**XBee communication module:**

*Provides functions to transmit, receive and interpret data using the XBee radio transmitter and the E128 SCI sub-system*

**Module level variables (statics)**

Receive complet flag

Receive packet index

Receive packet array

Receive packet buffer array

Transmit packet array

Transmit index

Waterlogged flag

**XBee\_Initialize:**

*Initializes the e128 SCI settings*

Use the SCI1CR registers to set the following

One pin operation

SCI Enabled in Wait Mode

RXD and TXD NOT connected internally

Eight Bit Data

Address mark wakeup

Idle character bit count beings after stop bit

No Parity

Transmit Interrupt Enabled

NO Transmit Complete Interrupt

Receive Interrupt Enabled

NO Idle Line Interrupt

TXD Enabled

RXD Enabled

NO Receiver Wake Up

**SCI1 Interrupt:**

*Deals with the receive buffer full and transfer buffer clear interrupts.*

Read status and data registers to clear interrupt flag and prepare for next byte transfer

If it is a receive buffer full interrupt

Put incoming data byte into next space in receive\_packet array

Increment receive index

If receive index = 2 then set framelength = received data

If receive index is bigger than the length of the message

Set receive complete flag

Reset receive index to 0

Copy receive packet into receive packet buffer

If the interrupt was a transfer reg clear

Put the next byte from transmit packet onto data-bus

Increment transmit index

If transmit index > packet length

Reset transmit index to 0

Turn off TX clear interrupt (next transfer initiated will turn this back on)

**XBee\_Transmit:**

*Takes in variables that encode the DCH controls and initiates transfer of the packet*

Set the values of the bytes in the transmit array

byte0 = Xbee start byte (0x7e)

byte1 = length MSB = 0

byte2 = length LSB = 8

byte3 = 1 (API send message code)

byte4 = unique packet id (incremented every transmission)

byte5 = boat address MSB

byte6 = boat address LSB

byte7 = 0 (options)

byte8 = outgoing data

byte9 = outgoing data

byte10 = outgoing data

compute checksum as 0xff – sum(bytes3-10) store that in byte11

read status and data registers to set for next transfer

turn on transfer reg clear interrupt

write first databyte to transfer reg

increment transmit index

**XBee\_Receive:**

*When called this function checks if a new message has been received and if so interprets it*  if the receive complete flag is not set the return

else clear receive complete flag

compute the checksum, print error message if it doesn’t match

If message is broadcast message return

If message in buffer is status message

Switch on 4th byte in buffer, print appropriate message to screen

If message is targeted then switch on the first data byte

(the first data byte contains either the pot ID number or a specific data code)

Cases:

Red team

Green team

White team

Waterlogged

Set waterlogged flag

Fixed

Clear water logged flag

Unloaded

Add crabs on boat to crabs on shore

Crabs on boat = 0

Start of season

Verification

For all the preceding output appropriate message on LCD screen

PotID

Display pot ID, accumulation rate, total crabs on the boat, total on shore.