



AT&T Shape Hackathon QuickStart Instructions AT&T Cellular IoT Starter Kit

Note: Prior to going through this series of instructions, you should complete the AT&T IoT Starter Kit LTE Shield, M2X and AT&T Flow instruction guide.

Software Setup Instructions

 To download and modify the firmware for the kit, you must first create a developer account on the <u>https://developer.mbed.org/</u>. If you do not have an account, please go to the mbed web site and follow the instructions to create one now.

ARM mbed	Hardware - Documentation -	Code	Questions	Forum	Log In/Signup	Compiler
			Search	leveloper.mb	ed.org	Search
Login Username:		Signup				
Pve forgosten my usemame Password:			mb	ed		
Ive forgotten my password						
Login		Signup				

© mbed | blog | we're hiring! | support | service status | privacy policy | terms and conditions | Language; en ja es de

- 2. Next, you will need to install the necessary serial port driver to allow your Windows laptop to see the NXP FRDM-K64F board as a serial port. If you are using a MAC or Linux based laptop, you can skip this step. Go to https://developer.mbed.org/handbook/Windows-serial-configuration and following the instructions to install the driver. To perform this step, you will need the FRDM-K64F board and a USB cable. Connect one end of the USB cable to the SDA USB connector on the FDRM-K64F board (see diagram in the Hardware setup instructions below) and connect the other end to your laptop's USB port.
- 3. Although not required for the Hackathon demonstration, it may be useful to have a terminal program to monitor the output information from the FRDM-K64F program. The mbed site provides instructions for setting up a terminal on either a Windows, MAC, or Linux computer. Go to https://developer.mbed.org/handbook/Terminals and follow the respective instructions.
- 4. Now you will need to update the FRDM-K64F boot firmware to the latest version. Follow the instruction provided at: <u>https://developer.mbed.org/handbook/Firmware-FRDM-K64F</u>



Blectronics Marketing

🔆 Technology

5. Next, you will need to create your cellular application project that will run on the FRDM-K64F board. Log into your <u>https://developer.mbed.org/</u> account and click on the **Compiler** button in the top right corner of the screen. This will bring you to the development environment where you can import the code for the FRDM-K64F board that was developed for the SHAPE Hackathon. Your screen should appear as below.

mbed	Workspace Management	
new 🗸 🖺 Import 🔚 Save 🖡	🗟 Save All 🖆 Compile 🗸 🥭 Commit 🗸 🕜 Revision 🗠 🖙 🆓 🇞 🍾 🛄 Help	No device selected 🚸 Default 💌
Program Workspace 🤇	Workspace Management Manage your Program Workspace Listing all programs in your Program Workspace Image to filter the list Mame Tage Modified Description Your Program Workspace is empty. You can import a program or greate a new one.	Workspace Details

- 6. Click on the Import button (upper left ribbon location) to import the FRDM-K64F software project. Once Import is pressed, a pop-up window will appear asking you to Add Platform. Click the Add Platform and then select NXP Semiconductor from the left hand Platform Vendor bar. This will narrow the search items. Scroll down to select the FRDM-K64F platform.
- 7. Click the Add to your mbed Complier button on the right side of the screen.

ARMmbed	Hardware 🕶	Documentation •	Code	Questions	Forum	🌡 jbeneke 🕶		Compiler
Platforms » FRDM-K64F				Search	developer.mbe	ed.org		Search
FRDM-K64 The Freedom-K64F is an ultr	F a-low-cost developn	nent platform for Kinetic	5 K64, K63, a	nd K24 MCUs,			Platform Partner	tor
Overview		1- AL	ARM		1 Table	of Contents		mpiler
The Flagship FRDM-K64F ha sorts of devices, especially ti for connected applications, Cortex®-M4 core running up	s been designed by l hose requiring optin thanks to its power e p to 120MHz and em	NXP in collaboration wit nized size and price poir efficient Kinetis K64F MC nbedding 1024KB Flash.	h mbed for j its. The boar U featuring 256KB RAM	prototyping all rd is well sized an ARM® and lots of	1. Overvi 2. MCU F 3. Board	ew eatures Features	follow	

8. Then click on the **Open mbed Compiler** button also on the right side of the screen.



 (\mathcal{A})

- 10. Click on the Import button (upper left ribbon location) to import the FRDM-K64F software project.
- 11. In the **Import Wizard** window, type **Avnet** in the Search box and click **Search** to list Avnet created mbed projects.

Import Wiza	ard								
mbed	Import Select prog	a program f gram from the lis to import from U	f rom mbed t. You can also RL	i.org odrag&drop then	n in your worksp	ace.			Import!
Programs	Libraries	Bookmarked	Upload					Avnet	Search
isting publis	hed program	s on mbed.org							
Name		Tags		Author	Imports	Modified	Description		

- 12. Scroll through the list to find the **AvnetATT_shape_hackathon** project. If you click on the **Modified** header, you can sort the projects from newest to oldest and find the AvnetATT project easier.
- 13. Click on the line for the **AvnetATT_shape_hackathon** project to highlight it. Then click on the **Import** button in the upper right corner.
- 14. A pop-up window will appear showing where the source project is coming from. Click **Import**.

At this point, you could compile the program/firmware by pressing the **Compile** button. The web-based tools will compile the program, and download it to your laptop (typically into the Downloads folder). However, before you compile, you will need to customize a couple of parameters to make this firmware unique to your device and your AT&T IOT Services account. We will tackle this next.

Modifying the Firmware Parameters

Inside the mbed **AvnetATT_shape_hackathon** project, the parameters that are needed to customize your board are in the **config_me.h** file.

Program Workspace <	Pro	gram: /AvnetATT_shape_	hackathon				
My Programs	8	Type to filter the list 🔲 Match Case 📄 Whole Word					
AvnetATT_shape_hackath O FXOS8700CQ		Name	Size	Туре	Modified		
🖲 🧰 SerialBuffered	0	FXOS8700CQ		Published Library	moments ago		
config_me.h		SerialBuffered		Program Folder	moments ago		
hardware.h hTS221.h hTS221.driver.cpp t sensors.cpp sensors.cpp wrc_control.cpp wrc_control.h wrc_control.h		config_me.h	2.1 kB	C/C++ Header File	moments ago		
		hardware.h	0.1 kB	C/C++ Header File	moments ago		
		HTS221.h	2.9 kB	C/C++ Header File	moments ago		
		hts221_driver.cpp	6.0 kB	C/C++ Source File	moments ago		
	e	main.cpp	16.9 kB	C/C++ Source File	moments ago		
		sensors.cpp	15.5 kB	C/C++ Source File	moments ago		
		sensors.h	0.9 kB	C/C++ Header File	moments ago		
	E	wnc_control.cpp	12.3 kB	C/C++ Source File	moments ago		
		wnc_control.h	0.9 kB	C/C++ Header File	moments ago		
	6	mbed		Library Build	moments ago		



Electronics Marketing

Technolog

Flow parameters: This project assumes that you are using a fork of the "Starter Kit Base" project, which is a reference design that was created using AT&T's Flow (<u>https://flow.att.com</u>) that allows the creation of on-line virtualization and other IoT functionality.

The default parameters in the **config_me.h** file are set for a specific instance of this project. When you fork the original project, you get your own instance and it will have its own base address. At the bottom of the Flow environment, when you click on the **Endpoints** tab, you will see the URL information that is specific to your instance. Of note is the **Base URL**.

In the example below (as in the default mbed project), the Base URL is: https://run-west.att.io/1e464b19cdcde/774c88d68202/86694923d5bf28a/in/flow

You have to take note of two parts of this address. The **run-west.att.io** part is the server URL, and you have to make sure the **MY_SERVER_URL** field in **config_me.h** matches this. Then there is the rest of the base URL in green above, that needs to be pasted into the **FLOW_BASE_URL** field.

There is also a **FLOW_INPUT_NAME** field. This should match the name of the HTTP IN port in the Flow project that you want to send sensor data to. In the default Flow image below this input is inside the "Climate GET" port, and is named "**/climate**".



The **FLOW_DEVICE_NAME** field must contain the name of the instance of the Virtual Starter kit in Flow that you will be communicating with. Usually this will be "vstarterkit001", but if you have problems communicating you can verify that this is indeed correct. Note that this device will not be created until you click the "Initialize" input on the Virtual Device tab of the Starter Kit project in Flow. At that point it becomes available in M2X and you can see it there, as the **DEVICE SERIAL** field under Devices as in the image below.

TATE TO F Services M2X Devices Devices Devices Devices Virtual Starter Kit Created BY: Devices Intermed and the second and the seco	Electronics	s Marketing 🔅 Technology	
M2X		AT&T IoT Services 🗸	
Devices > Virtual Starter Kit Virtual Starter Kit Edt Defet		M2X Devices Distributions Deshb	oards
Virtual Starter Kit Edit Delete		Devices > Virtual Starter Kit	
Image: Created BY: Image: Device ID: Image: Device Serial: Device Serial: Vstarterkit001 Copy Image: Device ID: Image: Device Serial: Image: Device Serial: Vstarterkit001 Copy Image: Device ID: Image: Device Serial: Image: Device Serial: Image: Device Serial: Vstarterkit001 Copy Image: Device Serial: Image: Device Ser		Virtual Starter Kit 🖽 Dele	
Image: Mail: % PRIMARY ENDPOINT: Image: Tags Image: Image: Most recent Location: % PRIMARY API KeY: Image: Tags		CREATED BY: DEVICE ID:	3c99a645e9ef56504ae Copy vstarterkit001 Copy
MOST RECENT LOCATION: PRIMARY API KEY:		EMAIL: Contract Contr	T:
No location available d8e190b401ffe3a4ee78b272110e036c Copy		MOST RECENT LOCATION: No location available	3a4ee78b272110e036c Copy

- 1. After making these modifications to the **config_me.h** file, click the **Save** button along the top bar.
- 2. To compile the program with the customized settings for your device and AT&T IoT Services account, click the **Compile** button along the top bar. The compiler should successfully compile your source files and download the binary programming file, **AvnetATT_shape_hackathon_K64F.bin** to your laptop. Depending on the browser you are using or the browser's download settings, this file may automatically get placed in your Downloads folder or prompt you for a location to save the file. You may wish to move this file to a local project folder for later use in this tutorial.
- 3. Disconnect the USB cable from your laptop to remove power from the FRDM-K64F board.

You have now built the software that will be needed by the FRDM-K64F board to run the SHAPE Hackathon demonstration.

Hardware Setup Instructions

1. Plug the Avnet Cellular shield onto the Arduino connectors on FRDM-K64F board.





- 2. Plug the supplied AT&T SIM card into the SIM socket (X3) on the shield.
- 3. Screw the two antennas to the shield SMAs connectors (X1, X2).
- 4. The cellular shield requires 5V power that is supplied by the included AC/DC power adapter. It is not recommended to use a laptop USB port for this power source due to the limited power capabilities that most USB ports have. Plug a microUSB cable into the 5V USB shield power (X5) on the cellular shield and connect the other end of the cable to the provided AC/DC supply. LED1 will light green.
- 5. Plug a microUSB cable into the SDA USB (J26) port on the FRDM-K64F board. Connect the other end of the cable to a USB port on your laptop. The kit should enumerate as a standard **MBED** USB drive.
- 6. From your laptop, locate your compiled AvnetATT_shape_hackathon_K64F.bin file and drag and drop this file to the FRDM-K64F board (MBED device) connected to your laptop via the USB cable. This will program the non-volatile Flash memory on the FRDM-K64F board with your application firmware. Programming should take about 10 seconds, during which the FRDM-K64F board green LED should go through several flashing sequences.
- Once the download is complete, unplug both the USB cable from your laptop and from the AC/DC power adapter to power down the system. Then reconnect the cables, connecting the 5V USB power cable to the power adapter first, followed by the SDA USB cable to the laptop.
- 8. The User LED on the FRDM-K64F board should be RED while the wireless connection is established. It will then turn BLUE when it successfully connects to the AT&T network. The LED then turns GREEN when it connects to your Flow project and gets a response back. These results



Electronics Marketing

can also be monitored through a terminal connection to the SDA USB serial port on the FRDM-K64F board.

9. In your Flow design, you should see debug messages as the temperature and humidity sensor data starts coming in. Debug messages can be viewed by clicking on the Debug tab at the bottom of the Flow environment. There are a couple of Debug ports (green) in the Flow design and you can enable or disable them by clicking the tab/dot on their right side. When you have only the http-in Debug port enabled, you should see messages from the "Climate GET" HTTP input appearing as in the image below.



This concludes the Quick Start Instructions for your AT&T Cellular IoT Starter Kit.

To download additional information on the kit, including board schematics, visit: www.cloudconnectkits.org/product/att-cellular-iot-starter-kit