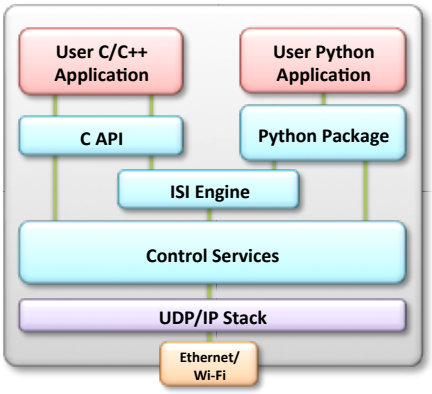
ECHELON



Internet-ready Control Networking

The IzoT Device Stack enables developers to build networks of communicating devices, as part of the Industrial Internet of Things, using a 32-bit processor and operating system as the processing engine for connected devices. With the IzoT Device Stack, your devices can exchange data with each other on an easy-to-use, publish-subscribe data model over IP. Your IzoT devices can collect data from physical sensors built to monitor things including temperature, humidity, light-level, power-consumption, or moisture, and make the data available to other IzoT devices within the community. Using data received from other IzoT devices or local sensors, your IzoT devices can also control physical actuators such as LED dimmers, motor controllers, damper controllers, and solenoids. And with the IzoT Server, you can build Web pages on which to monitor and control all of your IzoT devices simultaneously.

A reference implementation of the IzoT Device Stack is available for the popular Raspberry Pi. Reference example devices are also available for the Raspberry Pi.

Application Development with Python

The IzoT Device Stack includes the IzoT Python Package, built in the Python 3 programming language, which allows you to build device application software. The IzoT Python Package makes it easy for your Python applications to define their input and output datapoints in relation to other IzoT devices in local networks. Using the IzoT Python Package, your devices notify other devices about all the available datapoints and discover datapoints on other devices so no data slip through the cracks.

Device Communication with Datapoints

The IzoT Device Stack includes a rich set of data types and profiles that simplifies defining the network inputs, outputs, and properties for your IzoT applications. You can use the IzoT Analog Input profile to create analog sensors for tasks including power-consumption, temperature, humidity, or light-level monitoring. You can use the IzoT Load Control profile to design myriad load control devices such as LED controllers, lamp dimmers, fan controllers, sunblind actuators, or pool pump controllers. You also have the option to choose from one hundred other standard profiles to rapidly implement a wide variety of devices. IzoT profiles automatically define many useful input and output datapoints for sensors and actuators. They also define many useful network-accessible properties that you can use to configure your applications. The implementation of a typical profile takes no more than a single line of Python code.

Network Organization with Interoperable Self-Installation (ISI)

The power of the IzoT Platform comes from its ability to easily create communities of interacting devices. These communities form using the IzoT Interoperable Self-Installation (ISI) protocol implemented within the IzoT Device Stack. Every IzoT-capable device automatically joins the community on your local IP sub-network, all the while notifying all the devices in the community of possible datapoints to collect. When compatible IzoT device discover one another, they automatically reconfigure themselves to exchange data without requiring the use of any additional servers or tools.

IzoT Device Examples

Source code and hardware designs are included with the IzoT Device Stack for three example IzoT devices. The example devices are: a 2-channel, 3-color LED controller; 2.8" TFT touchscreen display, and an environmental sensor. These examples offer a platform for building your own applications and devices. You can build the example devices using Raspberry Pis and I/O hardware shown in the schematics, or you can adapt and port the examples to an alternative processor and operating system. You can also build the LED Controller using a Raspberry Pi and a PiFace I/O board, should you wish to avoid building new hardware.

LED Controller

The LED Controller is a 2-channel, 3-color 3W LED controller. It can be controlled from other IzoT devices, or from Web pages via an IzoT Server. The LED Controller accepts network input datapoint updates to turn the LEDs on or off, or to manage the brightness or color of the LEDs. The LED Controller also supports scenes, meaning a single scene input from a device like a Touch Keypad or Web page can appear as a unique pattern or sequence of colors and brightness supporting smooth transitions from one scene to the next. Used alongside the Touch Keypads and Environment Sensors, the LED Controller demonstrates a sophisticated multi-color LED lighting control system with scene and occupancy control.

Touch Keypad

The Touch Keypad is a 2.8" TFT touchscreen display controller that implements a multi-button keypad. This keypad can be used to control other IzoT devices like the LED Controller. The Touch Keypad displays touch buttons, which control the scene, light level, and color. When you touch a button on the Keypad, it generates a network output datapoint providing an update to connected LED Controllers. The compact touchscreen is ideal for use as a wall mounted lighting control keypad, with an easy-to-use touchscreen display.

Environment Sensor

The Environment Sensor is a multi-sensor with a passive-infrared occupancy sensor, temperature sensor, humidity sensor, and light-level sensor. Multiple Environment Sensors in an area work together to determine occupancy of the area based on inputs from all the sensors, and can be used to control the LED Controllers in the area based on area occupancy and light-level. Example Web pages included with the IzoT Server provide an example of how to monitor and configure sensors, and how to display charts that display sensor values over time.

Downloading the IzoT Device Stack

A beta release IzoT Device Stack is available as a free download from **http://iiot.echelon.com/get-started**. The download includes ready-to-run software for the Raspberry Pi, as well as full source code for the IzoT Device Stack. You can either modify the current source code for the Raspberry Pi, or port it to other compatible processors and operating systems. The production 1.0 software will be available in Q1 2014, in two versions: free (without support) and premium (with support and expanded rights).

Specifications

Reference Implementation Target Platform

Raspberry Pi Model B with 512 MB RAM and Raspbian Linux
The source code may be ported to other compatible processors running Linux, Microsoft Windows, or other operating systems with POSIX services

Reference ImplementationTarget Flash Memory

8 GB minimum SD flash card

Available Application Development Languages

Python 3.2, C++, and C

ECHELON®