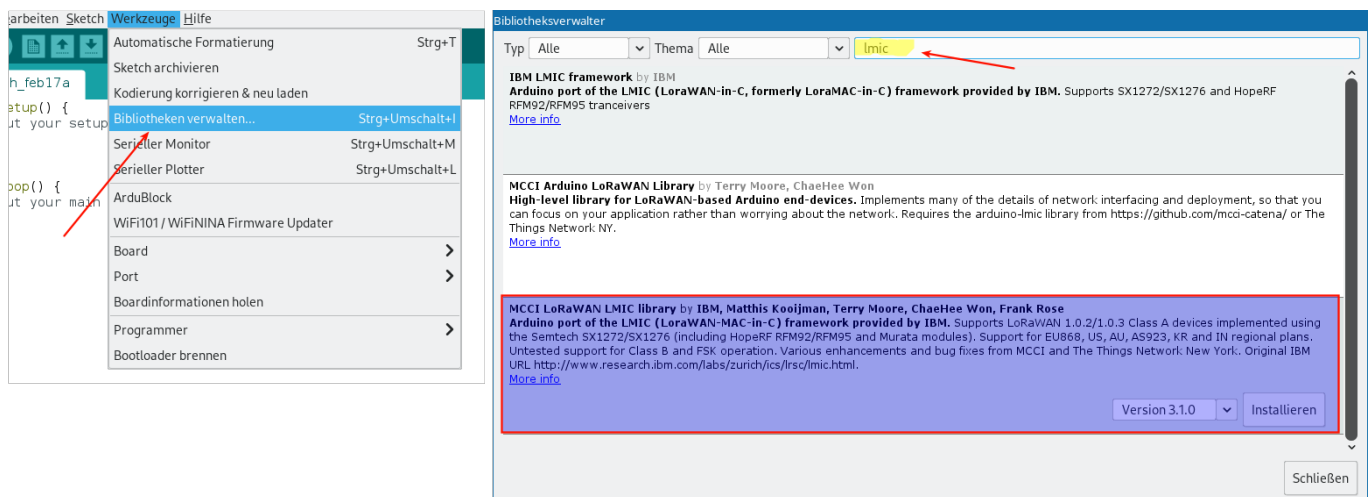


Ein "Standardsensor" mit dem LoRaShield

LoraShield aufstecken

Bibliothek installieren

Um das Shield ansprechen zu können, muss die LMIC Bibliothek installiert werden:



Sketch

hellolora.ino

```
// MIT License
//
// https://github.com/gonzalocasas/arduino-uno-dragino-lorawan/blob/master
// LICENSE
// Based on examples from
// https://github.com/matthijskooijman/arduino-lmic
// Copyright (c) 2015 Thomas Telkamp and Matthijs Kooijman

// Adaptions: Andreas Spiess

#include <lmic.h>
#include <hal/hal.h>
// #include <credentials.h>

#ifdef CREDENTIALS
static const u1_t NWKSKEY[16] = NWKSKEY1;
static const u1_t APPSKEY[16] = APPSKEY1;
static const u4_t DEVADDR = DEVADDR1;
#else
```

```
static const u1_t NWKSKY[16] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00};
static const u1_t APPSKY[16] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00};
static const u4_t DEVADDR = 0x00000000;
#endif

// These callbacks are only used in over-the-air activation, so they
// are
// left empty here (we cannot leave them out completely unless
// DISABLE_JOIN is set in config.h, otherwise the linker will
// complain).
void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 20;

// Pin mapping Dragino Shield
const lmick_pinmap lmick_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 6, 7},
};
void onEvent (ev_t ev) {
    if (ev == EV_TXCOMPLETE) {
        Serial.println(F("EV_TXCOMPLETE (includes waiting for RX
windows)"));
        // Schedule next transmission
        os_setTimedCallback(&sendjob,
os_getTime()+sec2osticks(TX_INTERVAL), do_send);
    }
}

void do_send(osjob_t* j){
    // Payload to send (uplink)
    static uint8_t message[] = "hi";

    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
```

```
        // Prepare upstream data transmission at the next possible
time.
        LMIC_setTxData2(1, message, sizeof(message)-1, 0);
        Serial.println(F("Sending uplink packet..."));
    }
    // Next TX is scheduled after TX_COMPLETE event.
}

void setup() {
    Serial.begin(115200);
    Serial.println(F("Starting..."));

    // LMIC init
    os_init();

    // Reset the MAC state. Session and pending data transfers will be
discarded.
    LMIC_reset();

    // Set static session parameters.
    LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);

    // Disable link check validation
    LMIC_setLinkCheckMode(0);

    // TTN uses SF9 for its RX2 window.
    LMIC.dn2Dr = DR_SF9;

    // Set data rate and transmit power for uplink (note: txpow seems
to be ignored by the library)
    LMIC_setDrTxpwr(DR_SF12,14);

    // Start job
    do_send(// MIT License
//
https://github.com/gonzalocasas/arduino-uno-dragino-lorawan/blob/master
/LICENSE
// Based on examples from
https://github.com/matthijskooijman/arduino-lmic
// Copyright (c) 2015 Thomas Telkamp and Matthijs Kooijman

// Adaptions: Andreas Spiess

#include <lmic.h>
#include <hal/hal.h>
//#include <credentials.h>

#ifdef CREDENTIALS
static const u1_t NWKSKEY[16] = NWKSKEY1;
static const u1_t APPSKEY[16] = APPSKEY1;
static const u4_t DEVADDR = DEVADDR1;
```

```
#else
static const u1_t NWKSKY[16] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
static const u1_t APPSKY[16] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
static const u4_t DEVADDR = 0x00000000;
#endif

// These callbacks are only used in over-the-air activation, so they
// are
// left empty here (we cannot leave them out completely unless
// DISABLE_JOIN is set in config.h, otherwise the linker will
// complain).
void os_getArtEui (u1_t* buf) { }
void os_getDevEui (u1_t* buf) { }
void os_getDevKey (u1_t* buf) { }

static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 20;

// Pin mapping Dragino Shield
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 6, 7},
};
void onEvent (ev_t ev) {
    if (ev == EV_TXCOMPLETE) {
        Serial.println(F("EV_TXCOMPLETE (includes waiting for RX
windows)"));
        // Schedule next transmission
        os_setTimedCallback(&sendjob,
os_getTime()+sec2osticks(TX_INTERVAL), do_send);
    }
}

void do_send(osjob_t* j){
    // Payload to send (uplink)
    static uint8_t message[] = "hi";

    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    }
}
```

```
    } else {  
        // Prepare upstream data transmission at the next possible  
time.  
        LMIC_setTxData2(1, message, sizeof(message)-1, 0);  
        Serial.println(F("Sending uplink packet..."));  
    }  
    // Next TX is scheduled after TX_COMPLETE event.  
}  
  
void setup() {  
    Serial.begin(115200);  
    Serial.println(F("Starting..."));  
  
    // LMIC init  
    os_init();  
  
    // Reset the MAC state. Session and pending data transfers will be  
discarded.  
    LMIC_reset();  
  
    // Set static session parameters.  
    LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);  
  
    // Disable link check validation  
    LMIC_setLinkCheckMode(0);  
  
    // TTN uses SF9 for its RX2 window.  
    LMIC.dn2Dr = DR_SF9;  
  
    // Set data rate and transmit power for uplink (note: txpow seems  
to be ignored by the library)  
    LMIC_setDrTxpow(DR_SF12,14);  
  
    // Start job  
    do_send(&sendjob);  
}  
  
void loop() {  
    os_runloop_once();  
}  
&sendjob);  
}  
  
void loop() {  
    os_runloop_once();  
}
```

Last
update:
17.02.2020 19:28 faecher:nwt:lorawan:uebersicht:lorashield:start <https://wiki.qg-moessingen.de/faecher:nwt:lorawan:uebersicht:lorashield:start?rev=1581964113>

From:
<https://wiki.qg-moessingen.de/> - **QG Wiki**

Permanent link:
<https://wiki.qg-moessingen.de/faecher:nwt:lorawan:uebersicht:lorashield:start?rev=1581964113>

Last update: **17.02.2020 19:28**

