**Slå av melding**

UNLOG COM1 GPGGA

Slå på melding

gpgga COM1 1

**Sette i RTK base modus**

MODE BASE TIME 60

→ Mål i 60 sekunder, og bruk snittet av posisjonene som baseposisjon.

Du kan endre 60 til opp til 3600 sekunder for høyere nøyaktighet.

(Se også MODE BASE TIME 60 5 for å legge til et posisjonstoleransekrav.)

**ESP32-S3-Devkit1-C sender**

**A hand holding a wire

AI-generated content may be incorrect.**

**ESP32-S3-Devkit1-C mottager**

**A hand holding a piece of electrical equipment

AI-generated content may be incorrect.**

#include <RFM69.h>

#include <SPI.h>

#define NODE\_ID        2        // This is the Gateway ID (receiver)

#define NETWORK\_ID     100

#define FREQUENCY      RF69\_868MHZ

#define RFM69\_CS       10

#define RFM69\_IRQ      9

#define RFM69\_RST      -1

RFM69 radio(RFM69\_CS, RFM69\_IRQ, true);

void

setup() {

  Serial.begin(115200);

  delay(10);

  SPI.begin(12, 13, 11, 10); // Explicit SPI pins, matching sender

  if (RFM69\_RST != -1) {

    pinMode(RFM69\_RST, OUTPUT);

    digitalWrite(RFM69\_RST, LOW);

    delay(10);

    digitalWrite(RFM69\_RST, HIGH);

    delay(10);

  }

  Serial.println("Starting RFM69 receiver...");

  if (!radio.initialize(FREQUENCY, NODE\_ID, NETWORK\_ID)) {

    Serial.println("Radio initialization failed!");

    while (1);

  }

  radio.setHighPower();

  radio.encrypt(NULL); // No encryption, matching sender

  Serial.println("RFM69 initialized");

}

void loop() {

  if (radio.receiveDone()) { // Check if data is received

    if (radio.DATALEN > 0) {

      Serial.print("Received from Node ");

      Serial.print(radio.SENDERID);

      Serial.print(": [");

      for (byte i = 0; i < radio.DATALEN; i++) {

        Serial.print((char)radio.DATA[i]);

      }

      Serial.print("], RSSI: ");

      Serial.println(radio.RSSI);

      // Acknowledge receipt (optional, but useful for reliability)

      if (radio.ACKRequested()) {

        radio.sendACK();

        Serial.println("ACK sent");

      }

      // Blink LED to indicate reception

      digitalWrite(RGB\_BUILTIN, HIGH);

      delay(200);

      digitalWrite(RGB\_BUILTIN, LOW);

    }

  }

}