

1.1. МИКРОКОМПЬЮТЕР, ТҮҮНИЙ ХЭРЭГЛЭЭ

Микрокомпьютер гэдэг бол программ гэж нэрлэгддэг зааварчлага инструкцүүдийн багцаар удирдагдан тооцоололт болон электроник, мэдээллийн харилцаа холбооны үйлдлүүдийг хийж гүйцэтгэдэг машин юм. **Программ** нь компьютерийн санах ой дотор оршиж, компьютерийн дотоод электроник төхөөрөмжүүдээр сэргээгдэн боловсруулагддаг. Программын үр дүн **гаралтын төхөөрөмжүүдээр** дамжин тухайлбал видео дэлгэц болох монитор, эсвэл хэвлэгч төхөөрөмжөөр дамжин хэрэглэгч хүнд хүрдэг. Компьютер нь олон арван сая үйлдлийг асар өндөр хурдтайгаар туйлын нягт нямбай, алдаа мадаггүй хийж гүйцэтгэдэг бөгөөд **Computing** гэдэг тооцоолох гэсэн үгнээс гаралтай нэр томъёо юм. Ердийн нэг калкулятор ч гэсэн компьютерийн нэг хэлбэр мөн. Сампин хэдийгээр тооцоолуур мөн ч түүн дотор хадгалагддаг программ хангамж гэж байхгүй билээ. Тиймээс компьютер гэж үзэж болохгүй.

Хүн компьютерийг маш олон зүйлд хэрэглэдэг. Бизнес болон санхүүгийн үйл ажиллагаанд компьютер нь эд барааны худалдаа гүйлгээнээс эхлээд үйлчлүүлэгчийн банкны дансны гүйлгээг хүртэл удирддаг. Гэрт бол жижигхэн компьютер хаа сайгүй бий. Өнгөт телевизор зөвхөн телевизор биш юм. Суваг хайх, олсон сувгаа хадгалах, алсын удирдлагатай холбогдож телевизорыг удирдах зэрэг үйлдлүүдийг хийж гүйцэтгэх бичил компьютер түүн дотор байдаг. Гар утас гэдэг бол маш хүчтэй нарийн үйлдэлтэй компьютер билээ. Халаасны компьютер (**Pocket PC**) хэмээх маш жижигхэн хэдий ч ширээний компьютерийн хийх үйлдлүүдийн ихэнхийг гүйцэтгэдэг компьютер хэрэглээнд гараанд нэлээд хэдэн жил болжээ.

Автомашинд л гэхэд түлшний зохицуулалт, асаалтын системийг удирддаг, автомат агааржуулалт, хөргөлт, халаалт, хулгайн дохиололийг удирддаг гэх мэт хэд хэдэн компьютер ажилладаг.

Компьютер хүнийг хөгжөөн баясадаг хөгжим, дүрс тоглуулагч, тэдгээрийг боловсруулж найруулахаас гадна хүний зурж чадахгүй дүрсийг график, анимациар үүсгэж байгаль дээр үүсдэггүй дуу чимээг ч үүсгэж гаргадаг.

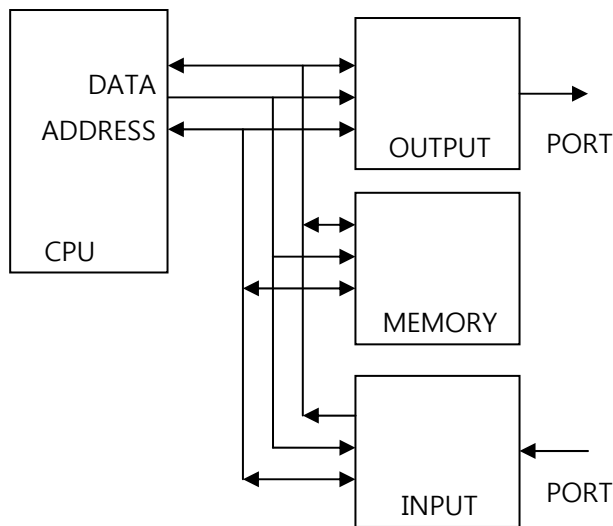
Бидний хэрэглэдэг телевизор шиг дэлгэцтэй, хажуудаа процессор гэгдэх дөрвөлжин хайрцагтай зүйл бол (хар, саарал ч бас байдаг) персональ буюу нэгэн төрлийн **микрокомпьютер** юм. Маш олон төрлийн компьютерийн нэг хэлбэр гэж ойлгож болно. Харин энд бидний судлах **микроконтроллер** нь микрокомпьютерийн хамгийн жижиг төрөл бөгөөд компьютерт байвал зохих бүх хэсгүүд буюу төхөөрөмжүүд түүний дотоод бүтцэд байдаг.

Аливаа автомат төхөөрөмжүүдийн гол тархи нь микрокомпьютер байдаг. Микродолгионы зуухын, угаалгын машины, систем хөгжмийн ажиллагааг удирдахад микрокомпьютер, тэр дундаа микроконтроллер оролцдог. Хамгийн өргөн хэрэглэгддэг телевизорын суваг санах, солих, дуу болон дүрсийн тод бүдгийг удирдах зэрэг үйлдлүүдийг бяцхан компьютер зохицуулж байдаг гэдгийг өмнө дурдсан билээ. Түүнчлэн роботын үйл ажиллагааг компьютер удирдах нь ойлгомжтой. Хэрвээ та ямар нэгэн автомат төхөөрөмж, робот зохион бүтээхийг хүсэж байгаа бол микроконтроллерийг программчилж сурах хэрэгтэй. Мөн үүний өмнө та электроникийн тодорхой мэдлэгтэй байхаас гадна программчлалын тухай алгоритм зохиох түвшний мэдлэгтэй байвал тун сайн. Миний бичсэн “Электроникийн үндэс” номыг авч үзэж болно. Ялангуяа микроконтроллерийг ашиглахад тоон электроникийн үндсэн мэдлэг зайлшгүй шаардлагатай юм.

1.2. МИКРОКОМПЬЮТЕРИЙН БҮТЭЦ

Компьютерийн эд ангийн хэсэг ба түүний цахилгаан элементүүдийг хатуу тоноглол (**Hardware**) гэдэг. Энэ нь:

- Мэдээлэл болон программын зааварчлага инструкцүүдийг хадгалах Санах Ой (**Memory**)
- Санах ой дахь мэдээллийг, мөн тэнд байгаа инструкцүүдийн дагуу боловсруулж, гарсан үр дүн болох мэдээллүүдийг бусад төхөөрөмжүүд рүү дамжуулах гол үүргийг гүйцэтгэдэг Төв Процессорын төхөөрөмж (**Central Processing Unit – CPU**)
- Хулгана, товчлуур, сканер гэх мэт **оролтын** төхөөрөмжүүдийн удирдлага (**Input devices control**)
- Дэлгэц, хэвлэгч гэх мэт **гаралтын** төхөөрөмжүүдийн удирдлага (**Output devices control**) зэргээс бүрдэнэ.



Зураг 1.1. Компьютерийн бүтэц

Харин эдгээр төхөөрөмжүүд хоорондоо холбох шугамуудаар (**Buses**) холбогддог.

Хүнээр бол Программ хангамж нь оюун ухаан, мэдээлэл бол мэдрэлийн судсаар гүйж буй сэрэл, CPU болон MEMORY бол уураг тархи, мэдрэх эрхтэнүүд бол оролтын төхөөрөмж, хөл гар, ярианы эрхтэнүүд бол гаралтын төхөөрөмж болох юм. Мэдрэлийн үй түмэн судсууд бол BUS буюу төхөөрөмжүүдийг хооронд нь холбогч шугамууд юм.

Холбогч шугамууд маш олон зэрэгцээ цахилгаан дамжуулагч утаснуудаас бүрдэг бөгөөд гүйцэтгэж байгаа үүргээр нь 3 үндсэн бүлэгт хуваадаг. Үүнд:

а. Мэдээллийн шугам – DATA BUS. Энэ шугамаар тоон мэдээлэл процессороос гадагш болон процессор луу гэсэн 2 чиглэлээр дамжих бөгөөд мэдээллийг хамгийн багадаа байтаар буюу 8 бит мэдээлэл дамжуулахаар хийгдсэн байдаг. Хамгийн сүүлийн үеийн Пентьюм процессорууд 64 бит буюу нэгэн зэрэг 8 байт мэдээлэл дамжуулдаг мэдээллийн шугамтай байдаг.

б. Хаягийн шугам – ADDRESS BUS. Энэ шугамаар хаягийн мэдээлэл зөвхөн процессороос бусад төхөөрөмжүүд рүү очно. Энэ шугамаар процессор өөрөө ямар хаягнаас мэдээлэл авах болон ямар хаяг руу мэдээлэл дамжуулахаа мэдэгддэг.

в. Удирдлагын шугам – CONTROL BUS. Энэ шугамаар процессор мэдээллийг төхөөрөмжүүдээс авахыг болон төхөөрөмжүүд рүү өгөхийг хүсэх, төхөөрөмжүүдэд өөрөө завтай эсэхээ мэдэгдэх зэрэг удирдлагын мэдээллүүдийг дамжуулдаг. Харин төхөөрөмжүүд энэ шугамаар процессорт өөрсдийн бэлэн байдал эсвэл бэлэн бус байдал, мөн процессорыг яаралтай дуудаж (**Interrupt**) өөртөө хандуулах зэрэг удирдлагын дохиог дамжуулдаг. Иймээс удирдлагын шугамын зарим нь процессороос төхөөрөмжүүд рүү, зарим нь төхөөрөмжүүдээс процессор луу чиглэсэн нэг чигтэй шугамууд байна.

Санах ойд уг компьютерийн хийж гүйцэтгэх үйлдлүүдийн зааварчлага инструкцүүдийг хийх дараалалын дагуу байрлуулсан программ хангамж байхаас гадна уг программ хангамжийн хэрэглэж боловсруулах мэдээллүүд хамт байна. Түүнчлэн программ хангамж гүйцэтгэгдээд гарсан үр дүнг хадгалах хэсэг бас шаардлагатай болно. Программ хангамжийг **Software** – Зөөлөн тоноглол гэж орчуулж болно. Микроконтроллерийн программ хангамжийг хадгалж байдаг санах ойг **программын санах ой (Code memory)** гэх бөгөөд тооцоолон бодох үйл ажиллагааг явуулж байхдаа түр зуур мэдээлэл хадгалах санах ойг хэрэглэдэг бөгөөд үүнийг шуурхай санах ой (**Random Access Memory – RAM**) гэж нэрлэдэг.

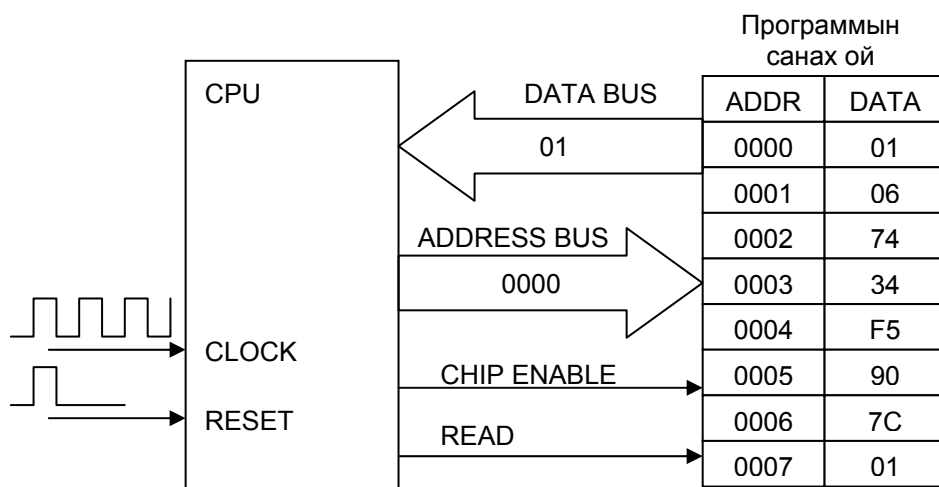
Оролт, гаралтын төхөөрөмжүүдийн цаад талд порт (**PORT**) гэгдэх мэдээлэл оруулах гаргах шугамууд байдаг. Эдгээр шугамууд буюу портуудаар гаднын төхөөрөмжүүдтэй холбогдон ажиллах болно. Харин портуудыг ажиллах зарчимаар нь зэрэгцээ порт (**Parallel Port**), цуваа порт (**Serial Port**) гэх мэтчилэн ангилдаг.

Эдгээр төхөөрөмжүүдийг тус тусад нь олон хавтан дээр угсарсан байвал **олон хавтант компьютер** буюу **Multiboard Computer** гэж нэрлэнэ. Харин нэг хавтан дээр бүгдийг нь угсарсан байвал **Singleboard Computer** гэнэ. Түүнчлэн энэ үндсэн төхөөрөмжүүдийг нэг хагас дамжуулагч кристалл дээр багтаан суурилуулж нэг микросхем болгон үйлдвэрлэн гаргасныг **нэг кристаллт компьютер** буюу **Single Chip Computer** гэдэг байна. Singleboard Computer болон Single Chip микрокомпьютерийг **Микроконтроллер** гэж нэрлэдэг. Учир нь өмнөх сэдэвт дурдсанчлан ийм төрлийн компьютерүүдийг автомат төхөөрөмжүүдийн удирдлагад голчлон хэрэглэдэг учир ингэж нэрлэсэн байна. Бид микроконтроллерийн ажиллагаатай танилцахдаа **Atmel** фирмийн нэг кристаллт **AT89C2051** микроконтроллерийн ажилагааг судлах замаар сурах болно. Энэхүү микроконтроллер нь дотроо 2Кбайт программын санах ой, 128байт шуурхай санах ой, оролт гаралтын программчлагддаг 15 шугамууд буюу портуудтайгаас гадна мэдээллийг цуваа кодоор дамжуулах, хүлээн авах оролт гаралтын порттой хамгийн өргөн хэрэглэгддэг төхөөрөмж юм. Түүнчлэн дотроо тус бүр нь 16 битийн 2 тоолуур/таймертай ба аналог хүчдэлийг харьцуулагч компаратор суурилуулагдсанаараа бусад контроллеруудаас ялгаатай юм. Түүний программын санах ойг тусгай программчилдаг төхөөрөмж ашиглан программ суурилуулдаг ба 1000 орчим удаа бичиж, арилгаж болдог байна.

1.3. МИКРОКОНТРОЛЛЕРИЙН ПРОГРАММ ХАНГАМЖ

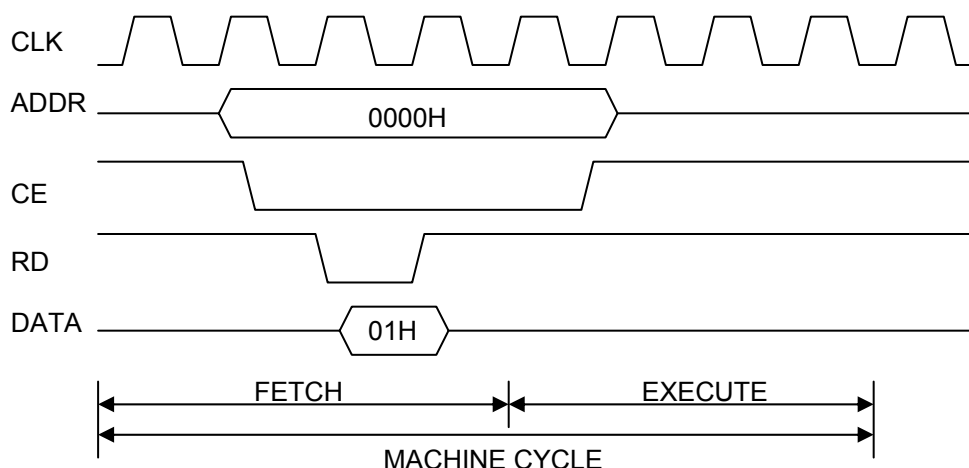
Микроконтроллер нь бусад компьютерүүдийн адил дээд түвшний хэл дээр ажиллаж болох боловч энэ тохиолдолд дээд түвшний хэлээс микроконтроллерийн өөрийнх нь машин код руу хөрвүүлдэг хөрвүүлэгч программ хангамжийг уг микроконтроллерт суурилуулсан байх шаардлагатай болдог. Ийм микроконтроллерийн систем нь программчлан ажиллуулахад хялбархан байдаг боловч түүний нарийн ажиллагааны тухай судалж ойлгоход хүндрэлтэй байдаг. Иймд техникийн түвшинд түүний ажиллагааг маш сайн ойлгоё гэвэл доод түвшний хэлийг судлах шаардлагатай юм. Ийм микроконтроллерийн системд элдэв дээд түвшний хэлний хөрвүүлэгч болон бусад нарийн төвөгтэй дэлгэц, товчлуур зэрэг төхөөрөмжүүд шаардлагагүй тун энгийн хялбар бүтэцтэй болно. Мөн дээд түвшний хэлээс машин код руу хөрвүүлэгдсэн программ нь шууд доод түвшний хэл дээр бичсэн программаас нарийн бүтэцтэй болж их санах ой шаардахаас гадна ажиллахдаа мөн удаан ажиллах болдог.

Аливаа микроконтроллерийг программчлан ажиллуулахад түүний машин код буюу хамгийн анхан шатны программ хангамжийн тухай ойлголттой байх шаардлагатай. Машин код гэдэг нь программын санах ойд байрласан 16тын тоонууд буюу байтуудын цуврал юм. Эдгээр байтууд нь тус бүр нэг нэг зааварчлага инструкцүүд болно. Аливаа микроконтроллер нь тэжээлийн хүчдэлд залгагдан ажиллаж эхлэхдээ (**Reset**) анхны төлөвөөсөө эхлэх бөгөөд программын санах ойн хамгийн эхний буюу 0-р байтыг уншиж, уг байт нь ямар инструкц (зааварчлага команд) болохыг тайлж, түүний дагуу үйлдэл хийснээр ажиллаж эхлэнэ. Ингээд дараа дараагийн инструкцүүдийг гүйцэтгэснээр программ ажилладаг. Бүх инструкцүүд ажиллаад программ дахин давтагдахаар хийгдээгүй бол программ дуусах бөгөөд үүнийг **Program Terminated** гэнэ. Харин программ хэзээ ч төгсөхгүйгээр эхнээсээ болон хаа нэг хэсгээсээ дахин ажиллахаар зохиогдсон байвал үүнийг программын цикл (**Cycle** эсвэл **Loop**) гэнэ. Программ нь ажиллаж байх явцад тухайн программыг түр таслаад өөр программ ажиллаж болно. Үүнийг **Program Interrupted** гэх бөгөөд таслаад ажиллаж эхэлсэн программ дуусмагц өмнөх тасарсан программ тасарсан газраасаа үргэлжлэн ажилладаг. Тасалдсан программ руу буцаж очихийг **Return** гэнэ. Оролт гаралтын төхөөрөмжүүдээс дохио ирж программыг тасалдуулсан бол **Hardware Interrupt**, харин программ өөрөө өөр программ руу шилжээд буцаж ирж байвал дэд программ дуудах (**Call Subroutine**) гэж ярьдаг. Нэг микроконтроллерт хэд хэдэн Hardware Interrupt байж болно. Тэдгээр нь өөр өөр түвшинтэй байдаг. Өөрөөр хэлбэл хоёр тасалдалт тохиолдолоор нэгэн зэрэг ирвэл микроконтроллер аль нэгийг нь эхэлж гүйцэтгээд дараа нь нөгөөхийг нь гүйцэтгэх шаардлагатай юм. Тэгвэл аль нь түрүүлж ажиллахыг зааж өгч болдог. Үүнийг **Interrupt Priority** гэж нэрлэнэ.



Зураг 1.2. Микропроцессор ба программын санах ой

Микропроцессор хаягийн шугамаар (Address Bus) анхны 0000h хаягийг гаргаад, Chip Enable (**CE**) болон Read (**RD**) удирдлагын дохионуудыг гаргавал программын санах ой CE дохиогоор сонгогдон 0-р хаяг дээрх мэдээллээ мэдээллийн шугамаар (Data Bus) гаргана. CPU нь RD дохиог гаргасан учраас уг мэдээллийг өөрийн доторхи инструкцийг тайлах регистрт (Instruction Decoder) оруулж тайлна. Ингээд уг инструкцийн хий гэсэн үйлдлийг гүйцэтгэнэ. Үүний дараа хаягийн мэдээллийг хадгалж байдаг Program Counter (PC) регистрийн утгыг нэгээр нэмэгдүүлэн дараагийн 0001H хаяг дээрх мэдээллийг уншиж дараагийн үйлдлийг гүйцэтгэнэ. Нэг инструкцийг уншаад, түүнийг гүйцэтгэж дуусах хүртэлх хугацааг **нэг машин цикл** гэдэг. Үүнийг хугацааны диаграммаар илэрхийлнэ.



Зураг 1.3. Нэг машин циклийн хугацааны диаграмм

Аливаа микропроцессор үйлдлээ Clock (**CLK**) дохионы тактаар гүйцэтгэдэг. Хэрэв энэ дохио байхгүй бол процессорын зүрх нь зогсоно гэсэн үг. Процессор анхны хаягаа CE дохиотой гармагц программын санах ой сонгогдоно. Үүний дараа RD буюу унших дохио процессороос гармагц программын санах ой мэдээллээ гаргана. Ингэж санах ойгоос инструкцийг уншиж авах ажиллагааг **Fetch** гэдэг. Уг инструкцийг гүйцэтгэх ажиллагааг **Execute** гэнэ. Энэ хоёр ажиллагаа нь нэг машин цикл болдог. Нэг байт нь 00H-FFH хүртэл 256 утга авдаг учраас 8 битийн микроконтроллер нь 256 ялгаатай инструкцтэй байх боломжтой юм. Эдгээр инструкцүүдийг ашиглан микроконтроллерийн программийг бий болгоно. Үнэндээ энэ олон байтуудын учрыг олж программ болгоно гэдэг бол ямар ч мэдлэгтэй хүнд хэцүү байх нь ойлгомжтой. Иймээс машин кодыг хүний ойлгодог программчлалын хэл хэлбэрт бичиж болдог. Үүнийг **Ассемблерийн хэл** гэнэ. Ассемблерийн хэл дээр бичсэн программыг Ассемблер хөрвүүлэгчээр хөрвүүлээд **Object** файл болгоно. Object файлийг 16-тын тоон файл **HEX** болгон хувиргагч программын тусламжтай хувиргаад микроконтроллерийн программын санах ойд суурилуулах машин код бэлэн болно. Бид энэ номоор Intel фирмийн 8031 микроконтроллерийн ассемблерийн хэл дээр хэрхэн программ бичиж AT89C2051 микроконтроллерт суурилуулан ажиллуулахад суралцах болно.

1.3.1. Ассемблер гэж юу вэ?

Ассемблерийн хэл гэдэг бол хүн хялбар ойлгох хэлбэрээр бичсэн машин кодууд буюу инструкцууд гэж хэлж болно. Өмнө бид микроконтроллерийн программ гэдэг нь инструкцүүдийн цуврал гэдгийг ойлгосон. Тэгвэл чухам ямар үйлдэл хийдэг инструкцүүд байх вэ гэсэн асуулт зүй ёсоор гарч ирнэ. Инструкцүүдийг: **а.** Дэс дараалан гүйцэтгэгдэж байсан программын дундаас өөр тийшээ үсрэх, **б.** Дэс дараалан дуудах болон түүнээс буцах гэх мэт машины үйлдлийг удирдах инструкцүүд, **в.** Арифметик болон Логик үйлдлүүд хийж гүйцэтгэх инструкцүүд, **г.** Мэдээллийг санах ой, дотоод региструуд болон оролт гаралтын төхөөрөмжүүд хооронд зөөх, шилжүүлэх инструкцүүд гэх мэтчилэн бүлэглэж болно. Дээрх үйлдлүүд бүгд нийлээд тодорхой үйл явцыг удирдах программ хангамж болно гэсэн үг юм. Ассемблерийн хэл сурна гэдэг бол инструкцүүдийг судлана гэсэн үг юм. Инструкцүүдээр программ бичнэ. Программын бүтцийг тодорхой жишээн дээр үзвэл ойлгоход дөхөмтэй байх болно.

Дараахь жишээ нь микроконтроллерийн программыг машин код болон ассемблерийн хэл дээр бичсэн хоёр хувилбараар харуулав.

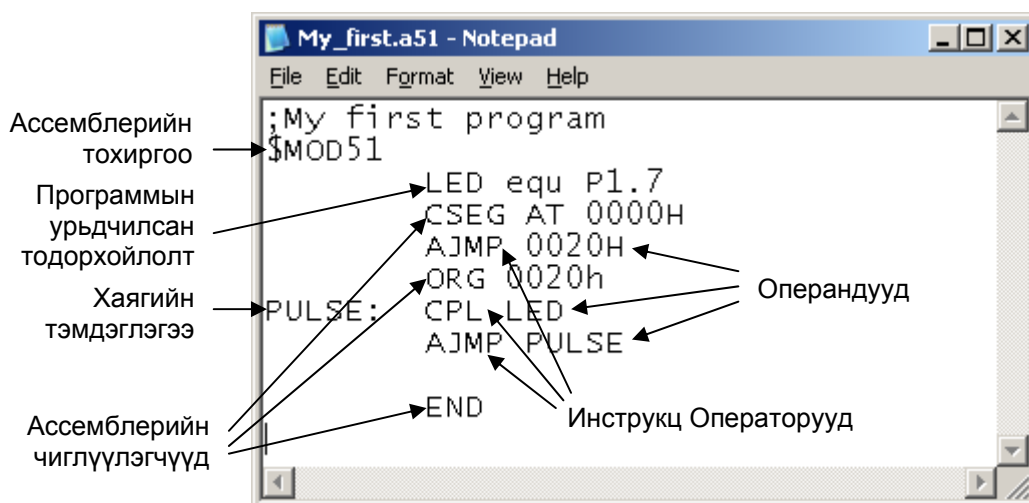
0000H: 0106	AJMP 06H	;Программын гүйцэтгэл 0006H ;хаягт шилжинэ.
0002H: 7434	MOV A,#34H	;А регистрт 34H тоог хийнэ.
0004H: F590	MOV P1,A	;Порт 1 рүү А-д буй тоог гаргана.
0006H: 7C01	MOV R4,#1H	;Регистр 4-т 1H тоог хийнэ.
0008H: EC	MOV A,R4	;А-д регистр 4-т байгаа тоог хийнэ.
0009H: F590	MOV P1,A	;Порт 1 рүү А-д буй тоог гаргана.
000BH: 23	RL A	;А регистрийн утгыг баруун тийш нэг ;битээр шилжүүлнэ.
000CH: FC	MOV R4,A	;Регистр 4-т А-д буй тоог хийнэ.
000DH: 0108	AJMP 08H	;Программын гүйцэтгэл 0008H ;хаягт шилжинэ.

Машин код

Мнемоник код

Программын мөрийн тайлбар зүүлт

Машин кодоос харахад зарим инструкцүүд 2 байтаас бүтэж байна. Ийм төрлийн инструкцүүд гүйцэтгэгдэхдээ 2 машин цикл хугацаа шаардана. Харин нэг байт инструкцүүд нэг машин циклд гүйцэтгэгдэнэ. Программаас 2 байт инструкцииин 2 дахь байт нь уг үйлдэлд оролцох мэдээлэл болох нь харагдаж байна. Энэ мэдээллийг **Операнд** буюу үйлдэлд оролцогч гэж нэрлэдэг. Харин MOV, AJMP гэсэн инструкцииин эхний хэсгийг **Оператор** гэнэ. Бид ассемблерийн хэл дээр программ бичихдээ зөвхөн мнемоник кодыг бичнэ. Программаа хүнд улам ойлгомжтой болгохын тулд тайлбар зүүлтийг хэрэглэдэг. Тайлбар зүүлтийг хийхдээ уг үйлдлийг ямар зорилгоор хийж байгааг сайн тайлбарлах хэрэгтэй. Дээрх жишээн дээр бол та бүгд анх удаа ассемблерийн тухай үзэж байгаа учир мнемоник нь яг ямар үйлдэл хийж байгааг тайлбарлаж бичив. Цаашид бодит программ дээр бол мнемоникийг тайлбарлах биш ямар зорилготой уг үйлдлийг хийж байгааг тайлбарлах нь зүйтэй.



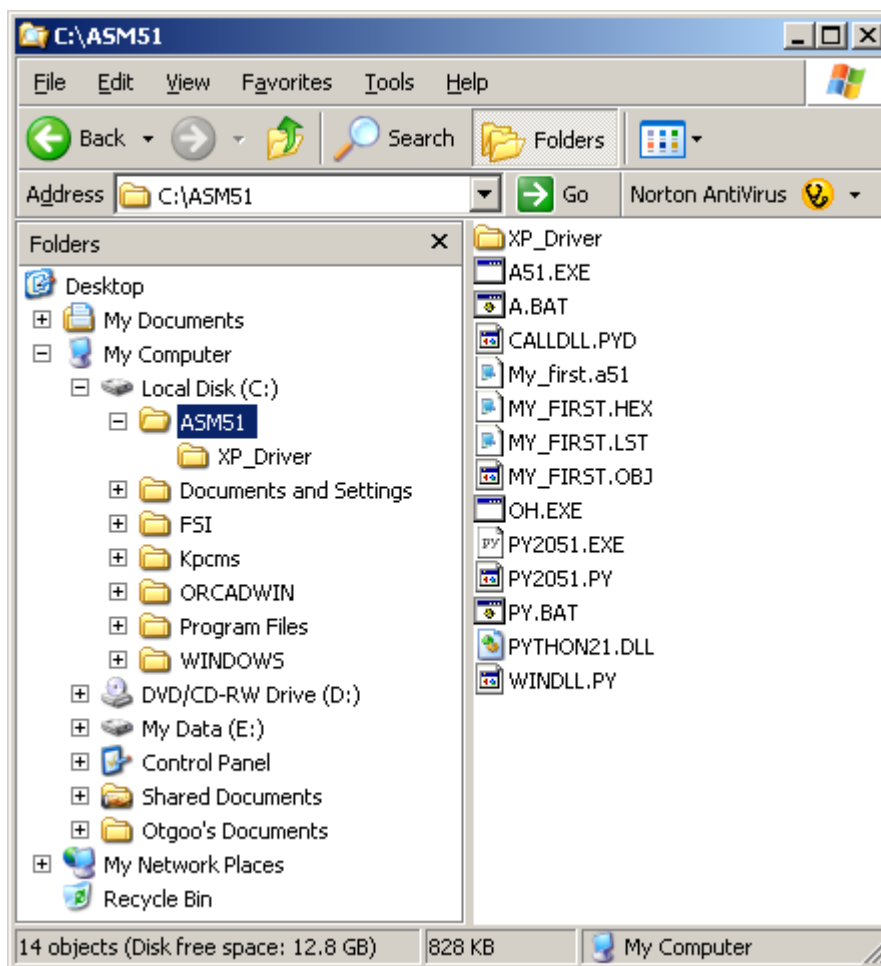
Зураг 1.4. Ассемблерийн хэл дээрх программын бүтэц

Ассемблерийн хэл дээрх программыг бичихдээ дурын текст боловсруулах программыг ашиглаж болно. Зураг 1.4-т үзүүлсэн жишээн дээр Notepad текст мэдээлэл боловсруулах программыг ашиглаж байна. Бичиж буй программынхаа нэрийг өгч хадгалахдаа “a51” гэсэн өргөтгөл нэрийг ашиглах хэрэгтэй. Ийм өргөтгөлтэй файлийг ассемблерийн хөрвүүлэгч ойлгож хөрвүүлнэ. Ассемблерийн программын бүтцийг тайлбарлая.

- Ассемблерийн тохиргоо нь ассемблерийн хөрвүүлэгч ямар горимд ажиллахыг зааж өгдөг. Энэ жишээнд ассемблерийг 8051 микроконтроллерийн горимд ажиллахыг зааж байна. Бид AT89C2051 микроконтроллерийг ашиглах учир ийм тохиргоонууд онц шаардлага байхгүй. Хэрэв 8032, 8052 мэтийн нэмэгдэл төхөөрөмжүүдтэй контроллерийг ашиглах гэж байгаа бол зайлшгүй тохиргоог зааж өгөх хэрэгтэй. Тэгэхгүй бол нэмэлт төхөөрөмжүүдтэй харьцах боломжийг хаана. Тухайлбал эдгээр контроллерууд 8031, 8051 контроллеруудаас илүү таймер тоолууртай байх жишээтэй.
- Программын урьдчилсан тодорхойлолтыг томоохон программуудад хэрэглэвэл программ бичих ажиллагааг нэлээд хөнгөвчилж өгдөг. Тухайлбал дээрх жишээн дээр микроконтроллерийн 1-р портын 7-р битээр нэгэн гэрэлтэгч диодыг асааж унтраахын тулд уг портын гаралтыг LED гэж нэрлэсэн байна. Ингэж тодорхойлоогүй тохиолдолд программ дотор LED-ийг асаахын тулд p1.7 гэж хандах хэрэгэй болно. Их том программ бичиж байвал үргэлж p1.7 гэж бичих нь зохимжгүй юм. Хэрэв зохион бүтээлтийн шаардлагаар p3.7-г гэрэлтэгч диодыг асааж унтраахаар соливол программ дотор байгаа бүх p1.7-г p3.7 болгох ажил ундрана. Дээрх тохиолдолд зөвхөн тодорхойлолтын ганц утгыг солиход хангалттай.
- Хаягийн тэмдэглэгээг бид программ дотроо өргөн ашиглах бөгөөд урьдчилан тодорхойлох шаардлага байхгүй. Харин ижил нэртэй хоёр хаягийн тэмдэглэгээ байж болохгүй. Хаягийн тэмдэглэгээг аль болох залхууралгүйгээр ойлгомжтой үг үсгээр тэмдэглэвэл зохино. Залхуурч маш богино тэмдэглэгээнүүд үүсгэсэнээр олон тэмдэглэгээ ойролцоо утгатай болж өөрсдөө толгой эргэх нь элбэг.
- Бидний цаашид бичих программуудын хамгийн гол хэсэг нь инструкцүүд болон операндууд байх болно. Бид 8031 бүлийн микроконтроллерийн бүх инструкцүүдийг үзэх болно. Бүх инструкцүүдийг бүрэн судлаагүйгээс программ бичих ажиллагаа хүндрэхээс гадна зарим олюутнууд маш тойруу замаар хамгийн хялбар үйлдлийг хийсэн байхтай олонтаа таарч байв.
- Ассемблерийн чиглүүлэгчүүдийг бид өргөн хэрэглэх болно. END буюу программын төгсгөл заагч нь программ бүрт зайлшгүй шаардлагатай. Нэгэнт манай программын санах ой 2кбайтаар хязгаарлагдах учир санах ойгоо ухаалгаар зохицуулж өгөхгүй бол томоохон систем хийхэд санах ой хүрэлцэхгүй болох тохиолдол гардаг.

1.3.2. Микроконтроллерийг программчлах

Ассемблерийн хэл дээр программ бичсэний дараагийн шатны үйл ажиллагаа бол уг программаа машин код руу хөрвүүлэх, улмаар программчлагч төхөөрөмжийн тусламжтай микроконтроллерийн программын санах ойд шарж суулгах явдал юм. Ингэснээр бяцхан микроконтроллер маань бидний хүссэн үйлдлийг бидний зохиосон программын дагуу гүйцэтгэнэ.



Зураг 1.5. Микроконтроллерийг программчлахад
хэрэглэгдэх программ хангамжууд

Бид микроконтроллерийг программчлахын тулд үндсэндээ 2 төрлийн программ ашигладаг. Үүнд:

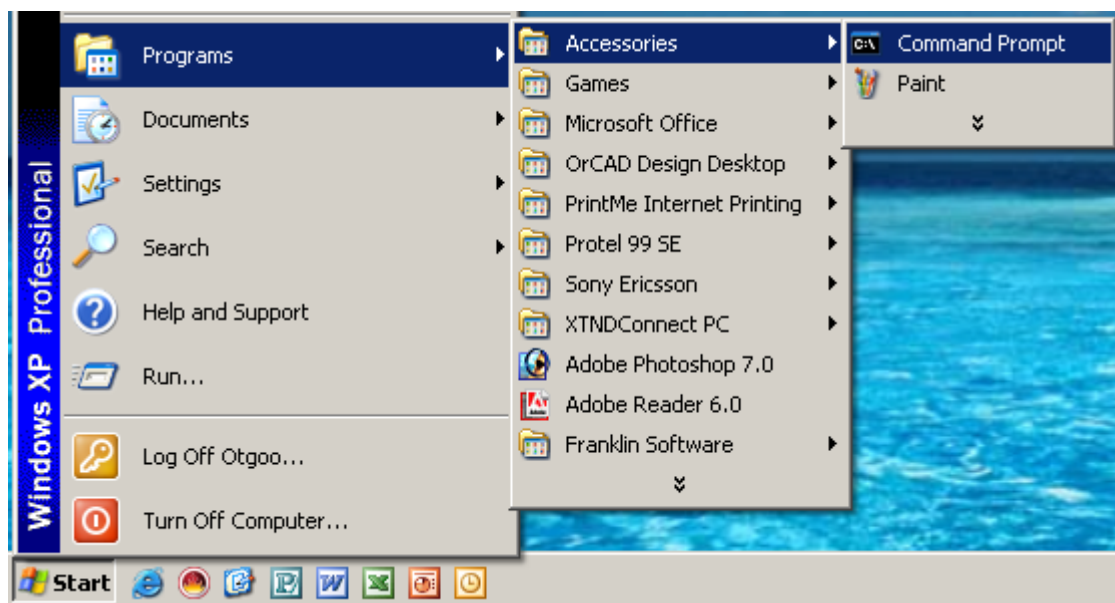
1. A.BAT, A51.EXE, OH.EXE файлууд нь Ассемблерийн хөрвүүлэгч программууд.
2. PY.BAT, PY2051.EXE файлууд нь микроконтроллерт программыг шарах буюу программын санах ойг программчлах зориулалттай программ болно.

Микроконтроллерийг программчилж эхлэхийн тулд дээрх программуудыг агуулж байгаа **ASM51** фолдерийг номтой хамт ирсэн CD-нээс компьютерийнхаа C: дискэнд хуулахад хангалттай. Бичсэн программаа Ассемблерийн хөрвүүлэгч программыг ашиглан хөрвүүлэнэ. Notepad-аар үүсгэсэн “a51” өргөтгөлтэй программыг A51.EXE программын тусламжтай хөрвүүлэхэд “LST” болон “OBJ” өргөтгөлтэй файлууд үүснэ. “LST” өргөтгөлтэй файл нь LIST буюу хөрвүүлэлтийн явцад үүсэх машин код болон ассемблерийн мнемоникийг хамтад нь агуулсан текст файл юм. Хэрэв бичсэн программанд алдаа байвал энэ алдааны тухай мэдээлэл энэхүү LST файланд тодорхой бичигдэнэ. Хэрэв алдаа гараагүй бол “OBJ” өргөтгөлтэй Object файл хөрвүүлэгдэн үүснэ. Энэ файлыг OH.EXE (Object to Hex converter) программаар хөрвүүлэн “HEX” өргөтгөлтэй 16-тын тоон файл үүсгэнэ. Энэ файлыг программчлагч төхөөрөмжийн тусламжтай микроконтроллерт шардаг. Энэхүү хөрвүүлэгч хоёр программыг дэс дараалуулан ажиллуулах ажиллагааг автоматаар дуудах зориулалттай A.BAT нэртэй batch файлыг энд үүсгэсэн байна.

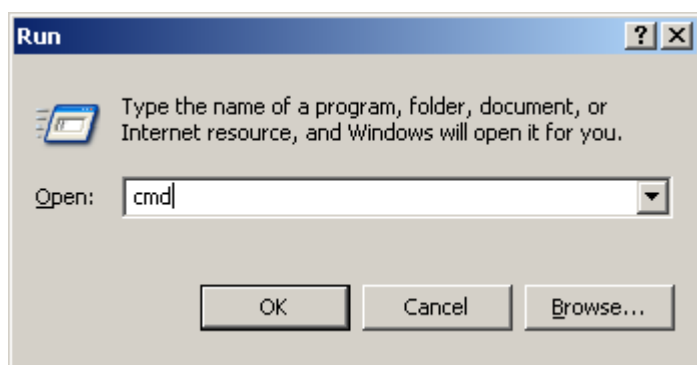
Дараах зааварчлагаануудыг яг даган биелүүлэхгүй бол бидний микроконтроллер программчлах ажиллагаа бүтэлгүйтэх учир маш хянуур хийх хэрэгтэйг анхааруулая.

Дээрх ASM51 фолдерт XP_Driver дэд фолдерийг хуулсан байгаа. Энэ фолдер дотор Windows XP үйлдлийн системтэй компьютер дээр микроконтроллерийг программчлагч төхөөрөмжийг ажиллуулахад шаардлагатай драйвер программын суурилуулагч InstallDriver.exe программ агуулагдана. Хэрэв та XP үйлдлийн систем дээр (Windows 2000 мөн хамаарна) ажилладаг бол энэхүү драйвер программыг суулгаад компьютерээ дахин эхлүүлэх (Restart) хэрэгтэй. Ингэснээр микроконтроллерийг программчлах төхөөрөмж ажиллахад бэлэн болно.

Бидний хэрэглэх MCS-51 Macro Assembler программ нь MS-DOS үйлдлийн системийн орчинд ажилладаг программ хангамж юм. Иймд микроконтроллерийг программчлахад хэт орчин үеийн хүчтэй компьютер харин ч тохиромжгүй. Windows 98 болон Windows XP үйлдлийн системтэй компьютер ашиглаж болно. Бүр өмнөх үеийн MS-DOS үйлдлийн систем бүхий компьютер ч ашиглаж болно. Энд бид Windows XP үйлдлийн системтэй компьютер дээр хэрхэн ажиллахыг жишээ болгон авлаа. Windows үйлдлийн системийн орчин үеийн хувилбарууд нь MS-DOS үйлдлийн системийн программ хангамжуудыг дэмжин ажиллахдаа төдийлэн сайнгүй боловч DOS үйлдлийн системийн интерфэйсийг өөртөө агуулсан байдаг. Иймд DOS үйлдлийн системийн интерфэйс болох Command Prompt-ийг дуудаж ажиллуулна.



Зураг 1.6. XP үйлдлийн системд DOS интерфэйсийг ажиллуулах



Зураг 1.7. Run цонхыг ашиглаж дуудах

Энэ интерфэйсийг дуудах өөр нэг арга бол Start цэсийн Run... сонголтонд cmd гэж бичсэнээр ажиллуулж болно. (Зураг 1.7)

Ингээд өмнө хуулсан ASM51 фолдер луугаа очих хэрэгтэй. Ингэхийн тулд **cd\asm51** командыг оруулж Enter товчийг дарна. Энэ бичсэнээс өмнөх хэсэг нь тухайн компьютерээс хамаарч өөр байж болох бөгөөд автоматаар гарч ирсэн байна. .

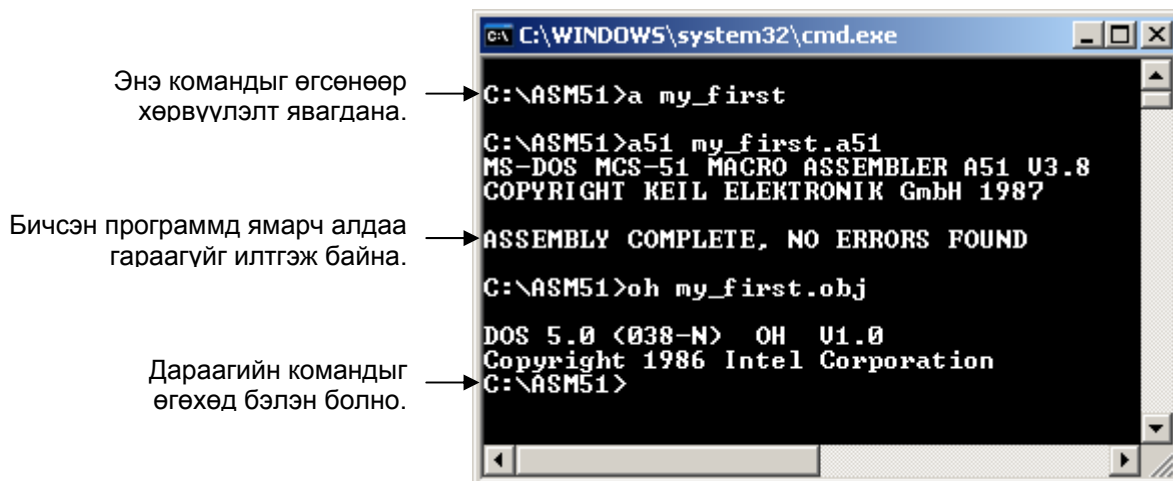


Зураг 1.8. Ажиллах фолдер луу шилжих

DOS үйлдлийн системийн Command Prompt интерфэйсээс ассемблерийн хөрвүүлэгчийг дуудаж ажиллуулана. Мөн ассемблерийн хөрвүүлэгчийн үүсгэсэн Object файлийг дахин хөрвүүлэх ажиллагааг ч эндээс хийдэг. Энэ хоёр ажиллагааг байнга хамтад нь хийх шаардлагатай учир автоматаар хоёуланг нь ажиллуулах A.BAT файлийг үүсгэн ашигладаг.

Одоо хамгийн анхны бяцхан программаа хөрвүүлцгээе. Ассемблерийн хөрвүүлэгч A51.EXE болон объект хөрвүүлэгч OH.EXE программуудыг дэс дараалуулан автоматаар ажиллуулах A.BAT файлыг хөрвүүлэх ассемблерийн хэл дээр бичигдсэн файлынхаа нэртэй оруулна. Энэ нь:

C:\ASM51>a my_first мөрийг оруулаад enter товчийг дарснаар хэрэгжинэ. (Зураг 1.9) Энд **C:\ASM51>**–ийг бичих шаардлагагүй, цаанаасаа автоматаар гарч ирсэн байгаа.



Зураг 1.9. Ассемблерийн программыг хөрвүүлэх ажиллагаа

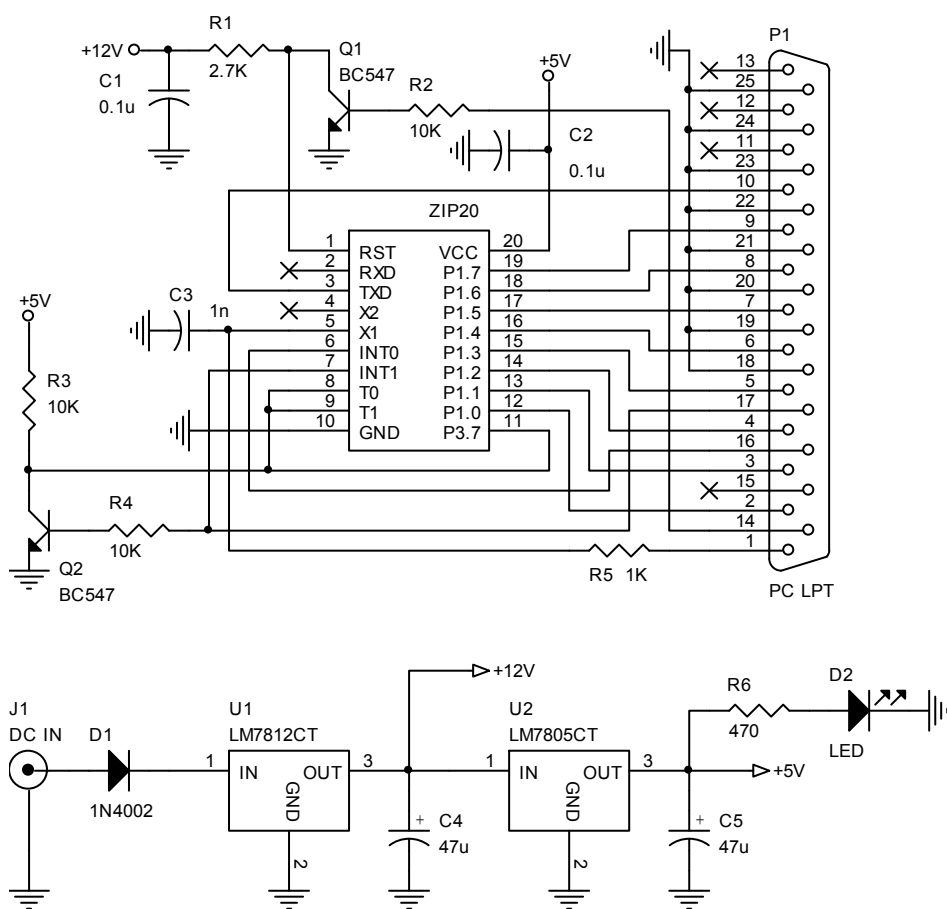
Урьдчилан бэлдсэн A.BAT файлийг хэрэглэхгүйгээр:

C:\ASM51>a51 my_first.a51 командийг гараараа оруулж өгсөний дараа алдаагүй хөрвүүлэгдсэн тохиолдолд:

C:\ASM51>oh my_first.obj командийг мөн гараараа оруулж өгнө. Ийнхүү хоёр аргын аль нэгээр нь амжилттай хөрвүүлсэний дараа Windows Explorer программын тусламжтай ASM51 фолдероо нээж үзвэл MY_FIRST.LST, MY_FIRST.OBJ, MY_FIRST.HEX гэсэн 3 шинэ файл үүссэн байх болно. Бидний хэрэглэх файл бол MY_FIRST.HEX юм.

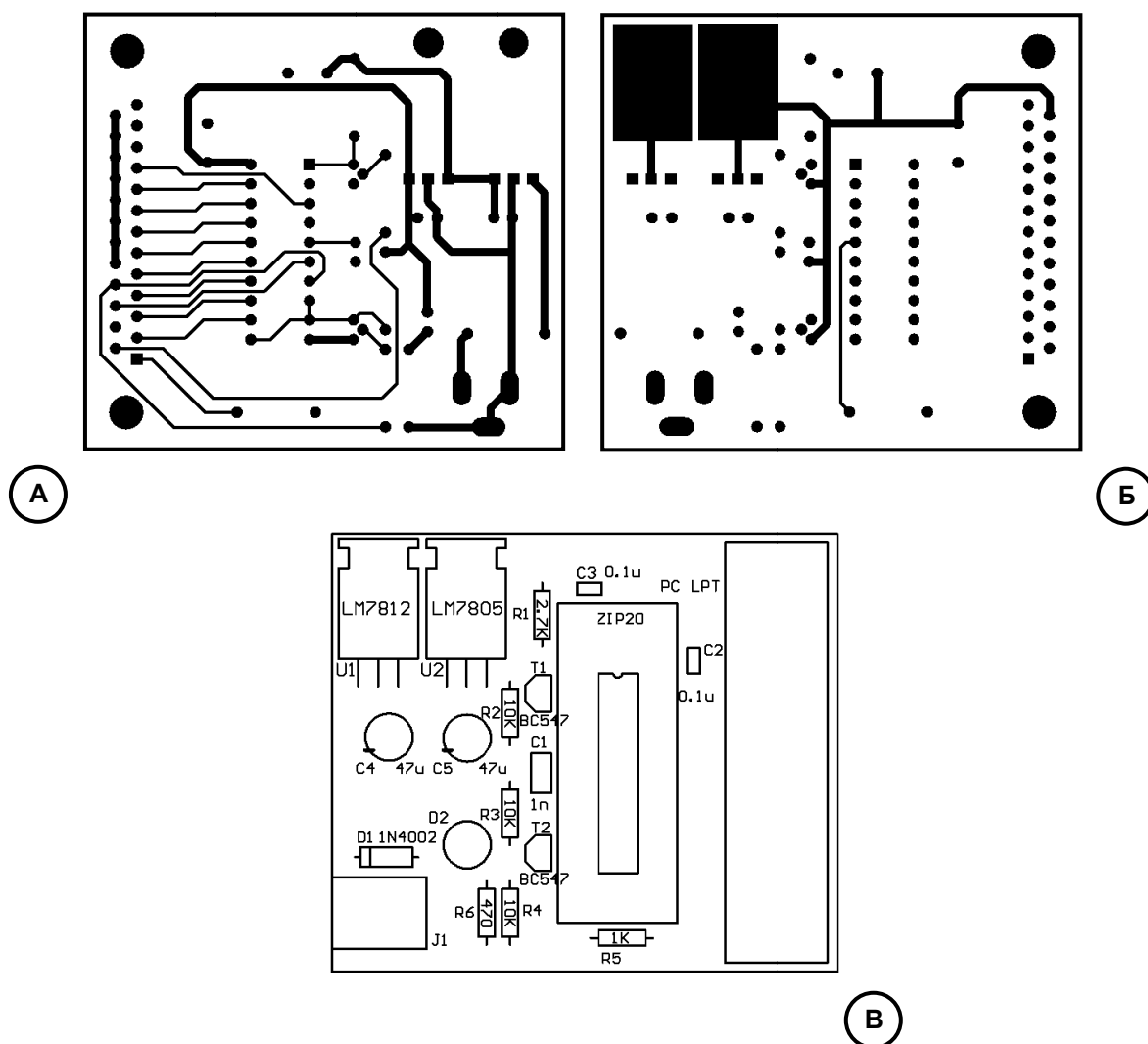
1.3.3. Микроконтроллер программчлагч төхөөрөмж

Зураг 1.10-т компьютерийн параллель портыг ашиглан Атмел фирмийн AT89C2051 микроконтроллерийг программчлагч программаторын зарчмын схемийг үзүүлэв. Энэхүү хэлхээ нь тун хялбар хэлхээ бөгөөд зөвхөн дээр дурдсан микроконтроллерийг программчлана гэдгийг анхаарах хэрэгтэй. Программаторыг компьютерийн параллель портонд түүний DB25F төрлийн залгуураар нь шууд залгаж болохоос гадна уртасгагч кабель ашиглаж болно. Гэхдээ хэт урт кабель хэрэглэвэл программатор буруу ажиллаж болно. Программаторын тэжээлийн үүсгүүр нь тогтмол 14-18V хүчдэлтэй адаптороос тэжээгдэнэ.



Зураг 1.10. Хялбар программаторын зарчмын схем

Тэжээлийн хэсэг нь интеграл хүчдэл тогтворжуулагчууд болох 7812, 7805 элементүүдийг ашиглана. Энд голдоо эерэг потенциал бүхий залгууртай адапторыг хэрэглэж болох ба буруу туйлтай адаптор залгавал D1 диод хаагдаж программатор дээр тавигдсан интеграл тогтворжуулагчууд болон AT89C2051 микроконтроллерийг шатахаас хамгаална. Харин D2 гэрэлт диод нь тэжээлийн хүчдэл байгаа эсэхийг илтгэх юм. Энэ хэлхээнд +12V хүчдэл нь микроконтроллерийн программчлагч хүчдэл V_{PP} -ийн үүрэг гүйцэтгэх ба +5V хүчдэл нь программчлагдах микроконтроллерийг тэжээх үүрэгтэй. Хэрэв тэжээлд дор хаяж 14V бүхий адаптор хэрэглэхгүй бол 7812 хүчдэл тогтворжуулагчаас 12V-оос бага хүчдэл гарч, улмаар контроллер программчлагдахгүй байж болно.

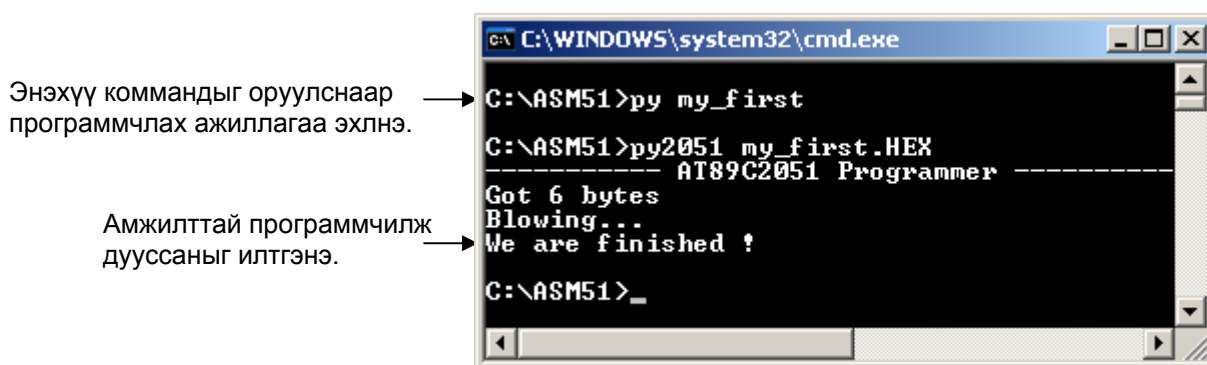


Зураг 1.11. Программаторын хэвлэмэл хавтан
А. Доод тал, Б. Дээд тал, В. Элементүүдийн байрлал

Одоо программаторын элементүүдийг хэвлэмэл хавтан дээр угсрах үлдлээ. Угсрахдаа дараахь зүйлсийг сайтар анхаарах хэрэгтэй. Гагнуурын алхныхаа хошууг үргэлж цэвэрхэн байлгахын зэрэгцээ маш сайн чанарын дундаа давирхайтай тугалга хэрэглэнэ. Элементүүдээ хавтандаа сайн суулгаж нямбай гагнах хэрэгтэй. U1, U2 элементүүд нь хоорондоо тун ижил учир байрыг нь сольж гагнуузай. Нэг нь +12V хүчдэл тогтворжуулдаг, нөгөө нь +5V хүчдэл тогтворжуулдаг учир байр нь солигдвол программатор маань ажиллахгүй болно. D1, D2 диодууд болон C4, C5 конденсаторууд нь туйлтай элементүүд учир холбогдох туйлыг нь сайтар шалгаж байж гагнана. Конденсаторуудын сөрөг туйлыг түүний биен дээр тэмдэглэсэн байгаа. Тэр тэмдэглэгээ нь хэвлэмэл хавтан дээрх конденсаторын дугуй дүрсэн дээр байгаа “-” тэмдэгтэй таарч байх ёстой. D1 диодны хувьд хэвлэмэл хавтан дээрх диодны дүрсэн дээр нэг талд нь тавигдсан тэмдэг шиг цагаан бүслүүр диодны биен дээр байгаа. Харин D2 диодны хоёр хөл нь урт богино байгааг анзаарах хэрэгтэй. Урт хөл нь C5 конденсатор луу харж, богино хөл нь R6 эсэргүүцэгч рүү харж холбогдоно. Бүр итгэлтэй байя гэвэл J1 залгуур, D1 диод, U1, U2 тогтворжуулагчууд, C4, C5 конденсаторуудыг гагнаад J1 залгуурт тэжээлийн адапторыг залга.

Ингээд D2 диодыг гагнах нүх рүү нь суулгавал зөв суулгасан тохиолдолд асах болно. Адаптороо салгаад диодоо гагнах хэрэгтэй.

Үлдсэн элементүүдийг нямбай гагнаад программатор бэлэн болно. Эсэргүүцэгчүүд болон C1, C2 конденсаторууд тодорхой туйлгүй учир хааш нь ч харууж гагнаж болно. Харин T1, T2 транзисторуудыг зурагт харуулсан байрлалаар нь гагнавал алдаа гарахгүй. Хамгийн сүүлд ZIP20 суурийг LPT портонд залгах DB25F залгууртай гагнах нь тохиромжтой. Одоо бэлэн болсон программаторыг шалгая. Программаторийг компьютерийн принтер холбодог LPT портонд залгаад AT89C2051 микроконтроллерийг түүний ZIP20 сууринд суурилуулна. Дараа нь тэжээлийн адаптороо залга. Ингээд өмнө нь ассемблерийн хөрвүүлэлт хийж байсан Command Prompt буюу DOS интерфэйсийг ажиллуулаад C:\ASM51 фолдер луу очих хэрэгтэй.



Зураг 1.12. Программаторыг ажиллуулах программ хангамж

Энд “py my_first” командыг оруулбал программатор ажиллаж эхлэнэ. Хэрвээ “We are finished !” мэдээллийн оронд “Programming aborted.” мэдээлэл дэлгэцэнд гарч байвал программчлах ажиллагаа амжилттай болоогүйг илтгэнэ. Программатор буруу угсрагдсан, программчлагдах AT89C2051 микроконтроллер ажиллагаагүй байж болохоос гадна тэжээлийн адапторын хүчдэл +14V-оос бага байх үед ийм алдаа гардагыг анхаар.