wireless

EZMac Software MAC Layer Module

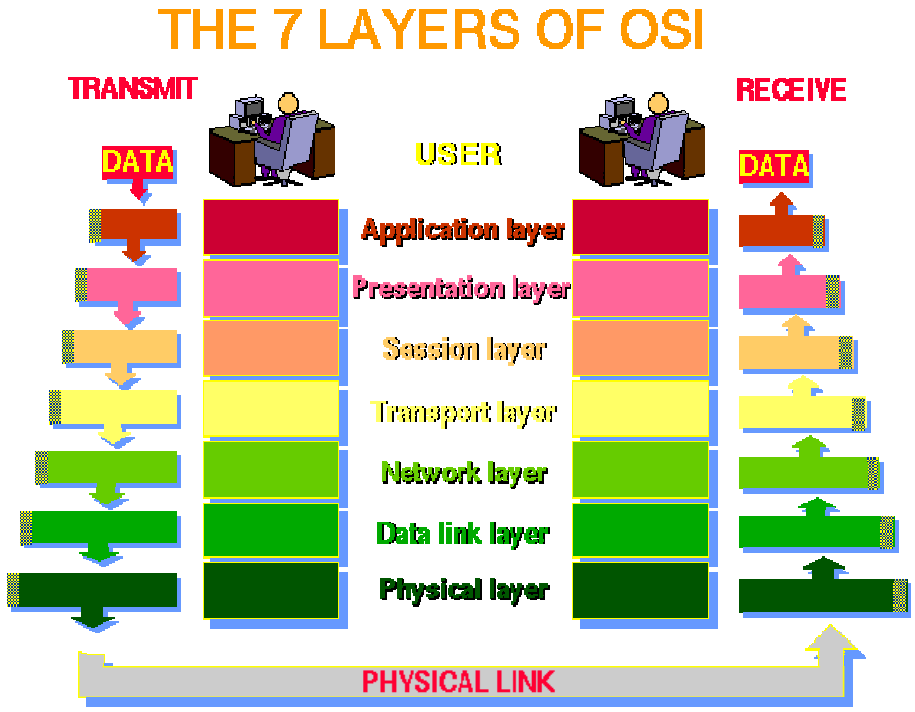
INTEGRATION

CONFIDENTIAL

Module: RF8 v1.0r

What is EZMac? (1)

EZMac is realizing the two lowest layers of the OSI model



- **Physical layer:**
  This layer conveys the bit stream – in this case radio signal - through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining the physical aspects.
  Based on IA4420/21
- **Data Link layer:**
  At this layer, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sub layers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer.
- The **MAC sub layer** controls how a node on the network gains access to the data and permission to transmit it.
- The **LLC sub layer** controls frame synchronization, flow control and error checking.

What is EZMac? (2)

- In simple embedded communication applications in most cases not all of the ISO layers are used
- Usually only up to the transport layer is used
- Sometimes even the network layer is not needed

Only the application layer sits above EZMac

- "EZ" because it is **Easy** to
  - Learn
  - Build into an existing application
  - Use

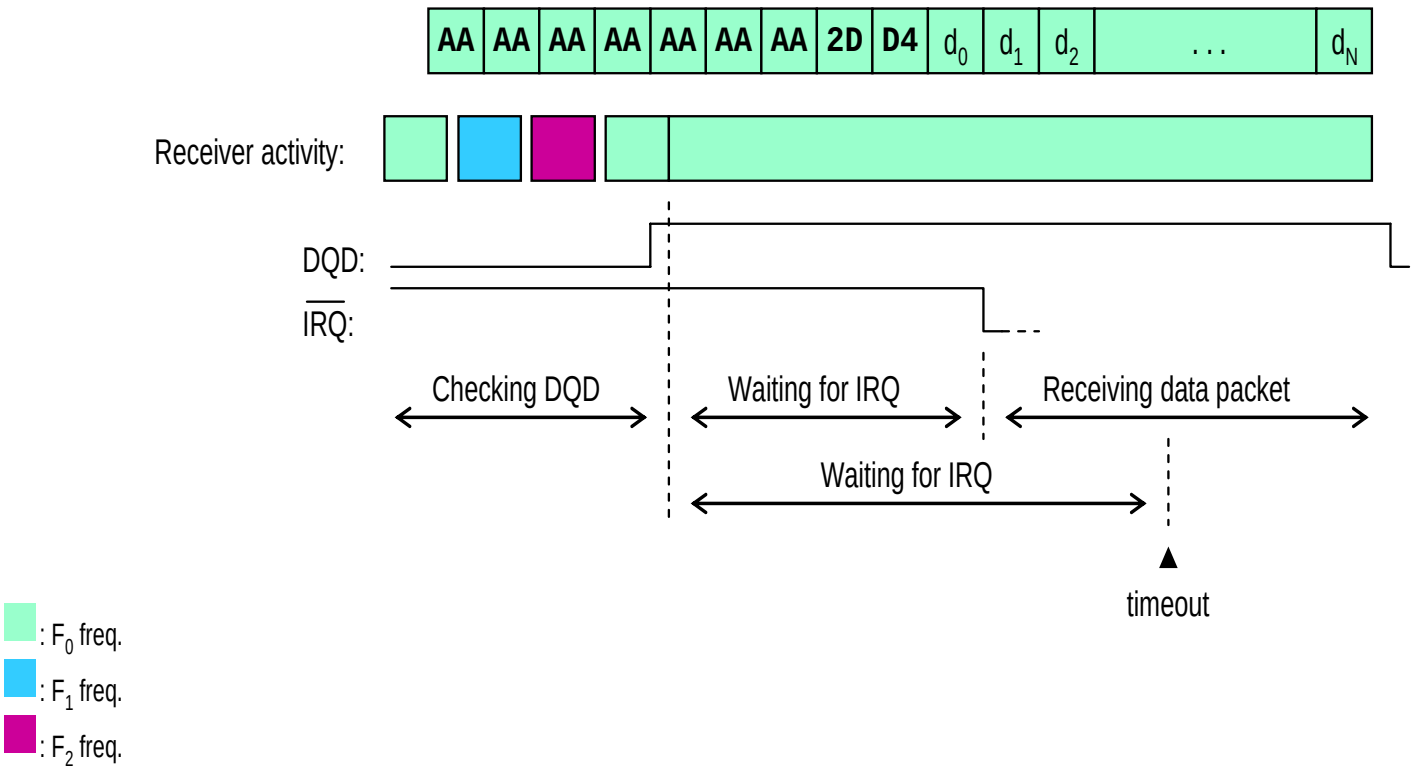
EZMac Benefits

- Software interface **hides the RF layer**
- Dramatically **reduces the development time**
- Half duplex packet transmission between RF nodes
- Flexible addressing (direct, multicast, broadcast)
- Frequency hopping
- Multiple error detection
- Real time packet filtering

Frequency Hopping (1)

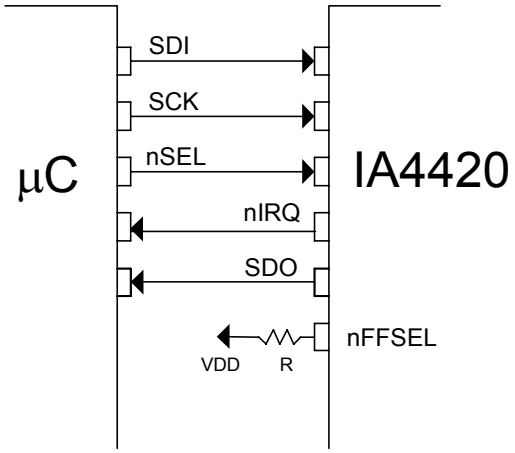
- EZMac uses automatic hopping only during receiving
- Transmit hopping is the task of the application layer
- While receiving, it scans all of the enabled channels in a circular manner.
- Uses the EZRadio DQD signal to check the channel

Frequency hopping (2)



Hardware Requirements

- 5 I/O pins (including IRQ pin)
- One 16 bit timer
- Internal RC oscillator
- F_{osc} >= datarate * 250



Memory	PIC16F… parts	PIC18F… parts
ROM	3368 words	5752 bytes
ROM, including unused segments	3607 words	5752 bytes
RAM, at main() level	111 bytes	121 bytes
RAM, worst case	142 bytes	152 bytes

Software Interface (1)

- Implemented as a state machine
- Precompiling options:
  - Select frequency band (315 / 434 / 868 / 915MHZ)
  - Select data rate (9.6 / 19.2 / 38.4 / 57.6 / 115.2kbps)
- Running in the **BACKGROUND** in two interrupt routines (one timer ISR and one external ISR)
- Control and Status registers: 8-bit variables

  To set up:
  - Used frequencies, Addressing mode, Filters, Error detection

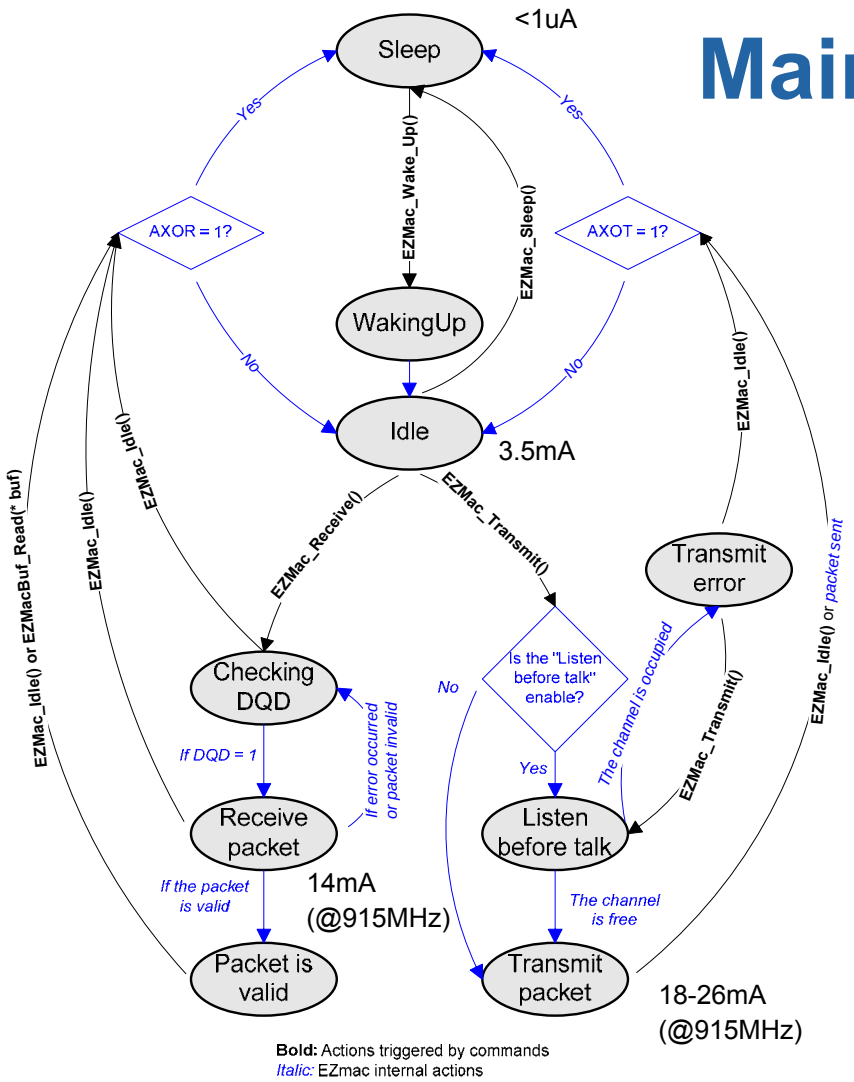
  To read:
  - EZMac status
  - Received packet properties (eg.:sender. addr., data, error counters etc.)

Software Interface (2)

Commands: the following C functions:

- char EZMac_Wake_up (void)
- char EZMac_Sleep (void)
- char EZMac_Transmit (void)
- char EZMac_Receive (void)
- char EZMac_Idle (void)
- char EZMacReg_Write (MacRegs Name, char Value)
- char EZMacReg_Read (MacRegs Name, char* Value)
- char EZMacBuf_Write (char* buf)
- char EZMacBuf_Read (char* buf)

Main States of EZMac

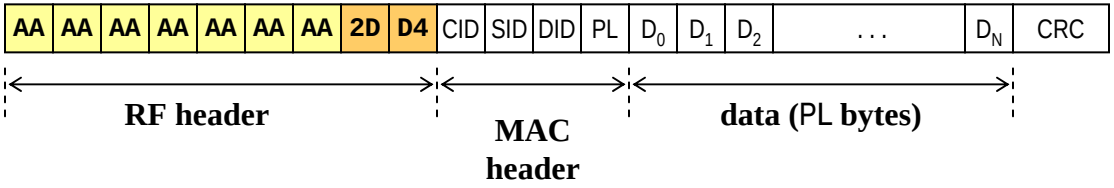


- The Power Consumption of the Physical Layer is also controlled
- Supports sleep mode
- Adjustable RF output power (range)

Addressing Modes

- Two 8-bit address fields for sender and destination addresses in normal addressing mode
- The network topology should be defined in the application layer
- In reduced addressing mode the address fields are half length
- In network addressing mode there is one 8/16bit long address field to address the destination network (broadcast message in a given network)
- Plus 1 byte address field: Customer ID (CID)
  Issued by Integration, increasing the address space.
  Optional, but recommended to increase the reliability

Packet Configuration

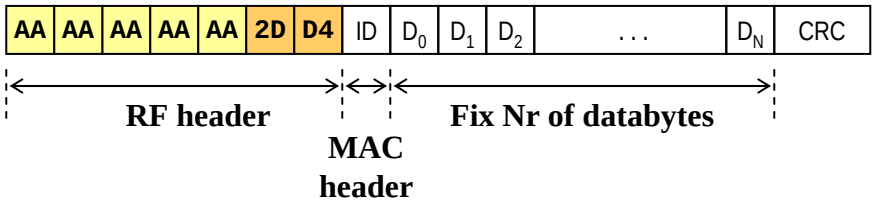


Name	Description	Size (bytes)
AA AA AA	Preamble	min. 3
2D D4	Synchron pattern	2
CID*	Customer ID	1
SID**	Sender ID	1
DID**	Destination ID	1
PL*	Packet Length	1
$D_0..D_N$	Data bytes	0...16
CRC	Cyclic Redundancy Code	2

*: optional fields

**: In reduced address range these IDs can be implemented on a single byte

Reduced Packet Configuration

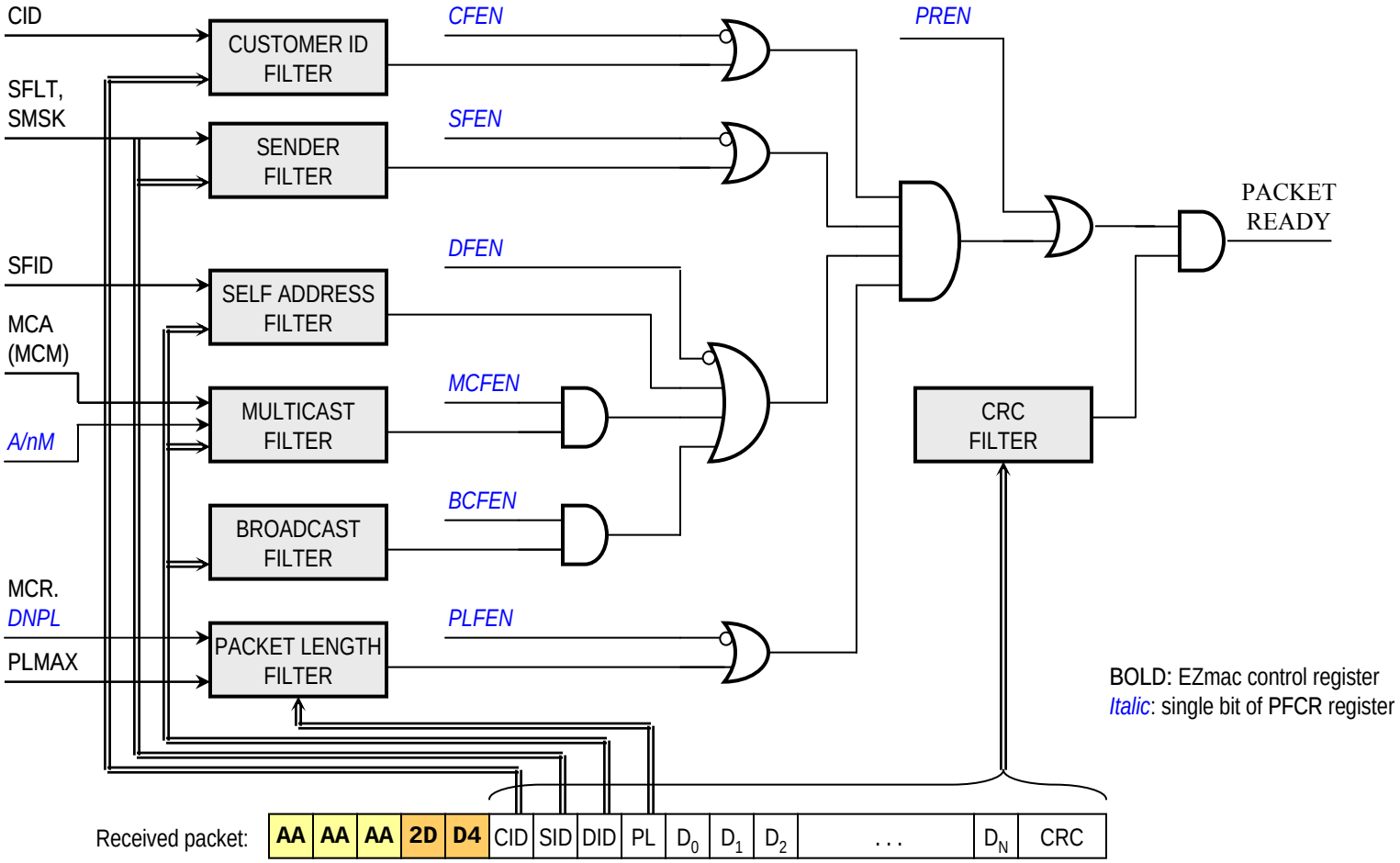


MAC header:

SI.3	SI.2	SI.1	SI.0	DI.3	DI.2	DI.1	DI.0

SI: Sender ID (0…15)
DI: Destination ID (0…15)

Real-Time Packet Filtering

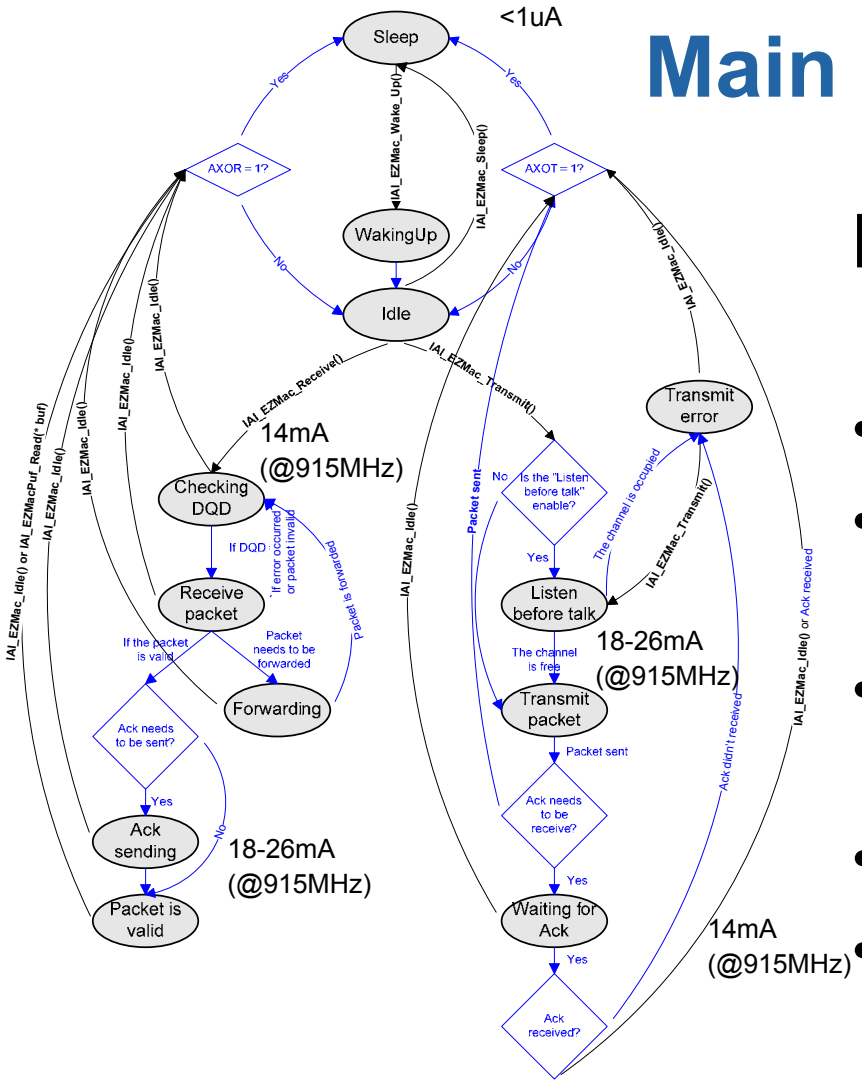


EZMac Plus

What is EZMac Plus?

- EZMac with several new features:
  - Automatic acknowledgement message sending
  - Packet forwarding capability
  - Improved Listen Before Talk function
- Additional header byte (Control byte)
  - Packet forwarding request
  - Acknowledgement request bit
  - 4 bit Packet ID
- Incompatible with the original EZMac (packet structure)

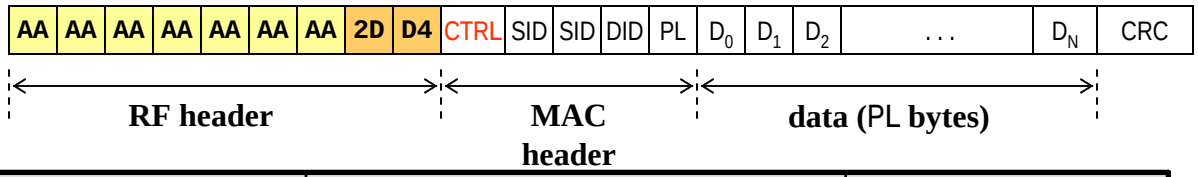
Main States of EZMac Plus



New states of the EZMac Plus:

- Waiting for acknowledgement
- Acknowledgement transmission
- Acknowledgemenet receiving error
- Listen Before Talk error
- Packet forwarding

Packet Configuration (new header byte)

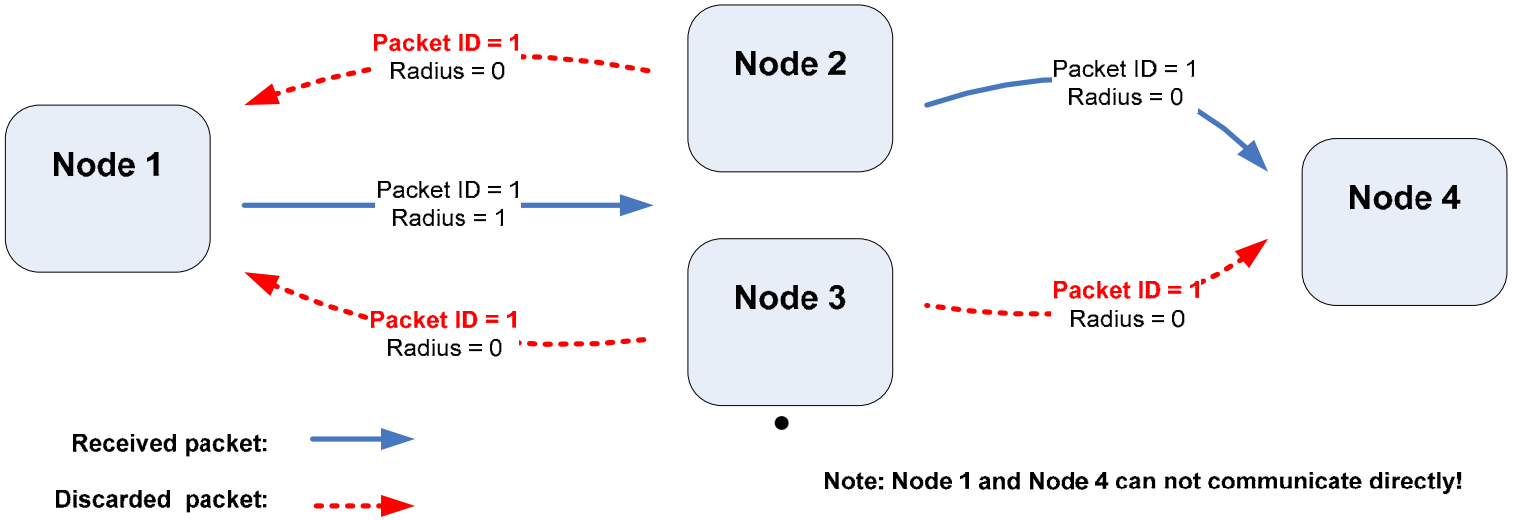


Name	Description	Size (bytes)
AA AA AA	Preamble	min. 3
2D D4	Synchron pattern	2
CTRL	**Control byte**	**1**
CID*	Customer ID	1
SID**	Sender ID	1
DID**	Destination ID	1
PL*	Packet Length	1
$D_0..D_N$	Data bytes	0...16
CRC	Cyclic Redundancy Code	2

*: optional fields

**: In reduced address range these IDs can be implemented on a single byte

Packet forwarding



- The node decreases the Radius field before forwarding the packet
- If the Radius is 0, the node doesn't forward the packet
- Every node stores the packet ID and the source address of the received packets → it doesn't forward the same packet again.

EZMac comparison 1.

EZMac features	EZMac Lite	EZMac	EZMac Plus
Support all power states (sleep, idle, rx, tx)	●	●	●
Normal address mode (each address field is 8bit)	●	●	●
Reduced address mode (each address filed is 4bit)	-	●	●
Network address mode	-	●	-
Multicast addressing	-	●	-
Error detection – error counters	-	●	●
Dynamic packet length	-	●	●
Fix packet length	●	●	●
ARSSI measurement	●	●	●
Packet filtering	●	●	●
Automatic acknowledgement	-	-	●
Packet forwarding	-	-	●
Listen before talk	●	●	● Note 1
Compatible with	EZMac	EZMac Lite	None

Note1: it uses a modified Listen before talk, which provides better performance.

EZMac comparison 2.

EZMac features	EZMac Lite	EZMac	EZMac Plus
Supported processor families	PIC16, PIC18	PIC16, PIC18	PIC18
Supported compilers	Hi-tech PIC	Hi-tech PIC, CCS-C	Hi-tech PIC, CCS-C
Code size (Hi-tech PICC – PIC16 family)	3841 byte ROM 150 byte RAM	4625 byte ROM 156 byte RAM	-
Code size (Hi-tech PICC18)	3218 kword ROM 174 byte RAM	4555 kword ROM 180 byte RAM	5982 kword ROM 204 byte RAM
Code size (CCS-C PIC16 family)	-	4337 byte ROM 131 byte RAM	-
Code size (CCS-C PIC18 family)	-	3208 kword ROM 141 byte RAM	4095 kword ROM 162 byte RAM

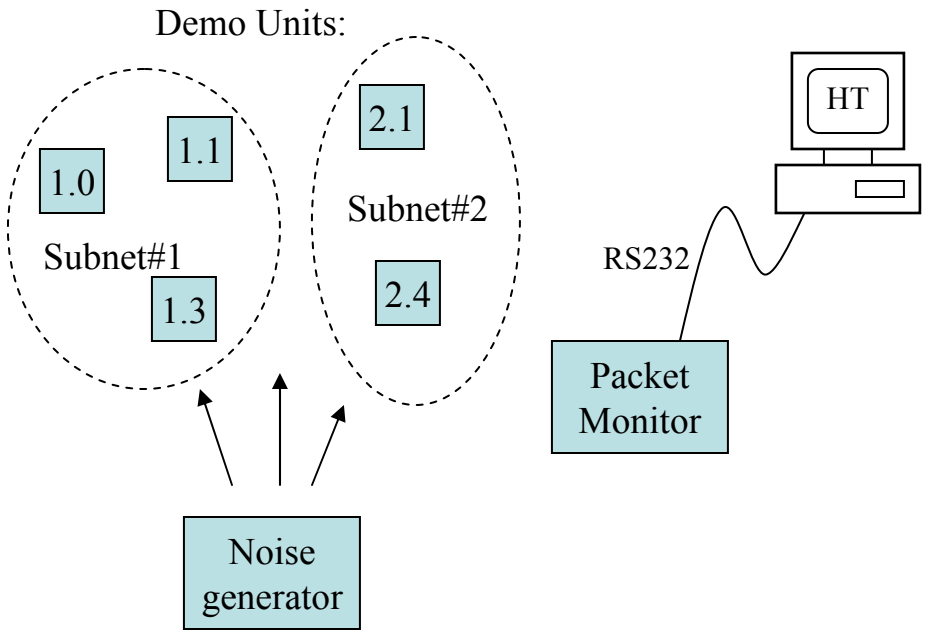
EZMac Network Demo

Demonstration

- Virtual light control
- It uses the standard EZMac
- Units are configured into groups (subnets)
- Commands: On, Off, Toggle
- Control strategy:
  - Direct control (single unit)
  - Group control (several unit)
  - System control (all the available units)

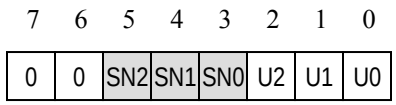
Communication Network

- Units are using 3 frequencies
- Packet Monitor is scanning all the frequencies, reporting to the Terminal program
- Noise generator can block the frequencies

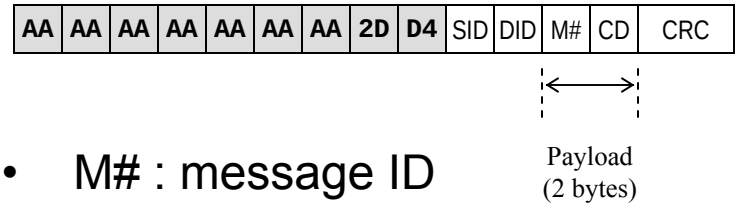


Packet Configuration

- Address bytes are split to Subnet and Unit fields:



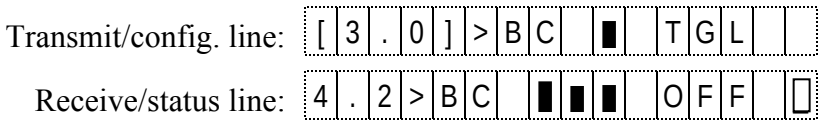
- Fixed packet length, 2 data bytes:



- M# : message ID
- CD : command ID

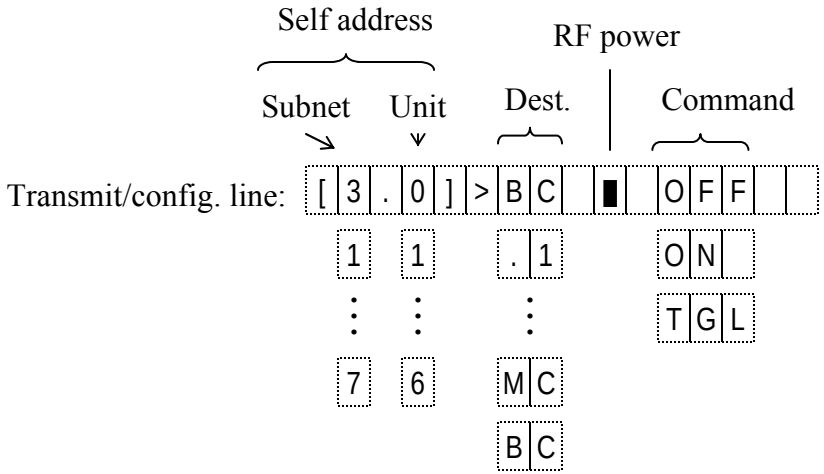
Demo Unit Operation

- 3 buttons, 2x16 character LCD
- Adjustable parameters:
  - Destination address
  - Command selection
  - Button1: parameter select
  - Button2: parameter set
  - Button3: send packet
- 2line LCD:



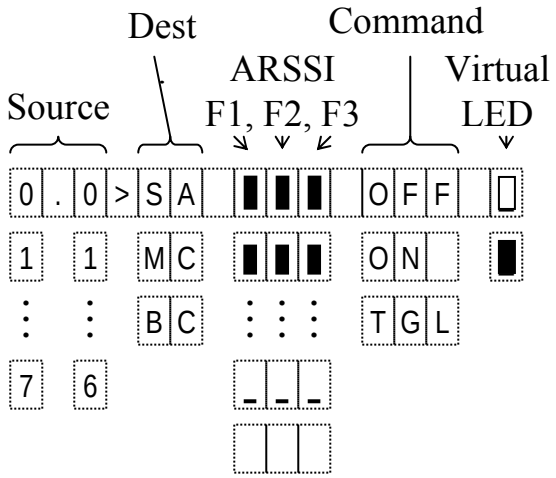
Demo Unit Operation II

- Transmit line:
  - Destination address and the Command can be set by buttons
  - Self Address (Subnet and Unit ID) is set by DIP switches
  - Unit ID ‘7’ is the group (multicast) address



Demo Unit Operation III

- Receive line:
  - Informs about the last received packet
  - Shows the status of the virtual LED
  - Indicates the strength of the received RF signal of each frequency (ARSSI)



Monitoring

- Uses any standard ASCII Terminal
  - Continuously scans the frequencies for valid packets
  - Reports on packet receiving
  - Indicates the strength of the received RF signal of each frequency (ARSSI)