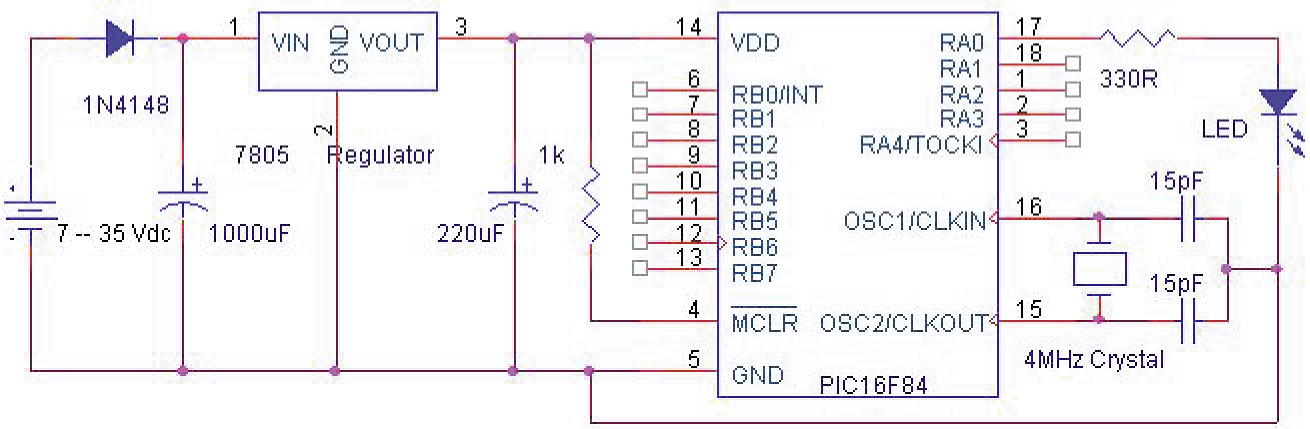


## MICROCONTROLLERS

14 ලිපිය - 2 කොටස

# පරිගණක භාෂාව ඇසුරෙන් ක්‍රමලේඛ ගොඩනැංවීම

ක්ෂුද්‍ර පාලක සහ ක්ෂුද්‍ර සකසන (Microcontrollers and Microprocessors) සඳහා ක්‍රමලේඛ සැකසීමට යොදාගන්නා පරිගණක භාෂා පිළිබඳව අපි පසුගිය සතියේ සාකච්ඡා කළෙමු. එහි දී විශේෂයෙන් ම සඳහන් කළ කරුණක් වූයේ පහළ මට්ටමේ පරිගණක භාෂාවක් (උදා : Assembly language) යොදාගත හොත් ක්‍රමලේඛ සකසන්නාට දෘඪාංග කෙරෙහි වැඩි බලපෑමක් සිදු කළ හැකි බවයි. තවත් ලෙසකින් කිව හොත් ක්ෂුද්‍ර පාලකයේ හෝ ක්ෂුද්‍ර සකසනයේ අභ්‍යන්තරයේ සිදුවන්නේ කුමක්දැයි යන්න පිළිබඳ පැහැදිලි අවබෝධයක් ක්‍රමලේඛ සකසන්නාට ලැබේ. එම නිසා වඩාත් කාර්යක්ෂම වූත් ඵලදායී වූත් ක්‍රමලේඛයක් ගොඩනැංවීමට අවස්ථාව ලැබේ. එහෙත් එවන් පරිගණක භාෂාවක් උපයෝගී කරගෙන ක්‍රමලේඛ ගොඩනැංවීම තරමක් අසීරු කාර්යයකි. යම් කිසි තැනක ගැටලුවක් මතු වුව හොත් එය නිරාකරණය කරගැනීමට සෑහෙන කාලයක් ගත වේ. එමෙන්ම ක්‍රමලේඛ ගොඩනැංවීමට ගත වන කාලය ද සාපේක්ෂව ඉහළ මට්ටමක පවතී. තව ද ආධුනිකයකුට එම විෂය තේරුම් ගැනීමට ද සෑහෙන කාලයක් ගත වේ. ඒ හේතුවෙන් ක්ෂුද්‍ර සකසන සඳහා ක්‍රමලේඛ (පරිගණක ක්‍රමලේඛ) සැකසීම සඳහා මේ වන විට පහළ මට්ටමේ පරිගණක භාෂා යොදා ගැනෙනුයේ ඉතා ම අඩුවෙනි.

එහෙත් ක්ෂුද්‍රපාලක (Microcontrollers) සඳහා ක්‍රමලේඛ ගොඩනැංවීමට පහළ මට්ටමේ පරිගණක භාෂා තවමත් යොදාගැනේ. ඉදිරියටත් එය එසේ වීමට පුළුවන. මේ වන විට පහළ මට්ටමේ පරිගණක භාෂාවක් වන Assembly language ඇසුරින් PIC මයික්‍රොකොන්ට්‍රෝලර් සඳහා ක්‍රමලේඛ ගොඩනංවන ආකාරය අපි විස්තර කර ඇත්තෙමු. එබැවින් අප ගේ මීළඟ උත්සාහය වනුයේ ඉහළ මට්ටමේ පරිගණක භාෂාවන් යොදාගෙන PIC මයික්‍රොකොන්ට්‍රෝලර් සඳහා ක්‍රමලේඛ ගොඩනංවන ආකාරය පිළිබඳව පැහැදිලි අවබෝධයක් පාඨක ඔබට ලබා දීමට ය.

වර්තමානයේ දී ක්ෂුද්‍ර සකසන සඳහා (පරිගණක සඳහා) ක්‍රමලේඛ ගොඩනැංවීමට සකස් කරන ලද ඉහළ මට්ටමේ පරිගණක භාෂා විශාල ප්‍රමාණයක් පවතී. C, C++, C#, Basic, Java, Delphi ඉන් සමහරකි. C++ සහ C# යනු C නැමැති පරිගණක භාෂාවේ ම වැඩිදියුණු කළ අවස්ථාවන් ය. එමෙන්ම ක්ෂුද්‍රපාලක (Microcontrollers) සඳහාත් එවැනි ඉහළ මට්ටමේ පරිගණක භාෂා නිර්මාණය කොට ඇත. උදාහරණ ලෙස MikroC, CCSX, PIC C, PICBASIC, Micro Pascal පරිගණක භාෂා දැක්විය හැකි ය. ඉන් Micro C, CCSX සහ PIC C යනු C පරිගණක භාෂාව මත පදනම්ව එක් එක් සමාගම් මගින් නිපදවන ලද මෘදුකාංග (C Compilers) වන අතර PIC BASIC යනු Basic නැමැති පරිගණක භාෂාව මත පදනම් ව සකස් කරන ලද්දකි. Micro Pascal යනු ද ඉහළ මට්ටමේ පරිගණක භාෂාවක් වන Pascal භාෂාව මත පදනම් ව සකස් කරන ලද්දකි. මෙලෙස ඉහළ මට්ටමේ භාෂා කිහිපයක් ම දක්නට ලැබෙන බැවින් ඉන් සුදුසු එක් භාෂාවක් තෝරාගත යුතු ව ඇත.

එබැවින් අප ගේ ඉදිරි කටයුතු සඳහා C පරිගණක භාෂාව තෝරාගැනීමට අපි අදහස් කළෙමු. එය මත පදනම් ව PIC මයික්‍රොන්ට්‍රෝලර් සඳහා සකස් කරන ලද මෘදුකාංග (Compiler) කිහිපයක් ම ඉහත සඳහන් විය. එයින් Mikro C අපි තෝරාගන්නෙමු. එම මෘදුකාංගය www;mikroe.com/en/compilers/microc/pic/download.htm

නමැති වෙබ් පිටුවට පිවිසීමෙන් ලබාගත හැකි ය. එය නොමිලයේ ලබා ගත හැකි ය. එහෙත් ලිවිය හැකි ක්‍රමලේඛයේ ප්‍රමාණය 2k (ආසන්න වශයෙන් උපදෙස් 2000ක්) දක්වා සීමා වේ. අප විසින් ලියනු ලබන බොහෝ ක්‍රමලේඛ එම සීමාව ඉක්මවා නො යන බැවින් ගැටලුවක් ඇති නො වේ. අප විසින් මුලින් ම ඉදිරිපත් කරන ලද නිර්මාණය වන ආලෝක විමෝචක දියෝඩයක් තිබීම හා දැල්වීමට යොදාගෙන Mikro C පිළිබඳව හැදෑරීම අරඹමු. රූප සටහන අංක 1න් අදාළ පරිපථ සටහන දැක්වේ.

තත්පර 1කට වරක් LED බල්බය දැල්වීම හා නිවීම සඳහා Mikro C මෘදුකාංගය යොදාගෙන සකස් කරන ලද ක්‍රමලේඛය පහත දැක්වේ.

```
Void main ()
  { PORTA = 0;
    TRIS A = 0;
        While (1)
            { PORT A = ~PORT A;
              Delay_ms(1000);
            }
        }
```

mikroc මෘදුකාංගය ඔබේ පරිගණකය තුළ ස්ථාපනය කර එහි Project → New Project හරහා ගොස් නව ව්‍යාපෘතියක් අරඹන්න. එවිට ලැබෙන මුහුණතේ අවශ්‍ය කරන දත්ත ලබා දීමෙන් පසුව ඉහත සඳහන් කළ ක්‍රමලේඛය සටහන් කිරීමට අවශ්‍ය අවස්ථාව ලැබේ. එම ක්‍රමලේඛය නිවැරදිව සටහන් කර Project → Build හරහා ගොස් එම ක්‍රමලේඛය යන්ත්‍ර භාෂාවට (Machine) හරවා ගන්න. එය (Compile කිරීම ලෙස හැඳින්වේ.) එවිට .hex යන දිගුව සහිතව අදාළ ක්‍රමලේඛයේ මයික්‍රොකොන්ටෝලරයට තේරුම් ගත හැකි උපදෙස් ගොනුව ලැබේ. ඉන් පසුව එය ක්‍රමලේඛනය කිරීමේ උපකරණයක් භාවිතයෙන් මයික්‍රොකොන්ට්‍රෝලරය තුළට ඇතුළත් කර ගන්න. ඉන් පසුව නිවැරදිව පරිපථය සකස් කොට විදුලිය ලබාදුන් විට LED බල්බය තත්පරයක් දැල්වී තිබී ඊළඟ තත්පරයේ දී නිවේ. මෙය දිගින් දිගට ම නො නවත්වා සිදු වේ.

ඉහත සඳහන් කළ ක්‍රමලේඛයන් ඒ සඳහා ගොඩනැංවූ 4 වන ලිපියේ සඳහන් ක්‍රමලේඛයක් සංසන්දනය කර බලන්න. ඒ දෙකෙන් ම සිදු වනුයේ එක ම කාරණය වුවත් ක්‍රමලේඛ දෙකෙහි විශාල වෙනස්කමක් පවතී. ඉහත දැක්වූ ක්‍රමලේඛය ඉතා ම සරල බව පෙනී යනු ඇත.

එහි Void main() යනු main නැමැති ප්‍රධාන ක්‍රියාවලිය හඳුන්වා දී ඇති ආකාරයයි. මෙලෙස වරහන් දෙකක් සමග හඳුන්වා දෙන ක්‍රියාවලි Functioins ලෙස නම් කර ඇත. එහි void මගින් කියවෙනුයේ මෙම ක්‍රියාවලිය මගින් කිසි ම අගයක් පිට නො කරනවා වැනි අදහසකි. ඒ පිළිබඳ වැඩි විස්තර ඉදිරියේ දී සාකච්ඡා කරමු. ඉන් පසුව විවෘත කර තිබෙන සඟල වරහන හා ඒ කෙළින් ම පහළින් වසා තිබෙන සඟල වරහන අතර ඇති උපදෙස් මාලාව මෙමෙ main ක්‍රියාවලියට අයත් වේ. එහි තිබෙන මුල් ම උපදෙස් දෙක වන PORT A = 0; යන TRIS A + 0; මගින් A තොටුපළෙහි අග්‍ර ප්‍රතිදානයක් ලෙස සකස් කර ඒවායේ වෝල්ටීයතාවන් 0 ලෙස සකස් කෙරේ. හැම උපදෙසක් අවසානයේ දී ම ; සලකුණ යෙදීම අත්‍යවශ්‍ය වේ. එය C භාෂාවේ රීතියකි. ඉන් පසුව While(l) මගින් කියවනුයේ ඊට පහළින් පුරවා ඇති සහ අවසන් කර ඇති සඟල වරහන් තුළ තිබෙන උපදෙස් නිරන්තරයෙන් ක්‍රියාත්මක කළ යුතු බවයි. ඒ යටතේ එන පළමු උපදෙස PORT A = ~POTR A; යන A තොටුපළේ අග්‍රවල තිබෙන තාර්කික අගය "1" හෝ "O" ලබාගෙන එහි විලෝමය එනම් "0" හෝ "1" නැවත එම අග්‍රවලට ම ලියන ලෙස දෙනු ලබන උපදෙසයි. එනම් LED බල්බය දැල්වී තිබේ නම් නිවන ලෙසත් නිවී තිබේ නම් දල්වන ලෙසත් දෙන උපදෙසකි. ඊළඟට තිබෙන Delay_ms(1000); උපදෙස මගින් තත්පරයක හෙවත් මිලි තත්පර 1000ක කාල පමාවක් ලබා දේ. මෙම උපදෙස් දෙක දිගින් දිගට ම ක්‍රියාත්මක වේ.

මොරටුව විශ්වවිද්‍යාලයයේ විද්‍යුත් හා විදුලි සංදේශ අංශයේ
ගාමිණී ජයසිංහ
කෝලිත ධර්මප්‍රිය