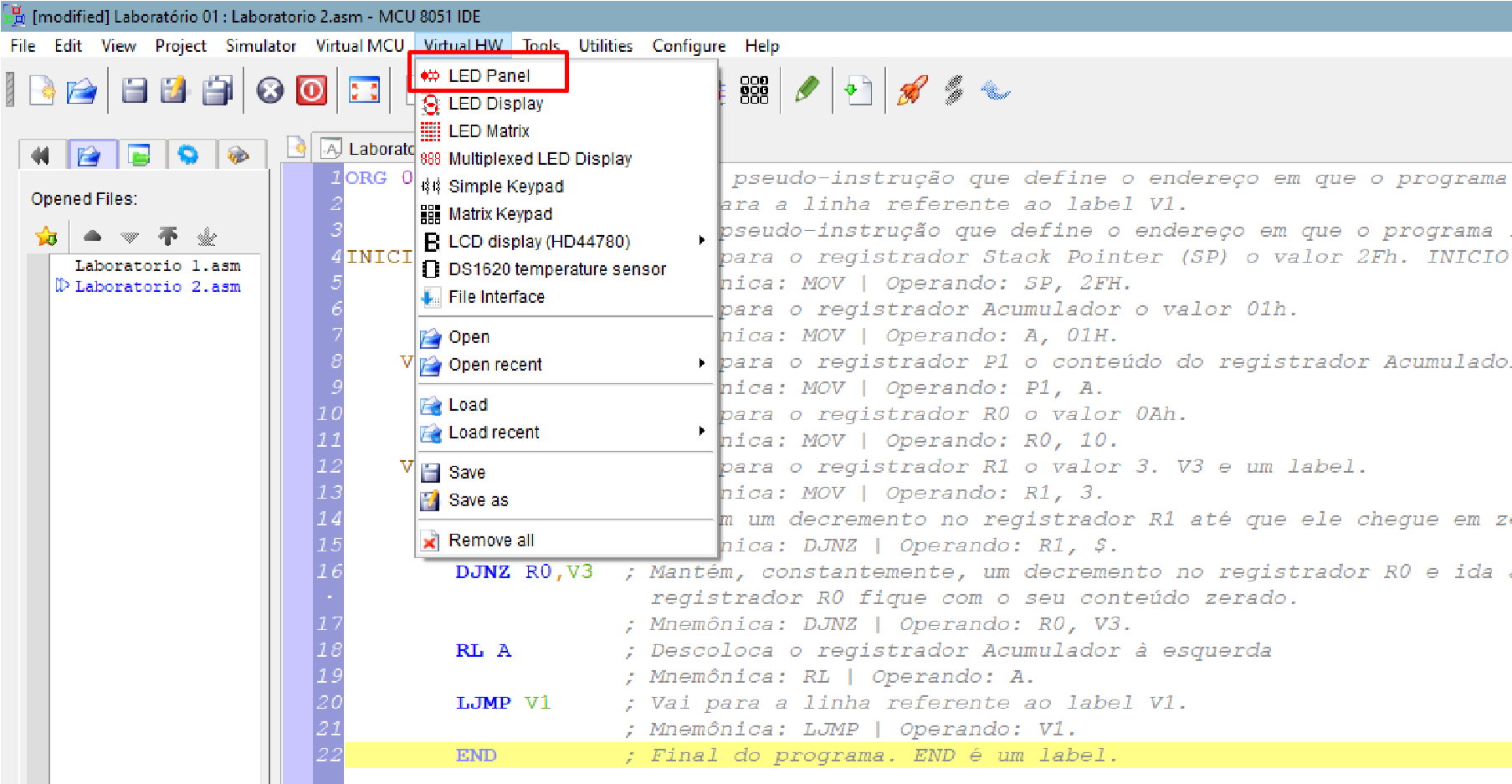
1. Resumo:

Neste trabalho em questão é possível observar alguns conceitos básicos e importantes sobre a programação do micro-controlador MCS51 em Assembly a partir do simulador MCU8051 IDE. Foi possível absorver algumas noções que dizem respeito à rotação de led 's com a utilização do simulador e dos comandos da linguagem Assembly.

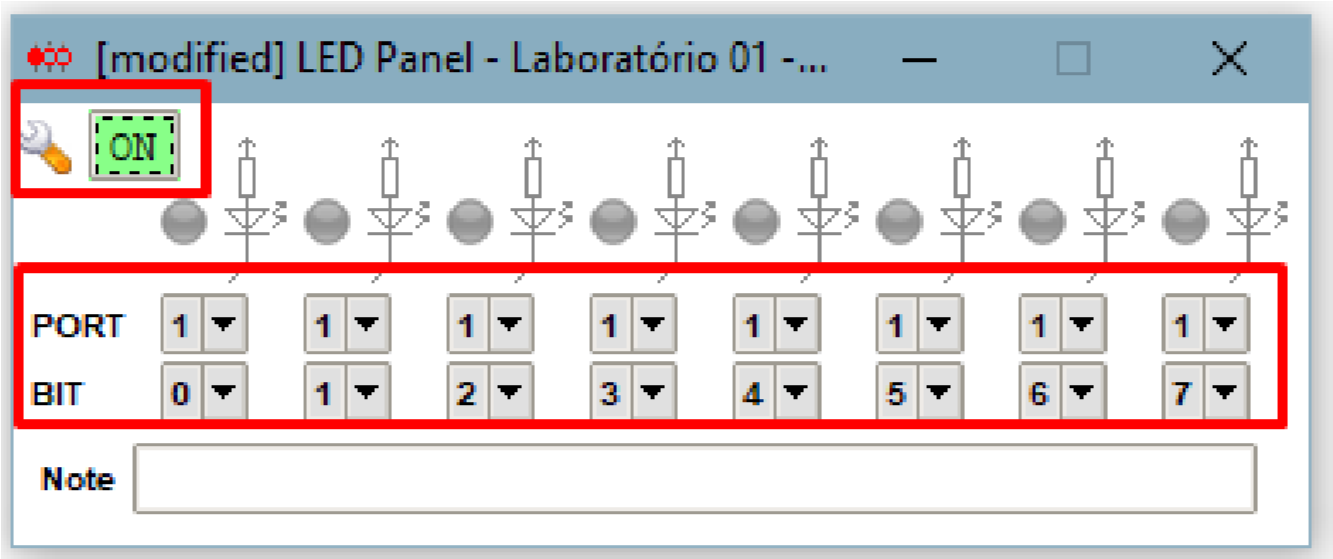
1. Introdução:

O presente trabalho tem como objetivo introduzir o estudo do microcontrolador MCS51 a partir do uso do simulador MCU8051 IDE. Tendo o Assembly como linguagem dominante na programação do referente microcontrolador, se faz necessário também o estudo da mesma a partir de consultas em seu Datasheet, que é nada mais do que uma folha com dados e especificações técnicas e de desempenho do produto levado em consideração, que no nosso caso é o microcontrolador MCS51.

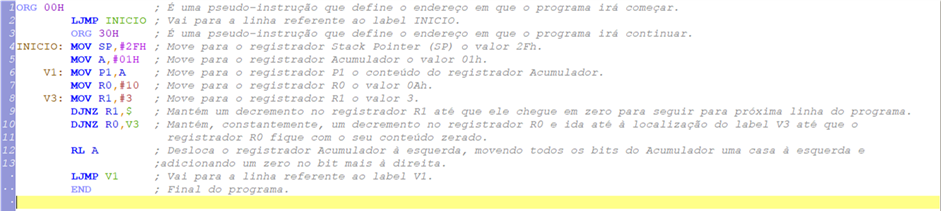
1. Métodos:
   1. Objetivos: O experimento em questão tem como objetivo induzir o estudo e a prática da programação do microcontrolador MCS51 a partir do simulador MCU8051 IDE com o uso da linguagem. Assembly. Tornando possível estudar a rotação de led 's no simulador.
   2. Equipamentos Necessário: Computador com MCS 51 instalado.
   3. Procedimentos:
      1. Inicialmente, é necessário fazer a ligação dos led’s nos bits repectivos das portas.



Configura-se o painel ativando cada um dos oito led’s a uma porta e um bit do microcontrolador.

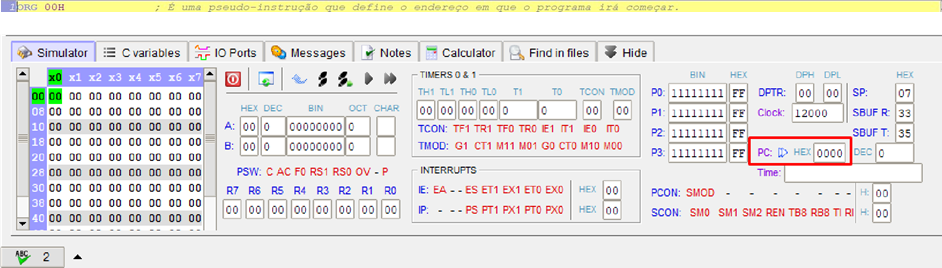


* + 1. Programa em LR
       1. ORG 00H: É uma pseudo-instrução que define o endereço em que o programa irá começar ou continuar.
       2. LJMP INICIO: É um label que tem como função apontar para um determinado trecho de código.
       3. ORG 30H: É uma pseudo-instrução que define o endereço em que o programa irá começar ou continuar.
       4. INICIO: MOV SO,#2FH: Move para o registrador SP um determinado valor.
       5. MOV A,#01H: Move para o registrador Acumulador um determinado valor.
       6. V1:MOV P1,A: Move para o registrador P1 o conteúdo do registrador Acumulador.
       7. MOV R0,#10: Move para o registrador P1 o conteúdo do registrador Acumulador.
       8. V3:MOV R1,#3: Move para o registrador R1 um determinado valor.
       9. DJNZ R1, $: Mantém um decremento no registrador R1 até que ele chegue a zero, só assim o programa dará procedência para as próximas linhas de comando.
       10. DJNZ R0, V3: Mantém, constantemente, um decremento no registrador R0 e ida até à localização do label V3 até que o registrador R0 fique com o seu conteúdo zerado.
       11. RL A: Desloca o registrador Acumulador à esquerda, movendo todos os bits do Acumulador uma casa à esquerda e adicionando um zero no bit mais à direita.
       12. LJMP V1: É um label que tem como função apontar para um determinado trecho de código.
       13. END: Define o final do programa.
    2. Funcionamento.



Comentários, execução do código:

* + - 1. ORG 00H: Indicador de início para o programa começar no endereço 0 H e, com PC = 0;



* + - 1. LJMP INICIO: Aponta o programa para a linha do Label INICIO.



* + - 1. ORG 30H