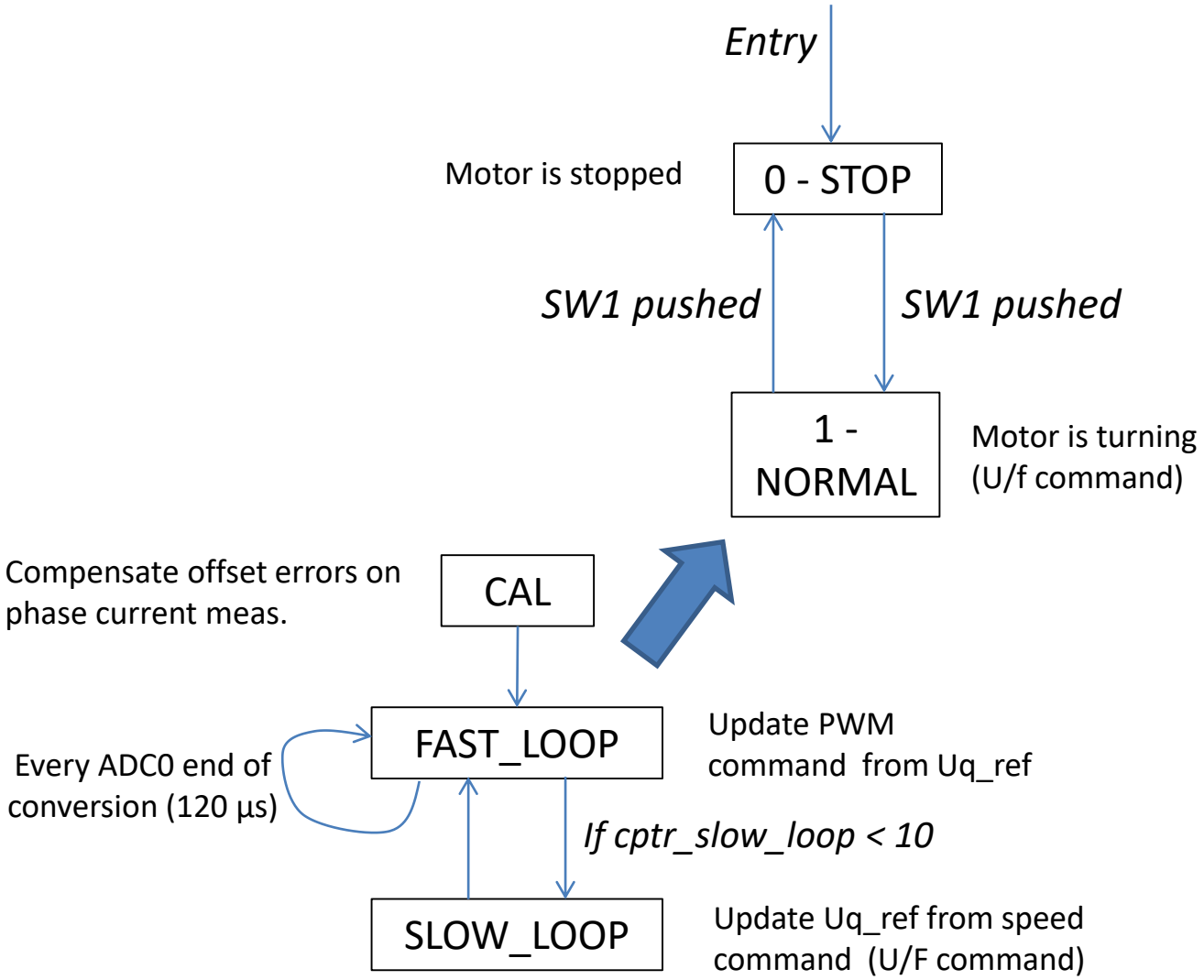
HW/SW block diagram



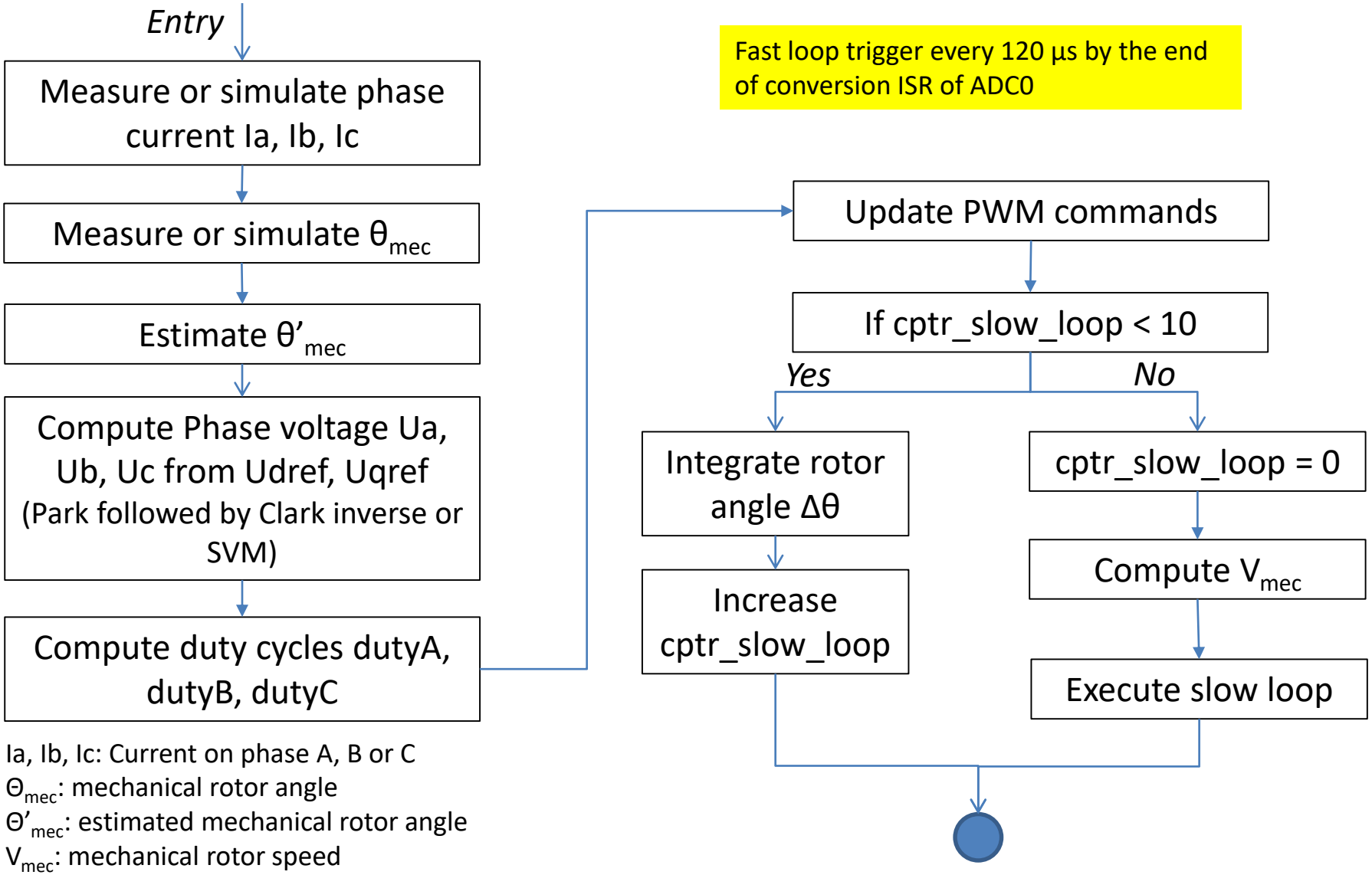
Software flow chart – Overall view



Software flow chart – Normal vs. stop mode

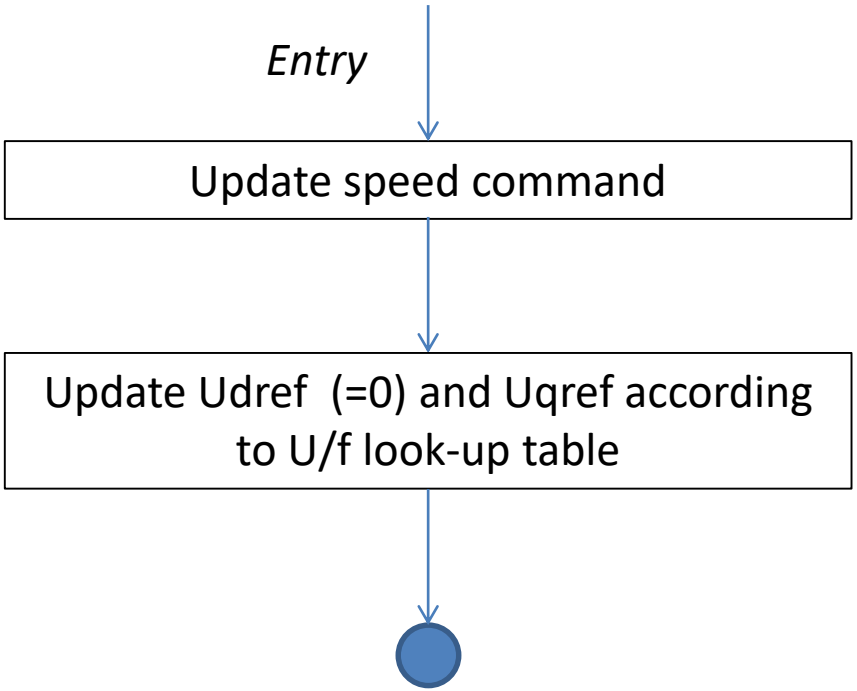


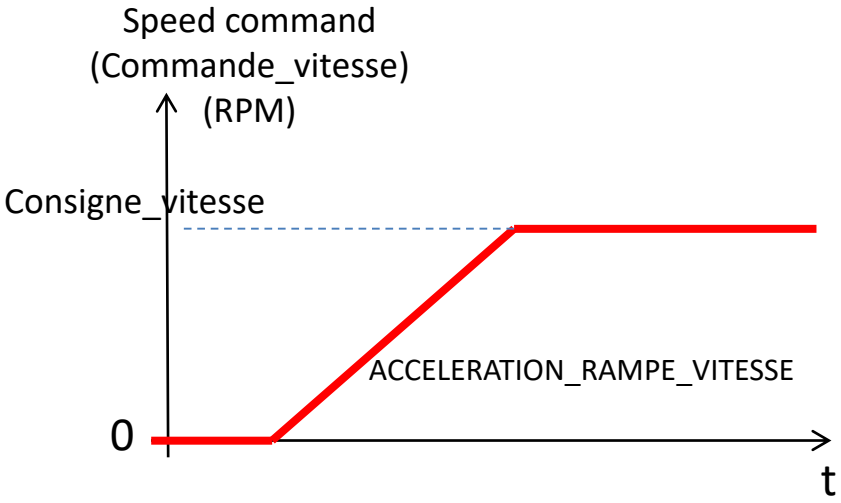
Software flow chart – NORMAL Mode – Fast loop



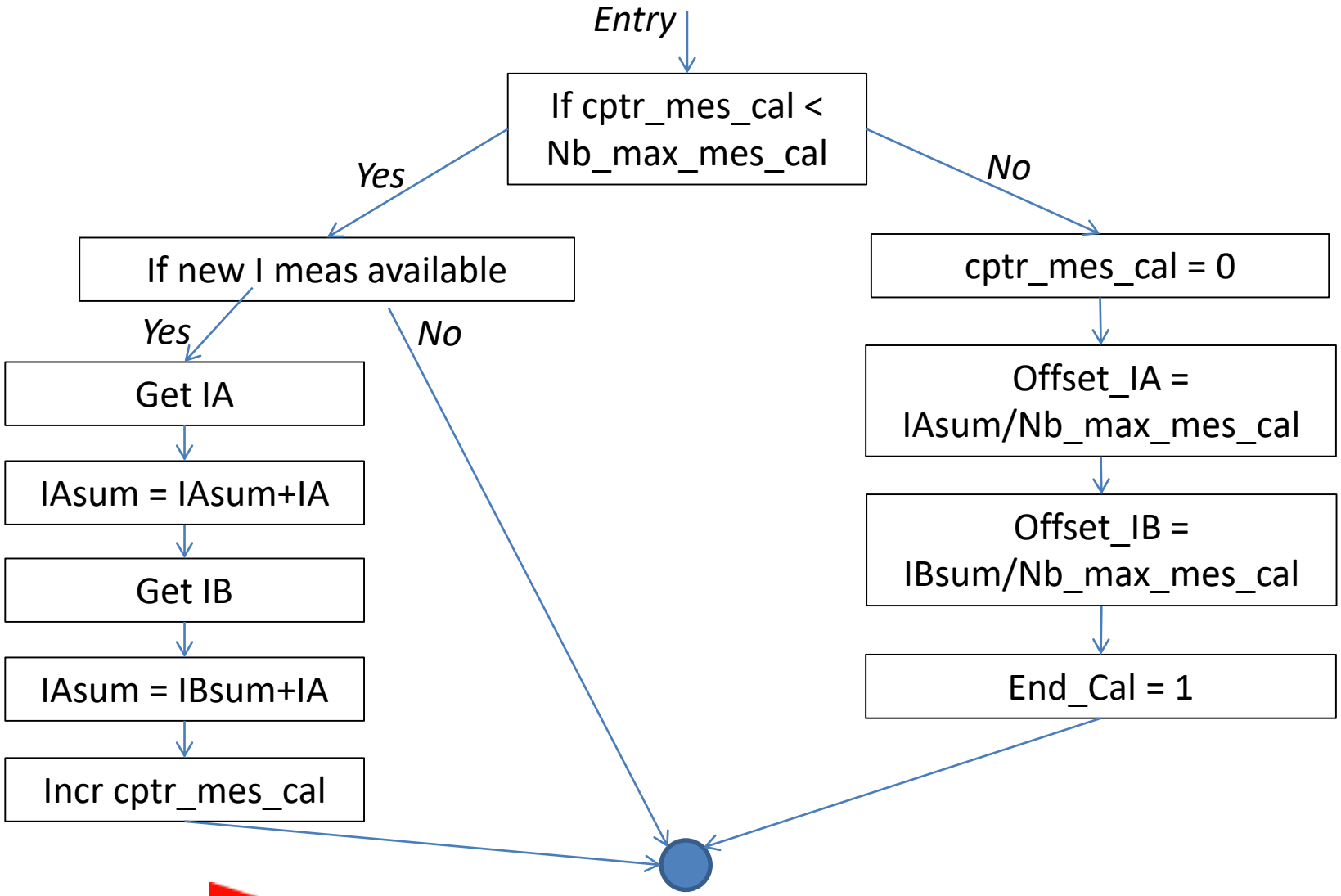
Ia, Ib, Ic: Current on phase A, B or C
Θ_{mec}: mechanical rotor angle
Θ'_{mec}: estimated mechanical rotor angle
V_{mec}: mechanical rotor speed
SVM = standard Space Vector Modulation

Software flow chart – Slow loop





Software flow chart – Mode CAL



Software architecture

Init & Config composants

State Machine

Application layer

Motor command (fast/slow loop, U/f…)

Motor simu. model (current, rotor angle and speed)

AMMCLIB

Middleware layer

Gate driver GD3000 (init, config, update PWM, speed, status, reset…)

Current sensor (init, config, get IA, IB, estimate IC, status, reset…)

User Interface (start/stop motor, get ref speed)

Resolveur PGA411 (init, config, get position, speed, status, reset…)

Mode entry

Clock config

GPIO

ADC

CTU

DSPI

LINFlex

Timer

Freemaster

FlexPWM

Driver layer

Software architecture

ClockConfig.c/.h	Initialization of clock buses and oscillators
Cmd_moteur.c/.h	Functions for the motor command (e.g. fast/slow loop execution, U/f look-up table, …)
Config_eDMA.c/.h	Configuration of DMA transfer
ConfigADC.c/.h	Configuration of analog-to-digital converters (for phase current acquisition)
ConfigCTU.c/.h	Configuration of Cross Triggering Unit (synchronization between PWM and ADC)
ConfigPIT.c/.h	Configuration of the Periodic Interrupt Timer
ConfigPWM.c/.h	Configuration of the phase PWM signals
Current_Sensor.c/.h	Configuration of current sensing chain and extraction of phase current
DSPI_Config.c.h	Configuration of SPI buses (MCU - gate driver and MCU – resolver interface
gd3000.c	Configuration of gate driver GD3000, activation/deactivation of inverter command, diagnosis
ModeEntryAndConfig.c/.h	Configuration of MCU mode entry
Modele_moteur_virtuel.c/.h	Simulation model of PMSM
pga411.c/.h	Configuration of resolver interface PGA411, acquisition of rotor angle and speed, diagnosis
UARTConfig.c/.h	UART configuration (to use Freemaster)
User_Interface.c/.h	Configuration of the user interface