



Mobile devices, such as smartphones, are present in all aspects of our lives. Mobiles are replacing other electronic accessories at a fast pace and integrating their functionalities: music players, GPS, cameras, etc. The role they play in our lives has continued to grow from that of a phone to the role of permanent companion.

Mobiles offer us a user interface that is adapted to our habits and that evolves seamlessly. Other appliances have gone so far as to adopt their look and feel. Today,

however, controlling other appliances and equipment using **human-machine interfaces (HMI)** on our mobiles is becoming increasingly common.

This is why IoTize’s TapNLink offers a global solution for controlling appliances and equipment from HMI that are based on mobile devices.

TapNLink addresses all the facets of integrating a mobile HMI for any electronic system. TapNLink offers companies the advantages of a complete, qualified hardware/software solution that implements by configuration only.



This article provides an overview of IoTize’s solution for creating mobile HMIs. It introduces the key requirements that design teams must address and illustrates the advantages of the TapNLink approach.

What does it mean to provide an HMI on a Mobile?

Creating user interface or the Human-Machine Interface (HMI) for a system on a mobile device cannot be reduced to a simple choice of software development or connectivity. There are many facets that design teams have to consider:

The mobile app

Starting from the mobile device, the first thing they'll need is an 'app.' Mass market products must offer apps for Android, iOS and sometimes Windows operating systems. These can be developed using:

- A native platform that is dedicated to an OS (ex. Java\Kotlin for Android, ObjectiveC\Swift for iOS)
- Or a cross-platform framework (native or hybrid).

TapNLink allows designers to choose between native and hybrid app development. Getting started, the hybrid cross-platform option is favored. TapNLink's configuration environment (*IoTize Studio*) has an integrated 'App Generator' based on Ionic. With just a few clicks and without any programming knowledge, designers can generate user interface apps that are ready to publish to app stores. These apps get users past the initial hurdles of app development so they can test ideas rapidly with little experience. These can be refined using other tools and they do not preclude users from developing apps for TapNLink using other app development frameworks.

An app launcher

Because more and more electronic devices can be controlled from mobile devices, you will soon have hundreds of apps installed on your mobile. The challenge comes when you need to select the right App, launch it and start the communication with the right device. A good, automatic launcher will come in handy. On the market today, there are options to automatically launch apps:

- **QR codes**, which require a QR code scanning app.
- **Near Field Communications (NFC)**, which works with a simple 'touch' and without installing or launching other apps.

TapNLink modules embed an NFC tag. The first time you touch a new TapNLink with an NFC equipped mobile device, the right app is installed (with your authorization). Then the NFC tag can be used each time to launch that app. You just approach your mobile to a system's NFC tag (within 3 cm), the right app is launched and the communication is established between the mobile and the system.

Wireless communication channels

Using wired connectors for communication between electronic equipment and mobiles is obsolete. Small and fragile connectors are not the right solution for frequent usage. Wireless communications such as NFC, Bluetooth Low Energy (BLE) and Wi-Fi overcome the durability problem.

With TapNLink, you have the choice between NFC, BLE or Wi-Fi. NFC is always present as an app launcher, even when BLE or Wi-Fi is used as the main communication channel. TapNLink integrates the antennas and all the required firmware for these channels. On the mobile, IoTize provides all the software to handle the mobile's low-level communication protocols.

Security mechanisms

Security is not optional in wireless communication. Security mechanisms include role-based user management (actions are allowed/prohibited depending on users' roles), a user authentication system, and encryption of communications to avoid eavesdropping.

TapNLink manages all aspects of security and security configuration is implicit in the configuration process, thus it cannot be overlooked. When configuring TapNLink, the designer defines user profiles, access rights and associates users with those profiles. Profiles (and the associated access rights) are associated with data in the target system, which will be visible or modifiable via the HMI app on a mobile. TapNLink also implements a strong mechanism for user authentication and full encryption of communication channels so that designers don't have to implement this themselves.

Hardware integration

To establish wireless communications with electronic equipment, you need a radio transceiver, antenna and the associated components that make everything work. Implementing all this requires experience and specific know-how regarding NFC and 2.4 GHz wireless protocols.

TapNLink modules integrate the electronic radio transceivers and antennas for you. The NFC antenna also often needs to be off-boarded from your main electronic design for accessibility. TapNLink's design resolves all these problems for you. Mechanically, TapNLink can be fixed to the front panel of an electronic system. Customizable plastic cases are also available as options.

Firmware integration

Communication with a mobile requires a stack protocol and various firmware blocks to manage things like access control and security. The cohabitation of the target system's firmware and the communication firmware can result in conflicts.

Handling these software bricks separately on different processors provides modularity that improves reliability, security, flexibility and facilitates future evolutions. TapNLink manages all the communication firmware independent of the target system's processor. TapNLink firmware does not require any coding – only configuration. The configuration environment not only configures the wireless module but, depending on the selected wire protocol, also generates any C source code that may be needed for the target system's firmware. This code has a very small footprint and needs very little processor time.

Regulatory certification

Radio communication requires various certifications depending on the intended geographic market of the finished product. Wi-Fi and BLE are used everywhere in the world, but require different authorizations when entering specific geographic markets. For example, devices must comply with the CE-RED directive in European Union, and with FCC regulations in the USA. All these certifications take time, money to attain and present a risk of rejection.

TapNLink modules are themselves certified to facilitate certification of your finished product. For example, TapNLink is currently certified for the European Union (CE), USA (FCC) and Canada (IC). In North America, FCC and IC certificates are transferrable to systems that integrate TapNLink. For Europe, an overall final test is required for the system, but this testing is facilitated by TapNLink's CE pre-certification. Certifications for other markets are possible upon customer request.

Firmware update

Firmware-Over-The-Air (FOTA) is an increasingly common market demand. Because of continuously evolving technologies, features and threats, the ability to update firmware efficiently and reliably is a must.

TapNLink modules include a secure, signed mechanism for firmware updates over a wireless channel. For ARM Cortex M based processors, TapNLink can also update the firmware of the target system that it is connected to. To implement this feature securely, IoTize provides designers all the necessary tools and applications.

Additional advanced features

Future TapNLink modules integrate a variety of additional, helpful features:

- **On-module data logging:** TapNLink embedded memory will store selected data at a configurable sampling rate. This data can be recuperated and analyzed locally on a mobile device, or securely packed and sent to a server.
- **Remote communication from the internet:** TapNLink resources will enable remote control of your system with full end-to-end security using WLAN or LPWAN wireless interfaces. With NFC-only or NFC + BLE modules, a user's mobile can be used as a relay to the internet. Communication is encrypted end-to-end and cannot be eavesdropped by a malicious mobile.
- **Low power mode:** TapNLink will offer low-power processors, sleep mode and wireless wake up features to minimize energy consumption for systems that require this.

TapNLink Benefits

TapNLink modules bring you all the expertise and features needed to quickly build an HMI for a mobile. The only thing you need to do is just configure them for your system. With TapNLink, you benefit from:

- **Savings on your development investment:** Handling any one facet of TapNLink's implementation (mobile apps, wireless design, security, etc.) is costly, challenging and risky. These costs, challenges and risks are magnified dramatically if you want to address them properly in a project that encompasses the full scope of TapNLink capabilities.
- **Reduced time-to-market:** Even with a good team and the 5-10 specialists needed for such a project, you'll advance faster with a global, ready-to-use solution. TapNLink just needs to be configured to be adapted to your system.
- **Reduced Risks:** Even if your specialists are good, you can hope that the different aspects will be successfully handled with a probability of 90%. But if you evaluate the risk of failure when 5 good specialists are working together, success for the overall project is no longer 90%, but $90\%^5 = 60\%$.

With TapNLink, you start from a working prototype (PoC) within just a few minutes. Reaching the production phase won't be much more difficult. It only requires:

- Selecting the appropriate options.
- Configuring the module.




- Modifying your system slightly for the target protocol.
- Publishing the generated app.

TapNLink options

The table below summarizes the options for:

- The mobile app development framework.
- The type of module depending on the wireless protocol you need.
- The communication protocol between your electronic system and TapNLink.

All solutions are possible among the different combinations: **HNS, CBP, AWU, HNP, HNU, CBS**, etc. A total of $3^3 = 27$ combinations are all valid. Just select one cell of each row in the table below:

Your HMI			
Mobile App (choose language)	<p>HTML (H) auto generation</p> <p>This is a pushbutton solution. The generated HTML/JS is displayed by our generic 'Tap Manager' that is available on Google Play Store and Apple App Store.</p>	<p>Ionic (C) auto-generation</p> <p>This is a pushbutton solution. Once generated, the cross-platform app should be customized and built in order to be published on App stores.</p>	<p>APIs (A) for Native Android or iOS</p> <p>Using our libraries, build Apps without having to handle low level communication between TapNLink and the mobile.</p>
Host protocol (choose Tap type)	<p>NFC-only (N)</p>  <p>Configure systems from a mobile (a.k.a. 3-Stroke Configuration).</p>	<p>NFC + BLE (B)</p>  <p>Configure and monitor systems comfortably from a mobile.</p>	<p>NFC + Wi-Fi (W)</p>  <p>Monitor systems locally on a mobile or remotely (permanent internet access required).</p>
Target protocol (choose one)	<p>SWD (S) for Cortex-M MCUs</p> <p>Make PoCs without modifying existing hardware or firmware of the target system.</p>	<p>S3P (P) Only 2 GPIOs</p> <p>Universal, secure and flexible solution. It can be used on the SWD port for most of MCUs.</p>	<p>UART (U) With or without Modbus</p> <p>A legacy solution that is quite efficient when the hardware is available.</p>
Your System (target system)			

Which module for your system?

NFC, NFC + BLE, or NFC + Wi-Fi: all the modules are not equal. They match different contexts, different categories of use cases:

- NFC-only is a low cost solution. For the periodic configuration of the target system, it is the perfect choice.
- NFC + BLE is convenient for on-site monitoring of the target system from a mobile. NFC automatically launches the app and starts the communication. BLE range makes monitoring more comfortable.
- NFC + Wi-Fi is the right solution for target systems that connect to the internet via a local network, or for coordinating equipment remotely. The TapNLink connects as a 'station' to a local Wi-Fi network. It can also provide an on-site access point for mobiles, but its only advantages over BLE are increased range and transmission speed.

Which framework for app development?

To have an app for both Android and iOS, use the Cordova\Ionic cross-platform framework. However, the choice of native development makes sense if performance and speed are concerns.

During development, to create a rapid PoC, use the HTML\JS WebApp generated by IoTize Studio via 'Tap Manager' App. This IoTize app handles the rendering of the WebApp and the underlying communication channels.

The configuration environment can also generate an Ionic app project ready to be built using the Ionic framework and published on app stores (refer to Google and Apple App Store websites for information about app publishing).

Which protocol for the target system?

3 standard wire protocols are available to connect TapNLink to a microcontroller:

- SWD is a debug protocol for ARM Cortex M-based MCUs. Just enable the debug port on the MCU and connect TapNLink to it. No modification of system hardware or software is necessary. Just specify which variables of the application to monitor in the TapNLink configuration. TapNLink will access these variables. To easily create a PoC, this is a magical solution.
- S3P is an evolution of SPI. It is described on IoTize website. It requires two GPIOs (General Purpose Input/Output pins) and a small piece of C code. This code is automatically generated by IoTize Studio. S3P is the preferred solution for its security and needs very few processor resources.
- UART is the most restrictive. It requires an UART on the target system's MCU. When selected, a simple Modbus agent is automatically generated. This agent needs to be added to the code of the target system firmware project. Legacy UARTs are very popular, and Modbus very well-known in the industry.

How immediate is TapNLink solution?

Of course, the final integration will take more than a couple of minutes. The effort depends on what you have selected in the previous table. The choice of the type of TapNLink itself does not make things more or less complex. For an instant solution, choose **HxS** (*HTML app – Any wireless protocol – SWD wire protocol*). Choices for the App framework and the Protocol, may require more effort so:

- Consider a couple of hours for S3P or UART. For both, IoTize Studio will generate the source files, which have to be added to the target system firmware project.
- Auto-generated Ionic apps have to be built using the Ionic framework and then publishing it to the app stores.
- Development time of apps based on our APIs (Swift/Java/JavaScript) will depends what needs to be implemented above the communication bridge with the target system.

Note that if changes are required to meet future needs for a different wireless communication channel between the target system and mobiles, it is only necessary to change the type of TapNLink module. There is no need to go through the whole configuration and app development process. The same configurations work for the different wireless protocols.

NFC advanced features

TapNLink includes additional features based-on the NFC communication protocol. This enhances TapNLink with the following functionalities:

- For the ‘NFC only’ modules, [*3-stroke™ configuration*](#) provides the most effective way to configure a target system from and advanced HMI at a very low cost.
- For NFC with BLE or Wi-Fi, NFC eliminates the need for permanent advertising signals. This reduces energy consumption and unwanted radio emissions. It also automates launch of the right app and handles the pairing for the radio communication.
- In all cases, NFC is unobtrusive and efficient. It automates authentication and eliminates the risks of eavesdropping.

Conclusion

Human-Machine Interfaces on mobiles are becoming very common today. However, they are still very complex to implement. TapNLink brings together all the know-how and solves all the difficulties that a team of developers would encounter. It makes the overall connectivity a generic component that is easy to add to any electronic system.