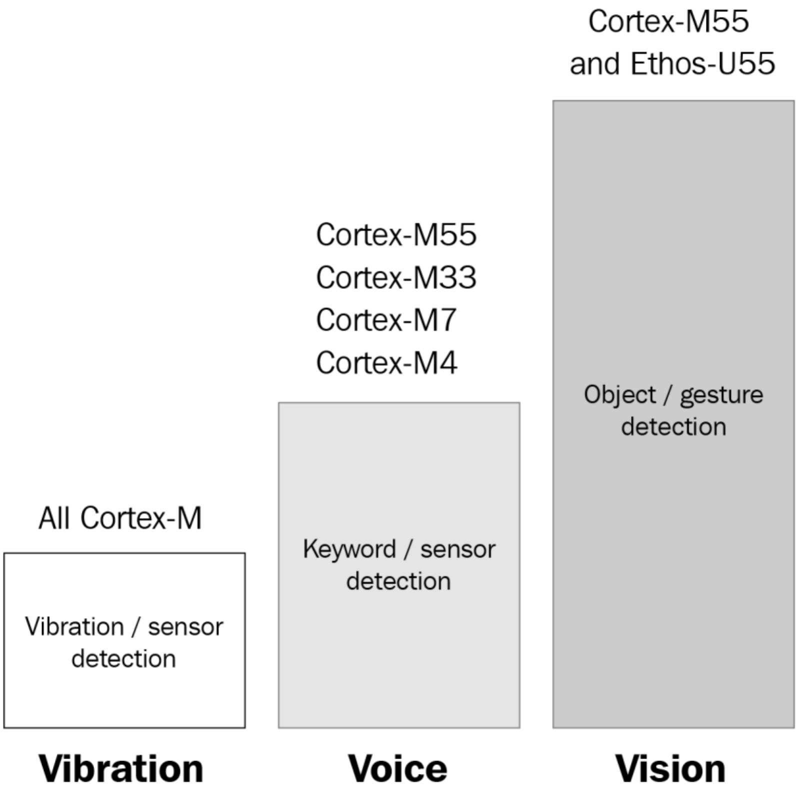
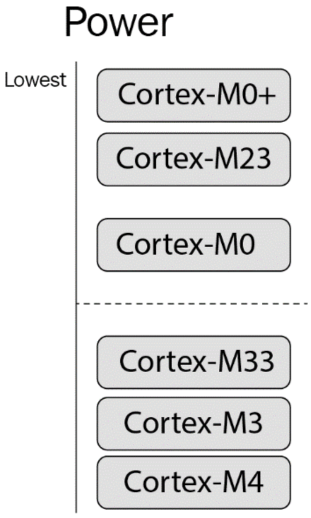
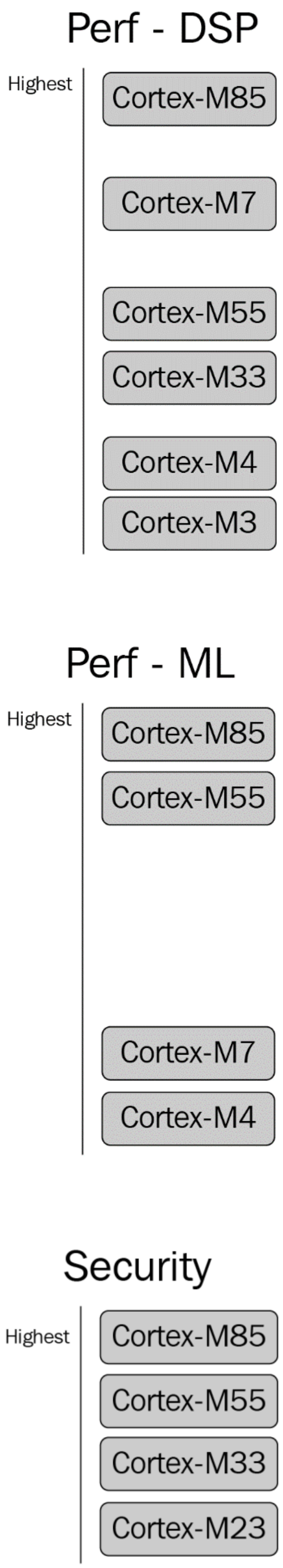
Chapter 1: Selecting the Right Hardware







Perf - DSP

Highest

Cortex-M85

Cortex-M7

Cortex-M55

Cortex-M33

Cortex-M4

Cortex-M3

Perf - ML

Highest

Cortex-M85

Cortex-M55

Cortex-M7

Cortex-M4

Security

Highest

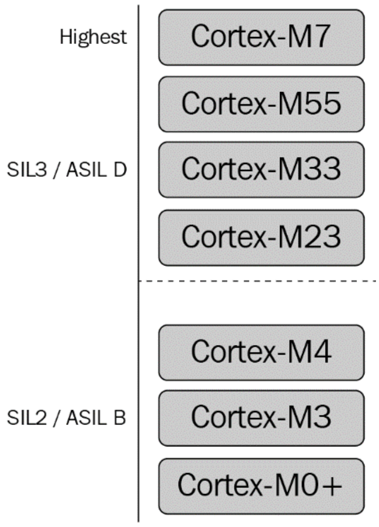
Cortex-M85

Cortex-M55

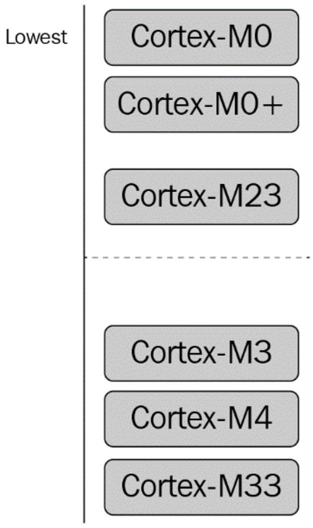
Cortex-M33

Cortex-M23

Safety


Highest
Cortex-M7
Cortex-M55
SIL3 / ASIL D
Cortex-M33
Cortex-M23
Cortex-M4
SIL2 / ASIL B
Cortex-M3
Cortex-M0+

Cost / Area


Lowest
Cortex-M0
Cortex-M0+
Cortex-M23
Cortex-M3
Cortex-M4
Cortex-M33

Arm Cortex-M0+

Dynamic Power	3.8µW/MHz

Arm Cortex-M23

Dynamic Power	3.86µW/MHz

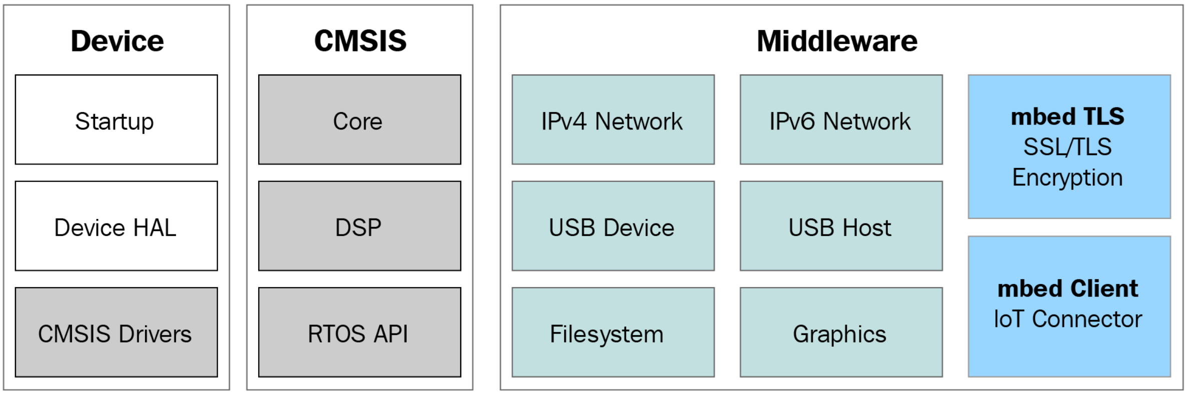
Arm Cortex-M4

Dynamic Power	12.26µW/MHz

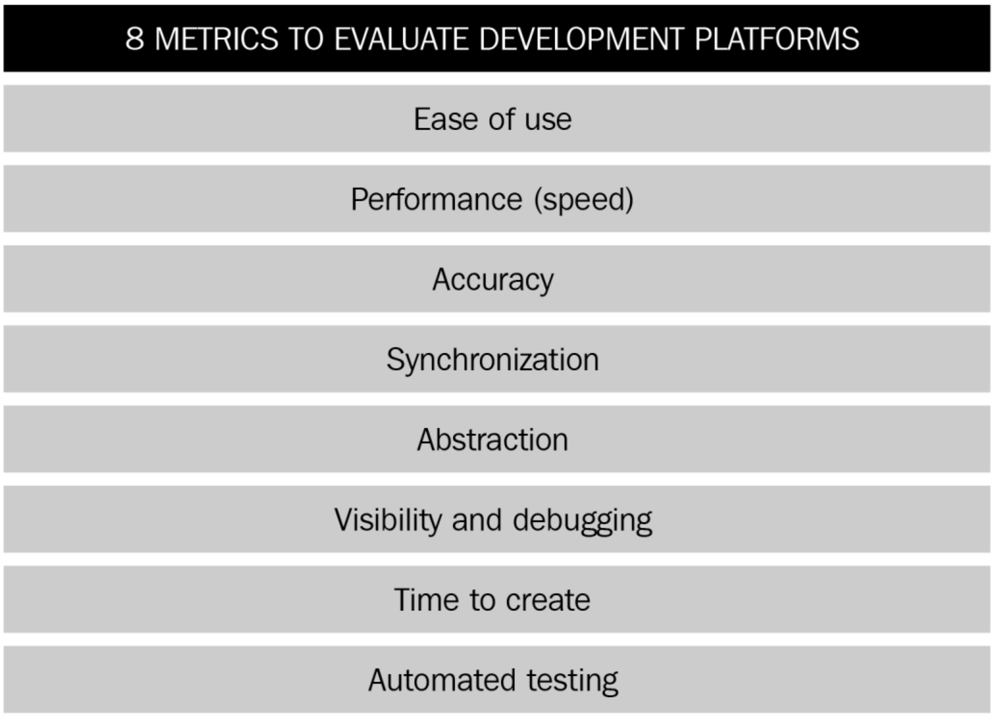
Arm Cortex-M33

Dynamic Power	12.0µW/MHz

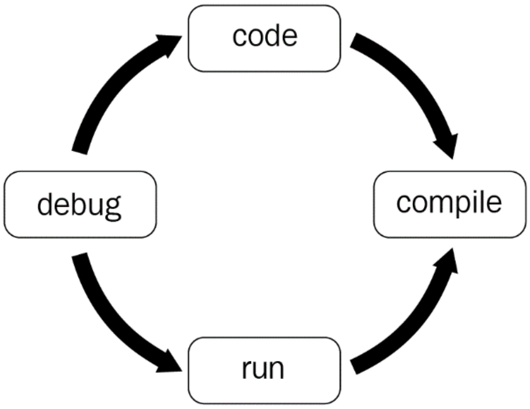
Chapter 2: Selecting the Right Software



Chapter 3: Selecting the Right Tools



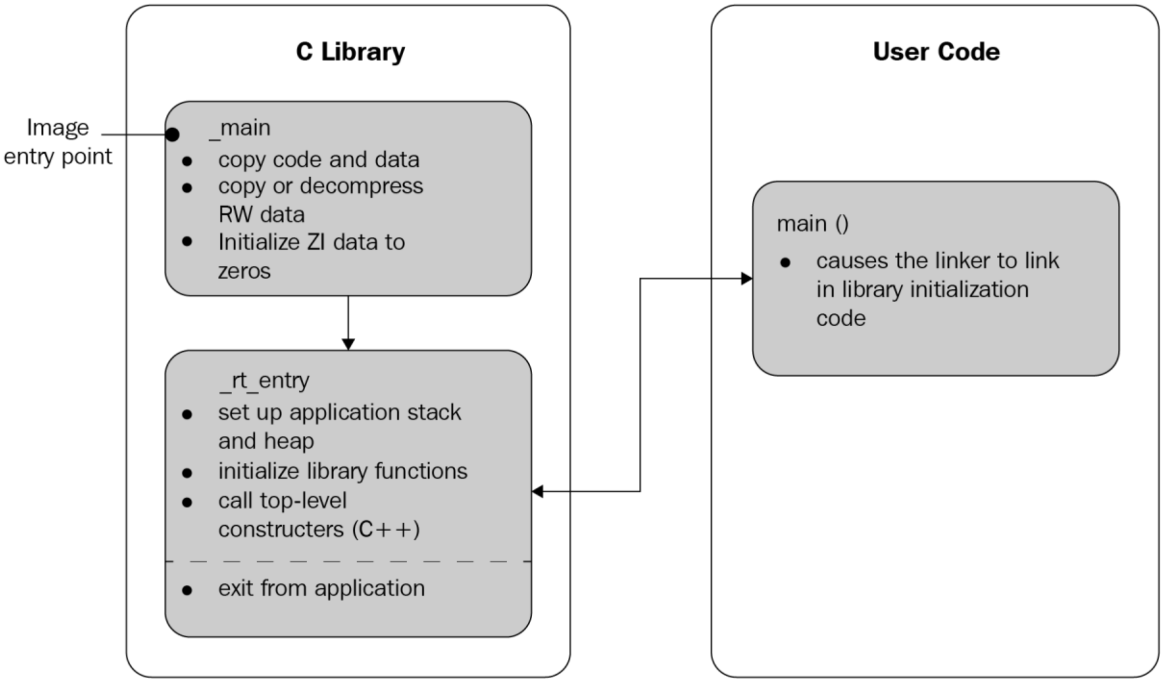
8 METRICS TO EVALUATE DEVELOPMENT PLATFORMS
Ease of use
Performance (speed)
Accuracy
Synchronization
Abstraction
Visibility and debugging
Time to create
Automated testing

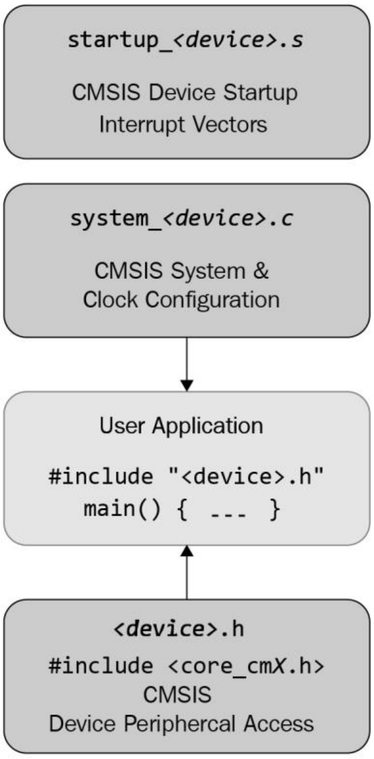


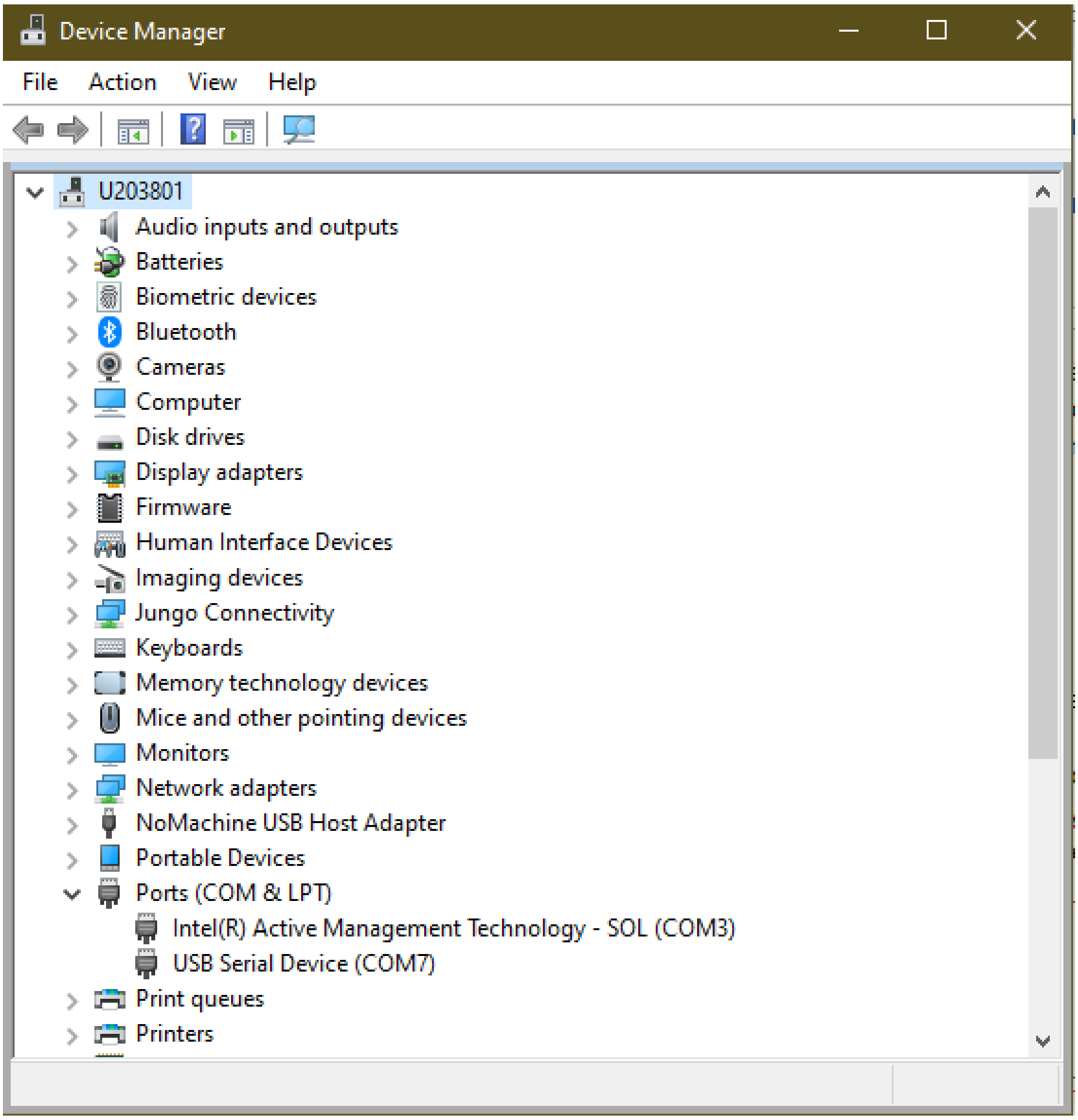
	Interactive	**Automated**
Simplicity Mostly all developers know how to use the environment and tools well	**Critical** Where most developers do day-to-day work, must be as streamlined as possible	**Helpful** Typically, only one or a few dedicated members actively work on the details
Replicability The same errors can be seen in other developers' environments	**Helpful** May not be developing the same area of code, or when working alone	**Critical** To streamline development, enabling all developers to be able to see the error
Longevity Able to be used for longer than the current project	**Depends** May need to change environments to match the needs of diverse projects, but it certainly helps to have a consistent bit for years to come	**Critical** The value of scaling up tests comes when you don't have to keep setting up unique automated flows each time

	VirtualBox	**Multipass**	**Docker**
Host OS	Any	Any	Any
Guest OS	Any	Linux (Ubuntu)	Any
Image size	**Gigabytes** (**GB**)	GB	**Megabytes** (**MB**)
Startup time	Minutes	A minute	Seconds
Ideal for	Robust GUI development	Command-line general Linux development	Specific, repeatable services

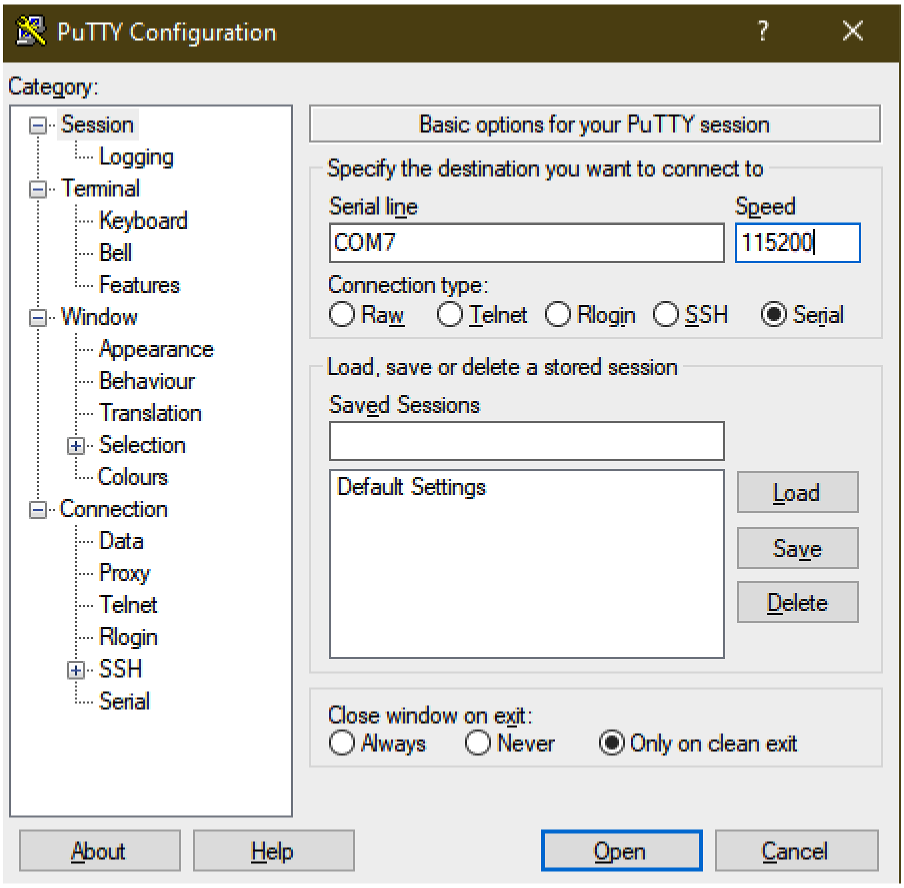
Chapter 4: Booting to Main


C Library
User Code
Image entry point
_main
• copy code and data
• copy or decompress RW data
• Initialize ZI data to zeros
_rt_entry
• set up application stack and heap
• initialize library functions
• call top-level constructers (C++)
• exit from application
main ()
• causes the linker to link in library initialization code





U203801
- Audio inputs and outputs
- Batteries
- Biometric devices
- Bluetooth
- Cameras
- Computer
- Disk drives
- Display adapters
- Firmware
- Human Interface Devices
- Imaging devices
- Jungo Connectivity
- Keyboards
- Memory technology devices
- Mice and other pointing devices
- Monitors
- Network adapters
- NoMachine USB Host Adapter
- Portable Devices
- Ports (COM & LPT)
  - Intel(R) Active Management Technology - SOL (COM3)
  - USB Serial Device (COM7)
- Print queues
- Printers



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM7

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

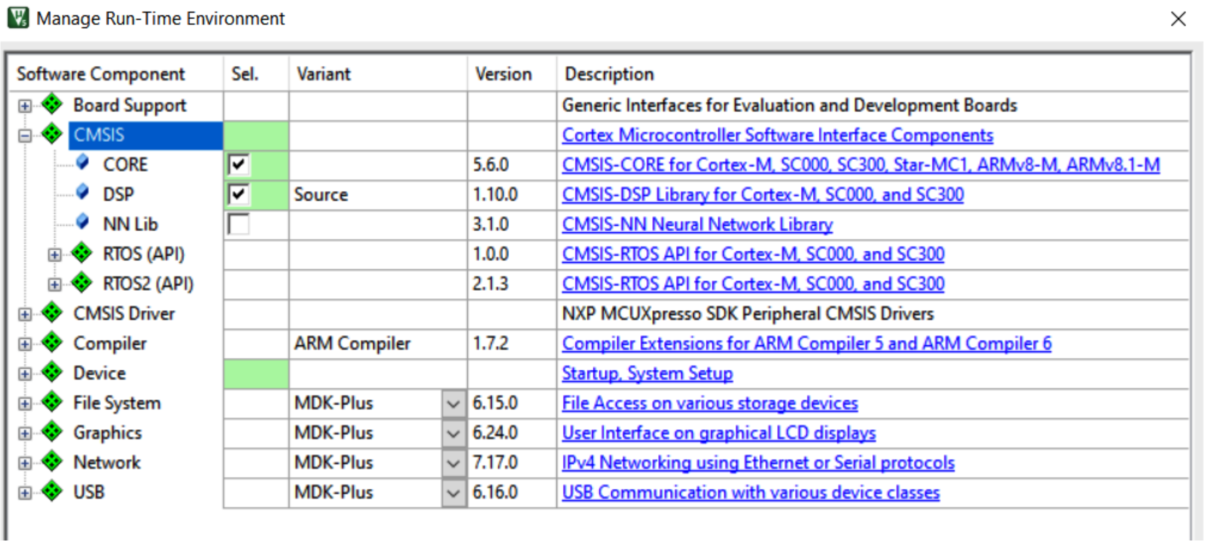
Load, save or delete a stored session

Saved Sessions

Default Settings

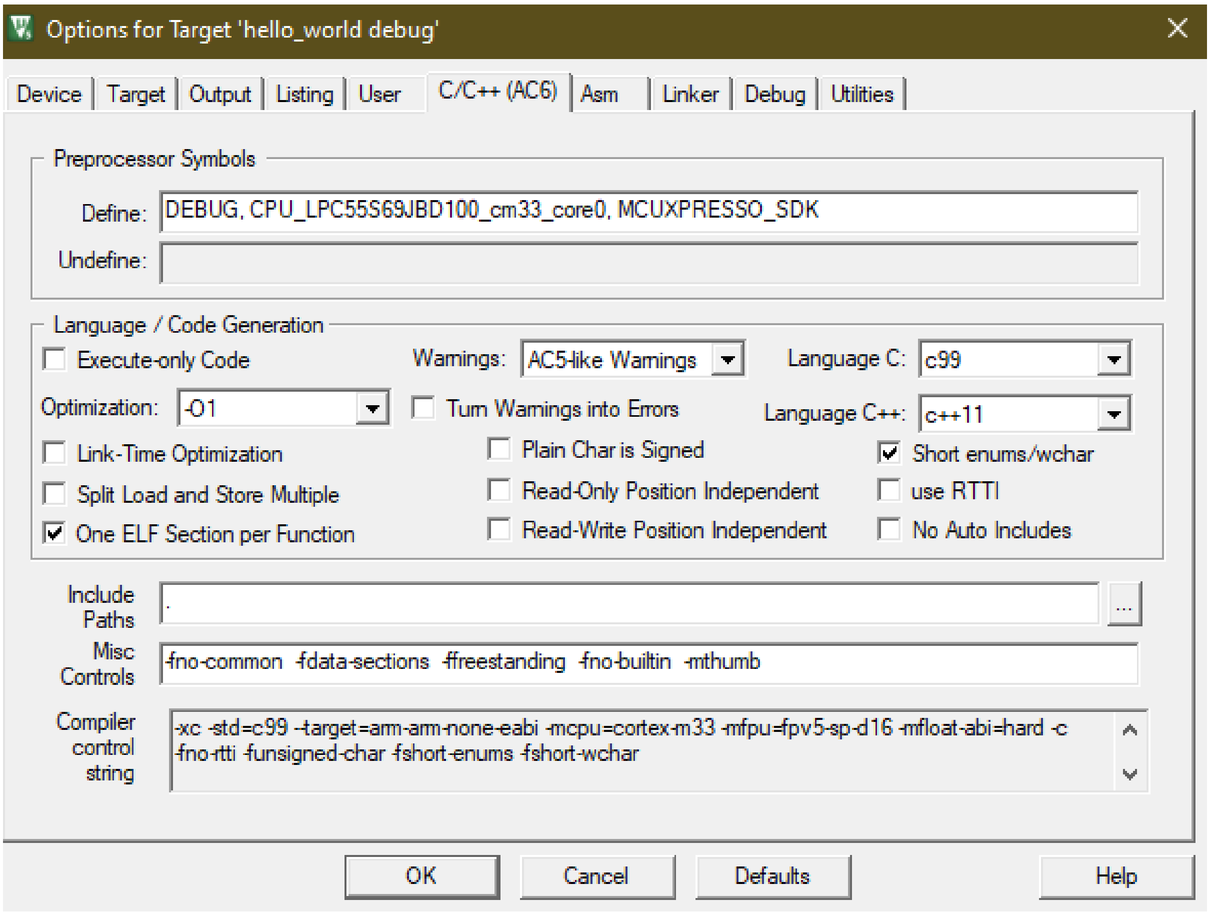
Close window on exit: ○ Always ○ Never ● Only on clean exit

Chapter 5: Optimizing Performance



Manage Run-Time Environment

Software Component	Sel.	Variant	Version	Description
Board Support				Generic Interfaces for Evaluation and Development Boards
CMSIS				Cortex Microcontroller Software Interface Components
CORE	☑		5.6.0	CMSIS-CORE for Cortex-M, SC000, SC300, Star-MC1, ARMv8-M, ARMv8.1-M
DSP	☑	Source	1.10.0	CMSIS-DSP Library for Cortex-M, SC000, and SC300
NN Lib	□		3.1.0	CMSIS-NN Neural Network Library
RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver				NXP MCUXpresso SDK Peripheral CMSIS Drivers
Compiler		ARM Compiler	1.7.2	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Device				Startup, System Setup
File System		MDK-Plus	6.15.0	File Access on various storage devices
Graphics		MDK-Plus	6.24.0	User Interface on graphical LCD displays
Network		MDK-Plus	7.17.0	IPv4 Networking using Ethernet or Serial protocols
USB		MDK-Plus	6.16.0	USB Communication with various device classes



Options for Target 'hello_world debug'

Device | Target | Output | Listing | User | C/C++ (AC6) | Asm | Linker | Debug | Utilities

Preprocessor Symbols

Define: DEBUG, CPU_LPC55S69JBD100_cm33_core0, MCUXPRESSO_SDK

Undefine:

Language / Code Generation

□ Execute-only Code

Warnings: AC5-like Warnings

Language C: c99

Optimization: -O1

□ Turn Warnings into Errors

Language C++: c++11

□ Link-Time Optimization

□ Plain Char is Signed

☑ Short enums/wchar

□ Split Load and Store Multiple

□ Read-Only Position Independent

□ use RTTI

☑ One ELF Section per Function

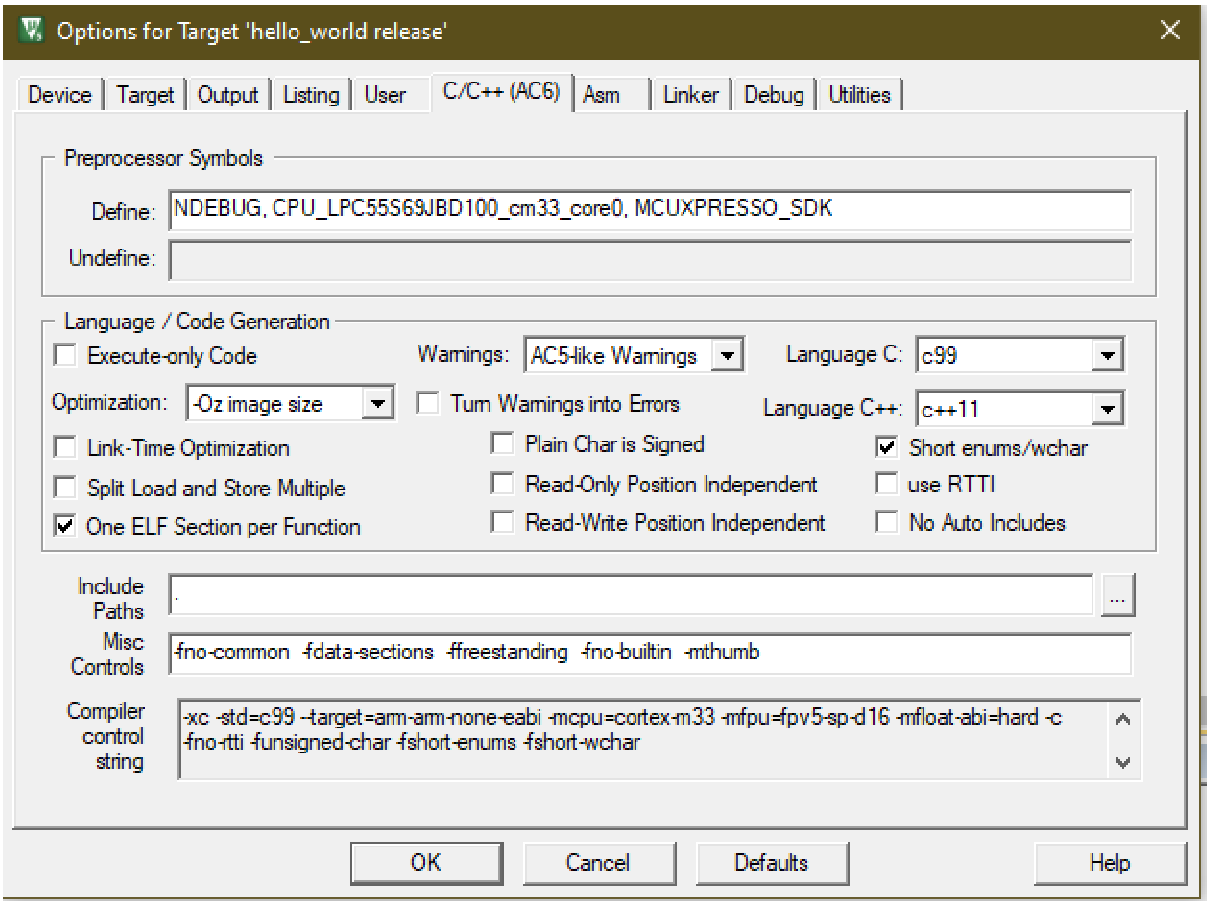
□ Read-Write Position Independent

□ No Auto Includes

Include Paths: .

Misc Controls: -fno-common -fdata-sections -ffreestanding -fno-builtin -mthumb

Compiler control string: -xc -std=c99 --target=arm-arm-none-eabi -mcpu=cortex-m33 -mfpu=fpv5-sp-d16 -mfloat-abi=hard -c -fno-rtti -funsigned-char -fshort-enums -fshort-wchar



Preprocessor Symbols

Define: NDEBUG, CPU_LPC55S69JBD100_cm33_core0, MCUXPRESSO_SDK

Undefine:

Language / Code Generation

□ Execute-only Code

Warnings: AC5-like Warnings

Language C: c99

Optimization: -Oz image size

□ Turn Warnings into Errors

Language C++: c++11

□ Link-Time Optimization

□ Plain Char is Signed

☑ Short enums/wchar

□ Split Load and Store Multiple

□ Read-Only Position Independent

□ use RTTI

☑ One ELF Section per Function

□ Read-Write Position Independent

□ No Auto Includes

Include Paths: .

Misc Controls: -fno-common -fdata-sections -ffreestanding -fno-builtin -mthumb

Compiler control string:

```
-xc -std=c99 --target=arm-arm-none-eabi -mcpu=cortex-m33 -mfpu=fpv5-sp-d16 -mfloat-abi=hard -c
-fno-rtti -funsigned-char -fshort-enums -fshort-wchar
```

CMAKE_BUILD_TYPE	GCC flags used for optimization
Release	`-O3` (max optimization)
Debug	`-Og` and `-g` (max debug)
RelwithDebInfo	`-O2` and `-g` (good optimization with debug)
MinSizeRel	`-Os` (minimum code size)

Implementation	Number of cycles
1: Plain C code	41,668
2: CMSIS-DSP for multiply then add	54,539
3: CMSIS-DSP for dot product	41,808

Ideal for	Arm Compiler for Embedded flags for optimization
Performance	`-Ofast or -O3`
Debug, low performance	`-O0`
Debug, more optimized	`-O1`
Minimum code size	`-Oz`

Implementation	Number of cycles
1: Plain C code	2,398
2: CMSIS-DSP for multiply then add	9,765
3: CMSIS-DSP for dot product	2,422
Minimum code size	-Oz

Implementation	Number of cycles
1: Plain C code	42,696
2: CMSIS-DSP for multiply then add	56,902
3: CMSIS-DSP for dot product	42,937

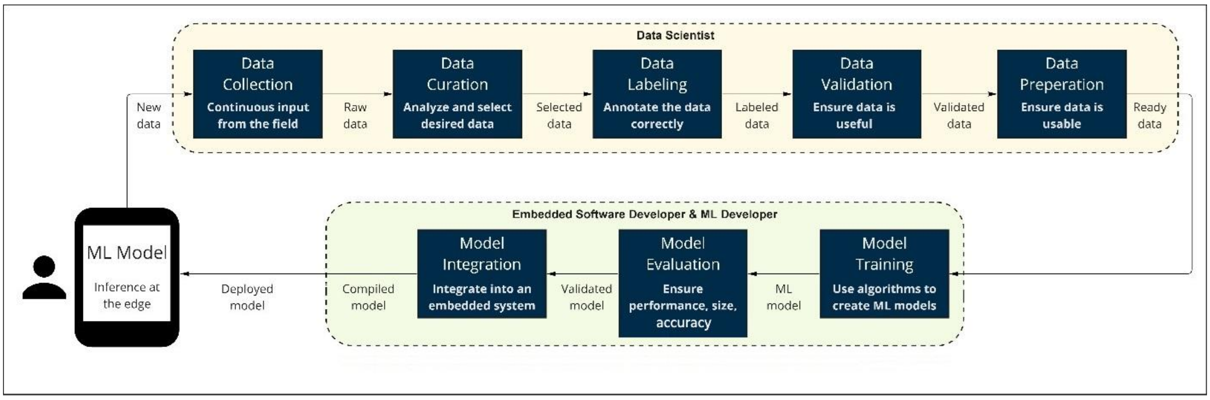
Implementation	Number of cycles
1: Plain C code	41,668
2: CMSIS-DSP for multiply then add	60,887
3: CMSIS-DSP for dot product	41,034

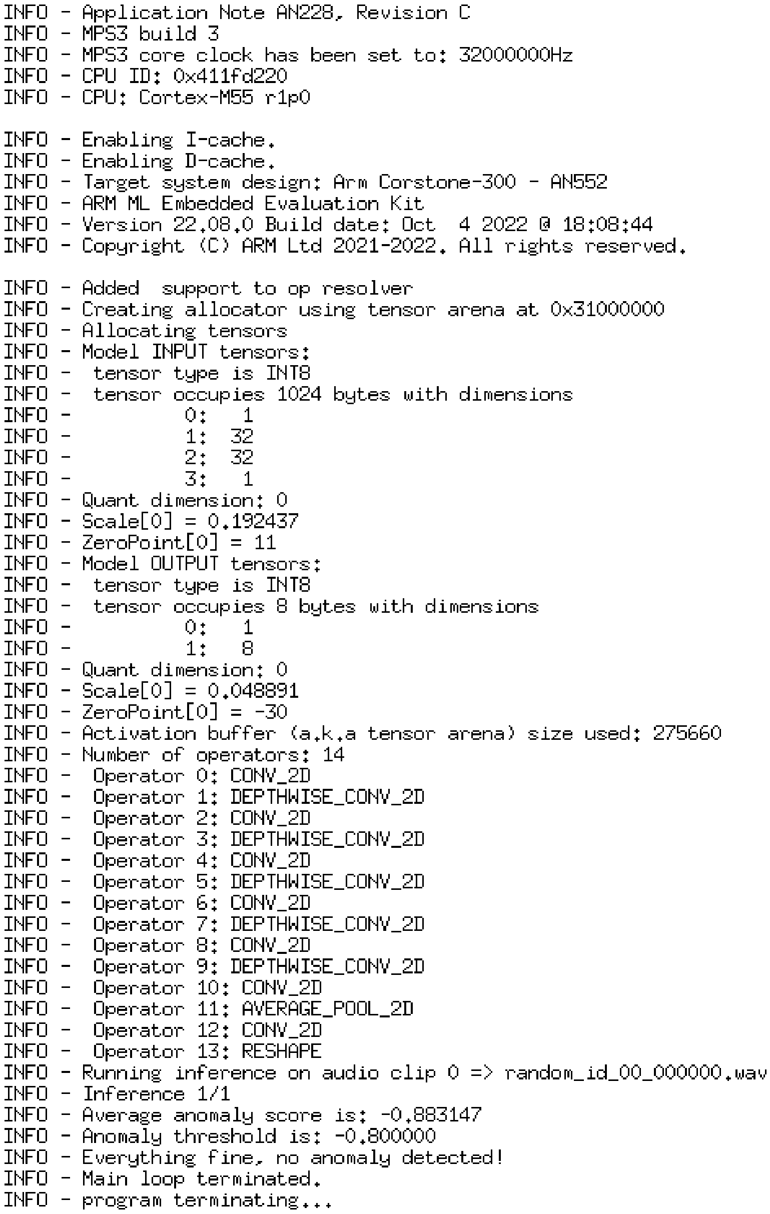
Implementation	Number of cycles
1: Plain C code	2,100
2: CMSIS-DSP for multiply then add	8,891
3: CMSIS-DSP for dot product	1,857

Implementation	Number of cycles
1: Plain C code	1,752
2: CMSIS-DSP for multiply then add	11,246
3: CMSIS-DSP for dot product	1,652

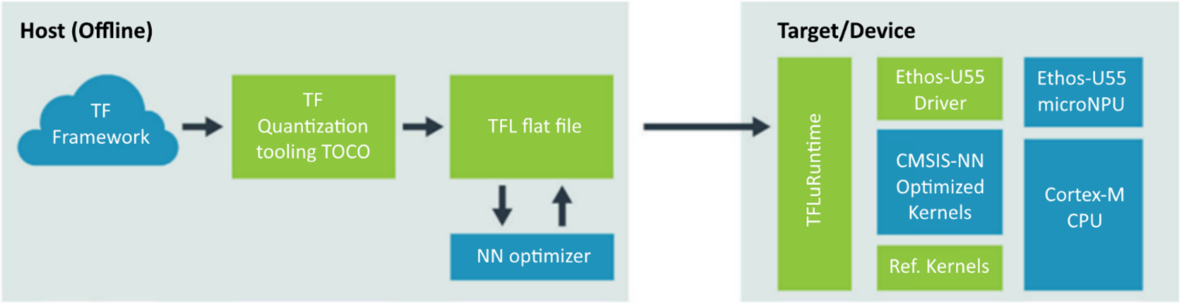
Implementation	Number of cycles
1: Plain C code	4,294,967,295
2: CMSIS-DSP for multiply then add	11,390,997
3: CMSIS-DSP for dot product	5,201,475

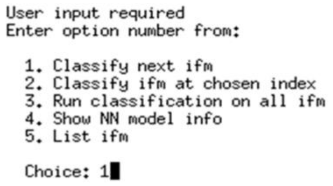
Chapter 6: Leveraging Machine Learning



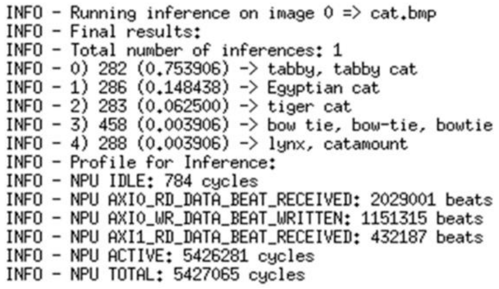


```
INFO - Application Note AN228, Revision C
INFO - MPS3 build 3
INFO - MPS3 core clock has been set to: 32000000Hz
INFO - CPU ID: 0x411fd220
INFO - CPU: Cortex-M55 r1p0

INFO - Enabling I-cache.
INFO - Enabling D-cache.
INFO - Target system design: Arm Corstone-300 - AN552
INFO - ARM ML Embedded Evaluation Kit
INFO - Version 22.08.0 Build date: Oct  4 2022 @ 18:08:44
INFO - Copyright (C) ARM Ltd 2021-2022. All rights reserved.

INFO - Added  support to op resolver
INFO - Creating allocator using tensor arena at 0x31000000
INFO - Allocating tensors
INFO - Model INPUT tensors:
INFO -  tensor type is INT8
INFO -  tensor occupies 1024 bytes with dimensions
INFO -          0:   1
INFO -          1:  32
INFO -          2:  32
INFO -          3:   1
INFO - Quant dimension: 0
INFO - Scale[0] = 0.192437
INFO - ZeroPoint[0] = 11
INFO - Model OUTPUT tensors:
INFO -  tensor type is INT8
INFO -  tensor occupies 8 bytes with dimensions
INFO -          0:   1
INFO -          1:   8
INFO - Quant dimension: 0
INFO - Scale[0] = 0.048891
INFO - ZeroPoint[0] = -30
INFO - Activation buffer (a.k.a tensor arena) size used: 275660
INFO - Number of operators: 14
INFO -  Operator 0: CONV_2D
INFO -  Operator 1: DEPTHWISE_CONV_2D
INFO -  Operator 2: CONV_2D
INFO -  Operator 3: DEPTHWISE_CONV_2D
INFO -  Operator 4: CONV_2D
INFO -  Operator 5: DEPTHWISE_CONV_2D
INFO -  Operator 6: CONV_2D
INFO -  Operator 7: DEPTHWISE_CONV_2D
INFO -  Operator 8: CONV_2D
INFO -  Operator 9: DEPTHWISE_CONV_2D
INFO -  Operator 10: CONV_2D
INFO -  Operator 11: AVERAGE_POOL_2D
INFO -  Operator 12: CONV_2D
INFO -  Operator 13: RESHAPE
INFO - Running inference on audio clip 0 => random_id_00_000000.wav
INFO - Inference 1/1
INFO - Average anomaly score is: -0.883147
INFO - Anomaly threshold is: -0.800000
INFO - Everything fine, no anomaly detected!
INFO - Main loop terminated.
INFO - program terminating...
```





```
User input required
Enter option number from:

  1. Classify next ifm
  2. Classify ifm at chosen index
  3. Run classification on all ifm
  4. Show NN model info
  5. List ifm

  Choice: 1█
```

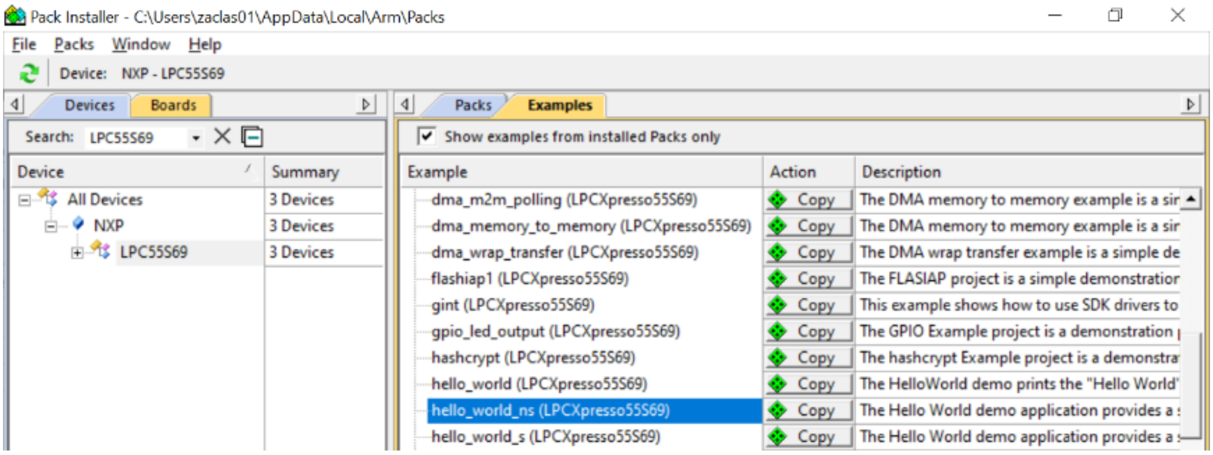


```
INFO - Running inference on image 0 => cat.bmp
INFO - Final results:
INFO - Total number of inferences: 1
INFO - 0) 282 (0.753906) -> tabby, tabby cat
INFO - 1) 286 (0.148438) -> Egyptian cat
INFO - 2) 283 (0.062500) -> tiger cat
INFO - 3) 458 (0.003906) -> bow tie, bow-tie, bowtie
INFO - 4) 288 (0.003906) -> lynx, catamount
INFO - Profile for Inference:
INFO - NPU IDLE: 784 cycles
INFO - NPU AXI0_RD_DATA_BEAT_RECEIVED: 2029001 beats
INFO - NPU AXI0_WR_DATA_BEAT_WRITTEN: 1151315 beats
INFO - NPU AXI1_RD_DATA_BEAT_RECEIVED: 432187 beats
INFO - NPU ACTIVE: 5426281 cycles
INFO - NPU TOTAL: 5427065 cycles
```

Platform	Arm Virtual Hardware – Corstone-300
Software	ML Application
Environment	Amazon Elastic Compute Cloud (EC2) (Amazon Web Services (AWS) account required)
Host OS	Ubuntu Linux
Compiler	Arm Compiler for Embedded
IDE	-

Platform	Arm Virtual Hardware – Corstone-300
Software	ML Application
Environment	Amazon EC2 (AWS account required)
Host OS	Ubuntu Linux
Compiler	Arm Compiler for Embedded
IDE	-

Platform	Arm Virtual Hardware – Corstone-300
Software	ML Application
Environment	Amazon EC2 (AWS account required)
Host OS	Ubuntu Linux
Compiler	Arm Compiler for Embedded
IDE	-



Pack Installer - C:\Users\zaclas01\AppData\Local\Arm\Packs

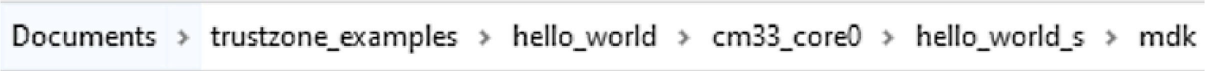
Device: NXP - LPC55S69

Search: LPC55S69

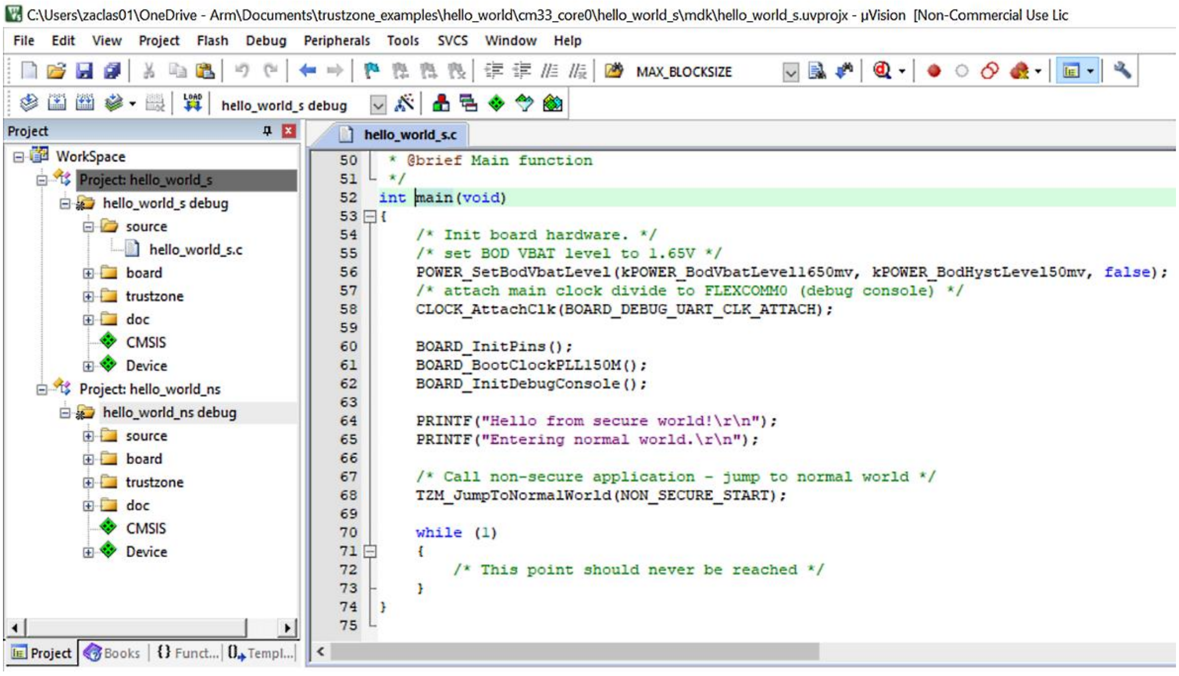
Device	Summary
All Devices	3 Devices
NXP	3 Devices
LPC55S69	3 Devices

☑ Show examples from installed Packs only

Example	Action	Description
dma_m2m_polling (LPCXpresso55S69)	Copy	The DMA memory to memory example is a sir
dma_memory_to_memory (LPCXpresso55S69)	Copy	The DMA memory to memory example is a sir
dma_wrap_transfer (LPCXpresso55S69)	Copy	The DMA wrap transfer example is a simple de
flashiap1 (LPCXpresso55S69)	Copy	The FLASIAP project is a simple demonstratior
gint (LPCXpresso55S69)	Copy	This example shows how to use SDK drivers to
gpio_led_output (LPCXpresso55S69)	Copy	The GPIO Example project is a demonstration
hashcrypt (LPCXpresso55S69)	Copy	The hashcrypt Example project is a demonstra
hello_world (LPCXpresso55S69)	Copy	The HelloWorld demo prints the "Hello World"
hello_world_ns (LPCXpresso55S69)	Copy	The Hello World demo application provides a
hello_world_s (LPCXpresso55S69)	Copy	The Hello World demo application provides a

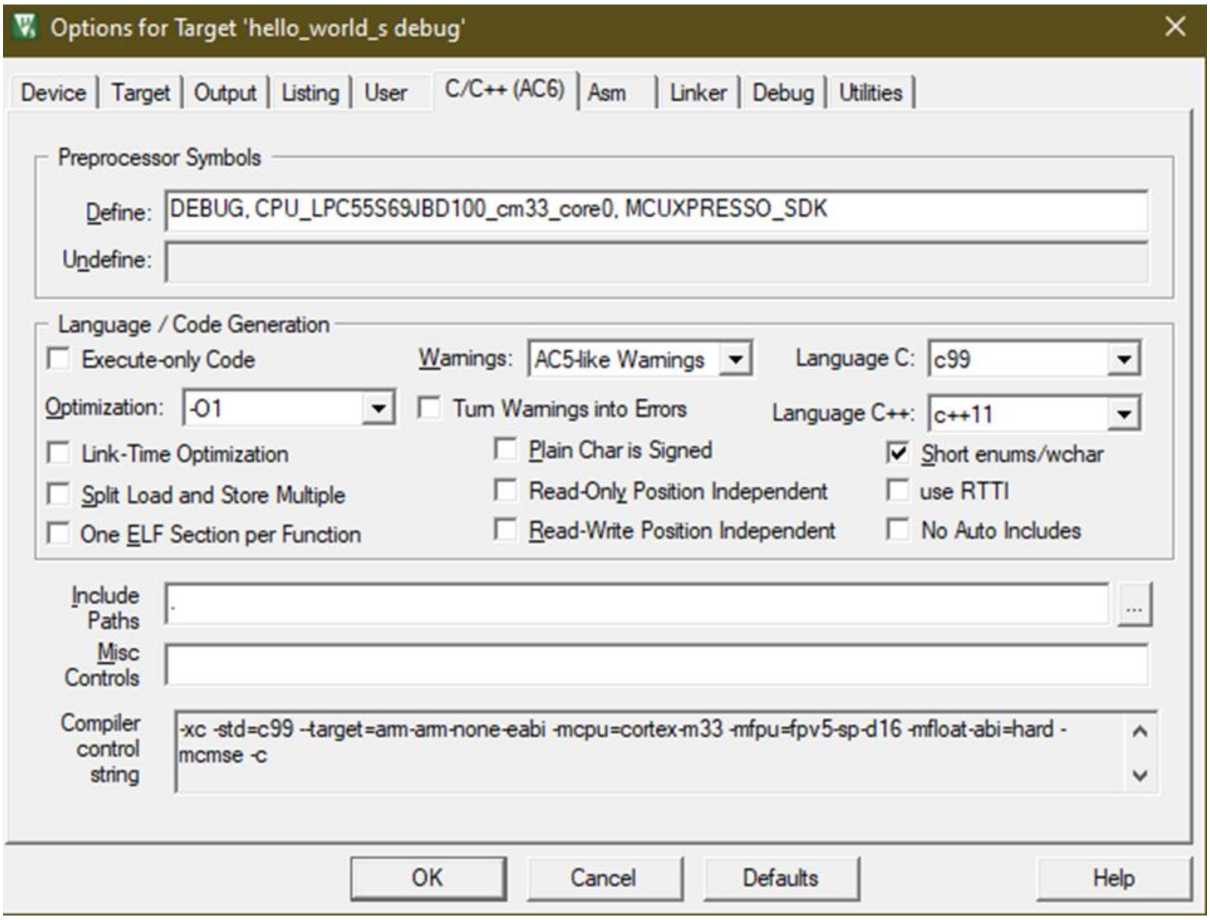


Documents › trustzone_examples › hello_world › cm33_core0 › hello_world_s › mdk



C:\Users\zaclas01\OneDrive - Arm\Documents\trustzone_examples\hello_world\cm33_core0\hello_world_s\mdk\hello_world_s.uvprojx - µVision [Non-Commercial Use Lic

```
 * @brief Main function
 */
int main(void)
{
    /* Init board hardware. */
    /* set BOD VBAT level to 1.65V */
    POWER_SetBodVbatLevel(kPOWER_BodVbatLevel1650mv, kPOWER_BodHystLevel50mv, false);
    /* attach main clock divide to FLEXCOMM0 (debug console) */
    CLOCK_AttachClk(BOARD_DEBUG_UART_CLK_ATTACH);

    BOARD_InitPins();
    BOARD_BootClockPLL150M();
    BOARD_InitDebugConsole();

    PRINTF("Hello from secure world!\r\n");
    PRINTF("Entering normal world.\r\n");

    /* Call non-secure application - jump to normal world */
    TZM_JumpToNormalWorld(NON_SECURE_START);

    while (1)
    {
        /* This point should never be reached */
    }
}
```



Define: DEBUG, CPU_LPC55S69JBD100_cm33_core0, MCUXPRESSO_SDK

Undefine:

□ Execute-only Code

Warnings: AC5-like Warnings

Language C: c99

Optimization: -O1

□ Turn Warnings into Errors

Language C++: c++11

□ Link-Time Optimization

□ Plain Char is Signed

☑ Short enums/wchar

□ Split Load and Store Multiple

□ Read-Only Position Independent

□ use RTTI

□ One ELF Section per Function

□ Read-Write Position Independent

□ No Auto Includes

Include Paths: .

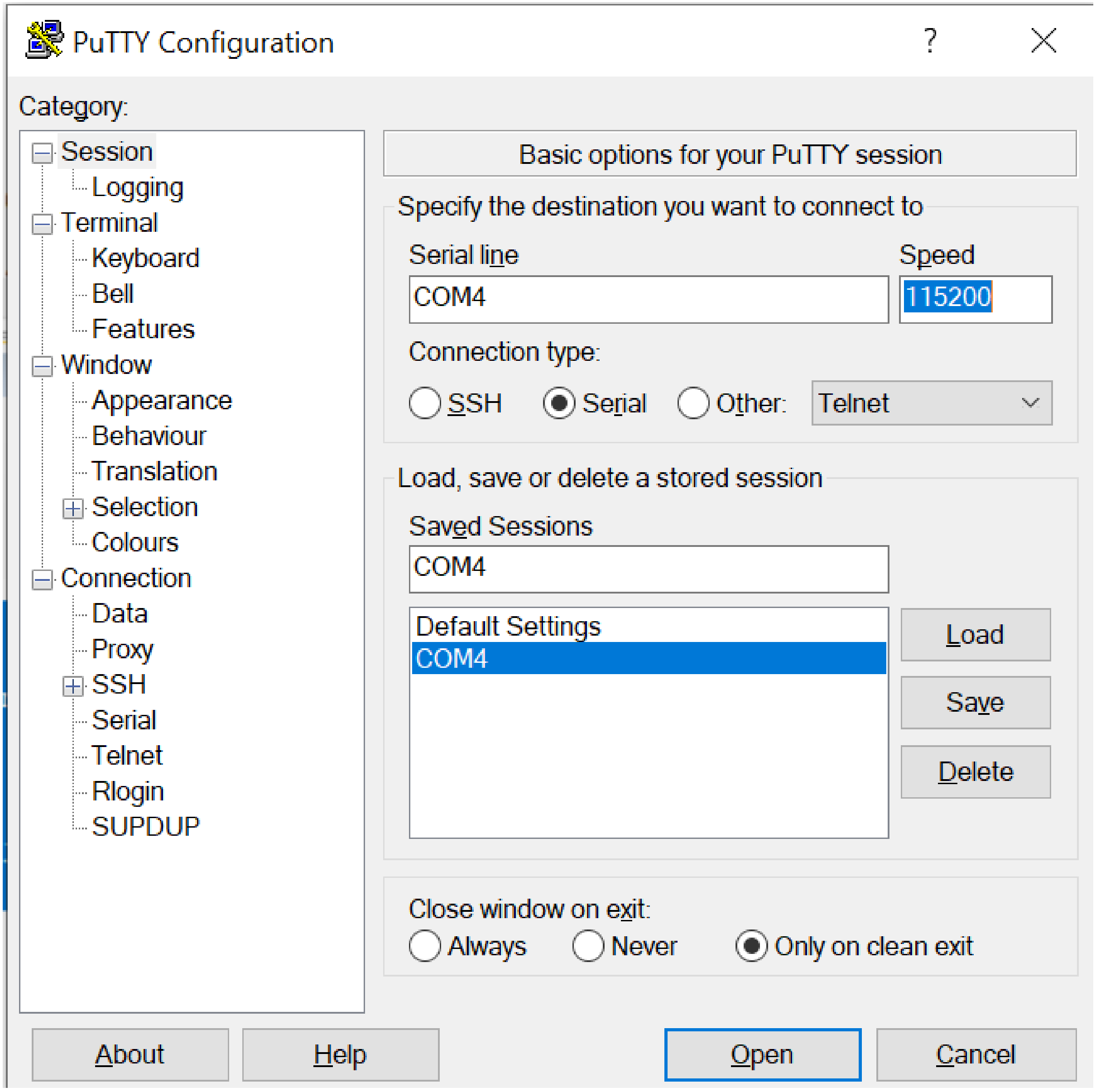
Misc Controls:

Compiler control string:

```
-xc -std=c99 --target=arm-arm-none-eabi -mcpu=cortex-m33 -mfpu=fpv5-sp-d16 -mfloat-abi=hard -
mcmse -c
```

```
 TZM_JumpToNormalWorld
;;; .\tzm_api.c (351)
        0x10000b48:    b580        ..     PUSH     {r7,lr}
        0x10000b4a:    f64e5208    N..R   MOV      r2,#0xed08
        0x10000b4e:    f2ce0202    ....   MOVT     r2,#0xe002
        0x10000b52:    6801        .h     LDR      r1,[r0,#0]
        0x10000b54:    f3818888    ....   MSR      MSP_NS,r1
        0x10000b58:    6010        .`     STR      r0,[r2,#0]
        0x10000b5a:    6840        @h     LDR      r0,[r0,#4]
        0x10000b5c:    e92d0ff0    -...   PUSH     {r4-r11}
        0x10000b60:    f0200001     ...   BIC      r0,r0,#1
        0x10000b64:    b0a2        ..     SUB      sp,sp,#0x88
        0x10000b66:    ec2d0a00    -...   VLSTM    sp
        0x10000b6a:    4601        .F     MOV      r1,r0
        0x10000b6c:    4602        .F     MOV      r2,r0
        0x10000b6e:    4603        .F     MOV      r3,r0
        0x10000b70:    4604        .F     MOV      r4,r0
        0x10000b72:    4605        .F     MOV      r5,r0
        0x10000b74:    4606        .F     MOV      r6,r0
        0x10000b76:    4607        .F     MOV      r7,r0
        0x10000b78:    4680        .F     MOV      r8,r0
        0x10000b7a:    4681        .F     MOV      r9,r0
        0x10000b7c:    4682        .F     MOV      r10,r0
;;; .\tzm_api.c (357)
        0x10000b7e:    4683        .F     MOV      r11,r0
        0x10000b80:    4684        .F     MOV      r12,r0
        0x10000b82:    f3808c00    ....   MSR      APSR_nzcvqg,r0 ;
formerly CPSR_fs
        0x10000b86:    4784        .G     BLXNS    r0
        0x10000b88:    ec3d0a00    =...   VLLDM    sp
        0x10000b8c:    b022        ".     ADD      sp,sp,#0x88
        0x10000b8e:    e8bd0ff0    ....   POP      {r4-r11}
        0x10000b92:    bd80        ..     POP      {r7,pc}
```

```
    DbgConsole_Printf_NSE
        0x1000fe00:    e97fe97f    ....    SG        ; [0x1000fc08]
        0x1000fe04:    f7f0bd42    ..B.    B
 __acle_se_DbgConsole_Printf_NSE ; 0x1000088c
    StringCompare_NSE
        0x1000fe08:    e97fe97f    ....    SG        ; [0x1000fc10]
        0x1000fe0c:    f7f0bdc2    ....    B
 __acle_se_StringCompare_NSE ; 0x10000994
```



Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM4

Speed: 115200

Connection type: ○ SSH ● Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: COM4

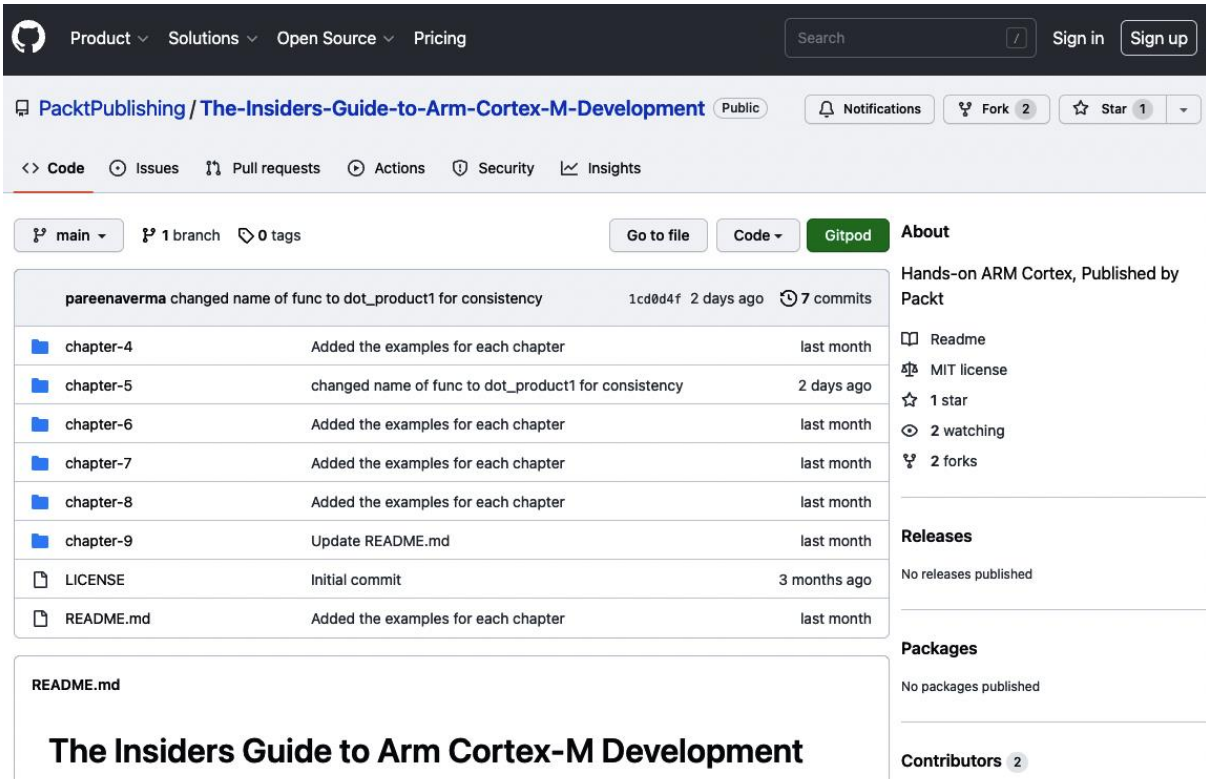
Default Settings
COM4

Close window on exit: ○ Always ○ Never ● Only on clean exit

```
    tfm_psa_connect_veneer
        0x11000678:    e97fe97f    ....    SG        ; [0x11000480]
        0x1100067c:    f008beda    ....    B.W
__acle_se_tfm_psa_connect_veneer ; 0x11009434
```

```
    __acle_se_tfm_psa_connect_veneer
        0x11009434:    ed6dcf81    m...    STCL     p15,c12,[sp,#-0x204]!
        0x11009438:    b500        ..      PUSH     {lr}
        0x1100943a:    f016fe18    ....    BL       psa_connect_cross ;
0x1102006e
        0x1100943e:    f85deb04    ]...    POP      {lr}
        0x11009442:    ec9f0a10    ....    VLDM     pc,{s0-s15} ; ?   ;
[0x11009484] = 0
        0x11009446:    e89f900e    ....    LDM      pc,{r1-r3,r12,pc} ; ?
        0x1100944a:    ecfdcf81    ....    LDCL     p15,c12,[sp],#0x204
        0x1100944e:    4774        tG      BXNS     lr
```

Chapter 8: Streamlining with the Cloud



PacktPublishing / The-Insiders-Guide-to-Arm-Cortex-M-Development

pareenaverma changed name of func to dot_product1 for consistency 1cd0d4f 2 days ago 7 commits

chapter-4	Added the examples for each chapter	last month
chapter-5	changed name of func to dot_product1 for consistency	2 days ago
chapter-6	Added the examples for each chapter	last month
chapter-7	Added the examples for each chapter	last month
chapter-8	Added the examples for each chapter	last month
chapter-9	Update README.md	last month
LICENSE	Initial commit	3 months ago
README.md	Added the examples for each chapter	last month

README.md

The Insiders Guide to Arm Cortex-M Development

About

Hands-on ARM Cortex, Published by Packt

Readme
MIT license
1 star
2 watching
2 forks

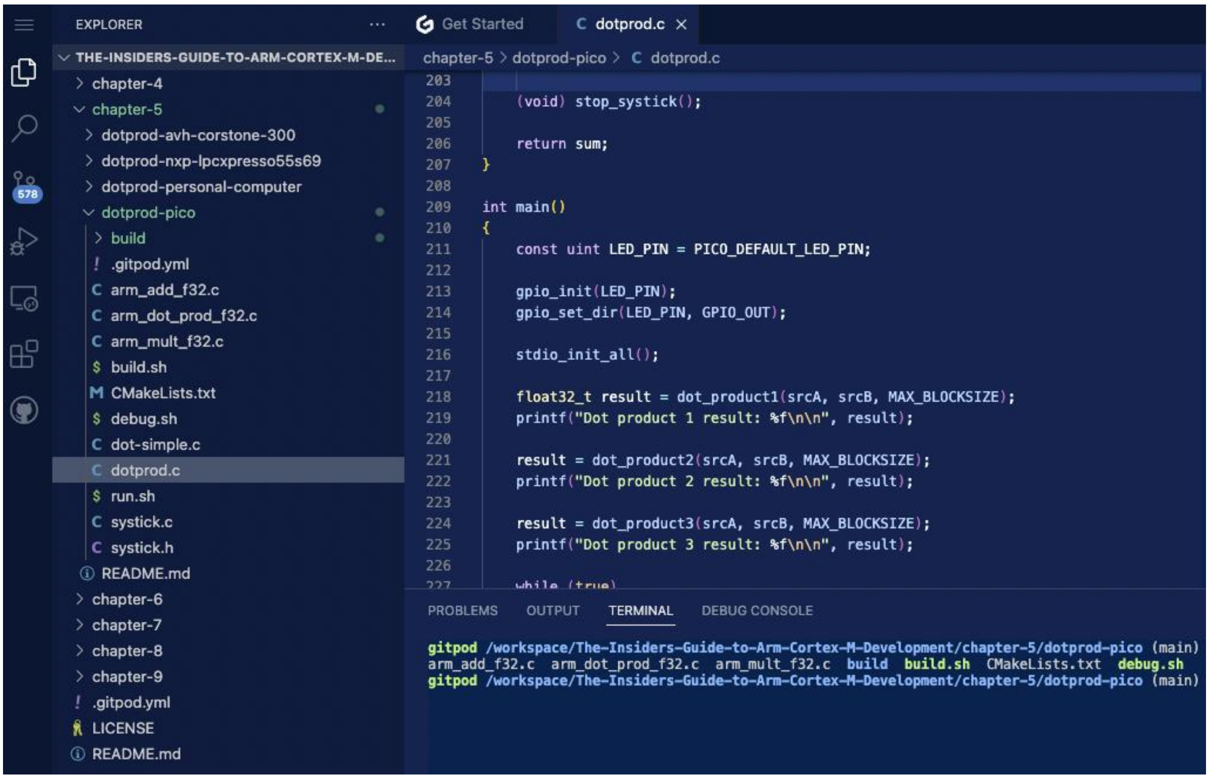
Releases

No releases published

Packages

No packages published

Contributors 2

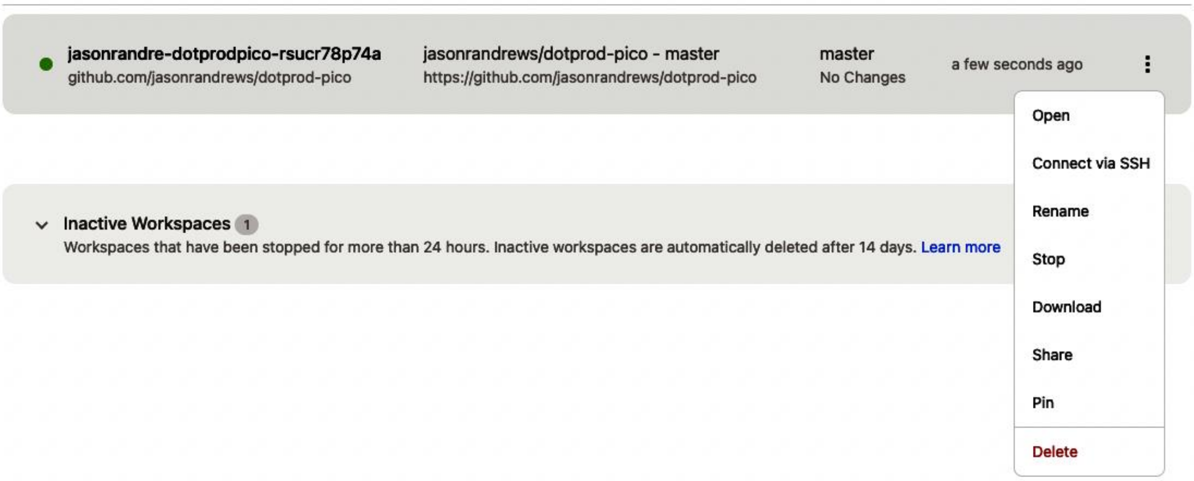


```
THE-INSIDERS-GUIDE-TO-ARM-CORTEX-M-DE...
  > chapter-4
  v chapter-5
    > dotprod-avh-corstone-300
    > dotprod-nxp-lpcxpresso55s69
    > dotprod-personal-computer
    v dotprod-pico
      > build
      ! .gitpod.yml
      C arm_add_f32.c
      C arm_dot_prod_f32.c
      C arm_mult_f32.c
      $ build.sh
      M CMakeLists.txt
      $ debug.sh
      C dot-simple.c
      C dotprod.c
      $ run.sh
      C systick.c
      C systick.h
    README.md
  > chapter-6
  > chapter-7
  > chapter-8
  > chapter-9
  ! .gitpod.yml
  LICENSE
  README.md
```

chapter-5 > dotprod-pico > C dotprod.c

```c

    (void) stop_systick();

    return sum;
}

int main()
{
    const uint LED_PIN = PICO_DEFAULT_LED_PIN;

    gpio_init(LED_PIN);
    gpio_set_dir(LED_PIN, GPIO_OUT);

    stdio_init_all();

    float32_t result = dot_product1(srcA, srcB, MAX_BLOCKSIZE);
    printf("Dot product 1 result: %f\n\n", result);

    result = dot_product2(srcA, srcB, MAX_BLOCKSIZE);
    printf("Dot product 2 result: %f\n\n", result);

    result = dot_product3(srcA, srcB, MAX_BLOCKSIZE);
    printf("Dot product 3 result: %f\n\n", result);

    while (true)
```

```
gitpod /workspace/The-Insiders-Guide-to-Arm-Cortex-M-Development/chapter-5/dotprod-pico (main)
arm_add_f32.c  arm_dot_prod_f32.c  arm_mult_f32.c  build  build.sh  CMakeLists.txt  debug.sh
gitpod /workspace/The-Insiders-Guide-to-Arm-Cortex-M-Development/chapter-5/dotprod-pico (main)
```



jasonrandre-dotprodpico-rsucr78p74a
github.com/jasonrandrews/dotprod-pico

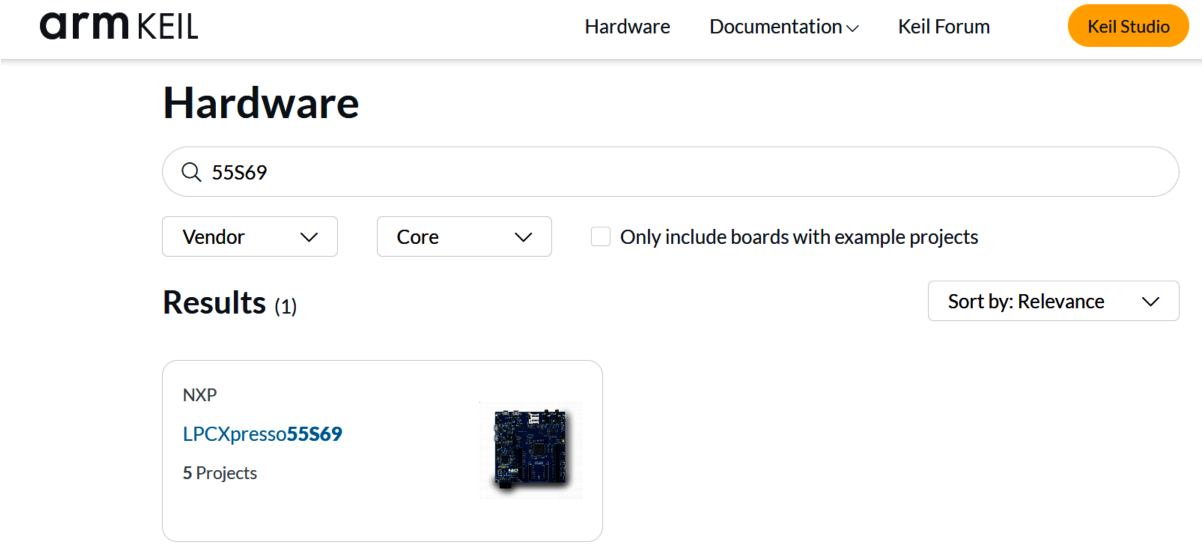
jasonrandrews/dotprod-pico - master
https://github.com/jasonrandrews/dotprod-pico

master
No Changes

a few seconds ago

Inactive Workspaces 1

Workspaces that have been stopped for more than 24 hours. Inactive workspaces are automatically deleted after 14 days. Learn more



Hardware

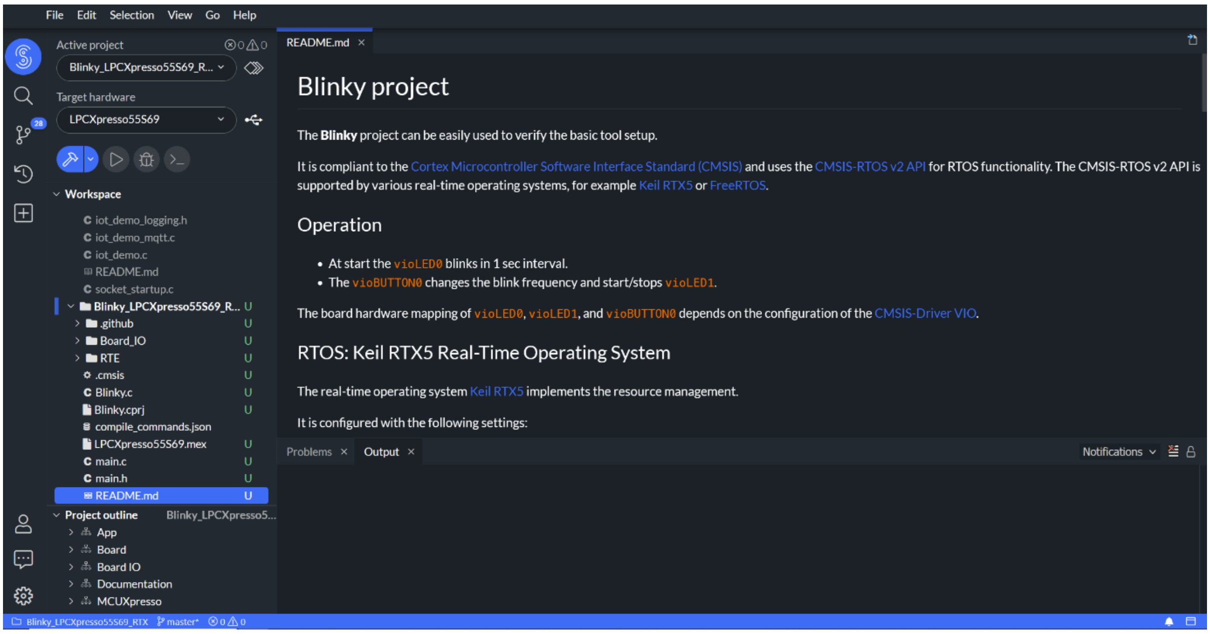
55S69

□ Only include boards with example projects

Results (1)

Sort by: Relevance

NXP
LPCXpresso55S69
5 Projects



Blinky project

The **Blinky** project can be easily used to verify the basic tool setup.

It is compliant to the Cortex Microcontroller Software Interface Standard (CMSIS) and uses the CMSIS-RTOS v2 API for RTOS functionality. The CMSIS-RTOS v2 API is supported by various real-time operating systems, for example Keil RTX5 or FreeRTOS.

Operation

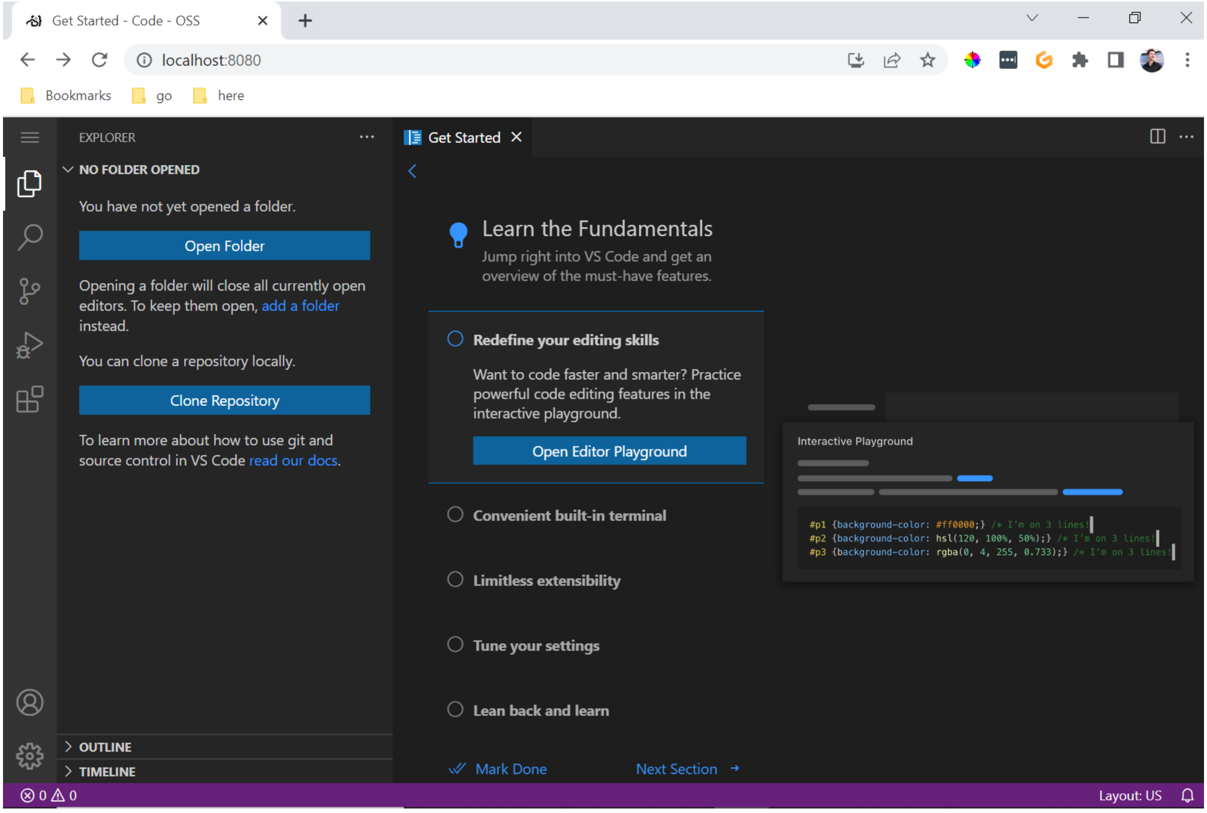
- At start the `vioLED0` blinks in 1 sec interval.
- The `vioBUTTON0` changes the blink frequency and start/stops `vioLED1`.

The board hardware mapping of `vioLED0`, `vioLED1`, and `vioBUTTON0` depends on the configuration of the CMSIS-Driver VIO.

RTOS: Keil RTX5 Real-Time Operating System

The real-time operating system Keil RTX5 implements the resource management.

It is configured with the following settings:



NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

To learn more about how to use git and source control in VS Code read our docs.

Learn the Fundamentals

Jump right into VS Code and get an overview of the must-have features.

○ Redefine your editing skills

Want to code faster and smarter? Practice powerful code editing features in the interactive playground.

Open Editor Playground

○ Convenient built-in terminal

○ Limitless extensibility

○ Tune your settings

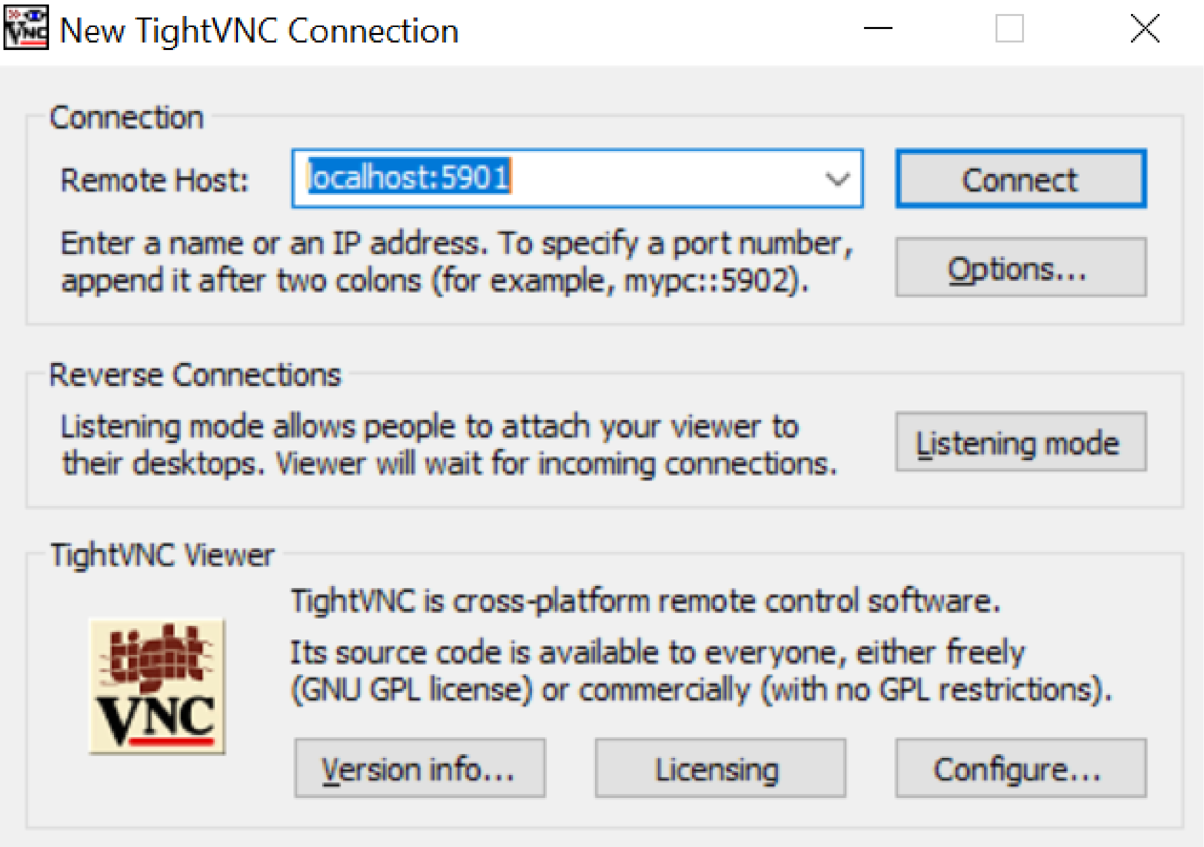
○ Lean back and learn

✓ Mark Done

Next Section →

Interactive Playground

```
#p1 {background-color: #ff0000;} /* I'm on 3 lines!
#p2 {background-color: hsl(120, 100%, 50%);} /* I'm on 3 lines!
#p3 {background-color: rgba(0, 4, 255, 0.733);} /* I'm on 3 lines!
```



New TightVNC Connection

Connection

Remote Host: localhost:5901

Connect

Enter a name or an IP address. To specify a port number, append it after two colons (for example, mypc::5902).

Options...

Reverse Connections

Listening mode allows people to attach your viewer to their desktops. Viewer will wait for incoming connections.

Listening mode

TightVNC Viewer

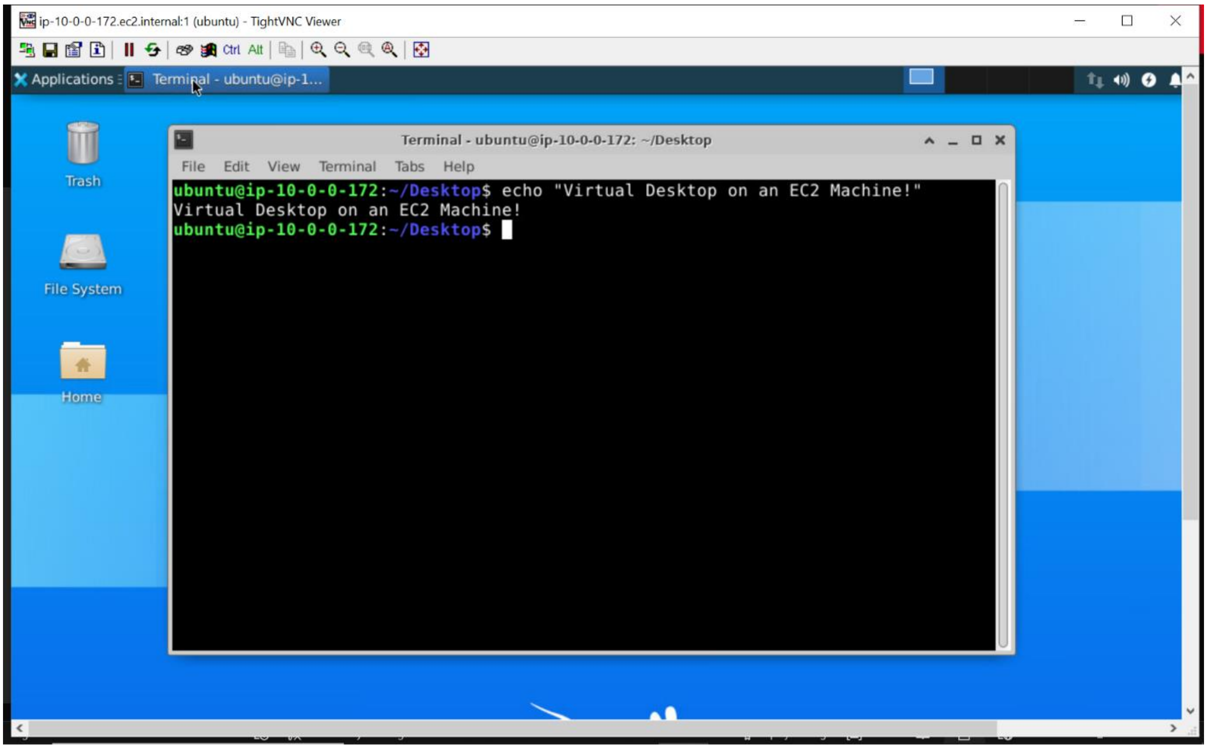
TightVNC is cross-platform remote control software.

Its source code is available to everyone, either freely (GNU GPL license) or commercially (with no GPL restrictions).

Version info...

Licensing

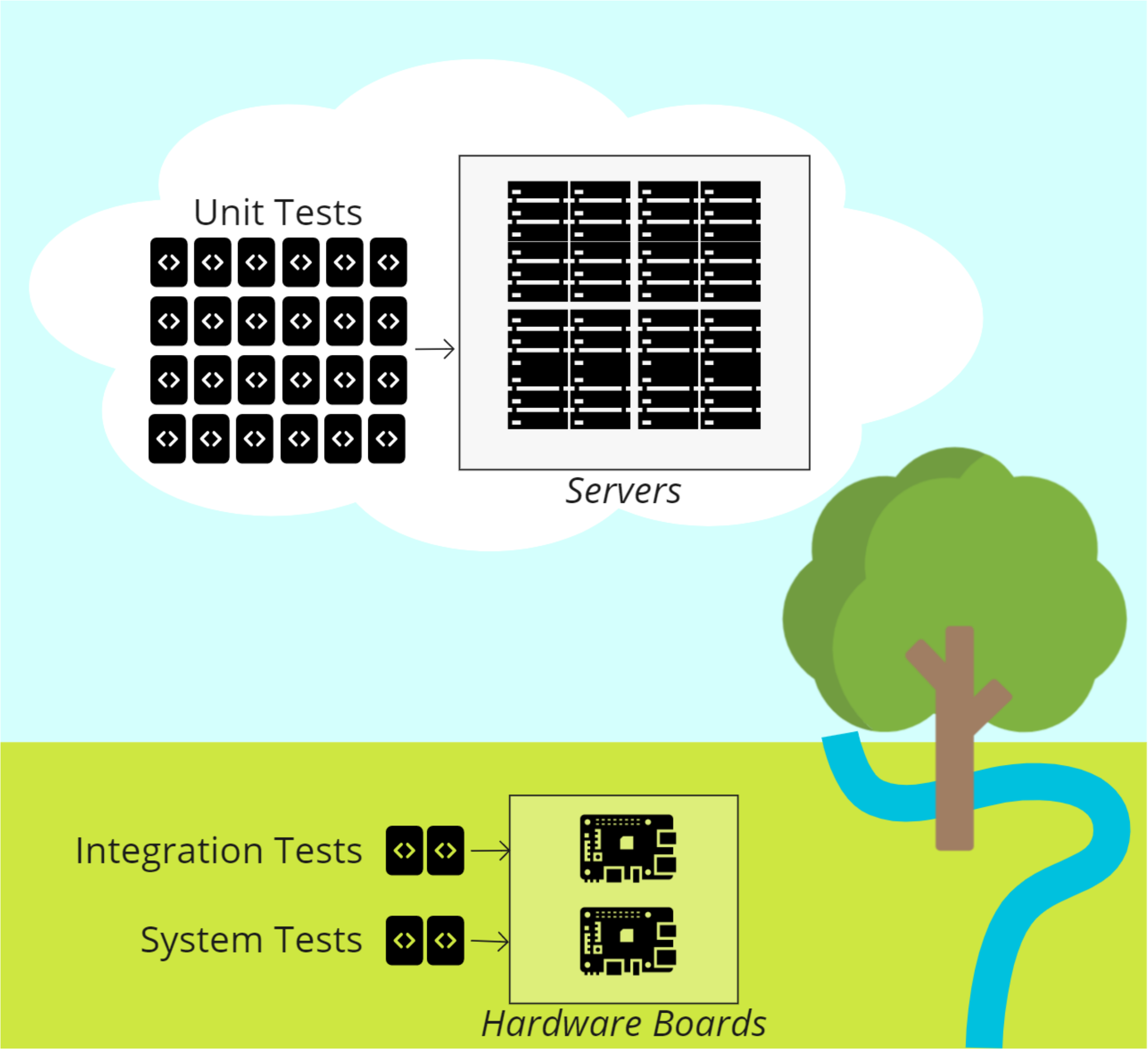
Configure...

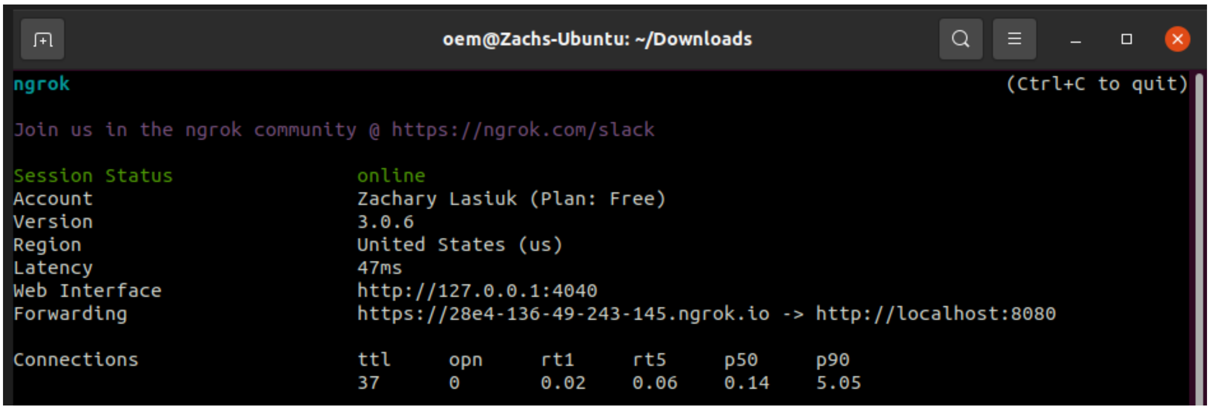


ip-10-0-0-172.ec2.internal:1 (ubuntu) - TightVNC Viewer
Ctrl
Alt
Applications
Terminal - ubuntu@ip-1...
Trash
File System
Home
Terminal - ubuntu@ip-10-0-0-172: ~/Desktop
File
Edit
View
Terminal
Tabs
Help

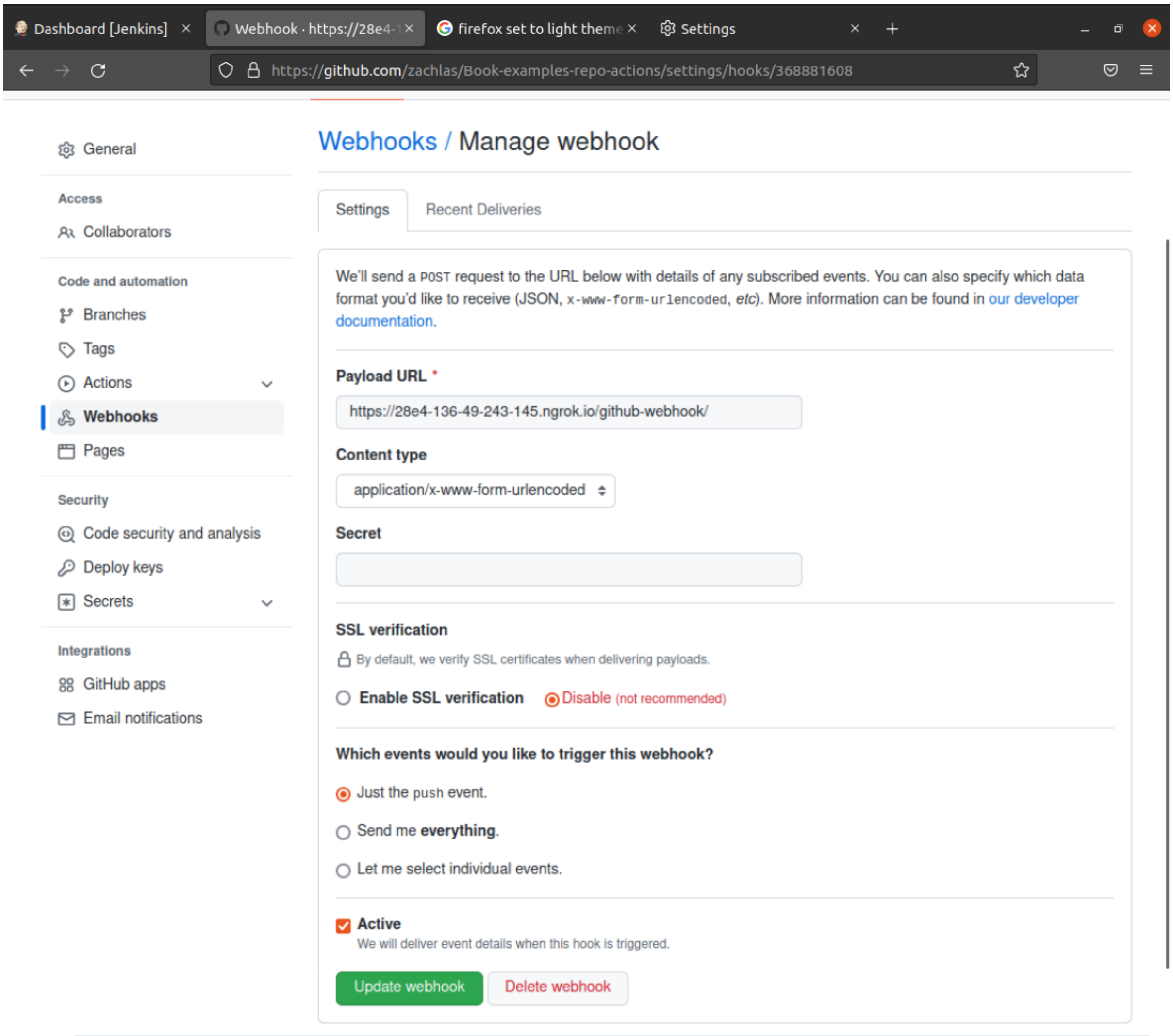
```
ubuntu@ip-10-0-0-172:~/Desktop$ echo "Virtual Desktop on an EC2 Machine!"
Virtual Desktop on an EC2 Machine!
ubuntu@ip-10-0-0-172:~/Desktop$
```

Chapter 9: Implementing Continuous Integration





```
ngrok                                                                            (Ctrl+C to quit)

Join us in the ngrok community @ https://ngrok.com/slack

Session Status                online
Account                       Zachary Lasiuk (Plan: Free)
Version                       3.0.6
Region                        United States (us)
Latency                       47ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://28e4-136-49-243-145.ngrok.io -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              37      0       0.02    0.06    0.14    5.05
```



Webhooks / Manage webhook

Settings | Recent Deliveries

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *

https://28e4-136-49-243-145.ngrok.io/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

SSL verification

By default, we verify SSL certificates when delivering payloads.

○ Enable SSL verification ● Disable (not recommended)

Which events would you like to trigger this webhook?

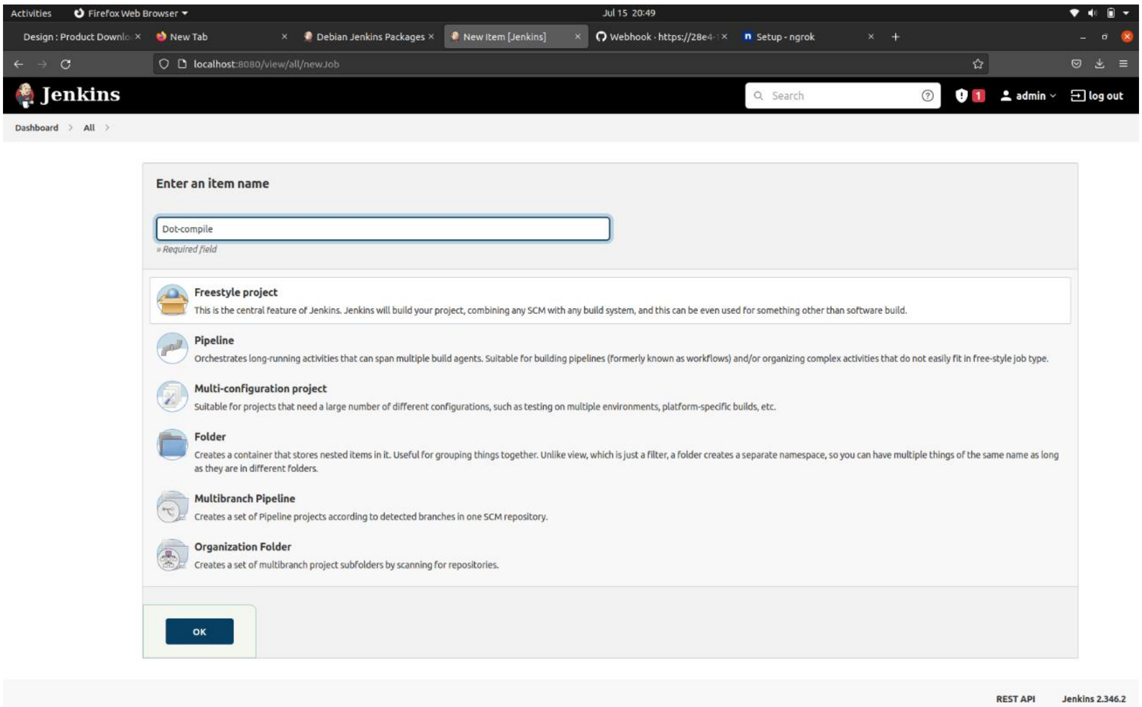
● Just the `push` event.

○ Send me **everything**.

○ Let me select individual events.

☑ **Active**

We will deliver event details when this hook is triggered.



Enter an item name

Dot-compile

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder

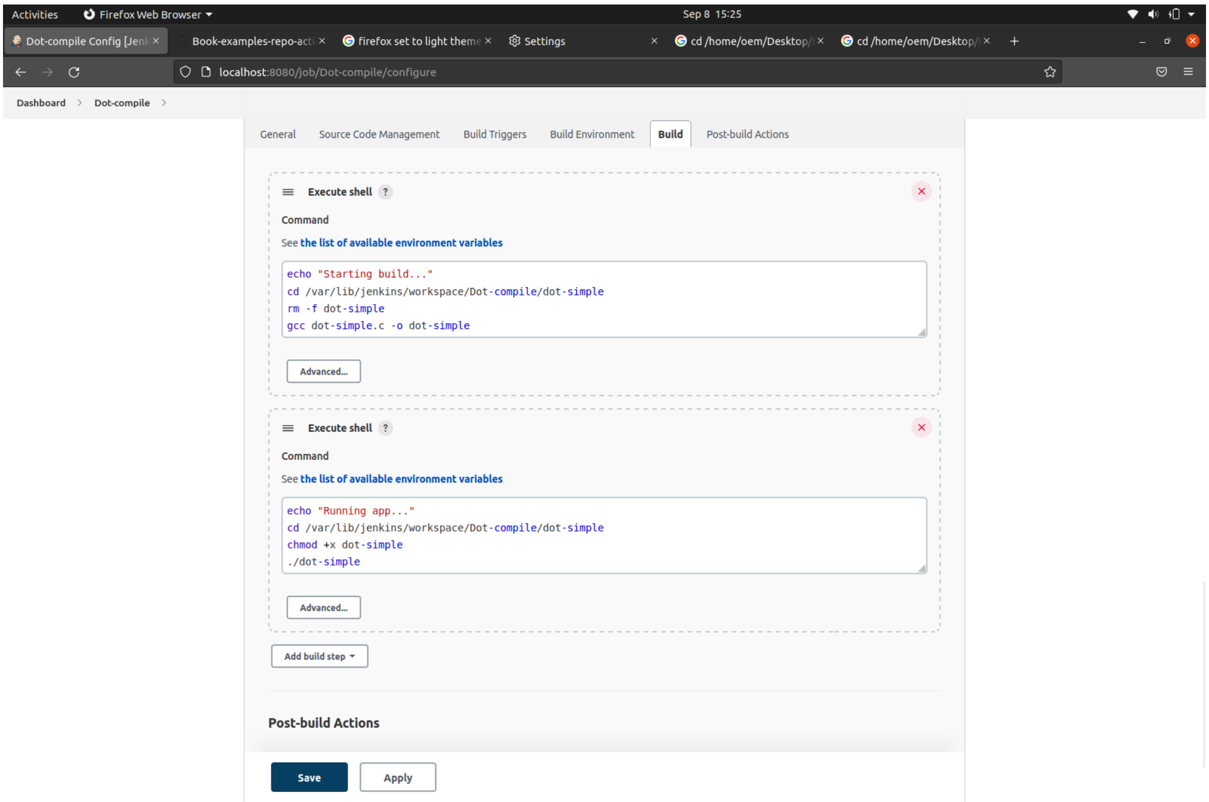
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline

Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder

Creates a set of multibranch project subfolders by scanning for repositories.



Execute shell

Command

See the list of available environment variables

```
echo "Starting build..."
cd /var/lib/jenkins/workspace/Dot-compile/dot-simple
rm -f dot-simple
gcc dot-simple.c -o dot-simple
```

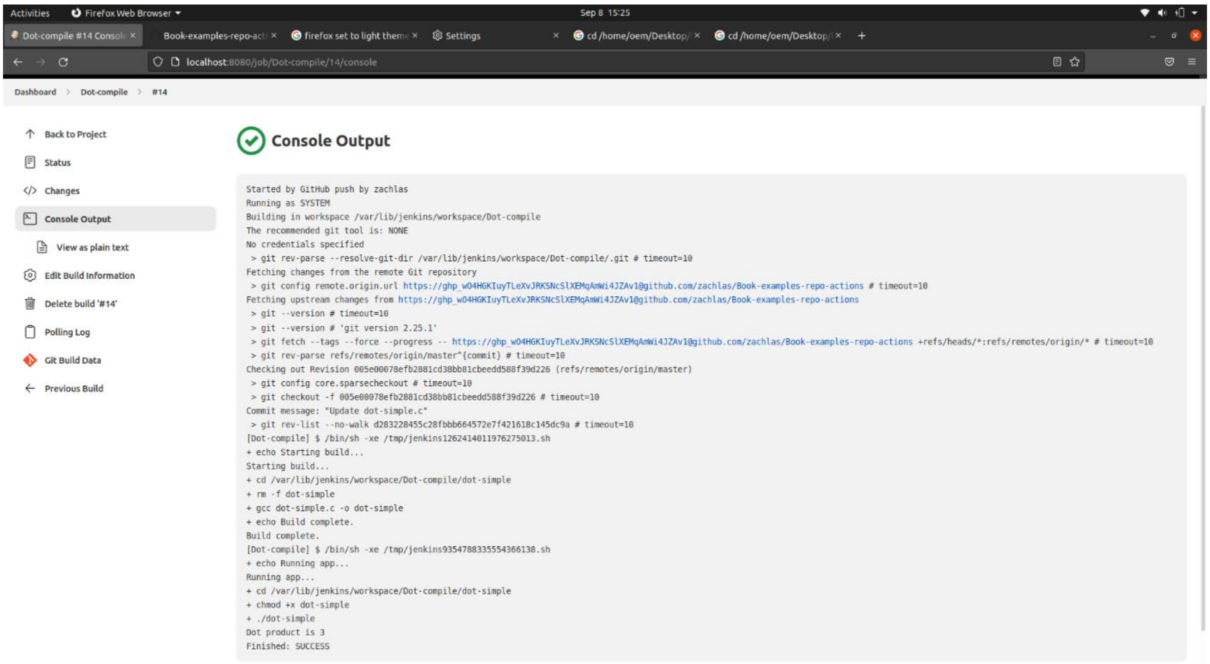
Execute shell

Command

See the list of available environment variables

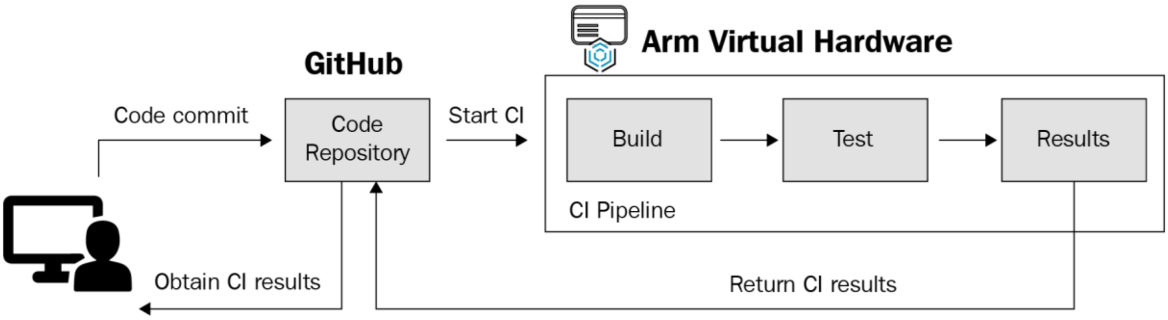
```
echo "Running app..."
cd /var/lib/jenkins/workspace/Dot-compile/dot-simple
chmod +x dot-simple
./dot-simple
```

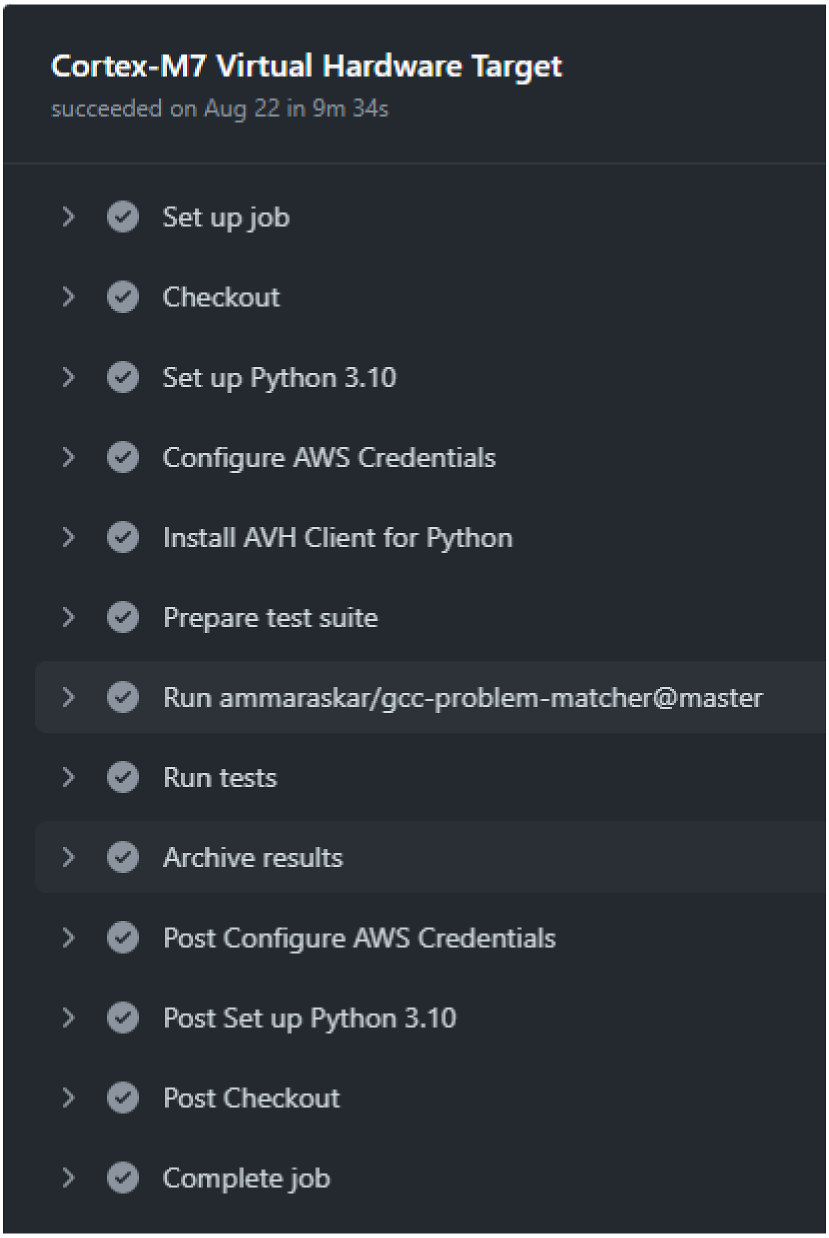
Post-build Actions



Console Output

```
Started by GitHub push by zachlas
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Dot-compile
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Dot-compile/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://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@github.com/zachlas/Book-examples-repo-actions # timeout=10
Fetching upstream changes from https://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@github.com/zachlas/Book-examples-repo-actions
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://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@github.com/zachlas/Book-examples-repo-actions +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 005e00078efb2881cd38bb81cbeedd588f39d226 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 005e00078efb2881cd38bb81cbeedd588f39d226 # timeout=10
Commit message: "Update dot-simple.c"
 > git rev-list --no-walk d283228455c28fbbb664572e7f421618c145dc9a # timeout=10
[Dot-compile] $ /bin/sh -xe /tmp/jenkins1262414011976275013.sh
+ echo Starting build...
Starting build...
+ cd /var/lib/jenkins/workspace/Dot-compile/dot-simple
+ rm -f dot-simple
+ gcc dot-simple.c -o dot-simple
+ echo Build complete.
Build complete.
[Dot-compile] $ /bin/sh -xe /tmp/jenkins9354788335554366138.sh
+ echo Running app...
Running app...
+ cd /var/lib/jenkins/workspace/Dot-compile/dot-simple
+ chmod +x dot-simple
+ ./dot-simple
Dot product is 3
Finished: SUCCESS
```



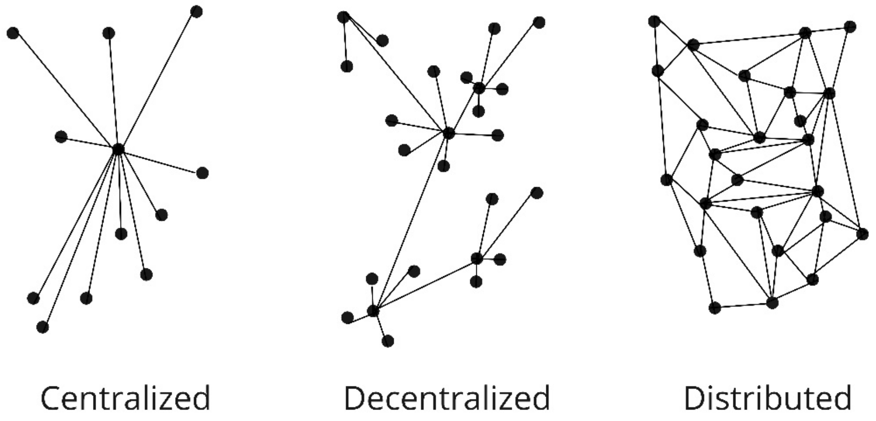


Cortex-M7 Virtual Hardware Target

succeeded on Aug 22 in 9m 34s

- ✓ Set up job
- ✓ Checkout
- ✓ Set up Python 3.10
- ✓ Configure AWS Credentials
- ✓ Install AVH Client for Python
- ✓ Prepare test suite
- ✓ Run ammaraskar/gcc-problem-matcher@master
- ✓ Run tests
- ✓ Archive results
- ✓ Post Configure AWS Credentials
- ✓ Post Set up Python 3.10
- ✓ Post Checkout
- ✓ Complete job

Chapter 10: Looking Ahead


Centralized
Decentralized
Distributed