Activitatiele tehnice principale desfasurate in cadrul contractului si anexei

a) Activitatea de proiectare electronica

Activiatea de proiectare electronica consta in realizarea de scheme electronice pentru produsele solicitate de client asa cum se prevede in contract.

Proiectarea circuitelor electronice este procesul de creare a circuitelor care constau din componente pasive și active, iar asamblarea acestora creează o cale astfel încât curentul electric să îndeplinească funcții specifice. Atunci când se proiectează un astfel de circuit pe calculator, circuitul electronic este reprezentat printr-o diagramă schematică. Fiecare element fizic de circuit este identificat printr-un simbol grafic corespunzător și prin informații despre parametrii săi. Procesul de proiectare electronică pe calculator oferă posibilitatea de a edita ulterior componentele selectate și întregul circuit.

Din punct de vedere informatic, aceasta este o activitate foarte dificilă și care necesită multă muncă, astfel încât au început să fie dezvoltate instrumente care să sprijine procesul de proiectare a circuitului și instrumente care să permită verificarea performanțelor și a punctelor slabe ale circuitului general în etapa de preimplementare. Acest lucru a dus la dezvoltarea simulatoarelor de circuite electronice. Acestea permit, pe baza observațiilor, să se tragă concluzii cu privire la comportamentul obiectelor fizice în anumite condiții (în cazul simulatoarelor bune, pot fi modificate condiții cum ar fi temperatura mediului ambiant) și să se facă corecții ale sistemului înainte ca acesta să fie creat fizic.

Simularea circuitelor oferă o perspectivă critică asupra comportamentului circuitelor electronice. Având în vedere costul și timpul implicat în fabricarea circuitelor electronice, în special a circuitelor integrate, este mult mai practic să se verifice comportamentul și performanța circuitului folosind simularea circuitului înainte de începerea producției.

Realizarea schemelor electronice a fost realizata folosind software recunoscut industrial conform normelor in vigoare.

Proiectarea electronica s-a realizat urmarind specificatiile de produs ale clientului urmarind atingerea specificatiilor de produs. S-au folosit tehnici specifice de proiectare pentru curenti mici, curenti mari, tensiuni mari si semnale de radio frecventa.

Toate componentele folosite sunt cu origini verificate pentu a asigura reproductibilitea produsului pe un termen de cel putin 3 ani. Au fost realizate cel putin 5 produse iar in anumite cazuri au fost realizate si varinte derivate din produsul initial.

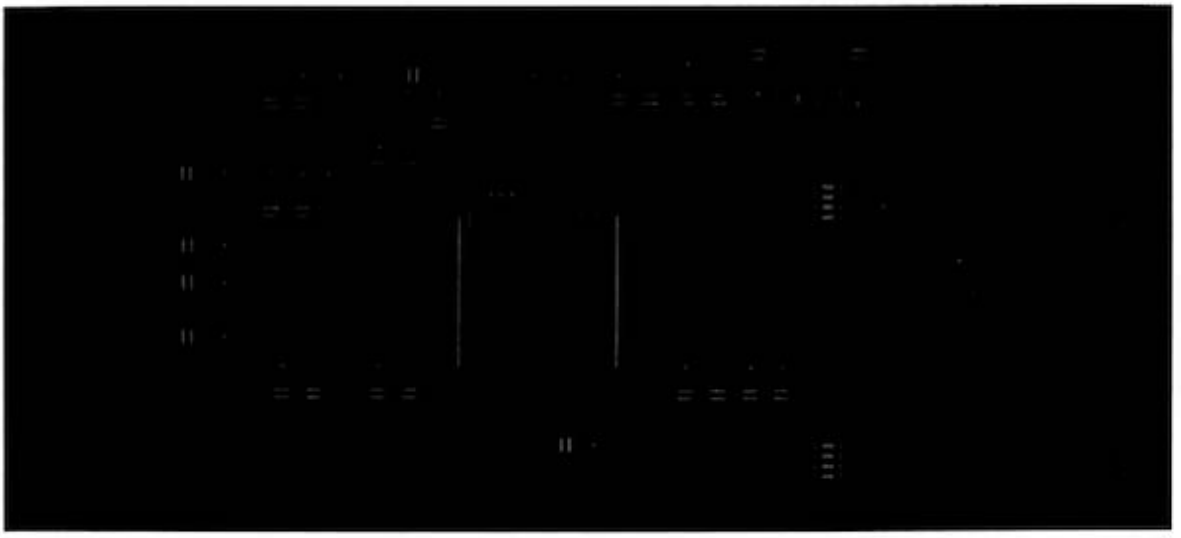
b) Activitatea de cercetare fundamentala in electronica

Cercetarea fundamentala a fost realizata in urma constatarii nevoii de implementare a unor tehnologii noi pentru produsele realizate.

Aceasta se refera la studiul unor componete electronice avansate d.p.d.v tehnologic cum ar fi microprocesoarele de ultima generatie ARM si componentele asociate lor.

O alta ramura a acestei activitati este cea de cercetare a componentelor electronice pentru radio freventa (RF),masuratori specifice de precizie pentru integrarea acestora in produse dar si evaluarea unor module care folosesc diverse tehnologii radio deja integrate cum ar fi modulele GPS , GSM si LTE.

Dispozitivele electronice supuse cercetarii fac parte din toata gama de componenete electronice de mica si mare putere folosite la crearea produselor.



Exemplu schema cu microprocesor ARM

Pentru design-ul radio se folosesc componente speciale alea caror specificatii prinvind atenuarea de insertie si returnloss-ul sunt caracteristici fundamentale. Mai mult de 5 astfel de dispozitive sunt evaluate pentru a lua decizia finala pentru implementarea intr-un front end RF. Deasemenea, amplificatoarele de banda larga cu zgomot redus sunt evaluate pentru a fi inegrate in produse.

Exemplu schema cu componente RF

Proiectarea si realizarea de circuite imprimate

Prin cablaj imprimat se înţelege un ansamble de conductoare de conexiune plasate în unul, două sau mai multe planuri paralele, fixate pe un suport rigid sau flexibil şi formând un ansamblu. Conductoarele, realizate sub formă de benzi sau suprafeţe metalice, se numesc conductoare imprimate.

Prin circuit imprimat, se înţelege de obicei, ansamblul suport izolant, conductoarele imprimate şi componente fixate definitiv pe suport. Adesea, noţiunile de cablaj imprimat şi circuit imprimat, se confundă. Tehnologia cablajelor imprimate are numeroase şi importante avantaje, printre care: • asigură un grad de compactizare ridicat; • se reduce volomul si dificultatea operaţiilor de montare şi asamblare, care pot fi com- plet automatizate; • asigură o mare reproductibilitate în dispunerea pieselor şi conductoarelor de conexiune, ceea ce simplifică enorm relgajele, acordările şi permite controlul cuplajelor parazite; • identificarea pieselor şi traseelor este uşoară şi se simplifică depanarea; • consumul de cupru se reduce drastic, conductoarele putând fi dimensionate în funcţie de cerinţele electrice (intensitate curent, inductanţă şi rezistenţă etc.); • fiabilitatea şi mentenabilitatea produselor sunt ridicate; • costurile sunt reduse. Singurul dezavantaj – dacă se poate vorbi despre aşa ceva, este că, pentru a pune în valoare avantajele, execuţia trebuie făcută pe maşini, în sistem industrial; realizarea artizanală (manuală) nu este corespunzătoare, cu excepţia unor cablaje simple, de exemplu pentru teste. In prezent, progresele tehnologice au determinat: • ieftenirea cablajelor, a căror utilizare şi în montaje experimentale a devenit o practică curentă; • diversificarea tipurilor de cablaje imprimate (flexibile, multistrat), a tehnologiilor de asamblare (asamblarea cu piese montate pe suprafaţă); • extinderea tehnologiilor specifice cablajelor imprimate la realizarea unor componente imprimate (rezistoare, bobine, condensatoare), a unor componente precum şi în alte domenii (bobinaje pentru motoare electrice miniatură, suspensii şi arcuri pentru meca- nisme fine etc.)

Diversitatea cablajelor este foarte mare şi clasificările se pot face din mai multe puncte de vedere.

După însuşirile mecanice ale suportului izolant, există cablaje pe suport rigid şi pe suport flexibil.
După numărul de plane în care se află conductoarele, există cablaje monostrat, dublu strat şi multistrat. •
După modul de realizare a contactelor dintre conductoarele aflate în plane diferite, există cablaje cu găuri nemetalizate şi cu găuri metalizate.
După tehnologia de fabricaţie, există: ► Cablaje realizate prin tehnologii subtractive, în care, plecând de la un suport izolant placat cu folie de cupru, conductoarele se obţin prin îndepărtarea metalului în regiu- nile care trebuie să fie electroizolante.

Aceste tehnologii sunt cele mai utilizate. Cableje ralizate prin tehnologii aditive, în care conductoarele se obţin prin depunere de metal pe suprtul izolant. Cablaje realizate prin tehnologii de sinteză, în care conductoarele şi izolantul inter- mediar se obţin prin depuneri succesive de metal şi dielectric.

Suporturile izolante trebuie să satisfacă un număr mare de cerinţe generale, cum sunt: • proprietăţi electrice bune şi stabile: rezistivitate de volum şi suprafaţă mari, pierderi (tgδ) mica, permitivitate (ε r) mică, rigiditate dielectrică mare, ...; • proprietăţi mecanice bune şi stabile: rezistenţă la solicitări mecanice, posibilitate de pre- lucrare prin tăiere, ştanţare, aşchiere, etc.; trebuie avut în vedere că solicitările mecanice sunt preluate în totalitate de suport, conductoarele imprimate neavând rezistenţă mecanică; • stabilitate dimensională şi a tuturor însuşirilor, în timp şi la acţiunea factorilor de mediu (temperatură, umiditate, şocuri, vibraţii, substanţe chimice, ...) • neinflamabilitate (unele norme impun şi autostingerea), rezistenţă la temperatura de lipire, absorbţie şi adsorbţie a apei minime; • preţ de cost redus. Pentru suporturile cablajelor flexibile se mai impun: • flexibilitate bună (rază de curbură minimă sub 1-3 mm); • coeficient de alungire la întindere cât mai mic (conductoarele de cupru nu suportă decât alungiri foarte mici); • rezistenţă la rupere mare.

Materialele stratificate sunt cele mai folosite pentru suporturi rigide, în tehnologii substractive şi aditive. Acestea se fabrică dintr-un material de bază în starturi (hârtie, ţesătură din fibre de sticlă) impregnate cu materiale de umplutură (lianţi, răşini) şi tratate termice sub presiune. În cazul semifabricatelor placate, înainte de tartare termică, se execută acoperirea cu folii de cupru pe una din ambele feţe. Stratificatele se livrează sub formă de plăci, cu dimensiuni maxime de 2000x2000 mm. Grosimile plăcilor sunt de 0.5 ... 3.2 mm; pentru cablaje mono şi dublu start grosimile uzuale sunt de 1.2 – 1.8 mm. Abaterile admise în grosime sunt mici ± 0.05 ... ± 0.1 mm.

Cablaje cu găuri metalizate se fabrică numai din materiale cu ţesătură din fibră de sticlă, uzual sticlotextolit. În compoziţia răşinii, de regulă, se introduc şi cantităţi mici de materiale de adaos, pentru îmbunătăţirea unor proprietăţi (de exemplu pentru neinflamabilitate şi autostingere).

Suporturile pentru cablaje flexibile se realizează din materiale stratificate, din mase plastice termoplaste (rar) sau termorigide, sub formă de folii cu grosimi de 0,5 – 1,5mm. Foli- ile stratificate se fac din ţesătură din fibre de sticlă foarte rară, impregnată cu răşină epoxidică şi au flexibilitatea destul de bună (raza minimă de curbură circa 2mm). Dintre masele plastice termorigide, des folosite sunt poliimidele (Kapton), cu bune însuşiri electrice (ε r = 3,3 –3,7, tgδ = 0,008), termice (rezistă la peste 400°C) şi mecanice (comparabile cu ale foliilor strati- ficate cu răşină epoxidică), asigurând o bună aderenţă a conductoarelor. Mai rar se folosesc foliile din Teflon (politetrafluoretilenă – PTFE). Dintre masele plastice termoplaste, se utilizează poliamidele şi poliesteri (PETP – Mylar, ..)

Pentru realizarea cablajelor circuitelor cu mare densitate de componente, în care se folosesc circuite integrate complexe, cu multe terminale, cablajele dublu strat fără metalizarea găurilor se pot folosi cu mare dificultate. Cablajele fiind complicate, nu se pot face toate conexiunile pe o singură faţă iar numărul trecerilor care trebuie realizate cu fire este foarte mare (spaţiu ocupat mare, manoperă multă, erori frecvente). Soluţia problemei constă în utilizarea cablajelor cu găuri metalizate, iar pentru circuitele foarte complicate a cablajelor multistrat, care sunt tot cu găuri metalizate. Fabricaţia cablajelor cu găuri metalizate (fig. 1.15) începe cu prelucrările mecanice, dintre care cea mai importantă

operaţie este găurirea. Aceasta se face numai prin aşchiere, folosind burghie speciale, antrenate cu viteză foarte mare (peste 20000 rot/min) pe maşini de precizie, asigurând o mare netezime a pereţilor găurilor. Găurile sunt curăţate cu grijă, chiar în timpul execuţiei prin suflare cu aer sub presiune. In continuare, se procedează la catalizarea întregii suprafeţe, inclusiv a interiorului găurilor. Catalizarea constă în acoperirea cu o substanţă care favorizează depunerea chimică a cuprului şi asigură o bună aderenţă a acestuia la suport (obişnuit clorid de paladiu). Apoi se face o cuprare chimică, prin care se depune un strat foarte subţire de cupru (1 – 5μm)cu rol de asigurare a conductibilităţii întregii suprafeţe. Cuprarea chimică se face în băi de reducere a sărurilor de cupru, este lentă şi scumpă şi de aceea stratul este subţire.

Pentru fabricarea cablajelor cu 3 sau 4 straturi, tehnologia subtractivă este în prezent mult folosită. Succesiunea operaţiilor apare în fig. 1.16. Procesul începe cu fabricarea cablajelor imprimate nemetalizate fără executarea găurilor, pe două sau mai multe plăci din semifabricat placat (dublu strat + dublu strat sau dublu strat + mono strat); se foloseşte tehnologia descrisă la punctul 1. Apoi se suprapun plăcile, cu izolant intermediar (material de bază + r ăşină) şi se presează la cald, obţinând un ansamblu rigid.

Pe ansamblu se execut ă găurirea, apoi se face o corodare a izolantului din găuri, pe mică adâncime, astfel încât cuprul să iasă puţin (câţiva μm) în relief, asigurând un mai bun contact cu metalizarea găurii. Urmează metalizarea găurilor (şi a conductoarelor exterioare), după procedeul des- cris la punctul 3. În producţie, probleme dificile apar la realizarea contactelor între conductoare din dife- rite straturi, datorită abaterilor poziţiilor plăcilor şi conductoarelor faţă de cele ideale, lăţimile conductoarelor şi distanţele dintre ele fiind foarte mici (adesea de 0,2 - 0,3mm).

Pentru bune rezultate, toleranţele în poziţionare sunt mici, de ordinul 10 – 20μm. Poziţionarea precisă a conductoarelor se obţine prin imprimarea desenelor cu procedeul fotografic şi tratare chimică corespunzătore.

Poziţionarea precisă a plăcilor la suprapunere se face utilizând găuri de ghidare în plăci şi ştifturi (tije) de ghidare. Executarea precisă a găurilor (poziţie, diametru ...) se obţine prin prelucrarea mecanică pe maşini speciale, conduse de calculatoare în a căror memorie este introdus planul de găurire.

O placă de cablaj imprimat brut este realizată dintr-un strat izolator, de grosime care poate varia de la câteva zecimi de mm până la ordinul câtorva mm, pe care se află o folie de cupru (simplu strat) sau două (dublu strat). Stratul izolator are în general grosimea de 1,6 mm, dar această valoare nu reprezintă un standard, deoarece depinde de foarte mulți factori, în general mecanici și tehnologici. Uzual, ca izolator, se folosește materialul cunoscut sub numele de FR4 sau Rogers pentru frecvente innalte.

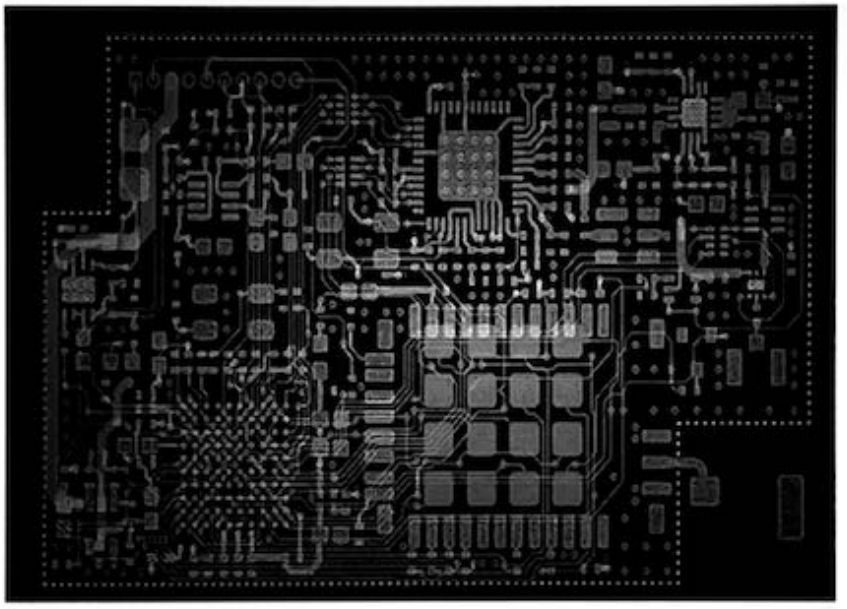
Circuitul imprimat final se realizează prin metode foto și chimice. Un circuit imprimat poate fi cu simplă față (strat conductor), dublă față sau multistrat. Circuitele imprimate multistrat sunt realizate prin suprapunerea succesivă a mai multor circuite dublu strat, separate între ele printr-un strat izolator, de obicei din material identic cu cel al cablajului brut. Trecerea transversală de la un strat la altul se realizează cu ajutorul vias-urilor și/sau a pinilor TH.

Vias-urile pot fi TH (cu trecere dintr-o parte în alta a cablajului),buried (stratul de început cât și cel de sfârșit sunt în interiorul cablajului), sau blind (se pleacă de pe un strat exterior și se ajunge pe un strat interior).

Proiectarea si realizarea circuitelor imprimate a fost realizata de asemenea folosind instrumente software dedicate si recunoscute de industrie.

Realizarea proiectarii circuitelor imprimate (PCB) a fost precedata de generarea unor fisire de executie standard pentru industria specifica urmarindu-se realizarea fizica a acestora si in final popularea lor cu componentele electronice pentru produsul final.

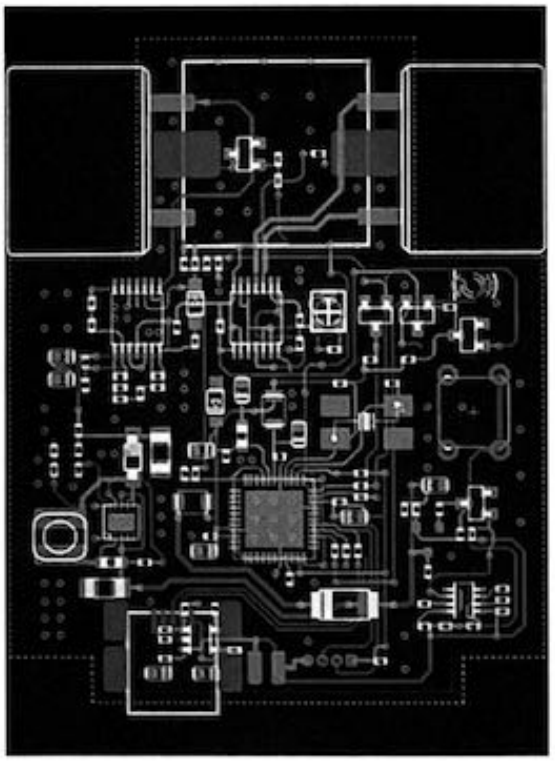
Circuitele imprimate corespund celor mai exigente standarde si sunt caracterizate de o proiectare adecvata fie ca este vorba de frecvente mici , curenti mari sau frecvente innalte. Au fost folosite tehnici de routing folosind tehnologie multistrat cu pana la 4 nivele (layers).

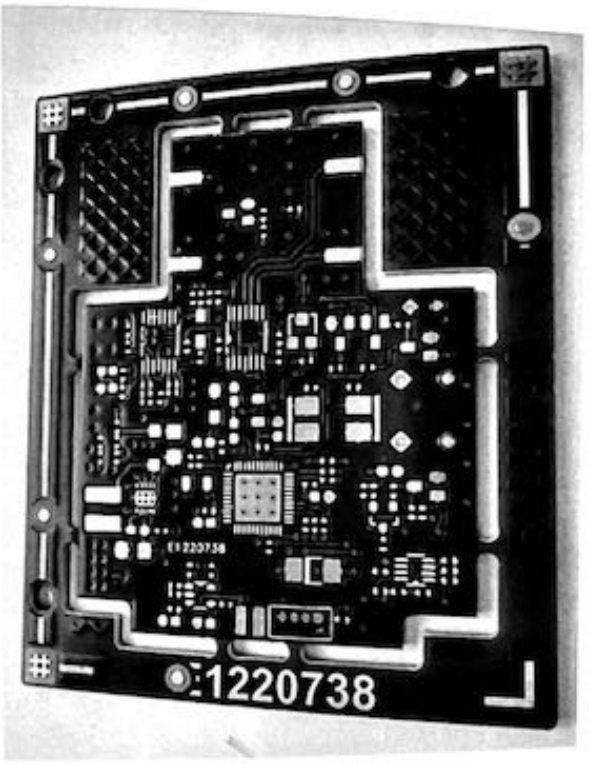


Exemplu circuit imprimat 4 straturi.

Dupa proiectarea circutelor in soft-ul de CAD acestea au fost realizate la fabrici specializate cum ar fi EUROCIRCUITS din Belgia.

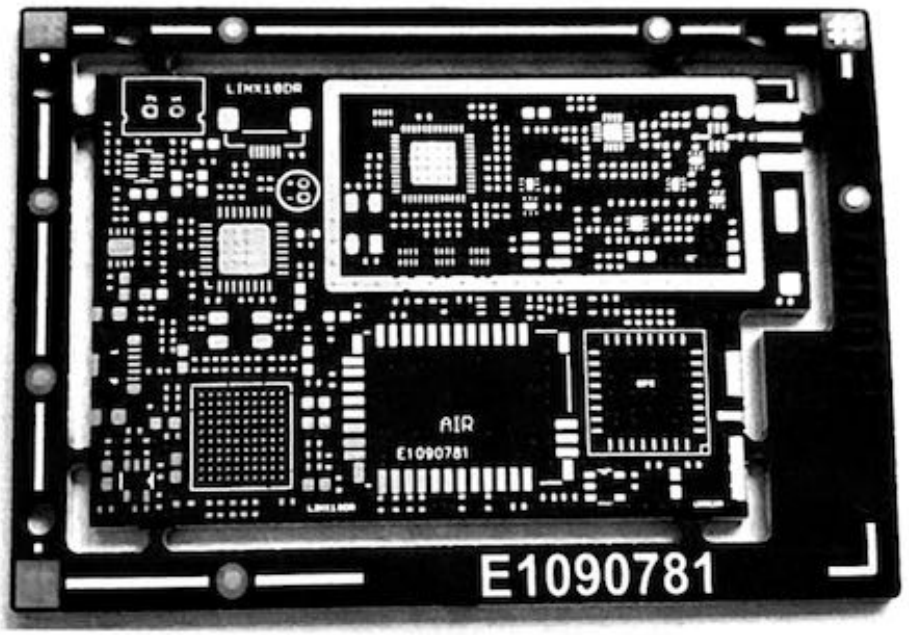
Pentru realizarea prototipurilor PCB-urile au fost populate cu componente electronice trough hole (THT) sau sourface mount (SMT) in mod manual.



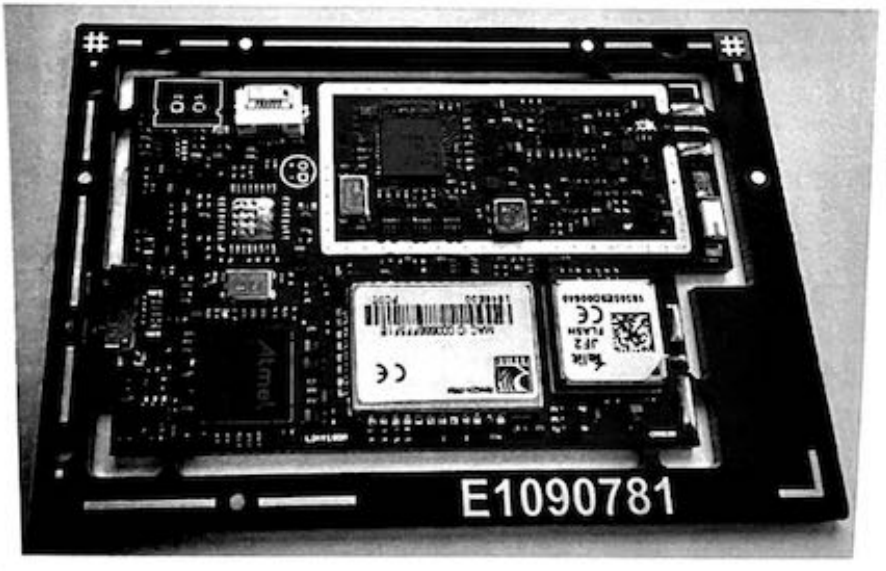


Exemplu circuit imprimat 2 straturi si produsul incasetat

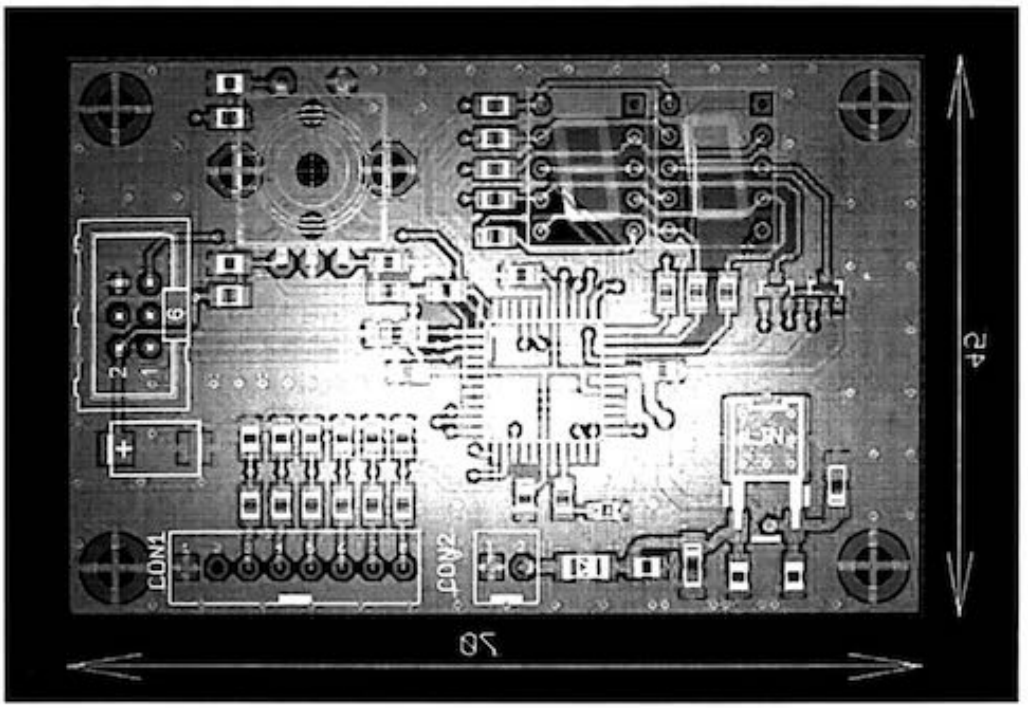
Prosdusul final incasetat simulare 3D



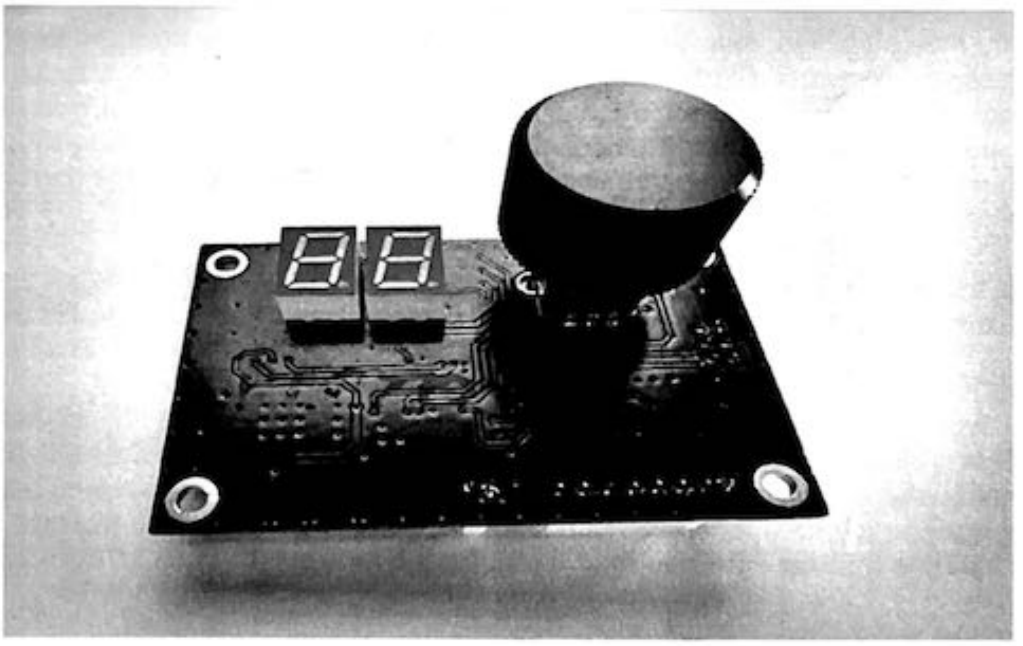
Circuit imprimat 4 straturi realizat fizic



Circuit imprimat 4 straturi realizat fizic si populat cu compenente *electronice*



Circuit imprimat 2 straturi proiectatre



Circuit imprimat 2 straturi realizat fizic

Dezvoltare de firmware pentru microcontrolere si solutii embedded

Dezvoltarea firmware-ului este procesul de scriere a codului care rulează un hardware încorporat mai degrabă decât pe un computer cu drepturi depline. Se face pe un microcontroler sau un microprocesor. Firmware-ul nu este considerat în întregime software, ci cod care gestionează senzorii, perifericele, motoarele .

Spre deosebire de software, acesta nu poate fi schimbat sau actualizat în mod repetat. Prin urmare, trebuie să funcționeze impecabil pentru anii următori. Este destul de ușor să dai exemple de astfel de software încorporat, unele dintre ele sunt; termostat, un senzor, radio. Nu interacționați cu firmware-ul într-un fel de computer sau aplicație, mai degrabă decât să se dovedească a fi ceva la care în general, ca utilizator, nici nu vă gândiți.

Software-ul include un set de comenzi care sunt programate pentru a instrui computerul. Pe de altă parte, firmware-ul este un set de linii de cod care ajută la controlul dispozitivului hardware. Software-ul este dezvoltat prin utilizarea atât a limbajelor de nivel scăzut, cât și a limbajelor de nivel înalt, unde în firmware este de obicei format din limbaje de nivel scăzut. Software-ul rulează în principal pe procesoare principale, cum ar fi CPU, pe de altă parte, firmware-ul rulează pe procesoare mici. Software-ul are, în general, dimensiuni mari, între 100 kiloocteți (kb) și câțiva gigaocteți (GB), iar firmware-ul sunt în general mici în intervale de dimensiuni de aproape câțiva kiloocteți (kb). Software-ul include caracteristici precum funcționalitatea, gradul de utilizare, fiabilitatea etc., iar firmware-ul include caracteristici precum efectuarea tuturor controalelor.

În orice program software convențional, intrările sunt citite, calculele sunt efectuate și apoi un rezultat este o ieșire. Cu toate acestea, un program de firmware citește de obicei o valoare a pinului (tensiune) ca intrare și iese, nu imprimă () așa cum ar face software-ul, ci prin schimbarea tensiunii unei iesiri logice. Pentru a concluziona cu aceasta, putem spune că un software nu poate fi niciodată firmware dar firmware-ul poate fi software.

Dezvoltare de firmware pentru microcontrolere sau alte solutii embedded a fost realizata cu ajutorul compilatoarelor pentru limbajul C. In numite cazuri si in funtie de situatie s-a lucrat si in assembler.

Uneltele folosite pentru dezvoltarea de firmware sunt consacrate si a fost urmarita constant reutilizarea surselor de cod pentru proiecte noi. Acest lucru a dus la o realizare mai rapida a firmware-ului fara a fi nevoie de efortul initial depus in proiectele anterioare.

A fost urmarita folosirea procesoarelor si microprocesoarelor din aceeasi familie (spre exemplu ARM sau AVR) pentru a fi asigurata migrarea intre ele fara efort substantial.

Spre deosebire de PC-uri şi laptopuri, care sunt destinate lucrului cu mai multe programe în acelaşi timp (multitasking), microcontroller-ele sunt destinate unor aplicaţii individuale, adică rulează mereu un singur program (firmware-ul). Asta înseamnă că procesoarele din microcontrollere nu au de muncit la fel de mult ca cele de pe PC-uri şi din acest motiv producătorii de microcontroller- e le-au dotat cu procesoare a căror frecvenţă de lucru rar depăşeşte 100 – 200 MHz (faţă de 2 - 3 Ghz cât are un PC/ laptop obişnuit). Reducerea de performanţă la strictul necesar a permis microcontrollerelor să aibă un preţ mic, ceea ce argumentează gradul de răspândire pe care îl putem observa astăzi.

Memoria RAM Atunci când realizăm niste calcule ceva mai complexe, nu putem face totul „în cap", ci avem nevoie să notăm temporar anumite rezultate pe o bucată de hârtie. În mod similar şi procesoarele au nevoie să-şi noteze temporar nişte rezultate, numai că ele nu o fac pe o bucată de hârtie ci în ceea ce se numeşte memorie RAM. RAM este prescurtarea de la Random Access Memory - memorie cu acces aleator, adică o memorie din care datele pot fi accesare (citite sau scrise) individual (spre deosebire de alte tipuri de memorie în care procesorul este nevoit să citească sau să scrie şi alte locaţii de memorie, în afară de cele care îl interesează în acel moment). Recapitulând, memoria RAM este deci o „ciornă" pe care procesorul o foloseşte pentru a-şi face calculele. Ea este o memorie volatilă, adică datele memorate în ea se şterg atunci când alimentarea microcontroller-ului este întreruptă, această memorie neputând fi folosită pentru salvarea permanentă a datelor. Reducerea performanţelor la necesităţi, face ca resursele de RAM ale unui microcontroller să fie cu mult inferioare celor din componenţa unui PC. De regulă, microcontroller-ele dispun de memorii RAM de zeci sau sute de kilobiţi (kb, mai rar megabiţi).

Memoria Program (FLASH) Memoria Program este memoria în care este scris firmware-ul. În ultimii ani, memoria program folosită este de tip FLASH (adică exact ca cea din stick-urile flash USB). Conţinutul memoriei program, scris în ea în momentul instalării firmware-ului, am putea spune că este „o listă de instrucţiuni al microcontroller-ului". Ulterior, microcontroller-ul nu face decât să urmărească întocmai instrucţiunile din această listă de instrucţiuni, fără a-i modifica în nici un fel conţinutul. Memoria program este nevolatilă, continutul ei fiind disponibil şi după deconectarea alimentării, sau resetarea/ repornirea microcontroller-ului. Spre deosebire de memoria RAM, memoria FLASH este garantată doar pentru un număr de maxim de scrieri (de ex. 10000). Firmware-urile sunt programe mult mai mici decât programele pentru PC, şi din acest motiv, rareori depăşesc 1-2 Mb.

Memoria pentru salvări de date (EEPROM) În aplicaţiile practice, microcontroller-ele trebuie uneori să memoreze nişte date undeva unde să le poate regăsi intacte oricând,

chiar şi după restartarea/ repornirea echipamentului pe care îl controlează. Din punct de vedere fizic, această memorie este de tip EEPROM, adică Electrically Erasable Programmable Read-Only Memory, o memorie doar pentru citire (Read Only Memory), care poate fi ştearsă în condiţii speciale (utilizând tensiuni şi câmpuri electrostatice mult mai mari decât cele de operare) şi rescrisă. În timpul funcţionării microcontrollerului, datele sunt de obicei doar citite (nu şi scrise). În această categorie de date memorate intră: valori de calibrare, setări personalizate, mesaje de eroare, etc. Optimizarea resurselor microcontroller-elor a determinat ca memoria EEPROM să aibă în general valori de Mb. Principiul de funcţionare face ca memoriile EEPROM să fie garantate pentru un număr de scrieri de aprox. 100.000.

Pini pentru Intrări şi Ieşiri programabile Cei mai mulţi pini de care dispune un microcontroller sunt pini prin care acesta comunică cu restul lumii. Aceşti pini sunt numiţi intrări şi ieşiri programabile pentru că destinaţia lor nu este definitiv fixată prin construcţie (aşa cum este cazul componentelor electronice obişnuite) ci este stabilită de instrucţiunile scrise în firmware. Astfel, printr-o programare corespunzătoare intrările şi ieşirile programabile pot fi configurate să aibă funcţii de: •

Intrare digitală. În această configuraţie pinul respectiv poate primi şi interpreta doar semnale digitale: 0 logic (de obicei, dacă tensiunea aplicată la intrare apropiată de 0 V)

sau 1 logic (de obicei, dacă tensiunea intrării este apropiată de tensiunea de alimentare a microcontrollerului);

Ieşire digitală. Pe pinul respectiv microcontroller-ul poate livra doar semnale digitale: 0 logic (aprox. 0V) sau 1 logic (o tensiune practic egală cu tensiunea de alimentare a microcontroller- ului);

Intrare analogică. Tensiunea aplicată pe un astfel de pin, este dirijată către un convertor analog digital (convertor A/ D). În acest mod, microcontroller-ul poate „înţelege" şi alte semnale de intrare, nu doar pe cele de 0 şi 1 logic. Sau altfel spus, un pin configurat ca intrare digitală nu înţelege decât 5

ninele de alb şi negru, pe când un pin configurat ca intrare analogică poate înţelege şi o multitudine de nuanţe de gri. Numărul acestor nuanţe de gri depinde de rezoluţia convertorului A/ D cu care este dotat microcontroller-ul. Dacă rezoluţia acestuia este de 10 biţi, acel pin va putea „deosebi" 1024 de „niveluri de gri"; •

Ieşire PWM (Pulse Width Modulation). Semnalul PWM este un tren de impulsuri cu frecvenţă constantă, dar cu durată variabilă (modulată). Dacă la un pin al unui microcontroller configurat ca ieşire PWM conectăm un filtru RC, care să integreze aceste impulsuri, obţinem o tensiune aproximativ continuă, având valoarea proporţională cu durata impulsurilor. În acest mod, din microcontroller putem obţine semnale de ieşire analogice. Rezoluţia unei ieşiri configurate ca ieşire PWM este de obicei doar de 8 biţi (adică doar 256 de „nivele").

Pini oscilator - cuarţ La baza funcţionării oricărui procesor stă un oscilator. Semnalul creat de acest oscilator este cunoscut sub numele de semnal de ceas, semnal de tact, sau clock-ul sistemului. El, în principiu, are două funcţii: • •

dictează viteza de lucru a procesorului; asigură sincronizarea tuturor proceselor realizate de procesor.

Necesitatea sincronizării proceselor executate de un procesor devine mai evidentă atunci când avem un modul electronic în care mai multe procesoare trebuie să comunice între ele. Fără a avea o sicronizare precisă între procesele fiecărui procesor, ar fi ca şi cum am avea o orchestră în care fiecare interpret îşi cântă partitura cu altă viteză. Pentru a îndeplini funcţiile menţionate mai sus, oscilatorul unui procesor trebuie să menţină o frecvenţă de oscilaţie foarte constantă, independent de condiţiile de mediu (temperatură, tensiune de alimentare, umiditate, etc.). Din acest motiv, în marea majoritate a cazurilor, procesoarele sunt „dirijate" de oscilatoare cu cuarţ. Pinii oscilator cuarţ sunt deci pinii prin care microcontrollerului i se poate ataşa un cuarţ/ oscilator cu cuarţ.

Pini pentru setarea tensiunii referinţă a convertorului analogic – digital. Ştim deja că un convertor A/ D transformă un semnal analogic (de exemplu tensiunea de 3,25 V) într-un semnal digital (de ex. numărul 658 în cod hexa). Fiecare convertor A/ D poate converti doar o anumită gamă de semnale de intrare (tensiuni) într-o anumită gamă de semnale digitale de ieşire (numere), cu o anumită eroare maximă de conversie. Pentru a înţelege ce este eroarea de conversie, să presupunem un caz foarte simplu: un convertor A/D care poate transforma gama 0 - 5V în gama numerelor de la 0 la 5. Pentru 0 V la intrare vom obţine numărul 0 la ieşire, pentru 1V la intrare vom obţine numărul 1 la ieşire şi aşa mai

departe. Însă ce număr vom obţine la ieşire când la intrare aplicăm un semnal de 0,7 V ? Convertorul nu ne poate da la ieşire numărul 0,7 (pentru că el nu poate oferi la ieşire decât numere întregi) aşa că va aproxima rezultatul conversiei la cel mai apropiat număr întreg. Deoarece 0,7 este mai aproape de 1 decât de 0, rezultatul conversiei va fi 1. Rezultatul unei conversii 100% precise ar fi numărul 0,7, ceea ce înseamnă că în exemplul nostru avem o eroare de conversie de 1V – 0,7V = 0,3 V. Deci, eroarea maximă de conversie arată valoarea maximă a aproximării pe care respectivul convertor A/D o poate face.

Aceasta se calculează împărţind gama semnalelor de intrare la numărul valorilor posibile la ieşire, ceea ce înseamnă ca în exemplu nostru vom avea o eroare de conversie de maxim 5V/ 5 = 1V. Parametrii unui convertor A/D pot fi ajustaţi după necesităţi. Pentru un convertor A/D de 10 biţi, adică unul care poate oferi la ieşire numere între 0 şi 1023 (cum este cazul majorităţii 6 convertoarelor A/D din microcontroller-e) putem avea de exemplu următoarele moduri de funcţionare: • pentru semnale analogice de intrare între 0 şi 5V, convertorul A/D oferă la ieşire numere între 0 şi 1023. În acest caz, eroarea maximă de conversie este de maxim 5V/ 1024 = 4,88 mV. De ce am împărţit la 1024 şi nu la 1023 cât am spus că poate fi numărul maxim dat la ieşirea convertorului A/D ? Pentru că am ţinut cont şi de numărul „0" care este şi el una din valorile pe care convertorului A/D îl poate oferi ca rezultat; • pentru semnale analogice de intrare între 0 şi 1V, convertorul A/D oferă la ieşire numere între 0 şi 1023. În acest caz, eroarea maximă de conversie este de 1V/ 1024 = 0,9765 mV. Observăm că în primul mod de funcţionare, convertorul A/D poate lucra cu o gamă mai mare de tensiuni de intrare (0V -5 V) însă cu o eroare maximă de conversie mai mare. În cel de-al doilea caz, situaţia este exact invers: convertorul suportă o gamă mai mică de tensiuni de intrare (0-1V), însă eroarea maximă de conversie este mult mai mică.

Nu putem avea concomitent şi gama mare de semnale de intrare şi eroare de conversie mică, motiv pentru care trebuie să ne decidem asupra unui compromis între cele două. După luarea acestei decizii, apare întrebarea: cum impunem convertorului A/D din microcontroller să adopte gama de semnale de intrare (şi implicit eroarea maximă de conversie) pe care o dorim? Răspunsul este: prin aplicarea unei tensiuni de referinţă pe

pinii care în fig. 1.1 sunt numiţi pini setare tensiune referinţă convertor analogic – digital. Ce valoare trebuie să aibă această tensiune de referinţă? Exact valorea tensiunii maxime din gama de semnale de intrare dorită. Mai precis, dacă dorim ca gama semnalelor acceptate la intrare să fie între 0 si 2,5896542 V, tensiunea de referinţă trebuie să fie şi ea tot 2,5896542 V. Multe microcontroller-e încorporează câteva surse de tensiuni de referinţă care pot fi activate prin instrucţiunile scrise în firmware. Valorile acestor tensiuni de referinţă „interne" sunt fixe şi au de obicei valorile: 1,1V/ 2,5V/ 5V.

Pini de alimentare Acesti pini fac ceea ce le sugerează şi numele: alimentează microcontroller-ul şi tot ce este conţinut în el. Din punct de vedere al tensiunii de alimentare există două tipuri de microcontrollere: cele alimentate la 3,3V şi cele alimentate la 5V.

Principalele funcţii realizabile de către un microcontroller Un microcontroller, programat corespunzător, poate deci realiza funcţiile multor alte tipuri de circuite electrice „clasice". Pe lângă acestea, microcontrollere pot realiza funcţii care altfel ar fi deosebit de greu de realizat (comunicaţii de date fără erori, afişare de date pe diferite tipuri de display-uri, etc.). Nu în ultimul rând versatilitatea microcontroller-elor le permite cercetătorilor

(profesionişti sau amatori) să-şi imagineze şi să dezvolte sisteme embedded cu noi şi noi functionalităţi. În continuare, vom enumera cele mai întâlnite funcţionalităţi posibile ale microcontrollerelor.

Preluarea de date de la senzori. Senzorii sunt dispozitive care transformă un semnal de natură neelectrică (presiune, forţă, viteză, temperatură, etc.) într-un semnal de natură electrică. Cu ajutorul unui senzor adecvat, un microcontroller poate „citi" practic orice fel de mărime fizică. b. Introducerea de comenzi prin intermediul tastelor şi encoderelor (de ex. rotiţa de la mouse). Controlul unui sistem analogic poate presupune utilizarea de butoane complicate şi de potenţiometre. În contrast, comenzile unui microcontroller se poate efectua cu ajutorul unor taste simple, ieftine, durabile şi nu în ultimul rând, elegante.

Afişarea de date pe diferite tipuri de display-uri. În sisteme analogice, pentru fiecare informaţie de afişat, este necesar un dispozitiv de afişare (un bec, un led, etc). Într-un sistem cu microcontroller, acelaşi display poate fi folosit pentru a afişa o multitudine de informaţii d. Implementarea de structuri de meniuri. Într-un sistem analogic, practic pentru orice comandă trebuie să ai un buton dedicat. Într-un sistem cu microcontroller în care s-a creat o structură de meniuri, este posibil un număr uriaş de comenzi folosind doar câteva taste. e. Prelucrarea facilă şi rapidă a datelor de intrare. Practic nu există funcţie de prelucrare a semnalelor care să nu poate fi implementată cu succes cu ajutorul unui microcontroller.

Memorarea de date. În contrast cu sistemele analogice, sistemele cu microcontroller pot memora practic orice tip de date (valoare impusă, valori medii, maxime, minime, alarme, etc). g. Crearea de semnale de ieşire de tip On/ Off şi PWM. Semnalele de ieşire de tip On/ Off pot fi folosite pentru funcţii de activare/ dezactivare (de exemplu a unor becuri, sonerii, electromagneţi, etc.), iar al doilea poate fi folosit pentru funcţii de control precis (a turaţiei unui motor, a intensităţii luminoase a unui bec/ led, a tensiunii de ieşire a unui regulator de tensiune, etc).

Programarea microcontrollerelor Noi oamenii suntem obişnuiţi cu limbaje ce conţin litere, semne de punctuaţie şi cifre, dar firmware-ul şi implicit microcontroller-ul foloseşte limbajul digital (nu ştie decât de 0 şi de 1 logic iar firmware-ul este şi el o listă de instrucţiuni de lucru care nu conţine altceva decât şiruri de 0 şi de 1). Programarea la nivel de bit, sau altfel spus în cod maşină, este deosebit de dificilă, din acest motiv cercetătorii în informatică au dezvoltat nişte utilitare ajutătoare: o limbaje de programare, care sunt colecţii de cuvinte speciale care pot fi utilizate pentru a scrie firmware-ul. Fişierul sau fişierele în care este scris firmware-ul poartă denumirea de Cod sursă; o compilatoare, care sunt nişte programe care au funcţie de translator – traduc codul sursă într-un limbaj pe care microcontroller-ul îl poate înţelege (limbaj denumit şi cod maşină). Cele două utilitare de mai sus se regăsesc în componenţa unor programe sau set-uri de programe care au denumirea generică de medii de dezvoltare integrate (Integrated Development Environment - IDE). Pentru fiecare familie de microcontrollere, şi există pe piaţă o foarte mare varietate de tipuri de microcontrollere şi de firme producătoare, există medii de dezvoltare recomandate. Iată câteva din cele mai întâlnite: AVR studio - este un mediu de dezvoltare profesional creat pentru familia AVR de microcontrollere produse de ATMEL (începând din 1996), optimizat spre a fi utilizat cu uşurinţă şi de amatori. AVR a fost una

dintre primele familii de microcontrolere care a utilizat o memorie flash on-chip pentru stocarea programelor, spre deosebire de alte tipuri: ROM, EPROM, sau EEPROM, utilizate de alte microcontrolere la momentul respectiv.

Pentru microcontrolerele AVR se găsesc o multitudine de aplicații, ele fiind de asemenea, utilizate în linia Arduino de sisteme open source. MPLABX IDE – un mediu de dezvoltare destinat micrcontrollerelor produse de firma MICROCHIP. La fel ca şi AVR Studio, MPLABX IDE este un mediu de dezvoltare profesional; Arduino IDE. Este un mediu de dezvoltare destinat îndeosebi constructorilor amatori. Cu ajutorul acestuia se pot realiza firmware-uri pentru microcontrollerele AVR - ATMEL, şi nu numai, mai nou Arduino suportând dezvoltări şi pe alte tipuri de microcontrollere. Când este finalizat de scris codul sursă într-un limbaj de programare/ mediu de dezvoltare, vine momentul să instalăm acel cod în memoria program a microcontroller-ului. Pentru aceasta, în primul rând trebuie conectat microcontroller-ul de programat la PC-ul/ laptopul pe care avem codul sursă, problemă care se rezolvă fie cu un simplu cablu USB (ca în cazul modulelor de tip Arduino) fie prin intermediul unui modul special numit programator (de exemplu REAL ICE). După realizarea conexiunii cu microcontroller-ul nu ne mai rămâne decât să apăsăm butonul de upload din ferestra de lucru a mediului de dezvoltare, iar acesta face totul: transformă codul sursă în limbajul microcontroller-ului şi apoi îl instalează în memoria program a acestuia.

Etapele dezvoltării unui proiect cu microcontrollere a. Înainte de a ne apuca să construim o aplicaţie cu microcontrollere, este recomandat să desenăm o schemă bloc în care să adaugăm toate funcţionalităţile de care proiectul are nevoie, precum şi toate legăturile dintre ele. Obţinem astfel o imagine de ansamblu a proiectului, care conţine toate resursele de care are nevoie pentru a funcţiona aşa cum dorim

Cunoscând blocurile funcţionale necesare proiectului, putem deja să realizăm ce parametri ar trebui să aibă microcontrollerul, adică: •

numărul minim de pini din fiecare categorie: intrări analogice, intrări/ ieşiri digitale, ieşiri

PWM, • viteza de lucru, • dimensiunea memoriei program. Criteriul cel mai important în alegerea microcontroller-ului este numărul minim de pini, toate celelalte caracteristici fiind natural legate de acest parametru. De exemplu, pentru o aplicaţie mai complexă, sunt necesare mai multe semnale (adică mai mulţi pini), deci de o prelucrare de date mai rapidă (viteză mare de lucru) şi de un firmware cu multe instrucţiuni (o memorie program mare).

Cunoscând schema bloc şi tipul de microcontroller ce-l vom folosi, trecem la proiectarea parţii hardware (cablaj imprimat, componente electrice, electronice, electromecanice, optice, etc). d. Crearea firmwareului - proiectarea părţii software. Mai întâi se scrie o parte din firmware, se instalează pe microcontroller (care este conectat la partea hardware), se verifică dacă sistemul se comportă bine, iar apoi se adaugă funcţionalităţi. Dacă nu, se modifică firmware-ul şi se reia procedura. e. Testarea aplicaţiei, care să are ca scop depistarea de bug-uri (greşeli) în firmware, care în anumite circumstanţe pot afecta parţial sau total funcţionarea sistemului; evaluarea comportării în timp a sistemului.

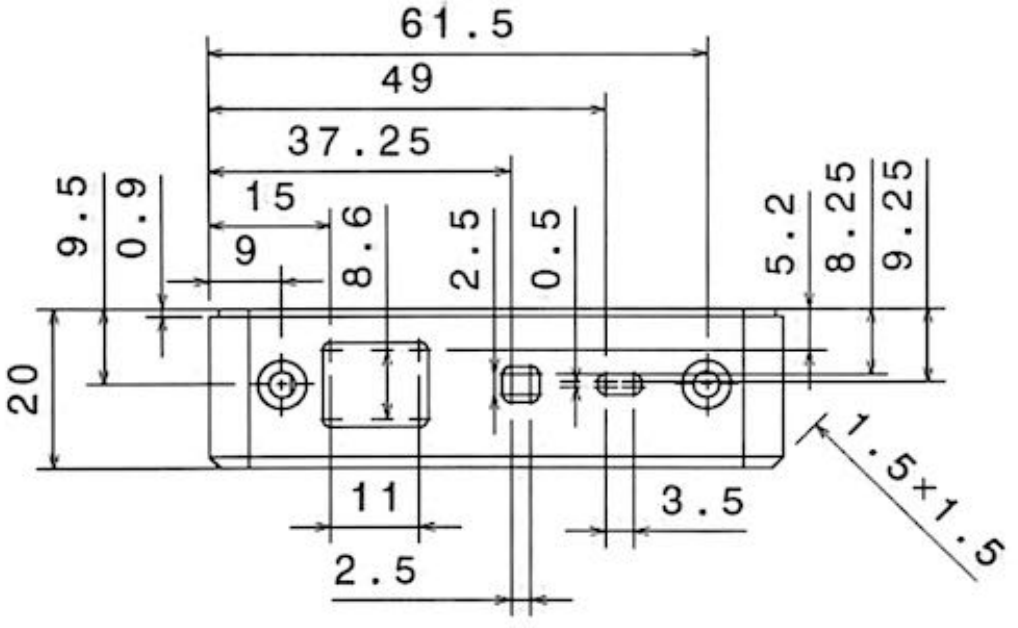
Desenul tehnic industrial foloseste ca metoda de reprezentare proiectia cilindrica ortogonala pe doua sau mai multe plane de proiectie cu scopul de a se obtine imagini nedeformate a tuturor dimensiunilor. Pentru obiectele de complexitate mare, când proiectiile pe doua sau trei plane de proiectie nu redau imaginea completa a obiectului, se recurge la proiectia pe mai multe plane, care formeaza asanumitul cub de proiectie.

Trecerea de la dispunerea spatiala la cea plana se realizeaza prin rabaterea fetelor cubului de proiectie, pe planul vertical, obtinându-se sase proiectii, în corespondenta (fig.1.2), dupa cum urmeaza: 1 – vederea din fata, dupa directia A, se obtine pe planul vertical din spate; este numita si proiectie principala, ea cuprinzând cele mai multe detalii de forma si dimensiuni ale obiectului; 2 – vederea de sus, dupa directia B, se obtine pe planul orizontal inferior si se amplaseaza sub proiectia principala;

Vederea din stânga, dupa directia C, se obtine pe planul lateral din dreapta si se amplaseaza în dreapta proiectiei principale;

Vederea din dreapta, dupa directia D, se obtine pe planul lateral din stânga si se amplaseaza în stânga proiectiei principale;

Vederea de jos, dupa directia E, se obtine pe planul orizontal superior si se amplaseaza deasupra proiectiei principale;



Desen tehnic cu cote

Vederea din spate, dupa directia F, se obtine pe planul vertical din fata si se amplaseaza în dreapta sau stânga proiectiei principale dupa proiectia C, respectiv D. E

Piesele care pot fi utilizate în orice pozitie (suruburi, piulite, stifturi, arbori, axe, tije etc.) se reprezinta de regula în pozitia de prelucrare (sau de asamblare), adica cu axa orizontala. În cele mai multe cazuri, la reprezentarea unui obiect sunt suficiente una, doua sau trei proiectii, respectiv dupa directiile A, B si C.

Simbolurile grafice de identificare pentru cele doua metode se amplaseaza pe desen numai daca este strict necesar, în stânga sau în interiorul indicatorului. Metoda E

Când la scara desenului, o anumita portiune nu este suficient de clara, se utilizeaza reprezentarea în detaliu . Aceasta se încadreaza într-un cerc sau dreptunghi trasat cu linie continua subtire si se reprezinta la o scara de marire, în vedere sau sectiune. Detaliul se limiteaza cu linie de ruptura (sau nu) si poate cuprinde amanunte nereprezentate în portiunea din care provine. Elementele repetitive (gauri, danturi, suruburi, piulite) pot fi reprezentate complet o singura data în totalitate, în pozitii extreme sau pe o mica

portiune , restul elementelor fiind reprezentate simplificat. Numarul, forma si pozitia elementelor se coteaza sau se indica în câmpul desenului.

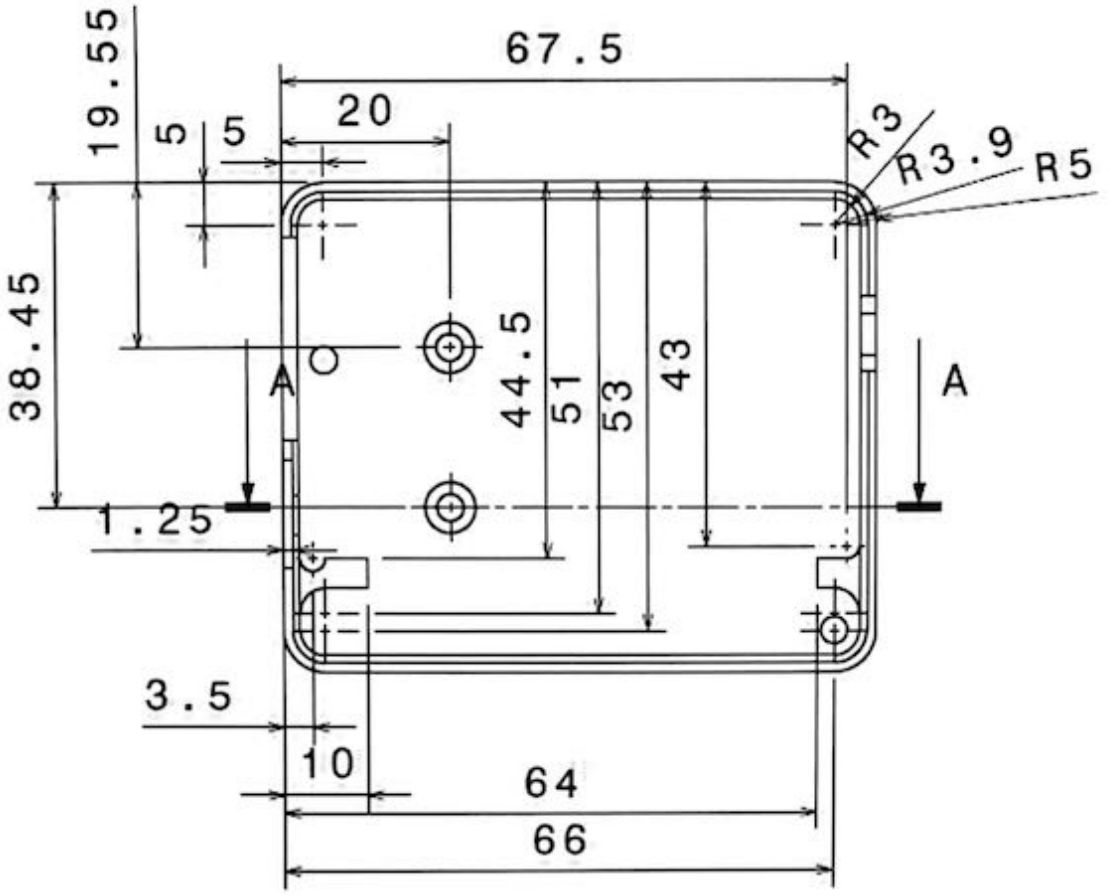
Clasificarea sectiunilor Pentru reprezentarea unui desen ce reprezinta o piesa cu unul sau mai multe goluri de-a lungul axelor sau paralele cu acestea, se foloseste reprezentarea în sectiune pe unul, doua sau mai multe plane de proiectie ale sistemului ortogonal de reprezentare. Sectiunea este reprezentarea în proiectie ortogonala pe un plan a obiectului, dupa intersectarea acestuia cu o suprafata fictiva de sectionare si îndepartarea imaginara a partii obiectului aflata între observator si suprafata respectiva.

Conturul si muchiile reale vizibile din sectiune se reprezinta cu linie continua groasa, cu exceptia sectiunilor suprapuse

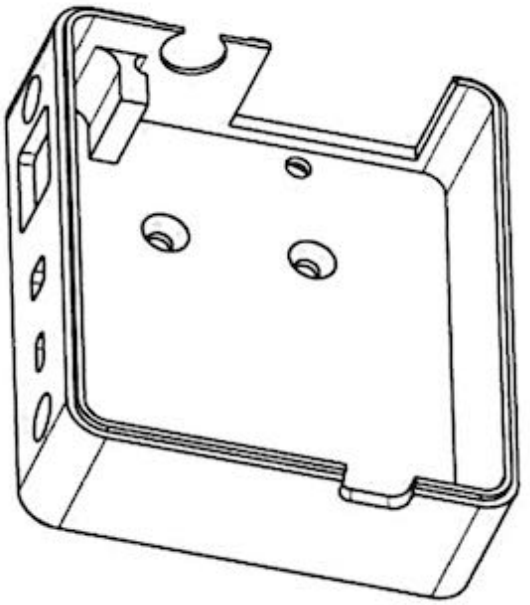
Suprafetele rezultate prin sectionare se hasureaza; 3. În cazul sectiunilor în trepte, în locul de trecere dintr-un plan în altul, se recomanda decalarea hasurilor

Piesele pline cum sunt suruburile, stifturile, niturile, arborii, osiile, precum si bielele, spitele, nervurile se reprezinta nesectionate chiar daca axa lor longitudinala se afla în planul de sectionare. Nervurile, tablele sau aripile, se reprezinta sectionate numai în cazul sectiunilor transversale.

Portiunile sectiunilor frânte neparalele cu unul din planele de proiectie se rotesc pâna ajung paralele cu planul considerat, pentru ca sectiunea sa rezulte nedeformata . Daca partea înclinata este cuprinsa între doua plane paralele, aceasta nu se mai rabate



Desen tehnic cu cote



Desen tehnic vedere ansamblu general 3 D

Sectiunile propriu-zise suprapuse, deplasate sau intercalate, se executa în proiectie vazute din stânga sau de sus si fara notarea traseului de sectionare. La reprezentarea jumatate vedere – jumatate sectiune, sectiunea se reprezinta în dreapta axei, daca axa proiectiei este verticala si respectiv sub axa, daca axa proiectiei este orizontala

Traseul de sectionare, reprezentând urma suprafetei de sectionare pe planul de proiectie, se reprezinta cu linie-punct subtire având la capete si în locurile de frângere segmente de linii groase.

Pe segmentele de la capete se amplaseaza sageti orientate în sensul de privire si respectiv de proiectare sau de amplasare a proiectiei. În dreptul sagetilor si daca este cazul si în locurile de frângere a traseului de sectionare, se înscrie litera de identificare a sectiunii. Sectiunile cu forma identica se noteaza cu aceeasi litera si se reprezinta o singura data.

Proiectiile reprezentate în pozitie rotita sau desfasurate se noteaza cu simbolul respectiv, amplasat deasupra proiectiei

Directia hasurilor poate fi spre dreapta sau spre stânga astfel încât sa nu coincida cu orientarea liniilor de contur sau a axelor. De aceea se admit hasurari sub unghiuri de 300 sau 600, pentru evitarea paralelismelor . Distanta dintre hasuri se alege în functie de marimea suprafetei hasurate si poate fi de minim 1 mm.

Directia si distanta dintre hasuri se pastreaza aceleasi pentru un obiect pe toate

reprezentarile acestuia executate pe acelasi format. Piesele dintr-un ansamblu care sunt în contact se hasureaza cu linii orientate distinct.

Sectiunile pieselor cu grosime pe desen sub 2 mm se pot înnegri, lasând între sectiunile alaturate un spatiu de minim 1 mm Daca o cota este înscrisa în zona hasurilor, atunci acestea se întrerup.

Reprezentarea rupturilor Ruptura este reprezentarea în proiectie ortogonala pe un plan a unei piese, asa cum ar arata aceasta dupa ce ar fi îndepartata o parte din ea, separând aceasta parte de restul piesei printr-o linie ondulata subtire (tip C). Ruptura se foloseste în scopul:

reprezentarii unor detalii interioare, acoperite în vedere - reducerii spatiului ocupat de reprezentare pe desen, fara sa fie afectata claritatea si precizia acesteia
detalierii unei parti la o scara de marire (fig.2.13). Linia de ruptura se executa cu linie continua subtire si este ondulata pentru rupturi în piese de orice forma si din orice material. Linia de ruptura nu trebuie sa coincida cu o muchie sau cu o linie de contur a obiectului sau sa fie trasata în continuarea acestora.

Cotarea este o operatie importanta în activitatea de proiectare si trebuie sa se execute astfel încât sa ofere toate datele necesare întelegerii si executiei obiectului reprezentat.

Modul de cotare difera în functie de destinatie, gradul de detaliere si continutul obiectului reprezentat. Astfel se pot deosebi desene de studio, de executie (desen de piesa), de operatie, de ansamblu, de montaj etc. În cele ce urmeaza se prezinta principalele reguli, metode si principii de cotare a desenelor de piese. Cotarea este operatia de înscriere pe un desen a dimeniunilor formelor geometrice simple din care este alcatuita piesa, precum si a celor care stabilesc pozitia reciproca a acestora.

Dimensiunile înscrise pe desen (cotele) pot rezulta: prin masurarea obiectului existent (în activitatea de relevare), prin calcule, sau sunt alese constructiv (în activitatea de proiectare). Cota este valoarea numerica a dimensiunii elementului cotat, înscrisa direct pe desen sau printr-un simbol literal, în cazul desenelor care cuprind tabele de dimensiuni. În functie de rolul pe care îl au în definirea unui obiect, cotele pot fi: functionale (F); nefunctionale (NF) sau auxiliare (AUX).

Operatia de cotare se realizeaza folosind urmatoarele elemente :

liniile ajutatoare;
linia de cota;
liniile de indicatie; - extremitatile liniei de cota si punctul de origine; - valoarea propriu-zisa a cotei .

Liniile ajutatoare se executa cu linie continua subtire. Ca linii ajutatoare pot fi folosite liniile de contur sau de axa. Prelungirile liniilor de cota pot fi folosite ca linii ajutatoare numai în cazul cotarii profilurilor curbe.

Liniile de cota sunt liniile deasupra carora se înscriu cotele respective. Ele se traseaza cu linie continua subtire si se dispun paralel cu elementele la care se refera, putând fi dupa caz, drepte, frânte sau sub forma de arc de cerc, în cazul cotarii dimensiunilor unghiulare sau a arcelor de cerc . Distanta dintre doua linii de cota paralele, precum si distanta dintre linia de cota si linia de contur, paralela cu aceasta, trebuie sa fie de min. 7mm. Trebuie evitata pe cât posibil intersectarea liniilor de cota între ele, precum si intersectarea acestora cu linii de indicatie si linii ajutatoare; în acest scop se recomanda dispunerea liniilor de cota în afara conturului proiectiilor obiectului reprezentat, în ordinea crescatoare a cotelor. În cazul în care aceasta dispunere nu este posibila, liniile de cota nu se întrerup.

Datele privind toleranta geometrica se înscriu într-un cadru dreptunghiular, împartit în doua sau mai multe casute trasate cu linie continua subtire. În aceste spatii se înscriu urmatoarele elemente:
simbolul tolerantei geometrice;
valoarea tolerantei în mm - litera (literele) de identificare a bazei (bazelor) de referinta
Dimensiunile caracterelor sunt aceleasi cu cele utilizate pentru înscrierea cotelor. Daca se prescrie o toleranta fara indicarea bazei de referinta, aceasta se aplica la toate suprafetele paralele cu suprafata pe care este indicata toleranta.

Daca zona tolerata este circulara sau cilindrica, se înscrie simbolul φ înaintea tolerantei . Daca pentru un element este necesar sa se indice doua sau mai multe tolerante, acestea se înscriu una sub alta .

Indicarea elementului tolerat Cadrul dreptunghiular pentru înscrierea tolerantei se leaga de elementul tolerat (suprafata la care se refera toleranta) printr-o linie de indicatie (dreapta sau frânta), terminata cu o sageata care se poate sprijini:

Desenul unei piese este reprezentarea în proiectie ortogonala a tuturor formelor geometrice componente ale acesteia, într-un numar corespunzator de proiectii astfel încât piesa sa fie clar si complet determinata. La baza întocmirii unui desen stau regulile si normele de reprezentare si cotare prezentate în capitolele anterioare. Desenele se întocmesc fie dupa piese existente, numite desene de releveu, fie sunt concepute de proiectant numite desene de proiect.

Desenul poate fi executat sub forma de schita sau sub forma de desen la scara. Schita este un desen executat, în general, cu mâna libera, în creion, pe un format corespunzator unei reprezentari cât mai clare, dupa model sau din conceptie, în proportie marita sau micsorata, în limitele aproximatiei vizuale. Schita serveste la întocmirea desenelor de executie.

Întocmirea schitei Întocmirea schitei dupa o piesa model comporta doua etape principale:

A) Studiul preliminar al piesei, care consta în: - identificarea piesei, prin care se stabileste denumirea piesei, rolul ei functional, a pozitiei de functionare în ansamblul din care face parte si a legaturilor cu piesele învecinate; - studiul tehnologic, prin care se stabileste

procedeul de obtinere a piesei, materialul din care este confectionata si prelucrarile la care a fost supusa; - analiza formelor geometrice ale piesei (exterioare si interioare); - stabilirea pozitiei de reprezentare (în pozitie de functionare sau pozitie de prelucrare), astfel încât în proiectia principala sa apara cât mai multe detalii de forma si dimensionale; - stabilirea numarului necesar de proiectii (vederi, sectiuni sau proiectii combinate), în functie de gradul de complexitate al piesei.

B) Executia grafica a schitei, cu urmatoarele faze: - alegerea formatului, trasarea indicatorului si trasarea dreptunghiurilor minime de încadrare ale proiectiilor, cu linie subtire, urmarind ca distantele dintre dreptunghiuri sa permita înscrierea cotelor (minim 20-25 mm); - trasarea (cu linie punct subtire) a axelor de simetrie ale formelor geometrice ce compun piesa, în toate dreptunghiurile de încadrare, depasind conturul acestora cu 2-3 mm; - trasarea conturului exterior, pe toate proiectiile, cu linie continua subtire

trasarea conturului interior pentru proiectiile care cuprind sectiuni, marcându-se si traseele de sectionare .
trasarea filetelor, a racordarilor, a muchiilor fictive si a celor acoperite (daca e cazul);
cotarea schitei, care consta în masurarea dimensiunilor pe piesa si înscrierea pe desen a cotelor, utilizând simboluri unde este cazul;
hasurarea suprafetelor rezultate dupa sectionare;
îngrosarea liniilor de contur si stergerea dreptunghiurilor de încadrare;

Întocmirea desenului la scara Desenele tehnice se executa dupa schita, la scara,

prin scara întelegându-se raportul dintre dimensiunea liniara a reprezentarii unui segment al unui obiect de pe un desen original si dimensiunea liniara reala a segmentului respectiv.

Scara (SR EN ISO 5455:1997) se alege în functie de complexitate si dimensiunile obiectului de reprezentat si de destinatia desenului respectiv. Ea trebuie sa fie suficient de mare pentru a permite interpretarea corecta a datelor furnizate de desenul respectiv.

Scara si dimensiunile obiectului de reprezentat influenteaza alegerea formatului de desen. Scarile de reprezentare se aleg conform tabelului 8.1. Tabelul 8.1 Scari de marire 2:1 10:1 50:1 5:1 20:1 100:1 ..... Scara de marime naturala 1:1 Scari de micsorare 1:2 1:10 1:50 1:200 1:5 1:20 1:100 1:500 ..... Scara principala a desenului se inscrie în indicator, într-o rubrica rezervata. Daca pe un desen exista proiectii executate la scari diferite (vedere, sectiune, detaliu), scarile corespunzatoare acestora se înscriu lânga sau sub notarea proiectiilor respective. Exemplu: A-A C 5:1 1:2 Executarea propriu-zisa a desenului la scara, respecta, în principiu, aceeasi succesiune de faze ca la întocmirea schitei.

Desenul de ansamblu este reprezentarea grafica a unui grup de piese (elemente) legate organic si functional între ele, ce compun o masina, o instalatie sau numai o subgrupa din acestea (subansamblu). Dintr-o astfel de reprezentare trebuie sa reiasa clar forma si pozitia reciproca a pieselor componente, modul si ordinea lor de asamblare, modul de functionare al ansamblului, dimeniunile necesare pentru montare si functionare.
Desenul de ansamblu se întocmeste dupa regulile din STAS 6134-84, având în vedere si regulile de reprezentare privind dispunerea proiectiilor, vederi, sectiuni, rupturi, cotare etc., prezentate în capitolele anterioare.

Reguli de reprezentare :

Desenul de ansamblu trebuie sa cuprinda numarul minim de proiectii necesar pentru definirea clara a pozitiei relative a tuturor componentelor, pentru pozitionarea acestora si pentru înscrierea cotelor aferente.
Pozitia de reprezentare a unui ansamblu coincide cu pozitia sa de functionare.

În cazul asamblarii a doua piese între care exista un joc rezultat din dimensiuni nominale diferite, se reprezinta separate liniile de contur ale fiecarei piese . În cazul asamblarii a doua piese fara joc sau între care exista joc rezultat din abateri limita la aceleasi dimensiuni nominale, suprafata de contact se reprezinta printr-o singura linie de contur, comuna celor doua piese

Daca un plan de sectiune nu contine anumite elemente (suruburi, piulite, stifturi etc.) necesar a fi reprezentate pe proiectia respectiva, acestea pot fi reprezentate rabatute pe planul de sectionare, cu linie-doua puncte subtire.

Fiecare componenta distincta a ansamblului reprezentat pe desen este identificata printr-un numar de pozitie distinct, corespunzator numarului din tabelul de componenta al desenului respectiv.

Fiecare numar se înscrie la extremitatea unei linii de indicatie trasata cu linie continua subtire si terminata cu un punct îngrosat în interiorul conturului componentei pozitionate sau cu o sageata sprijinita pe linia de contur a componentei respective daca ea este reprezentata în sectiune prin înnegrire.

Liniile de indicatie se traseaza înclinat astfel încât sa nu se confunde cu alte linii de contur, linii de axe, elemente de cotare sau hasuri. Se va evita intersectarea acestora cu linii de cota sau ajutatoare. Ele nu trebuie sa fie sistematic paralele sau sa se intersecteze între ele fiind admis sa fie frânte o singura data.

Numerele de pozitie se înscriu cu cifre arabe, având înaltimea egala cu dublul dimensiunii nominale a scrierii utilizate pentru cifrele de cota din desenul respective. Numerele de pozitie se înscriu în afara conturului proiectiei respective, grupate în rânduri verticale si (sau) orizontale, fara a fi subliniate sau încercuite.

Componentele (piese sau subansambluri) se pozitioneaza în proiectia unde ele apar mai clar si pot fi identificate mai usor.

Numerele de pozitie se înscriu în ordine crescatoare, în acelasi sens (trigonometric sau invers trigonometric) pe fiecare proiectie în parte.

Ansamblul de informaţii tehnice date pe un suport de informaţii, conform unor reguli convenite, poartă numele de desen tehnic . Spre deosebire de desenul artistic unde aproape ori ce este permis, desenul tehnic este constrâns de unele reguli generale şi convenţii numite standarde de stat (STAS). Acestea sunt un fel de legi şi au rolul de a asigura unitatea şi uniformitatea în desenul tehnic, adică acelaşi obiect să fie reprezentat în acelaşi fel, ca formă, ca proporţia dimensiunilor, simboluri, de către orice proiectant, pentru a se evita confuziile în interpretarea desenului.

Desenul tehnic se defineşte ca fiind reprezentarea şi determinarea grafică a unor obiecte pe baza unor convenţii, stabilite anterior, astfel încât să se cunoască sau să se poată determina în mod cât mai clar, caracteristicile reale ale obiectului (formă, dimensiuni, etc.) Desenul tehnic, este folosit în diferite domenii în scop ştiinţific cât şi termic, în industrie, în diferite ramuri: construcţii de maşini ,pentru asamblarea diferitelor piese ,în industria chimică , în construcţii , în arhitectură, etc.

Proiectarea mecanică, cunoscută și sub denumirea de proiectare a mașinii sau proiectare inginerească, este procesul de proiectare a pieselor sau componentelor eficiente pentru dispozitive mecanice . Cei care lucrează în domeniul proiectării mecanice studiază modul în care componentele mecanice funcționează în diferite situații pentru a crea un sistem fiabil. Scopul major al proiectării mecanice este de a dezvolta modele de mașini care să fie sigure și aliniate cu specificațiile clientului.

Proiectarea mecanică este importantă deoarece oferă schițe și scheme esențiale pentru sistemele mecanice pe care profesioniștii le folosesc pentru a construi dispozitive sigure și operaționale. Procesele de proiectare mecanică bine definite ajută la creerea produse sau componente care funcționează conform așteptărilor și îndeplinesc așteptările clienților. Proiectarea mecanică este, de asemenea, importantă deoarece urmează un proces stabilit care are ca rezultat crearea unor sisteme de calitate.

Pașii comuni într-un proces de proiectare mecanică sau inginerească includ:

Cercetare: acest pas din procesul de proiectare îi ajută pe designeri să înțeleagă mai multe despre obiectivele și limitările unui proiect.

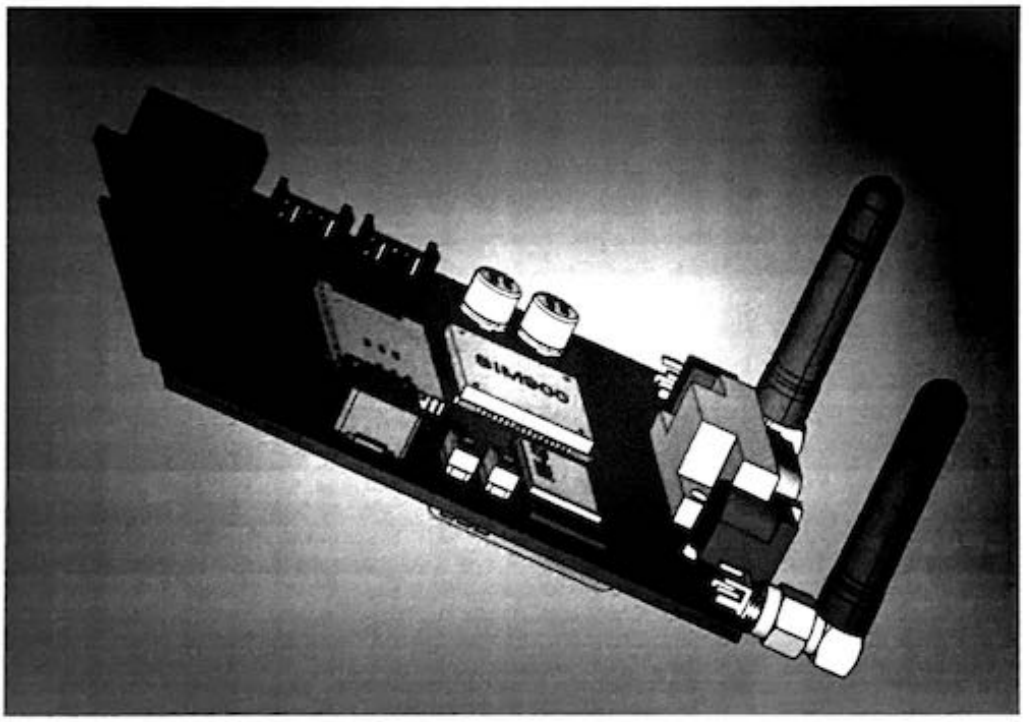
Imaginare: Pasul de imaginație într-un proces de proiectare mecanică este atunci când designerii vizualizează soluții la constrângerile proiectului, urmărind să prevadă probleme sau eșecuri.

Planificare: Odată ce designerii vizualizează toate rezultatele posibile, selectează și detaliază planurile pentru implementarea unui design.

Creare: etapa de creare are ca rezultat dezvoltarea unui prototip al unui dispozitiv sau al unui sistem proiectat.

Testare: Această etapă analizează funcționalitatea, integritatea și fiabilitatea unui dispozitiv.

Îmbunătățirea: Odată ce proiectanții analizează rezultatele unei faze de testare, aceștia pot dezvolta dispozitive cu componente îmbunătățite și pot reporni procesul de proiectare.



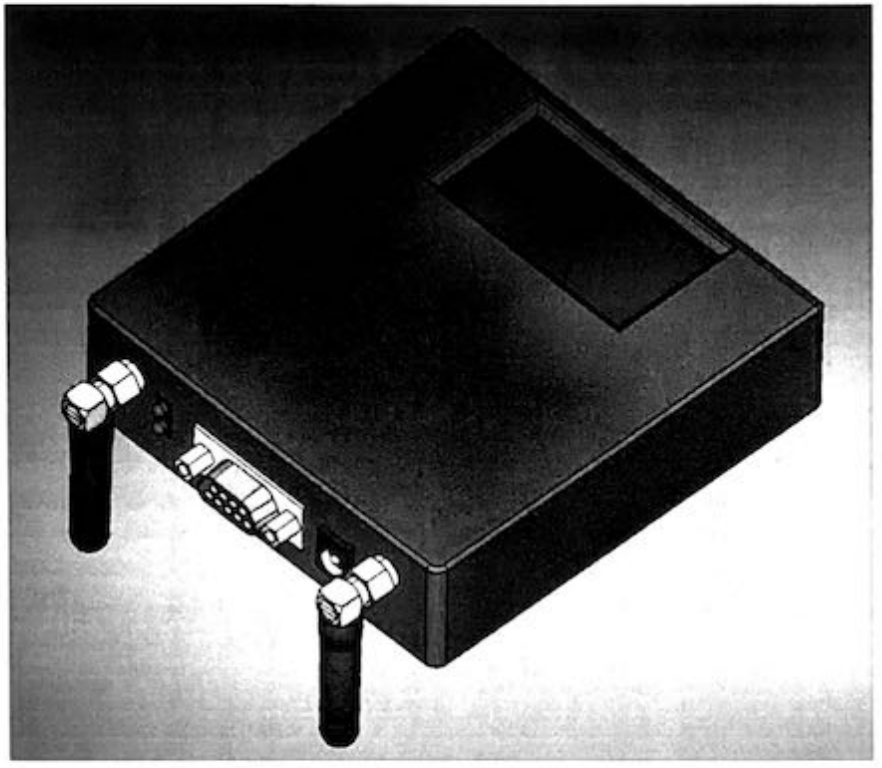
Exemplu de proiectare mecanica si perspectiva componente

Pentru realiarea de produse noi, in cele mai mute cazuri activitatea a inceput cu definirea si realizarea produsului d.p.d.v mecanic, in primul rand.

Produsul a fost realizat intr-un soft CAD 3D si prezentat clientului pentru aprobare. Dupa acesta faza s-au realizat desenele de executie pentru relizarea mecanica efectiva a produsului.

La proiectarea mecanica initiala s-a tinut cont de toate componentele incluse astfel incat varianta finala de produs sa fie la fel sau cat mai aproape de produsul prezentat in varianta 3D initiala si fara modificari.

Produsele sunt realizate plecand de la unele componente standard in unele cazuri (profile de aluminiu extrudat) dar s-a lucrat si in mono-blocuri de aluminiu precum si cupru prin prelucrare pe masina cu comanda numerica CNC. Alte lucrari necesare au fost relizate folosind strunjirea normala ,gaurirea si chiar frezarea manuala, sablarea.

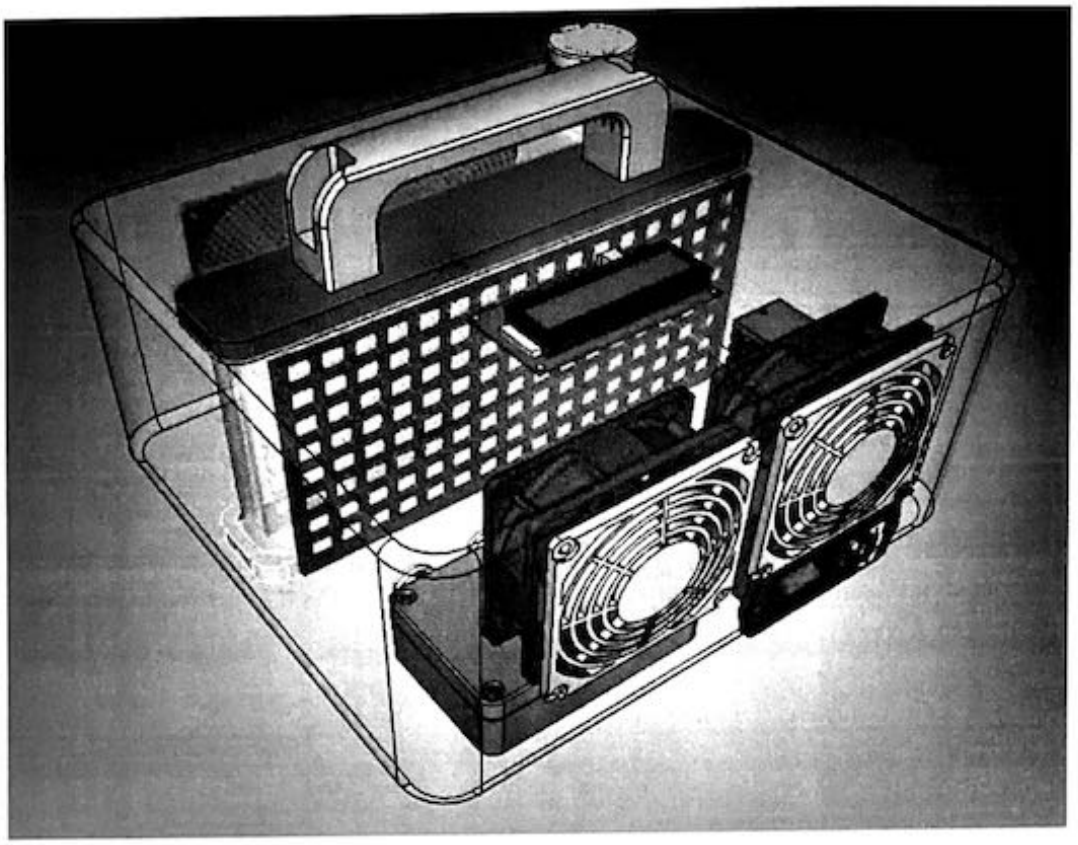


Exemplu de proiectare mecanica carcasa si vizualizare finala produs

f) Cercetare materiale si soltii pentru implementare prototipe

Realizarea produselor a fost realizata folosind mai multe tehnologii disponbile pe piata. In general, produsele au fost incasetate in carcase de aluminiu sau duraluminiu de aviatie.

Alte categorii de produse unde rezistenta mecanica la socuri si vibratii nu a fost considerata o problema majora a fost folosita tehnologia printarii 3D folosind materiale similare ABS-ului. Pe baza acestor print-uri se poate valida design-ul in vederea proiectarii unei matrite pentru masini de injectie dedicate.



Exemplu de proiectare mecanica produs concept

La produsele cu destinatie speciala s-a realizat placarea cu argint (argintare electrochimica) sau chiar aurire (depunere de aur electrochimica).

Au fost realizate incercari de prelucrare mecanica pe masini CNC pentru materiale platice tip ABS , poliamide dar si ceramica prelucrabila (CNC machining for ceramics).

Product Development Agreement

This Product Development Agreement ("Agreement") is made and effective 09.01.2018, by and between Cristian Tanase ("Developer"), with the address at :

Romania , Bucharest ,Sector 3

and Nasui Dorel Vasile, with adress at:

Romania , Bucharest ,Sector 2

1. Definitions

1.1. "Confidential Information" shall mean all material and information supplied by Customer that has or will come into Developer's possession or knowledge of Developer in connection with its performance here under. "Confidential Information" shall also mean all material and information supplied by the Developer to the Customer, including but not limited to hardware & software schematics, source code, and any other materials and technologies developed in conjunction with this Product.

Confidential information does not include information that:

(a) is or becomes public knowledge through no fault of Developer;

(b) Developer knew before Customer disclosed it;

(c) Developer obtains from sources other than Customer who owe no duty of confidentiality to Customer; or

(d) Developer independently develops.

1.2. "Deliverables" shall mean a listing of all items to be delivered to Customer under this Agreement.

1.3. "Derivative Work" shall mean a work that is based on any preexisting works, such as a revision, modification, translation, abridgment, condensation, expansion, or any other form in which such preexisting works may be recast, transformed, or adapted, and that, if prepared without authorization of the owner of the copyright in such preexisting work, would constitute a copyright infringement under the United States Copyright Act.

"Derivative Work" shall mean a work that is based on any preexisting works...

1.4. Developer develops the LINX10DR as defined in EXHIBIT A "Product Specifications" hereafter called the " Product".

1.5. "Open Source License Agreement" means the terms under which Product is licensed in source code form to the general public for use without charge, including without limitation any license agreement that

(a) conditions the use or distribution of any Product that incorporates such Hardware and Software on the disclosure, licensing or distribution of the source code of such program (including such Software) and hardware schematics.

(b) otherwise materially limits a licensee's freedom of action with regard to seeking compensation in connection with licensing or distributing such Product or Software in object code form, including without limitation the GNU General Public License or the GNU Lesser General Public License.

1.5. "Open Source Software" means Software licensed pursuant to the terms of an Open Source License Agreement.

1.6. "Schedule" shall mean the delivery dates for each Deliverable.

1.7. "Specifications" shall mean the specifications for the Product as directed by Customer, together with any modifications that may be agreed to in writing by the parties during the term of this Agreement.

2. Duties and Responsibilities

2.1. Specifications. Customer shall define the Specifications, Deliverables, and Schedules [with input from Developer].

2.2. Development. Developer shall design, develop, and implement the Product in accordance with the Specifications, Deliverables and Schedule.

2.3. Training.

(a) Training Scope. Developer shall provide Customer with 1 month of training OR such training as may reasonably be requested by Customer] on the use of the Product.

(b) Training Dates and Locations. The training will be conducted on such dates and locations as the parties may agree.

2.4.Maintenance. Developer shall perform remedial and preventive maintenance for the Product after its acceptance so that the Product continues to perform in accordance with the technical design. Customer and Developer shall negotiate the terms and price of such

maintenance services, but Developer shall not charge Customer more than $5000 per year for the first two years of maintenance services after acceptance of the Product. Customer shall have the right to terminate such maintenance services at any time upon thirty (30) days written notice to Developer. Developer shall have the right to terminate such maintenance services upon thirty (30) days written notice to Customer if Customer is in material breach of the maintenance agreement between Customer and Developer and remains in material breach for such thirty (30) days.

3. Delivery and Acceptance

3.1. Acceptance Period. Customer will have 30 days following the date of delivery to assess and test the Product.

3.2. Completion. If Developer, in the sole opinion of Customer, delivers the Product in accordance with the Specifications, Deliverables, or Schedule, then Developer shall be deemed to have completed its delivery obligations.

3.3. Rejection. If Developer, in the sole opinion of Customer, fails to deliver the Product in accordance with the Specifications, Deliverables, or Schedule, then:

(a) Notification. Customer shall detail in writing its grounds for rejection; and

(b) Rectification. Developer shall promptly and use its best efforts to correct the Product and upon delivery of such correction, the process of acceptance, completion, and rejection shall restart.

(c) Continued Failure. If the Developer's corrections, in the sole opinion of Customer, fails to deliver the Product in accordance with the Specifications, Deliverables, or Schedule, then Customer may elect to:

(i) terminate the agreement, or

(ii) adjust the Specifications, Deliverables, or Schedule.

4. Changes

4.1. Change Orders. Customer may request changes to the Specifications, Deliverables, or Schedules.

4.2. Additional Time or Expense. If the proposed change will, in the reasonable opinion of Developer, require a delay in delivery of the Product or would result in additional expense, then:

(a) Customer and Developer shall confer; and

(b) Customer may elect to either:

(i) withdraw its proposed change, or

(ii) require Developer to deliver the Product with the proposed change, subject to the delay and/or additional expense.

5. Payment.

5.1. Development Costs. Customer shall pay Developer:

(a)

(b)

5.2. Expenses. Subject to Customer's prior approval.

5.3. Training Costs. The training shall be provided at no additional cost to Customer.

Developer Account EURO :

Developer Account RON :

Code SWIFT : BTRLRO 22
Bank : Banca Transilvania

6. Ownership of Product. Developer agrees that the development of the Product is "work for hire" within the meaning of the Copyright Act of 1976, as amended from time to time, and that the Product shall be the sole property of Customer. Developer assigns to Customer its entire right, title and interest in anything created or developed by Developer for Customer under this Agreement ("Product") including all patents, copyrights, trade secrets and other proprietary rights. This assignment is conditioned upon full payment of the compensation due Developer under this Agreement.

7. Term. This Agreement shall commence upon 01.09.2018 and continue until all of the obligations of the parties have been performed or until earlier terminated as provided herein.

8. Representations

16.2. Notices. Any notice required by this Agreement or given in connection with it, shall be in writing and shall be given to the appropriate party by E-mail, Skype, personal delivery or a recognized overnight delivery service such as FedEx.

If to Developer: Romania , Bucharest ,Sector 3,

If to Customer: Nasui Dorel Vasile

16.3. Entire Agreement. This Agreement contains the entire agreement between the parties and supersedes all understandings and agreements whether written or oral.

16.4. Amendment. No amendment or modification of this Agreement is valid unless in writing, signed by the parties.

16.5. Governing Law. This Agreement is governed by the laws of State of Illinois without regard to any conflict of law principles.

16.6. Force Majeure. Except with regard to payment obligations, either party shall be excused from delays in performing or from failing to perform its obligations under this Agreement to the extent the delays or failures result from causes beyond the reasonable control of the party.

16.7. No Waiver. The waiver or failure of either party to exercise any right provided in this agreement shall not be deemed a waiver of any other right or remedy to which the party may be entitled.

16.8. Severability. If any provision of this Agreement is invalid, illegal, or unenforceable, the remainder of this Agreement will remain in full force and effect.

In Witness whereof, the parties have executed this Agreement as of the date first written above.

Customer: _____ Date: 09.01.2018

Developer: _____ Date: 09.01.2018

EXHIBIT A: Product Specifications

LINX 10DR is a product dedicated to radio communications on specially assigned ISM bands - Industrial Scientific Medical.

LINX 10DR is specially designed to transmit voice as well as digital modulation data. The voice is transmitted in the accepted standard for the radio networks, namely 300 ... 3KHz (expandable) and the data is transmitted raw in a proprietary format / protocol.

ISM frequencies are intended for data traffic without licensing. Both in Europe and the US the 868 MHz and 915 MHz frequency bands are relatively untapped and can be used for secure transmissions over long distances. Interference on these bands is small or

very small. In particular, the 915 Mhz band can be used in the US for both voice and data without any restrictions. The maximum permissible power is 30 dBm (1W).

Depending on the technical solution chosen, radio links are very stable and relatively high speed for both voice and data.

A number of radio type commands are available to be controlled from the application running on Android. Some of these are:

-> LINX 10DR battery status and associated Android device battery status
-> Channel on which communication is made
-> The received signal level of the correspondent
-> PTT button (see note below)
-> Volume
-> Other functions

Note: For greater ergonomics, once the channel and other transmission parameters have been set, the external hardware PTT of LINX 10DR can be used as well.

The LINX 10DR product will be equipped with internal speaker and microphone in a classical radio stile. This will facilitate to operate the LINX 10DR like a regular radio by pressing the PTT button whenever the user needs to.

1. User Interface

The main interface of LINX 10DR is ***Android OS.***
Connecting to LINX 10DR is via Bluetooth or USB.

LINX 10DR is equipped with hardware & firmware to support both connections.

The device's hardware interface is composed of:

1. *The LINX 10DR*
2. *The USB connector*
3. *Signal LED*
4. *JACK connector for external connection of a power supply*
5. *ON / OFF button*
6. *915 MHz radio antenna as well as Bluetooth*

*The software / **Android OS** of the device is composed of:*

Radio Settings Interface:

In the Android interface for Radio Settings you can set the following functions for LINX 10DR:

1. *Emission power level*
2. *Frequency work channel*
3. *VOX level (for microphone attached)*
4. *Scan adjacent channels to watch if there is radio activity on other frequencies*
5. *Set the device to emit a "ring ring" sound when the messages arrive*
6. *FIND ME function - The LINX 10DR device sounds a beep on the Android query to be found if it is not in sight*
7. *LOW Battery Attention - The LINX 10DR device emits periodic audible signals when the battery is low*
8. *Set the device to show the GPS position or not to other radio traffic participants*
9. *Set the "Device ID". This may be a proper name or number.*
10. *LINX 10DR has a dedicated strength - meter that shows the strength of the signal from the correspondents -> Rx Channel Strength*
11. *Spectrum monitor function*
    This function allows you to display the adjacent channels and the power of the signal present for each channel at scan time
12. *Advanced settings -> Only for service mode or special radio settings - advanced users. Maybe to miss the final commercial realise.*

CHAT Interface:

The Android CHAT interface has the following features:

1. Display the LINX 10DR currently participating in CHAT
2. Shows the LINX 10DR devices present in the radio network but not participating. Can be invited by message by pressing the plus button "+"
3. By pressing the minus "-" button in the interface some devices can be removed from the list "participating" (master mode)
4. Scan for friends passes the LINX 10DR device in scan / query mode to see if other LINX 10DR active devices are available

VOICE Interface:

The VOICE Android interface is available in the following format:

1. Display the LINX 10DR currently participating in CHAT at the current time in the GRUP
2. Shows the LINX 10DR devices present in the radio network but not participating. Can be invited by message by pressing the plus button "+".

3. You can make PRIVATE CALL to a single person in the group by drag & drop for the desired person in the PRIVATE PTT TO.
4. MAIN PTT key to operate the main PTT.

Using MAIN PTT:

MAIN PTT gets the functionality of the two buttons above PTT GROUP and PTT Private.
If PRIVATE PTT has the green box active then MAIN PTT will make a private call only to the person in the PRIVATE PTT TO list. In the case of the above photo you will there is private conversation encrypted only with "BRENDA".

If "PTT GROUP" has the green box active then MAIN PTT will make a private call only to people in the RADIO ID IN GROUP list. All people in GROUP will be able to listen to the broadcast

The **EMR** button is only used in emergency situations.Its action has the effect of sending it to all GROUP participants GPS position and alarm status. Moreover, the radio will act as well BEACON.

EXHIBIT B: Deliverables

The Developer delivers to the customer the following:

1. All the software developed for the product
2. All the firmware developed especially for this product
3. All the manufacturing blueprints, manufacturing techniques , etc.
4. All associated electrical schematics and mechanical schematics developed for the product.

The above mentioned documents stored in electronic format or printed are in fact the propriety of the customer.

EXHIBIT C: Schedules

The product is developed between September 2018 and March 2020 as follows:

1. Between September 2018 and October 2018 a study for ISM chip used is conducted.
2. October2018 , the same study takes place but for the micro controller/microprocessor to be used in the design.
3. October 2018 → November 2018 the additional components are selected. We refer here to filters, antennas, RF Switches etc.
4. November 2018 → December 2018 is the first PCB design period. The PCB is in "development" board stage, used to develop firmware and test the design but is not in the final shape.
5. December 2018 – PCB Manufacturing in Europe or Asia
6. January 2019 first half – The realization of the prototype
7. January 2019 second half – The hardware interfaces for uC is tested
8. February 2019 → March 2019 – Development of the firmware for communication protocol
9. March 2019 → May 2019 – Development of the protocol between LINX10DR and Android OS
10. May 2019 → June 2019 – Final implementation of the protocol
11. June 2019 → August 2019 – Final implementation for communication between LINX10DR and Android for USB and Bluetooth.

12. *September 2019 – Overall system testing*
13. *October 2019→ November 2019 – Final hardware design for the final product.*
14. *November 2019 – Final PCB manufacturing in Europe or Asia*
15. *December 2019 – Testing for the final product*
16. *January 2020 → February 2019 – Manufacturing process*

Product Development Agreement Addendum

THIS AMENDING AGREEMENT dated 03.01.2020 (MM/DD/YYYY)

BETWEEN:

Nasui Dorel Vasile

– AND-

Tanase Cristian

Background

A. Dorel Nasui Vasile and Tanase Cristian (the "Parties") entered into the "Product Development Agreement " dated 01.09.2018 (MM/DD/YYYY), for the purpose of developing LINX10DR product.

B. The Parties wish to amend the Product Development Agreement on the terms and conditions set forth in this Amending Product Development Agreement (the "Addendum").

C. This Addendum is the first amendment to the Product Development Agreement.

IN CONSIDERATION OF the Parties agreeing to amend their obligations in the existing Product Development Agreement, and other valuable consideration, the receipt and sufficiency of which is hereby acknowledged, the Parties agree to keep, perform, and fulfil the promises, conditions and agreements below:

Amendments

1. The Product Development Agreement is amended as follows:

a. Cristian Tanase as Developer, will further develop hardware, software and firmware based products as per Customer request.

b. Tanase Cristian as Developer will provide technical assistance to the Client and fulfill client requests.

No Other Change

2. Except as otherwise expressly provided in this Addendum, all of the terms and conditions of the Product Development Agreement remain unchanged.

Miscellaneous Terms

3. Capitalised terms not otherwise defined in this Addendum will have the meanings ascribed to them in the Product Development Agreement. Headings are inserted for the convenience of the parties only and are not to be considered when interpreting this Addendum.

Governing Law

4. Subject to the terms of the Product Development Agreement, it is the intention of the Parties that this Addendum, and all suits and special proceedings under this Addendum, be construed in accordance with and governed, to the exclusion of the law of any other forum, by the laws of United States.

IN WITNESS WHEREOF the Parties have duly affixed their signatures on 01.03.2020 (MM/DD/YYYY)

Name: Tanase Cristian

Address: Address:Aleea Ion Agarbiceanu 3-11, Bucharest S3, Romania

Name:Nasui Dorel Vasile

Address: Strada Trei Scaune nr 5, Bucharest S2 , Romania