

eIQ Toolkit[™] for i.MX RT1170 Model Runner Getting Started

Revision 4 August 2023



Contents

1	Lab Overview	.3
2	Hardware Requirements	.3
3	Software Requirements	.3
4	Model Runner with DeepViewRT.4.1 Import Model Runner Project.4.2 Run the DeepViewRT Model Runner Example4.3 Collect Profiling Information	.3 4 5
5	Model Runner with TFLM or Glow. 5.1 Import Model Runner Glow Project. 5.2 Run the DeepViewRT Model Runner Example.	.9 9 10
6	Conclusion1	1



1 Lab Overview

This document will cover how use Model Runner on an i.MX RT1170 EVK to get model profiling information on a target device. It will also cover how to use Model Runner with Glow or TFLM to send data to those inference engines running on an embedded device over an Ethernet connection. However profiling information is only available when using the DeepViewRT inference engine.

2 Hardware Requirements

This lab is written for the i.MX RT1170 evaluation board. This lab can also be used with the following evaluation boards that support DeepViewRT in the SDK by downloading their respective SDK packages:

- i.MX RT1050
- i.MX RT1060
- i.MX RT1064
- i.MX RT1160
- i.MX RT1170

3 Software Requirements

The following pieces of software are required:

- <u>MCUXpresso IDE</u>
- <u>MCUXpresso SDK for the board being used</u>
 - Make sure to unzip the package as some files inside of it will be used.
- <u>eIQ Toolkit</u>

4 Model Runner with DeepViewRT

If you have a model in the DeepViewRT format, the first step to generating the model profiling information is to load and run the DeepViewRT Model Runner SDK project on the target device. This project can be found in the MCUXpresso SDK.

4.1 Import Model Runner Project

- 1. Open MCUXpresso IDE and select a workspace.
- 2. Add the RT1170 SDK zip file to the Installed SDKs window
- 3. In the QuickStart menu, select Import SDK example(s)...





4. Select the RT1170-EVK (evkmimxrt1170) and click on Next



5. Expand the **elQ** category and select the **deepviewrt_modelrunner_cm7** example. Click on the **Finish** button to import the project into your workspace.

Import projects				
oject name prefix: evkmimut1170	× Project name suffic:			
Use default location				
cation C:\Users\mua06332.WBI\Documents\MCUXpressoiDE_11.7.0_9198\eig\ei	kmimut1170		Brow	ise,
Project Type	Project Options			
C Project C++ Project C Static Library C++ Static Library	SDR Debug Console () Semihost () UART () Example default Capy sources () Import other files			
xamples		ès 8	× F	E
type to filter				
Name	Description	Version		1
Image: An expression Image: An expression	Deployment Crease Label targe example towards the demonstration of the date we object. Desployment Classification of the date of the demonstration of the date of			

4.2 Run the DeepViewRT Model Runner Example

Now build and run the project that was just imported

- 1. Connect an Ethernet cable to the **1G ENET** Ethernet port (J4) and connect the other end to a router that is on the same network as your PC. This is how the board will download the model and images from the eIQ Portal
- 2. Connect a micro-B USB cable to the board on J11
- 3. Open TeraTerm or other terminal program, and connect to the COM port that the board enumerated as. Use 115200 baud, 1 stop bit, no parity as the terminal settings.
- 4. In the Quickstart panel, click on **Build**. You should see no errors in the Console tab.





5. In the Quickstart pane select **Debug**



6. It will ask which interface to use. Select the CMSIS-DAP probe and click on OK.

Probes discovered				
Connect to target: MI	MXRT1176xxxxxx			
1 probe found. Select th	e probe to use:			
Available attache	d probes			
Name	Serial number	Type	Manufactur	IDE Debug Mo
CMSIS-DAP	02280000082c	Link	ARM	Non-Stop
Supported Probes (tick/	untick to enable/disable)			
MCUXpresso IDE Lie	nkServer (inc. CMSIS-DAF	P) probes		
P&E Micro probes				
M SEGGER I-Link prob	es			
Probe search options				
Search again				
Remember my selection	on (for this Launch config	gunation)		
(?)			OK	Cancel
~				

7. The debugger will download the firmware and open up the debug view. Click on the Resume button to start running.

modelrunner - evkmimxrt1170_deepviewrt_modelrunner/source//wip_dhcp_freertos.c - MCUXpresso IDE					
File Edit Source Refactor Navigate Search Project ConfigTools Run	Analysis FreeRTOS Window Help				
🗂 ▾ 🔜 🐚 🗞 ▾ 🗞 ▾ 🖬 🖉 🏷 🖬 🔍 🚺 🔲 🖛 🛪 🌫 🧟	> - 時 - 時 - 時 - 時 - 時 - 日 - 日 - 日 - 日 - 日				
🍋 Project Explorer 🔅 👫 Peripherals+ 🚟 Registers 🗱 Faults 👘 🗖 🗌	† Debug ⊠				
Image: Second					
> 25 component > 26 decic > 26 doc > 26 drivers > 26 drivers > 26 freetos > 26 hisip > 26 hisip > 26 hisip > 26 hisip > 26 hisip	modelnumer. [gdb00]proc(42000].threadGroup[1].gdb00]proc(42000].OSthread[1].thread[1].frame[0] [ghvig.thcp_freetosc [sd229/r] [ghvig.thcp_freetosc [sd239/r] [sd259/r] [sd259/r]				
B source FreeRTOSConfig.h being there for a the a	450 #if defined(CPU_MIWART1064DVL6A) 451 80ARD_InitBootPins(); 452 80ARD_InitBootPins();				

8. In the terminal you should see the board acquire an IP address from the router. Make note of this IP address as it'll be used in the next section.



4.3 Collect Profiling Information

Now that firmware is loaded on the board, eIQ Toolkit can be used to collect the profiling information for that DeepViewRT model.

- 1. Open eIQ Toolkit
- 2. From the Home screen, click your existing eIQ Portal project. You can use the Flower project from the previous lab document.
- 3. Make your way to the Validation screen like done in the previous lab document.



4. By default the validation will run on the local PC. But you can run it on the EVK instead by click on the **Validation Target** button



5. Click on Add Device in the dialog box that comes up



6. Then add the board by giving it a name and the IP address that was printed out into the terminal, with the port **:10818** at the end. Then click on **ADD**

			Remote Device	es	
nline De	vices			CANC	EL .
Name	RT1170-E	/K			
URL	http:// 192	168.86.61:10818			_

7. Then click on the new board just added to that it says Selected underneath it and turns purple. And then click on Close at the bottom.

		Remote Devices		
Inline Devices			SCAN DEVICES	ADD DEVICE
Default 127.0.0.1:10818	RT1170-EVK 192.168.86.61:10818 Selected			
Iffline Devices				
NXP i MX 8 Series An	plications Processors - Learn Mor			
NXP I.MX RT Crossov DeepView Vision Star	er MCUs – Learn More ter Kits – Learn More			
		CLOSE		



8. Verify the **Target Name** and **Target URI** look correct for your board and then click on **Validate** to download the model to the RT1170-EVK over the Ethernet connection and start giving it images from your Dataset to do the validation on.

elQ Portal flowers		
VALIDATION SETTINGS	^	
Softmax Threshold	0	L
•		U.
		L
Use Quantized Model	Not Available	L
Input Data Type		U.
Maintain Existing Da	ta Type 🗸 🗸	U.
Output Data Type		
Maintain Existing Da	ta Type 👻	
Post Training Quantiza	ation	L
VALIDATION STATS	^	L
Target Name	RT1170-EVK	U.
Target URI	192.168.86.6	L
		U.
Accuracy	-	U,
Samples Count		
Model Upload		
Image Decode	•	
VALIDATION	TARGET	
VALIDA	ATE	
DEEPVIEW DEVP	ACK ADD-ON	
Z TRAINER	DEPLOY >	

9. eIQ Portal will begin sending over the Ethernet cable the images in the Validation category to the board for inferencing.



10. You should see a confusion matrix come up that shows how accurate the model is for different images as well as the overall execution time for an inference. In this case it was 119ms.





11. The profile information can be viewed by clicking on the **Model Tool** button



12. You should see the model diagram come up with the timings as recorded on the EVK:



13. You can click on a particular operation to see the time in milliseconds and the percentage of overall time that that particular layer used. This can be used to decide which layers to optimize.

elQ Model Tool 2.7.5		- 🗆 ×
File Edit View ManageTargets Profiling Help		
	NODE PROPERTI	ts ×
	type	conv2d ?
	name	model/expanded_conv_depthwise_relu/Relu6;model/expanded_conv_de
1+128+128+3	ATTRIBUTES	
	activation	reluó +
conv2d weights (1+1+1+1+1)	dilations	[0.0.0] +
bias (10) http://192.168.06.61.10818.7.885	groups	16 *
retu6	head	[0,1,1,0] ÷
1-04-04-10	strides	[UUU] *
com/2d	tail	[0,1,1,0] *
weights: (1) = 3 - 16 - 1) blac: (16) http://1782 64.64.108108 http://1782 64.64.108108	INPUTS	
reluti	input	name: model/Conv1_relu/Relu6;model/bn_Conv1/fusedBatchNc *
1×64×64×10	weights	name: model/expanded_conv_depthwise_BN/FusedBatchNormV *
com/2d weights (1,1,16,4)	bias	name: model/expanded_conv_depthwise_BN/fusedBatchNormV *
Naio (3) http://102.168.84.61.10888.2.2.65	OUTPUTS	
1×64×64×8	model/expande	name: model/expanded_conv_depthwise_relu/Relui(model/expi *
weights: (1+1+(=+40) bias (42)	TIMINGS	
http://192.160.86.61.108188.7.0844 retufi	http://192.168.8	8.90900 ms (7.50244 %)
Target Attached Successfully	ADD THIS N INPUT OV	ADD THIS NODE AS ERRIDE OUTPUT OVERRIDE
CLOSE		



5 Model Runner with TFLM or Glow

Model Runner can also be used to send images from a PC to a Glow model running on the board. This can be useful when verifying a Glow model on hardware. The same can also be done with TensorFlow Lite Micro (TFLM). Note that with these two inference engines that profiling information cannot be collected.

5.1 Import Model Runner Glow Project

- 1. Open MCUXpresso IDE and select a workspace.
- 2. Add the RT1170 SDK zip file to the Installed SDKs window if not done in previous section.
- 3. In the QuickStart menu, select Import SDK example(s)...



4. Select the RT1170-EVK (evkmimxrt1170) and click on Next



5. Expand the **elQ** category and select the **deepviewrt_modelrunner_glow_cm7** example. Click on the **Finish** button to import the project into your workspace.





5.2 Run the DeepViewRT Model Runner Example

Now build and run the project that was just imported

- 6. Connect an Ethernet cable to the **1G ENET** Ethernet port (J4) and connect the other end to a router that is on the same network as your PC. This is how the board will download the model and images from the eIQ Portal
- 7. Connect a micro-B USB cable to the board on J11
- 8. Open TeraTerm or other terminal program, and connect to the COM port that the board enumerated as. Use 115200 baud, 1 stop bit, no parity as the terminal settings.
- 9. In the Quickstart panel, click on **Build**. You should see no errors in the Console tab.



10. In the Quickstart pane select **Debug**

🕛 Quickstart Panel 💥 🗱 🗱	riables 🧕 Breakpoints	
MCUXpresso ID Project: evkmimxrt117	E - Quickstart Pa 0_deepviewrt_modelrun	nel ner-glow [Debug]
Create or import a project		
New project Import SDK examp Import project(s) f	ole(s) rom file system	
✓ Build your project		
Suild Clean		
		📧 🕶 🔛 👻 💌
Terminate, Build an	nd Debug	

11. It will ask which interface to use. Select the CMSIS-DAP probe and click on OK.

ailable attach	ed probes			
Name	Serial number	Type	Manufactur	IDE Debug Mo
CMSIS-DAP	02290000082c	Link	ARM	Non-Stop
Supported Probes (tick	/untick to enable/disable))		
MCUXpresso IDE L Dist Micro prober	inkserver (inc. CMSIS-DAP	r) probes		
SEGGER J-Link pro	bes			
Probe search options				

12. The debugger will download the firmware and open up the debug view. Click on the Resume button to start running.

modeiranner - evkmimski i roljaepviewit_modeiranner/soarce/wip_incertos.c - wcoopresso ibe					
File Edit Source Refactor Navigate Search Project ConfigTools Ru	n Analysis FreeRTOS Window Help				
🗂 ▾ 📾 🐚 🕷 ▾ 🚳 🖓 🐃 💷 🕷 🛑 💷 🛤 🌫 🤅	o				
🍋 Project Explorer 🛛 🚼 Peripherals+ 🕮 Registers 🎋 Faults 🛛 🖓 🗖	称 Debug 23				
🖻 🧐 🏹 🖶 🤣 🗶 🖌 🕴	IS evkmimxrt1170_deepviewrt_modelrunner LinkServer Debug [C/C++ (NXP Semiconductors) MCU Application]				
 ✓ iffer ekminunt1170, deepviewrt, modelrunner «Master» < Debug» > Ø Project Settings > Ø Includes > Ø Includes > Ø CMSIS > Ø bord 					
> 😂 component	a modelrunner.c C (adb/01.proc(420001.threadGroup[i11.adb/01.proc(420001.OSthread[11).thread[11.frame[0] C Iwip dhcp freertos.c 23				
> 🖾 device	441				
> 🖉 drivers	4420 /*1				
> 🐸 eiq	443 " gorlet Main function. 444 */				
> 😂 freertos	458 int main(void)				
> 😝 Iwip	447 gpio pin config t gpio config = {kGPIO DigitalOutput, 0, kGPIO NoIntmode};				
> 😫 phy	448				
V 🕮 source	449 BOARD_COTT_BAPU(); 450 Bif defined(DULNTMATIAGADU(6A)				
> 📔 FreeRTOSConfig.h	451 BOARD_InitBootPins();				
> c lwip_dhcp_freertos.c	8452 BOARD InitBootClocks();				



13. In the terminal you should see the board acquire an IP address from the router. Make note of this IP address as it'll be used in the next section.



- 14. The project is running a Mobilenet image classification model which can categorize images into one of 1,000 different categories. You can then send images via Ethernet to the RT1170 board running that Glow model.
- 15. Open up a Windows command prompt
- 16. Navigate to an example panda image named **sample.jpg** in the RT1170 SDK at **<SDK unzip location>\boards\evkmimxrt1170\eiq_examples\deepviewrt_labelimage\source\data**
- 17. In the Windows Command Prompt type the following which will use a program called curl to send the image to the board using its IP address from the previous step. You will get the inference time in nanoseconds and the result. This command should be one long continuous line:

curl -D - -XPOST -H "Content-Type: image/*" --data-binary "@sample.jpg" "http://192.168.86.61:10818/v1?run=1"



- 18. The label number for this model is offset by 1 from this list of labels found <u>here</u>. The model should correctly identified the image as a giant panda, which is item 388 in that list due to the offset.
- 19. The default models that come with the SDK example can be replaced with your own custom model by replacing the model.c/.h/.o files in the example.
- 20. If using TFLM make sure to use the Release target for high optimization compiler settings which will have a large impact on inference time for that inference engine.

6 Conclusion

This lab demonstrated how to use the Model Runner feature to do profiling of a model on an embedded system.