IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

LUCIO DEVELOPMENT LLC,	§	
	§	
Plaintiff,	§	Case No: 1:17-cv-1153
	§	
vs.	§	PATENT CASE
	§	
MEDIATEK USA, INC.,	§	
	§	
Defendant.	§	
	§	

COMPLAINT

Plaintiff Lucio Development LLC ("Plaintiff" or "Lucio") files this Complaint against MediaTek USA, Inc. ("Defendant" or "MediaTek") for infringement of United States Patent No. 7,069,546 (hereinafter "the '546 Patent").

PARTIES AND JURISDICTION

1. This is an action for patent infringement under Title 35 of the United States Code. Plaintiff is seeking injunctive relief as well as damages.

2. Jurisdiction is proper in this Court pursuant to 28 U.S.C. §§ 1331 (Federal Question) and 1338(a) (Patents) because this is a civil action for patent infringement arising under the United States patent statutes.

3. Plaintiff is a Texas limited liability company with its office address at 555 Republic Dr., Suite 200, Plano, Texas 75074.

4. On information and belief, Defendant is a Delaware corporation with a place of business at 5914 W. Courtyard Drive, Austin, TX 78730. On information and belief, Defendant is registered to conduct business in Texas and may be served through its registered

agent, CT Corporation System, 1999 Bryan Street, Suite 900, Dallas, TX 75201-3136.

5. This Court has personal jurisdiction over Defendant because Defendant has committed, and continues to commit, acts of infringement in this District, has conducted business in this District, and/or has engaged in continuous and systematic activities in this District.

6. On information and belief, Defendant's instrumentalities that are alleged herein to infringe were and continue to be used, imported, offered for sale, and/or sold in this District.

VENUE

7. Venue is proper in this District pursuant to 28 U.S.C. §1400(b) because acts of infringement are occurring in this District and Defendant has a regular and established place of business in this District. For instance, on information and belief, Defendant has a regular and established place of business at 5914 W. Courtyard Drive, Austin, TX 78730.

COUNT I
(INFRINGEMENT OF UNITED STATES PATENT NO. 7,069,546)

8. Plaintiff incorporates paragraphs 1 through 7 herein by reference.

9. This cause of action arises under the patent laws of the United States and, in particular, under 35 U.S.C. §§ 271, *et seq*.

10. Plaintiff is the owner by assignment of the '546 Patent with sole rights to enforce the '546 Patent and sue infringers.

11. A copy of the '546 Patent, titled "Generic Framework for Embedded Software Development," is attached hereto as Exhibit A.

12. The '546 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

13. On information and belief, Defendant has infringed and continues to infringe

one or more claims, including at least Claim 1, of the '546 Patent by making, using, importing, selling, and/or offering for sale a software platform for embedded software development, which is covered by at least Claim 1 of the '546 Patent. Defendant has infringed and continues to infringe the '546 Patent directly in violation of 35 U.S.C. § 271.

14. Defendant, sells, offers to sell, and/or uses embedded software development packages including, without limitation, the LinkIt and/or LinkIt One software developer kit (SDK), and any similar products ("Product"), which infringe at least Claim 1 of the '546 Patent.

15. The Product is a framework (e.g., a software development kit) for multiple Hardware Development Kits (HDK) such as Linkit 7687 HDK, Wi-Fi Modules and Arduino. Defendant and/or its customers specifically use the LinkIt SDK to produce embedded software. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

1. Introduction

1.1. What is LinkIt?

MediaTek LinkIt™ is a collection of development platforms designed for the creation and prototyping of Wearables and Internet of Things (IoT) devices. These development platforms are offered in two distinct families:

- LinkIt ONE, for simple application use wearables and IoT devices such as smart wristband, smart safety and tracking devices. These devices provide the user with feedback and control options on the device, and can exchange data and control messages with users, other smart devices, and cloud applications using GSM messaging, GPRS, Wi-Fi or Bluetooth connections.
- LinkIt Connect, for one application use IoT devices such as smart bulbs and smart appliances that are controlled from cloud services or smartphones over Wi-Fi or Bluetooth connections.

Each development platform in turn will offer one or more chipset and API variants designed to meet specific development and device requirements. And to enable the creation and prototyping of devices, each variant includes the below items:

- One or more HDKs to enable the prototyping of devices.
- An SDK to enable the creation of firmware or software for devices.
- One or more hardware reference designs that can be used as the basis for board layouts for the final product.
- Comprehensive documentation, such as API references, developer guides, chipset descriptions and pin-out diagrams.

Source: https://labs.Mediatek.com/en/search?lang=en&page=1&q=sdk&tab=all&type=all

1.2. MediaTek LinkIt ONE development platform

The MediaTek LinkIt ONE development platform (see Figure 1) offers a robust yet flexible development platform for wearable and IoT devices. The platform consists of the following:

- System-on-Chip (SoC) MediaTek MT2502 (Aster), the world's smallest commercial SoC for Wearables and IoT devices, and its energy efficient Wi-Fi and GPS companion chipsets.
- LinkIt ONE APIs.
- LinkIt ONE Hardware Development Kit (HDK).
- LinkIt ONE Software Development Kit (SDK).

Figure 1 The components of the LinkIt ONE development platform

Source: https://labs.Mediatek.com/en/search?lang=en&page=1&q=sdk&tab=all&type=all

As shown in Figure 2, using the LinkIt ONE SDK you create an Arduino Sketch to make use of the LinkIt ONE APIs. These APIs execute over the run-time environment to enable you to access the features of the LinkIt ONE development board.

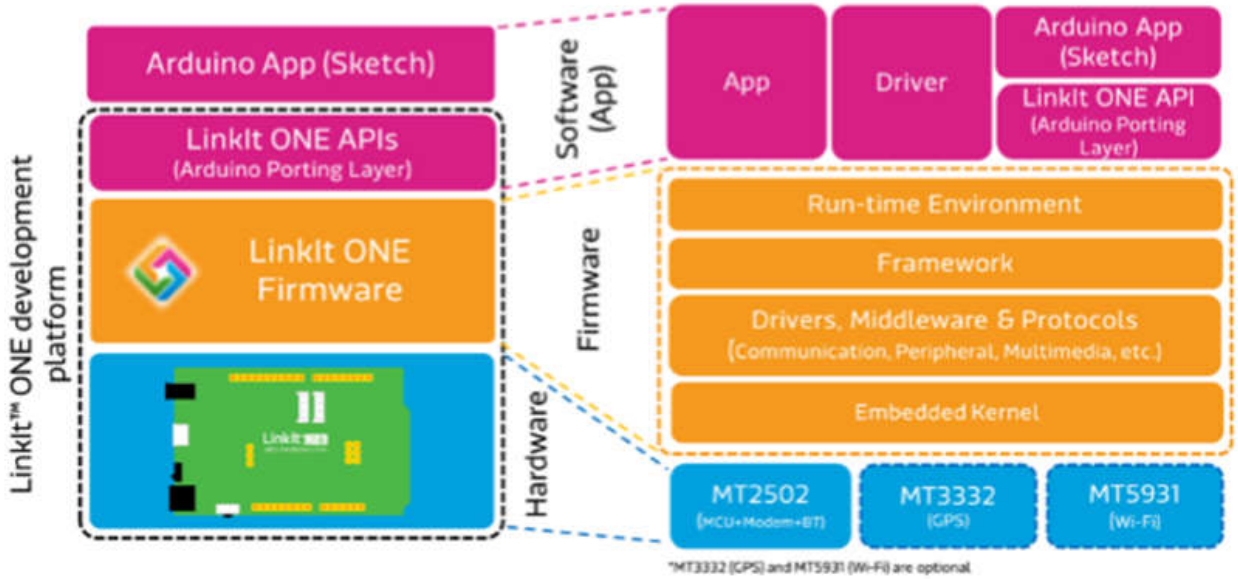


Figure 2 The architecture of the LinkIt ONE development platform

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE

1.3.2. Running Arduino sketches

To execute a sketch, LinkIt ONE SDK compiles the sketch into a LinkIt ONE executable (VXP file) as shown in Figure 3. The IDE plug-in then loads the VXP file into the file system of LinkIt ONE development board. When the LinkIt ONE development board boots up, it automatically executes the loaded VXP file. The VXP executable is then loaded by the run-time environment.

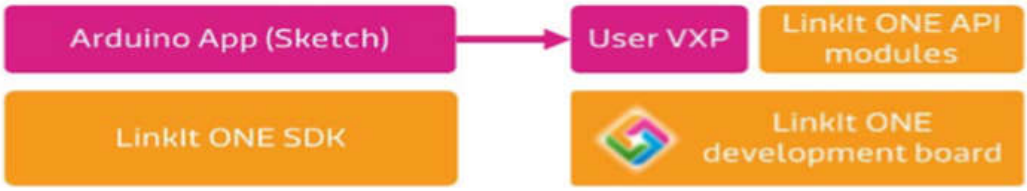


Figure 3 The development and test process

While the sketch itself is written as a single-threaded loop, the run-time environment is a multi-threaded soft-realtime environment. The VXP runs as a standalone thread. The Arduino plug-in wrapper layer is responsible for sending requests to other service modules, which run on separate threads.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE

16. The Product provides one or more generic application handler programs (e.g., LinkIt provides a Hardware Abstraction Layer (HAL) containing CMSIS and other programs,

functions and data structures which are common and uniform across all supported HDK and Arduino). The generic programs comprise computer program code for performing generic application functions common to multiple types of hardware modules used in a communication environment (e.g., the generic code provides common and generic functions to multiple hardware modules, such as LinkIt 7687 HDK, Wi-Fi Modules and Arduino). Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

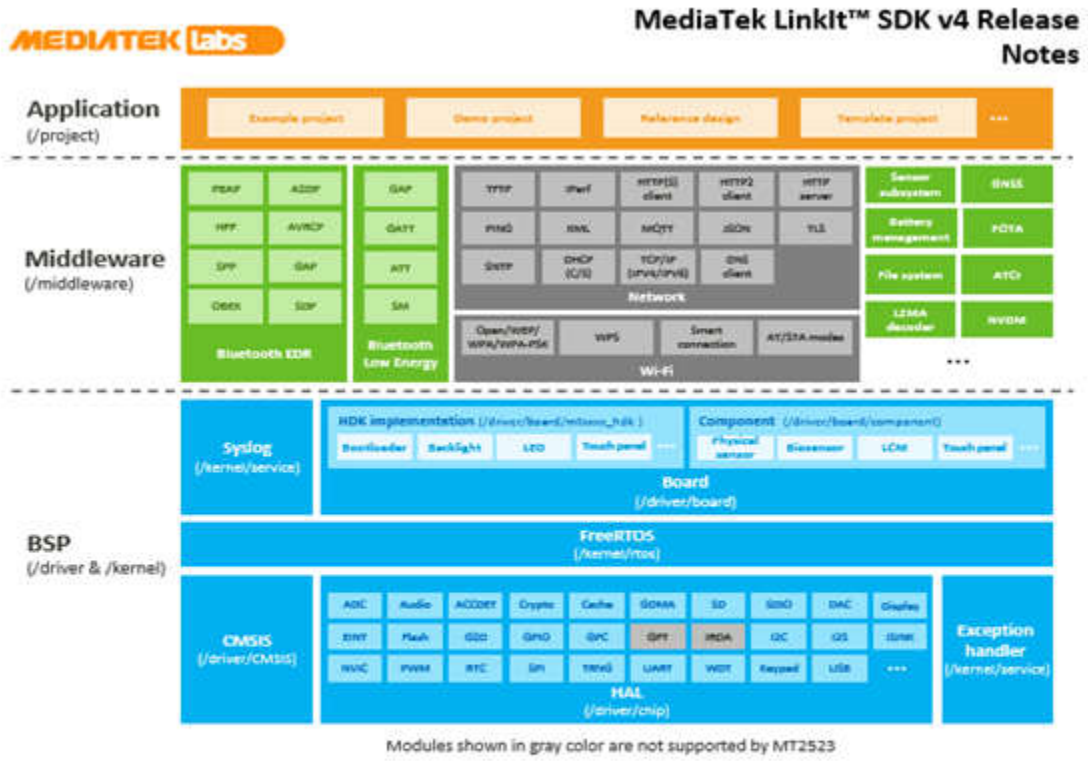


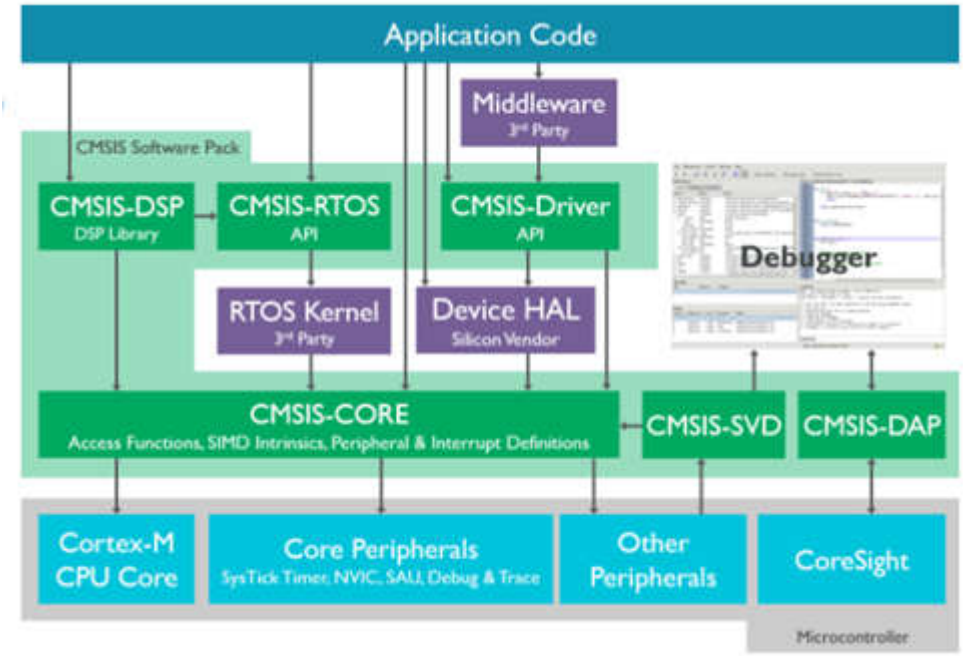
Figure 2. Architecture layout of the LinkIt SDK v4 for LinkIt 2523 HDK

A functional block in grey means the HDK does not support the feature. The top layer includes the application projects running on the SDK. They are based on Middleware, OS and HAL layers. These layers provide rich features for application development, such as the Middleware provides the Bluetooth Low Energy Stack, GNSS, FOTA,

Source: https://labs.Mediatek.com/en/search?lang=en&page=1&q=sdk&tab=all&type=all

Starting from CMSIS-CORE, a vendor-independent hardware abstraction layer for Cortex-M processors, CMSIS has since expanded into areas such as software component management and reference debugger interfaces. Creation of software is a major cost factor in the embedded industry. Standardizing the software interfaces across all Cortex-M silicon vendor products, especially when creating new projects or migrating existing software to a new device, means significant cost reductions.

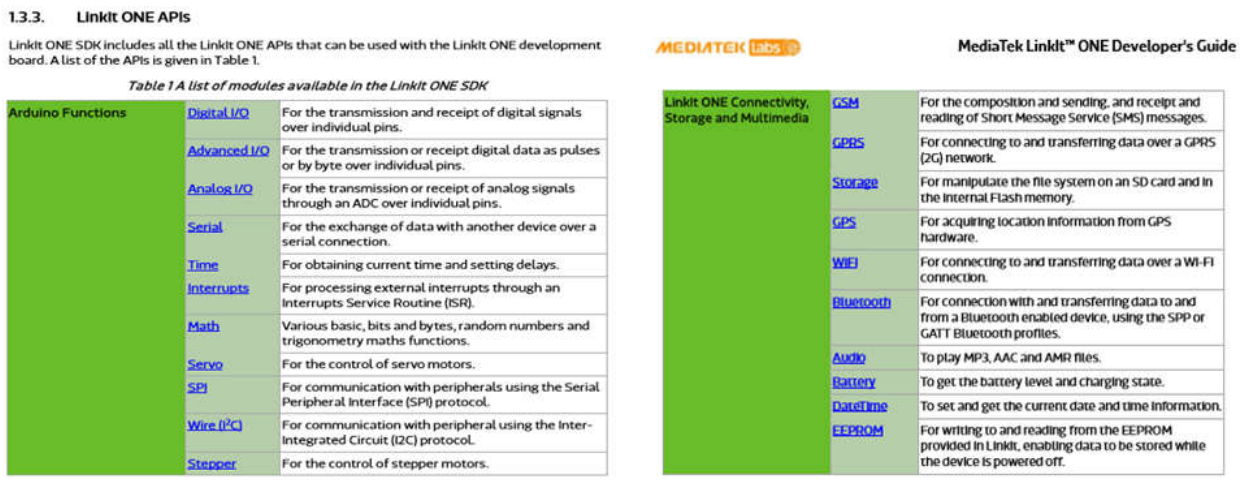
CMSIS is defined in close cooperation with various silicon and software vendors and provides a common approach to interface to peripherals, real-time operating systems, and middleware components. It simplifies software reuse, reducing the learning curve for new microcontroller developers and cutting the time-to market for devices.



Source: https://developer.arm.com/embedded/cmsis

17. The Product includes generating specific application handler code to associate the generic functions with the specific functions at a device driver for at least one of the types of hardware modules. For example, in addition to the generic drivers HAL, LinkIt also includes specific application handler code that is specific to the application (such as GSM/GPRS, Bluetooth, GPS, SD Flash, Digital IO, etc.) and specific to particular hardware (such as HDKs and Arduino boards). Certain elements of this limitation are illustrated in the

screenshots below and in the screenshots referenced in connection with other elements herein.



1.3.3. LinkIt ONE APIs

LinkIt ONE SDK includes all the LinkIt ONE APIs that can be used with the LinkIt ONE development board. A list of the APIs is given in Table 1.

Table 1 A list of modules available in the LinkIt ONE SDK

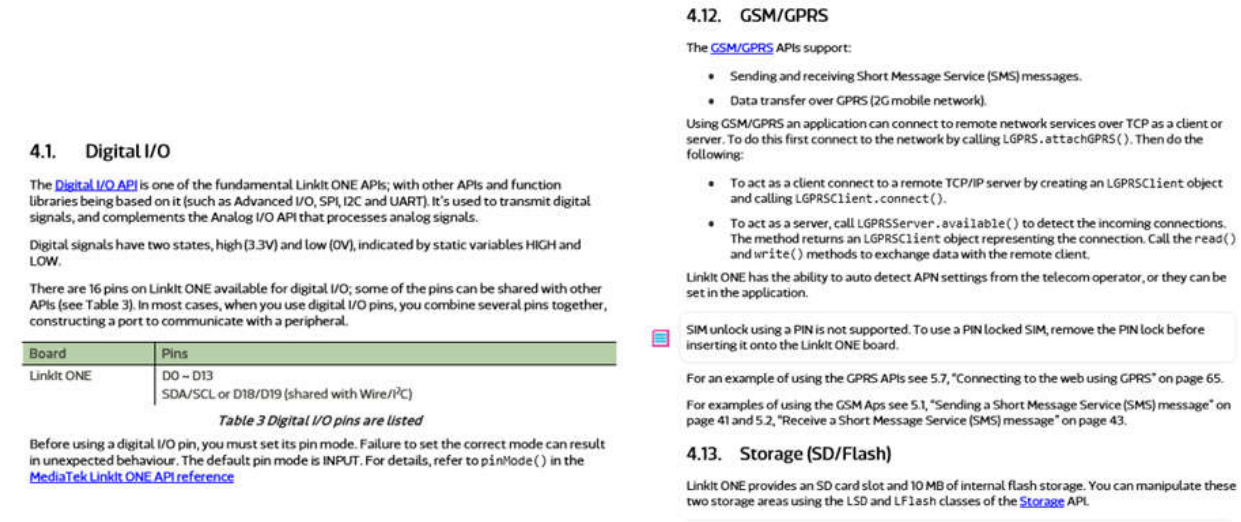
Arduino Functions	Digital I/O	For the transmission and receipt of digital signals over individual pins.
	Advanced I/O	For the transmission or receipt digital data as pulses or by byte over individual pins.
	Analog I/O	For the transmission or receipt of analog signals through an ADC over individual pins.
	Serial	For the exchange of data with another device over a serial connection.
	Time	For obtaining current time and setting delays.
	Interrupts	For processing external interrupts through an Interrupts Service Routine (ISR).
	Math	Various basic, bits and bytes, random numbers and trigonometry maths functions.
	Servo	For the control of servo motors.
	SPI	For communication with peripherals using the Serial Peripheral Interface (SPI) protocol.
	Wire (I²C)	For communication with peripheral using the Inter-Integrated Circuit (I2C) protocol.
	Stepper	For the control of stepper motors.

MEDIATEK Labs

MediaTek LinkIt™ ONE Developer's Guide

LinkIt ONE Connectivity, Storage and Multimedia	GSM	For the composition and sending, and receipt and reading of Short Message Service (SMS) messages.
	GPRS	For connecting to and transferring data over a GPRS (2G) network.
	Storage	For manipulate the file system on an SD card and in the internal Flash memory.
	GPS	For acquiring location information from GPS hardware.
	WiFi	For connecting to and transferring data over a Wi-Fi connection.
	Bluetooth	For connection with and transferring data to and from a Bluetooth enabled device, using the SPP or GATT Bluetooth profiles.
	Audio	To play MP3, AAC and AMR files.
	Battery	To get the battery level and charging state.
	DateTime	To set and get the current date and time information.
	EEPROM	For writing to and reading from the EEPROM provided in LinkIt, enabling data to be stored while the device is powered off.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE



4.1. Digital I/O

The Digital I/O API is one of the fundamental LinkIt ONE APIs; with other APIs and function libraries being based on it (such as Advanced I/O, SPI, I2C and UART). It's used to transmit digital signals, and complements the Analog I/O API that processes analog signals.

Digital signals have two states, high (3.3V) and low (0V), indicated by static variables HIGH and LOW.

There are 16 pins on LinkIt ONE available for digital I/O; some of the pins can be shared with other APIs (see Table 3). In most cases, when you use digital I/O pins, you combine several pins together, constructing a port to communicate with a peripheral.

Board	Pins
LinkIt ONE	D0 – D13 SDA/SCL or D18/D19 (shared with Wire/I²C)

Table 3 Digital I/O pins are listed

Before using a digital I/O pin, you must set its pin mode. Failure to set the correct mode can result in unexpected behaviour. The default pin mode is INPUT. For details, refer to `pinMode()` in the MediaTek LinkIt ONE API reference

4.12. GSM/GPRS

The GSM/GPRS APIs support:

- Sending and receiving Short Message Service (SMS) messages.
- Data transfer over GPRS (2G mobile network).

Using GSM/GPRS an application can connect to remote network services over TCP as a client or server. To do this first connect to the network by calling `LGPRS.attachGPRS()`. Then do the following:

- To act as a client connect to a remote TCP/IP server by creating an `LGPRSClient` object and calling `LGPRSClient.connect()`.
- To act as a server, call `LGPRSServer.available()` to detect the incoming connections. The method returns an `LGPRSClient` object representing the connection. Call the `read()` and `write()` methods to exchange data with the remote client.

LinkIt ONE has the ability to auto detect APN settings from the telecom operator, or they can be set in the application.

SIM unlock using a PIN is not supported. To use a PIN locked SIM, remove the PIN lock before inserting it onto the LinkIt ONE board.

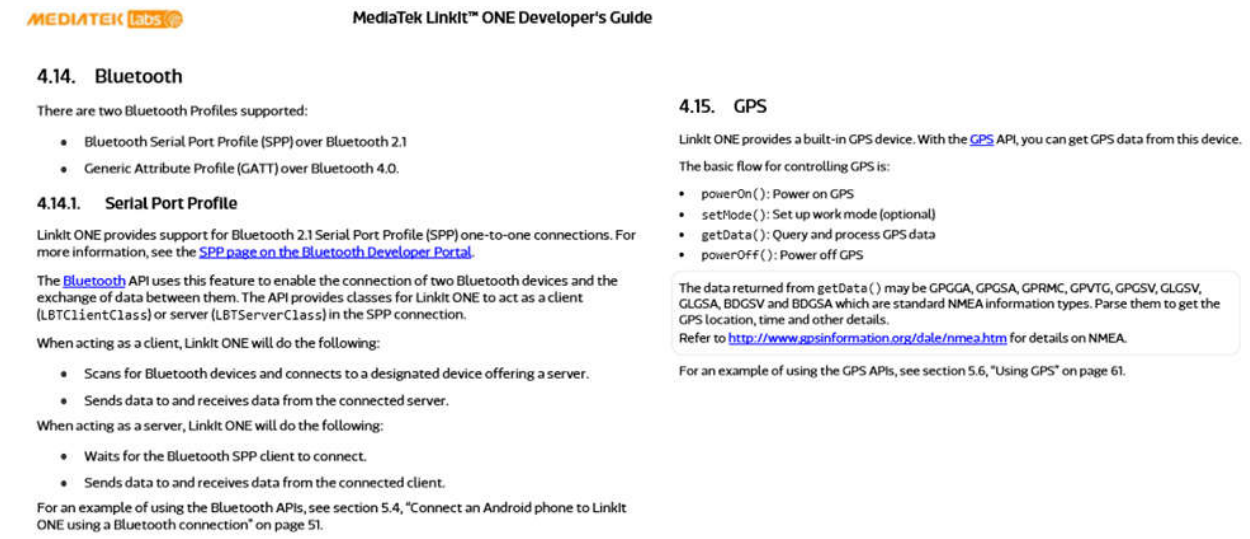
For an example of using the GPRS APIs see 5.7, "Connecting to the web using GPRS" on page 65.

For examples of using the GSM Aps see 5.1, "Sending a Short Message Service (SMS) message" on page 41 and 5.2, "Receive a Short Message Service (SMS) message" on page 43.

4.13. Storage (SD/Flash)

LinkIt ONE provides an SD card slot and 10 MB of internal flash storage. You can manipulate these two storage areas using the `LSD` and `LFlash` classes of the Storage API.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE



MediaTek LinkIt™ ONE Developer's Guide

4.14. Bluetooth

There are two Bluetooth Profiles supported:

- Bluetooth Serial Port Profile (SPP) over Bluetooth 2.1
- Generic Attribute Profile (GATT) over Bluetooth 4.0.

4.14.1. Serial Port Profile

LinkIt ONE provides support for Bluetooth 2.1 Serial Port Profile (SPP) one-to-one connections. For more information, see the SPP page on the Bluetooth Developer Portal.

The Bluetooth API uses this feature to enable the connection of two Bluetooth devices and the exchange of data between them. The API provides classes for LinkIt ONE to act as a client (`LBTClientClass`) or server (`LBTServerClass`) in the SPP connection.

When acting as a client, LinkIt ONE will do the following:

- Scans for Bluetooth devices and connects to a designated device offering a server.
- Sends data to and receives data from the connected server.

When acting as a server, LinkIt ONE will do the following:

- Waits for the Bluetooth SPP client to connect.
- Sends data to and receives data from the connected client.

For an example of using the Bluetooth APIs, see section 5.4, "Connect an Android phone to LinkIt ONE using a Bluetooth connection" on page 51.

4.15. GPS

LinkIt ONE provides a built-in GPS device. With the GPS API, you can get GPS data from this device.

The basic flow for controlling GPS is:

- `powerOn()`: Power on GPS
- `setMode()`: Set up work mode (optional)
- `getData()`: Query and process GPS data
- `powerOff()`: Power off GPS

The data returned from `getData()` may be GPGGA, GPGSA, GPRMC, GPVTG, GPGSV, GLGSV, GLGSA, BDGSV and BDGSA which are standard NMEA information types. Parse them to get the GPS location, time and other details.
Refer to http://www.gpsinformation.org/dale/nmea.htm for details on NMEA.

For an example of using the GPS APIs, see section 5.6, "Using GPS" on page 61.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE

18. The Product generates specific application handler code and defines a specific element in the specific code to be handled by one of the generic application functions for that hardware module. For example**,** LinkIt generates system-specific application handler code by defining a specific element such as functions and data structures corresponding to specific HDKs and hardware components such as GSM/GPRS, Bluetooth, GPS, SD Flash, Digital IO, etc. that extend or otherwise connect the system-specific application handler code to the functions and data structures defined and made available by the HAL. When specific functions are written for handling defined specific elements, the specific functions must be registered. LinkIt accordingly contains data structures that register and embed the required functions. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.

4.14.2. Generic Attribute Profile

LinkIt ONE also provides support for Bluetooth 4.0 Generic Attribute Profile (GATT). For more information see, the GATT page on the Bluetooth Developer Portal.

The Bluetooth API uses this feature to enable the connection of two Bluetooth devices and the exchange of data between them. The API provides classes for LinkIt ONE to act as a client (`LGATTClient`) or server (`LGATTServer`) in the GATT connection. To enable you to work with the data exchanged by these classes, the class `LGATTService` provides functions to access service details and data.

When acting as a client (GAP central device), LinkIt ONE will do the following:

- Creates a GATT client.
- Scans for Bluetooth devices providing the desired standard or custom GATT Profile and connects to a designated server device.
- Sends data to and requests data from the connected server.

When acting as a server (GAP peripheral device) LinkIt ONE will do the following:

- Creates a GATT server and defines the services it offers (as a set of services that comply with a GATT profile or a set of defined and custom services to create a custom profile).
- Waits for the Bluetooth GATT client to connect.
- Sends data to and receives data from the client.

4.12. GSM/GPRS

The GSM/GPRS APIs support:

- Sending and receiving Short Message Service (SMS) messages.
- Data transfer over GPRS (2G mobile network).

Using GSM/GPRS an application can connect to remote network services over TCP as a client or server. To do this first connect to the network by calling `LGPRS.attachGPRS()`. Then do the following:

- To act as a client connect to a remote TCP/IP server by creating an `LGPRSClient` object and calling `LGPRSClient.connect()`.
- To act as a server, call `LGPRSServer.available()` to detect the incoming connections. The method returns an `LGPRSClient` object representing the connection. Call the `read()` and `write()` methods to exchange data with the remote client.

LinkIt ONE has the ability to auto detect APN settings from the telecom operator, or they can be set in the application.

SIM unlock using a PIN is not supported. To use a PIN locked SIM, remove the PIN lock before inserting it onto the LinkIt ONE board.

For an example of using the GPRS APIs see 5.7, "Connecting to the web using GPRS" on page 65.

For examples of using the GSM Aps see 5.1, "Sending a Short Message Service (SMS) message" on page 41 and 5.2, "Receive a Short Message Service (SMS) message" on page 43.

4.13. Storage (SD/Flash)

LinkIt ONE provides an SD card slot and 10 MB of internal flash storage. You can manipulate these two storage areas using the `LSD` and `LFlash` classes of the Storage API.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE

4.14. Bluetooth

There are two Bluetooth Profiles supported:

- Bluetooth Serial Port Profile (SPP) over Bluetooth 2.1
- Generic Attribute Profile (GATT) over Bluetooth 4.0.

4.14.1. Serial Port Profile

LinkIt ONE provides support for Bluetooth 2.1 Serial Port Profile (SPP) one-to-one connections. For more information, see the SPP page on the Bluetooth Developer Portal.

The Bluetooth API uses this feature to enable the connection of two Bluetooth devices and the exchange of data between them. The API provides classes for LinkIt ONE to act as a client (`LBTClientClass`) or server (`LBTServerClass`) in the SPP connection.

When acting as a client, LinkIt ONE will do the following:

- Scans for Bluetooth devices and connects to a designated device offering a server.
- Sends data to and receives data from the connected server.

When acting as a server, LinkIt ONE will do the following:

- Waits for the Bluetooth SPP client to connect.
- Sends data to and receives data from the connected client.

For an example of using the Bluetooth APIs, see section 5.4, "Connect an Android phone to LinkIt ONE using a Bluetooth connection" on page 51.

4.15. GPS

LinkIt ONE provides a built-in GPS device. With the GPS API, you can get GPS data from this device.

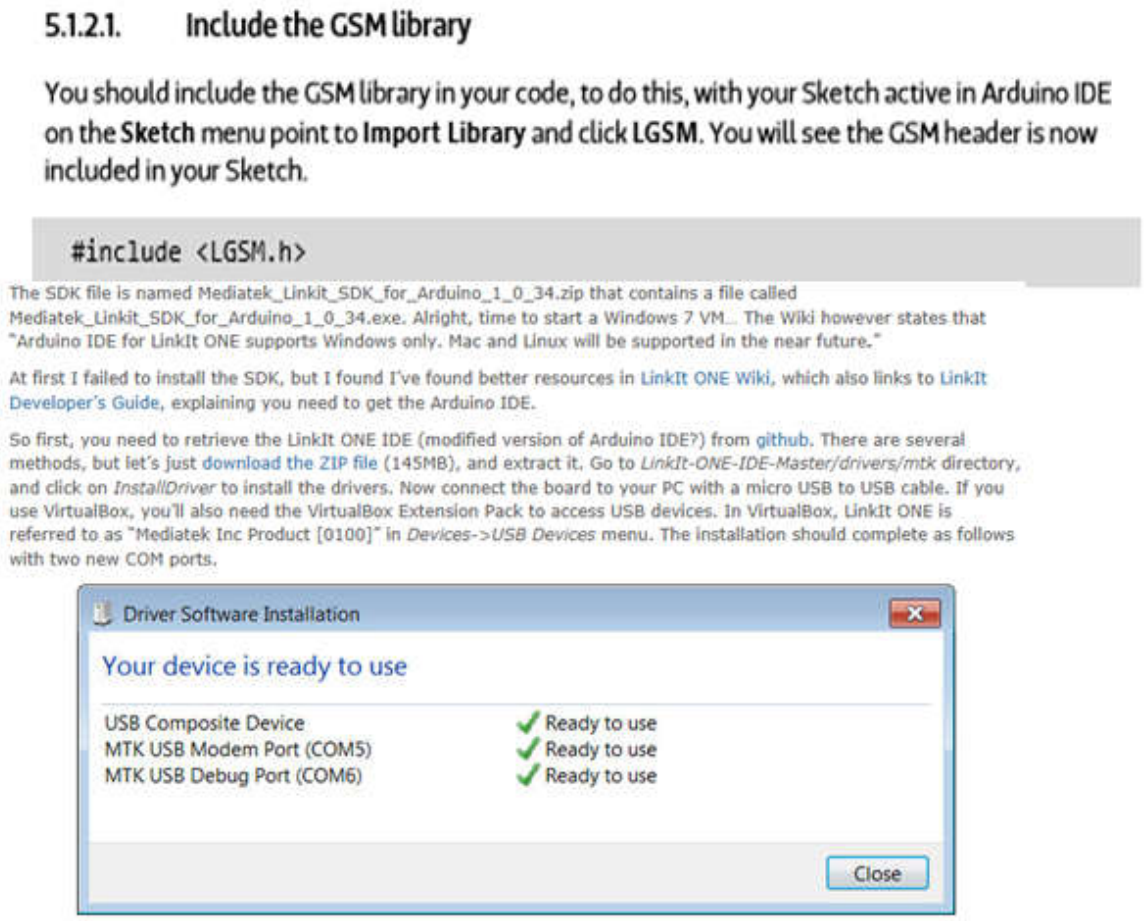
The basic flow for controlling GPS is:

- `powerOn()`: Power on GPS
- `setMode()`: Set up work mode (optional)
- `getData()`: Query and process GPS data
- `powerOff()`: Power off GPS

The data returned from `getData()` may be GPGGA, GPGSA, GPRMC, GPVTG, GPGSV, GLGSV, GLGSA, BDGSV and BDGSA which are standard NMEA information types. Parse them to get the GPS location, time and other details.
Refer to http://www.gpsinformation.org/dale/nmea.htm for details on NMEA.

For an example of using the GPS APIs, see section 5.6, "Using GPS" on page 61.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE



5.1.2.1. Include the GSM library

You should include the GSM library in your code, to do this, with your Sketch active in Arduino IDE on the **Sketch** menu point to **Import Library** and click **LGSM**. You will see the GSM header is now included in your Sketch.

```
#include <LGSM.h>
```

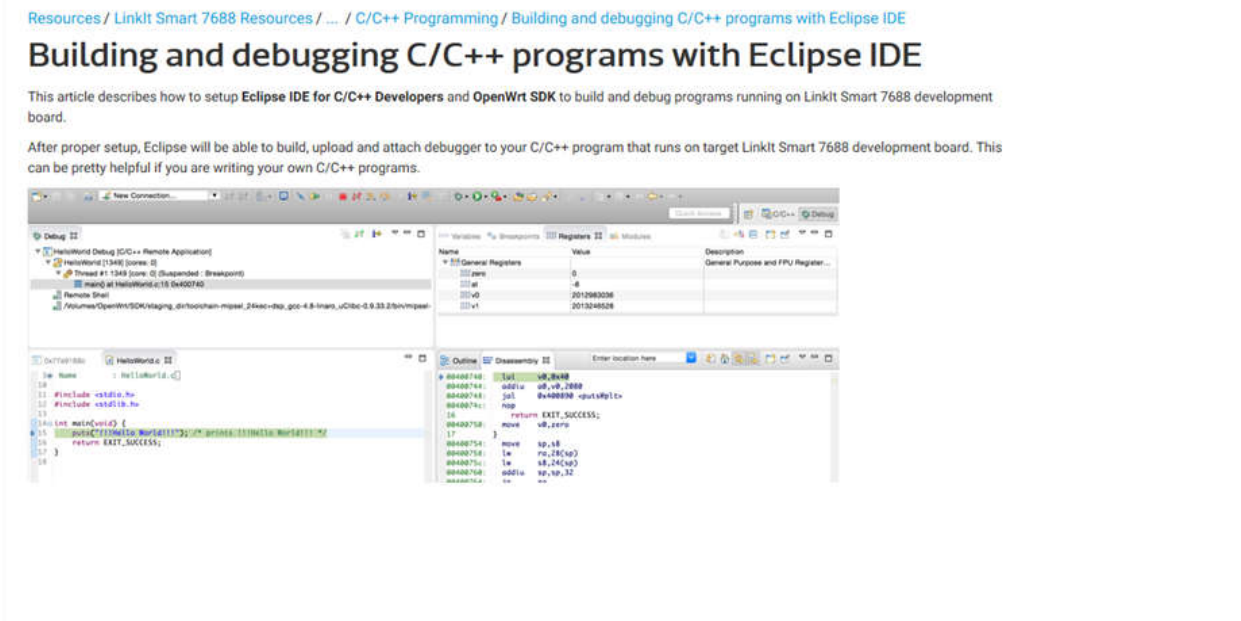
The SDK file is named Mediatek_Linkit_SDK_for_Arduino_1_0_34.zip that contains a file called Mediatek_Linkit_SDK_for_Arduino_1_0_34.exe. Alright, time to start a Windows 7 VM... The Wiki however states that "Arduino IDE for LinkIt ONE supports Windows only. Mac and Linux will be supported in the near future."

At first I failed to install the SDK, but I found I've found better resources in LinkIt ONE Wiki, which also links to LinkIt Developer's Guide, explaining you need to get the Arduino IDE.

So first, you need to retrieve the LinkIt ONE IDE (modified version of Arduino IDE?) from github. There are several methods, but let's just download the ZIP file (145MB), and extract it. Go to *LinkIt-ONE-IDE-Master/drivers/mtk* directory, and click on *InstallDriver* to install the drivers. Now connect the board to your PC with a micro USB to USB cable. If you use VirtualBox, you'll also need the VirtualBox Extension Pack to access USB devices. In VirtualBox, LinkIt ONE is referred to as "Mediatek Inc Product [0100]" in *Devices->USB Devices* menu. The installation should complete as follows with two new COM ports.

Source: Mediatek LinkIt Developer's Guide, https://labs.Mediatek.com/en/download/gMfB20OE

19. When a specific application is needed for a particular hardware, the generic functions and the specific functions are compiled together to yield a machine readable code. Mediatek and/or its customers compile the generic functions and the specific functions using LinkIt and/or any other compiling IDE supported by Mediatek. Certain elements of this limitation are illustrated in the screenshots below and in the screenshots referenced in connection with other elements herein.



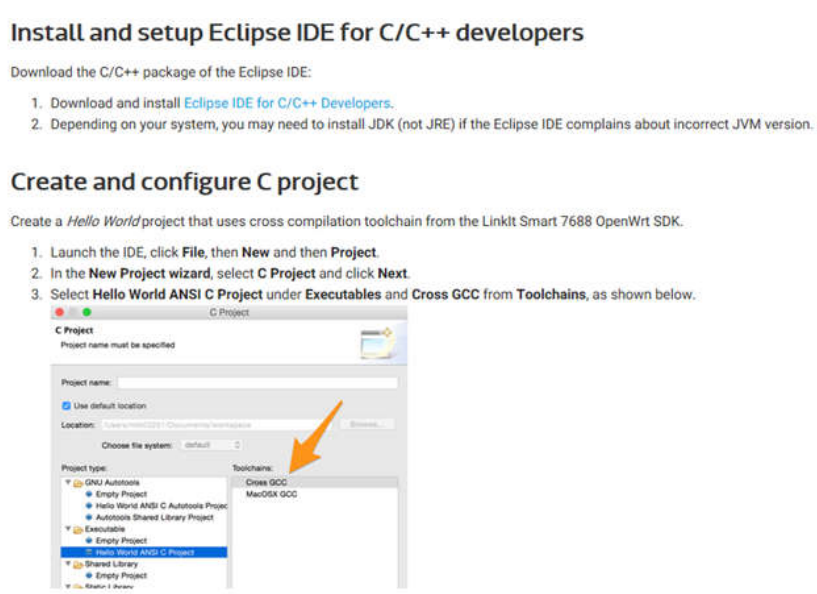
Resources / LinkIt Smart 7688 Resources / ... / C/C++ Programming / Building and debugging C/C++ programs with Eclipse IDE

Building and debugging C/C++ programs with Eclipse IDE

This article describes how to setup **Eclipse IDE for C/C++ Developers** and **OpenWrt SDK** to build and debug programs running on LinkIt Smart 7688 development board.

After proper setup, Eclipse will be able to build, upload and attach debugger to your C/C++ program that runs on target LinkIt Smart 7688 development board. This can be pretty helpful if you are writing your own C/C++ programs.

Source: https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/c-c++-programming/building-and-debugging-c-c++-programs-with-eclipse-ide



Install and setup Eclipse IDE for C/C++ developers

Download the C/C++ package of the Eclipse IDE:

1. Download and install Eclipse IDE for C/C++ Developers.
2. Depending on your system, you may need to install JDK (not JRE) if the Eclipse IDE complains about incorrect JVM version.

Create and configure C project

Create a *Hello World* project that uses cross compilation toolchain from the LinkIt Smart 7688 OpenWrt SDK.

1. Launch the IDE, click **File**, then **New** and then **Project**.
2. In the **New Project wizard**, select **C Project** and click **Next**.
3. Select **Hello World ANSI C Project** under **Executables** and **Cross GCC** from **Toolchains**, as shown below.

Source: https://docs.labs.mediatek.com/resource/linkit-smart-7688/en/tutorials/c-c++-programming/building-and-debugging-c-c++-programs-with-eclipse-ide

20. Defendant's actions complained of herein will continue unless Defendant is enjoined by this court.

21. Defendant's actions complained of herein are causing irreparable harm and monetary damage to Plaintiff and will continue to do so unless and until Defendant is enjoined and restrained by this Court.

22. Plaintiff is in compliance with 35 U.S.C. § 287.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff asks the Court to:

(a) Enter judgment for Plaintiff on this Complaint on all causes of action asserted herein;

(b) Enter an Order enjoining Defendant, its agents, officers, servants, employees, attorneys, and all persons in active concert or participation with Defendant who receive notice of the order from further infringement of United States Patent No. 7,069,546 (or, in the alternative, awarding Plaintiff a running royalty from the time of judgment going forward);

(c) Award Plaintiff damages resulting from Defendant's infringement in accordance with 35 U.S.C. § 284;

(d) Award Plaintiff pre-judgment and post-judgment interest and costs; and

(e) Award Plaintiff such further relief to which the Court finds Plaintiff entitled under law or equity.

Dated: December 11, 2017

Respectfully submitted,

/s/ Jay Johnson

JAY JOHNSON
State Bar No. 24067322
D. BRADLEY KIZZIA
State Bar No. 11547550
KIZZIA JOHNSON, PLLC
1910 Pacific Ave., Suite 13000
Dallas, Texas 75201
(214) 451-0164
Fax: (214) 451-0165
jay@kjpllc.com
bkizzia@kjpllc.com

ATTORNEYS FOR PLAINTIFF

EXHIBIT A