

# KOCAELİ ÜNİVERSİTESİ

## ELEKTRONİK VE HABERLEŞME

### MÜHENDİSLİĞİ

**Proje Adı:**LoRa Tabanlı Sensör Verisi İletim Sistemi

**Ders Adı:**Mühendislik Tasarım 1

**Öğrenci Adı:**Muhammet Emin GÜNEL

**Numara:**220208004

**Öğretim Üyesi:**Anıl ÇELEBİ

**Tarih:**05/01/20026

## **İçindekiler**

1. Giriş
2. Projenin amacı
3. Sistem mimarisi
  - 3.1. Donanım bileşenleri
  - 3.2. Yazılım mimarisi
    - 3.2.1. Verici’de kullanılan yazılımı
    - 3.2.2. Alıcı’da kullanılan yazılımı
  - 3.3. Sistem blok diyagramı
4. Test prosedürleri
  - 4.1. Donanım test prosedürleri
  - 4.2. Yazılım test prosedürleri
5. Haberleşme yapısı ve parametreler
6. Maliyet Analizi
7. Sonuç
8. Kaynakça

## **1. Giriş**

Bu projede, Raspberry Pi Pico 2 W ve Ai-Thinker RA-01 LoRa modülleri kullanılarak iki nokta arasında kablosuz veri aktarımı gerçekleştirilmesi hedeflenmektedir. Verici tarafta BME280 sıcaklık–nem–basınç sensöründen alınan veriler LoRa P2P protokolü ile alıcı node’a aktarılmakta, alıcı bu veriyi terminalde göstermektedir.

## **2. Projenin Amacı**

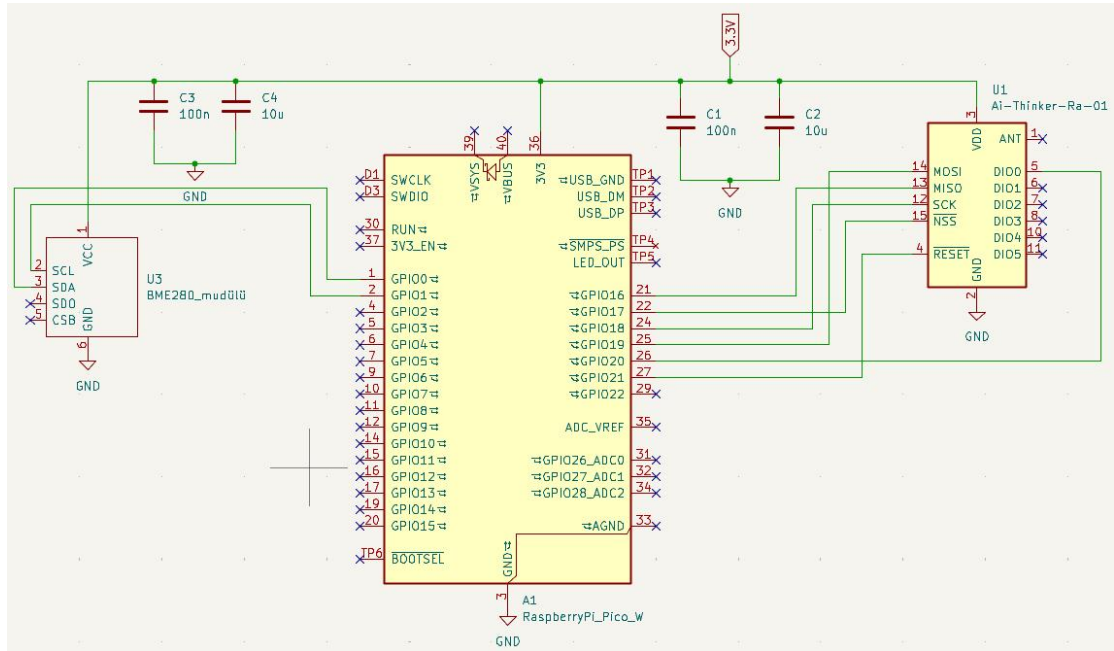
- Düşük güç tüketimli, uzun menzilli kablosuz haberleşme sistemi tasarlamak.
- LoRa P2P (Peer-to-Peer) haberleşmesini uygulamak.
- Sensör verisini güvenli şekilde iki cihaz arasında iletmek.

## **3. Sistem Mimarisi**

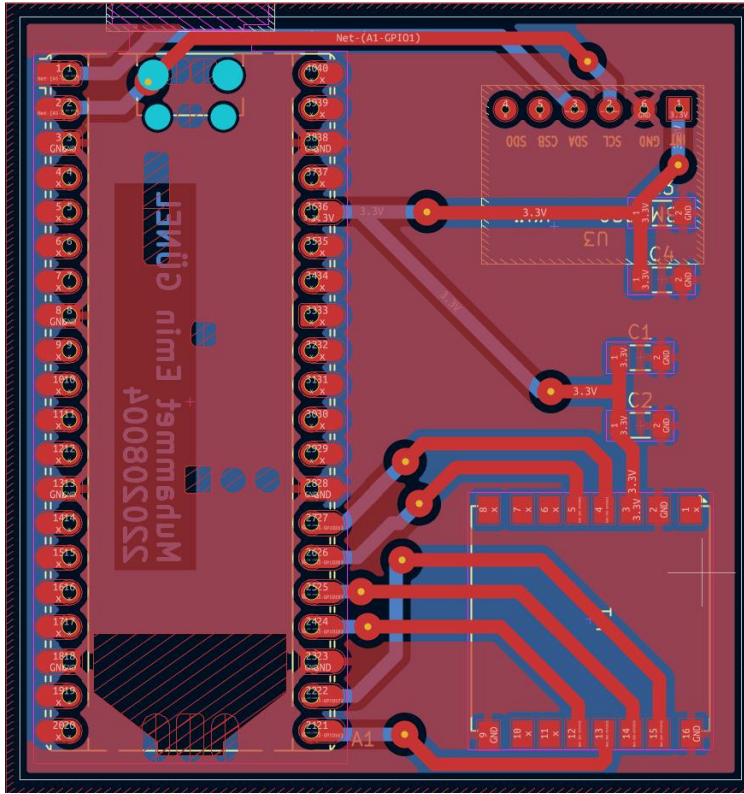
### 3.1. Donanım Bileşenleri

- Raspberry Pi Pico 2W
- BME280 Sensörü
- AI Thinker RA-01(SX1278)
- Anten
- Güç Beslemesi

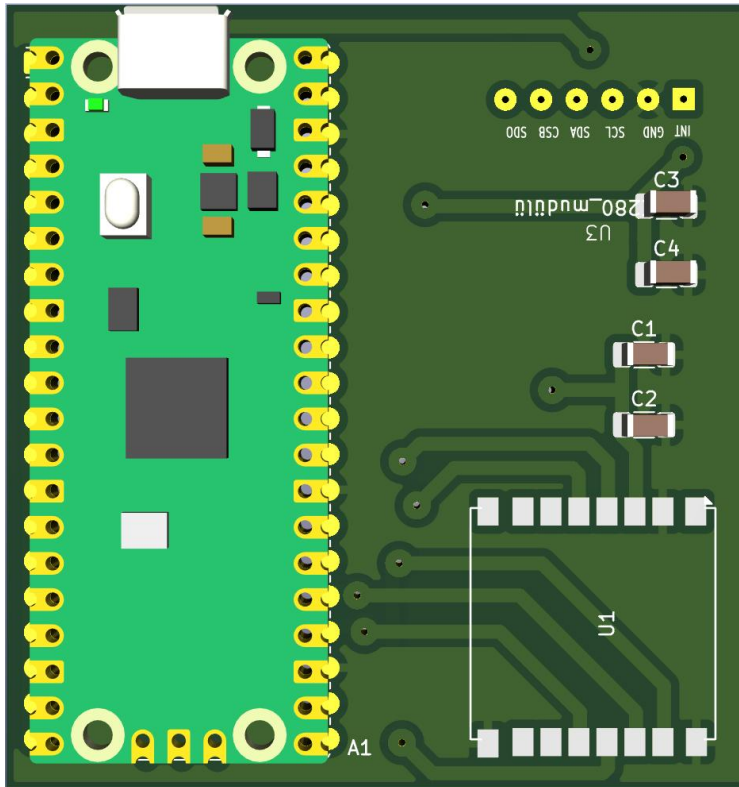
Sistem, bir verici (TX) ve bir alıcı (RX) düğümden oluşmaktadır. Verici düğümde BME280 sensörü kullanılarak çevresel veriler ölçülmüş, bu veriler mikrodenetleyici tarafından işlenerek LoRa modülü üzerinden kablosuz olarak iletilmiştir.



Şekil 3.1. Verici düğüm şematik



Şekil 3.2 Verici PCB



Şekil 3.3 Verici 3D görsel

### 3.2.Yazılım Mimarisi

- Sensör okuma
- Veri paketleme
- LoRa gönderim
- RX tarafında alma ve gösterme

#### 3.2.1.Verici’de kullanılan yazılımı

```
#include <SPI.h>
#include <Wire.h>
#include <RadioLib.h>
#include <Adafruit_BME280.h>

// ----- LoRa PINLER -----
#define LORA_CS 17
#define LORA_DIO0 20
#define LORA_RST 21

#define SPI_SCK 18
#define SPI_MISO 16
#define SPI_MOSI 19

// ----- BME280 PINLER -----
#define SDA_PIN 0
#define SCL_PIN 1

SX1278* lora;
Adafruit_BME280 bme;

int packetCounter = 0;

void setup() {
  Serial.begin(115200);
  while (!Serial) { delay(10); }
  Serial.println("BME280 + LoRa TX basladi");

  // ---- I2C (BME280) ----
  Wire.setSDA(SDA_PIN);
  Wire.setSCL(SCL_PIN);
  Wire.begin();

  if (!bme.begin(0x76)) {
    Serial.println("BME280 bulunamadi!");
    while (1);
  }
}
```

```

}
Serial.println("BME280 OK");

// ---- SPI (LoRa) ----
SPI.setSCK(SPI_SCK);
SPI.setRX(SPI_MISO);
SPI.setTX(SPI_MOSI);
SPI.begin();

lora = new SX1278(new Module(LORA_CS, LORA_DIO0,
LORA_RST));

if (lora->begin(433.0) != RADIOLIB_ERR_NONE) {
  Serial.println("LoRa init hatasi!");
  while (1);
}

lora->setSpreadingFactor(7);
lora->setBandwidth(125.0);
lora->setCodingRate(5);
lora->setOutputPower(17);

Serial.println("LoRa TX hazır");
}

void loop() {
  // ---- SENSOR OKUMA ----
  float temperature = bme.readTemperature();
  float humidity = bme.readHumidity();
  float pressure = bme.readPressure() / 100.0;

  // ---- VERİ PAKETLEME ----
  String packet =
    "ID:" + String(packetCounter) +
    ";T:" + String(temperature, 1) +
    ";H:" + String(humidity, 1) +
    ";P:" + String(pressure, 1);

  // ---- LoRa GONDER ----
  int state = lora->transmit(packet);

  if (state == RADIOLIB_ERR_NONE) {
    Serial.print("GONDERİLDİ: ");
  }
}

```

```

    Serial.println(packet);
  } else {
    Serial.println("TX HATASI");
  }

  packetCounter++;
  delay(2000);
}

```

### 3.2.2. Alıcı'da kullanılan yazılım

```

#include <SPI.h>
#include <RadioLib.h>

// ----- LoRa PINLER -----
#define LORA_CS  17
#define LORA_DIO0 20
#define LORA_RST 21

// ----- SPI PINLER -----
#define SPI_SCK  18
#define SPI_MISO 16
#define SPI_MOSI 19

SX1278* lora;

void setup() {
  Serial.begin(115200);
  while (!Serial) { delay(10); }
  Serial.println("=== RX KART ===");

  // ---- SPI baslat ----
  SPI.setSCK(SPI_SCK);
  SPI.setRX(SPI_MISO);
  SPI.setTX(SPI_MOSI);
  SPI.begin();
  Serial.println("SPI OK");

  // ---- SX1278 ----
  lora = new SX1278(new Module(LORA_CS, LORA_DIO0,
LORA_RST));

  int state = lora->begin(433.0);
  if (state != RADIOLIB_ERR_NONE) {

```

```

    Serial.print("LoRa init HATA, kod = ");
    Serial.println(state);
    while (1);
}

// ---- TX ile AYNI parametreler ----
lora->setSpreadingFactor(7);
lora->setBandwidth(125.0);
lora->setCodingRate(5);

Serial.println("RX modunda, paket bekleniyor...");
}

void loop() {
    String incoming;

    int state = lora->receive(incoming);

    if (state == RADIOLIB_ERR_NONE) {
        Serial.print("ALINDI: ");
        Serial.println(incoming);

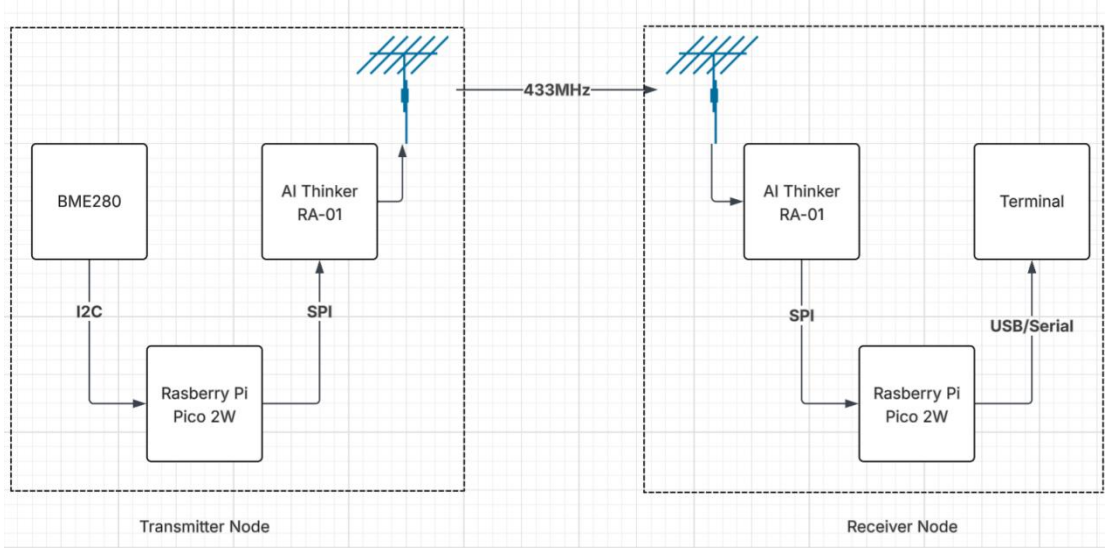
        Serial.print("RSSI: ");
        Serial.print(lora->getRSSI());
        Serial.print(" dBm | SNR: ");
        Serial.print(lora->getSNR());
        Serial.println(" dB");
        Serial.println("-----");
    }
}

```

Yazılım yapısı modüler olarak tasarlanmıştır. Sensör okuma, veri paketleme ve LoRa haberleşme işlemleri ayrı fonksiyonlar halinde gerçekleştirilmiştir.



### 3.3.Sistem Blok Diyagramı



Şekil 3.4.Blok diyagram

## 4. Test Prosedürleri

### 4.1. Donanım Test Prosedürleri

Test kodu	Açıklama	Beklenen	Pass/Fail
Hw1	PCB’de basılan yolların kısa devre testi	Yolun iki ucu arasında multimetrenin kısa devre modunda iken tutulduğunda bip sesi gelmesi.	Pass
Hw2	GND ve VCC arasında kısa devre testi	GND ve VCC arasında multimetre kısa devre modunda iken tutulduğunda bip sesi gelmemesi..	Pass
Hw3	GND ve yollar arasında kısa devre testi	VCC ve yollar arasında multimetre kısa devre modunda ilken gezdirildiğinde bip sesi gelmem	Pass
Hw4	GND ve yollar arasında kısa devre testi	GND ve yollar arasında multimetre kısa devre modunda iken tutulduğunda bip sesi gelmemesi.	Pass

## 4.2. Yazılım Test Prosedürleri

Test kodu	Açıklama	Beklenen	Pass/Fail
L1	Sanity (BME280 + LoRa Init)	Sensör ve RA-01 çalışması doğrulama	Pass
L2	PST Testi (100 paket)	Gönderilen ve alınan paket sayısının birbirine eşit olması.	Pass
L3	Sensör Testi (30 paket)	Sıcaklık, nem ve basınç değerlerinin alıcı tarafta eksiksiz görülmesi beklenir.	Pass
L4	Soak (20 dk)	RSSI ve SNR değerleri ilk ve son paket arasında farklılıkların büyük olmaması beklenir.	Pass

## 5. Haberleşme yapısı ve parametreler

Frekans	433MHz
Spedind factor	SF7
Bandwidth	125kHz
Coding rate	4/5
TX güç	17dBm

LoRa haberleşmesi P2P modunda gerçekleştirilmiş olup, sistem parametreleri kısa mesafede düşük gecikme ve kararlı iletişim sağlayacak şekilde ayarlanmıştır.

## 6. Maliyet analizi

2x Raspberry pi pico 2w	383.53 TL
2x AI Thinker RA-01	438.60 TL
4x Kapasitör	10 TL
1x BME280 sensör	236.50 TL

## 7. Sonuç

Bu çalışmada, LoRa tabanlı bir sensör veri iletim sistemi başarıyla gerçekleştirilmiştir. Yapılan testler sonucunda sistemin kısa mesafede kararlı ve güvenilir şekilde çalıştığı gösterilmiştir.

## 8. Kaynakça

<https://pip.raspberrypi.com/documents/RP-008304-DS-1-pico-2-w-datasheet.pdf>

<http://pdf.direnc.net/upload/bme280-i2c-basinc-sicaklik-ve-nem-sensoru-datasheet.pdf>

<https://pdf.direnc.net/upload/ra-01-antenli-lora-modulu-datasheet.pdf>