

ELTN-Workshop

Praxisteil: Dragino-Paket mit Sensoren, Nodes und Gateway „von 0 auf 100“

Einrichtung TTN-Webseite: Account, Gateway, Application, Node

- 1 Im Browser auf <https://www.thethingsnetwork.org> gehen
- 2 Account einrichten
 - 1 Signup
 - 2 Username, Mailadresse, Passwort eingeben
 - 3 Create Account
 - 4 Validierungsmail abrufen, in der Mail Activate Account anklicken
 - 5 → Webseite: Welcome back ...
- 3 Gateway einrichten
 - 1 CONSOLE anklicken
 - 2 register gateway anklicken, Formular REGISTER GATEWAY erscheint
 - 3 Seriennummer des Gateways ermitteln: Auf dem Typenschild steht als MAC-Adresse "wifi: A8:40:41:xx:xx:xx" – die letzten sechs Hex-Ziffern sind die Seriennummer xxxxxx.
 - 4 I'm using the legacy packet forwarder → aktivieren
 - 5 darüber erscheint jetzt Gateway EUI → 6 Bytes der WiFi-MAC-Adresse vom Typenschild + FF FF eingeben, also a84041xxxxxfff
 - 6 Description → Beschreibung eingeben
 - 7 Frequency Plan → Europe 868 MHz
 - 8 Router → bleibt auf ttn-router-eu
 - 9 Location → auf Karte auswählen
 - 10 Antenna Placement → indoor
 - 11 Register Gateway anklicken
 - 12 Gateway Overview erscheint
- 4 Application einrichten (falls nicht Zugriff auf eine vorhandene Application erteilt wurde)
 - 1 Applications anklicken
 - 2 add application
 - 3 Application ID → Identifier nach Wahl
 - 4 Description → Beschreibung
 - 5 Handler → ttn-handler-eu
 - 6 Add application anklicken
 - 7 Übersicht APPLICATIONS erscheint, neue Applikation ist mit aufgelistet
- 5 Nodes (Devices) einrichten
 - 1 gewünschte Applikation anklicken, Overview erscheint
 - 2 register device anklicken
 - 3 Device ID → Identifier nach Wahl, z.B. dragino-01-node0

- 4 Device EUI → Pfeil-Icon anklicken, ändert sich zu Stift-Icon + this field will be generated
- 5 Register anklicken
- 6 Device Overview erscheint
- 7 Settings anklicken
- 8 Activation Method → auf ABP ändern (LG01 unterstützt kein OTAA!)
- 9 Frame Counter Checks → deaktivieren
- 10 Save anklicken
- 11 Device Overview erscheint
- 12 Werte notieren / speichern für Sketch: Network Session Key, App Session Key, Device Address
 - 1 Augen-Icon anklicken, Key wird angezeigt
 - 2 "<>"-Icon anklicken, Key wird als C-Datenstruktur (msb) angezeigt
 - 3 Copy-Icon anklicken, Key wird in Pastebuffer kopiert
- 13 Zweites Devices ebenso einrichten, z.B. dragino-01-node1

Einrichtung LoRaWAN-Gateway Dragino LG01 für lokales WLAN

- 1 Anschließen:
 - 1 LoRa-Stummelantenne (Wichtig!!! LoRa-Geräte **niemals** ohne Antenne betreiben!)
 - 2 Netzteil
- 2 Mit Computer ins WLAN-Netzwerk dragino-xxxxxx verbinden, kein Passwort oder Passwort dragino-dragino.
- 3 Im Browser auf 10.130.1.1 oder dragino-xxxxxx.local gehen, Username root, Passwort dragino → Login
- 4 Statusübersicht erscheint
- 5 Menü Network
- 6 Abschnitt Internet Access: Access Internet Via → WiFi Client
- 7 Ausfüllen:
 - 1 SSID → Name des vorhandenen WLAN-Netzwerks (Freifunk)
 - 2 Encryption → z.B. WPA2 (None)
 - 3 Password → Passwort des WLAN-Netzwerks (leer lassen)
 - 4 Way to Get IP → DHCP
 - 5 Display Net Connection: nicht ändern
- 8 Save & Apply
- 9 "Applying changes"...
- 10 Mit Computer in das WLAN gehen, das soeben eingetragen wurde (Freifunk). Dragino-Router erscheint als Gerät „dragino-xxxxxx“.
- 11 Im Browser auf dragino-xxxxxx.local gehen, anmelden
- 12 Statusübersicht erscheint, IPv4 WAN Status zeigt Verbindung ins WLAN.
- 13 Firmwareversion prüfen: 4.3.4 erforderlich, ggf.
 - 1 <http://www.dragino.com/downloads/index.php?dir=motherboards/ms14/Firmware/loT/>

- 2 → IoT-build-v4.3.4-... herunterladen → .bin-Datei im Downloadordner
- 3 Menüoption System → Backup / Flash Firmware anklicken
- 4 Flash new firmware image:
 - 1 Keep settings → aktiv lassen
 - 2 Image: Choose File anklicken → .bin-Datei aus dem Downloadordner auswählen
 - 3 Flash image... anklicken (Datei wird hochgeladen, dauert eine Weile)
 - 4 Proceed anklicken (Update beginnt, dauert ebenfalls eine Weile)
 - 5 Während des Updates blinkt die 1. LED, dann folgt ein Reset, bei dem alle LEDs kurz aufleuchten
- 5 Im Browser neu verbinden + wieder anmelden
- 6 Statusübersicht erscheint, Firmwareversion entspricht jetzt der hochgeladenen Version.
- 14 Passwort ändern:
 - 1 Menüoption System → Administration
 - 2 Password + Confirmation: jeweils das neue Passwort angeben
 - 3 Save & Apply

Einrichtung TTN-Setup im Gateway

- 1 Im Browser <http://www.dragino.com/downloads/index.php?dir=motherboards/lq01/sketch/> aufrufen
- 2 Download der Datei single_pkt_fwd_v003.ino.hex
- 3 Im Browser Weboberfläche des Routers aufrufen
- 4 Menüoption Sensor → Flash MCU anklicken
- 5 Upload Image to MCU:
 - 1 MCUImage: Choose File anklicken → Datei aus Downloadordner auswählen
 - 2 Flash Image... anklicken (Programmierung beginnt, dauert eine Weile)
- 6 Upload Result: ... Sketch uploaded successfully.
- 7 Reboot des Gateways (z.B. System → Reboot)
- 8 Nach Reboot: Weboberfläche aufrufen, anmelden
- 9 Menüoption Sensor → MicroController
- 10 prüfen: MCU Version = Dateiname der hochgeladenen Datei
- 11 Menüoption Sensor → LoRa / LoRaWAN anklicken
- 12 LoRa Gateway Settings:
 - 1 Radio Settings:
 - 1 TX Frequency → 868100000 (= 868.1 MHz)
 - 2 RX Frequency → 868100000 (= 868.1 MHz)
 - 3 Rest bleibt
 - 2 LoRaWAN Server Settings:
 - 1 Server Address → router.eu.thethings.network
 - 2 Server Port → bleibt auf 1700
 - 3 Gateway ID → Gateway EUI eintragen wie zuvor auf TTN

- registriert (ohne Präfix EUI-)
- 3 Save & Apply
- 13 Menüoption Sensor → IoT Server anklicken
- 14 Select IoT Server:
 - 1 IoT Server → LoRaWAN
 - 2 Save & Apply

⇒ evtl. Problem? — Status → System log:

Sat Oct 13 12:40:39 2018 kern.notice syslog: [IoT]: Internet Connection Check: FAIL

Sat Oct 13 12:40:40 2018 kern.notice syslog: [IoT]: DNS Resolve Check: FAIL

Aber: auf TTN Console → Gateways → eui-a84041xxxxxxffff: Gateway Status "connected"

Einrichtung Arduino IDE

- 1 Download von <https://www.arduino.cc/en/Main/Software> – aktuell 1.8.7, rechts passende Hardwareplattform auswählen (Windows / macOS / Linux)
- 2 Installieren
- 3 Starten
- 4 (Windows:) Datei → Einstellungen, (macOS:) Cmd-Komma, (Linux:) ...
- 5 In Feld "Additional Boards Manager URLs" folgenden String eintragen:
- 6 http://www.dragino.com/downloads/downloads/YunShield/package_dragino_yun_test_index.json
- 7 (falls dort schon etwas steht, mit Komma anhängen)
- 8 Menüoption Tools → Boards → Boards Manager aufrufen
- 9 ins Suchfeld eingeben: Dragino
- 10 Eintrag "Dragino Yun by Dragino Technology" anklicken, Button Install erscheint
- 11 Install anklicken, Installationsbalken erscheint und verschwindet dann
- 12 Close anklicken

Einfachnode

- 1 Codebeispiel für Einfachnode: Simple_Node.ino
- 2 Sketch in Arduino IDE öffnen
- 3 Wert u1_t NWKSKY[16] ersetzen durch Network Session Key des ersten Devices (s.o.)
- 4 Wert u1_t APPSKY[16] ersetzen durch Network Session Key (s.o.)
- 5 Wert u4_t DEVADDR ersetzen durch Device Address (s.o.)
- 6 Ggf. Wert TX_INTERVAL auf kürzeres Intervall setzen (z.B. 10 statt

- 60 Sekunden)
- 7 Zeilen
- 8 `LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B), BAND_CENTI);`
- 9 mit Kanalnummern von 1 bis 8 auskommentieren, nur Kanal 0 bleibt stehen.
- 10 Darunter ergänzen:
- 11 `for(int i=1; i<=8; i++) LMIC_disableChannel(i);`
- 12 Zeile
- 13 `Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));`
- 14 ändern zu
- 15 `Serial.print(F("EV_TXCOMPLETE (includes waiting for RX windows); Frequency = "));`
- 16 `Serial.println(LMIC.freq);`
- 17 Node mit einfachem LoRa-Shield (ohne GPS) per USB-Kabel anschließen
- 18 Tools → Boards → Arduino/Genuino Uno anklicken
- 19 Tools → Port → passenden Port anklicken (macOS: /dev/cu.wchusbserial...)
- 20 Tools → Serial Monitor öffnen, 115200 Bd einstellen
- 21 Sketch → Upload anklicken
- 22 Node sendet Pakete
- 23 Kontrolle 1: Serial Monitor: Ausgabe: Packet queued ... EV_TXCOMPLETE (includes waiting for RX windows)
- 24 Kontrolle 2: TTN Console → Gateways → Gateway ... → Traffic zeigt Pakete

Problem: Nicht alle Pakete werden empfangen (vgl. Sequenznummern)

Sensornode

- 1 DHT11-Bibliothek installieren:
 - 1 ZIPfile: DHTlib.zip
 - 2 ZIPfile auspacken
 - 3 Verzeichnis DHTlib verschieben nach (macOS:)
 - ~/Documents/Arduino/libraries
 - 4 IDE neu starten
- 2 Codebeispiel für Sensornode: Temp_Node.ino
- 3 Sketch in Arduino IDE öffnen
- 4 Die beiden Schlüssel und die Deviceadresse des **ersten** Devices eintragen
- 5 Blauen Wettersensor DHT11 an den Node anschließen:
 - 1 Arduino GND – schwarzes Kabel – Sensor GND
 - 2 Arduino A1 – weißes Kabel – Sensor DATA
 - 3 Arduino 3V3 – oranges Kabel – Sensor VCC (Achtung, nicht an 5V anschließen!)
- 6 Node an USB anschließen

- 7 Sketch → Upload anklicken
- 8 Node sendet Pakete
- 9 Kontrolle 1: Serial Monitor: Ausgabe: Packet queued ...
EV_TXCOMPLETE (includes waiting for RX windows)
- 10 Kontrolle 2: TTN Console → Gateways → Gateway ... → Traffic zeigt Pakete
- 11 Payload Decoder einrichten: Auf der TTN Console zur Application gehen, Payload Formats anklicken
- 12 Payload Format → Zustrom (sollte voreingestellt sein)
- 13 Decoder anklicken
- 14 JavaScript-Code einfügen
- 15 save payload functions anklicken

Trackingnode

- 1 TinyGPS-Bibliothek in Arduino IDE installieren:
 - 1 ZIPfile: TinyGPS13.zip
 - 2 ZIPfile auspacken
 - 3 Verzeichnis TinyGPS-13 verschieben nach (macOS:)
~/Documents/Arduino/libraries
 - 4 IDE neu starten
- 2 Codebeispiel für Trackingnode: TempTrack_Node.ino
- 3 Sketch in Arduino IDE öffnen
- 4 Die beiden Schlüssel und die Deviceadresse des **zweiten** Devices eintragen
- 5 Blauen Wettersensor DHT11 an den Node anschließen:
 - 1 Arduino GND – schwarzes Kabel – Sensor GND
 - 2 Arduino A1 – weißes Kabel – Sensor DATA
 - 3 Arduino 3V3 – oranges Kabel – Sensor VCC (Achtung, nicht an 5V anschließen!)
- 6 Die beiden GPS-Jumper GPS_RX und GPS_TX umsetzen von rechts (.##) nach links (##.)
- 7 GPS-Datenkabel verbinden:
 - 1 Arduino-GPS-Shield GPS_RX (1.Pin) – grünes Kabel – Arduino A2
 - 2 Arduino-GPS-Shield GPS_TX (1.Pin) – gelbes Kabel – Arduino A3
- 8 Node an USB anschließen
- 9 Sketch → Upload anklicken
- 10 Node sendet Pakete; wenn die grüne LED im Sekundentakt blinkt, besteht GPS-Empfang
- 11 Kontrolle 1: Serial Monitor: Ausgabe: Packet queued ...
EV_TXCOMPLETE (includes waiting for RX windows)
- 12 Kontrolle 2: TTN Console → Gateways → Gateway ... → Traffic zeigt Pakete, Payload Decoder liefert Koordinaten

TTN Mapper einrichten

- 1 Console: Applikation aufrufen, Integrations

- 2 Add Integration → TTN Mapper
 - 1 Process ID → z.B. maptest
 - 2 E-mail address → (Mailadresse eingeben)
 - 3 Port filter → z.B. 3 für den Beispielcode
 - 4 Experiment name → (leer lassen)
 - 5 Add integration anklicken
- 3 Im Browser https://ttnmapper.org/special_maps.php aufrufen
- 4 Device ID des Tracking-Nodes eintragen
- 5 View map anklicken
- 6 Karte mit Mapping-Daten wird aufgelegt

Tracking-Sensornode

- 1 DHT11-Sensor vom anderen Node umbauen auf den GPS-Node (gleiche Anschlüsse)
- 2 Code überprüfen und hochladen
- 3 Kontrolle 1: Serial Monitor: Ausgaben mit Luftdaten und Koordinaten
- 4 Kontrolle 2: TTN Console → Gateways → Gateway ... → Traffic zeigt Pakete, Payload Decoder liefert Luftdaten und Koordinaten

Datenauswertung

- 1 Voraussetzung: Server mit Node-RED ist eingerichtet
- 2 TTN Console → Overview → ACCESS KEYS → default key: Kopier-Icon anklicken
- 3 Kommandozeile:
 - 1 Aufruf
 - 2 `mosquitto_sub -v -h eu.thethings.network -t "eltn-test/devices/+/up" -u eltn-test -P (Access Key)`
 - 3 Kontrolle: Aufgabe von Topic sowie Paketdaten (als JSON)
- 4 Node-RED
 - 1 Node-RED-Oberfläche im Browser aufrufen
 - 2 ggf. neuen Flow-Tab anlegen
 - 3 Node input → mqtt auf Flow ziehen, doppelklicken
 - 4 Server → Add new mqtt broker... → Stift-Icon
 - 1 Name → TTN - ELTN
 - 2 Connection
 - 1 Server → eu.thethings.network
 - 3 Security
 - 1 Username → eltn-test
 - 2 Password → default access key wie oben kopiert
 - 4 Add anklicken
 - 5 Topic → # oder eltn-test/devices/+/up
 - 6 Done anklicken
 - 7 Test (Debugausgabe)
 - 1 Node Output → debug auf Flow ziehen
 - 2 mqtt-Node mit debug-Node verbinden

- 3 Deploy anklicken
- 4 Debug anklicken, auf nächste Message warten
- 5 Kontrolle: JSON-Datensatz und alle einzelnen Felder (bei Topic #) werden ausgegeben

8 Kartendarstellung:

- 1 Node function → json
 - 1 Action → Always convert to JavaScript Object
 - 2 Property → msg.payload
- 2 Node function → change, Regeln:
 - 1 Set msg.dev_id to JSONata payload.dev_id
 - 2 Set msg.payload to JSONata payload.payload_fields
 - 3 Set msg.payload.lat to JSONata payload.latitude
 - 4 Set msg.payload.lon to JSONata payload.longitude
 - 5 Set msg.payload.name to JSONata dev_id
- 3 Node location → world map
- 4 Nodes verbinden: mqtt → json → change → world map
- 5 Deploy
- 6 Karte mit ctrl-shift-M aufrufen