Arm Virtual Hardware Examples

Version 1.0

Knowledge Base Article

Confidential

Copyright © 2022 Arm Limited (or its affiliates).
All rights reserved.

Issue 01
107668_0100_01_en

Arm Virtual Hardware Examples

Knowledge Base Article

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	21 September 2022	Non-Confidential	First release

Proprietary Notice

Confidentiality Status

Product Status

The information in this document is Final, that is for a developed product.

Feedback

Arm® welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Capture Input from the Console....4
1.1 Retrieving Character Input from the User....4
1.2 Detecting When the User Presses Return....4
1.3 Modify How the Terminal Handles “newline”....4

2. iMX8m Arm Cortex Complex Examples....5
2.1 Blink the Led on the iMX8m Arm Cortex Complex device....5
2.1.1 Blink the LED....5

3. STM32U5 IoT Discovery Kit Examples....7
3.1 STM32CubeU5 TFM example....7
3.1.3 Next steps....7
3.2 STM32CubeU5 WiFiBasics example....7

4. Raspberry Pi 4 Example....14
4.1 Raspberry Pi vMMIO example....14
4.1.4 Flow Diagram....14
4.1.5 Summary....15

5. API Examples....16
5.1 Python and Javascript API Examples....16
5.2 GitHub Actions Example - Building Firmware, Uploading, and Testing....16

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

1. Capture Input from the Console

Learn how to capture character and newline console input.

1.1 Retrieving Character Input from the User

It is possible to capture character input from the console using the scanf C library function.

For example, capture a character input using

```
scanf("%c", &input_char);
```

1.2 Detecting When the User Presses Return

Our UI console sends a new line (`\n`) when you hit return but doesn't send a carriage return (`\r`). When detecting a new line in code, use `\n` rather than `\r`.

For example the following will run, whereas using `\r` will hang:

```
while ( getchar() != '\n');
```

1.3 Modify How the Terminal Handles "newline"

Alternatively, you can adjust the terminal settings to not map CR to NL using the command `stty -icrnl`.

You can then send a `\r` by pressing Ctrl+M or `\n` by pressing Ctrl+J.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

2. iMX8m Arm Cortex Complex Examples

iMX8m Arm Cortex Complex examples:

- Blink the Led on the iMX8m Arm Cortex Complex device

2.1 Blink the Led on the iMX8m Arm Cortex Complex device

This tutorial will demonstrate how to blink the LED in the iMX8m Arm Cortex Complex device. It follows from the Quickstart for iMX8m Arm Cortex Complex tutorial.

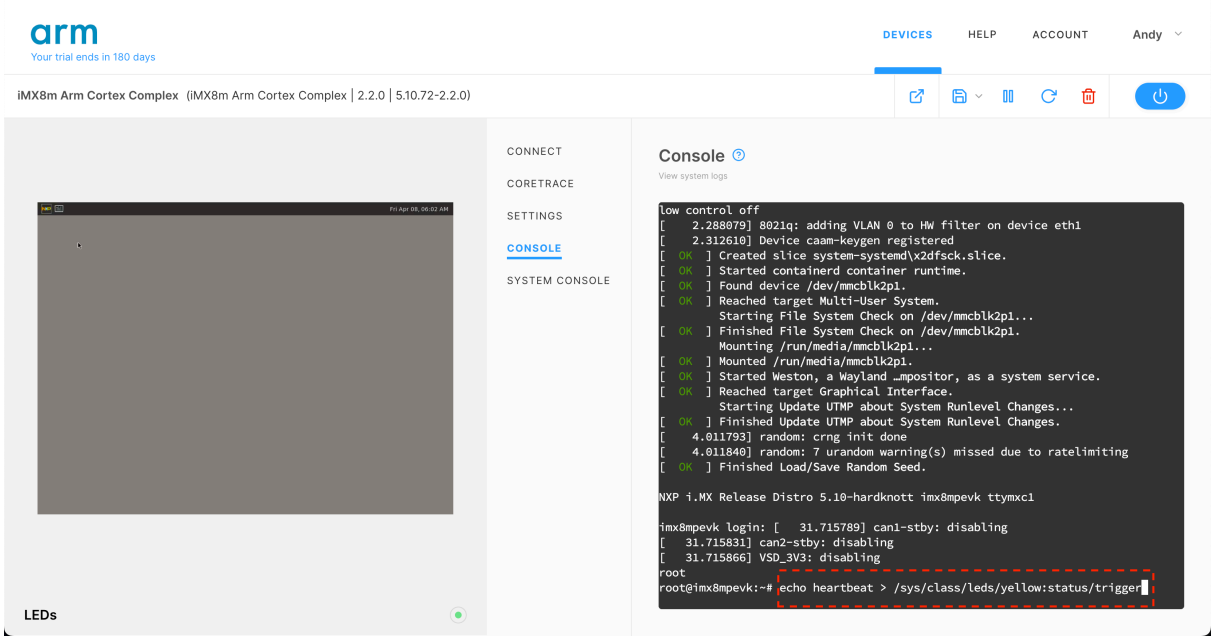
2.1.1 Blink the LED

To blink the LED, do the following:

1. Following on from the tutorial OR go to a iMX8m Arm Cortex Complex device that has the console logged in.
2. To blink the LED type `echo heartbeat > /sys/class/leds/yellow:status/trigger` into the console.

   The LED representation is at the bottom of the desktop. In the current state it is on and solid.

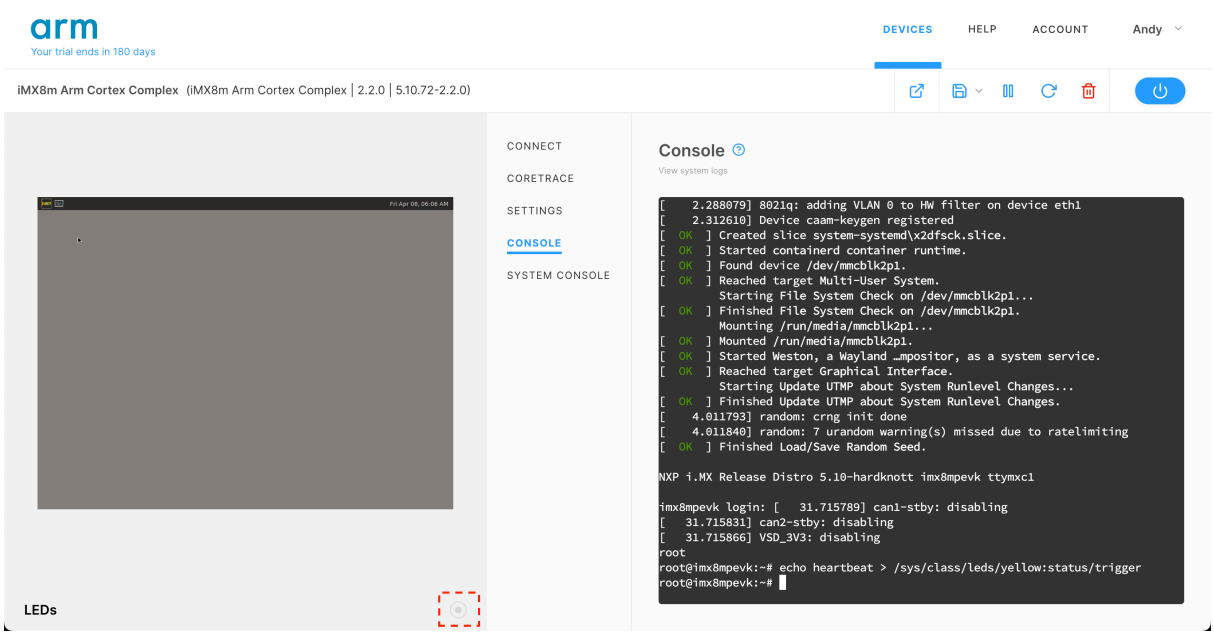
   Figure 2-1: LED on and solid



3. The LED will now be blinking.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 2-2: LED blinking



Find out more

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

3. STM32U5 IoT Discovery Kit Examples

STM32U5 IoT Discovery Kit examples:

- STM32CubeU5 TFM example
- STM32CubeU5 WiFiBasics example

3.1 STM32CubeU5 TFM example

STM32U5 Trusted Firmware M tests firmware package example.

For this example the kernel component in the firmware package is itself a ZIP archive, containing various binaries and `load.txt` file describing their location in memory:

```
name:tfm_s_init.bin load:0xc024000
name:tfm_ns_init.bin load:0xc052000
name:loader.bin load:0xc1fa000
name:TFM_SBSFU_Boot.bin load:0xc002000
```

3.1.3 Next steps

Other STM32U5 IoT Discovery Kit examples;

- STM32CubeU5 WiFiBasics example
- STM32CubeU5 IOT_HTTP_WebServer (This is the STM32U5 IoT Discovery Kit Quickstart guide)

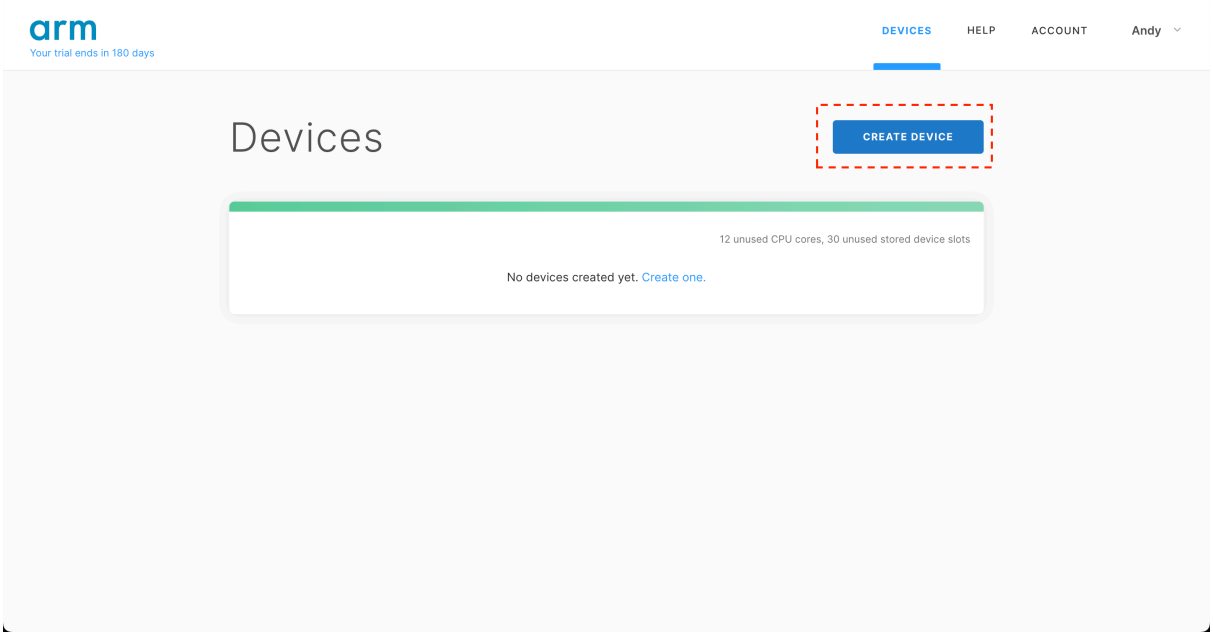
3.2 STM32CubeU5 WiFiBasics example

This example will test basic WiFi networking functions:

1. Log in to AVH with your Arm user account. If you do not have or need to create an Arm account, register at https://developer.arm.com/.
2. When logged in you land on the AVH Devices page. Click CREATE DEVICE.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-1: Create device



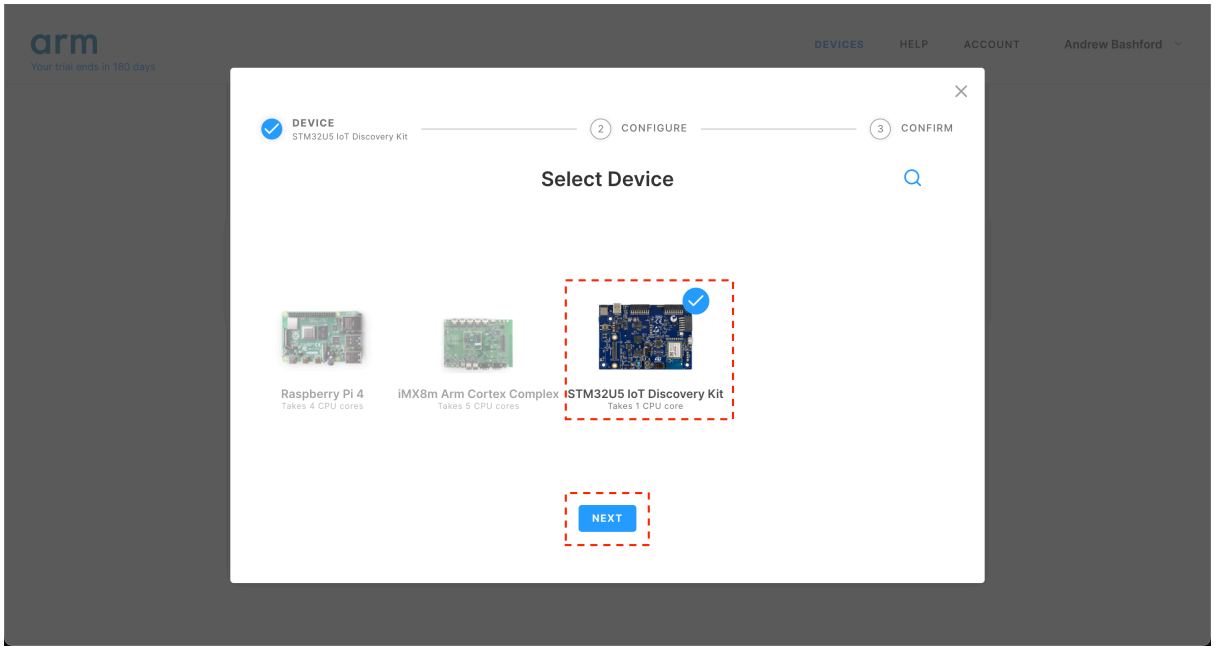
3. Click on the STM32U5 IoT Discovery Kit to select it. Then click NEXT.

Note

You require 1 free CPU core to build this device or it will be disabled.

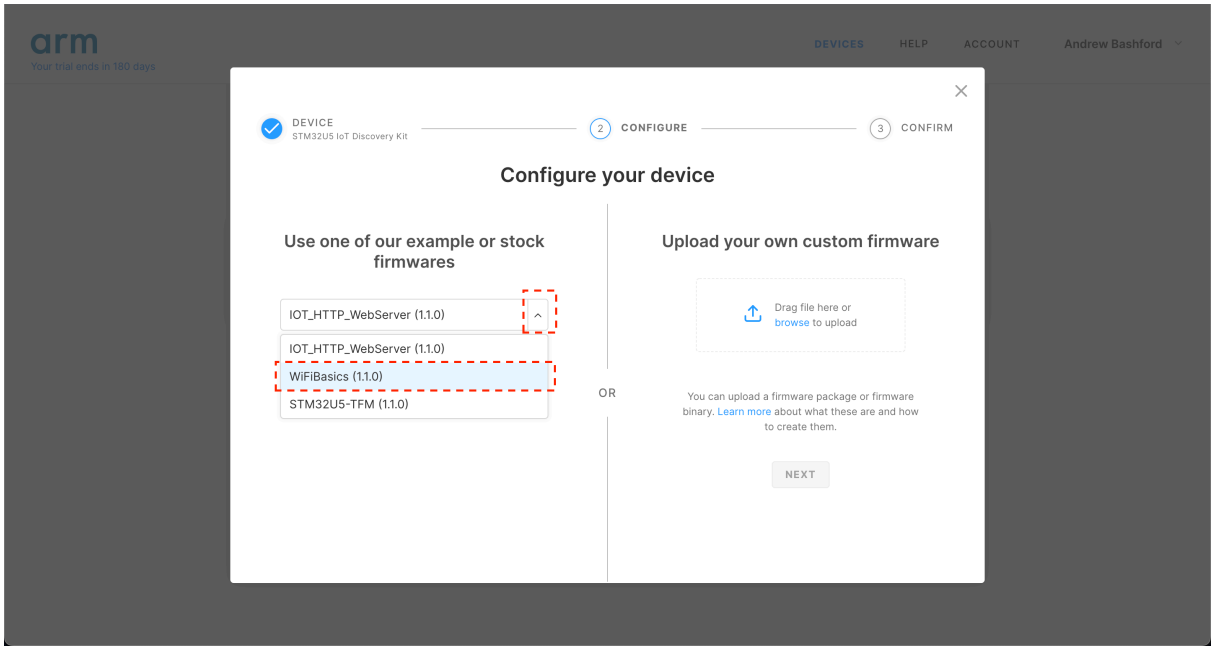
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-2: Select device



4. In this next step, you will see a list of pre-configured examples/stock firmware or you can upload your own custom firmware. In this example, we will use the selection `WiFiBasics (1.1.0)` in the stock firmware section.

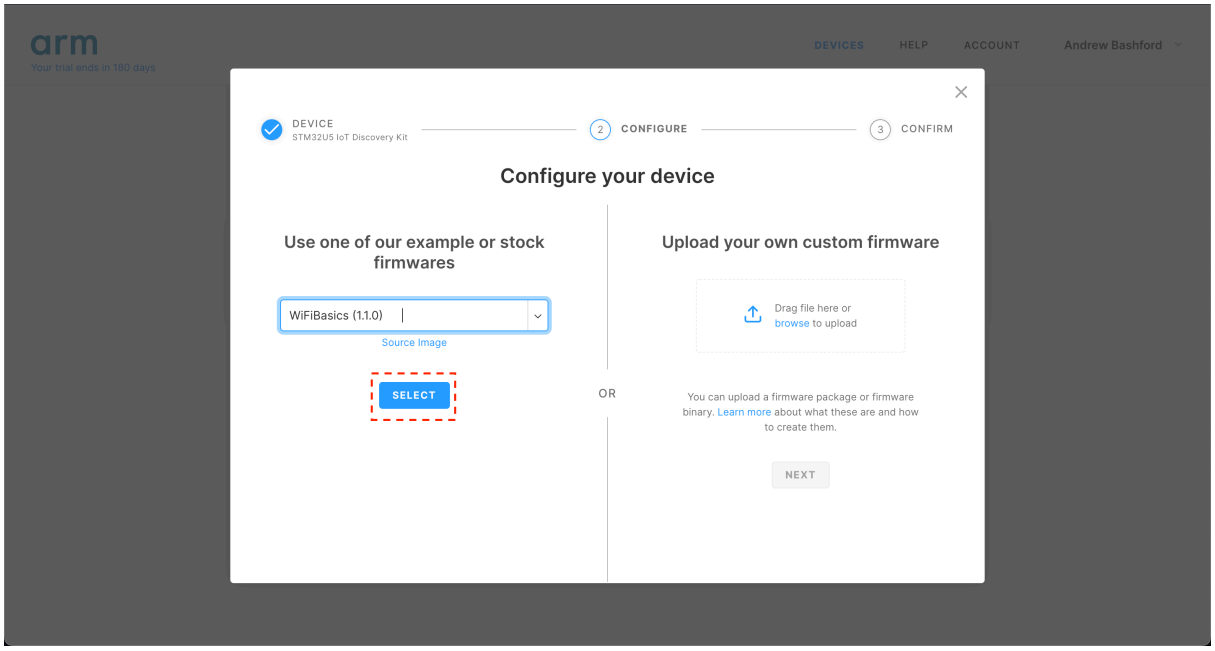
Figure 3-3: Configure your device



5. Click SELECT.

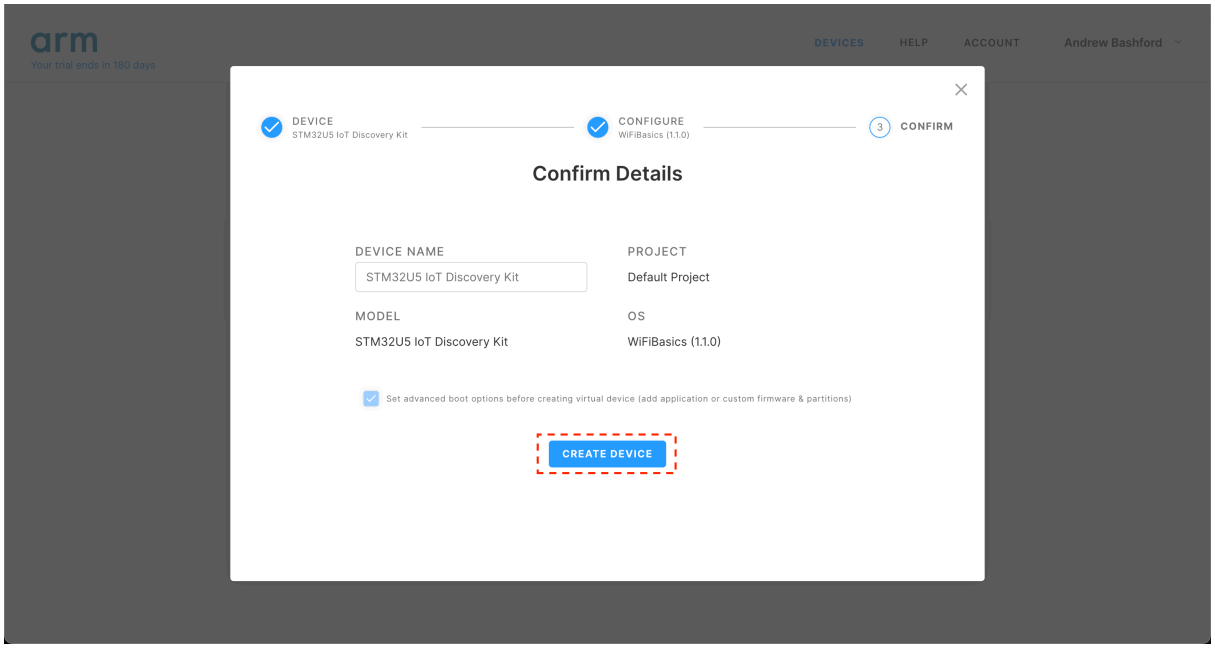
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-4: Configure your device



6. In the confirmation screen, click CREATE DEVICE using the default name and the advance boot options will be checked by default.

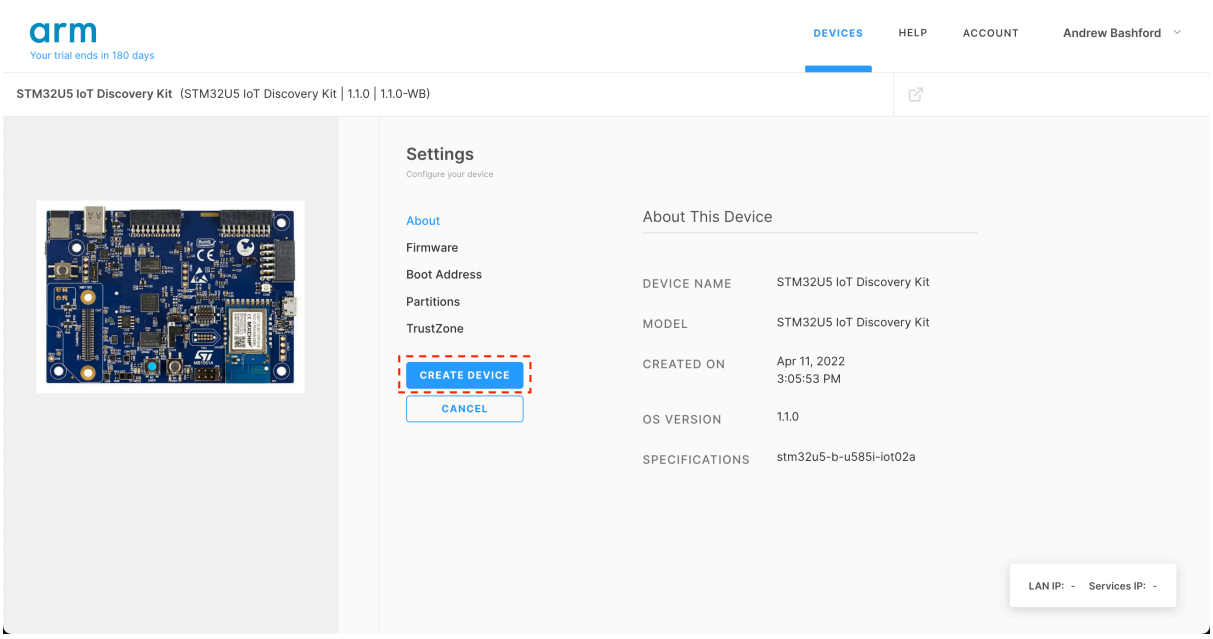
Figure 3-5: Confirm Details



7. In the advanced boot options, we will just use the default. Click CREATE DEVICE.

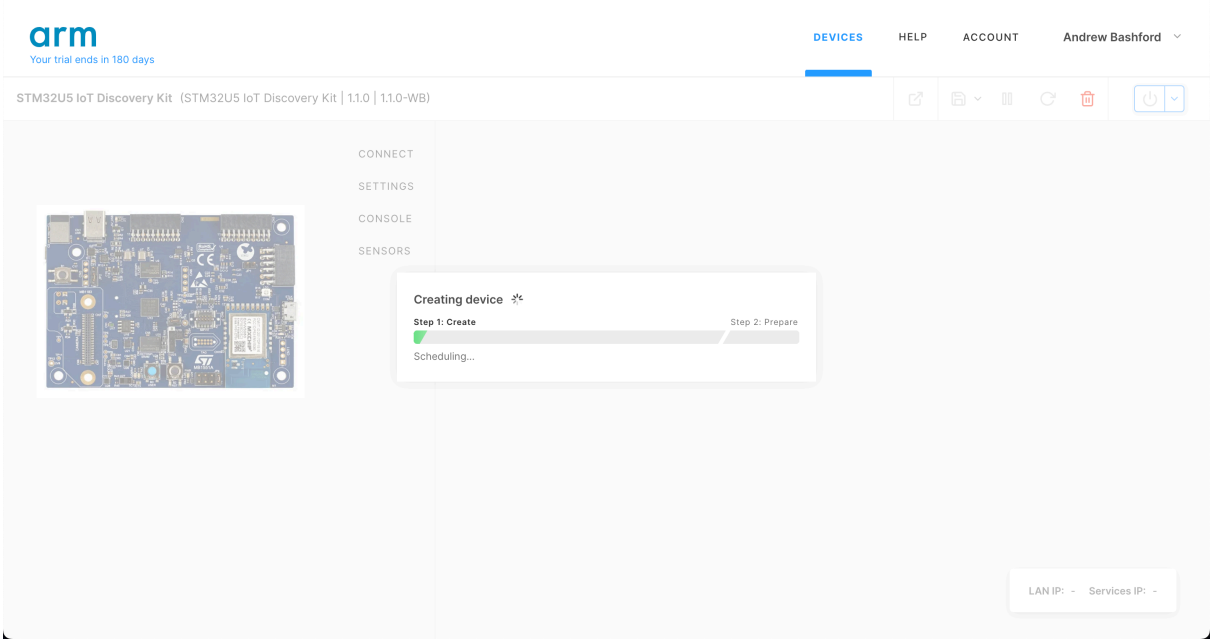
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-6: Settings



8. The device will then be created. It takes a few moments...

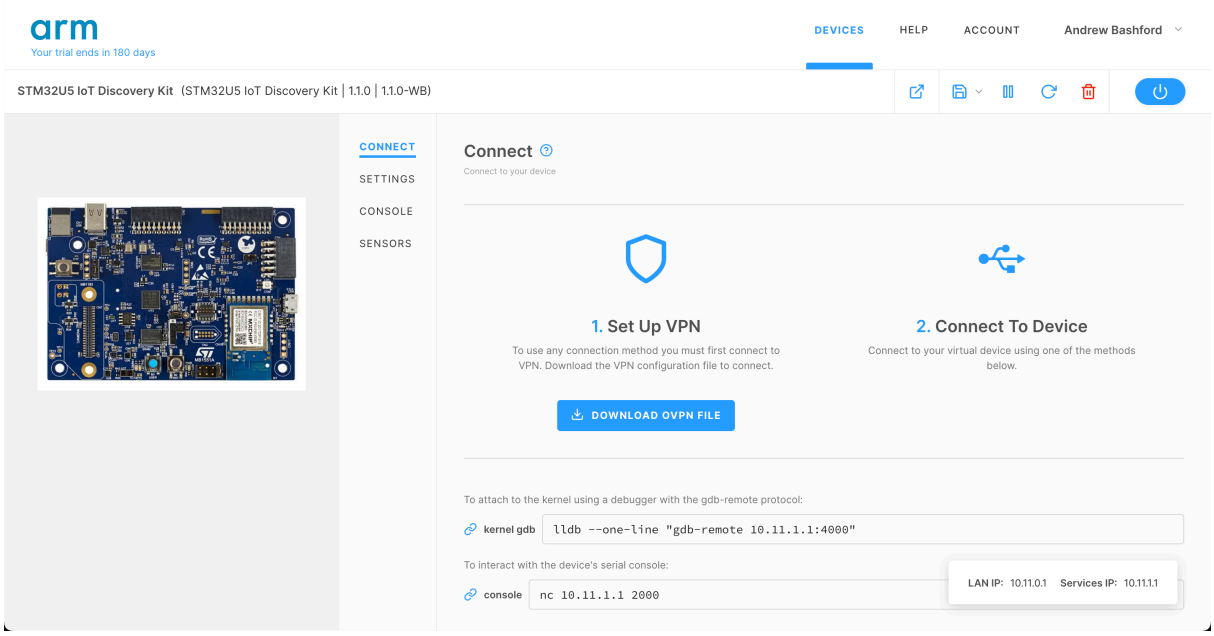
Figure 3-7: Creating device



9. Your STM32U5 IoT Discovery Kit is now ready to go!

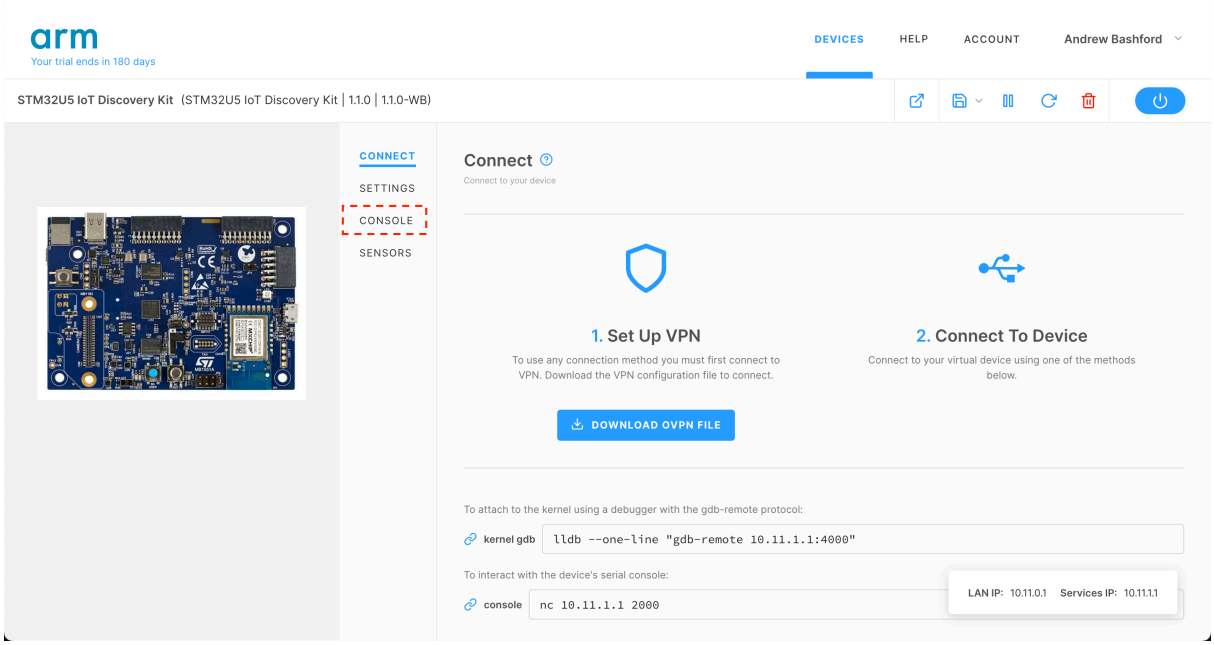
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-8: Connect



10. Click CONSOLE to go to the Console.

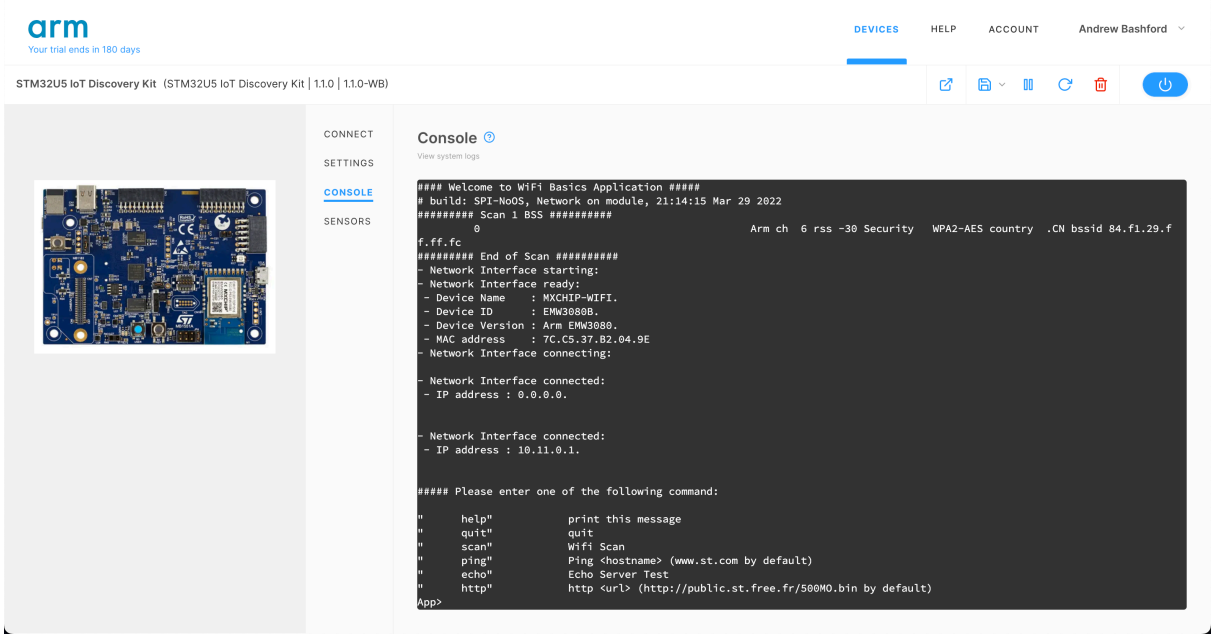
Figure 3-9: Click CONSOLE



11. Model scans for the available WiFi network and connects to the default one ("arm" in this case). There is a console menu to test various functions. For example, `http` command downloads binary from the ST website.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 3-10: Console



Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

4. Raspberry Pi 4 Example

Raspberry Pi examples:

- Raspberry Pi vMMIO example

4.1 Raspberry Pi vMMIO example

This example will exercise vMMIO using the RaspberryPi model by demonstrating an external (over the network) MMIO handler and a local (RaspberryPi userspace) driver.

Read more in the AVH vMMIO Introduction article.

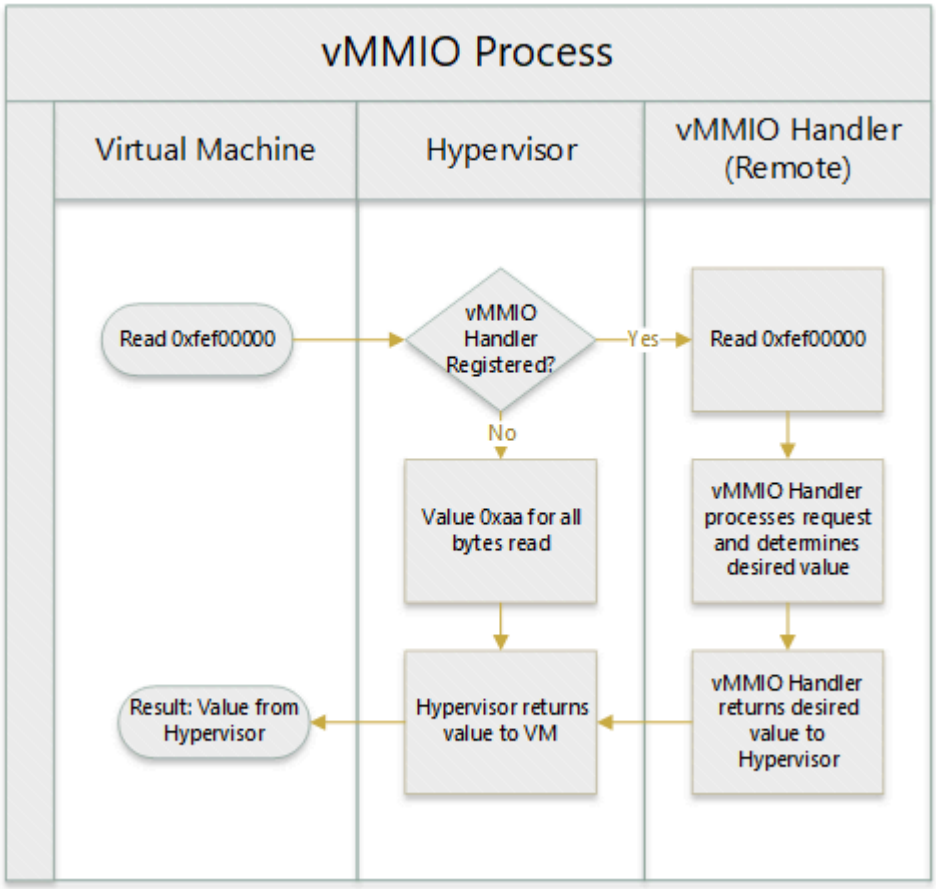
4.1.4 Flow Diagram

The flow followed in this example is illustrated below. The Writing to MMIO With a vMMIO Handler Registered example follows the “vMMIO Handler Registered” No path while the Writing to

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

MMIO With a vMMIO Handler Registered example has a remote handler registered and follows the Yes path.

Figure 4-1: Flow diagram



4.1.5 Summary

In summary, `vmmio-test-outside` is registering as an vMMIO read and write handler for a specified address range within the VM. When the physical address range specified is read from or written to, this handler receives a callback allowing it to handle the request. In this example, the `vmmio-test-inside` binary opened `/dev/mem` and simulates the OS or some other entity reading or writing to physical memory in order to exercise the callbacks in the registered vMMIO handler.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

5. API Examples

API examples:

- Python and Javascript API Examples
- GitHub Actions Example - Building Firmware, Uploading, and Testing

5.1 Python and Javascript API Examples

Visiting the AVH GitHub where you will find a repo with examples for both Python and Javascript.

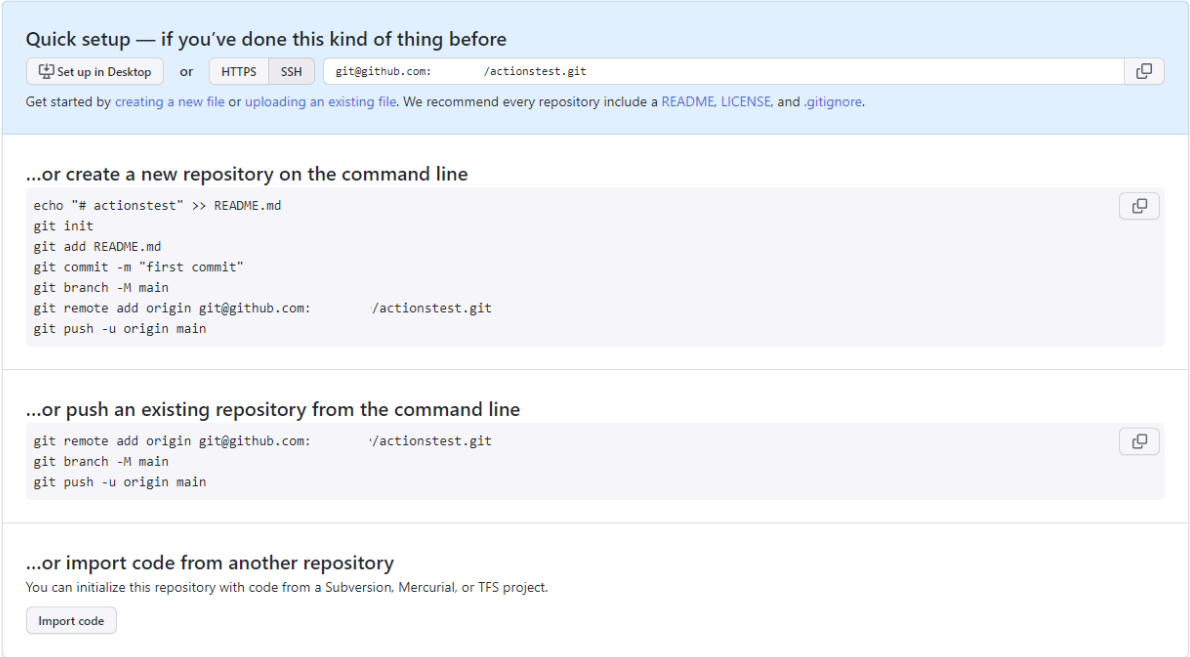
5.2 GitHub Actions Example - Building Firmware, Uploading, and Testing

The following steps help you to set up GitHub Actions.

1. To get a VM ready for the GitHub runner, follow the steps in the Running Docker on AVH Pi article.
2. Ensure you have an AVH API Token by following the steps in the Obtaining AVH API Token article.
3. Create an empty GitHub repository for your runner to accept jobs from by following the steps in this GitHub Article.

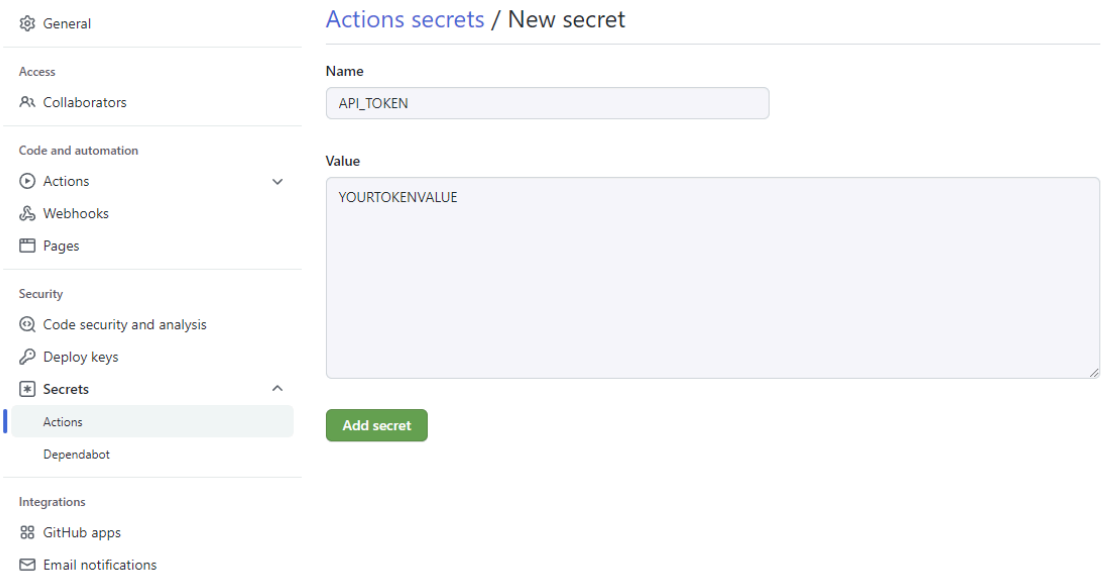
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 5-1: Create an empty GitHub repository



4. Add a "API_TOKEN" secret to your repository with a value of your API Token by following the steps in this GitHub Article.

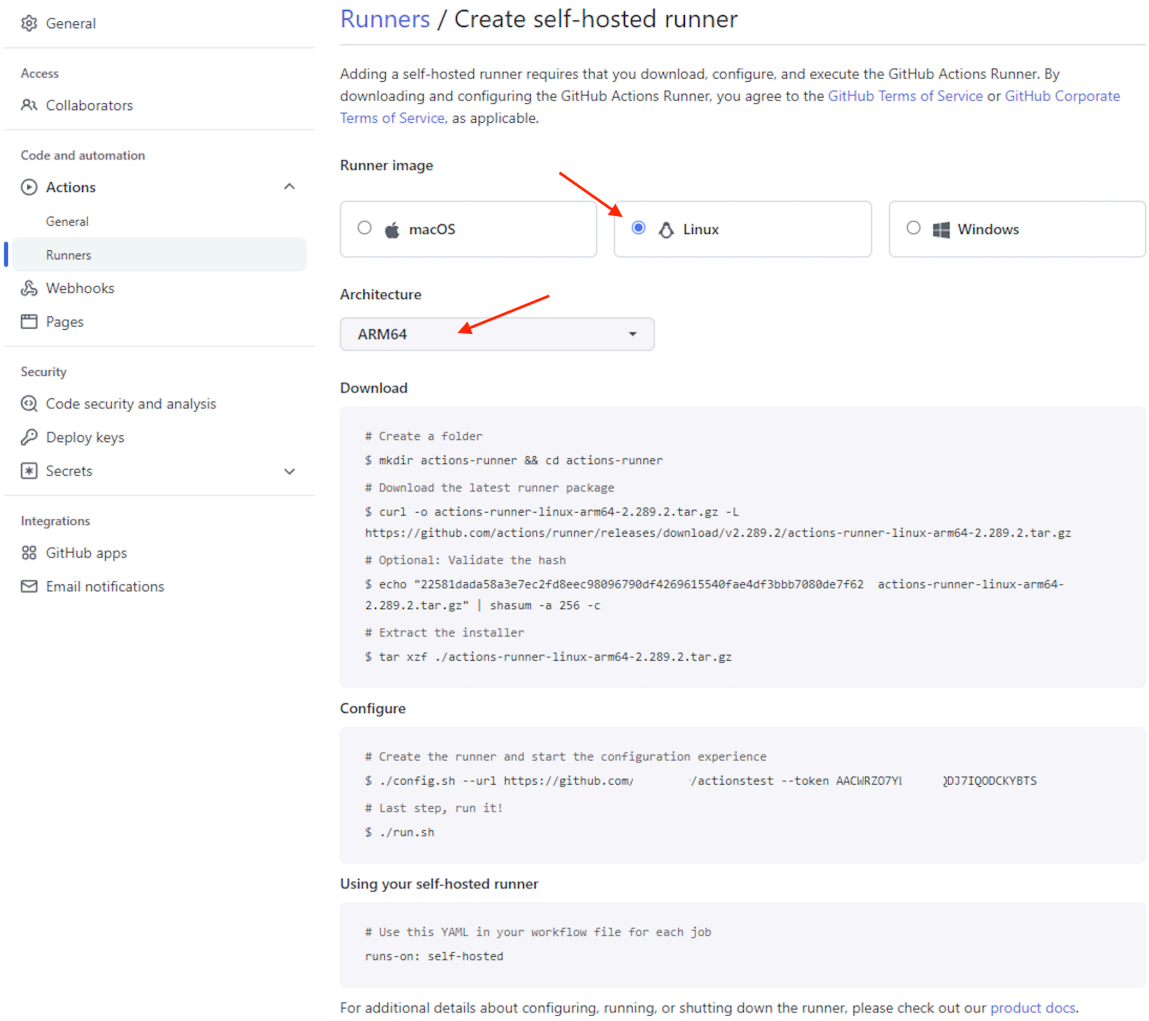
Figure 5-2: API_TOKEN



5. To configure and start the runner, follow the steps in this Adding Self Hosted Runners article. Select "Linux" as the OS and "ARM64" as the architecture.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

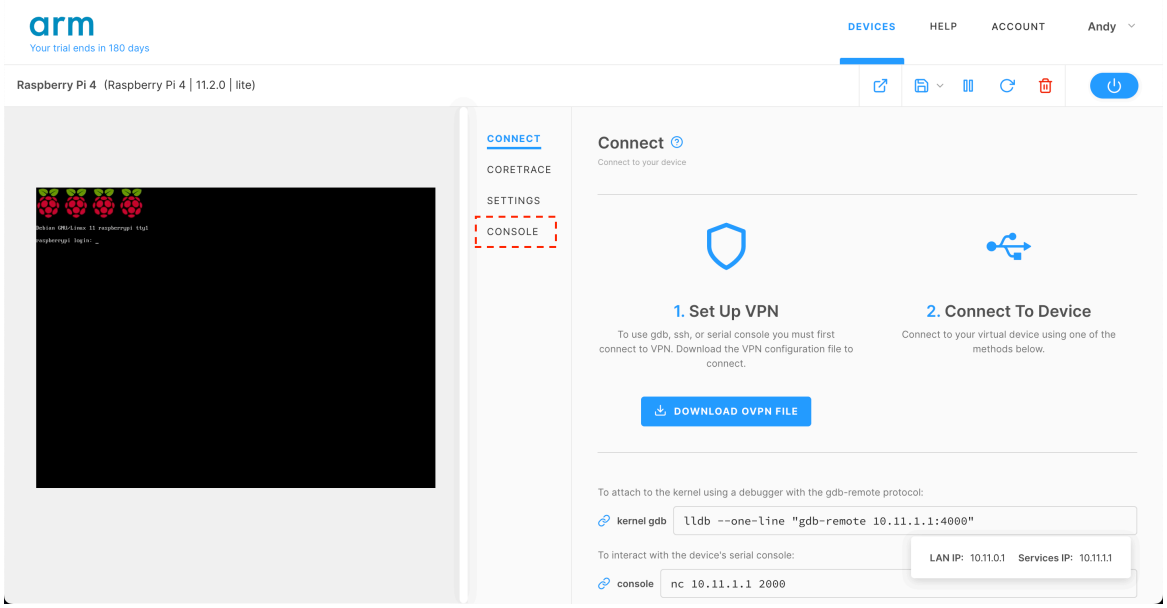
Figure 5-3: Adding self hosted runners



a. In AVH, select the Raspberry Pi VM created previously. Then select “CONSOLE” to access a shell to run the commands referenced above.

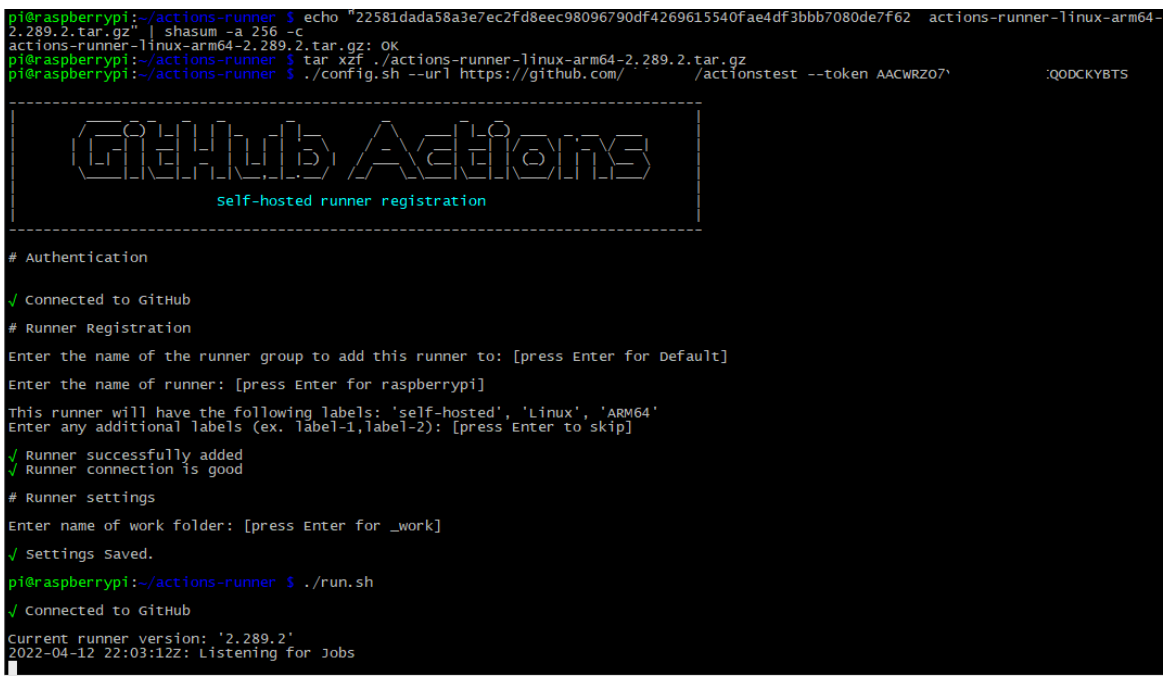
Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 5-4: Select Console



b. Upon completion of runner setup you should see something similar to:

Figure 5-5: Console output



c. Do not cancel the running process or your runner will be unavailable until it is run again. The runner can also be installed as a service, but that is not covered in this article.

6. Clone the avh-api repository on your usual local development system from GitHub with: `git clone https://github.com/ARM-software/avh-api.git`

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

Figure 5-6: Clone the avh-api repository

```
pi@raspberrypi:~$ git clone https://github.com/ARM-software/avh-api.git
Cloning into 'avh-api'...
remote: Enumerating objects: 1256, done.
remote: Counting objects: 100% (1256/1256), done.
remote: Compressing objects: 100% (638/638), done.
remote: Total 1256 (delta 600), reused 1227 (delta 573), pack-reused 0
Receiving objects: 100% (1256/1256), 25.91 MiB | 31.97 MiB/s, done.
Resolving deltas: 100% (600/600), done.
pi@raspberrypi:~$
```

Note

We used a raspberry pi for this in this example, this will likely be your laptop or desktop system.

7. Enter the `javascript/sensordemo-docker` directory

Figure 5-7: Enter the directory

```
pi@raspberrypi:~$ cd avh-api/javascript/sensordemo-docker/
pi@raspberrypi:~/avh-api/javascript/sensordemo-docker$
```

8. Run `make clean gitrepo GITREMOTE='<url to Github repo you created>'` in the console.

Note

This will overwrite any contents of the repo passed as GITREMOTE=.

Figure 5-8: Run GITREMOTE=

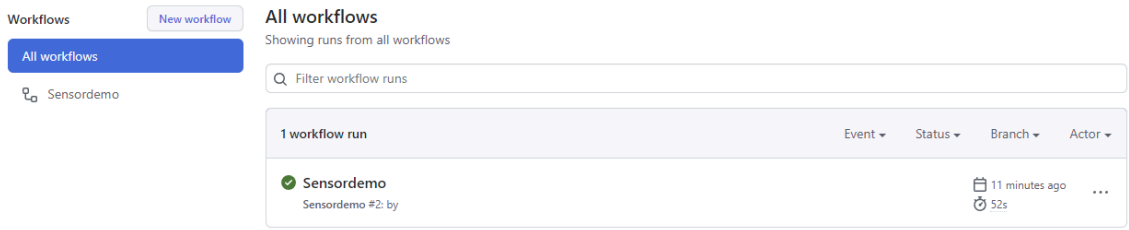
```
pi@raspberrypi:~/avh-api/javascript/sensordemo-docker $ make clean gitrepo GITREMOTE='https://github.com/        /actionstest.git'
rm -rf gitrepo
git archive HEAD --prefix="gitrepo/" | tar -xf -
cp -a ../api gitrepo/api
cd gitrepo; \
  git init; \
  mkdir -p .github/workflows; \
  mv github_workflow.yml .github/workflows/avh_build_and_test.yml; \
```

```
 create mode 100755 run_docker.sh
Example repository created in /home/pi/avh-api/javascript/sensordemo-docker/gitrepo
cd gitrepo; \
  git remote add origin https://github.com/        /actionstest.git; \
  git push -fu origin $(git rev-parse --abbrev-ref HEAD)
Username for 'https://github.com':
Password for 'https://        @github.com':
Enumerating objects: 750, done.
Counting objects: 100% (750/750), done.
Delta compression using up to 4 threads
Compressing objects: 100% (717/717), done.
Writing objects: 100% (750/750), 24.39 MiB | 8.82 MiB/s, done.
Total 750 (delta 215), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (215/215), done.
To https://github.com/        /actionstest.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
pi@raspberrypi:~/avh-api/javascript/sensordemo-docker $
```

9. You should see an action run on your repository executed from your AVH VM.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential

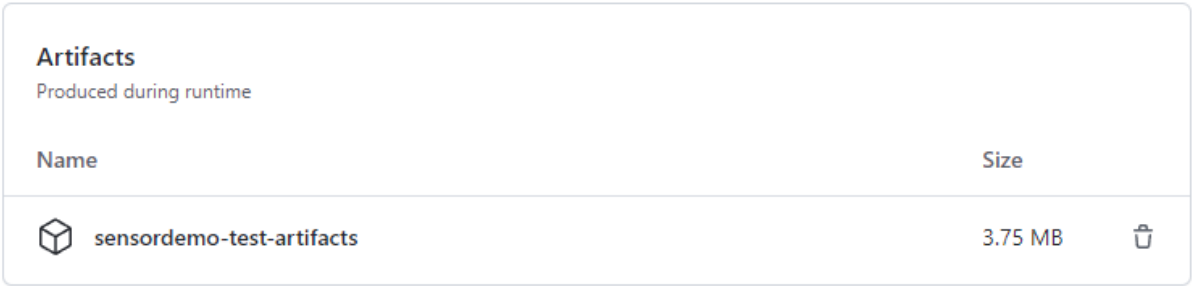
Figure 5-9: Executed action



This will compile a new firmware for STM32u5 on the pi VM, then upload it to a new STM32 VM in Arm Virtual Hardware, and run a test against it to ensure it behaved properly.

10. You will get a built firmware and the test results as the artifacts downloadable from GitHub when the action completes.

Figure 5-10: Artifacts



11. Any changes made to the repository, such as changing how the firmware behaves and pushes to GitHub, will trigger a new run.

Copyright © 2022 Arm Limited (or its affiliates). All rights reserved.
Confidential