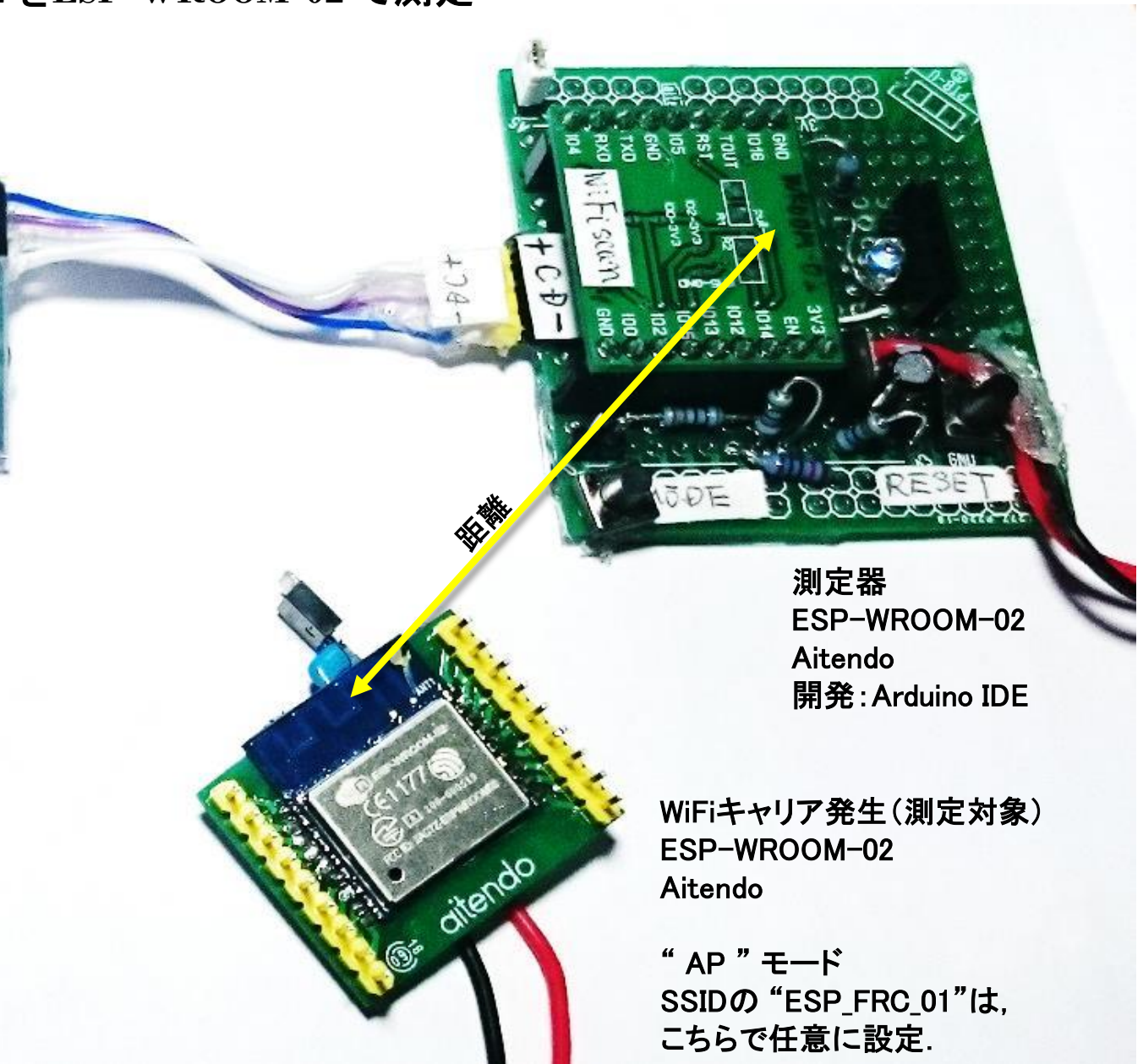


— ESP-WROOM-02 をESP-WROOM-02 で測定 —



AQM1602XA-RN-GBW-IIC Akizuki
LCD 1602D1 by Strawberry-Linux 互換性あり

ACM1602NI は未だ、表示できず。
バックライト付きなので使用した……
タイミングを様々と変えてみたが……



測定器
ESP-WROOM-02
Aitendo
開発: Arduino IDE

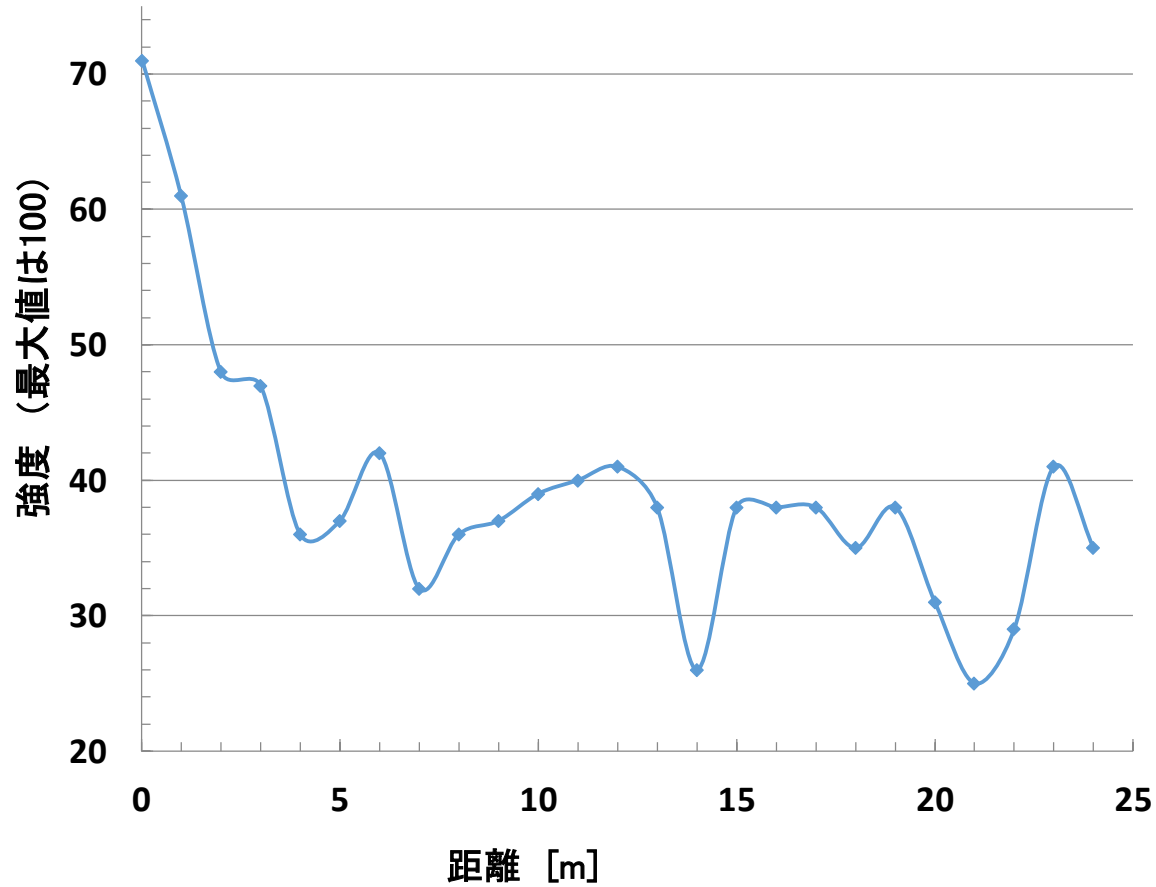
WiFiキャリア発生(測定対象)
ESP-WROOM-02
Aitendo

“ AP ” モード
SSIDの“ESP_FRC_01”は、
こちらで任意に設定.

ESP8266 受信強度測定結果

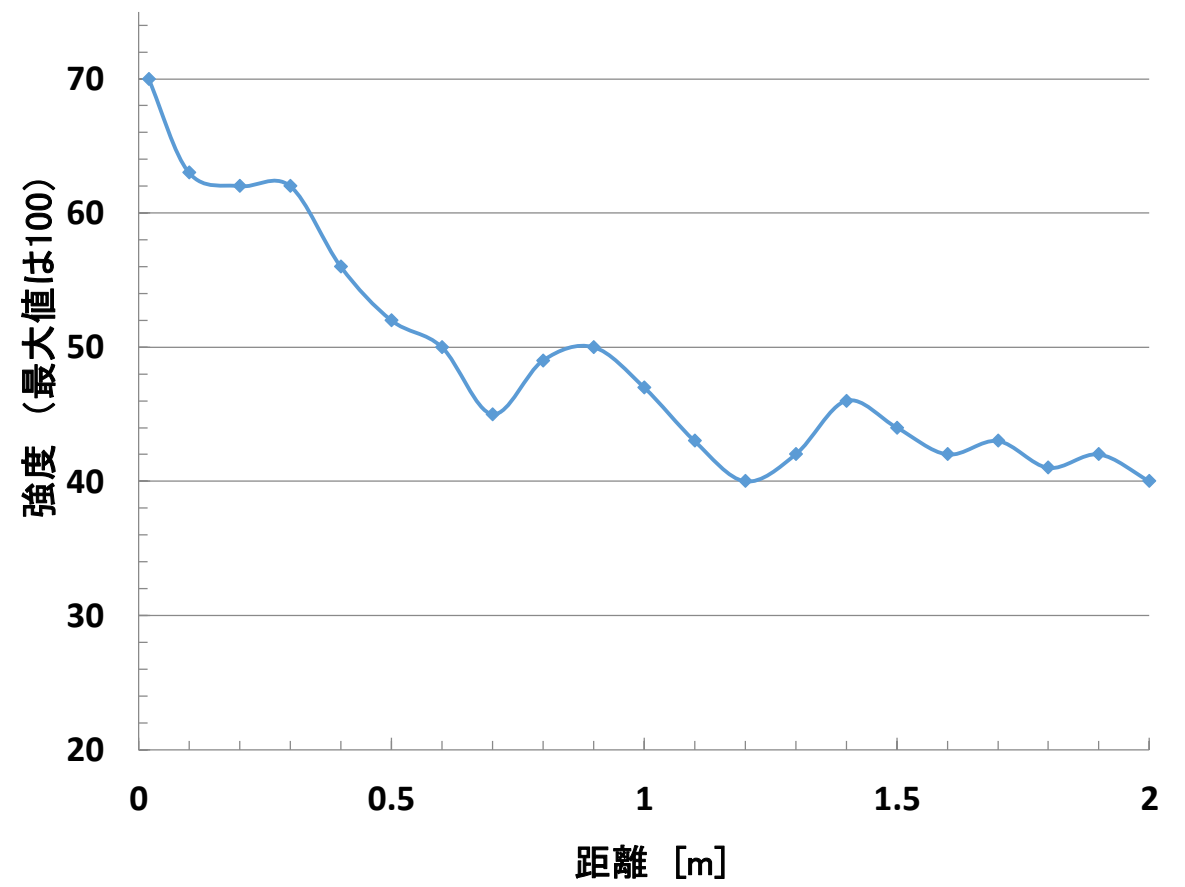
—— 方向探知基礎データ ——

遠距離



条件: 鉄筋4F廊下

近距離

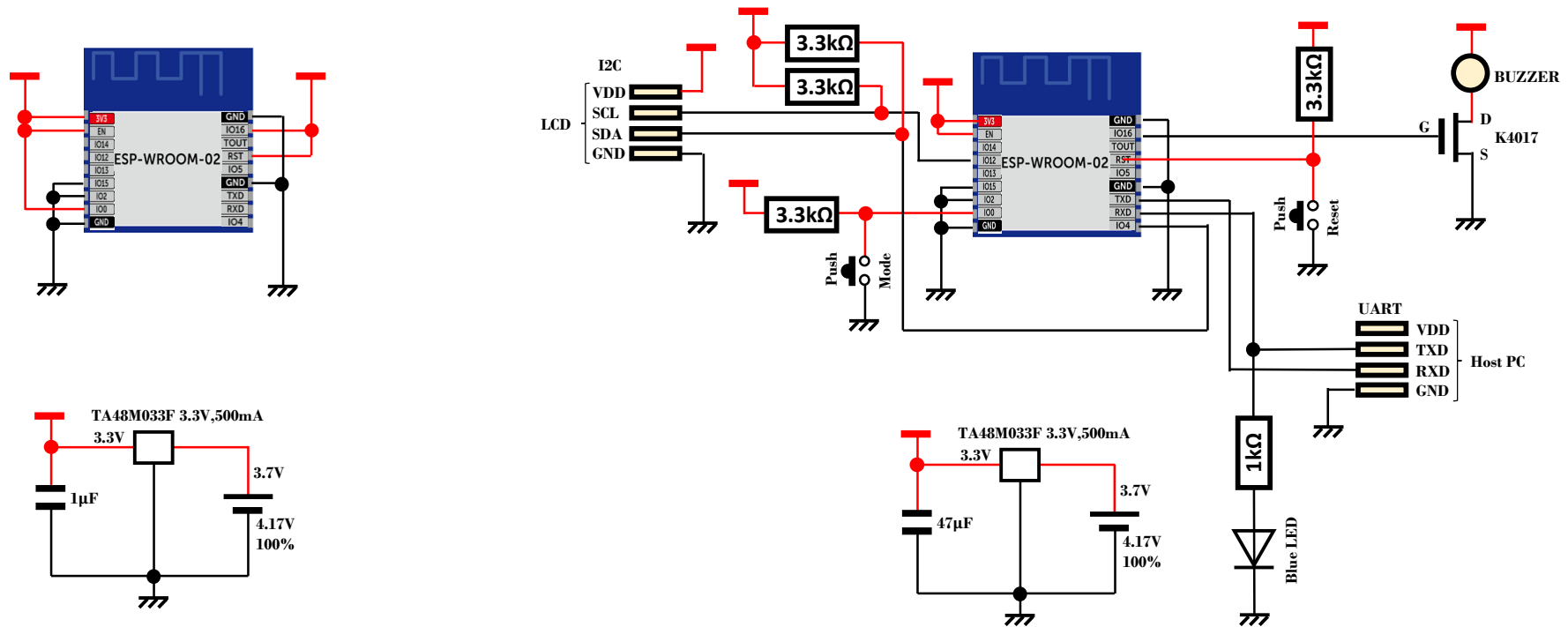


条件: 木造2F

APモードWiFiキャリア発生 ・ 受信強度測定回路図

WiFiキャリア発生(測定対象)
ESP-WROOM-02
Aitendo

測定器
ESP-WROOM-02
Aitendo
開発: Arduino IDE



超軽量R/C機発見装置, 事情により有線で取付けできない物体(生物など含む)の存否, 位置検出など応用範囲は広い

他人様ESP8266の元図 http://qiita.com/umi_kappa/items/ac3d37db44a2dcfe71fd

```

/* 2017.3.8
WiFi_ESP8266_1602D1_V3_Inten.ino
---- WiFi Radio Intensity Detector ----
WiFi module WROOM-02 by aitendo
LCD 1602D1 by Strawberry-Linux

ref: http://blog.goo.ne.jp/sim00/e/ab138be751d447bcdb0eecaaca232214 */

#include <Wire.h>
#include "ESP8266WiFi.h"

#define ADDR 0x3e
#define C_Low 0x70
#define C_High 0x56 // 0x5c?
#define Cont 9 // Contrast

uint8_t cmd_cr[] = {0xc0}; // C/R
uint8_t cmd_cl[] = {0x01}; // Clear Display

void setup() {
uint8_t cmd_init1[] = {0x38, 0x39, 0x14};
uint8_t cmd_init2[] = {C_Low | (Cont & 0x0f), C_High | (Cont >> 4 & 3), 0x6c};
uint8_t cmd_init3[] = {0x38, 0x0d, 0x01}; // 0x0d => 0x0c ?

uint8_t mess_1[] = "Setup done";

Serial.begin(115200);
delay(10);
//Wire.begin(4, 14);
Wire.begin(4, 12);
delay(40);

command(cmd_init1, sizeof(cmd_init1));
command(cmd_init2, sizeof(cmd_init2));
delay(300);
command(cmd_init3, sizeof(cmd_init3));
delay(2);

WiFi.mode(WIFI_STA);
delay(100);
Serial.println();
Serial.println("Setup done");

write(mess_1, sizeof(mess_1));
delay(000);
command(cmd_cl, sizeof(cmd_cl));
delay(10);
}

```

```

void loop() {
uint8_t scan_start[] = "scan start -----";
uint8_t scan_done[] = "scan done -----";
uint8_t scan_ssid[] = "          ";
char scan_inte[] = "          ";
int Inten;
int i, j;
String data;
while (1) {
command(cmd_cl, sizeof(cmd_cl));
delay(10);
write(scan_start, sizeof(scan_start));
Serial.println("scan start");
int n = WiFi.scanNetworks();
command(cmd_cr, sizeof(cmd_cr));
write(scan_done, sizeof(scan_done));
delay(500);
Serial.println("scan done");
if (n == 0)
Serial.println("no networks found");
else
{
Serial.println();
Serial.print(n);
Serial.println(" networks found");
Serial.println("-----");
command(cmd_cl, sizeof(cmd_cl));
delay(1);
for (i = 0; i < n; ++i) {
Serial.print(i + 1);
Serial.print(" ");

for (int j = 0; j < 16; ) {
if ( (WiFi.SSID(i))[j - 1] != ' ' ) {
scan_ssid[j] = (WiFi.SSID(i))[j];
++j;
}
}
for (j = 0; j < 16; j++) {
scan_inte[j] = ' ';
}
Inten = (WiFi.RSSI(i));
Inten += 100;
dtostrf(Inten, -5, 0, scan_inte);
command(cmd_cl, sizeof(cmd_cl));
delay(1);
write(scan_ssid, sizeof(scan_ssid));
command(cmd_cr, sizeof(cmd_cr));
for (int j = 0; j < 16; ) {
scan_ssid[j] = scan_inte[j] ;
++j;
}
}
}
}
}

```

```

write(scan_ssid, sizeof(scan_ssid));
Serial.print(WiFi.SSID(i));
Serial.print(" (");
Serial.print(scan_inte);
Serial.print(")");
Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? " " : "*");
delay(1000);
}
}
Serial.println("");
}
}
void command(uint8_t *cmd, size_t len) {
size_t i;
for (i = 0; i < len; i++) {
Wire.beginTransmission(ADDR);
Wire.write(0x00);
Wire.write(cmd[i]);
Wire.endTransmission();
delayMicroseconds(27); // 26.3us
}
}
void write(uint8_t *cmd, size_t len) {
size_t i;
for (i = 0; i < len; i++) {
Wire.beginTransmission(ADDR);
Wire.write(0x40);
Wire.write(cmd[i]);
Wire.endTransmission();
delayMicroseconds(27); // 26.3us
}
}
delay(1);
}
}

```