WHITE PAPER

IoT Firmware for Connected Devices – Considerations & Challenges

Central to the tremendous growth in Internet of Things (IoT) are the Things, which are the devices we use every day, and the data that they provide. Most Things have limited capability and need to be designed keeping in mind their use, location, accessibility, connectivity and power availability.

The key device design considerations comprise the processor type, bare metal or OS, firmware footprint and updates, connectivity, security and power. IoT firmware implementation needs to address critical challenges such as extending the battery life, incorporating robust data encryption, supporting communication protocols and connectivity options, and ensuring reliable operations.

THINXTREAM®

Introduction

In the era of Internet of Things (IoT), approximately 500 million Things or devices are being connected to the Internet every month, and that number is growing. While the benefits of being connected are tremendous, so are the challenges and risks.

A good percentage of these connected devices (or IoT devices, and we will use these terms interchangeably) have very specific functionality and are constrained devices, with limited capability. They need to be carefully designed keeping in mind their use, location, accessibility, connectivity and power requirements. In this white paper, we examine the important design considerations and challenges involved in the development of embedded, IoT firmware.

IoT Device Architecture

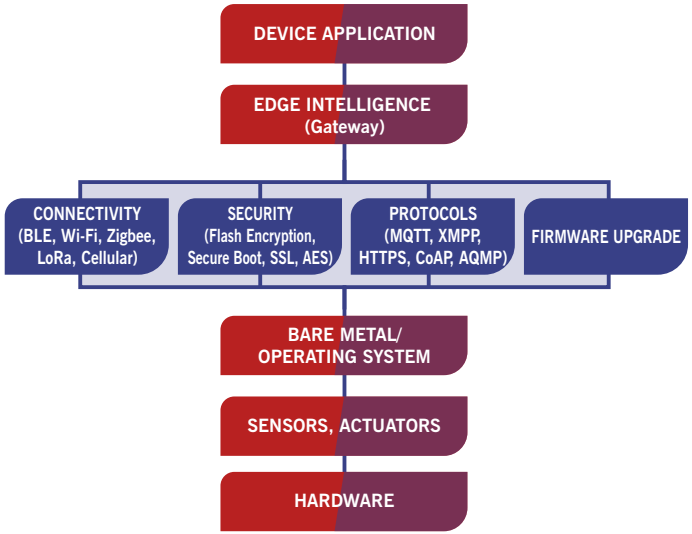

Figure 1. Typical Architecture of an IoT Device

At a generic level, IoT devices have sensors and actuators connected to them. The sensors provide the measurements to the device, which then formats or transforms them prior to sending them to gateways or directly to the Internet. IoT devices can also receive commands through the Internet from a master controller and perform certain actions using the actuators.

Selecting the appropriate hardware, connectivity and protocols is critical to the functioning of an IoT device. The connected device needs to have the capability to securely store the firmware binaries, device information and sensor data. It also needs to be able to transfer the data to other devices or the cloud through secure communication channels.

Design Considerations

Device Hardware Selection: This is a critical decision in the design and implementation of an IoT device. Does it require a microprocessor? Or, is a microcontroller sufficient for its needs? Microprocessors are usually chosen where the device requires the capability for multiple functionalities. Using a microprocessor also warrants the use of an operating system (OS), which makes it easier to implement solutions. The OS takes care of all the book-keeping needs of the device and we don't have to reinvent the wheel. On the other hand, microcontrollers consume less power and are mostly used to implement a specific functionality. Using a microcontroller with a Bare Metal design, we can design and implement a much smaller set of features that meets the device requirements, without over-designing. In addition to the main controller, the selection of the connectivity modules, sensors and actuators are also important as they determine the overall power consumption.

Operating System vs. Bare Metal: This is another critical aspect, which needs to be considered before implementation of IoT firmware. Having an OS increases firmware footprint, but enables faster development of device functionality. Using Bare Metal programming, we can reduce the firmware footprint without adding OS overheads. However, the time required for implementation

increases. Microcontroller chip vendors have been making it easier by providing ready-to-use libraries that can be integrated with the main application. These libraries also comprise Hardware Abstraction Libraries, which offer good levels of abstraction, so that firmware developers need not be aware of the inner workings of the microcontroller.

Firmware Footprint: Footprint for IoT firmware needs to be kept to a minimum, since RAM is very limited in such devices. Typically, around 64 KB of RAM is available to execute the device application. Flash memory has replaced traditional ROMs. Low-cost microcontrollers offer flash memory of more than 2 MB. This needs to be considered in case data collected by the IoT device needs to be temporarily stored prior to transmission or for local intelligence.

Connectivity: This is key to develop any device that connects to the Internet. Without connectivity, there is no IoT. Connectivity could be within a Body Area Network or a Personal Area Network, Local Area Network or a Wide Area Network. There are several protocols available in each of these networks with different benefits and challenges. Also, connecting to different networks have different power requirements, which need to be considered. For instance, for Wide

Area Networks, devices using protocols like LoRa consume lesser power than 2G. For any connectivity, we need to select modules that have low power consumption.

Power Requirements: This varies based on the device functionality. Most devices today run on batteries. Connected devices in remote locations, where power supply is not available, have no option but to rely on batteries. Typical batteries offer power from 12V downwards. Some of the devices which are "Configure and Forget", need to run on batteries that should last for at least a year. This puts a great emphasis on the power-saving features of the IoT device. It should use only as much power as it requires and not more. When it is not in active use, it needs to sleep ensuring that the power usage is minimal. In many cases, the device may need to be switched off and woken up only when an event occurs.

Firmware Update: It is the ability of the IoT firmware to be upgraded in the field. As the devices are connected wirelessly, it has to be done over-the-air (OTA) using low power consumption and low data rate transmission with protocols such as 802.15.4 and ZigBee. This is a critical functionality that cannot be overemphasized. Many OEM manufacturers realize this very late only

when there is an issue in the field. Chip vendors provide defect fixes for libraries supplied by them, device functionality may have a defect that needs to be fixed and device functionality may need to be upgraded. These offer more value to users, who need not replace a device to get a new feature.

Data Security: IoT security has become one of the most important aspects of design for IoT firmware. As the number of such devices increase exponentially, so does the base for security attacks. Imagine a situation where a hacker can gain control of a connected device at a user's facility. Any compromise provides opportunities for malicious elements to obtain control or tamper with the device, sniff sensitive information and cause unintended device behavior. Hence, firmware developers need to focus on secure design and implementation, and encrypted connection and storage. Even for very low-cost connected devices costing a few dollars, IoT security is key and should not be ignored.

Communication: Firmware developers needs to consider the device application from a communication perspective. Many connected devices produce massive amounts of data continuously. Sending it and getting it processed in the cloud could lead to latency in responding

to issues, huge bandwidth requirements and high costs. It is also likely that we cannot rely on network connectivity in certain device locations, which may be intermittent, with the connected device going offline resulting in hazardous situations. Where decision making needs to be quick, we can consider designing connected devices with Edge Intelligence within the network, where they can process the data locally and take decisions faster without connecting to the cloud every time. This also helps in filtering the data required to be sent to the cloud, resulting in lower bandwidth needs and costs.

Implementation Challenges

Battery Life

Connected devices need to provide real time capability as demanded by the application. However, most connected devices are powered by batteries and are expected to last on a single charge of battery for a long time. Battery drain is a cause of customer dissatisfaction and could result in product returns.

There are several factors that could reduce battery life, which the designer needs to consider and use methods to reduce power consumption.

- **High Clock Frequency:** Power is used during the leading and trailing edge of clock cycles. So, the faster the clock that the processor runs, the more power it is going to consume. Hence, reduction in clock speed can be considered based on the performance requirements of the application. This will help in conserving the battery.

- **Low Power Modes (Sleep Modes):** Most microcontrollers and microprocessors offer mechanisms to go to sleep and deep sleep, switching off most of the peripherals. Some microcontrollers can pause the clock as well during sleep and once the connected device is active, they start executing from where it was paused. These features need to be used by the application if there is no processing being performed, or the device is waiting for a certain event. Once an event is received, the device can be programmed to wake up, perform the specific activity and go back to sleep again.

- **Protocol Selection:** Technologies and protocols need to be chosen wisely to reduce power consumption. HTTPS is considered to be more power-hungry compared to MQTT. Also, connected devices that communicate a lot of data will consume more power. Hence, one needs to have a clear strategy on the frequency and quantity of data exchange between connected devices and servers over the Internet.

Data Encryption

IoT security is a big challenge faced by connected device manufacturers. Designers need to use secure encrypted channels to connect to central servers and transmit/receive data. Keys have to be stored securely and in encrypted format.

Connected devices that store sensitive information in non-encrypted format or transmitting data in clear text are an obvious target for hackers. Hence, any sensitive data (user name, passwords, certificates, keys, etc.) stored in the EEPROM or flash memory needs to be encrypted by firmware designers prior to storing. Chip vendors today provide mechanisms to encrypt flash content that helps in securing the device. Where certificates (that are required by cloud servers) need to be generated and stored in the device, chip vendors provide one-stop solutions to generating the keys and burning them during the manufacturing process. This is done in highly secure environments.

Designers need to use encryption methods for connected devices to demonstrate their integrity. A simple cyclic redundancy check (CRC) can help identify if the firmware has been changed on the device. Hence, device integrity checks help identify malicious firmware changes.

Devices also need to be built with the capability to upgrade their firmware to add security patches, fix application defects and add new feature upgrades. Without this capability, connected devices will continue to be vulnerable to security threats.

Communication Protocols

The more proprietary the communication protocols, the more difficult it is to connect and integrate the connected device to the Internet. The more standard the communication protocols used by devices, better is their extensibility. Hence, firmware designers need to keep this in mind during development of the connected device.

Today, many connected devices are being connected to popular IoT platforms like AWS® IoT, Azure® IoT Hub, Google Cloud™ IoT, IBM Watson™ IoT, etc. It is easier to connect devices to these platforms if they support protocols specified by these vendors.

Connectivity Options

Appropriate connectivity solutions need to be considered during the design phase. There are several low power protocols being defined specifically for device connectivity which can be explored by the firmware designer.

If connected devices are deployed in remote locations, they may need to be designed to support 4G LTE capability instead of a Wi-Fi or Bluetooth capability. Also, we may need to plan for intermittent connectivity in remote locations.

Similarly, for devices that will communicate to a Body or Personal Area Network, BLE (Bluetooth Low Energy) solutions can be considered instead of a Bluetooth Classic or a Wi-Fi solution.

Edge Intelligence

Firmware designers need to consider the emerging trend of incorporating of intelligence in the connected device, instead of relying on the cloud to overcome various latency and communication constraints. Edge intelligence and edge analytics enables data analytics in real-time and on site where data collection is occurring. This imposes additional constraints on firmware architecture, processing power, RAM and data storage. Edge intelligent, connected devices can interact with all the other devices in the local network by collecting, storing, filtering and analyzing the device data. It also adds an additional layer of network security to filter, bundle and encrypt the data before sending it to the cloud.

Conclusion

In developing firmware for connected devices, businesses must consider not just the intended application functionality, but also other factors such as battery life, data security and encryption, communication protocols, operating environment, connectivity options, and edge intelligence.

As an IoT services provider, Thinxtream has expertise in developing secure and energy-efficient solutions for IoT firmware across ARM® based and other microcontrollers and microprocessors from Microchip®, Freescale™, Espressif™, Texas Instruments™, Intel®, Marvell® and ST®. By leveraging the IoT expertise built over a decade, Thinxtream ensures cost-effective, quality and timely delivery of connected devices and IoT solutions.

References

- Internet of Things for Architects
  - Book by Perry Lea
- Practical Internet of Things Security
  - Book by Brian Russell and Drew Van Dure
- A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications by Mahmoud Shuker Mahmoud and Auday Mahmoud

Thinxtream Technologies is a global software company with a portfolio of innovative software platforms, products, components, solutions, patents, competences and services for Internet of Things (IoT) across several industry verticals and applications, successfully enabling leading customers, including Fortune 500 companies, meet their application, product and business goals.

Interested in learning more? For more information contact:

Thinxtream Technologies Pte. Ltd.
220 Orchard Road #05-01
Midpoint Orchard
SINGAPORE 238852
Phone: +65 66358625
Email: info@thinxtream.com

Thinxtream Technologies, Inc.
10260 SW Greenburg Road
Suite 400 Portland, OR 97223,
U.S.A
Phone: +1 503 293-3598
Email: info@thinxtream.com

www.thinxtream.com

LinkedIn/thinxtream

TT-WP-009-1-1118

THINXTREAM®