March 04, 2019

Internet of Things Device with Controllable Lighting Elements

Melissa Daniels

Rachel Weinstein Petterson

Carson Holgate

Aieswarya Sayee Manicka

Internet of Things Device with Controllable Lighting Elements

Inventors: Melissa Daniels, Rachel Weinstein Petterson,
Carson Holgate, and Aieswarya Sayee Manicka

Summary

Aspects of the present disclosure are directed to a device with controllable lighting elements or other output elements such as audio, vibratory, gaseous/liquid, and/or olfactory output elements. In particular, the interactive device of the present disclosure can include a structure fashioned in the shape of some fauna or flora (e.g., a tree) that can selectively operate the output elements (e.g., illuminate the lighting elements) based on collected sensor data regarding the surrounding environment. The structure can also respond to specific user interactions and/or receive predetermined control signals that control the output elements.

Thus, in one example, the device can include a tree-like framework that includes lighting elements such as light emitting diodes (LEDs). The lighting elements can be illuminated or dimmed to achieve a variety of effects including pulsing or blinking in a manner that captures the attention of onlookers. The illuminating device can be appropriately shaped and colored to more closely appear like an actual tree and can include a structure that is fashioned to include artificial ornamentation that resembles a trunk, branches, or leaves (e.g., a metallic trunk with plastic branches and leaves).

Further, devices of the present disclosure can include various sensors that detect the state of the area around the illuminating device. For example, the illuminating device can be configured to respond to sound, light, pressure, and various gestures performed by onlookers. The sensors can include force resistive sensors, light sensors, audio sensors, and/or sensors that can detect specific gestures (e.g., using cameras and/or motion sensors).

Thus, aspects of the present disclosure are directed to fashioning an artificial device into the shape of a natural object (e.g., a tree) and controlling lighting elements on the device via some combination of predetermined signals generated by a microcontroller and/or sensors on the device that respond to user input or other environmental input.

The illuminating device can stand on its own or be mounted on some other surface (e.g., a wall) to enhance its visibility. As such, the illuminating device can perform the functions of an interactive art installation that leverages sensors and various Internet of Things platforms to create a visually impressive light show.

Example Figures

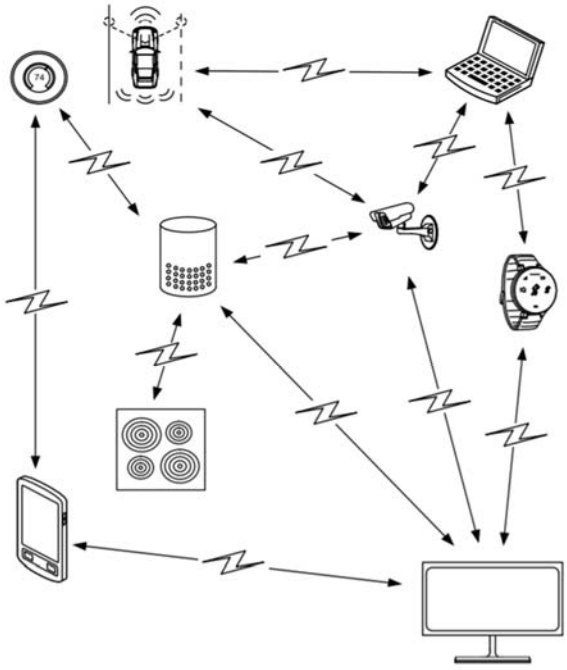

Fig. 1

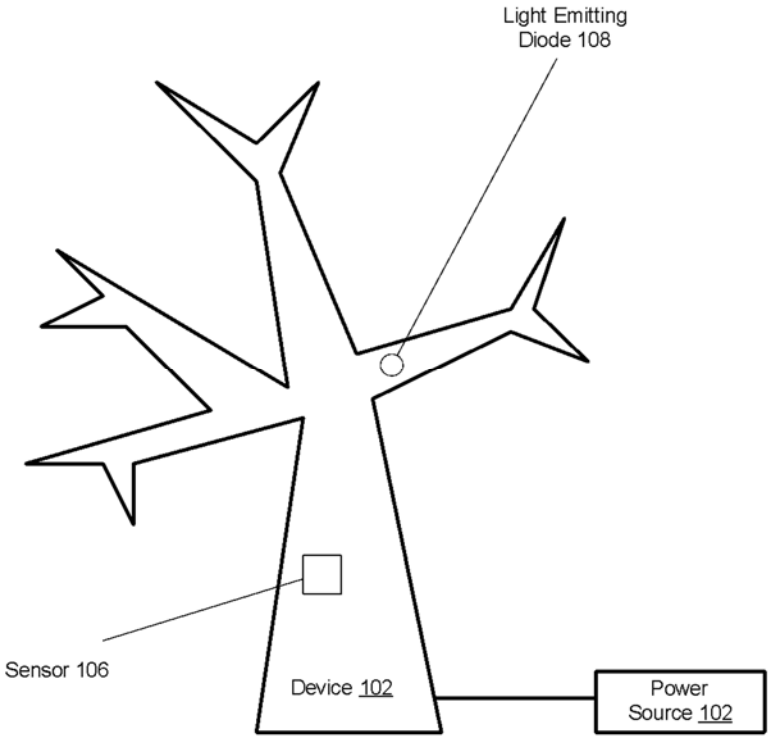

Fig. 3

Detailed Description

The present disclosure is directed to device that includes lighting elements that can be configured to activate in response to control signals which can be programmed or the result of sensor interactions. Further, the illuminating device can operate as an Internet of Things (IoT) device within an IoT environment.

Referring to Fig. 3, which is included in this document, example embodiments of illuminating devices are described herein. The illuminating device (e.g., the device 102) can include lighting elements (e.g., the Light Emitting Diode 108) that can produce light based on control signals sent to the respective lighting element. For example, the lighting elements can use electrical power supplied by a power source (e.g., a wall outlet or a battery such as the power source 102) and produce white light from some portion of the lighting elements of the illuminating device. Fig. 4 provides photographs of an example embodiment of a tree-shaped interactive lighting device as described herein.

The illuminating device can include a computing device (e.g., an i.MX 7 Dual Processor) that uses an operating system that can operate on embedded hardware of the illuminating device (e.g., Android Things). Further, the illuminating device can include a communications interface that can send or receive control signals that can activate and adjust the brightness and/or color of lighting elements (e.g., the Light Emitting Diode 108) of the illuminating device. For example, the illuminating device can receive signals that control lighting elements of the illuminating device via the Internet.

In some embodiments, the illuminating device can include sensors, including the sensor 106, that can detect the environment around the illuminating device and send signals that can be used to produce various lighting effects. For example, the illuminating device can include microphones that detect ambient noise levels (e.g., noise produced by people talking and moving through the environment) around the illuminating device. The illuminating device can then use the ambient noise levels (and/or some value associated with aggregate noise levels) to generate various lighting effects on the device by brightening, dimming, or selectively activating lighting elements on the device. For example, LEDs can be illuminated more brightly at localized areas of louder ambient noise. Other output effects can be performed as well (e.g., simulated raining and/or birds chirping).

By way of further example, the device can include tactile sensors (e.g., capacitive sensors and/or force resistive sensors) that can detect a human touch. Based on detection of human touch, the device can generate lighting effects by activating lighting elements proximate the tactile sensor that was touched. Other output effects can be performed as well (e.g., simulated raining and/or birds chirping).

In some embodiments, the device can activate subsets of the lighting elements at different times in order to achieve different visual effects. For example, the illuminating device can brighten and dim the lighting elements at a rate similar to a human respiratory rate (e.g., brighten and dim the lighting elements fifteen times per minute) or heart rates (e.g., brighten and dim the lighting elements sixty times per minute).

In some embodiments, the illuminating device can include sections that can accommodate fauna or flora (e.g., moss). For example, the illuminating device can include fauna or flora accommodating sections that are separate from electronic components of the illuminating. The fauna or flora accommodating sections of the illuminating can include grooves or indentations to better hold fauna or flora in place. Further, the fauna or flora accommodating sections can be colored in accordance with the color of the fauna or flora (e.g., green or brown) in keeping with a naturalistic theme.

In some embodiments, the illuminating device will listen for data from associated sensors (e.g., sensors on the 'trunk' portion of the illuminating device). The illuminating device can then aggregate that data into some floating point value that will trigger different kinds of animations and lights (or other output effects) for different thresholds. For example, the proportion of the illuminating elements that are illuminated can be based on thresholds associated with the amount of pressure that is placed on tactile sensors of the illuminating device or the number of times that the illuminating device is touched within a predetermined time period (e.g., the number of touches per minute).

In some embodiments, the base state of the illuminating device can include illuminating small numbers of the lighting elements to give the appearance of fireflies flitting about the illuminating device. Further, in response to some threshold level of inputs to the sensors being

achieved, the illuminating device can trigger a more elaborate or bright spectacle of illumination. For example, the lighting elements can appear to explode in the manner of fireworks once some threshold level of user input (e.g., a noise level exceeding a predetermined number of decibels) has been achieved.

In some embodiments, the tree Quercus virginiana can be a good basis for the tree-like illuminating device. Also known as the Southern Live Oak, Quercus virginiana, is native to the southern US, particularly Florida. The trees live for centuries and can be anywhere from smallish shrub shaped to large sprawling behemoths. The tree has plenty of low hanging and even ground supported branches that would make good attachment points for interactive elements. In addition, these trees often support parasitic plants like moss, ferns, and mistletoe that we can use to strategically hide exposed electronics. In another example, the device can be shaped to mimic a Joshua tree.

The design can be scaled back to meet any space or financial requirements by cutting it in half and putting it against a wall, cutting it into quarters and putting the trunk of the tree in a corner, and/or scaling the whole thing down to a more manageable size.

There are many options for lighting a large tree. For example, Android Things has an existing driver to handle apa102 light strips. The apa102 light strips, also known as DotStar LED strips, are strips of individually addressable RGB LEDs. They can be attached together to build arbitrarily long strips but most are sold in 5 meter lengths.

In some embodiments, each 5 meter strip requires a 5V/10A power supply and clock and SPI lines to drive it. As one example, the i.MX7D board has 4 SPI lines. Multiple strips can theoretically share the same clock signal so each board can power four, five meter strands.

In one specific example embodiment, the tree has four boards powering the lighting on the tree, with each board controlling four, five meter strands of LEDs for a grand total of 80 meters of lights. Each board can run the same software and may either respond to input just from its quadrant and/or also take commands from some master, aggregate input. As an example, half of the strands can start at the base of the trunk and run up as far as they'll go and the other half can start at the top of the tree and wrap through the branches. Each board and each strand will typically need its own power supply. This means we'll need at least 20 outlets to run the example tree described above.

In one specific example embodiment, the tree can have 2-3 low-hanging or ground-supported branches spread out evenly. The rest of the branches can branch off at at least about 6' height so that the average human being can walk under them and get to the trunk. This will also be helpful for setup and troubleshooting.

As an example, a 6' trunk gives about 10 feet of remaining LED strip to work with for the bottom of the trunk based strands. Thus, in some embodiments, the branches can reach out no further than 10 feet from the trunk to accommodate this. Thus, in a specific example, a roughly 20x20 foot area is needed to install the tree. Around the tree, some kind of ground cover (e.g., AstroTurf) can be provided to cover the power lines that run to the sensors.

In some embodiments, the software operations can be performed by a Firebase real time database, which can easily be plugged in to leverage predefined libraries. One data structure is as follows: Structure: Sensors, <sensor id>, Value: [0, 1]. Each sensor application can be responsible for knowing its corresponding ID and turning its inputs into a value between zero and one inclusive. Writing that value to the database will send an update to the tree board that is listening and it will react accordingly. Cloud Functions can be used on the backends to build

aggregate scores which can be sent to the boards controlling the light show. In such fashion, the tolerances during the day can be adjusted (e.g., if it is too noisy, then the sensor ranges can be dynamically adjusted).

Referring now to Figure 1, Figure 1 depicts a block diagram of an example IoT environment according to example implementations of the present disclosure. As illustrated in Figure 1, in some implementations, the IoT environment includes a plurality of different devices, each of which can be referred to as an IoT device. An example IoT device can be an intelligent, environmentally-sensing, and/or network-connected device configured to communicate with a central server or cloud service, a control device, and/or one or more additional IoT devices to perform any number of operations (e.g., in response to received commands). IoT devices can, in some instances, also be referred to as or include "smart" devices and/or "connected" devices.

Each IoT device can be a stand-alone physical device or can, in some instances, be an embedded device that is embedded within a larger device or system. Each IoT device can include electronics, software, sensors, actuators, and/or other components, including various components that sense, measure, control, and/or otherwise interact with the physical world. An IoT device can further include various components (e.g., a network interface, antennas, receivers, and/or the like) that enable the IoT device to send and/or receive data or other information from one or more other IoT devices and/or to a central system.

In particular, the various IoT devices can be communicatively connected to one another via one or more communications networks. The networks can be wide area networks, local area networks, personal area networks, piconets, cellular networks, other forms of networks, and/or combinations thereof. The networks can be wired and/or wireless. The networks can be private and/or public. As examples, two or more of the IoT devices can communicate with one another

using a Wi-Fi network (e.g., a home network), Bluetooth, Bluetooth Low Energy, Zigbee, Radio-Frequency Identification (RFID), machine to machine connections, inductive communications, optical communications, infrared communications, other communications techniques or protocols, and/or combinations thereof. For example, an IoT device might communicatively connect with a first nearby device using Bluetooth while also communicatively connecting with a second nearby device using Wi-Fi.

In some implementations, each IoT device can have a unique identifier. For example, the identifier for each IoT device can include and/or be based on an Internet Protocol address associated with such IoT device, a manufacturer associated with such IoT device, a location at which such IoT device is positioned, a model number of such IoT device, a functionality of such IoT device, and/or other device characteristics. In some implementations, a given IoT device can locate and/or communicate with another IoT device based on its unique identifier. In some implementations, the identifier assigned to an IoT device can be modified by a user and/or owner of such IoT device.

In particular, in some implementations, a user can assign one or more identifiers to the IoT devices within a device topology representation. The device topology representation can describe and/or organize a group of IoT devices (e.g., based on location with one or more structures such as one or more homes, offices, vehicles, and/or the like). The identifiers can be chosen by the user and associated with the respective IoT devices within the device topology representation. The identifier(s) can include but are not limited to names, nicknames, and/or aliases selected for the IoT devices by the user. In this manner, the identifiers can be names or aliases of the respective IoT devices that the user is likely to use when identifying the IoT devices for requested control or command operations (e.g., “turn on the kitchen lights”).

An IoT device can be mobile or can be stationary. In some implementations, an IoT device can be capable of autonomous or semi-autonomous motion.

In some implementations, an IoT device can be controlled or perform operations in response to communications received by the IoT device over a network. For example, an IoT device can be controlled by a control device that is communicatively connected to the IoT device. The control device can communicate directly with the IoT device or can communicate indirectly with the IoT device (e.g., over or using a mesh network). The control device can itself be an IoT device or the control device can be a device that is not considered part of the IoT environment. For example, the control device can be a server device that operates as part of cloud computing system. The commands can be in response to or generated based on a user input or can be generated without user input.

Thus, in one example, an IoT device can receive communications from a control device and the IoT device can perform operations in response to receipt of such communications. The performed operations can be internal operations (e.g., changing an internal setting or behavior) or external operations (e.g., interacting with the physical world in some way). The IoT device and the control device can be co-located or can be remotely located from each other.

As an example, the control device can be or include a user device such as a smartphone, tablet, computing device that is able to be worn, laptop, gaming console or device, virtual or augmented reality headset, and/or the like. As another example, the control device can be a server computing device. As another example, the control device can itself be an IoT device. For example, the control device can be a so-called “smart speaker” or other home control or automation device.

In some implementations, a user can interact with a control device (e.g., which can be an IoT device) to input data into or otherwise control the IoT environment. For example, the control device can include and execute a software application and/or other software programs that provide a user interface that enables entry of user input. The software applications can be executed entirely at the control device or can be web applications where some portion of the program or functionality is executed remotely (e.g., by a server connected over the Internet) and, in some implementations, the client-side logic runs in a web browser. Thus, in some implementations, a web server capable of sending, receiving, processing, and storing web pages or other information may be utilized.

In some implementations, a cloud service may be used to provision or administer the IoT devices. For example, a cloud computing system can enable or perform managed and/or integrated services that allow users to easily and securely connect, manage, and ingest IoT data from globally dispersed IoT devices at a large scale, process and analyze/visualize that data in real time, and/or implement operational changes and take actions as needed. In particular, in some implementations, the cloud computing system can employ a publication subscription model and can aggregate dispersed device data into a single global system that integrates seamlessly with data analytics services. An IoT data stream can be used for advanced analytics, visualizations, machine learning, and more to help users improve operational efficiency, anticipate problems, and/or build rich models that better describe and optimize the user's home or business. The cloud system can enable any number of dispersed IoT device to connect through protocol endpoints that use automatic load balancing and horizontal scaling to ensure smooth data ingestion under any condition.

In some implementations, the cloud system can include or implement a device manager. For example, the device manager can allow individual IoT devices to be configured and managed securely in a fine- or coarse-grained way. Management can be done through a console or programmatically. The device manager can establish the identity of a device and can provide the mechanism for authenticating a device when connecting. The device manager can also maintain a logical configuration of each device and can be used to remotely control the device from the cloud.

In some implementations, an IoT device can include an artificial intelligence-based assistant or software agent. A user can interact with the artificial intelligence-based assistant via a control device, directly through the IoT device, or any other method of interaction. The artificial intelligence-based assistant can perform tasks or services based on user input and/or contextual awareness (e.g., location awareness), including acting as a control device to control other IoT devices. In some implementations, an IoT device (e.g., an artificial intelligence-based assistant on such device) can access information from a variety of online sources (such as weather conditions, traffic congestion, news, stock prices, user schedules, retail prices, etc.).

The artificial intelligence-based assistant or software agent can be stored and implemented by a single device (e.g., a single IoT device) or can be spread across multiple devices and implemented by some (e.g., dynamically changing) combination of such multiple devices.

In some implementations, an IoT device can include (e.g., as part of an artificial intelligence-based assistant) one or more machine-learned models that assist in understanding user commands, determining context, and/or other actions. Example machine-learned models include artificial neural networks such as feed-forward neural networks, recurrent neural

networks, convolutional neural networks, autoencoders, generative adversarial networks, and/or other forms, structures, or arrangements of neural networks. Additional example machine-learned models include regression models, decision tree-based models (e.g., random forests), Bayesian models, clustering models, linear models, non-linear models, and/or other forms, structures, or arrangements of machine-learned models. Machine-learned models can be trained using supervised learning techniques or unsupervised learning techniques. Machine-learned models can be stored and implemented on the IoT device or can be stored and implemented in the cloud and the IoT device can leverage the models by communicating with cloud devices. Feedback or other forms of observed outcomes can be used to re-train models to improve their performance. Models can be personalized to one or more users or environments by re-training on data specific to such users or environments.

In some implementations, the artificial intelligence-based assistant can perform concierge-type tasks such as, for example, making dinner reservations, purchasing event tickets, making travel arrangements, and/or the like. In some implementations, the artificial intelligence-based assistant can provide information based on voice input or commands (e.g., by accessing information from online sources). In some implementations, the artificial intelligence-based assistant can automatically perform management or data-handling tasks based on online information and events, including, in some instances, without user initiation or interaction.

In some implementations, a control device (e.g., which may be an IoT device) can include components such as a mouse, a keyboard, buttons, knobs, a touch-sensitive component (e.g., touch-sensitive display screen or touch pad), and/or the like to receive input from the user via physical interaction.

In some implementations, the control device can include one or more microphones to capture audio signals and the device can process the audio signals to comprehend and respond to audio commands (e.g., voice commands) provided by a user or by some other device. Thus, in some implementations, the IoT devices can be controlled based on voice commands from a user. For instance, a vocalization from a user can be received by a control device. The vocalization can be a command spoken by a user proximate to the control device. The control device can control itself and/or one or more of the IoT devices based on the vocalization.

In some implementations, one or more vocalization(s) may be used as an interface between a user and an artificial intelligence-based assistant. For example, a user may vocalize a command which the artificial intelligence-based assistant may identify, process, and/or execute or cause to be executed. The vocalized command may be directed at the artificial intelligence-based assistant.

As one example, the vocalization may indicate a user desire to interact with or control another IoT device (e.g., lowering a thermostat setting, locking a door, turning off a light, increasing volume, etc.). The artificial intelligence-based assistant may communicate the command to the desired IoT device which can respond by executing or otherwise effectuating the user command. As another example, the vocalization can include a user commanding the artificial intelligence based assistant to perform a task (e.g., input an event into a calendar, retrieve information, set a reminder, make a list, define a word, read the first result of an internet search, etc.).

In some implementations, speech recognition or processing (e.g., natural language processing) can be performed on the vocalization to comprehend the command provided by the vocalization. For instance, data indicative of the vocalization can be provided to one or more

language models (e.g., machine-learned models) to determine a transcription of or otherwise process the vocalization.

In some implementations, processing the vocalization or other user input can include determining one or more IoT devices to control and/or one or more actions to be performed by the selected IoT devices. For instance, a semantic interpretation of the vocalization (e.g., a transcript of the vocalization) can be determined using one or more semantic interpretation techniques (e.g., natural language processing techniques). The semantic interpretation can provide a representation of the conceptual meaning of the vocalization, thereby also providing an interpretation of the intent of the user.

In some implementations, the interpretation of the vocalization can be determined based at least in part on the device topology representation. For instance, the device topology representation can be accessed to determine the one or more selected IoT devices and/or actions to be performed. As one example, the device topology representation can be accessed and compared against a transcription of the vocalization to determine a match between one or more terms included in the transcription and one or more terms associated with the IoT device topology representation (e.g., “kitchen lights”). In some implementation, the identity of the speaker can be ascertained and used to process the vocalization (e.g., such as to process commands that include possessive modifiers: “brew a cup of my favorite roast of coffee”).

In some implementations, the control device (e.g., which may be an IoT device) can include a vision system that includes one or more image sensors (e.g., visible-spectrum cameras, infrared cameras, LIDAR systems, and/or the like) that capture imagery. The device can process the imagery to comprehend and respond to image-based commands or other input such as, for example, gesture commands provided by the user. In some implementations, the vision system

may incorporate or perform facial movement identification (e.g. lip reading) capabilities while, in other implementations, the vision system may additionally or alternatively incorporate hand shape (e.g. hand gestures, sign language, etc.) identification capabilities. Facial movement and/or hand shape identification capabilities may allow a user to give commands a control device in addition or alternatively to voice control.

In some implementations, in response to the image data of the facial or hand gesture, the control device can determine one or more IoT devices to control and/or one or more actions to be performed (e.g., by the selected IoT devices). Interpretation of image data that depicts lip reading and/or sign language may be achieved through any method of image data analysis. The interpretation can provide a representation of the conceptual meaning of the image data. In this manner, the interpretation of the image data can provide an interpretation of the intent of the user in performing the gesture(s).

In some implementations, the interpretation can be determined based at least in part on the device topology representation. For instance, the device topology representation can be accessed to determine the one or more selected IoT devices and/or the action to be performed. For example, the device topology representation can be accessed and compared against the image data to determine a match between one or more aspects of the image data and one or more aspects associated with the IoT device topology representation (e.g., the user may be pointing to a specific IoT device when providing a voice command or a gesture command).

In further implementations, gaze recognition can be performed on the captured imagery to identify an object or device that is the subject of a gaze of the user. A user command (e.g., voice or gesture) can be interpreted in light of (e.g., as applied to) the object or device that is the subject of the gaze of the user.

In some implementations, the vision system may be used as an interface between a user and an artificial intelligence-based assistant. The captured image data may be interpreted and/or recognized by the artificial intelligence-based assistant.

In some implementations, the selected IoT devices and/or the actions to be performed can be determined based at least in part on contextual data (e.g., location of user, day of the week, user data history, historical usage or command patterns, user wardrobe, etc.) For instance, in response to receiving a command from a user, a location of the user, a time of day, one or more past commands, and/or other contextual information can be determined. The location can be determined using various suitable location determination techniques. The location determination technique can, for example, be determined based at least in part on the control device to which the user provides the vocalization.

As one example, if the control device is an IoT device that is specified in the device topology representation, the user location can be mapped to the structure and/or room to which the control device is assigned in the device topology representation. As another example, if the control device is a user device not specified in the device topology representation, the user location can be determined using one or more location determination techniques, such as techniques using wireless access points or short range beacon devices associated with one or more IoT devices, and/or other suitable location determination techniques. In some implementations, the user location can be mapped to one or more structures and/or rooms specified in the device topology representation. In some implementations, the control device and/or other IoT devices can also process audio signals and/or imagery to comprehend and respond to contextual information. As examples, triangulation and/or beamforming techniques can be used to determine the location of the user based on receipt of the voice command at

multiple different audio sensors. In some implementations, multiple possible user commands or requests can be disambiguated based on the contextual information.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, devices, or features described herein may enable collection of user information (e.g., information about a user's preferences, a user's activities, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

Figure 2 provides a block diagram of an example software stack that can be included on an IoT device. The software stack shown in Figure 2 is provided as one example only. Various different IoT devices can have any number of different software and/or hardware configurations which may be of greater or lesser complexity to that shown in Figure 2.

In some implementations, an IoT device can include and execute one or more computer applications (also known as software applications) or other computing programs. The IoT device can execute the application(s) to perform various functions, including collection of data, communication of data, and/or responding to or fulfilling received commands. In some implementations, the software applications can be native applications.

In some implementations, the software application(s) on an IoT device can be downloaded and installed by or at the direction of the user. In other implementations, the software application(s) can be default applications that come pre-programmed onto the IoT device. In some implementations, the software application(s) can be periodically updated (e.g., via download of update packages). The software application(s) can be closed source applications or can be open source applications. The software applications can be stand-alone applications or can be part of an operating system of the IoT device. The software applications can be embodied in computer-readable code or instructions that are stored in memory and then accessed and executed or followed by one or more processors of the IoT device.

In some implementations, the software application(s) on an IoT device can be user-facing applications such as a launcher or a browser. In other implementations, the IoT device does not include any user-facing applications but, for example, is instead designed to boot directly into applications developed specifically for the device.

More particularly, in some implementations, an IoT device can include or otherwise be implemented upon or in conjunction with an IoT platform that includes a number of elements. The IoT platform can include an operating system. The operating system can, for example, have been optimized for use in the IoT environments (tuned for faster boot times and/or lower memory footprint). The operating system and other platform elements may be able to receive secure and managed updates from the platform operator. The IoT platform can include hardware that is accessible and easy to integrate.

The IoT platform can also enable application developers to build applications using a rich framework provided by an operating system software development kit (SDK) and platform services, including, for example, the same user interface toolkit, multimedia support, and

connectivity application programming interfaces (APIs) used by developers of mobile applications for larger devices such as smartphones. Applications developed for the IoT device can integrate with various services using one or more client libraries. For example, the applications can use the libraries to interact with services such as application deployment and monitoring services, machine learning training and inference services, and/or cloud storage services. The applications can use the APIs and/or support libraries to better integrate with hardware, including, for example, custom hardware. This can include support for peripheral interfaces and device management. The device can also include a number of native libraries, including, for example, C/C++ libraries, runtime libraries, core libraries, and/or the like. Updates to one or more of these components can be deployed over the air and/or automatically when updates are available.

In some implementations, an IoT device (e.g., the software applications executed thereby) can utilize APIs for communicating between a multitude of different software applications, operating systems, databases, libraries, enterprises, graphic interfaces, or any other component of the IoT environment disclosed herein. For instance, a first software application executed on a first IoT device can invoke a second software application via an API call to launch the second software application on a second IoT device.

In some implementations, the applications can run on a single or variety of operating system platforms including but not limited to OS X, WINDOWS, UNIX, IOS, ANDROID, SYMBIAN, LINUX, or embedded operating systems such as VxWorks.

The IoT device can include one or more processors and a memory. The one or more processors can be any suitable processing device (e.g., a processor core, a microprocessor, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), System

on a Chip (SoC), a controller, a microcontroller, etc.) and can be one processor or a plurality of processors that are operatively connected. The memory can include one or more non-transitory computer-readable storage mediums, such as RAM, ROM, EEPROM, EPROM, flash memory devices, magnetic disks, etc., and combinations thereof. The memory can store data and instructions which are executed by the processor to cause the IoT device to perform operations. The IoT devices can, in some instances, include various other hardware components as well, including, for example, a communications interface to enable communication over any number of networks or protocols, sensors, and/or other components.

In some implementations, the IoT device can include or be constructed using one or more System on Module (SoM) architectures. Each SoM can be a fully integrated component that can be dropped directly into a final design. Modules can be manufactured separately and combined to form the device. In some implementations, the device software can include a hardware abstraction layer and kernel which may be packaged as a board support package for the modules. In other implementations, different, non-modular architectures can be used.

Example IoT devices include or can be associated with an air conditioning or HVAC system, lighting device, a television or other home theater or entertainment system, security system, automatic door or window locking system, thermostat device, home energy manager, home automation system, audio speaker, camera device, treadmill, weight scale, smart bed, irrigation system, garage door opener, appliance (e.g., refrigerator, dishwasher, hot water heater, furnace, stove, fireplace, etc.), baby monitor, fire alarm, smoke alarm, medical devices, livestock tracking devices, cameras, beacon devices, a phone (e.g., smartphone), a computerized watch (e.g., a smart watch), a fitness tracker, computerized eyewear, computerized headwear (e.g., a head mounted display such as a virtual reality of augmented reality display), other types of

computing devices that are able to be worn, a tablet, a personal digital assistant (PDA), a laptop computer, a desktop computer, a gaming system, console, or controller, a media player, a remote control, utility meter, an electronic book reader, a navigation system, a vehicle (e.g., car, boat, or plane/drone) or embedded vehicle system, an environmental, food, or pathogen monitor, search and rescue devices, a traffic control device (e.g., traffic light), traffic monitor, climate (e.g., temperature, humidity, brightness, etc.) sensor, agricultural machinery and/or sensors, factory controller, GPS receivers, printers, motor (e.g., electric motor), and/or other suitable device or system.

The technology discussed herein makes reference to servers, databases, software applications, and other computer-based systems, as well as actions taken and information sent to and from such systems. One of ordinary skill in the art will recognize that the inherent flexibility of computer-based systems allows for a great variety of possible configurations, combinations, and divisions of tasks and functionality between and among components. For instance, server processes discussed herein may be implemented using a single server or multiple servers working in combination. Databases and applications may be implemented on a single system or distributed across multiple systems. Distributed components may operate sequentially or in parallel.

Figures

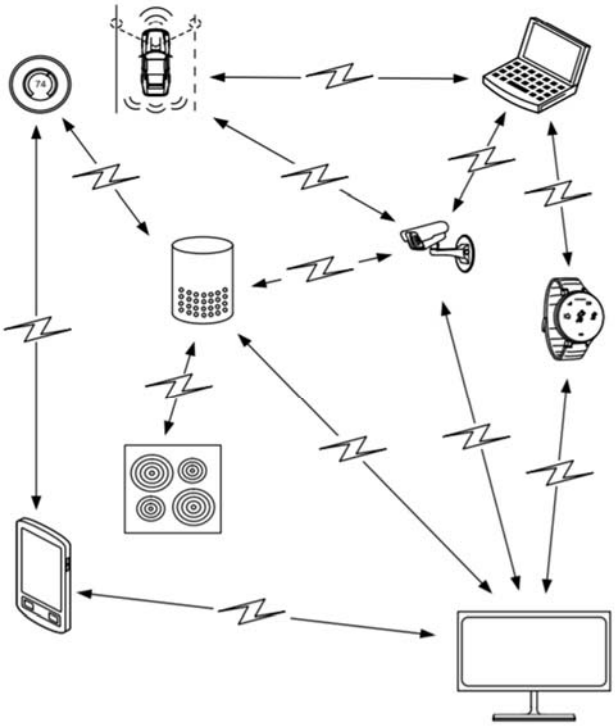

Fig. 1

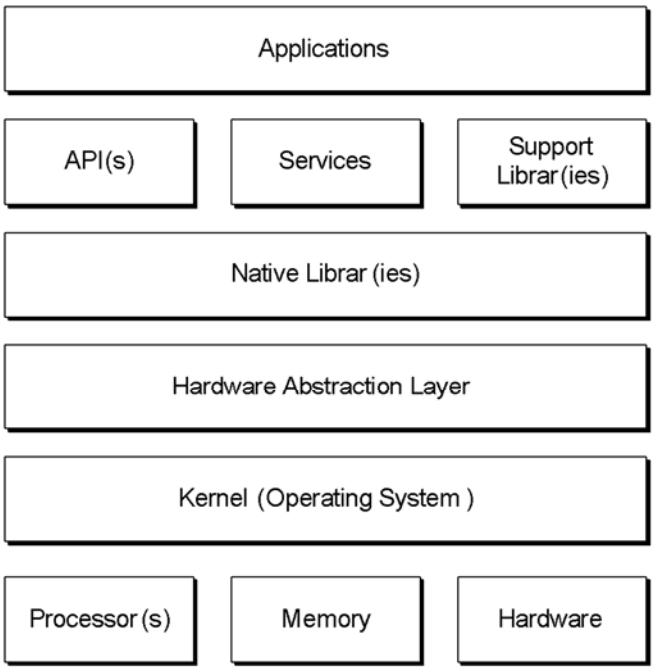

Fig. 2

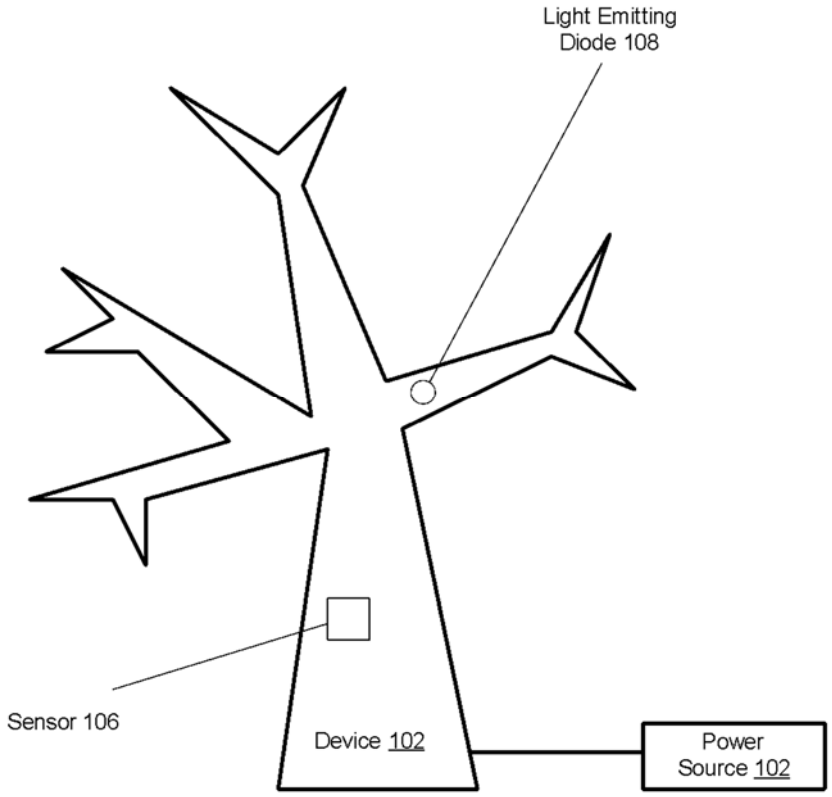
Light Emitting
Diode 108
Sensor 106
Device 102
Power
Source 102

Fig. 3


androidthings

Fig. 4