

Quick Start Guide to Smart.IO

V0.1 September 6th, 2017

richard@imagecraft.com

Richard Man, ImageCraft, <https://imagecraft.com/smartio>

What does Smart.IO do?

Smart.IO allows embedded designers to create a modern UI for any embedded design. The most important attribute of the Smart.IO toolkit is also one that for a number of people takes a few moments to sink in:

"Let's emphasize this most important point: you do NOT need to write ANY BLE/wireless or app code. None, zero, zilch. Smart.IO does that for you, and the UI runs on any iOS and Android devices." With Smart.IO, prototyping and even products are no longer limited by "hardware knobs, bits, and bobs".

Components of Smart.IO

The Smart.IO toolkit is made up of both a hardware module and software components. This table lists the major components and which documents are applicable to each of them.

Component	Description	Relevant Document
Smart.IO hardware module	Provides hardware for BLE communication and the firmware support to implement the Smart.IO API	Smart.IO Hardware Integration Guide
Host Interface Layer	Source code in Standard C that an embedded user compiles with their host MCU firmware. Converts the Smart.IO API calls into command stream for the Smart.IO firmware.	Host Interface Layer
Smart.IO API	The API that the embedded users call to build and interact with the UI	Smart.IO Software API Smart.IO User Interface Elements
iOS and Android Smart.IO apps	Smartphone apps that interface with a Smart.IO-enabled device and provide the UI for the device	Briefly described in this document

Smart.IO Module

This is a small 15mmx26mm module with an ST BlueNRG1 BLE SoC (Bluetooth Low Energy System on a chip). It provides the BLE functionality

Host Interface Layer

This piece of software runs on the host MCU and is provided as a set of Standard C source code files. It provides the Smart.IO API interface to the host MCU.

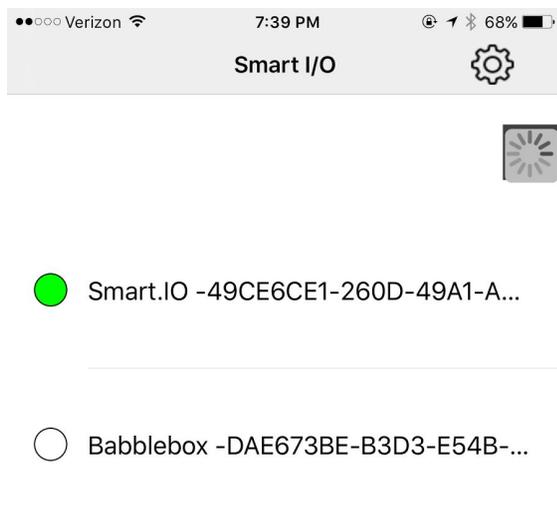
Smart.IO API

This is the API that the host MCU calls to create the UI on the smartphone. The host interface layer converts the host function calls into command streams, and sends them to the Smart.IO firmware running on the Smart.IO hardware module. From there, the commands are sent to the smartphone app for execution.

Smartphone App

The smartphone app displays the UI created by the Smart.IO calls, and sends the data from user interactions (e.g. changing of a slider) back to the host MCU for processing.

You can download the iOS or Android Smart.IO app from the Apple App Store and the Google Play Store respectively. When you start the app, it will scan for any Smart.IO-enabled devices. On iOS, that looks something like this:



Devices with green dots are ones which the Smart.IO app can connect to. “Smart.IO” is the device name, and the long hex string following that is the UUID (Universal Unique Identifier).

The default name for a device using the Smart.IO toolkit is “Smart.IO”, but embedded users / designers may and should change the name of their Smart.IO-enabled devices.

The app connects to the Smart.IO-enabled device by tapping on its name. The host MCU firmware will be notified when a connection is made. It is then up to the host MCU firmware to run the UI generation code to build the API.

Complete Data Flow of a Smart.IO API Call

The following is the complete data flow of a single Smart.IO API call.

Host MCU Firmware	Smart.IO Host Interface Layer	Smart.IO firmware	BLE ¹ firmware	Smart.IO smartphone app
<i>Data flow direction</i> →				
Smart.IO API call				
	Convert API request to command stream			
	Initiate SPI data transfer			
		Process SPI into BLE data		
			BLE transfer	
				GUI actions
← <i>Data flow direction</i>				
				Command response
			BLE transfer	
		Process BLE data into SPI stream		

¹ ST BlueNRG-1 BLE SoC

		Pull Host IRQ high		
	Interrupt handler reads SPI data stream			
	Return data to host MCU			
Receive return value				

Where to Go from Here

First, you need to enable your hardware with the Smart.IO module. See the document [Hardware Integration Guide](#).

Next, you need to port the Host Interface Layer to your MCU and compiler. Check the webpage <https://imagecraft.com/download/smart-io-downloads> to see if a port is already available. (See the document [Host Interface Layer](#).)

Download and install the iOS or Android app.

The sample host interface code includes a demonstration program. It has more than half a dozen examples of creating simple UI pages. You can run them and observe the resulting UI on the phone app, and experiment with different API calls or function arguments to produce the best effect. (Be sure to consult the documents [Software API](#) and [Smart.IO User Interface Elements](#).)

Smart.IO is a new and evolving product; please be sure to send feedback / comments / suggestions to info@imagecraft.com