

Pentru Incepatori.....	5
Ce este Arduino ?.....	5
Instalare Drivere Arduino.....	7
Arduino IDE, Program Arduino, Upload.....	8
Anatomia unui program Arduino.....	9
Arduino Blink.....	10
Led Brick, ON/OFF.....	11
Led Brick, intensitate variabila.....	13
Breadboard.....	16
Debug Serial.....	17
Buton Brick.....	19
Senzor Lumina Brick.....	21
Mini Difuzor.....	23
Senzor Temperatura Brick.....	25
Senzor Umiditate Brick.....	27
Despre Shield-uri Arduino.....	29
Modalitati de Comunicare Intre Arduino si Componente.....	30
Citire Date pe Porturile Analogice sau Digitale.....	30
Comunicare Seriala.....	31
Software Serial.....	32
SPI.....	33
I2C.....	35
Senzori Acceleratie.....	36
ADXL335.....	38
ADXL345.....	41
BMA180.....	43
MMA8452Q.....	46
Senzori Biometrici.....	48
Alcool – MQ3.....	48
Senzori de Curent.....	49

Senzori de Forta.....	50
Senzori de Apasare.....	50
Senzor de Atingere HotPot.....	52
Senzor de Indoire.....	54
Senzor Piezo.....	56
Giroscoape.....	58
Inertial Measurement Unit.....	59
6 Grade de Libertate :	61
ITG-3200 + ADXL345.....	61
9 Grade de Libertate : 9DOF ITG-3200 + ADXL345 + HMC5843.....	62
Senzori de Distanta.....	63
Sharp 3-40 cm, Sharp 10-80 cm si Sharp 15-150 cm.....	64
Sharp digital 10 cm, Sharp digital 5 cm.....	68
Senzor de Linie Analogic.....	69
Senzori Prezenta Umana.....	71
Senzor PIR.....	71
Sunet.....	75
Microfon.....	75
Microfon cu Breakout Board.....	76
Presiune Atmosferica – BMP085.....	78
Presiune Atmosferica – MPL115A1	81
Temperatura si Umiditate – SHT15.....	82
Temperatura si Umiditate – DHT22.....	83
Temperatura – TMP102.....	86
Temperatura la Distanta – MLX90614.....	90
Shield LCD 16X2.....	93
WiFly Shield.....	94
WiFly Shield – Client Web.....	94
WiFly Shield – Server Web.....	96
Ethernet Shield.....	98
Client Web, IP prin DHCP.....	99

Client Web, IP static.....	104
Server Web.....	106
Client Web, cu DNS.....	110
JoyStick Shield.....	111
MP3 Player Shield.....	113
MP3 Trigger.....	115
Music Instrument Shield.....	119
Kit ProtoShield.....	129
Kit ProtoShield Mega.....	131
USB Host Shield.....	132
LCD.....	135
LCD 16 X 2, LCD 16 X 4, LCD 20 X 4.....	135
LCD 16 X 2 pe I2C, LCD 20 X 4 pe I2C.....	140
Servomotoare.....	150
Servomotoare cu rotatie continua.....	154
Elemente de electricitate si dispozitive electrice simple.....	154
Analogie electricitate – curgerea fluidelor.....	154
Divizorul de tensiune.....	157
Circuite electrice cu bifurcatii.....	158
Condensatorul.....	159
Semnalul PWM.....	160
Proiecte cu Arduino.....	162
Semafor, Trei Led-uri Brick.....	162
Lampa de veghe.....	163
Buton + LED.....	164
In ritmul muzicii : Microfon + Led.....	165
Termometru : Senzor Temperatura Brick + LED-uri.....	168
Stocarea online a datelor culese de Arduino, folosind Google Docs.....	171
Arduino Leonardo pe post de tastatura.....	177
Arduino Leonardo pe post de mouse.....	180
Logare presiune atmosferica, umiditate, nivel de iluminare si	

temperatura in Excel, cu Arduino Leonardo.....	181
Mouse cu accelerometru si Arduino Leonardo.....	185
Ultimul tweet al unui user Twitter afisat pe LCD Shield, folosind Ethernet Shield (sau Arduino Ethernet).....	195
Harpa cu laser.....	202
Toba cu senzori piezo.....	209
Cum sa faci o floare sa te traga de maneca atunci cand uiti sa o uzi.....	219
Accelerometru pentru automobil.....	223

Pentru Incepatori

Ce este Arduino ?

Arduino este una dintre cele mai simple de utilizat platforme cu microcontroller. Te poti gandii la el ca la un minicalculator (are puterea de calcul a unui computer obisnuit de acum 15 ani), fiind capabil sa culeaga informatii din mediu si sa reactioneze la acestea.

In jurul lui Arduino exista un ecosistem de dispozitive extrem de bine dezvoltat. Orice fel de informatie ti-ai dori sa culegi din mediu, orice fel de conexiuni cu alte sisteme ai avea nevoie, exista o sansa foarte mare sa gasesti un dispozitiv pentru Arduino capabil sa iti ofere ceea ce ai nevoie.

Astfel, daca discutam despre preluarea de informatii din mediu, mai jos sunt doar cateva exemple de senzori : senzori ce determina nivelul de alcool in aerul respirat, senzor de incendiu, gaz GPL, monoxid de carbon, acceleratii ale dispozitivelor in miscare, curent consumat de diverse dispozitive casnice, forta de apasare, gradul de rotire, cartele RFID, distante, nivel de iluminare, directia nordului, prezenta umana, sunet, temperatura, umiditate, presiune atmosferica sau video. Daca ne referim la posibilitatea de a ne conecta cu alte sisteme, exista placi de retea Ethernet pentru Arduino capabile sa comunice informatii prin Internet, dispozitive capabile sa transmita date prin conexiune radio, placi de retea WIFI, dispozitive GSM pentru Arduino (capabile sa trimita / receptioneze SMS-uri, sa initieze apeluri de voce sau sa trimita date prin reseaua 3G) sau conectori Bluetooth pentru conectarea Arduino cu telefonul mobil sau laptop. In zona mecanica, exista motoare de curent continuu (utilizate pentru

robotica), motoare pas cu pas (utilizate de obicei in zona industrială) sau servomotoare, controlate foarte exact. Pentru afisarea informatiilor preluate, exista ecrane LCD pentru Arduino, incepand cu cele mai simple (LCD text cu 16 caractere) pana la ecran LCD grafice. In prima parte a acestei carti vom dezvolta integrarea Arduino cu fiecare dintre dispozitivele pe care le-am amintit mai sus (si alte cateva in plus). In cea de-a doua parte vom prezenta pe larg o serie de proiecte interesante realizate prin combinarea Arduino cu unul sau mai multe dispozitive dintre cele enumerate mai devreme.

Pentru a-ti face o idee despre ceea ce se poate face cu Arduino, in continuare cateva exemple de proiecte care se pot realiza cu Arduino (proiecte pe care le vom detalia pe larg in cea de-a doua parte a cartii) :

–senzor de alcool conectat la Arduino, Arduino conectat la PC; atunci cand nivelul de alcool in aerul expirat depaseste un anumit nivel prestabilit, computerul iti da un warning cand incerci sa postezi lucruri pe Facebook.

–robot autonom care ocoleste obstacole.

–robot controlat prin Bluetooth folosind telefonul mobil sau laptop-ul

–dispozitiv pentru pictat oua ("the EggBot").

–Arduino + senzor temperatura + senzor umiditate + senzor presiune atmosferica + placa de retea Ethernet care transmite datele de mediu pe Google Docs, la fiecare 10 secunde.

–mana robotica, bazata pe o manusa cu senzori de indoire si servomotoare.

–Quadcopter-e.

–OpenEnergyMonitor, sistem bazat pe Arduino care monitorizeaza energia electrica consumata in casa.

–prastie reala (din lemn) combinata cu Arduino, un accelerometru si Bluetooth, pentru jucat Angry Birds pe PC.

–acces bazat pe cartele RFID + notificari prin Twitter.

In cazul Arduino, totul este foarte simplu de utilizat. Ai nevoie de circa 5 minute ca sa iti instalezi mediul de dezvoltare si sa scrii primul tau program pe Arduino. Nu este necesar sa lipesti fire, tot ce ai nevoie este un port USB liber.

Instalare Drivere Arduino

Primul lucru pe care trebuie sa il faci inainte de a putea programa placa Arduino, este sa ii instalezi driver-ele pe PC. Daca rulezi Linux sau MAC, ai scapat, nu ai nevoie de driver-e. Daca insa rulezi Window, va trebui instalezi si drivere-le, ca pentru orice alt dispozitiv hardware. Driver-ele sunt disponibile in mediul de dezvoltare, pe care il vei descarca de la adresa <http://arduino.cc/en/Main/Software> . Descarca arhiva .zip, si dezarchiveaz-o intr-o director la alegerea ta. Conecteaza apoi placa Arduino la portul USB. PC-ul va detecta ca ai un nou dispozitiv, si iti va cere sa instalezi drivere pentru el. Alege optiunea de instalare prin care ii spui Windows ca ai tu drivere, si ii ceri sa nu se conecteze la Internet pentru a cauta in mod automat. Selecteaza directorul "drivers" din directorul unde ai dezarhivat arhiva .zip de la pasul precedent (NU directorul "FTDI Drivers" !). Procesul continua apoi cu "Next". In final, daca

deschizi Device Manager, in sectiunea "Ports (COM & LPT)" ar trebui sa vezi o intrare numita "Arduino UNO".

Aici - <http://arduino.cc/en/Guide/Windows> - gasesti un ghid mai detaliat, cu screenshot-uri referitor la instalarea driver-elor pentru Arduino.

Arduino IDE, Program Arduino, Upload

Dupa ce ai instalat drivere-ele, urmatorul pas este sa iti programezi placa Arduino. Ruleaza programul "arduino", din mediul de dezvoltare pe care tocmai l-ai descarcat la pasul precedent (vezi screenshot-ul de mai jos). Arduino se conecteaza la PC prin intermediul unui port serial. Primul pas pe care il ai de facut este sa determini acest port. Cea mai simpla cale este de conectezi placa, astepti circa 30 de secunde – 1 minut, ca sa fii sigur ca a fost detectata de PC, iar apoi deschizi meniul "Tools -> Serial Port". Ar trebui sa vezi una sau mai multe intrari. Memoreaza-le (sau scrie-le pe o foaie de hartie / fa un screenshot). Deconecteaza placa Arduino din portul USB (scoate cablul din PC). Deschide din nou meniul "Tools -> Serial Port". Acel port care a disparut este portul asociat placii Arduino. Conecteaza din nou placa Arduino la PC, asteapta sa fie recunoscuta de PC, si apoi selecteaza portul respectiv din meniul "Tools -> Serial Port".

Urmatorul pas este selectarea tipului de placa cu care lucrezi. Din meniul "Tools -> Board", selecteaza tipul de placa cu care lucrezi (Arduino Uno, Leonardo, Mega, etc).



Anatomia unui program Arduino

Orice program Arduino are doua sectiuni. Sectiunea "setup", care este rulata doar o singura data, atunci cand placa este alimentata (sau este apasat butonul "Reset"), si sectiunea "loop", care este rulata in ciclu, atat timp cat este alimentata placa. Sa luam un exemplu.

```
void setup() {
    //codul scris aici ruleaza o singura data
}
void loop() {
    //codul scris aici ruleaza tot timpul
}
```

Astfel, in rutina "setup" vom pune de obicei cod de initializare, iar in rutina "loop" vom scrie partea principala a programului nostru. Este interesant sa dai acum cateva pagini mai departe si sa urmaresti acest lucru pe exemplele din capitolele urmatoare.

Arduino Blink

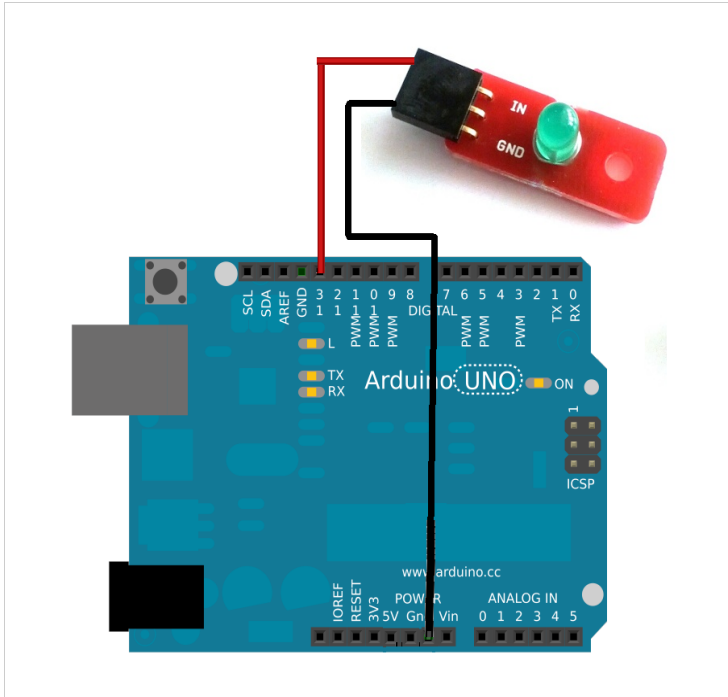
Pentru cel de-al doilea exemplu Arduino, vom folosi un led montat din fabricatie pe placa. Placa Arduino are din constructie un led conectat la pinul digital 13. Acest led se aprinde atunci cand pinul 13 digital este pus in HIGH (din programul scris pe Arduino) se stinge atunci cand pinul 13 este pus in LOW.

```
void setup() {
    pinMode(13, OUTPUT);
}
void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Daca urci acest program pe placa Arduino si urmaresti led-ul conectat la pinul 13 vei observa ca el clipeste o data pe secunda. Sa analizam un pic programul de mai sus. In sectiunea "setup" avem o singura instructiune, care declara ca pinul 13 digital va fi folosit in continuare pentru a controla un dispozitiv (daca am fi vrut sa citim informatie din mediul – sa atasam un senzor la el – am fi scris "pinMode(13, INPUT)"). Aceasta instructiune se executa o singura data. Urmeaza sectiunea "loop", care aprinde led-ul conectat la pinul 13 ("digitalWrite(13, HIGH)"), asteapta o secunda, il stinge si iarasi asteapta o secunda. Asa cum am spus mai sus, instructiunile din sectiunea "loop" se executa ciclic, astfel ca imediat dupa ce s-a terminat un ciclu, led-ul iarasi se aprinde, apoi iarasi se stinge si tot asa pana cand taiem alimentarea placii. Exemplul urmat este extrem de similar cu acesta, doar ca in locul led-ului montat din fabrica pe placa Arduino vom utiliza un led fizic conectat la pinul 13 al placii.

Led Brick, ON/OFF

Un led brick reprezinta cea mai simpla posibilitate de a genera un semnal luminos folosind Arduino. Un led brick necesita un port digital liber pentru a fi controlat de catre Arduino.



Arduino GND	GND Led
Arduino Digital 13	IN Led

```

void setup() {
    pinMode(13, OUTPUT);
}
void loop() {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}

```

Rutina *setup*, care se executa exact o singura data cand Arduino este alimentat, declara pinul digital 13 (cel la care am conectat led-ul) ca fiind un pin *de iesire* (in sensul ca va controla un dispozitiv extern conectat la Arduino, si nu va citi o informatie digitala din mediu).

Rutina *loop*, care se executa atata timp cat Arduino este alimentat, aprinde led-ul ridicand pinul 13 in HIGH (mai exact, face ca tensiunea pinului sa fie 5 V), apoi asteapta o secunda, stinge led-ul (tensiunea pinului devine 0 V) si apoi iarasi asteapta o secunda.

De ce nu merge ?

—sigur ai conectat corect firele ?

—sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done Uploading") ?

—daca scoti firul de conectare din pinul 13 (marcat pe led cu "IN") si il muti in pinul VCC, se aprinde led-ul ? (daca nu, atunci led-ul probabil este defect)

Led Brick, intensitate variabila

In exemplul precedent am vazut cum putem aprinde si stinge un led. In acest exemplu vom folosi acelasi le, dar il vom aprinde variabil. Chiar daca Arduino nu poate scoate tensiune variabila pe porturile digitale (scoate ori 5V ori 0V), exista o

posibilitate de a genera un semnal de putere variabila pe unul dintre porturile sale. Acest lucru este posibil prin generarea unui semnal dreptunghiular, care se plimba periodic intre 0V si 5V, foarte rapid. In functie de cat timp sta in 5V si cat timp sta in 0V, puterea semnalului variaza. Numele acestui gen de semnal este "PWM". Daca vrei sa citesti mai multe despre semnalul PWM, vezi si capitolul "Analogie electricitate - curgerea fluidelor".

Este interesant de remarcat faptul ca doar 6 din cei 13 pini ai Arduino UNO sunt capabili sa genereze semnal PWM (pinii 3, 5, 6, 9, 10 si 11).

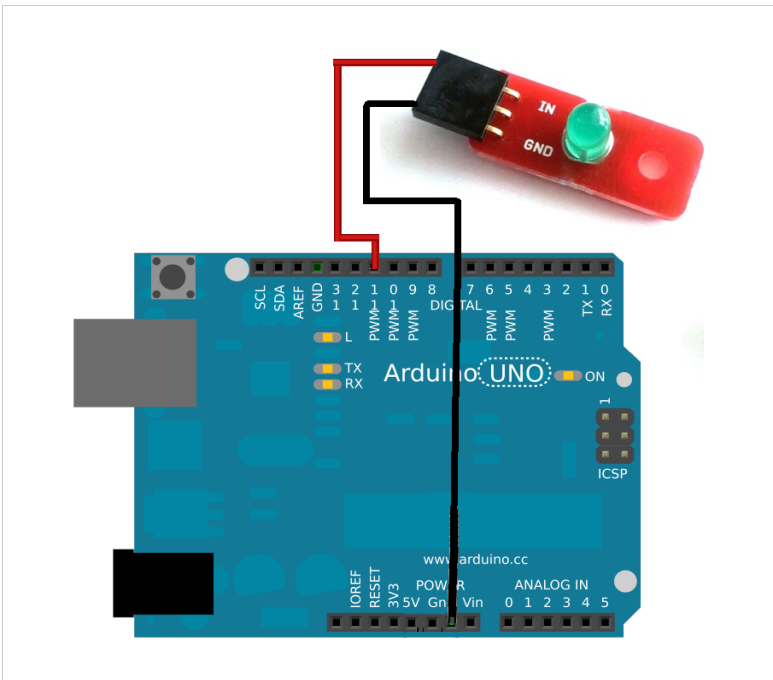
Pentru cazul Arduino Mega, avem 15 pini PWM (de la 2 la 13 si de la 44 la 46).

```
void setup() {
    pinMode(11, OUTPUT);
}
void loop() {
    for (int i = 0; i < 255; i++){
        analogWrite(11, i);
        delay(50);
    }
    for (int i = 255; i > 0; i--){
        analogWrite(11, i);
        delay(50);
    }
}
```

Rutina *setup*, care se executa exact o singura data cand Arduino este alimentat, declara pinul digital 11 (cel la care am conectat led-ul) ca fiind un pin *de iesire*.

In rutina *loop* este interesanta instructiunea *analogWrite*, care defineste puterea semnalului PWM de iesire. Ca parametri,

instrucțiunea *analogWrite* primește pinul (11, în cazul nostru), și puterea semnalului (variabilă, de la 0 la 255). Aceasta instrucțiune este apelată într-un ciclu *for*, care modifică valoarea variabilei *i* între 0 și 255. Efectul va fi ca led-ul să se aprindă gradat până la maxim, iar apoi să se stingă treptat.



Arduino GND	GND Led
Arduino Digital 11	IN Led

De ce nu merge ?

–sigur ai conectat corect firele ?

–sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done Uploading" ?)

–daca scoti firul de conectare din pinul 13 (marcat pe led cu "IN") si il muti in pinul VCC, se aprinde led-ul ? (daca nu, atunci led-ul probabil este defect)

Breadboard

Un breadboard este un dispozitiv care permite conectarea extrem de simpla a componentelor electronice, fara lipituri. Pentru a conecta dispozitivele se folosesc fire tata-tata (cu pini la ambele capete), care se introduc in gaurile din breadboard.

Gaurile existente in breadboard sunt conectate intre ele (de obicei pe linie), astfel incat firele introduse pe aceeasi linie vor fi conectate intre ele. In imaginea de mai jos am marcat cu linie neagra pinii conectati intre ei (eu am marcat doar pentru primele 3 coloane, dar toate liniile breadboard-ului sunt conectate intre ele). Un exemplu frecvent de utilizare a breadboard-ului este acela in care dorim sa conectam simultan mai multe dispozitive brick la Arduino (care are doar un singur pin de 5V, si doar 3 pini de GND). In aceasta situatie, vom conecta folosind fire tata-tata pinul de 5V Arduino la una dintre liniile breadboard-ului, la alta linie din breadboard vom conecta unul dintre pinii GND, si in acest mod vom avea disponibile inca patru pini care sunt conectati la 5V (cei care sunt pe aceeasi linie cu cel conectat la 5V), si patru pini conectati la GND (cei care sunt pe aceeasi linie cu GND). Vei folosi acest lucru aproape pentru fiecare dintre exemplele si proiectele din aceasta carte. Pentru toate proiectele

care urmeaza am considerat ca folosirea unui breadboard se subintelege peste tot pe unde apar doar fire legate impreuna in schema.

Exista multe tipuri de breadboard, mai mari sau mai mici. Unul dintre cele mai mici breadboard-uri este cel de aici - http://www.robofun.ro/breadboard/breadboard_mini , care este suficient pentru situatia in care vrei sa alimentezi mai multe dispozitive folosind acelasi Arduino. Un breadboard ceva mai mare (necesar pentru atunci cand vrei sa mai adaugi si alte componente pe breadboard, in afara de componente brick) este acesta – <http://www.robofun.ro/breadboard/breadboard-82x52x10> . Evident, daca si acesta este prea mic pentru ce ai nevoie, poti oricand inlantui doua sau mai multe breadboard-uri intre ele, cu fire.

Debug Serial

Asa cum spuneam intr-o sectiune precedenta, o data ce ai urcat programul pe Arduino, acesta ruleaza pe procesorul Arduino, si nu pe PC. La fel de bine poti deconecta complet Arduino de la calculator si sa il alimentezi cu o baterie, programul va continua sa ruleze. Sunt situatii (si nu putine!) cand rezultatele rularii programului sunt cu totul altele decat iti doresti tu, si atunci iti doresti sa ai acces in interiorul Arduino ca sa poti vedea ce se intampla acolo. Din fericire, exista si o alta solutie, ceva mai simpla. Cablul USB de conectare la calculator, pe langa alimentarea Arduino, poate transmite si date catre PC sau de la PC catre Arduino. Acest lucru este extrem de util pentru a vizualiza pe PC valorile din programul care ruleaza pe Arduino.

De exemplu, sa spunem ca avem un senzor de lumina conectat la Arduino si vrem sa aprindem un led atunci cand

nivelul de iluminare scade sub o anumita valoare. Am scris programul, l-am urcat pe Arduino, dar cand testam, lucrurile nu functioneaza corect. Ne-am dori sa vedem ce valoare citeste senzorul de lumina, ca sa vedem daca pragul setat de noi in program este corect. Vom face acest lucru trimitand prin cablul USB valoarea citita de senzorul de lumina si vizualizand aceasta valoare pe PC.

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    int lumina = analogRead(0);  
    Serial.println(lumina);  
    delay(10);  
}
```

Pentru a rula codul sursa de mai sus nu este necesar sa conectezi un senzor de lumina. Poti sa urci pur si simplu programul de mai sus pe Arduino si apoi sa deschizi Serial Monitor (din meniul "Tools", alegi optiunea "Serial Monitor"). Vei vedea o serie de valori aleatoare afisate pe ecran (citiri ale portului analogic 0 al Arduino, la care nu este conectat nimic).

Instructiunile interesante din programul de mai sus sunt "Serial.begin(9600)", care initializeaza o comunicare seriala intre Arduino si PC cu viteza de 9600 de biti pe secunda si "Serial.println(lumina)", care trimite valoarea variabilei "lumina" catre PC.

Poti utiliza aceasta metoda ori de cate ori vrei sa vezi ce valori au variabilele din programul tau Arduino.

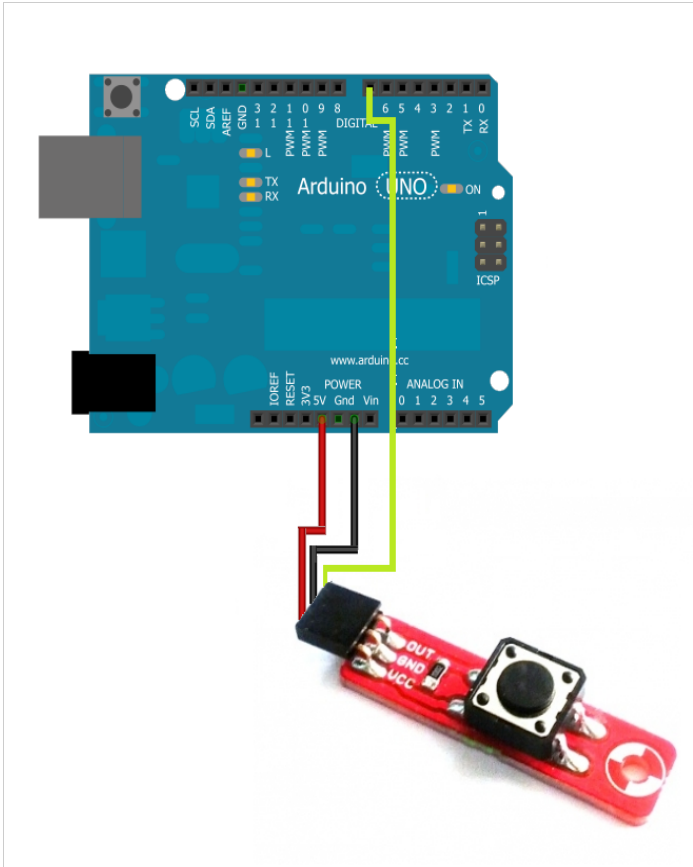
De ce nu merge ?

–daca in loc sa vezi valori numerice in Serial Monitor, primesti o eroare atunci cand alegi optiunea, atunci cel mai probabil portul serial selectat de tine nu este cel pe care este conectat Arduino; mergi in meniul "Tools" -> "Serial Port" si alege o alta optiune. Daca nu ai nici o idee care este optiunea corecta, atunci deconecteaza Arduino de PC si intra din noul in "Tools "-> "Serial Port". Portul care a disparut este acel port pe care era conectat Arduino. Reconecteaza acum Arduino, si selecteaza-l pe acesta.

– daca in loc sa vezi valori numerice in Serial Monitor vezi o serie de caractere ciudate care se schimba continuu, inseamna ca rata de transfer selectata in Serial Monitor nu este aceeaasi cu rata de transfer selectata in codul sursa Arduino. Verifica in dreapta jos a ferestrei Serial Monitor ca valoarea selectata sa fie 9600 (aceeasi pe care am selectat-o in functia "setup" din codul Arduino – "Serial.begin(9600);").

Buton Brick

Am vazut in exemplele precedente cum putem folosi porturile digitale Arduino pentru a comanda dispozitive din exterior (led-uri, in exemplele de pana acum). Acum vom vedea cum putem folosi un port digital Arduino pentru a citi informatie din mediu (starea unui buton).



Arduino GND	GND Buton
Arduino 5V	VCC Buton
Arduino Digital 7	OUT Buton

```
void setup() {
    pinMode(7, INPUT);
    Serial.begin(9600);
}
void loop() {
    int stareButon = digitalRead(7);
    Serial.println(stareButon);
    delay(10);
}
```

Primul lucru interesant este faptul ca acum pinul digital 7 (cel la care am conectat un buton) este setat in mod INPUT (spre deosebire de exemplele precedente, unde era de tip OUTPUT). Asta pentru ca urmeaza sa il folosim ca sa citim informatie din mediu.

Rutina loop citeste starea butonului (care poate fi 0 sau 1 – apasat sau destins) si afiseaza aceasta stare in consola seriala. Poti vedea aceasta informatie deschizand Serial Monitor in Arduino IDE.

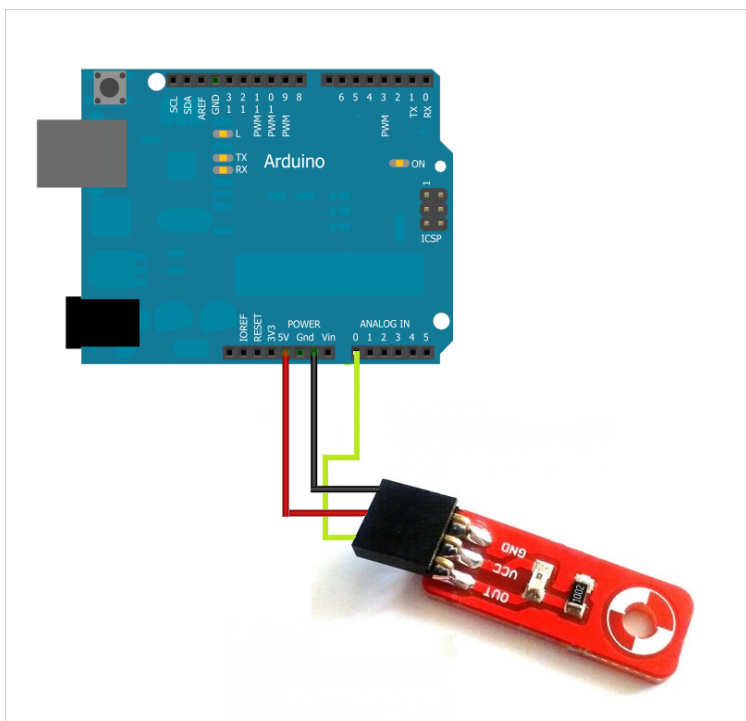
De ce nu merge ?

–sigur ai conectat corect firele ?

–sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done Uploading") ?

Senzor Lumina Brick

Dupa ce am comandat led-uri si am citit valori digitale din mediu, senzorul de lumina este primul exemplu de citire a valorilor analogice din mediu. Un senzor de lumina da o valoare numerica intre 0 si 1023, valoare proportionala cu nivelul de iluminare din mediul ambiant.



Arduino GND	GND Senzor
Arduino 5V	VCC Senzor
Arduino Analog 0	OUT Senzor

```
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    int nivelIluminare = analogRead(0);  
    Serial.println(nivelIluminare);  
    delay(10);  
}
```

In rutina *setup* singurul lucru pe care il facem este sa initializam o comunicare seriala cu PC-ul, pe care o vom folosi ca sa transmitem si sa vizualizam pe PC valorile citite de senzorul de lumina.

Rutina *loop* citeste valoarea data de senzorul de lumina (conectat la portul serial 0) si afiseaza aceasta valoare in consola seriala. Poti vedea aceasta informatie deschizand Serial Monitor in Arduino IDE.

Pentru a testa ca lucrurile functioneaza corect, pune degetul peste senzorul de lumina. Vei observa ca valoarea pe care o vezi in Serial Monitor scade.

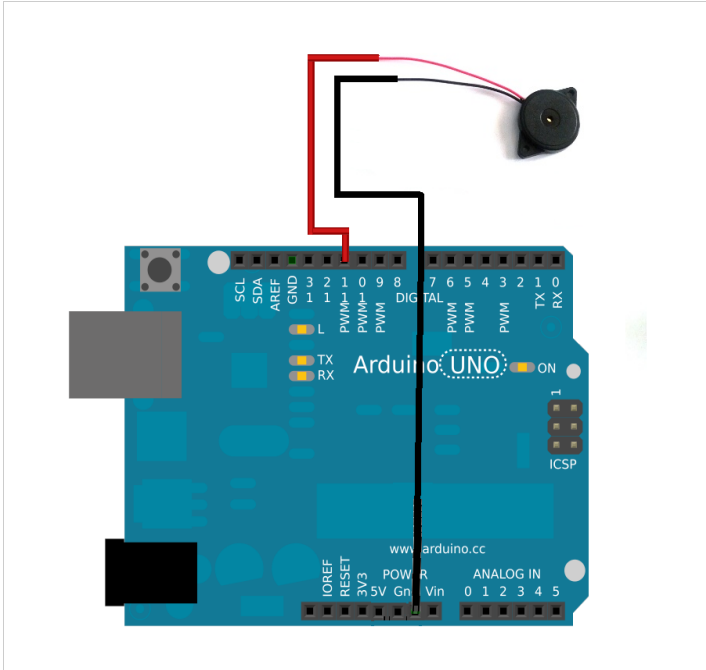
De ce nu merge ?

–sigur ai conectat corect firele ?

–sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done Uploading") ?

Mini Difuzor

Mini difuzorul este cea mai simpla modalitate de a crea sunete cu Arduino. Utilizarea acestuia este extrem de simpla. Conectezi firul negru la pinul GND al placii Arduino, iar firul rosu la un pin PWM digital al placii Arduino. Placa Arduino UNO are 6 astfel de pini (3, 5, 6, 9, 10 si 11), iar placa Arduino Mega are 15 astfel de pini (de la pinul 2 pana la pinul 13 si de la pinul 44 la pinul 46). Un semnal de tip PWM este un semnal care se misca foarte rapid intre 5V si 0V, astfel incat membrana difuzorului este si ea miscata la fel de rapid, generand sunete. Frecventa cu care semnalul se misca intre 5V si 0V determina frecventa sunetului. Daca vrei sa intelegi mai multe despre semnalul PWM, vezi si capitolul "Analogie electricitate - curgerea fluidelor".



Arduino GND	Fir Negru Difuzor
Arduino Digital 11	Fir Rosu Difuzor

```

void setup() {
    pinMode(11, OUTPUT);
}

void loop() {
    for (int i = 1500; i < 4000; i++) {
        tone(11, i);
        delay(10);
    }
}

```

```

    for (int i = 4000; i > 1500; i--) {
        tone(11, i);
        delay(10);
    }
}

```

Partea interesanta din codul de mai sus este instructiunea *tone*, care primeste ca parametri pinul la care este conectat difuzorul (in cazul nostru pinul 11) si frecventa sunetului (in cazul nostru, variabila *i*). Variabila *i* se modifica intre 1500 de Hertzi si 4000 de Hertzi. Efectul obtinut este cel de sirena. Ca sa opresti sunetul complet, instructiunea este *noTone(<pin>)*; In cazul de mai sus, *noTone(11)* opreste complet sunetul.

Senzor Temperatura Brick

Senzorul de temperatura brick este un alt exemplu de senzor care ofera valori analogice care depind de temperatura din mediul ambiant. Din valorile citite de la senzori se obtine valoarea temperaturii in mediul ambiant in grade Celsius, aplicand o formula matematica simpla, formula prezentata in codul sursa de mai jos.

```

void setup() {
    Serial.begin(9600);
}

void loop() {
    Serial.println(citesteTempInCelsius());
}

```

```

}

float citesteTempInCelsius() {
    float temperaturaMediata = 0;
    float sumaTemperatura;
    for (int i =0; i<10; i++) {
        int reading = analogRead(0);
        float voltage = reading * 5.0;
        voltage /= 1024.0;
        float temperatureCelsius = (voltage - 0.5)
* 100 ;
        sumaTemperatura = sumaTemperatura +
temperatureCelsius;
    }
    return sumaTemperatura / (float)count;
}

```

La fel ca in exemplele precedente, in rutina *setup* singurul lucru pe care il facem este sa initializam o comunicare seriala cu PC-ul, pe care o vom folosi ca sa transmitem si sa vizualizam pe PC valorile citite de senzorul de temperatura.

Rutina loop nu face altceva decat sa apeleze rutina "citesteTempInCelsius" care calculeaza temperatura in grade Celsius pe baza valorii citite de la senzor. Pentru a diminua influenta surselor de erori asupra citirilor, temperatura se calculeaza pe baza a zece citiri succesive, care sunt mediate.

Ca sa testezi ca lucrurile functioneaza corect, pune degetul peste senzorul de temperatura. Vei observa ca valoarea pe care o vezi in Serial Monitor creste.

De ce nu merge ?

–sigur ai conectat corect firele ?

–sigur programul s-a incarcat pe Arduino (ai vazut mesajul "Done

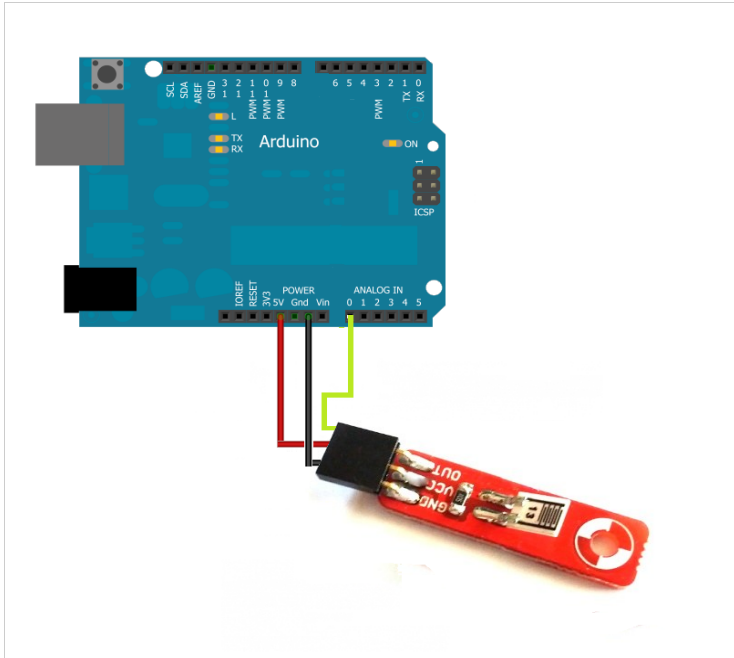
Uploading") ?

Senzor Umiditate Brick

Senzorul de umiditate brick este un senzor care ofera o valoare analogica care depinde de nivelul de umiditate din mediul ambiant. Din valoarea citita de la senzor se poate deduce valoarea exacta a umiditatii (%), dar deducerea formulei matematice este destul de dificila, asa ca in cele ce urmeaza ne vom multumi sa obtinem un nivel calitativ al nivelului umiditatii ("mai umed", "mai putin umed"). Mai exact vom obtine o valoare care variaza in functie de umiditate exact in acelasi mod tot timpul. Daca astazi valoarea citita pe senzor este 453, si ieri a fost tot 453, atunci vom putea spune "astazi umiditatea din aer are acelasi nivel ca si ieri".

Ca o paranteza, daca ai nevoie de un senzor foarte precis, si care iti ofera direct nivelul umiditatii din aer, in unitati standard, atunci iti sugerez STH15, un senzor etalonat si extrem de precis -

http://www.robofun.ro/senzor_temperatura_umiditate_sht15.



Arduino GND	GND Senzor
Arduino 5V	VCC Senzor
Arduino Analog 0	OUT Senzor

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int nivelUmiditate = analogRead(0);  
  Serial.println(nivelUmiditate);  
  delay(50);  
}
```

}

Codul sursa de mai sus este aproape identic cu cel de la senzorul de lumina, asa ca nu voi insista asupra lui.

Ca sa testezi ca lucrurile functioneaza corect, sufla peste senzor, usor. Vei observa ca valoarea pe care o vezi in Serial Monitor se modifica, datorita faptului ca respiratia ta contine vapori de apa. Alta varianta este sa desfaci o sticla de plastic care contine apa pe jumata, si sa introduci senzorul in prima jumatate a sticlei (NU in apa, in zona in care sticla este goala).