# General description

The profile is designed for the ED1608 Full version; The Profile is used for tracking objects, sending measurement data and detecting rotation and movement.

The unit can be configured by sending commands via the downlink channel..

Tracker Sensor

The unit has two main states:

* Not Moving, or Idle; The unit will detect motion via the accelerometer every second and after 20 (default) seconds of motion it will go to the moving state
* Moving: The unit will try to get a GPS fix and send it every minute,

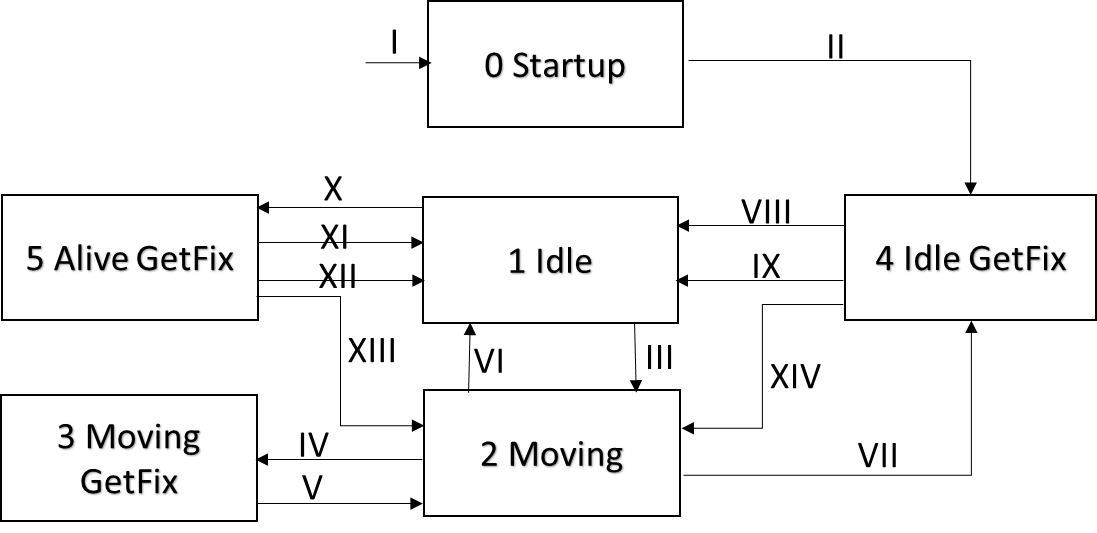
In the idle state all sensors, the CPU and the GPS receiver are switched of, only the accelerometer is active. It tries to detect motion and notifies the CPU immediately when motion is detected.. This is extremely energy efficient and uses less than 30 uA.

# Formal description

The ED1608 Tracker Sensor operates as a “State Machine” it has XXX formal states:

|  |  |  |  |
| --- | --- | --- | --- |
| State # | State | Action |  |
| 0 | Startup | Initial | Initialize ED1608 and wait 10 seconds |
|  |  | Continuous | Goto IdleState |
| 1 | Idle | Initial | Disable GPS |
|  |  | Continuous | Check accelerometer for Moving  Check Alive Timer |
| 2 | Moving | Initial | Enable GPS when configured to do so |
|  |  | Continuous | Check accelerometer for Idle  Goto Moving GetFix every 1 (configure) minute |
| 3 | Moving Getfix | Initial | Enable GPS |
|  |  | Continuous | Wait for GPS Fix (Max 4 minutes) |
| 4 | Idle GetFix | Initial | Enable GPS |
|  |  | Continuous | check accelerometer for Moving  Wait for GPS Fix (Max 4 minutes) |
| 5 | Alive Getfix | Initial | Enable GPS |
|  |  | Continuous | check accelerometer for Moving  Wait for GPS Fix (Max 4 minutes) |

The states and their transitions are shown in the following figure:



The state transitions are defined as follows:

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | Startup | II | 4 | Idle Getfix | Condition | After 10 seconds  Sent MsgIDReboot  Sent MsgIDAlive |

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Idle | III | 2 | Moving | Condition | Accelerometer detects movement for #DTMove seconds  Sent Message MsgIDStart |
|  |  | X | 5 | Alive GetFix | Condition | Every 6 hours |

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 2 | Moving | VI | 1 | Idle | Condition | No more movement for #DTIdle seconds AND  Valid GPS Fix |
|  |  | VII | 4 | Idle GetFix | Condition | No more movement for #DTIdle seconds AND  No Valid GPS Fix |
|  |  | IV | 6 | Moving GetFix | Condition | Every #UTMoving minutes |

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Idle GetFix | XIV | 2 | Moving | Condition | Movement detected for #DTMove seconds |
|  |  | X | 1 | Idle | Condition | After #TOGPSFix seconds  Sent Message MsgIDStopNoFix |
|  |  | VIII | 1 | Idle | Condition | GPSFix found then wait 20 seconds  Sent Message MsgIDStopFixOk |

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | Alive GetFix | XIII | 2 | Moving | Condition | Accelerometer Detects Movement for #DTMove seconds  Sent Message MsgIDStart |
|  |  | XI | 1 | Idle | Condition | After 4 minutes  Sent Message MsgIDGPSAbort |
|  |  | XII | 1 | Idle | Condition | GPSFix found  Sent Message MsgIDAlivePos |

| Start State | | Transition | End State | | Description | |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | Moving GetFix | V | 2 | Moving | Condition | After #TOGPSfix minutes  Sent Message MsgIDGPSAbort |
|  |  |  |  |  | Condition | GPSFix found  Sent Message MsgIDMovingFix |

Traveled Distance

The ED1608 General Sensor profile when tracker is enabled registers the traveled distance. This distance will be used for future trip registration and be reported in a to be developed Trip Message (Future)

Note: The distance meter uses the Geofence functionality to determine the interval of traveled distance calculation. Setting the #Geofence to “0” will disable traveled distance registration.

Vibration Sensor

The vibration sensor detects vibration of the ED1608. It registers the frequency and the amplitude of the three most intense vibrations in the spectrum.

The vibration detection is able to detect frequencies up to 650Hz.

The unit scans for vibrations at a certain interval #ScanIntVibrate, when #ScanIntVibrate is set to ‘0’ no vibration is detected anymore

When a Vibration is detected the unit can send an alarm message (if #EnVibrAl is set to 1, default is Off). The alarm will be send again if the frequency of the vibration changes.

The unit will periodically sent an Vibration Message. The period can be changed via Parameter #UTVibrating.

After sending an alarm message and after sensing the vibration message the vibration is set to ‘0’. So the next message contains the most recent vibration data.

Running hours

During vibration the unit accumulates running hours.

Rotation Sensor

The ED1608 General Sensor profile has two Rotation Sensors on board.

1. The magnetic rotation sensor, detects changes in the magnetic field and sends a rotation or an alarm message when a change is detected. If the Parameter #RotMagAl is On, the Sensor will sent an Alarm message otherwise the Sensor will sent a Rotation Message.  
   When the Parameter #RotMagnSens is set to “0” detection is switched off.  
   The sensitivity of the magnetic rotation sensor can differ a lot on different locations. The default Value will work in most circumstances. But can be changed via a command.
2. The Gravity rotation sensor, detects changes in the earth’s gravity field and sends a rotation or an alarm message when a change is detected. If the Parameter #RotGravAl is On, the Sensor will sent an Alarm message otherwise the Sensor will sent a Rotation Message.  
   When the Parameter #RotGravSens is set to “0” detection is switched off.  
   The sensitivity of the gravity rotation sensor can be specified in degrees (1..90) . After 1 minute of absolutely no movement at all the sensor sets its reference orientation and starts detecting rotation.

Movement Alarm Sensor

The movement alarm sensor detects movement. After 3 seconds of movement it sends an Alarm Message. The sensor is then deactivated for 30 seconds to 1 minute. After that it will again sent an alarm message

The sensor can be switched off using the #EnMotAlarm command.

Barometer/Temperature/Relative Humidity and Beam Level Sensor

The ED1608 can be configured to send its sensor values on a regular time interval.

The time interval can be configured using the #UTSensors command.

The time is given in minutes. If the parameter #UTSensors is set to “0” no updates are sent anymore

Geofence

The ED1608 can be configured to send GeoFence violations as alarm messages.

Default the messages are Off.

The alarm can be switched on with the #GeoFenceAl Command.

The default value for the GeoFence Radius can be changed also. For this the #GeoFence command is used.

After a GeoFence violation, the center of the GeoFence is set to the new position and the sensor starts monitoring GeoFence violation around this new center point.

In order to detect GeoFence violations the GPS sensor must be on.

NOTE: Setting the #GeoFence to “0”, will disable the GeoFence Alarms also, but it will also disable KM/Mileage registration, since the GeoFence functionality is also used to set a distance for new KM/Mileage calculation.

1Wire Temperature Sensor

1Wire Temperature sensors, type DS1820 can be connected to the ED1608. A special 2 wire connector is available.

The maximum 1 wire temperature sensors that can be connected to the interface is 5.

The 1Wire measurements are sent

LED status

The ED1608 has three LED’s give the following information:

RED-Led

* Fast flashes : LoRa Transmission
* 1 blink per second: Tracker Moving
* 2 blinks per second: Getting GPS Fix
* 3 blinks per second: Vibration detected
* 3 long blinks: SigFox Transmission

GREEN-Led

* Flashes during power up

YELLOW-Led

* Fast flash: LoRa Downlink Message

Parameters

The following parameters can be changed via downlink messages

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Description** | **Factory Value** | **Command to change** |
| #DTMove | Delay to Move | 20 seconds | 0x01 |
| #DTIdle | Delay to Idle | 30 seconds | 0x02 |
| #ScanIntVibrate | Scan Interval for vibration | 0 seconds  (no scan) | 0x03 |
| #TOGPSFix | Time out for GPS Fix | 240 Seconds | 0x04 |
| #UTMoving | Update Time Moving  0..255 Min  0 Means no updates of position while moving | 1 minute | 0x05 |
| #UTVibrating | Update Time while Vibrating | 10 minutes | 0x06 |
| #UTIdle | Update Time while Idle | 6 hours | 0x07 |
| #UTRot | Update Time while rotating detected  <not used yet> | 1 minute | 0x08 |
| #TORot | Time out Rotated state  <not used yet> | 2 minutes | 0x09 |
| #UTSensors | Update time sensor values | 5 minutes | 0x0A |
| #EnMotAlarm | Motion Detection Alarm 1 is enable 0 is disable | 0 (off) | 0x0B |
| #RotMagAl | Rotation Magnetic Detection //1 is enable 0 is disable | 0 (off) | 0x0C |
| #RotGravAl | Rotation Gravity Detection//1 is enable 0 is disable | 0 (off) | 0x0D |
| #TrackerOn | Enable tracker //1 is enable 0 is disable tracker | 0 (off) | 0x0E |
| #TripOn | Enable Trip Reporting //1 is enable 0 is disable tripreporting | 0 (off) | 0x0F |
| #RotMagnSens | Magnetic Rotation Sensitivity  (0 = Off)  (try 100 to actually use it for Rotation Detection) | 0(off) | 0x10 |
| #RotGravSens | Gravity Rotation Sensitivity in Degrees < 90  (0 = Off) | 0 (off) | 0x11 |
| #GeoFence | Geofence Radius in meters / 10 ( 0 = Off) | 5 (=50m) | 0x12 |
| #GeoFenceAl | GeoFence Alarm //1 is enable 0 is disable | 0 (Off) | 0x13 |
| No longer used |  |  | 0x14 |
| #WAGPSFix | Wait after GPS fix (only when going to Idle) | 20 seconds | 0x15 |
| #AccSens | Sensitivity of the accelerometer | 5 | 0x16 |
| #GPSOnMov | GPS On while Moving | 0 (Off) | 0x17 |
| #UT1WireT | Update Time 1 Wire sensors in minutes | 0 (Off) | 0x18 |
| #LPWANUse | Which LPWAN Network to use  **(if enabled)** | 1 (SigFox)  2 (**LoRa)** (def)  3 (SigFox&LoRa) | 0x19 |
| #DRMoving | Datarate while moving (7..12) | 0x0C (SF12) | 0x1A |
| #UseGravXYZ | 1. Sent GenSensMsg 2. Sent GenSensGravMsg | 0 (= sent GenSensMsg) | 0x1C |
| #EnVibrAl | Enable Vibration Alarm 0 or 1 | 0 (No Alarm send) | 0x1D |
| #CmdMonitorOn | Enable the Terminal:  0 = Off (also disables input of commands)  1 = On  2 = No DebugVar | 0 (Off) for Basic  1 (On) for other types | 0x1F |
| #UTAnalog | Update Time Analog Inputs (Including Battery Voltage) in minutes | 0 (Off) | 0x20 |

Commands

Commands are defined by their number as mentioned in the table above.

Commands are sent as downlink messages.

The downlink message format is as follows:

typedef struct {

byte CmdSeq; // Should be incremented every command given

byte Cmd; // Command number as mentioned in table above

byte NewValue; // The new value of the parameter to be changed

} TGenSensCmd;

A command is answered by the ED1608 via a TAliveMsg. The parameter CmdAck in this message containes the last received command.

There are a number of special commands:

**Reboot command**

has the following content:

byte CmdSeq = 0xFE

byte Cmd = 0xFE

byte NewValue = 0xEF

he ED1608 will perform a reboot

**Factory Reset command**

has the following content:

byte CmdSeq = 0xEF

byte Cmd = 0xFF

byte NewValue = 0xFE

The ED1608 will restore parameters to factory default and perform a reboot

**Set APEUI command**

has the following content:

byte CmdSeq = 0x<specified by user>

byte Cmd = 0xFD

byte APPEUI[8] = <specified by user>

The ED1608 will Store the APPEUI and perform a reboot

Example : 01FD0102030405060708 in a downlink sequence will set the

APPEUI to 0x0102030405060708.

The DEVEUI and APPSKEY cannot be changed.

A factory reset will restore the original APPSEUI.

**Set ABP Parameters**

has the following content:

byte CmdSeq = 0x<specified by user>

byte Cmd = 0xFC

byte DevAddr[4] = <specified by user>

byte AppSKey[16] = <specified by user>

byte NwkSKey[16] = <specified by user>

The ED1608 will Store the ABP (Activation by personalization parameters) and perform a reboot

Example : 01FC0102030405060708090A0B0C0D0E0F10111213141516171819101A1B1C1D1E1F20212223

in a downlink sequence will set the

DevAddr = 01020304

AppsKey = 05060708090A0B0C0D0E0F1011121314

NwkSKey = 1516171819101A1B1C1D1E1F20212223

A factory reset will restore the original ABP Parameters (usually empty)

Formal PayLoad description

General Sensor Message Format

This is the latest message format, to fully use the General Sensor capabilities.

There are currently 12 message types

* MsgIDAlive MsgID 0x00
* MsgIDTracking MsgID 0x01
* MsgIDGenSens MsgID 0x02
* MsgIDRot MsgID 0x03
* MsgIDAlarm MsgID 0x04
* MsgID1WireT MsgID 0x06
* MsgIDRunning MsgID 0x07
* MsgIDVibrate MsgID 0x08
* MsgIDAnalog MsgID 0x09
* MsgIdGenSensGravMsg MsgID 0x0A (Same as MsgIDGenSens, but with raw gravity values)
* MsgIdDailyReport MsgID 0x0B
* MsgIDReboot MsgID 0x0E

The messages always contain the most recent data. If no updated data is available the old data is sent. The payloads can be decrypted via the 1M2M Payload decoding JSON service

**http://1m2m.eu/services/GETPAYLOAD?Human=0&PL=***0102096100064f7a3c07a50300000000*

typedef struct {

byte MsgId; // Message Identification Value = 0x00

byte Battery; // 0..255 == 0%..100%

uint8 Profile; // For internal use

uint8 CmdAck; // Sequence number of last received Command

byte GPSFixAge; // bit 0..7 = Age of last GPS Fix in Minutes MsgIDsee above),

byte SatCnt\_HiLL; // bit 0..4 = SatInFix, bit5 Latitude 24 bit 6,7 =

Longitude 24,25

byte Lat[3]; // bit 0..23 = latitude bit 0..23

byte Lon[3]; // bit 0..23 = longitude bit 0..23

}TAliveMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x01

unsigned int Start :1; // Start Message

unsigned int Move :1; // Object Moving

unsigned int Stop :1; // Object Stopped

unsigned int Vibr :1; // Vibration Detected

int16 Temp; // Temperature in 0,01 degC

byte GPSFixAge; // bit 0..7 = Age of last GPS Fix in Minutes,

byte SatCnt\_HiLL; // bit 0..4 = SatInFix, bit5 Latitude 25 bit 6,7 =

Longitude 25,26

byte Lat[3]; // bit 0..23 = latitude bit 0..23

byte Lon[3]; // bit 0..23 = longitude bit 0..23

}TTrackMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x02

byte Status; // Content Depends on Message ID ==for future use

word BaromBar; // Air Pressure in mBar = MsgIDMsgIDBaromBar +100.000)/100)

int16 Temp; // in 0,01 degC

byte Humidity; // Relative Humidity in %

int8 LevelX; // Inverse Sinus of Beam Level in Deg X-Direction -128 =

// -90 Degr .. +127 = +90 Degr

int8 LevelY; // Inverse Sinus of Beam Level in Deg Y-Direction -128 =

// -90 Degr .. +127 = +90 Degr

int8 LevelZ; // Inverse Sinus of Beam Level in Deg Z-Direction -128 =

// -90 Degr .. +127 = +90 Degr

uint8 VibAmp; // Amplitude of Vibration Detected == Future

uint8 VibFreq; // Approx. Frequency of Vibration Detected in Hz

// Future

}TGenSensMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x03

unsigned int GravRotAl :1; // Gravity Rotation Detected

unsigned int MagRot :1; // Mag Rotation Detected

int8 GravX; // Gravity in X-Direction 64 ~~ 1G

int8 GravY; // Gravity in Y-Direction 64 ~~ 1G

int8 GravZ; // Gravity in Z-Direction 64 ~~ 1G

int8 MagX; // Magnetic Field in X-direction 10 uTesla

int8 MagY; // Magnetic Field in Y-direction 10 uTesla

int8 MagZ; // Magnetic Field in Z-direction 10 uTesla

}TRotMsg;

typedef struct {

byte MsgID; // Message Identification Value = 0x04

unsigned int GravRotAl :1; // Gravity Rotation Detected

unsigned int MagRot :1; // Magnetic Rotation Detected

unsigned int MotAlarm :1; // Motion Alarm detected

unsigned int GeoFenceAl:1; // GeoFence Violation Detected

unsigned int VibrAl :1; // Vibration Alarm Detected

int16 Temp; // Temperature in 0,01 Celcius

byte Hum; // Relative Humidity in %

word BaromBar; // Air Pressure in Mbar=MsgIDMsgIDBaromBar +100.000)/100) }TAlarmMsg;

Typedef struct {

byte MsgID; // Message Identification Value = 0x06

byte NumOfSensors; // Number of 1Wire sensors currently connected

word Temp[5]; // Store for temperatures

// bit 0..11 Temperature in 0,1 Celcius + 550

// Temperature range 0 = -55.0C, 1800 = 125.0C

// bit 13..16 ShortID (0..15)

}T1WireTMsg;

typedef struct {

byte MsgID; // Message Identification Value = 0x08

byte MaxdX; // Maximum deviation in AccelerometerX

byte MaxdY; // Maximum deviation in AccelerometerY

byte MaxdZ; // Maximum deviation in AccelerometerZ

byte Max1Freq; // Frequency with highest amplitude

// Frequency = Max1Freq \* 630/53

byte Max1Ampl; // Amplitude of Frequency with highest Amplitude

byte Max2Freq; // Frequency with second highest amplitude

// Frequency = Max2Freq \* 630/53

byte Max2Ampl; // Amplitude of Frequency with second highest Amplitude

byte Max3Freq; // Frequency with third highest amplitude

// Frequency = Max3Freq \* 630/53

byte Max3Ampl; // Amplitude of Frequency with third highest Amplitude

byte vAgcVibr; // Gain Value Vibration Detection 0x00=2G, 0x01=4G,

0x02=8G, 0x03=16G

}TVibrMsg;

typedef struct {

byte MsgID; // Message Identification Value = 0x09

int16 VBat; // Battery voltage in mV

int16 AnalogIn1; // AnalogIn 1 in mV

int16 AnalogIn2; // AnalogIn 2 in mV

int16 AnalogIn3; // future use

int16 Analogin4; // future use

}TAnalogMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x0A

byte Status; // Content Depends on Message ID ==for future use

word BaromBar; // Air Pressure in mBar

int16 Temp; // in 0,01 degC

byte Humidity; // Relative Humidity in %

int8 GravX; // Accelerometer X

int8 GravY; // Accelerometer Y

int8 GravZ; // Accelerometer Z

uint8 VibAmp; // Amplitude of Vibration Detected == Future

uint8 VibFreq; // Approx. Frequency of Vibration Detected in Hz

}TGenSensGravMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x0B

byte Status; // Content Depends on Message ID ==for future use

byte MinTemp; // Minimum Temperature DgrC -27..+100 DgrC since

last DailyRep Message

byte MaxTemp; // Maximum Temperature DgrC -27..+100 DgrC since

last DailyRep Message

byte MinHum; // Minimum Humidity since last DailyRep Message

byte MaxHum; // Maximum Humidity since last DailyRep Message

byte MaxBaro; // Maximum Baro since last DailyRep Message

byte MinBaro; // Minimum Baro since last DailyRep Message

word RunHrs; // Running Hours (in hours)

word KM; // Distance traveled in KM

}TDailyRepMsg;

typedef struct {

byte MsgId; // Message Identification Value = 0x0E

byte RebootReason; // For internal use

uint8 Profile; // For internal use

uint8 CmdAck; // Last received Command

dword 1M2MID; // 1M2M Serial number

byte SrcID; // Reboot reason source file ID incl. reboot reason

word LineNR; // Reboot reason line number

byte Version; // Firmware Version bit 0..3 Low 4..7 High

}TReboot;

When the unit is idle battery consumption is approximately 15uA

When moving with GPS On battery consumption is on average 30mA

Battery life heavily depends on the amount of time the unit is moving

**LoRa Connection behavior of the ED1608 is as follows:**

**ABP**

When ABP Parameters are set the unit first tries to get an ABP connection.

It tries to get a downlink message by sending out Acked messages.

As soon as a downlink message is received the connection is established.

When after a time out period of 20 minutes no Ack is received, or no ABP Parameters are set the unit switches to OTAA connection.

**OTAA**

It tries to join a network. Since joining starts at SF7 (short range) after two retries the units increases it spreading factor. This goes on until SF12 is reached, obeying the duty cycle limits this process of joining takes up to one hour, after that the units stops connecting and goes to sleep for 1 hour before trying to re-connect.

**Connection Lost**

If the unit is connected it actively guards its connection.

After 64 transmissions without any downlink answer the units starts requesting ack’s. If after 32 transmissions without an answer the unit assumes the connection is gone and increases its datarate. When the device has reached SF12 it assumes the connection is lost. The unit will go idle for 1 hour and tehn start to reconnect.

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| V1.08 | Baseline |  |  |
| V1.8 | Final version | 04-05-2016 |  |
| V1.9 | BugFix | 20-06-2016 | * Added UTMoving == 0 -> now means no updates while moving * BugFix Command to change message formats is now ignored, choosing classic message format lead to device not sending data anymore * Battery in Alive Message now 0..254 is 0..100% |
| V1.A | BugFix/  Feature | 13-07-2016 | * Removed bug when #DTMove and #DTIdle are set at value > 127 * Removed issue with brown-out when radio is switched on in latest HW Batch (Full Only) * Removed issue with acquiring keys failing now and then * Added the enable/disable monitor command (0x1F) |
| V1.B | BugFix/  Feature | 30-07-2016 | * Improved TX Performance with LoRa * Removed bug (introduced in v1.A) where after join failure device stops sending messages |