

## Selection Criteria for ZigBee Development Kits

This article gives an overview about different considerations, when it comes to prioritizing ZigBee Development Kits, supplied by different vendors. Before selection of any kit, some of the OEMs' requirements that could pop-out are: suitable investment to be put in & choosing a right vendor as per application needs, less development time period, immediate hardware resources availability for production, etc. To meet these requirements, the following issues based on the resources provided within a development kit can be considered; henceforth, lead them to choose a kit which best fits their needs. The target audience can be system architects, software developers and R&D engineers; who are involved in the selection of vendor. The reader is expected to have some knowledge about networking and awareness about general software development platforms provided within development kits.

Contents mentioned further such as Additional Costs, Software Stack and Test Tools are some of the basic issues concerning to the choosing of a development kit at the first place before purchase. Remaining ones such as Software, Support and Documentation are known from particular development kit's advertised online content and by experience of kit. These can be viewed as next phase in prioritizing the kits and are the ones to be considered by OEM for the ease of product development. In the end, a small conclusion is given with an outline of suggestions.

### Additional Costs

Before opting for a Development Kit, there are some of the minor additional costs that need to be considered in parallel in addition to the cost to purchase it. These in general pertains to the ones doing smaller projects or for whom investment to acquire a kit software is also one of the primary issues.

Firstly, one must consider costs corresponding to Compiler / IDE Multiple User License. Dev. kit's provided software platform generally use licensed IDEs such as Kiel, IAR Work Bench, etc, or vendor specific IDEs with licensed Tool chains. The price of each such licensed IDE is around 10-20 dkk just for single-user and a bit more for multi-user license. If you are the one in such above mentioned category of OEMs, then you can consider opting for a kit that uses free one like AVR Studio/WinAVR or a kit with licensed one which is compatible to their already owned IDE used for their other running projects. This cuts the price paid for licensed ones and if multiple developers are involved in the project, still more cost is cut by getting rid of taking multi-user license.

### Protocol Analyzer

Next is about additional cost for Protocol Analyzer (i.e. sniffer). Some vendors give proprietary network analyzers which are much simpler than third party analyzers such as Daintree and Peryton. Based on the user application needs or network features to be observed, purchase of third party analyzers can be considered. Secondly, if the third ones are needed, then going for a kit which is also a compatible hardware to the third ones software, do even cut some more cost. For example, if Jennic kit is bought, then it can even act as sniffer hardware to the Daintree software. There lies no need to buy Daintree Network Adapter instead.

## Cost for Support

Third is about cost for Support during development. Some vendors support without any time limitation (and this statement is inferred from non-listing of any price to be paid for the validity of support-time-period and from my past experience). But some vendor kits' have additional costs for Support, based on time span of assistance. Obviously, it is more dedicated professional assistance in all aspects compared to the former ones which usually don't charge. If you are the one expecting such dedicated professional support, then you need to know which of the vendors provide such and the price to be paid 1 or 1 ½ year support and further for its extension.

## Software Stack

All of the applications do not need to be built on ZigBee stack and is somehow dependant on the scalability of the network expected with the devices which are getting connected. If your requirement is to have sufficient number of devices communicating within direct range, then it is good enough to build applications on top of 802.15.4 layers instead of ZigBee; henceforth, opt for a kit giving source code for 802.15.4 MAC/PHY Layers & good API (i.e. in the sense with good programming guide & easy interface to HAL layer). Software stack for these two layers is often termed as "Open MAC". In case, if there is a need of better network management & security and more network scalability; then it is obvious for you to first opt for a kit which is compliant to the latest specification called as "ZigBee Pro" version. ZigBee stack is specifically composed of Network layer, application support sublayer (APS) and further on top public profiles are provided by the vendors (note: only few are providing).

Even though it's mentioned as "development kit is compliant to the ZigBee stack", it may not mean that APS layer is completely implemented within the software package. Apparently, this is a very rare case and is observed in the earlier first version of ZigBee kits; but it has to be kept in mind. OEMs are suggested to see whether it's completely implemented software package and if lucky enough, any public application profiles are included too.

## Test Tools

Vendors provide different tools that could help to evaluate the kits. Out of such, one is RF testing tool. Firstly, not all vendors provide it. Secondly, even if they have it, it is provided only after request from the OEM, as a source code in some of the cases. Some other vendors who provide it directly with the kit are usually like GUI tools. Not sure whether one can make any flexible RF settings in the GUI software given directly, but the source code given under request can definitely be more suitable to change settings as per evaluation needs.

Time factor during product development is always important for fast prototyping or developing. In order for this, the basic resources (provided within the development kit) that could be beneficial in saving time and investment during development are summarized further.

## Software

Some of the following issues related to software can be assumed as minor issues for the developers. But, since these are commonly observed in most of the ZigBee kits available in the market, they should also be considered before opting for a kit.

To start with the Integrated Development Environment (IDE), general usability features in it can also influence the debugging time. For example, all IDEs do not show what a particular bit in that register corresponds to. Therefore you need to refer the data sheet every time. Having a naming convention of the bits in the IDE does help to some extent. Some IDEs do not support 'Go to Function' option when cursor is positioned at prototype of function and right clicked. With this option, you can directly go to the function where it is defined. The other alternative search option, though it is bit time consuming is 'Edit □ Find in Files'. Not all IDEs have the former, but the second one also doesn't really work in some event though that option is there; thus irritating every time to manually open each and every file to see where that function is defined (i.e. for example in AVR studio). These can be considered as minor usability defects in IDE.

## Debugging

Coming to facilities provided in the kit for debugging other than advanced features of IDE, the indications such as LCD display, serial data transfer, more LED indicators, etc on the ZigBee development boards, can also help to speed up the debug process.

Secondly, it is better to have as many sample applications as possible with good documentation. All vendors do not provide any automated software to generate code for Hardware Abstraction Layer (HAL). So, it is worth enough if the kit provides sufficient number of sample code examples for the available digital peripherals on the board.

Next set of considerations are about the software tools provided in addition to the general software which all vendors provide. Two of such tools which can be assumed to be important, are mentioned further.

## Boot Loader/ AT command set

First is the Boot Loader/ AT command set, which facilitates to upload the firmware via air either directly to the node or through the network (i.e. with its open air software update capability). It reduces the upgrade costs of the product or sensor devices after installation in large networks, pilot testing, etc.

## GUI/ PC interface tool

Next one is source code generating GUI/ PC interface tool. Usually, OEMs' programming template or their own application code, are all made with the assistance of source code from the sample applications given in the kit. But a faster way is to have a GUI tool generate these source code templates; which can thus considerably reduce the time to write it either from the scratch or from the sample applications given. Presently, some of the kits are already providing such tools for the two layers such as Application Profiles and Hardware Abstraction Layer (HAL). Profile builder GUI tool for ZigBee Public Application Profiles such as Home Automation Profile, Smart Energy Profile, etc, can be considered as one of the most important for faster development and henceforth, may be the reason for higher price tags. A tool for HAL is not usually present in all of the kits and is also the user interface software to automatically generate code in the background, while configuring GPIOs (digital peripheral settings) on the MCU or other hardware options.

## Support

Development kit Support Team should be the first ones committed to help the developers at maximum extent in the shortest time possible. Usually, support team of every vendor advertises to provide assistance within some 'x' period of hours which may not be the same from one another (i.e. for example 24 hours, 72 hours, 5 working days, etc.). So you can consider response-time-period also as one of the selection criteria. Some vendors even keep a track of case progress in their web support interface, which do help in giving the up-to-date status. And this logging & tracking cases facility is not usually seen with all vendors and, for now atleast seen with those who charge for dedicated professional support based on time period. Besides this they even support in other aspects like RF design, external hardware interfacing, etc. So if you are particular about support based on some of such above mentioned factors, then you may prioritize the kits based on it.

The other possible way of getting support is from User Forums. Not all vendors officially maintain it for their wireless product division or development kit platforms. Sometimes these user forums

- avoid the need of asking & waiting from support center. Sometimes, in-depth queries do actually take around 1 week response-time (from some vendors); rather within few days from the members of the forum;
- do actually help to participate in active discussions with other developers besides support team, who use the same kit;
- give a quick overview of the problems, which most of the others usually come across.

So it is advisable for the vendors to maintain forums and for you to consider this just as an option to save time while problems are encountered. On the other side, you should also be aware of giving out confidential information to the competitors.

## Documentation

Out of all kit documentation, Application Programmer's Interface (API) Guide is the main source code supporting document that has to be in easy understandable and fast interpretable way. It gives an overview about the programming style, stack call routines, conventions of the functions used within stack, etc. Though the following issues about it and user guide, are minor things for the developers, they can be considered for the selection criteria of development kits.

- Interpretation of Programming Environment: Detailed explanation of API functions is always needed, but in order to get the basic understanding of networking using these functions, a visual representation is usually easier to understand. Not all kits provide this; and most of them give textual description instead. For example, figure representation with communication between two nodes (i.e. Coordinator on left and End Device on right) using API functions right from the network formation by Coordinator till End Device leaving network, is the most advisable for fast interpretation.
- Style of Documentation: This is some what peculiar point to be noted. Probably everybody wouldn't have seen 'dOxygen' style of documentation. It's usually used to see how all the source, header files and functions within the program are linked to each other i.e. all of the descriptions are hyperlinked. It is a user friendly and fast interpretable way of documentation, but very rarely observed in development kit

guides. This style, helps to easily navigate back and forth to the functions and their dependencies; avoiding the need of scrolling number of pages.

- Documentation for tips to reduce the stack usage or memory resources like ram, eeprom & flash memory and step by step explanation of all call routines, atleast for one or two sample applications given in the kit; are advisable.

For every sample application given in the kit, a user guide must be provided in-detail to help the developer in getting an overview about using the boards or a feel of the software. User guide is the one, which tells about the setups, events to be done, outcomes seen step by step. If this is not given, in general, it consumes more time for the basic feel through of the development kit. It is suggested to give step by step explanation of events & outcomes.

## Conclusion

During the investigation of the different vendors of wireless dev. kits, we have gained a comprehensive knowledge about what to look for in a good development suite. The features described are included to give a wide scope of possible issues for consideration. Out of all, IDE, source code generating software tools, software stack, test tools and dedicated support can be considered as common issues for most of the OEMs.