

```
// >--> MODULO SENSOR UMIDADE SOLO
// >--> TEMPO ENTRE MEDIDAS CONSECUTIVAS 1 MINUTO! PARA TESTES
EXCLUSIVAMENTE!!!
/* ADS1115 - This code is designed to work with the
ADS1115_I2CADC I2C Mini
Module available from ControlEverything.com.
https://www.controleverything.
com/content/Analog-Digital-Converters?
sku=ADS1115_I2CADC#tabs-0-product_tabset-2

>--> ATENCAO NAO FUNCIONA COM A ULTIMA VERSAO E SIM COM A
PENULTIMA DO DRIVER!
>--> ESP8266 - IDE DRIVER VERSION - 2.4.0 - rc1 - 23/02/2018 */

/* - - - - -
- - - - */
/* Soil Moisture Jet Fill Tensiometers
Product Features >--> http://www.surechem.com.my/products.php?
cat=2017
Direct measurement of soil water tension.
Allows easy replacement of the ceramic cup and dial gauge, and
addition
of extension tubes and the Service Cap. Soil Moisture Jet Fill
Tensiometers
available in a variety of lengths, ranging from 6 inches (15 cm)
to
60 inches (1.5 m). Insertion Tools can be used for coring a hole
in the
soil to accept these units. The Service Kit, available
separately, is
used to refill and maintain the tensiometer. An additional
accessory
available for all tensiometers is the "low tension" dialgauge.
This
gauge has a scale division from 0 to 50 centibars. This will
allow a
better and more accurate reading of the soil suction upto 50
centibars. */
/* - - - - -
- - - - */
```

```

/* MPX5700 - Series - freescale
0 to 700 kPa (0 to 101.5 psi) >-----> 0,00 to 7 bar
15 to 700 kPa (2.18 to 101.5 psi) >---> 0,15 to 7 bar
0.2 to 4.7 V Output >---> 700 kPa = 4,7 V >---> 1013 hPa >--->
101,3 cBar
Logo para a pressao atmosferica de 1013 hPa >--> Vout = 0,
680157142 V
Para 0,8465 V que esta medindo agora é equivalente a Patm = 126,
07 kPa. */

/* - - - - -
- - - -*/

/* 1 centiBar = 1 kPa = 10 hPa >--> 50 centibar = 50 kPa >-->
valor
tipico de presao para medida de tensao so solo! */

/* - - - - -
- - - -*/

// VERSAO PARA TESTES - NAO COMPATIVEL COM SISTEMA DO OLIVO
// >--> TEMPO ENTRE MEDIDAS CONSECUTIVAS 1 MINUTO!
// intervalo = 60000; // TEMPO DE SUBIDA PARA TESTES DE SENSORES
BME280
// VERSAO 2 - TAVARES - LED1 MUDOU PARA ATL-5 ANTES ERA LED0 NO
ATL-3
// VERSAO PARA TESTES NO SISTEMA ANTIGO DO RPi -
LINUX/APACHE/MYSQL/PHP
/* https://github.com/esp8266/Arduino
Arduino core for ESP8266 WiFi chip This project brings support
for
ESP8266 chip to the Arduino environment. It lets you write
sketches using
familiar Arduino functions and libraries, and run them directly
on
ESP8266, no external microcontroller required. ESP8266 Arduino
core comes
with libraries to communicate over WiFi using TCP and UDP, set up
HTTP,
mDNS, SSDP, and DNS servers, do OTA updates, use a file system in
flash
memory, work with SD cards, servos, SPI and I2C
peripherals.
*/

/* - - - - -

```

/ \*

VERSAO RPi TESTE DO CODIGO - UPDATED 050118  
VERAO 3.0 DATA: 01072017  
COMPILADO NA VERSAO ARDUINO: 1.8.1

PLACA WIFI ESP8266-07 AT THINKER  
PROGRAMA: MINI ESTACAO CLIMATICA  
CONTÃM SENSORES: BMP-180 E DHT22

CONFIGURACAO PLACA GRAVACAO - ESP-07  
FUNCIONA COM BIBLIOTECE ESP-07 COMMUNITY ATE

VERSAO 2.3.0

ATENCAO - VERSAO 2.4 NAO FUNCIONA PARA ESTE MODELO

ESP-07

ATENCAO NAO COMPILAR ESP-07 NA VERSAO 2.4 OU

SUPERIOR!!!!

```
PLACA:                GENERIC ESP8266 MODULE
FLASH MODE:           DIO
FLASH SIZE:           1M (512K SPIFFS)
DEBUG PORT:           DISABLED <--<
DEBUG LEVEL:          RIEN <--<
RESET MOTHOD:         ck
FLASH FREQUENCY:      40 MHz
CPU FREQUENCY:        80 MHz
UPLOAD SPEED:         115200
PORTA: PORTA ESP CONECTADA AO COMPUTADOR
```

CONFIGURACAO PLACA GRAVACAO - ESP-12E

```
PLACA:          NODE MCU 1.0 (ESP-12E MODULE)
CPU FREQUENCY:  80 MHz
FLASH SIZE:     4M (1M SPIFFS)
UPLOAD SPEED:   115200
PORTA:  PORTA ESP CONECTADA AO COMPUTADOR
```

 $\ast/$ 

---

— — — — \*

```
// AGROTECHLINK.COM - ESP8266 - PROGRAM HEADER TEMPLATE - 2017 -
```

NOVEMBER

```
//
//                                TODOS OS DIREITOS SAO RESERVADOS
/* - - - - -
- - - - */
/*  CADA GPIO POSSUI UMA IDENTIFICACAO ESPECIFICA
    PORTAS UTILIZADAS NAS PLACAS DA MINI ESTACAO CLIMATICA
    ATL3      >--> GPIO-16
    ATL4      >--> GPIO-14
    ATL5      >--> GPIO-12 + LED1
    ATL7      >--> GPIO-05 + SCL >--> PULLUP INTERNO / SENSOR
BMP-180 (PRESSAO)
    ATL8      >--> GPIO-04 + SDA >--> PULLUP INTERNO / SENSOR
BMP-180 (PRESSAO)
    /RST      >--> ---[10k]--+3V3  -//- JUMPER COM 0V P/ "RESET"
                +++[103]---CERAMIC CAPACITOR
    CH-PD     >--> ---[10k]--+3V3 + 103 CERAMIC CAPACITOR
                +++[103]---CERAMIC CAPACITOR
    GPIO-02   >--> ---[10k]--+3V3
    GPIO-00   >--> ---[10k]--+3V3  -//- JUMPER COM 0V P/ "FLASH"
    GPIO-15   >--> ---[10k]---0V
    RX + TX   >--> CONEXOES PARA GRAVADOR EXTERNO
    PROCEDIMENTO PARA GRAVACAO COM GRAVADOR FTDI
    FTDI-TX   >--> ATL-RX
    FTDI-RX   >--> ATL-TX
    FTDI-3V3  >--> ATL-3V3
    FTDI-0V   >--> ATL-0V
    NUNCA ALIMENTAR ESTE MODULO DIRETAMENTE PELO GRAVADOR
    OU USB DO COMPUTADOR! */
/* - - - - -
- - - - */
// LIVRARIAS EXTERNAS PARA FUNCIONAMENTO DOS SENSORES - CONEXAO -
DADOS
/* - - - - -
- - - - */
#include      <ESP8266WiFi.h>                // BIBLIOTECA WiFi DO ESP8266
#include      <Wire.h>                        // NECESSÁRIO PARA
COMUNICACAO I2C (PRESSAO)
#include      "ADS1115.h"
#include      <MySQL_Connection.h>           // CONEXAO COM BANCO DE DADOS
#include      <MySQL_Cursor.h>               // CONEXAO COM BANCO DE DADOS
```

```

/* - - - - -
- - - -*/

// AGROTECHLINK MINI ESTACAO CLIMATICA - PINOUTS - DEFINES -
DESCRICOES

/* - - - - -
- - - -*/

#define          ATL3          16          // GPIO-16 | + LED0 >-->
PRIMEIRA PLACA

#define          ATL4          15          // GPIO-15 + ESTADO NORMAL
DO ESP / PERMITE ROTINAS E RESTART

#define          ATL5          12          // GPIO-12 + LED1 >-->
PLACAS VERSAO TAVARES

#define          ATL7          5           // GPIO-05 + SCL >--> PULLUP
INTERNO / SENSOR BMP-180 (PRESSAO)

#define          ATL8          4           // GPIO-04 + SDA >--> PULLUP
INTERNO / SENSOR BMP-180 (PRESSAO)

#define          ATL9          2           // GPIO-02 + LED NATIVO DO
ESP8266 / PERMITE ROTINAS E RESTART

/* - - - - -
- - - -*/

// SENSOR PINS E CONFIGURACAO Wi-Fi

/* - - - - -
- - - -*/

#define          WIFI_SSID      "ATLRPi"    // NOME DA INTERNET DO
RASPBERRY-PI

#define          WIFI_PASSWORD "agrotechlinkPI2017" // SENHA DA
INTERNET

/* - - - - -
- - - -*/

// DEFINICAO DAS VARIAVEIS GLOBAIS

/* - - - - -
- - - -*/

char              MAC[25];                // VARIAVEL MAC PARA
O MySQL

String            mac;                    // VARIAVEL MAC
STRING TO CHAR PARA O MySQL

double            V_mpx;                  // SENSOR UMIDADE DO SOLO

unsigned long      tempoPrevio = 0;        // VARIAVEL DE
CONTROLE DE TEMPO

unsigned long      intervalo = 20000;      // VARIAVEL PARA

```



```

0.0625mV (default)
    ads.setMode(MODE_CONTIN);          // Continuous conversion mode
    ads.setRate(RATE_32);              // 128SPS (default)
    ads.setOSMode(OSMODE_SINGLE);}    // Set to start a
single-conversion
/* gain ADC PGA - choose one as follows - The ADC gain (PGA),
Device
operating mode, Data rate can be changed via the following
functions
GAIN_TWO - 2x gain    +/- 2.048V  1 bit = 0.0625mV (default)
GAIN_TWOTHIRDS - 2/3x gain +/- 6.144V  1 bit = 0.1875mV
GAIN_ONE - 1x gain    +/- 4.096V  1 bit = 0.125mV
GAIN_FOUR - 4x gain   +/- 1.024V  1 bit = 0.03125mV
GAIN_EIGHT - 8x gain  +/- 0.512V  1 bit = 0.015625mV
GAIN_SIXTEEN - 16x gain +/- 0.256V  1 bit = 0.0078125mV
mode - ADC conversion mode - choose one as follows:
MODE_CONTIN // Continuous conversion mode
MODE_SINGLE  // Power-down single-shot mode (default)
rate - samples per second - choose one as follows:
RATE_8 | RATE_16 | RATE_32 | RATE_64 | RATE_128 (default) |
RATE_250 | RATE_475 | RATE_860
osmode - choose one as follows:
OSMODE_SINGLE // Set to start a single-conversion */
/* - - - - -
- - - - */

void setup() {
    pinMode(ATL5, OUTPUT);    digitalWrite(ATL5, HIGH);    //
GPIO-16 + LED0 / INICIA HIGH E TERMINA SETUP LOW
    pinMode(ATL4, OUTPUT);    digitalWrite(ATL4, HIGH);    //
GPIO-15 + ESTADO NORMAL DO ESP / HIGH
    pinMode(ATL9, OUTPUT);    digitalWrite(ATL9, HIGH);    //
GPIO-02 + ESTADO NORMAL DO ESP / HIGH
/* - - - - -
- - - - */

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        yield();}

/* - - - - -
- - - - */

    mac = WiFi.macAddress();

```

```

/* - - - - -
- - - - */
unsigned num = 1;
unsigned mysqlResposta;
mysqlResposta = conn.connect(server_addr, 3306, user, password);
while (conn.connect(server_addr, 3306, user, password) != true)
{yield();}
/* - - - - -
- - - - */
digitalWrite(ATL5, LOW);    // GPIO-16 + LED0 / DESLIGA SETUP OK!
ADSconfig();  ads.begin();}
/* - - - - -
- - - - */
// FIM DO SETUP E CONFIGURACOES. INICIO DO LOOP.
/* - - - - -
- - - - */
void loop() {
    unsigned long currentMillis = millis();
    if (currentMillis - tempoPrevio >= intervalo) {        // SOBE OS
PRIMEIROS DADOS NO PRIMEIRO MINUTO
        digitalWrite(ATL5, HIGH);                            // GPIO-16
+ LED0 | LIGADO. ESTOU VIVO!
        tempoPrevio = currentMillis;
//  intervalo = 300000;                                // 5 MINUTOS (TEMPO DE
SUBIDA)
        intervalo = 60000;                                // 1 MINUTO (TEMPO DE
SUBIDA PARA TESTES DE SENSORES BME280)
/* - - - - -
- - - - */
        int16_t ReadSoil;
        ReadSoil = ads.Measure_SingleEnded(0);
        V_mpx = ReadSoil * Vgain;
/* - - - - -
- - - - */
char SV_mpx[6], query[170], SC_cnt[255];
// CONVERTENDO DADOS DOS SENSORES PARA STRINGS
        dtostrf(V_mpx, 4, 4, SV_mpx);
        dtostrf(C_cnt, 4, 0, SC_cnt);
        mac.toCharArray(MAC, 25);
/* - - - - -

```



```
- - - -*/  
sprintf(query, INSERT_SQL, MAC, SV_mpx, SC_cnt);  
// CONCATENANDO A STRING INSERT_SQL PARA GRAVACAO NO BANCO DE  
DADOS  
    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);  
    cur_mem->execute(query);        // SUBINDO DADOS PARA O BANCO  
    delete cur_mem;                // DELETANDO A QUERY EXECUTADA DA  
MEMORIA  
        C_cnt++;                    // INCREMENTA O CONTADOR DE  
MEDICOES  
} yield();  
digitalWrite(ATL5, LOW);}          // LED1 | DESLIGA AO FINAL DO ENVIO  
PARA O RPi  
/* - - - - -  
- - - -*/  
  
// MAIN FUNCTION END - FINAL  
/* - - - - -  
- - - -*/
```