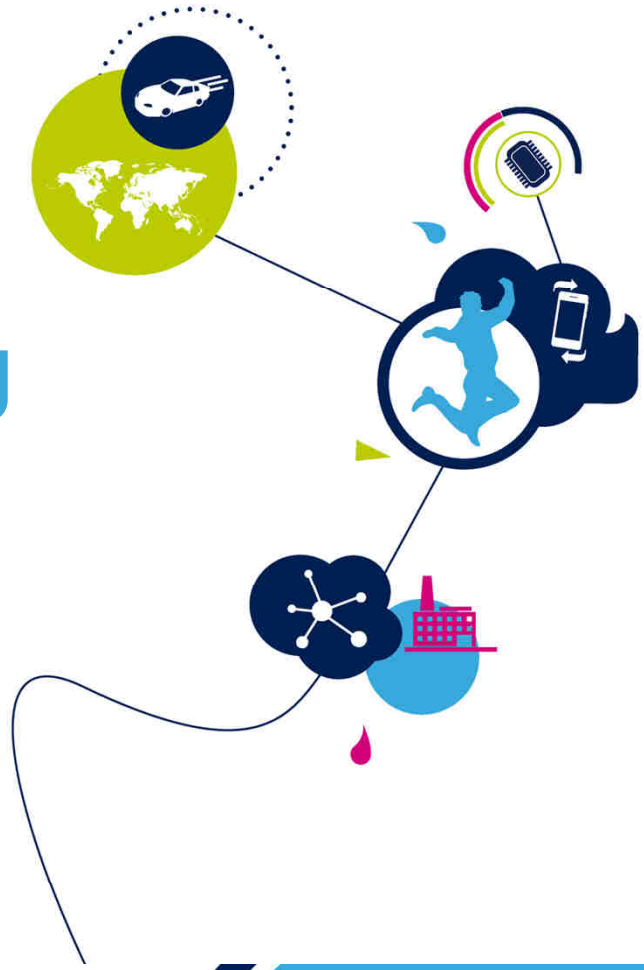
STM32G4 Mixed Signal MCU

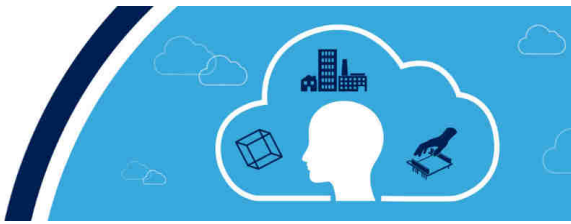
Hands-On Workshop

Karthik Chidambaranathan
Jason Stover
Bruno Montanari
The STMicroelectronics Team



life.augmented

Technology Tour 2019
Boston, MA | November 5



Lab Requirements

You need to have administration rights on your Laptop Computer to be able to install drivers and software and also to do the workshop.

System & Software requirements for Laptop Computer:

- Windows® 7 and later or MacBook running Windows using Parallels, VM Fusion
- Install Java™ Run Time Environment for 1.8.0 or later. If Java™ is not installed on your computer, we are providing the installer for the latest Java on the USB drive.

Minimum Recommended Hardware Requirements for Laptop Computer:

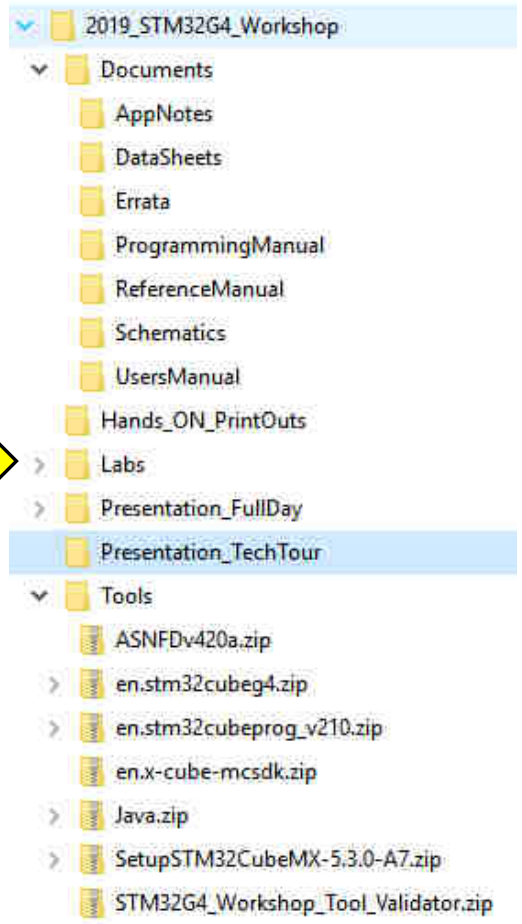
- Type A USB port
- 2+ GHz processor
- 4 GB of system memory (RAM)
- 10 GB of available disk space

Note: For laptops with only USB Type C ports, you will need a Type A (female) to Type C (male) adapter.

- STM32CubeMX: version 5.3.0 minimum
- STM32CubeG4: version 1.1.0 minimum
- STM32CubeProgrammer: version 2.1.0 minimum
- Keil MDK-ARM: version 5.28a minimum
- Keil MDK-ARM License: Professional Evaluation License Add On
- Tera Term or equivalent terminal emulator
- Java Runtime Environment: 1.8.0 minimum

- Copy the contents of your thumb drive or download to your **C:** root
- If not done automatically, name the top folder 2019_STM32G4_Workshop

This is the Labs folder. You will be here many times.

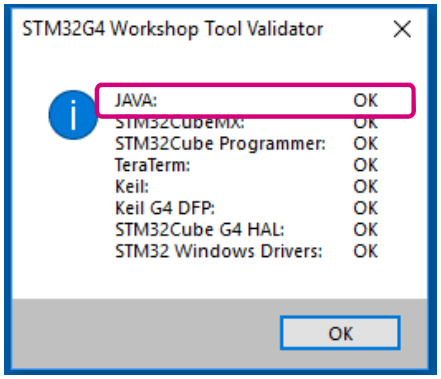


Note: We don't recommend copying to a deep path such as your Desktop or Documents because long path names can create build issues with some compilers.

STM32G4 Workshop Installation Guide – Java JRE 1.8.0+

Java™ Run Time Environment 1.8.0 or later is required to run Cube software tools.

To check if you have the correct version of Java installed please launch the app "**STM32G4_Workshop_Tool_Validator_vx.x.exe**" on the USB drive.

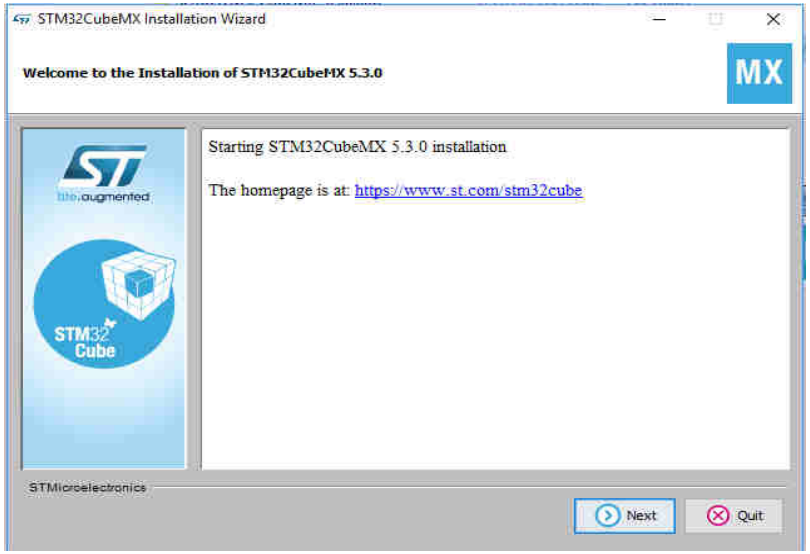


If needed, the Java installer is located in the **Java** directory:
To install Java right Click on "**jre-8u201-windows-i586.exe**" and Run as administrator.

life.augmented

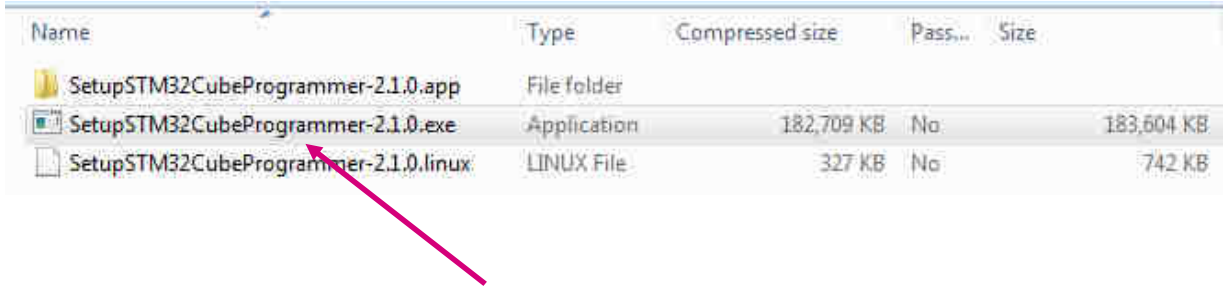
Unzip ***en.SetupSTM32CubeMX-5.3.0.zip*** and you will see this:

Right Click on **SetupSTM32CubeMX-5.3.0.exe** and Run as administrator:

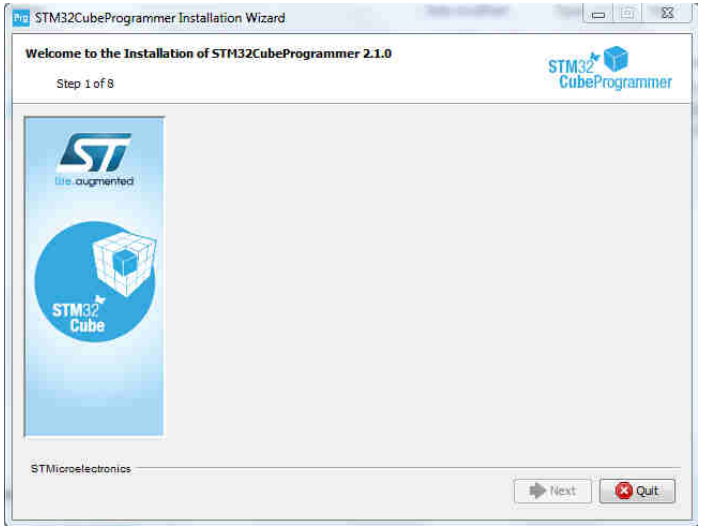


life.augmented

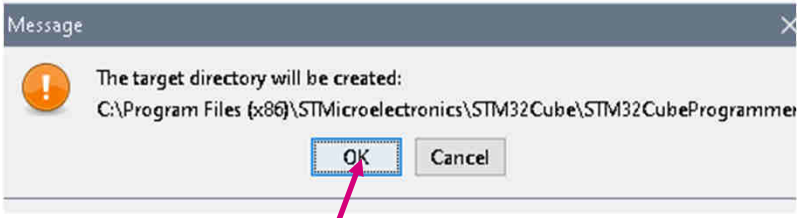
Unzip ***en.stm32cubeprog_v2.1.0.zip*** and you will see this:



Right Click on ***SetupSTM32CubeProgrammer-2.1.0.exe*** and Run as administrator:

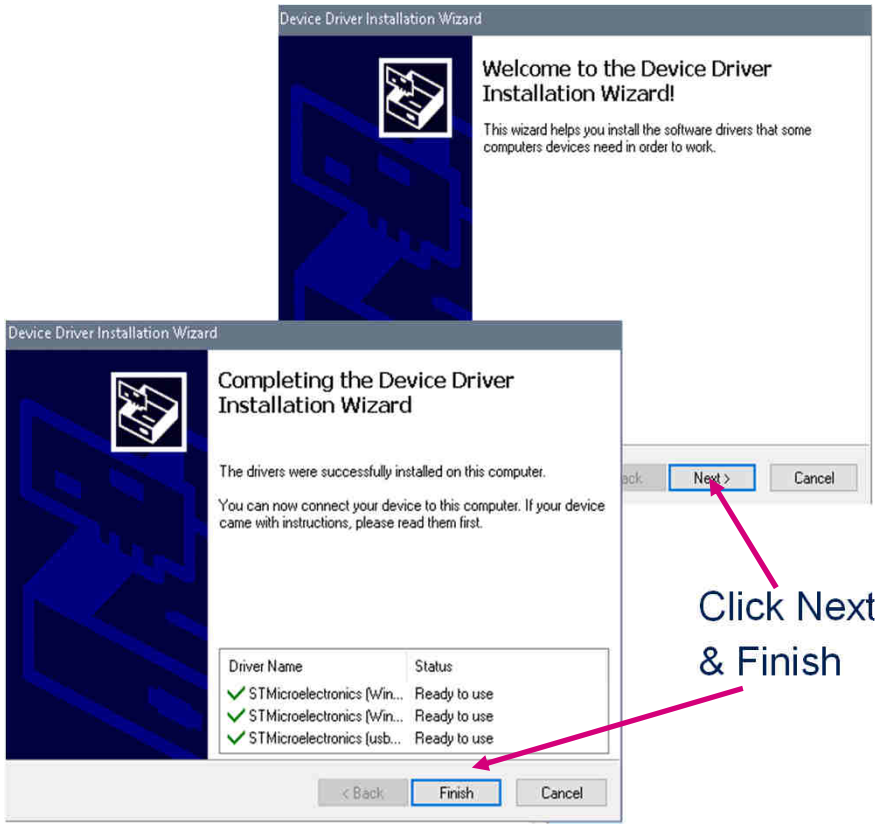


life.augmented



Click OK to create the Directory
(if the directory does not already exist)

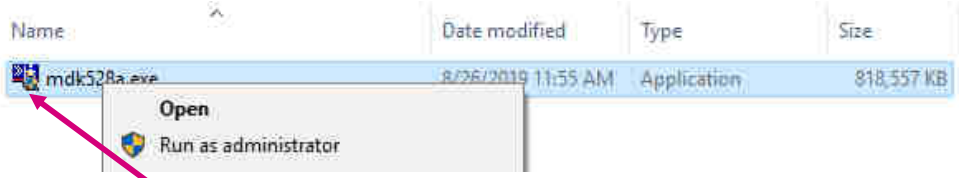
Install the Windows drivers if prompted



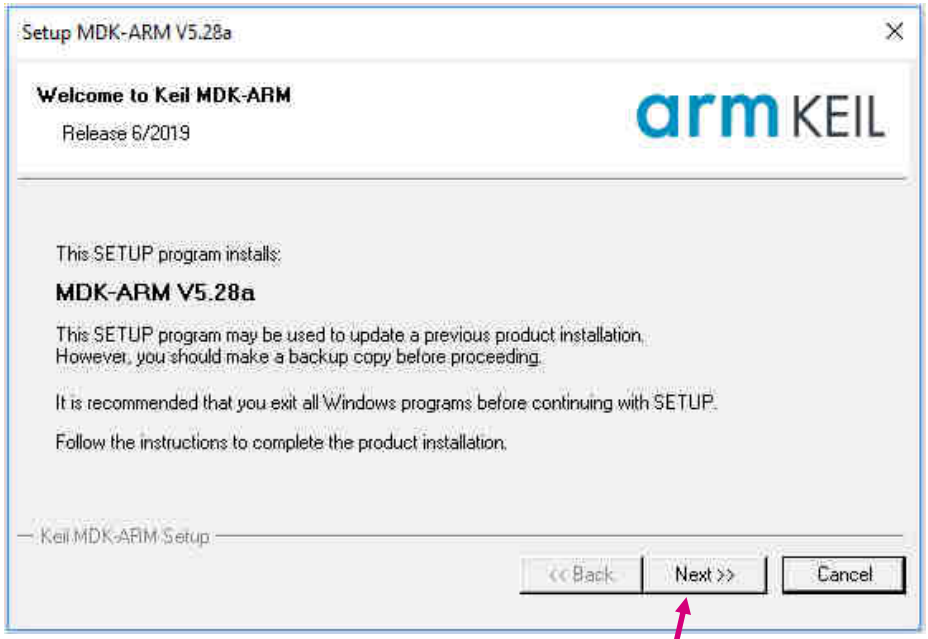
Click Next
& Finish

life.augmented

STM32G4 Workshop Installation Guide – Keil MDK-ARM v5.28a



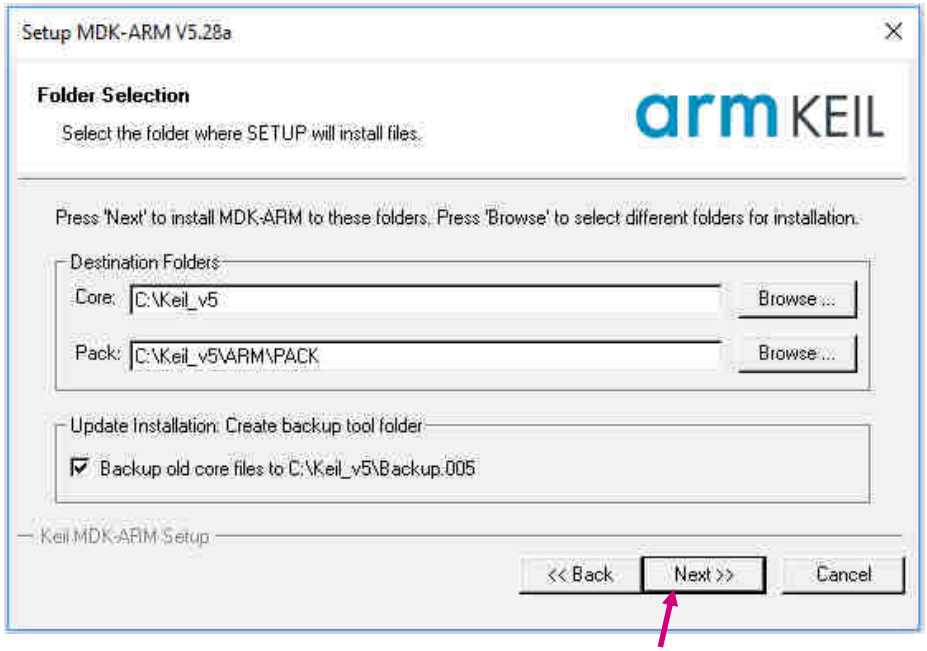
Right Click on **mdk528a.exe** and Run as administrator:



Click Next

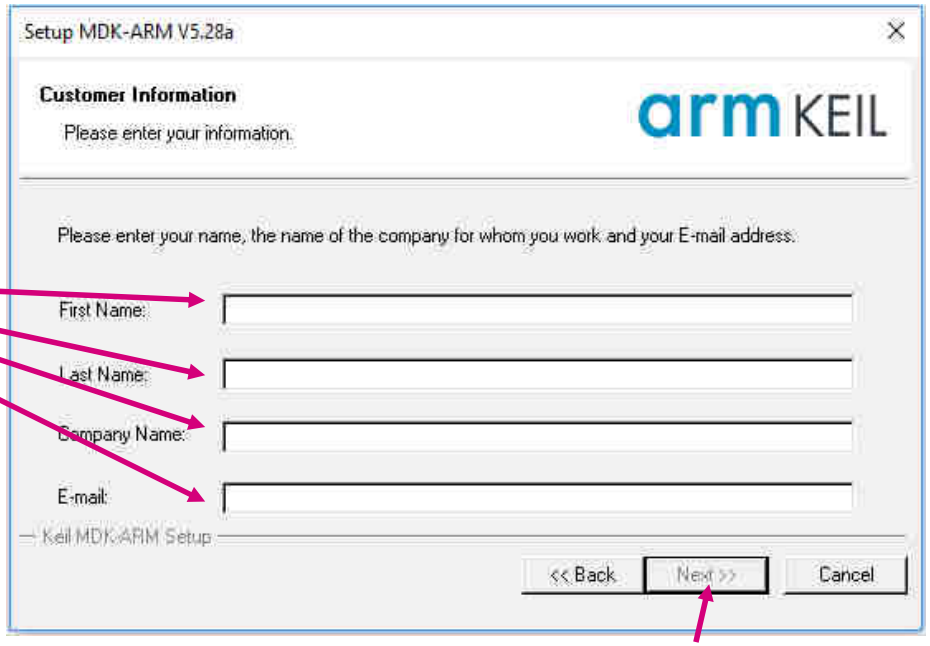
life.augmented

Leave the default folder locations as is



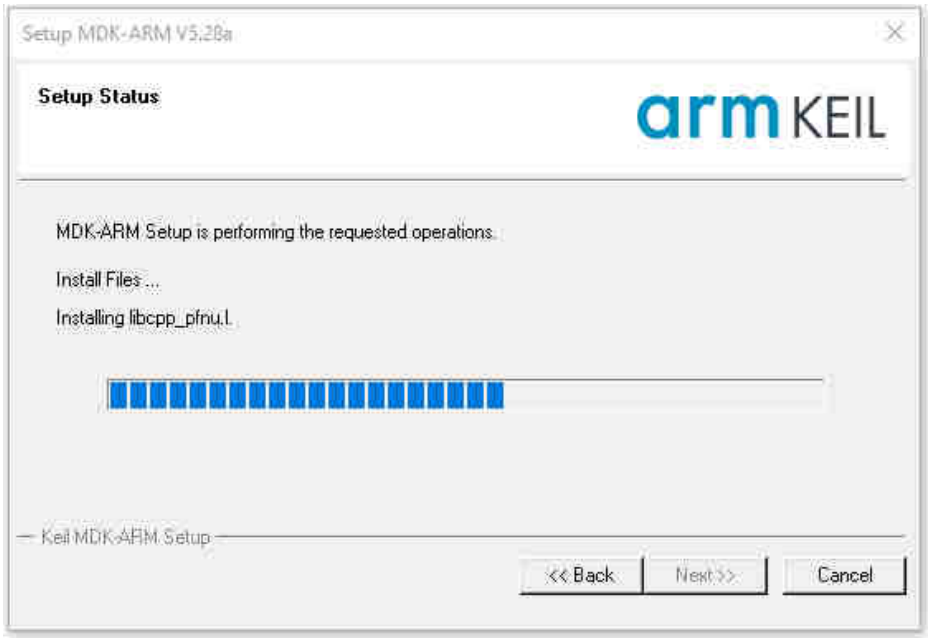
Click Next

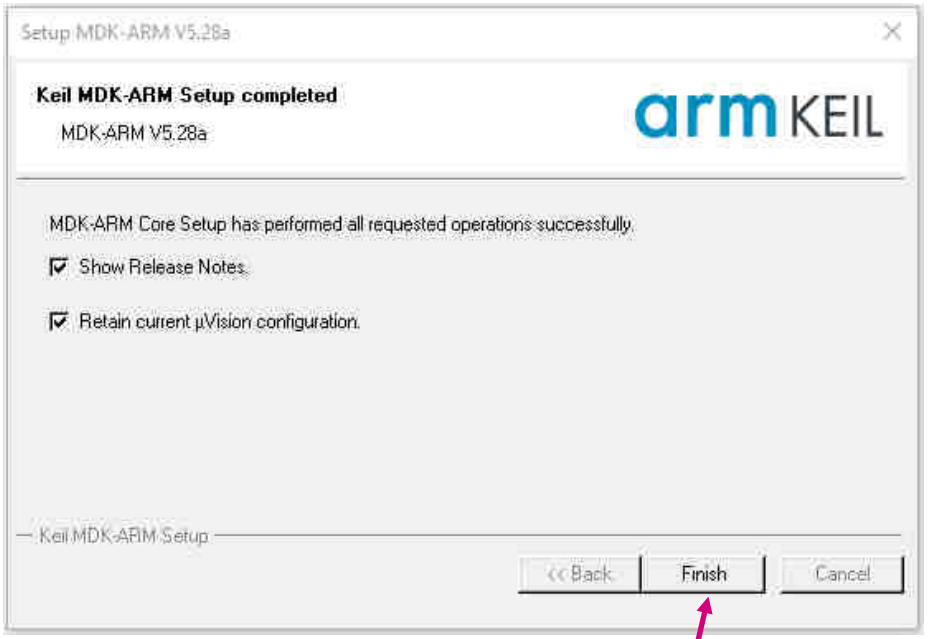
Enter your contact information



Click Next

The installation should run and look like this:

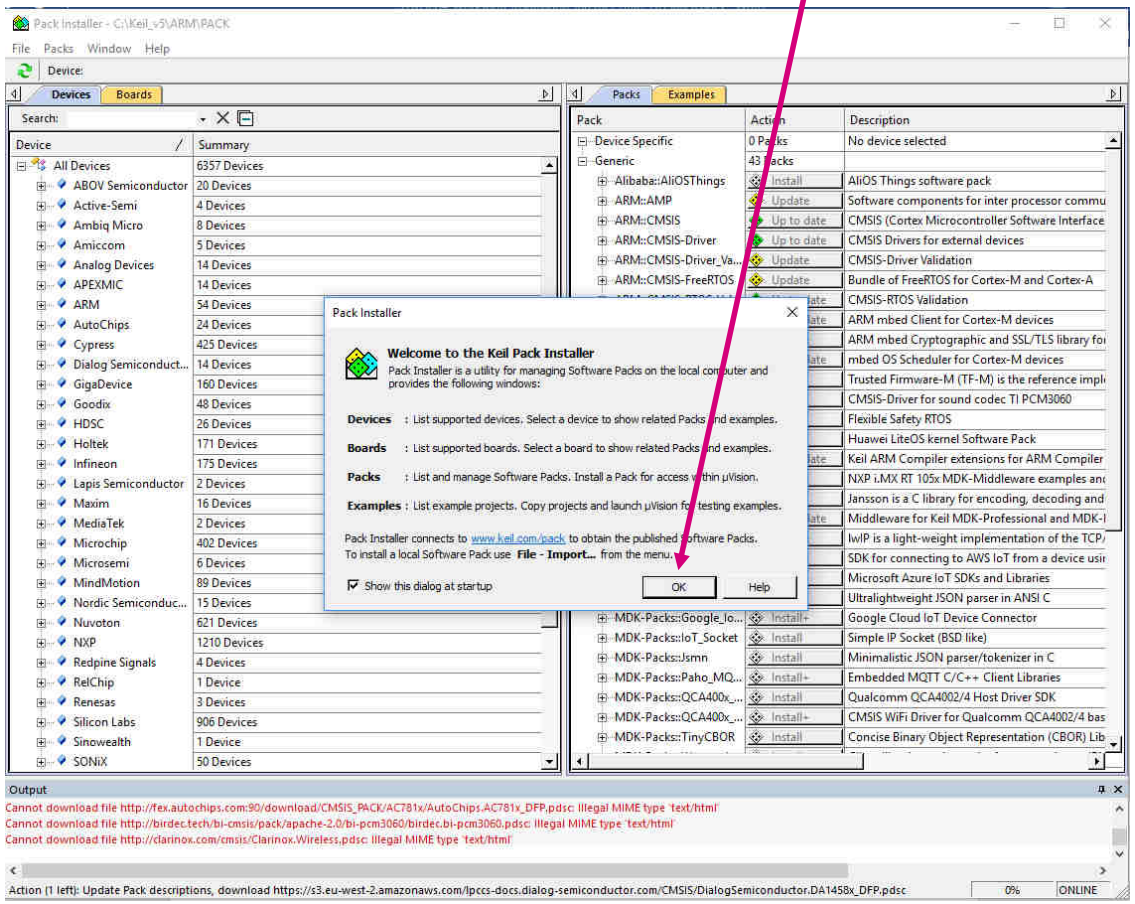




Once the installation has completed click Finish

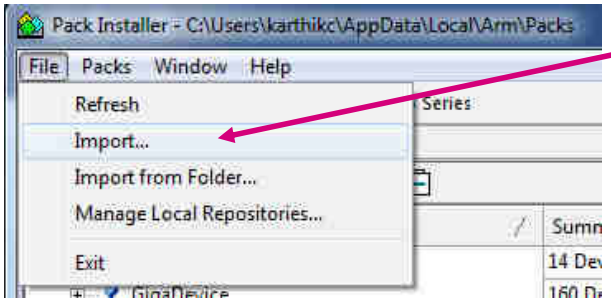
STM32G4 Workshop Installation Guide – Keil MDK-ARM v5.28a

The pack installer should automatically start, click OK to close the welcome screen



life.augmented

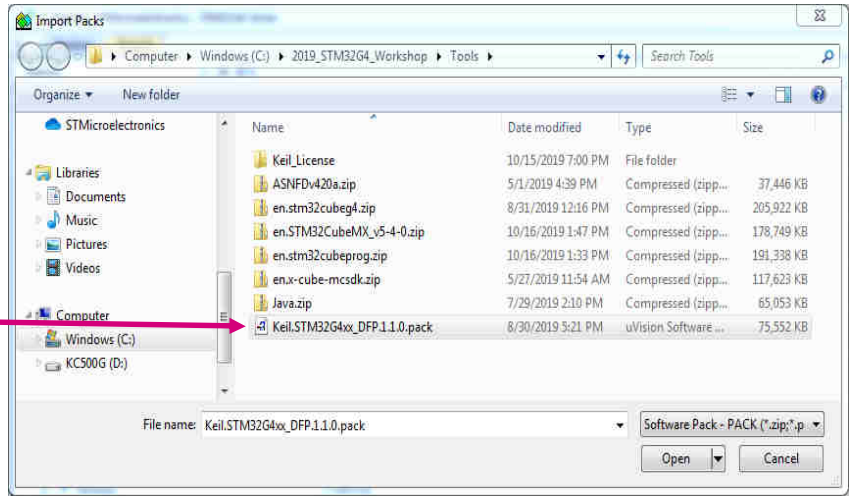
STM32G4 Workshop Installation Guide – Keil MDK-ARM v5.28a



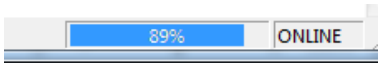
1 Click on File > Import

2

Choose this file to import:
C:\2019_STM32G4_Workshop\Tools\Keil.STM32G4xx_DFP.1.1.0.pack

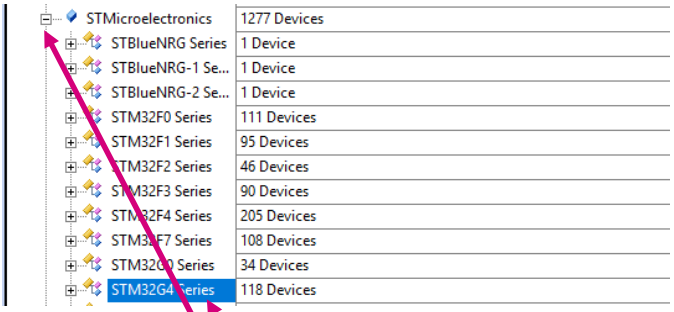


3 Wait until progress bar on bottom right corner is completed to 100%

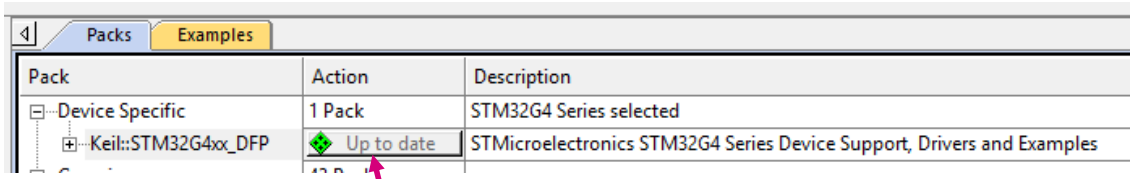


life.augmented

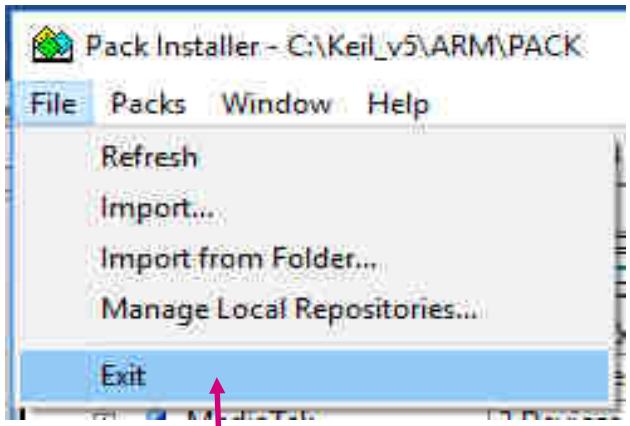
STM32G4 Workshop Installation Guide – Keil MDK-ARM v5.28a



1 On the “Device” (left) side of the screen scroll down and click on the “+” to expand “STMicroelectronics” and then click on the “STM32G4 Series” to select it



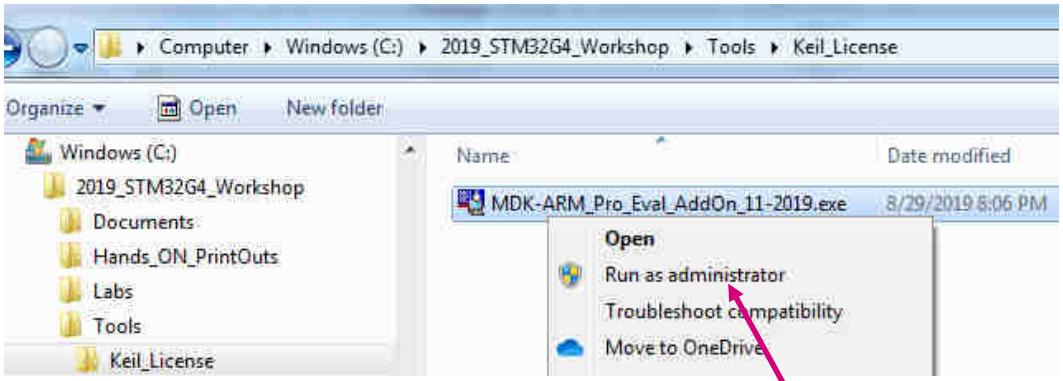
2 On the “Pack” (right) side of the screen, next to the “Keil: STM32G4xx_DFP”, once complete it should show “Up to date” 



3 Click on “File” and then “Exit” to quick the pack installer, installation is now complete

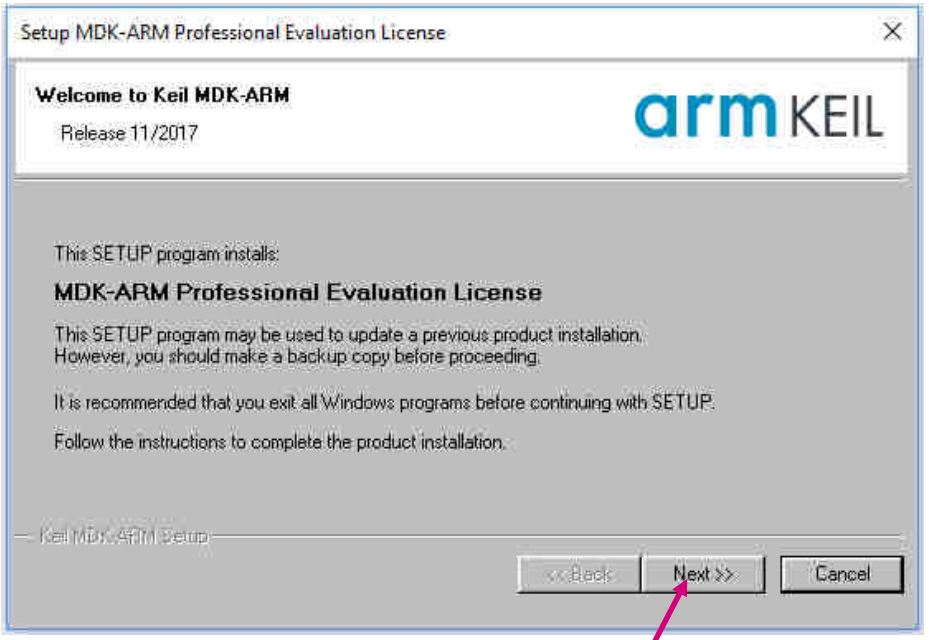
life.augmented

Keil MDK-ARM License Installation



C:\2019_STM32G4_Workshop\Tools\Keil_License\
MDK-ARM_Pro_Eval_AddOn_11-2019.exe

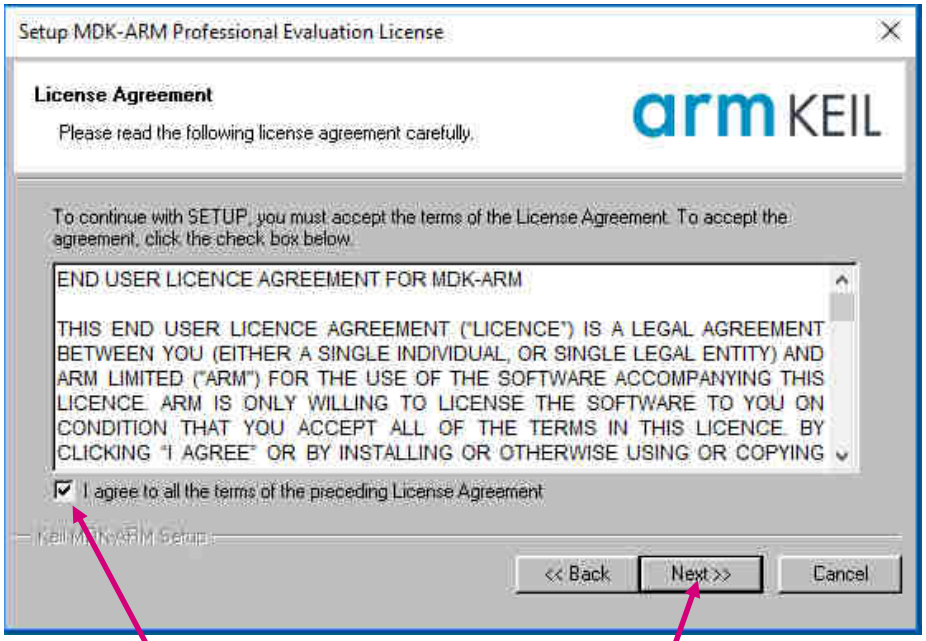
Right click and click “Run as administrator” to install the Professional Evaluation **License Add On**



Click **Next**

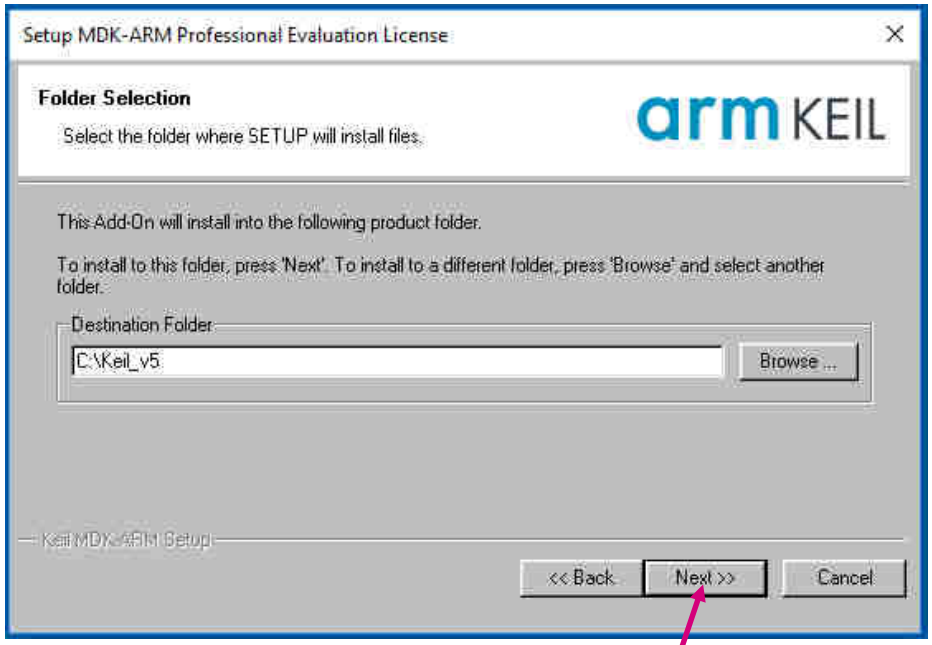
life.augmented

Keil MDK-ARM License Installation



Click the check box to agree to the terms

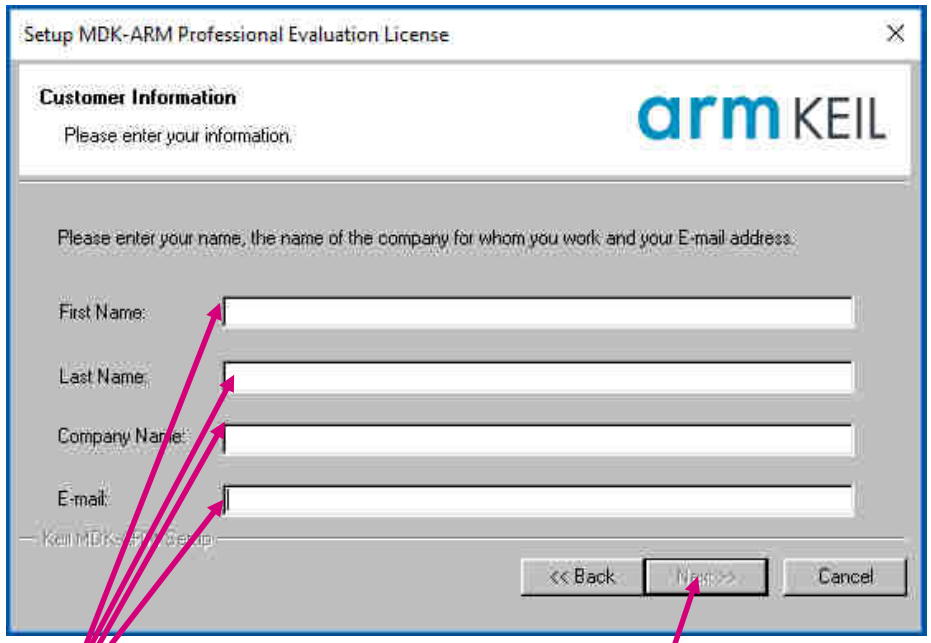
Click **Next**



Click **Next**

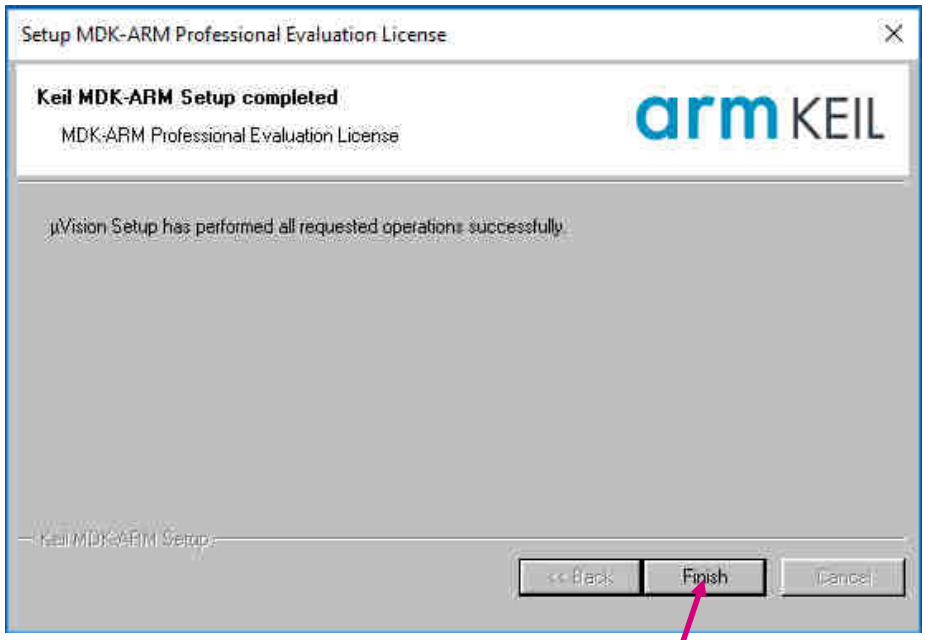
life.augmented

Keil MDK-ARM License Installation



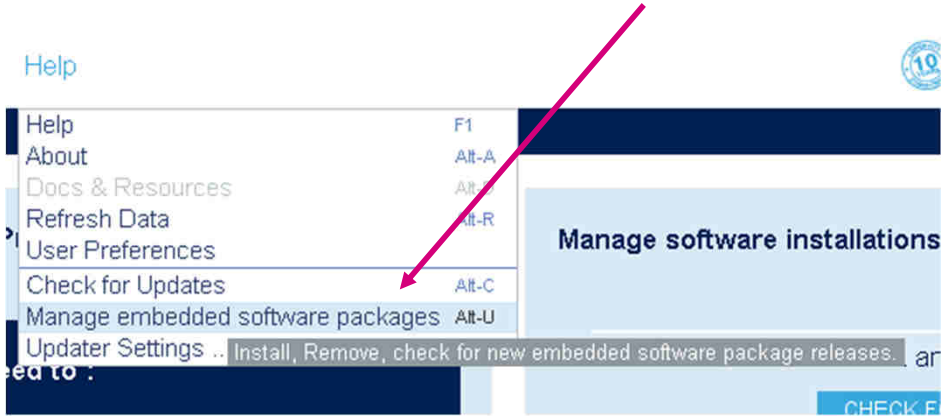
Enter your contact information

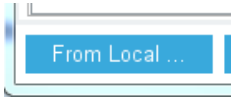
Click **Next**



Click **Finish**

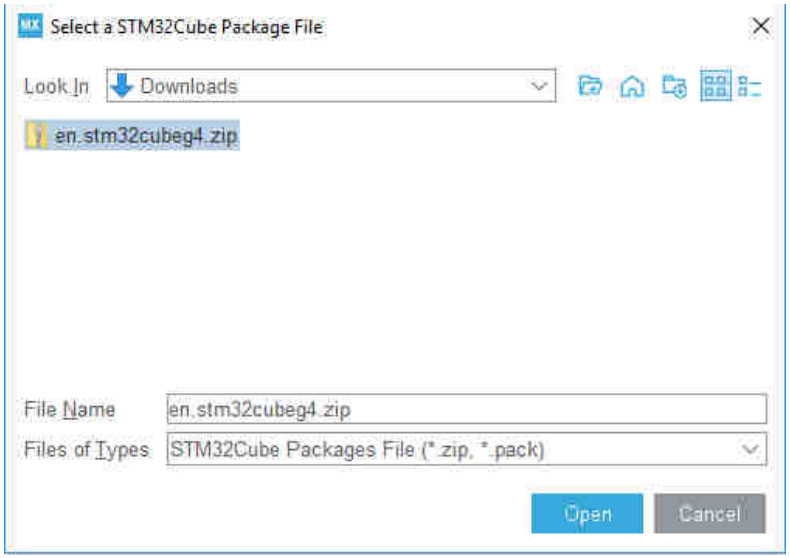
Open CubeMX and select ***Manage embedded software packages***





Select ***From Local…***

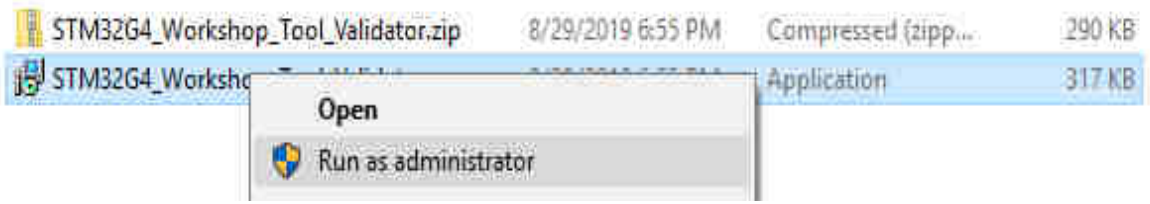
Select ***en.stm32cubeg4.zip***



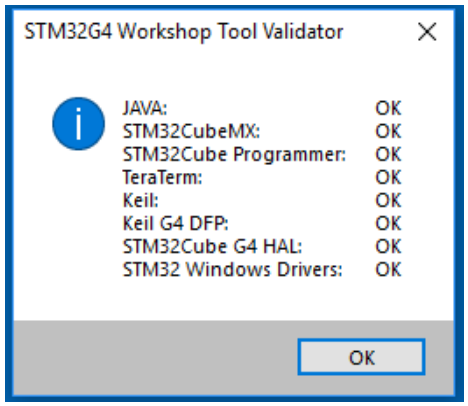
life.augmented

Validate toolchain installations

Run the ***STM32G4_Workshop_Tool_Validator.exe***



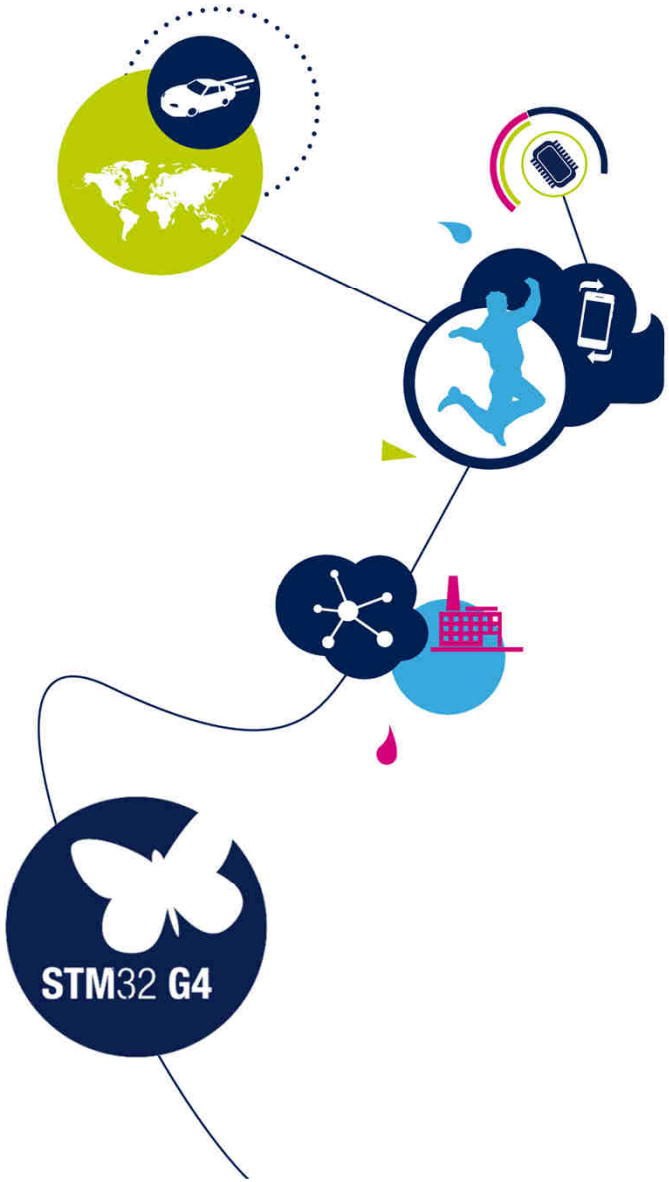
See a workshop proctor for any installation issues



life.augmented

STM32G4 mixed-signal MCU workshop

Introduction to the STM32G4
Mainstream Series
Mixed Signal MCU



life.augmented

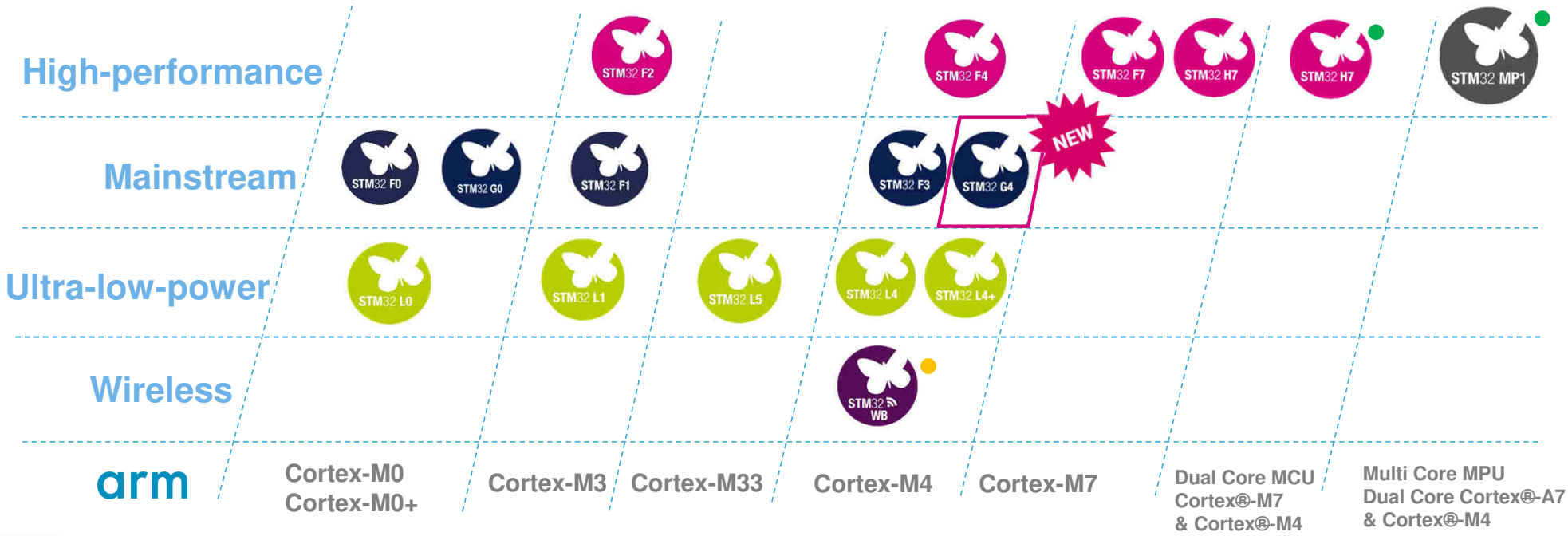
STM32G4 mixed-signal MCU workshop

Agenda

Software Installation
Introduction to the STM32G4 Series **Hands On: STM32G474 Discovery Kit Demo** **Hands On: DAC Signal Generation** **Hands On: Programmable Gain Amplifier**
Break
Hands On: CORDIC Hardware Accelerator **Filter Math Accelerator (FMAC) Overview** **Hands On: Dual Bank Flash Live Firmware Upgrade** **Wrap Up**

STM32 portfolio

17 product series / More than 50 product lines



Note ●: Cortex-M0+ Radio Co-processor, ● : Cortex-M4 Co-processor

STM32 Rolling Longevity Commitment

Longevity commitment is renewed every year

starting January 1st 2019

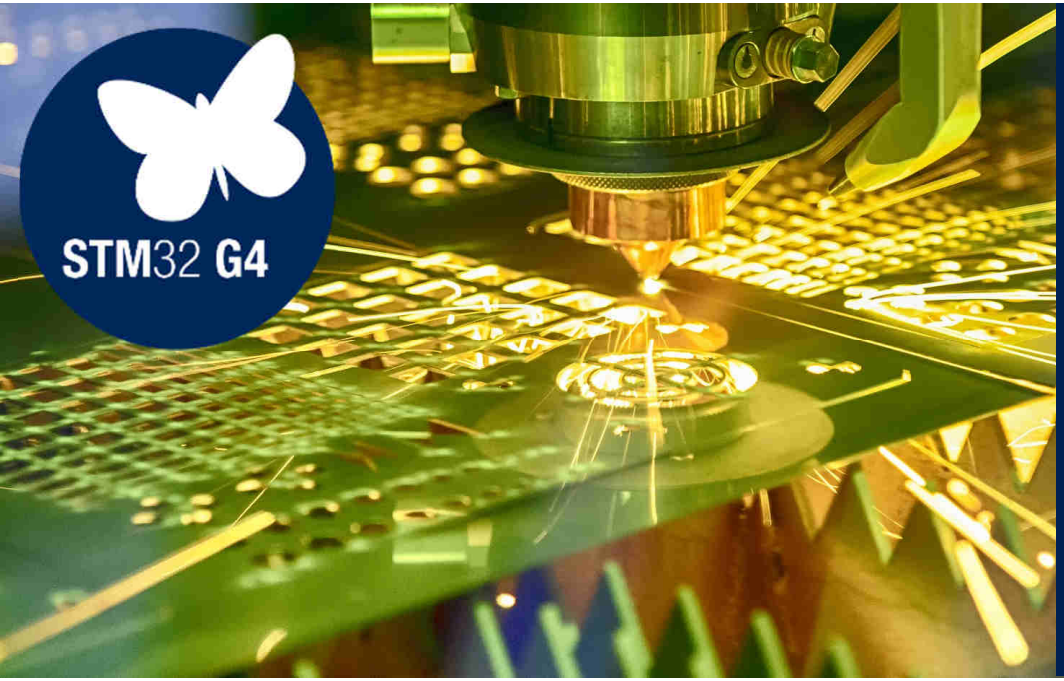
➔ Until 2029

• **STM32F1**	(launched in **2007**)	**22 years** of commitment
• **STM32L1**	(launched in **2009**)	**20 years** of commitment
• **STM32F2**	(launched in **2010**)	**19 years** of commitment
• **STM32F4**	(launched in **2011**)	**18 years** of commitment
• **STM32F0**	(launched in **2012**)	**17 years** of commitment
• **STM32F3**	(launched in **2012**)	**17 years** of commitment
• **STM32L0**	(launched in **2013**)	**16 years** of commitment
• **STM32F7**	(launched in **2014**)	**15 years** of commitment
• **STM32L4**	(launched in **2015**)	**14 years** of commitment
• **STM32L4+**	(launched in **2016**)	**13 years** of commitment
• **STM32H7**	(launched in **2017**)	**12 years** of commitment
• **STM32WB**	(launched in **2018**)	**11 years** of commitment
• **STM32G0**	(launched in **2018**)	**11 years** of commitment
• **STM32G4**	(launched in **2019**)	**10 years** of commitment

life.augmented

STM32G4 Series

Ideal for applications requiring MCU with advanced and rich analog peripherals



- Control applications (Motor Control…)
- Industrial equipment
- Instrumentation and Measurement
- Digital Power
  - Digital SMPS (switch mode power supply)
  - PFC (power factor correction)

life.augmented

STM32G4 Series – Key Messages

Performance

- Arm® Cortex®-M4 at 170 MHz
- 213 DMIPS and 550 CoreMark® results
- Better dynamic power consumption (163µA/MHz)
- ART Accelerator™ (dynamic cache)
- Mathematical accelerators
- CCM-SRAM Routine Booster (static cache)

Rich Integrated Analog and Digital

- Op-Amps (Built-in gain), DACs, Comparators
- 12-bit ADCs 4Msps with hardware oversampling
- CAN-FD (flexible data rate – 8Mbps bit rate)
- High resolution timer (184 ps)
- USB type-C Power Delivery3.0
- 1% RC accuracy [-5°..90°C], 2% full T° range

Safety and security focus

- Dual Bank Flash with ECC (error code correction)
- Securable Memory Area
- Hardware encryption AES-256
- SIL, Class-B
- SRAM with Parity bit

Secure Live Upgrade

Functional safety design packages

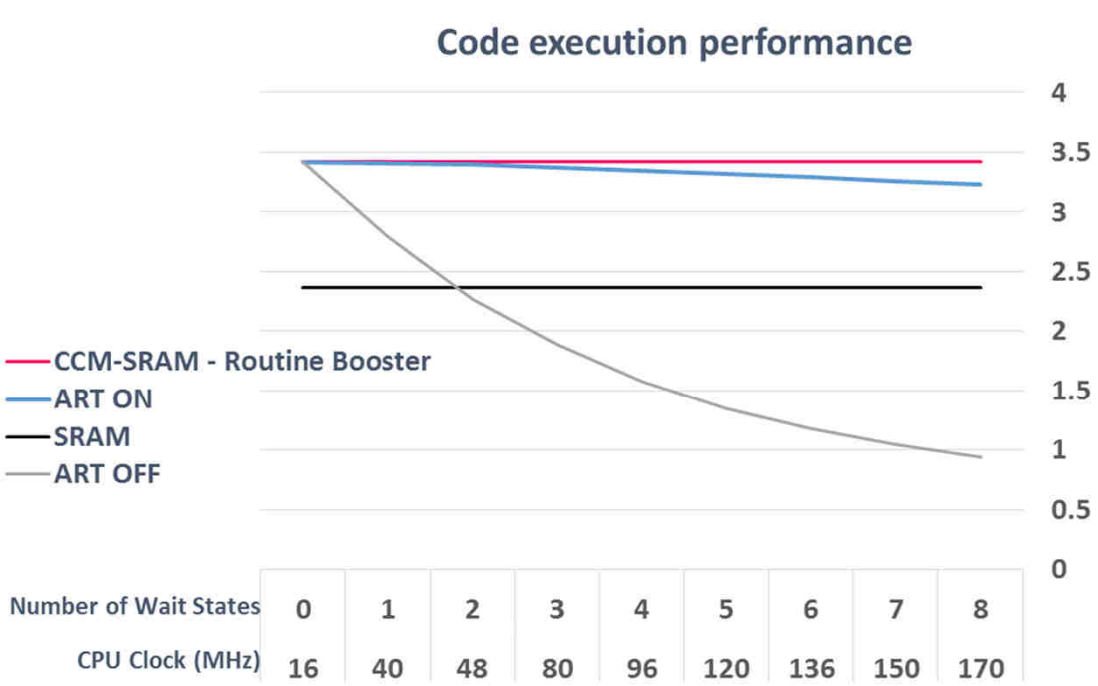
Complete portfolio

- Complements existing STM32F3 Series portfolio
- From -40°C up to 85 or 125°C devices
- From 32- up to 128-pin
- From 32KB to 512KB Flash

life.augmented

Greater Performance

Pure 170 MHz CPU performance (Arm® Cortex®-M4) with 3 accelerators



Arm Cortex-M4 with **FPU**

Up to 170 MHz CPU frequency

Up to 213 DMIPS and 550 CoreMark® results

3 different **HW accelerators**:

- **ART accelerator** (~dynamic cache) → Full code acceleration (average)
- **Routine Booster** CCM-SRAM (~static cache) → determinism preserved
- **Mathematical** (CORDIC + FMAC) NEW

life.augmented

Mathematical Accelerators

Function acceleration and CPU offload

1. CORDIC (Trigo)

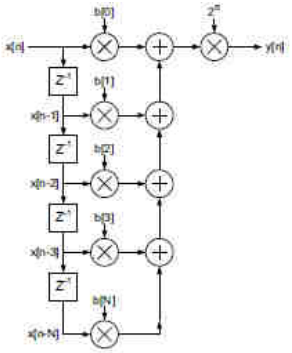
- Very helpful for Field Oriented Motor Control method (FOC)

- Vector rotation (polar to rectangular): Sin, Cos
- Vector translation (rectangular to polar): Atan2, Modulus
- Sinh, Cosh, Exp
- Atan, Atanh
- Square root
- Ln

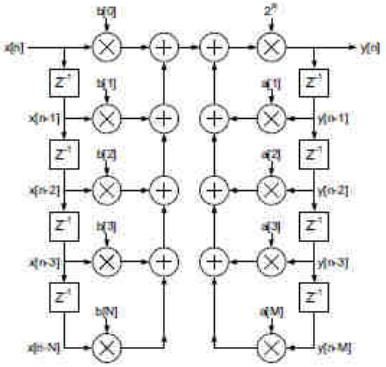
2. Filter Math Accelerator (FMAC)

- Can be used to create
  - 3p3z Compensator (→ Digital power)
  - Sigma Delta modulator
  - Noise Shaper

FIR filter

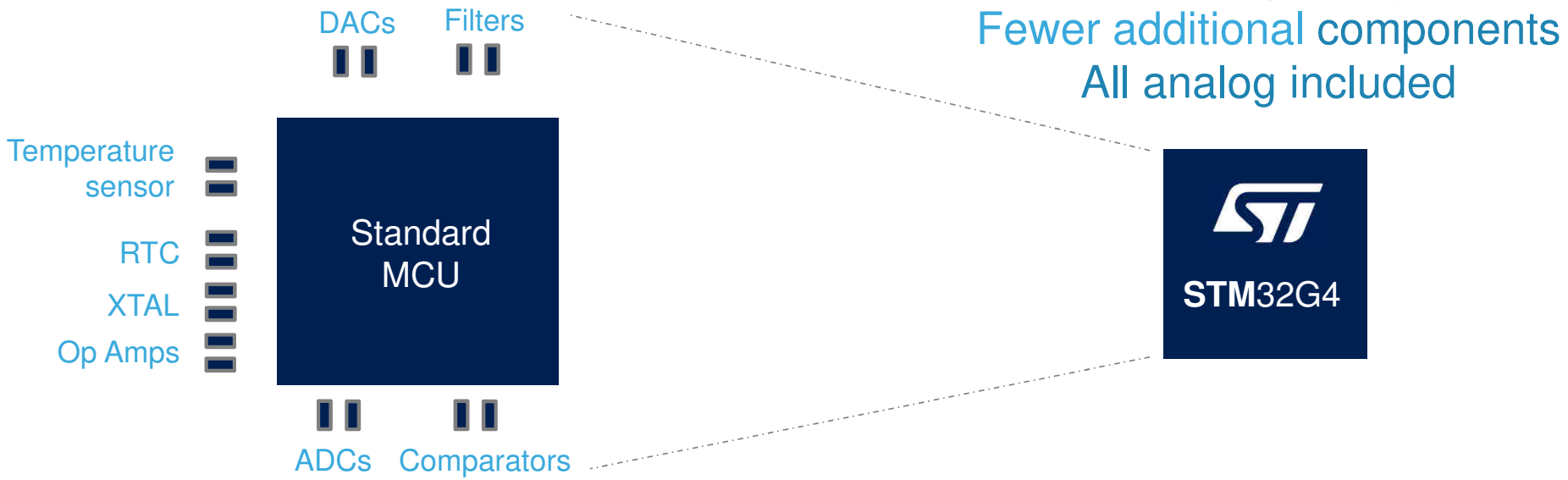


IIR filter



Reducing PCB Size and BOM Cost

System-on-Chip – All-in-one solution



Rich, Advanced Analog

Mixed-signal SoC for wide variety of applications

ADC (up to 5)	**Values**
Topology	**SAR 12-bit** + HW oversampling → **16-bit**
Sampling rate	Up to **4 Msps**
Input	Single-ended and differential
Offset and Gain compensation	Auto calibration to reduce gain and offset

DAC (up to 7)	**Values**
Sampling rate	**15 Msps** (internal) 1Msps (from buffered output)
Settling time	16ns

Op-Amp (up to 6)	**Values**
GBW	**13 MHz**
Slew rate	**45 V/µs**
Offset	3mV over full T° range 1.5mV @ 25°C
PGA Gain (accuracy)	2, 4, 8, 16, -1,-3,-7,-15 **(1%)** 32, 64, -31,-63 (2%)

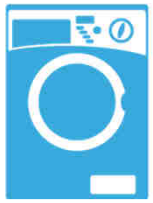
Comparator (up to 7)	**Values**
Power supply	1.62 .. 3.6V
Propagation delay	**16.7ns**
Offset	-6 .. +2 mV
Hysteresis	8 steps: 0, 9, 18, 27, 36, 45, 54, 63 mV

Key Features for Targeted Applications

Motor Control

Home appliances, E-bikes, Air Conditioning

- Fast CPU 170MHz
- Mathematical accelerator (Cordic)
- Advanced Motor Control timers
- Fast comparators
- 4Msps ADC-12bit + HW oversampling
- Op-Amp with built-in gain (PGA)
- DAC-12bit
- 1% RC accuracy
  (UART communication w/o external Xtal)



High-End Consumer

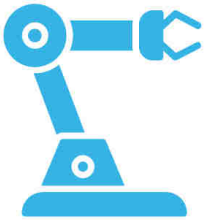
Rechargeable devices, drones, toys

- Low-thickness, small form-factor
- Low consumption in run mode ~ 160µA/MHz
- Embedded analog
- SAI (Sound Audio Interface)
- USB type-C Power Delivery 3.0

Industrial devices Measurements

Industrial equipment

- Fast CPU 170MHz
- Mathematical accelerator (Cordic)
- High temperature 125°C
- CAN FD support
- SPI, USART, I²C
- Advanced timers
- Real Time Clock with backup registers
- Dual bank flash for **live** upgrade
- AES & security



Digital Power

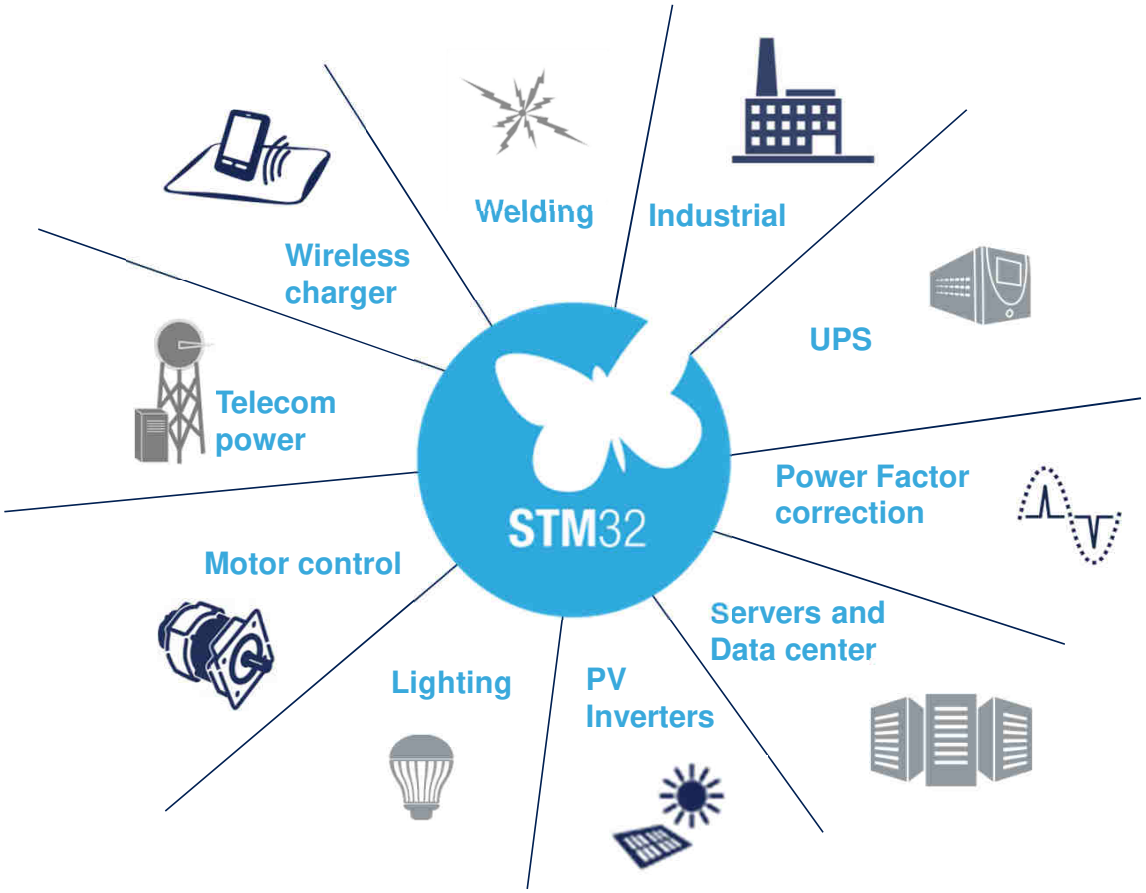
Servers, Telecom, EV Charging station

- Fast CPU 170 MHz
- Mathematical accelerator (Filtering)
- 12ch High Resolution timer (184ps)
- 4Msps ADC-12bit + HW oversampling
- Fast comparators (17ns)
- Embedded analog
- Dual bank flash for **live** upgrade
- AES & security
- FMAC for 3p3z compensation

life.augmented

Ease Digital Power Conversion

Enhance your digital power solutions using the STM32G4's full features High Resolution Timer (HRTIM)

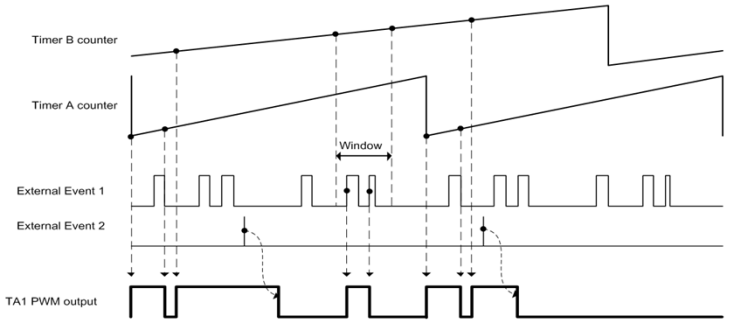


life.augmented

HRTimer – Not only High Resolution…

High resolution PWM

- 12 channels with 184ps resolution on frequency and duty cycle
- 184ps is equivalent to 5.4GHz timer clock



Flexible PWM generation

- 7x independent time base to create various shape of PWM
- 6x complementary pair PWM outputs
- Up to 32 set/reset transition per PWM period thx to the built-in crossbar
- Master/Slave configuration for multi phase converter

Multiple Event handler

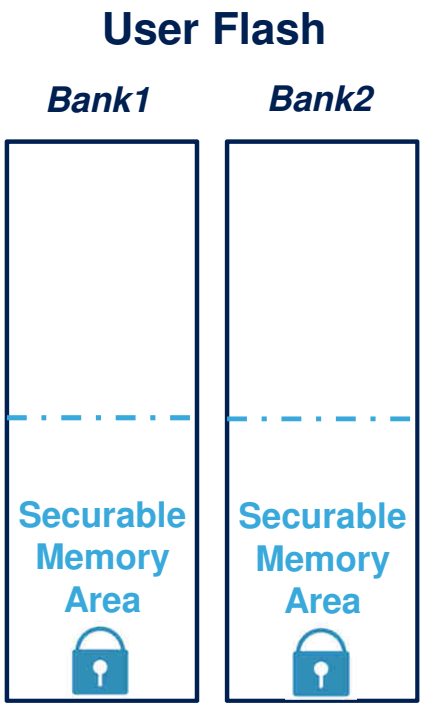
- 6x Digital and Analog fault input
- 10x Events cycle to cycle current control or PWM restart (constant Ton/Toff)
- Blanking, windowing and digital filter

12 independent channels

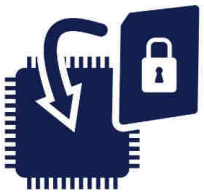
- Any topology supported from 1x 12 PWM (triple interleaved LLC (servers application) up to 12x1 PWM (multiple independent buck converters (lighting)

Greater Security

Integrated security features, ready for tomorrow's needs



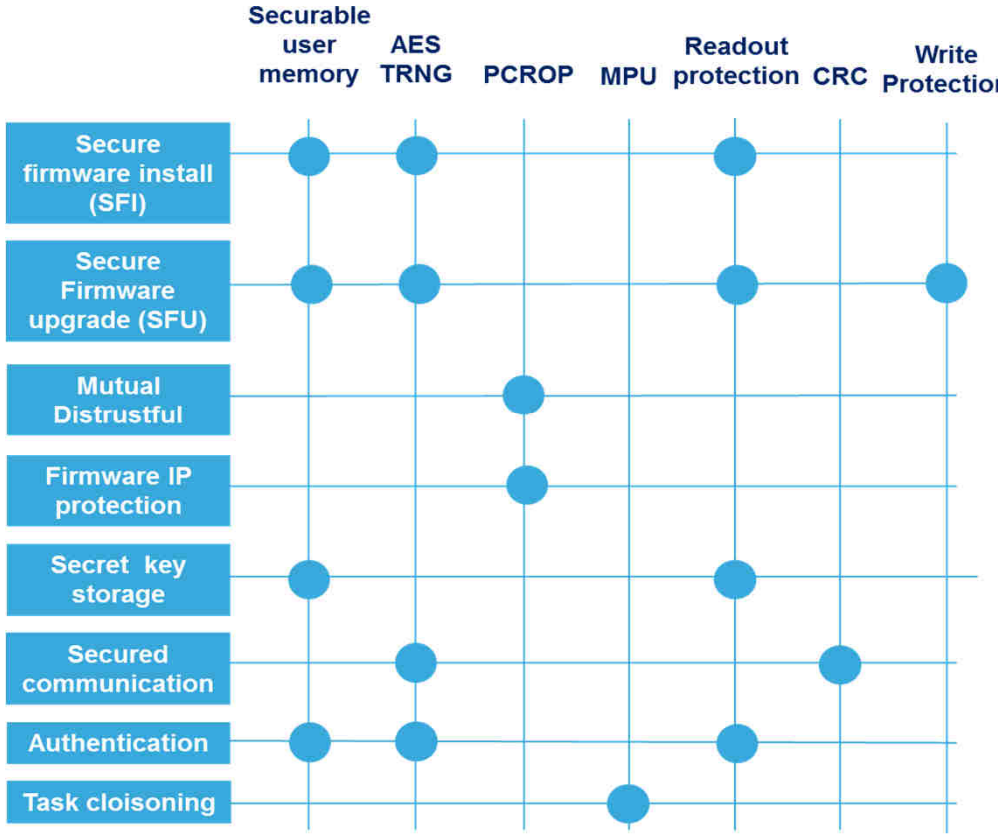
STM32Trust: a complete toolset
an ecosystem for embedded security



STM32 Trust

https://www.st.com/stm32trust

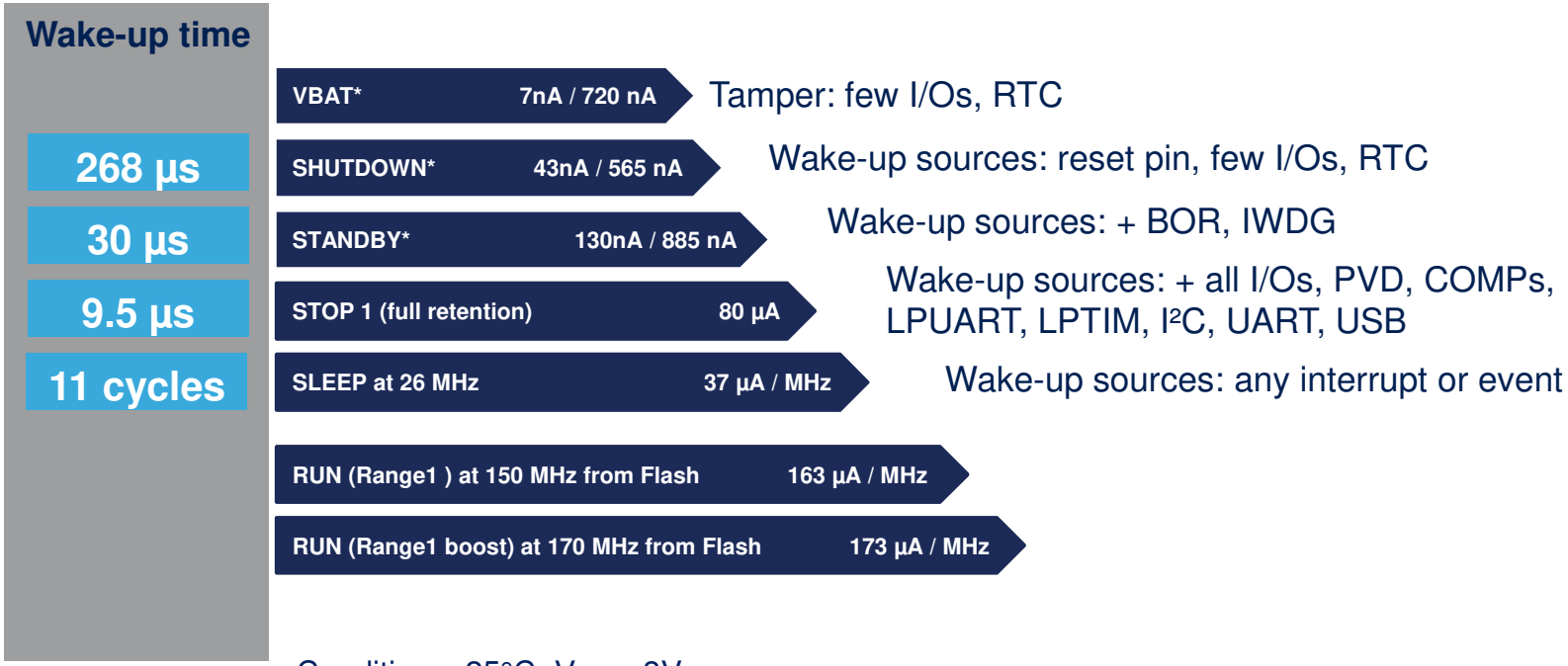
Securable Memory Area:
- Configurable size
- Can be secured once exiting
  - No more access nor debug possible
- Good fit to store critical data
  - Critical routines
  - Keys



life.augmented

Dynamic Efficiency Modes

When Mainstream MCU Series meets low-power requirements

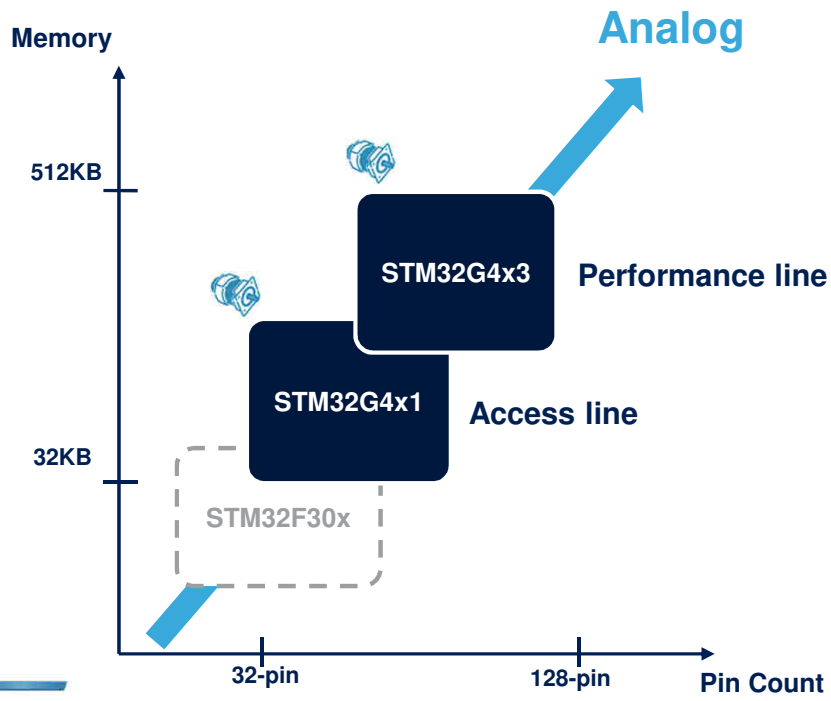


Conditions: 25°C, V_{DD} = 3V

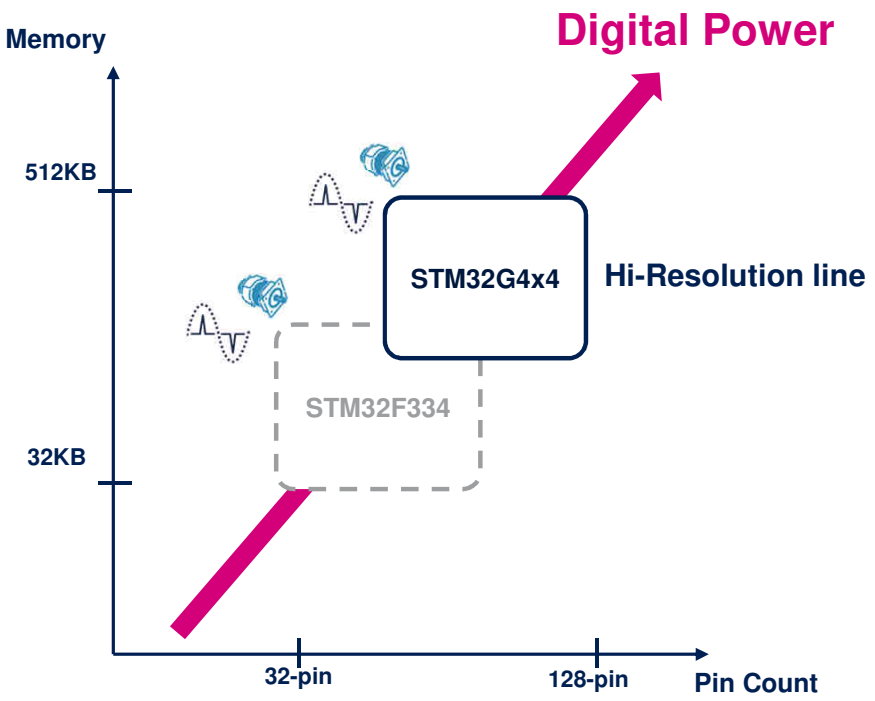
Note : * without RTC / with RTC

STM32G4 Products Lines

General Purpose



Applications Specific



life.augmented

Extensive & Innovative Peripheral Set

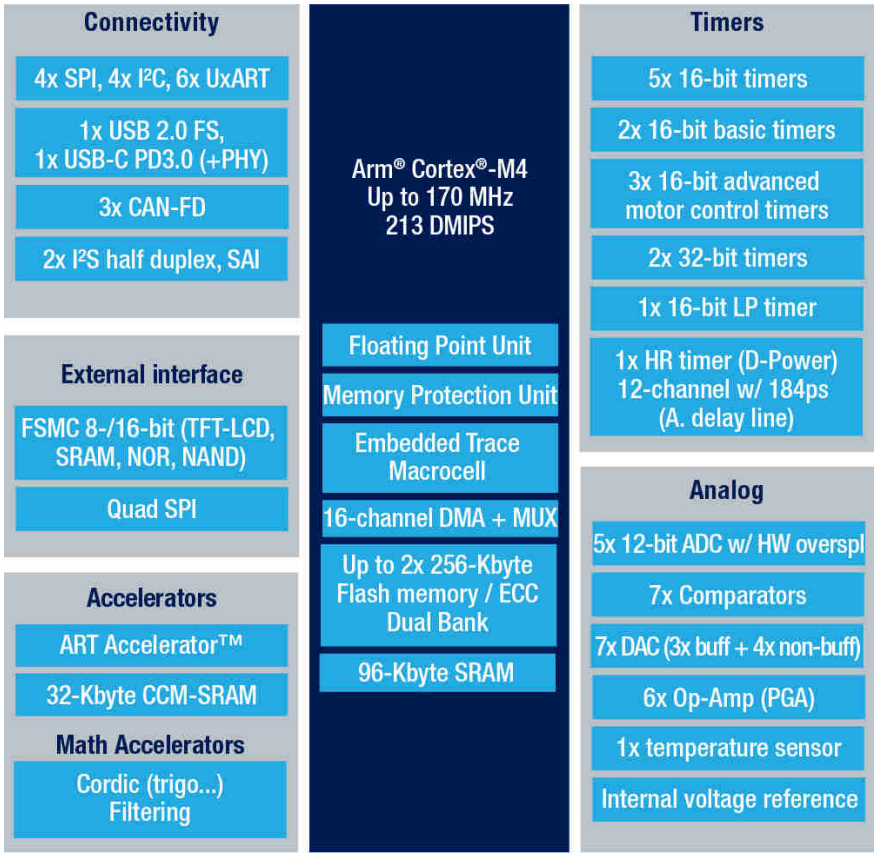
No compromise on what matters

Unit parameters	STM32G474 Hi-Resolution line	STM32G473 Performance line	STM32G431 Access line
Core, frequency	Arm Cortex-M4, 170 MHz		
Flash (max)	512 Kbytes (2x256KB dual bank)		128 Kbytes single bank
RAM (up to)	96 Kbytes		22 Kbytes
CCM –SRAM (code-SRAM)	32 Kbytes		10 Kbytes
12-bit ADC SAR	4x 12-bit 4 MSPS		2x 12-bit 4 MSPS
Comparator	7		4
Op amp with 4 built-in gain values with 1% accuracy	6		3
12-bit DAC	7		4
Motor Control timer	3x (170 MHz)		2x (170 MHz)
CAN-FD	3x		1x
12 channel Hi-resolution Timer	1x	-	-
Power supply	1.72 to 3.6 V		

life.augmented

High Resolution and Performance lines [128KB .. 512KB]

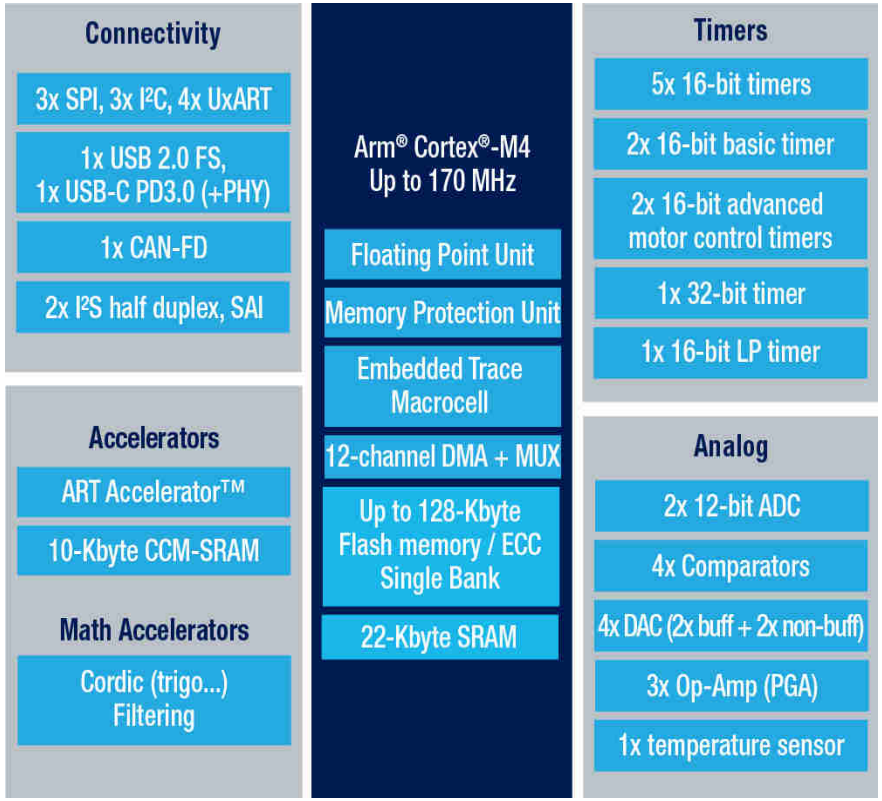
- 32-bit Arm Cortex-M4 core with FPU
- ART + CCM-SRAM + Mathematic Accelerators
- Dual Bank Flash with ECC
- SRAM with Parity bit
- +/- 1% internal clock
- 1.72 to 3.6V power supply
- Up to 125°C



- High resolution timer
- 3x Advanced Motor Control timers
- Rich Advanced Analog
- 3x CAN Flexible Data rate
- USB-C Power Delivery3.0
- Advanced Security and Safety features
- Robustness: highest level 5 / FTB/ESD - IEC 61000-4-4

Access line [32KB .. 128KB] and up to 512KB in H1-2020 !

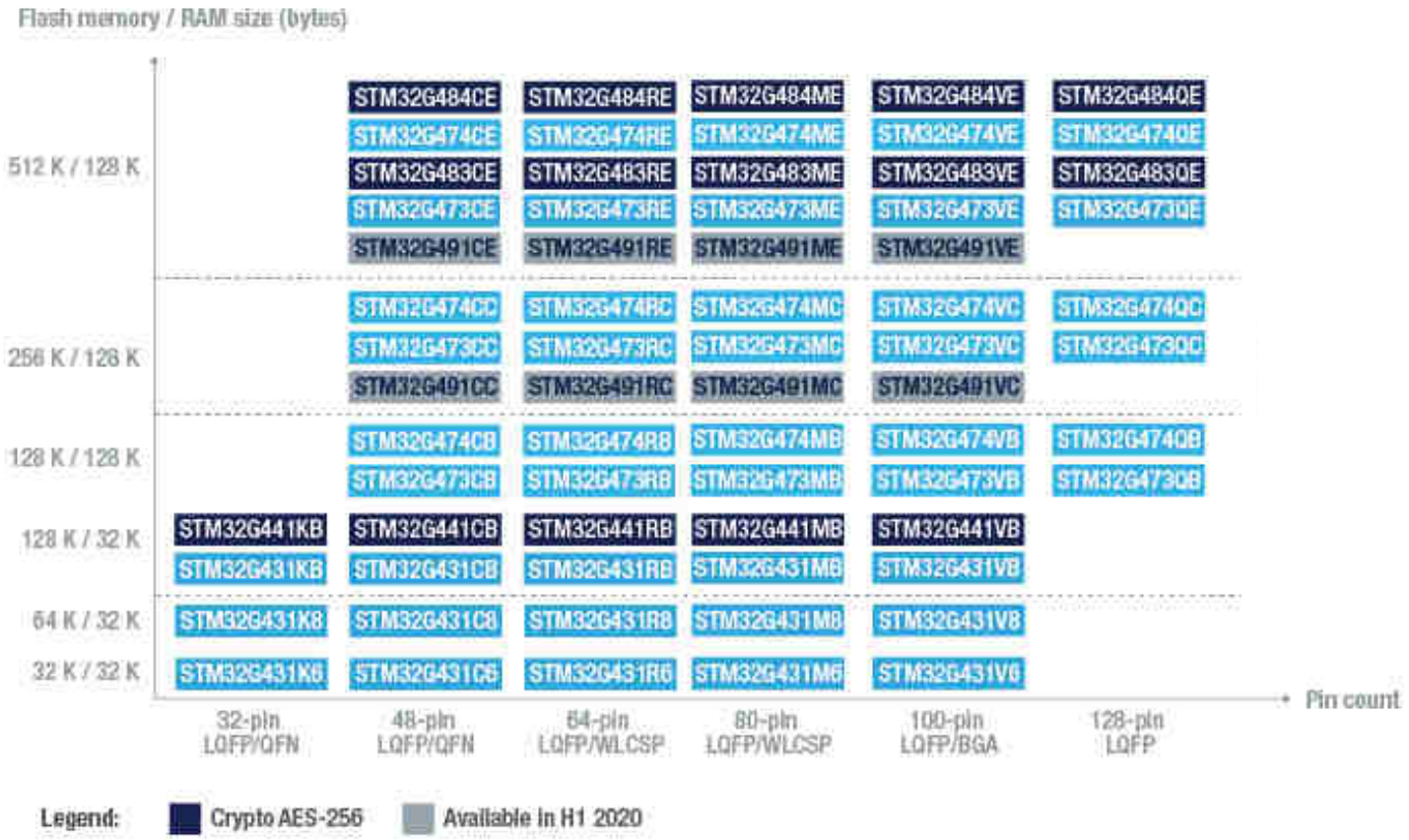
- 32-bit Arm Cortex-M4 core with FPU
- ART + CCM-SRAM + Mathematic Accelerators
- Single Bank Flash with ECC
- SRAM with Parity bit
- +/- 1% internal clock
- 1.72 to 3.6V power supply
- Up to 125°C



- 2x Advanced Motor Control timers
- Rich Advanced Analog
- CAN Flexible Data rate
- USB-C Power Delivery3.0
- Advanced Security and Safety features
- Robustness: highest level 5 / FTB/ESD - IEC 61000-4-4

life.augmented

STM32G4 Portfolio



life.augmented

Broad Portfolio

Portfolio extended to support budget applications efficiently

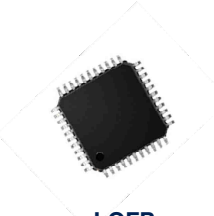
More memory and pin counts

Flash memory (bytes)	32-pin LQFP QFN	48-pin LQFP QFN	64-pin LQFP BGA WLCSP	80-pin LQFP WLCSP	100-pin LQFP BGA	121-pin BGA	128-pin LQFP
512 K		✓	✓	✓	✓	✓	✓
256 K		✓	✓	✓	✓	✓	✓
128 K	✓	✓	✓	✓	✓	✓	✓
64 K	✓	✓	✓	✓	✓		
32 K	✓	✓	✓	✓	✓		

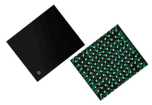
More packages

WLCSP

QFN



LQFP



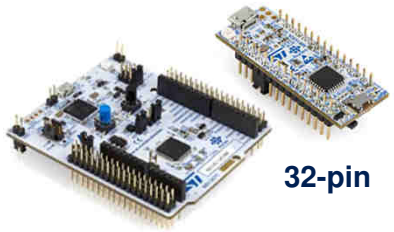
BGA

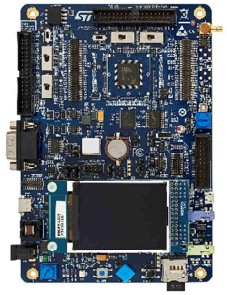
Note: new packages in STM32 portfolio

life.augmented

STM32G4 Hardware Solutions

Accelerate evaluation, prototyping and design





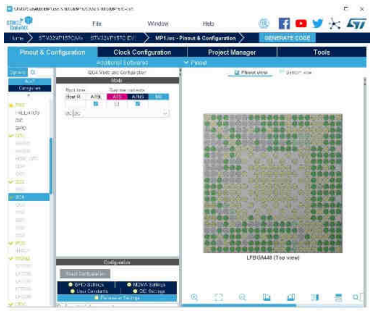
STM32 Nucleo	Evaluation boards	Motor Control Pack	Discovery kits
Flexible prototyping	**Full feature STM32G4 evaluation**	**Full feature for Motor Control and Analog**	**Key feature prototyping**
• NUCLEO-G431RB • NUCLEO-G474RE • NUCLEO-G431KB	• STM32G484E-EVAL • STM32G474E-EVAL • STM32G474E-EVAL1	• P-NUCLEO-IHM03	• B-G474E-DPOW1 • B-G431B-ESC1

life.augmented

STM32G4 Software Tools

Complete support of Arm Cortex-M ecosystem

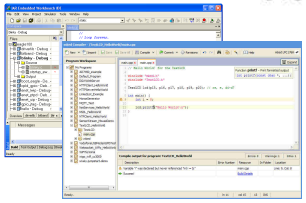
STM32 CubeMX



arm KEIL IAR SYSTEMS

eclipse

STM32 CubeIDE



All-in-one STM32 programming tool
Multi-mode, user-friendly

STM32CubeMX	IDEs Compile and Debug	STM32 Programming Tool
STM32CubeMX • Configure and generate Code • Conflicts solver	**Flexible Solutions** • Partners IDE, like IAR and Keil • Free IDE based on Eclipse, like STM32CubeIDE*	**STM32CubeProgrammer** • Flash and/or system memory • GUI or command line interface

* SW examples will be available in Q4 19

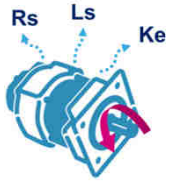
life.augmented

Dedicated Ecosystems

Motor Control

- **Complete ecosystem** (HW boards, SW Development Kit (SDK), docs and trainings)
  - X-CUBE-MCSDK (v5.4)
    - Motor Control FW library based on STM32Cube HAL and LL
    - Motor control workbench: Graphical configurator of the motor control library linked with STM32CubeMx
  - P-NUCLEO-IHM03: Motor Control Nucleo pack
    - NUCLEO-G431RB Nucleo-64
    - X-NUCLEO-IHM16M1 motor driver expansion board
    - Low Voltage motor

- **State of the art algorithms** (FOC, 6-step, sensorless…)
- **Motor Profiler**: Plug and spin your motor within less than one minute



Digital Power

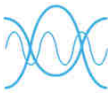
- **Complete ecosystem** (HW boards, FW examples, SW tools, docs and trainings)
- **Dedicated HRTIM Cook Book - AN4539:** How to operate the Hi-Resolution timer in different topology
- **Digital Power training** (PSU and PFC) – based on STM32 G4 series – done in collaboration with Biricha (from Q4 2019)

STM32G4 Series – Take Away

Analog-rich MCUs for mixed-signal applications

Performance

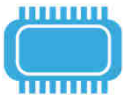
170MHz Cortex-M4 coupled with 3x accelerators



Rich and Advanced Integrated Analog

ADC, DAC, Op-Amp, Comp.

Safety and security focus



Large portfolio available from NOW!

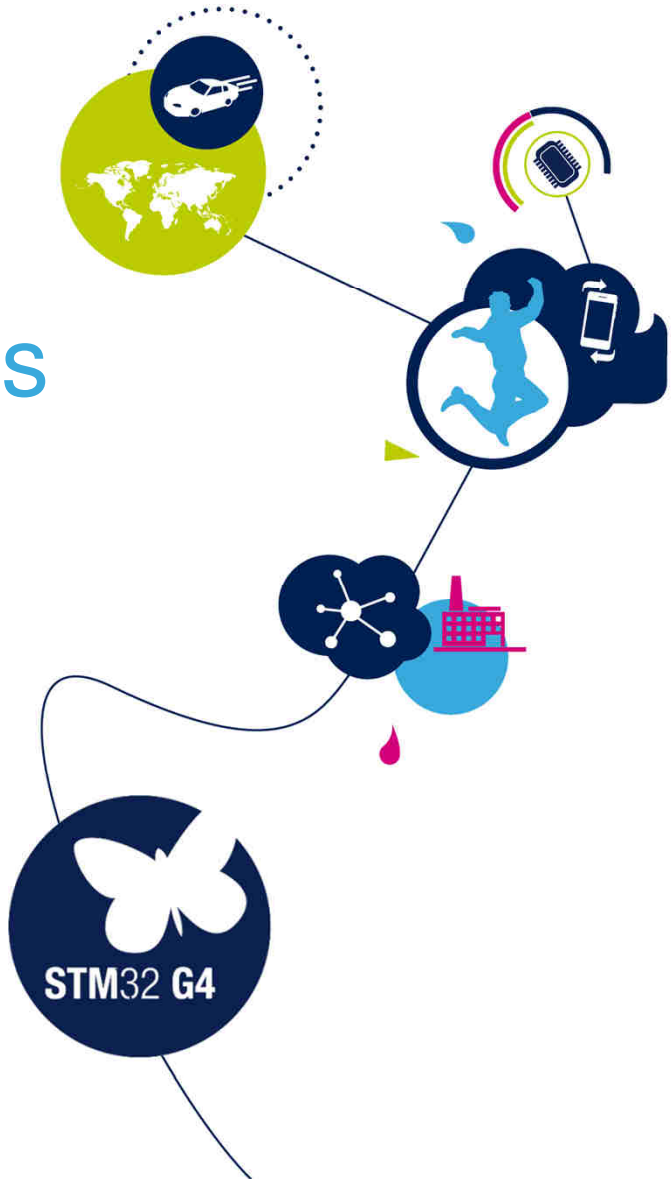
32..512KB Flash memory

32..128-pin packages

life.augmented

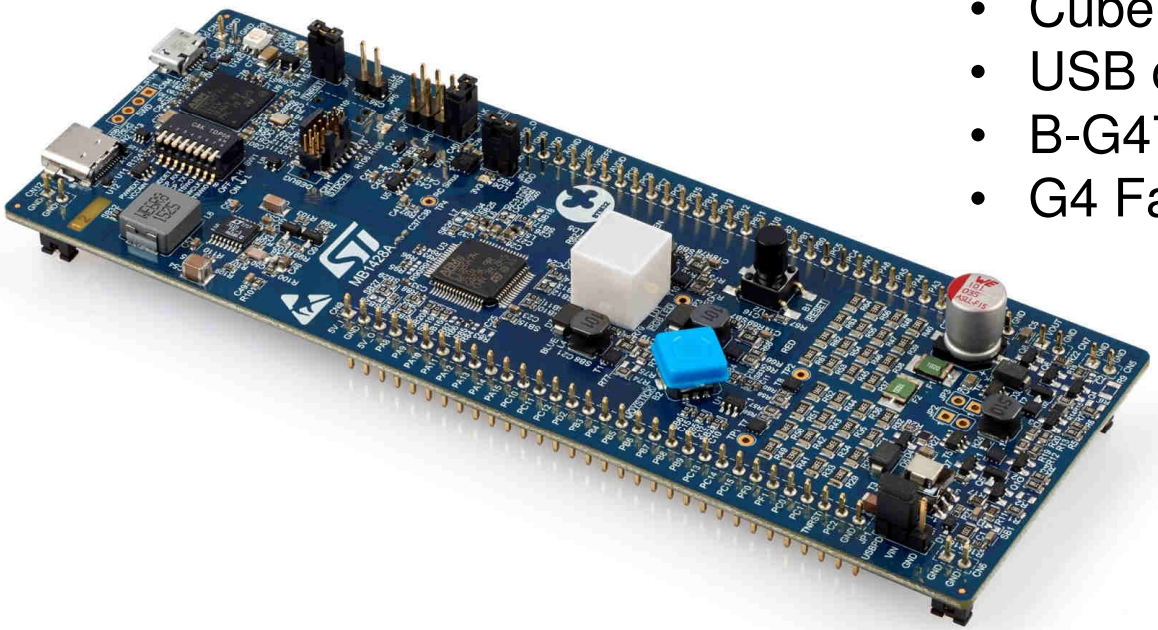
STM32G4 Mainstream Series Mixed Signal MCU

Demo Hands On



life.augmented

Systems Check



- Cube Programmer
- USB cable
- B-G474E-DPOW1 Discovery Board
- G4 Family Cube Library

B-G474E-DPOW1

Jumpers – 1 of 2

Do not attach USB cable here

CN2
USB Type-C™ receptacle

All switches Down

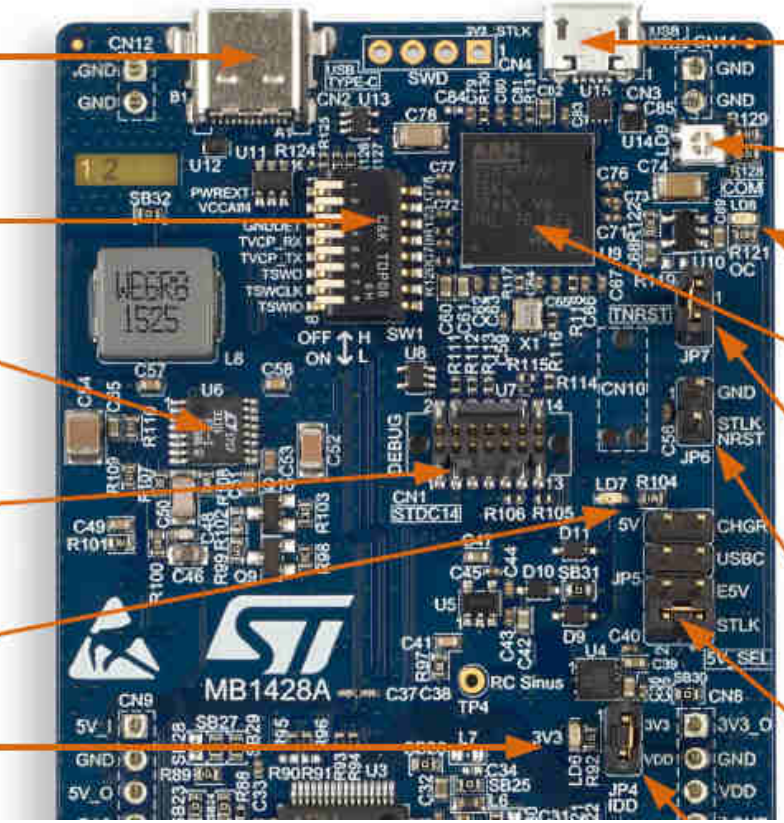
SW1
Octal mechanical switch

U6
LTC3114-1

CN1
STDC14 debug connector

LD7
5V LED

LD6
3V3 LED



CN3
USB micro-B ST-LINK receptacle

Attach USB cable here

LD9
COM LED

LD8
overcurrent LED

U9
STLINK-V3E MCU

JP7
TNRST

INSTALLED

JP6
STLK NRST

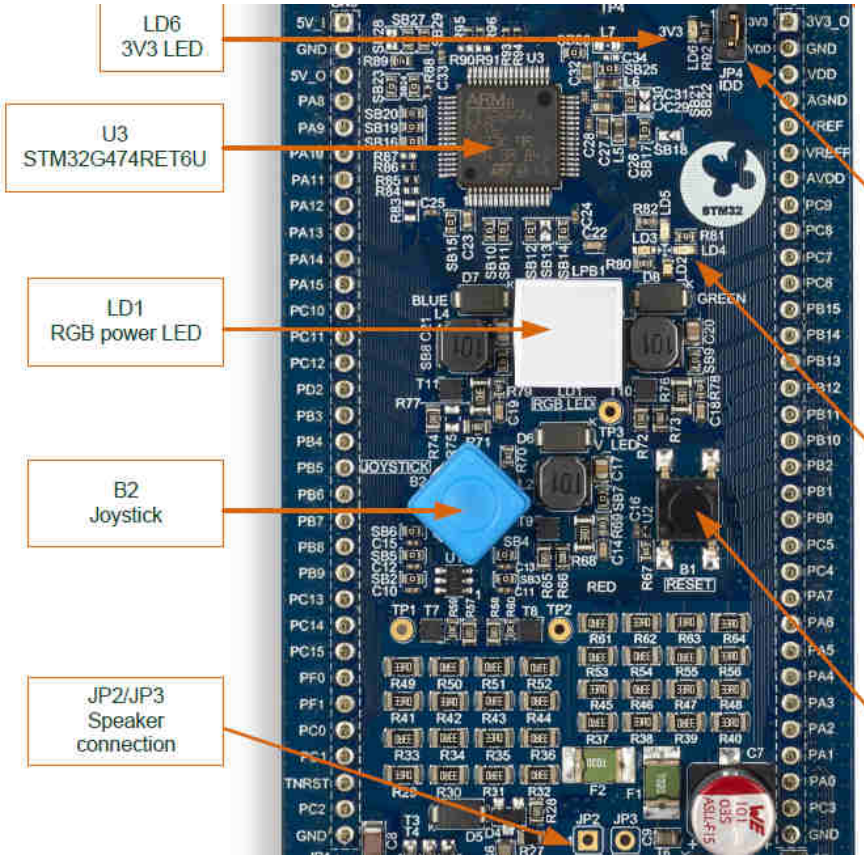
OPEN

JP5
5V_SELECTION

INSTALLED on STLK

life.augmented

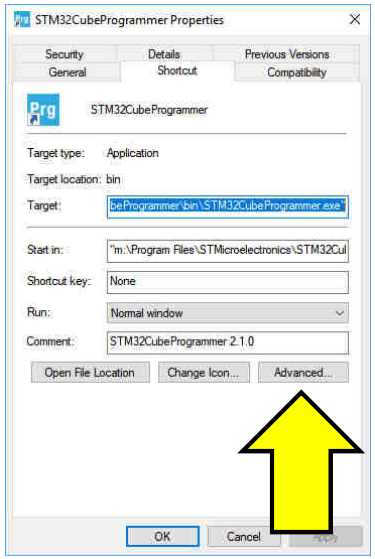
Jumpers – 2 of 2

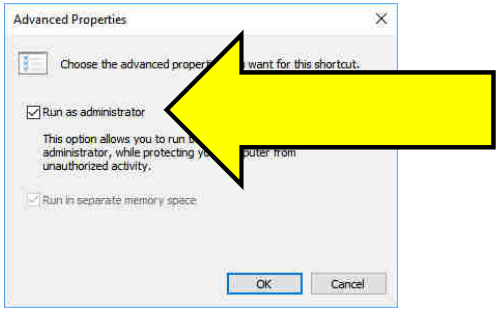


INSTALLED

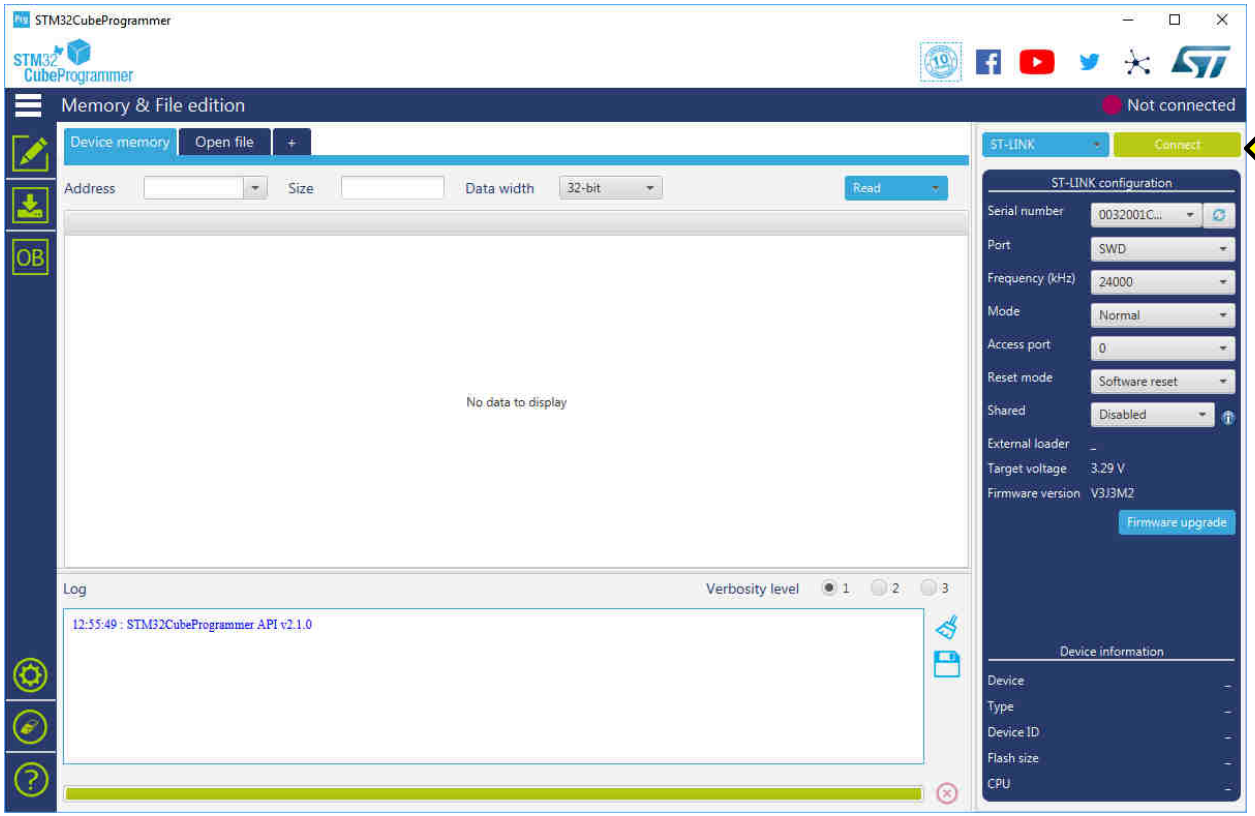
CubeProgrammer

Start up



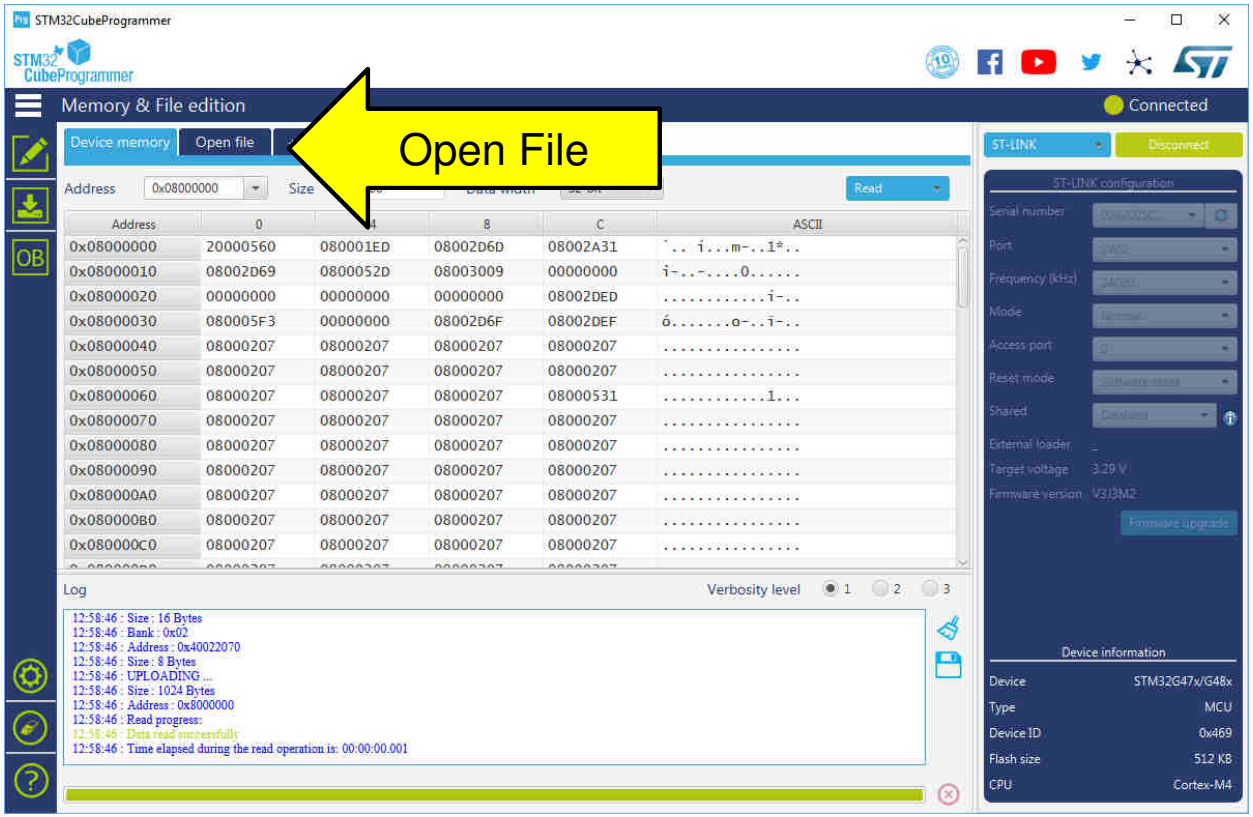


Splash Page - Connect

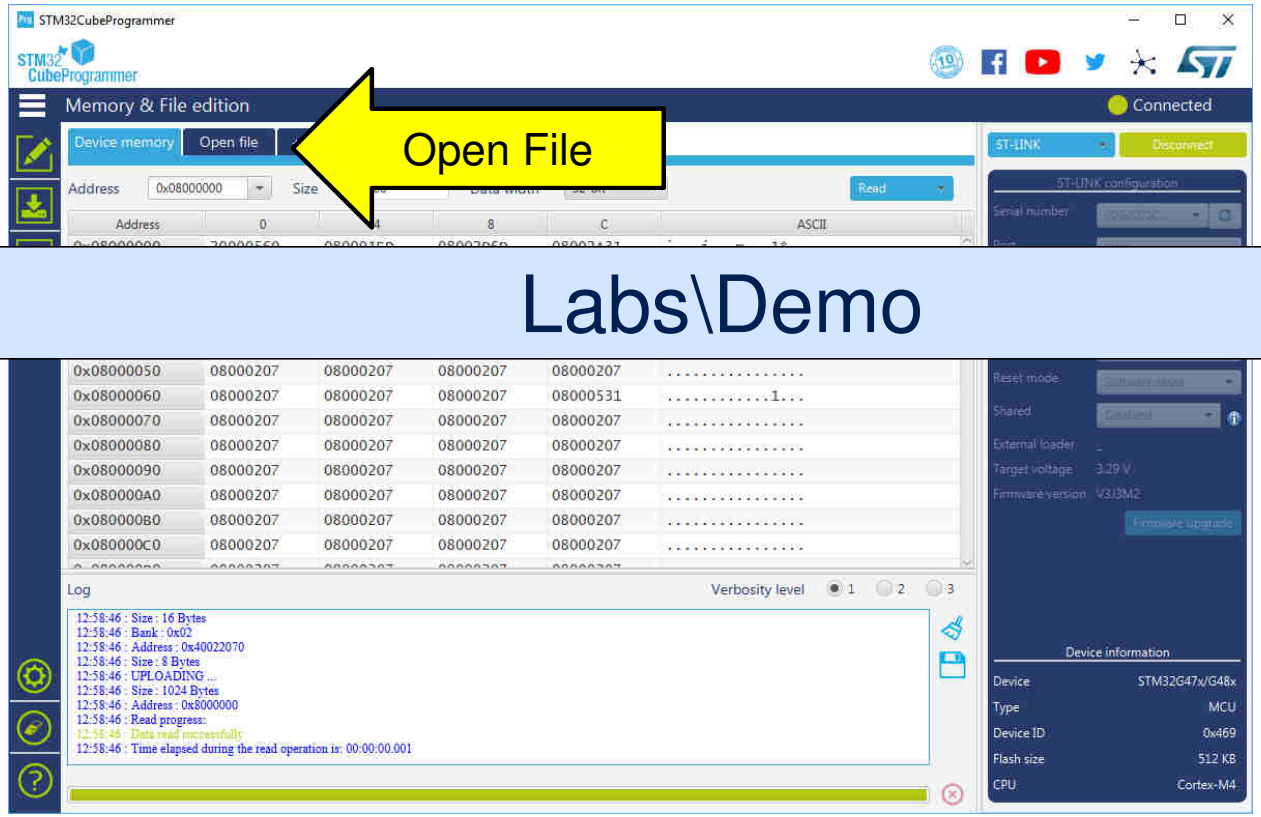


Connect

Splash Page – Open File

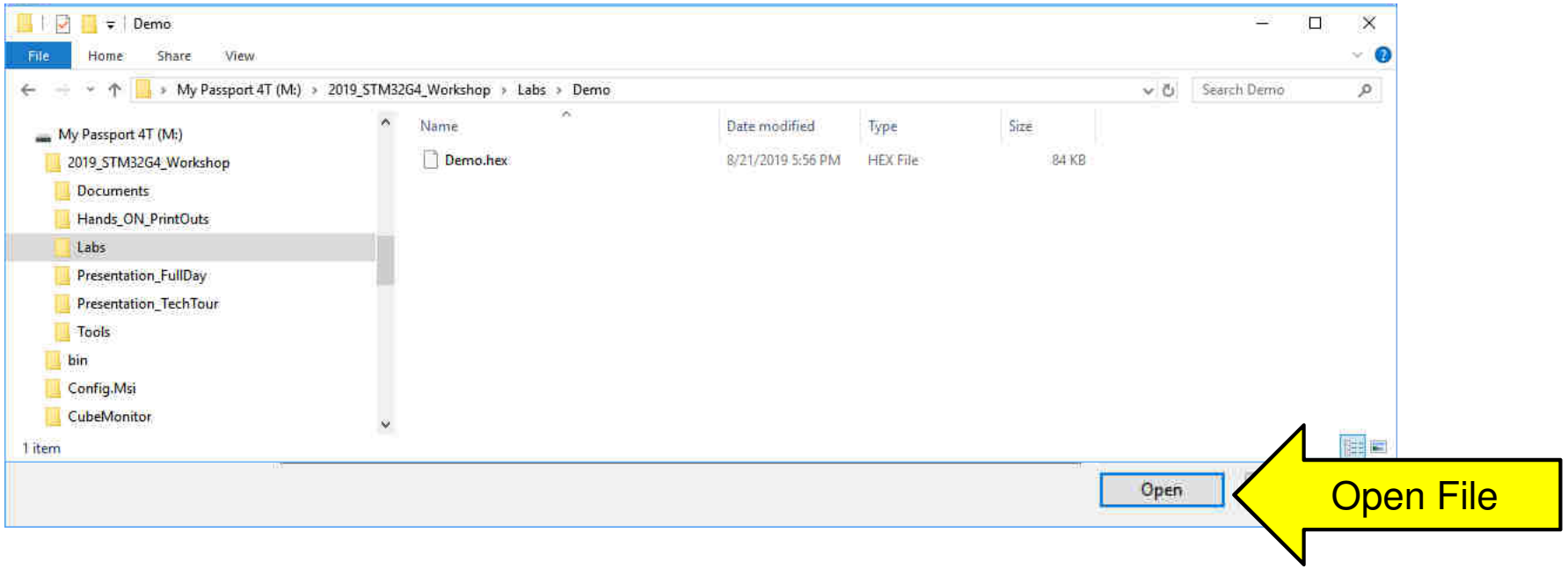


Find Demo.hex



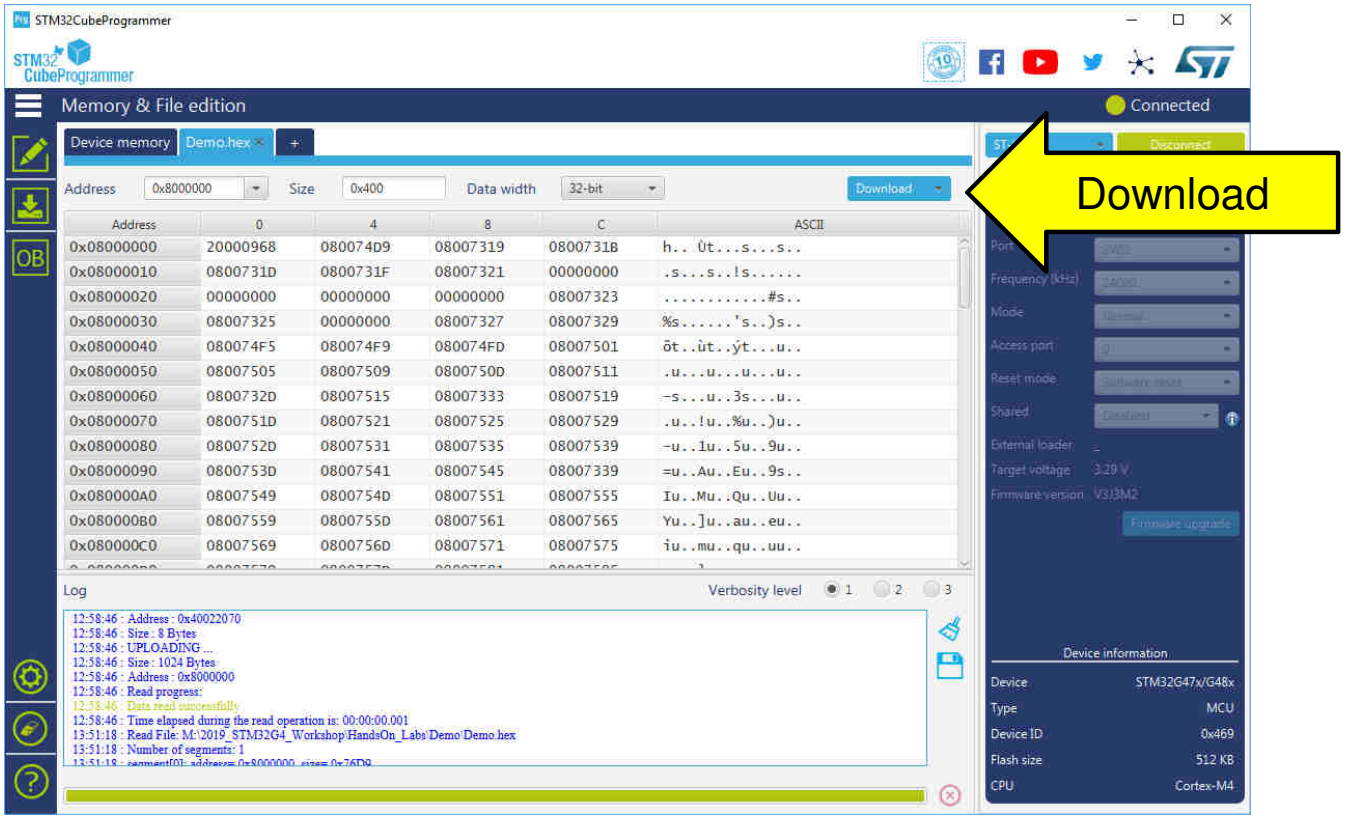
Labs\Demo

File Folder

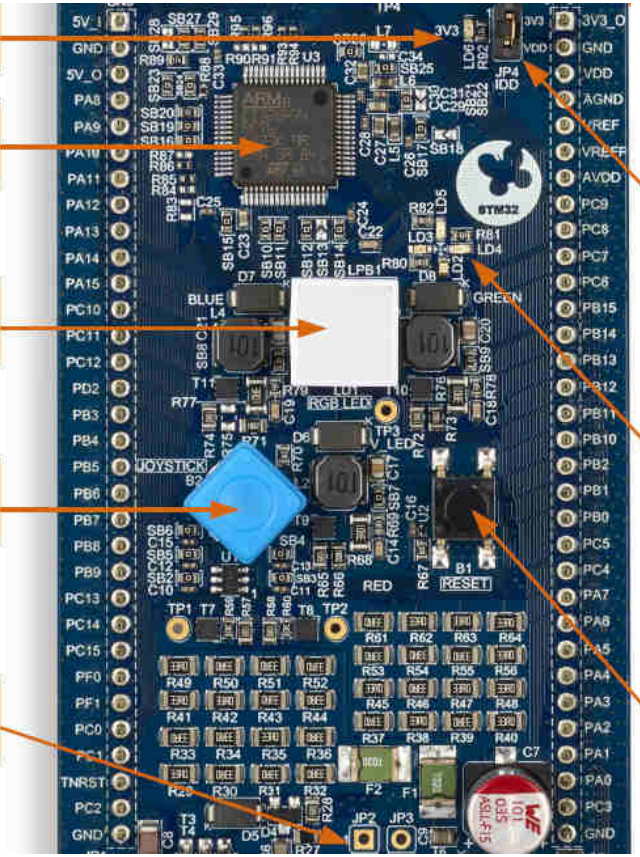

Demo
File
Home
Share
View
My Passport 4T (M:) › 2019_STM32G4_Workshop › Labs › Demo
Search Demo
My Passport 4T (M:)
2019_STM32G4_Workshop
Documents
Hands_ON_PrintOuts
Labs
Presentation_FullDay
Presentation_TechTour
Tools
bin
Config.Msi
CubeMonitor
Name
Date modified
Type
Size
Demo.hex
8/21/2019 5:56 PM
HEX File
84 KB
1 item
Open
Open File

CubeProgrammer

Move Demo.hex from the PC to the Discovery board



Press RESET (Black Button) – LED's cycle



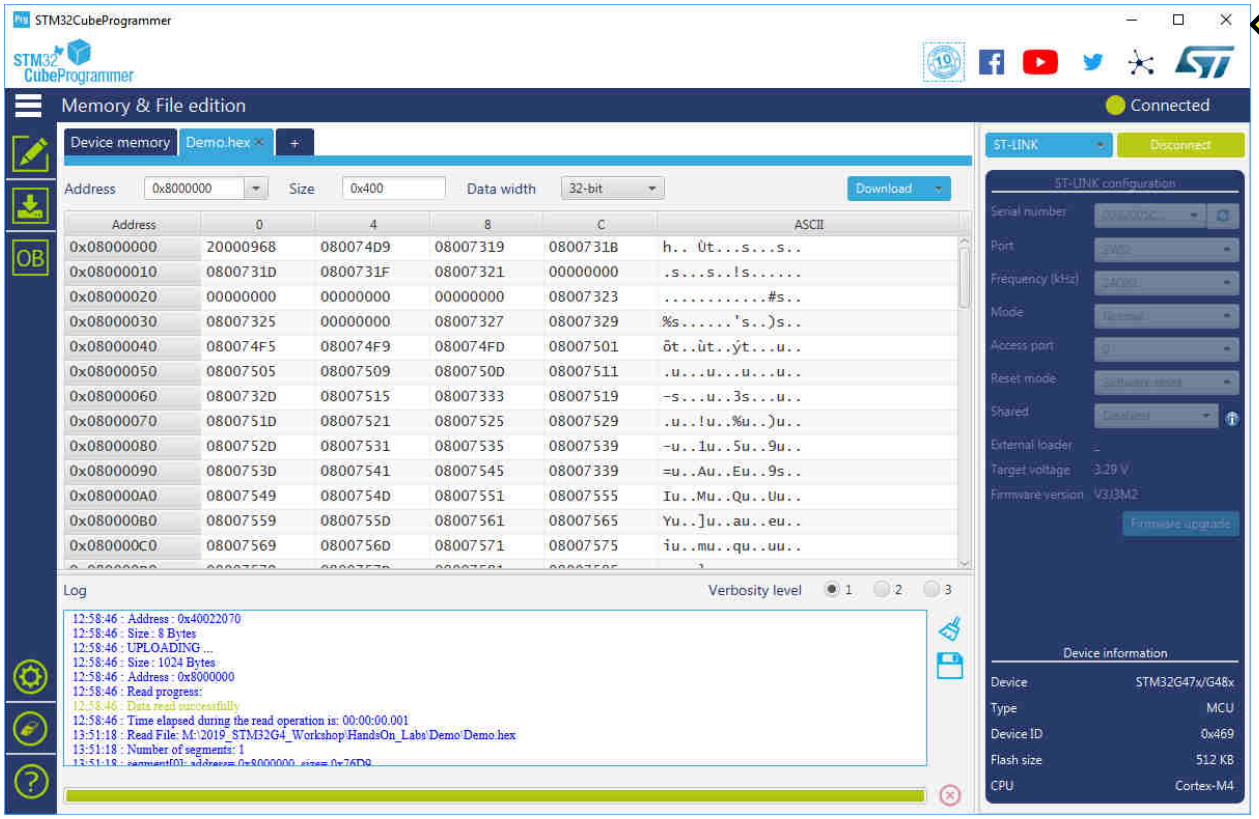
LED's Cycle

PRESS Quickly

life.augmented

CubeProgrammer

Shut down the CubeProgrammer to release the VCOM port……



Exit

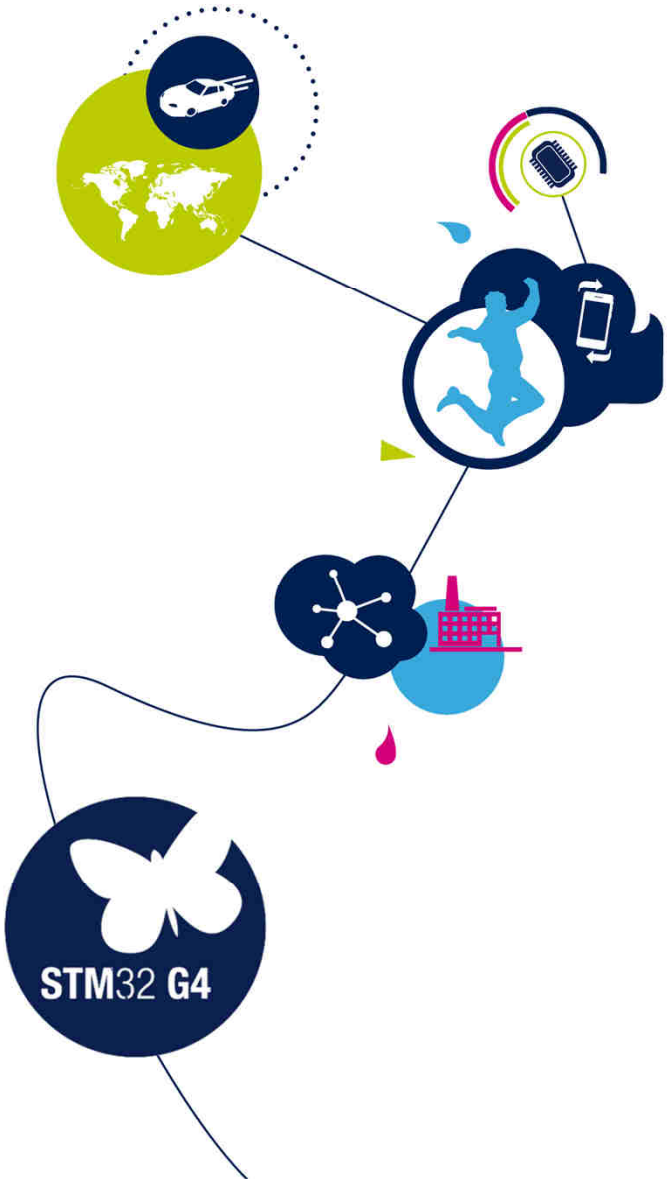
Getting back home – “CompletedLab”

For all of today’s Hands On Labs:

1. The *.hex file is available to you as a final product of the lab if you get lost along the way and still want to see the result of the lab.
2. The *.uvprojx file is available to you as a final result in case you want to pull it up in KEIL MDK-ARM and just walk through the files you see there.
3. The final STM32CubeMx *.ioc file for each lab is made available to you if you want to see the finished product of the lab or if you want to extend the lab with your own design ideas.

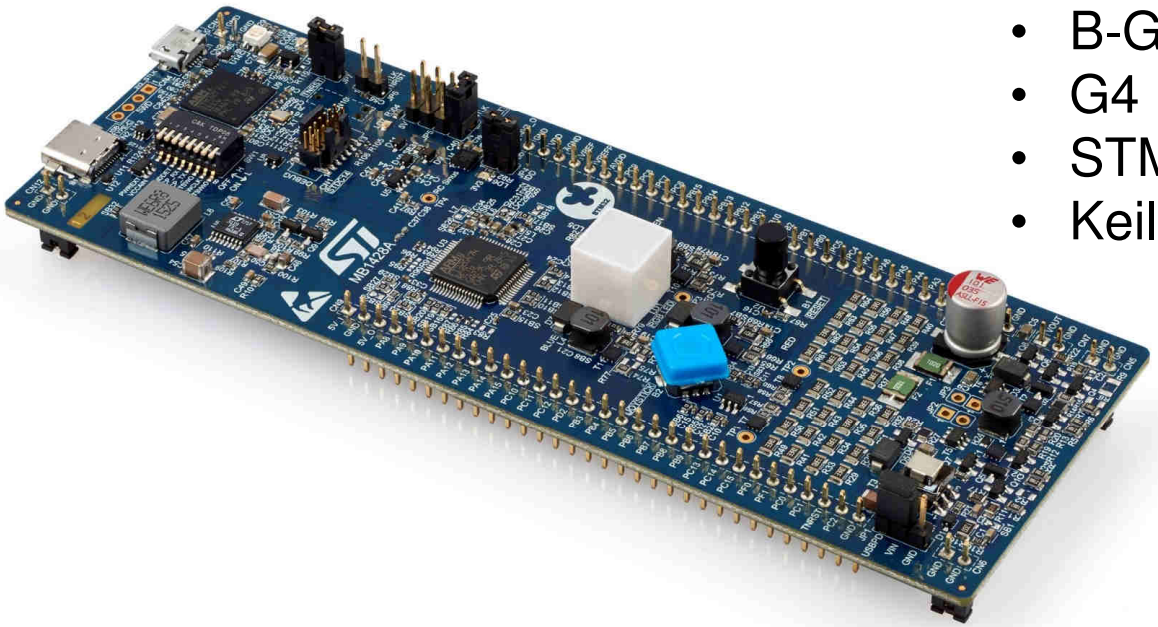
STM32G4 mixed-signal MCU workshop

Hands On:
DAC Signal Generation



life.augmented

Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM

Rich, Advanced Analog

Mixed-signal SoC for wide variety of applications

ADC (up to 5)	**Values**
Topology	**SAR 12-bit** + HW oversampling → **16-bit**
Sampling rate	Up to **4 Msps**
Input	Single-ended and differential
Offset and Gain compensation	Auto calibration to reduce gain and offset

DAC (up to 7)	**Values**
Sampling rate	**15 Msps** (internal) 1Msps (from buffered output)
Settling time	16ns

Op-Amp (up to 6)	**Values**
GBW	**13 MHz**
Slew rate	**45 V/µs**
Offset	3mV over full T° range 1.5mV @ 25°C
PGA Gain (accuracy)	2, 4, 8, 16, -1,-3,-7,-15 **(1%)** 32, 64, -31,-63 (2%)

Comparator (up to 7)	**Values**
Power supply	1.62 .. 3.6V
Propagation delay	**16.7ns**
Offset	-6 .. +2 mV
Hysteresis	8 steps: 0, 9, 18, 27, 36, 45, 54, 63 mV

life.augmented

Digital to Analog Converter

G4 DAC

- 3 external outputs:
  - DAC1-OUT1 and OUT2, 1Msps
- 4 internal outputs:
  - DAC3 and DAC4
  - OUT1 and OUT2, 15Msps
- Double data DMA
- Sample & Hold
- Left or Right data alignment
- Signed or unsigned data input format {Q1.15, Q1.11, Q1.7}

Internally connected to:

- Voltage reference
- External Triggers
- Comparator inputs
- Op Amp inputs
- Multiple Timers (+ HRTimer)
- DMA

Digital to Analog Converter

G4 DAC Sample and Hold

DAC conversion during the “sample and hold“ mode has three phases:
1. Sampling phase: the “sample and hold“ element is charged with the desired voltage.
2. Holding phase: the DAC output is tri-stated (High-Z) to maintain the “sample and hold“ element’s stored
electrical charge.
3. Refresh phase: due to leakage coming from several sources, a refresh phase is essential to maintain the output voltage at the desired value (+/-LSB).

Digital to Analog Converter

Dual Conversion from DMA

We will use the DAC dual channel mode with DMA to generate signals on both DAC channels at the same time.

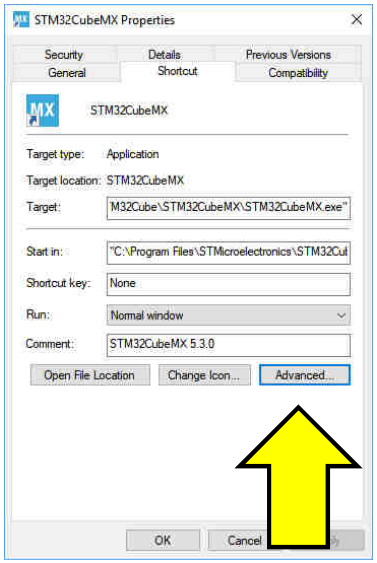
DAC conversions are made with a timer trigger and data fed from DMA.

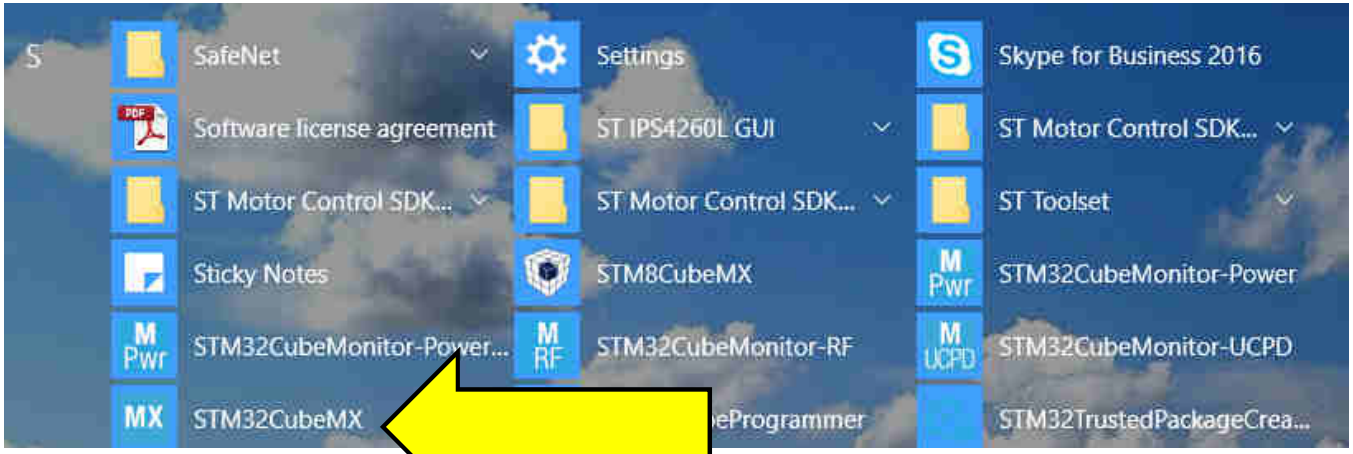
The timer is configured to trigger the DAC channels at about 1kHz.
DMA is feeding both channels at the time.
Channel 1 pattern is ~0V/~1.65V/~3.3V/~1.65V
Channel 2 pattern is ~1.65V/~3.3V/~1.65V/~0V

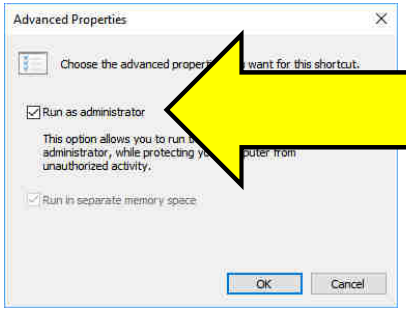
We will use the built in Logic Analyzer today instead of an o'scope on the DACs

Start up

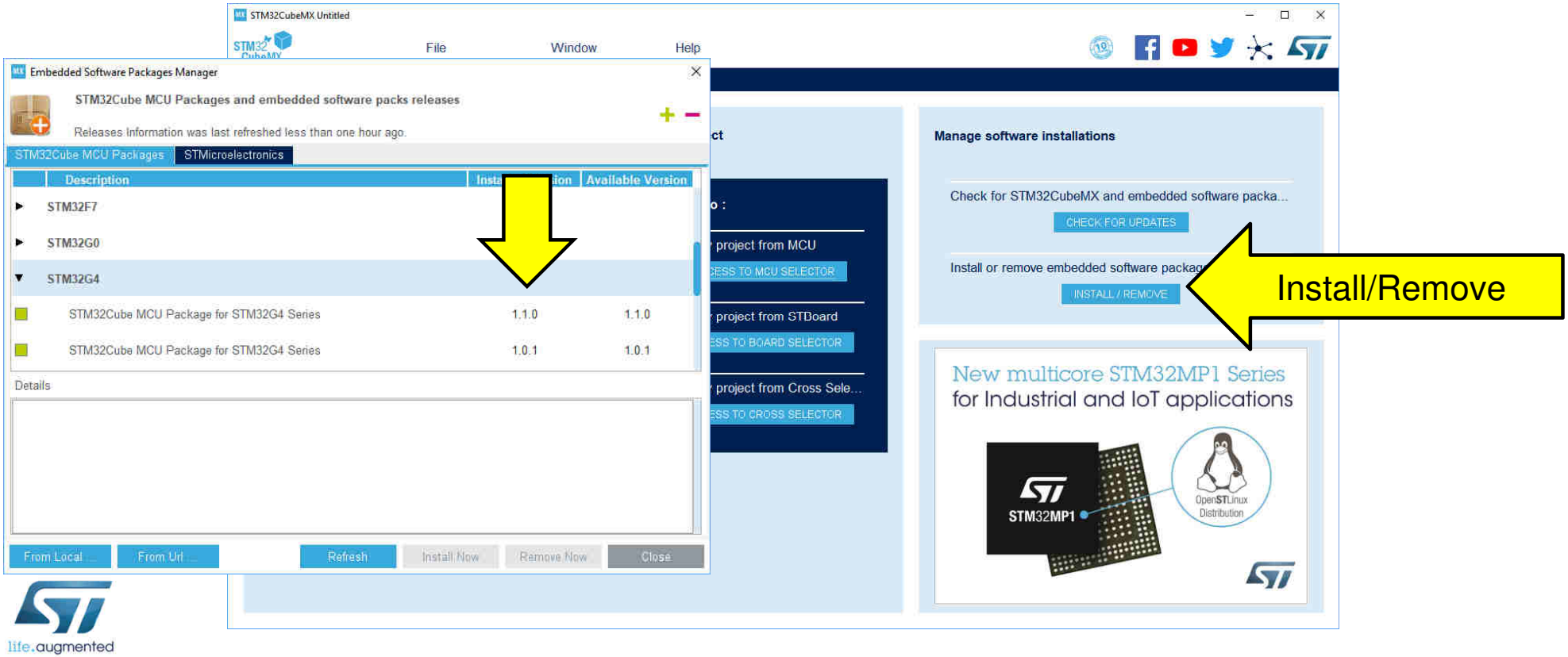

STM32Cu...


STM32CubeMX Properties
Security
Details
Previous Versions
General
Shortcut
Compatibility
STM32CubeMX
Target type: Application
Target location: STM32CubeMX
Target: "M32Cube\STM32CubeMX\STM32CubeMX.exe"
Start in: "C:\Program Files\STMicroelectronics\STM32Cul
Shortcut key: None
Run: Normal window
Comment: STM32CubeMX 5.3.0
Open File Location
Change Icon...
Advanced...
OK
Cancel


S
SafeNet
Settings
Skype for Business 2016
Software license agreement
ST IPS4260L GUI
ST Motor Control SDK...
ST Motor Control SDK...
ST Motor Control SDK...
ST Toolset
Sticky Notes
STM8CubeMX
STM32CubeMonitor-Power
STM32CubeMonitor-Power...
STM32CubeMonitor-RF
STM32CubeMonitor-UCPD
STM32CubeMX
eProgrammer
STM32TrustedPackageCrea...

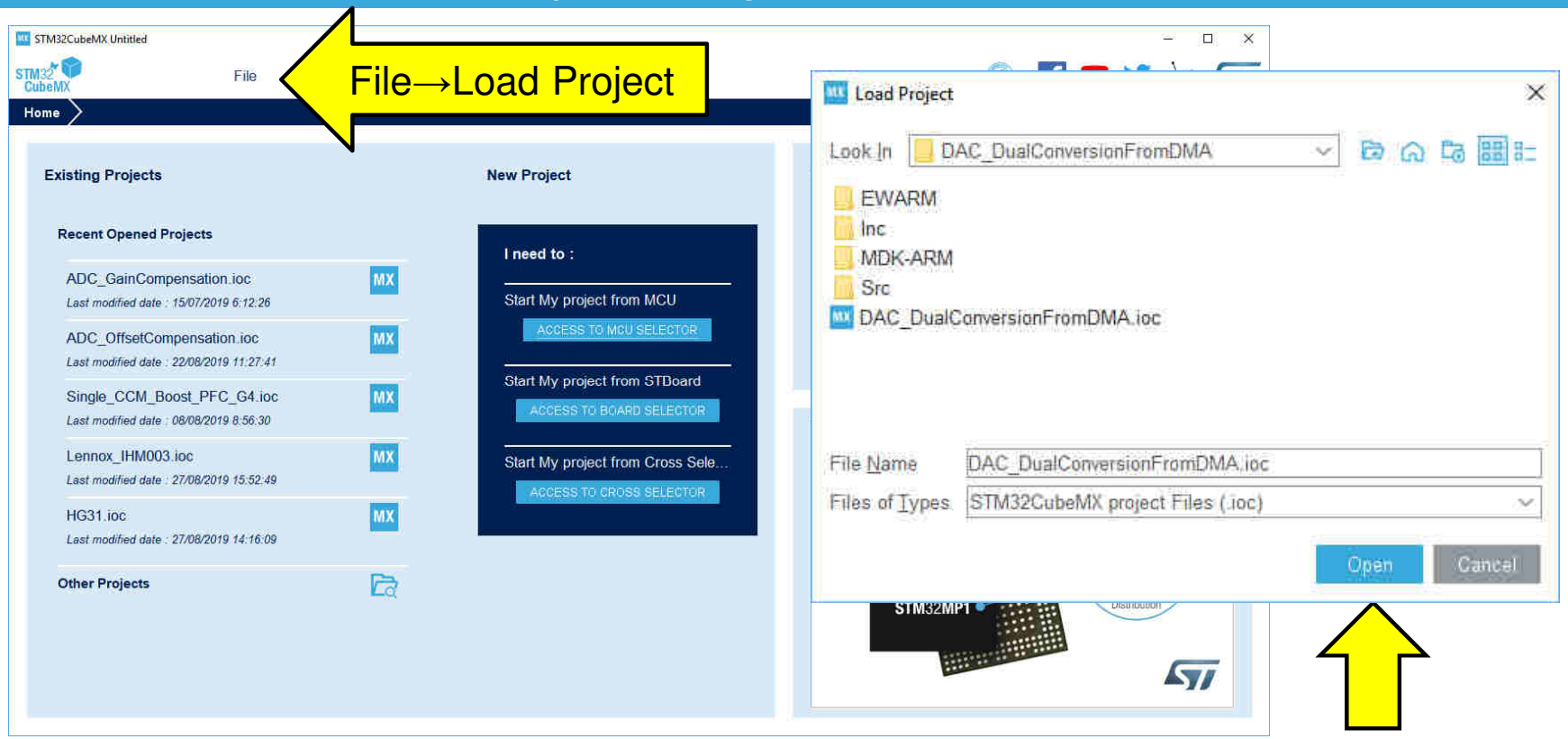

Advanced Properties
Choose the advanced properti want for this shortcut.
Run as administrator
This option allows you to run
administrator, while protecting yo puter from
unauthorized activity.
Run in separate memory space
OK
Cancel

Splash Page – G4 Library Installed?



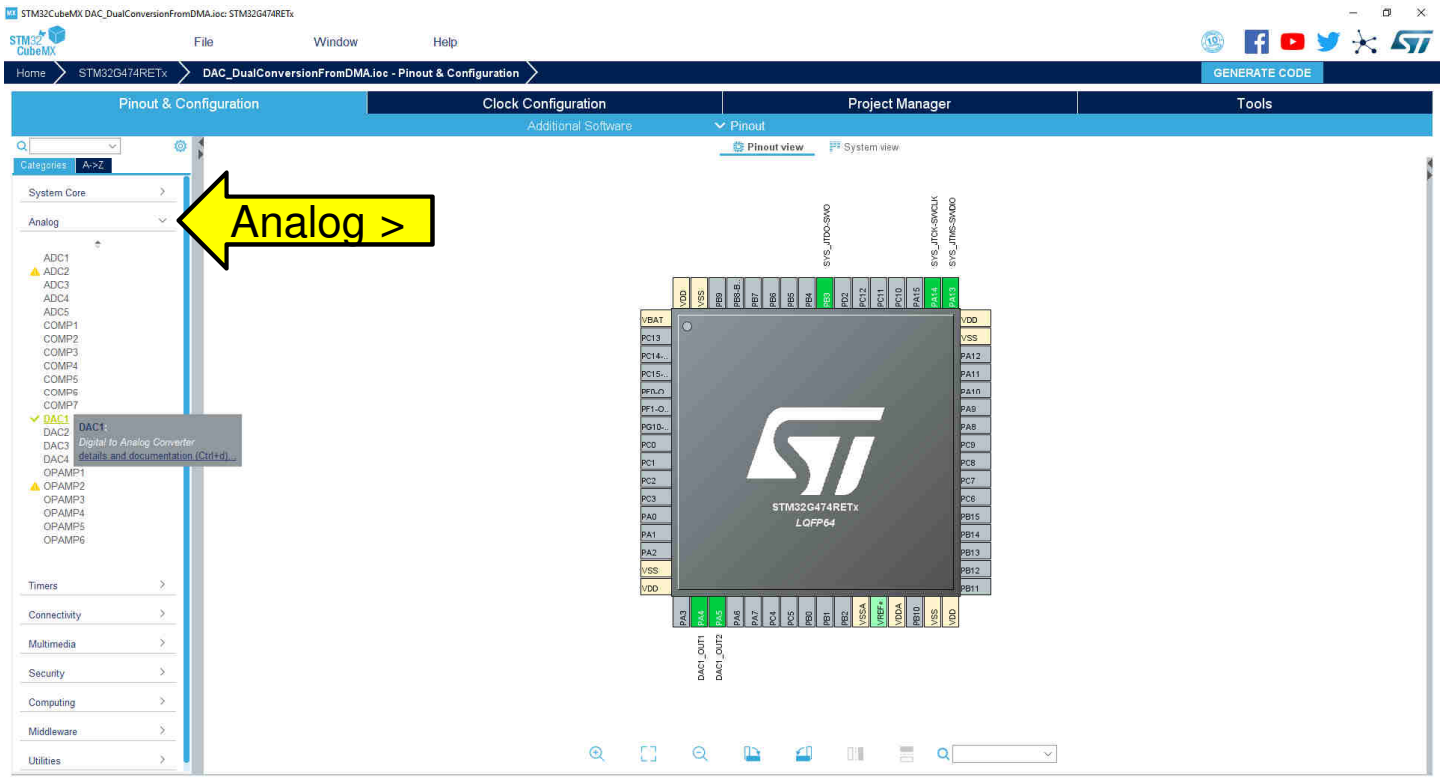
life.augmented

Splash Page



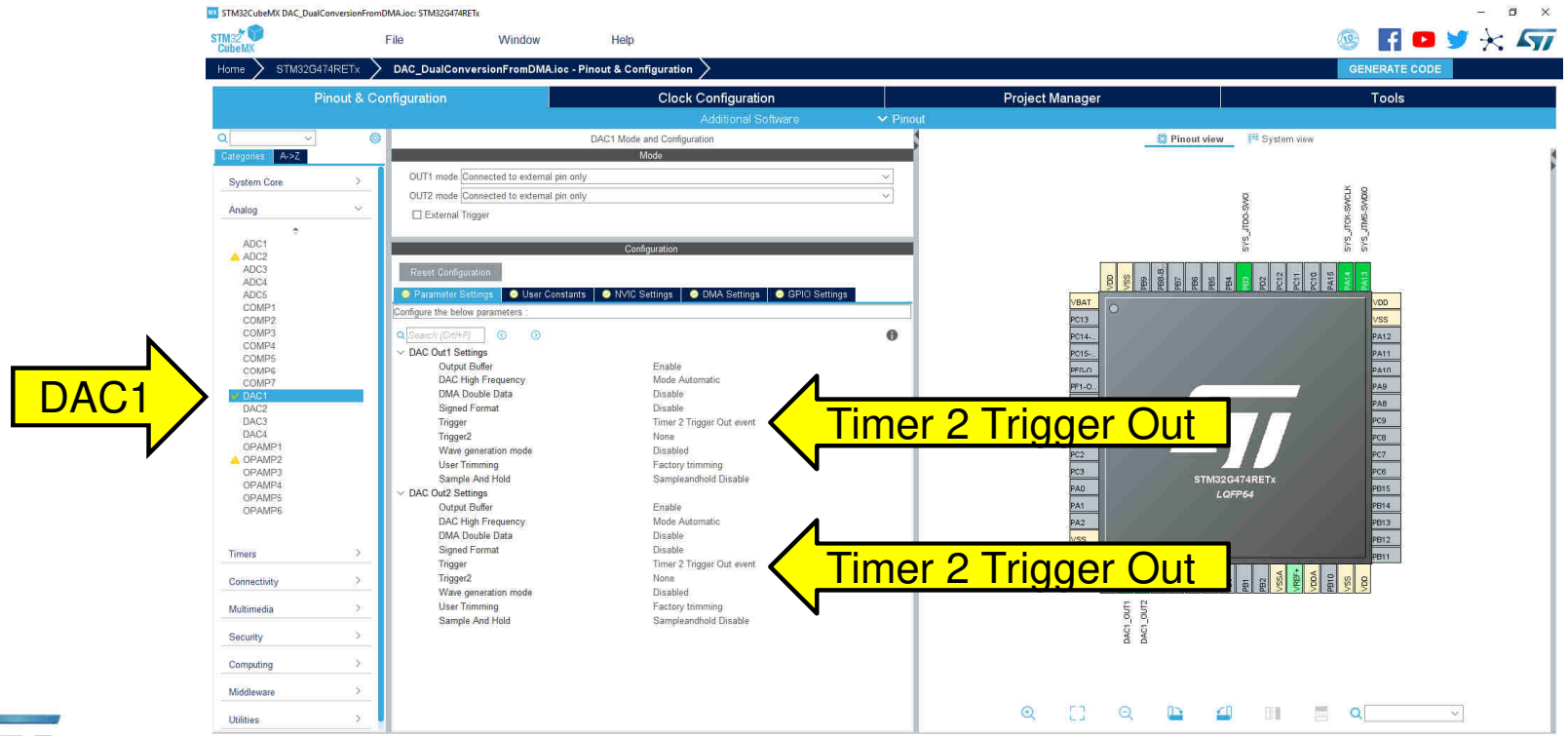
C:\2019_STM32G4_Workshop\Labs\DAC

Graphic Design Representation

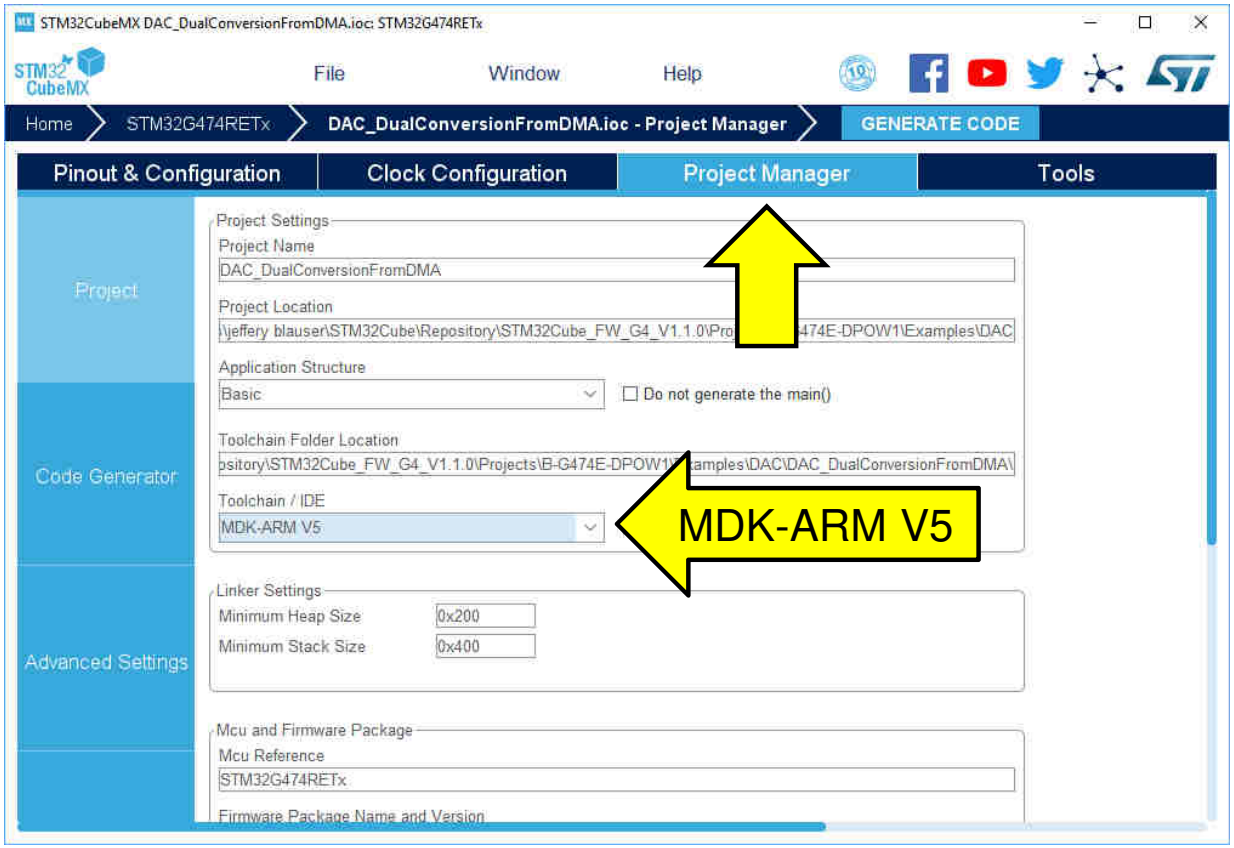


life.augmented

DAC1, Timer 2

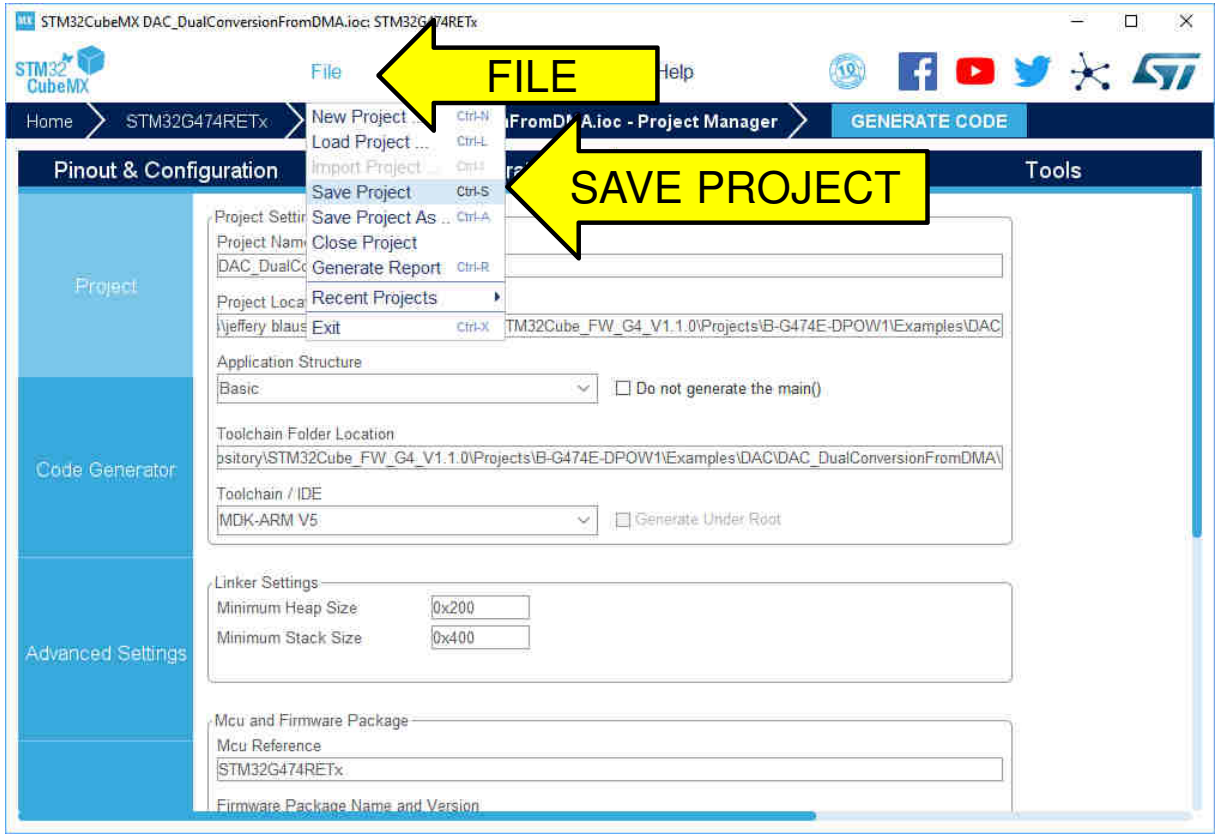

DAC1
Timer 2 Trigger Out
Timer 2 Trigger Out
STM32CubeMx DAC_DualConversionFromDMA.ioc: STM32G474RETx
File
Window
Help
Home
STM32G474RETx
DAC_DualConversionFromDMA.ioc - Pinout & Configuration
GENERATE CODE
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Additional Software
Pinout
DAC1 Mode and Configuration
Mode
OUT1 mode
Connected to external pin only
OUT2 mode
Connected to external pin only
External Trigger
Configuration
Reset Configuration
Parameter Settings
User Constants
NVIC Settings
DMA Settings
GPIO Settings
Configure the below parameters :
DAC Out1 Settings
DAC Out2 Settings
Output Buffer
DAC High Frequency
DMA Double Data
Signed Format
Trigger
Trigger2
Wave generation mode
User Trimming
Sample And Hold
Enable
Mode Automatic
Disable
Timer 2 Trigger Out event
None
Factory trimming
Sampleandhold Disable
Pinout view
System view
STM32G474RETx
LQFP64

Project Manager – Keil MDK-ARM V5

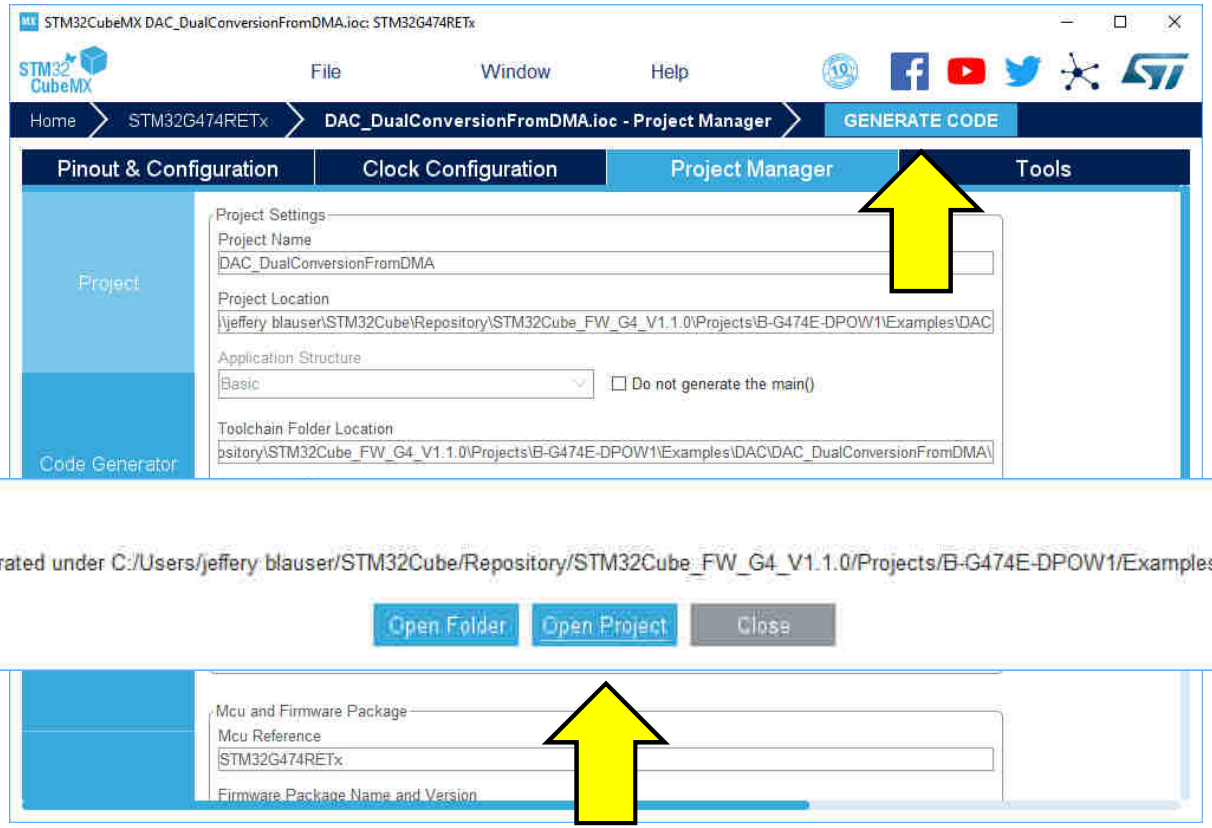


CubeProgrammer

FILE→SAVE PROJECT

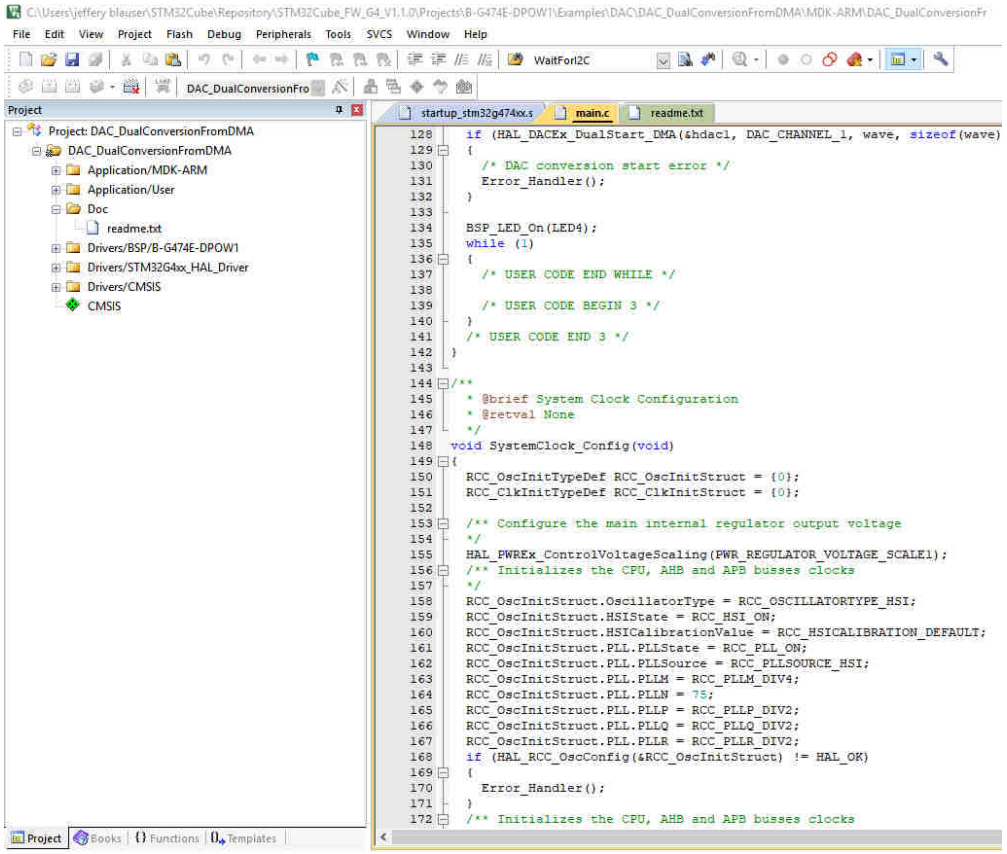


Generate Code

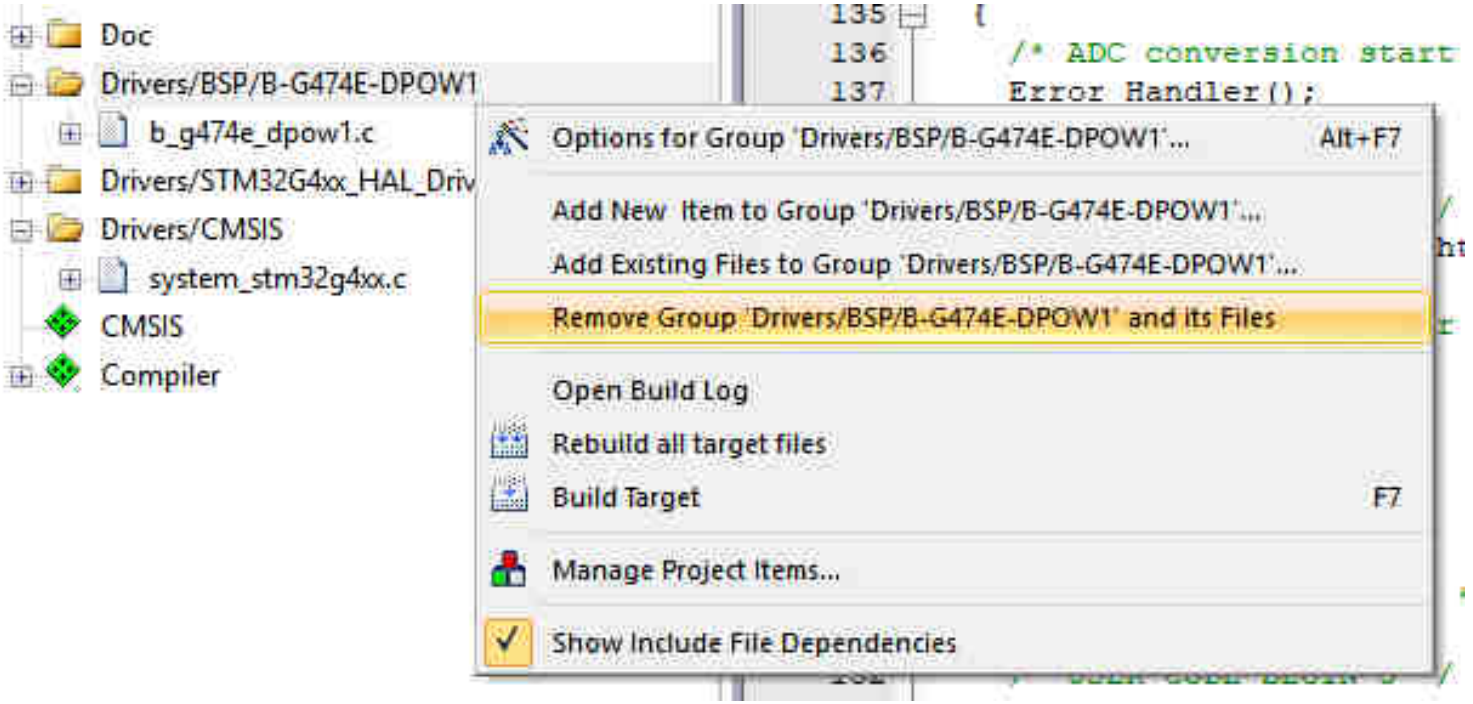


Keil MDK-ARM

Keil Start up page

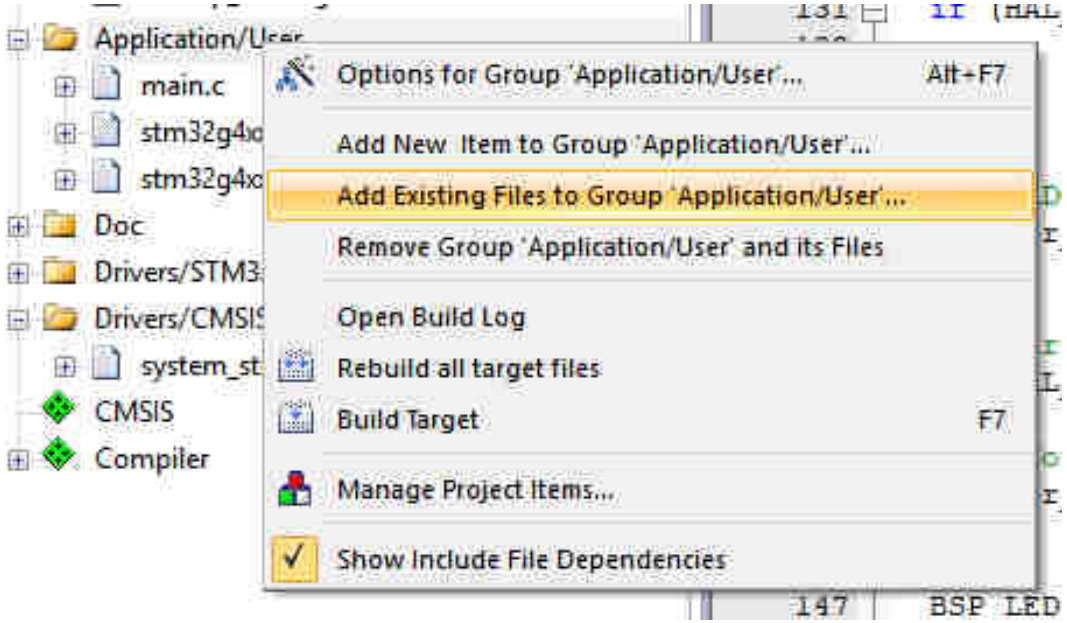


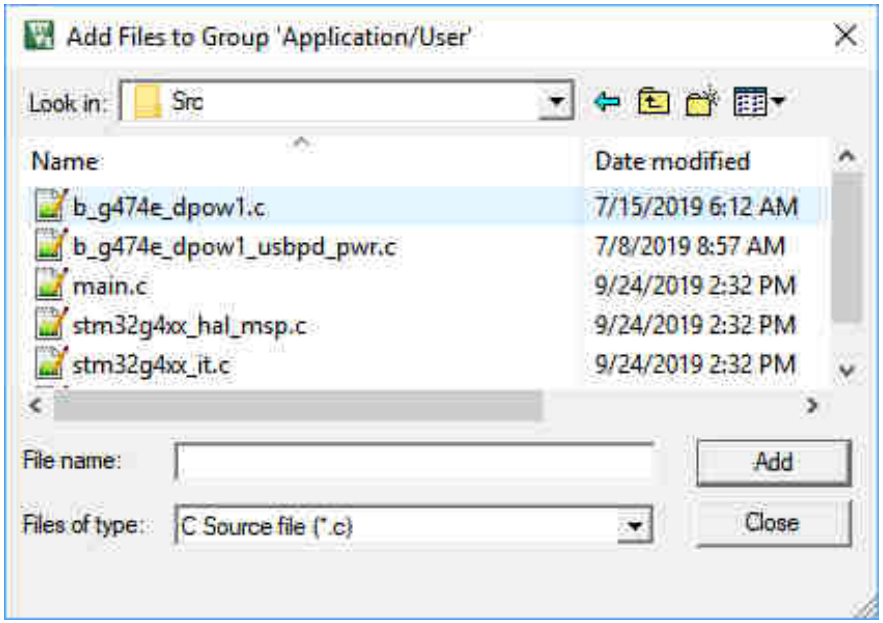
Delete Drivers Folder



Keil MDK-ARM

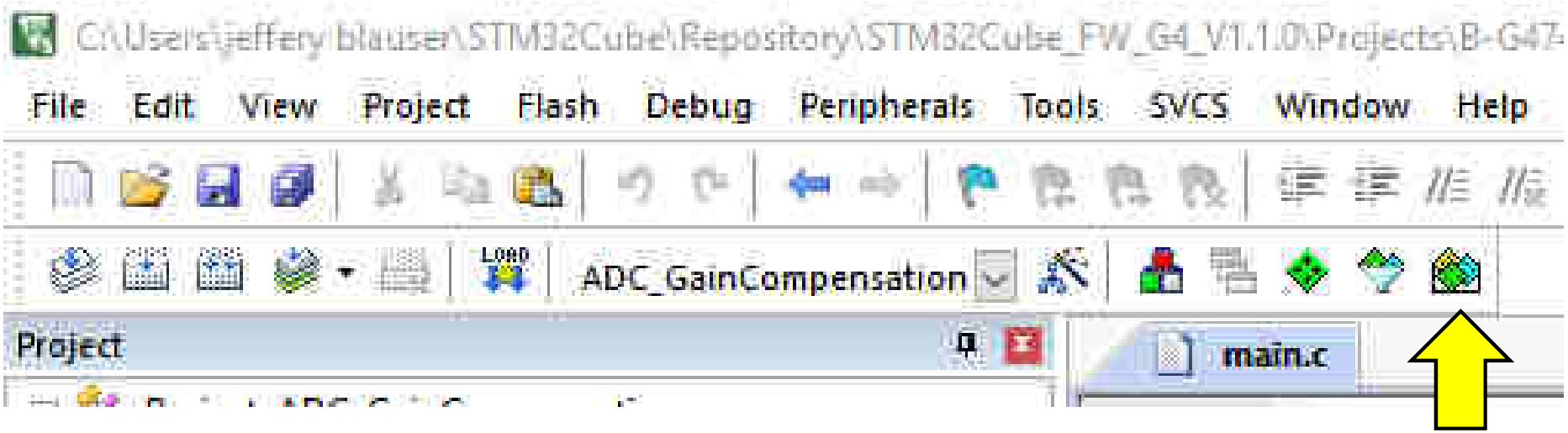
Application/User folder add b_g474e_dpow1.c from SRC folder - CLOSE





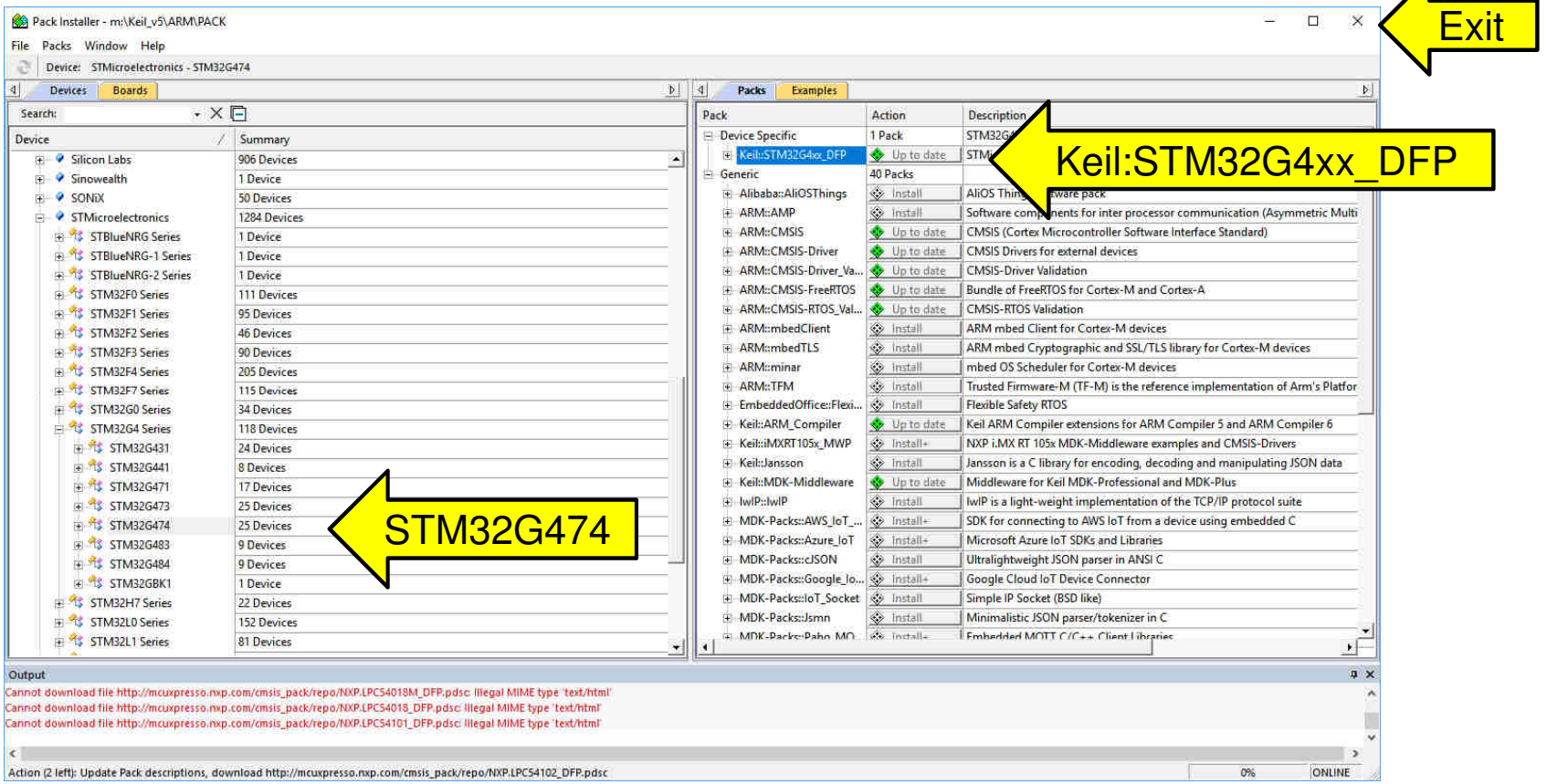
Keil MDK-ARM

Check the Device Pack



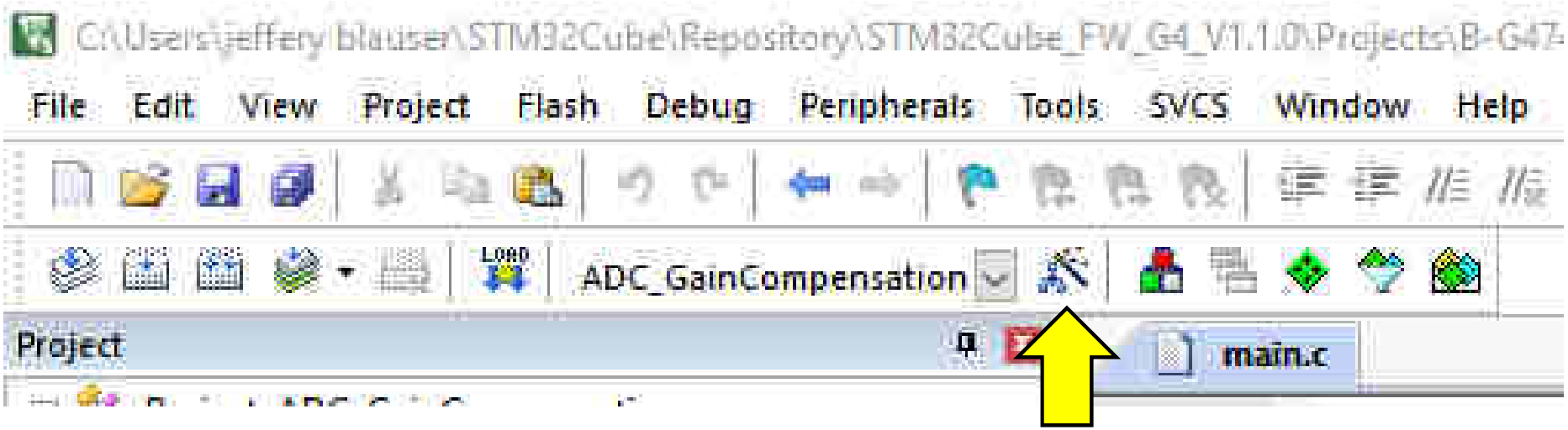
Keil MDK-ARM

G4 Device Pack



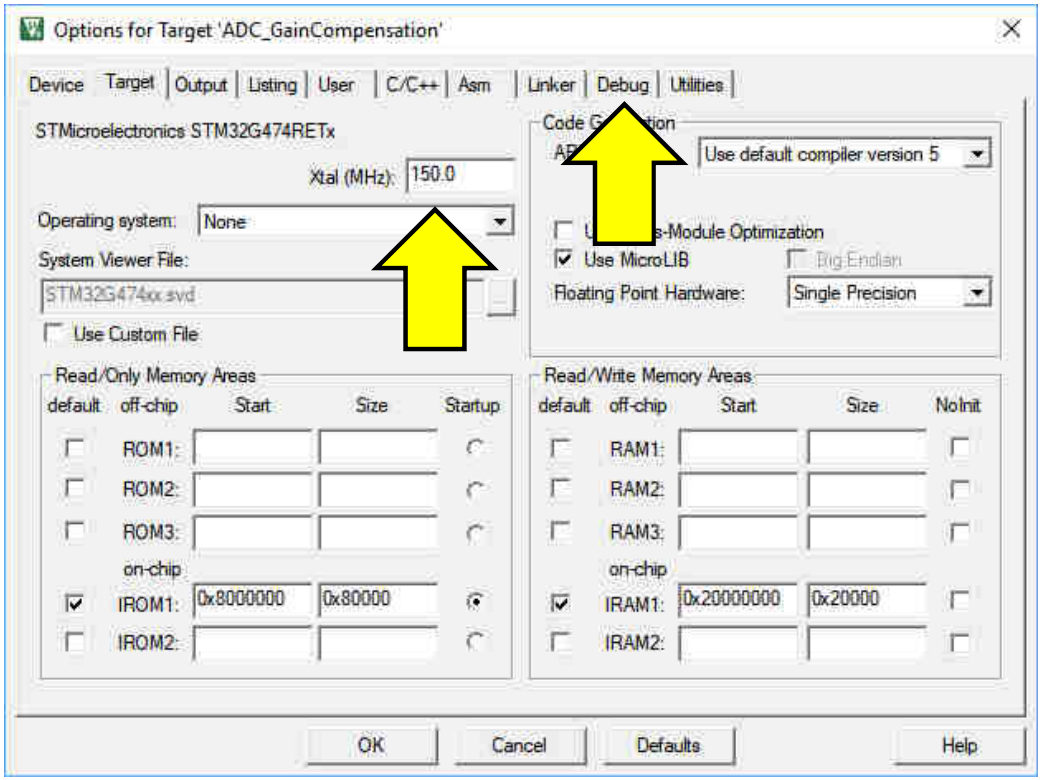
Keil MDK-ARM

Check the Options

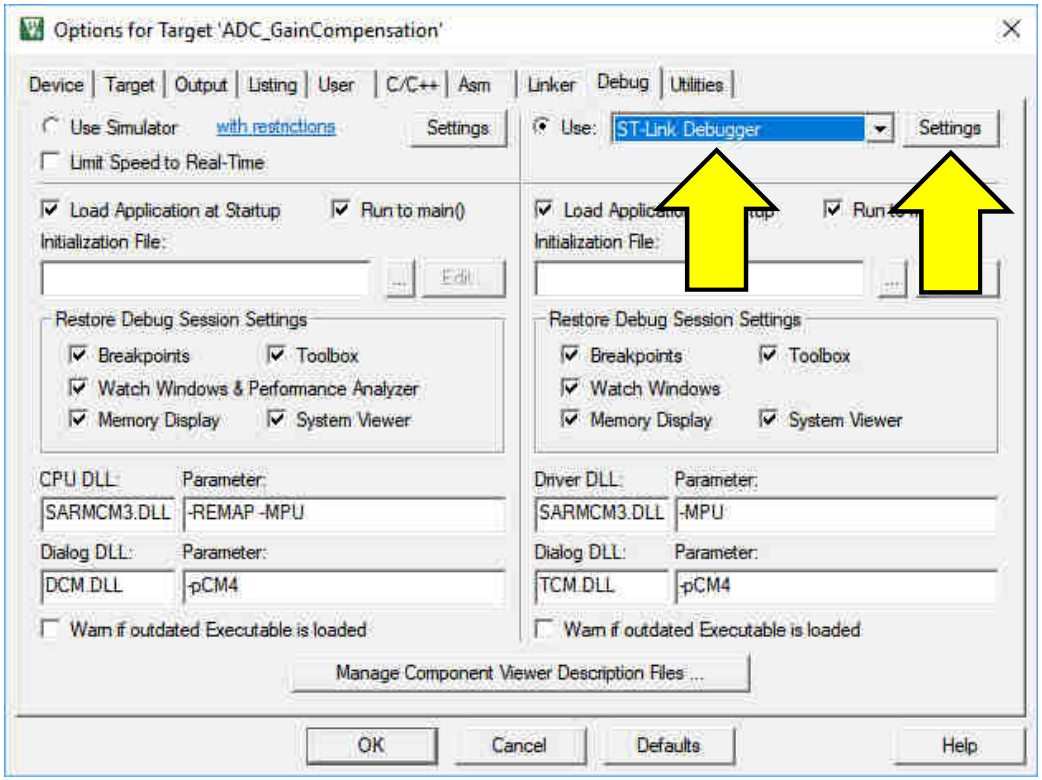


Keil MDK-ARM

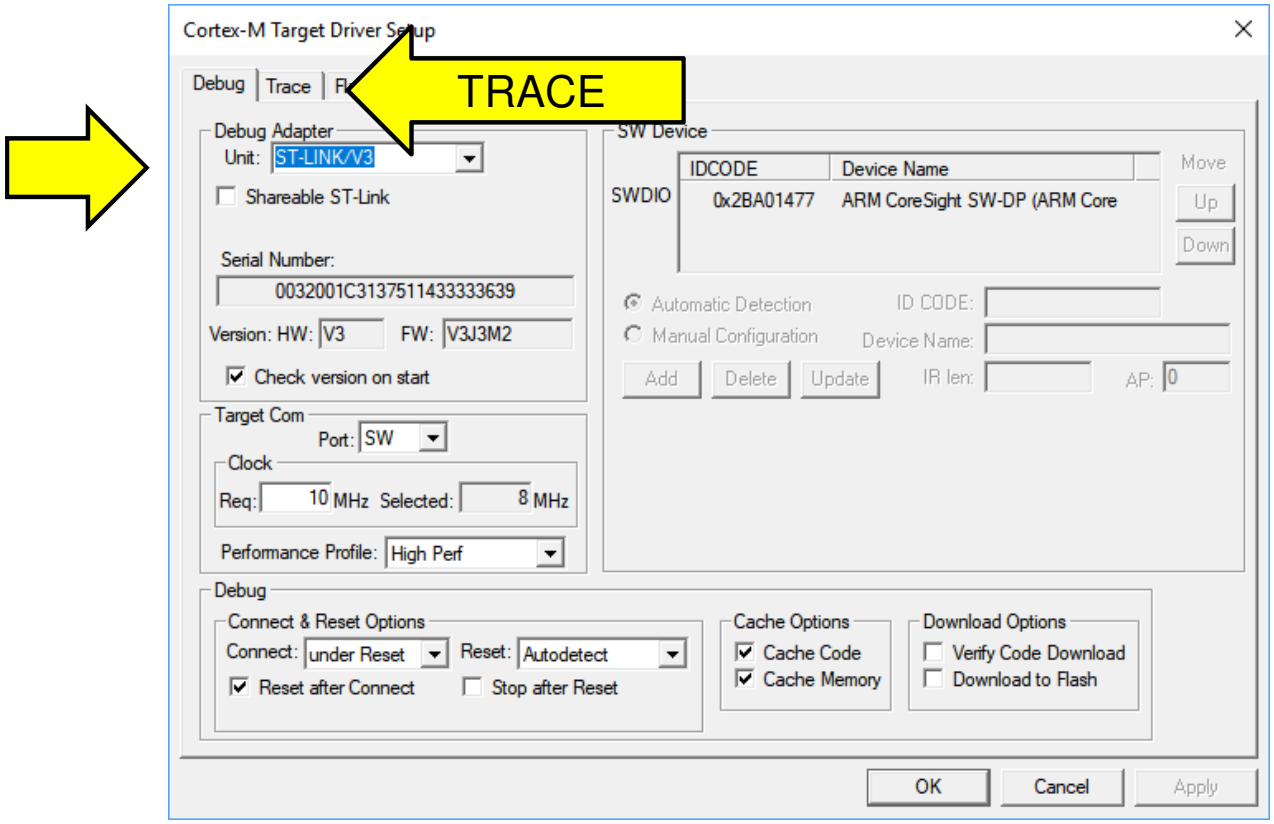
Clock speed and Debug



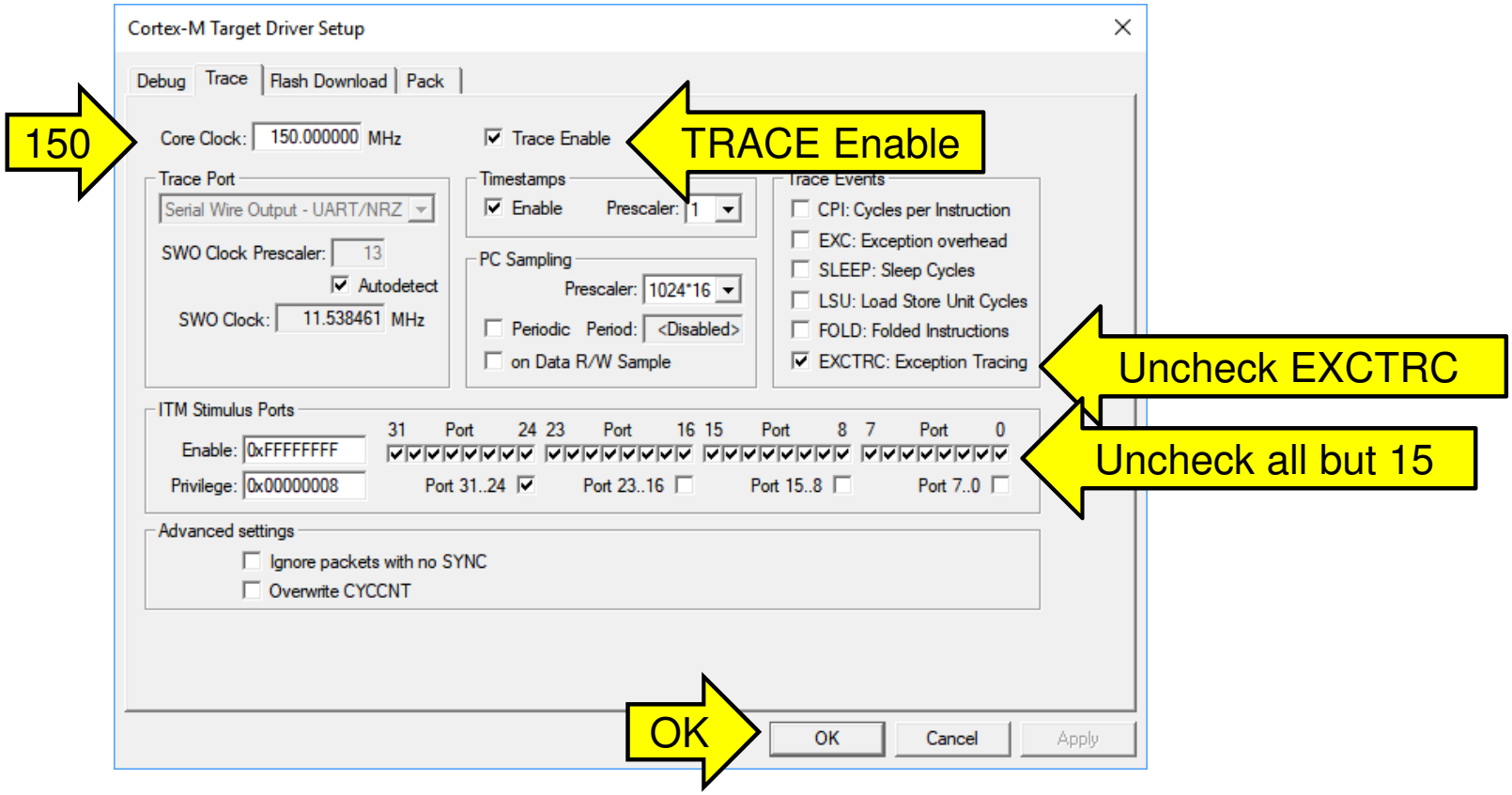
ST-Link Debugger - Settings



ST-Link/V3 & TRACE

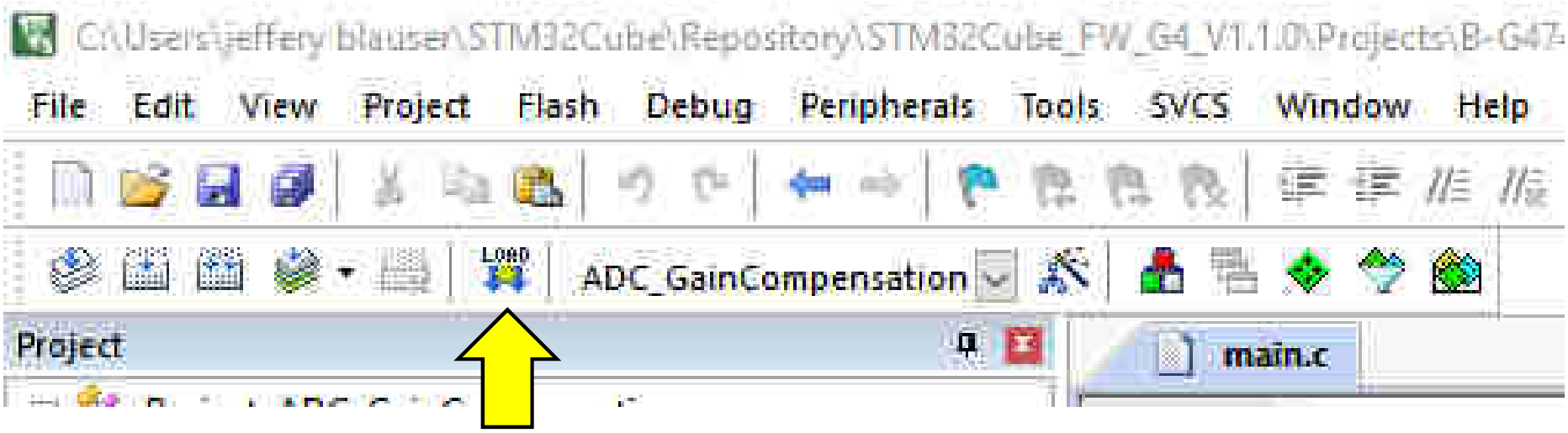


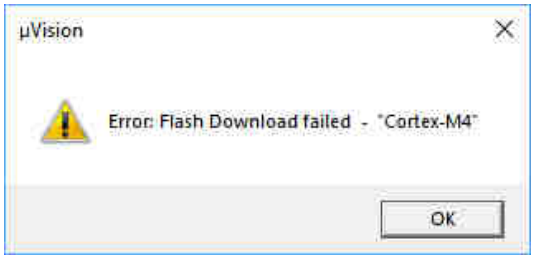
Core Clock & TRACE Enable

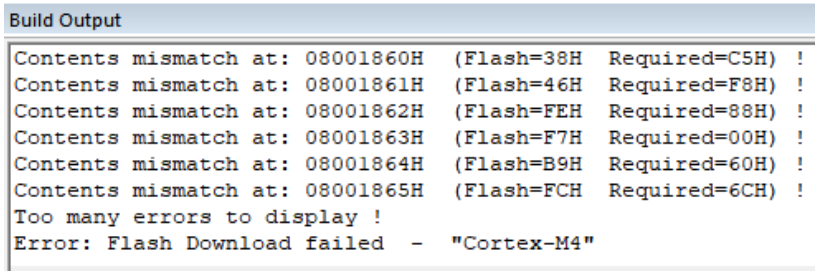


Keil MDK-ARM

Load

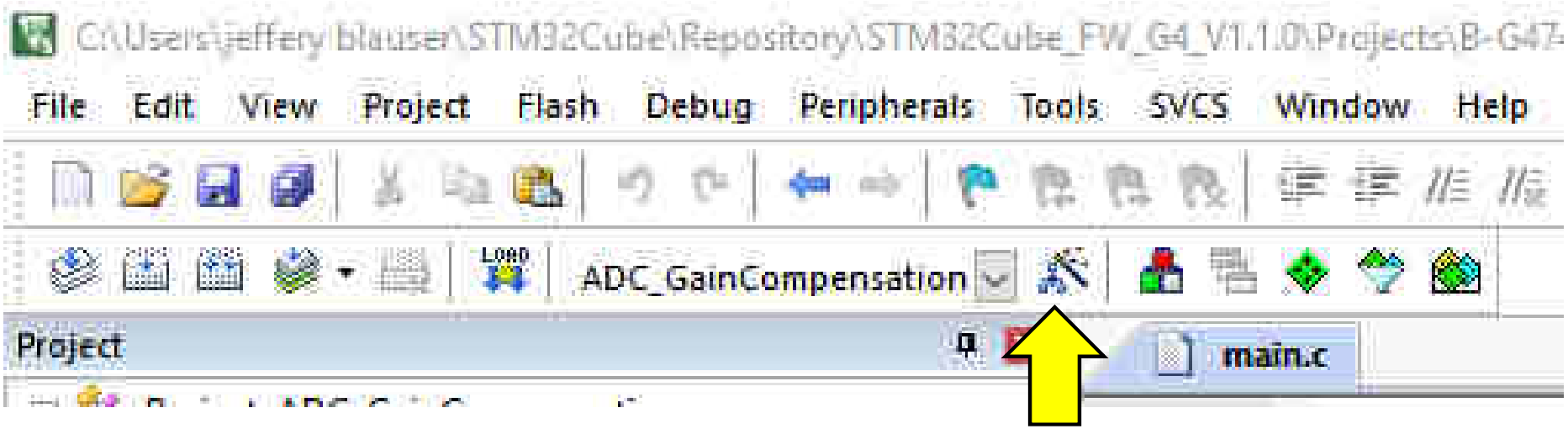




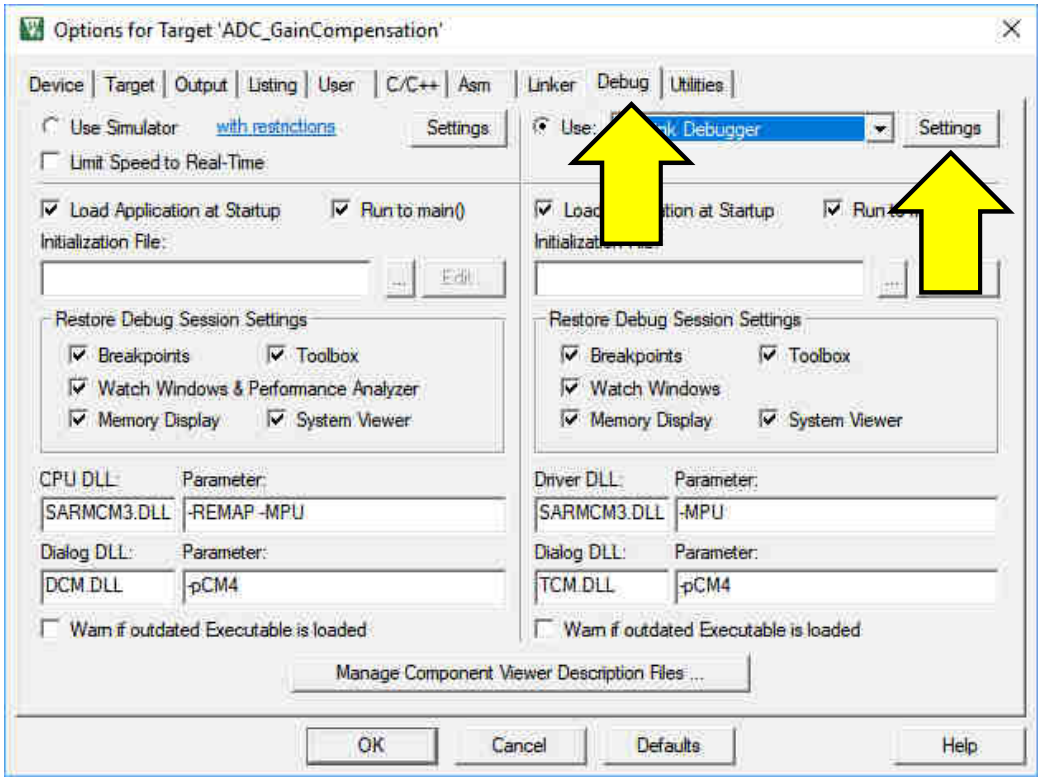


Keil MDK-ARM

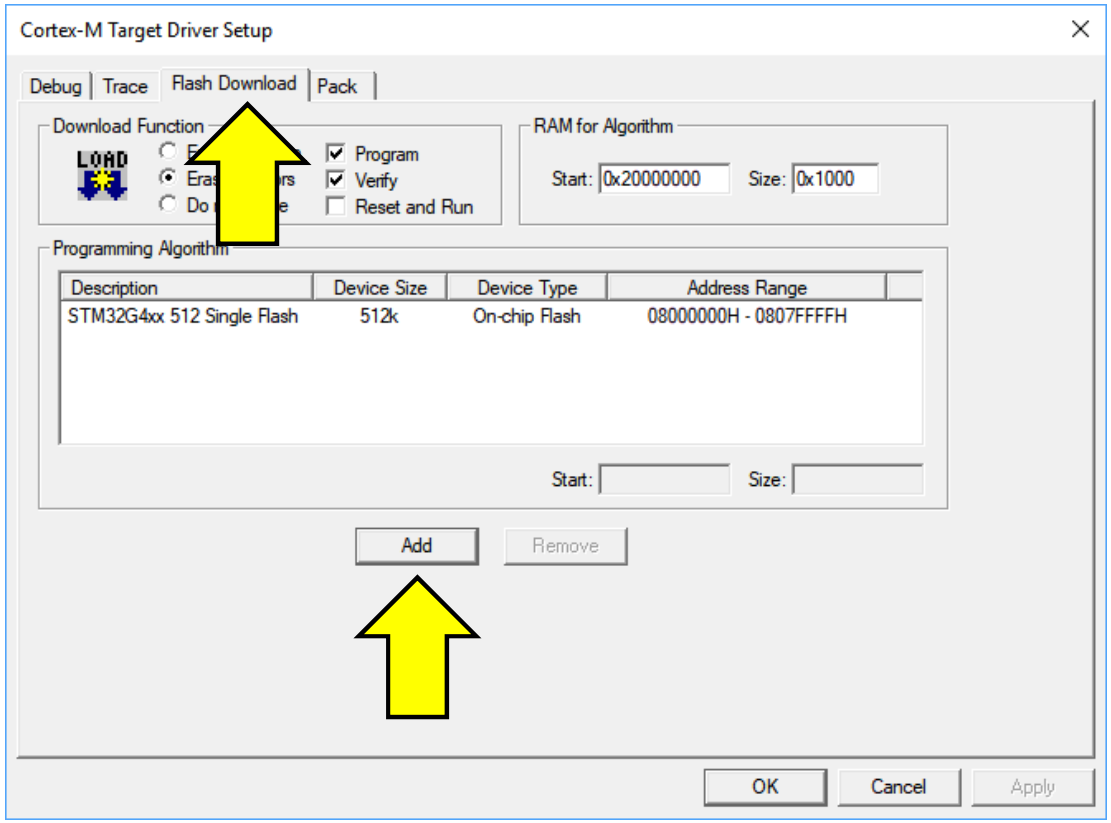
Options



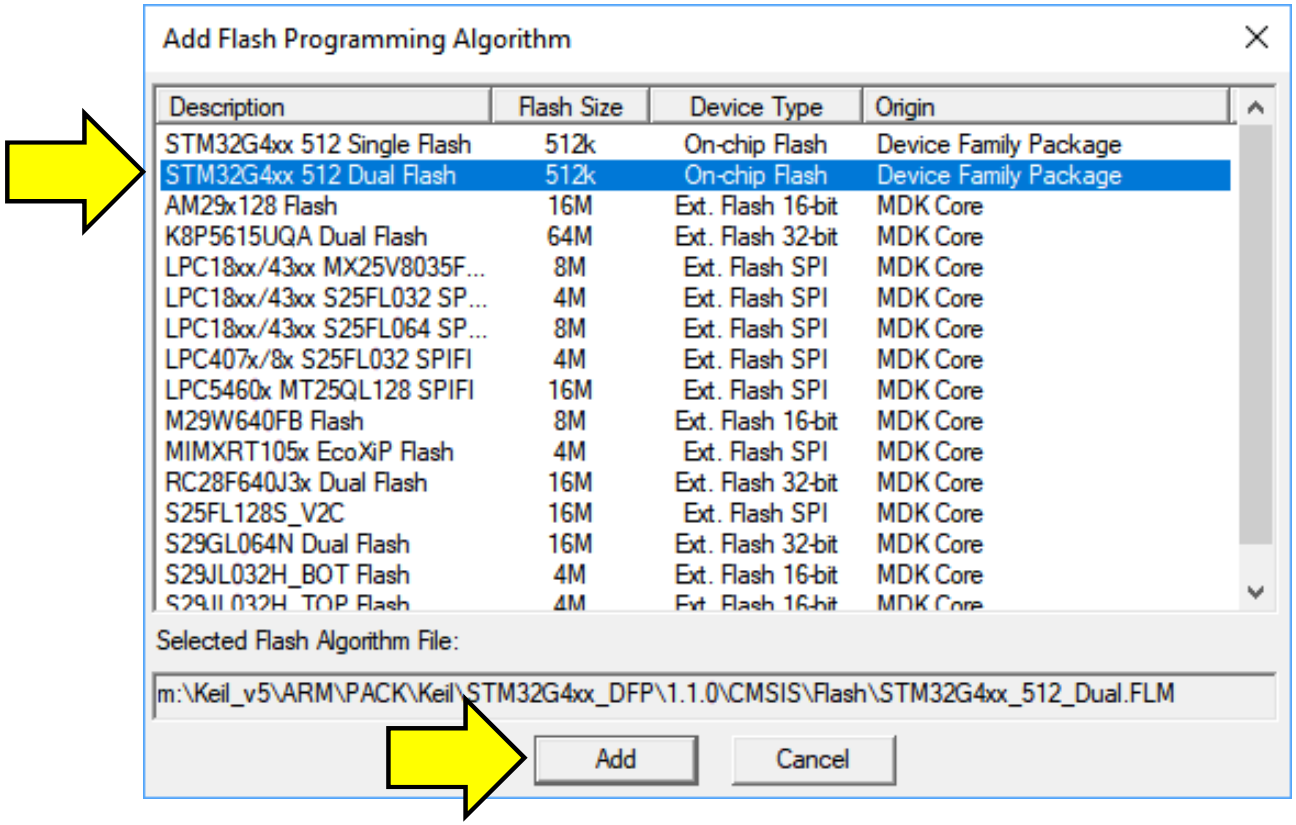
ST-Link Debugger - Settings



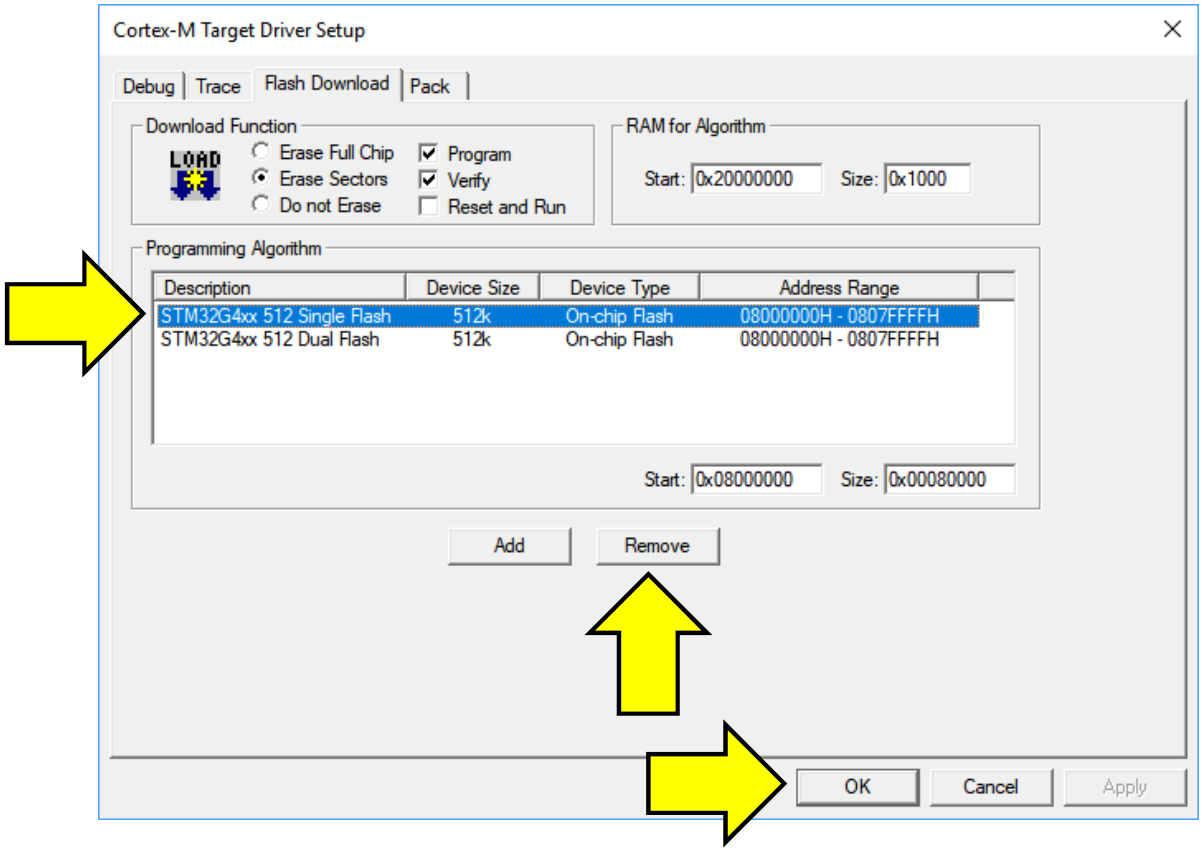
Flash Download → Add



STM32G4xx 512 Dual Flash



Remove Single Flash



Place 2 new variables for the Analyzer – main.c about line 56

```
51  /* USER CODE BEGIN PV */
52  /* Private variables ------
53
54  uint32_t wave[4] = {0x07FF0
55
56  uint16_t Display_DAC1DOR1;
57  uint16_t Display_DAC1DOR2;
58
59  /* USER CODE END PV */
```

See c:\2019_STM32G4_Workshop\Labs\DAC
For the file CodeLinesForMain_DAC.txt for cut and paste
Or open the file from the Doc folder in the Editor

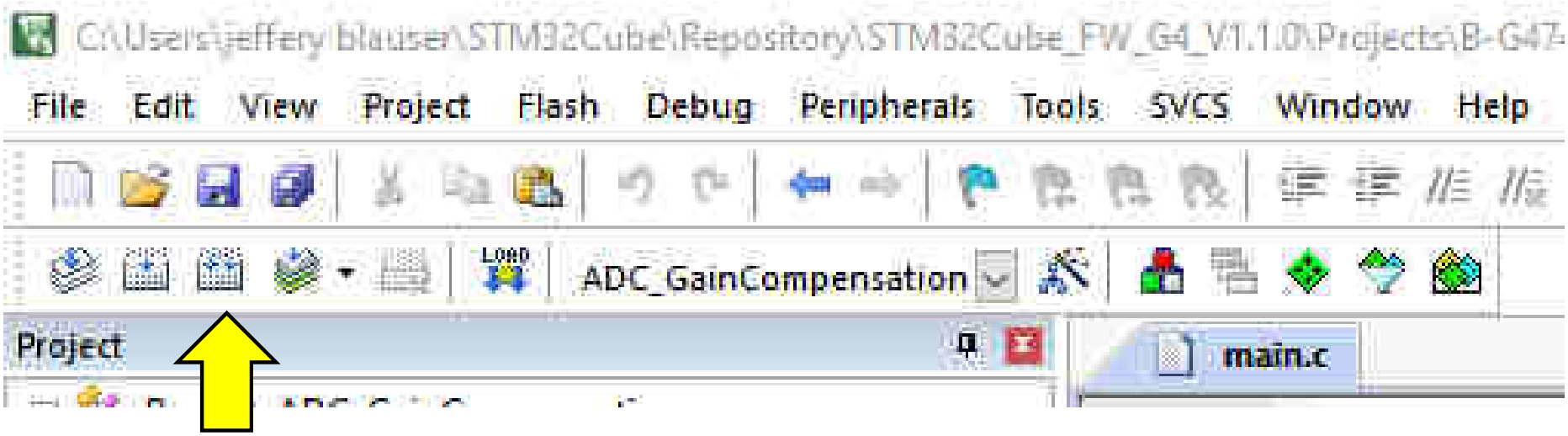
Execute 3 new lines for the Analyzer – main.c about line 143

```
138   while (1)
139   {
140     /* USER CODE END WHILE */
141
142     /* USER CODE BEGIN 3 */
143     Display_DAC1DOR1 = DAC1->DOR1;
144     Display_DAC1DOR2 = DAC1->DOR2;
145     HAL_Delay(25);
146   }
```

See c:\2019_STM32G4_Workshop\Labs\DAC
For the file CodeLinesForMain_DAC.txt for cut and paste.
Or open the file from the Doc folder in the Editor

Keil MDK-ARM

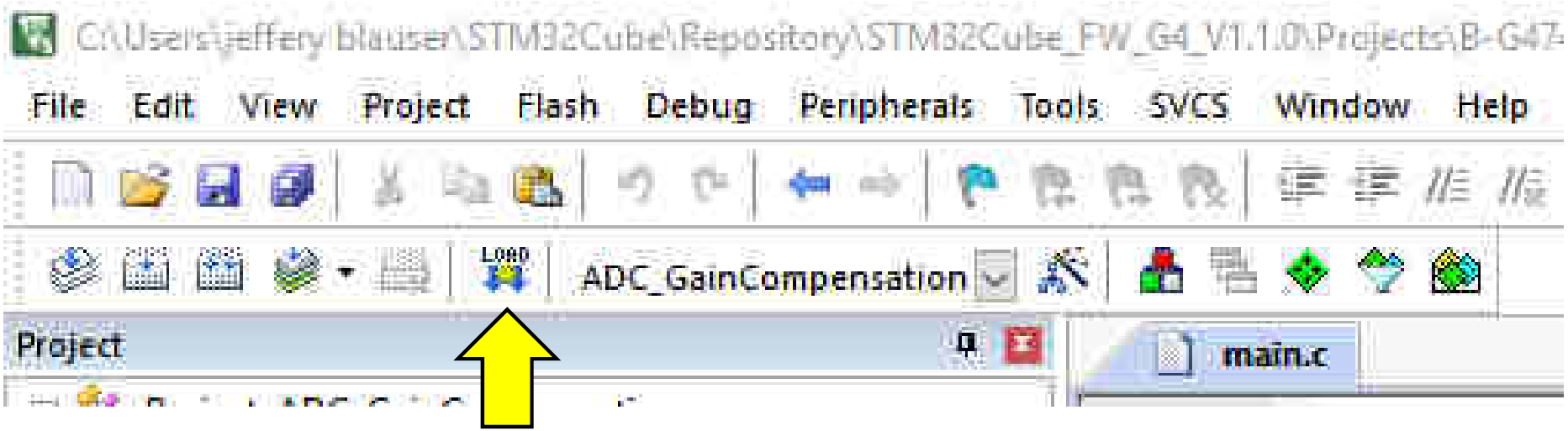
Build




```
Build Output
Program Size: Code=6572 RO-data=576 RW-data=92 ZI-data=1204
FromELF: creating hex file...
DAC_DualConversionFromDMA\DAC_DualConversionFromDMA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"DAC_DualConversionFromDMA\DAC_DualConversionFromDMA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:01:38
```

Keil MDK-ARM

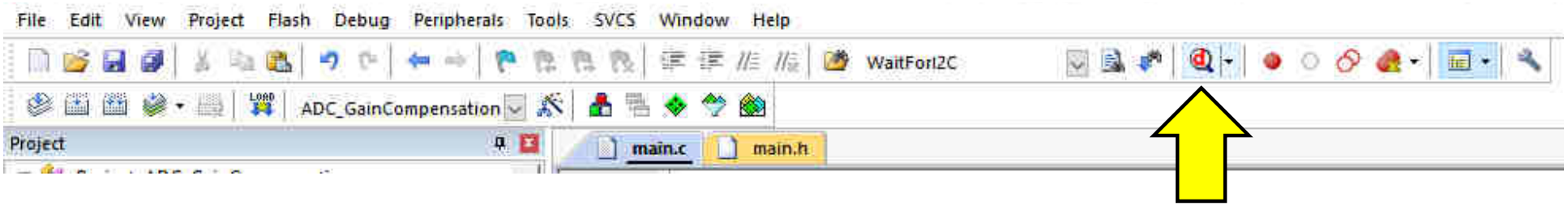
Load

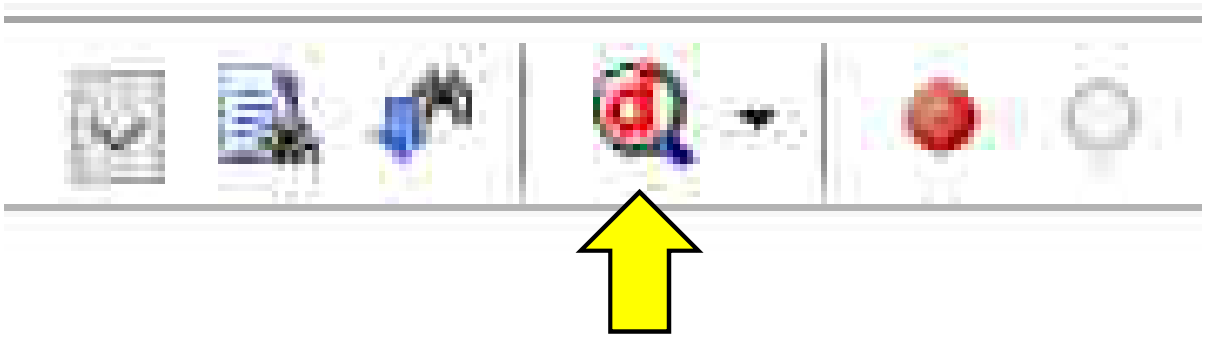



```
Build Output
Build Time Elapsed:  00:01:38
Load "DAC_DualConversionFromDMA\\DAC_DualConversionFromDMA.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 14:51:03
```

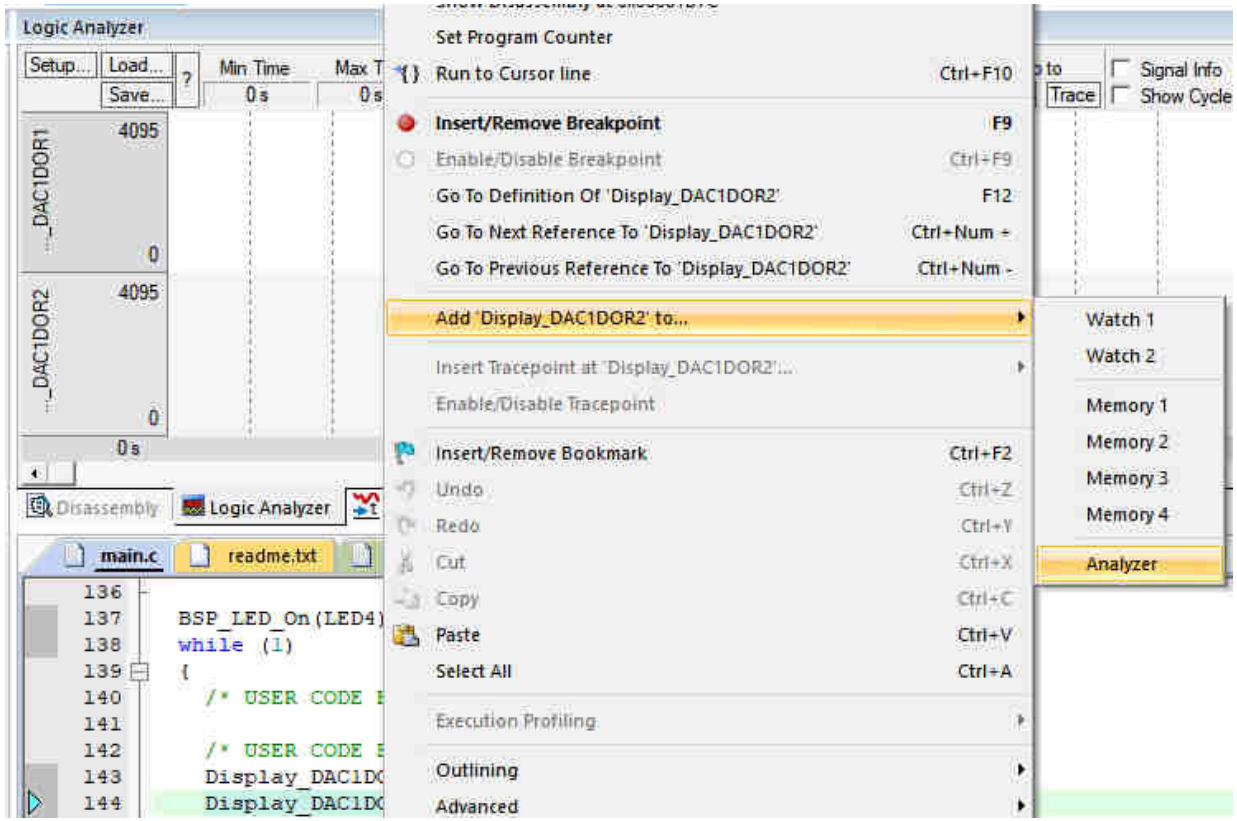
Keil MDK-ARM

Debugger



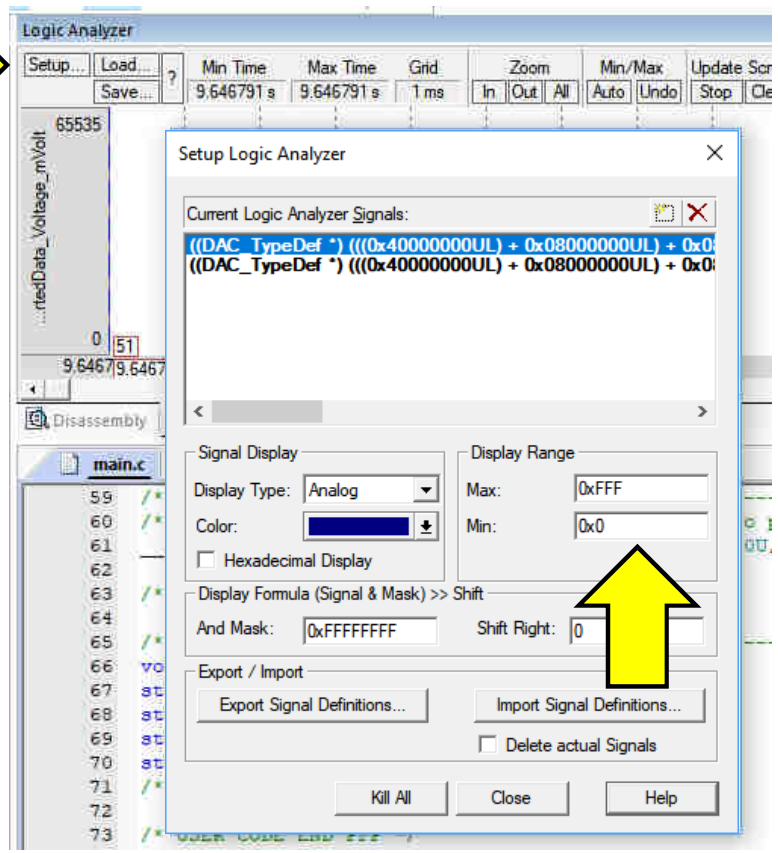


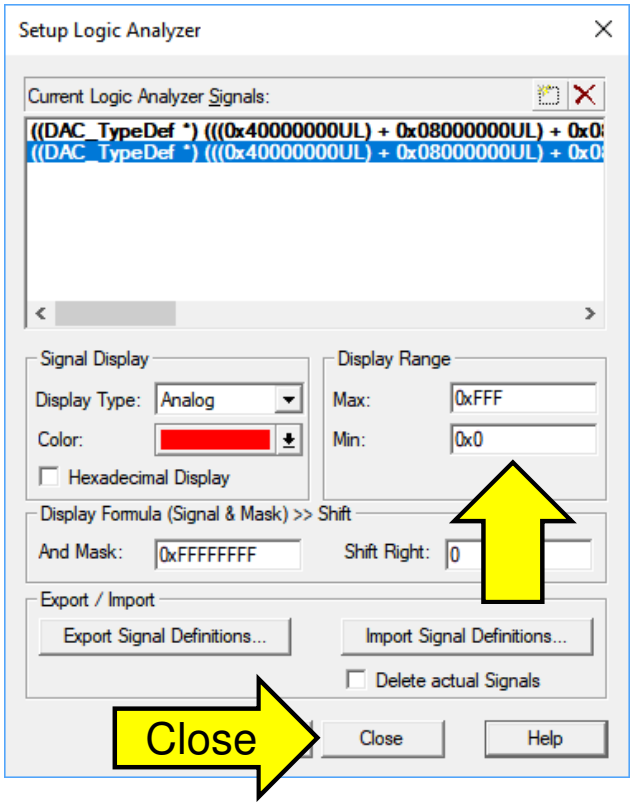
Add to Analyzer Display_DAC1DOR1 & 2



Setup – Min=0x0, Max = 0xFFF

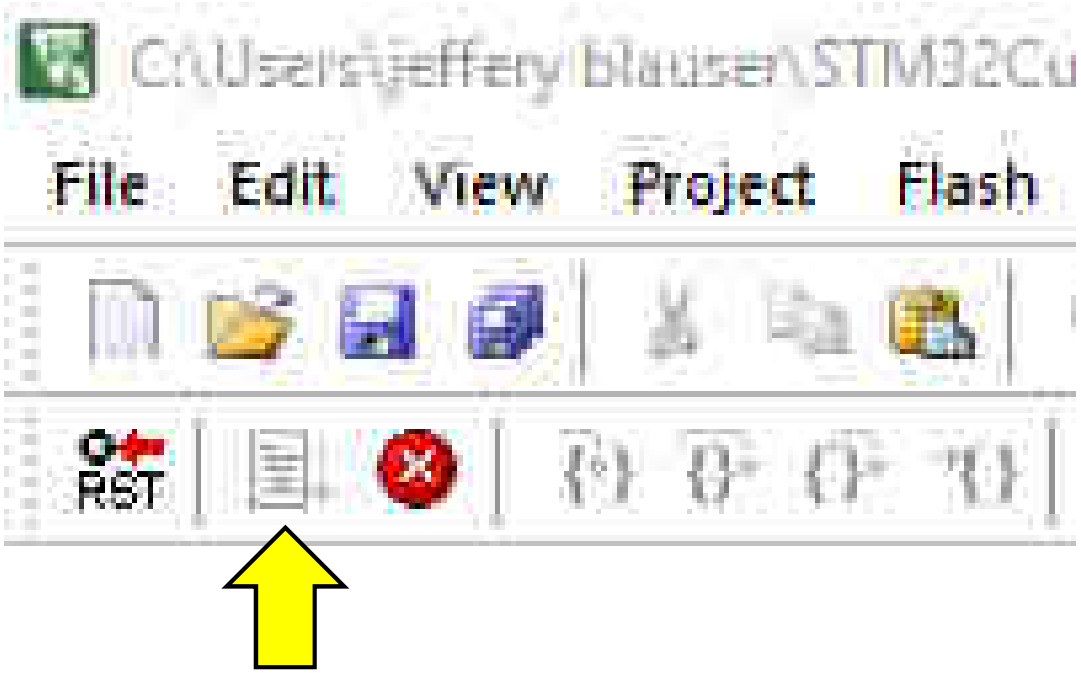
Setup



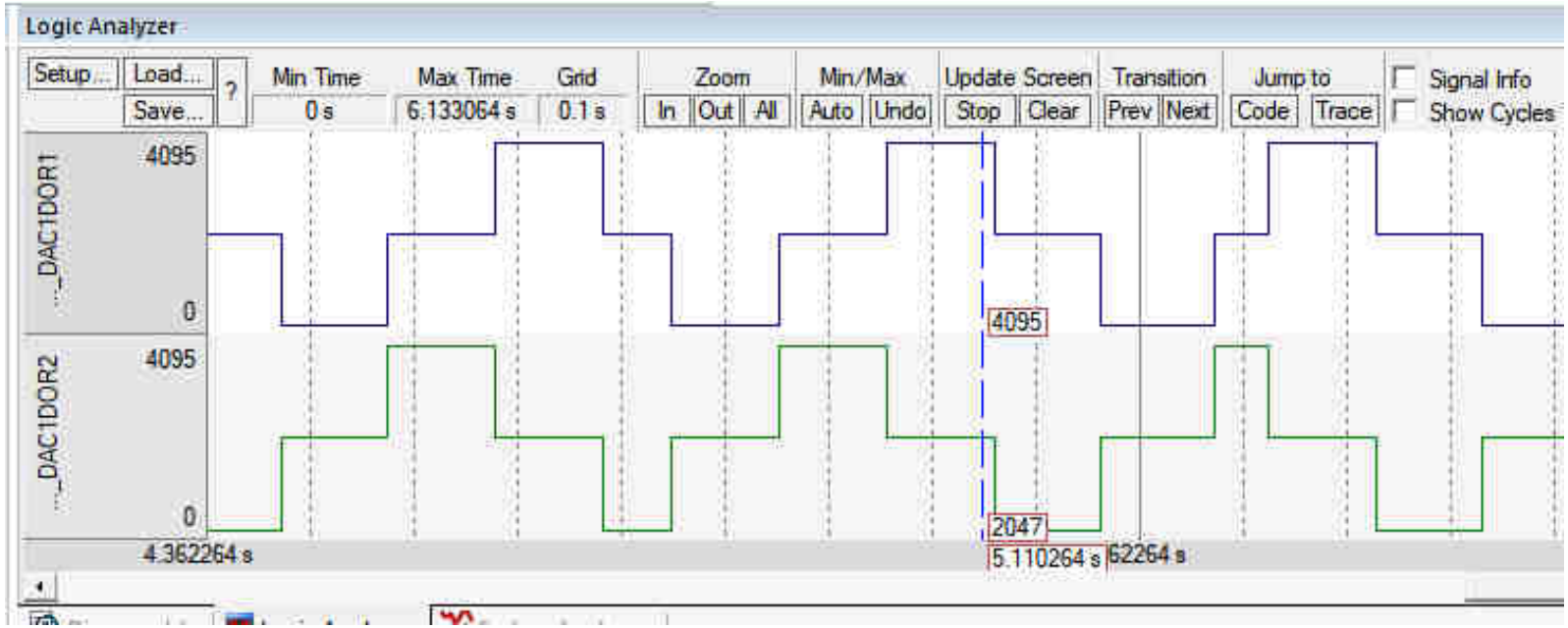


Keil MDK-ARM

RUN

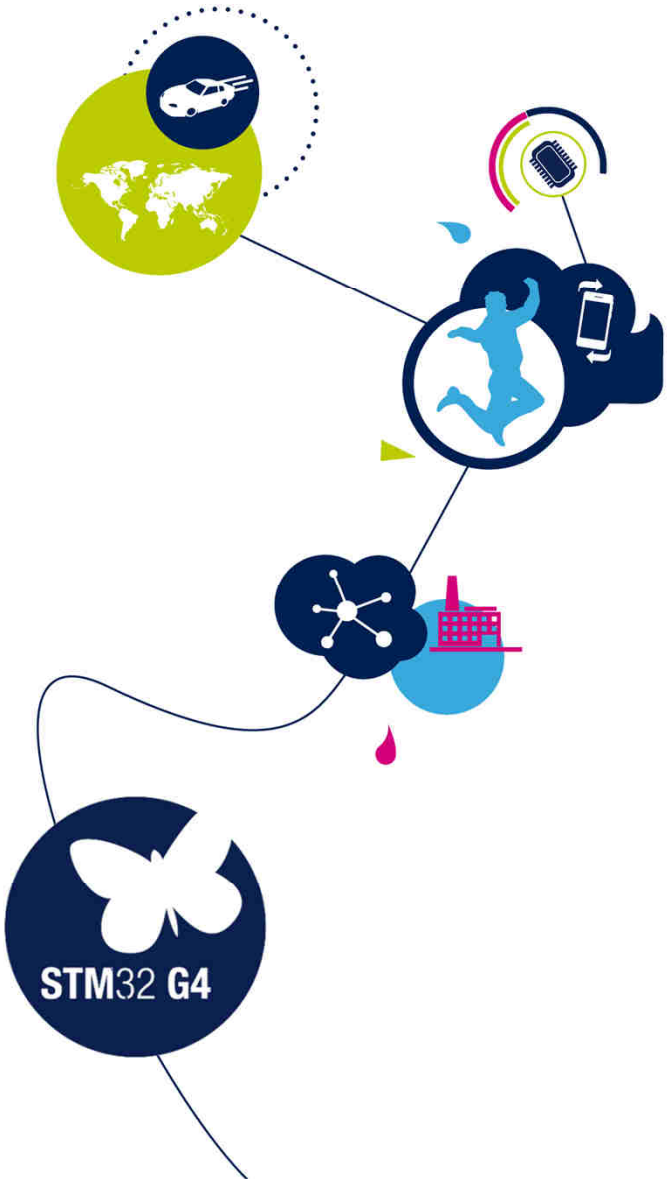


Logic Analyzer Output



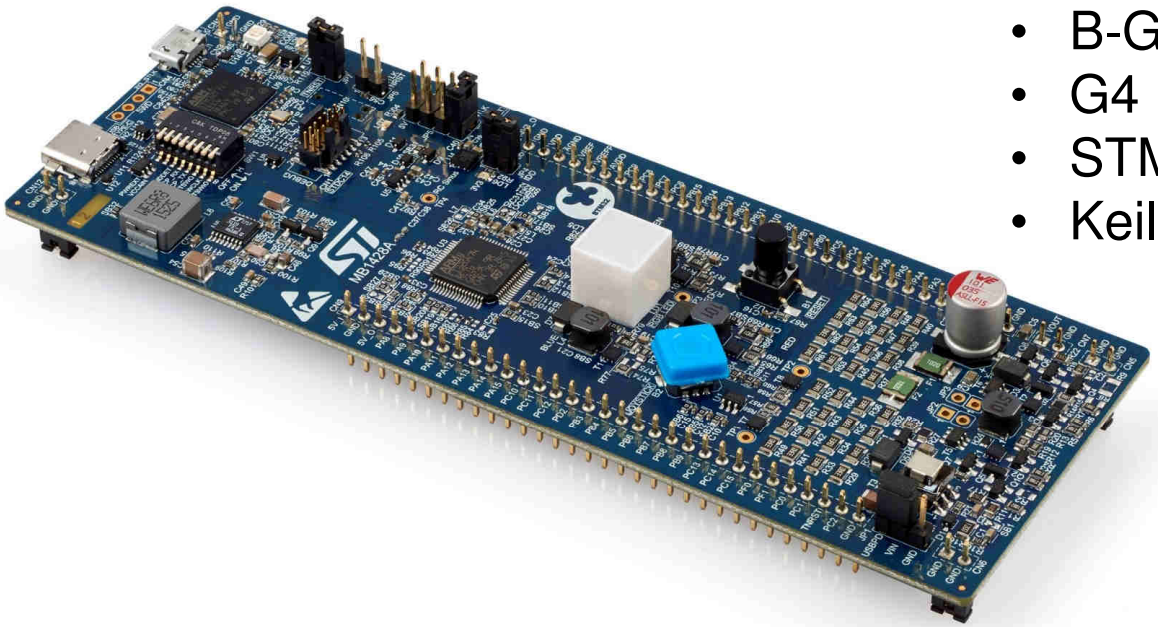
STM32G4 mixed-signal MCU workshop

Hands On:
Programmable Gain Amplifier



life.augmented

Systems Check



- USB cable
- B-G474E-DPOW1 Discovery Board
- G4 Family Cube Library
- STM32CubeMx
- Keil MDK-ARM

Rich, Advanced Analog

Mixed-signal SoC for wide variety of applications

ADC (up to 5)	**Values**
Topology	**SAR 12-bit** + HW oversampling → **16-bit**
Sampling rate	Up to **4 Msps**
Input	Single-ended and differential
Offset and Gain compensation	Auto calibration to reduce gain and offset

DAC (up to 7)	**Values**
Sampling rate	**15 Msps** (internal) 1Msps (from buffered output)
Settling time	16ns

Op-Amp (up to 6)	**Values**
GBW	**13 MHz**
Slew rate	**45 V/µs**
Offset	3mV over full T° range 1.5mV @ 25°C
PGA Gain (accuracy)	2, 4, 8, 16, -1,-3,-7,-15 **(1%)** 32, 64, -31,-63 (2%)

Comparator (up to 7)	**Values**
Power supply	1.62 .. 3.6V
Propagation delay	**16.7ns**
Offset	-6 .. +2 mV
Hysteresis	8 steps: 0, 9, 18, 27, 36, 45, 54, 63 mV

Programmable Gain Amplifiers

G4 PGA

- 6 PGA's available
- Non inverting gain 2/4/8/16/32/64
- Inverting gain -1/-3/-7/-15/-31/-63
- Rail-to-Rail input and output
- Low input bias current
- Low input offset voltage
- 15MHz GBW product
- High speed mode for better slew rate

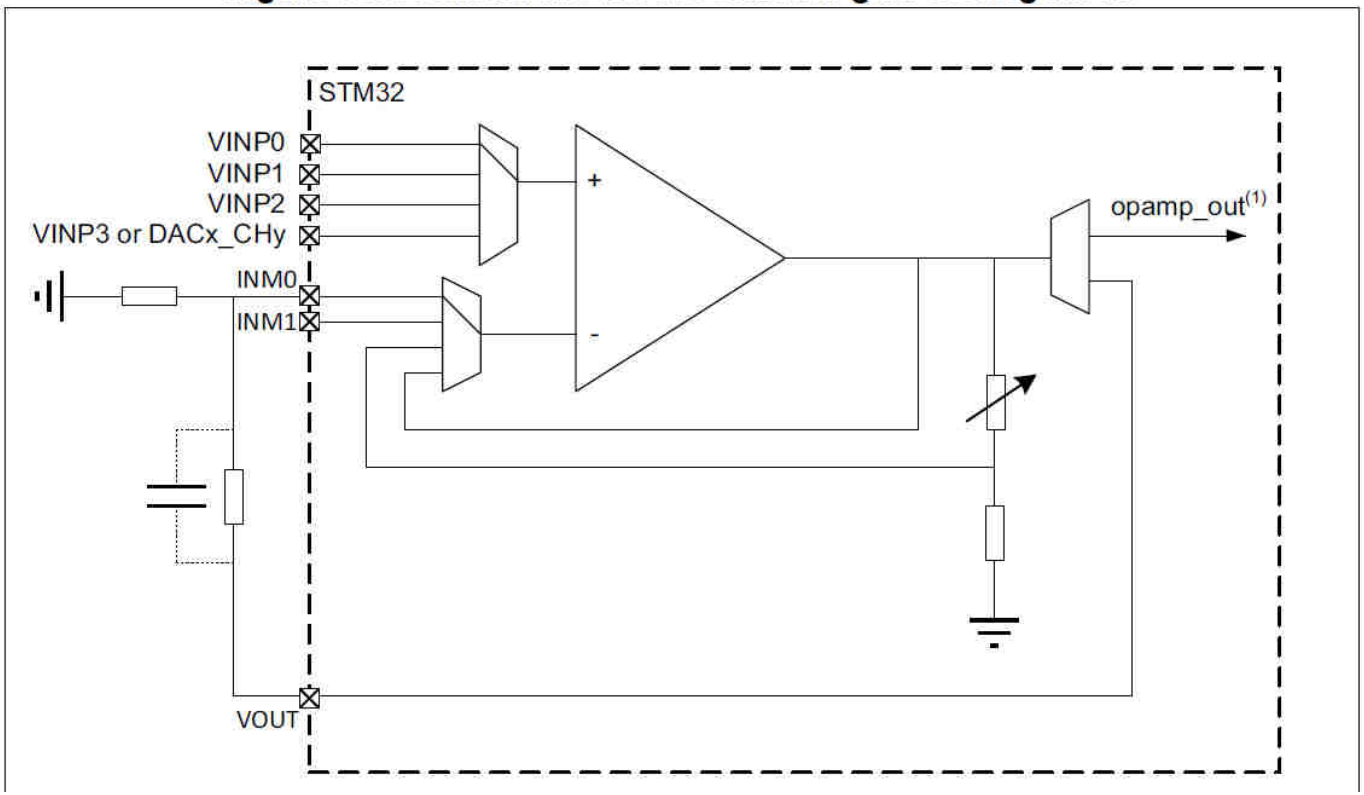
Internally connected to:

- ADC inputs
- External pins, both inputs and outputs
- Internal resistors
- DAC3 & DAC4 outputs

Programmable Gain Amplifiers

G4 Op Amp Stand Alone Mode

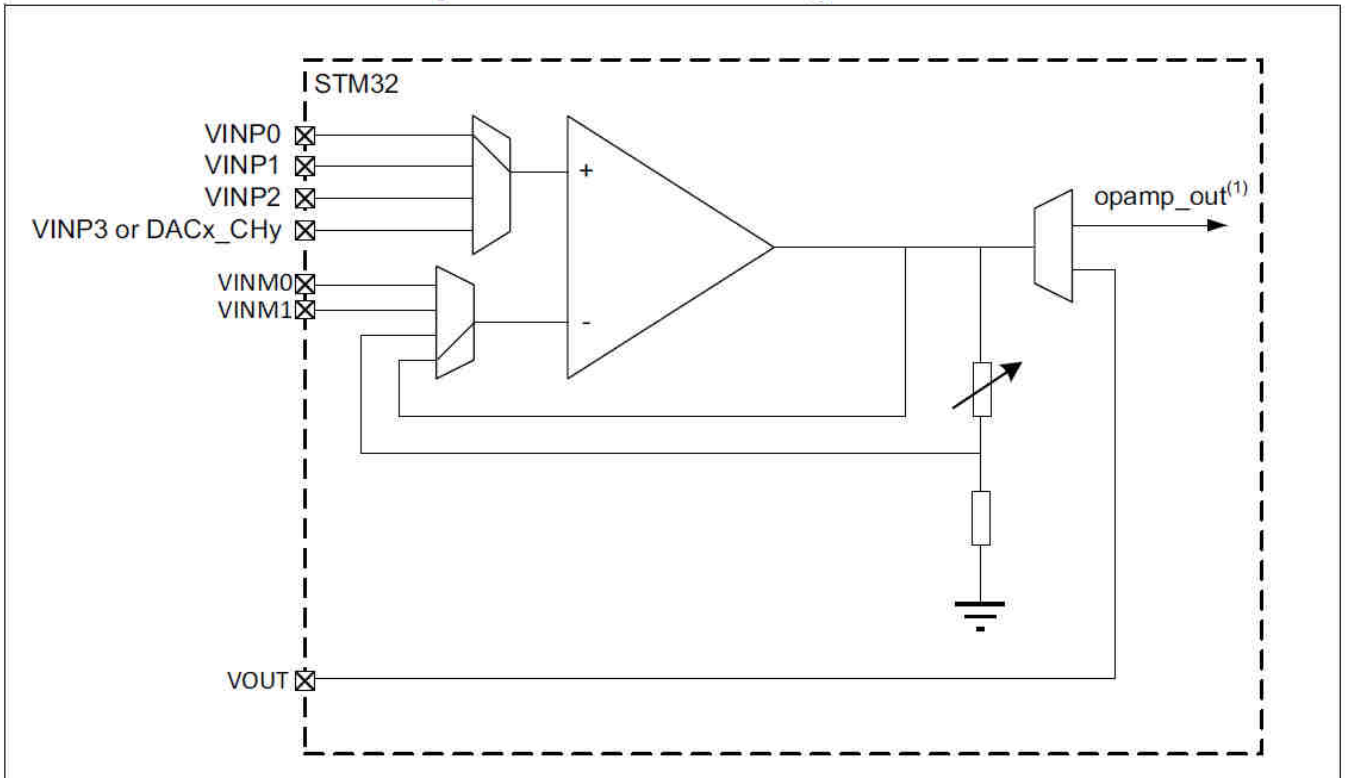
Figure 168. Standalone mode: external gain setting mode



Programmable Gain Amplifiers

G4 Op Amp Follower Mode

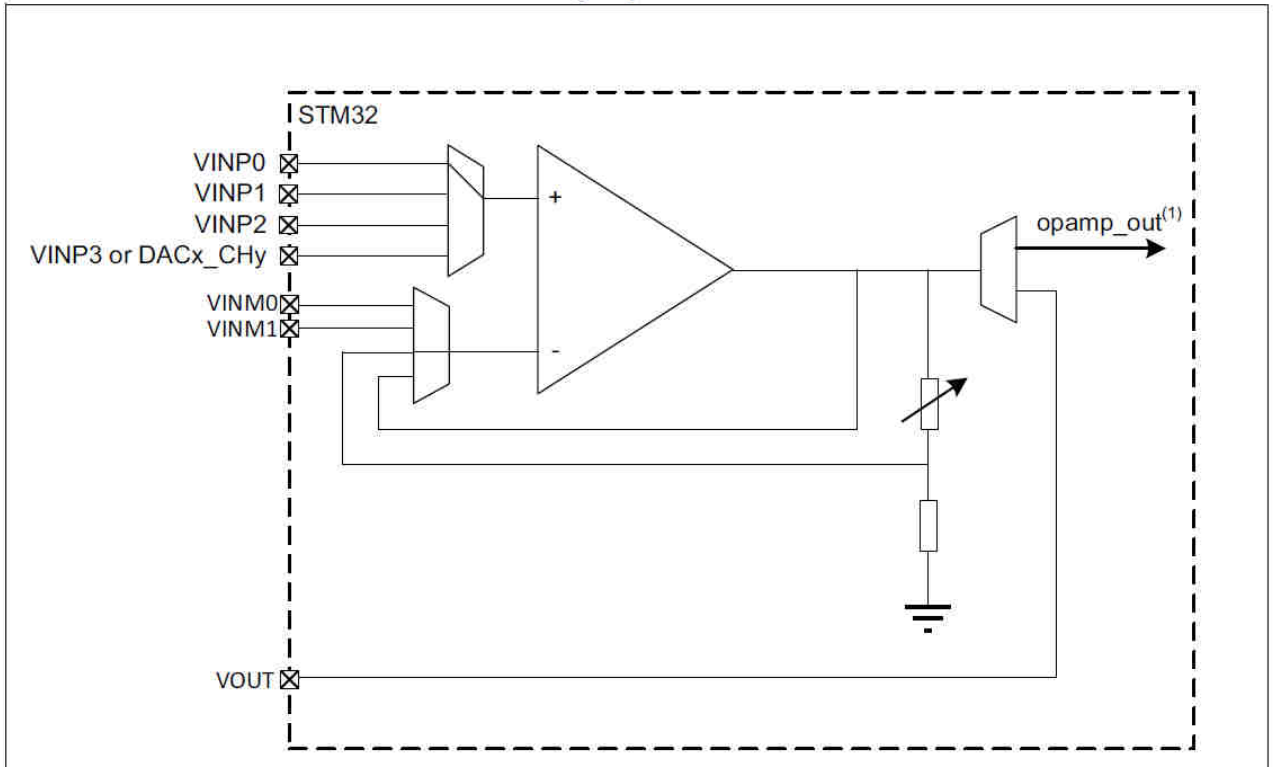
Figure 169. Follower configuration



Programmable Gain Amplifiers

G4 Op Amp PGA Mode – This is what we will use today

Figure 170. PGA mode, internal gain setting (x2/x4/x8/x16/x32/x64), inverting input not used



PGA Hands On

Lab steps 1 of 2

- The DMA provides samples (sine wave) to the DAC.
  - The DAC peripheral generates a sine wave signal on DAC1_OUT1(PA4) which is amplified by the OPAMP3. (Later DAC3_OUT2 as well)
  - The OPAMP3 amplified output is on PB1 with gain of either 2 or 4.
    The OPAMP gain is changed on the fly while the OPAMP remains enabled.
  - The test steps are:
  - Step 1:
    - DAC: normal power mode
    - OPAMP: normal power mode with gain = 2
    - DMA: circular mode - DMA half transfer IT handled by CPU

PGA Hands On

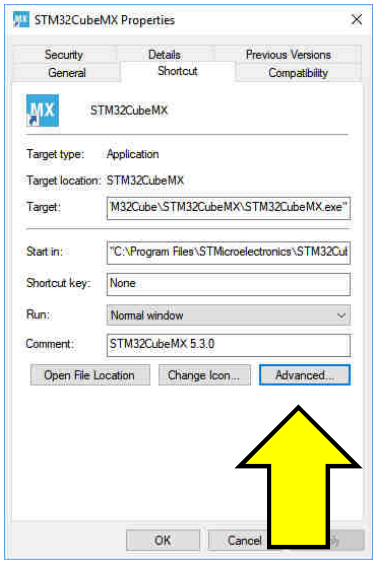
Lab steps 2 of 2

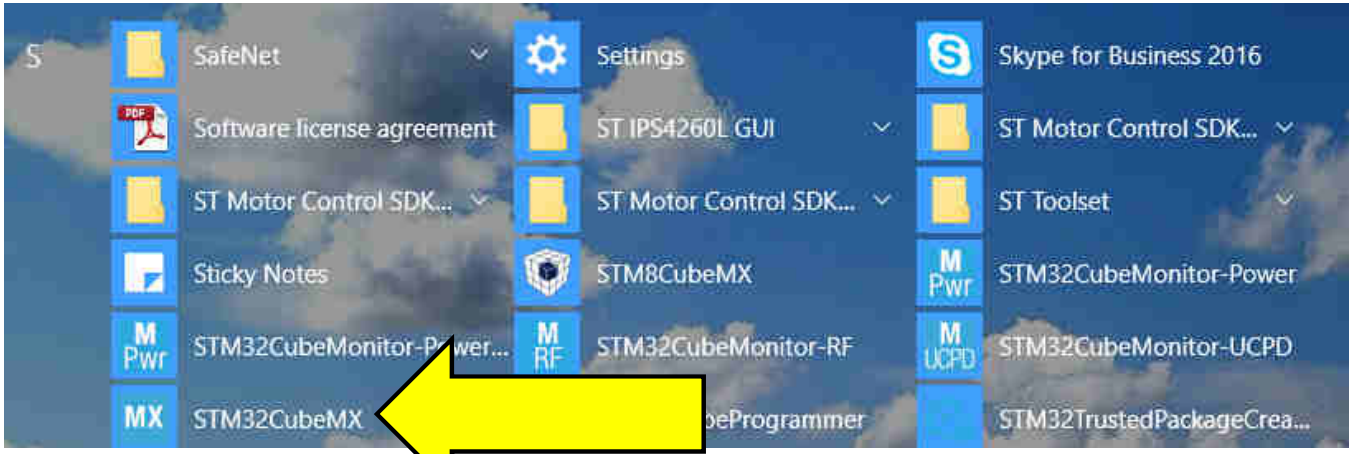
- Step 2:
  - DAC: normal power mode
  - OPAMP: normal power mode
    - gain is changed from 2 to 4 or from 4 to 2
  - DMA: circular mode - DMA half transfer IT handled by CPU

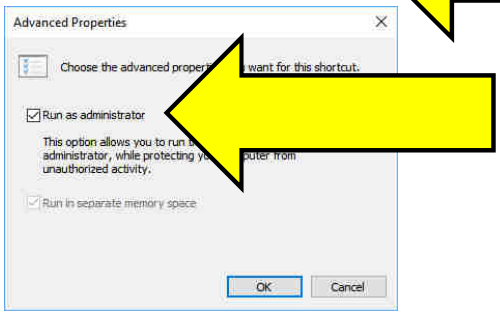
We will use the built in Logic Analyzer today instead of an o'scope on the DACs and Op Amps.

Start up

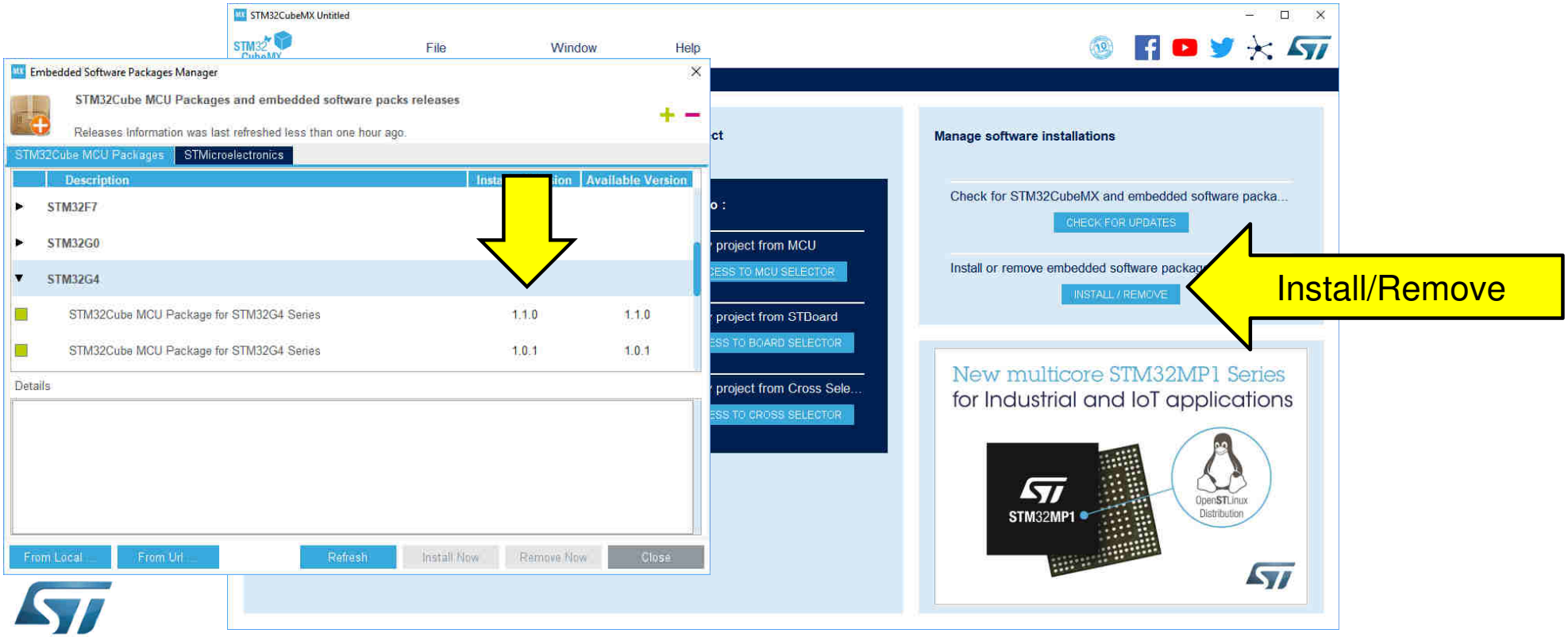

STM32Cu...


STM32CubeMX Properties
Security
Details
Previous Versions
General
Shortcut
Compatibility
STM32CubeMX
Target type: Application
Target location: STM32CubeMX
Target: "M32Cube\STM32CubeMX\STM32CubeMX.exe"
Start in: "C:\Program Files\STMicroelectronics\STM32Cul
Shortcut key: None
Run: Normal window
Comment: STM32CubeMX 5.3.0
Open File Location
Change Icon...
Advanced...
OK
Cancel


S
SafeNet
Settings
Skype for Business 2016
Software license agreement
ST IPS4260L GUI
ST Motor Control SDK...
ST Motor Control SDK...
ST Motor Control SDK...
ST Toolset
Sticky Notes
STM8CubeMX
STM32CubeMonitor-Power
STM32CubeMonitor-P wer...
STM32CubeMonitor-RF
STM32CubeMonitor-UCPD
STM32CubeMX
beProgrammer
STM32TrustedPackageCrea...

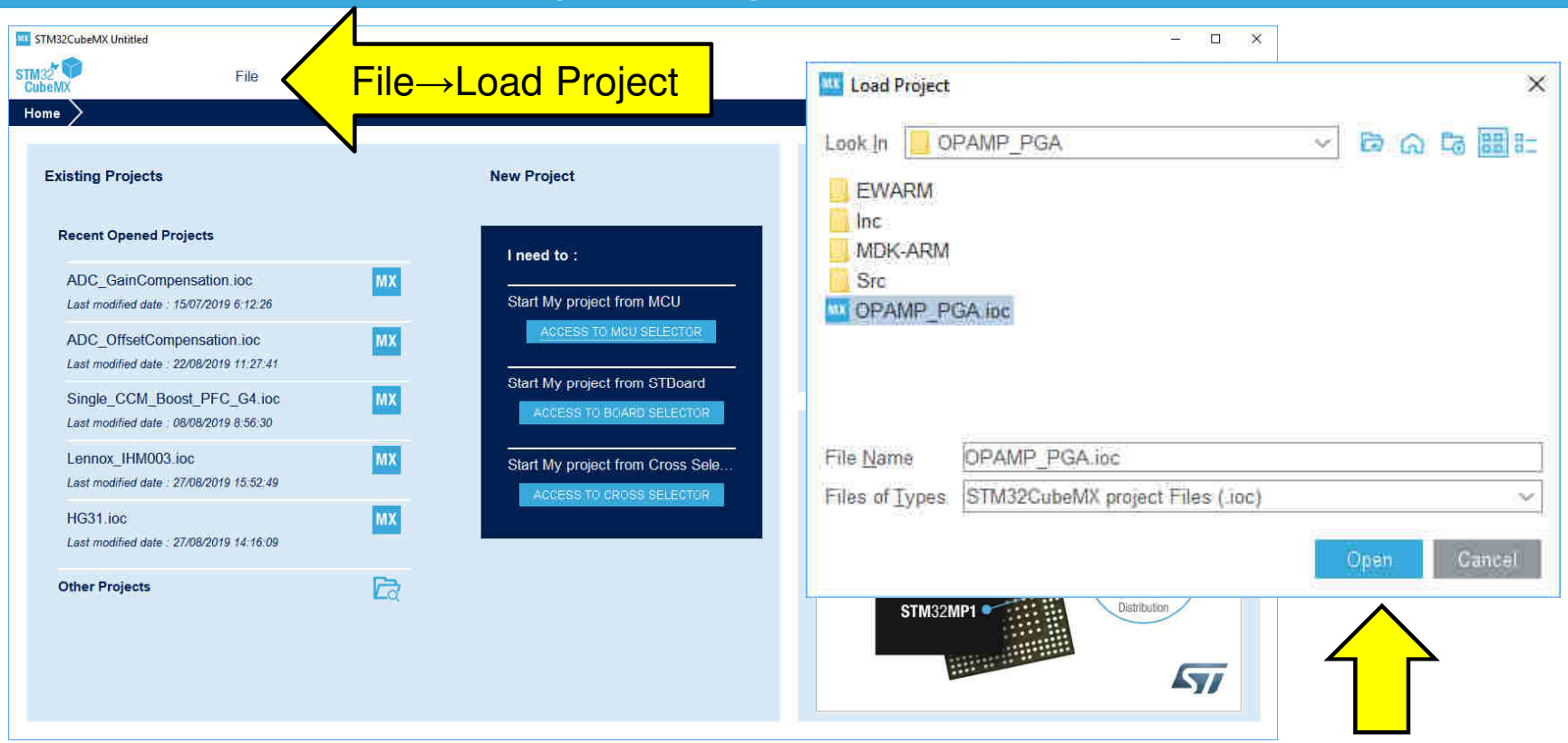

Advanced Properties
Choose the advanced properti want for this shortcut.
Run as administrator
This option allows you to run
administrator, while protecting yo puter from
unauthorized activity.
Run in separate memory space
OK
Cancel

Splash Page – G4 Library Installed?



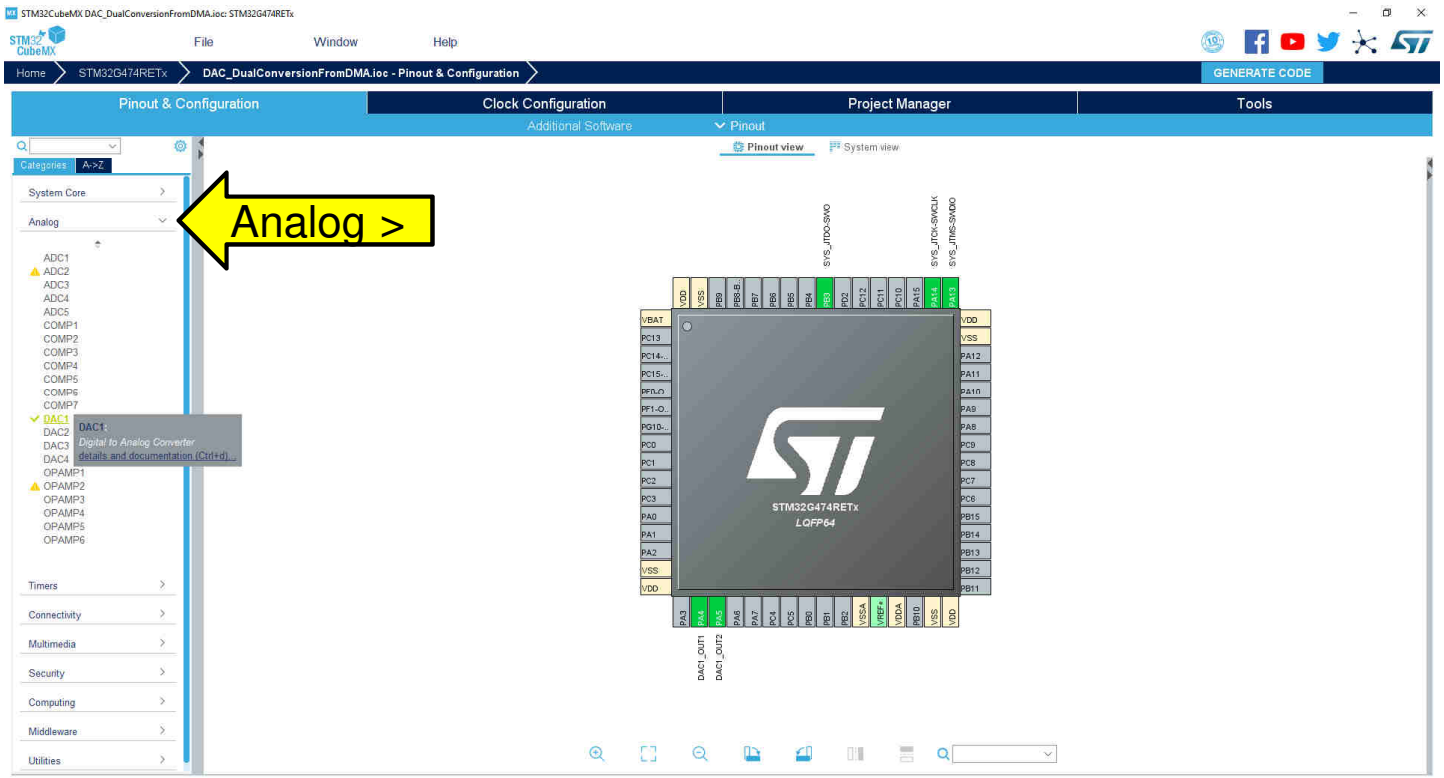
life.augmented

Splash Page

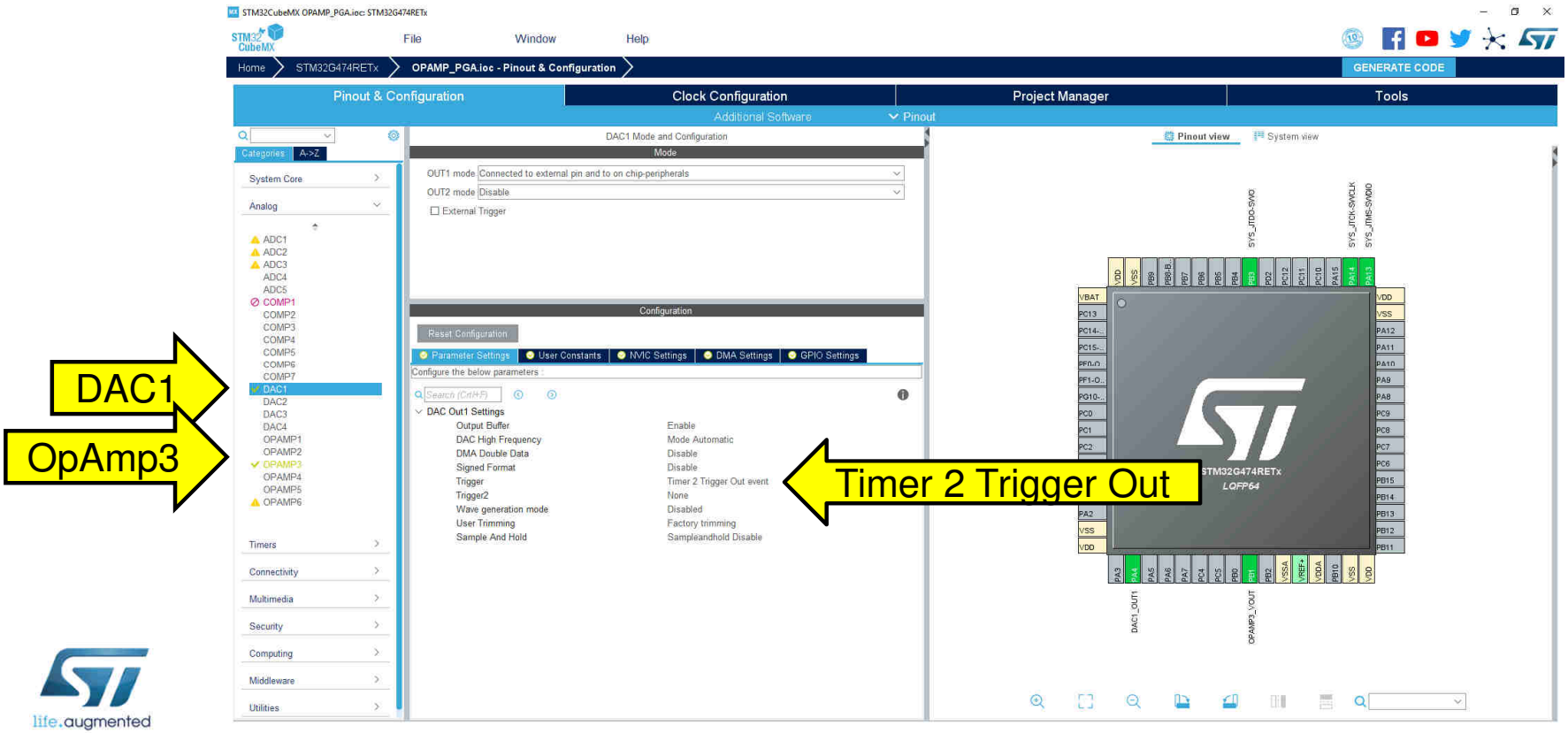


C:\2019_STM32G4_Workshop\Labs\OpAmp

Graphic Design Representation – Click on Analog>

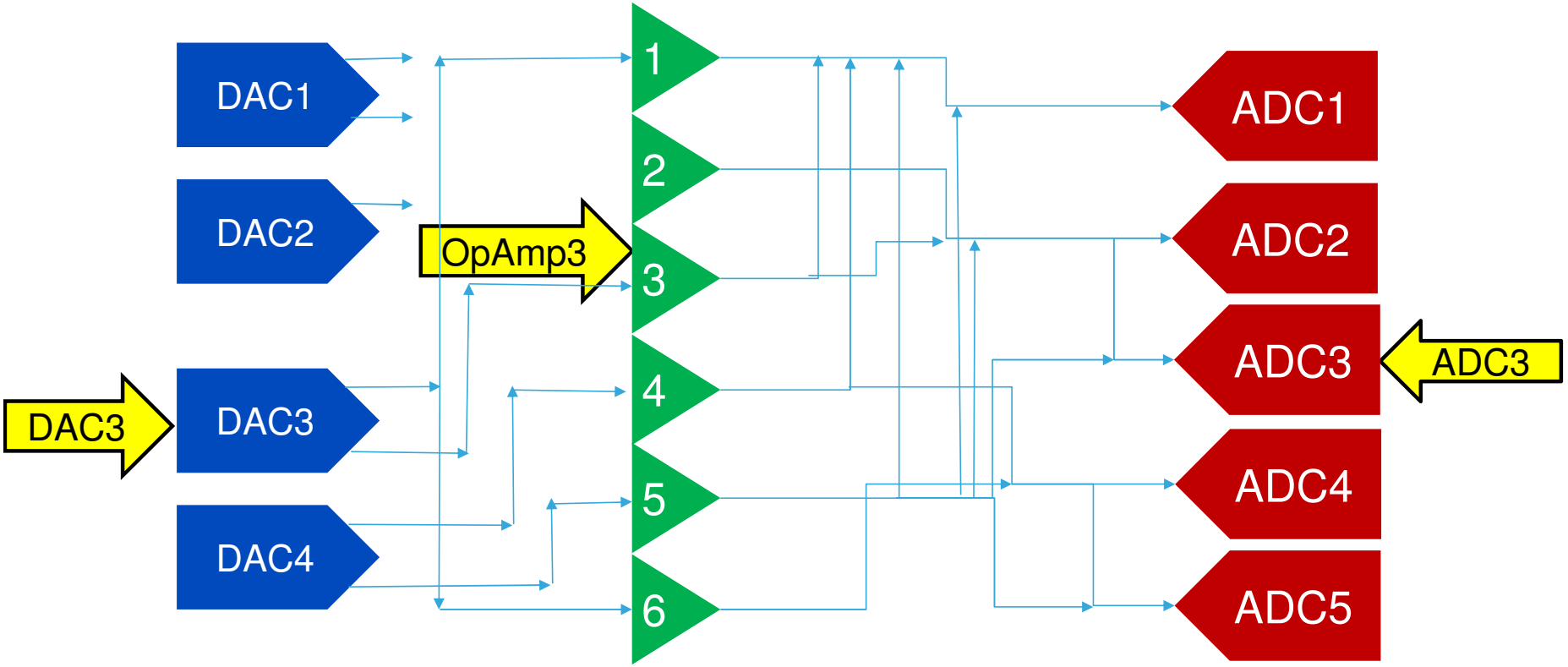


DAC1 (PA4), Timer 2, Op Amp 3 (PB1)



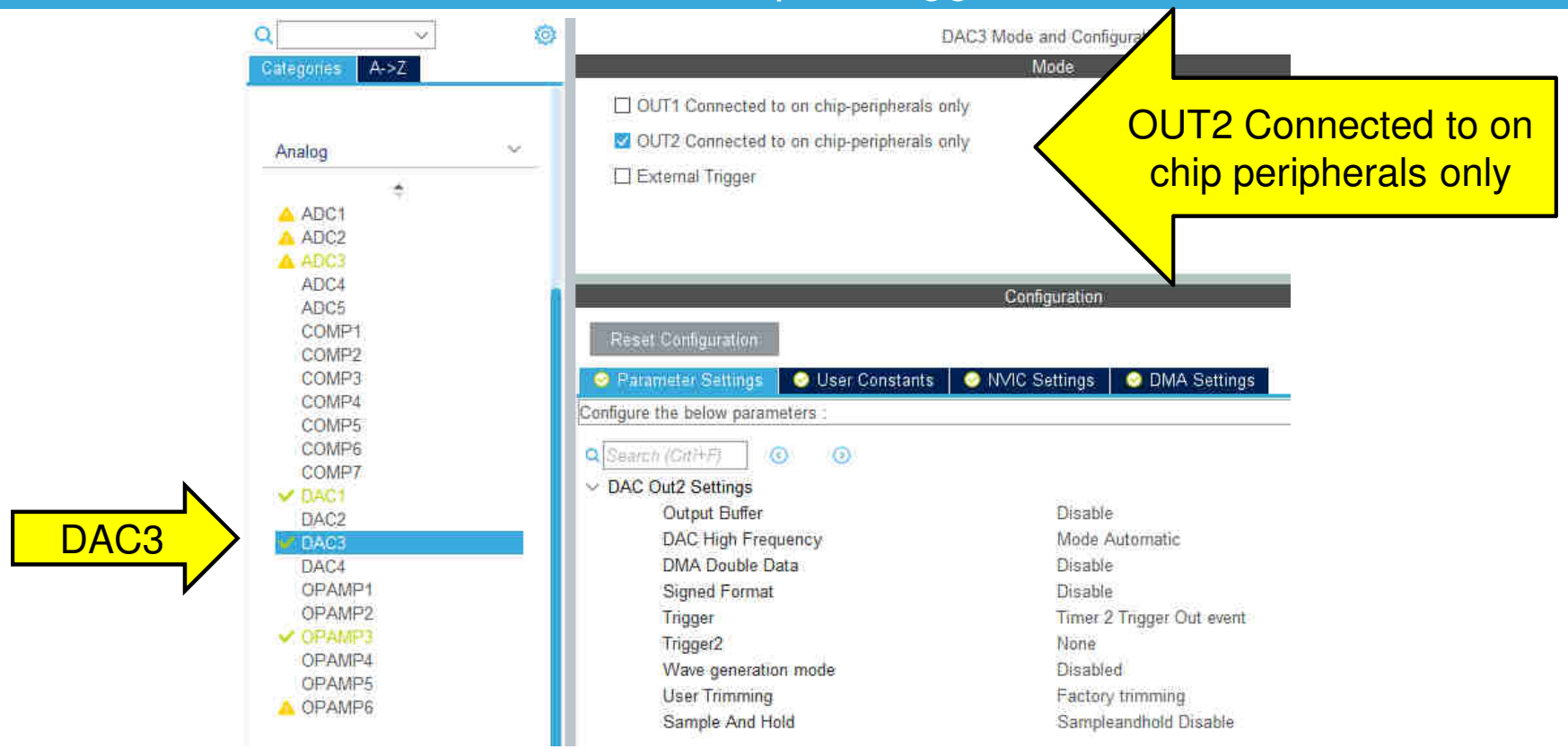
life.augmented

DAC3_OUT2 - Op Amp 3 – ADC3

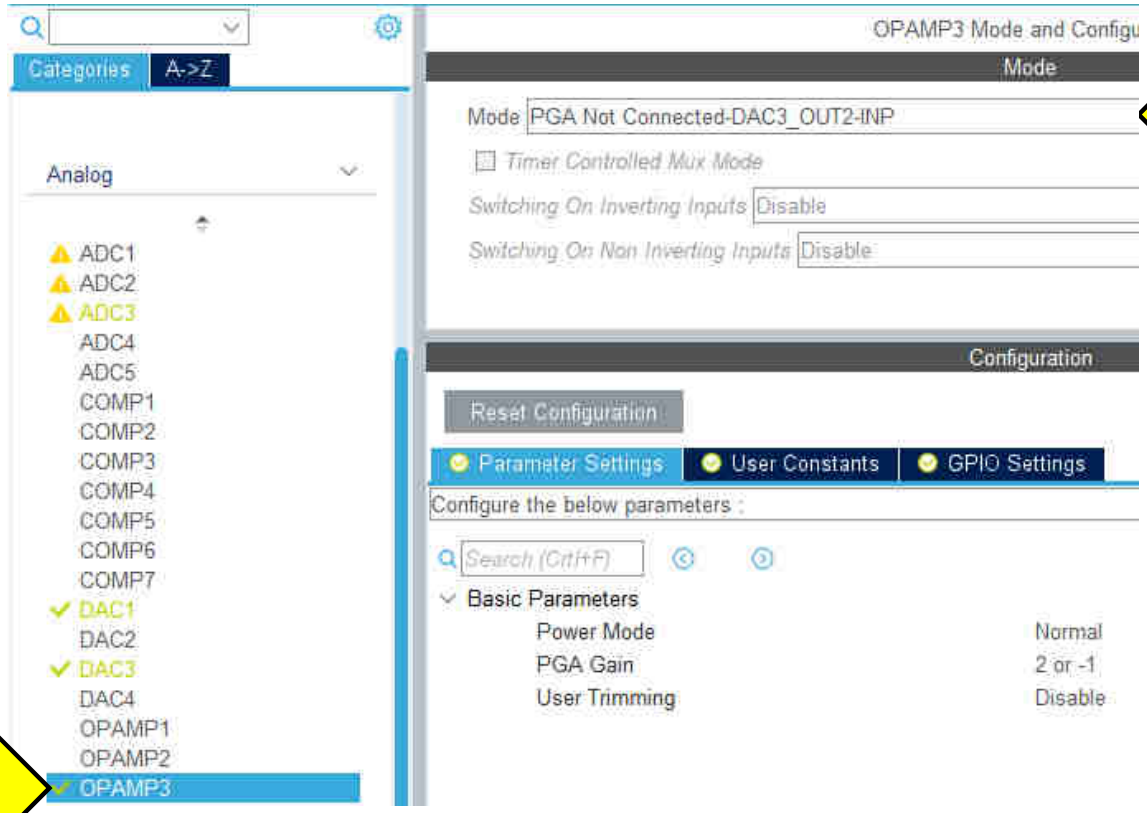


DAC3_OUT2 - Op Amp 3 – ADC3

DAC3 – Output, Trigger



Op Amp 3 - Mode

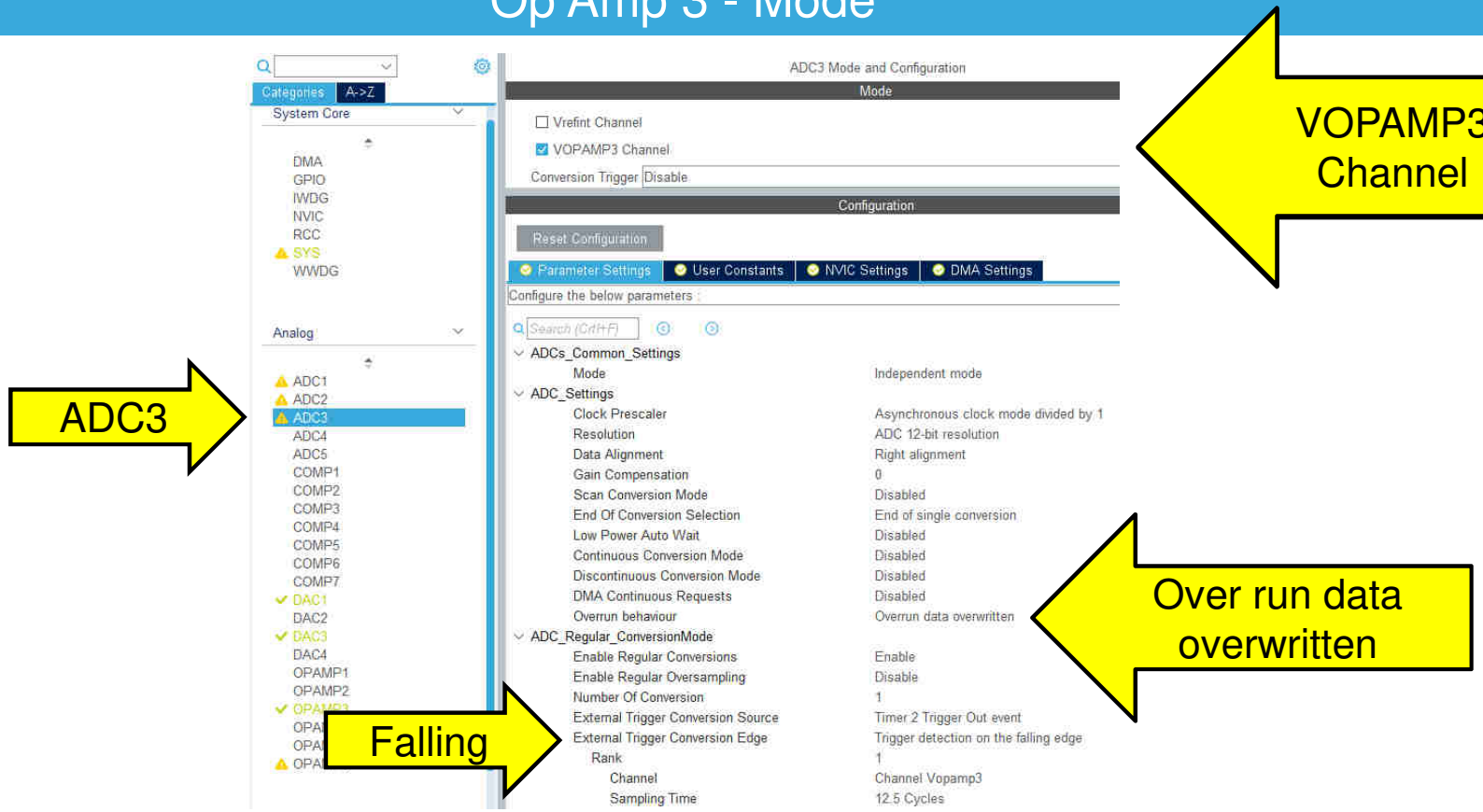


PGA Not Connected – DAC3_OUT2-INP

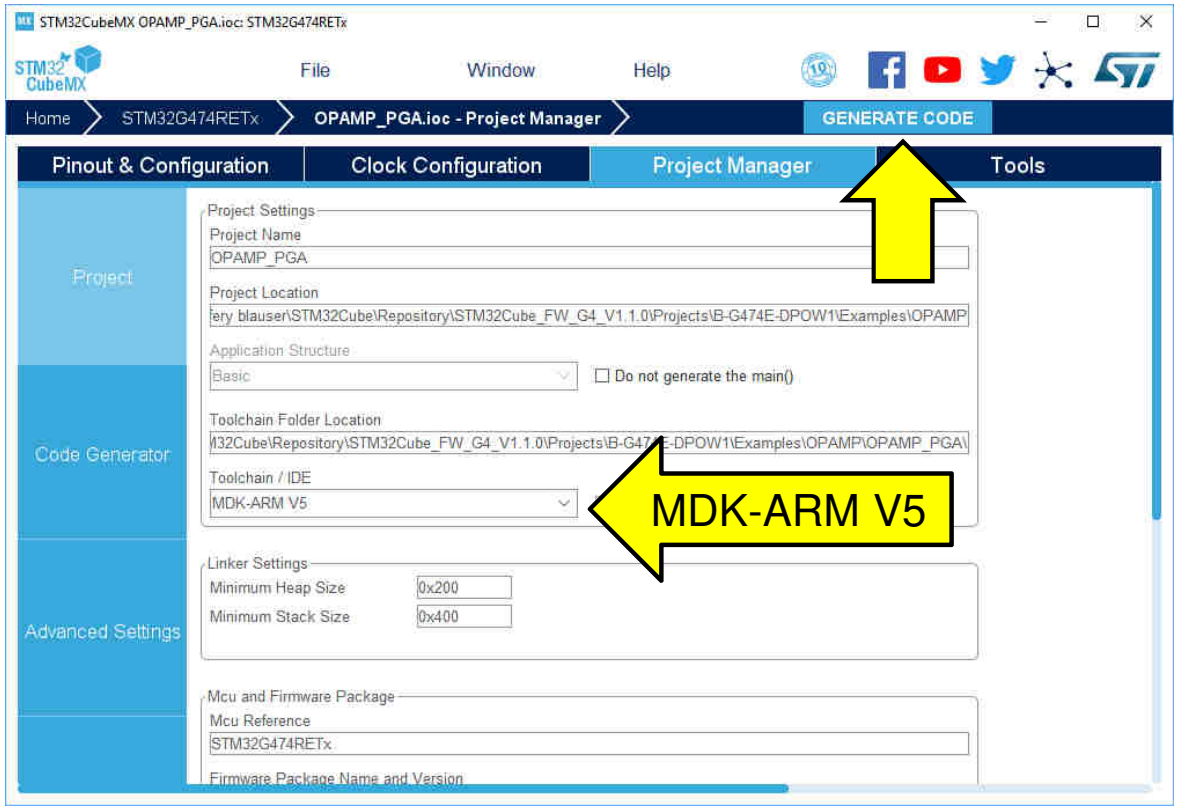
OpAmp3

life.augmented

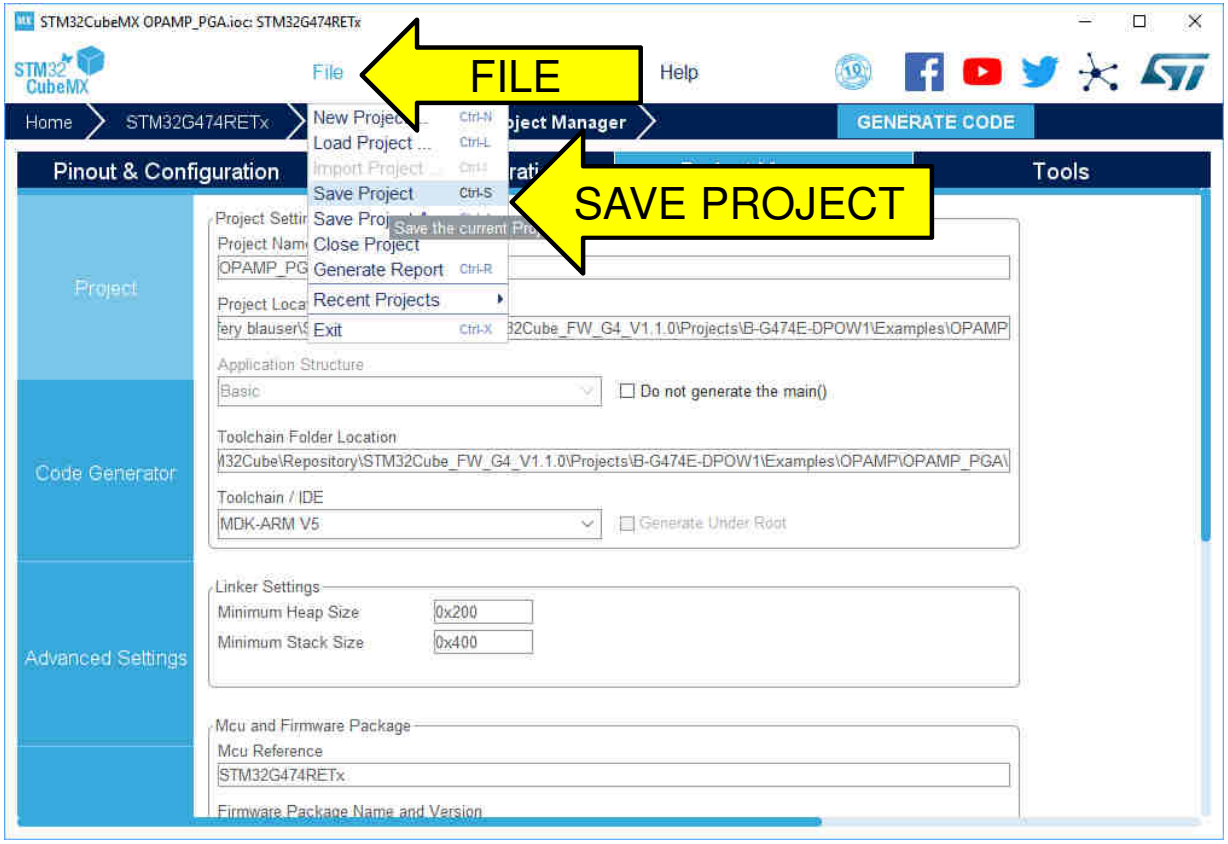
Op Amp 3 - Mode


ADC3
Falling
VOPAMP3 Channel
Over run data overwritten
Categories
A->Z
System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
WWDG
Analog
ADC1
ADC2
ADC3
ADC4
ADC5
COMP1
COMP2
COMP3
COMP4
COMP5
COMP6
COMP7
DAC1
DAC2
DAC3
DAC4
OPAMP1
OPAMP2
ADC3 Mode and Configuration
Mode
Vrefint Channel
VOPAMP3 Channel
Conversion Trigger Disable
Configuration
Reset Configuration
Parameter Settings
User Constants
NVIC Settings
DMA Settings
Configure the below parameters :
Search (Crtl+F)
ADCs_Common_Settings
Mode Independent mode
ADC_Settings
Clock Prescaler Asynchronous clock mode divided by 1
Resolution ADC 12-bit resolution
Data Alignment Right alignment
Gain Compensation 0
Scan Conversion Mode Disabled
End Of Conversion Selection End of single conversion
Low Power Auto Wait Disabled
Continuous Conversion Mode Disabled
Discontinuous Conversion Mode Disabled
DMA Continuous Requests Disabled
Overrun behaviour Overrun data overwritten
ADC_Regular_ConversionMode
Enable Regular Conversions Enable
Enable Regular Oversampling Disable
Number Of Conversion 1
External Trigger Conversion Source Timer 2 Trigger Out event
External Trigger Conversion Edge Trigger detection on the falling edge
Rank 1
Channel Channel Vopamp3
Sampling Time 12.5 Cycles

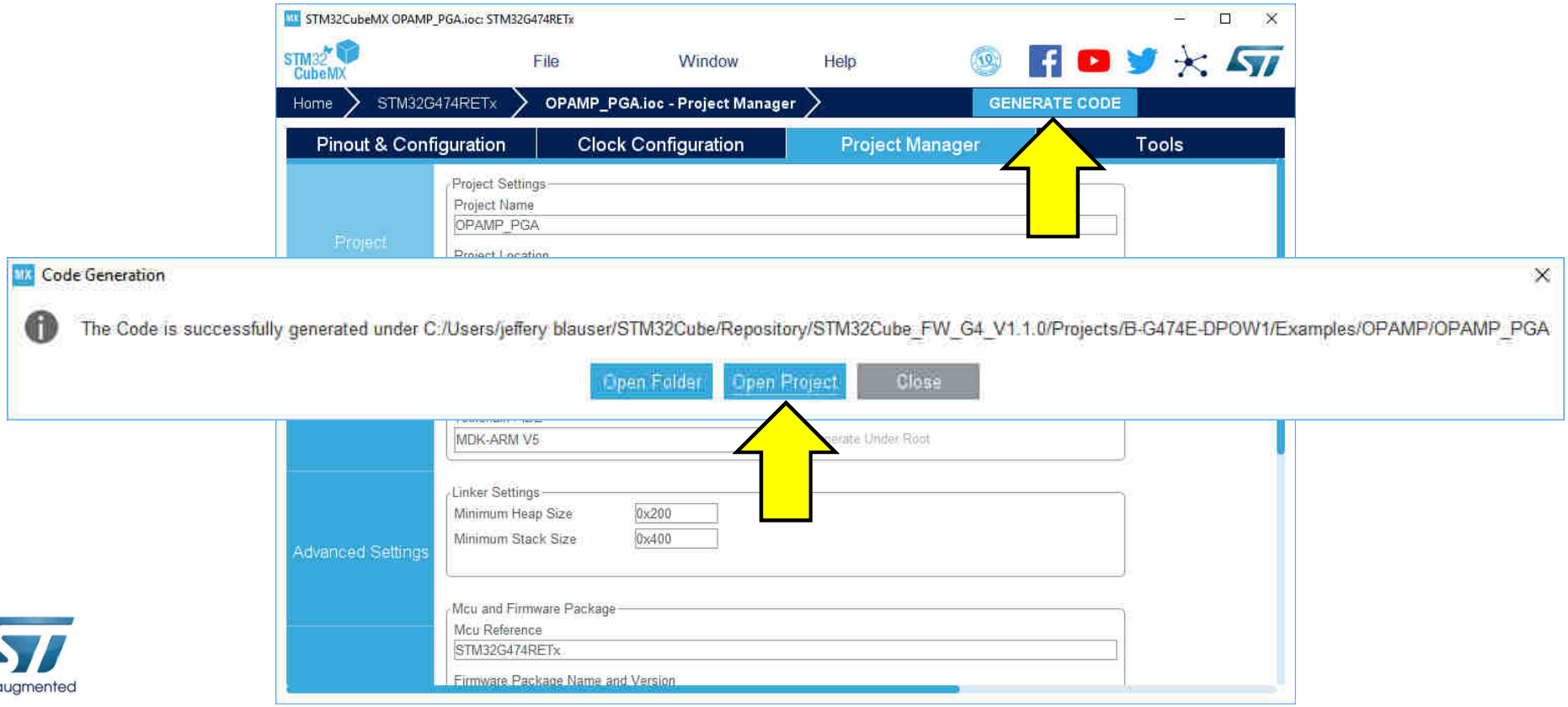
Project Manager – Keil MDK-ARM V5



FILE→SAVE PROJECT

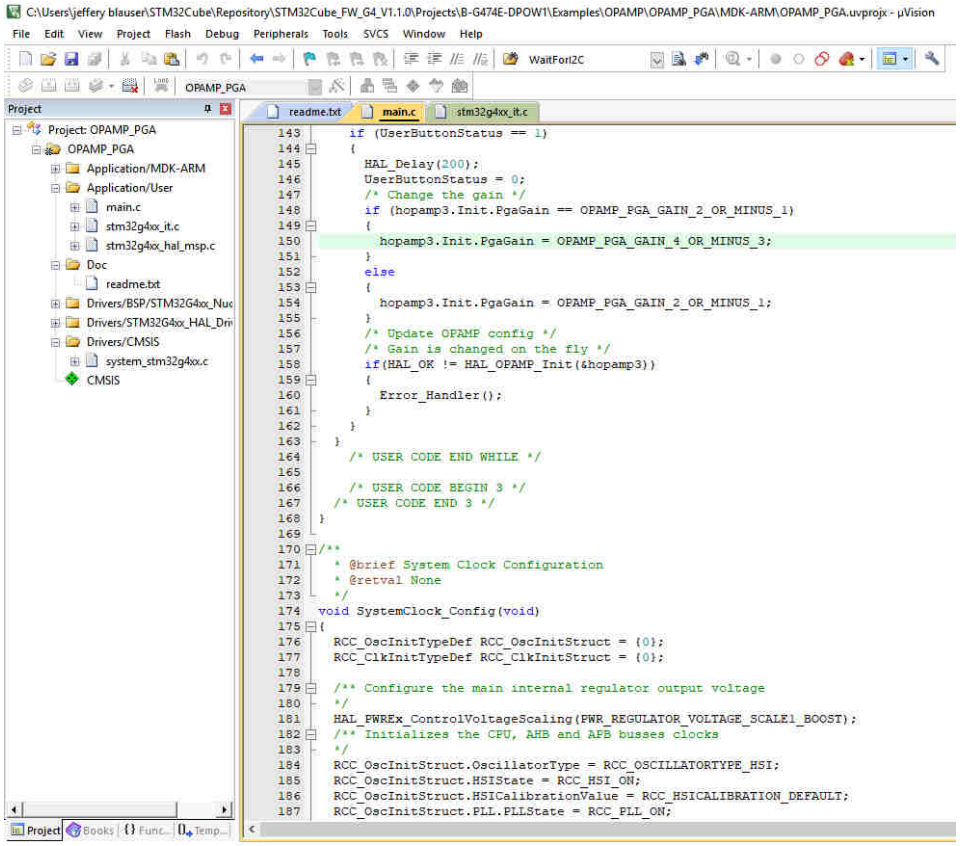


Generate Code→Open Project

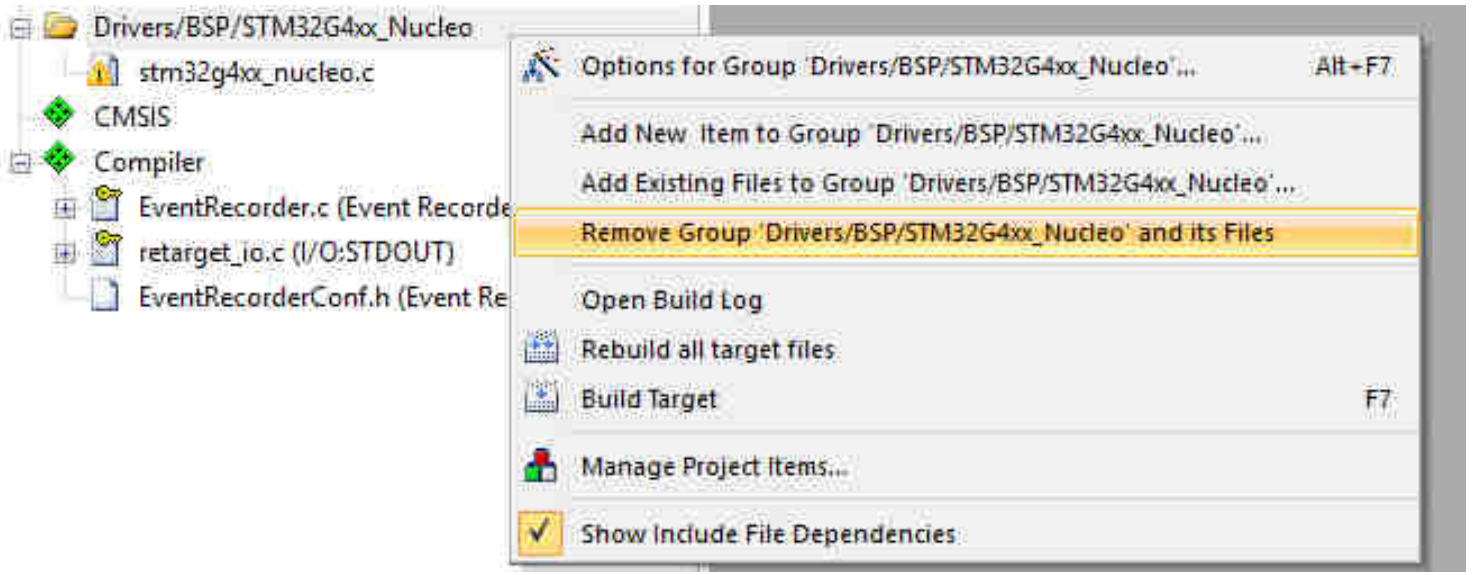


life.augmented

Keil Start up page

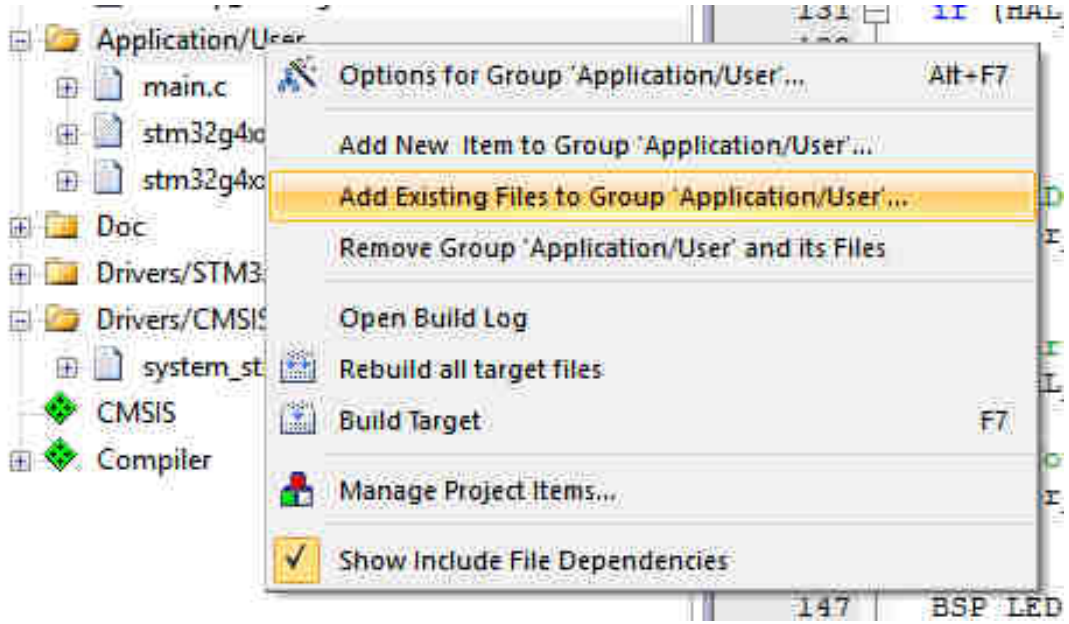


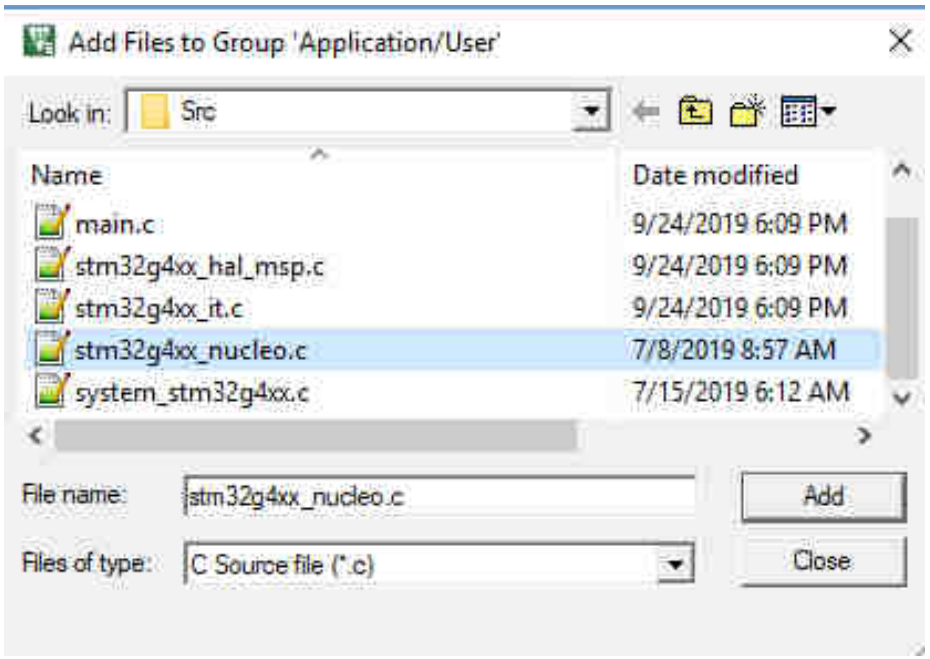
Delete Drivers Folder



Keil MDK-ARM

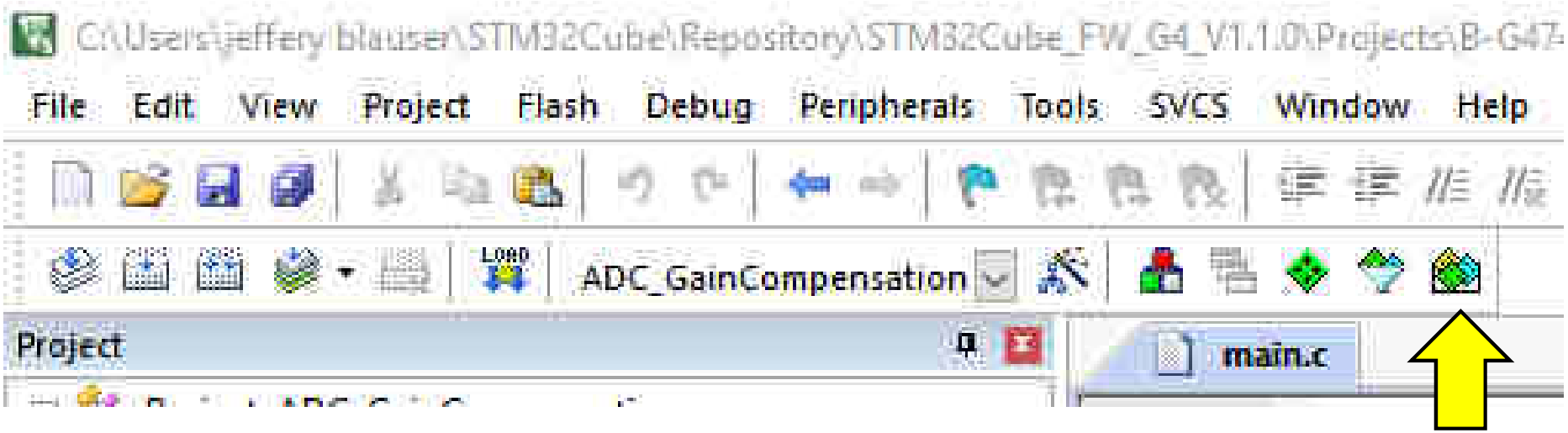
Application/User folder stm32g4xx_nucleo.c from SRC folder - CLOSE





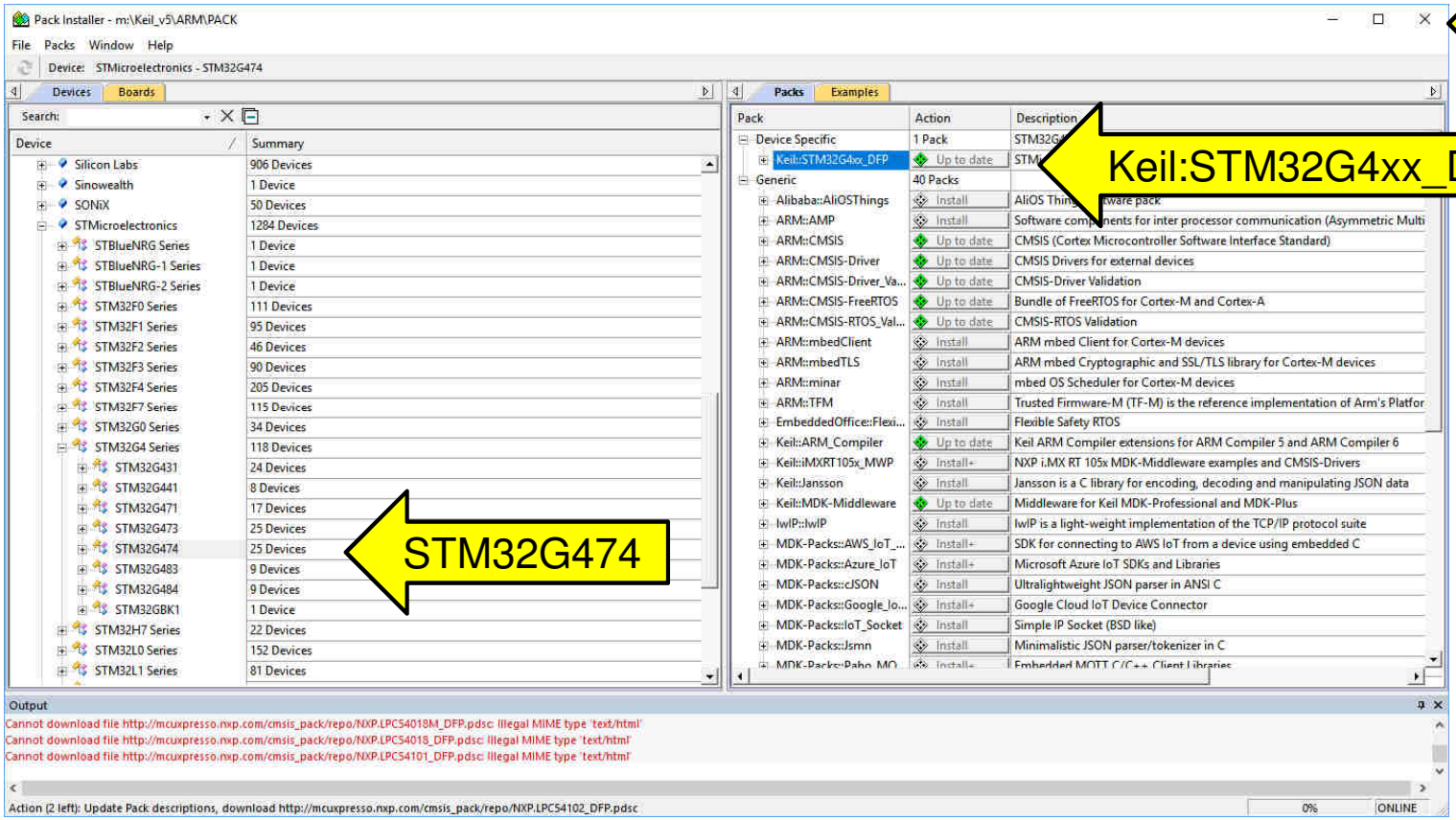
Keil MDK-ARM

Check the Device Pack



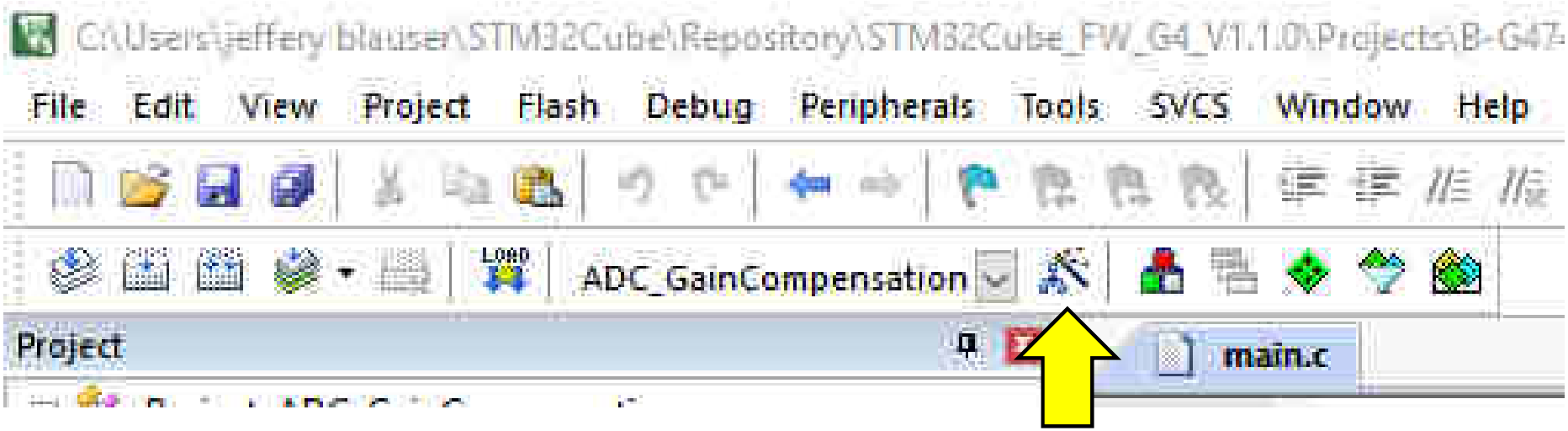
Keil MDK-ARM

G4 Device Pack

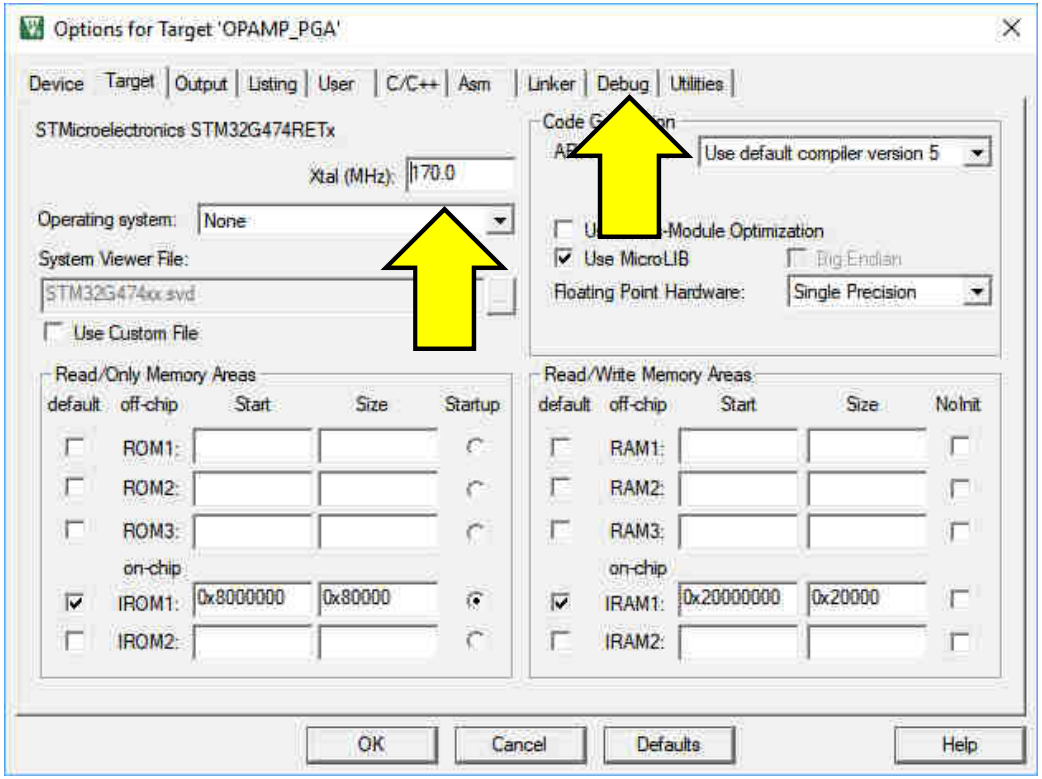


Keil MDK-ARM

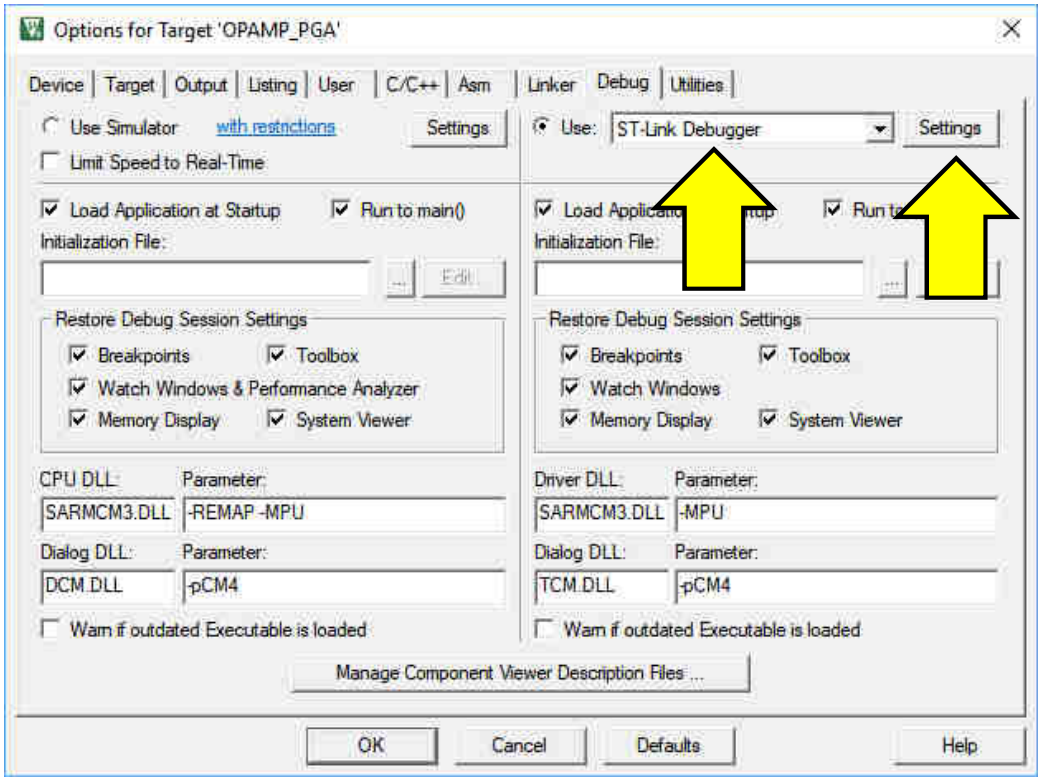
Check the Options



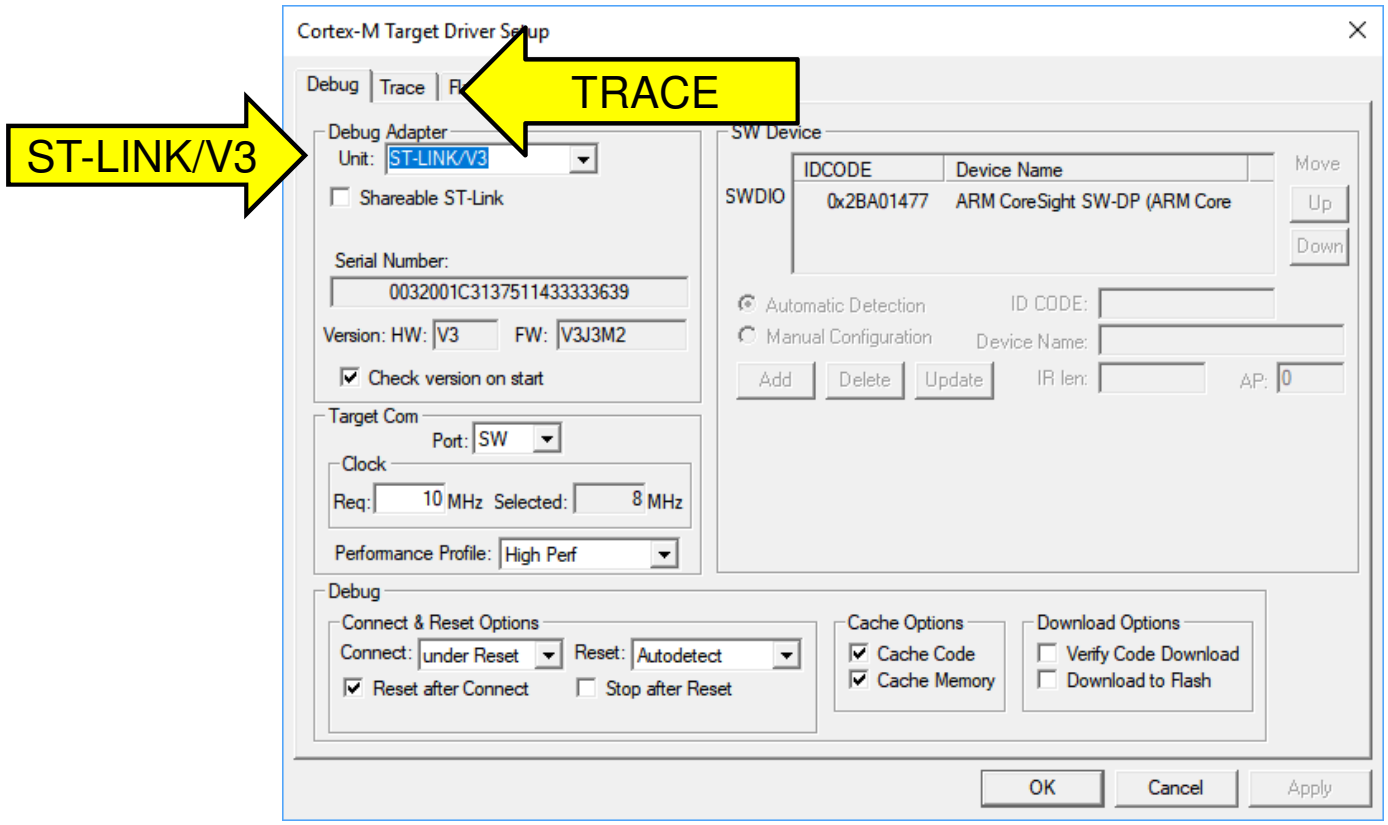
Clock speed and Debug



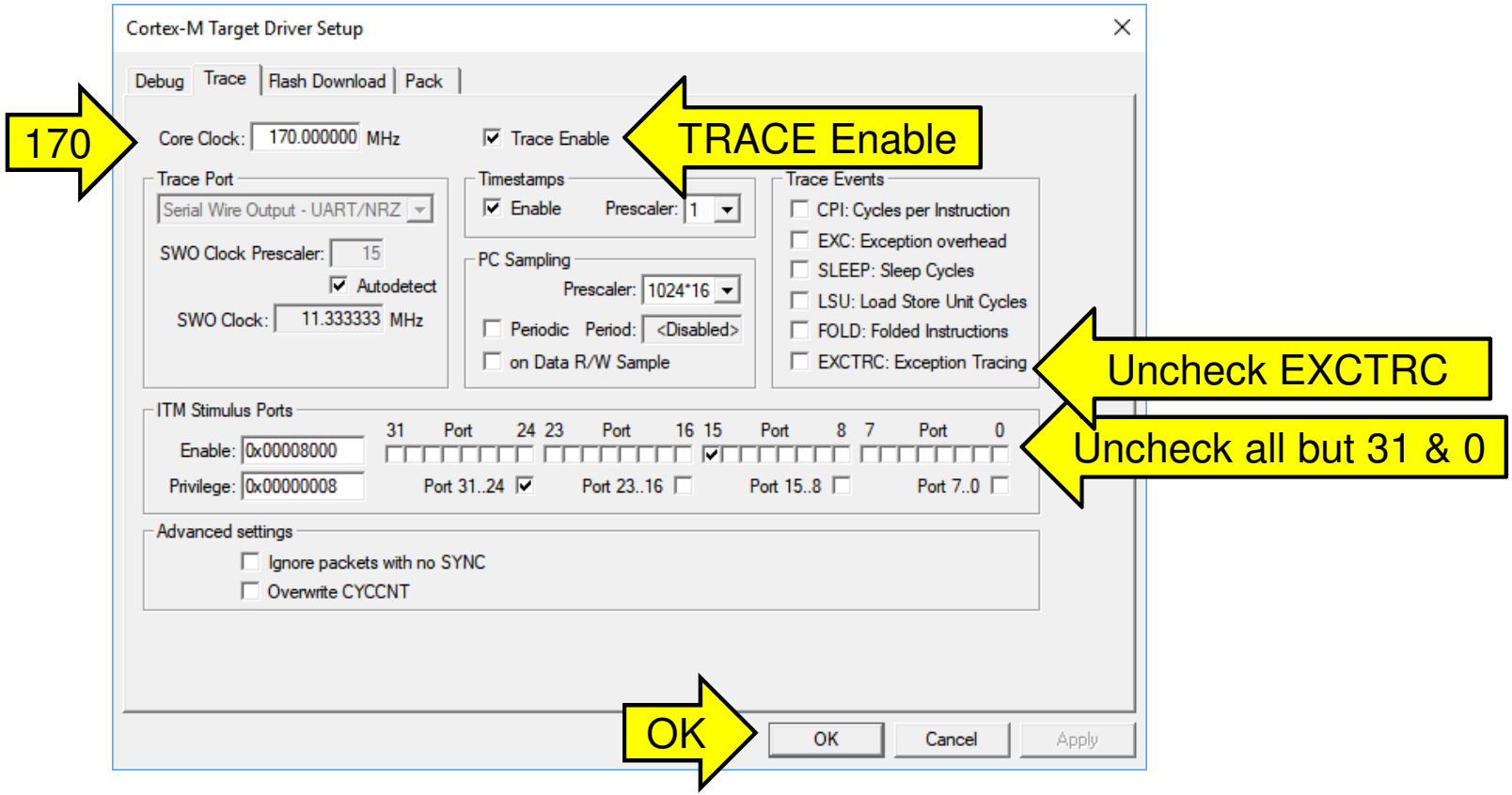
ST-Link Debugger - Settings



ST-Link/V3 & TRACE

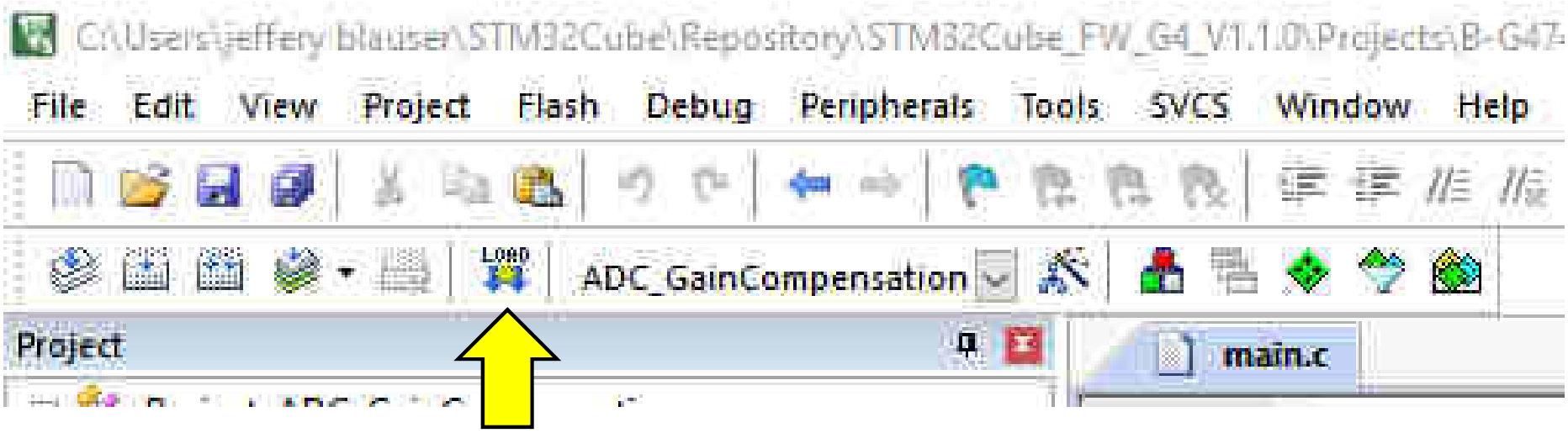


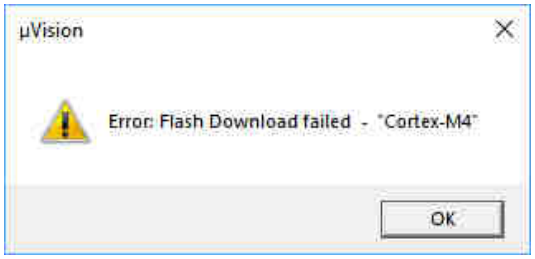
Core Clock & TRACE Enable



Keil MDK-ARM

Load



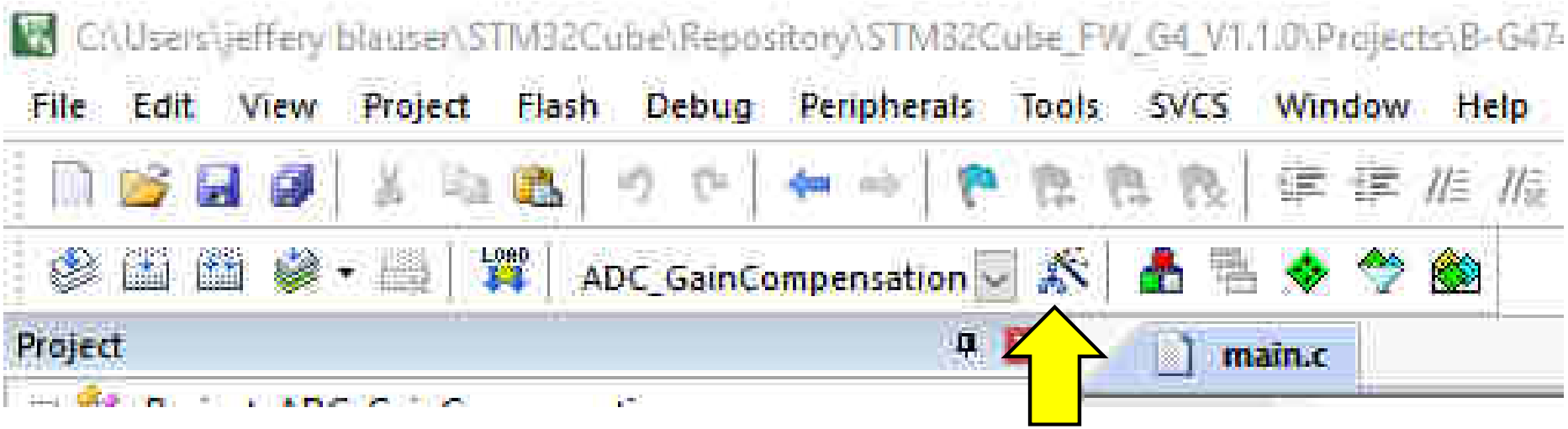


Build Output

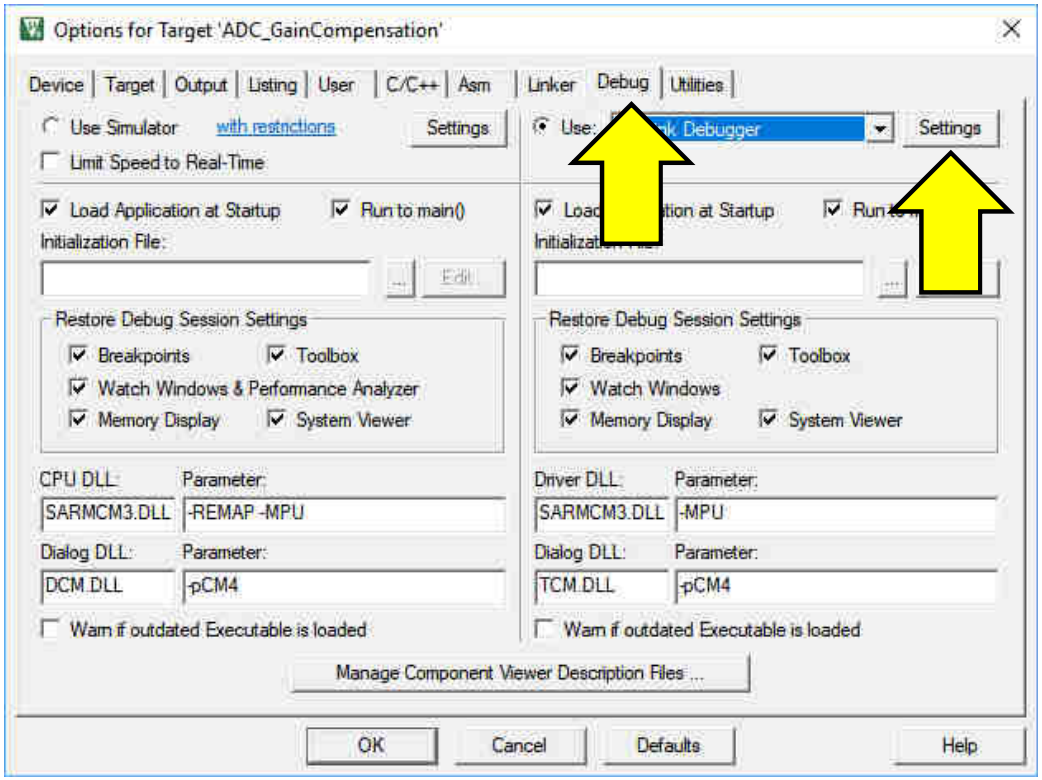
```
Contents mismatch at: 08001860H  (Flash=38H  Required=C5H) !
Contents mismatch at: 08001861H  (Flash=46H  Required=F8H) !
Contents mismatch at: 08001862H  (Flash=FEH  Required=88H) !
Contents mismatch at: 08001863H  (Flash=F7H  Required=00H) !
Contents mismatch at: 08001864H  (Flash=B9H  Required=60H) !
Contents mismatch at: 08001865H  (Flash=FCH  Required=6CH) !
Too many errors to display !
Error: Flash Download failed  -  "Cortex-M4"
```

Keil MDK-ARM

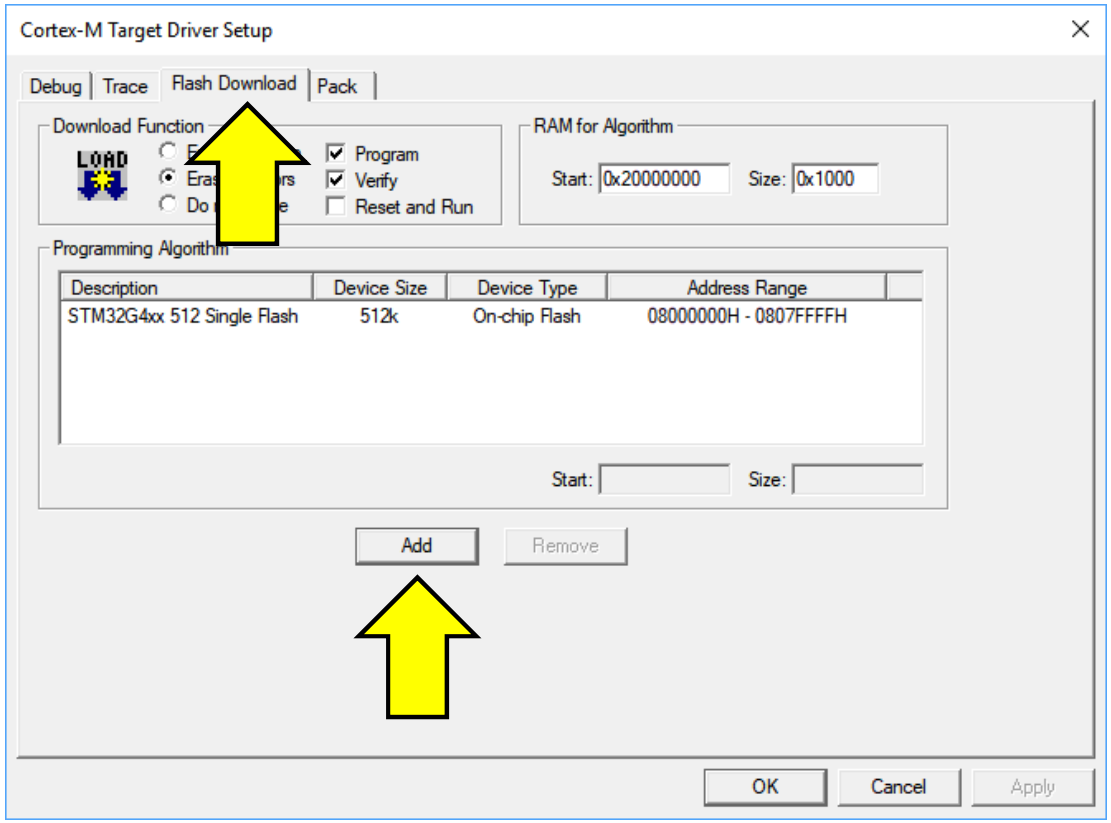
Options



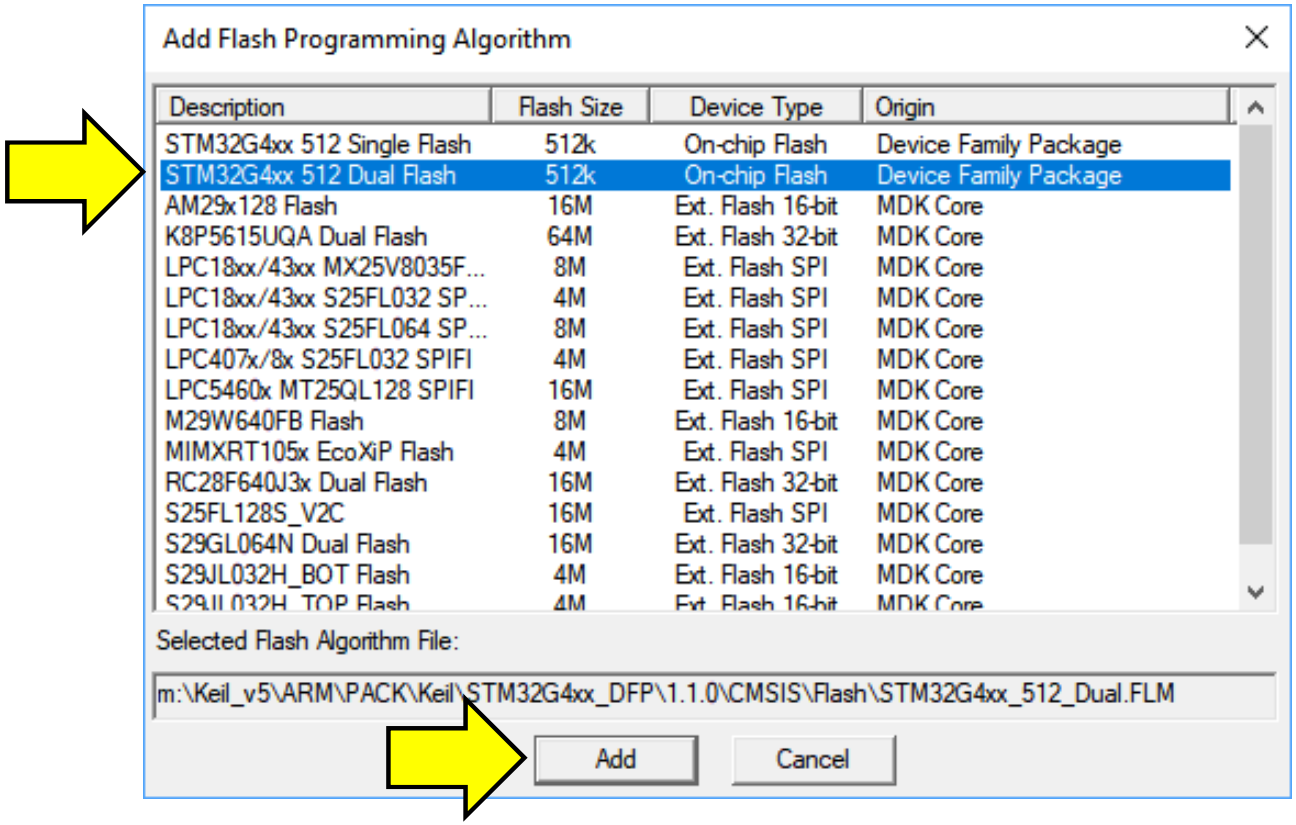
ST-Link Debugger - Settings



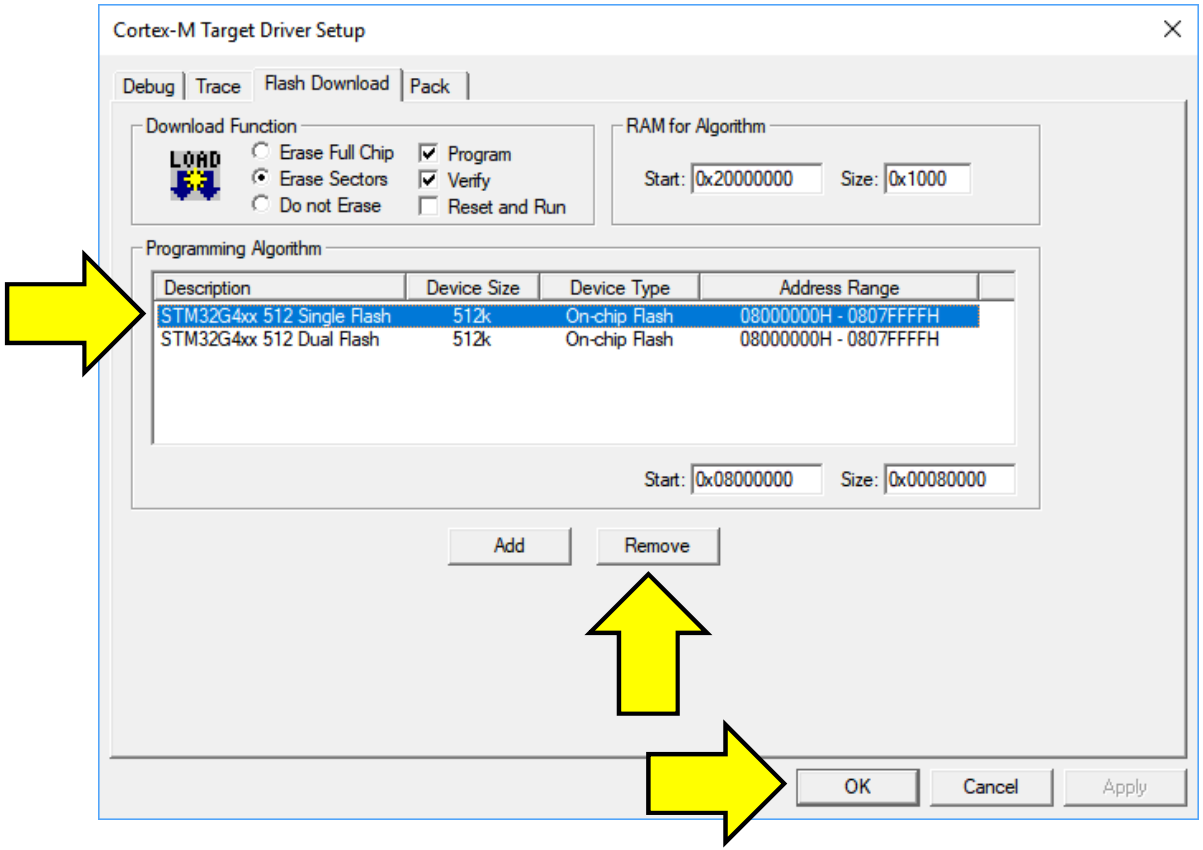
Flash Download → Add



STM32G4xx 512 Dual Flash

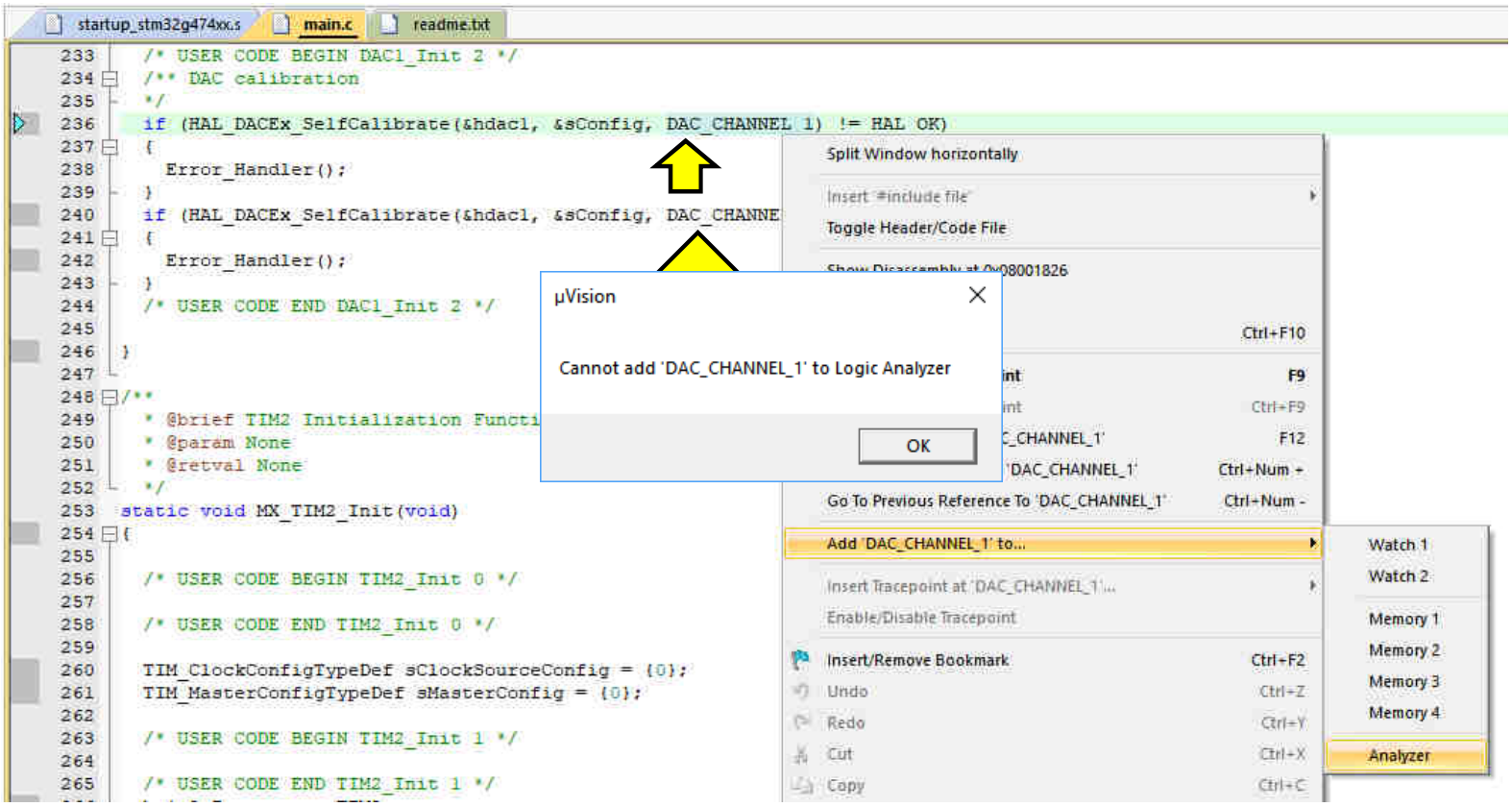


Remove Single Flash

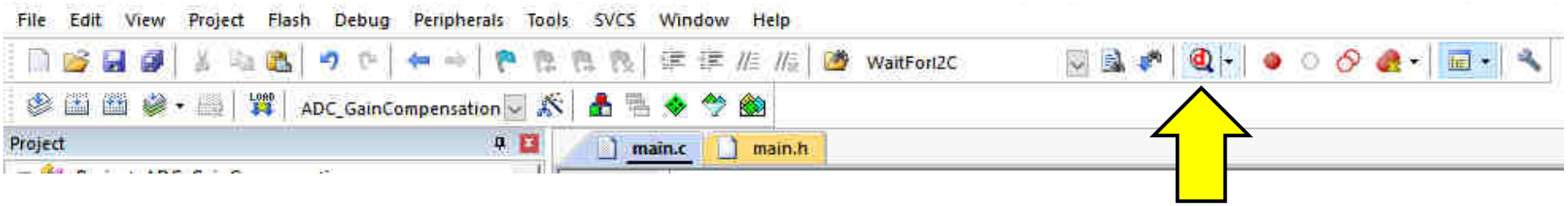


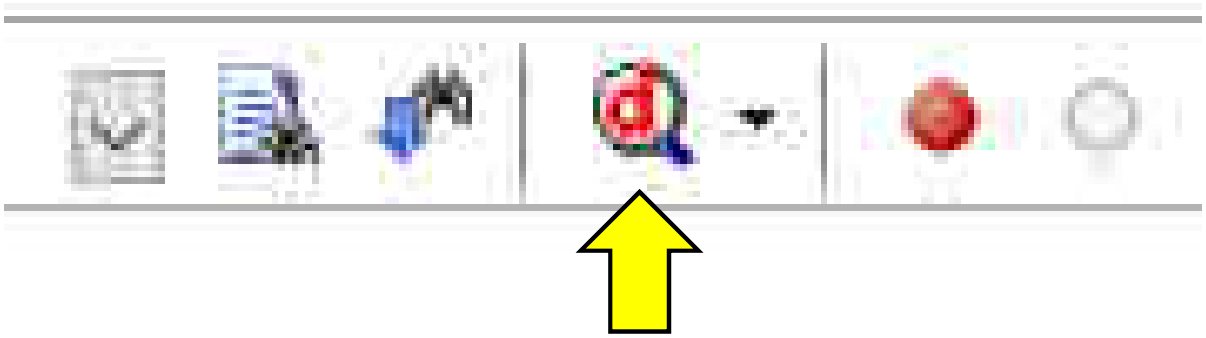
Keil MDK-ARM

Place variable in the Analyzer – main.c about lines 235 & 240



Debugger – Exit back to the Editor





Add 3 new variables in main.c about line 59

```
54  /* USER CODE BEGIN PV */
55  /* Private variables ---------------------------
56  /* set to 1 after User push-button interrupt  */
57  __IO uint32_t UserButtonStatus = 0;
58
59  uint16_t  Display_DAC1DOR1;
60  uint16_t  Display_DAC3DOR2;
61  uint16_t  Display_ADC3IN1;
```

See C:\2019_STM32G4_Workshop\Labs\OPAMP CodeLinesForOpAmp.txt if you prefer to copy and paste
Or open the file from the Doc folder in the Editor

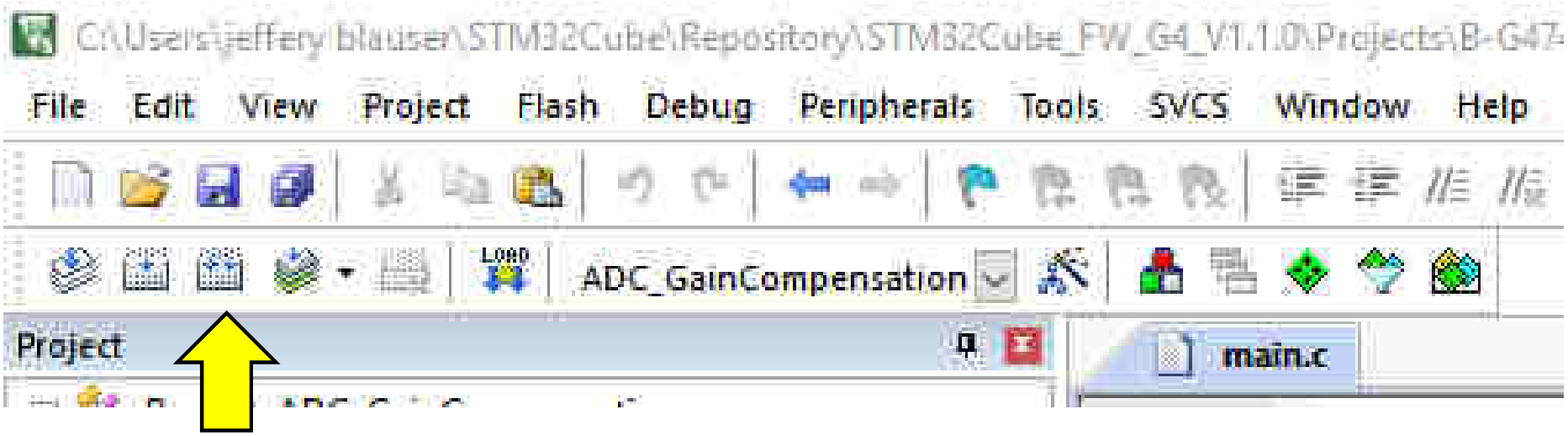
Add new lines in main.c about line 177

```
174           Error_Handler();
175         }
176       }
177       /* Display the global variables in the Logic Analyzer */
178       Display_DAC1DOR1 = DAC1->DOR1;
179       HAL_Delay(1);   //1 millisecond delay to transfer data to the PC
180       Display_DAC3DOR2 = DAC3->DOR1;
181       HAL_Delay(1);   //1 millisecond delay to transfer data to the PC
182       Display_ADC3IN1 = ADC3->DR;
183       HAL_Delay(1);   //1 millisecond delay to transfer data to the PC
184     }
185       /* USER CODE END WHILE */
```

See C:\2019_STM32G4_Workshop\Labs\OPAMP CodeLinesForOpAmp.txt if you prefer to copy and paste
Or open the file from the Doc folder in the Editor

Keil MDK-ARM

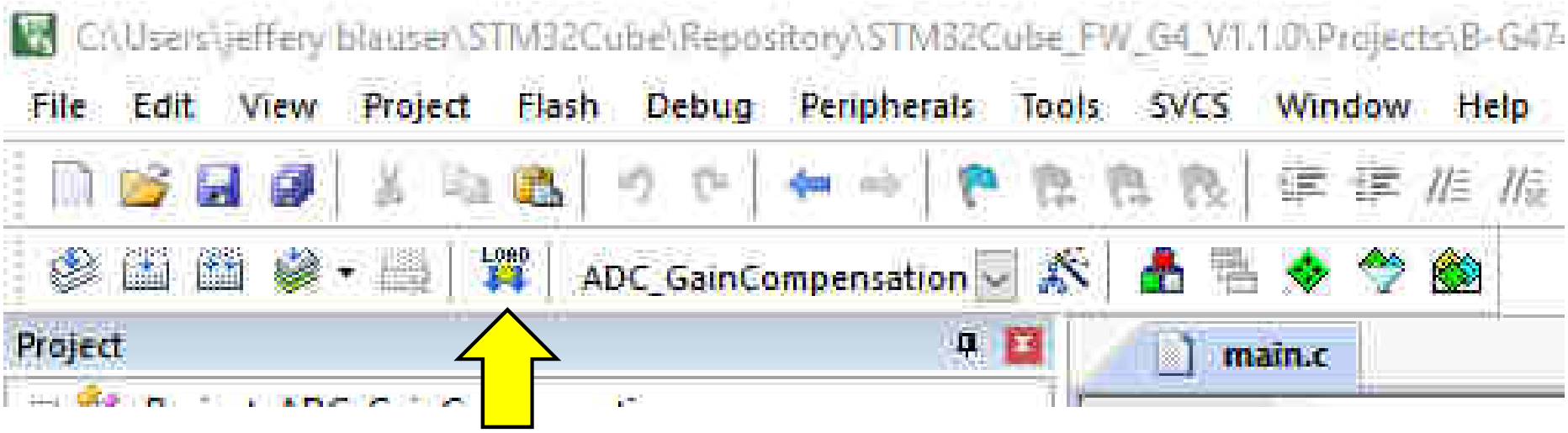
Build




```
Build Output
Program Size: Code=6516 RO-data=596 RW-data=44 ZI-data=1268
FromELF: creating hex file...
OPAMP_PGA\OPAMP_PGA.hex: Warning: Q9931W: Your license for feature Keil will expire in 30 days
Finished: 0 information, 1 warning and 0 error messages.
"OPAMP_PGA\OPAMP_PGA.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:01:22
```

Keil MDK-ARM

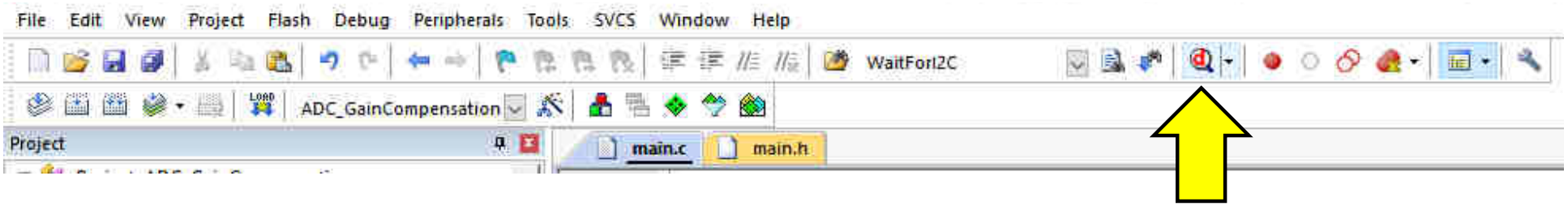
Load

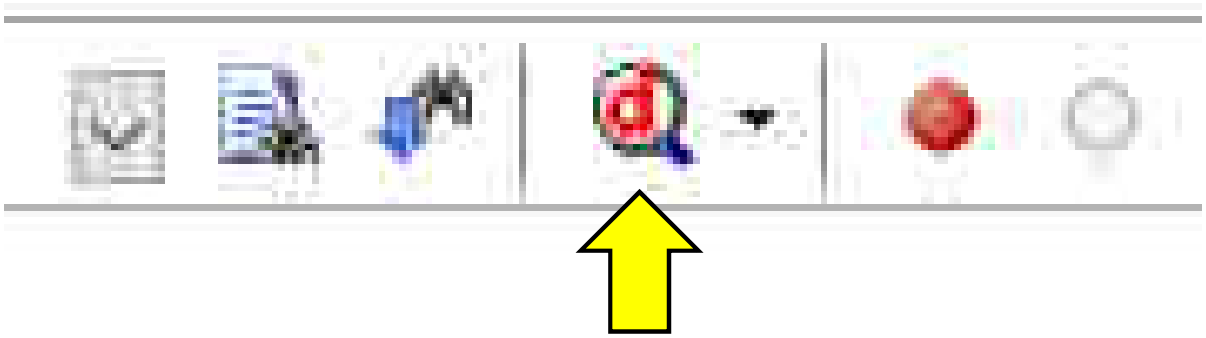



```
Build Output
Build Time Elapsed:  00:01:22
Load "OPAMP_PGA\\OPAMP_PGA.axf"
Erase Done.
Programming Done.
Verify OK.
Flash Load finished at 17:34:57
```

Keil MDK-ARM

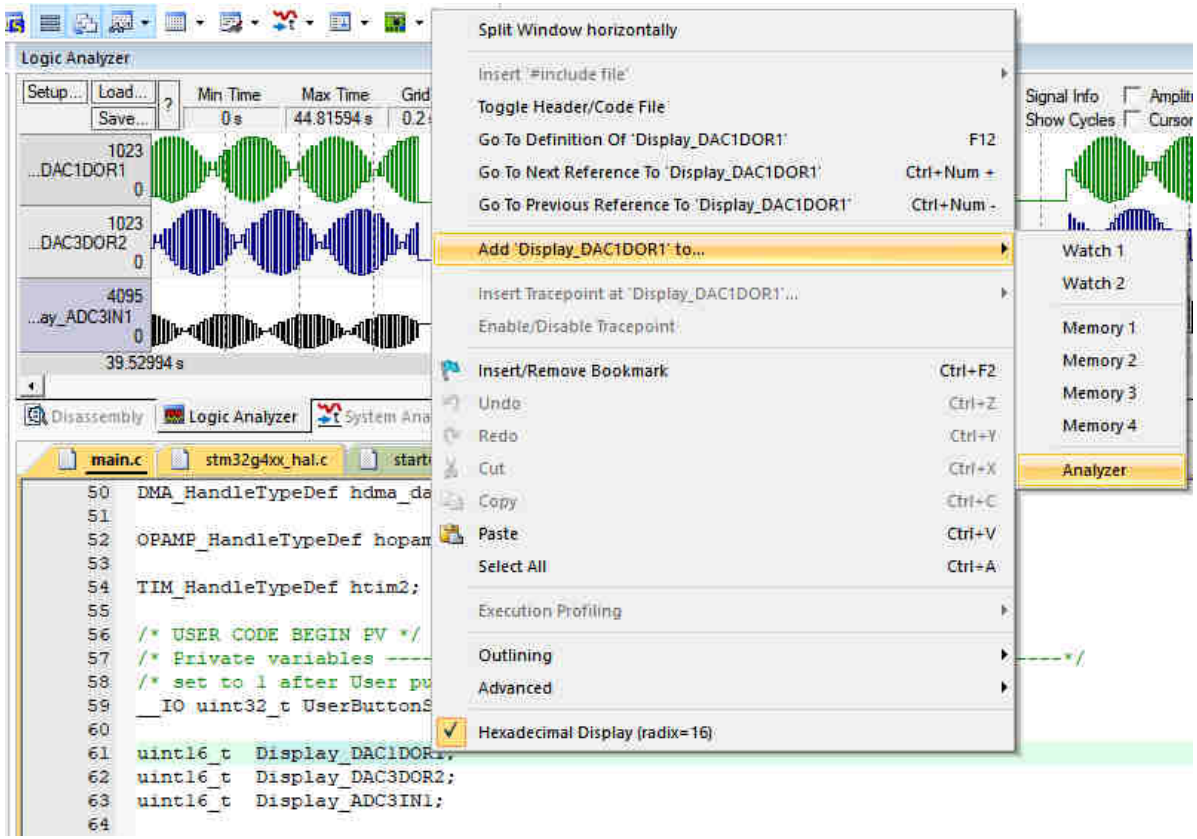
Debugger



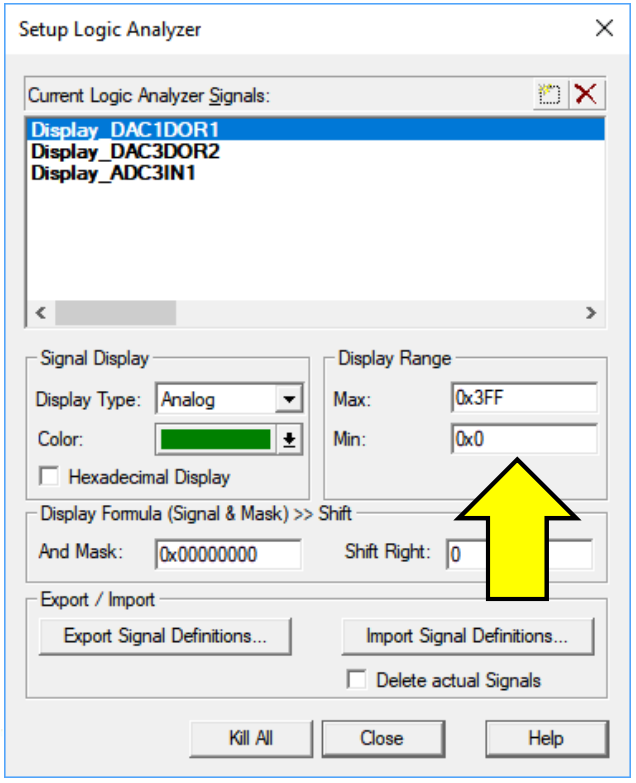


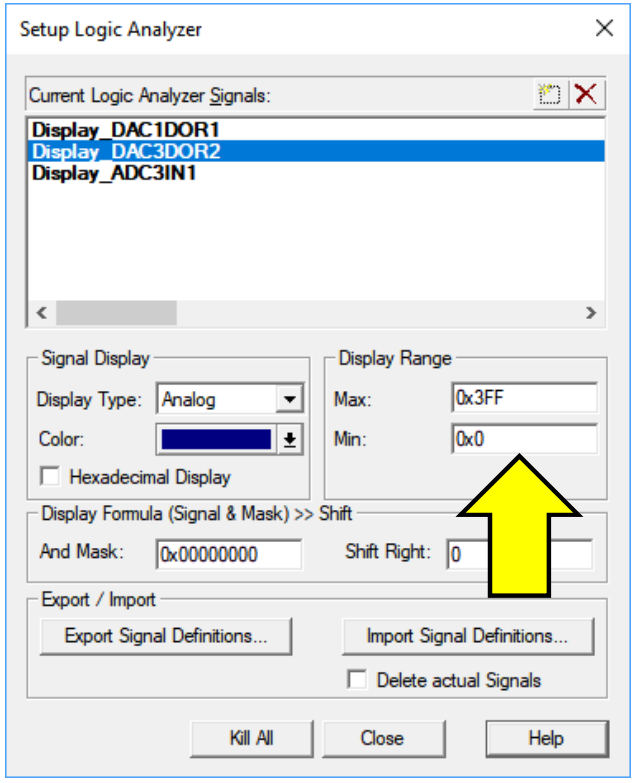
Keil MDK-ARM

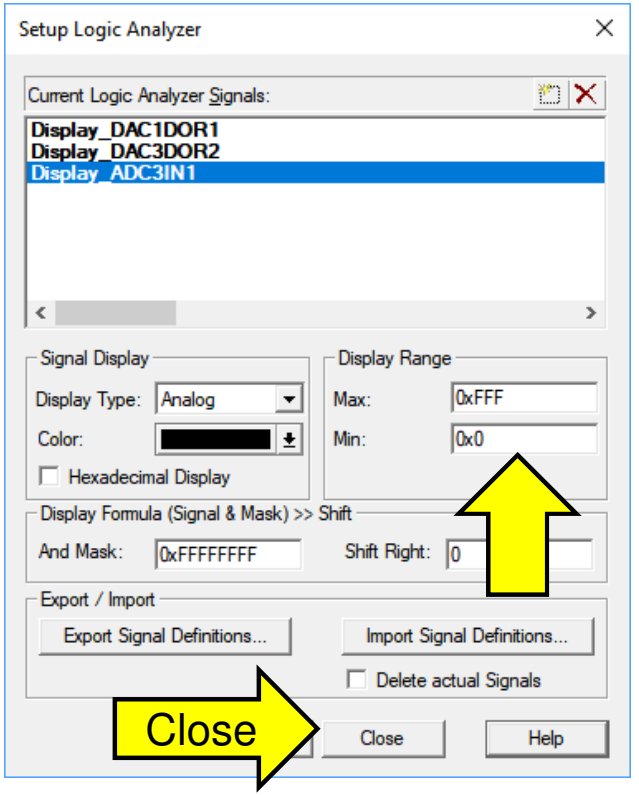
Find the 3 Display_xxx about main.c line 61 – right click on each – add to Analyzer



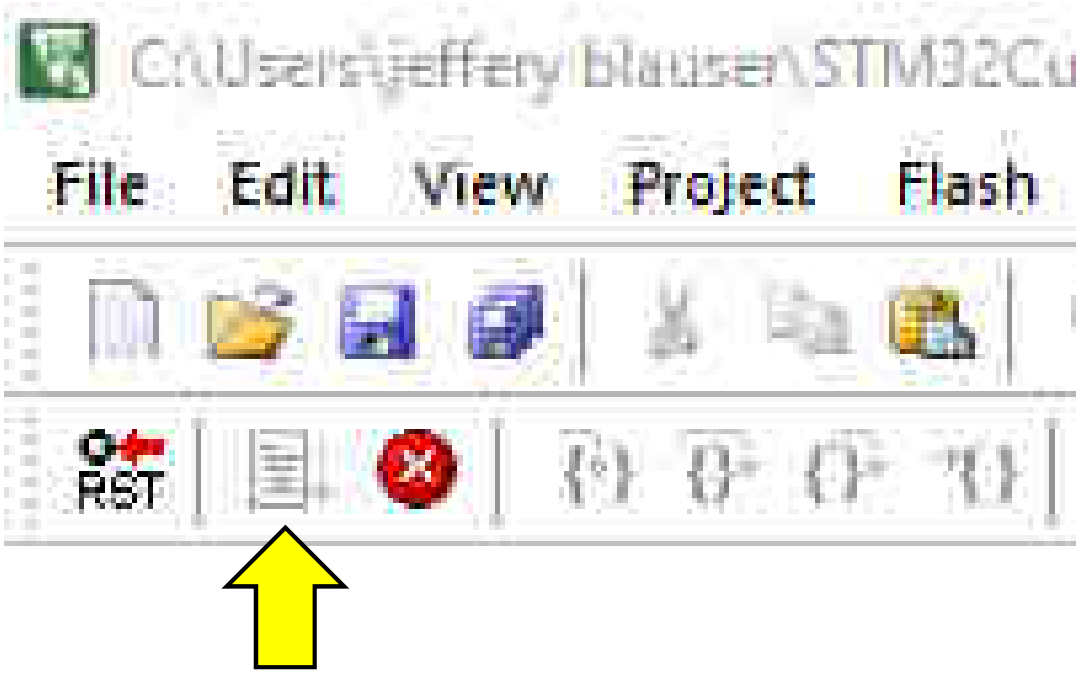
Setup – Min=0x0, Max = 0x3FF for DACs, then Min=0x0, Max = 0xFFF for ADC - Close



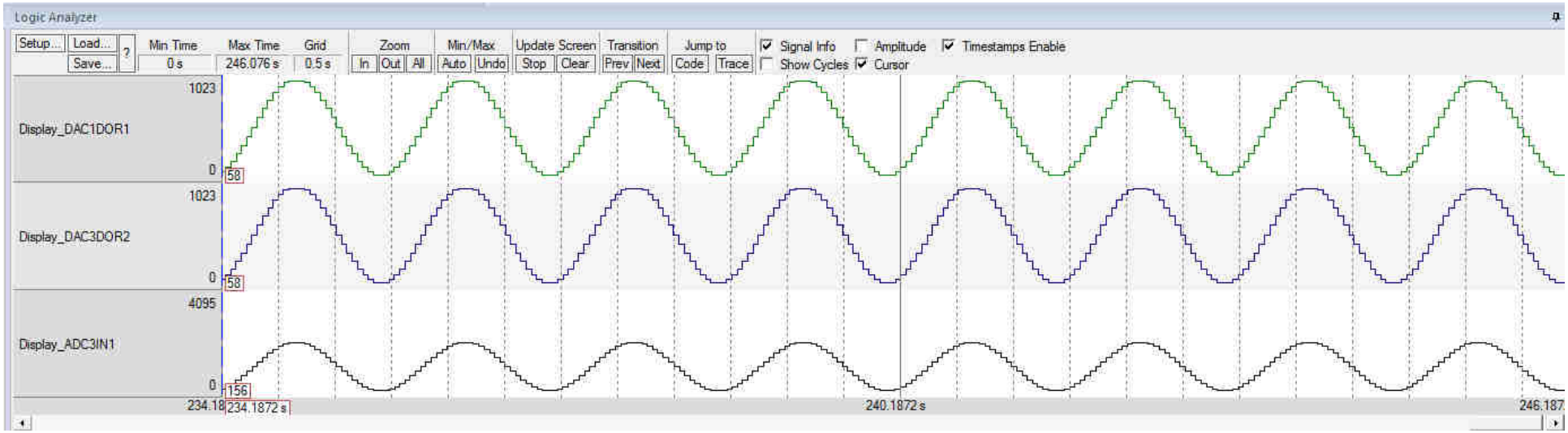


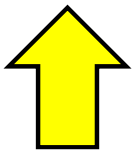


RUN



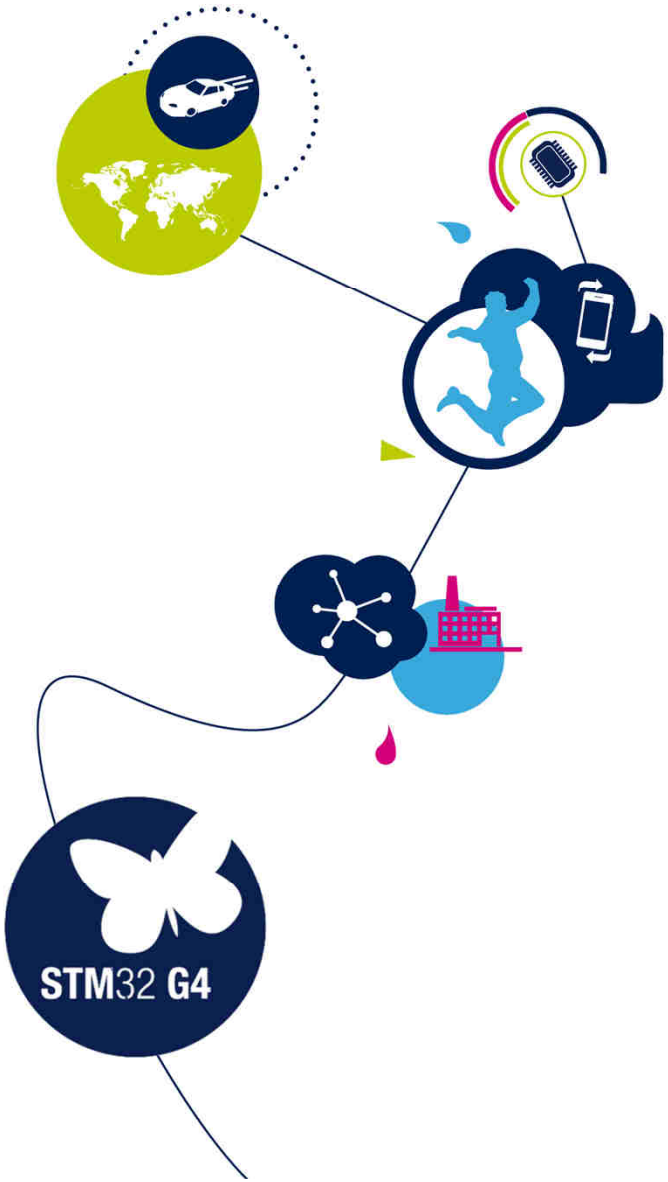
RUN


Logic Analyzer
Setup...
Load...
Save...
Min Time
0 s
Max Time
246.076 s
Grid
0.5 s
Zoom
In
Out
All
Min/Max
Auto
Undo
Update Screen
Stop
Clear
Transition
Prev
Next
Jump to
Code
Trace
Signal Info
Show Cycles
Amplitude
Cursor
Timestamps Enable
Display_DAC1DOR1
1023
0
58
Display_DAC3DOR2
1023
0
58
Display_ADC3IN1
4095
0
156
234.18
234.1872 s
240.1872 s
246.187



STM32G4 mixed-signal MCU workshop

Hands On:
CORDIC Hardware Accelerator



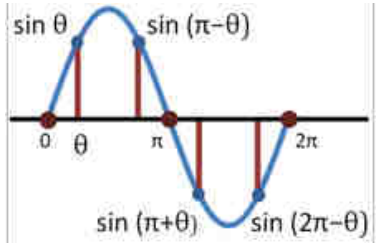
life.augmented

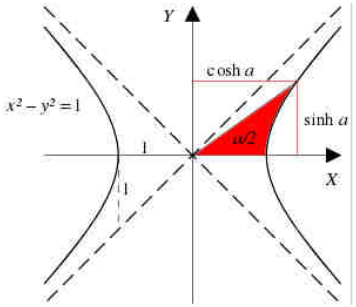
CORDIC: Main features

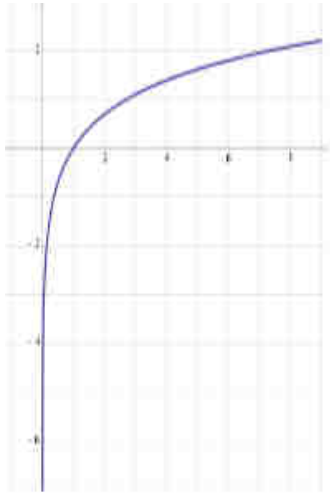
- 24-bit CORDIC rotation engine
  - Trigonometric and Hyperbolic functions
- Circular and Hyperbolic modes
- Rotation and Vectoring modes
- Programmable precision
- Low latency AHB slave interface
- DMA read and write channels

Cordic: Main features

- Supported functions:
  - Sine, Cosine: Vector rotation (polar to rectangular)
  - Arctangent, Modulus: Vector translation (rectangular to polar)
  - Hyperbolic cosine
  - Hyperbolic sine
  - Hyperbolic tangent
  - Natural logarithm
  - Square root







- Fixed point signed input and output
  - Q1.31 or Q1.15 format
- Performance improvement relative to software
  - Sine and Cosine functions are typically 5x faster than software execution

CORDIC - Precision

- Precision is dependent on number of iterations

Function	Number of iteration (precision)	Number of cycles	Max residual error	
			q1.31 format	q1.15. format
Sin, Cos, Phase, Mod, Atan	4	1	2^{-3}	2^{-3}
	8	2	2^{-7}	2^{-7}
	12	3	2^{-11}	2^{-11}
	16	4	2^{-15}	2^{-15}
	20	5	2^{-18}	2^{-16}
	24	6	2^{-19}	2^{-16}

CORDIC - Precision

Function	Number of iteration (precision)	Number of cycles	Max residual error	
			q1.31 format	q1.15. format
Sinh, Cosh, Atanh, Ln	4	1	2^{-2}	2^{-2}
	8	2	2^{-6}	2^{-6}
	12	3	2^{-10}	2^{-10}
	16	4	2^{-13}	2^{-13}
	20	5	2^{-17}	2^{-15}
	24	6	2^{-18}	2^{-15}
Sqrt	4	1	2^{-7}	2^{-7}
	8	2	2^{-14}	2^{-14}
	12	3	2^{-19}	2^{-15}

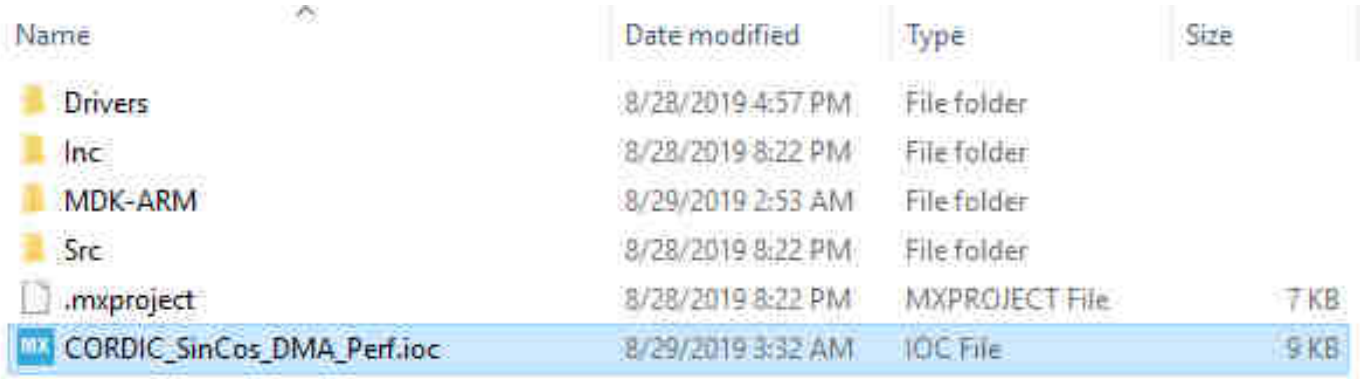
CORDIC accelerator

Objective:

- To show how to configure the CORDIC accelerator.
  - Calculate the SIN and COS of angles with 6 cycles of precision.
- Enable the Event Recorder to measure the time for the CORDIC and ARM Math Library functions.
- Display CORDIC results in the Logic Analyzer.

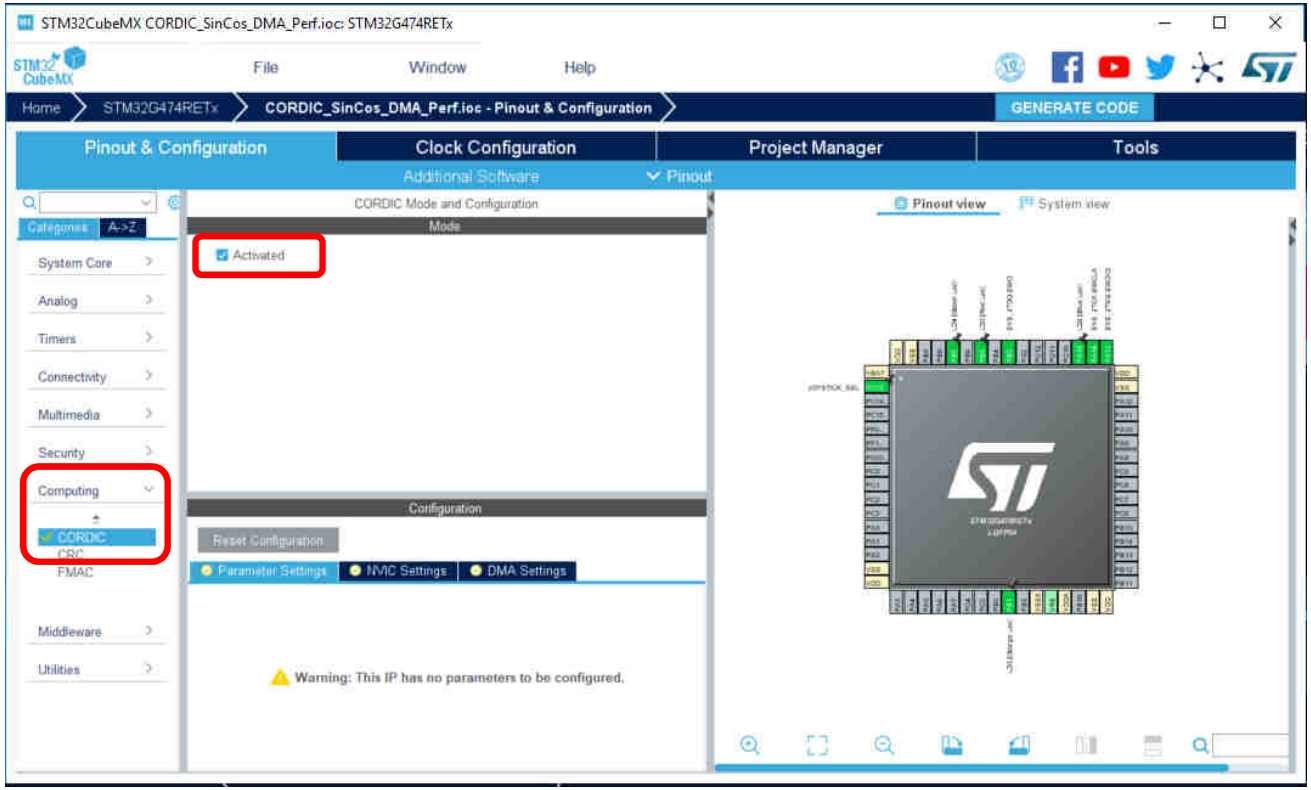
CORDIC configuration

- Open CubeMX project “CORDIC_SinCos_DMA_Perf.ioc” file
  - \labs\Cordic\CORDIC_SinCos_DMA_Perf



CORDIC configuration

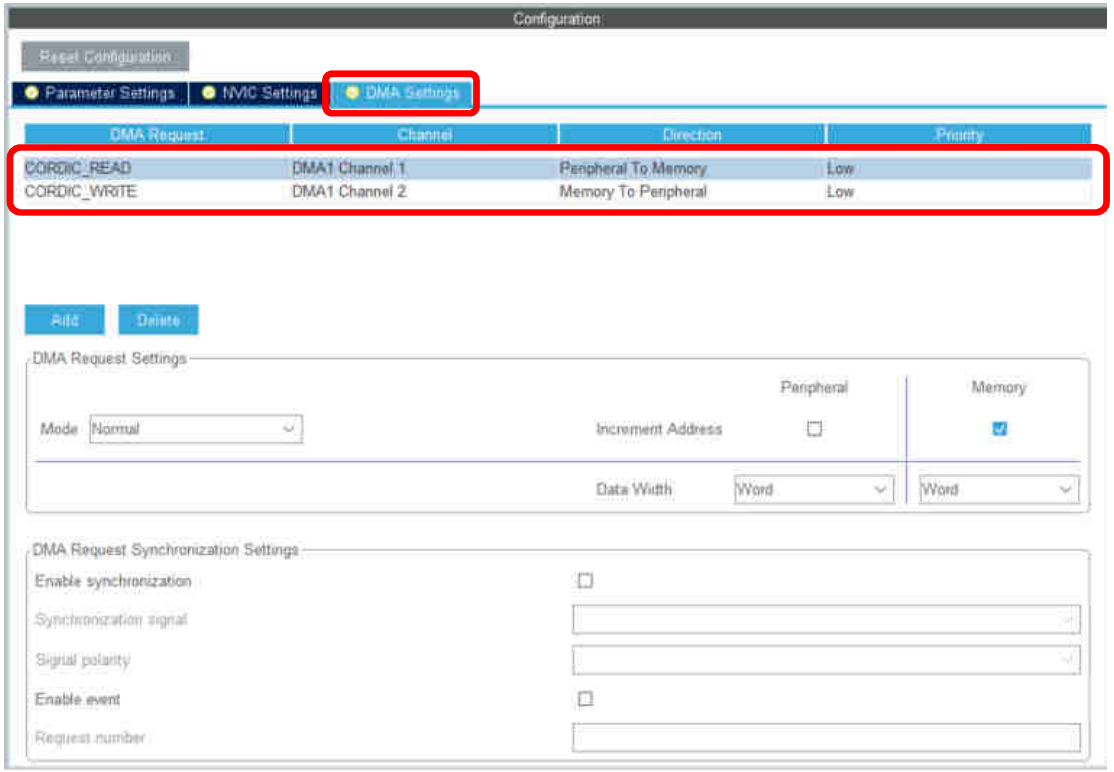
- Expand “Computing” Category
- Select “CORDIC”
  - CHECK “Activated”



life.augmented

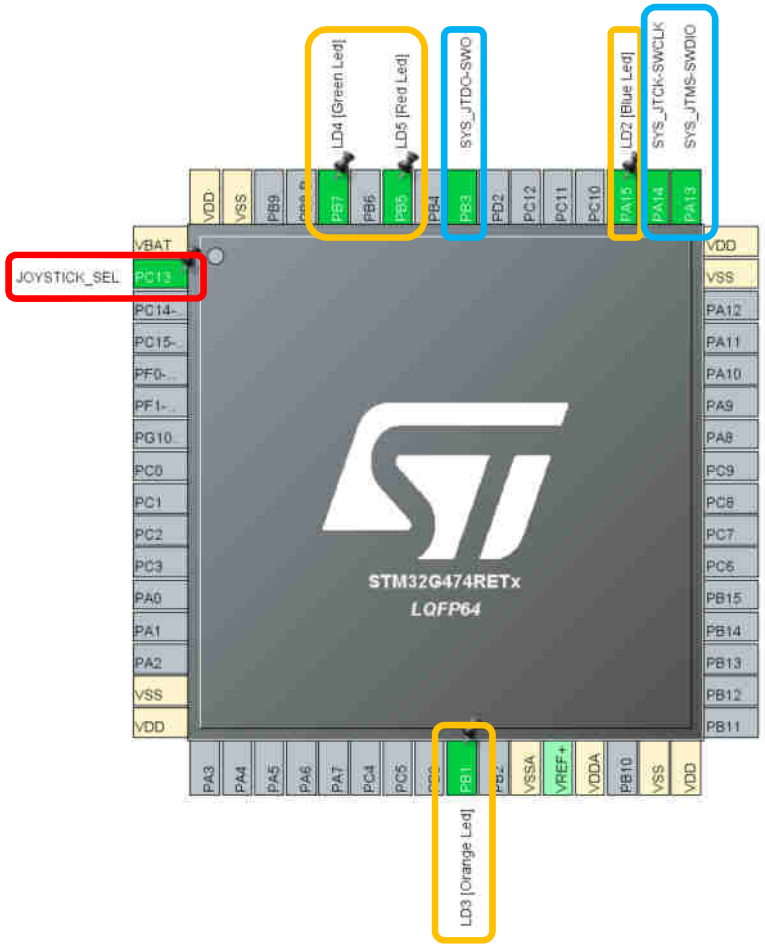
CORDIC configuration

- DMA Request is enabled to transfer data in and out of CORDIC
- 32-bit Data Width (Word)



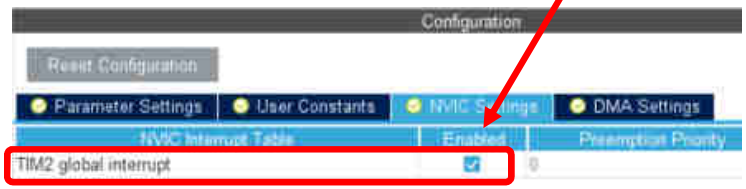
Additional Peripheral configuration

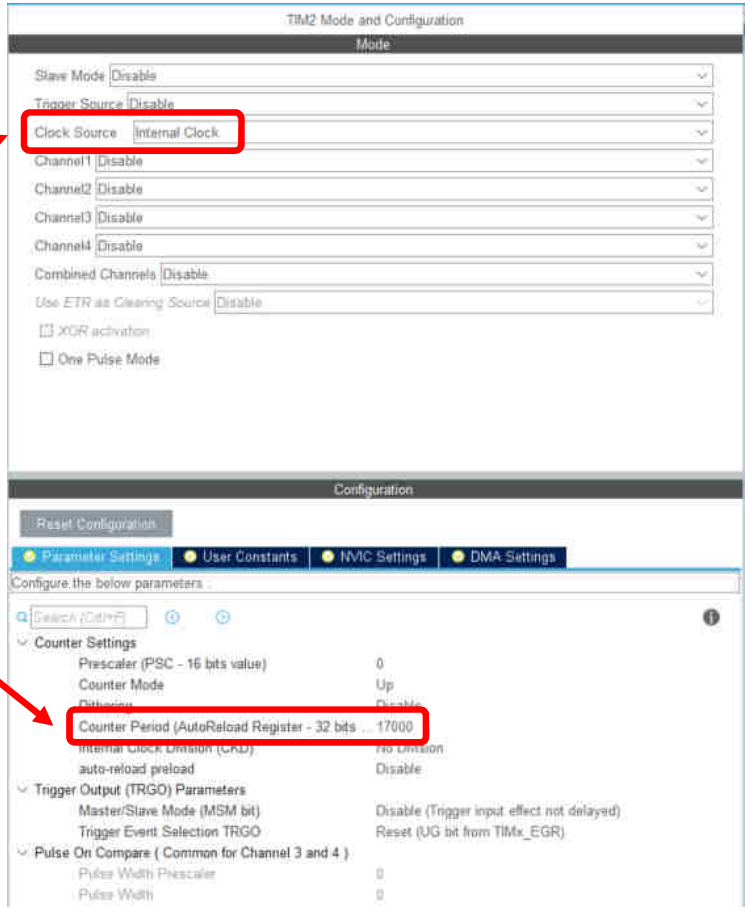
- JOYSTICK_SEL: EXTI13
  - PC13
- LD2-5: GPIO_Output
  - PA15, PB1, PB5 and PB7
- Serial Wire Debug with Trace (SWO):
  - PA13, PA14 and PB3



Additional Peripheral configuration

- Timer 2: Debug Logic Analyzer
- Internal Clock Source = 170 MHz
- Counter Period = 17000
  - 10 kHz timebase interrupt
- TIM2 global interrupt Enabled

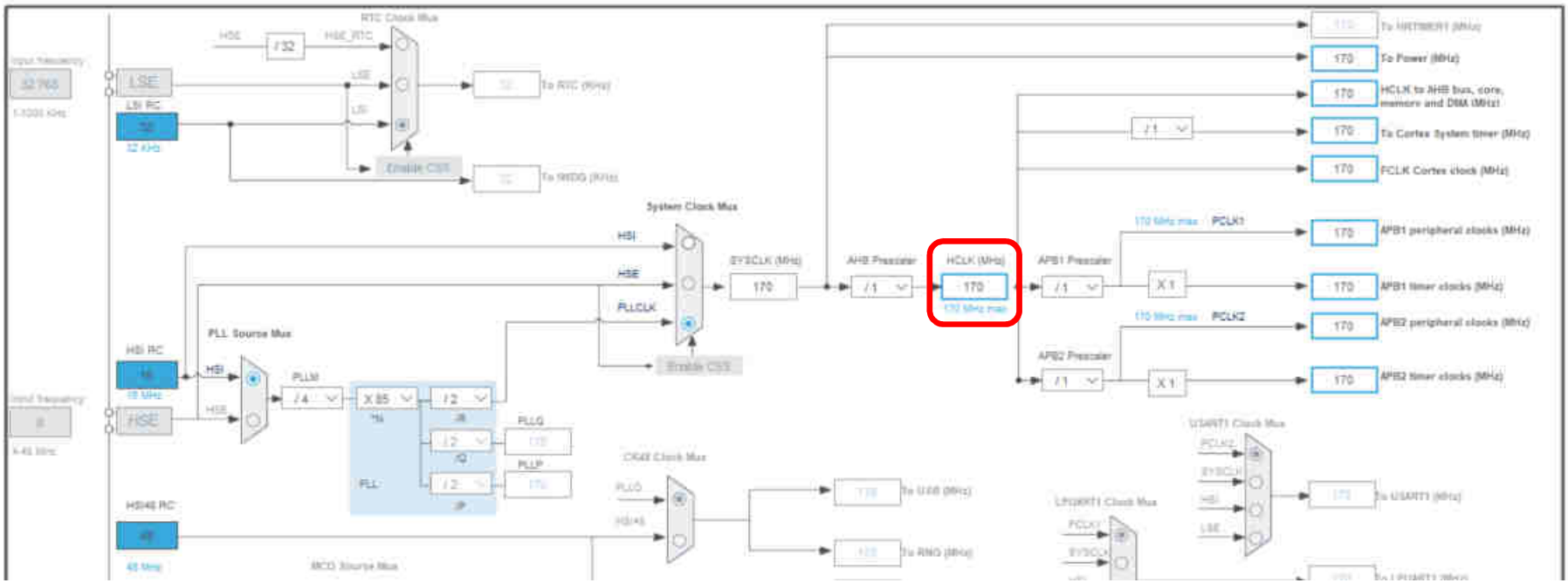




life.augmented

Hands-On: CORDIC

- Validate “HCLK (MHz)” is set to 170



life.augmented

Hands-On: CORDIC

- Generate Code

- Open Project in MDK-ARM

Hands-On: CORDIC

- aAnglesCordic - Array of angles for CORDIC Q1.31 format (Line 57)

```
56  /* Array of angles for CORDIC Q1.31 format, regularly incremented from 0 to 2*pi */
57  static int32_t aAnglesCordic[ARRAY_SIZE] =
58  {
59    0x00000000, 0x04000000, 0x08000000, 0x0C000000,
```

- aAnglesLib – Array of angles for ARM Math library (Line 78)

```
77  /* Array of angles for arm math library Q1.31 format, regularly incremented from 0 to 2*pi */
78  static int32_t aAnglesLib[ARRAY_SIZE] =
79  {
80    0x00000000, 0x02000000, 0x04000000, 0x06000000,
```

- aRefSin & aRefCos – Array of Sin & Cos result reference (Line 99 & 120)

```
 98  /* Array of reference sines in Q1.31 format */
 99  static int32_t aRefSin[ARRAY_SIZE] =
100  {
101    0x00000000, 0x0C8BD35E, 0x18F8B83C, 0x25280C5D,
```

```
119  /* Array of reference cosines in Q1.31 format */
120  static int32_t aRefCos[ARRAY_SIZE] =
121  {
122    0x80000000, 0x7F62368F, 0x7D8A5F3F, 0x7A7D055B,
```

- aResults – Array of calculation results (Line 142)

```
140 /* Array of calculation results in Q1.31 format.
141    Will contain alternatively Sine and Cosine of input angles */
142 static int32_t aResults[2 * ARRAY_SIZE];
```

CORDIC configuration

- Sine calculation function (Line 219)
- 6 cycles of precision (Line 220)
- No Scaling (Line 221)
- One input angle data (Line 222)
  - Q1.31 format for input data (Line 224)
- Two output data (Sine and Cosine) (Line 223)
  - Q1.31 format for input data (Line 225)

```
217   /* USER CODE BEGIN 2 */
218   /*##-2- Configure the CORDIC peripheral ###################
219   sCordicConfig.Function         = CORDIC_FUNCTION_SINE;
220   sCordicConfig.Precision        = CORDIC_PRECISION_6CYCLES;
221   sCordicConfig.Scale            = CORDIC_SCALE_0;
222   sCordicConfig.NbWrite          = CORDIC_NBWRITE_1;
223   sCordicConfig.NbRead           = CORDIC_NBREAD_2;
224   sCordicConfig.InSize           = CORDIC_INSIZE_32BITS;
225   sCordicConfig.OutSize          = CORDIC_OUTSIZE_32BITS;
```

CORDIC configuration

- Step = 0
  - Calculation made with CORDIC
  - Simple call to HAL_CORDIC_Calculate_DMA

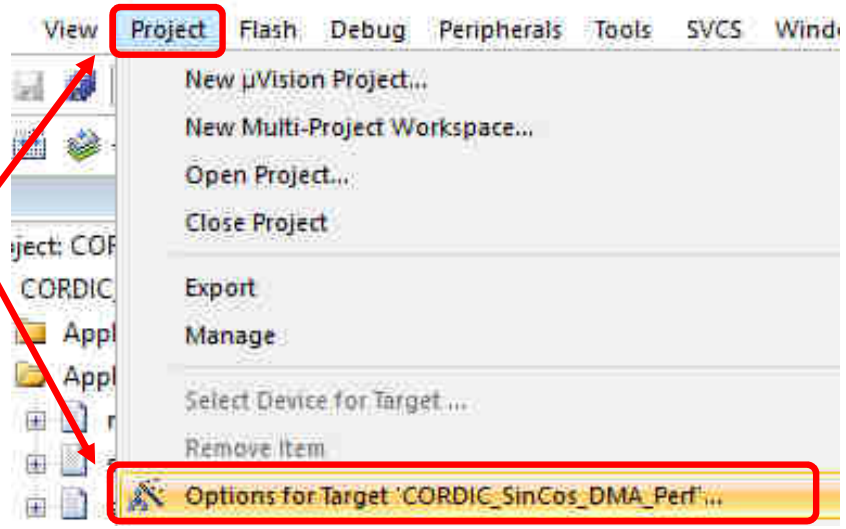
```
246       /*##################### Calculation using CORDIC ##########################*/
247       case 0:
248         for (uint32_t i = 0; i < LOOP_NB; i++)
249         {
250
251           /* Start calculation of sines in DMA mode */
252           if (HAL_CORDIC_Calculate_DMA(&hcordic, aAnglesCordic, aResults,
253                                        ARRAY_SIZE, CORDIC_DMA_DIR_IN_OUT) != HAL_OK)
254           {
255             /* Processing Error */
256             Error_Handler();
257           }
258
```

- Step = 1
  - Calculation made with ARM Math library
  - Lines 277 through 283 are called to perform same CORDIC operation

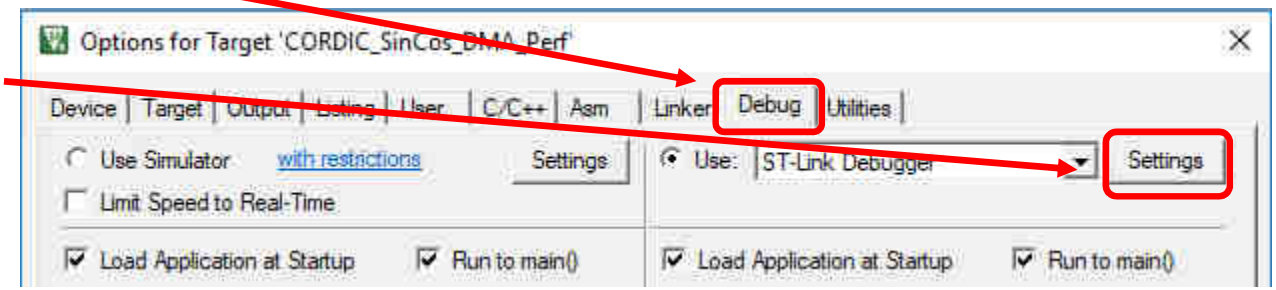
```
272       /*##################### Calculation using arm math library ##############*/
273       case 1:
274         for (uint32_t i = 0; i < LOOP_NB; i++)
275         {
276
277           for (uint32_t j = 0; j < ARRAY_SIZE; j++)
278           {
279             /* Calculate sine */
280             aResults[2*j] = arm_sin_q31(aAnglesLib[j]);
281
282             /* Calculate cosine */
283             aResults[(2*j) + 1] = arm_cos_q31(aAnglesLib[j]);
284           }
285
```

Enable Debug Trace

- Open “Options for Target ‘CORDIC_SinCos_DMA_Perf’ …” window.

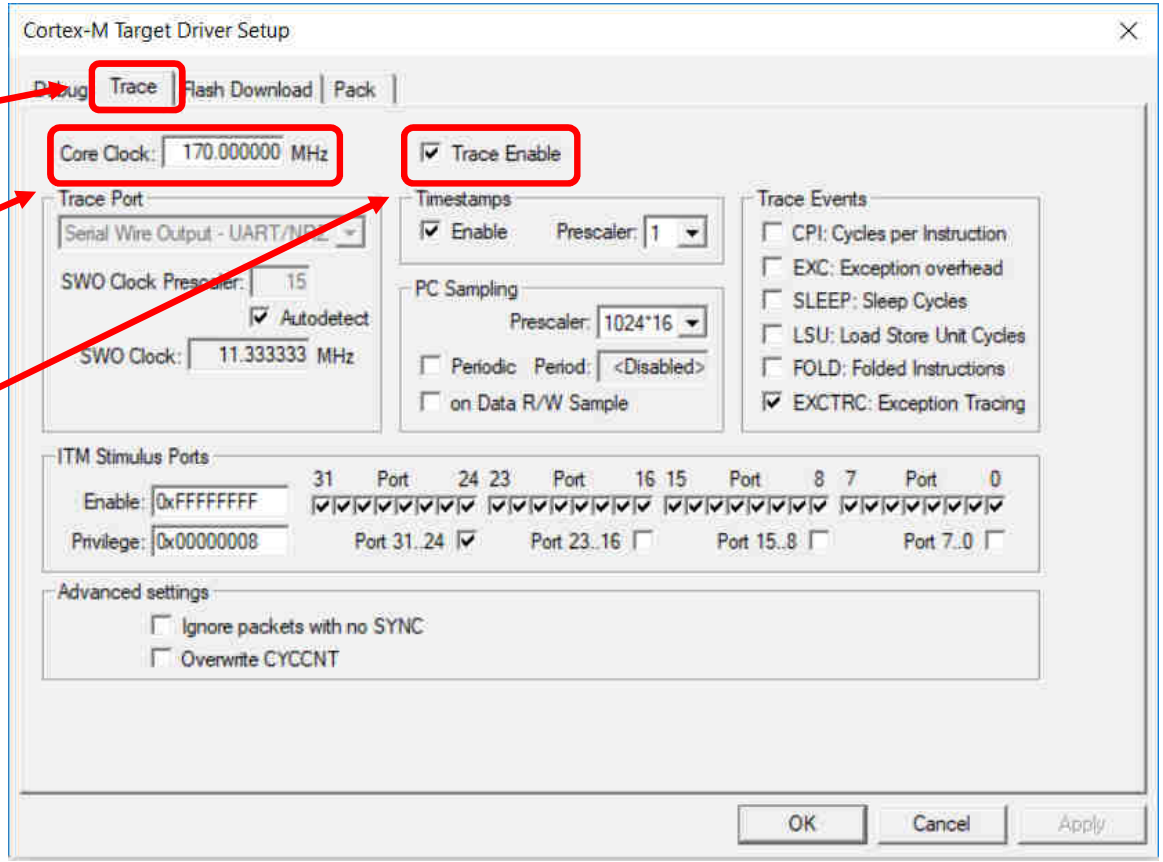


- Switch to “Debug” tab
- Click “Settings” button



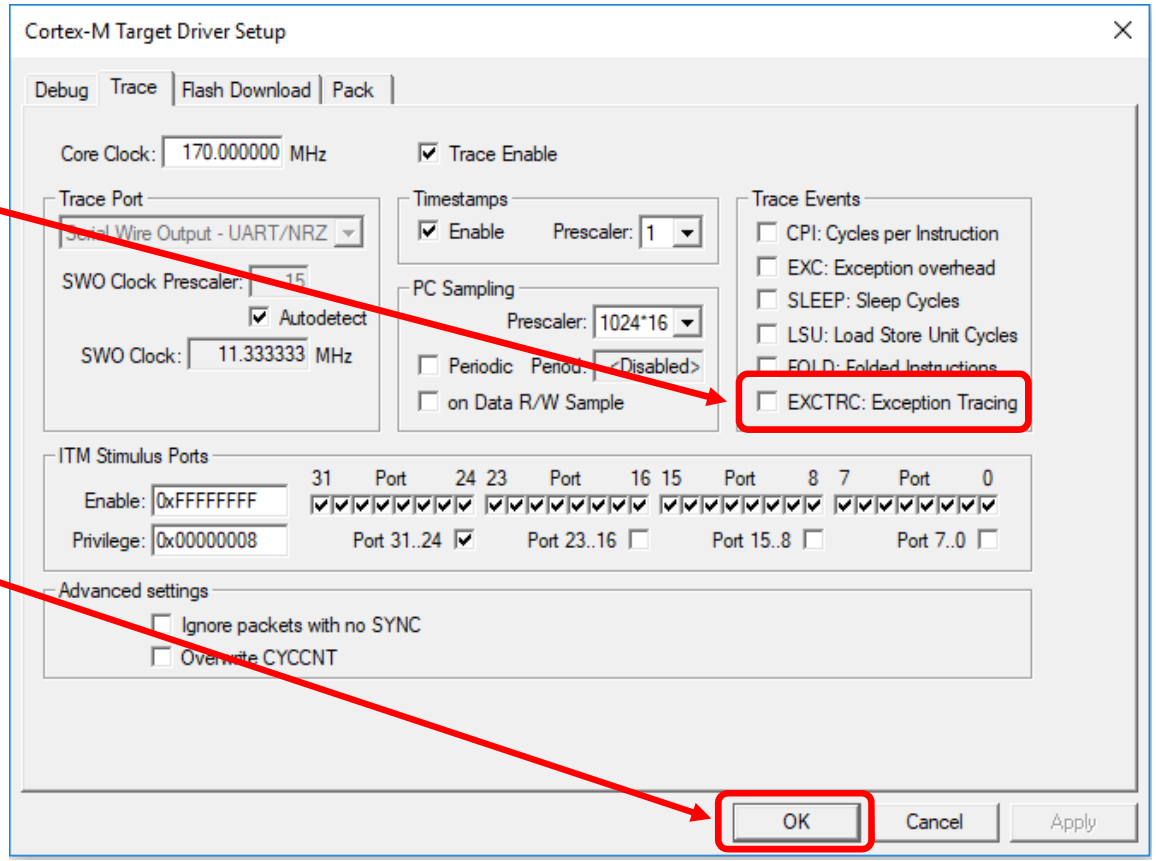
Enable Debug Trace

- Switch to “Trace” tab
- Set “Core Clock” = 170 MHz
- Click “Trace Enable” check box



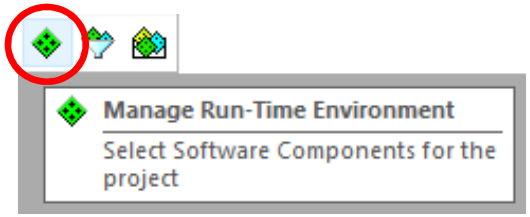
Enable Debug Trace

- Disable “Exeception Tracing”
  - Uncheck Box
- Click “OK” to close setup window

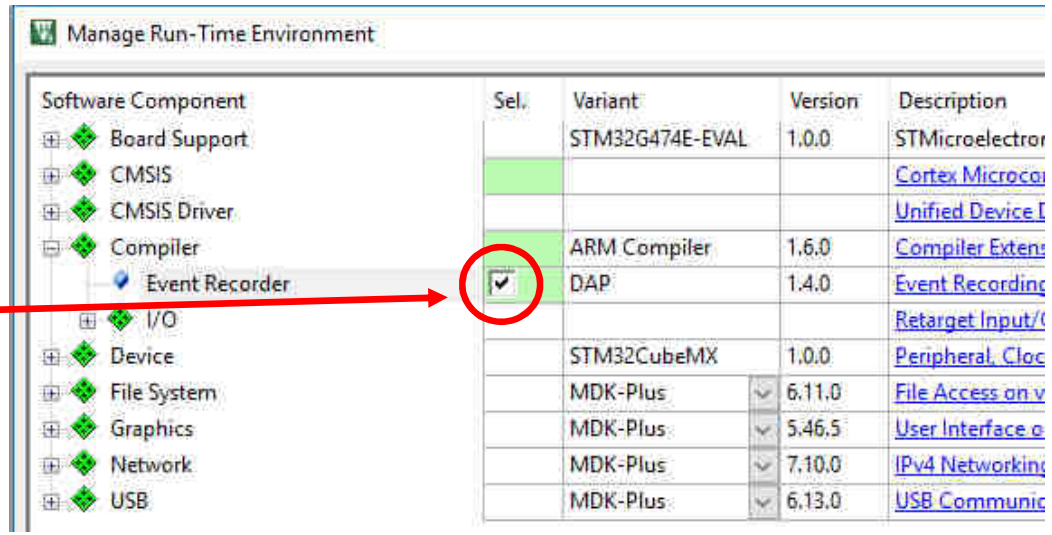


Enable Event Recorder

- Click on Manage Run-Time Environment



- Expand “Compiler”
- Enable “Event Recorder”
  - Check Box
- Click “OK”



Enable Event Recorder

- Open “main.c” under Application/User folder

- Line 27: Add (uncomment)

#include “EventRecorder.h

```
24  /* Private includes ----------------------
25  /* USER CODE BEGIN Includes */
26  #include <arm_math.h>
27  #include "EventRecorder.h"
28  /* USER CODE END Includes */
```

- Line 191 & 192: Add (uncomment)

EventRecorderInitialize(EventRecordAll, 1);
EventRecorderStart();

```
178  int main(void)
179  {
180    /* USER CODE BEGIN 1 */
181    /* STM32G4xx HAL library initialization:
182         - Configure the Flash prefetch
183         - Systick timer is configured by default as source of time base, but user
184           can eventually implement his proper time base source (a general purpose
185           timer for example or other time source), keeping in mind that Time base
186           duration should be kept 1ms since PPP_TIMEOUT_VALUEs are defined and
187           handled in milliseconds basis.
188         - Set NVIC Group Priority to 4
189         - Low Level Initialization
190       */
191    EventRecorderInitialize(EventRecordAll, 1);
192    EventRecorderStart();
193    /* USER CODE END 1 */
```

Enable Event Recorder

- Add (uncomment)

Line 250: EventRecord2(0,i,1);
Line 258: EventRecord2(0,i,2);

```
246         /*##################### Calculation using CORDIC ########################*/
247         case 0:
248           for (uint32_t i = 0; i < LOOP_NB; i++)
249           {
250             EventRecord2(0,i,1);
251             /* Start calculation of sines in DMA mode */
252             if (HAL_CORDIC_Calculate_DMA(&hcordic, aAnglesCordic, aResults,
253                                          ARRAY_SIZE, CORDIC_DMA_DIR_IN_OUT) != HAL_OK)
254             {
255               /* Processing Error */
256               Error_Handler();
257             }
258             EventRecord2(0,i,2);
259
```

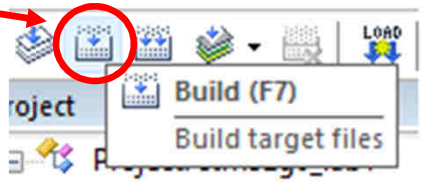
- Add (uncomment)

Line 276: EventRecord2(1,i,1);
Line 285: EventRecord2(1,i,2);

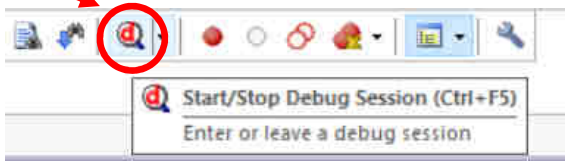
```
272         /*##################### Calculation using arm math library #############*/
273         case 1:
274           for (uint32_t i = 0; i < LOOP_NB; i++)
275           {
276             EventRecord2(1,i,1);
277             for (uint32_t j = 0; j < ARRAY_SIZE; j++)
278             {
279               /* Calculate sine */
280               aResults[2*j] = arm_sin_q31(aAnglesLib[j]);
281
282               /* Calculate cosine */
283               aResults[(2*j) + 1] = arm_cos_q31(aAnglesLib[j]);
284             }
285             EventRecord2(1,i,2);
286           }
```

Hands-On: CORDIC

- Click the “Build”



- “Start Debug” Session

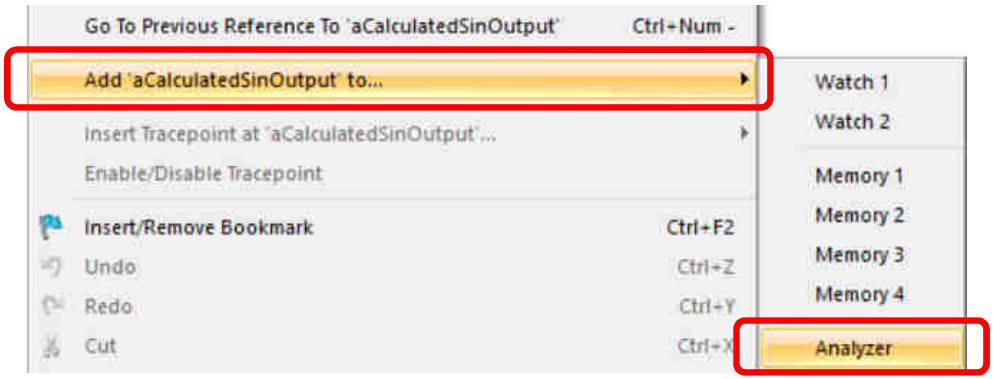


Enable Logic Analyzer

- On line 152, Right mouse click on “aCalculatedSinOutput” variable

```
151  /* Logic Analyzer variables */
152  static int32_t aCalculatedSinOutput = 0;
153  static int32_t aCalculatedCosOutput = 0;
154  int32_t aResults_ptr = 0;
```

- Move mouse over “Add ‘aCalculatedSinOutput’ to…”
- Click on “Analyzer”

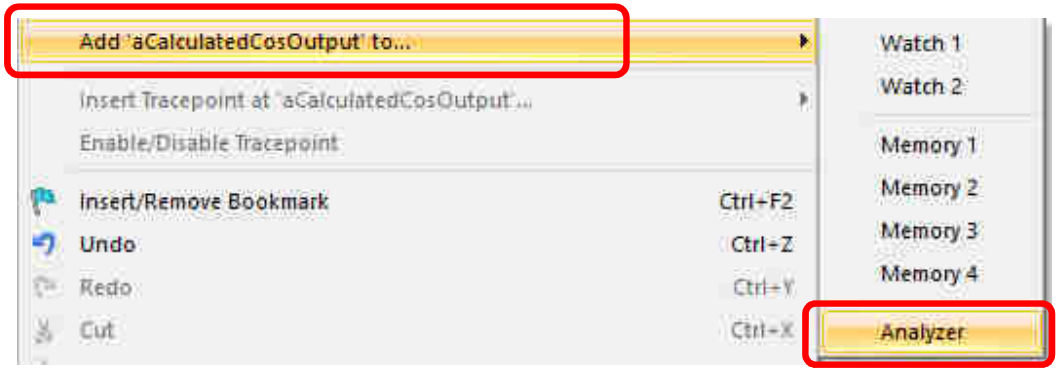


Enable Logic Analyzer

- On line 153, Right mouse click on “aCalculatedCosOutput” variable

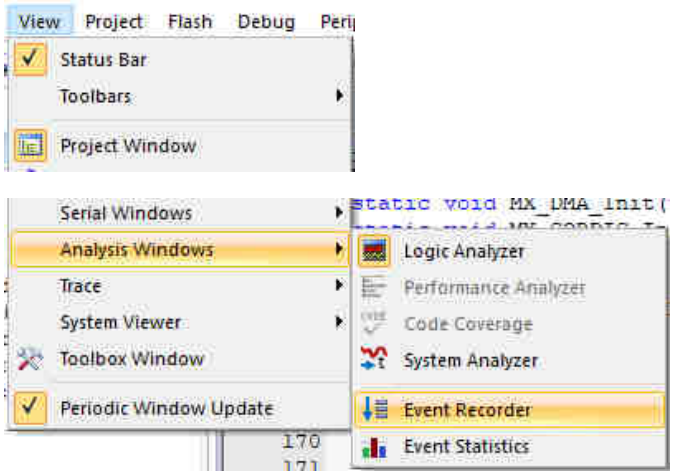
```
151  /* Logic Analyzer variables */
152  static int32_t aCalculatedSinOutput = 0;
153  static int32_t aCalculatedCosOutput = 0;
154  int32_t aResults_ptr = 0;
```

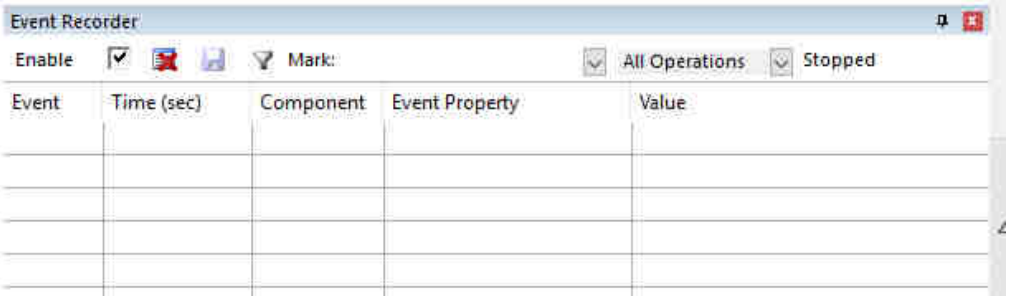
- Move mouse over “Add ‘aCalculatedCosOutput’ to…”
- Click on “Analyzer”



View Event Recorder

- Go to “View” menu
- Select “Analysis Windows”
- Select “Event Recorder”

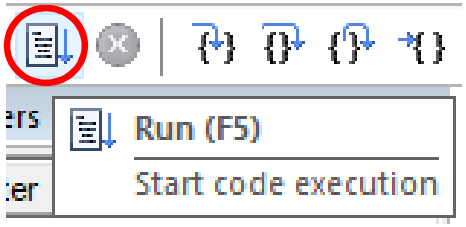




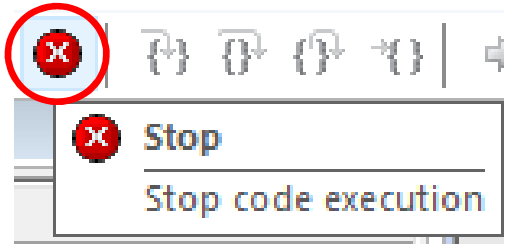
life.augmented

Run the Project

- “Run” the project
  - Capture trace data
  - Press “JOYSTICK_SEL” triggers “CORDIC” or “ARM Math library”
  - Green Led blinking
    - Fast blink (CORDIC used)
    - Slow blink (ARM Math Lib. Used)

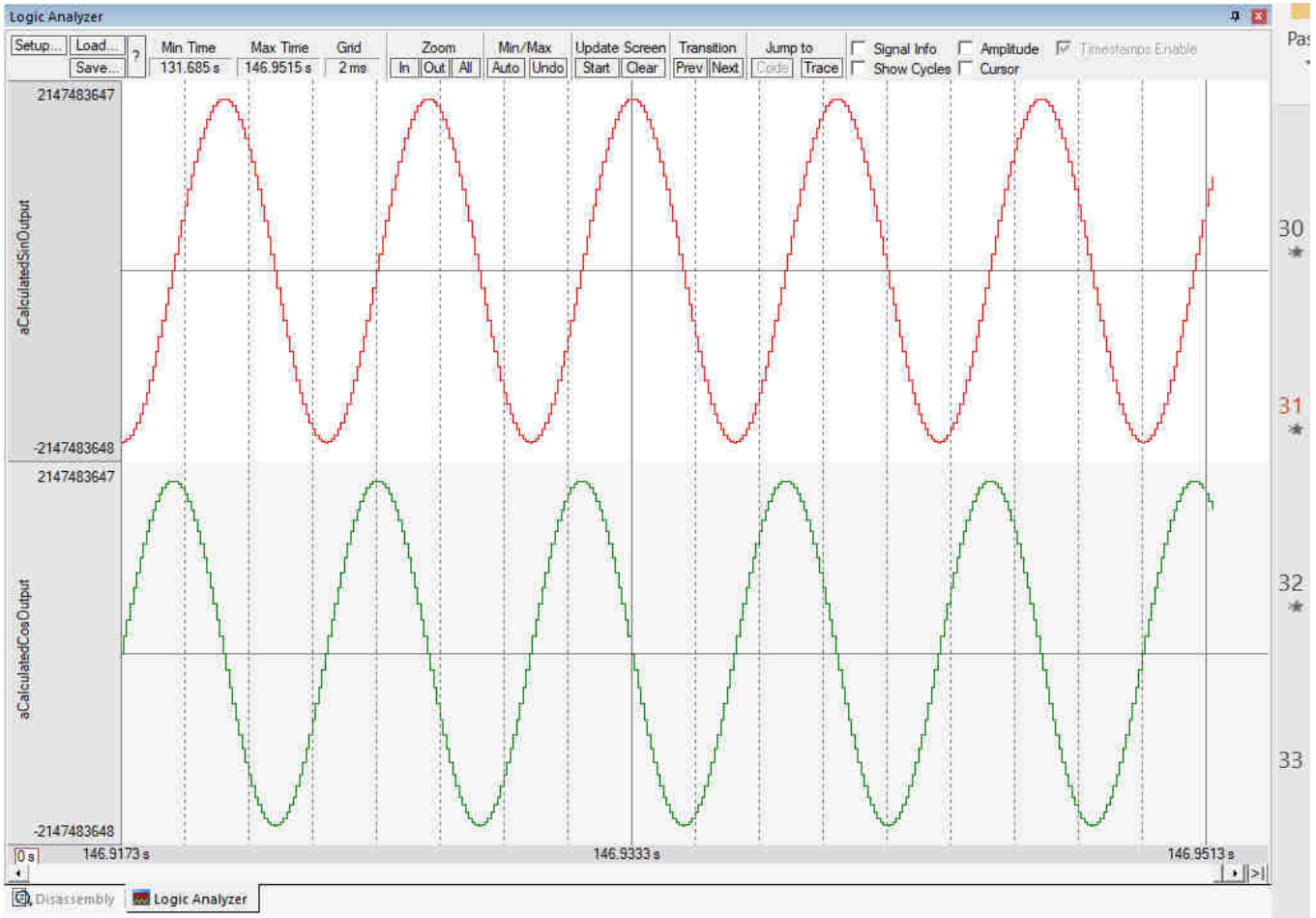


- “Stop” the project execution



Logic Analyzer

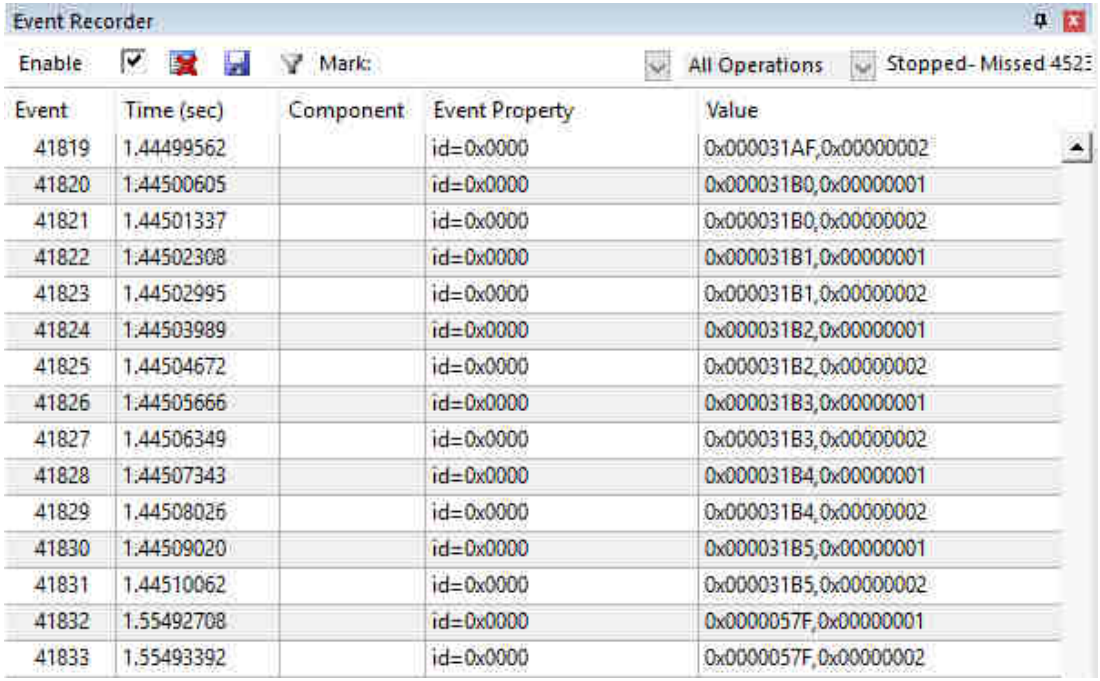
- CORDIC results are shown in Logic Analyzer



life.augmented

Event Recorder

- Events Captured
  - Event Number
  - Event Time Stamp
  - Event Property
    - Id = 0 (CORDIC)
    - Id = 1 (ARM Math Library)
  - Event Values
    - 1st: Loop Number
    - 2nd: Start(0x01) and End(0x02) of array calculation



Event	Time (sec)	Component	Event Property	Value
41819	1.44499562		id=0x0000	0x000031AF,0x00000002
41820	1.44500605		id=0x0000	0x000031B0,0x00000001
41821	1.44501337		id=0x0000	0x000031B0,0x00000002
41822	1.44502308		id=0x0000	0x000031B1,0x00000001
41823	1.44502995		id=0x0000	0x000031B1,0x00000002
41824	1.44503989		id=0x0000	0x000031B2,0x00000001
41825	1.44504672		id=0x0000	0x000031B2,0x00000002
41826	1.44505666		id=0x0000	0x000031B3,0x00000001
41827	1.44506349		id=0x0000	0x000031B3,0x00000002
41828	1.44507343		id=0x0000	0x000031B4,0x00000001
41829	1.44508026		id=0x0000	0x000031B4,0x00000002
41830	1.44509020		id=0x0000	0x000031B5,0x00000001
41831	1.44510062		id=0x0000	0x000031B5,0x00000002
41832	1.55492708		id=0x0000	0x0000057F,0x00000001
41833	1.55493392		id=0x0000	0x0000057F,0x00000002

	Execution Time (measured values)
CORDIC	6.84 µS
ARM Math Library	42.24 µS

- Theoretical Calculation:
  - 64 Array Size
  - 2 functions output: sin and cos
  - 24bit precision: 6cycles
  - Expected time: 4.52us
    =(1/170MHz)*64*2*6

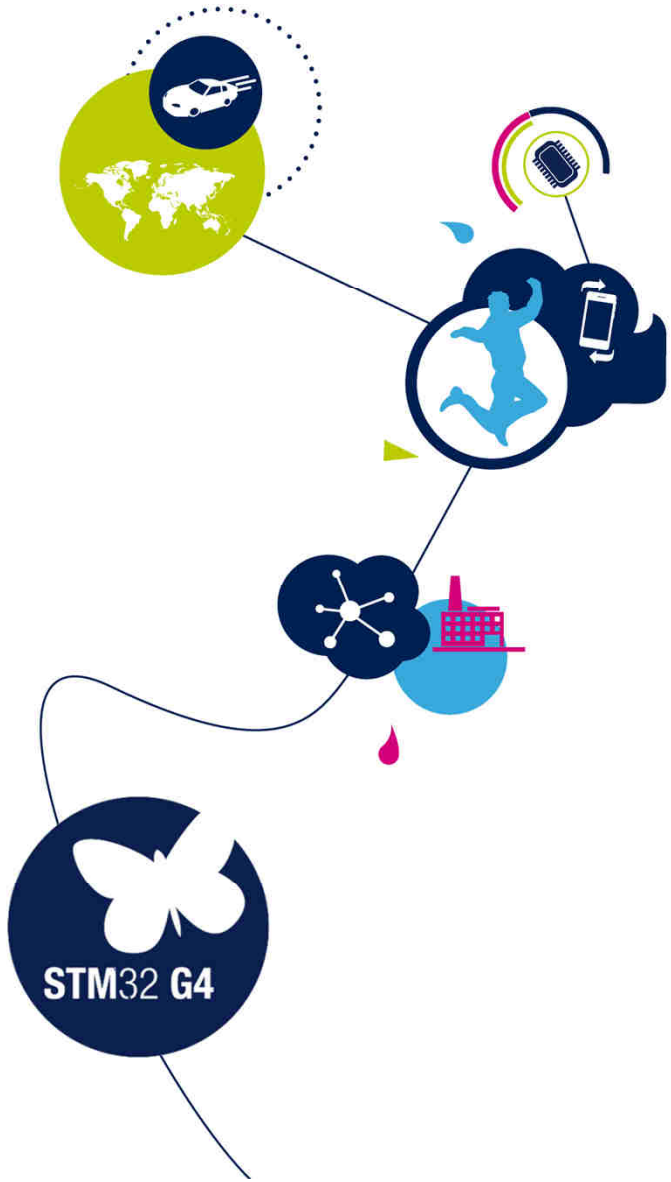
life.augmented

CORDIC: Additional Notes

- Use Cube Low Level driver to reduce HAL overhead
- AN5325: Getting started with the CORDIC accelerator using the STM32CubeG4 MCU Package
- Keil Application Note 321: Event Recorder Debugging with STM32G Cortex-M0/M0+

STM32G4 mixed-signal MCU workshop

Filter Math Acclerator (FMAC) Overview



life.augmented

Filter Math Accelerator

G4 FMAC

- Multiplier (16x16)
- Accumulator (24bit + 2bit)
- Address generation logic
- Input and Output Circular Buffers
- FIR & IIR filters possible
- Dot product, convolution, correlation
- Does not stall the CPU
- 256 x 16bit local memory
- Built-in sequencer
- Runs at the System Clock speed

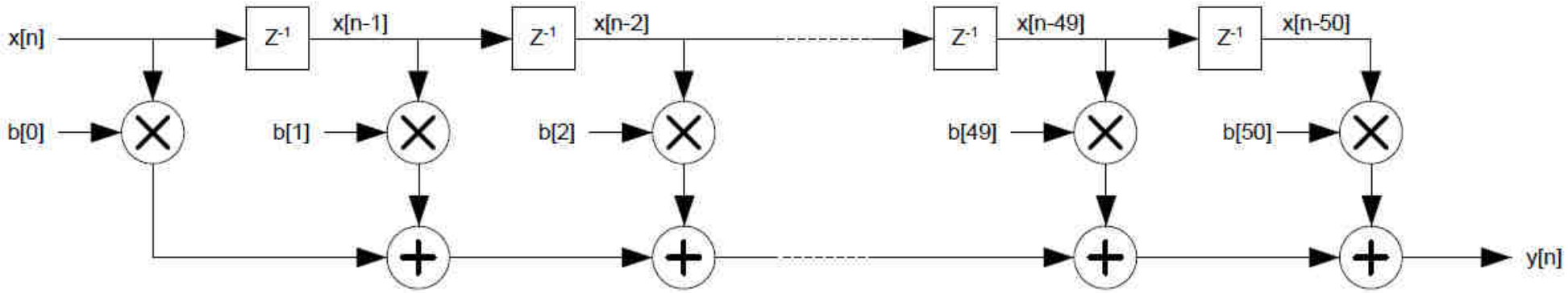
Internally connected to:

- DMA
- AHB slave interface
- NVIC

Filter Math Accelerator

G4 Adaptive FIR Filter

Figure 7. **51-tap FIR filter**



- FIR filter is a convolution of the signal with filter coefficients
- The number of coefficients determines the minimum notch size
- Calculating N coeffcients requires N*N operations

Filter Math Accelerator

G4 Adaptive FIR Filter - Quantities

- Filter Coefficients will probably be in floating point format
- Coefficient conversion to 16 bit fixed point is necessary
- Fixed point format is Q1.15
  - Bits [0..14] represent fractional digits
  - Bit 15 is a sign bit
  - Range = -1.0 (0x8000) to +0.999969482421875 (0x7FFF)
- Proper design will keep the absolute value coeffiecients < or = 1
  - If not, all coefficients will need to be divided by the greatest coefficient
- To convert floating to fixed point, the Floating Point Unit is used
  - FPU takes 8 cycles per conversion
  - 51 coefficients take ~408 clock cycles

Filter Math Accelerator

Lab steps 1 of 6

This Hands On Lab illustrates noise cancellation using an adaptive FIR filter.

The experiment will be run on two input frames:

- a gaussian noise with two tone interferers.

- a gaussian noise with three tone interferers.

Create individual coefficient sets.

The result is a "whitened" noise.

Figure 1. Gaussian noise with two-tone interferer

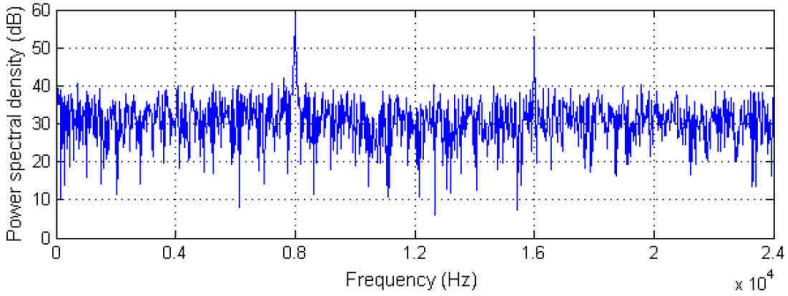
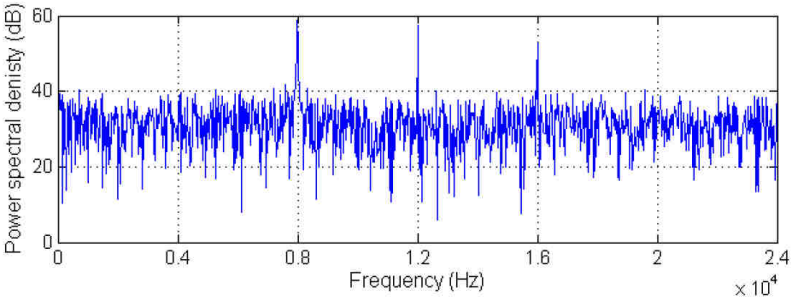


Figure 4. Gaussian noise with three-tone interferer



Filter Math Accelerator

Lab steps 2 of 6

- The FMAC peripheral will be configured to perform an adaptive FIR filter.
- There will be 51 feed-forward taps.
- The gain is equal to 0.
- The input and output thresholds are set to 1.
- The clipping feature can be enabled or disabled thanks to the compilation switch CLIP_ENABLED in main.h.

Filter Math Accelerator

Lab steps 3 of 6

The FMAC peripheral is making the best use of its 256 x 16-bit internal memory.

The local memory is divided into three buffers:

- 0 to 50: coefficient buffer;
- 51 to 150: input buffer;
- 151 to 250: output buffer;
- 251 to 255: unused.

Filter Math Accelerator

Lab steps 4 of 6

- DMA mode is used to transfer input data from memory TO the FMAC peripheral.
- DMA mode is used to transfer output data FROM FMAC peripheral to memory.
- The coefficient buffer contains the vector of feed-forward taps
- This is named the B buffer
  - (size comprised in [2:127]; here, size N = 51 elements).
- This buffer is initialized during the configuration step in DMA mode.

Filter Math Accelerator

Lab steps 5 of 6

The size of the input buffer is the sum of:
- the size of the coefficient vector B (N = 51 elements);
- the size of the additional space D1 needed for throughput improvement (minimum: 0; here, D1 = 49 elements).

The input buffer is filled during the preload step in DMA mode;
new values are added in DMA mode once the preloaded ones have been consumed.

The size of the output buffer is the one of the additional space D2 needed for a better throughput (minimum: 1; here, D2 = 100 elements).
The output buffer is read in DMA mode.

Filter Math Accelerator

Lab steps 6 of 6

STM32 board LEDs are used to monitor the example status:

- LED4 is ON when correct FMAC FIR results have been calculated.

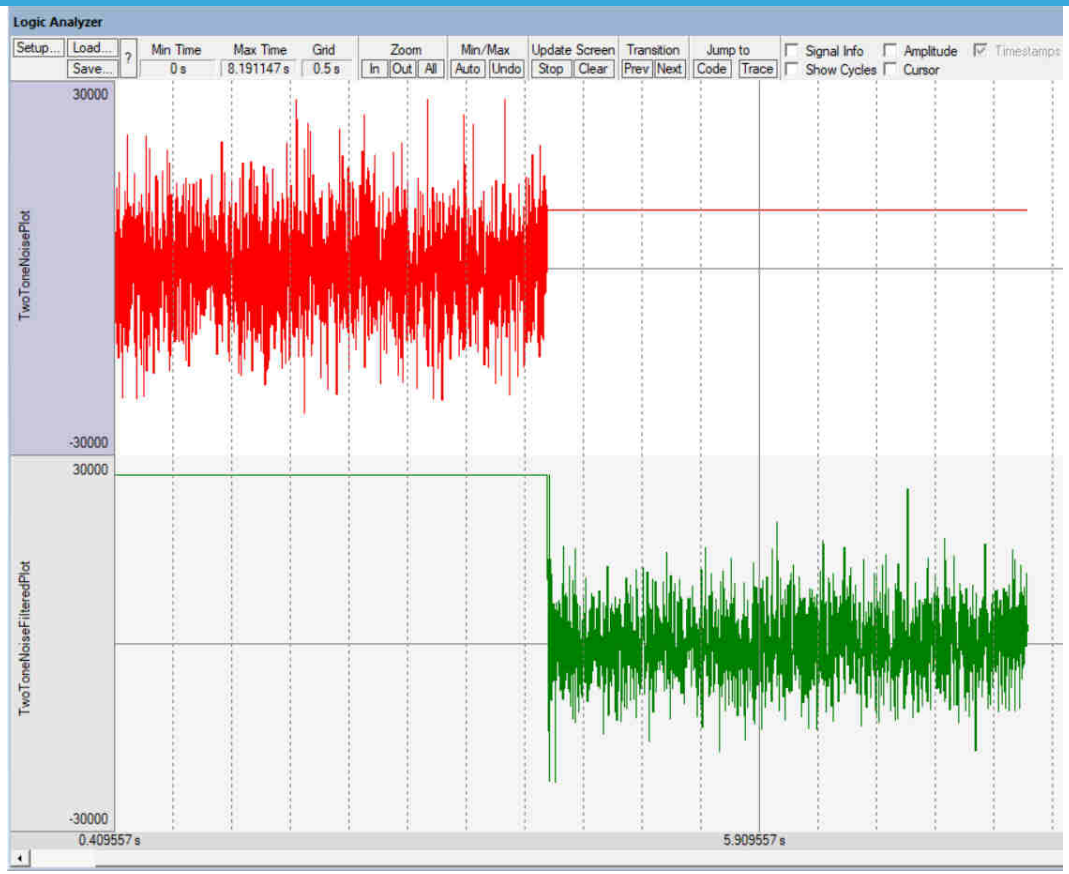
- LED5 is blinking (1 second period) if an error is detected
  in the FIR filter results or if there is an initialization or configuration error.

The filtered output is displayed in the Terminal I/O if the compilation switch PRINT_OUTPUT is enabled in main.h.

Filter Math Accelerator

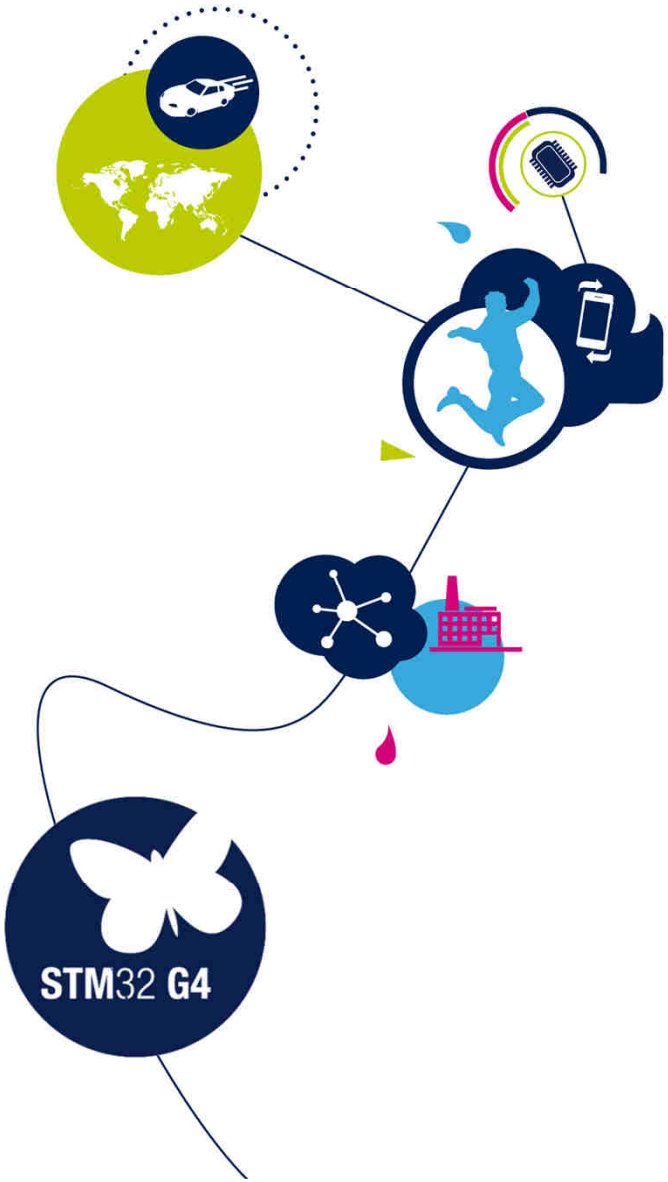
How to view it

- Red shows 2 tone noise
- Green the filtered signal



STM32G4 mixed-signal MCU workshop

Hands On:
Dual Bank Flash
Live Firmware Upgrade



life.augmented

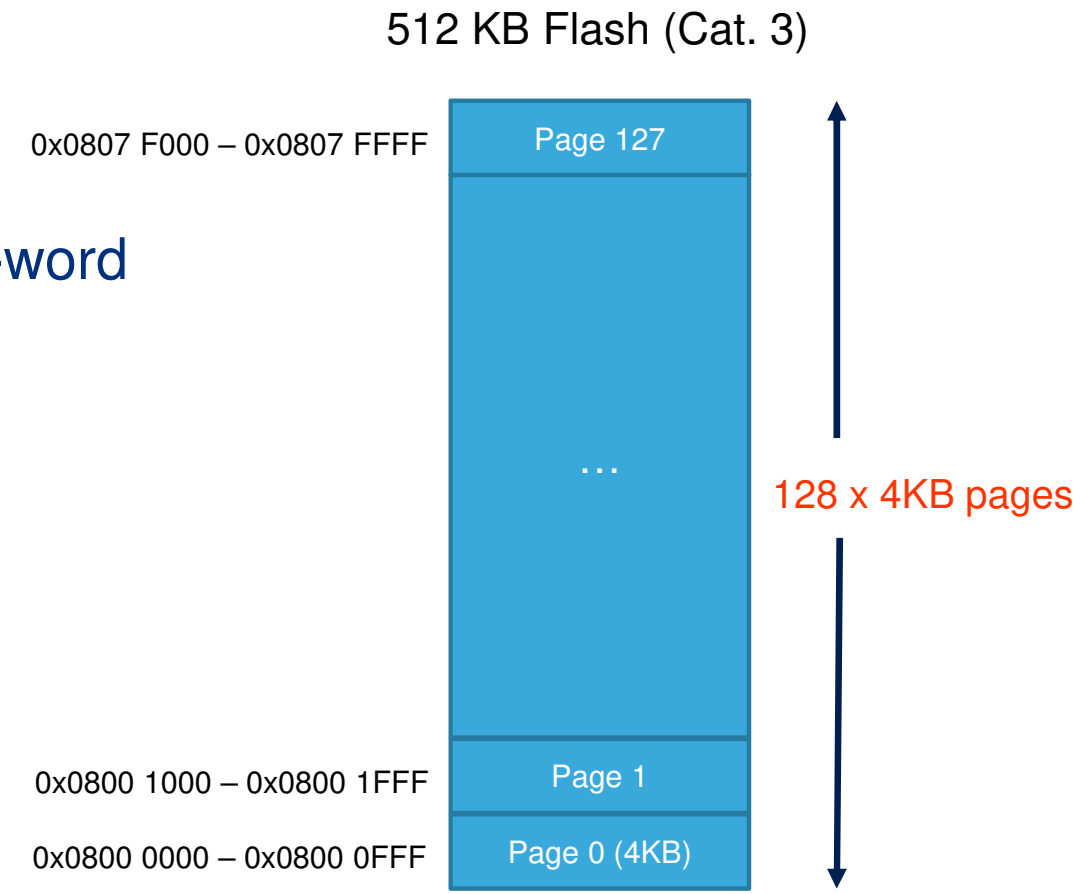
Dual Bank flash

Objective:

- To show the application running from one bank and writing to the second bank(Read-While-Write). Demonstrate the live upgrade possibility.
- Using a Terminal (TeraTerm) to send updated application binary file via the Virtual Comm Port.

512 KB die Flash Organization

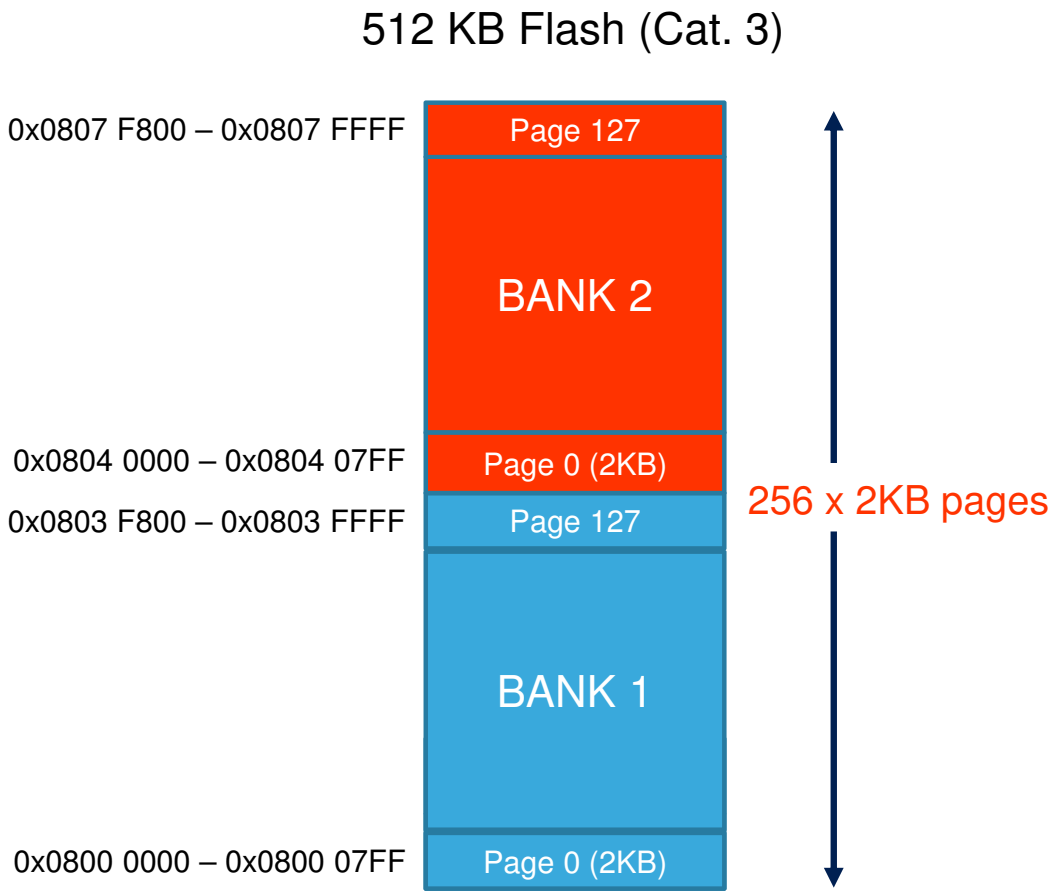
- DBANK = 0 (Single Bank)
  - 128 bits read access
- ECC: 8 bits per 64 bit double-word
- 4 Write protection areas
- Page and Mass erase



life.augmented

512 KB die Flash Organization

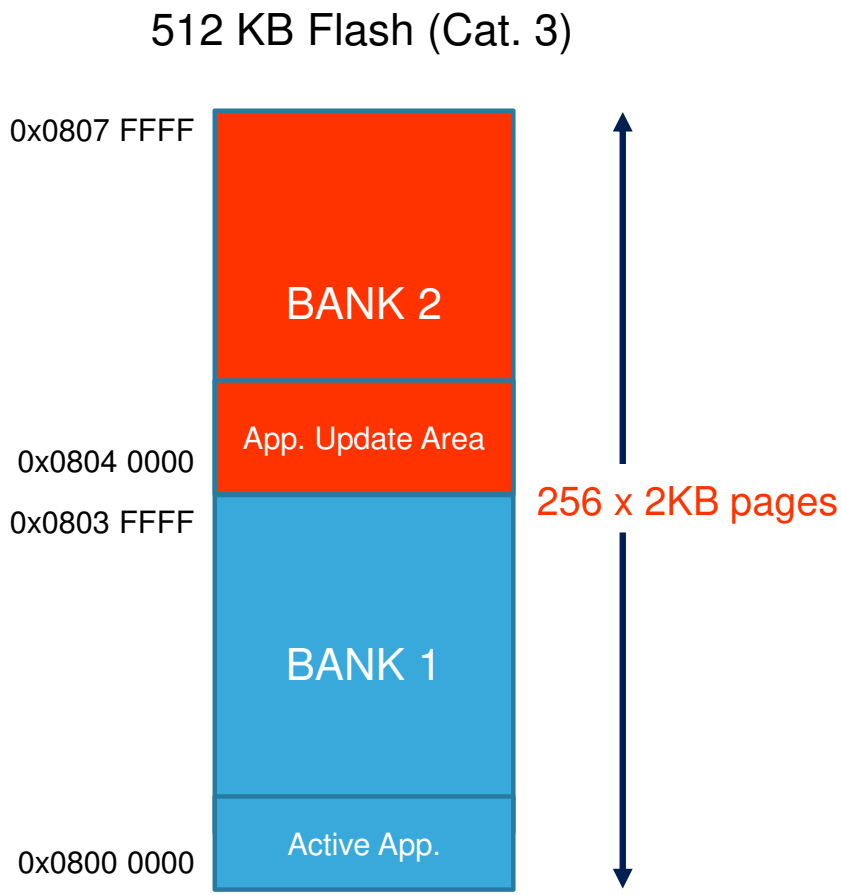
- DBANK = 1 (Dual Bank - Default)
  - 64 bits read access
- Read-While-Write (RWW)
- ECC: 8 bits per 64 bit double-word
- 2 Write protection areas per bank
- Page, Bank and Mass erase



life.augmented

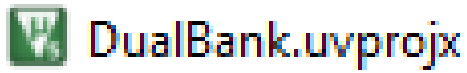
512 KB die Flash Organization

- Active application executing from Bank 1
- Using serial port to receive update
  - Temporary storage in RAM
- Write to BANK 2 area
- Switch application execution to BANK 2

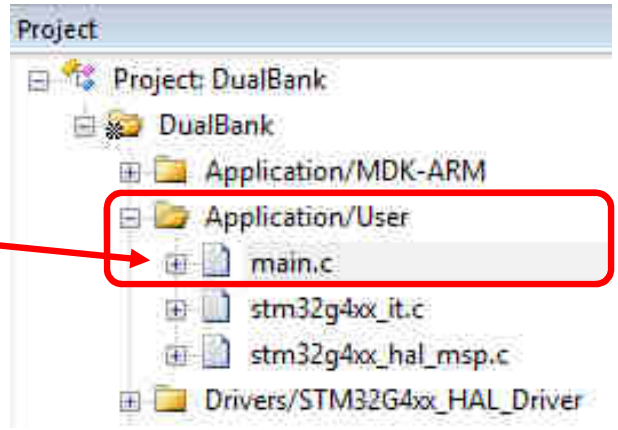


Hands-On: Dual Bank flash

- Open the DualBank project in Keil MDK-ARM folder
  - DualBank\MDK-ARM\DualBank.uvprojx



- Expand “Application/User” folder
- Open “main.c” file
  - Initialize the Clocks, GPIO and Peripherals
  - Main while loop (State Machine)
    - Handles BANK 2 page erase
    - Handles BANK 2 page write
    - Jump and Execute in Bank 2



IRQ Callback functions

- Timer 2 IRQ Callback(Line 481)
  - 10 Hz period
  - LD_ptr: Tracks LED to toggle
- GPIO EXTI Callback (Line 509)
  - Detects rising edge of JOYSTICK_SEL button
- UART Rx complete Callback (Line 524)
  - Signals that the USART Rx DMA has completed

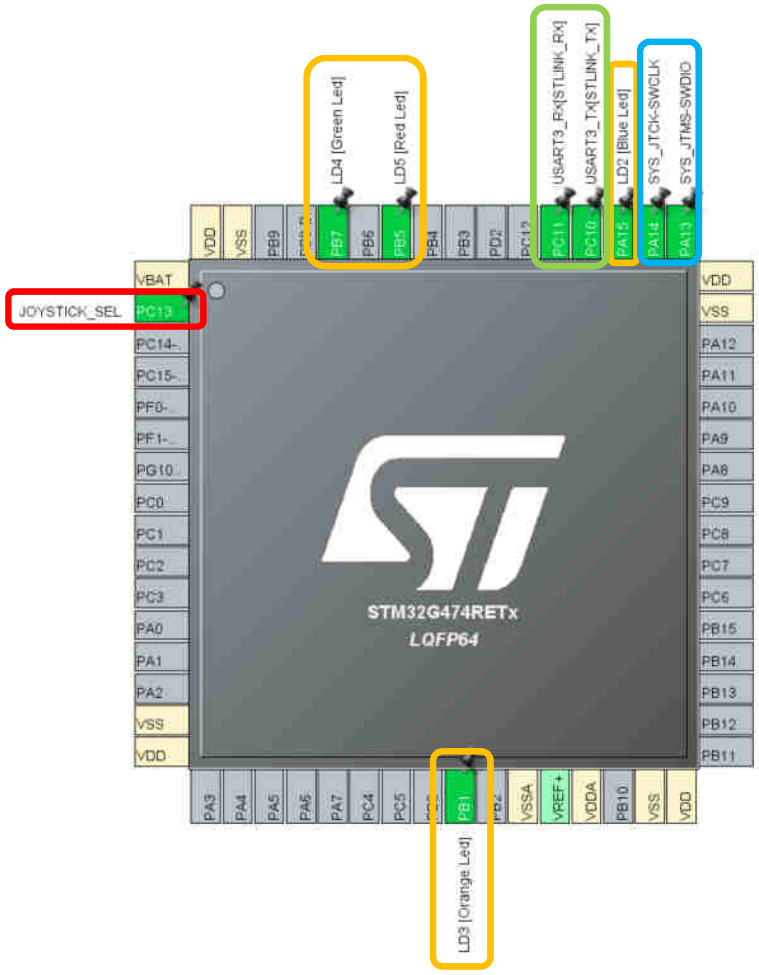
```
481  void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
482  {
483    switch ( LD_ptr )
484    {
```

```
509  void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
510  {
511    if ( ( GPIO_Pin == JOYSTICK_SEL_Pin ) && ( UpdateState == 0 ) )
```

```
524  void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
525  {
526    rx_complete = 1;
527  }
```

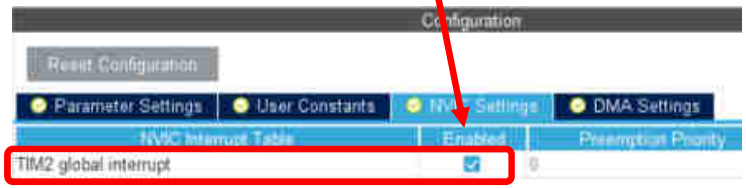
Peripheral configuration

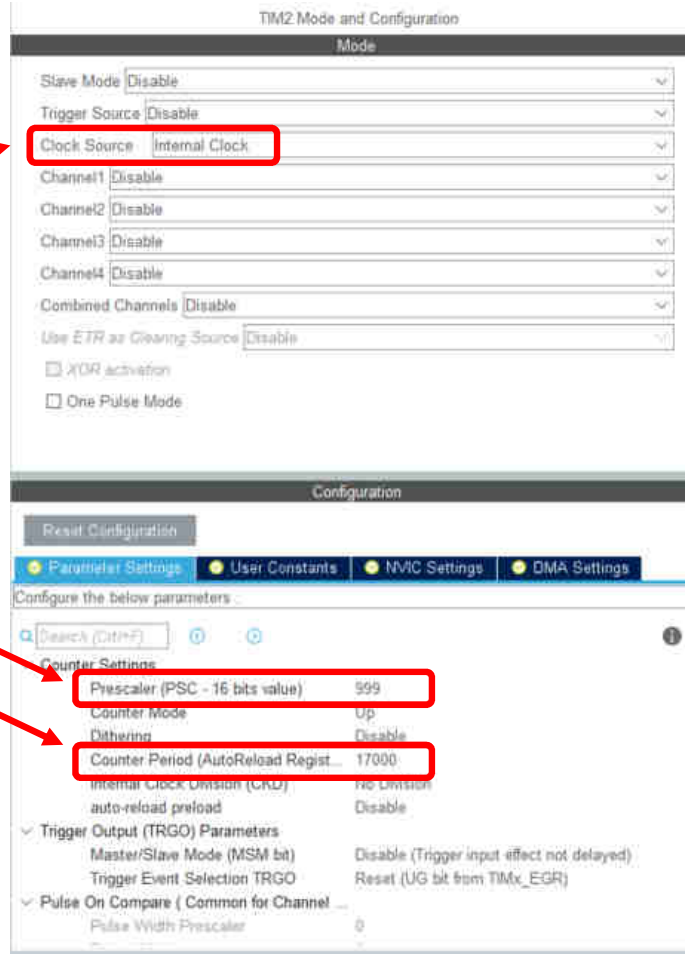
- JOYSTICK_SEL: EXTI13
  - PC13
- LD2-5: GPIO_Output
  - PA15, PB1, PB5 and PB7
- ST-LINK VCP: USART3
  - PC10 and PC13
- Serial Wire Debug with Trace (SWO):
  - PA13, PA14 and PB3



Peripheral configuration

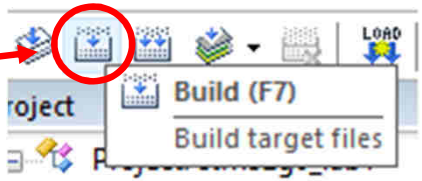
- Timer 2: LED blink rate timer
- Internal Clock Source = 170 MHz
- Prescaler = 999
- Counter Period = 17000
  - 10 Hz timebase interrupt
- TIM2 global interrupt Enabled



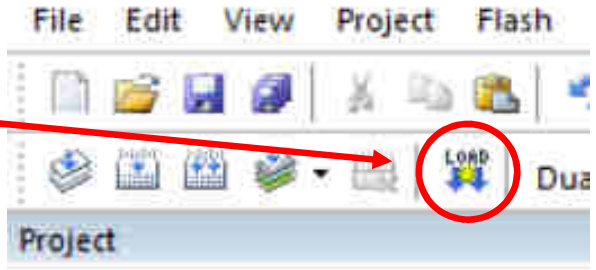


Build the Project

- “Build” project



- “LOAD” project to flash

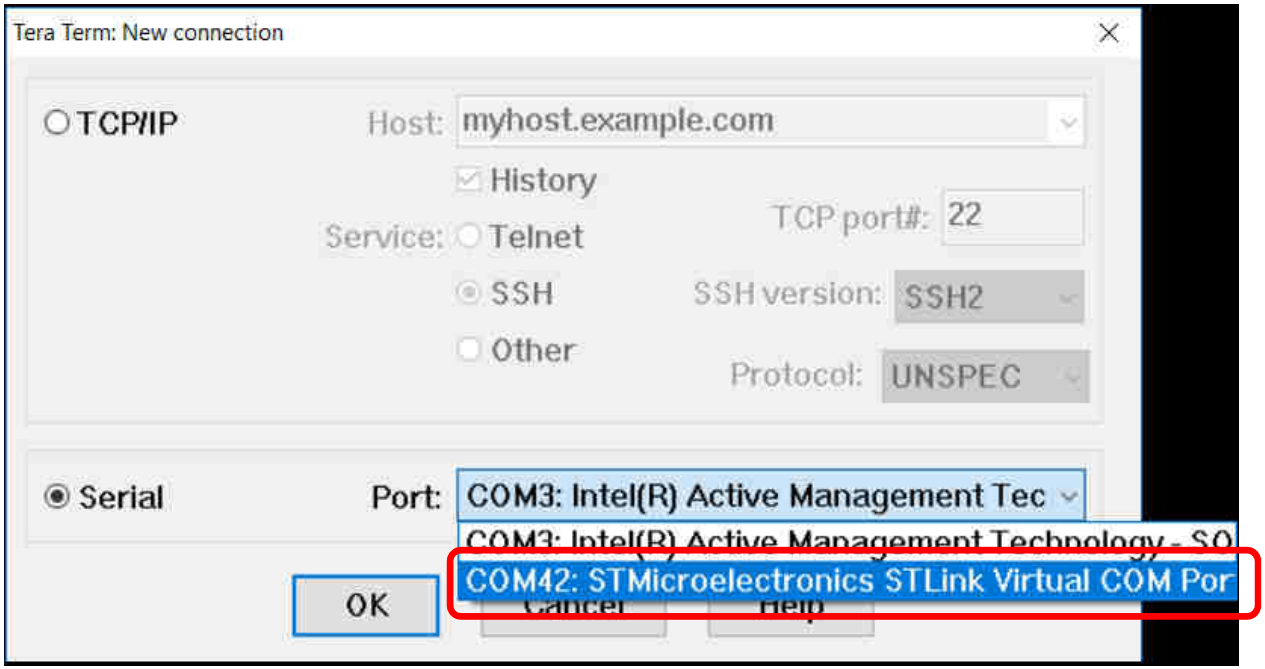


- Press the “RESET” button on the Discovery
  - 4 LEDs blinking in rotating pattern.

Hands On: Dual Bank

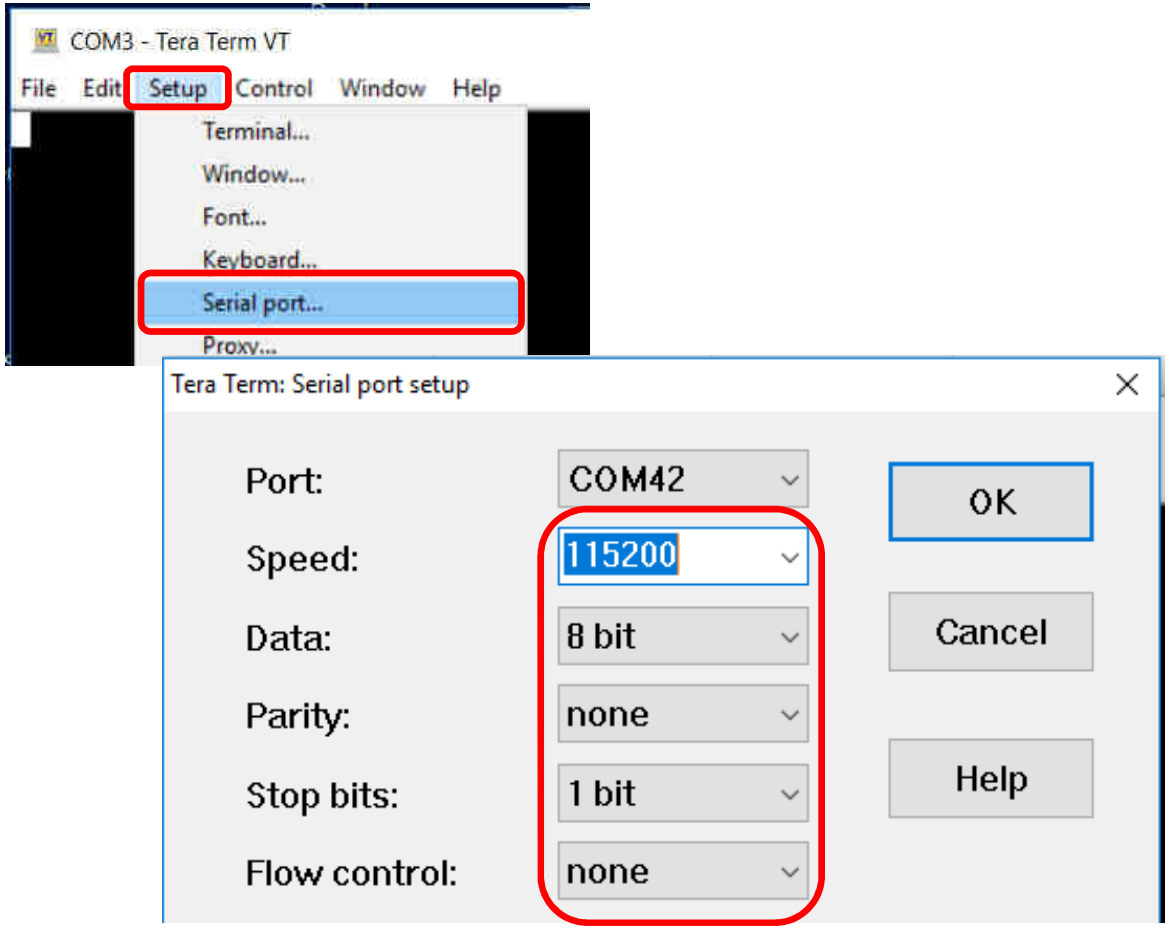
- Launch “TeraTerm”

- Open “New Connection”
  - Click “Serial” button
  - Change Port to “STLink Vitual COM Port”
  - Click “OK”



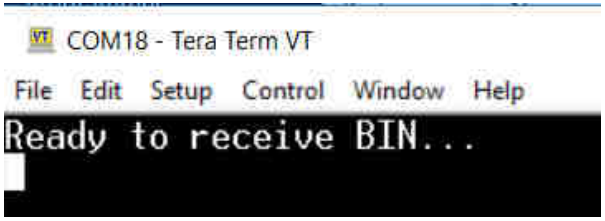
Hands On: Dual Bank

- Verify “Serial port..”
  - Baud = 115200
  - Data = 8 bit
  - Parity = none
  - Stop = 1 bit
  - Flow control = none

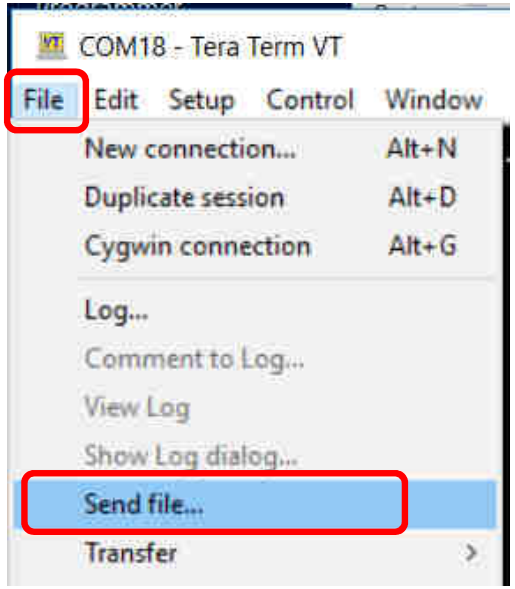


Hands On: Dual Bank

- Press “JOYSTICK_SEL” on Discovery



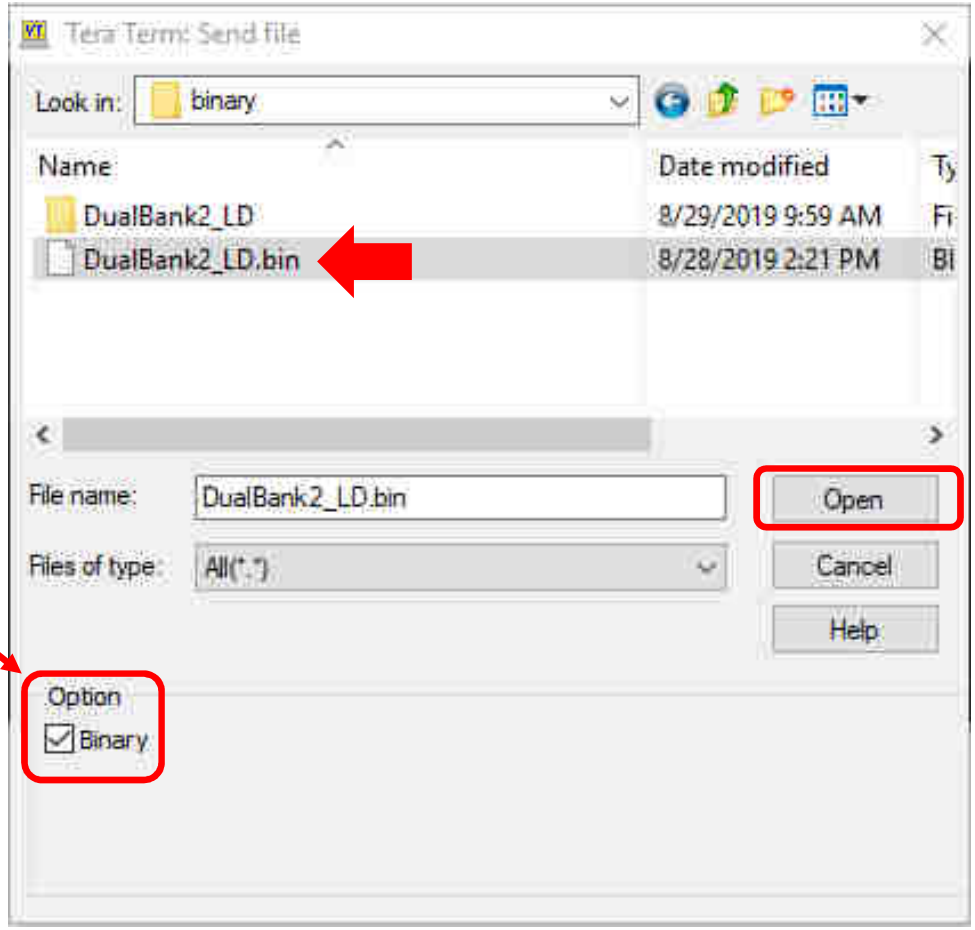
- Send binary file in TeraTerm
  - File > Send file…



life.augmented

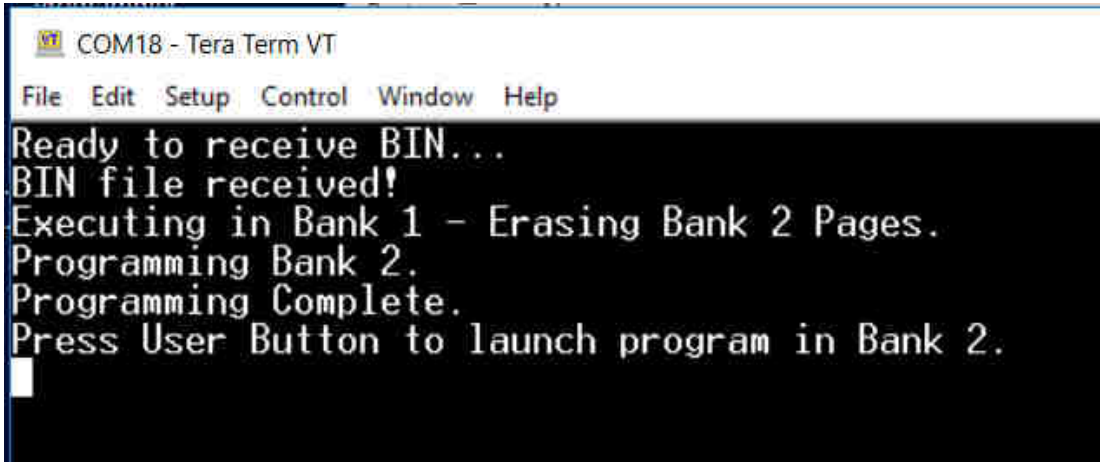
Hands On: Dual Bank

- Select “DualBank2_LD.bin” file
  - DualBank\binary folder
- Check “Binary” Option
- Click “Open”



Hands On: Dual Bank

- TeraTerm will display programming status



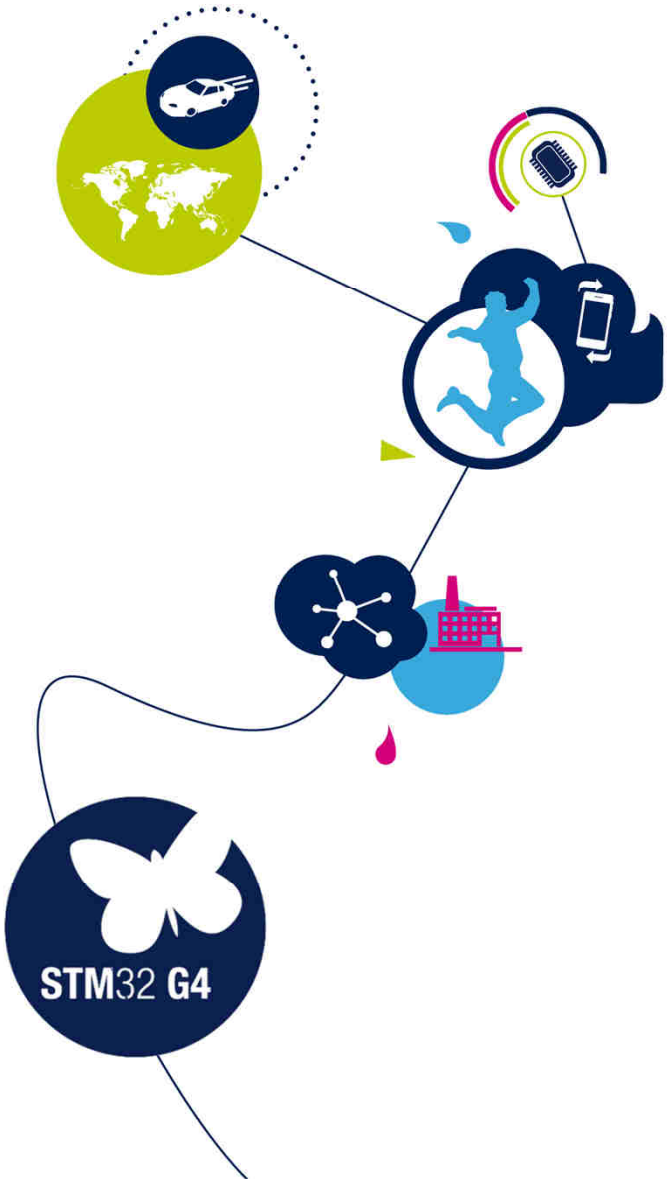
- Press “User Button” to launch program in Bank 2
  - LEDs blink pattern has changed.

Dual Bank: Additional Notes

- Push board “RESET” button to switch execution back to Bank 1
  - To select boot from Bank 2, set the BFB2 bit in the user option bytes
- Reference Manual RM0440: STM32G4 Series advanced ARM-based 32-bit MCUs
  - Section 3: Embedded Flash memory for category 3 devices
- AN2606: STM32 microcontroller system memory boot mode

STM32G4 mixed-signal MCU workshop

Wrap Up



life.augmented

STM32G4 Workshop Related Survey

Use your phone to scan the QR code or type the link into your browser.

https://www.surveymonkey.com/r/PC5DHBM

life.augmented

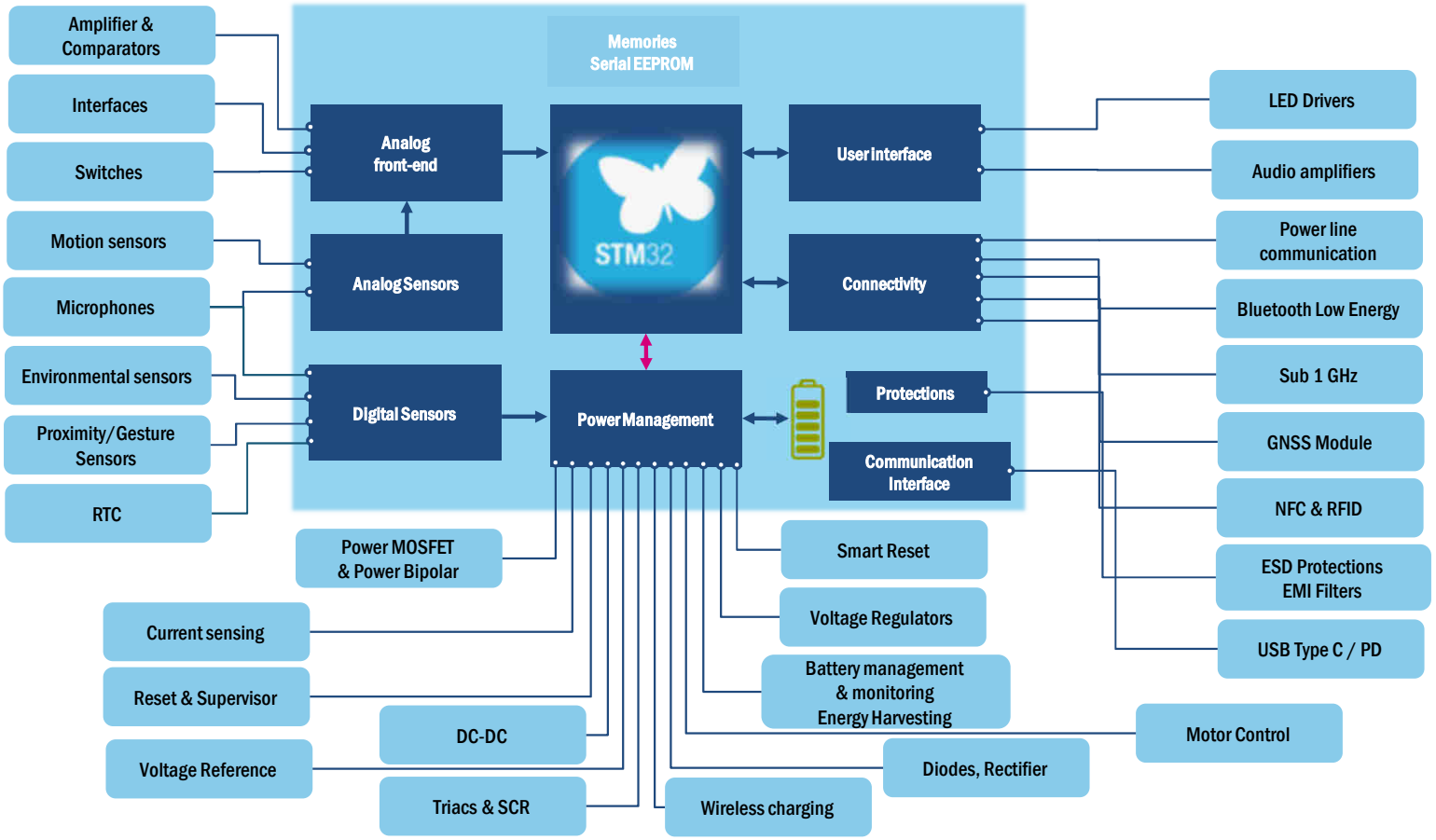
We Greatly Value Your Feedback - Thank you!

STMicroelectronics: one of the broadest Standard Product Portfolios

THINK ST FIRST
ST Standard Products

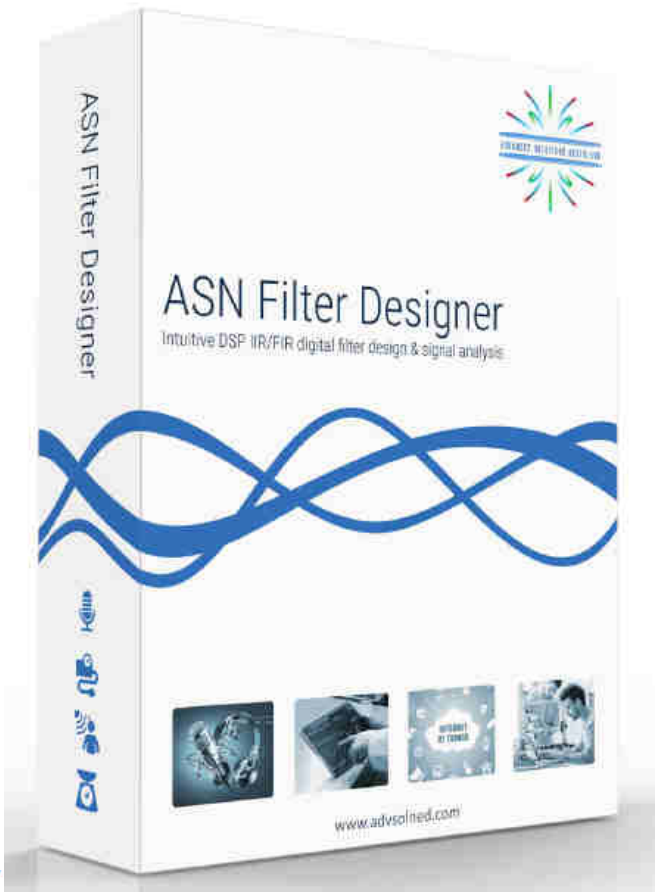
- Protection devices
- Power bipolar transistors
- Power MOSFETs
- Amplifiers and Comparators
- Linear voltage regulators
- EEPROM
- Rectifiers
- SCRs and TRIACs
- Interface ICs
- Voltage references
- Reset and supervisor ICs

The largest offerings of companion products for every STM32 Microcontrollers



life.augmented

ASN Filter Designer



Powerful DSP experimentation platform

Design and fine-tune your digital filter within minutes, and then test it in real-time with a variety of test signals (including your own datasets) via the tool's built-in signal analyser.

Real-time design feedback

No need to explicitly define technical specifications, just sketch your requirements and let the tool automatically fill in the exact specifications for you!

Live math scripting

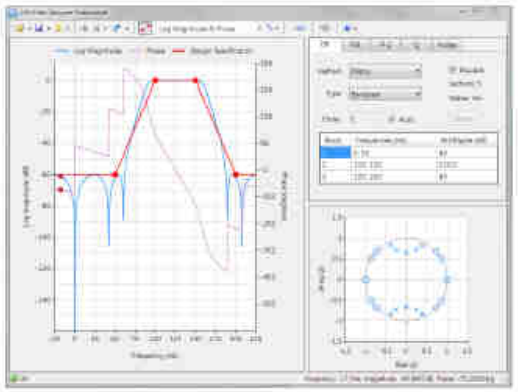
Interactively experiment with specialised transfer functions, pole-zero pairs and other mathematical functions with ASN FilterScript.

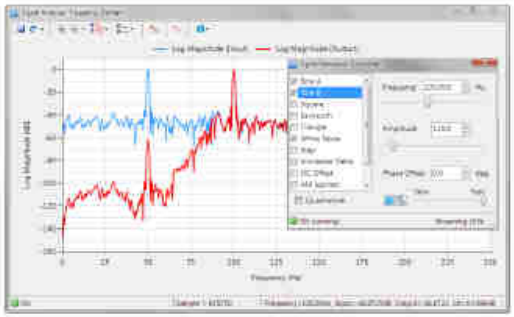
Automatic documentation

Get a detailed specification report of all designed filter blocks for official project documentation, which can be pasted directly into Microsoft Word/Powerpoint.

Deploy to Arm CMSIS-DSP, C/C++, Python, Matlab and more

Easily integrate your designed filter blocks with your other algorithms in Arm CMSIS-DSP, C/C++, Python, Matlab, Octave, Scilab or Labview using provided royalty free software development frameworks.





Download a free demo
http://www.advsolned.com/asn_filter_designer/

Biricha Digital Power Design Tool

Digital Power workshops with BIRICHA

How to implement Digital Power Supplies (PSU) and Power Factor Correction (PFC)

STM32 PSU/PFC design
Step-in Digital Power technology

Biricha Digital Power:
World leading expertise and training in Digital Power

Workshop based on STM32F334 and STM32G474 Nucleo and its dedicated Digital Power expansion board

STMicroelectronics brings industry leading ST MCUs together with Biricha's tools and training

life.augmented

Appendix – Reference Documents

Datasheet and Reference Manual Links

Datasheets

STM32G431xx	STM32G431x6 STM32G431x8 STM32G431xB
STM32G441xB	STM32G441xB
STM32G473xx	STM32G473xB STM32G473xC STM32G473xE
STM32G474xx	STM32G474xB STM32G474xC STM32G474xE
STM32G483xE	STM32G483xE
STM32G484xE	STM32G484xE

Reference Manual

RM0440	STM32G4 Series Reference Manual

Appendix – Reference Documents

Application Note Links

AN2834	How to get the best ADC accuracy
AN4013	STM32 Cross series timer overview
AN4232	Getting started with analog comparators
AN4296	Using CCM SRAM with IAR, Keil, and GNU based tool chains
AN4539	HRTIM Cookbook
AN4899	STM32 GPIO configuration for hardware settings and low power consumption
AN4989	STM32 debug toolbox
AN5305	Digital filter implementation with the FMAC
AN5306	Operational Amplifier usage in STM32G4
AN5310	Guideline for using analog features of the G4 vs. the F3
AN5315	STM32Cube firmware examples for STM32G4
AN5325	Getting started with the CORDIC accelerator
AN5345	High-brightness RGB LED control using the B-G474E-DPOW1 Discovery kit
AN5346	STM32G4 use tips and recommendations

Appendix – Reference Documents

User Manual Links

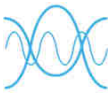
UM1721	Developing applications on STM32Cube with FatFS
UM1722	Developing applications on STM32Cube with RTOS
UM2454	STM32G4 Seris Safety Manual
UM2492	Getting started with STM32CubeG4 for STM32G4 Series
UM2552	USB Power Delivery
UM2570	Description of STM32G4 HAL and low-layer drivers
UM2573	STM32CubeG4 Nucleo demonstraion firmware
UM2577	Discovery kit with STM32G474RE MCU

STM32G4 Series – Take Away

Analog-rich MCUs for mixed-signal applications

Performance

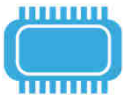
170MHz Cortex-M4 coupled with 3x accelerators



Rich and Advanced Integrated Analog

ADC, DAC, Op-Amp, Comp.

Safety and security focus



Large portfolio available from NOW!

32..512KB Flash memory

32..128-pin packages

life.augmented

Releasing Your Creativity

/STM32 @ST_World community.st.com stm32g4-online-training

life.augmented

www.st.com/STM32G4