

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



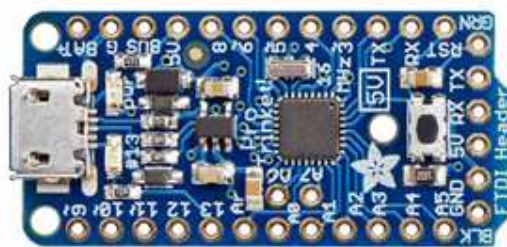
Codul sursă din acest document este licențiat

Public-Domain

Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără nici un fel de limitări dar autorii nu își asumă nici o răspundere pentru pagubele pricinuite de implementările realizate de utilizatori. Schemele și codul sursă au un rol educativ și nu sunt gândite pentru a fi utilizate în mediu de producție (industrial, casnic sau comercial).

# Utilizarea bibliotecii V-USB

V-USB (1) este o componentă software ce implementează comunicația USB (low-speed USB / USB 1.1) pentru orice microcontroler din familia Atmel AVR – cu alte cuvinte permite comunicația USB pentru microcontrolerele din această familie chiar dacă nu au un port USB hardware (similar cu biblioteca SoftwareSerial (2) ce permite utilizarea oricăror doi pini ai plăcii Arduino pentru comunicația UART). Prin intermediul acestei biblioteci putem utiliza orice microcontroler Atmel AVR pentru o conexiune directă USB fără a fi nevoie de un circuit suplimentar de conversie UART – USB (FTDI sau microcontroler cu port hardware USB – 8U2 sau 16U2). Singurele restricții sunt ca circuitul microcontroler să aibă cel puțin 2kB memorie flash, 128 bytes memorie RAM și să funcționeze la cel puțin 12MHz – condiții îndeplinite fără probleme de microcontrolerul ATmega328P ce echipează placa de dezvoltare Arduino Uno. O aplicație imediată a acestei biblioteci o reprezintă plăcile de dezvoltare compatibile Arduino ce nu dețin circuite suplimentare față de microcontroler (nu au circuit de conversie UART – USB și implicit au un cost mai mic), de exemplu placa de dezvoltare Adafruit Pro Trinket (3) – imagine alăturată. Placa Adafruit Pro Trinket este echipată cu un microcontroler ATmega328P, la fel ca și placa Arduino Uno, are dimensiuni și cost apropiate de o placă de dezvoltare Arduino Pro Mini (4) dar nu necesită programator extern FTDI pentru încărcarea programelor. La fel ca și celelalte plăci de dezvoltare Arduino, placa Pro Trinket funcționează pe baza unui bootloader rezident în memoria flash a microcontrolerului ce permite transferarea programului de pe USB în memoria program internă. Bootloader-ul plăcii Pro Trinket se bazează pe biblioteca V-USB și funcționează în mod similar cu bootloader-ele altor plăci Arduino. Singurul dezavantaj al unei astfel de soluții este procedura mai greoaie de upload (de încărcare a programului). Circuitele suplimentare prezente pe plăcile Arduino se ocupau cu transferul programului USB - memorie internă program lansând în execuție bootloader-ul intern printr-o operație de reset a microcontrolerului. Din cauza absenței circuitului care să declanșeze operația de reset această operație trebuie efectuată manual – înainte de operația de upload (încărcare a programului) trebuie apăsat butonul de reset și operația de încărcare trebuie efectuată într-un interval de 10 secunde (atâta timp cât placa se află în Bootloader Mode (5)).



## V-USB for Arduino

Aplicabilitatea bibliotecii V-USB nu se rezumă doar la dezvoltarea de bootloadere ce permit încărcarea programului în memoria internă a microcontrolerului. Prin intermediul bibliotecii V-USB putem dezvolta periferice USB bazate pe microcontrolere Atmel AVR de tip HID (Human Interface Device – tastatură, mouse), CDC (Communications Device Class – modemuri, mass storage) sau chiar MIDI (Musical Instrument Digital Interface) fără a fi limitați la clasa de dispozitive USB de tip USB-to-serial (USB Serial Converter) oferite de driverul obișnuit al plăcilor de dezvoltare Arduino. Site-ul V-USB oferă o colecție semnificativă de exemple de proiecte diverse de astfel de dispozitive ([7](#)).

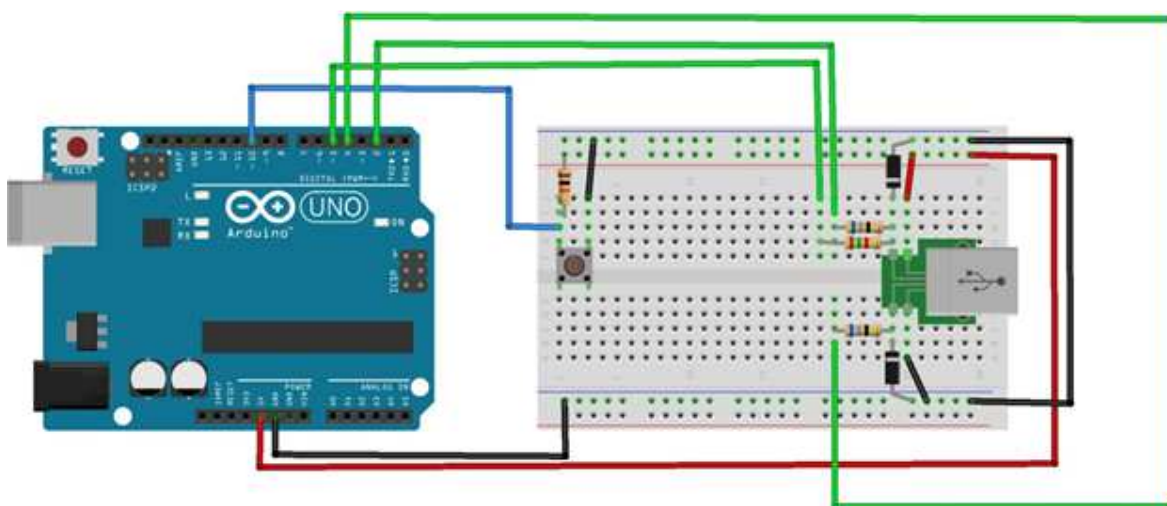
Chiar dacă biblioteca V-USB este scrisă în limbajul C nu poate fi folosită direct în mediul Arduino IDE fiind dezvoltată sub mediul WinAVR. Din fericire există o portare parțială a bibliotecii V-USB sub forma unei biblioteci specifice Arduino IDE și anume V-USB for Arduino ([6](#)). Prin intermediul acestei biblioteci o placă Arduino Uno se poate transforma într-un dispozitiv USB personalizat. Biblioteca este format din trei componente independente: UsbDevice (permite implementarea unui dispozitiv USB generic), UsbKeyboard (permite implementarea unui dispozitiv USB HID de tip tastatură), UsbStream (permite implementarea unui dispozitiv de comunicație / transfer de date).

Biblioteca V-USB for Arduino poate fi utilizată fără nici un fel de problemă sub versiunile Arduino IDE 1.0.x dar necesită o mică corecție sub versiunile mai recente (1.5.x, 1.6.x): declarațiile realizare în memoria program (PROGMEM) trebuie să fie de tip constantă (const) - în toate fișierele .c și .h declarațiile de tip PROGMEM trebuie modificate în PROGMEM const.

## Realizarea unui dispozitiv de tip „Carte de vizită”

În cadrul acestei secțiuni vom realiza un periferic de tip HID (mai exact tastatură) ce va completa în mod automat datele de contact ale deținătorului (proiect inspirat de [8](#)). Sistemul va fi format dintr-o placă de dezvoltare Arduino Uno, un buton (plus o rezistență de 10Kohm) și partea de interfațare electrică cu conexiunea USB (2 diode Zener 3.6V maxim 0.5W, 2 rezistențe 68ohm și 1 rezistență 1.5Kohm). La apăsarea butonului sistemul va trimite pe USB textul de semnătură (carte de vizită) ca și cum ar fi tastat.

Butonul este conectat la pinul digital 10, acesta este ținut la Vcc (+5V) prin intermediul unei rezistențe de 10Kohm. Apăsarea butonului trage pinul digital în "0" logic (0V). Conexiunea electrică USB presupune ca liniile D+, D- să aibă o rezistență de 68ohm în serie și câte o diodă Zener de 3.6V la masă (pentru adaptarea nivelului logic de 5V la 3.6V). Linia D+ necesită și o rezistență de pull-up de 1.5Kohm comandată de pinul digital 5. Linia D- se conectează la pinul digital 4 și linia D+ la pinul digital 2. Liniile conexiunii USB Vcc și GND se interconectează la pinii 5V și GND ai plăcii Arduino asigurând alimentarea sistemului în modul de funcționare periferic USB de tip HID. Schema de interconectare este următoarea:



Programul ce va rula pe placa de dezvoltare va fi încărcat utilizând Arduino IDE și conexiunea USB a plăcii.

```
#include "UsbKeyboard.h"
#include <util/delay.h>

#define BUTTON_PIN 10

String mesaj;

void setup() {
    pinMode(BUTTON_PIN, INPUT);
    TIMSK0&=! (1<<TOIE0);
```

```

}

void loop() {
    UsbKeyboard.update();
    if (digitalRead(BUTTON_PIN) == 0) {
        mesaj = "ING. POPESCU ION";
        KeyStrokeAlpNum(mesaj);
        mesaj = "DIRECTOR TEHNIC EMBEDDED DEVICES SRL";
        KeyStrokeAlpNum(mesaj);
        mesaj = "MOBIL 0789.000.233";
        KeyStrokeAlpNum(mesaj);
        mesaj = "FIX 021.456.30.03";
        KeyStrokeAlpNum(mesaj);
        UsbKeyboard.sendKeyStroke(KEY_ENTER);
        _delay_ms(200);    }
}

```

După încărcarea programului conexiunea USB cu placa de dezvoltare se va deconecta și se va utiliza conexiunea USB prin intermediul mufei de pe breadboard.

În cadrul programului este necesară includerea bibliotecii UsbKeyboard.h. În secțiunea setup se va configura pinul digital 10 ca intrare (pin utilizat pentru buton) și se va dezactiva timerul 0 (utilizat de mediul Arduino IDE pentru funcțiile de temporizare) pentru a nu interfera cu temporizarea utilizată în biblioteca V-USB. Secțiunea loop va conține un apel periodic al procedurii UsbKeyboard.update() – apel necesar pentru a efectua operații obligatorii în cadrul comunicației USB, operații invocate de procedura usbpool() (9) a bibliotecii V-USB; trimiterea de caractere prin intermediul conexiunii USB este condiționată de apăsarea butonului (digitalRead(BUTTON\_PIN) == 0) – trimiterea efectivă a codului unei taste se realizează prin intermediul funcțiilor UsbKeyboard.sendKeyStroke(cod\_tasta) pentru o singură tastă și KeyStrokeAlpNum(mesaj) pentru un șir de taste.

```

void KeyStrokeAlpNum(String Sc){
    int sPoint = Sc.length();
    for (int x = 0; x < sPoint; x++){

```

```
int y = x + 1;
if (Sc.substring(x, y) == "H"){
    UsbKeyboard.sendKeyStroke(KEY_H, MOD_SHIFT_LEFT);
}
else if (Sc.substring(x, y) == "L"){
    UsbKeyboard.sendKeyStroke(KEY_L, MOD_SHIFT_LEFT);
}
else if (Sc.substring(x, y) == " "){
    UsbKeyboard.sendKeyStroke(KEY_SPACE);
}
else if (Sc.substring(x, y) == "1"){
    UsbKeyboard.sendKeyStroke(KEY_1);
}
else if (Sc.substring(x, y) == "2"){
    UsbKeyboard.sendKeyStroke(KEY_2);
}
else if (Sc.substring(x, y) == "3"){
    UsbKeyboard.sendKeyStroke(KEY_3);
}
else if (Sc.substring(x, y) == "4"){
    UsbKeyboard.sendKeyStroke(KEY_4);
}
else if (Sc.substring(x, y) == "5"){
    UsbKeyboard.sendKeyStroke(KEY_5);
}
else if (Sc.substring(x, y) == "6"){
    UsbKeyboard.sendKeyStroke(KEY_6);
}
else if (Sc.substring(x, y) == "7"){
    UsbKeyboard.sendKeyStroke(KEY_7);
}
else if (Sc.substring(x, y) == "8"){
    UsbKeyboard.sendKeyStroke(KEY_8);
}
else if (Sc.substring(x, y) == "9"){
    UsbKeyboard.sendKeyStroke(KEY_9);
}
```

```
    }  
else if (Sc.substring(x, y) == "0"){  
    UsbKeyboard.sendKeyStroke(KEY_0);  
}  
else if (Sc.substring(x, y) == "."){  
    UsbKeyboard.sendKeyStroke(55);  
}  
else if (Sc.substring(x, y) == "-"){  
    UsbKeyboard.sendKeyStroke(45);  
}  
else if (Sc.substring(x, y) == "A"){  
    UsbKeyboard.sendKeyStroke(KEY_A, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "B"){  
    UsbKeyboard.sendKeyStroke(KEY_B, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "C"){  
    UsbKeyboard.sendKeyStroke(KEY_C, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "D"){  
    UsbKeyboard.sendKeyStroke(KEY_D, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "E"){  
    UsbKeyboard.sendKeyStroke(KEY_E, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "F"){  
    UsbKeyboard.sendKeyStroke(KEY_F, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "G"){  
    UsbKeyboard.sendKeyStroke(KEY_G, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "I"){  
    UsbKeyboard.sendKeyStroke(KEY_I, MOD_SHIFT_LEFT);  
}  
else if (Sc.substring(x, y) == "J"){  
    UsbKeyboard.sendKeyStroke(KEY_J, MOD_SHIFT_LEFT);  
}
```

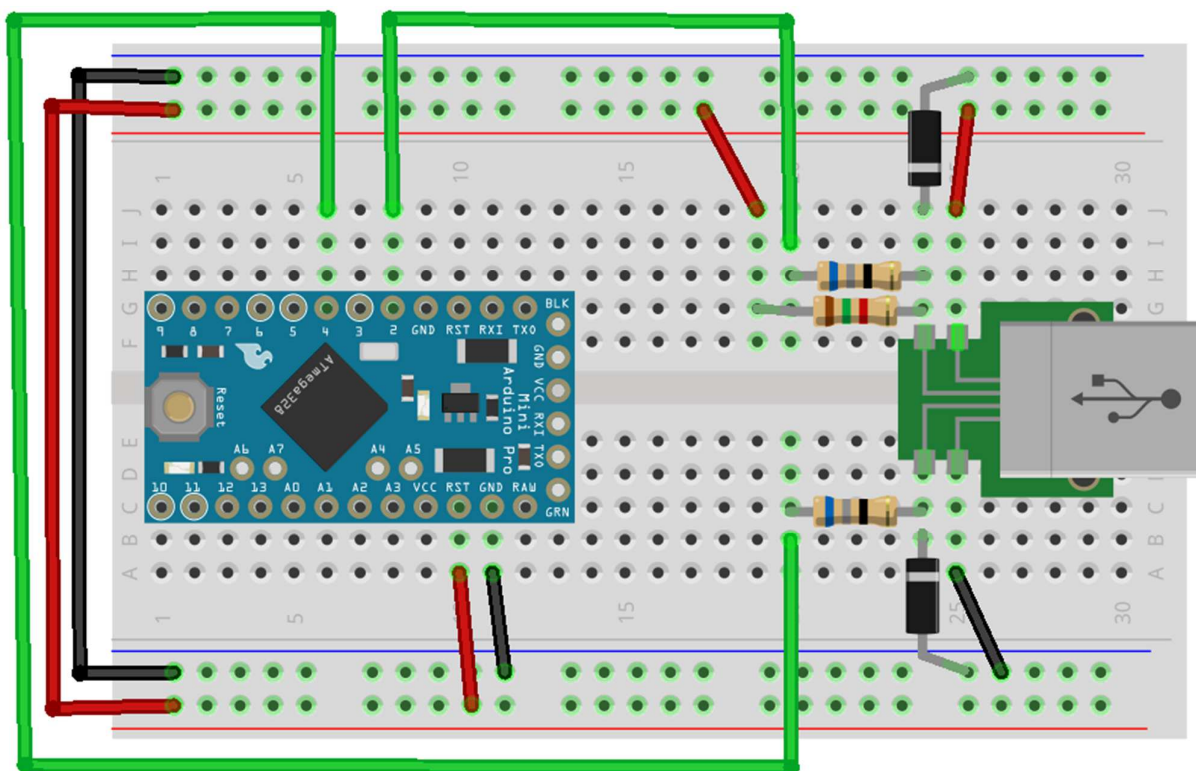
```
    }  
    else if (Sc.substring(x, y) == "K"){  
        UsbKeyboard.sendKeyStroke(KEY_K, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "M"){  
        UsbKeyboard.sendKeyStroke(KEY_M, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "N"){  
        UsbKeyboard.sendKeyStroke(KEY_N, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "O"){  
        UsbKeyboard.sendKeyStroke(KEY_O, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "P"){  
        UsbKeyboard.sendKeyStroke(KEY_P, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "Q"){  
        UsbKeyboard.sendKeyStroke(KEY_Q, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "R"){  
        UsbKeyboard.sendKeyStroke(KEY_R, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "S"){  
        UsbKeyboard.sendKeyStroke(KEY_S, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "T"){  
        UsbKeyboard.sendKeyStroke(KEY_T, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "U"){  
        UsbKeyboard.sendKeyStroke(KEY_U, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "V"){  
        UsbKeyboard.sendKeyStroke(KEY_V, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "W"){  
        UsbKeyboard.sendKeyStroke(KEY_W, MOD_SHIFT_LEFT);  
    }  
}
```



```
    }  
    else if (Sc.substring(x, y) == "X"){  
        UsbKeyboard.sendKeyStroke(KEY_X, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "Y"){  
        UsbKeyboard.sendKeyStroke(KEY_Y, MOD_SHIFT_LEFT);  
    }  
    else if (Sc.substring(x, y) == "Z"){  
        UsbKeyboard.sendKeyStroke(KEY_Z, MOD_SHIFT_LEFT);  
    }  
}  
UsbKeyboard.sendKeyStroke(KEY_ENTER);  
}
```

## Realizarea unui dispozitiv de tip mouse

De această dată vom realiza un sistem bazat pe placa de dezvoltare Arduino Pro Mini 5V (4) (pentru a reduce costurile și dimensiunea sistemului) plecând de la exemplul hid-mouse pus la dispoziție de biblioteca V-USB. Compilarea exemplului necesită instalarea mediului de dezvoltare WinAVR (12) iar încărcarea lui pe placa de dezvoltare necesită un programator ISP (de exemplu Arduino ISP (10) sau Pocket AVR(11)). Schema de interconectare a componentelor (placă de dezvoltare și componente de interfațare USB – similare cu proiectul precedent) este:



Pentru încărcarea programului se urmează următorii pași:

- Se instalează mediul de dezvoltare WinAVR (12).
- Se actualizează utilitarul avrdude (13) (avrdude.exe și avrdude.conf) din directorul de instalare al mediului WinAVR (uzual C:\WinAVR-20100110\bin) pentru a putea lucra cu microcontrolerul ATmega328P.
- Se conectează programatorul ISP pe pinii RST, MOSI (pinul 11), MISO (pinul 12), SCK (pinul 13), GND, 5V ai plăcii de dezvoltare Arduino Pro Mini.
- Se deschide o fereastră de comandă de tip consolă (click-dreapta cu tasta shift apăsată) pentru directorul hid-mouse (vusb-20121206\examples).

- În fereastra de comandă se tastează make hex pentru a compila exemplul și a obține fișierul binar hex ce urmează a fi încărcat în memoria microcontrolerului.
- În fereastra de comandă se tastează comanda  

```
avrdude -c usbtiny -p m328p -U hfuse:0xDE:m -U lfuse:w:0xFF:m
```

 pentru a inițializa biții de fuse (14) ai microcontrolerului; se consideră utilizarea unui programator USBtinyISP (precum Pocket AVR).
- În fereastra de comandă se tastează comanda  

```
avrdude -c usbtiny -p m328p -U flash:w:main.hex:i
```

 pentru a încărca programul în memoria internă a microcontrolerului.
- Se deconectează programatorul ISP și se conectează sistemul prin intermediul conexiunii USB.

Sistemul odată conectat se va comporta ca un dispozitiv de tip mouse (descriptor dispozitiv în imaginea următoare) – cursorul de mouse se va mișca pe ecran pe o traiectorie de tip cerc. Proiectul poate constitui un bun punct de plecare pentru dezvoltarea unor dispozitive de tip mouse – de exemplu se poate încerca dezvoltarea unui sistem care să miște cursorul de mouse cu ajutorul a patru butoane (sus, jos, dreapta, stânga).



## Referințe on-line

(1) V-USB - A Firmware-Only USB Driver for Atmel AVR Microcontrollers

<http://www.obdev.at/products/vusb/index.html>

(2) SoftwareSerial Library

<http://arduino.cc/en/Reference/softwareSerial>

(3) Pro Trinket - 5V 16MHz

[https://www.robofun.ro/pro-trinket-5v-16?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/pro-trinket-5v-16?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(4) Arduino Pro Mini 328 - 5V/16MHz

[https://www.robofun.ro/arduino\\_pro\\_mini\\_328\\_16mhz?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/arduino_pro_mini_328_16mhz?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(5) Introducing Pro Trinket - Using the USB Bootloader

<https://learn.adafruit.com/introducing-pro-trinket/starting-the-bootloader>

(6) vusb-for-arduino Port of the V-USB project to run on the Arduino platform

<https://code.google.com/p/vusb-for-arduino/>

(7) Projects Based on V-USB

<http://www.obdev.at/products/vusb/projects.html>

(8) USB Business Card

<http://eleccelerator.com/usb-business-card/>

(9) V-USB Driver API

<http://vusb.wikidot.com/driver-api>

(10) Arduino ISP

[https://www.robofun.ro/arduino-isp?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/arduino-isp?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(11) Programator Pocket AVR

[https://www.robofun.ro/platforme/avr/programator\\_pocket\\_avr?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/platforme/avr/programator_pocket_avr?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(12) WinAVR : AVR-GCC for Windows

<http://winavr.sourceforge.net/download.html>

(13) AVRDUDE - AVR Downloader/UploaDEr

<http://www.nongnu.org/avrdude/>

(14) Arduino / ATmega 328P fuse settings

<http://www.martyncurrey.com/arduino-atmega-328p-fuse-settings/>