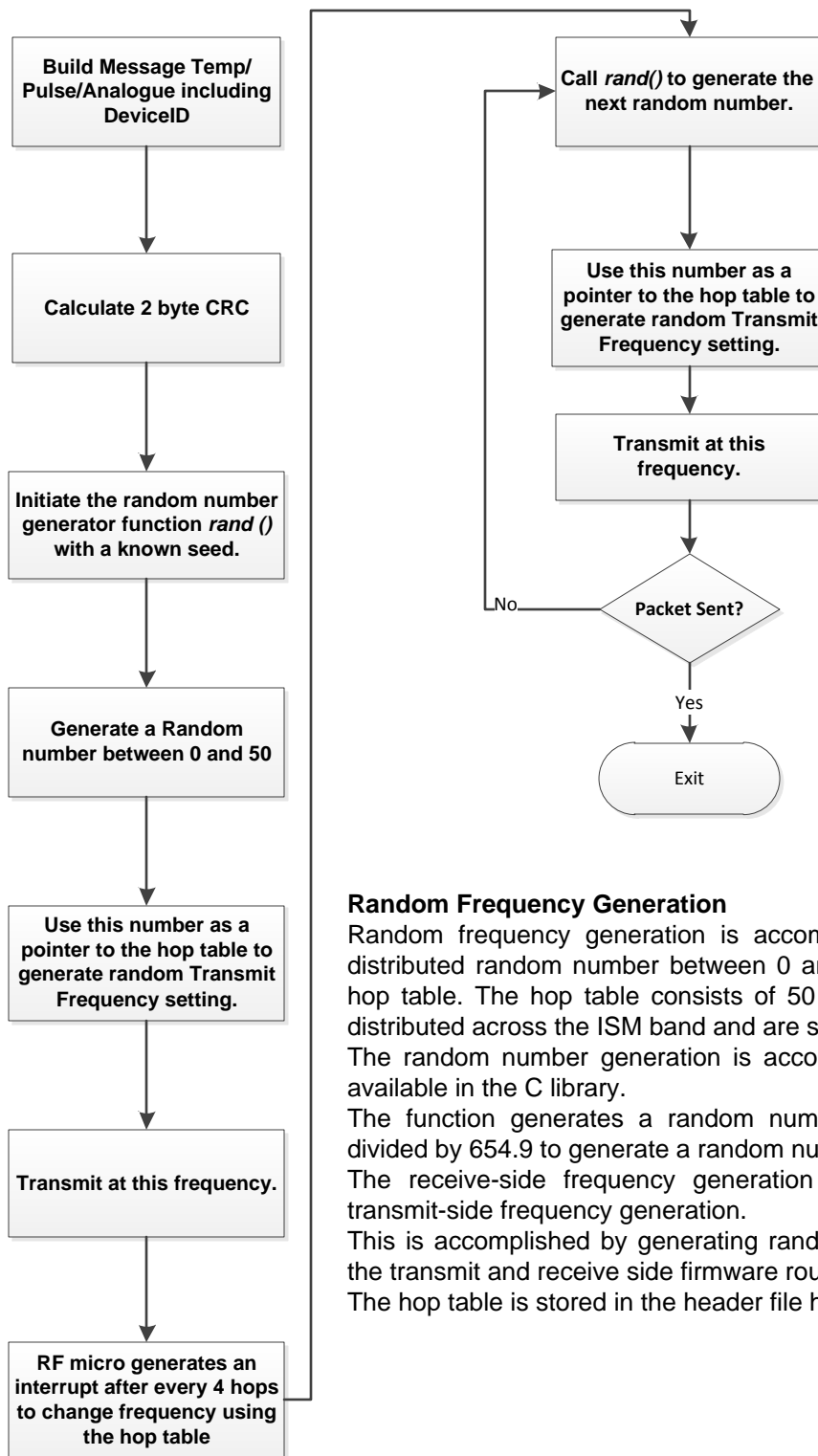# Transmit frequency hopping algorithm

Every 5 minutes from powering up the device it transmits it's payload to the receiver. The pre-amble detect sequence is initiated on the channel 0. The hopping sequence commences when the receive detects the pre-amble on channel 0. the payload is then sent over a random channel sequence defined below.

```
┌─────────────────────────┐              ┌─────────────────────────┐
│ Build Message Temp/     │              │ Call rand() to generate │
│ Pulse/Analogue including│              │ the next random number. │
│ DeviceID                │              │                         │
└───────────┬─────────────┘              └───────────┬─────────────┘
            │                                         │
            ▼                                         ▼
┌─────────────────────────┐              ┌─────────────────────────┐
│ Calculate 2 byte CRC    │              │ Use this number as a    │
│                         │              │ pointer to the hop table│
│                         │              │ to generate random      │
│                         │              │ Transmit Frequency      │
│                         │              │ setting.                │
└───────────┬─────────────┘              └───────────┬─────────────┘
            │                                         │
            ▼                                         ▼
┌─────────────────────────┐              ┌─────────────────────────┐
│ Initiate the random     │              │ Transmit at this        │
│ number generator        │              │ frequency.              │
│ function rand () with a │              │                         │
│ known seed.             │              └───────────┬─────────────┘
└───────────┬─────────────┘                          │
            │                                         ▼
            ▼                                     ◇ Packet Sent? ◇ ──No──┐
┌─────────────────────────┐                          │                  │
│ Generate a Random       │                         Yes                 │
│ number between 0 and 50 │                          ▼                  │
└───────────┬─────────────┘                     ( Exit )               │
            │
            ▼
┌─────────────────────────┐
│ Use this number as a    │
│ pointer to the hop table│
│ to generate random      │
│ Transmit Frequency      │
│ setting.                │
└───────────┬─────────────┘
            │
            ▼
┌─────────────────────────┐
│ Transmit at this        │
│ frequency.              │
└───────────┬─────────────┘
            │
            ▼
┌─────────────────────────┐
│ RF micro generates an   │
│ interrupt after every 4 │
│ hops to change frequency│
│ using the hop table     │
└─────────────────────────┘
```

Random Frequency Generation

Random frequency generation is accomplished by generating a uniformly distributed random number between 0 and 50 using this as a pointer to the hop table. The hop table consists of 50 channels frequency pairs uniformly distributed across the ISM band and are spaced approximately 50 KHz apart.

The random number generation is accomplished using the function *rand ()* available in the C library.

The function generates a random number between 0 and 32768; this is divided by 654.9 to generate a random number between 0 and 50.
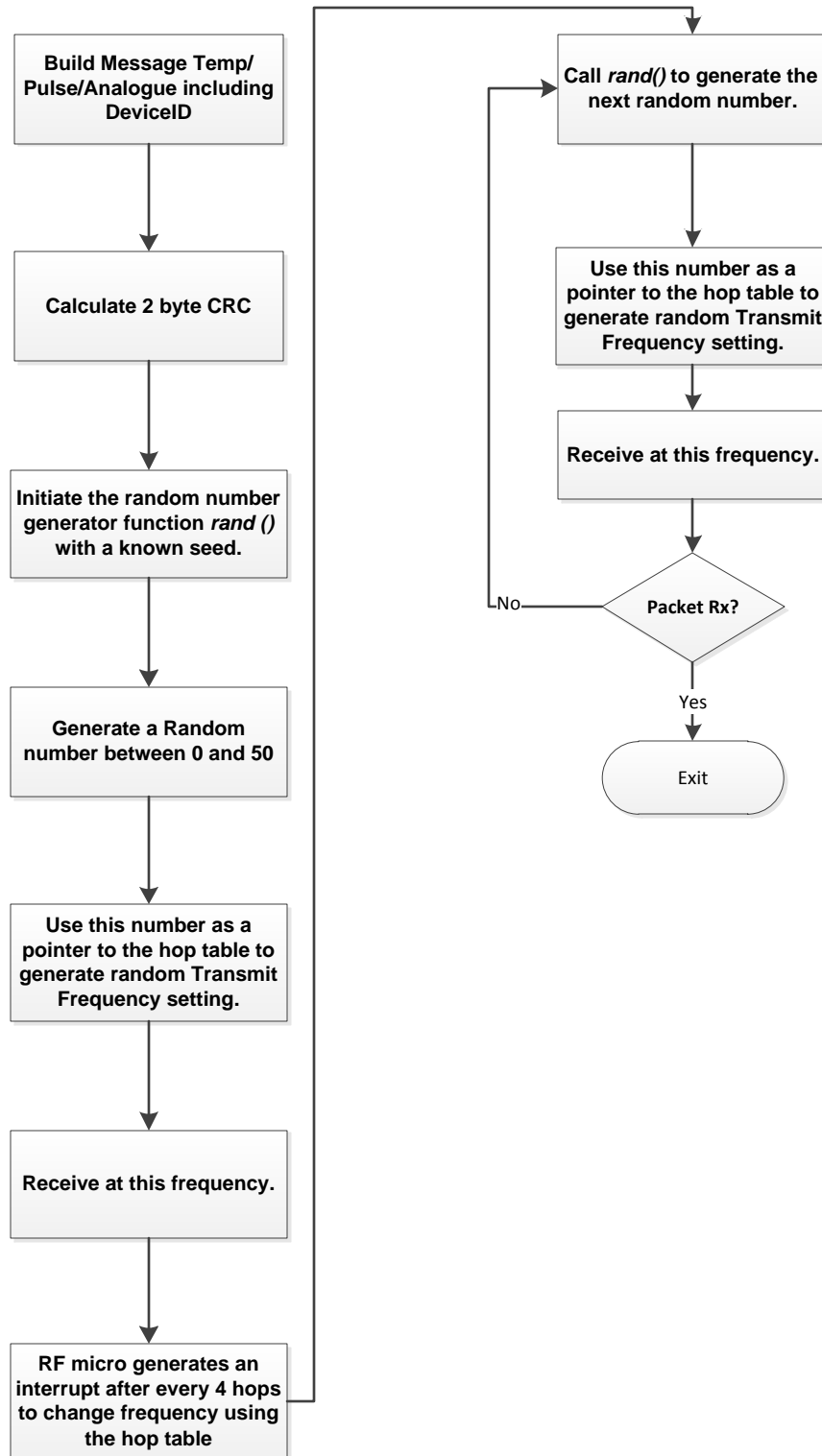
The receive-side frequency generation has to be synchronous with the transmit-side frequency generation.

This is accomplished by generating random numbers *with the same seed* in the transmit and receive side firmware routines.

The hop table is stored in the header file hop.h in the firmware.

Receive frequency hopping algorithm

The receiver Hops to the same frequencies in the same random order as the transmitter. This is accomplished by calling the *rand ()* function with the same seed in both transmit and receive routines. The receiver changes frequency once a Pre-amble is detected . The time between each hop is the same on both transmit and receive as they are both using the same delay routine and same device platform.

```
Build Message Temp/
Pulse/Analogue including
DeviceID
        |
        v
Calculate 2 byte CRC
        |
        v
Initiate the random number
generator function rand ()
with a known seed.
        |
        v
Generate a Random
number between 0 and 50
        |
        v
Use this number as a
pointer to the hop table to
generate random Transmit
Frequency setting.
        |
        v
Receive at this frequency.
        |
        v
RF micro generates an
interrupt after every 4 hops
to change frequency using
the hop table
```

```
Call rand() to generate the
next random number.
        |
        v
Use this number as a
pointer to the hop table to
generate random Transmit
Frequency setting.
        |
        v
Receive at this frequency.
        |
        v
    < Packet Rx? >  --No-->
        |
       Yes
        |
        v
     ( Exit )
```

Hop Table

```
const long HoppingFrequencies[] =
        {
        911000000,
        921000000,
        921500000,
        925500000,
        917000000,
        914500000,
        918500000,
        924000000,
        903500000,
        908000000,
        925000000,
        903000000,
        920000000,
        922500000,
        907000000,
        909000000,
        907500000,
        920500000,
        904000000,
        916500000,
        917500000,
        910500000,
        926500000,
        927500000,
        905000000,
        904500000,
        918000000,
        912000000,
        912500000,
        923500000,
        905500000,
        906500000,
        909500000,
        913000000,
        916000000,
        924500000,
        906000000,
        926000000,
        908500000,
        919000000,
        919500000,
        911500000,
        915500000,
        915000000,
        913500000,
        927000000,
        923000000,
        922000000,
        910000000,
        914000000,
        911000000
        };
```