IzoT™ Product Questions

Where can I get source code for the IzoT Device Stack and the IzoT Server Stack?

Full source code for both stacks is included with the IzoT SDK download at iiot.echelon.com/get-started.

Do the IzoT stacks only run on a Raspberry Pi?

No, the IzoT stacks can be ported to any compatible 32-bit or 64-bit processors and operating systems. They have been tested on a Raspberry Pi running Raspbian Linux, but do not have any dependencies on the Raspberry Pi platform with the exception of the example applications. The operating system dependencies are isolated to a single operating system abstraction layer module so that any operating system that supports the functions in the abstraction layer can be used.

How are the IzoT example applications dependent on a Raspberry Pi?

The hardware for the example applications uses the GPIO and I2C interfaces on the Raspberry Pi. The examples are not required to use the IzoT Platform, and the examples all provide an option to run without attached hardware.

What kind of Raspberry Pi should I get to run the IzoT Platform on a Pi?

The IzoT stacks have been tested on the $35 Model B Raspberry Pi. For best performance, use a Samsung UHS-I 8GB (or larger) SD Card.

What operating system should I use for the Pi?

The IzoT stacks have been tested on Raspbian Linux which is a Debian wheezy port, optimized for the Raspberry Pi. You can get Raspbian Linux from www.raspberrypi.org/downloads. You can also install Raspbian Linux from the New Out Of Box Software (NOOBS) available from the same location.

Can I run both the IzoT Device Stack and the IzoT Server Stack on the same Pi?

Due to the overhead of the Web server and application framework used by the IzoT Server Stack, the IzoT Device Stack cannot be used on the same Pi.

What programming languages can I use to create my IzoT-enabled device applications?

You can use Python 3, C++, or C. The IzoT Python Package makes Python 3 the most productive option because the IzoT Python Package provides a much higher-level interface for

interfacing your application with an IzoT network. A low-level C API is available for C and C++ developers, but developing C and C++ IzoT applications is more difficult than developing Python 3 IzoT applications.

What programming languages can I use to create my IzoT-enabled Web applications?

You can use any programming language that supports generating REST requests and processing REST JSON, XML, or text responses. For example, you can use Javascript, Java, Python, Perl, Ruby, or any of the .NET languages. The example Web pages included with the IzoT Server Stack are written in HTML5 and Javascript with the jQuery library.

Is an IzoT server required in an IzoT network?

No. An IzoT network can consist solely of IzoT-enabled devices running the IzoT Device Stack, with no servers required. The IzoT-enabled devices can discover each other and establish peer-to-peer communication without a server. An IzoT server running the IzoT Server Stack is useful for providing a Web interface to the devices on the IzoT network.

Where can I get the sensors and LED controllers used by the IzoT example applications?

The sensors are available from Adafruit and the RGB Star LED Controllers are available from Solarbotics. Following is a summary of the modules used:

- Adafruit Model 166 Round Force-Sensitive Resistor
- Adafruit Model 189 Passive Infrared Sensor
- Adafruit Model 335 2.8” 320x240 18-bit Color TFT Touchscreen Display
- Adafruit Model 385 DHT22 Temperature-Humidity Sensor
- Adafruit Model 439 TSL2561 Digital Light Sensor
- Adafruit Model 815 16-Channel 12-Bit PWM Driver
- Adafruit Model 1083 ADS 1015 12-Bit 4-Channel ADC
- Solarbotics Model 60160 3 Watt RGB Star LED Controller

Source: http://iiot.echelon.com/support