



# Smart.IO

## The Modern UI for Your Embedded Tech

Version 1.0 Nov 10, 2017

ImageCraft [info@imagecraft.com](mailto:info@imagecraft.com)

<https://imagecraft.com/smartio>

# The Embedded Systems User Interface Problem

- Microcontrollers are getting smaller, more powerful, and less expensive.
- The possibilities for Internet of Things (IoT) and other “smart” products are endless: smart dimmer switches, smart battery chargers, trackers, etc.

*However, without a good user interface, they are still quite "dumb".*

Traditional interfaces include: hardware knobs, switches, LEDs, LCDs etc.

→ Tedious for prototyping, and hard to create elegant designs for production.



# The Smart.IO Solution

- Only about \$20 (much less in volume),
- And less than a day's work,

.....and you can make **Smart.IO** work for you!

Smart → Smarter → **Smart.IO**



# What is Smart.IO?

- **Smart.IO** is a toolkit which **allows a smartphone or tablet to act as the Graphical User Interface (GUI)** for embedded systems.
- The embedded designers do not write the UI app. **Smart.IO** has a **programmable app**, to customize your interface requirements.
- The embedded designers do not write the wireless code. **Smart.IO** uses **wireless technology** transparently.
- **Smart.IO** is self-contained, and **can be used with any microcontroller**.



# Benefits of Smart.IO

- Smartphones have high-resolution screens and input options, they are ubiquitous, and they usually don't get misplaced like dedicated remotes.
- Smart.IO is **easy to incorporate** into a design
- It's **simple and easy to create a UI** with Smart.IO
- Smart.IO provides **significant cost savings**, compared to alternatives.
- With Smart.IO's programmability, **improved capabilities can be added** to the GUI, or its functions can be changed, **without needing to update the hardware** on the device itself.



# Benefits to the End Users

- **Wireless Control** - End users do not need to physically touch or be right next to their gadgets.
- **Better Control** - A Smartphone screen and keyboard provide a much more versatile and user experience than traditional limited hardware controls. Even a comprehensive device user manual can be featured on the screen with Smart.IO-controlled device.



# Smart.IO Components

- **Smart.IO chip module:** a 15mmx25mm chip module which designers incorporate into their hardware design.
- **Smart.IO Software API** (Application Programming Interface): all software interaction with the Smart.IO chip is conducted through a set of easy-to-use API functions.
- **Smart.IO app:** a smartphone app, available on iOS and Android devices, provides the user interface. For production use, device makers may license customized versions of the app specific to their devices from ImageCraft.



# Benefits As Compared To Hardware I/O

Smart.IO is **inexpensive** (about \$20, less in quantity)

Commonly-used current hardware I/O component costs:

- 2-line graphic LCD: \$10+
- Touch screen LCD: \$20+
- Low resolution LED segment: \$1
- Knobs, sliders, etc.: \$1-\$2

All of these take up space in products and affect the appearance of the products.



# Benefits As Compared To "Rolling Your Own"

By using Smart.IO, you are saving both time and expense:

- No need to design and debug wireless technology hardware
- No need to write wireless code
- No need to design and implement communications protocol
- No need to write a smartphone app and port it to both iOS and Android

**This can easily save 12+ person-months of work.** Smart.IO allows you to make prototypes, and release your product earlier, for **faster Time To Market.**

Plus, **your production hardware costs will actually be lower**, as Smart.IO can be manufactured in volume, driving down the component cost.



# Smart.IO Chip And Wireless Technology

- The **use of the wireless technology is transparent** to the embedded designers.
- The underlying technology of Smart.IO uses **Bluetooth Low Energy (BLE)** with future upgrade to wifi technology possible.
- **BLE is a common wireless protocol** available in all later-generation smartphones and tablets, even without a data plan or internet connection.
- The smartphone or tablet could be **used up to 10 meters away** from the IoT device.



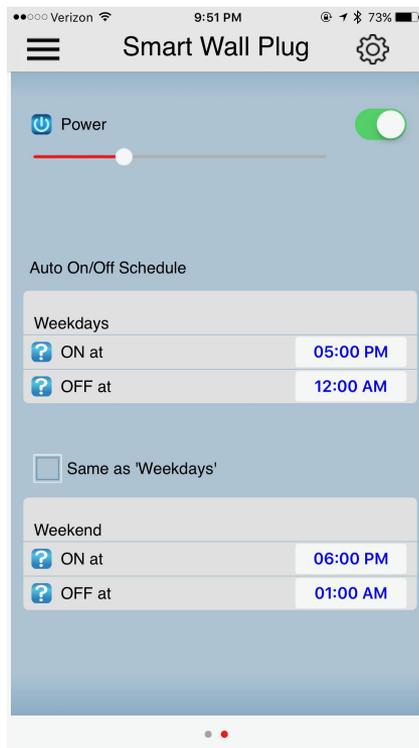
# Security: Built-in

The **Smart.IO API** and the **Smart.IO app** are **designed with security in mind.**

A customized app includes your **private security key**, so that the Smart.IO chip in your systems will only work with your own customized app.



# Embedded System Requirements



The Smart.IO Toolkit **can be used with any common 3.3V MCU**, from 8 bits to 32 bits.

The hardware interface to the Smart.IO chip is provided through **high speed SPI**.

The interface to the software API is released in **ANSI C compatible source code**. The interface code compiles to 8-10K bytes of code memory with around 2K RAM footprint.



# Device Independent UI

**The Smart.IO UI API is device independent.** The embedded designers do not need to specify coordinates and other display attributes.

**HOWEVER:** The Smart.IO generated UI still takes advantage of the full resolution of the smartphone, regardless of make and model, on both iOS and Android OS.



# Volume Production

For large volume production, **a customized Smart.IO app will perform as fast as a native app written from scratch** specifically for the product.

**Designers may license our IP** without using our hardware module to drive down the manufacturing costs to very low levels.



# Example UI Design

- Create the UI when connected and restore UI control states (omitted in the example below)
- Firmware actions are based on Input changes (callback functions)
- Save states when disconnected (omitted in the example below)



# MCU Main Example

```
int main(void) {
    Setup();

    SmartIO_Init(Connect_CB, Disconnect_CB);
    int last_state = 0;
    while (1) {
        while (last_state == connected)
            if (SPI_State == SPI_SMARTIO_ASYNC_REQUEST)
                SmartIO_ProcessUserInput();

        if (connected) {
            DelayTenth(1);
            CreateUI();
        }
        last_state = connected;
    }
}
```



# CreateUI Example

```
void CreateUI(void) {  
    tHandle p0;  
    tHandle u0, u1;  
  
    p0 = SmartIO_MakePage();  
    SmartIO_AppTitle("Smart Wall Plug");  
    u0 = SmartIO_MakeOnOffButton(0, 0, 1, Button1);  
    SmartIO_AddText(u0, "Power");  
    SmartIO_SetSliceIcon(u0, SMARTIO_ICON_POWER);  
    u1 = SmartIO_MakeSlider(1, 0, 30, Slider1);  
    SmartIO_UpdateSlider(u1+1, current_light_value);  
  
    SmartIO_EnableIf(u0+1, u1+1, 0);  
    ...  
}
```



# Slider Callback Code Sample

```
uint16_t current_light_value;
...
void Slider1(uint16_t val) {
    current_light_value = val;
    timer2.ChangePWMDutyCycle(PA2_CHANNO,
        current_light_value);
    SmartIO_ToggleLED(3);
}
```

