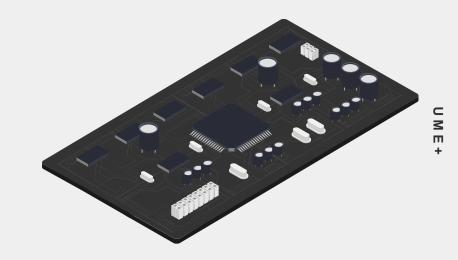




ULTRA HIMEAS

Ultrasonic Height Measurement Device with Wemos D1 Mini, Firebase, and Kodular.

Project | 3ISA2 [Group 3]



TEAM



Leader



Riki Awal Syahputra (2120010136)

Developer



Faza Rama Nugraha (2120010291)

SYSTEM ANALYSIS

2 SCHEMATIC

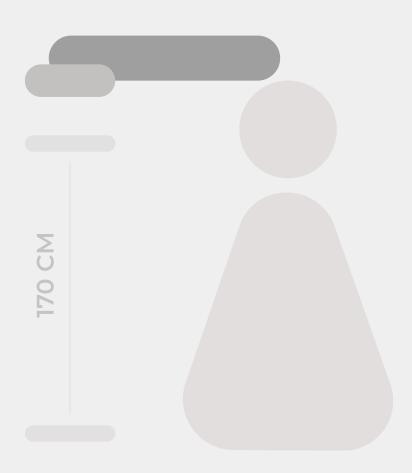
3 CODE

4 HARDWARE

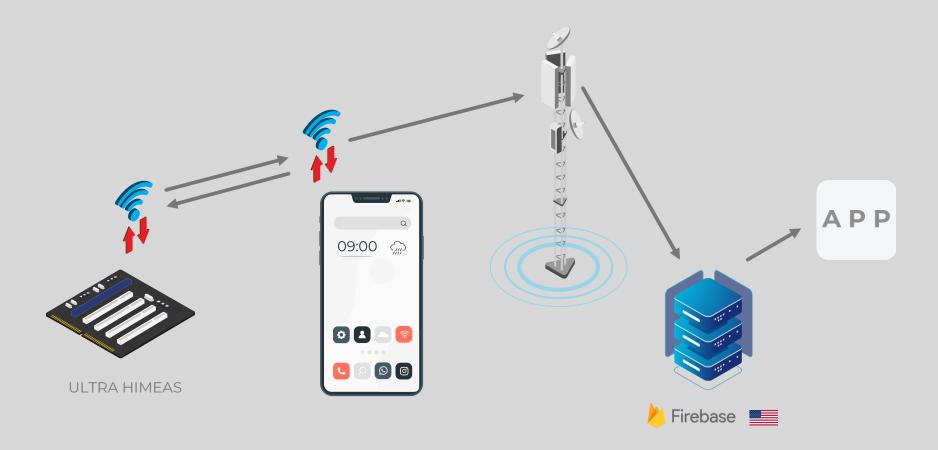
5 APP

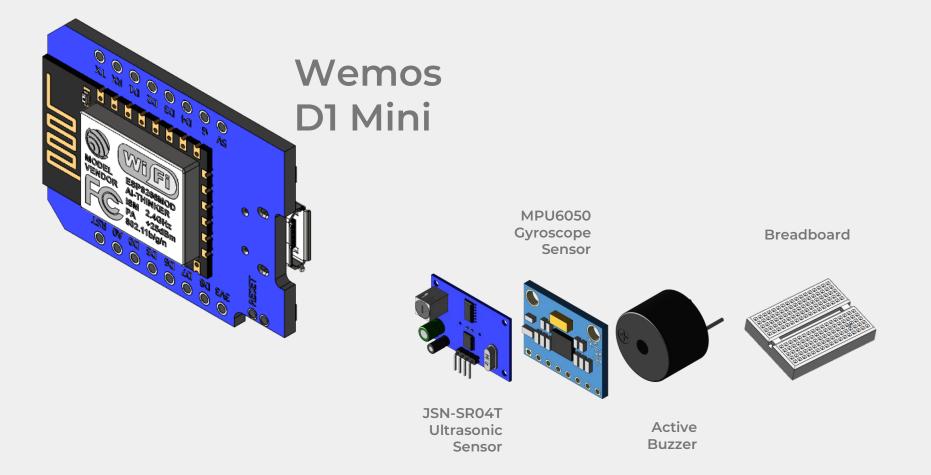
SYSTEM ANALYSIS

"Ultra Himeas is a portable device that can measure people's height using a single ultrasonic probe, in addition, this device uses a gyroscope sensor to make sure the accuracy of measurement, designed to plug and play system, plug into the adapter, connect the wifi, open the app, and place the device on top of user's head and point the ultrasonic sensor to the ground, then the data can be view on the app."



Flow Process



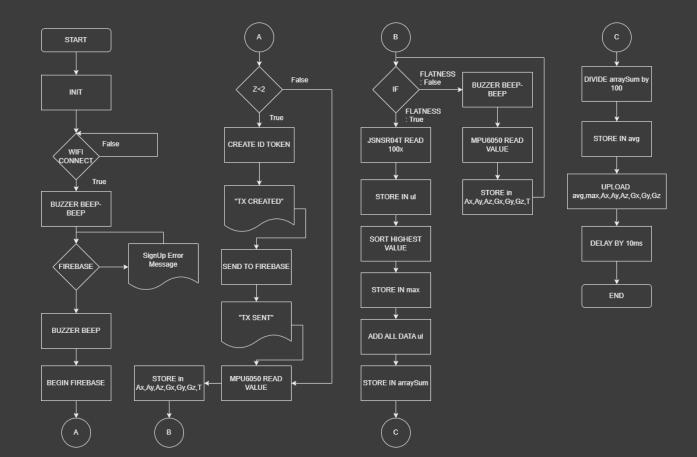


MAIN COMPONENT

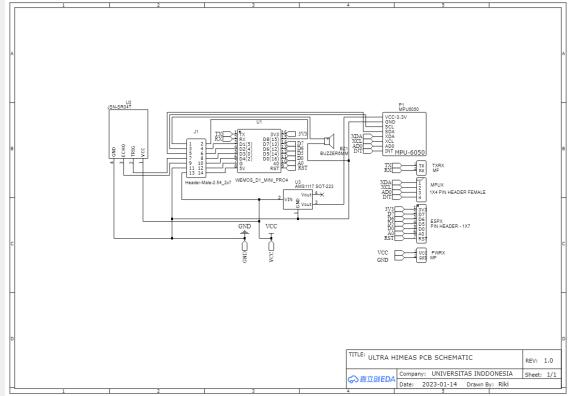


SCHEMATIC

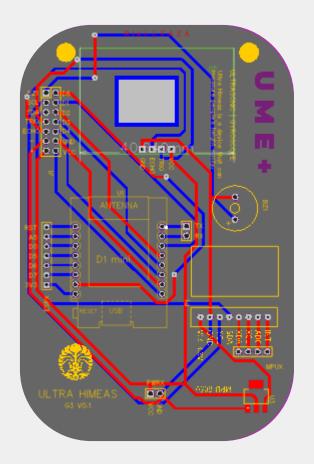
Flowchart

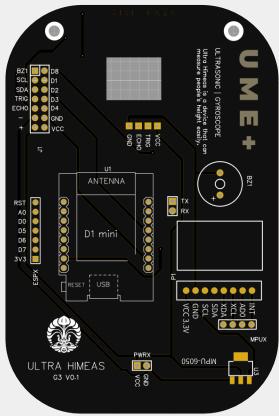


Schematic Circuit



Schematic PCB







CODE

CODE

ALGORITHM (AVG MODE)

HARDWARE

Identity Integration
MPU6050 ISNSP

RANDOM . WIFI . FIREBASE . BUZZER

SR04T ASE . Processing
. BUBLE SORT ALGORITHM
(HST MODE) . AVG

SOFTWARE (Kodular)

SCREEN INIT . CLOCK . FIREBASE . READ . UPDATE

```
.
```

```
#include <Arduino.h>
#if defined(ESP32)
#include <WiFi.h>
#elif defined(ESP8266)
#include <ESP8266WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"
#include <HCSR04.h>
#include <Adafruit MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
#define DATABASE_URL "https://ultra-himeas-4a302-default-rtdb.firebaseio.com"
FirebaseData fbdo;
FirebaseConfig config;
unsigned long sendDataPrevMillis = 0;
int count = 0;
bool signupOK = false;
const uint8_t MPU6050SlaveAddress = 0x68;
const uint8_t scl = D2;
 const uint8_t sda = D1;
```

```
const uint16_t AccelScaleFactor = 16384;
const uint16_t GyroScaleFactor = 131;
const uint8_t MPU6050_REGISTER_SMPLRT_DIV = 0x19;
const uint8_t MPU6050_REGISTER_USER_CTRL = 0x6A;
const uint8_t MPU6050_REGISTER_PWR_MGMT_1 = 0x68;
const uint8_t MPU6050_REGISTER_PWR_MGMT_2 = 0x6C;
const uint8_t MPU6050_REGISTER_CONFIG
const uint8_t MPU6050_REGISTER_GYRO_CONFIG = 0x18;
const uint8_t MPU6050_REGISTER_ACCEL_CONFIG = 0x1C;
const uint8_t MPU6050_REGISTER_FIF0_EN = 0x23;
const uint8_t MPU6050_REGISTER_INT_ENABLE = 0x38;
const uint8_t MPU6050_REGISTER_ACCEL_XOUT_H = 0x3B;
const uint8_t MPU6050_REGISTER_SIGNAL_PATH_RESET = 0x68;
const char* ssid = "IPHONE 20 PRO";
const char* password = "mokorutoaitokotoz";
float ul[100];
String randomString = " ";
void setup() {
  pinMode(D8, OUTPUT);
  MPU6050_Init();
  digitalWrite(D8,HIGH);
  randomSeed(analogRead(0));
```

```
Serial.print("Connecting to ");
WiFi.hostname("HIMEAS AX01");
digitalWrite(D8,LOW);
digitalWrite(D8,HIGH);
while (WiFi.status() != WL_CONNECTED) {
Serial.println("WiFi connected");
digitalWrite(D8,HIGH);
digitalWrite(D8,LOW);
Serial.print("IP address: ");
```

config.database_url = DATABASE_URL;

```
if (Firebase.signUp(&config, &auth, "", "")){
   Serial.println("ok");
   digitalWrite(D8,HIGH);
   delay(10);
digitalWrite(D8,LOW);
 else{
 Firebase.begin(&config, &auth);
void loop() {
   char randomChar = (char)(random(48, 123));
   if ((randomChar >= 58 && randomChar <= 64) || (randomChar >= 91 && randomChar <= 96)) {
    randomChar = (char)(random(48, 58) + random(65, 91) + random(97, 123));
   randomString += randomChar;
 Serial.println("TX CREATED");
 Serial.println("TX HAS BEEN SEND");
 else{}
 double Ax, Ay, Az, T, Gx, Gy, Gz;
```

```
Read_RawValue(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_XOUT_H);
Ax = (double)AccelX/AccelScaleFactor;
Ay = (double)AccelY/AccelScaleFactor;
Az = (double)AccelZ/AccelScaleFactor;
Gx = (double)GyroX/GyroScaleFactor;
Gy = (double)GyroY/GyroScaleFactor;
Gz = (double)GyroZ/GyroScaleFactor;
Serial.print("Ax: "); Serial.print(Ax);
Serial.print(" Ay: "); Serial.print(Ay);
Serial.print(" Az: "); Serial.print(Az);
Serial.print(" T: "); Serial.print(T);
Serial.print(" Gx: "); Serial.print(Gx);
Serial.print(" Gy: "); Serial.print(Gy);
Serial.print(" Gz: "); Serial.println(Gz);
if (Ax>-0.05 && Ax<0.11 && Ay>-0.10 && Ay<0.16 && Az>0.95 && Az<1.03){
 for (int i=0 ;i<100 ; i++) {
    digitalWrite(D8, HIGH);
    delay(1);
   Serial.println("SEARCH THE MAX VAL");
 float max,current, maxI;
 for(int j = 0; j < 100; j++){
  current = ul[j];</pre>
```

Serial.println("SEARCH THE AVG VAL");

unsigned int index; long arraySum;

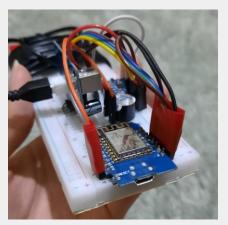
```
for(index = 0; index < sizeof(ul)/sizeof(ul[0]); index++)</pre>
   { arraySum += ul[index]; }
   float avg = arraySum/100;
   Serial.print("JSN MAX VALUE: ");
   Serial.print("JSN AVG VALUE: ");
   upload(avg, max, Ax, Ay, Az, T, Gx, Gy, Gz);
 else{
 digitalWrite(D8, HIGH);
 digitalWrite(D8,LOW);
 void I2C_Write(uint8_t deviceAddress, uint8_t regAddress, uint8_t data){
void upload(float avg, float max, double Ax,double Ay,double Az,double T,double Gx,double Gy,double Gz)
   Firebase.RTDB.setInt(&fbdo, "SENSE/ULTRA/HEIGHT(AVG)", avg);
   Firebase.RTDB.setInt(&fbdo, "SENSE/ULTRA/HEIGHT(HST)", max);
```

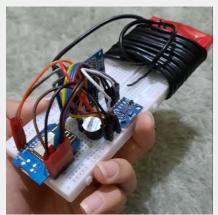
Firebase.RTDB.setInt(&fbdo, "SENSE/GYRO/AY", Ay);

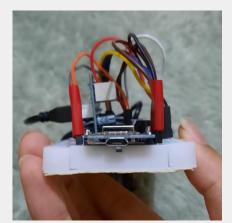
```
Firebase.RTDB.setInt(&fbdo, "SENSE/GYRO/GZ", Gz);
   Firebase.RTDB.setInt(&fbdo, "SENSE/GYRO/TEMP", T);
   Serial.println();
   Serial.println("UPLOADED");
   Serial.println("PATH: " + fbdo.dataPath());
   Serial.println("TYPE: " + fbdo.dataType());
void Read_RawValue(uint8_t deviceAddress, uint8_t regAddress){
 Wire.beginTransmission(deviceAddress);
 Wire.endTransmission();
 Wire.requestFrom(deviceAddress, (uint8_t)14);
 AccelX = (((int16_t)Wire.read()<<8) | Wire.read());</pre>
 AccelY = (((int16_t)Wire.read()<<8) | Wire.read());</pre>
 AccelZ = (((int16 t)Wire.read()<<8) | Wire.read());</pre>
 Temperature = (((int16_t)Wire.read()<<8) | Wire.read());</pre>
 GvroX = (((int16 t)Wire.read()<<8) | Wire.read());</pre>
 GyroY = (((int16 t)Wire.read()<<8) | Wire.read());</pre>
 GyroZ = (((int16_t)Wire.read()<<8) | Wire.read());</pre>
void MPU6050_Init(){
 delay(150);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SMPLRT_DIV, 0x07);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_1, 0x01);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_PWR_MGMT_2, 0x00);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_CONFIG, 0x00);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_GYRO_CONFIG, 0x00);//set +/-250 degree/second full
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_ACCEL_CONFIG, 0x00);// set +/- 2g full scale
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_FIF0_EN, 0x00);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_INT_ENABLE, 0x01);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_SIGNAL_PATH_RESET, 0x00);
 I2C_Write(MPU6050SlaveAddress, MPU6050_REGISTER_USER_CTRL, 0x00);
```

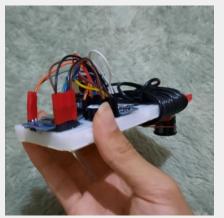


HARDWARE

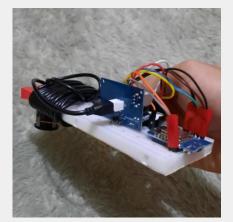












PROTOTYPE



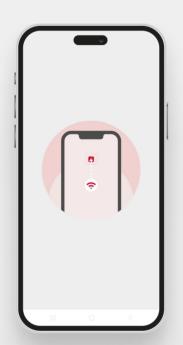
UMET

INTRODUCING

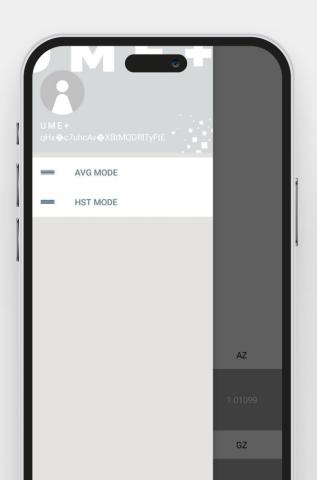




SPLASH SCREEN









HEIGHT RESULT



GYRO		
AX	AY	AZ
0.08228	0.03271	1.00488
GX	GY	GZ
-18.8855	9.56489	-2.06107
qHx�c7uhcAv�XBtMQDRITyFtE		
m	0	<

"Accuracy based on flatness value created by gyroscope sensor, when the minimum's fulfilled, the device take the height data and send the last gyroscope data into the app"



User's name

Hardware Device's Unique ID

Average Calculation Algorithm Mode

Highest Value Calculation Algorithm Mode







"Highest Value Algorithm (HST Mode) is a mode that going to calculate and show the highest value of 100 measurements per section using an ultrasonic sensor"

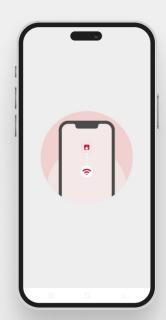


HST MODE

"Average Value Algorithm (AVG Mode) is a mode that going to calculate and show the average value of 100 measurements per section using an ultrasonic sensor"



AVG MODE















https://bit.ly/3DhhFww

THANKS!

Eʻriki20082003@gmail.com IGʻ@riqyuzet