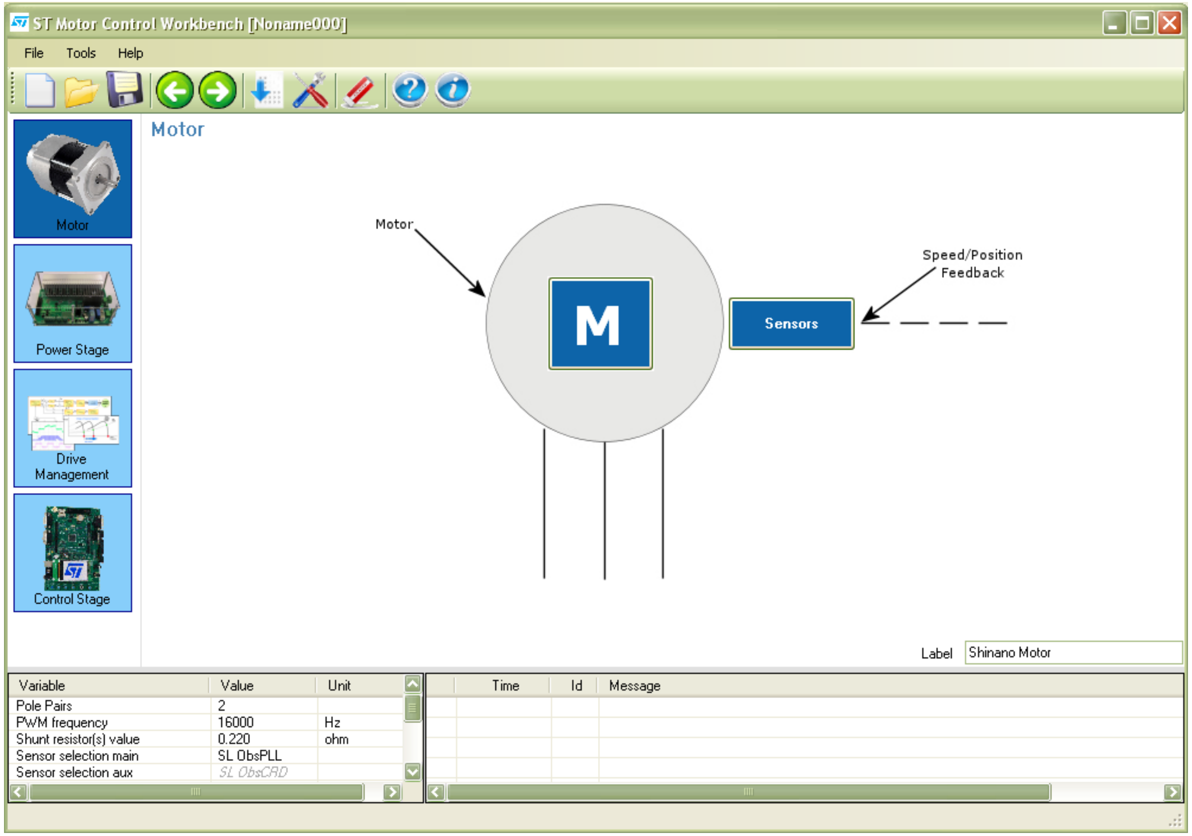
Agenda

- 2nd day – Morning
  - User interface
    - Serial Protocal
    - DAC
    - LCD
    - DAC customization
  - STM32 MC Workbench presentation in detail

- 2nd day – Afternoon
  - Quick Start
    - Config the firmware lib with Workbench
    - LCD User Interface
    - IAR IDE – MC Workspace
  - Practical hints in motor tuning:
    - Draw an arbitrary sensorless start-up waveform
    - Open loop feature
    - Faults generation
    - Motor start-up
  - Demo, Q&A

Config the firmware lib with MC WB – PMSM motor parameters

- It contains
  - Electrical motor parameters
  - Motor sensor parameters



Label: Shinano Motor

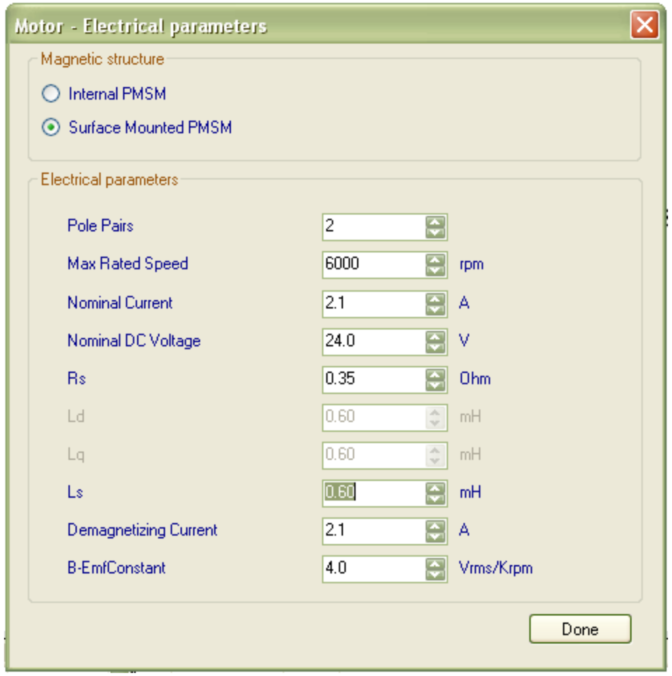
Variable	Value	Unit
Pole Pairs	2	
PWM frequency	16000	Hz
Shunt resistor(s) value	0.220	ohm
Sensor selection main	SL ObsPLL	
Sensor selection aux	*SL ObsCRD*	

Time	Id	Message

Config the firmware lib with MC WB –

PMSM motor parameters

- Select either Internal PMSM or Surface Mounted PMSM according to the magnetic structure of your motor
- If you don’t have this information you need to measure both Ld and Lq inductance for verifying it
  - If 2*(Lq-Ld)/(Ld+Lq) <15% motor may be considered a SM-PMSM



Motor - Electrical parameters

Magnetic structure

○ Internal PMSM

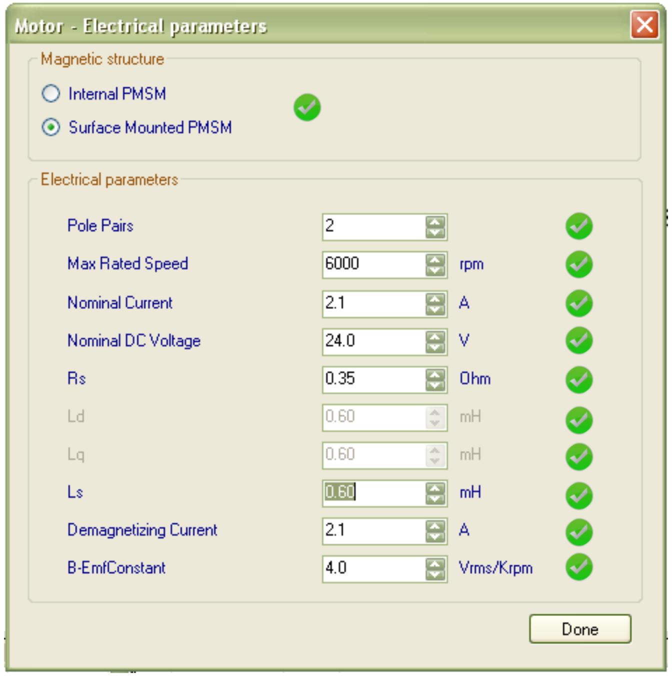
● Surface Mounted PMSM

Electrical parameters

Pole Pairs	2	
Max Rated Speed	6000	rpm
Nominal Current	2.1	A
Nominal DC Voltage	24.0	V
Rs	0.35	Ohm
Ld	0.60	mH
Lq	0.60	mH
Ls	0.60	mH
Demagnetizing Current	2.1	A
B-EmfConstant	4.0	Vrms/Krpm

Config the firmware lib with MC WB – PMSM motor parameters

- Max rated speed (rpm)
  - Should be provided by motor producer (if not, set it to max application speed)
  - Maximum motor rated speed above which motor can get damaged
  - Maximum application speed must be lower than this value
- Nominal current (in A, 0-to-peak)
  - Motor rated current, must be provided by motor producer
  - It will be used to limit the imposed motor phase current during normal operation
- Nominal DC voltage
  - Nominal DC bus voltage from which the motor should run, must be provided by motor producer
- Demagnetizing current
  - Rotor demagnetizing current, may be provided by motor producer (if not use default value, i.e. motor nominal current)
  - Used to limit the amount of target negative Id during flux weakening



Motor - Electrical parameters

Magnetic structure

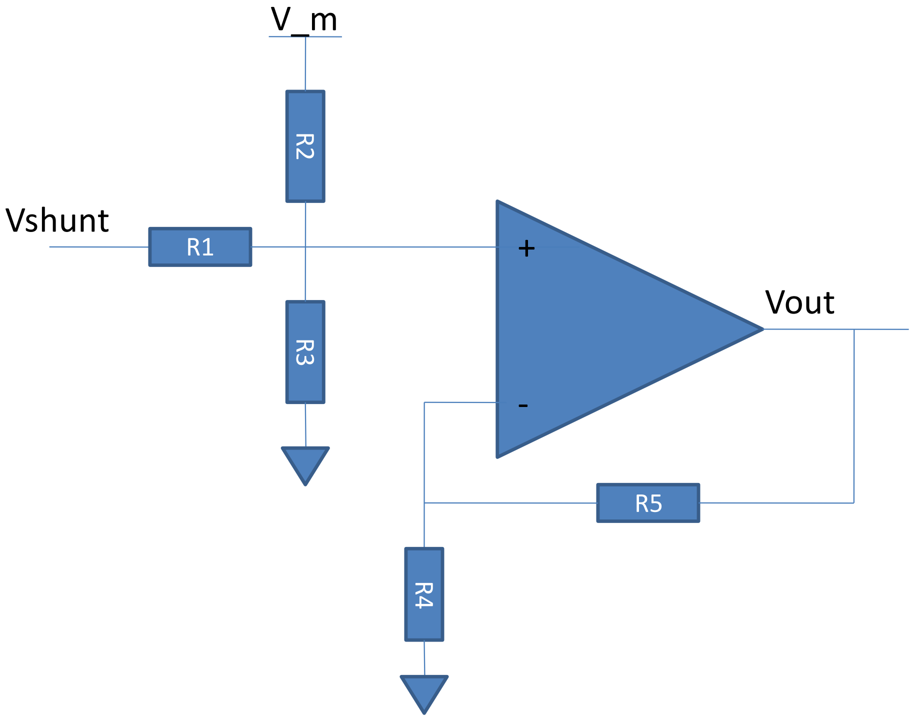
○ Internal PMSM

● Surface Mounted PMSM ✓

Electrical parameters

Pole Pairs	2		✓
Max Rated Speed	6000	rpm	✓
Nominal Current	2.1	A	✓
Nominal DC Voltage	24.0	V	✓
Rs	0.35	Ohm	✓
Ld	0.60	mH	✓
Lq	0.60	mH	✓
Ls	0.60	mH	✓
Demagnetizing Current	2.1	A	✓
B-EmfConstant	4.0	Vrms/Krpm	✓

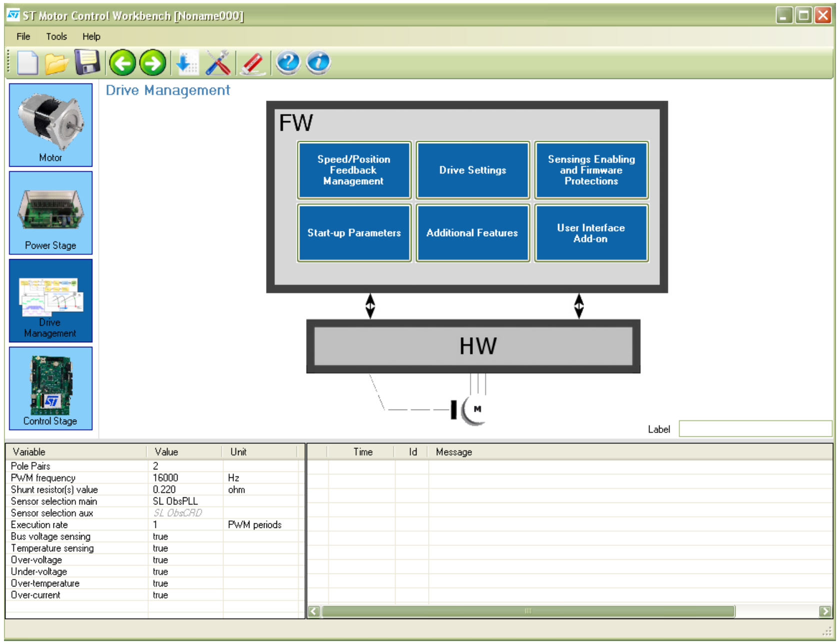
Config the firmware lib with MC WB – Operational amplifier gain network



$$A_V\ gain = \left(1 + R5/R4\right)\cdot\frac{\left(R2 // R3\right)}{\left(\left(R2 // R3\right)+R1\right)}$$

Config the firmware lib with MC WB – How to configure drive parameters

- The list of initial setting should be as follow (leave default values if not here differently specified):
  - In Speed/position feedback management, select the main speed sensor to be used and enter the maximum speed that will be allowed by the drive (e.g. you may enter maximum application speed)
  - In Drive settings, if working with value line, set PWM frequency (F_{PWM}) and Torque and flux regulator execution rate (n) so that $F_{FOC} = F_{PWM} / n$ <= 10kHz (limit is higher when working with real sensors)
  - In Drive settings, if the final goal is to work with two motors, set PWM frequency (F_{PWM}) and Torque and flux regulator execution rate (n) so that $F_{FOC} = F_{PWM} / n$ <= 16kHz (limit is higher when working with real sensors)
  - In Drive settings, decrease execution rate of torque and flux regulator down to 2000 rad/s if power stage → current reading topology is single shunt
  - In Sensing enabling and FW protections, uncheck those sensing not supported by power stage
  - If in sensor-less, in start-up parameters it’s required to enter the rev-up settings (use basic profile at the beginning of your development)
  - If in sensor-less, in Drive settings, initially set default target speed to at least 20% of maximum application speed



Variable	Value	Unit
Pole Pairs	2	
PWM frequency	16000	Hz
Shunt resistor(s) value	0.220	ohm
Sensor selection main	SL ObsPLL	
Sensor selection aux	SL ObsCRD	
Execution rate	1	PWM periods
Bus voltage sensing	true	
Temperature sensing	true	
Over-voltage	true	
Under-voltage	true	
Over-temperature	true	
Over-current	true	

Time	Id	Message

- In additional feature, start without any additional method (in case add them later)

life.augmented

Config the firmware lib with MC WB – ST control stages parameters

	STM3210B-EVAL	STM3210E-EVAL	STM32100B-EVAL	STEVAL-IHM022v1
MCU and clock selection				
MCU selection	Performance line medium density	Performance line high density	Value line medium density	Performance line high density
CPU frequency (MHz)	72	72	24	72
Supply voltage (V)	3.3	3.3	3.3	3.3
Analog inputs motor 1				
ADC Channel selection for phase U	ADC12_11	ADC12_11	-	ADC123_IN11
ADC Channel selection for phase V	ADC12_12	ADC12_12	-	ADC123_IN12
ADC Channel selection for phase W	ADC12_13	ADC12_13	-	ADC123_IN13
ADC Channel for current reading	ADC12_12	ADC12_12	ADC1_12	ADC123_IN12
Bus voltage feedback ADC channel	ADC12_3	ADC12_IN10	ADC1_3	ADC123_IN3
Temperature feedback ADC Channel	ADC12_IN10	ADC12_IN15	ADC1_10	-
Analog inputs motor 2				
ADC Channel selection for phase U	-	-	-	ADC123_IN10
ADC Channel selection for phase V	-	-	-	ADC12_IN14 (J2 in 3-5, 4-6 position) or ADC3_IN6 (J2 in 1-3, 2-4 position)
ADC Channel selection for phase W	-	-	-	ADC123_IN15
ADC Channel for current reading	-	-	-	ADC12_IN14 (J2 in 3-5, 4-6 position) or ADC3_IN6 (J2 in 1-3, 2-4 position)
Bus voltage feedback ADC channel	-	-	-	ADC12_IN14 (J2 in 1-3, 2-4 position)
Temperature feedback ADC Channel	-	-	-	-

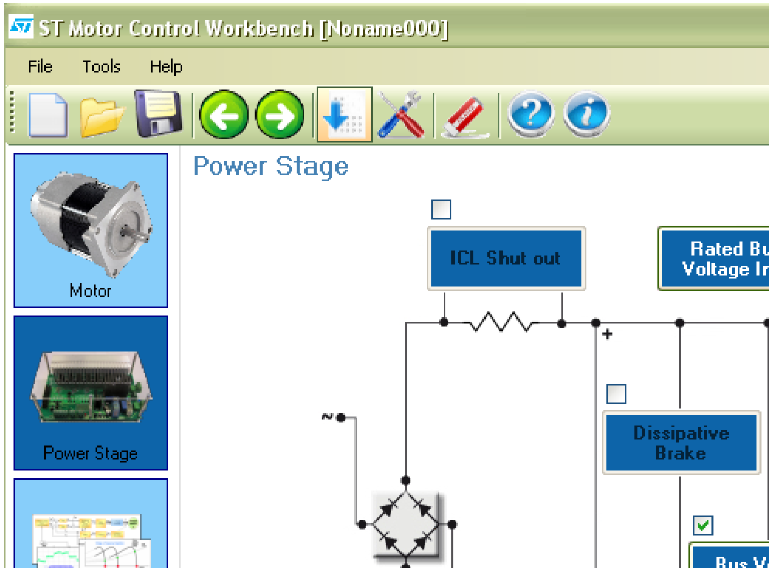
Config the firmware lib with MC WB –

ST control stages parameters

	STM3210B-EVAL	STM3210E-EVAL	STM32100B-EVAL	STEVAL-IHM022v1
Digital I/O-				Main connector
Inverter driving signals timer	TIM1	TIM8 *	TIM1	TIM1
TIM1 remapping	Full re-map	-	Full re-map	Full re-map
Encoder interface timer	TIM2	TIM2	TIM2	TIM2
Encoder timer remap	No re-map	No re-map (check JP4)	No re-map	No re-map (check jumpers)
Hall sensor interface timer	TIM2	TIM2	TIM2	TIM2
Timer remap	No re-map	No re-map (check JP4)	No re-map	No re-map (check jumpers)
Dissipative brake output port, pin	D,13	A, 3 (check R2)	D,13	B, 8
Inrush current limiter	D,10	B, 12	D,10	D,10
Over-current protection disabling	D,13	A, 3 (check R2)	D,13	B, 8
USART1 remapping	No re-map	No re-map	No re-map	No re-map
Digital I/O motor 2				Sub1 connector
Inverter driving signals timer	-	-	-	TIM8
TIM1 remapping	-	-	-	-
Encoder interface timer	-	-	-	TIM4 (check jumpers)
Encoder timer remap	-	-	-	Re-mapped
Hall sensor interface timer	-	-	-	TIM4 (check jumpers)
Timer remap	-	-	-	Re-mapped
Dissipative brake output port, pin	-	-	-	B, 9
Inrush current limiter	-	-	-	D, 11
Over-current protection disabling	-	-	-	B, 9

Config the firmware lib with MC WB – Parameters generation

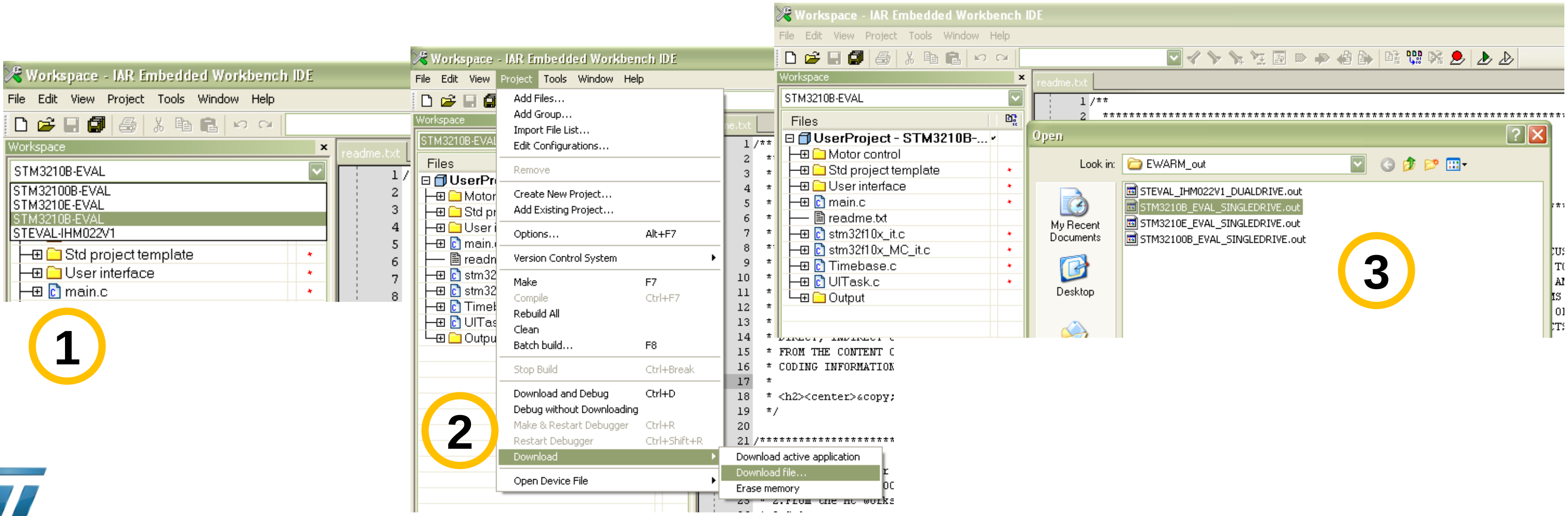
- Once all the required parameters have been filled in the ST MC Workbench, click on the ‘Generation’ button and select the output path (*Your working folder*\System & Drive Params)



LCD User Interface –

LCD User Interface download

- First time - just first time - before downloading the customized STM32 FOC library v3.0 in a supported ST evalboard, the LCD UI shall be downloaded to the microcontroller.
- Binaries are located in: $InstallationFolder\LCD Project\EWARM_out
- Use EWARM to download binary file
- Open workspace, select User Project configuration and binary file according to your Eval Board



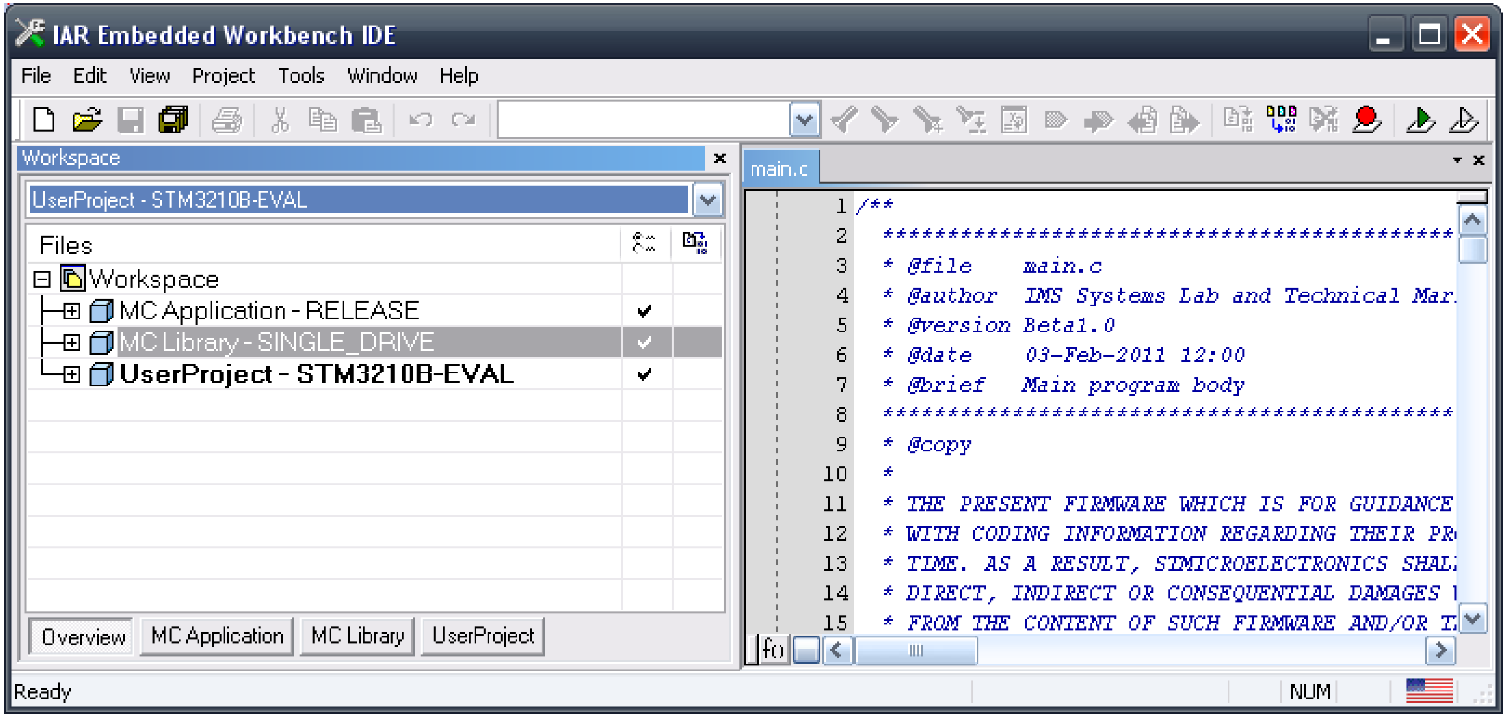
Available workspaces – IAR IDE

- The IAR workspace is located in:

  $InstallationFolder\Project\EWARM\Workspace.eww

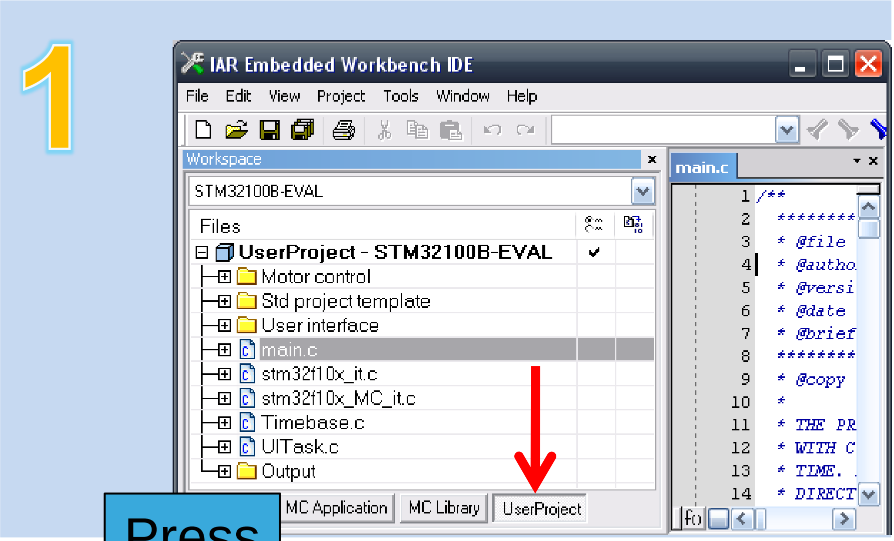
- Alternative choice, if using STM32F103xx, is to open the FreeRTOS workspace:

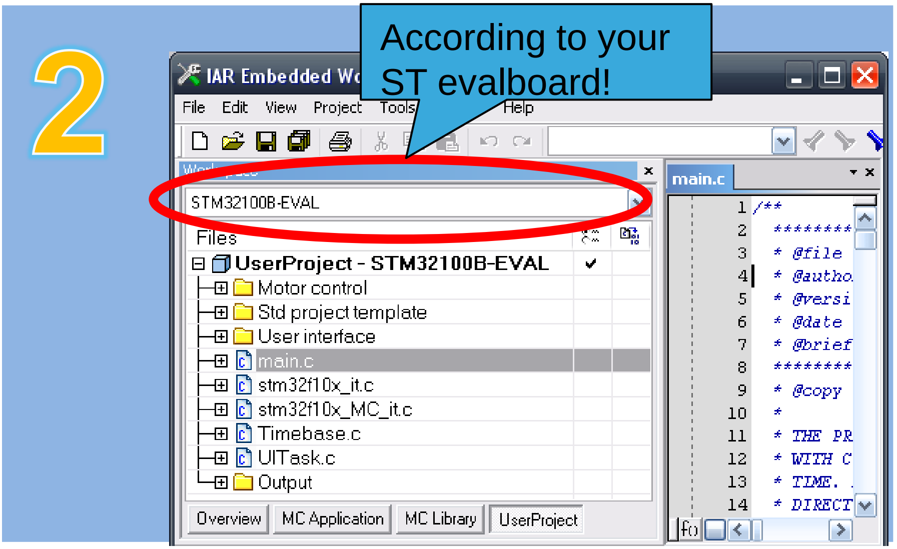
  $InstallationFolder\FreeRTOS\Project\EWARM\RTOS_Workspace.eww


IAR Embedded Workbench IDE
File
Edit
View
Project
Tools
Window
Help
Workspace
UserProject - STM3210B-EVAL
Files
Workspace
MC Application - RELEASE ✓
MC Library - SINGLE_DRIVE ✓
UserProject - STM3210B-EVAL ✓
Overview
MC Application
MC Library
UserProject
main.c
f()
Ready
NUM

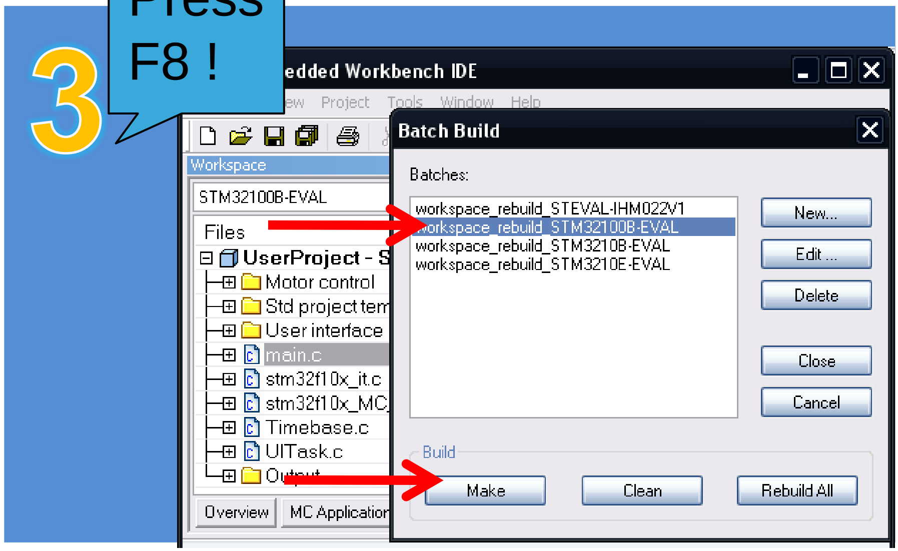
```
/**
  ********************************************
  * @file    main.c
  * @author  IMS Systems Lab and Technical Mar
  * @version Beta1.0
  * @date    03-Feb-2011 12:00
  * @brief   Main program body
  ********************************************
  * @copy
  *
  * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE
  * WITH CODING INFORMATION REGARDING THEIR PR
  * TIME. AS A RESULT, STMICROELECTRONICS SHAL
  * DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES 
  * FROM THE CONTENT OF SUCH FIRMWARE AND/OR T
```

Motor control workspace download



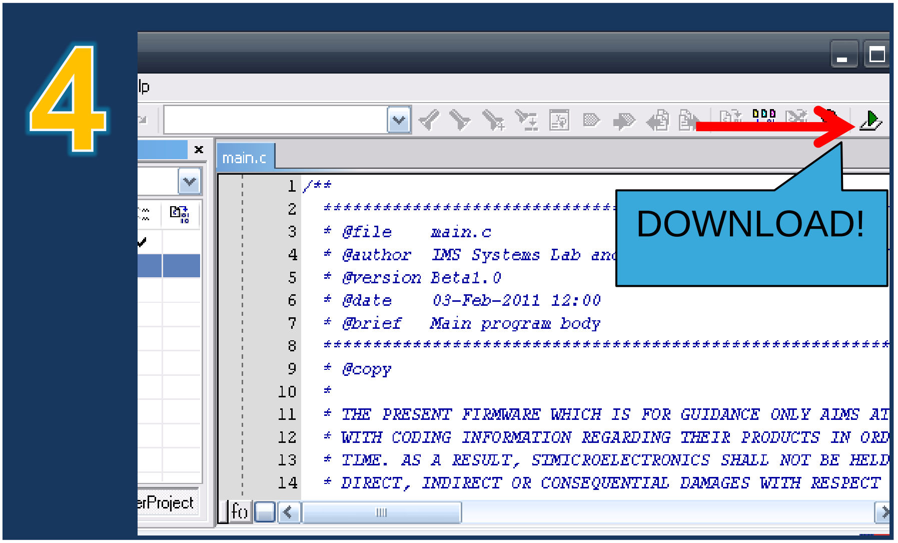


According to your ST evalboard!



Press F8 !

Batches:
workspace_rebuild_STEVAL-IHM022V1
workspace_rebuild_STM32100B-EVAL
workspace_rebuild_STM3210B-EVAL
workspace_rebuild_STM3210E-EVAL



```
/**
  ******************************
  * @file    main.c
  * @author  IMS Systems Lab and
  * @version Beta1.0
  * @date    03-Feb-2011 12:00
  * @brief   Main program body
  ******************************************************
  * @copy
  *
  * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT
  * WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORD
  * TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD
  * DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT
```

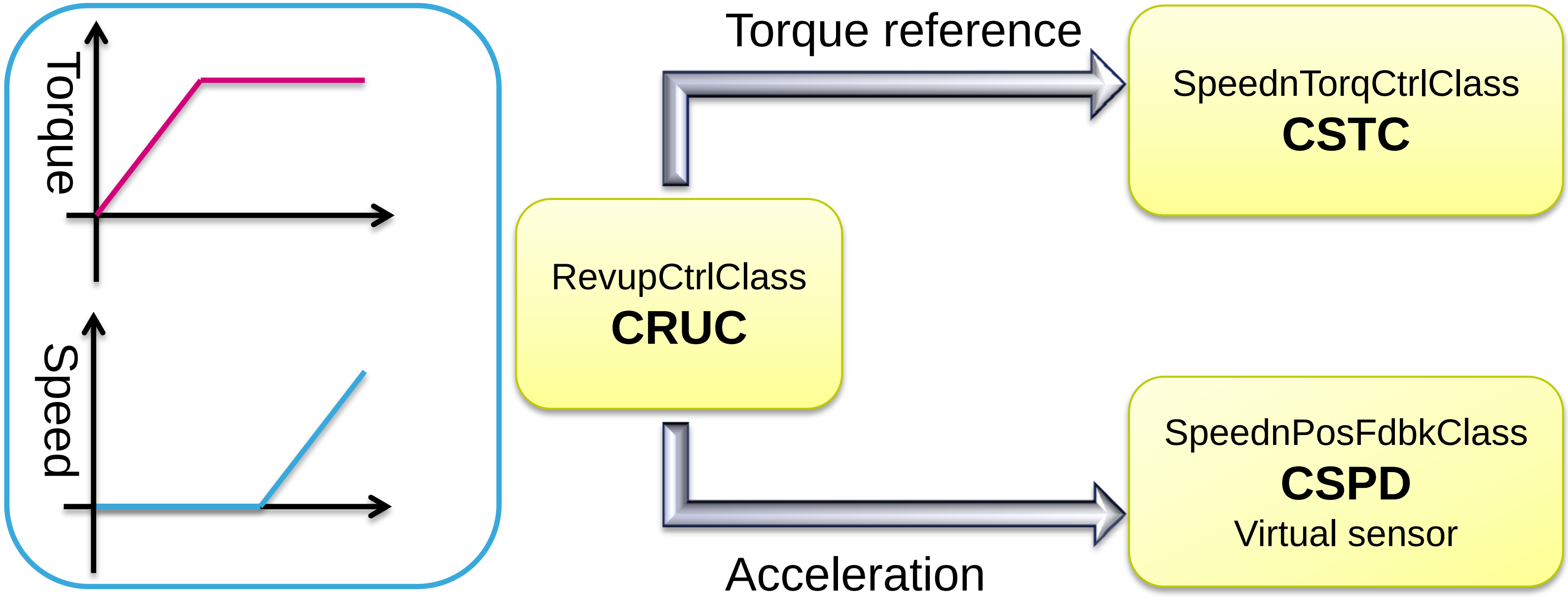
DOWNLOAD!

Agenda

- 2nd day – Morning
  - User interface
    - Serial Protocal
    - DAC
    - LCD
    - DAC customization
  - STM32 MC Workbench presentation in detail
- 2nd day – Afternoon
  - Quick Start
    - Config the firmware lib with Workbench
    - LCD User Interface
    - IAR IDE – MC Workspace
  - Practical hints in motor tuning:
    - Draw an arbitrary sensorless start-up waveform
    - Open loop feature
    - Faults generation
    - Motor start-up
  - Demo, Q&A

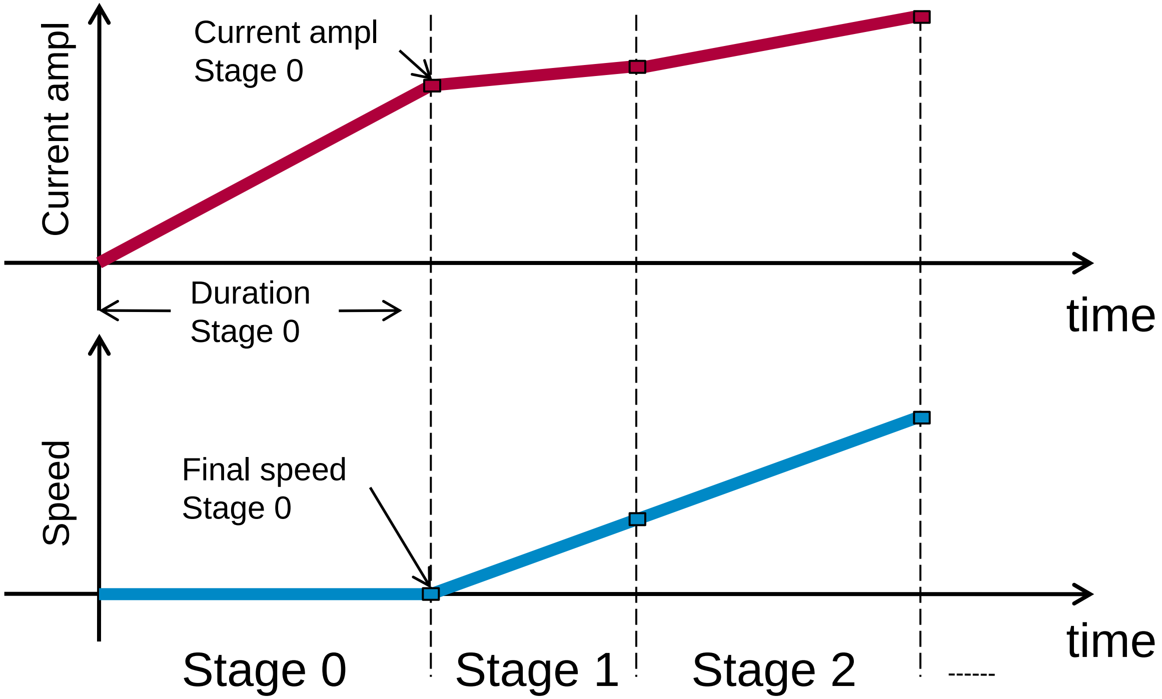
Draw an arbitrary sensor-less start-up waveform

- Rev-up controller is used to program a sensorless startup.
- It acts during the “START” state and send commands to the speed and torque controller and to the virtual speed sensor to execute the programmed sequence.



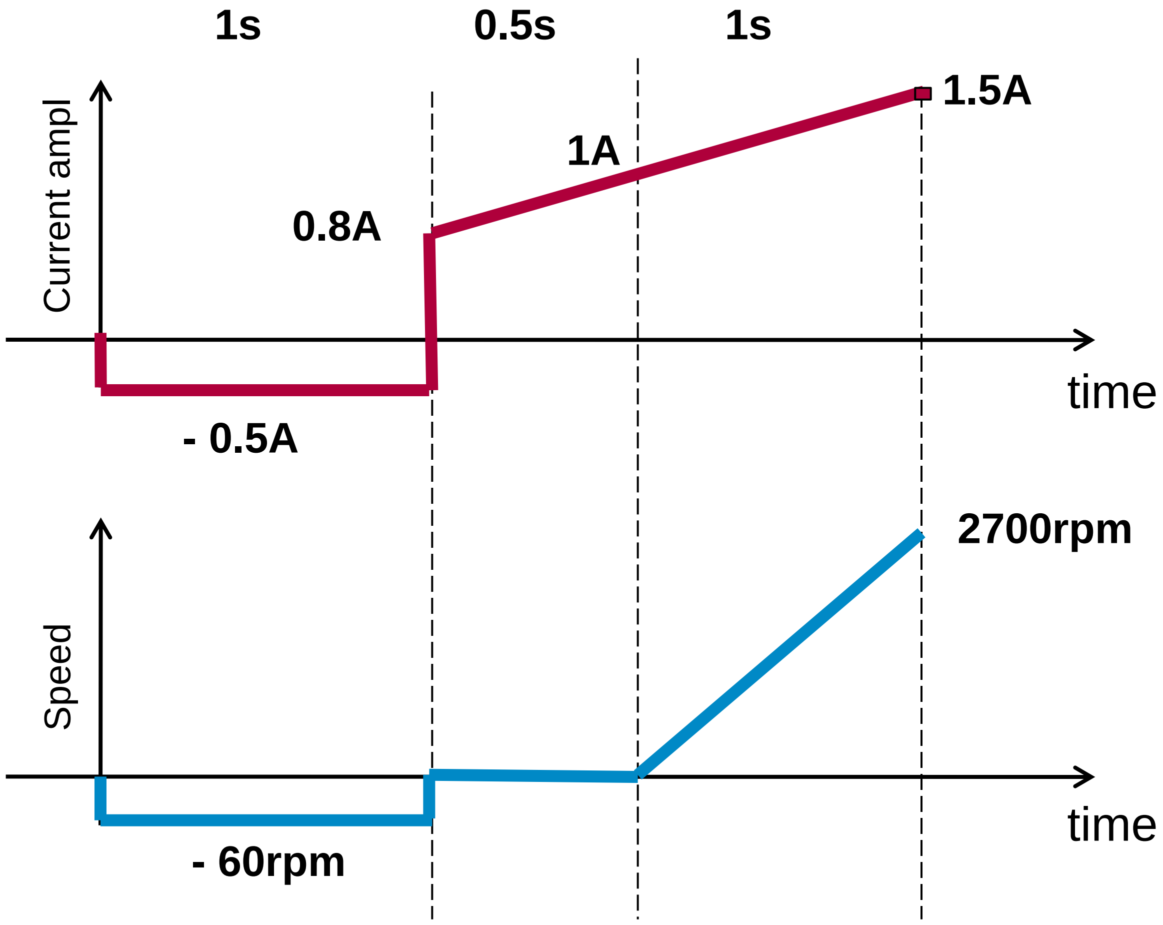
Rev-up controller stages

- The programmed rev-up sequence is composed by a number of stages.
- For each stage is possible to define the duration, the final torque reference and the final speed of the virtual sensor.
- It is possible also to define the starting electrical angle.



Rev-up controller stages

- Another case



How to customize the sensor-less start-up procedure: first settings

- After having customized motor, power stage and control stage parameters, one should customize *Drive management* → *start-up parameters* as follow:
  - Set *current ramp initial and final values* equal to motor *nominal current* value / 2
  - Set *speed ramp final value* to around 30% of *maximum application speed*
  - According to motor inertia it may be required to increase the *speed ramp duration*
  - Set *minimum start-up output speed* to 15% of maximum application speed (if required, decreased it later)
  - Set estimated speed band tolerance lower limit to 93.75%
  - Enable the alignment at the beginning of your development (*duration* 2000ms, *final current ramp value* from 0.5 to 1 times motor nominal current according to load)

open loop tuning

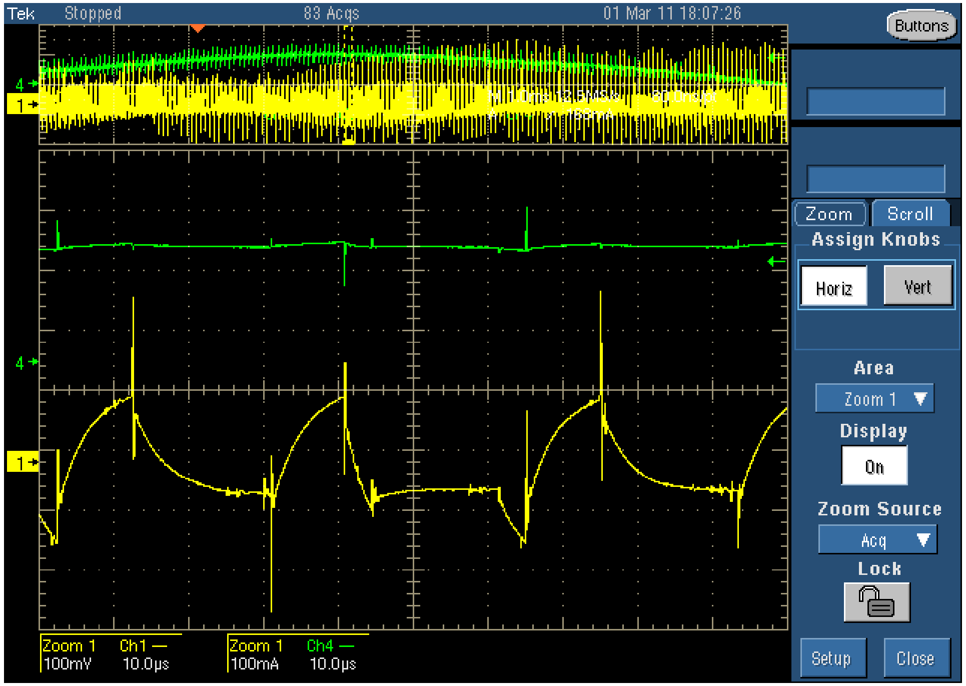
- Useful feature for HW debugging(current reading parameters)

- Implemented by exploiting sensor-less rev-up algorithm
  - It's not supported by Workbench. So open loop tunging is enabled manually:
    - Motor1 – in Drive parameters.h:
      - `#define  OPEN_LOOP_FOC              ENABLE`
    - Motor 2 – in Drive parameters2.h
      - `#define  OPEN_LOOP_FOC2             ENABLE`
  - A special Rev-up stage is programmed:
    - Motor1 – in Parameters conversion_F10X.h
    - Motor2 – in Parameters conversion_F10X motor 2.h

    So that the total duration is about several minutes(during that stage Vd=0 and Vq= fixed value which is defined in "parameters.h/parameters2.h")
  - After 'Start motor' command, a three-phase voltage with programmable frequency and amplitude will be generated
  - A rev-up failure will be then generated
  - If needed, ack the error and re-start the motor

HW validation 1/2

- To validate the HW, it’s convenient to apply a three-phase voltage with a given speed and check for correctness of current signal feedback
- ‘Open loop’ configuration is intended to be used in this case
- Below capture shows the single-shunt based current feedback (on MCU pin) for the examined HW



Tek Stopped 83 Acqs 01 Mar 11 18:07:26

Zoom 1 Ch1 — 100mV 10.0µs

Zoom 1 Ch4 — 100mA 10.0µs

Area: Zoom 1

Display: On

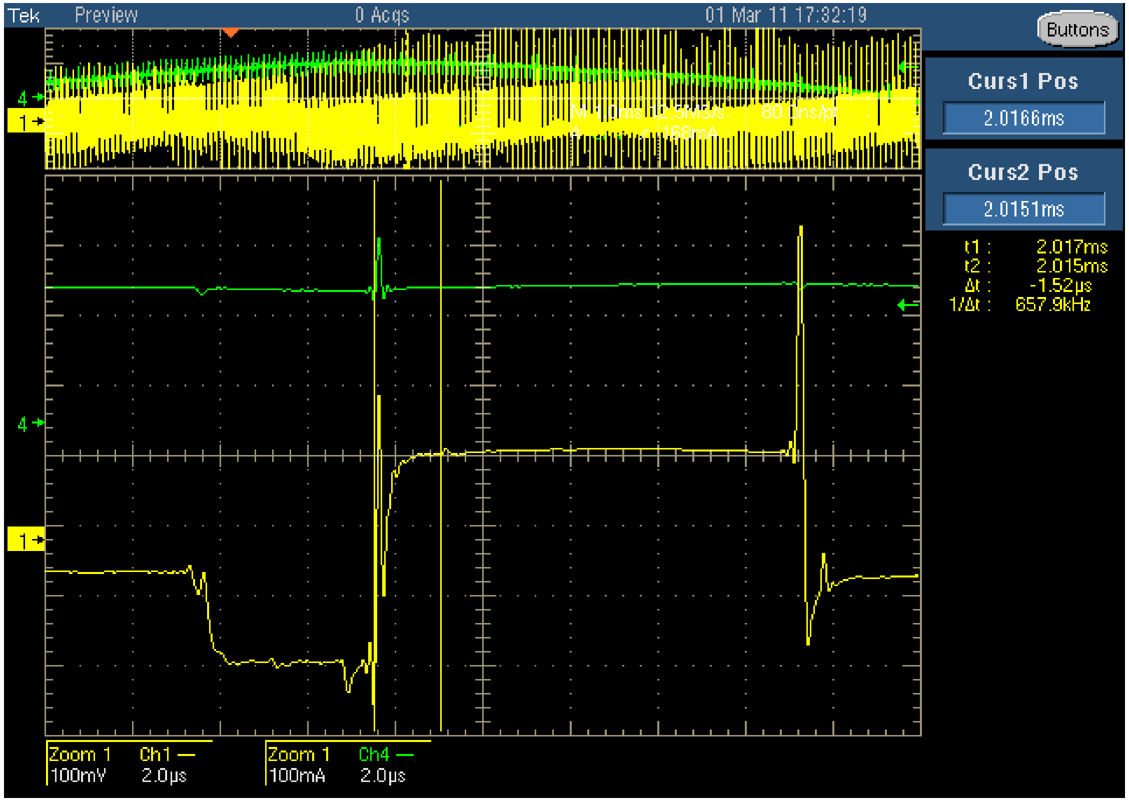
Zoom Source: Acq

- This is clearly over-filtered! (R*C= 3.7µs)

HW validation 2/2

- This is how it should have looked like (no filters)….

- Trise ≈ 1.5µs



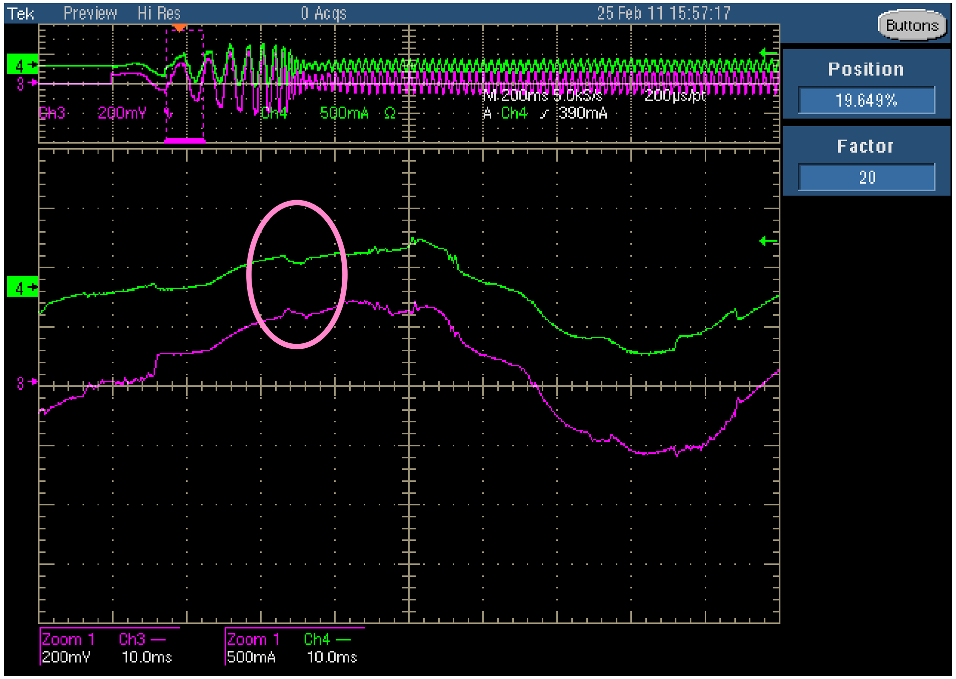
Curs1 Pos
2.0166ms

Curs2 Pos
2.0151ms

```
t1 :    2.017ms
t2 :    2.015ms
Δt :    -1.52µs
1/Δt :  657.9kHz
```

Current measurement errors

- An error in the read current often results in a distortion of the real current with opposite sign



Tek Preview Hi Res 0 Acqs 25 Feb 11 15:57:17

Ch3 200mV Ch4 500mA Ω

M 200ms 5.0kS/s 200µs/pt

A Ch4 ⁄ 390mA

Position 19.649%

Factor 20

Zoom 1 Ch3 — 200mV 10.0ms

Zoom 1 Ch4 — 500mA 10.0ms

Real current (probe)

Read current (DAC)

- If the error is deterministic (always at the same point) it is probably due to the algorithm
- If it is not, some noise is entering the feedback gain (Tnoise?, VIPer,…)

faults generation

- Possible source of faults:
  - Under-voltage
  - Over-voltage
    - Different actions can be taken on its occurrence
  - Over-temperature
  - Rev-up failure
  - Speed feedback
  - SW error
    - FOC duration too long for the selected PWM frequency
    - A unallowed switch has been requested to the state machine

How to customize the sensor-less start-up procedure: troubleshooting 1/4

- First of all, double check consistency of parameters inserted in ST MC Workbench with special reference to motor electrical parameters, current sensing and bus voltage sensing
  - If any of the parameters is notably incorrect, motor will never starts!
  - If necessary, run the motor in open loop and measure Tnoise and Trise

- Problem: ‘Over-current’ fault message appears and the motor do not even try to start
  - 1st possible source: wrong current sensing topology has been selected in *power stage → current sensing*
    - Solution: select right current sensing configuration
  - 2nd possible source: wrong current sensing parameters
    - Solution: check power stage parameters which have been correctly inserted
  - 3rd possible source: current regulation loop bandwidth is too high for this HW
    - Solution: in *drive parameters → drive settings* decrease current regulation bandwidth (normally down to 2000 rad/sec for 3shunt topology and 1000 rad/s for single shunt topology)

How to customize the sensor-less start-up procedure: troubleshooting 2/4

- Problem: ‘SW error’ fault message appears and the motor do not even try to start
  - Source: the FOC execution rate is too high and computation can not be ended in time
  - Solution: In *Drive settings*, decrease ratio between *PWM frequency* and *Torque and flux regulator execution rate* (e.g. increasing Torque and flux regulator execution rate by one)

- Problem: Motor initially moves but then doesn’t rev-up, then fault message ‘Rev-up failure’ appears
  - Source: typically this happens cause the current provided to the motor is not enough for making it accelerate so fast
    - 1st possible solution: decrease acceleration rate by increasing *Start-up parameters → speed ramp duration* (being *Start-up parameters → speed ramp final value* set to about 30% of maximum application speed)
    - 2nd possible solution: increase start-up current by increasing *current ramp initial* and *final* values up to *motor → nominal current*
    - Enabling ‘Alignment phase’ (at least at the beginning of the development) makes start-up more deterministic, use around 2000ms, half of nominal current as first settings

How to customize the sensor-less start-up procedure: troubleshooting 3/4

- Problem: The rotor moves and accelerate following the ramp-up profile but then it stops and the fault message ‘Rev-up failure’ appears (a mix of following problem sources can be occurring):
  - 1st possible source: Observer gain G2 is too high and this makes speed reconstruction a bit noisy (never recognized as reliable). A mix of following solutions could be required:
    - 1st possible solution: decrease observer gain G2 by successive steps: /2, /4, /6, /8
    - 2nd possible solution: Enlarge Drive parameters→ *Speed/position feedback management* → *variance threshold* so as to make rotor locked check less ‘demanding’. (up to 80% for PLL and 400% for CORDIC)
  - 2nd possible source: the “window” where the reliability of the estimation is checked is too small
    - 1st possible solution: increase *speed ramp final* value to around 40% of maximum application speed
    - 2nd possible solution: decrease *minimum start-up output speed* to 10% of maximum application speed

How to customize the sensor-less start-up procedure: troubleshooting 4/4

- Problem: The rotor moves and accelerate following the ramp-up profile but then it stops and the fault message ‘Speed feedback’ appears
  - Use speed ramps: having a target speed gently going from the start-up output speed to the final target will avoid abrupt variations of torque demand that could spoil B-emf estimation
  - A mix of following problem sources can be occurring:
  - 1st possible source: Observer gain G2 is too high and this makes speed reconstruction a bit noisy (for the selected speed PI gains). A mix of following solutions could be required:
    - 1st possible solution: decrease observer gain G2 by successive steps: /2, /4, /6, /8
    - 2nd possible solution: Run motor in torque mode, if trouble doesn’t exist in torque mode, it means speed regulator gains are not optimal try changing them
  - 2nd possible source: frequent situation when the start-up has been validated too early
    - Solution: Try increasing *Start-up parameters* → *consecutive successful start-up output test* (normally to not more than 4-5), being *minimum start-up output speed* set to 15% of maximum application speed (if required, decreased it later)

- Problem: motor runs but current are not sinusoidal at all
  - 1st possible source: speed PI gains are not good
    - Solution: decrease Kp gain (and act on Ki evaluating speed regulation over/under shooting during transients)

Demo / Q&A

- Demo

- Q&A

Thank You !

www.st.com/MCU
www.st.com/stm8
www.st.com/stm32
www.st.com/stm32f1(f0,f2,f3,f4)

life.augmented