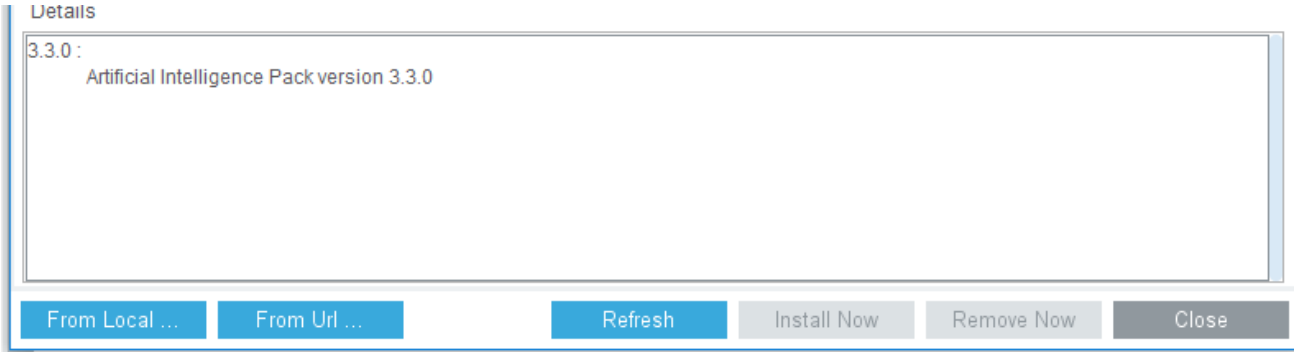
Artificial Intelligence(AI) on STM32

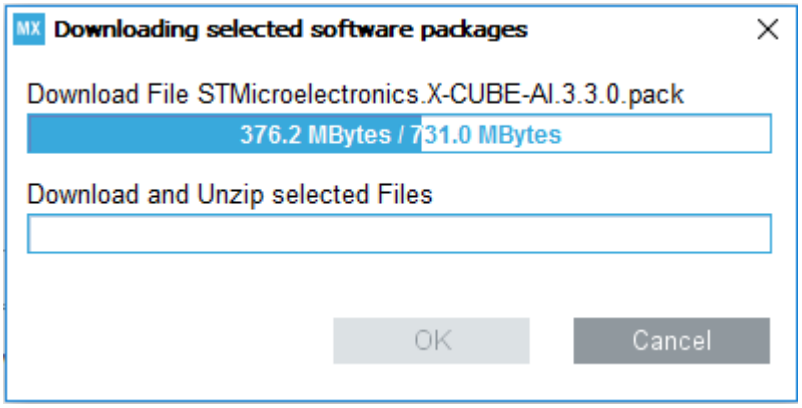
STM32 ARM Cortex-M တွေမှာ AI ပရောဂျက်တွေကိုအလုပ်လုပ်စေဖို့ရန်အတွက် X-CUBE-AI ဆိုတဲ့ expansion package လေးတစ်ခုထွက်လာပါတယ်။ ဒီနေရာမှာအဓိကထားပြီးပြောမှာကတော့ - X-CUBE-AI r3.3.0 နဲ့ Embedded inference client API 1.0.0 တို့ကိုအခြေခံပြီးပြောမှာဖြစ်ပါတယ်။

X-CUBE-AI အား install လုပ်ခြင်း

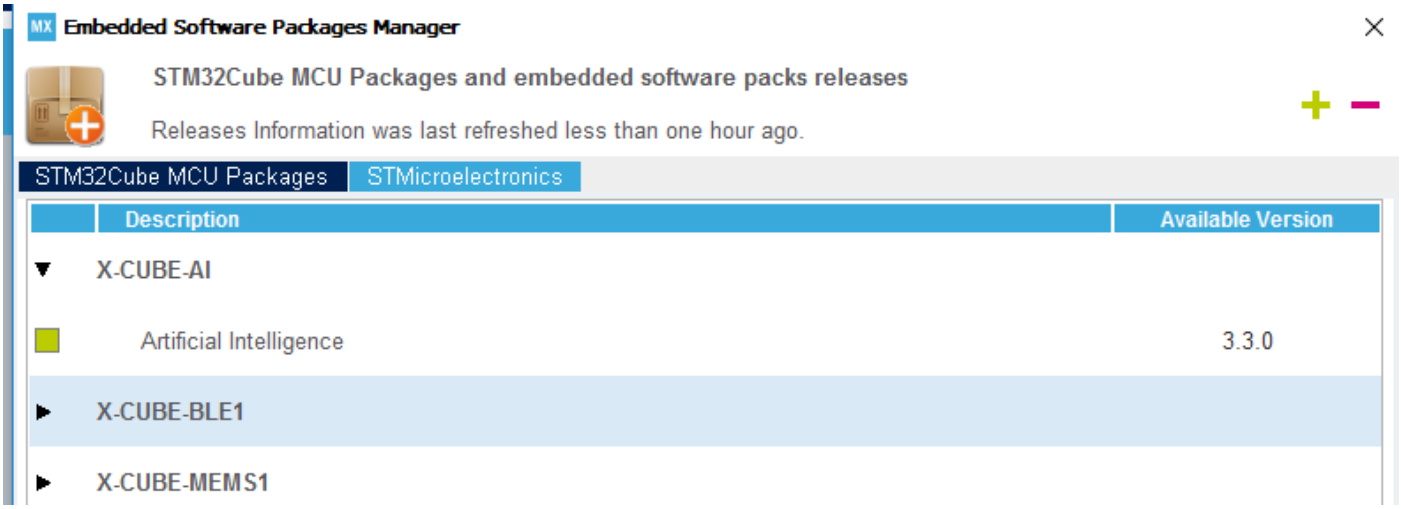
Help > "Manage Embedded Software Packages" ကိုဝင်ပါ။



Updated list ကိုရဖို့ရန်အတွက် "Refresh" ဆိုတဲ့ခလုတ်ကိုနှိပ်လိုက်ပါ။ ပြီးရင် X-CUBE-AI ကိုတင်ဖို့ရန်အတွက် "STMicroelectronics" tab ကိုသွားပါ။ "Refresh" ကိုနှိပ်မှာ အဲဒီ tab ပေါ်မှာဖြစ်ပါတယ်။ ပြီးရင် X-CUBE-AI အောက်မှာနောက်ဆုံးထွက်နေတဲ့ဗားရှင်းကိုရွေးပြီးတော့ "Install Now" ကိုနှိပ်ပါ။

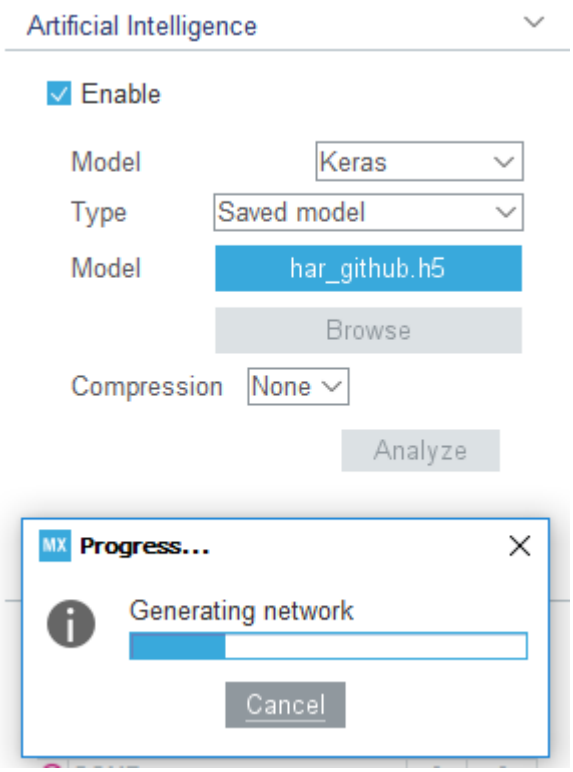


အထက်ပါပုံအတိုင်း ယခုစာအုပ်ရေးနေစဉ်ထွက်တဲ့ X-CUBE-AI.3.3.0.pack ကို download ပြီးတော့ zip ဖိုင်ကိုဖြည်ကာတင်ပေးမှာဖြစ်ပါတယ်။ တင်ပြီးသွားရင်တော့ အောက်ပါပုံအတိုင်းဖြစ်နေပါလိမ့်မယ်။



STM32 AI project အသစ်တစ်ခုတည်ဆောက်ခြင်း

File> New Project ကိုဝင်ပါမယ်။ ကိုယ်သုံးမယ့် MCU သို့မဟုတ် board ကိုရွေးချယ်ရပါမယ်။ ကိုယ်သုံးမယ့် MCU မှာ ပရောဂျက်အရ STM32 Netural Network Library ကိုသိမ်းဆည်ဖို့ရန်အတွက် embedded memory လုံလုံလောက်လောက်ရှိမရှိကိုသိဖို့ရန် MCU Filter ကနေသိနိုင်ပါတယ်။



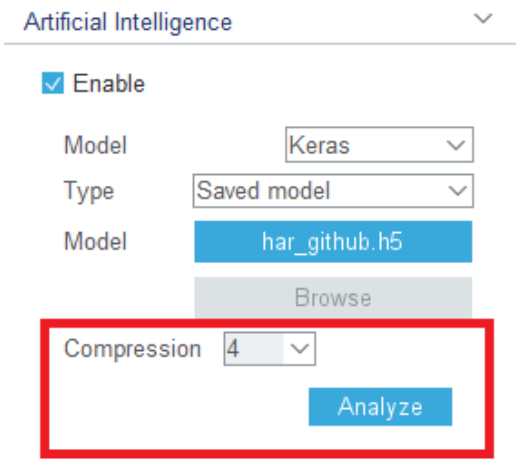
"Artificial Intelligence" မှာ "Enable" လုပ်လိုက်ပါတယ်။ နမူနာပြမယ့် Deep Learning (DL) model ကို အောက်ပါ github link မှာ download ဆွဲပြီးတော့ထည့်လိုက်ပါတယ်။

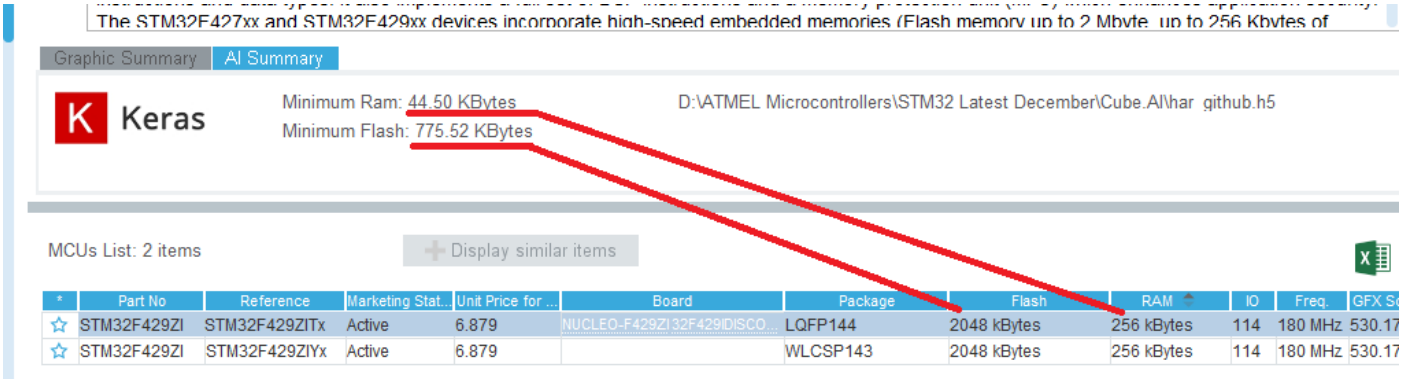
https://github.com/Shahnawax/HAR-CNN-Keras

အထက်ပါ link ကနေ "model.h5" ဆိုတဲ့ deep learning model ကို ဒေါင်းလုပ်ရယူပြီးတော့ "har_github.h5" ဆိုပြီးတော့ နာမည်ပြင်ပေးလိုက်ပါတယ်။ ပြီးရင် Model နေရာမှာ "Keras" ကိုရွေးခဲ့လိုက်ပါ

မယ်။ Type ကတော့ “Saved model” ပေးခဲ့ပါမယ်။ Compression တန်ဖိုး “None” ပေးပြီးတော့ “Analyze” ကို နှိပ်လိုက်တဲ့အခါမှာတော့ - Keras ရဲ့ minimum embedded memory ကိုအောက်ပါအတိုင်းမြင်ရမှာဖြစ်ပါတယ်။

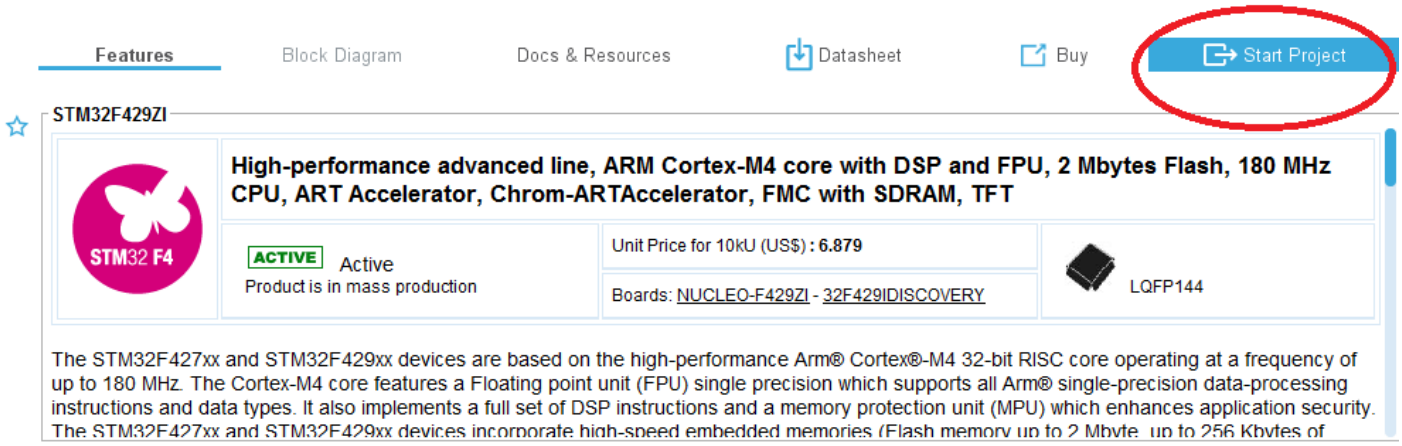
ကျွန်တော်တို့ယခုသုံးမယ့် board က STM32F429I-Discovery ဖြစ်ပါတယ်။

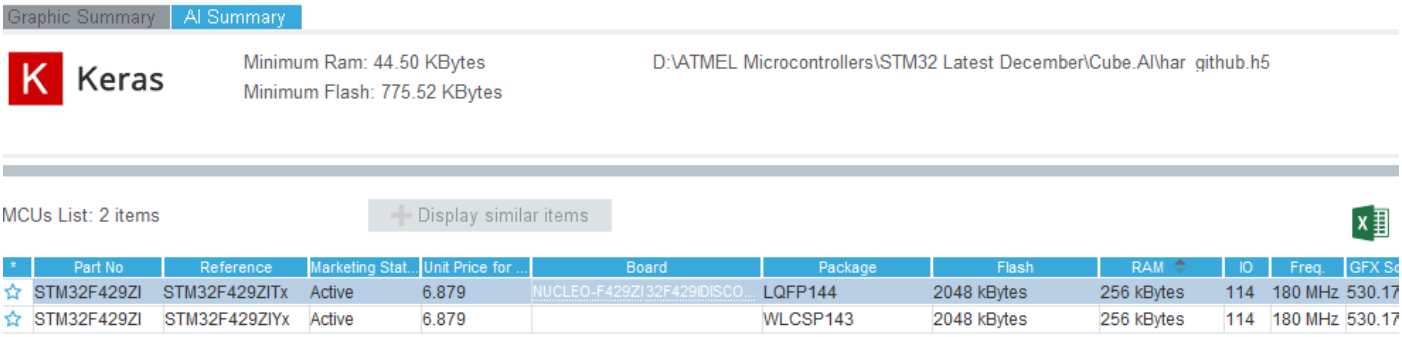




Compression တန်ဖိုး “4” ပေးပြီးတော့ “Analyze” လုပ်လိုက်တဲ့အခါမှာ - ယခုသုံးမယ့် STM32F429 အတွက် လုံလောက်တဲ့ Embedded Memory ရှိတာကိုတွေ့ရမှာဖြစ်ပါတယ်။

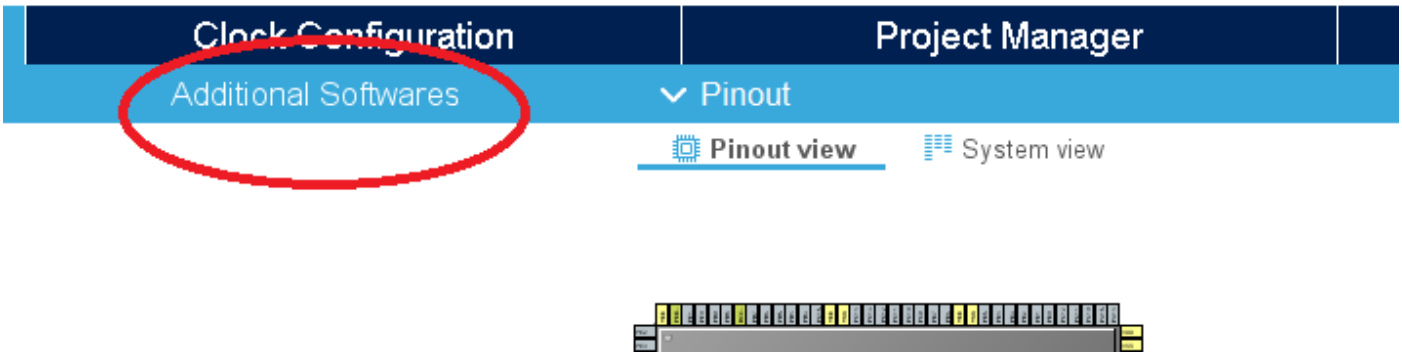
အောက်ပါပုံမှာပြထားတဲ့အတိုင်း “Start Project” ကိုနှိပ်ပြီးတော့ ပရောဂျက်ကိုစတင်လိုက်ပါတယ်။





*	Part No	Reference	Marketing Stat...	Unit Price for ...	Board	Package	Flash	RAM	IO	Freq.	GFX Sc
☆	STM32F429ZI	STM32F429ZITx	Active	6.879	NUCLEO-F429ZI 32F429IDISCO...	LQFP144	2048 kBytes	256 kBytes	114	180 MHz	530.17
☆	STM32F429ZI	STM32F429ZIYx	Active	6.879		WLCSP143	2048 kBytes	256 kBytes	114	180 MHz	530.17

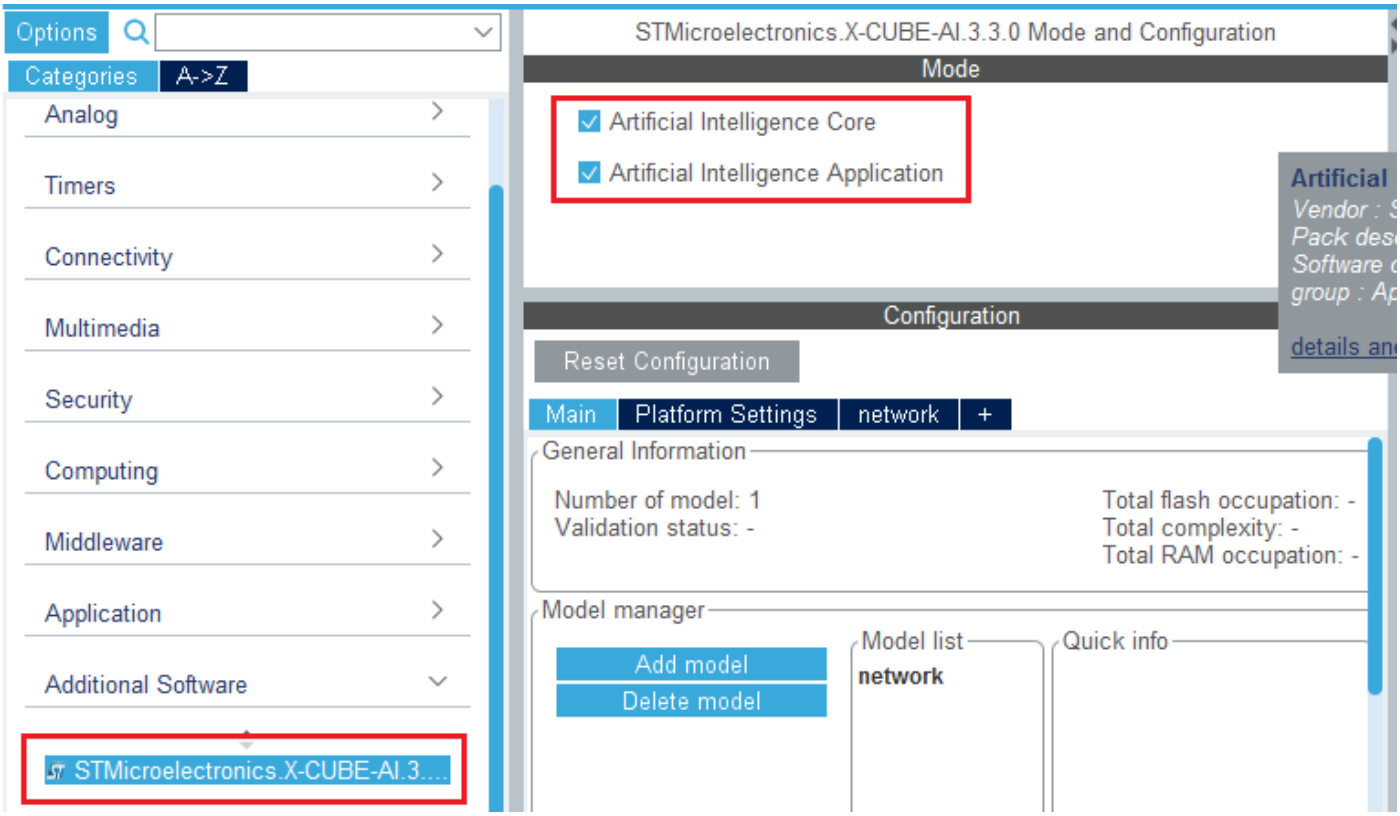
"Additional Softwares" ထဲကိုဝင်ပါမယ်။



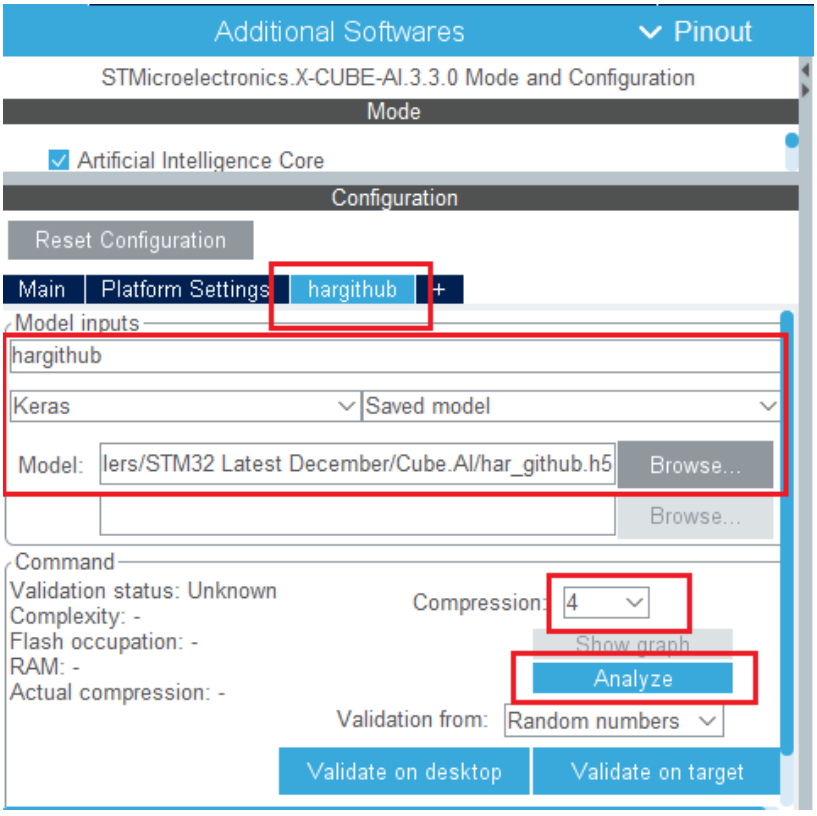
နာမူနာအနေနဲ့ Application မှာ "Validation" ကိုရွေးခဲ့ပြီးတော့ CORE မှာ check လုပ်ခဲ့ပါမယ်။

Vendor	Pack/Bundle	Pack Version	Class	Pack Action	Group/Subgroup	Selection
STMicroelectronics	X-CUBE-AI/Application	3.3.0	Artificial Intelligence	Installed		
STMicroelectroni	X-CUBE-AI/Application	3.3.0	Artificial Intelligence		Application	Validation
STMicroelectronics	X-CUBE-AI/Core	3.3.0	Artificial Intelligence	Installed		
STMicroelectroni	X-CUBE-AI/Core	3.3.0	Artificial Intelligence		CORE/	☑
STMicroelectronics	X-CUBE-BLE1/Application	4.2.0	Wireless	Install*		
STMicroelectronics	X-CUBE-BLE1/BlueNRG-MS	4.2.0	Wireless	Install*		
STMicroelectronics	X-CUBE-MEMS1/Application	5.2.1	Device	Install*		
STMicroelectronics	X-CUBE-MEMS1/MEMS	5.2.1	Board Extension	Install*		
STMicroelectronics	X-CUBE-MEMS1/MEMS	5.2.1	Board Component	Install*		
STMicroelectronics	X-CUBE-MEMS1/STM32Cube_Custom_BSP_Drivers	5.2.1	Board Support	Install*		

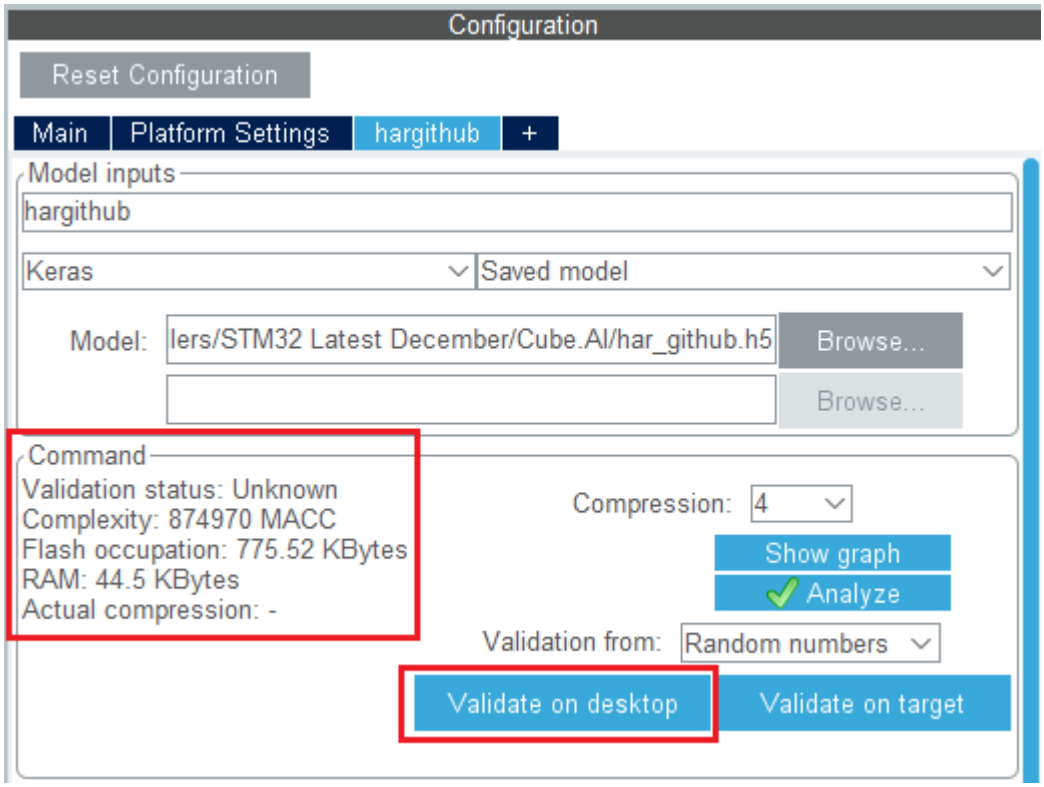
"Additional Software" အောက်က STMicroelectronics X-CUBE-AI x.x.x ကိုဝင်လိုက်ပါ။ အဲဒီမှာ Artificial Intelligence Core နဲ့ Artificial Intelligence Application ကိုအမှန်ခြစ်တပ်ပြီးဖွင့်ထားခဲ့ပါမယ်။



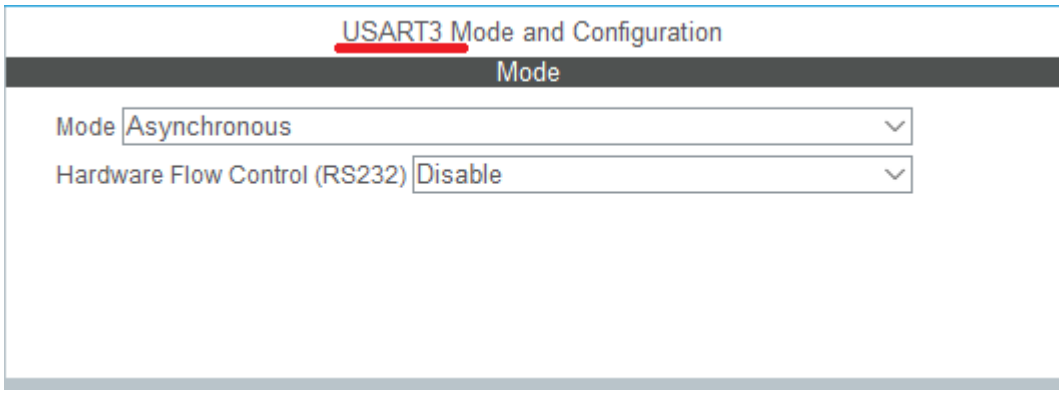
အောက်ပါပုံအတိုင်း network နာမည်၊ model၊ နဲ့ Compression ကိုသတ်မှတ်ပြီးရင် "Analyze" ကိုနှိပ်ပါ။



"Anaylze" လုပ်လို့ပြီးသွားရင် "Validate on desktop" ကိုနှိပ်ပါ။

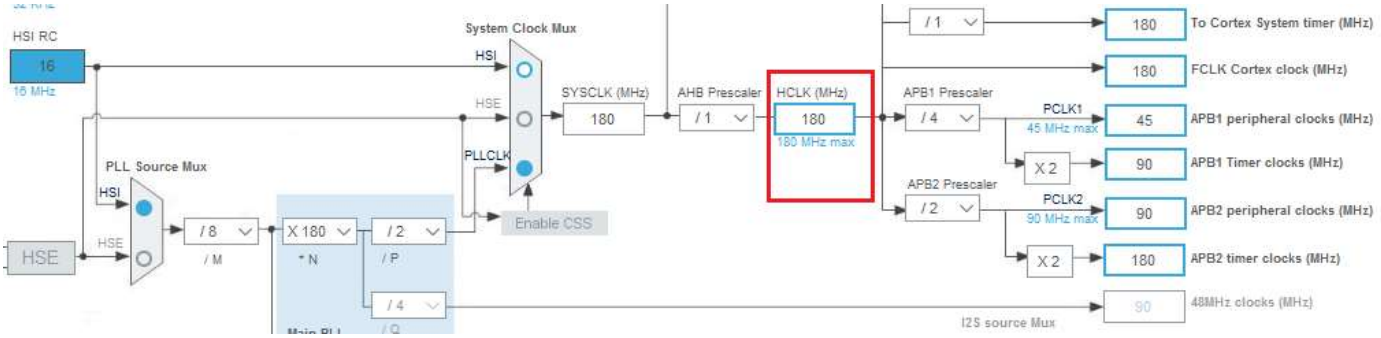


Data တွေကို USART ကနေထုတ်ပြဖို့ရန်အတွက် USART3 ကို "Asynchronous" mode သတ်မှတ်ခဲ့ပါမယ်။

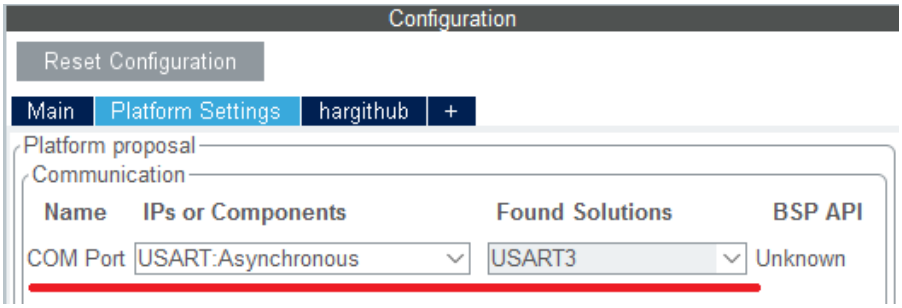


"Neutral Network" ကိုသုံးမယ်ဆိုရင် "CRC" ကမပါမဖြစ်ပါရပါတယ်။ မပါလို့မရပါဘူး။ X-CUBE-AI ကို ပရောဂျက်ထဲမှာထည့်လိုက်ပြီဆိုတာနဲ့ အလိုအလျောက် enable လုပ်ပြီးသားဖြစ်နေပါတယ်။

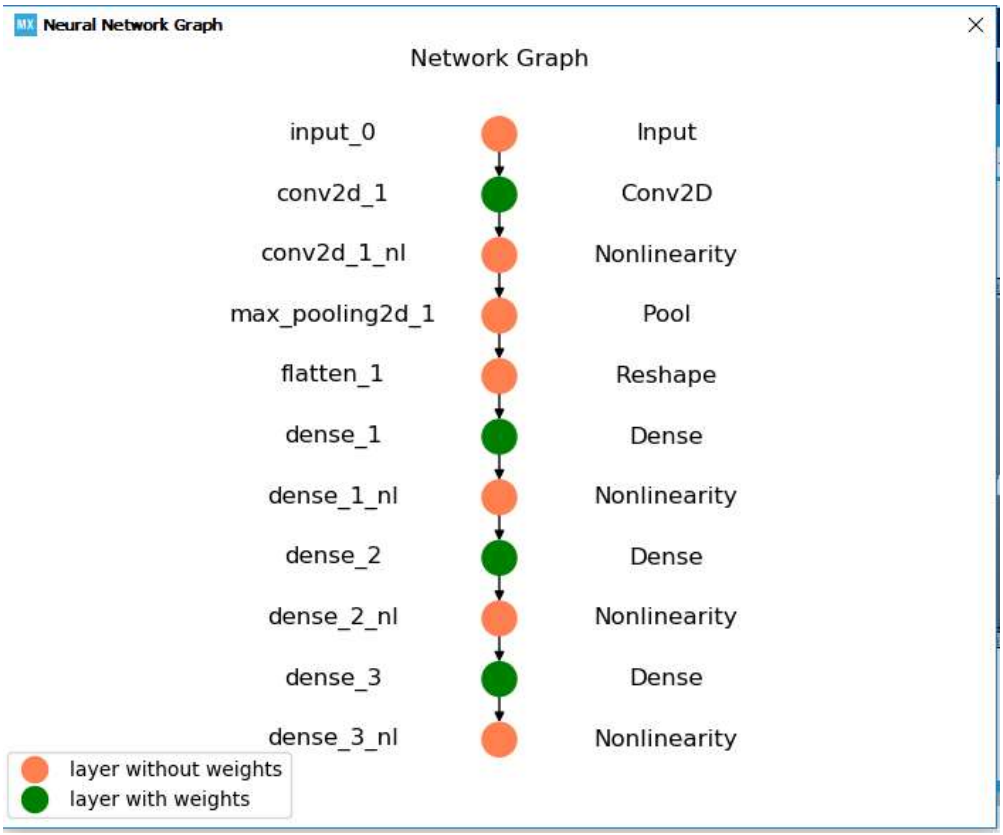
Clock Setting ကို STM32F429 အတွက် အမြင့်ဆုံးပေးနိုင်သလောက် 180 MHz ပေးထားခဲ့ပါမယ်။



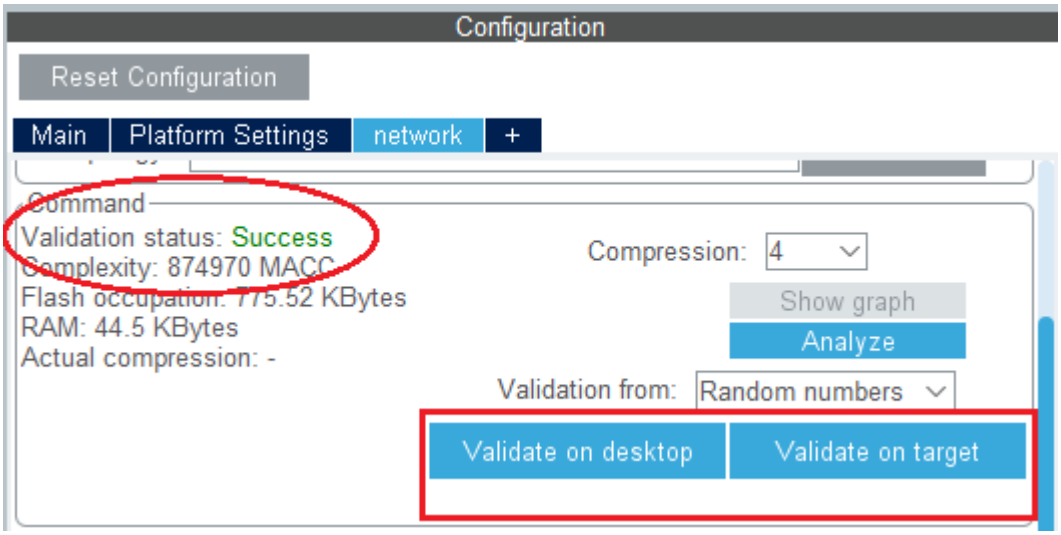
ဒေတာပြဖို့ communication port ကို AI wizard မှာအောက်ပါပုံအတိုင်းပေးထားခဲ့လိုက်ပါမယ်။



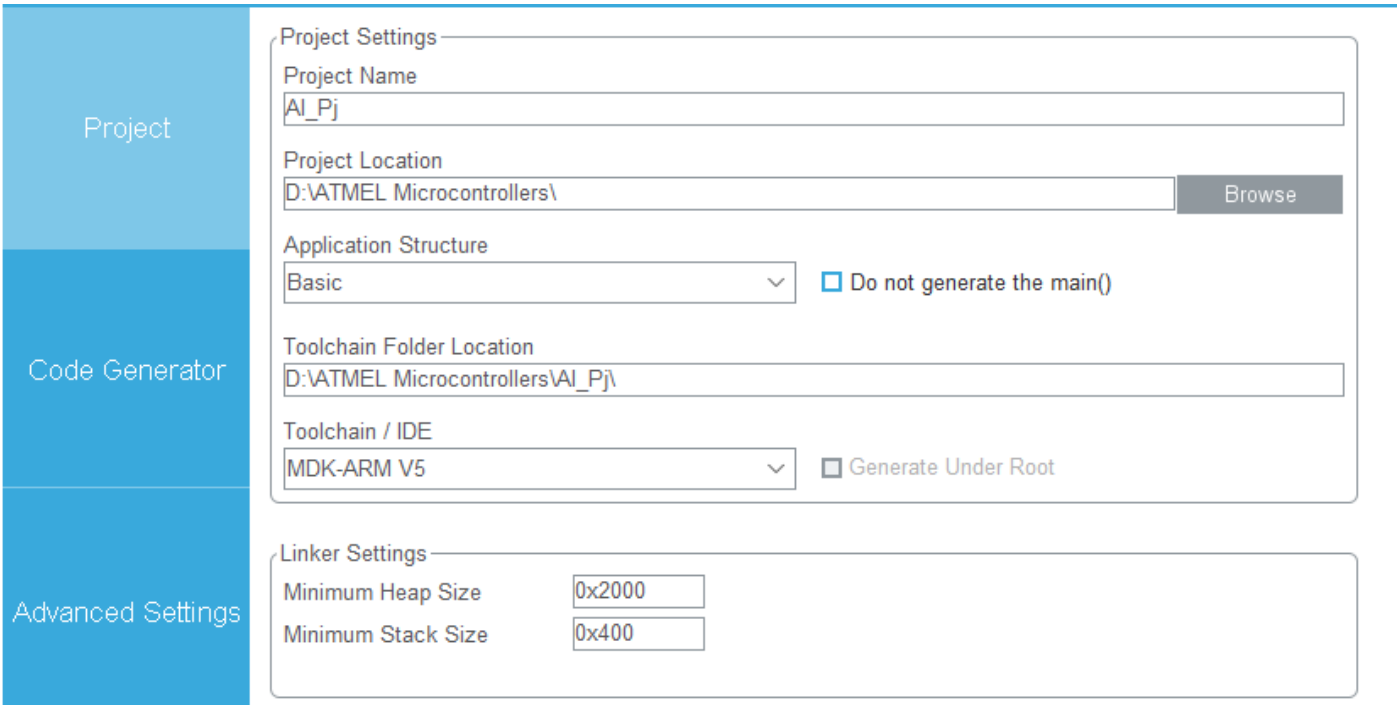
ထည့်လိုက်တဲ့ Deep Learning Model ရဲ့ structural imformation ကိုကြည့်ဖို့အတွက် "Show graph" ဆိုတဲ့ ခလုတ်ကိုနှိပ်ပြီးတော့ ကြည့်နိုင်ပါတယ်။ အ



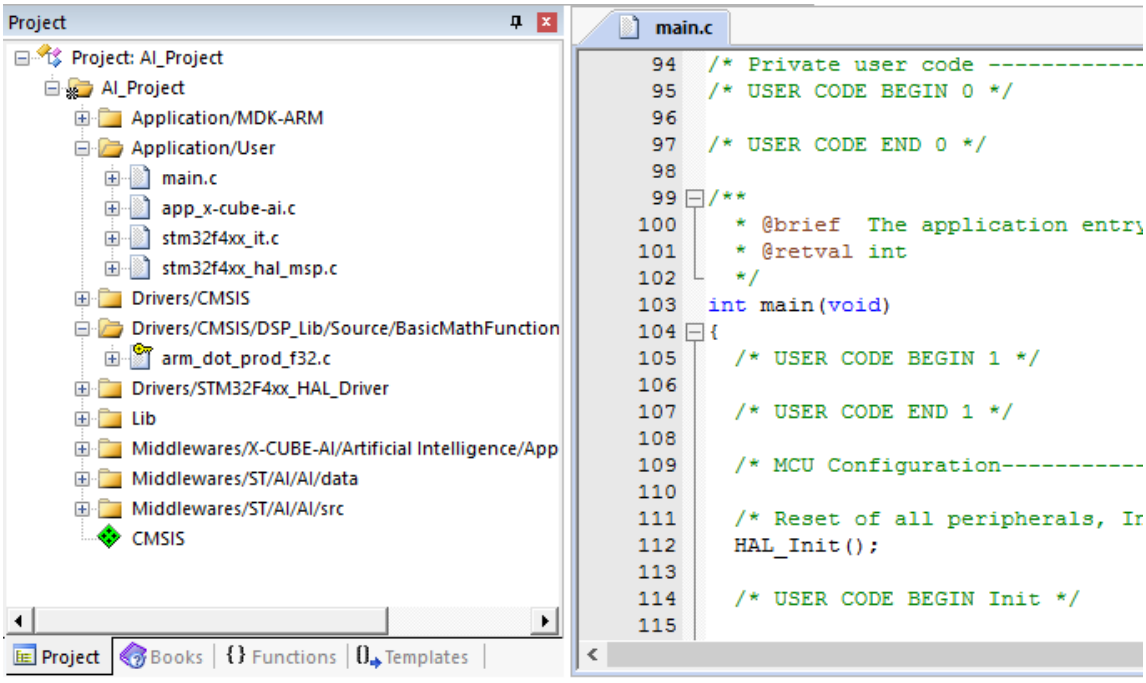
ထုတ်ပေးလိုက်တဲ့ C Model ကို validate လုပ်ဖို့ရန်အတွက် ["validate on desktop"] ဆိုတဲ့ခလုတ်ကို နှိပ်လိုက်ပါ။ Validate လုပ်တာအဆင်ပြေရင် အောက်ပါပုံအတိုင်း "success" ဖြစ်တာကိုမြင်ရမှာဖြစ်ပါတယ်။



Keil uVision 5 မှာ STM32 AI project ကိုရေးနိုင်ဖို့ရန်အတွက် - ထုတ်လိုက်ပါမယ်။ ထုံးစံအတိုင်း ပရောဂျက်ရဲ့နာမည်ကိုပေးမယ်၊ Project Location ပေးမယ် Toolchain ကိုရွေးမယ်။ "AI Validation" ကိုစမ်းလို့အဆင်ပြေအောင် heap size ကို 2 kb အနည်းဆုံးထားခဲ့ရပါမယ်။ ဒါကြောင့် "Minimum Heap Size" ကို 0x2000 ပေးထားခဲ့လိုက်ပါတယ်။ အားလုံးပြီးသွားရင်တော့ "Generate Code" ဆိုတဲ့ခလုတ်ကိုနှိပ်ပြီးတော့ Keil မှာ ကုဒ်ရေးဖို့ရန်အတွက်ထုတ်လိုက်ပါတော့တယ်။



ပရောဂျက်ကိုထုတ်ပြီးရင် build လုပ်လိုက်ပါ။ 0 Error(s), 0 Warning(s) ဆိုရင်တော့ STM32 AI project ကိုအောင်မြင်စွာဖန်တီးပြီးသွားပါပြီ။



Build Output

```
*** Using Compiler 'V5.06 update 6 (build 750)', folder: 'C:\Keil_v5\ARM\ARMCC\Bin'
Build target 'AI_Project'
"AI_Project\AI_Project.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

```
<project_name>
    |- Drivers\CMSIS\DSP_Lib\Source\BasicMathFunctions
    |                                    \_ arm_dot_prod_f32.c
    |- Inc
    |    |- bsp_ai.h                    /* generated files by STMCubeMX */
    |    |- constants_ai.h              /* for AI sample application */
    | ...
    |--Middlewares
        \-- ST/AI
            | /*====================================================================*/
            |-- AI         /* generated NN library by the X-CUBE-AI core */
            |     |-- data
            |     |     |-- <name_1>_data.c     /* specialized NN weight/bias */
            |     |     |-- <name_1>_data.h
            |     |     |-- <name_2>_data.c
            |     |     \-- <name_2>_data.h
            |     |-- include
            |     |     |-- <name_1>.h          /* specialized public/client API */
            |     |     |-- <name_2>.h
            |     |     \- *.h                  /* generic and private header files */
            |     |-- lib
            |     |     \-- network_runtime.a   /* generic run-time library */
            |     \-- src
            |         |-- <name_1>.c            /* specialized NN implementation*/
            |         \-- <name_2>.c
            | /*====================================================================*/
            |
            \-- Application
                    |-- SystemPerformance       /* generic sample application */
                        |-- Inc
                        | \-- aiSystemPerformance.h
                        \-- Src
                            \-- aiSystemPerformance.c
```