AC – DC echipamente electronice srl
Cluj-Napoca, str. Muncitorilor, nr. 18, ap 8
tel: 0745-659909 www.acdcelectronics.ro

Placa de dezvoltare SDK cu WiFi ESP-12 si accesorii

Manual de utilizare

WiFi ESP-12 comunica cu dispozitive performante folosind interfete spi, i2c si i2s:

- spi → micro SD card
- spi → display ST7735
- spi → memorie sram 23LCV1024
- i2c → traductor BMP180 pentru temperatura, presiune si altitudine
- i2c → expander MCP23017
- i2c → memorie flash 24AA256
- i2s → audio DAC MAX98357

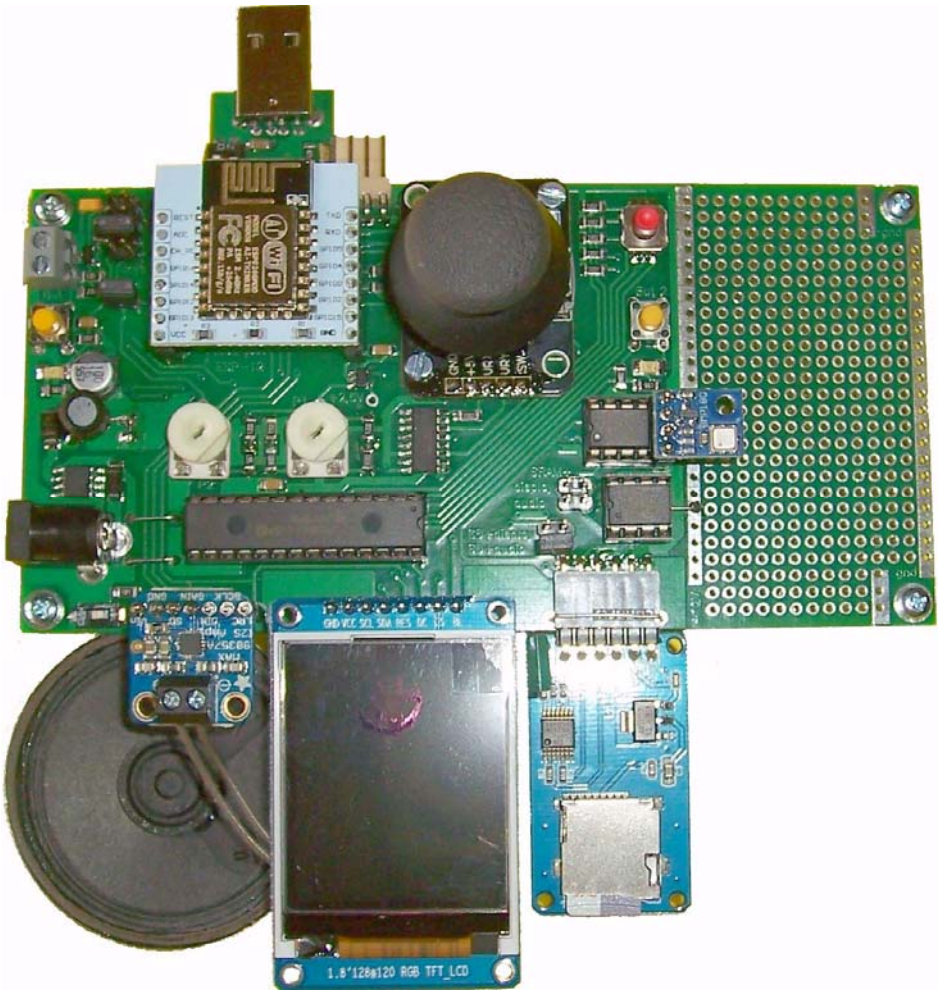


Fig. 1. Placa SDK pentru modulul ESP-12, echipata cu accesorii

Nota 1: Pentru o navigare usoara, recomandam sa deschideti cuprinsul acestui document .pdf. Placa SDK - ESP-12 serveste la experimentarea comunicatiilor cu dispozitivele de mai sus.

Capitole dedicate prezinta utilizarea convertorului ADC, joystick analogic si joystick digital.

Placa serveste totodata la elaborarea, dezvoltarea aplicatiilor complexe cu aceste dispozitive sau altele, vezi capitol.

Nota 2: Documentatia placii contine 60 exemple Arduino cu dispozitivele mentionate, librarii selectate, modificate si nou realizate.

Exemplele sunt simple si asigura repere solide pentru realizarea unor lucrari de laborator, proiecte de an si lucrari de diploma pentru studentii facultatilor cu profil electric.

Amplasarea principalelor componente

Schema electronica e construita in jurul dispozitivelor ESP-12 si expander i2c MCP23017.

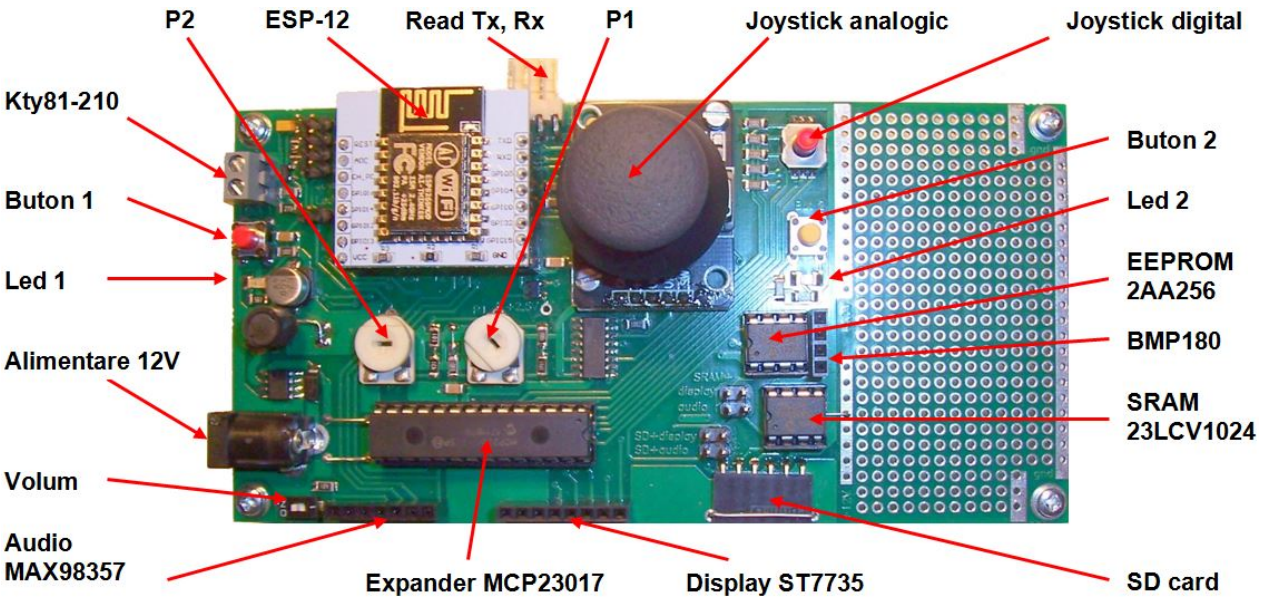


Fig. 2. Amplasarea principalelor componente

Jumpere

Functionarea simultana a dispozitivelor i2c expander MCP23017, SD card, display ST7735 si audio MAX98357 necesita 11 pini GPIO, insa ESP-12 ofera doar 10 !

Nota 3: Solutia e partajarea pinilor GPIO. Audio MAX98357 elibereaza 3 pini atunci cand lipseste iar display ST7735 elibereaza 2 pini.

Jumperele servesc la:

- Partajarea unor pini GPIO
- Conectarea unor subansamble de circuit la pinul ADC

E necesar sa folositi corect jumperele:

- Kty81-210
- Analog var
- Analog ref
- Analog joystick
- Led 1
- SD + display
- SD + audio
- SRAM + display
- SRAM + audio

Jumperele sunt marcate in clar pe circuitul imprimat.

Pinii GPIO ai modulului ESP-12

ESP-12 are 16 pini. Pinii Gnd, Vcc, Ch_Pd, Reset, Txd si ADC au intrebuintare unica. Raman doar 10 pini la dispozitia utilizatorului.

Pinii interfetelor spi, i2c, i2s si uart sunt neconfigurabili.

Pinii spi:

- SCK pin GPIO14
- MISO pin GPIO12
- MOSI pin GPIO13

Pinii i2c:

- SCL pin GPIO5
- SDA pin GPIO4

Pinii i2s:

- LRC pin GPIO2
- BCLK pin GPIO15
- DIN pin GPIO3, partajat cu RXD

Pinii uart:

- RXD pin GPIO3
- TXD pin GPIO1

Pinii expandati de i2c MCP23017

Circuitul expander i2c MCP23017 asigura 16 pini suplimentari dig I/O pe porturile A si B. Numerotarea pinilor se face antiorar, incepand de la GPA0 (pinul 0) pana la GPB7 (pinul 15).

Nota 4: Interfata i2c e emulata software (bit bitbanging) si lucreaza la 400kHz.
Timpul de raspuns e 200µs. Pentru comparatie, pinii rapizi GPIO raspund in 5µs.

Sunt 7 intrari expandate:

- BUTON_1 13
- BUTON_2 14
- DIG_JOY_1 12
- DIG_JOY_2 9
- DIG_JOY_3 10
- DIG_JOY_4 13
- DIG_JOY_6 8

Sunt 4 iesiri expandate:

- CS_SD_EXP 0
- CS_RAM_EXP 1
- LED_2 2
- LCD_BACKLIGHT 5

Raman 5 pini expandati, disponibili pentru utilizari viitoare.

Functionarea SD card, display ST7735 si audio MAX98357

Procedura i2s cu MAX98357 si procedura grafica cu ST7735 sunt foarte rapide.
Atunci cand aceste dispozitive folosesc fisiere in SD card cer ca si SD card sa fie rapid. Functionarea e posibila doar atunci cand pinii chip select sunt pini rapizi GPIO.

Cele trei dispozitive **pot functiona simultan** atunci cand display ST7735 si audio MAX98357 nu folosesc fisiere salvate in SD card. Pot folosi fisiere salvate in flash, in memoria program si server web http.

In acest caz SD card foloseste chip select expandat si indeplineste alte activitati ex. achizitie date, scrierea, citirea valorilor in fisiere .txt, etc.

Atunci cand unul din dispozitivele display ST7735 sau audio MAX98357 folosesc fisiere salvate in SD card, programul atribuie automat un pin rapid GPIO pentru chip select SD card:

- Chip select SD+audio atunci cand folositi SD card si audio MAX98357.
- Chip select SD+display atunci cand folositi SD card si display ST7735.

Nota 5: Jumperele SD+audio si SD+display trebuie pozitionate corespunzator acestor situatii !

Pregatirea pentru lucru

Instalarea Arduino si librariilor ESP8266

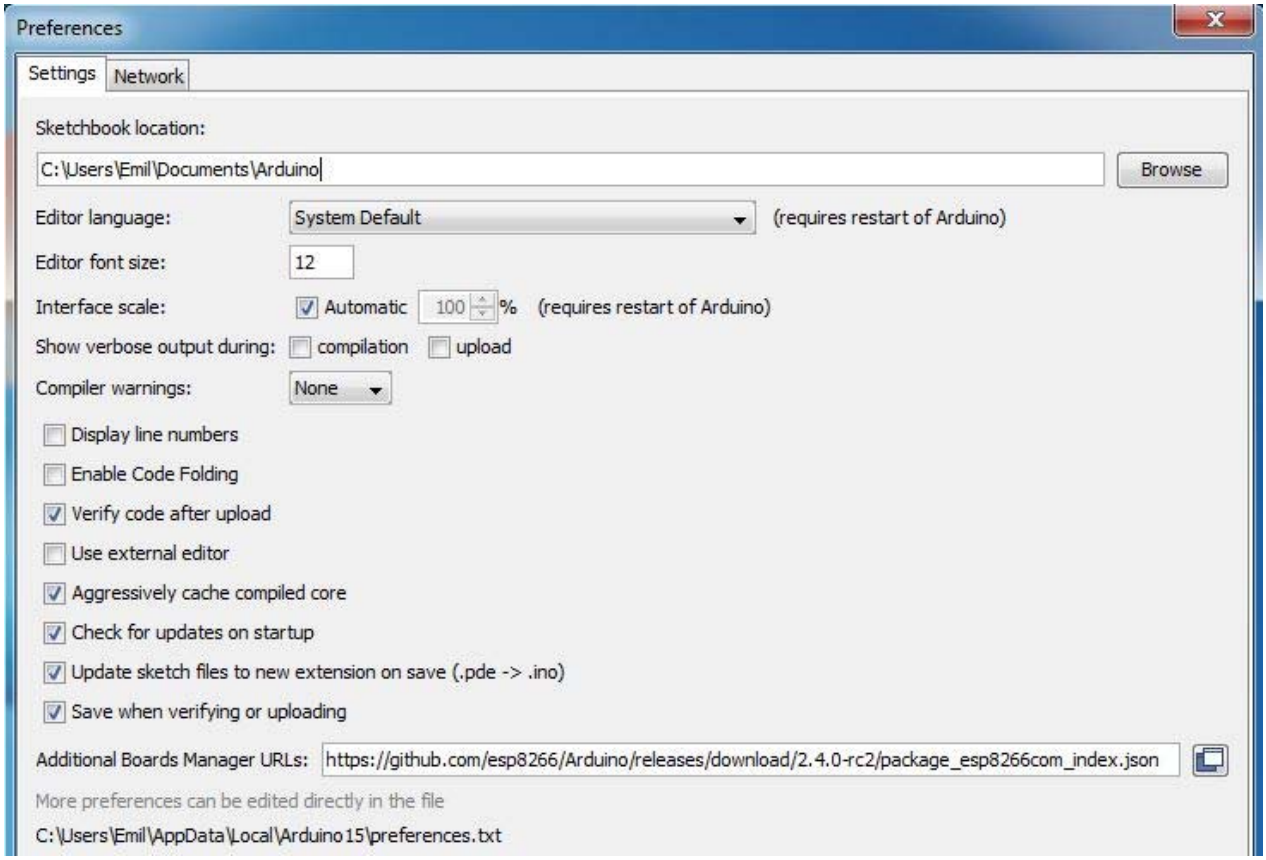
Toate exemplele sunt verificate cu mediul de dezvoltare **Arduino** → versiunea **1.8.5** → aici.

Nota 6: E avantajos sa-l instalati direct in radacina C:/ sau D:/.
In acest fel aveti un acces mai usor la librariile standard, folosind Notepad.

Toate exemplele sunt verificate cu libraria ESP8266, versiunea **2.4.0-rc2**.

Deschideti File / Preferences. Copiati linkul urmator in Aditional Boards Manager URLs:

https://github.com/esp8266/Arduino/releases/download/2.4.0-rc2/package_esp8266com_index.json



Apasati OK.

Deschideti Tools / Boards / Boards Manager, coborati pana la esp8266
si instalati versiunea 2.4.0-rc2.

Nota 7: In windows XP sunt erori la compilarea programelor.
Solutia e prezentata aici, rezovata de Robert Clementi in 6 martie 2016.

Instalarea librariilor auxiliare

Descarcati arhiva placa_SDK_esp-12 de pe site.
Copiati in radacina Arduino directorul placa_SDK_esp-12.

Directorul placa_SDK_esp-12 \ anexe \ librarii are 11 librarii auxiliare:

- Adafruit-GFX-Library-master
- Adafruit-MCP23017-Arduino-Library-master
- Adafruit-ST7735-Library-master
- BMP180-Breakout_Arduino-Library-master
- ESP-12_TestBoard
- ESP8266_Spiram-master
- ESP8266Audio-master
- Morse
- SD_expander
- SpiRAM_expander
- Time-master

Copiati aceste librarii in directorul Arduino \ libraries.

ESP12TestBoard.h e libraria proprie a placii de dezvoltare si cuprinde procedura expander MCP23017, definitii pentru butoane, leduri, chip select etc.

Procedura expander MCP23017 este apelata atat in programul sursa cat si in librariile modificate SDexp.h si SpiRAMexp.h.

Modificarile librariilor SDexp.h si SpiRAMexp.h sunt minore si se refera doar la atribuirea chip select expandat pentru SD card si spi SRAM atunci cand este nevoie.

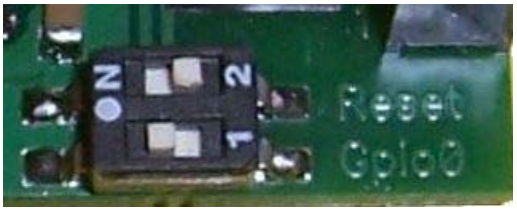
Atunci cand aceste dispozitive folosesc chip select pini rapizi GPIO, librariile modificate functioneaza in forma originala a librariilor SD.h si SpiRAM.h.

Instalarea uneltelor

Directorul placa_test_esp-12 \ anexe \ tools cuprinde doua unelte necesare.
Copiati ESP8266FS si EspExceptionDecoder in Arduino \ tools.

Pregatirea placii de dezvoltare cu ESP-12

Placa de dezvoltare functioneaza impreuna cu convertorul USB-UART pentru ESP-01.



Convertorul asigura incarcarea programelor, alimentarea placii SDK cu 3,3V si comunicatia cu monitorul serial Arduino.

Comutatoarele pinilor Reset si GPIO0 sunt marcate in clar pe circuitul imprimat.

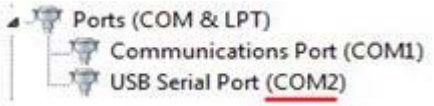
Placa de dezvoltare are conectorul pereche cu 8 pini montat pe partea inferioara.
Conectati convertorul USB-UART la placa de dezvoltare ESP-12 conform fig.1.

Configurarea mediului de dezvoltare Arduino

Deschideti mediul de dezvoltare Arduino.

Conectati un cablu prelungitor USB intre PC si convertorul USB-UART.

In Device Manager, PC-ul vede COM portul convertorului USB-UART:



Nota 8: Puteti modifica numarul COM portului in Device Manager / USB Serial port (COM2) / Properties / Port Settings / Advanced.

Selectati acest port in Arduino / Tools / Port. Selectati Upload Speed 115200.

In Arduino / Tools selectati:

- Board: Generic ESP8266 Module
- FlashFrequency: 80MHz
- CPU Frequency: 160MHz
- Flash Size: 4M(3M SPIFFS)

Nota 9: Partajarea Flash Size arata ca 1 Mbyte e alocat programului si 3 Mbyte sunt alocati fisierelor de date.

Denumirea fisierelor de date este in format 8.3. Numele fisierului are max 8 caractere ex. pisica_1, urmata de **.** si extensia 3 caractere ex. bmp, wav, mp3, txt etc.

Incarcarea fisierelor de date in flash

Deschideti i2s_mp3_spiffs.ino, similar primului exemplu de initiere.

Directorul Data atasat acestui exemplu contine fisiere .bmp, .wav si .mp3.
Aceste fisiere vor fi incarcate in memoria flash 3M SPIFFS disponibila in ESP8266.

Pregatiti comutatoarele GPIO0 si Reset pentru modul de lucru download programm.

Inchideti monitorul serial Arduino. Selectati Tools / ESP8266 Sketch Data Upload.

Incarcarea e marcata la 3%, sunt 3Mbyte si dureaza ceva timp chiar la 115200 baud.

Nota 10: Fisierele de date raman incarcate in flash, independent de incarcarea exemplelor.

Pregatirea SD card

Incarcati in SD card fisierele de date cuprinse in placa_SDK_esp-12 \ anexe \ fisiere_sd_card.

Rularea exemplelor

Exemplele program sursa se afla in placa_SDK_esp-12 si sunt grupate pe categorii.

Important: Extrageti jumperele folosite in exemplele anterioare.

Fiecare exemplu arata jumperele folosite: niciunul, unul singur, maxim doua.
Singurele jumpere nevinovate sunt cele analogice, pinul ADC nefiind partajat.

Nota 11: Schema electronica trebuie sa fie intotdeauna in fata dumneavoastra !

Nota 12: Cititi cu atentie notele informative de la inceputul exemplelor.
Aflati astfel dispozitivele auxiliare pe care le folositi si cum sa pozitionati jumperele.

Exemple de initiere

Pornim de la exemplul cel mai simplu blink_led.ino.

Acest exemplu se afla in Arduino / placa_SDK_esp-12 / exemple initiere / blink_led
si foloseste LED_1 si buzer conectat la GPIO16.

Faceti dublu click pe blink_led.ino si programul se deschide in mediul de dezvoltare Arduino.

Folosirea comutatoarelor GPIO0 si Reset in modul de lucru Download programm

Incarcati programul in ESP8266. Sunt trei pasi:

1. Aduceti comutatorul GPIO0 in stare ON (la masa). ESP8266 este configurat astfel pentru incarcarea programului.
2. Resetati circuitul ESP8266. In acest scop aduceti comutatorul Reset in stare ON (la masa) dupa care reveniti in starea OFF.
3. Selectati Arduino / Sketch / Verify / Compile. Daca rezultatul compilarii e OK, selectati Upload in Arduino / Sketch / Upload.

In cursul incarcarii programului, Arduino prezinta mesajul:

"Sketch uses 222,201 bytes (51%) of program storage space. Maximum is 434,160 bytes.
Global variables use 31,576 bytes (38%) of dynamic memory, leaving 50,344 bytes for local variables. Maximum is 81,920 bytes.
Uploading 226352 bytes from C:\Users\Emil\AppData\Local\Temp\build89dffa342730d36061756adb8c22691b.tmp/My_Blink_ESP.ino.bin to flash at 0x00000000
..... [36%]
..... [72%]
..... [100%]"

Modulul ESP8266 trece imediat la rularea programului.

Folosirea comutatoarelor GPIO0 si Reset in modul de lucru Run programm

Sunt trei modalitati de rulare a unui program deja incarcat:

1. Aduceti comutatorul GPIO0 in stare OFF. Resetati circuitul ESP8266. In acest scop aduceti comutatorul Reset in stare ON (la masa) dupa care reveniti in stare OFF.
2. La power up, modulul ruleaza programul doar daca GPIO0 si Reset sunt in OFF. Functia Reset e indeplinita automat.

Nota 13: Programul e lansat in executie odata cu un power up provenind de la sursa externa 12V. In acest caz **nu** este folosit convertorul USB-UART.

Blink led expander

Exemplifica folosirea iesirilor expandate de circuitul i2c MCP23017.
Procedura expander e definita in libraria Esp12TestBoard.h.

Procedura expander este globala si poate fi apelata atat in program cat si in librariile SDexp.h si SpiRAMexp.h.

Acest exemplu foloseste LED_2.

Rulati blink_led_expander.ino.

Morse SOS library

Exemplifica realizarea propriiilor librarii astfel incat procedurile sunt apelate in program.
Acest exemplu foloseste LED_1 conectat la GPIO16.

Codul scris in program este mai scurt, elegant.
Deschideti NotePad si studiati libraria <morse.h>, <morse.cpp>.

Rulati morse_sos_library.ino.

PWM fade led

Exemplifica folosirea unei iesiri PWM cu rezolutia factorului de umplere = 1/1024.
Rezultatul apare atat luminos pe LED_1 cat si acustic pe buzer, frecventa fundamentala 1kHz.

Rulati pwm_fade_led.ino.

Buton simplu

BUTON_1 comanda LED_1 si LED_2.

Rulati simple_button.ino.

NTP UDP client

WiFi ESP8266 este client UDP, interogheaza serverul ro.pool.ntp.org si obtine ora UTC.

Rulati ntp_udp_client.ino. Exemplul ntp_wake_clock.ino trateaza zona de timp si ora de vara.

Transmiterea seriala a comenzilor

Exemplifica transmiterea seriala a comenzilor de la monitorul serial Arduino.

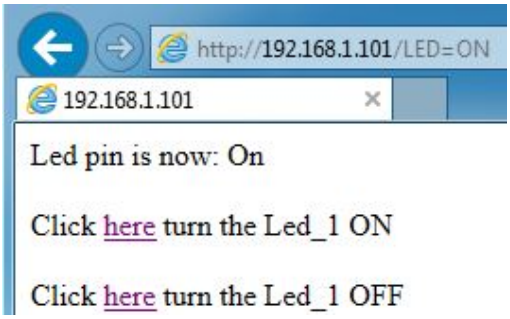
Rulati send_serial_command.ino.

Comanda din browser a ledului

WiFi ESP8266 este server HTTP si detine propria pagina web.

Adresa wireless LAN o aflati imediat dupa reset in monitorul serial Arduino.

Comanda ledului se face in browser de la orice PC si smartphone conectate la router:



Rulati web_command_led.ino.

Comanda din Punctul de Acces la Statie

Exemplele prezinta comanda WiFi din Punctul de Acces la Statie, pentru aprinderea unui led.

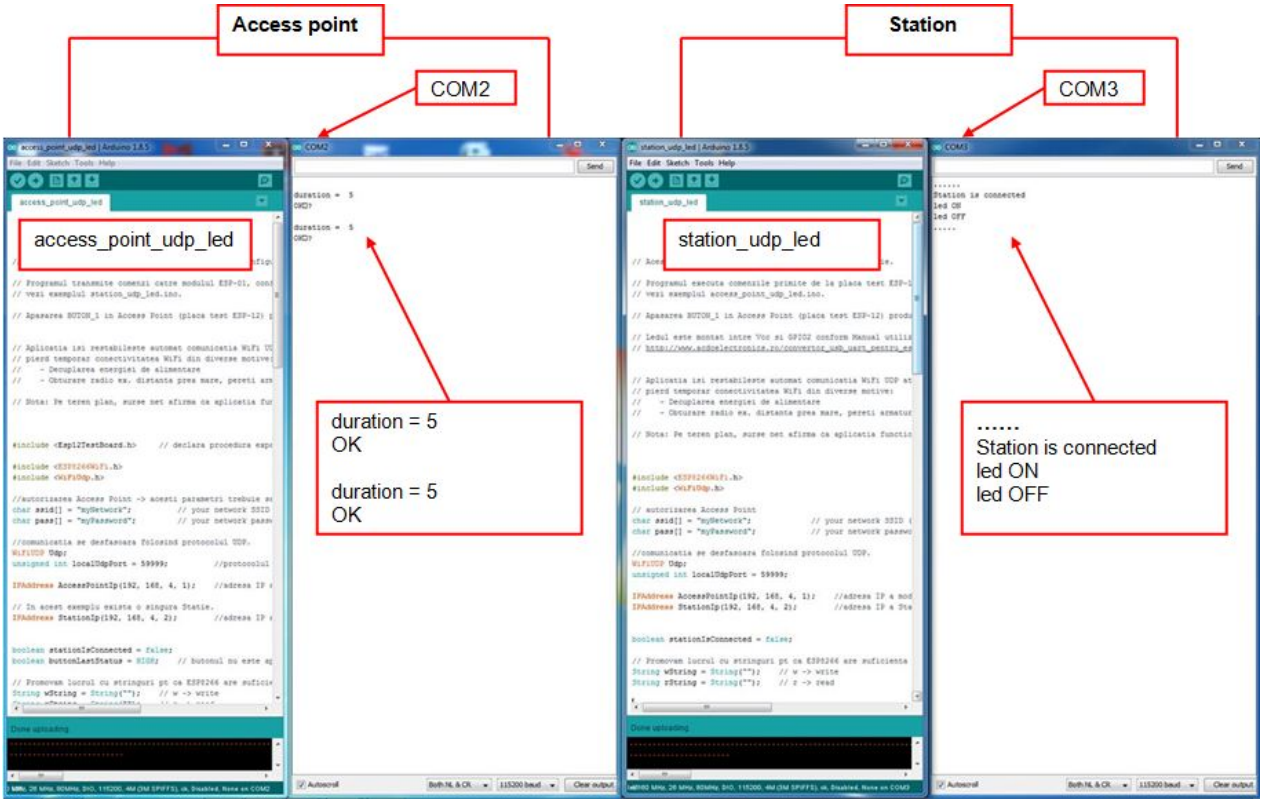
Access Point e placa de dezvoltare ESP-12 incarcata cu access_point_udp_led.ino.

Statia e un modul ESP-01 incarcat cu station_udp_led.ino.

Nota 14: E avantajos sa deschideti in Arduino ambele exemple.
E necesar sa identificati in Device manager / ports doua COM porturi diferite ex. COM2, COM3 si sa selectati corespunzator aceste porturi in Arduino.

In acest fel, in monitoarele seriale Arduino veti observa in timp real comunicatia intre cele doua dispozitive WiFi ESP8266.

Comanda incepe la apasarea butonului BUTTON_1 in Access Point si se sfarseste odata cu receptionarea mesajului OK transmis de Statie. Durata unei comenzi este ~ 5 milisecunde.



Aplicatia isi restabileste automat comunicatia WiFi UDP atunci cand Statia sau Access Point pierd temporar conectivitatea WiFi din diverse motive:

- Decuplarea energiei de alimentare
- Obturare radio ex. distanta prea mare, pereti armatura fier, alte conditii geografice

Surse net afirma ca distanta conexiunii WiFi ajunge pana la 500m pe teren plan.

Nota 15: Utilizarea modulului WiFi ESP-01 si montarea ledului aferent e prezentata in manualul de utilizare al convertorului USB-UART pentru ESP-01.

Alte exemple mai elaborate sunt prezentate in capitolele joystick digital si joystick analogic.

Nota 16: Conexiunea WiFi asigura transmiterea comenzilor in ambele sensuri.
Cu alte cuvinte puteti realiza alte exemple in care comenzile se transmit de la Statie la Access point sau bilateral.

Exemple ISR

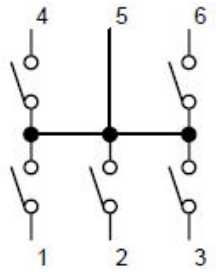
Exemplifica procedura ISR → Interrupt Service Routine, procedura fundamentala atunci cand desfasurati o activitate in paralel cu activitatile din bucla principala loop().

ISR asigura functionarea pseudo multitasking a activitatilor.

Rulati exemple din acest capitol.
Alte exemple sunt incluse in joystick digital si prelucrarile analogice ADC.

Exemple joystick digital

Joystick digital foloseste intrari expandate si insumeaza 5 comutatoare. Combinatia acestora defineste 5 stari: sus, jos, stanga, dreapta si apasat. In plan orizontal, sus, jos au semnificatia de inainte, inapoi.



	Pin 1	Pin 2	Pin 3	Pin 4	Pin 6
Up	0	0	0	1	1
Down	1	0	1	0	0
Left	1	0	0	1	0
Right	0	0	1	0	1
Push	1	1	1	1	1

Rulati dig_joystick_0.ino.

Comanda WiFi cu viteza constanta a doua motoare cc

Utilizari propuse → Comanda WiFi a unui vehicul cu deplasare pe doua coordonate ex:

- Actionarea unui pod rulant si a macaralei
- Un vehicul care ocoleste obstacole
- Plantarea rasadurilor intr-o sera
- Un elevator intr-o magazie cu rafturi pe verticala, pozitionate lateral fata de un culoar de trecere

Comanda digitala apasat o folositi cum doriti.

Aplicatia foloseste doua module ESP8266 care comunica in WiFi, protocol UDP.

Access Point e placa de dezvoltare ESP-12 incarcata cu joystick_digital_access_point_udp.ino.

Statia e un modul ESP-01 incarcat cu joystick_digital_station_udp.ino.

Aplicatia isi restabileste automat comunicatia WiFi UDP atunci cand Statia sau Access Point pierd temporar conectivitatea WiFi din diverse motive:

- Decuplarea energiei de alimentare
- Obturare radio ex. distanta prea mare, pereti armatura fier, alte conditii geografice

Surse net afirma ca distanta conexiunii WiFi ajunge pana la 500m pe teren plan.

Aplicatia e demonstrativa. Comenzile receptionate de Statie sunt doar printate.

Dezvoltarea aplicatiei implica realizarea vehiculului comandat prevazut cu acumulator, motoare de curent continuu, limitatoare de cursa si schimbarea de sens.

Exemple analogice ADC

Partea din stanga a schemei electronice e alocata utilizarii convertorului analogic ADC:

- Traductor de temperatura KTY81-210
- Divizor rezistiv in gama 0 ~ 1,0V.
- Divizor rezistiv in gama 0,935 ~ 0,985V pentru determinarea valorii referintei analogice Aref.
- Joystick analogic X / Y si multiplexor analogic 74HC4051

Traductorul de temperatura KTY81-210

Acest exemplu si corectia valorii analogice Aref sunt prezentate detaliat in tutorialul nostru.

Rulati adc_kty81_210.ino, adc_kty81_210_isr.ino si web_kty81_210.ino.

PWM fade led ADC

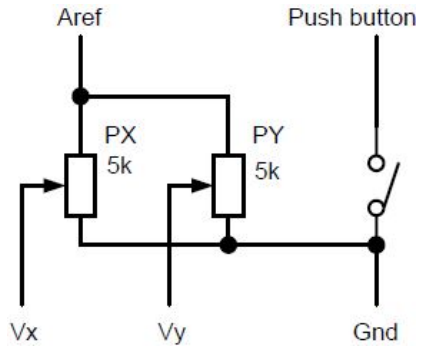
Exemplifica folosirea intrarii analogice ADC conectata la potentiometrul P2.

Frecventa fundamentala e 1kHz insa timbrul si intensitatea sunetului variaza in functie de factorul de umplere. Sunetul buzerului se aude cel mai clar la 50% PWM.

Semnalul PWM poate fi vizualizat cu osciloscopul pe pinul GPIO16.
Rulati pwm_fade_led_adc.ino.

Joystick analogic

Joystick analog are doi potentiometri alimentati la valoarea referintei analogice Aref si un buton.



Rulati joystick_analog_0.ino.

Aref se determina experimental conform tutorialul nostru. In acest scop folositi exemplul check_analog_ref.ino → asigurati 1023 pasi ADC din potentiometrul P1.

Potentiometrii joystick-ului sunt pozitionati pe mijloc si asigura ~ Aref / 2, respectiv ~ 512 pasi.
Multiplexorul analogic 74HC4051 e comandat pe linia S0. S1 si S2 sunt legate la masa.

Nota 17: Joystick analogic functioneaza in lipsa convertorului i2s audio MAX98357, pentru ca partajeaza aceiasi pini.

Comanda WiFi PWM a doua motoare cc

Utilizare propusa → Comanda unui vehicul cu viteza variabila → motor inainte, inapoi si directie stanga, dreapta. Comanda digitala la apasarea butonului o folositi cum doriti.

Aplicatia foloseste doua module ESP8266 care comunica in WiFi, protocol UDP.

Access Point e placa de dezvoltare ESP-12 incarcata cu joystick_analog_access_point_udp.ino. Potentiometrii Px si Rx sunt alimentati la valoarea referintei analogice Aref, conform exemplu precedent joystick_analog_0.ino.

E realizata calibrarea automata a potentiometrilor pentru pozitia de repaus.

Statia e un modul ESP-01 incarcat cu joystick_analog_station_udp.ino.

Aplicatia isi restabileste automat comunicatia WiFi UDP atunci cand Statia sau Access Point pierd temporar conectivitatea WiFi din diverse motive:

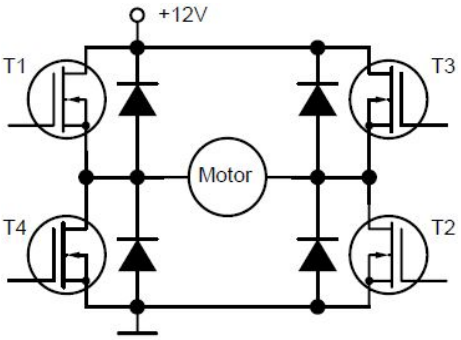
- Decuplarea energiei de alimentare
- Obturare radio ex. distanta prea mare, pereti armatura fier, alte conditii geografice

Surse net afirma ca distanta conexiunii WiFi ajunge pana la 500m pe teren plan.

Aplicatia e demonstrativa. Comenzile PWM receptionate de Statie sunt doar printate.

Dezvoltarea aplicatiei implica realizarea vehiculului prevazut cu acumulator, motoare de curent continuu, cremaliera pentru directie, limitatoare de cursa, variatia prin PWM a tensiunii de alimentare a motoarelor si schimbarea de sens. Directia poate fi realizata si cu un servomotor.

Puntea electronica in **H** asigura viteza variabila si schimbarea de sens folosind comanda PWM.



Perechile de tranzistoare T1,T2, respectiv T3, T4 sunt comandate alternativ in PWM.
Motorul ramane in repaus atunci cand toate tranzistoarele sunt blocate.
Diodele asigura calea de evacuare spre acumulator a energiei inductive generata de motorul cc.

Joystick analogic si display grafic

Acest exemplu foloseste joystick analogic si display grafic ST7735.
Display grafic afiseaza pozitia joystick sub forma unui punct miscator, valorile conversiei ADC pe axele x, y si starea butonului.

Rulati joystick_analog_display_pos.ino.

Exemple i2c

Traductorul i2c BMP180, temperatura si presiune

Traductorul BMP180 are adresa i2c = 0x77h specificata in libraria SFE_BMP180.h.
Rulati i2c_bmp180.ino.

Extensie i2c eeprom cu 24AA256

ESP8266 are 4MB memorie flash, suficienta pentru orice program si numarul ciclurilor stergere, scriere este ridicat.

De ce este nevoie uneori de memorie eeprom suplimentara?

Sunt aplicatii in care diverse valori trebuie salvate in flash de mai multe ori intr-o zi.
Dupa un timp mai lung sau mai scurt numarul maxim al ciclurilor stergere, scriere e depasit si nu mai putem conta pe ESP8266 !

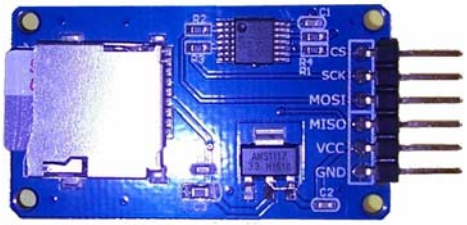
Memoria eeprom 24AA256 are 1.000.000 de cicluri stergere, scriere !

Pinii A0, A1, A2 sunt cablati la masa. Adresa i2c = 0x50.

Rulati i2c_eeprom_24aa256_0.ino, i2c_eeprom_24aa256_1.ino si i2c_eeprom_24aa256_2.ino.

Exemple spi cu micro SD card

Placa micro SD card are sursa interna 5V → 3,3V.
Strapam sursa interna pentru a alimenta placa direct la 3,3V.



Rulati CardInfo.ino, Dataloger.ino, Files.ino, listfiles.ino, ReadFile.ino, ReadWrite.ino.

Exemple spi cu extensie SRAM

Sram ESP8266 e aprox 80Kbyte si in functie de cerintele aplicatiei devine uneori insuficient, vezi acest capitol.

Placa e dotata cu extensie sram 128Kbyte furnizata de circuitul 24LCV1024.
Placa functioneaza si cu extensie sram 256 si 512 Kbiti, configurabila in exemplele urmatoare.

Adresarea memoriei se face la nivel de octet.
In functie de marimea memoriei, adresa are 2 sau 3 octeti, configurabila automat in librarie.

Rulati spi_sram_functions.ino, spi_sram_string.ino si spi_sram_test.ino.

Exemple cu display grafic ST7735

Prezinta functionarea afisajului grafic 1,8'' cu ST7735. Afiseaza pe rand patru fisiere .bmp:




Exemplifica controlul expander al luminii de fundal backlight cu BUTON_1.

Afisajul grafic si fisierele .bmp au 128 x 160 pixeli.
Puteti realiza propriiile fisiere convertind, editand alte fisiere imagine. Folositi ex. IfranView.

Fisier .bmp salvat in SD card

Rulati lcd_bmp_file_sd.ino.

Fisier .bmp salvat in flash

Rulati lcd_bmp_file_spiffs.ino.

Fisier .bmp salvat in web server HTTP

Rulati lcd_bmp_file_http.ino.

Prezentarea performantelor grafice ST7735

Rulati lcd_graphictest.ino, lcd_rotationtest.ino si lcd_text_size.ino.

Exemple audio cu MAX98375

Fisierele ex .wav si .mp3 sunt salvate in SD card, memoria flash, memoria program sau server web http.

Circuitul i2s MAX98357 e un convertor DAC.
Inductanta foarte mare a difuzorului integreaza in curent semnalul de la iesirea MAX98357.

Semnalul poate fi vizualizat cu un osciloscop si o sonda de curent bransata pe unul din conductoarele difuzorului.

Volumul se regleaza din comutatorul Volum 3dB ~ 12dB, vezi fig. 2.
Volumul maxim = 12dB este in pozitia stanga.

Nota 18: Libraria esp8266Audio.h este foarte recenta. Cititi mai multe de la autorul librariei.

Functii periodice definite in program

In program sunt definite tabelar 4 functii periodice: sinus, triunghi, dinte fierastrau, dreptunghi.
O perioada e definita in 360 puncte cu valori cuprinse intre -1 si +1.

Sunetul e livrat la 2kHz, corespunzator unei frecvente de esantionare = 44.100 Hz.
Sinus are sunetul cel mai curat. Armonici superioare se aud la dreptunghi si dinte fierastrau.

Rulati i2s_functions.ino.

Fisier .wav salvat in flash

Functioneaza cu fisiere .wav standard. Fisierele .wav standard au un format simplu, vezi link.

Headerul e unic, declarat la inceput si frecventa de esantionare e 44.100 Hz.
Fisierele .wav compresate nu vor functiona.

Rulati i2s_wav_spiffs.ino.

Fisier .wav salvat in memoria program

Rulati PlayWAVFromPROGMEM.ino.

Fisier .mp3 salvat in flash

Fisierele .mp3 au o structura mult mai complexa. Headerul se repeta pe parcursul fisierului. Utilizatorii au posibilitate sa decupeze portiuni din fisierul original, ex. folosind Audacity.

Rulati i2s_mp3_spiffs.ino.

Fisier .mp3 salvat in SD card

Rulati i2s_mp3_sd_card.ino.

Fisier .mp3 salvat in web server HTTP

Resursele sram ESP8266 nu sunt suficiente pentru procedurile WiFi si procesarea complexa a fisierelor mp3 salvate in web server.

Memoria aditionala sram 23LCV1024 a avut initial un scop mai mult didactic insa in acest caz rolul ei e structural.

Nota 19: Autorul librariei esp8266Audio.h a realizat recent aceasta importanta imbunatatire.

Rulati i2s_mp3_http_spi_ram.ino si StreamMP3FromHTTP_SPIRAM.ino.

Dezvoltarea aplicatiilor cu ESP-12

Placa de dezvoltare cu ESP-12 e un instrument puternic pentru experimentarea a noi aplicatii si proiectarea viitoarelor placi de circuit pe care vor functiona.

Puteti experimenta orice aplicatie functionand simultan cu SD card, display ST7735 si audio MAX98357. Realizarea in practica inseamna utilizarea acelorasi conexiuni si librarii.

Atunci cand folositi doar o parte din dispozitivele de mai sus si cei 10 pini ESP-12 sunt suficienti, renuntati la expander MCP23017 si obtineti simplificari hardware.

Nota 20: Puteti realiza orice conexiune indepartand circuitele integrate de pe placa de devoltare. Noua conexiune o realizati prin cabluri folosind soclurile si conectoarele existente. Nu aveti astfel nevoie de statie de lipit.

Conform schemei electronice a placii de dezvoltare, sunt doua moduri de rulare a programelor:

- Alimentare +3,3V si vizualizare mesaje seriale folosind convertorul USB-UART
- Alimentare 5~22V externa si folosirea dispozitivului USB de citire al traficului serial UART

Folosirea spatiului de dezvoltare

Placa are un spatiu generos de dezvoltare cu gauri metalizate, pentru experimentarea diverselor montaje electronice. Atunci cand experimentati aveti in vedere ca:

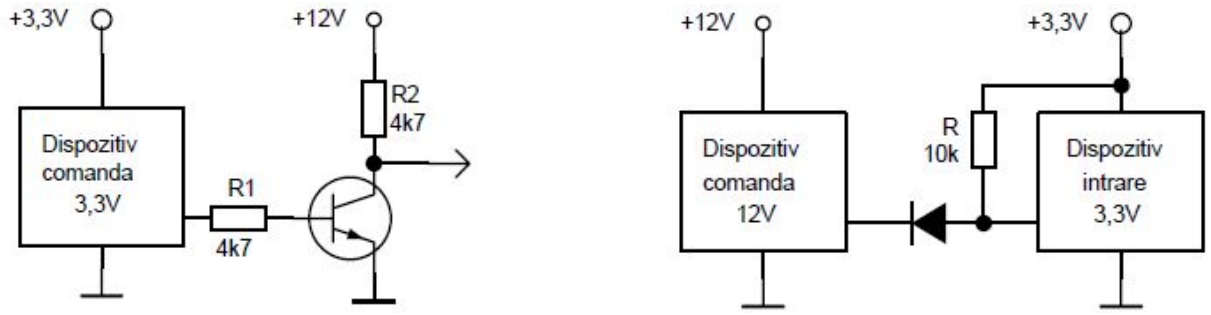
- Aveti deja o referinta 2,5V.
- Aveti 5 intrari, iesiri digitale disponibile, expandate de MCP23017.
- Puteti folosi tensiunea de alimentare externa ex. 12V

Folosirea tensiunii de alimentare externa

ESP-12 si circuitele electronice de pe placa sunt alimentate la 3,3V, provenind alternativ de la:

- Convertorul USB-UART
- Alimentarea externa 12V

Atunci cand folositi releu iesire, ventilator sau alt dispozitiv care necesita tensiune mai mare de alimentare, puteti utiliza alimentarea externa 12V accesibila in spatiul de dezvoltare.



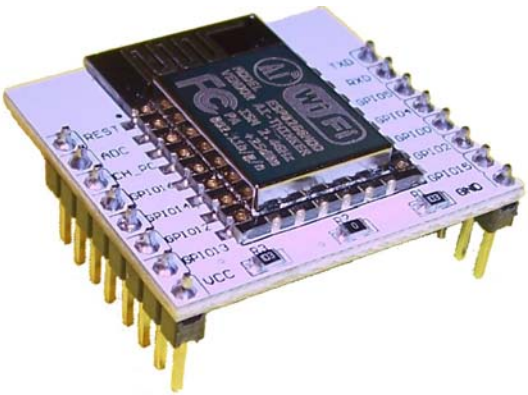
Dispozitivele sunt comandate de la pinii ESP-12 sau expander MCP23017.
E necesar un circuit electronic care adapteaza comanda 3,3V la 12V.

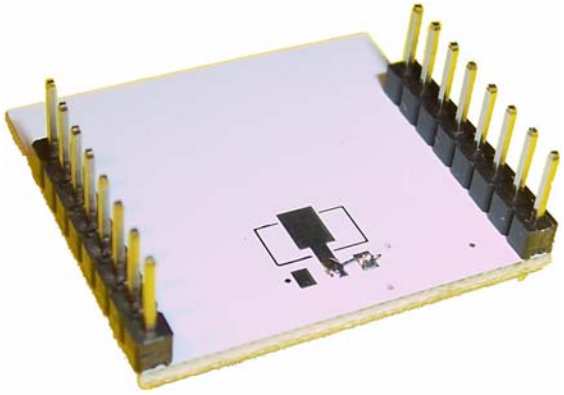
Cel mai simplu circuit e tranzistorul NPN colector in gol.
Aveti in vedere ca acest circuit inverseaza semnalul de comanda.

Atunci cand un dispozitiv alimentat la 12V comanda o intrare suportand max. 3,3V, ex. ESP-12 sau expander MCP23017, folositi dioda de separare.

ESP-12 montat pe placa adaptoare cu pas 2,54mm

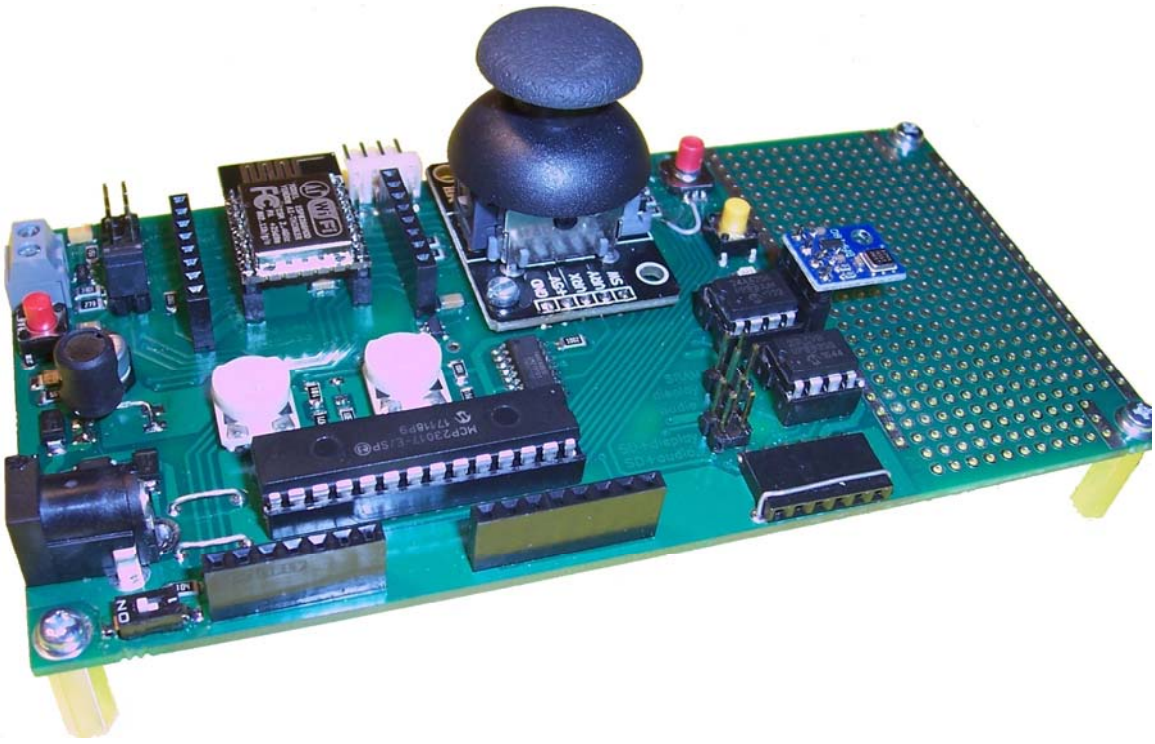
Placa de dezvoltare se livreaza impreuna cu modulul ESP-12 montat pe placa adaptoare prevazuta cu conectoare pas 2,54mm.





Sursa de alimentare e strapata pentru a alimenta placa direct la 3,3V.

ESP-12 prevazut cu conectoare 2mm



Placa de dezvoltare are conectoare mama cu pas 2mm prevazute pentru montarea ESP-12 cu conectoarele pereche lipite.

Placa de dezvoltare asigura astfel dezvoltarea altor aplicatii si incarcarea programului Arduino. ESP-12 urmeaza sa fie mutat pe placa proiectata optimizat a noii aplicatii.

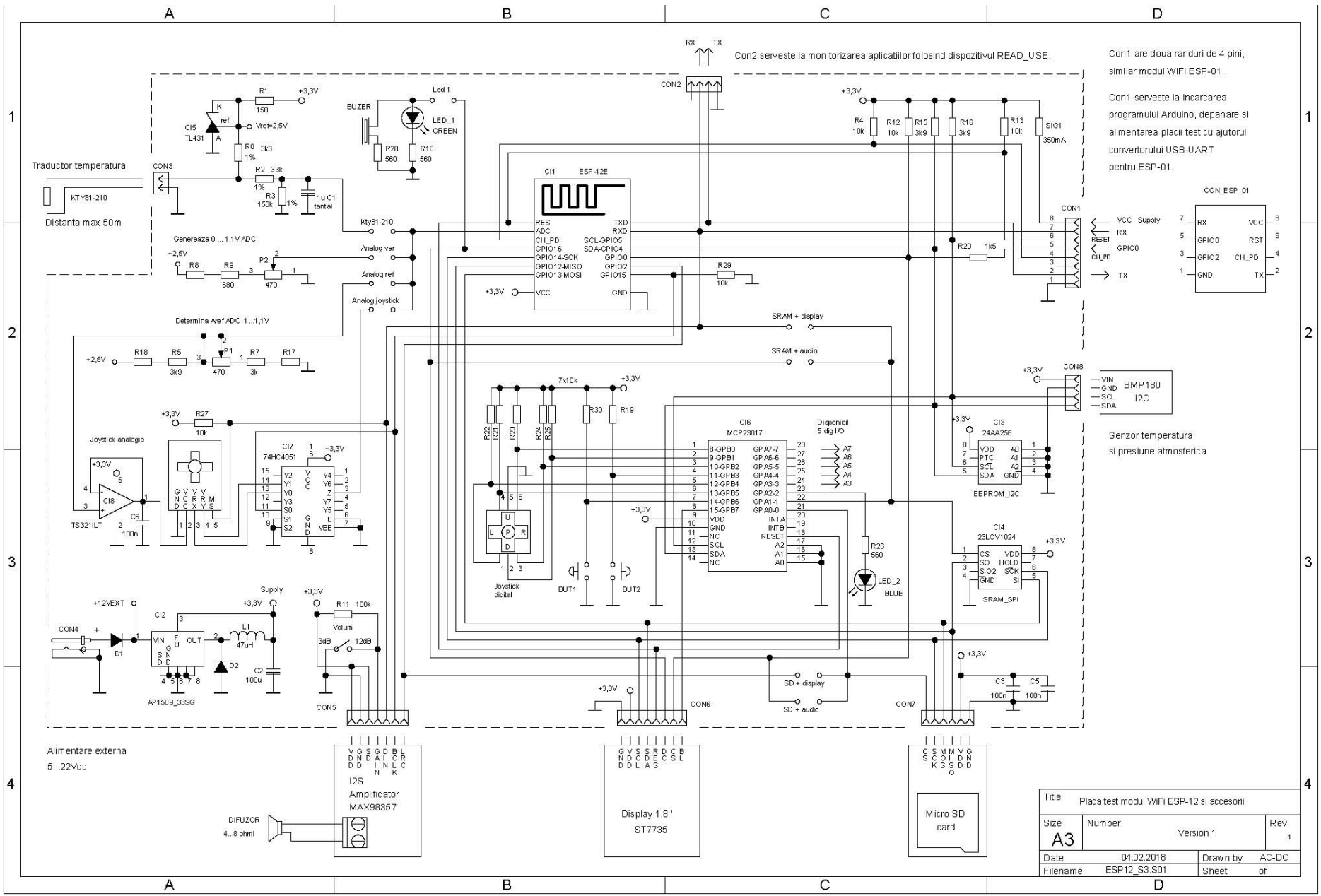
Rezistente de polarizare pentru ESP-12

Atunci cand elaborati o noua schema electronica cu ESP-12, e necesar sa puneti urmatoarele rezistente de polarizare, ex. valoare 10k:

- R13, pull up pe pinul RES
- R4, pull up pe pinul CH_PD
- R12, pull up pe pinul GPIO0
- R29, pull down pe pinul GPIO15

Nota 21: Rezistentele R4 si R29 exista pe placa adaptoare cu conectoare pas 2,54mm.

Schema electronica


Con2 serveste la monitorizarea aplicatiilor folosind dispozitivul READ_USB.
Con1 are doua randuri de 4 pini, similar modul WiFi ESP-01.
Con1 serveste la incarcarea programului Arduino, depanare si alimentarea placii test cu ajutorul convertorului USB-UART pentru ESP-01.
Traductor temperatura
Distanta max 50m
KTY81-210
Genereaza 0 ... 1,1V ADC
Determina Aref ADC 1...1,1V
Joystick analogic
TS321ILT
74HC4051
MCP23017
Disponibil 5 dig I/O
Joystick digital
BUT1
BUT2
LED_1 GREEN
LED_2 BLUE
ESP-12E
SRAM + display
SRAM + audio
24AA256
EEPROM_I2C
BMP180 I2C
Senzor temperatura si presiune atmosferica
23LCV1024
SRAM_SPI
Alimentare externa 5...22Vcc
AP1509_33SG
I2S Amplificator MAX98357
DIFUZOR 4...8 ohmi
Display 1,8'' ST7735
Micro SD card
SD + display
SD + audio
Title Placa test modul WiFi ESP-12 si accesorii
Size A3
Number
Version 1
Rev 1
Date 04.02.2018
Drawn by AC-DC
Filename ESP12_S3.S01
Sheet of