RF Transceiver Selection Guide

Which hardware and firmware is right for my application?

FlexiPanel

Contents

Summary

FlexiPanel offers a wide variety of RF transceiver solutions to suit different applications. *We do this to keep your product simple!* This approach works better than a one-size-fits-all policy, which results in very complex products full of features you don't need.

Most of our modules can operate in a number of modes depending on the onboard firmware configuration. Product selection often not only involves choosing a transceiver module but also choosing the firmware that it runs. Firmware provided by FlexiPanel is free for use with our modules. There are no hidden licensing or registration costs.

The key considerations that impact transceiver selection are:

- **Communications Standard:** The ground rules by which devices communicate. The protocol will dictate the types of device that the module will be able to interoperate with.

- **Hardware Platform:** The electronic transceiver module chosen (Pixie, UZBee, etc). This will affect its range, how you can talk to it, which firmware options are possible and, of course, the price paid.

- **Communications Modes:** How devices address each other (broadcast, point-to-point, etc).

- **Power Management:** Defines when and how devices are allowed to sleep in order to save power.

- **Application Interface:** How your application integrates with the transceiver.

Hardware Selection Matrix

		EasyBee	Pixie	Pixie Lite	UZBee	LinkMatik	Toothpick
Comms Standard	IEEE802.15.4	✓	✓	✓	✓		
	ZigBee	✓	✓	✓	✓		
	MailBox		✓	✓			
	Bluetooth					✓	✓
	Remote Client						✓
Comms Mode	Slave	✓	✓	✓	✓	✓	✓
	Broadcast	✓	✓	✓	✓		
	Addressed	✓	✓	✓	✓	✓	✓
Application Interface & Firmware	API		MAC API ZigBee Stack MailBox API	MAC API ZigBee Stack MailBox API			Toothpick Svcs
	Command	(Resident)	PixieMAC Z-Pixie Gateway	PixieMAC Z-Pixie	MACdongle Zongle	(Resident)	Slave Stamp Edition
	Transparent		StarLite PixieCOM	StarLite	StarLite USB	(Resident)	
	Off-the-shelf		Pixie DARC Switcher	Pixie DARC Switcher	Sniffer SwitcherPC		DARC-I DARC-II HappyTerminal OpenTooth BlueLock BlueControl

Firmware Selection Matrix

	Comms Std	Application Interface	Hardware	Applications Examples
MAC API	IEEE 802.15.4	API	Pixie, Pixie Lite	Sensor Network IEEE 802.15.4 Interoperation
EasyBee resident firmware		Command	EasyBee	Custom products
PixieMAC			Pixie, Pixie Lite	Sensor Network IEEE 802.15.4 Interoperation
MACdongle			UZBee	PC PAN IEEE 802.15.4 Interoperation
StarLite		Transparent	Pixie, Pixie Lite	Serial Cable Replacement, Bus Cable Replacement PC PAN
StarLite USB			UZBee	
15.4 Sniffer		Off-The-Shelf		Packet sniffing
Microchip ZigBee Stack	ZigBee	API	Pixie, Pixie Lite	ZigBee Interoperation
Z-Pixie		Command		
Zongle			UZBee	
Pixie Switcher		Off-The-Shelf	Pixie, Pixie Lite	Switching
Switcher PC			UZBee	
Mailbox API	MailBox	API	Pixie, Pixie Lite	Sensor Network
Pixie Gateway		Command		PC PAN Sensor Network
PixieCOM		Transparent		Bus Cable Replacement PC PAN
Pixie DARC		Off-The-Shelf		Data acquisition / remote control
Toothpick Services API	Bluetooth	API	Toothpick	Custom Product Development
Toothpick Slave		Command		
Toothpick Stamp Edition				
LinkMatik resident firmware		Command / Transparent	LinkMatik	Serial Cable Replacement PC PAN
DARC-I		Off-The-Shelf	Toothpick	Data acquisition / remote control
BlueLock				Access Control
BlueControl	Remote-Client	Off-The-Shelf	Toothpick	Remote control
DARC-II				Data acquisition / remote control
HappyTerminal				Terminal emulator
OpenTooth				Access Control

Communications Standards

The communications standard (or *protocol*) governs how devices communicate, and so who a module can interoperate with. The protocol also specifies how other services work such as error correction and security.

Some protocols specify 'profiles' which define, to some extent, the nature of the data being conveyed. This allows for interoperability between manufacturers.

All protocols introduce some concept of hierarchy among devices, with a 'master' or 'coordinator' creating a network, and 'routers', 'end devices' and 'slaves' that participate within it.

The different standards have various licensing requirements. All devices must be certified as compliant for use within an administrative region. Certification is expensive. FlexiPanel devices are supplied certified FCC and CE compliant, so this involves no additional cost to you.

In addition, some standards require registration with industry bodies, and others require devices to be allocated a globally unique identification number. FlexiPanel's philosophy is to absorb these bulk costs. Our RF transceivers are pre-qualified and pre-licensed, with no extra costs. (Unless you customize them beyond the scope of their original intended use.)

IEEE 802.15.4

Strengths: Simplicity, low cost.

Weaknesses: Network range, interoperability.

The IEEE 802.15.4 communications protocol is a low power, low data rate communications protocol. (Practically speaking, approximately 38.4 kbaud in FlexiPanel products.)

Communication is single-hop between up to 65K devices. Messages may be broadcast and any node can address any other.

Devices are allowed to sleep, but in doing so they must rely on the central coordinator to cache messages for them, and their ability to participate in direct communications with other devices will be limited.

The protocol is low cost and easy to implement. It is the protocol of choice if greater complexity is not required.

No profiles are defined by the IEEE 802.15.4 standard. Data is simply transferred as a payload in a packet which may be up to 127 bytes, including packet addressing headers.

Each IEEE 802.15.4 device must be assigned a unique MAC address. IEEE 802.15.4 products from FlexiPanel are pre-assigned MAC addresses. In some instances it is not possible to store a MAC address on the product when shipped and you will need to request an allocation of MAC addresses from us.

ZigBee

Strengths: Network range, low power, interoperability.

Weaknesses: Complexity, evolutionary nature.

ZigBee extends the IEEE 802.15.4 protocol to multi-hop ('mesh') communications. This adds to the addressing header load, but considerably extends the range of the network by allowing messages to be forwarded on behalf of other devices.

Devices may sleep, but if they do, they cannot forward messages on behalf of other devices. The availability of mains-powered routers does, however, provide the possibility of an extensive network where most devices may be battery powered for several years, if not for the shelf life of the battery.

In order to implement multi-hop message routing and to devolve decision making, an underlying tree structure is defined by a Stack Profile. General message routing is not restricted by the tree structure. However, the Stack Profile does define rigid quantities such as the number of children a parent device may have, and how many hops a device is allowed not be from the coordinator. ZigBee devices employing different Stack Profiles are not mutually compatible.

The ZigBee Alliance, which defines the ZigBee Specification, emphasizes interoperability between manufacturers. Public profiles are defined in detail through industry-wide consensus. This has the advantage that it encourages an official standard, but these standards sometimes prove slow to come to fruition and thus become rather evolving standards.

'Private' profiles, defined by individual manufacturers, are permitted within the ZigBee specification, at the cost of registering those profiles with the ZigBee Alliance and

demonstrating conformance with the standard. Publication and free use of such profiles is encouraged in the interest of allowing *de facto* standards to emerge on merit.

Providers of ZigBee hardware are required to be members of the ZigBee Alliance, at a financial cost and a burden of proof of interoperability. Where FlexiPanel provides a complete firmware solution, it meets these costs.

MailBox

Strengths: Simplicity, network range, low power.

Weaknesses: Interoperability, program size

The MailBox protocol is a free-use private ZigBee profile developed by FlexiPanel to provide generic data communications over a ZigBee network. Its purpose is to simplify application development where the application does not fit into any of the existing public profiles. It allows applications to be rapidly implemented and deployed. For example it is much easier to use the MailBox API than the ZigBee stack. Also, it avoids incurring any ZigBee Alliance-related registration costs.

The MailBox protocol ensures data will arrive at the right place at the right time, but it is up to the application to know what to do with it. MailBox in not an industry-defined standard, but we make it freely available and encourage its use, license free.

Bluetooth

Strengths: High data rates, interoperability.

Weaknesses: Network size & range, higher power.

The Bluetooth communications protocol is intended for moderately high data rates (practically speaking, approximately 500 kbaud in FlexiPanel products).

Bluetooth is also well established in consumer products. To communicate with existing mobile phones, headsets, PDAs, etc, it is the favorable choice. Power saving modes exist, but battery powered operation is not a serious option unless frequent recharging is possible.

Communication is single-hop and limited to direct unicast conversations between a master and up to seven slaves. (FlexiPanel products allow up to four slaves only.)

A number of profiles are available for Bluetooth, including SPP (serial data), SCO (audio), DUN (dial-up networking, *i.e.* sending commands to a phone) and OPP (file transfer).

Providers of products employing Bluetooth are required to become Adopter Members of the Bluetooth Organization, however, this is free. FlexiPanel's products are ore-qualified as Bluetooth compliant and unique MAC addresses are pre-assigned under license.

Remote Client

Strengths: Rapid application development

Weaknesses: Limited flexibility.

An interesting profile developed for the Bluetooth protocol by FlexiPanel Ltd is the FlexiPanel Remote Client Interface. This is a user interface server which allows user interfaces to be created on remote devices such as a PC, PDA or mobile phone using generic client software on the remote device. No software development on the remote client is required. This allows for rapid development of advanced applications.

That said, generic client firmware has its limitations and it lacks the slick look-and-feel required for high-volume professional and consumer products.

Hardware Platforms

FlexiPanel provides several transceiver modules to address different communications protocols and applications interfaces. All include integral antennas and have FCC/CE certification.

EasyBee

EasyBee is a pre-qualified IEEE 802.15.4 transceiver based on the CC2420 chip. It may therefore be used as part of an IEEE 802.15.4, ZigBee or MailBox system. It has no on-board microcontroller, allowing you to add the microcontroller of your choice, but the burden of providing the firmware will be up to you.

Pixie

Pixie is an EasyBee with a PIC microcontroller added. Microchip Technology provide a free ZigBee stack complete with source code, making this a very compelling solution for ZigBee and Mailbox applications.

Pixie Lite

Pixie is an EasyBee with a low-end PIC microcontroller added. It is the optimum solution for IEEE 802.15.4 devices and reduced function ZigBee devices. It costs about 20% less than the Pixie.

UZBee

UZBee is an EasyBee with a USB-enabled PIC microcontroller added and housed in a USB adapter casing, complete with USB plug. It has roughly the same capabilities as Pixie Lite, plus the USB service. It is an ideal solution for gateways to PCs and from there to the internet. It also makes a nice packet sniffer application.

The current UZBee does not have enough memory to be a ZigBee coordinator or MailBox device. We're working on that.

LinkMatik

LinkMatik is a Bluetooth transceiver capable of supporting SPP (with up to 4 connections), SCO, DUN and OPP profiles. These profiles are controlled using a command interface. It is also possible to configure it so it can operate as a single-connection transparent serial device; indeed this is the configuration in which it is shipped.

Toothpick

The LinkMatik is a powerful radio but it lacks programmability. Toothpick addresses this by combining it with a PIC 18LF6722 microcontroller, which has many interfaces such as SPI, serial, A/D and PWM. It also contains a Remote Client user interface server, making it a powerful tool for developing autonomous Bluetooth applications.

Various off-the-shelf firmware options are available for Toothpick. Most are focused on data acquisition and remote control and are ideal for one-off sensor and control applications.

Communications Modes

The more ambitions a radio communications network gets, the harder it gets to specify who is to be talking to whom. Some effort goes into avoiding the need to specify a recipient where possible. This allows 'transparent' interfaces to be developed, where only data transferred between the application and the transceiver; commands are not required.

Slave

Slaves communicate with one other device only Depending on the implementation, that might be:

- the master or coordinator
- a parent router
- the last device to send a message to it (mirroring)
- as instructed by another device (redirecting)
- as dictated by non-volatile ROM settings
- as dictated by I/O settings

Slaves are useful for implementing simple devices in a multi-device network, since the slave does not need to decide who to send messages to.

Broadcast

Broadcast devices transmit data simultaneously to all devices in a multi-device network. No recipient address needs to be specified. Any addressing issues must be solved at the application level. This is useful for working with legacy systems, for example RS485 cable replacement systems. It is, however, rather wasteful of resources in multi-hop systems if most applications are going to ignore most messages.

Generally, broadcast messages are not acknowledged. This would normally be done at the application level.

Addressed

Addressed communications are directed towards specific devices. In the case of multi-hop communications, the network will determine the appropriate route to the destination.

Addressing adds a lot of complication. It is not really possible to implement a transparent applications interface if an address needs to be specified. This is not just because addresses need to be specified, but because devices need to work out which address belongs to which device. The MailBox application goes some way to address this by assigning addresses according to the function that a device performs rather than according to its location within a network.

Power Management

The radio receivers on these devices are very sophisticated in order to maximize reception quality. The drawback is they consume as much

power in receive mode as transmit mode – approximately 25mA. The only way so significantly reduce power is to negotiate sleeping periods.

Sniff Park and Hold

Bluetooth sleep modes where devices can sleep for short (sub-second) periods while occasionally polling to determine if they need to exit sleep mode.

Rx-Off-When Idle

Some IEEE 802.15.4 devices, including Mailbox and ZigBee end devices, can sleep indefinitely. The parent stores messages for them.

If such a device doesn't need to receive messages, it can sleep until it has a reason to transmit. This is why, for example, a ZigBee light switch can last practically indefinitely on a tiny battery.

Devices which might need to receive a message must poll their parent frequently enough to respond in a timely manner. The more responsive the device, the greater the power consumption. For example an LCD readout might need to update once per second in a status indicator application, or once per day in a supermarket shelf pricing application.

Network Sleep

The MailBox communication standard allows the entire network to negotiate synchronous sleep periods. This is useful where routers and/or the coordinator need to be battery powered.

Application Interface

The application interface is how the transceiver communicates with your application.

API Interface

The lowest cost approach is to integrate the application firmware on the transceiver module's microcontroller in tandem with the communications stack. This requires an Applications Programming Interface ('API'). This highly integrated approach requires some firmware development tools and skills. Our APIs are C-language based.

Command Interface

In a command interface, instructions are sent, usually via a serial port, to the transceiver. This offers full flexibility to integrate with any application environment, but it is not as low cost as an API.

Transparent Interface

A transparent interface is possible for some devices that do not need to specify destination addresses or custom settings. In a transparent interface, there are no commands; only data passes between the application and the transceiver. This is useful for cable replacement in legacy systems. The costs are no greater than a command interface, but it comes at the price of less flexibility.

Off-The-Shelf Applications

An off-the-shelf application is one which uses the API interface to develop a complete modular solution where there is popular demand.

Firmware Descriptions

The following firmware is available free of charge for FlexiPanel RF products. Most can be downloaded from *www.FlexiPanel.com*. If you do not buy firmware preloaded on a device, you must program the firmware in. For Toothpick products, this is possible via the Bluetooth link; for UZBee products, via the USB link. Pixie and Pixie Lite need to be programmed using a PIC programmer such as the ICD2 form Microchip Technology.

API Interfaces

MAC API is an IEEE 802.15.4 API interface for Pixie and Pixie Lite.

The *Microchip Stack for ZigBee* is an API with full source code which may be used to develop ZigBee applications on Pixie and Pixie Lite. It is provided and supported by Microchip Technology and can be downloaded from *www.microchip.com*.

MailBox API is an API interface for the MailBox protocol for Pixie and Pixie Lite.

Toothpick Services API is a Bluetooth / Remote Client API interface for Toothpick.

Command Interfaces

EasyBee resident firmware uses an SPI-based command interface.

PixieMAC is an IEEE 802.15.4 command interface for Pixie and Pixie Lite.

MACdongle is an IEEE 802.15.4 command interface for UZBee. It is also used by FlexiPanel's Sniffer software for sniffer applications.

Z-Pixie is a ZigBee command interface for Pixie and Pixie Lite.

Zongle is a ZigBee command interface for UZBee. Currently it can support reduced function devices only. It is also used by FlexiPanel's Switcher PC software for PC-based ZigBee switching.

Pixie Gateway is a MailBox command interface for Pixie.

LinkMatik resident firmware uses a serial-based command interface.

Toothpick Slave is a command interface for Toothpick.

Toothpick Stamp Edition is a command interface for Toothpick specifically intended for accepting commands form BASIC Stamps. It is available exclusively from Parallax Inc (*www.parallax.com*).

Transparent Interfaces

Transparent devices automatically connect at power-up and are then ready to transfer serial.

StarLite is a transparent IEEE 802.15.4 interface for Pixie and Pixie Lite. It can be configured in a number of slave or broadcast modes.

StarLite USB is a transparent IEEE 802.15.4 interface for UZBee.

PixieCOM is transparent MailBox interface for Pixie and Pixie Lite. It can be configured in a number of slave or broadcast modes.

LinkMatik can be turned into a transparent master or slave device using one-time configuration commands.

Off-The-Shelf Applications

Many of the off-the-shelf applications are provided complete with source code to allow them to be customized to specific requirements.

Pixie Switcher is a ZigBee switching application for Pixie and Pixie Lite.

15.4 Sniffer is an IEEE 802.15.4 / ZigBee / MailBox packet sniffing application for UZBee.

Switcher PC is a ZigBee switching application for UZBee.

Pixie DARC is a MailBox data acquisition and remote control application for Pixie and Pixie Lite, complete with source code.

DARC-I is a command-driven data acquisition and remote control application for Toothpick, complete with source code.

BlueControl is a remote control application for Toothpick. It is normally supplied as the complete BlueControl product, but the source code is available on request.

DARC-II is a remote-client -driven data acquisition and remote control application for Toothpick, complete with source code.

HappyTerminal is a remote-client -driven TTL-level terminal emulator application for Toothpick, complete with source code.

BlueLock is an access control application for Toothpick. It is normally supplied as the complete BlueLock product, but the source code is available on request.

OpenTooth is a remote-client-driven access control application for Toothpick, complete with source code.

Applications Examples

Here are some common applications and appropriate hardware / firmware choices.

Serial Cable Replacement

In a serial cable replacement application, a PC communicates wirelessly with one other remote device.

If large amount of data is to be transferred, consider a Bluetooth adapter on the PC and a LinkMatik on the remote device. (Check the range of the adapter – some are only 10m.)

If the remote device is to be battery powered, consider StarLite USB on the PC and StarLite on the remote device.

Bus Cable Replacement

In a bus cable replacement application, devices communicate wirelessly using broadcast

messages. Data send input at any device will be output by all devices.

If possible, StarLite and/or StarLite USB should be used since they are very efficient at broadcast messaging.

If extended range is required, PixieCOM may be used, although this can lead to considerable network traffic and so needs to be implemented with care.

If extended range is required, but the destination address is actually in an identifiable position in the data, consider a custom application to imitate a bus, but where the data is actually unicast only to the required destination.

PC PAN

In a PC PAN, a single node on a PC (or PDA) can communicate with a number of individual remote devices.

If large amount of data is to be transferred, consider a Bluetooth adapter on the PC and a LinkMatik on the remote devices. Care should be taken to select a USB Bluetooth adapter for the PC which can handle multiple connections. The upper limit will be seven remote devices. (Check the range of the adapter – some are only 10m.)

If a greater number of devices or lower power is required, consider a StarLite USB on the PC and a StarLite on the remote devices.

If a multi-hop range is required, consider Pixie Gateway on the PC, via a USB to TTL serial adapter, and PixieCOM on the remote devices. The USB to TTL serial adapter makes the PC side a bit clunky, but in due course we hope to upgrade UZBee so you will be able to use that instead in future.

Sensor Network

In a sensor network, a large number of intelligent devices interact together.

If a low-cost solution is required, consider using an API-based solution such as MAC API or MailBox API.

For rapid integration into existing systems, consider using a command-based solution such as PixieMAC or MailBox Gateway. This allows you to develop the application within your existing host device.

IEEE 802.15.4 Interoperation

To develop applications that will interoperate within IEEE 802.15.4 networks, consider using the MAC API on Pixie devices, PixieMAC for command-driven nodes and MACdongle on UZBee for PC interfaces.

ZigBee Interoperation

To develop applications that will interoperate within ZigBee networks, consider using the Microchip Stack for ZigBee on Pixie devices, Z-Pixie for command-driven nodes and Zongle on UZBee for PC interfaces.